Module 4 - Instructions

Oliver Engist

04/04/2021

In the next assignment we want to replicate some plots from the paper "Female Socialization: How Daughters Affect Their Legislator Fathers' Voting on Women's Issues" (Washington, 2008). The paper explores whether having a daughter makes politicians more sensitive to women's rights issues and how this is reflected in their voting behavior. The main identifying assumption is that after controlling for the number of children, the gender composition is random. This might be violated if families that have a preference for girls keep having children until they have a girl. In this assignment we will prepare a dataset that allows us to test whether families engage in such a "female child stopping rule".

Setup

- Load the libraries "Rio" and "tidyverse"
- Change the path of the working directory to your working directory.

```r
# Suggested code chunks containing R code for solving the questions by
# Carl Edvin Steinvall, 23830.

# Loading the libraries "Rio" and "tidyverse"
library(rio)
library(tidyverse)

# Setting the working directory.
setwd("/Users/carledvinsteinvall/Desktop/7316 R Course/Assignments/Assignment 4/Module_4")
```

- import the data sets *basic.dta* and *genold108.dta*
- create a subset of the 108th congress from the *basic* dataset
- join this subset with the *genold* dataset

```r
# Importing the two data sets and assigning them to two different objects.
basic_dta <- import("basic.dta")
genold_dta <- import("genold108.dta")

# Creating a subset of the 108th congress from the basic dataset using filter()
# with the logical statement 'congress == 108' as argument. Assigning the subset
# to a new object.
basic_108th <- basic_dta %>%
  filter(congress == 108)

# Joining the datasets using the function left_join().
data <- left_join(basic_108th, genold_dta)
```

Data preparation

- check table 1 in the appendix of the paper and decide which variables are necessary for the analysis (check the footnote for control variables)
- drop all other variables.

```
# Starting off by looking at the data set and having a glimpse at the variables.
# glimpse(data)

# The variables that are kept for the analysis are used as arguments in select().
# I assign these changes to the same data object.
data <- data %>%
  select(genold, ngirls, nboys, totchi, white, female, party, age, srvlng, rgroup, region)

# converting to tibble
data <- data %>% as_tibble()

# Having a look at the new data object.
# str(data)
```

- Recode *genold* such that gender is a factor variable and missing values are coded as NAs.

```
# Replacing "" with NAs for the variable genold.
data$genold <- data$genold %>% na_if("")

# Recoding genold as a factor variable.
data <- data %>%
  mutate(genold = factor(genold))

# Checking if the recoding of genold seems correct.
str(data$genold)
```

```
##  Factor w/ 2 levels "B","G": NA NA 2 1 NA 2 2 NA NA NA ...
```

- Recode *party* as a factor with 3 levels (D, R, I)

```
# Looking at the structure of the variable party.
# str(data$party)

# Observing that an observation of 1 corresponds to Democrat, 2 to Republican and
# 3 to independent. Applying this to the vector argument of levels. Further, the
# corresponding labels wanted are "D", "R" and "I".
data <- data %>%
  mutate(party = factor(party, levels = c(1, 2, 3), labels = c("D", "R", "I")))

str(data$party)
```

```
##  Factor w/ 3 levels "D","R","I": 1 1 2 2 1 1 1 1 2 1 ...
```

- Recode *rgroup* and *region* as factors.

```
# Recoding the variable rgroup as factor (assigning labels based on description
# in the column of the tibble)
data$rgroup <- data$rgroup %>%
  factor(levels = c(0:4), labels = c("None", "Prot", "Cath/Orth", "Othchr", "Jewish"))

str(data$rgroup)
```

```
##  Factor w/ 5 levels "None","Prot",..: 1 5 2 2 2 2 2 3 2 2 ...
```

```
# Recoding the variable region as factor. No arguments needed.
data$region <- data$region %>% factor()

str(data$region)
```

```
##  Factor w/ 9 levels "1","2","3","4",..: 9 2 6 4 7 1 2 9 6 9 ...
```

- generate variables for age squared and service length squared

```
# The variables for age squared and service length squared are created using
# mutate().
data <- data %>%
  mutate(agesq = age^2) %>%
  mutate(srvlngsq = srvlng^2)
```

- create an additional variable of the number of children as factor variable

```
# Creating a variable of number of children as factor variable.
data <- data %>%
  mutate(nchildren = factor(totchi))

str(data$nchildren)
```

```
##  Factor w/ 12 levels "0","1","2","3",..: 1 4 2 7 4 3 3 5 6 1 ...
```

Replicationg Table 1 from the Appendix

We haven't covered regressions in R yet. Use the function *lm()*. The function takes the regression model (formula) and the data as an input. The model is written as $y \sim x$, where x stands for any linear combination of regressors (e.g. $y \sim x_1 + x_2 + female$). Use the help file to understand the function.

