Replication: The Structure of Inequality and the Politics of Redistribution

Filippo Teoldi, Zara Riaz and Julian Gerez

October 23rd, 2018

This .Rmd file replicates the analyses in **Lupu, N. and Pontusson, J., (2011), The Structure of Inequality and the Politics of Redistribution**. It requires that the following packages be installed: (1) `haven`, (2) `zoo`, (3) `dplyr`, and (4) `panelAR`. If you do not have these packages installed, use `install.packages()` to install them.

Data cleaning

Loading the dataset and generating new variables

First we open the dataset and assign it to object `data` with the `haven` package, which allows us to open .dta files. The dataset can be downloaded at http://www.noamlupu.com/LupPon_APSR.dta.

```
library('haven')
data <- read_dta("LupPon_APSR.dta")
```

With the data open, we can begin by changing the values of certain variables so that they match the values in the paper. To begin, the authors redefine their disproportionality measures, `disp_gall` as inverse disproportionality measures, which can be represented by the "old" variable multiplied by -1.

```
data$disp_gall <- data$disp_gall*-1
```

The variable for female participation, `fempar`, and for annual net union density, `union` are multiplied by 100 so that they are rescaled.

```
data$fempar <- data$fempar*100
data$union <- data$union*100
```

The variables `pjoint` and `disp_gall`, are partisanship and disproportionality, respectively. These are standardized from [0,1]. To do so, we are defining a function, `range01`, which standardizes the range of a variable such that it takes on values from 0 to 1, and applying `range01` to the two variables.

```
range01 <- function(x, ...) {(x - min(x, ...)) / (max(x, ...) - min(x, ...))}

data$stdpjoint <- range01(data$pjoint, na.rm = TRUE)
data$stdpdisp_gall <- range01(data$disp_gall, na.rm = TRUE)
```

Interpolation of missing values

Next, we interpolate missing values in the data. The variables that we are defining as the "new" interpolated variables are: `pratio9050` (the 90-50 ratio, extrapolated), `pratio5010` (the 50-10 ratio, extrapolated), `pratio9050s` (the 90-50 ratio, not extrapolated), `pratio5010s` (the 50-10 ratio, not extrapolated), `pforeign` (the percentage of foreign-born residents in the population) and `pvoc` (a measure of vocational training).

Missing values are interpolated *for each country*, rather than for the dataset as a whole, so we write a loop to define the object `data_countries` as a list of the data (with these aforementioned new variables) subsetted by each country.

The `zoo` package allows use to use the function `na.approx` to linearly interpolate missing values for each variable. We use a set of loops that interpolates missing values indexed for each country, `i`, in our list of data.frames, `data_countries`, *for each variable*.[1] Finally, we can use `rbind` to bind this new list into a single `data.frame`, and remove our list of data.frames, `data_countries`.

```
data_countries <- lapply(unique(data$country), function(x)
  subset(data, data$country==x)
)

library('zoo')

for (i in 1:length(data_countries)){
data_countries[[i]] <- cbind(data_countries[[i]], sapply(c(5:8, 16, 19), function(y)
   na.approx(data_countries[[i]][,y],
             x = index(data_countries[[i]][,3],
                       data_countries[[i]][,y]), na.rm = FALSE)
  ))
}

data <- do.call("rbind", data_countries)

names(data)[24:29] <- c("pratio9050", "pratio5010", "pratio9050s",
                        "pratio5010s", "pforeign", "pvoc")

rm(data_countries)
```

Generating new variables with interpolated variables

We generate an immigration measure, `fpop` which reflects the percentage of the population that is foreign-born by using our interpolated measure `pforeign`, multiplying it by 1000, and dividing this result by `pop`, which is total population.

```
data$pforeign <- data$pforeign*1000
data$fpop <- (data$pforeign/data$pop)*100
```

Our last variable-generating step before moving on to calculating the averages for the redistribution models is to create additional measures of inequality as defined by manipulations to our existing measures of inequality: `ratio9010`, `ratio9010s`, `skew`, and `skews`.

[1] This is what `data_countries[[i]][,y>23]` refers to, where `i` is each country and `y` represents the new variables. The 24th column is `pratio9050`, the 25th column `pratio5010`, and so on. Each of these are interpolated using the original variables, which are represented in `data_countries[[i]][,z>5]`, where `z` represents the original variables corresponding the new variables (i.e. `pratio9050` is interpolated using `ratio9050`, which is in the 5th column, and so on). Note that the index along which the function is operating is by year (`data_contries[[i]][,3]`) for *every* variable. In other words, we are replacing the variables of interest in each country for missing years.

```
data$ratio9010 <- data$pratio9050*data$pratio5010
data$ratio9010s <- data$pratio9050s*data$pratio5010s # not extrapolated
data$skew <- data$pratio9050/data$pratio5010
data$skews <- data$pratio9050s/data$pratio5010s # not extrapolated
```

Calculate moving averages for redistribution models

Because data on redistribution are unequally spaced for the period of the study, the authors use a time-series cross-sectional model where the indepdent variables are averaged across the period since the last redistribution observation.

First, we generate the `since` variable, which as mentioned, represents the years since the last redistribution, `redist`, for each country `i`. We remake our list of the subset of countries as before and define `since` (the 35th column in each data.frame `i` of `data_countries`, or `data_countries[[i]][35]`) accordingly by creating a new logical vector, `nona`, that tells us when the `redist` variable is and is not defined for each country. We rename the new column and then remove `data_countries` and `nona`.

```
data_countries <- lapply(unique(data$country), function(x)
  subset(data, data$country==x)
)

for (i in 1:length(data_countries)){
  data_countries[[i]] <- cbind(data_countries[[i]], NA)
  nona <- !is.na(data_countries[[i]][,4])
  data_countries[[i]][,35][nona] <- c(NA, diff(data_countries[[i]][,3][nona]))
}

data <- do.call("rbind", data_countries)
names(data)[35] <- "since"
rm(data_countries, nona)
```

Now we can calculate the moving averages. Again, we create our country list of data.frames `data_countries`. Then, for each variable `a` (within the `sapply loop`), we use the function `rollapply` to calculate a moving average of the variable for `j` in 1:10 iterations. We do this for each country `i` and use `rbind` to bind this new list into a single `data.frame`, `data` before removing `data_countries`.

```
library('dplyr')

data_countries <- lapply(unique(data$country), function(x)
  subset(data, data$country==x)
)

for (i in 1:length(data_countries)){
  for (j in 1:10) {
    data_countries[[i]] <- cbind(data_countries[[i]],
        sapply(c(31, 24, 25, 22, 33, 23, 29, 12, 30, 10, 11, 13), function(a)
          lag(rollapply(data_countries[[i]][,a], j,
                        FUN = mean, fill = NA, align = "right"),1)
    ))
  }
}
data <- do.call("rbind", data_countries)
rm(data_countries)
```

Next, we must reorder our columns so that each variable `a` is matched with its corresponding 10 moving averages.

```
temp <- data[,c(1:35, 35+1, 35+1+12, 35+1+24, 35+1+36, 35+1+48, 35+1+60, 35+1+72,
                35+1+84, 35+1+96, 35+1+108)]

for (i in 2:12) {
  temp <- cbind(temp, data[,c(35+i, 35+i+12, 35+i+24, 35+i+36, 35+i+48, 35+i+60,
                              35+i+72, 35+i+84, 35+i+96, 35+i+108)])
}

data <- temp
rm(temp)
```

