Homework 2

Michael Carrion

February 3, 2020

1.

```
myData <- read.csv("nes2008.csv")
fullOLS <- lm(biden ~ . ,myData)
summary(fullOLS)
```

```
##
## Call:
## lm(formula = biden ~ ., data = myData)
##
## 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 <- mean(fullOLS$residuals^2)
cat("MSE:",round(mse,2))
```

```
## MSE: 395.27
```

As can be seen above, female participants exhibit slightly more warmth towards Biden than male participants (on average 4 sentiment points). Older individuals are, on average, marginally more likely to be warm to Biden than younger individuals (holding all other factors constant), though it remains unclear if age is a statistically significant feature. Education is associated inversely with warmth towards Biden: An individual with an extra year of education is more likely to be 0.3 points colder towards Biden than a similar individual with one year less of education (though it is also unclear if education is a statistically significant predictor). Unsurprisingly, an individual identifying as a democrat is likely to be 15 sentiment points warmer towards Biden than an independent, and a Republican is likely to be 15 sentiment points colder towards Biden than an independent. Overall, our model has an adjusted R-squared of 0.28, suggesting a relatively good fit. Further, when fit on the entire dataset, our mean squared error is 395.27 sentiment points squared.

2.

```
set.seed(118)
samples <- sample(1:nrow(myData), nrow(myData)*0.5, replace = FALSE)
train <- myData[samples, ]
```

```r
test <- myData[-samples, ]

myOLS2 <- lm(biden ~ . ,train)
summary(myOLS2)
```

```
##
## Call:
## lm(formula = biden ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -75.625 -11.843   1.787  12.066  45.120
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 54.09094    4.58997  11.785  < 2e-16 ***
## female       3.92408    1.36444   2.876  0.00412 **
## age          0.04320    0.04018   1.075  0.28261
## educ        -0.06747    0.28816  -0.234  0.81494
## dem         17.21029    1.51834  11.335  < 2e-16 ***
## rep        -15.53252    1.89446  -8.199 8.35e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.24 on 897 degrees of freedom
## Multiple R-squared:  0.287,  Adjusted R-squared:  0.283
## F-statistic: 72.21 on 5 and 897 DF,  p-value: < 2.2e-16
```

Thus, the coefficients do not differ greatly from those in the previous model. The warmth effect of being a democrat is perhaps slightly greater in this new model. Also, the adjusted R-squared is 0.283, suggesting that this model is a marginally better fit.

```r
preds <- predict(myOLS2, newdata = test)
mse2 <- mean((test$biden - preds)^2)
cat("MSE:",mse2)
```

```
## MSE: 386.4843
```

