Homework 4

Rohan Krishnan

2024-02-26

Problem 1

No submission required.

Problem 2

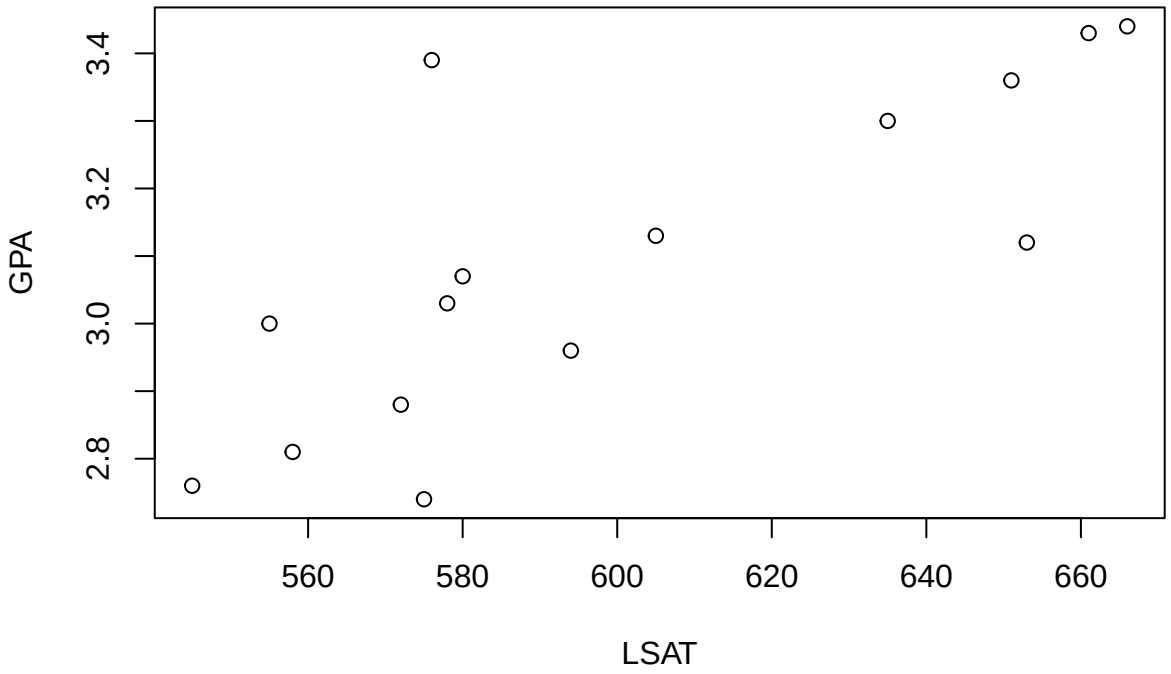
ISLR Chapter 5 Conceptual Exercise 4

We can use bootstrapping to re-sample X and Y and fit the model multiple times. Then, we can generate a distribution of standard deviations over each bootstrapped sample and fit a standard bootstrapped 95% CI to come up with a range of estimates or calculate a bias-corrected point estimate of the standard deviation across the bootstrapped samples.

Problem 3

(a)

```
#Install and load bootstrap package
#install.packages("bootstrap")
library(bootstrap)
#Plot law data
data(law)
plot(law)
```