- Run the regression $total.children = \beta_0 + \beta_1 gender.oldest + \gamma' X$ where γ stands for a vector of coefficients and X is a matrix that contains all columns that are control variables.[1]
- Save the main coefficient of interest (β_1)

```
# Regression of the variable totchi on genold and the covariates from Table 1
# in the appendix.
regression <- lm(totchi ~ genold + white + female + party + age + agesq +
                   srvlng + srvlngsq + rgroup + region, data = data)

# Saving the parameter estimate by assigning it to an object.
beta1_full <- regression$coefficients["genoldG"]
beta1_full
```

```
##     genoldG
## -0.08388331
```

- Run the same regression separately for Democrats and Republicans (assign the independent to one of the parties). Save the coefficient and standard error of *genold*

```
# Recoding observations of independents so that they count to the Democratic party.
# The rows in the data object are selected by a logical statement and the column
# selected is the seventh column which is the column for party.
data[data$party=="I",7] <- "D"
```

[1] This is just a short notation instead of writing the full model with all control variables $totchi = \beta_0 + \beta_1 genold + \gamma_1 age + \gamma_2 age^2 + \gamma_3 Democrat + ... + \epsilon$ which quickly gets out of hand for large models.

```
# Assigning the Democrat subset of observations to a new object.
data_dem <- data %>%
  filter(party == "D")

# Running the same regression as previously, but by just applying the subsetted
# data set with observations of democrats. Removing party from the covariates.
regression_dem <- lm(totchi ~ genold + white + female + age + agesq +
                   srvlng + srvlngsq + rgroup + region, data = data_dem)

# Saving the parameter estimate by assigning it to an object.
beta1_dem <- regression_dem$coefficients["genoldG"]
beta1_dem
```

```
##    genoldG
## 0.07291434
```

```
# Assigning the Republican subset of observations to a new object (like I did
# for the democratic subset).
data_rep <- data %>% filter(party == "R")

# Running the regression for Republicans similar to the regression for democrats.
regression_rep <- lm(totchi ~ genold + white + female + age + agesq +
                   srvlng + srvlngsq + rgroup + region, data = data_rep)

# Saving the parameter estimate by assigning it to an object.
beta1_rep <- regression_rep$coefficients["genoldG"]
beta1_rep
```

```
##    genoldG
## -0.2823933
```

- Collect all the *genold* coefficients from the six regressions, including their standard errors and arrange them in a table as in the paper.

```
# Full congress
# Extracting the standard error of genold from the full congress regression.
sum_full <- summary(regression)
std_full <- sum_full$coefficients[2,2]

# Removing object.
rm("sum_full")

# Running the regression for ngirls on genold and covariates for full congress.
# According to the notes of the table in the appendix, the number of daughters
# regressions also include fixed effects for total number of children. So, I add
# the variable of number of children as a factor variable (nchildren) created
# before.
regression_full_ngirls <- lm(ngirls ~ genold + white + female + party + age +
                            agesq + srvlng + srvlngsq + rgroup + region +
                            nchildren, data = data)

# Saving the parameter estimate by assigning it to a new object.
beta1_full_ngirls <- regression_full_ngirls$coefficients["genoldG"]

# Extracting the std. of genold like before.
```

```r
sum_full_ngirls <- summary(regression_full_ngirls)
std_full_ngirls <- sum_full_ngirls$coefficients[2,2]

# Removing object.
rm("sum_full_ngirls")
```

```r
# Democrats
# Repeating the above on the Democratic party subset.
# Extracting the standard error of genold from the Democratic party regression.
sum_dem <- summary(regression_dem)
std_dem <- sum_dem$coefficients[2,2]

# Removing object
rm("sum_dem")

# Running the regression for ngirls on genold and covariates for Democratic
# subset. I omit the party variable like previously.
regression_dem_ngirls <- lm(ngirls ~ genold + white + female + age + agesq +
                            srvlng + srvlngsq + rgroup + region + nchildren,
                            data = data_dem)

# Saving the parameter estimate by assigning it to an object.
beta1_dem_ngirls <- regression_dem_ngirls$coefficients["genoldG"]

# Extracting the std. error of genold like before.
sum_dem_ngirls <- summary(regression_dem_ngirls)
std_dem_ngirls <- sum_dem_ngirls$coefficients[2,2]

# Removing object
rm("sum_dem_ngirls")
```

```r
# Republicans
# Repeating the above on the Republican subset.
sum_rep <- summary(regression_rep)
std_rep <- sum_rep$coefficients[2,2]

# Removing unnecessary object
rm("sum_rep")

# Running the regression for ngirls on genold and covariates for Republican subset.
regression_rep_ngirls <- lm(ngirls ~ genold + white + female + age + agesq +
                            srvlng + srvlngsq + rgroup + region + nchildren,
                            data = data_rep)

# Saving the parameter estimate.
beta1_rep_ngirls <- regression_rep_ngirls$coefficients["genoldG"]

# Extracting the std. of genold like before.
sum_rep_ngirls <- summary(regression_rep_ngirls)
std_rep_ngirls <- sum_rep_ngirls$coefficients[2,2]

# Removing object
rm("sum_rep_ngirls")
```

```
# Table
# Setting up the table:
# Assigning the estimated betas and standard errors to two vectors (ordered the
# same way as in the table from the appendix)
beta <- c(beta1_full_ngirls, beta1_full, beta1_dem_ngirls, beta1_dem,
            beta1_rep_ngirls, beta1_rep)
se <- c(std_full_ngirls, std_full, std_dem_ngirls, std_dem,
          std_rep_ngirls, std_rep)

# Combining the two vectors by row and assigning them to a new object.
table <- rbind(beta, se)

# Rounding the elements to two digits as in the Table 1 in the appendix.
table <- round(table, 2)

# Assigning names to the rows and columns
rownames(table) <- c("1st_child_female", "Standard_err")
colnames(table) <- c("ngirls_full", "totchi_full", "ngirls_dem", "totchi_dem",
                     "ngirls_rep", "totchi_rep")
```

- print the table

```
print(table)
```

```
##                  ngirls_full totchi_full ngirls_dem totchi_dem ngirls_rep
## 1st_child_female        1.36       -0.08       1.39       0.07       1.23
## Standard_err            0.08        0.15       0.11       0.18       0.11
##                  totchi_rep
## 1st_child_female      -0.28
## Standard_err           0.23
```