Homework 5

Steven Barnett

10/28/2020

Problem 3

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

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.

In the original data set, there are 886,930 observations with 70 variables for each. That is over 64 million data points (62,085,100).

After cleaning and tidying the data, we have just over 20 million data points (20,328,804 data points).

```
# Load data from remote location. Save locally
## world_bank_url <- "http://databank.worldbank.org/data/download/Edstats_csv.zip"
## world_bank_data <- fread(world_bank_url)
## world_bank_data <- fread("./dwnldd_data/Edstats_csv/EdStatsData.csv", header = TRUE)
## saveRDS(world_bank_data, "dwnldd_data/world_bank_data_raw.RDS")
world_bank_data <- readRDS("dwnldd_data/world_bank_data_raw.RDS")

# Create maps of redundant information in dataset
country_code_map <- world_bank_data %>%
  select(c("Country Code", "Country Name")) %>%
  distinct()
indicator_map <- world_bank_data %>%
  select(c("Indicator Code", "Indicator Name")) %>%
  distinct()

# Tidy data: remove columns with all NA, gather years from headers to cell
# values, remove blank observations
world_bank_tidy <- world_bank_data %>%
  select_if(~sum(!is.na(.)) > 0) %>%
  gather(key = "Year", value = "Value", 5:69) %>%
  drop_na() %>%
  select(!c("Country Name"))

literacy_rate_codes <- c("UIS.LPP.AG15T99", "UIS.LP.AG15T99",
                         "UIS.LP.AG15T99.F", "UIS.LP.AG15T99.M",
                         "SE.ADT.LITR.ZS", "SE.ADT.LITR.FE.ZS",
                         "UIS.LR.AG15T99.GPI", "SE.ADT.LITR.MA.ZS")

# Select the literacy rates for Mexico and Colombia in the Year 2015
country_compare_data <- world_bank_tidy %>%
  filter(`Country Code` %in% c("MEX", "COL")) %>%
  filter(`Indicator Code` %in% literacy_rate_codes) %>%
```

```r
  filter(`Year` == 2015) %>%
  select(!Year) %>%
  select(!`Indicator Code`) %>%
  spread(`Country Code`, `Value`)

knitr::kable(country_compare_data)
```

