Political Alignment in the 2016 European Social Survey

Daniel Hopkins & Rob Verbeek

8/12/2019

Contents

0. Introduction

We explore what variables predict political affiliation in European countries, and what models allow us to best make those predictions. We take a large dataset from the European Social Survey, filter and clean it, and select the best predictive variables using two different techniques. Then we fit a linear model to the data, comparing the one-predictor and five-predictor models. Finally, we fit a hierarchical (partially pooled) model to the data, so that we can preserve the distinctions between different countries, while also taking advantage of common political sentiments across Europe. Then we show that this model performs better than the pooled model at prediction and validation.

1. Data

Our data is from the European Social Survey conducted in 2016-2017. The data has more than 30,000 survey results across more than 500 variables in 21 countries. We first filtered the columns with non-spectrum (i.e. categorical) data and also filtered out columns that were country-specific or had a high number of non-answers. After filterning, we were left with nearly 22,000 observations across 105 variables.

Data filtering details

The ESS dataset is very large, with a lot of different types of values and country-specific questions. So before running our models, we cleaned our data. The python code is not show here for space reasons. Cleaning the data consists of the following steps:

Remove country-specific data:\ We want all variables to be available for all countries, so we remove country-specific data.

Remove nominal data:\ The dataset has a lot of questions that are nominal (things like job-type, political party voted for). These are filtered out, because they cannot be put on a scale.

Remove sparse questions:\ We remove the questions that have more tan 10% invalid answers (e.g. not answered, no preference, etc..). This is mainly done so that in a later stage of filtering, these questions don't cause the filtering of too many datapoints.

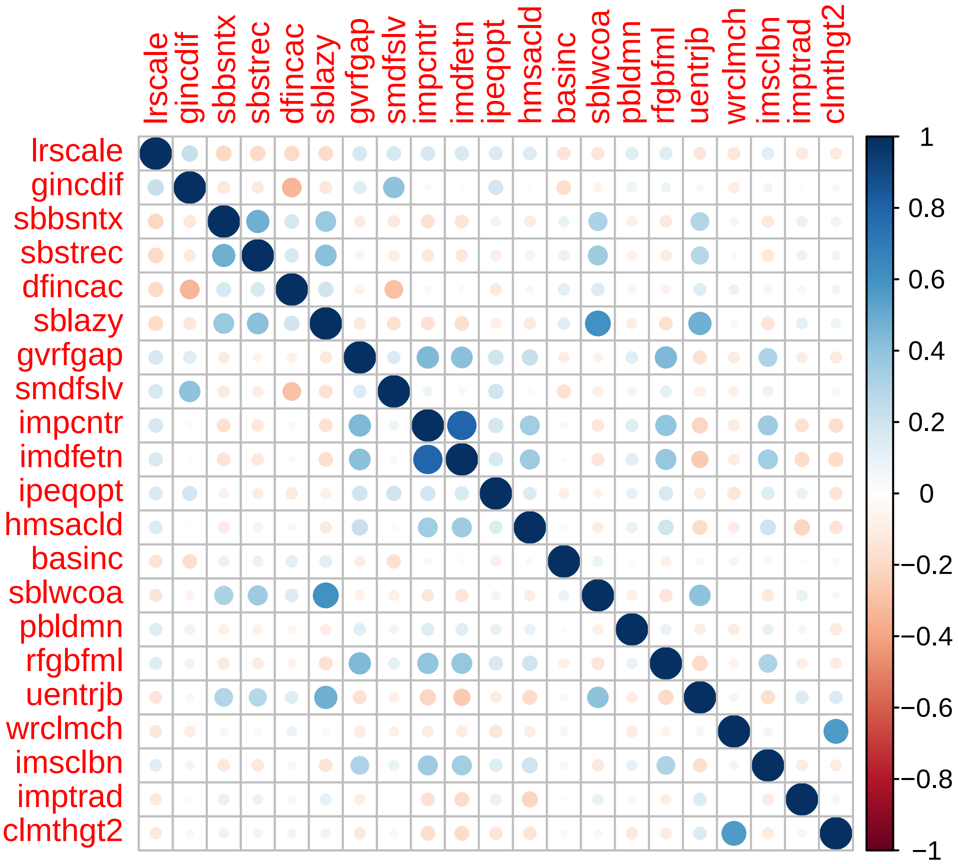
Change scales:\ The questions come in a lot of different scales, like 1-10, 1-5, 1-4 but also yes-no questions. These questions are all scaled to the 1-1- scale. For most scales this is done using a simple transformation, the yes-no questions are mapped to the values 7 and 3.

```
ESSData <- read.csv(file="data/filtered_data2.csv", header=TRUE, sep=",")

M = length(ESSData$lrscale)
country = ESSData$cntry
age = 2016 - ESSData$yrbrn
age = age/max(age)
drops = c("cntry","yrbrn")
ESSData = ESSData[,!(names(ESSData) %in% drops)] / 10
ESSData = data.frame(age,ESSData)

SAMPLESIZE = 10000

sample_rows = sample(seq(1,M),SAMPLESIZE)
test_rows = sample(seq(1,SAMPLESIZE),1000)

sample_data = data.frame(ESSData[sample_rows,])
training_data = data.frame(sample_data[-test_rows,])
test_data = data.frame(sample_data[test_rows,])
```

2. Variable Selection

We used two methods for variable selection. The first method is a simple correlation. Figure 1 plots a correlation matrix for the 20 variables that correlate best with the "lrscale". The plot shows that all of these variables have slight positive or negative correlations with the outcome variable.

```
c = cor(data.frame(sample_data))
lrscale_corr = c["lrscale",]
corr_best_variables = lrscale_corr[order(abs(lrscale_corr),decreasing=TRUE)][1:21]
corr_best_columns = order(abs(lrscale_corr),decreasing=TRUE)[1:21]
corr_best_variables = colnames(ESSData)[corr_best_columns[2:21]]
corrplot(cor(data.frame(sample_data[corr_best_columns])))
```



This is an ok selection method. *However*, it fails to account for variables that are highly correlated with eachother. For example, we can see that the variables "imdfetn" ("Allow many/few immigrants of different race/ethnic group from majority") and "impcntr" ("Allow many/few immigrants from poorer countries outside Europe") are highly correlated, so they might carry redundant information for the model. That is why the next method, variable selection using a fitted linear model and the varsel() function, is better.Instead of using the correlation matrix method for variable selection, we can fit a generalized linear model where the outcome follows all the 105 variables (excluding the country variable).

```
t_prior <- student_t(df = 5, location = 0.5, scale = 0.1)
fit_lm_all <- stan_glm(lrscale ~ ., data=sample_data, prior = t_prior, prior_intercept = t_prior, iter
summary(fit_lm_all, digits=2)
```

After fitting the model, we use the CRAN varsel() function, which uses cross-validation to pick the variables that contribute most to the model accuracy.

```
vs <- varsel(fit_lm_all, nv_max=40)
saveRDS(vs, file = "variable_selection_40_tprior.rds")
```

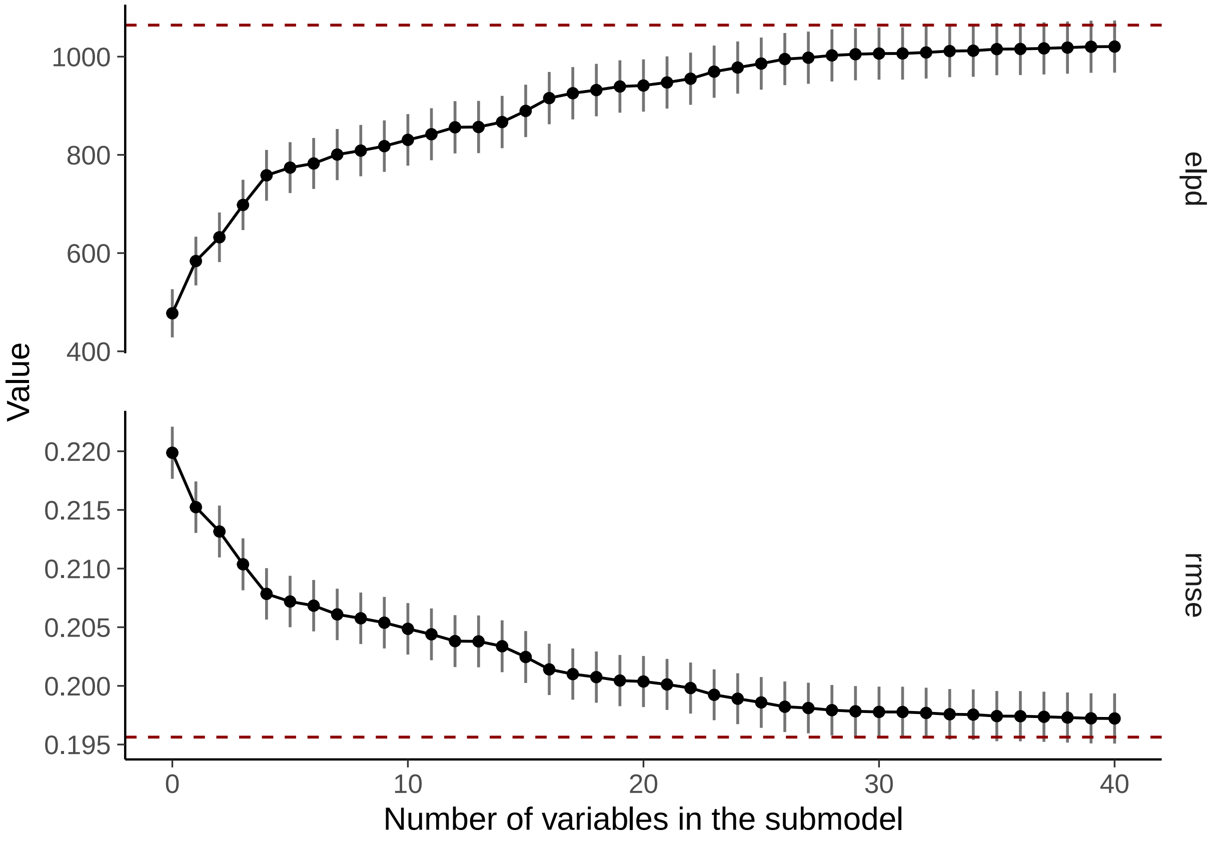
The five most explanatory variables, according to the varSel function are:

1) gincdif: "Government should reduce differences in income levels (Agree<–>Disagree)"
2) dfincac: "Large differences in income acceptable to reward talents and efforts (Agree<–>Disagree)"
3) impcntr: "Allow many/few immigrants from poorer countries outside Europe (Many<–>Few)"
4) rlgblg: "Belonging to a particular religion or denomination (Yes/No)"
5) sbsrnen: "Favour subsidize renewable energy to reduce climate change (Agree<–>Disagree)"

They are all scale questions except for the binary religion variable.

You can also see the plot, which shows the improved explanatory power of each additional variable. In this case, since the data is complex, the added power is not very great, but the model shows clear improvement across the first five variables, which slows after the fifth variable. The lower line shows decreasing error and the upper line shows an increasing elpd log-probability of the data.

```
vs = readRDS(file = "variable_selection_40_tprior.rds")
varsel_plot(vs, stats=c('elpd', 'rmse'))
```



```
varsel_best_variables = colnames(ESSData)[vs$vind[1:5]]
varsel_best_variables
```

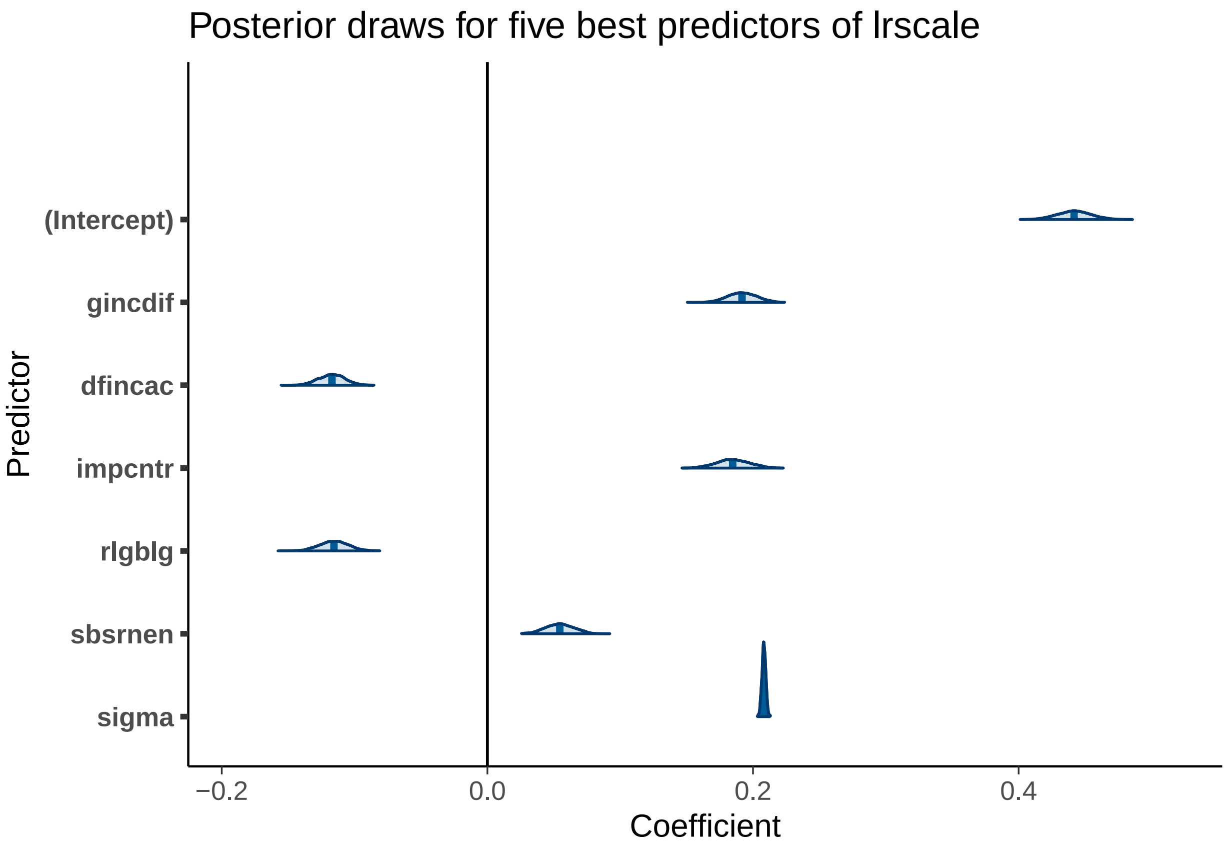
```
## [1] "dfincac" "gincdif" "rlgblg"  "impcntr" "sbsrnen"
```

```
sample_data_best_5 = cbind(sample_data$lrscale,sample_data[,vs$vind[1:5]])
```

The following plot shows the 95% distribution of posterior draws for the linear weights for each of these five variables and the bias term. Each distribution sits clearly on the positive or negative side of 0, indicating confident predictive value.

```
fit_lm_5 = stan_glm(lrscale ~ ., data=sample_data_best_5, prior = t_prior, prior_intercept = t_prior, i
```

```
pplot<-plot(fit_lm_5, "areas", prob = 0.95, prob_outer = 1)
pplot+ geom_vline(xintercept = 0) + labs(title="Posterior draws for five best predictors of lrscale",
        x ="Coefficient", y = "Predictor")
```



3. Models

Note about Stan

We used rstanarm for our stan implementations. Rstanarm has prebuilt models that take a very flexible statistical formula parameter. We use the stan_lm, stan_glm and stan_lmer functions. We used a gaussian error term for all functions, so the lm and glm functions are similar. The major difference is that the stan_lmer function allows for interactions between parameters, i.e. it can implement partially pooled (hierarchical) models. We did write stan models, similar to the hierarchical model used in Exercise 7, but ended up using the rstanarm implementation for our final report.

Note about Priors

For all models in this report, we use a weakly informative student-t prior. We choose student-t because it has fatter tails than the gaussian distribution, reflecting the greater frequency of extreme political beliefs.

Pooled model

The pooled model considers all data the same, thus not take into account the country each datapoint is from. It fits one linear model to predict the left-right scale, based on all input variables. This model was also used above after variable selection to show the slope-paramater distributions for each of the five predictors.

Below is a summary of the simple linear five-predictor pooled model fitted above.

```r
summary(fit_lm_5, digits=2)
```

```
##
## Model Info:
##  function:     stan_glm
##  family:       gaussian [identity]
##  formula:      lrscale ~ .
##  algorithm:    sampling
##  sample:       2000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 10000
##  predictors:   6
##
## Estimates:
##               mean   sd    10%   50%   90%
## (Intercept)  0.44   0.01  0.43  0.44  0.46
## gincdif      0.19   0.01  0.18  0.19  0.21
## dfincac     -0.12   0.01 -0.13 -0.12 -0.11
## impcntr      0.18   0.01  0.17  0.18  0.20
## rlgblg      -0.12   0.01 -0.13 -0.12 -0.10
## sbsrnen      0.05   0.01  0.04  0.05  0.07
## sigma        0.21   0.00  0.21  0.21  0.21
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.51   0.00 0.51  0.51  0.52
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##               mcse Rhat n_eff
## (Intercept)   0.00 1.00 2191
## gincdif       0.00 1.00 1866
## dfincac       0.00 1.00 1751
## impcntr       0.00 1.00 1957
## rlgblg        0.00 1.00 2272
## sbsrnen       0.00 1.00 2040
## sigma         0.00 1.00 2831
## mean_PPD      0.00 1.00 2205
## log-posterior 0.06 1.00  919
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

One-predictor pooled model

We also fit a one-predictor pooled model for the sake of loo comparison.

```
t_prior <- student_t(df = 5, location = 0.5, scale = 0.1)
fit_lm_1 <- stan_glm(lrscale ~ gincdif, data=sample_data_best_5, prior = t_prior, prior_intercept = t_p
```

Separate model

The seperate model simply fits a different five-predictor linear model for each of the 21 countries. We generate this model for contrast in the plot that follows.

```
sample_data_best_5 = cbind(country[sample_rows],sample_data_best_5)
colnames(sample_data_best_5)[1] = "country"

nopooled <- stan_lmer(formula = lrscale ~ 0 + (1 + gincdif + impcntr + dfincac + rlgblg + sbsrnen | cou
```

This model has six parameters for each country (five slopes and an intercept), so there are 126 r-hat values. Instead of printing all of these, following is the mean of the r-hat values, which is very close to 1.

