Homework 4: Multiple Linear Regression

Name: HARISH NANDHAN SHANMUGAM

This assignment is due on Gradescope by **Friday February 21st at 5:00PM**. If you submit the assignment by this deadline, you will receive 2 bonus points. If you need a little extra time, you may submit your work by **Monday February 24th at 5:00PM**. Your solutions to theoretical questions should be done in Markdown directly below the associated question. Your solutions to computational questions should include any specified R code and results as well as written commentary on your conclusions. Remember that you are encouraged to discuss the problems with your classmates, but **you must write all code and solutions on your own**.

NOTES:

- There are 2 total questions on this assignment.
- If you're not familiar with typesetting math directly into Markdown then by all means, do your work on paper first and then typeset it later. Remember that there is a reference guide linked here. **All** of your written commentary, justifications and mathematical work should be in Markdown.
- Because you can technically evaluate notebook cells in a non-linear order, it's a good idea to do Kernel → Restart & Run All as a check before submitting your solutions. That way if we need to run your code you will know that it will work as expected.
- It is **bad form** to make your reader interpret numerical output from your code. If a question asks you to compute some value from the data you should show your code output **AND** write a summary of the results in Markdown directly below your code.
- This probably goes without saying, but... For any question that asks you to calculate something, you **must show all work and justify your answers to receive credit**. Sparse or nonexistent work will receive sparse or nonexistent credit.

Problem 1 (STAT 5010 Students Only) (50 Points) Trace of the hat matrix

Define the trace of a square matrix

$$A=\begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix}$$

to be $tr(A)=\sum_{i=1}^{n} a_{i,i}$, i.e., the sum of the diagonal elements of A.

PART A: Let B be a $m \times n$ matrix and C $n \times m$ matrix. Prove that $tr(BC) = tr(CB)$.

The trace of a square matrix A is defined as:

$$tr(A) = \sum_{i=1}^{k} A_{ii} \text{(Sum of its diagonal elements)}$$

For two matrices $B(m \times n)$ and $C(n \times m)$, we consider the product $BC(m \times m)$:

$$tr(BC) = \sum_{i=1}^{m} (BC)_{ii} \text{(Sum of diagonal elements of } BC)$$

Expanding the diagonal elements of BC:

$$(BC)_{ii} = \sum_{k=1}^{n} B_{ik} C_{ki} \text{(Matrix multiplication formula for diagonal elements)}$$

Thus, the trace of BC is:

$$tr(BC) = \sum_{i=1}^{m} \sum_{k=1}^{n} B_{ik} C_{ki}$$

Since summation is commutative, we can swap the summation order:

$$tr(BC) = \sum_{k=1}^{n} \sum_{i=1}^{m} C_{ki} B_{ik}$$

Now, consider the trace of $CB(n \times n)$:

$$tr(CB) = \sum_{i=1}^{n} (CB)_{ii}$$

Expanding the diagonal elements of CB:

$$(CB)_{ii} = \sum_{k=1}^{m} C_{ik} B_{ki}$$

Thus, the trace of CB is:

$$tr(CB) = \sum_{i=1}^{n} \sum_{k=1}^{m} C_{ik} B_{ki}$$

Comparing the two results:

$$tr(BC) = \sum_{k=1}^{n} \sum_{i=1}^{m} C_{ki} B_{ik} \text{(Equation 1)}$$

$$tr(CB) = \sum_{i=1}^{n} \sum_{k=1}^{m} C_{ik} B_{ki} \text{(Equation 2)}$$

Since both expressions are identical, we conclude:

$$tr(BC) = tr(CB) \text{(Hence proved)}$$

PART B: Let H be the hat matrix as defined in class. Prove that $tr(H) = p+1$, where p is the number of parameters in the regression.

The hat matrix H is defined as:

$$H = X(X^T X)^{-1} X^T$$

where X has dimensions $N \times (p+1)$, with p predictors and 1 intercept.

