HW02

Minyoung Do

2/2/2020

For this exercise we consider the following functional form,

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon,$$

where Y is the Joe Biden feeling thermometer, and $[X_1 \dots X_p]$ are the predictive features, including age, gender, education, Democrat, and Republican.

The Questions

1. (10 points) Estimate the MSE of the model using the traditional approach. That is, fit the linear regression model using the entire dataset and calculate the mean squared error for the entire dataset. Present and discuss your results at a simple, high level.

```
# linear regression model across the entire dataset
reg <- lm(biden ~ female + age + educ + dem + rep, nes08)

# creating a tidy table
reg %>%
  tidy() %>%
  kable() %>%
  kable_styling("striped", full_width = F, position = "left") %>%
  column_spec(1, bold = T, background = "pink") %>%
  add_footnote("DV: Feeling Thermometer Towards Biden",
               notation = "alphabet")
```

term	estimate	std.error	statistic	p.value
(Intercept)	58.8112590	3.1244366	18.822996	0.0000000
female	4.1032301	0.9482286	4.327258	0.0000159
age	0.0482589	0.0282474	1.708438	0.0877274
educ	-0.3453348	0.1947796	-1.772952	0.0764057
dem	15.4242556	1.0680327	14.441745	0.0000000
rep	-15.8495061	1.3113624	-12.086290	0.0000000

[a] DV: Feeling Thermometer Towards Biden

```
# mse value
Metrics::mse(nes08$biden, reg$fitted.values)
```

```
## [1] 395.2702
```

Of all five variables we have included in the model, the p-values suggest that only `female`, `dem`, and `rep` have significant relations to the feeling thermometer towards Biden. The coefficient is -15.85 for `rep` and 15.42 for `dem`, while `female` has a coefficient of 4.1. Thus, `female` and `dem` have a positive relationship with the feeling thermometer, while `rep` is negatively correlated to the feeling thermometer. These values also indicate that the effect of `dem` and `rep` is bigger than `female` in the estimated model. That is, the feeling thermometer values towards Biden increase by 15.42 for every unit change in `dem` and decrease by 15.84 for every unit change in `rep`.

The smaller the means squared error, the closer this model is to finding the regression line that fits the best. The mean squared error of 395.27 therefore seems that this linear model might not have the best-fitting line yet, though there is a chance that it may be impossible to get a smaller value than 395.27. I cannot determine anything yet only with one MSE value, as 395.27 might be as good as it gets and we do not have any MSE value to compare; this leaves room for us to check if there is a line that fits better than the one of this model.

2. (30 points) Calculate the test MSE of the model using the simple holdout validation approach.

(5 points) Split the sample set into a training set (50%) and a holdout set (50%). Be sure to set your seed prior to this part of your code to guarantee reproducibility of results.

```
# spliting the dataset into train and test sets
set.seed(4850)
samples <- sample(1:nrow(nes08),
                  nrow(nes08)*0.5,
                  replace = FALSE)
train <- nes08[samples, ]
test <- nes08[-samples, ]
```

(5 points) Fit the linear regression model using only the training observations.

```
reg_train <- lm(biden ~ female + age + educ + dem + rep, train)
reg_train %>%
  tidy() %>%
  kable() %>%
  kable_styling("striped", full_width = F, position = "left") %>%
  column_spec(1, bold = T, background = "pink")
```

term	estimate	std.error	statistic	p.value
(Intercept)	55.0770284	4.2712980	12.8946819	0.0000000
female	3.9876619	1.3061181	3.0530638	0.0023320
age	0.0775095	0.0383172	2.0228371	0.0433862
educ	-0.1596919	0.2665662	-0.5990702	0.5492774
dem	15.7834861	1.4641512	10.7799562	0.0000000
rep	-14.9866781	1.8518974	-8.0926072	0.0000000

(10 points) Calculate the MSE using only the test set observations.

```
reg_test <- lm(biden ~ female + age + educ + dem + rep, test)
Metrics::mse(test$biden, reg_test$fitted.values)
```

```
## [1] 415.2141
```

