Xavier Fernández-i-Marín, Markus Hinterleitner, Christoph Knill, Yves Steinebach

Replication material for ‘Do Governments put their Money where their Mouth is? Policy Adoption and Administrative Resource Provision in 15 OECD Countries’

2024-09-20 17:10

Contents

1
Introduction

2
Data description

```r
# RUN
source("load_packages.R")
inv.logit <- function(x) return(1 / (1 + (exp(-(x)))))
pal.sector <- c(rgb(0, 149, 64, max = 255), rgb(255, 0, 0, max = 255))
ihs <- function(x) return(log(x + sqrt(x^2 + 1)))
library(ggridges)
```

Get the year-to-year portfolio.

```r
# RUN
load("../accupol-extended.RData")
pps.y <- pps %>%
  filter(Measure == "Size") %>%
  select(-Measure.label) %>%
  pivot_wider(names_from = Measure, values_from = value) %>%
  relocate(Sector) %>%
  mutate(Sector = fct_relevel(Sector, "Environmental")) %>%
  filter(!Country %in% c("Turkey", "Mexico")) %>%
  droplevels()
```

Describe portfolio spaces.

```r
pps %>%
  filter(Measure == "Space") %>%
  select(Sector, Measure, value, Measure.label) %>%
  unique()
nIe <- length(unique(PE$Instrument))
nTe <- length(unique(PE$Target))
nIs <- length(unique(PS$Instrument))
nTs <- length(unique(PS$Target))
```

- Environmental, instruments: 13.
- Environmental, targets: 50.
- Social, instruments: 7.
- Social, targets: 28.

Produce dummy variables when portfolio size increases in the environmental portfolio due to: subsidies, taxes and liability schemes

```r
# RUN
env.specific.instruments.growth <- PE %>%
  group_by(Sector, Country, Instrument, Target) %>%
  arrange(Sector, Country, Instrument, Target, Year) %>%
  mutate(Creation = ifelse(covered == 1 & lag(covered) == 0, 1, 0)) %>%
  ungroup() %>%
  mutate(C.Subsidies = ifelse(Instrument == "105" & Creation == 1, 1, 0)) %>% # "Subsidy / Tax"
  mutate(C.Taxes = ifelse(Instrument == "104" & Creation == 1, 1, 0)) %>% # "Tax / Levy"
  mutate(C.Liability_Schemes = ifelse(Instrument == "106" & Creation == 1, 1, 0)) %>% # "Liability scheme"
  select(Sector, Country, Year, starts_with("C.")) %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "CreationInstrument", values_to = "value") %>%
  group_by(Sector, Country, Year, CreationInstrument) %>%
  summarize(value = ifelse(sum(value) > 0, 1, 0)) %>%
  mutate(value = ifelse(is.na(value), 0, value)) %>%
  mutate(value = as.integer(value)) %>%
  ungroup() %>%
  pivot_wider(names_from = CreationInstrument, values_from = value) %>%
  filter(Country %in% pps.y$Country) %>%
  mutate(Sector = as.factor(Sector)) %>%
  mutate(Country = as.factor(Country)) %>%
  right_join(pps.y %>%
             select(Sector, Country, Year)) %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”CreationInstrument”, values_to = ”value”) %>%
  mutate(value = ifelse(is.na(value), 0, value)) %>%
  mutate(value = as.integer(value)) %>%
  pivot_wider(names_from = CreationInstrument, values_from = value)
```

Get the implementation capacity

```
# RUN
path ← ”~/est/state_capacity-measurement/”


load(paste0(path, ”scores-restricted-environmental-moneymouth.RData”))
scores.environmental.regular ← scores %>%
  mutate(Model = ”Reference”)

load(paste0(path, ”scores-restricted-social-moneymouth.RData”))
scores.social.regular ← scores %>%
  mutate(Model = ”Reference”)

ic ← scores.environmental.regular %>%
  bind_rows(scores.social.regular) %>%
  mutate(Score = inv.logit(Score)) %>%
  select(-c(high_90, low_90)) %>%
  select(Sector, Country, Year, Capacity = Score) %>%
  mutate(Country = as.character(Country)) %>%
  mutate(Country = ifelse(Country == ”United States of America”, ”United States”, Country)) %>%
  mutate(Country = factor(Country)) %>%
  mutate(Sector = factor(Sector))
```

Get the government expenditure

```
# RUN
path ← ”~/est/country_data/government_expenditure_budgets/”
load(paste0(path, ”government_expenditure.RData”))
# loads government.expenditure
```

Get the administration costs of social protection

```
# RUN
load(”data-social_administrative_costs/data-social_administrative_costs.RData”)
# loads sac

mmc.sac ← pps %>%
  filter(Measure == ”Size”) %>%
  select(-Measure.label) %>%
  pivot_wider(names_from = Measure, values_from = value) %>%
  relocate(Sector) %>%
  mutate(Sector = fct_relevel(Sector, ”Environmental”)) %>%
  droplevels() %>%
  left_join(sac) %>%
  group_by(Sector, Country) %>%
  arrange(Sector, Country, Year) %>%
  mutate(Size.diff = Size - lag(Size)) %>%
  mutate(Cost.diff = Cost - lag(Cost)) %>%
  ungroup() %>%
  mutate(Compensation.SAC = log(exp(Cost.diff) / exp(Size.diff))) %>%
  filter(!is.na(Compensation.SAC)) %>%
  mutate(Country = factor(Country))


save(mmc.sac, file = ”data-moneymouth-compensation-sac.RData”)
```

Mix portfolio size with either implementation capacity or government expenditure and work with their differences

to calculate compensation:

- BCR $= log(\text{Size}/\text{Capacity})$
- Compensation $= log(\frac{exp(\Delta\text{Capacity})}{exp(\Delta\text{Size})})$
- Compensation $= log(\frac{\Delta\text{Expenditure}}{exp(\Delta\text{Size})})$

```
# RUN
d.full ← pps.y %>%
  left_join(ic) %>%
  left_join(expenditure.accupol) %>%
  mutate(BCR = log(Size / Capacity)) #%>%

d.full ← d.full %>%
  filter(Year >= 1992) %>%
  filter(!Country %in% c("Australia", "Canada", "Japan",
                         "New Zealand",
                         "United States",
                         "United Kingdom")) %>%
  droplevels()


d.diff.capacity <- d.full %>%
  select(Sector, Country, Year,
         Capacity,
         starts_with("Expenditure.")) %>%
  pivot_longer(-c(Sector, Country, Year),
               names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Variable) %>%
  arrange(Sector, Country, Variable ,Year) %>%
  mutate(diff.1 = case_when(
    Variable %in% c("Capacity", "Expenditure.Percent", "Expenditure.Index 2010") ~ value - lag(value),
    Variable %in% c("Expenditure.Absolute", "Expenditure.Absolute 2010") ~ (value - lag(value))/lag(value),
    )) %>%
  mutate(diff.2 = case_when(
    Variable %in% c("Capacity", "Expenditure.Percent", "Expenditure.Index 2010") ~ lead(value) - lag(value),
    Variable %in% c("Expenditure.Absolute", "Expenditure.Absolute 2010") ~ (lead(value) - lag(value))/lag(value),
    )) %>%
  mutate(diff.3 = case_when(
    Variable %in% c("Capacity", "Expenditure.Percent", "Expenditure.Index 2010") ~ lead(value, 2) - lag(value),
    Variable %in% c("Expenditure.Absolute", "Expenditure.Absolute 2010") ~ (lead(value, 2) - lag(value))/lag(value),
    )) %>%
  #
  ungroup() %>%
  select(-value) %>%
  pivot_longer(-c(Sector, Country, Year, Variable),
               names_to = "Lag", values_to = "Capacity") %>%
  mutate(Lag = str_replace(Lag, "diff\\.", "Lag: ")) %>%
  select(Sector, Country, Year, Variable, Lag, Capacity) %>%
  pivot_wider(names_from = Variable, values_from = Capacity)



d.diff.size <- d.full %>%
  pivot_longer(-c(Sector, Country, Year),
               names_to = "Variable", values_to = "value") %>%
  filter(Variable %in% c("Size")) %>%
  group_by(Sector, Country, Variable) %>%
  arrange(Sector, Country, Variable, Year) %>%
  mutate(diff.1 = value - lag(value)) %>%
  ungroup() %>%
  select(-value) %>%
  pivot_longer(-c(Sector, Country, Year, Variable),
               names_to = "Lag", values_to = "Size") %>%
  mutate(Lag = str_replace(Lag, "diff\\.", "Lag: ")) %>%
  select(Sector, Country, Year, Size) %>%
  mutate(Size.bin = ifelse(Size > 0, 1, 0))


d.diff.compensation <- left_join(d.diff.capacity, d.diff.size) %>%
  mutate(Compensation = ((Expenditure.Absolute) / exp(Size) )) %>%
  mutate(Compensation.IHS = ((Expenditure.Absolute) / (ihs(Size) + 1)) ) %>%
  mutate(Compensation.LEE = log(exp(Expenditure.Absolute) / exp(Size)) ) %>%
  mutate(Compensation.plus1 = ((Expenditure.Absolute) / (Size + 1) )) %>%
  mutate(Compensation.log.pre = ((Expenditure.Absolute) / exp(Size) )) %>%
  mutate(Compensation.log = log( Compensation.log.pre + abs(min(Compensation.log.pre, na.rm = TRUE)) + 1)) %>%
  group_by(Sector, Country) %>%
  mutate(Expenditure.Absolute.bin = ifelse(Expenditure.Absolute > sd(Expenditure.Absolute, na.rm = TRUE), 1, 0)) %>%
  ungroup() %>%
  mutate(Compensation.bin = as.integer(Size.bin == 1 & Expenditure.Absolute.bin == 1))

d.diff.compensation <- d.diff.compensation %>%
  mutate(`Compensated events` = case_when(
           (Size.bin == 1 & Expenditure.Absolute.bin == 1) ~ "Compensated",
           (Size.bin == 1 & Expenditure.Absolute.bin != 1) ~ "Uncompensated",
           TRUE ~ "Non-event")) %>%
  mutate(`Compensated events` = fct_relevel(`Compensated events`, c("Non-event",
                                                                    "Uncompensated",
                                                                    "Compensated")))


d <- d.diff.compensation %>%
  select(Sector, Country, Year, Lag, Compensation, Compensation.bin,
         `Compensated events`,
         Compensation.log,
         Compensation.IHS,
         Compensation.LEE,
         Compensation.plus1) %>%
  left_join(d.full) %>%
  select(-c(`Chain linked volumes, index 2010=100`,
            `Chain linked volumes (2010), million euro`,
             `Current prices, million euro`))
```

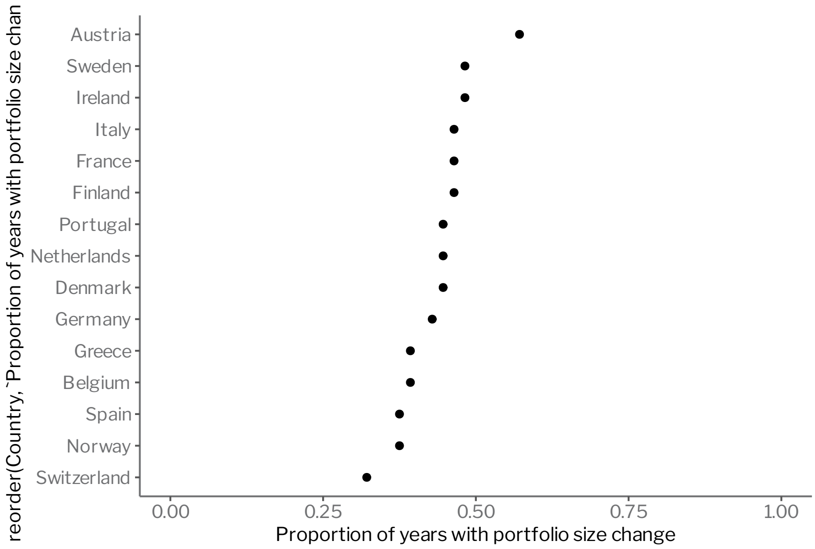
Report the percentage of years in which there is portfolio size change from year to year

```
d.diff.size %>%
  filter(!is.na(Size)) %>%
  summarize(`Prop Years with portfolio size change` = length(which(Size ≠ 0)) / n()) %>%
  mykbl(”Overall proportion of years with portfolio size change”)
```

Table 2.1: Overall proportion of years with portfolio size change

Prop Years with portfolio size change
0.44

```
d.diff.size %>%
  filter(!is.na(Size)) %>%
  group_by(Country) %>%
  summarize(`Proportion of years with portfolio size change` = length(which(Size ≠ 0)) / n()) %>%
  ggplot(aes(x = `Proportion of years with portfolio size change`,
             y = reorder(Country,
                         `Proportion of years with portfolio size change`))) +
  geom_point() +
  expand_limits(x = c(0, 1))
```



```
# RUN
mmc ← d # money mouth compensation
save(mmc, d.diff.compensation, env.specific.instruments.growth, file = ”data-moneymouth-compensation.RData”)

# Show the compensations
# as an example, to convince the reviewer
# do it for the averages of countries

av.compensations ← d.full %>%
  filter(Sector == ”Social”) %>%
  group_by(Year) %>%
  summarize(Size = mean(Size),
            Expenditure = mean(Expenditure.Absolute)) %>%
  ungroup() %>%
  arrange(Year) %>%
  mutate(Size.diff = (Size - lag(Size))) %>% # / lag(Size))
  mutate(Expenditure.diff = ( (Expenditure - lag(Expenditure)) / lag(Expenditure) ) ) %>%
  mutate(Compensation = Expenditure.diff / exp(Size.diff)) %>%
  mutate(`Δ Portfolio size = 0` = Expenditure.diff / exp(0)) %>%
  mutate(`Δ Portfolio size = 1` = Expenditure.diff / exp(1)) %>%
  mutate(`Δ Portfolio size = 2` = Expenditure.diff / exp(2)) %>%
  mutate(`Δ Portfolio size = 3` = Expenditure.diff / exp(3))


av.compensations %>%
  select(Year, starts_with(”Δ Portfolio size”)) %>%
  pivot_longer(-Year, names_to = ”Size diff”, values_to = ”Compensation”) %>%
  ggplot(aes(x = Year, y = Compensation)) +
  geom_line() +
  facet_grid(~ `Size diff`)
```

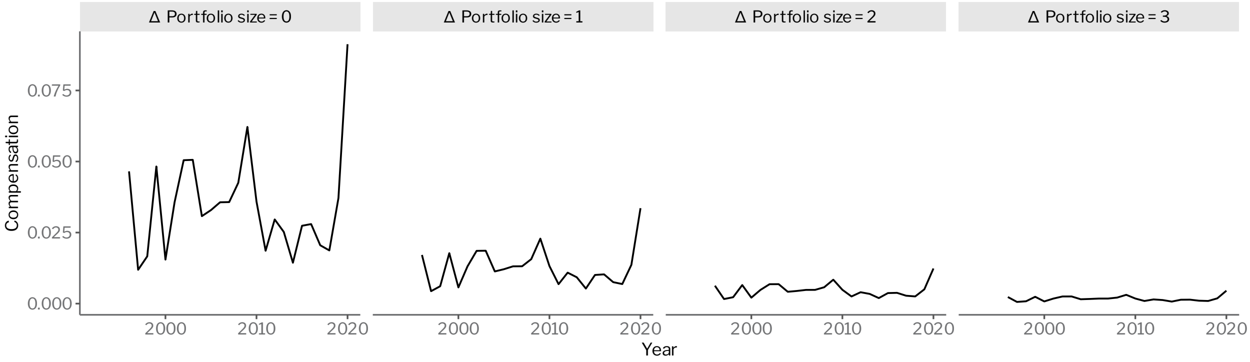


Figure 2.1: For the average percent change in expenditure across countries in the social sector, scenarios of compensation for different portfolio size changes.

```
av.compensations %>%
  rename(`ΔExpenditure` = Expenditure.diff) %>%
  select(Year, `ΔExpenditure`, starts_with("Compensation ")) %>%
  filter(Year ⩾ 1996) %>%
  mykbl("Example of compensations for average expenditure percent changes from year to year (social sector).", digits = 4)
```

Table 2.2: Example of compensations for average expenditure percent changes from year to year (social sector).

Year	ΔExpenditure
1996	0.0465
1997	0.0119
1998	0.0166
1999	0.0482
2000	0.0155
2001	0.0356
2002	0.0505
2003	0.0506
2004	0.0308
2005	0.0329
2006	0.0357
2007	0.0357
2008	0.0425
2009	0.0622
2010	0.0359
2011	0.0186
2012	0.0296
2013	0.0252
2014	0.0144
2015	0.0274
2016	0.0280
2017	0.0205
2018	0.0187
2019	0.0371
2020	0.0913

```
d.full %>%
  filter(Sector == "Social") %>%
  filter(Year ⩾ 1996) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(Size.diff = Size - lag(Size)) %>%
  ungroup() %>%
  count(Size.diff == 0) %>%
  na.omit() %>%
  mutate(p = n / sum(n)) %>%
  mykbl("Proportion of portfolio size NOT changing from year to year.")
```

Table 2.3: Proportion of portfolio size NOT changing from year to year.

Size.diff == 0	n	p

FALSE	84	0.23
TRUE	276	0.77

```
d.full %>%
  filter(Sector == "Social") %>%
  filter(Year >= 1996) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(Size.diff = Size - lag(Size)) %>%
  ungroup() %>%
  filter(Size.diff != 0) %>%
  count(Size.diff > 0) %>%
  na.omit() %>%
  mutate(p = n / sum(n)) %>%
  mykbl("Proportion of portfolio size increasing (or decreasing) when there is a change from year to year.")
```

Table 2.4: Proportion of portfolio size increasing (or decreasing) when there is a change from year to year.

Size.diff > 0	n	p
FALSE	18	0.21
TRUE	66	0.79

```
d.full %>%
  filter(Sector == "Social") %>%
  filter(Year >= 1996) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(Size.diff = Size - lag(Size)) %>%
  ungroup() %>%
  mutate(Direction = case_when(
    Size.diff > 0 ~ "Expansion",
    Size.diff < 0 ~ "Dismantling",
    Size.diff == 0 ~ "Status Quo")) %>%
  na.omit() %>%
  filter(Direction != "Status Quo") %>%
  group_by(Direction) %>%
  summarize(Min = min(abs(Size.diff) * 100),
            Max = max(abs(Size.diff)) * 100) %>%
  mykbl("Percent points change by direction of policy change.")
```

Table 2.5: Percent points change by direction of policy change.

