HW-4

Dennis Goldenberg

2024-02-15

Homework 4 - Predictive Modeling in Finance and Insurance

```
library(ggplot2)
library(patchwork)
library(readxl)
```

1. ANOVA, one factor

```
#read in data, remove empty column
phosData <- read_excel("Table 6.19-PlasmaPhosphate-1.xls", skip = 2,
                       sheet = "Sheet1", .name_repair = "unique_quiet")
phosData <- phosData[1:12, 1:3]
colnames(phosData) <- c("H_O","N_O", "C")
```

a. Test of difference of means

Our hypotheses are as follows:

$$H_0 : \mu_{H_O} = \mu_{N_O} = \mu_C \text{ and } H_a : \text{ at least one different}$$

To test, I first calculate the means $\bar{x}_{H_O}, \bar{x}_{N_O}, \bar{x}_C$. There are 3 levels, so 2 degrees of freedom between sum of squares, and $n-3$ degrees of freedom in the error sum of squares. I calculate the mean sum of squares for both and generate the F-statistic:

```
means <- colMeans(phosData, na.rm = TRUE)
oMean <- sum(phosData, na.rm = TRUE)/sum(!is.na(phosData))
MSE <- 0
MSB <- 0
for (i in colnames(phosData)) {
  MSE <- MSE + sum((phosData[,i] - means[i])^2, na.rm = TRUE)
  MSB <- MSB + sum(!is.na(phosData[,i]))*((means[i] - oMean)^2)
}
MSE <- MSE/(sum(!is.na(phosData)) - (dim(phosData)[2]))
MSB <- unname(MSB/(dim(phosData)[2] - 1))
sprintf("Test statistic: %f", MSB/MSE)
```

```
## [1] "Test statistic: 11.650806"
```

I then calculate the p-value of this statistic, using $ndf = 2$ and $ddf = n - 3$ for the test statistic:

```
p_1a <- pf(MSB/MSE, df1 = dim(phosData)[2] - 1,
   df2 = sum(!is.na(phosData)) - (dim(phosData)[2]), lower.tail = F)
sprintf("p value: %f", p_1a)
```

```
## [1] "p value: 0.000208"
```

Note that $p_{1a} < 0.05$, meaning we have **enough evidence to reject** H_0; there is evidence that the means for different treatments provide different results.

1b. 95% confidence interval, difference of means

Normality is assumed; therefore, given that the estimate of $\mu_{H-O} - \mu_{N-O} = \bar{x}_{H-O} - \bar{x_{N-O}}$, the bounds for the confidence interval are:

$$\bar{x}_{H_O} - \bar{x}_{N_O} \pm z_{.975}\hat{\text{SE}}(\bar{x}_{H_O} - \bar{x_{N_O}}) = \bar{x}_{H_O} - \bar{x}_{N_O} \pm 1.96 * S_P\sqrt{\frac{1}{n_{H_O}} + \frac{1}{n_{N_O}}}$$

Here, the pooled sample variance is $S_P = \sqrt{\frac{(n_{H_O}-1)s^2_{H_O}+(n_{N_O}-1)s^2_{N_O}}{n_{H_O}+n_{N_O}-2}}$. I thus calculate the lower and upper bound:

```
n_HO <- sum(!is.na(phosData[,"H_O"]))
n_NO <- sum(!is.na(phosData[,"N_O"]))
S_P <- sqrt(((n_HO - 1)*var(phosData$H_O, na.rm = TRUE)
             + (n_NO - 1)*var(phosData$N_O, na.rm = TRUE))/
              (n_HO + n_NO - 2))
LCI <- unname(means["H_O"] - means["N_O"])[1] - 1.96*S_P*sqrt(1/n_HO + 1/n_NO)
UCI <- unname(means["H_O"] - means["N_O"])[1] + 1.96*S_P*sqrt(1/n_HO + 1/n_NO)
diffMeansconf <- c(LCI, UCI)
names(diffMeansconf) <- c("Lower Bound", "Upper Bound")
diffMeansconf
```

```
## Lower Bound Upper Bound
## -0.09881829  1.11472738
```

