XAVIER FERNÁNDEZ I MARÍN, CHRISTOPH KNILL, YVES STEINEBACH

STUDYING POLICY DESIGN QUALITY IN COMPARATIVE PERSPECTIVE

Contents

1 Introduction

This report presents the data analysis for "Studying Policy Design Quality in Comparative Perspective", published in the *American Review of Political Science*.

First it contains a brief chapter with the description of the data, and reproduces the figures in the article and the online appendix.

Then in contains two chapters on the analysis of environmental performance, one with the main model and another one with the robustness check from a variation of the main model.

Later, it contains four chapters on the analysis of portfolio diversity, one with the main model and three more with various robustness checks for different specifications.

Finally, one chapter compares the different robustness models of portfolio diversity against the main model.

2

Data description

```r
library(PolicyPortfolios)
data(consensus)
```

Show the portfolios of selected countries / years for illustration purposes.

```r
D <- consensus %>%
  filter(Sector == "Environmental") %>%
  droplevels() %>%
  pp_measures()

consensus.without.labels <- consensus %>%
  mutate(Target = as.factor(as.numeric(Target))) %>%
  mutate(Instrument = as.factor(as.numeric(Instrument)))

rM <- pp_array(filter(consensus, Sector == "Environmental" & Country == "France" & Year == 1976), return_ma

manual.aid.fr.1976 <- round(D$value[D$Country == "France" & D$Year == 1976 & D$Sector == "Environmental" &
manual.aid.fr.2005 <- round(D$value[D$Country == "France" & D$Year == 2005 & D$Sector == "Environmental" &
manual.aid.us.1976 <- round(D$value[D$Country == "United States" & D$Year == 1976 & D$Sector == "Environmen
manual.aid.us.2005 <- round(D$value[D$Country == "United States" & D$Year == 2005 & D$Sector == "Environmen

# France: growth in size and diversity
f1 <- pp_plot(droplevels(filter(consensus.without.labels, Sector == "Environmental")),
              id = list(Country = "France", Year = 1976),
              subtitle = FALSE, caption = "") +
  labs(subtitle = paste0("AID: ", manual.aid.fr.1976)) +
  theme(plot.subtitle = element_text(size = 3.0))
f2 <- pp_plot(droplevels(filter(consensus.without.labels, Sector == "Environmental")),
              id = list(Country = "France", Year = 2005),
              subtitle = FALSE, caption = "") +
  labs(subtitle = paste0("AID: ", manual.aid.fr.2005)) +
  theme(plot.subtitle = element_text(size = 3.0))
#grid.arrange(f1, f2, ncol = 2)

f3 <- pp_plot(droplevels(filter(consensus.without.labels, Sector == "Environmental")),
              id = list(Country = "United States", Year = 1976),
              subtitle = FALSE, caption = "") +
  labs(subtitle = paste0("AID: ", manual.aid.us.1976)) +
  theme(plot.subtitle = element_text(size = 3.0))
f4 <- pp_plot(droplevels(filter(consensus.without.labels, Sector == "Environmental")),
              id = list(Country = "United States", Year = 2005),
              subtitle = FALSE, caption = "") +
  labs(subtitle = paste0("AID: ", manual.aid.us.2005)) +
  theme(plot.subtitle = element_text(size = 3.0))
#grid.arrange(f1, f2, ncol = 2)

grid.arrange(f1, f2, f3, f4, ncol = 2)
```

Description of the distribution of Average Instrument Diversity.

```r
consensus %>%
  filter(Sector == "Environmental") %>%
```

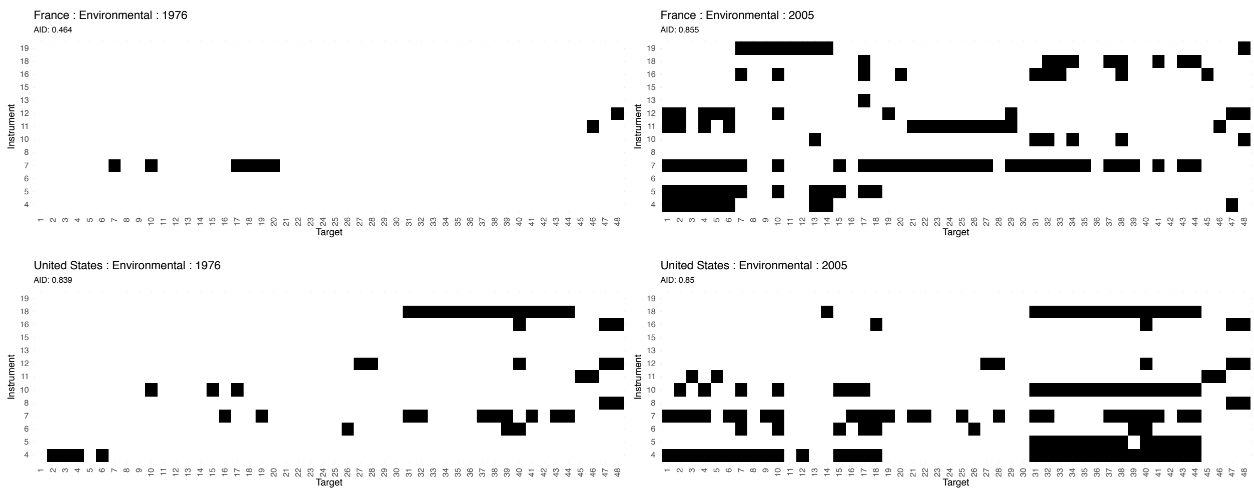

Figure 2.1: Example portfolios comparing Average Instrument Diversity. France: +size +diversity; United States: +size =diversity.

```
  droplevels() %>%
  pp_measures() %>%
  filter(Measure == "Div.aid") %>%
  select(Country, Year, Measure.label, value) %>%
  spread(Measure.label, value) %>%
  ggplot(aes(y = reorder(Country, `Diversity (Average Instrument Diversity)`, mean))) +
  geom_boxplot(aes(x = `Diversity (Average Instrument Diversity)`)) +
  expand_limits(x = c(0, 1)) +
  xlab("Average Instrument Diversity") +
  ylab("Country")
```

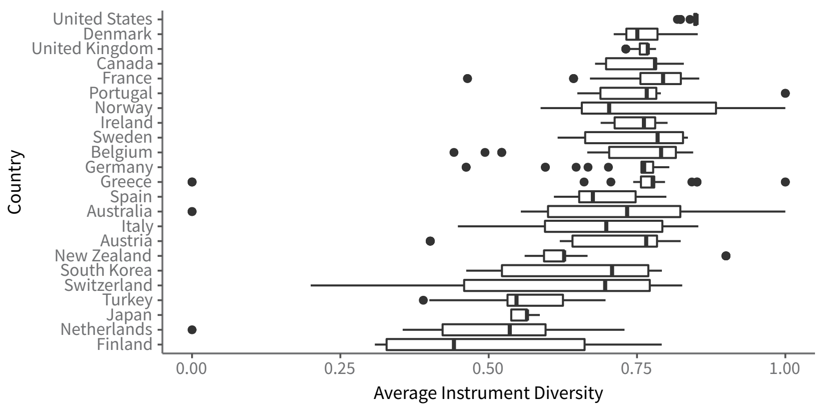

Figure 2.2: Descriptive statistics for portfolio diversity (Average Instrument Diversity). Environmental sector.

```
consensus %>%
  filter(Sector == "Environmental") %>%
  droplevels() %>%
  pp_measures() %>%
  filter(Measure == "Div.aid") %>%
ggplot(aes(x = Year, y = value)) +
  geom_line() +
  ylab("Diversity (Average Instrument Diversity") +
  facet_wrap(~ Country) +
  ggtitle("Diversity (Average Instrument Diversity)")
```

Several measures of diversity, compared

```
consensus %>%
  filter(Sector == "Environmental") %>%
  droplevels() %>%
```

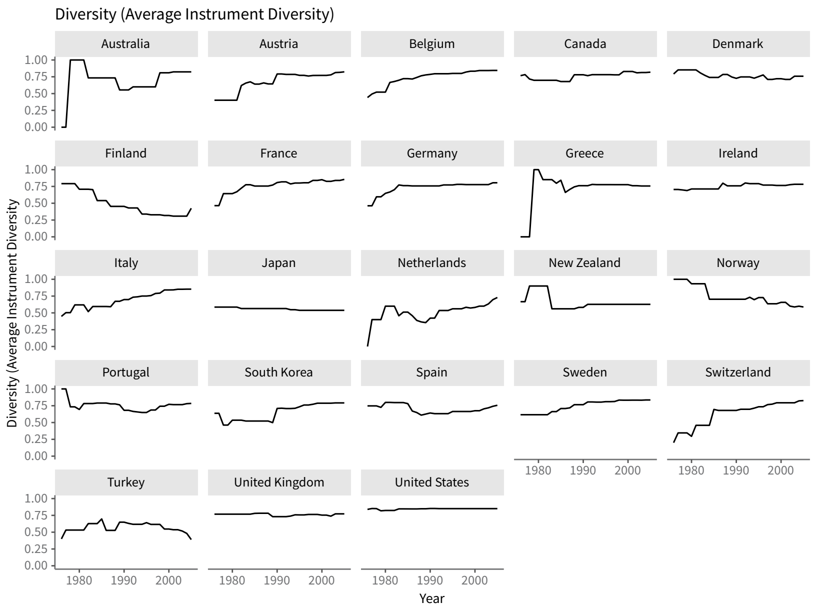

Figure 2.3: Temporal evolution of average instrument diversity, by country.

```
pp_measures() %>%
filter(Measure %in% c("Size", "Div.aid", "Div.gs", "C.eq", "Eq.sh")) %>%
mutate(Measure = Measure.label) %>%
select(Country, Sector, Year, Measure, value) %>%
spread(Measure, value) %>%
gather(Measure, Diversity, -c(Country, Sector, Year,
          `Diversity (Average Instrument Diversity)`,
          `Portfolio size`)) %>%
ggplot(aes(x = `Diversity (Average Instrument Diversity)`,
           y = Diversity,
           color = Measure,
           size = `Portfolio size`)) +
geom_point(alpha = 0.5) +
facet_grid(Measure ~ Sector)
```

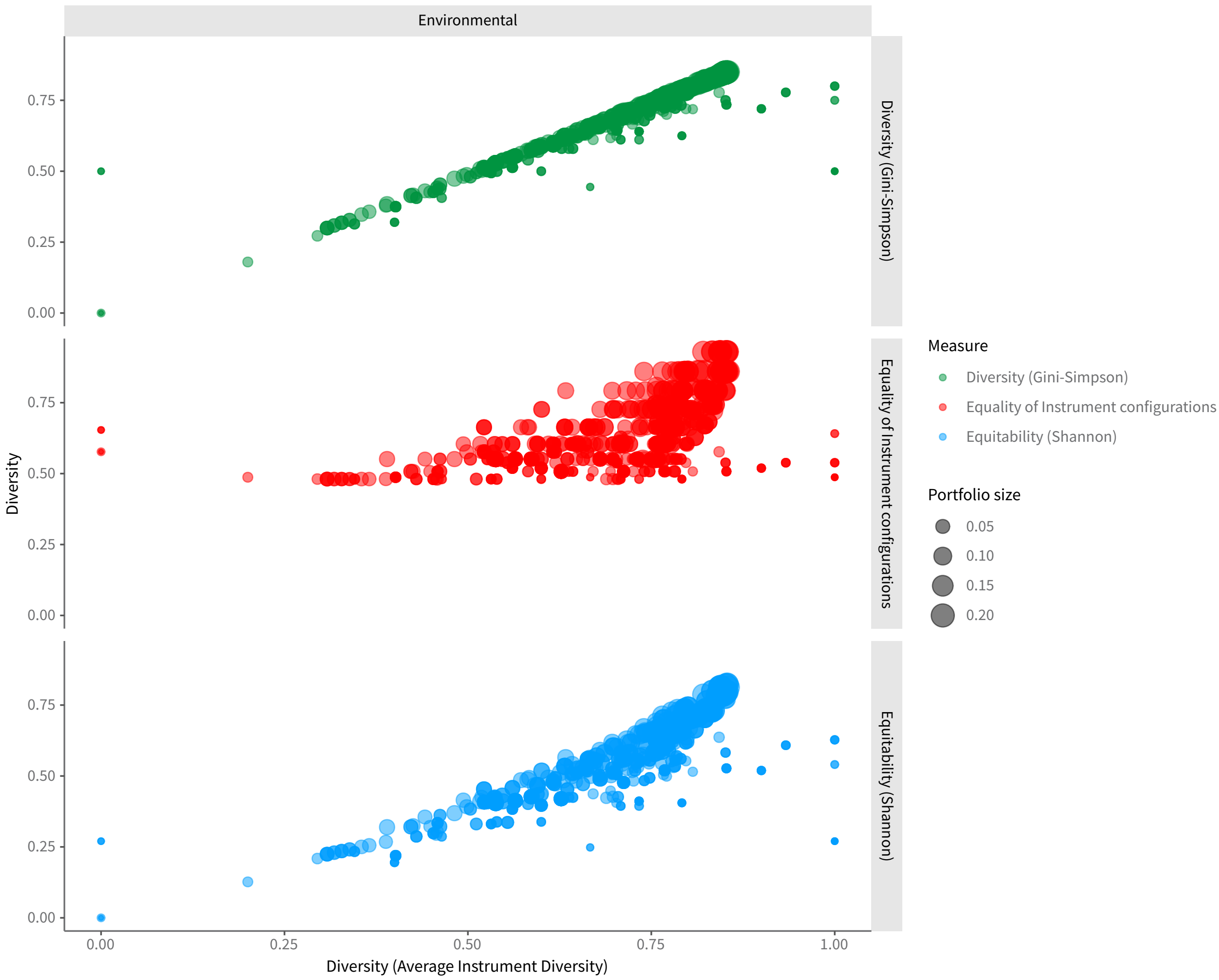

Figure 2.4: Average instrument diversity compared against other measures of diversity, by sector.

3

Explanatory model of performance

```
library(PolicyPortfolios)
data(consensus)

ca <- consensus %>%
  filter(Sector == "Environmental") %>%
  droplevels() %>%
  pp_measures() %>%
  filter(Measure %in% c("Div.aid", "Size")) %>%
  select(-Measure.label) %>%
  spread(Measure, value) %>%
  rename(Diversity = Div.aid)

countries <- country.coverage <- as.character(levels(ca$Country))
nC <- length(countries)
years <- range(ca$Year)
save(countries, years, country.coverage, file = "details.RData")
```

3.1 Performance

```
perf <- foreign::read.dta("jahn/PoEP_Replication_Data/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% 1980:2010)
```

3.2 Covariates

World Development Indicators - Revenue.

```
load("wdi/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

```
countries <- as.character(levels(ca$Country))
load("wdi/wdi.RData")
wdi <- wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi <- subset(wdi, year >= 1976 & year <= 2005)
wdi$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(pred

#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(


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(pred

# 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$`2005`/wdi.gcr.w$`1976`)


# 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("vdem/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("wgi/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("./polcon/polcon2017.RData") # loads polcon
```

For the "Green parties", data comes from Volkens (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("./manifesto/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 <= "2005-12-31")

# 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("1976-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}
```

Save and arrange for analysis.

```
d.perf <- perf %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("Korea, Republic of", "Turkey")) %>%
  # Delete Years for which we don't have data
  filter(Year >= 1980 & Year <= 2005) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Specific1980")) %>%
  droplevels()

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)

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("get-eu_time.R") # generates eu.ms

diversity <- ca %>%
  select(Sector, Country, Year, Diversity) %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  filter(Sector == "Environmental") %>%
  filter(Year >= 1980 & Year <= 2005) %>%
  select(-Sector) %>%
  droplevels() %>%
  reshape2::acast(Country ~ Year, value.var = "Diversity")
if ( length(which(!dimnames(diversity)[[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 <= 2005) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  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("wdi/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 <= 2005) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  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("wdi/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2005) %>%
  mutate(urban = std(urban)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

load("wdi/wdi-industry.RData") # industry
industry <- industry %>%
```

```r
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2005) %>%
  mutate(industry = std(industry)) %>%
  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 <= 2005) %>%
  mutate(trade = std(trade)) %>%
  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 <= 2005) %>%
  mutate(value = std(value)) %>%
  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 <= 2005) %>%
  mutate(polcon = std(polcon)) %>%
  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 <= 2005) %>%
  mutate(value = std(value)) %>%
  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 <- ca %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1980 & Year <= 2005) %>%
  mutate(Size = std(logit(Size))) %>%
  ungroup() %>%
  reshape2::acast(Country ~ Year, variable.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) %>%
```

```r
  reshape2::acast(country ~ year, value.var = "eu")
if ( length(which(!dimnames(eu)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

green.socialist <- cpm %>%
  ungroup() %>%
  mutate(calendar.year = as.numeric(format(date, "%Y"))) %>%
  mutate(calendar.month = as.numeric(format(date, "%m"))) %>%
  mutate(year = ifelse(calendar.month > 6, calendar.year + 1, calendar.year)) %>%
  # Manually delete some elections that are held too close
  # and assigned both to the same year
  # Delete the first one, as in case of instability it is unlikely
  # that they have passed legislation
  filter(!(country == "Japan" & date == as.Date("1979-10-07"))) %>%
  filter(!(country == "Greece" & date == as.Date("1989-11-05"))) %>%
  filter(!(country == "Denmark" & date == as.Date("1987-09-08"))) %>%
  #
  select(country, year, family, p.seats) %>%
  full_join(expand.grid(country = country.label,
                        year = (min(year.label.numeric)-5):max(year.label.numeric),
                        family = c("Socialist", "Green"))) %>%
  group_by(country, family) %>%
  arrange(country, family, year) %>%
  # A bit clumsy, but works
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  ungroup() %>%
  select(country, year, family, p.seats) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(p.seats = ifelse(is.na(p.seats), 0, p.seats)) %>%
  group_by(family) %>%
  mutate(p.seats = std(p.seats)) %>%
  ungroup() %>%
  unique() %>%
  reshape2::acast(family ~ country ~ year, value.var = "p.seats")
if ( length(which(!dimnames(green.socialist)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here
if (!dimnames(green.socialist)[[1]][1] == "Green" & sector.label[1] == "Environmental") stop("Ep! There is

# Use only the green dimension
green <- green.socialist[1,,]

DJ <- list(
  Y = unname(Y),
  diversity = unname(diversity),
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  gdp.growth = unname(gdp.growth),
  urban = unname(urban),
  industry = unname(industry),
  industry.means = 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),
  green = green,
  nC = nC,
  nI = nI,
  nY = nY)
nC   # Number of countries

## [1] 21

nS   # Number of sectors
```

```
## [1] 2
```

```
years  # Range of years
```

```
## [1] 1976 2005
```

3.3 Model

```
M <- "performance-tscs"
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,c,t])
        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] * diversity[c,t]
                  + rho[i,c] * (Y[i,c,t-1] - mu[i,c,t-1] )
        tau[i,c,t] <- 1 / sigma.sq[i,c,t]
        sigma.sq[i,c,t] <- exp(
                                  lambda_c[i,c])
        resid[i,c,t] <- Y[i,c,t] - mu[i,c,t]
      }
      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] * diversity[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]
    }

    #
    # Priors for variance component
    #
    lambda[1,i] ~ dnorm(0, 2^-2)
    lambda[2,i] ~ dnorm(0, 2^-2)

    #
    # Priors for the intercept
    #
    alpha[i] ~ dunif(0, 100)

    #
    # Priors for the control variables
    #
    for (b in 1:8) {
      beta[b,i] ~ dnorm(0, 1^-2)
    }

    #
    # Priors for main effects
    #
    for (t in 1:2) {
      theta[t,i] ~ dnorm(0, 1^-2)
    }
  }

  # 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], 0.05^-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")
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))

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 = 4, adapt = 1e2, burnin = 1e3, 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 = ""))

ggmcmc(ggs(s, family = "sigma|alpha|beta|lambda|nu|rho"),
       plot = c("traceplot", "crosscorrelation", "caterpillar"),
       param_page = 8, file = paste("ggmcmc-partial-", M, ".pdf", sep = ""))
```

3.4 Model results

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)
ggs_caterpillar(S.lambda.c, label = "Country") +
  facet_grid(~ Indicator) +
  ggtitle("Variance component (countries)")
```

Auto-regressive components.

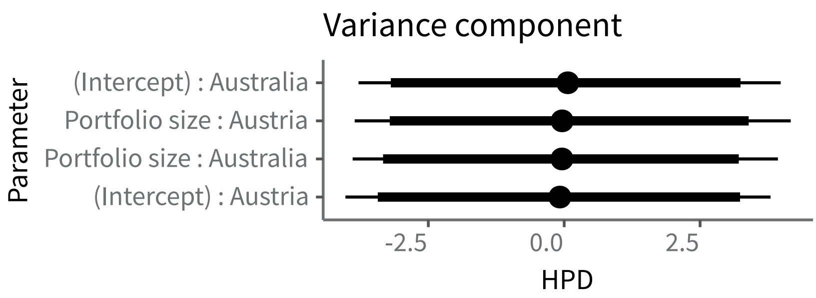

Figure 3.1: Variance component.

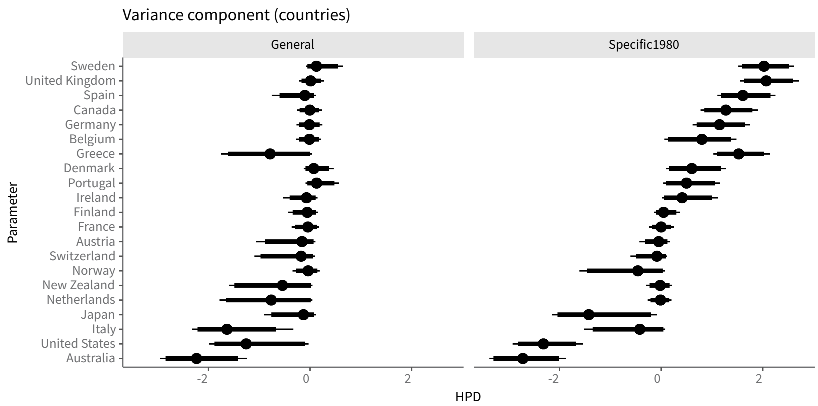

Figure 3.2: Variance component (country varying intercepts).

```
L.rho ← plab("rho", list(Indicator = indicator.label, Country = country.label))
S.rho ← ggs(s, family = "rho", par_labels = L.rho)
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)") +
  scale_colour_xfim()
```

Lagged dependent variable.

```
L.delta ← plab("delta", list(Indicator = indicator.label, Country = country.label))
S.delta ← ggs(s, family = "delta", par_labels = L.delta)
ggs_caterpillar(S.delta, label = "Country") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color=Indicator) +
  ggtitle("Lagged dependent variable (countries)") +
  scale_colour_xfim()
```

```
L.thetas ← plab("theta", list(Variable = c("Portfolio size", "Diversity"),
                              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)
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(),
```

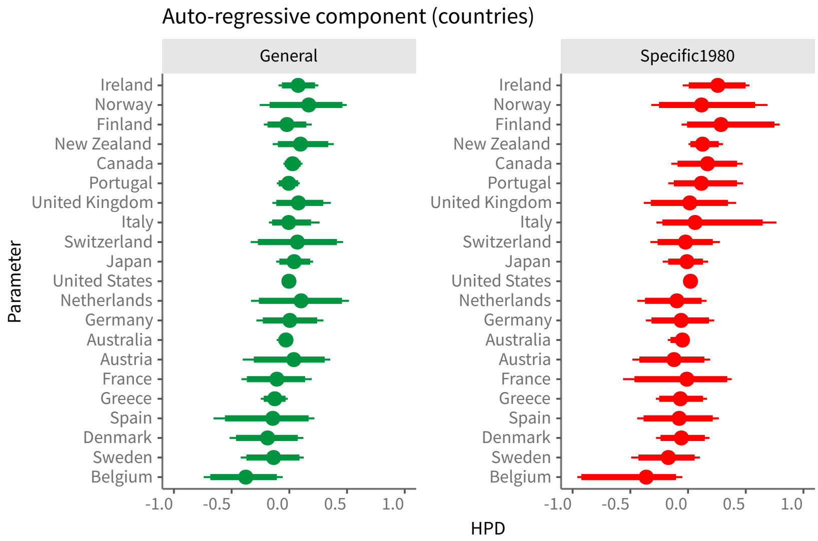

Figure 3.3: Auto-regressive component.

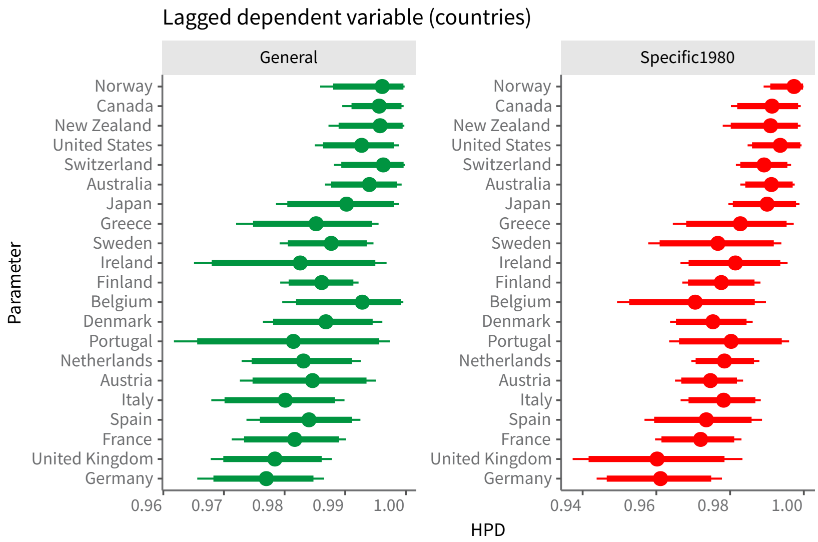

Figure 3.4: Lagged dependent variable.

```
    `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	General	-0.19	0.13	0.06	0.94	-0.19
Portfolio size	Specific1980	-0.24	0.12	0.02	0.98	-0.24
Diversity	General	0.62	0.24	1.00	0.00	0.64
Diversity	Specific1980	0.67	0.31	1.00	0.00	0.69
GDP pc	General	-0.60	0.19	0.00	1.00	-0.60
GDP pc	Specific1980	-0.64	0.21	0.00	1.00	-0.64
Trade	General	-0.24	0.26	0.18	0.82	-0.24
Trade	Specific1980	0.06	0.34	0.58	0.42	0.05
EU	General	0.45	0.19	0.99	0.01	0.45
EU	Specific1980	0.50	0.23	0.98	0.02	0.50
GDP growth	General	0.27	0.08	1.00	0.00	0.28
GDP growth	Specific1980	-0.04	0.08	0.28	0.72	-0.04
Urban	General	-0.23	0.29	0.21	0.79	-0.23
Urban	Specific1980	0.27	0.41	0.78	0.22	0.30
Industry	General	0.13	0.18	0.76	0.24	0.13
Industry	Specific1980	0.16	0.20	0.81	0.19	0.17

```
ggs_caterpillar(S.betas, label = "Variable") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color = Indicator) +
  geom_vline(xintercept = 0, lty = 3) +
  scale_colour_xfim()
```

Figure 3.5: Slopes with the effects of control and main variables on performance, by performance indicator.

```
S.betas %>%
  filter(Indicator %in% c("General", "Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                          levels = rev(c("Diversity", "Portfolio size",
                                         "EU",
                                         "GDP pc", "GDP growth",
                                         "Industry", "Urban",
                                         "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  facet_wrap(~ Indicator, scales="free") +
  geom_vline(xintercept = 0, lty = 3) #+

S.betas %>%
  filter(Indicator %in% c("General")) %>%
  mutate(Variable = factor(as.character(Variable),
                          levels = rev(c("Diversity",
                                         "Portfolio size",
                                         "EU",
                                         "GDP pc", "GDP growth",
                                         "Industry", "Urban",
                                         "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle("General performance")
```

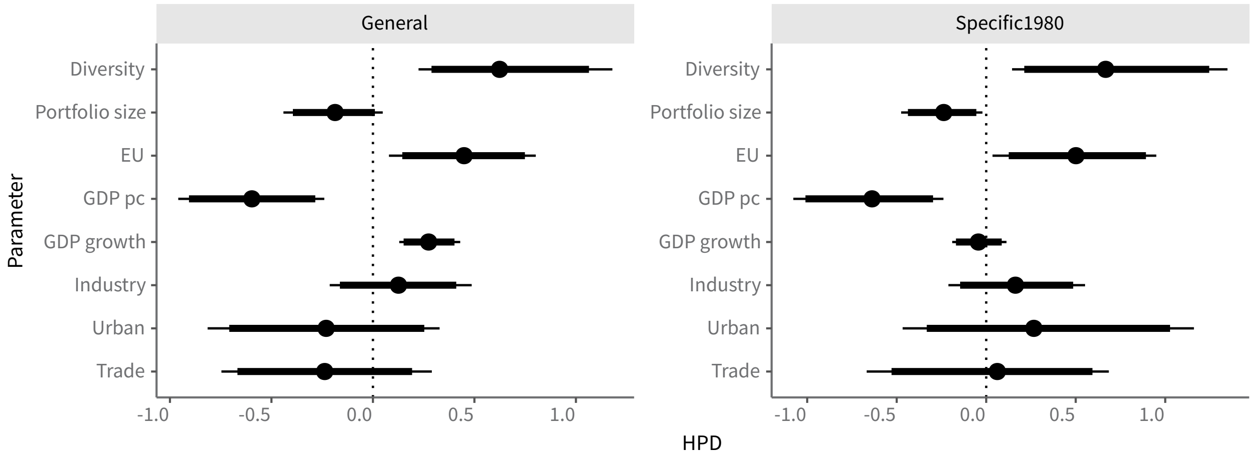

Figure 3.6: Slopes with the effects of control and main variables on performance, by performance indicator.

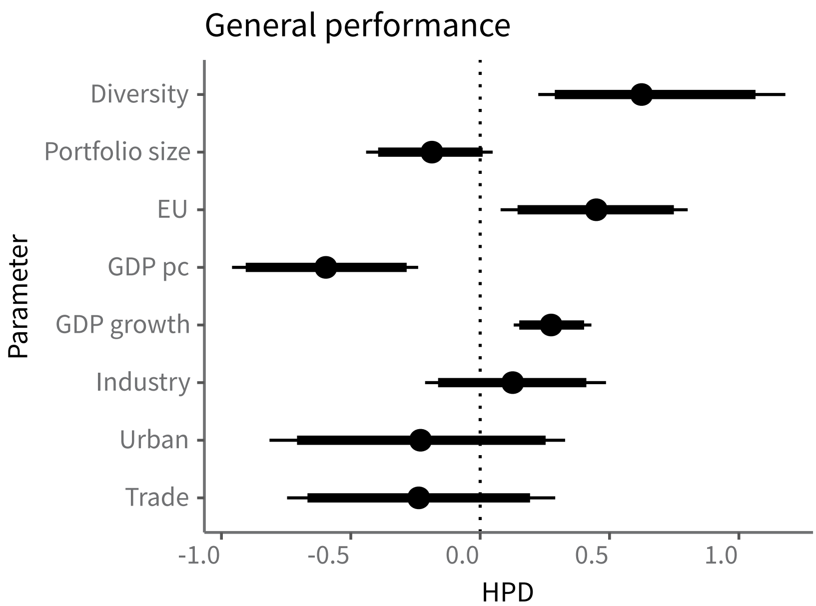

Figure 3.7: Slopes with the effects of control and main variables on performance. General performance

```
S.betas %>%
  filter(Indicator %in% c("Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                           levels = rev(c("Diversity",
                                          "Portfolio size",
                                          "EU",
                                          "GDP pc", "GDP growth",
                                          "Industry", "Urban",
                                          "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle("Specific 1980 performance")
```

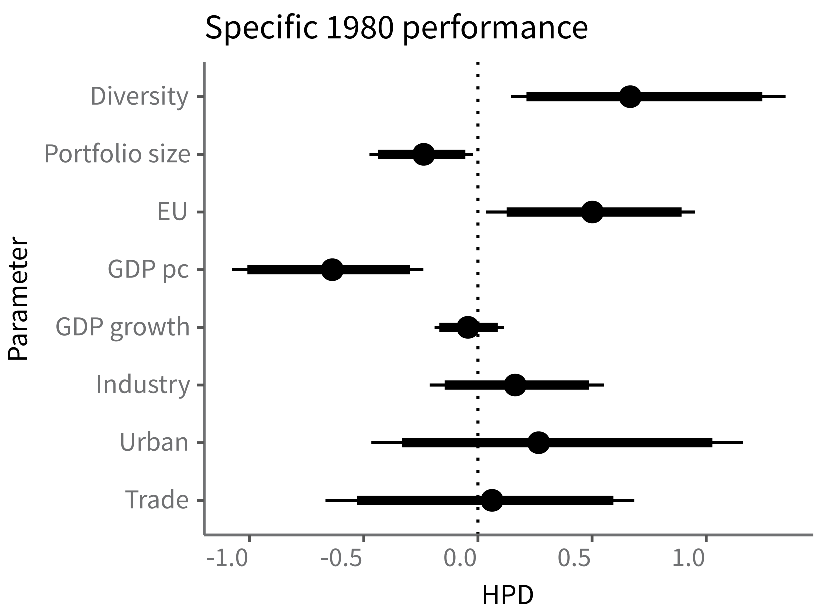

Figure 3.8: Slopes with the effects of control and main variables on performance. Specific 1980 performance

```
S.betas %>%
  filter(Indicator %in% c("General", "Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                           levels = rev(c("Diversity",
                                          "Portfolio size",
                                          "EU",
                                          "GDP pc", "GDP growth",
                                          "Industry", "Urban",
                                          "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  facet_wrap(~ Indicator, scales="free") +
  geom_vline(xintercept = 0, lty = 3) #+

ci(S.betas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = Indicator, color = Indicator)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.5)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.5)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.5)) +
  ylab("HPD") + xlab("Parameter") +
  geom_vline(xintercept = 0, lty = 3)

ggplot(S.betas, aes(x = value, color = Indicator, fill = Indicator)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ Variable, ncol = 5, scales = "free") +
  xlab("HPD") +
  geom_vline(xintercept = 0, lty = 3)
```

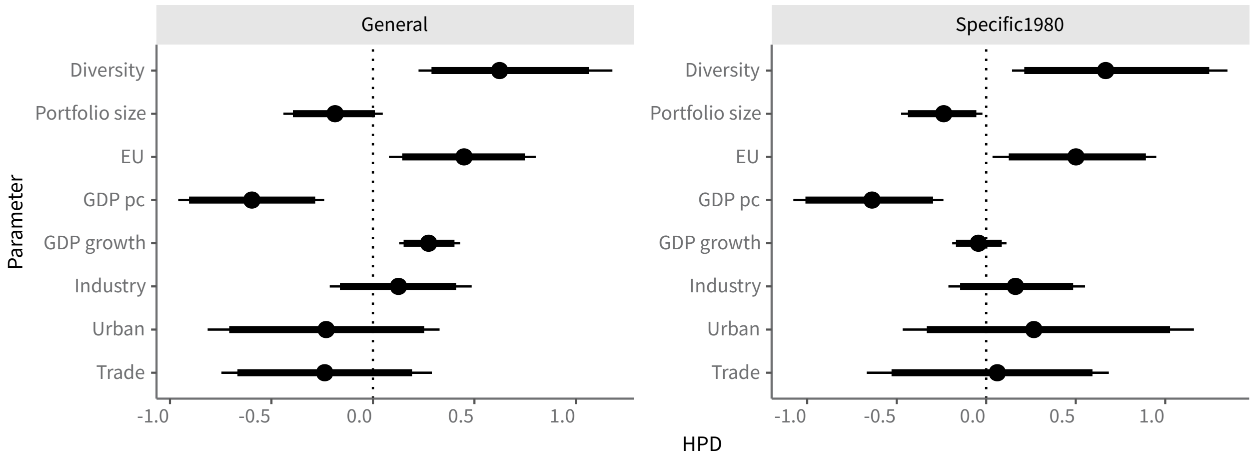

Figure 3.9: Slopes with the effects of control and main variables on performance, by performance indicator.

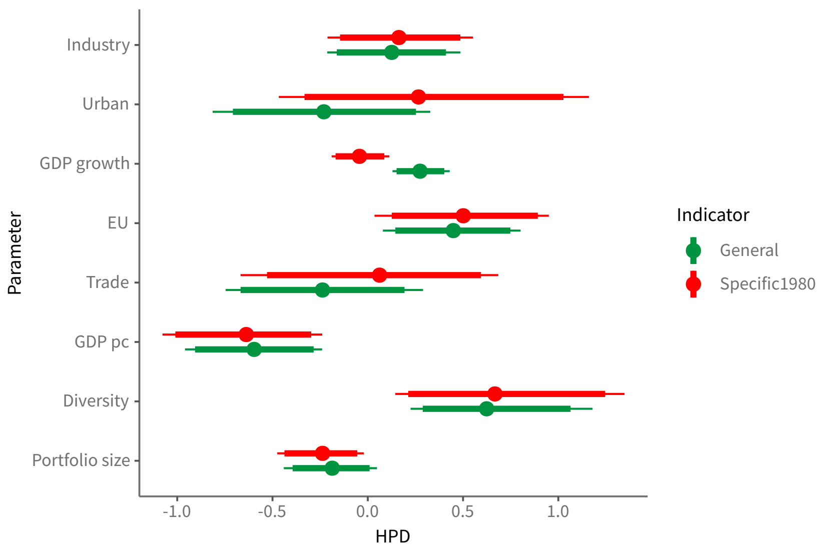

Figure 3.10: Slopes with the effects of control and main variables on performance.

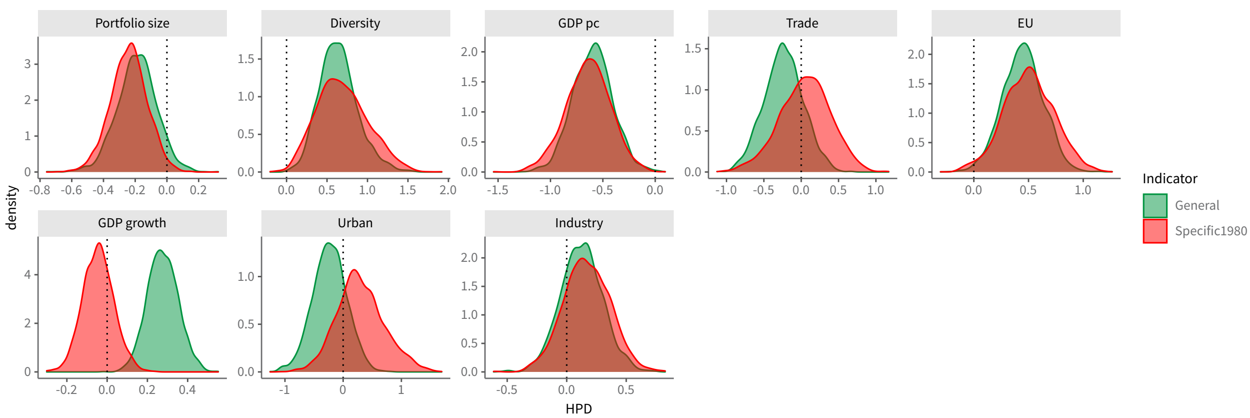

Figure 3.11: Slopes with the effects of control and main variables on performance.

Variables by evidence.

```
S.betas %>%
  filter(value != 0) %>%
  group_by(Variable, Indicator) %>%
  summarize(`Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
    `Mean expected effect` = mean(value)) %>%
  group_by(Variable, Indicator) %>%
  mutate(max = max(abs(`Prob > 0`), abs(`Prob < 0`))) %>%
  arrange(desc(max)) %>%
  select(-max) %>%
  kable()
```

Variable	Indicator	Prob > 0	Prob < 0	Mean expected effect
Diversity	General	1.00	0.00	0.64
GDP pc	Specific1980	0.00	1.00	-0.64
GDP growth	General	1.00	0.00	0.28
GDP pc	General	0.00	1.00	-0.60
Diversity	Specific1980	1.00	0.00	0.69
EU	General	0.99	0.01	0.45
Portfolio size	Specific1980	0.02	0.98	-0.24
EU	Specific1980	0.98	0.02	0.50
Portfolio size	General	0.06	0.94	-0.19
Trade	General	0.18	0.82	-0.24
Industry	Specific1980	0.81	0.19	0.17
Urban	General	0.21	0.79	-0.23
Urban	Specific1980	0.78	0.22	0.30
Industry	General	0.76	0.24	0.13
GDP growth	Specific1980	0.28	0.72	-0.04
Trade	Specific1980	0.58	0.42	0.05

```
S.betas.gral <- S.betas %>%
  filter(Indicator == "General") %>%
  mutate(Parameter = Variable)

ggs_caterpillar(S.betas.gral) +
  geom_vline(xintercept = 0, lty = 3)
```

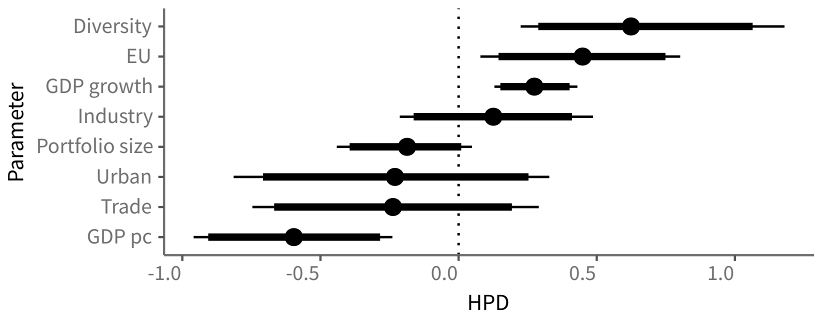

Figure 3.12: Slopes for the effects of control variables on performance (Global performance indicator).

3.5 *Model evaluation*

What is the model fit?

```
L.data <- plab("resid", list(Indicator = indicator.label,
                             Country = country.label,
                             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)
```

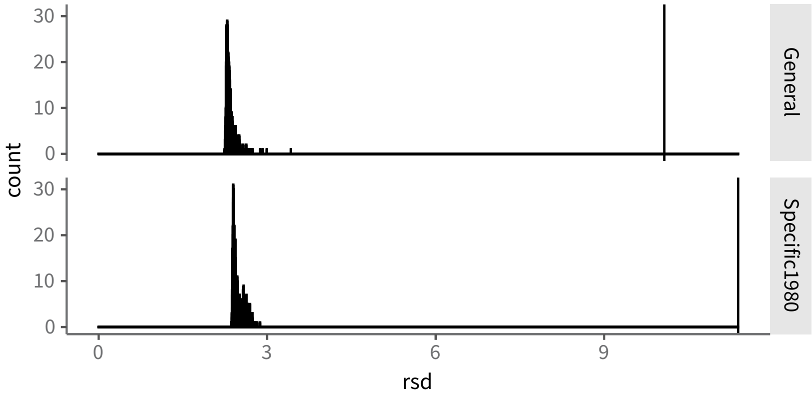

Figure 3.13: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

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

Indicator	Pseudo.R2
General	0.77
Specific1980	0.78

Which are the observations with higher residuals, further away from the expectation?[1]

[1] Negative residuals are cases where the country has lower innovation than expected according to the model.

```
L.data ← plab("resid", list(Indicator = indicator.label,
                            Country = country.label,
                            Year = year.label))

S.resid ← ggs(s, family = "resid", par_labels = L.data) %>%
  filter(!str_detect(Country, "Z-"))

S.resid %>%
  group_by(Country, Indicator) %>%
  summarize(Residual = mean(value)) %>%
  mutate(`Mean absolute residual` = abs(Residual)) %>%
  ungroup() %>%
  arrange(desc(`Mean absolute residual`)) %>%
  select(-`Mean absolute residual`) %>%
  slice(1:20) %>%
  kable()

ci.betas ← ci(S.betas) %>%
  mutate(Model = M.lab)

save(ci.betas, file = paste0("ci_betas-", M, ".RData"))
```

Country	Indicator	Residual
Canada	General	0.98
United States	General	0.87
Portugal	General	-0.83
United States	Specific1980	0.77
Greece	Specific1980	-0.66
Sweden	Specific1980	-0.54
Greece	General	-0.51
New Zealand	Specific1980	-0.51
Australia	General	0.44
Germany	Specific1980	-0.42
Canada	Specific1980	0.42
Finland	General	0.41
Japan	General	-0.40
Ireland	General	-0.39
Australia	Specific1980	0.39
Belgium	Specific1980	0.36
United Kingdom	Specific1980	-0.31
Portugal	Specific1980	-0.30
Netherlands	Specific1980	0.30
Japan	Specific1980	-0.29

4

Explanatory model of performance (robustness, with interaction)

```
library(PolicyPortfolios)
data(consensus)

ca <- consensus %>%
  filter(Sector == "Environmental") %>%
  droplevels() %>%
  pp_measures() %>%
  filter(Measure %in% c("Div.aid", "Size")) %>%
  select(-Measure.label) %>%
  spread(Measure, value) %>%
  rename(Diversity = Div.aid)

countries <- as.character(levels(ca$Country))
nC <- length(countries)
years <- range(ca$Year)
save(countries, years, file = "details.RData")
```

4.1 Performance

```
perf <- foreign::read.dta("jahn/PoEP_Replication_Data/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% 1980:2010)
```

4.2 Covariates

World Development Indicators - Revenue.

```
load("wdi/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

```
countries <- as.character(levels(ca$Country))
load("wdi/wdi.RData")
wdi <- wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi <- subset(wdi, year >= 1976 & year <= 2005)
wdi$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(pred

#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(


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(pred

# 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$`2005`/wdi.gcr.w$`1976`)


# 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("vdem/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("wgi/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("./polcon/polcon2017.RData") # loads polcon
```

For the "Green parties", data comes from Volkens (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("./manifesto/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<="2005-12-31")

# 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("1976-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}
```

Save and arrange for analysis.

```
d.perf <- perf %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("Korea, Republic of", "Turkey")) %>%
  # Delete Years for which we don't have data
  filter(Year >= 1980 & Year <= 2005) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Specific1980")) %>%
  droplevels()

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)

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("get-eu_time.R") # generates eu.ms

diversity <- ca %>%
  select(Sector, Country, Year, Diversity) %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  filter(Sector == "Environmental") %>%
  filter(Year >= 1980 & Year <= 2005) %>%
  select(-Sector) %>%
  droplevels() %>%
  reshape2::acast(Country ~ Year, value.var = "Diversity")
if ( length(which(!dimnames(diversity)[[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 <= 2005) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  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("wdi/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 <= 2005) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  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("wdi/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2005) %>%
  mutate(urban = std(urban)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
```

```
load("wdi/wdi-industry.RData") # industry
industry <- industry %>%
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1980 & year <= 2005) %>%
  mutate(industry = std(industry)) %>%
  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 <= 2005) %>%
  mutate(trade = std(trade)) %>%
  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 <= 2005) %>%
  mutate(value = std(value)) %>%
  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 <= 2005) %>%
  mutate(polcon = std(polcon)) %>%
  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 <= 2005) %>%
  mutate(value = std(value)) %>%
  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 <- ca %>%
#  filter(Measure == "Size") %>%
  filter(Sector == "Environmental") %>%
#  spread(Measure, value) %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1980 & Year <= 2005) %>%
  mutate(Size = std(logit(Size))) %>%
  ungroup() %>%
  reshape2::acast(Country ~ Year, variable.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")

green.socialist <- cpm %>%
  ungroup() %>%
  mutate(calendar.year = as.numeric(format(date, "%Y"))) %>%
  mutate(calendar.month = as.numeric(format(date, "%m"))) %>%
  mutate(year = ifelse(calendar.month > 6, calendar.year + 1, calendar.year)) %>%
  # Manually delete some elections that are held too close
  # and assigned both to the same year
  # Delete the first one, as in case of instability it is unlikely
  # that they have passed legislation
  filter(!(country == "Japan" & date == as.Date("1979-10-07"))) %>%
  filter(!(country == "Greece" & date == as.Date("1989-11-05"))) %>%
  filter(!(country == "Denmark" & date == as.Date("1987-09-08"))) %>%
  #
  select(country, year, family, p.seats) %>%
  full_join(expand.grid(country = country.label,
                        year = (min(year.label.numeric)-5):max(year.label.numeric),
                        family = c("Socialist", "Green"))) %>%
  group_by(country, family) %>%
  arrange(country, family, year) %>%
  # A bit clumsy, but works
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  mutate(p.seats = ifelse(is.na(p.seats), lag(p.seats), p.seats)) %>%
  ungroup() %>%
  select(country, year, family, p.seats) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(p.seats = ifelse(is.na(p.seats), 0, p.seats)) %>%
  group_by(family) %>%
  mutate(p.seats = std(p.seats)) %>%
  ungroup() %>%
  unique() %>%
  reshape2::acast(family ~ country ~ year, value.var = "p.seats")
if ( length(which(!dimnames(green.socialist)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here
if (!dimnames(green.socialist)[[1]][1] == "Green" & sector.label[1] == "Environmental") stop("Ep! There is

# Use only the green dimension
green <- green.socialist[1,,]


DJ <- list(
  Y = unname(Y),
  diversity = unname(diversity),
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  gdp.growth = unname(gdp.growth),
  urban = unname(urban),
  industry = unname(industry),
  industry.means = 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),
  green = green,
  nC = nC, #nC.real = nC.real,
  nI = nI,
  nY = nY)#,
```

```
nC   # Number of countries
```

```
## [1] 21
```

```
nS   # Number of sectors
```

```
## [1] 2
```

```
years  # Range of years
```

```
## [1] 1976 2005
```

4.3 Model

```
M ← "performance-tscs-robustness"
M.lab ← "Robustness, with interaction"
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,c,t])
        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] * diversity[c,t]
                 + theta[3,i] * portfolio.size[c,t] * diversity[c,t]
                 + rho[i,c] * (Y[i,c,t-1] - mu[i,c,t-1] )
        tau[i,c,t] ← 1 / sigma.sq[i,c,t]
        sigma.sq[i,c,t] ← exp(lambda_c[i,c])
        resid[i,c,t] ← Y[i,c,t] - mu[i,c,t]
      }
      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] * diversity[c,1]
                 + theta[3,i] * portfolio.size[c,1] * diversity[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]
    }

    #
    # Priors for variance component
    #
    lambda[1,i] ~ dnorm(0, 2^-2)
    lambda[2,i] ~ dnorm(0, 2^-2)

    #
    # Priors for the intercept
    #
    alpha[i] ~ dunif(0, 100)

    #
    # Priors for the control variables
    #
    for (b in 1:8) {
      beta[b,i] ~ dnorm(0, 1^-2)
    }

    #
```

```
    # Priors for main effects
    #
    for (t in 1:3) {
      theta[t,i] ~ dnorm(0, 1^-2)
    }
  }

  # 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], 0.05^-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")
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))

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 = 4, adapt = 1e2, burnin = 1e3, 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 = ""))
ggmcmc(ggs(s, family = "sigma|alpha|beta|lambda|nu|rho"),
       plot = c("traceplot", "crosscorrelation", "caterpillar"),
       param_page = 8, file = paste("ggmcmc-partial-", M, ".pdf", sep = ""))
```

4.4 Model results

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).