```
mean(rhat(nopooled))
```

```
## [1] 1.000743
```

Here are the slope and intercept parameters for each country for the non-pooled model.

```
coef(nopooled)
```

```
## $country
##    (Intercept)     gincdif     impcntr     dfincac      rlgblg      sbsrnen
## BE   0.4684797  0.14620275  0.15778149 -0.14952079 -0.08128349  0.029814836
## CH   0.3483624  0.20923357  0.35088768 -0.15282690 -0.12796376  0.128411509
## CZ   0.4492845  0.29931151 -0.15030580 -0.09147635  0.05939828 -0.005872177
## DE   0.3571158  0.11626325  0.26508461 -0.08383327 -0.10209357  0.047526274
## EE   0.4809279  0.19360042 -0.01031825 -0.07806491  0.03559833  0.004365258
## ES   0.4996440  0.12548028  0.21680667 -0.17190492 -0.21293460  0.040624581
## FI   0.5191972  0.23716300  0.18245982 -0.16574624 -0.11365187  0.064791504
## FR   0.4070906  0.20463416  0.35841397 -0.12462205 -0.18426137  0.054854397
## GB   0.4137925  0.22663623  0.15853187 -0.19224782 -0.07990139  0.106007759
## IE   0.5531354  0.09768450  0.03500348 -0.12880890 -0.12004328  0.044510279
## IL   0.4324232  0.07942658  0.15996784  0.05190296 -0.10404624  0.054398782
## IS   0.5239602  0.22504884  0.22722246 -0.21265979 -0.13734304  0.102040866
## IT   0.3656845  0.10853496  0.33093389 -0.01471195 -0.14570388  0.012186580
## LT   0.5459543  0.14481806 -0.01742677 -0.09819437 -0.06798947  0.044269867
## NL   0.4676184  0.19086991  0.22379678 -0.18380980 -0.11660467  0.099329350
## NO   0.3962856  0.27489023  0.24509687 -0.22770588 -0.05377324  0.136561404
## PL   0.5031760 -0.01378706  0.31697635 -0.01401876 -0.26576135  0.051323730
## PT   0.4637441  0.16656464  0.11785513 -0.10078860 -0.12241459  0.029837841
## RU   0.5334035  0.12587511 -0.05525453 -0.08205701 -0.03599867  0.039400076
## SE   0.4683706  0.28137839  0.30973800 -0.27072163 -0.13083274  0.131092059
## SI   0.4375140  0.05541477  0.26943558 -0.03876636 -0.22393099  0.018697198
##
## attr(,"class")
## [1] "coef.mer"
```

Hierarchical model

The previous model is pooled, assuming there is no differences between countries. The partially pooled model allows for differentation between countries. Each country will have it's own intecept and slope per variable, but these parameters are fitted to an overall distribution. This still allows the model to generalize over the entire dataset, while being able to fit a more narrow distribution on each country individually.

```
fit_hier_2 <- stan_lmer(lrscale ~ 1 + gincdif + impcntr + dfincac + rlgblg + sbsrnen + (1 + gincdif + i
```

We apologize to the reader, the following summary has many coefficients, but we want to show the r-hat and n_eff values for this final model.

