PS2

Bohan Yin

2/1/2020

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.

```
biden_lm_1 <- lm(biden ~ ., data = data_biden)
(traditional_result <- tidy(biden_lm_1))
```

```
## # A tibble: 6 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  58.8       3.12       18.8  2.69e-72
## 2 female        4.10      0.948       4.33 1.59e- 5
## 3 age           0.0483    0.0282      1.71 8.77e- 2
## 4 educ         -0.345     0.195      -1.77 7.64e- 2
## 5 dem          15.4       1.07       14.4  8.14e-45
## 6 rep         -15.8       1.31      -12.1  2.16e-32
```

```
mse_1 <- mean(biden_lm_1$residuals^2)
```

- Overall, from the table we can see that parameters including being a female and political affiliation are more significant in influencing the attitude towards Biden. In particular, being a democrat means you like Biden more than an independent, and being a republican means you like Biden less than in independent. Mean squared error (MSE) is the average of the squared errors. The linear regerssion model using the entire dataset generates a relatively large MSE of 395.2701693, meaning that the error of this model is large, the predictive power of this model is low, and it requires adjustment on building the 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.**
    - (5 points) *Fit* the linear regression model using *only* the *training* observations.
    - (10 points) Calculate the *MSE* using *only* the *test* set observations.
    - (10 points) How does this value compare to the training MSE from question 1? Present numeric comparison and discuss a bit.

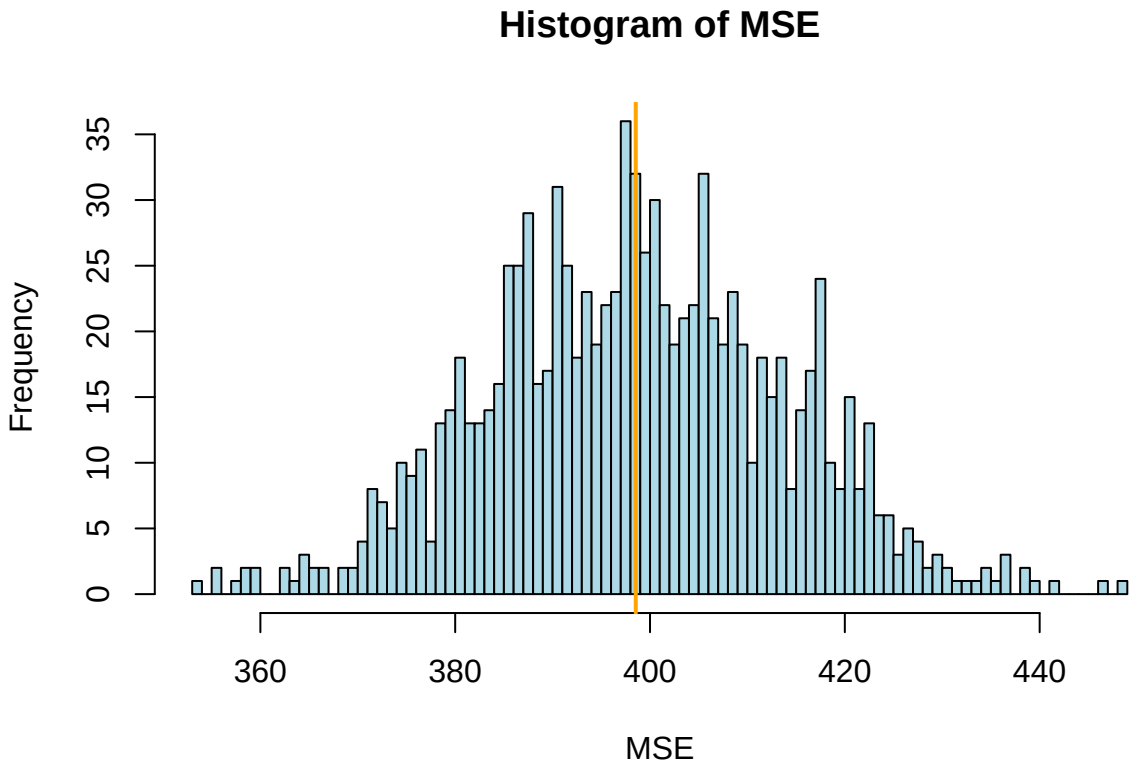
```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        432.
```

- The MSE using the holdout validation approach is 431.60092. It is larger than the mse in first question. This is reasonable because the training data is smaller than the full dataset. After we split the data in

half and train the model on first half, the testing result based on the other half of the data might lead to worse model performance. However, just one instance is not sufficient to see the predictive power of this model, so we might need to repeat the validation approach more times with different splits to see a more holistic result.

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.

```
set.seed(1234)
## repeat 1000 times
mse_1000 <- replicate(1000, {
  split <- initial_split(data = data_biden,
                         prop = 0.5) ## split data in half
  train <- training(split)
  test <- testing(split)
  model <- lm(biden ~ ., data = train)
  biden_mse <- augment(model, newdata = test) %>%
    mse(truth = biden, estimate = .fitted)
  return(biden_mse$.estimate)
  })
{hist(mse_1000,
     breaks = 100,
     main = "Histogram of MSE",
     col = "lightblue",
     xlab = "MSE")
abline(v=mean(mse_1000),col="orange", lwd = 2)}
```



- The histogram shows that as we repeat the validation process 1000 times more, the distribution of mse tends to become normal and the mean smaller, but very close to 400, which is similar with what we got from question 1. Overall, the mse of this model is relatively high, but the bootstrap prediction might perform slightly better than the traditional approach.

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.

```
set.seed(1995)
# bootstrapped estimates of the parameter estimates and standard errors
lm_coefs <- function(splits, ...) {
  ## use `analysis` or `as.data.frame` to get the analysis data
  mod <- lm(..., data = analysis(splits))
  tidy(mod)
}

biden_boot <- data_biden %>%
  bootstraps(1000) %>%
  mutate(coef = map(splits, lm_coefs, as.formula(biden ~ .)))

biden_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(.estimate = mean(estimate),
            .se = sd(estimate, na.rm = TRUE)) %>%
  rename(estimate = .estimate,
         std.error = .se) %>%
  # join with traditional apporach results for comparison purpose
  left_join(traditional_result, by = 'term', suffix = c("_boot", "_trad")) %>%
  subset(select = -c(statistic, p.value))
```

```
## # A tibble: 6 x 5
##   term        estimate_boot std.error_boot estimate_trad std.error_trad
##   <chr>               <dbl>          <dbl>         <dbl>          <dbl>
## 1 (Intercept)       58.8            3.01         58.8            3.12
## 2 age                0.0480         0.0280        0.0483         0.0282
## 3 dem               15.4            1.10         15.4            1.07
## 4 educ              -0.341          0.200        -0.345          0.195
## 5 female             4.12           0.947         4.10           0.948
## 6 rep              -15.8            1.45        -15.8            1.31
```

- The estimated parameters from both approaches are roughly the same in terms of their coefficients. In terms of standard error, three of the variables' standard error from bootstrap is very slightly larger than those from traditional approach, given the fact that in general they are indistinguishable. One possible explanation for the differences could be that traditional approach relies on distribution assumption, whereas bootstrap approach does not, as traditional approach performs better with correct distributional assumption. In this example, bootstrap does not generate a distinguishable result compared with the traditonal approach. Conceptually bootstrap is ideal when we are reluctant to make distributional assumptions, and also when the sample size is not large enough. Under that condition, bootstrap approach allows a more robust estimate result.