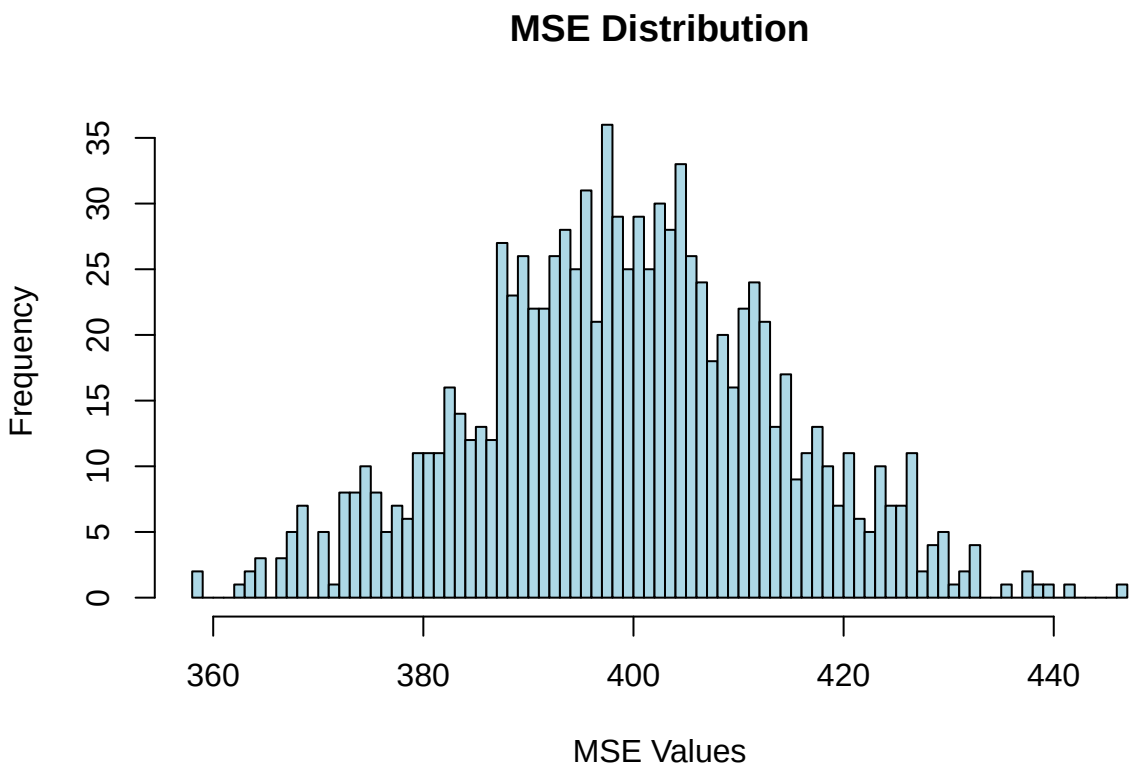
(10 points) How does this value compare to the training MSE from question 1? Present numeric comparison and discuss a bit.

Two mean squared errors can be compared in order to determine how well they explain a set of observations, and the general rule of thumb is that the smaller MSE, the better. If MSE is 0, it means that the estimator $\hat{\theta}$ predicts observations of the parameter θ with perfect accuracy. In this sense, MSE from the question 1, 395.27, which is smaller than the test MSE of 415.21, tells us that the model based on the original dataset has a better prediction accuracy than the test set model. This is because the entire dataset has 1807 observations, while the test set has 50% of the observations (904), which means the original model has more information to learn from than the test set.