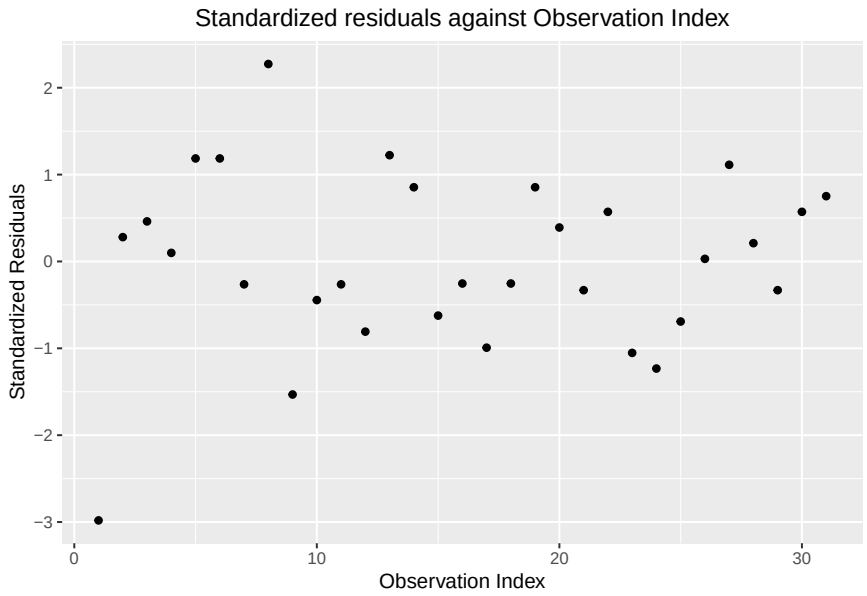
1c. Standard Residual Plot

I reread the data into a different format, time one column for group and one column for phosphate count for simplicity in implementation to the model:

```
phosDataC <- phosData <- read_excel("Table 6.19-PlasmaPhosphate-1.xls",
                        skip = 2, sheet = "Sheet1",
                        range = cell_cols("E:F"), .name_repair = "unique_quiet")
phosDataC$Group <- factor(phosDataC$Group)
lm_1c <- lm("phosphate ~ factor(Group)", data = phosDataC)
```

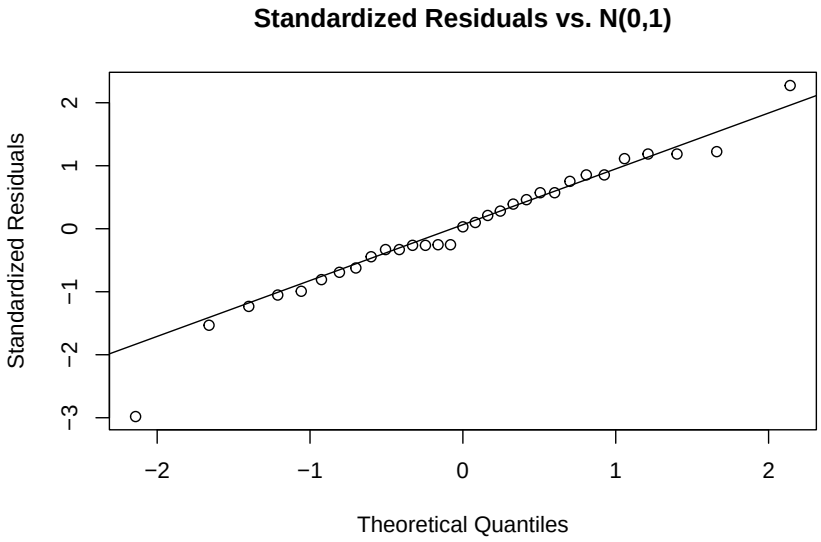
Then, I plot the standardized residuals against observation index:

```
std_resid <- rstandard(lm_1c)
resid_table <- data.frame(cbind(1:length(std_resid), std_resid))
colnames(resid_table) <- c("Index", "std_resid")
ggplot(data = resid_table) + geom_point(aes(x = Index, y = std_resid)) +
  ylab("Standardized Residuals") + xlab("Observation Index") +
  ggtitle("Standardized residuals against Observation Index") +
  theme(plot.title = element_text(hjust = 0.5))
```



Then, I produce a normal probability plot:

```
qqnorm(y = std_resid, ylab = "Standardized Residuals",
       xlab = "Theoretical Quantiles",
       main = "Standardized Residuals vs. N(0,1)")
qqline(y = std_resid)
```



Note that the fit is very good, with the exception of the 1st observation, having a standardized residual of almost -3. Thus, it can be concluded that the residuals are approximately normal.

