

# HW5\_jaeyounglee

Jaeyoung Lee

October 27, 2020

## Problem 1

Mission Complete

## Problem 2

Mission Complete

## 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](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.

```
### Choose two countries by random sampling ###
set.seed(1000292)
country <-
  fread(input = "./Edstats/EdStatsCountry.csv", header = TRUE) %>%
  select('Country Code')
sample_country <- sample(country$'Country Code', size = 2)
sample_country # South Korea and Greece
```

```
## [1] "KOR" "GRC"
```

```
### Load data ###
edstats <- fread(input = "./Edstats/EdStatsData.csv", header = TRUE)
dim(edstats) # Dimension of raw data
```

```
## [1] 886930    70
```

```
names(edstats) = c('country', 'code_country',
                   'indicator', 'code_indicator',
                   1970:2017, seq(2020, 2100, by = 5), 'V70') # Rename the columns
edstats <- edstats %>% select(-4, -(53:70)) # Remove future and indicator code

### Choose two countries ###
```

```
edstats_kor <- edstats %>% filter(code_country == sample_country[1]) # South Korea
edstats_grc <- edstats %>% filter(code_country == sample_country[2]) # Greece
```

```
### Gather the data and remove missing values ###
```

```
gather_kor <- edstats_kor %>%
  gather(key = "year", value = "n", 4:51) %>%
  filter(!is.na(n))
```

```
gather_grc <- edstats_grc %>%
  gather(key = "year", value = "n", 4:51) %>%
  filter(!is.na(n))
```

```
#### Spread data and remove redundant variables ####
```

```
spread_kor <- gather_kor %>% spread(key = indicator, value = n)
```

```
spread_grc <- gather_grc %>% spread(key = indicator, value = n)
```

```
# Find columns that have more than 10 missing values
```

```
# Use South Korea data to make equivalent columns for both countries
```

```
nuisance_col <- NULL
```

```
for(i in 1:ncol(spread_kor)){
  column <- spread_kor %>% select(i)
  count_missing <- sum(is.na(column))
  if(count_missing > 10){
    nuisance_col <- c(nuisance_col, i)
  }
}
```

```
# Find columns which name contain gender
```

```
name_kor <- spread_kor %>% select(-nuisance_col) %>% names
index_kor <- which(str_detect(name_kor, 'ale'))
```

```
name_grc <- spread_grc %>% select(-nuisance_col) %>% names
index_grc <- which(str_detect(name_grc, 'ale'))
```

```
# Remove redundant variables
```

```
tidy_kor <- spread_kor %>% select(-nuisance_col) %>% select(-index_kor)
tidy_grc <- spread_grc %>% select(-nuisance_col) %>% select(-index_grc)
```

```
# Dimension of tidy datasets
```

```
dim(tidy_kor) # Dimension of data for Korea
```

```
## [1] 47 98
```

```
dim(tidy_grc) # Dimension of data for Greece
```

```
## [1] 47 94
```

```
# Select indicators that we are interested in for each dataset
```

```
interest_kor <- tidy_kor %>% select(3, 27, 29)
```

```
interest_grc <- tidy_grc %>% select(3, 30, 31)
```

```
# Make year as numeric
interest_kor$year <- as.numeric(interest_kor$year)
interest_grc$year <- as.numeric(interest_grc$year)

# Summary table
knitr::kable(summary(interest_kor), caption = 'Summary of indicators, Korea')
```

Table 1: Summary of indicators, Korea

year	GDP per capita (constant 2005 US)  <i>GNI(currentUS)</i>	
Min. :1970	Min. : 1815	Min. :9.153e+09
1st Qu.:1982	1st Qu.: 4033	1st Qu.:7.418e+10
Median :1993	Median :10280	Median :3.691e+11
Mean :1993	Mean :11580	Mean :4.878e+11
3rd Qu.:2004	3rd Qu.:18237	3rd Qu.:8.272e+11
Max. :2016	Max. :25459	Max. :1.416e+12

```
knitr::kable(summary(interest_grc), caption = 'Summary of indicators, Greece')
```

Table 2: Summary of indicators, Greece

year	GDP per capita (current US\$)	GNI, PPP (current international \$)
Min. :1970	Min. : 1494	Min. :1.385e+11
1st Qu.:1982	1st Qu.: 5200	1st Qu.:1.791e+11
Median :1993	Median :11091	Median :2.601e+11
Mean :1993	Mean :12371	Mean :2.400e+11
3rd Qu.:2004	3rd Qu.:18274	3rd Qu.:2.874e+11
Max. :2016	Max. :31997	Max. :3.316e+11
NA	NA	NA's :20

Two countries are sampled from the raw data. The tables are summary of tidy data sets with some interesting indicators. The dimension of data for Korea is (47, 98) and that for Greece is (47, 94).

