Pset 2, Machine Learning

Rohen Shah

2/3/2020

Question 1

The code below produces a standard linear regression model for the entire dataset. The estimated regression is:

$$Y = 58.81 + 4.10Fem + 0.048Age - .345Educ + 15.42Dem - 15.85Rep$$

The MSE of this regerssion is 395.25, which can be used to compute the R^2 value of .28, indicating that the linear model explains 28 percent of variance in the outcome. The MSE itself is quite large, relative to the value of the y-variables, which is consistent with the fairly low R-squared value.

```
nes <- read_csv("nes2008.csv")
```

```
## Parsed with column specification:
## cols(
##   biden = col_double(),
##   female = col_double(),
##   age = col_double(),
##   educ = col_double(),
##   dem = col_double(),
##   rep = col_double()
## )
```

```
reg1 <- lm(biden ~ female + age + educ + dem + rep, data=nes)
summary(reg1)
```

```
##
## 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
```

```r
mse(reg1, nes)
```

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

Question 2

The code below calculates estimates a model using just the training data (half of the sample) and finds that the MSE in this half is 366.7. This number is smaller than the MSE from part (1), which makes sense because this model is minimizing the squared errors (loss function) from just the training half of the data, which is why it might do a better job than the model in (1) that was trying to minimize the MSE from the entire dataset.

The linear model on the training dataset was then used to compute the MSE for the "test" observations (the other half of the dataset) and the MSE was 426.3. This is a higher average distance from the mean than the overall sample, which is reasonable given that this is an out of sample prediction rather than a "fit" where the model itself is based on the same observations.

```r
set.seed(42)
nes_split <- initial_split(data = nes, prop = 0.5)
nes_train <- training(nes_split)
nes_test <- testing(nes_split)

reg2 <- lm(biden ~ female + age + educ + dem + rep, data=nes_train)
mse(reg2, nes_train)
```

```
## [1] 366.6938
```

```r
mse(reg2, nes_test)
```

```
## [1] 426.2762
```

Question 3

After 1000 iterations, we can see that the average out of sample (test data) MSE is 399.19, which is fairly close to the original entire sample's value but is slightly higher. The histogram below shows the distribution of the MSE's observed.

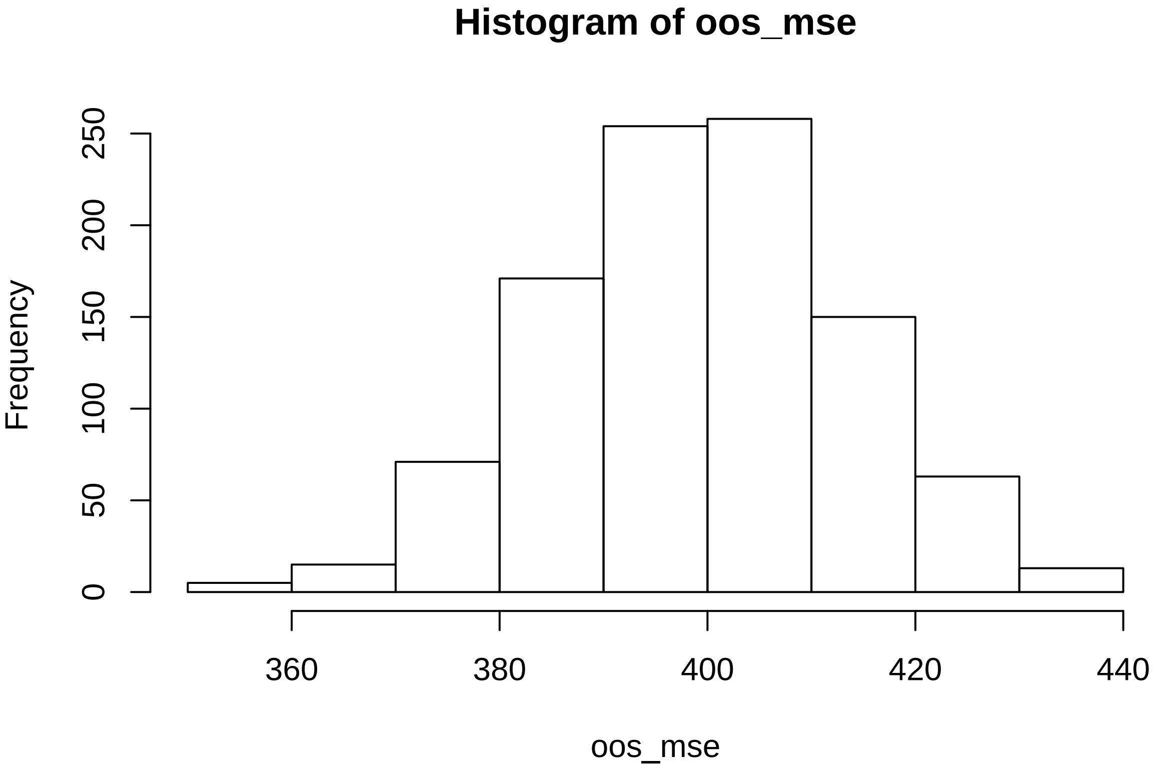
```r
is_mse= c(1:1000)
oos_mse= c(1:1000)
set.seed=12345
for (i in 1:1000) {
nes_split <- initial_split(data = nes, prop = 0.5)
nes_train <- training(nes_split)
nes_test <- testing(nes_split)
reg3 <- lm(biden ~ female + age + educ + dem + rep, data=nes_train)
is_mse[i] = mse(reg3, nes_train)
oos_mse[i] =mse(reg3, nes_test)
}
mean(is_mse)
```

```
## [1] 394.0764
```

```r
mean(oos_mse)
```

```
## [1] 399.1925
```

```r
hist(oos_mse)
```



