Xavier Fernández i Marín, Christoph Knill, Christina Steinbacher and Yves Steinebach

Bureaucratic Quality and the Gap between Policy Accumulation and Implementation Capacities — Full report

2023-09-13 14:51

Contents

1 *Introduction* 7

2 *Explain Portfolio size over implementation capacity: Reference model* 9

3 *Explain Portfolio size over implementation capacity: VPI in 2 dimensions* 59

4 *Explain Portfolio size over implementation capacity: Continuous learning (instruments)* 71

5 *Explain Portfolio size over implementation capacity: Steep learning (instruments)* 83

6 *Explain Portfolio size over implementation capacity: Capped learning (instruments)* 95

7 *Explain Portfolio size over implementation capacity: Lag 5 years* 107

8 *Explain Portfolio size over implementation capacity: Lag 7 years* 119

9 *Explain Portfolio size over implementation capacity: No smoothed lag, but plain lag* 131

10 *Explain Portfolio size over implementation capacity: Burden as subtraction* 143

11 Explain Portfolio size over implementation capacity: With state capacity 155

12 Explain Portfolio size over implementation capacity: Model with generosity instead of administrative spending 167

13 Explain Portfolio size over implementation capacity: VPI with 2 values (low/high, and middle category as high) 179

14 Explain Portfolio size over implementation capacity: VPI with 2 values (low/high, and middle category as low) 191

15 Explain Portfolio size over implementation capacity: Continuous learning (targets) 203

16 Explain Portfolio size over implementation capacity: Steep learning (targets) 215

17 Explain Portfolio size over implementation capacity: Capped learning (targets) 227

18 Explain Portfolio size over implementation capacity: With regional authority 239

19 Explain Portfolio size over implementation capacity: Gap standardized 251

20 Explain Portfolio size over implementation capacity: Outcome is standardized implementation burden 263

21 Explain Portfolio size over implementation capacity: Outcome is standardized implementation capacity 275

22 Model comparison 287

23 Implementation capacity: Report of loadings / discriminations and correlations 297

24 Explanatory model of performance 299

Programming environment 317

1
Introduction

This report presents the findings presented in the article "Bureaucratic Quality and the Gap between Policy Accumulation and Implementation Capacities" published at the *American Political Science Review*.

2

Explain Portfolio size over implementation capacity: Reference model

m-312

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  # Outcome variable
  #filter(Case %in% "Regular : Only sample countries : Bounded : Plain ratio") %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  # Sample countries
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  #mutate(Dimension = ifelse(Dimension == "Implementation burden", "Implementation costs", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  #mutate(Dimension = ifelse(Dimension == "Policy design", "Policy feedback", Dimension))
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                 `Top-down (VPI)` = seq(0, 6, by = 1/3),
                 `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                 `Top-down (VPI)` = 0,
                 `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
```

```
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()
```

```
# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%

#            mutate(Measure = "Size",
#                   Measure.label = "Portfolio size",
#                   Version = "Accupol"))

full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
              select(Origin, Destination, p.contiguous),
              by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
#  mutate(`Contiguity dependency (BCG)` = std(contiguity.dependency.id)) %>%
#  mutate(`Contiguity dependency (PS)` = std(contiguity.dependency.ps)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
              ungroup() %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
#  mutate(trade.dependency.id = std(trade.dependency.id)) %>%
#  mutate(trade.dependency.ps = std(trade.dependency.ps)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))
```

```
#interdependency.trade <- interdependency.trade %>%
#  reshape2::acast(Sector ~ Country ~ Year, value.var = "trade.dependency")
#if ( length(which(!dimnames(interdependency.trade)[[2]] == country.label)) > 0) stop("Ep! There is a mistake
```

```
#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)
```

```
Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
```

```
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon
```

```
    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
```

```
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
set.seed(14719)
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

#
# Figure A7
#
L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


#
# Figure A6
#
L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)
```

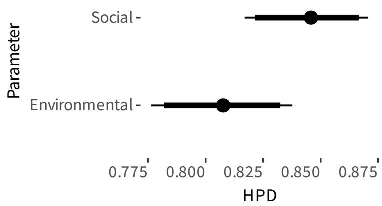


Figure 2.1: Auto-regressive components.

```
ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-312.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")
```

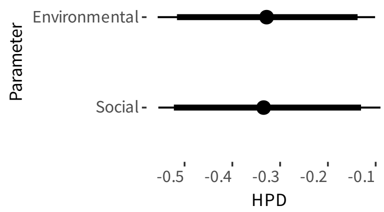


Figure 2.2: Heteroskedasticity controls (Political constraints).

```
#
# Figure A8
#
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-312.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

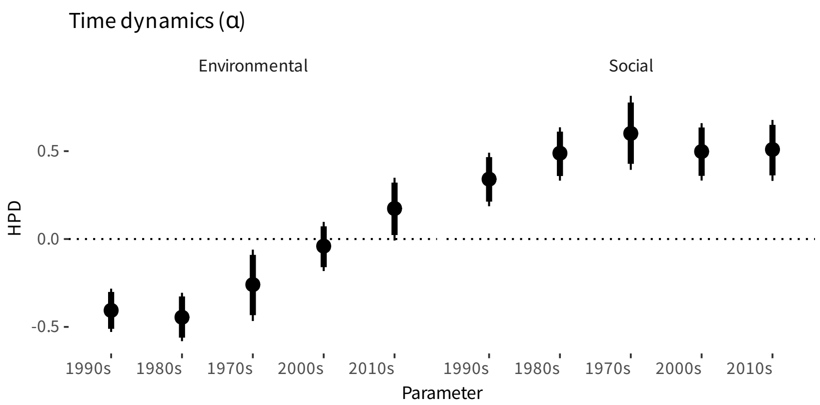


Figure 2.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-312.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))


L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
```

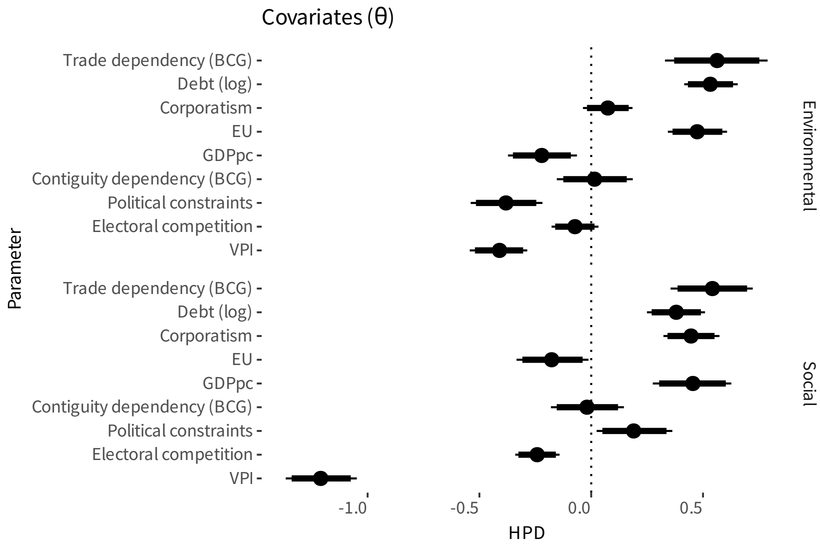


Figure 2.4: Covariates.

```
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

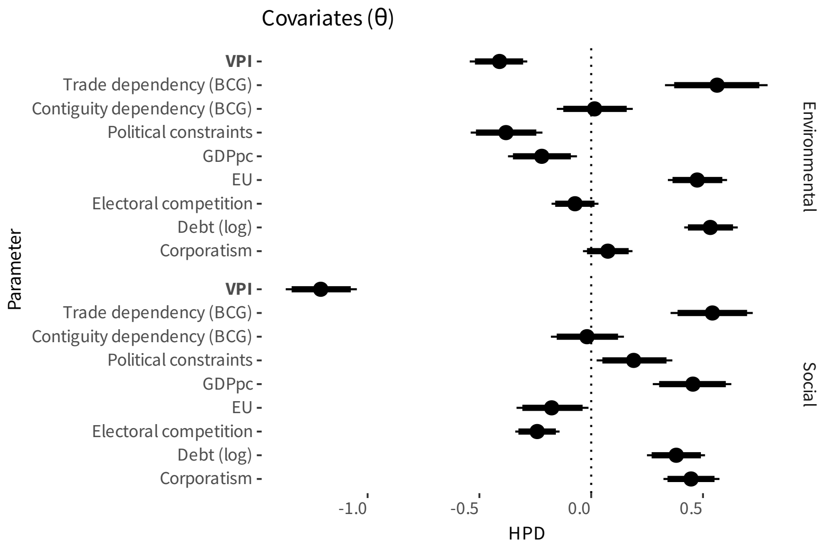


Figure 2.5: Covariates.

```
#
# Figure 5
#
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

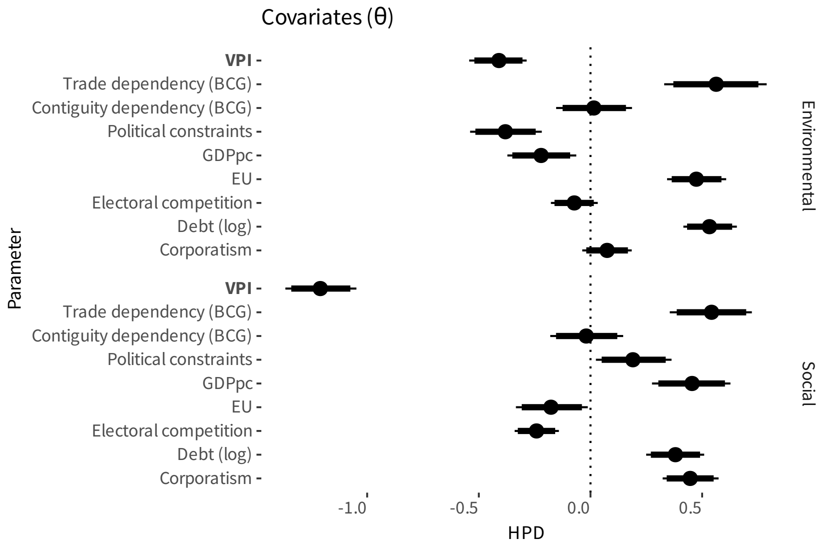


Figure 2.6: Covariates. Figure 5 in the article.

```
L.Sigma <- plab("Sigma", list(Sector = sector.label,
                              Variable.1 = c(variable.label,
                                             variable.sector.label),
                              Variable.2 = c(variable.label,
                                             variable.sector.label)))

S.Sigma <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma)

vcov.Sigma <- ci(S.Sigma) %>%
  select(Sector, Variable.1, Variable.2, vcov = median) %>%
  mutate(vcov = ifelse(Variable.1 == Variable.2, NA, vcov)) %>%
  mutate(Variable.1 = factor(as.character(Variable.1), rev(levels(Variable.1))))

ggplot(vcov.Sigma, aes(x = Variable.2, y = Variable.1, fill = vcov)) +
  geom_raster() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  facet_grid(~ Sector) +
#   scale_fill_continuous_diverging(palette = "Blue-Red", limits = c(-1, 1))
  scale_fill_continuous_diverging(palette = "Blue-Red")
```

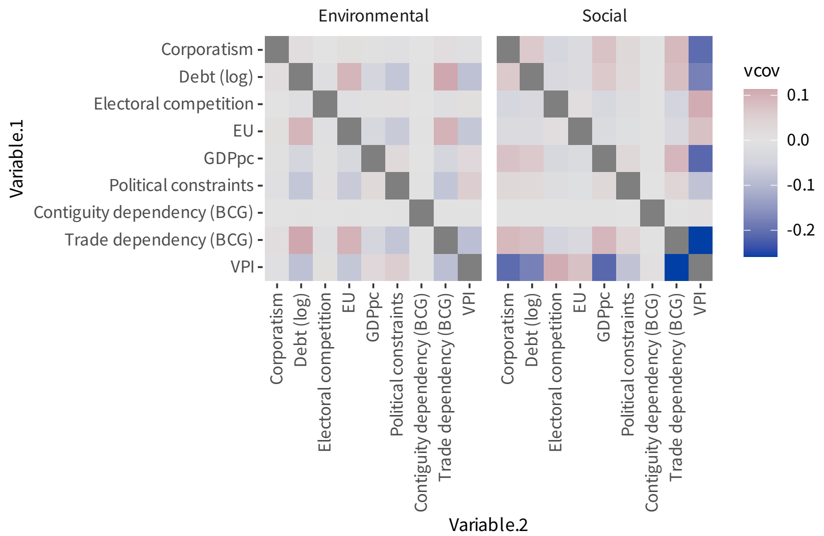


Figure 2.7: Variance-covariance matrix.

```
d.corr <- d %>%
  select(Country, Year,
```

```
        VPI, `Burden capacity gap` = `Implementation deficit`,
        Debt, `Electoral competition`, EU,
        GDPpc, `Political constraints`) %>%
  unique()

corrplot::corrplot(cor(select(d.corr, -c(Country, Year)), use = "complete.obs"),
  method = "number", diag = FALSE, type = "lower",
  tl.cex = 0.6)
```

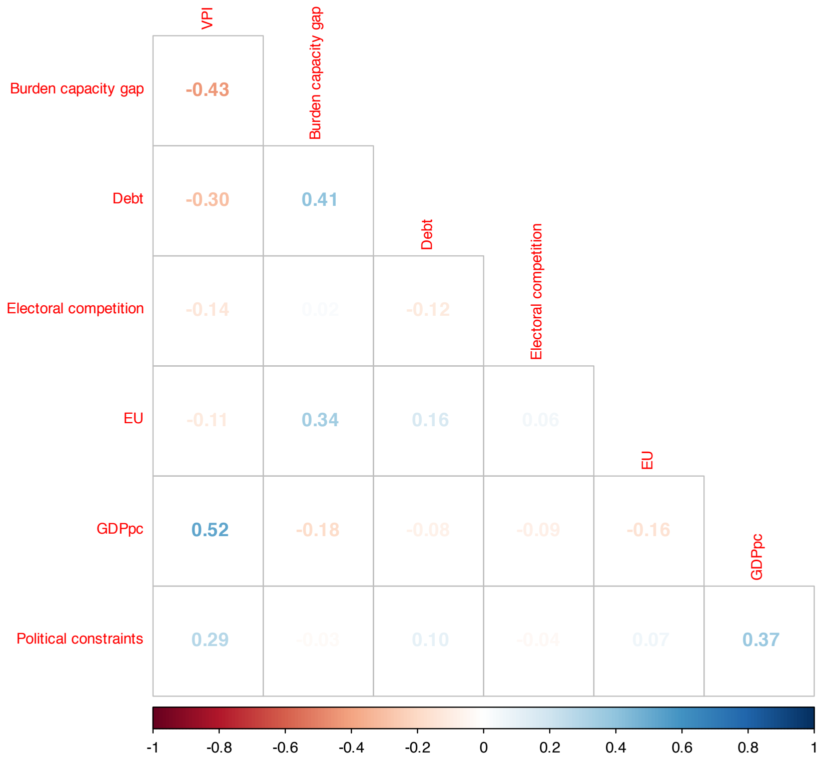


Figure 2.8: Correlation matrix.

Probabilities of interest.

```
### Probabilities of interest
tb <- S.theta %>%
  filter(Covariate != "(Intercept)") %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(`P(positive association)` = round(length(which(value > 0)) / n() * 100, 1)) %>%
  mutate(`P(negative association)` = 100 - `P(positive association)`) %>%
  # sort
  mutate(psum = `P(positive association)`^2 + `P(negative association)`^2) %>%
  # end sort
  gather(Probability, value, -Sector, -Covariate, -psum) %>%
  mutate(value = ifelse(value < 50, "", paste0(value, " %"))) %>%
  spread(Probability, value) %>%
  arrange(desc(psum)) %>%
  ungroup() %>%
  select(-psum)

tc <- "Probabilities of interest: probability of finding either a positive or a negative association, conditio
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = T)
}
```

Table 2.1: Probabilities of interest: probability of finding either a positive or a negative association, conditional on priors.

Sector	Covariate	P(negative association)	P(positive association)
Environmental	Debt (log)		100 %
Environmental	EU		100 %
Environmental	Political constraints	100 %	
Environmental	Trade dependency (BCG)		100 %
Environmental	VPI	100 %	
Social	Corporatism		100 %
Social	Debt (log)		100 %
Social	Electoral competition	100 %	
Social	GDPpc		100 %
Social	Trade dependency (BCG)		100 %
Social	VPI	100 %	
Environmental	GDPpc	99.6 %	
Social	Political constraints		98.8 %
Social	EU	98.3 %	
Environmental	Electoral competition	91.3 %	
Environmental	Corporatism		90.6 %
Social	Contiguity dependency (BCG)	59.2 %	
Environmental	Contiguity dependency (BCG)		56.5 %

2.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                              Country = country.label,
                              Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)
```

```
S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

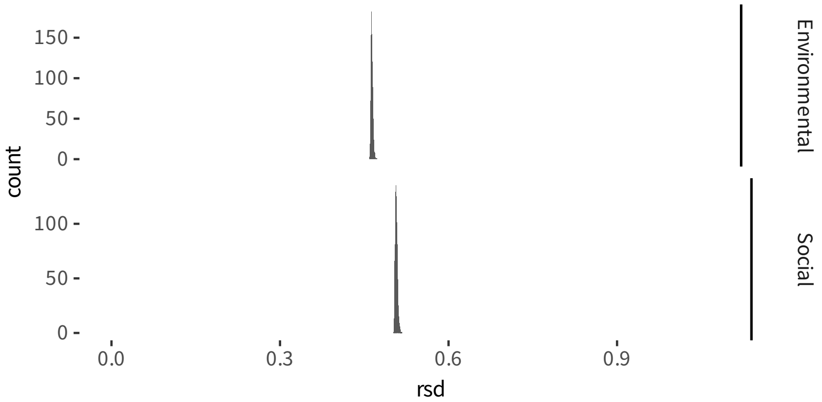


Figure 2.9: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

Sector	Pseudo.R2
Environmental	0.5864
Social	0.5549

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Reference model. Coefficient point estimates (median of the posterior distribution),

tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-312}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s

print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.56	(0.117)	[0.33 : 0.79]
Debt (log)	0.53	(0.062)	[0.42 : 0.66]
EU	0.47	(0.068)	[0.34 : 0.61]
VPI	-0.41	(0.065)	[-0.54 : -0.29]
Political constraints	-0.38	(0.083)	[-0.54 : -0.22]
GDPpc	-0.22	(0.08)	[-0.37 : -0.064]
Corporatism	0.07	(0.057)	[-0.037 : 0.18]
Electoral competition	-0.07	(0.054)	[-0.18 : 0.033]
Contiguity dependency (BCG)	0.01	(0.087)	[-0.15 : 0.19]
** Goodness of fit (R2)	0.59	(0.00146)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.21	(0.081)	[-1.4 : -1]
Trade dependency (BCG)	0.54	(0.093)	[0.36 : 0.72]
GDPpc	0.45	(0.09)	[0.28 : 0.63]
Corporatism	0.45	(0.064)	[0.32 : 0.57]
Debt (log)	0.38	(0.067)	[0.25 : 0.51]
Electoral competition	-0.24	(0.051)	[-0.34 : -0.14]
Political constraints	0.19	(0.087)	[0.024 : 0.36]
EU	-0.18	(0.082)	[-0.33 : -0.012]
Contiguity dependency (BCG)	-0.02	(0.083)	[-0.18 : 0.15]
** Goodness of fit (R2)	0.56	(0.00175)	[0.55 : 0.56]

Table 2.2: Model parameters. Reference model. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a6.tex")
```

2.2 Magnitude of effects

Interpretation in terms of expected values.

```
L.mu <- plab("mu", list(Sector = sector.label,
                        Country = country.label,
                        Year = year.label.numeric)) %>%
  filter(str_detect(Country, "^Z-"))
ci.mu <- ggs(s, family = "^mu\\[", par_labels = L.mu, sort = FALSE) %>%
  mutate(Year = as.integer(as.character(Year))) %>%
  ci()

f1 <-

  ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 1:19))) %>%
  filter(Year ==  min(Year)) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  ggplot(aes(x = `Top-down (VPI)`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
#  expand_limits(y = c(0, max(D$Y, na.rm = TRUE))) +
  expand_limits(y = c(0.97, 1.03)) +
  facet_grid(Sector ~ .) +
  xlab(variable.sector.label[1]) +
  ggtitle("(a)") +
  ylab("Expected\nburden capacity gap")


f2 <- ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 19 + (1:19)))) %>%
  filter(Year ==  min(Year)) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  ggplot(aes(x = `Bottom-up (VPI)`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
#  expand_limits(y = c(0, max(D$Y, na.rm = TRUE))) +
  expand_limits(y = c(0.97, 1.03)) +
  facet_grid(Sector ~ .) +
```

```
  xlab(variable.sector.label[2]) +
  ggtitle(”(b)”) +
  ylab(”Expected\nburden capacity gap”)

f3 <- ci.mu %>%
  filter(Country %in% paste0(”Z-”, sprintf(”%02d”, 19 + (20:29)))) %>%
  filter(Year ==  min(Year)) %>%
  left_join(pc) %>%
  ggplot(aes(x = `Political constraints`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
#  expand_limits(y = c(0, max(D$Y, na.rm = TRUE))) +
  expand_limits(y = c(0.97, 1.03)) +
  facet_grid(Sector ~ .) +
  ggtitle(”(c)”) +
  ylab(”Expected\nburden capacity gap”)

cowplot::plot_grid(f1, f2, f3, ncol = 3)
```

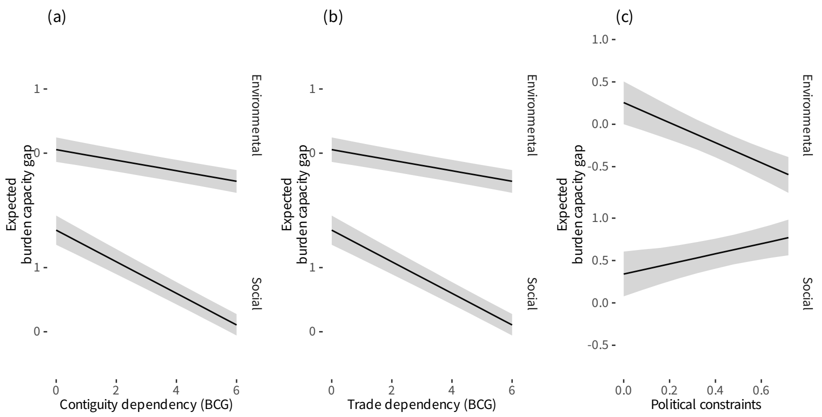


Figure 2.10: Magnitude of the effects: expected change in the burden capacity gap when (a) vertical policy integration or (b) political constraints move from the minimum to the maximum observed. In all cases the rest of the variables are fixed at their means, and year is 1976.

```
#
# Figure 6
#
ci.mu %>%
  filter(Country %in% paste0(”Z-”, sprintf(”%02d”, 1:19))) %>%
  filter(Year ==  min(Year)) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  ggplot(aes(x = `VPI`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_grid(~ Sector) +
  ylab(”Expected\nburden capacity gap”)
```

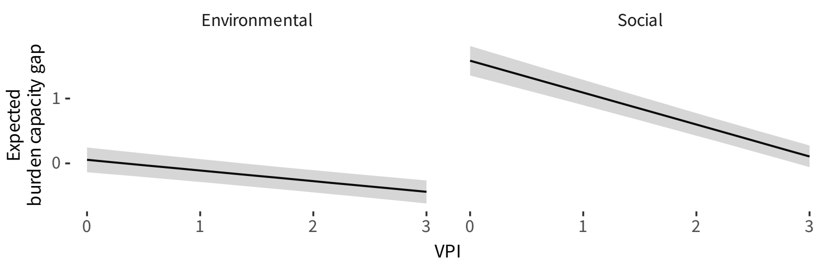


Figure 2.11: Magnitude of the effects: expected change in the burden capacity gap when vertical policy integration moves from the minimum to the maximum observed. In all cases the rest of the variables are fixed at their means. Figure 6 in the article.

Expected burden capacity gap difference

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 1:19))) %>%
  filter(Year ==  min(Year)) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  group_by(Sector) %>%
  filter((VPI == min(VPI) | VPI == max(VPI))) %>%
  select(Sector, VPI, median) %>%
  ungroup() %>%
  pivot_wider(names_from = VPI, values_from = median) %>%
  mutate(Difference = `0` - `3`) %>%
  kable()
```

2.3 *Descriptive figures of variables*

```
#
# Figure A2
#
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = Year, y = PS, color = Sector)) +
  geom_line() +
  facet_wrap(~ Country) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_manual(values = pal.sector) +
  ylab("Implementation burden") +
  ggtitle("Implementation burden")
```

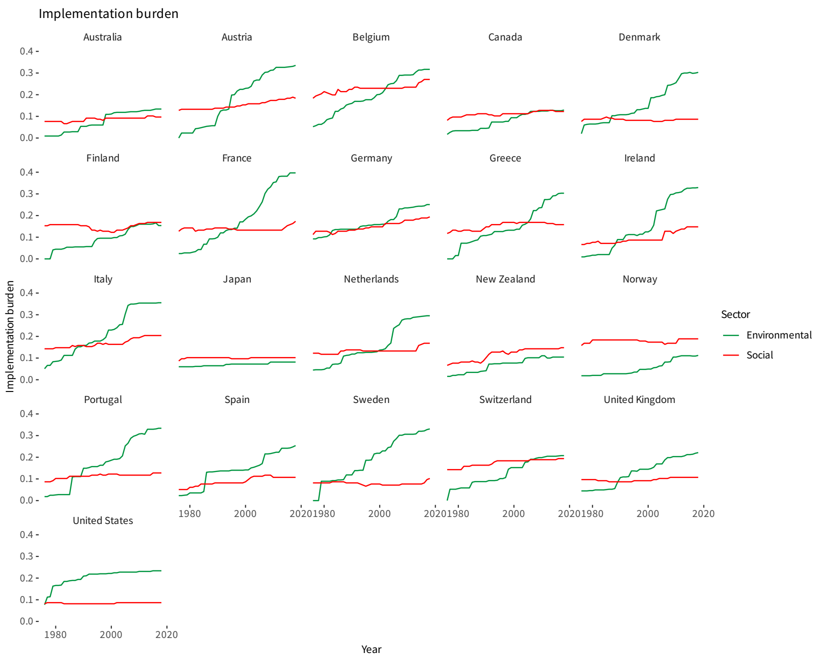


Figure 2.12: Sector-specific evolution of portfolio size.

```
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  filter(Sector == "Environmental") %>%
  ggplot(aes(x = Year, y = PS)) +
  geom_line() +
  facet_wrap(~ Country) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_discrete_qualitative(rev = TRUE) +
  ggtitle("Implementation burden")
```

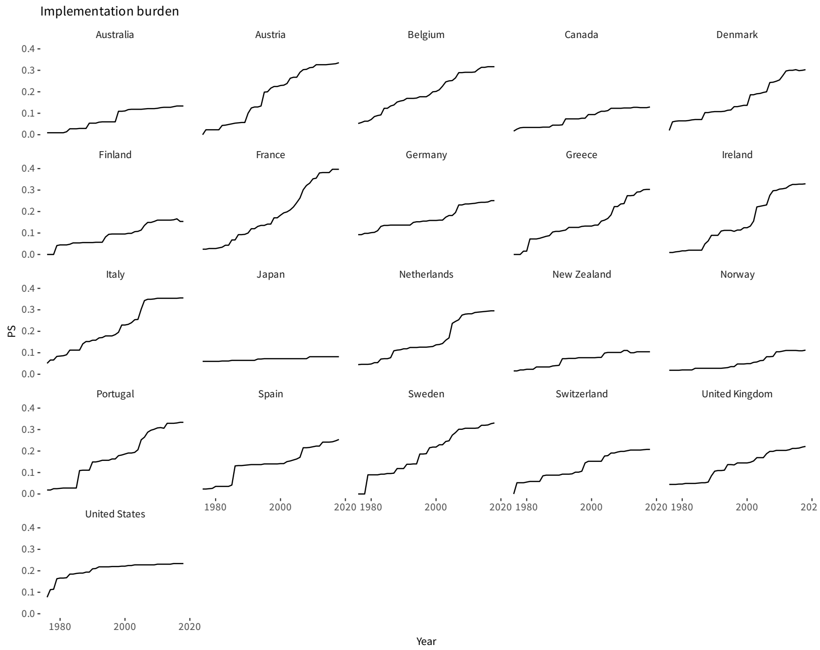


Figure 2.13: Evolution of environmental portfolio size.

```
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  filter(Sector == "Environmental") %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(gr = (PS - lag(PS)) / PS) %>%
  select(Country, Year, PS, gr) %>%
  summarize(`Average growth rate` = mean(gr, na.rm = TRUE)) %>%
  ggplot(aes(x = `Average growth rate`, y = reorder(Country, `Average growth rate`))) +
    geom_point() +
    ylab("Country") +
    ggtitle("Average growth rate")
```

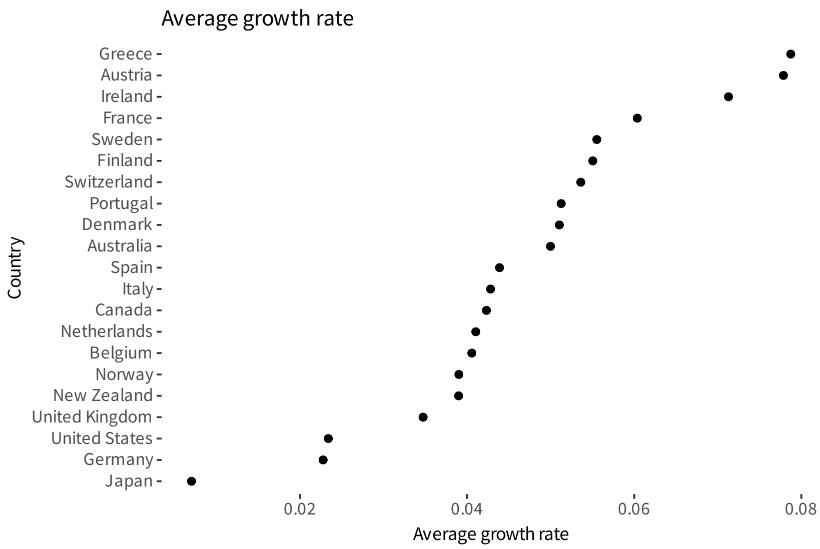


Figure 2.14: Average environmental growth rate.

```
#
# Figure A3
#
d %>%
```

```
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = Year, y = IC, color = Sector)) +
  ylab("Implementation capacity") +
  #
  geom_line() +
  facet_wrap(~ Country) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_manual(values = pal.sector) +
  ggtitle("Implementation capacity")
```

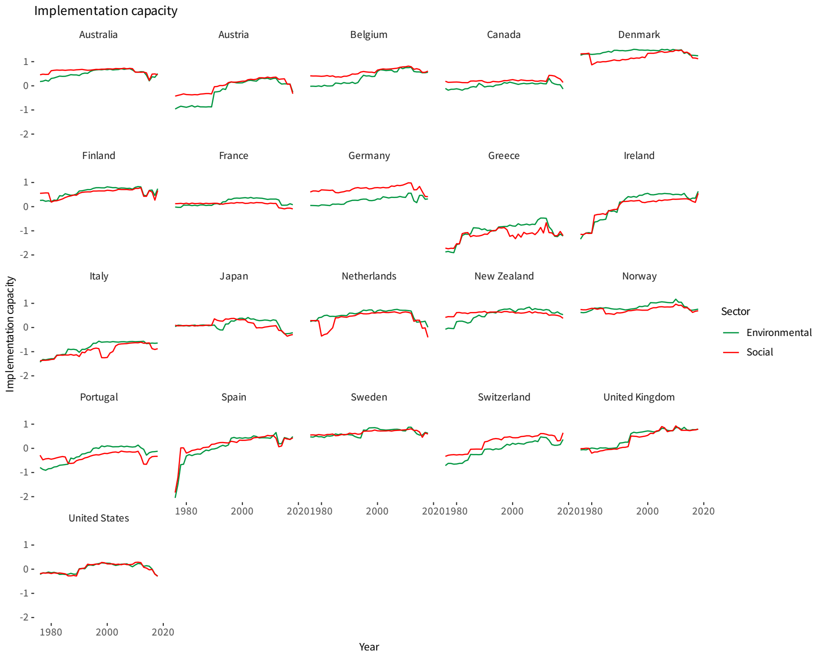


Figure 2.15: Sector-specific temporal evolution of implementation capacity.

```
#ggplot(sc, aes(x = Year, y = Score, color = Sector)) +
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  filter(Sector == "Environmental") %>%
  ggplot(aes(x = Year, y = IC)) +
  ylab("Implementation capacity") +
  #
  geom_line() +
  facet_wrap(~ Country) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_discrete_qualitative(rev = TRUE) +
  ggtitle("Implementation capacity")


#
# Figure A4
#
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = Year, y = `Implementation deficit`, color = Sector)) +
  geom_line() +
  facet_wrap(~ Country) +
  ylab("Burden capacity gap") +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_manual(values = pal.sector) +
  ggtitle("Burden capacity gap: Implementation burden / Implementation capacity")
```

Burden capacity gap: mean and standard deviation by sector.

```
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  group_by(Sector) %>%
```

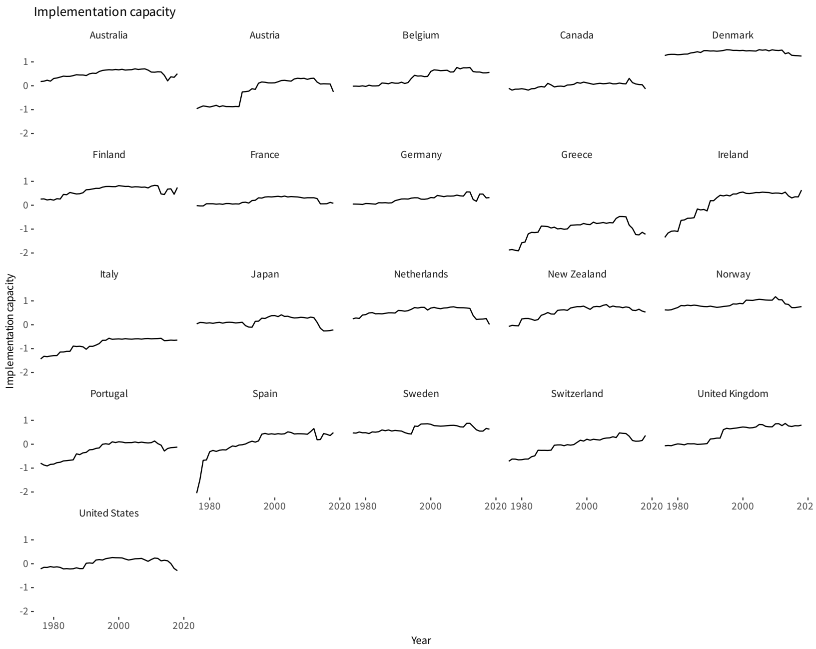


Figure 2.16: Sector-specific temporal evolution of implementation capacity.

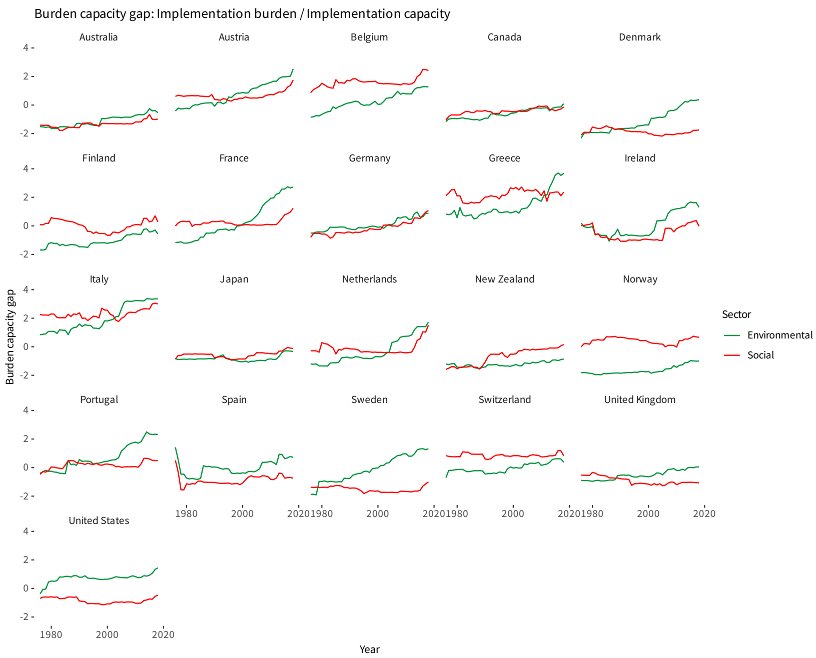


Figure 2.17: Sector-specific evolution of the burden capacity gap: portfolio size/implementation capacity.

```
  summarize(Mean = mean(`Implementation deficit`),
            SD = sd(`Implementation deficit`)) %>%
  kable()
```

Burden capacity gap: minimum and maximum by sector.

```
d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  group_by(Sector, Country) %>%
  summarize(`Average implementation deficit` = mean(`Implementation deficit`)) %>%
  group_by(Sector) %>%
  filter(`Average implementation deficit` %in% c(min(`Average implementation deficit`),
                                                 max(`Average implementation deficit`))) %>%
  select(Country, Sector, `Average implementation deficit`) %>%
  kable()

ps.ic.vpi <- d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  select(Country, Sector, Year,
         `Portfolio size` = PS,
         `Implementation capacity` = IC) %>%
  left_join(
    vpi.original %>%
      filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
      spread(Dimension, VPI) %>%
      mutate(VPI = (`Top-down (VPI)` + `Bottom-up (VPI)`) / 2) %>%
      gather(Dimension, VPI, -c(Sector, Country, Year)) %>%
      mutate(Dimension = fct_relevel(Dimension, c("VPI",
                                                  "Top-down (VPI)",
                                                  "Bottom-up (VPI)"))) %>%
      filter(Dimension == "VPI") %>%
      select(-Dimension)) %>%
  pivot_longer(-c(Country, Sector, Year),
               names_to = "Variable", values_to = "value")

ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
              names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point(aes(size = VPI)) +
    geom_text_repel() +
    facet_grid(Feature ~ Sector, scales = "free_x")


ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
              names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                        VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                        VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                        VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  mutate(VPI2 = ifelse(VPI == "VPI: Low [0,2)", "low", "high")) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
```

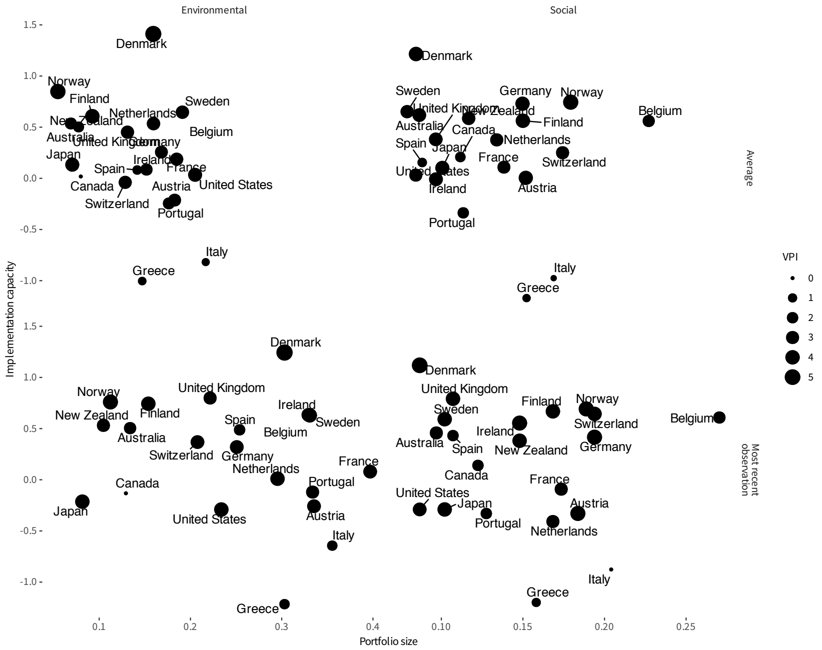


Figure 2.18: Implementation capacity against portfolio size, with VPI as dot size. Columns represent sectors, and rows represent either the average of the sample or the last period observed. Scale is adapted to each sector.

```
             label = Country)) +
          geom_point(aes(color = VPI, shape = VPI)) +
    geom_text_repel() +
    scale_shape_manual(#name = ”VPI”,
                       values = c(1, 19, 19)) +#,
    scale_color_manual(#name = ”VPI”, #guide = ”none”,
                      values = c(NA, ”grey50”, ”black”)) +
    facet_grid(Feature ~ Sector, scales = ”free_x”)


ps.ic.vpi.av <- ps.ic.vpi %>%
  filter(Variable != ”VPI”) %>%
  group_by(Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from = Variable, values_from = Average)


ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
              names_to = ”Feature”, values_to = ”v”) %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                       VPI >= 0 & VPI < 2 ~ ”VPI: Low [0,2)”,
                       VPI >= 2 & VPI < 4 ~ ”VPI: Medium [2,4)”,
                       VPI >= 4 ~ ”VPI: High [4,6]”)) %>%
  mutate(VPI = fct_relevel(VPI, ”VPI: Low [0,2)”, ”VPI: Medium [2,4)”, ”VPI: High [4,6]”)) %>%
  mutate(VPI2 = ifelse(VPI == ”VPI: Low [0,2)”, ”low”, ”high”)) %>%
  filter(Feature == ”Average”) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
          geom_point(aes(color = VPI, shape = VPI)) +
    geom_text_repel(family = ”Source Sans Pro”) +
    scale_shape_manual(values = c(1, 19, 19)) +
    scale_color_manual(values = c(NA, ”grey50”, ”black”)) +
    geom_vline(data = ps.ic.vpi.av, aes(xintercept = `Portfolio size`), lty = 3) +
```

Figure 2.19: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent VPI, and rows represent the combination of sector and either the average of the sample or the last period observed.

```
geom_hline(data = ps.ic.vpi.av, aes(yintercept = `Implementation capacity`), lty = 3) +
facet_grid(. ~ Sector, scales = "free_x") +
theme_bw()
```

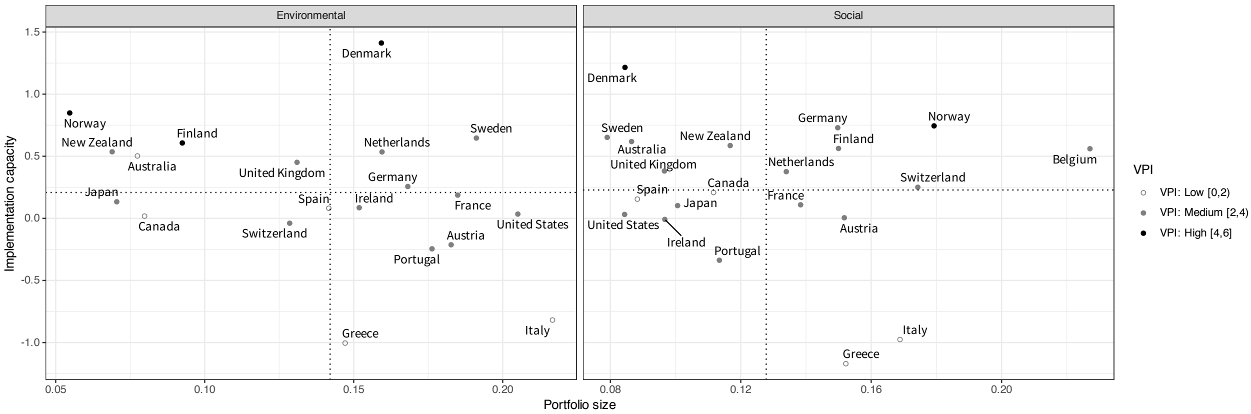


Figure 2.20: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent sector.

```
ps.ic.vpi.av <- ps.ic.vpi %>%
  filter(Variable != "VPI") %>%
  group_by(Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from = Variable, values_from = Average)


ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
              filter(Year == max(Year)) %>%
              rename(`Most recent\observation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
```

```
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(VPI >= 0 & VPI < 1 ~ ”VPI: Low [0,1)”,
                         VPI >= 1 & VPI < 2 ~ ”VPI: Low-Medium [1,2)”,
                         VPI >= 2 & VPI < 3 ~ ”VPI: Medium [2,3)”,
                         VPI >= 3 & VPI < 4 ~ ”VPI: Medium-High [3,4)”,
                         VPI >= 4 ~ ”VPI: High [4,)”)) %>%
  mutate(VPI = fct_relevel(VPI, ”VPI: Low [0,1)”,  ”VPI: Low-Medium [1,2)”,
                          ”VPI: Medium [2,3)”, ”VPI: Medium-High [3,4)”,
                          ”VPI: High [4,)”)) %>%
  mutate(VPI2 = ifelse(VPI %in% c(”VPI: Low [0,1)”, ”VPI: Low-Medium [1,2)”),  ”low”, ”high”)) %>%
  filter(Feature == ”Average”) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
          geom_point(aes(color = VPI, shape = VPI)) +
    xlab(”Implementation burden”) +
    geom_text_repel(family = ”Source Sans Pro”) +
    scale_shape_manual(values = c(1, 1, 19, 19, 19)) +
    scale_color_manual(values = c(”grey80”, ”black”, ”grey80”, ”grey40”, ”black”)) +
    geom_vline(data = ps.ic.vpi.av, aes(xintercept = `Portfolio size`), lty = 3) +
    geom_hline(data = ps.ic.vpi.av, aes(yintercept = `Implementation capacity`), lty = 3) +
    facet_grid(. ~ Sector, scales = ”free_x”) +
    theme_bw()
```

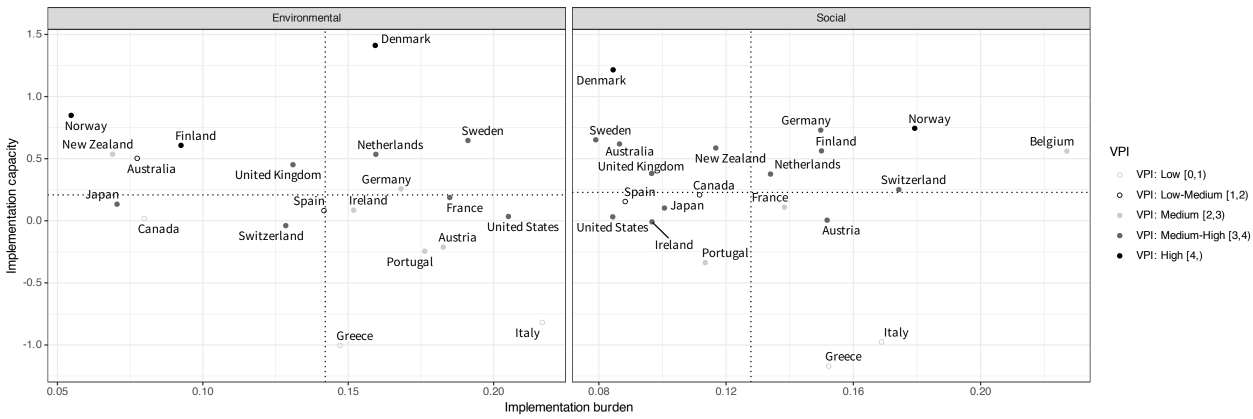


Figure 2.21: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent sector.