Because we are applying lagged rolling means, each iteration `j` of the `rollapply` function is "shifted down" one row for each `j`. The following section of the code "fills" rows across for and only for each variable `a`.

```
for(x in 37:45){
    for(y in 2:nrow(data)){
        if(is.na(data[y,x])){
            data[y,x] <- data[y,x-1]
        }
    }
}

for(a in c(0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110)){
  for(x in c(37:45+a)){
      for(y in 2:nrow(data)){
          if(is.na(data[y,x])){
              data[y,x] <- data[y,x-1]
          }
      }
  }
}
```

Finally, we can create our dependent variables: `dvratio9010`, `dvpratio9050`, `dvpratio5010`, `dvstdpjoint`, `dvskew`, `dvstddisp_gall`, `dvpvoc`, `dvunion`, `dvfpop`, `dvfempar`, `dvunempl`, and `dvturnout`. We define the twelve empty slots in columns 156:167 of the data and conditionally fill these columns for each variable `a` depending on the variable `since` (which you'll recall is in the 35th column of `data`, or `data[,35]`). In doing so, we match these moving averages to redistribution observations to values that correspond to the correct moving average based on the period of redistribution.[2]

We then delete these extraneous columns and rename our variables.

```
data[,156:167] <- NA

for (a in 0:11) {
  data[,156+a] <- case_when(
    data[,35] %in% 1 ~ data[,36+a*10],
    data[,35] %in% 2 ~ data[,37+a*10],
```

[2]Note that there are three possible scenarios here: (1) a redistribution observation is observed 1 year after the previous: the independent variable takes on its 1-year lagged value; (2) a redistribution observation is observed n years ago, where n is 2-10: the independent variable takes on its nth year moving average value; (3) a redstribution observation is the first observation for the country: the independent variable takes on its 10th year moving average value.

```
    data[,35] %in% 3 ~ data[,38+a*10],
    data[,35] %in% 4 ~ data[,39+a*10],
    data[,35] %in% 5 ~ data[,40+a*10],
    data[,35] %in% 6 ~ data[,41+a*10],
    data[,35] %in% 7 ~ data[,42+a*10],
    data[,35] %in% 8 ~ data[,43+a*10],
    data[,35] %in% 9 ~ data[,44+a*10],
    data[,35] %in% 10 ~ data[,45+a*10]
  )

data[,156+a][is.na(data[,35]) & !is.na(data[,4])] <-
  data[,45][is.na(data[,35]) & !is.na(data[,4])]
}

data <- data[,c(1:35,156:167)]

names(data)[36:47] <- c("dvratio9010", "dvpratio9050", "dvpratio5010", "dvstdpjoint",
                        "dvskew", "dvstddisp_gall", "dvpvoc", "dvunion", "dvfpop",
                        "dvfempar", "dvunempl", "dvturnout")
```

Create social spending variables

To estimate the model using the 2nd dependent variable (`socspend`), we create five-year moving averages for this variable and all independent variables `b`, that is `ma_socspend`, `ma_pratio9050s`, `ma_pratio5010s`, `ma_pratio9010s`, `ma_skews`, `ma_dreher`, `ma_pop65`, `ma_stdpjoint`, `ma_stddisp_gall`, `ma_fempar`, `ma_unempl`, `ma_union`, `ma_turnout`, `ma_pvoc`, `ma_fpop`. We rename the variables accordingly.

```
data <- cbind(data, sapply(c(20, 26:27, 32, 34, 21, 15, 22:23, 10:13, 29:30), function(b)
  (lag(data[,b], 1)+lag(data[,b], 2)+lag(data[,b], 3)+lag(data[,b], 4)+lag(data[,b], 5))/5
))

names(data)[48:62] <- c("ma_socspend", "ma_pratio9050s", "ma_pratio5010s",
                        "ma_pratio9010s", "ma_skews", "ma_dreher", "ma_pop65",
                        "ma_stdpjoint", "ma_stddisp_gall", "ma_fempar", "ma_unempl",
                        "ma_union", "ma_turnout", "ma_pvoc", "ma_fpop")
```

We are now ready to reproduce the tables in the paper.

Replication

We begin by removing the variables we don't need and subsetting and sorting the data into a new data.frame: `redistsample`.

```
data_redist <- data[,c(1:4, 37:38, 36, 40, 47, 45, 41:42, 43, 46 )]

redistsample <- data[!is.na(data$redist),]

redistsample <- redistsample[with(redistsample, order(id, year)),]
```

Then we lag the main outcome variable, `redist` (redistribution), generating `redist_lag`, and set the time series variable, `time`.

```
redistsample$redist_lag <- unlist(by(redistsample,redistsample$id,function(x){
  c(NA,x[,"redist"][1:(length(x[,"redist"])-1)])
  }))

redistsample$time <- unlist(by(redistsample,redistsample$id,function(x) seq(1:nrow(x))))
```

Table 2

The first 8 specifications correspond to the second table[3] in the paper, and concern the determinants of redistribution.

Specification 1:

```
library('panelAR')
out1 <- panelAR(redist ~ redist_lag + dvpratio9050 + dvpratio5010 + dvturnout +
                  dvfempar + dvstddisp_gall + dvpvoc + dvunion + dvunempl,
                data=redistsample, panelVar='id', timeVar='time', autoCorr='ar1',
                panelCorrMethod='pcse',rho.na.rm=TRUE, panel.weight='t-1',
                bound.rho=TRUE)
summary(out1)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       68 Avg obs. per panel 4.5333
##  Number of panels: 15 Max obs. per panel 9
##  Number of times:  9  Min obs. per panel 1
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.26666   11.15944  -0.293 0.770776
## redist_lag      0.50658    0.12652   4.004 0.000179 ***
## dvpratio9050    3.81044    3.35976   1.134 0.261402
## dvpratio5010   -4.76833    2.06327  -2.311 0.024405 *
## dvturnout       0.09781    0.03644   2.684 0.009454 **
## dvfempar        0.09134    0.05464   1.672 0.099973 .
## dvstddisp_gall  0.07253    2.54464   0.029 0.977360
## dvpvoc          0.01860    0.03668   0.507 0.613909
## dvunion         0.08862    0.03736   2.372 0.021029 *
## dvunempl        0.12415    0.13443   0.923 0.359580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.8886
## Wald statistic: -708.2307, Pr(>Chisq(9)): 1
```

[3]Table 1 in the paper is a descriptive table and is not pertinent to our analysis.

