Problem Set 2: Uncertainty, Holdouts, and Bootstrapping

Casey Mallon

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.

```
set.seed(888)
model_traditional<- glm(biden~female+age+educ+dem+rep,data=NES_data)
traditional_MSE<- mse(model_traditional,NES_data)
traditional_MSE
```

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

The mean squared error for the *entire* dataset is 395.2702, which is the mean (average) squared difference between the predicted Biden feeling thermometer values and the observed Biden feeling thermometer values. This is a fairly high MSE since a value of 0 would be a perfectly fit model. So even though this model incorporates all of the independent variables included in the NES_data, it does not efficiently estimate a voter's feelings toward Biden. In other words, even with all these variables included in the model, there is still a large amount of error.

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

- 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.

```
##Training Set:
set.seed(888)
Biden_split <- initial_split(data = NES_data,
                             prop = 0.5)
Biden_train <- training(Biden_split)
##Holdout (Testing) Set:
Biden_test <- testing(Biden_split)
```

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

```
training_model<-glm(biden~female+age+educ+dem+rep,data=Biden_train)
```

- Calculate the MSE using only the test set observations.

```
set.seed(888)
Biden_split <- initial_split(data = NES_data,
                             prop = 0.5)
Biden_train <- training(Biden_split)
Biden_test <- testing(Biden_split)
training_model<-glm(biden~female+age+educ+dem+rep,data=Biden_train)

Biden_test_mse <- augment(training_model, newdata = Biden_test)%>%
  rcfss::mse(truth = biden, estimate = .fitted)%>% select(.estimate)%>%as.numeric()
Biden_test_mse
```

