Exam2

Rohin Balkundi

6/26/2020

### 1. Clearing the environment in R

rm(list=ls(all=TRUE))

### 2. Importing the dataset

library(rio)  
inequality\_data = import("inequality.xlsx")

### 3.Is the data a cross sectional or panel dataset?

The dataset is a cross sectional dataset since all the data is collecting from a specific year (2015), this was confirmed when taking a quick peek at the 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

### 4.Providing the scores for Denmark and Sweden

subset(inequality\_data, country == "Denmark"| country == "Sweden")

## iso2c country inequality\_gini year  
## 40 DK Denmark 28.2 2015  
## 174 SE Sweden 29.2 2015

The scores are 28.2 and 29.2 for Denmark and Swweden respectively.

### 5.Score for Brazil

subset(inequality\_data, country == "Brazil")

## iso2c country inequality\_gini year  
## 13 BR Brazil 51.9 2015

As seen above, the score for Brazil is 51.9

### 6. Higher or lower inequality gini score?

It is better to have a lower inequality gini score since that way the distribution of income is more central and overall would lead to less equality.

### 7. Quick Peek at the data frame

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.Writing the command

Writing the command

#writing a command to remove the accent  
accent.remove <- function(x){  
 ##for 1 character substitutions  
 old1 <- "ú"  
 new1 <- "u"  
 ###use chartr to make the replacements  
 s1 <- chartr(old1,new1,x)  
}

Running the command on the country column

##Running the command on the country column   
inequality\_data$country=accent.remove(inequality\_data$country)

Checking the data to ensure the accent is gone

###Checking 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 and run the head command again

####9. sort the gini score  
inequality\_data=inequality\_data[order(inequality\_data$inequality\_gini),]  
 ## Check the head to see the top 5 countries  
 head(inequality\_data)

## 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  
## 160 SK Slovak Republic 26.5 2015

### 10. find the mean inequality\_gini score

### Mean gini score  
 mean(inequality\_data$inequality\_gini, na.rm=TRUE) ##Noticed there were NA's in the dataset

## [1] 36.81375

### 11. Using ifelse to create dummy variables

##Creating the variable  
inequality\_data$high\_inequality = "NA"  
# using ifelse to fill the column in  
inequality\_data$high\_inequality =ifelse(test = inequality\_data$inequality\_gini > 36.81375, yes = 1, no = 0)  
  
  
#Creating the variable low inequality  
inequality\_data$low\_inequality = "NA"  
inequality\_data$low\_inequality =ifelse(test = inequality\_data$inequality\_gini > 36.81375, yes = 0, no = 1)

### 12. Run a cross tab

# Load doby  
library(doBy)  
#Run the cross tab  
summaryBy(inequality\_gini ~ high\_inequality, data=inequality\_data, FUN=c(mean,length))

## high\_inequality inequality\_gini.mean inequality\_gini.length  
## 1 0 31.25870 46  
## 2 1 44.32941 34  
## 3 NA NA 123

### 13. Creating a loop to print names

#Creating an organization vector   
 orgs <- c('World Bank','African Development Bank','Bill and Melinda Gates foundation')  
 #Creating the loop to print the orgs  
 for (i in orgs){  
 print(i)  
 }

## [1] "World Bank"  
## [1] "African Development Bank"  
## [1] "Bill and Melinda Gates foundation"

### 14. Choose a variable

I chose the poverty headcount ratio at $1.90 a day, I think it is a good measure of inequality since it would show how the level of poverty in a region relative to the purchasing power parity of 1.90 a day and would give me a percentage.

### 15. Import the Variable

#Loading WDI  
library(WDI)  
poverty\_data = WDI(country= "all",  
 indicator = c("SI.POV.DDAY"), ## name of the indicator  
 start = 2015, end = 2015, extra = FALSE, cache = NULL) ##want it to be 2015 since the other data is 2015 as well

### 16. Rename the Variable

#Load  
library(data.table) #loading data table  
setnames(poverty\_data,"SI.POV.DDAY","Poverty\_headcount\_ratio") ## Renaming  
head(poverty\_data) #checking to make sure it goes through

## iso2c country Poverty\_headcount\_ratio  
## 1 1A Arab World NA  
## 2 S3 Caribbean small states NA  
## 3 B8 Central Europe and the Baltics NA  
## 4 V2 Early-demographic dividend NA  
## 5 Z4 East Asia & Pacific 2.3  
## 6 4E East Asia & Pacific (excluding high income) NA  
## year  
## 1 2015  
## 2 2015  
## 3 2015  
## 4 2015  
## 5 2015  
## 6 2015