Direction	Min	Max
Dismantling	0.51	1.02
Expansion	0.51	4.08

3

Explanatory model for capacity expansion: Capacity expansion. Baseline. Up to 5 year effects. Robust/clustered errors by country. Non-nested Varying intercepts by year and by country. AR(1). Sophisticated comparison of means. Hierarchical between sectors. All lags included.

mm-tscs-104

```r
load("data-moneymouth-compensation.RData") # loads 'mmc'

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
#year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

3.1 Covariates

VPI

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

vpi.original <- vpi

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

```
# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon
load(”data-parlgov/data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = year.start:year.finish)

fd.pre ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(sd) %>%
  left_join(m.country.interpolation.annual) %>%
  arrange(Sector, Country, Year) %>%
  relocate(Sector, Country, Year)


# Create capacity lag and binary indicators for portfolio increase
fd ← fd.pre %>%
  group_by(Sector, Country) %>%
  arrange(Sector, Country, Year) %>%
  mutate(Expenditure.Absolute.diff = (Expenditure.Absolute - lag(Expenditure.Absolute)) / lag(Expenditure.Absolute)) %>%
  mutate(Size.diff = (Size - lag(Size))) %>%
  mutate(Growth = ifelse(Size > lag(Size), 1, 0)) %>%
  ungroup() #%>%


Y.df ← fd %>%
  select(Sector, Country, Year, Expenditure.Absolute.diff) %>%
  group_by(Sector) %>%
  mutate(Expenditure.Absolute.diff = std1(Expenditure.Absolute.diff)) %>%
  ungroup()

Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = ”Expenditure.Absolute.diff”)

Y.df %>%
  summarize(pMissing = length(which(is.na(Expenditure.Absolute.diff))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(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)

X.df ← fd %>%
  select(Sector, Country, Year, Growth)
```

```
X ← X.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Growth")

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  X = unname(X)
)

inits ← function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    theta = array(runif(nS * nC * 5, 0, 2), dim = c(nS, nC, 5))
  )
}


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 6:nY) {
        #Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        Y[s,c,y] ~ dt(mu[s,c,y], tau[s,c], nu[s])
        mu[s,c,y] ←
                     alpha[s,y] +
                     delta[s,c] +
                     theta[s,c,1] * X[s,c,y] +
                     theta[s,c,2] * X[s,c,y-1] +
                     theta[s,c,3] * X[s,c,y-2] +
                     theta[s,c,4] * X[s,c,y-3] +
                     theta[s,c,5] * X[s,c,y-4] +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      #
      mu[s,c,5] ←
                   alpha[s,5] +
                   delta[s,c] +
                   theta[s,c,1] * X[s,c,5] +
                   theta[s,c,2] * X[s,c,5-1] +
                   theta[s,c,3] * X[s,c,5-2] +
                   theta[s,c,4] * X[s,c,5-3]
      #
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      #lambda_c[s,c] ~ dt(0, 0.1^-2, 3)
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      delta[s,c] ~ dnorm(0, 1^-2)
      #
      for (t in 1:5) {
        #theta[s,c,t] ~ dnorm(1, 1^-2)
        #theta[s,c,t] ~ dnorm(0, 1^-2)
        #theta[s,c,t] ~ dt(0.5, 0.5^-2, 3)
        theta[s,c,t] ~ dnorm(Theta[c,t], tau.theta[s])T(0,)
      }
    }
    tau.theta[s] ~ dgamma(1, 1)
    sigma.theta[s] ← 1 / sqrt(tau.theta[s])
    #
    rho[s] ~ dunif(-1, 1)
    for (y in 1:nY) {
      alpha[s,y] ~ dnorm(0.5, 1^-2)
    }
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  for (c in 1:nC) {
    for (t in 1:5) {
      Theta[c,t] ~ dnorm(0, 1^-2)T(0,)
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par ← NULL
par ← c(par, "theta")
par ← c(par, "alpha")
par ← c(par, "delta")
par ← c(par, "Theta", "sigma.theta")
par ← c(par, "nu")
par ← c(par, "rho")
par ← c(par, "lambda_c")
par ← c(par, "lambda")

sample.length ← c("test", "middle", "serious")
sample.length ← sample.length[2]
if (sample.length == "test") {
  chains ← 2
  adapt ← 1e2
  burnin ← 5e2
  run ← 5e2
  thin ← 1
} else if (sample.length == "middle") {
  chains ← 3
  adapt ← 2e2
  burnin ← 1e4
  run ← 2e3
  thin ← 1
} else if (sample.length == "serious") {
  chains ← 4
  adapt ← 5e2
  burnin ← 5e4
  run ← 2e3
  thin ← 10
}

method ← "parallel"
```

Data passed:

```
str(D)
```

```
## List of 5
##  $ Y : num [1:2, 1:15, 1:19] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC: int 15
##  $ nS: int 2
##  $ nY: int 19
##  $ X : num [1:2, 1:15, 1:19] NA NA NA NA NA NA NA NA NA NA ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 6:nY) {
##                  Y[s,c,y] ~ dt(mu[s,c,y], tau[s,c], nu[s])
##         mu[s,c,y] ←
##                       alpha[s,y] +
##                       delta[s,c] +
##                       theta[s,c,1] * X[s,c,y] +
##                       theta[s,c,2] * X[s,c,y-1] +
##                       theta[s,c,3] * X[s,c,y-2] +
##                       theta[s,c,4] * X[s,c,y-3] +
##                       theta[s,c,5] * X[s,c,y-4] +
##                       rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       #
##       mu[s,c,5] ←
##                     alpha[s,5] +
##                     delta[s,c] +
##                     theta[s,c,1] * X[s,c,5] +
##                     theta[s,c,2] * X[s,c,5-1] +
##                     theta[s,c,3] * X[s,c,5-2] +
##                     theta[s,c,4] * X[s,c,5-3]
##       #
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##             lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       delta[s,c] ~ dnorm(0, 1^-2)
##       #
##       for (t in 1:5) {
##                                  theta[s,c,t] ~ dnorm(Theta[c,t], tau.theta[s])T(0,)
##       }
##     }
##     tau.theta[s] ~ dgamma(1, 1)
##     sigma.theta[s] <- 1 / sqrt(tau.theta[s])
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (y in 1:nY) {
##       alpha[s,y] ~ dnorm(0.5, 1^-2)
##     }
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   for (c in 1:nC) {
##     for (t in 1:5) {
##       Theta[c,t] ~ dnorm(0, 1^-2)T(0,)
##     }
##   }
## }
```

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

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

L.Theta <- plab("Theta", list(Country = country.label,
                              Lag = paste("Lag", 1:5)))

S.Theta <- ggs(s, family = "^Theta", par_labels = L.Theta)

ci.Theta <- ci(S.Theta)
```

Generate a table with the highest probability lags

```
tb <- ci(S.Theta) %>%
  select(Country, Lag, median) %>%
  group_by(Country) %>%
  arrange(desc(median)) %>%
  slice(1) %>%
  ungroup() %>%
  mutate(Lag = as.integer(as.numeric(str_replace(Lag, ”Lag ”, ””)))) %>%
  select(Country, Lag) %>%
  arrange(Lag)

tc ← ”Higehely likely lags at each country.”
if (knitr::is_latex_output()) {
  kable(tb, format = ”latex”, caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tb, format = ”html”, caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = ”center”, bootstrap_options = ”striped”, full_width = T)
}
```

Table 3.1: Highely likely lags at each country.

Country	Lag
Austria	2
Greece	2
Ireland	2
Finland	3
France	3
Germany	3
Netherlands	3
Portugal	3
Denmark	4
Italy	4
Norway	4
Spain	4
Sweden	4
Belgium	5
Switzerland	5

```
tb %>%
  kbl(format = ”latex”, caption = tc, booktabs = TRUE,
      position = ”ht”,
      label = ”higly-likely-lags-104”) %>%
  kable_styling(font_size = 8) %>%
  save_kable(file = ”tab-higly-likely-lags-104.tex”)

tb %>%
  write.table(file = ”highly-likely-lags-104.csv”, sep = ”\t”, row.names = FALSE)

# Also show the capped ones
tb %>%
  mutate(Lag = ifelse(Lag > 3, 3, Lag)) %>%
  arrange(Lag, Country) %>%
  kbl(format = ”latex”, caption = tc, booktabs = TRUE,
      position = ”ht”,
      label = ”higly-likely-lags-104-capped”) %>%
  kable_styling(font_size = 8) %>%
  save_kable(file = ”tab-higly-likely-lags-104.tex”)
```

4
Assessment of social administrative costs (SAC)

```
load("data-moneymouth-compensation.RData") # loads 'mmc'
mmc.nonsac <- mmc
load("data-moneymouth-compensation-sac.RData") # loads 'mmc.sac'

env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc.sac

set.year.start <- 1980
if (compensation.is.sac) {
  set.year.start <- 2009
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 2010
year.finish <- max(years)
year.finish <- 2020

comp.exp.costs <- full_join(
  mmc.sac %>%
    filter(Sector == "Social") %>%
    droplevels() %>%
    select(Country, Year, Psac = Cost),
  mmc.nonsac %>%
    filter(Lag == "Lag: 1") %>%
    filter(Sector == "Social") %>%
    droplevels() %>%
    select(Country, Year, Expenditure = Expenditure.Absolute)
  ) %>%
  arrange(Country, Year) %>%
  filter(!is.na(Psac) & !is.na(Expenditure)) %>%
  droplevels() %>%
  mutate(Expenditure.Administration = Expenditure * Psac/100)

comp.exp.costs %>%
  filter(Country %in% c("Germany", "Switzerland")) %>%
  droplevels() %>%
  ggplot(aes(x = Expenditure, y = Expenditure.Administration)) +
  geom_point() +
  ggpubr::stat_cor(aes(label = ..r.label..)) +
  scale_x_continuous(labels = label_number(scale_cut = cut_si("B"))) +
  scale_y_continuous(labels = label_number(scale_cut = cut_si("B"))) +
  facet_wrap(~ Country, scales = "free")
```

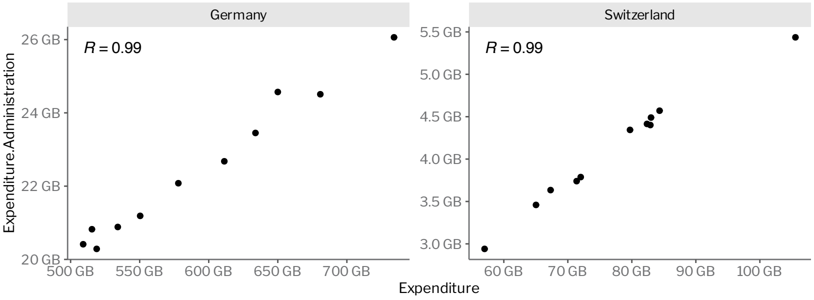


Figure 4.1: Correlation between full expenditure and expenditure on administrative costs, by country.

5

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties.

mm-tscs-2301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

5.1 *Covariates*

VPI

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

vpi.original ← vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ≥ year.start)

Y.df ← fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  # add intercept
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
```

```
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

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

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


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     #alpha[s,id.decade[y]] +
                     #alpha[s] +
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))
```

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA  ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10  ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2  ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627  ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528  ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0  ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                                                                 alpha[s,y] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] ← 1 / sigma.sq[s,c]
##       sigma.sq[s,c] ← exp(sigma[s,c])
##       sigma[s,c] ← lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] ←
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] ← exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }

## }
```

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

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

5.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2301.RData"))
```

6 Explanatory model for compensation: Baseline. Lag 1. Country specific lags. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties.

mm-tscs-2304

```r
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

country.lags <- read.table("highly-likely-lags-104.csv", header = TRUE, sep = "\t") %>%
  tibble() %>%
  mutate(Country = factor(Country)) %>%
  # restrict to maximum 3
  mutate(Lag = ifelse(Lag > 3, 3, Lag)) %>%
  rename(CountryLag = Lag)

mmc <- left_join(mmc, country.lags)

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

6.1 Covariates

VPI

```r
load("data-vpi/vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
```

```
vpi ← vpi %>%
  #
  filter(!Country %in% c(”Mexico”, ”Turkey”)) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == ”Implementation costs”, ”Top-down (VPI)”, Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == ”Policy feedback”, ”Bottom-up (VPI)”, Dimension))

vpi.original ← vpi

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

# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon
load(”data-parlgov/data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd.pre ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ⩾ year.start)


fd.compensation ← fd.pre %>%
  group_by(Sector, Country) %>%
  arrange(Sector, Country, Year) %>%
  mutate(Ediff = NA) %>%
  mutate(Ediff = ifelse(CountryLag == 1, (Expenditure.Absolute - lag(Expenditure.Absolute)) / lag(Expenditure.Absolute), Ed
  mutate(Ediff = ifelse(CountryLag == 2, (lead(Expenditure.Absolute, n = 1) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 3, (lead(Expenditure.Absolute, n = 2) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 4, (lead(Expenditure.Absolute, n = 3) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 5, (lead(Expenditure.Absolute, n = 4) - lag(Expenditure.Absolute)) / lag(Expenditure
  ungroup() %>%
  # Without exp diff, no log
  mutate(Compensation = (Ediff / (exp(Size - lag(Size, 1)))))


fd ← fd.compensation


Y.df ← fd.compensation %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

countryNotSwitzerland ← which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     #alpha[s,id.decade[y]] +
                     #alpha[s] +
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 10^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA  ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10  ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2  ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627  ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528  ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                                                                   alpha[s,y] +
##                         inprod(X[c,y-1,], theta[s,1:nV]) +
##                         inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                         rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                     alpha[s,1] +
##                     inprod(X[c,1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 10^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
```

```
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

6.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2304.RData"))
```

7

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. Cross-Lagged panel model with Fixed Effects (ML-SEM). w/o number of parties.

mm-tscs-2305

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

7.1 Covariates

VPI

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

vpi.original ← vpi

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


# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon
load(”data-parlgov/data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ⩾ year.start)

Y.df ← fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
```

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

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  # add binary variables
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 <- rep(0, (nV + nVS))
```

```
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,y] +
                     beta[s] * Y[s,c,y-1] +
                     gamma[s,c] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   gamma[s,c] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      gamma[s,c] ~ dt(0, 0.1^-2, 3)
    }
    beta[s] ~ dunif(-1, 1)
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste(”models/model-”, M, ”.bug”, sep = ””))

par ← NULL
par ← c(par, ”theta”)
par ← c(par, ”alpha”)
par ← c(par, ”nu”)
par ← c(par, ”Sigma”)
par ← c(par, ”rho”)
par ← c(par, ”lambda_c”)
par ← c(par, ”lambda”)
par ← c(par, ”beta”, ”gamma”)


sample.length ← c(”test”, ”middle”, ”serious”)
sample.length ← sample.length[3]
if (sample.length == ”test”) {
  chains ← 2
  adapt ← 1e2
  burnin ← 5e2
  run ← 5e2
  thin ← 1
} else if (sample.length == ”middle”) {
  chains ← 3
  adapt ← 2e2
  burnin ← 1e4
  run ← 2e3
  thin ← 1
} else if (sample.length == ”serious”) {
  chains ← 4
  adapt ← 5e2
  burnin ← 5e4
  run ← 2e3
  thin ← 10
}

method ← ”parallel”
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                          : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                         : int 15
##  $ nS                         : int 2
##  $ nY                         : int 16
##  $ countryNotSwitzerland      : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland         : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition    : int 1
##  $ id.decade                  : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                   : int 3
##  $ X                          : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                         : int 5
##  $ XS                         : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                        : int 6
##  $ b0                         : num [1:5] 0 0 0 0 0
##  $ B0                         : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
```

```
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,y] +
##                      beta[s] * Y[s,c,y-1] +
##                      gamma[s,c] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                    alpha[s,1] +
##                    gamma[s,c] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       gamma[s,c] ~ dt(0, 0.1^-2, 3)
##     }
##     beta[s] ~ dunif(-1, 1)
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average",
    "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2305.RData"))
```

8 Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/ environmental NGOs (by population). w/o parties in government.

mm-tscs-3301

```
load(”data-moneymouth-compensation.RData”) # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”Variable”, values_to = ”value”) %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == ”Lag: 1”)

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = ”sample_description.RData”)
```

8.1 Covariates

VPI

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

vpi.original <- vpi

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

# VPI
#   average of the 2 dimensions
#   top-down + bottom-up
#   only bottom-up
#   only top-down
```

- World Development Indicators: GDP per capita
- IMF: Debt
- Electoral competition
- EU
- ParlGov: number of parties in government, time horizon
- Henisz: Political constraints

```
# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed

# Environmental NGOs
load("data-engos/environmental_ngos.RData") # ngos
ngos <- mutate(ngos, Sector = "Environmental")


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  left_join(ngos) %>%
  filter(Year ⩾ year.start)

Y.df ← fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = ”Compensation”)

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

countryNotSwitzerland ← which(ifelse(country.label == ”Switzerland”, 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == ”Switzerland”, 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = ”value”)

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         ENGO_pop,
         Size) %>%
  unique() %>%
  mutate(`ENGOs by population (log)` = log(ENGO_pop)) %>%
  select(-ENGO_pop) %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  mutate(`ENGOs by population (log)` = ifelse(Sector ≠ "Environmental", 0, `ENGOs by population (log)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }

    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                          : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA  ...
##  $ nC                         : int 15
##  $ nS                         : int 2
##  $ nY                         : int 16
##  $ countryNotSwitzerland      : int [1:14] 1 2 3 4 5 6 7 8 9 10  ...
##  $ countrySwitzerland         : int 15
##  $ varNotElectoralCompetition: int [1:6] 2 3 4 5 6 7
##  $ varElectoralCompetition    : int 1
##  $ id.decade                  : num [1:16] 1 1 1 1 1 2 2 2 2 2  ...
##  $ nDecades                   : int 3
##  $ X                          : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627  ...
##  $ nV                    : int 5
##  $ XS                    : num [1:2, 1:15, 1:16, 1:7] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                   : int 7
##  $ b0                    : num [1:5] 0 0 0 0 0
##  $ B0                    : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,y] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
```

```
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

8.2 *Model interpretation*

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)", "ENGOs by population (log)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)", "ENGOs by population (log)",
    "Ideology, average",
    "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-3301.RData"))
```

9

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o parties in government.

mm-tscs-4301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

9.1 Covariates

VPI

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

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)

Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

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

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


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                       alpha[s,y] +
##                       inprod(X[c,y-1,], theta[s,1:nV]) +
##                       inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                       rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                     alpha[s,1] +
##                     inprod(X[c,1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

9.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average",
    "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))
#  filter(!(Sector == "Environmental" & Covariate %in% c("Unemployment", "Elderly")))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-4301.RData"))
```

10
Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. Salience from Eurobarometer.

mm-tscs-102301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

10.1 Covariates

VPI

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

```r
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated, from Eurobarometer
load("data-eurobarometer/data-eurobarometer.RData") # loads eb.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(eb.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)

Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)
varNotSalience <- which(ifelse(variable.sector.label == "Issue salience", 0, 1) == 1)
varSalience <- which(ifelse(variable.sector.label == "Issue salience", 1, 0) == 1)
avg.social.salience <- apply(XS[2,,,varSalience], 2, mean, na.rm = TRUE)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
```

```
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  varNotSalience = varNotSalience, varSalience = varSalience,
  avg.social.salience = avg.social.salience,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     #alpha[s,id.decade[y]] +
                     #alpha[s] +
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in varNotSalience) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
  for (v in varSalience) {
    for (c in 1:nC) {
      for (y in 1:nY) {
        XS[1,c,y,v] ~ dnorm(0.5, 0.25^-2) # Environmental sector
        XS[2,c,y,v] ~ dnorm(avg.social.salience[y] + 0.5, 0.25^-2) # Social sector
      }
    }
  }
}'
write(m, file = paste(”models/model-”, M, ”.bug”, sep = ””))

par <- NULL
par <- c(par, ”theta”)
par <- c(par, ”alpha”)
par <- c(par, ”nu”)
par <- c(par, ”Sigma”)
par <- c(par, ”rho”)
par <- c(par, ”lambda_c”)
par <- c(par, ”lambda”)

sample.length <- c(”test”, ”middle”, ”serious”)
sample.length <- sample.length[3]
if (sample.length == ”test”) {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == ”middle”) {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == ”serious”) {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- ”parallel”
```

Data passed:

```
str(D)
```

```
## List of 19
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ varNotSalience            : int [1:5] 1 3 4 5 6
##  $ varSalience               : int 2
##  $ avg.social.salience       : Named num [1:16] -0.212 -0.212 -0.212 -0.212 -0.212 ...
##   ..- attr(*, ”names”)= chr [1:16] ”1995” ”1996” ”1997” ”1998” ...
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] ←
##                                                                 alpha[s,y] +
##                       inprod(X[c,y-1,], theta[s,1:nV]) +
##                       inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                       rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] ← 1 / sigma.sq[s,c]
##       sigma.sq[s,c] ← exp(sigma[s,c])
##       sigma[s,c] ← lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] ←
##                     alpha[s,1] +
##                     inprod(X[c,1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] ← exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in varNotSalience) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
##   for (v in varSalience) {
##     for (c in 1:nC) {
##       for (y in 1:nY) {
##         XS[1,c,y,v] ~ dnorm(0.5, 0.25^-2)         XS[2,c,y,v] ~ dnorm(avg.social.salience[y] + 0.5, 0.25^-2)       }
##     }
##   }
## }
```

