Problem Set 2

Artemis Yang

1/30/2020

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.

```r
# load libraries
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
## -- Attaching packages -------------------------------------------------------------------------------

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v stringr 1.4.0
## v tidyr   1.0.0     v forcats 0.4.0
## v readr   1.3.1

## -- Conflicts ----------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(ISLR)
library(broom)
library(rsample)
library(rcfss)
library(yardstick)
```

```
## For binary classification, the first factor level is assumed to be the event.
## Set the global option `yardstick.event_first` to `FALSE` to change this.
```

```
##
## Attaching package: 'yardstick'

## The following object is masked from 'package:readr':
##
##     spec
```

```
library(skimr)

# import data
setwd("/Users/artemisyang/Dropbox/MACS33002 Introduction to Machine Learning/problem-set-2")
nes_data <- read.csv(file="nes2008.csv")

# estimate MSE
lm1 <- lm(biden ~ female+age+educ+dem+rep, data=nes_data)
out <- summary(lm1)
tidy(lm1)
```

```
## # 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 <- augment(lm1, newdata = nes_data) %>%
    mse(truth = biden, estimate = .fitted))
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        395.
```

The MSE of the fitted model is 395.2702. This is the average of the squared errors, which is commonly used to evaluate the performance of a model for regression problems. The smaller the MSE, the closer we are to finding the line of best fit.

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.

```
set.seed(1234)
split <- initial_split(data = nes_data,
                       prop = 0.5)
train <- training(split)
test <- testing(split)
```

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

```
lm2 <- lm(biden ~ female+age+educ+dem+rep, data=train)
summary(lm2)
```

```
##
## Call:
## lm(formula = biden ~ female + age + educ + dem + rep, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.880 -11.950   1.929  11.899  46.124
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  58.68937    4.30323  13.638  < 2e-16 ***
## female        4.41344    1.28889   3.424 0.000644 ***
## age           0.04460    0.03858   1.156 0.247980
## educ         -0.18263    0.26831  -0.681 0.496251
## dem          13.63872    1.45353   9.383  < 2e-16 ***
## rep         -18.76842    1.78349 -10.523  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.11 on 898 degrees of freedom
## Multiple R-squared:  0.3085, Adjusted R-squared:  0.3046
## F-statistic: 80.12 on 5 and 898 DF,  p-value: < 2.2e-16
```

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

```
(mse <- augment(lm2, newdata = test) %>%
  mse(truth = biden, estimate = .fitted))
```

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

- (10 points) How does this value compare to the training MSE from question 1? Present numeric comparison and discuss a bit. The MSE of the test set using the model fitted from the traning set is 431.6, which is larger than the MSE from question 1. This means that the model does not fit the test observations very well.

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.

