Exam2_Taeyoung Lee

Taeyoung Lee

6/26/2020

Contents

Questions

1. Please clear the environment in R.

```
# clear the environment
rm(list=ls())
```

2. Load the "inequality" dataset into R, and save the data frame as 'inequality_data'.

```
# Load the dataset and save data frame
library(rio)
inequality_data = import("inequality.xlsx", which = 1)
```

3. Is this dataset a cross-sectional or panel dataset? Explain why in words and provide some R code to prove that your answer is correct.

This data set is a cross-section dataset because it only has 2015 data.

```
# See summary of year
summary(inequality_data$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2015    2015    2015    2015    2015    2015
```

4.The data frame contains a variable called inequality_gini. It corresponds to the inequality Gini index, which "measures the extent to which the distribution of income (or, in some cases, consumption expenditure) among individuals or households within an economy deviates from a perfectly equal distribution." In simple terms, there is a lot of inequality when there are a lot of rich people and a lot of poor people but not a lot of middle-class people. There is low inequality when most people are earning about the same amount of income. Scandinavian countries like Sweden and Denmark tend to have the most optimal Gini index scores. Using the subset command, provide the inequality_gini scores for Denmark and Sweden.

```r
# Check the inequality score for Demark
subset(inequality_data, country=="Denmark")
```

```
##    iso2c country inequality_gini year
## 40    DK Denmark            28.2 2015
```

```r
# Check the inequality score for Sweden
subset(inequality_data, country=="Sweden")
```

```
##     iso2c country inequality_gini year
## 174    SE  Sweden            29.2 2015
```

5. Since Brazil started the Bolsa Familia conditional cash transfer program in 1990s, inequality in Brazil has decreased significantly. Just the same, inequality in Brazil is very high comparatively. Using the subset command, please show the inequality_gini score for Brazil.

```r
# Check the inequality score for Brazil
subset(inequality_data, country=="Brazil")
```

```
##    iso2c country inequality_gini year
## 13    BR  Brazil            51.9 2015
```

6. Given your answers to the previous questions, is it better to have a high or low inequality_gini scores?

It is better to have a low inequality_gini score because Scandinavian countries like Sweden and Denmark that have low inequality show lower scores than Brazil.

7. Use the head command to get a quick peak at the data frame.

```r
# Take a quick peak
head(inequality_data)
```

```
##   iso2c country inequality_gini year
## 1    AL Albania            32.9 2015
## 2    AM Armenia            32.4 2015
## 3    AT Austria            30.5 2015
## 4    BY Belarús            25.6 2015
## 5    BE Belgium            27.7 2015
## 6    BZ  Belize              NA 2015
```

8. Write a function called “accent.remove” to remove the accent on Belarus, apply that function, and run the head command again to show that you removed the accent.

```r
# Remove accent for Belarus
accent.remove <- function(s)  {
  old1 <- "ú"
  new1 <- "u"
  s1 <- chartr(old1, new1, s)
}
```

```r
# finish the accent FIX
inequality_data$country = accent.remove(inequality_data$country)

# Take a quick peak at the data
head(inequality_data)
```

```
##   iso2c country inequality_gini year
## 1    AL Albania            32.9 2015
## 2    AM Armenia            32.4 2015
## 3    AT Austria            30.5 2015
## 4    BY Belarus            25.6 2015
## 5    BE Belgium            27.7 2015
## 6    BZ  Belize              NA 2015
```

9. Sort the data by the countries with the lowest inequality_gini scores and then run the head command again to show what the top 5 countries are.

```r
# Sort the data
inequality_data = inequality_data[order(inequality_data$inequality_gini),]

# Check the top 5 countries
head(inequality_data, n =5)
```

```
##     iso2c        country inequality_gini year
## 161    SI       Slovenia            25.4 2015
## 190    UA        Ukraine            25.5 2015
## 4      BY        Belarus            25.6 2015
## 39     CZ Czech Republic            25.9 2015
## 92     XK         Kosovo            26.5 2015
```