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

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

10.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-102301.RData"))
```

11

Explanatory model for compensation: Baseline. Lag 1. Country specific lags. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. Salience from Eurobarometer.

mm-tscs-102304

```
load(”data-moneymouth-compensation.RData”) # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”Variable”, values_to = ”value”) %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == ”Lag: 1”)

country.lags ← read.table(”highly-likely-lags-104.csv”, header = TRUE, sep = ”\t”) %>%
  tibble() %>%
  mutate(Country = factor(Country)) %>%
  # restrict to maximum 3
  mutate(Lag = ifelse(Lag > 3, 3, Lag)) %>%
  rename(CountryLag = Lag)

mmc ← left_join(mmc, country.lags)

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = ”sample_description.RData”)
```

11.1 Covariates

VPI

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

vpi.original ← vpi

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

# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-eurobarometer/data-eurobarometer.RData") # loads eb.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)
#  Year = year.start:year.finish)
#universe ← filter(universe, Country ≠ "Switzerland")

fd.pre ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
              group_by(Sector, Country, Year) %>%
              summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(eb.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ≥ year.start)


fd.compensation ← fd.pre %>%
  group_by(Sector, Country) %>%
  arrange(Sector, Country, Year) %>%
  mutate(Ediff = NA) %>%
  mutate(Ediff = ifelse(CountryLag == 1, (Expenditure.Absolute - lag(Expenditure.Absolute)) / lag(Expenditure.Absolute), E
  mutate(Ediff = ifelse(CountryLag == 2, (lead(Expenditure.Absolute, n = 1) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 3, (lead(Expenditure.Absolute, n = 2) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 4, (lead(Expenditure.Absolute, n = 3) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 5, (lead(Expenditure.Absolute, n = 4) - lag(Expenditure.Absolute)) / lag(Expenditure
  #select(Country, Year, CountryLag, Expenditure.Absolute, Ediff)
  ungroup() %>%
  # With exp diff
  #mutate(Compensation = log(exp(Ediff) / (exp(Size - lag(Size, 1)))))
  # Without exp diff
  #mutate(Compensation = log(Ediff / (exp(Size - lag(Size, 1))))) %>%
  #mutate(Compensation = ifelse(is.nan(Compensation), 0, Compensation))
  # Without exp diff, no log
  mutate(Compensation = (Ediff / (exp(Size - lag(Size, 1)))))


fd ← fd.compensation


Y.df ← fd.compensation %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

countryNotSwitzerland ← which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

variable.label ← dimnames(X)[[3]]
nV ← length(variable.label)
```

```
XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)
varNotSalience <- which(ifelse(variable.sector.label == "Issue salience", 0, 1) == 1)
varSalience <- which(ifelse(variable.sector.label == "Issue salience", 1, 0) == 1)
avg.social.salience <- apply(XS[2,,,varSalience], 2, mean, na.rm = TRUE)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  varNotSalience = varNotSalience, varSalience = varSalience,
  avg.social.salience = avg.social.salience,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                    alpha[s,y] +
                    inprod(X[c,y-1,], theta[s,1:nV]) +
                    inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                    rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                  alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 10^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in varNotSalience) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
  for (v in varSalience) {
    for (c in 1:nC) {
      for (y in 1:nY) {
        XS[1,c,y,v] ~ dnorm(0.5, 0.25^-2) # Environmental sector
        XS[2,c,y,v] ~ dnorm(avg.social.salience[y] + 0.5, 0.25^-2) # Social sector
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 19
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ varNotSalience            : int [1:5] 1 3 4 5 6
##  $ varSalience               : int 2
##  $ avg.social.salience       : Named num [1:16] -0.212 -0.212 -0.212 -0.212 -0.212 ...
##   ..- attr(*, "names")= chr [1:16] "1995" "1996" "1997" "1998" ...
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,y] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
```

```
##       theta[s,t] ~ dnorm(0, 10^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in varNotSalience) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
##   for (v in varSalience) {
##     for (c in 1:nC) {
##       for (y in 1:nY) {
##         XS[1,c,y,v] ~ dnorm(0.5, 0.25^-2)          XS[2,c,y,v] ~ dnorm(avg.social.salience[y] + 0.5, 0.25^-2)         }
##     }
##   }
## }
```

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

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

11.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-102304.RData"))
```

12 Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. Cross-Lagged panel model with Fixed Effects (ML-SEM). w/o number of parties. Salience from Eurobarometer.

mm-tscs-102305

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

12.1 Covariates

VPI

```
load("data-vpi/vpi_wgi.RData")
vpi.implementation <- "No model, simple addition"
vpi <- vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
```

```r
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-eurobarometer/data-eurobarometer.RData") # loads eb.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(eb.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)

Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
```

```r
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  # add binary variables
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)
varNotSalience <- which(ifelse(variable.sector.label == "Issue salience", 0, 1) == 1)
varSalience <- which(ifelse(variable.sector.label == "Issue salience", 1, 0) == 1)
```

```
avg.social.salience <- apply(XS[2,,,varSalience], 2, mean, na.rm = TRUE)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  varNotSalience = varNotSalience, varSalience = varSalience,
  avg.social.salience = avg.social.salience,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     alpha[s,y] +
                     beta[s] * Y[s,c,y-1] +
                     gamma[s,c] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   gamma[s,c] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      gamma[s,c] ~ dt(0, 0.1^-2, 3)
    }
    beta[s] ~ dunif(-1, 1)
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in varNotSalience) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
  for (v in varSalience) {
    for (c in 1:nC) {
      for (y in 1:nY) {
        XS[1,c,y,v] ~ dnorm(0.5, 0.25^-2) # Environmental sector
        XS[2,c,y,v] ~ dnorm(avg.social.salience[y] + 0.5, 0.25^-2) # Social sector
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "nu")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda_c")
par <- c(par, "lambda")
par <- c(par, "beta", "gamma")

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 19
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ varNotSalience            : int [1:5] 1 3 4 5 6
##  $ varSalience               : int 2
##  $ avg.social.salience       : Named num [1:16] -0.212 -0.212 -0.212 -0.212 -0.212 ...
##   ..- attr(*, "names")= chr [1:16] "1995" "1996" "1997" "1998" ...
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] ←
##                     alpha[s,y] +
##                     beta[s] * Y[s,c,y-1] +
##                     gamma[s,c] +
##                     inprod(X[c,y-1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] ← 1 / sigma.sq[s,c]
##       sigma.sq[s,c] ← exp(sigma[s,c])
##       sigma[s,c] ← lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] ←
##                   alpha[s,1] +
##                   gamma[s,c] +
##                   inprod(X[c,1,], theta[s,1:nV]) +
##                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       gamma[s,c] ~ dt(0, 0.1^-2, 3)
##     }
##     beta[s] ~ dunif(-1, 1)
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] ← exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in varNotSalience) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
##   for (v in varSalience) {
##     for (c in 1:nC) {
##       for (y in 1:nY) {
##         XS[1,c,y,v] ~ dnorm(0.5, 0.25^-2)         XS[2,c,y,v] ~ dnorm(avg.social.salience[y] + 0.5, 0.25^-2)       }
##     }
##   }
## }
```

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

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

12.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average",
    "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0(”ci-theta-102305.RData”))
```

13
Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/ environmental NGOs (by population). w/o parties in government. Salience from Eurobarometer.

mm-tscs-103301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

13.1 Covariates

VPI

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

vpi.original ← vpi

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

# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon
load(”data-parlgov/data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated, from Eurobarometer
load(”data-eurobarometer/data-eurobarometer.RData”) # loads eb.country.interpolation.annual

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed

# Environmental NGOs
load(”data-engos/environmental_ngos.RData”) # ngos
ngos ← mutate(ngos, Sector = ”Environmental”)


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(eb.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  left_join(ngos) %>%
  filter(Year ⩾ year.start)

# Correlation between ENGOs and Salience
```

```
corr.sal.engos <- fd %>%
  filter(Sector == "Environmental") %>%
  mutate(`ENGOs by population (log)` = log(ENGO_pop)) %>%
  select(Country, Salience, `ENGOs by population (log)`) %>%
  group_by(Country) %>%
  summarize(`Salience, country average (Eurobarometer)` = mean(Salience, na.rm = TRUE),
            `ENGOs by population (log), country average` = mean(`ENGOs by population (log)`, na.rm = TRUE)) %>%
  ungroup()

corr.sal.engos %>%
  ggplot(aes(x = `Salience, country average (Eurobarometer)`,
             y = `ENGOs by population (log), country average`,
             label = Country)) +
  geom_point() +
  ggpubr::stat_cor(aes(label = ..r.label..)) +
  geom_text_repel()
```

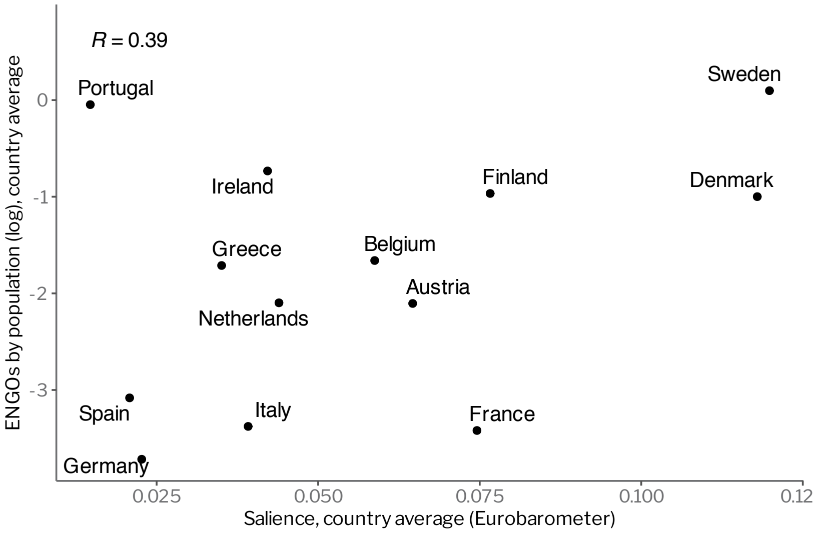


Figure 13.1: Correlation between country averages of ENGOs by population and Salience.

```
fd %>%
  filter(Sector == "Environmental") %>%
  mutate(`ENGOs by population (log)` = log(ENGO_pop)) %>%
  select(Country, Salience, `ENGOs by population (log)`) %>%
  summarize(`Correlation Salience (Eurobarometer) vs. ENGOs by population (log)` =
              cor(Salience, `ENGOs by population (log)`, use = "pairwise.complete.obs")) %>%
  mykbl("Correlation Salience / ENGOs.")
```

Table 13.1: Correlation Salience / ENGOs.

Correlation Salience (Eurobarometer) vs. ENGOs by population (log)
0.31

14

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. w/ combined Veto Players and opposition divergence. w/ VPI original component 6, organization.

mm-tscs-15692301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

14.1 Covariates

VPI

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

vpi.original ← vpi

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

# VPI
#   average of the 2 dimensions
#   top-down + bottom-up
#   only bottom-up
#   only top-down
# Bottom-up: Articulation, Consultation, Evaluation
# Top-down: Accountability, Resources, Organization
vpi ← vpi.es %>%
  filter(Variable == ”Organisation”) %>%
  select(Sector, Country, Year, Organisation = value) %>%
  distinct()


# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon and opposition
load(”data-parlgov/230918-data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)



fd <- fd %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))



Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    #`Divergent opposition (SD LeftRight)`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         Organisation,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ ”Environmental”, 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = ”value”)

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == ”Electoral competition”, 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == ”Electoral competition”, 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 2 3 4 5 6 7
##  $ varElectoralCompetition   : int 1
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                         alpha[s,y] +
##                         inprod(X[c,y-1,], theta[s,1:nV]) +
##                         inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                         rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                       alpha[s,1] +
##                       inprod(X[c,1,], theta[s,1:nV]) +
##                       inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

14.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Blame avoidance opportunities",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-15692301.RData"))
```

15 Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. w/ administrative reform.

mm-tscs-141212301

```
load(”data-moneymouth-compensation.RData”) # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”Variable”, values_to = ”value”) %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == ”Lag: 1”)

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = ”sample_description.RData”)
```

15.1 Covariates

VPI

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

vpi.original ← vpi

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


# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon and opposition
load(”data-parlgov/230918-data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Regional authority
load(”data-regional_authority/data-rai.RData”) #  loads rai

# Administrative reforms
load(”data-administrative_reforms/data-administrative_reforms.RData”) # admin.reforms

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(rai) %>%
  left_join(policy.cost) %>%
  left_join(admin.reforms) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ⩾ year.start)
```

```
fd <- fd %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))



Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

year.group <- cut(year.label.numeric, 8)
year.group.label <- levels(year.group)
year.group <- as.numeric(year.group)
nYG <- length(year.group.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`,
    #`Divergent opposition (SD LeftRight)`,
    #`Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
    `Administrative coordination reforms (n)`,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  year.group = year.group, nYG = nYG,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits ← function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    alpha = array(rep(0, nS *nYG), dim = c(nS, nYG)),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.1), dim = c(nS, (nV + nVS)))
  )
}


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,year.group[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nYG) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
    }
    alpha[s,1] ~ dnorm(0, 5^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 2
  adapt <- 5e2
  burnin <- 6e4
  run <- 2e3
  thin <- 5
}

method <- "parallel"
```

15.2 *Descriptive illustration of portfolio and expenditure growth*

Identify capacity increases not related to policy growth.

```
fd %>%
  group_by(Sector, Country) %>%
  arrange(Sector, Country, Year) %>%
  mutate(diff.expenditure = (Expenditure.Absolute - lag(Expenditure.Absolute)) /
         lag(Expenditure.Absolute)) %>%
  mutate(size.diff = Size - lag(Size)) %>%
  ungroup() %>%
  filter(Sector == "Social") %>%
  ggplot(aes(x = Year, y = diff.expenditure)) +
  geom_line(aes(y = diff.expenditure)) +
  geom_point(data = . %>% filter(size.diff != 0), alpha = 0.5,
             aes(y = min(diff.expenditure, na.rm = TRUE), size = size.diff)) +
  facet_wrap(~ Country, scales = "free") +
  scale_x_continuous(breaks = seq(1980, 2010, 10)) +
  theme(legend.position = "bottom") +
  scale_alpha(guide = 'none')
```

Data passed:

```
str(D)
```

```
## List of 18
```

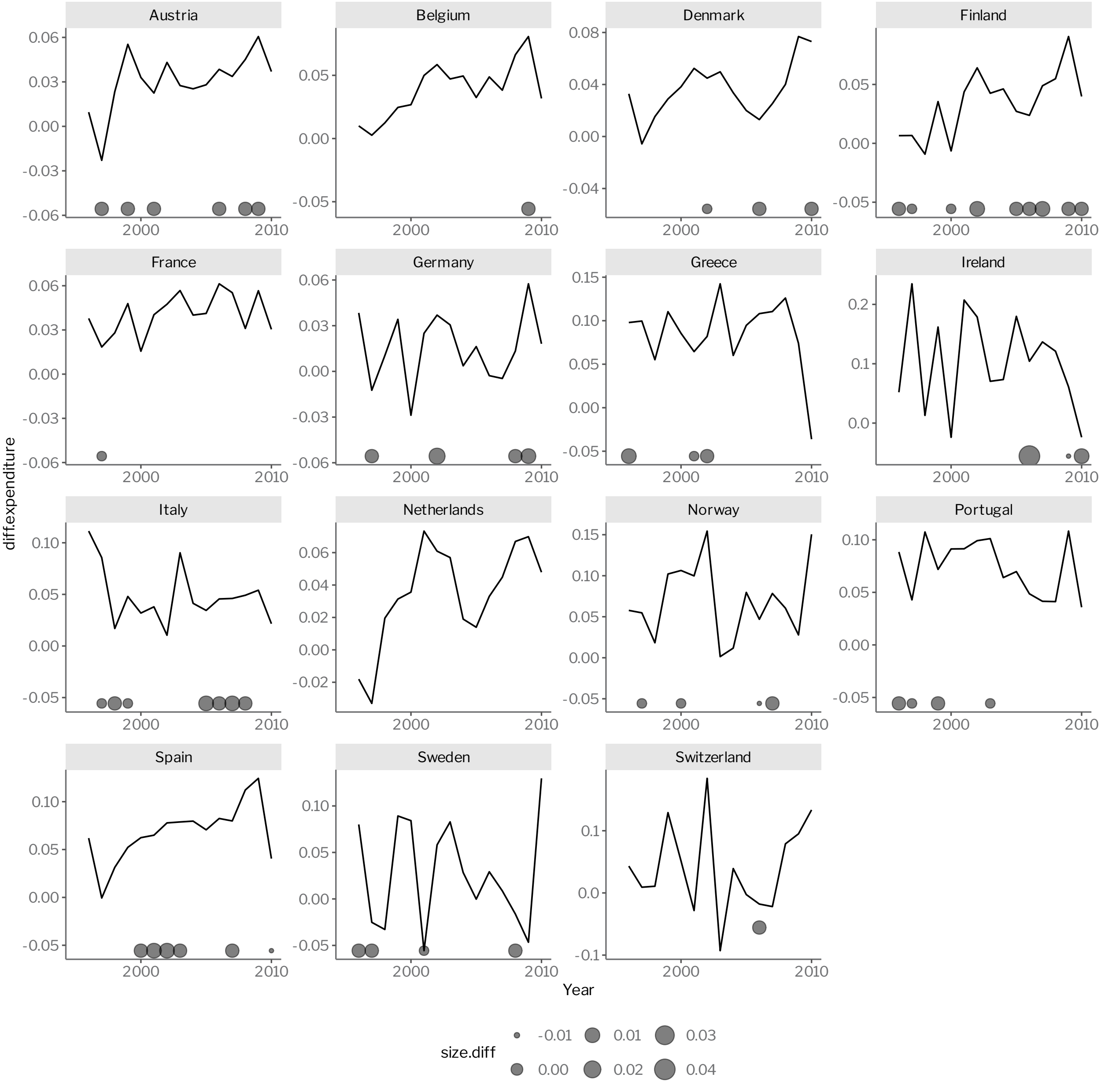


Figure 15.1: Temporal evolution of diff expenditure with policy growth size (dots).

```
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ year.group                : num [1:16] 1 1 2 2 3 3 4 4 5 5 ...
##  $ nYG                       : int 8
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 1 3 4 5 6 7
##  $ varElectoralCompetition   : int 2
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] 0.654 -0.226 0.654 -0.226 NA ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] ←
##                      alpha[s,year.group[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] ← 1 / sigma.sq[s,c]
##       sigma.sq[s,c] ← exp(sigma[s,c])
##       sigma[s,c] ← lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] ←
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nYG) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 5^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

15.3 Model

Model equation:

$$
\begin{aligned}
Y_{c,y} \sim &\quad \mathcal{N}(\mu_{c,y}, \sigma_c) && \text{Main data component} \\
\mu_{c,y} = &\quad \alpha_y + \theta_v * X_{c,y-1,v} + \rho_c * (Y_{c,y-1} - \mu_{c,y-1}) && \text{Main linear model} \\
\sigma_c = &\quad exp(\lambda 0 + \lambda_c && \text{Error component, clustered} \\
\theta_v \sim &\quad \mathcal{N}(0, 5) && \text{Priors for explanatory variables} \\
\alpha_y \sim &\quad \mathcal{N}(\alpha_{y-1}, 0.1) && \text{Priors for the temporal dynamics} \\
\rho_c \sim &\quad \mathcal{U}(-1, 1) && \text{Priors for the auto-regressive component}
\end{aligned}
$$

Where:

- c: Country
- y: Year
- v: Covariate
- $y_{c,y}$: Continuous variable with the compensation for a specific country (c) and year (y).

- $X_{c,y,v}$ Matrix with the explanatory values for each covariate (v).
- λ_c: Countrys-specific clustered errors.
- θ: Main effects of interests.
- α_y: Varying intercepts for temporal dynamics, specified as a Kalman filter.
- $\rho_{s,c}$: Auto regressive component of order 1.

