Xavier Fernández i Marín, Yves Steinebach, Christian Aschenbrenner

Global empirical analysis of carbon pricing policies

2021-04-23 10:32

Contents

1
Introduction

This report presents the results of the analysis of carbon pricing policies. A first part analyzes the process of adoption of several concrete policies. The second part performs a measurement model of the policies at the end of the period to generate a measurement model of the intensity of carbon policies. Next, a cluster analysis on the individual data of policies is performed. Finally, an analysis of the factors explaining different intensities in the carbon policies is presented.

```
setwd("adoption/")
```

2

Data cleaning and preparation

```
# Write an empty file with country names
cc ← codelist %>%
  select(iso2c, iso3c, Country = country.name.en, WorldBank = wb_api3c) %>%
  filter(!is.na(iso2c)) %>%
  filter(!is.na(WorldBank))

write.table(cc, file = "country_names.csv", sep = ";", row.names = FALSE)

orig ← read.csv("carbon_pricing.csv", check.names = FALSE) %>%
  as_tibble()

time.span ← 1990:2019
country.coverage ← cc$Country
event.considered ← c("Carbon tax",
                     "ETS", "ETS_without EU",
                     "Carbon Pricing", "Carbon Pricing_withoutEU")
event.considered.proper.name ← c("Carbon tax",
                                 "ETS", "ETS (no EU)",
                                 "Carbon pricing", "Carbon pricing (no EU)")
event.considered.noneu ← c("ETS (no EU)", "Carbon pricing (no EU)")

d.full ← tibble(expand.grid(
  Country = country.coverage,
  Year = time.span,
  Outcome = all_of(event.considered.proper.name)))

save(time.span, country.coverage, event.considered, file = "details.RData")

# Ensure there are no cases outside
cases.outside ← orig %>%
  select(event.considered) %>%
  gather(Outcome, Year) %>%
  filter(!(Year %in% time.span) & !is.na(Year))
if (dim(cases.outside)[1] > 0)
  stop("There are cases of carbon tax outside the designed time span")
```

2.1 Events

The events to consider are:

```
event.considered.proper.name
```

```
→ [1] "Carbon tax"          "ETS"                    "ETS (no EU)"
→ [4] "Carbon pricing"      "Carbon pricing (no EU)"
```

```
d.event <- orig %>%
  tibble() %>%
  select(Country, all_of(event.considered))
names(d.event)[-1] <- event.considered.proper.name

d.event <- d.event %>%
  gather(Outcome, Year, -Country) %>%
  mutate(Event = as.integer(if_else(!is.na(Year), 1, 0))) %>%
  mutate(Year = as.integer(as.numeric(Year))) %>%
  mutate(Outcome = as.factor(Outcome)) %>%
  filter(!is.na(Year)) %>%
  full_join(d.full) %>%
  select(Country, Year, Outcome, Event)
```

We assign zeros to the event not yet done, and keep NAs only for when the event is no more at risk. Then, we are assuming that once the event happens, it can not happen again, the country is no more at risk of considering it.

```
d.event <-
  d.event %>%
  mutate(Event = ifelse(is.na(Event), 0, Event)) %>%
  group_by(Country, Outcome) %>%
  arrange(Country, Outcome, Year) %>%
  mutate(sumEvent = cumsum(Event)) %>%
  mutate(Event = ifelse(sumEvent > 0 & Event != 1, NA, Event)) %>%
  rename(Adopted = sumEvent) %>%
  ungroup() %>%
  filter(!is.na(Year))
```

2.2 *Covariates -*

GDP per capita, World Bank. "GDP per capita (constant 2010 US$)". Contains `iso2c`, which is useful to further identify countries.

```
load("../../adoption/wdi-gdppc.RData") # Loads gdp.pc
gdp.pc <- gdp.pc %>%
  mutate(`GDPpc (log)` = log(GDPpc))
```

State expenditure, World Bank. "General government final consumption expenditure (% of GDP)". Contains `iso2c`, which is useful to further identify countries.

```
load("../../adoption/wdi-state_expenditure.RData") # Loads state.expenditure
```

Tax revenue, World Bank. "Tax revenue (% of GDP)".

```
load("../../adoption/wdi-tax_revenue.RData") # Loads state.expenditure
```

CO2 emissions, World Bank. “CO2 emissions (metric tons per capita)”.

Problem with Sudan (2015:2018), as it has 0 tons, which means the log can not be taken. Then, half of the minimum value is assigned to be able to process the logarithm.

```
load("../../adoption/wdi-co2pc.RData") # Loads co2.pc
#filter(co2.pc, CO2pc == 0)
assign.co2pc <- min(co2.pc$CO2pc[co2.pc$CO2pc != 0], na.rm = TRUE) / 2
co2.pc <- co2.pc %>%
  mutate(CO2pc = ifelse(CO2pc == 0, assign.co2pc, CO2pc)) %>%
  mutate(`CO2pc (log)` = log(CO2pc)) %>%
  select(-CO2pc)
#filter(co2.pc, `CO2pc (log)` == min(`CO2pc (log)`, na.rm = TRUE))
```

Oil rents plus *Coal rents* World Bank, “Oil rents (% of GDP)” + “Coal rents (% of GDP)”.

```
load("../../adoption/wdi-oil_rents.RData") # loads oil.rents
load("../../adoption/wdi-coal_rents.RData") # loads coal.rents
ff.rents <- left_join(oil.rents, coal.rents) %>%
  mutate(`Fossil fuel rents` = `Oil rents` + `Coal rents`) %>%
  mutate(`Fossil fuel rents (log)` = log(`Fossil fuel rents` + 1))
```

Population, World Bank. “Population, total”.

```
load("../../adoption/wdi-population.RData") # Loads population
population <- population %>%
  mutate(`Population (log)` = log(Population))
```

Democracy: V-Dem, Electoral dimension is used.

```
load("~/est/country_data/v-dem-v9/v_dem-ra-1950_2018.RData") # loads vdem
vdem <- vdem %>%
  filter(Democracy == "Electoral") %>%
#  filter(Country %in% countries) %>%
  filter(Year >= min(time.span) & Year <= max(time.span)) %>%
  select(Country, iso3c,
         Year, `Democracy (Electoral)` = value)
```

Government effectiveness. Data retrieved manually from the World Bank page on Governance indicators. Only the “Government Effectiveness: Estimation” is used. The data is only available between 1996 and 2005.

```
load("~/est/country_data/governance_indicators-world_bank/full-version/wgi-full-v191112.RData")
gov.eff <- wgi %>%
  tibble() %>%
  filter(indicator == "Government effectiveness") %>%
  rename(Country = country, Year = year) %>%
  rename(`Government effectiveness` = Estimate) %>%
  #select(Country, Year, `Government effectiveness`) %>%
  select(country.code, Year, `Government effectiveness`) %>%
  #arrange(Country, Year)
  arrange(country.code, Year)
```

We copy values some years:

- 1990 to 1995 are assigned the value of 1996.
- 1997, 1999, 2001 and 2019 are assigned the value of the previous year

```
gov.eff <- bind_rows(gov.eff,
  #
  # 1996 assigned to 1991:1995
  #
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1990),
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1991),
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1992),
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1993),
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1993),
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1995),
  #
  # assign values of previous years
  #
  filter(gov.eff, Year == 1996) %>%
    mutate(Year = 1997),
  filter(gov.eff, Year == 1998) %>%
    mutate(Year = 1999),
  filter(gov.eff, Year == 2000) %>%
    mutate(Year = 2001),
  filter(gov.eff, Year == 2018) %>%
    mutate(Year = 2019))
```

Political constraints from Henisz (2002).

```
load("~/est/country_data/polcon/polcon2017.RData") # loads polcon
polcon <- polcon %>%
  tibble() %>%
  select(Country.Polity = country.polity,
         #CountryWB = country.wb,
         #iso3c = country.wb, # not really the World Bank, but the iso3c
         #ioc = country.wb, # not really the World Bank, but the ioc
         #cowc = country.wb, # not really the World Bank, but the COW code
         #wb3c = country.wb,
         Year = year, `Political constraints` = polcon) %>%
  mutate(Year = as.integer(Year)) %>%
  filter(Year >= min(time.span) & Year <= max(time.span))
```

Debt, from the IMF "Debt % of GDP"

```
load("~/est/country_data/debt/imf_debt.RData")
debt <- imf.debt %>%
  filter(Year %in% time.span) %>%
```

```
  mutate(`Debt (log)` = log(Debt)) %>%
  select(-Debt)
```

Financial crises, from the IMF "Financial Crises" Binary status.

```
load("~/est/country_data/imf-financial_crises/financial_crises.RData")
financial.crises <- crises %>%
  filter(Year %in% time.span)
```

Climate change vulnerability, from the GAIN, as the ND-GAIN index. No data from before 1994 neither after 2017, and it is copied backwards and onwards.

We do not take the vulnerability index per se, but we take the means of two of the three dimensions of the index, namely exposure and sensitivity, letting aside adaptive capacity, that is the socio-economical compendium that we already have.

```
# Loads ccv
load("~/est/country_data/climate_change_vulnerability/200305-climate_change_vulnerability.RData")

ccv <- bind_rows(ccv,
  #
  # 1995 assigned to 1990:1994
  #
  filter(ccv, Year == 1996) %>%
    mutate(Year = 1990),
  filter(ccv, Year == 1996) %>%
    mutate(Year = 1991),
  filter(ccv, Year == 1996) %>%
    mutate(Year = 1992),
  filter(ccv, Year == 1996) %>%
    mutate(Year = 1993),
  filter(ccv, Year == 1996) %>%
    mutate(Year = 1994),
  #
  # assign values of 2017 to 2018:2019
  filter(ccv, Year == 2017) %>%
    mutate(Year = 2018),
  filter(ccv, Year == 2017) %>%
    mutate(Year = 2019))
```

EU membership

```
load("../../adoption/eu.RData") # Loads eum
```

Subnational tax

```
subn.tax.plain <- orig %>%
  select(Country, `Subnational tax` = Subnational_tax) %>%
  mutate(application = ifelse(!is.na(`Subnational tax`), 1, `Subnational tax`))
subn.tax <- expand.grid(Country = country.coverage, Year = time.span) %>% #, `Subnational tax` = 0) %>%
  tibble() %>%
  left_join(subn.tax.plain, by = c("Country" = "Country", "Year" = "Subnational tax")) %>%
```

```
  mutate(`Subnational tax` = 0) %>%
  group_by(Country) %>%
  arrange(Country, Year) %>%
  mutate(`Subnational tax` = `Subnational tax` + application) %>%
  mutate(`Subnational tax` = if_else(is.na(`Subnational tax`), 0, `Subnational tax`)) %>%
  mutate(`Subnational tax` = cumsum(`Subnational tax`)) %>%
  ungroup() %>%
  filter(Year %in% time.span) %>%
  select(Country, Year, `Subnational tax`)
```

Ratification of Kyoto protocol and Paris agreement

```
rat ← expand.grid(Country = country.coverage, Year = 1970:2020,
                  Agreement = c("Kyoto Ratification", "Paris Ratification")) %>%
  tibble()

d.ratification ← read.csv("ratification.csv", na.strings = "NA") %>%
  as_tibble() %>%
  select(-iso2c, -iso3c, -WorldBank) %>%
  gather(Agreement, Year, -Country) %>%
  mutate(Ratification = if_else(!is.na(Year), 1, 0))

ratification ← rat %>%
  left_join(d.ratification) %>%
  group_by(Country, Agreement) %>%
  arrange(Country, Agreement, Year) %>%
  mutate(Ratification = if_else(is.na(Ratification), 0, Ratification)) %>%
  mutate(Ratification = cumsum(Ratification)) %>%
  ungroup() %>%
  filter(Year %in% time.span) %>%
  spread(Agreement, Ratification)
```

Borders, as shared borders (only mainland), to generate weighting matrices. Comes from CEPII, Mayer and Zignago (2011).

```
# This "load()" is not required here, but in the models
load("../../adoption/geography.RData") # M.distances and M.borders
# RW.M.distances is a row-weighted matrix of distances (multiplied by 100)
```

Trade as bilateral trade, to generate weighting matrices. Comes from the Correlates of War v4.0..

```
# This "load()" is not required here, but in the models
load("../../adoption/trade.RData") # loads trade

d ← d.event %>%
  mutate(iso3c = countrycode(Country, origin = "country.name.en", destination = "iso3c")) %>%
  mutate(iso2c = countrycode(Country, origin = "country.name.en", destination = "iso2c")) %>%
  full_join(select(gdp.pc, -Country), by = c("iso2c" = "iso2c", "Year" = "Year")) %>%
  full_join(select(state.expenditure, -Country), by = c("iso2c" = "iso2c", "Year" = "Year")) %>%
  filter(!is.na(Country)) %>% # iso2c is SO, for Somalia
  full_join(select(vdem, -Country), by = c("iso3c" = "iso3c", "Year" = "Year")) %>%
  filter(!is.na(Country)) # iso3c are DDR PSG SML SOM XHX YMD ZZB
```

```
# Join CO2 emissions
d ← d %>%
  left_join(select(co2.pc, -Country),
            by = c("iso2c" = "iso2c", "Year" = "Year"))

# Join fossil fuel rents
d ← d %>%
  left_join(select(ff.rents, -Country),
            by = c("iso2c" = "iso2c", "Year" = "Year"))



# Join population
d ← d %>%
  left_join(select(population, -Country),
            by = c("iso2c" = "iso2c", "Year" = "Year"))


# Painful join with PolCon
d ← d %>%
##  # Load World Bank API 3c names for matching Polcon
#  mutate(wb3c = countrycode(iso3c, origin = "iso3c", destination = "wb_api3c")) %>%
#  full_join(select(polcon, -Country), by = c("wb3c" = "CountryWB", "Year" = "Year"))# %>%
#  # Load World Bank API 3c names for matching Polcon
#  mutate(ioc = countrycode(iso3c, origin = "iso3c", destination = "ioc")) %>%
#  full_join(select(polcon, -Country), by = c("ioc" = "ioc", "Year" = "Year"))# %>%
# CUW, FRO, GIB, GRL, IMN, MAC, MAF, MNP, NCL, PYF, SXM, TCA, VIR
#  mutate(cowc = countrycode(iso3c, origin = "iso3c", destination = "cowc")) %>%
#  full_join(select(polcon, -Country), by = c("cowc" = "cowc", "Year" = "Year"))# %>%
  mutate(p4.name = countrycode(iso3c, origin = "iso3c", destination = "p4.name")) %>%
  mutate(p4.name = ifelse(is.na(p4.name), Country, p4.name)) %>%
  #full_join(select(polcon, -Country), by = c("cowc" = "cowc", "Year" = "Year"))# %>%
  left_join(polcon, by = c("p4.name" = "Country.Polity", "Year" = "Year")) %>%
  select(-p4.name)



# Join with GovEff
d ← d %>%
  mutate(country.code = countrycode(iso3c, origin = "iso3c", destination = "wb")) %>%
  left_join(gov.eff, by = c("country.code" = "country.code", "Year" = "Year")) %>%
  filter(Year ≥ min(time.span) & Year ≤ max(time.span)) %>%
  select(-country.code) %>%
  unique()

# Join with IMF Debt
d ← d %>%
#tmp ← d %>%
  #mutate(Country.WB = countrycode(iso3c, origin = "iso3c", destination = "wb.name")) %>%
  #mutate(Country.WB = countrycode(iso3c, origin = "iso3c", destination = "country.name")) %>%
```

```
  left_join(debt %>%
            select(-Source, -Country))
#          by = c("iso3c" = "Country.WB", "Year" = "Year")) %>%
#  select(-Country.WB)

#filter(tmp, is.na(`Debt (log)`) & Year == 2015 & Outcome == "ETS")
# 58 missing with WDI 1.2.0 and destination = "country.name"
# 51 missing with WDI 1.1.2 and destination = "wb.name"


# Join with IMF Financial crises
d <- d %>%
#  mutate(Country.WB = countrycode(iso3c, origin = "iso3c", destination = "wb.name")) %>%
  left_join(financial.crises %>%
            select(-country.code, -Country, -imf))# %>%
#             rename(Country.WB = Country),
#          by = c("Country.WB" = "Country.WB", "Year" = "Year")) %>%
#  select(-Country.WB)



# Join with Climate change vulnerability
d <- d %>%
  left_join(select(ccv, -Country), by = c("iso3c" = "iso3c", "Year" = "Year")) %>%
#  rename(Vulnerability = `Vulnerability (Exposure, Sensitivity)`)
  rename(Vulnerability = Exposure)


# Join with EU membership
d <- d %>%
  left_join(eum, by = c("Country" = "Country", "Year" = "Year"))


# Join with subnational tax
d <- d %>%
  left_join(subn.tax, by = c("Country" = "Country", "Year" = "Year"))


# Join with ratification
d <- d %>%
  left_join(ratification, by = c("Country" = "Country", "Year" = "Year"))


# Clean
d <- d %>%
  select(Country, iso2c, iso3c, Year,
         Outcome, Event, Adopted,
         `GDPpc (log)`, `State expenditure`,
         `Population (log)`,
         `Democracy (Electoral)`,
         `Political constraints`, `Government effectiveness`,
```

```
         `Debt (log)`, `Financial crisis`,
         `Vulnerability`, `CO2pc (log)`,
         `Fossil fuel rents (log)`,
         `EU`, `Subnational tax`,
         `Kyoto Ratification`, `Paris Ratification`) %>%
  arrange(Country, Year)

if (length(which(is.na(d$Country))) != 0) stop("There are countries with NA in their name.")

# Remove countries with no name
d <- d %>%
  filter(!is.na(Country))

# Manually assign Austria 2015 onwards again at risk of Carbon tax
d <- d %>%
  mutate(Event = if_else(Country == "Australia" &
                         Year >= 2015 &
                         Outcome == "Carbon tax", 0, Event))
```

We start with the following number of countries

```
countries.start <- unique(d$Country)
length(countries.start)
```

```
→ [1] 216
```

For the cleaning, we avoid countries with too much missing data:

- Countries for which we do not have *ANY* data.
- Countries for which the mean of missingness is above a threshold.
- Countries for which the missingness in at least one of the variables is above a threshold (not implemented).

```
# This is the full list
uc <- tibble::tibble(`Potential countries` = unique(d$Country))
tc <- "List of all potential countries."
if (knitr::is_latex_output()) {
  kable(uc, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(latex_options = c("hold_position", "repeat_header"))
} else {
  kable(uc, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(bootstrap_options = "striped", full_width = FALSE)
}

d.prob.missing <- d %>%
  select(-iso2c, -iso3c, -Outcome, -Event) %>%
  gather(Variable, value, -Country, -Year) %>%
  group_by(Country, Variable) %>%
  summarize(pMissing = length(which(is.na(value))) / n())
d.full.missing <- d.prob.missing %>%
  group_by(Country) %>%
  summarize(FullMissing = ifelse(sum(pMissing) == n(), TRUE, FALSE)) %>%
  filter(FullMissing)
```

```
threshold.missingness <- 0.5
d.partial.missing <- d.prob.missing %>%
  group_by(Country) %>%
  summarize(MeanMissing = mean(pMissing)) %>%
  filter(MeanMissing > threshold.missingness)


threshold.some.missingness <- 0.9
d.partial.some.missing <- d.prob.missing %>%
  group_by(Country) %>%
  summarize(SomeMissing = if_else(any(pMissing > threshold.some.missingness), TRUE, FALSE)) %>%
  filter(SomeMissing > threshold.some.missingness)


d <- d %>%
  filter(!Country %in% c(d.full.missing$Country, d.partial.missing$Country))
```

We finish with the following number of countries

```
countries.universe <- unique(d$Country)
length(countries.universe)
```

```
→ [1] 203
```

The maximum proportion that, overall, any country can have on missing data is:

```
threshold.missingness
```

```
→ [1] 0.5
```

List of countries considered

```
uc <- tibble::tibble(`Countries in the analysis` = countries.universe)
tc <- "List of countries in the analysis."
if (knitr::is_latex_output()) {
  kable(uc, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    #kable_styling(latex_options = c("hold_position", "repeat_header"))
    kable_styling(latex_options = c("hold_position"))
} else {
  kable(uc, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(bootstrap_options = "striped", full_width = FALSE)
}
```

Table 2.1: List of countries in the analysis.

Countries in the analysis
Afghanistan
Albania
Algeria
Andorra
Angola

Antigua & Barbuda
Argentina
Armenia
Aruba
Australia

Austria
Azerbaijan
Bahamas
Bahrain
Bangladesh

Barbados
Belarus
Belgium
Belize
Benin

Bermuda
Bhutan
Bolivia
Bosnia & Herzegovina
Botswana

Brazil
Brunei
Bulgaria
Burkina Faso
Burundi

Cambodia
Cameroon
Canada
Cape Verde
Cayman Islands

Central African Republic
Chad
Chile
China
Colombia

Comoros
Congo - Brazzaville
Congo - Kinshasa
Costa Rica
Côte d'Ivoire

Croatia
Cuba
Cyprus
Czechia
Denmark

Djibouti

Dominica
Dominican Republic
Ecuador
Egypt
El Salvador
Equatorial Guinea
Eritrea
Estonia
Eswatini
Ethiopia
Faroe Islands
Fiji
Finland
France
Gabon
Gambia
Georgia
Germany
Ghana
Greece
Greenland
Grenada
Guatemala
Guinea
Guinea-Bissau
Guyana
Haiti
Honduras
Hong Kong SAR China
Hungary
Iceland
India
Indonesia
Iran
Iraq
Ireland
Israel
Italy
Jamaica
Japan
Jordan
Kazakhstan
Kenya
Kiribati
Kuwait
Kyrgyzstan

Laos
Latvia
Lebanon
Lesotho
Liberia
Libya
Liechtenstein
Lithuania
Luxembourg
Macao SAR China
Madagascar
Malawi
Malaysia
Maldives
Mali
Malta
Marshall Islands
Mauritania
Mauritius
Mexico
Micronesia (Federated States of)
Moldova
Monaco
Mongolia
Montenegro
Morocco
Mozambique
Myanmar (Burma)
Namibia
Nauru
Nepal
Netherlands
New Zealand
Nicaragua
Niger
Nigeria
North Korea
North Macedonia
Norway
Oman
Pakistan
Palau
Palestinian Territories
Panama
Papua New Guinea
Paraguay

Peru
Philippines
Poland
Portugal
Puerto Rico
Qatar
Romania
Russia
Rwanda
Samoa
San Marino
São Tomé & Príncipe
Saudi Arabia
Senegal
Serbia
Seychelles
Sierra Leone
Singapore
Slovakia
Slovenia
Solomon Islands
Somalia
South Africa
South Korea
South Sudan
Spain
Sri Lanka
St. Kitts & Nevis
St. Lucia
St. Vincent & Grenadines
Sudan
Suriname
Sweden
Switzerland
Syria
Taiwan
Tajikistan
Tanzania
Thailand
Timor-Leste
Togo
Tonga
Trinidad & Tobago
Tunisia
Turkey
Turkmenistan

Tuvalu
Uganda
Ukraine
United Arab Emirates
United Kingdom
United States
Uruguay
Uzbekistan
Vanuatu
Venezuela
Vietnam
Yemen
Zambia
Zimbabwe

List of countries in the initial list that are no more part of the universe due to severe missingness.

```
countries.removed ← countries.start[which(!countries.start %in% countries.universe)]

uc ← tibble::tibble(`Countries removed` = countries.removed)
tc ← "List of countries removed due to severe missingness."
if (knitr::is_latex_output()) {
  kable(uc, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    #kable_styling(latex_options = c("hold_position", "repeat_header"))
    kable_styling(latex_options = c("hold_position"))
} else {
  kable(uc, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(bootstrap_options = "striped", full_width = FALSE)
}
```

Table 2.2: List of countries removed due to severe missingness.

Countries removed
American Samoa
British Virgin Islands
Curaçao
French Polynesia
Gibraltar
Guam
Isle of Man
New Caledonia
Northern Mariana Islands
Saint Martin (French part)
Sint Maarten
Turks & Caicos Islands

U.S. Virgin Islands

```
save(d,
     file = "carbon_pricing-adoption.RData")
```

3
Data description

```r
load("carbon_pricing-adoption.RData")
load("details.RData")
```

Number of countries

```r
length(unique(d$Country))
```

```
→ [1] 203
```

Cumulative hazard rate (Figure 3.1) following a simple survival model with no covariates.

```r
library(survival)
ch <- NULL
for (o in 1:length(unique(d$Outcome))) {
  mf0 <- survfit(Surv(Year, Event) ~ 1,
                 data = filter(d, Outcome == unique(d$Outcome)[o]))
  d.mf0 <- tibble::tibble(Year = summary(mf0)$time,
                          `Cumulative hazard` = summary(mf0)$cumhaz,
                          Outcome = unique(d$Outcome)[o])
  ch <- bind_rows(ch, d.mf0)
}

ggplot(ch, aes(x = Year, y = `Cumulative hazard`, color = Outcome)
  geom_line()
```

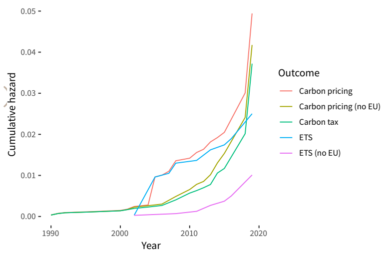


Figure 3.1: Cumulative hazard.

There are very few instances of events actually happening:

```r
# Proportion of event happening
peh <- d %>%
  group_by(Outcome, Event) %>%
  summarize(N = n()) %>%
  group_by(Outcome) %>%
  filter(!is.na(Event)) %>%
  mutate(Proportion = N / sum(N))

tc <- "Proportion of events happening"
if (knitr::is_latex_output()) {
  kable(peh, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE)
} else {
  kable(peh, format = "html", caption = tc, booktabs = TRUE) %>%
```

```
    kable_styling(bootstrap_options = "striped", full_width = FALSE)
}
```

Table 3.1: Proportion of events happening

Outcome	Event	N	Proportion
Carbon pricing	0	5399	0.9906
Carbon pricing	1	51	0.0094
Carbon pricing (no EU)	0	5655	0.9937
Carbon pricing (no EU)	1	36	0.0063
Carbon tax	0	5711	0.9946
Carbon tax	1	31	0.0054
ETS	0	5586	0.9931
ETS	1	39	0.0069
ETS (no EU)	0	6027	0.9987
ETS (no EU)	1	8	0.0013

Events happen at the following times (Figure 3.2).

```
et <- d %>%
  filter(Event == 1) %>%
  group_by(Outcome, Year) %>%
  summarize(`Number of events` = n())
ggplot(et, aes(x = Year, y = 0, size = `Number of events`)) +
  geom_point() +
  facet_grid(Outcome ~ .) +
  ylab("") +
  scale_size_continuous(breaks = sort(unique(et$`Number of events`))) +
  theme(axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank())

et <- d %>%
  filter(Event == 1) %>%
  filter(Outcome == "Carbon pricing") %>%
  group_by(Year) %>%
  summarize(`Adoptions` = n())
f1 <- ggplot(et, aes(xmin = Year - 0.5, xmax = Year + 0.5,
              ymin = 0, ymax = Adoptions)) +
  geom_rect() +
  xlab("Year") + ylab("Adoptions")

et <- d %>%
  filter(Outcome == "Carbon pricing") %>%
  group_by(Year, Adopted) %>%
  summarize(nAdopted = n()) %>%
  ungroup() %>%
  spread(Adopted, nAdopted) %>%
```

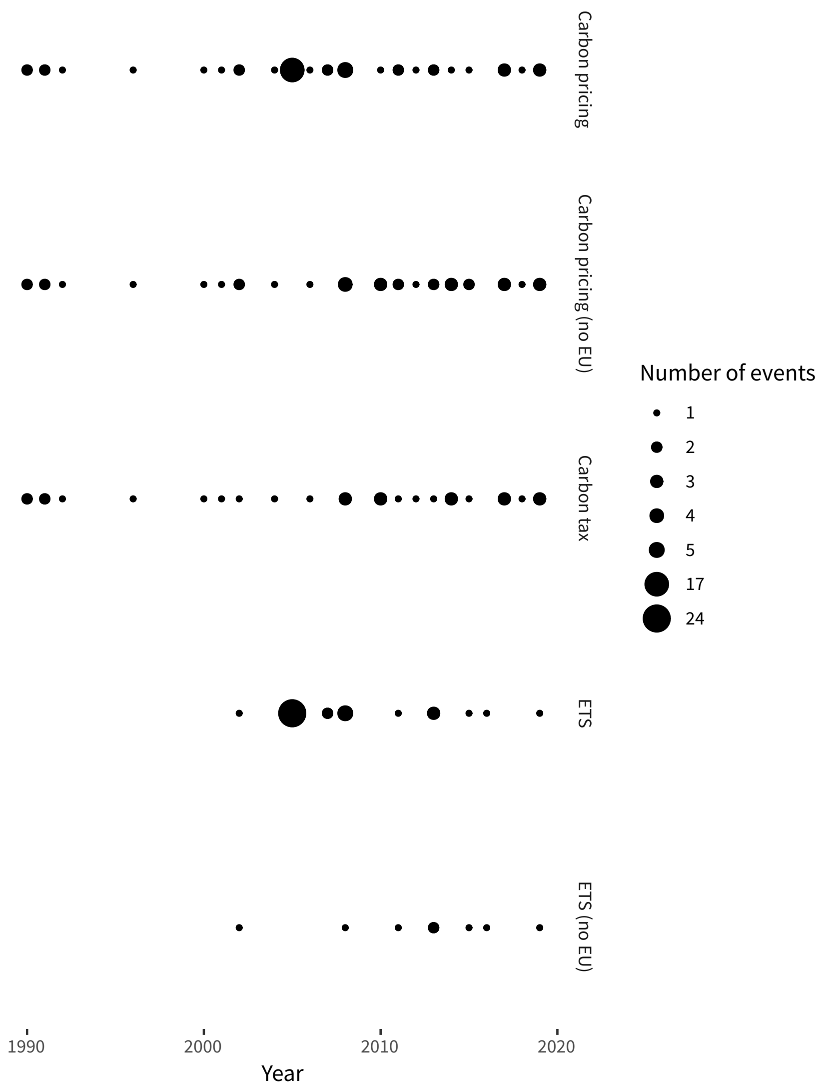


Figure 3.2: Temporal distribution of events.

```
  mutate(`Cumulative adoptions (proportion)` = `1` / (`0` + `1`))
f2 <- ggplot(et, aes(x = Year, y = `Cumulative adoptions (proportion)`)) +
  geom_line()

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

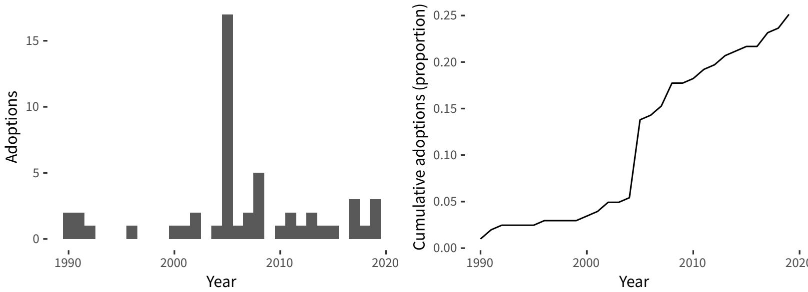


Figure 3.3: Carbon pricing policy diffusion: temporal distribution of adoptions and cumulative proportion of adopters.

Distribution of missingness by covariates and event.

```
m.cov.ev <- d %>%
  ungroup() %>%
  filter(!is.na(Event)) %>%
  select(-Country, -iso2c, -iso3c, -Year) %>%
  gather(Variable, value, -Event, -Outcome) %>%
  group_by(Outcome, Variable, Event) %>%
  summarize(pMissing = length(which(is.na(value))) / n()) %>%
  ungroup() %>%
  spread(Event, pMissing) %>%
  rename(`Any policy` = `0`, Policy = `1`) %>%
  arrange(desc(Policy))

tc <- "Distribution of missing values by variable and event, sorted by
proportion of missingness in the policy adoption column."
if (knitr::is_latex_output()) {
  kable(m.cov.ev, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 8)
} else {
  kable(m.cov.ev, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 8, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.2: Distribution of missing values by variable and event, sorted by proportion of missingness in the policy adoption column.

Outcome	Variable	Any policy	Policy
Carbon tax	Debt (log)	0.2665	0.2581
Carbon tax	Financial crisis	0.2042	0.2581
Carbon tax	Political constraints	0.3416	0.2581
Carbon pricing (no EU)	Political constraints	0.3415	0.2500
ETS (no EU)	Debt (log)	0.2671	0.2500
ETS (no EU)	Financial crisis	0.2076	0.2500
ETS (no EU)	Political constraints	0.3350	0.2500
Carbon pricing (no EU)	Debt (log)	0.2649	0.2222
Carbon pricing (no EU)	Financial crisis	0.2019	0.2222

Carbon pricing	Political constraints	0.3449	0.2157
Carbon pricing	Debt (log)	0.2663	0.1569
Carbon pricing	Financial crisis	0.2004	0.1569
Carbon tax	Democracy (Electoral)	0.1802	0.1290
Carbon tax	State expenditure	0.2168	0.1290
ETS	Political constraints	0.3389	0.1282
ETS (no EU)	Democracy (Electoral)	0.1762	0.1250
Carbon pricing (no EU)	Democracy (Electoral)	0.1809	0.1111
Carbon pricing (no EU)	State expenditure	0.2189	0.1111
Carbon tax	Fossil fuel rents (log)	0.1138	0.0968
Carbon tax	GDPpc (log)	0.0795	0.0968
Carbon pricing (no EU)	Fossil fuel rents (log)	0.1149	0.0833
Carbon pricing (no EU)	GDPpc (log)	0.0803	0.0833
Carbon pricing	Democracy (Electoral)	0.1867	0.0784
Carbon pricing	State expenditure	0.2293	0.0784
ETS	Debt (log)	0.2646	0.0769
ETS	Financial crisis	0.2003	0.0769
Carbon tax	CO2pc (log)	0.0574	0.0645
Carbon tax	Population (log)	0.0151	0.0645
Carbon pricing	Fossil fuel rents (log)	0.1204	0.0588
Carbon pricing	GDPpc (log)	0.0841	0.0588
Carbon pricing (no EU)	CO2pc (log)	0.0580	0.0556
Carbon pricing (no EU)	Population (log)	0.0152	0.0556
ETS	Democracy (Electoral)	0.1826	0.0513
Carbon pricing	CO2pc (log)	0.0608	0.0392
Carbon pricing	Population (log)	0.0159	0.0392
ETS	Fossil fuel rents (log)	0.1190	0.0256
ETS	GDPpc (log)	0.0840	0.0256
ETS	Government effectiveness	0.1072	0.0256
ETS	State expenditure	0.2252	0.0256
Carbon pricing	Government effectiveness	0.1100	0.0196
Carbon pricing	Adopted	0.0000	0.0000
Carbon pricing	EU	0.0222	0.0000
Carbon pricing	Kyoto Ratification	0.0000	0.0000
Carbon pricing	Paris Ratification	0.0000	0.0000
Carbon pricing	Subnational tax	0.0000	0.0000
Carbon pricing	Vulnerability	0.0611	0.0000
Carbon pricing (no EU)	Adopted	0.0000	0.0000
Carbon pricing (no EU)	EU	0.0212	0.0000
Carbon pricing (no EU)	Government effectiveness	0.1073	0.0000
Carbon pricing (no EU)	Kyoto Ratification	0.0000	0.0000
Carbon pricing (no EU)	Paris Ratification	0.0000	0.0000
Carbon pricing (no EU)	Subnational tax	0.0000	0.0000
Carbon pricing (no EU)	Vulnerability	0.0584	0.0000
Carbon tax	Adopted	0.0000	0.0000
Carbon tax	EU	0.0210	0.0000
Carbon tax	Government effectiveness	0.1063	0.0000
Carbon tax	Kyoto Ratification	0.0000	0.0000
Carbon tax	Paris Ratification	0.0000	0.0000
Carbon tax	Subnational tax	0.0000	0.0000
Carbon tax	Vulnerability	0.0578	0.0000
ETS	Adopted	0.0000	0.0000
ETS	CO2pc (log)	0.0614	0.0000
ETS	EU	0.0215	0.0000
ETS	Kyoto Ratification	0.0000	0.0000
ETS	Paris Ratification	0.0000	0.0000
ETS	Population (log)	0.0181	0.0000
ETS	Subnational tax	0.0000	0.0000
ETS	Vulnerability	0.0591	0.0000
ETS (no EU)	Adopted	0.0000	0.0000
ETS (no EU)	CO2pc (log)	0.0569	0.0000
ETS (no EU)	EU	0.0199	0.0000

ETS (no EU)	Fossil fuel rents (log)	0.1123	0.0000
ETS (no EU)	GDPpc (log)	0.0796	0.0000
ETS (no EU)	Government effectiveness	0.1015	0.0000
ETS (no EU)	Kyoto Ratification	0.0000	0.0000
ETS (no EU)	Paris Ratification	0.0000	0.0000
ETS (no EU)	Population (log)	0.0168	0.0000
ETS (no EU)	State expenditure	0.2107	0.0000
ETS (no EU)	Subnational tax	0.0000	0.0000
ETS (no EU)	Vulnerability	0.0548	0.0000

Countries with higher missing values in the events.

```
m.cov.na.country <- d %>%
  ungroup() %>%
  filter(!is.na(Event)) %>%
  select(-Outcome, -Event) %>%
  unique() %>%
  select(-iso2c, -iso3c, -Year) %>%
  gather(Variable, value, -Country) %>%
  group_by(Country, Variable) %>%
  summarize(pMissing = length(which(is.na(value))) / n()) %>%
  ungroup() %>%
  arrange(desc(pMissing))

m.cov.na.country.top <- m.cov.na.country %>%
  filter(pMissing == 1) %>%
  select(-pMissing)

tc <- "Pairs of countries/variables where all data is missing."
if (knitr::is_latex_output()) {
  kable(m.cov.na.country.top, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 8)
} else {
  kable(m.cov.na.country.top, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 8, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 3.3: Pairs of countries/variables where all data is missing.

Country	Variable
Andorra	Debt (log)
Andorra	Democracy (Electoral)
Andorra	Financial crisis
Andorra	Fossil fuel rents (log)
Andorra	Government effectiveness
Andorra	Political constraints
Andorra	State expenditure
Antigua & Barbuda	Democracy (Electoral)
Antigua & Barbuda	Political constraints
Antigua & Barbuda	State expenditure
Aruba	Debt (log)
Aruba	Democracy (Electoral)
Aruba	Financial crisis
Aruba	Political constraints
Aruba	Vulnerability

Bahamas	Democracy (Electoral)
Bahamas	Political constraints
Barbados	Political constraints
Belize	Democracy (Electoral)
Belize	Political constraints
Bermuda	Debt (log)
Bermuda	Democracy (Electoral)
Bermuda	Financial crisis
Bermuda	Political constraints
Bermuda	Vulnerability
Brunei	Democracy (Electoral)
Brunei	Political constraints
Cape Verde	Political constraints
Cayman Islands	Debt (log)
Cayman Islands	Democracy (Electoral)
Cayman Islands	Financial crisis
Cayman Islands	Political constraints
Cayman Islands	State expenditure
Cayman Islands	Vulnerability
Central African Republic	Financial crisis
Central African Republic	Political constraints
Congo - Brazzaville	Political constraints
Congo - Kinshasa	Government effectiveness
Congo - Kinshasa	Political constraints
Cuba	Debt (log)
Cuba	Financial crisis
Dominica	Democracy (Electoral)
Dominica	Political constraints
Dominica	State expenditure
Dominican Republic	Political constraints
Eswatini	EU
Faroe Islands	Debt (log)
Faroe Islands	Democracy (Electoral)
Faroe Islands	Financial crisis
Faroe Islands	Government effectiveness
Faroe Islands	Political constraints
Faroe Islands	Vulnerability
Fiji	State expenditure
Greenland	Debt (log)
Greenland	Democracy (Electoral)
Greenland	Financial crisis
Greenland	Political constraints
Greenland	Vulnerability
Grenada	Democracy (Electoral)
Grenada	Political constraints
Grenada	State expenditure
Hong Kong SAR China	Financial crisis
Hong Kong SAR China	Political constraints
Hong Kong SAR China	Vulnerability
Kiribati	Democracy (Electoral)
Kiribati	Political constraints
Liechtenstein	Debt (log)
Liechtenstein	Democracy (Electoral)
Liechtenstein	Financial crisis
Liechtenstein	Fossil fuel rents (log)
Liechtenstein	Political constraints
Liechtenstein	State expenditure
Macao SAR China	Debt (log)
Macao SAR China	Democracy (Electoral)
Macao SAR China	EU
Macao SAR China	Financial crisis
Macao SAR China	Political constraints

Macao SAR China	Vulnerability
Maldives	Political constraints
Maldives	State expenditure
Malta	Political constraints
Marshall Islands	Democracy (Electoral)
Marshall Islands	Fossil fuel rents (log)
Marshall Islands	Political constraints
Micronesia (Federated States of)	Democracy (Electoral)
Micronesia (Federated States of)	Financial crisis
Micronesia (Federated States of)	Fossil fuel rents (log)
Micronesia (Federated States of)	Political constraints
Micronesia (Federated States of)	State expenditure
Monaco	CO2pc (log)
Monaco	Debt (log)
Monaco	Democracy (Electoral)
Monaco	Financial crisis
Monaco	Fossil fuel rents (log)
Monaco	Government effectiveness
Monaco	Political constraints
Monaco	State expenditure
Montenegro	Political constraints
Nauru	Debt (log)
Nauru	Democracy (Electoral)
Nauru	Financial crisis
Nauru	Political constraints
Nauru	State expenditure
North Korea	Debt (log)
North Korea	Financial crisis
North Korea	Fossil fuel rents (log)
North Korea	GDPpc (log)
North Korea	Political constraints
North Korea	State expenditure
North Macedonia	EU
Palau	Debt (log)
Palau	Democracy (Electoral)
Palau	Fossil fuel rents (log)
Palau	Political constraints
Palestinian Territories	Debt (log)
Palestinian Territories	Financial crisis
Palestinian Territories	Government effectiveness
Palestinian Territories	Political constraints
Palestinian Territories	Vulnerability
Puerto Rico	CO2pc (log)
Puerto Rico	Democracy (Electoral)
Puerto Rico	Financial crisis
Puerto Rico	Political constraints
Puerto Rico	Vulnerability
Romania	Government effectiveness
Samoa	Democracy (Electoral)
Samoa	Political constraints
Samoa	State expenditure
San Marino	CO2pc (log)
San Marino	Democracy (Electoral)
San Marino	Fossil fuel rents (log)
San Marino	Government effectiveness
San Marino	Political constraints
San Marino	State expenditure
São Tomé & Príncipe	Political constraints
São Tomé & Príncipe	State expenditure
Seychelles	Political constraints
Slovakia	Political constraints
Solomon Islands	Political constraints

Somalia	Debt (log)
Somalia	EU
Somalia	GDPpc (log)
South Africa	CO_2pc (log)
South Africa	Fossil fuel rents (log)
South Africa	GDPpc (log)
South Africa	Population (log)
South Africa	State expenditure
South Korea	Political constraints
South Sudan	Political constraints
South Sudan	Vulnerability
St. Kitts & Nevis	Democracy (Electoral)
St. Kitts & Nevis	Political constraints
St. Kitts & Nevis	State expenditure
St. Lucia	Democracy (Electoral)
St. Lucia	Political constraints
St. Vincent & Grenadines	Democracy (Electoral)
St. Vincent & Grenadines	Political constraints
Sudan	Political constraints
Suriname	Political constraints
Syria	GDPpc (log)
Taiwan	CO_2pc (log)
Taiwan	Financial crisis
Taiwan	Fossil fuel rents (log)
Taiwan	GDPpc (log)
Taiwan	Population (log)
Taiwan	State expenditure
Taiwan	Vulnerability
Timor-Leste	Debt (log)
Timor-Leste	Government effectiveness
Timor-Leste	Political constraints
Tonga	Democracy (Electoral)
Tonga	Political constraints
Trinidad & Tobago	Political constraints
Trinidad & Tobago	State expenditure
Tuvalu	Democracy (Electoral)
Tuvalu	Fossil fuel rents (log)
Tuvalu	Political constraints
Tuvalu	State expenditure
Vanuatu	Political constraints
Yemen	State expenditure
Zambia	CO_2pc (log)
Zambia	Fossil fuel rents (log)
Zambia	GDPpc (log)
Zambia	Population (log)
Zambia	State expenditure

Distribution of events on values of covariates (Figure 3.4, for the covariates in the fifth year before the end of the period).

