STAT 1361 - Homework 4

Gordon Lu

3/4/2021

Problem 2)

ISLR Chapter 5 Conceptual Exercise 4

Solution:

We can use bootstrapping to estimate the standard deviation of our predictions. Let's say we're given some sample (X) of size N from the population (P). Normally, in order to estimate the std, we could just sample the population a large number of times and view the distribution of the std after sampling over and over again. However, this is infeasible since we only have access to a certain subset of the data, X. As such, we will take a random sample (X') of X of size N (same size as the sample) with replacement (this is called a bootstrap sample), calculate its standard deviation, and repeat this process B times, each time generating a new bootstrap sample and calculating a new standard deviation on X'. Those B standard deviations are represented as $\hat{\theta}_1 \dots \hat{\theta}_b$. The process we will be using is bagging (bootstrap aggregating). In order to calculate the estimated standard deviation, we can take the average of the std. estimates as follows:

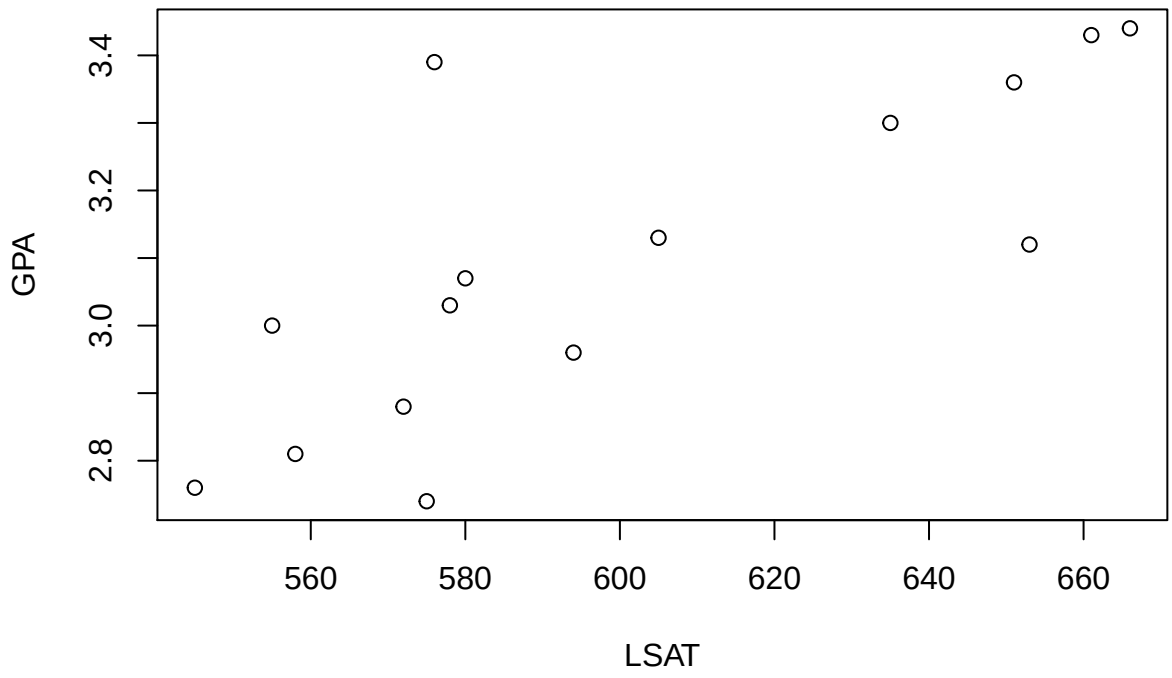
$$\bar{\theta} = \frac{1}{B} \sum_{i=1}^{B} \hat{\theta}_i$$

Problem 3)

3a)

The law data is stored in the bootstrap package in R. Does there appear to be a strong relationship? Calculate the correlation.

```
library(bootstrap)
data(law)
plot(law)
```




```
law_r <- cor(law)[2]
law_r
```

```
## [1] 0.7763745
```

Yes, there appears to be a strong relationship. The correlation between GPA and the LSAT is 0.7764, which is fairly close to 1.

3b)

Take B = 1000 bootstrap replicates of the data to get 1000 bootstrap estimates of correlation. Create a histogram of the bootstrap correlations. Insert a red vertical line in the histogram showing the correlation calculated on the original data.

