STAT 534 Statistical Data Science I: Exam I (Take-Home Portion)

Erick Calderon-Morales

Fall 2021

Due Date: October 17 (Sunday)

Instructions: There are 50 points plus 10 extra credits on this portion of the exam. You may use any reference material for your exam. However, you are not to discuss any questions about the exam with anyone other than yourself. Please contact me with any questions and I will respond as soon as possible within the same day. Be sure to write/type your answers in complete sentences and show your work. Any items including R output not commented on will receive no credit. Attach your R code as an appendix or as a separate file. (If you use R markdown, you may keep the code inserted inline.)

```r
# Packages
library(tidyverse)
library(janitor)
```

1. (20+10 pts) In this problem, we will generate simulated data, and will then apply best subset selection to this data using the validation method. The goal is to explore that as model size increases, the training error will necessarily decrease, but the testing error may not.

(a) (5 pts) Generate a data set with p = 20 predictors and n = 1000 observations according to the model:

$$Y = \beta_0 \ + \ \beta_1 X_1 \ ... \ + \ \beta_p X_p \ + \ \epsilon$$

where $X_j \sim N(0, 1)$ and $\epsilon \ \sim N(0, 1)$ independently. Randomly select your β values but let some elements to be exactly zero. Then split your data set into a training set containing 100 observations and a testing set containing 900 observations.

```r
set.seed(123)

# Generate e values with mean 0 and sd 1
epsilon <- rnorm(1000, mean = 0, sd = 0)

# Generate x values with mean 0 and sd 1
n = 1000
variables = 20

# Create empty data frame
empty_data_set <- data.frame(matrix(numeric(variables * n),
                                    ncol = variables,
                                    nrow = n))

for (each_variable in seq(along = 1:variables)){

     # Get random data and append to data frame
     empty_data_set[,each_variable] <- rnorm(1000, mean = 0, sd = 1)
}
```

```
# Clean data set
x_variables <-
    empty_data_set %>%
    clean_names()
```

```
set.seed(123)

# Generate Y using my betas and simulated data
y <- x_variables$x1       + 9*(x_variables$x2)   + 7*(x_variables$x3)   +
     65*(x_variables$x4)  + 0*(x_variables$x5)   + 5*(x_variables$x6)   +
     75*(x_variables$x7)  + 76*(x_variables$x8)  + 34*(x_variables$x9)  +
     12*(x_variables$x10) + 34*(x_variables$x11) + 45*(x_variables$x12) +
     82*(x_variables$x13) + 23*(x_variables$x14) + 0*(x_variables$x15)  +
     90*(x_variables$x16) + 0*(x_variables$x17)  + 1*(x_variables$x18)  +
     19*(x_variables$x19) + 20*(x_variables$x19) + epsilon
```

```
# Join data
data_set <- cbind(y,x_variables)
```

```
# slip data into train and test
# Get index
train <- sample(1:n, 900)

# Test set
data_set_train <- data_set[train,]
nrow(data_set_train)
```

```
[1] 900
```

```
# Train set
data_set_test <-  data_set[-train,]
nrow(data_set_test)
```

```
[1] 100
```

(b) (3 pts) Perform best subset selection on the training set, and plot the training set MSE associated with the best model of each size. (Hint: regsubsets() returns error (or residual) sum of squares (rss) for each model and MSE = RSS/n.)