Specification 2 (remove outliers):

```r
# defining outliers
mod1.resid <- out1$residuals
index <- which(abs((mod1.resid-mean(mod1.resid))/sd(mod1.resid)) <= 1.5)
# creating a new subset without these observations
redistsample_noout<- out1$model[index,]

# running same model as spec1 with new subset
out2 <- panelAR(redist ~ redist_lag + dvpratio9050 + dvpratio5010 + dvturnout +
                  dvfempar + dvstddisp_gall + dvpvoc + dvunion + dvunempl,
                data=redistsample_noout, panelVar='id', timeVar='time',
                autoCorr='ar1', panelCorrMethod='pcse',rho.na.rm=TRUE,
                panel.weight='t-1', bound.rho=TRUE)
summary(out2)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       58 Avg obs. per panel 3.8667
##  Number of panels: 15 Max obs. per panel 8
##  Number of times:  9  Min obs. per panel 1
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     0.57080    7.27261   0.078   0.9378
## redist_lag      0.49404    0.07800   6.333 7.74e-08 ***
## dvpratio9050    6.04188    2.81801   2.144   0.0371 *
## dvpratio5010   -6.58628    1.32426  -4.974 8.82e-06 ***
## dvturnout       0.06427    0.02554   2.516   0.0153 *
## dvfempar        0.07852    0.03606   2.178   0.0344 *
## dvstddisp_gall -2.46670    2.05462  -1.201   0.2358
## dvpvoc          0.01582    0.02327   0.680   0.4999
## dvunion         0.12558    0.01634   7.686 6.59e-10 ***
## dvunempl        0.04132    0.10911   0.379   0.7066
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.931
## Wald statistic: 2323.5872, Pr(>Chisq(9)): 0
```

Specification 3 (no controls):

```r
out3 <- panelAR(redist ~ dvpratio9050 + dvpratio5010 + as.factor(id),
                data=redistsample, panelVar='id', timeVar='time', autoCorr='ar1',
                panelCorrMethod='pcse',rho.na.rm=TRUE, panel.weight='t-1',
                bound.rho=TRUE)
summary(out3)
```

```
##
```

```
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       77 Avg obs. per panel 5.1333
##  Number of panels: 15 Max obs. per panel 10
##  Number of times:  10 Min obs. per panel 1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       28.6464     1.5050  19.035  < 2e-16 ***
## dvpratio9050       3.1021     3.6452   0.851 0.398135
## dvpratio5010      -5.2122     3.3449  -1.558 0.124429
## as.factor(id)3    13.0196     1.7397   7.484 3.73e-10 ***
## as.factor(id)4    -1.7132     1.6163  -1.060 0.293442
## as.factor(id)5    11.3941     1.9599   5.814 2.51e-07 ***
## as.factor(id)6    10.5042     1.7852   5.884 1.92e-07 ***
## as.factor(id)7     1.0597     1.0530   1.006 0.318290
## as.factor(id)8    -2.2701     1.0771  -2.108 0.039243 *
## as.factor(id)9     0.8848     0.8576   1.032 0.306374
## as.factor(id)12    5.2643     1.6292   3.231 0.002002 **
## as.factor(id)14    5.4071     1.5194   3.559 0.000736 ***
## as.factor(id)15   12.4256     1.5801   7.864 8.37e-11 ***
## as.factor(id)16  -13.8440     0.8686 -15.939  < 2e-16 ***
## as.factor(id)17   -2.5672     0.9446  -2.718 0.008577 **
## as.factor(id)18   -7.1873     0.8568  -8.389 1.07e-11 ***
## as.factor(id)20  -11.2544     0.6635 -16.962  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.8907
## Wald statistic: -7396611.786, Pr(>Chisq(16)): 1
```

Specification 4 (no controls, no outliers):

```r
# defining outliers
mod3.resid <- out3$residuals
index <- which(abs((mod3.resid-mean(mod3.resid))/sd(mod3.resid)) <= 1.5)
# creating a new subset without these observations
redistsample_noout<- out3$model[index,]
# running same model as spec3 with new subset
out4 <- panelAR(redist ~ dvpratio9050 + dvpratio5010 + as.factor(id),
                data=redistsample_noout, panelVar='id', timeVar='time',
                autoCorr='ar1', panelCorrMethod='pcse',rho.na.rm=TRUE,
                panel.weight='t-1', bound.rho=TRUE)
summary(out4)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       67 Avg obs. per panel 4.4667
##  Number of panels: 15 Max obs. per panel 8
```

```
##  Number of times:  10 Min obs. per panel 1
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        28.6063     1.1576  24.712  < 2e-16 ***
## dvpratio9050        0.5803     2.7141   0.214 0.831569
## dvpratio5010       -2.5564     2.5130  -1.017 0.313938
## as.factor(id)12     5.4961     1.5691   3.503 0.000980 ***
## as.factor(id)14     5.2873     1.3689   3.863 0.000323 ***
## as.factor(id)15    11.5201     0.8651  13.317  < 2e-16 ***
## as.factor(id)16   -14.0451     0.8292 -16.938  < 2e-16 ***
## as.factor(id)17    -2.4030     0.7185  -3.344 0.001570 **
## as.factor(id)18    -7.2720     0.8358  -8.701  1.4e-11 ***
## as.factor(id)20   -11.2567     0.6801 -16.552  < 2e-16 ***
## as.factor(id)3     12.9580     1.5070   8.598  2.0e-11 ***
## as.factor(id)4     -3.8450     1.2026  -3.197 0.002407 **
## as.factor(id)5     12.6117     0.5709  22.090  < 2e-16 ***
## as.factor(id)6      9.4638     0.6985  13.549  < 2e-16 ***
## as.factor(id)7      1.4003     0.8467   1.654 0.104432
## as.factor(id)8     -2.2843     1.1069  -2.064 0.044247 *
## as.factor(id)9      0.4146     0.6867   0.604 0.548729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9572
## Wald statistic: -5939207.5779, Pr(>Chisq(16)): 1
```

Specification 5 (using skew as main inequality measure):

```
out5<- panelAR(redist ~ redist_lag + dvratio9010 + dvskew + dvturnout + dvfempar +
                 dvstddisp_gall + dvpvoc + dvunion + dvunempl, data=redistsample,
               panelVar='id', timeVar='time', autoCorr='ar1',
               panelCorrMethod='pcse',rho.na.rm=TRUE, panel.weight='t-1',
               bound.rho=TRUE)
summary(out5)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       68 Avg obs. per panel 4.5333
##  Number of panels: 15 Max obs. per panel 9
##  Number of times:  9  Min obs. per panel 1
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -14.73371    9.19697  -1.602 0.114585
## redist_lag         0.49211    0.12412   3.965 0.000204 ***
## dvratio9010       -0.01548    1.13592  -0.014 0.989172
## dvskew            10.17135    3.67271   2.769 0.007529 **
## dvturnout          0.10182    0.03629   2.806 0.006819 **
## dvfempar           0.08536    0.05333   1.601 0.114901
```

```
## dvstddisp_gall  -0.06816    2.45060  -0.028 0.977905
## dvpvoc           0.01991    0.03702   0.538 0.592875
## dvunion          0.09013    0.03607   2.499 0.015316 *
## dvunempl         0.11177    0.13563   0.824 0.413280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.8918
## Wald statistic: -151.0234, Pr(>Chisq(9)): 1
```