```
cov ← d %>%
#  filter(Year == max(time.span)) %>%
  select(-Outcome, -Event) %>%
  unique()

ev ← d %>%
  filter(Event == 1) %>%
  select(Country, Year, Outcome)
```

```
#cov.ev <- left_join(cov, ev) %>%
#cov.ev <- inner_join(ev, cov) %>%
cov.ev <- left_join(cov, ev) %>%
  gather(Variable, value, -Country, -iso2c, -iso3c, -Year, -Outcome) %>%
  mutate(Outcome = ifelse(is.na(Outcome), "Any policy", Outcome)) %>%
  unique()

ggplot(cov.ev, aes(x = value, y = Outcome)) +
  geom_point(data = filter(cov.ev, Year == max(Year)-5), alpha = 0.2) +
  geom_point(data = filter(cov.ev, Outcome != "Any policy"), alpha = 0.5, color = "red") +
  facet_wrap(Variable ~ ., scales = "free", ncol = 1)


# Calculate variable means and keep adoption in the last year aside
d.carbon.adoption.lastyear <- d %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(Year == max(Year)) %>%
  select(Country, Adopted) %>%
  mutate(`Carbon pricing` = ifelse(Adopted == 1, "Adopted", "Not adopted")) %>%
  select(-Adopted)

d.carbon.covs <- d %>%
  filter(Outcome == "Carbon pricing") %>%
  select(-c(iso2c, iso3c, Year, Event, Adopted,
            `Debt (log)`, `Paris Ratification`,
            Outcome)) %>%
  gather(Variable, value, -Country) %>%
  group_by(Country, Variable) %>%
  summarize(value = mean(value, na.rm = TRUE)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  mutate(Variable = str_replace(Variable, " ", "\n")) %>%
  spread(Variable, value)

d.carbon <- left_join(d.carbon.adoption.lastyear, d.carbon.covs)

my_dens <- function(data, mapping, ..., low = "#132B43", high = "#56B1F7") {
  ggplot(data = data, mapping=mapping) +
         geom_density(..., alpha=0.3) +
    scale_color_discrete_qualitative(palette = "Dark 2")
}

#my_points <- function(data, mapping, ...) {
#  ggplot(data = data, mapping = mapping) +
#    geom_point(..., alpha = 0.5) +
#    scale_color_discrete_qualitative(palette = "Dark 2")
#}
#my_boxplot <- function(data, mapping, ...) {
#  ggplot(data = data, mapping = mapping) +
#    geom_boxplot(...) +
#    scale_color_discrete_qualitative(palette = "Dark 2")
#}
#my_bar <- function(data, mapping, ...) {
```

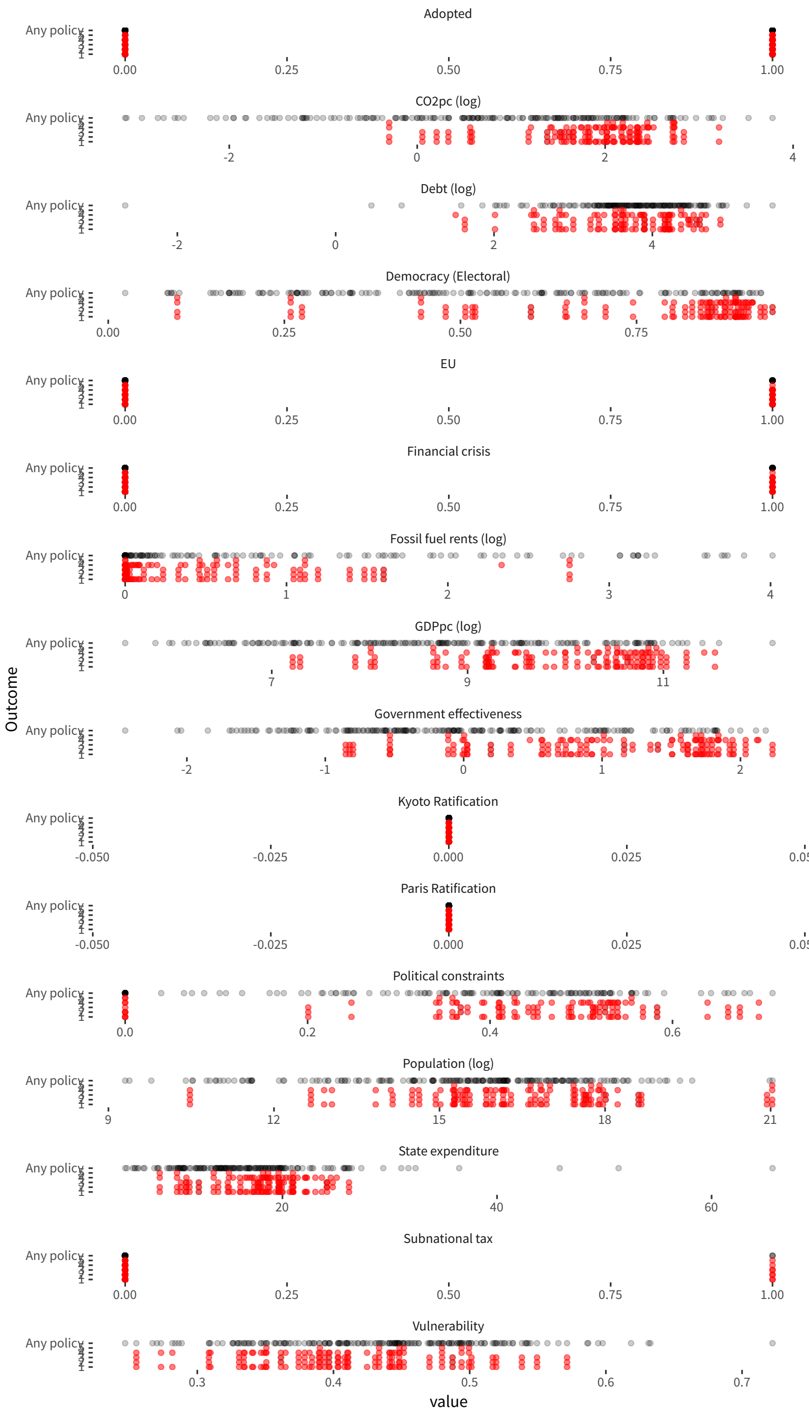


Figure 3.4: Distribution of the values of covariates and events. Covariates are only included in the fifth year before the end of the period.

```
#  ggplot(data = data, mapping = mapping) +
#    geom_bar( ... ) +
#    scale_color_discrete_qualitative(palette = "Dark 2") +
#    scale_fill_discrete_qualitative(palette = "Dark 2")
#}
#my_histogram <- function(data, mapping, ...) {
#  ggplot(data = data, mapping = mapping) +
#    geom_histogram( ... ) +
#    scale_color_discrete_qualitative(palette = "Dark 2") +
#    scale_fill_discrete_qualitative(palette = "Dark 2")
#}
#
#my_cor <- function(data, mapping, method="pearson", ndp=2, sz=5, stars=FALSE, ...) {
# x <- eval_data_col(data, mapping$x)
# y <- eval_data_col(data, mapping$y)
# corr <- cor.test(x, y, method=method)
# est <- corr$estimate
# lb.size <- sz* abs(est)
# if(stars){
#   stars <- c("***", "**", "*", "")[findInterval(corr$p.value,
#                                                c(0, 0.001,
#                                                  0.01, 0.05,
#                                                  1))]
#    lbl <- paste0(round(est, ndp), stars)
#  } else {
#    lbl <- round(est, ndp)
#  }
#  ggplot(data=data, mapping=mapping) +
#    annotate("text", x=mean(x, na.rm=TRUE), y=mean(y, na.rm=TRUE),
#             label=lbl, size=lb.size, ... ) +
#    theme(panel.grid = element_blank()) +
#    scale_color_discrete_qualitative(palette = "Dark 2")
#}
#ggpairs(select(d.carbon, -Country),
#  upper = list(continuous = my_cor,
#               combo = my_boxplot,
#               discrete = my_bar),
#  lower = list(continuous = my_points, combo = my_bar),
#  diag = list(continuous = my_dens, discrete = my_bar),
#  mapping = aes(color = `Carbon pricing`))#, fill = `Carbon pricing`))


ggplot <- function( ... ) {
  ggplot2::ggplot( ... ) +
    scale_color_discrete_qualitative(palette = "Dark 2") +
    scale_fill_discrete_qualitative(palette = "Dark 2")
}
unlockBinding("ggplot",parent.env(asNamespace("GGally")))
assign("ggplot",ggplot,parent.env(asNamespace("GGally")))

ggpairs(select(d.carbon, -Country),
```

```
  lower = list(continuous = wrap("points", alpha = 0.5)),
  diag = list(continuous = my_dens),
  mapping = aes(color = `Carbon pricing`))
```

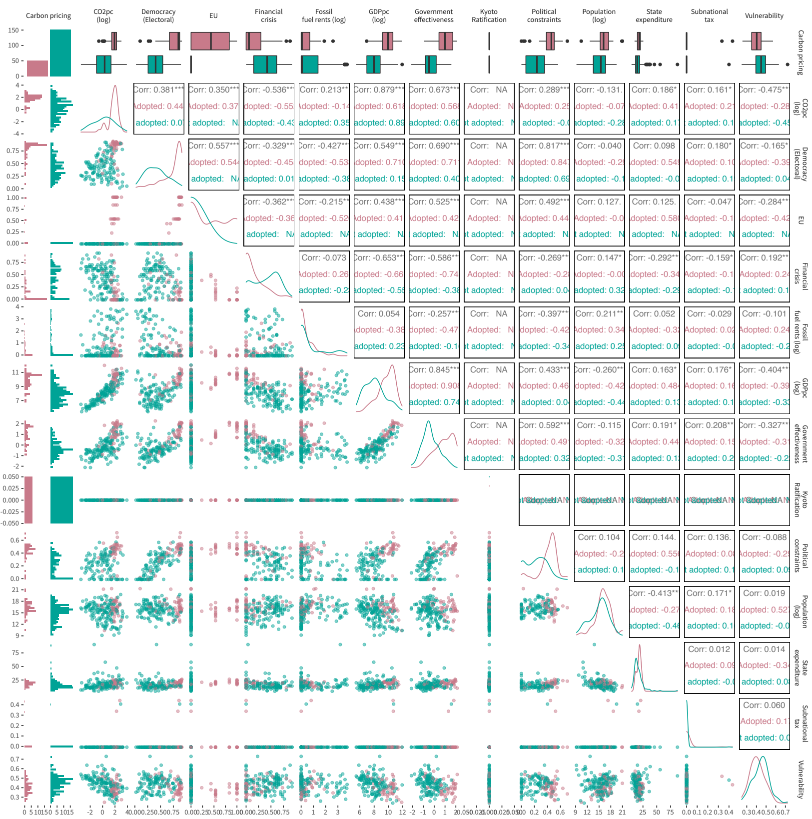


Figure 3.5: Distribution of the values of covariates, by carbon pricing adoption. Values represent country means for the period considered. Carbon adoption is taken on the last year considered.

```
ggplot(cov.ev, aes(x = value)) +
  geom_histogram() +
  facet_grid(Outcome ~ Variable, scales = "free")
```

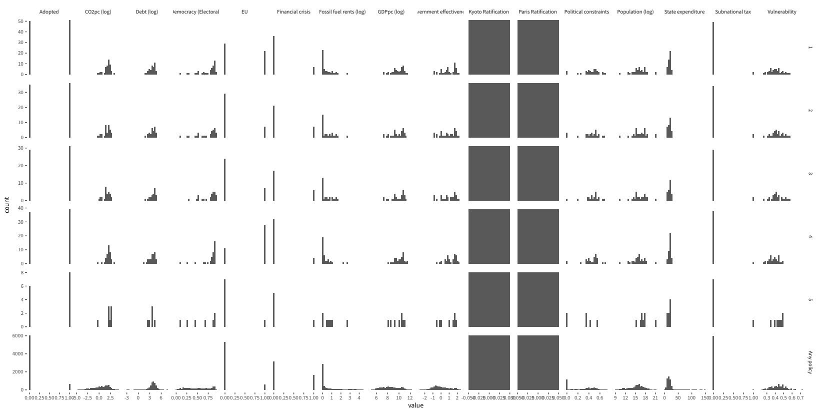


Figure 3.6: Distribution of the values of covariates by types of events.

```
ggplot(cov.ev, aes(x = value, color = Outcome)) +
  geom_density(data = filter(cov.ev, Outcome == "Any policy"), color = "black", lwd = 2) +
```

```
  geom_density(data = filter(cov.ev, Outcome ≠ "Any policy")) +
  facet_wrap(~ Variable, scales = "free")
```

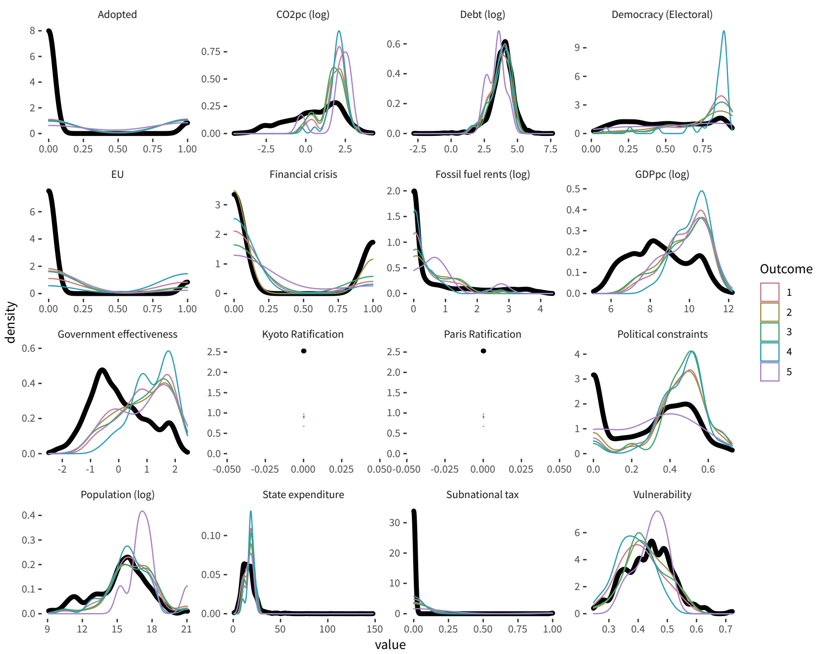


Figure 3.7: Distribution of the values of covariates by types of events. The thick black line correspondes to cases without any policy.

Correlation between covariates

```
library(corrplot)
X.cov ← d %>%
  select(#Country, Year,
         `GDPpc (log)`,
         `Population (log)`,
         `Democracy (Electoral)`,
         `Political constraints`,
         `Government effectiveness`,
         #`Debt (log)`,
         `Financial crisis`,
         `Vulnerability`,
         `CO2pc (log)`) %>%
  as.matrix()

corrplot(cor(X.cov, use = "complete.obs"), method = "number")
```

Ranges of explanatory variables

```
d %>%
  select(#Country, Year,
         `GDPpc (log)`,
         `Population (log)`,
         `Democracy (Electoral)`,
         `Political constraints`,
         `Government effectiveness`,
         `Vulnerability`,
```

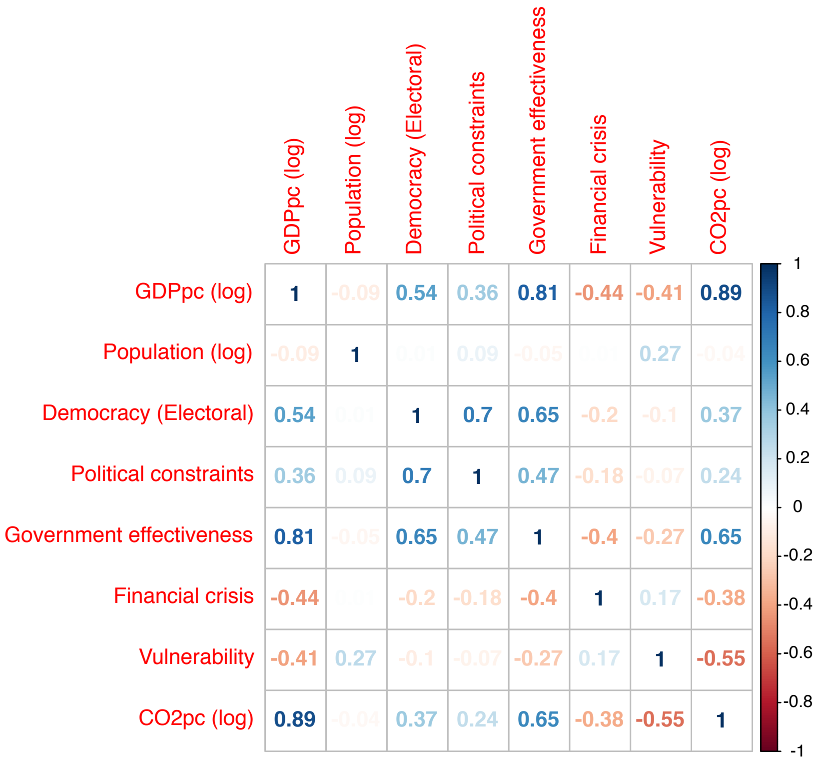


Figure 3.8: Correlation between explanatory variables.

```
      `CO2pc (log)`) %>%
  gather(Variable, value) %>%
  group_by(Variable) %>%
  summarize(Min = min(value, na.rm = TRUE),
            Mean = mean(value, na.rm = TRUE),
            Max = max(value, na.rm = TRUE))
```

```
→ # A tibble: 7 x 4
→   Variable                    Min    Mean    Max
→   <chr>                     <dbl>   <dbl>  <dbl>
→ 1 CO2pc (log)              -4.54    0.578   4.25
→ 2 Democracy (Electoral)     0.014   0.514   0.948
→ 3 GDPpc (log)               5.10    8.48   12.2
→ 4 Government effectiveness -2.48   -0.0127  2.44
→ 5 Political constraints     0       0.283   0.726
→ 6 Population (log)          9.10   15.3    21.1
→ 7 Vulnerability             0.247   0.435   0.722
```

4
Model: Baseline 001

```
# Model:
M
```

→ [1] "Baseline 001"

- Event history analysis using logistic regression for discrete time (years).
- Time dependence is fixed over the period, with a dummy in 2005 and for EU models to capture the effect of EU in a single period. More complex time dependency does not show any significant gain and complicates the estimation.
- Bayesian inference with weakly informative priors.

  Data preparation:

- Standardization to 0.5 standard deviations for all covariates, following Gelman (2008).

  Model equation:

$$
\begin{aligned}
Y_{t,c,o} &\sim & \mathcal{B}(\pi_{t,c,o}) & \quad \text{Main data component} \\
\pi_{t,c,o} &= & \text{logit}\alpha_o + \rho_o + (\theta_{o,v} X_{t,c,v}) & \quad \text{Main linear model} \\
\alpha_o &\sim & \mathcal{N}(-7, 2) & \quad \text{Rare event prior} \\
\rho_o &\sim & \mathcal{N}(0, 2) & \quad \text{Year 2005 effect} \\
\theta_{o,v} &\sim & \mathcal{MVN}(0, \Sigma_o) & \quad \text{Priors for explanatory variables} \\
\Sigma_o &\sim & \mathcal{W}(0, 1) & \quad \text{Prior for the variance-covariance matrix}
\end{aligned}
$$

  Where:

- o: Outcomes
- t: Time
- c: Country
- $y_{t,c,o}$: Binary variable that captures whether in a specific outcome (o), country (c) and year (t) there has been an adoption of a policy (1) or not (0).
- α_o: Baseline hazard.
- ρ_o: Shock in 2005 produced by simultaneous EU adoption.
- $\theta_{o,v}$: Effects of covariates (v), in each outcome o.
- Σ_o: Variance-covariance matrix to minimize multicollinearity.

Deal with interdependence data either in matrix or tidy formats

```
# No need to delete countries once the event has happened
# as they are already NA, and are needed for the full dataset
#da <- d %>%
#  filter(!is.na(Event))
da <- d
if (test) {
  da <- da %>%
    filter(Country %in% sample(unique(da$Country), size = length(unique(da$Country)) * 0.5))
}

###############################################################################
###### Interdependence using tidy approach
contiguity <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(contiguity.dependency = sum(wAdopted, na.rm = TRUE))

distance <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.distance),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.distance) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(distance.dependency = sum(wAdopted, na.rm = TRUE))

trade.dependency <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(trade.dependency = sum(wAdopted, na.rm = TRUE)) %>%
```

```r
  ungroup()

# For competition dependency we need both the imports and the exports

trade.partner.dependency <-
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
#  select(Origin, Destination, Year, Outcome, wAdopted) %>%
  mutate(wAdopted = ifelse(is.na(wAdopted), 0, wAdopted)) %>%
  ungroup()

trade.partner.others.imports <-
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
              select(Origin, Destination, Year, p.Imports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted.imports = Adopted * p.Imports) %>%
  filter(Origin != Destination) %>%
  # Rename countries to better control the matrices
  # Partner refers to the trade partner of the first origin country
  # ThirdCountry refers to the competitor of the first origin country
  #   through the Destination==Partner
  rename(Partner = Origin, ThirdCountry = Destination) %>%
  select(Partner, ThirdCountry, Year, Outcome, wAdopted.imports) %>%
  mutate(wAdopted.imports = ifelse(is.na(wAdopted.imports), 0, wAdopted.imports)) %>%
  group_by(Partner, Year, Outcome) %>%
  summarize(w.ThirdCountry.imp = sum(wAdopted.imports)) %>%
  ungroup()

trade.competition <-
  left_join(trade.partner.dependency,
              trade.partner.others.imports,
              by = c("Destination" = "Partner",
                     "Year" = "Year",
                     "Outcome" = "Outcome")) %>%
  mutate(wAdopted.dual = wAdopted * w.ThirdCountry.imp) %>%
  filter(Origin != Destination) %>%
  group_by(Origin, Year, Outcome) %>%
  summarize(trade.competition = sum(wAdopted.dual, na.rm = TRUE)) %>%
  ungroup() %>%
  rename(Country = Origin)


##### End interdependence using tidy approach
###############################################################################
```

```
###############################################################################
####### Interdependence using oWeighting matrices
## First use only relevant countries and then row-normalize
#M.distances <- M.distances[match(country.label, dimnames(M.distances)[[1]]),
#                           match(country.label, dimnames(M.distances)[[2]])]
#RW.M.distances <- 100 * (1 / M.distances) /
#  apply(1 / M.distances, 1, sum, na.rm = TRUE)
#RW.M.distances[is.na(RW.M.distances)] <- 0
#stopifnot(dimnames(RW.M.distances)[[1]] == country.label)
#stopifnot(dimnames(RW.M.distances)[[2]] == country.label)
#
#M.borders <- M.borders[match(country.label, dimnames(M.borders)[[1]]),
#                           match(country.label, dimnames(M.borders)[[2]])]
#RW.M.borders <- M.borders / apply(M.borders, 1, sum, na.rm = TRUE)
#RW.M.borders[is.na(RW.M.borders)] <- 0
#stopifnot(dimnames(RW.M.borders)[[1]] == country.label)
#stopifnot(dimnames(RW.M.borders)[[2]] == country.label)
#
## Trade exports (dependency)
#M.trade <- M.trade[match(country.label, dimnames(M.trade)[[1]]),
#                           match(country.label, dimnames(M.trade)[[2]]),]
#RW.M.trade <- array(0, dim = dim(M.trade), dimnames = dimnames(M.trade))
#for (t in 1:nT) {
#  RW.M.trade[,,t] <- M.trade[,,t] / apply(M.trade[,,t], 1, sum, na.rm = TRUE)
#}
#
#RW.M.trade[is.na(RW.M.trade)] <- 0
#stopifnot(dimnames(RW.M.trade)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade)[[2]] == country.label)
#
## Trade imports
#M.trade.imports <- M.trade.imports[match(country.label, dimnames(M.trade.imports)[[1]]),
#                           match(country.label, dimnames(M.trade.imports)[[2]]),]
#RW.M.trade.imports <- array(0, dim = dim(M.trade.imports), dimnames = dimnames(M.trade.imports))
#for (t in 1:nT) {
#  RW.M.trade.imports[,,t] <- M.trade.imports[,,t] / apply(M.trade.imports[,,t], 1, sum, na.rm = TRUE)
#}
#
#RW.M.trade.imports[is.na(RW.M.trade.imports)] <- 0
#stopifnot(dimnames(RW.M.trade.imports)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade.imports)[[2]] == country.label)
#
#
###### End interdependence using matrices
###############################################################################

Y <- reshape2::acast(da, Year ~ Country ~ Outcome, value.var = "Event")
Y.adopted <- reshape2::acast(da, Year ~ Country ~ Outcome, value.var = "Adopted")

# Work with fewer data
```

```
#if (test) {
#  Y <- Y[-c(1:10),,c(1,2)]
#  Y.adopted <- Y.adopted[-c(1:10),,c(1,2)]
#}

# Time
nT <- dim(Y)[1]
time.label <- time.span
#if (test) time.label <- time.label[-c(1:10)]
stopifnot(nT == length(time.label))
year.2005 <- ifelse(time.label == 2005, 1, 0)

# Outcomes
nO <- dim(Y)[3]
outcome.label <- dimnames(Y)[[3]]
outcome.has.eu <- ifelse(outcome.label %in% c("Carbon pricing", "ETS"), 1, 0)
outcome.is.tax <- ifelse(outcome.label %in% c("Carbon tax"), 1, 0)
outcome.is.ets <- ifelse(outcome.label %in% c("ETS", "ETS (no EU)"), 1, 0)


# Countries and covariates
X <- da %>%
  ### Non binary variables
  select(Country, Year,
         `GDPpc (log)`,
         `State expenditure`,
         `Population (log)`, # OK
         `Fossil fuel rents (log)`,
         `Democracy (Electoral)`, # not very well
#         `Political constraints`, # quite bad
#         `Government effectiveness`, # reasonable
#         #`Debt (log)`,
         `Vulnerability`, # not very well
         `CO2pc (log)`) %>% # not very well
  unique() %>%
  gather(Variable, value, -Country, -Year) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  ### Add binary variables
  bind_rows(select(da, Country, Year,
                   `Kyoto Ratification`,  # problematic
                   #`Paris Ratification`,
                   `Financial crisis`, # problematic
                   `Subnational tax`, # works well
                   `EU`) %>%  # works well
            unique() %>%
            gather(Variable, value, -Country, -Year)) %>%
  ###
  ### Add EU * GDPpc interaction
#  spread(Variable, value) %>%
```

```
#  mutate(`EU * GDPpc (log)` = EU * `GDPpc (log)`) %>%
#  gather(Variable, value, -c(Country, Year)) %>%
  ###
  ### Add Intercept
  spread(Variable, value) %>%
  mutate(`(Intercept)` = 1) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  ###
  reshape2::acast(Year ~ Country ~ Variable, value.var = "value")
country.label <- dimnames(X)[[2]]
nC <- length(country.label)
stopifnot(dimnames(Y)[[2]] == dimnames(X)[[2]])
#if (test) X <- X[-c(1:10),,]
##X <- cbind(1, X)
##dimnames(X)[[2]][1] <- "(Intercept)"

X.interdependence <-
  select(da, Country, Year, Outcome) %>%
  left_join(contiguity) %>%
  left_join(distance) %>%
  left_join(trade.dependency) %>%
  left_join(trade.competition) %>%
  mutate(Country = as.factor(Country),
         Year = as.integer(Year),
         Outcome = as.factor(Outcome)) %>%
  gather(Variable, value, -c(Country, Year, Outcome)) %>%
  group_by(Variable, Outcome) %>%
  mutate(value = std.zero(value)) %>%
  ungroup() %>%
#  filter(Variable != "distance.dependency") %>%
  mutate(Variable = factor(Variable, levels = c("contiguity.dependency",
                                                "distance.dependency",
                                                "trade.dependency",
                                                "trade.competition"))) %>%
#  spread(Variable, value) %>%
#  left_join(select(da, Country, Year, Outcome, `EU`)) %>%
#  mutate(eu.trade.dependency = EU * trade.dependency) %>%
#  select(-EU) %>%
#  gather(Variable, value, -c(Country, Year,  Outcome)) %>%
  filter(Variable %in% c("trade.competition")) %>%
#  filter(Variable %in% c("contiguity.dependency", "trade.competition")) %>%
  reshape2::acast(Outcome ~ Year ~ Country ~ Variable, value.var = "value")
stopifnot(dimnames(Y)[[2]] == dimnames(X.interdependence)[[3]])


# Mix covariates in the X object with those of the spatial matrices
covariate.label <- c(dimnames(X)[[3]],
#                     "Interdependence (Borders)",
#                     "EU * Interdependence (Trade dependency)",
                    "Interdependence (Trade competition)",
#                     "Interdependence (Trade dependency)",
```

```
                    "Tax already adopted", "ETS already adopted")
nCov <- length(covariate.label)

covariate.label.order <- c(
  "GDPpc (log)",                              # Economic
#  "EU * GDPpc (log)",
  "State expenditure",
  "Financial crisis", #
  "CO2pc (log)",                              # Contribution to CC
  "Fossil fuel rents (log)",
  "Population (log)", "Vulnerability",
  "Democracy (Electoral)",                    # Institutional
#  "Interdependence (Borders)",                # Interdependence
#  "Interdependence (Trade dependency)",
  "Interdependence (Trade competition)",
#  "EU * Interdependence (Trade dependency)",
  "EU",
  "Kyoto Ratification",
  "Subnational tax",
  "Tax already adopted", "ETS already adopted")


b0 <- rep(0, nCov)
b0[1] <- -7           # prior for rare events in the intercept
B0 <- diag(nCov)
#diag(B0) <- 2.5^-2
diag(B0) <- 1^-2
diag(B0) <- 0.5^-2
diag(B0) <- 10
#Omega <- diag(nCov)
#diag(Omega) <- 0.2^-2
#d0 <- rep(0, 2)
#D0 <- diag(2)
#diag(D0) <- 1^-2

# Restrictions on already adopted
B0.1 <- B0.2 <- B0.3 <- B0.4 <- B0.5 <- B0
diag(B0.1)[(nCov - 1):nCov] <- 0.001
diag(B0.2)[(nCov - 1):nCov] <- 0.001
diag(B0.3)[(nCov - 1)] <- 0.001
diag(B0.4)[(nCov - 0)] <- 0.001
diag(B0.5)[(nCov - 0)] <- 0.001

# Restrictions on the effect for 2005
rho.restrictions <- ifelse(outcome.has.eu != 1, 0, NA)

# See Pavlou et al, pg 1163-1164
# Follows soft shrinkage by Rockove et al
c <- 10 # degree of separation between the spike and the slab
delta <- 0.1 # threshold of practical significance
c <- 10 # degree of separation between the spike and the slab
```

```
delta <- 0.2 # threshold of practical significance
epsilon <- sqrt(2 * log(c) * c^2 / (c^2 - 1))
varspike <- (delta/epsilon)^2
varslab <- varspike * c^2


# Prepare the predicted probabilities vector,
# to avoid passing them all. Only relevant ones:
# Countries at risk in 2019
L.pi <- plab("pi", list(Year = time.label,
                        Country = country.label,
                        Outcome = outcome.label)) %>%
  mutate(Year = as.numeric(as.character(Year)))
L.pi.relevant <- L.pi %>%
  left_join(select(da, Country, Year, Outcome, Event)) %>%
  filter(Year == max(time.span)) %>%
  filter(Event == 0)
relevant.pp <- as.character(L.pi.relevant$Parameter)


D <- list(
  nO = nO, nT = nT, nC = nC,
  year_2005 = year.2005, outcome_has_eu = outcome.has.eu,
  outcome_is_tax = outcome.is.tax,
  outcome_is_ets = outcome.is.ets,
  X = unname(X), nCov = nCov,
  X_interdependence = unname(X.interdependence),
#  RW_M_borders = unname(RW.M.borders),
#  RW_M_distances = unname(RW.M.distances),
#  RW_M_trade = unname(RW.M.trade),
#  RW_M_trade_imports = unname(RW.M.trade.imports),
  b0 = b0,
  B0.1 = B0.1,
  B0.2 = B0.2,
  B0.3 = B0.3,
  B0.4 = B0.4,
  B0.5 = B0.5,
  df = nCov + 1,
  rho = rho.restrictions,
#  d0 = d0, D0 = D0,
#  Omega = Omega, df = nCov + 1,
#  varspike = varspike, varslab = varslab,
  Y_adopted = unname(Y.adopted),
  Y = unname(Y))

write.table(da, file = "exported_treated_data.csv", sep = ";", row.names = FALSE)
```

List the countries that without a policy in place, would make the most influence on other countries because of trade interdependency if they would adopt it.

```
# This is part of how to calculate trade dependency
```

```
tb ←
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  arrange(desc(p.Exports)) %>%
  filter(Country %in% country.coverage) %>%
  filter(Adopted == 0) %>% # only consider non-adopters
  filter(Year == max(time.span)) %>%
  group_by(Outcome, Destination) %>%
  summarize(av.p.exports = mean(p.Exports, na.rm = TRUE)) %>%
  ungroup() %>%
  group_by(Outcome) %>%
  arrange(Outcome, desc(av.p.exports)) %>%
  slice(1:5)


tc ← "Potential aggregated influence of each country if it would change from no adoption to adoption, as m

if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}

# Report events-per-variable (Pavlou et al)
# Problem when EPR < 10
da %>%
  group_by(Outcome) %>%
  summarize(SumEvents = length(which(Event == 1))) %>%
  mutate(`Events per variable (EPV)` = SumEvents / nCov) %>%
  arrange(desc(`Events per variable (EPV)`))

→ # A tibble: 5 x 3
→   Outcome                 SumEvents ‘Events per variable (EPV)‘
→   <fct>                       <int>                       <dbl>
→ 1 Carbon pricing                 51                       3.4
→ 2 ETS                            39                       2.6
→ 3 Carbon pricing (no EU)         36                       2.4
→ 4 Carbon tax                     31                       2.07
→ 5 ETS (no EU)                     8                       0.533

m ← 'model {
  for (o in 1:nO) {
```

```
    for (c in 1:nC) {
      for (t in 1:nT) {
        Y[t,c,o] ~ dbern(pi[t,c,o])
        #logit(pi[t,c,o]) <- alpha[t,o] + inprod(X[t,c,], theta[o,])
        logit(pi[t,c,o]) <- #alpha[o]
                                        inprod(X[t,c,1:(nCov-3)], theta[o,1:(nCov-3)])
                                      + (rho[o] * year_2005[t] * outcome_has_eu[o])
#                                      + theta[o,nCov-3] * X_interdependence[o,t,c,1]
                                      + theta[o,nCov-2] * X_interdependence[o,t,c,1]
                                      + theta[o,nCov-1] * (outcome_is_ets[o] * Y_adopted[t,c,3])
                                      + theta[o,nCov-0]   * (outcome_is_tax[o] * Y_adopted[t,c,4])
      }
    }
    #
    # Kalman filter for time trends
    #
    #alpha[o] ~ dt(-7, 1^-2, 3)
    rho[o] ~ dnorm(2, 3^-2)
#    for (t in 2:nT) {
##      alpha[t,o] ~ dnorm(alpha[t-1,o] +
##                        (rho[o] * year_2005[t] * outcome_has_eu[o]) -
##                        (rho[o] * year_2005[t-1] * outcome_has_eu[o])
##                      , tau.alpha[o])
#      alpha[t,o] ~ dnorm(alpha[t-1,o] +
#                        (rho[o] * year_2005[t] * outcome_has_eu[o]) -
#                        (rho[o] * year_2005[t-1] * outcome_has_eu[o])
#                      , 0.3^-3)
#    }
#    rho[o] ~ dnorm(2, 3^-2)
#    alpha[1,o] ~ dnorm(-8, 3^-2)
###    sigma.alpha[o] ~ dt(0, 0.01, 1)T(0,)
##    tau.alpha[o] ~ dgamma(7, 0.3)
##    sigma.alpha[o] <- 1 / sqrt(tau.alpha[o])
#    #
    #
    ##### Main effects
    #
  }
  theta[1,1:nCov] ~ dmnorm(b0[1:nCov], Omega.1[1:nCov,1:nCov])
  theta[2,1:nCov] ~ dmnorm(b0[1:nCov], Omega.2[1:nCov,1:nCov])
  theta[3,1:nCov] ~ dmnorm(b0[1:nCov], Omega.3[1:nCov,1:nCov])
  theta[4,1:nCov] ~ dmnorm(b0[1:nCov], Omega.4[1:nCov,1:nCov])
  theta[5,1:nCov] ~ dmnorm(b0[1:nCov], Omega.5[1:nCov,1:nCov])
  Omega.1[1:nCov,1:nCov] ~ dwish(B0.1, df)
  Omega.2[1:nCov,1:nCov] ~ dwish(B0.2, df)
  Omega.3[1:nCov,1:nCov] ~ dwish(B0.3, df)
  Omega.4[1:nCov,1:nCov] ~ dwish(B0.4, df)
  Omega.5[1:nCov,1:nCov] ~ dwish(B0.5, df)
  Sigma[1,1:nCov,1:nCov] <- inverse(Omega.1[1:nCov,1:nCov])
  Sigma[2,1:nCov,1:nCov] <- inverse(Omega.2[1:nCov,1:nCov])
  Sigma[3,1:nCov,1:nCov] <- inverse(Omega.3[1:nCov,1:nCov])
```

```
  Sigma[4,1:nCov,1:nCov] <- inverse(Omega.4[1:nCov,1:nCov])
  Sigma[5,1:nCov,1:nCov] <- inverse(Omega.5[1:nCov,1:nCov])
  #
  # Missing data
  #
  for (cov in 1:(nCov-3)) { # obviate the interdependence variables
    for (c in 1:nC) {
      # No time trend for missingness
      for (t in 1:nT) {
        X[t,c,cov] ~ dnorm(0, 1^-2)
      }
    }
  }
  for (cov in 1:1) { # the number of interdependence variables included
    for (c in 1:nC) {
      for (t in 1:nT) {
        for (o in 1:nO) {
          X_interdependence[o,t,c,cov] ~ dnorm(0, 0.5^-2)
        }
      }
    }
  }
}'
write(m, file = paste0("models/model-", M.lab, ".bug"))

par <- NULL
#par <- c(par, "alpha")
#par <- c(par, "sigma.alpha")
par <- c(par, "rho")
par <- c(par, "theta")
#par <- c(par, "sigma.theta")
#par <- c(par, "Theta")
#par <- c(par, "tau.Theta")
#par <- c(par, "mu.Theta")
#par <- c(par, "delta", "Delta")
#par <- c(par, "sigma.delta")
par <- c(par, "Sigma")
if (run.pcp) {
  par <- c(par, "pi")
} else {
  par <- c(par, relevant.pp) # only selected cases
}
#par <- c(par, "prec_theta")

#inits.alpha <- array(-8, dim = c(nT, nO))
inits.alpha <- array(-8, dim = c(nO))
inits <- list(
  list(.RNG.seed=14717, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha),
  list(.RNG.seed=14718, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha - 2),
```

```
  list(.RNG.seed=14719, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha + 2))

t0 ← proc.time()
rj ← run.jags(model = paste0("models/model-", M.lab, ".bug"),
               data = dump.format(D, checkvalid = FALSE),
#               inits = inits,
               modules = c("glm", "lecuyer"),
               n.chains = chains, thin = thin,
               adapt = adapt, burnin = burnin, sample = sample,
               monitor = par, method = "parallel", summarise = FALSE)
s ← as.mcmc.list(rj)
save(s, file = paste0("sample-", M.lab, ".RData"))
proc.time() - t0

→ [1] 3

→ [1] 2500

→ [1] 1

→ [1] 2000

→ [1] 2056

#ggmcmc(ggs(s, family = "^theta|^alpha|^Sigma|^rho"),
ggmcmc(ggs(s, family = "^theta|^alpha|^rho"),
       file = paste0("ggmcmc-", "all", "-", M.lab, ".pdf"),
       plot = c("traceplot", "crosscorrelation", "caterpillar", "geweke"))

ggs(s, family = "^theta\\[1,|rho\\[1") %>%
  ggs_crosscorrelation()

ggmcmc(ggs(s, family = "^alpha\\["),
       file = paste0("ggmcmc-", "alpha", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "theta"),
       file = paste0("ggmcmc-", "theta", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "Sigma"),
       file = paste0("ggmcmc-", "Sigma", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "sigma.alpha"),
       file = paste0("ggmcmc-", "sigma_alpha", "-", M.lab, ".pdf"),
       plot = c("traceplot", "crosscorrelation", "caterpillar"))

#ggmcmc(ggs(s, family = "mu.Theta"),
#       file = paste0("ggmcmc-", "mu_Theta", "-", M.lab, ".pdf"),
#       plot = c("traceplot", "crosscorrelation", "caterpillar"))
```

```r
L.sigma.theta <- plab("sigma.theta", list(Covariate = covariate.label))
L.sigma.delta <- plab("sigma.delta", list(Covariate = c("Interdependence (Borders)",
                                                  "Interdependence (Trade)")))
L.sigma.theta <- bind_rows(L.sigma.theta, L.sigma.delta)
S.sigma.theta <- ggs(s, family = "^sigma.theta\\[|^sigma.delta\\[", par_labels = L.sigma.theta)
ggs_caterpillar(S.sigma.theta) +
  ggtitle("Between outcome standard deviations")
#  geom_vline(xintercept = 0, lty = 3) +