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

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

15.4 Model interpretation

Auto-regressive component.

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

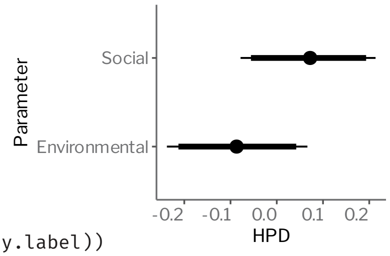


Figure 15.2: Auto-regressive components.

Variance components (country varying errors).

```
L.lambda.c ← plab("lambda_c", list(Sector = sector.label, Country = country.label))
S.lambda.c ← ggs(s, family = "lambda_c\\[", par_labels = L.lambda.c) %>%
  filter(!str_detect(Country, "^Z-"))
ggs_caterpillar(S.lambda.c, label = "Country") +
  facet_grid(~ Sector) +
  ggtitle("Variance component (countries)")
```

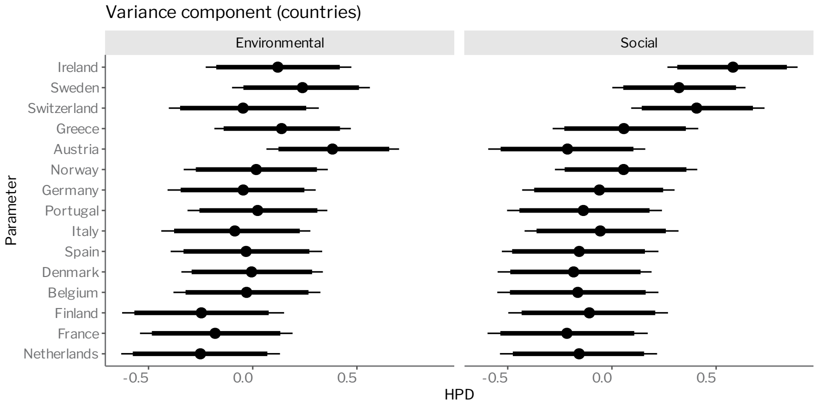


Figure 15.3: Variance component (country varying errors).

```
L.alpha ← plab("alpha", list(Sector = sector.label,
                               Year = year.label))
S.alpha ← ggs(s, family = "^alpha", par_labels = L.alpha)

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

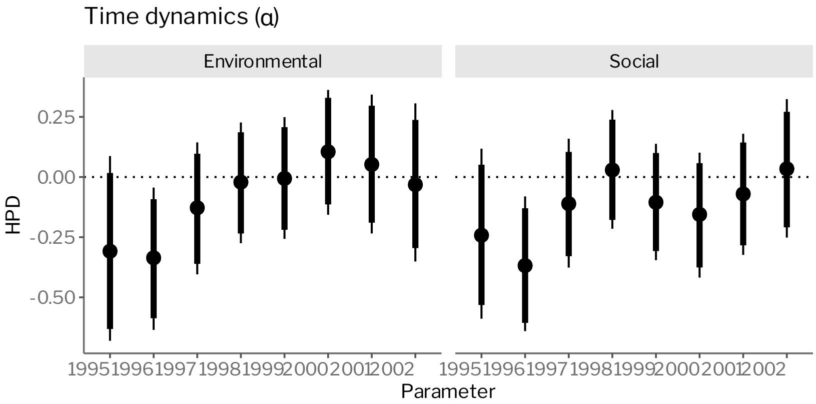


Figure 15.4: Temporal dynamics.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = cov.rename.reorder(Covariate))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-141212301.RData"))

S.theta %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, "). Reference.", sep = "")))
```

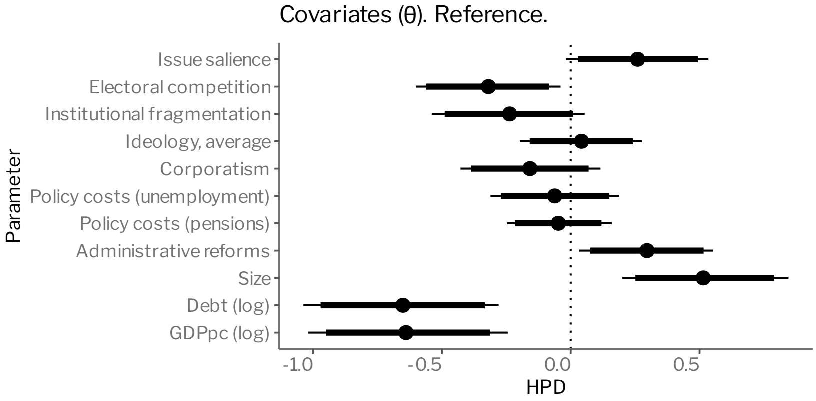


Figure 15.5: Covariates. Social.

16

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. Cross-Lagged panel model with Fixed Effects (ML-SEM). w/o number of parties. w/ administrative reforms.

mm-tscs-151212301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

16.1 Covariates

VPI

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

vpi.original ← vpi

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

# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon and opposition
load(”data-parlgov/230918-data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Regional authority
load(”data-regional_authority/data-rai.RData”) #  loads rai

# Administrative reforms
load(”data-administrative_reforms/data-administrative_reforms.RData”) # admin.reforms

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(rai) %>%
  left_join(policy.cost) %>%
  left_join(admin.reforms) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ⩾ year.start)



fd ← fd %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))



Y.df ← fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = ”Compensation”)

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

year.group ← cut(year.label.numeric, 8)
year.group.label ← levels(year.group)
year.group ← as.numeric(year.group)
nYG ← length(year.group.label)

countryNotSwitzerland ← which(ifelse(country.label == ”Switzerland”, 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == ”Switzerland”, 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = ”value”)

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
    `Administrative coordination reforms (n)`,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  year.group = year.group, nYG = nYG,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits ← function() {
  theta.pre = array(rnorm(nS * (nV + nVS), 0, 0.01), dim = c(nS, (nV + nVS)))
  theta.pre[2,7] ← 0.5
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    alpha = array(rep(0, nS *nYG), dim = c(nS, nYG)),
    theta = theta.pre
  )
}


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,year.group[y]] +
                     beta[s] * Y[s,c,y-1] +
                     gamma[s,c] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   gamma[s,c] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      gamma[s,c] ~ dt(0, 0.1^-2, 3)
    }
    beta[s] ~ dunif(-1, 1)
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 10^-2)
    }
    for (y in 2:nYG) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
    }
    alpha[s,1] ~ dnorm(0, 5^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par ← NULL
par ← c(par, "theta")
par ← c(par, "alpha")
par ← c(par, "nu")
par ← c(par, "Sigma")
par ← c(par, "rho")
par ← c(par, "lambda_c")
par ← c(par, "lambda")
par ← c(par, "beta", "gamma")

sample.length ← c("test", "middle", "serious", "brutal")
sample.length ← sample.length[3]
if (sample.length == "test") {
  chains ← 2
  adapt ← 1e2
  burnin ← 5e2
  run ← 5e2
  thin ← 1
} else if (sample.length == "middle") {
  chains ← 3
  adapt ← 2e2
  burnin ← 1e4
  run ← 2e3
  thin ← 1
} else if (sample.length == "serious") {
  chains ← 4
  adapt ← 5e2
  burnin ← 5e4
  #burnin ← 55e3
  run ← 2e3
  thin ← 10
  #thin ← 2
} else if (sample.length == "brutal") {
  chains ← 4
  adapt ← 5e2
  burnin ← 2e5
  run ← 2e3
  thin ← 50
}

method ← "parallel"
```

Data passed:

```
str(D)
```

```
## List of 18
## $ Y                        : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
## $ nC                       : int 15
## $ nS                       : int 2
##  $ nY                        : int 16
##  $ year.group                : num [1:16] 1 1 2 2 3 3 4 4 5 5 ...
##  $ nYG                       : int 8
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 1 3 4 5 6 7
##  $ varElectoralCompetition   : int 2
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] 0.654 -0.226 0.654 -0.226 NA ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] ←
##                      alpha[s,year.group[y]] +
##                      beta[s] * Y[s,c,y-1] +
##                      gamma[s,c] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] ← 1 / sigma.sq[s,c]
##       sigma.sq[s,c] ← exp(sigma[s,c])
##       sigma[s,c] ← lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] ←
##                    alpha[s,1] +
##                    gamma[s,c] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       gamma[s,c] ~ dt(0, 0.1^-2, 3)
##     }
##     beta[s] ~ dunif(-1, 1)
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
```

```
##       theta[s,t] ~ dnorm(0, 10^-2)
##     }
##     for (y in 2:nYG) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 5^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

16.2 Model interpretation

```
f1 <- S.rho %>%
  filter(Sector == "Social") %>%
  mutate(Parameter = "Auto-correlation") %>%
  ggs_caterpillar() +
  ggtitle(expression(paste("Auto-correlation (", rho, ")", sep = "")))

f2 <- S.alpha %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Year", sort = FALSE) +
  coord_flip() +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Time dynamics (", alpha, ")", sep = "")))

f3 <- S.lambda.c %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Country") +
  ggtitle(expression(paste("Variance component, countries (", lambda, ")", sep = "")))


plot_grid(plot_grid(f1, f2, ncol = 1, rel_heights = c(0.2, 0.8)), f3, ncol = 2)
```

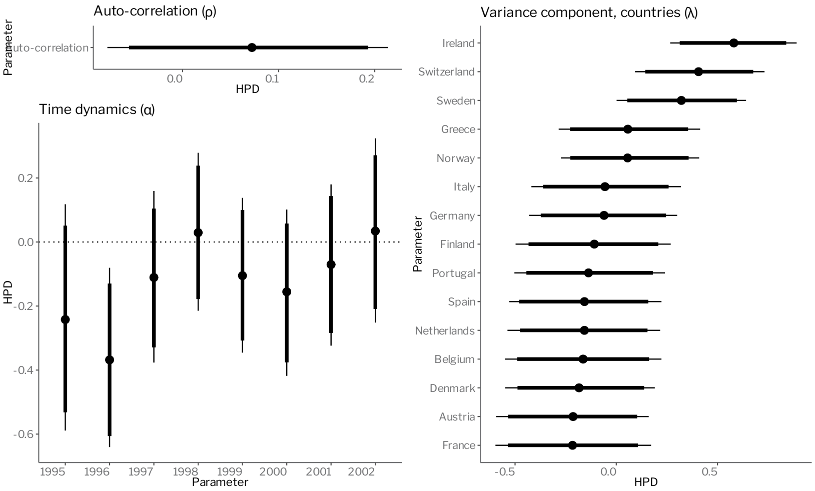


Figure 16.1: Social sector. Auxiliary parameters.

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = cov.rename.reorder(Covariate))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-151212301.RData"))

S.theta %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, "). ML-SEM.", sep = "")))
```

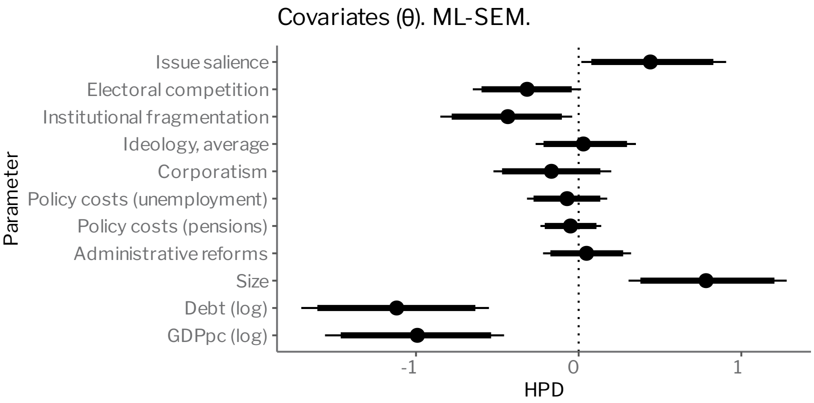


Figure 16.2: Covariates. Social.

17

Explanatory model for compensation: Baseline. Lag 1. Country specific lags. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. w/ administrative reforms.

mm-tscs-161212301

```r
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

country.lags <- read.table("highly-likely-lags-104.csv", header = TRUE, sep = "\t") %>%
  tibble() %>%
  mutate(Country = factor(Country)) %>%
  # restrict to maximum 3
  mutate(Lag = ifelse(Lag > 3, 3, Lag)) %>%
  rename(CountryLag = Lag)

mmc <- left_join(mmc, country.lags)
####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

17.1 Covariates

VPI

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

vpi.original <- vpi

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

# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon and opposition
load("data-parlgov/230918-data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Regional authority
load("data-regional_authority/data-rai.RData") #  loads rai

# Administrative reforms
load("data-administrative_reforms/data-administrative_reforms.RData") # admin.reforms

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd.pre <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(rai) %>%
  left_join(policy.cost) %>%
  left_join(admin.reforms) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)



fd.pre <- fd.pre %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))

fd.compensation <- fd.pre %>%
  group_by(Sector, Country) %>%
  arrange(Sector, Country, Year) %>%
  mutate(Ediff = NA) %>%
  mutate(Ediff = ifelse(CountryLag == 1, (Expenditure.Absolute - lag(Expenditure.Absolute)) / lag(Expenditure.Absolute), E
  mutate(Ediff = ifelse(CountryLag == 2, (lead(Expenditure.Absolute, n = 1) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 3, (lead(Expenditure.Absolute, n = 2) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 4, (lead(Expenditure.Absolute, n = 3) - lag(Expenditure.Absolute)) / lag(Expenditure
  mutate(Ediff = ifelse(CountryLag == 5, (lead(Expenditure.Absolute, n = 4) - lag(Expenditure.Absolute)) / lag(Expenditure
  ungroup() %>%
  mutate(Compensation = (Ediff / (exp(Size - lag(Size, 1)))))


fd <- fd.compensation


Y.df <- fd.compensation %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

year.group <- cut(year.label.numeric, 8)
year.group.label <- levels(year.group)
year.group <- as.numeric(year.group)
nYG <- length(year.group.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
    `Administrative coordination reforms (n)`,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  year.group = year.group, nYG = nYG,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    alpha = array(rep(0, nS *nYG), dim = c(nS, nYG)),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.1), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     alpha[s,year.group[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 10^-2)
    }
    for (y in 2:nYG) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
    }
    alpha[s,1] ~ dnorm(0, 5^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious", "brutal")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
} else if (sample.length == "brutal") {
  chains <- 4
  adapt <- 5e2
  burnin <- 2e5
  run <- 2e3
  thin <- 50
}
```

```
method ← ”parallel”
```

Data passed:

```
str(D)
```

```
## List of 18
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA  ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ year.group                : num [1:16] 1 1 2 2 3 3 4 4 5 5  ...
##  $ nYG                       : int 8
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10  ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 1 3 4 5 6 7
##  $ varElectoralCompetition   : int 2
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2  ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627  ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] 0.654 -0.226 0.654 -0.226 NA  ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0  ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] ←
##                      alpha[s,year.group[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] ← 1 / sigma.sq[s,c]
##       sigma.sq[s,c] ← exp(sigma[s,c])
##       sigma[s,c] ← lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] ←
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 10^-2)
##     }
##     for (y in 2:nYG) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 5^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
t0 <- proc.time()
set.seed(14719)
rj <- run.jags(model = paste("models/model-", M, ".bug", sep = ""),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = "glm",
               n.chains = chains,
               adapt = adapt,
               burnin = burnin, sample = run,
               thin = thin,
               monitor = par, method = method, summarise = FALSE)
s <- as.mcmc.list(rj)
proc.time() - t0
save(s, file = paste("samples-", M, ".RData", sep = ""))
```

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

17.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                                  variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = cov.rename.reorder(Covariate))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-161212301.RData"))
```

```
S.theta %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, "). Country-specific lags.", sep = "")))
```

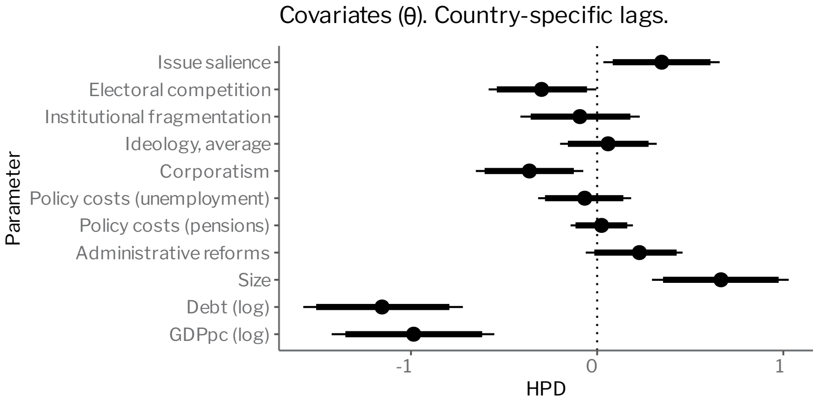


Figure 17.1: Covariates. Social.

18
Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. w/ administrative reform. w/ RAs managerial autonomy. w/ ideological difference with next government.

mm-tscs-171212301

```
load(”data-moneymouth-compensation.RData”) # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”Variable”, values_to = ”value”) %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == ”Lag: 1”)

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = ”sample_description.RData”)
```

18.1 Covariates

VPI

```
load(”data-vpi/vpi_wgi.RData”)
vpi.implementation ← ”No model, simple addition”
```

```
vpi ← vpi %>%
  #
  filter(!Country %in% c(”Mexico”, ”Turkey”)) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == ”Implementation costs”, ”Top-down (VPI)”, Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == ”Policy feedback”, ”Bottom-up (VPI)”, Dimension))

vpi.original ← vpi

vpi ← vpi %>%
  group_by(Sector, Country, Dimension) %>%
  arrange(Sector, Country, Dimension, Year) %>%
  mutate(VPI = ifelse(Year ≥ 1978, zoo::rollmean(VPI, k = 3, fill = NA, align = ”right”), VPI)) %>%
  ungroup()
```

- World Development Indicators: GDP per capita
- IMF: Debt
- Electoral competition
- EU
- ParlGov: number of parties in government, time horizon
- Henisz: Political constraints

```
# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon and opposition
load(”data-parlgov/230918-data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Regulatory agencies
load(”data-regulatory_agencies/data-regulatory_agencies.RData”) # loads ra

# Regional authority
load(”data-regional_authority/data-rai.RData”) #  loads rai

# Administrative reforms
load(”data-administrative_reforms/data-administrative_reforms.RData”) # admin.reforms

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)
```

```
fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(rai) %>%
  left_join(policy.cost) %>%
  left_join(admin.reforms) %>%
  left_join(ra) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)



fd <- fd %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))



Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

year.group <- cut(year.label.numeric, 8)
year.group.label <- levels(year.group)
year.group <- as.numeric(year.group)
nYG <- length(year.group.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`,
    `Ideological difference with next government, absolute`,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
    `Administrative coordination reforms (n)`,
         `Managerial autonomy`,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  year.group = year.group, nYG = nYG,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits ← function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    alpha = array(rep(0, nS *nYG), dim = c(nS, nYG)),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.1), dim = c(nS, (nV + nVS)))
    #theta = array(rnorm(nS * (nV), 0, 0.5), dim = c(nS, (nV)))
  )
}


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,year.group[y]] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nYG) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
    }
    alpha[s,1] ~ dnorm(0, 5^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious", "brutal")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 5
} else if (sample.length == "brutal") {
  chains <- 4
  adapt <- 5e2
  burnin <- 1e5
  run <- 2e3
  thin <- 100
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 18
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ year.group                : num [1:16] 1 1 2 2 3 3 4 4 5 5 ...
##  $ nYG                       : int 8
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:7] 1 3 4 5 6 7 8
##  $ varElectoralCompetition   : int 2
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:6] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 6
##  $ XS                        : num [1:2, 1:15, 1:16, 1:8] 0.654 -0.226 0.654 -0.226 NA ...
##  $ nVS                       : int 8
##  $ b0                        : num [1:6] 0 0 0 0 0 0
##  $ B0                        : num [1:6, 1:6] 1 0 0 0 0 0 0 1 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,year.group[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nYG) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 5^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

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

18.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = cov.rename.reorder(Covariate))

ci.theta ← ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-171212301.RData"))

S.theta %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, "). Extended.", sep = "")))
```

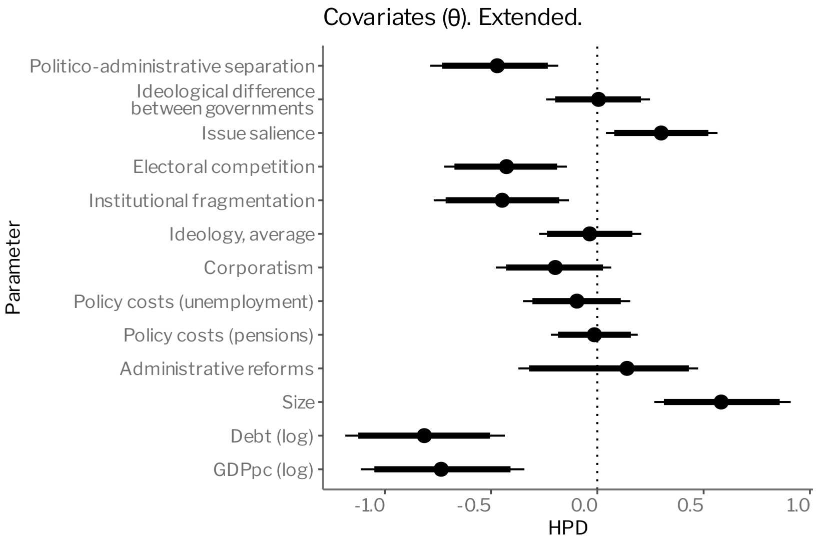


Figure 18.1: Covariates. Social.