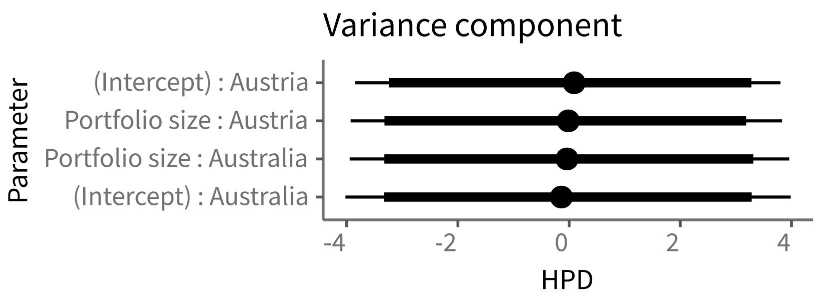

Figure 4.1: Variance component.

```
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)
ggs_caterpillar(S.lambda.c, label = "Country") +
  facet_grid(~ Indicator) +
  ggtitle("Variance component (countries)")
```

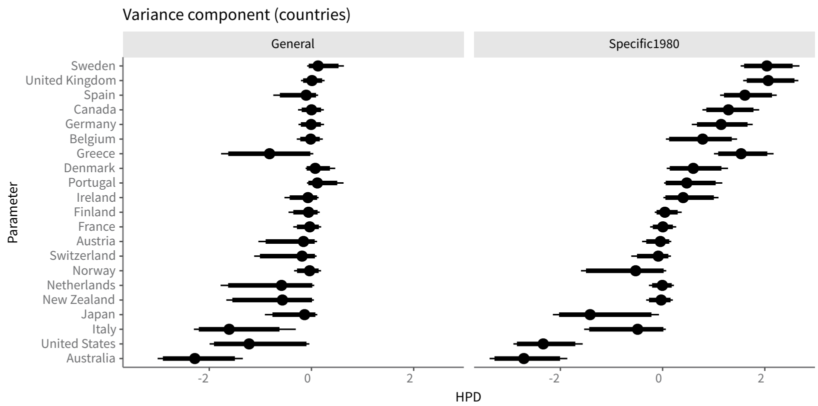

Figure 4.2: Variance component (country varying intercepts).

Auto-regressive components.

```
L.rho ← plab("rho", list(Indicator = indicator.label, Country = country.label))
S.rho ← ggs(s, family = "rho", par_labels = L.rho)
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)") +
  scale_colour_xfim()
```

Lagged dependent variable.

```
L.delta ← plab("delta", list(Indicator = indicator.label, Country = country.label))
S.delta ← ggs(s, family = "delta", par_labels = L.delta)
ggs_caterpillar(S.delta, label = "Country") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color=Indicator) +
  ggtitle("Lagged dependent variable (countries)") +
  scale_colour_xfim()
```

```
L.thetas ← plab("theta", list(Variable = c("Portfolio size",
                                          "Diversity", "Portfolio size * Diversity"),
                        Indicator = indicator.label))

L.betas ← plab("beta", list(Variable = c("GDP pc", "Trade", "EU", "Green",
                                        "GDP growth", "Urban", "Industry",
```

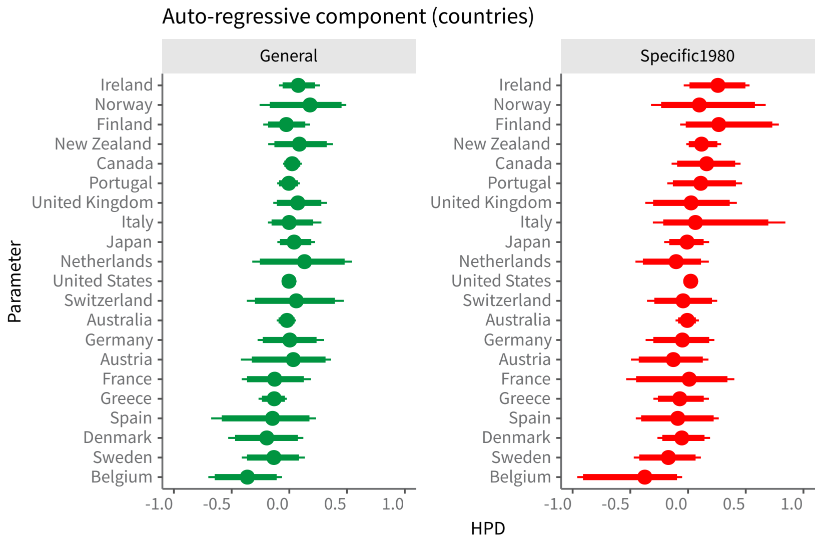

Figure 4.3: Auto-regressive component.

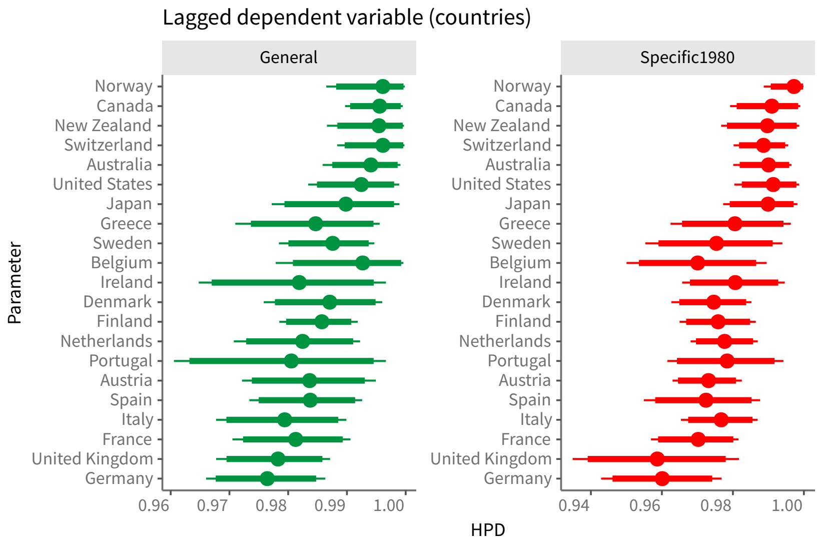

Figure 4.4: Lagged dependent variable.

```
                                         "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)
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	General	-0.40	0.51	0.21	0.79	-0.40
Portfolio size	Specific1980	-0.99	0.49	0.02	0.98	-0.99
Diversity	General	0.64	0.25	1.00	0.00	0.66
Diversity	Specific1980	0.77	0.33	1.00	0.00	0.80
Portfolio size * Diversity	General	0.30	0.64	0.66	0.34	0.29
Portfolio size * Diversity	Specific1980	0.98	0.62	0.94	0.06	0.99
GDP pc	General	-0.62	0.20	0.00	1.00	-0.62
GDP pc	Specific1980	-0.65	0.21	0.00	1.00	-0.66
Trade	General	-0.23	0.27	0.20	0.80	-0.23
Trade	Specific1980	0.02	0.35	0.52	0.48	0.01
EU	General	0.46	0.19	0.99	0.01	0.46
EU	Specific1980	0.49	0.24	0.98	0.02	0.49
GDP growth	General	0.28	0.07	1.00	0.00	0.28
GDP growth	Specific1980	-0.04	0.08	0.33	0.67	-0.03
Urban	General	-0.25	0.30	0.20	0.80	-0.25
Urban	Specific1980	0.24	0.41	0.74	0.26	0.26
Industry	General	0.13	0.18	0.77	0.23	0.13
Industry	Specific1980	0.13	0.19	0.76	0.24	0.13

```
ggs_caterpillar(S.betas, label = "Variable") +
  facet_wrap(~ Indicator, scales="free") +
  aes(color = Indicator) +
  geom_vline(xintercept = 0, lty = 3) +
  scale_colour_xfim()
```

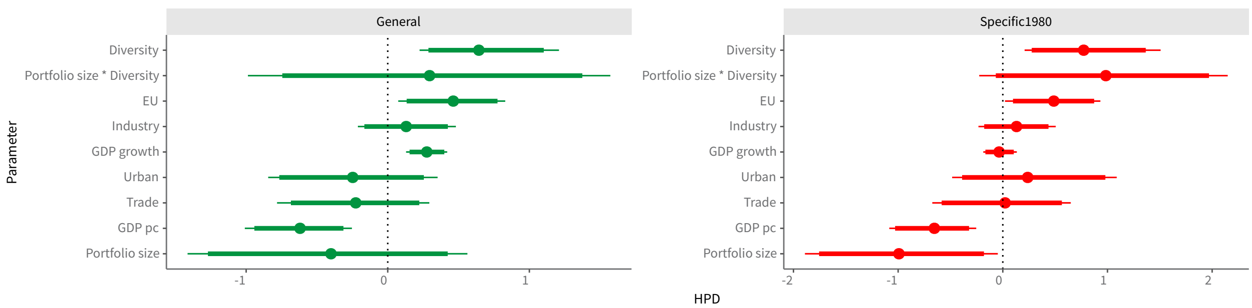

Figure 4.5: Slopes with the effects of control and main variables on performance, by performance indicator.

```
S.betas %>%
  filter(Indicator %in% c("General", "Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                           levels = rev(c("Diversity",
                                          "Portfolio size * Diversity",
                                          "Portfolio size",
                                          "EU",
                                          "GDP pc", "GDP growth",
```

```
                                                "Industry", "Urban",
                                                "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  facet_wrap(~ Indicator, scales="free") +
  geom_vline(xintercept = 0, lty = 3) #+
```

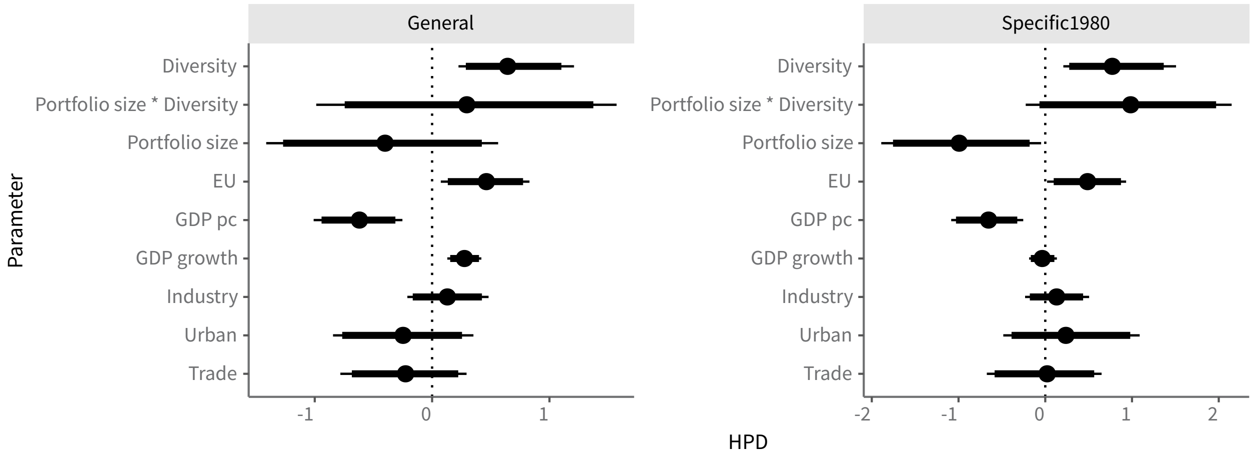

Figure 4.6: Slopes with the effects of control and main variables on performance, by performance indicator.

```
S.betas %>%
  filter(Indicator %in% c("General")) %>%
  mutate(Variable = factor(as.character(Variable),
                           levels = rev(c("Diversity",
                                          "Portfolio size * Diversity",
                                          "Portfolio size",
                                          "EU",
                                          "GDP pc", "GDP growth",
                                          "Industry", "Urban",
                                          "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle("General performance")
```

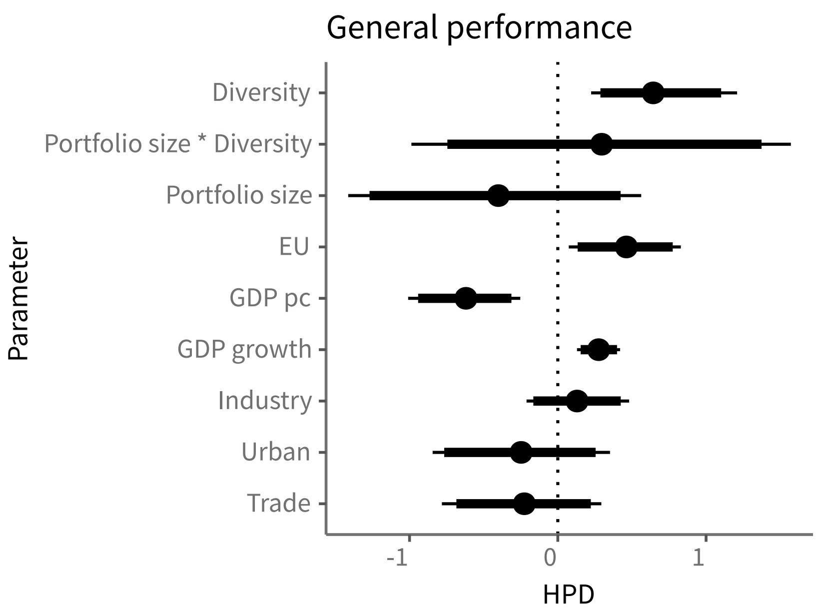

Figure 4.7: Slopes with the effects of control and main variables on performance. General performance

```
S.betas %>%
  filter(Indicator %in% c("Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                           levels = rev(c("Diversity",
                                          "Portfolio size * Diversity",
                                          "Portfolio size",
                                          "EU",
                                          "GDP pc", "GDP growth",
                                          "Industry", "Urban",
                                          "Trade")))) %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle("Specific 1980 performance")
```

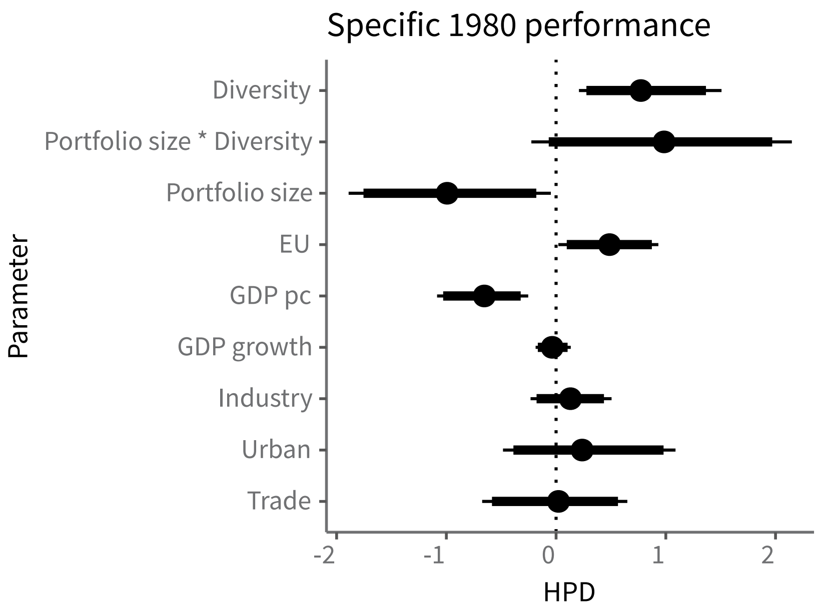

Figure 4.8: Slopes with the effects of control and main variables on performance. Specific 1980 performance

```
S.betas %>%
  filter(Indicator %in% c("General", "Specific1980")) %>%
  mutate(Variable = factor(as.character(Variable),
                           levels = rev(c("Diversity",
                                          "Portfolio size * Diversity",
                                          "Portfolio size",
                                          "EU",
                                          "GDP pc", "GDP growth",
                                          "Industry", "Urban",
                                          "Trade")))) %>%
  filter(Variable ≠ "Portfolio size * Diversity") %>%
ggs_caterpillar(label = "Variable", sort = FALSE) +
  facet_wrap(~ Indicator, scales="free") +
  geom_vline(xintercept = 0, lty = 3) #+


ci(S.betas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = Indicator, color = Indicator)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.5)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.5)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.5)) +
  ylab("HPD") + xlab("Parameter") +
  geom_vline(xintercept = 0, lty = 3)
```

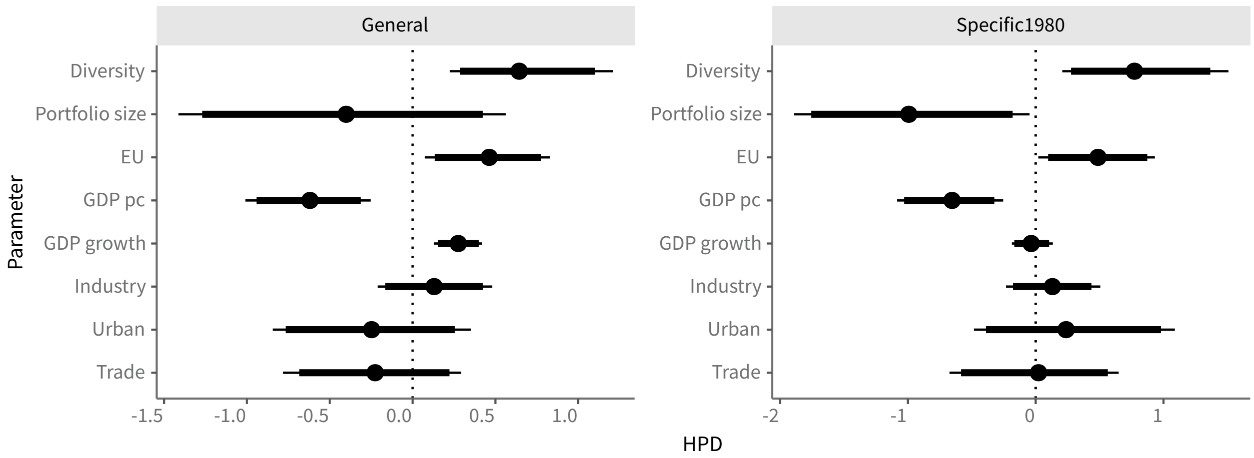

Figure 4.9: Slopes with the effects of control and main variables on performance, by performance indicator.

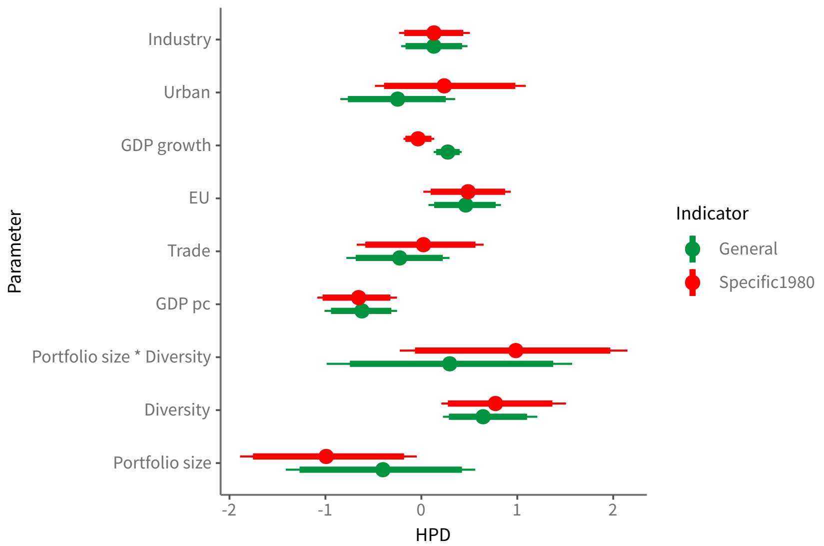

Figure 4.10: Slopes with the effects of control and main variables on performance.

```
ggplot(S.betas, aes(x = value, color = Indicator, fill = Indicator)) +
  geom_density(alpha = 0.5) +
  facet_wrap(~ Variable, ncol = 5, scales = "free") +
  xlab("HPD") +
  geom_vline(xintercept = 0, lty = 3)
```

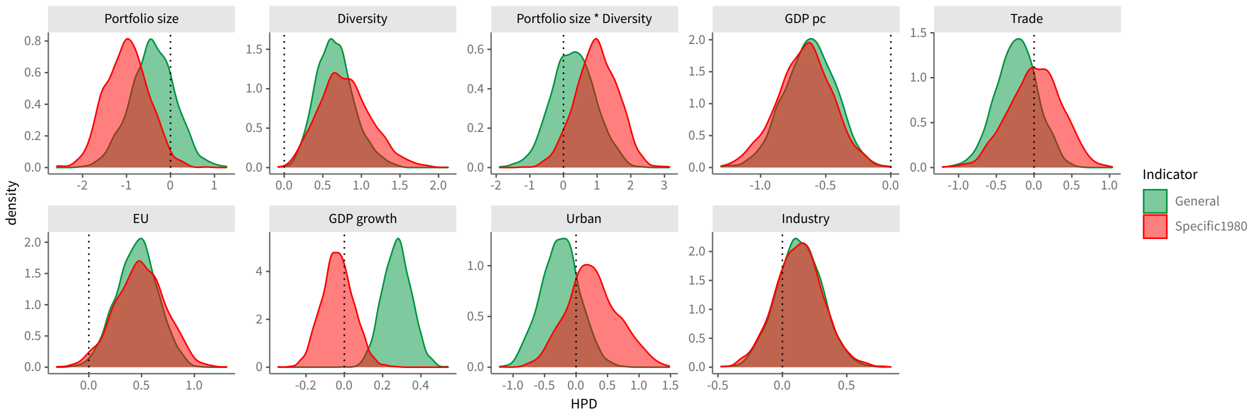

Figure 4.11: Slopes with the effects of control and main variables on performance.

Variables by evidence.

```
S.betas %>%
  filter(value ≠ 0) %>%
  group_by(Variable, Indicator) %>%
  summarize(`Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
    `Mean expected effect` = mean(value)) %>%
  group_by(Variable, Indicator) %>%
  mutate(max = max(abs(`Prob > 0`), abs(`Prob < 0`))) %>%
  arrange(desc(max)) %>%
  select(-max) %>%
  kable()
```

Variable	Indicator	Prob > 0	Prob < 0	Mean expected effect
GDP pc	General	0.00	1.00	-0.62
GDP pc	Specific1980	0.00	1.00	-0.66
Diversity	General	1.00	0.00	0.66
GDP growth	General	1.00	0.00	0.28
Diversity	Specific1980	1.00	0.00	0.80
EU	General	0.99	0.01	0.46
Portfolio size	Specific1980	0.02	0.98	-0.99
EU	Specific1980	0.98	0.02	0.49
Portfolio size * Diversity	Specific1980	0.94	0.06	0.99
Trade	General	0.20	0.80	-0.23
Urban	General	0.20	0.80	-0.25
Portfolio size	General	0.21	0.79	-0.40
Industry	General	0.77	0.23	0.13
Industry	Specific1980	0.76	0.24	0.13
Urban	Specific1980	0.74	0.26	0.26
GDP growth	Specific1980	0.33	0.67	-0.03
Portfolio size * Diversity	General	0.66	0.34	0.29
Trade	Specific1980	0.52	0.48	0.01

```
S.betas.gral ← S.betas %>%
  filter(Indicator == "General") %>%
  mutate(Parameter = Variable)

ggs_caterpillar(S.betas.gral) +
  geom_vline(xintercept = 0, lty = 3)
```

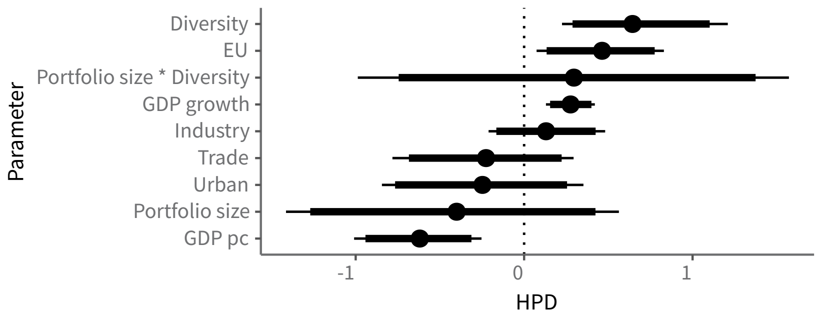

Figure 4.12: Slopes for the effects of control variables on performance (Global performance indicator).

4.5 Model evaluation

What is the model fit?

```
L.data <- plab("resid", list(Indicator = indicator.label,
                             Country = country.label,
                             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)
```

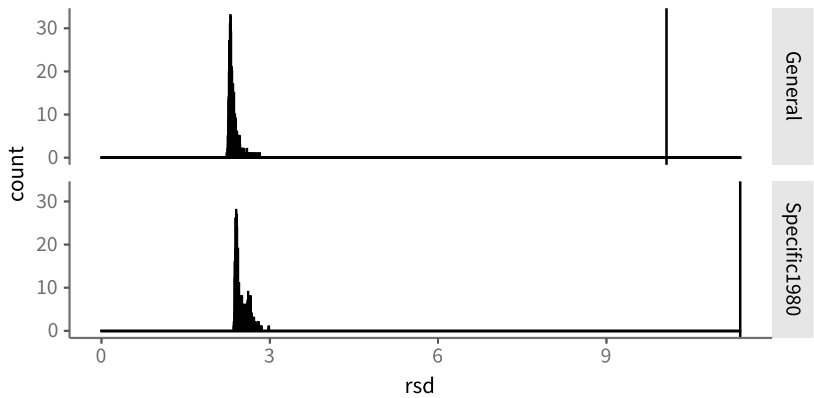

Figure 4.13: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

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

Which are the observations with higher residuals, further away from the expectation?[1]

[1] Negative residuals are cases where the country has lower innovation than expected according to the model.

Indicator	Pseudo.R2
General	0.77
Specific1980	0.78

```
L.data ← plab("resid", list(Indicator = indicator.label,
                           Country = country.label,
                           Year = year.label))

S.resid ← ggs(s, family = "resid", par_labels = L.data) %>%
  filter(!str_detect(Country, "Z-"))

S.resid %>%
  group_by(Country, Indicator) %>%
  summarize(Residual = mean(value)) %>%
  mutate(`Mean absolute residual` = abs(Residual)) %>%
  ungroup() %>%
  arrange(desc(`Mean absolute residual`)) %>%
  select(-`Mean absolute residual`) %>%
  slice(1:20) %>%
  kable()
```

Country	Indicator	Residual
Canada	General	0.99
United States	General	0.88
Portugal	General	-0.83
United States	Specific1980	0.77
Greece	Specific1980	-0.66
Sweden	Specific1980	-0.54
New Zealand	Specific1980	-0.50
Greece	General	-0.50
Australia	General	0.45
Germany	Specific1980	-0.44
Finland	General	0.41
Australia	Specific1980	0.40
Ireland	General	-0.39
Canada	Specific1980	0.39
Japan	General	-0.38
Belgium	Specific1980	0.35
United Kingdom	Specific1980	-0.31
Portugal	Specific1980	-0.31
Netherlands	Specific1980	0.29
United Kingdom	General	0.29

```
ci.betas ← ci(S.betas) %>%
  mutate(Model = M.lab)

save(ci.betas, file = paste0("ci_betas-", M, ".RData"))
```

5

Explanatory model of instrument diversity

Data is TSCS, by sector.

```
library(PolicyPortfolios)
data(consensus)

D <- bind_rows(
  # Calculate portfolio measures sector by sector
  consensus %>%
    filter(Sector == "Environmental") %>%
    droplevels() %>%
    pp_measures(),
  consensus %>%
    filter(Sector == "Social") %>%
    droplevels() %>%
    pp_measures())


# Add more fake countries

# Z-01 Increases portfolio size from minimum to maximum observed size
#      over time
range.size.env <- range(filter(D, Sector == "Environmental" & Measure == "Size")$value)
range.size.soc <- range(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Environmental", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.env[1], range.size.env[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Social", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.soc[1], range.size.soc[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

# Z-02, Z-11 Fixes government effectiveness to its mean
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))


# Z-12:Z-21 Fixes portfolio size for veto players to move
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)
```

```r
D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))
```

```r
diversity <- D %>%
  mutate(Country = as.factor(Country)) %>%
  select(-Measure) %>%
  rename(Measure = Measure.label) %>%
  spread(Measure, value) %>%
  mutate(`Diversity/W (Average Instrument Diversity)` =
         `Diversity (Average Instrument Diversity)` / (1 - `Proportion of targets covered`)) %>%
  select(Sector, Country, Year,
    `Diversity (Average Instrument Diversity)`,
    `Diversity (Gini-Simpson)`,
    `Equitability (Shannon)`,
    `Equality of Instrument configurations`)

countries <- as.character(unique(D$Country))
nC <- length(countries)
years <- range(D$Year)
```

5.1 Performance

```r
perf <- foreign::read.dta("jahn/PoEP_Replication_Data/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% 1980:2010)

d.perf <- perf %>%
  # Delete Years for which we don't have data
  filter(Year >= 1980 & Year <= 2005) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Water", "Specific1980")) %>%
  droplevels()

Y.performance <- reshape2::acast(d.perf, Indicator ~ Country ~ Year, value.var = "Performance")
nYperformance <- dim(Y.performance)[3]
```

5.2 Covariates

World Development Indicators - Revenue.

```r
load("wdi/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("wdi/wdi.RData")
wdi <- wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi <- subset(wdi, year >= 1976 & year <= 2005)
wdi$country[wdi$country=="Korea, Rep."] <- "South Korea"
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(pred

#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(


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(pred

# 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$`2005`/wdi.gcr.w$`1976`)


# 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()
```

```r
# 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)
```

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.

```r
load("wgi/wgi-full-v201029.RData")
gov.eff.original <- wgi %>%
  filter(indicator == "Government effectiveness") %>%
  select(country, year, value = Estimate) %>%
  expand_grid(Sector = c("Environmental", "Social"))
```

Add fake government effectiveness

```r
range.ge.env <- range.ge.soc <- range(gov.eff.original$value, na.rm = TRUE)

gov.eff.original <- gov.eff.original %>%
  bind_rows(expand_grid(tibble(Sector = "Environmental",
                               country = paste0("Z-", sprintf("%02d", 2:11)),
                               value = seq(range.ge.env[1],
                                           range.ge.env[2],
                                           length.out = length(2:11))),
                        year = min(D$Year):max(D$Year)))

gov.eff.original <- gov.eff.original %>%
  bind_rows(expand_grid(tibble(Sector = "Social",
                               country = paste0("Z-", sprintf("%02d", 2:11)),
                               value = seq(range.ge.soc[1],
                                           range.ge.soc[2],
                                           length.out = length(2:11))),
                        year = min(D$Year):max(D$Year)))
```

Political Constraints - polconIII. An indicator of "veto players" comes from Henisz (2002). The indicator "estimates the feasibility of policy change. [That is], the extent to which a change in the preferences of any one actor may lead to a change in government policy". Higher values represent systems with higher constraints.

```r
load("polcon/polcon2017.RData") # loads polcon
```

Add fake political constraints values.

```r
polcon.d <- polcon %>%
  as_tibble() %>%
  filter(year %in% years[1]:years[2]) %>%
  mutate(country.polity = as.character(country.polity)) %>%
  mutate(country.polity = ifelse(country.polity == "Germany West", "Germany", country.polity)) %>%
  filter(country.polity %in% countries) %>%
  select(Country = country.polity, Year = year, polcon)

range.polcon <- polcon.d %>%
  select(polcon) %>%
  unlist(., use.names = FALSE) %>%
  range()

polcon.d <- polcon.d %>%
  bind_rows(expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                               polcon = seq(range.polcon[1],
                                            range.polcon[2],
                                            length.out = length(12:21))),
                        Year = min(D$Year):max(D$Year)))
```

For the "Green parties", data comes from Volkens (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("manifesto/cpm-consensus.RData")
cpm$country <- as.character(cpm$country)
cpm$country[cpm$country=="Korea"] <- "South Korea"
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<="2005-12-31")

# 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, as.numeric(diff(c(as.Date("1976-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}
```

For the salience of each topic, we employ data from Volkens (2013), weighting the proportion of votes to each party and the importance that each party gives to environmental issues or the expansion of social welfare.

```
load("manifesto/201029-cpm-salience.RData")

salience <- cpm.salience %>%
  filter(Country %in% countries) %>%
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  filter(Year %in% years[1]:years[2])
```

Border contiguity

```
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% countries,
                       dimnames(M.borders)[[2]] %in% countries]
```

Trade dependency

```
load("trade/trade.RData")

rm(M.trade, M.trade.imports)
```

Save and arrange for analysis.

```
diversity <- diversity %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("South Korea", "Turkey")) %>%
```

```
  droplevels()
diversity.l <- diversity %>%
  gather(Measure, value, -c(Sector, Country, Year))
Y <- reshape2::acast(diversity.l,
                     Sector ~ Country ~ Year ~ Measure, value.var = "value")
nS <- dim(Y)[1]
nC <- dim(Y)[2]
nY <- dim(Y)[3]
nD <- dim(Y)[4]

country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.fake <- 21
nC.real <- nC - nC.fake
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)

year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(as.numeric(year.label))
nY <- length(year.label)

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)

diversity.label <- dimnames(Y)[[4]]
nD <- length(diversity.label)

nB <- 11
b0 <- rep(0, nB)
B0 <- diag(nB)
diag(B0) <- 1^-2

# 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("get-eu_time.R") # generates eu.ms

# Fake countries
f.c <- paste0("Z-", sprintf("%02d", 1:21))
# Fake years
f.y <- year.label.numeric

GDPpc <- wdi %>%
  select(country, year, gdp.capita) %>%
  filter(country %in% country.label) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, 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")

trade.df <- wdi %>%
  select(country, year, trade) %>%
  filter(country %in% country.label) %>%
  mutate(trade = std(trade)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, trade = 0))
trade <- trade.df %>%
  reshape2::acast(country ~ year, value.var = "trade")
if ( length(which(!dimnames(trade)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

constraints.d <- expand_grid(Country = country.label,
                             Year = year.label.numeric) %>%
  left_join(polcon.d) %>%
  mutate(original.polcon = polcon) %>%
  mutate(polcon = std(polcon)) %>%
  mutate(polcon = ifelse(str_detect(Country, "^Z-") & is.na(polcon), 0, polcon))
constraints <- constraints.d %>%
```

```
  reshape2::acast(Country ~ Year, value.var = "polcon")
if ( length(which(!dimnames(constraints)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

gov.eff.df <- expand.grid(
    country = country.label,
    year = year.label.numeric,
    Sector = sector.label) %>%
  left_join(gov.eff.original) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1976 & year <= 2005) %>%
  mutate(value = std(value)) %>%
  select(Sector, country, year, value) %>%
  mutate(value = ifelse(str_detect(country, "^Z-") & is.na(value), 0, value))
gov.eff <- gov.eff.df %>%
  reshape2::acast(Sector ~ country ~ year, value.var = "value")
if ( length(which(!dimnames(gov.eff)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")

min.discard.zero <- function(x) return(min(x[x!=0]))
portfolio.size <-
  D %>%
  filter(Measure == "Size") %>%
  spread(Measure, value) %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector) %>%
  mutate(Size = ifelse(Size == 0, min.discard.zero(Size)/2, Size)) %>%
  mutate(Size = std(logit(Size))) %>%
  mutate(Size = ifelse(str_detect(Country, "Z2"), 0, Size)) %>%
  ungroup() %>%
  select(Country, Sector, Year, Size) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Size")
if ( length(which(!dimnames(portfolio.size)[[2]] == 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")

# Green socialist as salience
green.socialist <- salience %>%
  group_by(Sector) %>%
  mutate(Salience = std(Salience)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label, Country = f.c, Year = f.y, Salience = 0)) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Salience")
if ( length(which(!dimnames(green.socialist)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here

# Borders in tidy data
interdependency.contiguity <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wDiversity = value * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
```

```
  summarize(contiguity.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year ⩾ 1976 & Year ⩽ 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(contiguity.dependency = std(contiguity.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        contiguity.dependency = 0))

interdependency.contiguity ← interdependency.contiguity %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "contiguity.dependency")
if ( length(which(!dimnames(interdependency.contiguity)[[3]] == country.label)) > 0) stop("Ep! There is a m

# Trade in tidy data
interdependency.trade ←
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wDiversity = value * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
  summarize(trade.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year ⩾ 1976 & Year ⩽ 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(trade.dependency = std(trade.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        trade.dependency = 0))

interdependency.trade ← interdependency.trade %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "trade.dependency")
if ( length(which(!dimnames(interdependency.trade)[[3]] == country.label)) > 0) stop("Ep! There is a mistak


# Performance part
# Match it with the general Y, but only for the environmental sector
d.perf.fake ← expand_grid(
  Country = country.label[str_detect(country.label, "^Z-") ],
  Year = year.label.numeric,
  Indicator = unique(d.perf$Indicator),
  Performance = NA)

Y.performance ← diversity.l %>%
  select(Sector, Country, Year, Measure) %>%
  left_join(d.perf) %>%
  left_join(d.perf.fake) %>%
  filter(Sector == "Environmental") %>%
  group_by(Indicator) %>%
  mutate(Performance = std1(Performance)) %>%
  ungroup() %>%
  reshape2::acast(Indicator ~ Country ~ Year ~ Measure, value.var = "Performance")
# manually get rid of ghost performance for missing values
Y.performance ← Y.performance[-4,,,]

nP ← dim(Y.performance)[1]
performance.label ← dimnames(Y.performance)[[1]]
if ( length(which(!dimnames(Y.performance)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")
```

```
load("wdi/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 %in% year.label.numeric) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "gdp.growth")
gdp.growth <- gdp.growth[,-dim(gdp.growth)[2]]
if ( length(which(!dimnames(gdp.growth)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
gdp.growth.means <- apply(gdp.growth, 1, mean, na.rm = TRUE)
gdp.growth.means[is.nan(gdp.growth.means)] <- 0

load("wdi/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(urban = std(urban)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
urban <- urban[,-dim(urban)[2]]
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
urban.means <- apply(urban, 1, mean, na.rm = TRUE)
urban.means[is.nan(urban.means)] <- 0

load("wdi/wdi-industry.RData") # industry
industry <- industry %>%
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(industry = std(industry)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "industry")
industry <- industry[,-dim(industry)[2]]
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)
industry.means[is.nan(industry.means)] <- 0


DP <- list(
  Y = unname(Y),
  Y.performance = unname(Y.performance), nP = nP,
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  constraints = unname(constraints),
  gov.eff = unname(gov.eff),
  gov.eff.mean.observed = unname(apply(gov.eff, c(1, 2), mean, na.rm = TRUE)),
  interdependency.contiguity = unname(interdependency.contiguity),
  interdependency.trade = unname(interdependency.trade),
  portfolio.size = unname(portfolio.size),
  eu = unname(eu),
  green.socialist = unname(green.socialist),
  gdp.growth = unname(gdp.growth), gdp.growth.means = unname(gdp.growth.means),
  urban = unname(urban), urban.means = unname(urban.means),
  industry = unname(industry), industry.means = unname(industry.means),
  nC = nC,
  id.fake.countries = id.fake.countries,
  id.real.countries = id.real.countries,
  id.decade = id.decade, nDecades = nDecades,
  nB = nB, b0 = b0, B0 = B0,
  nS = nS,
  nD = nD,
  nY = nY)

nC   # Number of countries (including fake ones)
```

```
## [1] 42
```

```
nS   # Number of sectors
```

```
## [1] 2
```

```
years  # Range of years
```

```
## [1] 1976 2005
```

5.3 Model

```
M <- "diversity-main"
M.lab <- "Diversity (AID), Main"
m <- "model {
  #
  # Data part at the observational level
  #
  for (d in 1:nD) {
    for (s in 1:nS) {
      for (c in 1:nC) {
        for (t in 2:nY) {
          Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
          mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
                     + beta[1,s,d] * GDPpc[c,t-1]
                     + beta[4,s,d] * gov.eff[s,c,t-1]
                     + beta[5,s,d] * portfolio.size[s,c,t-1]
                     + beta[6,s,d] * trade[c,t-1]
                     + beta[7,s,d] * eu[c,t-1]
                     + beta[8,s,d] * green.socialist[s,c,t-1] # col 1 green, col 2 socialist
                     + beta[9,s,d] * constraints[c,t-1]
                     + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
                     + beta[11,s,d] * interdependency.trade[d,s,c,t]
                     + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
          tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
          sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
                                  + lambda[2,s,d] * portfolio.size[s,c,t]
                                  + gamma[c,d])
          resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
        }
        Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
        mu[s,c,1,d] <- alpha[s,d,1]
                     + beta[1,s,d] * GDPpc[c,1]
                     + beta[4,s,d] * gov.eff[s,c,1]
                     + beta[5,s,d] * portfolio.size[s,c,1]
                     + beta[6,s,d] * trade[c,1]
                     + beta[7,s,d] * eu[c,1]
                     + beta[8,s,d] * green.socialist[s,c,1] # col 1 green, col 2 socialist
                     + beta[9,s,d] * constraints[c,1]
                       + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
                       + beta[11,s,d] * interdependency.trade[d,s,c,1]

        resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
        tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
        sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
                                + lambda[2,s,d] * portfolio.size[s,c,1]
                                + gamma[c,d])
      }

      #
      # Degrees of freedom of GR
      #
      nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
      nu.trans[s,d] ~ dunif(0, 1)

      #
      # Priors for variance component
      #
      lambda[1,s,d] ~ dnorm(0, 2^-2)
      lambda[2,s,d] ~ dnorm(0, 2^-2)

      #
      # Priors for the intercept
```

```
  #
  for (decade in 1:nDecades) {
    alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
  }
  Alpha[s,d] ~ dunif(0, 1)
  tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
  sigma.alpha[s,d] ~  dunif(0, 0.5)

  #
  # Priors for the control variables
  #
  beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
  Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
  Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
}
#
# Data part for  performance
#
for (p in 1:nP) {
  for (c in 1:nC) {
    for (t in 2:nY) {
      Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
      mu.performance[p,c,t,d] <- eta.performance[1,p,d]
                                 + eta.performance[2,p,d] * resid[1,c,t,d]
                                 + eta.performance[3,p,d] * Y[1,c,t-1,d]
                                 + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
                                 + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
                                 + eta.performance[6,p,d] * GDPpc[c,t-1]
                                 + eta.performance[7,p,d] * trade[c,t-1]
                                 + eta.performance[8,p,d] * eu[c,t-1]
                                 + eta.performance[9,p,d] * gdp.growth[c,t-1]
                                 + eta.performance[10,p,d] * urban[c,t-1]
                                 + eta.performance[11,p,d] * industry[c,t-1]
    }
    Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
    mu.performance[p,c,1,d] <- eta.performance[1,p,d]
                               + eta.performance[2,p,d] * resid[1,c,1,d]
                               + eta.performance[3,p,d] * Y[1,c,1,d]
                               + eta.performance[4,p,d] * portfolio.size[1,c,1]
                               + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
                               + eta.performance[6,p,d] * GDPpc[c,1]
                               + eta.performance[7,p,d] * trade[c,1]
                               + eta.performance[8,p,d] * eu[c,1]
                               + eta.performance[9,p,d] * gdp.growth[c,1]
                               + eta.performance[10,p,d] * urban[c,1]
                               + eta.performance[11,p,d] * industry[c,1]
  }
  tau.performance[p,d] ~ dgamma(0.001, 0.001)
  sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
  for (e in 1:11) {
    eta.performance[e,p,d] ~ dnorm(0, 1^-2)
  }
}

# Variance component, varying intercepts by country
for (c in 1:nC) {
  gamma[c,d] ~ dnorm(0, 0.2^-2)
}
#
# AR(1) parameters
#
for (c in id.real.countries) {
  for (s in 1:nS) {
    rho[s,c,d] ~ dunif(-1, 1)
  }
  for (p in 1:nP) {
    rho.performance[p,c,d] ~ dunif(-1, 1)
  }
}
for (c in id.fake.countries) {
  for (s in 1:nS) {
    rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
  }
  for (p in 1:nP) {
    rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
  }
```

```
    }
  }

  # SEM Part for portfolio size
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (t in 2:nY) {
        portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
        mu.ps[s,c,t] <- alpha.ps[s,c]
                      + delta[1,s] * GDPpc[c,t-1]
                      + delta[2,s] * gov.eff[s,c,t-1]
                      + delta[3,s] * green.socialist[s,c,t-1]
                      + delta[4,s] * constraints[c,t-1]
      }
      tau.ps[s,c] ~ dgamma(0.1, 0.1)
      sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
      alpha.ps[s,c] ~ dnorm(0, 1^-2)
    }
    for (d in 1:4) {
      delta[d,s] ~ dnorm(0, 1^-2)
    }
  }
  # Mediated effects
  for (d in 1:nD) {
    for (s in 1:nS) {
      pi[1,s,d] <- delta[1,s] * beta[1,s,d]       # GDPpc
      pi[2,s,d] <- delta[2,s] * beta[4,s,d]       # gov.eff
      pi[3,s,d] <- delta[3,s] * beta[8,s,d]       # green
      pi[4,s,d] <- delta[4,s] * beta[9,s,d]       # constraints
    }
  }


  #
  # Missing data
  #
  for (c in 1:nC) {
    for (t in 1:nY) {
      gov.eff[1,c,t] ~ dnorm(mean(gov.eff.mean.observed[1,c]), 0.05^-2)
      gov.eff[2,c,t] ~ dnorm(gov.eff[1,c,t], 100)
      gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
      urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
      industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
    }
  }
  for (s in 1:nS) {
    for (c in 1:nC) {
      # Reverse years back for NA in early years
      for (t in 1:(nY-1)) {
        green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
      }
      for (d in 1:nD) {
        for (t in 1:(nY-1)) {
          interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
        }
      }
    }
  }
}"
write(m, file= paste("models/model-", M, ".bug", sep = ""))
par <- NULL
par <- c(par, "alpha", "beta", "theta", "sigma")
par <- c(par, "Alpha", "sigma.alpha")
par <- c(par, "Sigma")
par <- c(par, "lambda", "gamma")
par <- c(par, "nu")
par <- c(par, "rho")
par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 2:nY,
                        Diversity = 1:nD) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
```

```
  unlist(., use.names = FALSE)
par ← c(par, par.fake)
par.resid ← expand_grid(Sector = 1:nS,
                        Country = id.real.countries,
                        Year = 2:nY,
                        Diversity = 1:nD) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par ← c(par, par.resid)
par ← c(par, "delta", "alpha.ps", "sigma.ps", "pi")
par ← c(par, "eta.performance", "sigma.performance")
inits ← list(
  list(.RNG.name="base::Super-Duper", .RNG.seed=10),
  list(.RNG.name="base::Super-Duper", .RNG.seed=20),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=30),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=20))

t0 ← proc.time()
rj ← run.jags(model = paste("models/model-", M, ".bug", sep = ""),
  data = dump.format(DP, checkvalid=FALSE),
  inits = inits,
  modules = "glm",
  n.chains = 1, adapt = 2e2, burnin = 1e3, sample = 2e3, thin = 1,
  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 = ""))

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

```
## model {
##   #
##     #
##   for (d in 1:nD) {
##     for (s in 1:nS) {
##       for (c in 1:nC) {
##         for (t in 2:nY) {
##           Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
##           mu[s,c,t,d] ← alpha[s,d,id.decade[t]]
##                     + beta[1,s,d] * GDPpc[c,t-1]
##                     + beta[4,s,d] * gov.eff[s,c,t-1]
##                     + beta[5,s,d] * portfolio.size[s,c,t-1]
##                     + beta[6,s,d] * trade[c,t-1]
##                     + beta[7,s,d] * eu[c,t-1]
##                     + beta[8,s,d] * green.socialist[s,c,t-1]                  + beta[9,s,d] * constrain
##                     + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
##                     + beta[11,s,d] * interdependency.trade[d,s,c,t]
##                     + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
##           tau[s,c,t,d] ← 1 / sigma.sq[s,c,t,d]
##           sigma.sq[s,c,t,d] ← exp(lambda[1,s,d]
##                                 + lambda[2,s,d] * portfolio.size[s,c,t]
##                                 + gamma[c,d])
##           resid[s,c,t,d] ← Y[s,c,t,d] - mu[s,c,t,d]
##         }
##         Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
```

```
##         mu[s,c,1,d] <- alpha[s,d,1]
##                  + beta[1,s,d] * GDPpc[c,1]
##                  + beta[4,s,d] * gov.eff[s,c,1]
##                  + beta[5,s,d] * portfolio.size[s,c,1]
##                  + beta[6,s,d] * trade[c,1]
##                  + beta[7,s,d] * eu[c,1]
##                  + beta[8,s,d] * green.socialist[s,c,1]                  + beta[9,s,d] * constraints[c,1
##                    + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
##                    + beta[11,s,d] * interdependency.trade[d,s,c,1]
##
##         resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
##         tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
##         sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
##                               + lambda[2,s,d] * portfolio.size[s,c,1]
##                               + gamma[c,d])
##       }
##
##
##       #
##             #
##       nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
##       nu.trans[s,d] ~ dunif(0, 1)
##
##       #
##             #
##       lambda[1,s,d] ~ dnorm(0, 2^-2)
##       lambda[2,s,d] ~ dnorm(0, 2^-2)
##
##       #
##             #
##       for (decade in 1:nDecades) {
##         alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
##       }
##       Alpha[s,d] ~ dunif(0, 1)
##       tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
##       sigma.alpha[s,d] ~  dunif(0, 0.5)
##
##       #
##             #
##       beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
##       Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
##       Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
##     }
##     #
##       #
##     for (p in 1:nP) {
##       for (c in 1:nC) {
##         for (t in 2:nY) {
##           Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
```

```
##          mu.performance[p,c,t,d] <- eta.performance[1,p,d]
##                                   + eta.performance[2,p,d] * resid[1,c,t,d]
##                                   + eta.performance[3,p,d] * Y[1,c,t-1,d]
##                                   + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
##                                   + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
##                                   + eta.performance[6,p,d] * GDPpc[c,t-1]
##                                   + eta.performance[7,p,d] * trade[c,t-1]
##                                   + eta.performance[8,p,d] * eu[c,t-1]
##                                   + eta.performance[9,p,d] * gdp.growth[c,t-1]
##                                   + eta.performance[10,p,d] * urban[c,t-1]
##                                   + eta.performance[11,p,d] * industry[c,t-1]
##        }
##        Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
##        mu.performance[p,c,1,d] <- eta.performance[1,p,d]
##                                 + eta.performance[2,p,d] * resid[1,c,1,d]
##                                 + eta.performance[3,p,d] * Y[1,c,1,d]
##                                 + eta.performance[4,p,d] * portfolio.size[1,c,1]
##                                 + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
##                                 + eta.performance[6,p,d] * GDPpc[c,1]
##                                 + eta.performance[7,p,d] * trade[c,1]
##                                 + eta.performance[8,p,d] * eu[c,1]
##                                 + eta.performance[9,p,d] * gdp.growth[c,1]
##                                 + eta.performance[10,p,d] * urban[c,1]
##                                 + eta.performance[11,p,d] * industry[c,1]
##      }
##      tau.performance[p,d] ~ dgamma(0.001, 0.001)
##      sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
##      for (e in 1:11) {
##        eta.performance[e,p,d] ~ dnorm(0, 1^-2)
##      }
##    }
##
##        for (c in 1:nC) {
##      gamma[c,d] ~ dnorm(0, 0.2^-2)
##    }
##    #
##        #
##    for (c in id.real.countries) {
##      for (s in 1:nS) {
##        rho[s,c,d] ~ dunif(-1, 1)
##      }
##      for (p in 1:nP) {
##        rho.performance[p,c,d] ~ dunif(-1, 1)
##      }
##    }
##    for (c in id.fake.countries) {
##      for (s in 1:nS) {
##        rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##      }
```

```
##       for (p in 1:nP) {
##         rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##       }
##     }
##   }
##
##     for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (t in 2:nY) {
##         portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
##         mu.ps[s,c,t] <- alpha.ps[s,c]
##                        + delta[1,s] * GDPpc[c,t-1]
##                        + delta[2,s] * gov.eff[s,c,t-1]
##                        + delta[3,s] * green.socialist[s,c,t-1]
##                        + delta[4,s] * constraints[c,t-1]
##       }
##       tau.ps[s,c] ~ dgamma(0.1, 0.1)
##       sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
##       alpha.ps[s,c] ~ dnorm(0, 1^-2)
##     }
##     for (d in 1:4) {
##       delta[d,s] ~ dnorm(0, 1^-2)
##     }
##   }
##     for (d in 1:nD) {
##     for (s in 1:nS) {
##       pi[1,s,d] <- delta[1,s] * beta[1,s,d]             pi[2,s,d] <- delta[2,s] * beta[4,s,d]
##   }
##
##
##
##   #
##     #
##   for (c in 1:nC) {
##     for (t in 1:nY) {
##       gov.eff[1,c,t] ~ dnorm(mean(gov.eff.mean.observed[1,c]), 0.05^-2)
##       gov.eff[2,c,t] ~ dnorm(gov.eff[1,c,t], 100)
##       gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
##       urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
##       industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
##     }
##   }
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##             for (t in 1:(nY-1)) {
##         green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
##       }
##       for (d in 1:nD) {
##         for (t in 1:(nY-1)) {
```

```
##             interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
##           }
##         }
##       }
##     }
## }
ggmcmc(ggs(s, family = "sigma|alpha|beta|lambda|rho|nu"),
       param_page = 10, file = paste("ggmcmc-full-", M, ".pdf", sep = ""))
```

5.4 Model results

Variance components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             "Portfolio size",
                                             "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda, label = "Variable", comparison = "Sector") +
  aes(color = Sector) +
  facet_wrap(~ Diversity) +
  theme(legend.position = "right") +
  ggtitle("Variance component")
```

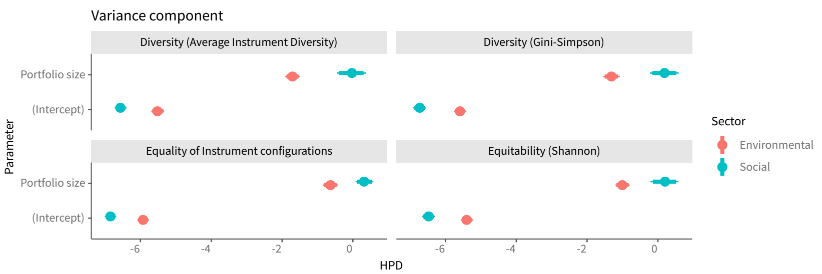

Figure 5.1: Variance component.

```
rm(S.lambda)
L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             "Portfolio size",
                                             "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
S.lambda %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable") +
  ggtitle("Variance component")

rm(S.lambda)
```

Variance components (country varying intercepts).

```
L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.gamma, label = "Country") +
  facet_grid(~ Diversity) +
  ggtitle("Variance component (countries)")
```

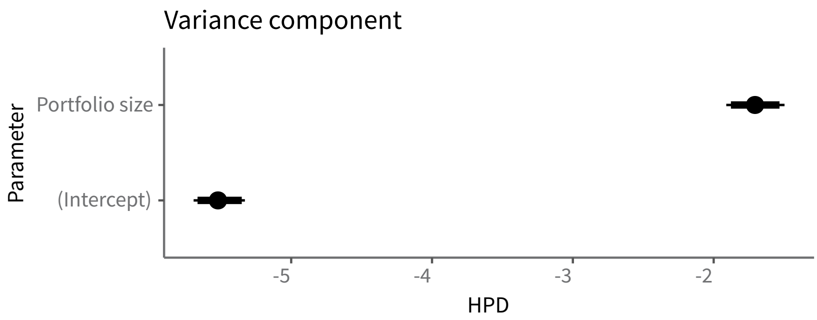

Figure 5.2: Variance component.