2. ANOVA, two factor (unblanaced)

First, I read in the data and turn the explanatory variables into factors:

```
facData <- read_excel("Table 6.21-Unbalanced data-1.xls", skip = 2,
                      sheet = "Sheet1", .name_repair = "unique_quiet")
facData$factorA <- factor(facData$factorA)
facData$factorB <- factor(facData$factorB)
```

2a. Testing for interaction effects

Note that factor A has 3 levels, and factor B has 2 levels; therefore, there will be $(3-1)(2-1) = 2$ coefficients corresponding to interaction effects in a full model. Assuming $A1$ and $B1$ as our reference groups, our hypotheses are:

$$H_0 : \beta_{\text{AA2BB2}} = \beta_{\text{AA3BB2}} = 0 \text{ (no interaction) and } H_a : \text{at least one } \beta_{\text{AB}} \neq 0$$

I run the linear model that includes interactions and the model without interactions:

```
lm2_inter <- lm("data ~ factorA + factorB + factorA*factorB", data = facData)
lm2_noInter <- lm("data ~ factorA + factorB", data = facData)
```

These models are nested, and there are 2 coefficients being tested. Thus, in the F statistic, there are 2 degrees of freedom in the numerator and $10-2-(2+1+1) = 4$ in the denominator, I calculate the following F-statistic:

$$F_{stat} = \frac{SSE_{\text{red}} - SSE_{\text{full}}}{q * s^2_{full}}$$

```
f_2a <- (sum(residuals(lm2_noInter)^2) - sum(residuals(lm2_inter)^2))/
  (2 * (summary(lm2_inter)$sigma)^2)
sprintf("F-stat = %f", f_2a)
```

```
## [1] "F-stat = 0.428571"
```

I calculate the p-value of this statistic:

```
sprintf("p-value = %f", pf(f_2a, df1 = 2, df2 = 4, lower.tail = FALSE))
```

```
## [1] "p-value = 0.678201"
```

Note that $0.678201 > 0.05$, so I **fail to reject** H_0.

2b. Testing effect of factor A, controlling for B

The hypotheses in this example are:

$$H_0 : \beta_{\text{A2}} = \beta_{\text{A3}} = 0 \text{ and } H_a : \text{at least one } \beta_{\text{A}} \neq 0$$

If the null hypothesis is true, than the level of factor A has no statistically significant bearing on the data. I have already generated the full model in this case (the one with no interactions). So, I generate the reduced model, and calculate the same f-statistic as in part 2a (here, there would be 2 degrees of freedom in the numerator, and $10-(3+1) = 6$ in the denominator as the full model has 3 parameters):

```
lm2_onlyB <-  lm("data ~ factorB", data = facData)
f_2b <- (sum(residuals(lm2_onlyB)^2) - sum(residuals(lm2_noInter)^2))/
  (2 * (summary(lm2_noInter)$sigma)^2)
sprintf("F-stat = %f", f_2b)
```

```
## [1] "F-stat = 9.023529"
```

I then calculate the p-value of this statistic:

```
sprintf("p-value = %f", pf(f_2b, df1 = 2, df2 = 6, lower.tail = FALSE))
```

```
## [1] "p-value = 0.015533"
```

Since $0.155 < 0.05$, I reject H_0 at $\alpha = 0.05$; factor A seems to be statistically significant.

2c. Testing the effect of factor A, standalone

In this case, the hypotheses are the same as in part B, but the full model is the model with only factor A in it, while the reduced model is a model with only an intercept (in this case the mean of the data). I calculate the F-statistic (in this case, the numerator has 2 degrees of freedom, as before, but denominator has $10 - (2 + 1) = 7$ degrees of freedom, as the model with only factor A has no other parameters):

```
lm2_onlyA <- lm("data ~ factorA", data = facData)
lm2_justMean <- lm("data ~ factorB", data = facData)
f_2c <- (sum(residuals(lm2_justMean)^2) - sum(residuals(lm2_onlyA)^2))/
  (2 * (summary(lm2_onlyA)$sigma)^2)
sprintf("F-stat = %f", f_2c)
```

```
## [1] "F-stat = 6.233333"
```

I then calculate the p-value

```
sprintf("p-value = %f", pf(f_2c, df1 = 2, df2 = 7, lower.tail = FALSE))
```

```
## [1] "p-value = 0.027882"
```

Since $0.0279 < 0.05$, I reject H_0 at $\alpha = 0.05$; once again, factor A seems to be statistically significant.