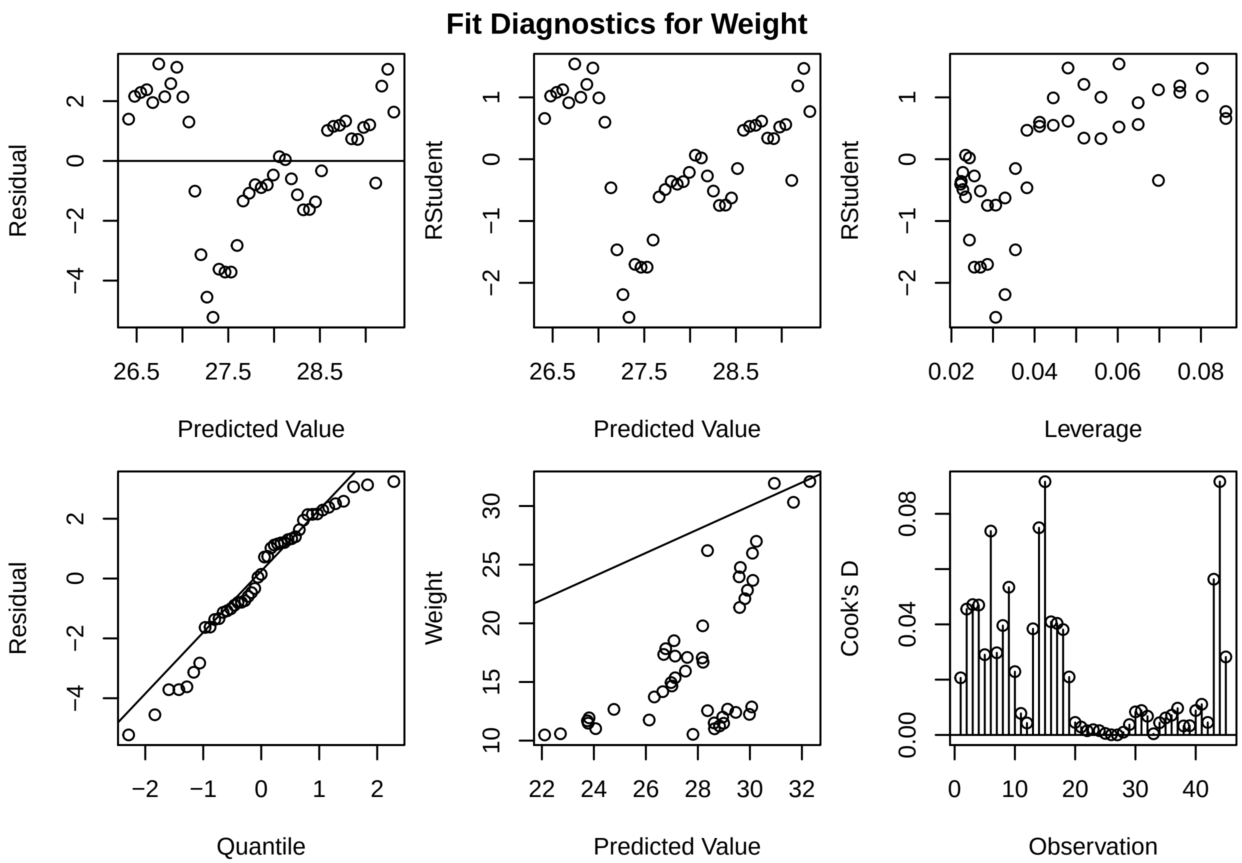
Indicator Name	COL	MEX
Adult illiterate population, 15+ years, % female	4.981971e+01	6.013334e+01
Adult illiterate population, 15+ years, both sexes (number)	2.101738e+06	5.055690e+06
Adult illiterate population, 15+ years, female (number)	1.047080e+06	3.040155e+06
Adult illiterate population, 15+ years, male (number)	1.054658e+06	2.015535e+06
Adult literacy rate, population 15+ years, both sexes (%)	9.424505e+01	9.447228e+01
Adult literacy rate, population 15+ years, female (%)	9.441582e+01	9.348550e+01
Adult literacy rate, population 15+ years, gender parity index (GPI)	1.003750e+00	9.784000e-01
Adult literacy rate, population 15+ years, male (%)	9.406317e+01	9.554933e+01

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.

```r
# Load data from Bulgaria for Vocational Secondary Enrollment
bulgaria_data <- world_bank_tidy %>%
  filter(`Country Code` == "BGR") %>%
  filter(`Indicator Code` == "SE.SEC.ENRL.VO.ZS") %>%
  select(c("Year", "Value"))

colnames(bulgaria_data) <- c("Year", "VocationalEnrollment")
bulgaria_data$Year <- as.integer(bulgaria_data$Year)

bulgaria_lm <- lm(VocationalEnrollment ~ Year, bulgaria_data)

resids <- residuals(bulgaria_lm)
fitted_values <- fitted(bulgaria_lm)

par(mfrow = c(2, 3))
par(mar = c(4.5, 4, 0.5, 0.5), oma = c(0, 0, 1, 0), cex = .75)

plot(x = fitted_values, y = resids, ylab = "Residual",
     xlab = "Predicted Value")
abline(h = 0)

student_resids <- rstudent(bulgaria_lm)
plot(x = fitted_values, y = student_resids, ylab = "RStudent",
     xlab = "Predicted Value")

xi <- as.matrix(bulgaria_data$Year)
yi <- as.matrix(bulgaria_data$VocationalEnrollment)
n <- length(xi)
X <- matrix(c(rep(1, n), xi), nrow = n, ncol = 2)
Y <- as.matrix(yi)
```

```
H <- X %*% solve(t(X) %*% X) %*% t(X)
leverage <- diag(H)
plot(x = leverage, y = student_resids, ylab = "RStudent",
     xlab = "Leverage")

qqnorm(resids, ylab = "Residual", xlab = "Quantile", main = "")
qqline(resids)

abs_residual <- abs(resids)
bulgaria_lm_no_weights <- lm(abs_residual ~ bulgaria_data$Year)
weights <- 1 / ((fitted(bulgaria_lm_no_weights))^2)

bulgaria_lm_with_weights <- lm(VocationalEnrollment ~ Year,
                               bulgaria_data, weights = weights)
plot(yi, sqrt(weights) * yi, ylab = "Weight", xlab = "Predicted Value")
abline(c(0, 1))

cooks <- cooks.distance(bulgaria_lm)
plot(x = 1:45, y = cooks, ylab = "Cook's D", xlab = "Observation")
abline(h = 0)
for (index in 1:45) {
  segments(x0 = index, y0 = 0, x1 = index, y1 = cooks[index])
}

mtext("Fit Diagnostics for Weight", outer = TRUE, cex = 0.9, font = 2)
```