Specification 6 (skew as main measure, no outliers):

```
mod5.resid <- out5$residuals
index <- which(abs((mod5.resid-mean(mod5.resid))/sd(mod5.resid)) <= 1.5)
#creating a new subset without these observations
redistsample_noout<- out5$model[index,]
#running same model as spec5 with new subset
out6<- panelAR(redist ~ redist_lag + dvratio9010 + dvskew + dvturnout + dvfempar +
                 dvstddisp_gall + dvpvoc + dvunion + dvunempl,
               data=redistsample_noout, panelVar='id', timeVar='time',
               autoCorr='ar1', panelCorrMethod='pcse',rho.na.rm=TRUE,
               panel.weight='t-1', bound.rho=TRUE)
summary(out6)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       58 Avg obs. per panel 3.8667
##  Number of panels: 15 Max obs. per panel 8
##  Number of times:  9  Min obs. per panel 1
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -12.43089    6.18074  -2.011   0.0499 *
## redist_lag       0.48096    0.07362   6.533 3.83e-08 ***
## dvratio9010     -0.16200    0.94572  -0.171   0.8647
## dvskew          12.98571    2.58573   5.022 7.48e-06 ***
## dvturnout        0.06363    0.02581   2.466   0.0173 *
## dvfempar         0.07440    0.03485   2.135   0.0379 *
## dvstddisp_gall  -2.37649    1.93445  -1.229   0.2252
## dvpvoc           0.01183    0.02326   0.509   0.6134
## dvunion          0.12312    0.01525   8.073 1.71e-10 ***
## dvunempl         0.05119    0.10653   0.480   0.6331
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9346
## Wald statistic: 2147.9427, Pr(>Chisq(9)): 0
```

Specification 7 (skew as main measure, no controls, country fixed effects):

```
out7 <- panelAR(redist ~ dvratio9010 + dvskew + as.factor(id),
                data=redistsample, panelVar='id', timeVar='time',
                autoCorr='ar1', panelCorrMethod='pcse',rho.na.rm=TRUE,
                panel.weight='t-1', bound.rho=TRUE)
summary(out7)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       77 Avg obs. per panel 5.1333
##  Number of panels: 15 Max obs. per panel 10
##  Number of times:  10 Min obs. per panel 1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       26.1460     4.1339   6.325 3.50e-08 ***
## dvratio9010        0.2668     1.3637   0.196 0.845568
## dvskew            -1.4132     0.3999  -3.534 0.000795 ***
## as.factor(id)3    13.5290     1.9265   7.023 2.29e-09 ***
## as.factor(id)4    -4.5807     1.6324  -2.806 0.006752 **
## as.factor(id)5    12.2500     2.3403   5.234 2.23e-06 ***
## as.factor(id)6    11.5595     2.0283   5.699 3.88e-07 ***
## as.factor(id)7     1.4686     1.1581   1.268 0.209678
## as.factor(id)8    -2.3365     1.2967  -1.802 0.076586 .
## as.factor(id)9    -0.8564     1.7633  -0.486 0.628956
## as.factor(id)12    5.5030     1.6957   3.245 0.001920 **
## as.factor(id)14    6.0782     1.6855   3.606 0.000634 ***
## as.factor(id)15   13.8717     1.7522   7.917 6.80e-11 ***
## as.factor(id)16  -14.3658     0.6894 -20.839  < 2e-16 ***
## as.factor(id)17   -3.4290     1.2397  -2.766 0.007530 **
## as.factor(id)18   -8.3412     2.1239  -3.927 0.000224 ***
## as.factor(id)20  -12.3655     2.0052  -6.167 6.46e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.8823
## Wald statistic: -292002.1794, Pr(>Chisq(16)): 1
```

Specification 8 (skew as main measure, no controls, fixed effects without outliers):

```
mod7.resid <- out7$residuals
index <- which(abs((mod7.resid-mean(mod7.resid))/sd(mod7.resid)) <= 1.5)
#creating a new subset without these observations
redistsample_noout<- out7$model[index,]
#running same model as spec7 with new subset
out8 <- panelAR(redist ~ dvratio9010 + dvskew + as.factor(id),
                data=redistsample_noout, panelVar='id', timeVar='time',
                autoCorr='ar1', panelCorrMethod='pcse',rho.na.rm=TRUE,
```

```
                 panel.weight='t-1', bound.rho=TRUE)
summary(out8)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       69 Avg obs. per panel 4.6
##  Number of panels: 15 Max obs. per panel 10
##  Number of times:  10 Min obs. per panel 1
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        23.8616     2.9902   7.980 1.37e-10 ***
## dvratio9010         0.9970     1.0352   0.963 0.339941
## dvskew             -1.2783     0.2256  -5.666 6.45e-07 ***
## as.factor(id)12     6.9216     1.7428   3.971 0.000221 ***
## as.factor(id)14     6.7963     1.7195   3.953 0.000234 ***
## as.factor(id)15    13.1220     1.1977  10.956 4.05e-15 ***
## as.factor(id)16   -14.3899     0.6619 -21.742  < 2e-16 ***
## as.factor(id)17    -3.4598     1.0232  -3.381 0.001376 **
## as.factor(id)18    -9.3134     1.8616  -5.003 6.82e-06 ***
## as.factor(id)20   -13.3046     1.7024  -7.815 2.50e-10 ***
## as.factor(id)3     13.9067     1.4967   9.292 1.23e-12 ***
## as.factor(id)4     -5.3429     1.4257  -3.748 0.000449 ***
## as.factor(id)5     13.6265     0.7529  18.099  < 2e-16 ***
## as.factor(id)6     10.4516     0.7107  14.705  < 2e-16 ***
## as.factor(id)7      1.1046     1.0914   1.012 0.316176
## as.factor(id)8     -3.4456     0.7727  -4.459 4.42e-05 ***
## as.factor(id)9     -1.6377     1.6292  -1.005 0.319445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9491
## Wald statistic: -5896554.336, Pr(>Chisq(16)): 1
```

Table 3

For the next table, we use the same 8 specifications but replace our dependent variable with social spending (socspend) and the 5-year moving averages of the independent variable names.

We use the full data set for these specifications, except when we drop the outliers. The independent variables included are: `socspend_lag mapratio9050s mapratio5010s mapop65 mafempar maturnout mastddisp_gall mapvoc maunion maunempl madreher gdpgrowth`; the "ma" prefixes denotes moving averages.