10. What is the mean inequality_gini score? Provide the relevant R code.

```r
# Check the mean
mean(inequality_data$inequality_gini, na.rm = TRUE)
```

```
## [1] 36.81375
```

11. Using the ifelse command, create two new dummy variables, high_inequality and low_inequality, which takes values of either zero or one. The low_inequality variable should correspond to countries with inequality_gini scores below the mean. The high_inequality variable should correspond to countries with inequality_gini scores above the mean. (Note: we will not accept answers that do not use the ifelse command to create the variables.)

```r
# Creating dummy variables-- using ifelse
inequality_data$low_inequality = ifelse(inequality_data$inequality_gini < 36.81, yes = 1, no = 0 )
inequality_data$high_inequality = ifelse(inequality_data$inequality_gini > 36.81, yes = 1, no = 0 )

# Take a quick peak
head(inequality_data)
```

```
##     iso2c         country inequality_gini year low_inequality high_inequality
## 161    SI        Slovenia            25.4 2015              1               0
## 190    UA         Ukraine            25.5 2015              1               0
## 4      BY         Belarus            25.6 2015              1               0
## 39     CZ  Czech Republic            25.9 2015              1               0
## 92     XK          Kosovo            26.5 2015              1               0
## 160    SK Slovak Republic            26.5 2015              1               0
```

12. Run a cross-tab using the high_inequality and low_inequality variables that you created in the previous question. The cross-tab should provide the mean inequality_gini score and number of observations for each category of inequality. (Note: if you had trouble using the ifelse command, we couldn't provide points for the previous question. However, you can create the variables using the indexing method)

```
# Running a cross-tab
library(doBy)
```

```
## Warning: package 'doBy' was built under R version 3.6.2
```

```
summaryBy(high_inequality ~ low_inequality, data=inequality_data, FUN=c(mean,length))
```

```
##   low_inequality high_inequality.mean high_inequality.length
## 1              0                    1                     34
## 2              1                    0                     46
## 3             NA                   NA                    123
```

13. The World Bank, the African Development Bank, and the Bill and Melinda Gates Foundation are all working on reducing inequality in Africa. Write a for loop that prints the names of these three actors. (Note: we will not accept answers that do not provide a for loop.)

```
# Create an organization vector
orgs <- c('World Bank', 'the African Development Bank', 'the Bill and Melinda Gates Foundation')

#create the for loop
for (i in orgs){
print (i)
}
```

```
## [1] "World Bank"
## [1] "the African Development Bank"
## [1] "the Bill and Melinda Gates Foundation"
```

14. Use this website to find a variable from the World Development Indicators that you think is correlated with inequality. Tell us what variable you picked and why you picked it. (Don't worry if your prediction is not correct. We just want you to provide some rationale.)

I think the variable "Income share held by highest 20%" will be highly correlated with inequality because the more income share the highest 20% hold, the higher inequlity score would be.

15. Import that variable directly into R. (Note: if you are having trouble, read Mike Denly's Canvas announcement from the other day.)

```r
# add some data from the World Development Indicators (WDI)
remotes::install_github('vincentarelbundock/WDI')
```

```
## Skipping install of 'WDI' from a github remote, the SHA1 (5b516c96) has not changed since last instal
##   Use `force = TRUE` to force installation
```

```r
# WDI package?
library(WDI)

head(WDI)
```

```
##
## 1 function (country = "all", indicator = "NY.GDP.PCAP.KD", start = 1960,
## 2     end = 2020, extra = FALSE, cache = NULL)
## 3 {
## 4     if (!is.character(country)) {
## 5         stop("The `country` argument must be a character vector")
## 6     }
```

```r
# Import the variable

income_data = WDI(country = "all",
    indicator = c("SI.DST.05TH.20"), # indicator from web
    start = 2015, end = 2015, extra = FALSE, cache = NULL)
```

