Problem Set 2: Uncertainty, Holdouts, and Bootstrapping

Jinglin Yang

Feb 2020

```
library(ggplot2)
library(tidyverse)
library(ISLR)
library(broom)
library(rsample)
library(rcfss)
library(yardstick)
library(magrittr)
```

Question 1.

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 square error for the entire dataset. Present and discuss your results at a simple, high level.

```
> nes2008<- read.csv("C:/Users/lenovo/Desktop/problem-set-2-master/nes2008.csv",header=T)
> attach(nes2008)
> names(nes2008)
[1] "biden"  "female" "age"    "educ"   "dem"    "rep"
>
> # fit the linear regression model
> lm.fit<- lm(biden~female + age + educ + dem + rep)
> summary(lm.fit)

Call:
lm(formula = biden ~ female + age + educ + dem + rep)

Residuals:
    Min      1Q  Median      3Q     Max
-75.546 -11.295   1.018  12.776  53.977

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  58.81126    3.12444  18.823  < 2e-16 ***
female        4.10323    0.94823   4.327 1.59e-05 ***
```

```
age           0.04826    0.02825   1.708   0.0877 .
educ         -0.34533    0.19478  -1.773   0.0764 .
dem          15.42426    1.06803  14.442  < 2e-16 ***
rep         -15.84951    1.31136 -12.086  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 19.91 on 1801 degrees of freedom
Multiple R-squared:  0.2815,    Adjusted R-squared:  0.2795
F-statistic: 141.1 on 5 and 1801 DF,  p-value: < 2.2e-16
>
> # calculate the MSE for the entire dataset
> (mse <- augment(lm.fit, newdata = nes2008) %>%
+   mse(truth = biden, estimate = .fitted))
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 mse     standard        395.
```

The MSE for the entire dataset using the linear regression model is 395.2702. And the R^2 is 0.2815. The large MSE and the low R^2 suggest that the data does not fit the model well.

All the estimated parameters are significant at least at 0.1 level. Especially the estimates of female, dem, rep are highly statistically significant. The results suggest that females have more feeling of "warmth" towards Joe Biden than males on average. And on average, democratic people have more feeling of "warmth" towards Joe Biden than independents, republican people have less feeling of "warmth" towards Joe Biden than independents. Moreover, older people tend to have more feeling of "warmth" while people with higher education tend to have less feeling of "warmth".

Question 2.

Calculate the test MSE of the model using the simple holdout validation approach.

1. Split the sample set into a training set (50%) and a holdout set (50%).

```
> set.seed(1234)
> nes_split <- initial_split(data = nes2008,
+                             prop = 0.5)
> nes_train <- training(nes_split)
> nes_test <- testing(nes_split)
```

2. Fit the linear regression model using only the training observations.

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

Call:
lm(formula = biden ~ female + age + educ + dem + rep, data = nes_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
```

3. Calculate the MSE using only the test set observations.