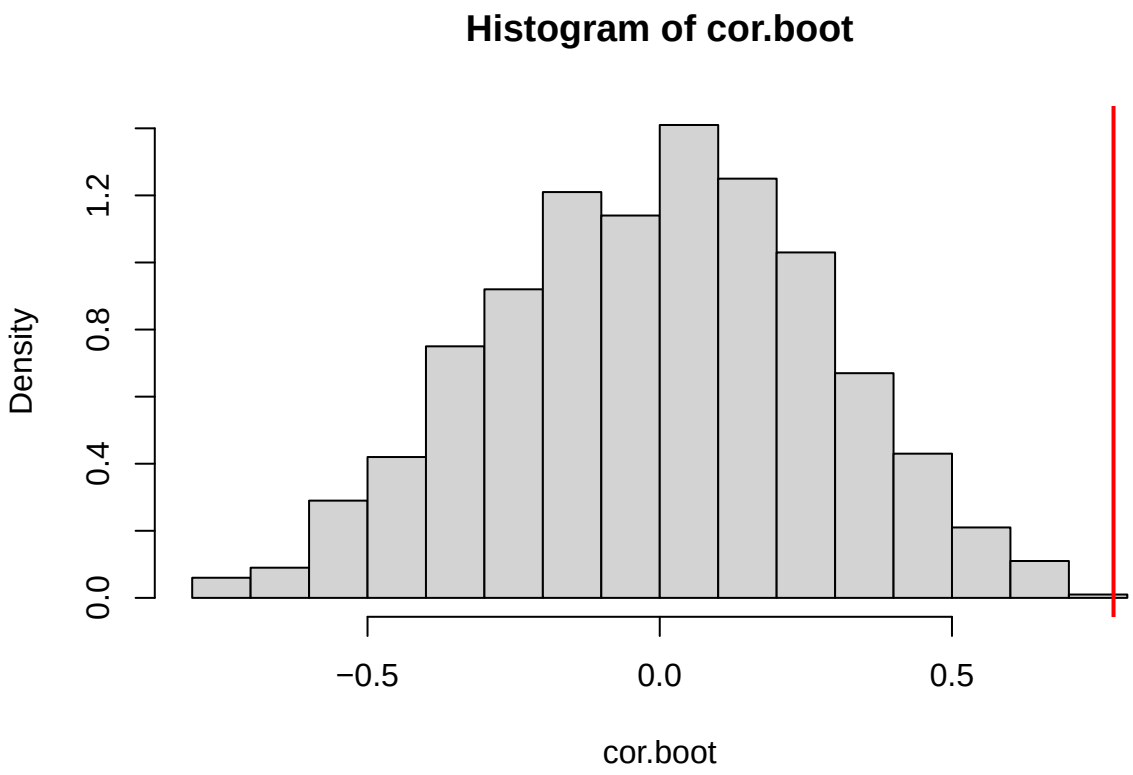


There appears to be a moderate to strong positive correlation between LSAT score and GPA.

```
#Calculate correlation
original.cor <- cor(law$LSAT, law$GPA)
```

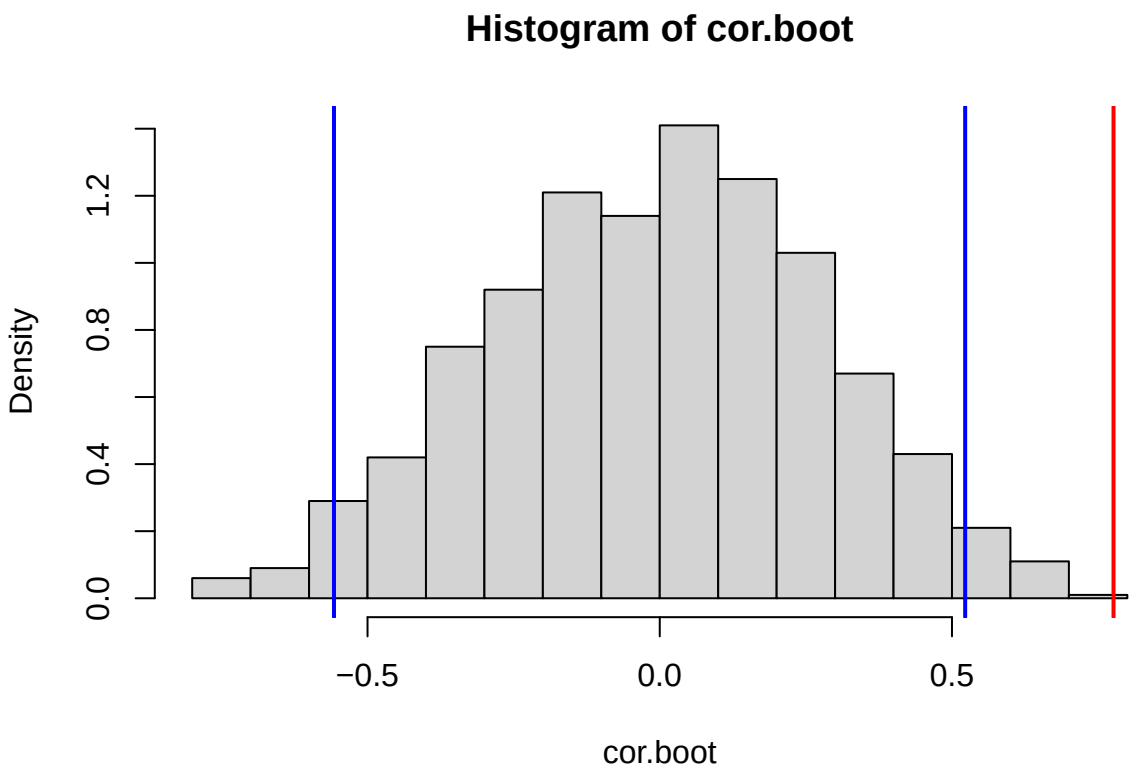
(b)

```
#Define number of bootstrap samples and vector of nBoot length to store bootstrap estimates
nBoot <- 1000
cor.boot <- rep(0,nBoot)
#Generate bootstrap samples
set.seed(100)
for (i in 1:nBoot) {
    LSATperm <- sample(law$LSAT,replace=T)
    GPAperm <- sample(law$GPA, replace = T)
    cor.boot[i] <- cor(LSATperm, GPAperm)
}
#Create histogram of bootstrap samples and insert line at original correlation
hist(cor.boot,freq=FALSE,breaks=20)
abline(v=cor(law$LSAT,law$GPA),col='red',lwd=2)
```