L.theta <- plab("theta", list(Outcome = outcome.label,
                              Covariate = covariate.label))
#L.delta <- plab("delta", list(Outcome = outcome.label,
#                              Covariate = c("Interdependence (Borders)",
#                                            "Interdependence (Trade)")))
#L.theta <- bind_rows(L.theta, L.delta)
S.theta <- ggs(s, family = "^theta\\[|^delta\\[", par_labels = L.theta) %>%
  mutate(Model = M) %>%
  filter(Covariate != "(Intercept)") %>%
  mutate(Covariate = factor(Covariate, rev(covariate.label.order)))

save(S.theta, file = paste("samples-theta-", M.lab, ".RData", sep = ""))

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Outcome)


S.theta %>%
  filter(Outcome %in% c("Carbon pricing", "Carbon pricing (no EU)")) %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Outcome)


S.theta %>%
  filter(Outcome %in% c("Carbon pricing")) %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)


ggs_caterpillar(S.theta, label = "Outcome", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Covariate)
```

Variance-covariance matrices

```r
L.Sigma.Omega <- plab("Sigma", list(
                              Outcome = outcome.label,
                              Covariate.1 = covariate.label,
                              Covariate.2 = covariate.label))
S.Sigma.Omega <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma.Omega)
```

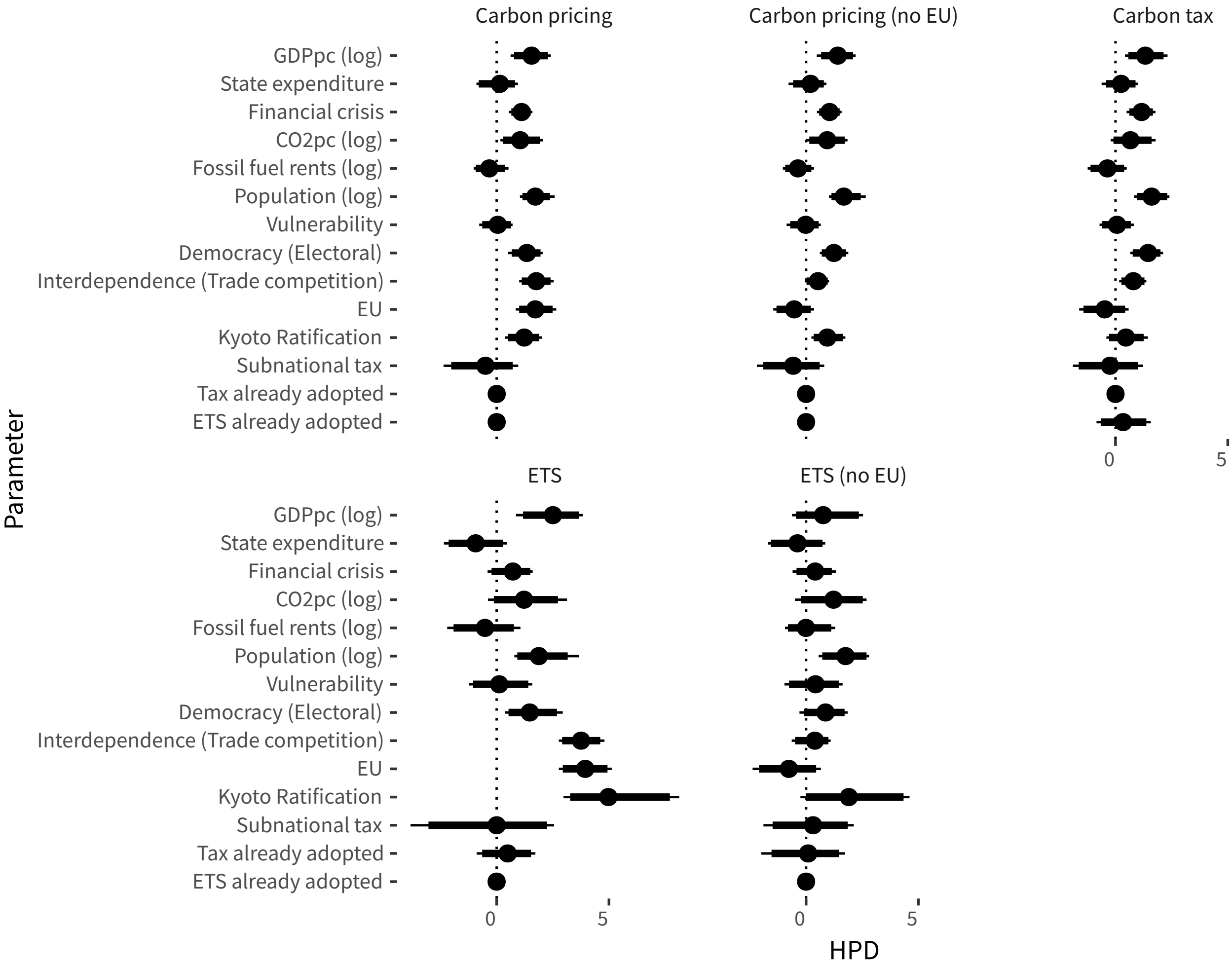


Figure 4.1: HPD of the effects of covariates on the likelihood of the event, by outcome.

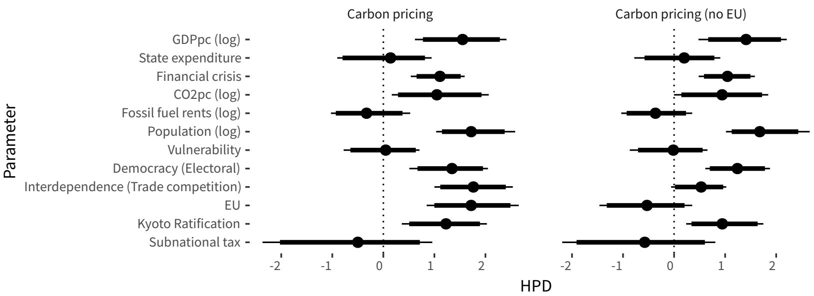


Figure 4.2: HPD of the effects of covariates on the likelihood of the event, by outcome. Only carbon pricing, with and without EU.

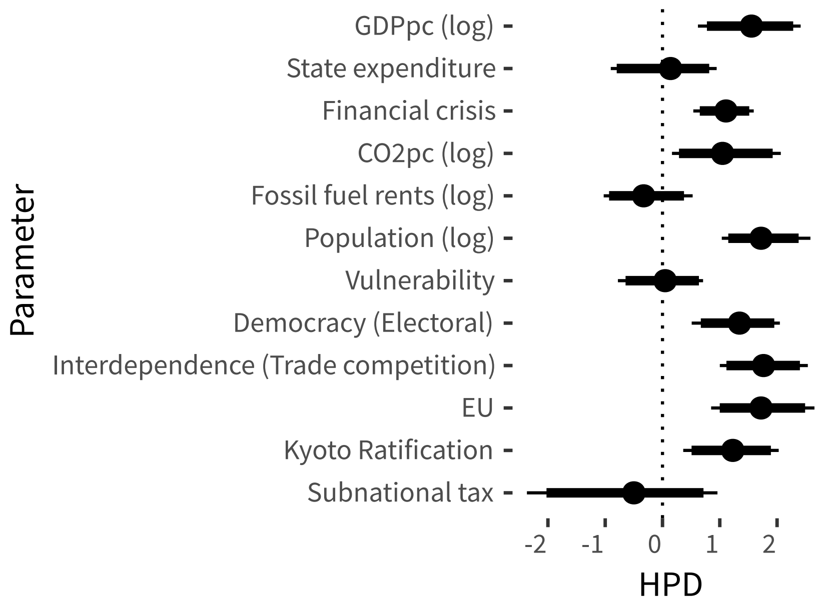


Figure 4.3: HPD of the effects of covariates on the likelihood of the event, by outcome. Only carbon pricing with the EU.

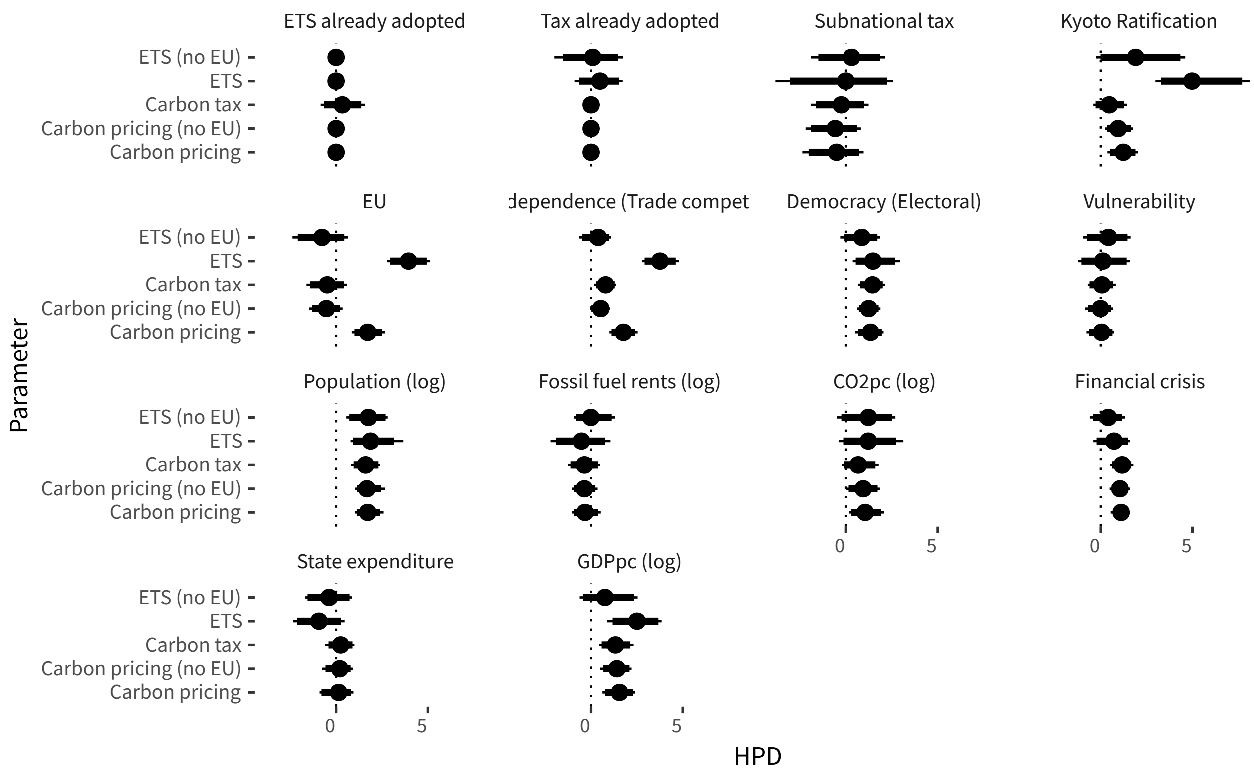


Figure 4.4: HPD of the effects of covariates on the likelihood of the event, by Covariate.

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

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

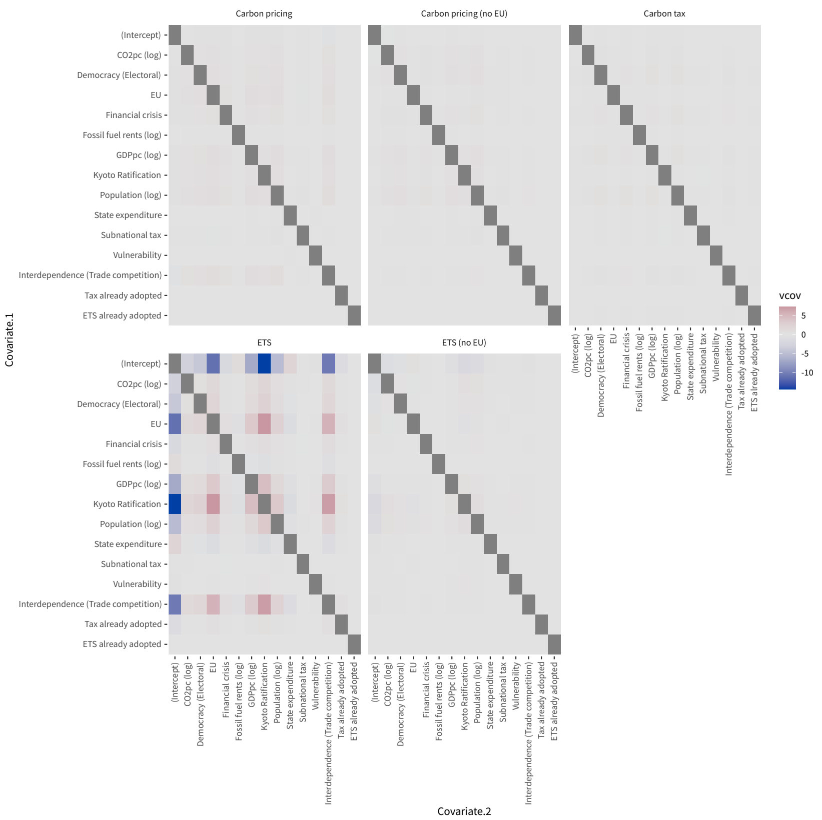


Figure 4.5: Variance-covariance matrix of main effects.

```r
or <- function(x, significant = 2) {
  or <- as.character(signif((x - 1) * 100, significant))
  or[or < 0] <- paste0("\U25Bd ", str_replace(or[or < 0], "-", ""), "%")
  or[or > 0] <- paste0("\U25B3 ", or[or > 0], "%")
  or[or == 0] <- "="
  return(or)
}

tb <- S.theta %>%
  filter(Covariate != "(Intercept)") %>%
  ci() %>%
  arrange(Outcome, desc(abs(median))) %>%
  mutate(`Odds Ratio` = exp(median)) %>%
  mutate(`Expected effect` = or(`Odds Ratio`)) %>%
```

```
  select(Outcome, Covariate, `Odds Ratio`, `Expected effect`)

tc ← "Odds ratios of expected effect sizes, and sorted by magnitude and outcome."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 4.1: Odds ratios of expected effect sizes, and sorted by magnitude and outcome.

Outcome	Covariate	Odds Ratio	Expected effect
Carbon pricing	Interdependence (Trade competition)	5.8329	△ 480%
Carbon pricing	Population (log)	5.5877	△ 460%
Carbon pricing	EU	5.5857	△ 460%
Carbon pricing	GDPpc (log)	4.7252	△ 370%
Carbon pricing	Democracy (Electoral)	3.8354	△ 280%
Carbon pricing	Kyoto Ratification	3.4119	△ 240%
Carbon pricing	Financial crisis	3.0307	△ 200%
Carbon pricing	CO2pc (log)	2.8523	△ 190%
Carbon pricing	Subnational tax	0.6068	▽ 39%
Carbon pricing	Fossil fuel rents (log)	0.7195	▽ 28%
Carbon pricing	State expenditure	1.1509	△ 15%
Carbon pricing	Vulnerability	1.0472	△ 4.7%
Carbon pricing	Tax already adopted	1.0006	△ 0.063%
Carbon pricing	ETS already adopted	0.9995	▽ 0.046%
Carbon pricing (no EU)	Population (log)	5.3572	△ 440%
Carbon pricing (no EU)	GDPpc (log)	4.0960	△ 310%
Carbon pricing (no EU)	Democracy (Electoral)	3.4564	△ 250%
Carbon pricing (no EU)	Financial crisis	2.8463	△ 180%
Carbon pricing (no EU)	Kyoto Ratification	2.5704	△ 160%
Carbon pricing (no EU)	CO2pc (log)	2.5616	△ 160%
Carbon pricing (no EU)	Subnational tax	0.5627	▽ 44%
Carbon pricing (no EU)	Interdependence (Trade competition)	1.7001	△ 70%
Carbon pricing (no EU)	EU	0.5884	▽ 41%
Carbon pricing (no EU)	Fossil fuel rents (log)	0.6941	▽ 31%
Carbon pricing (no EU)	State expenditure	1.2191	△ 22%
Carbon pricing (no EU)	Vulnerability	0.9851	▽ 1.5%
Carbon pricing (no EU)	Tax already adopted	1.0000	▽ 0.0037%
Carbon pricing (no EU)	ETS already adopted	1.0000	▽ 0.00065%
Carbon tax	Population (log)	4.9919	△ 400%
Carbon tax	Democracy (Electoral)	4.2694	△ 330%
Carbon tax	GDPpc (log)	3.7818	△ 280%
Carbon tax	Financial crisis	3.1985	△ 220%
Carbon tax	Interdependence (Trade competition)	2.2117	△ 120%
Carbon tax	CO2pc (log)	1.9451	△ 95%
Carbon tax	EU	0.6182	▽ 38%
Carbon tax	Kyoto Ratification	1.5884	△ 59%
Carbon tax	Fossil fuel rents (log)	0.6976	▽ 30%
Carbon tax	ETS already adopted	1.4030	△ 40%
Carbon tax	State expenditure	1.2824	△ 28%
Carbon tax	Subnational tax	0.7831	▽ 22%
Carbon tax	Vulnerability	1.0677	△ 6.8%

Carbon tax	Tax already adopted	1.0001	△ 0.012%
ETS	Kyoto Ratification	144.5723	△ 14000%
ETS	EU	51.5886	△ 5100%
ETS	Interdependence (Trade competition)	42.8457	△ 4200%
ETS	GDPpc (log)	12.2306	△ 1100%
ETS	Population (log)	6.5265	△ 550%
ETS	Democracy (Electoral)	4.3581	△ 340%
ETS	CO2pc (log)	3.3762	△ 240%
ETS	State expenditure	0.3917	▽ 61%
ETS	Financial crisis	2.0532	△ 110%
ETS	Fossil fuel rents (log)	0.5932	▽ 41%
ETS	Tax already adopted	1.6264	△ 63%
ETS	Vulnerability	1.1192	△ 12%
ETS	ETS already adopted	0.9964	▽ 0.36%
ETS	Subnational tax	0.9992	▽ 0.08%
ETS (no EU)	Kyoto Ratification	6.7039	△ 570%
ETS (no EU)	Population (log)	5.8002	△ 480%
ETS (no EU)	CO2pc (log)	3.3897	△ 240%
ETS (no EU)	Democracy (Electoral)	2.3779	△ 140%
ETS (no EU)	EU	0.4638	▽ 54%
ETS (no EU)	GDPpc (log)	2.1358	△ 110%
ETS (no EU)	Vulnerability	1.5196	△ 52%
ETS (no EU)	Financial crisis	1.4979	△ 50%
ETS (no EU)	Interdependence (Trade competition)	1.4781	△ 48%
ETS (no EU)	State expenditure	0.6806	▽ 32%
ETS (no EU)	Subnational tax	1.3625	△ 36%
ETS (no EU)	Tax already adopted	1.0968	△ 9.7%
ETS (no EU)	Fossil fuel rents (log)	0.9929	▽ 0.71%
ETS (no EU)	ETS already adopted	0.9994	▽ 0.06%

```
require(xtable)
tb.dump <- tb %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
  select(-Outcome)
tc <- "Odds ratios of expected effect sizes, by magnitude."

print(xtable(tb.dump,
             caption = tc,
             label = "tab:ees",
             tabular.environment = "longtable"),
      file = "table_expected_effect_sizes-001.tex",
      #size = "footnotesize",
      size = "normalsize",
      include.rownames = FALSE)

S.theta %>%
  filter(Outcome %in% c("Carbon tax", "ETS")) %>%
  ci() %>%
  ggplot(aes(ymin = low, ymax = high,
             y = median, x = Covariate,
             color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.3)) +
```

```
  geom_linerange(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.3)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Harmonic")
```

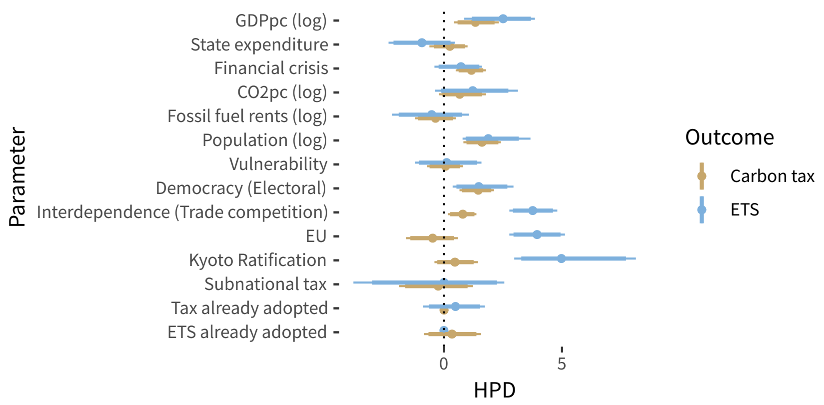


Figure 4.6: HPD of the effects of covariates on the likelihood of the event, by outcome. Only Tax and ETS.

```
L.alpha <- plab("alpha", list(Outcome = outcome.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_labels = L.alpha) %>%
  mutate(value = inv.logit(value))
ggs_caterpillar(S.alpha)

L.alpha <- plab("alpha", list(Year = time.label, Outcome = outcome.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_labels = L.alpha) %>%
  mutate(Year = as.numeric(as.character(Year))) %>%
  mutate(value = inv.logit(value))

ci(S.alpha, thick_ci = c(0.1, 0.9)) %>%
  ggplot(aes(x = Year, y = median, color = Outcome)) +
  geom_line() +
  ylab("Hazard")

ci(S.alpha, thick_ci = c(0.1, 0.9)) %>%
  ggplot(aes(x = Year, y = median, ymin = low, ymax = high,
             fill = Outcome, color = Outcome)) +
  geom_line() +
  geom_ribbon(alpha = 0.3) +
  facet_wrap(~ Outcome) +
  ylab("Hazard")
```

List the countries most likely to adopt each policy.

```
L.pi <- plab("pi", list(Year = time.label,
                        Country = country.label,
                        Outcome = outcome.label)) %>%
  mutate(Year = as.numeric(as.character(Year)))
S.pi <- ggs(s, family = "^pi\\[", par_labels = L.pi)

tpp <- S.pi %>%
```

```
  mutate(value = inv.logit(value)) %>%
  ci() %>%
  select(Country, Outcome, `Predicted probability` = median) %>%
  group_by(Outcome) %>%
  arrange(desc(`Predicted probability`)) %>%
  slice(1:5)
```

```
tc <- "Countries with higher posterior median predicted probabilities of adopting each policy
outcome. Top 5 by Outcome."
if (knitr::is_latex_output()) {
  kable(tpp, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tpp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 4.2: Countries with higher posterior median predicted probabilities of adopting each policy outcome. Top 5 by Outcome.

Country	Outcome	Predicted probability
Russia	Carbon pricing	0.5111
Turkey	Carbon pricing	0.5072
Bosnia & Herzegovina	Carbon pricing	0.5061
Brazil	Carbon pricing	0.5057
Qatar	Carbon pricing	0.5056
Germany	Carbon pricing (no EU)	0.5062
Qatar	Carbon pricing (no EU)	0.5054
Brazil	Carbon pricing (no EU)	0.5053
Saudi Arabia	Carbon pricing (no EU)	0.5052
United States	Carbon pricing (no EU)	0.5050
China	Carbon tax	0.5078
Germany	Carbon tax	0.5076
United States	Carbon tax	0.5059
Italy	Carbon tax	0.5053
Australia	Carbon tax	0.5051
Singapore	ETS	0.5514
South Africa	ETS	0.5433
Bosnia & Herzegovina	ETS	0.5112
Libya	ETS	0.5048
Russia	ETS	0.5047
South Africa	ETS (no EU)	0.5093
Singapore	ETS (no EU)	0.5092
Japan	ETS (no EU)	0.5042
India	ETS (no EU)	0.5019
Russia	ETS (no EU)	0.5012

```
S.pi

# Manually calculate PCP
# as ggmcmc's ggs_pcp() is not ready for matrices as input for outcome
```

```r
threshold <- da %>%
  group_by(Outcome) %>%
  summarize(Threshold = length(which(Event == 1)) / n())

S <- inner_join(S.pi, select(da, Country, Year, Outcome, Event)) %>%
  left_join(threshold) %>%
  mutate(Correct = if_else( (value < Threshold & Event == 0) |
                             (value > Threshold & Event == 1),
                           TRUE, FALSE)) %>%
  group_by(Outcome, Iteration, Chain) %>%
  summarize(PCP = length(which(Correct)) / n())


ggplot(S, aes(x = PCP)) +
  geom_histogram() +
  facet_grid(~ Outcome) +
  expand_limits(x = c(0, 1))

t.pcp <- S %>%
  group_by(Outcome) %>%
  summarize(`Median PCP` = mean(PCP))

tc <- "Posterior median percent correctly predicted, by Outcome."
if (knitr::is_latex_output()) {
  kable(tpp, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tpp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}

rm(S)
invisible(gc())
```

5

Model: Baseline 002, Environmental Kuznets Curve

```
# Model:
M
```

```
→ [1] "Baseline 002, Environmental Kuznets Curve"
```

- Event history analysis using logistic regression for discrete time (years).
- Time dependence is fixed over the period, with a dummy in 2005 and for EU models to capture the effect of EU in a single period. More complex time dependency does not show any significant gain and complicates the estimation.
- Bayesian inference with weakly informative priors.

  Data preparation:

- Standardization to 0.5 standard deviations for all covariates, following Gelman (2008).

  Model equation:

$$
\begin{aligned}
Y_{t,c,o} \sim & \quad \mathcal{B}(\pi_{t,c,o}) && \text{Main data component} \\
\pi_{t,c,o} = & \quad \mathrm{logit}\alpha_o + \rho_o + (\theta_{o,v} X_{t,c,v}) && \text{Main linear model} \\
\alpha_o \sim & \quad \mathcal{N}(-7, 2) && \text{Rare event prior} \\
\rho_o \sim & \quad \mathcal{N}(0, 2) && \text{Year 2005 effect} \\
\theta_{o,v} \sim & \quad \mathcal{MVN}(0, \Sigma_o) && \text{Priors for explanatory variables} \\
\Sigma_o \sim & \quad \mathcal{W}(0, 1) && \text{Prior for the variance-covariance matrix}
\end{aligned}
$$

Where:

- o: Outcomes
- t: Time
- c: Country
- $y_{t,c,o}$: Binary variable that captures whether in a specific outcome (o), country (c) and year (t) there has been an adoption of a policy (1) or not (0).
- α_o: Baseline hazard.
- ρ_o: Shock in 2005 produced by simultaneous EU adoption.
- $\theta_{o,v}$: Effects of covariates (v), in each outcome o.
- Σ_o: Variance-covariance matrix to minimize multicollinearity.

Deal with interdependence data either in matrix or tidy formats

```
# No need to delete countries once the event has happened
# as they are already NA, and are needed for the full dataset
#da <- d %>%
#  filter(!is.na(Event))
da <- d
if (test) {
  da <- da %>%
    filter(Country %in% sample(unique(da$Country), size = length(unique(da$Country)) * 0.5))
}

##############################################################################
###### Interdependence using tidy approach
contiguity <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(contiguity.dependency = sum(wAdopted, na.rm = TRUE))

distance <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.distance),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.distance) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(distance.dependency = sum(wAdopted, na.rm = TRUE))

trade.dependency <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(trade.dependency = sum(wAdopted, na.rm = TRUE)) %>%
```

```
  ungroup()

# For competition dependency we need both the imports and the exports

trade.partner.dependency <-
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
#  select(Origin, Destination, Year, Outcome, wAdopted) %>%
  mutate(wAdopted = ifelse(is.na(wAdopted), 0, wAdopted)) %>%
  ungroup()

trade.partner.others.imports <-
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Imports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted.imports = Adopted * p.Imports) %>%
  filter(Origin != Destination) %>%
  # Rename countries to better control the matrices
  # Partner refers to the trade partner of the first origin country
  # ThirdCountry refers to the competitor of the first origin country
  #   through the Destination==Partner
  rename(Partner = Origin, ThirdCountry = Destination) %>%
  select(Partner, ThirdCountry, Year, Outcome, wAdopted.imports) %>%
  mutate(wAdopted.imports = ifelse(is.na(wAdopted.imports), 0, wAdopted.imports)) %>%
  group_by(Partner, Year, Outcome) %>%
  summarize(w.ThirdCountry.imp = sum(wAdopted.imports)) %>%
  ungroup()

trade.competition <-
  left_join(trade.partner.dependency,
            trade.partner.others.imports,
            by = c("Destination" = "Partner",
                   "Year" = "Year",
                   "Outcome" = "Outcome")) %>%
  mutate(wAdopted.dual = wAdopted * w.ThirdCountry.imp) %>%
  filter(Origin != Destination) %>%
  group_by(Origin, Year, Outcome) %>%
  summarize(trade.competition = sum(wAdopted.dual, na.rm = TRUE)) %>%
  ungroup() %>%
  rename(Country = Origin)


##### End interdependence using tidy approach
################################################################################
```

```
################################################################################
####### Interdependence using oWeighting matrices
## First use only relevant countries and then row-normalize
#M.distances <- M.distances[match(country.label, dimnames(M.distances)[[1]]),
#                           match(country.label, dimnames(M.distances)[[2]])]
#RW.M.distances <- 100 * (1 / M.distances) /
#  apply(1 / M.distances, 1, sum, na.rm = TRUE)
#RW.M.distances[is.na(RW.M.distances)] <- 0
#stopifnot(dimnames(RW.M.distances)[[1]] == country.label)
#stopifnot(dimnames(RW.M.distances)[[2]] == country.label)
#
#M.borders <- M.borders[match(country.label, dimnames(M.borders)[[1]]),
#                           match(country.label, dimnames(M.borders)[[2]])]
#RW.M.borders <- M.borders / apply(M.borders, 1, sum, na.rm = TRUE)
#RW.M.borders[is.na(RW.M.borders)] <- 0
#stopifnot(dimnames(RW.M.borders)[[1]] == country.label)
#stopifnot(dimnames(RW.M.borders)[[2]] == country.label)
#
## Trade exports (dependency)
#M.trade <- M.trade[match(country.label, dimnames(M.trade)[[1]]),
#                           match(country.label, dimnames(M.trade)[[2]]),]
#RW.M.trade <- array(0, dim = dim(M.trade), dimnames = dimnames(M.trade))
#for (t in 1:nT) {
#  RW.M.trade[,,t] <- M.trade[,,t] / apply(M.trade[,,t], 1, sum, na.rm = TRUE)
#}
#
#RW.M.trade[is.na(RW.M.trade)] <- 0
#stopifnot(dimnames(RW.M.trade)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade)[[2]] == country.label)
#
## Trade imports
#M.trade.imports <- M.trade.imports[match(country.label, dimnames(M.trade.imports)[[1]]),
#                           match(country.label, dimnames(M.trade.imports)[[2]]),]
#RW.M.trade.imports <- array(0, dim = dim(M.trade.imports), dimnames = dimnames(M.trade.imports))
#for (t in 1:nT) {
#  RW.M.trade.imports[,,t] <- M.trade.imports[,,t] / apply(M.trade.imports[,,t], 1, sum, na.rm = TRUE)
#}
#
#RW.M.trade.imports[is.na(RW.M.trade.imports)] <- 0
#stopifnot(dimnames(RW.M.trade.imports)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade.imports)[[2]] == country.label)
#
#
###### End interdependence using matrices
################################################################################

Y <- reshape2::acast(da, Year ~ Country ~ Outcome, value.var = "Event")
Y.adopted <- reshape2::acast(da, Year ~ Country ~ Outcome, value.var = "Adopted")

# Work with fewer data
```

```r
#if (test) {
#  Y <- Y[-c(1:10),,c(1,2)]
#  Y.adopted <- Y.adopted[-c(1:10),,c(1,2)]
#}

# Time
nT <- dim(Y)[1]
time.label <- time.span
#if (test) time.label <- time.label[-c(1:10)]
stopifnot(nT == length(time.label))
year.2005 <- ifelse(time.label == 2005, 1, 0)

# Outcomes
nO <- dim(Y)[3]
outcome.label <- dimnames(Y)[[3]]
outcome.has.eu <- ifelse(outcome.label %in% c("Carbon pricing", "ETS"), 1, 0)
outcome.is.tax <- ifelse(outcome.label %in% c("Carbon tax"), 1, 0)
outcome.is.ets <- ifelse(outcome.label %in% c("ETS", "ETS (no EU)"), 1, 0)


# Countries and covariates
X <- da %>%
  ### Non binary variables
  select(Country, Year,
         `GDPpc (log)`,
         `State expenditure`,
         `Population (log)`, # OK
         `Fossil fuel rents (log)`,
         `Democracy (Electoral)`, # not very well
#         `Political constraints`, # quite bad
#         `Government effectiveness`, # reasonable
#         #`Debt (log)`,
         `Vulnerability`, # not very well
         `CO2pc (log)`) %>% # not very well
  unique() %>%
  gather(Variable, value, -Country, -Year) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  # Deal with Kuznets curve and CO2 emmissions
  spread(Variable, value) %>%
  mutate(CO2.1 = ifelse(`CO2pc (log)` < quantile(`CO2pc (log)`, 0.25, na.rm = TRUE), 1, 0)) %>%
  mutate(CO2.1 = ifelse(is.na(CO2.1), 0, CO2.1)) %>%
  mutate(CO2.2 = ifelse(`CO2pc (log)` > quantile(`CO2pc (log)`, 0.25, na.rm = TRUE) &
                        `CO2pc (log)` < quantile(`CO2pc (log)`, 0.50, na.rm = TRUE), 1, 0)) %>%
  mutate(CO2.2 = ifelse(is.na(CO2.2), 0, CO2.2)) %>%
  mutate(CO2.3 = ifelse(`CO2pc (log)` > quantile(`CO2pc (log)`, 0.50, na.rm = TRUE) &
                        `CO2pc (log)` < quantile(`CO2pc (log)`, 0.75, na.rm = TRUE), 1, 0)) %>%
  mutate(CO2.3 = ifelse(is.na(CO2.3), 0, CO2.3)) %>%
  mutate(CO2.4 = ifelse(`CO2pc (log)` > quantile(`CO2pc (log)`, 0.75, na.rm = TRUE), 1, 0)) %>%
  mutate(CO2.4 = ifelse(is.na(CO2.4), 0, CO2.4)) %>%
```

```
  gather(Variable, value, -c(Country, Year)) %>%
  filter(Variable != "CO2pc (log)") %>%
  #
  ### Add binary variables
  bind_rows(select(da, Country, Year,
                   `Kyoto Ratification`,  # problematic
                   #`Paris Ratification`,
                   `Financial crisis`, # problematic
                   `Subnational tax`, # works well
                   `EU`) %>%  # works well
            unique() %>%
            gather(Variable, value, -Country, -Year)) %>%
  ###
  ### Add EU * GDPpc interaction
#  spread(Variable, value) %>%
#  mutate(`EU * GDPpc (log)` = EU * `GDPpc (log)`) %>%
#  gather(Variable, value, -c(Country, Year)) %>%
  ###
  ### Add Intercept
  spread(Variable, value) %>%
  mutate(`(Intercept)` = 1) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  ###
  reshape2::acast(Year ~ Country ~ Variable, value.var = "value")
country.label <- dimnames(X)[[2]]
nC <- length(country.label)
stopifnot(dimnames(Y)[[2]] == dimnames(X)[[2]])
#if (test) X <- X[-c(1:10),,]
##X <- cbind(1, X)
##dimnames(X)[[2]][1] <- "(Intercept)"

X.interdependence <-
  select(da, Country, Year, Outcome) %>%
  left_join(contiguity) %>%
  left_join(distance) %>%
  left_join(trade.dependency) %>%
  left_join(trade.competition) %>%
  mutate(Country = as.factor(Country),
         Year = as.integer(Year),
         Outcome = as.factor(Outcome)) %>%
  gather(Variable, value, -c(Country, Year, Outcome)) %>%
  group_by(Variable, Outcome) %>%
  mutate(value = std.zero(value)) %>%
  ungroup() %>%
#  filter(Variable != "distance.dependency") %>%
  mutate(Variable = factor(Variable, levels = c("contiguity.dependency",
                                                "distance.dependency",
                                                "trade.dependency",
                                                "trade.competition"))) %>%
#  spread(Variable, value) %>%
#  left_join(select(da, Country, Year, Outcome, `EU`)) %>%
```

```
#  mutate(eu.trade.dependency = EU * trade.dependency) %>%
#  select(-EU) %>%
#  gather(Variable, value, -c(Country, Year,  Outcome)) %>%
  filter(Variable %in% c("trade.competition")) %>%
#  filter(Variable %in% c("contiguity.dependency", "trade.competition")) %>%
  reshape2::acast(Outcome ~ Year ~ Country ~ Variable, value.var = "value")
stopifnot(dimnames(Y)[[2]] == dimnames(X.interdependence)[[3]])


# Mix covariates in the X object with those of the spatial matrices
covariate.label <- c(dimnames(X)[[3]],
#                     "Interdependence (Borders)",
#                     "EU * Interdependence (Trade dependency)",
                     "Interdependence (Trade competition)",
#                     "Interdependence (Trade dependency)",
                     "Tax already adopted", "ETS already adopted")
nCov <- length(covariate.label)

covariate.label.order <- c(
  "GDPpc (log)",                                  # Economic
#  "EU * GDPpc (log)",
  "State expenditure",
  "Financial crisis", #
                                                  # Contribution to CC
  "CO2.1",
  "CO2.2",
  "CO2.3",
  "CO2.4",
  "Fossil fuel rents (log)",
  "Population (log)", "Vulnerability",
  "Democracy (Electoral)",                        # Institutional
#  "Interdependence (Borders)",                    # Interdependence
#  "Interdependence (Trade dependency)",
  "Interdependence (Trade competition)",
#  "EU * Interdependence (Trade dependency)",
  "EU",
  "Kyoto Ratification",
  "Subnational tax",
  "Tax already adopted", "ETS already adopted")


b0 <- rep(0, nCov)
b0[1] <- -7           # prior for rare events in the intercept
B0 <- diag(nCov)
#diag(B0) <- 2.5^-2
diag(B0) <- 1^-2
diag(B0) <- 0.5^-2
diag(B0) <- 10
#Omega <- diag(nCov)
#diag(Omega) <- 0.2^-2
#d0 <- rep(0, 2)
```

```
#D0 <- diag(2)
#diag(D0) <- 1^-2

# Restrictions on already adopted
B0.1 <- B0.2 <- B0.3 <- B0.4 <- B0.5 <- B0
diag(B0.1)[(nCov - 1):nCov] <- 0.001
diag(B0.2)[(nCov - 1):nCov] <- 0.001
diag(B0.3)[(nCov - 1)] <- 0.001
diag(B0.4)[(nCov - 0)] <- 0.001
diag(B0.5)[(nCov - 0)] <- 0.001

# Restrictions on the effect for 2005
rho.restrictions <- ifelse(outcome.has.eu != 1, 0, NA)

# See Pavlou et al, pg 1163-1164
# Follows soft shrinkage by Rockove et al
c <- 10 # degree of separation between the spike and the slab
delta <- 0.1 # threshold of practical significance
c <- 10 # degree of separation between the spike and the slab
delta <- 0.2 # threshold of practical significance
epsilon <- sqrt(2 * log(c) * c^2 / (c^2 - 1))
varspike <- (delta/epsilon)^2
varslab <- varspike * c^2


# Prepare the predicted probabilities vector,
# to avoid passing them all. Only relevant ones:
# Countries at risk in 2019
L.pi <- plab("pi", list(Year = time.label,
                        Country = country.label,
                        Outcome = outcome.label)) %>%
  mutate(Year = as.numeric(as.character(Year)))
L.pi.relevant <- L.pi %>%
  left_join(select(da, Country, Year, Outcome, Event)) %>%
  filter(Year == max(time.span)) %>%
  filter(Event == 0)
relevant.pp <- as.character(L.pi.relevant$Parameter)


D <- list(
  n0 = n0, nT = nT, nC = nC,
  year_2005 = year.2005, outcome_has_eu = outcome.has.eu,
  outcome_is_tax = outcome.is.tax,
  outcome_is_ets = outcome.is.ets,
  X = unname(X), nCov = nCov,
  X_interdependence = unname(X.interdependence),
#  RW_M_borders = unname(RW.M.borders),
#  RW_M_distances = unname(RW.M.distances),
#  RW_M_trade = unname(RW.M.trade),
#  RW_M_trade_imports = unname(RW.M.trade.imports),
  b0 = b0,
```

```
  B0.1 = B0.1,
  B0.2 = B0.2,
  B0.3 = B0.3,
  B0.4 = B0.4,
  B0.5 = B0.5,
  df = nCov + 1,
  rho = rho.restrictions,
#  d0 = d0, D0 = D0,
#  Omega = Omega, df = nCov + 1,
#  varspike = varspike, varslab = varslab,
  Y_adopted = unname(Y.adopted),
  Y = unname(Y))

write.table(da, file = "exported_treated_data.csv", sep = ";", row.names = FALSE)
```

List the countries that without a policy in place, would make the most influence on other countries because of trade interdependency if they would adopt it.

