Problem Set 2

Soowon Jo

2/2/2020

Problem Set 2: Uncertainty, Holdouts, and Bootstrapping

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

```
##   Attaching packages                                    tidyverse 1.3.0

##   ggplot2 3.2.1      purrr   0.3.3
##   tibble  2.1.3      dplyr   0.8.3
##   tidyr   1.0.0      stringr 1.4.0
##   readr   1.3.1      forcats 0.4.0

##   Conflicts                                   tidyverse_conflicts()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'modelr'

## The following object is masked from 'package:rcfss':
##
##     mse
```

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

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

```
mse(nes_lm, nes)
```

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

When the model is ideally trained, MSE becomes closer to zero, meaning that actual outputs exactly match the expected outputs. However, the result shown above indicates that age and education do not have a significant effect on predicting how one would feel about Biden. Whether the person is female or either a Democrat or a Republican, play significant predictors on how people would feel about Biden. To be specific, a female is more likely to rate 4.1 points higher on average on the feeling thermometer meaning the person have more feelings of warmth towards Biden compared to how much warmth feeling a male has toward Biden. Moreover, a Democrat is more likely to rate Biden 15.4 points higher on average while a Republican is more likely to rate Biden 15.9 points lower on average.

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

```
## 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 objects are masked from 'package:modelr':
##
##     mae, mape, rmse
```

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

```
##
## Attaching package: 'broom'
```

```
## The following object is masked from 'package:modelr':
##
##     bootstrap
```

2.1 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.

```r
set.seed(1234)

auto_split <- initial_split(data = nes,
                            prop = 0.5)
auto_train <- training(auto_split)
auto_test <- testing(auto_split)
```

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

```r
nes_train_lm <- lm(biden ~ female + age + educ + dem + rep, data = auto_train)
summary(nes_train_lm)
```

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

2.3 Calculate the MSE using only the test set observations.

```r
MSE <- function(estimatedValues, actualValues, sampleSize) {
    sum((estimatedValues - actualValues)^2) / sampleSize;
}

estimatedValues = predict(nes_train_lm, subset(auto_test, select=c("female", "age","educ","dem","rep")))
actualValues = auto_test$biden;
sampleSize = nrow(auto_test);

MSE(estimatedValues, actualValues, sampleSize)
```

```
## [1] 431.6009
```

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

```
compare = mse(nes_lm, nes) - MSE(estimatedValues, actualValues, sampleSize)
compare
```

```
## [1] -36.33075
```

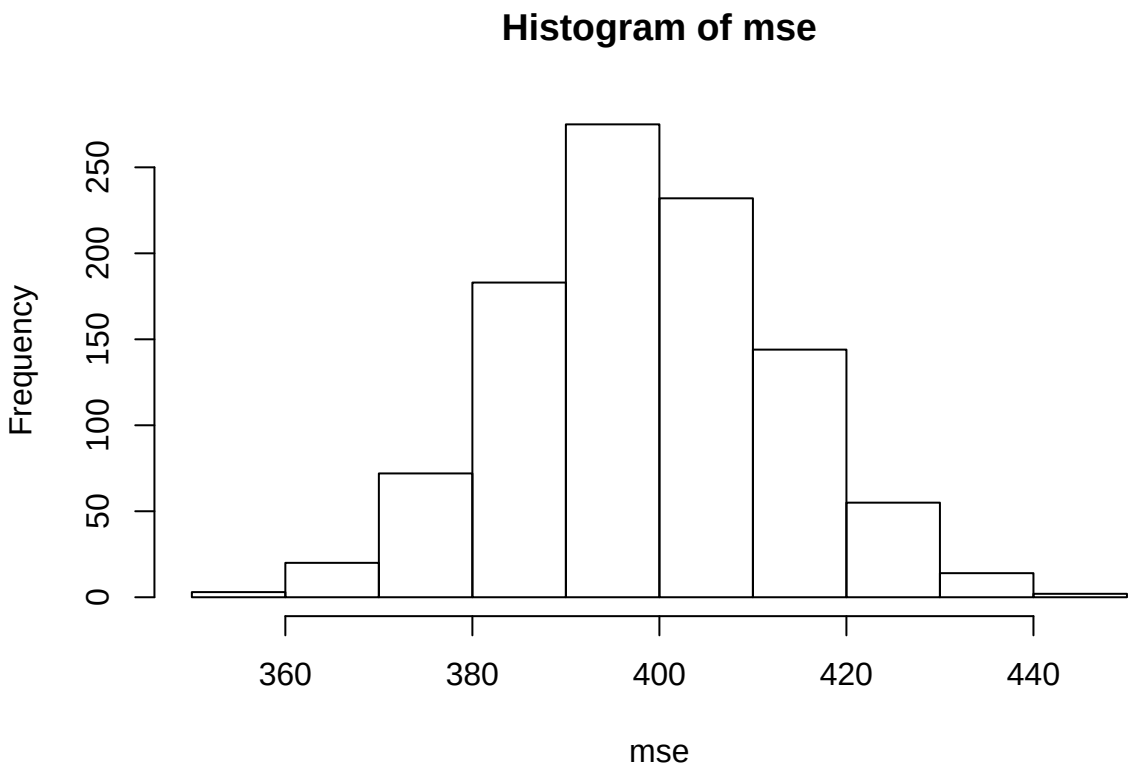
The model trained on the entire dataset has a lower MSE than the model trained on only the test set from question 2.3. This result may be the case since fewer observations used in the model give more noise in the data, which ultimately leads to increase in MSE.