19

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. w/ administrative reform.

mm-tscs-2r141212301

```
load(”data-moneymouth-compensation.RData”) # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”Variable”, values_to = ”value”) %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == ”Lag: 1”)

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = ”sample_description.RData”)
```

19.1 Covariates

VPI

```
load(”data-vpi/vpi_wgi.RData”)
vpi.implementation ← ”No model, simple addition”
vpi ← vpi %>%
  #
  filter(!Country %in% c(”Mexico”, ”Turkey”)) %>%
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == ”Implementation costs”, ”Top-down (VPI)”, Dimension)) %>%
```

```r
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon and opposition
load("data-parlgov/230918-data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Regional authority
load("data-regional_authority/data-rai.RData") #  loads rai

# Administrative reforms
load("data-administrative_reforms/data-administrative_reforms.RData") # admin.reforms

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(rai) %>%
  left_join(policy.cost) %>%
  left_join(admin.reforms) %>%
  left_join(m.country.interpolation.annual %>%
            filter(Sector %in% c("Environmental", "Social")) %>%
            mutate(Sector = factor(Sector)) %>%
            select(Country, Sector, Year, Salience) ) %>%
  left_join(m.country.interpolation.annual %>%
            filter(Sector %in% c("Environmental", "Social345minus")) %>%
            mutate(Sector = factor(ifelse(Sector == "Social345minus", "Social", Sector))) %>%
            select(Country, Sector, Year, SalienceGovWseats) %>%
            rename(SalienceGov = SalienceGovWseats) )  %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ⩾ year.start)



fd ← fd %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))



Y.df ← fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

year.group ← cut(year.label.numeric, 8)
year.group.label ← levels(year.group)
year.group ← as.numeric(year.group)
nYG ← length(year.group.label)

countryNotSwitzerland ← which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience, #SalienceGov,
    `Administrative coordination reforms (n)`,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  year.group = year.group, nYG = nYG,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits ← function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    alpha = array(rep(0, nS *nYG), dim = c(nS, nYG)),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.1), dim = c(nS, (nV + nVS)))
  )
}


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                    alpha[s,year.group[y]] +
                    inprod(X[c,y-1,], theta[s,1:nV]) +
                    inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                    rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                  alpha[s,1] +
                  inprod(X[c,1,], theta[s,1:nV]) +
                  inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nYG) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
    }
    alpha[s,1] ~ dnorm(0, 5^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par ← NULL
par ← c(par, "theta")
par ← c(par, "alpha")
par ← c(par, "nu")
par ← c(par, "Sigma")
par ← c(par, "rho")
par ← c(par, "lambda_c")
par ← c(par, "lambda")


sample.length ← c("test", "middle", "serious")
sample.length ← sample.length[3]
if (sample.length == "test") {
  chains ← 2
  adapt ← 1e2
  burnin ← 5e2
  run ← 5e2
  thin ← 1
} else if (sample.length == "middle") {
  chains ← 3
  adapt ← 2e2
  burnin ← 1e4
  run ← 2e3
  thin ← 1
} else if (sample.length == "serious") {
  chains ← 2
  adapt ← 5e2
  burnin ← 6e4
  run ← 2e3
  thin ← 5
}

method ← "parallel"
```

Data passed:

```
str(D)
```

```
## List of 18
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ year.group                : num [1:16] 1 1 2 2 3 3 4 4 5 5 ...
##  $ nYG                       : int 8
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 1 3 4 5 6 7
##  $ varElectoralCompetition   : int 2
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2  ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627  ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] 0.654 -0.226 0.654 -0.226 NA  ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0  ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,year.group[y]] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nYG) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 5^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }
```

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

19.2 *Model interpretation*

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = cov.rename.reorder(Covariate))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2r141212301.RData"))

S.theta %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, "). Reference.", sep = "")))


fd %>%
  filter(Sector == "Social") %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  ggplot(aes(x = Year, y = `Institutional fragmentation`)) +
  geom_line() +
  facet_wrap(~ Country, ncol = 5) +
  expand_limits(y = c(0, 1)) +
  ggtitle("Institutional fragmentation")
```

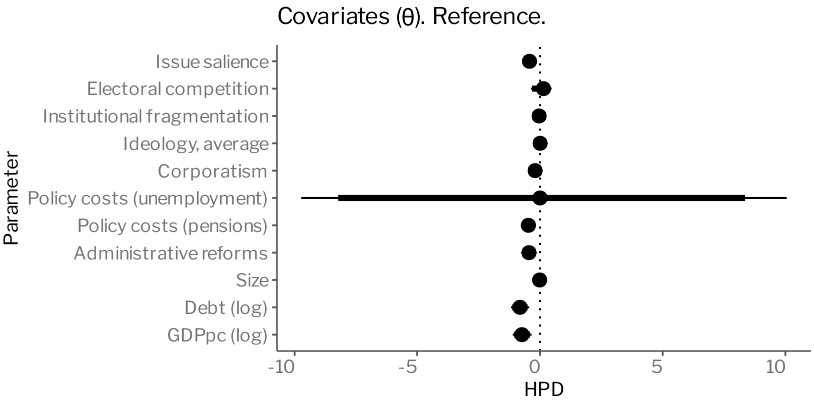


Figure 19.1: Covariates. Social.

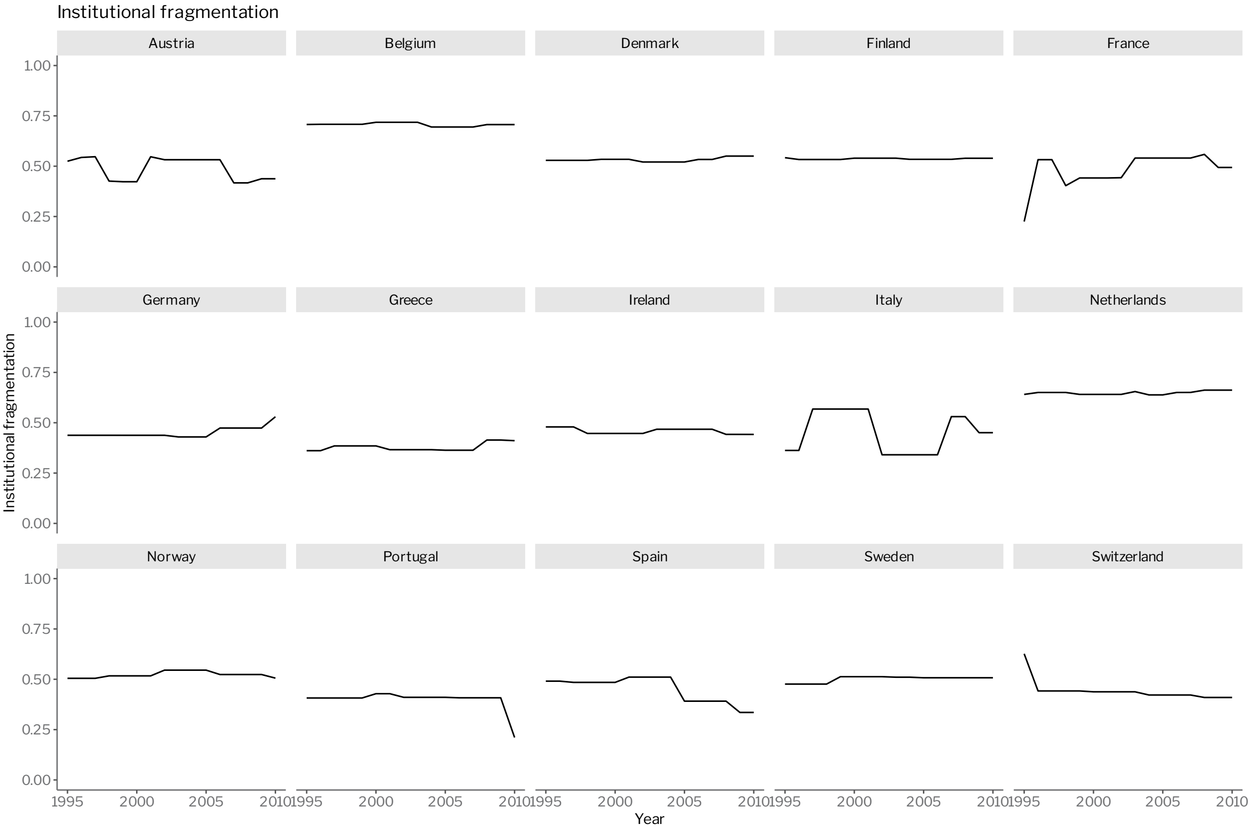


Figure 19.2: Institutional fragmentation.

```
fd %>%
  filter(Sector == "Social") %>%
  rename(
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  select(Country, Year, starts_with("Policy costs")) %>%
  pivot_longer(-c(Country, Year), names_to = "Policy", values_to = "Cost (Diff)") %>%
  ggplot(aes(x = Year, y = `Cost (Diff)`, color = Policy)) +
  geom_line() +
  facet_wrap(~ Country, ncol = 5) +
  geom_hline(yintercept = 0, lty = 3) +
  ggtitle("Policy costs.") +
  scale_color_manual(values = c("black", "grey50")) +
  theme(legend.position = "bottom")
```

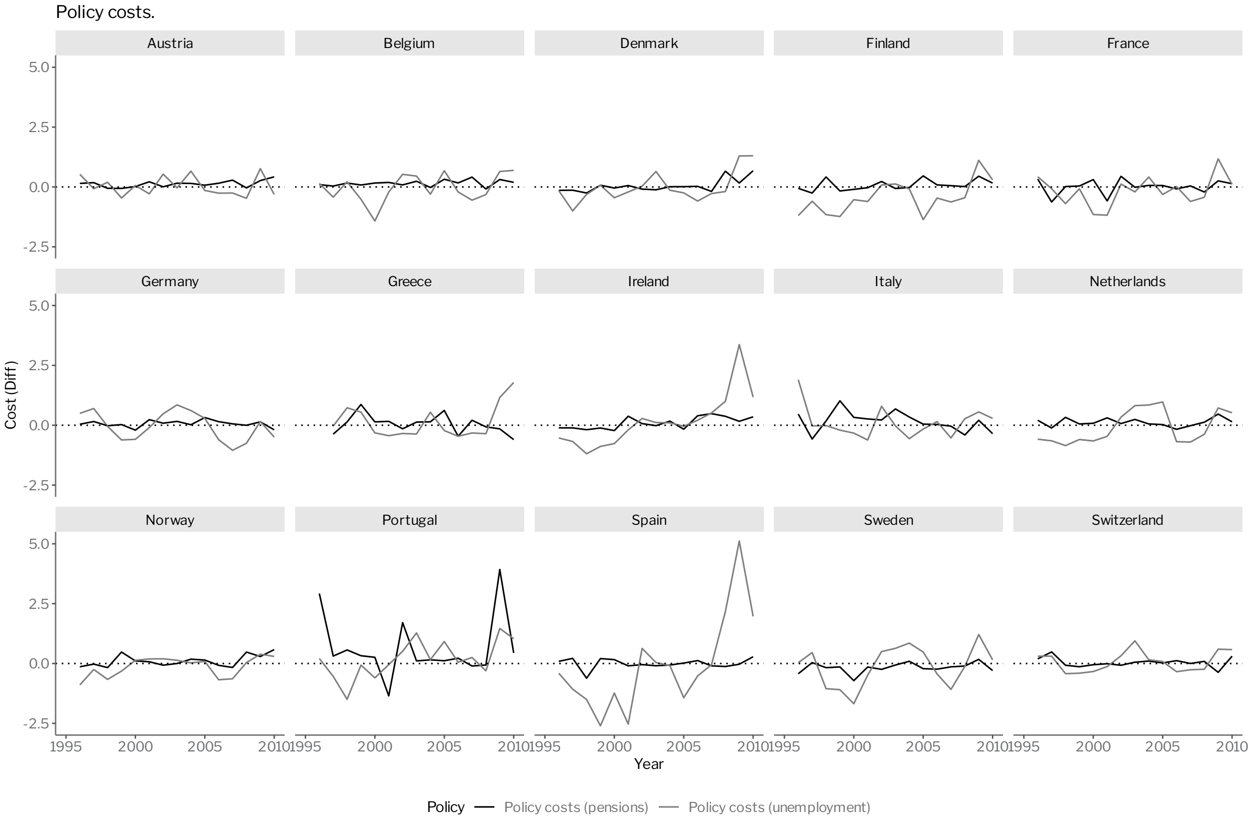


Figure 19.3: Policy costs (Diff).

20 Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. Cross-Lagged panel model with Fixed Effects (ML-SEM). w/o number of parties. w/ administrative reform.

mm-tscs-2r141212305

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

20.1 Covariates

VPI

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

vpi.original <- vpi

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

# GDPpc
load(”data-wdi/data-wdi.RData”) # loads wdi

# Number of political parties, time horizon and opposition
load(”data-parlgov/230918-data-parlgov-governments.RData”)

# Electoral competition
load(”data-electoral_competition/data-electoral_competition.RData”) # loads ec

# Debt
load(”data-debt/imf_debt.RData”) # loads imf.debt

# EU
load(”data-eu/data-eu.RData”) # loads eum.yearly

# Political constraints
load(”data-polcon/polcon2017.RData”) # loads polcon

# Corporatism
load(”data-corporatism/corporatism_jahn.RData”) # loads corporatism

# Social sector indicators, unemployment and elderly
load(”data-social_policy/data-sd.RData”) # loads sd

# Salience, interpolated
load(”data-salience/data-manifesto.RData”) # loads m.country.interpolation.annual

# Regional authority
load(”data-regional_authority/data-rai.RData”) #  loads rai

# Administrative reforms
load(”data-administrative_reforms/data-administrative_reforms.RData”) # admin.reforms

# Replacement rates
load(”data-replacement_rates/data-replacement_rate.RData”) # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with(”Cost”)) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(opposition) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(rai) %>%
  left_join(policy.cost) %>%
  left_join(admin.reforms) %>%
  left_join(m.country.interpolation.annual %>%
            filter(Sector %in% c("Environmental", "Social")) %>%
            mutate(Sector = factor(Sector)) %>%
            select(Country, Sector, Year, Salience) ) %>%
  left_join(m.country.interpolation.annual %>%
            filter(Sector %in% c("Environmental", "Social345minus")) %>%
            mutate(Sector = factor(ifelse(Sector == "Social345minus", "Social", Sector))) %>%
            select(Country, Sector, Year, SalienceGovWseats) %>%
            rename(SalienceGov = SalienceGovWseats) )  %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)



fd <- fd %>%
  mutate(`Blame avoidance opportunities` =
           std1(`Political constraints`) +
           std1(`Divergent opposition (SD LeftRight)`))



Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

year.group <- cut(year.label.numeric, 8)
year.group.label <- levels(year.group)
year.group <- as.numeric(year.group)
nYG <- length(year.group.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience, #SalienceGov,
    `Administrative coordination reforms (n)`,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

D <- list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  year.group = year.group, nYG = nYG,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

inits <- function() {
  list(
    .RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6),
    alpha = array(rep(0, nS *nYG), dim = c(nS, nYG)),
    theta = array(rnorm(nS * (nV + nVS), 0, 0.1), dim = c(nS, (nV + nVS)))
  )
}


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     alpha[s,year.group[y]] +
                     beta[s] * Y[s,c,y-1] +
                     gamma[s,c] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   gamma[s,c] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
      gamma[s,c] ~ dt(0, 0.1^-2, 3)
    }
    beta[s] ~ dunif(-1, 1)
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nYG) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
    }
    alpha[s,1] ~ dnorm(0, 5^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

par <- NULL
par <- c(par, "theta")
par <- c(par, "alpha")
par <- c(par, "nu")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda_c")
par <- c(par, "lambda")
par <- c(par, "beta", "gamma")

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 2
  adapt <- 5e2
  burnin <- 6e4
  run <- 2e3
  thin <- 5
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 18
##  $ Y                            : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA  ...
##  $ nC                           : int 15
##  $ nS                           : int 2
##  $ nY                           : int 16
##  $ year.group                   : num [1:16] 1 1 2 2 3 3 4 4 5 5  ...
##  $ nYG                       : int 8
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 1 3 4 5 6 7
##  $ varElectoralCompetition   : int 2
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] 0.654 -0.226 0.654 -0.226 NA ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,year.group[y]] +
##                      beta[s] * Y[s,c,y-1] +
##                      gamma[s,c] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                    alpha[s,1] +
##                    gamma[s,c] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       gamma[s,c] ~ dt(0, 0.1^-2, 3)
##     }
##     beta[s] ~ dunif(-1, 1)
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##       for (y in 2:nYG) {
##         alpha[s,y] ~ dnorm(alpha[s,y-1], 0.2^-2)
##       }
##       alpha[s,1] ~ dnorm(0, 5^-2)
##       #
##       lambda[s] ~ dnorm(0, 1^-2)
##       nu[s] <- exp(nu.log[s])
##       nu.log[s] ~ dunif(0, 5)
##     }
##     #
##       #
##     for (v in 1:(nV)) {
##       for (c in 1:nC) {
##         X[c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##     for (v in 1:(nVS)) {
##       for (c in 1:nC) {
##         for (s in 1:nS) {
##           XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##           for (y in 2:nY) {
##             XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##           }
##         }
##       }
##     }
## }
```

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

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

20.2 Model interpretation

```
L.gamma <- plab("gamma", list(Sector = sector.label, Country = country.label))
S.gamma <- ggs(s, family = "gamma", par_labels = L.gamma)
S.gamma %>%
  filter(Sector == "Social") %>%
ggs_caterpillar(label = "Country") +
  ggtitle("Varying intercepts by country")


L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
```

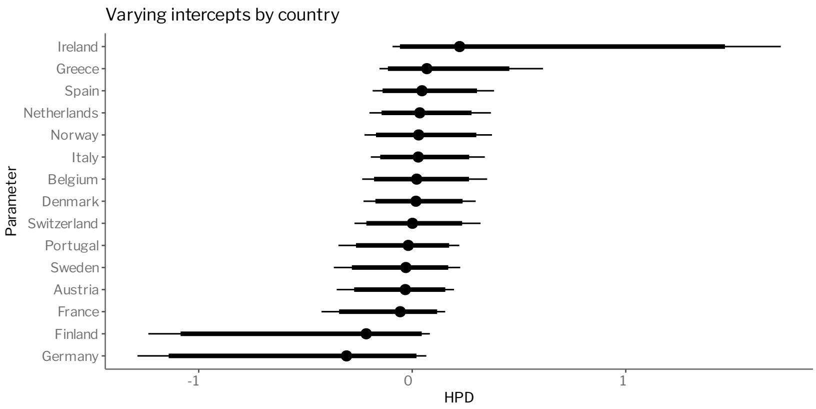


Figure 20.1: Varying intercepts by country. Social sector.

```
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = cov.rename.reorder(Covariate))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2r141212305.RData"))

S.theta %>%
  filter(Sector == "Social") %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  ggtitle(expression(paste("Covariates (", theta, "). Reference.", sep = "")))
```

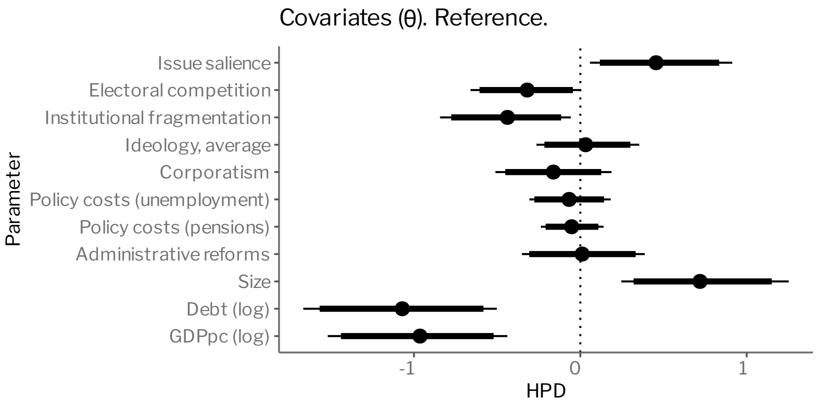


Figure 20.2: Covariates. Social.