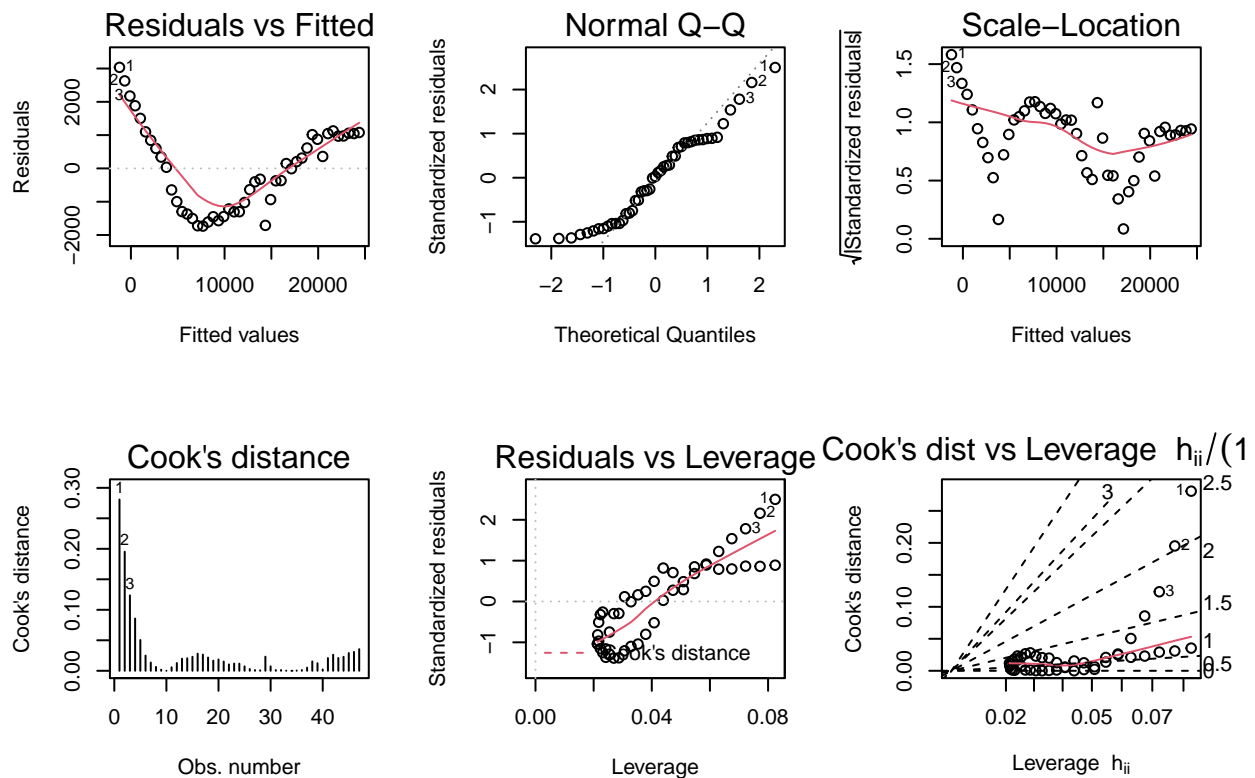
## Problem 4

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.

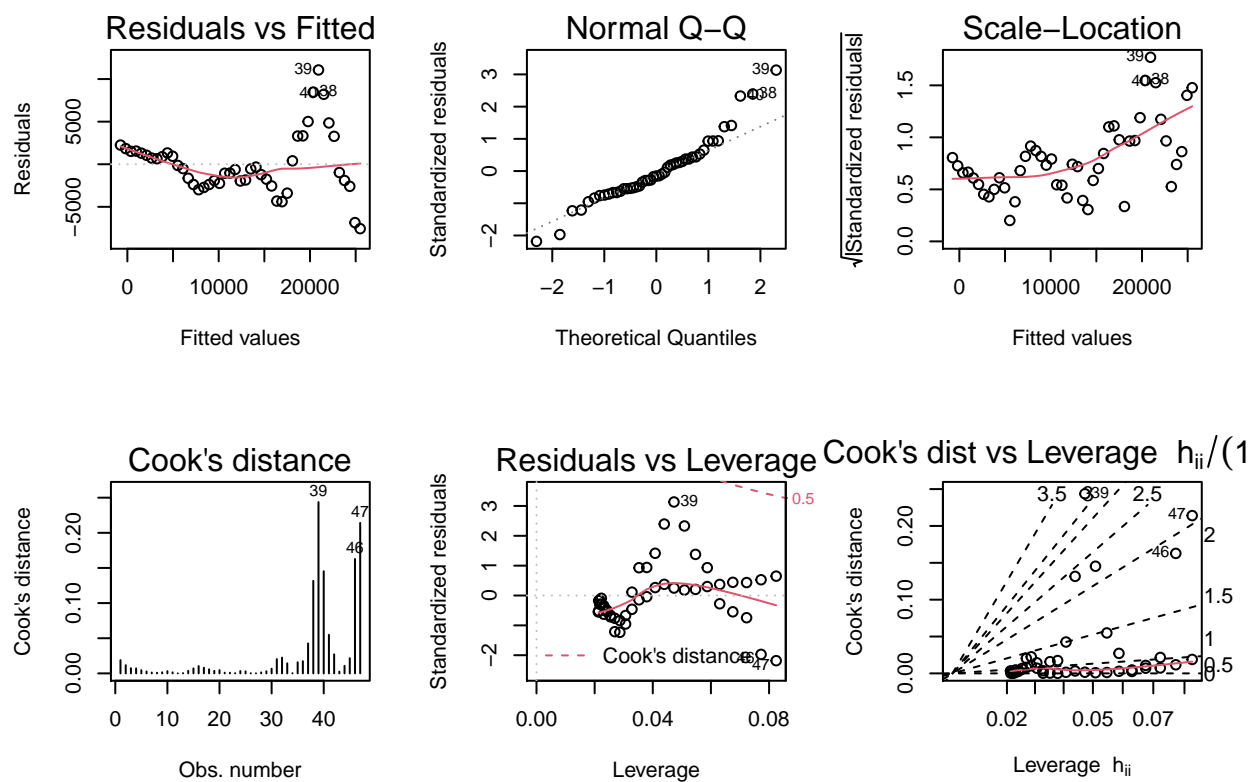
```
# Simple linear regression diagnostics using base R function
# Make names simpler
names(interest_kor) <- c('year', 'gdp', 'gni')
names(interest_grc) <- c('year', 'gdp', 'gni')

# Fit simple linear model
# Want to know the linear relationship between GDP and Year for each country
lm_kor <- lm(gdp ~ year, data = interest_kor)
lm_grc <- lm(gdp ~ year, data = interest_grc)

# Model diagnostic plots
# Korea
par(mfrow = c(2,3))
plot(lm_kor, 1:6)
```