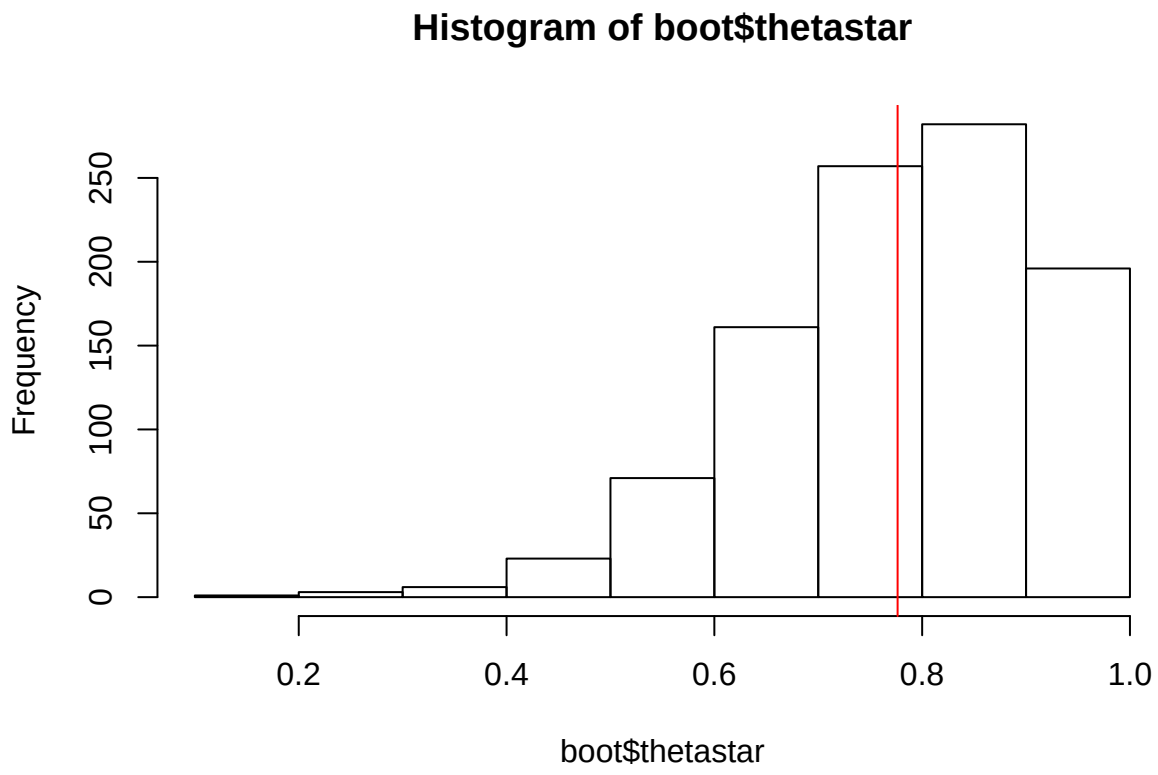
Solution:

```
set.seed(1)

calc_rho <- function(r, data){
  cor(data[r, "LSAT"], data[r, "GPA"])
}
rho_0 <- cor(law$LSAT, law$GPA)
```

```r
boot <- bootstrap(x = 1:nrow(law), nboot = 1000, theta = calc_rho, law)
mean.thetastar <- mean(boot$thetastar)
rho_lb <- quantile(boot$thetastar, 0.025)
rho_ub <- quantile(boot$thetastar, 0.975)

hist(boot$thetastar)
abline(v=law_r, col = "red")
```

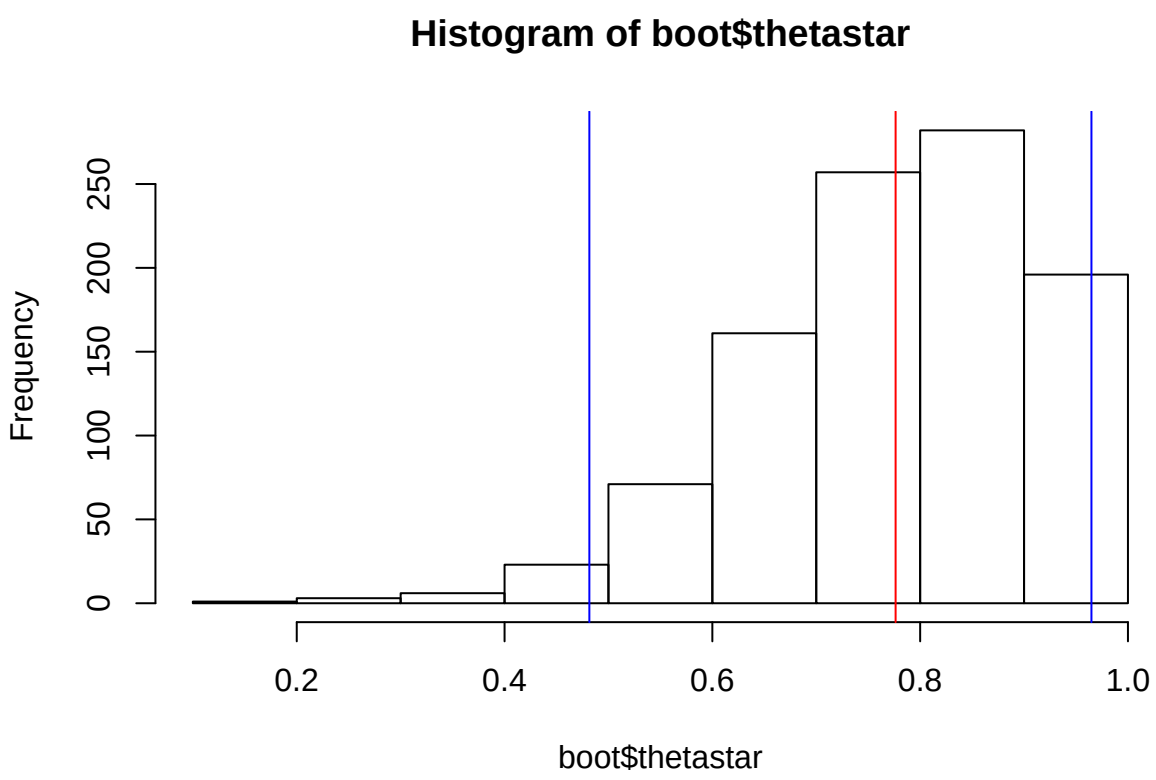


3c)

Calculate the bootstrap percentile confidence interval for correlation and insert blue vertical lines in the histogram from part (b) at the upper and lower limits. Based on this, could we reject the null hypothesis that the true correlation is equal to 0.5?