Again we begin by removing variables we don't need into our new data.frame `data_socspend`, sorting the data, and creating the lag and time variables.

```
data_socspend <- data[,c(1:3, 18, 20, 48:62 )]

data_socspend  <- data_socspend [with(data_socspend , order(id, year)),]
```

```r
data_socspend$socspend_lag <- unlist(by(data_socspend ,data_socspend$id,

data_socspend$time <- unlist(by(
  data_socspend ,data_socspend $id,function(x) seq(1:nrow(x))))
```

Specification 9:

```r
out9 <- panelAR(socspend ~ socspend_lag + ma_pratio9050s + ma_pratio5010s + ma_pop65 +
                  ma_turnout + ma_fempar + ma_stddisp_gall + ma_pvoc + ma_union +
                  ma_unempl + ma_dreher + gdpgrowth,
                data=data_socspend , panelVar='id', timeVar='time',
                autoCorr='psar1', panelCorrMethod='pcse',
                rho.na.rm=TRUE, bound.rho=TRUE)
summary(out9)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       311 Avg obs. per panel 17.2778
##  Number of panels: 18  Max obs. per panel 24
##  Number of times:  26  Min obs. per panel 2
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.145915   1.230513  -0.119 0.905687
## socspend_lag     0.906723   0.019469  46.572  < 2e-16 ***
## ma_pratio9050s   1.495763   0.405373   3.690 0.000267 ***
## ma_pratio5010s  -0.328159   0.369963  -0.887 0.375791
## ma_pop65        -0.095127   0.031547  -3.015 0.002787 **
## ma_turnout       0.006940   0.004102   1.692 0.091698 .
## ma_fempar        0.005885   0.008428   0.698 0.485554
## ma_stddisp_gall -0.885339   0.229047  -3.865 0.000136 ***
## ma_pvoc          0.021944   0.006986   3.141 0.001852 **
## ma_union         0.010789   0.002942   3.667 0.000290 ***
## ma_unempl       -0.067141   0.020026  -3.353 0.000903 ***
## ma_dreher        0.009512   0.006626   1.436 0.152158
## gdpgrowth       -0.183994   0.017779 -10.349  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9902
## Wald statistic: 26039.2026, Pr(>Chisq(12)): 0
```

Specification 10 (remove outliers):

```r
#defining outliers
mod9.resid <- out9$residuals
index <- which(abs((mod9.resid-mean(mod9.resid))/sd(mod9.resid)) <= 1.5)
```

```r
#creating a new subset without these observations
data_noout<- out9$model[index,]

#running same model as spec9 with new subset
out10 <- panelAR(socspend ~ socspend_lag + ma_pratio9050s + ma_pratio5010s + ma_pop65+ ma_turnout + ma_
summary(out10)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       278 Avg obs. per panel 15.4444
##  Number of panels: 18  Max obs. per panel 23
##  Number of times:  26  Min obs. per panel 1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.745e-01  8.409e-01   0.683 0.495083
## socspend_lag     9.185e-01  1.247e-02  73.630  < 2e-16 ***
## ma_pratio9050s   1.151e+00  2.533e-01   4.546 8.34e-06 ***
## ma_pratio5010s  -5.708e-01  2.229e-01  -2.561 0.011005 *
## ma_pop65        -7.825e-02  2.039e-02  -3.838 0.000155 ***
## ma_turnout       3.862e-03  3.055e-03   1.264 0.207197
## ma_fempar       -6.228e-05  6.261e-03  -0.010 0.992071
## ma_stddisp_gall -6.740e-01  1.325e-01  -5.088 6.85e-07 ***
## ma_pvoc          9.871e-03  4.010e-03   2.462 0.014458 *
## ma_union         9.622e-03  1.959e-03   4.913 1.57e-06 ***
## ma_unempl       -4.538e-02  1.510e-02  -3.006 0.002904 **
## ma_dreher        1.535e-02  3.784e-03   4.056 6.57e-05 ***
## gdpgrowth       -1.959e-01  1.429e-02 -13.709  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9965
## Wald statistic: 70536.626, Pr(>Chisq(12)): 0
```

Specification 11 (no controls):

```r
out11 <- panelAR(socspend ~ ma_pratio9050s + ma_pratio5010s +gdpgrowth + as.factor(id), data=data_socsp
summary(out11)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       320 Avg obs. per panel 17.7778
##  Number of panels: 18  Max obs. per panel 25
##  Number of times:  27  Min obs. per panel 2
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)      6.82494    3.14797   2.168 0.030944 *
## ma_pratio9050s   5.61222    2.25360   2.490 0.013305 *
## ma_pratio5010s  -3.35524    1.71780  -1.953 0.051727 .
## gdpgrowth       -0.11175    0.02693  -4.149 4.35e-05 ***
## as.factor(id)3   8.49026    0.87559   9.697  < 2e-16 ***
## as.factor(id)4   3.69566    1.14482   3.228 0.001384 **
## as.factor(id)5   8.02785    0.94621   8.484 1.02e-15 ***
## as.factor(id)6   6.21078    1.68864   3.678 0.000279 ***
## as.factor(id)7   4.42905    0.92705   4.778 2.78e-06 ***
## as.factor(id)8   4.04811    0.75597   5.355 1.71e-07 ***
## as.factor(id)9   1.25074    0.86405   1.448 0.148799
## as.factor(id)12  7.32814    2.11335   3.468 0.000602 ***
## as.factor(id)14  7.07454    0.92688   7.633 3.11e-13 ***
## as.factor(id)15  9.68632    1.19093   8.133 1.13e-14 ***
## as.factor(id)16  1.93336    1.63398   1.183 0.237662
## as.factor(id)17  3.30033    0.82090   4.020 7.36e-05 ***
## as.factor(id)18 -2.45947    0.95226  -2.583 0.010276 *
## as.factor(id)19  0.21730    0.85844   0.253 0.800337
## as.factor(id)20  0.15436    1.03638   0.149 0.881698
## as.factor(id)22 -2.56812    0.77894  -3.297 0.001095 **
## as.factor(id)23  4.03562    0.88948   4.537 8.27e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9644
## Wald statistic: 7847.8845, Pr(>Chisq(20)): 0
```

Specification 12 (no controls, no outliers):

```
#defining outliers
mod11.resid <- out11$residuals
index <- which(abs((mod11.resid-mean(mod11.resid))/sd(mod11.resid)) <= 1.5)
#creating a new subset without these observations
data_noout<- out11$model[index,]
#running same model as spec11 with new subset
out12 <- panelAR(socspend ~ ma_pratio9050s + ma_pratio5010s +gdpgrowth+ as.factor(id), data=data_noout,
summary(out12)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       283 Avg obs. per panel 15.7222
##  Number of panels: 18  Max obs. per panel 25
##  Number of times:  27  Min obs. per panel 2
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      6.88605    3.17857   2.166 0.031182 *
## ma_pratio9050s   6.09251    1.93819   3.143 0.001862 **
## ma_pratio5010s  -3.51681    1.30628  -2.692 0.007555 **
## gdpgrowth       -0.10622    0.02149  -4.943 1.37e-06 ***
```

```
## as.factor(id)12  6.94961    2.63433   2.638 0.008836 **
## as.factor(id)14  6.53944    0.85450   7.653 3.77e-13 ***
## as.factor(id)15  9.26637    0.90913  10.193  < 2e-16 ***
## as.factor(id)16  1.30411    1.53199   0.851 0.395407
## as.factor(id)17  2.60879    0.75916   3.436 0.000685 ***
## as.factor(id)18 -3.21497    0.95750  -3.358 0.000903 ***
## as.factor(id)19 -0.40735    0.80432  -0.506 0.612968
## as.factor(id)20 -0.65959    1.02258  -0.645 0.519474
## as.factor(id)22 -3.12773    1.05574  -2.963 0.003331 **
## as.factor(id)23  3.38521    0.99412   3.405 0.000765 ***
## as.factor(id)3   7.89684    0.81165   9.729  < 2e-16 ***
## as.factor(id)4   3.36569    0.99503   3.383 0.000828 ***
## as.factor(id)5   7.66790    0.75996  10.090  < 2e-16 ***
## as.factor(id)6   5.14304    0.78005   6.593 2.36e-10 ***
## as.factor(id)7   4.07394    0.87690   4.646 5.37e-06 ***
## as.factor(id)8   3.37266    0.71089   4.744 3.44e-06 ***
## as.factor(id)9   0.55493    0.87630   0.633 0.527113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.981
## Wald statistic: 9064.0524, Pr(>Chisq(20)): 0
```

