Problem Set 2 - Runhua Li

Runhua Li

2/1/2020

```
rm(list = ls())
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

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

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

```
library(ISLR)
library(broom)
library(rsample)
library(rcfss)
library(yardstick)
```

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

```
nes08 <- read.csv("/Users/RunhuaLi/R/nes2008.csv")
set.seed(4321)
```

1. The Traditional Approach

In this section I will fit a linear regression model using the entire dataset, and then calculate the MSE of it using the entire dataset.

```
lmod1 <- lm(biden ~ female + age + educ + dem + rep, data = nes08)
summary(lmod1)
```

```
##
## Call:
## lm(formula = biden ~ female + age + educ + dem + rep, data = nes08)
##
## 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
```

```
entire_mse <- augment(lmod1, newdata = nes08) %>%
  mse(truth = biden, estimate = .fitted)
entire_mse
```

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

The linear model coefficients suggest that females and democrates tend to like Biden more, while republicans tend to dislike Biden. The coefficients of age and education are not statistically significant on the level of 0.05.

The MSE is 395.2702. Note that this MSE is calculated using the entire dataset, which is also used in training the linear model, i.e., the linear model is already fitted on the same dataset used to estimate MSE. Therefore this MSE does not tell us how this model performs on other data.

2. Simple Holdout

```
nes08_split <- initial_split(data = nes08, prop = 0.5)

nes08_training <- training(nes08_split)
nes08_test <- testing(nes08_split)

lmod2 <- lm(biden ~ female + age + educ + dem + rep, data = nes08_training)
summary(lmod2)
```

```
##
## Call:
## lm(formula = biden ~ female + age + educ + dem + rep, data = nes08_training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -70.375 -11.115  -0.118  12.258  54.761
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  60.59545    4.37689  13.844  < 2e-16 ***
## female        3.35211    1.32800   2.524   0.0118 *
## age           0.01860    0.03978   0.467   0.6403
## educ         -0.43340    0.27829  -1.557   0.1197
```

```
## dem          16.11853    1.48128  10.882  < 2e-16 ***
## rep         -15.40452    1.86697  -8.251 5.56e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.72 on 898 degrees of freedom
## Multiple R-squared:  0.2818, Adjusted R-squared:  0.2778
## F-statistic: 70.47 on 5 and 898 DF,  p-value: < 2.2e-16
```

```
test_mse <- augment(lmod2, newdata = nes08_test) %>%
  mse(truth = biden, estimate = .fitted)
test_mse
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 mse     standard        406.
```

The value of MSE in question 2 (386.0989) is smaller than that in question 1 (395.2702). Note that this is not always true. It depends on the split between training dataset and testing dataset. For example, splitting the datasets with different seeds generates different results. The following question also demonstrates this point.

However, there's something we can say about this MSE and the previous one. Conceptually, this MSE tend to be smaller. The reason is, this MSE is calculated using the test dataset and estimates generated from a linear model that is not fitted to the same test dataset. In contrast, the previous MSE is calculated using a dataset already used to fit the model. As herr the process of fitting a model is exactly to minimze squared error (OLS), the previous MSE tend to be smaller.

3. Monte Carlo Cross Validation

```
mse_MC <- 1

for(i in 1:1000) {
  nes08_MC_split <- initial_split(nes08)
  nes08_MC_train <- training(nes08_MC_split)
  nes08_MC_test <- testing(nes08_MC_split)

  lmod_MC <- lm(biden ~ female + age + educ + dem + rep,
                data = nes08_MC_train)

  mse <- augment(lmod_MC, newdata = nes08_MC_test) %>%
    mse(truth = biden, estimate = .fitted)
  mse_MC <- c(mse_MC, mse$.estimate)
}

mse_MC <- mse_MC[-1]

mean(mse_MC)
```

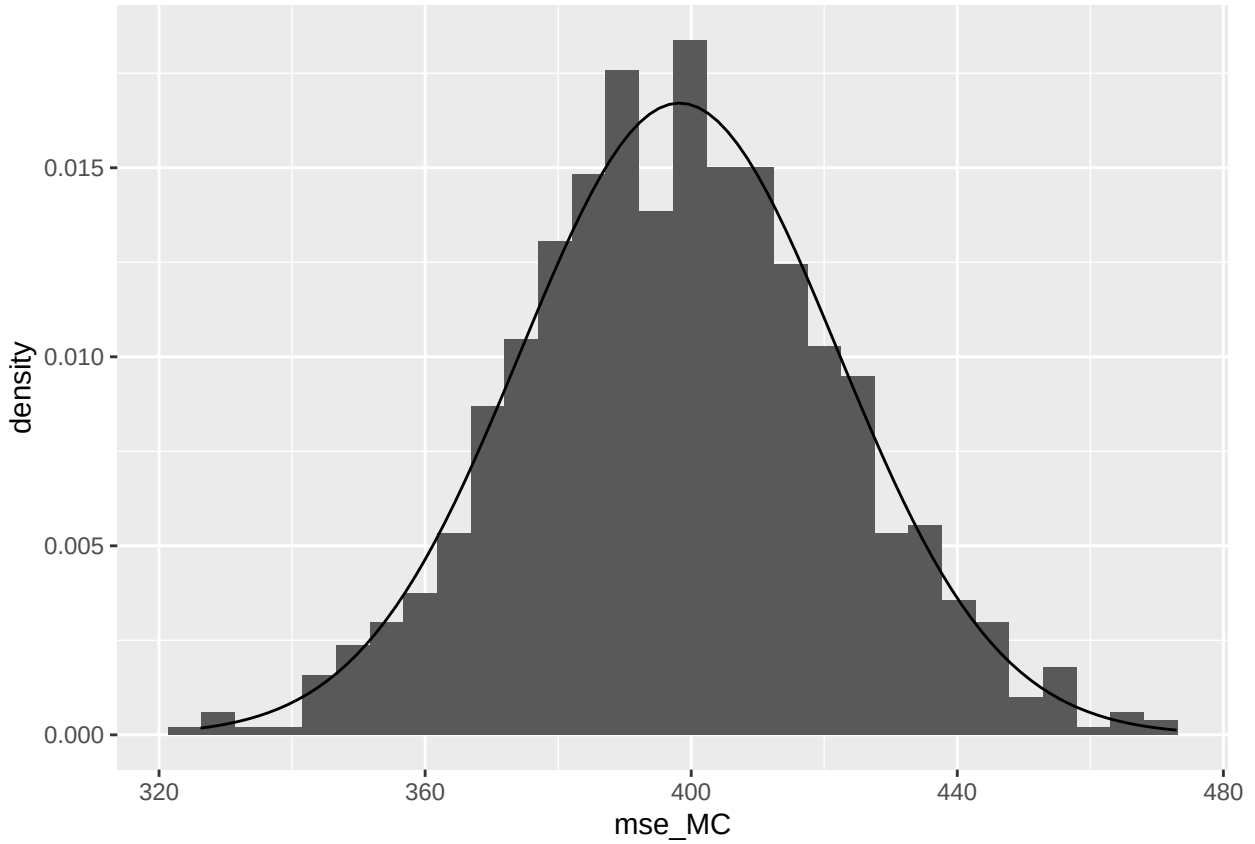
```
## [1] 398.2225
```