```r
summary(fit_hier_2, digits=2)
```

```
##
## Model Info:
##  function:     stan_lmer
##  family:       gaussian [identity]
##  formula:      lrscale ~ 1 + gincdif + impcntr + dfincac + rlgblg + sbsrnen +
##     (1 + gincdif + impcntr + dfincac + rlgblg + sbsrnen | country)
##  algorithm:    sampling
##  sample:       2000 (posterior sample size)
##  priors:       see help('prior_summary')
##  observations: 10000
##  groups:       country (21)
##
## Estimates:
##                                          mean   sd    10%   50%   90%
## (Intercept)                             0.46   0.02  0.44  0.46  0.48
## gincdif                                 0.17   0.02  0.14  0.17  0.20
## impcntr                                 0.18   0.03  0.14  0.18  0.22
## dfincac                                -0.12   0.02 -0.15 -0.12 -0.09
## rlgblg                                 -0.12   0.02 -0.15 -0.12 -0.09
## sbsrnen                                 0.06   0.02  0.04  0.06  0.08
## b[(Intercept) country:BE]               0.00   0.03 -0.04  0.00  0.05
## b[gincdif country:BE]                  -0.02   0.04 -0.07 -0.02  0.03
## b[impcntr country:BE]                  -0.01   0.05 -0.07 -0.01  0.05
## b[dfincac country:BE]                  -0.03   0.04 -0.07 -0.02  0.02
## b[rlgblg country:BE]                    0.04   0.04 -0.01  0.03  0.08
## b[sbsrnen country:BE]                  -0.03   0.03 -0.08 -0.03  0.01
## b[(Intercept) country:CH]              -0.07   0.05 -0.13 -0.06 -0.01
## b[gincdif country:CH]                   0.03   0.04 -0.03  0.02  0.08
## b[impcntr country:CH]                   0.15   0.06  0.08  0.15  0.22
## b[dfincac country:CH]                  -0.05   0.04 -0.11 -0.05  0.01
## b[rlgblg country:CH]                   -0.02   0.04 -0.08 -0.02  0.03
## b[sbsrnen country:CH]                   0.05   0.04  0.00  0.05  0.11
## b[(Intercept) country:CZ]               0.01   0.05 -0.05  0.01  0.07
## b[gincdif country:CZ]                   0.13   0.04  0.07  0.13  0.18
## b[impcntr country:CZ]                  -0.32   0.06 -0.39 -0.32 -0.25
## b[dfincac country:CZ]                   0.03   0.04 -0.02  0.03  0.08
## b[rlgblg country:CZ]                    0.15   0.05  0.08  0.15  0.21
## b[sbsrnen country:CZ]                  -0.07   0.03 -0.11 -0.07 -0.03
## b[(Intercept) country:DE]              -0.08   0.04 -0.12 -0.08 -0.03
## b[gincdif country:DE]                  -0.06   0.04 -0.12 -0.06 -0.02
## b[impcntr country:DE]                   0.08   0.05  0.03  0.08  0.14
## b[dfincac country:DE]                   0.03   0.04 -0.02  0.03  0.07
## b[rlgblg country:DE]                    0.01   0.04 -0.04  0.01  0.05
## b[sbsrnen country:DE]                  -0.03   0.03 -0.07 -0.03  0.01
## b[(Intercept) country:EE]               0.02   0.04 -0.04  0.02  0.07
## b[gincdif country:EE]                   0.02   0.04 -0.03  0.03  0.08
## b[impcntr country:EE]                  -0.18   0.05 -0.24 -0.17 -0.11
## b[dfincac country:EE]                   0.04   0.04 -0.02  0.04  0.09
## b[rlgblg country:EE]                    0.14   0.04  0.09  0.14  0.20
## b[sbsrnen country:EE]                  -0.05   0.04 -0.09 -0.05  0.00
## b[(Intercept) country:ES]               0.02   0.05 -0.04  0.02  0.08
## b[gincdif country:ES]                  -0.03   0.05 -0.09 -0.03  0.04
## b[impcntr country:ES]                   0.05   0.06 -0.03  0.05  0.13
## b[dfincac country:ES]                  -0.04   0.05 -0.10 -0.04  0.02
## b[rlgblg country:ES]                   -0.08   0.05 -0.15 -0.08 -0.03
## b[sbsrnen country:ES]                  -0.03   0.04 -0.07 -0.02  0.02
## b[(Intercept) country:FI]               0.04   0.04 -0.01  0.04  0.09
## b[gincdif country:FI]                   0.07   0.04  0.02  0.07  0.12
## b[impcntr country:FI]                   0.00   0.05 -0.06  0.01  0.07
## b[dfincac country:FI]                  -0.03   0.04 -0.08 -0.03  0.01
## b[rlgblg country:FI]                    0.00   0.04 -0.04  0.01  0.05
## b[sbsrnen country:FI]                   0.02   0.03 -0.02  0.02  0.06
## b[(Intercept) country:FR]              -0.03   0.04 -0.08 -0.03  0.01
## b[gincdif country:FR]                   0.03   0.04 -0.02  0.03  0.08
## b[impcntr country:FR]                   0.16   0.05  0.10  0.16  0.23
## b[dfincac country:FR]                  -0.01   0.04 -0.06 -0.01  0.04
## b[rlgblg country:FR]                   -0.07   0.04 -0.12 -0.07 -0.02
## b[sbsrnen country:FR]                   0.00   0.04 -0.05  0.00  0.04
## b[(Intercept) country:GB]              -0.03   0.04 -0.08 -0.03  0.01
## b[gincdif country:GB]                   0.05   0.04  0.00  0.05  0.10
## b[impcntr country:GB]                  -0.02   0.05 -0.09 -0.02  0.05
## b[dfincac country:GB]                  -0.07   0.04 -0.12 -0.07 -0.02
## b[rlgblg country:GB]                    0.03   0.04 -0.02  0.03  0.08
## b[sbsrnen country:GB]                   0.03   0.03 -0.01  0.03  0.07
## b[(Intercept) country:IE]               0.06   0.04  0.01  0.06  0.12
## b[gincdif country:IE]                  -0.05   0.04 -0.11 -0.05  0.00
## b[impcntr country:IE]                  -0.12   0.05 -0.19 -0.12 -0.05
## b[dfincac country:IE]                   0.01   0.04 -0.04  0.01  0.07
## b[rlgblg country:IE]                    0.00   0.04 -0.05  0.00  0.06
## b[sbsrnen country:IE]                  -0.01   0.03 -0.05 -0.01  0.03
## b[(Intercept) country:IL]              -0.01   0.04 -0.06 -0.01  0.04
## b[gincdif country:IL]                  -0.10   0.05 -0.15 -0.10 -0.04
## b[impcntr country:IL]                  -0.02   0.05 -0.09 -0.02  0.05
## b[dfincac country:IL]                   0.16   0.04  0.11  0.16  0.22
## b[rlgblg country:IL]                   -0.02   0.07 -0.11 -0.02  0.06
## b[sbsrnen country:IL]                   0.00   0.04 -0.05  0.00  0.04
## b[(Intercept) country:IS]               0.04   0.04 -0.02  0.04  0.10
## b[gincdif country:IS]                   0.06   0.05  0.00  0.06  0.13
## b[impcntr country:IS]                   0.04   0.06 -0.04  0.04  0.12
## b[dfincac country:IS]                  -0.07   0.05 -0.13 -0.07 -0.01
## b[rlgblg country:IS]                   -0.01   0.05 -0.07 -0.01  0.05
## b[sbsrnen country:IS]                   0.06   0.04  0.01  0.06  0.11
## b[(Intercept) country:IT]              -0.06   0.05 -0.12 -0.05  0.00
## b[gincdif country:IT]                  -0.08   0.06 -0.15 -0.07 -0.01
## b[impcntr country:IT]                   0.13   0.06  0.05  0.13  0.21
## b[dfincac country:IT]                   0.09   0.05  0.03  0.09  0.15
## b[rlgblg country:IT]                   -0.04   0.05 -0.10 -0.04  0.02
## b[sbsrnen country:IT]                  -0.05   0.04 -0.11 -0.05  0.00
## b[(Intercept) country:LT]               0.05   0.05  0.00  0.05  0.11
## b[gincdif country:LT]                  -0.01   0.06 -0.08 -0.01  0.06
## b[impcntr country:LT]                  -0.16   0.07 -0.24 -0.16 -0.08
## b[dfincac country:LT]                   0.03   0.05 -0.03  0.03  0.10
## b[rlgblg country:LT]                    0.03   0.06 -0.04  0.04  0.10
## b[sbsrnen country:LT]                  -0.01   0.04 -0.06 -0.01  0.05
## b[(Intercept) country:NL]               0.00   0.04 -0.05  0.00  0.05
## b[gincdif country:NL]                   0.02   0.04 -0.03  0.02  0.07
## b[impcntr country:NL]                    0.05   0.05 -0.02  0.05  0.12
## b[dfincac country:NL]                   -0.06   0.04 -0.11 -0.06 -0.01
## b[rlgblg country:NL]                     0.00   0.04 -0.05  0.00  0.05
## b[sbsrnen country:NL]                    0.04   0.04  0.00  0.04  0.09
## b[(Intercept) country:NO]               -0.04   0.04 -0.09 -0.03  0.01
## b[gincdif country:NO]                    0.09   0.04  0.04  0.09  0.15
## b[impcntr country:NO]                    0.06   0.06 -0.01  0.06  0.13
## b[dfincac country:NO]                   -0.11   0.04 -0.17 -0.11 -0.06
## b[rlgblg country:NO]                     0.05   0.04  0.00  0.05  0.10
## b[sbsrnen country:NO]                    0.07   0.04  0.02  0.07  0.12
## b[(Intercept) country:PL]                0.03   0.04 -0.03  0.02  0.08
## b[gincdif country:PL]                   -0.17   0.05 -0.23 -0.17 -0.10
## b[impcntr country:PL]                    0.14   0.06  0.06  0.14  0.22
## b[dfincac country:PL]                    0.11   0.05  0.05  0.11  0.18
## b[rlgblg country:PL]                    -0.14   0.06 -0.22 -0.14 -0.07
## b[sbsrnen country:PL]                    0.01   0.04 -0.05  0.01  0.06
## b[(Intercept) country:PT]                0.00   0.04 -0.04  0.00  0.05
## b[gincdif country:PT]                    0.01   0.05 -0.05  0.01  0.07
## b[impcntr country:PT]                   -0.05   0.06 -0.14 -0.05  0.02
## b[dfincac country:PT]                    0.02   0.04 -0.03  0.02  0.08
## b[rlgblg country:PT]                    -0.01   0.04 -0.07 -0.01  0.04
## b[sbsrnen country:PT]                   -0.03   0.04 -0.08 -0.03  0.01
## b[(Intercept) country:RU]                0.04   0.05 -0.02  0.04  0.10
## b[gincdif country:RU]                   -0.03   0.05 -0.09 -0.03  0.04
## b[impcntr country:RU]                   -0.20   0.07 -0.28 -0.20 -0.11
## b[dfincac country:RU]                    0.05   0.05 -0.02  0.05  0.12
## b[rlgblg country:RU]                     0.07   0.05  0.01  0.07  0.14
## b[sbsrnen country:RU]                   -0.02   0.04 -0.07 -0.02  0.04
## b[(Intercept) country:SE]                0.01   0.04 -0.04  0.01  0.06
## b[gincdif country:SE]                    0.11   0.05  0.05  0.11  0.18
## b[impcntr country:SE]                    0.12   0.06  0.04  0.12  0.19
## b[dfincac country:SE]                   -0.15   0.04 -0.20 -0.14 -0.09
## b[rlgblg country:SE]                    -0.01   0.04 -0.06 -0.01  0.04
## b[sbsrnen country:SE]                    0.08   0.04  0.03  0.08  0.13
## b[(Intercept) country:SI]               -0.01   0.04 -0.07 -0.01  0.04
## b[gincdif country:SI]                   -0.10   0.05 -0.17 -0.10 -0.04
## b[impcntr country:SI]                    0.08   0.06  0.00  0.08  0.16
## b[dfincac country:SI]                    0.08   0.05  0.01  0.08  0.14
## b[rlgblg country:SI]                    -0.11   0.05 -0.17 -0.11 -0.05
## b[sbsrnen country:SI]                   -0.05   0.05 -0.11 -0.04  0.01
## sigma                                    0.20   0.00  0.20  0.20  0.20
## Sigma[country:(Intercept),(Intercept)]   0.00   0.00  0.00  0.00  0.01
## Sigma[country:gincdif,(Intercept)]       0.00   0.00  0.00  0.00  0.00
## Sigma[country:impcntr,(Intercept)]       0.00   0.00  0.00  0.00  0.00
## Sigma[country:dfincac,(Intercept)]       0.00   0.00  0.00  0.00  0.00
## Sigma[country:rlgblg,(Intercept)]        0.00   0.00  0.00  0.00  0.00
## Sigma[country:sbsrnen,(Intercept)]       