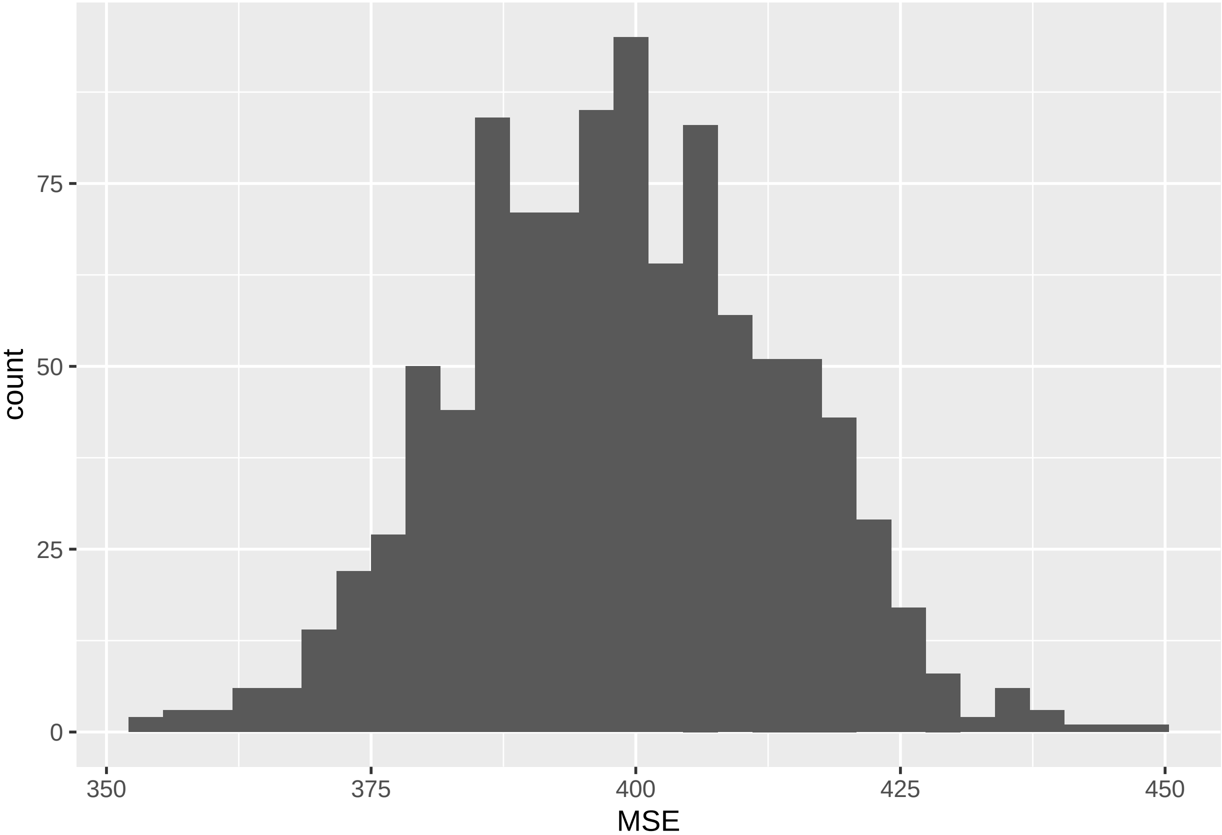
```
# create a 2*1000 tibble
sv_mse <- tibble(times=1:1000, MSE=0)

# repeat simple validation approach 1000 times, store MSE values
x <- 1
```

```
repeat {
  split <- initial_split(data = nes_data,
                                 prop = 0.5)
  train <- training(split)
  test <- testing(split)
  lm <- lm(biden ~ female+age+educ+dem+rep, data=train)
  res <- augment(lm, newdata = test) %>%
  mse(truth = biden, estimate = .fitted)
  res
  sv_mse[x,2] <- res$.estimate
  x = x+1
  if (x == 1001){
    break
  }
}

# graph the distribution of MSE values
ggplot(sv_mse) + geom_histogram(aes(x = MSE))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The distribution of the 1000 MSEs follows a normal distribution, which is consistent with the random sampling process. The median is about 400, which is close to the MSE obtained from question 1. However, we can see that the MSE from question 2 is beyond 425, which is at the far right side of the distribution. This tells us that cross validation is very important in ensuring the accuracy of a model, as there is too much randomness using just one split.

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.

```
# parameter estimates and standard errors from question 1
tidy(lm1)
```

```
## # 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
```

```
# bootstrapped estimates of the parameter estimates and standard errors
set.seed(1234)
lm_coefs <- function(splits, ...) {
  mod <- lm(..., data = analysis(splits))
  tidy(mod)
}

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

biden_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(.estimate = mean(estimate),
            .se = sd(estimate, na.rm = TRUE))
```

```
## # A tibble: 6 x 3
##   term        .estimate    .se
##   <chr>           <dbl>  <dbl>
## 1 (Intercept)   58.9    3.04
## 2 age            0.0471 0.0294
## 3 dem           15.4    1.05
## 4 educ          -0.343  0.196
## 5 female         4.10   0.926
## 6 rep          -15.9    1.39
```

The estimated parameters and standard errors from the original model are generally the same as from the bootstrapped estimates. This is because bootstrapped estimates do not rely on any distributional assumptions, whereas the traditional estimates do. Conceptually, we would prefer to use bootstrap method when our data cannot meet the distributional assumptions, where the bootstrap estimator will give more accurate results. Another possible situation to use bootstrap is when our data is not large enough. In this case, bootstrap can potentially increase the robustness of our estimate. Also, even if the distributional assumptions are met and the data is large, we can still use bootstrap as a robustness check, as the bootstrap method is not biased by distributional assumptions.