(c)

```
#Find 95 percentile estimate of bootstrap sample
ci.boot <- c(quantile(cor.boot, 0.025),quantile(cor.boot, 0.975))
#Add interval to histogram
hist(cor.boot,freq=FALSE,breaks=20)
abline(v=cor(law$LSAT,law$GPA),col='red',lwd=2)
abline(v = ci.boot[1], col = "blue", lwd = 2)
abline(v = ci.boot[2], col = "blue", lwd = 2)
```

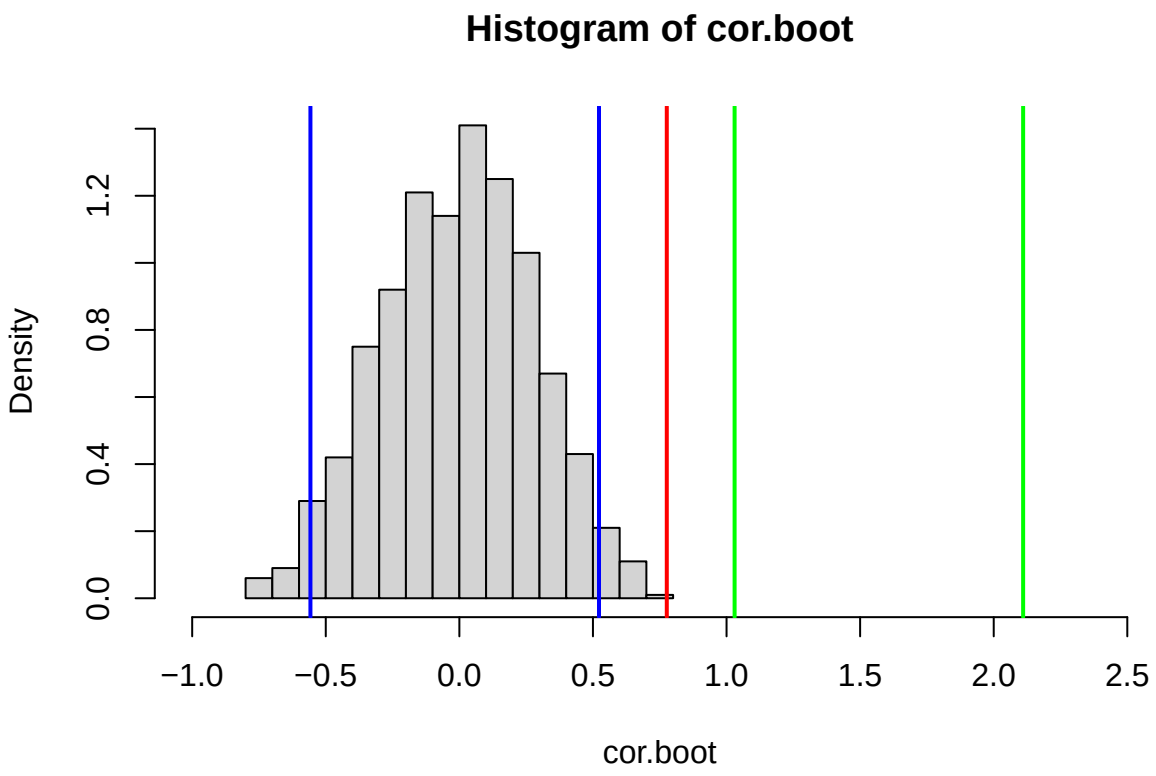


Based on the above histogram and the 95% confidence interval of [–0.557, 0.523], we cannot reject the null hypothesis that the true correlation is equal to 0.50.