21

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. Government ideology as Manifesto weighted.

mm-tscs-2r100000002301

```
load(”data-moneymouth-compensation.RData”) # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = ”Variable”, values_to = ”value”) %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == ”Lag: 1”)

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = ”sample_description.RData”)
```

21.1 Covariates

VPI

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

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual %>%
            filter(Sector %in% c("Environmental", "Social")) %>%
            mutate(Sector = factor(Sector)) %>%
            select(Country, Sector, Year, Salience) ) %>%
  left_join(m.country.interpolation.annual %>%
            filter(Sector %in% c("Environmental", "Social345minus")) %>%
            mutate(Sector = factor(ifelse(Sector == "Social345minus", "Social", Sector))) %>%
            select(Country, Sector, Year, SalienceGovWseats) %>%
            rename(SalienceGov = SalienceGovWseats) )  %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)

Y.df <- fd %>%
  select(Sector, Country, Year, Compensation) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

countryNotSwitzerland ← which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience, SalienceGov,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
         `Government salience` = SalienceGov,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

variable.sector.label ← dimnames(XS)[[4]]
nVS ← length(variable.sector.label)
```

```
varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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

sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:6] 2 3 4 5 6 7
##  $ varElectoralCompetition   : int 1
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:4] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 4
##  $ XS                        : num [1:2, 1:15, 1:16, 1:7] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 7
##  $ b0                        : num [1:4] 0 0 0 0
##  $ B0                        : num [1:4, 1:4] 1 0 0 0 0 1 0 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                       alpha[s,y] +
##                       inprod(X[c,y-1,], theta[s,1:nV]) +
##                       inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                       rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##       lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                     alpha[s,1] +
##                     inprod(X[c,1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##         }
##       }
##     }
## }
```

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

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

21.2 Model interpretation

```
L.theta ← plab(”theta”, list(Sector = sector.label,
                             Covariate = c(variable.label,
                                           variable.sector.label)))
S.theta ← ggs(s, family = ”^theta”, par_labels = L.theta) %>%
  filter(!(Sector == ”Social” & Covariate %in% c(”Policy costs (dummy)”))) %>%
  filter(!(Sector == ”Environmental” & Covariate %in% c(”Policy costs (pensions)”, ”Policy costs (unemployment)”))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    ”Issue salience”, ”Electoral competition”, #”Parties in government”,
    ”Institutional fragmentation”,
    ”Policy costs (unemployment)”, ”Policy costs (pensions)”, ”Policy costs (dummy)”,
    ”Ideology, average”, ”Corporatism”,
    ”Size”,
    ”Debt (log)”, ”GDPpc (log)”))))
#  filter(!(Sector == ”Environmental” & Covariate %in% c(”Unemployment”, ”Elderly”)))

ci.theta ← ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0(”ci-theta-2r100000002301.RData”))
```

22 Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. Compensation with IHS(diffSize).

mm-tscs-2r200000002301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

22.1 Covariates

VPI

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

```
vpi.original ← vpi

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

# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  ← left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()

universe ← expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd ← universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year ⩾ year.start)

Y.df ← fd %>%
  select(Sector, Country, Year, Compensation.IHS) %>%
  rename(Compensation = Compensation.IHS) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y ← Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                   alpha[s,y] +
                   inprod(X[c,y-1,], theta[s,1:nV]) +
                   inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                   rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      #lambda_c[s,c] ~ dt(0, 0.1^-2, 3)
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                 alpha[s,1] +
                 inprod(X[c,1,], theta[s,1:nV]) +
                 inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))

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


sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                          : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                         : int 15
##  $ nS                         : int 2
##  $ nY                         : int 16
##  $ countryNotSwitzerland      : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland         : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ id.decade                  : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                   : int 3
##  $ X                          : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                         : int 5
##  $ XS                         : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                        : int 6
##  $ b0                         : num [1:5] 0 0 0 0 0
##  $ B0                         : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,y] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                       inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                       rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##         resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##       }
##       tau[s,c] <- 1 / sigma.sq[s,c]
##       sigma.sq[s,c] <- exp(sigma[s,c])
##       sigma[s,c] <- lambda[s] + lambda_c[s,c]
##             lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##       #
##       Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##       mu[s,c,1] <-
##                     alpha[s,1] +
##                     inprod(X[c,1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##     }
##     #
##     rho[s] ~ dunif(-1, 1)
##     for (t in 1:(nV + nVS)) {
##       theta[s,t] ~ dnorm(0, 5^-2)
##     }
##     for (y in 2:nY) {
##       alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##     }
##     alpha[s,1] ~ dnorm(0, 1^-2)
##     #
##     lambda[s] ~ dnorm(0, 1^-2)
##     nu[s] <- exp(nu.log[s])
##     nu.log[s] ~ dunif(0, 5)
##   }
##   #
##     #
##   for (v in 1:(nV)) {
##     for (c in 1:nC) {
##       X[c,1,v] ~ dnorm(0, 0.5^-2)
##       for (y in 2:nY) {
##         X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##       }
##     }
##   }
##   for (v in 1:(nVS)) {
##     for (c in 1:nC) {
##       for (s in 1:nS) {
##         XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##   }
## }

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

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

22.2 Model interpretation

```
L.theta ← plab("theta", list(Sector = sector.label,
                             Covariate = c(variable.label,
                                           variable.sector.label)))
S.theta ← ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))

ci.theta ← ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2r200000002301.RData"))
```

23
Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. Compensation with (diffSize + 1)).

mm-tscs-2r300000002301

```r
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary <- env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc <- mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start <- 1980
if (compensation.is.budget) {
  set.year.start <- 1995
}
set.year.end <- 2020
if (run.forecast) {
  set.year.end <- set.year.end + forecasted.years
}

countries <- country.coverage <- as.character(levels(mmc$Country))
nC <- length(countries)
years <- range(mmc$Year)
year.start <- min(years)
year.start <- 1995
year.finish <- max(years)
year.finish <- 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

23.1 Covariates

VPI

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

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
              group_by(Sector, Country, Year) %>%
              summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)

Y.df <- fd %>%
  select(Sector, Country, Year, Compensation.plus1) %>%
  rename(Compensation = Compensation.plus1) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")
```

```
Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label ← dimnames(Y)[[1]]
nS ← length(sector.label)
country.label ← dimnames(Y)[[2]]
nC ← length(country.label)
year.label ← dimnames(Y)[[3]]
year.label.numeric ← as.integer(year.label)
nY ← length(year.label)

countryNotSwitzerland ← which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland ← which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df ←
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X ← X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df ← fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector ≠ "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector ≠ "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS ← XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition ← which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)

b0 ← rep(0, (nV + nVS))
B0 ← diag((nV + nVS))
b0 ← rep(0, (nV))
B0 ← diag((nV))
```

```
diag(B0) ← 2.5^-2
diag(B0) ← 1^-2

D ← list(
  Y = unname(Y),
  nC = nC, nS = nS, nY = nY,
  countryNotSwitzerland = countryNotSwitzerland, countrySwitzerland = countrySwitzerland,
  varNotElectoralCompetition = varNotElectoralCompetition, varElectoralCompetition = varElectoralCompetition,
  id.decade = id.decade, nDecades = nDecades,
  X = unname(X),
  nV = nV,
  XS = unname(XS),
  nVS = nVS,
  b0 = b0, B0 = B0
)

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


m ← 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] ←
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] ← Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] ← 1 / sigma.sq[s,c]
      sigma.sq[s,c] ← exp(sigma[s,c])
      sigma[s,c] ← lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] ←
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] ← exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste(”models/model-”, M, ”.bug”, sep = ””))

par ← NULL
par ← c(par, ”theta”)
par ← c(par, ”alpha”)
par <- c(par, "nu")
par <- c(par, "Sigma")
par <- c(par, "rho")
par <- c(par, "lambda_c")
par <- c(par, "lambda")


sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                           : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                          : int 15
##  $ nS                          : int 2
##  $ nY                          : int 16
##  $ countryNotSwitzerland       : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland          : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition    : int 1
##  $ id.decade                   : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                    : int 3
##  $ X                           : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                          : int 5
##  $ XS                          : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                         : int 6
##  $ b0                          : num [1:5] 0 0 0 0 0
##  $ B0                          : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                       alpha[s,y] +
##                       inprod(X[c,y-1,], theta[s,1:nV]) +
##                       inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                       rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##           resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##         }
##         tau[s,c] <- 1 / sigma.sq[s,c]
##         sigma.sq[s,c] <- exp(sigma[s,c])
##         sigma[s,c] <- lambda[s] + lambda_c[s,c]
##         lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##         #
##         Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##         mu[s,c,1] <-
##                     alpha[s,1] +
##                     inprod(X[c,1,], theta[s,1:nV]) +
##                     inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##       }
##       #
##       rho[s] ~ dunif(-1, 1)
##       for (t in 1:(nV + nVS)) {
##         theta[s,t] ~ dnorm(0, 5^-2)
##       }
##       for (y in 2:nY) {
##         alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##       }
##       alpha[s,1] ~ dnorm(0, 1^-2)
##       #
##       lambda[s] ~ dnorm(0, 1^-2)
##       nu[s] <- exp(nu.log[s])
##       nu.log[s] ~ dunif(0, 5)
##     }
##     #
##       #
##     for (v in 1:(nV)) {
##       for (c in 1:nC) {
##         X[c,1,v] ~ dnorm(0, 0.5^-2)
##         for (y in 2:nY) {
##           X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##         }
##       }
##     }
##     for (v in 1:(nVS)) {
##       for (c in 1:nC) {
##         for (s in 1:nS) {
##           XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##           for (y in 2:nY) {
##             XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##           }
##         }
##       }
##     }
## }

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

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

23.2 Model interpretation

```
L.theta <- plab("theta", list(Sector = sector.label,
                              Covariate = c(variable.label,
                                            variable.sector.label)))
S.theta <- ggs(s, family = "^theta", par_labels = L.theta) %>%
  filter(!(Sector == "Social" & Covariate %in% c("Policy costs (dummy)"))) %>%
  filter(!(Sector == "Environmental" & Covariate %in% c("Policy costs (pensions)", "Policy costs (unemployment)"))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    "Issue salience", "Electoral competition", #"Parties in government",
    "Institutional fragmentation",
    "Policy costs (unemployment)", "Policy costs (pensions)", "Policy costs (dummy)",
    "Ideology, average", "Corporatism",
    "Size",
    "Debt (log)", "GDPpc (log)"))))
#  filter(!(Sector == "Environmental" & Covariate %in% c("Unemployment", "Elderly")))

ci.theta <- ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0("ci-theta-2r300000002301.RData"))
```

24

Explanatory model for compensation: Baseline. Lag 1. Robust/clustered errors by country. Year trend by Kalman filter. AR(1). No interactions. Social policy costs in difference, and environmental policy costs as binary. w/o number of parties. Compensation with LEE log(exp(%diffCap) / exp(diffSize)).

mm-tscs-2r400000002301

```
load("data-moneymouth-compensation.RData") # loads 'mmc'

# Convert env.specific.instruments.growth to a single  binary variable
env.specific.instruments.growth.summary ← env.specific.instruments.growth %>%
  pivot_longer(-c(Sector, Country, Year), names_to = "Variable", values_to = "value") %>%
  group_by(Sector, Country, Year) %>%
  summarize(Costs = ifelse(sum(value) > 0, 1, 0)) %>%
  ungroup()

mmc ← mmc %>%
  filter(Lag == "Lag: 1")

####### Get the specific values to process
set.year.start ← 1980
if (compensation.is.budget) {
  set.year.start ← 1995
}
set.year.end ← 2020
if (run.forecast) {
  set.year.end ← set.year.end + forecasted.years
}

countries ← country.coverage ← as.character(levels(mmc$Country))
nC ← length(countries)
years ← range(mmc$Year)
year.start ← min(years)
year.start ← 1995
year.finish ← max(years)
year.finish ← 2010

save(countries, years, year.start, year.finish, file = "sample_description.RData")
```

24.1 Covariates

VPI

```
load("data-vpi/vpi_wgi.RData")
vpi.implementation ← "No model, simple addition"
vpi ← vpi %>%
  #
  filter(!Country %in% c("Mexico", "Turkey")) %>%
```

```r
  droplevels() %>%
  #
  mutate(Dimension = as.character(Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Implementation costs", "Top-down (VPI)", Dimension)) %>%
  mutate(Dimension = ifelse(Dimension == "Policy feedback", "Bottom-up (VPI)", Dimension))

vpi.original <- vpi

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


# GDPpc
load("data-wdi/data-wdi.RData") # loads wdi

# Number of political parties, time horizon
load("data-parlgov/data-parlgov-governments.RData")

# Electoral competition
load("data-electoral_competition/data-electoral_competition.RData") # loads ec

# Debt
load("data-debt/imf_debt.RData") # loads imf.debt

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

# Political constraints
load("data-polcon/polcon2017.RData") # loads polcon

# Corporatism
load("data-corporatism/corporatism_jahn.RData") # loads corporatism

# Social sector indicators, unemployment and elderly
load("data-social_policy/data-sd.RData") # loads sd

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Replacement rates
load("data-replacement_rates/data-replacement_rate.RData") # loads rr and rr.imp, which are imputed


# Create policy cost and its lag
policy.cost  <- left_join(sd, rr.imp) %>%
  mutate(Cost.Unemployment = Unemployment * RR.Unemployment) %>%
  mutate(Cost.Pensions = Elderly * RR.Pensions) %>%
  select(Country, Year, starts_with("Cost")) %>%
  filter(!is.na(Country)) %>%
  filter(Country %in% countries) %>%
  filter(Year %in% year.start:year.finish) %>%
  arrange(Country, Year) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Cost Unemployment (diff)` = Cost.Unemployment - lag(Cost.Unemployment)) %>%
  mutate(`Cost Pensions (diff)` = Cost.Pensions - lag(Cost.Pensions)) %>%
  ungroup()


universe <- expand_grid(
  Sector = c(),
  Country = countries,
  Year = (year.start - 1):year.finish)

fd <- universe %>%
  left_join(mmc) %>%
  left_join(wdi) %>%
  left_join(vpi %>%
            group_by(Sector, Country, Year) %>%
            summarize(VPI = mean(VPI))) %>%
  left_join(spread(vpi, Dimension, VPI)) %>%
  left_join(governments) %>%
  left_join(ec.full.year.average) %>%
  left_join(imf.debt) %>%
  left_join(eum.yearly) %>%
  left_join(polcon) %>%
  left_join(corporatism) %>%
  left_join(policy.cost) %>%
  left_join(m.country.interpolation.annual) %>%
  left_join(env.specific.instruments.growth.summary) %>%
  filter(Year >= year.start)

Y.df <- fd %>%
  select(Sector, Country, Year, Compensation.LEE) %>%
  rename(Compensation = Compensation.LEE) %>%
  group_by(Sector) %>%
  mutate(Compensation = std1(Compensation)) %>%
  ungroup()
Y <- Y.df %>%
  reshape2::acast(Sector ~ Country ~ Year, value.var = "Compensation")

Y.df %>%
  summarize(pMissing = length(which(is.na(Compensation))) / n())

sector.label <- dimnames(Y)[[1]]
nS <- length(sector.label)
country.label <- dimnames(Y)[[2]]
nC <- length(country.label)
year.label <- dimnames(Y)[[3]]
year.label.numeric <- as.integer(year.label)
nY <- length(year.label)

countryNotSwitzerland <- which(ifelse(country.label == "Switzerland", 0, 1) == 1)
countrySwitzerland <- which(ifelse(country.label == "Switzerland", 1, 0) == 1)

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)


X.df <-
  fd %>%
  select(Country, Year,
    GDPpc,
    Debt,
    Corporatism,
    `Ideology, average`,
    `Political constraints`#,
    ) %>%
  rename(`Institutional fragmentation` = `Political constraints`) %>%
  unique() %>%
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt) %>%
  mutate(`GDPpc (log)` = log(GDPpc)) %>%
  select(-GDPpc) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  spread(Variable, value) %>%
  #
  gather(Variable, value, -c(Country, Year)) %>%
  # nan to na
  mutate(value = ifelse(is.nan(value), NA, value))
X <- X.df %>%
  reshape2::acast(Country ~ Year ~ Variable, value.var = "value")

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

XS.df <- fd %>%
  select(Country, Sector, Year,
         `Electoral competition`, Salience,
         `Cost Pensions (diff)`,
         `Cost Unemployment (diff)`,
         Size) %>%
  unique() %>%
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  group_by(Variable, Sector) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # add interactions
  spread(Variable, value) %>%
  mutate(`Cost Pensions (diff)` = ifelse(Sector != "Social", 0, `Cost Pensions (diff)`)) %>%
  mutate(`Cost Unemployment (diff)` = ifelse(Sector != "Social", 0, `Cost Unemployment (diff)`)) %>%
  rename(`Issue salience` = Salience,
    `Policy costs (pensions)` = `Cost Pensions (diff)`,
    `Policy costs (unemployment)` = `Cost Unemployment (diff)`) %>%
  # add binary variables
  left_join(env.specific.instruments.growth.summary) %>%
  mutate(Costs = ifelse(Sector != "Environmental", 0, Costs)) %>%
  rename(`Policy costs (dummy)` = Costs) %>%
  #
  gather(Variable, value, -c(Country, Sector, Year)) %>%
  mutate(value = ifelse(is.nan(value), NA, value))

XS <- XS.df %>%
  reshape2::acast(Sector ~ Country ~ Year ~ Variable, value.var = "value")

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

varNotElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 0, 1) == 1)
varElectoralCompetition <- which(ifelse(variable.sector.label == "Electoral competition", 1, 0) == 1)
```

```
b0 <- rep(0, (nV + nVS))
B0 <- diag((nV + nVS))
b0 <- rep(0, (nV))
B0 <- diag((nV))
diag(B0) <- 2.5^-2
diag(B0) <- 1^-2