16. Rename the variable that you imported into something that we can actually understand.

```r
# Rename variables
library(data.table)
setnames(income_data,"SI.DST.05TH.20", "income_share")

# Take a quick peak
head(income_data)
```

```
##   iso2c                                    country income_share year
## 1    1A                                 Arab World           NA 2015
## 2    S3                      Caribbean small states           NA 2015
## 3    B8               Central Europe and the Baltics           NA 2015
## 4    V2                   Early-demographic dividend           NA 2015
## 5    Z4                           East Asia & Pacific           NA 2015
## 6    4E East Asia & Pacific (excluding high income)           NA 2015
```

17. Merge the new variable into the other dataset, using inequality_data as the x and and your new data frame as the y. When merging use the command that only keeps the rows in your x data frame. Call your new data frame merged_df. Ensure that you have no variables with .x or .y at the end of them in your new merged_df, while at the same time ensuring there are still variables like country and year.

```r
# Merging data with a left-join
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## v ggplot2 3.3.1     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::between()   masks data.table::between()
## x dplyr::filter()    masks stats::filter()
## x dplyr::first()     masks data.table::first()
## x dplyr::lag()       masks stats::lag()
## x dplyr::last()      masks data.table::last()
## x dplyr::order_by()  masks doBy::order_by()
## x purrr::transpose() masks data.table::transpose()
```

```
merged_df = left_join(x=inequality_data,
                      y=income_data,
                      by =c("country", "year"))

# Check the variable names of new data frame
names(merged_df)
```

```
## [1] "iso2c.x"         "country"         "inequality_gini" "year"
## [5] "low_inequality"  "high_inequality" "iso2c.y"         "income_share"
```

```
# Check if the country for iso2c.x and iso2c.y are matched
library(tidyverse)

merged_df<-
  merged_df %>%
  mutate(iso2c_match = ifelse(iso2c.x == iso2c.y,
                              "yes",
                              "no"))

# drop iso2c.x and rename iso2c.y
merged_df <-
  merged_df %>%
  select(-c("iso2c.x")) %>%
  rename("iso2c" = "iso2c.y")
```

