Homework 5

Due Wednesday Nov 4, 2020

2020-10-26

For each assignment, turn in by the due date/time. Late assignments must be arranged prior to submission. In every case, assignments are to be typed neatly using proper English in Markdown.

This week, we spoke about Exploratory Data Analysis and plotting. To begin the homework, we will as usual, start by loading, munging and creating tidy data sets. In this homework, our goal is to create informative (and perhaps pretty) plots showing features or perhaps deficiencies in the data.

Problem 1

Work through the Swirl "Exploratory_Data_Analysis" lesson parts 1 - 10. If you need some review of ggplot, see the tutorial on Rstudio.cloud.

I took above lessons.

Problem 2

Create a new R Markdown file within your local GitHub repo folder (file–>new–>R Markdown–>save as).

The filename should be: HW5_lastname, i.e. for me it would be HW5_Settlage

You will use this new R Markdown file to solve the following problems.

Problem 3

Using tidy concepts, get and clean the following data on education from the World Bank.

http://databank.worldbank.org/data/download/Edstats_csv.zip

How many data points were there in the complete dataset? In your cleaned dataset?

Choosing 2 countries, create a summary table of indicators for comparison.

```
library(tidyr)
library(tidyverse)
```

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

```
## v ggplot2 3.3.0     v dplyr   0.8.5
## v tibble  3.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
## v purrr   0.3.4
## -- Conflicts ---------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(knitr)

# Import Data
setwd("~/STAT5014_youngjin/Edstats_csv")
eco_data <- read.csv("EdStatsData.csv",skip=1,header=FALSE)

# V70 is just blank column
# So I removed it
summary(eco_data$V70)
```

```
##    Mode    NA's
## logical  886930
```

```r
eco_data <- eco_data[,-70]

# Assign column names
colnames(eco_data) <- c("Country Name","Country Code","Indicator Name","Indicator Code",
                        1970:2017,seq(2020,2100,by=5))
colnames(eco_data)
```

```
##  [1] "Country Name"   "Country Code"   "Indicator Name" "Indicator Code"
##  [5] "1970"           "1971"           "1972"           "1973"
##  [9] "1974"           "1975"           "1976"           "1977"
## [13] "1978"           "1979"           "1980"           "1981"
## [17] "1982"           "1983"           "1984"           "1985"
## [21] "1986"           "1987"           "1988"           "1989"
## [25] "1990"           "1991"           "1992"           "1993"
## [29] "1994"           "1995"           "1996"           "1997"
## [33] "1998"           "1999"           "2000"           "2001"
## [37] "2002"           "2003"           "2004"           "2005"
## [41] "2006"           "2007"           "2008"           "2009"
## [45] "2010"           "2011"           "2012"           "2013"
## [49] "2014"           "2015"           "2016"           "2017"
## [53] "2020"           "2025"           "2030"           "2035"
## [57] "2040"           "2045"           "2050"           "2055"
## [61] "2060"           "2065"           "2070"           "2075"
## [65] "2080"           "2085"           "2090"           "2095"
## [69] "2100"
```

```r
# There are so many countries and so many indicators
# I choose France and United Kingdom for comparison
# I will use first 2 indicators for comparison between France and United Kingdom
length(unique(eco_data$`Country Name`))
```

```
## [1] 242
```

```r
head(unique(eco_data$`Country Name`),10)
```

```
##  [1] "Arab World"
##  [2] "East Asia & Pacific"
##  [3] "East Asia & Pacific (excluding high income)"
##  [4] "Euro area"
##  [5] "Europe & Central Asia"
##  [6] "Europe & Central Asia (excluding high income)"
##  [7] "European Union"
##  [8] "Heavily indebted poor countries (HIPC)"
##  [9] "High income"
## [10] "Latin America & Caribbean"
```

```r
length(unique(eco_data$`Indicator Name`))
```

```
## [1] 3665
```

```r
head(unique(eco_data$`Indicator Name`),10)
```

```
##  [1] "Adjusted net enrolment rate, lower secondary, both sexes (%)"
##  [2] "Adjusted net enrolment rate, lower secondary, female (%)"
##  [3] "Adjusted net enrolment rate, lower secondary, gender parity index (GPI)"
##  [4] "Adjusted net enrolment rate, lower secondary, male (%)"
##  [5] "Adjusted net enrolment rate, primary, both sexes (%)"
##  [6] "Adjusted net enrolment rate, primary, female (%)"
##  [7] "Adjusted net enrolment rate, primary, gender parity index (GPI)"
##  [8] "Adjusted net enrolment rate, primary, male (%)"
##  [9] "Adjusted net enrolment rate, upper secondary, both sexes (%)"
## [10] "Adjusted net enrolment rate, upper secondary, female (%)"
```

```r
# Filtering data
eco_data_f <- head(eco_data %>% subset(`Country Name`=='France'),2)
eco_data_u <- head(eco_data %>% subset(`Country Name`=='United Kingdom'),2)
eco_data_f_u <- rbind(eco_data_f,eco_data_u)
eco_data_f_u <- eco_data_f_u[,-c(2,4)]

# Making summary statistics
eco_data_f_u_summary <- data.frame(eco_data_f_u[,1:2],matrix(NA,4,6))
colnames(eco_data_f_u_summary)[3:8] <- c("Min","1st Qu.","Median","Mean","3rd Qu.","Max.")
eco_data_f_u_summary[1,3:8] <- summary(as.numeric((eco_data_f_u[1,-c(1,2)])))[-7]
eco_data_f_u_summary[2,3:8] <- summary(as.numeric((eco_data_f_u[2,-c(1,2)])))[-7]
eco_data_f_u_summary[3,3:8] <- summary(as.numeric((eco_data_f_u[3,-c(1,2)])))[-7]
eco_data_f_u_summary[4,3:8] <- summary(as.numeric((eco_data_f_u[4,-c(1,2)])))[-7]

# Making summary table
eco_data_f_u_summary <- gather(eco_data_f_u_summary,key="statistic",value="value","Min","1st Qu.","Medi
eco_data_f_u_summary <- spread(data=eco_data_f_u_summary,key="Country.Name",value='value')

# Summary table
kable(eco_data_f_u_summary,caption = "Comparison between France and United Kingdom")
```

