Homework 5

Ankur Patel

November 3, 2020

Problem 3:

```
library(tidyverse)
```

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

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
edstats <- read.table("EdStatsData.csv", header = TRUE, sep = ",")
edstats_tidy <- gather(edstats,key = "year", value = "Estimation",c(5:69))
#We choose Turkey and Vietnam for the two countries.
edstats_trimmed <- subset(edstats_tidy, ï..Country.Name == c("Turkey","Vietnam"))
#We extract the unique indicator codes and we will do a summary table based
#on the first 4
unique_indicators <- unique(edstats_trimmed$Indicator.Code)
indicators_touse <- unique_indicators[1:4]
#edstats_tosummarize is the dataframe that we form from edstats_trimmed
#by subsetting on the 4 indicator codes
edstats_tosummarize <- subset(edstats_trimmed, Indicator.Code == indicators_touse)
#The next two dataframes are what we use to make the summaries and is formed
#by subetting edstats_tosummarize on country
edstats_tosummarize_Turkey <- subset(edstats_tosummarize, ï..Country.Name == "Turkey")
edstats_tosummarize_Vietnam <- subset(edstats_tosummarize, ï..Country.Name == "Vietnam")
#estimation_matrix will be a dataframe with row 1 = Turkey, row 2 = Vietnam
#and the columns being the indicator codes
estimation_matrix <- matrix(nrow = 2, ncol = 4)
for (i in 1:4)
{
  #the current indicator
  curr_indicator <- indicators_touse[i]
  #current data based off the current indicator for both Turkey and Vietnam
  curr_data_Turkey <- subset(edstats_tosummarize_Turkey,Indicator.Code == curr_indicator)
```

