Manipulation of Variables

Pedro Girardi

December 11, 2018

```
library(tidyverse)
library(data.table)
library(lme4)
```

Importing STATA Exported .csvs and selecting variables of interest

```
setwd("PME_stata/")#set directory to where the STATA csv files are
listOfCsv = list.files()# get a list of all files

# create data frame with one of the files (Selecting for the variables of interest)
stataCsv = fread(file = listOfCsv[[1]]) %>%
  select(v035, v040, v050, v070, v075, v203, v409, v4191, v208,
         v301, v307, v234, idind)

# runs for loop that reads one file and appends rows of columns of interest to previous df
for (fileCsv in listOfCsv[-1]){
  currentFile = fread(file = fileCsv) %>%
    select(v035, v040, v050, v070, v075, v203, v409, v4191, v208,
           v301, v307, v234, idind)
  stataCsv = bind_rows(stataCsv, currentFile)
}
fwrite(stataCsv, "finalPME.csv")
```

```
finalPME = fread("finalPME.csv")

# Sample 10% of households and select important columns
hhSample = sample(finalPME$idind, length(unique(finalPME$idind))/20, replace = F)
reducedPME = finalPME %>% filter(idind %in% hhSample)
fwrite(reducedPME, file = "reducedPME.csv")
```

```
reducedPME = fread("reducedPME.csv") %>%
  mutate(v035 = as.factor(v035))
original_reducedPME = reducedPME
```

Merging Dataset with Macro Data

```
macro.df = fread("macrodata.csv") %>%
  mutate(Region = as.factor(Region))
metropolitanVectorNames = list(26, 29, 31, 33, 35, 43)
names(metropolitanVectorNames) = unique(macro.df$Region)[unique(macro.df$Region)!="Brasil"]
levels(reducedPME$v035) = names(metropolitanVectorNames)
reducedPME = left_join(reducedPME, macro.df, by=c("v035"="Region", "v070"="Month", "v075"="Year"))
```

Flagging low income people

```
reducedPME = reducedPME %>%
  filter(!is.na(v4191),
         v4191!=999999999) %>%
  mutate(realWage = v4191/PriceIndex) %>%
  group_by(month_year = as.factor(paste0(as.character(v070),as.character(v075)))) %>%
  mutate(PoorThreshold = quantile(realWage, probs=.2),
         Poor = as.factor(ifelse(realWage<PoorThreshold, T, F)))

lowIncomeIdInd = reducedPME %>%
  filter(Poor==T)
lowIncomeIdInd = unique(lowIncomeIdInd$idind)

length(unique(reducedPME$idind))
reducedPME = reducedPME %>%
  filter(idind %in% lowIncomeIdInd)
length(unique(reducedPME$idind))
# About 1/3 of people are flagged as "poor"
```

Working with income variable

```
reducedPME = reducedPME %>%
                filter(realWage > 0) %>%  #filters wage of 0
                group_by(idind) %>%
                mutate(lagRealWage = lag(realWage),
                       pctChangeWage = (realWage-lagRealWage)/(lagRealWage))
# Check why most people have ~2.4 obs
```

Checking issues with not enough observations

```
numberIdind = reducedPME %>%
  group_by(idind) %>%
  summarise(n = n())
numberIdind %>% group_by(n) %>% summarise(n()/nrow(numberIdind))

numberIdind_full = original_reducedPME %>%
  filter(!is.na(v4191)) %>%
  group_by(idind) %>%
  summarise(n = n(), maxWage = max(v4191, na.rm = T))
wagesPropr_numberObs = numberIdind_full %>% group_by(n) %>% summarise(prop=n()/nrow(numberIdind_full),
                                                      meanMaxWage = mean(maxWage, na.rm = T))
# check if mean income varies by different number of observations
ggplot(data=wagesPropr_numberObs,aes(x=n)) +
  geom_col(aes(y=meanMaxWage)) +
  theme_minimal()

ggplot(data=wagesPropr_numberObs,aes(x=n)) +
  geom_col(aes(y=prop)) +
  theme_minimal()
```

Runs the regression

```
# should plot a sample of 100, maybe 150 people with their logWage vs inflationTax points

lm1 = lm(data=reducedPME, pctChangeWage ~ Inflation + lagInflation + ChangeInGDP + v035)
summary(lm1)

lm2 = update(lm1, realWage ~.)
summary(lm2)

lm3 = update(lm1, log(realWage) ~ .)
summary(lm3)

lm4 = lm(data=reducedPME, log(realWage) ~ inflationTax + lagInflationTax + ChangeInGDP + v035)
summary(lm4)

plot(lm4)
plot(cooks.distance(lm4) ~ reducedPME$realWage)
```

```
# Trying to fit LMER models
lmer1 <- lmer(log(realWage) ~ v035 + lagInflation + ChangeInGDP + (lagInflation | idind), data = reduce
summary(lmer1)

# check if I need random slopes!!!

# what happens if there are lots of individual observations?
```

Writing full dataset

```
fwrite(reducedPME, "complete.csv")
```