0.00   0.00  0.00  0.00  0.00
## Sigma[country:gincdif,gincdif]           0.01   0.00  0.01  0.01  0.02
## Sigma[country:impcntr,gincdif]           0.00   0.00  0.00  0.00  0.00
## Sigma[country:dfincac,gincdif]          -0.01   0.00 -0.01 -0.01  0.00
## Sigma[country:rlgblg,gincdif]            0.00   0.00  0.00  0.00  0.01
## Sigma[country:sbsrnen,gincdif]           0.00   0.00  0.00  0.00  0.00
## Sigma[country:impcntr,impcntr]           0.02   0.00  0.01  0.01  0.02
## Sigma[country:dfincac,impcntr]           0.00   0.00 -0.01  0.00  0.00
## Sigma[country:rlgblg,impcntr]           0.00   0.00 -0.01  0.00  0.00
## Sigma[country:sbsrnen,impcntr]          0.00   0.00  0.00  0.00  0.00
## Sigma[country:dfincac,dfincac]          0.01   0.00  0.01  0.01  0.02
## Sigma[country:rlgblg,dfincac]           0.00   0.00  0.00  0.00  0.00
## Sigma[country:sbsrnen,dfincac]          0.00   0.00 -0.01  0.00  0.00
## Sigma[country:rlgblg,rlgblg]            0.01   0.00  0.00  0.01  0.01
## Sigma[country:sbsrnen,rlgblg]           0.00   0.00  0.00  0.00  0.00
## Sigma[country:sbsrnen,sbsrnen]          0.00   0.00  0.00  0.00  0.01
##
## Fit Diagnostics:
##            mean   sd   10%   50%   90%
## mean_PPD 0.51   0.00 0.51  0.51  0.52
##
## The mean_ppd is the sample average posterior predictive distribution of the outcome variable (for de
##
## MCMC diagnostics
##                                         mcse Rhat n_eff
## (Intercept)                             0.00 1.00 1588
## gincdif                                 0.00 1.00  765
## impcntr                                 0.00 1.00  814
## dfincac                                 0.00 1.01  710
## rlgblg                                  0.00 1.00 1207
## sbsrnen                                 0.00 1.00 1165
## b[(Intercept) country:BE]               0.00 1.00 1535
## b[gincdif country:BE]                   0.00 1.00 1121
## b[impcntr country:BE]                   0.00 1.00 1025
## b[dfincac country:BE]                   0.00 1.00  969
## b[rlgblg country:BE]                    0.00 1.00 1690
## b[sbsrnen country:BE]                   0.00 1.00 1677
## b[(Intercept) country:CH]               0.00 1.00 1397
## b[gincdif country:CH]                   0.00 1.00 1571
## b[impcntr country:CH]                   0.00 1.00 1633
## b[dfincac country:CH]                   0.00 1.00 1231
## b[rlgblg country:CH]                    0.00 1.00 1738
## b[sbsrnen country:CH]                   0.00 1.00 1712
## b[(Intercept) country:CZ]               0.00 1.00 1069
## b[gincdif country:CZ]                   0.00 1.00  980
## b[impcntr country:CZ]                   0.00 1.00 1466
## b[dfincac country:CZ]                   0.00 1.00 1291
## b[rlgblg country:CZ]                    0.00 1.00 1236
## b[sbsrnen country:CZ]                   0.00 1.00 1527
## b[(Intercept) country:DE]               0.00 1.00 1377
## b[gincdif country:DE]                   0.00 1.00 1341
## b[impcntr country:DE]                   0.00 1.00 1277
## b[dfincac country:DE]                   0.00 1.00 1058
## b[rlgblg country:DE]                    0.00 1.00 1635
## b[sbsrnen country:DE]                   0.00 1.00 1447
## b[(Intercept) country:EE]               0.00 1.00  991
## b[gincdif country:EE]                   0.00 1.00 1016
## b[impcntr country:EE]                   0.00 1.00 1409
## b[dfincac country:EE]                   0.00 1.00  912
## b[rlgblg country:EE]                    0.00 1.00 1745
## b[sbsrnen country:EE]                   0.00 1.00 1640
## b[(Intercept) country:ES]               0.00 1.00  883
## b[gincdif country:ES]                  0.00 1.00  857
## b[impcntr country:ES]                  0.00 1.00 1409
## b[dfincac country:ES]                  0.00 1.00 1166
## b[rlgblg country:ES]                   0.00 1.00 1393
## b[sbsrnen country:ES]                  0.00 1.00 1486
## b[(Intercept) country:FI]              0.00 1.00 1371
## b[gincdif country:FI]                  0.00 1.00 1079
## b[impcntr country:FI]                  0.00 1.00 1415
## b[dfincac country:FI]                  0.00 1.01  998
## b[rlgblg country:FI]                   0.00 1.00 1958
## b[sbsrnen country:FI]                  0.00 1.00 1860
## b[(Intercept) country:FR]              0.00 1.00 1456
## b[gincdif country:FR]                  0.00 1.00 1154
## b[impcntr country:FR]                  0.00 1.00 1236
## b[dfincac country:FR]                  0.00 1.00 1077
## b[rlgblg country:FR]                   0.00 1.00 1712
## b[sbsrnen country:FR]                  0.00 1.00 1787
## b[(Intercept) country:GB]              0.00 1.00 1718
## b[gincdif country:GB]                  0.00 1.00 1192
## b[impcntr country:GB]                  0.00 1.00 1535
## b[dfincac country:GB]                  0.00 1.00 1298
## b[rlgblg country:GB]                   0.00 1.00 1701
## b[sbsrnen country:GB]                  0.00 1.00 1466
## b[(Intercept) country:IE]              0.00 1.00 1182
## b[gincdif country:IE]                  0.00 1.00 1297
## b[impcntr country:IE]                  0.00 1.00 1229
## b[dfincac country:IE]                  0.00 1.00 1328
## b[rlgblg country:IE]                   0.00 1.00 1378
## b[sbsrnen country:IE]                  0.00 1.00 1461
## b[(Intercept) country:IL]              0.00 1.00 1182
## b[gincdif country:IL]                  0.00 1.00 1395
## b[impcntr country:IL]                  0.00 1.00 1689
## b[dfincac country:IL]                  0.00 1.00 1103
## b[rlgblg country:IL]                   0.00 1.00 1209
## b[sbsrnen country:IL]                  0.00 1.00 1501
## b[(Intercept) country:IS]              0.00 1.00 1552
## b[gincdif country:IS]                  0.00 1.00 1381
## b[impcntr country:IS]                  0.00 1.00 1761
## b[dfincac country:IS]                  0.00 1.00 1320
## b[rlgblg country:IS]                   0.00 1.00 2056
## b[sbsrnen country:IS]                  0.00 1.00 1742
## b[(Intercept) country:IT]              0.00 1.00 1380
## b[gincdif country:IT]                  0.00 1.00 1498
## b[impcntr country:IT]                  0.00 1.00 1956
## b[dfincac country:IT]                  0.00 1.00 1425
## b[rlgblg country:IT]                   0.00 1.00 2003
## b[sbsrnen country:IT]                  0.00 1.00 1959
## b[(Intercept) country:LT]              0.00 1.00 1638
## b[gincdif country:LT]                  0.00 1.00 1819
## b[impcntr country:LT]                  0.00 1.00 2003
## b[dfincac country:LT]                  0.00 1.00 1737
## b[rlgblg country:LT]                   0.00 1.00 2058
## b[sbsrnen country:LT]                  0.00 1.00 1874
## b[(Intercept) country:NL]              0.00 1.00 1573
## b[gincdif country:NL]                  0.00 1.00 1086
## b[impcntr country:NL]                  0.00 1.00 1702
## b[dfincac country:NL]                  0.00 1.00 1240
## b[rlgblg country:NL]                   0.00 1.00 1992
## b[sbsrnen country:NL]                  0.00 1.00 1879
## b[(Intercept) country:NO]              0.00 1.01 1381
## b[gincdif country:NO]                  0.00 1.00 1119
## b[impcntr country:NO]                  0.00 1.00 1214
## b[dfincac country:NO]                  0.00 1.01  950
## b[rlgblg country:NO]                   0.00 1.00 1763
## b[sbsrnen country:NO]                  0.00 1.00 1664
## b[(Intercept) country:PL]              0.00 1.00 1128
## b[gincdif country:PL]                  0.00 1.00 1489
## b[impcntr country:PL]                  0.00 1.00 1838
## b[dfincac country:PL]                  0.00 1.00 1193
## b[rlgblg country:PL]                   0.00 1.00 1568
## b[sbsrnen country:PL]                  0.00 1.00 1847
## b[(Intercept) country:PT]              0.00 1.00 1639
## b[gincdif country:PT]                  0.00 1.00 1348
## b[impcntr country:PT]                  0.00 1.00 1822
## b[dfincac country:PT]                  0.00 1.00 1294
## b[rlgblg country:PT]                   0.00 1.00 1641
## b[sbsrnen country:PT]                  0.00 1.00 1755
## b[(Intercept) country:RU]              0.00 1.00 1749
## b[gincdif country:RU]                  0.00 1.00 1531
## b[impcntr country:RU]                  0.00 1.00 1903
## b[dfincac country:RU]                  0.00 1.00 1810
## b[rlgblg country:RU]                   0.00 1.00 2180
## b[sbsrnen country:RU]                  0.00 1.00 1524
## b[(Intercept) country:SE]              0.00 1.00 1063
## b[gincdif country:SE]                  0.00 1.00 1173
## b[impcntr country:SE]                  0.00 1.00 1716
## b[dfincac country:SE]                  0.00 1.00 1263
## b[rlgblg country:SE]                   0.00 1.00 2020
## b[sbsrnen country:SE]                  0.00 1.00 1588
## b[(Intercept) country:SI]              0.00 1.00 1226
## b[gincdif country:SI]                  0.00 1.00 1316
## b[impcntr country:SI]                  0.00 1.00 1649
## b[dfincac country:SI]                  0.00 1.00 1406
## b[rlgblg country:SI]                   0.00 1.00 1693
## b[sbsrnen country:SI]                  0.00 1.00 1850
## sigma                                  0.00 1.00 2554
## Sigma[country:(Intercept),(Intercept)] 0.00 1.00  806
## Sigma[country:gincdif,(Intercept)]     0.00 1.01  416
## Sigma[country:impcntr,(Intercept)]     0.00 1.00  705
## Sigma[country:dfincac,(Intercept)]     0.00 1.01  408
## Sigma[country:rlgblg,(Intercept)]      0.00 1.00  602
## Sigma[country:sbsrnen,(Intercept)]     0.00 1.00  889
## Sigma[country:gincdif,gincdif]         0.00 1.01  623
## Sigma[country:impcntr,gincdif]         0.00 1.00 1140
## Sigma[country:dfincac,gincdif]         0.00 1.00  651
## Sigma[country:rlgblg,gincdif]          0.00 1.00 1007
## Sigma[country:sbsrnen,gincdif]         0.00 1.00  867
## Sigma[country:impcntr,impcntr]         0.00 1.00 1089
## Sigma[country:dfincac,impcntr]         0.00 1.00 1069
## Sigma[country:rlgblg,impcntr]          0.00 1.00 1061
## Sigma[country:sbsrnen,impcntr]         0.00 1.00 1223
## Sigma[country:dfincac,dfincac]         0.00 1.00  733
## Sigma[country:rlgblg,dfincac]          0.00 1.00 1161
## Sigma[country:sbsrnen,dfincac]         0.00 1.00  685
## Sigma[country:rlgblg,rlgblg]           0.00 1.00 1003
## Sigma[country:sbsrnen,rlgblg]          0.00 1.00  993
## Sigma[country:sbsrnen,sbsrnen]         0.00 1.00  878
## mean_PPD                               0.00 1.00 2115
## log-posterior                          0.70 1.01  319
##
## For each parameter, mcse is Monte Carlo standard error, n_eff is a crude measure of effective sample
```

Following are the slope and intercept parameters for each country in the hierarchical model, displayed more concisely. Note that they generally agree at least about the direction of the correlation, except in a few interesting cases.