```
  curr_data_Vietnam <- subset(edstats_tosummarize_Vietnam,Indicator.Code == curr_indicator)
  #fill in the estimation_matrix
  estimation_matrix[1,i] <- mean(curr_data_Turkey$Estimation, na.rm = TRUE)
  estimation_matrix[2,i] <- mean(curr_data_Vietnam$Estimation, na.rm = TRUE)
}
#convert estimation_matrix to data frame and then give row names, column #names and form a table
estimation_matrix <- data.frame(estimation_matrix)
row.names(estimation_matrix) <- c("Turkey","Vietnam")
colnames(estimation_matrix) <- indicators_touse
knitr::kable(estimation_matrix)
```

	UIS.NERA.2.F	UIS.NERA.2.M	SE.PRM.TENR.FE	SE.PRM.TENR.MA
Turkey	77.12232	86.61464	94.16762	96.44012
Vietnam	NaN	NaN	NaN	NaN

There were 886930 points in the complete dataset. After tidying the data, there were 57650450 observations. For Vietnam, there were no values for the indicator codes we selected.

Problem 4:

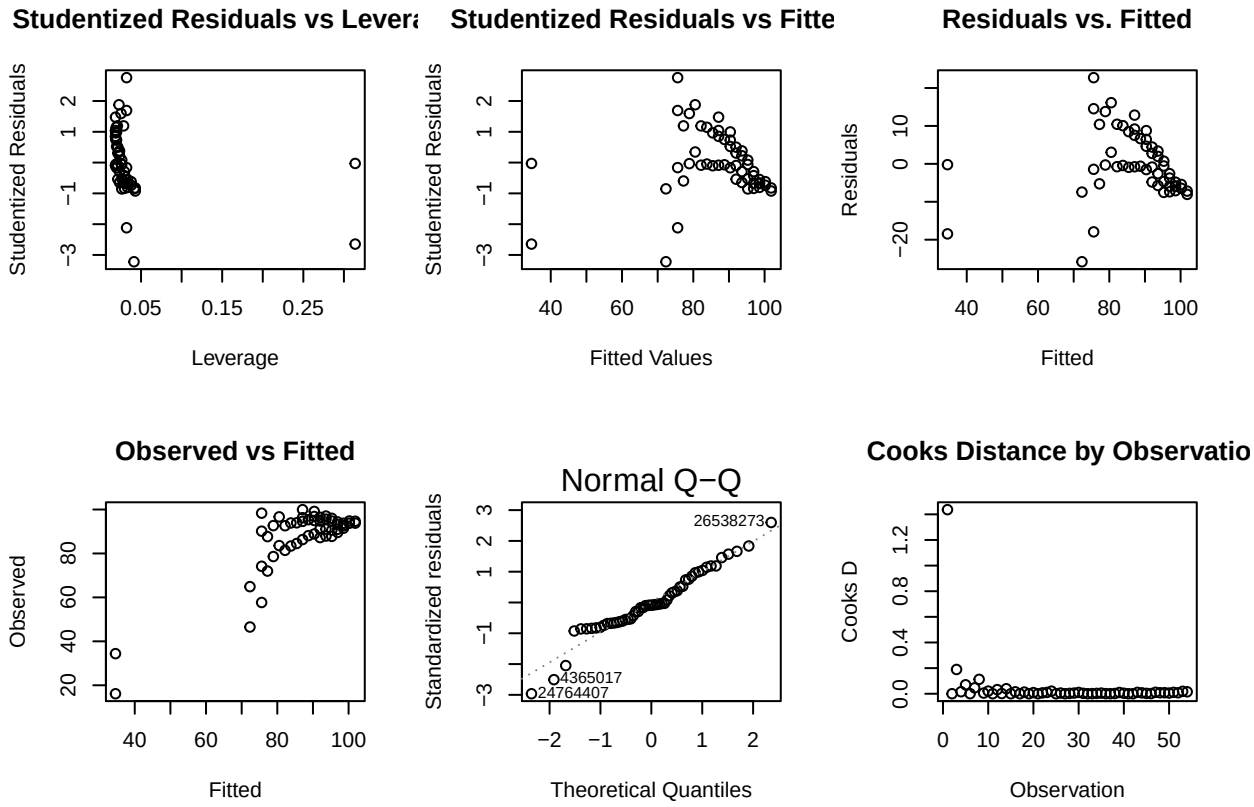
```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
#make a function to clean up the year column
convert_YearString <- function(y)
{
  y_new <- as.numeric(substr(y,start = 2, stop = 5))
  return(y_new)
}
#remove NAs where estimation is missing
edstats_tosummarize_Turkey <- edstats_tosummarize_Turkey[complete.cases(edstats_tosummarize_Turkey$Esti
#clean the year column and assign it to the dataframe we want to work with,
#in this case edstats_tosummarize_Turkey
dim_edstats_trimmed_Turkey <- dim(edstats_tosummarize_Turkey)
year_cleaned <- vector(length = dim_edstats_trimmed_Turkey[1])
for (i in 1:dim_edstats_trimmed_Turkey[1])
{
  year_cleaned[i] <- convert_YearString(edstats_tosummarize_Turkey$year[i])
}
year_cleaned <- year_cleaned[1:dim_edstats_trimmed_Turkey[1]]
edstats_tosummarize_Turkey$year <- year_cleaned
#fit the linear model and do some plots
```

```r
lm_Turkey <- lm(Estimation ~ year, data = edstats_tosummarize_Turkey)
par(mfrow = c(2,3))
#Studentized Residuals vs leverage
plot(studres(lm_Turkey)~hatvalues(lm_Turkey), xlab = "Leverage", ylab = "Studentized Residuals", main =
#Studentized Residuals vs Fitted
plot(studres(lm_Turkey)~predict(lm_Turkey), xlab = "Fitted Values", ylab = "Studentized Residuals", mai
#Residuals vs fitted
plot(resid(lm_Turkey) ~ predict(lm_Turkey), xlab = "Fitted", ylab = "Residuals", main = "Residuals vs.
#Observed vs Fitted data
plot(edstats_tosummarize_Turkey$Estimation ~ predict(lm_Turkey), xlab = "Fitted", ylab = "Observed", ma
#Residuals vs Quantile plot
plot(lm_Turkey, which = 2)
#Cooks distance by observation
plot(cooks.distance(lm_Turkey) ~ seq(1,54), xlab = "Observation", ylab = "Cooks D", main = "Cooks Dista
```



Problem 5:

```r
#use autoplot from ggfortify to do the plots
library(ggfortify)
autoplot(lm_Turkey, which = c(1:6))
```

```
## Warning: `arrange_()` is deprecated as of dplyr 0.7.0.
## Please use `arrange()` instead.
```

```
## See vignette('programming') for more help
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_warnings()' to see where this warning was generated.
```