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

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


m <- 'model {
  for (s in 1:nS) {
    for (c in 1:nC) {
      for (y in 2:nY) {
        Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
        mu[s,c,y] <-
                     alpha[s,y] +
                     inprod(X[c,y-1,], theta[s,1:nV]) +
                     inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
                     rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
        resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
      }
      tau[s,c] <- 1 / sigma.sq[s,c]
      sigma.sq[s,c] <- exp(sigma[s,c])
      sigma[s,c] <- lambda[s] + lambda_c[s,c]
      lambda_c[s,c] ~ dnorm(0, 0.2^-2)
      #
      Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
      mu[s,c,1] <-
                   alpha[s,1] +
                   inprod(X[c,1,], theta[s,1:nV]) +
                   inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
    }
    #
    rho[s] ~ dunif(-1, 1)
    for (t in 1:(nV + nVS)) {
      theta[s,t] ~ dnorm(0, 5^-2)
    }
    for (y in 2:nY) {
      alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
    }
    alpha[s,1] ~ dnorm(0, 1^-2)
    #
    lambda[s] ~ dnorm(0, 1^-2)
    nu[s] <- exp(nu.log[s])
    nu.log[s] ~ dunif(0, 5)
  }
  #
  # Missing data
  #
  for (v in 1:(nV)) {
    for (c in 1:nC) {
      X[c,1,v] ~ dnorm(0, 0.5^-2)
      for (y in 2:nY) {
        X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
      }
    }
  }
  for (v in 1:(nVS)) {
    for (c in 1:nC) {
      for (s in 1:nS) {
        XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
        for (y in 2:nY) {
          XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
        }
      }
    }
  }
}'
write(m, file = paste("models/model-", M, ".bug", sep = ""))
```

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


sample.length <- c("test", "middle", "serious")
sample.length <- sample.length[3]
if (sample.length == "test") {
  chains <- 2
  adapt <- 1e2
  burnin <- 5e2
  run <- 5e2
  thin <- 1
} else if (sample.length == "middle") {
  chains <- 3
  adapt <- 2e2
  burnin <- 1e4
  run <- 2e3
  thin <- 1
} else if (sample.length == "serious") {
  chains <- 4
  adapt <- 5e2
  burnin <- 5e4
  run <- 2e3
  thin <- 10
}

method <- "parallel"
```

Data passed:

```
str(D)
```

```
## List of 16
##  $ Y                         : num [1:2, 1:15, 1:16] NA NA NA NA NA NA NA NA NA NA ...
##  $ nC                        : int 15
##  $ nS                        : int 2
##  $ nY                        : int 16
##  $ countryNotSwitzerland     : int [1:14] 1 2 3 4 5 6 7 8 9 10 ...
##  $ countrySwitzerland        : int 15
##  $ varNotElectoralCompetition: int [1:5] 2 3 4 5 6
##  $ varElectoralCompetition   : int 1
##  $ id.decade                 : num [1:16] 1 1 1 1 1 2 2 2 2 2 ...
##  $ nDecades                  : int 3
##  $ X                         : num [1:15, 1:16, 1:5] 0.984 0.7809 0.175 0.0407 -0.5627 ...
##  $ nV                        : int 5
##  $ XS                        : num [1:2, 1:15, 1:16, 1:6] -0.493 -0.493 0.175 0.175 -0.528 ...
##  $ nVS                       : int 6
##  $ b0                        : num [1:5] 0 0 0 0 0
##  $ B0                        : num [1:5, 1:5] 1 0 0 0 0 0 1 0 0 0 ...
```

Model in JAGS:

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

```
## model {
##   for (s in 1:nS) {
##     for (c in 1:nC) {
##       for (y in 2:nY) {
##         Y[s,c,y] ~ dnorm(mu[s,c,y], tau[s,c])
##         mu[s,c,y] <-
##                      alpha[s,y] +
##                      inprod(X[c,y-1,], theta[s,1:nV]) +
##                      inprod(XS[s,c,y-1,], theta[s,(nV + 1):(nV + nVS)]) +
##                      rho[s] * (Y[s,c,y-1] - mu[s,c,y-1])
##          resid[s,c,y] <- Y[s,c,y] - mu[s,c,y]
##        }
##        tau[s,c] <- 1 / sigma.sq[s,c]
##        sigma.sq[s,c] <- exp(sigma[s,c])
##        sigma[s,c] <- lambda[s] + lambda_c[s,c]
##        lambda_c[s,c] ~ dnorm(0, 0.2^-2)
##        #
##        Y[s,c,1] ~ dnorm(mu[s,c,1], tau[s,c])
##        mu[s,c,1] <-
##                    alpha[s,1] +
##                    inprod(X[c,1,], theta[s,1:nV]) +
##                    inprod(XS[s,c,1,], theta[s,(nV + 1):(nV + nVS)])
##      }
##      #
##      rho[s] ~ dunif(-1, 1)
##      for (t in 1:(nV + nVS)) {
##        theta[s,t] ~ dnorm(0, 5^-2)
##      }
##      for (y in 2:nY) {
##        alpha[s,y] ~ dnorm(alpha[s,y-1], 0.1^-2)
##      }
##      alpha[s,1] ~ dnorm(0, 1^-2)
##      #
##      lambda[s] ~ dnorm(0, 1^-2)
##      nu[s] <- exp(nu.log[s])
##      nu.log[s] ~ dunif(0, 5)
##    }
##    #
##      #
##    for (v in 1:(nV)) {
##      for (c in 1:nC) {
##        X[c,1,v] ~ dnorm(0, 0.5^-2)
##        for (y in 2:nY) {
##          X[c,y,v] ~ dnorm(X[c,y-1,v], 0.02^-2)
##        }
##      }
##    }
##    for (v in 1:(nVS)) {
##      for (c in 1:nC) {
##        for (s in 1:nS) {
##          XS[s,c,1,v] ~ dnorm(0, 0.5^-2)
##          for (y in 2:nY) {
##            XS[s,c,y,v] ~ dnorm(XS[s,c,y-1,v], 0.02^-2)
##          }
##        }
##     }
##   }
## }
```

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

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

24.2 Model interpretation

```
L.theta ← plab(”theta”, list(Sector = sector.label,
                             Covariate = c(variable.label,
                                           variable.sector.label)))
S.theta ← ggs(s, family = ”^theta”, par_labels = L.theta) %>%
  filter(!(Sector == ”Social” & Covariate %in% c(”Policy costs (dummy)”))) %>%
  filter(!(Sector == ”Environmental” & Covariate %in% c(”Policy costs (pensions)”, ”Policy costs (unemployment)”))) %>%
  mutate(Covariate = fct_relevel(Covariate, rev(c(
    ”Issue salience”, ”Electoral competition”, #”Parties in government”,
    ”Institutional fragmentation”,
    ”Policy costs (unemployment)”, ”Policy costs (pensions)”, ”Policy costs (dummy)”,
    ”Ideology, average”, ”Corporatism”,
    ”Size”,
    ”Debt (log)”, ”GDPpc (log)”))))

ci.theta ← ci(S.theta) %>%
  mutate(Model = M.lab,
         `VPI implementation` = vpi.implementation)
save(ci.theta, file = paste0(”ci-theta-2r400000002301.RData”))
```

25
Model comparison

25.1 Salience: Party manifesto and Eurobarometer

```
d ← NULL

# Salience, interpolated
load("data-salience/data-manifesto.RData") # loads m.country.interpolation.annual

# Salience, interpolated, from Eurobarometer
load("data-eurobarometer/data-eurobarometer.RData") # loads eb.country.interpolation.annual

d ← bind_rows(mutate(m.country.interpolation.annual, Source = "Manifesto") %>%
               select(Country, Sector, Year, Salience, Source),
               mutate(eb.country.interpolation.annual, Source = "Eurobarometer")) %>%
  pivot_wider(names_from = Source, values_from = Salience)  %>%
  filter(Year ⩾ 1995)

d ← filter(d, Sector %in% c("Social", "Environmental")) %>% droplevels()
```

25.2 Time-lagged cross correlation

Between two salience indicators: Party manifesto and Eurobarometer. Pearson correlation does not give information about directionality and which one leads and which one follows. So we use time-lagged cross-correlation.

Following this notation we define `x` as Eurobarometer and `y` as Party Manifesto. When `h`, is negative, then Eurobarometer *leads* Party Manifesto.

In the time-lagged cross-correlations we would like to observe at negative lags that there are peaks, meaning that Eurobarometer leads Manifesto.

```
f.tlcc ← function(x, y, lag.max = 10) {
  if (sd(x) ≠ 0 & sd(y) ≠ 0) {
    ac ← as.vector(ccf(x, y, lag.max = lag.max, plot = FALSE)$acf)
  } else {
    ac ← rep(NA, (lag.max * 2) + 1)
  }
  d ← tibble(Lag = -lag.max:lag.max, AC = ac)
  return(d)
}

#d %>%
#  filter(!(Country == "Belgium" & Sector == "Environmental")) %>%
#  filter(!complete.cases(.)) %>%
#  count(Sector, Country) %>%
#  data.frame()

tlcc ← d %>%
  filter(!(Country == "Belgium" & Sector == "Environmental")) %>%
  filter(!(Country %in% c("Norway", "Switzerland"))) %>%
  na.omit() %>%
  droplevels() %>%
  group_by(Country, Sector) %>%
  summarize(f.tlcc(Eurobarometer, Manifesto)) %>%
  ungroup()

tlcc %>%
  filter(Sector == "Social") %>%
  ggplot(aes(x = Lag, y = AC, group = Country)) +
  geom_line() +
  geom_hline(yintercept = 0, lty = 3) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Country)
```

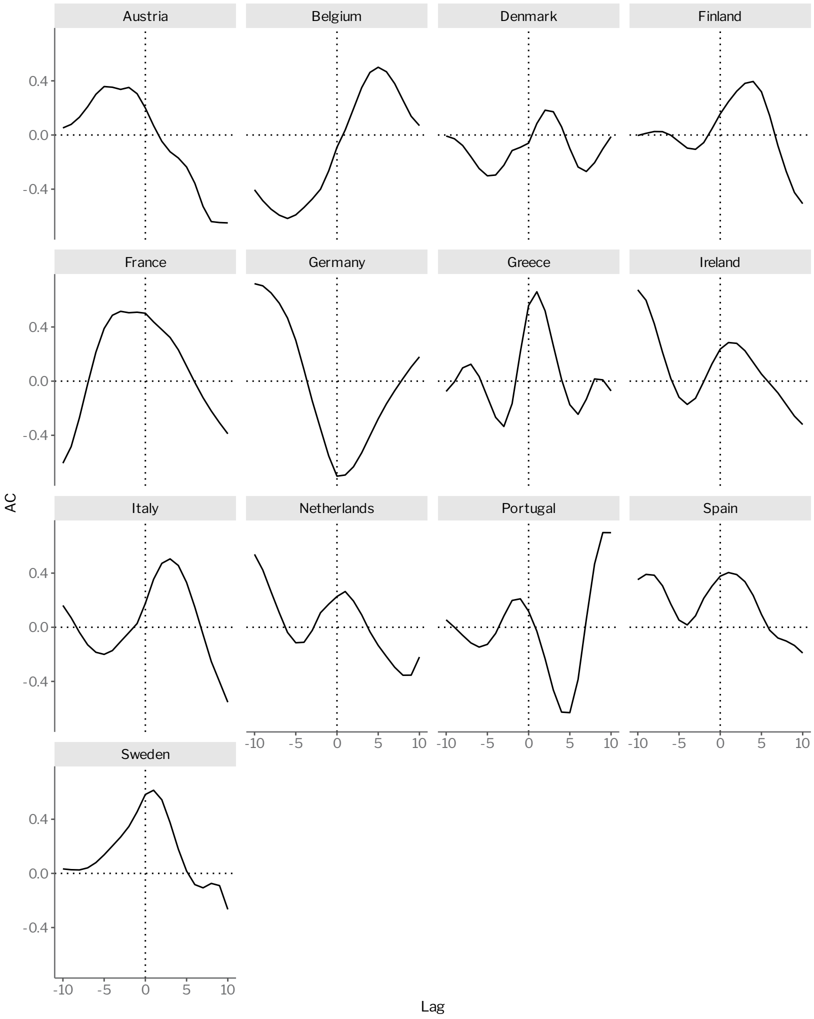


Figure 25.1: Time-lagged cross-correlations between Eurobarometer and Parties. Social sector.

25.3 *w/ Environmental NGOs*

```
d <- NULL

load("ci-theta-2301.RData")
ci.theta <- mutate(ci.theta, Model = "Reference (Manifesto)")
d <- bind_rows(d, ci.theta)

load("ci-theta-3301.RData")
ci.theta <- mutate(ci.theta, Model = "w/ environmental NGOs by population (Manifesto)")
d <- bind_rows(d, ci.theta)

load("ci-theta-102301.RData")
ci.theta <- mutate(ci.theta, Model = "Reference (Eurobarometer)")
d <- bind_rows(d, ci.theta)

load("ci-theta-103301.RData")
ci.theta <- mutate(ci.theta, Model = "w/ environmental NGOs by population (Eurobarometer)")
d <- bind_rows(d, ci.theta)

d <- d %>%
  mutate(Model = fct_relevel(Model, c("Reference (Manifesto)",
                                      "Reference (Eurobarometer)",
                                      "w/ environmental NGOs by population (Manifesto)",
                                      "w/ environmental NGOs by population (Eurobarometer)")))
```

```
d %>%
  filter(Sector == "Environmental") %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Model,
             color = Model)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark") +
  ggtitle(expression(paste("Covariates (", theta, "). Environmental sector", sep = ""))) #+
```

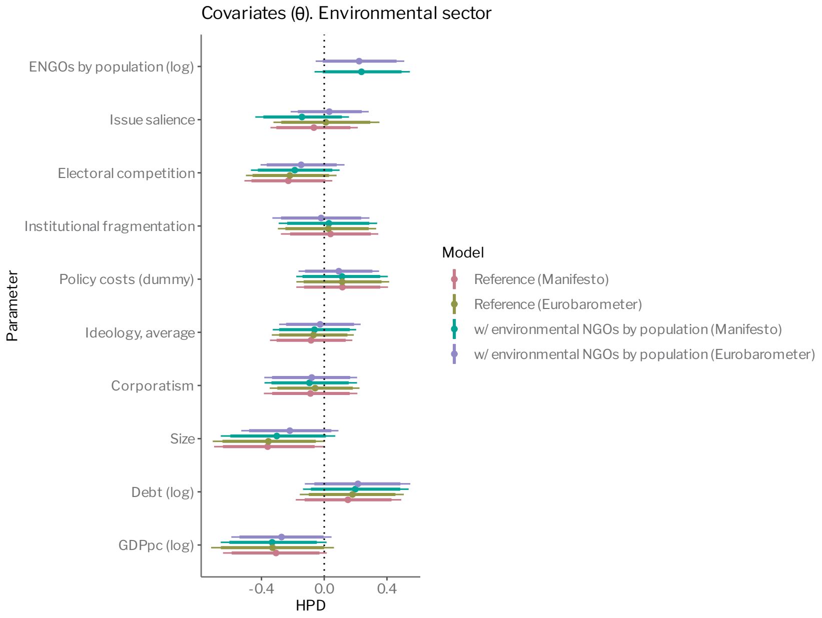


Figure 25.2: Comparison of model results with and without environmental NGOs. Environmental sector.

25.4 Salience: Manifesto vs. Eurobarometer

```
d ← NULL

load("ci-theta-2301.RData")
ci.theta ← mutate(ci.theta, Model = "Reference", Salience =  "Manifesto")
d ← bind_rows(d, ci.theta)

load("ci-theta-2304.RData")
ci.theta ← mutate(ci.theta, Model = "Country-specific lags", Salience = "Manifesto")
d ← bind_rows(d, ci.theta)

load("ci-theta-2305.RData")
ci.theta ← mutate(ci.theta, Model = "ML-SEM", Salience = "Manifesto")
d ← bind_rows(d, ci.theta)


load("ci-theta-102301.RData")
ci.theta ← mutate(ci.theta, Model = "Reference", Salience =  "Eurobarometer")
d ← bind_rows(d, ci.theta)

load("ci-theta-102304.RData")
ci.theta ← mutate(ci.theta, Model = "Country-specific lags", Salience = "Eurobarometer")
d ← bind_rows(d, ci.theta)

load("ci-theta-102305.RData")
ci.theta ← mutate(ci.theta, Model = "ML-SEM", Salience = "Eurobarometer")
d ← bind_rows(d, ci.theta)

d ← d %>%
  mutate(Model = fct_relevel(Model, c("Reference"))) %>%
  mutate(Salience = fct_relevel(Salience, c("Manifesto")))
```

```
d %>%
  filter(Sector == "Social") %>%
  filter(Model == "Reference") %>%
  droplevels() %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Salience,
             color = Salience)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark") +
  theme(legend.position = "bottom")
```

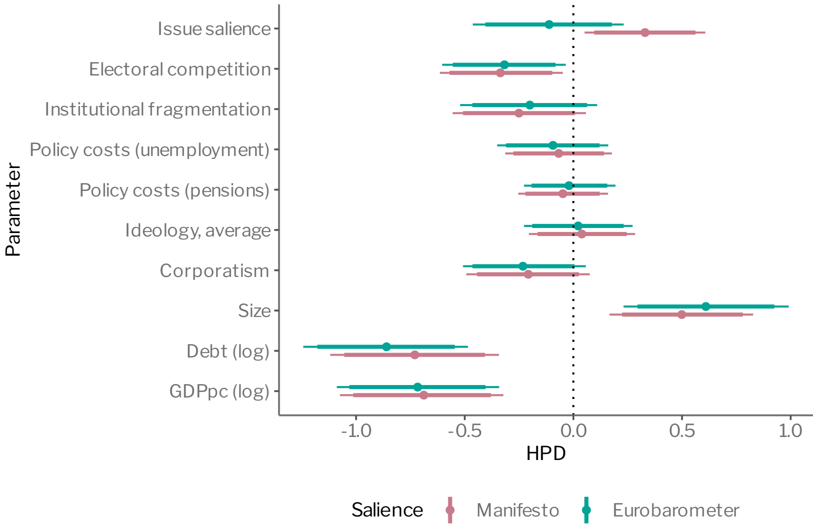


Figure 25.3: Comparison of model results with salience coming from the Manifesto or the Eurobarometer. Reference.

```
d %>%
  filter(Sector == "Social") %>%
  filter(Model == "Country-specific lags") %>%
  droplevels() %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Salience,
             color = Salience)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark") +
  ggtitle("Country-specific lags") +
  theme(legend.position = "bottom")
```

```
d %>%
  filter(Sector == "Social") %>%
  filter(Model == "ML-SEM") %>%
  droplevels() %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Salience,
             color = Salience)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
```

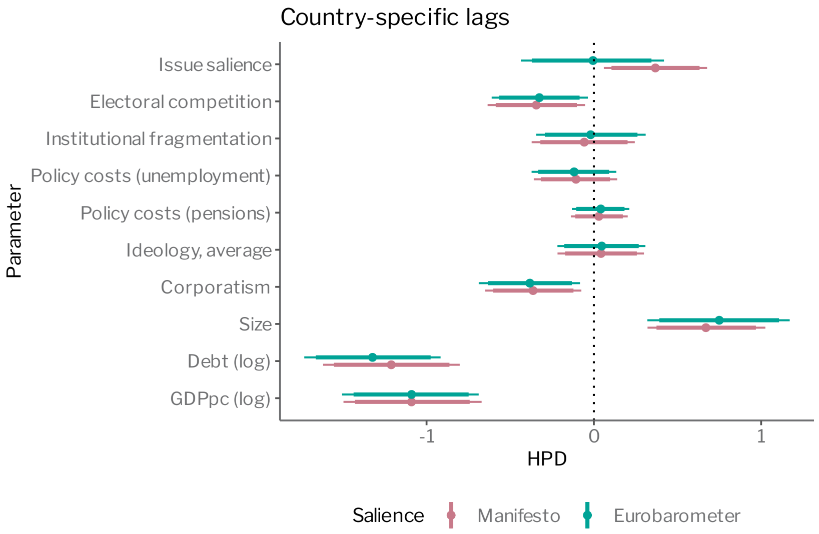


Figure 25.4: Comparison of model results with salience coming from the Manifesto or the Eurobarometer. Country-specific lags.

```
                   size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark") +
  ggtitle("ML-SEM") +
  #scale_color_grey() +
  theme(legend.position = "bottom")
```

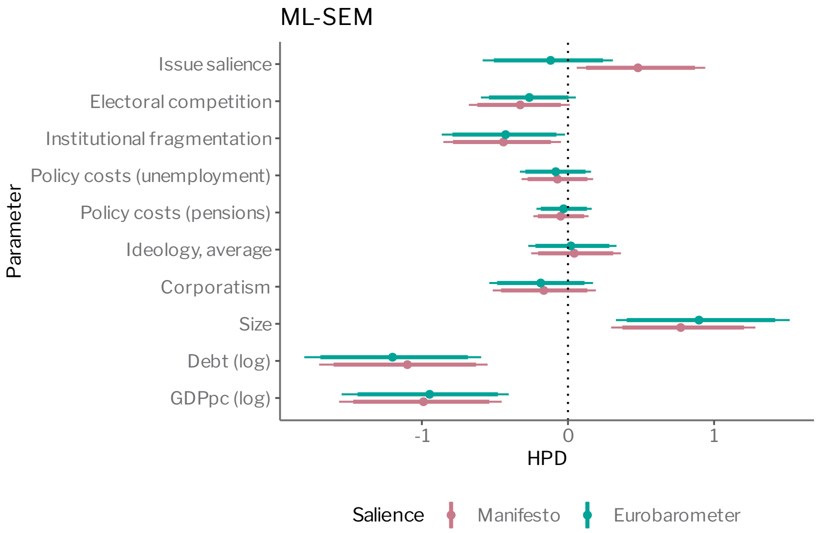


Figure 25.5: Comparison of model results with salience coming from the Manifesto or the Eurobarometer. ML-SEM.