18. In merged_df, remove the missing data on the basis of inequality_gini and your new variable that you took from the World Development Indicators.

```r
# Remove missing values
na.omit(merged_df, select=c("inequality_gini", "income_share"))
```

```
##                country inequality_gini year low_inequality high_inequality iso2c
## 1             Slovenia            25.4 2015              1                0    SI
## 2              Ukraine            25.5 2015              1                0    UA
## 3              Belarus            25.6 2015              1                0    BY
## 4       Czech Republic            25.9 2015              1                0    CZ
## 5               Kosovo            26.5 2015              1                0    XK
## 6      Slovak Republic            26.5 2015              1                0    SK
## 7              Iceland            26.8 2015              1                0    IS
## 8           Kazakhstan            26.8 2015              1                0    KZ
## 9              Moldova            27.0 2015              1                0    MD
## 10             Finland            27.1 2015              1                0    FI
## 11              Norway            27.5 2015              1                0    NO
## 12             Belgium            27.7 2015              1                0    BE
## 13             Denmark            28.2 2015              1                0    DK
## 14         Netherlands            28.2 2015              1                0    NL
## 15     Kyrgyz Republic            29.0 2015              1                0    KG
## 16              Sweden            29.2 2015              1                0    SE
## 17               Malta            29.4 2015              1                0    MT
## 18             Hungary            30.4 2015              1                0    HU
## 19             Austria            30.5 2015              1                0    AT
## 20             Croatia            31.1 2015              1                0    HR
## 21             Germany            31.7 2015              1                0    DE
## 22   Egypt, Arab Rep.             31.8 2015              1                0    EG
## 23             Ireland            31.8 2015              1                0    IE
## 24              Poland            31.8 2015              1                0    PL
## 25         Switzerland            32.3 2015              1                0    CH
## 26             Armenia            32.4 2015              1                0    AM
## 27             Estonia            32.7 2015              1                0    EE
## 28              France            32.7 2015              1                0    FR
## 29             Tunisia            32.8 2015              1                0    TN
## 30             Albania            32.9 2015              1                0    AL
## 31      United Kingdom            33.2 2015              1                0    GB
## 32            Pakistan            33.5 2015              1                0    PK
## 33          Luxembourg            33.8 2015              1                0    LU
## 34              Cyprus            34.0 2015              1                0    CY
## 35          Tajikistan            34.0 2015              1                0    TJ
## 36              Latvia            34.2 2015              1                0    LV
## 37            Ethiopia            35.0 2015              1                0    ET
## 38               Italy            35.4 2015              1                0    IT
## 39            Portugal            35.5 2015              1                0    PT
## 40     North Macedonia            35.6 2015              1                0    MK
## 41         Gambia, The            35.9 2015              1                0    GM
## 42             Romania            35.9 2015              1                0    RO
## 43              Greece            36.0 2015              1                0    GR
## 44            Thailand            36.0 2015              1                0    TH
## 45               Spain            36.2 2015              1                0    ES
## 46             Georgia            36.5 2015              1                0    GE
## 47           Lithuania            37.4 2015              0                1    LT
## 48               Tonga            37.6 2015              0                1    TO
## 49 Russian Federation             37.7 2015              0                1    RU
```

```
## 50            Myanmar           38.1 2015              0              1    MM
## 51           Bulgaria           38.6 2015              0              1    BG
## 52              China           38.6 2015              0              1    CN
## 53         Montenegro           39.0 2015              0              1    ME
## 54 Iran, Islamic Rep.           39.5 2015              0              1    IR
## 55            Uruguay           40.1 2015              0              1    UY
## 56             Serbia           40.5 2015              0              1    RS
## 57        El Salvador           40.6 2015              0              1    SV
## 58              Kenya           40.8 2015              0              1    KE
## 59          Indonesia           41.0 2015              0              1    ID
## 60           Malaysia           41.0 2015              0              1    MY
## 61      Cote d'Ivoire           41.5 2015              0              1    CI
## 62         Cabo Verde           42.4 2015              0              1    CV
## 63             Turkey           42.9 2015              0              1    TR
## 64               Togo           43.1 2015              0              1    TG
## 65               Peru           43.4 2015              0              1    PE
## 66              Chile           44.4 2015              0              1    CL
## 67        Philippines           44.4 2015              0              1    PH
## 68 Dominican Republic           45.2 2015              0              1    DO
## 69            Ecuador           46.0 2015              0              1    EC
## 70            Bolivia           46.7 2015              0              1    BO
## 71           Paraguay           47.6 2015              0              1    PY
## 72              Benin           47.8 2015              0              1    BJ
## 73         Costa Rica           48.4 2015              0              1    CR
## 74           Honduras           49.6 2015              0              1    HN
## 75             Panama           50.8 2015              0              1    PA
## 76           Colombia           51.1 2015              0              1    CO
## 77             Brazil           51.9 2015              0              1    BR
## 78           Botswana           53.3 2015              0              1    BW
## 79             Zambia           57.1 2015              0              1    ZM
## 80            Namibia           59.1 2015              0              1    NA
##    income_share iso2c_match
## 1          35.1         yes
## 2          35.6         yes
## 3          35.5         yes
## 4          35.9         yes
## 5          36.1         yes
## 6          35.0         yes
## 7          36.5         yes
## 8          36.9         yes
## 9          36.9         yes
## 10         36.7         yes
## 11         36.6         yes
## 12         36.5         yes
## 13         37.7         yes
## 14         37.4         yes
## 15         38.8         yes
## 16         37.6         yes
## 17         38.1         yes
## 18         38.4         yes
## 19         38.4         yes
## 20         38.4         yes
## 21         39.7         yes
## 22         41.5         yes
```

```
## 23         40.2         yes
## 24         39.3         yes
## 25         40.2         yes
## 26         40.7         yes
## 27         40.4         yes
## 28         40.9         yes
## 29         40.9         yes
## 30         40.8         yes
## 31         40.6         yes
## 32         42.8         yes
## 33         41.0         yes
## 34         42.1         yes
## 35         41.7         yes
## 36         41.5         yes
## 37         43.0         yes
## 38         41.4         yes
## 39         42.7         yes
## 40         41.1         yes
## 41         43.6         yes
## 42         40.8         yes
## 43         41.8         yes
## 44         43.8         yes
## 45         42.1         yes
## 46         43.4         yes
## 47         44.2         yes
## 48         45.4         yes
## 49         45.3         yes
## 50         45.7         yes
## 51         44.3         yes
## 52         45.4         yes
## 53         44.3         yes
## 54         46.4         yes
## 55         46.0         yes
## 56         45.0         yes
## 57         47.2         yes
## 58         47.5         yes
## 59         48.4         yes
## 60         47.3         yes
## 61         47.8         yes
## 62         48.7         yes
## 63         49.2         yes
## 64         48.6         yes
## 65         48.7         yes
## 66         51.2         yes
## 67         50.9         yes
## 68         51.1         yes
## 69         51.3         yes
## 70         51.1         yes
## 71         52.7         yes
## 72         52.1         yes
## 73         53.9         yes
## 74         54.0         yes
## 75         55.3         yes
## 76         55.9         yes
```

```
## 77          56.8          yes
## 78          58.5          yes
## 79          61.3          yes
## 80          63.7          yes
```