(d)

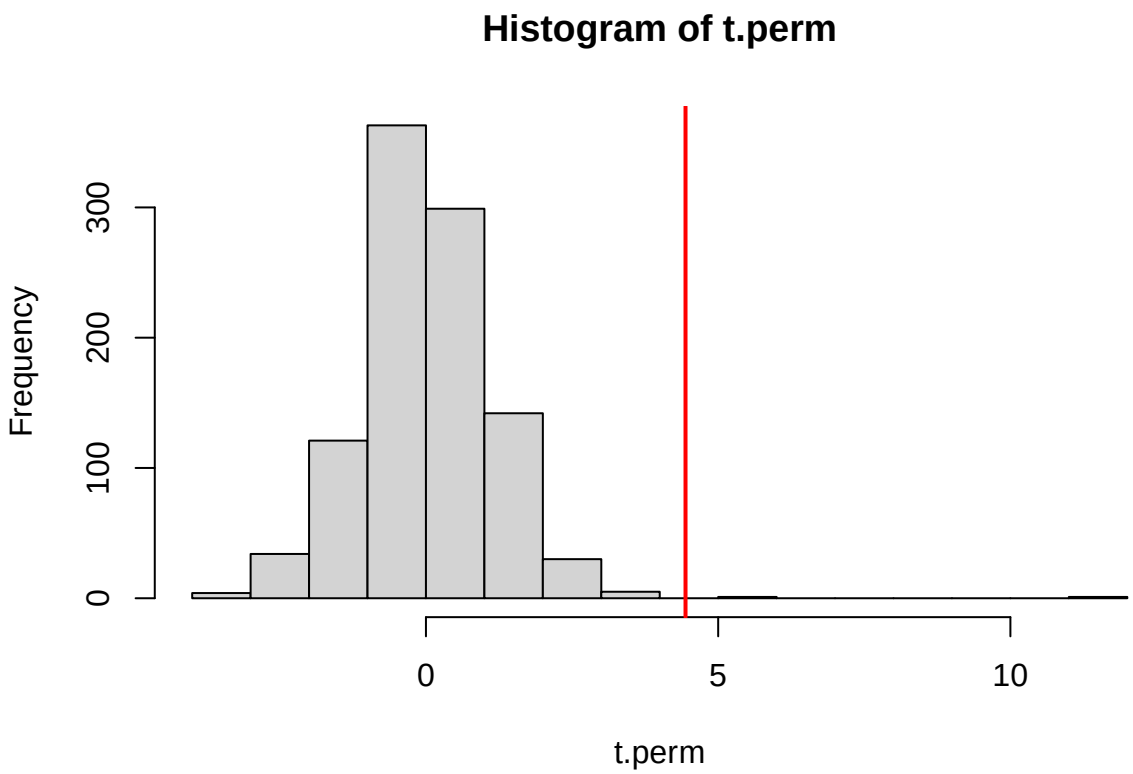
```
#Calculate bootstrap estimate of bias
mean.boot <- mean(cor.boot)
#Calculate bias
bias.boot <- mean.boot - original.cor
#Calculate bias-corrected confidence interval
standard.ci.boot <- c((2*original.cor - quantile(cor.boot, 0.975)),
                      (2*original.cor - quantile(cor.boot, 0.025)))

hist(cor.boot,freq=FALSE,breaks=20, xlim =c(-1.0, 2.5))
abline(v=cor(law$LSAT,law$GPA),col='red',lwd=2)
abline(v = ci.boot[1], col = "blue", lwd = 2)
abline(v = ci.boot[2], col = "blue", lwd = 2)
abline(v = standard.ci.boot[1], col = "green", lwd = 2)
abline(v = standard.ci.boot[2], col = "green", lwd = 2)
```



Based on the above histogram, we can safely reject the null hypothesis that the true correlation is equal to 0.50 as the bias-corrected confidence interval (green) does not contain 0.50. **(e)**

```
#Create two groups and test for significance
Group1 <- law$LSAT
Group2 <- law$GPA
t0 <- cor.test(Group1, Group2)$statistic
#Perform 1000 permutations and recover t-statistics
nperm <- 1000
t.perm <- rep(0,nperm)
set.seed(100)
for(i in 1:nperm){
  Group2 <- sample(Group2)
  t.perm[i] <- cor.test(Group1, Group2)$statistic
}
#Graph t-statistics with red line at t0
hist(t.perm, breaks = 20)
abline(v = t0, col = "red", lwd = 2)
```




```
#Calculate explicit p-value
p.3e <- mean(t.perm > t0); p.3e
```

```
## [1] 0.002
```

Our original t-statistic falls quite far outside of the distribution permuted t-statistics, indicating that there is strong evidence that the true correlation is not equal to 0. The p-value is 0.002, which means that very few observations fall beyond our original t-statistic.