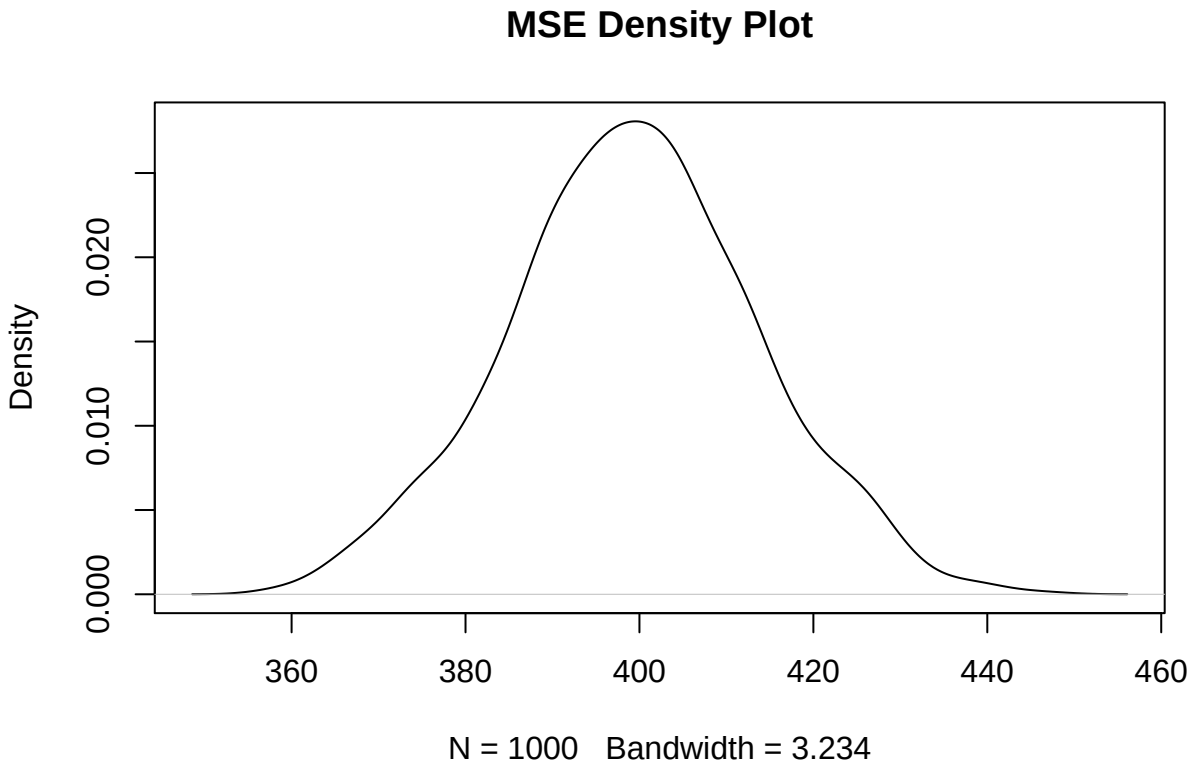
3. (30 points) Repeat the simple validation set approach from the previous question 1000 times, using 1000 different splits of the observations into a training set and a test/validation set. Visualize your results as a sampling distribution (hint: think histogram or density plots). Comment on the results obtained.

```
mse <- vector("double", 1000)
for(i in 1:1000){
  train <- sample(1:nrow(nes08), nrow(nes08)*0.5, replace=F)
  test <- setdiff(1:nrow(nes08), train)
  model <- lm(biden ~ female + age + educ + dem + rep, nes08[train,])
  pred <- predict(model, nes08[test,])
  x <- nes08$biden[test] - pred
  mse[i] <- mean(x*x)
}
```

```r
hist(mse, breaks=100, main="MSE Distribution", col = "lightblue", xlab = "MSE Values")
```




```r
dens <- density(mse)
plot(dens, main = "MSE Density Plot")
```




```
mse %>%
  summary() %>%
  tidy() %>%
  kable() %>%
  kable_styling("striped", full_width = F, position = "left")
```

minimum	q1	median	mean	q3	maximum
358.2712	389.5918	399.2483	399.2708	408.7616	446.3941

These plots show a distribution of MSE values across all 1000 different splits of the dataset. Both mean and median points are at 399.2, and 50% of the data points are between the first quartile (`q1`) and the third auartile (`q3`). From the histogram, we can observe that almost all of the MSE values fall between 360 and 440 except for a very few observations. The smallest MSE value out of 1000 samples is 358.27, while the largest MSE is 446.39 The curve in the density plot looks very similar to the bell curve of normal distribution, yet some fluctuations are found. These plots tell us that the initial MSE we obtained from the regression model of the original dataset is below the mean, meaning it had a better prediction accuracy than the average.