Solution:

```r
hist(boot$thetastar)
abline(v=law_r, col = "red")
abline(v=rho_lb, col = "blue")
abline(v=rho_ub, col = "blue")
```



No, we would fail to reject the null, since a correaltion of 0.5 lies within the 95% confidence interval [0.482, 0.965]. As such, we are 95% confident that r = 0.5 belongs to the distribution and is a feasible value for the true correlation.

3d)

Calculate the bootstrap estimate of bias as well as the (standard) bias corrected bootstrap percentile confidence interval. Insert additional green vertical lines in the histogram showing the bounds for this new interval. According to this, could we reject the null hypothesis that the true correlation is equal to 0.5?

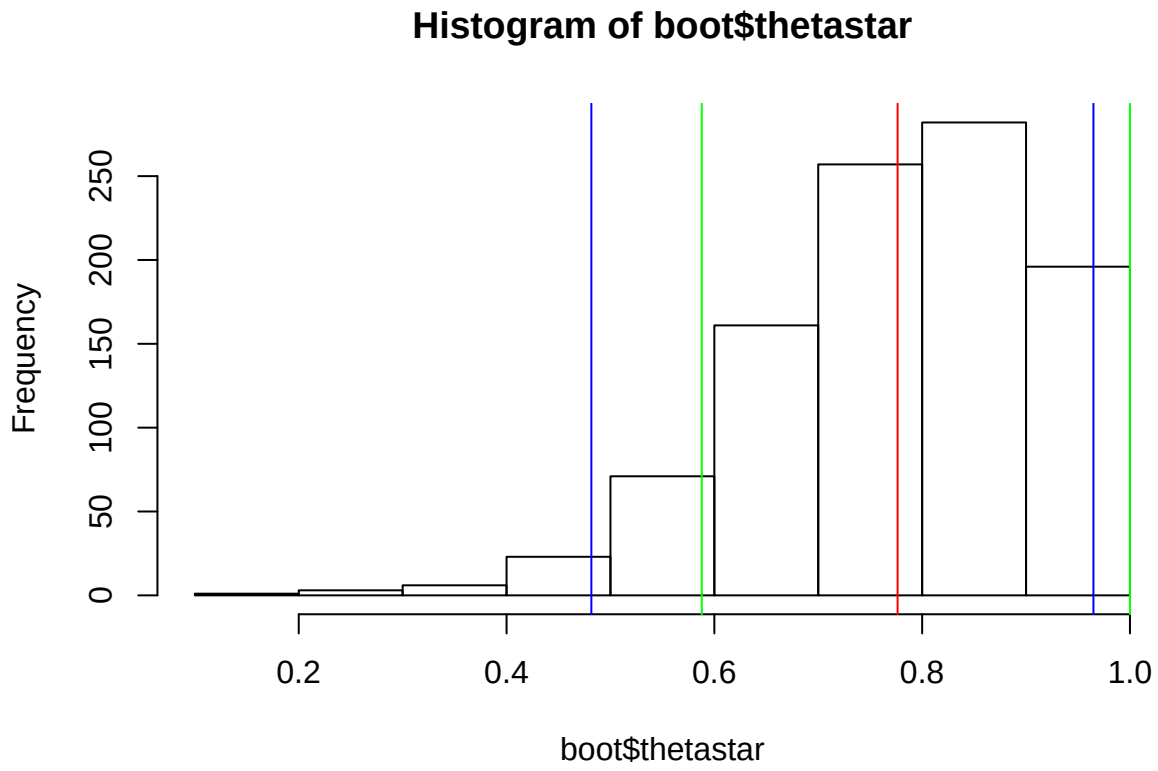
Solution:

```
rho_corrected_lb <- (2*law_r - quantile(boot$thetastar, 0.975))
rho_corrected_ub <- min(2*law_r - quantile(boot$thetastar, 0.025),1)
hist(boot$thetastar)
rho_corrected_lb
```

```
##     97.5%
## 0.5878914
```

```
abline(v=law_r, col="red")
abline(v=rho_ub, col="blue")
```

```r
abline(v=rho_lb, col="blue")
abline(v=rho_corrected_lb, col="green")
abline(v=rho_corrected_ub, col="green")
```



Yes, we can reject the null. The new 95% confidence interval is [0.589, 1.0]. Since 0.5 is not contained within the confidence interval, r = 0.5 is not a feasible value for the true correlation.

3e)

Based on these confidence intervals, you should see strong evidence that the true correlation is not equal to 0. Design a permutation test to explicitly test this.

Solution:

```r
perc <- ecdf(boot$thetastar)
perc(0.0)
```

```
## [1] 0
```

Our permutation test will be setup similar to a hypothesis test to determine if there is a correlation:

Then the null hypothesis is: $H_o : \rho = 0$

and the alternative hypothesis is: $H_a : \rho \neq 0$

Let the test statistic be the the hypothesized correlation, 0:

$t^* = \rho_0 = 0$

To find the p-value of the estimate, we would calculate the percentile of the null hypothesis estimate, being, t^*. As seen in the code above, the percentile of a correlation being 0, indicates that the p-value < 0.001. Thus, the p-value is statistically significant and deviates from the distribution by a significant amount, which allows us to reject H_o, and conclude that 0 is not a feasible estimate of the true correlation.

Problem 4)

4a)

Generate a training dataset with 50 observations on two features sampled from a (continuous) standard uniform distribution. Add a response to the data frame of the form $Y = X_1 + X_2 + \varepsilon$ where $\varepsilon \sim N(0, 0.25^2)$.