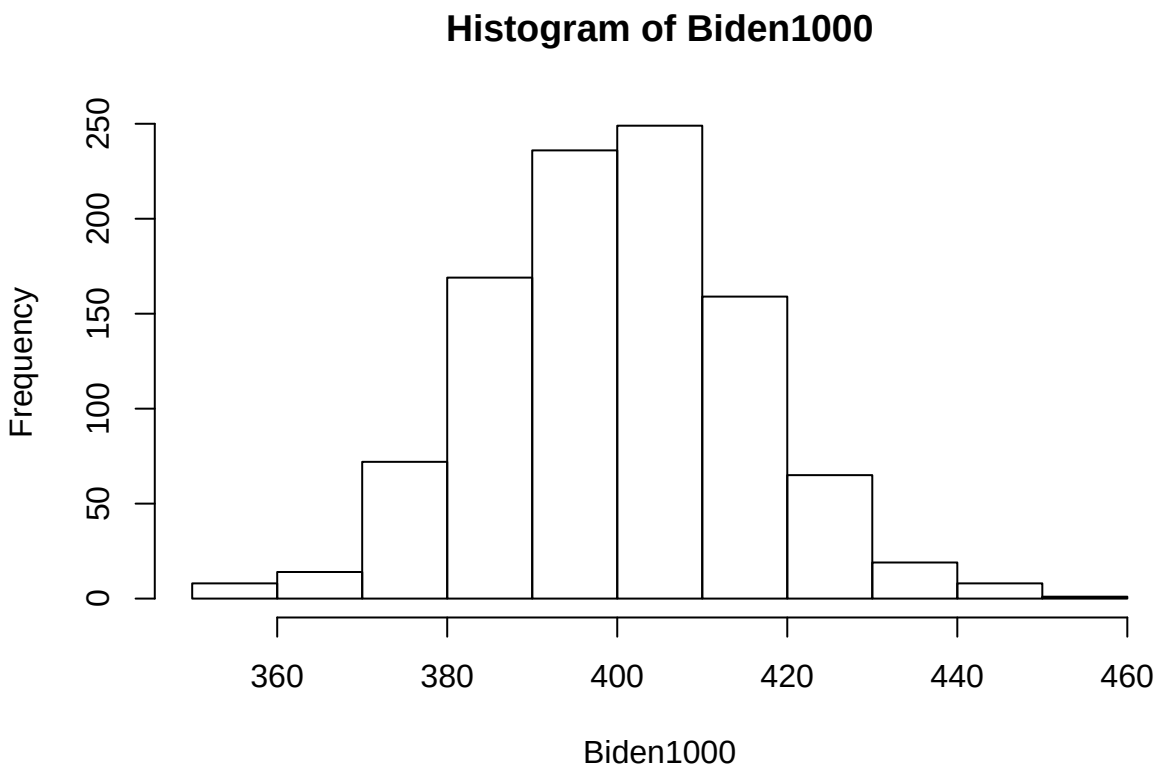
```
## [1] 401.909
```

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

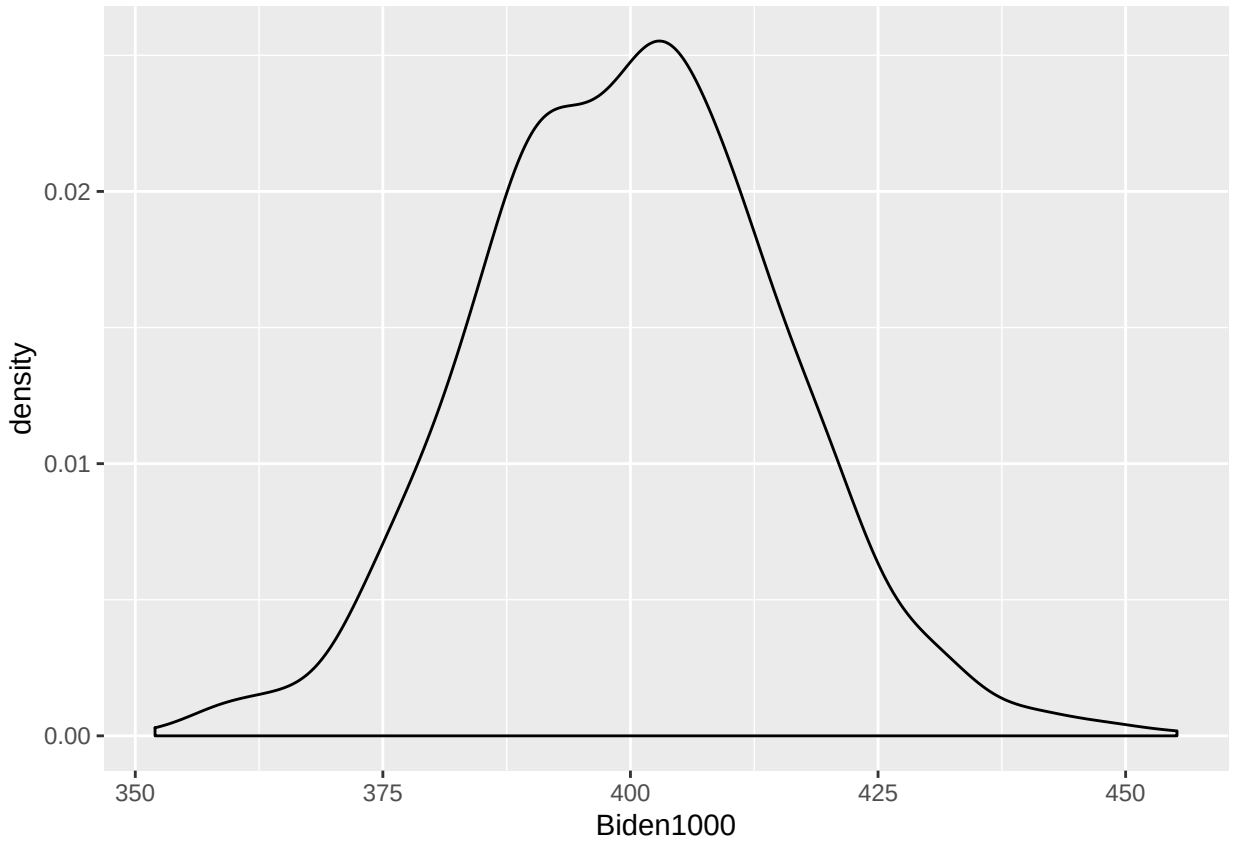
The MSE from question 1 was 395.2702 while the MSE from this model was 401.909. The two values are not far off from each other since the model based on the whole data set and the model based on a sampling of the data set use the same independent variables to explain feeling thermometer values toward biden. Additionally, the second model is a sample of the first, so it makes sense that without the other half of the data available it would produce a slightly higher MSE compared to the more wholistic first model.