```
coefficients <- coef(fit_hier_2)[1]
coefficients
```

```
## $country
##    (Intercept)     gincdif      impcntr     dfincac      rlgblg     sbsrnen
## BE   0.4607673 0.150095318  0.168809891 -0.14745119 -0.08450281  0.02721378
## CH   0.3953852 0.193444551  0.327109689 -0.17146160 -0.14283000  0.11124577
## CZ   0.4674733 0.294617997 -0.138202613 -0.09435103  0.02668550 -0.01115184
## DE   0.3820177 0.106070950  0.261999576 -0.09618109 -0.11148146  0.03446064
## EE   0.4759900 0.195022369  0.005143797 -0.08362406  0.02388162  0.01426966
## ES   0.4759366 0.144004180  0.229543800 -0.16072611 -0.20127634  0.03687233
## FI   0.4973119 0.240034900  0.185174994 -0.15602706 -0.11393664  0.08272701
## FR   0.4255124 0.196516377  0.339208112 -0.13516375 -0.18878076  0.05960990
## GB   0.4295218 0.221873260  0.162618067 -0.19188965 -0.09124526  0.08987253
## IE   0.5188940 0.114535279  0.060801626 -0.10960793 -0.11680342  0.04873840
## IL   0.4502956 0.071813295  0.160484370  0.03971054 -0.14275873  0.05712728
## IS   0.4971450 0.232188014  0.219347616 -0.19685014 -0.13107939  0.12112143
## IT   0.4051312 0.095237220  0.305427888 -0.03670170 -0.16152266  0.01231931
## LT   0.5074493 0.163440268  0.020789678 -0.08822995 -0.08253097  0.05293923
## NL   0.4633320 0.192795775  0.227424883 -0.18165940 -0.12125820  0.10344121
## NO   0.4245345 0.260779307  0.239866665 -0.23772130 -0.07109477  0.12857349
## PL   0.4827863 0.003952157  0.319761818 -0.01199575 -0.25883894  0.06653449
## PT   0.4612851 0.174984325  0.128018690 -0.10329189 -0.13011996  0.02776008
## RU   0.4970951 0.143684275 -0.018714636 -0.07432036 -0.04617073  0.04400025
## SE   0.4661337 0.280170034  0.296390488 -0.26725168 -0.12744197  0.13862202
## SI   0.4450973 0.066231535  0.263039825 -0.04428844 -0.22493820  0.01570329
```

Effective Sample Size

As we can see in the summary for each of the three models (displayed only for the hierarchical model), the effective sample size for each parameter is at least 3000, which is plenty!

Rhat convergence

All of the models also have r hat values of 1.0, so the HMC chains were both convergent and independent.

HMC-Specific diagnostics

We received no warnings about tree depth or divergences. Therefore, we assume that the HMC chains ran properly and never exceeded the default maximum tree depth (10). Nor did we get any divergent transitions, so there was no need to adjust the default adapt_delta parameter (0.8).

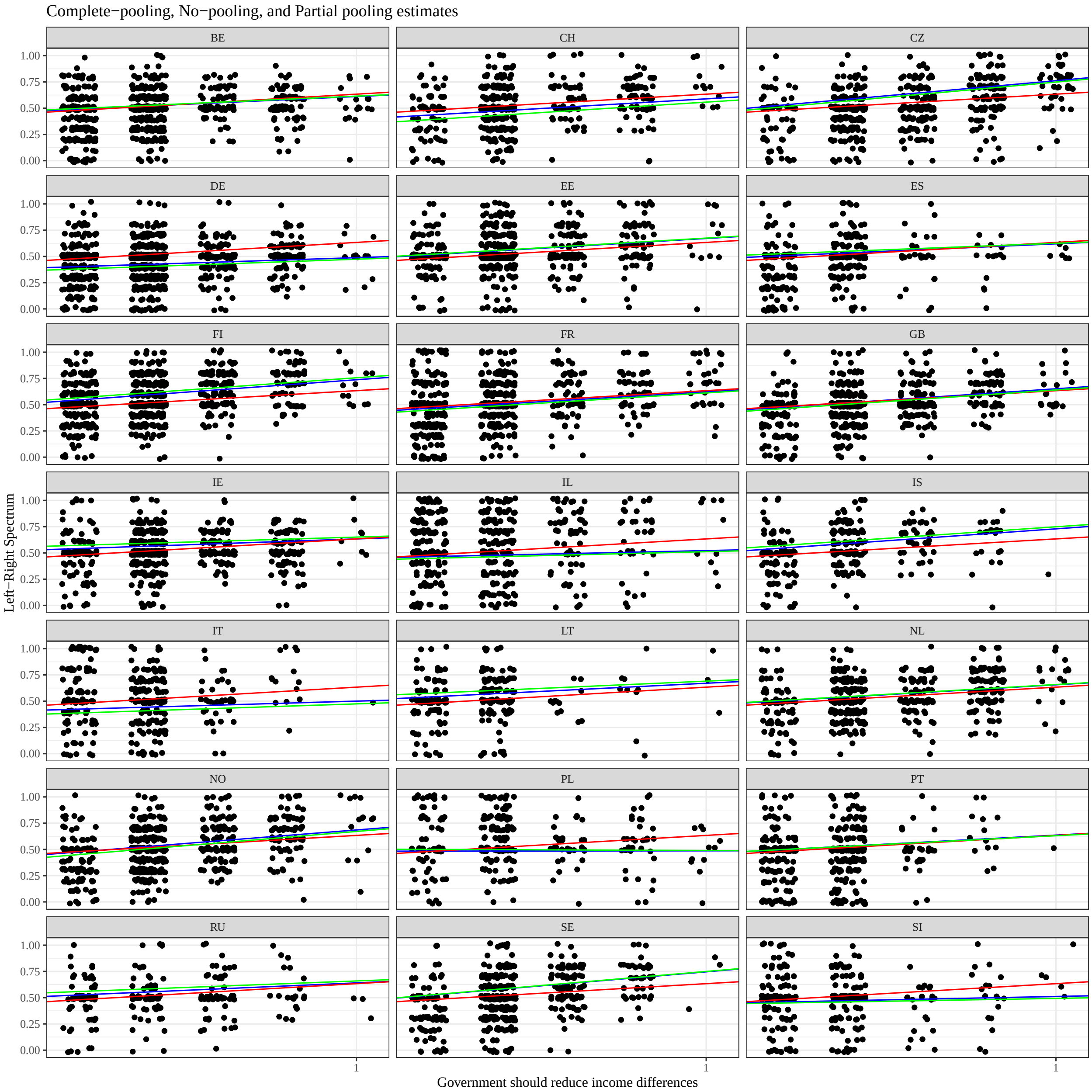
Interesting anomalies:

Sweden: Has steep slopes for most variables, possibly indicating greater political polarization along these variables. Poland and Slovenia: slope different sign than other countries for many variables, indicating that "left" and "right" carry different connotations in these countries. Italy: Very large slope for impcntr, possibly indicating that immigration is a very divisive topic in Italy right now. EE: Max slope for rlgblg, possibly indicating that religion is a very divisive topic in Estonia. LT: Min slope for impcntr, possibly indicating that immigration is not a very important political topic in Lithuania right now. CZ: Min slope for sbsrnen, possibly indicating that climate change is not a critical political issue in Czechia right now.

4. Plot of all models in all countries

The following plot shows each of the 3 models for each of the 21 countries in the survey. The line for the pooled model is the same for each country, but the others differ. Notice that the hierarchical model is similar to the seperate model, but skews slightly towards the pooled result.

```
# (0) Set axes & choose schools
y <- sample_data_best_5$lrscale
x <- sample_data_best_5$gincdif
countryid <- sample_data_best_5$country
sel.cntry <- unique(country)

# (1) Subset 8 of the schools; generate data frame
df <- data.frame(y, x, countryid)
df8 <- subset(df, countryid %in% sel.cntry)

# (2) Assign complete-pooling, no-pooling, partial pooling estimates
a_pooled <- coef(fit_lm_5)[1]
b_pooled <- coef(fit_lm_5)[2]

a_nopooled <- coef(nopooled)$country[,1]
b_nopooled <- coef(nopooled)$country[,2]

a_part_pooled <- coef(fit_hier_2)$country[, 1]
b_part_pooled <- coef(fit_hier_2)$country[, 2]

df8$a_pooled <- a_pooled
df8$b_pooled <- b_pooled

df8$a_nopooled <- a_nopooled[df8$countryid]
df8$b_nopooled <- b_nopooled[df8$countryid]

df8$a_part_pooled <- a_part_pooled[df8$countryid]
df8$b_part_pooled <- b_part_pooled[df8$countryid]

ggplot(data = df8,
       aes(x = x, y = y)) +
  facet_wrap(facets = ~ countryid,
             ncol = 3) +
  theme_bw() +
  geom_jitter(position = position_jitter(width = .05,
                                         height = 0.02)) +
  geom_abline(aes(intercept = a_part_pooled,
                  slope = b_part_pooled),
              linetype = "solid",
              color = "blue",
              size = 0.5) +

  geom_abline(aes(intercept = a_pooled,
                slope = b_pooled),
            linetype = "solid",
            color = "red",
            size = 0.5) +

  geom_abline(aes(intercept = a_nopooled,
              slope = b_nopooled),
          linetype = "solid",
          color = "green",
          size = 0.5) +

  scale_x_continuous(breaks = c(0, 1)) +
  labs(title = "Complete-pooling, No-pooling, and Partial pooling estimates",
       x = "Government should reduce income differences",
       y = "Left-Right Spectrum")+theme_bw( base_family = "serif")
```



5. Validation and Leave-One-Out Analysis

Below, we can see the loo output for the best 5 variable simple linear model. All k values are less than 0.5, so the loo output has good convergence. The p_loo value is about 8, which is quite high for a ten parameter model (11 parameters including the bias term).