Solution:

```
set.seed(1)
train.x1 <- runif(50)
train.x2 <- runif(50)
train.eps <- rnorm(50, mean = 0, sd = 0.25)
train.y <- train.x1 + train.x2 + train.eps
train.df <- data.frame(train.y, train.x1, train.x2)
```

4b)

Generate another test dataset with 30 observations using the same setup. Use the training data from part (a) to construct a linear model with X1 and X2 – no interaction terms or higher-order polynomial terms. Calculate the MSE on the test set created here and call this MSE_0.

Solution:

```
test.x1 <- runif(30)
test.x2 <- runif(30)
test.eps <- rnorm(30, mean = 0, sd = 0.25)
test.y <- test.x1 + test.x2 + test.eps
test.df <- data.frame(test.y, test.x1, test.x2)
names(test.df) <- c("train.y", "train.x1", "train.x2")
model <- lm(train.y ~ train.x1 + train.x2, data = train.df)
summary(model)
```

```
##
## Call:
## lm(formula = train.y ~ train.x1 + train.x2, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51877 -0.16512  0.03754  0.11099  0.61852
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.11193    0.09695  -1.154    0.254
## train.x1     1.21371    0.12582   9.647 1.01e-12 ***
## train.x2     1.05454    0.12936   8.152 1.51e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2394 on 47 degrees of freedom
## Multiple R-squared:  0.7819, Adjusted R-squared:  0.7726
## F-statistic: 84.26 on 2 and 47 DF,  p-value: 2.866e-16
```

```
mse <- function(x, yTrue, yPred){
  mean((yTrue[x] - yPred[x])^2)
}

pred <- predict(model, test.df)
mse_0 <- mse(1:length(test.y), test.y, pred)
mse_0
```

```
## [1] 0.08869928
```

The MSE on the test set, MSE_0 is 0.08870.

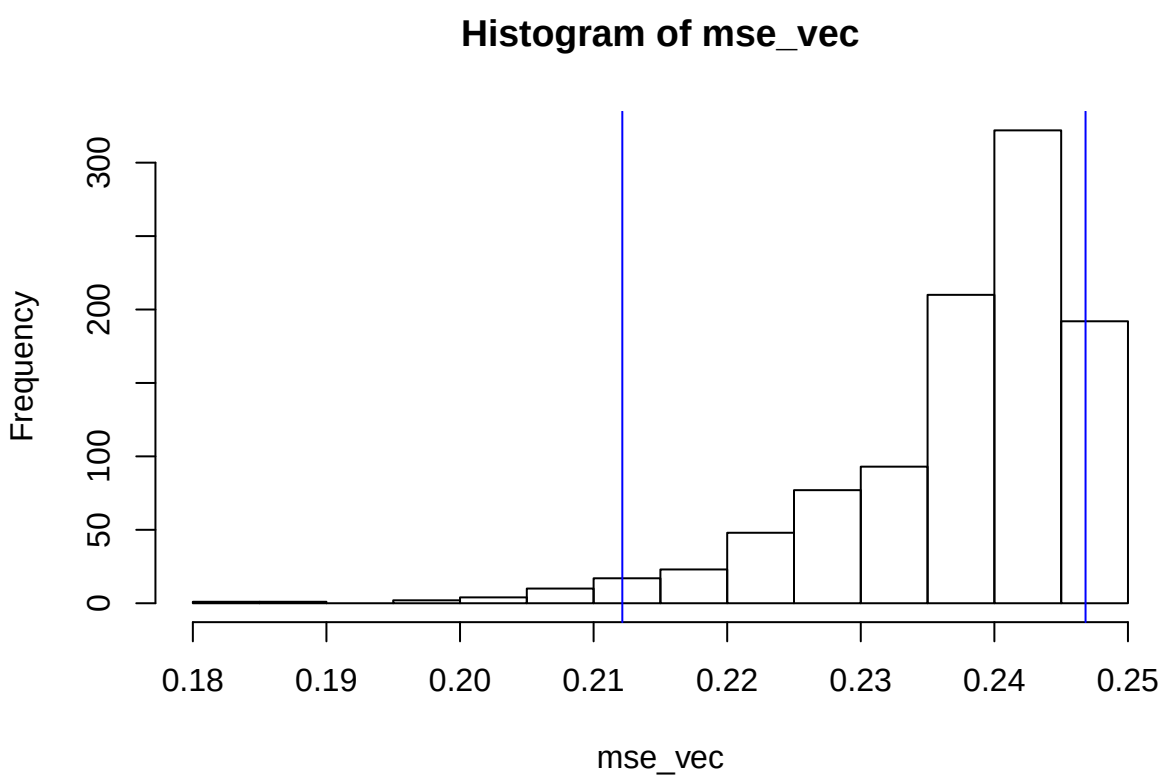
4c)

Recall that an (overall) F-test tests whether any of the features are significant. Using the test MSE as our statistic, devise a permutation-test-equivalent to the overall F-test. Carry out the test with 1000 permutations. Can you reject the null hypothesis that none of the predictors are significant using your test?