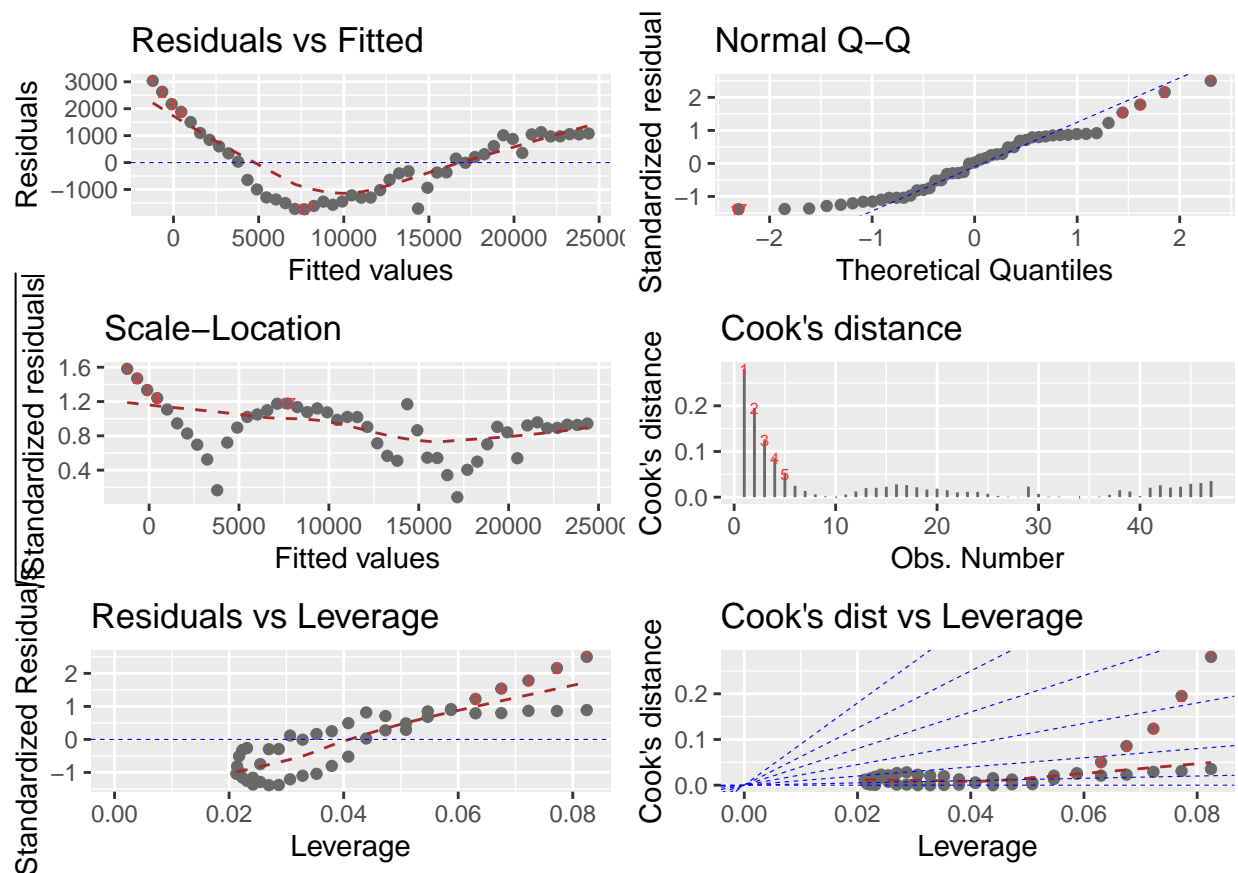
```
# Greece
par(mfrow = c(2,3))
plot(lm_grc, 1:6)
```