```
loo_1 <- loo(fit_lm_1)
loo_5 <- loo(fit_lm_5)
loo_all <- loo(fit_lm_all)
loo_hier <- loo(fit_hier_2)
loo_5
```

```
##
## Computed from 2000 by 10000 log-likelihood matrix
```

```
##
##          Estimate    SE
## elpd_loo   1499.0  76.0
## p_loo         7.6   0.2
## looic     -2998.0 152.0
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

Loo_compare() also shows a clear preference for the five-variable model over the one-variable model. The elpd difference is about 175 between the two models, with a standard error of only 20, so the preference is highly significant.

```
loo_compare(loo_5, loo_1)
```

```
##          elpd_diff se_diff
## fit_lm_5    0.0       0.0
## fit_lm_1 -290.8      25.8
```

The following result is interesting. The hierarchical model is only slightly favored over the five-predictor model. However, the hierarchical model preserves country-specific information, so it is preferred for analysis.

```
loo_compare(loo_hier, loo_5)
```

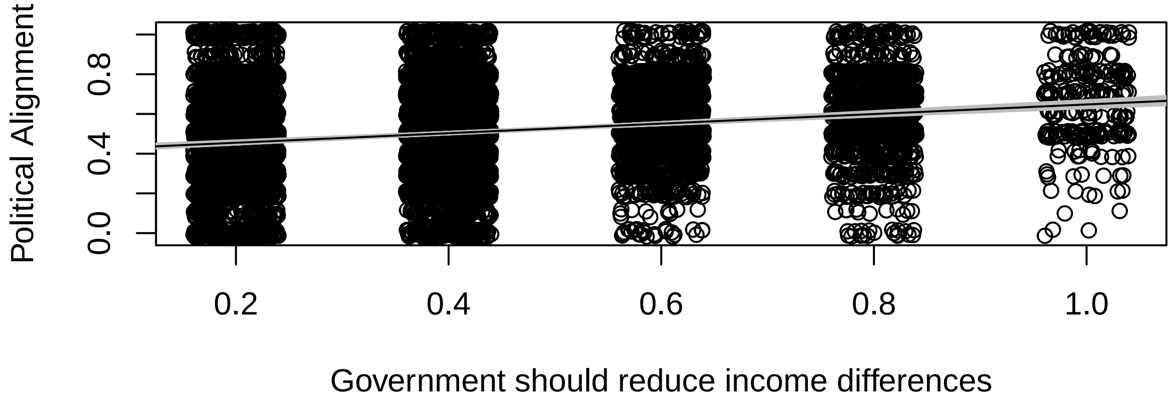
```
##            elpd_diff se_diff
## fit_hier_2    0.0       0.0
## fit_lm_5   -272.7      23.6
```

6. Posterior Predictive Checking

Now that we have selected the best variable to use for our models, we can fit a generalized linear model on the first predictor, “Government should reduce differences in income levels” and compare that with the five-variable and all-variable simple linear models.

Here is a simple, best-fit line for the pooled model on the first predictor. This is not a great method of checking, because it does not preserve any of the Bayesian uncertainty.

```
sims <- as.matrix(fit_lm_1)
n_sims <- nrow(sims)
subset <- sample(n_sims, 1000)
plot(jitter(sample_data_best_5$gincdif,1), jitter(sample_data_best_5$lrscale,1), xlab="Government shoul
for(i in subset){
  abline(sims[i,1], sims[i,2], col="gray")
}
abline(coef(fit_lm_1))
```



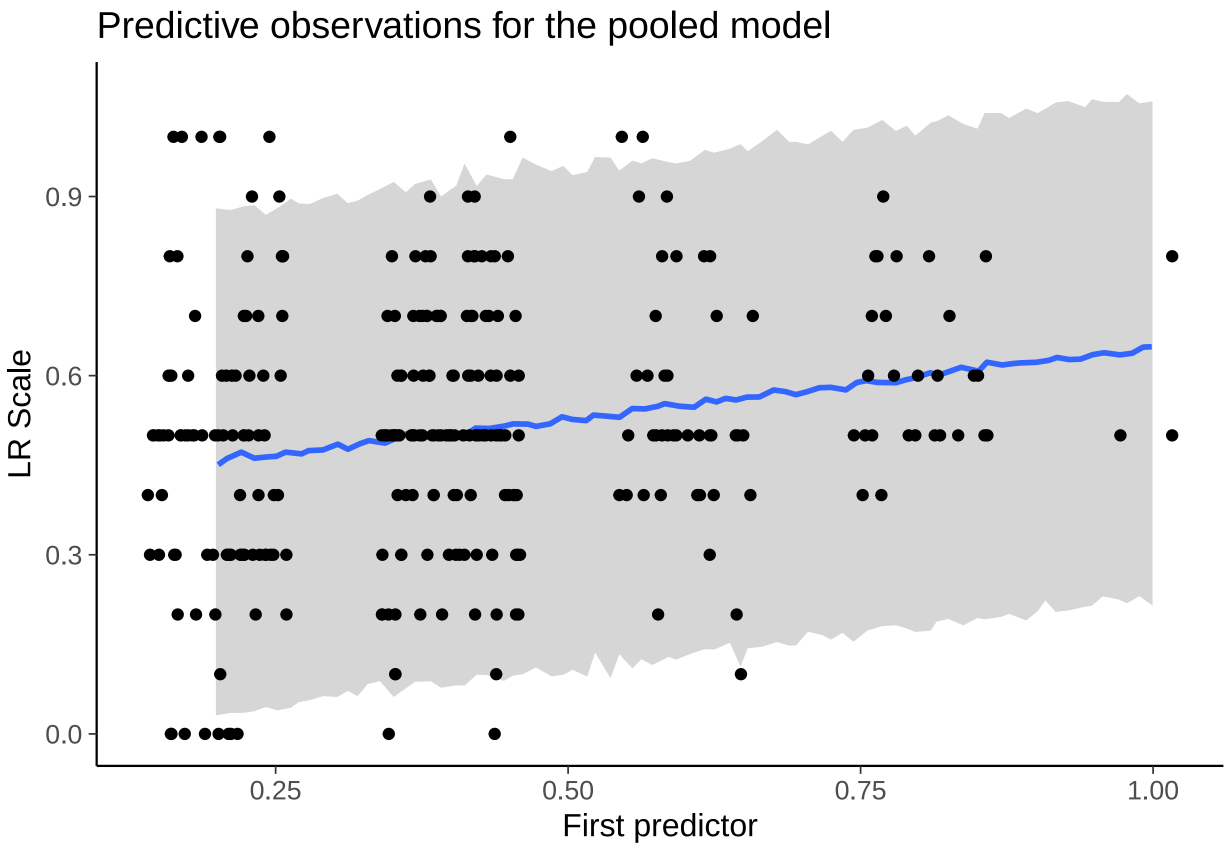
Instead we use the rstanarm posterior_predict function, which generates new data to be compared to the original data.

```
g_rng <- range(sample_data_best_5$gincdif)
g_steps <- seq(g_rng[1], g_rng[2], length.out = 80)
new_data <- data.frame(
  observation = seq_along(g_steps),
  gincdif = g_steps)
pred_post <- posterior_predict(fit_lm_1, newdata = new_data)
dim(pred_post)
```

```
## [1] 4000   80
```

