# Homework 1: Bias, Variance, & Resampling

MACS 30100: Perspectives on Computational Modeling
University of Chicago

Betsy Priem

## Overview

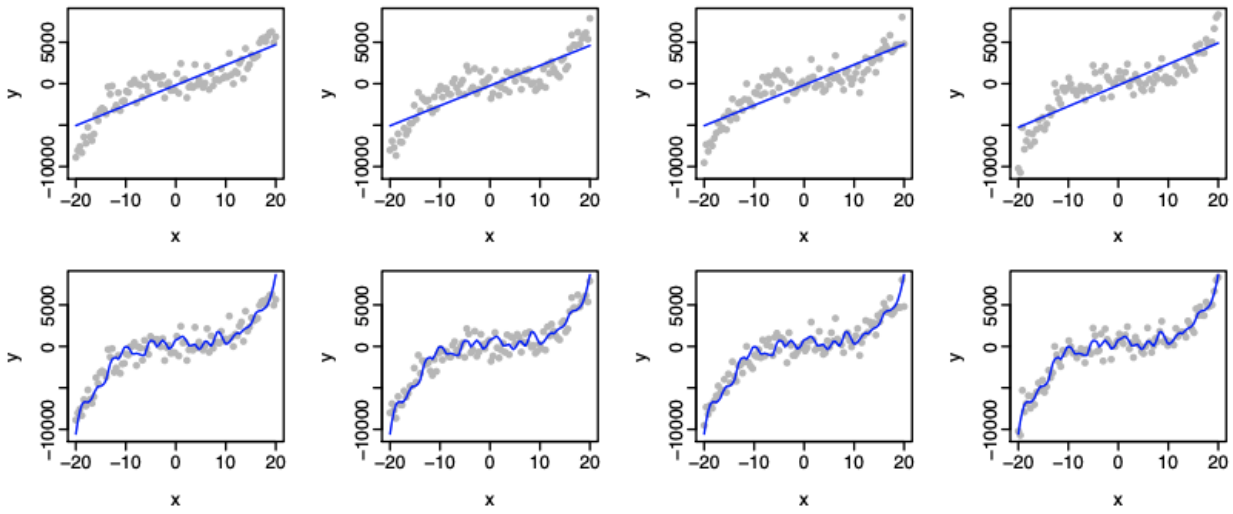
For each of the following prompts, produce responses *with* code in-line. While you are encouraged to stage and draft your problem set solutions using any files, code, and data you'd like within the private repo for the assignment, *only the final, rendered PDF with responses and code in-line will be graded.*

## Bias & Variance

1. (10 points) Consider the following eight plots based on model fits, assuming the data generating process,