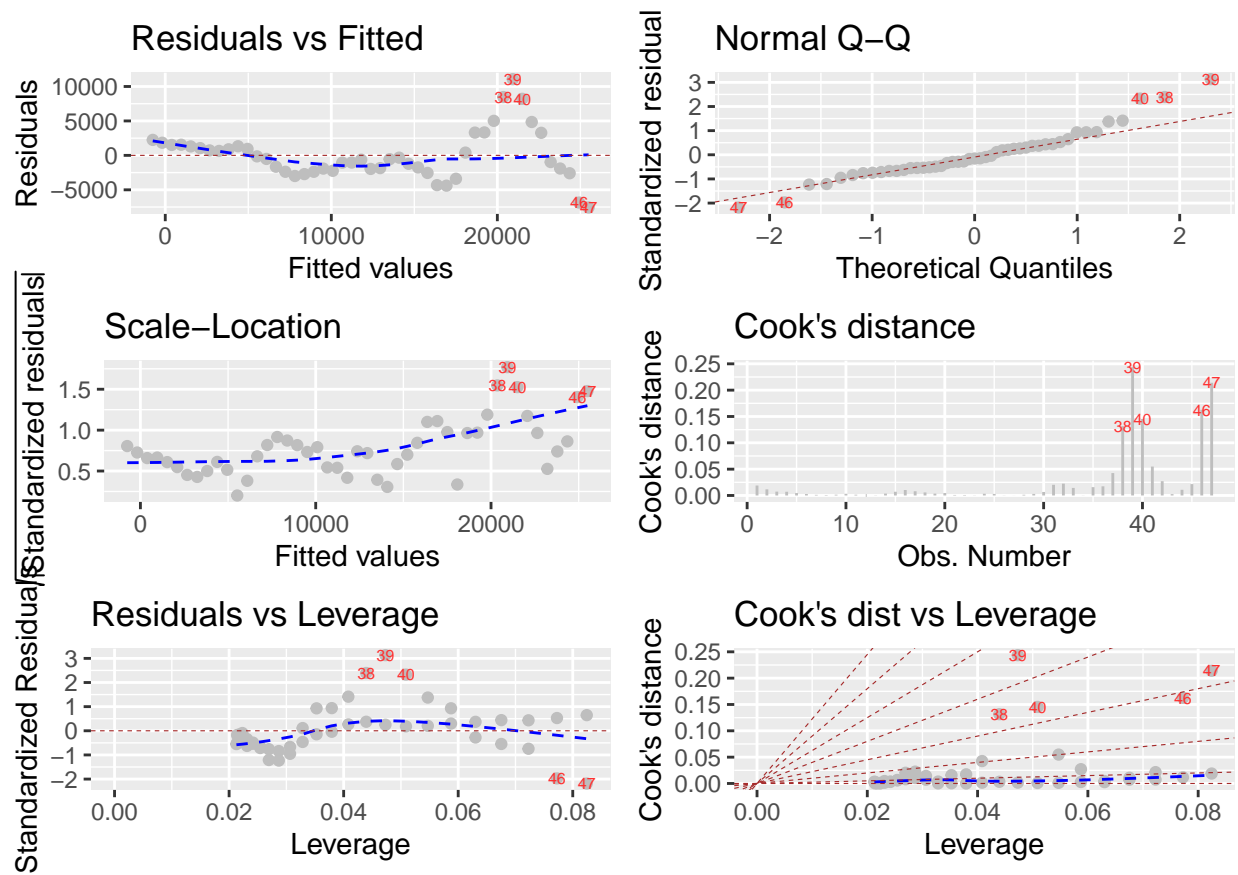
## Problem 5

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.

```
# Simple linear regression diagnostics using ggplot2
# Using autoplot
# Reference : https://cran.r-project.org/web/packages/ggfortify/vignettes/plot\_lm.html
autoplot(lm_kor, 1:6, colour = 'dimgrey',
         smooth.colour = 'brown', smooth.linetype = 'dashed',
         ad.colour = 'blue',
         label.size = 2, label.n = 5, label.colour = 'firebrick1',
         ncol = 2)
```



```
autoplot(lm_grc, 1:6, colour = 'grey',
         smooth.colour = 'blue', smooth.linetype = 'dashed',
         ad.colour = 'brown',
         label.size = 2, label.n = 5, label.colour = 'firebrick1',
         ncol = 2)
```



```
# Using ggplot
# Reference : https://rpubs.com/therimalaya/43190
# Korea

# Residual plot
plot_resid_kor <- ggplot(lm_kor, aes(.fitted, .resid)) +
  geom_point() +
  stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values")+ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot") +
  theme_bw()

# Normal Q-Q plot
qq_kor <- ggplot(lm_kor, aes(sample = lm_kor$residuals)) +
  stat_qq(size = 2) +
  stat_qq_line(color = 'red')+
  xlab("Theoretical Quantiles")+ylab("Standardized Residuals") +
  ggtitle("Normal Q-Q") +
  theme_bw()