Problem 4

(a)

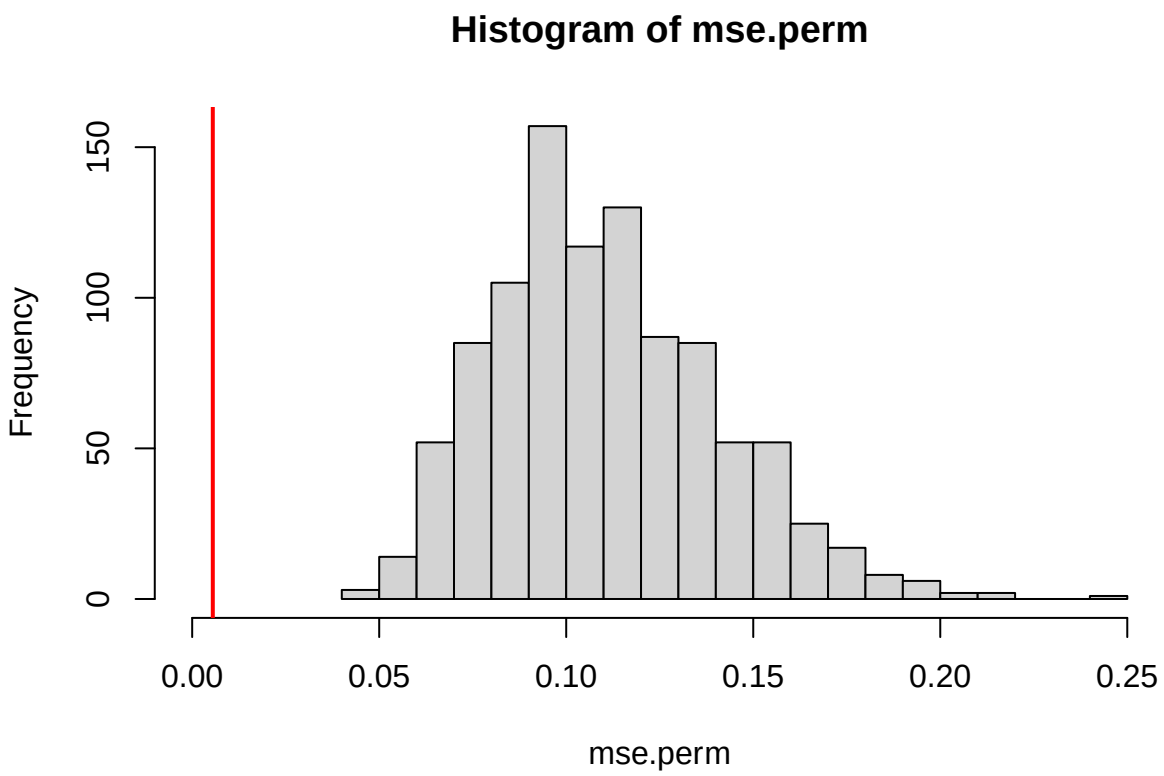
```
#Generate train data set
set.seed(100)
x1 <- runif(50)
x2 <- runif(50)
y <- x1 + x2 + rnorm(50, 0,0.25^2)
train <- data.frame(y, x1, x2)
```

(b)

```
#Generate test data set
set.seed(100)
test <- data.frame(x1 = runif(30),
                   x2 = runif(30))
test$y <- test$x1 + test$x2 + rnorm(30, 0 , 0.25^2)
test <- test[,c(3,1,2)]
#Create linear model
lin.mod <- lm(y ~ x1 + x2, data = train)
#Calculate test MSE
mse0 <- mean((test$y - predict(lin.mod, test))^2)
```

(c)

```
#Perform 1000 permutations and generate joint-F-test MSE statistics
nperm <- 1000
mse.perm <- rep(0, nperm)
set.seed(100)
for(i in 1:nperm){
  perm.train <- train
  perm.train$x1 <- sample(train$x1)
  perm.train$x2 <- sample(train$x2)
  lin.mod.perm <- lm(y ~ x1 + x2, data = perm.train)
  #perm.test <- test
  #perm.test$x1 <- sample(test$x1)
  #perm.test$x2 <- sample(test$x2)
  mse.perm[i] <- mean((test$y - predict(lin.mod.perm, test))^2)
}
#Create histogram with mse0 as a red line
hist(mse.perm, breaks = 20, xlim = c(0,0.25))
abline(v = mse0, col = "red", lwd =2)
```




```
#Calculate p-value
p.4c <- mean(mse.perm < mse0); p.4c
```