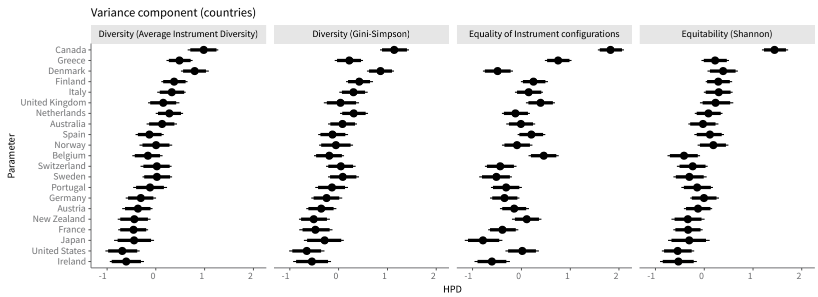

Figure 5.3: Variance component (country varying intercepts).

```
rm(S.gamma)

L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-")) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)")
ggs_caterpillar(S.gamma, label = "Country") +
  ggtitle("Variance component (countries)")
```

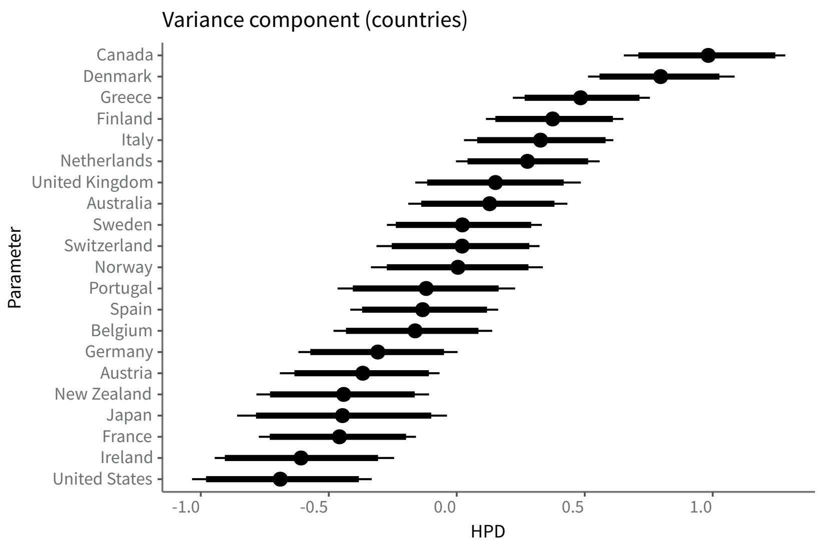

Figure 5.4: Variance component (country varying intercepts).

```
rm(S.gamma)
```

Auto-regressive components.

```r
L.rho <- plab("rho", list(Sector = sector.label,
                          Country = country.label,
                          Diversity = diversity.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  facet_grid(Diversity ~ Sector, scales="free") +
  aes(color=Sector) +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()
```

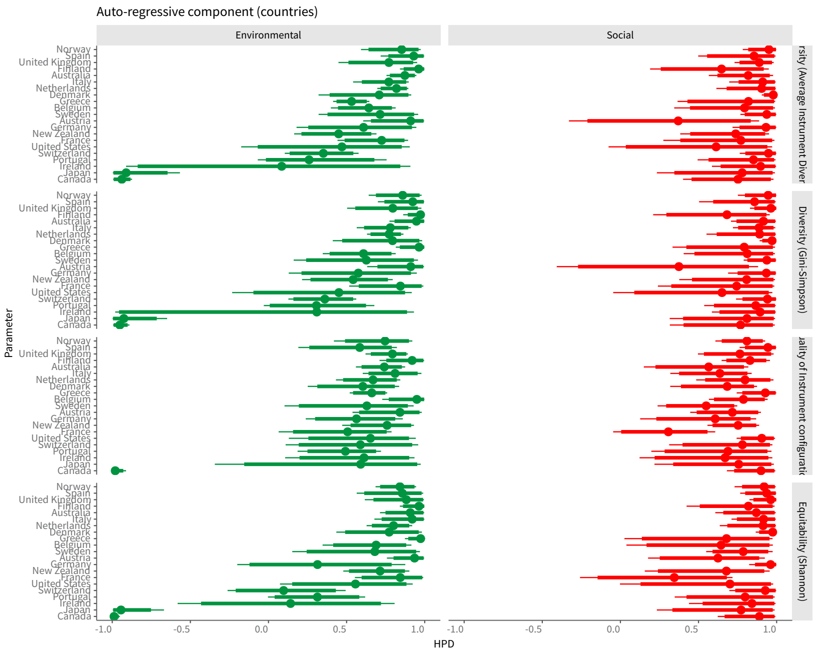

Figure 5.5: Auto-regressive component.

```r
rm(S.rho)

L.rho <- plab("rho", list(Sector = sector.label,
                          Country = country.label,
                          Diversity = diversity.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()


rm(S.rho)
```

Intercepts

```r
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Diversity = diversity.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_label = L.alpha)
ci(S.alpha) %>%
  ggplot(aes(x = Decade, y = median, ymin = Low, ymax = High)) +
```

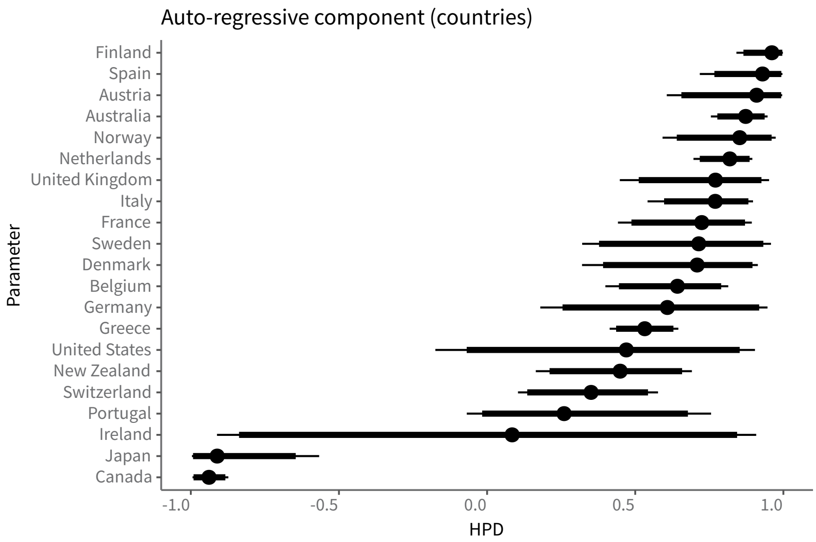

Figure 5.6: Auto-regressive component.

```
  geom_point() +
  geom_linerange() +
  facet_grid(Diversity ~ Sector) +
  ggtitle("Temporal trend")

covariates.label <- c("GDP pc",
                                          "Consensus",
                                          "Deliberation",
                                          "Government effectiveness",
                                          "Portfolio size",
                                          "Trade", "EU", "Salience",
                                          "Political constraints",
                                          "Interdependency (Contiguity)",
                                          "Interdependency (Trade)")

L.betas <- plab("beta", list(Variable = covariates.label,
                             Sector = sector.label,
                        Diversity = diversity.label))

S.betas <- ggs(s, family = "^beta\\[", par_labels = L.betas) %>%
  filter(!Variable %in% c("Deliberation", "Consensus"))
ci.betas <- ci(S.betas)
save(ci.betas, file = paste0("ci-betas-", M, ".RData"))

S.betas %>%
  group_by(Variable, Sector) %>%
  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()


ci(S.betas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

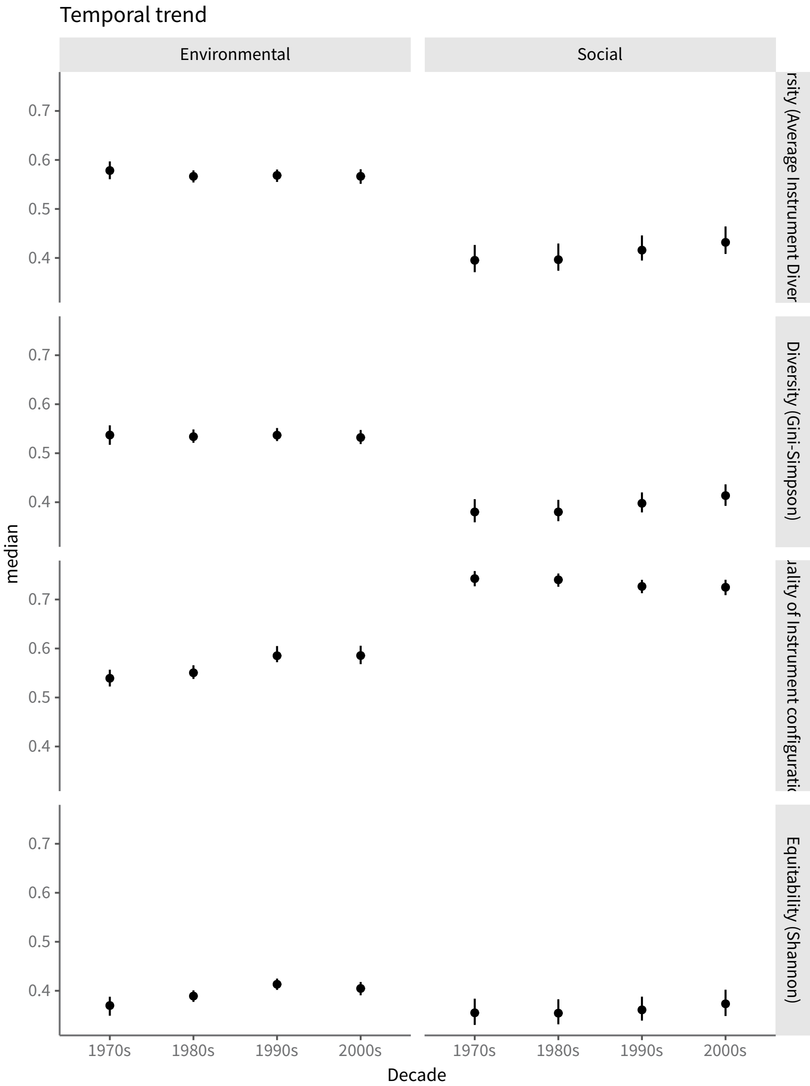

Figure 5.7: Intercepts.

Variable	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	-0.04	0.02	0.00	1.00	-0.04
GDP pc	Social	-0.08	0.05	0.25	0.75	-0.06
Government effectiveness	Environmental	0.40	0.10	1.00	0.00	0.37
Government effectiveness	Social	0.39	0.09	1.00	0.00	0.36
Portfolio size	Environmental	0.15	0.03	1.00	0.00	0.16
Portfolio size	Social	0.13	0.02	1.00	0.00	0.14
Trade	Environmental	-0.01	0.01	0.13	0.87	-0.01
Trade	Social	-0.06	0.01	0.00	1.00	-0.06
EU	Environmental	0.00	0.01	0.38	0.62	0.00
EU	Social	0.01	0.01	0.91	0.09	0.01
Salience	Environmental	-0.02	0.01	0.00	1.00	-0.02
Salience	Social	-0.02	0.01	0.12	0.88	-0.02
Political constraints	Environmental	-0.06	0.02	0.04	0.96	-0.05
Political constraints	Social	-0.12	0.03	0.00	1.00	-0.11
Interdependency (Contiguity)	Environmental	0.06	0.02	1.00	0.00	0.06
Interdependency (Contiguity)	Social	-0.02	0.02	0.31	0.69	-0.02
Interdependency (Trade)	Environmental	0.02	0.03	0.72	0.28	0.00
Interdependency (Trade)	Social	0.00	0.01	0.36	0.64	0.00

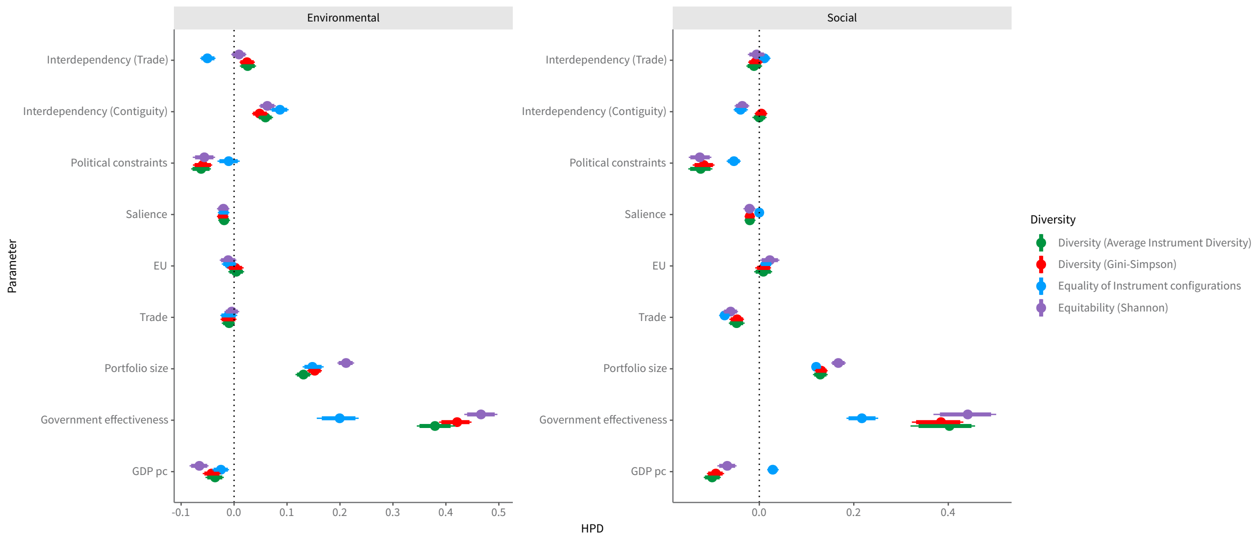

Figure 5.8: Slopes with the effects on diversity, by sector.

```
ci(S.betas) %>%
  filter(Sector == "Environmental") %>%
ggplot(aes(x = Variable,
           y = median,
           group = Diversity,
           color = Diversity)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

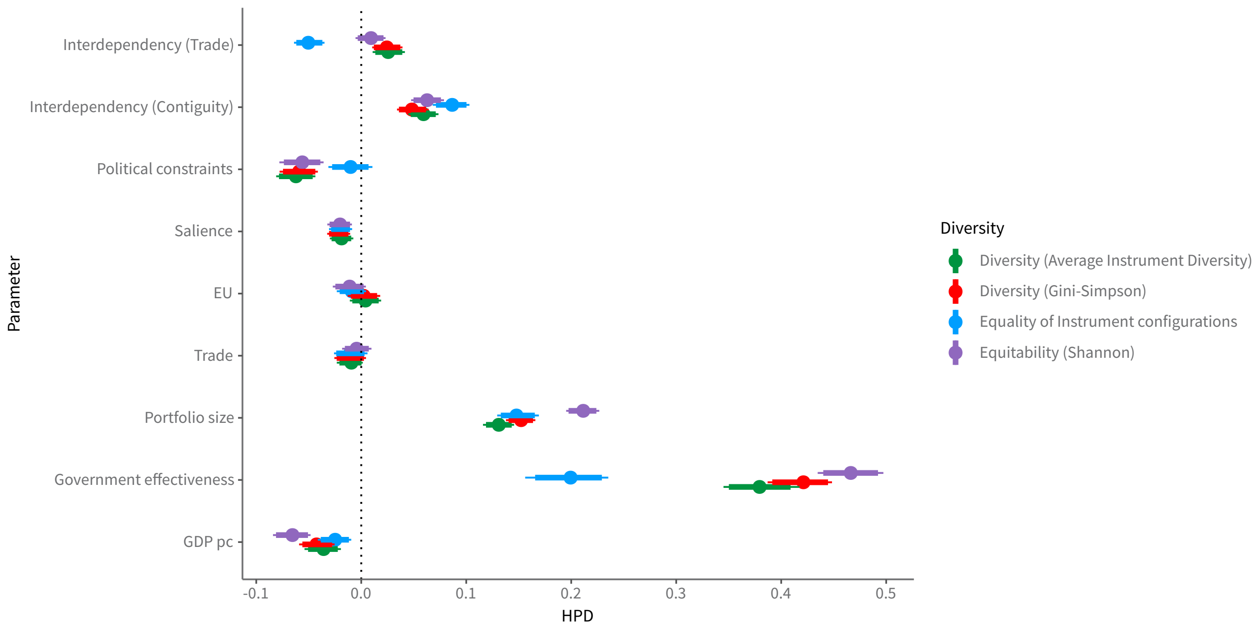

Figure 5.9: Slopes with the effects on diversity, for the environmental sector.

```
ci(S.betas) %>%
ggplot(aes(x = Variable,
```

```
              y = median,
              group = Sector, color = Sector)) +
  coord_flip() +
  facet_wrap(~ Diversity, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

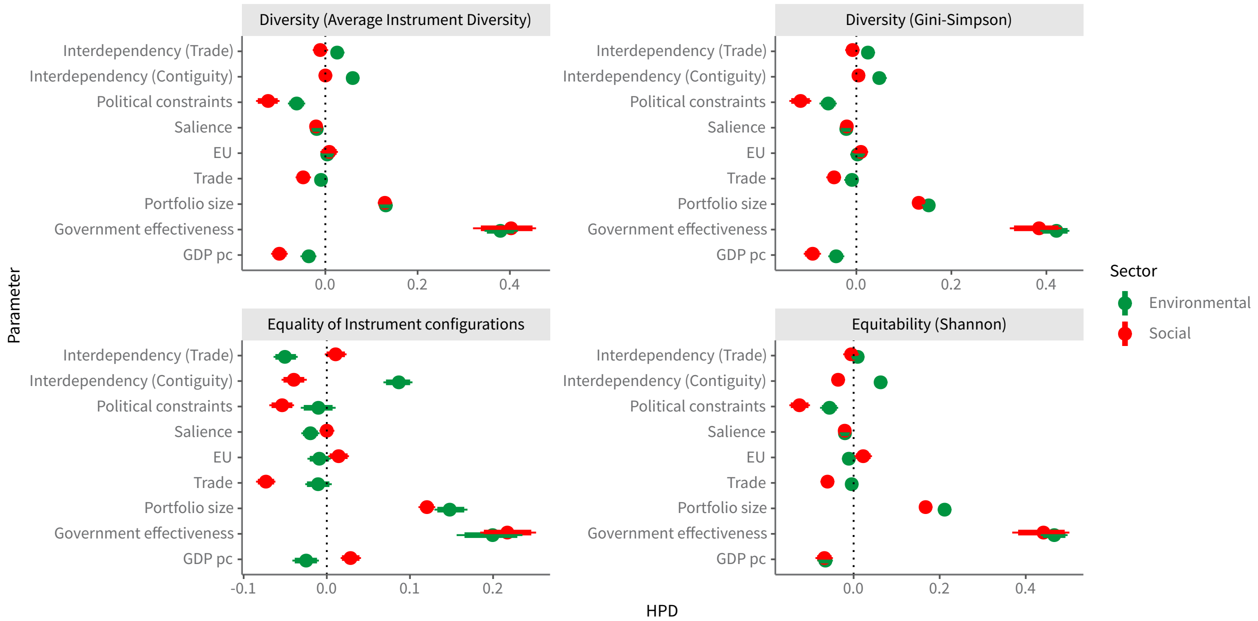

Figure 5.10: Slopes with the effects on diversity, by diversity indicator.

```
ci(S.betas) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggplot(aes(x = Variable,
              y = median,
              group = Sector, color = Sector)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)


ggplot(S.betas, aes(x = value, color = Sector, fill = Sector)) +
  geom_density(alpha = 0.5) +
  facet_grid(Diversity ~ Variable, scales = "free") +
  xlab("HPD") +
  geom_vline(xintercept = 0, lty = 3)
```

Variables by evidence.

```
S.betas %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(value != 0) %>%
  group_by(Sector, Diversity, Variable) %>%
  summarize(`Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
    `Mean expected effect` = mean(value)) %>%
  group_by(Sector, Diversity, Variable) %>%
```

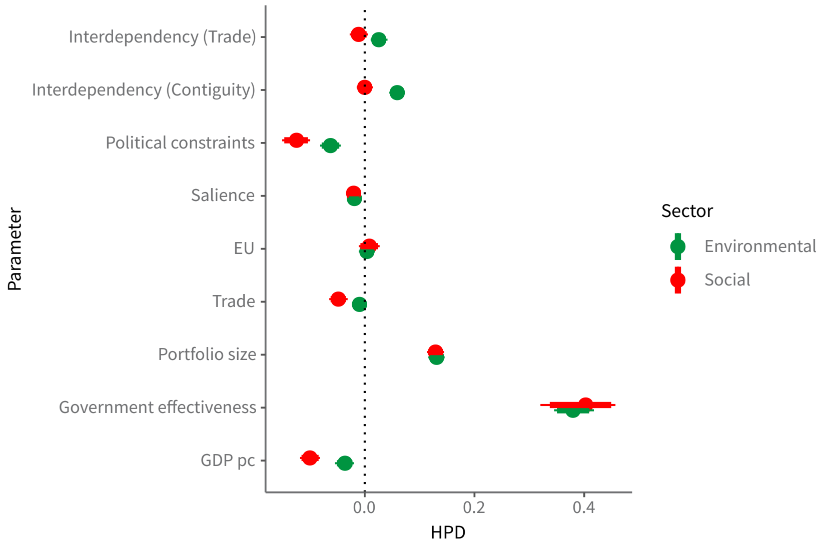

Figure 5.11: Slopes with the effects on AID diversity, by sector

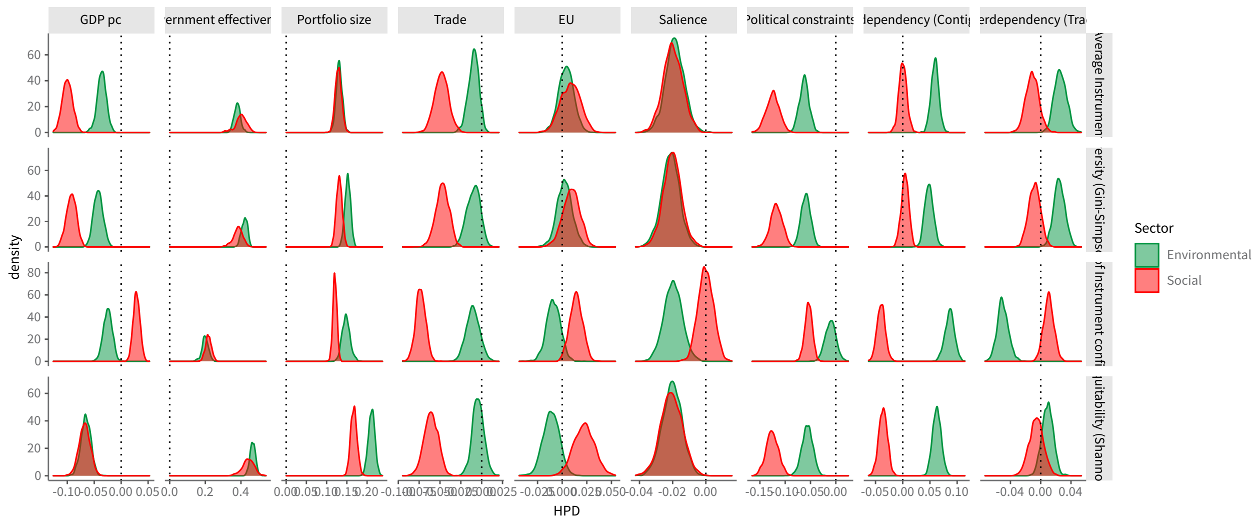

Figure 5.12: Slopes with the effects on diversity, comparing sectors.

```
  mutate(max = max(abs(`Prob > 0`), abs(`Prob < 0`))) %>%
  arrange(desc(max)) %>%
  select(-max, -Diversity) %>%
  kable()
```

Diversity	Sector	Variable	Prob > 0	Prob < 0	Mean expected effect
Diversity (Average Instrument Diversity)	Environmental	GDP pc	0.00	1.00	-0.04
Diversity (Average Instrument Diversity)	Environmental	Government effectiveness	1.00	0.00	0.38
Diversity (Average Instrument Diversity)	Environmental	Portfolio size	1.00	0.00	0.13
Diversity (Average Instrument Diversity)	Environmental	Political constraints	0.00	1.00	-0.06
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Contiguity)	1.00	0.00	0.06
Diversity (Average Instrument Diversity)	Social	GDP pc	0.00	1.00	-0.10
Diversity (Average Instrument Diversity)	Social	Government effectiveness	1.00	0.00	0.40
Diversity (Average Instrument Diversity)	Social	Portfolio size	1.00	0.00	0.13
Diversity (Average Instrument Diversity)	Social	Trade	0.00	1.00	-0.05
Diversity (Average Instrument Diversity)	Social	Political constraints	0.00	1.00	-0.12
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Trade)	1.00	0.00	0.03
Diversity (Average Instrument Diversity)	Social	Salience	0.00	1.00	-0.02
Diversity (Average Instrument Diversity)	Environmental	Salience	0.00	1.00	-0.02
Diversity (Average Instrument Diversity)	Environmental	Trade	0.06	0.94	-0.01
Diversity (Average Instrument Diversity)	Social	Interdependency (Trade)	0.10	0.90	-0.01
Diversity (Average Instrument Diversity)	Social	EU	0.79	0.21	0.01
Diversity (Average Instrument Diversity)	Environmental	EU	0.70	0.30	0.00
Diversity (Average Instrument Diversity)	Social	Interdependency (Contiguity)	0.50	0.50	0.00

```
S.betas.env <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) #%>%

ggs_caterpillar(S.betas.env) +
  geom_vline(xintercept = 0, lty = 3)
```

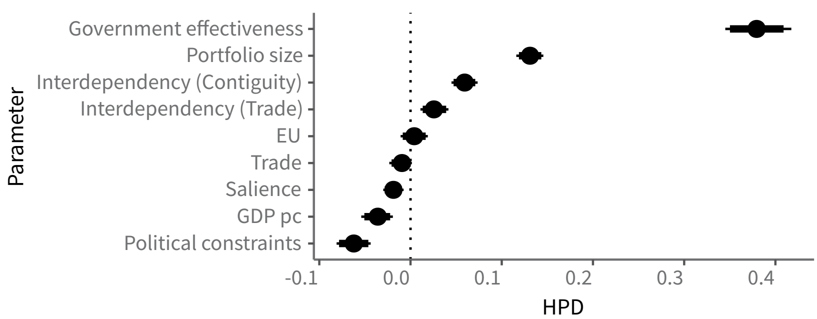

Figure 5.13: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env)

S.betas.env.sorted <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) %>%
  mutate(Parameter = fct_relevel(Parameter, rev(c("Political constraints",
                                                  "Government effectiveness",
                                  "Salience",
                                  "GDP pc", "Trade", "EU",
                                  "Interdependency (Contiguity)",
                                  "Interdependency (Trade)",
                                  "Portfolio size"))))

ggs_caterpillar(S.betas.env.sorted, sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)
```

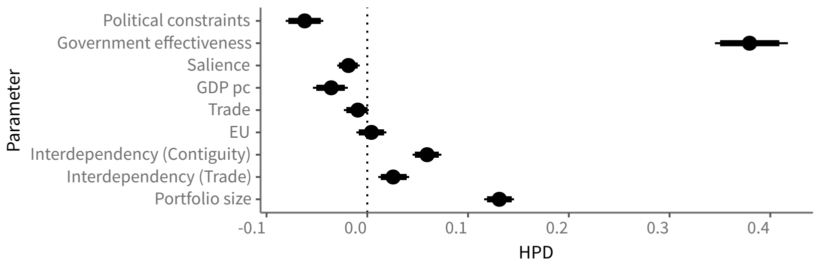

Figure 5.14: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env.sorted)

ci.betas <- ci(S.betas) %>%
  mutate(Model = M.lab)

save(ci.betas, file = paste0("ci_betas-", M, ".RData"))
rm(S.betas, ci.betas)

L.deltas <- plab("delta", list(Variable = c("GDP pc",
                                            "Government effectiveness",
                                            "Salience",
                                            "Political constraints"),
                        Sector = sector.label))

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

S.deltas %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	1.41	0.07	1.00	0.00	1.41
GDP pc	Social	0.45	0.04	1.00	0.00	0.45
Government effectiveness	Environmental	0.10	0.09	0.85	0.15	0.10
Government effectiveness	Social	-0.19	0.08	0.04	0.96	-0.18
Salience	Environmental	0.13	0.03	1.00	0.00	0.13
Salience	Social	-0.04	0.02	0.01	0.99	-0.04
Political constraints	Environmental	0.14	0.06	0.99	0.01	0.13
Political constraints	Social	0.22	0.06	1.00	0.00	0.22

```
ci(S.deltas) %>%
ggplot(aes(x = Variable,
           y = median,
           group = Sector)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)

rm(S.deltas)

L.pi <- plab("pi", list(Variable = c("GDP pc",
                                     "Government effectiveness",
                                     "Salience",
                                     "Political constraints"),
                        Sector = sector.label,
```

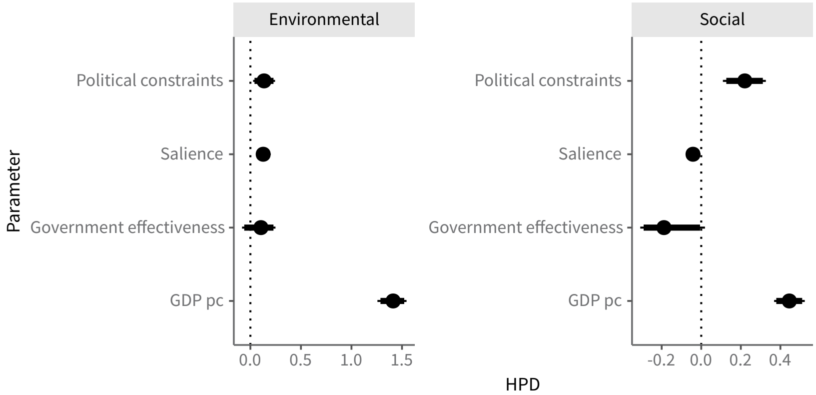

Figure 5.15: Slopes with the direct effects on portfolio size, by sector.

```
                      Diversity = diversity.label))
S.pis ← ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable ≠ "GDP pc")
S.pis %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Government effectiveness	Environmental	0.03	0.03	0.85	0.15	0.04
Government effectiveness	Social	-0.06	0.03	0.04	0.96	-0.06
Salience	Environmental	0.00	0.00	0.00	1.00	0.00
Salience	Social	0.00	0.00	0.87	0.13	0.00
Political constraints	Environmental	-0.01	0.00	0.05	0.95	-0.01
Political constraints	Social	-0.02	0.01	0.00	1.00	-0.02

```
ci(S.pis) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()

L.pi ← plab("pi", list(Variable = c("GDP pc",
                                    "Government effectiveness",
                                    "Salience",
                                    "Political constraints"),
                      Sector = sector.label,
                      Diversity = diversity.label))
S.pis ← ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable ≠ "GDP pc")
S.pis %>%
```

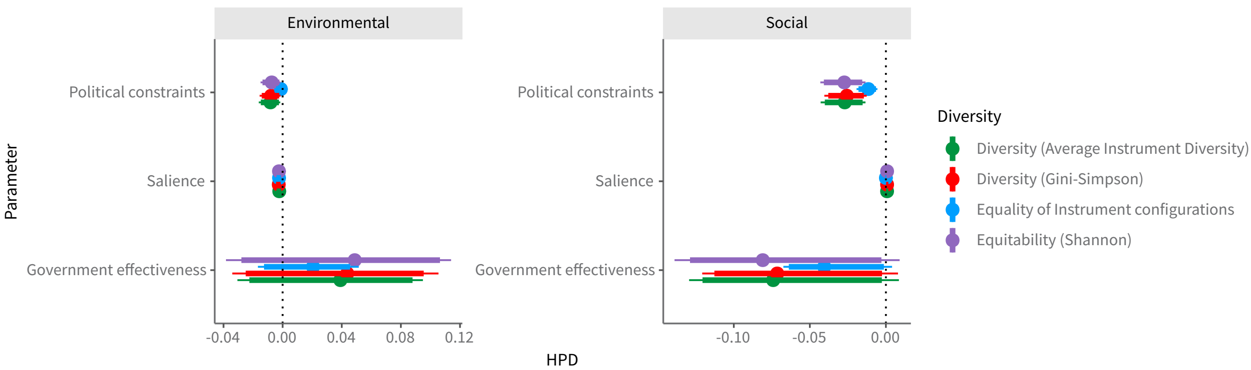

Figure 5.16: Mediated effects on diversity, by sector.

```
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Government effectiveness	Environmental	0.03	0.03	0.85	0.15	0.04
Government effectiveness	Social	-0.06	0.03	0.04	0.96	-0.06
Salience	Environmental	0.00	0.00	0.00	1.00	0.00
Salience	Social	0.00	0.00	0.87	0.13	0.00
Political constraints	Environmental	-0.01	0.00	0.05	0.95	-0.01
Political constraints	Social	-0.02	0.01	0.00	1.00	-0.02

```
S.pis %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable")
```

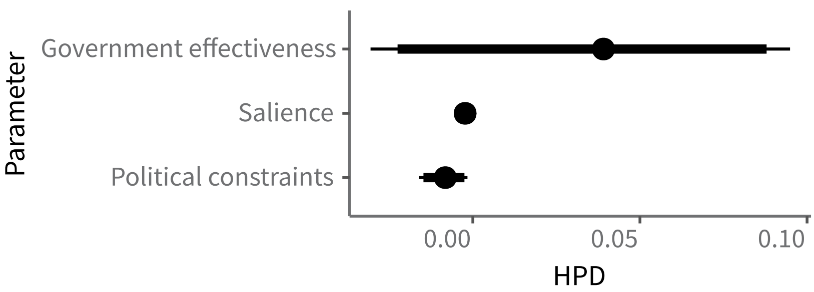

Figure 5.17: Mediated effects on diversity, for AID and the environmental sector.

```
rm(S.pis)

L.eta <- plab("eta.performance", list(Covariate = c("(Intercept)", "Residuals",
                                                   "Diversity", "Portfolio size",
                                                   "Diversity * Portfolio size",
                                                   "GDPpc", "Trade", "EU",
                                                   "GDP growth", "Urban", "Industry"),
                                      Performance = performance.label,
                                      Diversity = diversity.label))
S.eta <- ggs(s, family = "eta.performance", par_label = L.eta)

ggs_caterpillar(S.eta, label = "Covariate", comparison = "Performance") +
```

```
  geom_hline(yintercept = 0, lty = 3) +
  aes(color = Performance) +
  theme(legend.position = "right") +
  facet_wrap(~ Diversity)
```

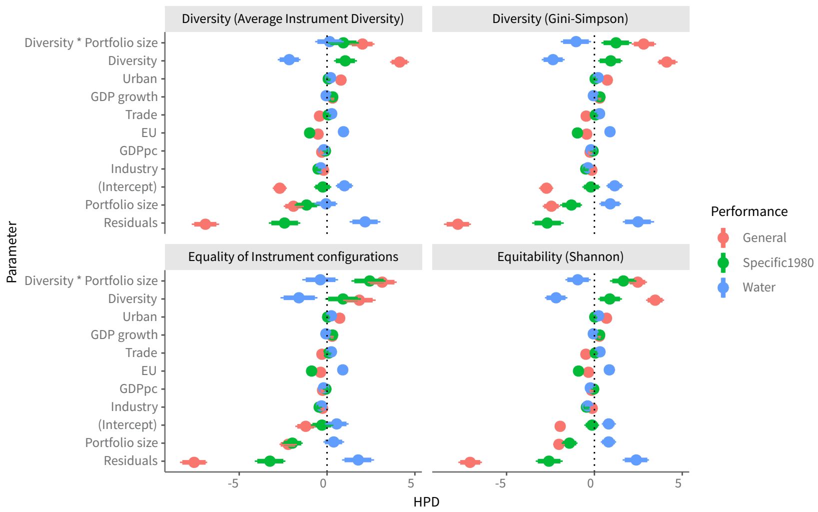

Figure 5.18: Direct effects on Environmental performance, by performance indicator and Diversity measure.

```
S.eta %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Performance == "General") %>%
ggs_caterpillar(label = "Covariate") +
  geom_vline(xintercept = 0, lty = 3)
```

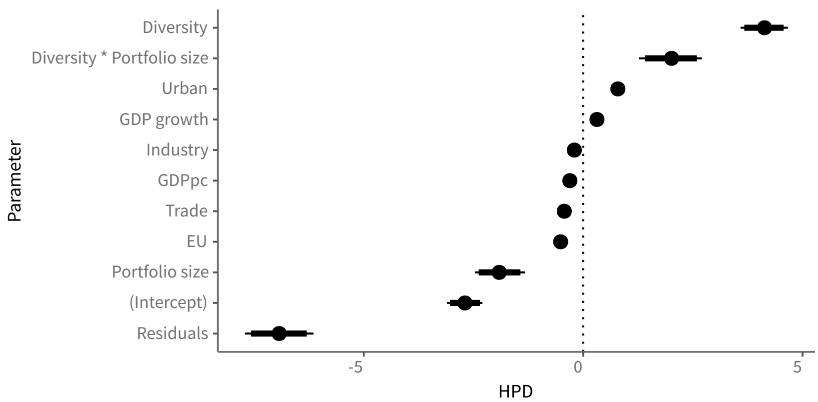

Figure 5.19: Direct effects on Environmental performance, for general performance and using AID as diversity.

```
rm(S.eta)
```

5.5 V-Cov

```
L.Sigma <- plab("Sigma", list(Covariate.1 = covariates.label,
                              Covariate.2 = covariates.label,
                              Sector = sector.label,
                              Diversity = diversity.label))
S.Sigma <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma) %>%
  filter(!Covariate.1 %in% c("Deliberation", "Consensus")) %>%
  filter(!Covariate.2 %in% c("Deliberation", "Consensus"))
```

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

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

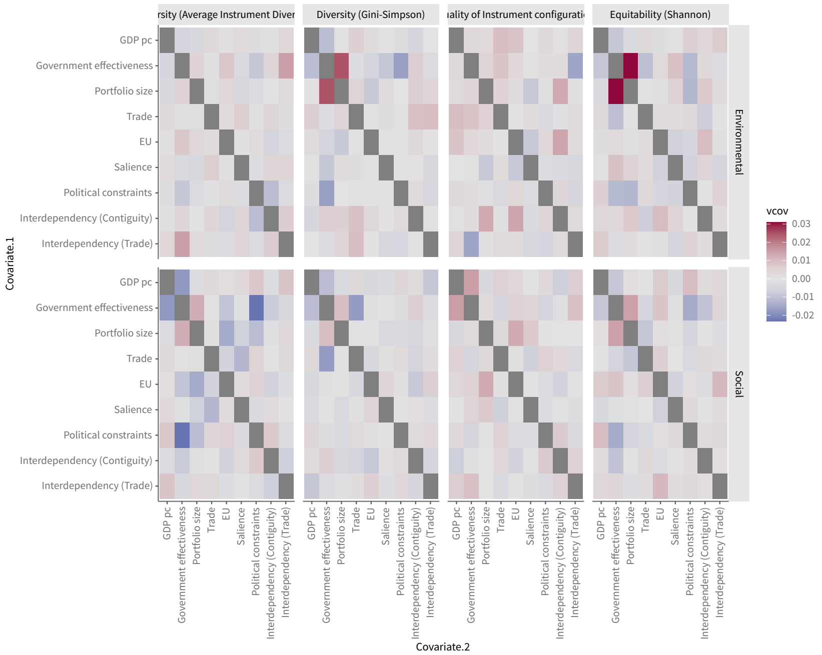

Figure 5.20: Variance-covariance matrix of main effects.

```
rm(S.Sigma, vcov.Sigma)
```

5.6 Model evaluation

What is the model fit?

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

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1, Country = Var2,
         Year = Var3, Diversity = Var4,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector, Diversity) %>%
  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, Diversity) %>%
  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(Diversity ~ Sector) +
  expand_limits(x = 0)
```

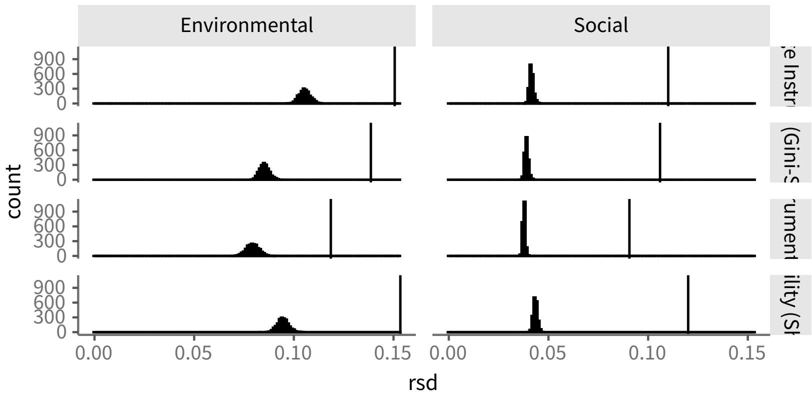

Figure 5.21: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

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

Sector	Diversity	Pseudo.R2
Environmental	Diversity (Average Instrument Diversity)	0.30
Environmental	Diversity (Gini-Simpson)	0.39
Environmental	Equality of Instrument configurations	0.33
Environmental	Equitability (Shannon)	0.38
Social	Diversity (Average Instrument Diversity)	0.62
Social	Diversity (Gini-Simpson)	0.63
Social	Equality of Instrument configurations	0.58
Social	Equitability (Shannon)	0.64

```
rm(S.rsd)
```

Which are the observations with higher residuals, further away from the expectation?[1]

[1] Negative residuals are cases where the country has lower diversity than expected according to the model.

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

ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(!str_detect(Country, "Z-")) %>%
  group_by(Country) %>%
  summarize(Residual = mean(value)) %>%
  mutate(`Mean absolute residual` = abs(Residual)) %>%
  ungroup() %>%
  arrange(desc(`Mean absolute residual`)) %>%
  select(-`Mean absolute residual`) %>%
  slice(1:8) %>%
  kable()
```

Country	Residual
Canada	-0.18
Greece	0.14
Australia	0.11
Finland	-0.10
Netherlands	-0.08
Norway	0.07
Portugal	0.06
Denmark	0.05

5.7 Magnitude of effects

```
L.mu <- plab("mu", list(Sector = sector.label,
                        Country = country.label,
                        Year = year.label.numeric,
                        Diversity = diversity.label)) %>%
  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()

ci.mu %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  ggplot(aes(x = Year, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen portfolio size goes\nfrom minimum to maximum\nover time")
```

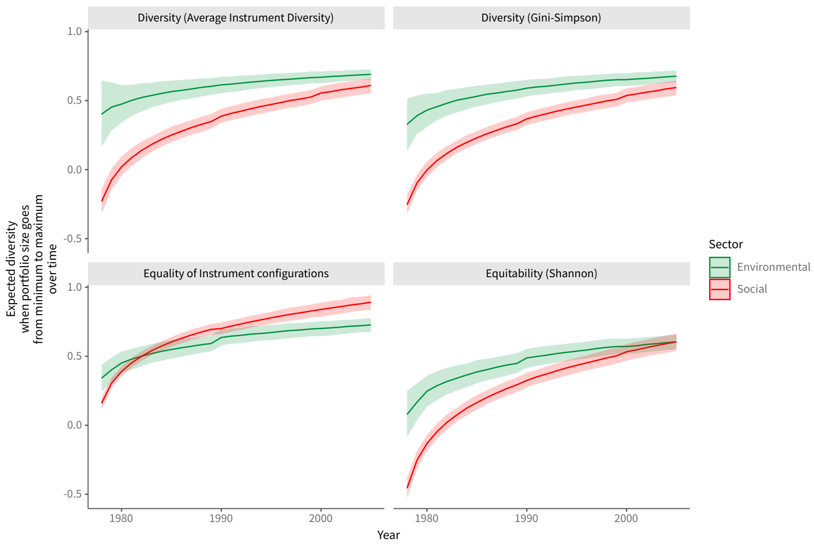

Figure 5.22: Magnitude of the effects: change in the expected diversity when portfolio size goes from the minimum to the maximum observed value over time, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year == max(Year)) %>%
  left_join(gov.eff.df %>%
            rename(Country = country, Year = year)) %>%
  rename(`Government effectiveness` = value) %>%
  ggplot(aes(x = `Government effectiveness`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
```

```
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen government effectiveness goes\nfrom minimum to maximum")
```

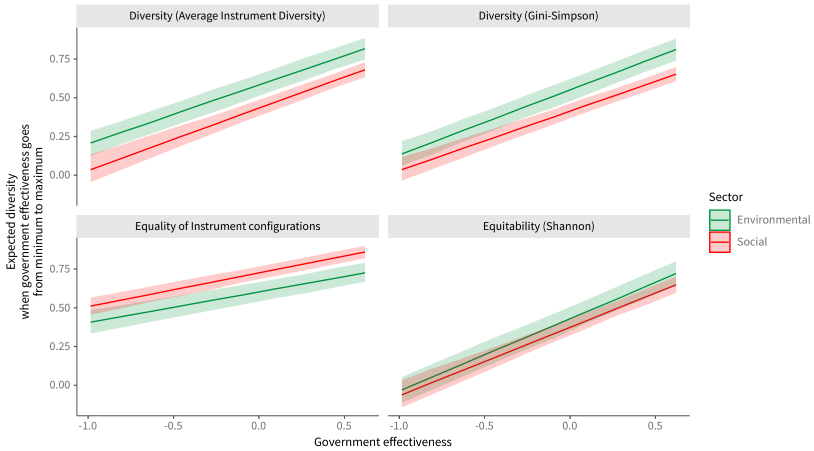

Figure 5.23: Magnitude of the effects: change in the expected diversity when government effectiveness goes from the minimum to the maximum observed value, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year == max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = polcon) %>%
  ggplot(aes(x = `Political constraints`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen political constraints go\nfrom minimum to maximum")
```

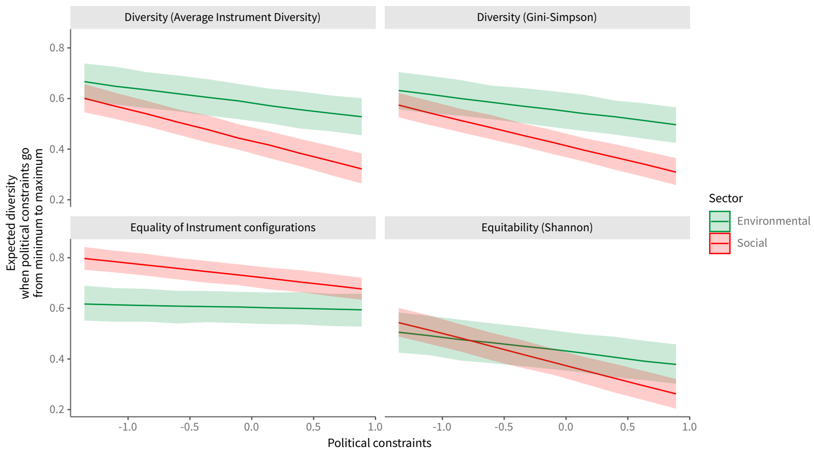

Figure 5.24: Magnitude of the effects: change in the expected diversity when political constraints go from the minimum to the maximum observed value, with the rest of the variables at their means.

```
f1 <-
  ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  left_join(D %>%
            filter(Measure == "Size") %>%
            select(Country, Sector, Year, value) %>%
```

```
            rename(`Portfolio size` = value)) %>%
  ggplot(aes(x = `Portfolio size`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(a)") +
  ylab("Expected diversity")

f2 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year ==  max(Year)) %>%
  left_join(gov.eff.df %>%
            rename(Country = country, Year = year)) %>%
  rename(`Government effectiveness` = value) %>%
  ggplot(aes(x = `Government effectiveness`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(b)") +
  ylab("Expected diversity")

f3 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year ==  max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = original.polcon) %>%
  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, 1)) +
  ggtitle("(c)") +
  ylab("Expected diversity")

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

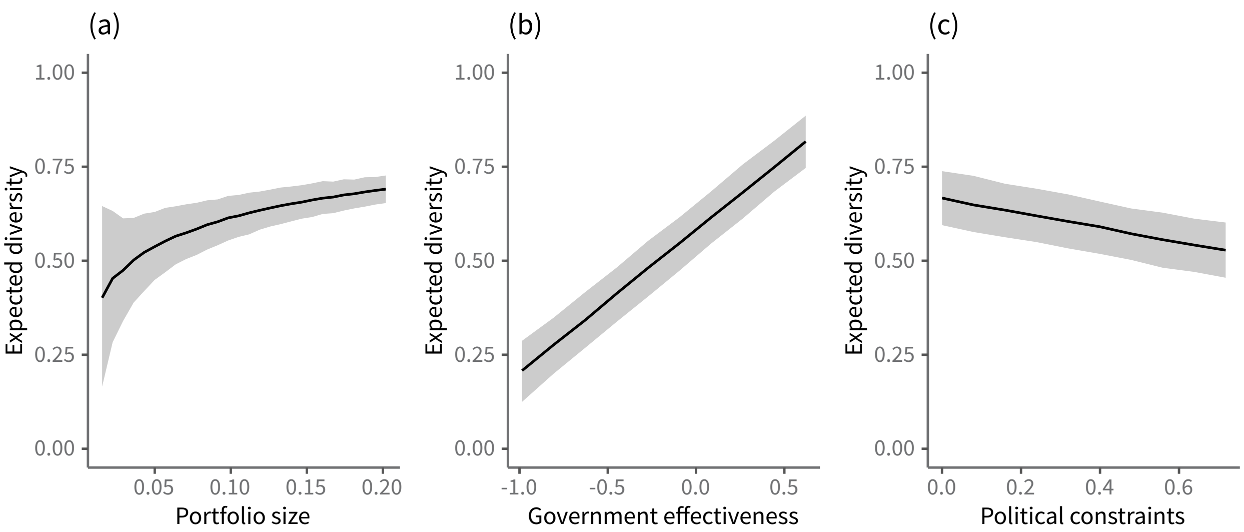

Figure 5.25: Magnitude of the effects: change in the expected diversity when (a) portfolio size, (b) government effectiveness or (c) political constraints move from the minimum to the maximum observed. In all cases the rest of the variables are fixed at their means.

6

Explanatory model of instrument diversity (robustness, VPI)

Data is TSCS, by sector.

