STAT 6012: Linear Models for Data Science
Class Activity 5

Due date: Monday, July 22 by 10:50 am Via Canvas.

Complete the following questions an R Markdown file and submit your compiled HTML file. If you are working in a group, list the names (last, first) of the group members in alphabetical order of last names.

The attached dataset contains information compiled by the World Health Organization and the United Nations to track factors that affect life expectancy in 2015.

Our goal is to build a simple linear regression model to predict `life.Expectancy` using `GDP` based on the least squares method. To make calculations easier, the following sample R code subsets the data on the two variables, as well as removes all missing values.

```
life_data2<-select(life_data,Life.expectancy, GDP)%>%
  na.omit()
```

1. [3] Exploratory data analysis:

   (a) [1] Visualize the two variables to ascertain if a linear model is suitable.

   (b) [1] Calculate the strength of linear relationship between the two variables.

   (c) [1] Based on Part (a) and (b), is a linear model appropriate for predicting `life.Expectancy` using `GDP`?

   We will now proceed to estimate the coefficients for a simple linear model.

2. [2] We proved in class that the least square estimates (also the same as the MLE) for the slope is $\frac{rs_y}{s_x}$ and the y-intercept is $\bar{y} - \hat{\beta}_1\bar{x}$. Use this result to estimate the coefficients for the simple linear model.

3. [1] Find the estimates of the simple linear regression using the `lm()` function in R.

4. We also deduced in class that the general least square estimates for y-intercept and slope(s) is given by:

   $$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

   where $\mathbf{X}$ is the design matrix, and $\mathbf{Y}$ is vector of values of the response variable.

   (a) [2] Use R to generate the design matrix, $\mathbf{X}$ for this simple linear regression mondel.

   (b) [2] Use R to find the model coefficients using $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$.

   Hints:

   - The `rep()` can help generate a value by a specified number of times.
   - `as.matrix()` might be helpful to convert a dataframe to a matrix.
   - `%*%` helps with matrix multiplication.
   - `t()` function helps find the transpose of a matrix.
   - `solve()` function helps find the inverse of a matrix.