```
# Loop over every country, and calculate the difference in the trade
# interdependence value that would make if it would have a policy adopted
# as of the last year
trade.diff <- array(NA, dim = c(nC, nO),
                    dimnames = list(Country = country.label, Outcome = outcome.label))
trade.var <- which(dimnames(X.interdependence)[[4]] == "trade.competition")
for (o in 1:nO) {
  for (c in 1:nC) {
    if (Y.adopted[nT,c,o] == 0) {
      #influence.if.adopting <- RW.M.trade[,c,nT]
      influence.if.adopting <- X.interdependence[o,nT,c,trade.var]
      trade.diff[c,o] <- sum(influence.if.adopting, na.rm = TRUE)
    }
  }
}

trade.diff <- tbl_df(as.data.frame.table(trade.diff)) %>%
  rename(Influence = Freq)

tb <- trade.diff %>%
  group_by(Outcome) %>%
  arrange(Outcome, desc(Influence)) %>%
  slice(1:5)

tc <- "Potential aggregated influence of each country if it would change from no adoption to adoption, as m

if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

```
# Report events-per-variable (Pavlou et al)
# Problem when EPR < 10
da %>%
  group_by(Outcome) %>%
  summarize(SumEvents = length(which(Event == 1))) %>%
  mutate(`Events per variable (EPV)` = SumEvents / nCov) %>%
  arrange(desc(`Events per variable (EPV)`))
```

```
→ # A tibble: 5 x 3
→   Outcome                 SumEvents ‘Events per variable (EPV)‘
→   <fct>                       <int>                       <dbl>
→ 1 Carbon pricing                 51                       2.83
→ 2 ETS                            39                       2.17
→ 3 Carbon pricing (no EU)         36                       2
→ 4 Carbon tax                     31                       1.72
→ 5 ETS (no EU)                     8                       0.444
```

```
m <- 'model {
  for (o in 1:nO) {
    for (c in 1:nC) {
      for (t in 1:nT) {
        Y[t,c,o] ~ dbern(pi[t,c,o])
        #logit(pi[t,c,o]) <- alpha[t,o] + inprod(X[t,c,], theta[o,])
        logit(pi[t,c,o]) <- #alpha[o]
                                    inprod(X[t,c,1:(nCov-3)], theta[o,1:(nCov-3)])
                                  + (rho[o] * year_2005[t] * outcome_has_eu[o])
#                                  + theta[o,nCov-3] * X_interdependence[o,t,c,1]
                                  + theta[o,nCov-2] * X_interdependence[o,t,c,1]
                                  + theta[o,nCov-1] * (outcome_is_ets[o] * Y_adopted[t,c,3])
                                  + theta[o,nCov-0]   * (outcome_is_tax[o] * Y_adopted[t,c,4])
      }
    }
    #
    # Kalman filter for time trends
    #
    #alpha[o] ~ dt(-7, 1^-2, 3)
    rho[o] ~ dnorm(2, 3^-2)
#    for (t in 2:nT) {
##      alpha[t,o] ~ dnorm(alpha[t-1,o] +
##                         (rho[o] * year_2005[t] * outcome_has_eu[o]) -
##                         (rho[o] * year_2005[t-1] * outcome_has_eu[o])
##                       , tau.alpha[o])
#      alpha[t,o] ~ dnorm(alpha[t-1,o] +
#                         (rho[o] * year_2005[t] * outcome_has_eu[o]) -
#                         (rho[o] * year_2005[t-1] * outcome_has_eu[o])
#                       , 0.3^-3)
#    }
#    rho[o] ~ dnorm(2, 3^-2)
#    alpha[1,o] ~ dnorm(-8, 3^-2)
###    sigma.alpha[o] ~ dt(0, 0.01, 1)T(0,)
##    tau.alpha[o] ~ dgamma(7, 0.3)
##    sigma.alpha[o] <- 1 / sqrt(tau.alpha[o])
```

```
#    #
    #
    ##### Main effects
    #
  }
  theta[1,1:nCov] ~ dmnorm(b0[1:nCov], Omega.1[1:nCov,1:nCov])
  theta[2,1:nCov] ~ dmnorm(b0[1:nCov], Omega.2[1:nCov,1:nCov])
  theta[3,1:nCov] ~ dmnorm(b0[1:nCov], Omega.3[1:nCov,1:nCov])
  theta[4,1:nCov] ~ dmnorm(b0[1:nCov], Omega.4[1:nCov,1:nCov])
  theta[5,1:nCov] ~ dmnorm(b0[1:nCov], Omega.5[1:nCov,1:nCov])
  Omega.1[1:nCov,1:nCov] ~ dwish(B0.1, df)
  Omega.2[1:nCov,1:nCov] ~ dwish(B0.2, df)
  Omega.3[1:nCov,1:nCov] ~ dwish(B0.3, df)
  Omega.4[1:nCov,1:nCov] ~ dwish(B0.4, df)
  Omega.5[1:nCov,1:nCov] ~ dwish(B0.5, df)
  Sigma[1,1:nCov,1:nCov] ← inverse(Omega.1[1:nCov,1:nCov])
  Sigma[2,1:nCov,1:nCov] ← inverse(Omega.2[1:nCov,1:nCov])
  Sigma[3,1:nCov,1:nCov] ← inverse(Omega.3[1:nCov,1:nCov])
  Sigma[4,1:nCov,1:nCov] ← inverse(Omega.4[1:nCov,1:nCov])
  Sigma[5,1:nCov,1:nCov] ← inverse(Omega.5[1:nCov,1:nCov])
  #
  # Missing data
  #
  for (cov in 1:(nCov-3)) { # obviate the interdependence variables
    for (c in 1:nC) {
      # No time trend for missingness
      for (t in 1:nT) {
        X[t,c,cov] ~ dnorm(0, 1^-2)
      }
    }
  }
  for (cov in 1:1) { # the number of interdependence variables included
    for (c in 1:nC) {
      for (t in 1:nT) {
        for (o in 1:nO) {
          X_interdependence[o,t,c,cov] ~ dnorm(0, 0.5^-2)
        }
      }
    }
  }
}'
write(m, file = paste0("models/model-", M.lab, ".bug"))

par ← NULL
#par ← c(par, "alpha")
#par ← c(par, "sigma.alpha")
par ← c(par, "rho")
par ← c(par, "theta")
#par ← c(par, "sigma.theta")
#par ← c(par, "Theta")
#par ← c(par, "tau.Theta")
```

```
#par ← c(par, "mu.Theta")
#par ← c(par, "delta", "Delta")
#par ← c(par, "sigma.delta")
par ← c(par, "Sigma")
if (run.pcp) {
  par ← c(par, "pi")
} else {
  par ← c(par, relevant.pp) # only selected cases
}
#par ← c(par, "prec_theta")

#inits.alpha ← array(-8, dim = c(nT, nO))
inits.alpha ← array(-8, dim = c(nO))
inits ← list(
  list(.RNG.seed=14717, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha),
  list(.RNG.seed=14718, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha - 2),
  list(.RNG.seed=14719, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha + 2))

t0 ← proc.time()
rj ← run.jags(model = paste0("models/model-", M.lab, ".bug"),
              data = dump.format(D, checkvalid = FALSE),
#              inits = inits,
              modules = c("glm", "lecuyer"),
              n.chains = chains, thin = thin,
              adapt = adapt, burnin = burnin, sample = sample,
              monitor = par, method = "parallel", summarise = FALSE)
s ← as.mcmc.list(rj)
save(s, file = paste0("sample-", M.lab, ".RData"))
proc.time() - t0

→ [1] 3

→ [1] 2500

→ [1] 1

→ [1] 2000

→ [1] 2131

#ggmcmc(ggs(s, family = "^theta|^alpha|^Sigma|^rho"),
ggmcmc(ggs(s, family = "^theta|^alpha|^rho"),
       file = paste0("ggmcmc-", "all", "-", M.lab, ".pdf"),
       plot = c("traceplot", "crosscorrelation", "caterpillar", "geweke"))

ggs(s, family = "^theta\\[1,|rho\\[1") %>%
  ggs_crosscorrelation()

ggmcmc(ggs(s, family = "^alpha\\["),
       file = paste0("ggmcmc-", "alpha", "-", M.lab, ".pdf"),
```

```
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "theta"),
       file = paste0("ggmcmc-", "theta", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "Sigma"),
       file = paste0("ggmcmc-", "Sigma", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "sigma.alpha"),
       file = paste0("ggmcmc-", "sigma_alpha", "-", M.lab, ".pdf"),
       plot = c("traceplot", "crosscorrelation", "caterpillar"))

#ggmcmc(ggs(s, family = "mu.Theta"),
#       file = paste0("ggmcmc-", "mu_Theta", "-", M.lab, ".pdf"),
#       plot = c("traceplot", "crosscorrelation", "caterpillar"))

L.sigma.theta <- plab("sigma.theta", list(Covariate = covariate.label))
L.sigma.delta <- plab("sigma.delta", list(Covariate = c("Interdependence (Borders)",
                                                        "Interdependence (Trade)")))
L.sigma.theta <- bind_rows(L.sigma.theta, L.sigma.delta)
S.sigma.theta <- ggs(s, family = "^sigma.theta\\[|^sigma.delta\\[", par_labels = L.sigma.theta)
ggs_caterpillar(S.sigma.theta) +
  ggtitle("Between outcome standard deviations")
#  geom_vline(xintercept = 0, lty = 3) +

L.theta <- plab("theta", list(Outcome = outcome.label,
                              Covariate = covariate.label))
#L.delta <- plab("delta", list(Outcome = outcome.label,
#                              Covariate = c("Interdependence (Borders)",
#                                            "Interdependence (Trade)")))
#L.theta <- bind_rows(L.theta, L.delta)
S.theta <- ggs(s, family = "^theta\\[|^delta\\[", par_labels = L.theta) %>%
  mutate(Model = M) %>%
  filter(Covariate != "(Intercept)") %>%
  mutate(Covariate = factor(Covariate, rev(covariate.label.order)))

save(S.theta, file = paste("samples-theta-", M.lab, ".RData", sep = ""))

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Outcome)


S.theta %>%
  filter(Outcome %in% c("Carbon pricing", "Carbon pricing (no EU)")) %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Outcome)
```

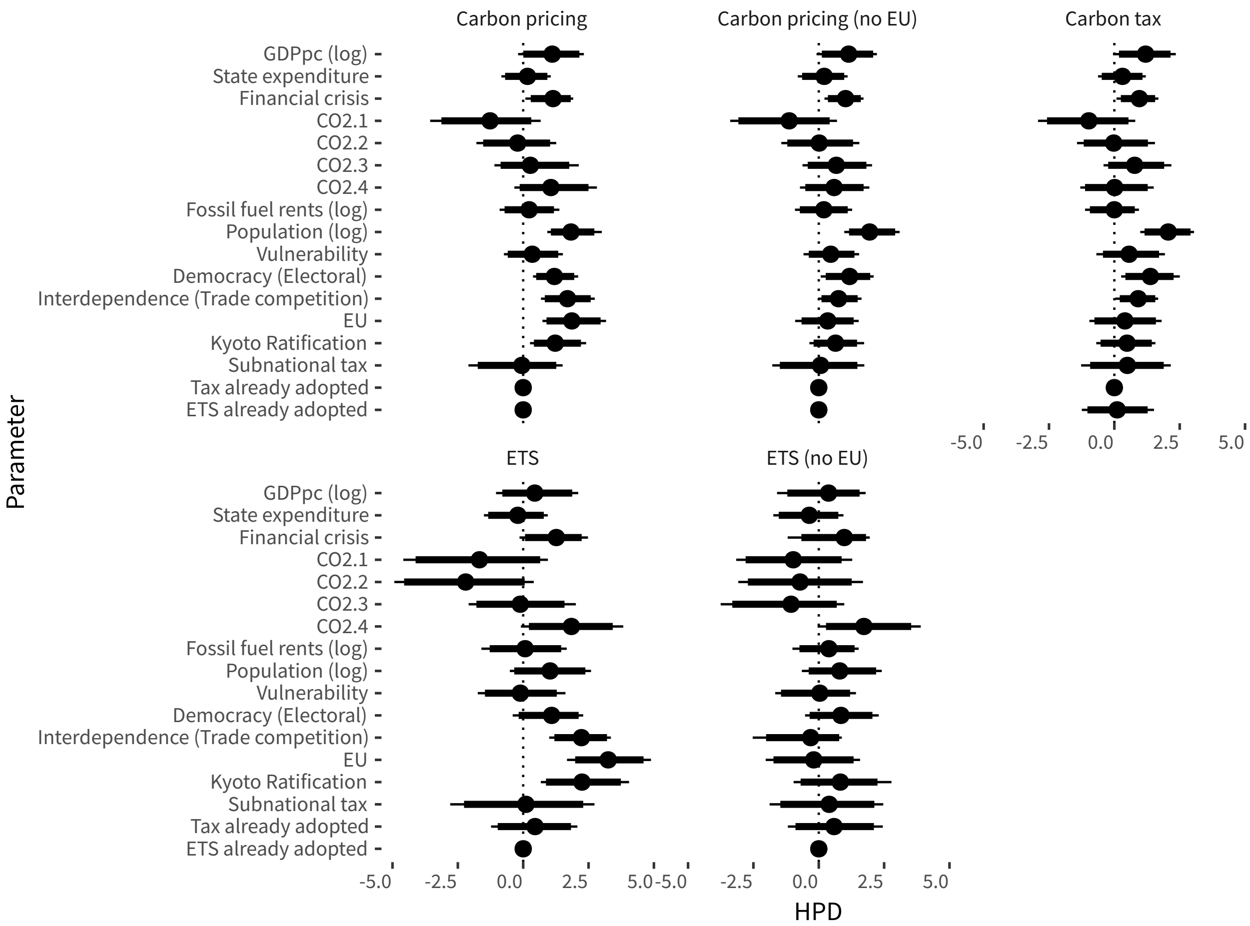


Figure 5.1: HPD of the effects of covariates on the likelihood of the event, by outcome.

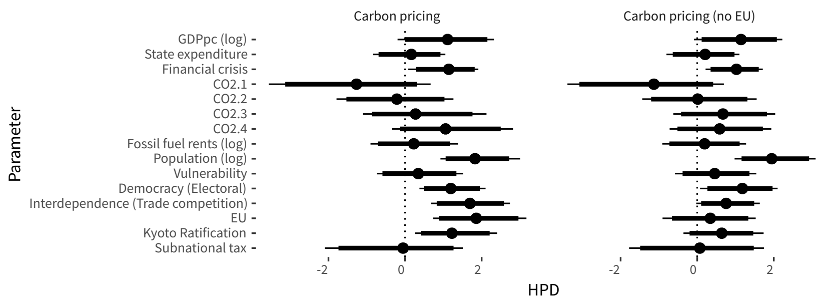


Figure 5.2: HPD of the effects of covariates on the likelihood of the event, by outcome. Only carbon pricing, with and without EU.

```
S.theta %>%
  filter(Outcome %in% c("Carbon pricing")) %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)
```

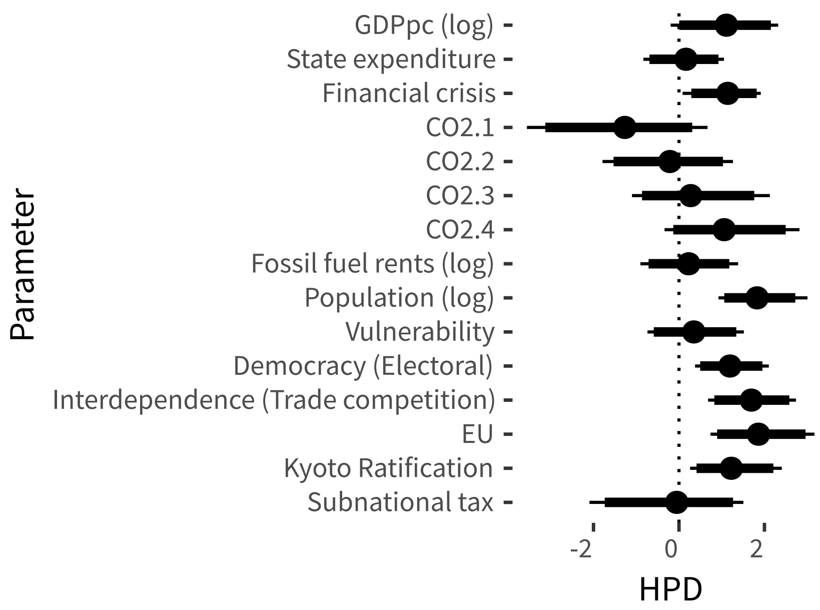


Figure 5.3: HPD of the effects of covariates on the likelihood of the event, by outcome. Only carbon pricing with the EU.

```
ggs_caterpillar(S.theta, label = "Outcome", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Covariate)
```

Variance-covariance matrices

```
L.Sigma.Omega <- plab("Sigma", list(
                              Outcome = outcome.label,
                              Covariate.1 = covariate.label,
                              Covariate.2 = covariate.label))
S.Sigma.Omega <- ggs(s, family = "^Sigma\\[", par_labels = L.Sigma.Omega)

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

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


or <- function(x, significant = 2) {
  or <- as.character(signif((x - 1) * 100, significant))
```

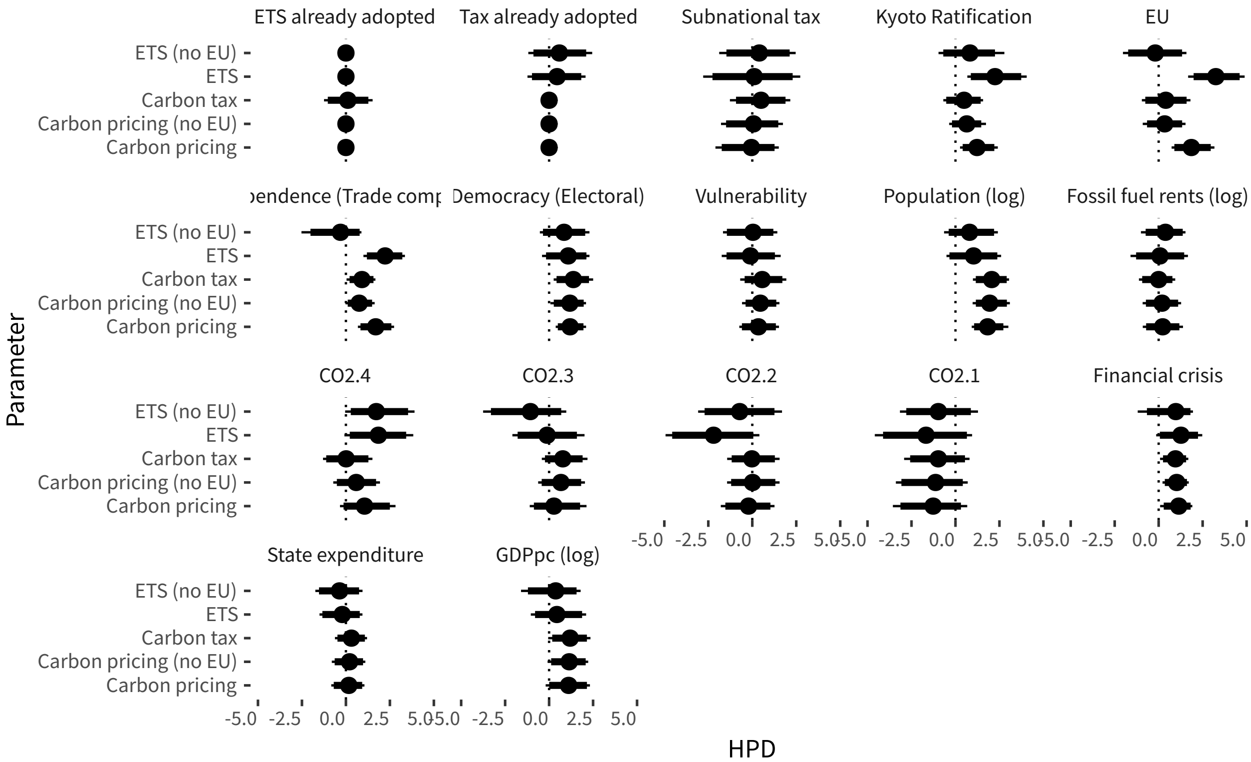


Figure 5.4: HPD of the effects of covariates on the likelihood of the event, by Covariate.

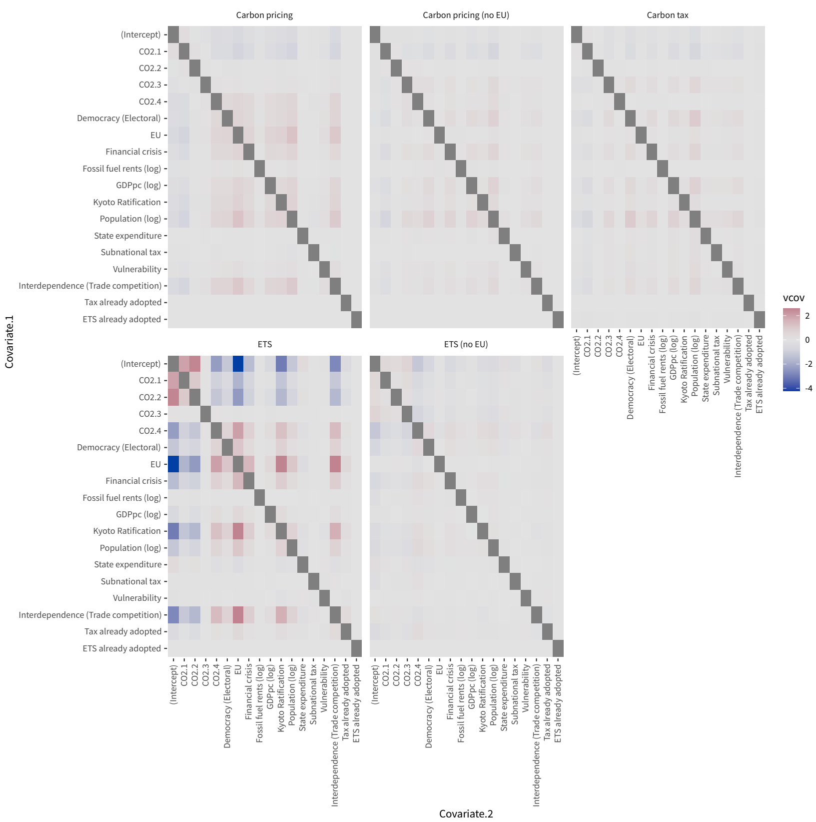


Figure 5.5: Variance-covariance matrix of main effects.

```r
  or[or < 0] <- paste0("\U25Bd ", str_replace(or[or < 0], "-", ""), "%")
  or[or > 0] <- paste0("\U25B3 ", or[or > 0], "%")
  or[or == 0] <- "="
  return(or)
}

tb <- S.theta %>%
  filter(Covariate != "(Intercept)") %>%
  ci() %>%
  arrange(Outcome, desc(abs(median))) %>%
  mutate(`Odds Ratio` = exp(median)) %>%
  mutate(`Expected effect` = or(`Odds Ratio`)) %>%
  select(Outcome, Covariate, `Odds Ratio`, `Expected effect`)

tc <- "Odds ratios of expected effect sizes, and sorted by magnitude and outcome."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 5.1: Odds ratios of expected effect sizes, and sorted by magnitude and outcome.

Outcome	Covariate	Odds Ratio	Expected effect
Carbon pricing	EU	6.4235	△ 540%
Carbon pricing	Population (log)	6.2188	△ 520%
Carbon pricing	Interdependence (Trade competition)	5.4413	△ 440%
Carbon pricing	CO2.1	0.2815	▽ 72%
Carbon pricing	Kyoto Ratification	3.4019	△ 240%
Carbon pricing	Democracy (Electoral)	3.2988	△ 230%
Carbon pricing	Financial crisis	3.1320	△ 210%
Carbon pricing	GDPpc (log)	3.0368	△ 200%
Carbon pricing	CO2.4	2.8812	△ 190%
Carbon pricing	Vulnerability	1.4150	△ 41%
Carbon pricing	CO2.3	1.3153	△ 32%
Carbon pricing	Fossil fuel rents (log)	1.2567	△ 26%
Carbon pricing	CO2.2	0.8083	▽ 19%
Carbon pricing	State expenditure	1.1786	△ 18%
Carbon pricing	Subnational tax	0.9498	▽ 5%
Carbon pricing	ETS already adopted	0.9998	▽ 0.02%
Carbon pricing	Tax already adopted	1.0001	△ 0.0064%
Carbon pricing (no EU)	Population (log)	6.9993	△ 600%
Carbon pricing (no EU)	Democracy (Electoral)	3.2533	△ 230%
Carbon pricing (no EU)	GDPpc (log)	3.1406	△ 210%
Carbon pricing (no EU)	CO2.1	0.3226	▽ 68%
Carbon pricing (no EU)	Financial crisis	2.7872	△ 180%
Carbon pricing (no EU)	Interdependence (Trade competition)	2.1282	△ 110%
Carbon pricing (no EU)	CO2.3	1.9592	△ 96%
Carbon pricing (no EU)	Kyoto Ratification	1.8994	△ 90%
Carbon pricing (no EU)	CO2.4	1.7965	△ 80%
Carbon pricing (no EU)	Vulnerability	1.5793	△ 58%
Carbon pricing (no EU)	EU	1.4086	△ 41%

Carbon pricing (no EU)	State expenditure	1.2307	△ 23%
Carbon pricing (no EU)	Fossil fuel rents (log)	1.2142	△ 21%
Carbon pricing (no EU)	Subnational tax	1.0716	△ 7.2%
Carbon pricing (no EU)	CO2.2	1.0151	△ 1.5%
Carbon pricing (no EU)	Tax already adopted	0.9994	▽ 0.063%
Carbon pricing (no EU)	ETS already adopted	1.0002	△ 0.019%
Carbon tax	Population (log)	7.8546	△ 690%
Carbon tax	Democracy (Electoral)	3.9687	△ 300%
Carbon tax	GDPpc (log)	3.3098	△ 230%
Carbon tax	CO2.1	0.3765	▽ 62%
Carbon tax	Financial crisis	2.6021	△ 160%
Carbon tax	Interdependence (Trade competition)	2.4878	△ 150%
Carbon tax	CO2.3	2.1773	△ 120%
Carbon tax	Vulnerability	1.7575	△ 76%
Carbon tax	Subnational tax	1.6447	△ 64%
Carbon tax	Kyoto Ratification	1.6275	△ 63%
Carbon tax	EU	1.5004	△ 50%
Carbon tax	State expenditure	1.3658	△ 37%
Carbon tax	ETS already adopted	1.1139	△ 11%
Carbon tax	CO2.2	0.9768	▽ 2.3%
Carbon tax	CO2.4	1.0123	△ 1.2%
Carbon tax	Fossil fuel rents (log)	0.9988	▽ 0.12%
Carbon tax	Tax already adopted	0.9996	▽ 0.044%
ETS	EU	25.8159	△ 2500%
ETS	Kyoto Ratification	9.4798	△ 850%
ETS	Interdependence (Trade competition)	9.2869	△ 830%
ETS	CO2.2	0.1107	▽ 89%
ETS	CO2.4	6.3223	△ 530%
ETS	CO2.1	0.1876	▽ 81%
ETS	Financial crisis	3.5545	△ 260%
ETS	Democracy (Electoral)	2.9668	△ 200%
ETS	Population (log)	2.7966	△ 180%
ETS	Tax already adopted	1.5744	△ 57%
ETS	GDPpc (log)	1.5559	△ 56%
ETS	State expenditure	0.8134	▽ 19%
ETS	CO2.3	0.8936	▽ 11%
ETS	Vulnerability	0.8975	▽ 10%
ETS	Subnational tax	1.1112	△ 11%
ETS	Fossil fuel rents (log)	1.0763	△ 7.6%
ETS	ETS already adopted	1.0008	△ 0.076%
ETS (no EU)	CO2.4	5.6182	△ 460%
ETS (no EU)	CO2.3	0.3458	▽ 65%
ETS (no EU)	CO2.1	0.3768	▽ 62%
ETS (no EU)	Financial crisis	2.6535	△ 170%
ETS (no EU)	Democracy (Electoral)	2.3352	△ 130%
ETS (no EU)	Kyoto Ratification	2.2816	△ 130%
ETS (no EU)	Population (log)	2.2293	△ 120%
ETS (no EU)	CO2.2	0.4887	▽ 51%
ETS (no EU)	Tax already adopted	1.7941	△ 79%
ETS (no EU)	Subnational tax	1.4894	△ 49%
ETS (no EU)	Fossil fuel rents (log)	1.4704	△ 47%
ETS (no EU)	GDPpc (log)	1.4543	△ 45%
ETS (no EU)	State expenditure	0.6951	▽ 30%
ETS (no EU)	Interdependence (Trade competition)	0.7338	▽ 27%
ETS (no EU)	EU	0.8241	▽ 18%
ETS (no EU)	Vulnerability	1.0398	△ 4%

ETS (no EU)	ETS already adopted	1.0005	△ 0.05%

```r
require(xtable)
tb.dump <- tb %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
  select(-Outcome)
tc <- "Odds ratios of expected effect sizes, by magnitude."

print(xtable(tb.dump,
             caption = tc,
             label = "tab:ees",
             tabular.environment = "longtable"),
      file = "table_expected_effect_sizes-002.tex",
      #size = "footnotesize",
      size = "normalsize",
      include.rownames = FALSE)

S.theta %>%
  filter(Outcome %in% c("Carbon tax", "ETS")) %>%
  ci() %>%
  ggplot(aes(ymin = low, ymax = high,
             y = median, x = Covariate,
             color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.3)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Harmonic")
```

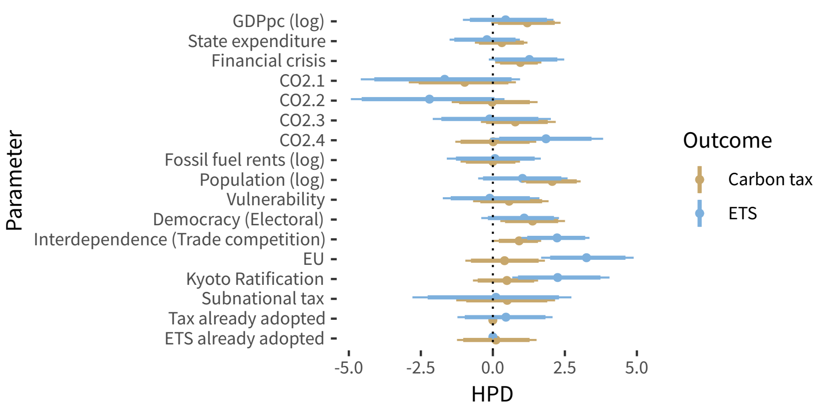


Figure 5.6: HPD of the effects of covariates on the likelihood of the event, by outcome. Only Tax and ETS.

```r
L.alpha <- plab("alpha", list(Outcome = outcome.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_labels = L.alpha) %>%
  mutate(value = inv.logit(value))
ggs_caterpillar(S.alpha)
```

```r
L.alpha <- plab("alpha", list(Year = time.label, Outcome = outcome.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_labels = L.alpha) %>%
  mutate(Year = as.numeric(as.character(Year))) %>%
  mutate(value = inv.logit(value))

ci(S.alpha, thick_ci = c(0.1, 0.9)) %>%
  ggplot(aes(x = Year, y = median, color = Outcome)) +
  geom_line() +
  ylab("Hazard")

ci(S.alpha, thick_ci = c(0.1, 0.9)) %>%
  ggplot(aes(x = Year, y = median, ymin = low, ymax = high,
             fill = Outcome, color = Outcome)) +
  geom_line() +
  geom_ribbon(alpha = 0.3) +
  facet_wrap(~ Outcome) +
  ylab("Hazard")
```

List the countries most likely to adopt each policy.

```r
L.pi <- plab("pi", list(Year = time.label,
                        Country = country.label,
                        Outcome = outcome.label)) %>%
  mutate(Year = as.numeric(as.character(Year)))
S.pi <- ggs(s, family = "^pi\\[", par_labels = L.pi)

tpp <- S.pi %>%
  mutate(value = inv.logit(value)) %>%
  ci() %>%
  select(Country, Outcome, `Predicted probability` = median) %>%
  group_by(Outcome) %>%
  arrange(desc(`Predicted probability`)) %>%
  slice(1:5)


tc <- "Countries with higher posterior median predicted probabilities of adopting each policy
outcome. Top 5 by Outcome."
if (knitr::is_latex_output()) {
  kable(tpp, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tpp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 5.2: Countries with higher posterior median predicted probabilities of adopting each policy outcome. Top 5 by Outcome.

Country	Outcome	Predicted probability
Cuba	Carbon pricing	0.5049
Grenada	Carbon pricing	0.5046
Georgia	Carbon pricing	0.5029
Equatorial Guinea	Carbon pricing	0.5029

Colombia	Carbon pricing	0.5026
Canada	Carbon pricing (no EU)	0.5106
Estonia	Carbon pricing (no EU)	0.5073
Cyprus	Carbon pricing (no EU)	0.5039
Grenada	Carbon pricing (no EU)	0.5038
Jamaica	Carbon pricing (no EU)	0.5034
Canada	Carbon tax	0.5080
Cyprus	Carbon tax	0.5059
Estonia	Carbon tax	0.5057
Equatorial Guinea	Carbon tax	0.5036
Angola	Carbon tax	0.5034
Haiti	ETS	0.5069
Cuba	ETS	0.5036
Comoros	ETS	0.5019
Colombia	ETS	0.5009
Argentina	ETS	0.5006
Haiti	ETS (no EU)	0.5033
Comoros	ETS (no EU)	0.5021
France	ETS (no EU)	0.5010
Kenya	ETS (no EU)	0.5008
Grenada	ETS (no EU)	0.5008

```r
S.pi

# Manually calculate PCP
# as ggmcmc's ggs_pcp() is not ready for matrices as input for outcome

threshold <- da %>%
  group_by(Outcome) %>%
  summarize(Threshold = length(which(Event == 1)) / n())

S <- inner_join(S.pi, select(da, Country, Year, Outcome, Event)) %>%
  left_join(threshold) %>%
  mutate(Correct = if_else( (value < Threshold & Event == 0) |
                            (value > Threshold & Event == 1),
                          TRUE, FALSE)) %>%
  group_by(Outcome, Iteration, Chain) %>%
  summarize(PCP = length(which(Correct)) / n())


ggplot(S, aes(x = PCP)) +
  geom_histogram() +
  facet_grid(~ Outcome) +
  expand_limits(x = c(0, 1))

t.pcp <- S %>%
  group_by(Outcome) %>%
  summarize(`Median PCP` = mean(PCP))

tc <- "Posterior median percent correctly predicted, by Outcome."
if (knitr::is_latex_output()) {
  kable(tpp, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
```

```
    kable_styling(font_size = 10)
} else {
  kable(tpp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}

rm(S)
invisible(gc())
```

6

Model: Baseline 003

```
# Model:
M
```

```
→ [1] "Baseline 003"
```

- Event history analysis using logistic regression for discrete time (years).
- Time dependence is fixed over the period, with a dummy in 2005 and for EU models to capture the effect of EU in a single period. More complex time dependency does not show any significant gain and complicates the estimation.
- Bayesian inference with weakly informative priors.

Data preparation:

- Standardization to 0.5 standard deviations for all covariates, following Gelman (2008).

Model equation:

$$
\begin{aligned}
Y_{t,c,o} \sim & \quad \mathcal{B}(\pi_{t,c,o}) && \text{Main data component} \\
\pi_{t,c,o} = & \quad \text{logit}\alpha_o + \rho_o + (\theta_{o,v} X_{t,c,v}) && \text{Main linear model} \\
\alpha_o \sim & \quad \mathcal{N}(-7, 2) && \text{Rare event prior} \\
\rho_o \sim & \quad \mathcal{N}(0, 2) && \text{Year 2005 effect} \\
\theta_{o,v} \sim & \quad \mathcal{MVN}(0, \Sigma_o) && \text{Priors for explanatory variables} \\
\Sigma_o \sim & \quad \mathcal{W}(0, 1) && \text{Prior for the variance-covariance matrix}
\end{aligned}
$$

Where:

- o: Outcomes
- t: Time
- c: Country
- $y_{t,c,o}$: Binary variable that captures whether in a specific outcome (o), country (c) and year (t) there has been an adoption of a policy (1) or not (0).
- α_o: Baseline hazard.
- ρ_o: Shock in 2005 produced by simultaneous EU adoption.
- $\theta_{o,v}$: Effects of covariates (v), in each outcome o.
- Σ_o: Variance-covariance matrix to minimize multicollinearity.

Deal with interdependence data either in matrix or tidy formats

```
# No need to delete countries once the event has happened
# as they are already NA, and are needed for the full dataset
#da ← d %>%
#  filter(!is.na(Event))
da ← d
if (test) {
  da ← da %>%
    filter(Country %in% sample(unique(da$Country), size = length(unique(da$Country)) * 0.5))
}

##############################################################################
###### Interdependence using tidy approach
contiguity ←
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.contiguous) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(contiguity.dependency = sum(wAdopted, na.rm = TRUE))

distance ←
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.distance),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.distance) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(distance.dependency = sum(wAdopted, na.rm = TRUE))

trade.dependency ←
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(trade.dependency = sum(wAdopted, na.rm = TRUE)) %>%
```

```
  ungroup()

# For competition dependency we need both the imports and the exports

trade.partner.dependency <-
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
#  select(Origin, Destination, Year, Outcome, wAdopted) %>%
  mutate(wAdopted = ifelse(is.na(wAdopted), 0, wAdopted)) %>%
  ungroup()

trade.partner.others.imports <-
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Imports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted.imports = Adopted * p.Imports) %>%
  filter(Origin != Destination) %>%
  # Rename countries to better control the matrices
  # Partner refers to the trade partner of the first origin country
  # ThirdCountry refers to the competitor of the first origin country
  #   through the Destination==Partner
  rename(Partner = Origin, ThirdCountry = Destination) %>%
  select(Partner, ThirdCountry, Year, Outcome, wAdopted.imports) %>%
  mutate(wAdopted.imports = ifelse(is.na(wAdopted.imports), 0, wAdopted.imports)) %>%
  group_by(Partner, Year, Outcome) %>%
  summarize(w.ThirdCountry.imp = sum(wAdopted.imports)) %>%
  ungroup()

trade.competition <-
  left_join(trade.partner.dependency,
            trade.partner.others.imports,
            by = c("Destination" = "Partner",
                   "Year" = "Year",
                   "Outcome" = "Outcome")) %>%
  mutate(wAdopted.dual = wAdopted * w.ThirdCountry.imp) %>%
  filter(Origin != Destination) %>%
  group_by(Origin, Year, Outcome) %>%
  summarize(trade.competition = sum(wAdopted.dual, na.rm = TRUE)) %>%
  ungroup() %>%
  rename(Country = Origin)


##### End interdependence using tidy approach
##############################################################################
```

```
################################################################################
####### Interdependence using oWeighting matrices
## First use only relevant countries and then row-normalize
#M.distances <- M.distances[match(country.label, dimnames(M.distances)[[1]]),
#                           match(country.label, dimnames(M.distances)[[2]])]
#RW.M.distances <- 100 * (1 / M.distances) /
#  apply(1 / M.distances, 1, sum, na.rm = TRUE)
#RW.M.distances[is.na(RW.M.distances)] <- 0
#stopifnot(dimnames(RW.M.distances)[[1]] == country.label)
#stopifnot(dimnames(RW.M.distances)[[2]] == country.label)
#
#M.borders <- M.borders[match(country.label, dimnames(M.borders)[[1]]),
#                           match(country.label, dimnames(M.borders)[[2]])]
#RW.M.borders <- M.borders / apply(M.borders, 1, sum, na.rm = TRUE)
#RW.M.borders[is.na(RW.M.borders)] <- 0
#stopifnot(dimnames(RW.M.borders)[[1]] == country.label)
#stopifnot(dimnames(RW.M.borders)[[2]] == country.label)
#
## Trade exports (dependency)
#M.trade <- M.trade[match(country.label, dimnames(M.trade)[[1]]),
#                           match(country.label, dimnames(M.trade)[[2]]),]
#RW.M.trade <- array(0, dim = dim(M.trade), dimnames = dimnames(M.trade))
#for (t in 1:nT) {
#  RW.M.trade[,,t] <- M.trade[,,t] / apply(M.trade[,,t], 1, sum, na.rm = TRUE)
#}
#
#RW.M.trade[is.na(RW.M.trade)] <- 0
#stopifnot(dimnames(RW.M.trade)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade)[[2]] == country.label)
#
## Trade imports
#M.trade.imports <- M.trade.imports[match(country.label, dimnames(M.trade.imports)[[1]]),
#                           match(country.label, dimnames(M.trade.imports)[[2]]),]
#RW.M.trade.imports <- array(0, dim = dim(M.trade.imports), dimnames = dimnames(M.trade.imports))
#for (t in 1:nT) {
#  RW.M.trade.imports[,,t] <- M.trade.imports[,,t] / apply(M.trade.imports[,,t], 1, sum, na.rm = TRUE)
#}
#
#RW.M.trade.imports[is.na(RW.M.trade.imports)] <- 0
#stopifnot(dimnames(RW.M.trade.imports)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade.imports)[[2]] == country.label)
#
#
###### End interdependence using matrices
################################################################################

Y <- reshape2::acast(da, Year ~ Country ~ Outcome, value.var = "Event")
Y.adopted <- reshape2::acast(da, Year ~ Country ~ Outcome, value.var = "Adopted")

# Work with fewer data
```

```
#if (test) {
#  Y <- Y[-c(1:10),,c(1,2)]
#  Y.adopted <- Y.adopted[-c(1:10),,c(1,2)]
#}

# Time
nT <- dim(Y)[1]
time.label <- time.span
#if (test) time.label <- time.label[-c(1:10)]
stopifnot(nT == length(time.label))
year.2005 <- ifelse(time.label == 2005, 1, 0)

# Outcomes
nO <- dim(Y)[3]
outcome.label <- dimnames(Y)[[3]]
outcome.has.eu <- ifelse(outcome.label %in% c("Carbon pricing", "ETS"), 1, 0)
outcome.is.tax <- ifelse(outcome.label %in% c("Carbon tax"), 1, 0)
outcome.is.ets <- ifelse(outcome.label %in% c("ETS", "ETS (no EU)"), 1, 0)


# Countries and covariates
X <- da %>%
  ### Non binary variables
  select(Country, Year,
         `GDPpc (log)`,
         `State expenditure`,
         `Population (log)`, # OK
         `Fossil fuel rents (log)`,
         `Democracy (Electoral)`, # not very well
#         `Political constraints`, # quite bad
#         `Government effectiveness`, # reasonable
#         #`Debt (log)`,
         `Vulnerability`, # not very well
         `CO2pc (log)`) %>% # not very well
  unique() %>%
  gather(Variable, value, -Country, -Year) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  ### Add binary variables
  bind_rows(select(da, Country, Year,
                   `Kyoto Ratification`,  # problematic
                   #`Paris Ratification`,
                   `Financial crisis`, # problematic
                   `Subnational tax`, # works well
                   `EU`) %>%  # works well
            unique() %>%
            gather(Variable, value, -Country, -Year)) %>%
  ###
  ### Add EU * GDPpc interaction
#  spread(Variable, value) %>%
```

```
#  mutate(`EU * GDPpc (log)` = EU * `GDPpc (log)`) %>%
#  gather(Variable, value, -c(Country, Year)) %>%
  ###
  ### Add Intercept
  spread(Variable, value) %>%
  mutate(`(Intercept)` = 1) %>%
  gather(Variable, value, -c(Country, Year)) %>%
  ###
  reshape2::acast(Year ~ Country ~ Variable, value.var = "value")
country.label <- dimnames(X)[[2]]
nC <- length(country.label)
stopifnot(dimnames(Y)[[2]] == dimnames(X)[[2]])
#if (test) X <- X[-c(1:10),,]
##X <- cbind(1, X)
##dimnames(X)[[2]][1] <- "(Intercept)"

X.interdependence <-
  select(da, Country, Year, Outcome) %>%
  left_join(contiguity) %>%
  left_join(distance) %>%
  left_join(trade.dependency) %>%
  left_join(trade.competition) %>%
  mutate(Country = as.factor(Country),
         Year = as.integer(Year),
         Outcome = as.factor(Outcome)) %>%
  gather(Variable, value, -c(Country, Year, Outcome)) %>%
  group_by(Variable, Outcome) %>%
  mutate(value = std.zero(value)) %>%
  ungroup() %>%
#  filter(Variable != "distance.dependency") %>%
  mutate(Variable = factor(Variable, levels = c("contiguity.dependency",
                                                "distance.dependency",
                                                "trade.dependency",
                                                "trade.competition"))) %>%
##  spread(Variable, value) %>%
##  left_join(select(da, Country, Year, Outcome, `EU`)) %>%
##  mutate(eu.trade.dependency = EU * trade.dependency) %>%
##  select(-EU) %>%
##  gather(Variable, value, -c(Country, Year,  Outcome)) %>%
#  filter(Variable %in% c("trade.competition")) %>%
  filter(Variable %in% c("trade.dependency")) %>%
##  filter(Variable %in% c("contiguity.dependency", "trade.competition")) %>%
  reshape2::acast(Outcome ~ Year ~ Country ~ Variable, value.var = "value")
stopifnot(dimnames(Y)[[2]] == dimnames(X.interdependence)[[3]])


# Mix covariates in the X object with those of the spatial matrices
covariate.label <- c(dimnames(X)[[3]],
#                      "Interdependence (Borders)",
#                      "EU * Interdependence (Trade dependency)",
#                      "Interdependence (Trade competition)",
```

```
                     "Interdependence (Trade dependency)",
                     "Tax already adopted", "ETS already adopted")
nCov ← length(covariate.label)

covariate.label.order ← c(
  "GDPpc (log)",                              # Economic
#  "EU * GDPpc (log)",
  "State expenditure",
  "Financial crisis", #
  "CO2pc (log)",                              # Contribution to CC
  "Fossil fuel rents (log)",
  "Population (log)", "Vulnerability",
  "Democracy (Electoral)",                    # Institutional
#  "Interdependence (Borders)",                # Interdependence
  "Interdependence (Trade dependency)",
#  "Interdependence (Trade competition)",
#  "EU * Interdependence (Trade dependency)",
  "EU",
  "Kyoto Ratification",
  "Subnational tax",
  "Tax already adopted", "ETS already adopted")


b0 ← rep(0, nCov)
b0[1] ← -7            # prior for rare events in the intercept
B0 ← diag(nCov)
#diag(B0) ← 2.5^-2
diag(B0) ← 1^-2
diag(B0) ← 0.5^-2
diag(B0) ← 10
#Omega ← diag(nCov)
#diag(Omega) ← 0.2^-2
#d0 ← rep(0, 2)
#D0 ← diag(2)
#diag(D0) ← 1^-2

# Restrictions on already adopted
B0.1 ← B0.2 ← B0.3 ← B0.4 ← B0.5 ← B0
diag(B0.1)[(nCov - 1):nCov] ← 0.001
diag(B0.2)[(nCov - 1):nCov] ← 0.001
diag(B0.3)[(nCov - 1)] ← 0.001
diag(B0.4)[(nCov - 0)] ← 0.001
diag(B0.5)[(nCov - 0)] ← 0.001

# Restrictions on the effect for 2005
rho.restrictions ← ifelse(outcome.has.eu ≠ 1, 0, NA)

# See Pavlou et al, pg 1163-1164
# Follows soft shrinkage by Rockove et al
c ← 10 # degree of separation between the spike and the slab
delta ← 0.1 # threshold of practical significance
```

```
c ← 10 # degree of separation between the spike and the slab
delta ← 0.2 # threshold of practical significance
epsilon ← sqrt(2 * log(c) * c^2 / (c^2 - 1))
varspike ← (delta/epsilon)^2
varslab ← varspike * c^2


# Prepare the predicted probabilities vector,
# to avoid passing them all. Only relevant ones:
# Countries at risk in 2019
L.pi ← plab("pi", list(Year = time.label,
                        Country = country.label,
                        Outcome = outcome.label)) %>%
  mutate(Year = as.numeric(as.character(Year)))
L.pi.relevant ← L.pi %>%
  left_join(select(da, Country, Year, Outcome, Event)) %>%
  filter(Year == max(time.span)) %>%
  filter(Event == 0)
relevant.pp ← as.character(L.pi.relevant$Parameter)


D ← list(
  n0 = n0, nT = nT, nC = nC,
  year_2005 = year.2005, outcome_has_eu = outcome.has.eu,
  outcome_is_tax = outcome.is.tax,
  outcome_is_ets = outcome.is.ets,
  X = unname(X), nCov = nCov,
  X_interdependence = unname(X.interdependence),
#  RW_M_borders = unname(RW.M.borders),
#  RW_M_distances = unname(RW.M.distances),
#  RW_M_trade = unname(RW.M.trade),
#  RW_M_trade_imports = unname(RW.M.trade.imports),
  b0 = b0,
  B0.1 = B0.1,
  B0.2 = B0.2,
  B0.3 = B0.3,
  B0.4 = B0.4,
  B0.5 = B0.5,
  df = nCov + 1,
  rho = rho.restrictions,
#  d0 = d0, D0 = D0,
#  Omega = Omega, df = nCov + 1,
#  varspike = varspike, varslab = varslab,
  Y_adopted = unname(Y.adopted),
  Y = unname(Y))

write.table(da, file = "exported_treated_data.csv", sep = ";", row.names = FALSE)
```