# Standardized residual plot
plot_standard_resid_kor <- ggplot(lm_kor, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm = TRUE) +
  xlab("Fitted Value") +
```

```

ylab(expression(sqrt("|Standardized residuals|"))) +
ggtitle("Scale-Location") +
theme_bw()

# Cook's distance
cook_kor <- ggplot(lm_kor, aes(seq_along(.cooks), .cooks)) +
  geom_bar(stat="identity", position="identity") +
  xlab("Obs. Number") +
  ylab("Cook's distance") +
  ggtitle("Cook's distance")+theme_bw()

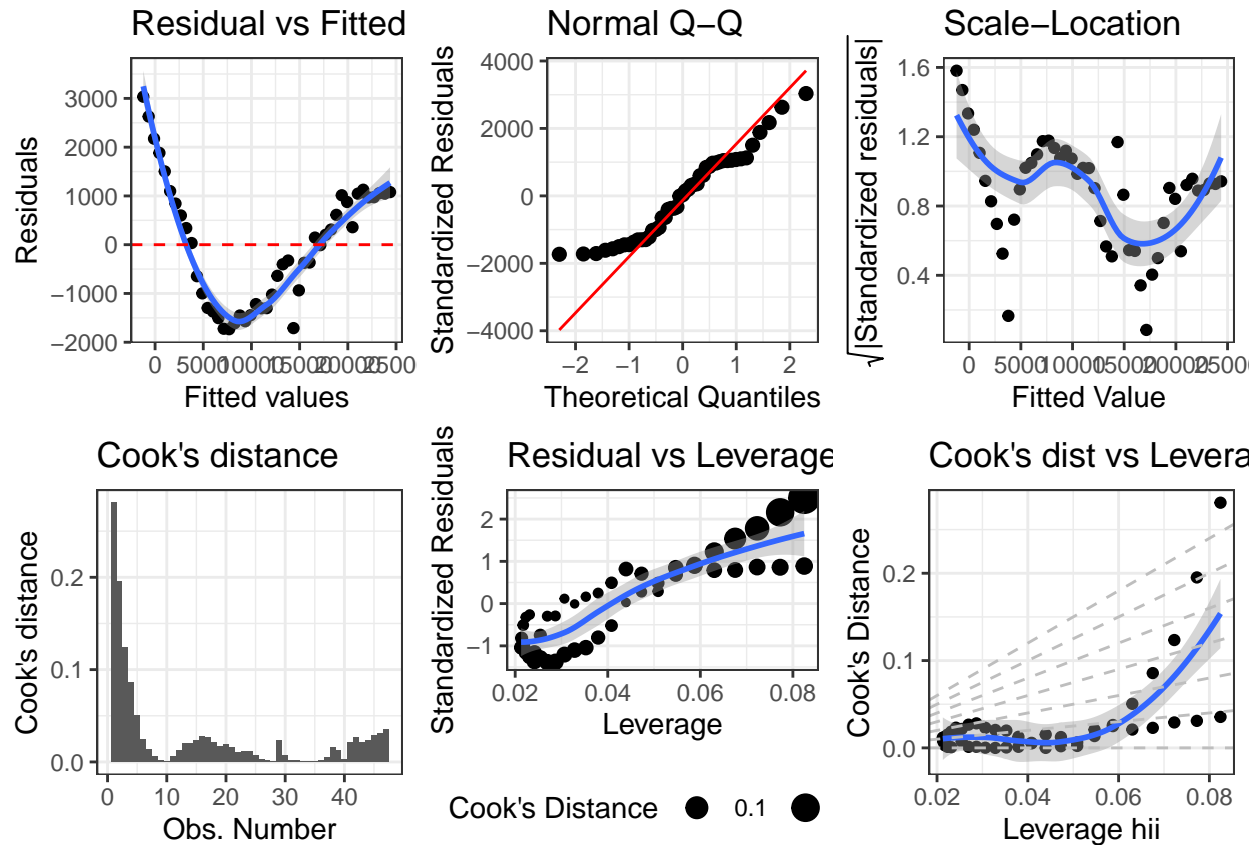
# Residual vs Leverage
leverage_kor <- ggplot(lm_kor, aes(.hat, .stdresid)) +
  geom_point(aes(size=.cooks), na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage") +
  ylab("Standardized Residuals") +
  ggtitle("Residual vs Leverage Plot") +
  scale_size_continuous("Cook's Distance", range=c(1,5)) +
  theme_bw() +
  theme(legend.position="bottom")

# Cook's distance vs Leverage
lev_cook_kor <- ggplot(lm_kor, aes(.hat, .cooks)) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage hii")+ylab("Cook's Distance") +
  ggtitle("Cook's dist vs Leverage hii/(1-hii)") +
  geom_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed") +
  theme_bw()

# Diagnostic plot
ggarrange(plot_resid_kor, qq_kor,
  plot_standard_resid_kor, cook_kor,
  leverage_kor, lev_cook_kor, nrow = 2, ncol = 3)