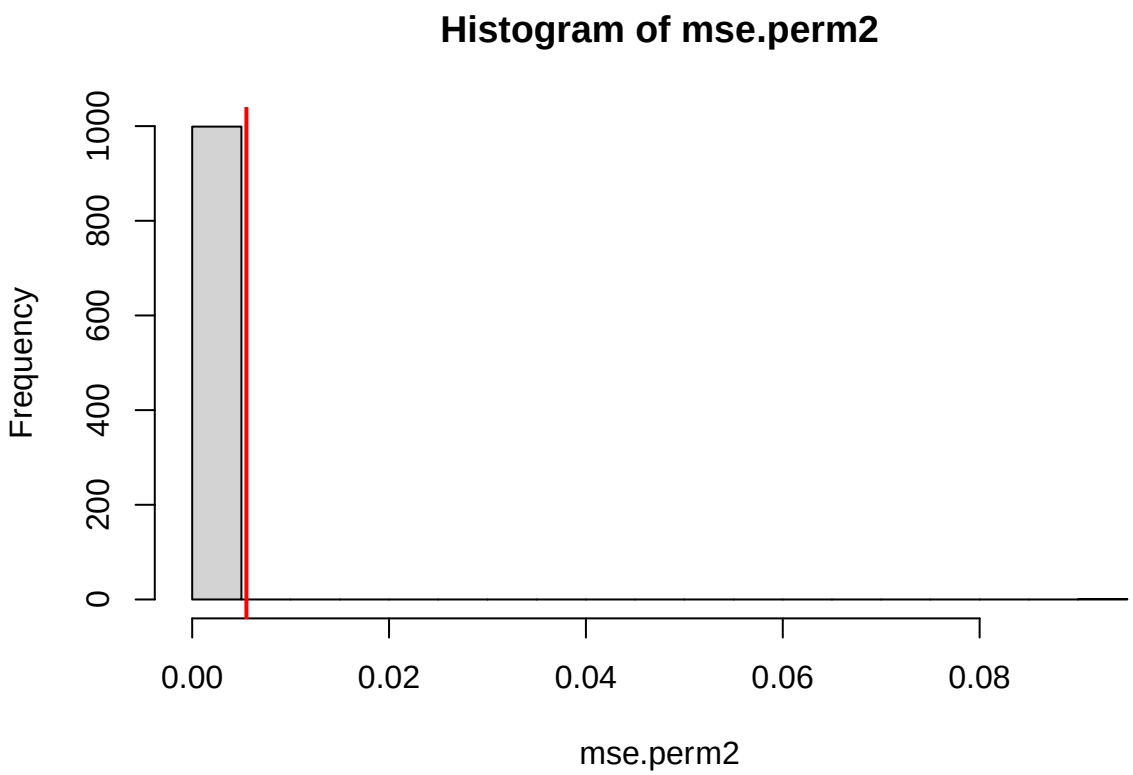
```
## [1] 0
```

From the above histogram and p-value, we can clearly reject the null hypothesis that none of the predictors are significant as our original MSE falls completely to the left of our permuted MSEs and has a p-value of 0.

(d)

```
#Perform 1000 permutations and generate individual-t-test MSE statistics
nperm <- 1000
mse.perm2 <- rep(0, nperm)
```

```r
set.seed(100)
for(i in nperm){
  perm.train <- train
  perm.train$x2 <- sample(train$x2)
  lin.mod.perm2 <- lm(y ~ x1 + x2, data = perm.train)
  mse.perm2[i] <- mean((test$y - predict(lin.mod.perm2, test))^2)
}
#Create histogram with mse0 as a red line
hist(mse.perm2, breaks = 20)
abline(v = mse0, col = "red", lwd = 2)
```




```r
#Calculate p-value
p.4d <- mean(mse.perm2<mse0); p.4d
```

```
## [1] 0.999
```

I cannot figure out what I did wrong for this question. I keep getting a large p-value and heavily left skewed distribution. Intuitively, from the DGP, we know that this test should output a small p-value and indicate that X2 is a significant predictor.

(e)