$$y = x^3 - 2x^2 + 1.5x + \epsilon.$$

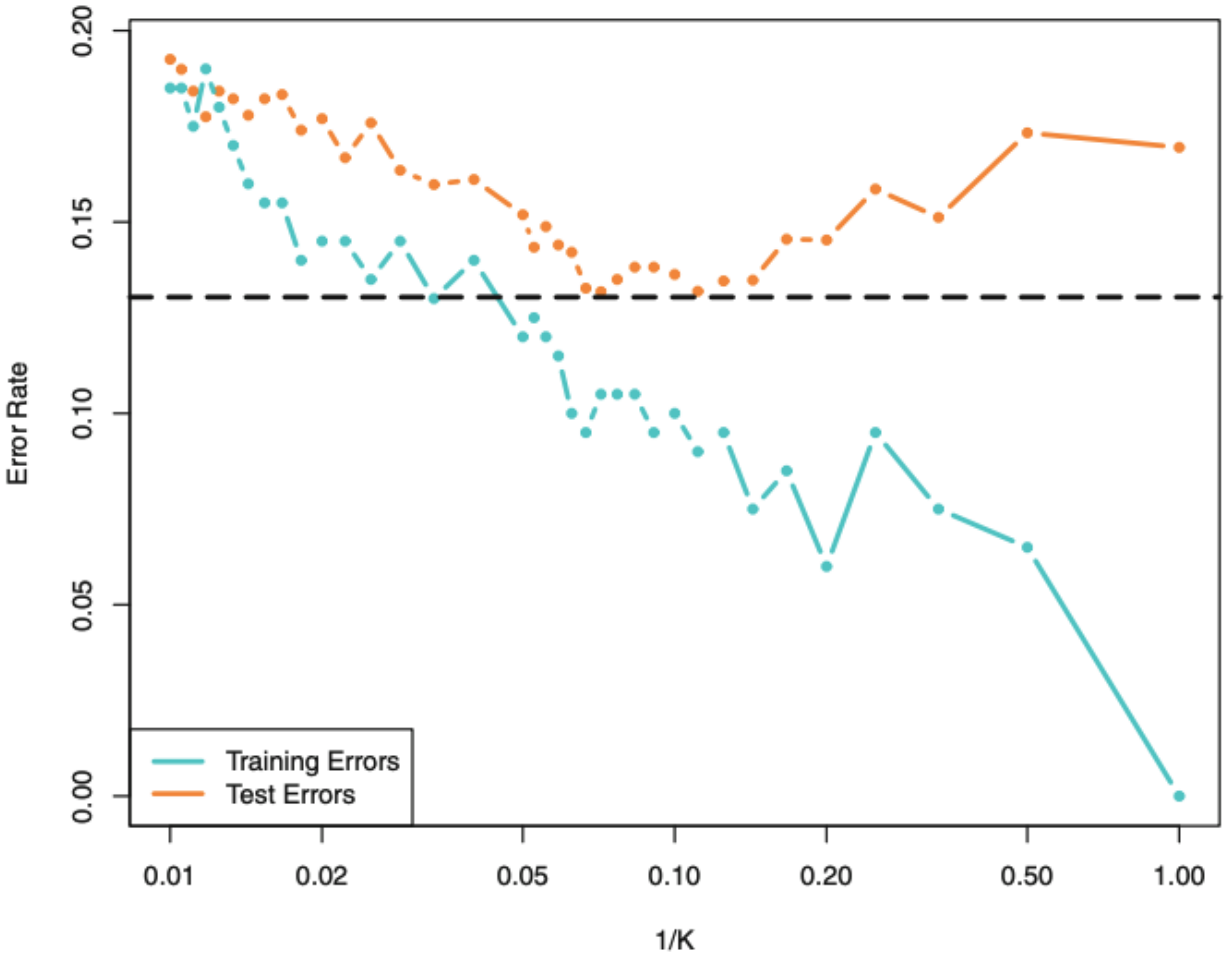
Specifically, the figure shows two different fits of a model (one for each row) on each of the four samples (one for each column). So, e.g., column 1 shows two versions of a model fit to the same sample of 100 observations, which were drawn at random from the data generating process in the equation above. *Describe the difference between the two models in terms of complexity, bias, and variance. Responses should be at least a few sentences.*




**The top row shows a linear model (less complexity) that has higher bias and lower variance. This means that each observation will likely deviate more from the model (representing the *higher bias* and mistakes in the linear model's ability to make accurate predictions), while the linear model fit will likely make more consistent predictions across samples because the error the model generates will not be as reactive to different distributions of data (i.e., the *lower***

***variance*). The bottom row illustrates a more complex, nonlinear model fit of the data. The trade-off with using this more complex modeling is that while bias (mistakes in predictions) is much lower, variance is far higher compared to the linear, less complex models - making it more error prone against different random samples.**

2. (10 points) Building on the previous question and considering the following figure from ch. 2 of the ISL book, think about training and test error moving from a less flexible model towards more a more flexible model. Specifically, the figure is becoming more "flexible" as the number of neighbors, $k$, decreases, thereby picking up more local behavior. We haven't yet covered kNN or other supervised classifiers, but the logic of the figure should be apparent, where the level of flexibility of a model will directly influence training and testing error. *Explain why these two curves have the shapes they do. Responses should be at least a few sentences.*




**The training curve shows a decline as the flexibility increases, which makes sense because the increased flexibility means the model will more closely fit the data (decreasing the error rate). The test curve is U-shaped because with high restrictions (i.e., the least flexible model) the model will be highly inaccurate and have a higher error rate. At the optimal level of flexibility (in this case at around k=10), the model will make the fewest mistakes and generate a minimal error rate, representing the dip in the U-shape. The right-hand part of the U-shape indicates that with increasing flexibility this method generates a model that will overfit the data and become much more prone to error (signaled by the increase in the error rate).**

3. (10 points) When the sample size, $n$, is very large, and the number of predictors, $p$, is small, would we expect the performance of a flexible model to be better or worse than an inflexible method? Justify your answer.

**With a large sample size and small number of predictors you would expect the flexible model to do better than an inflexible model. This is because the flexible model would fit the data better and have a lower MSE. An inflexible model (like a linear regression) would likely be highly biased because fitting a straight line through a large sample that likely is more heterogeneous than a sample with a smaller $n$ will result in creating a fit that has observations that are far from that line.**

4. (10 points) When the number of predictors, $p$ is very large, and the sample size, $n$, is small, would we expect the performance of a flexible model to be better or worse than an inflexible method? Justify your answer.