Specification 13 (using skew as main inequality measure):

```r
out13<- panelAR(socspend ~ socspend_lag + ma_pratio9010s + ma_skews + ma_pop65+ ma_turnout + ma_fempar +
summary(out13)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       311 Avg obs. per panel 17.2778
##  Number of panels: 18  Max obs. per panel 24
##  Number of times:  26  Min obs. per panel 2
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.610886   0.943473  -0.647 0.517816
## socspend_lag     0.902839   0.019613  46.034  < 2e-16 ***
## ma_pratio9010s   0.301425   0.138033   2.184 0.029762 *
## ma_skews         1.742371   0.524366   3.323 0.001002 **
## ma_pop65        -0.091851   0.031669  -2.900 0.004005 **
## ma_turnout       0.006614   0.004044   1.635 0.103034
## ma_fempar        0.003970   0.008222   0.483 0.629555
## ma_stddisp_gall -0.884103   0.231822  -3.814 0.000166 ***
## ma_pvoc          0.020638   0.006801   3.035 0.002622 **
## ma_union         0.010319   0.002857   3.612 0.000356 ***
## ma_unempl       -0.068237   0.020075  -3.399 0.000768 ***
## ma_dreher        0.009557   0.006588   1.451 0.147932
## gdpgrowth       -0.183262   0.017776 -10.310  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9906
## Wald statistic: 26942.4294, Pr(>Chisq(12)): 0
```

Specification 14 (skew as main measure, no outliers):

```
mod13.resid <- out13$residuals
index <- which(abs((mod13.resid-mean(mod13.resid))/sd(mod13.resid)) <= 1.5)
#creating a new subset without these observations
data_noout<- out13$model[index,]
#running same model as spec13 with new subset
out14<- panelAR(socspend ~ socspend_lag + ma_pratio9010s + ma_skews + ma_pop65+ ma_turnout + ma_fempar +
summary(out14)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       277 Avg obs. per panel 15.3889
##  Number of panels: 18  Max obs. per panel 23
##  Number of times:  26  Min obs. per panel 1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.410050   0.700699  -0.585 0.558912
## socspend_lag     0.913948   0.012345  74.034  < 2e-16 ***
## ma_pratio9010s   0.146015   0.086176   1.694 0.091374 .
## ma_skews         1.651670   0.310761   5.315 2.27e-07 ***
## ma_pop65        -0.076433   0.019826  -3.855 0.000145 ***
## ma_turnout       0.003861   0.003097   1.247 0.213577
## ma_fempar       -0.001290   0.006147  -0.210 0.833903
## ma_stddisp_gall -0.663355   0.133772  -4.959 1.27e-06 ***
## ma_pvoc          0.008948   0.003822   2.341 0.019957 *
## ma_union         0.009306   0.001942   4.791 2.77e-06 ***
## ma_unempl       -0.045883   0.015125  -3.034 0.002658 **
## ma_dreher        0.015510   0.003745   4.142 4.65e-05 ***
## gdpgrowth       -0.196880   0.014378 -13.693  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9966
## Wald statistic: 69497.4805, Pr(>Chisq(12)): 0
```

Specification 15 (skew as main measure, no controls, country fixed effects):

```
out15 <- panelAR(socspend ~ ma_pratio9010s + ma_skews +gdpgrowth+ as.factor(id), data=data_socspend, pa
summary(out15)
```

```
##
```

```
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       320 Avg obs. per panel 17.7778
##  Number of panels: 18  Max obs. per panel 25
##  Number of times:  27  Min obs. per panel 2
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.81381    4.18187  -0.434 0.664795
## ma_pratio9010s    0.93795    0.51466   1.822 0.069384 .
## ma_skews          9.66842    3.16309   3.057 0.002440 **
## gdpgrowth        -0.11299    0.02671  -4.230 3.10e-05 ***
## as.factor(id)3    8.36931    0.80804  10.358  < 2e-16 ***
## as.factor(id)4    3.46661    1.00353   3.454 0.000631 ***
## as.factor(id)5    7.89565    0.88363   8.935  < 2e-16 ***
## as.factor(id)6    5.89641    1.69937   3.470 0.000598 ***
## as.factor(id)7    3.96228    0.90508   4.378 1.66e-05 ***
## as.factor(id)8    3.92978    0.68943   5.700 2.87e-08 ***
## as.factor(id)9    0.99141    0.80535   1.231 0.219280
## as.factor(id)12   7.37395    2.00964   3.669 0.000288 ***
## as.factor(id)14   7.24239    0.84316   8.590 4.90e-16 ***
## as.factor(id)15   9.30916    1.17764   7.905 5.20e-14 ***
## as.factor(id)16   1.86124    1.52031   1.224 0.221824
## as.factor(id)17   3.25743    0.78237   4.164 4.10e-05 ***
## as.factor(id)18  -2.83392    0.94752  -2.991 0.003013 **
## as.factor(id)19  -0.10936    0.84557  -0.129 0.897178
## as.factor(id)20  -0.29498    1.00838  -0.293 0.770089
## as.factor(id)22  -2.74804    0.72589  -3.786 0.000185 ***
## as.factor(id)23   4.10152    1.07787   3.805 0.000172 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9612
## Wald statistic: 8018.629, Pr(>Chisq(20)): 0
```

Specification 16 (skew as main measure, no controls, fixed effects without outliers):

```r
mod15.resid <- out15$residuals
index <- which(abs((mod15.resid-mean(mod15.resid))/sd(mod15.resid)) <= 1.5)
#creating a new subset without these observations
data_noout<- out15$model[index,]
#running same model as spec15 with new subset
out16 <- panelAR(socspend ~ ma_pratio9010s + ma_skews +gdpgrowth + as.factor(id), data=data_noout, panel
summary(out16)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       284 Avg obs. per panel 15.7778
##  Number of panels: 18  Max obs. per panel 25
```

```
##  Number of times:  27  Min obs. per panel 2
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.94597    3.44117  -0.275 0.783611
## ma_pratio9010s   1.05307    0.50644   2.079 0.038554 *
## ma_skews         9.10277    2.42079   3.760 0.000209 ***
## gdpgrowth       -0.10680    0.02082  -5.129 5.67e-07 ***
## as.factor(id)12  7.07737    1.99530   3.547 0.000461 ***
## as.factor(id)14  6.72084    0.77274   8.697 3.75e-16 ***
## as.factor(id)15  9.05316    0.87198  10.382  < 2e-16 ***
## as.factor(id)16  1.22785    1.39584   0.880 0.379852
## as.factor(id)17  2.53526    0.71553   3.543 0.000468 ***
## as.factor(id)18 -3.61639    0.94062  -3.845 0.000151 ***
## as.factor(id)19 -0.58198    0.80306  -0.725 0.469276
## as.factor(id)20 -0.98717    0.98957  -0.998 0.319403
## as.factor(id)22 -3.23900    0.87000  -3.723 0.000241 ***
## as.factor(id)23  3.45391    1.36255   2.535 0.011829 *
## as.factor(id)3   7.87045    0.75014  10.492  < 2e-16 ***
## as.factor(id)4   2.76669    0.88374   3.131 0.001941 **
## as.factor(id)5   7.62955    0.72092  10.583  < 2e-16 ***
## as.factor(id)6   4.93868    0.75147   6.572 2.65e-10 ***
## as.factor(id)7   3.75955    0.92569   4.061 6.44e-05 ***
## as.factor(id)8   3.31962    0.66538   4.989 1.10e-06 ***
## as.factor(id)9   0.30091    0.82451   0.365 0.715434
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9806
## Wald statistic: 8086.6194, Pr(>Chisq(20)): 0
```