```
ps.ic.vpi.av <- ps.ic.vpi %>%
  filter(Variable != ”VPI”) %>%
  group_by(Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from = Variable, values_from = Average)
```

```
ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
              filter(Year == max(Year)) %>%
              rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = ”Feature”, values_to = ”v”) %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                         VPI >= 0 & VPI < 2 ~ ”VPI: Low [0,2)”,
                         VPI >= 2 & VPI < 4 ~ ”VPI: Medium [2,4)”,
                         VPI >= 4 ~ ”VPI: High [4,6]”)) %>%
  mutate(VPI = fct_relevel(VPI, ”VPI: Low [0,2)”, ”VPI: Medium [2,4)”, ”VPI: High [4,6]”)) %>%
  mutate(VPI2 = ifelse(VPI == ”VPI: Low [0,2)”, ”low”, ”high”)) %>%
  filter(Feature == ”Average”) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
```

```
             label = Country)) +
          geom_point(aes(color = VPI, shape = VPI)) +
    geom_text_repel(family = "Source Sans Pro") +
    scale_shape_manual(values = c(1, 19, 19)) +
    scale_color_manual(values = c(NA, "grey50", "black")) +
    geom_vline(data = ps.ic.vpi.av, aes(xintercept = `Portfolio size`), lty = 3) +
    geom_hline(data = ps.ic.vpi.av, aes(yintercept = `Implementation capacity`), lty = 3) +
    facet_grid(. ~ Sector, scales = "free_x")
```

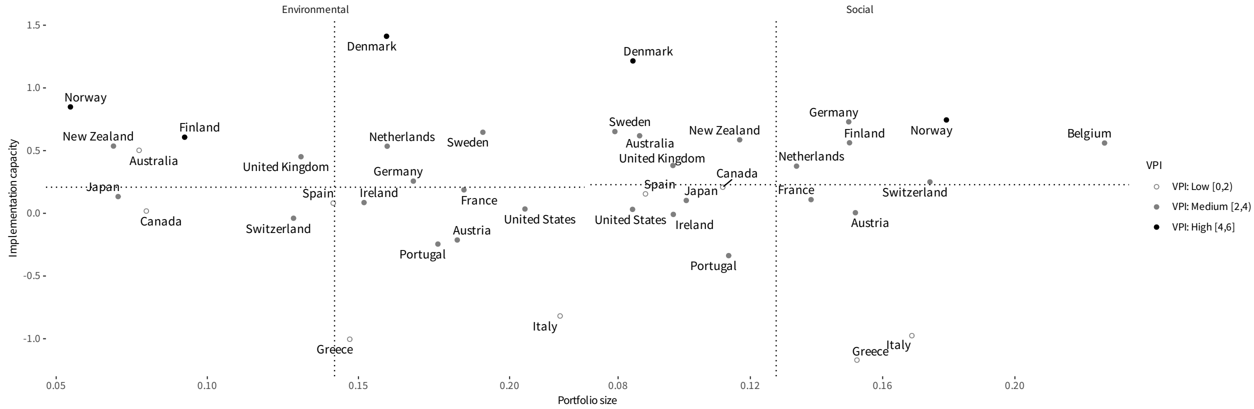


Figure 2.22: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent sector.

```
ps.ic.vpi.av <- ps.ic.vpi %>%
  filter(Variable != "VPI") %>%
  group_by(Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from = Variable, values_from = Average)


ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
              filter(Year == max(Year)) %>%
              rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                          VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                          VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                          VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  mutate(VPI2 = ifelse(VPI == "VPI: Low [0,2)", "low", "high")) %>%
  filter(Feature == "Average") %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
          geom_point(aes(color = VPI, shape = VPI), size = 2.0) +
    geom_text_repel(family = "Source Sans Pro") +
    scale_shape_manual(values = c(1, 19, 19)) +
    scale_color_manual(values = c(NA, "grey50", "black")) +
    geom_vline(data = ps.ic.vpi.av, aes(xintercept = `Portfolio size`), lty = 3) +
    geom_hline(data = ps.ic.vpi.av, aes(yintercept = `Implementation capacity`), lty = 3) +
    facet_grid(. ~ Sector, scales = "free_x") +
    theme_bw()


ps.ic.vpi.av <- ps.ic.vpi %>%
  filter(Variable != "VPI") %>%
  group_by(Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  pivot_wider(names_from = Variable, values_from = Average)
```

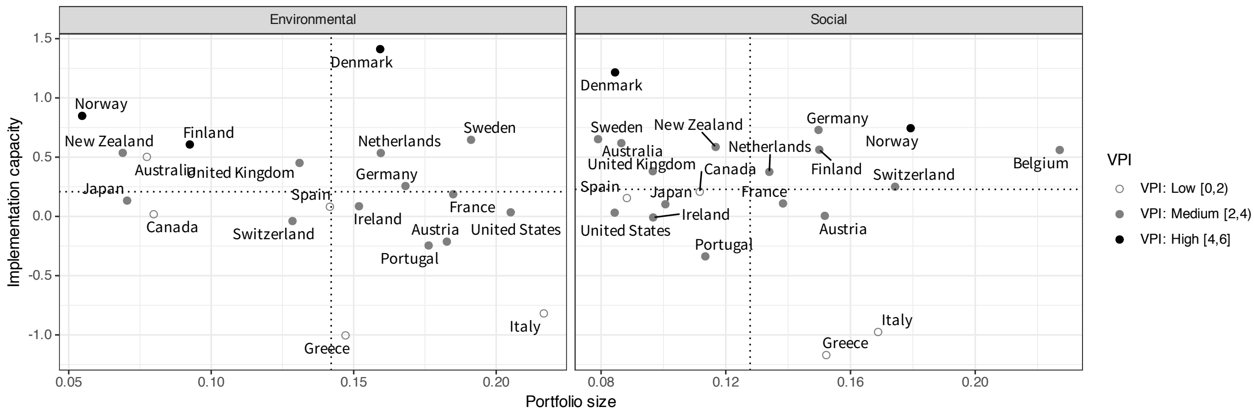


Figure 2.23: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent sector.

```
ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
              filter(Year == max(Year)) %>%
              rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                        VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                        VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                        VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  mutate(VPI2 = ifelse(VPI == "VPI: Low [0,2)", "low", "high")) %>%
  filter(Feature == "Average") %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
          geom_point(aes(color = VPI, shape = VPI)) +
    geom_text_repel(family = "Source Sans Pro") +
    scale_shape_manual(values = c(1, 19, 19)) +
    scale_color_manual(values = c(NA, "grey50", "black")) +
    geom_vline(data = ps.ic.vpi.av, aes(xintercept = `Portfolio size`), lty = 3) +
    geom_hline(data = ps.ic.vpi.av, aes(yintercept = `Implementation capacity`), lty = 3) +
    facet_grid(. ~ Sector, scales = "free_x")
```

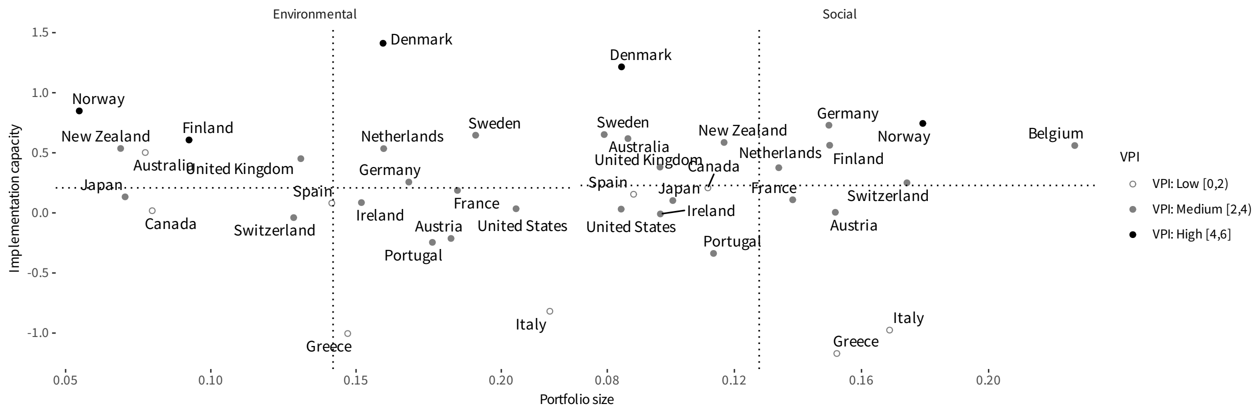


Figure 2.24: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent sector.

```
ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = ”Feature”, values_to = ”v”) %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                        VPI >= 0 & VPI < 2 ~ ”VPI: Low [0,2)”,
                        VPI >= 2 & VPI < 4 ~ ”VPI: Medium [2,4)”,
                        VPI >= 4 ~ ”VPI: High [4,6]”)) %>%
  mutate(VPI = fct_relevel(VPI, ”VPI: Low [0,2)”, ”VPI: Medium [2,4)”, ”VPI: High [4,6]”)) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point() +
    geom_text_repel() +
    facet_grid(Sector + Feature ~ VPI, scales = ”free_x”)
```

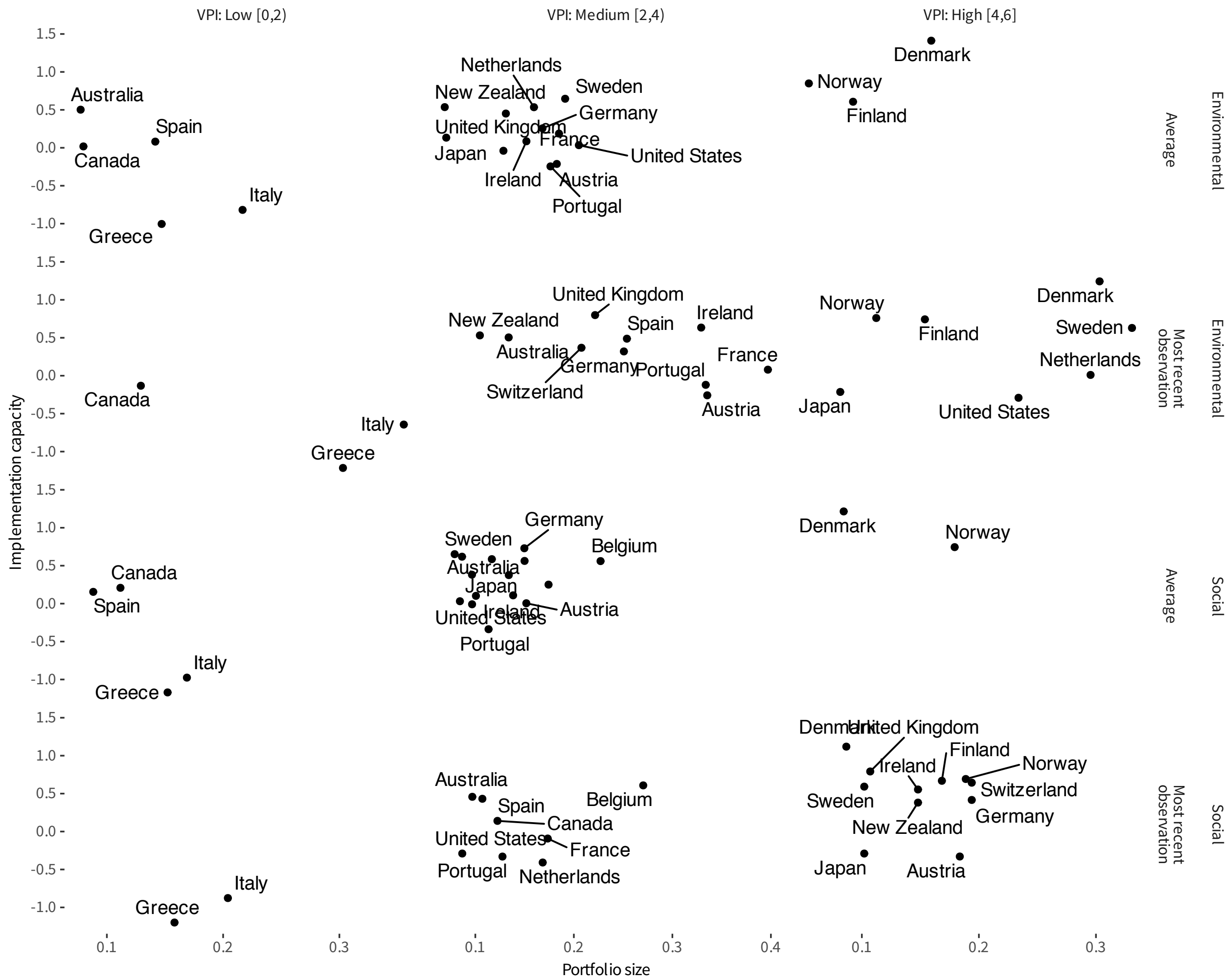


Figure 2.25: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent VPI, and rows represent the combination of sector and either the average of the sample or the last period observed.

```
f1 <- ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
```

```
  left_join(ps.ic.vpi %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                       VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
         VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
         VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  filter(Feature == "Average") %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point() +
    geom_text_repel() +
    facet_grid(Sector ~ VPI, scales = "free_x") +
    ggtitle("Average over time")

f2 <- ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                       VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                       VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                       VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  filter(Feature != "Average") %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point() +
    geom_text_repel() +
    facet_grid(Sector ~ VPI, scales = "free_x") +
    ggtitle("Last observed period")

plot_grid(f1, f2, nrow = 2)


f1 <- ps.ic.vpi %>%
  filter(Sector == "Environmental") %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
            filter(Sector == "Environmental") %>%
            filter(Year == max(Year)) %>%
            rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                       VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                       VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                       VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point() +
    geom_text_repel() +
```

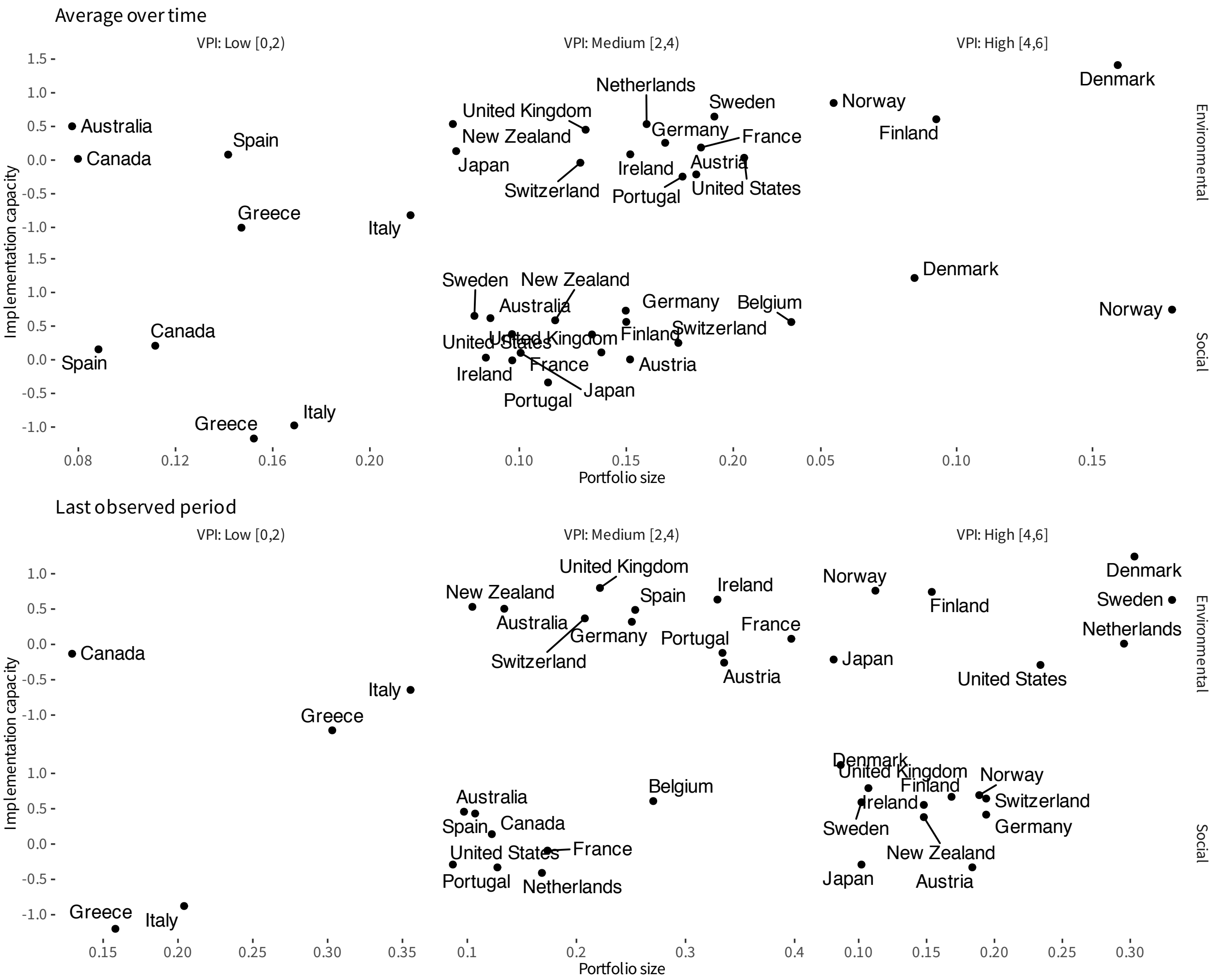


Figure 2.26: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent VPI, and rows represent the sector. For either the average of the series, or the last period observed.

```
    facet_grid(Feature ~ VPI, scales = "free_x") +
    ggtitle("Environmental")

f2 <- ps.ic.vpi %>%
  filter(Sector == "Social") %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
              filter(Sector == "Social") %>%
              filter(Year == max(Year)) %>%
              rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                           VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                           VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                           VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point() +
    geom_text_repel() +
    facet_grid(Feature ~ VPI, scales = "free_x") +
    ggtitle("Social")

plot_grid(f1, f2, nrow = 2)


ps.ic.vpi %>%
  group_by(Country, Sector, Variable) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  ungroup() %>%
  left_join(ps.ic.vpi %>%
              filter(Year == max(Year)) %>%
              rename(`Most recent\nobservation` = value)) %>%
  select(-Year) %>%
  pivot_longer(-c(Country, Sector, Variable),
               names_to = "Feature", values_to = "v") %>%
  pivot_wider(names_from = Variable, values_from = v) %>%
  filter(!is.na(VPI)) %>%
  mutate(VPI = case_when(
                           VPI >= 0 & VPI < 2 ~ "VPI: Low [0,2)",
                           VPI >= 2 & VPI < 4 ~ "VPI: Medium [2,4)",
                           VPI >= 4 ~ "VPI: High [4,6]")) %>%
  mutate(VPI = fct_relevel(VPI, "VPI: Low [0,2)", "VPI: Medium [2,4)", "VPI: High [4,6]")) %>%
  ggplot(aes(x = `Portfolio size`,
             y = `Implementation capacity`,
             label = Country)) +
    geom_point() +
    geom_text_repel() +
    facet_grid(Sector + Feature ~ VPI, scales = "free_x") +
    theme_bw()


d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = Year, y = `Implementation deficit`, color = Sector)) +
  geom_line() +
  facet_wrap(~ Country, scales = "free") +
  ylab("Burden capacity gap") +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_discrete_qualitative(rev = TRUE) +
  #ggtitle("Burden capacity gap: Portfolio size / Implementation capacity")
  ggtitle("Burden capacity gap: Implementation burden / Implementation capacity")


sort.deficit <- d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  group_by(Country) %>%
  summarize(Average.deficit = mean(`Implementation deficit`, na.rm = TRUE)) %>%
```

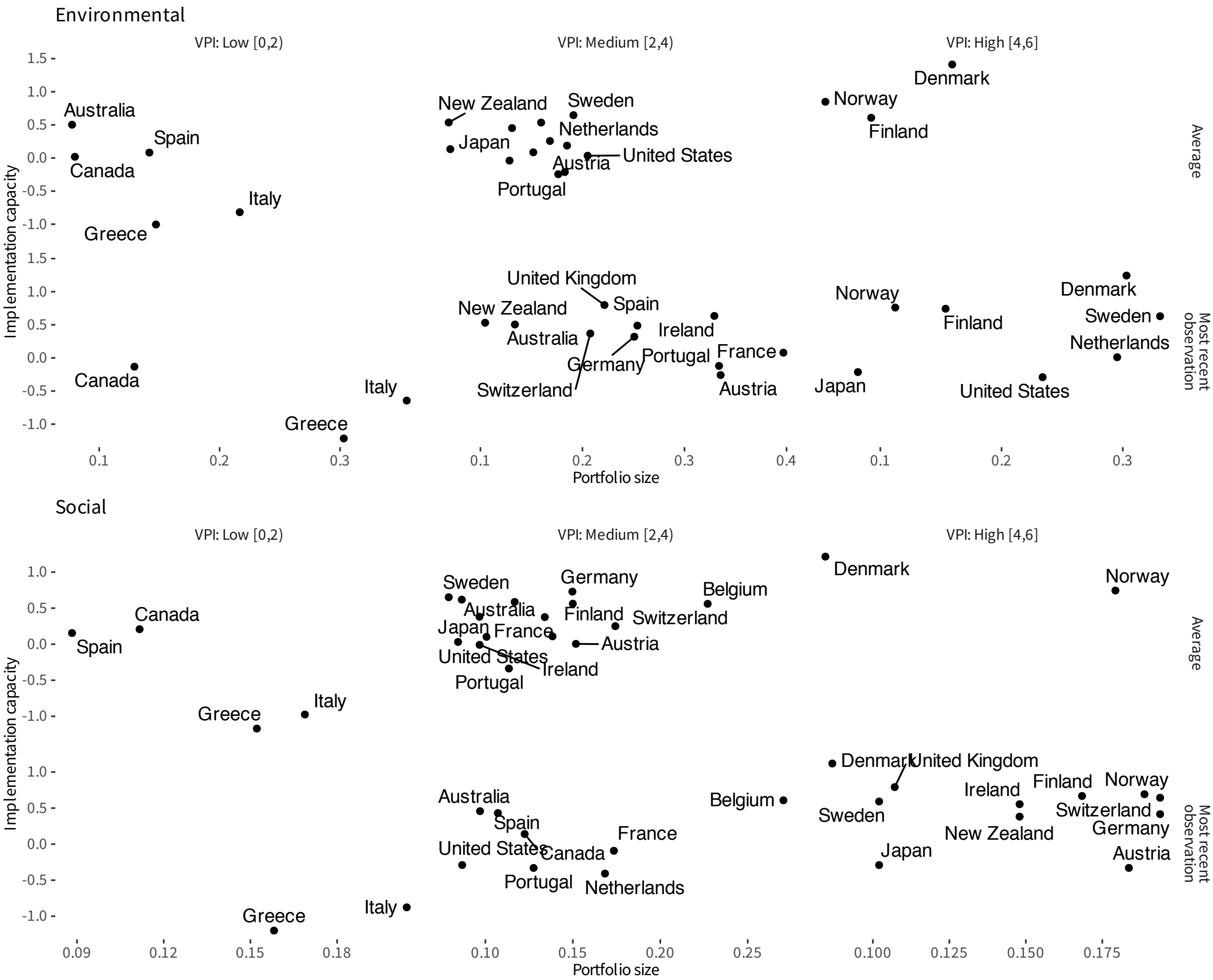


Figure 2.27: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent VPI, and rows represent the average of the sample or the last period observed. Environmental and Social sectors.

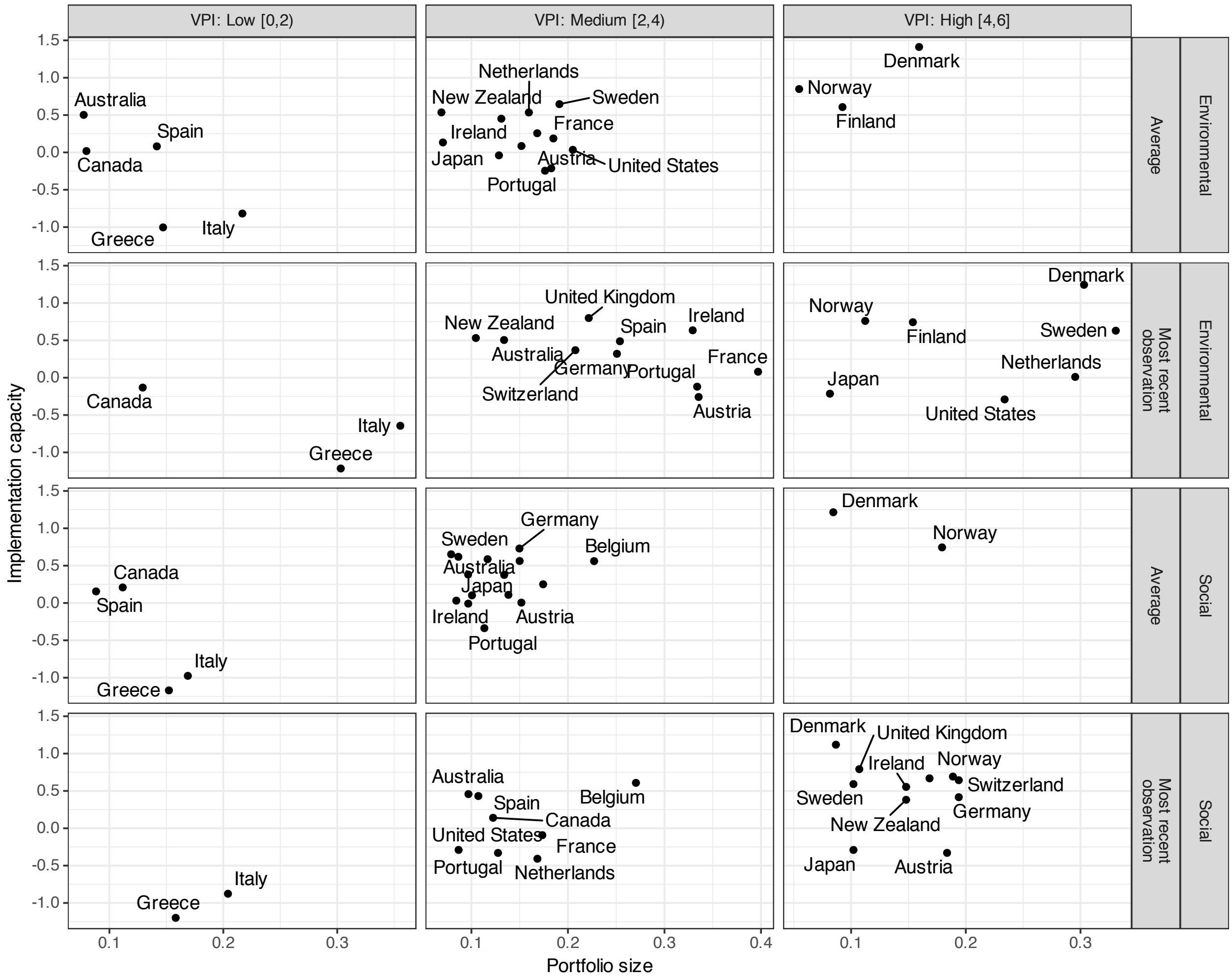


Figure 2.28: Implementation capacity against portfolio size, at different degrees of VPI. Columns represent VPI, and rows represent the combination of sector and either the average of the sample or the last period observed.

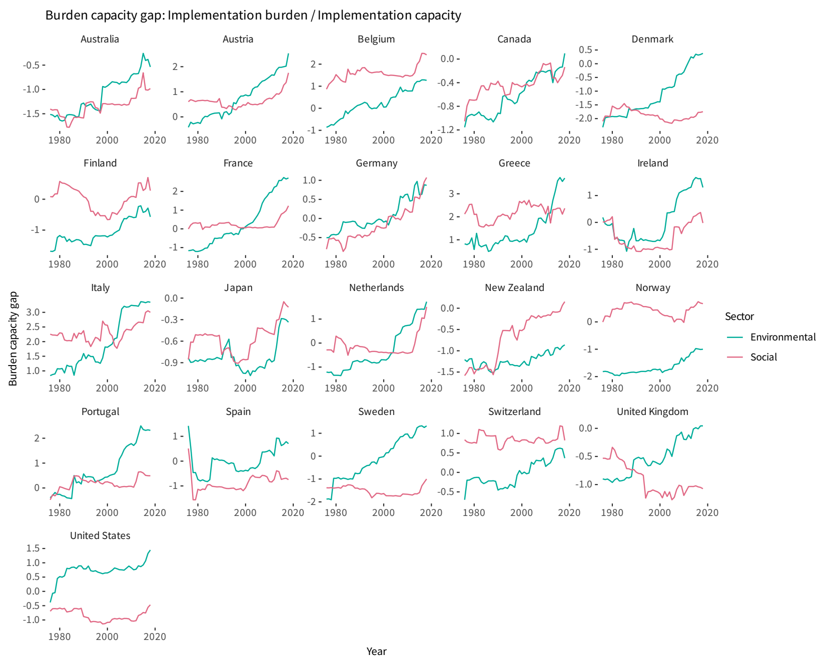


Figure 2.29: Sector-specific evolution of the burden capacity gap: portfolio size/implementation capacity. Each country has its own vertical scale.

```
  arrange(Average.deficit) %>%
  select(Country) %>%
  unlist(., use.names = FALSE)

d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  mutate(Country = fct_relevel(Country, sort.deficit)) %>%
  ggplot(aes(x = `Implementation deficit`, y = Country)) +
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  #geom_point() +
  facet_grid(~ Sector) +
  #xlab("Burden capacity gap:\nPortfolio size / Implementation capacity") +
  xlab("Burden capacity gap: Implementation burden / Implementation capacity") +
  #ggtitle("Portfolio size / Implementation capacity")
  ggtitle("Implementation burden / Implementation capacity")
```

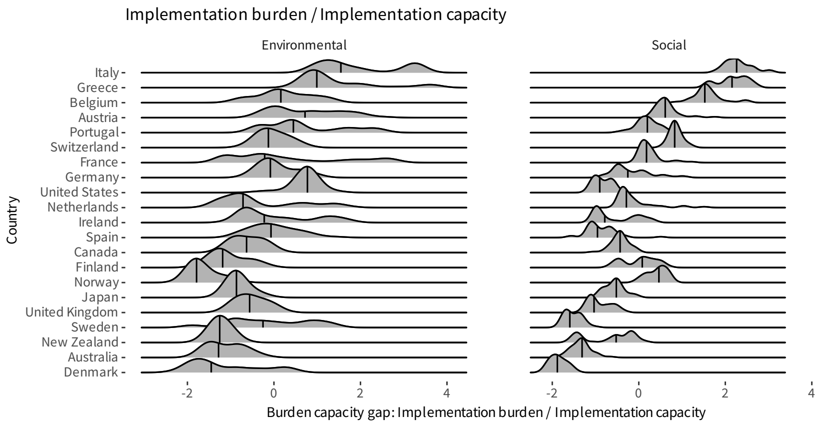


Figure 2.30: Distribution of burden capacity gap (portfolio size / implementation capacity) in the countries considered, by sector. Countries are sorted by average burden capacity gap across time and sectors. Vertical lines indicate median burden capacity gap in each sector.

```
f1 <- d %>%
  filter(Sector == "Environmental") %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = `Implementation deficit`, y = reorder(Country, `Implementation deficit`, median, na.rm = TRUE
  #geom_density_ridges(panel.scaling = FALSE, scale = 1) +
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  ylab("") +
  #xlab("Burden capacity gap:\nPortfolio size / Implementation capacity") +
  xlab("Burden capacity gap:\nImplementation burden / Implementation capacity") +
  ggtitle("Environmental")

f2 <- d %>%
  filter(Sector == "Social") %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = `Implementation deficit`, y = reorder(Country, `Implementation deficit`, median, na.rm = TRUE
#  geom_density_ridges(panel.scaling = FALSE, scale = 1) +
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  ylab("") +
  #xlab("Burden capacity gap:\nPortfolio size / Implementation capacity") +
  xlab("Burden capacity gap:\nImplementation burden / Implementation capacity") +
  ggtitle("Social")

plot_grid(f1, f2, ncol = 2)


#
# Figure 3
#
f1 <- d %>%
  filter(Sector == "Environmental") %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = `Implementation deficit`, y = reorder(Country, `Implementation deficit`, median, na.rm = TRUE
  #geom_density_ridges(panel.scaling = FALSE, scale = 1) +
```

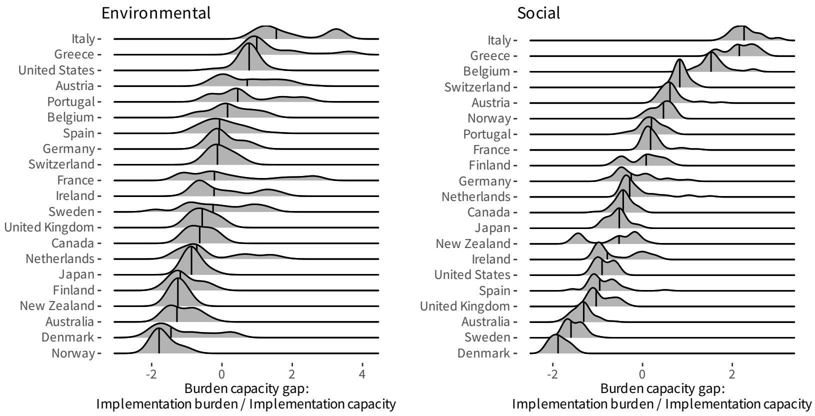


Figure 2.31: Distribution of burden capacity gap (portfolio size / implementation capacity) in the countries considered, by sector. Countries are sorted by median capacity gap across time.

```
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  ylab(””) +
  xlab(”Burden capacity gap:\nImplementation burden / Implementation capacity”) +
  theme(axis.title.x = element_text(size = 9)) +
  ggtitle(”Environmental”)

f2 <- d %>%
  filter(Sector == ”Social”) %>%
  filter(!Country %in% paste0(”Z-”, sprintf(”%02d”, 1:(19+29)))) %>%
  ggplot(aes(x = `Implementation deficit`, y = reorder(Country, `Implementation deficit`, median, na.rm = TRUE
#  geom_density_ridges(panel.scaling = FALSE, scale = 1) +
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  ylab(””) +
  xlab(”Burden capacity gap:\nImplementation burden / Implementation capacity”) +
  theme(axis.title.x = element_text(size = 9)) +
  ggtitle(”Social”)

plot_grid(f1, f2, ncol = 2)
```

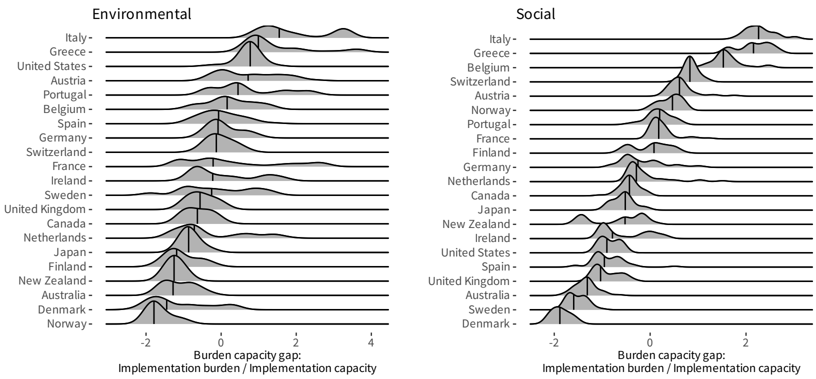


Figure 2.32: Distribution of burden capacity gap (portfolio size / implementation capacity) in the countries considered, by sector. Countries are sorted by median capacity gap across time. Figure 3 in the article.

```
f1 <- d %>%
  filter(Sector == ”Environmental”) %>%
  filter(!Country %in% paste0(”Z-”, sprintf(”%02d”, 1:(19+29)))) %>%
  mutate(Year = as.factor(Year)) %>%
  ggplot(aes(x = `Implementation deficit`, y = Year)) +
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  ylab(””) +
  xlab(”Burden capacity gap:\nImplementation burden / Implementation capacity”) +
  ggtitle(”Environmental”)

f2 <- d %>%
```

```
  filter(Sector == "Social") %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  mutate(Year = as.factor(Year)) %>%
  ggplot(aes(x = `Implementation deficit`, y = Year)) +
  geom_density_ridges() +
  stat_density_ridges(quantile_lines = TRUE, quantiles = 2) +
  ylab("") +
  xlab("Burden capacity gap:\nImplementation burden / Implementation capacity") +
  ggtitle("Social")

plot_grid(f1, f2, ncol = 2)
```

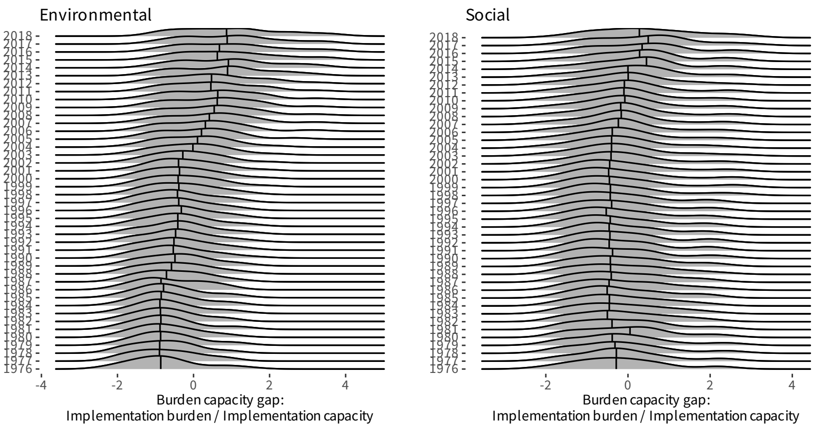


Figure 2.33: Distribution of burden capacity gap (portfolio size / implementation capacity) averaged by countries over time, by sector.

```
f1 <- d %>%
  filter(Sector == "Environmental") %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  group_by(Year) %>%
  summarize(average = mean(`Implementation deficit`, na.rm = TRUE)) %>%
  ggplot(aes(x = Year, y = average)) +
  geom_line() +
  ylab("Average burden capacity gap:\nPortfolio size / Implementation capacity") +
  ggtitle("Environmental")

f2 <- d %>%
  filter(Sector == "Social") %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  group_by(Year) %>%
  summarize(average = mean(`Implementation deficit`, na.rm = TRUE)) %>%
  ggplot(aes(x = Year, y = average)) +
  geom_line() +
  ylab("Average burden capacity gap:\nPortfolio size / Implementation capacity") +
  #xlab("Burden capacity gap:\nPortfolio size / Implementation capacity") +
  xlab("Burden capacity gap:\nImplementation burden / Implementation capacity") +
  ggtitle("Social")

plot_grid(f1, f2, ncol = 2)


tb <- d %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  group_by(Sector, Country) %>%
  summarize(`Average` = mean(`Implementation deficit`),
            `Median` = median(`Implementation deficit`))

tc <- "Mean and median for country burden capacity gaps, by sector."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = T)
}
```

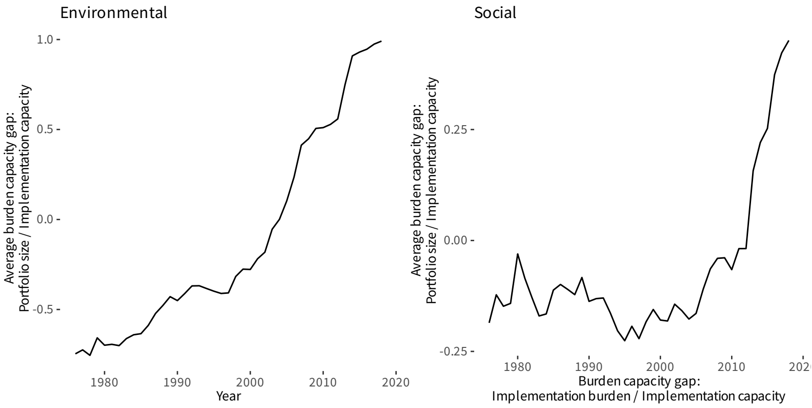


Figure 2.34: Distribution of burden capacity gap (portfolio size / implementation capacity) averaged by countries over time, by sector.

Table 2.3: Mean and median for country burden capacity gaps, by sector.

Sector	Country	Average	Median
Environmental	Australia	-1.1128	-1.2801
Environmental	Austria	0.7462	0.7187
Environmental	Belgium	0.2432	0.1596
Environmental	Canada	-0.6065	-0.6324
Environmental	Denmark	-1.1290	-1.4495
Environmental	Finland	-1.0586	-1.1853
Environmental	France	0.3245	-0.2138
Environmental	Germany	0.0737	-0.0825
Environmental	Greece	1.4187	0.9911
Environmental	Ireland	0.0767	-0.2239
Environmental	Italy	1.9658	1.5474
Environmental	Japan	-0.8190	-0.8660
Environmental	Netherlands	-0.3027	-0.7146
Environmental	New Zealand	-1.2294	-1.2535
Environmental	Norway	-1.6491	-1.7885
Environmental	Portugal	0.7275	0.4495
Environmental	Spain	-0.0073	-0.0681
Environmental	Sweden	-0.1084	-0.2529
Environmental	Switzerland	-0.0244	-0.1281
Environmental	United Kingdom	-0.5226	-0.5620
Environmental	United States	0.7230	0.7743
Social	Australia	-1.3390	-1.3106
Social	Austria	0.6442	0.6112
Social	Belgium	1.5934	1.5264
Social	Canada	-0.4355	-0.4318
Social	Denmark	-1.8637	-1.8827
Social	Finland	0.0145	0.0803
Social	France	0.2401	0.1786
Social	Germany	-0.1600	-0.2533
Social	Greece	2.1797	2.1558
Social	Ireland	-0.5328	-0.7864
Social	Italy	2.2970	2.2620
Social	Japan	-0.5617	-0.5189
Social	Netherlands	-0.1230	-0.2868

Social	New Zealand	-0.6931	-0.5235
Social	Norway	0.4222	0.4655
Social	Portugal	0.1835	0.1983
Social	Spain	-0.8754	-0.9548
Social	Sweden	-1.5196	-1.5942
Social	Switzerland	0.8505	0.8295
Social	United Kingdom	-0.9100	-1.0340
Social	United States	-0.8428	-0.9023

```
vpi.original %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  spread(Dimension, VPI) %>%
  mutate(VPI = (`Top-down (VPI)` + `Bottom-up (VPI)`) / 2) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year)) %>%
  mutate(Dimension = fct_relevel(Dimension, c("VPI",
                                              "Top-down (VPI)",
                                              "Bottom-up (VPI)"))) %>%
  filter(Dimension == "VPI") %>%
  ggplot(aes(x = Year, y = VPI, color = Sector)) +
  geom_line() +
  facet_wrap(~ Country) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_discrete_qualitative(rev = TRUE) +
  ggtitle("Vertical Policy-Process Integration")
```

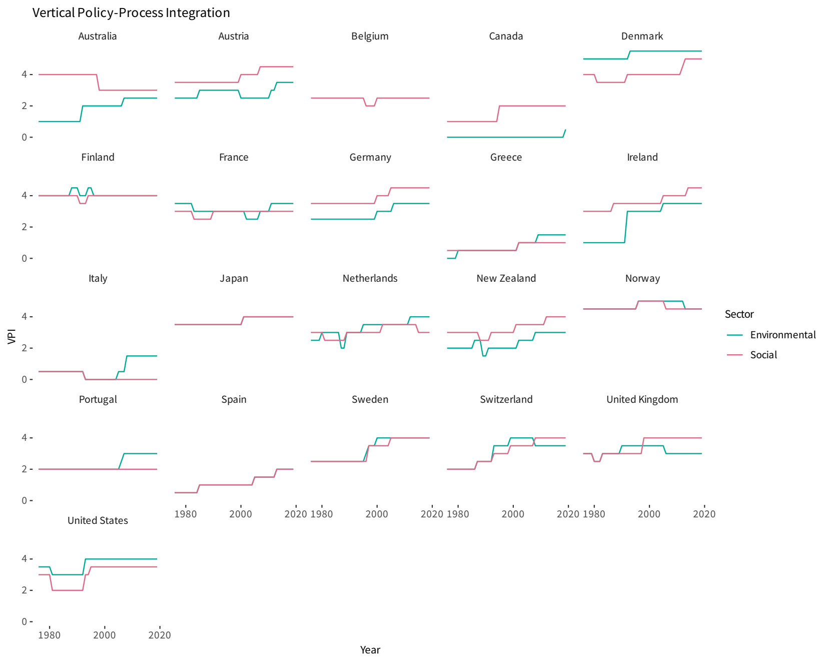


Figure 2.35: Sector-specific evolution of vertical policy integration.

```
#
# Figure 4
#
vpi.original %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  spread(Dimension, VPI) %>%
  mutate(VPI = (`Top-down (VPI)` + `Bottom-up (VPI)`) / 2) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year)) %>%
  mutate(Dimension = fct_relevel(Dimension, c("VPI",
                                              "Top-down (VPI)",
                                              "Bottom-up (VPI)"))) %>%
  filter(Dimension == "VPI") %>%
  ggplot(aes(x = Year, y = VPI, color = Sector)) +
  geom_line(alpha = 0.9) +
```

```
  facet_wrap(~ Country, ncol = 7) +
  #scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_x_continuous(breaks = c(1985, 2000, 2015)) +
  scale_color_manual(values = c("gray30", "gray70")) +
  theme(legend.position = "bottom") +
  ggtitle("Vertical Policy-Process Integration")
```

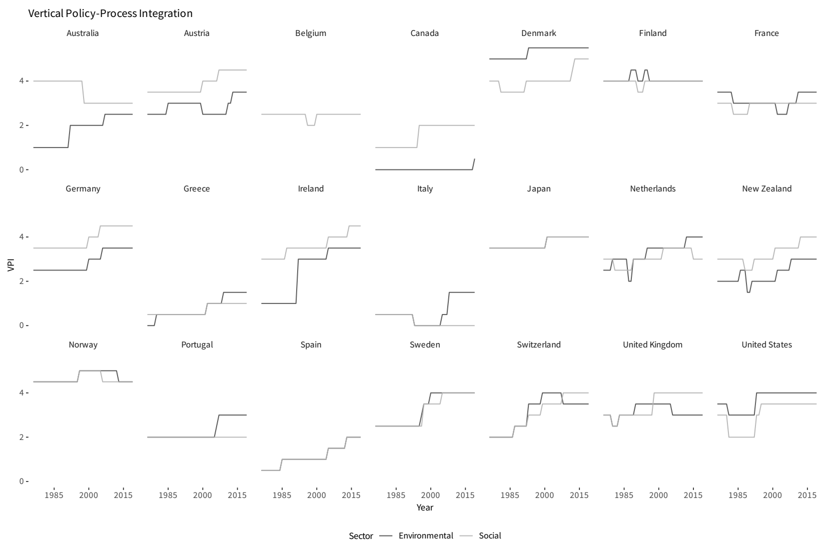


Figure 2.36: Sector-specific evolution of vertical policy integration.

```
#
# Figure A1
#
dims.vpi <- vpi.original %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  spread(Dimension, VPI) %>%
  mutate(VPI = (`Top-down (VPI)` + `Bottom-up (VPI)`) / 2) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year)) %>%
  mutate(Dimension = fct_relevel(Dimension, c("VPI",
                                              "Top-down (VPI)",
                                              "Bottom-up (VPI)")))


f1 <- dims.vpi %>%
  filter(Sector == "Environmental") %>%
  ggplot(aes(x = Year, y = VPI)) +
  geom_line(aes(lty = Dimension)) +
  facet_wrap(~ Country, ncol = 7) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_discrete_qualitative(rev = TRUE) +
  theme(legend.position = "none") +
  ggtitle("Environmental")

f2 <- dims.vpi %>%
  filter(Sector == "Social") %>%
  ggplot(aes(x = Year, y = VPI)) +
  geom_line(aes(lty = Dimension)) +
  facet_wrap(~ Country, ncol = 7) +
  scale_x_continuous(breaks = c(1980, 2000, 2020)) +
  scale_color_discrete_qualitative(rev = TRUE) +
  theme(legend.position = "bottom") +
  ggtitle("Social")

cowplot::plot_grid(f1, f2, nrow = 2)


Y.df %>%
  filter(!Country %in% paste0("Z-", sprintf("%02d", 1:(19+29)))) %>%
  ggplot(aes(x = Year, y = `Implementation deficit`)) +
  geom_line() +
```

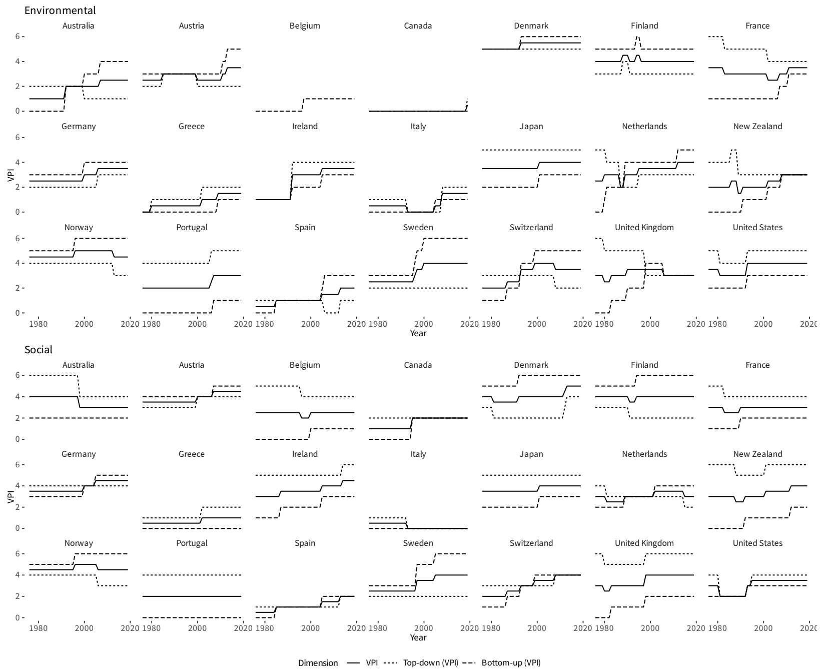


Figure 2.37: Sector-specific evolution of vertical policy integration.

