

MP_to_ESS

2025-04-22

Manifesto Project to European Social Survey

This repository is to help students merge the longitudinal Manifesto Project data to the European Social Survey.

```
library(manifestoR)
```

```
## Loading required package: NLP
```

```
## Loading required package: tm
```

```
## When publishing work using the Manifesto Corpus, please make sure to cite it correctly and to give t
```

```
##
```

```
## You can print citation and version information with the function mp_cite().
```

```
##
```

```
## Note that some of the scaling/analysis algorithms provided with this package were conceptually devel
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v dplyr      1.1.4      v readr      2.1.5
```

```
## v forcats    1.0.0      v stringr    1.5.1
```

```
## v ggplot2    3.5.1      v tibble     3.2.1
```

```
## v lubridate  1.9.3      v tidyr      1.3.1
```

```
## v purrr      1.0.2
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x ggplot2::annotate() masks NLP::annotate()
```

```
## x dplyr::filter()      masks stats::filter()
```

```
## x dplyr::lag()         masks stats::lag()
```

```
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

This code starts to download the data directly into R

```
mp_setapikey("manifesto_apikey.txt")
```

Download the data from the manifesto project into R what it returns is a dataframe where each row is a country-year-party observation so there are rows in every group for all parties in that election year, then the indicator

```
mpds<-mp_maindataset(version = "current", south_america =FALSE, download_format = NULL, apikey=NULL, ca
```

```
## Connecting to Manifesto Project DB API...
```

```
## Connecting to Manifesto Project DB API... corpus version: 2024-1
```

This code just fixes the date so that it is a 'year' measurement make the date only 4 digits

```
mpds<-mpds %>%
  mutate(date= (date*.01)) %>%
  mutate(date=round(date))
```

This selects the NWOL variables and filters the dataset to all elections in 1990 or after

Each row is a party in an election in a year

In the 'select' part, you need to add in the other variables in the MP data you are interested in so you can use it later. Here I filter the data so that it only includes parties in countries with the Radical Right party family, that are in the EU and after 1990.

```
mpds<-mpds %>%
  select(country, countryname, date, party, per601, per501, per608, absseat, totseats, eumember, parfam)
  filter(date> 1990 & parfam== 70 & eumember>1)
```

There is a small issue, Portugal only has one observation of a Nationalist party. All the other countries have at least 2

```
table(mpds$countryname)
```

```
##
##      Austria      Belgium      Bulgaria      Croatia      Cyprus
##          10           9           5           5           5
## Czech Republic  Denmark      Estonia      Finland      France
##           4          10           2           7           6
##      Germany      Greece      Hungary      Italy      Latvia
##           3          13           4          13           8
##    Lithuania  Netherlands      Poland      Portugal      Romania
##           3           11           2           1           3
##      Serbia      Slovakia      Slovenia      Spain      Sweden
##           5          11           5           2           4
##      Turkey
##           7
```

Lets filter portugal out of the analysis also then

```
mpds<- mpds %>% filter(countryname != 'Portugal')
```

This code makes a variable for the share of seats in parlment for each party in each election

```
mpds$seat_share<-(mpds$absseat/mpds$totseats)*100
```

This code makes a variable for each parties' score on per601 weighted by their share of parliment

```
mpds$per601_size<-mpds$per601*mpds$seat_share
mpds$per608_size<-mpds$per608*mpds$seat_share
mpds$per501_size<-mpds$per501*mpds$seat_share
```

The problem with this data is that the election years don't match with the ESS rounds. This means we need to get values for the years when the ESS was fielded. To do this I expand the dataframe so it includes a full sequence of years from 1990-2023

Notice I make it into a new data frame called 'mpdf1'.

```
mpds1<- mpds %>% complete(countryname, date= full_seq(date, period= 1), fill= list(per601_size= NA))
```

Then I interpolate the missing values for each year. Basically this is where the year would fall on a line if you connected the two observed values it falls between.

For this, you will need to install and load the imputeTS package

```
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method      from
##   as.zoo.data.frame zoo
```

```
mpds1<-mpds1 %>%
  group_by(countryname) %>%
  mutate(per601_size= imputeTS::na_interpolation(per601_size)) %>%
  mutate(per608_size= imputeTS::na_interpolation(per608_size)) %>%
  mutate(per501_size= imputeTS::na_interpolation(per501_size))
```

Check how many country-year observations you have. This code will show what countries have more than 1 country year observation, but notice it does NOT save over the dataframe. It shows that there are several country-year observations with more than one nationalist party.