Table 1: Comparison between France and United Kingdom

Indicator.Name	statistic	France	United Kingdom
Adjusted net enrolment rate, lower secondary, both sexes (%)	1st Qu.	83.87554	91.42144
Adjusted net enrolment rate, lower secondary, both sexes (%)	3rd Qu.	93.85780	97.48468
Adjusted net enrolment rate, lower secondary, both sexes (%)	Max.	96.35947	99.76832
Adjusted net enrolment rate, lower secondary, both sexes (%)	Mean	89.89283	92.87745
Adjusted net enrolment rate, lower secondary, both sexes (%)	Median	92.68289	95.15940
Adjusted net enrolment rate, lower secondary, both sexes (%)	Min	81.54556	79.18313
Adjusted net enrolment rate, lower secondary, female (%)	1st Qu.	86.54737	91.60085
Adjusted net enrolment rate, lower secondary, female (%)	3rd Qu.	94.96147	97.46266
Adjusted net enrolment rate, lower secondary, female (%)	Max.	97.77446	100.00000
Adjusted net enrolment rate, lower secondary, female (%)	Mean	91.71886	93.10216
Adjusted net enrolment rate, lower secondary, female (%)	Median	93.63483	95.36801
Adjusted net enrolment rate, lower secondary, female (%)	Min	81.40902	78.92239

```
# The number of data points in original data set
nrow(eco_data)
```

```
## [1] 886930
```

```
# The number of data points in filtered data set
nrow(eco_data_f_u)
```

```
## [1] 4
```

In the orginal data set, threre are 886,930 data points. But in my cleaned data set, since I selected two countries and two indices, there are 4 data points.

Problem 4

Using *base* plotting functions, create a single figure that is composed of the first two rows of plots from SAS's simple linear regression diagnostics as shown here: https://support.sas.com/rnd/app/ODSGraphics/examples/reg.html. Demonstrate the plot using suitable data from problem 3.

```
# Regression Data
# For France, I used UIS.NERA.2 (y) and UIS.NERA.2.F (x) for regression data
# NA values are removed
reg_data <- na.omit(data.frame(as.numeric(eco_data_f[1,]),as.numeric(eco_data_f[2,])))
```

```
## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA

## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA

## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA
## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA

## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA

## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA

## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA

## Warning in data.frame(as.numeric(eco_data_f[1, ]), as.numeric(eco_data_f[2, :
##          NA
```

```
colnames(reg_data) <- c("UIS.NERA.2","UIS.NERA.2.F")
head(reg_data)
```

```
##    UIS.NERA.2 UIS.NERA.2.F
## 12   82.56163     85.20014
## 14   82.67026     85.43028
## 15   82.62435     85.34531
## 17   82.71007     85.35452
## 18   83.65083     86.46804
## 20   85.85859     88.24866
```

```
# Regression fit
lmfit <- lm(reg_data[,1]~reg_data[,2])
```