List the countries that without a policy in place, would make the most influence on other countries because of trade interdependency if they would adopt it.

```
# This is part of how to calculate trade dependency
tb <-
  # The country that has adopted or not is the destination country
  select(da, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
              select(Origin, Destination, Year, p.Exports),
              by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  arrange(desc(p.Exports)) %>%
  filter(Country %in% country.coverage) %>%
  filter(Adopted == 0) %>% # only consider non-adopters
  filter(Year == max(time.span)) %>%
  group_by(Outcome, Destination) %>%
  summarize(av.p.exports = mean(p.Exports, na.rm = TRUE)) %>%
  ungroup() %>%
  group_by(Outcome) %>%
  arrange(Outcome, desc(av.p.exports)) %>%
  slice(1:5)



tc <- "Potential aggregated influence of each country if it would change from no adoption to adoption, as m

if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}

# Report events-per-variable (Pavlou et al)
# Problem when EPR < 10
da %>%
  group_by(Outcome) %>%
  summarize(SumEvents = length(which(Event == 1))) %>%
  mutate(`Events per variable (EPV)` = SumEvents / nCov) %>%
  arrange(desc(`Events per variable (EPV)`))
```

```
→ # A tibble: 5 x 3
→   Outcome                SumEvents 'Events per variable (EPV)'
→   <fct>                      <int>                        <dbl>
→ 1 Carbon pricing                51                        3.4
→ 2 ETS                           39                        2.6
→ 3 Carbon pricing (no EU)        36                        2.4
→ 4 Carbon tax                    31                        2.07
→ 5 ETS (no EU)                    8                        0.533
```

```
m <- 'model {
  for (o in 1:nO) {
```

```
    for (c in 1:nC) {
      for (t in 1:nT) {
        Y[t,c,o] ~ dbern(pi[t,c,o])
        #logit(pi[t,c,o]) <- alpha[t,o] + inprod(X[t,c,], theta[o,])
        logit(pi[t,c,o]) <- #alpha[o]
                                       inprod(X[t,c,1:(nCov-3)], theta[o,1:(nCov-3)])
                                     + (rho[o] * year_2005[t] * outcome_has_eu[o])
#                                     + theta[o,nCov-3] * X_interdependence[o,t,c,1]
                                     + theta[o,nCov-2] * X_interdependence[o,t,c,1]
                                     + theta[o,nCov-1] * (outcome_is_ets[o] * Y_adopted[t,c,3])
                                     + theta[o,nCov-0]   * (outcome_is_tax[o] * Y_adopted[t,c,4])
      }
    }
    #
    # Kalman filter for time trends
    #
    #alpha[o] ~ dt(-7, 1^-2, 3)
    rho[o] ~ dnorm(2, 3^-2)
#     for (t in 2:nT) {
##      alpha[t,o] ~ dnorm(alpha[t-1,o] +
##                          (rho[o] * year_2005[t] * outcome_has_eu[o]) -
##                          (rho[o] * year_2005[t-1] * outcome_has_eu[o])
##                        , tau.alpha[o])
#       alpha[t,o] ~ dnorm(alpha[t-1,o] +
#                          (rho[o] * year_2005[t] * outcome_has_eu[o]) -
#                          (rho[o] * year_2005[t-1] * outcome_has_eu[o])
#                        , 0.3^-3)
#     }
#     rho[o] ~ dnorm(2, 3^-2)
#     alpha[1,o] ~ dnorm(-8, 3^-2)
###     sigma.alpha[o] ~ dt(0, 0.01, 1)T(0,)
##     tau.alpha[o] ~ dgamma(7, 0.3)
##     sigma.alpha[o] <- 1 / sqrt(tau.alpha[o])
#     #
    #
    ##### Main effects
    #
  }
  theta[1,1:nCov] ~ dmnorm(b0[1:nCov], Omega.1[1:nCov,1:nCov])
  theta[2,1:nCov] ~ dmnorm(b0[1:nCov], Omega.2[1:nCov,1:nCov])
  theta[3,1:nCov] ~ dmnorm(b0[1:nCov], Omega.3[1:nCov,1:nCov])
  theta[4,1:nCov] ~ dmnorm(b0[1:nCov], Omega.4[1:nCov,1:nCov])
  theta[5,1:nCov] ~ dmnorm(b0[1:nCov], Omega.5[1:nCov,1:nCov])
  Omega.1[1:nCov,1:nCov] ~ dwish(B0.1, df)
  Omega.2[1:nCov,1:nCov] ~ dwish(B0.2, df)
  Omega.3[1:nCov,1:nCov] ~ dwish(B0.3, df)
  Omega.4[1:nCov,1:nCov] ~ dwish(B0.4, df)
  Omega.5[1:nCov,1:nCov] ~ dwish(B0.5, df)
  Sigma[1,1:nCov,1:nCov] <- inverse(Omega.1[1:nCov,1:nCov])
  Sigma[2,1:nCov,1:nCov] <- inverse(Omega.2[1:nCov,1:nCov])
  Sigma[3,1:nCov,1:nCov] <- inverse(Omega.3[1:nCov,1:nCov])
```

```
  Sigma[4,1:nCov,1:nCov] <- inverse(Omega.4[1:nCov,1:nCov])
  Sigma[5,1:nCov,1:nCov] <- inverse(Omega.5[1:nCov,1:nCov])
  #
  # Missing data
  #
  for (cov in 1:(nCov-3)) { # obviate the interdependence variables
    for (c in 1:nC) {
      # No time trend for missingness
      for (t in 1:nT) {
        X[t,c,cov] ~ dnorm(0, 1^-2)
      }
    }
  }
  for (cov in 1:1) { # the number of interdependence variables included
    for (c in 1:nC) {
      for (t in 1:nT) {
        for (o in 1:nO) {
          X_interdependence[o,t,c,cov] ~ dnorm(0, 0.5^-2)
        }
      }
    }
  }
}'
write(m, file = paste0("models/model-", M.lab, ".bug"))

par <- NULL
#par <- c(par, "alpha")
#par <- c(par, "sigma.alpha")
par <- c(par, "rho")
par <- c(par, "theta")
#par <- c(par, "sigma.theta")
#par <- c(par, "Theta")
#par <- c(par, "tau.Theta")
#par <- c(par, "mu.Theta")
#par <- c(par, "delta", "Delta")
#par <- c(par, "sigma.delta")
par <- c(par, "Sigma")
if (run.pcp) {
  par <- c(par, "pi")
} else {
  par <- c(par, relevant.pp) # only selected cases
}
#par <- c(par, "prec_theta")

#inits.alpha <- array(-8, dim = c(nT, nO))
inits.alpha <- array(-8, dim = c(nO))
inits <- list(
  list(.RNG.seed=14717, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha),
  list(.RNG.seed=14718, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha - 2),
```

```
  list(.RNG.seed=14719, .RNG.name="base::Mersenne-Twister",
       alpha = inits.alpha + 2))

t0 ← proc.time()
rj ← run.jags(model = paste0("models/model-", M.lab, ".bug"),
              data = dump.format(D, checkvalid = FALSE),
#              inits = inits,
              modules = c("glm", "lecuyer"),
              n.chains = chains, thin = thin,
              adapt = adapt, burnin = burnin, sample = sample,
              monitor = par, method = "parallel", summarise = FALSE)
s ← as.mcmc.list(rj)
save(s, file = paste0("sample-", M.lab, ".RData"))
proc.time() - t0

→ [1] 3

→ [1] 2500

→ [1] 1

→ [1] 2000

→ [1] 2056

#ggmcmc(ggs(s, family = "^theta|^alpha|^Sigma|^rho"),
ggmcmc(ggs(s, family = "^theta|^alpha|^rho"),
       file = paste0("ggmcmc-", "all", "-", M.lab, ".pdf"),
       plot = c("traceplot", "crosscorrelation", "caterpillar", "geweke"))

ggs(s, family = "^theta\\[1,|rho\\[1") %>%
  ggs_crosscorrelation()

ggmcmc(ggs(s, family = "^alpha\\["),
       file = paste0("ggmcmc-", "alpha", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "theta"),
       file = paste0("ggmcmc-", "theta", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "Sigma"),
       file = paste0("ggmcmc-", "Sigma", "-", M.lab, ".pdf"),
       plot = c("traceplot", "running", "crosscorrelation", "caterpillar"))

ggmcmc(ggs(s, family = "sigma.alpha"),
       file = paste0("ggmcmc-", "sigma_alpha", "-", M.lab, ".pdf"),
       plot = c("traceplot", "crosscorrelation", "caterpillar"))

#ggmcmc(ggs(s, family = "mu.Theta"),
#       file = paste0("ggmcmc-", "mu_Theta", "-", M.lab, ".pdf"),
#       plot = c("traceplot", "crosscorrelation", "caterpillar"))
```

```
L.sigma.theta ← plab("sigma.theta", list(Covariate = covariate.label))
L.sigma.delta ← plab("sigma.delta", list(Covariate = c("Interdependence (Borders)",
                                         "Interdependence (Trade)")))
L.sigma.theta ← bind_rows(L.sigma.theta, L.sigma.delta)
S.sigma.theta ← ggs(s, family = "^sigma.theta\\[|^sigma.delta\\[", par_labels = L.sigma.theta)
ggs_caterpillar(S.sigma.theta) +
  ggtitle("Between outcome standard deviations")
#  geom_vline(xintercept = 0, lty = 3) +

L.theta ← plab("theta", list(Outcome = outcome.label,
                              Covariate = covariate.label))
#L.delta ← plab("delta", list(Outcome = outcome.label,
#                              Covariate = c("Interdependence (Borders)",
#                                            "Interdependence (Trade)")))
#L.theta ← bind_rows(L.theta, L.delta)
S.theta ← ggs(s, family = "^theta\\[|^delta\\[", par_labels = L.theta) %>%
  mutate(Model = M) %>%
  filter(Covariate ≠ "(Intercept)") %>%
  mutate(Covariate = factor(Covariate, rev(covariate.label.order)))

save(S.theta, file = paste("samples-theta-", M.lab, ".RData", sep = ""))

ggs_caterpillar(S.theta, label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Outcome)


S.theta %>%
  filter(Outcome %in% c("Carbon pricing", "Carbon pricing (no EU)")) %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Outcome)


S.theta %>%
  filter(Outcome %in% c("Carbon pricing")) %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)


ggs_caterpillar(S.theta, label = "Outcome", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_wrap(~ Covariate)
```

Variance-covariance matrices

```
L.Sigma.Omega ← plab("Sigma", list(
                              Outcome = outcome.label,
                              Covariate.1 = covariate.label,
                              Covariate.2 = covariate.label))
S.Sigma.Omega ← ggs(s, family = "^Sigma\\[", par_labels = L.Sigma.Omega)
```

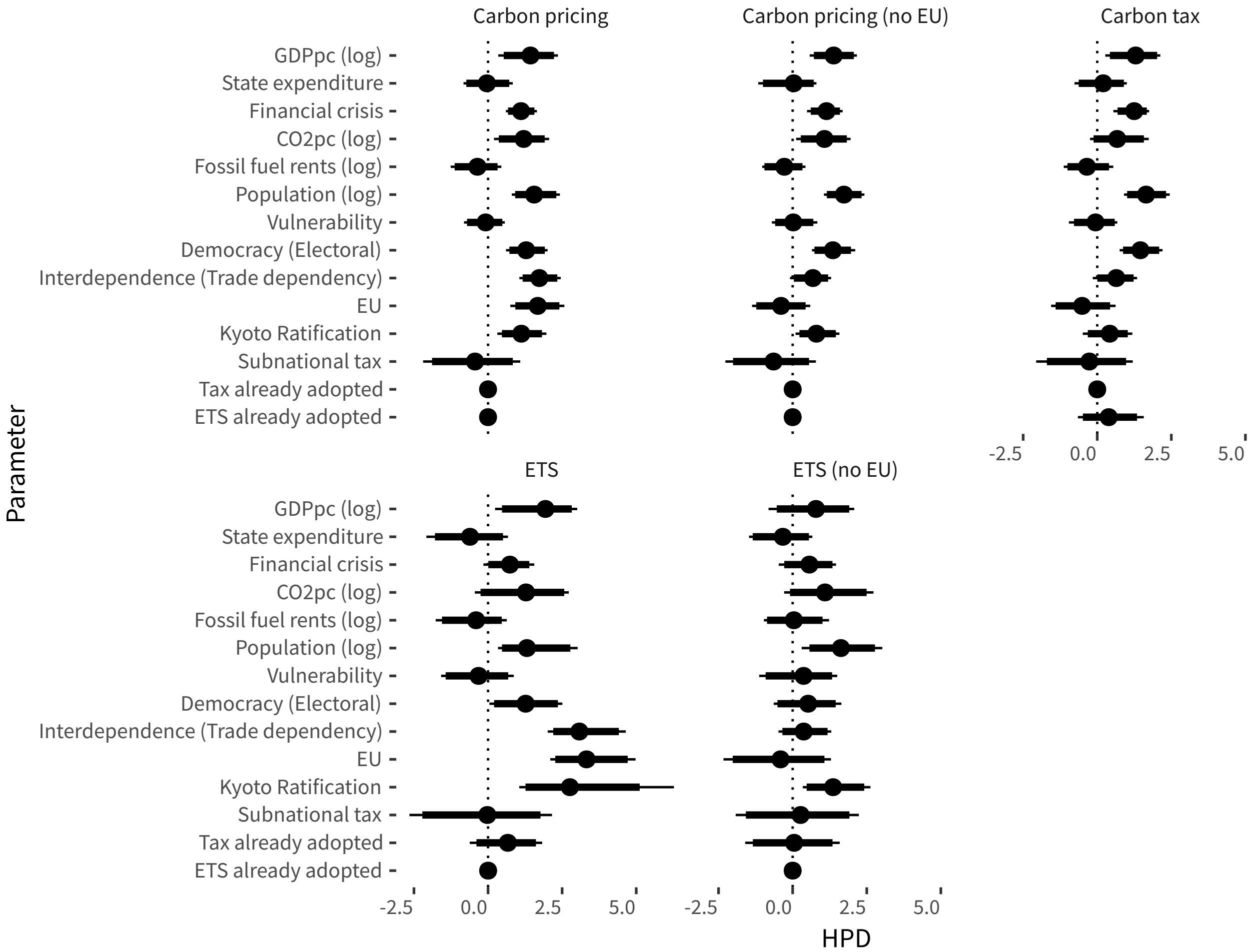


Figure 6.1: HPD of the effects of covariates on the likelihood of the event, by outcome.

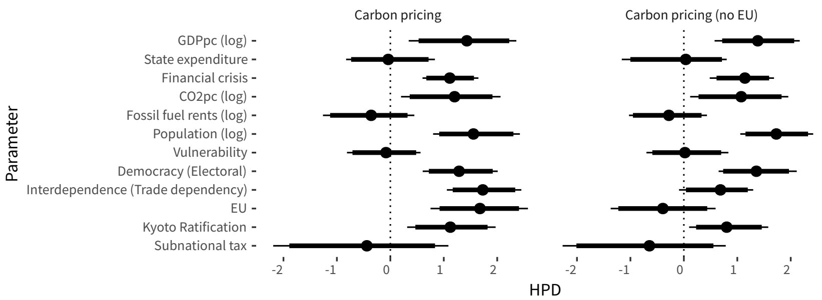


Figure 6.2: HPD of the effects of covariates on the likelihood of the event, by outcome. Only carbon pricing, with and without EU.

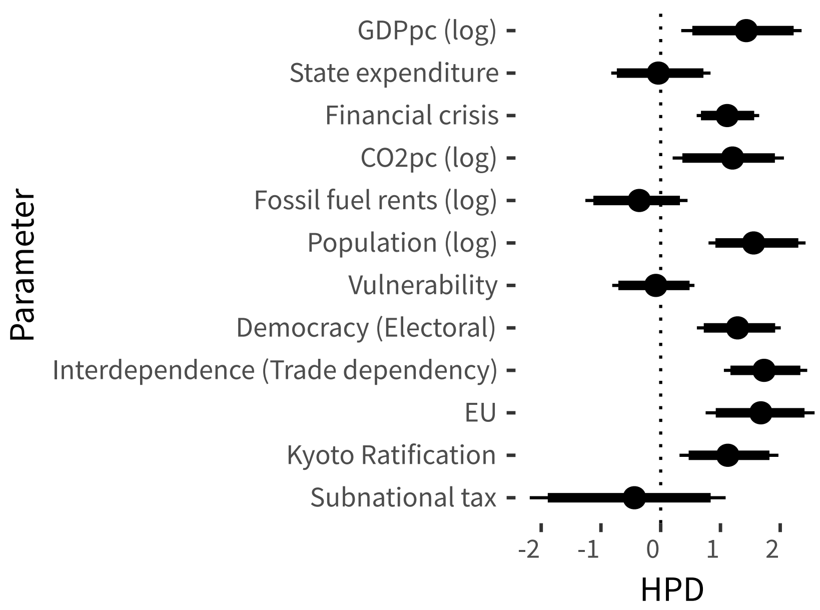


Figure 6.3: HPD of the effects of covariates on the likelihood of the event, by outcome. Only carbon pricing with the EU.

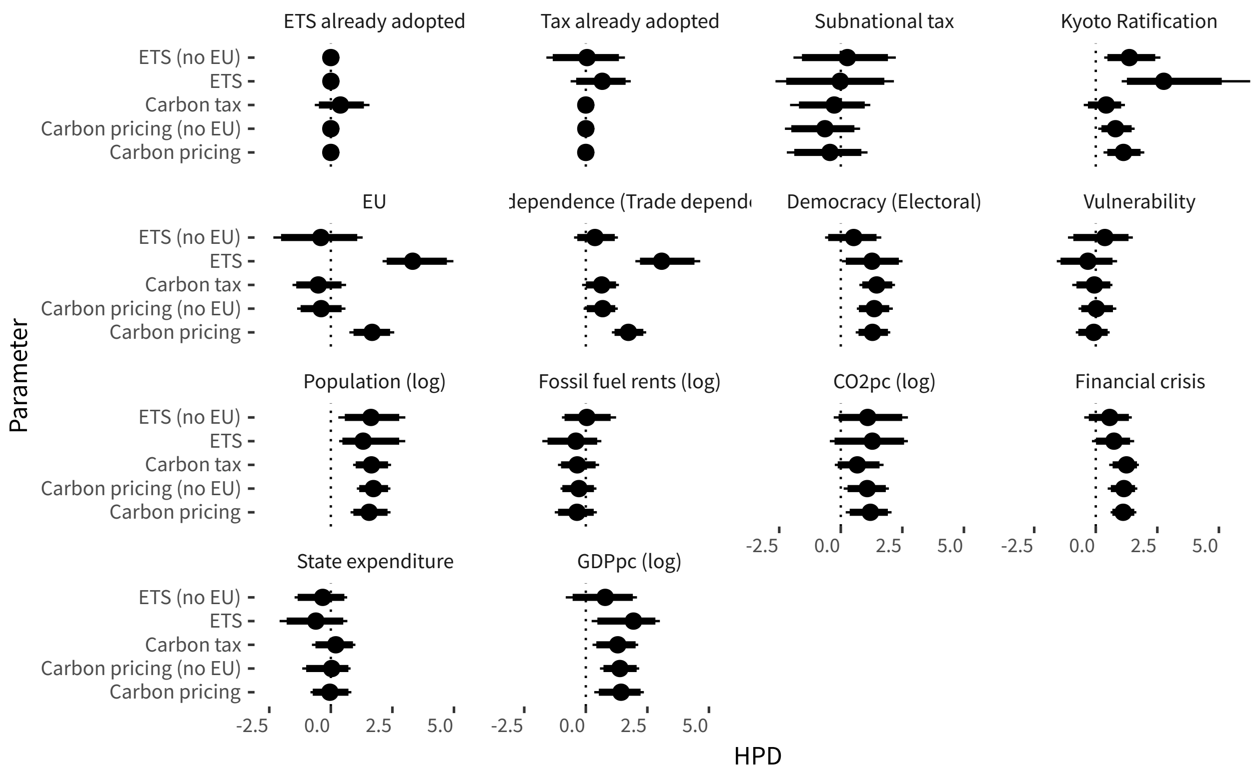


Figure 6.4: HPD of the effects of covariates on the likelihood of the event, by Covariate.

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

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

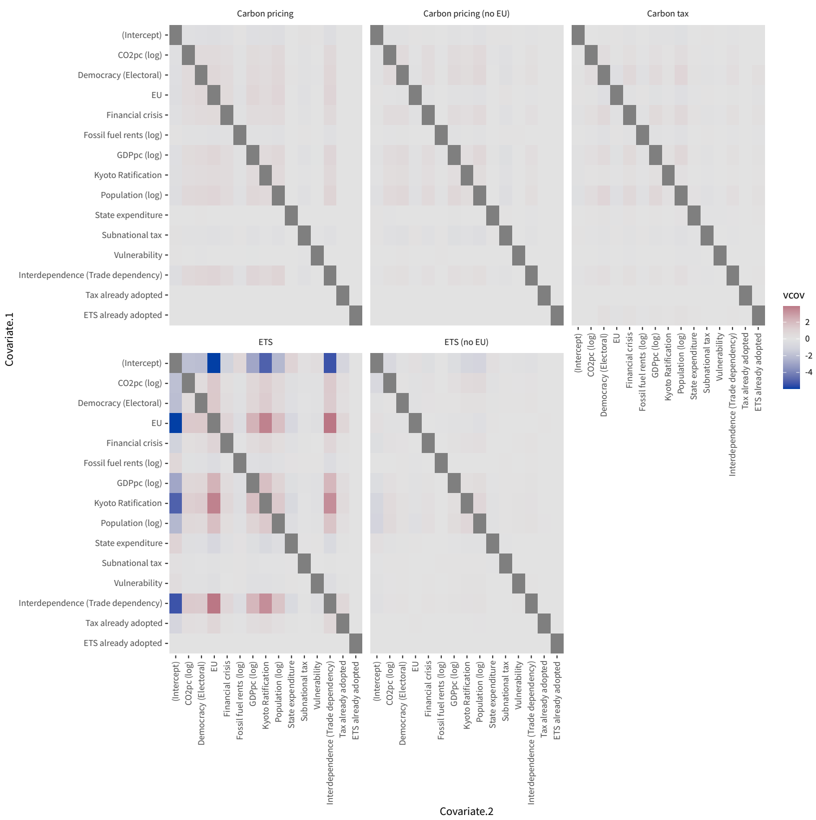


Figure 6.5: Variance-covariance matrix of main effects.

```
or <- function(x, significant = 2) {
  or <- as.character(signif((x - 1) * 100, significant))
  or[or < 0] <- paste0("\U25Bd ", str_replace(or[or < 0], "-", ""), "%")
  or[or > 0] <- paste0("\U25B3 ", or[or > 0], "%")
  or[or == 0] <- "="
  return(or)
}

tb <- S.theta %>%
  filter(Covariate != "(Intercept)") %>%
  ci() %>%
  arrange(Outcome, desc(abs(median))) %>%
  mutate(`Odds Ratio` = exp(median)) %>%
  mutate(`Expected effect` = or(`Odds Ratio`)) %>%
```

```
  select(Outcome, Covariate, `Odds Ratio`, `Expected effect`)

tc ← "Odds ratios of expected effect sizes, and sorted by magnitude and outcome."
if (knitr::is_latex_output()) {
  kable(tb, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tb, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 6.1: Odds ratios of expected effect sizes, and sorted by magnitude and outcome.

Outcome	Covariate	Odds Ratio	Expected effect
Carbon pricing	Interdependence (Trade dependency)	5.6400	△ 460%
Carbon pricing	EU	5.3480	△ 430%
Carbon pricing	Population (log)	4.7354	△ 370%
Carbon pricing	GDPpc (log)	4.1865	△ 320%
Carbon pricing	Democracy (Electoral)	3.6202	△ 260%
Carbon pricing	CO2pc (log)	3.3268	△ 230%
Carbon pricing	Kyoto Ratification	3.0737	△ 210%
Carbon pricing	Financial crisis	3.0404	△ 200%
Carbon pricing	Subnational tax	0.6443	▽ 36%
Carbon pricing	Fossil fuel rents (log)	0.6986	▽ 30%
Carbon pricing	Vulnerability	0.9221	▽ 7.8%
Carbon pricing	State expenditure	0.9616	▽ 3.8%
Carbon pricing	Tax already adopted	0.9994	▽ 0.064%
Carbon pricing	ETS already adopted	0.9997	▽ 0.031%
Carbon pricing (no EU)	Population (log)	5.6399	△ 460%
Carbon pricing (no EU)	GDPpc (log)	3.9886	△ 300%
Carbon pricing (no EU)	Democracy (Electoral)	3.8845	△ 290%
Carbon pricing (no EU)	Financial crisis	3.1345	△ 210%
Carbon pricing (no EU)	CO2pc (log)	2.9257	△ 190%
Carbon pricing (no EU)	Kyoto Ratification	2.2284	△ 120%
Carbon pricing (no EU)	Interdependence (Trade dependency)	1.9829	△ 98%
Carbon pricing (no EU)	Subnational tax	0.5251	▽ 47%
Carbon pricing (no EU)	EU	0.6752	▽ 32%
Carbon pricing (no EU)	Fossil fuel rents (log)	0.7558	▽ 24%
Carbon pricing (no EU)	State expenditure	1.0364	△ 3.6%
Carbon pricing (no EU)	Vulnerability	1.0205	△ 2%
Carbon pricing (no EU)	Tax already adopted	1.0006	△ 0.062%
Carbon pricing (no EU)	ETS already adopted	1.0000	△ 0.0011%
Carbon tax	Population (log)	5.2001	△ 420%
Carbon tax	Democracy (Electoral)	4.2931	△ 330%
Carbon tax	GDPpc (log)	3.6608	△ 270%
Carbon tax	Financial crisis	3.4799	△ 250%
Carbon tax	CO2pc (log)	1.9576	△ 96%
Carbon tax	Interdependence (Trade dependency)	1.9016	△ 90%
Carbon tax	EU	0.5999	▽ 40%
Carbon tax	Kyoto Ratification	1.5260	△ 53%
Carbon tax	ETS already adopted	1.4739	△ 47%
Carbon tax	Fossil fuel rents (log)	0.7062	▽ 29%
Carbon tax	Subnational tax	0.7666	▽ 23%
Carbon tax	State expenditure	1.2233	△ 22%
Carbon tax	Vulnerability	0.9445	▽ 5.6%

Carbon tax	Tax already adopted	0.9998	▽ 0.016%
ETS	EU	27.7476	△ 2700%
ETS	Interdependence (Trade dependency)	21.7697	△ 2100%
ETS	Kyoto Ratification	15.8179	△ 1500%
ETS	GDPpc (log)	6.9262	△ 590%
ETS	Population (log)	3.7114	△ 270%
ETS	CO2pc (log)	3.5996	△ 260%
ETS	Democracy (Electoral)	3.5548	△ 260%
ETS	Financial crisis	2.0906	△ 110%
ETS	Tax already adopted	1.9491	△ 95%
ETS	State expenditure	0.5439	▽ 46%
ETS	Fossil fuel rents (log)	0.6674	▽ 33%
ETS	Vulnerability	0.7260	▽ 27%
ETS	Subnational tax	0.9754	▽ 2.5%
ETS	ETS already adopted	1.0021	△ 0.21%
ETS (no EU)	Population (log)	5.0953	△ 410%
ETS (no EU)	Kyoto Ratification	3.9127	△ 290%
ETS (no EU)	CO2pc (log)	2.9793	△ 200%
ETS (no EU)	GDPpc (log)	2.1962	△ 120%
ETS (no EU)	Financial crisis	1.7592	△ 76%
ETS (no EU)	Democracy (Electoral)	1.6880	△ 69%
ETS (no EU)	EU	0.6655	▽ 33%
ETS (no EU)	Interdependence (Trade dependency)	1.4504	△ 45%
ETS (no EU)	Vulnerability	1.4474	△ 45%
ETS (no EU)	State expenditure	0.7185	▽ 28%
ETS (no EU)	Subnational tax	1.3070	△ 31%
ETS (no EU)	Tax already adopted	1.0512	△ 5.1%
ETS (no EU)	Fossil fuel rents (log)	1.0419	△ 4.2%
ETS (no EU)	ETS already adopted	1.0003	△ 0.033%

```
require(xtable)
tb.dump <- tb %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("Tax already adopted", "ETS already adopted")) %>%
  select(-Outcome)
tc <- "Odds ratios of expected effect sizes, by magnitude."

print(xtable(tb.dump,
             caption = tc,
             label = "tab:ees",
             tabular.environment = "longtable"),
      file = "table_expected_effect_sizes-003.tex",
      #size = "footnotesize",
      size = "normalsize",
      include.rownames = FALSE)

S.theta %>%
  filter(Outcome %in% c("Carbon tax", "ETS")) %>%
  ci() %>%
  ggplot(aes(ymin = low, ymax = high,
             y = median, x = Covariate,
             color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.3)) +
```

```
  geom_linerange(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.3)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Harmonic")
```

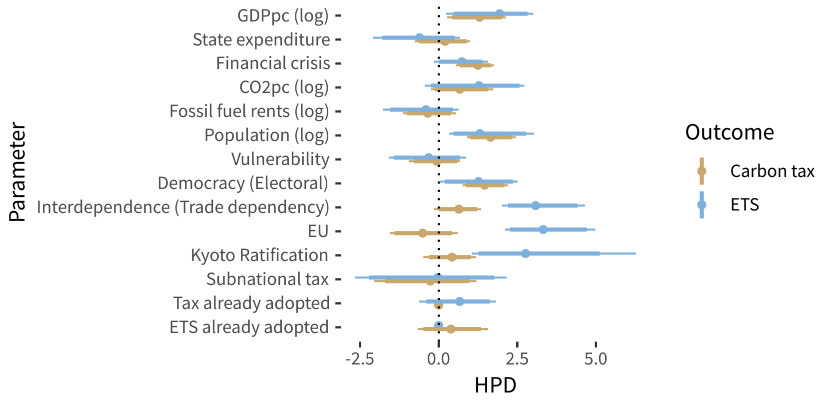


Figure 6.6: HPD of the effects of covariates on the likelihood of the event, by outcome. Only Tax and ETS.

```
L.alpha <- plab("alpha", list(Outcome = outcome.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_labels = L.alpha) %>%
  mutate(value = inv.logit(value))
ggs_caterpillar(S.alpha)

L.alpha <- plab("alpha", list(Year = time.label, Outcome = outcome.label))
S.alpha <- ggs(s, family = "^alpha\\[", par_labels = L.alpha) %>%
  mutate(Year = as.numeric(as.character(Year))) %>%
  mutate(value = inv.logit(value))

ci(S.alpha, thick_ci = c(0.1, 0.9)) %>%
  ggplot(aes(x = Year, y = median, color = Outcome)) +
  geom_line() +
  ylab("Hazard")

ci(S.alpha, thick_ci = c(0.1, 0.9)) %>%
  ggplot(aes(x = Year, y = median, ymin = low, ymax = high,
             fill = Outcome, color = Outcome)) +
  geom_line() +
  geom_ribbon(alpha = 0.3) +
  facet_wrap(~ Outcome) +
  ylab("Hazard")
```

List the countries most likely to adopt each policy.

```
L.pi <- plab("pi", list(Year = time.label,
                        Country = country.label,
                        Outcome = outcome.label)) %>%
  mutate(Year = as.numeric(as.character(Year)))
S.pi <- ggs(s, family = "^pi\\[", par_labels = L.pi)

tpp <- S.pi %>%
```

```
  mutate(value = inv.logit(value)) %>%
  ci() %>%
  select(Country, Outcome, `Predicted probability` = median) %>%
  group_by(Outcome) %>%
  arrange(desc(`Predicted probability`)) %>%
  slice(1:5)
```

```
tc <- "Countries with higher posterior median predicted probabilities of adopting each policy
outcome. Top 5 by Outcome."
if (knitr::is_latex_output()) {
  kable(tpp, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 9)
} else {
  kable(tpp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 9, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 6.2: Countries with higher posterior median predicted probabilities of adopting each policy outcome. Top 5 by Outcome.

Country	Outcome	Predicted probability
Russia	Carbon pricing	0.5085
Qatar	Carbon pricing	0.5084
United States	Carbon pricing	0.5070
Brazil	Carbon pricing	0.5056
Saudi Arabia	Carbon pricing	0.5055
Germany	Carbon pricing (no EU)	0.5058
United States	Carbon pricing (no EU)	0.5046
Qatar	Carbon pricing (no EU)	0.5045
Brazil	Carbon pricing (no EU)	0.5044
Saudi Arabia	Carbon pricing (no EU)	0.5040
China	Carbon tax	0.5078
Germany	Carbon tax	0.5052
United States	Carbon tax	0.5048
Australia	Carbon tax	0.5046
South Korea	Carbon tax	0.5041
South Africa	ETS	0.5658
Singapore	ETS	0.5579
Japan	ETS	0.5072
Mongolia	ETS	0.5067
Turkmenistan	ETS	0.5044
South Africa	ETS (no EU)	0.5101
Singapore	ETS (no EU)	0.5063
Japan	ETS (no EU)	0.5036
India	ETS (no EU)	0.5016
United States	ETS (no EU)	0.5013

```
S.pi

# Manually calculate PCP
# as ggmcmc's ggs_pcp() is not ready for matrices as input for outcome
```

```
threshold <- da %>%
  group_by(Outcome) %>%
  summarize(Threshold = length(which(Event == 1)) / n())

S <- inner_join(S.pi, select(da, Country, Year, Outcome, Event)) %>%
  left_join(threshold) %>%
  mutate(Correct = if_else( (value < Threshold & Event == 0) |
                            (value > Threshold & Event == 1),
                          TRUE, FALSE)) %>%
  group_by(Outcome, Iteration, Chain) %>%
  summarize(PCP = length(which(Correct)) / n())


ggplot(S, aes(x = PCP)) +
  geom_histogram() +
  facet_grid(~ Outcome) +
  expand_limits(x = c(0, 1))

t.pcp <- S %>%
  group_by(Outcome) %>%
  summarize(`Median PCP` = mean(PCP))

tc <- "Posterior median percent correctly predicted, by Outcome."
if (knitr::is_latex_output()) {
  kable(tpp, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(tpp, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}

rm(S)
invisible(gc())

source("load_packages.R")
load("carbon_pricing-adoption.RData")
```

7
Compare models: trade specifications

```
d ← NULL
load("samples-theta-baseline001.RData")
d ← bind_rows(d, S.theta)
load("samples-theta-baseline003.RData")
d ← bind_rows(d, S.theta)
rm(S.theta)
invisible(gc())

d %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("Tax already adopted",
                           "ETS already adopted")) %>%
  mutate(Covariate = as.character(Covariate)) %>%
  mutate(Covariate = ifelse(Covariate == "Interdependence (Trade dependency)",
                            "Trade interdependence", Covariate)) %>%
  mutate(Covariate = ifelse(Covariate == "Interdependence (Trade competition)",
                            "Trade competition", Covariate)) %>%
  ggs_caterpillar(label = "Covariate", comparison = "Model",
                  comparison_separation = 0.4) +
  aes(color = Model) +
  scale_color_manual("Model",
                     values = c("black", "grey70"),
                     breaks = c("Baseline 001", "Baseline 003"),
                     labels = c("Reference", "Alternative")) +
  theme(legend.position = "right")

setwd("patterns/")
```

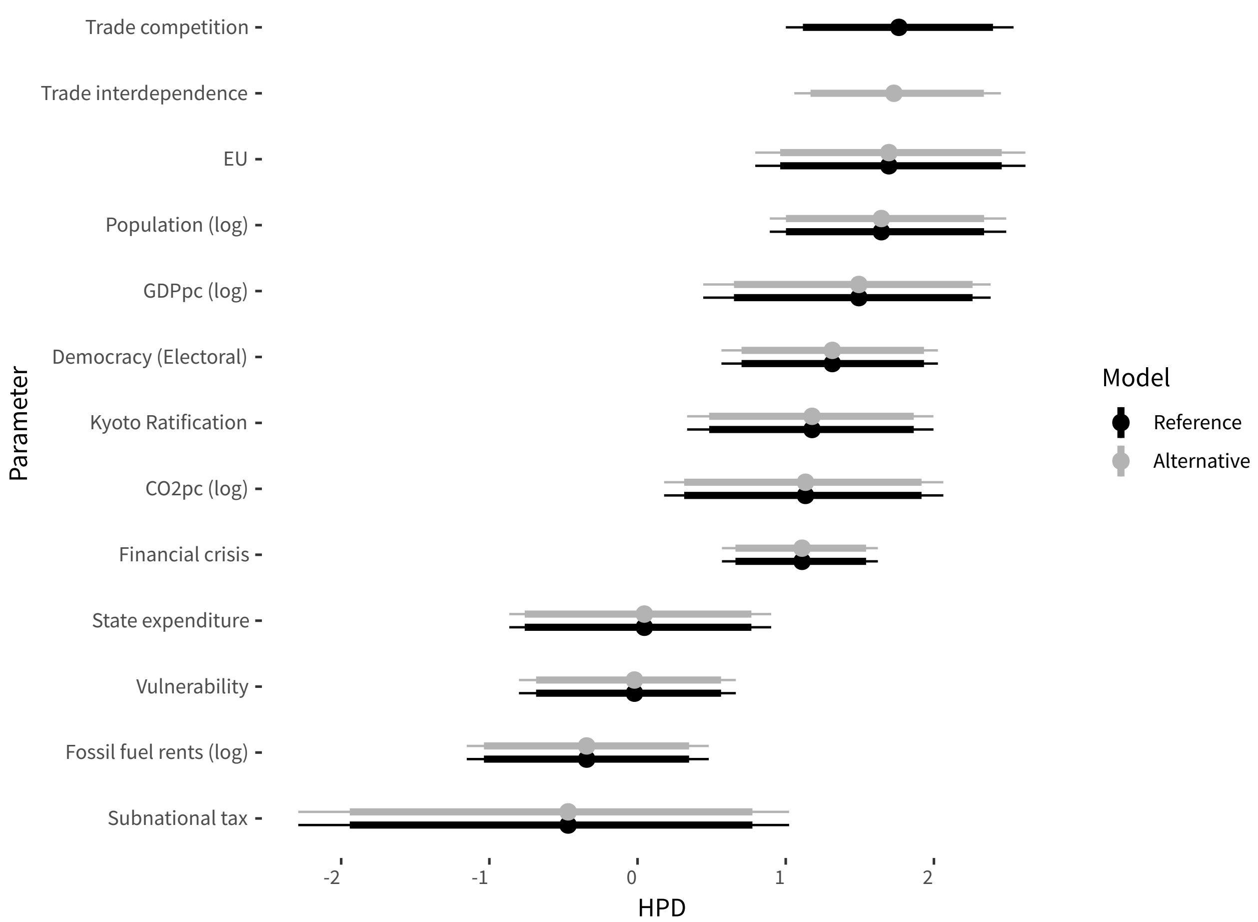


Figure 7.1: Model comparison using different trade effects specifications.