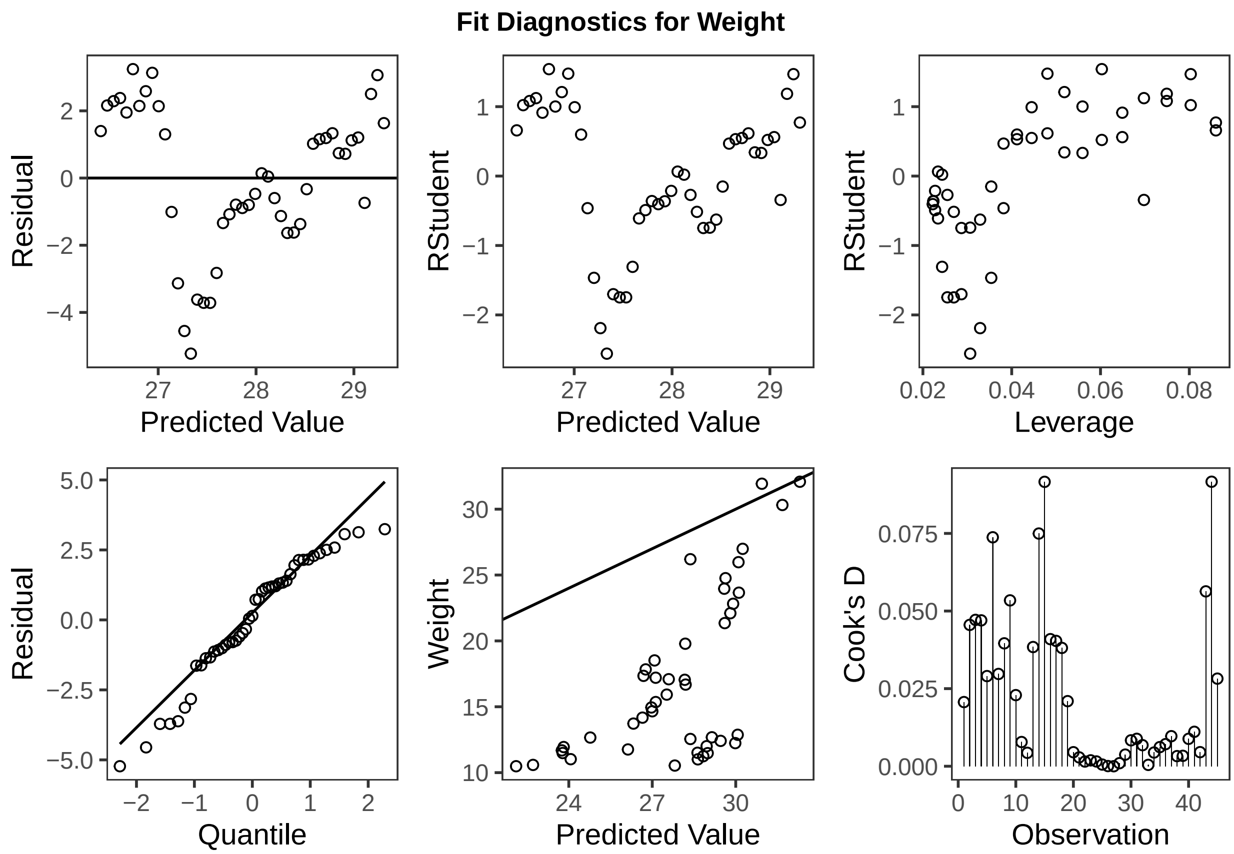


Problem 5

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.

```r
bulgaria_data <- world_bank_tidy %>%
  filter(`Country Code` == "BGR") %>%
  filter(`Indicator Code` == "SE.SEC.ENRL.VO.ZS") %>%
  select(c("Year", "Value"))

colnames(bulgaria_data) <- c("Year", "VocationalEnrollment")
bulgaria_data$Year <- as.integer(bulgaria_data$Year)

bulgaria_lm <- lm(VocationalEnrollment ~ Year, bulgaria_data)

resids <- residuals(bulgaria_lm)
fitted_values <- fitted(bulgaria_lm)

# par(mfrow = c(2, 3))
# par(mar = c(4.5, 4, 0.5, 0.5), oma = c(0, 0, 1, 0), cex = .75)

bulgaria_data <- cbind(bulgaria_data, resids)
bulgaria_data <- cbind(bulgaria_data, fitted_values)

plot1 <- ggplot(bulgaria_data, aes(x = fitted_values, y = resids)) +
  geom_point(shape = 1) +
  geom_hline(yintercept = 0) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("Predicted Value") +
  ylab("Residual")

bulgaria_data <- cbind(bulgaria_data, student_resids)
plot2 <- ggplot(bulgaria_data, aes(x = fitted_values, y = student_resids)) +
  geom_point(shape = 1) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("Predicted Value") +
  ylab("RStudent")

xi <- as.matrix(bulgaria_data$Year)
yi <- as.matrix(bulgaria_data$VocationalEnrollment)
n <- length(xi)
X <- matrix(c(rep(1, n), xi), nrow = n, ncol = 2)
Y <- as.matrix(yi)
H <- X %*% solve(t(X) %*% X) %*% t(X)
leverage <- diag(H)

bulgaria_data <- cbind(bulgaria_data, leverage)
plot3 <- ggplot(bulgaria_data, aes(x = leverage, y = student_resids)) +
  geom_point(shape = 1) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("Leverage") +