```





```
# Using ggplot
# Greece

# Residual plot
plot_resid_grc <- ggplot(lm_grc, aes(.fitted, .resid)) +
  geom_point() +
  stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values")+ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot") +
  theme_bw()

# Normal Q-Q plot
qq_grc <- ggplot(lm_grc, aes(sample = lm_grc$residuals)) +
  stat_qq(size = 2) +
  stat_qq_line(color = 'red')+
  xlab("Theoretical Quantiles")+ylab("Standardized Residuals") +
  ggtitle("Normal Q-Q") +
  theme_bw()

# Standardized residual plot
plot_standard_resid_grc <- ggplot(lm_grc, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm = TRUE) +
  xlab("Fitted Value") +
  ylab(expression(sqrt("|Standardized residuals|")))
```

```

ggtitle("Scale-Location") +
theme_bw()

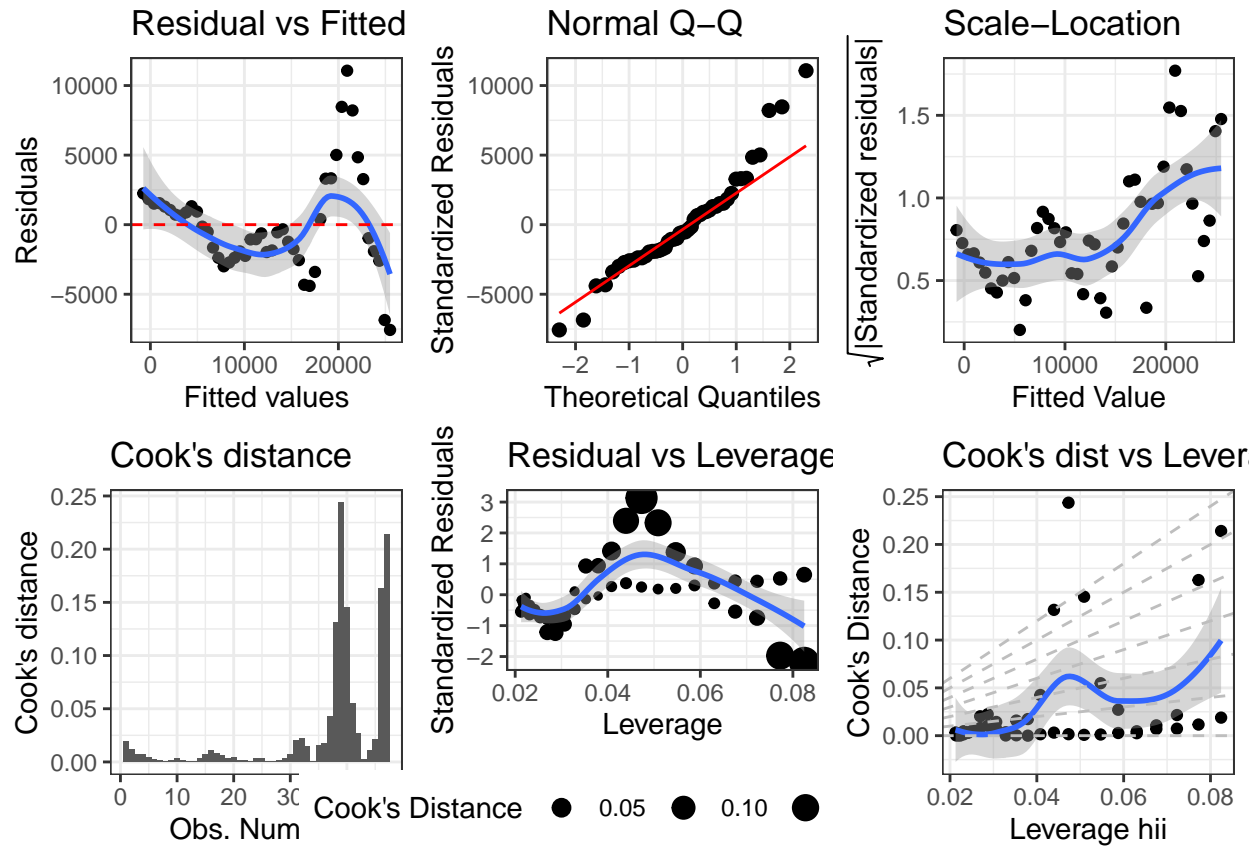
# Cook's distance
cook_grc <- ggplot(lm_grc, aes(seq_along(.cooks), .cooks)) +
  geom_bar(stat="identity", position="identity") +
  xlab("Obs. Number") +
  ylab("Cook's distance") +
  ggtitle("Cook's distance")+theme_bw()

# Residual vs Leverage plot
leverage_grc <- ggplot(lm_grc, aes(.hat, .stdresid)) +
  geom_point(aes(size=.cooks), na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage") +
  ylab("Standardized Residuals") +
  ggtitle("Residual vs Leverage Plot") +
  scale_size_continuous("Cook's Distance", range=c(1,5)) +
  theme_bw() +
  theme(legend.position="bottom")

# Cook's distance vs Leverage plot
lev_cook_grc <- ggplot(lm_grc, aes(.hat, .cooks)) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage hii")+ylab("Cook's Distance") +
  ggtitle("Cook's dist vs Leverage hii/(1-hii)") +
  geom_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed") +
  theme_bw()

# Diagnostic plot
ggarrange(plot_resid_grc, qq_grc,
  plot_standard_resid_grc, cook_grc,
  leverage_grc, lev_cook_grc, nrow = 2, ncol = 3)

