Module 2: Peer Reviewed Assignment

Outline:

The objectives for this assignment:

1. Mathematically derive the values of $\hat{\beta}_0$ and $\hat{\beta}_1$
2. Enhance our skills with linear regression modeling.
3. Learn the uses and limitations of RSS, ESS, TSS and R^2.
4. Analyze and interpret nonidentifiability.

General tips:

1. Read the questions carefully to understand what is being asked.
2. This work will be reviewed by another human, so make sure that you are clear and concise in what your explanations and answers.

```
In [ ]: # Load Required Packages
        library(RCurl) #a package that includes the function getURL(), which allows for reading data from github.
        library(tidyverse)
```

```
── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──

✓ ggplot2 3.3.0     ✓ purrr   0.3.4
✓ tibble  3.0.1     ✓ dplyr   0.8.5
✓ tidyr   1.0.2     ✓ stringr 1.4.0
✓ readr   1.3.1     ✓ forcats 0.5.0

── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ tidyr::complete() masks RCurl::complete()
✖ dplyr::filter()   masks stats::filter()
✖ dplyr::lag()      masks stats::lag()
```

Problem 1: Maximum Likelihood Estimates (MLEs)

Consider the simple linear regression model $Y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$ for $i = 1, \ldots, n, \;\; \varepsilon_i \sim N(0, \sigma^2)$. In the videos, we showed that the least squares estimator in matrix-vector form is $\widehat{\beta} = (\beta_0, \beta_1)^T = (X^T X)^{-1} X^T \mathbf{Y}$. In this problem, you will derive the least squares estimators for simple linear regression without (explicitly) using linear algebra.

Least squares requires that we minimize

$$f(\mathbf{x}; \beta_0, \beta_1) = \sum_{i=1}^{n} \left(Y_i - [\beta_0 + \beta_1 x_i] \right)^2$$

over β_0 and β_1.

1. (a) Taking Derivatives

Find the partial derivative of $f(\mathbf{x}; \beta_0, \beta_1)$ with respect to β_0, and the partial derivative of $f(\mathbf{x}; \beta_0, \beta_1)$ with respect to β_1. Recall that the partial derivative with respect to x of a multivariate function $h(x, y)$ is calculated by taking the derivative of h with respect to x while treating y constant.

b_zero

$$\frac{\partial f}{\partial \beta_0} = \frac{\partial}{\partial \beta_0} \sum_{i=1}^{n} (Y_i - [\beta_0 + \beta_1 x_i])^2$$

b_one

$$\frac{\partial f}{\partial \beta_1} = \frac{\partial}{\partial \beta_1} \sum_{i=1}^{n} (Y_i - [\beta_0 + \beta_1 x_i])^2$$

1. (b) Solving for $\hat{\beta}_0$ and $\hat{\beta}_1$

Use **1. (a)** to find the minimizers, $\widehat{\beta}_0$ and $\widehat{\beta}_1$, of f. That is, set each partial derivative to zero and solve for β_0 and β_1. In particular, show

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \qquad \text{and} \qquad \widehat{\beta}_0 = \bar{Y} - \widehat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(Y_i - \bar{Y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\beta}_0 = \bar{Y} - \hat{\beta}_1 \bar{x}$$

Problem 2: Oh My Goodness of Fit!

In the US, public schools have been slowly increasing class sizes over the last 15 years [https://stats.oecd.org/Index.aspx?DataSetCode=EDU_CLASS]. The general cause for this is because it saves money to have more kids per teacher. But how much money does it save? Let's use some of our new regression skills to try and figure this out. Below is an explanation of the variables in the dataset.

Variables/Columns:
School
Per-Pupil Cost (Dollars)
Average daily Attendance
Average Monthly Teacher Salary (Dollars)
Percent Attendance
Pupil/Teacher ratio

Data Source: E.R. Enlow (1938). "Do Small Schools Mean Large Costs?," Peabody Journal of Educaltion, Vol. 16, #1, pp. 1-11

In [ ]:

```
school.data = read_table("school.dat")
names(school.data) = c("school", "cost", "avg.attendance", "avg.salary", "pct.attendance", "pup.tch.ratio")
head(school.data)
dim(school.data)
```

```
Parsed with column specification:
cols(
  Adair = col_character(),
  `66.90` = col_double(),
  `451.4` = col_double(),
  `160.22` = col_double(),
  `90.77` = col_double(),
  `33.8` = col_double()
)
```