A key property of the trace function is its cyclic property:

$$tr(ABC) = tr(CAB) = tr(BAC)$$

Applying this property to the hat matrix:

$$tr(H) = tr\left(X(X^T X)^{-1} X^T\right)$$

Using the cyclic property of trace, we rearrange:

$$tr\left(X(X^T X)^{-1} X^T\right) = tr\left((X^T X)^{-1} X^T X\right)$$

Since $(X^T X)^{-1} X^T X = I_{p+1}$, where I_{p+1} is the identity matrix of size $(p+1) \times (p+1)$,

Taking the trace of the identity matrix:

$$tr(I_{p+1}) = p+1$$

Thus, we conclude:

$$tr(H) = p+1$$

Comparing the two results:

$$tr(H) = tr(I_{p+1}) \text{(Equation 1)}$$

$$p+1 = tr(H) \text{(Equation 2)}$$

Since both expressions are identical, we conclude:

$$tr(H) = p+1 \text{(Hence proved)}$$

Problem 2 - Multiple Linear Regression and Model Selection (50 points)

We will further examine the `Fish.csv` dataset in this problem.

"This dataset is a record of 7 common different fish species in fish market sales. With this dataset, a predictive model can be performed using machine friendly data and estimate the weight of fish can be predicted."

Response:

- Weight (in grams)

Features:

- Length1 (vertical length in cm)
- Length2 (diagonal length in cm)
- Length3 (cross length in cm)
- Height (in cm)
- Width (diagonal width in cm)

The species name of the fish is also given.

Part A: Read the data from the csv. As you are reading in `Fish.csv`, drop the species column as it is non-numerical.

Also, make sure to re-order the columns so that the response variable is the last column.

```
df <- read.csv("Fish.csv")
df <- df[, -1]
df <- df[, c(setdiff(names(df), "Weight"), "Weight")]
head(df)
```

```
  Length1 Length2 Length3 Height  Width  Weight
1 23.2    25.4    30.0    11.5200 4.0200 242
2 24.0    26.3    31.2    12.4800 4.3056 290
3 23.9    26.5    31.1    12.3778 4.6961 340
4 26.3    29.0    33.5    12.7300 4.4555 363
5 26.5    29.0    34.0    12.4440 5.1340 430
6 26.8    29.7    34.7    13.6024 4.9274 450
```

Part B: Fit a multiple linear regression model to the data.

- print the regression coeficients to the screen.
- Use a Markdown cell to specify the MLR model in the form: $ \hat{y} = \beta_0+\ beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p $

```
mlr_model = lm(Weight~Length1+Length2+Length3+Height+Width,data=df)
summary(mlr_model)
```

```

Call:
lm(formula = Weight ~ Length1 + Length2 + Length3 + Height + 
    Width, data = df)
```

```
Residuals:
    Min      1Q  Median      3Q     Max
-243.69  -65.10  -25.52   57.98  447.25

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -499.587     29.572 -16.894  < 2e-16 ***
Length1       62.355     40.209   1.551  0.12302
Length2       -6.527     41.759  -0.156  0.87601
Length3      -29.026     17.353  -1.673  0.09643 .
Height        28.297      8.729   3.242  0.00146 **
Width         22.473     20.372   1.103  0.27169
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 123.2 on 153 degrees of freedom
Multiple R-squared:  0.8853,     Adjusted R-squared:  0.8815
F-statistic: 236.2 on 5 and 153 DF,  p-value: < 2.2e-16
```

$ \hat{y} = -499.58 + 62.35 * Length1 - 6.52 * Length2 - 29.02 * Length3 + 28.29 * Height + 22.47 * Width $

Part C: Perform the appropriate statistical hypothesis test at the $\alpha = 0.01$ significance level to determine if *at least one* of the features is related to the the response y.