```



## Problem 6

Mission Complete

## Appendix

```
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)
library(data.table)
library(tidyverse)
library(ggplot2)
library(ggfortify)
library(ggpubr)
### Choose two countries by random sampling ###
set.seed(1000292)
country <-
  fread(input = "./Edstats/EdStatsCountry.csv", header = TRUE) %>%
  select('Country Code')
sample_country <- sample(country$'Country Code', size = 2)
sample_country # South Korea and Greece

### Load data ###
edstats <- fread(input = "./Edstats/EdStatsData.csv", header = TRUE)
dim(edstats) # Dimension of raw data
names(edstats) = c('country', 'code_country',
                   'indicator', 'code_indicator',
                   1970:2017, seq(2020, 2100, by = 5), 'V70') # Rename the columns
edstats <- edstats %>% select(-4, -(53:70)) # Remove future and indicator code

### Choose two countries ###
edstats_kor <- edstats %>% filter(code_country == sample_country[1]) # South Korea
edstats_grc <- edstats %>% filter(code_country == sample_country[2]) # Greece

### Gather the data and remove missing values ###
gather_kor <- edstats_kor %>%
  gather(key = "year", value = "n", 4:51) %>%
  filter(!is.na(n))

gather_grc <- edstats_grc %>%
  gather(key = "year", value = "n", 4:51) %>%
  filter(!is.na(n))

#### Spread data and remove redundant variables ###
spread_kor <- gather_kor %>% spread(key = indicator, value = n)

spread_grc <- gather_grc %>% spread(key = indicator, value = n)

# Find columns that have more than 10 missing values
# Use South Korea data to make equivalent columns for both countries
nuisance_col <- NULL
for(i in 1:ncol(spread_kor)){
  column <- spread_kor %>% select(i)
  count_missing <- sum(is.na(column))
  if(count_missing > 10){
    nuisance_col <- c(nuisance_col, i)
  }
}
```

```

# Find columns which name contain gender
name_kor <- spread_kor %>% select(-nuisance_col) %>% names
index_kor <- which(str_detect(name_kor, 'ale'))

name_grc <- spread_grc %>% select(-nuisance_col) %>% names
index_grc <- which(str_detect(name_grc, 'ale'))

# Remove redundant variables
tidy_kor <- spread_kor %>% select(-nuisance_col) %>% select(-index_kor)
tidy_grc <- spread_grc %>% select(-nuisance_col) %>% select(-index_grc)

# Dimension of tidy datasets
dim(tidy_kor) # Dimension of data for Korea
dim(tidy_grc) # Dimension of data for Greece

# Select indicators that we are interested in for each dataset
interest_kor <- tidy_kor %>% select(3, 27, 29)
interest_grc <- tidy_grc %>% select(3, 30, 31)

# Make year as numeric
interest_kor$year <- as.numeric(interest_kor$year)
interest_grc$year <- as.numeric(interest_grc$year)

# Summary table
knitr::kable(summary(interest_kor), caption = 'Summary of indicators, Korea')
knitr::kable(summary(interest_grc), caption = 'Summary of indicators, Greece')

# Simple linear regression diagnostics using base R function
# Make names simpler
names(interest_kor) <- c('year', 'gdp', 'gni')
names(interest_grc) <- c('year', 'gdp', 'gni')

# Fit simple linear model
# Want to know the linear relationship between GDP and Year for each country
lm_kor <- lm(gdp ~ year, data = interest_kor)
lm_grc <- lm(gdp ~ year, data = interest_grc)

# Model diagnostic plots
# Korea
par(mfrow = c(2,3))
plot(lm_kor, 1:6)

# Greece
par(mfrow = c(2,3))
plot(lm_grc, 1:6)

# Simple linear regression diagnostics using ggplot2
# Using autoplot
# Reference : https://cran.r-project.org/web/packages/ggfortify/vignettes/plot\_lm.html
autoplot(lm_kor, 1:6, colour = 'dimgrey',
         smooth.colour = 'brown', smooth.linetype = 'dashed',