```r
library(PolicyPortfolios)
data(consensus)

D <- bind_rows(
  # Calculate portfolio measures sector by sector
  consensus %>%
    filter(Sector == "Environmental") %>%
    droplevels() %>%
    pp_measures(),
  consensus %>%
    filter(Sector == "Social") %>%
    droplevels() %>%
    pp_measures())


# Add more fake countries

# Z-01 Increases portfolio size from minimum to maximum observed size
#      over time
range.size.env <- range(filter(D, Sector == "Environmental" & Measure == "Size")$value)
range.size.soc <- range(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Environmental", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.env[1], range.size.env[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Social", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.soc[1], range.size.soc[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

# Z-02, Z-11 Fixes government effectiveness to its mean
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))


# Z-12:Z-21 Fixes portfolio size for veto players to move
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)
```

```
D ← bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D ← bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))
```

```
diversity ← D %>%
  mutate(Country = as.factor(Country)) %>%
  select(-Measure) %>%
  rename(Measure = Measure.label) %>%
  spread(Measure, value) %>%
  mutate(`Diversity/W (Average Instrument Diversity)` =
         `Diversity (Average Instrument Diversity)` / (1 - `Proportion of targets covered`)) %>%
  select(Sector, Country, Year,
    `Diversity (Average Instrument Diversity)`,
    `Diversity (Gini-Simpson)`,
    `Equitability (Shannon)`,
    `Equality of Instrument configurations`)

countries ← as.character(unique(D$Country))
nC ← length(countries)
years ← range(D$Year)
```

6.1 Performance

```
perf ← foreign::read.dta("jahn/PoEP_Replication_Data/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% 1980:2010)

d.perf ← perf %>%
  # Delete Years for which we don't have data
  filter(Year ≥ 1980 & Year ≤ 2005) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Water", "Specific1980")) %>%
  droplevels()

Y.performance ← reshape2::acast(d.perf, Indicator ~ Country ~ Year, value.var = "Performance")
nYperformance ← dim(Y.performance)[3]
```

6.2 Covariates

World Development Indicators - Revenue.

```
load("wdi/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("wdi/wdi.RData")
wdi <- wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi <- subset(wdi, year >= 1976 & year <= 2005)
wdi$country[wdi$country=="Korea, Rep."] <- "South Korea"
wdi <- subset(wdi, country %in% countries)

# GDP pc in Ireland is bad
ireland.wdi <- subset(wdi, country=="Ireland")
#ireland.wdi

# 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(pred

#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(


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(pred

# 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
wdi.gcr.w <- cbind(wdi.gcr.w, gdpc.ratio=wdi.gcr.w$`2005`/wdi.gcr.w$`1976`)


# 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)
```

Vertical Policy Integration.

```
vpi.d <- as_tibble(read.table("vpi/vpi-raw.csv", header = TRUE))

# Multiply observations by year
vpi.d <- expand_grid(vpi.d,
                     Year = min(D$Year):max(D$Year))
```

Add fake vertical policy integration.

```
range.vpi.env <- range(filter(vpi.d, Sector == "Environmental")$VPI)
range.vpi.soc <- range(filter(vpi.d, Sector == "Social")$VPI)

vpi.d <- vpi.d %>%
  bind_rows(expand_grid(tibble(Sector = "Environmental",
                              Country = paste0("Z-", sprintf("%02d", 2:11)),
                              VPI = seq(range.vpi.env[1],
                                        range.vpi.env[2],
                                        length.out = length(2:11))),
                        Year = min(D$Year):max(D$Year)))

vpi.d <- vpi.d %>%
  bind_rows(expand_grid(tibble(Sector = "Social",
                              Country = paste0("Z-", sprintf("%02d", 2:11)),
                              VPI = seq(range.vpi.soc[1],
                                        range.vpi.soc[2],
                                        length.out = length(2:11))),
                        Year = min(D$Year):max(D$Year)))
```

Political Constraints - polconIII. An indicator of “veto players” comes from Henisz (2002). The indicator “estimates the feasibility of policy change. [That is], the extent to which a change in the preferences of any one actor may lead to a change in government policy”. Higher values represent systems with higher constraints.

```
load("polcon/polcon2017.RData") # loads polcon
```

Add fake political constraints values.

```
polcon.d <- polcon %>%
  as_tibble() %>%
  filter(year %in% years[1]:years[2]) %>%
  mutate(country.polity = as.character(country.polity)) %>%
  mutate(country.polity = ifelse(country.polity == "Germany West", "Germany", country.polity)) %>%
  filter(country.polity %in% countries) %>%
  select(Country = country.polity, Year = year, polcon)

range.polcon <- polcon.d %>%
  select(polcon) %>%
  unlist(., use.names = FALSE) %>%
  range()

polcon.d <- polcon.d %>%
  bind_rows(expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                              polcon = seq(range.polcon[1],
                                           range.polcon[2],
                                           length.out = length(12:21))),
                         Year = min(D$Year):max(D$Year)))
```

For the "Green parties", data comes from Volkens (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("manifesto/cpm-consensus.RData")
cpm$country <- as.character(cpm$country)
cpm$country[cpm$country=="Korea"] <- "South Korea"
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<="2005-12-31")

# 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, as.numeric(diff(c(as.Date("1976-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}
```

For the salience of each topic, we employ data from Volkens (2013), weighting the proportion of votes to each party and the importance that each party gives to environmental issues or the expansion of social welfare.

```
load("manifesto/201029-cpm-salience.RData")

salience <- cpm.salience %>%
  filter(Country %in% countries) %>%
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  filter(Year %in% years[1]:years[2])
```

Border contiguity

```
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% countries,
                       dimnames(M.borders)[[2]] %in% countries]
```

Trade dependency

```
load("trade/trade.RData")

rm(M.trade, M.trade.imports)
```

Save and arrange for analysis.

```
diversity <- diversity %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("South Korea", "Turkey")) %>%
```

```
  droplevels()
diversity.l <- diversity %>%
  gather(Measure, value, -c(Sector, Country, Year))
Y <- reshape2::acast(diversity.l,
                     Sector ~ Country ~ Year ~ Measure, value.var = "value")
nS <- dim(Y)[1]
nC <- dim(Y)[2]
nY <- dim(Y)[3]
nD <- dim(Y)[4]

country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.fake <- 21
nC.real <- nC - nC.fake
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)

year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(as.numeric(year.label))
nY <- length(year.label)

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)

diversity.label <- dimnames(Y)[[4]]
nD <- length(diversity.label)

nB <- 11
b0 <- rep(0, nB)
B0 <- diag(nB)
diag(B0) <- 1^-2

# 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("get-eu_time.R") # generates eu.ms

# Fake countries
f.c <- paste0("Z-", sprintf("%02d", 1:21))
# Fake years
f.y <- year.label.numeric

GDPpc <- wdi %>%
  select(country, year, gdp.capita) %>%
  filter(country %in% country.label) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, 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")

trade.df <- wdi %>%
  select(country, year, trade) %>%
  filter(country %in% country.label) %>%
#  bind_rows(expand_grid(country = f.c, year = f.y, trade = 0)) %>%
  mutate(trade = std(trade)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, trade = 0))
trade <- trade.df %>%
  reshape2::acast(country ~ year, value.var = "trade")
if ( length(which(!dimnames(trade)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

constraints.d <- expand_grid(Country = country.label,
                             Year = year.label.numeric) %>%
  left_join(polcon.d) %>%
  mutate(original.polcon = polcon) %>%
  mutate(polcon = std(polcon)) %>%
  mutate(polcon = ifelse(str_detect(Country, "^Z-") & is.na(polcon), 0, polcon))
```

```r
constraints <- constraints.d %>%
  reshape2::acast(Country ~ Year, value.var = "polcon")
if ( length(which(!dimnames(constraints)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

vpi.df <- expand_grid(Sector = c("Environmental", "Social"),
                      Country = country.label,
                      Year = year.label.numeric) %>%
  left_join(vpi.d) %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  mutate(value = std(VPI)) %>%
  select(Sector, Country, Year, VPI, value) %>%
#  mutate(value = ifelse(str_detect(Country, "^Z-01"), 0, value))
  mutate(value = ifelse(str_detect(Country, "^Z-") & is.na(value), 0, value))
vpi.A <- vpi.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "value")
if ( length(which(!dimnames(vpi.A)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")

min.discard.zero <- function(x) return(min(x[x!=0]))
portfolio.size <-
  D %>%
  filter(Measure == "Size") %>%
  spread(Measure, value) %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector) %>%
  mutate(Size = ifelse(Size == 0, min.discard.zero(Size)/2, Size)) %>%
  mutate(Size = std(logit(Size))) %>%
  mutate(Size = ifelse(str_detect(Country, "Z2"), 0, Size)) %>%
  ungroup() %>%
  select(Country, Sector, Year, Size) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Size")

if ( length(which(!dimnames(portfolio.size)[[2]] == 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")

# Green socialist as salience
green.socialist <- salience %>%
  group_by(Sector) %>%
  mutate(Salience = std(Salience)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label, Country = f.c, Year = f.y, Salience = 0)) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Salience")
if ( length(which(!dimnames(green.socialist)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here

# Borders in tidy data
interdependency.contiguity <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(geography %>%
              select(Origin, Destination, p.contiguous),
              by = c("Destination" = "Destination")) %>%
  mutate(wDiversity = value * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
```

```r
  summarize(contiguity.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(contiguity.dependency = std(contiguity.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        contiguity.dependency = 0))

interdependency.contiguity <- interdependency.contiguity %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "contiguity.dependency")
if ( length(which(!dimnames(interdependency.contiguity)[[3]] == country.label)) > 0) stop("Ep! There is a m

# Trade in tidy data
interdependency.trade <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wDiversity = value * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
  summarize(trade.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(trade.dependency = std(trade.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        trade.dependency = 0))

interdependency.trade <- interdependency.trade %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "trade.dependency")
if ( length(which(!dimnames(interdependency.trade)[[3]] == country.label)) > 0) stop("Ep! There is a mistak


# Performance part
# Match it with the general Y, but only for the environmental sector
d.perf.fake <- expand_grid(
  Country = country.label[str_detect(country.label, "^Z-") ],
  Year = year.label.numeric,
  Indicator = unique(d.perf$Indicator),
  Performance = NA)

Y.performance <- diversity.l %>%
  select(Sector, Country, Year, Measure) %>%
  left_join(d.perf) %>%
  left_join(d.perf.fake) %>%
  filter(Sector == "Environmental") %>%
  group_by(Indicator) %>%
  mutate(Performance = std1(Performance)) %>%
  ungroup() %>%
  reshape2::acast(Indicator ~ Country ~ Year ~ Measure, value.var = "Performance")
# manually get rid of ghost performance for missing values
Y.performance <- Y.performance[-4,,,]

nP <- dim(Y.performance)[1]
performance.label <- dimnames(Y.performance)[[1]]
if ( length(which(!dimnames(Y.performance)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")
```

```r
load("wdi/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 %in% year.label.numeric) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "gdp.growth")
gdp.growth <- gdp.growth[,-dim(gdp.growth)[2]]
if ( length(which(!dimnames(gdp.growth)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
gdp.growth.means <- apply(gdp.growth, 1, mean, na.rm = TRUE)
gdp.growth.means[is.nan(gdp.growth.means)] <- 0

load("wdi/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(urban = std(urban)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
urban <- urban[,-dim(urban)[2]]
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
urban.means <- apply(urban, 1, mean, na.rm = TRUE)
urban.means[is.nan(urban.means)] <- 0

load("wdi/wdi-industry.RData") # industry
industry <- industry %>%
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(industry = std(industry)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "industry")
industry <- industry[,-dim(industry)[2]]
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)
industry.means[is.nan(industry.means)] <- 0


DP <- list(
  Y = unname(Y),
  Y.performance = unname(Y.performance), nP = nP,
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  constraints = unname(constraints),
  vpi = unname(vpi.A),
  interdependency.contiguity = unname(interdependency.contiguity),
  interdependency.trade = unname(interdependency.trade),
  portfolio.size = unname(portfolio.size),
  eu = unname(eu),
  green.socialist = unname(green.socialist),
  gdp.growth = unname(gdp.growth), gdp.growth.means = unname(gdp.growth.means),
  urban = unname(urban), urban.means = unname(urban.means),
  industry = unname(industry), industry.means = unname(industry.means),
  nC = nC,
  id.fake.countries = id.fake.countries,
  id.real.countries = id.real.countries,
  id.decade = id.decade, nDecades = nDecades,
  nB = nB, b0 = b0, B0 = B0,
  nS = nS,
  nD = nD,
  nY = nY)

nC   # Number of countries (including fake ones)

## [1] 42
```

```
nS   # Number of sectors
```

```
## [1] 2
```

```
years  # Range of years
```

```
## [1] 1976 2005
```

6.3 Model

```
M <- "diversity-vpi"
M.lab <- "Diversity (AID), robustness, VPI"
m <- "model {
  #
  # Data part at the observational level
  #
  for (d in 1:nD) {
    for (s in 1:nS) {
      for (c in 1:nC) {
        for (t in 2:nY) {
          Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
          mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
                   + beta[1,s,d] * GDPpc[c,t-1]
                   + beta[4,s,d] * vpi[s,c,t-1]
                   + beta[5,s,d] * portfolio.size[s,c,t-1]
                   + beta[6,s,d] * trade[c,t-1]
                   + beta[7,s,d] * eu[c,t-1]
                   + beta[8,s,d] * green.socialist[s,c,t-1] # col 1 green, col 2 socialist
                   + beta[9,s,d] * constraints[c,t-1]
                   + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
                   + beta[11,s,d] * interdependency.trade[d,s,c,t]
                   + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
          tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
          sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
                                + lambda[2,s,d] * portfolio.size[s,c,t]
                                + gamma[c,d])
          resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
        }
        Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
        mu[s,c,1,d] <- alpha[s,d,1]
                 + beta[1,s,d] * GDPpc[c,1]
                 + beta[4,s,d] * vpi[s,c,1]
                 + beta[5,s,d] * portfolio.size[s,c,1]
                 + beta[6,s,d] * trade[c,1]
                 + beta[7,s,d] * eu[c,1]
                 + beta[8,s,d] * green.socialist[s,c,1] # col 1 green, col 2 socialist
                 + beta[9,s,d] * constraints[c,1]
                   + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
                   + beta[11,s,d] * interdependency.trade[d,s,c,1]

        resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
        tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
        sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
                              + lambda[2,s,d] * portfolio.size[s,c,1]
                              + gamma[c,d])
      }

      #
      # Degrees of freedom of GR
      #
      nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
      nu.trans[s,d] ~ dunif(0, 1)

      #
      # Priors for variance component
      #
      lambda[1,s,d] ~ dnorm(0, 2^-2)
      lambda[2,s,d] ~ dnorm(0, 2^-2)

      #
      # Priors for the intercept
      #
      for (decade in 1:nDecades) {
```

```
    alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
  }
  Alpha[s,d] ~ dunif(0, 1)
  tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
  sigma.alpha[s,d] ~  dunif(0, 0.5)

  #
  # Priors for the control variables
  #
  beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
  Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
  Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
}
#
# Data part for  performance
#
for (p in 1:nP) {
  for (c in 1:nC) {
    for (t in 2:nY) {
      Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
      mu.performance[p,c,t,d] <- eta.performance[1,p,d]
                               + eta.performance[2,p,d] * resid[1,c,t,d]
                               + eta.performance[3,p,d] * Y[1,c,t-1,d]
                               + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
                               + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
                               + eta.performance[6,p,d] * GDPpc[c,t-1]
                               + eta.performance[7,p,d] * trade[c,t-1]
                               + eta.performance[8,p,d] * eu[c,t-1]
                               + eta.performance[9,p,d] * gdp.growth[c,t-1]
                               + eta.performance[10,p,d] * urban[c,t-1]
                               + eta.performance[11,p,d] * industry[c,t-1]
    }
    Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
    mu.performance[p,c,1,d] <- eta.performance[1,p,d]
                             + eta.performance[2,p,d] * resid[1,c,1,d]
                             + eta.performance[3,p,d] * Y[1,c,1,d]
                             + eta.performance[4,p,d] * portfolio.size[1,c,1]
                             + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
                             + eta.performance[6,p,d] * GDPpc[c,1]
                             + eta.performance[7,p,d] * trade[c,1]
                             + eta.performance[8,p,d] * eu[c,1]
                             + eta.performance[9,p,d] * gdp.growth[c,1]
                             + eta.performance[10,p,d] * urban[c,1]
                             + eta.performance[11,p,d] * industry[c,1]
  }
  tau.performance[p,d] ~ dgamma(0.001, 0.001)
  sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
  for (e in 1:11) {
    eta.performance[e,p,d] ~ dnorm(0, 1^-2)
  }
}

# Variance component, varying intercepts by country
for (c in 1:nC) {
  gamma[c,d] ~ dnorm(0, 0.2^-2)
}
#
# AR(1) parameters
#
for (c in id.real.countries) {
  for (s in 1:nS) {
    rho[s,c,d] ~ dunif(-1, 1)
  }
  for (p in 1:nP) {
    rho.performance[p,c,d] ~ dunif(-1, 1)
  }
}
for (c in id.fake.countries) {
  for (s in 1:nS) {
    rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
  }
  for (p in 1:nP) {
    rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
  }
}
```

```
  }

  # SEM Part for portfolio size
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (t in 2:nY) {
        portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
        mu.ps[s,c,t] <- alpha.ps[s,c]
                      + delta[1,s] * GDPpc[c,t-1]
                      + delta[2,s] * vpi[s,c,t-1]
                      + delta[3,s] * green.socialist[s,c,t-1]
                      + delta[4,s] * constraints[c,t-1]
      }
      tau.ps[s,c] ~ dgamma(0.1, 0.1)
      sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
      alpha.ps[s,c] ~ dnorm(0, 1^-2)
    }
    for (d in 1:4) {
      delta[d,s] ~ dnorm(0, 1^-2)
    }
  }
  # Mediated effects
  for (d in 1:nD) {
    for (s in 1:nS) {
      pi[1,s,d] <- delta[1,s] * beta[1,s,d]       # GDPpc
      pi[2,s,d] <- delta[2,s] * beta[4,s,d]       # vpi
      pi[3,s,d] <- delta[3,s] * beta[8,s,d]       # green
      pi[4,s,d] <- delta[4,s] * beta[9,s,d]       # constraints
    }
  }


  #
  # Missing data
  #
  for (c in 1:nC) {
    for (t in 1:nY) {
      gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
      urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
      industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
    }
  }
  for (s in 1:nS) {
    for (c in 1:nC) {
      # Reverse years back for NA in early years
      for (t in 1:(nY-1)) {
        green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
      }
      for (d in 1:nD) {
        for (t in 1:(nY-1)) {
          interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
        }
      }
    }
  }
  for (s in 1:nS) {
    for (c in 1:nC) {
      vpi[s,c,1] ~ dnorm(0, 0.5^-2)
      for (t in 2:nY) {
        vpi[s,c,t] ~ dnorm(vpi[s,c,1], 100)
      }
    }
  }
}"
write(m, file= paste("models/model-", M, ".bug", sep = ""))
par <- NULL
par <- c(par, "alpha", "beta", "theta", "sigma")
par <- c(par, "Alpha", "sigma.alpha")
par <- c(par, "Sigma")
par <- c(par, "lambda", "gamma")
par <- c(par, "nu")
par <- c(par, "rho")
par.fake <- expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
```

```
                         Year = 2:nY,
                         Diversity = 1:nD) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.fake)
par.resid <- expand_grid(Sector = 1:nS,
                          Country = id.real.countries,
                          Year = 2:nY,
                          Diversity = 1:nD) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par <- c(par, par.resid)
par <- c(par, "delta", "alpha.ps", "sigma.ps", "pi")
par <- c(par, "eta.performance", "sigma.performance")
inits <- list(
  list(.RNG.name="base::Super-Duper", .RNG.seed=10),
  list(.RNG.name="base::Super-Duper", .RNG.seed=20),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=30),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=20))

t0 <- proc.time()
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
  data = dump.format(DP, checkvalid=FALSE),
  inits = inits,
  modules = "glm",
  n.chains = 1, adapt = 2e2, burnin = 1e3, sample = 2e3, thin = 1,
  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 = ""))

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

```
## model {
##   #
##     #
##   for (d in 1:nD) {
##     for (s in 1:nS) {
##       for (c in 1:nC) {
##         for (t in 2:nY) {
##           Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
##           mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
##                     + beta[1,s,d] * GDPpc[c,t-1]
##                     + beta[4,s,d] * vpi[s,c,t-1]
##                     + beta[5,s,d] * portfolio.size[s,c,t-1]
##                     + beta[6,s,d] * trade[c,t-1]
##                     + beta[7,s,d] * eu[c,t-1]
##                     + beta[8,s,d] * green.socialist[s,c,t-1]              + beta[9,s,d] * constrain
##                     + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
##                     + beta[11,s,d] * interdependency.trade[d,s,c,t]
##                     + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
##           tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
##           sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
##                                   + lambda[2,s,d] * portfolio.size[s,c,t]
##                                   + gamma[c,d])
```

```
##           resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
##         }
##         Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
##         mu[s,c,1,d] <- alpha[s,d,1]
##                  + beta[1,s,d] * GDPpc[c,1]
##                  + beta[4,s,d] * vpi[s,c,1]
##                  + beta[5,s,d] * portfolio.size[s,c,1]
##                  + beta[6,s,d] * trade[c,1]
##                  + beta[7,s,d] * eu[c,1]
##                  + beta[8,s,d] * green.socialist[s,c,1]                  + beta[9,s,d] * constraints[c,1
##                    + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
##                    + beta[11,s,d] * interdependency.trade[d,s,c,1]
##
##         resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
##         tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
##         sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
##                                 + lambda[2,s,d] * portfolio.size[s,c,1]
##                                 + gamma[c,d])
##       }
##
##
##       #
##             #
##       nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
##       nu.trans[s,d] ~ dunif(0, 1)
##
##       #
##             #
##       lambda[1,s,d] ~ dnorm(0, 2^-2)
##       lambda[2,s,d] ~ dnorm(0, 2^-2)
##
##       #
##             #
##       for (decade in 1:nDecades) {
##         alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
##       }
##       Alpha[s,d] ~ dunif(0, 1)
##       tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
##       sigma.alpha[s,d] ~  dunif(0, 0.5)
##
##       #
##             #
##       beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
##       Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
##       Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
##     }
##     #
##         #
##     for (p in 1:nP) {
```

```
##       for (c in 1:nC) {
##         for (t in 2:nY) {
##           Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
##           mu.performance[p,c,t,d] <- eta.performance[1,p,d]
##                                    + eta.performance[2,p,d] * resid[1,c,t,d]
##                                    + eta.performance[3,p,d] * Y[1,c,t-1,d]
##                                    + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
##                                    + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
##                                    + eta.performance[6,p,d] * GDPpc[c,t-1]
##                                    + eta.performance[7,p,d] * trade[c,t-1]
##                                    + eta.performance[8,p,d] * eu[c,t-1]
##                                    + eta.performance[9,p,d] * gdp.growth[c,t-1]
##                                    + eta.performance[10,p,d] * urban[c,t-1]
##                                    + eta.performance[11,p,d] * industry[c,t-1]
##         }
##         Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
##         mu.performance[p,c,1,d] <- eta.performance[1,p,d]
##                                  + eta.performance[2,p,d] * resid[1,c,1,d]
##                                  + eta.performance[3,p,d] * Y[1,c,1,d]
##                                  + eta.performance[4,p,d] * portfolio.size[1,c,1]
##                                  + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
##                                  + eta.performance[6,p,d] * GDPpc[c,1]
##                                  + eta.performance[7,p,d] * trade[c,1]
##                                  + eta.performance[8,p,d] * eu[c,1]
##                                  + eta.performance[9,p,d] * gdp.growth[c,1]
##                                  + eta.performance[10,p,d] * urban[c,1]
##                                  + eta.performance[11,p,d] * industry[c,1]
##       }
##       tau.performance[p,d] ~ dgamma(0.001, 0.001)
##       sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
##       for (e in 1:11) {
##         eta.performance[e,p,d] ~ dnorm(0, 1^-2)
##       }
##     }
##
##         for (c in 1:nC) {
##       gamma[c,d] ~ dnorm(0, 0.2^-2)
##     }
##     #
##         #
##     for (c in id.real.countries) {
##       for (s in 1:nS) {
##         rho[s,c,d] ~ dunif(-1, 1)
##       }
##       for (p in 1:nP) {
##         rho.performance[p,c,d] ~ dunif(-1, 1)
##       }
##     }
##     for (c in id.fake.countries) {
```

```
##       for (s in 1:nS) {
##         rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##       }
##       for (p in 1:nP) {
##         rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##       }
##     }
##   }
##
##     for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (t in 2:nY) {
##         portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
##         mu.ps[s,c,t] <- alpha.ps[s,c]
##                        + delta[1,s] * GDPpc[c,t-1]
##                        + delta[2,s] * vpi[s,c,t-1]
##                        + delta[3,s] * green.socialist[s,c,t-1]
##                        + delta[4,s] * constraints[c,t-1]
##       }
##       tau.ps[s,c] ~ dgamma(0.1, 0.1)
##       sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
##       alpha.ps[s,c] ~ dnorm(0, 1^-2)
##     }
##     for (d in 1:4) {
##       delta[d,s] ~ dnorm(0, 1^-2)
##     }
##   }
##     for (d in 1:nD) {
##     for (s in 1:nS) {
##       pi[1,s,d] <- delta[1,s] * beta[1,s,d]             pi[2,s,d] <- delta[2,s] * beta[4,s,d]
##   }
##
##
##
##   #
##     #
##   for (c in 1:nC) {
##     for (t in 1:nY) {
##       gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
##       urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
##       industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
##     }
##   }
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##             for (t in 1:(nY-1)) {
##         green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
##       }
##       for (d in 1:nD) {
```

```
##         for (t in 1:(nY-1)) {
##           interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
##         }
##       }
##     }
##   }
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       vpi[s,c,1] ~ dnorm(0, 0.5^-2)
##       for (t in 2:nY) {
##         vpi[s,c,t] ~ dnorm(vpi[s,c,1], 100)
##       }
##     }
##   }
## }
```

```
ggmcmc(ggs(s, family = "sigma|alpha|beta|lambda|rho|nu"),
       param_page = 10, file = paste("ggmcmc-full-", M, ".pdf", sep = ""))
```

6.4 Model results

Variance components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             "Portfolio size",
                                             "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda, label = "Variable", comparison = "Sector") +
  aes(color = Sector) +
  facet_wrap(~ Diversity) +
  theme(legend.position = "right") +
  ggtitle("Variance component")
```

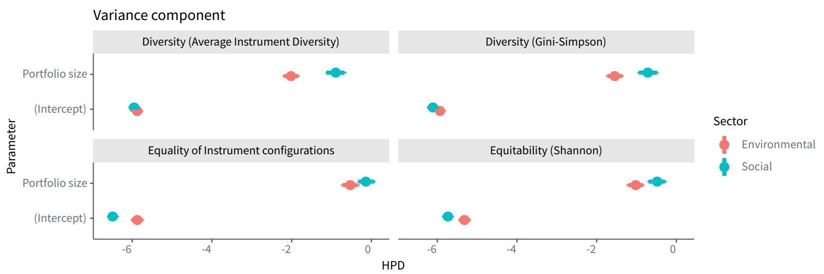

Figure 6.1: Variance component.

```
rm(S.lambda)
```

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             "Portfolio size",
                                             "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
S.lambda %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable") +
  ggtitle("Variance component")
```

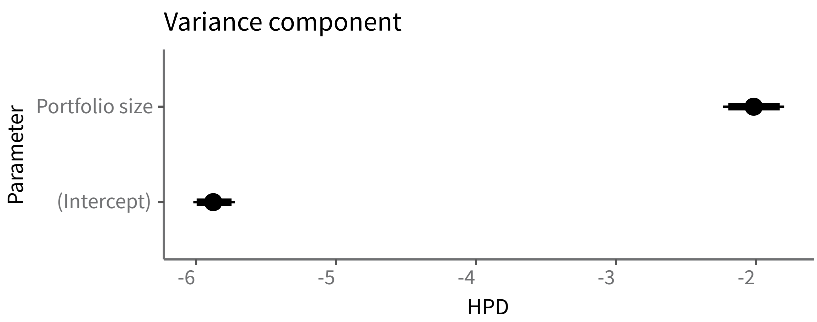

Figure 6.2: Variance component.

```
rm(S.lambda)
```

Variance components (country varying intercepts).

```
L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.gamma, label = "Country") +
  facet_grid(~ Diversity) +
  ggtitle("Variance component (countries)")
```

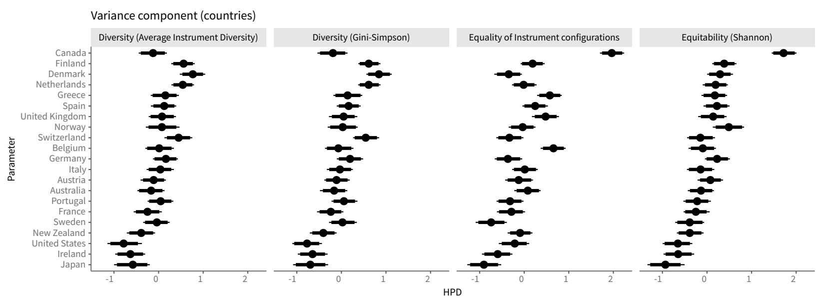

Figure 6.3: Variance component (country varying intercepts).

```
rm(S.gamma)
```

```
L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-")) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)")
ggs_caterpillar(S.gamma, label = "Country") +
  ggtitle("Variance component (countries)")
```

```
rm(S.gamma)
```

Auto-regressive components.

```
L.rho <- plab("rho", list(Sector = sector.label,
                          Country = country.label,
                          Diversity = diversity.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  facet_grid(Diversity ~ Sector, scales="free") +
```

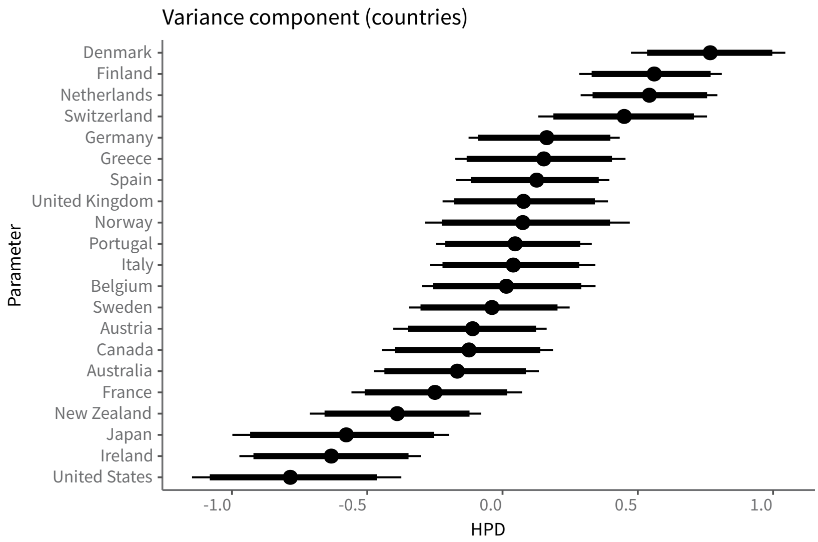

Figure 6.4: Variance component (country varying intercepts).

```
  aes(color=Sector) +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()
```

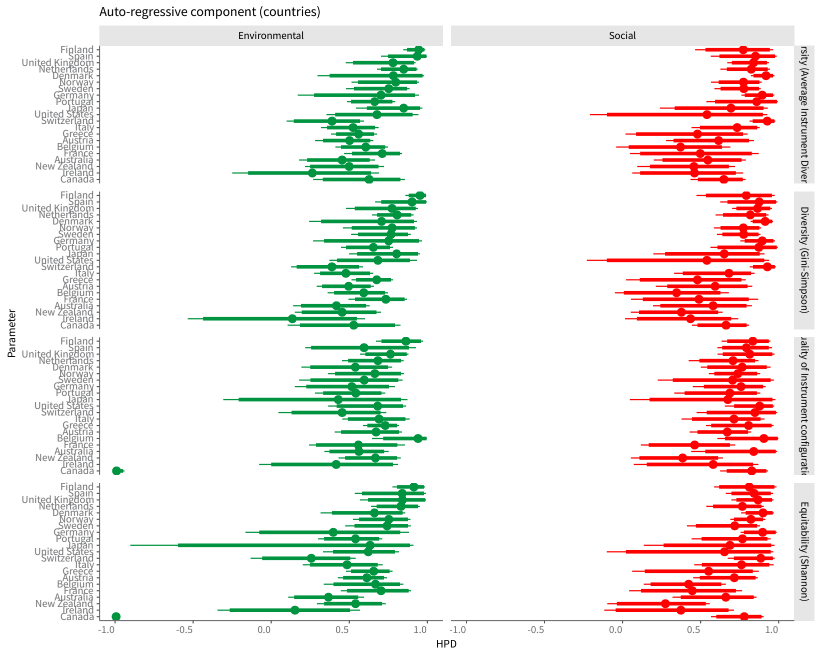

Figure 6.5: Auto-regressive component.

```
rm(S.rho)

L.rho <- plab("rho", list(Sector = sector.label,
                          Country = country.label,
                          Diversity = diversity.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
```

```
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()
```

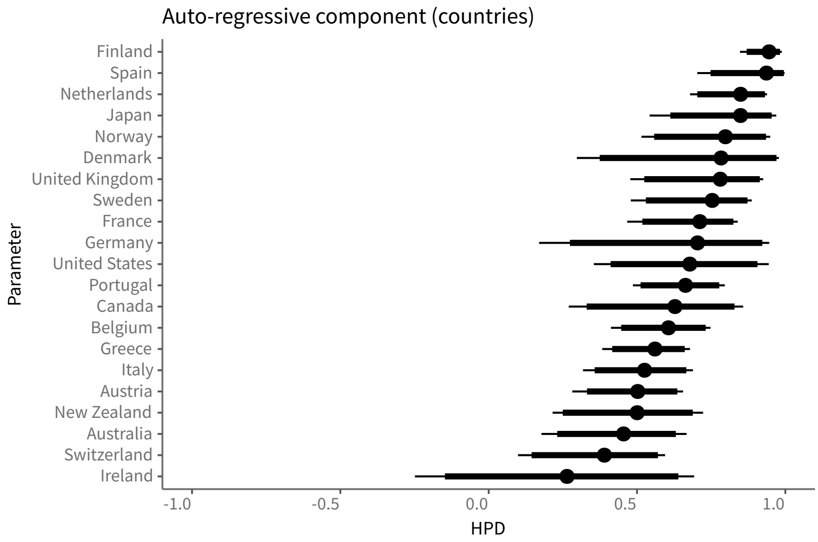

Figure 6.6: Auto-regressive component.

```
rm(S.rho)
```

Intercepts

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Diversity = diversity.label,
                              Decade = decade.label))

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

ci(S.alpha) %>%
  ggplot(aes(x = Decade, y = median, ymin = Low, ymax = High)) +
  geom_point() +
  geom_linerange() +
  facet_grid(Diversity ~ Sector) +
  ggtitle("Temporal trend")

covariates.label <- c("GDP pc",
                                        "Consensus",
                                        "Deliberation",
                                        "Policy feedback",
                                        "Portfolio size",
                                        "Trade", "EU", "Salience",
                                        "Political constraints",
                                        "Interdependency (Contiguity)",
                                        "Interdependency (Trade)")

L.betas <- plab("beta", list(Variable = covariates.label,
                             Sector = sector.label,
                       Diversity = diversity.label))

S.betas <- ggs(s, family = "^beta\\[", par_labels = L.betas) %>%
  filter(!Variable %in% c("Deliberation", "Consensus"))
ci.betas <- ci(S.betas)
save(ci.betas, file = paste0("ci-betas-", M, ".RData"))

S.betas %>%
  group_by(Variable, Sector) %>%
  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()
```

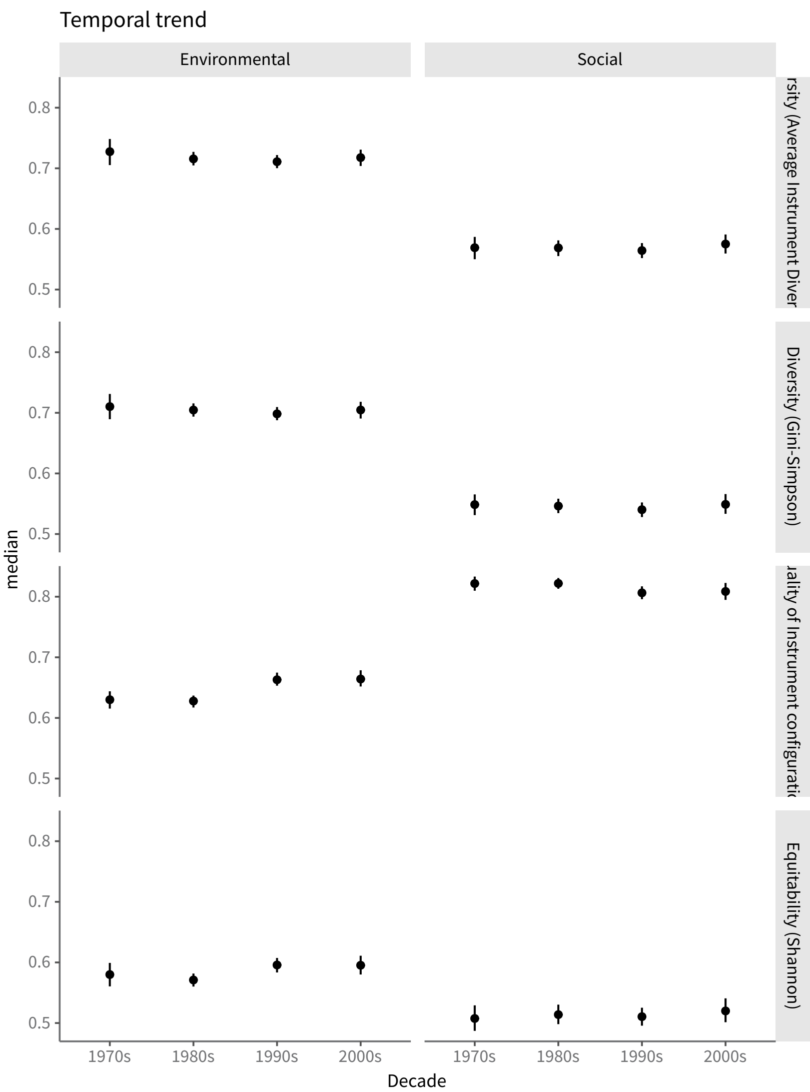

Figure 6.7: Intercepts.

Variable	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	-0.03	0.02	0.00	1.00	-0.04
GDP pc	Social	-0.03	0.03	0.26	0.74	-0.02
Policy feedback	Environmental	0.04	0.01	1.00	0.00	0.04
Policy feedback	Social	-0.04	0.03	0.25	0.75	-0.03
Portfolio size	Environmental	0.14	0.02	1.00	0.00	0.15
Portfolio size	Social	0.10	0.02	1.00	0.00	0.09
Trade	Environmental	-0.01	0.01	0.06	0.94	-0.01
Trade	Social	-0.02	0.02	0.02	0.98	-0.03
EU	Environmental	-0.02	0.01	0.01	0.99	-0.02
EU	Social	-0.01	0.02	0.43	0.57	-0.01
Salience	Environmental	0.00	0.01	0.34	0.66	0.00
Salience	Social	0.00	0.01	0.48	0.52	0.00
Political constraints	Environmental	-0.04	0.02	0.01	0.99	-0.04
Political constraints	Social	-0.03	0.02	0.00	1.00	-0.04
Interdependency (Contiguity)	Environmental	0.10	0.03	1.00	0.00	0.09
Interdependency (Contiguity)	Social	0.04	0.04	0.73	0.27	0.03
Interdependency (Trade)	Environmental	0.04	0.03	0.75	0.25	0.02
Interdependency (Trade)	Social	0.00	0.02	0.43	0.57	0.00

```
ci(S.betas) %>%
ggplot(aes(x = Variable,
               y = median,
               group = interaction(Sector, Diversity),
               color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

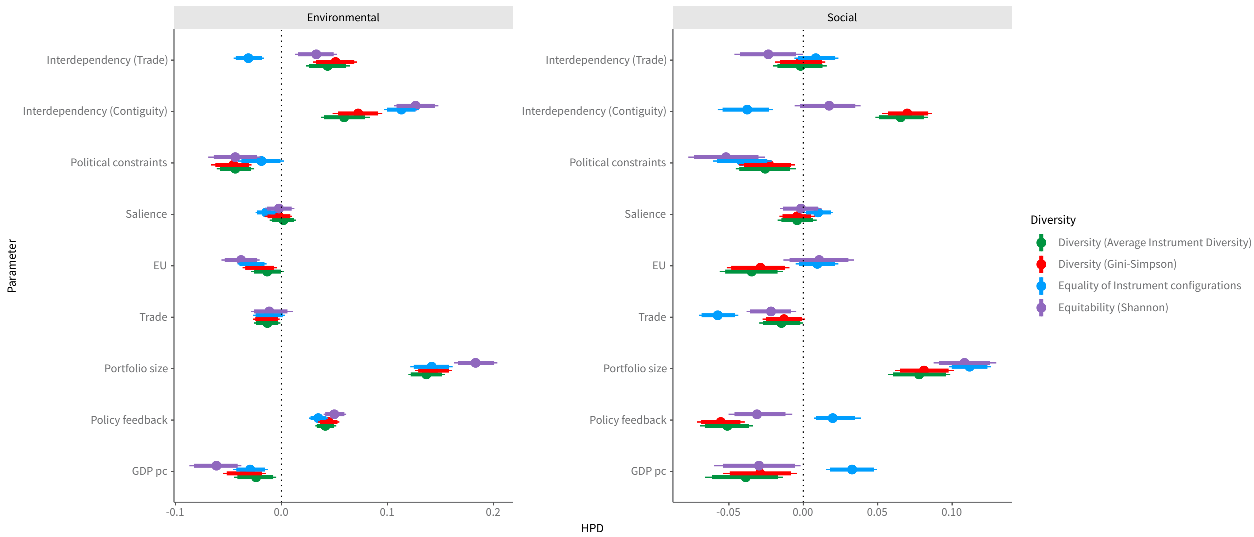

Figure 6.8: Slopes with the effects on diversity, by sector.

```
ci(S.betas) %>%
  filter(Sector == "Environmental") %>%
ggplot(aes(x = Variable,
               y = median,
               group = Diversity,
               color = Diversity)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

```
ci(S.betas) %>%
ggplot(aes(x = Variable,
               y = median,
               group = Sector, color = Sector)) +
  coord_flip() +
  facet_wrap(~ Diversity, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

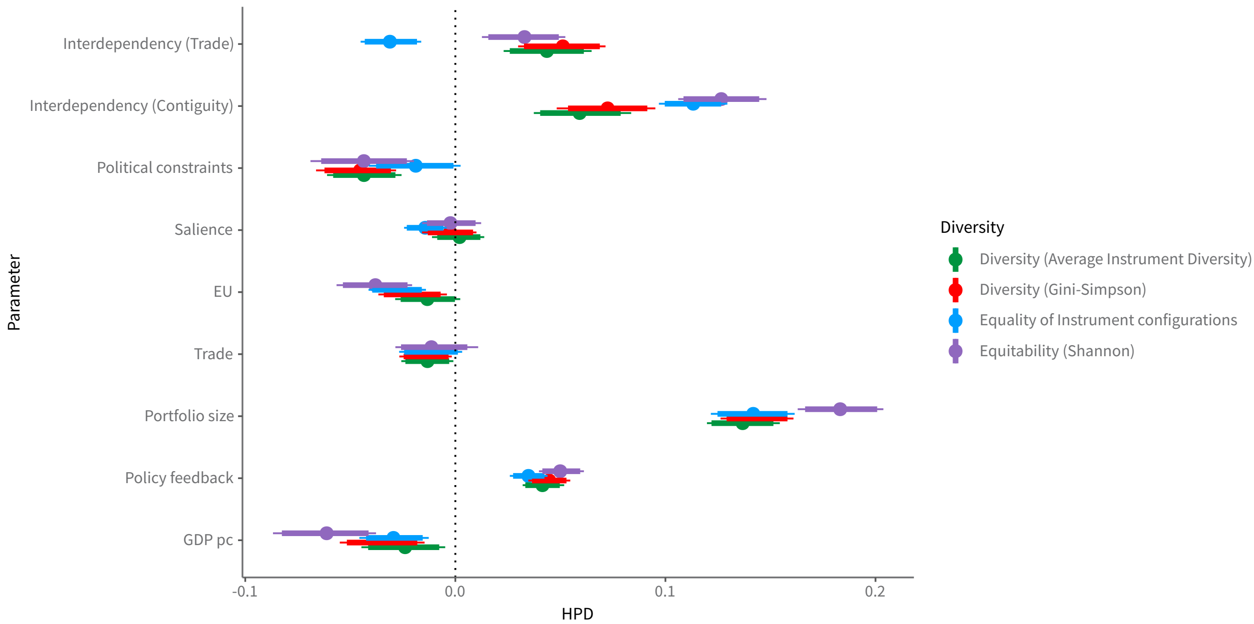

Figure 6.9: Slopes with the effects on diversity, for the environmental sector.

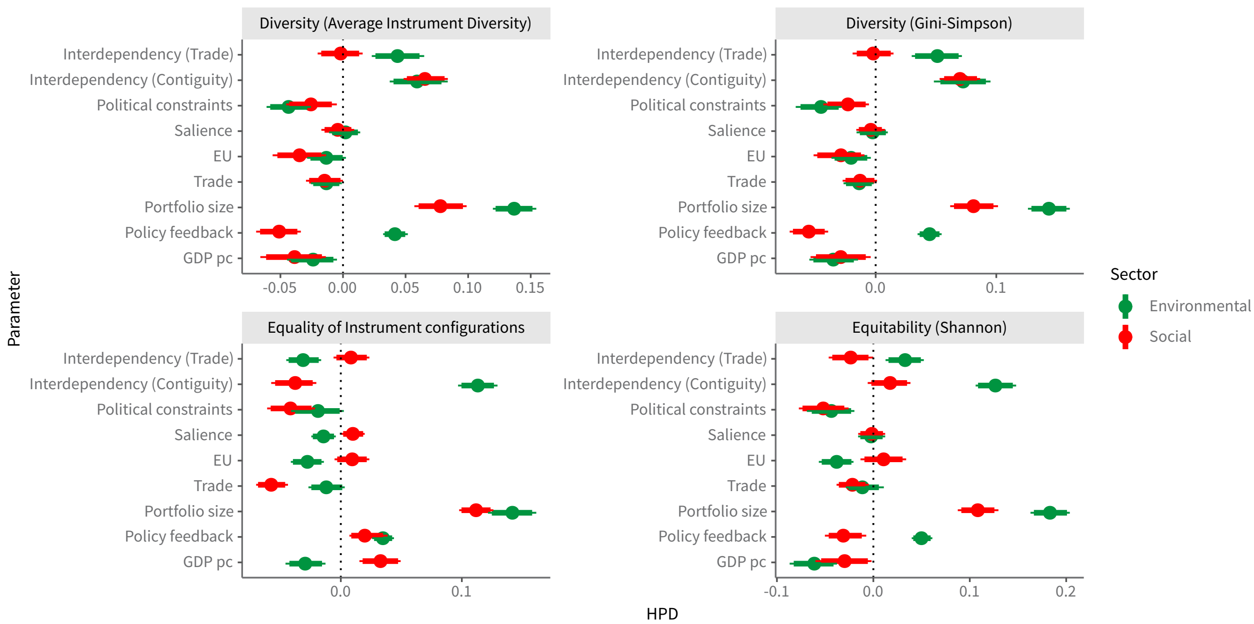

Figure 6.10: Slopes with the effects on diversity, by diversity indicator.

```
ci(S.betas) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggplot(aes(x = Variable,
             y = median,
             group = Sector, color = Sector)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

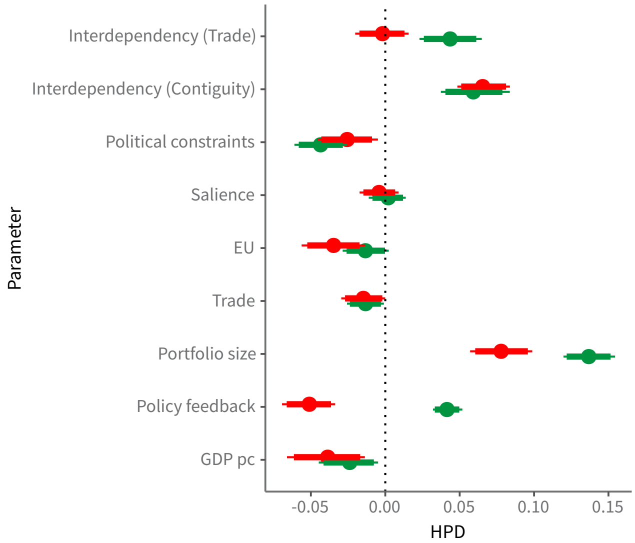

Figure 6.11: Slopes with the effects on AID diversity, by sector

```
ggplot(S.betas, aes(x = value, color = Sector, fill = Sector)) +
  geom_density(alpha = 0.5) +
  facet_grid(Diversity ~ Variable, scales = "free") +
  xlab("HPD") +
  geom_vline(xintercept = 0, lty = 3)
```

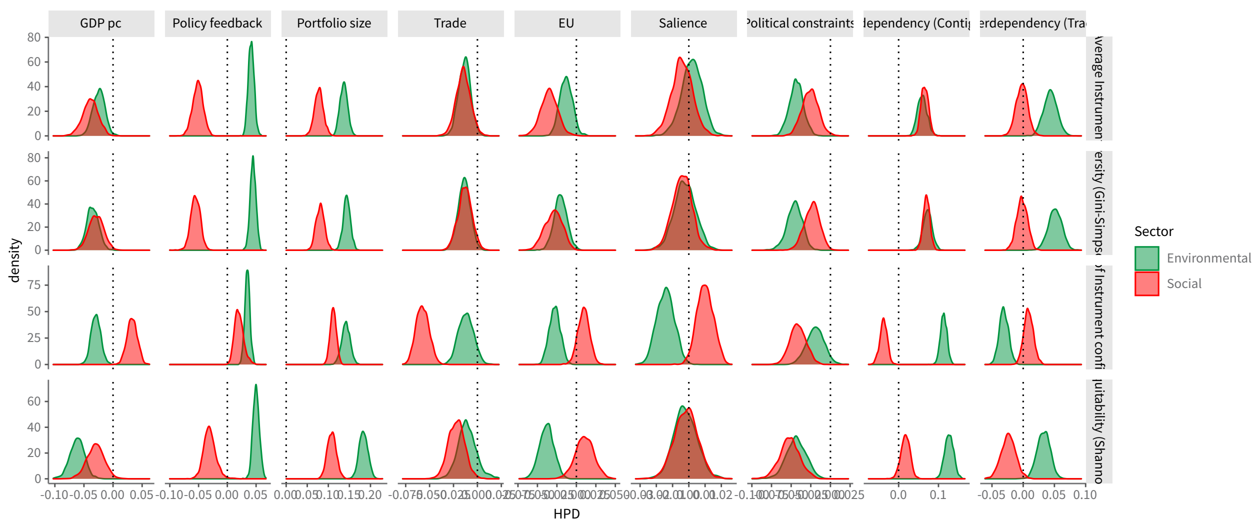

Figure 6.12: Slopes with the effects on diversity, comparing sectors.