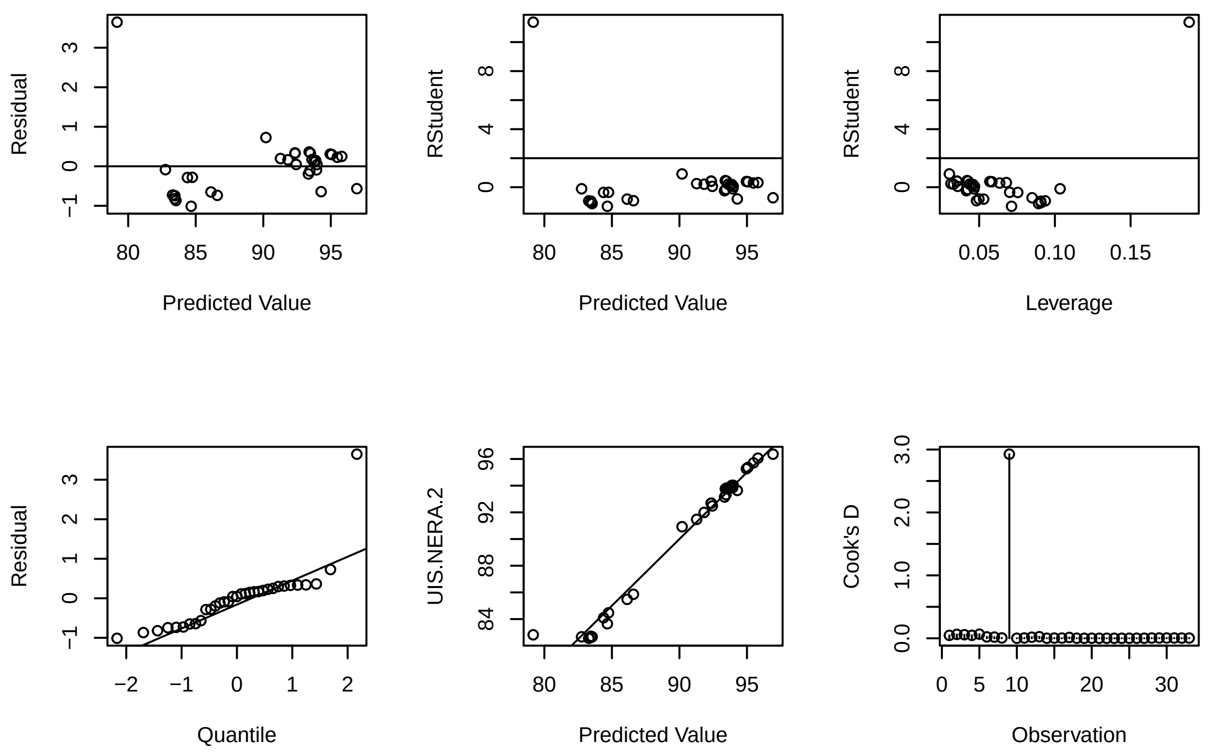
```
library(MASS)
```

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

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

```
library(sur)
```

```
## Warning: package 'sur' was built under R version 4.0.3
```

```
# Function
diagnotistics <- function(lmfit, reg_data){
  par(mfrow=c(2,3))

  plot(lmfit$fitted.values,lmfit$residuals,xlab="Predicted Value", ylab="Residual")
  abline(h=0)

  plot(lmfit$fitted.values,studres(lmfit),xlab="Predicted Value", ylab="RStudent")
  abline(h=2)
  abline(h=-2)

  plot(leverage(lmfit),studres(lmfit),xlab="Leverage", ylab="RStudent")
  abline(h=2)
  abline(h=-2)

  qqnorm(lmfit$residuals, xlab="Quantile",ylab="Residual",main="")
  qqline(lmfit$residuals)

  plot(lmfit$fitted.values,reg_data$UIS.NERA.2,xlab="Predicted Value", ylab=colnames(reg_data)[1])
  abline(0,1)

  plot(1:nrow(reg_data),cooks.distance(lmfit),xlab="Observation",ylab="Cook's D",type=c("h"))
  points(1:nrow(reg_data),cooks.distance(lmfit))
}

# Plot
diagnotistics(lmfit, reg_data)
```



Problem 5

Recreate the plot in problem 3 using ggplot2 functions. Note: there are many extension libraries for ggplot, you will probably find an extension to the ggplot2 functionality will do exactly what you want.