```
  facet_grid(Country ~ Sector) +
  scale_x_continuous(breaks = seq(1980, 2010, 10)) +
  theme(strip.text.y = element_text(angle = 0, hjust = 0),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank())


d %>%
  filter(!Country %in% paste0(”Z-”, sprintf(”%02d”, 1:(19+29)))) %>%
  select(Country, Year, Sector,
         PS, IC,
         `Burden capacity gap` = `Implementation deficit`) %>%
  rename(`Portfolio size` = PS,
         `Implementation capacity` = IC) %>%
  gather(Variable, value, -c(Country, Year, Sector)) %>%
  ggplot(aes(x = Year, y = value, color = Sector)) +
  geom_line() +
  ylab(””) +
  #facet_grid(Country ~ Variable, scales = ”free”) +
  facet_grid(Variable ~ Country, scales = ”free”) +
  scale_x_continuous(breaks = seq(1980, 2010, 10)) +
  theme(strip.text.y = element_text(angle = 0, hjust = 0),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        legend.position=”bottom”) +
  scale_color_discrete_qualitative(palette = ”Set2”, rev = TRUE)
```

Correlations between main variables

```
dcor.e <- d %>%
  filter(!str_detect(Country, ”^Z-”)) %>%
  filter(Sector == ”Environmental”) %>%
  select(Country, Year,
         `Burden capacity gap` = `Implementation deficit`,
         VPI,
         Debt, `Electoral competition`, EU,
         GDPpc, `Political constraints`,
#         `Portfolio size` = PS,
         `Implementation burden` = PS,
         `Bottom-up (VPI)`, `Top-down (VPI)`,
         `Implementation capacity` = IC) %>%
```

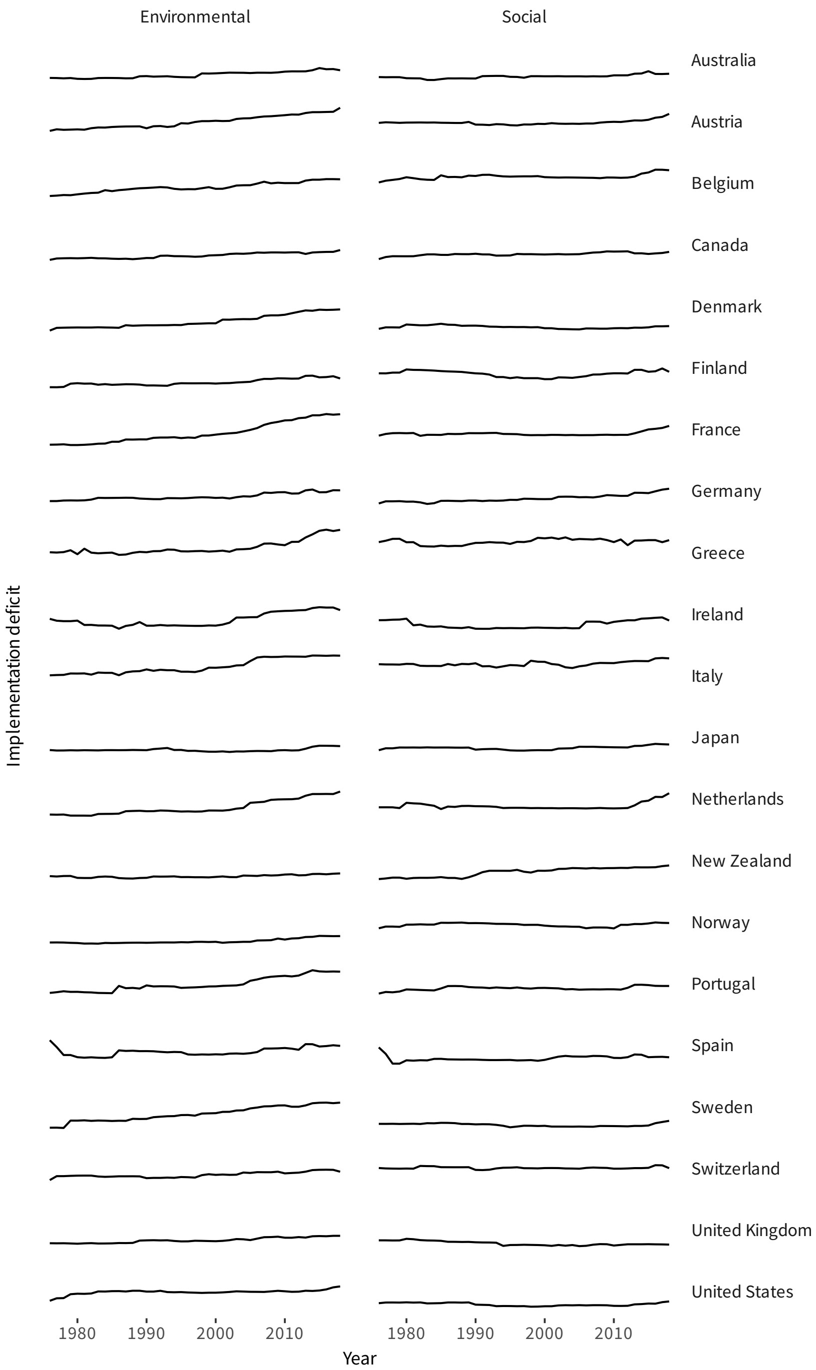


Figure 2.38: Temporal evolution of Portfolio size over implementation capacity, by sector.

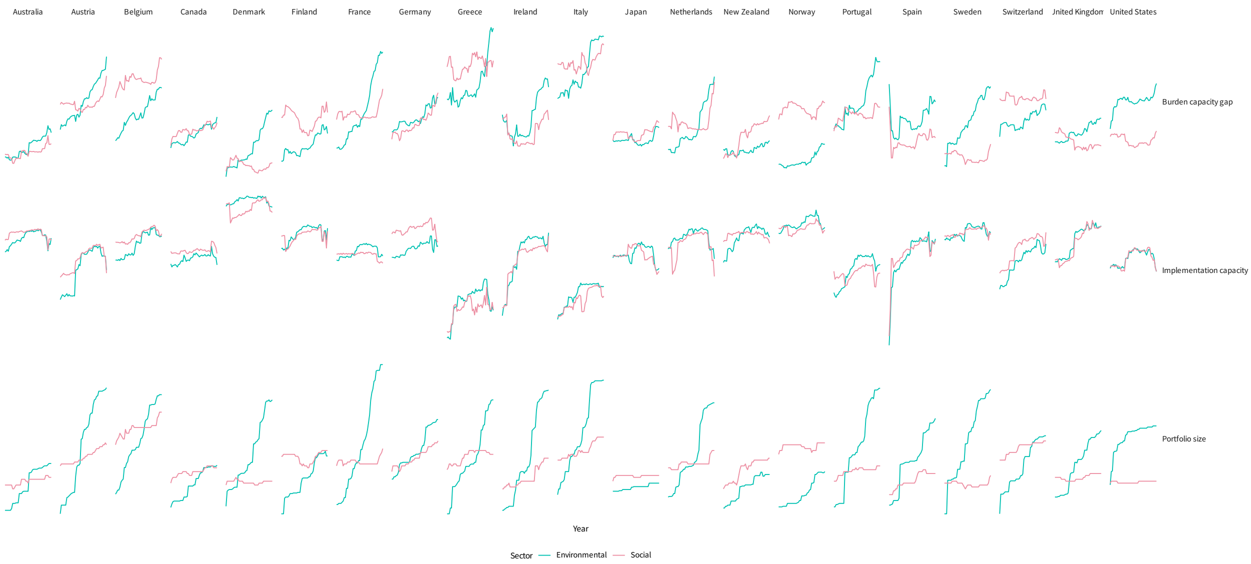


Figure 2.39: Temporal evolution main variables governing outcome, by sector.

```
  unique() %>%
  select(-c(Country, Year))

dcor.s <- d %>%
  filter(!str_detect(Country, "^Z-")) %>%
  filter(Sector == "Social") %>%
  select(Country, Year,
         `Burden capacity gap` = `Implementation deficit`,
         VPI,
         Debt, `Electoral competition`, EU,
         GDPpc, `Political constraints`,
#         `Portfolio size` = PS,
         `Implementation burden` = PS,
         `Bottom-up (VPI)`, `Top-down (VPI)`,
         `Implementation capacity` = IC) %>%
  unique() %>%
  select(-c(Country, Year))

# corrplot
par(mfrow = c(1, 2))
corrplot::corrplot(cor(dcor.e, use = "complete.obs"),
  method = "number", diag = FALSE, type = "lower",
#  tl.cex = 0.6,
  title = "Environmental")

corrplot::corrplot(cor(dcor.s, use = "complete.obs"),
  method = "number", diag = FALSE, type = "lower",
  tl.cex = 0.6,
  title = "Social")


#
# Figure A5
#
# ggplotcorr
f1 <- ggcorrplot::ggcorrplot(cor(dcor.e, use = "complete.obs"),
  method = "circle", hc.order = TRUE, outline.color = "white",
  type = "lower", lab = TRUE, digits = 2,
  title = "Environmental")

f2 <- ggcorrplot::ggcorrplot(cor(dcor.s, use = "complete.obs"),
  method = "circle", hc.order = TRUE, outline.color = "white",
  type = "lower", lab = TRUE, digits = 2,
  title = "Social")

cowplot::plot_grid(f1, f2, ncol = 2)
```

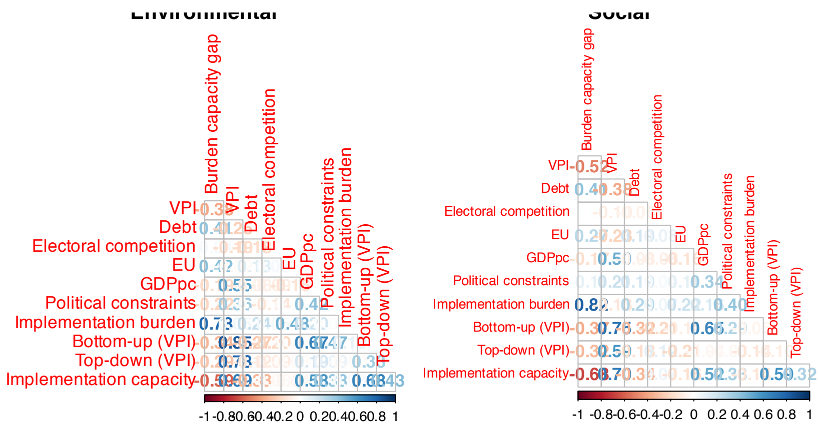


Figure 2.40: Correlation matrix between the variables that constitute the outcome and VPI.

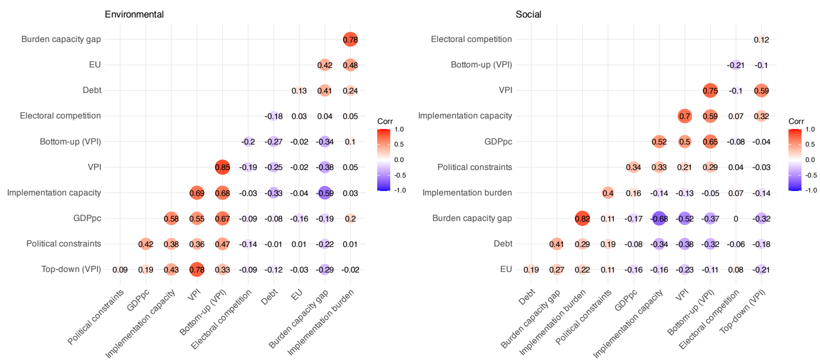


Figure 2.41: Correlation matrix between the variables that constitute the outcome and VPI.

```
# ggplotcorr
f1 <- ggcorrplot::ggcorrplot(cor(dcor.e, use = "complete.obs"),
  method = "circle", hc.order = TRUE, outline.color = "white",
  type = "lower", lab = TRUE, digits = 2,
  title = "Environmental")

f2 <- ggcorrplot::ggcorrplot(cor(dcor.s, use = "complete.obs"),
  method = "circle", hc.order = TRUE, outline.color = "white",
  type = "lower", lab = TRUE, digits = 2,
  title = "Social")

cowplot::plot_grid(f1, f2, nrow = 2)


# tab-descriptives
tc <- "Descriptive statistics."

tbns <- d %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Country, Year,
         VPI,
         Debt, `Electoral competition`, EU,
         GDPpc, `Political constraints`
         ) %>%
  mutate(`GDPpc (in 1,000s)` = GDPpc / 1000) %>%
  select(-GDPpc) %>%
  pivot_longer(-c(Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Variable) %>%
  summarize(Min = min(value, na.rm = TRUE),
            Mean = mean(value, na.rm = TRUE),
            Median = median(value, na.rm = TRUE),
            Max = max(value, na.rm = TRUE),
            SD = sd(value, na.rm = TRUE)) %>%
  ungroup()
```

Figure 2.42: Correlation matrix between the variables that constitute the outcome and VPI.

```
tbs <- d %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Country, Year,
         `Burden capacity gap` = `Implementation deficit`,
         `Implementation burden` = PS,
         `Bottom-up (VPI)`, `Top-down (VPI)`,
         `Implementation capacity` = IC) %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Variable) %>%
  summarize(Min = min(value, na.rm = TRUE),
            Mean = mean(value, na.rm = TRUE),
            Median = median(value, na.rm = TRUE),
            Max = max(value, na.rm = TRUE),
            SD = sd(value, na.rm = TRUE)) %>%
  ungroup()

tb <- bind_rows(tbns, tbs) %>%
  relocate(Sector)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tb2 <- tb %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-descriptives}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows("Environmental", rws.env[1], rws.env[length(rws.env)]) %>%
  pack_rows("Social", rws.soc[1], rws.soc[length(rws.soc)])

print(tb2)
```

Table 2.4: Descriptive statistics.

Variable	Min	Mean	Median	Max	SD
Debt	2.290	61.4502	55.2550	249.1100	34.7847
EU	0.000	0.5759	1.0000	1.0000	0.4943
Electoral competition	0.000	0.2393	0.1864	0.7538	0.2172
GDPpc (in 1,000s)	10.766	38.5091	36.2863	92.1195	14.9526
Political constraints	0.000	0.4730	0.4691	0.7181	0.0938
VPI	0.000	2.7975	3.0000	5.5000	1.3191
Environmental					
Bottom-up (VPI)	0.000	2.3787	2.0000	6.0000	1.9468
Burden capacity gap	-2.317	-0.1081	-0.2608	3.7072	1.1209
Implementation burden	0.000	0.1421	0.1246	0.3969	0.0930
Implementation capacity	-2.050	0.2083	0.2596	1.5095	0.5979
Top-down (VPI)	0.000	2.9174	3.0000	6.0000	1.5924
Social					
Bottom-up (VPI)	0.000	2.3787	2.0000	6.0000	1.9341
Burden capacity gap	-2.171	-0.0682	-0.2339	3.0530	1.1392
Implementation burden	0.051	0.1278	0.1276	0.2704	0.0417
Implementation capacity	-1.831	0.2284	0.3249	1.4766	0.5891
Top-down (VPI)	0.000	3.4109	4.0000	6.0000	1.5401

```
tb2 %>%
  save_kable(file = "TAB-descriptives.tex")

# Use the original vpi, instead of the one already lagged 3 years
vpi.original.3types <- left_join(
  vpi.original %>%
    group_by(Sector, Country, Year) %>%
    summarize(VPI = mean(VPI)) %>%
    ungroup(),
  vpi.original %>%
    pivot_wider(names_from = Dimension, values_from = VPI) %>%
    ungroup())
```

```
vpi.ic <- d %>%
  select(Country, Sector, Year,
         `Implementation capacity` = IC) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  left_join(vpi.original.3types) %>%
  unique() %>%
  pivot_longer(-c(Country, Sector, Year, `Implementation capacity`),
    names_to = "VPI specification", values_to = "VPI")


vpi6.ic <- d %>%
  select(Country, Sector, Year,
         `Implementation capacity` = IC) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  left_join(vpi.es %>%
            select(Sector, Country, Year, Variable, VPI = value)) %>%
  unique()

vpi.ic %>%
  ggplot(aes(x = `Implementation capacity`, y = jitter(VPI))) +
  geom_point(alpha = 0.2) +
  facet_grid(Sector ~ `VPI specification`)
```

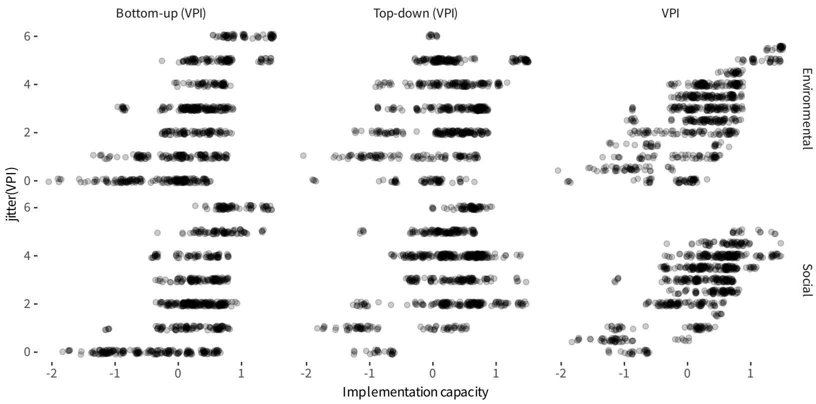


Figure 2.43: Implementation capacity and VPI.

```
vpi.ic.for.lags <- vpi.ic %>%
  filter(`VPI specification` == "VPI") %>%
  select(-`VPI specification`) %>%
  arrange(Sector, Country, Year)

vpi.ic.for.lags.env <- vpi.ic.for.lags %>%
  filter(Sector == "Environmental")

vpi.ic.for.lags.env %>%
  pivot_longer(-c(Country, Sector, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Variable) %>%
  mutate(value = std1(value)) %>%
  ungroup() %>%
  ggplot(aes(x = Year, y = value, color = Variable)) +
  geom_line() +
  facet_wrap(~ Country)


vpi.ic.for.lags.soc <- vpi.ic.for.lags %>%
  filter(Sector == "Social")

vpi.ic.for.lags.soc %>%
  pivot_longer(-c(Country, Sector, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Variable) %>%
  mutate(value = std1(value)) %>%
  ungroup() %>%
  ggplot(aes(x = Year, y = value, color = Variable)) +
  geom_line() +
  facet_wrap(~ Country)
```

Figure 2.44: Normalized VPI and Implementation capacities. Environmental sector.

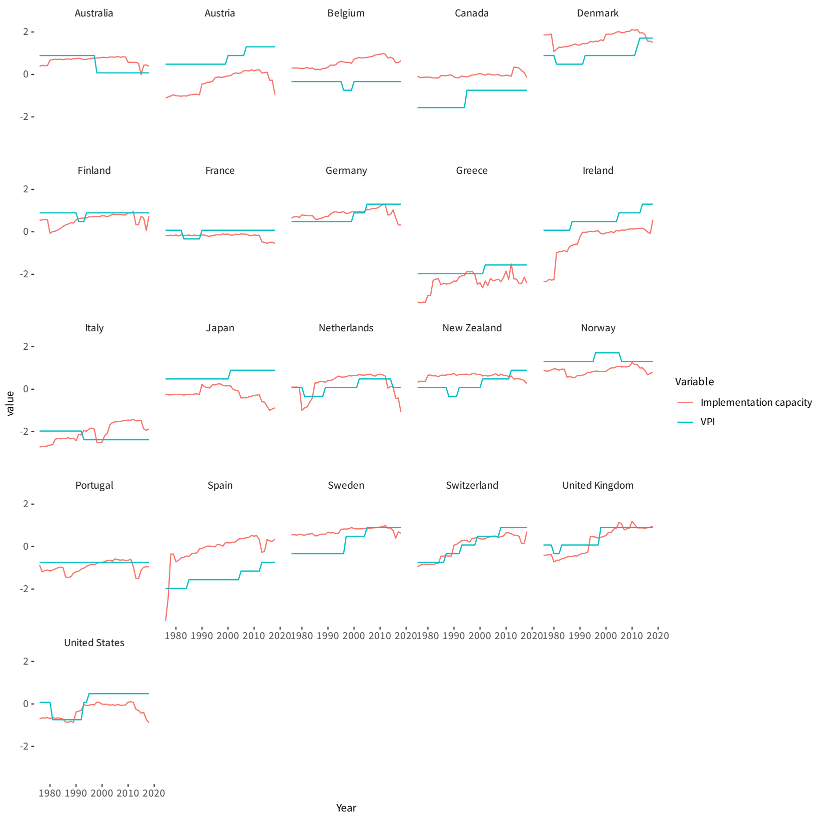


Figure 2.45: Normalized VPI and Implementation capacities. Social sector.

2.4 Example portfolio: Portugal

```
load("./data-accupol.RData")
PE.20 <- mutate(PE, Version = "Accupol")
environmental.cleaned <- PE.20 %>%
  select(-Version) %>%
  left_join(instrument.labels %>%
            filter(Sector == "Environmental") %>%
            mutate(instrument.id = as.factor(as.character(instrument.id))) %>%
            select(instrument, instrument.id),
    by = c("Instrument" = "instrument.id")) %>%
  mutate(Instrument = as.factor(instrument)) %>%
  select(-instrument) %>%
  mutate(Target = as.factor(as.numeric(Target)))

environmental.portugal.start <- environmental.cleaned %>%
  filter(Country == "Portugal") %>%
  filter(Year == 1976)
sis <- filter(pp_measures(environmental.portugal.start), Measure == "Size")$value
f1 <- pp_plot(environmental.portugal.start, subtitle = FALSE, caption = NULL) +
  theme(axis.text.x = element_text(size = 6)) +
  theme(axis.text.y = element_text(size = 10)) +
  #scale_y_discrete(breaks =
  #theme(axis.ticks.x =
  ggtitle(paste0("Portugal/Environmental (1976): ", percent(sis)))

environmental.portugal.end <- environmental.cleaned %>%
  filter(Country == "Portugal") %>%
  filter(Year == 2018)
sie <- filter(pp_measures(environmental.portugal.end), Measure == "Size")$value

# New positions covered at the end
environmental.portugal.change <- bind_rows(environmental.portugal.start, environmental.portugal.end) %>%
  spread(Year, covered) %>%
  mutate(`2018` = ifelse(`2018` == 1, `1976` + `2018`, 0)) %>%
  gather(Year, covered, -c(Country, Sector, Instrument, Target)) %>%
  filter(Year == "2018")



f2 <- pp_plot(environmental.portugal.change, subtitle = FALSE, caption = NULL) +
#  geom_tile(data = environmental.portugal.change) +
#  geom_tile(data = environmental.portugal.change) +
  scale_fill_manual(values = c("0" = "white", "1" = "gray", "2" = "black")) +
  theme(axis.text.x = element_text(size = 6)) +
  theme(axis.text.y = element_text(size = 10)) +
  ggtitle(paste0("Portugal/Environmental (2018): ", percent(sie)))

plot_grid(f1, f2, ncol = 2)
```

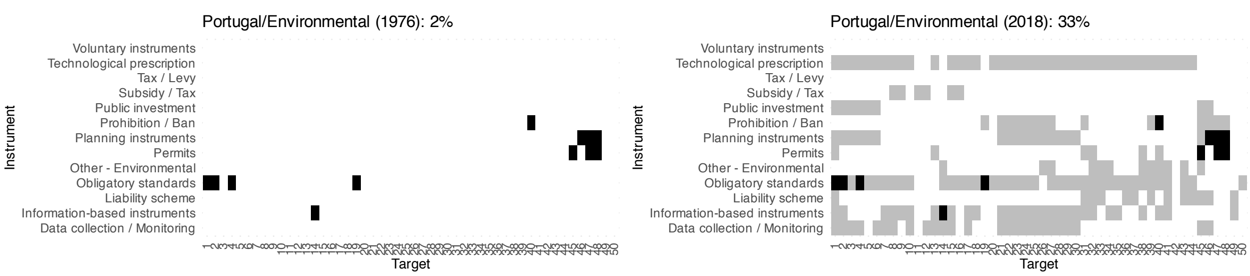


Figure 2.46: Environmental portfolio in Portugal, in 1976 and 2018, with its respective sizes. Grey tiles specify the portfolio spaces newly covered between the two periods.

```
#
# Figure 2
#
plot_grid(f1, f2, nrow = 2)
```

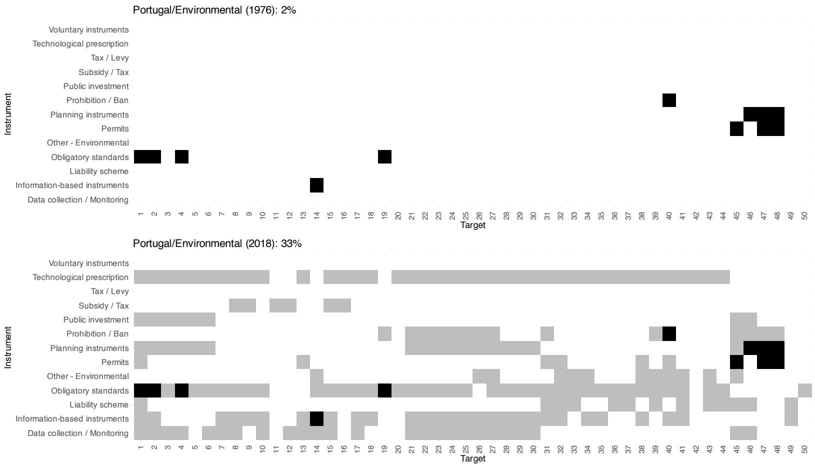


Figure 2.47: Environmental portfolio in Portugal, in 1976 and 2018, with its respective sizes. Grey tiles specify the portfolio spaces newly covered between the two periods. Figure 2 in the article.

2.5 Theoretical learning

```
library(ggplot2)
library(dplyr)
library(ggthemes)
library(tidyr)
library(forcats)


d <- tibble(
  `Number of policy instruments (of the same type)` = 1:10,
  `No learning` = rep(1, 10),
  `Continuous` = 1 - (0:9 / 10),
  `Steep` = `No learning` / (1:10)^2,
  `Capped` = c(1, rep(0.5, 9))
)

dl <- d %>%
  pivot_longer(-`Number of policy instruments (of the same type)`,
               names_to = "Learning",
               values_to = "Marginal\nimplementation\nburden") %>%
  mutate(Learning = fct_relevel(Learning, c("No learning", "Continuous", "Capped", "Steep")))

f <- ggplot(dl, aes(x = `Number of policy instruments (of the same type)`,
               y = `Marginal\nimplementation\nburden`)) +
  geom_line(color = "black", size = 2) +
  facet_grid(~ Learning) +
#  theme_minimal() +
#  guides(x = "none", y = "none") +
#  geom_rangeframe() +
  theme_tufte(base_family = "Helvetica") +
  theme(axis.ticks = element_blank(),
        axis.text = element_blank(),
        axis.line = element_line(colour = 'black', size=0.5, linetype='solid'))


#
# Figure 8
#
library(ggplot2)
library(dplyr)
library(ggthemes)
library(tidyr)
library(forcats)


d <- tibble(
  `Number of policy instruments (of the same type)` = 1:10,
  `No learning` = rep(1, 10),
  `Continuous` = 1 - (0:9 / 10),
  `Steep` = `No learning` / (1:10)^2,
  `Capped` = c(1, rep(0.5, 9))
```

```
)

dl <- d %>%
  pivot_longer(-`Number of policy instruments (of the same type)`,
               names_to = "Learning",
               values_to = "Marginal\nimplementation\nburden") %>%
  mutate(Learning = fct_relevel(Learning, c("No learning", "Continuous", "Capped", "Steep")))

f <- ggplot(dl, aes(x = `Number of policy instruments (of the same type)`,
               y = `Marginal\nimplementation\nburden`)) +
  geom_line(color = "black", size = 2) +
  facet_grid(~ Learning) +
  theme_tufte(base_family = "Helvetica") +
  theme(axis.ticks = element_blank(),
        axis.text = element_blank(),
        axis.line = element_line(colour = 'black', size=0.5, linetype='solid'))

print(f)
```

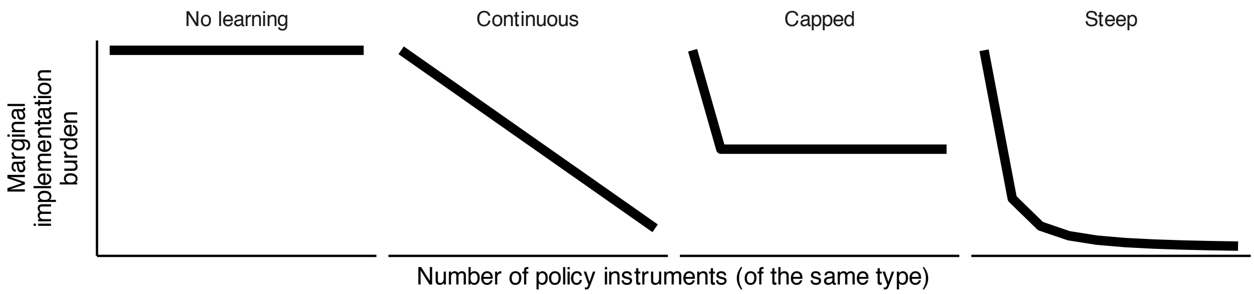


Figure 2.48: Theoretical learning. Figure 7 in the article.

3

Explain Portfolio size over implementation capacity: VPI in 2 dimensions

m-313

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  # Outcome variable
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  # Sample countries
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is “Regular : Only sample countries : Not bounded : Original ratio”.

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
```

```
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
```

```
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
```

```
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
```

```
           `Top-down (VPI)`, `Bottom-up (VPI)`,
           `Contiguity dependency (BCG)`,
           `Trade dependency (BCG)`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
```

```
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
##  $ Y         : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
##  $ nC        : int 69
##  $ nS        : int 2
##  $ nY        : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
```

```
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV        : int 6
##  $ XS        : num [1:2, 1:69, 1:43, 1:4] -0.4505 0.0466 0.295 0.5447 -0.4505 ...
##  $ nVS       : int 4
##  $ b0        : num [1:10] 0 0 0 0 0 0 0 0 0 0
##  $ B0        : num [1:10, 1:10] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
```

```
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))
```

```
load(file = paste("samples-", M, ".RData", sep = ""))
```

```
ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))
```

```
L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)
```

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                 Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-313.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")
```

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                               Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-313.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  #facet_grid(Sector ~ .) +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

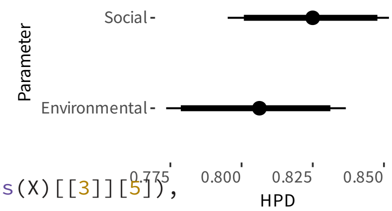


Figure 3.1: Auto-regressive components.

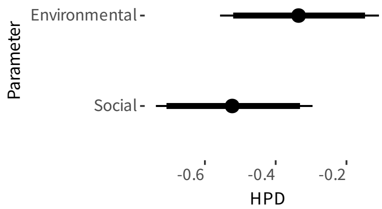


Figure 3.2: Heteroskedasticity controls (Political constraints).

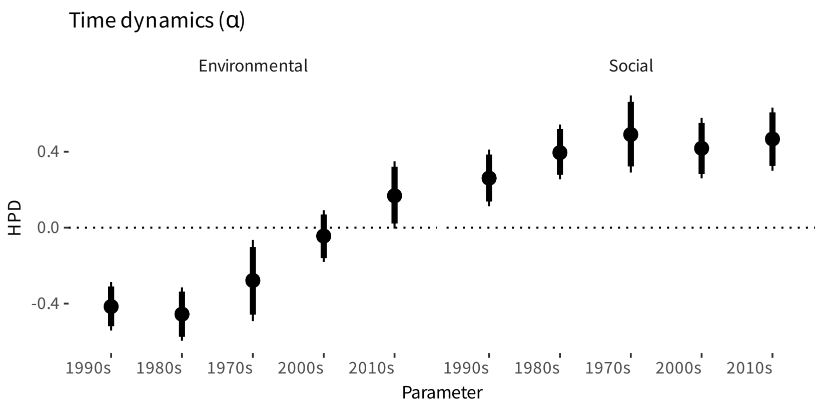


Figure 3.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-313.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

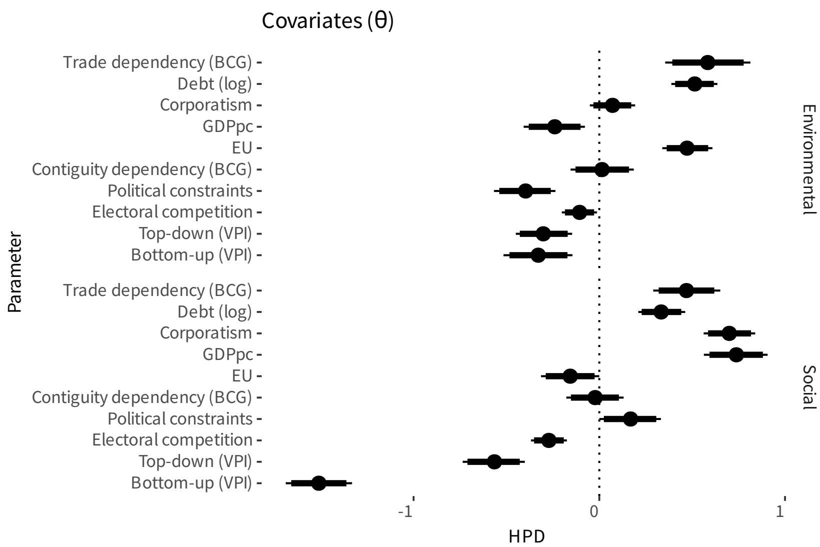


Figure 3.4: Covariates.

```
#
# Figure 7
#
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  mutate(Covariate = fct_relevel(Covariate, "Bottom-up (VPI)", "Top-down (VPI)", after = Inf))
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
```

```
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

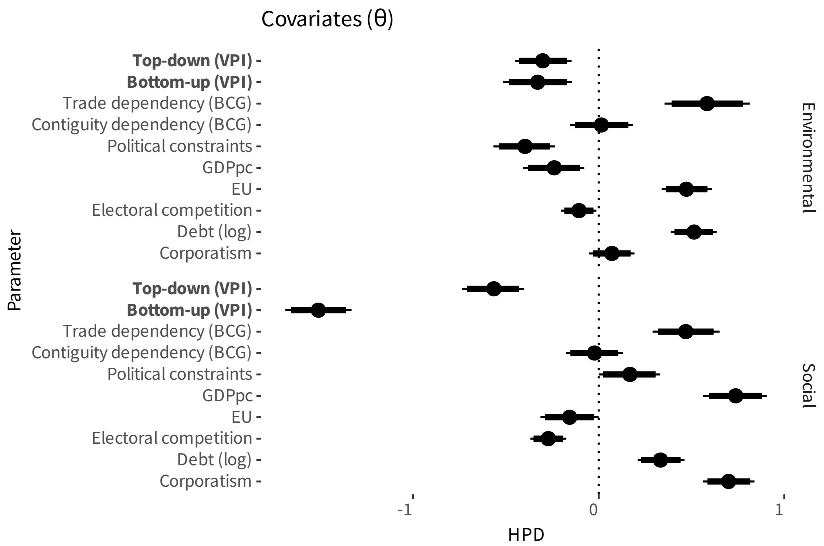


Figure 3.5: Covariates. Figure 7 of the article.

3.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5864
Social	0.5686

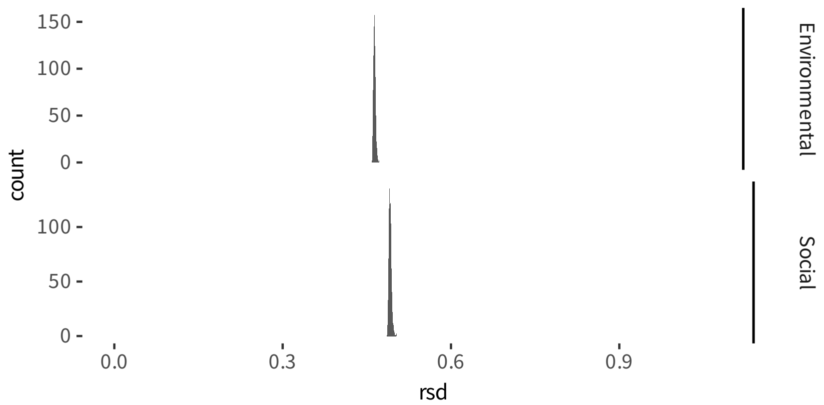


Figure 3.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. VPI in 2 dimensions. Coefficient point estimates (median of the posterior distributio


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-313}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a7.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.58	(0.116)	[0.35 : 0.81]
Debt (log)	0.51	(0.063)	[0.39 : 0.64]
EU	0.47	(0.067)	[0.34 : 0.61]
Political constraints	-0.40	(0.085)	[-0.57 : -0.24]
Bottom-up (VPI)	-0.33	(0.095)	[-0.52 : -0.14]
Top-down (VPI)	-0.30	(0.078)	[-0.45 : -0.15]
GDPpc	-0.24	(0.084)	[-0.41 : -0.077]
Electoral competition	-0.11	(0.048)	[-0.2 : -0.012]
Corporatism	0.07	(0.062)	[-0.051 : 0.19]
Contiguity dependency (BCG)	0.01	(0.088)	[-0.15 : 0.19]
** Goodness of fit (R2)	0.59	(0.0016)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
Bottom-up (VPI)	-1.51	(0.091)	[-1.7 : -1.3]
GDPpc	0.74	(0.088)	[0.56 : 0.91]
Corporatism	0.70	(0.071)	[0.56 : 0.84]
Top-down (VPI)	-0.56	(0.086)	[-0.74 : -0.4]
Trade dependency (BCG)	0.47	(0.091)	[0.29 : 0.65]
Debt (log)	0.33	(0.065)	[0.21 : 0.46]
Electoral competition	-0.27	(0.049)	[-0.37 : -0.17]
Political constraints	0.17	(0.086)	[-4.4e-05 : 0.33]
EU	-0.16	(0.08)	[-0.31 : -2e-04]
Contiguity dependency (BCG)	-0.02	(0.078)	[-0.18 : 0.13]
** Goodness of fit (R2)	0.57	(0.00186)	[0.56 : 0.57]

Table 3.1: Model parameters. VPI in 2 dimensions. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

4
Explain Portfolio size over implementation capacity: Continuous learning (instruments)

m-318

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Continuous : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Continuous : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                  `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                max(pc$`Political constraints`, na.rm = TRUE),
                                                length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
```

```
        `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
```

```
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
#par <- c(par, "Theta", "sigma_theta")
#par <- c(par, "nu")
#par <- c(par, "sigma")
par <- c(par, "Sigma")
par <- c(par, "rho")
#par <- c(par, "pi")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1

chains <- 1
adapt <- 2e2
burnin <- 8e3
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
##  $ Y         : num [1:2, 1:69, 1:43] -1.614 -1.411 -0.566 0.745 -0.95 ...
```

```
##  $ nC        : int 69
##  $ nS        : int 2
##  $ nY        : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV        : int 6
##  $ XS        : num [1:2, 1:69, 1:43, 1:3] -0.00295 0.01435 -0.10522 0.7112 -0.6143 ...
##  $ nVS       : int 3
##  $ b0        : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0        : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
```

```
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
set.seed(14720)
rj <- run.jags(model = paste(”models/model-”, M, ”.bug”, sep = ””),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = ”glm”,
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste(”samples-”, M, ”.RData”, sep = ””))

load(file = paste(”samples-”, M, ”.RData”, sep = ””))

ggmcmc(ggs(s, family = ”theta|rho|pi|lambda|alpha”, sort = FALSE),
           file = paste0(”ggmcmc-all-”, M, ”.pdf”),
           plot = c(”traceplot”, ”crosscorrelation”,
                    ”running”, ”caterpillar”))

L.rho <- plab(”rho”, list(Sector = sector.label))
S.rho <- ggs(s, family = ”rho”, par_labels = L.rho)
ggs_caterpillar(S.rho)
```

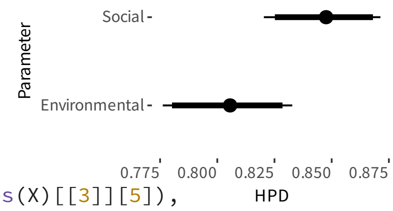


Figure 4.1: Auto-regressive components.

```
L.lambda <- plab(”lambda”, list(Variable = c(”(Intercept)”, dimnames(X)[[3]][5]),
                                 Sector = sector.label))
S.lambda <- ggs(s, family = ”^lambda”, par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0(”ci-lambda-318.RData”))


S.lambda %>%
  filter(Variable != ”(Intercept)”) %>%
ggs_caterpillar(label = ”Sector”)
```

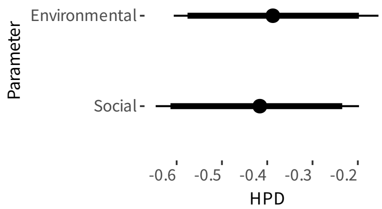


Figure 4.2: Heteroskedasticity controls (Political constraints).

```
L.alpha <- plab(”alpha”, list(Sector = sector.label,
                               Decade = decade.label))
S.alpha <- ggs(s, family = ”^alpha”, par_labels = L.alpha)
```

```
ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-318.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

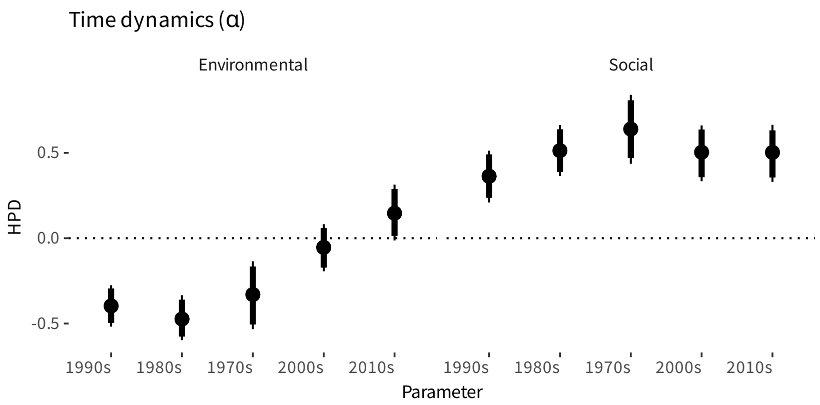


Figure 4.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-318.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

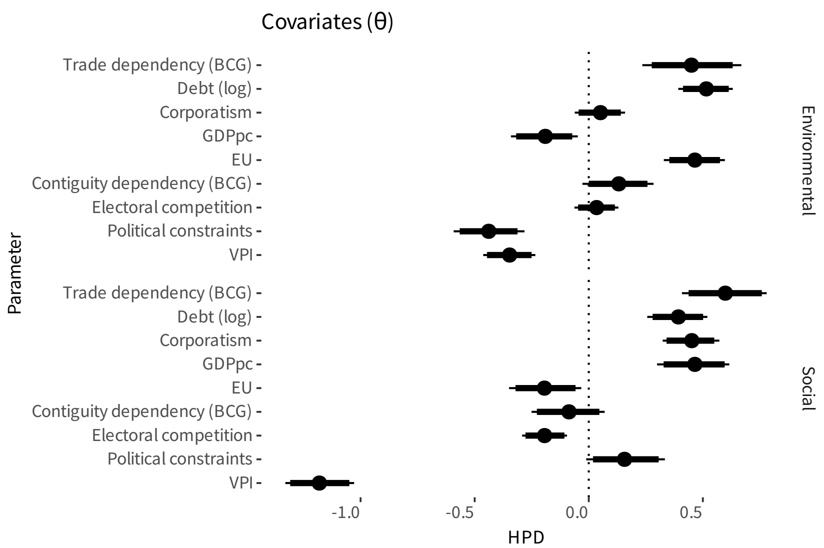


Figure 4.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

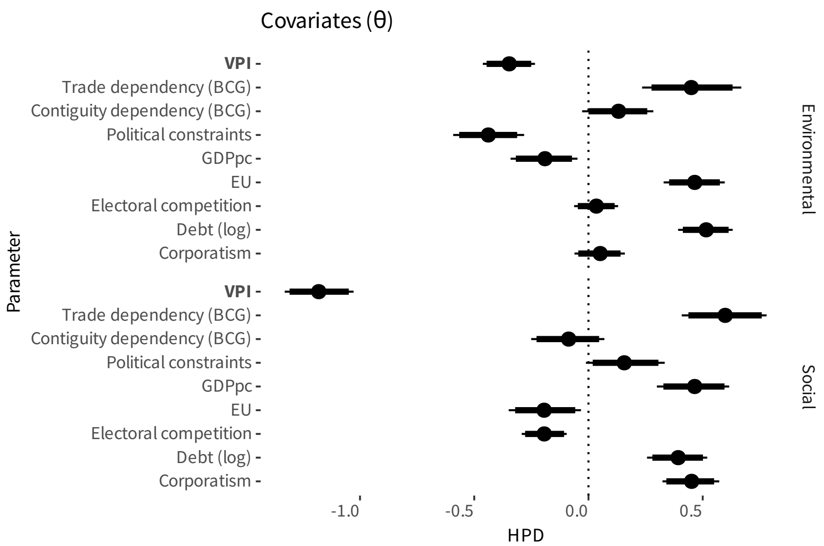


Figure 4.5: Covariates.

4.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
```

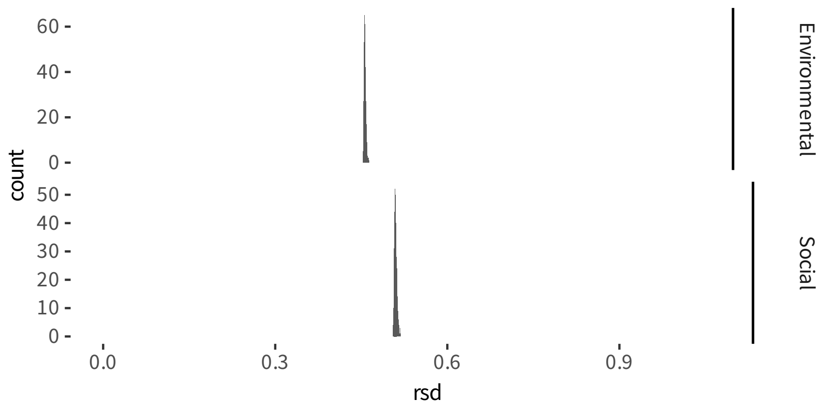


Figure 4.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5843
Social	0.5500

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Continuous learning (instruments). Coefficient point estimates (median of the posteri


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-318}", tc), label = NA,
```

```
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Debt (log)	0.52	(0.06)	[0.39 : 0.63]
EU	0.46	(0.069)	[0.33 : 0.6]
Trade dependency (BCG)	0.45	(0.109)	[0.23 : 0.67]
Political constraints	-0.44	(0.079)	[-0.59 : -0.28]
VPI	-0.35	(0.059)	[-0.46 : -0.23]
GDPpc	-0.19	(0.075)	[-0.34 : -0.048]
Contiguity dependency (BCG)	0.13	(0.081)	[-0.028 : 0.28]
Corporatism	0.05	(0.056)	[-0.061 : 0.16]
Electoral competition	0.03	(0.049)	[-0.063 : 0.13]
** Goodness of fit (R2)	0.58	(0.00147)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.18	(0.079)	[-1.3 : -1]
Trade dependency (BCG)	0.60	(0.095)	[0.41 : 0.78]
GDPpc	0.46	(0.082)	[0.3 : 0.62]
Corporatism	0.45	(0.064)	[0.32 : 0.57]
Debt (log)	0.39	(0.067)	[0.26 : 0.52]
EU	-0.19	(0.082)	[-0.35 : -0.033]
Electoral competition	-0.19	(0.051)	[-0.29 : -0.095]
Political constraints	0.16	(0.088)	[-0.011 : 0.33]
Contiguity dependency (BCG)	-0.09	(0.084)	[-0.25 : 0.07]
** Goodness of fit (R2)	0.55	(0.0018)	[0.55 : 0.55]

Table 4.1: Model parameters. Continuous learning (instruments). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a8.tex")
```

5

Explain Portfolio size over implementation capacity: Steep learning (instruments)

m-319

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Steep : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Steep : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                 `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                               max(pc$`Political constraints`, na.rm = TRUE),
                                               length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
```

```
      `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
```

```
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 2
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.98 -1.32 -1.35 1.19 -1.24 ...
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
```

```
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] -0.02512 -0.00161 -0.28111 0.91844 -0.55931 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                               lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                             lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
```

```
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
set.seed(14720)
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)

L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-319.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-319.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  #facet_grid(Sector ~ .) +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

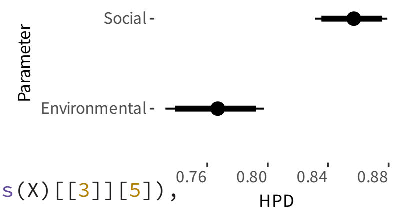


Figure 5.1: Auto-regressive components.

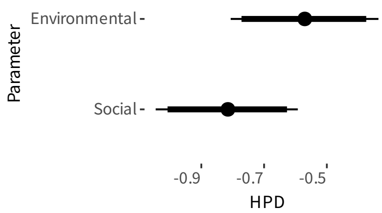


Figure 5.2: Heteroskedasticity controls (Political constraints).