Question 4

The bootstrap results for each coefficient is very similar to that of the original sample. The estimates of the parameters are very close to each other (usually within 2 decimal places), but the *standard errors are higher in the bootstrap.* The following are the estimates:

Female: 4.102 (bootstrap) and 4.103 (whole sample) Standard error of 1.34 (bootstrap) and .95 (whole sample)

Age: .0487 (bootstrap) and .0483 (whole sample) Standard error of .0399 (bootstrap) and .0283 (whole sample)

Educ: -.347 (bootstrap) and -.345 (whole sample) Standard error of .276 (bootstrap) and .195 (whole sample)

Dem: 15.41 (bootstrap) and 15.42 (whole sample) Standard error of 1.51 (bootstrap) and 1.07 (whole sample)

Rep: -15.83 (bootstrap) and -15.85 (whole sample) Standard error of 1.86 (bootstrap) and 1.31 (whole sample)

```r
fem_val= c(1:1000)
fem_se= c(1:1000)
age_val= c(1:1000)
age_se= c(1:1000)
educ_val= c(1:1000)
educ_se= c(1:1000)
dem_val= c(1:1000)
dem_se= c(1:1000)
rep_val= c(1:1000)
```

```r
rep_se= c(1:1000)

set.seed=12345 #Same seed so this replicates the iterations from part 3
for (i in 1:1000) {
nes_split <- initial_split(data = nes, prop = 0.5)
nes_train <- training(nes_split)
nes_test <- testing(nes_split)
reg3 <- lm(biden ~ female + age + educ + dem + rep, data=nes_train)
fem_val[i] = summary(reg3)$coefficients[2, 1]
fem_se[i] = summary(reg3)$coefficients[2, 2]
age_val[i] = summary(reg3)$coefficients[3, 1]
age_se[i] = summary(reg3)$coefficients[3, 2]
educ_val[i] = summary(reg3)$coefficients[4, 1]
educ_se[i] = summary(reg3)$coefficients[4, 2]
dem_val[i] = summary(reg3)$coefficients[5, 1]
dem_se[i] = summary(reg3)$coefficients[5, 2]
rep_val[i] = summary(reg3)$coefficients[6, 1]
rep_se[i] = summary(reg3)$coefficients[6, 2]
}

mean(fem_val)
```

```
## [1] 4.102446
```

```r
mean(fem_se)
```

```
## [1] 1.341706
```

```r
mean(age_val)
```

```
## [1] 0.04872291
```

```r
mean(age_se)
```

```
## [1] 0.03998528
```

```r
mean(educ_val)
```

```
## [1] -0.3471896
```

```r
mean(educ_se)
```

```
## [1] 0.2756597
```

```r
mean(dem_val)
```

```
## [1] 15.40852
```

```r
mean(dem_se)
```

```
## [1] 1.510126
```

```r
mean(rep_val)
```

```
## [1] -15.83218
```

```r
mean(rep_se)
```

```
## [1] 1.856876
```

```r
summary(reg1)
```

```
##
## 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
```