Table 4 (Immigration Models)

Specification 17 (redistribution as main DV, adding `dvfpop`):

```r
out17 <- panelAR(redist ~ redist_lag + dvskew + dvratio9010 + dvturnout + dvfempar + dvstddisp_gall + dv
summary(out17)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       60 Avg obs. per panel 4.2857
##  Number of panels: 14 Max obs. per panel 7
##  Number of times:  9  Min obs. per panel 1
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -2.449e+01  1.294e+01  -1.893 0.064331 .
## redist_lag      3.275e-01  1.308e-01   2.503 0.015680 *
## dvskew          9.625e+00  4.580e+00   2.102 0.040750 *
## dvratio9010    -6.260e-02  1.386e+00  -0.045 0.964163
```

```
## dvturnout       1.853e-01  4.909e-02   3.774 0.000434 ***
## dvfempar        2.037e-01  8.772e-02   2.322 0.024425 *
## dvstddisp_gall  2.708e+00  2.171e+00   1.247 0.218247
## dvpvoc         -6.184e-04  4.875e-02  -0.013 0.989930
## dvunion         6.080e-02  4.688e-02   1.297 0.200697
## dvunempl        3.971e-01  2.011e-01   1.975 0.053954 .
## dvfpop         -1.811e-01  1.025e-01  -1.766 0.083562 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.8874
## Wald statistic: -11348.6355, Pr(>Chisq(10)): 1
```

Specification 18 (remove outliers):

```
#defining outliers
mod17.resid <- out17$residuals
index <- which(abs((mod17.resid-mean(mod17.resid))/sd(mod17.resid)) <= 1.5)
#creating a new subset without these observations
redistsample_noout<- out17$model[index,]

#running same model as spec17 with new subset
out18 <- panelAR(redist ~ redist_lag + dvskew + dvratio9010+  dvturnout + dvfempar + dvstddisp_gall + dv
summary(out18)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:       51 Avg obs. per panel 3.6429
##  Number of panels: 14 Max obs. per panel 6
##  Number of times:  9  Min obs. per panel 1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -31.291892  10.700726  -2.924 0.005661 **
## redist_lag       0.416219   0.077086   5.399 3.29e-06 ***
## dvskew          12.444536   3.451294   3.606 0.000854 ***
## dvratio9010      0.161175   1.014062   0.159 0.874516
## dvturnout        0.151332   0.039303   3.850 0.000416 ***
## dvfempar         0.250800   0.070363   3.564 0.000962 ***
## dvstddisp_gall   1.110789   1.830732   0.607 0.547448
## dvpvoc           0.003078   0.029393   0.105 0.917122
## dvunion          0.060806   0.019967   3.045 0.004101 **
## dvunempl         0.492896   0.160611   3.069 0.003848 **
## dvfpop          -0.133491   0.054575  -2.446 0.018938 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9603
## Wald statistic: -139253.65, Pr(>Chisq(10)): 1
```

Specification 19 (social spending as main DV, adding dvfpop):

```
out19<- panelAR(socspend ~ socspend_lag + ma_pratio9010s + ma_skews + ma_pop65+ ma_turnout + ma_fempar +
summary(out19)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
## Unbalanced Panel Design:
##  Total obs.:        243 Avg obs. per panel 13.5
##  Number of panels: 18  Max obs. per panel 24
##  Number of times:  26  Min obs. per panel 2
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.4918034  1.7325066  -0.861 0.390102
## socspend_lag     0.8800085  0.0232813  37.799  < 2e-16 ***
## ma_pratio9010s   0.4588034  0.2121625   2.163 0.031615 *
## ma_skews         2.9732736  0.7947753   3.741 0.000232 ***
## ma_pop65        -0.1603101  0.0376743  -4.255 3.05e-05 ***
## ma_turnout       0.0105189  0.0052708   1.996 0.047151 *
## ma_fempar        0.0015069  0.0128523   0.117 0.906769
## ma_stddisp_gall -0.9623587  0.3059579  -3.145 0.001878 **
## ma_pvoc          0.0264837  0.0086943   3.046 0.002590 **
## ma_union         0.0144852  0.0047047   3.079 0.002331 **
## ma_unempl       -0.0909643  0.0251594  -3.616 0.000369 ***
## ma_dreher        0.0131189  0.0085420   1.536 0.125964
## ma_fpop          0.0004318  0.0100739   0.043 0.965845
## gdpgrowth       -0.2155466  0.0202647 -10.637  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9922
## Wald statistic: 14615.7897, Pr(>Chisq(13)): 0
```

Specification 20 (removing outliers):

```
#defining outliers
mod19.resid <- out19$residuals
index <- which(abs((mod19.resid-mean(mod19.resid))/sd(mod19.resid)) <= 1.5)
#creating a new subset without these observations
data_noout<- out19$model[index,]

#running same model as spec19 with new subset
out20<- panelAR(socspend ~ socspend_lag + ma_pratio9010s + ma_skews + ma_pop65+ ma_turnout + ma_fempar +
summary(out20)
```

```
##
## Panel Regression with AR(1) Prais-Winsten correction and panel-corrected standard errors
##
```

```
## Unbalanced Panel Design:
##  Total obs.:       221 Avg obs. per panel 12.2778
##  Number of panels: 18  Max obs. per panel 23
##  Number of times:  26  Min obs. per panel 1
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.734822   1.255487  -1.382 0.168525
## socspend_lag     0.891614   0.012766  69.845  < 2e-16 ***
## ma_pratio9010s   0.314261   0.107916   2.912 0.003984 **
## ma_skews         2.120679   0.466053   4.550 9.12e-06 ***
## ma_pop65        -0.121411   0.024563  -4.943 1.59e-06 ***
## ma_turnout       0.014224   0.005334   2.666 0.008270 **
## ma_fempar        0.008905   0.010131   0.879 0.380438
## ma_stddisp_gall -0.700325   0.177183  -3.953 0.000106 ***
## ma_pvoc          0.018266   0.004945   3.694 0.000283 ***
## ma_union         0.007989   0.003643   2.193 0.029425 *
## ma_unempl       -0.058746   0.016147  -3.638 0.000347 ***
## ma_dreher        0.014834   0.003566   4.160 4.66e-05 ***
## ma_fpop         -0.004906   0.008180  -0.600 0.549342
## gdpgrowth       -0.230880   0.015802 -14.611  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.9961
## Wald statistic: 49261.2509, Pr(>Chisq(13)): 0
```