  ylab("RStudent")
```

```r
plot4 <- ggplot(bulgaria_data, aes(sample = resids)) +
  stat_qq(shape = 1) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  geom_qq_line() +
  xlab("Quantile") +
  ylab("Residual")

abs_residual <- abs(resids)
bulgaria_lm_no_weights <- lm(abs_residual ~ bulgaria_data$Year)
weights <- 1 / ((fitted(bulgaria_lm_no_weights))^2)

bulgaria_lm_with_weights <- lm(VocationalEnrollment ~ Year,
                               bulgaria_data, weights = weights)
yi_sqrt_weights <- sqrt(weights) * yi
bulgaria_data <- cbind(bulgaria_data, yi_sqrt_weights)

plot5 <- ggplot(bulgaria_data, aes(x = VocationalEnrollment, y = yi_sqrt_weights)) +
  geom_point(shape = 1) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("Predicted Value") +
  ylab("Weight") +
  geom_abline(slope = 1, intercept = 0)

cooks <- cooks.distance(bulgaria_lm)
bulgaria_data <- cbind(bulgaria_data, cooks)

plot6 <- ggplot(bulgaria_data, aes(x = 1:45, y = cooks)) +
  geom_point(shape = 1) +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank()) +
  xlab("Observation") +
  ylab("Cook's D")

for (index in 1:45) {
  plot6 <- plot6 + geom_segment(x = index, y = 0, xend = index,
                                yend = cooks[index], size = 0.1)
}

grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, nrow = 2,
             top = textGrob("Fit Diagnostics for Weight",
                            gp = gpar(fontsize = 10, font = 2)))
```


Fit Diagnostics for Weight
Residual
Predicted Value
RStudent
Predicted Value
RStudent
Leverage
Residual
Quantile
Weight
Predicted Value
Cook's D
Observation