A tibble: 6 × 6

school	cost	avg.attendance	avg.salary	pct.attendance	pup.tch.ratio
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Calhoun	108.57	219.1	161.79	89.86	23.0
Capitol View	70.00	268.9	136.37	92.44	29.4
Connally	49.04	161.7	106.86	92.01	29.4
Couch	71.51	422.1	147.17	91.60	29.2
Crew	61.08	440.6	146.24	89.32	36.3
Davis	105.21	139.4	159.79	86.51	22.6

43 · 6

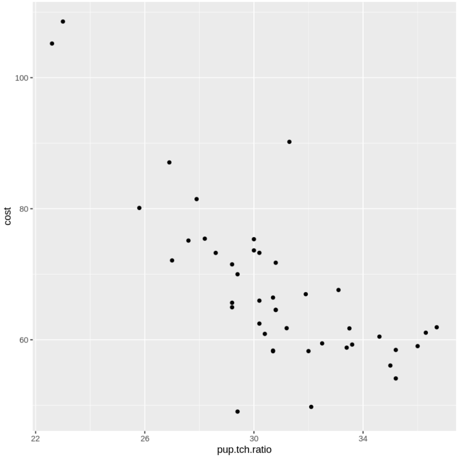
2. (a) Create a model

Begin by creating two figures for your model. The first with `pup.tch.ratio` on the x-axis and `cost` on the y-axis. The second with `avg.salary` on the x-axis and `cost` on the y-axis. Does there appear to be a relation between these two predictors and the response.

Then fit a multiple linear regression model with `cost` as the response and `pup.tch.ratio` and `avg.salary` as predictors.

In [ ]:

```
# Your Code Here
ggplot(school.data, aes(x = pup.tch.ratio, y = cost)) +
geom_point()
ggplot(school.data, aes(x = avg.salary, y = cost)) +
geom_point()
lm_mlr = lm(cost ~ pup.tch.ratio + avg.salary, data = school.data)
summary(lm_mlr)
```

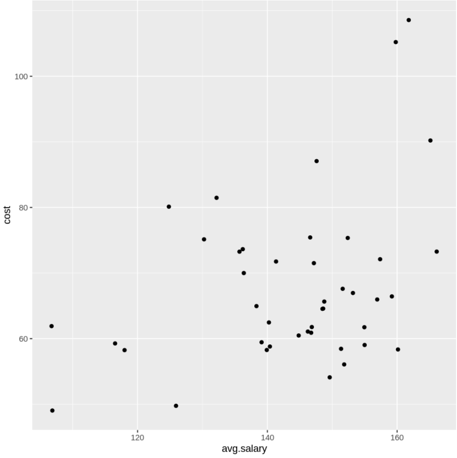


```
Call:
lm(formula = cost ~ pup.tch.ratio + avg.salary, data = school.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.8290  -5.2752  -0.8332   3.8253  19.6986 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   120.23756   17.73230   6.781 3.79e-08 ***
pup.tch.ratio  -2.82585    0.37714  -7.493 3.90e-09 ***
avg.salary      0.24061    0.08396   2.866   0.0066 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7.721 on 40 degrees of freedom
Multiple R-squared:  0.6372,	Adjusted R-squared:  0.6191 
F-statistic: 35.13 on 2 and 40 DF,  p-value: 1.559e-09
```



2. (b) RSS, ESS and TSS

In the code block below, manually calculate the RSS, ESS and TSS for your MLR model. Print the results.

In [ ]:

```
# Your Code Here
n = dim(school.data)[1]
p = 2
ess = sum((fitted(lm_mlr) - mean(school.data$cost))^2);
rss = sum(residuals(lm_mlr)^2);
tss = with(school.data, sum((cost - mean(cost))^2));
paste0("ESS:", round(ess,2))
paste0("RSS: ", round(rss,2))
paste0("TSS: ", round(tss,2))
```

'ESS:4188.57'

'RSS: 2384.6'

'TSS: 6573.17'

2. (c) Are you Squared?

Using the values from **2.b**, calculate the R^2 value for your model. Check your results with those produced from the `summary()` statement of your model.

In words, describe what this value means for your model.