Solution:

```
mse_vec <- c()
rand = train.df
# train.df
for(i in 1:1000){
  rand[] <- lapply(train.df, sample)
  # rand <- train.df[1:sample(,]
  model_i <- lm(train.y ~ train.x1 + train.x2, data = rand)
  pred_i <- predict(model_i, rand)
  rownames(rand) <- 1:nrow(rand)
  mse_vec[i] <- mse(1:length(rand$train.y), rand$train.y, pred_i)
}
# mse_vec

hist(mse_vec)
mse_lb = quantile(mse_vec, 0.025)
mse_ub = quantile(mse_vec, 0.975)
abline(v=mse_0, col = "red")
abline(v=mse_lb, col = "blue")
abline(v=mse_ub, col = "blue")
```




```
mse_lb
```

```
##      2.5%
## 0.2121486
```

```
mse_ub
```

```
##     97.5%
## 0.2468251
```

Our permutation test will be setup similar to a hypothesis test to determine if all predictors jointly are significant in predicting the response:

Then the null hypothesis is: $H_o : \beta_1 = \beta_2 = 0$

and the alternative hypothesis is: $H_a : \beta_1 \neq 0$ or $\beta_2 \neq 0$ or both $\beta_1 \neq 0, \beta_2 \neq 0$

After performing 1000 permuations, based on the above hisogram, our confidence interval is [0.2107, 0.2468]. The MLE estimate, 0.089 lies outside the 95% confidence interval. So, we reject the null hypothesis and conclude that x_1 and x_2 are jointly significant in predicting `y`.

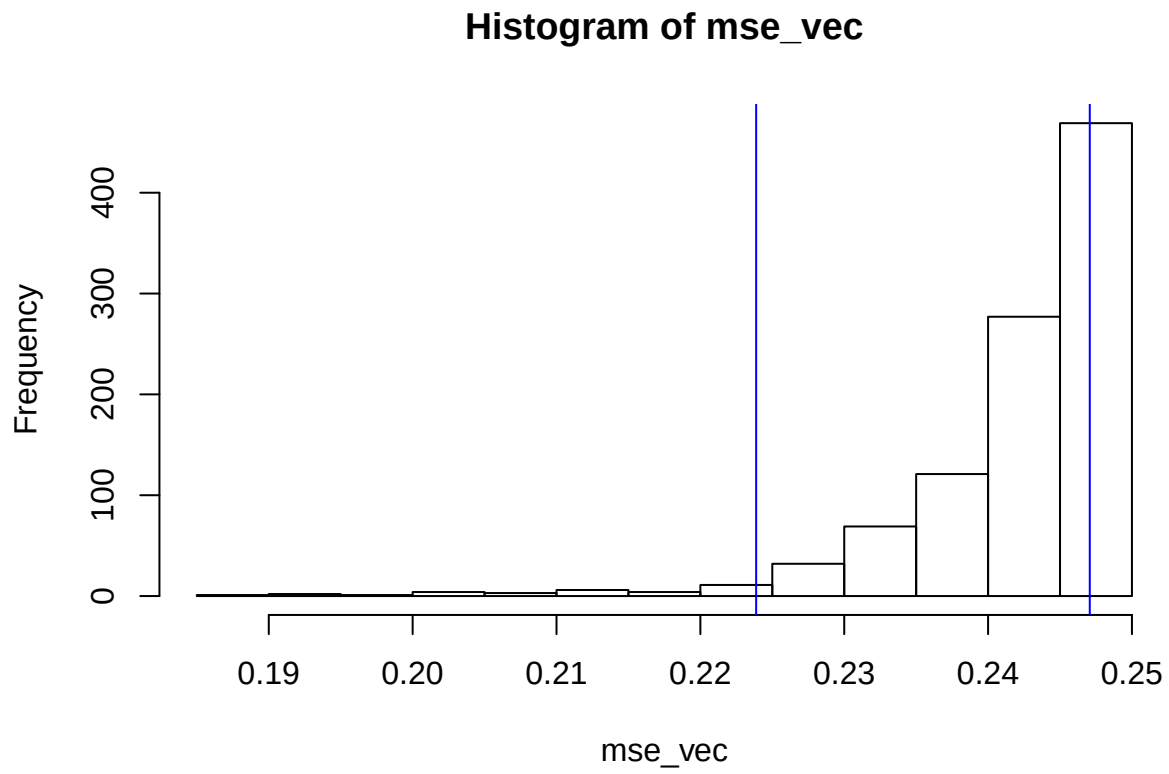
4d)

As we know, we can examine whether an individual feature is significant by looking at its corresponding t-test. Again using test MSE as the statistic of interest, devise a permutation-test-equivalent to the individual t-test. Carry out the test with 1000 permutations to test whether X_2 is significant. Can you reject the null hypothesis that $\beta_2 = 0$using your test?

Solution:

```
mse_vec <- c()
rand = train.df
# train.df
for(i in 1:1000){
  rand[] <- lapply(train.df, sample)
  # rand <- train.df[1:sample(,]
  model_i <- lm(train.y ~ train.x2, data = rand)
  pred_i <- predict(model_i, rand)
  rownames(rand) <- 1:nrow(rand)
  mse_vec[i] <- mse(1:length(rand$train.y), rand$train.y, pred_i)
}
# mse_vec

hist(mse_vec)
mse_lb = quantile(mse_vec, 0.025)
mse_ub = quantile(mse_vec, 0.975)
abline(v=mse_0, col = "red")
abline(v=mse_lb, col = "blue")
abline(v=mse_ub, col = "blue")
```




```
mse_lb
```

```
##      2.5%
```

```
## 0.2238784
```

```
mse_ub
```

```
##     97.5%
## 0.2470677
```

Our permutation test will be setup similar to a Partial F-test:

Then the null hypothesis is: $H_o : \beta_2 = 0$

and the alternative hypothesis is: $H_a: $ $\beta_2 \neq 0$

After performing 1000 permuations, based on the above hisogram, our confidence interval is [0.2249, 0.2471]. The MLE estimate, 0.089 lies outside the 95% confidence interval. So, we reject the null hypothesis and conclude that x_2 is significant in predicting `y`.