2d. Comparing the results from 2b and 2c

The difference between our tests for significance in 2b and 2c is that the test in 2c measured the standalone predictive power of factor A on the data (there were no other predictors in the model) while the test in 2b controlled for Factor B; that is, the significance of the coefficients corresponding to different levels of A was a measure of the predictive power of factor A over the mean within a particular level of factor B. Note that the p-value for 2b was lower than 2c; this indicates that **factor A was more powerful in explaining the variance within each level of factor B** than overall variance. This could be due to the unbalanced nature of the data or the fact that mean of data is different between factor B levels.

3. One-factor ANOVA

3a. Calculating the sum sq for Within Groups

To calculate SSB, I first need to calculate the overall mean:

$$\bar{x} = \frac{n_C\bar{x}_C + n_{NO}\bar{x}_{NO} + n_{HO}\bar{x}_{HO}}{n_C + n_{NO} + n_{HO}} = \frac{12(2.93) + 10(3.58) + 11(4.19)}{11 + 10 + 12} = 3.547$$

I use the formula provided in the slides:

$$\begin{aligned} SSB &= \sum_{i \in \text{C,NO,HO}} n_i(\bar{x}_i - \bar{x})^2 \\ &= 12(2.93 - 3.547)^2 + 10(3.58 - 3.547)^2 + 11(4.19 - 3.547)^2 \\ &= \mathbf{9.1271} \end{aligned}$$

3b. Filling in the other missing fields

The other values to be calculated are:

- Mean Squared Between (MSB)
- F statistic for between groups (F_{B})
- p-value for f-stat above
- Sum of Squared Within (SSW)
- Degrees of Freedom Within (df_W)
- Mean Squared Within (MSW)

I start with MSB, SSW, df_W and MSW, using formulas from the slides:

$$\begin{aligned} MSB &= \frac{SSB}{df_B} = \frac{9.1271}{2} = \mathbf{4.56355} \\ SSW &= SST - SSB = 23.7218 - 9.1271 = \mathbf{14.5947} \\ df_W &= n_{\text{total}} - \text{Number of Groups} = 33 - 3 = \mathbf{30} \\ MSW &= \frac{SSW}{df_W} = \frac{14.5947}{30} = \mathbf{0.48649} \end{aligned}$$

Then, I calculate the F-statistic for between groups:

$$F_b = \frac{MSB}{MSW} = \frac{4.56355}{0.48649} = \mathbf{9.38056}$$

For the p-value, note that the statistic above comes from an F distribution with 2 numerator degrees of freedom and $32 - 2 = 30$ denominator degrees of freedom:

```
sprintf("p-value = %f", pf(9.38056, df1 = 2, df2 = 30, lower.tail = FALSE))
```

```
## [1] "p-value = 0.000685"
```

3c. Are the means the same?

The test statistic and p-value correspond to testing $H_0 : \mu_C = \mu_{NO} = \mu_{HO}$. Since $0.000685 < 0.01$, I would reject H_0 at $\alpha = .01, .05,$ and/or $.1$. There is statistically significant evidence that at least one mean is different.