Variables by evidence.

```
S.betas %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(value != 0) %>%
  group_by(Sector, Diversity, Variable) %>%
  summarize(`Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
    `Mean expected effect` = mean(value)) %>%
  group_by(Sector, Diversity, Variable) %>%
  mutate(max = max(abs(`Prob > 0`), abs(`Prob < 0`))) %>%
  arrange(desc(max)) %>%
  select(-max, -Diversity) %>%
  kable()
```

Diversity	Sector	Variable	Prob > 0	Prob < 0	Mean expected effect
Diversity (Average Instrument Diversity)	Environmental	Policy feedback	1.00	0.00	0.04
Diversity (Average Instrument Diversity)	Environmental	Portfolio size	1.00	0.00	0.14
Diversity (Average Instrument Diversity)	Environmental	Political constraints	0.00	1.00	-0.04
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Contiguity)	1.00	0.00	0.06
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Trade)	1.00	0.00	0.04
Diversity (Average Instrument Diversity)	Social	Policy feedback	0.00	1.00	-0.05
Diversity (Average Instrument Diversity)	Social	Portfolio size	1.00	0.00	0.08
Diversity (Average Instrument Diversity)	Social	Interdependency (Contiguity)	1.00	0.00	0.07
Diversity (Average Instrument Diversity)	Social	GDP pc	0.00	1.00	-0.04
Diversity (Average Instrument Diversity)	Social	EU	0.00	1.00	-0.03
Diversity (Average Instrument Diversity)	Social	Political constraints	0.01	0.99	-0.03
Diversity (Average Instrument Diversity)	Environmental	GDP pc	0.01	0.99	-0.02
Diversity (Average Instrument Diversity)	Environmental	Trade	0.02	0.98	-0.01
Diversity (Average Instrument Diversity)	Social	Trade	0.02	0.98	-0.01
Diversity (Average Instrument Diversity)	Environmental	EU	0.05	0.95	-0.01
Diversity (Average Instrument Diversity)	Social	Salience	0.27	0.73	0.00
Diversity (Average Instrument Diversity)	Environmental	Salience	0.62	0.38	0.00
Diversity (Average Instrument Diversity)	Social	Interdependency (Trade)	0.42	0.58	0.00

```
S.betas.env <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) #%>%

ggs_caterpillar(S.betas.env) +
  geom_vline(xintercept = 0, lty = 3)
```

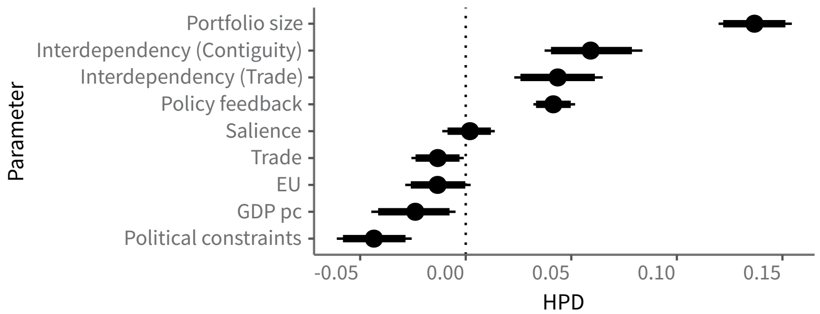

Figure 6.13: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env)

S.betas.env.sorted <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) %>%
  mutate(Parameter = fct_relevel(Parameter, rev(c("Political constraints",
                                                   "Policy feedback",
                                  "Salience",
```

```
                                                "GDP pc", "Trade", "EU",
                                                "Interdependency (Contiguity)",
                                                "Interdependency (Trade)",
                                                "Portfolio size"))))
ggs_caterpillar(S.betas.env.sorted, sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)
```

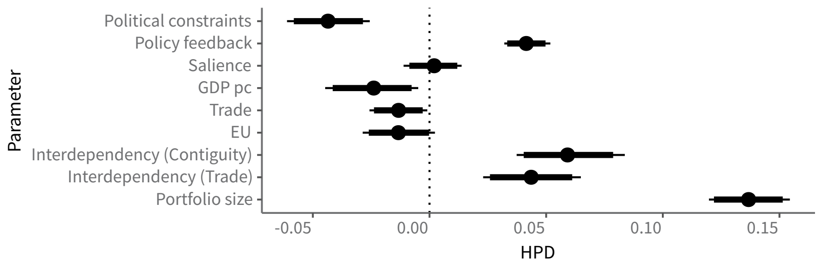

Figure 6.14: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env.sorted)

ci.betas <- ci(S.betas) %>%
  mutate(Model = M.lab)

save(ci.betas, file = paste0("ci_betas-", M, ".RData"))
rm(S.betas, ci.betas)

L.deltas <- plab("delta", list(Variable = c("GDP pc",
                                            "Policy feedback",
                                            "Salience",
                                            "Political constraints"),
                        Sector = sector.label))

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

S.deltas %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	1.43	0.07	1.00	0.00	1.42
GDP pc	Social	0.43	0.04	1.00	0.00	0.43
Policy feedback	Environmental	-0.33	0.22	0.06	0.94	-0.33
Policy feedback	Social	0.13	0.12	0.96	0.04	0.15
Salience	Environmental	0.13	0.03	1.00	0.00	0.13
Salience	Social	-0.04	0.02	0.00	1.00	-0.04
Political constraints	Environmental	0.14	0.06	0.99	0.01	0.14
Political constraints	Social	0.20	0.06	1.00	0.00	0.20

```
ci(S.deltas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = Sector)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

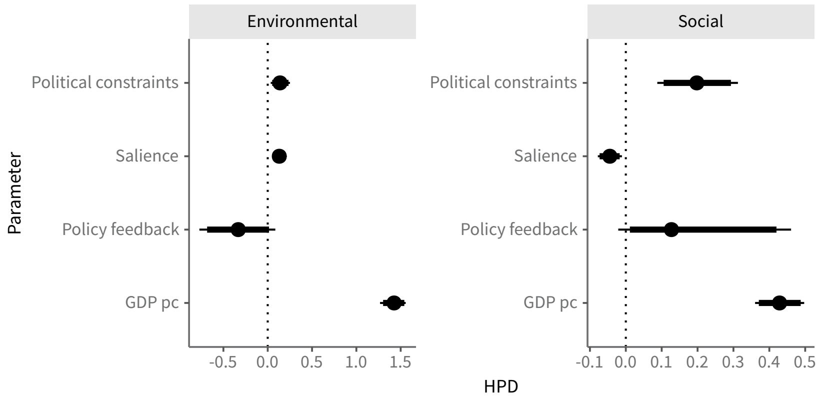

Figure 6.15: Slopes with the direct effects on portfolio size, by sector.

```
rm(S.deltas)

L.pi ← plab("pi", list(Variable = c("GDP pc",
                                     "Policy feedback",
                                     "Salience",
                                     "Political constraints"),
                       Sector = sector.label,
                       Diversity = diversity.label))

S.pis ← ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable ≠ "GDP pc")

S.pis %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Policy feedback	Environmental	-0.01	0.01	0.06	0.94	-0.01
Policy feedback	Social	0.00	0.01	0.27	0.73	0.00
Salience	Environmental	0.00	0.00	0.34	0.66	0.00
Salience	Social	0.00	0.00	0.52	0.48	0.00
Political constraints	Environmental	-0.01	0.00	0.02	0.98	-0.01
Political constraints	Social	-0.01	0.00	0.00	1.00	-0.01

```
ci(S.pis) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()

L.pi ← plab("pi", list(Variable = c("GDP pc",
                                     "Policy feedback",
                                     "Salience",
                                     "Political constraints"),
```

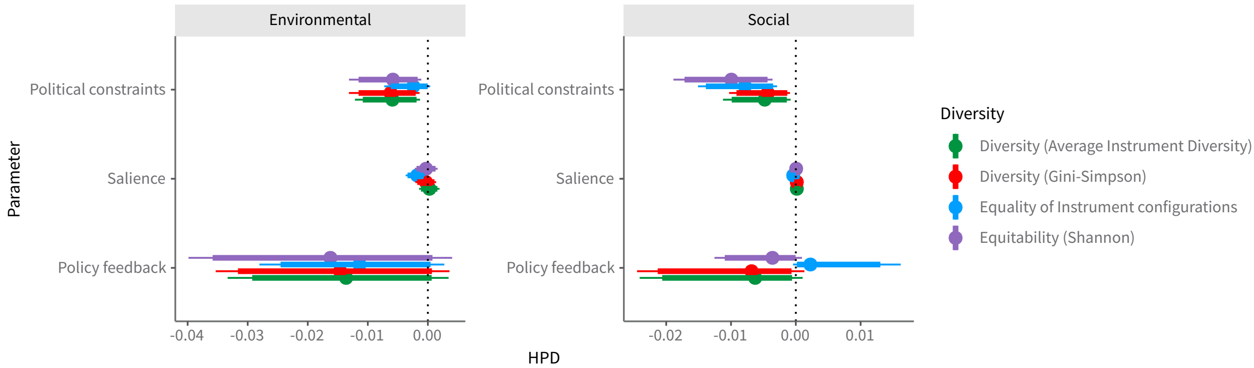

Figure 6.16: Mediated effects on diversity, by sector.

```
                          Sector = sector.label,
                          Diversity = diversity.label))
S.pis <- ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable != "GDP pc")
```

```
S.pis %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Policy feedback	Environmental	-0.01	0.01	0.06	0.94	-0.01
Policy feedback	Social	0.00	0.01	0.27	0.73	0.00
Salience	Environmental	0.00	0.00	0.34	0.66	0.00
Salience	Social	0.00	0.00	0.52	0.48	0.00
Political constraints	Environmental	-0.01	0.00	0.02	0.98	-0.01
Political constraints	Social	-0.01	0.00	0.00	1.00	-0.01

```
S.pis %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable")
```

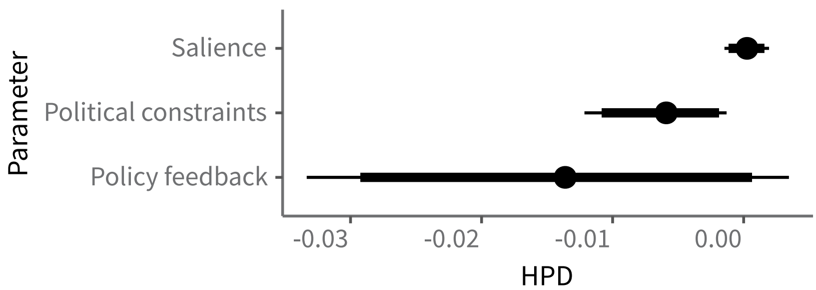

Figure 6.17: Mediated effects on diversity, for AID and the environmental sector.

```
rm(S.pis)
```

```
L.eta <- plab("eta.performance", list(Covariate = c("(Intercept)", "Residuals",
                                                     "Diversity", "Portfolio size",
                                                     "Diversity * Portfolio size",
                                                     "GDPpc", "Trade", "EU",
```

```
                                        "GDP growth", "Urban", "Industry"),
                                Performance = performance.label,
                                Diversity = diversity.label))
S.eta <- ggs(s, family = "eta.performance", par_label = L.eta)
```

```
ggs_caterpillar(S.eta, label = "Covariate", comparison = "Performance") +
  geom_hline(yintercept = 0, lty = 3) +
  aes(color = Performance) +
  theme(legend.position = "right") +
  facet_wrap(~ Diversity)
```

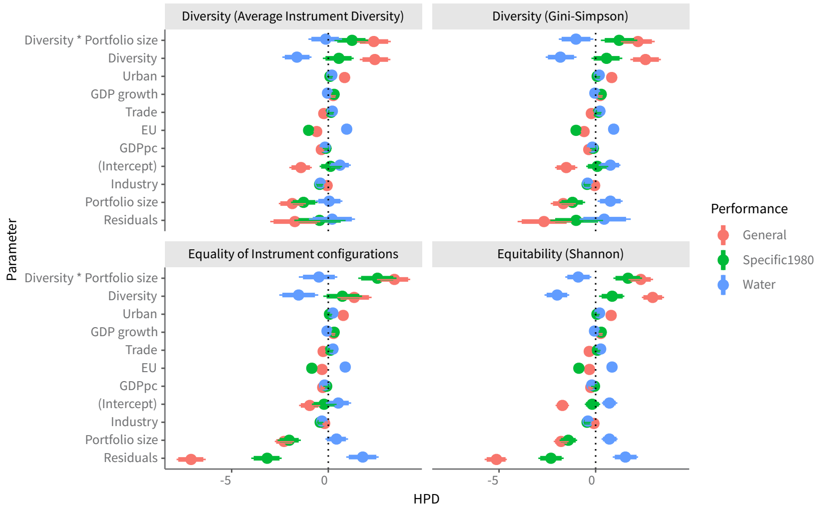

Figure 6.18: Direct effects on Environmental performance, by performance indicator and Diversity measure.

```
S.eta %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Performance == "General") %>%
ggs_caterpillar(label = "Covariate") +
  geom_vline(xintercept = 0, lty = 3)
```

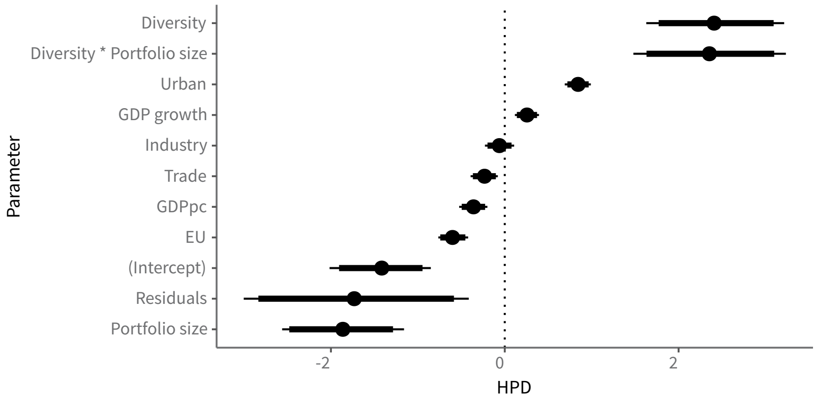

Figure 6.19: Direct effects on Environmental performance, for general performance and using AID as diversity.

```
rm(S.eta)
```

6.5 *V-Cov*

```
L.Sigma <- plab("Sigma", list(Covariate.1 = covariates.label,
                              Covariate.2 = covariates.label,
                              Sector = sector.label,
```

```
                              Diversity = diversity.label))
S.Sigma <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma) %>%
  filter(!Covariate.1 %in% c("Deliberation", "Consensus")) %>%
  filter(!Covariate.2 %in% c("Deliberation", "Consensus"))

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

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

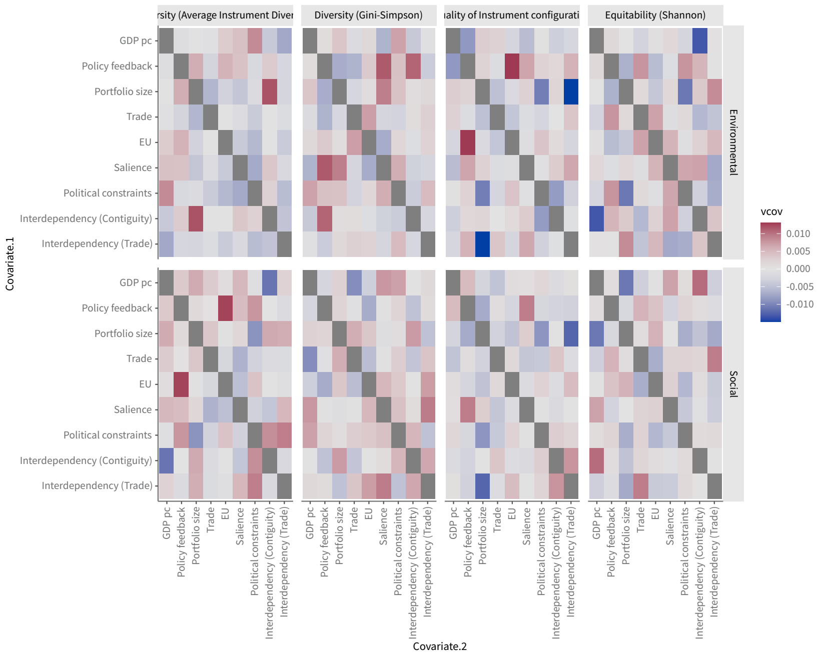

Figure 6.20: Variance-covariance matrix of main effects.

```
rm(S.Sigma, vcov.Sigma)
```

6.6 *Model evaluation*

What is the model fit?

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

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1, Country = Var2,
         Year = Var3, Diversity = Var4,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector, Diversity) %>%
  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, Diversity) %>%
  summarize(rsd = sd(value))

ggplot(S.rsd, aes(x = rsd)) +
#  geom_histogram(binwidth = 0.001) +
  geom_histogram(binwidth = 0.001) +
  geom_vline(data = Obs.sd, aes(xintercept = obs.sd)) +
  facet_grid(Diversity ~ Sector) +
  expand_limits(x = 0)
```

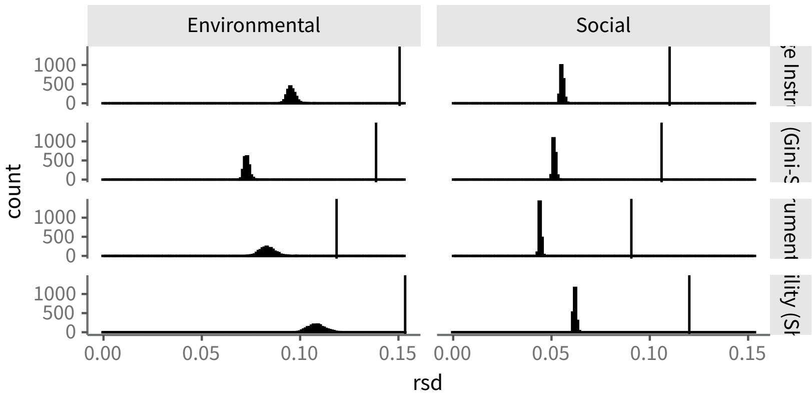

Figure 6.21: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

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

Sector	Diversity	Pseudo.R2
Environmental	Diversity (Average Instrument Diversity)	0.37
Environmental	Diversity (Gini-Simpson)	0.47
Environmental	Equality of Instrument configurations	0.30
Environmental	Equitability (Shannon)	0.29
Social	Diversity (Average Instrument Diversity)	0.50
Social	Diversity (Gini-Simpson)	0.51
Social	Equality of Instrument configurations	0.51
Social	Equitability (Shannon)	0.48

```
rm(S.rsd)
```

Which are the observations with higher residuals, further away from the expectation?[1]

[1] Negative residuals are cases where the country has lower diversity than expected according to the model.

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

ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(!str_detect(Country, "Z-")) %>%
  group_by(Country) %>%
  summarize(Residual = mean(value)) %>%
  mutate(`Mean absolute residual` = abs(Residual)) %>%
  ungroup() %>%
  arrange(desc(`Mean absolute residual`)) %>%
  select(-`Mean absolute residual`) %>%
  slice(1:8) %>%
  kable()
```

Country	Residual
Greece	0.09
Finland	-0.09
Netherlands	-0.07
Norway	0.06
Denmark	0.06
New Zealand	0.05
Australia	0.05
France	0.04

6.7 *Magnitude of effects*

```
L.mu <- plab("mu", list(Sector = sector.label,
                        Country = country.label,
                        Year = year.label.numeric,
                        Diversity = diversity.label)) %>%
  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()

ci.mu %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  ggplot(aes(x = Year, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen portfolio size goes\nfrom minimum to maximum\nover time")
```

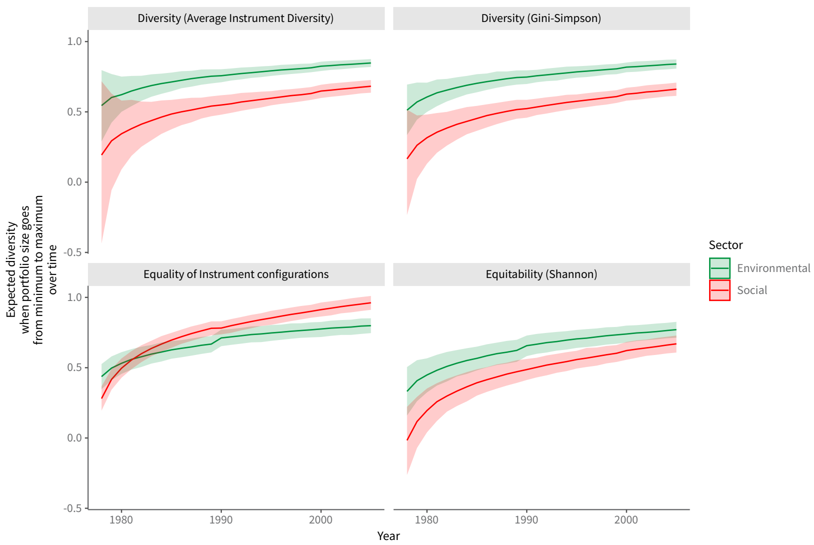

Figure 6.22: Magnitude of the effects: change in the expected diversity when portfolio size goes from the minimum to the maximum observed value over time, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year == max(Year)) %>%
  left_join(vpi.df) %>%
  rename(`Policy feedback` = value) %>%
  ggplot(aes(x = `Policy feedback`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
```

```
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen policy feedback goes\nfrom minimum to maximum")
```

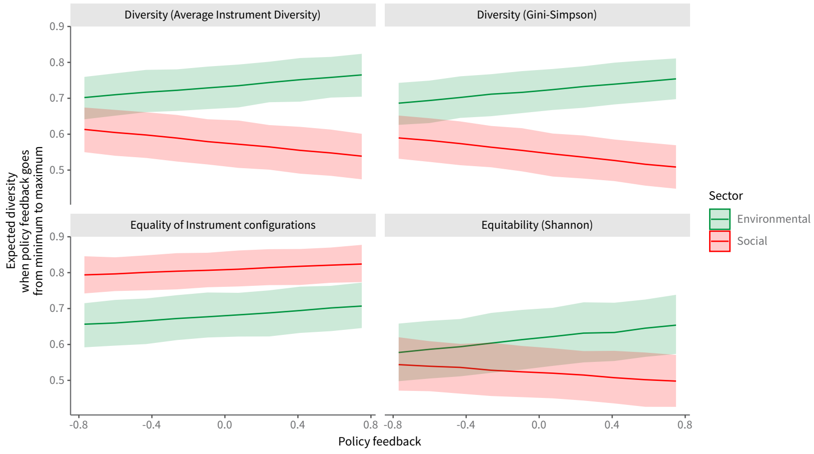

Figure 6.23: Magnitude of the effects: change in the expected diversity when government effectiveness goes from the minimum to the maximum observed value, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year == max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = polcon) %>%
  ggplot(aes(x = `Political constraints`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen political constraints go\nfrom minimum to maximum")
```

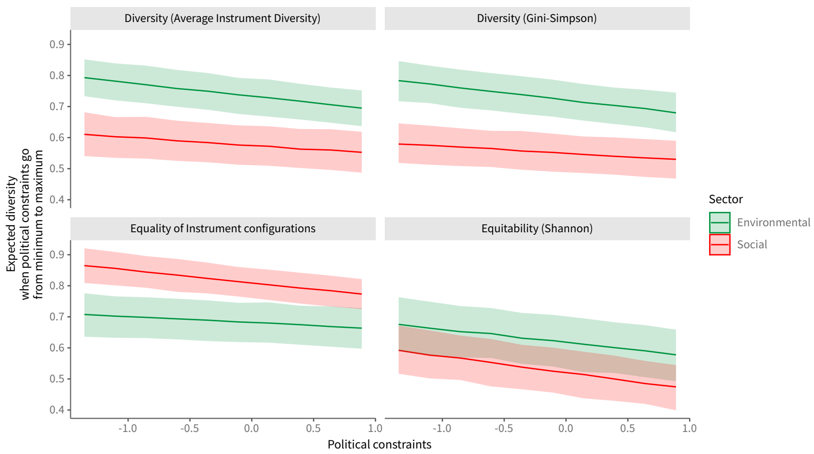

Figure 6.24: Magnitude of the effects: change in the expected diversity when political constraints go from the minimum to the maximum observed value, with the rest of the variables at their means.

```
f1 <-
  ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  left_join(D %>%
            filter(Measure == "Size") %>%
            select(Country, Sector, Year, value) %>%
            rename(`Portfolio size` = value)) %>%
```

```
  ggplot(aes(x = `Portfolio size`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(a)") +
  ylab("Expected diversity")

f2 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year ==  max(Year)) %>%
  left_join(vpi.df) %>%
  rename(`Policy feedback` = value) %>%
  ggplot(aes(x = `Policy feedback`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(b)") +
  ylab("Expected diversity")

f3 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year ==  max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = original.polcon) %>%
  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, 1)) +
  ggtitle("(c)") +
  ylab("Expected diversity")

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

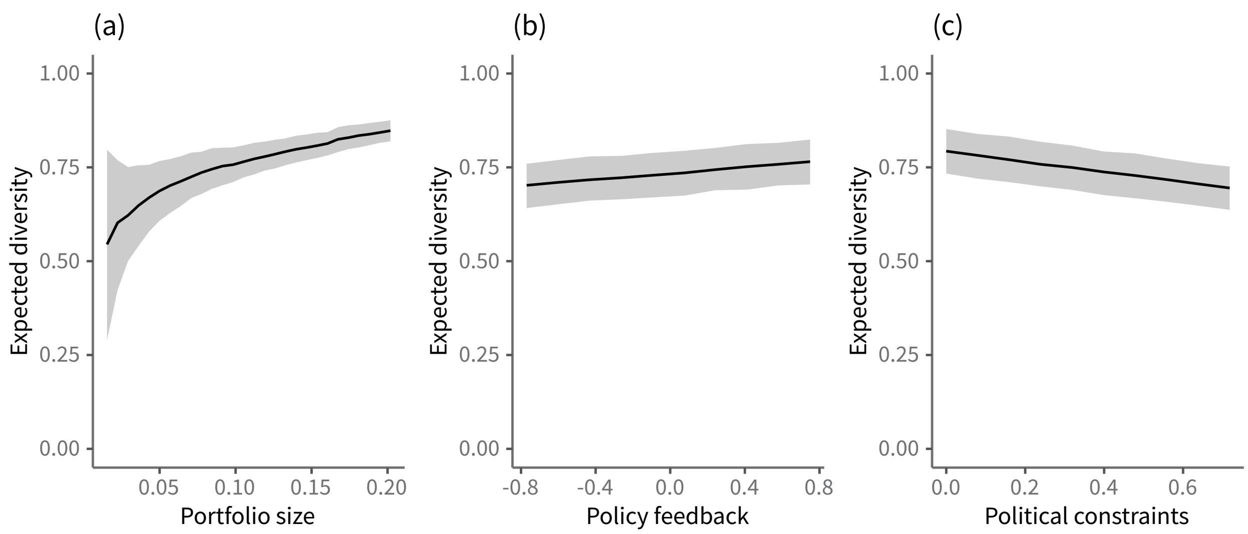

Figure 6.25: Magnitude of the effects: change in the expected diversity when (a) portfolio size, (b) government effectiveness or (c) political constraints move from the minimum to the maximum observed. In all cases the rest of the variables are fixed at their means.

7

Explanatory model of instrument diversity (robustness, market-based instrumemnts)

Data is TSCS, by sector.

```
library(PolicyPortfolios)
data(consensus)

c.eu <- c("Austria", "Belgium", "Denmark", "Finland",
          "France", "Germany", "Greece", "Ireland",
          "Italy", "Netherlands", "Portugal", "Spain",
          "Sweden", "United Kingdom")

# Target 13 (CO2 for combustion plants) + New Instrument; market-based
# All EU countries in the sample; only 2005
extra <- consensus %>%
  filter(Instrument == "Liability scheme") %>%
  mutate(Instrument = "New market-based") %>%
  mutate(covered = ifelse(Target == "Carbon dioxide (CO2) emissions from large combustion plants of the sma
                          Year == 2005 &
                          Country %in% c.eu,
                        1, 0)) %>%
# Target 10 (SO2 for combustion plants) + New Instrument; market-based
# US; for 1995 onwards
  mutate(covered = ifelse(Target == "Sulphur dioxide (SO2) emissions from large combustion plants of the sm
                          Year >= 1995 &
                          Country == "United States",
                        1, covered))

consensus <- consensus %>%
  bind_rows(extra) %>%
  mutate(Instrument = as.factor(Instrument))

D <- bind_rows(
  # Calculate portfolio measures sector by sector
  consensus %>%
    filter(Sector == "Environmental") %>%
    droplevels() %>%
    pp_measures(),
  consensus %>%
    filter(Sector == "Social") %>%
    droplevels() %>%
    pp_measures())


# Add more fake countries

# Z-01 Increases portfolio size from minimum to maximum observed size
#      over time
range.size.env <- range(filter(D, Sector == "Environmental" & Measure == "Size")$value)
range.size.soc <- range(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Environmental", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.env[1], range.size.env[2],
                                       length.out = length(min(D$Year):max(D$Year)))))
```

```
D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Social", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.soc[1], range.size.soc[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

# Z-02, Z-11 Fixes government effectiveness to its mean
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))

# Z-12:Z-21 Fixes portfolio size for veto players to move
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))
```

```
diversity <- D %>%
  mutate(Country = as.factor(Country)) %>%
  select(-Measure) %>%
  rename(Measure = Measure.label) %>%
  spread(Measure, value) %>%
  mutate(`Diversity/W (Average Instrument Diversity)` =
         `Diversity (Average Instrument Diversity)` / (1 - `Proportion of targets covered`)) %>%
  select(Sector, Country, Year,
    `Diversity (Average Instrument Diversity)`,
    `Diversity (Gini-Simpson)`,
    `Equitability (Shannon)`,
    `Equality of Instrument configurations`)

countries <- as.character(unique(D$Country))
nC <- length(countries)
years <- range(D$Year)
```

7.1 Performance

```
perf <- foreign::read.dta("jahn/PoEP_Replication_Data/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% 1980:2010)

d.perf ← perf %>%
  # Delete Years for which we don't have data
  filter(Year ⩾ 1980 & Year ⩽ 2005) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Water", "Specific1980")) %>%
  droplevels()

Y.performance ← reshape2::acast(d.perf, Indicator ~ Country ~ Year, value.var = "Performance")
nYperformance ← dim(Y.performance)[3]
```

7.2 Covariates

World Development Indicators - Revenue.

```
load("wdi/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("wdi/wdi.RData")
wdi ← wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi ← subset(wdi, year ⩾ 1976 & year ⩽2005)
wdi$country[wdi$country=="Korea, Rep."] ← "South Korea"
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(pred

#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(

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(pred

# 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$`2005`/wdi.gcr.w$`1976`)

# 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)
```

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("wgi/wgi-full-v201029.RData")
gov.eff.original <- wgi %>%
  filter(indicator == "Government effectiveness") %>%
  select(country, year, value = Estimate) %>%
  expand_grid(Sector = c("Environmental", "Social"))
```

Add fake government effectiveness

```
range.ge.env <- range.ge.soc <- range(gov.eff.original$value, na.rm = TRUE)

gov.eff.original <- gov.eff.original %>%
  bind_rows(expand_grid(tibble(Sector = "Environmental",
                               country = paste0("Z-", sprintf("%02d", 2:11)),
                               value = seq(range.ge.env[1],
                                           range.ge.env[2],
                                           length.out = length(2:11))),
                        year = min(D$Year):max(D$Year)))

gov.eff.original <- gov.eff.original %>%
  bind_rows(expand_grid(tibble(Sector = "Social",
                               country = paste0("Z-", sprintf("%02d", 2:11)),
                               value = seq(range.ge.soc[1],
                                           range.ge.soc[2],
                                           length.out = length(2:11))),
                        year = min(D$Year):max(D$Year)))
```

Political Constraints - polconIII. An indicator of "veto players" comes from Henisz (2002). The indicator "estimates the feasibility of policy change. [That is], the extent to which a change in the preferences of any one actor may lead to a change in government policy". Higher values represent systems with higher constraints.

```
load("polcon/polcon2017.RData") # loads polcon
```

Add fake political constraints values.

```
polcon.d <- polcon %>%
  as_tibble() %>%
  filter(year %in% years[1]:years[2]) %>%
  mutate(country.polity = as.character(country.polity)) %>%
  mutate(country.polity = ifelse(country.polity == "Germany West", "Germany", country.polity)) %>%
  filter(country.polity %in% countries) %>%
  select(Country = country.polity, Year = year, polcon)

range.polcon <- polcon.d %>%
  select(polcon) %>%
  unlist(., use.names = FALSE) %>%
  range()

polcon.d <- polcon.d %>%
  bind_rows(expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                               polcon = seq(range.polcon[1],
                                            range.polcon[2],
                                            length.out = length(12:21))),
                        Year = min(D$Year):max(D$Year)))
```

For the "Green parties", data comes from Volkens (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("manifesto/cpm-consensus.RData")
cpm$country <- as.character(cpm$country)
cpm$country[cpm$country=="Korea"] <- "South Korea"
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<="2005-12-31")

# 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, as.numeric(diff(c(as.Date("1976-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}
```

For the salience of each topic, we employ data from Volkens (2013), weighting the proportion of votes to each party and the importance that each party gives to environmental issues or the expansion of social welfare.

```
load("manifesto/201029-cpm-salience.RData")

salience <- cpm.salience %>%
  filter(Country %in% countries) %>%
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  filter(Year %in% years[1]:years[2])
```

Border contiguity

```
load("borders/geography.RData")

m.borders <- M.borders[dimnames(M.borders)[[1]] %in% countries,
                       dimnames(M.borders)[[2]] %in% countries]
```

Trade dependency

```
load("trade/trade.RData")

rm(M.trade, M.trade.imports)
```

Save and arrange for analysis.

```
diversity <- diversity %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  droplevels()

diversity.l <- diversity %>%
  gather(Measure, value, -c(Sector, Country, Year))
Y <- reshape2::acast(diversity.l,
                     Sector ~ Country ~ Year ~ Measure, value.var = "value")
nS <- dim(Y)[1]
nC <- dim(Y)[2]
nY <- dim(Y)[3]
nD <- dim(Y)[4]


country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
nC.fake <- 21
nC.real <- nC - nC.fake
id.real.countries <- 1:nC.real
id.fake.countries <- (nC.real + 1):nC

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)

year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(as.numeric(year.label))
nY <- length(year.label)

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)

diversity.label <- dimnames(Y)[[4]]
nD <- length(diversity.label)

nB <- 11
b0 <- rep(0, nB)
B0 <- diag(nB)
diag(B0) <- 1^-2


# 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("get-eu_time.R") # generates eu.ms

# Fake countries
f.c <- paste0("Z-", sprintf("%02d", 1:21))
# Fake years
f.y <- year.label.numeric
```

```
GDPpc <- wdi %>%
  select(country, year, gdp.capita) %>%
  filter(country %in% country.label) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, 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")

trade.df <- wdi %>%
  select(country, year, trade) %>%
  filter(country %in% country.label) %>%
  mutate(trade = std(trade)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, trade = 0))
trade <- trade.df %>%
  reshape2::acast(country ~ year, value.var = "trade")
if ( length(which(!dimnames(trade)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

constraints.d <- expand_grid(Country = country.label,
                             Year = year.label.numeric) %>%
  left_join(polcon.d) %>%
  mutate(original.polcon = polcon) %>%
  mutate(polcon = std(polcon)) %>%
  mutate(polcon = ifelse(str_detect(Country, "^Z-") & is.na(polcon), 0, polcon))
constraints <- constraints.d %>%
  reshape2::acast(Country ~ Year, value.var = "polcon")
if ( length(which(!dimnames(constraints)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")


gov.eff.df <- expand.grid(
    country = country.label,
    year = year.label.numeric,
    Sector = sector.label) %>%
  left_join(gov.eff.original) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1976 & year <= 2005) %>%
  mutate(value = std(value)) %>%
  select(Sector, country, year, value) %>%
  mutate(value = ifelse(str_detect(country, "^Z-") & is.na(value), 0, value))
gov.eff <- gov.eff.df %>%
  reshape2::acast(Sector ~ country ~ year, value.var = "value")
if ( length(which(!dimnames(gov.eff)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")


min.discard.zero <- function(x) return(min(x[x!=0]))
portfolio.size <-
  D %>%
  filter(Measure == "Size") %>%
  spread(Measure, value) %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector) %>%
  mutate(Size = ifelse(Size == 0, min.discard.zero(Size)/2, Size)) %>%
  mutate(Size = std(logit(Size))) %>%
  mutate(Size = ifelse(str_detect(Country, "Z2"), 0, Size)) %>%
  ungroup() %>%
  select(Country, Sector, Year, Size) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Size")
if ( length(which(!dimnames(portfolio.size)[[2]] == 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")

# Green socialist as salience
green.socialist <- salience %>%
  group_by(Sector) %>%
  mutate(Salience = std(Salience)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label, Country = f.c, Year = f.y, Salience = 0)) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Salience")
if ( length(which(!dimnames(green.socialist)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here

# Borders in tidy data
interdependency.contiguity <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  mutate(wDiversity = value * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
  summarize(contiguity.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(contiguity.dependency = std(contiguity.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        contiguity.dependency = 0))

interdependency.contiguity <- interdependency.contiguity %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "contiguity.dependency")
if ( length(which(!dimnames(interdependency.contiguity)[[3]] == country.label)) > 0) stop("Ep! There is a m

# Trade in tidy data
interdependency.trade <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(trade.p %>%
            ungroup() %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wDiversity = value * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
  summarize(trade.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(trade.dependency = std(trade.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        trade.dependency = 0))

interdependency.trade <- interdependency.trade %>%
```

```r
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "trade.dependency")
if ( length(which(!dimnames(interdependency.trade)[[3]] == country.label)) > 0) stop("Ep! There is a mistak

# Performance part
# Match it with the general Y, but only for the environmental sector
d.perf.fake <- expand_grid(
  Country = country.label[str_detect(country.label, "^Z-") ],
  Year = year.label.numeric,
  Indicator = unique(d.perf$Indicator),
  Performance = NA)

Y.performance <- diversity.l %>%
  select(Sector, Country, Year, Measure) %>%
  left_join(d.perf) %>%
  left_join(d.perf.fake) %>%
  filter(Sector == "Environmental") %>%
  group_by(Indicator) %>%
  mutate(Performance = std1(Performance)) %>%
  ungroup() %>%
  reshape2::acast(Indicator ~ Country ~ Year ~ Measure, value.var = "Performance")
# manually get rid of ghost performance for missing values
Y.performance <- Y.performance[-4,,,]

nP <- dim(Y.performance)[1]
performance.label <- dimnames(Y.performance)[[1]]
if ( length(which(!dimnames(Y.performance)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")

load("wdi/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 %in% year.label.numeric) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "gdp.growth")
gdp.growth <- gdp.growth[,-dim(gdp.growth)[2]]
if ( length(which(!dimnames(gdp.growth)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
gdp.growth.means <- apply(gdp.growth, 1, mean, na.rm = TRUE)
gdp.growth.means[is.nan(gdp.growth.means)] <- 0


load("wdi/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(urban = std(urban)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
urban <- urban[,-dim(urban)[2]]
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
urban.means <- apply(urban, 1, mean, na.rm = TRUE)
urban.means[is.nan(urban.means)] <- 0

load("wdi/wdi-industry.RData") # industry
industry <- industry %>%
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(industry = std(industry)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "industry")
industry <- industry[,-dim(industry)[2]]
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)
industry.means[is.nan(industry.means)] <- 0


DP <- list(
  Y = unname(Y),
  Y.performance = unname(Y.performance), nP = nP,
```

```
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  constraints = unname(constraints),
  gov.eff = unname(gov.eff),
  gov.eff.mean.observed = unname(apply(gov.eff, c(1, 2), mean, na.rm = TRUE)),
  interdependency.contiguity = unname(interdependency.contiguity),
  interdependency.trade = unname(interdependency.trade),
  portfolio.size = unname(portfolio.size),
  eu = unname(eu),
  green.socialist = unname(green.socialist),
  gdp.growth = unname(gdp.growth), gdp.growth.means = unname(gdp.growth.means),
  urban = unname(urban), urban.means = unname(urban.means),
  industry = unname(industry), industry.means = unname(industry.means),
  nC = nC,
  id.fake.countries = id.fake.countries,
  id.real.countries = id.real.countries,
  id.decade = id.decade, nDecades = nDecades,
  nB = nB, b0 = b0, B0 = B0,
  nS = nS,
  nD = nD,
  nY = nY)
nC   # Number of countries (including fake ones)
```

```
## [1] 42
```

```
nS   # Number of sectors
```

```
## [1] 2
```

```
years  # Range of years
```

```
## [1] 1976 2005
```

7.3 Model

```
M <- "diversity-mbi"
M.lab <- "Diversity (AID), robustness, Market-based instruments"
m <- "model {
  #
  # Data part at the observational level
  #
  for (d in 1:nD) {
    for (s in 1:nS) {
      for (c in 1:nC) {
        for (t in 2:nY) {
          Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
          mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
                         + beta[1,s,d] * GDPpc[c,t-1]
                         + beta[4,s,d] * gov.eff[s,c,t-1]
                         + beta[5,s,d] * portfolio.size[s,c,t-1]
                         + beta[6,s,d] * trade[c,t-1]
                         + beta[7,s,d] * eu[c,t-1]
                         + beta[8,s,d] * green.socialist[s,c,t-1] # col 1 green, col 2 socialist
                         + beta[9,s,d] * constraints[c,t-1]
                         + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
                         + beta[11,s,d] * interdependency.trade[d,s,c,t]
                         + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
          tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
          sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
                                  + lambda[2,s,d] * portfolio.size[s,c,t]
                                  + gamma[c,d])
          resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
        }
        Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
        mu[s,c,1,d] <- alpha[s,d,1]
                     + beta[1,s,d] * GDPpc[c,1]
                     + beta[4,s,d] * gov.eff[s,c,1]
                     + beta[5,s,d] * portfolio.size[s,c,1]
```

```
              + beta[6,s,d] * trade[c,1]
              + beta[7,s,d] * eu[c,1]
              + beta[8,s,d] * green.socialist[s,c,1] # col 1 green, col 2 socialist
              + beta[9,s,d] * constraints[c,1]
                + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
                + beta[11,s,d] * interdependency.trade[d,s,c,1]

    resid[s,c,1,d] ← Y[s,c,1,d] - mu[s,c,1,d]
    tau[s,c,1,d] ← 1 / sigma.sq[s,c,1,d]
    sigma.sq[s,c,1,d] ← exp(lambda[1,s,d]
                            + lambda[2,s,d] * portfolio.size[s,c,1]
                            + gamma[c,d])
  }


  #
  # Degrees of freedom of GR
  #
  nu[s,d] ← 1 + (-1 * log(nu.trans[s,d]))
  nu.trans[s,d] ~ dunif(0, 1)

  #
  # Priors for variance component
  #
  lambda[1,s,d] ~ dnorm(0, 2^-2)
  lambda[2,s,d] ~ dnorm(0, 2^-2)

  #
  # Priors for the intercept
  #
  for (decade in 1:nDecades) {
    alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
  }
  Alpha[s,d] ~ dunif(0, 1)
  tau.alpha[s,d] ← 1 / sqrt(sigma.alpha[s,d])
  sigma.alpha[s,d] ~  dunif(0, 0.5)

  #
  # Priors for the control variables
  #
  beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
  Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
  Sigma[1:nB,1:nB,s,d] ← inverse(Omega[1:nB,1:nB,s,d])
}
#
# Data part for  performance
#
for (p in 1:nP) {
  for (c in 1:nC) {
    for (t in 2:nY) {
      Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
      mu.performance[p,c,t,d] ← eta.performance[1,p,d]
                               + eta.performance[2,p,d] * resid[1,c,t,d]
                               + eta.performance[3,p,d] * Y[1,c,t-1,d]
                               + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
                               + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
                               + eta.performance[6,p,d] * GDPpc[c,t-1]
                               + eta.performance[7,p,d] * trade[c,t-1]
                               + eta.performance[8,p,d] * eu[c,t-1]
                               + eta.performance[9,p,d] * gdp.growth[c,t-1]
                               + eta.performance[10,p,d] * urban[c,t-1]
                               + eta.performance[11,p,d] * industry[c,t-1]
    }
    Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
    mu.performance[p,c,1,d] ← eta.performance[1,p,d]
                             + eta.performance[2,p,d] * resid[1,c,1,d]
                             + eta.performance[3,p,d] * Y[1,c,1,d]
                             + eta.performance[4,p,d] * portfolio.size[1,c,1]
                             + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
                             + eta.performance[6,p,d] * GDPpc[c,1]
                             + eta.performance[7,p,d] * trade[c,1]
                             + eta.performance[8,p,d] * eu[c,1]
                             + eta.performance[9,p,d] * gdp.growth[c,1]
                             + eta.performance[10,p,d] * urban[c,1]
                             + eta.performance[11,p,d] * industry[c,1]
  }
  tau.performance[p,d] ~ dgamma(0.001, 0.001)
```

```
      sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
      for (e in 1:11) {
        eta.performance[e,p,d] ~ dnorm(0, 1^-2)
      }
    }

    # Variance component, varying intercepts by country
    for (c in 1:nC) {
      gamma[c,d] ~ dnorm(0, 0.2^-2)
    }
    #
    # AR(1) parameters
    #
    for (c in id.real.countries) {
      for (s in 1:nS) {
        rho[s,c,d] ~ dunif(-1, 1)
      }
      for (p in 1:nP) {
        rho.performance[p,c,d] ~ dunif(-1, 1)
      }
    }
    for (c in id.fake.countries) {
      for (s in 1:nS) {
        rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
      }
      for (p in 1:nP) {
        rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
      }
    }
  }

  # SEM Part for portfolio size
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (t in 2:nY) {
        portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
        mu.ps[s,c,t] <- alpha.ps[s,c]
                      + delta[1,s] * GDPpc[c,t-1]
                      + delta[2,s] * gov.eff[s,c,t-1]
                      + delta[3,s] * green.socialist[s,c,t-1]
                      + delta[4,s] * constraints[c,t-1]
      }
      tau.ps[s,c] ~ dgamma(0.1, 0.1)
      sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
      alpha.ps[s,c] ~ dnorm(0, 1^-2)
    }
    for (d in 1:4) {
      delta[d,s] ~ dnorm(0, 1^-2)
    }
  }
  # Mediated effects
  for (d in 1:nD) {
    for (s in 1:nS) {
      pi[1,s,d] <- delta[1,s] * beta[1,s,d]       # GDPpc
      pi[2,s,d] <- delta[2,s] * beta[4,s,d]       # gov.eff
      pi[3,s,d] <- delta[3,s] * beta[8,s,d]       # green
      pi[4,s,d] <- delta[4,s] * beta[9,s,d]       # constraints
    }
  }


  #
  # Missing data
  #
  for (c in 1:nC) {
    for (t in 1:nY) {
      gov.eff[1,c,t] ~ dnorm(mean(gov.eff.mean.observed[1,c]), 0.05^-2)
      gov.eff[2,c,t] ~ dnorm(gov.eff[1,c,t], 100)
      gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
      urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
      industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
    }
  }
  for (s in 1:nS) {
    for (c in 1:nC) {
```

```
      # Reverse years back for NA in early years
      for (t in 1:(nY-1)) {
        green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
      }
      for (d in 1:nD) {
        for (t in 1:(nY-1)) {
          interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
        }
      }
    }
  }
}"
write(m, file= paste("models/model-", M, ".bug", sep = ""))
par ← NULL
par ← c(par, "alpha", "beta", "theta", "sigma")
par ← c(par, "Alpha", "sigma.alpha")
par ← c(par, "Sigma")
par ← c(par, "lambda", "gamma")
par ← c(par, "nu")
par ← c(par, "rho")
par.fake ← expand_grid(Sector = 1:nS,
                        Country = id.fake.countries,
                        Year = 2:nY,
                        Diversity = 1:nD) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par ← c(par, par.fake)
par.resid ← expand_grid(Sector = 1:nS,
                         Country = id.real.countries,
                         Year = 2:nY,
                         Diversity = 1:nD) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par ← c(par, par.resid)
par ← c(par, "delta", "alpha.ps", "sigma.ps", "pi")
par ← c(par, "eta.performance", "sigma.performance")
inits ← list(
  list(.RNG.name="base::Super-Duper", .RNG.seed=10),
  list(.RNG.name="base::Super-Duper", .RNG.seed=20),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=30),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=20))

t0 ← proc.time()
rj ← run.jags(model = paste("models/model-", M, ".bug", sep = ""),
  data = dump.format(DP, checkvalid=FALSE),
  inits = inits,
  modules = "glm",
  n.chains = 1, adapt = 2e2, burnin = 1e3, sample = 2e3, thin = 1,
  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 = ""))

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

```
## model {
##   #
##     #
##   for (d in 1:nD) {
##     for (s in 1:nS) {
##       for (c in 1:nC) {
##         for (t in 2:nY) {
```

```
##          Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
##          mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
##                   + beta[1,s,d] * GDPpc[c,t-1]
##                   + beta[4,s,d] * gov.eff[s,c,t-1]
##                   + beta[5,s,d] * portfolio.size[s,c,t-1]
##                   + beta[6,s,d] * trade[c,t-1]
##                   + beta[7,s,d] * eu[c,t-1]
##                   + beta[8,s,d] * green.socialist[s,c,t-1]                    + beta[9,s,d] * constrain
##                   + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
##                   + beta[11,s,d] * interdependency.trade[d,s,c,t]
##                   + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
##          tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
##          sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
##                                 + lambda[2,s,d] * portfolio.size[s,c,t]
##                                 + gamma[c,d])
##          resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
##        }
##        Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
##        mu[s,c,1,d] <- alpha[s,d,1]
##                 + beta[1,s,d] * GDPpc[c,1]
##                 + beta[4,s,d] * gov.eff[s,c,1]
##                 + beta[5,s,d] * portfolio.size[s,c,1]
##                 + beta[6,s,d] * trade[c,1]
##                 + beta[7,s,d] * eu[c,1]
##                 + beta[8,s,d] * green.socialist[s,c,1]                  + beta[9,s,d] * constraints[c,1
##                   + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
##                   + beta[11,s,d] * interdependency.trade[d,s,c,1]
##
##        resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
##        tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
##        sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
##                                 + lambda[2,s,d] * portfolio.size[s,c,1]
##                                 + gamma[c,d])
##      }
##
##
##      #
##            #
##      nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
##      nu.trans[s,d] ~ dunif(0, 1)
##
##      #
##            #
##      lambda[1,s,d] ~ dnorm(0, 2^-2)
##      lambda[2,s,d] ~ dnorm(0, 2^-2)
##
##      #
##            #
##      for (decade in 1:nDecades) {
```

```
##          alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
##        }
##        Alpha[s,d] ~ dunif(0, 1)
##        tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
##        sigma.alpha[s,d] ~  dunif(0, 0.5)
##
##        #
##              #
##        beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
##        Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
##        Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
##      }
##      #
##          #
##      for (p in 1:nP) {
##        for (c in 1:nC) {
##          for (t in 2:nY) {
##            Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
##            mu.performance[p,c,t,d] <- eta.performance[1,p,d]
##                                       + eta.performance[2,p,d] * resid[1,c,t,d]
##                                       + eta.performance[3,p,d] * Y[1,c,t-1,d]
##                                       + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
##                                       + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
##                                       + eta.performance[6,p,d] * GDPpc[c,t-1]
##                                       + eta.performance[7,p,d] * trade[c,t-1]
##                                       + eta.performance[8,p,d] * eu[c,t-1]
##                                       + eta.performance[9,p,d] * gdp.growth[c,t-1]
##                                       + eta.performance[10,p,d] * urban[c,t-1]
##                                       + eta.performance[11,p,d] * industry[c,t-1]
##          }
##          Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
##          mu.performance[p,c,1,d] <- eta.performance[1,p,d]
##                                     + eta.performance[2,p,d] * resid[1,c,1,d]
##                                     + eta.performance[3,p,d] * Y[1,c,1,d]
##                                     + eta.performance[4,p,d] * portfolio.size[1,c,1]
##                                     + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
##                                     + eta.performance[6,p,d] * GDPpc[c,1]
##                                     + eta.performance[7,p,d] * trade[c,1]
##                                     + eta.performance[8,p,d] * eu[c,1]
##                                     + eta.performance[9,p,d] * gdp.growth[c,1]
##                                     + eta.performance[10,p,d] * urban[c,1]
##                                     + eta.performance[11,p,d] * industry[c,1]
##        }
##        tau.performance[p,d] ~ dgamma(0.001, 0.001)
##        sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
##        for (e in 1:11) {
##          eta.performance[e,p,d] ~ dnorm(0, 1^-2)
##        }
##      }
```