In [ ]:

```
# Your Code Here
r_squared = 1-rss/tss;
paste0("R Squared:", round(r_squared,2), ". So, ",round(r_squared,2), "% of the vairance in cost can be explained by the responses")
```

'R Squared:0.64. So, 0.64% of the vairance in cost can be explained by the responses'

2. (d) Conclusions

Describe at least two advantages and two disadvantages of the R^2 value.

Advantages

- It is simple to check if data fits to the model.
- It is easy to calculate and understand.

Disadvantages

- It can be low but still indicate a good fit (if there is a lot of variability)
- It doesn’t explain causality

Problem 3: Identifiability

This problem might require some outside-of-class research if you haven't taken a linear algebra/matrix methods course.

Matrices and vectors play an important role in linear regression. Let's review some matrix theory as it might relate to linear regression.

Consider the system of linear equations

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{i,j} + \varepsilon_i, \tag{1}$$

for $i = 1, \ldots, n$, where n is the number of data points (measurements in the sample), and $j = 1, \ldots, p$, where

1. $p + 1$ is the number of parameters in the model.
2. Y_i is the i^{th} measurement of the *response variable*.
3. $x_{i,j}$ is the i^{th} measurement of the j^{th} *predictor variable*.
4. ε_i is the i^{th} *error term* and is a random variable, often assumed to be $N(0, \sigma^2)$.
5. β_j, $j = 0, \ldots, p$ are *unknown parameters* of the model. We hope to estimate these, which would help us characterize the relationship between the predictors and response.

3. (a) MLR Matrix Form

Write the equation above in matrix vector form. Call the matrix including the predictors X, the vector of Y_is $\mathbf{Y}$, the vector of parameters β, and the vector of error terms ε. (This is more LaTeX practice than anything else...)**

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

3. (b) Properties of this matrix

In lecture, we will find that the OLS estimator for β in MLR is $\widehat{\boldsymbol{\beta}} = (X^T X)^{-1} X^T \mathbf{Y}$. Use this knowledge to answer the following questions:

1. What condition must be true about the columns of X for the "Gram" matrix $X^T X$ to be invertible?

2. What does this condition mean in practical terms, i.e., does X contain a deficiency or redundancy?

3. Suppose that the number of measurements (n) is less than the number of model parameters ($p + 1$). What does this say about the invertibility of $X^T X$? What does this mean on a practical level?

4. What is true about about $\widehat{\boldsymbol{\beta}}$ if $X^T X$ is not invertible?

1. the columns of must be linearly independent
2. the predictor is not introducing any new information that hasn't already been found in another predictor
3. we have more paramaters than measurements - it wont' be invertable
4. it has infinite solutions

Problem 4: Downloading...

The following data were collected to see if time of day madea difference on file download speed. A researcher placed a file on a remote server and then proceeded to download it at three different time periods of the day. They downloaded the file 48 times in all, 16 times at each Time of Day (`time`), and recorded the Time in seconds (`speed`) that the download took.

4. (a) Initial Observations

The `downloading` data is loaded in and cleaned for you. Using `ggplot` , create a boxplot of `speed` vs. `time` . Make some basic observations about the three categories.

In [ ]:

```
# Load in the data and format it
downloading = read.csv("downloading.txt", sep="\t")
names(downloading) = c("time", "speed")
# Change the types of brand and form to categories, instead of real numbers
downloading$time = as.factor(downloading$time)
summary(downloading)
```

```
                 time        speed      
 Early (7AM)       :16   Min.   : 68.0  
 Evening (5 PM)    :16   1st Qu.:129.8  
 Late Night (12 AM):16   Median :198.0  
                         Mean   :193.2  
                         3rd Qu.:253.0  
                         Max.   :367.0  
```

In [ ]:

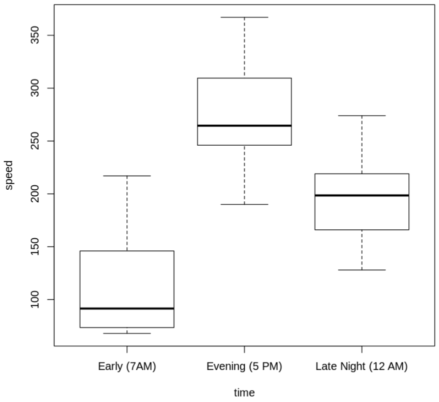
```
summary(lm(speed ~ time, data = downloading))
boxplot(speed ~ time, data = downloading)
```

```
Call:
lm(formula = speed ~ time, data = downloading)

Residuals:
    Min      1Q  Median      3Q     Max 
-83.312 -34.328  -5.187  26.250 103.625 

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)    
(Intercept)              113.37      11.79   9.619 1.73e-12 ***
timeEvening (5 PM)       159.94      16.67   9.595 1.87e-12 ***
timeLate Night (12 AM)    79.69      16.67   4.781 1.90e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 47.15 on 45 degrees of freedom
Multiple R-squared:  0.6717,	Adjusted R-squared:  0.6571 
F-statistic: 46.03 on 2 and 45 DF,  p-value: 1.306e-11
```