```
> (test_mse <- augment(lm.train, newdata = nes_test) %>%
+   mse(truth = biden, estimate = .fitted))
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 mse     standard        432.
```

The MSE using only the test set observations is 431.6009.

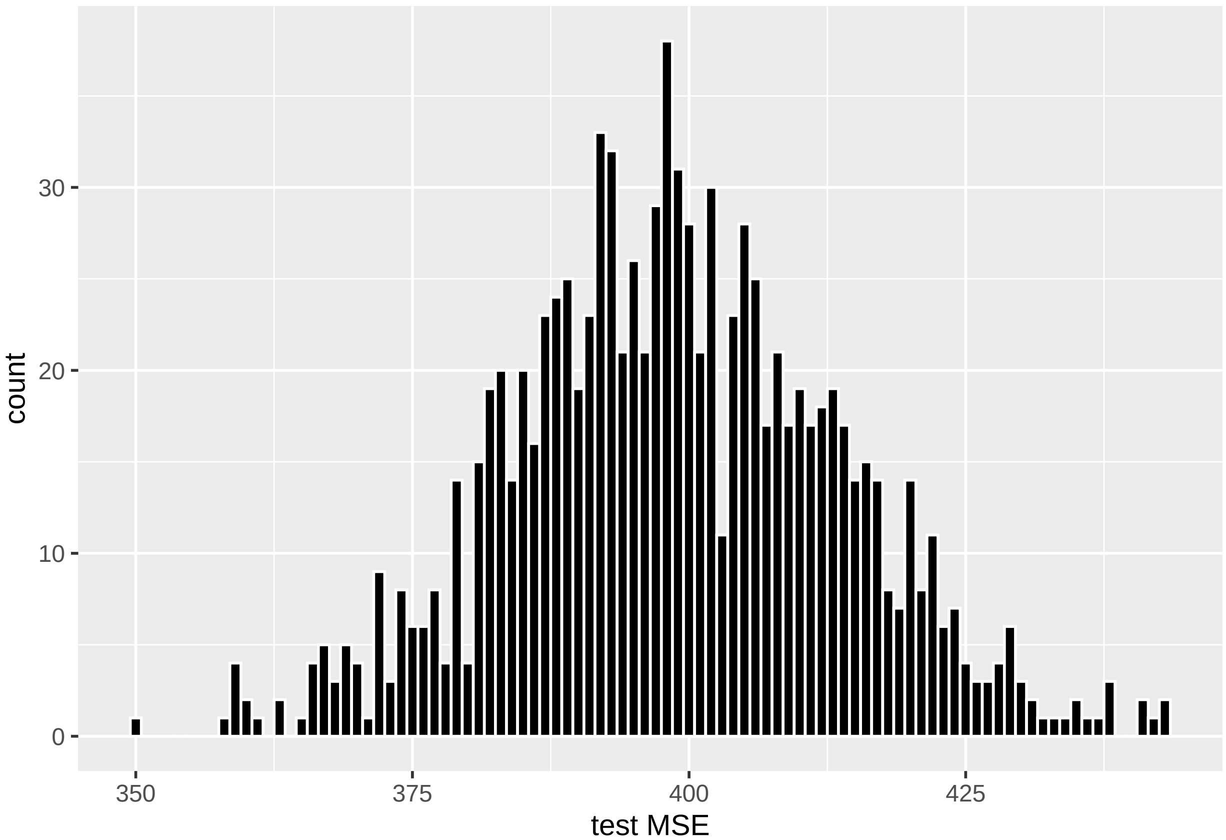
4. How does this value compare to the training MSE from question 1? Present numeric comparison and discuss a bit.

The MSE using the entire dataset is 395.2702, which is smaller than the MSE using only the test set: 431.6009. Because the mean absolute error is around 20 ($\sqrt{395} = 19.87, \sqrt{431} = 20.76$), and the range of the response variable is only 0-100, the training MSE and the test MSE are both high, suggesting model underfitting. However, because there is only one test MSE and the test MSE is affected by the seed we set before, we are not sure about the conclusion. We had better repeat the simple validation set approach for multiple times, which will be done in Question 3.

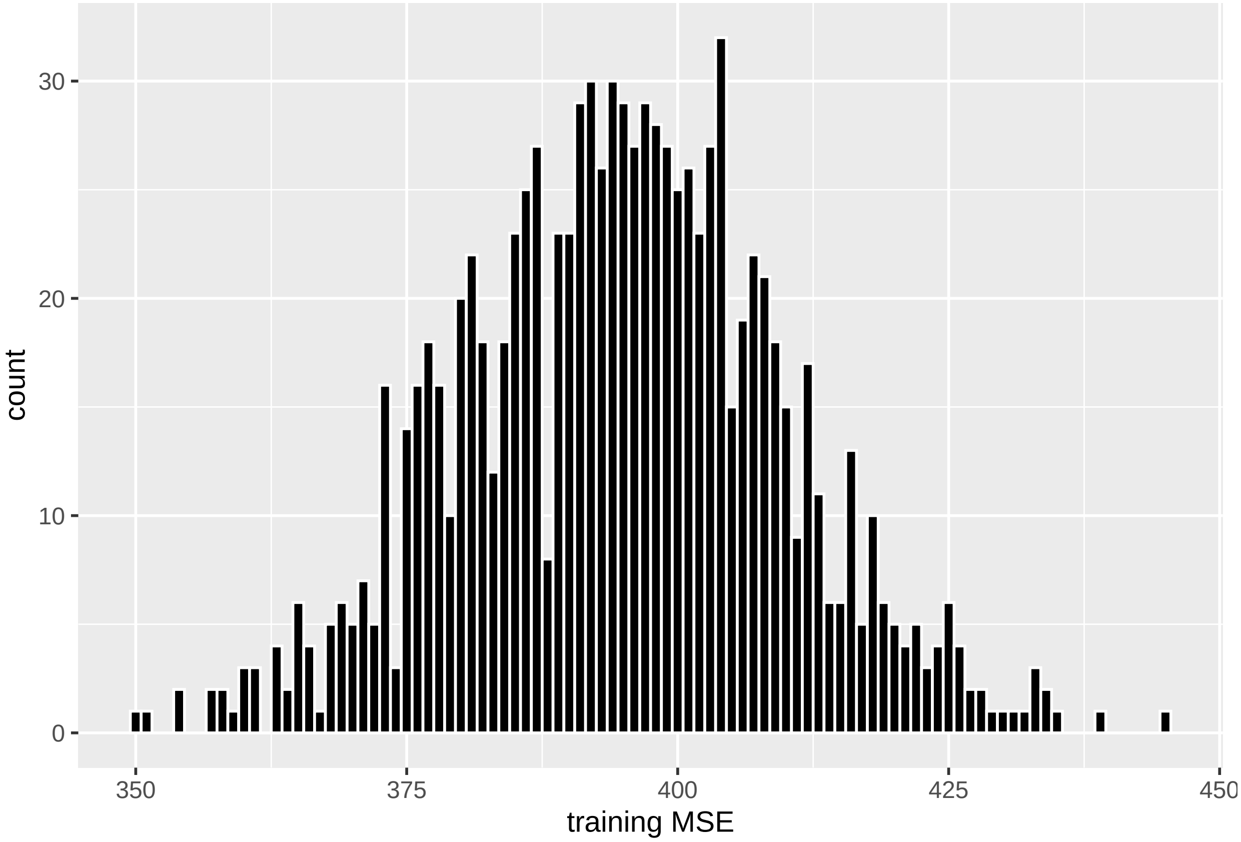
Question 3.

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.