8

Data cleaning and preparation

```r
orig <- read.csv("carbon_pricing-patterns.csv", check.names = FALSE) %>%
  as_tibble()
```

Work with configurations of policies, not with concrete countries.

The main adoptions (Tax and ETS) are hierarchically more important than the concrete levels of the policy adoption. Therefore, for the concrete policy configurations (continuous variables) their value is `NA` when the policy has not yet even being adopted. This way, we are adding into the model the fact that we are uncertain about their values. Do not use the information (zeros) of such cases to inform about the overall model behaviour.

```r
# Trust in ../adoption for the time span and country coverage
load("../adoption/details.RData")

# Clean names and stuff, but still leave original variables
orig.cleaned <- orig %>%
  select(-c(iso2c, iso3c, WorldBank, Notes)) %>%
  gather(Variable, value, -Country) %>%
  mutate(value = as.numeric(value)) %>%
  mutate(Variable = str_replace_all(Variable, "_", " ")) %>%
  spread(Variable, value)

# Variables are now binary when appropriate
binary.variables <- c("ETS Adoption", "Tax Adoption", "Cap")
continuous.variables.full <- c("ETS Coverage", "ETS Price", "ETS Revenue Use",
                          "Tax Coverage", "Tax Price", "Tax Revenue Use",
                          "Cap reduction")
continuous.variables <- c("ETS Coverage", "ETS Price",
                          "Tax Coverage", "Tax Price", "Cap reduction")

d <- orig.cleaned %>%
  gather(Variable, value, -Country) %>%
  mutate(value = if_else( (Variable %in% binary.variables) &
                          (!is.na(value)), 1, value)) %>%
  mutate(value = if_else(is.na(value), 0, value)) %>%
  spread(Variable, value)

# Add a combination of both ETS _AND_ Taxes
d <- d %>%
```

```
  mutate(`ETS & Tax Adoption` = `ETS Adoption` * `Tax Adoption`)
binary.variables <- c(binary.variables, "ETS & Tax Adoption")


# Make that continuous variables values' are NA in the case of not adoption
# avoiding being 0 and known
d <- d %>%
  mutate(`ETS Coverage` = ifelse(`ETS Adoption` == 0, NA, `ETS Coverage`)) %>%
  mutate(`ETS Price` = ifelse(`ETS Adoption` == 0, NA, `ETS Price`)) %>%
  mutate(`ETS Revenue Use` = ifelse(`ETS Adoption` == 0, NA, `ETS Revenue Use`)) %>%
  mutate(`Tax Coverage` = ifelse(`Tax Adoption` == 0, NA, `Tax Coverage`)) %>%
  mutate(`Tax Price` = ifelse(`Tax Adoption` == 0, NA, `Tax Price`)) %>%
  mutate(`Tax Revenue Use` = ifelse(`Tax Adoption` == 0, NA, `Tax Revenue Use`)) %>%
  mutate(`Cap reduction`= ifelse(Cap == 0, NA, `Cap reduction`))
```

Extract only the different country configurations and assign them to each country.

```
configurations <- d %>%
  select(-Country) %>%
  select(all_of(continuous.variables), all_of(binary.variables)) %>%
  distinct() %>%
  mutate(Configuration = 1:n()) %>%
  gather(Variable, value, -Configuration) %>%
  spread(Variable, value)

possible.configurations <- d %>%
  left_join(configurations)

#configurations.pricecoverage <-

all.configs <- d %>%
  mutate(ep = ifelse(is.na(`ETS Price`), 0, `ETS Price`)) %>%
  mutate(ec = ifelse(is.na(`ETS Coverage`), 0, `ETS Coverage`)) %>%
  mutate(tp = ifelse(is.na(`Tax Price`), 0, `Tax Price`)) %>%
  mutate(tc = ifelse(is.na(`Tax Coverage`), 0, `Tax Coverage`)) %>%
#  gather(Variable, value, -Country) %>%
#  filter(str_detect(Variable, "Price|Coverage")) %>%
#  mutate(value = ifelse(is.na(value), 0, value)) %>%
#  spread(Variable, value) %>%
#  mutate(ETS.pc = `ETS Coverage` * `ETS Price`) %>%
#  mutate(Tax.pc = `Tax Coverage` * `Tax Price`) %>%
  mutate(ETS.pc = ep * ec) %>%
  mutate(Tax.pc = tp * tc)

configurations.pricecoverage <- all.configs %>%
#  select(-Country) %>%
  select(ETS.pc, Tax.pc) %>%
  distinct() %>%
  mutate(Configuration = 1:n()) #%>%

#  gather(Variable, value, -Configuration) %>%
#  spread(Variable, value) %>%
```

```
#  filter(Configuration == 14)

#possible.configurations.pricecoverage <- d %>%
#  left_join(configurations.pricecoverage)
possible.configurations.pricecoverage <- configurations.pricecoverage %>%
  left_join(all.configs) %>%
  left_join(configurations.pricecoverage)

save(d,
     binary.variables, continuous.variables, continuous.variables.full,
     configurations, possible.configurations,
     configurations.pricecoverage, possible.configurations.pricecoverage,
     file = "carbon_pricing-patterns.RData")
```

9
Data description

```
load("carbon_pricing-patterns.RData")
```

Number of countries

```
dim(d)[1]
```

```
→ [1] 215
```

Number of configurations

```
dim(configurations)[1]
```

```
→ [1] 35
```

List of configurations with more than one country.

```
t.conf ← possible.configurations %>%
  group_by(Configuration) %>%
  summarize(N = n()) %>%
  filter(N > 1) %>%
  arrange(desc(N))

tc ← "Number of countries in each configuration, for the configurations with more than one country."
if (knitr::is_latex_output()) {
  kable(t.conf, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE)
} else {
  kable(t.conf, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(bootstrap_options = "striped", full_width = FALSE)
}
```

Table 9.1: Number of countries in each configuration, for the configurations with more than one country.

Configuration	N
1	164
5	15
2	3
4	2

Countries sharing configurations.

```
t.conf.countries <- possible.configurations %>%
  select(Country, Configuration) %>%
  filter(Configuration %in% t.conf$Configuration & Configuration != 1) %>%
  arrange(Configuration)

tc <- "Countries that share configurations."
if (knitr::is_latex_output()) {
  kable(t.conf.countries, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE)
} else {
  kable(t.conf.countries, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(bootstrap_options = "striped", full_width = FALSE)
}
```

Table 9.2: Countries that share configurations.

Country	Configuration
Albania	2
Zambia	2
Zimbabwe	2
Australia	4
Kazakhstan	4
Austria	5
Belgium	5
Bulgaria	5
Cyprus	5
Czechia	5
Germany	5
Greece	5
Hungary	5
Italy	5
Lithuania	5
Luxembourg	5
Malta	5
Netherlands	5
Romania	5
Slovakia	5

Configurations' Correlations (Figure 9.1).

```
library(corrplot)
cc <- configurations %>%
  select(-Configuration) %>%
  as.matrix()

corrplot(cor(cc), use = "complete.obs", method = "ellipse")
```

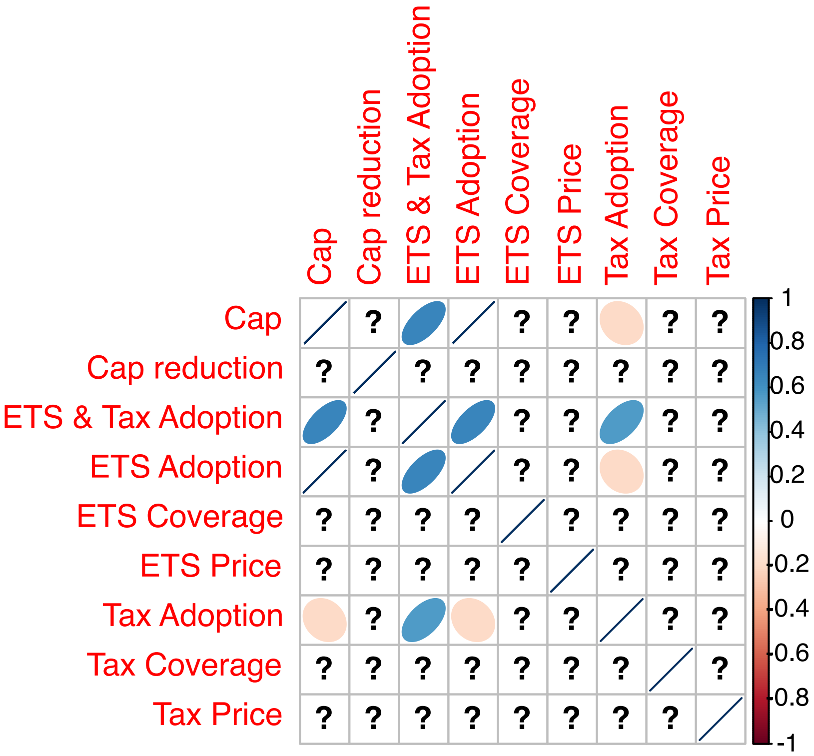


Figure 9.1: Correlations between variables, for configurations.

Configurations' Distribution of values by variable (Figure 9.2).

```
configurations %>%
  select(-Configuration) %>%
  gather(Variable, value) %>%
ggplot(aes(x = value)) +
  geom_histogram() +
  facet_wrap(~ Variable, ncol = 4, scales = "free")
```

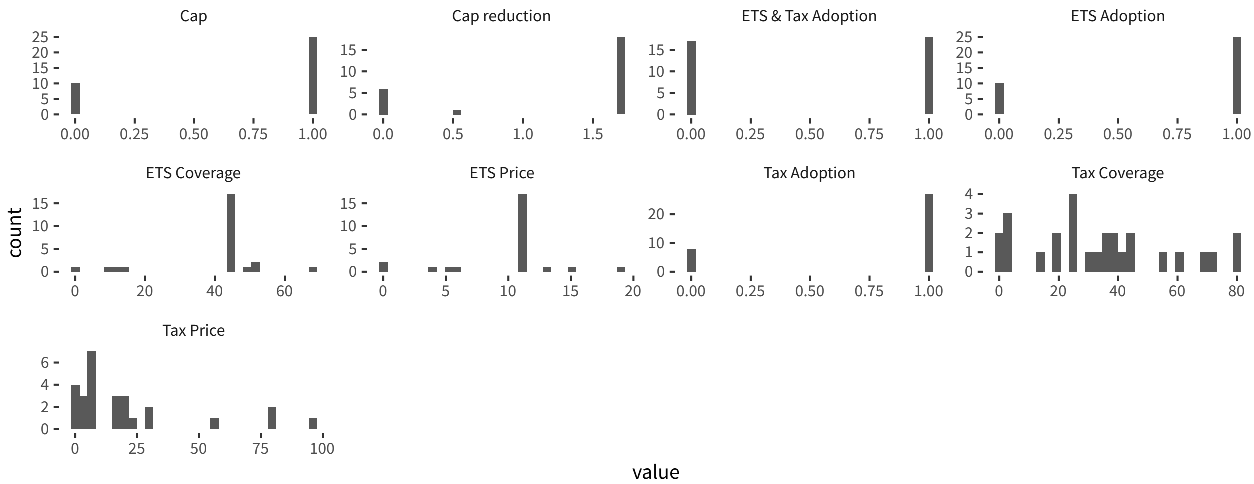


Figure 9.2: Distribution of values by variable, for the observations based on configurations.

10
Cluster

Avoid working with the full set of countries, and instead merge all the non-policy adopters into a single point

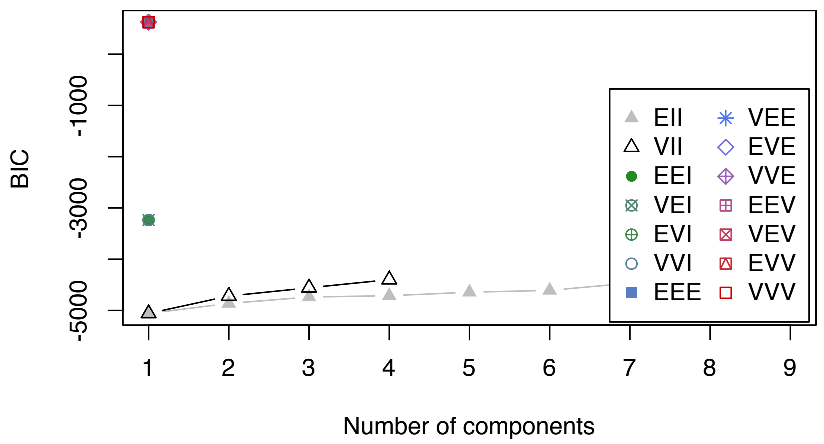
```r
countries.nopolicy <- d %>%
  gather(Variable, value, -Country) %>%
  group_by(Country) %>%
  summarize(All = sum(value, na.rm = TRUE)) %>%
  filter(All == 0) %>%
  select(-All)

d.country <- d %>%
  filter(!Country %in% countries.nopolicy$Country) %>%
  bind_rows(filter(d, Country == countries.nopolicy$Country[1])) %>%
  mutate(Country = if_else(Country == countries.nopolicy$Country[1], "NO POLICY", Country))

# Otherwise simply use d as d.country
# d.country <- d

# Specify the matrix with the values to perform cluster analysis on
X <- as.matrix(select(d.country, -Country),
               dimnames = list(Country = d.country$Country, var = names(d.country)[-1]))
X.no.na <- X
X.no.na[is.na(X)] <- 0

library(mclust)
cl.bic <- mclustBIC(X.no.na)
plot(cl.bic)
```



Generate a dendrogram 10.1 with all the cases, having used all variables (binary and continuous) with Gower distances.

```
X.dist ← daisy(X, metric = "gower")
hc ← hclust(X.dist)
hc$labels ← as.character(d.country$Country)
ggdendrogram(hc, size = 1)
```

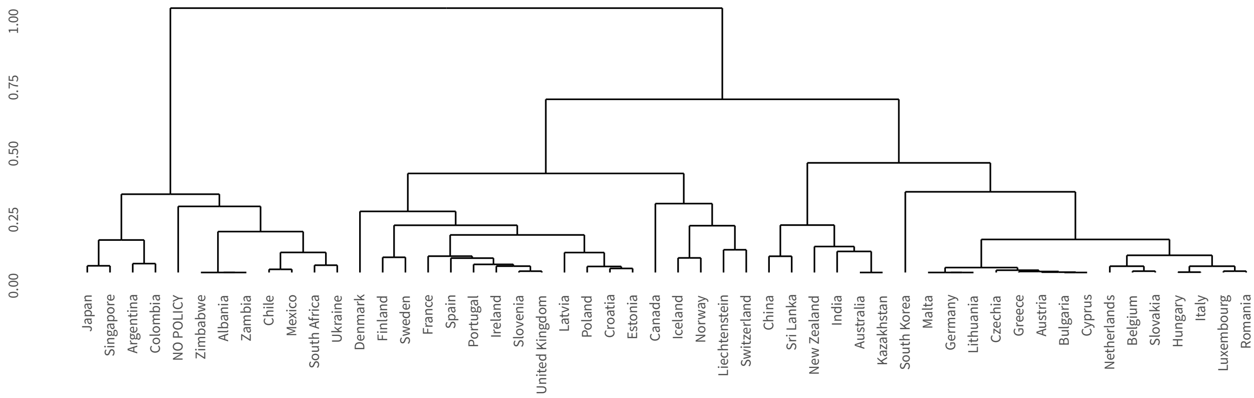


Figure 10.1: Dendrogram with gower distances.

We cut the dendrogram in a specific number of clusters

```
(cut.clusters ← 3)
```

```
→ [1] 3
```

Number and proportion of cases in each cluster.

```
ct ← cutree(hc, cut.clusters)
class.id ← bind_cols(d.country, tbl_df(data.frame(Cluster = as.factor(ct))))
table(class.id$Cluster)
```

```
→
→  1  2  3
→ 12 22 18
```

```
prop.table(table(class.id$Cluster))
```

```
→
→      1      2      3
→ 0.2308 0.4231 0.3462
```

```
# Function for alpha channel in densities at the diagonal
my_dens ← function(data, mapping, ..., low = "#132B43", high = "#56B1F7") {
  ggplot(data = data, mapping=mapping) +
         geom_density(..., alpha=0.3)
}
```

Descriptive Figure (10.2) with the distribution of values of each variable in the clusters.

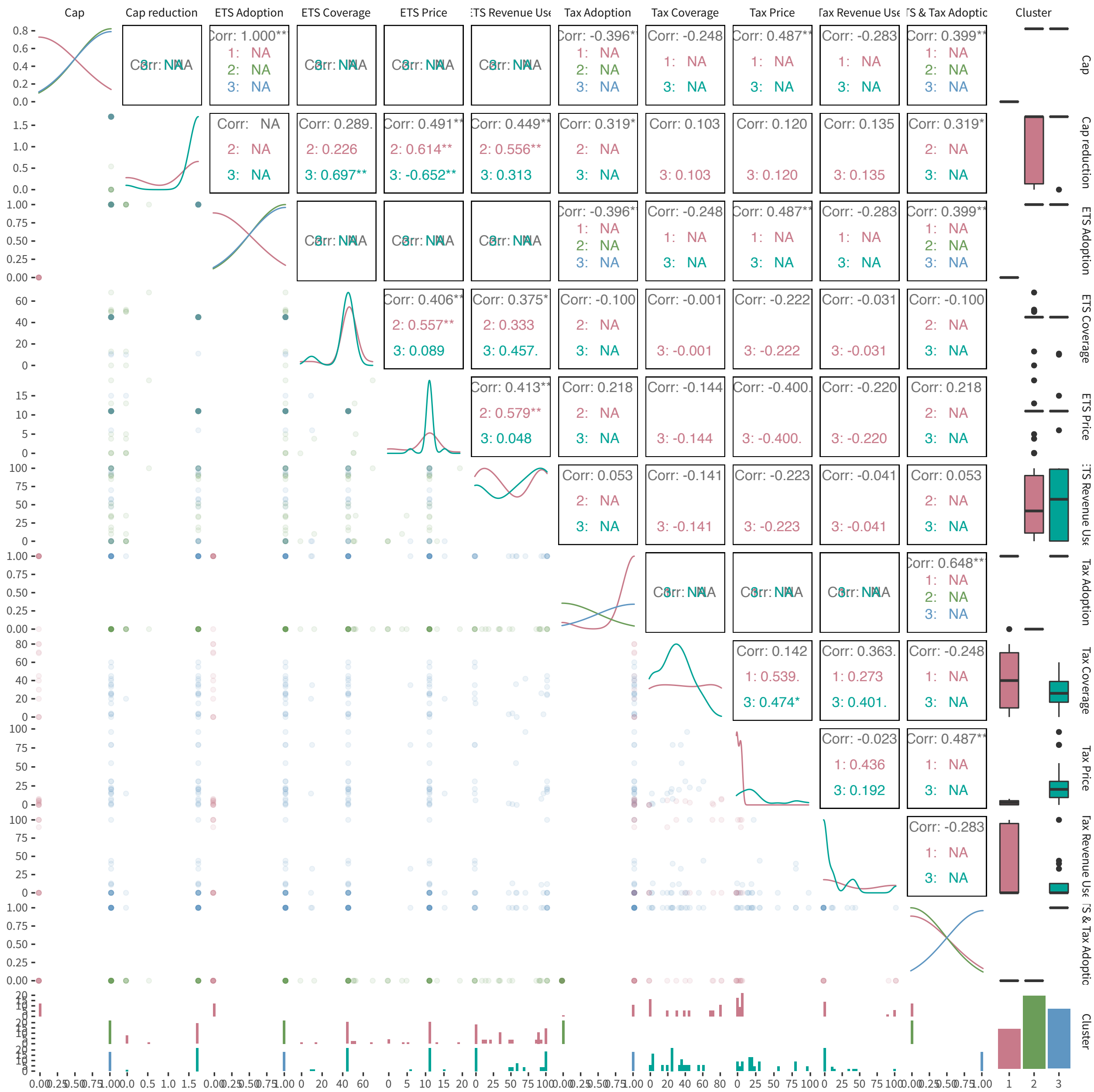


Figure 10.2: Classification of countries within clusters.

```
ggpairs(select(class.id, -Country),
  lower = list(continuous = wrap("points", alpha = 0.1)),
  diag = list(continuous = my_dens),
  mapping = aes(color = Cluster))
```

List of countries and the Cluster they belong to.

```
t.clusters <- class.id %>%
  select(Cluster, Country) %>%
  arrange(Cluster, Country)


tc <- "Clusters and countries."
if (knitr::is_latex_output()) {
  kable(t.clusters, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 8)
} else {
  kable(t.clusters, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 8, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 10.1: Clusters and countries.

Cluster	Country
1	Albania
1	Argentina
1	Chile
1	Colombia
1	Japan
1	Mexico
1	NO POLICY
1	Singapore
1	South Africa
1	Ukraine
1	Zambia
1	Zimbabwe
2	Australia
2	Austria
2	Belgium
2	Bulgaria
2	China
2	Cyprus
2	Czechia
2	Germany
2	Greece
2	Hungary
2	India
2	Italy
2	Kazakhstan
2	Lithuania
2	Luxembourg
2	Malta
2	Netherlands
2	New Zealand
2	Romania
2	Slovakia
2	South Korea
2	Sri Lanka

3	Canada
3	Croatia
3	Denmark
3	Estonia
3	Finland
3	France
3	Iceland
3	Ireland
3	Latvia
3	Liechtenstein
3	Norway
3	Poland
3	Portugal
3	Slovenia
3	Spain
3	Sweden
3	Switzerland
3	United Kingdom

11
Measurement model

```
# Model:
M
```

```
→ [1] "Baseline"
```

Data preparation:

- Standardization of continuous variables

Model description:

- Measurement model combining binary and continuous variables.
- The binary variables are treated using an Item-Response model.
- The continuous variables are treated using factor analysis.
- Based on Jordana, Fernández-i-Marín, and Bianculli (2018).
- A latent value of "carbon restriction" is assumed to generate a data that produces binary indicators for the adoption of certain policies and continuous indicators for the strength of such policies. We aim at extrating this latent value (ξ_c in our notation).
- We use the median of the posterior distribution of the latent value to represent the score of each country, following a geometrical loss function.

```
################## Select either to use configurations or countries
## Use countries
## Be aware that 'd' includes all sorts of variables, including variables not
## expected to be in the measurement model
#X.df <- d %>%
#  gather(Variable, value, -Country) %>%
#  group_by(Variable) %>%
#  mutate(value = if_else(Variable %in% continuous.variables, std(value), value)) %>%
#  ungroup() %>%
#  arrange(Country)
#X <- reshape2::acast(X.df, Country ~ Variable, value.var = "value")
#country.label <- dimnames(X)[[1]]


# Use configurations
# Be aware that configurations are already cleaned from variables not expected
# to be used in the measurement model
```

```
X.df ← configurations %>%
  # Configuration = 1 (no policy at all) is discarded
  # Beware that later, in the plab() function a +1 must be added
  filter(Configuration ≠ 1) %>%
  #
  #
  gather(Variable, value, -Configuration) %>%
  group_by(Variable) %>%
  mutate(value = if_else(Variable %in% continuous.variables, std(value), value)) %>%
#  mutate(value = if_else(Variable %in% continuous.variables, value / 100, value)) %>%
  ungroup() %>%
  arrange(Configuration)
X ← reshape2::acast(X.df, Configuration ~ Variable, value.var = "value")

#################

nO ← dim(X)[1] # Configurations / Countries


variable.label ← dimnames(X)[[2]]
position.binary ← which(variable.label %in% binary.variables)
position.continuous ← which(variable.label %in% continuous.variables)

X.binary ← X[,position.binary]
nBV ← dim(X.binary)[2]
X.continuous ← X[,position.continuous]
nCV ← dim(X.continuous)[2]


D ← list(
  nO = nO,
  nBV = nBV, X_binary = unname(X.binary),
  nCV = nCV, X_continuous = unname(X.continuous))

m ← 'model {
  for (o in 1:nO) {
    # -- Binary variables
    for (ib in 1:nBV) {
      X_binary[o,ib] ~ dbern(pi[o,ib])
      logit(pi[o,ib]) ← mu[o,ib]
      mu[o,ib] ← delta[ib,1] * (xi[o] - delta[ib,2])
    }
    for (ic in 1:nCV) {
      X_continuous[o,ic] ~ dnorm(mu.continuous[o,ic], tau.continuous[ic])
      mu.continuous[o,ic] ← gamma[ic,1] + (xi[o] * gamma[ic,2])
    }
  }
  # -- Priors for binary measurement
  for (ib in 1:nBV) {
#    delta[ib,1] ~ dnorm(0, 1^-2)T(0,)
#    delta[ib,1] ~ dnorm(1, 0.1^-2)T(0,)
    delta[ib,1] ~ dnorm(1, 0.5^-2)T(0,)
```

```
    delta[ib,2] ~ dnorm(0, 1^-2)
  }
  # -- Priors for continuous measurement
  for (ic in 1:nCV) {
    gamma[ic,1] ~ dnorm(0, 1^-2)
#    gamma[ic,2] ~ dnorm(0, 1^-2)T(0,)
#    gamma[ic,2] ~ dnorm(1, 0.1^-2)T(0,)
    gamma[ic,2] ~ dnorm(0, 0.5^-2)T(0,)
    tau.continuous[ic] ~ dt(0, 0.1^-2, 3)T(0,)
    sigma.continuous[ic] <- 1 / sqrt(tau.continuous[ic])
  }
  # -- Priors for scores of observations
  for (o in 1:nO) {
    xi[o] ~ dnorm(0, 1^-2)
  }
}'
write(m, file = paste0("models/model-", M.lab, ".bug"))

par <- NULL
par <- c(par, "gamma")
par <- c(par, "delta")
par <- c(par, "xi")

t0 <- proc.time()
rj <- run.jags(model = paste0("models/model-", M.lab, ".bug"),
               data = dump.format(D, checkvalid = FALSE),
               modules = c("glm", "lecuyer"),
               n.chains = chains, thin = thin,
               adapt = adapt, burnin = burnin, sample = sample,
               monitor = par, method = "parallel", summarise = FALSE)
s <- as.mcmc.list(rj)
save(s, file = paste0("sample-", M.lab, ".Rdata"))
proc.time() - t0

load(file = paste0("sample-", M.lab, ".Rdata"))

#ggmcmc(ggs(s, family = "gamma|delta"), plot = c("traceplot", "crosscorrelation", "caterpillar"))
ggmcmc(ggs(s, family = "xi|gamma|delta"), plot = c("traceplot", "crosscorrelation", "caterpillar"))

L.delta <- plab("delta", list(Variable = binary.variables,
                              Coefficient = c("Discrimination", "Difficulty")))
S.delta <- ggs(s, family = "delta", par_labels = L.delta) %>%
  filter(Coefficient == "Discrimination")
ggs_caterpillar(S.delta, label = "Variable")

L.gamma <- plab("gamma", list(Variable = continuous.variables,
                              Coefficient = c("Intercept", "Loading")))
S.gamma <- ggs(s, family = "gamma", par_labels = L.gamma) %>%
  filter(Coefficient == "Loading")
ggs_caterpillar(S.gamma, label = "Variable")

# Configurations
L.xi <- plab("xi", list(Configuration = (1:nO) + 1))
```

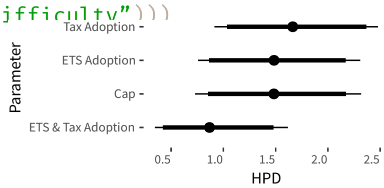


Figure 11.1: Discrimination parameters for binary variables.

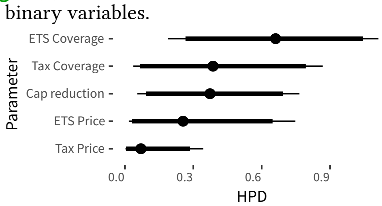


Figure 11.2: Loadings parameters for continuous variables.

```
S.xi <- ggs(s, family = "xi", par_labels = L.xi)
scores <- ci(S.xi) %>%
  select(Configuration, Score = median) %>%
  mutate(Configuration = as.numeric(as.character(Configuration))) %>%
  mutate(Model = M)

ggs_caterpillar(S.xi) + ylab("Configuration")
```

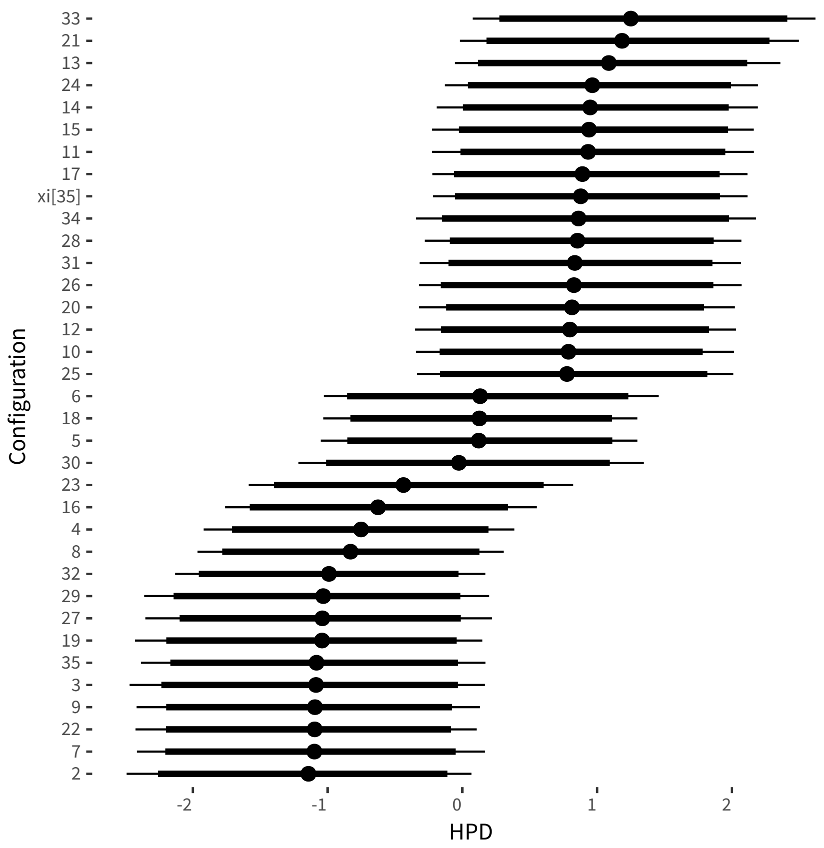


Figure 11.3: Caterpillar plot with the HPD of the configurations scores.

```
## Countries
#L.xi <- plab("xi", list(Country = country.label))
#S.xi <- ggs(s, family = "xi", par_labels = L.xi)
#scores <- ci(S.xi) %>%
#  select(Country , Score = median)
#save(scores, file = paste0("scores-", M.lab, ".Rdata"))
#
#ggs_caterpillar(S.xi) + ylab("Country")

# Configurations
scores.countries <- scores %>%
  left_join(select(possible.configurations, Country, Configuration))
ggplot(filter(scores.countries, Configuration != 1),
       aes(x = Score, y = reorder(Country, Score))) +
  geom_point() +
  ylab("Country")
```

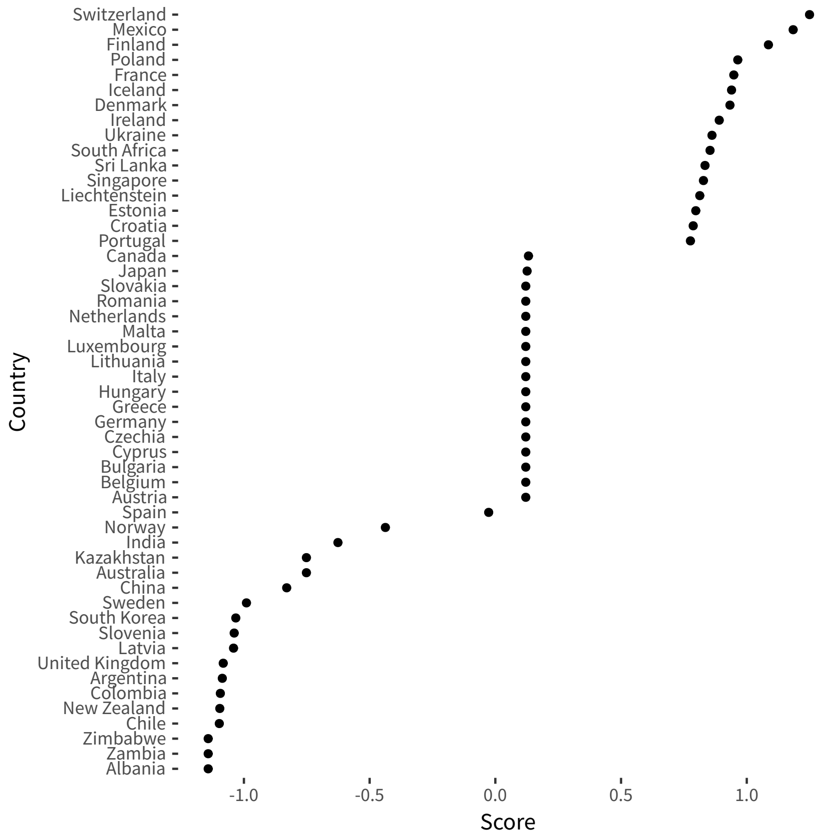


Figure 11.4: Scores of policy patterns for countries with a policy.

```
#scores.countries <- scores %>%
#  arrange(desc(Score))
#ggplot(slice(scores.countries, 1:60),
#       aes(x = Score, y = reorder(Country, Score))) +
#  geom_point() +
#  ylab("Country")
```

```
save(scores, scores.countries, file = paste0("scores-", M.lab, ".RData"))
```

12
Measurement model

```
# Model:
M
```

```
→ [1] "Baseline001"
```

Data preparation:

- Standardization of continuous variables

Model description:

- Measurement model combining binary and continuous variables.
- The binary variables are treated using an Item-Response model.
- The continuous variables are treated using factor analysis.
- Based on Jordana, Fernández-i-Marín, and Bianculli (2018).
- A latent value of "carbon restriction" is assumed to generate a data that produces binary indicators for the adoption of certain policies and continuous indicators for the strength of such policies. We aim at extrating this latent value (ξ_c in our notation).
- We use the median of the posterior distribution of the latent value to represent the score of each country, following a geometrical loss function.

```
################## Select either to use configurations or countries
## Use countries
## Be aware that 'd' includes all sorts of variables, including variables not
## expected to be in the measurement model
#X.df <- d %>%
#  gather(Variable, value, -Country) %>%
#  group_by(Variable) %>%
#  mutate(value = if_else(Variable %in% continuous.variables, std(value), value)) %>%
#  ungroup() %>%
#  arrange(Country)
#X <- reshape2::acast(X.df, Country ~ Variable, value.var = "value")
#country.label <- dimnames(X)[[1]]


# Use configurations
# Be aware that configurations are already cleaned from variables not expected
# to be used in the measurement model
```

```
X.df <-
  configurations.pricecoverage %>%
  # Configuration = 1 (no policy at all) is discarded
  # Beware that later, in the plab() function a +1 must be added
  filter(Configuration != 1) %>%
  #
  #
  gather(Variable, value, -Configuration) %>%
  # Select only price and coverage variables
#  filter(str_detect(Variable, "Price|Coverage")) %>%
  filter(str_detect(Variable, "\\.pc")) %>%
#  mutate(value = ifelse(is.na(value), 0, value)) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
#  mutate(value = if_else(Variable %in% continuous.variables, std(value), value)) %>%
#  mutate(value = if_else(Variable %in% continuous.variables, value / 100, value)) %>%
  ungroup() %>%
  arrange(Configuration)
X <- reshape2::acast(X.df, Configuration ~ Variable, value.var = "value")

#################

nO <- dim(X)[1] # Configurations / Countries


variable.label <- dimnames(X)[[2]]
position.binary <- which(variable.label %in% binary.variables)
position.continuous <- which(variable.label %in% continuous.variables)

X.binary <- X[,position.binary]
nBV <- dim(X.binary)[2]
X.continuous <- X[,position.continuous]
X.continuous <- X
nCV <- dim(X.continuous)[2]


D <- list(
  nO = nO,
#  nBV = nBV, X_binary = unname(X.binary),
  nCV = nCV, X_continuous = unname(X.continuous))

m <- 'model {
  for (o in 1:nO) {
#    # -- Binary variables
#    for (ib in 1:nBV) {
#      X_binary[o,ib] ~ dbern(pi[o,ib])
#      logit(pi[o,ib]) <- mu[o,ib]
#      mu[o,ib] <- delta[ib,1] * (xi[o] - delta[ib,2])
#    }
    for (ic in 1:nCV) {
      X_continuous[o,ic] ~ dnorm(mu.continuous[o,ic], tau.continuous[ic])
      mu.continuous[o,ic] <- gamma[ic,1] + (xi[o] * gamma[ic,2])
```

```
    }
  }
  # -- Priors for binary measurement
#   for (ib in 1:nBV) {
##     delta[ib,1] ~ dnorm(0, 1^-2)T(0,)
##     delta[ib,1] ~ dnorm(1, 0.1^-2)T(0,)
#     delta[ib,1] ~ dnorm(1, 0.5^-2)T(0,)
#     delta[ib,2] ~ dnorm(0, 1^-2)
#   }
  # -- Priors for continuous measurement
  for (ic in 1:nCV) {
    gamma[ic,1] ~ dnorm(0, 1^-2)
#     gamma[ic,2] ~ dnorm(0, 1^-2)T(0,)
#     gamma[ic,2] ~ dnorm(1, 0.1^-2)T(0,)
#     gamma[ic,2] ~ dnorm(0, 0.5^-2)T(0,)
    gamma[ic,2] ~ dnorm(0, 1^-2)T(0,)
    tau.continuous[ic] ~ dt(0, 0.1^-2, 3)T(0,)
    sigma.continuous[ic] ← 1 / sqrt(tau.continuous[ic])
  }
  # -- Priors for scores of observations
  for (o in 1:nO) {
    xi[o] ~ dnorm(0, 1^-2)
  }
}'
write(m, file = paste0("models/model-", M.lab, ".bug"))

par ← NULL
par ← c(par, "gamma")
#par ← c(par, "delta")
par ← c(par, "xi")

t0 ← proc.time()
rj ← run.jags(model = paste0("models/model-", M.lab, ".bug"),
              data = dump.format(D, checkvalid = FALSE),
              modules = c("glm", "lecuyer"),
              n.chains = chains, thin = thin,
              adapt = adapt, burnin = burnin, sample = sample,
              monitor = par, method = "parallel", summarise = FALSE)
s ← as.mcmc.list(rj)
save(s, file = paste0("sample-", M.lab, ".Rdata"))
proc.time() - t0

load(file = paste0("sample-", M.lab, ".Rdata"))

#ggmcmc(ggs(s, family = "gamma|delta"), plot = c("traceplot", "crosscorrelation", "caterpillar"))
ggmcmc(ggs(s, family = "xi|gamma|delta"), plot = c("traceplot", "crosscorrelation", "caterpillar"))

L.delta ← plab("delta", list(Variable = binary.variables,
                             Coefficient = c("Discrimination", "Difficulty")))
S.delta ← ggs(s, family = "delta", par_labels = L.delta) %>%
  filter(Coefficient == "Discrimination")
ggs_caterpillar(S.delta, label = "Variable")
```

```
L.gamma <- plab("gamma", list(Variable = continuous.variables,
                              Coefficient = c("Intercept", "Loading")))
S.gamma <- ggs(s, family = "gamma", par_labels = L.gamma) %>%
  filter(Coefficient == "Loading")
ggs_caterpillar(S.gamma, label = "Variable")
```

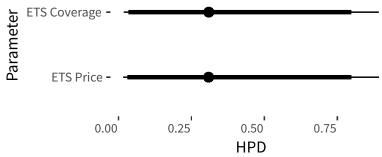


Figure 12.1: Loadings parameters for continuous variables.

```
# Configurations
L.xi <- plab("xi", list(Configuration = (1:nO) + 1))
S.xi <- ggs(s, family = "xi", par_labels = L.xi)
scores <- ci(S.xi) %>%
  select(Configuration, Score = median) %>%
  mutate(Configuration = as.numeric(as.character(Configuration))) %>%
  mutate(Model = M)

ggs_caterpillar(S.xi) + ylab("Configuration")
```

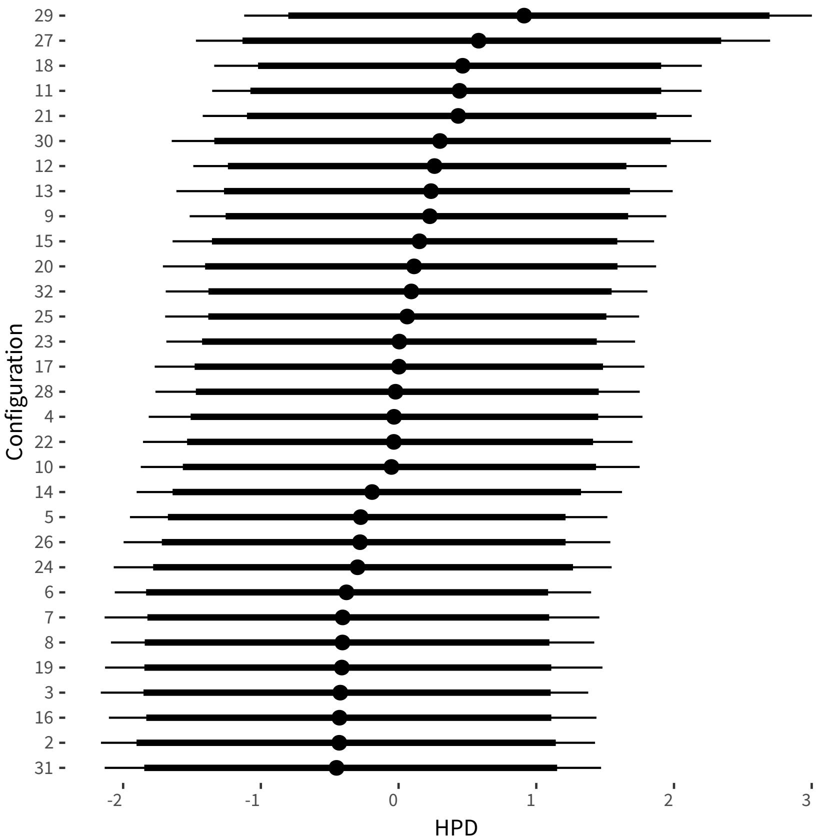


Figure 12.2: Caterpillar plot with the HPD of the configurations scores.