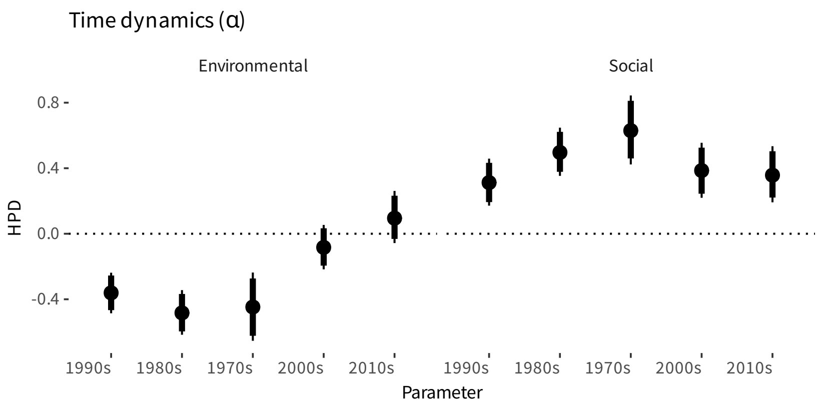


Figure 5.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-319.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

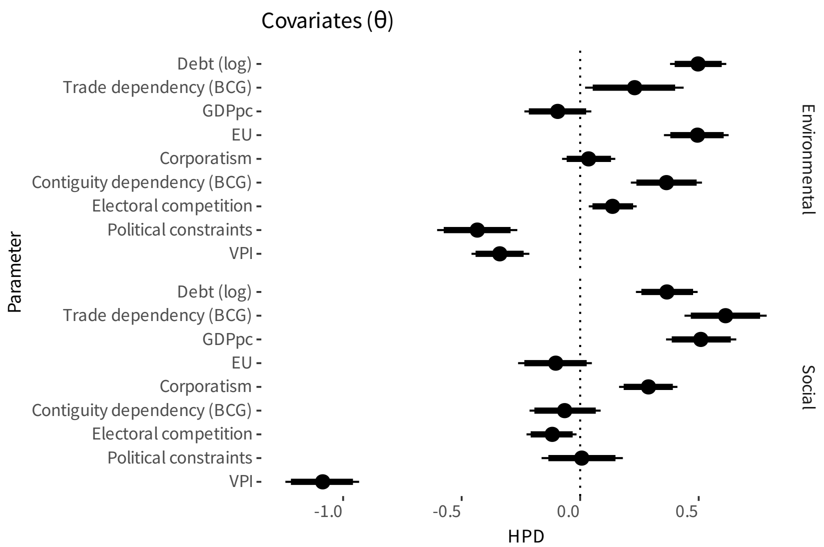


Figure 5.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
#  mutate(Covariate = fct_relevel(Covariate, "Bottom-up (VPI)", "Top-down (VPI)"))
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
#  scale_y_discrete(label = labels, breaks = breaks) +
```

```
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

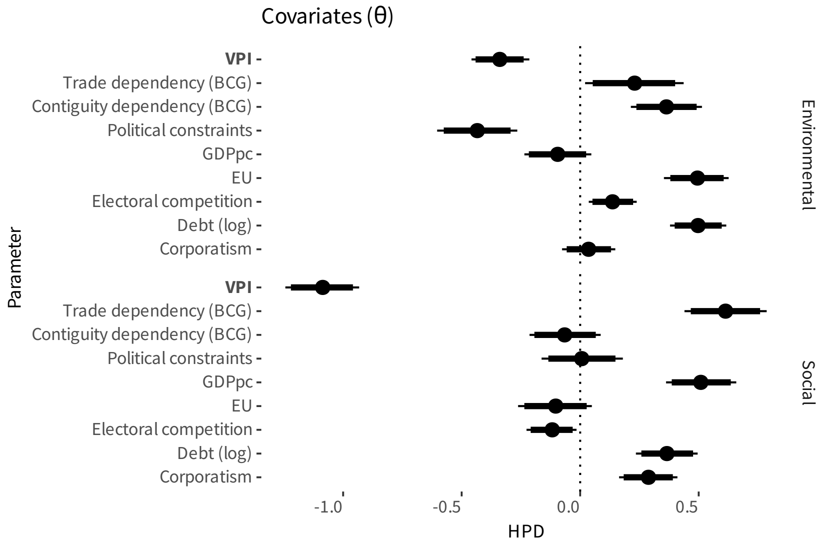


Figure 5.5: Covariates.

5.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5667
Social	0.5325

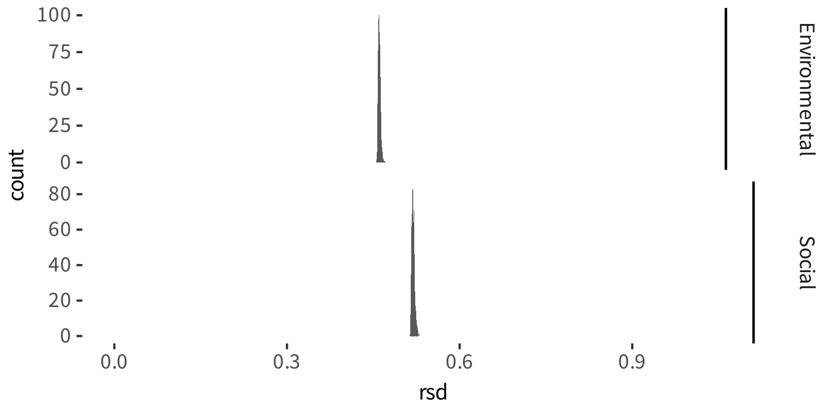


Figure 5.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Steep learning (instruments). Coefficient point estimates (median of the posterior di


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-319}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a9.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Debt (log)	0.50	(0.06)	[0.38 : 0.62]
EU	0.49	(0.069)	[0.35 : 0.63]
Political constraints	-0.43	(0.086)	[-0.6 : -0.27]
Contiguity dependency (BCG)	0.36	(0.077)	[0.21 : 0.51]
VPI	-0.34	(0.062)	[-0.46 : -0.21]
Trade dependency (BCG)	0.23	(0.106)	[0.021 : 0.44]
Electoral competition	0.14	(0.052)	[0.036 : 0.24]
GDPpc	-0.09	(0.073)	[-0.24 : 0.047]
Corporatism	0.04	(0.057)	[-0.077 : 0.15]
** Goodness of fit (R2)	0.57	(0.00183)	[0.56 : 0.57]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.09	(0.08)	[-1.2 : -0.93]
Trade dependency (BCG)	0.61	(0.09)	[0.44 : 0.79]
GDPpc	0.51	(0.075)	[0.36 : 0.66]
Debt (log)	0.37	(0.066)	[0.24 : 0.5]
Corporatism	0.29	(0.062)	[0.16 : 0.41]
Electoral competition	-0.12	(0.053)	[-0.23 : -0.015]
EU	-0.10	(0.08)	[-0.26 : 0.049]
Contiguity dependency (BCG)	-0.07	(0.077)	[-0.21 : 0.086]
Political constraints	0.01	(0.087)	[-0.16 : 0.18]
** Goodness of fit (R2)	0.53	(0.00215)	[0.53 : 0.54]

Table 5.1: Model parameters. Steep learning (instruments). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

6

Explain Portfolio size over implementation capacity: Capped learning (instruments)

m-320

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Capped : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Capped : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)
```

```
Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
```

```
        `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
```

```
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.564 -1.365 -0.523 0.726 -0.896 ...
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
```

```
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00156 0.02192 -0.09617 0.71116 -0.59639 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                               lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                             lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
```

```
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-320.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-320.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

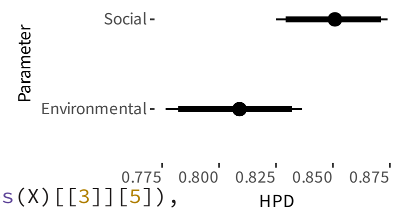


Figure 6.1: Auto-regressive components.

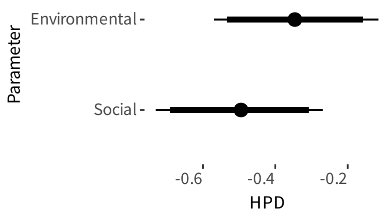


Figure 6.2: Heteroskedasticity controls (Political constraints).

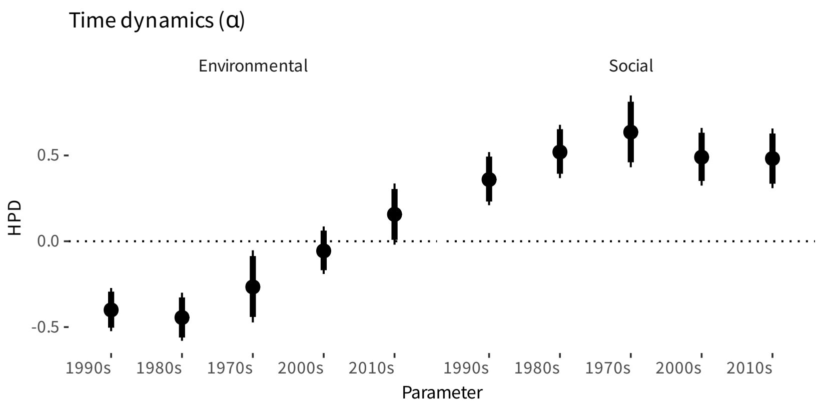


Figure 6.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-320.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

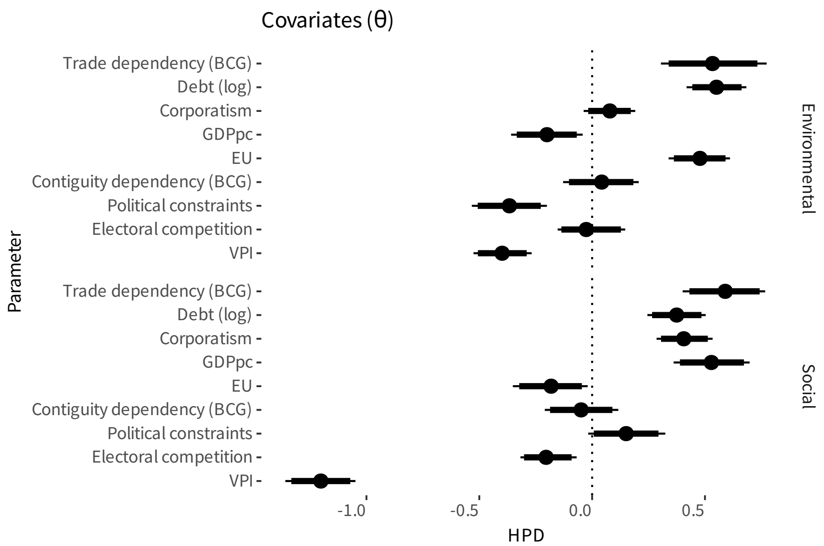


Figure 6.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

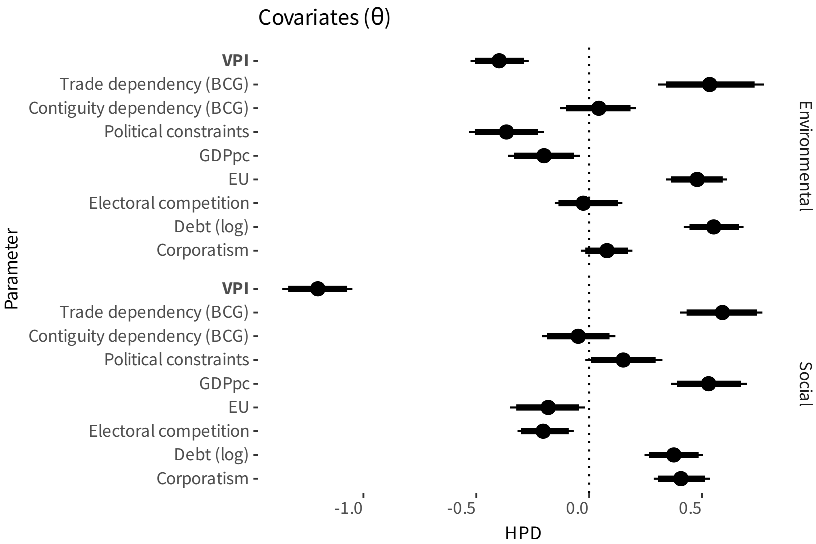


Figure 6.5: Covariates.

6.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5851
Social	0.5495

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

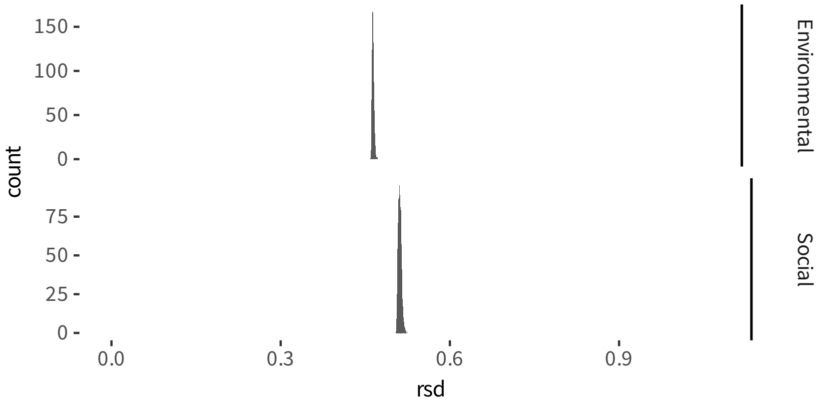


Figure 6.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Capped learning (instruments). Coefficient point estimates (median of the posterior d


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-320}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a10.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Debt (log)	0.55	(0.066)	[0.42 : 0.68]
Trade dependency (BCG)	0.53	(0.119)	[0.3 : 0.77]
EU	0.48	(0.069)	[0.34 : 0.61]
VPI	-0.40	(0.066)	[-0.53 : -0.27]
Political constraints	-0.37	(0.085)	[-0.53 : -0.2]
GDPpc	-0.20	(0.08)	[-0.36 : -0.042]
Corporatism	0.08	(0.058)	[-0.038 : 0.19]
Contiguity dependency (BCG)	0.04	(0.086)	[-0.13 : 0.21]
Electoral competition	-0.03	(0.083)	[-0.15 : 0.15]
** Goodness of fit (R2)	0.59	(0.0015)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.20	(0.08)	[-1.4 : -1]
Trade dependency (BCG)	0.59	(0.095)	[0.4 : 0.77]
GDPpc	0.53	(0.086)	[0.36 : 0.7]
Corporatism	0.41	(0.064)	[0.29 : 0.53]
Debt (log)	0.37	(0.066)	[0.24 : 0.5]
Electoral competition	-0.20	(0.064)	[-0.32 : -0.069]
EU	-0.18	(0.085)	[-0.35 : -0.02]
Political constraints	0.15	(0.088)	[-0.017 : 0.32]
Contiguity dependency (BCG)	-0.05	(0.084)	[-0.21 : 0.12]
** Goodness of fit (R2)	0.55	(0.00243)	[0.54 : 0.55]

Table 6.1: Model parameters. Capped learning (instruments). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

7
Explain Portfolio size over implementation capacity: Lag 5 years

m-321

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1981, zoo::rollmean(VPI, k = 5, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
```

```
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
```

```
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
```

```
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
##  $ Y         : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
##  $ nC        : int 69
##  $ nS        : int 2
##  $ nY        : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV        : int 6
```

```
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
```

```
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste(”models/model-”, M, ”.bug”, sep = ””),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = ”glm”,
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste(”samples-”, M, ”.RData”, sep = ””))

load(file = paste(”samples-”, M, ”.RData”, sep = ””))

ggmcmc(ggs(s, family = ”theta|rho|pi|lambda|alpha”, sort = FALSE),
           file = paste0(”ggmcmc-all-”, M, ”.pdf”),
           plot = c(”traceplot”, ”crosscorrelation”,
                    ”running”, ”caterpillar”))

L.rho <- plab(”rho”, list(Sector = sector.label))
S.rho <- ggs(s, family = ”rho”, par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab(”lambda”, list(Variable = c(”(Intercept)”, dimnames(X)[[3]][5]),
                                 Sector = sector.label))
S.lambda <- ggs(s, family = ”^lambda”, par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0(”ci-lambda-321.RData”))


S.lambda %>%
  filter(Variable != ”(Intercept)”) %>%
ggs_caterpillar(label = ”Sector”)


L.alpha <- plab(”alpha”, list(Sector = sector.label,
                               Decade = decade.label))
S.alpha <- ggs(s, family = ”^alpha”, par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0(”ci-alpha-321.RData”))

ggs_caterpillar(S.alpha, label = ”Decade”) +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste(”Time dynamics (”, alpha, ”)”, sep = ””)))
```

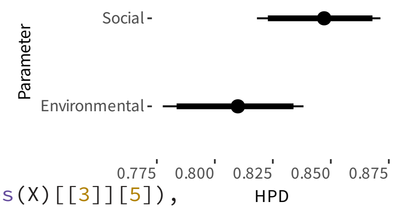


Figure 7.1: Auto-regressive components.

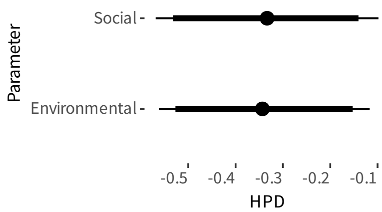


Figure 7.2: Heteroskedasticity controls (Political constraints).

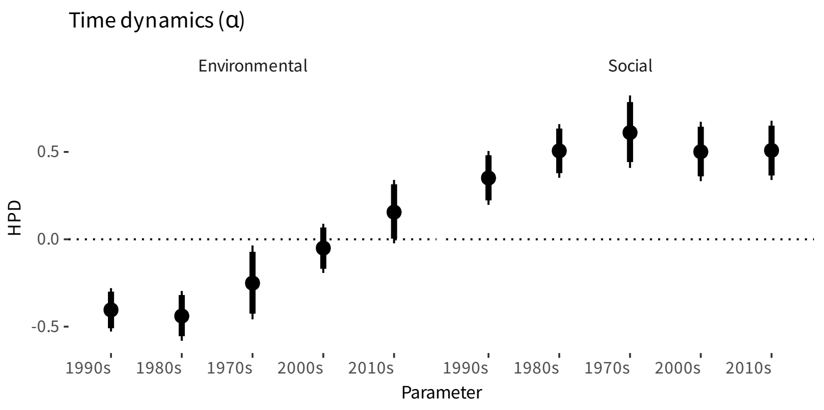


Figure 7.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-321.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

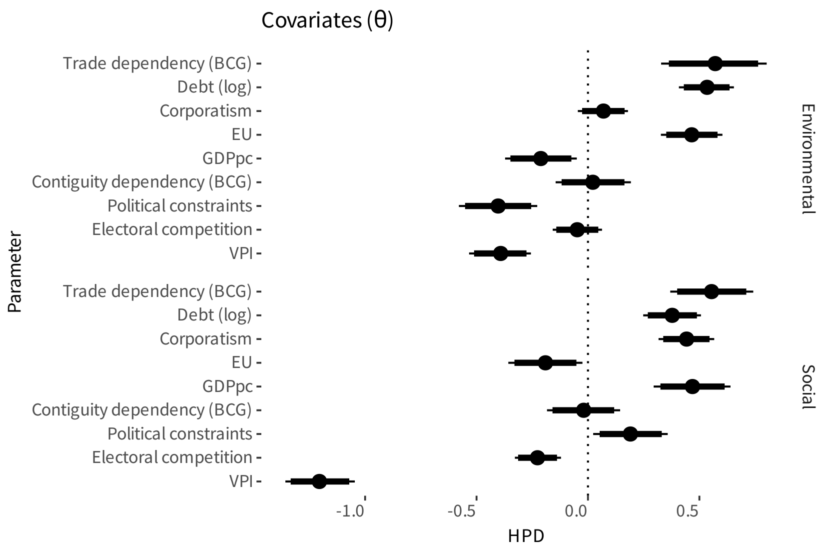


Figure 7.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

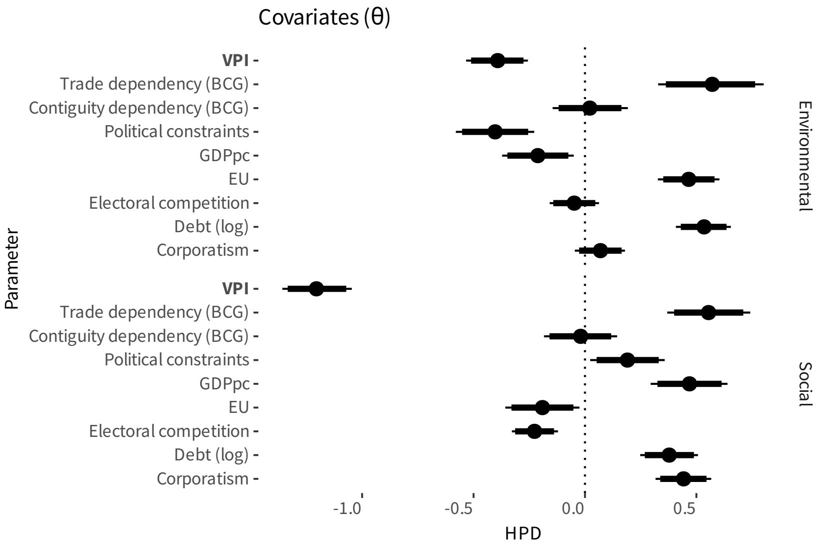


Figure 7.5: Covariates.

7.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5857
Social	0.5531

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

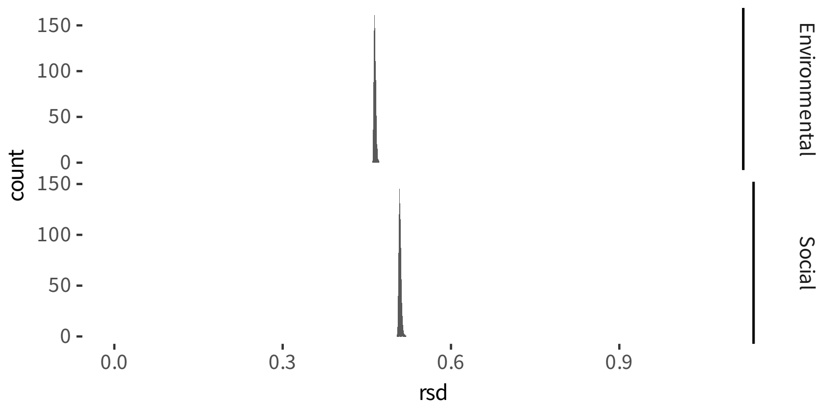


Figure 7.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Lag 5 years. Coefficient point estimates (median of the posterior distribution), SD r


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-321}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a11.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.57	(0.122)	[0.33 : 0.8]
Debt (log)	0.53	(0.062)	[0.41 : 0.65]
EU	0.47	(0.07)	[0.33 : 0.6]
Political constraints	-0.40	(0.09)	[-0.58 : -0.23]
VPI	-0.39	(0.072)	[-0.53 : -0.26]
GDPpc	-0.21	(0.083)	[-0.37 : -0.05]
Corporatism	0.07	(0.058)	[-0.046 : 0.18]
Electoral competition	-0.05	(0.057)	[-0.16 : 0.063]
Contiguity dependency (BCG)	0.02	(0.085)	[-0.15 : 0.19]
** Goodness of fit (R2)	0.59	(0.00153)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.21	(0.079)	[-1.4 : -1]
Trade dependency (BCG)	0.55	(0.094)	[0.37 : 0.74]
GDPpc	0.47	(0.087)	[0.29 : 0.64]
Corporatism	0.44	(0.064)	[0.32 : 0.57]
Debt (log)	0.38	(0.067)	[0.25 : 0.51]
Electoral competition	-0.23	(0.053)	[-0.33 : -0.12]
EU	-0.19	(0.084)	[-0.36 : -0.025]
Political constraints	0.19	(0.086)	[0.023 : 0.36]
Contiguity dependency (BCG)	-0.02	(0.085)	[-0.18 : 0.14]
** Goodness of fit (R2)	0.55	(0.0018)	[0.55 : 0.56]

Table 7.1: Model parameters. Lag 5 years. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

8

Explain Portfolio size over implementation capacity: Lag 7 years

m-322

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1983, zoo::rollmean(VPI, k = 7, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
```

```
        `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                              lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                            lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
```

```
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
## $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
## $ nC       : int 69
## $ nS       : int 2
## $ nY       : int 43
## $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
## $ nDecades : int 5
## $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
```

```
## $ nV : int 6
## $ XS : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
## $ nVS : int 3
## $ b0 : num [1:9] 0 0 0 0 0 0 0 0 0
## $ B0 : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
```

```
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-322.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-322.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

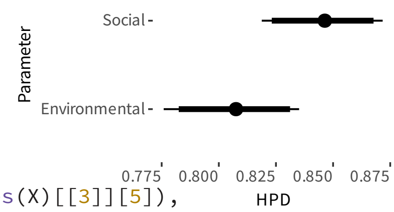


Figure 8.1: Auto-regressive components.

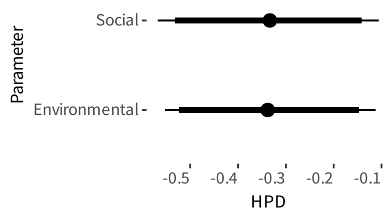


Figure 8.2: Heteroskedasticity controls (Political constraints).

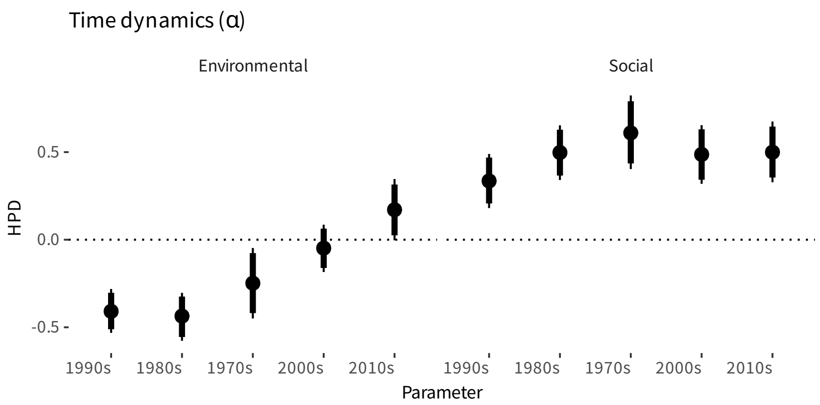


Figure 8.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-322.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

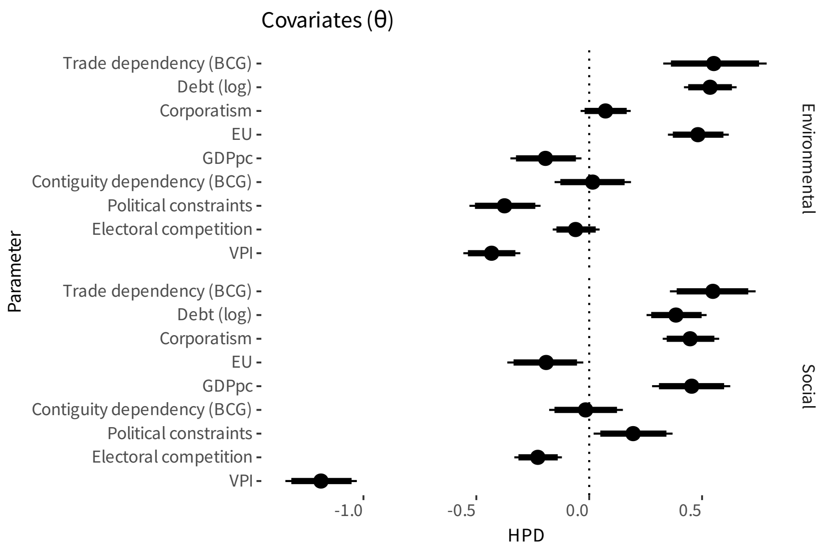


Figure 8.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

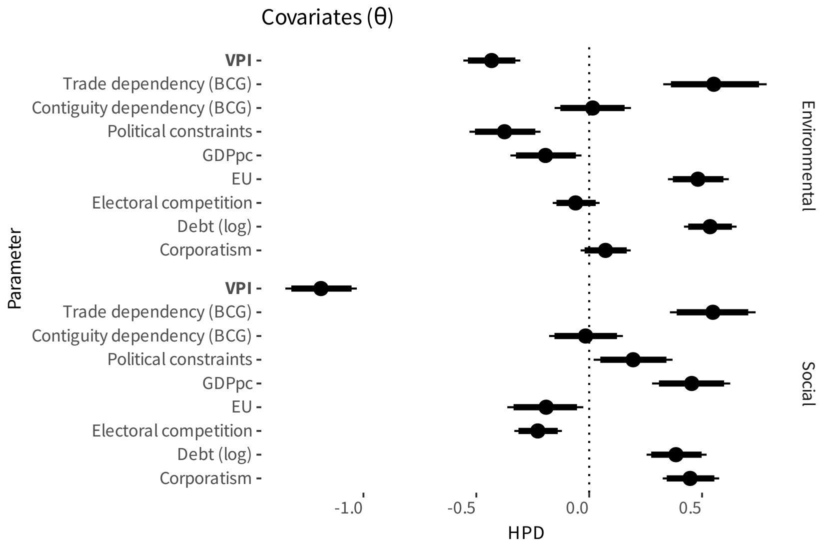


Figure 8.5: Covariates.

8.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5872
Social	0.5519

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

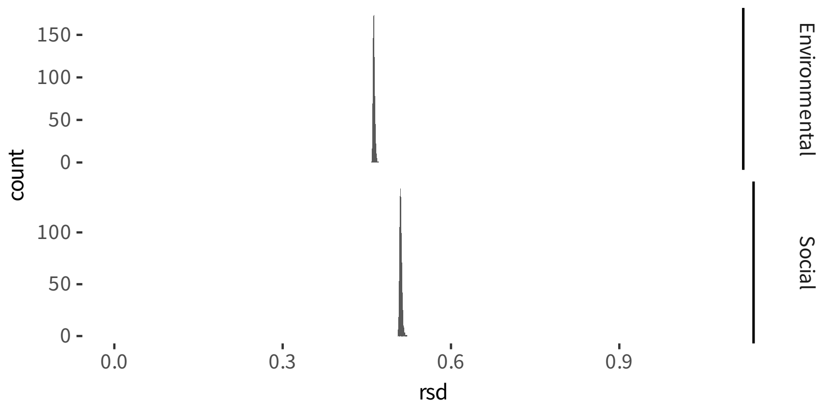


Figure 8.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Lag 7 years. Coefficient point estimates (median of the posterior distribution), SD r


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-322}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a12.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.55	(0.118)	[0.33 : 0.79]
Debt (log)	0.54	(0.059)	[0.42 : 0.65]
EU	0.48	(0.068)	[0.35 : 0.62]
VPI	-0.43	(0.064)	[-0.56 : -0.31]
Political constraints	-0.38	(0.08)	[-0.53 : -0.22]
GDPpc	-0.19	(0.08)	[-0.35 : -0.034]
Corporatism	0.07	(0.057)	[-0.038 : 0.18]
Electoral competition	-0.06	(0.053)	[-0.16 : 0.046]
Contiguity dependency (BCG)	0.02	(0.086)	[-0.15 : 0.19]
** Goodness of fit (R2)	0.59	(0.0014)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.19	(0.081)	[-1.3 : -1]
Trade dependency (BCG)	0.55	(0.096)	[0.36 : 0.74]
GDPpc	0.45	(0.089)	[0.28 : 0.62]
Corporatism	0.45	(0.065)	[0.33 : 0.58]
Debt (log)	0.38	(0.067)	[0.25 : 0.52]
Electoral competition	-0.23	(0.053)	[-0.33 : -0.12]
Political constraints	0.19	(0.089)	[0.019 : 0.37]
EU	-0.19	(0.085)	[-0.36 : -0.026]
Contiguity dependency (BCG)	-0.02	(0.084)	[-0.18 : 0.15]
** Goodness of fit (R2)	0.55	(0.00177)	[0.55 : 0.55]

Table 8.1: Model parameters. Lag 7 years. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

9

Explain Portfolio size over implementation capacity: No smoothed lag, but plain lag

m-323

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  mutate(VPI = ifelse(Year >= 1978, lag(VPI, n = 3), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
```

```
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
```

```
        `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
```

```
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
## $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
## $ nC       : int 69
## $ nS       : int 2
## $ nY       : int 43
## $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
## $ nDecades : int 5
## $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
```

```
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
```

```
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)
```

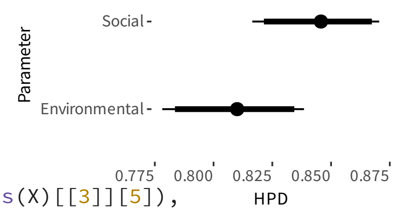


Figure 9.1: Auto-regressive components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-323.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")
```

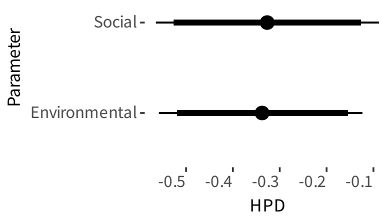


Figure 9.2: Heteroskedasticity controls (Political constraints).

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-323.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

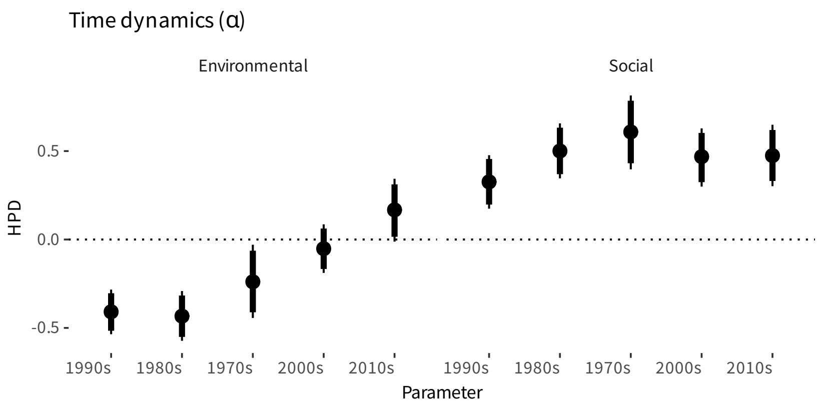


Figure 9.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-323.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

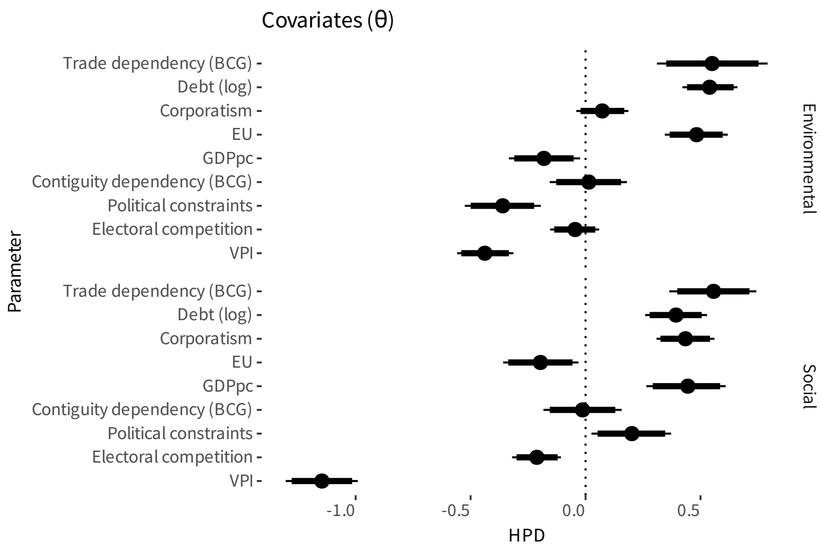


Figure 9.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

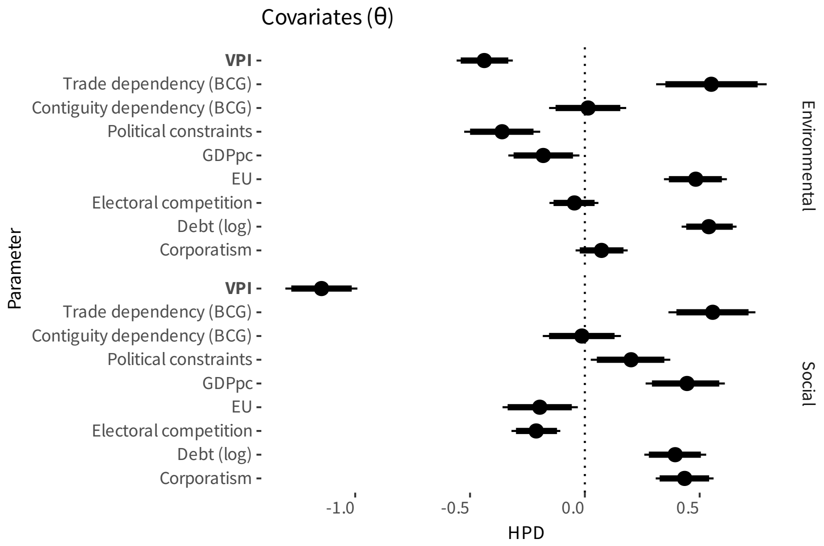


Figure 9.5: Covariates.

9.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                              Country = country.label,
                              Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5877
Social	0.5490

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

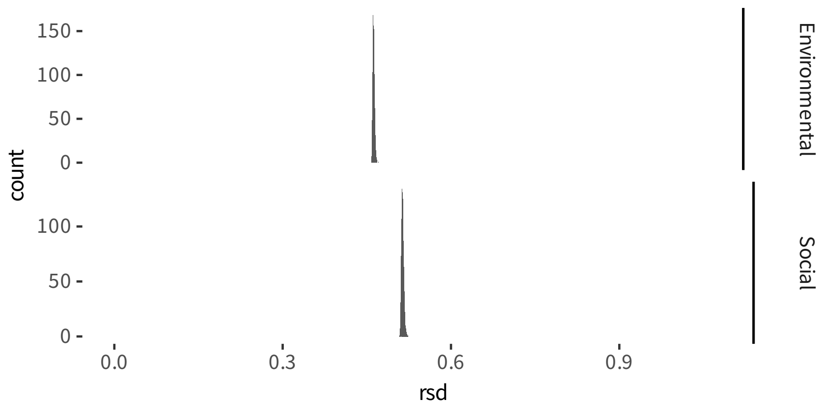


Figure 9.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. No smoothed lag, but plain lag. Coefficient point estimates (median of the posterior


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-323}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a13.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.55	(0.123)	[0.31 : 0.79]
Debt (log)	0.54	(0.061)	[0.42 : 0.66]
EU	0.48	(0.07)	[0.34 : 0.62]
VPI	-0.44	(0.063)	[-0.56 : -0.31]
Political constraints	-0.36	(0.084)	[-0.53 : -0.19]
GDPpc	-0.18	(0.079)	[-0.33 : -0.024]
Corporatism	0.07	(0.057)	[-0.04 : 0.19]
Electoral competition	-0.05	(0.054)	[-0.15 : 0.059]
Contiguity dependency (BCG)	0.01	(0.086)	[-0.16 : 0.18]
** Goodness of fit (R2)	0.59	(0.00145)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.15	(0.08)	[-1.3 : -0.99]
Trade dependency (BCG)	0.56	(0.096)	[0.36 : 0.74]
GDPpc	0.44	(0.089)	[0.26 : 0.61]
Corporatism	0.43	(0.065)	[0.31 : 0.56]
Debt (log)	0.39	(0.068)	[0.26 : 0.53]
Electoral competition	-0.21	(0.054)	[-0.32 : -0.11]
Political constraints	0.20	(0.088)	[0.026 : 0.37]
EU	-0.20	(0.084)	[-0.36 : -0.031]
Contiguity dependency (BCG)	-0.01	(0.086)	[-0.18 : 0.16]
** Goodness of fit (R2)	0.55	(0.00179)	[0.55 : 0.55]

Table 9.1: Model parameters. No smoothed lag, but plain lag. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

10
Explain Portfolio size over implementation capacity: Burden as subtraction

m-324

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Plain gap") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Plain gap".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
```

```
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
```

```
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
```

```
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
## $ Y          : num [1:2, 1:69, 1:43] -4.85 -2.95 NA -1.5 -2.88 ...
## $ nC         : int 69
## $ nS         : int 2
## $ nY         : int 43
## $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
## $ nDecades : int 5
## $ X          : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
## $ nV         : int 6
```

```
## $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.8 0.957 0.201 0.147 -0.656 ...
## $ nVS      : int 3
## $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
## $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
```

```
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste(”models/model-”, M, ”.bug”, sep = ””),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = ”glm”,
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste(”samples-”, M, ”.RData”, sep = ””))

load(file = paste(”samples-”, M, ”.RData”, sep = ””))

ggmcmc(ggs(s, family = ”theta|rho|pi|lambda|alpha”, sort = FALSE),
           file = paste0(”ggmcmc-all-”, M, ”.pdf”),
           plot = c(”traceplot”, ”crosscorrelation”,
                    ”running”, ”caterpillar”))

L.rho <- plab(”rho”, list(Sector = sector.label))
S.rho <- ggs(s, family = ”rho”, par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab(”lambda”, list(Variable = c(”(Intercept)”, dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = ”^lambda”, par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0(”ci-lambda-324.RData”))


S.lambda %>%
  filter(Variable != ”(Intercept)”) %>%
ggs_caterpillar(label = ”Sector”)


L.alpha <- plab(”alpha”, list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = ”^alpha”, par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0(”ci-alpha-324.RData”))

ggs_caterpillar(S.alpha, label = ”Decade”) +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste(”Time dynamics (”, alpha, ”)”, sep = ””)))
```

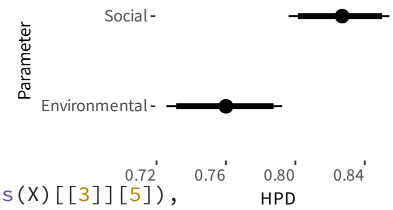


Figure 10.1: Auto-regressive components.

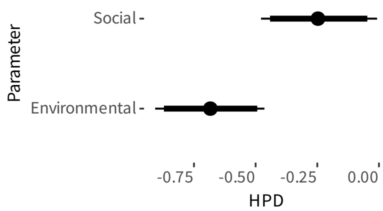


Figure 10.2: Heteroskedasticity controls (Political constraints).

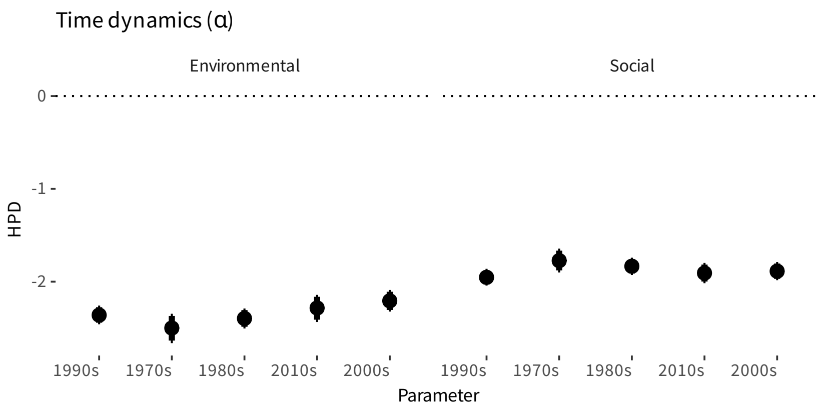


Figure 10.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-324.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

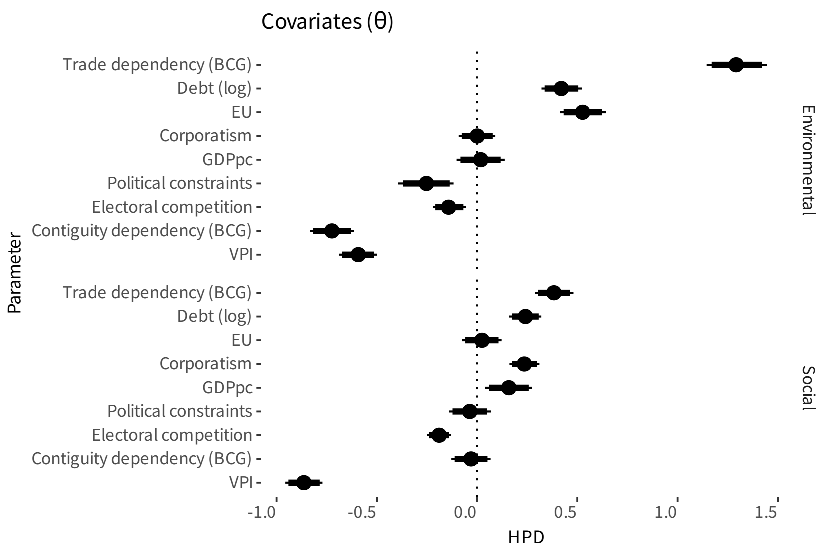


Figure 10.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

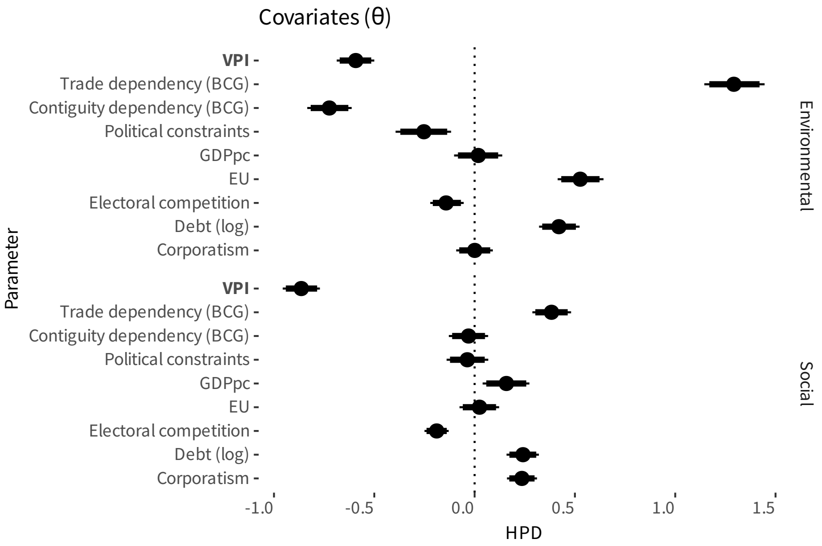


Figure 10.5: Covariates.

10.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.6326
Social	0.5797

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

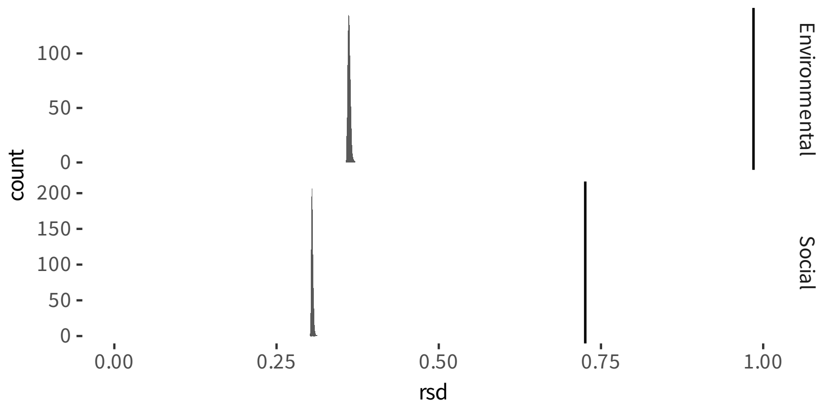


Figure 10.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Burden as subtraction. Coefficient point estimates (median of the posterior distribut


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-324}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a14.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	1.29	(0.076)	[1.1 : 1.4]
Contiguity dependency (BCG)	-0.72	(0.057)	[-0.83 : -0.61]
VPI	-0.59	(0.048)	[-0.69 : -0.5]
EU	0.53	(0.058)	[0.41 : 0.64]
Debt (log)	0.42	(0.051)	[0.32 : 0.52]
Political constraints	-0.25	(0.071)	[-0.39 : -0.12]
Electoral competition	-0.14	(0.042)	[-0.22 : -0.054]
GDPpc	0.02	(0.061)	[-0.1 : 0.14]
Corporatism	0.00	(0.047)	[-0.092 : 0.091]
** Goodness of fit (R2)	0.63	(0.00197)	[0.63 : 0.64]
y = Burden-Capacity gap (Social, N=903)			
VPI	-0.86	(0.047)	[-0.96 : -0.77]
Trade dependency (BCG)	0.38	(0.049)	[0.29 : 0.48]
Debt (log)	0.24	(0.041)	[0.16 : 0.32]
Corporatism	0.24	(0.038)	[0.16 : 0.31]
Electoral competition	-0.19	(0.031)	[-0.25 : -0.13]
GDPpc	0.16	(0.06)	[0.04 : 0.27]
Political constraints	-0.04	(0.053)	[-0.14 : 0.068]
Contiguity dependency (BCG)	-0.03	(0.05)	[-0.13 : 0.067]
EU	0.02	(0.05)	[-0.075 : 0.12]
** Goodness of fit (R2)	0.58	(0.00185)	[0.58 : 0.58]

Table 10.1: Model parameters. Burden as subtraction. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

11
Explain Portfolio size over implementation capacity: With state capacity

m-325

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is “Regular : Only sample countries : Not bounded : Original ratio”.