4e)

In this case, we have only two features so an individual t-test is equivalent to a partial F-test. Let's scale things up a bit. Using the same general procedure and model as above, create a training set with 500 observations on 10 features. Also create a test set with 50 observations.

Solution:

```
set.seed(1)
trainf.x <- matrix(0, nrow = 500, ncol = 10)
trainf.eps <- matrix(0, nrow = 500, ncol = 10)
for(i in 1:500){
  trainf.x[i, 1:10] = runif(10)
  trainf.eps[i, 1:10] = rnorm(10, mean = 0, sd = 0.25)
}
trainf.y <- trainf.x + trainf.eps
trainf.y <- apply(trainf.y, 1, sum)
trainf.df = data.frame(trainf.y, trainf.x)
names(trainf.df) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")

testf.x <- matrix(0, nrow = 50, ncol = 10)
testf.eps <- matrix(0, nrow = 50, ncol = 10)
for(i in 1:50){
  testf.x[i, 1:10] = runif(10)
  testf.eps[i, 1:10] = rnorm(10, mean = 0, sd = 0.25)
}
testf.y <- testf.x + testf.eps
testf.y <- apply(testf.y, 1, sum)
testf.df = data.frame(testf.y, testf.x)
names(testf.df) <- c("y", "x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")
model_full <- lm(y ~ ., data = trainf.df)
summary(model)
```

```
##
## Call:
```

```
## lm(formula = train.y ~ train.x1 + train.x2, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.51877 -0.16512  0.03754  0.11099  0.61852
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.11193    0.09695  -1.154    0.254
## train.x1     1.21371    0.12582   9.647 1.01e-12 ***
## train.x2     1.05454    0.12936   8.152 1.51e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2394 on 47 degrees of freedom
## Multiple R-squared:  0.7819, Adjusted R-squared:  0.7726
## F-statistic: 84.26 on 2 and 47 DF,  p-value: 2.866e-16
```

```
mse <- function(x, yTrue, yPred){
  mean((yTrue[x] - yPred[x])^2)
}

pred_full <- predict(model_full, testf.df)
mse_full <- mse(1:length(testf.df$y), testf.df$y, pred_full)
```

4f)

Using the data from part (d) and using test MSE as the statistic of interest, devise a permutation-test-equivalent to the partial F-test that will evaluate whether any of thefeatures X_8, X_9 or X_{10} are significant. Carry out the test with 1000 permutations. Can you reject the null hypothesis?

Solution:

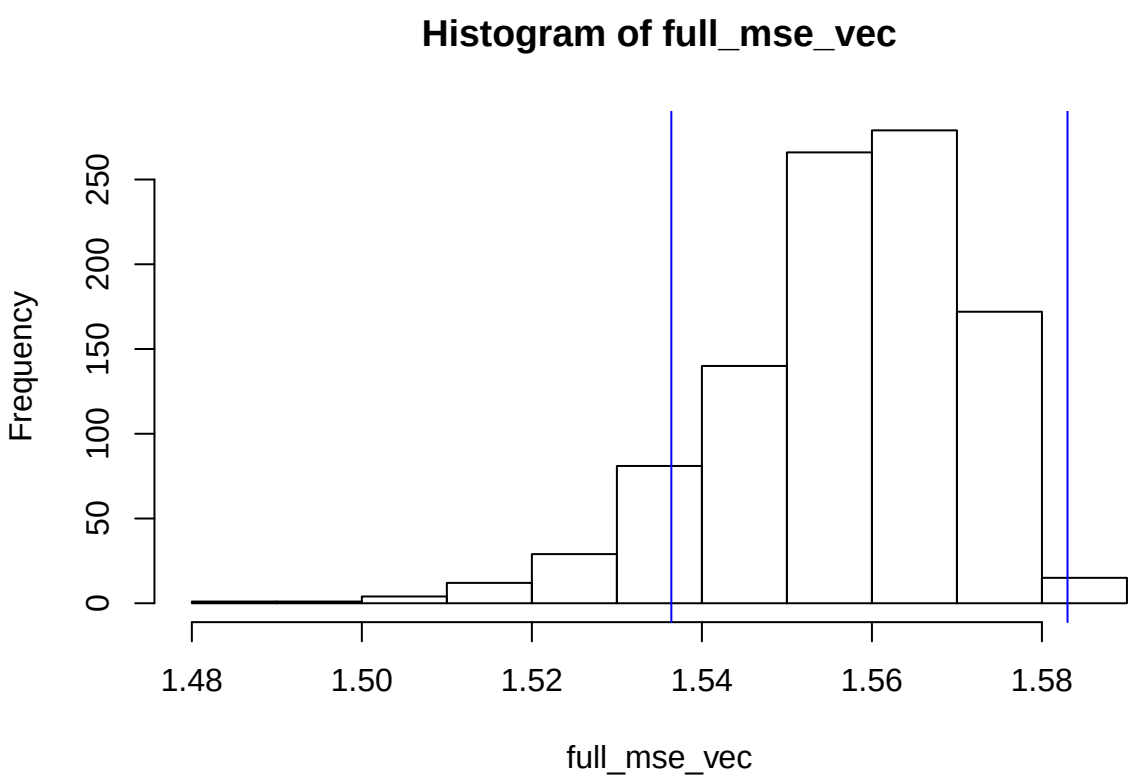
```
model_partial <- lm(y ~ . - x8 - x9 - x10, data = trainf.df)
summary(model_partial)
```

```
##
## Call:
## lm(formula = y ~ . - x8 - x9 - x10, data = trainf.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73255 -0.62282  0.04326  0.63493  2.83569
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2612     0.1965   6.419 3.23e-10 ***
## x1            1.1034     0.1461   7.551 2.12e-13 ***
## x2            1.0683     0.1506   7.093 4.59e-12 ***
## x3            1.0396     0.1517   6.851 2.20e-11 ***
## x4            1.2559     0.1493   8.410 4.48e-16 ***
```

```
## x5            1.0820     0.1447   7.476 3.55e-13 ***
## x6            0.9277     0.1509   6.147 1.64e-09 ***
## x7            0.9459     0.1469   6.440 2.85e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.954 on 492 degrees of freedom
## Multiple R-squared:  0.4364, Adjusted R-squared:  0.4284
## F-statistic: 54.43 on 7 and 492 DF,  p-value: < 2.2e-16
```

```
mse <- function(x, yTrue, yPred){
  mean((yTrue[x] - yPred[x])^2)
}
pred_partial <- predict(model_partial, testf.df)
mse_partial <- mse(1:length(testf.df$y), testf.df$y,pred_partial)
full_mse_vec <- c()
partial_mse_vec <- c()
rand = trainf.df
# train.df
for(i in 1:1000){
  rand[] <- lapply(trainf.df, sample)
  # rand <- train.df[1:sample(,]
  full_model_i <- lm(y ~ ., data = rand)
  partial_model_i <- lm(y ~ . - x8 - x9 - x10, data = rand)

  pred_full_i <- predict(full_model_i, rand)
  pred_partial_i <- predict(partial_model_i, rand)
  rownames(rand) <- 1:nrow(rand)
  full_mse_vec[i] <- mse(1:length(rand$y), rand$y, pred_full_i)
  partial_mse_vec[i] <- mse(1:length(rand$y), rand$y, pred_partial_i)
}
hist(full_mse_vec)
mse_lb = quantile(partial_mse_vec, 0.025)
mse_ub = quantile(partial_mse_vec, 0.975)
# abline(v=mse_0, col = "red")
abline(v=mse_lb, col = "blue")
abline(v=mse_ub, col = "blue")
```