4. National Life Expectancies

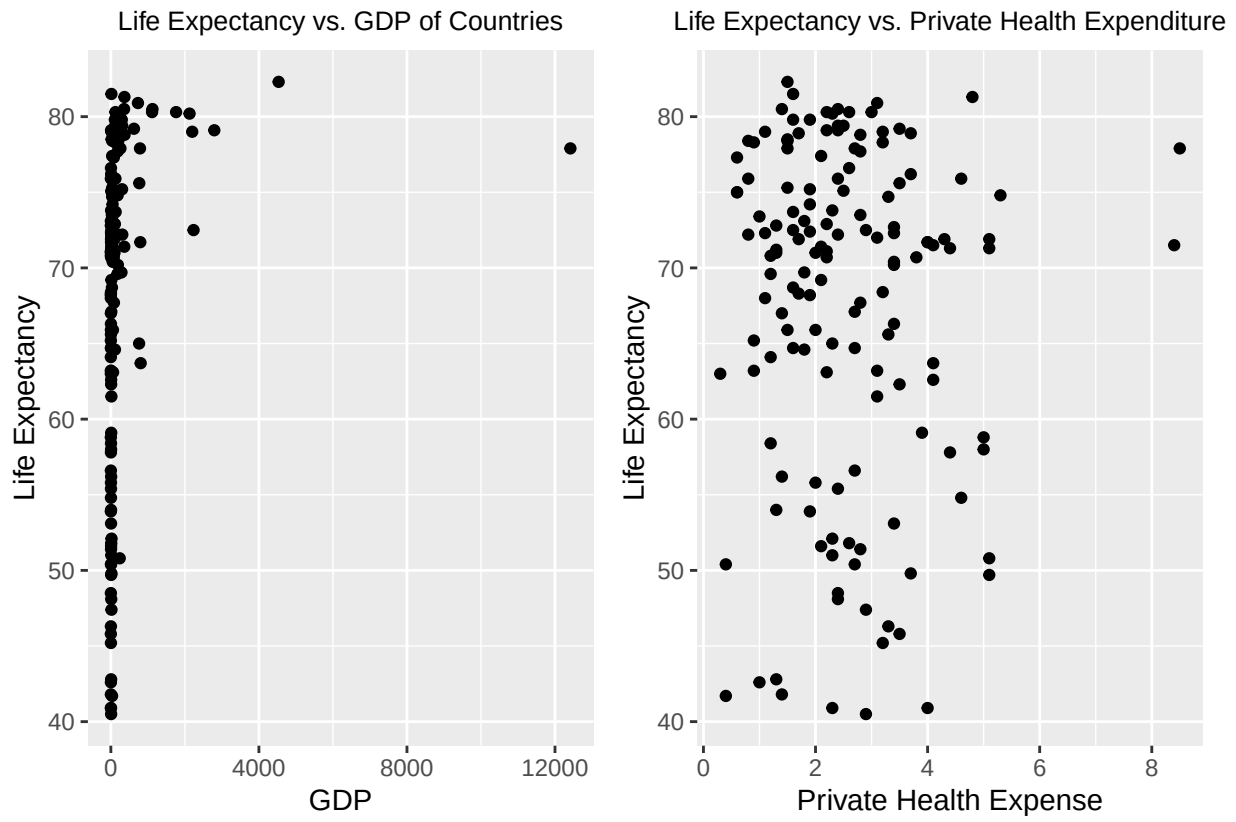
4a. Logarithmic transforms

I first read in data, and check where it is complete:

```
UNData <- read.csv("UNLifeExpectancy-4.csv", header = TRUE)
compData <- subset(UNData,complete.cases(PUBLICEDUCATION,
PRIVATEHEALTH,
FERTILITY,
GDP,
LIFEEXP))
```