```
sd(mse_MC)
```

```
## [1] 23.87378
```

```
ggplot(mapping = aes(x = mse_MC)) +
  geom_histogram(aes(y = ..density..)) +
  stat_function(fun = dnorm, args = list(mean = mean(mse_MC), sd = sd(mse_MC)))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The mean of the 1000 MSE obtained from this Monte Carlo Cross Validation (repeated holdouts) is 398.013, which is larger than the MSE in question 2 (386.0989). The 1000 MSEs demonstrates a distribution similar to a normal distribution. This mean of MSE is a better estimate of how well the model performs on a randome test dataset, whereas the simple holdout approach is largely affected by the arbitrary split between training and test dataset.

Note that this mean of MSE is larger than the MSE calculated using the entire dataset. This coincides with the conceptual discussion in question 2.

4. Bootstrap and Comparison

```
coef <- function(splits) {
  x <- analysis(splits)
  lmod_boot <- lm(biden ~ female + age + educ + dem + rep, data = x)
  tidy(lmod_boot)
}

nes08_boot <- nes08 %>%
  bootstraps(1000) %>%
  mutate(coef = map(splits, coef))
```

```
nes08_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(.estimate = mean(estimate),
            .se= sd(estimate), na.rm = TRUE)
```

```
## # A tibble: 6 x 4
##   term        .estimate    .se na.rm
##   <chr>           <dbl>  <dbl> <lgl>
## 1 (Intercept)   58.7    3.03   TRUE
## 2 age            0.0485 0.0282 TRUE
## 3 dem           15.4    1.05   TRUE
## 4 educ          -0.338  0.193  TRUE
## 5 female         4.11   0.944  TRUE
## 6 rep          -15.9    1.37   TRUE
```

```
summary(lmod1)
```

```
##
## Call:
## lm(formula = biden ~ female + age + educ + dem + rep, data = nes08)
##
## 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
```

The estimated coefficients using the bootstrap method and OLS are very close, so are the standard error estimates. See a numeric comparison:

```
coef_boot <- nes08_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(.estimate = mean(estimate))

coef_boot <- c(coef_boot$.estimate[1],
               coef_boot$.estimate[5],
               coef_boot$.estimate[2],
               coef_boot$.estimate[4],
               coef_boot$.estimate[3],
               coef_boot$.estimate[6])
```

```
coef_diff <- lmod1$coefficients - coef_boot

sd_boot <- nes08_boot %>%
  unnest(coef) %>%
  group_by(term) %>%
  summarize(.se= sd(estimate))

sd_boot <- c(sd_boot$.se[1],
             sd_boot$.se[5],
             sd_boot$.se[2],
             sd_boot$.se[4],
             sd_boot$.se[3],
             sd_boot$.se[6])

sd_lm <- summary.lm(lmod1)
sd_lm <- sd_lm$coefficients

sd_diff <- sd_lm[,2] - sd_boot

print("difference in coefficients")
```

```
## [1] "difference in coefficients"
```

```
coef_diff
```

```
##   (Intercept)        female           age          educ           dem
##  0.0914334198 -0.0052558494 -0.0002250749 -0.0077444541  0.0309637959
##           rep
##  0.0122681823
```

```
print("difference in standard errors")
```

```
## [1] "difference in standard errors"
```

```
sd_diff
```

```
##   (Intercept)        female           age          educ           dem
##  9.262268e-02  3.955254e-03  7.722507e-05  1.732712e-03  2.294665e-02
##           rep
## -5.628088e-02
```

Standard error of coefficients of age, education and female estimated using bootstrap are smaller than OLS estimates. The opposite is true for coefficients of democrats and republics.

Using bootstrap free us from assuming a distribution of the population. This is especially useful when we don't know the true population distribution and the size of existing datasets are not very large.

The problem of making assumption of the population distribution is that, when the assumption is wrong, some sample statistics might not be a good estimate of the population statistics. For example, the population mean might lie outside the sample mean's confidence interval (say 95%). In this case, bootstrap estimates of the population statistics can be a safer option.

In this specific case, the OLS assumption of population distribution (normality) is probably not inaccurate, because the estimated parameters and standard errors using the OLS and bootstrap are very close.