In this partial-F test, we want to examine if any of the features X_8, X_9, X_{10} are significant predictors of F. We have 1 reponse, y, and 10 predictors, $x_1 \dots x_{10}$. The full model is represented as $y = \beta_0 + \beta_1 X_1 + \dots \beta_{10} X_{10}$. The reduced model is represented as $y = \beta_0 + \beta_1 X_1 + \dots \beta_7 X_7$. As such, we perform a permutation test to examine whether the MSE of the reduced model is significantly lower than the MSE of the full model. If the reduced MSE is outside the 95% confidence interval of the permutation test estimate, then we can conclude the reduced model is a significant predictor of y. We proceed as follows:

$$H_o : \beta_8 = \beta_9 = \beta_{10} = 0$$

$$H_a : \exists\, i \in [8, 9, 10]\ \beta_i \neq 0$$

After forming the reduced model, we reach $MSE_{reduced} = 0.5531$. The full model's MSE produces $MSE_{full} = 0.3902$. The blue lines represent the 95% confidence interval of the estimate, which is [.268, 520]. Since $MSE_{reduced} \notin [1.53921, 1.583564]$, we reject H_o and conclude the reduced model, is significant in predicting y. So X_8, X_9, X_{10} are significant predictors.

Problem 5)

5a)

All else being equal, think about how you would conduct a standard parametric hypothesis test to evaluate whether the vaccine was effective. What is the parameter you're interested in? What are the null and alternative hypotheses? Carry out the test.

Solution:

The parameter of interest is the proportion of individuals who are infected with COVID-19. In particular, we are interested in the difference between individuals who received the placebo and have COVID-19 ant the individuals who received the vaccine and still got COVID-19.

Let p_1 represent the proportion of individuals who received the placebo and got COVID-19.

Let p_2 represent the proportion of individuals who received the vaccine and got COVID-19.

Then the null hypothesis is: $H_o : p_1 - p_2 = 0$

and the alternative hypothesis is: $H_a : p_1 - p_2 \neq 0$

So, we can perform a difference in proportions Z-test, as so:

```
set.seed(1)
val <- prop.test(x = c(162, 8), n = c(43000/2, 43000/2))
val
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(162, 8) out of c(43000/2, 43000/2)
## X-squared = 138.25, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  0.005931969 0.008393612
## sample estimates:
##      prop 1      prop 2
## 0.007534884 0.000372093
```

Since p < 0.01, we have sufficient evidence to conclude that the proportion of individuals who received the placebo and got COVID-19 (p_1) and the proportion of individuals who received the vaccine and got COVID-19 (p_2) are significantly different.

5b)

Now let's evaluate the claim using a randomization test – a close friend of the per-mutation test we discussed in class. To begin, imagine we have two groups of enrollees inthe trial. If the vaccine was not effective, what should we expect to see in the data?