- H0 : No predictor values are related to the target response $\beta 1 = \beta 2 = \beta 3 = \beta 4 \ldots \beta p = 0$
- H1 : Atleast one predictor value is related to the target response (βk is not equal to zero for k = 1,2,3..p)

The hypothesis test is performed using this formula:

$$F = \frac{\frac{SST - SSE}{p}}{\frac{SSE}{n - p - 1}}$$

```
SSE =  sum(residuals(mlr_model)^2)
SSR =  sum((fitted(mlr_model) - mean(df$Weight))^2)
SST =  sum(((df$Weight) - mean(df$Weight))^2)
cat("SSE: ", SSE)
cat("\nSSR: ", SSR)
cat("\nSST: ", SST)

p = 5
n = nrow(df)

F_test = ((SST - SSE)/p)/(SSE/(n-p-1))
alpha = 0.01
F_crit = qf(alpha, df1 = p, df2 = n-p-1, lower.tail = FALSE)
```

```
cat("\nF stat calc: ", F_test)
cat("\nF Critical: ", F_crit)
```

```
SSE:  2322653
SSR:  17924806
SST:  20247459
F stat calc:  236.152
F Critical:  3.139089
```

$F_{Calc} > F_{Crit}$ so, we reject the null hypothesis and state that atleast one of the predictors is related in predicting the target response.

Part D: Write a function `forward_select(df, resp_str, maxk)` that takes in the DataFrame, the name of the column corresponding to the response, and the maximum number of desired features, and returns a list of feature names corresponding to the `maxk` most important features via forward selection. At each stage in forward selection you should add the feature whose inclusion in the model would result in the lowest sum of squared errors (SSE). Use your function to determine the best $k = 3$ features to include in the model. Clearly indicate which feature was added in each stage.

Note: The point of this exercise is to see if you can implement **foward_select** yourself. You may not call any R method that explicitly performs forward selection.

```
forward_select = function(df, resp_str, maxk) {

  y = df[[resp_str]]
  X_all = df[, !(names(df) %in% resp_str)]

  selected_features = c()
  remaining_features = names(X_all)

  for (i in 1:maxk) {
    best_feature = NULL
    best_sse = Inf

    for (feature in remaining_features) {
      temp_features = c(selected_features, feature)
      X_temp = X_all[, temp_features, drop = FALSE]
      X_temp = cbind(1, X_temp)

      model = lm(y ~ ., data = data.frame(y = y, X_temp))

      sse = sum(residuals(model)^2)

      if (sse < best_sse) {
        best_sse = sse
        best_feature = feature
      }
    }
```

```
    if (!is.null(best_feature)) {
      selected_features = c(selected_features, best_feature)
      remaining_features = setdiff(remaining_features, best_feature)
      cat(sprintf("%d: feature '%s' with SSE = %.2f\n", i,
best_feature, best_sse))
    }
  }

  return(selected_features)
}

selected_features = forward_select(df, "Weight", 3)
selected_features
```

```
1: feature 'Length3' with SSE = 2996433.36
2: feature 'Width' with SSE = 2489410.25
3: feature 'Height' with SSE = 2473413.58

[1] "Length3" "Width"   "Height"
```

Part E: Write down the reduced multiple linear regression model, including estimated parameters, obtained by your forward selection process.

```
mlr_model_reduced = lm(Weight~Length3+Width+Height,data=df)
summary(mlr_model_reduced)
```

```

Call:
lm(formula = Weight ~ Length3 + Width + Height, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-246.79  -77.57  -33.26   82.47  453.51

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -521.000     29.331 -17.763  < 2e-16 ***
Length3       19.445      1.812  10.728  < 2e-16 ***
Width         62.833     14.560   4.315 2.83e-05 ***
Height         3.853      3.849   1.001    0.318
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 126.3 on 155 degrees of freedom
Multiple R-squared:  0.8778,    Adjusted R-squared:  0.8755
F-statistic: 371.3 on 3 and 155 DF,  p-value: < 2.2e-16
```

$ \hat{y} = -521 + 19.445 * Length3 + 62.833 * Width + 3.853 * Height $