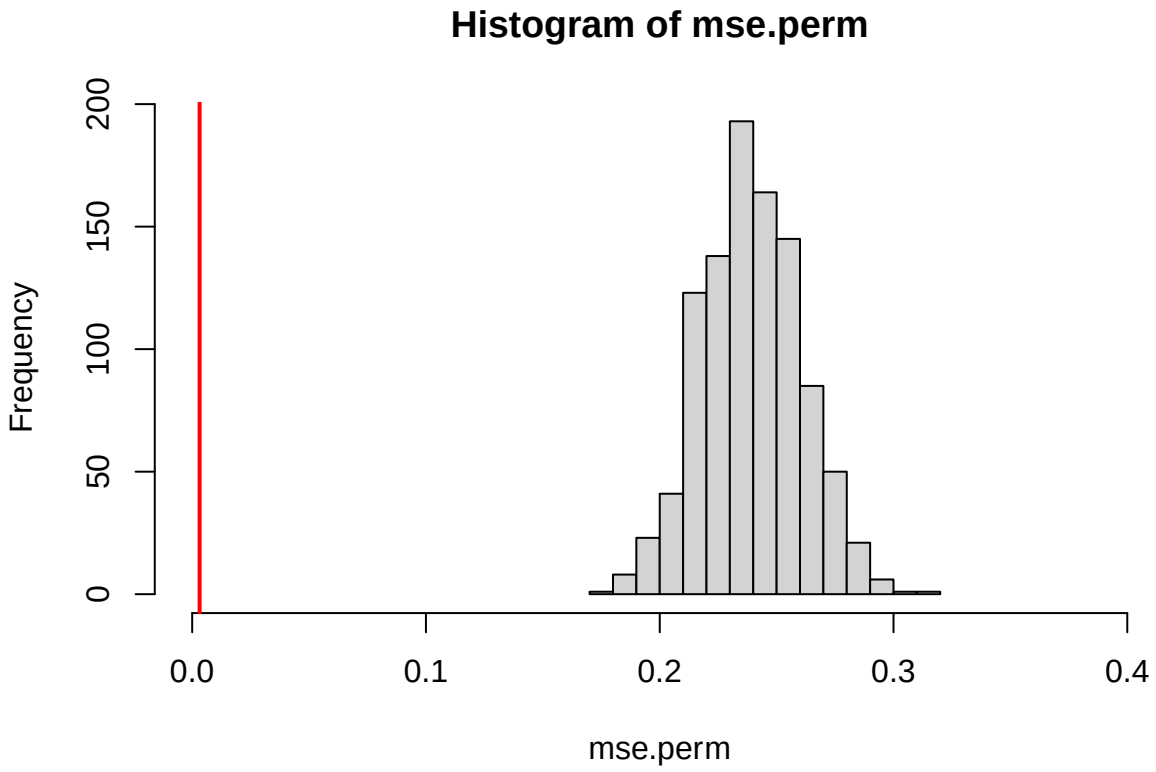
```r
#Create training set with 500 observations on 10 features
set.seed(100)
x_train <- matrix(runif(5000), nrow = 500)
y_train <- rowSums(x_train) + rnorm(500, 0, 0.25^2)
train <- data.frame(Y = y_train, x_train)
#Create testing set with 50 observations
x_test <- matrix(runif(500), nrow = 50)
y_test <- rowSums(x_test) + rnorm(50, 0, 0.25^2)
test <- data.frame(Y = y_test, x_test)
```

```r
#Create model
lin.mod <- lm(Y ~ ., data = train)
#Find mse0
mse0 <- mean((test$Y - predict(lin.mod, test))^2); mse0
```

```
## [1] 0.003171128
```

(f)

```r
#Set number of permutations and mse vector
nperm <- 1000
mse.perm <- rep(0, nperm)
set.seed(100)
for(i in 1:nperm){
  perm.train <- train
  perm.train$X8 <- sample(train$X8)
  perm.train$X9 <- sample(train$X9)
  perm.train$X10 <- sample(train$X10)
  lin.mod.perm <- lm(Y ~ ., data = perm.train)
  mse.perm[i] <- mean((test$Y - predict(lin.mod.perm, test))^2)
}
#Generate histogram with red line at mse0
hist(mse.perm, breaks = 20, xlim = c(0, 0.40))
abline(v = mse0, col = "red", lwd = 2)
```




```r
#Calculate p-value
p.4f <- mean(mse.perm < mse0)
```

As seen above, we get a p-value of 0 and our original MSE lies very far to the left of our permutation distribution. Therefore, we can reject the null hypothesis and conclude that at least one of X8, X9, and X10 have a statistically significant effect on Y.

Problem 5

(a) To conduct a standard parametric hypothesis test to evaluate the vaccine's efficacy, we would use a two-sided proportion z-test. In this test, we would compare the proportions of the those who got COVID between the treated and placebo groups. The null hypothesis would be $\rho_{vaccine} - \rho_{placebo} = 0$ and the alternative hypothesis would be $\rho_{vaccine} - \rho_{placebo} \neq 0$. After confirming that the proportions are statistically significantly different, we could conduct a one-sided test to determine which direction the difference lies.

```
#Generate variables
tot.vaccine <- 43000/2
tot.placebo <- 4300/2
cov.vaccine <- 8
cov.placebo <- 162
prop.test(x = c(cov.vaccine, cov.placebo),
          n = c(tot.vaccine, tot.placebo), alternative = "two.sided")
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(cov.vaccine, cov.placebo) out of c(tot.vaccine, tot.placebo)
## X-squared = 1529.1, df = 1, p-value < 2.2e-16
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.08639276 -0.06356073
## sample estimates:
##      prop 1      prop 2
## 0.000372093 0.075348837
```