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                  `Top-down (VPI)` = seq(0, 6, by = 1/3),
                  `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                  `Top-down (VPI)` = 0,
                  `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
```

```
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

# State capacity
load("data-state_capacity.RData") # loads sc

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
```

```r
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
              select(Origin, Destination, p.contiguous),
              by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
              ungroup() %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
```

```
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism) %>%
  left_join(sc)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `State capacity`,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)
```

```
XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
```

```
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
#par <- c(par, "Theta", "sigma_theta")
#par <- c(par, "nu")
#par <- c(par, "sigma")
par <- c(par, "Sigma")
par <- c(par, "rho")
#par <- c(par, "pi")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
```

```
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:7] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV       : int 7
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS      : int 3
##  $ b0       : num [1:10] 0 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:10, 1:10] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
```

```
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }

t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)

L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-325.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)
```

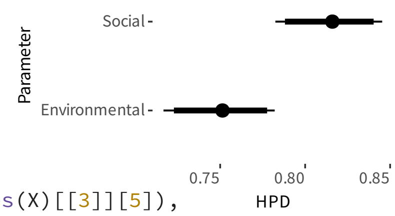


Figure 11.1: Auto-regressive components.

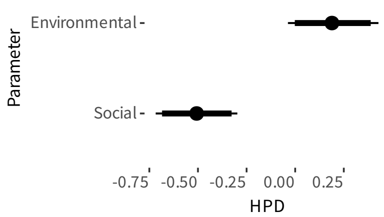


Figure 11.2: Heteroskedasticity controls (Political constraints).

```
ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-325.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

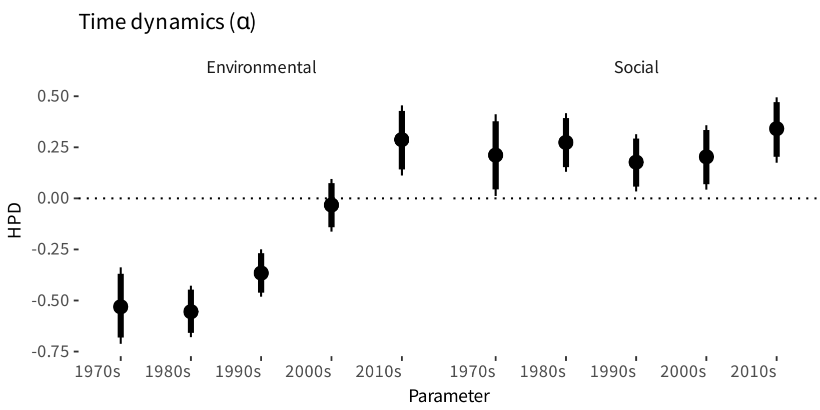


Figure 11.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                               Covariate = c(variable.label,
                                             variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-325.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

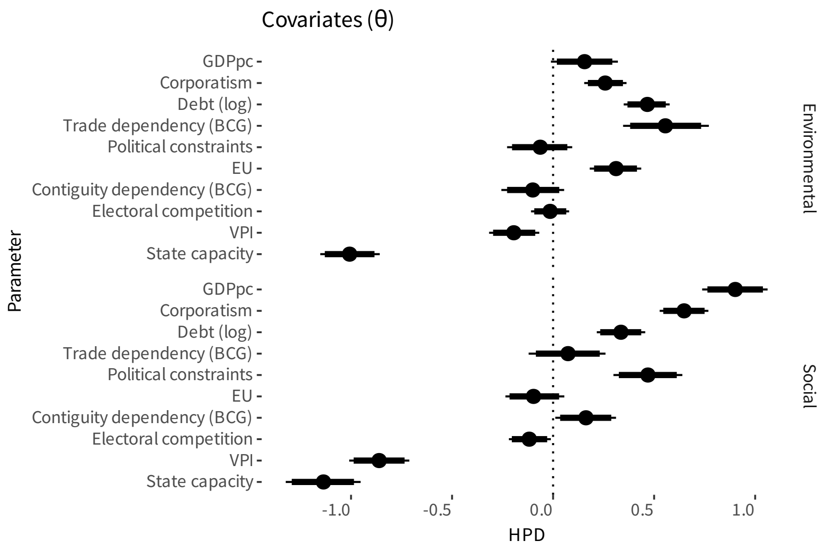


Figure 11.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

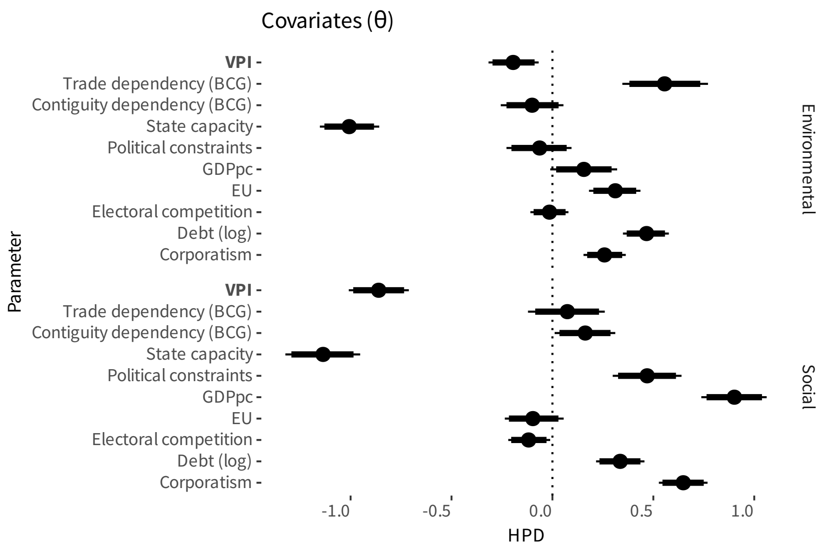


Figure 11.5: Covariates.

11.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
```

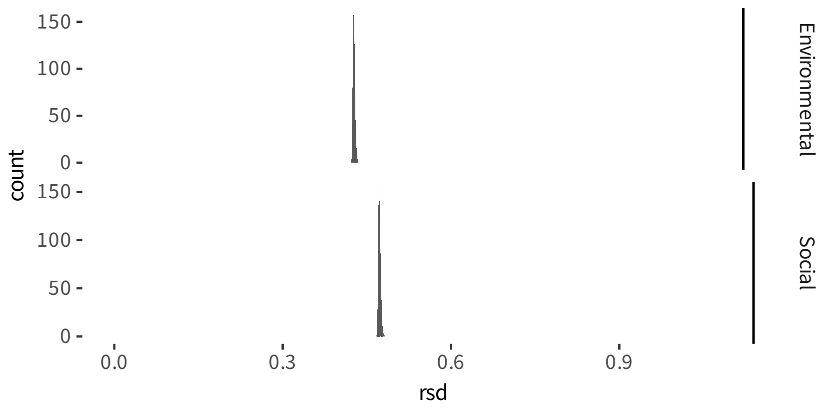


Figure 11.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.6191
Social	0.5850

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. With state capacity. Coefficient point estimates (median of the posterior distributio


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-325}", tc), label = NA,
```

```
    booktabs = TRUE,
    position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
State capacity	-1.01	(0.075)	[-1.2 : -0.86]
Trade dependency (BCG)	0.56	(0.108)	[0.35 : 0.77]
Debt (log)	0.47	(0.058)	[0.35 : 0.58]
EU	0.31	(0.065)	[0.18 : 0.44]
Corporatism	0.26	(0.053)	[0.15 : 0.36]
VPI	-0.19	(0.063)	[-0.32 : -0.068]
GDPpc	0.16	(0.084)	[-0.011 : 0.32]
Contiguity dependency (BCG)	-0.10	(0.078)	[-0.26 : 0.055]
Political constraints	-0.06	(0.082)	[-0.23 : 0.095]
Electoral competition	-0.01	(0.048)	[-0.11 : 0.079]
** Goodness of fit (R2)	0.62	(0.00161)	[0.62 : 0.62]
y = Burden-Capacity gap (Social, N=903)			
State capacity	-1.14	(0.094)	[-1.3 : -0.95]
GDPpc	0.90	(0.083)	[0.74 : 1.1]
VPI	-0.86	(0.076)	[-1 : -0.71]
Corporatism	0.65	(0.062)	[0.53 : 0.77]
Political constraints	0.47	(0.088)	[0.3 : 0.64]
Debt (log)	0.34	(0.061)	[0.22 : 0.46]
Contiguity dependency (BCG)	0.16	(0.076)	[0.011 : 0.31]
Electoral competition	-0.12	(0.053)	[-0.22 : -0.011]
EU	-0.10	(0.075)	[-0.24 : 0.056]
Trade dependency (BCG)	0.07	(0.097)	[-0.12 : 0.26]
** Goodness of fit (R2)	0.59	(0.00168)	[0.58 : 0.59]

Table 11.1: Model parameters. With state capacity. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a21.tex")
```

12 Explain Portfolio size over implementation capacity: Model with generosity instead of administrative spending

m-1312

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded Generosity : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded Generosity : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))
```

```
# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
```

```
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
              select(Origin, Destination, p.contiguous),
              by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
              ungroup() %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
```

```
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)
```

```
Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)
```

```
XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
```

```
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
#par <- c(par, "Theta", "sigma_theta")
#par <- c(par, "nu")
#par <- c(par, "sigma")
par <- c(par, "Sigma")
par <- c(par, "rho")
#par <- c(par, "pi")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.278 -0.415 0.332 -0.862 ...
##  $ nC       : int 69
```

```
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.0399 -0.07403 0.57357 -0.60633 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
```

```
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-1312.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
```

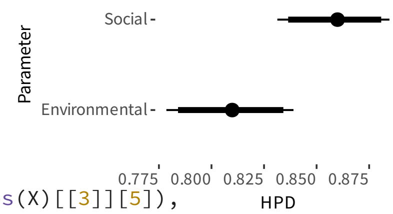


Figure 12.1: Auto-regressive components.

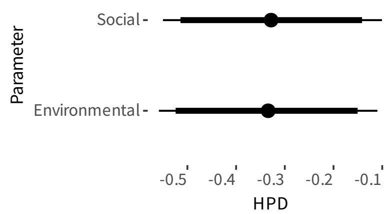


Figure 12.2: Heteroskedasticity controls (Political constraints).

```
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-1312.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

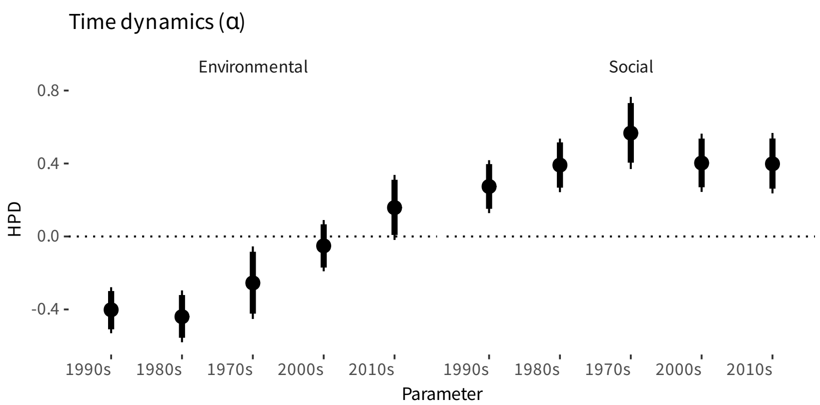


Figure 12.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-1312.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

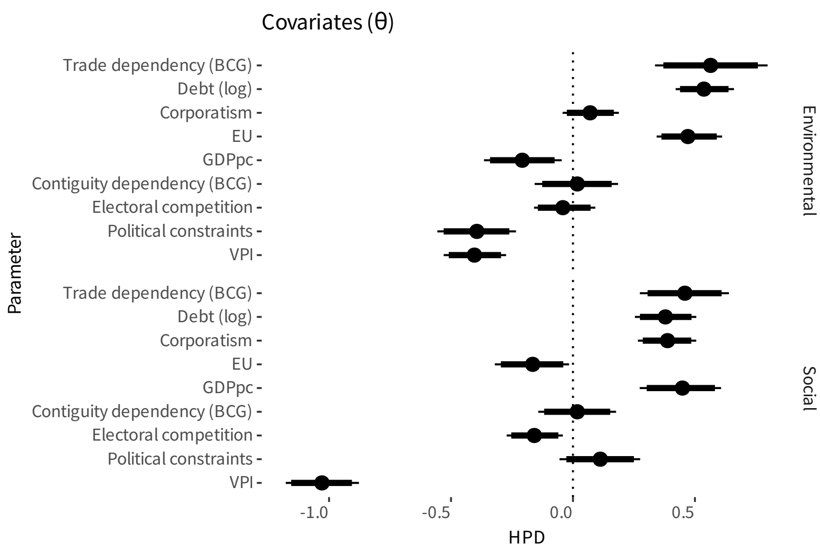


Figure 12.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
```

```
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

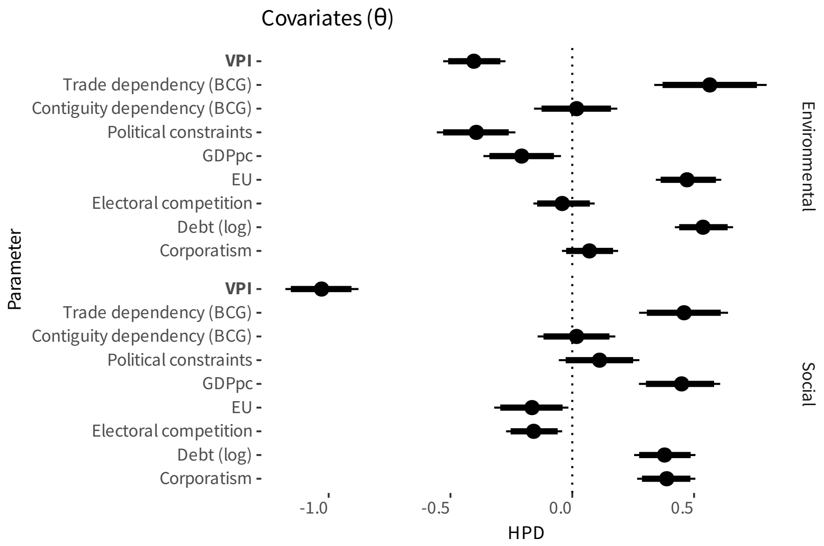


Figure 12.5: Covariates.

12.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
```

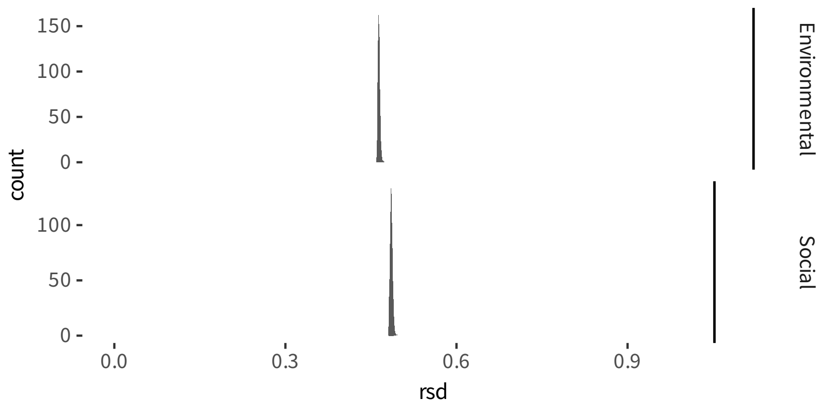


Figure 12.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5860
Social	0.5383

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`) %>%
  filter(Sector == "Social")

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Model with generosity instead of administrative spending. Only social sector. Coeffic


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-1312}", tc), label = NA,
```

```
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.03	(0.075)	[-1.2 : -0.88]
Trade dependency (BCG)	0.46	(0.092)	[0.27 : 0.64]
GDPpc	0.45	(0.085)	[0.27 : 0.61]
Corporatism	0.39	(0.06)	[0.27 : 0.51]
Debt (log)	0.38	(0.065)	[0.25 : 0.51]
EU	-0.17	(0.078)	[-0.32 : -0.016]
Electoral competition	-0.16	(0.059)	[-0.27 : -0.042]
Political constraints	0.11	(0.084)	[-0.055 : 0.28]
Contiguity dependency (BCG)	0.02	(0.081)	[-0.14 : 0.18]
** Goodness of fit (R2)	0.54	(0.00202)	[0.53 : 0.54]

Table 12.1: Model parameters. Model with generosity instead of administrative spending. Only social sector. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a15.tex")
```

13

Explain Portfolio size over implementation capacity: VPI with 2 values (low/high, and middle category as high)

m-314

```
load("data-implementation_deficit.RData")
```

```
####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition, middle value: high"
vpi <- vpi.es %>%
  mutate(value = ifelse(value == 2, 1, value)) %>%
  group_by(Sector, Country, Year, Dimension) %>%
  summarize(VPI = sum(value)) %>%
  ungroup()

vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                  `Top-down (VPI)` = seq(0, 6, by = 1/3),
                  `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                  `Top-down (VPI)` = 0,
```

```
                 `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                 VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))


# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                 `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                               max(pc$`Political constraints`, na.rm = TRUE),
                                               length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
```

```
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
```

```
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")
```

```
variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
```

```
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
## $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
## $ nC       : int 69
```

```
##  $ nS        : int 2
##  $ nY        : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV        : int 6
##  $ XS        : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS       : int 3
##  $ b0        : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0        : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                            lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
```

```
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)
```

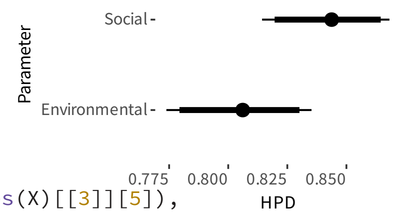


Figure 13.1: Auto-regressive components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-314.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")
```

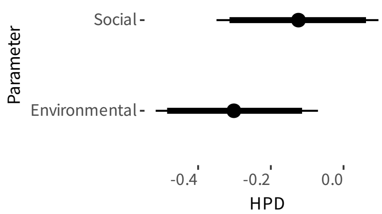


Figure 13.2: Heteroskedasticity controls (Political constraints).

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
```

```
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-314.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

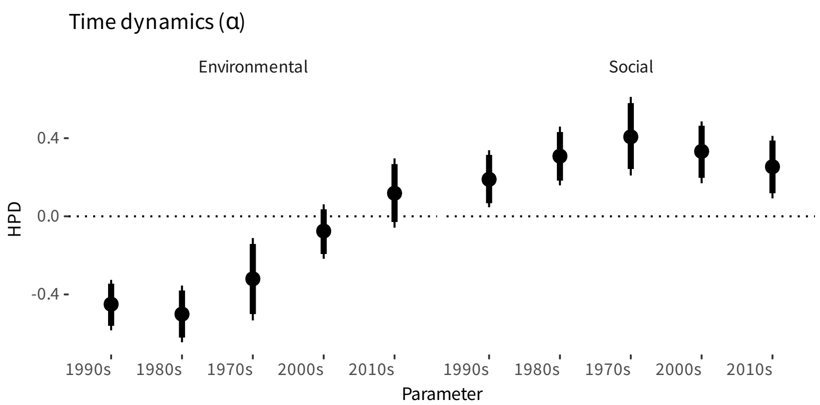


Figure 13.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-314.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

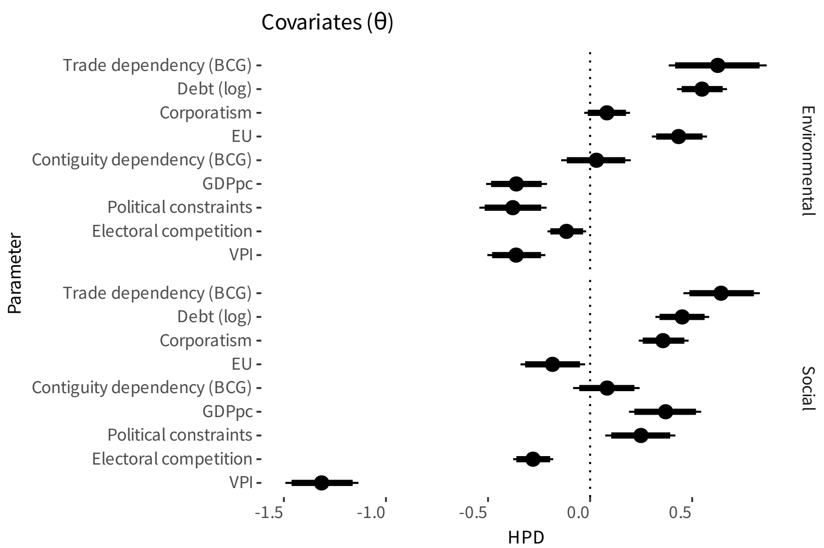


Figure 13.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
```

```
                                                variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

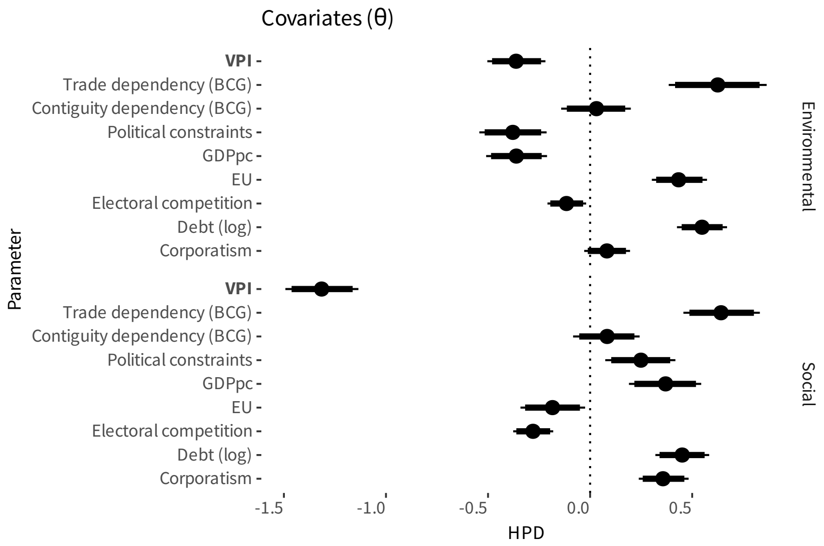


Figure 13.5: Covariates.

13.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
```

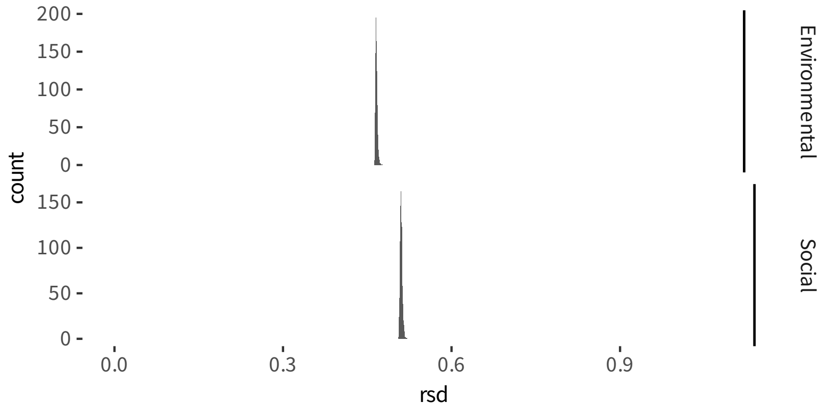


Figure 13.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.584
Social	0.552

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. VPI with 2 values (low/high, and middle category as high). Coefficient point estimate


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-314}", tc), label = NA,
      booktabs = TRUE,
```

```
    position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.63	(0.124)	[0.39 : 0.86]
Debt (log)	0.55	(0.062)	[0.43 : 0.67]
EU	0.43	(0.068)	[0.3 : 0.57]
Political constraints	-0.38	(0.084)	[-0.54 : -0.21]
VPI	-0.36	(0.073)	[-0.5 : -0.22]
GDPpc	-0.36	(0.075)	[-0.51 : -0.21]
Electoral competition	-0.12	(0.048)	[-0.21 : -0.019]
Corporatism	0.08	(0.057)	[-0.029 : 0.2]
Contiguity dependency (BCG)	0.03	(0.086)	[-0.14 : 0.2]
** Goodness of fit (R2)	0.58	(0.00139)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.32	(0.091)	[-1.5 : -1.1]
Trade dependency (BCG)	0.64	(0.096)	[0.46 : 0.83]
Debt (log)	0.45	(0.067)	[0.32 : 0.58]
GDPpc	0.37	(0.091)	[0.19 : 0.54]
Corporatism	0.36	(0.062)	[0.24 : 0.48]
Electoral competition	-0.28	(0.05)	[-0.38 : -0.18]
Political constraints	0.25	(0.088)	[0.074 : 0.42]
EU	-0.18	(0.081)	[-0.34 : -0.025]
Contiguity dependency (BCG)	0.08	(0.083)	[-0.083 : 0.24]
** Goodness of fit (R2)	0.55	(0.00163)	[0.55 : 0.55]

Table 13.1: Model parameters. VPI with 2 values (low/high, and middle category as high). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a16.tex")
```

14
Explain Portfolio size over implementation capacity: VPI with 2 values (low/high, and middle category as low)

m-1314

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition, middle value: low"
vpi <- vpi.es %>%
  mutate(value = ifelse(value == 1, 0, value)) %>%
  mutate(value = ifelse(value == 2, 1, value)) %>%
  group_by(Sector, Country, Year, Dimension) %>%
  summarize(VPI = sum(value)) %>%
  ungroup()

vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                  `Top-down (VPI)` = seq(0, 6, by = 1/3),
                  `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
```

```
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
```

```
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                      Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                      Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                      Country =
                        unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                      Year = 1976:2018,
                      `Contiguity dependency (BCG)` = NA,
                      `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                      Country =
                        unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                      Year = 1976:2018,
                      `Trade dependency (BCG)` = NA,
                      `Trade dependency (PS)` = NA))
```

```
#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)




Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")
```

```
variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
```

```
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)


## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
```

```
## $ nC       : int 69
## $ nS       : int 2
## $ nY       : int 43
## $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
## $ nDecades : int 5
## $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
## $ nV       : int 6
## $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
## $ nVS      : int 3
## $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
## $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
```

```
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)
```

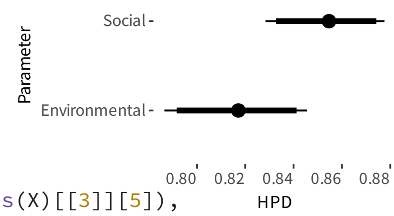


Figure 14.1: Auto-regressive components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-1314.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")
```

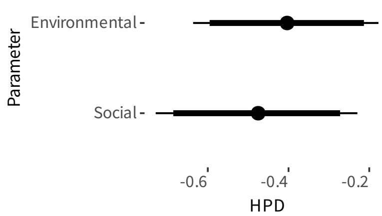


Figure 14.2: Heteroskedasticity controls (Political constraints).

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)
```

```
ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-1314.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

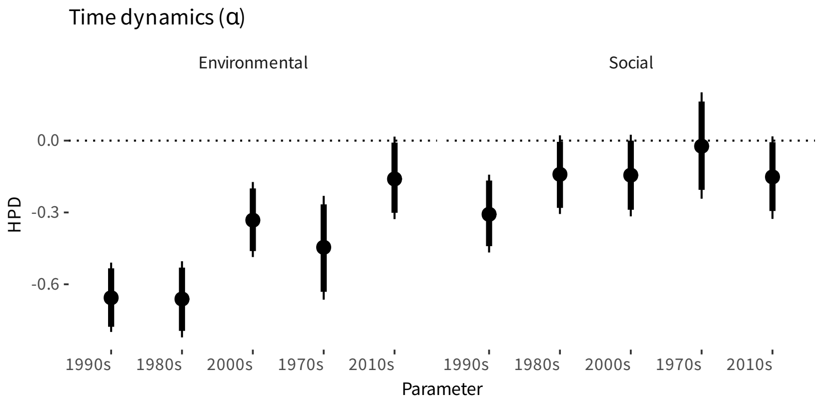


Figure 14.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-1314.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

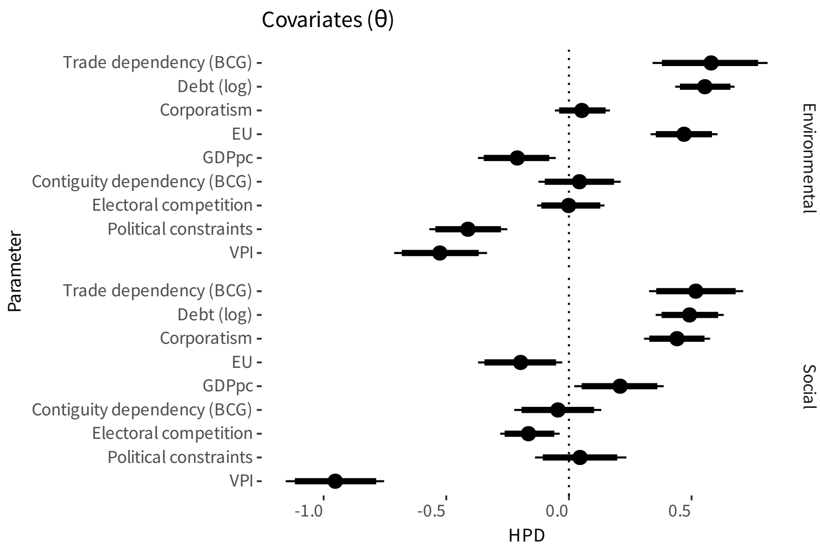


Figure 14.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

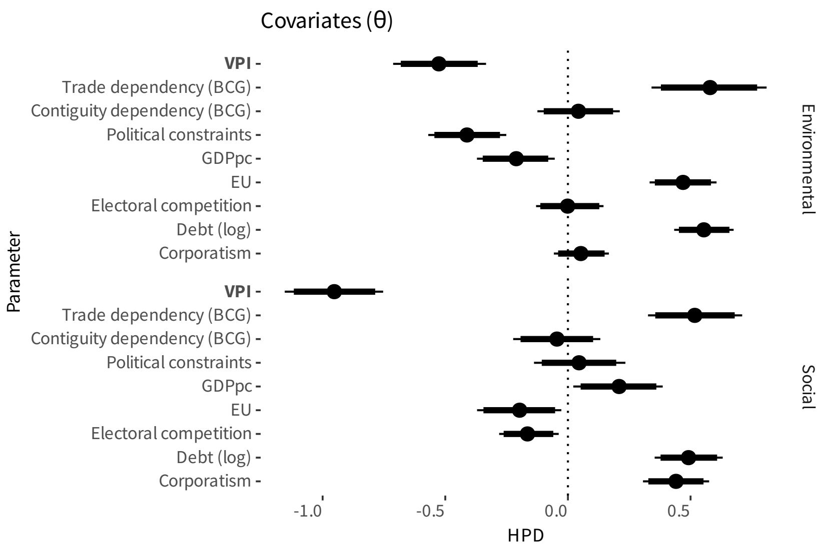


Figure 14.5: Covariates.

14.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
```

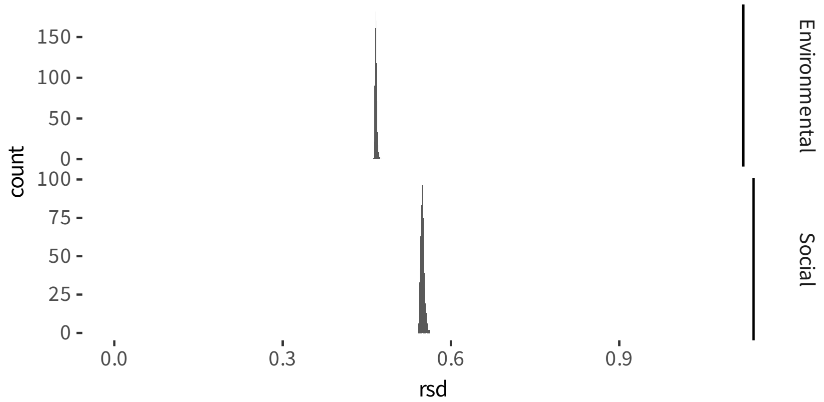


Figure 14.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5843
Social	0.5180

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. VPI with 2 values (low/high, and middle category as low). Coefficient point estimates


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-1314}", tc), label = NA,
```

```
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.58	(0.12)	[0.34 : 0.81]
Debt (log)	0.55	(0.062)	[0.43 : 0.68]
VPI	-0.53	(0.095)	[-0.71 : -0.33]
EU	0.47	(0.069)	[0.33 : 0.61]
Political constraints	-0.41	(0.081)	[-0.57 : -0.25]
GDPpc	-0.21	(0.081)	[-0.37 : -0.054]
Corporatism	0.05	(0.057)	[-0.057 : 0.17]
Contiguity dependency (BCG)	0.04	(0.085)	[-0.12 : 0.21]
Electoral competition	0.00	(0.074)	[-0.13 : 0.15]
** Goodness of fit (R2)	0.58	(0.00143)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-0.95	(0.103)	[-1.2 : -0.75]
Trade dependency (BCG)	0.52	(0.098)	[0.33 : 0.71]
Debt (log)	0.49	(0.07)	[0.35 : 0.63]
Corporatism	0.44	(0.068)	[0.31 : 0.58]
GDPpc	0.21	(0.094)	[0.022 : 0.39]
EU	-0.20	(0.089)	[-0.37 : -0.027]
Electoral competition	-0.16	(0.061)	[-0.28 : -0.038]
Political constraints	0.05	(0.094)	[-0.14 : 0.23]
Contiguity dependency (BCG)	-0.04	(0.09)	[-0.22 : 0.13]
** Goodness of fit (R2)	0.52	(0.00272)	[0.51 : 0.52]

Table 14.1: Model parameters. VPI with 2 values (low/high, and middle category as low). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a17.tex")
```

15
Explain Portfolio size over implementation capacity: Continuous learning (targets)

m-332

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "TwContinuous : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "TwContinuous : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                  `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                max(pc$`Political constraints`, na.rm = TRUE),
                                                length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
          select(Origin, Destination, p.contiguous),
          by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
          `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                      Country =
                        unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                      Year = 1976:2018,
                      `Contiguity dependency (BCG)` = NA,
                      `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
          ungroup() %>%
          select(Origin, Destination, Year, p.Exports),
          by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
        Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
          `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                      Country =
                        unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                      Year = 1976:2018,
                      `Trade dependency (BCG)` = NA,
                      `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
          group_by(Sector, Country, Year) %>%
          summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
```

```
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
```

```
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
## $ Y         : num [1:2, 1:69, 1:43] -1.6 -1.377 -0.528 0.599 -0.867 ...
## $ nC        : int 69
## $ nS        : int 2
## $ nY        : int 43
## $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
## $ nDecades : int 5
## $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
```

```
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00237 0.02689 -0.09345 0.62719 -0.62549 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                               lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                             lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
```

```
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }

t0 <- proc.time()
rj <- run.jags(model = paste(”models/model-”, M, ”.bug”, sep = ””),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = ”glm”,
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste(”samples-”, M, ”.RData”, sep = ””))

load(file = paste(”samples-”, M, ”.RData”, sep = ””))

ggmcmc(ggs(s, family = ”theta|rho|pi|lambda|alpha”, sort = FALSE),
           file = paste0(”ggmcmc-all-”, M, ”.pdf”),
           plot = c(”traceplot”, ”crosscorrelation”,
                    ”running”, ”caterpillar”))

L.rho <- plab(”rho”, list(Sector = sector.label))
S.rho <- ggs(s, family = ”rho”, par_labels = L.rho)
ggs_caterpillar(S.rho)
```

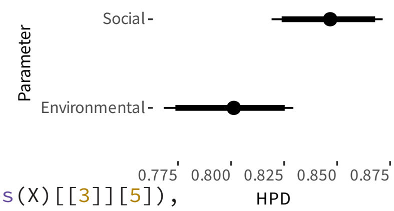


Figure 15.1: Auto-regressive components.

```
L.lambda <- plab(”lambda”, list(Variable = c(”(Intercept)”, dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = ”^lambda”, par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0(”ci-lambda-332.RData”))


S.lambda %>%
  filter(Variable != ”(Intercept)”) %>%
ggs_caterpillar(label = ”Sector”)
```

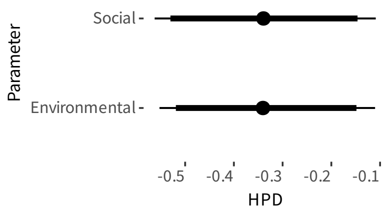


Figure 15.2: Heteroskedasticity controls (Political constraints).

```
L.alpha <- plab(”alpha”, list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = ”^alpha”, par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0(”ci-alpha-332.RData”))

ggs_caterpillar(S.alpha, label = ”Decade”) +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste(”Time dynamics (”, alpha, ”)”, sep = ””)))
```

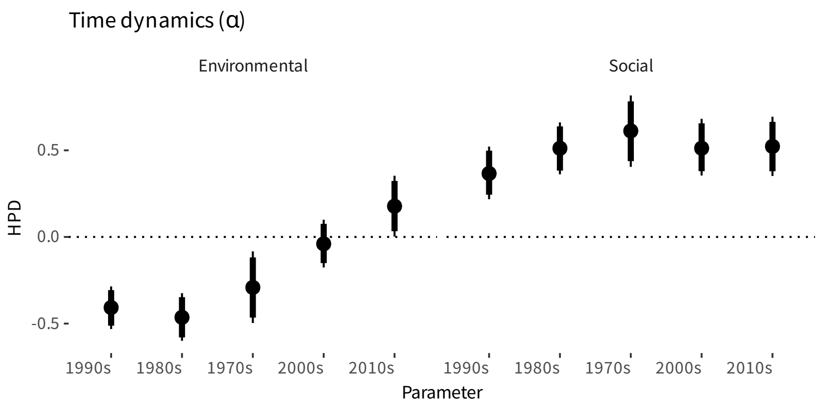


Figure 15.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-332.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

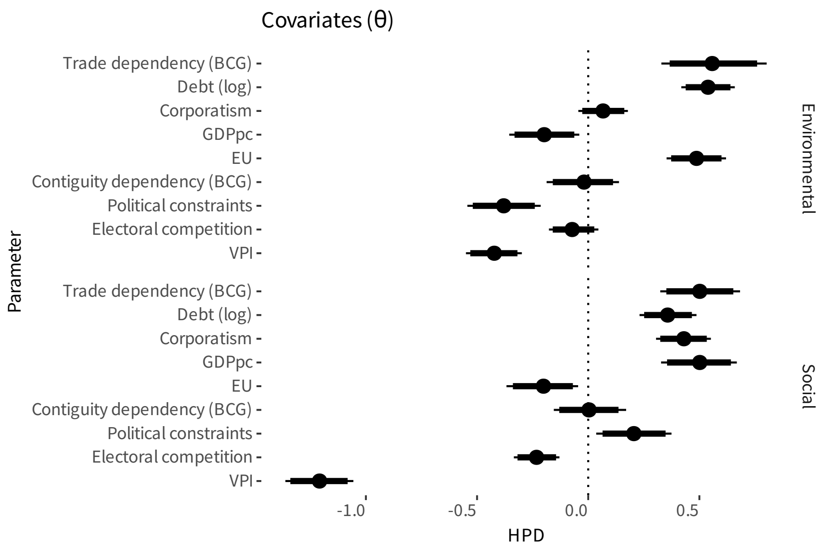


Figure 15.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

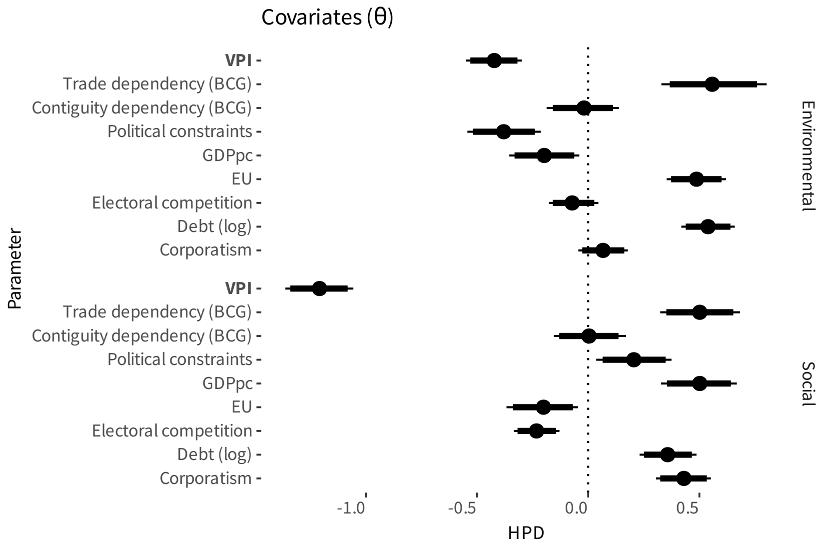


Figure 15.5: Covariates.

15.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5878
Social	0.5526

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

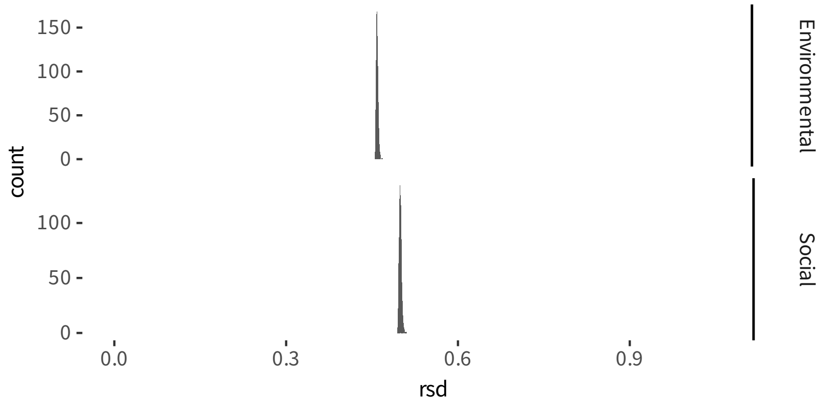


Figure 15.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Continuous learning (targets). Coefficient point estimates (median of the posterior d


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-332}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a18.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.56	(0.119)	[0.33 : 0.8]
Debt (log)	0.54	(0.061)	[0.42 : 0.66]
EU	0.49	(0.069)	[0.35 : 0.62]
VPI	-0.42	(0.065)	[-0.55 : -0.3]
Political constraints	-0.38	(0.085)	[-0.54 : -0.21]
GDPpc	-0.20	(0.081)	[-0.36 : -0.04]
Electoral competition	-0.07	(0.057)	[-0.18 : 0.046]
Corporatism	0.07	(0.057)	[-0.044 : 0.18]
Contiguity dependency (BCG)	-0.02	(0.082)	[-0.19 : 0.14]
** Goodness of fit (R2)	0.59	(0.00148)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.21	(0.079)	[-1.4 : -1.1]
GDPpc	0.50	(0.087)	[0.33 : 0.67]
Trade dependency (BCG)	0.50	(0.091)	[0.32 : 0.68]
Corporatism	0.43	(0.063)	[0.31 : 0.55]
Debt (log)	0.36	(0.065)	[0.23 : 0.49]
Electoral competition	-0.23	(0.052)	[-0.33 : -0.13]
Political constraints	0.21	(0.086)	[0.036 : 0.37]
EU	-0.20	(0.082)	[-0.37 : -0.045]
Contiguity dependency (BCG)	0.00	(0.082)	[-0.15 : 0.17]
** Goodness of fit (R2)	0.55	(0.00188)	[0.55 : 0.56]

Table 15.1: Model parameters. Continuous learning (targets). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

16 Explain Portfolio size over implementation capacity: Steep learning (targets)

m-333

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "TwSteep : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "TwSteep : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
```

```
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
```

```
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                              lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                            lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
```

```
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.947 -1.331 -0.973 0.562 -0.782 ...
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV       : int 6
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] -0.00459 0.03246 -0.13093 0.66895 -0.66305 ...
```

```
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
```

```
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                              Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-333.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                            Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-333.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  #facet_grid(Sector ~ .) +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

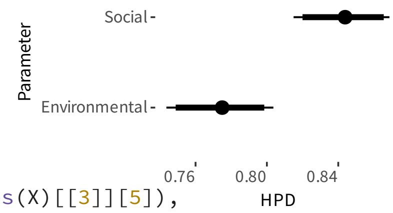


Figure 16.1: Auto-regressive components.

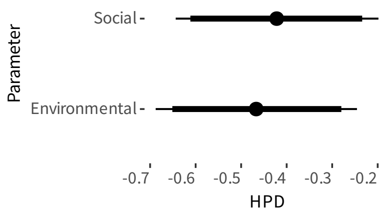


Figure 16.2: Heteroskedasticity controls (Political constraints).

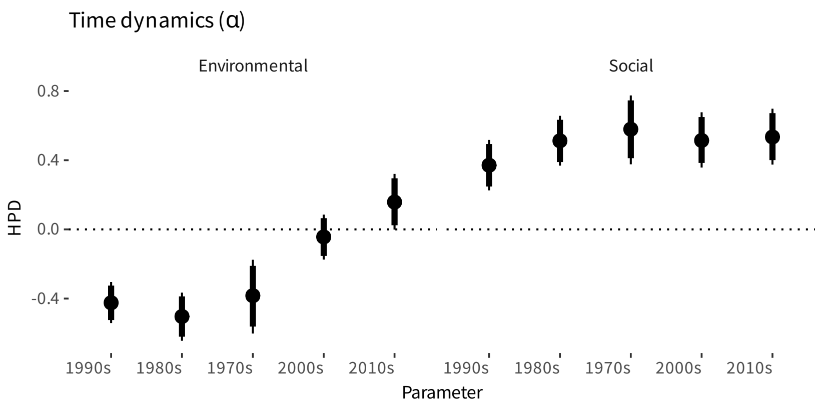


Figure 16.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-333.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

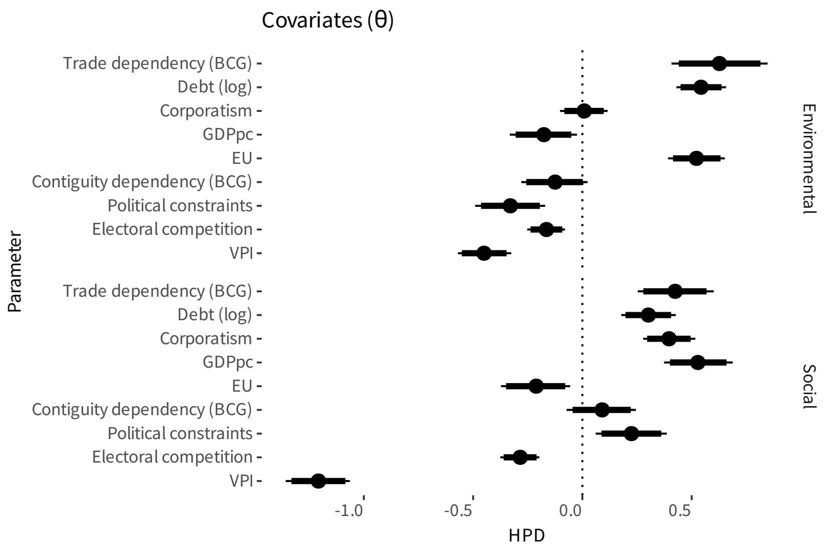


Figure 16.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

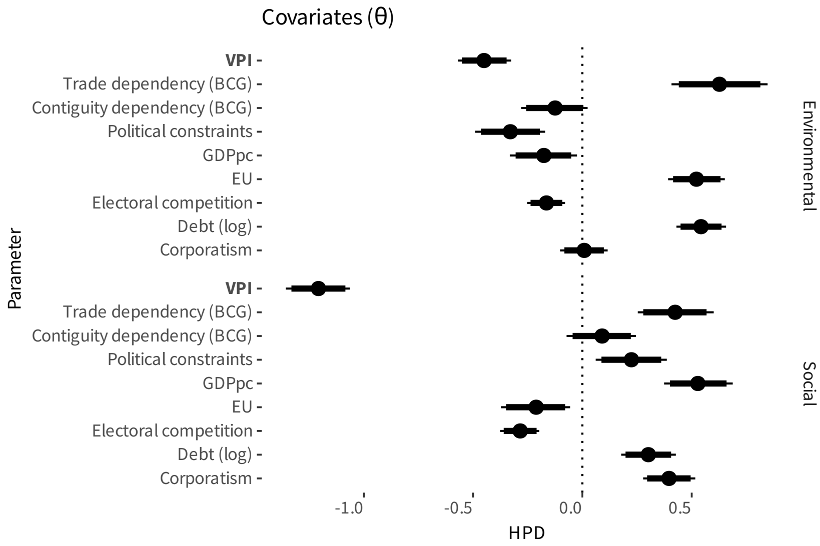


Figure 16.5: Covariates.