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.

```
Biden1000 <- replicate(1000,{
  Biden_split_1000<- initial_split(data = NES_data,
                             prop = 0.5)
  Biden_train_1000 <- training(Biden_split_1000)
  Biden_test_1000 <- testing(Biden_split_1000)
  training_model_1000<-glm(biden~female+age+educ+dem+rep,data=Biden_train_1000)
  Biden_test_mse_1000 <- augment(training_model_1000, newdata = Biden_test_1000) %>%
  rcfss::mse(truth = biden, estimate = .fitted)%>% select(.estimate)%>%as.numeric()
})
hist(Biden1000)
```




```
Biden1000_df<- as.data.frame(Biden1000)
ggplot(Biden1000_df,aes(Biden1000))+geom_density()
```



This histogram and density plot show that the validation set approach repeated 1000 times yielded a MSE centered about 400. This indicates that the average MSE for validation sets was 400, compared to the origional model (question 1) which had an MSE of 395.2702. This shows that multiple samples came pretty closer to converging on the true MSE of the original sample we had stratified.

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.

```
##Traditional Model
model_traditional<- glm(biden~female+age+educ+dem+rep,data=NES_data)
summary.glm(model_traditional)
```

```
##
## Call:
## glm(formula = biden ~ female + age + educ + dem + rep, data = NES_data)
##
## Deviance 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
##
## (Dispersion parameter for gaussian family taken to be 396.587)
##
##     Null deviance: 994144  on 1806  degrees of freedom
## Residual deviance: 714253  on 1801  degrees of freedom
## AIC: 15947
##
## Number of Fisher Scoring iterations: 2
```

```
##Bootstrap
lm_coefs <- function(splits, ...) {
  mod <- lm(..., data = analysis(splits))
  tidy(mod)
}
Biden_boot<- NES_data%>%
  as_tibble()%>%
  bootstraps(1000)%>%mutate(coef = map(splits, lm_coefs, as.formula(biden ~ female+age+educ+dem+rep)))

q4results<- Biden_boot%>%unnest(coef) %>%
  group_by(term) %>% summarize(.estimate = mean(estimate),
            .se = sd(estimate, na.rm = TRUE))%>%rename(estimate=.estimate, se=.se)
q4results
```

```
## # A tibble: 6 x 3
##   term        estimate     se
##   <chr>          <dbl>  <dbl>
## 1 (Intercept)  58.8    3.01
## 2 age           0.0477 0.0281
## 3 dem          15.4    1.09
## 4 educ         -0.344  0.188
## 5 female        4.09   0.926
## 6 rep         -15.9    1.40
```

The bootstrap model yields greater estimates for the overall Intercept estimate, age, Democrat, and education (greater in terms of absolute value). The bootstrap model also gives greater standard errors for Republican and Democrat, but interestingly provides lower standard errors than the traditional model for all other terms, including the intercept. Overall, the bootstrapped estimates of parameters are virtually identical, however the standard errors on the bootstrap estimates are slightly smaller.

Bootstrapping allows us to turn one dataset into several simulated samples so that we can perform analyses that would normally require us to have knowledge of the true population parameters of the original dataset, which are unknown in reality. The bootstrap method is not biased by distributional assumptions, unlike the traditional method, and can give us more robust estimates. Bootstrapping, however, is problematic when the statistic of interest is a maximum/minimum because the bootstrapped distribution will not converge on the true value of the statistic of interest as the number of bootstrap iterations increases. Additionally, bootstrapping is problematic when the sample sie is small.