Practice Final Question

Name (UNI)

Section 00#

Instructions (Read this completely first)

You should complete the exam by editting this file directly. Please knit the file often, so that if you make a mistake you catch it before the end of the exam. You will have exactly 30 minutes from the start time to complete the exam. **At the end you must turn in your knitted .pdf file and raw .Rmd file on Courseworks.**

When the time is up, you must shut your computer immediately. We will take off points from anyone whose computer is still open after time is up.

You may use your class notes for the exam, but not the internet. You absolutely may not communicate with anyone else during the exam. Doing so will result in an F in this class and likely result in termination from the MA program. Note that your time will be tight so you will not be able to look up every bit of code from your class notes.

Question 1 (11 points)

The following code imports the `diamonds_small.csv` dataset into `R` and stores it in a dataframe called `diamonds`. Then we use the `lm()` function to regress `price` (response) on `carat` (predictor) and save this result as `lm0`.

```
diamonds <- read.csv("diamonds_small.csv", as.is = TRUE, header = TRUE)
lm0      <- lm(price ~ carat, data = diamonds)
coefficients(lm0)
```

```
## (Intercept)       carat
##   -2256.361    7756.426
```

The estimates $\hat{\beta}_0 = -2256.4$ and $\hat{\beta}_1 = 7756.4$ that you just calculated with `lm()` are functions of the data values and are therefore themselves are random (they inherit variability from the data). If we were to recollect the diamonds data over and over again, the estimates would be different each time.

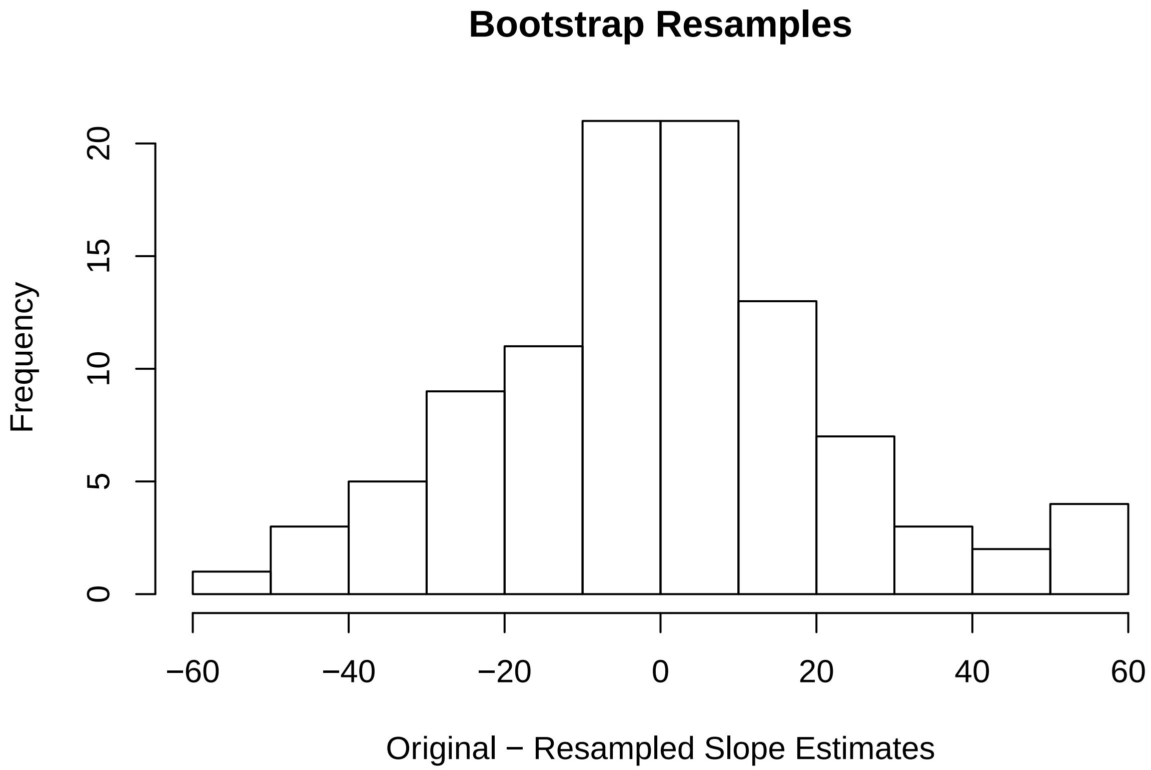
a. (5 points) Calculate $B = 100$ bootstrap resamples of the original data. Use each resampled dataset to provide new estimates of $\hat{\beta}_0$ and $\hat{\beta}_1$. Store these values in a $B \times 2$ matrix called `resampled_ests`.

```
library(plyr)

n <- nrow(diamonds)
B <- 100
resampled_values <- matrix(sample(1:n, n*B, replace = TRUE), nrow = B, ncol = n)
coef.func <- function(resamp_indices) {
  return(coefficients(lm(price ~ carat, data = diamonds[resamp_indices, ])))
}
resampled_ests <- aaply(resampled_values, 1, coef.func)
```

b. (3 points) Recall from lecture that $\left(\hat{\beta}_1^{(b)}\right)_{b=1}^{B} - \hat{\beta}_1$ approximates the sampling distribution of $\hat{\beta}_1 - \beta_1$ where β_1 is the population parameter, $\hat{\beta}_1$ is the estimate from our original dataset, and $\left(\hat{\beta}_1^{(b)}\right)_{b=1}^{B}$ are the B bootstrap estimates. Make a vector `diff_estimates` that holds the differences between the original estimate of $\hat{\beta}_1$ from `lm0` and the bootstrap estimates. It should have length B. Plot a histogram of the estimates of differences given in `diff_estimates`. Label the x-axis appropriately.

```
diff_estimates <- coefficients(lm0)[2] - resampled_ests[, "carat"]
hist(diff_estimates, xlab = "Original - Resampled Slope Estimates", main = "Bootstrap Resamples")
```



c. (1 point) Calculate the variance of the bootstrap estimates of $\hat{\beta}_1$.

```
var(resampled_ests[, "carat"])
```

```
## [1] 517.6379
```

d. (2 points) Finally we'd like to approximate confidence intervals for the regression coefficients. We estimate the confidence interval from the bootstrap estimates by finding a range of $\left(\hat{\beta}_1^{(b)}\right)_{b=1}^{B} - \hat{\beta}_1$ which holds $1 - \alpha$ percent of the values. In our case, let $\alpha = 0.05$, so we estmiate a confidence interval with level 0.95. Let `Cu` and `Cl` be the upper and lower limits of the confidence interval. Use the `quantile()` function to find the 0.025 and 0.975 quantiles of the vector `diff_estimates` calculated in 1.d. Then `Cu` is the sum of the original estimate of $\hat{\beta}_1$ from `lm0` with the upper quantile and `Cl` is the sum of the original estimate of $\hat{\beta}_1$ from `lm0` with the lower quantile.

```
Cl <- coefficients(lm0)[2] + quantile(diff_estimates, 0.025)
Cu <- coefficients(lm0)[2] + quantile(diff_estimates, 0.975)
int <- c(Cl, Cu)
int
```

```
##    carat    carat
## 7711.174 7809.089
```