```
> n <- 0
> test_mses <- data.frame(mse=numeric())
> while(n<1000){
+   set.seed(1234+n)
+   nes_split <- initial_split(data = nes2008,
+                              prop = 0.5)
+   nes_train <- training(nes_split)
+   nes_test <- testing(nes_split)
+   lm.train <- lm(data=nes_train, biden~female + age + educ + dem + rep)
+   test_mse <- augment(lm.train, newdata = nes_test) %>%
+   mse(truth = biden, estimate = .fitted)
+   test_mses <- rbind(test_mses,test_mse$.estimate)
+   n <- n+1
+ }
> names(test_mses)[1]="test_mse"
> # draw the histogram plots for the 1000 MSEs
> ggplot(data=test_mses,aes(x=test_mse)) +
+   geom_histogram(binwidth=1, color='white', fill='black') +
+   labs(x="test MSE")
```



From the histogram plot for the 1000 test MSEs, we can see that they are approximately normal distributed with mean around 400, which is close to the training MSE in Question 1. Also most of our test MSEs are between 360 and 440, suggesting that our test MSE is stably high. (Because $\sqrt{400} = 20$, and the range of the response variable: biden is only 0 – 100.) The result may be caused by underfitting or overfitting, depending on the value of training MSE. If the training MSE is also high, then we can conclude the model underfits. Otherwise, we may conclude that the model overfits. Therefore, I also draw a histogram plot for the 1000 training MSEs.



From the histogram plot above, I find that the sampling distribution of the 1000 training MSEs is quite similar to the the sampling distribution of the 1000 test MSEs. Therefore, I think the results suggest model underfitting.

Question 4.

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.

```
> attach(nes2008)
The following objects are masked from nes2008 (pos = 3):

    age, biden, dem, educ, female, rep
>
> # traditional parameter estimates and standard errors
> tidy(lm.fit)
# 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
> lm_coefs <- function(splits, ...) {
+   ## use `analysis` or `as.data.frame` to get the analysis data
+   mod <- lm(..., data = analysis(splits))
+   tidy(mod)
+ }
>
> nes_boot <- nes2008 %>%
+   bootstraps(1000) %>%
+   mutate(coef = map(splits, lm_coefs, as.formula(biden~female + age + educ + dem + rep)))
>
> nes_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    3.02
2 age            0.0485 0.0289
3 dem           15.4    1.03
4 educ          -0.356  0.194
5 female         4.10   0.965
6 rep          -15.9    1.40
```

Generally, both set of estimated parameters are almost the same. The bootstrapped estimates of parameters are slightly smaller than the estimated parameters from the original model in Question 1, except for the estimated parameter of female, which is 4.117 using the bootstrap and 4.103 using all of the available data. As for the standard errors, the bootstrapped estimates and the traditional estimates are virtually identical. Specifically, the standard errors using the bootstrap are slightly larger for the parameters of female, age, and republican, and they are slightly smaller for intercept, education, and democrat. These results suggest these estimates are precise.

The bootstrap method is a resampling technique for estimating a sampling distribution. And we can also use it to estimating SEs and CIs for anything calculated from the data, and it does not rely on any distributional assumptions. Generally speaking, if we draw our sample from an unknown distribution, in order to estimate the SEs and CIs, we could collect multiple, independent samples. But actually carrying out this scenario is usually not feasible (time, money, etc.). Fortunately, the bootstrap method provides a new way to generate “new samples” by repeated sampling from the sample we have with replacement. Specifically, the bootstrap procedure for B bootstrap rounds with a dataset of size n is,

- randomly draw a single observation and assign it to the j-th bootstrap sample;
- repeat until all the B bootstrap samples has size n.

By this way, we view the sample as a population and thus, sampling from it could give us more information about the sampling distribution. Generally, because the bootstrap estimates are not biased by distributional assumptions, they are more robust.