Solution:

If the vaccine was not effective, then we would expect that the proportion of individuals who received the placebo and got COVID-19 (p_1) and the proportion of individuals who received the vaccine and got COVID-19 (p_2) to not differ significantly. In other words, we would conclude that $p_1 - p_2 = 0$.

5c)

Given what you expect to see in the data, how could you measure whether that is in fact what you see. In other words, what's a reasonable test statistic?

Solution:

A reasonable test statistic would be the difference of the vaccine infection rate and the placebo infection rate. In other words,

$$t = p_1 - p_2$$

would be a reasonable test statistic.

5d)

In a classical two-group permutation test, we carry out the test by randomly shufflingthe group labels. In a randomization test, we randomly assign particular kinds of labelsthat we're interested in (in this case, the enrollees who happen to contract covid). Workout how to perform such a test in this framework based on your Solutions to (b) and (c)above and carry it out with 1000 randomizations. Plot a histogram of the test statistic values under the null (i.e. for each randomization) and overlay a line corresponding to the test statistic value calculated on the original data. What is your p-value?

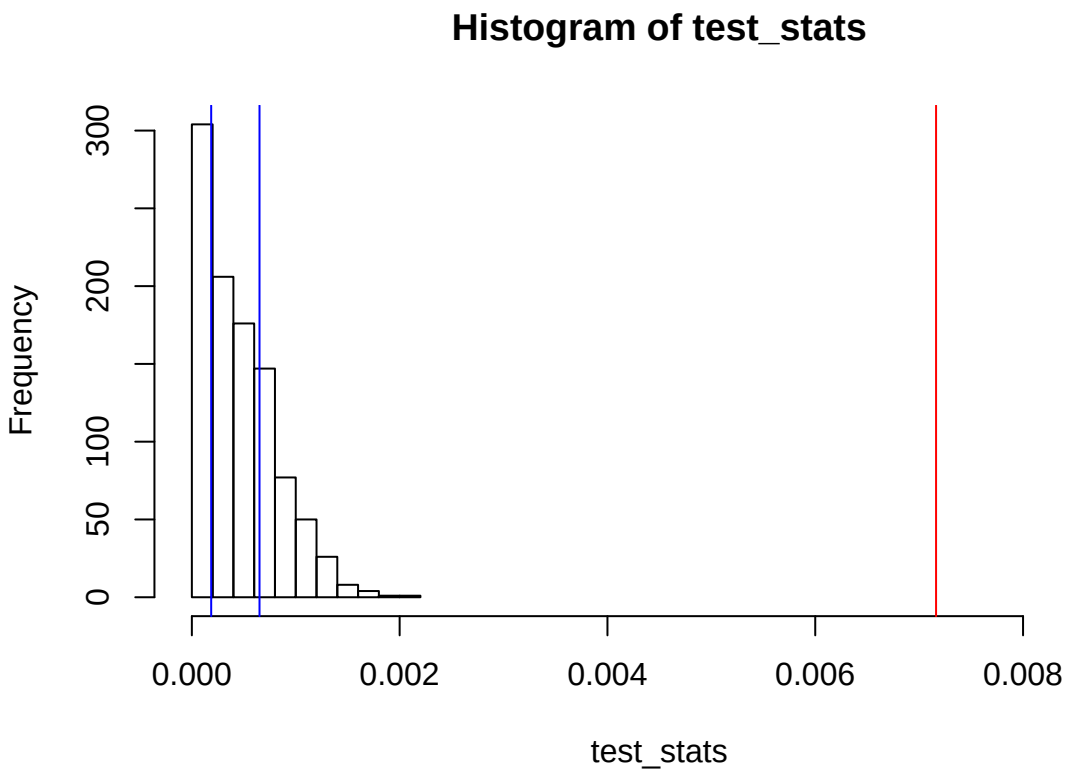
Solution:

```
set.seed(1)
test_stats <- c()

vaccine <- rep("Vaccine", 43000*0.5)
placebo <- rep("Placebo", 43000*0.5)
has_covid_vaccine <- c(rep(1, 8), rep(0, 43000/2 - 8))
has_covid_placebo <- c(rep(1, 162), rep(0, 43000/2 - 162))
colA <- c(vaccine, placebo)
colB <- c(has_covid_vaccine, has_covid_placebo)
df <- data.frame(colA, colB)
colnames(df) <- c("Group", "COVID")
for(i in 1:1000){
  perm.label <- sample(df$COVID, replace = F)
  rand <- data.frame(df$Group, perm.label)
  colnames(rand) <- c("Group", "COVID")
  test_stat = abs(nrow(subset(rand, rand$Group == "Vaccine" & rand$COVID == 1))/21500
                  - nrow(subset(rand, rand$Group == "Placebo" & rand$COVID == 1))/21500)
  test_stats[i] = test_stat
}
t_0 <- abs(8/(21500) - 162/(21500))
p <- mean(test_stats > abs(8/(21500) - 162/(21500)))
hist(test_stats, xlim = c(0, 0.009))
quant_tstat = rep(0, 2)
t_0
```

```
## [1] 0.007162791
```

```
quant_tstat[1] = quantile(test_stats)[2]
quant_tstat[2] = quantile(test_stats)[4]
abline(v = t_0, col = "red")
abline(v = quant_tstat[1], col = "blue")
abline(v = quant_tstat[2], col = "blue")
```



The p-value is `0`, so we would reject the null hypothesis and conclude that we have sufficient evidence to conclude that the proportion of individuals who received the placebo and got COVID-19 (p_1) and the proportion of individuals who received the vaccine and got COVID-19 (p_2) are significantly different.