Problem Set 2

Rei Bertoldi

2/2/2020

```
cal_mse <- function(df, new_df){
  reg_model <- glm(biden ~ female + age + educ + dem + rep, data = df)
  augment(reg_model, newdata = new_df) %>%
    mse(truth = biden, estimate = .fitted)
}
```

```
set.seed(10)
nes_split <- initial_split(data = nes,
                           prop = 0.5)
nes_train <- training(nes_split)
nes_test <- testing(nes_split)
```

Question 1

```
cal_mse(nes, nes)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        395.
```

The MSE is pretty large, 395. This number represents the squared difference between the estimated values and the actual value, in other words is illustrates the divergence from the fit line and our data points.

Question 2

```
cal_mse(nes_train, nes_test)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        401.
```

The MSE is larger than question one, 401. This MSE probably went us because we split the data, so our sample size is smaller and our bias is larger.

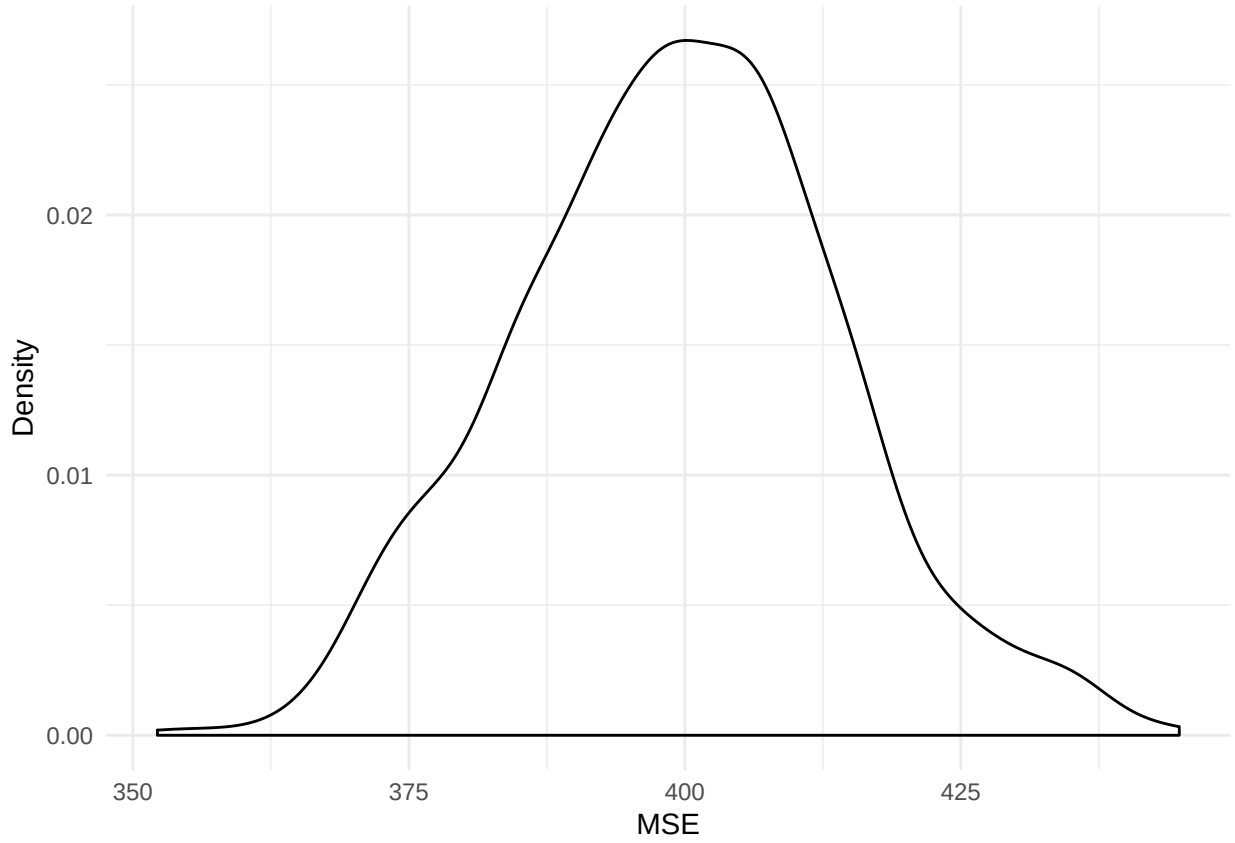
Question 3

```
set.seed(10)
mse_split <- function() {
  nes_split <- initial_split(data = nes,
                             prop = 0.5)
  nes_train <- training(nes_split)
```

```
  nes_test <- testing(nes_split)
  cal_mse(nes_train, nes_test)[[3]]
}

rep_results <- data.frame(replicate(n = 1000, mse_split()))
```

```
ggplot(rep_results) +
  geom_density(aes(x = `replicate.n...1000..mse_split...`)) +
  labs(x = "MSE",
       y = "Density") +
  theme_minimal()
```



The density graph represents the distribution of MSE from splitting, regressing and estimating the MSE 1000 times. The MSE estimates are centered around 400.

Question 4

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

```
lm_coefs <- function(nes, ...) {
  ## use `analysis` or `as.data.frame` to get the analysis data
  mod <- lm(..., data = analysis(nes))
  tidy(mod)
}
```

```
biden_boot <- nes %>%
  bootstraps(1000) %>%
```

```
  mutate(coef = map(splits, lm_coefs, as.formula(biden ~ female + age + educ + dem + rep)))

biden_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    3.11
## 2 age            0.0481 0.0288
## 3 dem           15.4    1.08
## 4 educ          -0.345  0.196
## 5 female         4.09   0.901
## 6 rep          -15.9    1.39
```

```
tidy(reg_results)
```

```
## # 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
```

The bootstrapped estimates and the true parameter estimates very similar, the differences being the tenths places. The standard errors are slightly larger, which is due to the fact that the bootstrapped estimates do not reply on any distributional assumption.