```
##
##        for (c in 1:nC) {
##      gamma[c,d] ~ dnorm(0, 0.2^-2)
##    }
##    #
##        #
##    for (c in id.real.countries) {
##      for (s in 1:nS) {
##        rho[s,c,d] ~ dunif(-1, 1)
##      }
##      for (p in 1:nP) {
##        rho.performance[p,c,d] ~ dunif(-1, 1)
##      }
##    }
##    for (c in id.fake.countries) {
##      for (s in 1:nS) {
##        rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##      }
##      for (p in 1:nP) {
##        rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##      }
##    }
##  }
##
##    for (s in 1:nS) {
##    for (c in 1:nC) {
##      for (t in 2:nY) {
##        portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
##        mu.ps[s,c,t] <- alpha.ps[s,c]
##                       + delta[1,s] * GDPpc[c,t-1]
##                       + delta[2,s] * gov.eff[s,c,t-1]
##                       + delta[3,s] * green.socialist[s,c,t-1]
##                       + delta[4,s] * constraints[c,t-1]
##      }
##      tau.ps[s,c] ~ dgamma(0.1, 0.1)
##      sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
##      alpha.ps[s,c] ~ dnorm(0, 1^-2)
##    }
##    for (d in 1:4) {
##      delta[d,s] ~ dnorm(0, 1^-2)
##    }
##  }
##    for (d in 1:nD) {
##    for (s in 1:nS) {
##      pi[1,s,d] <- delta[1,s] * beta[1,s,d]             pi[2,s,d] <- delta[2,s] * beta[4,s,d]
##  }
##
##
##
```

```
##   #
##     #
##   for (c in 1:nC) {
##     for (t in 1:nY) {
##       gov.eff[1,c,t] ~ dnorm(mean(gov.eff.mean.observed[1,c]), 0.05^-2)
##       gov.eff[2,c,t] ~ dnorm(gov.eff[1,c,t], 100)
##       gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
##       urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
##       industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
##     }
##   }
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##             for (t in 1:(nY-1)) {
##         green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
##       }
##       for (d in 1:nD) {
##         for (t in 1:(nY-1)) {
##           interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
##         }
##       }
##     }
##   }
## }
```

```
ggmcmc(ggs(s, family = "sigma|alpha|beta|lambda|rho|nu"),
       param_page = 10, file = paste("ggmcmc-full-", M, ".pdf", sep = ""))
```

7.4 Model results

Variance components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                              "Portfolio size",
                                              "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda, label = "Variable", comparison = "Sector") +
  aes(color = Sector) +
  facet_wrap(~ Diversity) +
  theme(legend.position = "right") +
  ggtitle("Variance component")
```

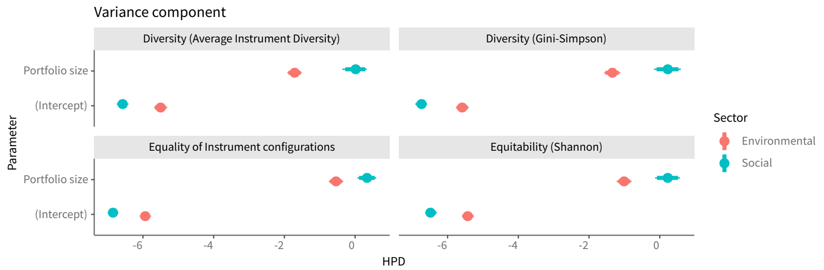

Figure 7.1: Variance component.

```r
rm(S.lambda)

L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                              "Portfolio size",
                                              "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
S.lambda %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable") +
  ggtitle("Variance component")
```

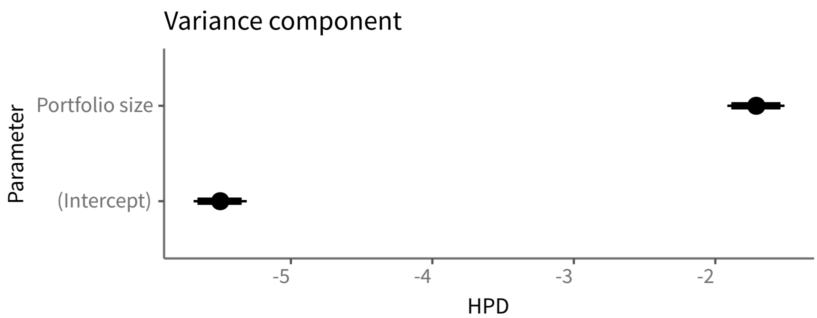

Figure 7.2: Variance component.

```r
rm(S.lambda)
```

Variance components (country varying intercepts).

```r
L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.gamma, label = "Country") +
  facet_grid(~ Diversity) +
  ggtitle("Variance component (countries)")
```

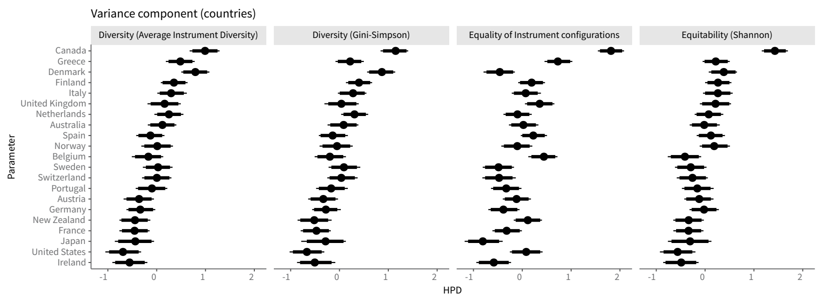

Figure 7.3: Variance component (country varying intercepts).

```r
rm(S.gamma)

L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-")) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)")
ggs_caterpillar(S.gamma, label = "Country") +
  ggtitle("Variance component (countries)")
```

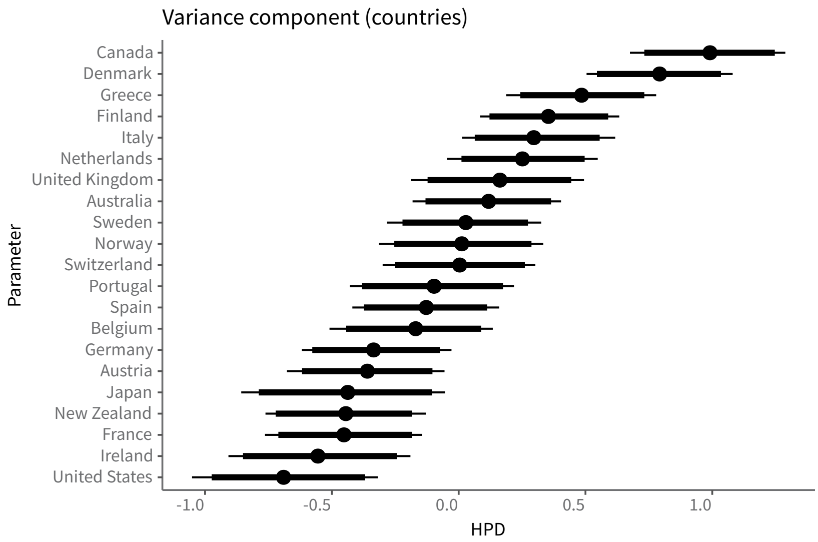

Figure 7.4: Variance component (country varying intercepts).

```
rm(S.gamma)
```

Auto-regressive components.

```
L.rho <- plab("rho", list(Sector = sector.label,
                          Country = country.label,
                          Diversity = diversity.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  facet_grid(Diversity ~ Sector, scales="free") +
  aes(color=Sector) +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()
```

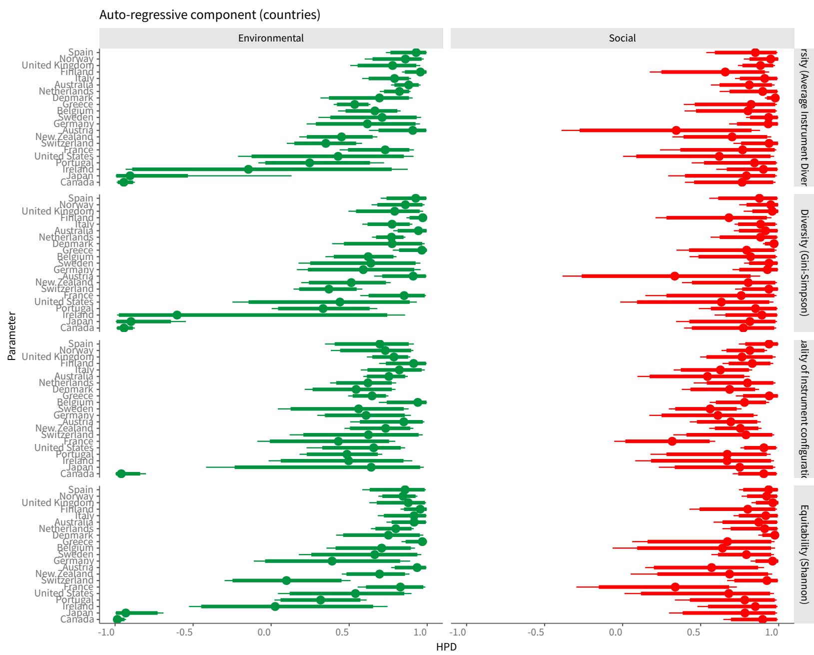

Figure 7.5: Auto-regressive component.

```
rm(S.rho)

L.rho <- plab("rho", list(Sector = sector.label,
                          Country = country.label,
                          Diversity = diversity.label))
S.rho <- ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()
```

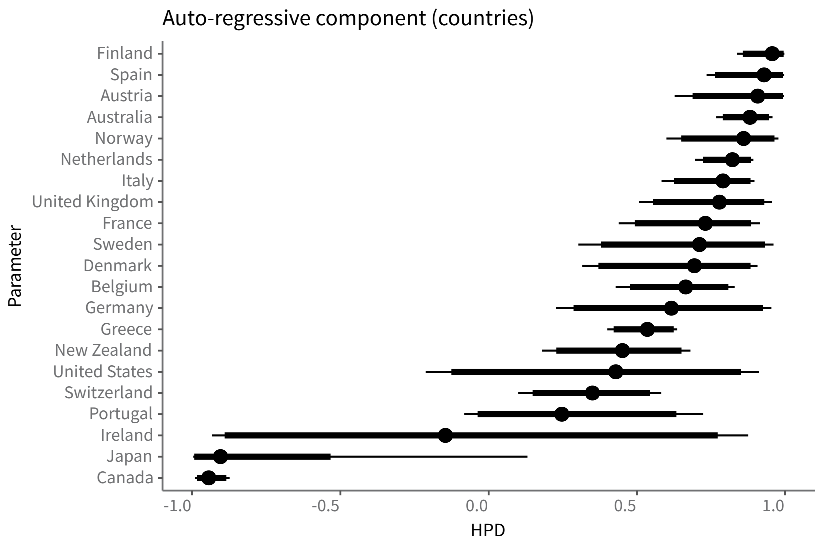

Figure 7.6: Auto-regressive component.

```
rm(S.rho)
```

Intercepts

```
L.alpha <- plab("alpha", list(Sector = sector.label,
                              Diversity = diversity.label,
                              Decade = decade.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_label = L.alpha)
ci(S.alpha) %>%
  ggplot(aes(x = Decade, y = median, ymin = Low, ymax = High)) +
  geom_point() +
  geom_linerange() +
  facet_grid(Diversity ~ Sector) +
  ggtitle("Temporal trend")

covariates.label <- c("GDP pc",
                      "Consensus",
                      "Deliberation",
                      "Government effectiveness",
                      "Portfolio size",
                      "Trade", "EU", "Salience",
                      "Political constraints",
                      "Interdependency (Contiguity)",
                      "Interdependency (Trade)")
L.betas <- plab("beta", list(Variable = covariates.label,
                             Sector = sector.label,
                        Diversity = diversity.label))
```

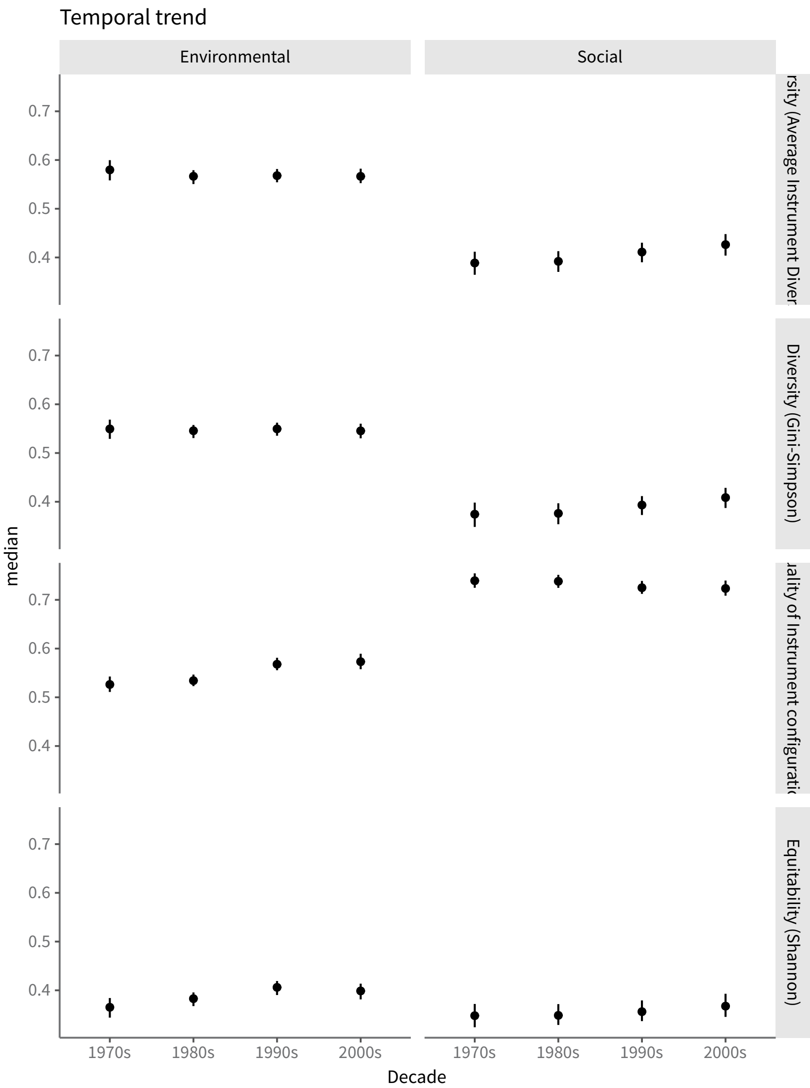

Figure 7.7: Intercepts.

```
S.betas <- ggs(s, family = "^beta\\[", par_labels = L.betas) %>%
  filter(!Variable %in% c("Deliberation", "Consensus"))
ci.betas <- ci(S.betas)
save(ci.betas, file = paste0("ci-betas-", M, ".RData"))

S.betas %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	-0.04	0.02	0.00	1.00	-0.04
GDP pc	Social	-0.08	0.05	0.25	0.75	-0.06
Government effectiveness	Environmental	0.39	0.11	1.00	0.00	0.34
Government effectiveness	Social	0.40	0.09	1.00	0.00	0.37
Portfolio size	Environmental	0.15	0.03	1.00	0.00	0.16
Portfolio size	Social	0.13	0.02	1.00	0.00	0.14
Trade	Environmental	-0.01	0.01	0.17	0.83	-0.01
Trade	Social	-0.06	0.01	0.00	1.00	-0.06
EU	Environmental	0.00	0.01	0.34	0.66	0.00
EU	Social	0.02	0.01	0.95	0.05	0.02
Salience	Environmental	-0.02	0.01	0.00	1.00	-0.02
Salience	Social	-0.02	0.01	0.11	0.89	-0.02
Political constraints	Environmental	-0.06	0.02	0.02	0.98	-0.05
Political constraints	Social	-0.12	0.03	0.00	1.00	-0.11
Interdependency (Contiguity)	Environmental	0.06	0.02	1.00	0.00	0.07
Interdependency (Contiguity)	Social	-0.02	0.02	0.30	0.70	-0.02
Interdependency (Trade)	Environmental	0.02	0.03	0.72	0.28	0.00
Interdependency (Trade)	Social	0.00	0.01	0.35	0.65	0.00

```
ci(S.betas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()


ci(S.betas) %>%
  filter(Sector == "Environmental") %>%
ggplot(aes(x = Variable,
              y = median,
              group = Diversity,
              color = Diversity)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

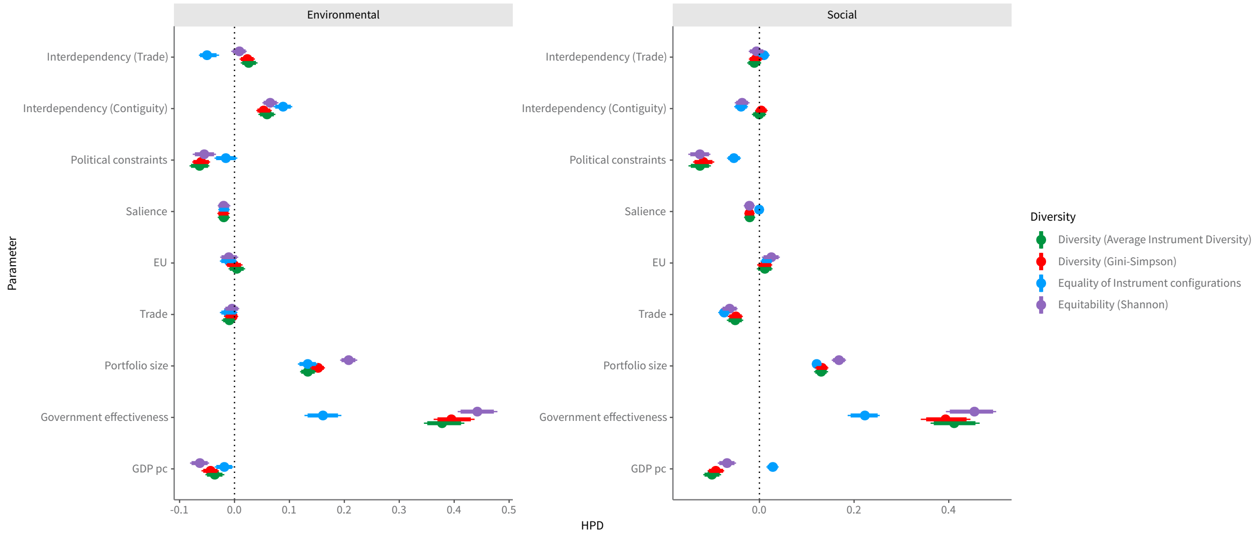

Figure 7.8: Slopes with the effects on diversity, by sector.

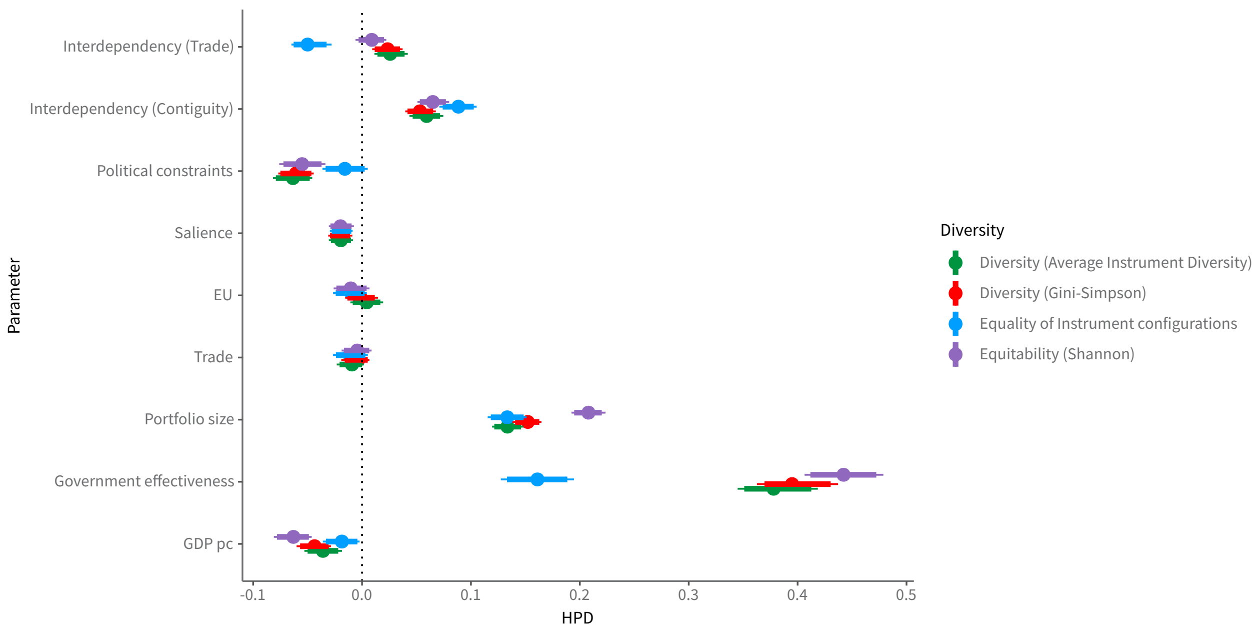

Figure 7.9: Slopes with the effects on diversity, for the environmental sector.

```
ci(S.betas) %>%
ggplot(aes(x = Variable,
           y = median,
           group = Sector, color = Sector)) +
  coord_flip() +
  facet_wrap(~ Diversity, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

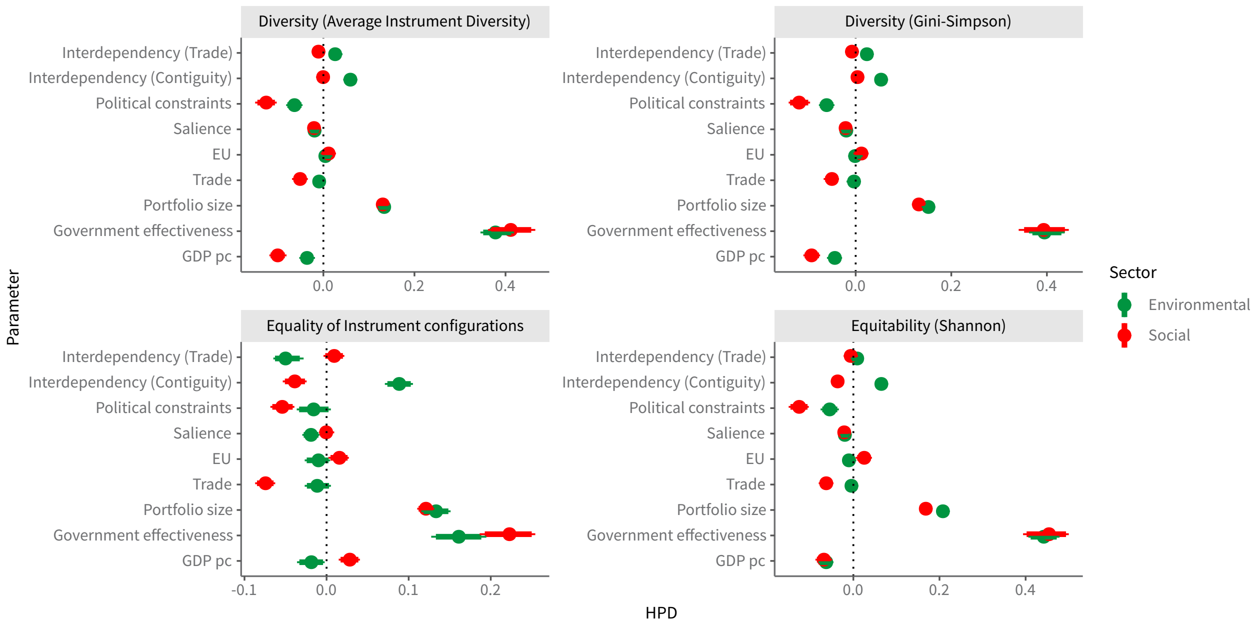

Figure 7.10: Slopes with the effects on diversity, by diversity indicator.

```
ci(S.betas) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggplot(aes(x = Variable,
           y = median,
           group = Sector, color = Sector)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

```
ggplot(S.betas, aes(x = value, color = Sector, fill = Sector)) +
  geom_density(alpha = 0.5) +
  facet_grid(Diversity ~ Variable, scales = "free") +
  xlab("HPD") +
  geom_vline(xintercept = 0, lty = 3)
```

Variables by evidence.

```
S.betas %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(value != 0) %>%
  group_by(Sector, Diversity, Variable) %>%
  summarize(`Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
```

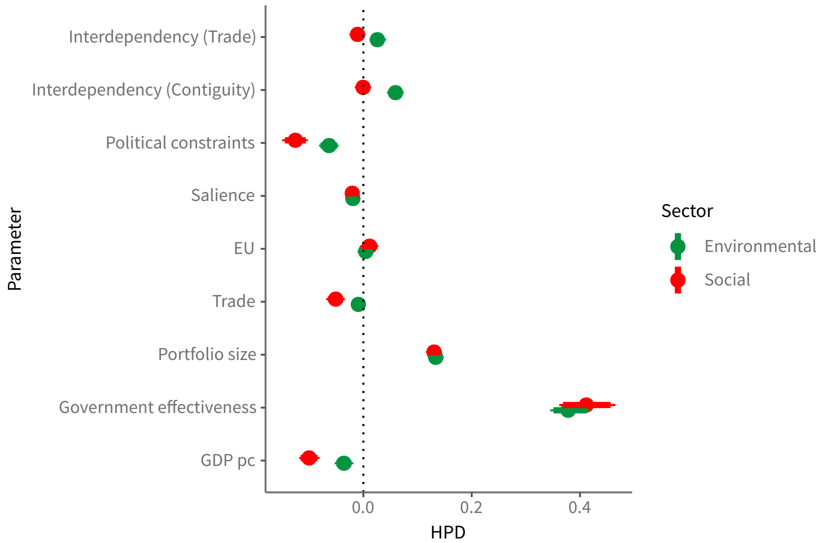

Figure 7.11: Slopes with the effects on AID diversity, by sector

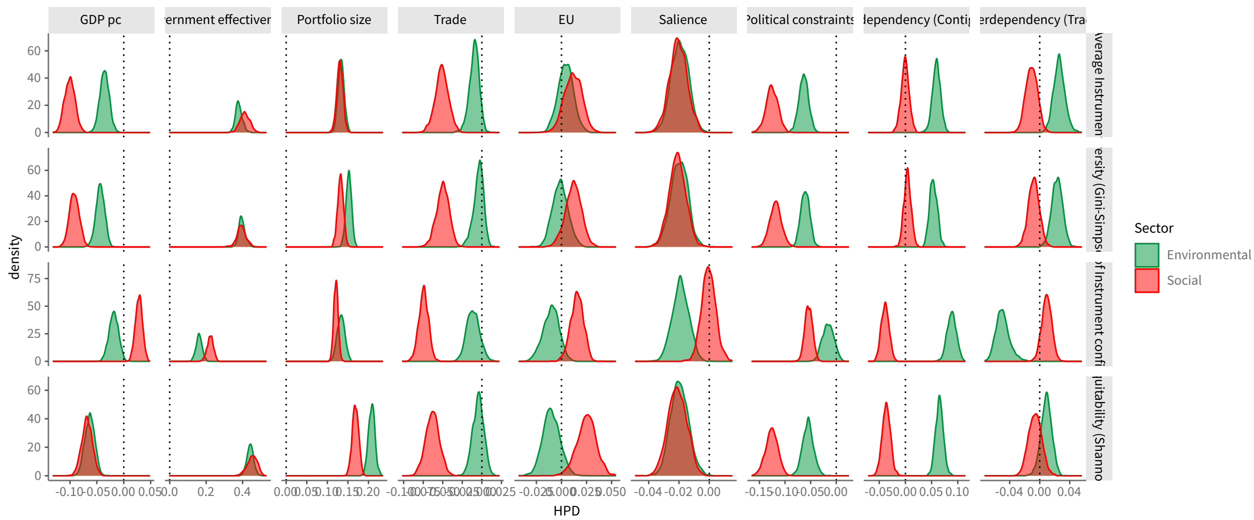

Figure 7.12: Slopes with the effects on diversity, comparing sectors.

```
    `Mean expected effect` = mean(value)) %>%
  group_by(Sector, Diversity, Variable) %>%
  mutate(max = max(abs(`Prob > 0`), abs(`Prob < 0`))) %>%
  arrange(desc(max)) %>%
  select(-max, -Diversity) %>%
  kable()
```

Diversity	Sector	Variable	Prob > 0	Prob < 0	Mean expected effect
Diversity (Average Instrument Diversity)	Environmental	GDP pc	0.00	1.00	-0.04
Diversity (Average Instrument Diversity)	Environmental	Government effectiveness	1.00	0.00	0.38
Diversity (Average Instrument Diversity)	Environmental	Portfolio size	1.00	0.00	0.13
Diversity (Average Instrument Diversity)	Environmental	Salience	0.00	1.00	-0.02
Diversity (Average Instrument Diversity)	Environmental	Political constraints	0.00	1.00	-0.06
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Contiguity)	1.00	0.00	0.06
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Trade)	1.00	0.00	0.03
Diversity (Average Instrument Diversity)	Social	GDP pc	0.00	1.00	-0.10
Diversity (Average Instrument Diversity)	Social	Government effectiveness	1.00	0.00	0.41
Diversity (Average Instrument Diversity)	Social	Portfolio size	1.00	0.00	0.13
Diversity (Average Instrument Diversity)	Social	Trade	0.00	1.00	-0.05
Diversity (Average Instrument Diversity)	Social	Salience	0.00	1.00	-0.02
Diversity (Average Instrument Diversity)	Social	Political constraints	0.00	1.00	-0.13
Diversity (Average Instrument Diversity)	Environmental	Trade	0.05	0.95	-0.01
Diversity (Average Instrument Diversity)	Social	Interdependency (Trade)	0.08	0.92	-0.01
Diversity (Average Instrument Diversity)	Social	EU	0.91	0.09	0.01
Diversity (Average Instrument Diversity)	Environmental	EU	0.71	0.29	0.00
Diversity (Average Instrument Diversity)	Social	Interdependency (Contiguity)	0.47	0.53	0.00

```
S.betas.env <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) #%>%

ggs_caterpillar(S.betas.env) +
  geom_vline(xintercept = 0, lty = 3)
```

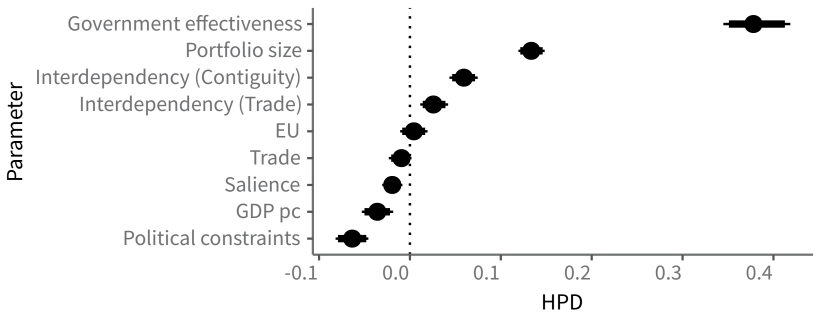

Figure 7.13: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env)

S.betas.env.sorted <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) %>%
  mutate(Parameter = fct_relevel(Parameter, rev(c("Political constraints",
                                                  "Government effectiveness",
                                 "Salience",
                                 "GDP pc", "Trade", "EU",
                                 "Interdependency (Contiguity)",
                                 "Interdependency (Trade)",
                                 "Portfolio size"))))

ggs_caterpillar(S.betas.env.sorted, sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)
```

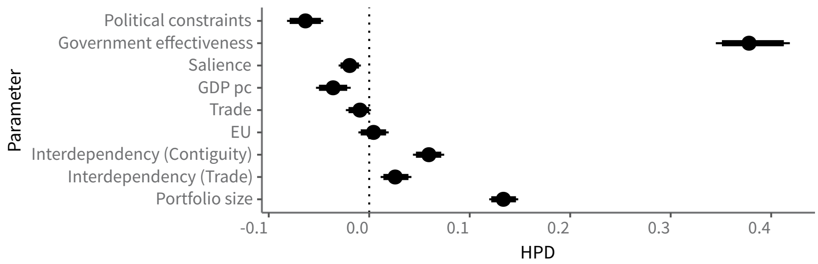

Figure 7.14: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env.sorted)

ci.betas <- ci(S.betas) %>%
  mutate(Model = M.lab)

save(ci.betas, file = paste0("ci_betas-", M, ".RData"))
rm(S.betas, ci.betas)

L.deltas <- plab("delta", list(Variable = c("GDP pc",
                                             "Government effectiveness",
                                             "Salience",
                                             "Political constraints"),
                               Sector = sector.label))

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

S.deltas %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	1.45	0.07	1.00	0.00	1.45
GDP pc	Social	0.45	0.04	1.00	0.00	0.45
Government effectiveness	Environmental	-0.18	0.17	0.14	0.86	-0.21
Government effectiveness	Social	-0.26	0.11	0.03	0.97	-0.27
Salience	Environmental	0.13	0.03	1.00	0.00	0.14
Salience	Social	-0.04	0.02	0.02	0.98	-0.04
Political constraints	Environmental	0.14	0.06	0.99	0.01	0.15
Political constraints	Social	0.24	0.06	1.00	0.00	0.24

```
ci(S.deltas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = Sector)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)


rm(S.deltas)

L.pi <- plab("pi", list(Variable = c("GDP pc",
                                      "Government effectiveness",
                                      "Salience",
                                      "Political constraints"),
                        Sector = sector.label,
```

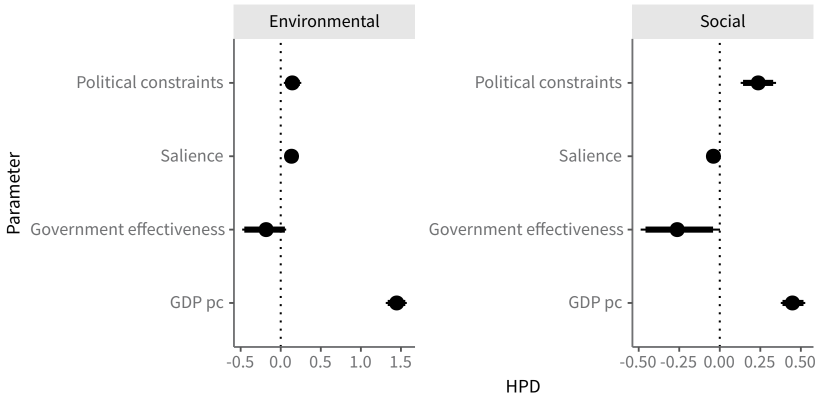

Figure 7.15: Slopes with the direct effects on portfolio size, by sector.

```
                          Diversity = diversity.label))
S.pis ← ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable ≠ "GDP pc")
S.pis %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Government effectiveness	Environmental	-0.06	0.06	0.14	0.86	-0.07
Government effectiveness	Social	-0.10	0.05	0.03	0.97	-0.10
Salience	Environmental	0.00	0.00	0.00	1.00	0.00
Salience	Social	0.00	0.00	0.87	0.13	0.00
Political constraints	Environmental	-0.01	0.00	0.03	0.97	-0.01
Political constraints	Social	-0.03	0.01	0.00	1.00	-0.03

```
ci(S.pis) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

```
L.pi ← plab("pi", list(Variable = c("GDP pc",
                                    "Government effectiveness",
                                    "Salience",
                                    "Political constraints"),
                       Sector = sector.label,
                       Diversity = diversity.label))
S.pis ← ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable ≠ "GDP pc")
S.pis %>%
```

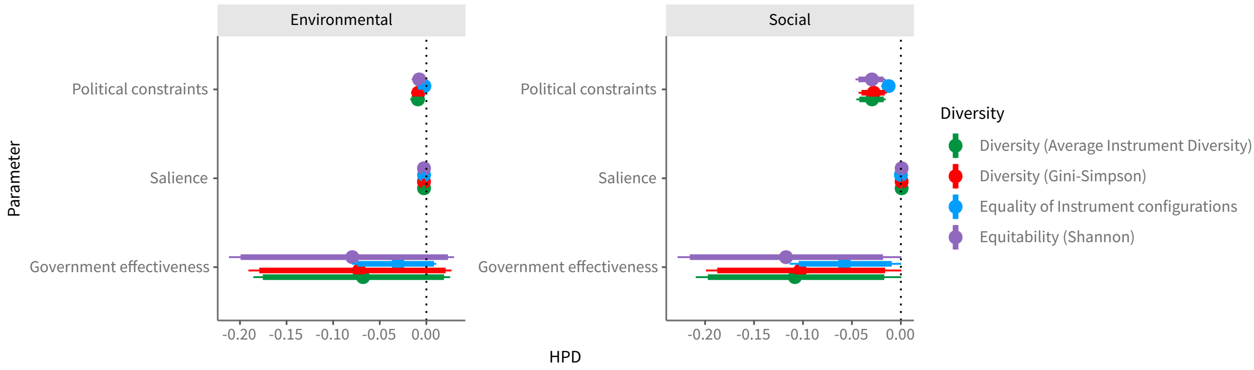

Figure 7.16: Mediated effects on diversity, by sector.

```
group_by(Variable, Sector) %>%
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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Government effectiveness	Environmental	-0.06	0.06	0.14	0.86	-0.07
Government effectiveness	Social	-0.10	0.05	0.03	0.97	-0.10
Salience	Environmental	0.00	0.00	0.00	1.00	0.00
Salience	Social	0.00	0.00	0.87	0.13	0.00
Political constraints	Environmental	-0.01	0.00	0.03	0.97	-0.01
Political constraints	Social	-0.03	0.01	0.00	1.00	-0.03

```
S.pis %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable")
```

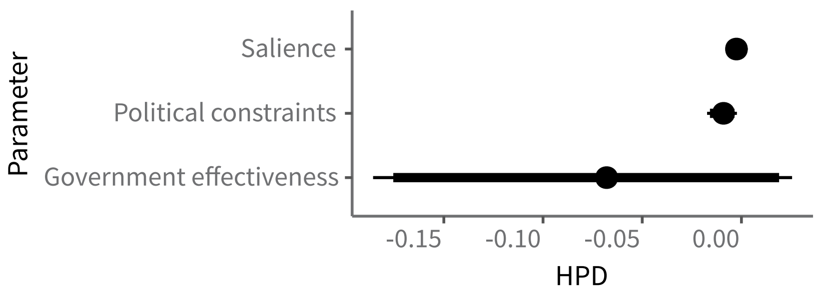

Figure 7.17: Mediated effects on diversity, for AID and the environmental sector.

```
rm(S.pis)
```

```
L.eta <- plab("eta.performance", list(Covariate = c("(Intercept)", "Residuals",
                                      "Diversity", "Portfolio size",
                                      "Diversity * Portfolio size",
                                      "GDPpc", "Trade", "EU",
                                      "GDP growth", "Urban", "Industry"),
                          Performance = performance.label,
                          Diversity = diversity.label))
S.eta <- ggs(s, family = "eta.performance", par_label = L.eta)
```

```
ggs_caterpillar(S.eta, label = "Covariate", comparison = "Performance") +
```

```
  geom_hline(yintercept = 0, lty = 3) +
  aes(color = Performance) +
  theme(legend.position = "right") +
  facet_wrap(~ Diversity)
```

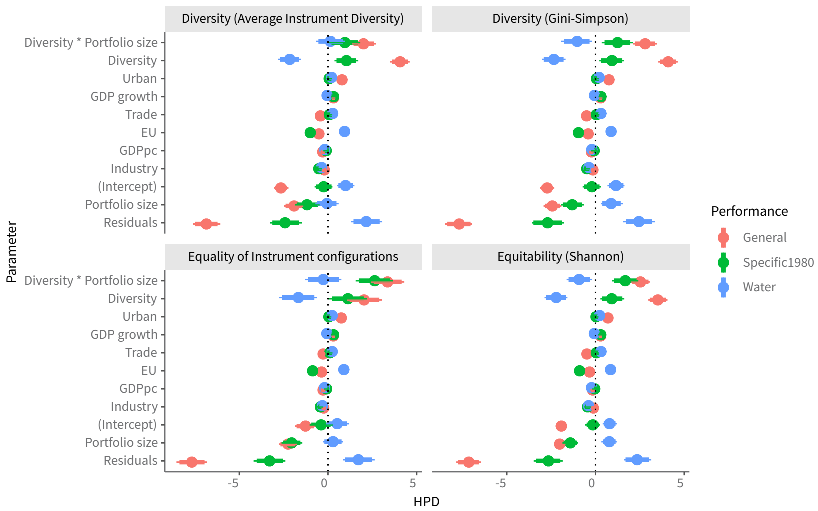

Figure 7.18: Direct effects on Environmental performance, by performance indicator and Diversity measure.

```
S.eta %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Performance == "General") %>%
ggs_caterpillar(label = "Covariate") +
  geom_vline(xintercept = 0, lty = 3)
```

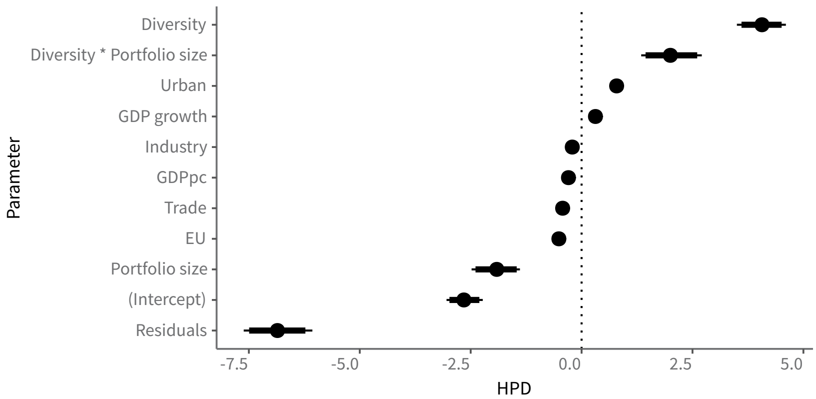

Figure 7.19: Direct effects on Environmental performance, for general performance and using AID as diversity.

```
rm(S.eta)
```

7.5 V-Cov

```
L.Sigma <- plab("Sigma", list(Covariate.1 = covariates.label,
                              Covariate.2 = covariates.label,
                              Sector = sector.label,
                              Diversity = diversity.label))
S.Sigma <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma) %>%
  filter(!Covariate.1 %in% c("Deliberation", "Consensus")) %>%
  filter(!Covariate.2 %in% c("Deliberation", "Consensus"))
```

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

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

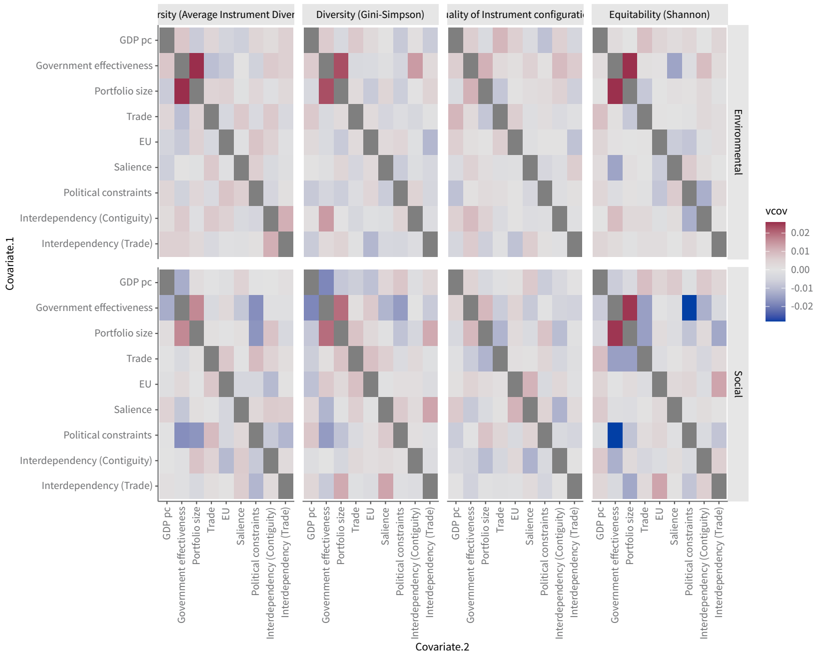

Figure 7.20: Variance-covariance matrix of main effects.

```
rm(S.Sigma, vcov.Sigma)
```

7.6 Model evaluation

What is the model fit?

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

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1, Country = Var2,
         Year = Var3, Diversity = Var4,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector, Diversity) %>%
  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, Diversity) %>%
  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(Diversity ~ Sector) +
  expand_limits(x = 0)
```

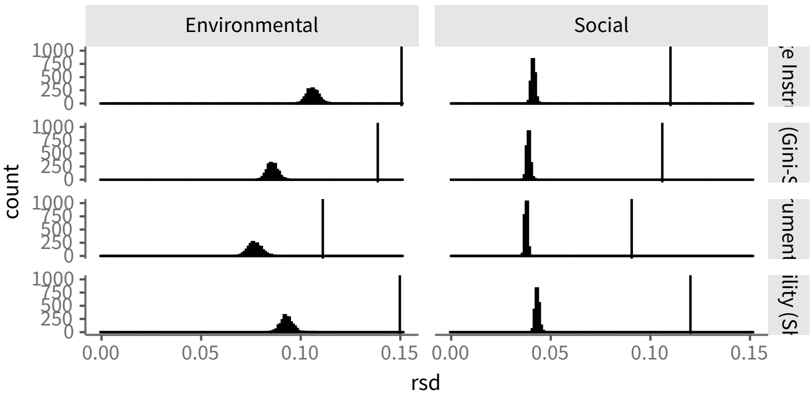

Figure 7.21: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

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

Sector	Diversity	Pseudo.R2
Environmental	Diversity (Average Instrument Diversity)	0.30
Environmental	Diversity (Gini-Simpson)	0.38
Environmental	Equality of Instrument configurations	0.31
Environmental	Equitability (Shannon)	0.38
Social	Diversity (Average Instrument Diversity)	0.63
Social	Diversity (Gini-Simpson)	0.63
Social	Equality of Instrument configurations	0.58
Social	Equitability (Shannon)	0.64

```
rm(S.rsd)
```

Which are the observations with higher residuals, further away from the expectation?[1]

[1] Negative residuals are cases where the country has lower diversity than expected according to the model.

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

ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(!str_detect(Country, "Z-")) %>%
  group_by(Country) %>%
  summarize(Residual = mean(value)) %>%
  mutate(`Mean absolute residual` = abs(Residual)) %>%
  ungroup() %>%
  arrange(desc(`Mean absolute residual`)) %>%
  select(-`Mean absolute residual`) %>%
  slice(1:8) %>%
  kable()
```

Country	Residual
Canada	-0.19
Greece	0.14
Australia	0.11
Finland	-0.09
Netherlands	-0.08
Norway	0.07
Portugal	0.06
Denmark	0.05

7.7 *Magnitude of effects*

```
L.mu <- plab("mu", list(Sector = sector.label,
                        Country = country.label,
                        Year = year.label.numeric,
                        Diversity = diversity.label)) %>%
  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()

ci.mu %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  ggplot(aes(x = Year, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen portfolio size goes\nfrom minimum to maximum\nover time")
```

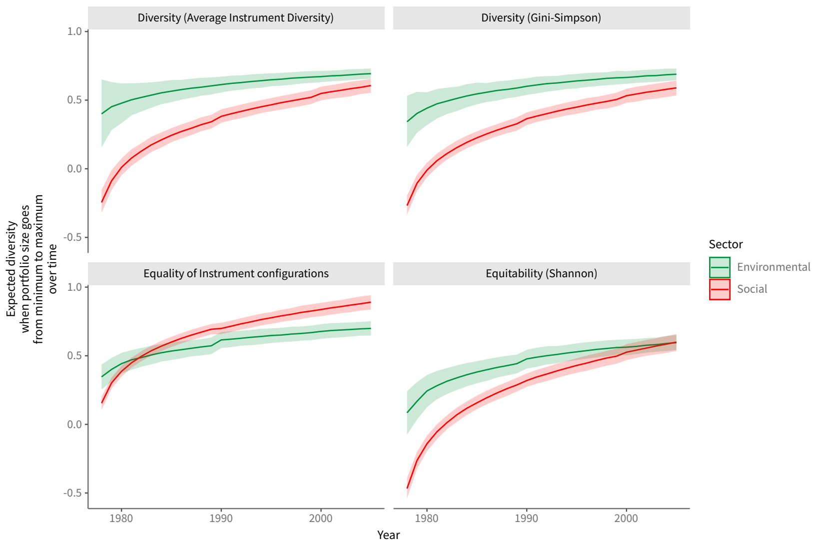

Figure 7.22: Magnitude of the effects: change in the expected diversity when portfolio size goes from the minimum to the maximum observed value over time, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year == max(Year)) %>%
  left_join(gov.eff.df %>%
            rename(Country = country, Year = year)) %>%
  rename(`Government effectiveness` = value) %>%
  ggplot(aes(x = `Government effectiveness`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
```

```
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen government effectiveness goes\nfrom minimum to maximum")
```

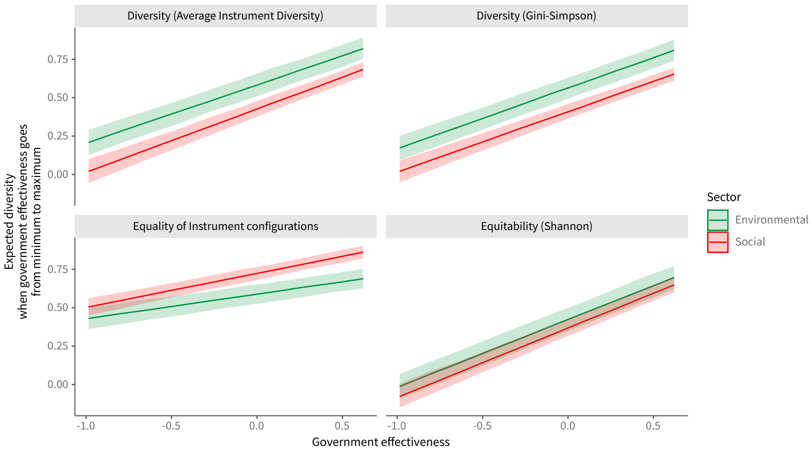

Figure 7.23: Magnitude of the effects: change in the expected diversity when government effectiveness goes from the minimum to the maximum observed value, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year == max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = polcon) %>%
  ggplot(aes(x = `Political constraints`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen political constraints go\nfrom minimum to maximum")
```

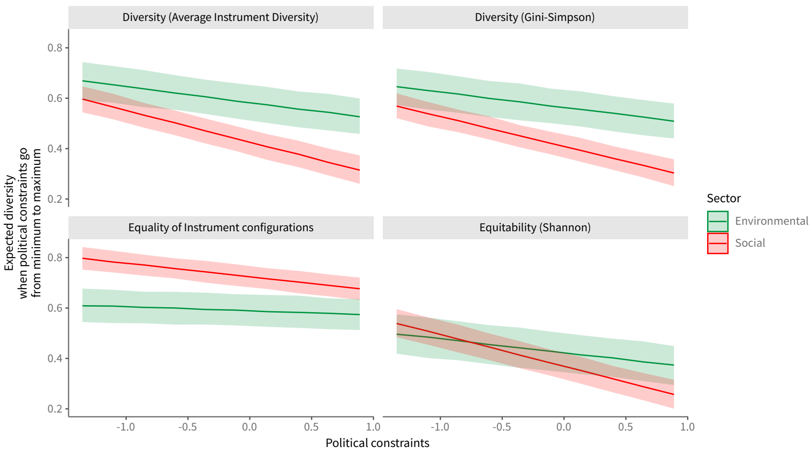

Figure 7.24: Magnitude of the effects: change in the expected diversity when political constraints go from the minimum to the maximum observed value, with the rest of the variables at their means.

```
f1 <-
  ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  left_join(D %>%
            filter(Measure == "Size") %>%
            select(Country, Sector, Year, value) %>%
```

```
            rename(`Portfolio size` = value)) %>%
  ggplot(aes(x = `Portfolio size`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(a)") +
  ylab("Expected diversity")

f2 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year == max(Year)) %>%
  left_join(gov.eff.df %>%
            rename(Country = country, Year = year)) %>%
  rename(`Government effectiveness` = value) %>%
  ggplot(aes(x = `Government effectiveness`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(b)") +
  ylab("Expected diversity")

f3 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year == max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = original.polcon) %>%
  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, 1)) +
  ggtitle("(c)") +
  ylab("Expected diversity")

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

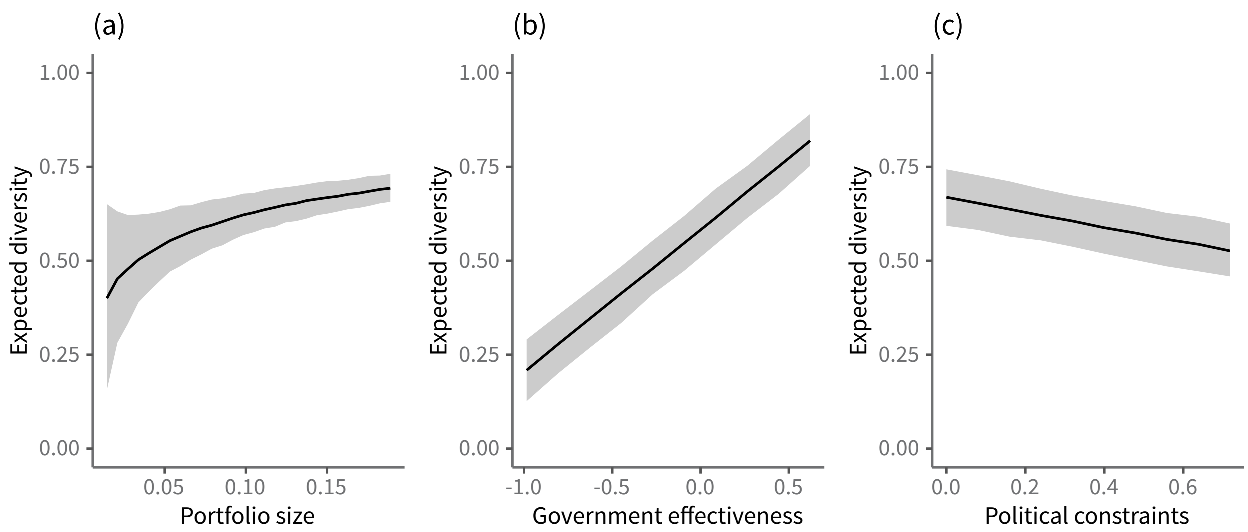

Figure 7.25: Magnitude of the effects: change in the expected diversity when (a) portfolio size, (b) government effectiveness or (c) political constraints move from the minimum to the maximum observed. In all cases the rest of the variables are fixed at their means.

8

Explanatory model of instrument diversity (robustness, with Democracy)

Data is TSCS, by sector.