4. (30 points) Compare the estimated parameters and standard errors from the original model in question 1 (the model estimated using all of the available data) to parameters and standard errors estimated using the bootstrap (B = 1000). Comparison should include, at a minimum, both numeric output as well as discussion on differences, similarities, etc. Talk also about the conceptual use and impact of bootstrapping.

```
mu_samp <- mean(nes08$biden)
sem_samp <- sqrt(mu_samp / nrow(nes08))

coefs <- function(splits, ...) {
  model <- lm(..., data = analysis(splits))
  tidy(model)
}

nest_boot <- nes08 %>%
  bootstraps(1000) %>%
  mutate(coef = map(splits, coefs,
                    as.formula(biden ~ female + age + educ + dem + rep)))

# Retrieving a dataframe of summarized mean bootstrap coefficient estimates
biden_bootstraps <- nest_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(
    boot.estimate = mean(estimate),
    boot.se = sd(estimate, na.rm = TRUE)
  )

reg1 <- tidy(reg)

biden_bootstraps %>%
  left_join(reg1, by = "term") %>%
  select("term", "boot.estimate", "estimate", "boot.se", "std.error", "p.value") %>%
  kable() %>%
  kable_styling("striped", full_width = F, position = "left") %>%
  column_spec(1, bold = T, background = "pink")
```

term	boot.estimate	estimate	boot.se	std.error	p.value
(Intercept)	58.7318405	58.8112590	3.1098220	3.1244366	0.0000000
age	0.0479361	0.0482589	0.0291298	0.0282474	0.0877274
dem	15.4252257	15.4242556	1.0541033	1.0680327	0.0000000
educ	-0.3398001	-0.3453348	0.1965720	0.1947796	0.0764057
female	4.1034407	4.1032301	1.0067043	0.9482286	0.0000159
rep	-15.7984757	-15.8495061	1.3486331	1.3113624	0.0000000

Compare the estimated parameters and standard errors from the original model in question 1 (the model estimated using all of the available data) to parameters and standard errors estimated using the bootstrap (B = 1000). Comparison should include, at a minimum, both numeric output as well as discussion on differences, similarities, etc. Talk also about the conceptual use and impact of bootstrapping.

As seen in the table above, the first two columns (excluding `term` column) show the estimated parameters, and the last two columns report the standard errors in both models we are to compare. At first sight, the coefficients and standard errors seem to be very similar to each other. The coefficients in the initial linear model have very small differences from the β estimates in the bootstrapped model. Au contraire, the SEs for `age`, `dem`, `female`, and `rep` in the original model are slightly smaller than the ones from the bootstrapped model; as smaller SE suggest that the model is more representative of the overall population, we can conclude that the original model was able to make a more precise predictions for *most* of the parameters. This indicates that the simple linear model, which is a parametric approach, could be more efficient than bootstrapping (non-parametric approach) in terms of prediction accuracy when accompanied with correct/valid assumptions. Therefore, the efficiency of each approach differs depending on conditions including data size, assumptions, and time constraints.

To understand when to use which approach, it is imperative to review some of the benefits and limitations of parametric and non-parametric approaches. First, parametric approaches are relatively easier to understand the process and interpret results compared to non-parametric approaches. The second factor to consider is data size; parametric approaches do not require as much training data as bootstrapping (non-parametric approach) to estimate the right function. Third, non-parametric approaches like bootstrapping have more flexibility in fitting a more complex functional forms, whereas parametric approaches are more constrained to use a specified functional form.