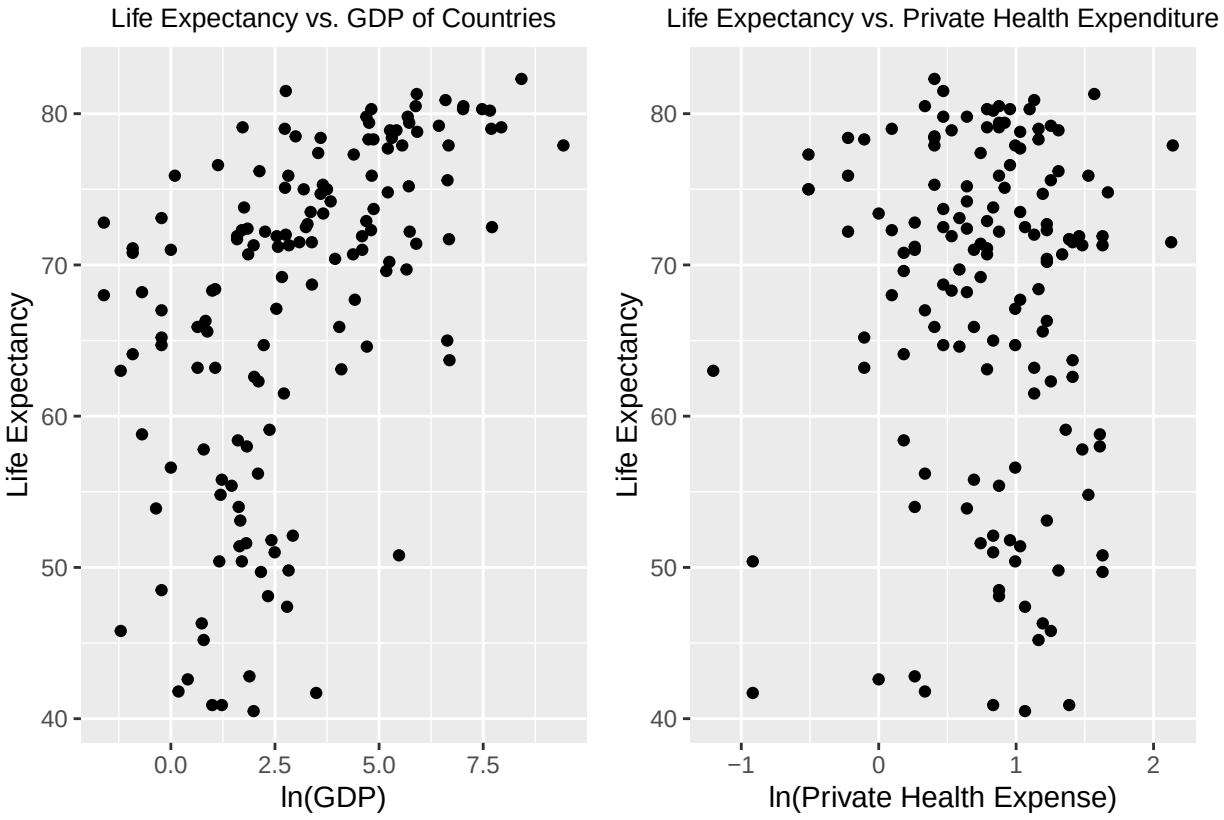
Then, I plot what is desired:

```
gdPlot <- ggplot(data = compData) + geom_point(aes(x = GDP, y = LIFEEXP)) +
  ggtitle("Life Expectancy vs. GDP of Countries") +
  ylab("Life Expectancy") + theme(plot.title = element_text(hjust = 0.5, size = 10))
phPlot <- ggplot(data = compData) + geom_point(aes(x = PRIVATEHEALTH, y= LIFEEXP)) +
  ggtitle("Life Expectancy vs. Private Health Expenditure") +
  xlab("Private Health Expense") + ylab("Life Expectancy") +
  theme(plot.title = element_text(hjust = 0.5, size = 10))
gdPlot + phPlot
```



Note that neither GDP nor Private Health Expenditures appear linearly related with Life Expectancy. There seems to be a relationship, but both are concentrated on the lower ends of their ranges, suggesting that taking the log of both of these variables may make the resulting relationship with life expectancy more linear. I do this in the plots below:

```
compData$lnHEALTH <- log(compData$PRIVATEHEALTH)
compData$lnGDP <- log(compData$GDP)
lngdPlot <- ggplot(data = compData) + geom_point(aes(x = lnGDP, y = LIFEEXP)) +
  ggtitle("Life Expectancy vs. GDP of Countries") + xlab("ln(GDP)") +
  ylab("Life Expectancy") + theme(plot.title = element_text(hjust = 0.5, size = 10))
lnphPlot <- ggplot(data = compData) + geom_point(aes(x = lnHEALTH, y= LIFEEXP)) +
  ggtitle("Life Expectancy vs. Private Health Expenditure") +
  xlab("ln(Private Health Expense)") + ylab("Life Expectancy") +
  theme(plot.title = element_text(hjust = 0.5, size = 10))
lngdPlot + lnphPlot
```



4b. Regression, Standard Residuals, and Leverage

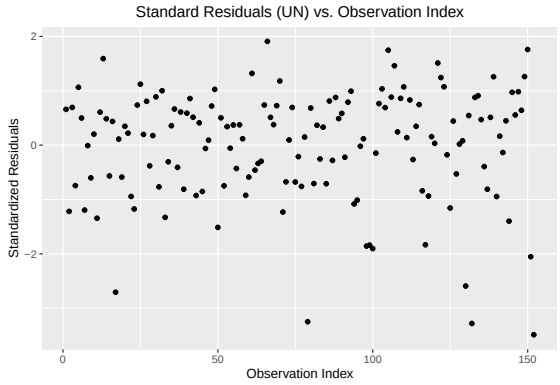
I return to the original dataset and run the regression requested:

```
unReg <- lm("LIFEEXP ~ FERTILITY + PUBLICEDUCATION + log(PRIVATEHEALTH)",
            data = UNData)
```