```
library(PolicyPortfolios)
data(consensus)

D <- bind_rows(
  # Calculate portfolio measures sector by sector
  consensus %>%
    filter(Sector == "Environmental") %>%
    droplevels() %>%
    pp_measures(),
  consensus %>%
    filter(Sector == "Social") %>%
    droplevels() %>%
    pp_measures())


# Add more fake countries

# Z-01 Increases portfolio size from minimum to maximum observed size
#      over time
range.size.env <- range(filter(D, Sector == "Environmental" & Measure == "Size")$value)
range.size.soc <- range(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Environmental", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.env[1], range.size.env[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

D <- bind_rows(D,
  tibble(Country = "Z-01", Sector = "Social", Year = min(D$Year):max(D$Year),
         Measure = "Size", value = seq(range.size.soc[1], range.size.soc[2],
                                       length.out = length(min(D$Year):max(D$Year)))))

# Z-02, Z-11 Fixes government effectiveness to its mean
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 2:11)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))


# Z-12:Z-21 Fixes portfolio size for veto players to move
```

```
mean.size.env <- mean(filter(D, Sector == "Environmental" & Measure == "Size")$value)
mean.size.soc <- mean(filter(D, Sector == "Social" & Measure == "Size")$value)

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Environmental",
                     Measure = "Size",
                     value = mean.size.env),
              Year = min(D$Year):max(D$Year)))

D <- bind_rows(D,
  expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                     Sector = "Social",
                     Measure = "Size",
                     value = mean.size.soc),
              Year = min(D$Year):max(D$Year)))


diversity <- D %>%
  mutate(Country = as.factor(Country)) %>%
  select(-Measure) %>%
  rename(Measure = Measure.label) %>%
  spread(Measure, value) %>%
  mutate(`Diversity/W (Average Instrument Diversity)` =
         `Diversity (Average Instrument Diversity)` / (1 - `Proportion of targets covered`)) %>%
  select(Sector, Country, Year,
    `Diversity (Average Instrument Diversity)`,
    `Diversity (Gini-Simpson)`,
    `Equitability (Shannon)`,
    `Equality of Instrument configurations`)

countries <- as.character(unique(D$Country))
nC <- length(countries)
years <- range(D$Year)
```

8.1 Performance

```
perf <- foreign::read.dta("jahn/PoEP_Replication_Data/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% 1980:2010)

d.perf <- perf %>%
  # Delete Years for which we don't have data
  filter(Year >= 1980 & Year <= 2005) %>%
  gather(Indicator, Performance, -Country, -Year) %>%
  # Select specific performance indicators
  filter(Indicator %in% c("General", "Water", "Specific1980")) %>%
  droplevels()

Y.performance <- reshape2::acast(d.perf, Indicator ~ Country ~ Year, value.var = "Performance")
nYperformance <- dim(Y.performance)[3]
```

8.2 Covariates

World Development Indicators - Revenue.

```r
load("wdi/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

```r
load("wdi/wdi.RData")
wdi <- wdi[,c("country", "year", "gdp", "population", "gdp.capita", "trade")]
wdi <- subset(wdi, year >= 1976 & year <=2005)
wdi$country[wdi$country=="Korea, Rep."] <- "South Korea"
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(pred

#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(


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(pred

# GDP per capita growth
# Another way at looking at resources, is to calculate
# the 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$`2005`/wdi.gcr.w$`1976`)


# 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)
```

Democracy:

V-Dem, use the average of the 5 dimensions

```
load("~/est/country_data/v-dem-v9/v_dem-ra-1950_2018.RData") # loads vdem
vdem <- vdem %>%
#  filter(Democracy == "Deliberative") %>%
  mutate(Country = as.character(Country)) %>%
  mutate(Country = ifelse(Country == "Germany (RDA)", "Germany", Country)) %>%
  mutate(Country = ifelse(Country == "United States of America", "United States", Country)) %>%
  mutate(Country = ifelse(Country == "South Korea", "Korea, Republic of", Country)) %>%
  filter(Country %in% countries) %>%
  mutate(Country = as.factor(Country)) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Country, Year) %>%
  summarize(Democracy = mean(value, na.rm = TRUE)) #%>%
#  expand_grid(Sector = c("Environmental", "Social"))
```

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("wgi/wgi-full-v201029.RData")
gov.eff.original <- wgi %>%
  filter(indicator == "Government effectiveness") %>%
  select(country, year, value = Estimate) %>%
  expand_grid(Sector = c("Environmental", "Social"))
```

Add fake government effectiveness

```
range.ge.env <- range.ge.soc <- range(gov.eff.original$value, na.rm = TRUE)

gov.eff.original <- gov.eff.original %>%
  bind_rows(expand_grid(tibble(Sector = "Environmental",
                               country = paste0("Z-", sprintf("%02d", 2:11)),
                               value = seq(range.ge.env[1],
                                           range.ge.env[2],
                                           length.out = length(2:11))),
                        year = min(D$Year):max(D$Year)))

gov.eff.original <- gov.eff.original %>%
  bind_rows(expand_grid(tibble(Sector = "Social",
                               country = paste0("Z-", sprintf("%02d", 2:11)),
                               value = seq(range.ge.soc[1],
                                           range.ge.soc[2],
                                           length.out = length(2:11))),
                        year = min(D$Year):max(D$Year)))
```

Political Constraints - polconIII. An indicator of "veto players" comes from Henisz (2002). The indicator "estimates the feasibility of policy change. [That is], the extent to which a change in the preferences of any one actor may lead to a change in government policy". Higher values represent systems with higher constraints.

```
load("polcon/polcon2017.RData") # loads polcon
```

Add fake political constraints values.

```
polcon.d <- polcon %>%
  as_tibble() %>%
  filter(year %in% years[1]:years[2]) %>%
  mutate(country.polity = as.character(country.polity)) %>%
  mutate(country.polity = ifelse(country.polity == "Germany West", "Germany", country.polity)) %>%
  filter(country.polity %in% countries) %>%
  select(Country = country.polity, Year = year, polcon)

range.polcon <- polcon.d %>%
  select(polcon) %>%
  unlist(., use.names = FALSE) %>%
  range()

polcon.d <- polcon.d %>%
  bind_rows(expand_grid(tibble(Country = paste0("Z-", sprintf("%02d", 12:21)),
                               polcon = seq(range.polcon[1],
                                            range.polcon[2],
                                            length.out = length(12:21))),
                         Year = min(D$Year):max(D$Year)))
```

For the "Green parties", data comes from Volkens (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("manifesto/cpm-consensus.RData")
cpm$country <- as.character(cpm$country)
cpm$country[cpm$country=="Korea"] <- "South Korea"
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<="2005-12-31")

# 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, as.numeric(diff(c(as.Date("1976-01-01"), series$date))))
    v[is.nan(v)] <- 0
    wmsf[C, 1+F] <- v
  }
}
```

For the salience of each topic, we employ data from Volkens (2013), weighting the proportion of votes to each party and the importance that each party gives to environmental issues or the expansion of social welfare.

```
load("manifesto/201029-cpm-salience.RData")

salience <- cpm.salience %>%
```

```
  filter(Country %in% countries) %>%
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  filter(Year %in% years[1]:years[2])
```

Border contiguity

```
load("borders/geography.RData")

m.borders ← M.borders[dimnames(M.borders)[[1]] %in% countries,
                      dimnames(M.borders)[[2]] %in% countries]
```

Trade dependency

```
load("trade/trade.RData")

rm(M.trade, M.trade.imports)
```

Save and arrange for analysis.

```
diversity ← diversity %>%
  # Delete Turkey and Korea
  filter(!Country %in% c("South Korea", "Turkey")) %>%
  droplevels()

diversity.l ← diversity %>%
  gather(Measure, value, -c(Sector, Country, Year))
Y ← reshape2::acast(diversity.l,
                    Sector ~ Country ~ Year ~ Measure, value.var = "value")
nS ← dim(Y)[1]
nC ← dim(Y)[2]
nY ← dim(Y)[3]
nD ← dim(Y)[4]


country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
nC.fake ← 21
nC.real ← nC - nC.fake
id.real.countries ← 1:nC.real
id.fake.countries ← (nC.real + 1):nC

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)

year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(as.numeric(year.label))
nY ← length(year.label)

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)

diversity.label ← dimnames(Y)[[4]]
nD ← length(diversity.label)

nB ← 11
b0 ← rep(0, nB)
B0 ← diag(nB)
diag(B0) ← 1^-2


# 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("get-eu_time.R") # generates eu.ms

# Fake countries
f.c ← paste0("Z-", sprintf("%02d", 1:21))
# Fake years
f.y ← year.label.numeric

GDPpc ← wdi %>%
  select(country, year, gdp.capita) %>%
```

```
  filter(country %in% country.label) %>%
  mutate(gdp.capita = std(gdp.capita)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, 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")

trade.df <- wdi %>%
  select(country, year, trade) %>%
  filter(country %in% country.label) %>%
  mutate(trade = std(trade)) %>%
  bind_rows(expand_grid(country = f.c, year = f.y, trade = 0))
trade <- trade.df %>%
  reshape2::acast(country ~ year, value.var = "trade")
if ( length(which(!dimnames(trade)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

democracy <- 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(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  mutate(Democracy = std(Democracy)) %>%
  select(Country, Year, Democracy) %>%
  bind_rows(expand_grid(Country = f.c, Year = f.y, Democracy = 0)) %>%
  reshape2::acast(Country ~ Year, value.var = "Democracy")
if ( length(which(!dimnames(democracy)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

constraints.d <- expand_grid(Country = country.label,
                             Year = year.label.numeric) %>%
  left_join(polcon.d) %>%
  mutate(original.polcon = polcon) %>%
  mutate(polcon = std(polcon)) %>%
  mutate(polcon = ifelse(str_detect(Country, "^Z-") & is.na(polcon), 0, polcon))
constraints <- constraints.d %>%
  reshape2::acast(Country ~ Year, value.var = "polcon")
if ( length(which(!dimnames(constraints)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")

gov.eff.df <- expand.grid(
    country = country.label,
    year = year.label.numeric,
    Sector = sector.label) %>%
  left_join(gov.eff.original) %>%
  filter(country %in% country.label) %>%
  filter(year >= 1976 & year <= 2005) %>%
  mutate(value = std(value)) %>%
  select(Sector, country, year, value) %>%
  mutate(value = ifelse(str_detect(country, "^Z-") & is.na(value), 0, value))
gov.eff <- gov.eff.df %>%
  reshape2::acast(Sector ~ country ~ year, value.var = "value")
if ( length(which(!dimnames(gov.eff)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")

min.discard.zero <- function(x) return(min(x[x!=0]))
portfolio.size <-
  D %>%
  filter(Measure == "Size") %>%
  spread(Measure, value) %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector) %>%
  mutate(Size = ifelse(Size == 0, min.discard.zero(Size)/2, Size)) %>%
  mutate(Size = std(logit(Size))) %>%
  mutate(Size = ifelse(str_detect(Country, "Z2"), 0, Size)) %>%
  ungroup() %>%
  select(Country, Sector, Year, Size) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Size")
if ( length(which(!dimnames(portfolio.size)[[2]] == 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")

# Green socialist as salience
green.socialist <- salience %>%
  group_by(Sector) %>%
  mutate(Salience = std(Salience)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label, Country = f.c, Year = f.y, Salience = 0)) %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Salience")
if ( length(which(!dimnames(green.socialist)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here

# Borders in tidy data
interdependency.contiguity <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(geography %>%
              select(Origin, Destination, p.contiguous),
              by = c("Destination" = "Destination")) %>%
  mutate(wDiversity = value * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
  summarize(contiguity.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
  group_by(Sector, Measure) %>%
  mutate(contiguity.dependency = std(contiguity.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        contiguity.dependency = 0))
interdependency.contiguity <- interdependency.contiguity %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "contiguity.dependency")
if ( length(which(!dimnames(interdependency.contiguity)[[3]] == country.label)) > 0) stop("Ep! There is a m

# Trade in tidy data
interdependency.trade <-
  select(diversity.l, Sector, Destination = Country, Year, Measure, value) %>%
  left_join(trade.p %>%
              ungroup() %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  mutate(wDiversity = value * p.Exports) %>%
  mutate(Origin = as.character(Origin),
         Destination = as.character(Destination)) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Sector, Country, Year, Measure) %>%
  summarize(trade.dependency = sum(wDiversity, na.rm = TRUE)) %>%
  ungroup() %>%
  filter(Country %in% country.label) %>%
  filter(Year >= 1976 & Year <= 2005) %>%
```

```
  group_by(Sector, Measure) %>%
  mutate(trade.dependency = std(trade.dependency)) %>%
  ungroup() %>%
  bind_rows(expand_grid(Sector = sector.label,
                        Country = f.c,
                        Year = f.y,
                        Measure = diversity.label,
                        trade.dependency = 0))
interdependency.trade <- interdependency.trade %>%
  reshape2::acast(Measure ~ Sector ~ Country ~ Year, value.var = "trade.dependency")
if ( length(which(!dimnames(interdependency.trade)[[3]] == country.label)) > 0) stop("Ep! There is a mistak

# Performance part
# Match it with the general Y, but only for the environmental sector
d.perf.fake <- expand_grid(
  Country = country.label[str_detect(country.label, "^Z-") ],
  Year = year.label.numeric,
  Indicator = unique(d.perf$Indicator),
  Performance = NA)

Y.performance <- diversity.l %>%
  select(Sector, Country, Year, Measure) %>%
  left_join(d.perf) %>%
  left_join(d.perf.fake) %>%
  filter(Sector == "Environmental") %>%
  group_by(Indicator) %>%
  mutate(Performance = std1(Performance)) %>%
  ungroup() %>%
  reshape2::acast(Indicator ~ Country ~ Year ~ Measure, value.var = "Performance")
# manually get rid of ghost performance for missing values
Y.performance <- Y.performance[-4,,,]

nP <- dim(Y.performance)[1]
performance.label <- dimnames(Y.performance)[[1]]
if ( length(which(!dimnames(Y.performance)[[2]] == country.label)) > 0) stop("Ep! There is a mistake here")

load("wdi/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 %in% year.label.numeric) %>%
  mutate(gdp.growth = std(gdp.growth)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "gdp.growth")
gdp.growth <- gdp.growth[,-dim(gdp.growth)[2]]
if ( length(which(!dimnames(gdp.growth)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
gdp.growth.means <- apply(gdp.growth, 1, mean, na.rm = TRUE)
gdp.growth.means[is.nan(gdp.growth.means)] <- 0


load("wdi/wdi-urban.RData") # urban
urban <- urban %>%
  select(country, year, urban = SP.URB.TOTL.IN.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(urban = std(urban)) %>%
  right_join(tibble(country = country.label)) %>%
  reshape2::acast(country ~ year, value.var = "urban")
urban <- urban[,-dim(urban)[2]]
if ( length(which(!dimnames(urban)[[1]] == country.label)) > 0) stop("Ep! There is a mistake here")
urban.means <- apply(urban, 1, mean, na.rm = TRUE)
urban.means[is.nan(urban.means)] <- 0

load("wdi/wdi-industry.RData") # industry
industry <- industry %>%
  select(country, year, industry = NV.IND.TOTL.ZS) %>%
  filter(country %in% country.label) %>%
  filter(year %in% year.label.numeric) %>%
  mutate(industry = std(industry)) %>%
  right_join(tibble(country = country.label)) %>%
```

```
  reshape2::acast(country ~ year, value.var = "industry")
industry <- industry[,-dim(industry)[2]]
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)
industry.means[is.nan(industry.means)] <- 0


DP <- list(
  Y = unname(Y),
  Y.performance = unname(Y.performance), nP = nP,
  GDPpc = unname(GDPpc),
  trade = unname(trade),
  democracy = unname(democracy),
  constraints = unname(constraints),
  gov.eff = unname(gov.eff),
  gov.eff.mean.observed = unname(apply(gov.eff, c(1, 2), mean, na.rm = TRUE)),
  interdependency.contiguity = unname(interdependency.contiguity),
  interdependency.trade = unname(interdependency.trade),
  portfolio.size = unname(portfolio.size),
  eu = unname(eu),
  green.socialist = unname(green.socialist),
  gdp.growth = unname(gdp.growth), gdp.growth.means = unname(gdp.growth.means),
  urban = unname(urban), urban.means = unname(urban.means),
  industry = unname(industry), industry.means = unname(industry.means),
  nC = nC,
  id.fake.countries = id.fake.countries,
  id.real.countries = id.real.countries,
  id.decade = id.decade, nDecades = nDecades,
  nB = nB, b0 = b0, B0 = B0,
  nS = nS,
  nD = nD,
  nY = nY)
nC    # Number of countries (including fake ones)
```

```
## [1] 42
```

```
nS    # Number of sectors
```

```
## [1] 2
```

```
years   # Range of years
```

```
## [1] 1976 2005
```

8.3 Model

```
M <- "diversity-democracy"
M.lab <- "Diversity (AID), robustness, democracy"
m <- "model {
  #
  # Data part at the observational level
  #
  for (d in 1:nD) {
    for (s in 1:nS) {
      for (c in 1:nC) {
        for (t in 2:nY) {
          Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
          mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
                     + beta[1,s,d] * GDPpc[c,t-1]
                     + beta[3,s,d] * democracy[c,t-1]
                     + beta[4,s,d] * gov.eff[s,c,t-1]
                     + beta[5,s,d] * portfolio.size[s,c,t-1]
                     + beta[6,s,d] * trade[c,t-1]
                     + beta[7,s,d] * eu[c,t-1]
                     + beta[8,s,d] * green.socialist[s,c,t-1] # col 1 green, col 2 socialist
                     + beta[9,s,d] * constraints[c,t-1]
                     + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
```

```
                 + beta[11,s,d] * interdependency.trade[d,s,c,t]
                 + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
        tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
        sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
                                + lambda[2,s,d] * portfolio.size[s,c,t]
                                + gamma[c,d])
        resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
      }
      Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
      mu[s,c,1,d] <- alpha[s,d,1]
               + beta[1,s,d] * GDPpc[c,1]
               + beta[3,s,d] * democracy[c,1]
               + beta[4,s,d] * gov.eff[s,c,1]
               + beta[5,s,d] * portfolio.size[s,c,1]
               + beta[6,s,d] * trade[c,1]
               + beta[7,s,d] * eu[c,1]
               + beta[8,s,d] * green.socialist[s,c,1] # col 1 green, col 2 socialist
               + beta[9,s,d] * constraints[c,1]
                 + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
                 + beta[11,s,d] * interdependency.trade[d,s,c,1]

      resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
      tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
      sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
                              + lambda[2,s,d] * portfolio.size[s,c,1]
                              + gamma[c,d])
    }


    #
    # Degrees of freedom of GR
    #
    nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
    nu.trans[s,d] ~ dunif(0, 1)

    #
    # Priors for variance component
    #
    lambda[1,s,d] ~ dnorm(0, 2^-2)
    lambda[2,s,d] ~ dnorm(0, 2^-2)

    #
    # Priors for the intercept
    #
    for (decade in 1:nDecades) {
      alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
    }
    Alpha[s,d] ~ dunif(0, 1)
    tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
    sigma.alpha[s,d] ~  dunif(0, 0.5)

    #
    # Priors for the control variables
    #
    beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
    Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
    Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
  }
  #
  # Data part for  performance
  #
  for (p in 1:nP) {
    for (c in 1:nC) {
      for (t in 2:nY) {
        Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
        mu.performance[p,c,t,d] <- eta.performance[1,p,d]
                                  + eta.performance[2,p,d] * resid[1,c,t,d]
                                  + eta.performance[3,p,d] * Y[1,c,t-1,d]
                                  + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
                                  + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
                                  + eta.performance[6,p,d] * GDPpc[c,t-1]
                                  + eta.performance[7,p,d] * trade[c,t-1]
                                  + eta.performance[8,p,d] * eu[c,t-1]
                                  + eta.performance[9,p,d] * gdp.growth[c,t-1]
                                  + eta.performance[10,p,d] * urban[c,t-1]
                                  + eta.performance[11,p,d] * industry[c,t-1]
      }
```

```
      Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
      mu.performance[p,c,1,d] <- eta.performance[1,p,d]
                               + eta.performance[2,p,d] * resid[1,c,1,d]
                               + eta.performance[3,p,d] * Y[1,c,1,d]
                               + eta.performance[4,p,d] * portfolio.size[1,c,1]
                               + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
                               + eta.performance[6,p,d] * GDPpc[c,1]
                               + eta.performance[7,p,d] * trade[c,1]
                               + eta.performance[8,p,d] * eu[c,1]
                               + eta.performance[9,p,d] * gdp.growth[c,1]
                               + eta.performance[10,p,d] * urban[c,1]
                               + eta.performance[11,p,d] * industry[c,1]
    }
    tau.performance[p,d] ~ dgamma(0.001, 0.001)
    sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
    for (e in 1:11) {
      eta.performance[e,p,d] ~ dnorm(0, 1^-2)
    }
  }

  # Variance component, varying intercepts by country
  for (c in 1:nC) {
    gamma[c,d] ~ dnorm(0, 0.2^-2)
  }
  #
  # AR(1) parameters
  #
  for (c in id.real.countries) {
    for (s in 1:nS) {
      rho[s,c,d] ~ dunif(-1, 1)
    }
    for (p in 1:nP) {
      rho.performance[p,c,d] ~ dunif(-1, 1)
    }
  }
  for (c in id.fake.countries) {
    for (s in 1:nS) {
      rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
    }
    for (p in 1:nP) {
      rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
    }
  }
}

# SEM Part for portfolio size
for (s in 1:nS) {
  for (c in 1:nC) {
    for (t in 2:nY) {
      portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
      mu.ps[s,c,t] <- alpha.ps[s,c]
                    + delta[1,s] * GDPpc[c,t-1]
                    + delta[2,s] * gov.eff[s,c,t-1]
                    + delta[3,s] * green.socialist[s,c,t-1]
                    + delta[4,s] * constraints[c,t-1]
    }
    tau.ps[s,c] ~ dgamma(0.1, 0.1)
    sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
    alpha.ps[s,c] ~ dnorm(0, 1^-2)
  }
  for (d in 1:4) {
    delta[d,s] ~ dnorm(0, 1^-2)
  }
}
# Mediated effects
for (d in 1:nD) {
  for (s in 1:nS) {
    pi[1,s,d] <- delta[1,s] * beta[1,s,d]       # GDPpc
    pi[2,s,d] <- delta[2,s] * beta[4,s,d]       # gov.eff
    pi[3,s,d] <- delta[3,s] * beta[8,s,d]       # green
    pi[4,s,d] <- delta[4,s] * beta[9,s,d]       # constraints
  }
}
```

```
  #
  # Missing data
  #
  for (c in 1:nC) {
    for (t in 1:nY) {
      gov.eff[1,c,t] ~ dnorm(mean(gov.eff.mean.observed[1,c]), 0.05^-2)
      gov.eff[2,c,t] ~ dnorm(gov.eff[1,c,t], 100)
      gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
      urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
      industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
    }
  }
  for (s in 1:nS) {
    for (c in 1:nC) {
      # Reverse years back for NA in early years
      for (t in 1:(nY-1)) {
        green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
      }
      for (d in 1:nD) {
        for (t in 1:(nY-1)) {
          interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
        }
      }
    }
  }
}"
write(m, file= paste("models/model-", M, ".bug", sep = ""))
par ← NULL
par ← c(par, "alpha", "beta", "theta", "sigma")
par ← c(par, "Alpha", "sigma.alpha")
par ← c(par, "Sigma")
par ← c(par, "lambda", "gamma")
par ← c(par, "nu")
par ← c(par, "rho")
par.fake ← expand_grid(Sector = 1:nS,
                       Country = id.fake.countries,
                       Year = 2:nY,
                       Diversity = 1:nD) %>%
  mutate(parameter = paste0("mu[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par ← c(par, par.fake)
par.resid ← expand_grid(Sector = 1:nS,
                        Country = id.real.countries,
                        Year = 2:nY,
                        Diversity = 1:nD) %>%
  mutate(parameter = paste0("resid[", Sector, ",", Country, ",", Year, ",", Diversity, "]")) %>%
  select(parameter) %>%
  unlist(., use.names = FALSE)
par ← c(par, par.resid)
par ← c(par, "delta", "alpha.ps", "sigma.ps", "pi")
par ← c(par, "eta.performance", "sigma.performance")
inits ← list(
  list(.RNG.name="base::Super-Duper", .RNG.seed=10),
  list(.RNG.name="base::Super-Duper", .RNG.seed=20),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=30),
  list(.RNG.name="base::Wichmann-Hill", .RNG.seed=20))

t0 ← proc.time()
rj ← run.jags(model = paste("models/model-", M, ".bug", sep = ""),
  data = dump.format(DP, checkvalid=FALSE),
  inits = inits,
  modules = "glm",
  n.chains = 1, adapt = 2e2, burnin = 1e3, sample = 2e3, thin = 1,
  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 = ""))
```

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

```
## model {
##   #
##     #
##   for (d in 1:nD) {
##     for (s in 1:nS) {
##       for (c in 1:nC) {
##         for (t in 2:nY) {
##           Y[s,c,t,d] ~ dnorm(mu[s,c,t,d], tau[s,c,t,d])
##           mu[s,c,t,d] <- alpha[s,d,id.decade[t]]
##                     + beta[1,s,d] * GDPpc[c,t-1]
##                     + beta[3,s,d] * democracy[c,t-1]
##                     + beta[4,s,d] * gov.eff[s,c,t-1]
##                     + beta[5,s,d] * portfolio.size[s,c,t-1]
##                     + beta[6,s,d] * trade[c,t-1]
##                     + beta[7,s,d] * eu[c,t-1]
##                     + beta[8,s,d] * green.socialist[s,c,t-1]                    + beta[9,s,d] * constrain
##                     + beta[10,s,d] * interdependency.contiguity[d,s,c,t]
##                     + beta[11,s,d] * interdependency.trade[d,s,c,t]
##                     + rho[s,c,d] * (Y[s,c,t-1,d] - mu[s,c,t-1,d] )
##           tau[s,c,t,d] <- 1 / sigma.sq[s,c,t,d]
##           sigma.sq[s,c,t,d] <- exp(lambda[1,s,d]
##                                   + lambda[2,s,d] * portfolio.size[s,c,t]
##                                   + gamma[c,d])
##           resid[s,c,t,d] <- Y[s,c,t,d] - mu[s,c,t,d]
##         }
##         Y[s,c,1,d] ~ dnorm(mu[s,c,1,d], tau[s,c,1,d])
##         mu[s,c,1,d] <- alpha[s,d,1]
##                   + beta[1,s,d] * GDPpc[c,1]
##                   + beta[3,s,d] * democracy[c,1]
##                   + beta[4,s,d] * gov.eff[s,c,1]
##                   + beta[5,s,d] * portfolio.size[s,c,1]
##                   + beta[6,s,d] * trade[c,1]
##                   + beta[7,s,d] * eu[c,1]
##                   + beta[8,s,d] * green.socialist[s,c,1]                  + beta[9,s,d] * constraints[c,1
##                     + beta[10,s,d] * interdependency.contiguity[d,s,c,1]
##                     + beta[11,s,d] * interdependency.trade[d,s,c,1]
##
##         resid[s,c,1,d] <- Y[s,c,1,d] - mu[s,c,1,d]
##         tau[s,c,1,d] <- 1 / sigma.sq[s,c,1,d]
##         sigma.sq[s,c,1,d] <- exp(lambda[1,s,d]
##                                 + lambda[2,s,d] * portfolio.size[s,c,1]
##                                 + gamma[c,d])
##       }
##
##
##       #
##             #
```

```
##       nu[s,d] <- 1 + (-1 * log(nu.trans[s,d]))
##       nu.trans[s,d] ~ dunif(0, 1)
##
##       #
##             #
##       lambda[1,s,d] ~ dnorm(0, 2^-2)
##       lambda[2,s,d] ~ dnorm(0, 2^-2)
##
##       #
##             #
##       for (decade in 1:nDecades) {
##         alpha[s,d,decade] ~ dnorm(Alpha[s,d], tau.alpha[s,d])
##       }
##       Alpha[s,d] ~ dunif(0, 1)
##       tau.alpha[s,d] <- 1 / sqrt(sigma.alpha[s,d])
##       sigma.alpha[s,d] ~  dunif(0, 0.5)
##
##       #
##             #
##       beta[1:nB,s,d] ~ dmnorm(b0, Omega[1:nB,1:nB,s,d])
##       Omega[1:nB,1:nB,s,d] ~ dwish(B0, nB + 1)
##       Sigma[1:nB,1:nB,s,d] <- inverse(Omega[1:nB,1:nB,s,d])
##     }
##     #
##         #
##     for (p in 1:nP) {
##       for (c in 1:nC) {
##         for (t in 2:nY) {
##           Y.performance[p,c,t,d] ~ dnorm(mu.performance[p,c,t,d], tau.performance[p,d])
##           mu.performance[p,c,t,d] <- eta.performance[1,p,d]
##                                    + eta.performance[2,p,d] * resid[1,c,t,d]
##                                    + eta.performance[3,p,d] * Y[1,c,t-1,d]
##                                    + eta.performance[4,p,d] * portfolio.size[1,c,t-1]
##                                    + eta.performance[5,p,d] * Y[1,c,t-1,d] * portfolio.size[1,c,t-1]
##                                    + eta.performance[6,p,d] * GDPpc[c,t-1]
##                                    + eta.performance[7,p,d] * trade[c,t-1]
##                                    + eta.performance[8,p,d] * eu[c,t-1]
##                                    + eta.performance[9,p,d] * gdp.growth[c,t-1]
##                                    + eta.performance[10,p,d] * urban[c,t-1]
##                                    + eta.performance[11,p,d] * industry[c,t-1]
##         }
##         Y.performance[p,c,1,d] ~ dnorm(mu.performance[p,c,1,d], tau.performance[p,d])
##         mu.performance[p,c,1,d] <- eta.performance[1,p,d]
##                                  + eta.performance[2,p,d] * resid[1,c,1,d]
##                                  + eta.performance[3,p,d] * Y[1,c,1,d]
##                                  + eta.performance[4,p,d] * portfolio.size[1,c,1]
##                                  + eta.performance[5,p,d] * Y[1,c,1,d] * portfolio.size[1,c,1]
##                                  + eta.performance[6,p,d] * GDPpc[c,1]
##                                  + eta.performance[7,p,d] * trade[c,1]
```

```
##                                  + eta.performance[8,p,d] * eu[c,1]
##                                  + eta.performance[9,p,d] * gdp.growth[c,1]
##                                  + eta.performance[10,p,d] * urban[c,1]
##                                  + eta.performance[11,p,d] * industry[c,1]
##       }
##       tau.performance[p,d] ~ dgamma(0.001, 0.001)
##       sigma.performance[p,d] <- 1 / sqrt(tau.performance[p,d])
##       for (e in 1:11) {
##         eta.performance[e,p,d] ~ dnorm(0, 1^-2)
##       }
##     }
##
##         for (c in 1:nC) {
##       gamma[c,d] ~ dnorm(0, 0.2^-2)
##     }
##     #
##         #
##     for (c in id.real.countries) {
##       for (s in 1:nS) {
##         rho[s,c,d] ~ dunif(-1, 1)
##       }
##       for (p in 1:nP) {
##         rho.performance[p,c,d] ~ dunif(-1, 1)
##       }
##     }
##     for (c in id.fake.countries) {
##       for (s in 1:nS) {
##         rho[s,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##       }
##       for (p in 1:nP) {
##         rho.performance[p,c,d] ~ dnorm(0.75, 0.2^-2)T(-1, 1)
##       }
##     }
##   }
##
##     for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (t in 2:nY) {
##         portfolio.size[s,c,t] ~ dnorm(mu.ps[s,c,t], tau.ps[s,c])
##         mu.ps[s,c,t] <- alpha.ps[s,c]
##                       + delta[1,s] * GDPpc[c,t-1]
##                       + delta[2,s] * gov.eff[s,c,t-1]
##                       + delta[3,s] * green.socialist[s,c,t-1]
##                       + delta[4,s] * constraints[c,t-1]
##       }
##       tau.ps[s,c] ~ dgamma(0.1, 0.1)
##       sigma.ps[s,c] <- 1 / sqrt(tau.ps[s,c])
##       alpha.ps[s,c] ~ dnorm(0, 1^-2)
##     }
```

```
##     for (d in 1:4) {
##       delta[d,s] ~ dnorm(0, 1^-2)
##     }
##   }
##     for (d in 1:nD) {
##     for (s in 1:nS) {
##       pi[1,s,d] <- delta[1,s] * beta[1,s,d]             pi[2,s,d] <- delta[2,s] * beta[4,s,d]
##   }
##
##
##
##   #
##     #
##   for (c in 1:nC) {
##     for (t in 1:nY) {
##       gov.eff[1,c,t] ~ dnorm(mean(gov.eff.mean.observed[1,c]), 0.05^-2)
##       gov.eff[2,c,t] ~ dnorm(gov.eff[1,c,t], 100)
##       gdp.growth[c,t] ~ dnorm(gdp.growth.means[c], 0.05^-2)
##       urban[c,t] ~ dnorm(urban.means[c], 0.05^-2)
##       industry[c,t] ~ dnorm(industry.means[c], 0.05^-2)
##     }
##   }
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##             for (t in 1:(nY-1)) {
##         green.socialist[s,c,t] ~ dnorm(green.socialist[s,c,t+1], 0.05^-2)
##       }
##       for (d in 1:nD) {
##         for (t in 1:(nY-1)) {
##           interdependency.trade[d,s,c,t] ~ dnorm(interdependency.trade[d,s,c,t+1], 0.05^-2)
##         }
##       }
##     }
##   }
## }
ggmcmc(ggs(s, family = "sigma|alpha|beta|lambda|rho|nu"),
       param_page = 10, file = paste("ggmcmc-full-", M, ".pdf", sep = ""))
```

8.4 Model results

Variance components.

```
L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             "Portfolio size",
                                             "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda, label = "Variable", comparison = "Sector") +
  aes(color = Sector) +
  facet_wrap(~ Diversity) +
  theme(legend.position = "right") +
  ggtitle("Variance component")
```

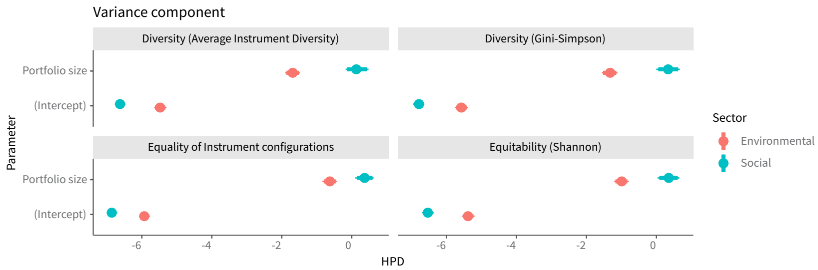

Figure 8.1: Variance component.

```
rm(S.lambda)

L.lambda <- plab("lambda", list(Variable = c("(Intercept)",
                                             "Portfolio size",
                                             "Constraints"),
                                Sector = sector.label,
                                Diversity = diversity.label))
S.lambda <- ggs(s, family = "lambda\\[", par_labels = L.lambda)
S.lambda %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable") +
  ggtitle("Variance component")
```

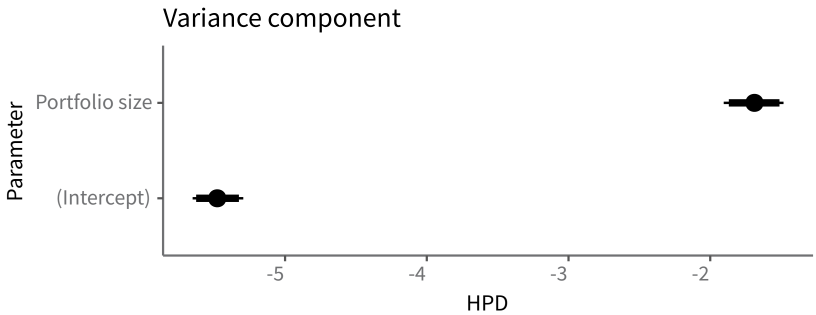

Figure 8.2: Variance component.

```
rm(S.lambda)
```

Variance components (country varying intercepts).

```
L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.gamma, label = "Country") +
  facet_grid(~ Diversity) +
  ggtitle("Variance component (countries)")

rm(S.gamma)

L.gamma <- plab("gamma", list(Country = country.label,
                              Diversity = diversity.label))
S.gamma <- ggs(s, family = "gamma\\[", par_labels = L.gamma)
S.gamma <- S.gamma %>%
  filter(!str_detect(Country, "^Z-")) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)")
ggs_caterpillar(S.gamma, label = "Country") +
  ggtitle("Variance component (countries)")
```

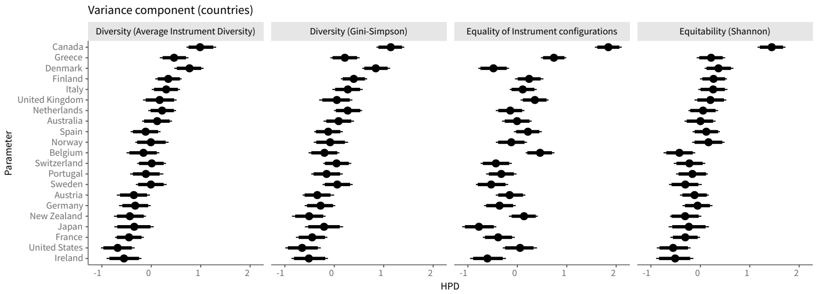

Figure 8.3: Variance component (country varying intercepts).

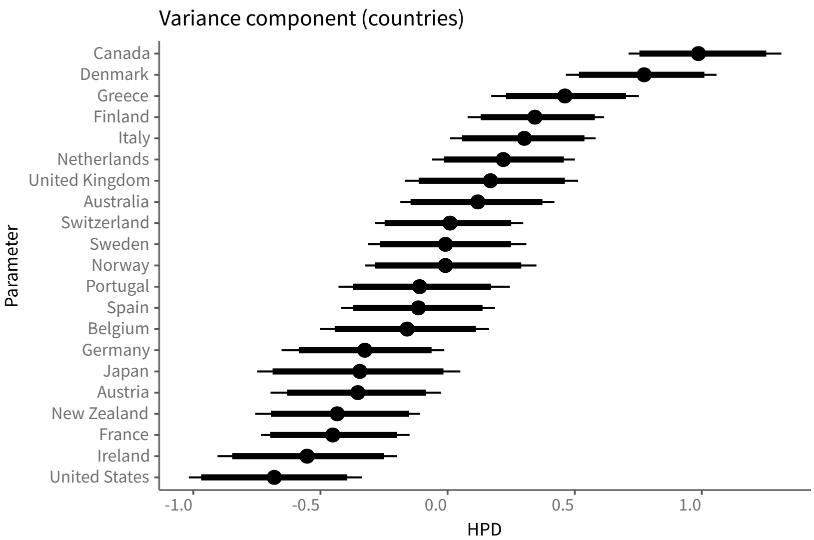

Figure 8.4: Variance component (country varying intercepts).

```
rm(S.gamma)
```

Auto-regressive components.

```
L.rho ← plab("rho", list(Sector = sector.label,
                         Country = country.label,
                         Diversity = diversity.label))
S.rho ← ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  facet_grid(Diversity ~ Sector, scales="free") +
  aes(color=Sector) +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()
```

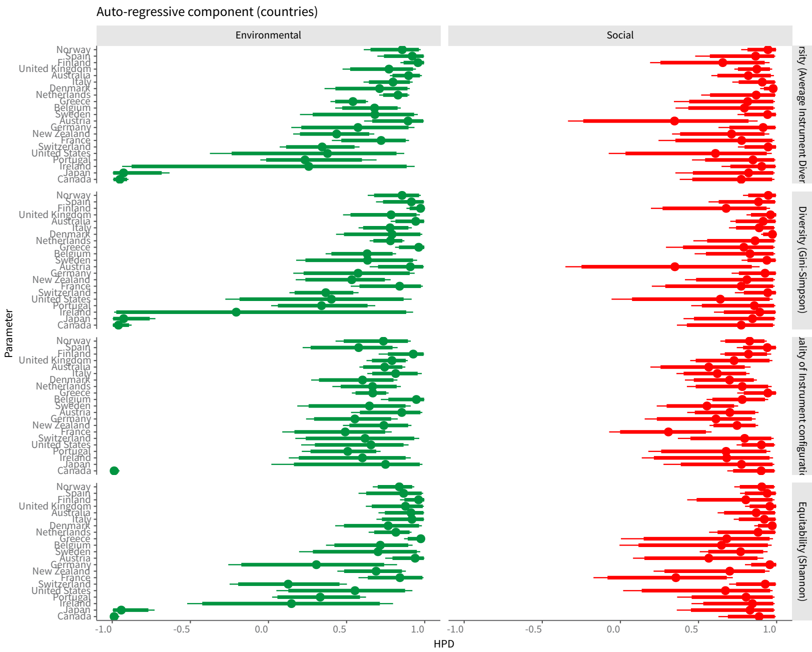

Figure 8.5: Auto-regressive component.

```
rm(S.rho)

L.rho ← plab("rho", list(Sector = sector.label,
                         Country = country.label,
                         Diversity = diversity.label))
S.rho ← ggs(s, family = "rho", par_labels = L.rho) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.rho, label = "Country") +
  expand_limits(x = c(-1, 1)) +
  ggtitle("Auto-regressive component (countries)") +
  scale_colour_xfim()


rm(S.rho)
```

Intercepts

```
L.alpha ← plab("alpha", list(Sector = sector.label,
                             Diversity = diversity.label,
                             Decade = decade.label))
```

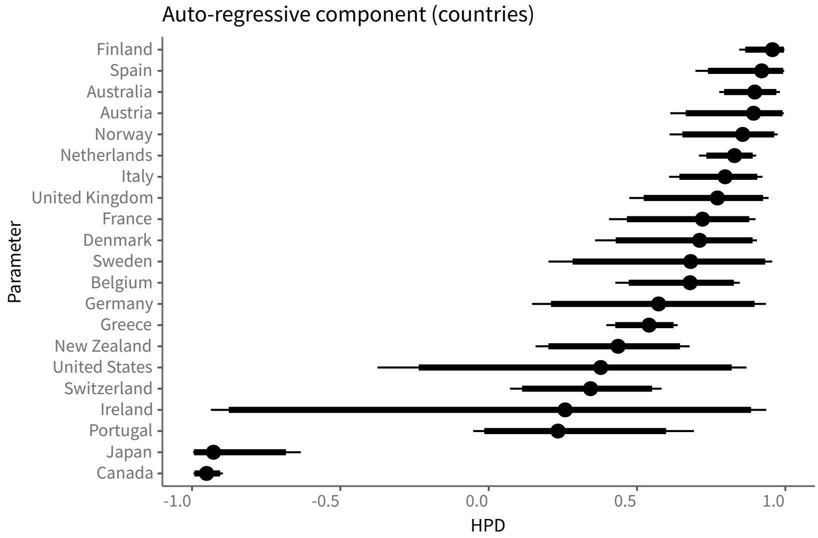

Figure 8.6: Auto-regressive component.

```
S.alpha <- ggs(s, family = "^alpha\\[", par_label = L.alpha)

ci(S.alpha) %>%
  ggplot(aes(x = Decade, y = median, ymin = Low, ymax = High)) +
  geom_point() +
  geom_linerange() +
  facet_grid(Diversity ~ Sector) +
  ggtitle("Temporal trend")


covariates.label <- c("GDP pc",
                                        "Consensus",
                                        "Democracy",
                                        "Government effectiveness",
                                        "Portfolio size",
                                        "Trade", "EU", "Salience",
                                        "Political constraints",
                                        "Interdependency (Contiguity)",
                                        "Interdependency (Trade)")

L.betas <- plab("beta", list(Variable = covariates.label,
                             Sector = sector.label,
                        Diversity = diversity.label))

S.betas <- ggs(s, family = "^beta\\[", par_labels = L.betas) %>%
  filter(!Variable %in% c("Deliberation", "Consensus"))
ci.betas <- ci(S.betas)
save(ci.betas, file = paste0("ci-betas-", M, ".RData"))

S.betas %>%
  group_by(Variable, Sector) %>%
  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()


ci(S.betas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
```

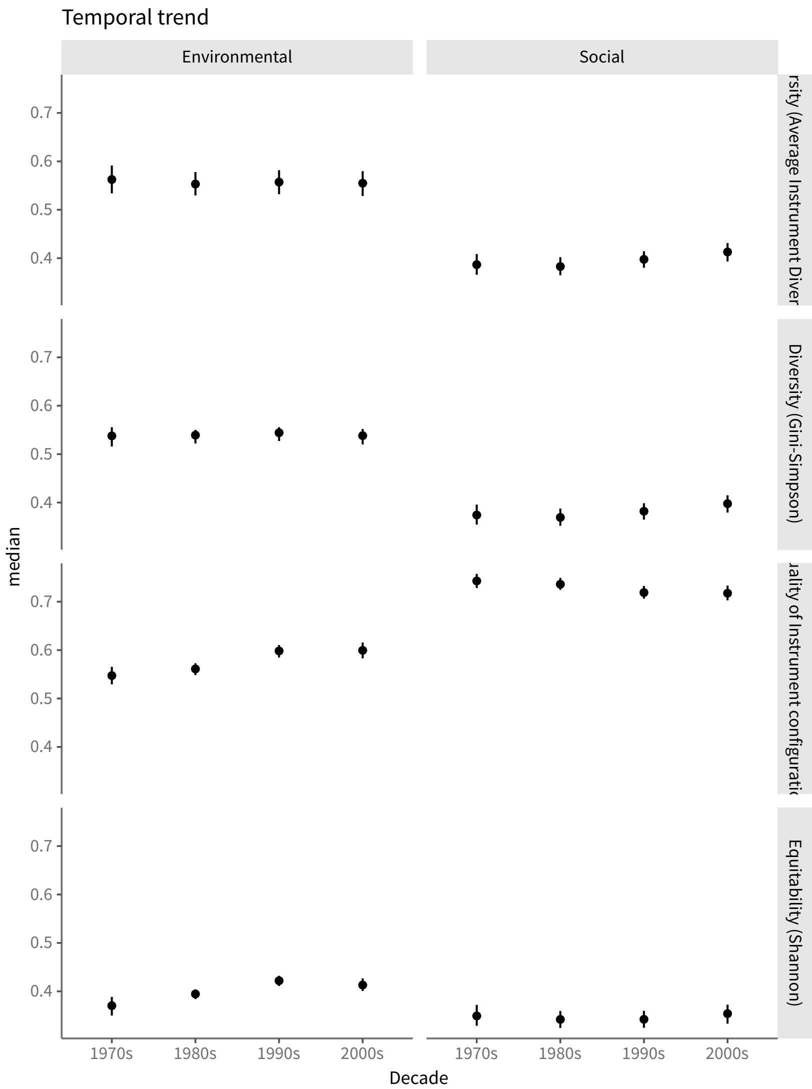

Figure 8.7: Intercepts.