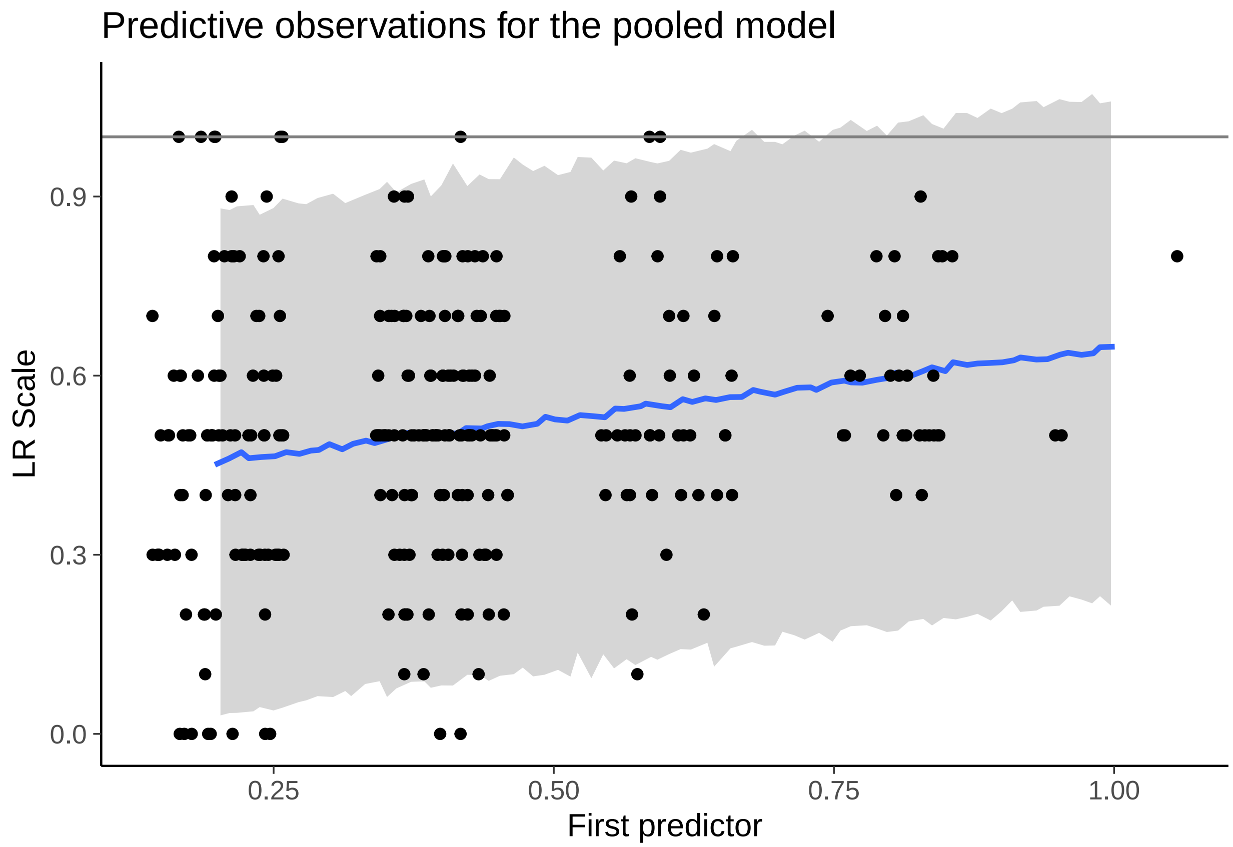
```
df_pred_post <- tidy_predictions(pred_post, new_data)
```

```
ggplot(sample_n(sample_data_best_5,300)) +
  aes(x = jitter(gincdif,1.5)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), data = df_pred_post,
              alpha = 0.4, fill = "grey60") +
  geom_line(aes(y = median), data = df_pred_post, colour = "#3366FF", size = 1) +
  geom_point(aes(y = lrscale)) +
  labs(title="Predictive observations for the pooled model",
        x ="First predictor", y = "LR Scale")
```



At the right end of the plot, some of the posterior predictions at the 95% level exceed 1, which is a small error with the model.

```
last_plot() +
  geom_hline(yintercept = 1, color = "grey50") +
  geom_label(x = 0, y = 1, label = "1")
```



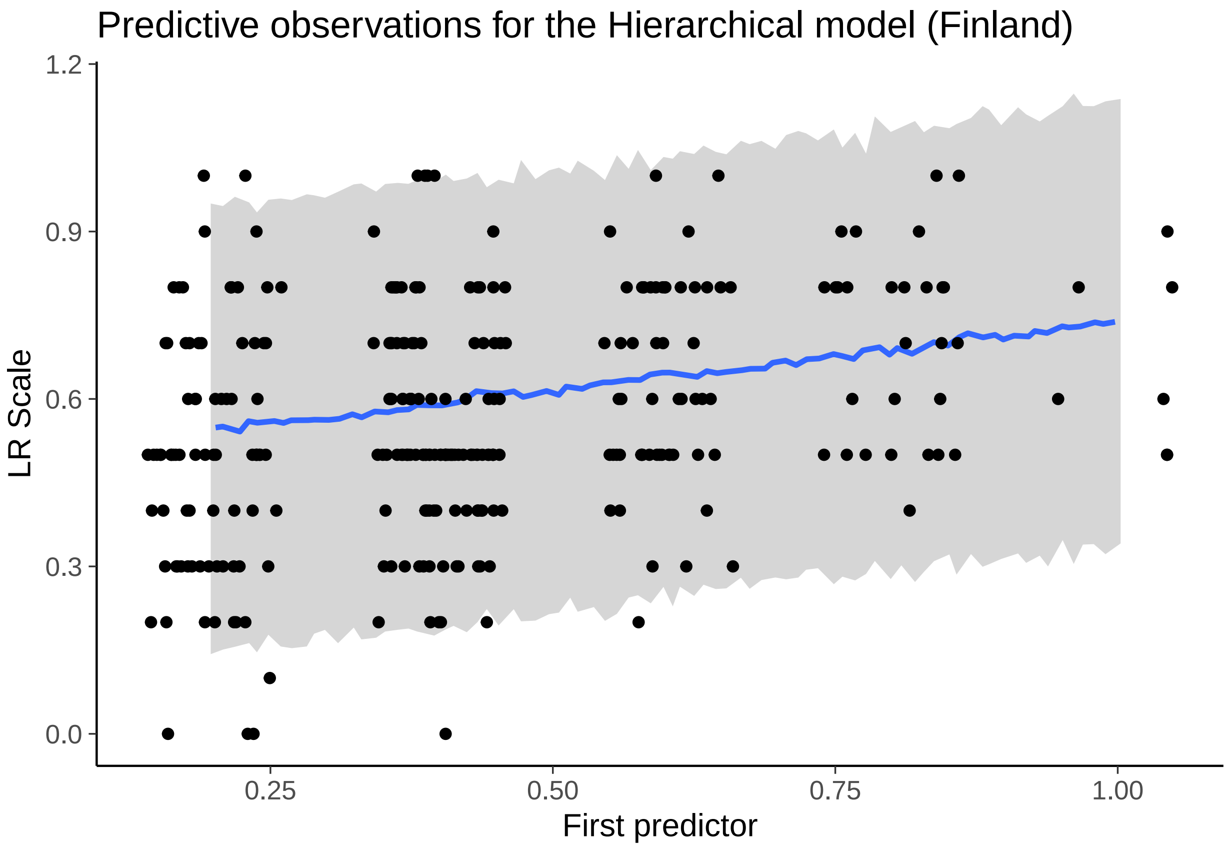
We repeat the posterior checking for the hierarchical model, using Finland as an example.

```
g_rng <- range(sample_data_best_5$gincdif)
g_steps <- seq(g_rng[1], g_rng[2], length.out = 80)

new_data_fi <- data.frame(
  observation = seq_along(g_steps),
  country = "FI",
  gincdif = g_steps,
  impcntr = g_steps,
  dfincac = g_steps,
  rlgblg = g_steps,
  sbsrnen = g_steps
  )
pred_post_fi <- posterior_predict(fit_hier_2, newdata = new_data_fi)
dim(pred_post_fi)
```

```
## [1] 2000   80
```

```
df_pred_post_fi <- tidy_predictions(pred_post_fi, new_data)
```

```
ggplot(sample_n(sample_data_best_5[which(sample_data_best_5$country=="FI"),],300)) +
  aes(x = jitter(gincdif,1.5)) +
  geom_ribbon(aes(ymin = lower, ymax = upper), data = df_pred_post_fi,
              alpha = 0.4, fill = "grey60") +
  geom_line(aes(y = median), data = df_pred_post_fi, colour = "#3366FF", size = 1) +
  geom_point(aes(y = lrscale)) +
  labs(title="Predictive observations for the Hierarchical model (Finland)",
        x ="First predictor", y = "LR Scale")
```



7. Problems and Potential Improvements

We would like to further understand the differences between countries. Also, we would love to be able to explore other connections in the data set, not only predicting the political spectrum alignment. Self-identification as "left" or "right" is a limited concept without much depth.

As far as the models go, one potential improvement is to use a different (non-Gaussian) link function. We do not know the distribution of our residuals, but it may not be Gaussian. In particular, the error distribution might have fatter tails, because many people have more extreme political beliefs.

Another potential improvement to explore would be to choose different variables for each of the 21 countries. The five best predictive variables might be different in Poland vs. Finland, for example, since Poland is more politically divided by social rather than economic issues.

8. Conclusion

The main conclusion is that, while Europe has common trends in what it means to identify as "left" and "right" on the political spectrum, but major differences still exist between different countries. These terms don't mean the same thing in each country, as you can see in the partially-pooled plots comparing, for example, Sweden and Poland.

Despite the differences, some trends are as-expected. For example, friendliness to immigrants certainly correlates, in most countries, with being more "left" on the political spectrum. In the countries that see different correlations, perhaps it is because–in the european context–they see themselves as possible migrants to other, wealthier European countries.

We are especially interested in those countries where the first variable "Government should reduce differences in income levels" does not correlate with the traditional left-right spectrum. We discussed with a Polish friend and learned that "left" in Poland is seen differently from the way it is seen in Western Europe. Left and right in Poland are more correlated with social issues such as religion and LGBT rights. We have no basis to make further claims, but it is an interesting avenue for research.

9. Code Resources

Correlation plots: https://avehtari.github.io/modelselection/diabetes.html// Posterior predictive plotting: https://www.tjmahr.com/visualizing-uncertainty-rstanarm/// Variable selection: https://mc-stan.org/projjpred/articles/quickstart.html// Partial pooling: http://mc-stan.org/rstanarm/articles/pooling.html// and https://mc-stan.org/users/documentation/case-studies/tutorial_rstanarm.html