25.5 *Administrative reform, RA’s managerial autonomy and ideological difference with next government*

```
d ← NULL

load("ci-theta-2301.RData")
ci.theta ← mutate(ci.theta, Model = "Reference")
d ← bind_rows(d, ci.theta)

load("ci-theta-171212301.RData")
ci.theta ← mutate(ci.theta, Model = "w/ Administrative reforms\nw/ Pol/Admin separation\nw/ Ideological difference between
d ← bind_rows(d, ci.theta)
```

```
d ← d %>%
  mutate(Covariate = cov.rename.reorder(Covariate)) %>%
  mutate(Model = fct_relevel(Model, c("Reference")))

d %>%
  filter(Sector == "Social") %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Model,
             color = Model)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark") +
  theme(legend.position = "bottom")
```

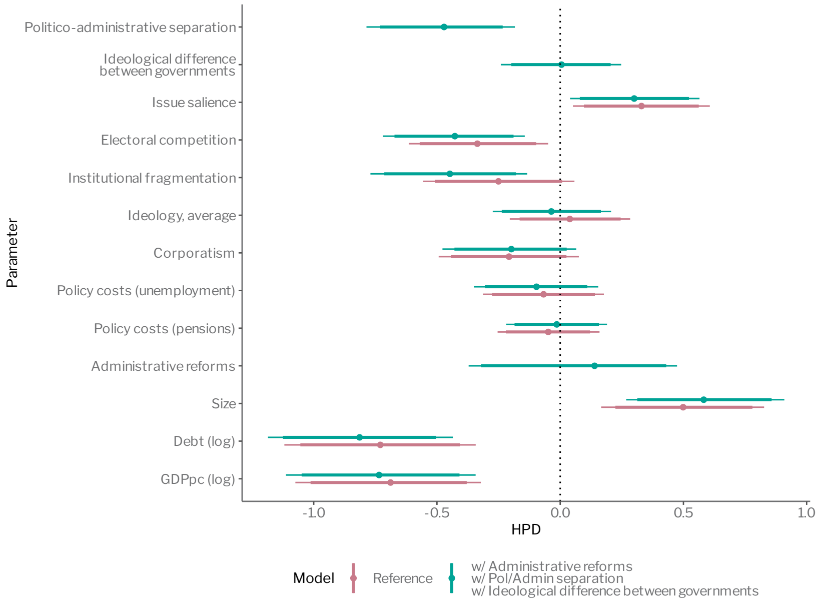


Figure 25.6: Comparison of results with specifications including administrative reform, RAs managerial autonomy and ideological difference with next government. Social sector.

25.6 VPI - Organizational component

```
d ← NULL

load("ci-theta-2301.RData")
ci.theta ← mutate(ci.theta, Model = "Reference")
d ← bind_rows(d, ci.theta)

load("ci-theta-15692301.RData")
ci.theta ← mutate(ci.theta, Model = "w/ Organization")
d ← bind_rows(d, ci.theta)

d ← d %>%
  mutate(Covariate = cov.rename.reorder(Covariate)) %>%
  mutate(Model = fct_relevel(Model, c("Reference")))

d %>%
  filter(Sector == "Social") %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Model,
             color = Model)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab(”Parameter”) + ylab(”HPD”) +
  scale_color_discrete_qualitative(palette = ”Dark”) +
  theme(legend.position = ”bottom”)
```

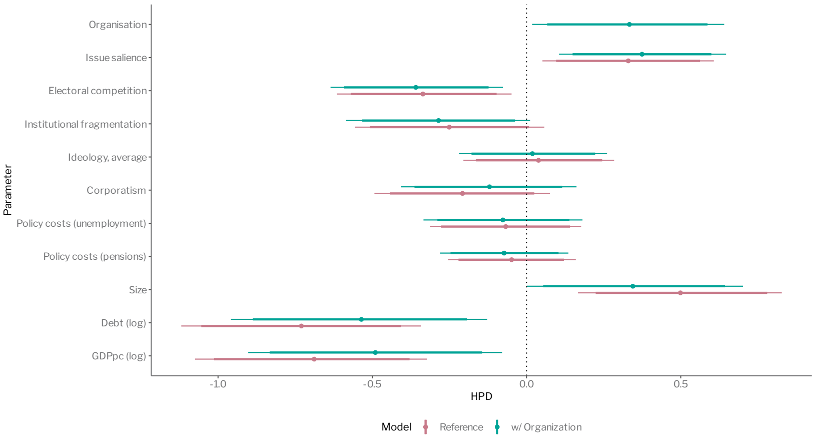


Figure 25.7: Comparison of results with specification including organizational component. Social sector.

25.7 *Compensation, denominator*

```
d ← NULL

load(”ci-theta-2301.RData”)
ci.theta ← mutate(ci.theta, Model = ”Reference”)
d ← bind_rows(d, ci.theta)

load(”ci-theta-2r200000002301.RData”)
ci.theta ← mutate(ci.theta, Model = ”IHS(DiffSize) + 1”)
d ← bind_rows(d, ci.theta)

load(”ci-theta-2r300000002301.RData”)
ci.theta ← mutate(ci.theta, Model = ”Diffsize + 1”)
d ← bind_rows(d, ci.theta)

load(”ci-theta-2r400000002301.RData”)
ci.theta ← mutate(ci.theta, Model = ”log(exp(DiffCap) / exp(DiffSize))”)
d ← bind_rows(d, ci.theta)


d ← d %>%
  mutate(Model = fct_relevel(Model, c(”Reference”)))

d %>%
  filter(Sector == ”Social”) %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
           ymin = low, y = median, ymax = high,
           group = Model,
           color = Model)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                position = position_dodge(width = 0.4),
                size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab(”Parameter”) + ylab(”HPD”) +
  scale_color_discrete_qualitative(palette = ”Dark”) +
  theme(legend.position = ”bottom”)
```

25.8 *Government ideology, weighted*

```
d ← NULL

load(”ci-theta-2301.RData”)
ci.theta ← mutate(ci.theta, Model = ”Reference”)
```

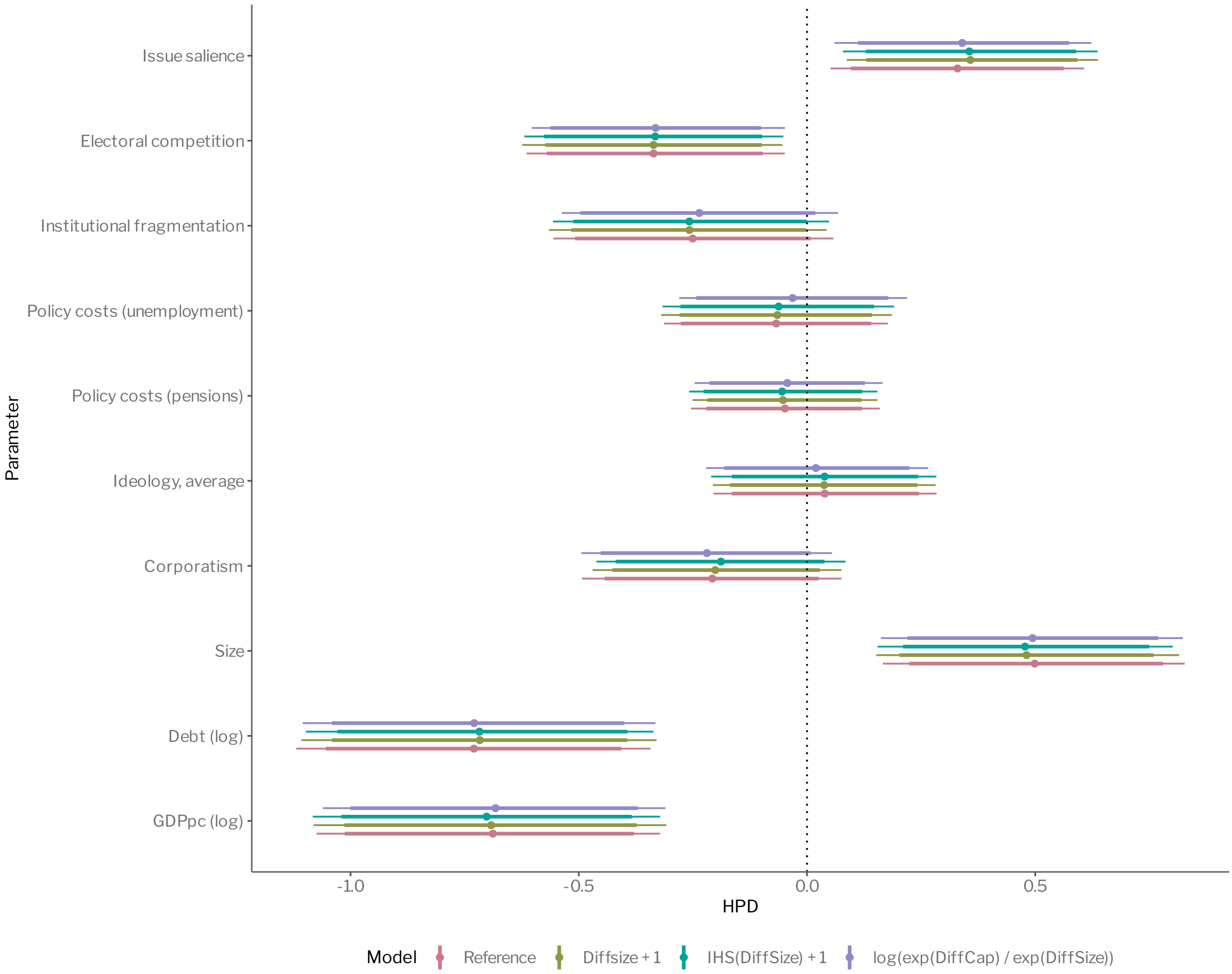


Figure 25.8: Comparison of model results. Different denominators in the compensation. Social sector

```
d ← bind_rows(d, ci.theta)

load("ci-theta-2r100000002301.RData")
ci.theta ← mutate(ci.theta, Model = "Government ideology\nManifesto, weighted")
d ← bind_rows(d, ci.theta)

d ← d %>%
  mutate(Model = fct_relevel(Model, c("Reference")))

d %>%
  filter(Sector == "Social") %>%
  droplevels() %>%
  ggplot(aes(x = Covariate,
             ymin = low, y = median, ymax = high,
             group = Model,
             color = Model)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.4), size = 1.5) +
  geom_linerange(position = position_dodge(width = 0.4)) +
  geom_linerange(aes(ymin = Low, ymax = High),
                 position = position_dodge(width = 0.4),
                 size = 1.0) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark") +
  theme(legend.position = "bottom")
```

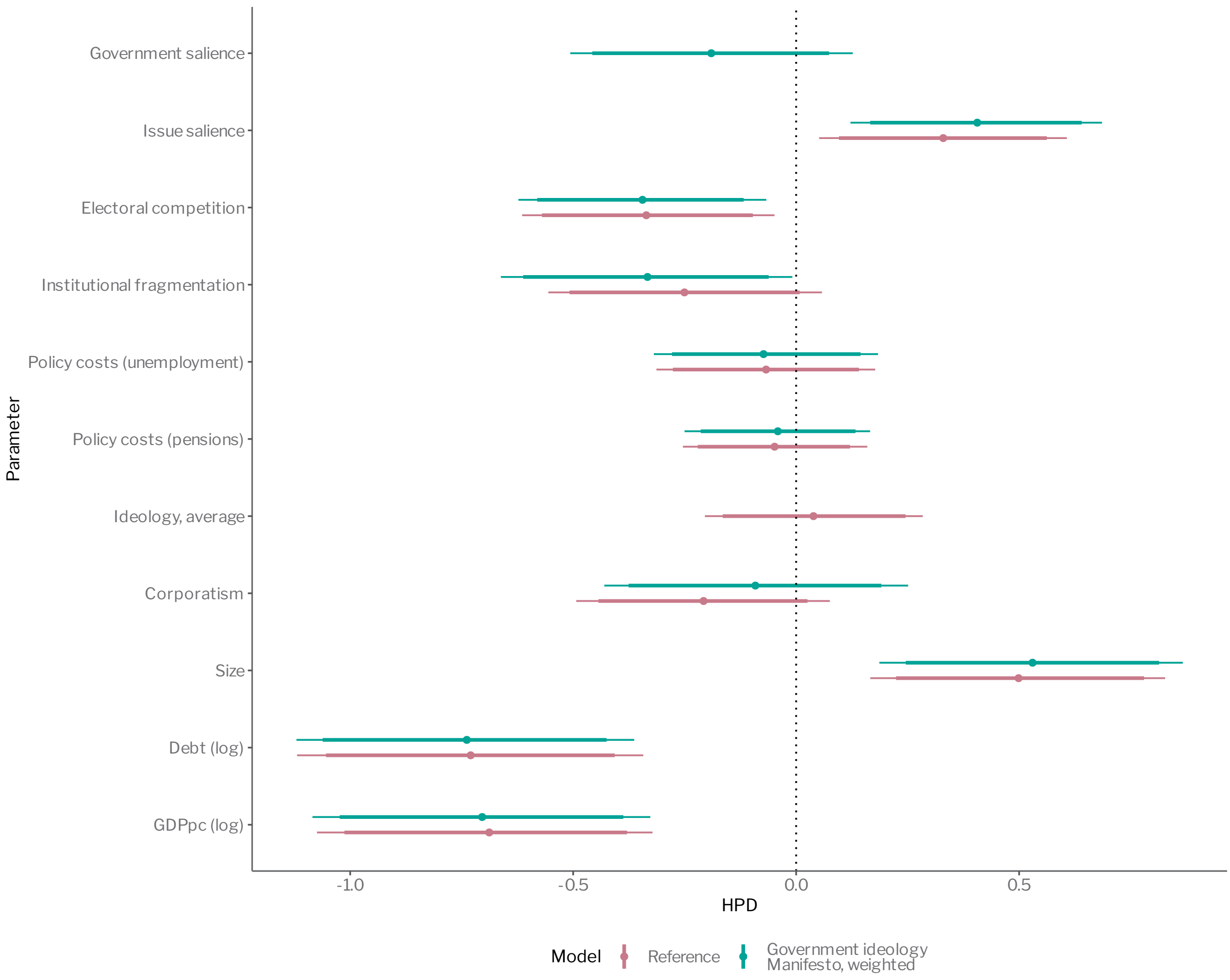


Figure 25.9: Comparison of model results. Different specifications of government ideology. Social sector

Programming environment

```
sessionInfo()
```

```
## R version 4.4.1 (2024-06-14)
## Platform: aarch64-unknown-linux-gnu
## Running under: Gentoo Linux
## 
## Matrix products: default
## BLAS:   /usr/lib64/blas/openblas/libblas.so.3
## LAPACK: /usr/lib64/libopenblasp-r0.3.26.so;  LAPACK version 3.12.0
## 
## locale:
##  [1] LC_CTYPE=ca_AD.UTF-8       LC_NUMERIC=C               LC_TIME=ca_AD.UTF-8
##  [4] LC_COLLATE=ca_AD.UTF-8     LC_MONETARY=ca_AD.UTF-8    LC_MESSAGES=ca_AD.UTF-8
##  [7] LC_PAPER=ca_AD.UTF-8       LC_NAME=C                  LC_ADDRESS=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=ca_AD.UTF-8 LC_IDENTIFICATION=C
## 
## time zone: Europe/Andorra
## tzcode source: system (glibc)
## 
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods   base
## 
## other attached packages:
##  [1] ggridges_0.5.6       tibble_3.2.1         forcats_1.0.0        corrplot_0.94
##  [5] cowplot_1.1.3        PolicyPortfolios_0.3 stringr_1.5.1        colorspace_2.1-1
##  [9] ggrepel_0.9.6        GGally_2.2.1         ggmcmc_1.5.1.1       runjags_2.2.2-4
## [13] rjags_4-16           coda_0.19-4.1        scales_1.3.0         ggthemes_5.1.0
## [17] gridExtra_2.3        kableExtra_1.4.0     tikzDevice_0.12.6    rmarkdown_2.28
## [21] knitr_1.48           ggplot2_3.5.1        tidyr_1.3.1          dplyr_1.1.4
## [25] extrafont_0.19       colorout_1.2-2
## 
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1   filehash_2.4-6     viridisLite_0.4.2  farver_2.1.2       fastmap_1.2.0
##  [6] digest_0.6.37      lifecycle_1.0.4    cluster_2.1.6      magrittr_2.0.3     compiler_4.4.1
## [11] rlang_1.1.4        tools_4.4.1        utf8_1.2.4         yaml_2.3.10        ggsignif_0.6.4
## [16] labeling_0.4.3     plyr_1.8.9         xml2_1.3.6         RColorBrewer_1.1-3 abind_1.4-8
## [21] withr_3.0.1        purrr_1.0.2        fansi_1.0.6        ggpubr_0.6.0       extrafontdb_1.0
## [26] MASS_7.3-61        tinytex_0.53       cli_3.6.3          vegan_2.6-8        generics_0.1.3
## [31] rstudioapi_0.16.0  reshape2_1.4.4     tufte_0.13         splines_4.4.1      parallel_4.4.1
## [36] vctrs_0.6.5        Matrix_1.7-0       carData_3.0-5      car_3.1-2          bookdown_0.40
## [41] rstatix_0.7.2      systemfonts_1.1.0  ineq_0.2-13        glue_1.7.0         ggstats_0.6.0
## [46] stringi_1.8.4      gtable_0.3.5       munsell_0.5.1      pillar_1.9.0       htmltools_0.5.8.1
## [51] R6_2.5.1           evaluate_0.24.0    lattice_0.22-6     backports_1.5.0    broom_1.0.6
## [56] Rcpp_1.0.13        svglite_2.1.3      nlme_3.1-166       Rttf2pt1_1.3.12    permute_0.9-7
## [61] mgcv_1.9-1         xfun_0.47          zoo_1.8-12         pkgconfig_2.0.3
```