Variable	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	-0.04	0.02	0.00	1.00	-0.04
GDP pc	Social	-0.08	0.05	0.25	0.75	-0.06
Democracy	Environmental	-0.01	0.01	0.09	0.91	-0.01
Democracy	Social	0.01	0.01	0.99	0.01	0.02
Government effectiveness	Environmental	0.41	0.11	1.00	0.00	0.36
Government effectiveness	Social	0.43	0.10	1.00	0.00	0.39
Portfolio size	Environmental	0.15	0.03	1.00	0.00	0.16
Portfolio size	Social	0.13	0.02	1.00	0.00	0.14
Trade	Environmental	-0.01	0.01	0.19	0.81	-0.01
Trade	Social	-0.06	0.01	0.00	1.00	-0.06
EU	Environmental	0.00	0.01	0.40	0.60	0.00
EU	Social	0.02	0.01	0.98	0.02	0.02
Salience	Environmental	-0.02	0.01	0.00	1.00	-0.02
Salience	Social	-0.02	0.01	0.09	0.91	-0.02
Political constraints	Environmental	-0.06	0.02	0.06	0.94	-0.05
Political constraints	Social	-0.13	0.03	0.00	1.00	-0.11
Interdependency (Contiguity)	Environmental	0.06	0.02	1.00	0.00	0.06
Interdependency (Contiguity)	Social	-0.02	0.02	0.25	0.75	-0.02
Interdependency (Trade)	Environmental	0.02	0.03	0.72	0.28	0.00
Interdependency (Trade)	Social	0.00	0.01	0.39	0.61	0.00

```
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

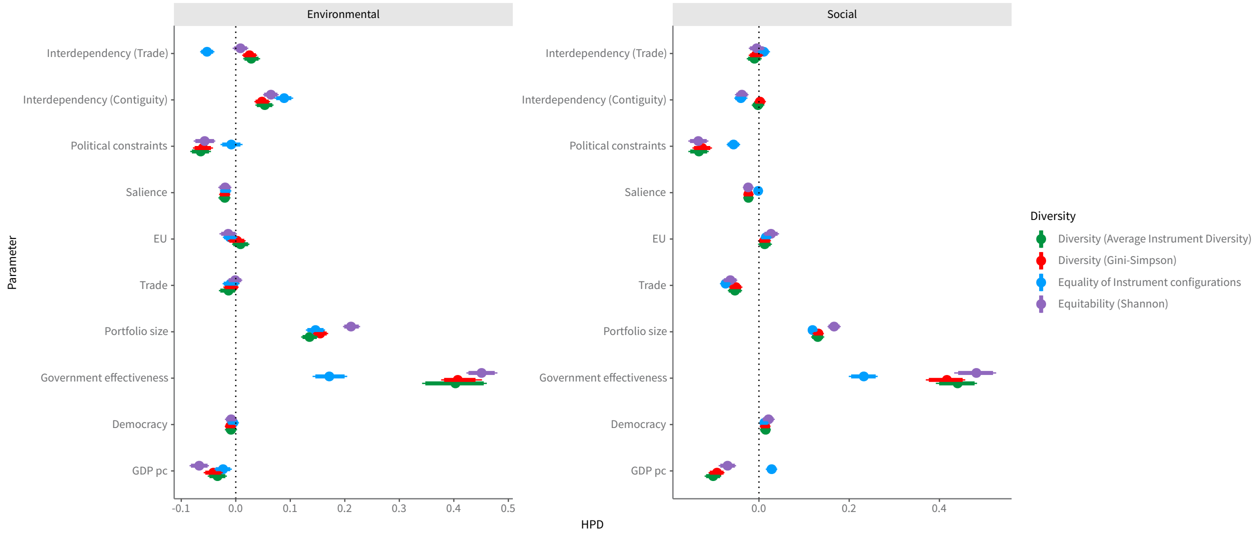

Figure 8.8: Slopes with the effects on diversity, by sector.

```
ci(S.betas) %>%
  filter(Sector == "Environmental") %>%
ggplot(aes(x = Variable,
              y = median,
              group = Diversity,
              color = Diversity)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()

ci(S.betas) %>%
ggplot(aes(x = Variable,
              y = median,
              group = Sector, color = Sector)) +
  coord_flip() +
  facet_wrap(~ Diversity, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)

ci(S.betas) %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggplot(aes(x = Variable,
              y = median,
              group = Sector, color = Sector)) +
  coord_flip() +
  geom_point(size = 3, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.2)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.2)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)
```

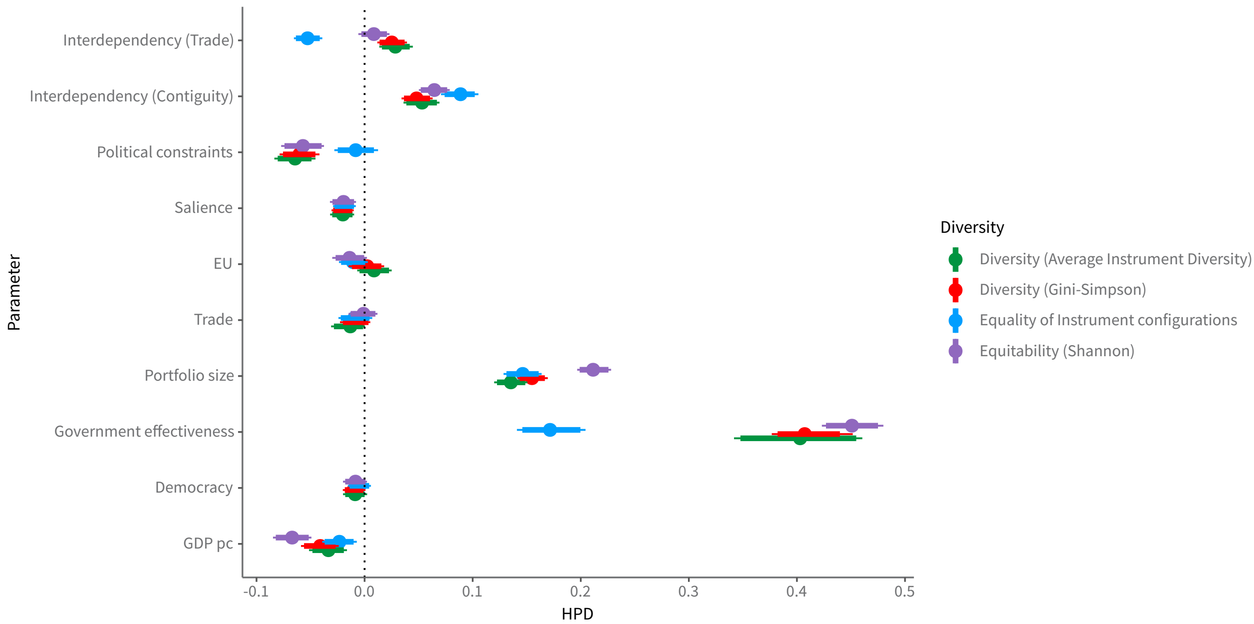

Figure 8.9: Slopes with the effects on diversity, for the environmental sector.

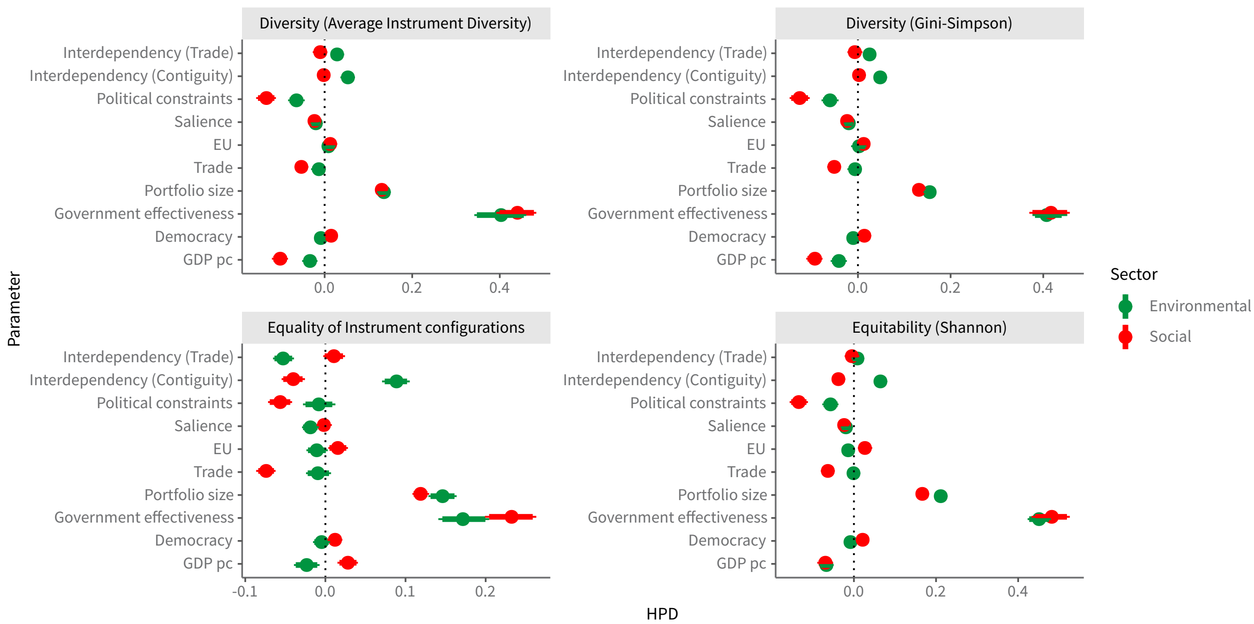

Figure 8.10: Slopes with the effects on diversity, by diversity indicator.

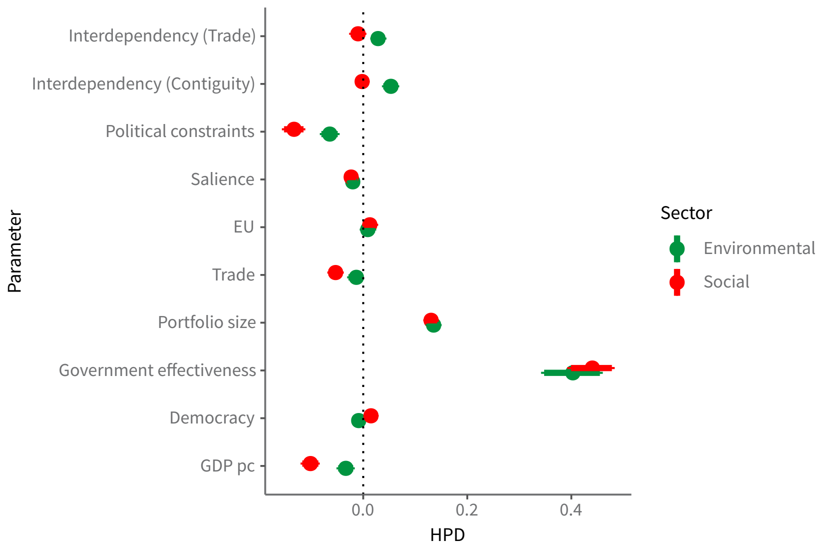

Figure 8.11: Slopes with the effects on AID diversity, by sector

```
ggplot(S.betas, aes(x = value, color = Sector, fill = Sector)) +
  geom_density(alpha = 0.5) +
  facet_grid(Diversity ~ Variable, scales = "free") +
  xlab("HPD") +
  geom_vline(xintercept = 0, lty = 3)
```

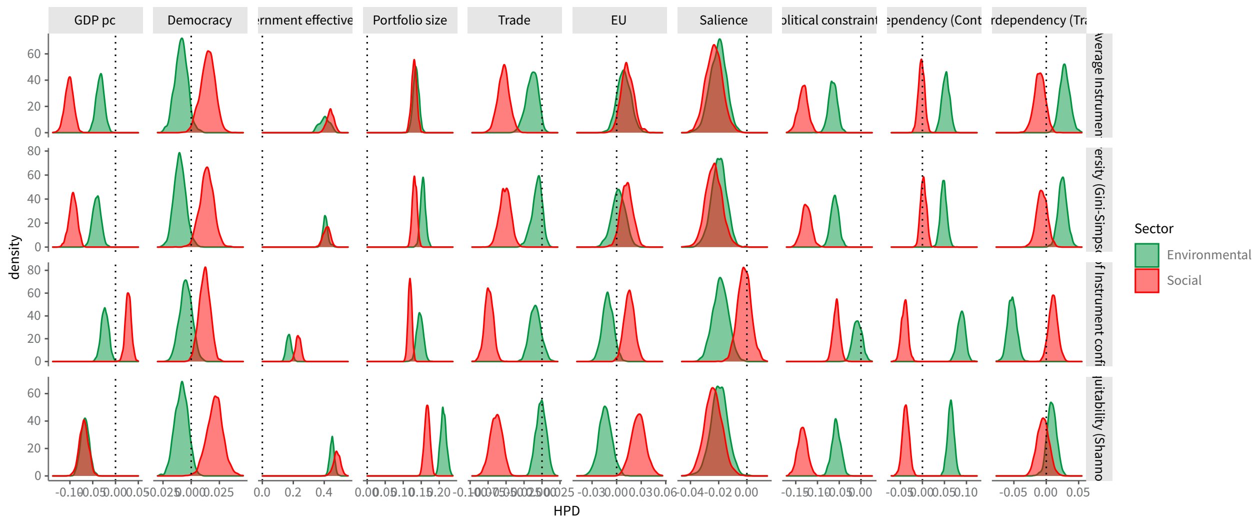

Figure 8.12: Slopes with the effects on diversity, comparing sectors.

Variables by evidence.

```
S.betas %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(value != 0) %>%
  group_by(Sector, Diversity, Variable) %>%
  summarize(`Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
    `Mean expected effect` = mean(value)) %>%
  group_by(Sector, Diversity, Variable) %>%
  mutate(max = max(abs(`Prob > 0`), abs(`Prob < 0`))) %>%
  arrange(desc(max)) %>%
  select(-max, -Diversity) %>%
  kable()
```

Diversity	Sector	Variable	Prob > 0	Prob < 0	Mean expected effect
Diversity (Average Instrument Diversity)	Environmental	GDP pc	0.00	1.00	-0.03
Diversity (Average Instrument Diversity)	Environmental	Government effectiveness	1.00	0.00	0.40
Diversity (Average Instrument Diversity)	Environmental	Portfolio size	1.00	0.00	0.14
Diversity (Average Instrument Diversity)	Environmental	Salience	0.00	1.00	-0.02
Diversity (Average Instrument Diversity)	Environmental	Political constraints	0.00	1.00	-0.06
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Contiguity)	1.00	0.00	0.05
Diversity (Average Instrument Diversity)	Environmental	Interdependency (Trade)	1.00	0.00	0.03
Diversity (Average Instrument Diversity)	Social	GDP pc	0.00	1.00	-0.10
Diversity (Average Instrument Diversity)	Social	Government effectiveness	1.00	0.00	0.44
Diversity (Average Instrument Diversity)	Social	Portfolio size	1.00	0.00	0.13
Diversity (Average Instrument Diversity)	Social	Trade	0.00	1.00	-0.05
Diversity (Average Instrument Diversity)	Social	Salience	0.00	1.00	-0.02
Diversity (Average Instrument Diversity)	Social	Political constraints	0.00	1.00	-0.13
Diversity (Average Instrument Diversity)	Social	Democracy	0.99	0.01	0.01
Diversity (Average Instrument Diversity)	Environmental	Trade	0.03	0.97	-0.01
Diversity (Average Instrument Diversity)	Social	EU	0.96	0.04	0.01
Diversity (Average Instrument Diversity)	Environmental	Democracy	0.05	0.95	-0.01
Diversity (Average Instrument Diversity)	Social	Interdependency (Trade)	0.12	0.88	-0.01
Diversity (Average Instrument Diversity)	Environmental	EU	0.86	0.14	0.01
Diversity (Average Instrument Diversity)	Social	Interdependency (Contiguity)	0.38	0.62	0.00

```
S.betas.env <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) #%>%

ggs_caterpillar(S.betas.env) +
  geom_vline(xintercept = 0, lty = 3)
```

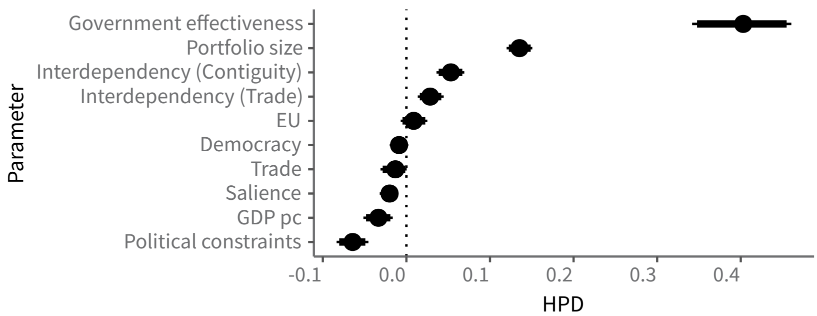

Figure 8.13: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
rm(S.betas.env)

S.betas.env.sorted <- S.betas %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  mutate(Parameter = as.character(Variable)) %>%
  mutate(Parameter = fct_relevel(Parameter, rev(c("Political constraints",
                                                  "Government effectiveness",
                                                  "Democracy",
                                   "Salience",
                                   "GDP pc", "Trade", "EU",
                                   "Interdependency (Contiguity)",
                                   "Interdependency (Trade)",
                                   "Portfolio size"))))

ggs_caterpillar(S.betas.env.sorted, sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)

rm(S.betas.env.sorted)

ci.betas <- ci(S.betas) %>%
  mutate(Model = M.lab)
```

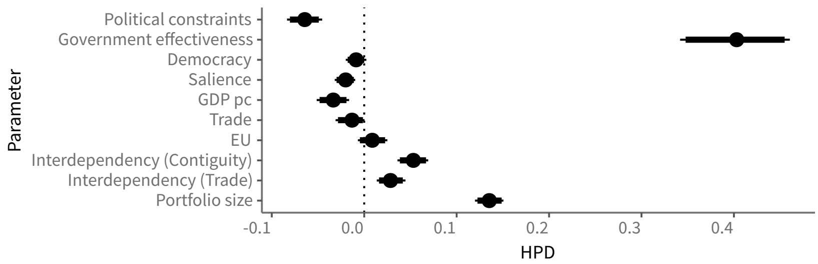

Figure 8.14: Slopes with the effects on portfolio diversity (AID). Environmental sector.

```
save(ci.betas, file = paste0("ci_betas-", M, ".RData"))
rm(S.betas, ci.betas)

L.deltas ← plab("delta", list(Variable = c("GDP pc",
                                            "Government effectiveness",
                                            "Salience",
                                            "Political constraints"),
                              Sector = sector.label))

S.deltas ← ggs(s, family = "^delta\\[", par_labels = L.deltas)

S.deltas %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
GDP pc	Environmental	1.42	0.07	1.00	0.00	1.42
GDP pc	Social	0.45	0.04	1.00	0.00	0.45
Government effectiveness	Environmental	0.01	0.26	0.51	0.49	-0.02
Government effectiveness	Social	-0.26	0.09	0.00	1.00	-0.25
Salience	Environmental	0.13	0.03	1.00	0.00	0.13
Salience	Social	-0.04	0.02	0.02	0.98	-0.04
Political constraints	Environmental	0.14	0.06	0.99	0.01	0.14
Political constraints	Social	0.24	0.06	1.00	0.00	0.24

```
ci(S.deltas) %>%
ggplot(aes(x = Variable,
           y = median,
           group = Sector)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3)


rm(S.deltas)

L.pi ← plab("pi", list(Variable = c("GDP pc",
                                     "Government effectiveness",
                                     "Salience",
                                     "Political constraints"),
                        Sector = sector.label,
                        Diversity = diversity.label))

S.pis ← ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable ≠ "GDP pc")
```

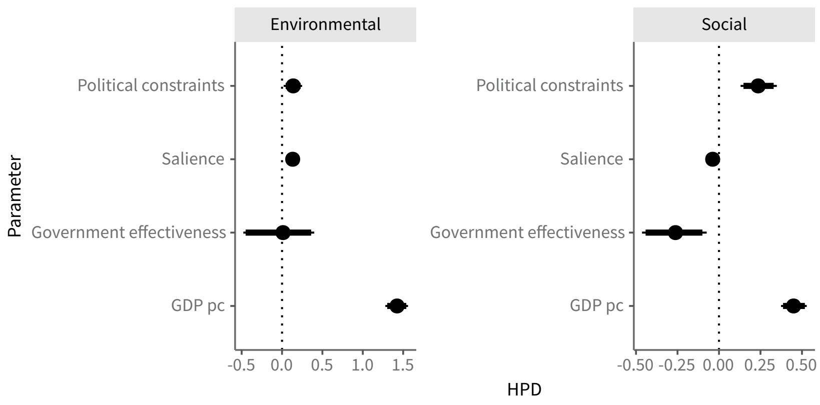

Figure 8.15: Slopes with the direct effects on portfolio size, by sector.

```
S.pis %>%
  group_by(Variable, Sector) %>%
  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	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Government effectiveness	Environmental	0.00	0.09	0.51	0.49	0.00
Government effectiveness	Social	-0.10	0.04	0.00	1.00	-0.10
Salience	Environmental	0.00	0.00	0.00	1.00	0.00
Salience	Social	0.00	0.00	0.89	0.11	0.00
Political constraints	Environmental	-0.01	0.00	0.07	0.93	-0.01
Political constraints	Social	-0.03	0.01	0.00	1.00	-0.03

```
ci(S.pis) %>%
ggplot(aes(x = Variable,
              y = median,
              group = interaction(Sector, Diversity),
              color = Diversity)) +
  coord_flip() +
  facet_wrap(~ Sector, scales="free") +
  geom_point(size = 3, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1.5, position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = low, ymax = high), size = 0.5, position = position_dodge(width = 0.3)) +
  ylab("HPD") + xlab("Parameter") +
  geom_hline(yintercept = 0, lty = 3) +
  scale_colour_xfim()
```

```
L.pi <- plab("pi", list(Variable = c("GDP pc",
                                     "Government effectiveness",
                                     "Salience",
                                     "Political constraints"),
                        Sector = sector.label,
                        Diversity = diversity.label))

S.pis <- ggs(s, family = "^pi\\[", par_labels = L.pi) %>%
  filter(Variable != "GDP pc")

S.pis %>%
  group_by(Variable, Sector) %>%
  summarize(median = median(value), sd = sd(value),
            `Prob > 0` = length(which(value > 0)) / n(),
    `Prob < 0` = length(which(value < 0)) / n(),
```

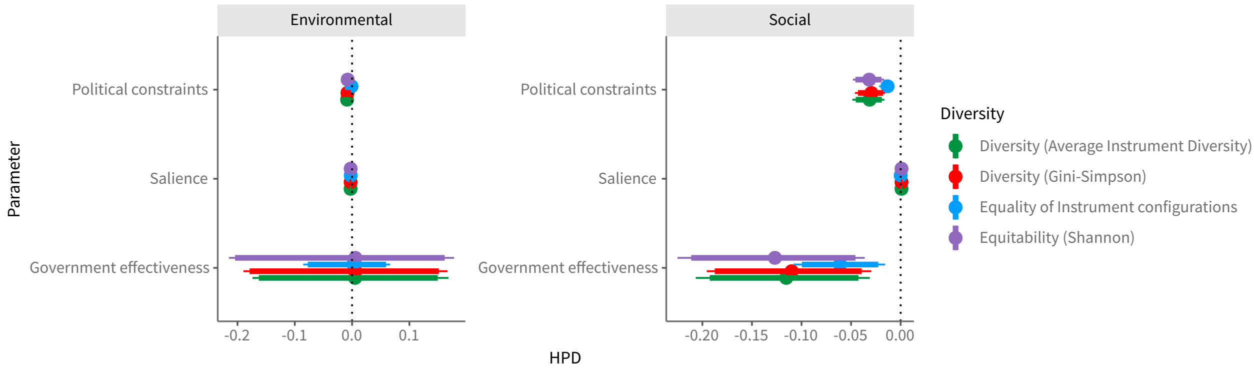

Figure 8.16: Mediated effects on diversity, by sector.

```
  `Mean expected effect` = mean(value)) %>%
  kable()
```

Variable	Sector	median	sd	Prob > 0	Prob < 0	Mean expected effect
Government effectiveness	Environmental	0.00	0.09	0.51	0.49	0.00
Government effectiveness	Social	-0.10	0.04	0.00	1.00	-0.10
Salience	Environmental	0.00	0.00	0.00	1.00	0.00
Salience	Social	0.00	0.00	0.89	0.11	0.00
Political constraints	Environmental	-0.01	0.00	0.07	0.93	-0.01
Political constraints	Social	-0.03	0.01	0.00	1.00	-0.03

```
S.pis %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggs_caterpillar(label = "Variable")
```

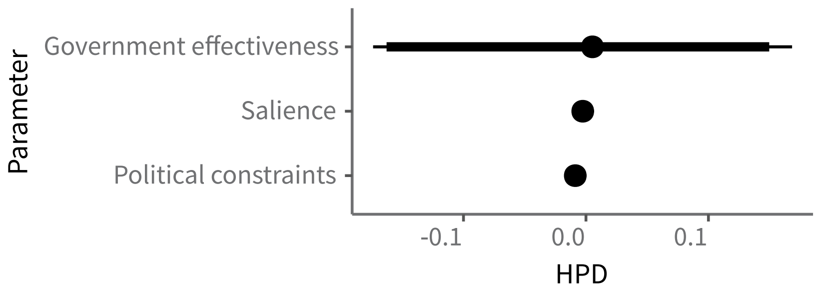

Figure 8.17: Mediated effects on diversity, for AID and the environmental sector.

```
rm(S.pis)

L.eta <- plab("eta.performance", list(Covariate = c("(Intercept)", "Residuals",
                                        "Diversity", "Portfolio size",
                                        "Diversity * Portfolio size",
                                        "GDPpc", "Trade", "EU",
                                        "GDP growth", "Urban", "Industry"),
                                     Performance = performance.label,
                                     Diversity = diversity.label))
S.eta <- ggs(s, family = "eta.performance", par_label = L.eta)

ggs_caterpillar(S.eta, label = "Covariate", comparison = "Performance") +
  geom_hline(yintercept = 0, lty = 3) +
  aes(color = Performance) +
  theme(legend.position = "right") +
  facet_wrap(~ Diversity)
```

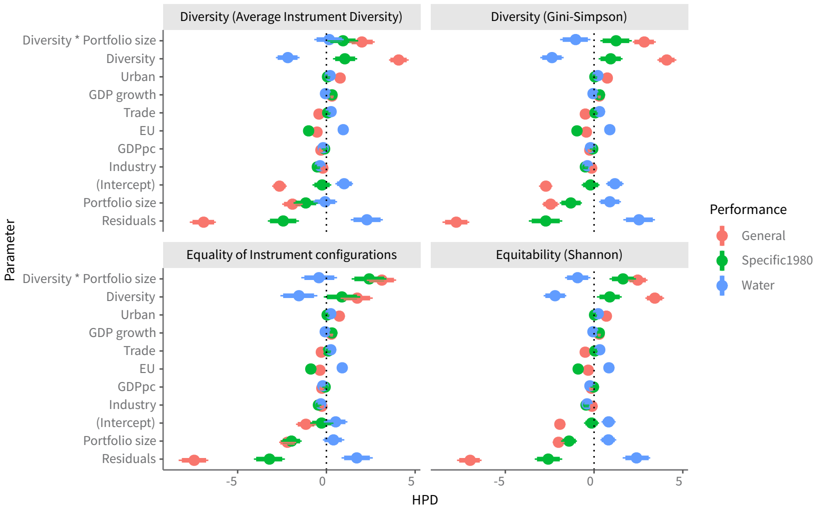

Figure 8.18: Direct effects on Environmental performance, by performance indicator and Diversity measure.

```
S.eta %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Performance == "General") %>%
ggs_caterpillar(label = "Covariate") +
  geom_vline(xintercept = 0, lty = 3)
```

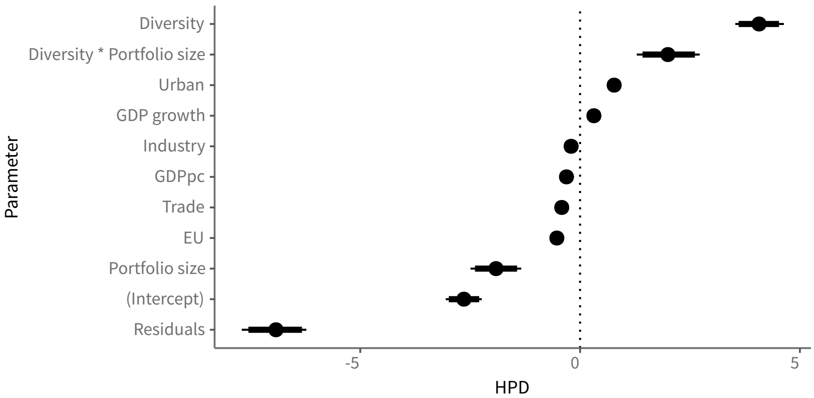

Figure 8.19: Direct effects on Environmental performance, for general performance and using AID as diversity.

```
rm(S.eta)
```

8.5 V-Cov

```
L.Sigma <- plab("Sigma", list(Covariate.1 = covariates.label,
                              Covariate.2 = covariates.label,
                              Sector = sector.label,
                              Diversity = diversity.label))
S.Sigma <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma) %>%
  filter(!Covariate.1 %in% c("Deliberation", "Consensus")) %>%
  filter(!Covariate.2 %in% c("Deliberation", "Consensus"))

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

ggplot(vcov.Sigma, aes(x = Covariate.2, y = Covariate.1, fill = vcov)) +
  geom_raster() +
```

```
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  facet_grid(Sector ~ Diversity) +
  scale_fill_continuous_diverging(palette = "Blue-Red")
```

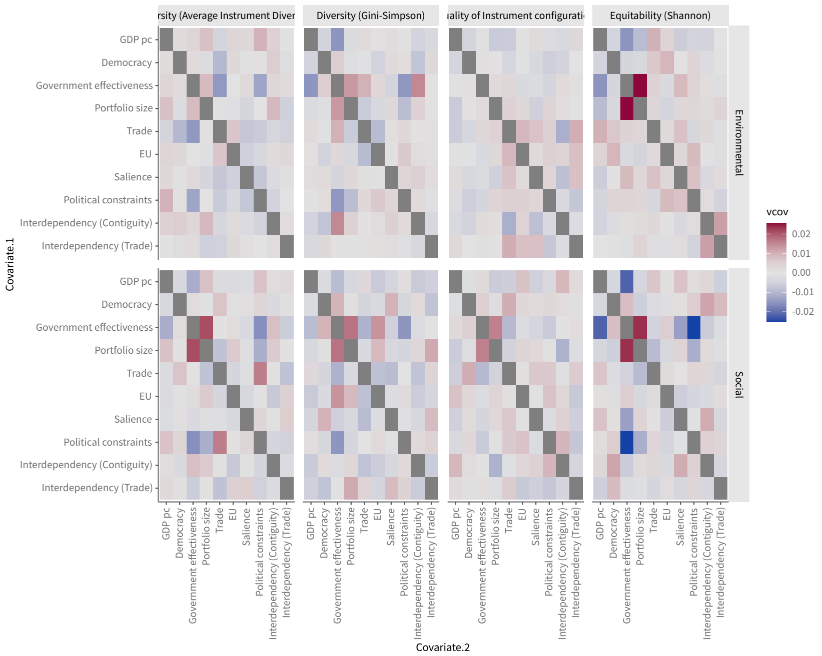

Figure 8.20: Variance-covariance matrix of main effects.

```
rm(S.Sigma, vcov.Sigma)
```

8.6 Model evaluation

What is the model fit?

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

Obs.sd <- Y %>%
  as.data.frame.table() %>%
  as_tibble() %>%
  rename(Sector = Var1, Country = Var2,
         Year = Var3, Diversity = Var4,
         value = Freq) %>%
  mutate(Year = as.integer(as.numeric(as.character(Year)))) %>%
  group_by(Sector, Diversity) %>%
  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, Diversity) %>%
  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(Diversity ~ Sector) +
  expand_limits(x = 0)


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

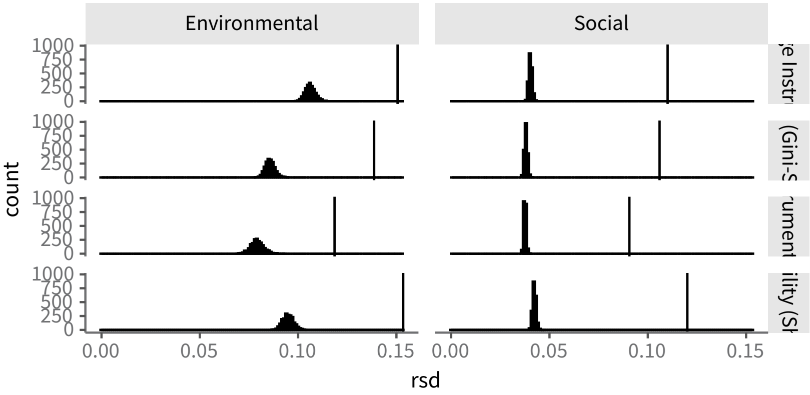

Figure 8.21: Model fit. Residual standard deviation (distribution) against observed standard deviation (vertical line).

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

Sector	Diversity	Pseudo.R2
Environmental	Diversity (Average Instrument Diversity)	0.30
Environmental	Diversity (Gini-Simpson)	0.38
Environmental	Equality of Instrument configurations	0.33
Environmental	Equitability (Shannon)	0.38
Social	Diversity (Average Instrument Diversity)	0.63
Social	Diversity (Gini-Simpson)	0.64
Social	Equality of Instrument configurations	0.59
Social	Equitability (Shannon)	0.65

```
rm(S.rsd)
```

Which are the observations with higher residuals, further away from the expectation?[1]

[1] Negative residuals are cases where the country has lower diversity than expected according to the model.

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

ggs(s, family = "resid", par_labels = L.data, sort = FALSE) %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(!str_detect(Country, "Z-")) %>%
  group_by(Country) %>%
  summarize(Residual = mean(value)) %>%
  mutate(`Mean absolute residual` = abs(Residual)) %>%
  ungroup() %>%
  arrange(desc(`Mean absolute residual`)) %>%
  select(-`Mean absolute residual`) %>%
  slice(1:8) %>%
  kable()
```

8.7 Magnitude of effects

```
L.mu <- plab("mu", list(Sector = sector.label,
                        Country = country.label,
                        Year = year.label.numeric,
                        Diversity = diversity.label)) %>%
  filter(str_detect(Country, "^Z-"))
```

Country	Residual
Canada	-0.19
Greece	0.14
Australia	0.11
Finland	-0.09
Netherlands	-0.07
Norway	0.07
Portugal	0.06
Denmark	0.05

```
ci.mu <- ggs(s, family = "^mu\\[", par_labels = L.mu, sort = FALSE) %>%
  mutate(Year = as.integer(as.character(Year))) %>%
  ci()

ci.mu %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  ggplot(aes(x = Year, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen portfolio size goes\nfrom minimum to maximum\nover time")
```

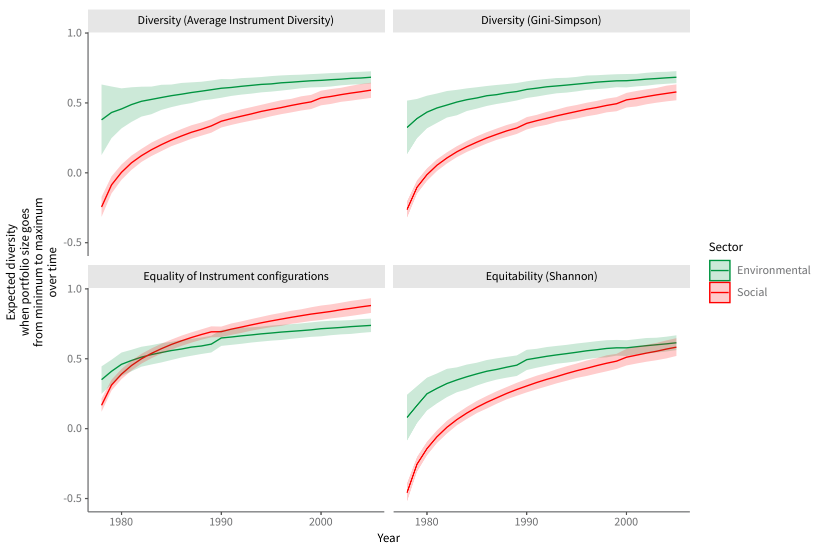

Figure 8.22: Magnitude of the effects: change in the expected diversity when portfolio size goes from the minimum to the maximum observed value over time, with the rest of the variables at their means.

```
ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year == max(Year)) %>%
  left_join(gov.eff.df %>%
            rename(Country = country, Year = year)) %>%
  rename(`Government effectiveness` = value) %>%
  ggplot(aes(x = `Government effectiveness`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen government effectiveness goes\nfrom minimum to maximum")

ci.mu %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year == max(Year)) %>%
```

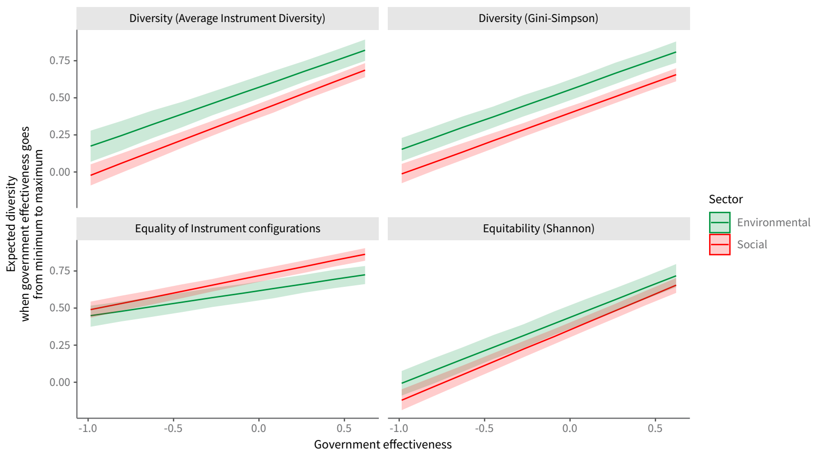

Figure 8.23: Magnitude of the effects: change in the expected diversity when government effectiveness goes from the minimum to the maximum observed value, with the rest of the variables at their means.

```
  left_join(constraints.d) %>%
  rename(`Political constraints` = polcon) %>%
  ggplot(aes(x = `Political constraints`, y = median,
             ymin = Low, ymax = High,
             color = Sector, fill = Sector)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  facet_wrap(~ Diversity) +
  ylab("Expected diversity\nwhen political constraints go\nfrom minimum to maximum")
```

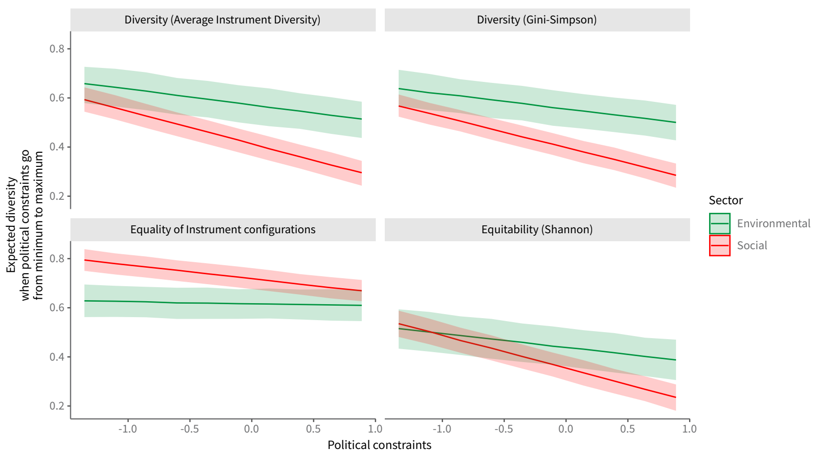

Figure 8.24: Magnitude of the effects: change in the expected diversity when political constraints go from the minimum to the maximum observed value, with the rest of the variables at their means.

```
f1 <-
  ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country == "Z-01") %>%
  filter(Year > min(Year)) %>%
  left_join(D %>%
            filter(Measure == "Size") %>%
            select(Country, Sector, Year, value) %>%
            rename(`Portfolio size` = value)) %>%
  ggplot(aes(x = `Portfolio size`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(a)") +
```

```
  ylab("Expected diversity")

f2 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 2:11))) %>%
  filter(Year ==  max(Year)) %>%
  left_join(gov.eff.df %>%
              rename(Country = country, Year = year)) %>%
  rename(`Government effectiveness` = value) %>%
  ggplot(aes(x = `Government effectiveness`, y = median,
             ymin = Low, ymax = High)) +
  geom_line() +
  geom_ribbon(alpha = 0.2, aes(color = NULL)) +
  expand_limits(y = c(0, 1)) +
  ggtitle("(b)") +
  ylab("Expected diversity")

f3 <- ci.mu %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Sector == "Environmental") %>%
  filter(Country %in% paste0("Z-", sprintf("%02d", 12:21))) %>%
  filter(Year ==  max(Year)) %>%
  left_join(constraints.d) %>%
  rename(`Political constraints` = original.polcon) %>%
  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, 1)) +
  ggtitle("(c)") +
  ylab("Expected diversity")

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

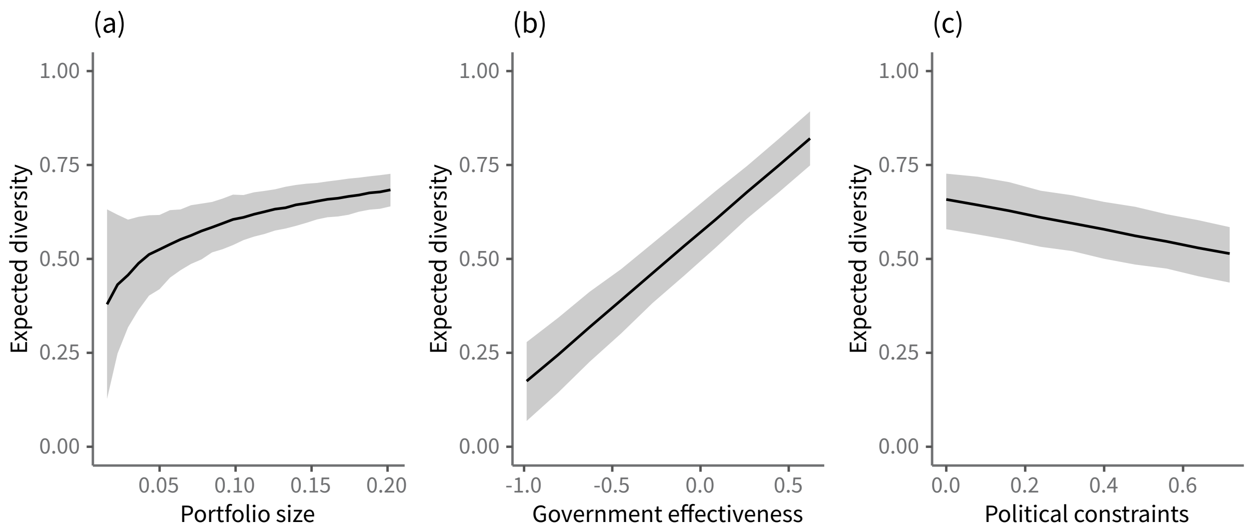

Figure 8.25: Magnitude of the effects: change in the expected diversity when (a) portfolio size, (b) government effectiveness or (c) political constraints move from the minimum to the maximum observed. In all cases the rest of the variables are fixed at their means.

9
Model comparison

9.1 Policy feedback

```
d ← NULL
load("ci-betas-diversity-main.RData")
d ← bind_rows(ci.betas %>%
              mutate(Model = "Government effectiveness") %>%
              mutate(Variable = as.character(Variable)) %>%
              mutate(Variable = ifelse(Variable == "Government effectiveness",
                                       "GovEff/PolFb", Variable)))

load("ci-betas-diversity-vpi.RData")
d ← bind_rows(d, ci.betas %>%
              mutate(Model = "Policy feedback (Atemporal)") %>%
              mutate(Variable = as.character(Variable)) %>%
              mutate(Variable = ifelse(Variable == "Policy feedback",
                                       "GovEff/PolFb", Variable)))

d %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggplot(aes(x = median, y = Variable, color = Model)) +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(xmin = Low, xmax = High), lwd = 0.8,
                 position = position_dodge(width = 0.3)) +
  geom_linerange(aes(xmin = low, xmax = high), lwd = 0.2,
                 position = position_dodge(width = 0.3)) +
  geom_vline(xintercept = 0, lty = 3) +
  scale_color_colorblind() +
  xlab("HPD") +
  ylab("Parameter") +
  theme(legend.position = "bottom")
```

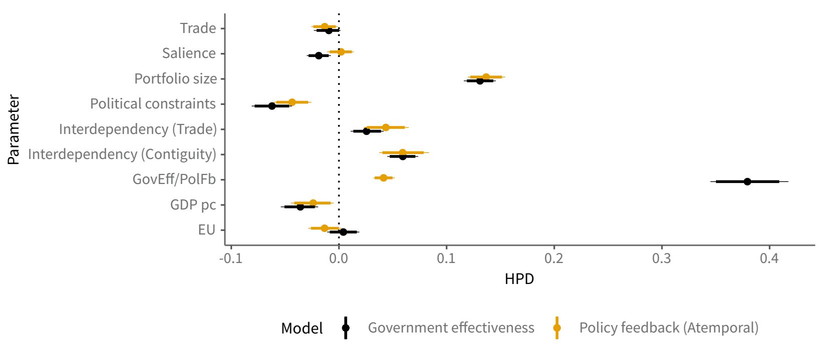

Figure 9.1: Model comparison for the same model specification, diversity measure, and different bureaucratic variable. Environmental Sector. Average Instrument Diversity.

```
rm(d)
```

9.2 Instrument-based mechanisms

```
d ← NULL
load("ci-betas-diversity-mbi.RData")
d.liability ← bind_rows(ci.betas %>%
                        mutate(Model = "Government effectiveness") %>%
                        mutate(Data = "New instrument"))

load("ci-betas-diversity-main.RData")
d ← bind_rows(ci.betas %>%
              mutate(Model = "Government effectiveness") %>%
              mutate(Data = "Original"))

d ← bind_rows(d, d.liability)

d %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  filter(Model == "Government effectiveness") %>%
  ggplot(aes(x = median, y = Variable, color = Data)) +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(xmin = Low, xmax = High), lwd = 0.8,
                 position = position_dodge(width = 0.3)) +
  geom_linerange(aes(xmin = low, xmax = high), lwd = 0.2,
                 position = position_dodge(width = 0.3)) +
  geom_vline(xintercept = 0, lty = 3) +
  scale_color_colorblind() +
  xlab("HPD") +
  ylab("Parameter") +
  theme(legend.position = "bottom")
```

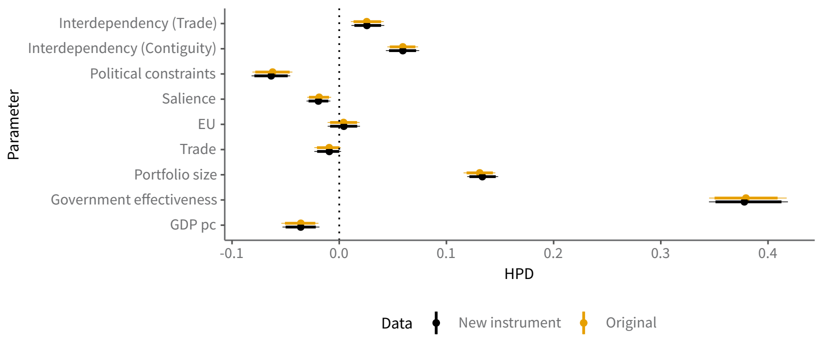

Figure 9.2: Model comparison for the same model specification, diversity measure, by dataset. Environmental Sector. Average Instrument Diversity.

```
rm(d)
```

9.3 Democracy

```
d ← NULL
load("ci-betas-diversity-main.RData")
d ← bind_rows(ci.betas %>%
              mutate(Model = "Baseline"))

load("ci-betas-diversity-democracy.RData")
d ← bind_rows(d, ci.betas %>%
              mutate(Model = "+ Democracy (V-Dem average)"))

d %>%
  filter(Sector == "Environmental") %>%
  filter(Diversity == "Diversity (Average Instrument Diversity)") %>%
  ggplot(aes(x = median, y = Variable, color = Model)) +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(xmin = Low, xmax = High), lwd = 0.8,
                 position = position_dodge(width = 0.3)) +
```

```
geom_linerange(aes(xmin = low, xmax = high), lwd = 0.2,
               position = position_dodge(width = 0.3)) +
geom_vline(xintercept = 0, lty = 3) +
scale_color_colorblind() +
xlab("HPD") +
ylab("Parameter") +
theme(legend.position = "bottom")
```

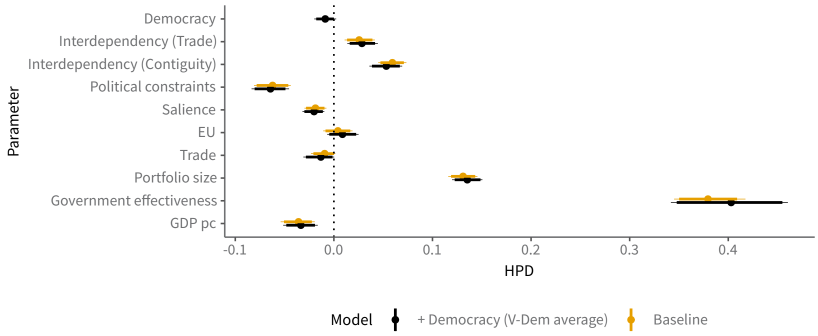

Figure 9.3: Model comparison for the same model specification, diversity measure, and different bureaucratic variable. Environmental Sector. Average Instrument Diversity.

Programming environment

```r
sessionInfo()
```

```
## R version 4.0.4 (2021-02-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Gentoo/Linux
##
## Matrix products: default
## BLAS:   /usr/lib64/blas/blis/libblas.so.3
## LAPACK: /usr/lib64/libopenblas_haswellp-r0.3.13.so
##
## locale:
##  [1] LC_CTYPE=ca_AD.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=ca_AD.UTF-8        LC_COLLATE=ca_AD.UTF-8
##  [5] LC_MONETARY=ca_AD.UTF-8    LC_MESSAGES=ca_AD.UTF-8
##  [7] LC_PAPER=ca_AD.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=ca_AD.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] forcats_0.5.1          PolicyPortfolios_0.2.2 ggrepel_0.9.1
##  [4] colorspace_2.0-0       stringr_1.4.0          GGally_2.1.1
##  [7] ggmcmc_1.5.1.1         runjags_2.2.0-2        rjags_4-10
## [10] coda_0.19-4            scales_1.1.1           ggthemes_4.2.4
## [13] extrafont_0.17         gridExtra_2.3          ggplot2_3.3.3
## [16] tidyr_1.1.3            dplyr_1.0.5            kableExtra_1.3.4
## [19] tikzDevice_0.12.3.1    rmarkdown_2.7          knitr_1.31
## [22] colorout_1.2-2
##
## loaded via a namespace (and not attached):
##  [1] httr_1.4.2         tufte_0.9          viridisLite_0.3.0  splines_4.0.4
##  [5] assertthat_0.2.1   yaml_2.2.1         Rttf2pt1_1.3.8     pillar_1.5.1
##  [9] lattice_0.20-41    glue_1.4.2         extrafontdb_1.0    digest_0.6.27
## [13] RColorBrewer_1.1-2 rvest_1.0.0        cowplot_1.1.1      htmltools_0.5.1.1
## [17] Matrix_1.3-2       plyr_1.8.6         pkgconfig_2.0.3    bookdown_0.21
## [21] purrr_0.3.4        webshot_0.5.2      svglite_2.0.0      tibble_3.1.0
## [25] mgcv_1.8-34        generics_0.1.0     farver_2.1.0       ellipsis_0.3.1
## [29] withr_2.4.1        magrittr_2.0.1     crayon_1.4.1       evaluate_0.14
## [33] fansi_0.4.2        nlme_3.1-152       MASS_7.3-53.1      xml2_1.3.2
## [37] foreign_0.8-81     vegan_2.5-7        tools_4.0.4        ineq_0.2-13
## [41] lifecycle_1.0.0    munsell_0.5.0      cluster_2.1.1      compiler_4.0.4
## [45] systemfonts_1.0.1  rlang_0.4.10       debugme_1.1.0      rstudioapi_0.13
## [49] filehash_2.4-2     labeling_0.4.2     gtable_0.3.0       DBI_1.1.1
## [53] reshape_0.8.8      reshape2_1.4.4     R6_2.5.0           utf8_1.1.4
## [57] permute_0.9-5      stringi_1.5.3      parallel_4.0.4     Rcpp_1.0.6
## [61] vctrs_0.3.6        tidyselect_1.1.0   xfun_0.21
```

Henisz, Witold J. 2002. "The Institutional Environment for Infrastructure Investment." *Industrial and Corporate Change* 11 (2): 355–89.

Volkens, Pola AND Merz, Andrea AND Lehmann. 2013. *The Manifesto Data Collection. Manifesto Project (Mrg/Cmp/Marpor). Version 2013b*. Berlin: Wissenschaftszentrum Berlin für Sozialforschung (WZB).