16.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5929
Social	0.5541

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

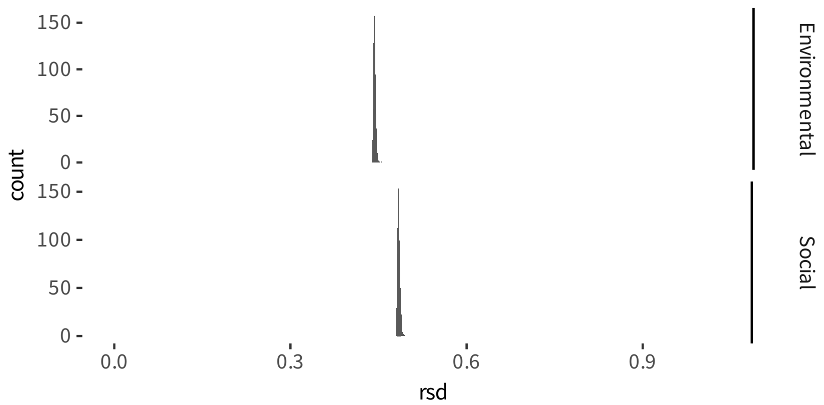


Figure 16.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Steep learning (targets). Coefficient point estimates (median of the posterior distri


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-333}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a19.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.63	(0.113)	[0.41 : 0.85]
Debt (log)	0.54	(0.058)	[0.43 : 0.66]
EU	0.52	(0.066)	[0.39 : 0.65]
VPI	-0.45	(0.062)	[-0.57 : -0.33]
Political constraints	-0.33	(0.081)	[-0.49 : -0.17]
GDPpc	-0.18	(0.077)	[-0.33 : -0.025]
Electoral competition	-0.16	(0.044)	[-0.25 : -0.079]
Contiguity dependency (BCG)	-0.12	(0.078)	[-0.28 : 0.024]
Corporatism	0.01	(0.055)	[-0.1 : 0.12]
** Goodness of fit (R2)	0.59	(0.00158)	[0.59 : 0.6]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.21	(0.075)	[-1.4 : -1.1]
GDPpc	0.53	(0.08)	[0.37 : 0.69]
Trade dependency (BCG)	0.42	(0.088)	[0.25 : 0.6]
Corporatism	0.40	(0.06)	[0.28 : 0.52]
Debt (log)	0.30	(0.064)	[0.18 : 0.43]
Electoral competition	-0.28	(0.046)	[-0.38 : -0.2]
Political constraints	0.22	(0.083)	[0.061 : 0.39]
EU	-0.21	(0.082)	[-0.37 : -0.056]
Contiguity dependency (BCG)	0.09	(0.081)	[-0.072 : 0.24]
** Goodness of fit (R2)	0.55	(0.00188)	[0.55 : 0.56]

Table 16.1: Model parameters. Steep learning (targets). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

17
Explain Portfolio size over implementation capacity: Capped learning (targets)

m-334

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "TwCapped : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "TwCapped : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
```

```
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%
```

```
full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
```

```
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
```

```
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
```

```
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
## $ Y         : num [1:2, 1:69, 1:43] -1.744 -1.375 -0.708 0.575 -0.792 ...
## $ nC        : int 69
## $ nS        : int 2
## $ nY        : int 43
## $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
## $ nDecades : int 5
## $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
## $ nV        : int 6
```

```
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] -0.000725 0.029377 -0.095097 0.677452 -0.628375 ...
##  $ nVS      : int 3
##  $ b0       : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])          }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
```

```
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }

t0 <- proc.time()
rj <- run.jags(model = paste(”models/model-”, M, ”.bug”, sep = ””),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = ”glm”,
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste(”samples-”, M, ”.RData”, sep = ””))

load(file = paste(”samples-”, M, ”.RData”, sep = ””))

ggmcmc(ggs(s, family = ”theta|rho|pi|lambda|alpha”, sort = FALSE),
           file = paste0(”ggmcmc-all-”, M, ”.pdf”),
           plot = c(”traceplot”, ”crosscorrelation”,
                    ”running”, ”caterpillar”))

L.rho <- plab(”rho”, list(Sector = sector.label))
S.rho <- ggs(s, family = ”rho”, par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab(”lambda”, list(Variable = c(”(Intercept)”, dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = ”^lambda”, par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0(”ci-lambda-334.RData”))


S.lambda %>%
  filter(Variable != ”(Intercept)”) %>%
ggs_caterpillar(label = ”Sector”)


L.alpha <- plab(”alpha”, list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = ”^alpha”, par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0(”ci-alpha-334.RData”))

ggs_caterpillar(S.alpha, label = ”Decade”) +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste(”Time dynamics (”, alpha, ”)”, sep = ””)))
```

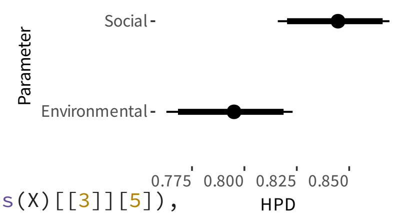


Figure 17.1: Auto-regressive components.

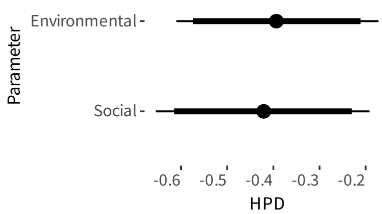


Figure 17.2: Heteroskedasticity controls (Political constraints).

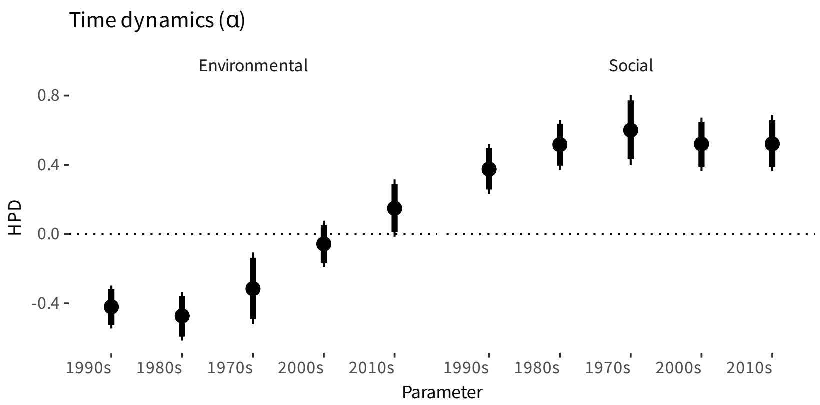


Figure 17.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-334.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

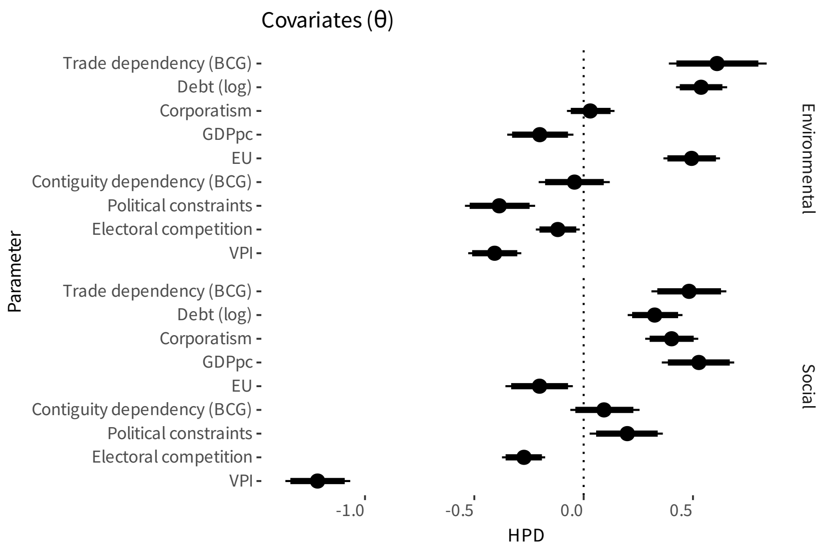


Figure 17.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

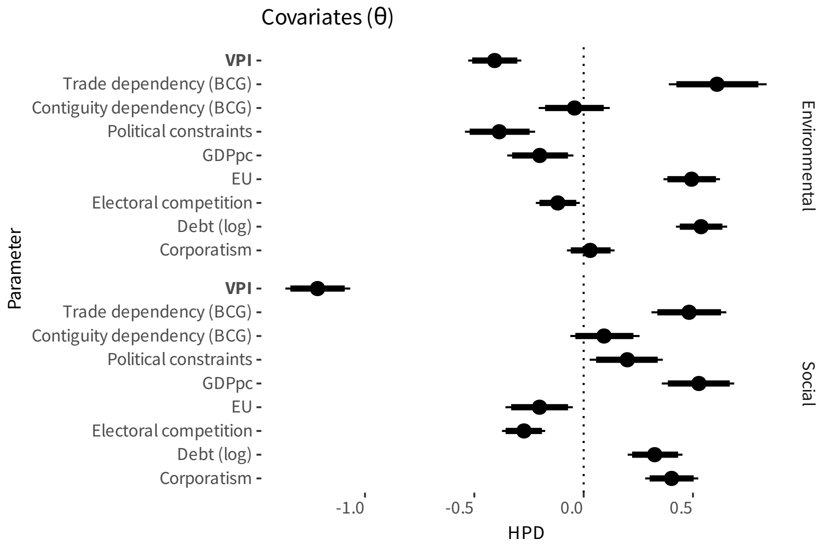


Figure 17.5: Covariates.

17.1 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5899
Social	0.5565

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

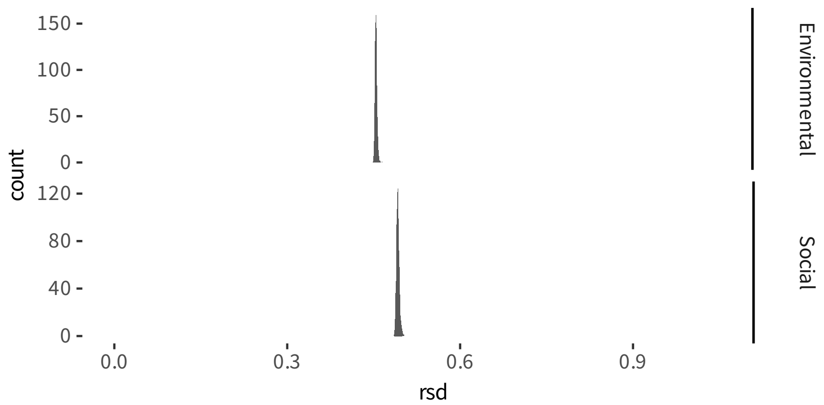


Figure 17.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Capped learning (targets). Coefficient point estimates (median of the posterior distr


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-334}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a20.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.61	(0.115)	[0.39 : 0.84]
Debt (log)	0.54	(0.059)	[0.42 : 0.66]
EU	0.49	(0.068)	[0.36 : 0.62]
VPI	-0.41	(0.062)	[-0.53 : -0.29]
Political constraints	-0.39	(0.083)	[-0.54 : -0.22]
GDPpc	-0.20	(0.077)	[-0.35 : -0.047]
Electoral competition	-0.12	(0.051)	[-0.22 : -0.018]
Contiguity dependency (BCG)	-0.04	(0.082)	[-0.21 : 0.12]
Corporatism	0.03	(0.056)	[-0.076 : 0.14]
** Goodness of fit (R2)	0.59	(0.00151)	[0.59 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.22	(0.076)	[-1.4 : -1.1]
GDPpc	0.53	(0.085)	[0.36 : 0.69]
Trade dependency (BCG)	0.48	(0.089)	[0.31 : 0.65]
Corporatism	0.40	(0.061)	[0.28 : 0.52]
Debt (log)	0.32	(0.064)	[0.2 : 0.45]
Electoral competition	-0.27	(0.05)	[-0.37 : -0.18]
EU	-0.20	(0.079)	[-0.36 : -0.049]
Political constraints	0.20	(0.085)	[0.027 : 0.36]
Contiguity dependency (BCG)	0.09	(0.08)	[-0.061 : 0.26]
** Goodness of fit (R2)	0.56	(0.00204)	[0.55 : 0.56]

Table 17.1: Model parameters. Capped learning (targets). Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

18

Explain Portfolio size over implementation capacity: With regional authority

m-342

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
```

```
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai
rai <- rai %>%
  mutate(`Regional authority` = (`Self-Rule` + `Shared rule`) / 2)

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
```

```
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
              select(Origin, Destination, p.contiguous),
              by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
              ungroup() %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
              group_by(Sector, Country, Year) %>%
              summarize(VPI = mean(VPI))) %>%
```

```
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`)
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`,
         `Regional authority`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
```

```
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
```

```
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.513 -1.404 -0.415 0.605 -0.862 ...
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
```

```
##  $ nDecades : int 5
##  $ X        : num [1:69, 1:43, 1:7] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV       : int 7
##  $ XS       : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS      : int 3
##  $ b0       : num [1:10] 0 0 0 0 0 0 0 0 0 0
##  $ B0       : num [1:10, 1:10] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
```

```
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-342.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-342.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
```

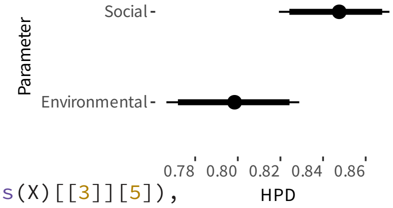


Figure 18.1: Auto-regressive components.

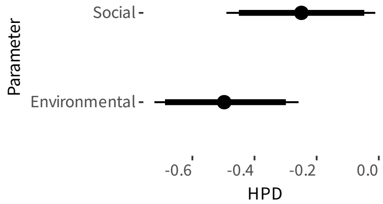


Figure 18.2: Heteroskedasticity controls (Political constraints).

```
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

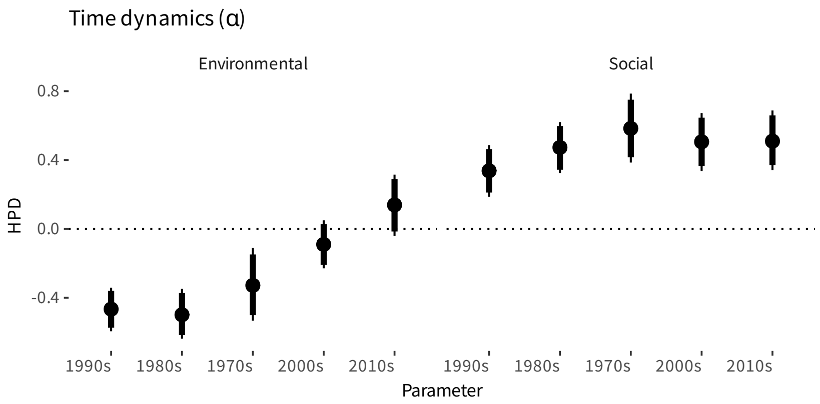


Figure 18.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-342.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

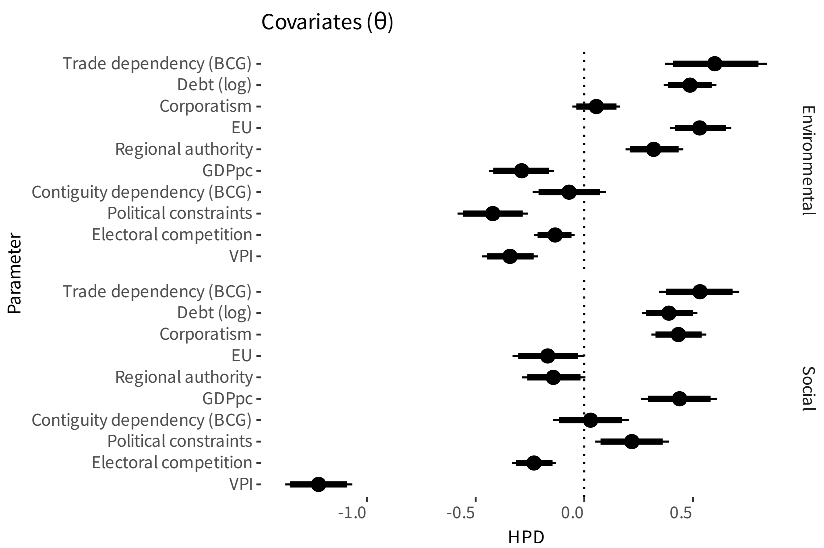


Figure 18.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
```

```
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

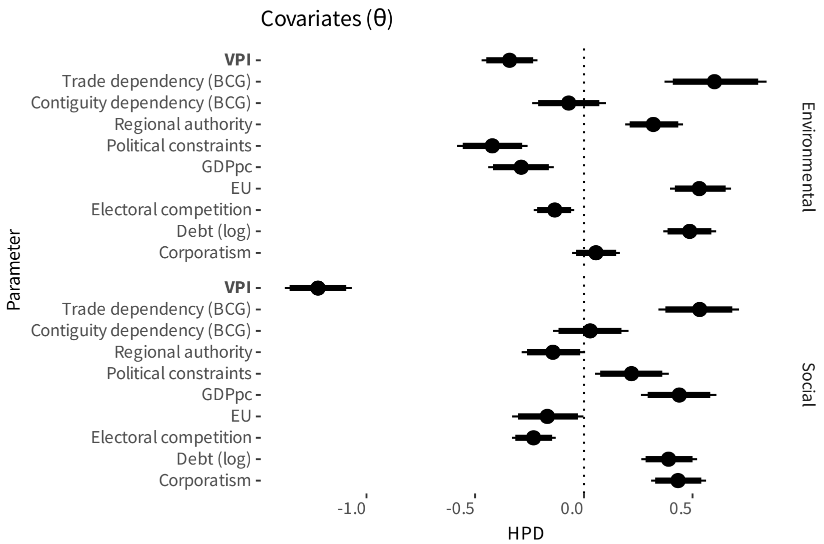


Figure 18.5: Covariates.

18.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

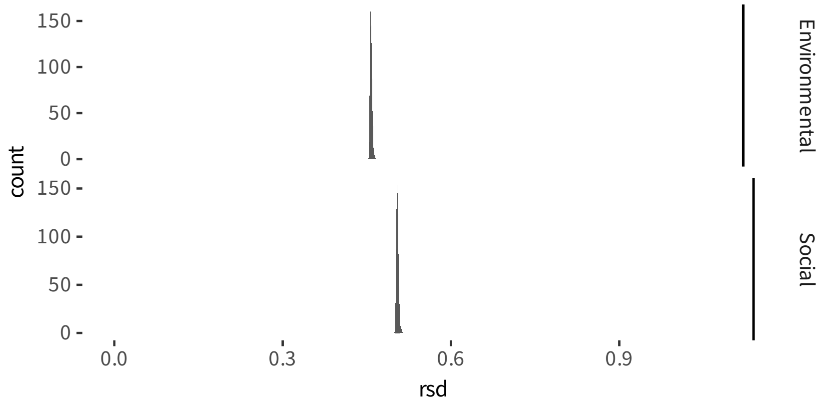


Figure 18.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

Sector	Pseudo.R2
Environmental	0.5919
Social	0.5570

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. With regional authority. Coefficient point estimates (median of the posterior distrib


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-342}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.60	(0.119)	[0.37 : 0.84]
EU	0.53	(0.072)	[0.4 : 0.68]
Debt (log)	0.49	(0.062)	[0.37 : 0.61]
Political constraints	-0.42	(0.083)	[-0.58 : -0.26]
VPI	-0.34	(0.066)	[-0.47 : -0.21]
Regional authority	0.32	(0.067)	[0.19 : 0.46]
GDPpc	-0.29	(0.078)	[-0.44 : -0.14]
Electoral competition	-0.13	(0.047)	[-0.23 : -0.044]
Contiguity dependency (BCG)	-0.07	(0.086)	[-0.24 : 0.1]
Corporatism	0.06	(0.057)	[-0.055 : 0.17]
** Goodness of fit (R2)	0.59	(0.00156)	[0.59 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.22	(0.079)	[-1.4 : -1.1]
Trade dependency (BCG)	0.53	(0.095)	[0.34 : 0.71]
GDPpc	0.44	(0.088)	[0.26 : 0.61]
Corporatism	0.43	(0.064)	[0.31 : 0.56]
Debt (log)	0.39	(0.066)	[0.26 : 0.52]
Electoral competition	-0.23	(0.051)	[-0.33 : -0.13]
Political constraints	0.22	(0.087)	[0.051 : 0.39]
EU	-0.17	(0.084)	[-0.33 : -0.0019]
Regional authority	-0.14	(0.074)	[-0.29 : 0.0053]
Contiguity dependency (BCG)	0.03	(0.089)	[-0.14 : 0.21]
** Goodness of fit (R2)	0.56	(0.00163)	[0.55 : 0.56]

Table 18.1: Model parameters. With regional authority. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a22.tex")
```

19

Explain Portfolio size over implementation capacity: Gap standardized

m-352

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
```

```
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
```

```
                         Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)



Y.df <- d %>%
  select(Country, Sector, Year, `Implementation deficit`) %>%
  mutate(`Implementation deficit` = std1(`Implementation deficit`))
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Implementation deficit")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
#         `Top-down (VPI)`, `Bottom-up (VPI)`) %>%
```

```
        `Contiguity dependency (BCG)`,
        `Trade dependency (BCG)`,
        `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
```

```
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
##  $ Y        : num [1:2, 1:69, 1:43] -1.261 -1.164 -0.289 0.613 -0.685 ...
##  $ nC       : int 69
##  $ nS       : int 2
##  $ nY       : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
```

```
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV        : int 6
##  $ XS        : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS       : int 3
##  $ b0        : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0        : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
```

```
##         }
##       }
##     }
##     for (v in 1:(nVS)) {
##       for (c in 1:nC) {
##         for (s in 1:nS) {
##           XS[s,c,1,v] ~ dunif(-1, 1)
##           for (y in 2:nY) {
##             XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##           }
##         }
##       }
##     }
## }
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)

L.lambda <- plab("lambda", list(Variable = c("(Intercept)", dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-352.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Sector")


L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-352.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

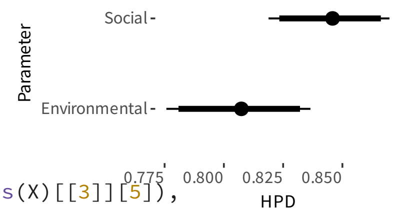


Figure 19.1: Auto-regressive components.

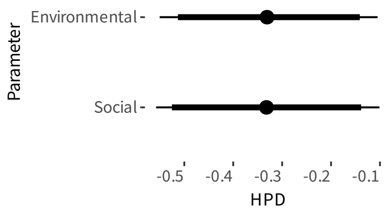


Figure 19.2: Heteroskedasticity controls (Political constraints).

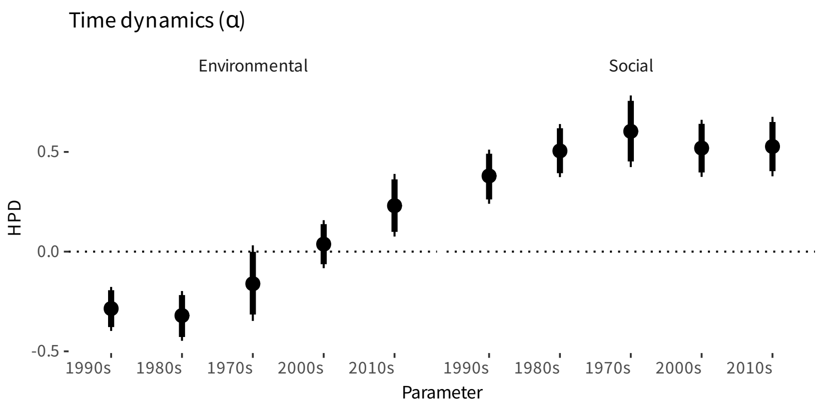


Figure 19.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-352.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

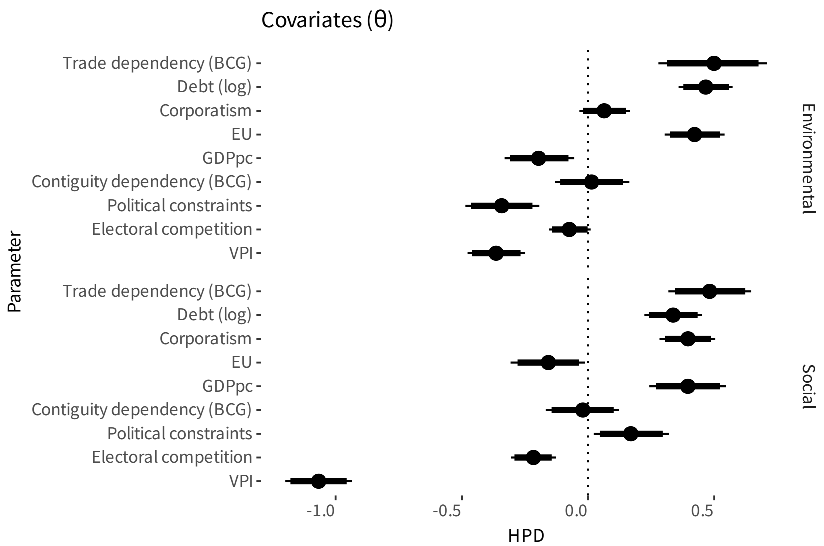


Figure 19.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

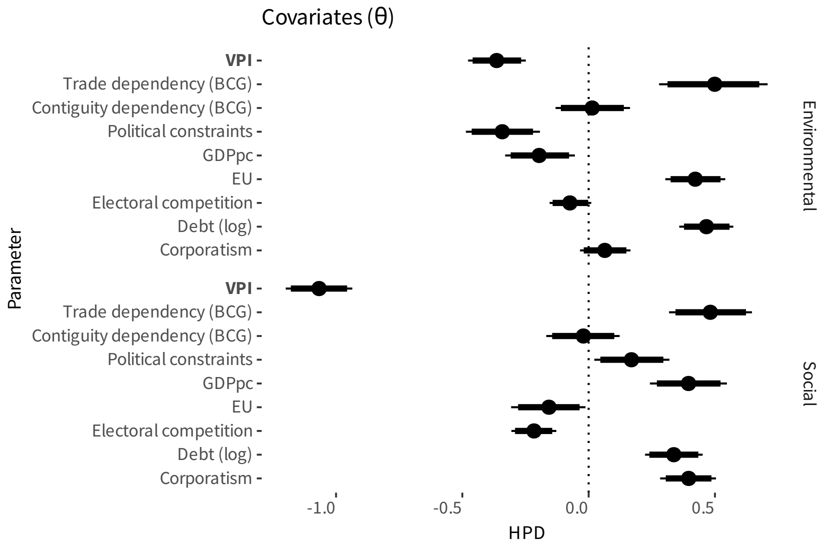


Figure 19.5: Covariates.

19.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                              Country = country.label,
                              Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.5865
Social	0.5551

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

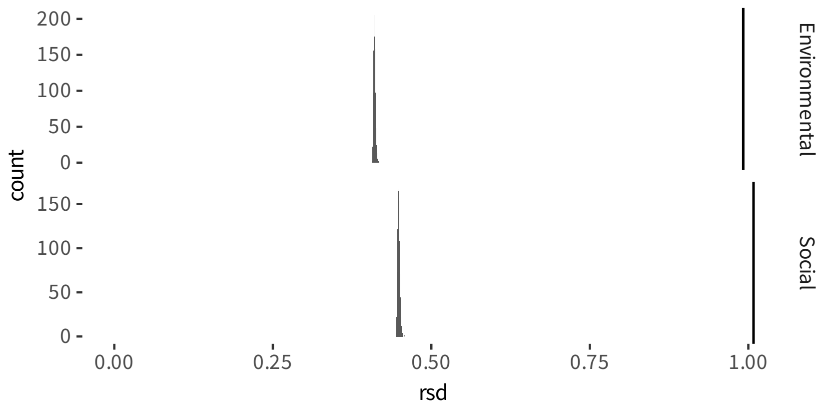


Figure 19.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Gap standardized. Coefficient point estimates (median of the posterior distribution),


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-352}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a23.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.50	(0.109)	[0.28 : 0.71]
Debt (log)	0.47	(0.054)	[0.36 : 0.57]
EU	0.42	(0.06)	[0.3 : 0.54]
VPI	-0.36	(0.058)	[-0.48 : -0.25]
Political constraints	-0.34	(0.074)	[-0.49 : -0.19]
GDPpc	-0.20	(0.07)	[-0.33 : -0.055]
Electoral competition	-0.07	(0.043)	[-0.15 : 0.01]
Corporatism	0.06	(0.051)	[-0.034 : 0.17]
Contiguity dependency (BCG)	0.01	(0.075)	[-0.13 : 0.16]
** Goodness of fit (R2)	0.59	(0.00144)	[0.58 : 0.59]
y = Burden-Capacity gap (Social, N=903)			
VPI	-1.07	(0.068)	[-1.2 : -0.94]
Trade dependency (BCG)	0.48	(0.085)	[0.32 : 0.65]
Corporatism	0.40	(0.056)	[0.28 : 0.5]
GDPpc	0.40	(0.077)	[0.24 : 0.55]
Debt (log)	0.34	(0.059)	[0.22 : 0.45]
Electoral competition	-0.22	(0.045)	[-0.31 : -0.13]
Political constraints	0.17	(0.076)	[0.022 : 0.32]
EU	-0.16	(0.074)	[-0.31 : -0.013]
Contiguity dependency (BCG)	-0.02	(0.074)	[-0.17 : 0.12]
** Goodness of fit (R2)	0.56	(0.00171)	[0.55 : 0.56]

Table 19.1: Model parameters. Gap standardized. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

20

Explain Portfolio size over implementation capacity: Outcome is standardized implementation burden

m-1352

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is “Regular : Only sample countries : Not bounded : Original ratio”.

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
```

```
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

# State capacity
load("data-state_capacity.RData") # loads sc

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
```

```
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))




#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
```

```
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism) %>%
  left_join(sc)



Y.df <- d %>%
  select(Country, Sector, Year, PS) %>%
  mutate(PS = std1(PS))
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "PS")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)
```

```
XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
         `Trade dependency (BCG)`,
         `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               #lambda[2,s] * X[c,y,6])     # polcon
                               lambda[2,s] * X[c,y,6]       # polcon
                             + lambda[3,s] * XS[s,c,y,3] # vpi
                             + lambda[4,s] * X[c,y,5]    # gdppc
                             + lambda[5,s] * X[c,y,1]    # corporatism
                             + lambda[6,s] * X[c,y,2]    # debt
                             + lambda[6,s] * X[c,y,3]    # elec comp
                             + lambda[8,s] * X[c,y,4] )  # eu
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             #lambda[2,s] * X[c,1,6])      # polcon
                             lambda[2,s] * X[c,1,6]        # polcon
                           + lambda[3,s] * XS[s,c,1,3]     # vpi
                           + lambda[4,s] * X[c,1,5]        # gdppc
                           + lambda[5,s] * X[c,1,1]        # corporatism
                           + lambda[6,s] * X[c,1,2]        # debt
```

```
                           + lambda[7,s] * X[c,1,3]       # elec comp
                           + lambda[8,s] * X[c,1,4] )     # eu

    }
    #tau[s] ~ dgamma(0.001, 0.001)
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:8) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1
```

```
method <- "parallel"
```

Data passed:

```
str(D)

## List of 12
##  $ Y         : num [1:2, 1:69, 1:43] -1.736 -0.807 -1.864 -0.102 -1.141 ...
##  $ nC        : int 69
##  $ nS        : int 2
##  $ nY        : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
##  $ nV        : int 6
##  $ XS        : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS       : int 3
##  $ b0        : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0        : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))

## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                                    lambda[2,s] * X[c,y,6]
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                                                lambda[2,s] * X[c,1,6]
##     }
##       #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
```

```
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:8) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))

load(file = paste("samples-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "theta|rho|pi|lambda|alpha", sort = FALSE),
           file = paste0("ggmcmc-all-", M, ".pdf"),
           plot = c("traceplot", "crosscorrelation",
                    "running", "caterpillar"))

L.rho <- plab("rho", list(Sector = sector.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             dimnames(X)[[3]][6],
                                             dimnames(XS)[[4]][3],
                                             dimnames(X)[[3]][c(5, 1, 2, 3, 4)]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "^lambda", par_labels = L.lambda)
```

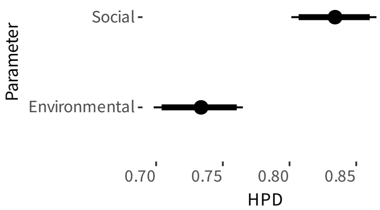


Figure 20.1: Auto-regressive components.

```
ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0("ci-lambda-1352.RData"))


S.lambda %>%
  filter(Variable != "(Intercept)") %>%
ggs_caterpillar(label = "Variable", comparison = "Sector")
```

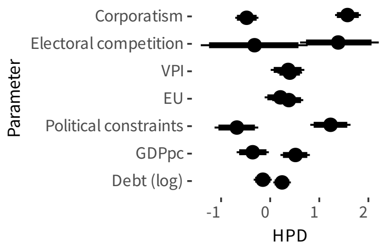


Figure 20.2: Heteroskedasticity controls (Political constraints).

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha", par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0("ci-alpha-1352.RData"))

ggs_caterpillar(S.alpha, label = "Decade") +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))
```

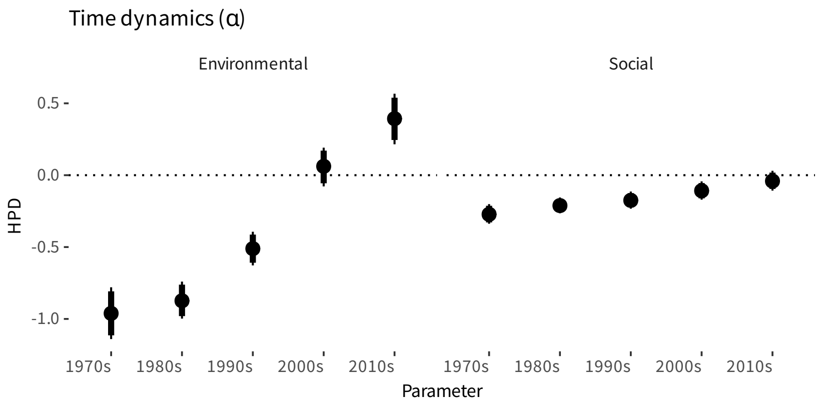


Figure 20.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-1352.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))


L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

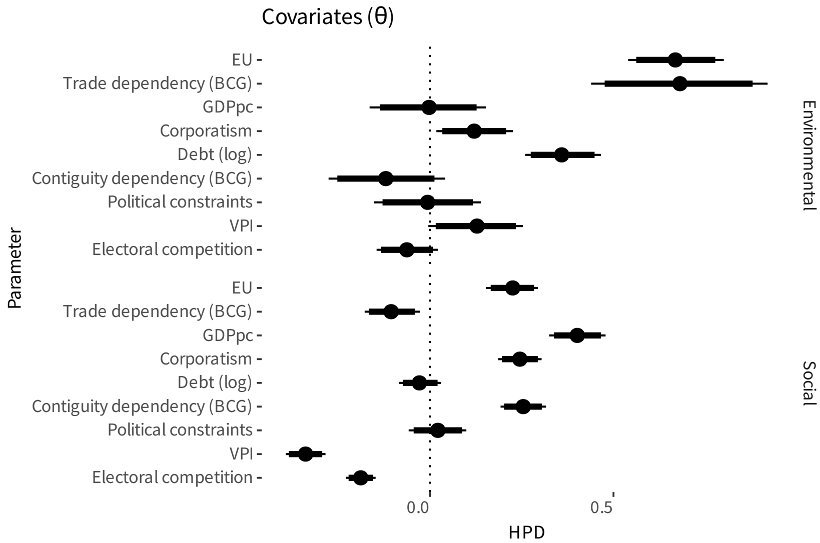


Figure 20.4: Covariates.

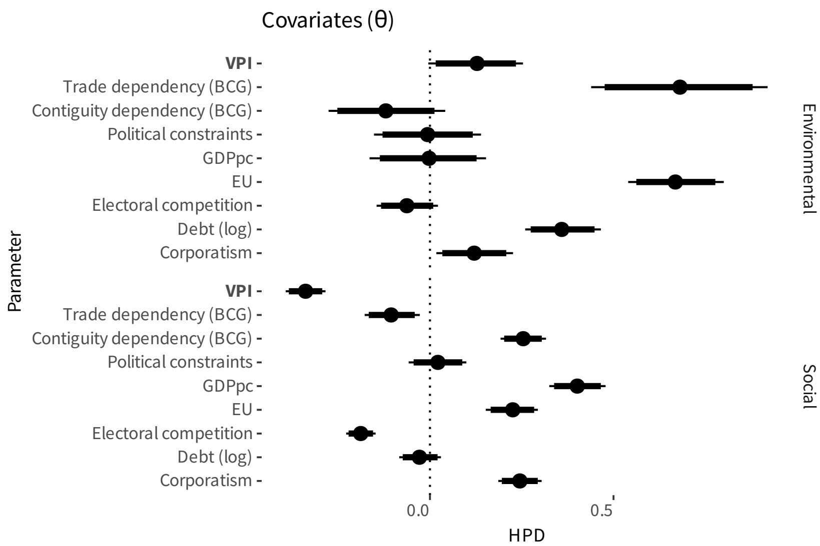


Figure 20.5: Covariates.

20.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)
```

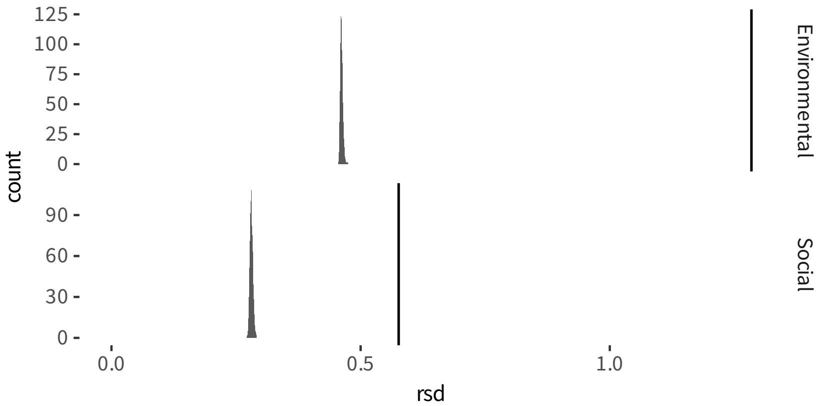


Figure 20.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.6405
Social	0.5128

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
```

```
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Outcome is standardized implementation burden. Coefficient point estimates (median of


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-1352}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
Trade dependency (BCG)	0.68	(0.123)	[0.44 : 0.92]
EU	0.67	(0.065)	[0.54 : 0.8]
Debt (log)	0.36	(0.052)	[0.26 : 0.47]
VPI	0.13	(0.066)	[-0.0044 : 0.25]
Corporatism	0.12	(0.054)	[0.017 : 0.23]
Contiguity dependency (BCG)	-0.12	(0.08)	[-0.28 : 0.042]
Electoral competition	-0.06	(0.043)	[-0.15 : 0.022]
Political constraints	-0.01	(0.075)	[-0.15 : 0.14]
GDPpc	0.00	(0.08)	[-0.16 : 0.15]
** Goodness of fit (R2)	0.64	(0.00191)	[0.64 : 0.64]
y = Burden-Capacity gap (Social, N=903)			
GDPpc	0.40	(0.039)	[0.33 : 0.48]
VPI	-0.34	(0.028)	[-0.39 : -0.28]
Contiguity dependency (BCG)	0.25	(0.032)	[0.19 : 0.32]
Corporatism	0.24	(0.03)	[0.19 : 0.3]
EU	0.23	(0.036)	[0.15 : 0.29]
Electoral competition	-0.19	(0.02)	[-0.23 : -0.15]
Trade dependency (BCG)	-0.11	(0.038)	[-0.18 : -0.028]
Debt (log)	-0.03	(0.029)	[-0.084 : 0.03]
Political constraints	0.02	(0.04)	[-0.058 : 0.099]
** Goodness of fit (R2)	0.51	(0.00501)	[0.5 : 0.52]

Table 20.1: Model parameters. Outcome is standardized implementation burden. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a24.tex")
```

21

Explain Portfolio size over implementation capacity: Outcome is standardized implementation capacity

m-2352

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels()
```

The chosen implementation is "Regular : Only sample countries : Not bounded : Original ratio".

```
load("vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

vpi <- vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year >= 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = "right"), VPI)) %>%
  ungroup()


# Add fake countries

# Countries for which VPI varies
vpi <- vpi %>%
  spread(Dimension, VPI) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 1:19)),
                   `Top-down (VPI)` = seq(0, 6, by = 1/3),
                   `Bottom-up (VPI)` = 0) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (1:19))),
                   `Top-down (VPI)` = 0,
                   `Bottom-up (VPI)` = seq(0, 6, by = 1/3)) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector))) %>%
  gather(Dimension, VPI, -c(Sector, Country, Year))

# Countries for whom vpi is average (0 gets added later), and polcon varies.
vpi <- vpi %>%
```

```
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   VPI = NA) %>%
            expand_grid(Year = unique(vpi$Year),
                        Sector = unique(vpi$Sector),
                        Dimension = unique(vpi$Dimension)))

# GDPpc
# GDp growth
# Trade openess
load("data-enlarged-wdi.RData") # loads wdi and wdi.average

# Government effectiveness
load("data-enlarged-government_effectiveness.RData") # loads ge and ge.average

# Political Constraints
load("data-enlarged-political_constraints.RData") # loads pc and pc.average
# Add fake countries for Political constrants
pc <- pc %>%
  bind_rows(tibble(Country = paste0("Z-", sprintf("%02d", 19 + (20:29))),
                   `Political constraints` = seq(min(pc$`Political constraints`, na.rm = TRUE),
                                                 max(pc$`Political constraints`, na.rm = TRUE),
                                                 length.out = 10)) %>%
            expand_grid(Year = min(pc$Year):max(pc$Year)))



# Veto players
load("data-enlarged-veto_players.RData") # loads vp and vp.average

# Socialist / Green
load("data-enlarged-green_socialist.RData") # loads green.socialist

# Leader / Liberal
load("data-enlarged-leader_liberal.RData") # loads leader.liberal

# Electoral competition
load("data-enlarged-electoral_competitiveness.RData") # loads ec

# Debt
load("data-enlarged-debt.RData") # loads debt

# EU
load("data-enlarged-eu.RData") # loads eum.yearly

# VDem CSO participation
load("data-enlarged-vdem.RData") # loads vdem

# Regional authority index
load("data-enlarged-rai.RData") # loads rai

# Trade data
load("trade/trade.RData")

# Border contiguity
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% unique(d.id$Country),
                       dimnames(M.borders)[[2]] %in% unique(d.id$Country)]

# Corporatism
load("corporatism_jahn.RData") # loads corporatism

std1 <- function(x) (x - mean(x, na.rm = TRUE)) / (1 * sd(x, na.rm = TRUE))
std <- function(x) (x - mean(x, na.rm = TRUE)) / (2 * sd(x, na.rm = TRUE))


universe <- d.id %>%
  select(Country, Sector, Year, `Implementation deficit`, PS, IC) %>%
  unique()


# Add fake countries
universe <- universe %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country = paste0("Z-", sprintf("%02d", 1:(19+29))),
```

```
                        Year = min(universe$Year):max(universe$Year))) #%>%


full.universe <- expand.grid(Country = unique(universe$Country),
                             Sector = unique(universe$Sector),
                             Year = unique(universe$Year))


# Contiguity, Border interdependence as tidy data
interdependency.contiguity <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wID = `Implementation deficit` * p.contiguous) %>%
  mutate(wPS = PS * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Contiguity dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Contiguity dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2019) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Contiguity dependency (BCG)` = NA,
                        `Contiguity dependency (PS)` = NA))


# Trade interdependence as tidy data
interdependency.trade <- universe %>%
  filter(!str_detect(Country, "^Z-")) %>%
  select(Sector, Destination = Country, Year, `Implementation deficit`, PS) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wID = `Implementation deficit` * p.Exports) %>%
  mutate(wPS = PS * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year) %>%
  summarize(`Trade dependency (BCG)` = sum(wID, na.rm = TRUE),
            `Trade dependency (PS)` = sum(wPS, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% unique(universe$Country)[!str_detect(unique(universe$Country), "^Z-")]) %>%
  filter(Year >= 1976 & Year <= 2018) %>%
  group_by(Sector) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = unique(universe$Sector),
                        Country =
                          unique(universe$Country)[str_detect(unique(universe$Country), "^Z-")],
                        Year = 1976:2018,
                        `Trade dependency (BCG)` = NA,
                        `Trade dependency (PS)` = NA))



#d <- d.id %>%
d <- universe %>%
  ungroup() %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(ge) %>%
```

```
  left_join(pc) %>%
  left_join(vp) %>%
  left_join(green.socialist) %>%
  left_join(leader.liberal) %>%#
  left_join(ec.full.year.average) %>%
  left_join(debt) %>%
  left_join(eum.yearly) %>%
  left_join(vdem) %>%
  left_join(rai) %>%
  left_join(interdependency.contiguity) %>%
  left_join(interdependency.trade) %>%
  left_join(corporatism)
```

```
Y.df <- d %>%
  select(Country, Sector, Year, IC) %>%
  mutate(PS = std1(IC))
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "IC")


sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.real <- 21
nC.fake <- nC - nC.real
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)
id.year.2000 <- which(year.label.numeric == 2000)

decade.text <- paste0(str_sub(year.label, 1, 3), "0s")
id.decade <- as.numeric(as.factor(decade.text))
decade.label <- levels(as.factor(decade.text))
nDecades <- length(decade.label)

# Prepare the matrix with control values
# Select variables
# Standardize their values
X <- d %>%
  select(Country, Year,
         GDPpc,
         `Debt`,
         `Political constraints`,
         Corporatism,
         `Electoral competition`
         ) %>%#,
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  # Add binary variables
  left_join(unique(select(d, Country, Year, EU))) %>%
  mutate(EU = ifelse(is.na(EU), 0, EU)) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label <- dimnames(X)[[3]]
nV <- length(variable.label)


XS <- d %>%
  select(Country, Sector, Year,
         `Contiguity dependency (BCG)`,
```

```
          `Trade dependency (BCG)`,
          `VPI`) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value)) %>%
  ungroup() %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label <- dimnames(XS)[[4]]
nVS <- length(variable.sector.label)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
diag(B0) <- 2.5^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.5), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
        mu[s,c,y] <- alpha[s,id.decade[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
        tau[s,c,y] <- 1 / sigma.sq[s,c,y]
        sigma.sq[s,c,y] <- exp(lambda[1,s] +
                               lambda[2,s] * X[c,y,5])     # polcon
      }
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
      mu[s,c,1] <- alpha[s,id.decade[1]] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      tau[s,c,1] <- 1 / sigma.sq[s,c,1]
      sigma.sq[s,c,1] <- exp(lambda[1,s] +
                             lambda[2,s] * X[c,1,5])     # polcon

    }
    #
    rho[s] ~ dunif(-1, 1)
    theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
    Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
    Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
    for (d in 1:nDecades) {
      alpha[s,d] ~ dnorm(0, 1^-2)
    }
    for (l in 1:2) {
      lambda[l,s] ~ dnorm(0, 25^-2)
    }
  }
  #
  # Missing data
  #
```

```
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dunif(-1, 1)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dunif(-1, 1)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda")

par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 1) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)

par.resid <- expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 1:nY) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)

chains <- 3
adapt <- 2e2
burnin <- 2e3
run <- 2e3
thin <- 1

chains <- 3
adapt <- 2e2
burnin <- 1e4
run <- 2e3
thin <- 1


method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 12
##  $ Y         : num [1:2, 1:69, 1:43] 0.176 0.456 -0.961 -0.427 -0.019 ...
##  $ nC        : int 69
##  $ nS        : int 2
##  $ nY        : int 43
##  $ id.decade: num [1:43] 1 1 1 1 2 2 2 2 2 2 ...
##  $ nDecades : int 5
##  $ X         : num [1:69, 1:43, 1:6] -0.201 0.928 0.321 -0.719 0.152 ...
```