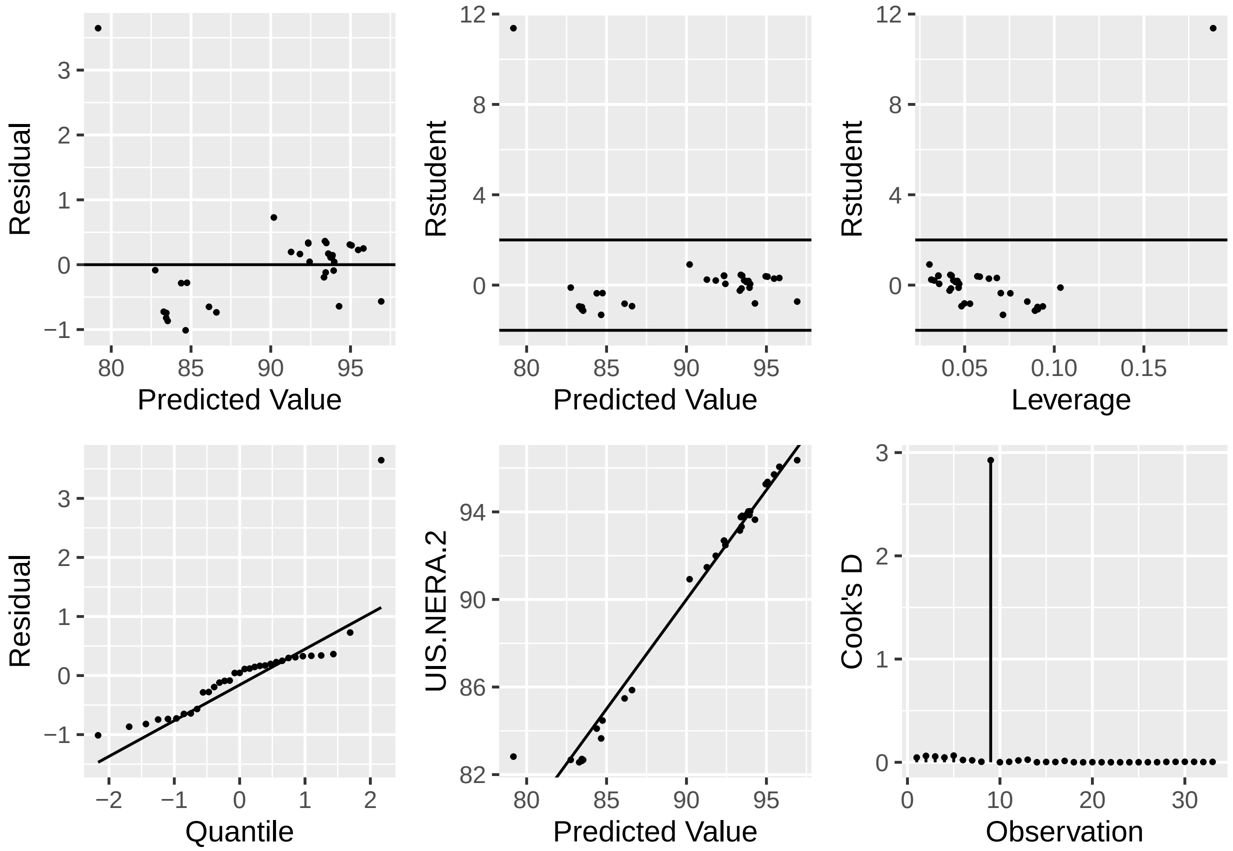
```
require(gridExtra)
```

```
## Loading required package: gridExtra
```

```
##
## Attaching package: 'gridExtra'
```

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

```
fit_res <- data.frame(lmfit$fitted.values,lmfit$residuals)
names(fit_res) <- c("fit","res")
fit_stu <- data.frame(lmfit$fitted.values,studres(lmfit))
names(fit_stu) <- c("fit","stu")
lev_stu <- data.frame(leverage(lmfit),studres(lmfit))
names(lev_stu) <- c("lev","stu")
res <- data.frame(y = lmfit$residuals)
fit_y <- data.frame(lmfit$fitted.values,reg_data$UIS.NERA.2)
names(fit_y) <- c("fit","y")
obs_cook <- data.frame(1:nrow(reg_data),cooks.distance(lmfit))
names(obs_cook) <- c("obs","cook")

plot1 <- ggplot(fit_res, aes(fit, res)) + geom_point(size = 0.5) + labs(y = "Residual", x="Predicted Val
plot2 <- ggplot(fit_stu, aes(fit, stu)) + geom_point(size = 0.5) + labs(y = "Rstudent", x="Predicted Val
plot3 <- ggplot(lev_stu, aes(lev, stu)) + geom_point(size = 0.5) + labs(y = "Rstudent", x="Leverage") +
plot4 <- ggplot(res, aes(sample = y)) + stat_qq(size = 0.5) + stat_qq_line() + labs(y = "Residual", x =
plot5 <- ggplot(fit_y, aes(fit, y)) + geom_point(size = 0.5) + labs(y = "UIS.NERA.2", x="Predicted Valu
plot6 <- ggplot(obs_cook, aes(x=obs, y=cook)) + geom_point(size = 0.5) + labs(y = "Cook's D", x="Observa

grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, ncol=3, nrow=2)
```



Problem 6

Finish this homework by pushing your changes to your repo.

Only submit the .Rmd and .pdf solution files. Names should be formatted HW5__lastname__firstname.Rmd and HW5__lastname__firstname.pdf