```

```

    ad.colour = 'blue',
    label.size = 2, label.n = 5, label.colour = 'firebrick1',
    ncol = 2)
autoplot(lm_grc, 1:6, colour = 'grey',
    smooth.colour = 'blue', smooth.linetype = 'dashed',
    ad.colour = 'brown',
    label.size = 2, label.n = 5, label.colour = 'firebrick1',
    ncol = 2)

# Using ggplot
# Reference : https://rpubs.com/therimalaya/43190
# Korea

# Residual plot
plot_resid_kor <- ggplot(lm_kor, aes(.fitted, .resid)) +
  geom_point() +
  stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values")+ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot") +
  theme_bw()

# Normal Q-Q plot
qq_kor <- ggplot(lm_kor, aes(sample = lm_kor$residuals)) +
  stat_qq(size = 2) +
  stat_qq_line(color = 'red')+
  xlab("Theoretical Quantiles")+ylab("Standardized Residuals") +
  ggtitle("Normal Q-Q") +
  theme_bw()

# Standardized residual plot
plot_standard_resid_kor <- ggplot(lm_kor, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm = TRUE) +
  xlab("Fitted Value") +
  ylab(expression(sqrt("|Standardized residuals|"))) +
  ggtitle("Scale-Location") +
  theme_bw()

# Cook's distance
cook_kor <- ggplot(lm_kor, aes(seq_along(.cooks), .cooks)) +
  geom_bar(stat="identity", position="identity") +
  xlab("Obs. Number") +
  ylab("Cook's distance") +
  ggtitle("Cook's distance")+theme_bw()

# Residual vs Leverage
leverage_kor <- ggplot(lm_kor, aes(.hat, .stdresid)) +
  geom_point(aes(size=.cooks), na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage") +
  ylab("Standardized Residuals") +

```

```

ggtitle("Residual vs Leverage Plot") +
scale_size_continuous("Cook's Distance", range=c(1,5)) +
theme_bw() +
theme(legend.position="bottom")

# Cook's distance vs Leverage
lev_cook_kor <- ggplot(lm_kor, aes(.hat, .cooksdist)) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage hii")+ylab("Cook's Distance") +
  ggtitle("Cook's dist vs Leverage hii/(1-hii)") +
  geom_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed") +
  theme_bw()

# Diagnostic plot
ggarrange(plot_resid_kor, qq_kor,
  plot_standard_resid_kor, cook_kor,
  leverage_kor, lev_cook_kor, nrow = 2, ncol = 3)

# Using ggplot
# Greece

# Residual plot
plot_resid_grc <- ggplot(lm_grc, aes(.fitted, .resid)) +
  geom_point() +
  stat_smooth(method="loess") +
  geom_hline(yintercept=0, col="red", linetype="dashed") +
  xlab("Fitted values")+ylab("Residuals") +
  ggtitle("Residual vs Fitted Plot") +
  theme_bw()

# Normal Q-Q plot
qq_grc <- ggplot(lm_grc, aes(sample = lm_grc$residuals)) +
  stat_qq(size = 2) +
  stat_qq_line(color = 'red')+
  xlab("Theoretical Quantiles")+ylab("Standardized Residuals") +
  ggtitle("Normal Q-Q") +
  theme_bw()

# Standardized residual plot
plot_standard_resid_grc <- ggplot(lm_grc, aes(.fitted, sqrt(abs(.stdresid)))) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm = TRUE) +
  xlab("Fitted Value") +
  ylab(expression(sqrt("|Standardized residuals|"))) +
  ggtitle("Scale-Location") +
  theme_bw()

# Cook's distance
cook_grc <- ggplot(lm_grc, aes(seq_along(.cooksdist), .cooksdist)) +
  geom_bar(stat="identity", position="identity") +
  xlab("Obs. Number") +
  ylab("Cook's distance") +

```

```

ggtitle("Cook's distance")+theme_bw()

# Residual vs Leverage plot
leverage_grc <- ggplot(lm_grc, aes(.hat, .stdresid)) +
  geom_point(aes(size=.cooks_d), na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage") +
  ylab("Standardized Residuals") +
  ggtitle("Residual vs Leverage Plot") +
  scale_size_continuous("Cook's Distance", range=c(1,5)) +
  theme_bw() +
  theme(legend.position="bottom")

# Cook's distance vs Leverage plot
lev_cook_grc <- ggplot(lm_grc, aes(.hat, .cooks_d)) +
  geom_point(na.rm=TRUE) +
  stat_smooth(method="loess", na.rm=TRUE) +
  xlab("Leverage hii")+ylab("Cook's Distance") +
  ggtitle("Cook's dist vs Leverage hii/(1-hii)") +
  geom_abline(slope=seq(0,3,0.5), color="gray", linetype="dashed") +
  theme_bw()

# Diagnostic plot
ggarrange(plot_resid_grc, qq_grc,
           plot_standard_resid_grc, cook_grc,
           leverage_grc, lev_cook_grc, nrow = 2, ncol = 3)

```