```
##  $ nV        : int 6
##  $ XS        : num [1:2, 1:69, 1:43, 1:3] 0.00419 0.02397 -0.07403 0.63222 -0.60633 ...
##  $ nVS       : int 3
##  $ b0        : num [1:9] 0 0 0 0 0 0 0 0 0
##  $ B0        : num [1:9, 1:9] 0.16 0 0 0 0 0 0 0 0 0 ...
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c,y])
##         mu[s,c,y] <- alpha[s,id.decade[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         tau[s,c,y] <- 1 / sigma.sq[s,c,y]
##         sigma.sq[s,c,y] <- exp(lambda[1,s] +
##                                lambda[2,s] * X[c,y,5])           }
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c,1])
##       mu[s,c,1] <- alpha[s,id.decade[1]] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       tau[s,c,1] <- 1 / sigma.sq[s,c,1]
##       sigma.sq[s,c,1] <- exp(lambda[1,s] +
##                              lambda[2,s] * X[c,1,5])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     theta[s,1:(nV + nVS)] ~ dmnorm(b0, Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     Omega[s,1:(nV + nVS),1:(nV + nVS)] ~ dwish(B0, (nV + nVS + 1))
##     Sigma[s,1:(nV + nVS),1:(nV + nVS)] <- inverse(Omega[s,1:(nV + nVS),1:(nV + nVS)])
##     for (d in 1:nDecades) {
##       alpha[s,d] ~ dnorm(0, 1^-2)
##     }
##     for (l in 1:2) {
##       lambda[l,s] ~ dnorm(0, 25^-2)
##     }
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dunif(-1, 1)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
```

```
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dunif(-1, 1)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }

t0 <- proc.time()
rj <- run.jags(model = paste(”models/model-”, M, ”.bug”, sep = ””),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = ”glm”,
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste(”samples-”, M, ”.RData”, sep = ””))

load(file = paste(”samples-”, M, ”.RData”, sep = ””))

ggmcmc(ggs(s, family = ”theta|rho|pi|lambda|alpha”, sort = FALSE),
           file = paste0(”ggmcmc-all-”, M, ”.pdf”),
           plot = c(”traceplot”, ”crosscorrelation”,
                    ”running”, ”caterpillar”))

L.rho <- plab(”rho”, list(Sector = sector.label))
S.rho <- ggs(s, family = ”rho”, par_labels = L.rho)
ggs_caterpillar(S.rho)


L.lambda <- plab(”lambda”, list(Variable = c(”(Intercept)”, dimnames(X)[[3]][5]),
                                Sector = sector.label))
S.lambda <- ggs(s, family = ”^lambda”, par_labels = L.lambda)

ci.lambda <- ci(S.lambda) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.lambda, file = paste0(”ci-lambda-2352.RData”))


S.lambda %>%
  filter(Variable != ”(Intercept)”) %>%
ggs_caterpillar(label = ”Sector”)


L.alpha <- plab(”alpha”, list(Sector = sector.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = ”^alpha”, par_labels = L.alpha)

ci.alpha <- ci(S.alpha) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.alpha, file = paste0(”ci-alpha-2352.RData”))

ggs_caterpillar(S.alpha, label = ”Decade”) +
  coord_flip() +
  facet_grid(~ Sector) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste(”Time dynamics (”, alpha, ”)”, sep = ””)))
```

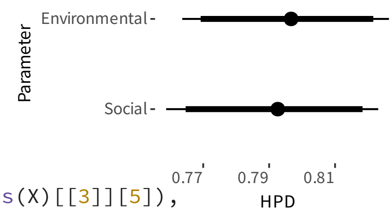


Figure 21.1: Auto-regressive components.

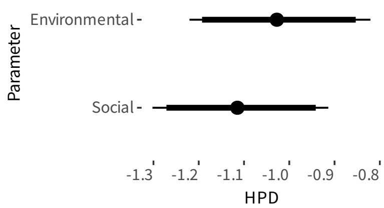


Figure 21.2: Heteroskedasticity controls (Political constraints).

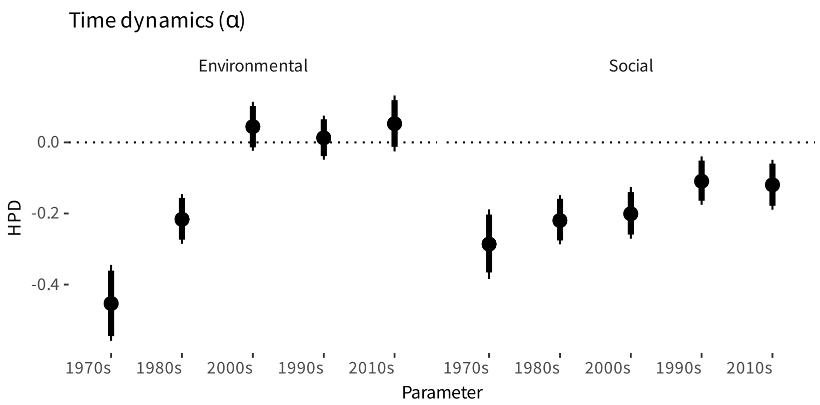


Figure 21.3: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)

ci.theta <- ci(S.theta) %>%
  mutate(Model = model.label,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2352.RData"))

ggs_caterpillar(S.theta, label = "Covariate") +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

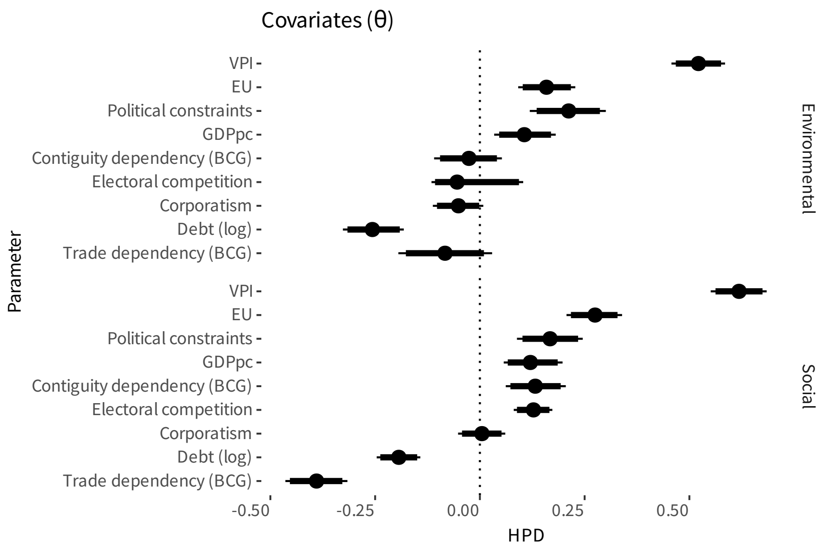


Figure 21.4: Covariates.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta)# %>%
breaks <- levels(S.theta$Covariate)
toBold <- as.character(breaks[str_detect(breaks, "VPI")])

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  facet_grid(Sector ~ .) +
  geom_vline(xintercept = 0, lty = 3) +
  theme(axis.text.y = element_text(face = ifelse(breaks %in% toBold, "bold", "plain"))) +
  ggtitle(expression(paste("Covariates (", theta, ")", sep = "")))
```

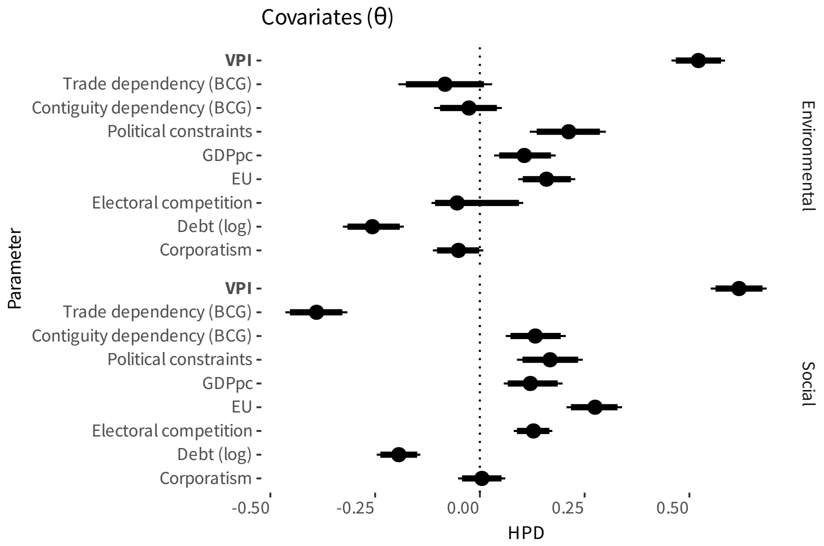


Figure 21.5: Covariates.

21.1 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Sector = sector.label,
                             Country = country.label,
                             Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1,
         Country = Var2,
         Year = Var3,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(!str_detect(Country, "^Z-")) %>%
  group_by(Iteration, Chain, Sector) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.0001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Sector ~ ., scales = "free") +
  expand_limits(x = 0)


S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Sector) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Sector	Pseudo.R2
Environmental	0.6115
Social	0.6290

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
```

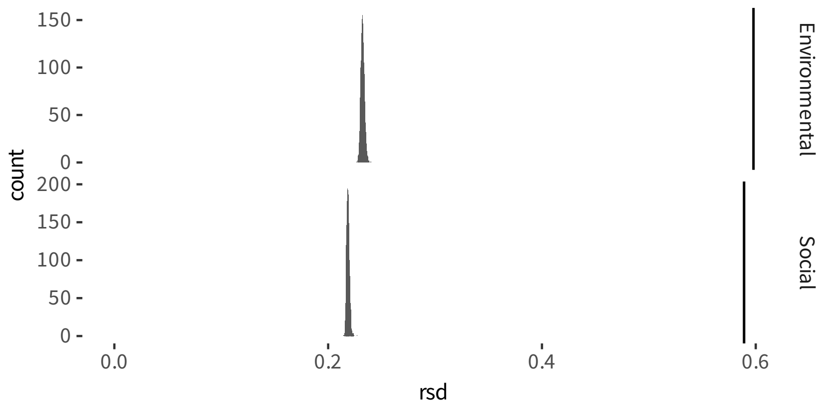


Figure 21.6: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
  group_by(Sector, Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)


rm(S.rsd)

tb <- S.theta %>%
  select(Iteration, Chain, Sector, Covariate, value) %>%
  group_by(Sector, Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  group_by(Sector) %>%
  arrange(Sector, id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Sector, Covariate, Coefficient, SD, `95% CI`)

rws.env <- which(tb$Sector == "Environmental")
rws.soc <- which(tb$Sector == "Social")

tc <- "Model parameters. Outcome is standardized implementation capacity. Coefficient point estimates (median


tb2 <- tb %>%
  ungroup() %>%
  select(-Sector) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-2352}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Burden-Capacity gap (Environmental, N=", nC.real * nY ,")"),  rws.env[1], rws.env[leng
  pack_rows(paste0("y = Burden-Capacity gap (Social, N=", nC.real * nY ,")"), rws.soc[1], rws.soc[length(rws.s
print(tb2)


tb2 %>%
  save_kable(file = "TAB-a25.tex")
```

Covariate	Coefficient	SD	95% CI
y = Burden-Capacity gap (Environmental, N=903)			
VPI	0.52	(0.033)	[0.46 : 0.59]
Debt (log)	-0.26	(0.038)	[-0.33 : -0.18]
Political constraints	0.21	(0.046)	[0.12 : 0.3]
EU	0.16	(0.035)	[0.091 : 0.23]
GDPpc	0.11	(0.038)	[0.034 : 0.18]
Trade dependency (BCG)	-0.08	(0.057)	[-0.19 : 0.029]
Electoral competition	-0.05	(0.071)	[-0.12 : 0.1]
Corporatism	-0.05	(0.031)	[-0.11 : 0.0082]
Contiguity dependency (BCG)	-0.03	(0.042)	[-0.11 : 0.053]
** Goodness of fit (R2)	0.61	(0.00274)	[0.61 : 0.62]
y = Burden-Capacity gap (Social, N=903)			
VPI	0.62	(0.034)	[0.55 : 0.68]
Trade dependency (BCG)	-0.39	(0.038)	[-0.46 : -0.32]
EU	0.27	(0.034)	[0.21 : 0.34]
Debt (log)	-0.19	(0.027)	[-0.25 : -0.14]
Political constraints	0.17	(0.04)	[0.088 : 0.25]
Contiguity dependency (BCG)	0.13	(0.037)	[0.062 : 0.21]
Electoral competition	0.13	(0.024)	[0.081 : 0.17]
GDPpc	0.12	(0.037)	[0.057 : 0.2]
Corporatism	0.00	(0.029)	[-0.052 : 0.061]
** Goodness of fit (R2)	0.63	(0.00224)	[0.62 : 0.63]

Table 21.1: Model parameters. Outcome is standardized implementation capacity. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

22
Model comparison

22.1 Smoothed lags with different delay periods

```
d <- NULL
load("ci-theta-312.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-321.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-322.RData")
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(`Rolling average` = as.numeric(str_match(OriginalModelLabel, "Lag ([0-9]+) years")[,2])) %>%
  mutate(`Rolling average` = ifelse(is.na(`Rolling average`), 3, `Rolling average`))

#
# Figure A9
#
d %>%
  ggplot(aes(x = Covariate,
         ymin = low, y = median, ymax = high,
         group = `Rolling average`,
         color = as.factor(`Rolling average`))) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
               position = position_dodge(width = 0.4),
               size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_discrete_sequential(palette = "YlOrRd", name = "Rolling average")
```

22.2 Smoothed lags vs. lags

```
d <- NULL
load("ci-theta-312.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-323.RData")
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(Lag = ifelse(Model == "Reference model", "Lag + rolling average", "Plain lag"))

#
# Figure A10
#
d %>%
  ggplot(aes(x = Covariate,
         ymin = low, y = median, ymax = high,
         group = `Lag`,
```

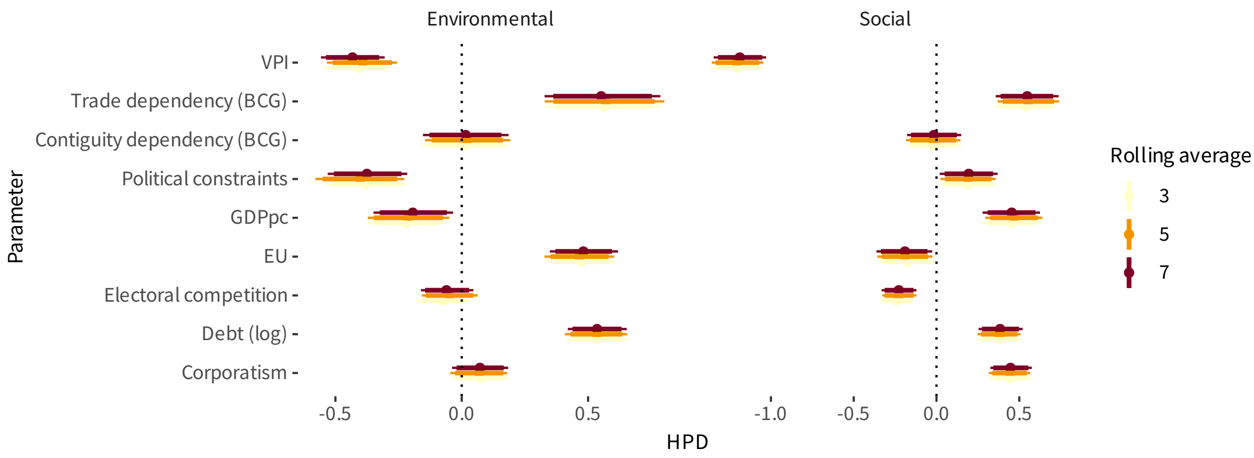


Figure 22.1: Comparison of model results between different rolling averages for VPI effects.

```
          color = `Lag`)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_manual(values = c("black", "#D55E00"))
```

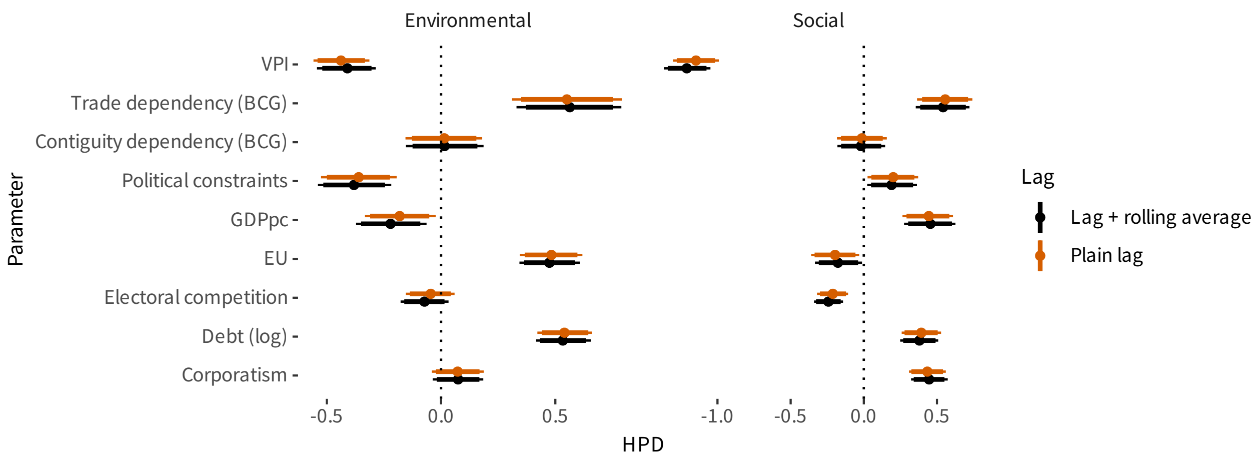


Figure 22.2: Comparison of model results between using either a simple lag or rolling averages for VPI effects.

22.3 Decay of portfolio space, Instruments.

```
d <- NULL
load("ci-theta-312.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-318.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-319.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-320.RData")
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(Learning = case_when(
```

```
        str_detect(OriginalModelLabel, "Continuous") ~ "Continuous learning",
        str_detect(OriginalModelLabel, "Steep") ~ "Steep learning",
        str_detect(OriginalModelLabel, "Capped") ~ "Capped learning",
        TRUE ~ "No learning")) %>%
  mutate(Learning = fct_relevel(Learning, "No learning"))

d %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Learning,
             color = Learning)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_manual(values = c("black", "#56B4E9", "maroon3",
                                "darkolivegreen4"))
```

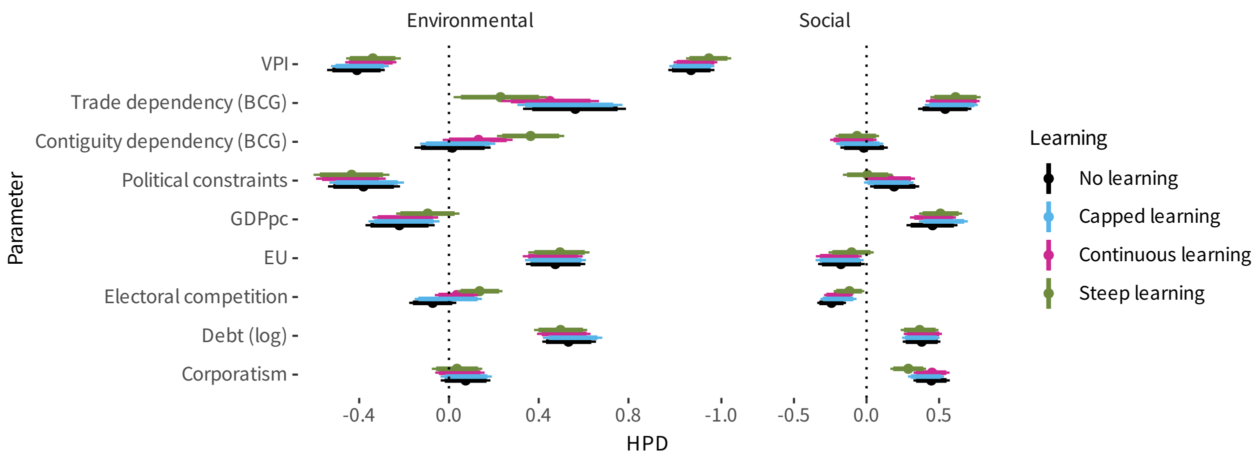


Figure 22.3: Comparison of model results using different learning functions.

```
#
# Figure 9
#
d %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Learning,
             color = Learning)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_grey()
```

22.4 *Decay of portfolio space, Targets*

```
d <- NULL
load("ci-theta-312.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-332.RData")
```

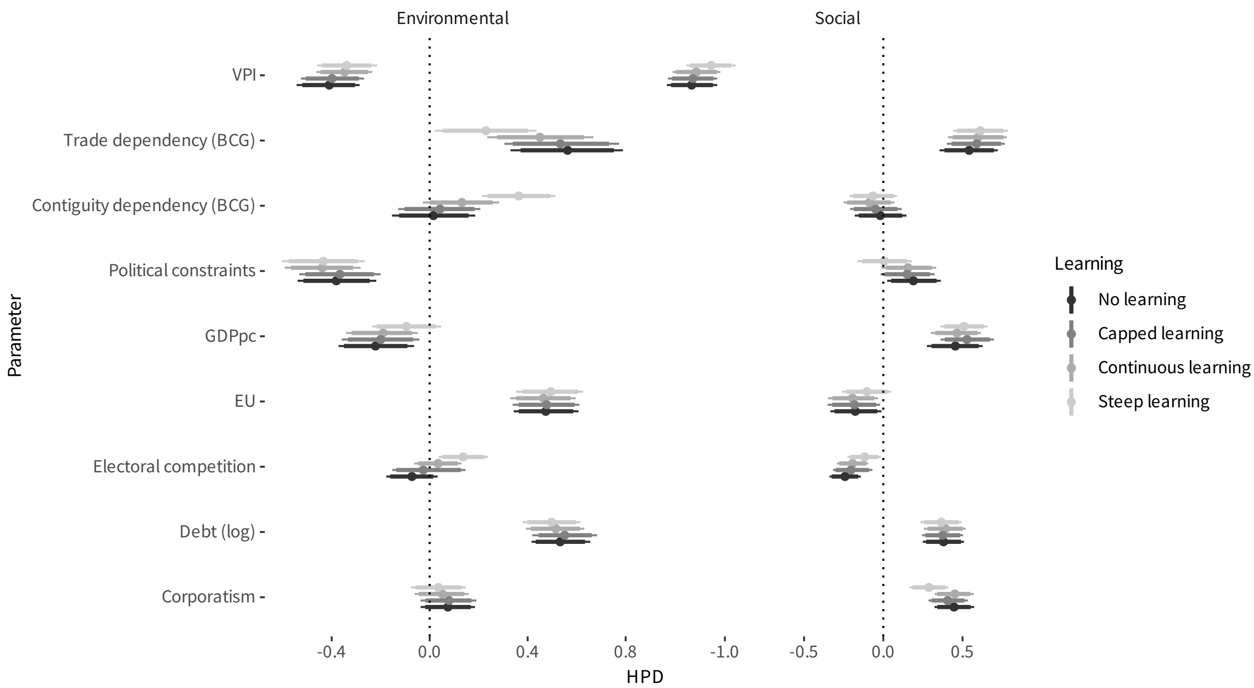


Figure 22.4: Comparison of model results using different learning functions. Grey scale.

```
d <- bind_rows(d, ci.theta)
load("ci-theta-333.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-334.RData")
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(Learning = case_when(
         str_detect(OriginalModelLabel, "Continuous") ~ "Continuous learning",
         str_detect(OriginalModelLabel, "Steep") ~ "Steep learning",
         str_detect(OriginalModelLabel, "Capped") ~ "Capped learning",
         TRUE ~ "No learning")) %>%
  mutate(Learning = fct_relevel(Learning, "No learning"))

#
# Figure A14
#
d %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Learning,
             color = Learning)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_manual(values = c("black", "#56B4E9", "maroon3",
                                "darkolivegreen4"))


d %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Learning,
             color = Learning)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
```

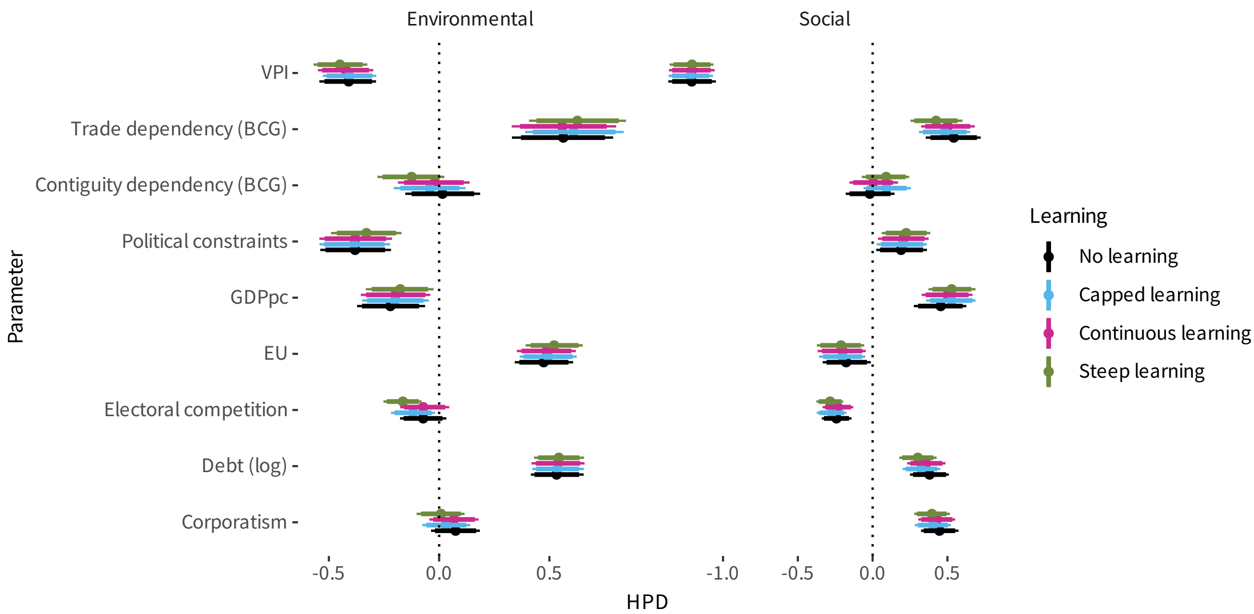


Figure 22.5: Comparison of model results using different learning functions. Weights by instruments.

```
geom_linerange(position = position_dodge(width = 0.4)) +
geom_linerange(aes(ymin = Low, ymax = High),
               position = position_dodge(width = 0.4),
               size = 1.0) +
geom_hline(aes(yintercept = 0), lty = 3) +
xlab("Parameter") + ylab("HPD") +
facet_grid(~ Sector, scales = "free") +
scale_color_grey()
```

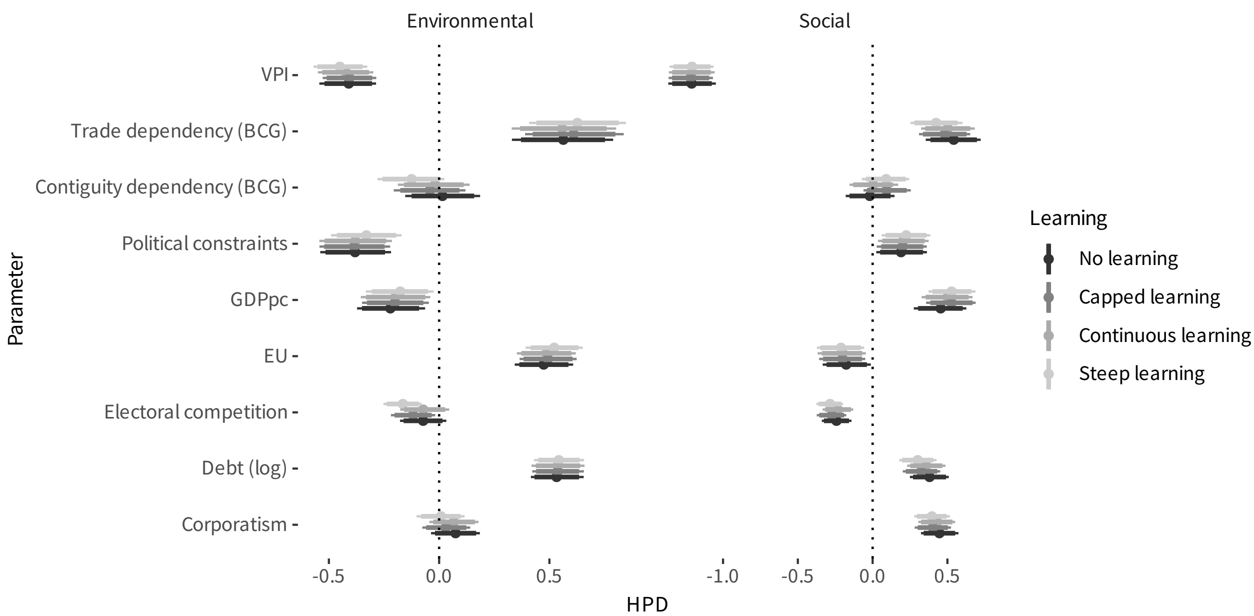


Figure 22.6: Comparison of model results using different learning functions. Grey scale. Weights by instruments.

22.5 Gap as subtraction vs as ratio

```
d <- NULL
load("ci-theta-312.RData")
d <- bind_rows(d, ci.theta)
```

```
load(”ci-theta-324.RData”)
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(Gap = ifelse(str_detect(OriginalModelLabel, ”Burden as subtraction”), ”Burden-Capacity (subtraction)”
  mutate(Gap = fct_relevel(Gap, ”Burden/Capacity (ratio)”))

#
# Figure A11
#
d %>%
  ggplot(aes(x = Covariate,
           ymin = low, y = median, ymax = high,
           group = Gap,
           color = Gap)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab(”Parameter”) + ylab(”HPD”) +
  facet_grid(~ Sector, scales = ”free”) +
  scale_color_manual(values = c(”black”, ”#56B4E9”))
```

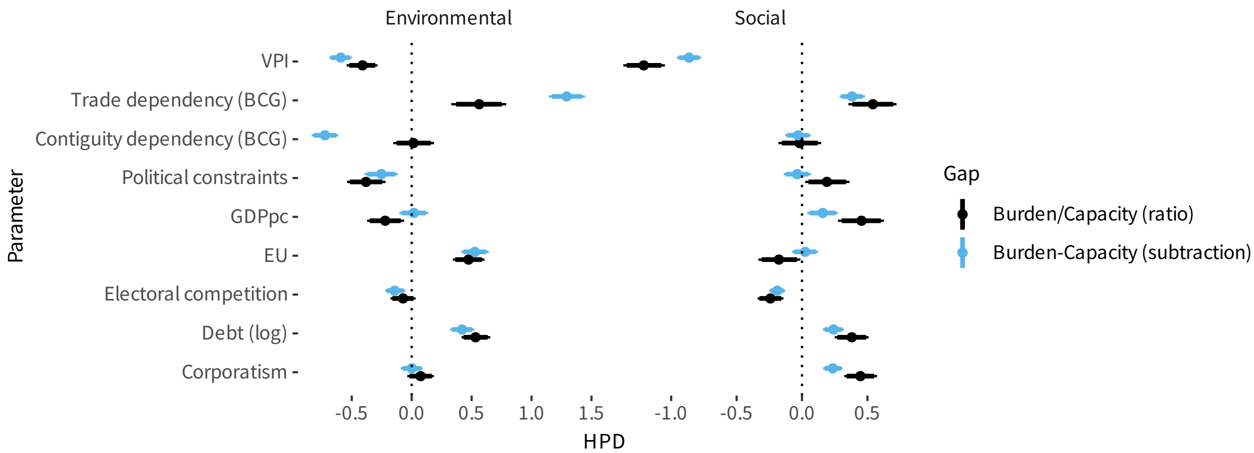


Figure 22.7: Comparison of model results between using the ratio or a subtraction, to calculate the gap.

22.6 Simplified VPI in 2 categories

```
d <- NULL
load(”ci-theta-312.RData”)
d <- bind_rows(d, ci.theta)
load(”ci-theta-314.RData”)
d <- bind_rows(d, ci.theta)
load(”ci-theta-1314.RData”)
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(VPI = ifelse(str_detect(OriginalModelLabel, ”middle category as high”), ”Simpler VPI\n(middle: high)”
  mutate(VPI = ifelse(str_detect(OriginalModelLabel, ”middle category as low”), ”Simpler VPI\n(middle: low)”,
  mutate(VPI = fct_relevel(VPI, ”Regular VPI”))

#
# Figure A13
#
d %>%
  ggplot(aes(x = Covariate,
           ymin = low, y = median, ymax = high,
```

```
          group = VPI,
          color = VPI)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_manual(values = c("black", "#56B4E9", "maroon3"))
```

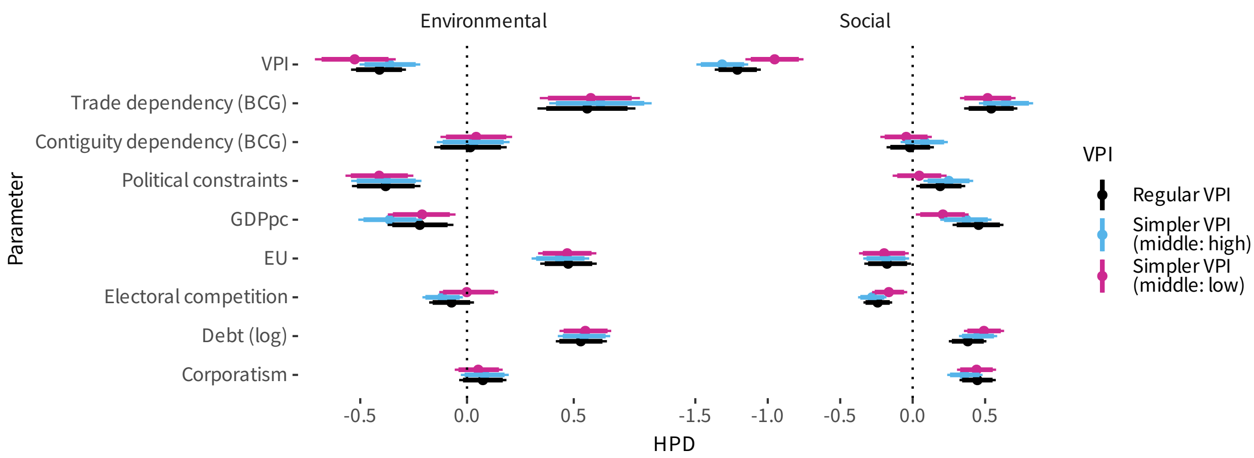


Figure 22.8: Comparison of model results between using VPI in 3 categories or simplifying it to two categories (with the middle category assigned to the upper/lower).

22.7 With/without state capacity

```
d <- NULL
load("ci-theta-312.RData")
d <- bind_rows(d, ci.theta)
load("ci-theta-325.RData")
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(OriginalModelLabel = Model) %>%
  mutate(`State capacity` = ifelse(str_detect(OriginalModelLabel, "state capacity"), "with State capacity", "R
  mutate(`State capacity` = fct_relevel(`State capacity`, "Reference"))


#
# Figure A15
#
d %>%
  ggplot(aes(x = Covariate,
          ymin = low, y = median, ymax = high,
          group = `State capacity`,
          color = `State capacity`)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  facet_grid(~ Sector, scales = "free") +
  scale_color_manual(values = c("black", "#56B4E9"))
```

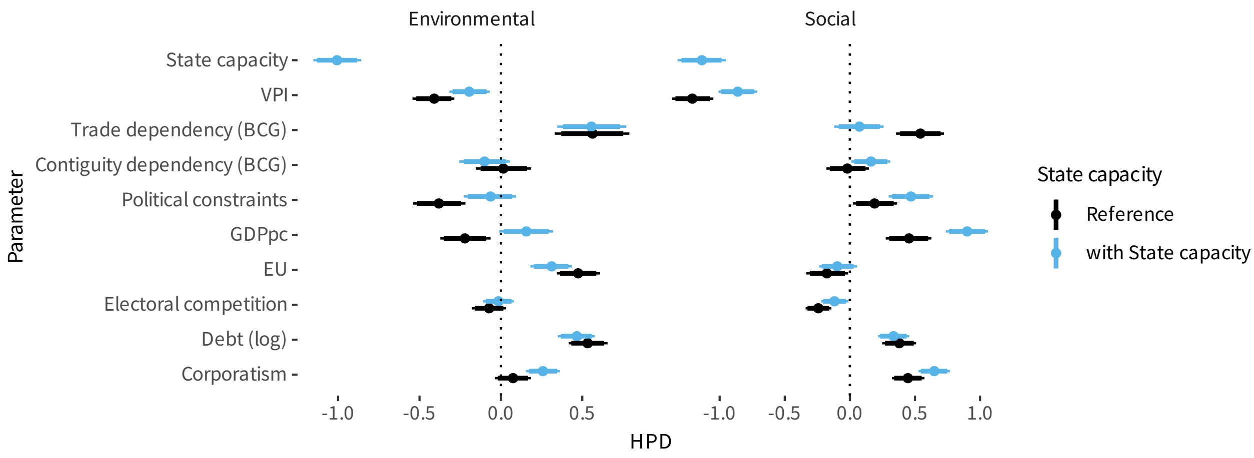


Figure 22.9: Comparison of model results between adding a control on state capacity.

22.8 *Implementation capacity has administrative costs or generosity instead*

```
d <- NULL
load(”ci-theta-312.RData”)
d <- bind_rows(d, ci.theta)
load(”ci-theta-1312.RData”)
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(`Implementation capacity contains` = ifelse(str_detect(Model, ”generosity instead of”), ”Generosity”,

#
# Figure A12
#
d %>%
  filter(Sector == ”Social”) %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = `Implementation capacity contains`,
             color = `Implementation capacity contains`)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab(”Parameter”) + ylab(”HPD”) +
  scale_color_manual(values = c(”black”, ”#56B4E9”))
```

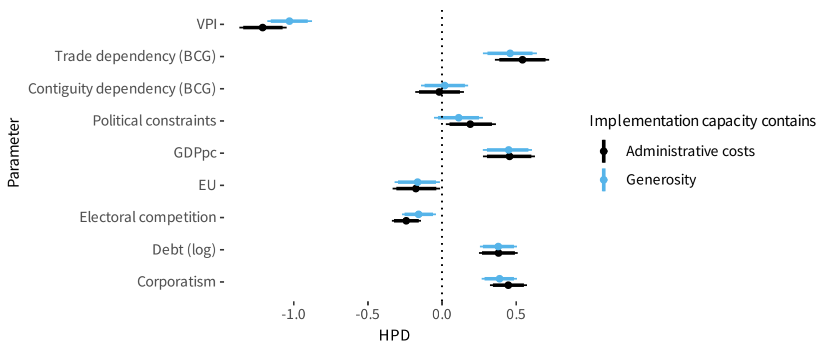


Figure 22.10: Comparison of model results between state capacity including administrative costs or generosity.

22.9 *Regional authority index*

```
d <- NULL
load(”ci-theta-312.RData”)
d <- bind_rows(d, ci.theta)
load(”ci-theta-342.RData”)
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(`Regional authority` = ifelse(str_detect(Model, ”With regional authority”), ”with Regional authority”
  mutate(`Regional authority` = fct_relevel(`Regional authority`, ”Reference”))

#
# Figure A16
#
d %>%
  filter(Sector == ”Social”) %>%
  ggplot(aes(x = Covariate,
           ymin = low, y = median, ymax = high,
           group = `Regional authority`,
           color = `Regional authority`)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab(”Parameter”) + ylab(”HPD”) +
  scale_color_manual(values = c(”black”, ”#56B4E9”))
```

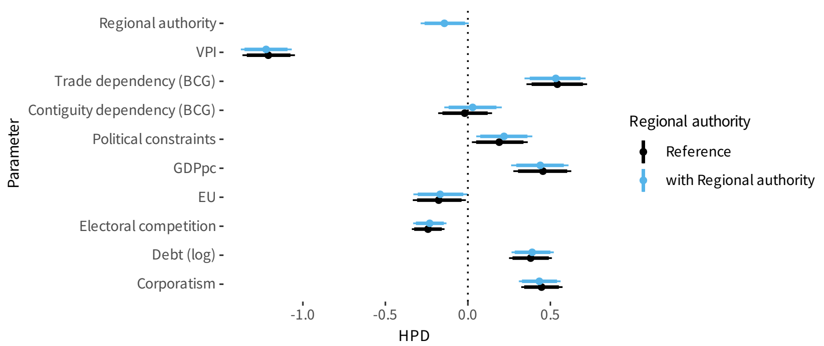


Figure 22.11: Comparison of model results when including a control for regional authority (average of self-rule and shared rule).

22.10 *Explaining the gap vs individual components*

```
d <- NULL
load(”ci-theta-352.RData”)
d <- bind_rows(d, ci.theta)
load(”ci-theta-1352.RData”)
d <- bind_rows(d, ci.theta)
load(”ci-theta-2352.RData”)
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(Outcome = case_when(
    str_detect(Model, ”Gap standardized”) ~ ”Reference (standardized)”,
    str_detect(Model, ”standardized implementation burden”) ~ ”Portfolio size”,
    str_detect(Model, ”standardized implementation capacity”) ~ ”Implementation capacity”,
    TRUE ~ NA_character_)) %>%
  mutate(Outcome = fct_relevel(Outcome, ”Reference (standardized)”))

#
# Figure A17
#
d %>%
```

```
filter(str_detect(Covariate, "VPI")) %>%
ggplot(aes(x = Covariate,
         ymin = low, y = median, ymax = high,
         group = Outcome,
         color = Outcome)) +
coord_flip() +
geom_point(position = position_dodge(width = 0.4), size = 1.5) +
geom_linerange(position = position_dodge(width = 0.4)) +
geom_linerange(aes(ymin = Low, ymax = High),
               position = position_dodge(width = 0.4),
               size = 1.0) +
geom_hline(aes(yintercept = 0), lty = 3) +
xlab("Parameter") + ylab("HPD") +
facet_grid(~ Sector, scales = "free") +
scale_color_manual(values = c("black", "#56B4E9", "maroon3"))
```

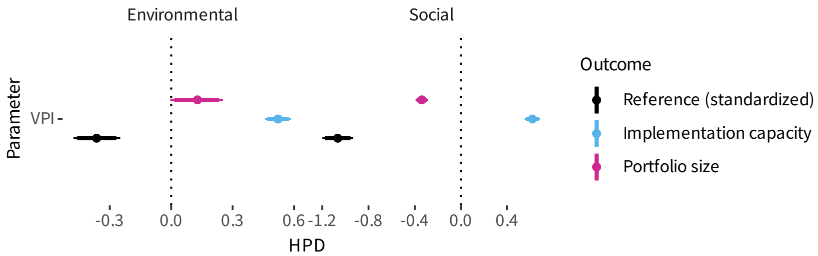


Figure 22.12: Comparison of model results between the gap and its constitutive parts.

23

Implementation capacity: Report of loadings / discriminations and correlations

```
load(file = "data-scores-restricted-environmental.RData")
g <- ci.gamma
corrs <- mutate(correlations, Sector = "cor(Env)")
load(file = "data-scores-restricted-social.RData")
g <- bind_rows(g, ci.gamma) %>%
  mutate(Sector = as.factor(str_replace(Model, "Restricted, ", "")))
corrs <- bind_rows(corrs, correlations %>%
                   mutate(Sector = "cor(Soc)"))


tb <- g %>%
  select(Component = Variable, Sector, Discrimination = median) %>%
  mutate(Discrimination = prettyNum(signif(Discrimination, 3))) %>%
  mutate(Discrimination = ifelse(Discrimination == "NA", "", Discrimination)) %>%
  pivot_wider(names_from = Sector, values_from = Discrimination) %>%
  mutate(Social = ifelse(is.na(Social), "", Social)) %>%
  mutate(Environmental = ifelse(is.na(Environmental), "", Environmental)) %>%
  mutate(Component = as.character(Component)) %>%
  arrange(Component)

tc <- "Discrimination parameters for a measurement model of implementation capacity, and the resulting correla
tb2 <- left_join(tb,
  corrs %>%
    pivot_wider(names_from = Sector, values_from = Correlation) %>%
    rename(Component = `Original variable`)) %>%
  mutate(`cor(Env)` = prettyNum(signif(`cor(Env)`, 3))) %>%
  mutate(`cor(Soc)` = prettyNum(signif(`cor(Soc)`, 3))) %>%
  mutate(`cor(Env)` = ifelse(`cor(Env)` == "NA", "", `cor(Env)`)) %>%
  mutate(`cor(Soc)` = ifelse(`cor(Soc)` == "NA", "", `cor(Soc)`)) %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:discrimination-correlations}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  add_header_above(c("", "Discrimination (point estimate)" = 2)) %>%
  kable_styling(latex_options = "scale_down") %>%
  kable_styling(font_size = 10)

print(tb2)
```

	Discrimination (point estimate)			
Component	Environmental	Social	cor(Env)	cor(Soc)
Administrative spending on active labour policy per population		0.0103		0.85
Environmental institutionalization	0.0132		0.573	
Information capacity	0.0711	0.0402	0.352	0.277
Professional bureaucratic remuneration	0.124	0.144	0.141	0.155
Professional criteria for appointment decisions in the state administration	0.717	0.685	0.801	0.731
Rigorous and impartial public administration	0.629	0.91	0.917	0.976
State authority over territory	0.317	0.262	0.435	0.424
Statistical Capacity score	0.43	0.336	-0.072	-0.0867
Tax revenue (% of GDP)	0.827	0.924	0.481	0.358
Taxes on income, profits and capital gains (% of revenue)	0.392	0.402	0.0808	0.0436
Taxes on international trade (% of revenue)	-0.0637	-0.068	-0.518	-0.433
Weberianess	-0.461	-0.406	0.0123	0.0124

Table 23.1: Discrimination parameters for a measurement model of implementation capacity, and the resulting correlations with the generated scores.

```
tb2 %>%
  save_kable(file = "TAB-discrimination-correlations.tex")
```

24
Explanatory model of performance

```
load("data-implementation_deficit.RData")

####### Get the specific values to process
d.id <- d.id %>%
  filter(Case %in% "Regular : Only sample countries : Not bounded : Original ratio") %>%
  # Sample countries
  # performance only available since 1980 and until 2012
  filter(Year >= 1980) %>%
  filter(Year <= 2012) %>%
  droplevels()


countries <- country.coverage <- as.character(levels(d.id$Country))
nC <- length(countries)
years <- range(d.id$Year)
```

24.1 Performance

```
perf <- foreign::read.dta("data-performance/Environmental_Performance_Chapter5.dta") %>%
  as_tibble() %>%
  select(Country = country, Year = year,
         General = PolGen100,
         Water = PolWat100,
         Mundane = Mundane100,
         Successfully = Success100,
         Specific1980 = LUPI82_1200,
         Specific2010 = LUPI07_1200) %>%
  mutate(Country = as.character(Country)) %>%
  mutate(Country = ifelse(Country == "UK", "United Kingdom", Country)) %>%
  mutate(Country = ifelse(Country == "US", "United States", Country)) %>%
  filter(Year %in% years[1]:years[2])
```

24.2 Covariates

World Development Indicators - Revenue.

```
load("data-performance/wdi-tax.RData")

tax.rev.l <- tax.rev %>%
  select(Country = country, tax.revenue = GC.TAX.TOTL.GD.ZS, Year = year) %>%
  group_by(Country) %>%
  summarize(tax.revenue = median(tax.revenue, na.rm = TRUE))
```

World Development Indicators:

- GDP per capita
- Trade

```
load("data-performance/wdi.RData")
wdi <- wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi <- subset(wdi, year >= 1980 & year <=2012)
```

```
wdi$country[wdi$country=="Korea, Rep."] <- "Korea, Republic of"
wdi <- subset(wdi, country %in% countries)

# GDP pc in Ireland is bad
ireland.wdi <- subset(wdi, country=="Ireland")

# So we use the combination of GDP and population
# to make a regression against the observed GDP per capita
# and impute accordingly.
ireland.wdi <- cbind(ireland.wdi, gdp.capita.div = ireland.wdi$gdp/ireland.wdi$population)
m.ireland <- lm(gdp.capita ~ gdp.capita.div, data=ireland.wdi)
wdi$gdp.capita[wdi$country=="Ireland" & is.na(wdi$gdp.capita)] <-
  predict(m.ireland, ireland.wdi)[1:(length(predict(m.ireland, ireland.wdi)) - (length(predict(m.ireland))))]

ireland.wdi <- subset(wdi, country=="Ireland")
ireland.wdi <- cbind(ireland.wdi, gdp.capita.div = ireland.wdi$gdp/ireland.wdi$population)
m.ireland <- lm(gdp.capita ~ year, data=ireland.wdi)
wdi$gdp.capita[wdi$country=="Ireland" & is.na(wdi$gdp.capita)] <-
  predict(m.ireland, ireland.wdi)[1:(length(predict(m.ireland, ireland.wdi)) - (length(predict(m.ireland))))]

# Switzerland is not so bad, but still problematic until 1979.
# But the procedure does not work, because GDP is also missing.
# So a simple imputation based on evolution over time is performed.
switzerland.wdi <- subset(wdi, country=="Switzerland")
switzerland.wdi <- cbind(switzerland.wdi, gdp.capita.div = switzerland.wdi$gdp/switzerland.wdi$population)
m.switzerland <- lm(gdp.capita ~ year, data=switzerland.wdi)
wdi$gdp.capita[wdi$country=="Switzerland" & is.na(wdi$gdp.capita)] <-
  predict(m.switzerland, switzerland.wdi)[1:(length(predict(m.switzerland, switzerland.wdi)) - (length(predict

#New Zealand only misses 1976' GDP per capita,
# so the same procedure than with Switzerland is used.
newzealand.wdi <- subset(wdi, country=="New Zealand")
newzealand.wdi <- cbind(newzealand.wdi, gdp.capita.div = newzealand.wdi$gdp/newzealand.wdi$population)
m.newzealand <- lm(gdp.capita ~ year, data=newzealand.wdi)
wdi$gdp.capita[wdi$country=="New Zealand" & is.na(wdi$gdp.capita)] <-
  predict(m.newzealand, newzealand.wdi)[1:(length(predict(m.newzealand, newzealand.wdi)) - (length(predict(m.n


switzerland.wdi <- subset(wdi, country=="Switzerland")
m.switzerland <- lm(trade ~ year, data=switzerland.wdi)
wdi$trade[wdi$country=="Switzerland" & is.na(wdi$trade)] <-
  predict(m.switzerland, switzerland.wdi)[1:(length(predict(m.switzerland, switzerland.wdi)) - (length(predict

# GDP per capita growth
# Another way at looking at resources, is to calculate
# how many times is the overall wealth per capita at the
# end of the period compared to the beginning.
wdi.gdp.capita.ratio <- subset(wdi[,c("country", "year", "gdp.capita")], year==min(year) | year==max(year))
wdi.gcr.w <- wdi.gdp.capita.ratio %>%
  spread(year, gdp.capita)
wdi.gcr.w <- cbind(wdi.gcr.w, gdpc.ratio=wdi.gcr.w$`2012`/wdi.gcr.w$`1980`)


# Data is averaged by country through all years.
wdi.c <- wdi %>%
  gather(variable, value, -country, -year) %>%
  group_by(country, variable) %>%
  summarize(m = median(value, na.rm = TRUE)) %>%
  ungroup()


# Include GDP per capita growth, ratio
wdi.l <- wdi.gcr.w %>%
  select(country, gdpc.ratio) %>%
  mutate(variable = "gdpc.ratio") %>%
  rename(m = gdpc.ratio) %>%
  select(country, variable, m) %>%
  bind_rows(wdi.c)
```

V-Dem:

- Deliberative democracy index

```
load("data-performance/v_dem-ra-1950_2018.RData") # loads vdem
```

Government effectiveness. Data retrieved manually from the World Bank page on Governance indicators. Only the "Government Effectiveness: Estimation" is used. The data is only available between 1996 and 2005.