19. Using the filter command and piping method, only keep the data with inequality_gini scores greater than 30. Save the new data frame as data_greater_30. (Note: we will not accept answers using subset.)

```
# Filter out thevariables
library(tidyverse)
data_greater_30 <-
  merged_df %>%  # pipe(%>%):
  dplyr::filter(!(inequality_gini <= 30))
```

20. Using data_greater_30, use to R to count how many countries have the sequence "ai" in their name.

```
grep("ai", data_greater_30)
```

```
## [1] 1
```

21. Use any command from the apply family to take the sum of inequality_gini in data_greater_30.

```
sapply(data_greater_30$inequality_gini,sum)
```

```
##  [1] 30.4 30.5 31.1 31.7 31.8 31.8 31.8 32.3 32.4 32.7 32.7 32.8 32.9 33.2 33.5
## [16] 33.8 34.0 34.0 34.2 35.0 35.4 35.5 35.6 35.9 35.9 36.0 36.0 36.2 36.5 37.4
## [31] 37.6 37.7 38.1 38.6 38.6 39.0 39.5 40.1 40.5 40.6 40.8 41.0 41.0 41.5 42.4
## [46] 42.9 43.1 43.4 44.4 44.4 45.2 46.0 46.7 47.6 47.8 48.4 49.6 50.8 51.1 51.9
## [61] 53.3 57.1 59.1
```

22. Label your variables in merged_df. Any labels will suffice.

```
#label the data
library(labelled)
```

```
## Warning: package 'labelled' was built under R version 3.6.2
```

```
# Drop countries match
merged_df$iso2c_match = NULL


var_label(merged_df) <- list(`country` = "Country",
                             `year` = "Year",
                             `inequality_gini` = "Inequality Gini Score",
                             `low_inequality` = "Inequality Gini Score < 36.81",
                             `high_inequality` = "Inequality Gini Score > 36.81",
                             `iso2c`= "ISO-2 Country Code")
```

23. Save the labeled data frame as a Stata dataset called final_data.

```r
# Save the data frame as a Stata dataset
library(rio)
export(merged_df, "cleaned_dataset.dta")
```

24. Save all of the files (i.e. .Rmd, .dta, .xlsx, .pdf/Word Doc), push them to your GitHub repo, and provide us with the link to that repo.

Git hup repo: https://github.com/taeyoung-lee/exam2