Table 5 (Partisanship)

The next table requires us to create a new data subset that excludes 1979 and earlier years. We repeat the previous procedures to set up the data for the models.

```
recentyears <- subset(data, year>1979)

recentyears <- recentyears[with(recentyears , order(id, year)),]

recentyears$time<- unlist(by(recentyears ,recentyears$id,function(x) seq(1:nrow(x))))
```

Specification 21 (IVs include skew, proportionality, and turnout):

```
out21<- panelAR(stdpjoint ~ ma_skews + ma_stddisp_gall + ma_turnout , data=recentyears, panelVar='id', 
summary(out21)
```

```
##
## Panel Regression with no autocorrelation and homoskedastic variance
##
## Unbalanced Panel Design:
##  Total obs.:       312 Avg obs. per panel 17.3333
##  Number of panels: 18  Max obs. per panel 25
##  Number of times:  25  Min obs. per panel 2
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.7705914  0.1488423   5.177 4.07e-07 ***
## ma_skews        -0.2368301  0.1378629  -1.718   0.0868 .
## ma_stddisp_gall -0.0549863  0.0592338  -0.928   0.3540
## ma_turnout      -0.0011143  0.0009402  -1.185   0.2369
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.0247
## Wald statistic: 7.8928, Pr(>Chisq(3)): 0.0483
```

Specification 22 (adding globalization):

```
out22<- panelAR(stdpjoint ~ ma_skews + ma_stddisp_gall + ma_turnout +ma_dreher , data=recentyears, panel
summary(out22)
```

```
##
## Panel Regression with no autocorrelation and homoskedastic variance
##
## Unbalanced Panel Design:
##  Total obs.:       312 Avg obs. per panel 17.3333
##  Number of panels: 18  Max obs. per panel 25
##  Number of times:  25  Min obs. per panel 2
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.4097101  0.1568137   8.990  < 2e-16 ***
## ma_skews        -0.2935124  0.1255806  -2.337   0.0201 *
## ma_stddisp_gall -0.0072254  0.0541960  -0.133   0.8940
## ma_turnout      -0.0008422  0.0008558  -0.984   0.3259
## ma_dreher       -0.0085208  0.0010553  -8.074 1.56e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1932
## Wald statistic: 74.7352, Pr(>Chisq(4)): 0
```

Specification 23 (adding immigration)

```
out23<- panelAR(stdpjoint ~ ma_skews + ma_stddisp_gall + ma_turnout + ma_dreher + ma_fpop , data=recent
summary(out23)
```

```
##
## Panel Regression with no autocorrelation and homoskedastic variance
##
## Unbalanced Panel Design:
##  Total obs.:       238 Avg obs. per panel 13.2222
##  Number of panels: 18  Max obs. per panel 25
```

```
##  Number of times:  25  Min obs. per panel 2
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.5604425  0.2171198   7.187 8.99e-12 ***
## ma_skews        -0.5218094  0.1572125  -3.319  0.00105 **
## ma_stddisp_gall -0.0203366  0.0609789  -0.334  0.73906
## ma_turnout       0.0007596  0.0010872   0.699  0.48546
## ma_dreher       -0.0084083  0.0014366  -5.853 1.64e-08 ***
## ma_fpop         -0.0030278  0.0026454  -1.145  0.25357
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-squared: 0.1635
## Wald statistic: 46.5121, Pr(>Chisq(5)): 0
```

```
#creating a new data frame with observations before 1980
olderyears <- subset(data, year<1980)

# creating aggdata gives an error
# aggdata <- aggregate(c(olderyears$stdpjoint, olderyears$skew, olderyears$stddisp_gall, olderyears$dre
```

Specification 24

```
# not working
out24 <- panelAR(stdpjoint ~ ma_skews + ma_stddisp_gall + ma_turnout, data=olderyears, panelVar='id', t
summary(out24)
```

Specification 25 (Specification 22 without outliers)

```
# not working
out25 <- panelAR(stdpjoint~maskews + mastddisp_gall + maturnout +madreher , data=olderyears, panelVar='
summary(out25)
```

Specification 26 (Specification 23 without outliers)

```
# not working
out26 <- panelAR(stdpjoint~maskews + mastddisp_gall + maturnout +madreher +mafpop , data=olderyears, pa
summary(out26)
```

Redistribution and Social Spending Models with partisanship measures (Table 6)

Specification 27 (Redistribution as main DV, adding dvstdpjoint)

```
# not matching
out27 <- panelAR(redist ~ redist_lag + dvskew + dvratio9010 + dvturnout + dvfempar + dvstddisp_gall + d
summary(out27)
```

Specification 28 (remove outliers)

```
# not working

#defining outliers
mod27.resid <- out27$residuals
index <- which(abs((mod27.resid-mean(mod27.resid))/sd(mod27.resid)) <= 1.5)
#creating a new subset without these observations
redistsample_noout<- out27$model[index,]

#running same model as spec17 with new subset
out28 <- panelAR(redist ~ redist.lag + dvskew + dvturnout + dvfempar + dvpropind + dvpvoc + dvunion + d
summary(out28)
```

Specification 29 (Social spending as main DV, adding dvstdpjoint)

```
# not working
out29<- panelAR(socspend ~ socspend.lag + maratio9010s + maskews + mapop65+ maturnout + mafempar + mast
summary(out29)
```

Specification 30 (removing outliers)

```
# not working
#defining outliers
mod29.resid <- out29$residuals
index <- which(abs((mod29.resid-mean(mod29.resid))/sd(mod29.resid)) <= 1.5)
#creating a new subset without these observations
redistsample_noout<- out29$model[index,]
#running same model as spec29 with new subset
out30<- panelAR(socspend ~ socspend.lag + maratio9010s + maskews + mapop65+ maturnout + mafempar + mast
summary(out30)
```

Extension

```
# not working

#Loading data that contains disaggregated data for 6 categories of social spending
disag_data <- read_dta(paste0(directory, "disag_spending.dta"))

#Linearly interpolating missing values
library('zoo')

disagdata_countries <- lapply(unique(disag_data$country), function(x)
  subset(disag_data, disag_data$country==x)
)





test <- disagdata_countries[[1]]
test2 <- md.pattern(test)

test3 <- mice(test,m=5,maxit=50,meth='pmm',seed=500)
summary(mice(test,m=5,maxit=50,meth='pmm',seed=500))

completedData <- complete(test3, 1)

for (i in 1:length(disagdata_countries)){
disagdata_countries[[i]] <- cbind(disagdata_countries[[i]], sapply(c(5:8), function(y)
   na.approx(disagdata_countries[[i]][,y], x = index(disagdata_countries[[i]][,3], disagdata_countries[
  ))
}

disag_data <- do.call("rbind", disagdata_countries)


#names(disag_data)[9:12] <- c("fambenefits_v2", "incapacity_v2", "pubspending_labor_v2", "public_unemp_

#rm(disagdata_countries)
```