4b(i). Plots of Standardized residuals

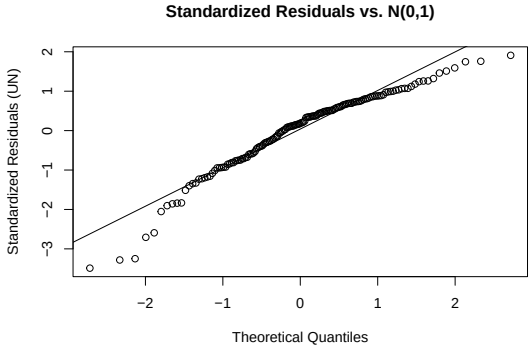
- I plot the standardized residuals against the observation index (the observation indices correspond to the subset of data that is complete) as well as against the normal probability plot:

```
std_resid_UN <- rstandard(unReg)
resid_table_UN <- data.frame(cbind(1:length(std_resid_UN), std_resid_UN))
colnames(resid_table_UN) <- c("Index", "std_resid")
ggplot(data = resid_table_UN) + geom_point(aes(Index, std_resid)) +
  ylab("Standardized Residuals") + xlab("Observation Index") +
  ggtitle("Standard Residuals (UN) vs. Observation Index") +
  theme(plot.title = element_text(hjust = 0.5))
```



Then, I plot it against the standard normal distribution:

```
qqnorm(y = std_resid_UN, ylab = "Standardized Residuals (UN)",
       xlab = "Theoretical Quantiles",
       main = "Standardized Residuals vs. N(0,1)")
qqline(y = std_resid_UN)
```



There may be some evidence that the residuals aren't quite normal, systematically undershooting the qqline at the extremes and overshooting near the center.

4b(ii). Identifying all outliers

- I use the method described in the text to find the indices of the outliers (indices correspond to original data set, not truncated complete data set):

```
outliers <- std_resid_UN > 3 | std_resid_UN < -3
outliers <- which(outliers == TRUE)
names(outliers)
```

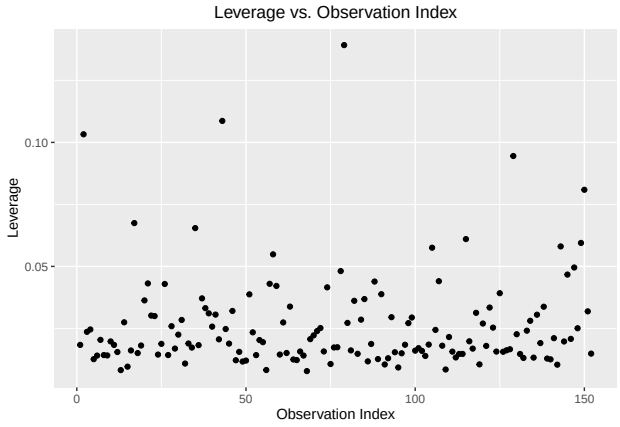
```
## [1] "95"  "159" "185"
```

So observations 95, 159, and 185 are outliers.

4b(iii). Providing a Plot for Leverage

- I provide a leverage plot (the observation indices correspond to the subset of data that is complete):

```
leverage <- hatvalues(unReg)
hat_matrix <- data.frame(cbind(1:length(leverage),leverage))
colnames(hat_matrix) <- c("Index", "Leverage")
ggplot(data = hat_matrix) + geom_point(aes(Index, Leverage)) +
  ylab("Leverage") + xlab("Observation Index") +
  ggtitle("Leverage vs. Observation Index") +
  theme(plot.title = element_text(hjust = 0.5))
```



4b(iv). Finding high leverage points

- Note that $\bar{h} = \frac{p+1}{N} = \frac{4}{152} = \frac{1}{38}$. Via the formula provided in the homework, I look for where $h_{ii} > 3 * \bar{h} = \frac{3}{38}$:

```
high_leverage <- leverage > (3/38)
high_leverage <- which(high_leverage == TRUE)
names(high_leverage)
```

```
## [1] "4"   "53"  "95"  "152" "183"
```

So observations 4, 53, 95, 152, and 183 (indices from original data set) are high leverage.