```
## Countries
#L.xi <- plab("xi", list(Country = country.label))
#S.xi <- ggs(s, family = "xi", par_labels = L.xi)
#scores <- ci(S.xi) %>%
#  select(Country , Score = median)
#save(scores, file = paste0("scores-", M.lab, ".Rdata"))
#
#ggs_caterpillar(S.xi) + ylab("Country")
```

```
# Just to check
configurations.pricecoverage %>%
  filter(Configuration %in% c(1, 2, 5, 33, 35))

X.df %>%
  spread(Variable, value) %>%
  filter(Configuration %in% c(1, 2, 5, 33, 35))

X[c(1, 5-1, 33-1, 35-1),]

X.df %>%
  group_by(Configuration) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  filter(Configuration %in% c(1, 2, 5, 33, 35))

X.df %>%
#  group_by(Configuration) %>%
#  summarize(value = mean(value, na.rm = TRUE)) %>%
  spread(Variable, value) %>%
  left_join(scores) %>%
  filter(Configuration %in% c(1, 5, 33, 35)) %>%
  select(-Model)

scores %>%
  filter(Configuration %in% c(1, 5, 33, 35)) %>%
  select(Configuration, Score)

X.df %>%
  group_by(Configuration) %>%
  summarize(Average = mean(value, na.rm = TRUE)) %>%
  left_join(scores) %>%
  ggplot(aes(x = Average, y = Score, label = Configuration)) +
  geom_text() +
  geom_text(data = . %>% filter(Configuration %in% c(5, 33)), color = "red") +
  geom_point(data = . %>% filter(Configuration %in% c(5, 33)), color = "blue")


d %>%
  filter(Country %in% c("Switzerland", "Slovakia")) %>%
  data.frame()

# Configurations
scores.countries <- scores %>%
  left_join(select(possible.configurations.pricecoverage, Country, Configuration))
ggplot(filter(scores.countries, Configuration != 1),
       aes(x = Score, y = reorder(Country, Score))) +
  geom_point() +
  ylab("Country")

#scores.countries <- scores %>%
#  arrange(desc(Score))
#ggplot(slice(scores.countries, 1:60),
```

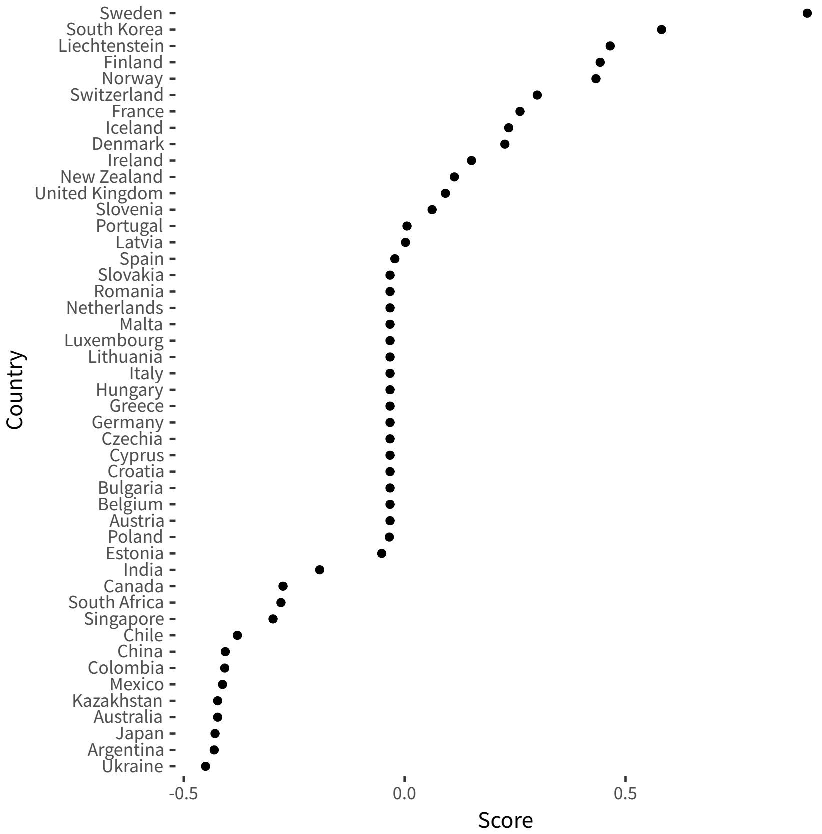


Figure 12.3: Scores of policy patterns for countries with a policy.

```
#       aes(x = Score, y = reorder(Country, Score))) +
#   geom_point() +
#   ylab("Country")

save(scores, scores.countries, file = paste0("scores-", M.lab, ".RData"))
```

13 Analysis: 004

```
# Model:
M
```

```
→ [1] "004"
```

- Standardization to 0.5 standard deviations for all covariates, following Gelman (2008).
- Outcome variable comes from the scores of the combination of policies.
- Explanatory variables come from the "adoption" part, where the means between 2010-2017 are taken to explain scores at a single point in time.
- A linear model is employed.
- We do NOT control for unequal variances (heteroskedasticity) in the EU and Financial crisis variables (whether belonging to the EU or having experienced a Financial crises varies the error). It shows no difference.
- We run a robust model with a t-distribution to avoid outliers from biasing our inference.
- We use standard weakly uninformative priors.
- For missing values in the explanatory variables, we use a conservative approach assigning them a probability distribution resembling the observed distribution.

```
load("scores-baseline.RData")
scores.countries.1d <- scores.countries %>%
  rename(`Carbon pricing` = Score)

load("scores-baseline-2dim.RData")
scores.countries.2d <- scores.countries %>%
  spread(Dimension, Score) %>%
  select(Configuration, ETS, Tax, Country)

da.full <- d # original data from "../adoption/"

da <- d %>% # original data from "../adoption/"
  # Consider only countries with scores
  # We only explain countries that have adopted the policy
  filter(Country %in% scores.countries$Country) %>%
  #
  # Consider countries without too much missingness
  #filter(!(Country %in% c("Liechtenstein", "South Africa", "Zambia"))) %>%
```

```
  #
  # Time range for which to calculate values
  filter(Year >= 2004 & Year <= 2019) %>%
  # Somalia does not have score
  filter(Country != "Somalia") %>%
  select(-iso2c, -iso3c, -Outcome, -Event) %>%
  unique() %>%
  gather(Variable, value, -Country, -Year) %>%
  group_by(Country, Variable) %>%
  summarize(value = mean(value, na.rm = TRUE)) %>%
  # Treat EU and Financial differently
  ungroup() %>%
  mutate(value = ifelse(Variable %in% c("EU", "Financial crisis",
                                       #"Kyoto Ratification",
                                       "Subnational tax"),
                        ifelse(value > 0, 1, 0),
                        value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  select(Country,
         `GDPpc (log)`,
         `State expenditure`,
         `Population (log)`,
         `Fossil fuel rents (log)`,
         `Democracy (Electoral)`,
         `Political constraints`,
         `Government effectiveness`,
         `EU`, `Subnational tax`,
         #`Debt (log)`,
         `Financial crisis`,
         #`Kyoto Ratification`,
         `Vulnerability`,
         `CO2pc (log)`) %>%
  left_join(scores.countries.1d) %>%
  left_join(scores.countries.2d) %>%
  select(-Configuration)
country.label <- as.character(da$Country)

###############################################################################
###### Interdependence using tidy approach
contiguity <-
  # The country that has adopted or not is the destination country
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(contiguity.dependency = sum(wAdopted, na.rm = TRUE))
```

```
distance ←
  # The country that has adopted or not is the destination country
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.distance),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.distance) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(distance.dependency = sum(wAdopted, na.rm = TRUE))

trade.dependency ←
  # The country that has adopted or not is the destination country
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin ≠ Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(trade.dependency = sum(wAdopted, na.rm = TRUE)) %>%
  ungroup()

# For competition dependency we need both the imports and the exports

trade.partner.dependency ←
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin ≠ Destination) %>%
#  select(Origin, Destination, Year, Outcome, wAdopted) %>%
  mutate(wAdopted = ifelse(is.na(wAdopted), 0, wAdopted)) %>%
  ungroup()

trade.partner.others.imports ←
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Imports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted.imports = Adopted * p.Imports) %>%
  filter(Origin ≠ Destination) %>%
  # Rename countries to better control the matrices
```

```
  # Partner refers to the trade partner of the first origin country
  # ThirdCountry refers to the competitor of the first origin country
  #   through the Destination==Partner
  rename(Partner = Origin, ThirdCountry = Destination) %>%
  select(Partner, ThirdCountry, Year, Outcome, wAdopted.imports) %>%
  mutate(wAdopted.imports = ifelse(is.na(wAdopted.imports), 0, wAdopted.imports)) %>%
  group_by(Partner, Year, Outcome) %>%
  summarize(w.ThirdCountry.imp = sum(wAdopted.imports)) %>%
  ungroup()

trade.competition <-
  left_join(trade.partner.dependency,
            trade.partner.others.imports,
            by = c("Destination" = "Partner",
                   "Year" = "Year",
                   "Outcome" = "Outcome")) %>%
  mutate(wAdopted.dual = wAdopted * w.ThirdCountry.imp) %>%
  filter(Origin != Destination) %>%
  group_by(Origin, Year, Outcome) %>%
  summarize(trade.competition = sum(wAdopted.dual, na.rm = TRUE)) %>%
  ungroup() %>%
  rename(Country = Origin)


##### End interdependence using tidy approach
##############################################################################

####### Weighting matrices
## First use only relevant countries and then row-normalize
#M.distances <- M.distances[match(country.label, dimnames(M.distances)[[1]]),
#                            match(country.label, dimnames(M.distances)[[2]])]
#RW.M.distances <- 100 * (1 / M.distances) /
#  apply(1 / M.distances, 1, sum, na.rm = TRUE)
#RW.M.distances[is.na(RW.M.distances)] <- 0
#stopifnot(dimnames(RW.M.distances)[[1]] == country.label)
#stopifnot(dimnames(RW.M.distances)[[2]] == country.label)
#
#M.borders <- M.borders[match(country.label, dimnames(M.borders)[[1]]),
#                            match(country.label, dimnames(M.borders)[[2]])]
#RW.M.borders <- M.borders / apply(M.borders, 1, sum, na.rm = TRUE)
#RW.M.borders[is.na(RW.M.borders)] <- 0
#stopifnot(dimnames(RW.M.borders)[[1]] == country.label)
#stopifnot(dimnames(RW.M.borders)[[2]] == country.label)
#
#
#M.trade <- M.trade[match(country.label, dimnames(M.trade)[[1]]),
#                            match(country.label, dimnames(M.trade)[[2]]),]
#RW.M.trade <- array(0, dim = dim(M.trade), dimnames = dimnames(M.trade))
## Get only the last but one year
#for (t in 1:(dim(RW.M.trade)[3])) {
#  RW.M.trade[,,t] <- M.trade[,,t] / apply(M.trade[,,t], 1, sum, na.rm = TRUE)
```

```
#}
#RW.M.trade <- RW.M.trade[,,(dim(RW.M.trade)[3] - 1)]
#RW.M.trade[is.na(RW.M.trade)] <- 0
#stopifnot(dimnames(RW.M.trade)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade)[[2]] == country.label)

# Outcome
y <- da %>%
  select(`Carbon pricing`, Tax, ETS) %>%
  as.matrix()
nOutcomes <- dim(y)[2]
outcome.label <- dimnames(y)[[2]]
outcome.has.tax <- ifelse(outcome.label %in% c("Tax"), 1, 0)
outcome.has.ets <- ifelse(outcome.label %in% c("ETS"), 1, 0)

# Explanatory variables are all standardized to 1 std deviation
# even originally binary ones, as they are weighted
X <- da %>%
  select(-c(`Carbon pricing`, Tax, ETS), -Model) %>%
  gather(Variable, value, -Country) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  spread(Variable, value) %>%
  #mutate(`EU * GDPpc (log)` = EU * `GDPpc (log)`) %>%
  select(-Country) %>%
  as.matrix()

X.interdependence <-

  # work with da.full to calculate the interdependencies
  select(da.full, Country, Year, Outcome) %>%
  left_join(contiguity) %>%
  left_join(distance) %>%
  left_join(trade.dependency) %>%
  left_join(trade.competition) %>%
  mutate(Country = as.factor(Country),
         Year = as.integer(Year),
         Outcome = as.factor(Outcome)) %>%
  gather(Variable, value, -c(Country, Year, Outcome)) %>%
#  filter(Variable != "distance.dependency") %>%
  mutate(Variable = factor(Variable, levels = c("distance.dependency",
                                                "contiguity.dependency",
                                                "trade.dependency",
                                                "trade.competition"))) %>%
  spread(Variable, value) %>%
#  # compute the EU interaction
#  left_join(select(da.full, Country, Year, Outcome, `EU`)) %>%
#  mutate(eu.trade.dependency = EU * trade.dependency) %>%
#  select(-EU) %>%
  # take average over time
  gather(Variable, value, -c(Country, Year,  Outcome)) %>%
```

```
  group_by(Country, Outcome, Variable) %>%
  summarize(time.average = mean(value, na.rm = TRUE)) %>%
  # take out values not in the analysis
  filter(Country %in% da$Country) %>%
  droplevels() %>%
  # take out variables non relevant
  filter(Outcome %in% c("Carbon pricing", "Carbon tax", "ETS")) %>%
  mutate(Outcome = factor(Outcome, levels = c("Carbon pricing", "Carbon tax", "ETS"))) %>%
  droplevels() %>%
  # standardize values
  group_by(Outcome, Variable) %>%
  mutate(time.average = std.zero(time.average)) %>%
  mutate(time.average = ifelse(is.nan(time.average), NA, time.average)) %>%
  filter(Variable %in% c(#"distance.dependency",
#                          "contiguity.dependency",
                          "trade.competition")) %>%
#                          "trade.dependency")) %>%
  reshape2::acast(Outcome ~ Country ~ Variable, value.var = "time.average")
stopifnot(da$Country == dimnames(X.interdependence)[[2]])
nCovI ← dim(X.interdependence)[3]


nO ← dim(y)[1]
country.label ← da$Country
X ← cbind("(Intercept)" = 1, X)
#X ← cbind(X, "GDPpc (log) * CO2pc" = X[,"GDPpc (log)"] * X[,"CO2pc"])
covariate.label ← c(dimnames(X)[[2]],
#                      "Interdependence (Borders)",
#                      "Interdependence (Distance)",
#                      "EU * Interdependence (Trade dependency)",
                     "Interdependence (Trade competition)",
#                      "Interdependence (Trade dependency)",
                     "ETS intensity", "Tax intensity")
#nCov ← dim(X)[2] + 4
nCov ← length(covariate.label)

covariate.label.order ← c("(Intercept)",
  "GDPpc (log)",
#  "EU * GDPpc (log)",
  "State expenditure",
  "Financial crisis", # Economic
  "CO2pc (log)",
  "Fossil fuel rents (log)", "Population (log)", "Vulnerability",   # Contribution to CC
  "Democracy (Electoral)",
  "Government effectiveness", "Political constraints",     # Institutional
#  "Interdependence (Borders)",
#  "Interdependence (Distance)",
#  "Interdependence (Trade dependency)",
  "Interdependence (Trade competition)",
#  "EU * Interdependence (Trade dependency)",
  "EU",
```

```
  #"Kyoto Ratification",
  "Subnational tax",
  "Tax intensity", "ETS intensity")


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

X.error ← X[,c("(Intercept)",
                "Vulnerability",
#                "Population (log)",
#                "GDPpc (log)",
#                "State expenditure",
#                "Political constraints",
#                "CO2pc (log)",
#                "Democracy (Electoral)",
#                "Financial crisis",
                "EU")]

nCovE ← dim(X.error)[2]
covariate.error.label ← dimnames(X.error)[[2]]
l0 ← rep(0, nCovE)
L0 ← diag(nCovE)
diag(L0) ← 0.5^-2

# Restrictions for theta variables that must be zero
theta.restrictions ← array(NA, dim = c(nOutcomes, nCov))
theta.restrictions[,(nCov-1):nCov] ← 0
theta.restrictions[which(outcome.has.tax == 1), (nCov-1)] ← NA
theta.restrictions[which(outcome.has.ets == 1), (nCov-0)] ← NA

B0.1 ← B0.2 ← B0.3 ← B0
# Carbon pricing
diag(B0.1)[(nCov - 1):nCov] ← 0.001 # high precision for when outcome has tax
# Tax
diag(B0.2)[nCov - 0] ← 0.001 # high precision for when outcome has tax
# ETS
diag(B0.3)[nCov - 1] ← 0.001 # high precision for when outcome has tax



D ← list(
  nO = nO,
  X = unname(X), nCov = nCov,
  nCovI = nCovI,
  X_error = unname(X.error), nCovE = nCovE,
  X_interdependence = unname(X.interdependence),
  b0 = b0, B0 = B0, df = nCov + 1,
  B0.1 = B0.1,
```

```
  B0.2 = B0.2,
  B0.3 = B0.3,
  l0 = l0, L0 = L0,
  nOutcomes = nOutcomes,
  outcome_has_tax = outcome.has.tax, outcome_has_ets = outcome.has.ets,
  theta = theta.restrictions,
  Y = unname(y),
  y = unname(y))

da.viz <- da %>%
  select(-Model) %>%
  gather(Variable, value, -c(Country, `Carbon pricing`, Tax, ETS)) %>%
  gather(Carbon, Score, -c(Country, Variable, value))

ggplot(da.viz, aes(x = value, y = Score)) +
  geom_point() +
  facet_grid(Carbon ~ Variable, scales = "free")
```

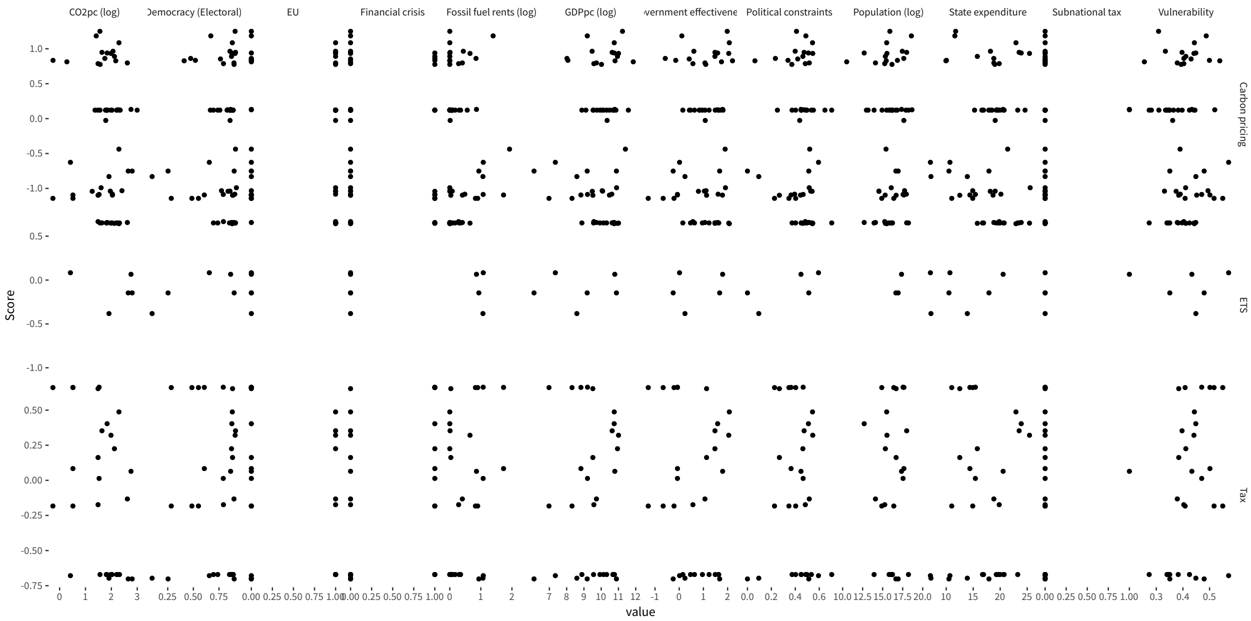


Figure 13.1: Scatterplots of the scores against the covariates.

```
m <- 'model {
  #
  # -- Observational component
  #
  for (O in 1:nOutcomes) {
    for (o in 1:nO) {
      #y[o,O] ~ dt(mu[o,O], tau[o,O], nu[O])
      #y[o,O] ~ dt(mu[o,O], tau[O], nu[O])
      #y[o,O] ~ dnorm(mu[o,O], tau[o,O])
      y[o,O] ~ dnorm(mu[o,O], tau[O])
      mu[o,O] <- inprod(X[o,1:(nCov-3)], theta[O,1:(nCov-3)])
#               + theta[O,(nCov-3)] * X_interdependence[O,o,1]
              + theta[O,(nCov-2)] * X_interdependence[O,o,1]
```

```
              + theta[O,(nCov-1)] * (outcome_has_tax[O] * Y[o,3])
              + theta[O,(nCov-0)] * (outcome_has_ets[O] * Y[o,2])
##      tau[o,O] <- 1 / sigma.sq[o,O]
##      sigma.sq[o,O] <- exp(inprod(X_error[o,], lambda[O,1:nCovE]))
#      tau[o,O] <- pow(sigma[o,O], -2)
#      sigma[o,O] <- exp(inprod(X_error[o,], lambda[O,1:nCovE]))
##      #sigma.sq[o,O] <- exp(lambda[O])
      resid[o,O] <- y[o,O] - mu[o,O]
    }
    #
    # -- Degrees of freedom, robust
    #
#    nu[O] <- exp(nu.log[O])
#    nu.log[O] ~ dunif(0, 4)
    #
    # -- Error component
    #
#    tau[O] ~ dgamma(0.001, 0.001)
#    sigma[O] <- 1 / sqrt(tau[O])
    tau[O] <- pow(sigma[O], -2)
    sigma[O] ~ dunif(0, 2)
    #
    # -- Priors for covariates and error variables
    #
    #theta[O,1:nCov] ~ dmnorm(b0, B0)
##    lambda[O,1] ~ dunif(-8, 2)    # Intercept of lambda, error exp()
###    lambda[O,1] ~ dnorm(-4, 1^-2)    # Intercept of lambda, error exp()
##    for (cove in 2:nCovE) {
##      lambda[O,cove] ~ dnorm(0, 1^-2)
##  }
#    for (cove in 1:nCovE) {
#      lambda[O,cove] ~ dnorm(0, 1^-2)
#    }
  }
  for (cov in 1:nCov) {
    theta[1,cov] ~ dnorm(0, 1^-2)
    theta[2,cov] ~ dnorm(0, 1^-2)
    theta[3,cov] ~ dnorm(0, 1^-2)
  }
#  theta[1,1:nCov] ~ dmnorm(b0, Omega.1)
#  theta[2,1:nCov] ~ dmnorm(b0, Omega.2)
#  theta[3,1:nCov] ~ dmnorm(b0, Omega.3)
  Omega.1 ~ dwish(B0.1, df)
  Omega.2 ~ dwish(B0.2, df)
  Omega.3 ~ dwish(B0.3, df)
  Sigma[1,1:nCov,1:nCov] <- inverse(Omega.1)
  Sigma[2,1:nCov,1:nCov] <- inverse(Omega.2)
  Sigma[3,1:nCov,1:nCov] <- inverse(Omega.3)
  #
  # -- Missing data
  #
```

```
  for (o in 1:nO) {
    for (cov in 1:(nCov-2-nCovI)) {
      X[o,cov] ~ dnorm(0, 1.5^-2)
      #X[o,cov] ~ dunif(-2, 2)
    }
    for (cov in 1:nCovI) {
      for (O in 1:nOutcomes) {
        X_interdependence[O,o,cov] ~ dnorm(0, 1.5^-2)
        #X_interdependence[O,o,cov] ~ dunif(-2, 2)
      }
    }
    for (cove in 1:nCovE) {
      X_error[o,cove] ~ dnorm(0, 1.5^-2)
      #X_error[o,cove] ~ dunif(-2, 2)
    }
  }
}'
write(m, file = paste0("models/model-explanatory-", M.lab, ".bug"))

par ← NULL
par ← c(par, "theta")
#par ← c(par, "delta")
#par ← c(par, "lambda")
par ← c(par, "resid")
#par ← c(par, "nu")
set.seed(14718)
inits ← function() {
  list(.RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6))
}

t0 ← proc.time()
rj ← run.jags(model = paste0("models/model-explanatory-", M.lab, ".bug"),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = c("glm", "lecuyer"),
               n.chains = chains, thin = thin,
               adapt = adapt, burnin = burnin, sample = sample,
               monitor = par, method = "parallel", summarise = FALSE)
s ← as.mcmc.list(rj)
save(s, file = paste0("sample-explanatory-", M.lab, ".RData"))
proc.time() - t0

load(file = paste0("sample-explanatory-", M.lab, ".RData"))

ggmcmc(ggs(s, family = "theta|lambda|nu"), plot = c("traceplot", "crosscorrelation", "caterpillar"))
ggmcmc(ggs(s, family = "theta|lambda"), plot = c("traceplot", "crosscorrelation", "caterpillar"))

ggs(s,
    family = "theta\\[1,15|theta\\[1,14|theta\\[1,4|lambda\\[1,1|lambda\\[1,5") %>% #,
#     par_label = bind_rows(L.lambda, L.theta)) %>%
  select(Iteration, Chain, Parameter, value) %>%
  spread(Parameter, value) %>%
  select(-Iteration, Chain) %>%
```

```
  ggpairs()

###
ggs(s, family = "theta\\[1,14") %>%
  group_by(Parameter) %>%
  summarize(ppos = length(which(value > 0)) / n())

ggs(s, family = "theta\\[1,14|theta\\[1,15") %>%
  ggs_running()

ggs(s, family = "theta\\[1,|lambda\\[1,") %>%
  spread(Parameter, value) %>%
  gather(Parameter, value, -c(Iteration, Chain, `theta[1,14]`)) %>%
  ggplot(aes(x = `theta[1,14]`, y = value, color = as.factor(Chain))) +
  geom_point(alpha = 0.1) +
  facet_wrap(~ Parameter, scales = "free")



ggs_crosscorrelation(ggs(s, family = "theta\\[3", par_labels = L.theta))

L.lambda <- plab("lambda", list(Outcome = outcome.label, Covariate = covariate.error.label))
S.lambda <- ggs(s, family = "^lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda, label = "Covariate") +
  geom_vline(xintercept = 0, lty = 3) +
  facet_grid(~ Outcome)

L.theta <- plab("theta", list(Outcome = outcome.label, Covariate = covariate.label))
#L.delta <- plab("delta", list(Covariate = c("Interdependence (Borders)",
#                                            "Interdependence (Trade)")))
#L.theta <- bind_rows(L.theta, L.delta)
S.theta <- ggs(s, family = "^theta\\[|^delta\\[", par_labels = L.theta) %>%
  mutate(Model = M) %>%
  mutate(Covariate = factor(Covariate, rev(covariate.label.order)))
save(S.theta, file = paste0("samples-theta-", M.lab, ".RData"))

S.theta %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_grid(~ Outcome)


S.theta %>%
  filter(Covariate != "(Intercept)") %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_grid(~ Outcome)


S.theta %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("(Intercept)",
                           "Tax intensity",
                           "ETS intensity")) %>%
```

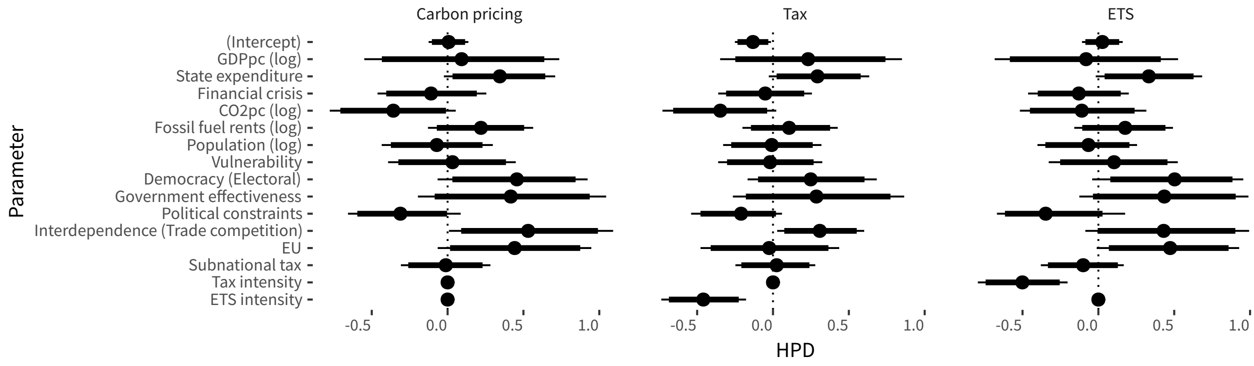


Figure 13.2: HPD of the effects of covariates

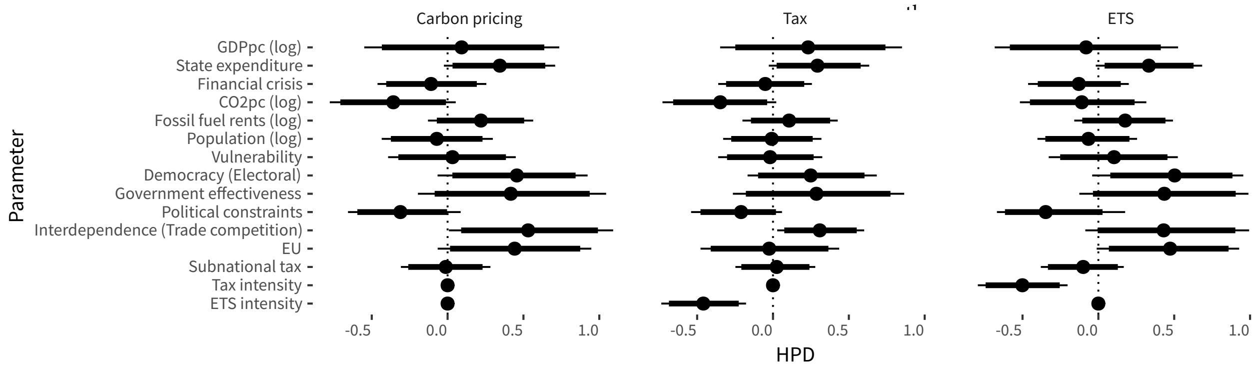


Figure 13.3: HPD of the effects of covariates on the score, without the intercept.

```
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)

S.theta %>%
  ci() %>%
  ggplot(aes(ymin = low, ymax = high,
             y = median, x = reorder(Covariate, median),
             color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.3)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark 2")

data.sd <- da %>%
  select(`Carbon pricing`, Tax, ETS) %>%
  gather(Outcome, value) %>%
  group_by(Outcome) %>%
  summarize(data.sd = sd(value, na.rm = TRUE))

L.rsd <- plab("resid", list(Observation = 1:nO, Outcome = outcome.label))
S.rsd <- ggs(s, family = "resid", par_labels = L.rsd, sort = FALSE) %>%
  group_by(Iteration, Chain, Outcome) %>%
  summarize(`Residual SD` = sd(value))
```

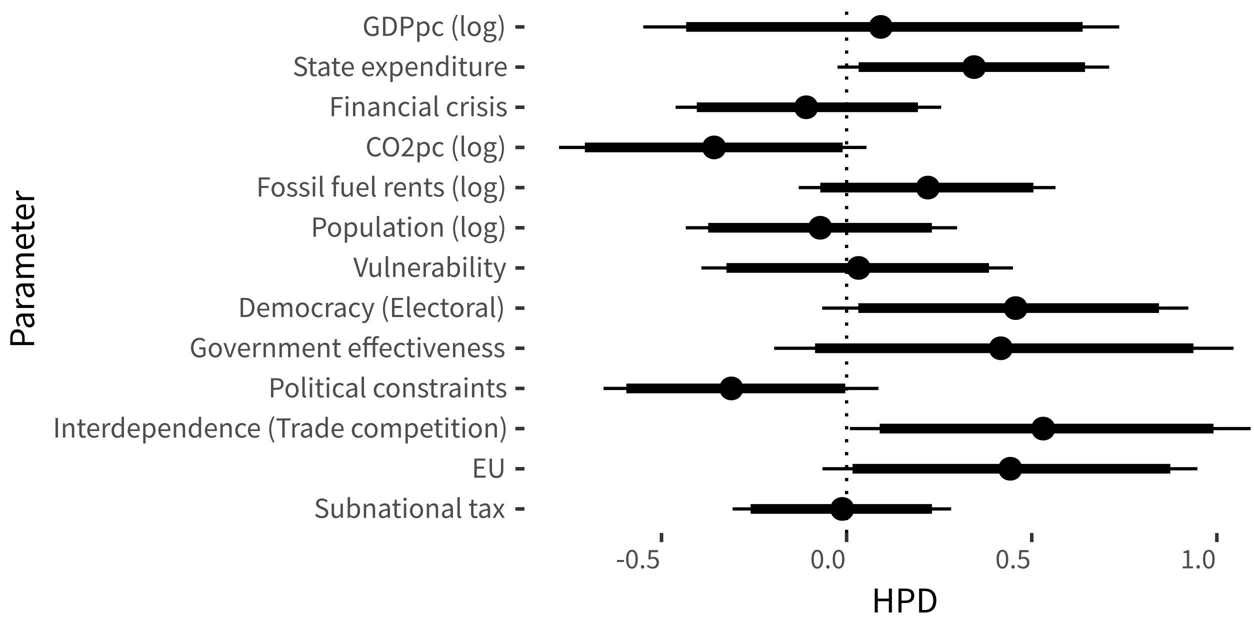


Figure 13.4: HPD of the effects of covariates on the score of carbon pricing, without the intercept and the intensity variables.

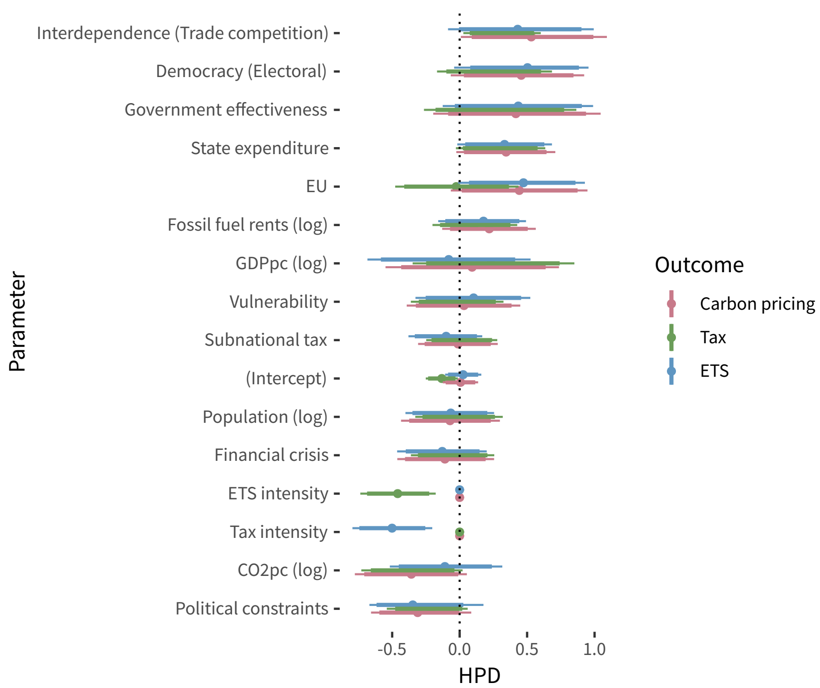


Figure 13.5: HPD of parameters, comparing outcomes.

```r
ggplot(S.rsd, aes(x = `Residual SD`)) +
  geom_histogram() +
  geom_vline(data = data.sd, aes(xintercept = data.sd), lty = 1) +
  expand_limits(x = 0) +
  facet_grid(~ Outcome)
```

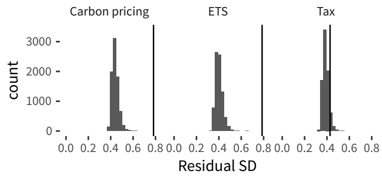


Figure 13.6: Residual standard deviation.

Model fit, pseudo-R^2

```r
S.rsd %>%
  group_by(Outcome) %>%
  summarize(MedianRSD = median(`Residual SD`)) %>%
  left_join(data.sd) %>%
  mutate(PseudoR2 = 1 - (MedianRSD / data.sd))
```

```
→ # A tibble: 3 x 4
→   Outcome        MedianRSD data.sd PseudoR2
→   <chr>              <dbl>   <dbl>    <dbl>
→ 1 Carbon pricing     0.436   0.783    0.443
→ 2 Tax                0.383   0.428    0.104
→ 3 ETS                0.398   0.787    0.493
```

Worstly predicted cases.

```r
L.data ← plab("resid", list(Country = country.label, Outcome = outcome.label))
S.country ← ggs(s, family = "resid", par_labels = L.data) %>%
  group_by(Country, Outcome) %>%
  summarize(`Average residual` = mean(value))


# Manually calculate PCP
# as ggmcmc's ggs_pcp() is not ready for matrices as input for outcome

wp.countries ← S.country %>% # worstly predicted cases
  ungroup() %>%
  arrange(desc(abs(`Average residual`))) %>%
  slice(1:20)

tc ← "Worstly predicted cases, by absolute average residual."
if (knitr::is_latex_output()) {
  kable(wp.countries, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(wp.countries, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 13.1: Worstly predicted cases, by absolute average residual.

Country	Outcome	Average residual
Chile	ETS	-0.8796
Switzerland	Carbon pricing	0.8583

Chile	Carbon pricing	-0.7393
India	ETS	0.7234
Ireland	Carbon pricing	0.6801
Ireland	Tax	0.6590
Iceland	Carbon pricing	0.6123
South Korea	ETS	0.6071
Netherlands	Carbon pricing	-0.5963
Switzerland	Tax	0.5906
Iceland	ETS	0.5883
Australia	Tax	-0.5853
Austria	Carbon pricing	-0.5763
Slovenia	Tax	0.5743
Lithuania	Tax	-0.5534
Albania	ETS	-0.5475
Japan	Carbon pricing	-0.5380
Canada	Carbon pricing	0.5340
New Zealand	Tax	-0.5241
Netherlands	Tax	-0.5178

```
load("carbon_pricing-patterns.RData")

scores.3 <- scores.countries.1d %>%
  left_join(select(configurations, Configuration, `ETS Adoption`, `Tax Adoption`)) %>%
  left_join(scores.countries.2d) %>%
  select(-c(Model, Configuration)) %>%
  gather(Policy, Intensity, -c(Country, `Tax Adoption`, `ETS Adoption`)) %>%
  mutate(shaded = ifelse(`ETS Adoption` == 0 & Policy == "ETS", TRUE, FALSE)) %>%
  mutate(shaded = ifelse(`Tax Adoption` == 0 & Policy == "Tax", TRUE, shaded))

ggplot(scores.3, aes(x = Intensity,
                     y = reorder(Country, Intensity),
                     color = shaded)) +
  geom_point() +
  facet_grid(~ Policy) +
  ylab("Country") +
  scale_color_manual(values = c("black", "grey")) +
  guides(color = FALSE)
```

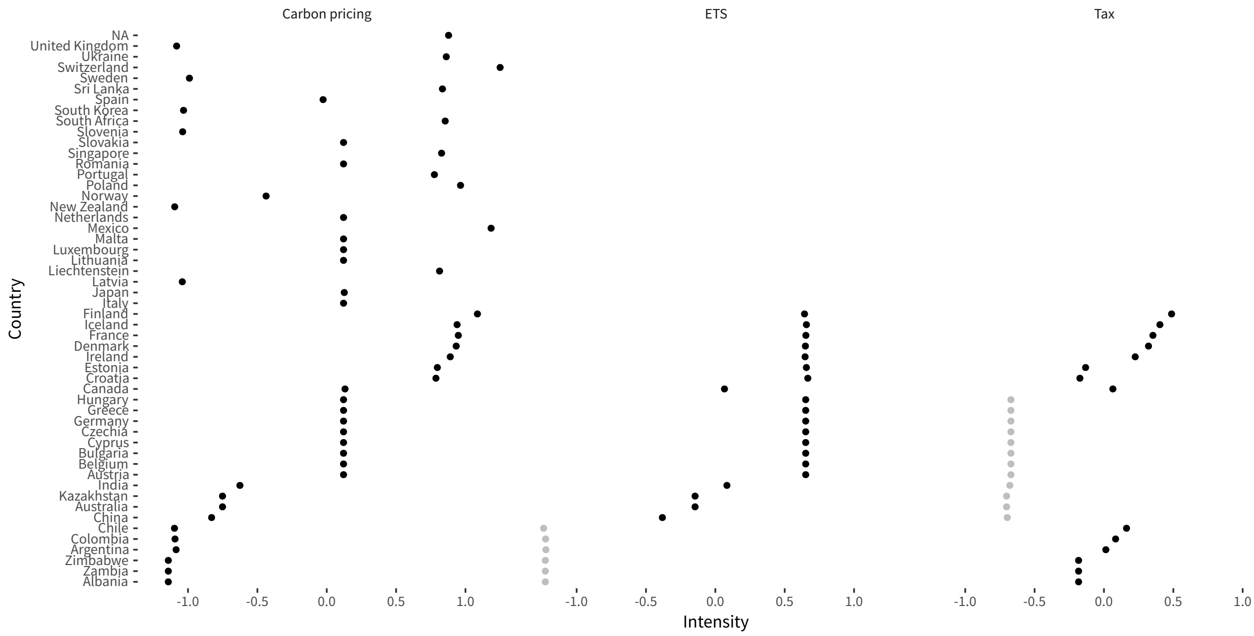


Figure 13.7: Policy intensity in carbon pricing, ETS and Tax.