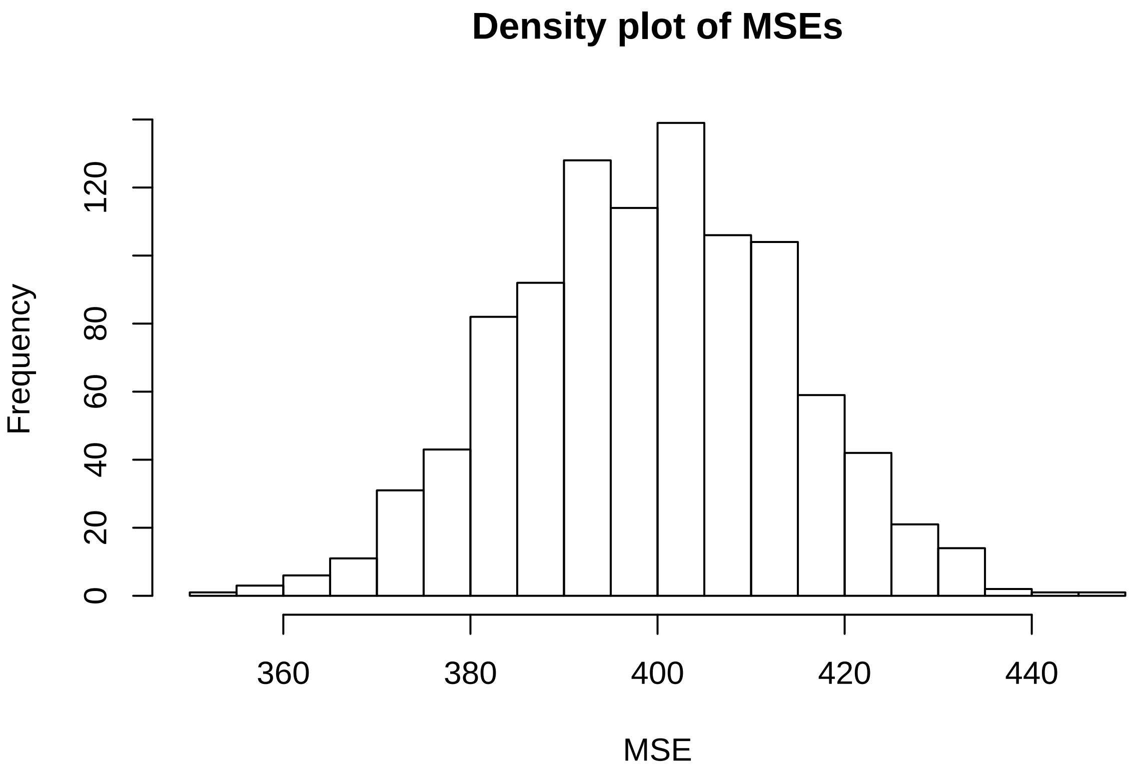
Interestingly, the mean squared error decreased from that of question 1, suggesting that our model has slightly better predictive capabilities. However, it's important to recognize that the MSE only decreased from 395 to 386, so this change isn't too dramatic. Further, this could likely be a result of the specific rows sampled, since (see Q3 below), on average we expect our trained model to generate a similar (or slightly larger) MSE than the original, entire model. Generally, however, it's not good practice to train and evaluate our model on same training dataset, since it would introduce an optimisitc bias due to overfitting.

3.

```r
i <- 1
myL <- list()
while (i<=1000){
    samples <- sample(1:nrow(myData), nrow(myData)*0.5, replace = FALSE)
    train <- myData[samples, ]
    test <- myData[-samples, ]
    myOLS <- lm(biden ~ . ,train)
    preds <- predict(myOLS, newdata = test)
    mse <- mean((test$biden - preds)^2)
```

```
    myL <- c(myL,mse)

    i<-i+1
}
hist(as.numeric(myL), breaks=25, xlab="MSE", main="Density plot of MSEs")
```



Thus, as can be seen above, the mean squared errors seem to be approximately normally distributed, with a mean of about 400. This suggests that, on average, fitting the model on only a training set performs marginally worse than fitting the model to the entire data set (but, it's unclear if this difference is statistically significant). However, even though the MSE is slightly larger in this new model, it's better practice to fit our model on only a training set (see Q2 above). Additionally, because this difference is small, it suggests that our original model (from Q1) is likely not overfitting the data significantly.

4.

```
lm_coefs <- function(splits, ...) {
  mod <- lm(..., data = analysis(splits))
  tidy(mod)
}
my_boot <- myData %>%
  bootstraps(1000) %>%
  mutate(coef = map(splits, lm_coefs, as.formula(biden~.)))

my_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)   58.8    3.08
## 2 age            0.0468 0.0300
## 3 dem           15.5    1.09
## 4 educ          -0.343  0.193
## 5 female         4.11   0.936
## 6 rep          -15.8    1.37
```

As can be seen above, the estimates for the coefficients are very similar to those obtained in question 1. The standard errors are also quite similar; some are slightly smaller in the bootstrapped model (e.g. female, education, dem), while others are slightly smaller in the original model. Because the bootstrapped method does not rely on distributional assumptions, its estimates are more robust than the estimates generated in question 1. Importantly, because the bootstrap method generates its SE estimates directly from the data, the bootstrapped 95% confidence interval includes the population mean.

Conceptual Motivation for Bootstrapping:

Boostrapping allows us to draw inferences about a population from a given sample. Specifically, by resampling with replacement, bootstrapping provides a relatively simple way to estimate standard errors and confidence intervals for estimators of a distribution, shedding light on the *variability* of the population in question. Bootstrapping is useful because we oftentimes don't have access to data about the entire population in question, but rather only a smaller sample. However, bootstrapping makes some important assumptions that aren't always met (such as the independence of samples), and thus results may need to be taken with a grain of salt; also, it can be computationally expensive. However, it is undoubtedly a useful technique in the world of machine learning.