```
mpds1 %>% group_by(countryname, date) %>%
  filter(n()>1)
```

```
## # A tibble: 61 x 15
## # Groups:   countryname, date [27]
##   countryname date country party per601 per501 per608 absseat totseats
##   <chr>      <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1 Austria    2006     42 42420 0      15.1  10.2    21     183
## 2 Austria    2006     42 42710 3.58   3.23  4.48     7     183
## 3 Austria    2008     42 42420 1.30   4.33  5.84    34     183
## 4 Austria    2008     42 42710 0.291  4.07  5.23    21     183
## 5 Belgium    2014     21 21710 5.07   3.10  3.60     1     150
## 6 Belgium    2014     21 21917 7.47   1.31  5.64     3     150
## 7 Croatia    2020     81 81711 6.08   1.97  0        66     151
## 8 Croatia    2020     81 81714 6.62   4.30  0        16     151
## 9 Cyprus     2016     55 55711 2.96   0      0        18     56
## 10 Cyprus    2016     55 55720 16.3   0      0         2     56
## # i 51 more rows
## # i 6 more variables: eumember <dbl>, parfam <dbl>, seat_share <dbl>,
## #   per601_size <dbl>, per608_size <dbl>, per501_size <dbl>
```

To do what Micke advised which would be to average the nationalist parties. You would do this by using something like the following code. It makes a new variable called 'per601_size_mean', but you can change it to whatever you want.

This code also selects only the country, date, and indicator columns. Then the distinct command removes the duplicate rows.

```
mpds1<-mpds1%>% group_by(countryname, date) %>% mutate(per601_size_mean= mean(per601_size)) %>%
  mutate(per608_size_mean= mean(per608_size)) %>%
  mutate(per501_size_mean= mean(per501_size)) %>%
  select(countryname, date, per601_size_mean, per608_size_mean, per501_size_mean) %>%
  distinct()
```

Now we can check again if there are any country years with more than 1 observation:

```
mpds1 %>% group_by(countryname, date) %>%
  filter(n()>1)
```

```
## # A tibble: 0 x 5
## # Groups:   countryname, date [0]
## # i 5 variables: countryname <chr>, date <dbl>, per601_size_mean <dbl>,
## #   per608_size_mean <dbl>, per501_size_mean <dbl>
```

Now that you have a tidy data set for the macro variable you will do the demeaning procedure.

For this bit of code you will need to install and load the datawizard package

This line of code makes a new dataframe with meaned (between) and demeaned (within) values.

```
library(datawizard)
```

```
##
## Attaching package: 'datawizard'
```

```
## The following object is masked from 'package:manifestoR':
##
##   rescale
```

```
mpds_demeaned<- demean(mpds1, select = c("per601_size_mean", "per608_size_mean", "per501_size_mean"), by=
```

```
#This next chunk of code is an example of how to check if the country names are spelled the same way
#and which countries overlap #First I cross reference what countries are overlapping
```

```
mpds_all[mpds_all$countrynameCOUNTRY NAME VARIABLE]
```

```
##Fix the country names that were not overlapping, this is an example of code for that. I think this is all
the 4 that need it.
```

```
mpds$countryname <-recode(mpds$countryname, 'Czech Republic'= 'Czechia', 'Russia'= 'Russian Federation',
'Slovakia'= 'Slovak Republic', 'Turkey'= 'Turkiye' )
```

```
#Check again to see what countries are overlapping, you should see the ones you fixed show up
mpds_all[mpds_all$countrynameCOUNTRY NAME VARIABLE]
```

```
#Then merge the macrodata back to the ess file ess_merged<- left_join(ESSDATA, mpds_all, by=
c('COUNTRY NAME VARIABLE'= 'countryname', 'ESSYEAR'= 'year'))
```

#Then here you will need to match the nationalist party family list with the individual respondents. I don't have code #for that. Give it some thought but if you get stuck we can talk about it.

#Modeling #For this you'll need to install and/or load the lme4 and lmerTest packages

```
m <- lmer(dependent_variable ~ x_between + x_within + (1 + x_within |
```

#Then I like to report results using the sjPlot package. There is a ton of useful stuff in there. For example if you want a #nice table of the results you can use:

```
tab_model(m)
```