### 17. Merge

## want to do a left join so X can keep all of its rows  
library(tidyverse)

## -- Attaching packages ----------------------------------------------------------------------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 1.0.0  
## v tidyr 1.0.0 v stringr 1.4.0  
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'dplyr' was built under R version 3.6.3

## -- Conflicts -------------------------------------------------------------------------------------------------------- tidyverse\_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 purrr::transpose() masks data.table::transpose()

merged\_df = left\_join(x= inequality\_data,  
 y= poverty\_data,  
 by = c("iso2c","year") #used iso2c since both datasets had them  
 )  
##Creating a variable that sees when names match and when they dont  
merged\_df <-  
 merged\_df %>%  
 mutate(countries\_match = ifelse(country.x == country.y,  
 "yes",  
 "no"))  
#Check for countries that dont match  
subset(merged\_df, countries\_match =="no")

## [1] iso2c country.x inequality\_gini   
## [4] year high\_inequality low\_inequality   
## [7] country.y Poverty\_headcount\_ratio countries\_match   
## <0 rows> (or 0-length row.names)

### All countries match, so I can get rid of either column, choose to get rid of Y  
merged\_df <-  
 merged\_df %>%  
 select(-c("country.y")) %>% # drop country.y  
 rename("country" = "country.x")  
## drop the countries match variable  
merged\_df$countries\_match = NULL  
## take a peak at the data to make sure its correct  
head(merged\_df)

## iso2c country inequality\_gini year high\_inequality low\_inequality  
## 1 SI Slovenia 25.4 2015 0 1  
## 2 UA Ukraine 25.5 2015 0 1  
## 3 BY Belarus 25.6 2015 0 1  
## 4 CZ Czech Republic 25.9 2015 0 1  
## 5 XK Kosovo 26.5 2015 0 1  
## 6 SK Slovak Republic 26.5 2015 0 1  
## Poverty\_headcount\_ratio  
## 1 0.0  
## 2 0.1  
## 3 0.0  
## 4 0.0  
## 5 0.4  
## 6 0.7

### 18.Remove the NA’s

###18. Remove NA's   
merged\_df <- na.omit(merged\_df, select = c("inequality\_gini", "Poverty\_headcount\_ratio"))  
### check for NA  
is.na(merged\_df$inequality\_gini)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

is.na(merged\_df$Poverty\_headcount\_ratio)

## [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

### 19 Filter for greater than 30

library(tidyverse)  
data\_greater\_30 <-  
 merged\_df %>%   
 dplyr::filter((inequality\_gini > 30)) ###Filter out any of these that have a gini score lower than 30  
 # quick check to make sure they are greater than 30  
 head(data\_greater\_30)

## iso2c country inequality\_gini year high\_inequality low\_inequality  
## 1 HU Hungary 30.4 2015 0 1  
## 2 AT Austria 30.5 2015 0 1  
## 3 HR Croatia 31.1 2015 0 1  
## 4 DE Germany 31.7 2015 0 1  
## 5 EG Egypt, Arab Rep. 31.8 2015 0 1  
## 6 IE Ireland 31.8 2015 0 1  
## Poverty\_headcount\_ratio  
## 1 0.6  
## 2 0.7  
## 3 0.6  
## 4 0.0  
## 5 1.3  
## 6 0.3

### 20. count how many countries have the sequence “ai” in it

### counting the number of "ai" in countries  
length(grep("ai",data\_greater\_30$country))

## [1] 2

### 21.take the sum of inequality gini)

sum(sapply(data\_greater\_30$inequality\_gini, sum))

## [1] 2482.8

### 22.rename the variables

library(labelled)

## Warning: package 'labelled' was built under R version 3.6.3

var\_label(merged\_df) <- list(`country` = "Country",  
 `year` = "year",  
 `inequality\_gini` = "Gini inequality score",  
 `high\_inequality` = "Dummy Variable if the Gini score is higher than the mean",  
 `low\_inequality` = "Dummy Variable if the Gini score is lower than the mean",  
 `iso2c`= "ISO-2 Country Code",  
 `Poverty\_headcount\_ratio` = "Poverty Headcount Ratio")

### 23. export

### 23, save merged df as a stata fle   
library(rio)  
export(merged\_df, file = "final\_data.dta")

### 24 push on github will be on canvas.