####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.

```
set.seed(5)
mse <- vector("double",1000)

for(i in 1:1000){
    train = sample(1:nrow(nes), 0.5*nrow(nes))
    test = setdiff(1:nrow(nes),train)
    mod <- lm(biden ~ female + age+ educ + dem + rep,
        data = nes[train,])
    pred <- predict(mod, nes[test,])
    x <- nes$biden[test]-pred
    mse[i] <- mean(x*x)
}

hist(mse)
```




```
mean(mse)
```

```
## [1] 398.6048
```

The average of MSEs generated from the test is approximately 398.6, which is very close to the MSE (395.2702) of the original model. Repeating the simple validation set approach, thus, performs quite well at approximating the parameters of the original model.

####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.

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

lm_coefs <- function(splits, ...){
    mod <- lm(..., data = analysis(splits))
    tidy(mod)
}

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

```
biden_boot_lm_df <- nes_boot %>%
    unnest(coef) %>%
    group_by(term) %>%
    summarize(boot.estimate = mean(estimate),
        boot.se = sd(estimate, na.rm = TRUE))

biden_lm_df <- tidy(nes_lm)

biden_lm_df <- biden_lm_df %>%
    left_join(biden_boot_lm_df, by="term") %>%
    select(c("term","boot.estimate","boot.se","estimate","std.error"))

biden_lm_df
```

```
## # A tibble: 6 x 5
##   term        boot.estimate boot.se estimate std.error
##   <chr>               <dbl>   <dbl>    <dbl>     <dbl>
## 1 (Intercept)       58.8     2.92    58.8       3.12
## 2 female             4.11    0.963    4.10      0.948
## 3 age                0.0480  0.0289   0.0483    0.0282
## 4 educ              -0.342   0.191   -0.345     0.195
## 5 dem               15.4     1.10    15.4       1.07
## 6 rep              -15.8     1.37   -15.8       1.31
```

The estimated parameters and standard errors retrieved from both original linear model and bootstrap model are almost identical. The major difference between the two outputs is that the standard errors across the input variables (with the exception of intercept coeffcients) generated in the bootstrap model were higher than those in the original linear model. From this result, we could assume that parametric approach, in which we make assumptions about the parameters (defining properties) of the population distribution from which the data is drawn, use more information than a non-parametric approach. Lower standard errors for the model's parameters, therefore, means that the estimations of parametric approach is more precise.

Bootstrap is a resampling method by independently sampling with replacement from an existing sample data with same sample size n, and performing inference among these resampled data. This method is useful when we have a very small sample size and a data whose distributional form is unknown. Sample sizes as small as 10 can also be usable for bootstrap and we are still able to create model ensemble by combining predictions from multiple models. The method is capable of doing so since it attempts to capture the structure of the data itself. With the use of resampling and model being fitted many times, the bootstrap method yields results that avoid fitting against the noise and pecularities of some individual sample in the data. However, bootstrapping will yield about the same results as fitting a parametric model when the data is large and follows a distribution.