14 Analysis: 007

```
# Model:
M
```

```
→ [1] "007"
```

- Standardization to 0.5 standard deviations for all covariates, following Gelman (2008).
- Outcome variable comes from the scores of the combination of policies.
- Explanatory variables come from the "adoption" part, where the means between 2010-2017 are taken to explain scores at a single point in time.
- A linear model is employed.
- We do NOT control for unequal variances (heteroskedasticity) in the EU and Financial crisis variables (whether belonging to the EU or having experienced a Financial crises varies the error). It shows no difference.
- We run a robust model with a t-distribution to avoid outliers from biasing our inference.
- We use standard weakly uninformative priors.
- For missing values in the explanatory variables, we use a conservative approach assigning them a probability distribution resembling the observed distribution.

```
load("scores-baseline.RData")
scores.countries.1d <- scores.countries %>%
  rename(`Carbon pricing` = Score)

load("scores-baseline-2dim.RData")
scores.countries.2d <- scores.countries %>%
  spread(Dimension, Score) %>%
  select(Configuration, ETS, Tax, Country)

da.full <- d # original data from "../adoption/"

da <- d %>% # original data from "../adoption/"
  # Consider only countries with scores
  # We only explain countries that have adopted the policy
  filter(Country %in% scores.countries$Country) %>%
  #
  # Consider countries without too much missingness
  #filter(!(Country %in% c("Liechtenstein", "South Africa", "Zambia"))) %>%
```

```
  #
  # Time range for which to calculate values
  filter(Year >= 2004 & Year <= 2019) %>%
  # Somalia does not have score
  filter(Country != "Somalia") %>%
  select(-iso2c, -iso3c, -Outcome, -Event) %>%
  unique() %>%
  gather(Variable, value, -Country, -Year) %>%
  group_by(Country, Variable) %>%
  summarize(value = mean(value, na.rm = TRUE)) %>%
  # Treat EU and Financial differently
  ungroup() %>%
  mutate(value = ifelse(Variable %in% c("EU", "Financial crisis",
                                        #"Kyoto Ratification",
                                        "Subnational tax"),
                        ifelse(value > 0, 1, 0),
                        value)) %>%
  mutate(value = ifelse(is.nan(value), NA, value)) %>%
  spread(Variable, value) %>%
  select(Country,
         `GDPpc (log)`,
         `State expenditure`,
         `Population (log)`,
         `Fossil fuel rents (log)`,
         `Democracy (Electoral)`,
         `Political constraints`,
         `Government effectiveness`,
         `EU`, `Subnational tax`,
         #`Debt (log)`,
         `Financial crisis`,
         #`Kyoto Ratification`,
         `Vulnerability`,
         `CO2pc (log)`) %>%
  left_join(scores.countries.1d) %>%
  left_join(scores.countries.2d) %>%
  select(-Configuration)
country.label <- as.character(da$Country)

##############################################################################
###### Interdependence using tidy approach
contiguity <-
  # The country that has adopted or not is the destination country
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.contiguous),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.contiguous) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(contiguity.dependency = sum(wAdopted, na.rm = TRUE))
```

```
distance <-
  # The country that has adopted or not is the destination country
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(geography %>%
            select(Origin, Destination, p.distance),
            by = c("Destination" = "Destination")) %>%
  # Multiply the adoption in other countries times the percentege of contiguity
  mutate(wAdopted = Adopted * p.distance) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(distance.dependency = sum(wAdopted, na.rm = TRUE))

trade.dependency <-
  # The country that has adopted or not is the destination country
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
  rename(Country = Origin) %>%
  group_by(Country, Year, Outcome) %>%
  summarize(trade.dependency = sum(wAdopted, na.rm = TRUE)) %>%
  ungroup()

# For competition dependency we need both the imports and the exports

trade.partner.dependency <-
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Exports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted = Adopted * p.Exports) %>%
  filter(Origin != Destination) %>%
#  select(Origin, Destination, Year, Outcome, wAdopted) %>%
  mutate(wAdopted = ifelse(is.na(wAdopted), 0, wAdopted)) %>%
  ungroup()

trade.partner.others.imports <-
  select(da.full, Destination = Country, Year, Outcome, Adopted) %>%
  left_join(trade.p %>%
            select(Origin, Destination, Year, p.Imports),
            by = c("Destination" = "Destination", "Year" = "Year")) %>%
  # Multiply the adoption in other countries times the percentege of exports
  mutate(wAdopted.imports = Adopted * p.Imports) %>%
  filter(Origin != Destination) %>%
  # Rename countries to better control the matrices
```

```
  # Partner refers to the trade partner of the first origin country
  # ThirdCountry refers to the competitor of the first origin country
  #   through the Destination==Partner
  rename(Partner = Origin, ThirdCountry = Destination) %>%
  select(Partner, ThirdCountry, Year, Outcome, wAdopted.imports) %>%
  mutate(wAdopted.imports = ifelse(is.na(wAdopted.imports), 0, wAdopted.imports)) %>%
  group_by(Partner, Year, Outcome) %>%
  summarize(w.ThirdCountry.imp = sum(wAdopted.imports)) %>%
  ungroup()

trade.competition <-
  left_join(trade.partner.dependency,
            trade.partner.others.imports,
            by = c("Destination" = "Partner",
                   "Year" = "Year",
                   "Outcome" = "Outcome")) %>%
  mutate(wAdopted.dual = wAdopted * w.ThirdCountry.imp) %>%
  filter(Origin != Destination) %>%
  group_by(Origin, Year, Outcome) %>%
  summarize(trade.competition = sum(wAdopted.dual, na.rm = TRUE)) %>%
  ungroup() %>%
  rename(Country = Origin)


##### End interdependence using tidy approach
##############################################################################

####### Weighting matrices
## First use only relevant countries and then row-normalize
#M.distances <- M.distances[match(country.label, dimnames(M.distances)[[1]]),
#                           match(country.label, dimnames(M.distances)[[2]])]
#RW.M.distances <- 100 * (1 / M.distances) /
#  apply(1 / M.distances, 1, sum, na.rm = TRUE)
#RW.M.distances[is.na(RW.M.distances)] <- 0
#stopifnot(dimnames(RW.M.distances)[[1]] == country.label)
#stopifnot(dimnames(RW.M.distances)[[2]] == country.label)
#
#M.borders <- M.borders[match(country.label, dimnames(M.borders)[[1]]),
#                           match(country.label, dimnames(M.borders)[[2]])]
#RW.M.borders <- M.borders / apply(M.borders, 1, sum, na.rm = TRUE)
#RW.M.borders[is.na(RW.M.borders)] <- 0
#stopifnot(dimnames(RW.M.borders)[[1]] == country.label)
#stopifnot(dimnames(RW.M.borders)[[2]] == country.label)
#
#
#M.trade <- M.trade[match(country.label, dimnames(M.trade)[[1]]),
#                           match(country.label, dimnames(M.trade)[[2]]),]
#RW.M.trade <- array(0, dim = dim(M.trade), dimnames = dimnames(M.trade))
## Get only the last but one year
#for (t in 1:(dim(RW.M.trade)[3])) {
#  RW.M.trade[,,t] <- M.trade[,,t] / apply(M.trade[,,t], 1, sum, na.rm = TRUE)
```

```
#}
#RW.M.trade <- RW.M.trade[,,(dim(RW.M.trade)[3] - 1)]
#RW.M.trade[is.na(RW.M.trade)] <- 0
#stopifnot(dimnames(RW.M.trade)[[1]] == country.label)
#stopifnot(dimnames(RW.M.trade)[[2]] == country.label)

# Outcome
y <- da %>%
  select(`Carbon pricing`, Tax, ETS) %>%
  as.matrix()
nOutcomes <- dim(y)[2]
outcome.label <- dimnames(y)[[2]]
outcome.has.tax <- ifelse(outcome.label %in% c("Tax"), 1, 0)
outcome.has.ets <- ifelse(outcome.label %in% c("ETS"), 1, 0)

# Explanatory variables are all standardized to 1 std deviation
# even originally binary ones, as they are weighted
X <- da %>%
  select(-c(`Carbon pricing`, Tax, ETS), -Model) %>%
  gather(Variable, value, -Country) %>%
  group_by(Variable) %>%
  mutate(value = std(value)) %>%
  ungroup() %>%
  spread(Variable, value) %>%
  #mutate(`EU * GDPpc (log)` = EU * `GDPpc (log)`) %>%
  select(-Country) %>%
  as.matrix()

X.interdependence <-

  # work with da.full to calculate the interdependencies
  select(da.full, Country, Year, Outcome) %>%
  left_join(contiguity) %>%
  left_join(distance) %>%
  left_join(trade.dependency) %>%
  left_join(trade.competition) %>%
  mutate(Country = as.factor(Country),
         Year = as.integer(Year),
         Outcome = as.factor(Outcome)) %>%
  gather(Variable, value, -c(Country, Year, Outcome)) %>%
#  filter(Variable != "distance.dependency") %>%
  mutate(Variable = factor(Variable, levels = c("distance.dependency",
                                                "contiguity.dependency",
                                                "trade.dependency",
                                                "trade.competition"))) %>%
  spread(Variable, value) %>%
#  # compute the EU interaction
#  left_join(select(da.full, Country, Year, Outcome, `EU`)) %>%
#  mutate(eu.trade.dependency = EU * trade.dependency) %>%
#  select(-EU) %>%
  # take average over time
  gather(Variable, value, -c(Country, Year,  Outcome)) %>%
```

```
  group_by(Country, Outcome, Variable) %>%
  summarize(time.average = mean(value, na.rm = TRUE)) %>%
  # take out values not in the analysis
  filter(Country %in% da$Country) %>%
  droplevels() %>%
  # take out variables non relevant
  filter(Outcome %in% c("Carbon pricing", "Carbon tax", "ETS")) %>%
  mutate(Outcome = factor(Outcome, levels = c("Carbon pricing", "Carbon tax", "ETS"))) %>%
  droplevels() %>%
  # standardize values
  group_by(Outcome, Variable) %>%
  mutate(time.average = std.zero(time.average)) %>%
  mutate(time.average = ifelse(is.nan(time.average), NA, time.average)) %>%
  filter(Variable %in% c(#"distance.dependency",
#                         "contiguity.dependency",
#                         "trade.competition")) %>%
                         "trade.dependency")) %>%
  reshape2::acast(Outcome ~ Country ~ Variable, value.var = "time.average")
stopifnot(da$Country == dimnames(X.interdependence)[[2]])
nCovI <- dim(X.interdependence)[3]


nO <- dim(y)[1]
country.label <- da$Country
X <- cbind("(Intercept)" = 1, X)
#X <- cbind(X, "GDPpc (log) * CO2pc" = X[,"GDPpc (log)"] * X[,"CO2pc"])
covariate.label <- c(dimnames(X)[[2]],
#                     "Interdependence (Borders)",
#                     "Interdependence (Distance)",
#                     "EU * Interdependence (Trade dependency)",
#                     "Interdependence (Trade competition)",
                     "Interdependence (Trade dependency)",
                     "ETS intensity", "Tax intensity")
#nCov <- dim(X)[2] + 4
nCov <- length(covariate.label)

covariate.label.order <- c("(Intercept)",
  "GDPpc (log)",
#  "EU * GDPpc (log)",
  "State expenditure",
  "Financial crisis", # Economic
  "CO2pc (log)",
  "Fossil fuel rents (log)", "Population (log)", "Vulnerability",   # Contribution to CC
  "Democracy (Electoral)",
  "Government effectiveness", "Political constraints",     # Institutional
#  "Interdependence (Borders)",
#  "Interdependence (Distance)",
  "Interdependence (Trade dependency)",
#  "Interdependence (Trade competition)",
#  "EU * Interdependence (Trade dependency)",
  "EU",
```

```
  #"Kyoto Ratification",
  "Subnational tax",
  "Tax intensity", "ETS intensity")


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

X.error ← X[,c("(Intercept)",
                "Vulnerability",
#                "Population (log)",
#                "GDPpc (log)",
#                "State expenditure",
#                "Political constraints",
#                "CO2pc (log)",
#                "Democracy (Electoral)",
#                "Financial crisis",
                "EU")]

nCovE ← dim(X.error)[2]
covariate.error.label ← dimnames(X.error)[[2]]
l0 ← rep(0, nCovE)
L0 ← diag(nCovE)
diag(L0) ← 0.5^-2

# Restrictions for theta variables that must be zero
theta.restrictions ← array(NA, dim = c(nOutcomes, nCov))
theta.restrictions[,(nCov-1):nCov] ← 0
theta.restrictions[which(outcome.has.tax == 1), (nCov-1)] ← NA
theta.restrictions[which(outcome.has.ets == 1), (nCov-0)] ← NA

B0.1 ← B0.2 ← B0.3 ← B0
# Carbon pricing
diag(B0.1)[(nCov - 1):nCov] ← 0.001 # high precision for when outcome has tax
# Tax
diag(B0.2)[nCov - 0] ← 0.001 # high precision for when outcome has tax
# ETS
diag(B0.3)[nCov - 1] ← 0.001 # high precision for when outcome has tax




D ← list(
  n0 = n0,
  X = unname(X), nCov = nCov,
  nCovI = nCovI,
  X_error = unname(X.error), nCovE = nCovE,
  X_interdependence = unname(X.interdependence),
  b0 = b0, B0 = B0, df = nCov + 1,
  B0.1 = B0.1,
```

```
  B0.2 = B0.2,
  B0.3 = B0.3,
  l0 = l0, L0 = L0,
  nOutcomes = nOutcomes,
  outcome_has_tax = outcome.has.tax, outcome_has_ets = outcome.has.ets,
  theta = theta.restrictions,
  Y = unname(y),
  y = unname(y))

da.viz <- da %>%
  select(-Model) %>%
  gather(Variable, value, -c(Country, `Carbon pricing`, Tax, ETS)) %>%
  gather(Carbon, Score, -c(Country, Variable, value))

ggplot(da.viz, aes(x = value, y = Score)) +
  geom_point() +
  facet_grid(Carbon ~ Variable, scales = "free")
```

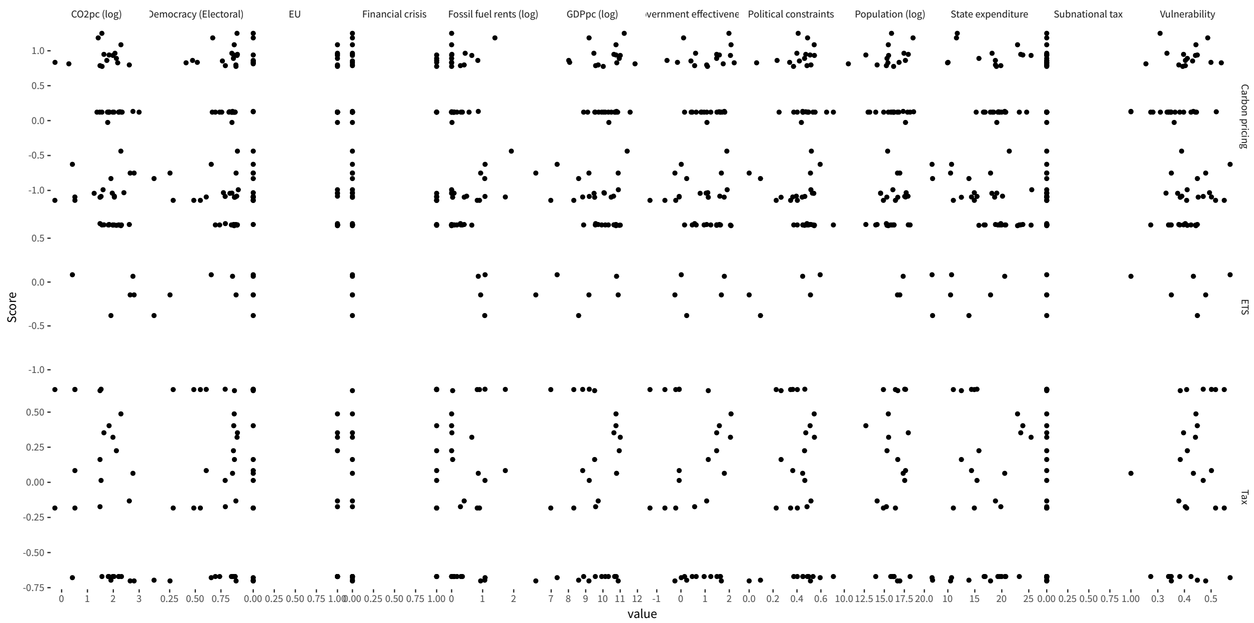


Figure 14.1: Scatterplots of the scores against the covariates.

```
m <- 'model {
  #
  # -- Observational component
  #
  for (O in 1:nOutcomes) {
    for (o in 1:nO) {
      #y[o,O] ~ dt(mu[o,O], tau[o,O], nu[O])
      #y[o,O] ~ dt(mu[o,O], tau[O], nu[O])
      #y[o,O] ~ dnorm(mu[o,O], tau[o,O])
      y[o,O] ~ dnorm(mu[o,O], tau[O])
      mu[o,O] <- inprod(X[o,1:(nCov-3)], theta[O,1:(nCov-3)])
#                 + theta[O,(nCov-3)] * X_interdependence[O,o,1]
                + theta[O,(nCov-2)] * X_interdependence[O,o,1]
```

```
              + theta[O,(nCov-1)] * (outcome_has_tax[O] * Y[o,3])
              + theta[O,(nCov-0)] * (outcome_has_ets[O] * Y[o,2])
##      tau[o,O] <- 1 / sigma.sq[o,O]
##      sigma.sq[o,O] <- exp(inprod(X_error[o,], lambda[O,1:nCovE]))
#      tau[o,O] <- pow(sigma[o,O], -2)
#      sigma[o,O] <- exp(inprod(X_error[o,], lambda[O,1:nCovE]))
##      #sigma.sq[o,O] <- exp(lambda[O])
      resid[o,O] <- y[o,O] - mu[o,O]
    }
    #
    # -- Degrees of freedom, robust
    #
#    nu[O] <- exp(nu.log[O])
#    nu.log[O] ~ dunif(0, 4)
    #
    # -- Error component
    #
#    tau[O] ~ dgamma(0.001, 0.001)
#    sigma[O] <- 1 / sqrt(tau[O])
    tau[O] <- pow(sigma[O], -2)
    sigma[O] ~ dunif(0, 2)
    #
    # -- Priors for covariates and error variables
    #
    #theta[O,1:nCov] ~ dmnorm(b0, B0)
##    lambda[O,1] ~ dunif(-8, 2)    # Intercept of lambda, error exp()
###    lambda[O,1] ~ dnorm(-4, 1^-2)    # Intercept of lambda, error exp()
##    for (cove in 2:nCovE) {
##      lambda[O,cove] ~ dnorm(0, 1^-2)
##  }
#    for (cove in 1:nCovE) {
#      lambda[O,cove] ~ dnorm(0, 1^-2)
#    }
  }
  for (cov in 1:nCov) {
    theta[1,cov] ~ dnorm(0, 1^-2)
    theta[2,cov] ~ dnorm(0, 1^-2)
    theta[3,cov] ~ dnorm(0, 1^-2)
  }
#  theta[1,1:nCov] ~ dmnorm(b0, Omega.1)
#  theta[2,1:nCov] ~ dmnorm(b0, Omega.2)
#  theta[3,1:nCov] ~ dmnorm(b0, Omega.3)
  Omega.1 ~ dwish(B0.1, df)
  Omega.2 ~ dwish(B0.2, df)
  Omega.3 ~ dwish(B0.3, df)
  Sigma[1,1:nCov,1:nCov] <- inverse(Omega.1)
  Sigma[2,1:nCov,1:nCov] <- inverse(Omega.2)
  Sigma[3,1:nCov,1:nCov] <- inverse(Omega.3)
  #
  # -- Missing data
  #
```

```
  for (o in 1:nO) {
    for (cov in 1:(nCov-2-nCovI)) {
      X[o,cov] ~ dnorm(0, 1.5^-2)
      #X[o,cov] ~ dunif(-2, 2)
    }
    for (cov in 1:nCovI) {
      for (O in 1:nOutcomes) {
        X_interdependence[O,o,cov] ~ dnorm(0, 1.5^-2)
        #X_interdependence[O,o,cov] ~ dunif(-2, 2)
      }
    }
    for (cove in 1:nCovE) {
      X_error[o,cove] ~ dnorm(0, 1.5^-2)
      #X_error[o,cove] ~ dunif(-2, 2)
    }
  }
}'
write(m, file = paste0("models/model-explanatory-", M.lab, ".bug"))

par ← NULL
par ← c(par, "theta")
#par ← c(par, "delta")
#par ← c(par, "lambda")
par ← c(par, "resid")
#par ← c(par, "nu")
set.seed(14718)
inits ← function() {
  list(.RNG.name = "base::Super-Duper", .RNG.seed = runif(1, 1, 1e6))
}

t0 ← proc.time()
rj ← run.jags(model = paste0("models/model-explanatory-", M.lab, ".bug"),
               data = dump.format(D, checkvalid = FALSE),
               inits = inits,
               modules = c("glm", "lecuyer"),
               n.chains = chains, thin = thin,
               adapt = adapt, burnin = burnin, sample = sample,
               monitor = par, method = "parallel", summarise = FALSE)
s ← as.mcmc.list(rj)
save(s, file = paste0("sample-explanatory-", M.lab, ".RData"))
proc.time() - t0

load(file = paste0("sample-explanatory-", M.lab, ".RData"))

ggmcmc(ggs(s, family = "theta|lambda|nu"), plot = c("traceplot", "crosscorrelation", "caterpillar"))
ggmcmc(ggs(s, family = "theta|lambda"), plot = c("traceplot", "crosscorrelation", "caterpillar"))

ggs(s,
    family = "theta\\[1,15|theta\\[1,14|theta\\[1,4|lambda\\[1,1|lambda\\[1,5") %>% #,
#     par_label = bind_rows(L.lambda, L.theta)) %>%
  select(Iteration, Chain, Parameter, value) %>%
  spread(Parameter, value) %>%
  select(-Iteration, Chain) %>%
```

```
  ggpairs()

###
ggs(s, family = "theta\\[1,14") %>%
  group_by(Parameter) %>%
  summarize(ppos = length(which(value > 0)) / n())

ggs(s, family = "theta\\[1,14|theta\\[1,15") %>%
  ggs_running()

ggs(s, family = "theta\\[1,|lambda\\[1,") %>%
  spread(Parameter, value) %>%
  gather(Parameter, value, -c(Iteration, Chain, `theta[1,14]`)) %>%
  ggplot(aes(x = `theta[1,14]`, y = value, color = as.factor(Chain))) +
  geom_point(alpha = 0.1) +
  facet_wrap(~ Parameter, scales = "free")



ggs_crosscorrelation(ggs(s, family = "theta\\[3", par_labels = L.theta))

L.lambda <- plab("lambda", list(Outcome = outcome.label, Covariate = covariate.error.label))
S.lambda <- ggs(s, family = "^lambda\\[", par_labels = L.lambda)
ggs_caterpillar(S.lambda, label = "Covariate") +
  geom_vline(xintercept = 0, lty = 3) +
  facet_grid(~ Outcome)

L.theta <- plab("theta", list(Outcome = outcome.label, Covariate = covariate.label))
#L.delta <- plab("delta", list(Covariate = c("Interdependence (Borders)",
#                                            "Interdependence (Trade)")))
#L.theta <- bind_rows(L.theta, L.delta)
S.theta <- ggs(s, family = "^theta\\[|^delta\\[", par_labels = L.theta) %>%
  mutate(Model = M) %>%
  mutate(Covariate = factor(Covariate, rev(covariate.label.order)))
save(S.theta, file = paste0("samples-theta-", M.lab, ".RData"))

S.theta %>%
  ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_grid(~ Outcome)


S.theta %>%
  filter(Covariate != "(Intercept)") %>%
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3) +
  facet_grid(~ Outcome)


S.theta %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("(Intercept)",
                           "Tax intensity",
                           "ETS intensity")) %>%
```

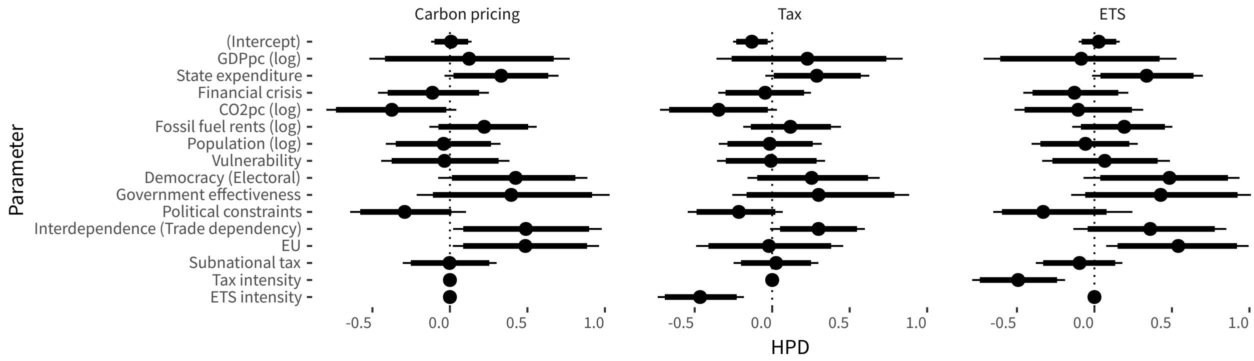


Figure 14.2: HPD of the effects of covariates on the score.

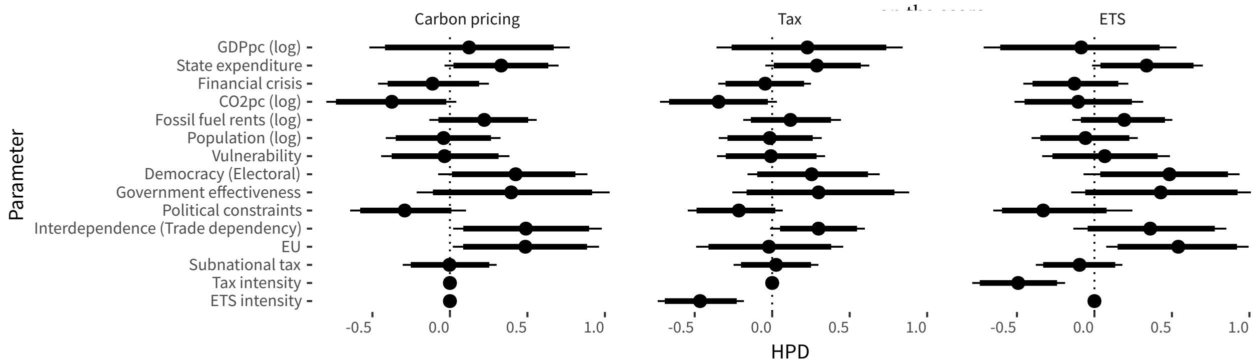


Figure 14.3: HPD of the effects of covariates on the score, without the intercept.

```
ggs_caterpillar(label = "Covariate", sort = FALSE) +
  geom_vline(xintercept = 0, lty = 3)

S.theta %>%
  ci() %>%
  ggplot(aes(ymin = low, ymax = high,
             y = median, x = reorder(Covariate, median),
             color = Outcome)) +
  coord_flip() +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_linerange(position = position_dodge(width = 0.3)) +
  geom_linerange(aes(ymin = Low, ymax = High), size = 1, position = position_dodge(width = 0.3)) +
  geom_hline(aes(yintercept = 0), lty = 3) +
  xlab("Parameter") + ylab("HPD") +
  scale_color_discrete_qualitative(palette = "Dark 2")

data.sd <- da %>%
  select(`Carbon pricing`, Tax, ETS) %>%
  gather(Outcome, value) %>%
  group_by(Outcome) %>%
  summarize(data.sd = sd(value, na.rm = TRUE))

L.rsd <- plab("resid", list(Observation = 1:nO, Outcome = outcome.label))
S.rsd <- ggs(s, family = "resid", par_labels = L.rsd, sort = FALSE) %>%
  group_by(Iteration, Chain, Outcome) %>%
  summarize(`Residual SD` = sd(value))
```

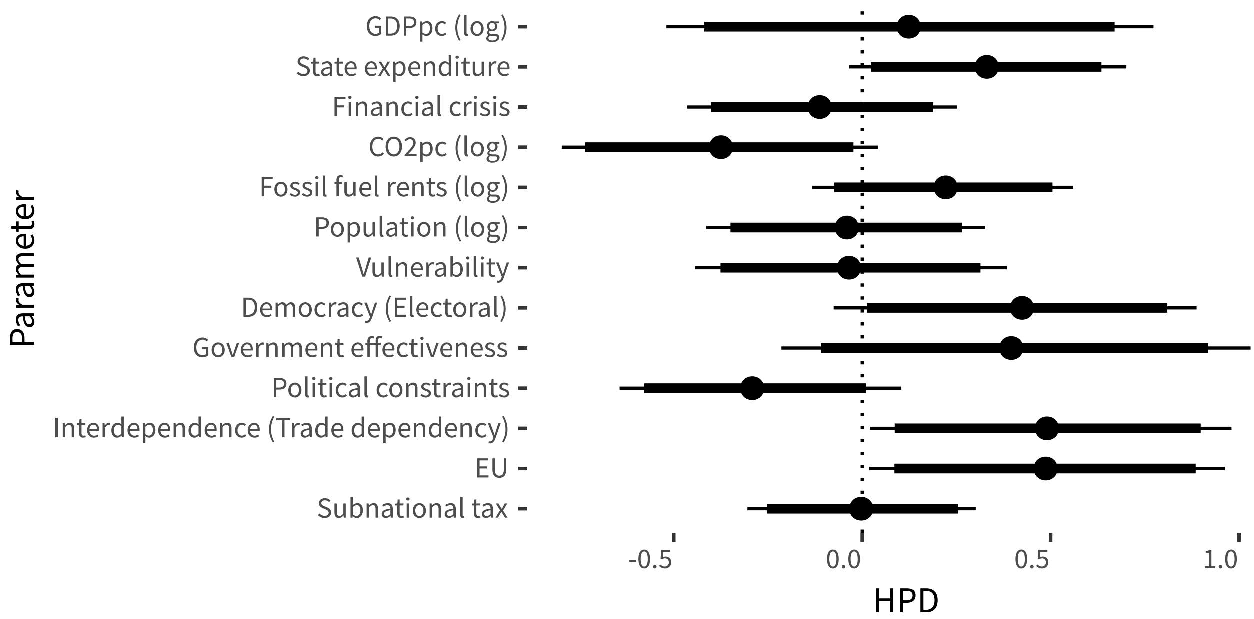


Figure 14.4: HPD of the effects of covariates on the score of carbon pricing, without the intercept and the intensity variables.

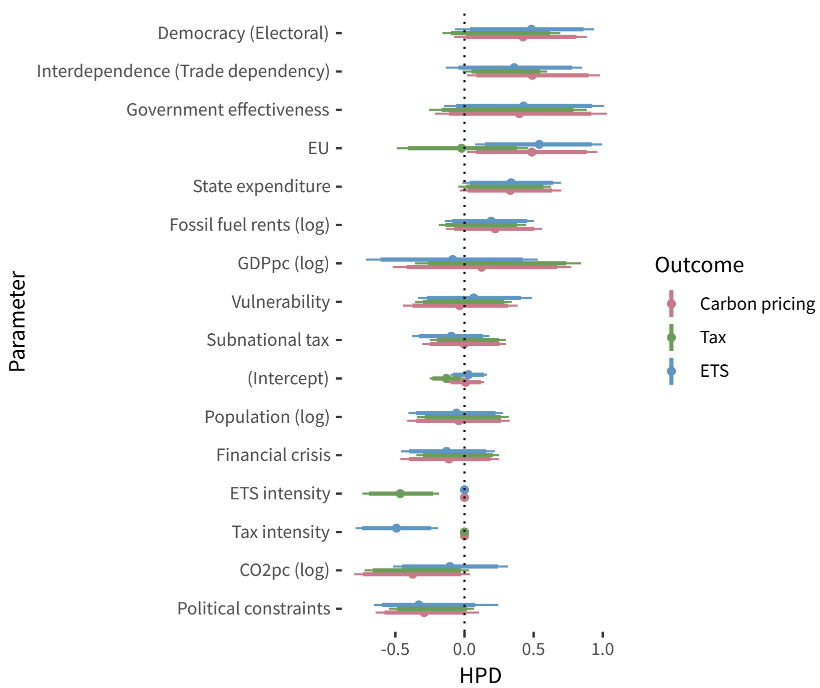


Figure 14.5: HPD of parameters, comparing outcomes.

```
ggplot(S.rsd, aes(x = `Residual SD`)) +
  geom_histogram() +
  geom_vline(data = data.sd, aes(xintercept = data.sd), lty = 1) +
  expand_limits(x = 0) +
  facet_grid(~ Outcome)
```

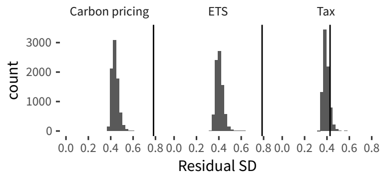


Figure 14.6: Residual standard deviation.

Model fit, pseudo-R^2

```
S.rsd %>%
  group_by(Outcome) %>%
  summarize(MedianRSD = median(`Residual SD`)) %>%
  left_join(data.sd) %>%
  mutate(PseudoR2 = 1 - (MedianRSD / data.sd))
```

```
→ # A tibble: 3 x 4
→   Outcome        MedianRSD data.sd PseudoR2
→   <chr>              <dbl>   <dbl>    <dbl>
→ 1 Carbon pricing     0.435   0.783   0.444
→ 2 Tax                0.387   0.428   0.0942
→ 3 ETS                0.403   0.787   0.487
```

Worstly predicted cases.

```
L.data <- plab("resid", list(Country = country.label, Outcome = outcome.label))
S.country <- ggs(s, family = "resid", par_labels = L.data) %>%
  group_by(Country, Outcome) %>%
  summarize(`Average residual` = mean(value))


# Manually calculate PCP
# as ggmcmc's ggs_pcp() is not ready for matrices as input for outcome

wp.countries <- S.country %>% # worstly predicted cases
  ungroup() %>%
  arrange(desc(abs(`Average residual`))) %>%
  slice(1:20)

tc <- "Worstly predicted cases, by absolute average residual."
if (knitr::is_latex_output()) {
  kable(wp.countries, format = "latex", caption = tc, longtable = TRUE, booktabs = TRUE) %>%
    kable_styling(font_size = 10)
} else {
  kable(wp.countries, format = "html", caption = tc, booktabs = TRUE) %>%
    kable_styling(font_size = 10, position = "center", bootstrap_options = "striped", full_width = F)
}
```

Table 14.1: Worstly predicted cases, by absolute average residual.

Country	Outcome	Average residual
Chile	ETS	-0.8945
Switzerland	Carbon pricing	0.8202

Chile	Carbon pricing	-0.7702
India	ETS	0.7320
Ireland	Carbon pricing	0.6947
Ireland	Tax	0.6493
Iceland	Carbon pricing	0.6374
South Korea	ETS	0.6249
Iceland	ETS	0.6079
Australia	Tax	-0.5955
Switzerland	Tax	0.5804
Slovenia	Tax	0.5602
New Zealand	Tax	-0.5549
Japan	Carbon pricing	-0.5536
Canada	Carbon pricing	0.5535
Netherlands	Carbon pricing	-0.5523
Austria	Carbon pricing	-0.5473
Albania	ETS	-0.5444
Lithuania	Tax	-0.5327
Japan	ETS	-0.5091

```r
load("carbon_pricing-patterns.RData")

scores.3 <- scores.countries.1d %>%
  left_join(select(configurations, Configuration, `ETS Adoption`, `Tax Adoption`)) %>%
  left_join(scores.countries.2d) %>%
  select(-c(Model, Configuration)) %>%
  gather(Policy, Intensity, -c(Country, `Tax Adoption`, `ETS Adoption`)) %>%
  mutate(shaded = ifelse(`ETS Adoption` == 0 & Policy == "ETS", TRUE, FALSE)) %>%
  mutate(shaded = ifelse(`Tax Adoption` == 0 & Policy == "Tax", TRUE, shaded))

ggplot(scores.3, aes(x = Intensity,
                     y = reorder(Country, Intensity),
                     color = shaded)) +
  geom_point() +
  facet_grid(~ Policy) +
  ylab("Country") +
  scale_color_manual(values = c("black", "grey")) +
  guides(color = FALSE)
```

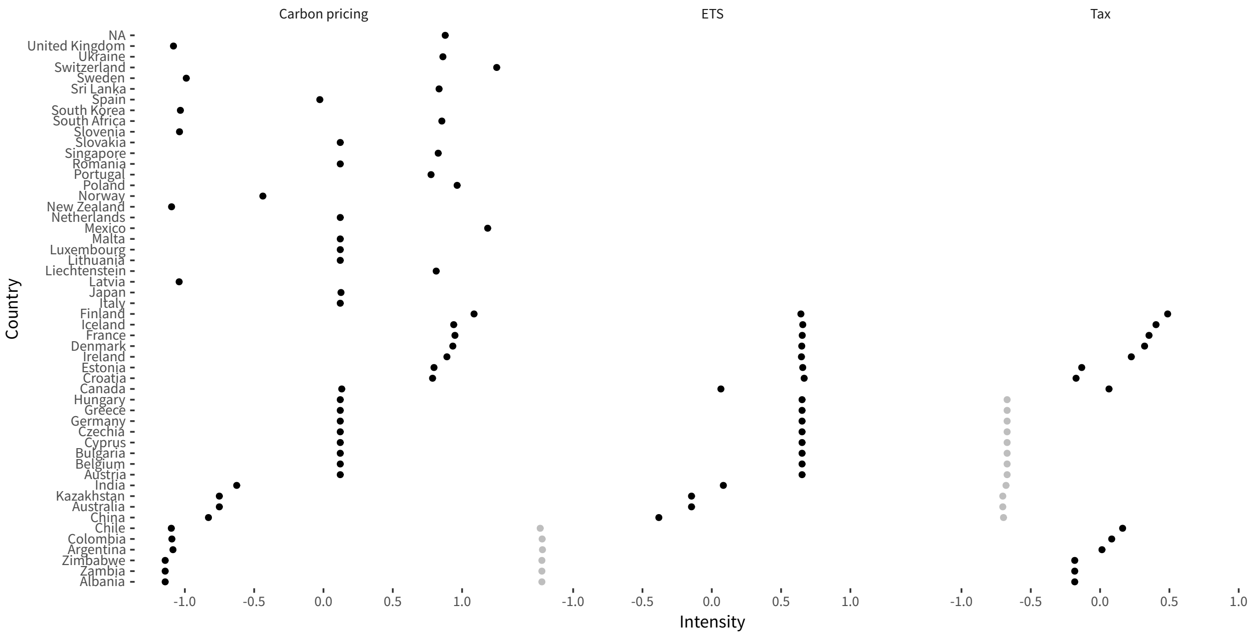


Figure 14.7: Policy intensity in carbon pricing, ETS and Tax.

15
Compare models explaining scores

```r
d <- NULL
load("samples-theta-004.RData")
d <- bind_rows(d, S.theta)
load("samples-theta-007.RData")
d <- bind_rows(d, S.theta)

d %>%
  filter(Outcome == "Carbon pricing") %>%
  filter(!Covariate %in% c("Tax intensity",
                           "ETS intensity")) %>%
  mutate(Covariate = as.character(Covariate)) %>%
  mutate(Covariate = ifelse(Covariate == "Interdependence (Trade dependency)",
                            "Trade interdependence", Covariate)) %>%
  mutate(Covariate = ifelse(Covariate == "Interdependence (Trade competition)",
                            "Trade competition", Covariate)) %>%
  ggs_caterpillar(label = "Covariate", comparison = "Model",
                  comparison_separation = 0.4) +
  aes(color = Model) +
  scale_color_manual("Model",
                     values = c("black", "grey70"),
                     breaks = c("004", "007"),
                     labels = c("Reference", "Alternative")) +
  theme(legend.position = "right")
```

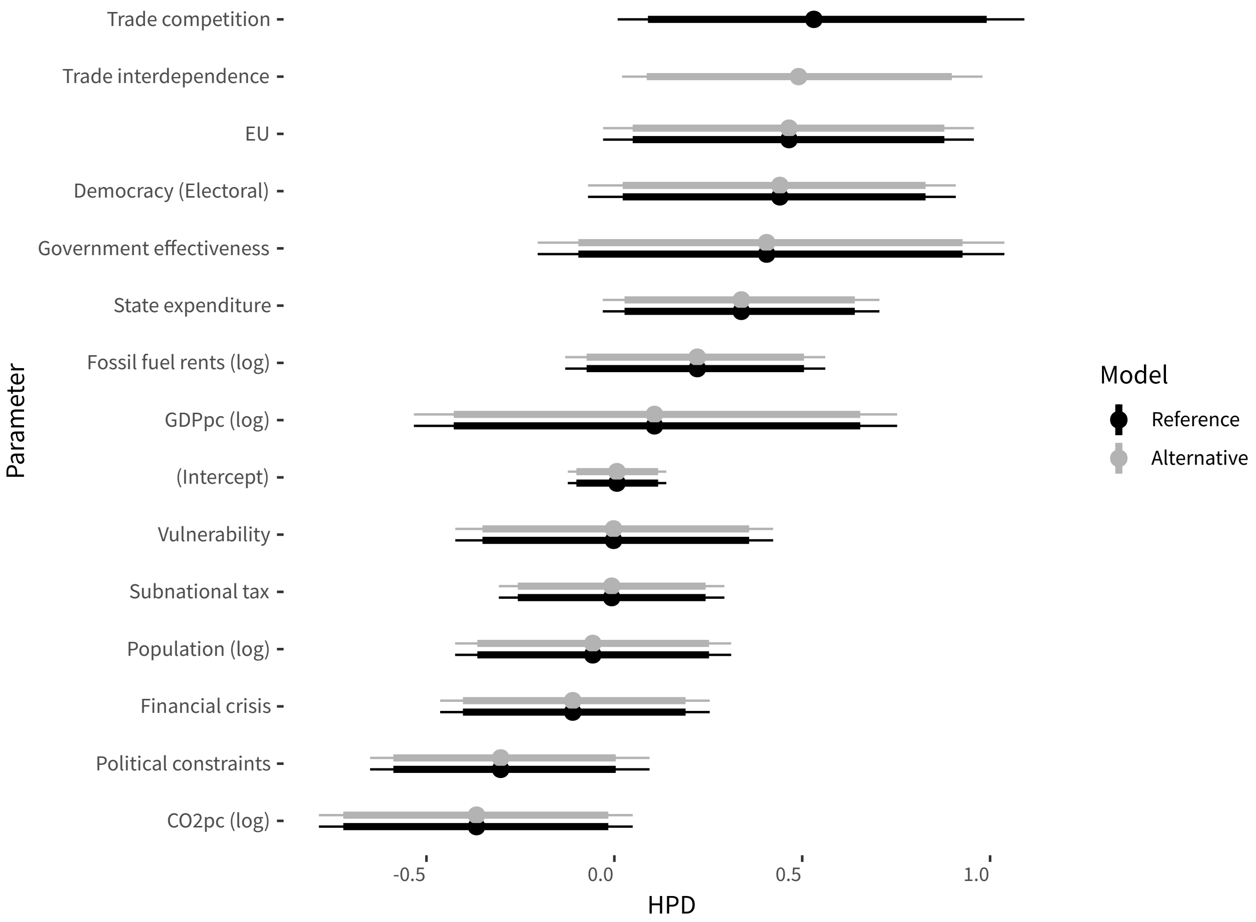


Figure 15.1: Model comparison using different trade effects specifications.

References

Gelman, Andrew. 2008. "Scaling Regression Inputs by Dividing by Two Standard Deviations." *Statistics in Medicine* 27: 2865–73. https://doi.org/10.1002/sim.3107.

Henisz, Witold J. 2002. "The Institutional Environment for Infrastructure Investment." *Industrial and Corporate Change* 11 (2): 355–89.

Jordana, Jacint, Xavier Fernández-i-Marín, and Andrea C Bianculli. 2018. "Agency Proliferation and the Globalization of the Regulatory State: Introducing a Data Set on the Institutional Features of Regulatory Agencies." *Regulation & Governance* 12 (4): 524–40. https://doi.org/10.1111/rego.12189.

Mayer, Thierry, and Soledad Zignago. 2011. "Notes on Cepii's Distances Measures: The Geodist Database."