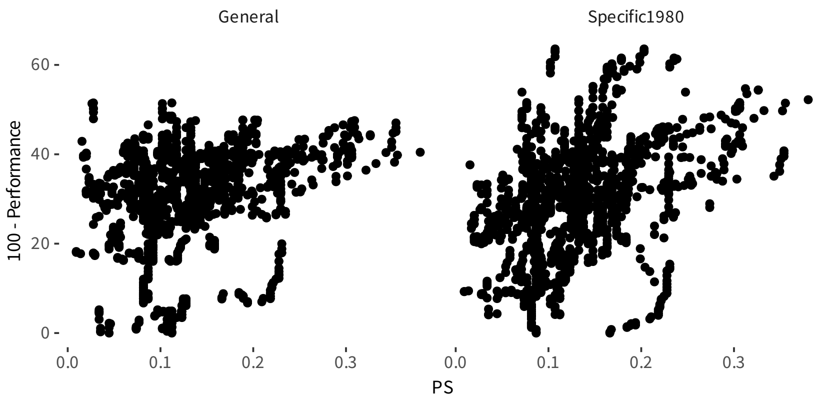
```
load("data-performance/government_effectiveness.RData")

gov.eff.original <- gov.eff
gov.eff.original$country <- as.character(gov.eff.original$country)
gov.eff.original$country[gov.eff.original$country=="Korea, Rep."] <- "Korea, Republic of"
gov.eff.original$country <- as.character(gov.eff.original$country)

gov.eff.l <- gov.eff.original %>%
  rename(gov.eff = value) %>%
  gather(variable, value, -country, -year) %>%
  group_by(country, variable) %>%
  summarize(m = median(value, na.rm = TRUE))
```

Political constraints

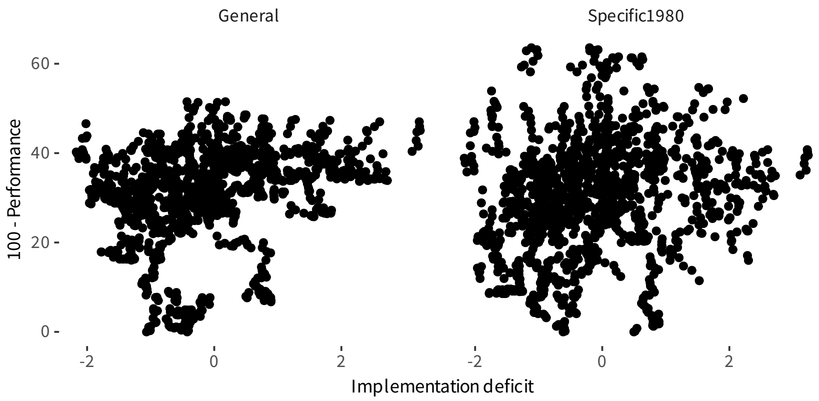
```
load("data-performance/polcon2017.RData") # loads polcon
```

For the "Green parties", data comes from @manifesto:2013. It provides dates of elections as well as shares of seats of several families of parties. We generate two indicators for the "green" and "socialist" ideology of the countries, with a weighted average of the proportion of seats and the duration of each legislature.

```
load("data-performance/cpm-consensus.RData")
cpm$country <- as.character(cpm$country)
cpm$country[cpm$country=="Korea"] <- "Korea, Republic of"
cpm$country[cpm$country=="Great Britain"] <- "United Kingdom"
cpm <- subset(cpm, country %in% countries)
cpm$country <- factor(cpm$country)
cpm <- subset(cpm, date>="1970-01-01" & date<="2012-12-31")

# Take only Green parties and Socialist=social democrats + communists
cpm$family <- as.character(cpm$family)
cpm$family[cpm$family=="Social democratic"] <- "Socialist"
cpm$family[cpm$family=="Communist"] <- "Socialist"
cpm <- subset(cpm, family=="Green" | family=="Socialist")
cpm$family <- factor(cpm$family)

# Aggregate duplications in Socialist
cpm <- cpm %>%
  group_by(country, date, family) %>%
  summarize(p.seats=sum(p.seats))

# Calculate the weighted means.
# Unfortunately, a ddply approach would be too complicated and a loop solves it quite quickly.
families <- c("Green", "Socialist")
wmsf <- data.frame(country=countries, Green=NA, Socialist=NA)
for (C in 1:nC) {
  for (F in 1:length(families)) {
    series <- subset(cpm, country==countries[C] & family==families[F])[,c(2, 4)]
    series <- series[order(series$date),]
    #v <- weighted.mean(series$p.seats, diff(c(as.Date("1976-01-01"), series$date)))
    v <- weighted.mean(series$p.seats, as.numeric(diff(c(as.Date("1980-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}

perf %>%
  select(Country, Year, General, Specific1980) %>%
  pivot_longer(-c(Country, Year), names_to = "Indicator", values_to = "Performance") %>%
  left_join(select(d.id, Country, Year, PS, `Implementation deficit`)) %>%
  ggplot(aes(x = PS, y = 100 - Performance)) +
  geom_point() +
  facet_grid(~ Indicator)
```




```
perf %>%
  select(Country, Year, General, Specific1980) %>%
  pivot_longer(-c(Country, Year), names_to = "Indicator", values_to = "Performance") %>%
  left_join(select(d.id, Country, Year, PS, `Implementation deficit`)) %>%
  ggplot(aes(x = `Implementation deficit`, y = 100 - Performance)) +
  geom_point() +
  facet_grid(~ Indicator)
```



Save and arrange for analysis.

```
# Add fake countries
d.id.fake.countries <- expand_grid(
  Sector = "Environmental",
  PS = seq(min(d.id$PS, na.rm = TRUE),
           max(d.id$PS, na.rm = TRUE),
           length.out = 5),
  `Implementation deficit` = seq(min(d.id$`Implementation deficit`, na.rm = TRUE),
                                 max(d.id$`Implementation deficit`, na.rm = TRUE),
                                 length.out = 5)) %>%
  mutate(Country = paste0("Z-", sprintf("%03d", 1:n()))) %>%
  expand_grid(Year = years[1]:years[2])
fake.countries <- unique(d.id.fake.countries$Country)
d.year.fake.countries <- expand_grid(Year = years[1]:years[2], Country = fake.countries)


d.perf <- perf %>%
  # Delete Korea
  filter(!Country == "Korea, Republic of") %>%
  # Delete Years for which we don't have data
  # Create the average between General and Specific 1980
  mutate("AvGralSpec80" = (General + Specific1980) / 2) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  group_by(Indicator) %>%
  mutate(Performance = std1(Performance)) %>%
  mutate(Performance = -Performance) %>%
  ungroup() %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Specific1980", "AvGralSpec80")) %>%
  droplevels()
d.perf <- bind_rows(d.perf,
  expand_grid(Country = fake.countries,
```

```
              Year = years[1]:years[2],
              Indicator = unique(d.perf$Indicator),
              Performance = NA))


Y <- reshape2::acast(d.perf, Indicator ~ Country ~ Year, value.var = "Performance")
nS <- dim(Y)[1]
nC <- dim(Y)[2]
nY <- dim(Y)[3]



country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
id.fake.countries <- which(str_detect(country.label, "^Z-"))
id.real.countries <- which(!str_detect(country.label, "^Z-"))

indicator.label <- dimnames(Y)[[1]]
sector.label <- dimnames(Y)[[2]]
nI <- length(indicator.label)

year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(as.numeric(year.label))
nY <- length(year.label)



# Function to assign zeros to the fake countries (mean value)
zero.fk <- function(x, id = id.fake.countries) { # zero to fake countries
  x[id] <- 0
  return(x)
}

source("data-performance/get-eu_time.R") # generates eu.ms


gap <- d.id %>%
  bind_rows(d.id.fake.countries) %>%
  select(Sector, Country, Year, `Implementation deficit`) %>%
  filter(Country %in% country.label) %>%
  filter(Sector == "Environmental") %>%
  select(-Sector) %>%
  filter(Year >= 1980 & Year <= 2012) %>%
  droplevels() %>%
  reshape2::acast(Country ~ Year, value.var = "Implementation deficit")
if ( length(which(!dimnames(gap)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")


GDPpc <- wdi %>%
  select(country, year, gdp.capita) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country = Country, year = Year), gdp.capita = 0)) %>%
  reshape2::acast(country ~ year, value.var = "gdp.capita")
if ( length(which(!dimnames(GDPpc)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

load("data-performance/wdi-gdpgrowth.RData") # gdp.growth
gdp.growth <- gdp.growth %>%
  select(country, year, gdp.growth = NY.GDP.MKTP.KD.ZG) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country = Country, year = Year), gdp.growth = 0)) %>%
  reshape2::acast(country ~ year, value.var = "gdp.growth")
if ( length(which(!dimnames(gdp.growth)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

load("data-performance/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(urban = std(urban)) %>%
```

```
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country = Country, year = Year), urban = 0)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

load("data-performance/wdi-industry.RData") # industry
industry <- industry %>%
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(industry = std(industry)) %>%
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country = Country, year = Year), industry = 0)) %>%
  reshape2::acast(country ~ year, value.var = "industry")
if ( length(which(!dimnames(industry)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
industry.means <- apply(industry, 1, mean, na.rm = TRUE)

trade <- wdi %>%
  select(country, year, trade) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(trade = std(trade)) %>%
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country = Country, year = Year), trade = 0)) %>%
  reshape2::acast(country ~ year, value.var = "trade")
if ( length(which(!dimnames(trade)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

deliberation <- vdem %>%
  mutate(Country = as.character(Country)) %>%
  mutate(Country = ifelse(Country == "United States of America", "United States", Country)) %>%
  mutate(Country = ifelse(Country == "Germany (RDA)", "Germany", Country)) %>%
  filter(Democracy == "Deliberative") %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1980 & Year <= 2012) %>%
  mutate(value = std(value)) %>%
  bind_rows(
    expand_grid(d.year.fake.countries, value = 0)) %>%
  select(Country, Year, value) %>%
  reshape2::acast(Country ~ Year, value.var = "value")
if ( length(which(!dimnames(deliberation)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

constraints <- polcon %>%
  mutate(country.polity = as.character(country.polity)) %>%
  mutate(country.polity = ifelse(country.polity == "Germany West", "Germany", country.polity)) %>%
  filter(country.polity %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(polcon = std(polcon)) %>%
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country.polity = Country, year = Year), polcon = 0)) %>%
  select(country.polity, year, polcon) %>%
  reshape2::acast(country.polity ~ year, value.var = "polcon")
if ( length(which(!dimnames(constraints)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")


gov.eff <- expand.grid(country = country.label, year = year.label.numeric) %>%
  left_join(gov.eff.original) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2012) %>%
  mutate(value = std(value)) %>%
  bind_rows(
    expand_grid(rename(d.year.fake.countries, country = Country, year = Year), polcon = 0)) %>%
  select(country, year, value) %>%
  reshape2::acast(country ~ year, value.var = "value")
if ( length(which(!dimnames(gov.eff)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")


portfolio.size <- d.id %>%
  bind_rows(d.id.fake.countries) %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1980 & Year <= 2012) %>%
  mutate(Size = std(logit(PS))) %>%
  ungroup() %>%
  reshape2::acast(Country ~ Year, value.var = "Size")
if ( length(which(!dimnames(portfolio.size)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
```

```
eu <- expand.grid(country = country.label, year = 1958:2020) %>%
  as_tibble() %>%
  mutate(eu = 0) %>%
  left_join(eu.ms, by = c("country" = "ms")) %>%
  mutate(eu = ifelse(year == ms.y, 1, eu)) %>%
  mutate(eu = ifelse(is.na(eu), 0, eu)) %>%
  group_by(country) %>%
  arrange(country, year) %>%
  mutate(eu = cumsum(eu)) %>%
  ungroup() %>%
  select(country, year, eu) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  reshape2::acast(country ~ year, value.var = "eu")
if ( length(which(!dimnames(eu)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")


DJ <- list(
  Y = unname(Y),
  gap = unname(gap),
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  gdp.growth = unname(gdp.growth),
  urban = unname(urban),
  industry = unname(industry),
  industry.means = unname(industry.means),
  deliberation = unname(deliberation),
  constraints = unname(constraints),
  gov.eff = unname(gov.eff),
  gov.eff.mean.observed = unname(apply(gov.eff, 1, mean, na.rm = TRUE)),
  portfolio.size = unname(portfolio.size),
  eu = unname(eu),
  id.real.countries = id.real.countries,
  id.fake.countries = id.fake.countries,
  nC = nC,
  nI = nI,
  nY = nY)

nC    # Number of countries
nS    # Number of sectors
years  # Range of years
```

24.3 Model

```
M <- "m-p-000"
M.lab <- "Baseline"
m <- "model {
  #
  # Data part at the observational level
  #
  for (i in 1:nI) {
    for (c in 1:nC) {
      for (t in 2:nY) {
        Y[i,c,t] ~ dnorm(mu[i,c,t], tau[i])
        mu[i,c,t] <- alpha[i]
                    + delta[i,c] * Y[i,c,t-1]
                  + beta[1,i] * GDPpc[c,t-1]
                  + beta[2,i] * trade[c,t-1]
                  + beta[3,i] * eu[c,t-1]
                  + beta[5,i] * gdp.growth[c,t-1]
                  + beta[6,i] * urban[c,t-1]
                  + beta[7,i] * industry[c,t-1]
                  + theta[1,i] * portfolio.size[c,t]
                  + theta[2,i] * gap[c,t]
                  + theta[3,i] * portfolio.size[c,t] * gap[c,t]
                  + rho[i,c] * (Y[i,c,t-1] - mu[i,c,t-1] )
        resid[i,c,t] <- Y[i,c,t] - mu[i,c,t]
      }
      Y[i,c,1] ~ dnorm(mu[i,c,1], tau[i])
      mu[i,c,1] <- alpha[i]
                  + beta[1,i] * GDPpc[c,1]
                  + beta[2,i] * trade[c,1]
```

```
                + beta[3,i] * eu[c,1]
                + beta[5,i] * gdp.growth[c,1]
                + beta[6,i] * urban[c,1]
                + beta[7,i] * industry[c,1]
                + theta[1,i] * portfolio.size[c,1]
                + theta[2,i] * gap[c,1]
                + theta[3,i] * portfolio.size[c,1] * gap[c,1]
      rho[i,c] ~ dunif(-1, 1)
      delta[i,c] ~ dunif(0, 1)
      resid[i,c,1] <- Y[i,c,1] - mu[i,c,1]
    }
    tau[i] ~ dgamma(0.001, 0.001)
    sigma[i] <- 1 / sqrt(tau[i])

    #
    # Priors for variance component
    #
    lambda[1,i] ~ dnorm(0, 10^-2)
    lambda[2,i] ~ dnorm(0, 10^-2)

    #
    # Priors for the intercept
    #
    #alpha[i] ~ dunif(0, 100)
    alpha[i] ~ dnorm(0, 10^-2)

    #
    # Priors for the control variables
    #
    for (b in 1:8) {
      beta[b,i] ~ dnorm(0, 5^-2)
    }

    #
    # Priors for main effects
    #
    for (t in 1:3) {
      theta[t,i] ~ dnorm(0, 5^-2)
    }

    #
    # Priors for varying intercepts
    #
    for (c in id.real.countries) {
      gamma[i,c] ~ dnorm(0, 1^-2)
      #gamma[i,c] ~ dt(0, 0.1^-2, 3)
    }
    for (c in id.fake.countries) {
      gamma[i,c] <- 0
    }
  }

  # Variance component, intercepts by country
  for (i in 1:nI) {
    for (c in 1:nC) {
      lambda_c[i,c] ~ dt(0, 0.1^-2, 3)
    }
  }

  #
  # Missing data
  #
  for (c in 1:nC) {
    for (t in 1:nY) {
      industry[c,t] ~ dnorm(industry.means[c], 0.5^-2)
    }
  }
}"
write(m, file= paste("models/model-", M, ".bug", sep = ""))
par <- NULL
par <- c(par, "alpha", "beta", "theta", "sigma")
par <- c(par, "lambda", "lambda_c")
par <- c(par, "delta")
par <- c(par, "nu")
par <- c(par, "rho")
```

```
par <- c(par, "resid")
par <- c(par, "gamma")
# add expected values for fake countries
mu.fake <- expand_grid(i = 1:nI, c = id.fake.countries)
par.mu.fake.countries <- paste0("mu[", mu.fake$i, ",", mu.fake$c, ",1]")
par <- c(par, par.mu.fake.countries)
inits <- list(
  list(.RNG.name="base::Super-Duper", .RNG.seed=1),
  list(.RNG.name="base::Super-Duper", .RNG.seed=2),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=3),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=2))
```

Model in JAGS:

```
cat(str_remove_all(m, "#.+\n"))
```

```
## model {
##   #
##     #
##   for (i in 1:nI) {
##     for (c in 1:nC) {
##       for (t in 2:nY) {
##         Y[i,c,t] ~ dnorm(mu[i,c,t], tau[i])
##         mu[i,c,t] <- alpha[i]
##                    + delta[i,c] * Y[i,c,t-1]
##                   + beta[1,i] * GDPpc[c,t-1]
##                   + beta[2,i] * trade[c,t-1]
##                   + beta[3,i] * eu[c,t-1]
##                   + beta[5,i] * gdp.growth[c,t-1]
##                   + beta[6,i] * urban[c,t-1]
##                   + beta[7,i] * industry[c,t-1]
##                   + theta[1,i] * portfolio.size[c,t]
##                   + theta[2,i] * gap[c,t]
##                   + theta[3,i] * portfolio.size[c,t] * gap[c,t]
##                   + rho[i,c] * (Y[i,c,t-1] - mu[i,c,t-1] )
##         resid[i,c,t] <- Y[i,c,t] - mu[i,c,t]
##       }
##       Y[i,c,1] ~ dnorm(mu[i,c,1], tau[i])
##       mu[i,c,1] <- alpha[i]
##                   + beta[1,i] * GDPpc[c,1]
##                   + beta[2,i] * trade[c,1]
##                   + beta[3,i] * eu[c,1]
##                   + beta[5,i] * gdp.growth[c,1]
##                   + beta[6,i] * urban[c,1]
##                   + beta[7,i] * industry[c,1]
##                   + theta[1,i] * portfolio.size[c,1]
##                   + theta[2,i] * gap[c,1]
##                   + theta[3,i] * portfolio.size[c,1] * gap[c,1]
##       rho[i,c] ~ dunif(-1, 1)
##       delta[i,c] ~ dunif(0, 1)
##       resid[i,c,1] <- Y[i,c,1] - mu[i,c,1]
##     }
##     tau[i] ~ dgamma(0.001, 0.001)
##     sigma[i] <- 1 / sqrt(tau[i])
```

```
##
##
##      #
##          #
##      lambda[1,i] ~ dnorm(0, 10^-2)
##      lambda[2,i] ~ dnorm(0, 10^-2)
##
##      #
##          #
##          alpha[i] ~ dnorm(0, 10^-2)
##
##      #
##          #
##      for (b in 1:8) {
##        beta[b,i] ~ dnorm(0, 5^-2)
##      }
##
##      #
##          #
##      for (t in 1:3) {
##        theta[t,i] ~ dnorm(0, 5^-2)
##      }
##
##      #
##          #
##      for (c in id.real.countries) {
##        gamma[i,c] ~ dnorm(0, 1^-2)
##            }
##      for (c in id.fake.countries) {
##        gamma[i,c] <- 0
##      }
##    }
##
##      for (i in 1:nI) {
##      for (c in 1:nC) {
##        lambda_c[i,c] ~ dt(0, 0.1^-2, 3)
##      }
##    }
##
##    #
##      #
##    for (c in 1:nC) {
##      for (t in 1:nY) {
##        industry[c,t] ~ dnorm(industry.means[c], 0.5^-2)
##      }
##    }
## }
```

```
t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
```

```
  data = dump.format(DJ, checkvalid=FALSE),
  inits = inits,
  modules = "glm",
  n.chains = 3, adapt = 1e2, burnin = 5e3, sample = 5e2, thin = 10,
  monitor = par, method = "parallel", summarise = FALSE)
s <- as.mcmc.list(rj)
save(s, file = paste("sample-", M, ".RData", sep = ""))
proc.time() - t0

load(file = paste("sample-", M, ".RData", sep = ""))

ggmcmc(ggs(s, family = "alpha|beta|theta|delta|lambda|rho"),
       param_page = 10, file = paste("ggmcmc-full-", M, ".pdf", sep = ""))
```

24.4 Model results

```
L.alpha <- plab("alpha", list(Indicator = indicator.label))
S.alpha <- ggs(s, family = "alpha\\[", par_labels = L.alpha)
ggs_caterpillar(S.alpha) +
  ggtitle("Intercept")
```

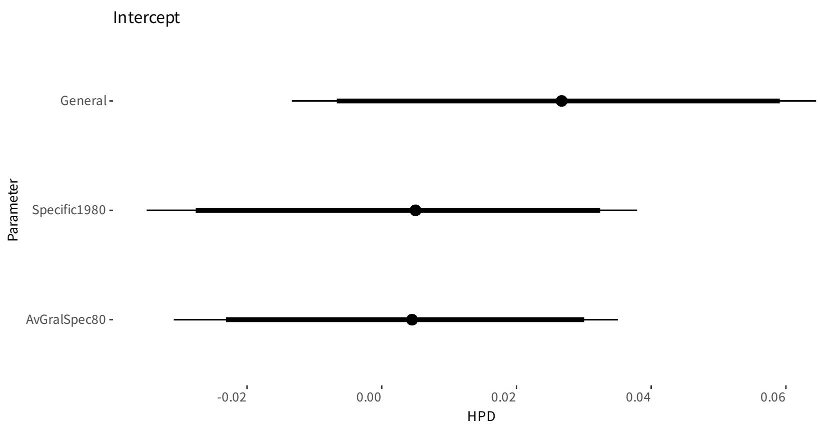


Figure 24.1: Main intercept.

Variance components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)", "Portfolio size"),
                                Sector = sector.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda) +
  ggtitle("Variance component")
```

Variance components (country varying intercepts).

```
L.lambda.c <- plab("lambda_c", list(Indicator = indicator.label, Country = country.label))
S.lambda.c <- ggs(s, family = "lambda_c\\[", par_labels = L.lambda.c) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.lambda.c, label = "Country") +
  facet_grid(~ Indicator) +
  ggtitle("Variance component (countries)")
```

Auto-regressive components.

```
L.rho <- plab("rho", list(Indicator = indicator.label, Country = country.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color=Indicator) +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)")
```

Lagged dependent variable.

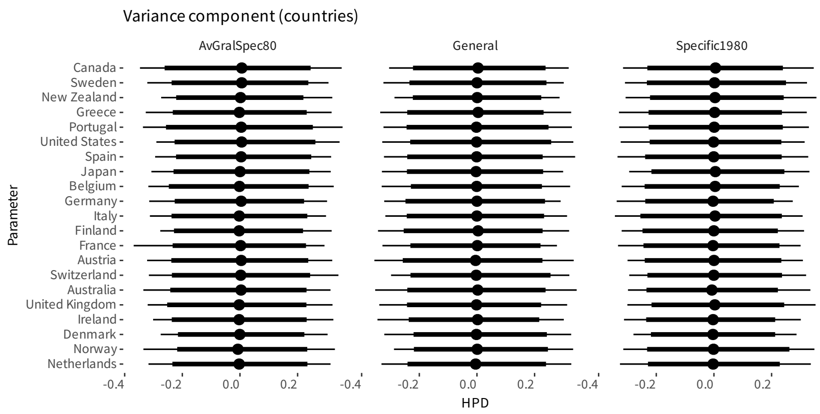


Figure 24.2: Variance component (country varying intercepts).

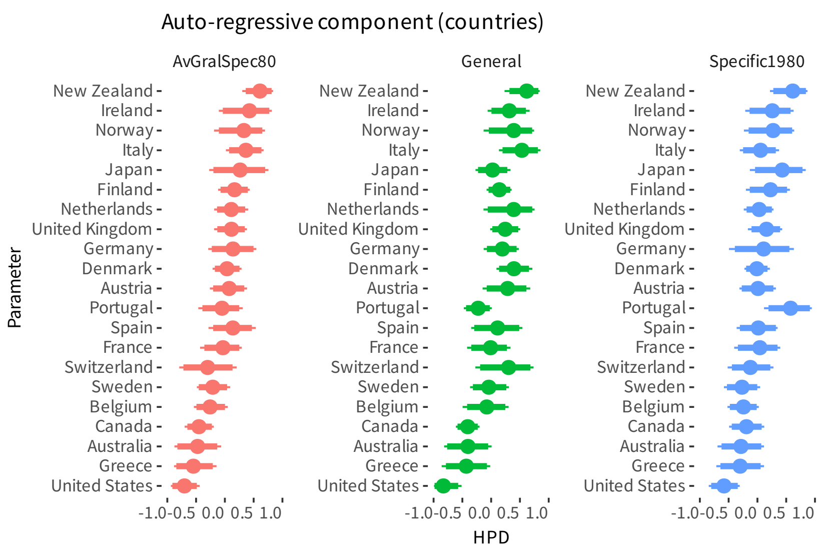


Figure 24.3: Auto-regressive component.

```
L.delta <- plab("delta", list(Indicator = indicator.label, Country = country.label))
S.delta <- ggs(s, family = "delta", par_labels = L.delta) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.delta, label = "Country") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color=Indicator) +
  ggtitle("Lagged dependent variable (countries)")
```

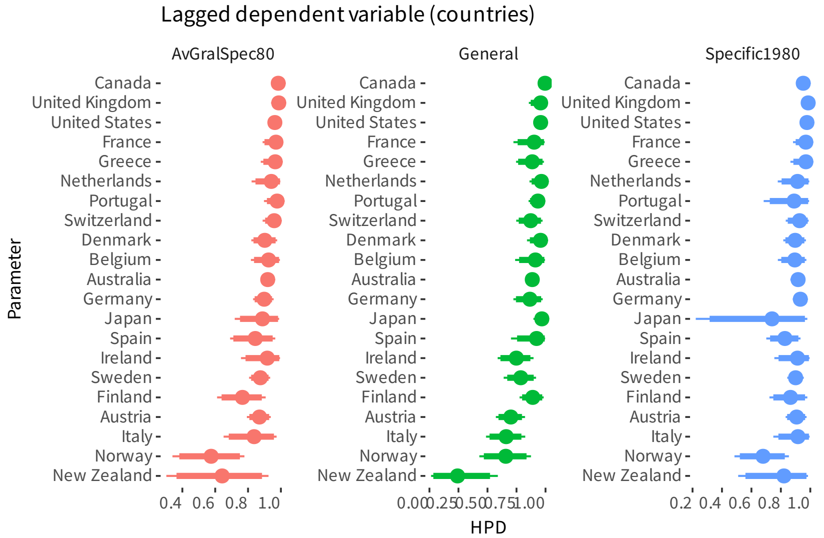
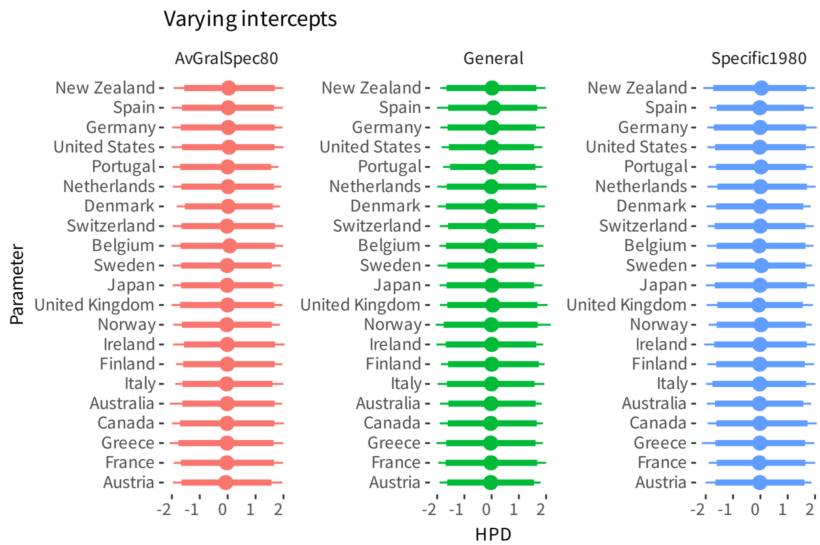


Figure 24.4: Lagged dependent variable.

Varying intercepts

```
L.gamma <- plab("gamma", list(Indicator = indicator.label, Country = country.label))
S.gamma <- ggs(s, family = "gamma", par_labels = L.gamma) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.gamma, label = "Country") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color=Indicator) +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Varying intercepts")
```




```
L.thetas <- plab("theta", list(Variable = c("Portfolio size", "Gap", "Portfolio size * Gap", "PS^2"),
                                Indicator = indicator.label))

L.betas <- plab("beta", list(Variable = c("GDP pc", "Trade", "EU", "Green",
                                           "GDP growth", "Urban", "Industry",
                                           "Consensus"),
```

```
                         Indicator = indicator.label))
L.betas <- bind_rows(L.thetas, L.betas)

S.betas <- ggs(s, family = "^beta\\[|^theta\\[", par_labels = L.betas) %>%
  filter(value != 0) #%>%
#  filter(Variable != "Portfolio size")
S.betas <- filter(S.betas, !Variable %in% c("Green", "Consensus"))

S.betas %>%
  group_by(Variable, Indicator) %>%
  summarize(median = median(value), sd = sd(value),
            `Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
    `Mean expected effect` = mean(value)) %>%
  kable()
```

Variable	Indicator	median	sd	Prob > 0	Prob < 0	Mean expected effect
Portfolio size	AvGralSpec80	0.1828	0.0577	0.9987	0.0013	0.1817
Portfolio size	General	-0.0703	0.0682	0.1580	0.8420	-0.0698
Portfolio size	Specific1980	0.3682	0.0724	1.0000	0.0000	0.3687
Gap	AvGralSpec80	0.0613	0.0148	1.0000	0.0000	0.0613
Gap	General	0.1353	0.0216	1.0000	0.0000	0.1353
Gap	Specific1980	0.0150	0.0180	0.7960	0.2040	0.0149
Portfolio size * Gap	AvGralSpec80	-0.0923	0.0291	0.0013	0.9987	-0.0922
Portfolio size * Gap	General	-0.0402	0.0280	0.0753	0.9247	-0.0399
Portfolio size * Gap	Specific1980	-0.0957	0.0370	0.0073	0.9927	-0.0954
GDP pc	AvGralSpec80	0.0518	0.0209	0.9913	0.0087	0.0519
GDP pc	General	0.1154	0.0286	1.0000	0.0000	0.1152
GDP pc	Specific1980	-0.0080	0.0254	0.3847	0.6153	-0.0080
Trade	AvGralSpec80	-0.0306	0.0195	0.0520	0.9480	-0.0308
Trade	General	0.0230	0.0248	0.8140	0.1860	0.0227
Trade	Specific1980	-0.0337	0.0208	0.0573	0.9427	-0.0333
EU	AvGralSpec80	0.1186	0.0246	1.0000	0.0000	0.1184
EU	General	0.1047	0.0276	1.0000	0.0000	0.1046
EU	Specific1980	0.0762	0.0264	0.9973	0.0027	0.0770
GDP growth	AvGralSpec80	-0.0622	0.0139	0.0000	1.0000	-0.0622
GDP growth	General	-0.0596	0.0146	0.0000	1.0000	-0.0597
GDP growth	Specific1980	-0.0508	0.0167	0.0007	0.9993	-0.0505
Urban	AvGralSpec80	0.0226	0.0186	0.8960	0.1040	0.0228
Urban	General	-0.0302	0.0258	0.1287	0.8713	-0.0300
Urban	Specific1980	0.0292	0.0227	0.9040	0.0960	0.0296
Industry	AvGralSpec80	0.2351	0.0131	1.0000	0.0000	0.2354
Industry	General	0.2256	0.0128	1.0000	0.0000	0.2257
Industry	Specific1980	0.2062	0.0152	1.0000	0.0000	0.2061

```
S.betas.gral <- S.betas %>%
  filter(Indicator == "AvGralSpec80") %>%
  mutate(Parameter = Variable)

ggs_caterpillar(S.betas.gral) +
  geom_vline(xintercept = 0, lty = 3)
```

24.5 Expected effects

```
L.mu <- plab("mu", list(Indicator = indicator.label,
                        Country = country.label,
```

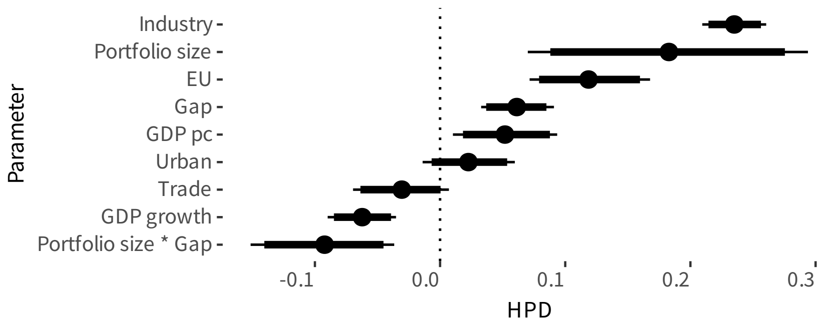


Figure 24.5: Slopes for the effects of control variables on performance (Global performance indicator).

```
                          Year = year.label))

S.mu <- ggs(s, family = "^mu\\[", par_labels = L.mu)
ci.mu <- ci(S.mu) %>%
  mutate(Year = as.integer(as.character(Year))) %>%
  left_join(d.id.fake.countries)
```

Marginal effects over portfolio size and implementation deficit

```
me.id <- S.mu %>%
  select(Iteration, Chain, Indicator, Country, Year, value) %>%
  mutate(Year = as.integer(as.character(Year))) %>%
  left_join(d.id.fake.countries) %>%
  filter(PS %in% c(
           min(PS, na.rm = TRUE),
           max(PS, na.rm = TRUE))) %>%
  mutate(PSextreme = ifelse(PS == min(PS, na.rm = TRUE), "Min", "Max")) %>%
  select(-c(Country, Year, Sector, PS)) %>%
  pivot_wider(names_from = PSextreme, values_from = value) %>%
  mutate(ME = Max - Min) %>% # marginal effect for the full range
  select(-c(Max, Min)) %>%
  group_by(Indicator, `Implementation deficit`) %>%
  summarize(ME = quantile(ME, c(0.05, 0.5, 0.95)), q = c("Low", "median", "High")) %>%
  pivot_wider(names_from = q, values_from = ME)

ggplot(me.id, aes(x = `Implementation deficit`, y = median,
             color = Indicator, fill = Indicator,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  ylab("Change in performance (AME)") +
  xlab("Implementation deficit") +
  geom_hline(yintercept = 0, lty = 3)


#
# Figure A18
#
me.id %>%
  filter(Indicator == "AvGralSpec80") %>%
  ggplot(aes(x = `Implementation deficit`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  ylab("Change in performance\n(AME)") +
  xlab("Burden capacity gap") +
  geom_hline(yintercept = 0, lty = 3)
```

24.6 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Indicator = indicator.label,
                             Country = country.label,
```

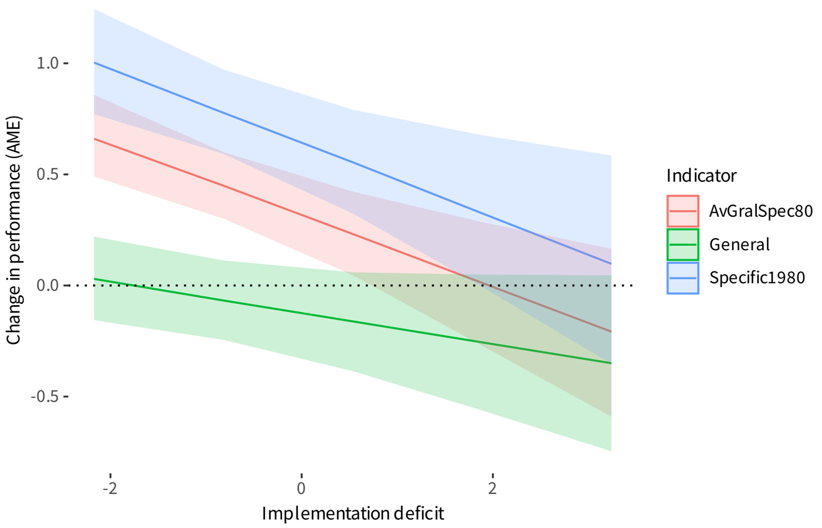


Figure 24.6: Expected effects over portfolio size.

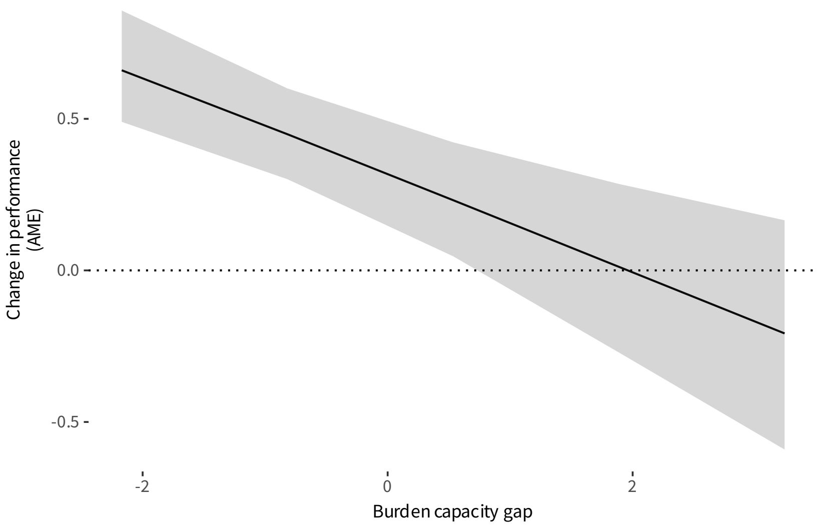


Figure 24.7: Effects of burden capacity gap on performance. Marginal effects of moving from minimum to maximum portfolio size at different levels of burden capacity gap.

```
                                      Year = year.label))

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Indicator = Var1, Country = Var2, Year = Var3, value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Indicator) %>%
  summarize(obs.sd = sd(value, na.rm = TRUE))

S.rsd <- ggs(s, family = "resid", par_labels = L.data) %>%
  filter(!str_detect(Country, "Z-")) %>%
  group_by(Iteration, Chain, Indicator) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
  geom_histogram(binwidth = 0.001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Indicator ~ .) +
  expand_limits(x = 0)
```

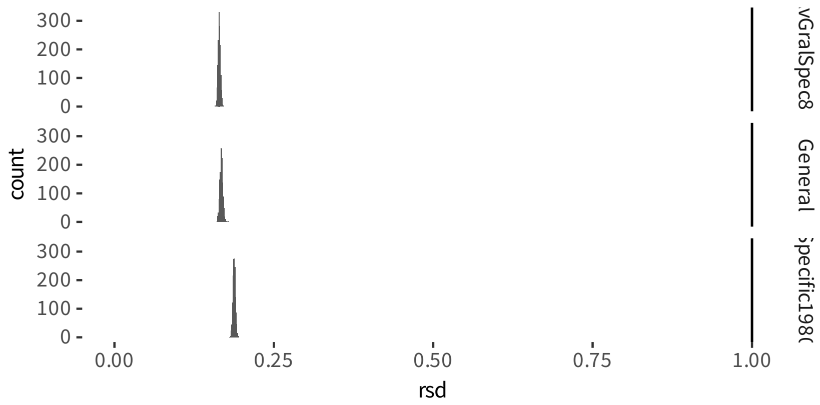


Figure 24.8: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

```
S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  group_by(Indicator) %>%
  summarize(Pseudo.R2 = mean((obs.sd - rsd) / obs.sd)) %>%
  kable()
```

Indicator	Pseudo.R2
AvGralSpec80	0.8356
General	0.8323
Specific1980	0.8122

```
r2s <- S.rsd %>%
  ungroup() %>%
  left_join(Obs.sd) %>%
  mutate(Covariate = factor("** Goodness of fit (R2)")) %>%
  group_by(Covariate) %>%
  mutate(value = (obs.sd - rsd) / obs.sd) %>%  # pseudo R2
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", signif(SD, 3), ")")) %>%
  select(Covariate, Coefficient, SD, `95% CI`)

tb <- S.betas %>%
  filter(Indicator %in% c("Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                                  levels = rev(c("Gap", "Portfolio size", "Portfolio size * Gap",
```

```
                                                "EU",
                                                "GDP pc", "GDP growth",
                                                "Industry", "Urban",
                                                "Trade")))) %>%
  rename(Covariate = Variable) %>%
  select(Iteration, Chain, Covariate, value) %>%
  group_by(Covariate) %>%
  summarize(Coefficient = median(value),
            SD = sd(value),
            CIlow = quantile(value, 0.025),
            CIhigh = quantile(value, 0.975)) %>%
  mutate(`95% CI` = paste0("[", signif(CIlow, 2), " : ", signif(CIhigh, 2), "]")) %>%
  mutate(SD = paste0("(", round(SD, 3), ")")) %>%
  mutate(id = 1) %>%
  bind_rows(r2s %>% mutate(id = 2)) %>%
  arrange(id, desc(abs(Coefficient))) %>%
  mutate(Coefficient = round(Coefficient, 2)) %>%
  select(Covariate, Coefficient, SD, `95% CI`)


tc <- "Model parameters. Outcome is environmental performance. Coefficient point estimates (median of the post

tb2 <- tb %>%
  ungroup() %>%
  kbl(format = "latex",
      caption = paste0("\\label{tab:tab-performance-000}", tc), label = NA,
      booktabs = TRUE,
      position = "ht") %>%
  kable_styling(font_size = 8) %>%
  pack_rows(paste0("y = Environmental performance (N=", (nC - length(id.fake.countries) ) * nY ,")"),  1, dim(

print(tb2)
```

Covariate	Coefficient	SD	95% CI
y = Environmental performance (N=693)			
Portfolio size	0.37	(0.072)	[0.23 : 0.51]
Industry	0.21	(0.015)	[0.18 : 0.23]
Portfolio size * Gap	-0.10	(0.037)	[-0.17 : -0.02]
EU	0.08	(0.026)	[0.028 : 0.13]
GDP growth	-0.05	(0.017)	[-0.083 : -0.017]
Trade	-0.03	(0.021)	[-0.075 : 0.0063]
Urban	0.03	(0.023)	[-0.012 : 0.074]
Gap	0.01	(0.018)	[-0.02 : 0.05]
GDP pc	-0.01	(0.025)	[-0.059 : 0.041]
** Goodness of fit (R2)	0.83	(0.0106)	[0.81 : 0.84]

Table 24.1: Model parameters. Outcome is environmental performance. Coefficient point estimates (median of the posterior distribution), SD refers to the standard deviation (uncertainty), and CI to the 95 percent credible interval.

```
tb2 %>%
  save_kable(file = "TAB-a26.tex")
```

Programming environment

```
cat('\\pagebreak')
```

```r
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: aarch64-unknown-linux-gnu (64-bit)
## Running under: Gentoo Linux
##
## Matrix products: default
## BLAS:   /usr/lib64/blas/openblas/libblas.so.3
## LAPACK: /usr/lib64/libopenblasp-r0.3.23.so;  LAPACK version 3.11.0
##
## locale:
##  [1] LC_CTYPE=ca_AD.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=ca_AD.UTF-8        LC_COLLATE=ca_AD.UTF-8
## locale:
##  [1] LC_CTYPE=ca_AD.UTF-8       LC_NUMERIC=C               LC_TIME=ca_AD.UTF-8
##  [4] LC_COLLATE=ca_AD.UTF-8     LC_MONETARY=ca_AD.UTF-8    LC_MESSAGES=ca_AD.UTF-8
##  [7] LC_PAPER=ca_AD.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=ca_AD.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Andorra
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggridges_0.5.4       tibble_3.2.1          forcats_1.0.0         scales_1.2.1
##  [5] corrplot_0.92        cowplot_1.1.1         PolicyPortfolios_0.3  stringr_1.5.0
##  [9] colorspace_2.1-0     ggrepel_0.9.3         GGally_2.1.2          ggmcmc_1.5.1.1
## [13] runjags_2.2.2-1.1    rjags_4-14            coda_0.19-4           ggthemes_4.2.4
## [17] extrafont_0.16.0.99  gridExtra_2.3         ggplot2_3.4.3         tidyr_1.3.0
## [21] dplyr_1.1.3          kableExtra_1.3.4.9000 tikzDevice_0.12.5     rmarkdown_2.24
## [25] knitr_1.43           colorout_1.2-2
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.4      xfun_0.40          lattice_0.21-8     vctrs_0.6.3       tools_4.3.1
##  [6] generics_0.1.3    parallel_4.3.1     fansi_1.0.4        cluster_2.1.4     pkgconfig_2.0.3
## [11] Matrix_1.6-1      RColorBrewer_1.1-3 webshot_0.5.5      filehash_2.4-5    lifecycle_1.0.3
## [16] ineq_0.2-13       farver_2.1.1       compiler_4.3.1     munsell_0.5.0     permute_0.9-7
## [21] htmltools_0.5.6   yaml_2.3.7         Rttf2pt1_1.3.0.99  pillar_1.9.0      extrafontdb_1.0
## [26] MASS_7.3-60       vegan_2.6-4        ggcorrplot_0.1.4.1 nlme_3.1-163      tidyselect_1.2.0
## [31] rvest_1.0.3       digest_0.6.33      stringi_1.7.12     reshape2_1.4.4    purrr_1.0.2
## [36] bookdown_0.35     labeling_0.4.3     splines_4.3.1      fastmap_1.1.1     grid_4.3.1
## [41] cli_3.6.1         magrittr_2.0.3     utf8_1.2.3         tufte_0.13        foreign_0.8-85
## [46] withr_2.5.0       httr_1.4.7         zoo_1.8-12         evaluate_0.21     viridisLite_0.4.2
## [51] mgcv_1.9-0        rlang_1.1.1        Rcpp_1.0.11        glue_1.6.2        xml2_1.3.5
## [56] svglite_2.1.1     rstudioapi_0.15.0  reshape_0.8.9      R6_2.5.1          plyr_1.8.8
## [61] systemfonts_1.0.4
```