

# Replication: The Structure of Inequality and the Politics of Redistribution

*Filippo Teoldi, Zara Riaz and Julian Gerez*

*October 23rd, 2018*

## Design declaration

First we start by loading in the `DeclareDesign` package and defining the elements of the design.

- `declare_population` refers to the sample size of the study. The study concerns country-year units. In this case, there are 858 observations.
- `delclare_potential_oucomes` refers to

```
library('DeclareDesign')
```

```
## Loading required package: randomizr
```

```
## Loading required package: fabricatr
```

```
## Loading required package: estimatr
```

```
design <-  
  declare_population(N = 858) +  
  declare_potential_outcomes() +  
  declare_estimand() +  
  declare_assignment() +  
  declare_estimator()
```

## Replication

First we open the dataset with the `haven` package, which allows us to open `.dta` files.

```
library('haven')  
directory <- "/Users/juliangerez/Google Drive/Semester_Fall 2018/Political Economy of Development/Lupu-  
data <- read_dta(paste0(directory, "LupPon_APSR.dta"))
```

Next, the authors define invert disproportionality measures, `disp_gall` as such:

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

Then the variables female participation, `fempar`, and annual net union density, `union` so that they are rescaled:

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

Then the variables `pjoint` and `disp_gall`, which are partisanship and disproportionality, respectively 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.

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

```
data$stdpjoint <- range01(data$pjoint)
data$stdpdisp_gall <- range01(data$disp_gall)
```

Next, we interpolate missing values:

```
library('zoo')
```

```
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##      as.Date, as.Date.numeric
```

```
data$pratio9050 <- NA
```

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

```
for (i in 1:length(data_countries)){
  data_countries[[i]][,24] <- na.approx(data_countries[[i]][,5], x = index(data_countries[[i]][,3], data_
})
```

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

```
data$pratio5010 <- NA
```

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

```
for (i in 1:length(data_countries)){
  data_countries[[i]][,25] <- na.approx(data_countries[[i]][,6], x = index(data_countries[[i]][,3], data_
})
```

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

```
data$pratio9050s <- NA
```

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

```
for (i in 1:length(data_countries)){
  data_countries[[i]][,26] <- na.approx(data_countries[[i]][,7], x = index(data_countries[[i]][,3], data_
})
```

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

```
data$pratio5010s <- NA
```

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

```

for (i in 1:length(data_countries)){
data_countries[[i]][,27] <- na.approx(data_countries[[i]][,8], x = index(data_countries[[i]][,3], data_
})

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

data$pforeign <- NA

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

for (i in 1:length(data_countries)){
data_countries[[i]][,28] <- na.approx(data_countries[[i]][,16], x = index(data_countries[[i]][,3], data_
})

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

data$pvoc <- NA

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

for (i in 1:length(data_countries)){
data_countries[[i]][,29] <- na.approx(data_countries[[i]][,19], x = index(data_countries[[i]][,3], data_
})

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

```

We generate an immigration measure, pforeign which reflects the percentage of the population that is foreign-born:

```

data$pforeign <- data$pforeign*1000
data$fpop <- data$pforeign/data$pop

```

Generate inequality measures

```

data$ratio9010 <- data$pratio9050*data$pratio5010
data$ratio9010s <- data$pratio9050s*data$pratio5010s
data$skew <- data$pratio9050/data$pratio5010
data$skews <- data$pratio9050s/data$pratio5010s

```

Data cleaning

Redistribution models

Social spending models

Immigration

Partisanship

Redistribution and social spending with partisanship

Robustness checks via design modification

Extension