speeds are highest in the evening and slowest in the morning

4. (b) How would we model this?

Fit a regression to these data that uses `speed` as the response and `time` as the predictor. Print the summary. Notice that the result is actually *multiple* linear regression, not simple linear regression. The model being used here is:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \varepsilon_i$$

where

1. $X_{i,1} = 1$ if the i^{th} download is made in the evening (5 pm).
2. $X_{i,2} = 1$ if the i^{th} download is made at night (12 am).

Note: If $X_{i,1} = 0$ and $X_{i,2} = 0$, then the i^{th} download is made in the morning (7am).

To confirm this is the model being used, write out the explicit equation for your model - using the parameter estimates from part (a) - and print out it's design matrix.

In [ ]:

```
# Your Code Here
model <- lm(speed ~ time, data = downloading)
beta_0 <- coef(model)[1]
beta_1 <- coef(model)[2]
beta_2 <- coef(model)[3]


paste0("Speed = ", beta_0, " + ", beta_1, "* Evening + ", beta_2, "* Night + ε")

design_matrix <- model.matrix(model)
head(design_matrix)
```

'Speed = 113.375 + 159.9375* Evening + 79.6875* Night + ε'

A matrix: 6 × 3 of type dbl

	(Intercept)	timeEvening (5 PM)	timeLate Night (12 AM)
1	1	0	0
2	1	0	0
3	1	0	0
4	1	0	0
5	1	0	0
6	1	0	0

4. (c) Only two predictors?

We have three categories, but only two predictors. Why is this the case? To address this question, let's consider the following model:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \beta_2 X_{i,3} + \varepsilon_i$$

where

1. $X_{i,1} = 1$ if the i^{th} download is made in the evening (5 pm).
2. $X_{i,2} = 1$ if the i^{th} download is made at night (12 am).
3. $X_{i,3} = 1$ if the i^{th} download is made in the morning (7 am).

Construct a design matrix to fit this model to the response, `speed`. Determine if something is wrong with it. Hint: Analyze the design matrix.

In [ ]:

```
design_matrix_wrong <- cbind(design_matrix, ifelse(design_matrix[, 2] == 0 & design_matrix[, 3] == 0, 1, 0))
colnames(design_matrix_wrong)[4] <- 'timeMorning (7 AM)'
head(design_matrix_wrong)
```

A matrix: 6 × 4 of type dbl

	(Intercept)	timeEvening (5 PM)	timeLate Night (12 AM)	timeMorning (7 AM)
1	1	0	0	1
2	1	0	0	1
3	1	0	0	1
4	1	0	0	1
5	1	0	0	1
6	1	0	0	1

'timeMorning (7 AM)' is a linear combination of 'timeEvening (5 PM)' and 'timeLate Night (12 AM)'

4. (d) Interpretation

Interpret the coefficients in the model from **4.b**. In particular:

1. What is the difference between the mean download speed at 7am and the mean download speed at 5pm?
2. What is the mean download speed (in seconds) in the morning?
3. What is the mean download speed (in seconds) in the evening?
4. What is the mean download speed (in seconds) at night?

In [ ]:

```r
meanMorning = mean(downloading$speed[downloading$time == 'Early (7AM)'])
meanEvening = mean(downloading$speed[downloading$time == 'Evening (5 PM)'])
meanNight = mean(downloading$speed[downloading$time == 'Late Night (12 AM)'])
paste0("1: ", meanEvening-meanMorning)
paste0("2: ", meanMorning)
paste0("3: ", meanEvening)
paste0("4: ", meanNight)
```

```
'1: 159.9375'
'2: 113.375'
'3: 273.3125'
'4: 193.0625'
```