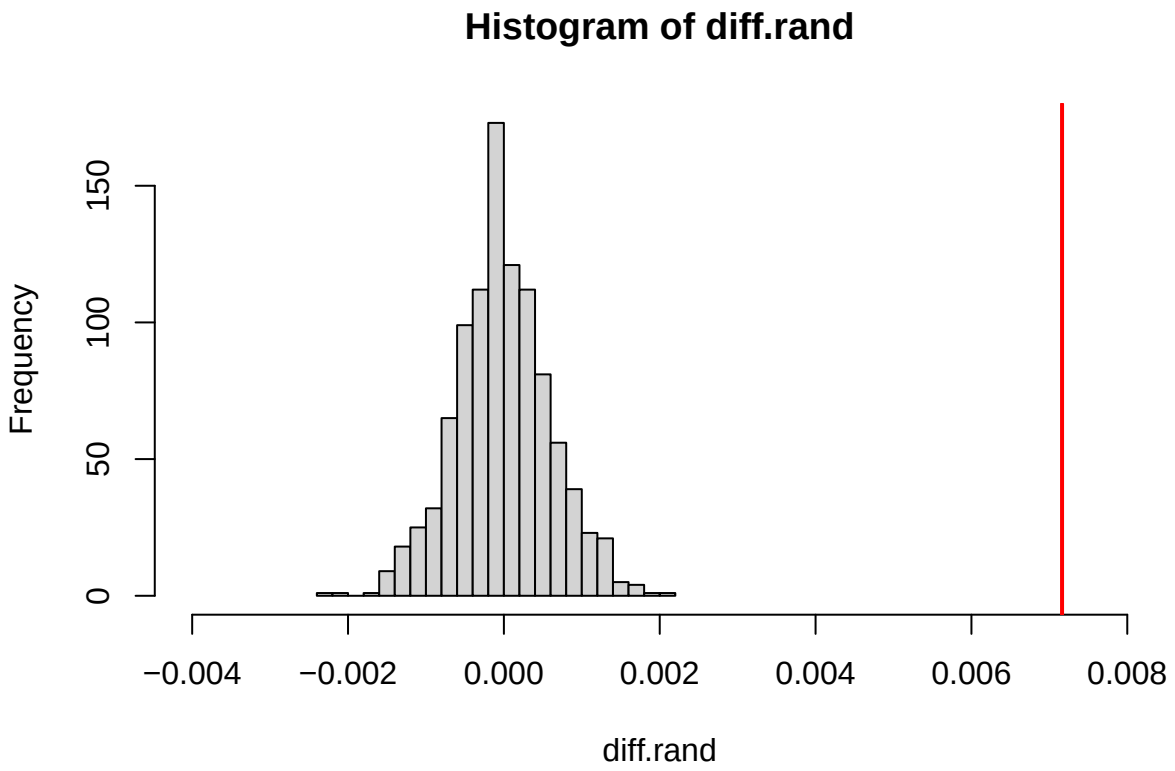
Based on the above results (p-vale of approximately 0), we can reject the null hypothesis that there is no difference in the proportion of treated and placebo patients that got COVID. **(b)** If the vaccine was *not* effective, we would expect to see a similar proportion of COVID infections across both groups. In other words, there would be no significant difference in the proportion of patients infected between the vaccine and placebo groups. **(c)** Given the above answers, we could use the difference in proportions of COVID infections between the vaccine and placebo group as a test statistic. If the actual difference in proportions falls above α in the distribution, we can conclude that there is a significant effect for the vaccine. **(d)**

```
#Generate sample data
tot.cov <- 170
tot.subjects <- 43000
cov.dist <- c(rep(1, tot.cov), rep(0, tot.subjects - tot.cov))
#Define number of randomizations and true proportions
nrand <- 1000
true.diff <- (162/21500) - (8/21500)
```

```
#Set seed and run randomizations
set.seed(100)
diff.rand <- rep(0,nrand)
for (i in 1:nrand) {
  rand_data <- sample(cov.dist)
  Group1 <- rand_data[1:(tot.subjects/2)]
  Group2 <- rand_data[((tot.subjects/2) + 1):tot.subjects]
  diff.rand[i] <- (sum(Group1) / length(Group1)) - (sum(Group2) / length(Group2))
}
#Calculate p-value
p.5d <- mean(diff.rand > true.diff); p.5d
```

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

```r
#Generate histogram with red line at the true difference in proportions
hist(diff.rand, breaks = 20, xlim = c(-0.004, 0.008))
abline(v = true.diff, col = "red", lwd = 2)
```



Based on the above results we can conclude that the difference in proportions of infected individuals between the vaccine and placebo groups is statistically significantly different than 0. Our p-value was 0.