4b(v). Finding intersection between high leverage, outlier

- Observation 95, which corresponds to the country of **Lesotho**, is both a high leverage point and an outlier. In the truncated data, this corresponds to observation 79. I find the cook's distance of this point:

```
index_trunc <- unname(high_leverage["95"])
sprintf("Lesotho Cook's Distance: %f",
        unname(cooks.distance(unReg)[index_trunc]))
```

```
## [1] "Lesotho Cook's Distance: 0.427029"
```

The formula for Cook's distance is:

$$D_i = \left(\frac{e_i}{s\sqrt{1-h_{ii}}}\right)^2 * \frac{h_{ii}}{(p+1)(1-h_{ii})}$$

Therefore, I calculate the proportion of cook's distance from both the standardized residual and leverage:

```
cooks_resid <- unname((unReg$residuals[index_trunc])/
  (summary(unReg)$sigma *sqrt(1 - leverage[index_trunc])))
cooks_leverage <- unname(leverage[index_trunc]/
                           ((3 + 1)*(1 - leverage[index_trunc])))
sprintf("From Outlier: %f", cooks_resid^2)
```

```
## [1] "From Outlier: 10.560379"
```

```
sprintf("From High Leverage: %f", cooks_leverage)
```

```
## [1] "From High Leverage: 0.040437"
```

If I combine these together, I get back the original cook's distance:

```
sprintf("Cook's Distance: %f", cooks_resid^2 * cooks_leverage)
```

```
## [1] "Cook's Distance: 0.427029"
```

4c. Variance Inflation Factor (VIF)

4c(i). Explanation of Collinearity and a Variance Inflation Factor

- **Collinearity** occurs when two explanatory variables exhibit a high correlation with one another, either positive or negative. Because of the variables' shared movement, slight alterations to data can cause significant swings in coefficients, as drastically different combinations of these two variables can cause similar results. This increases the variance of the linear model and its parameters. The **Variance Inflation Factor**, or VIF, is a variable-specific measure of the scale of this increased variance introduced by a particular variable due to its collinearity with all other variables; it is a function of the R^2 coefficient obtained when running a regression of a particular explanatory variable using all other explanatory variables as predictors.

4c(ii). What constitutes a large VIF?

- A VIF of **10 or higher** constitutes a large VIF for a particular explanatory variable.

4c(iii). Getting $VIF_{lnHEALTH}$

- The formula is $VIF_{lnHEALTH} = \frac{1}{1-R^2_{lnHEALTH}}$. I run the model to obtain the R^2:

```
lm4c_lnHEALTH <- lm("log(PRIVATEHEALTH) ~ FERTILITY + PUBLICEDUCATION",
                    data = UNData)

sprintf("R^2 of lnHEALTH: %f", summary(lm4c_lnHEALTH)$r.squared)
```

```
## [1] "R^2 of lnHEALTH: 0.013125"
```

From this, I calculate the VIF:

```
VIF_lnHEALTH <- 1/(1 - summary(lm4c_lnHEALTH)$r.squared)
sprintf("VIF of lnHEALTH: %f", VIF_lnHEALTH)
```

```
## [1] "VIF of lnHEALTH: 1.013299"
```

4c(iv). Calculating all VIF's without rerunning regression

- I use the following formula:

$$VIF_{x_j} = se\left(\hat{\beta}_j\right)^2 * \frac{s^2_{x_j} * (N-1)}{s^2}$$

  This formula is implemented below:

```
compForVAR <- subset(UNData,complete.cases(PUBLICEDUCATION,PRIVATEHEALTH,
                                           FERTILITY, LIFEEXP))
compForVAR$lnHEALTH <- log(compForVAR$PRIVATEHEALTH)
bigN <- dim(compForVAR)[1]

#For lnHEALTH
sigmasq_lnHEALTH <- unname(diag(vcov(unReg))["log(PRIVATEHEALTH)"])
VIF_lnHEALTH <- sigmasq_lnHEALTH * var(compForVAR$lnHEALTH) * (bigN - 1)/
  summary(unReg)$sigma^2

#For Fertility
sigmasq_Fertility <- unname(diag(vcov(unReg))["FERTILITY"])
VIF_Fertility <- sigmasq_Fertility * var(compForVAR$FERTILITY) * (bigN - 1)/
  summary(unReg)$sigma^2

# For Public Education
```

```r
sigma_sq_PE <- unname(diag(vcov(unReg))["PUBLICEDUCATION"])
VIF_PE <- sigma_sq_PE * var(compForVAR$PUBLICEDUCATION) * (bigN - 1)/
  summary(unReg)$sigma^2

VIFs <- c(VIF_Fertility, VIF_PE, VIF_lnHEALTH)
names(VIFs) <- c("Fertility VIF", "Public Ed. VIF", "ln(HEALTH) VIF")
VIFs
```

```
##  Fertility VIF Public Ed. VIF ln(HEALTH) VIF
##       1.018092       1.029114       1.013299
```