**In this case, the flexible model would perform worse compared to the inflexible method. The flexible method would likely overfit the data because of the greater number of predictors and small sample size. In trying to use all the predictors to explain the relationship between so few observations, the complex model would likely model associations that have no real bearing on the "real-world associations" (i.e., it would *overfit* the data). An inflexible model would stay truer to the expected "real-world associations" because it would be less sensitive to the higher number of predictors than the flexible method.**

5. (10 points) When the relationship between the predictors, $\mathbf{X}$, and response, $y$, is highly non-linear, would we expect the performance of a flexible model to be better or worse than an inflexible method? Justify your answer.

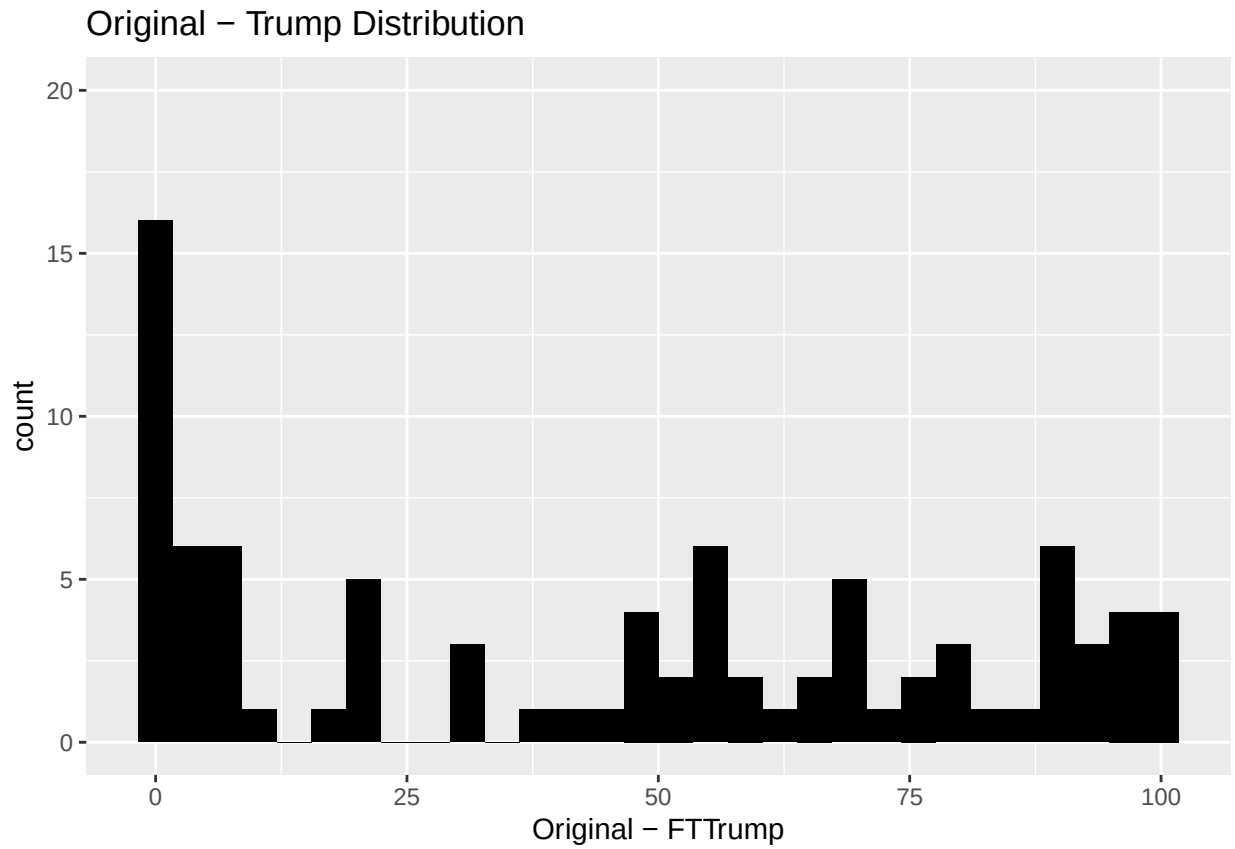
**With a non-linear relationship between predictors and the response, a flexible model would be better. While the flexible method would complicate interpretability, it would allow for better prediction because of the non-linear relationship between predictors and the response. However, there is a danger of overfitting the data with the flexible model, in which case the inflexible model would offer both easier an easier to interpret estimate of $f$ and more accurate predictions.**

6. (10 points) Why can minimizing the training mean squared error (MSE) lead to overfitting? *Responses should be at least a few sentences.*

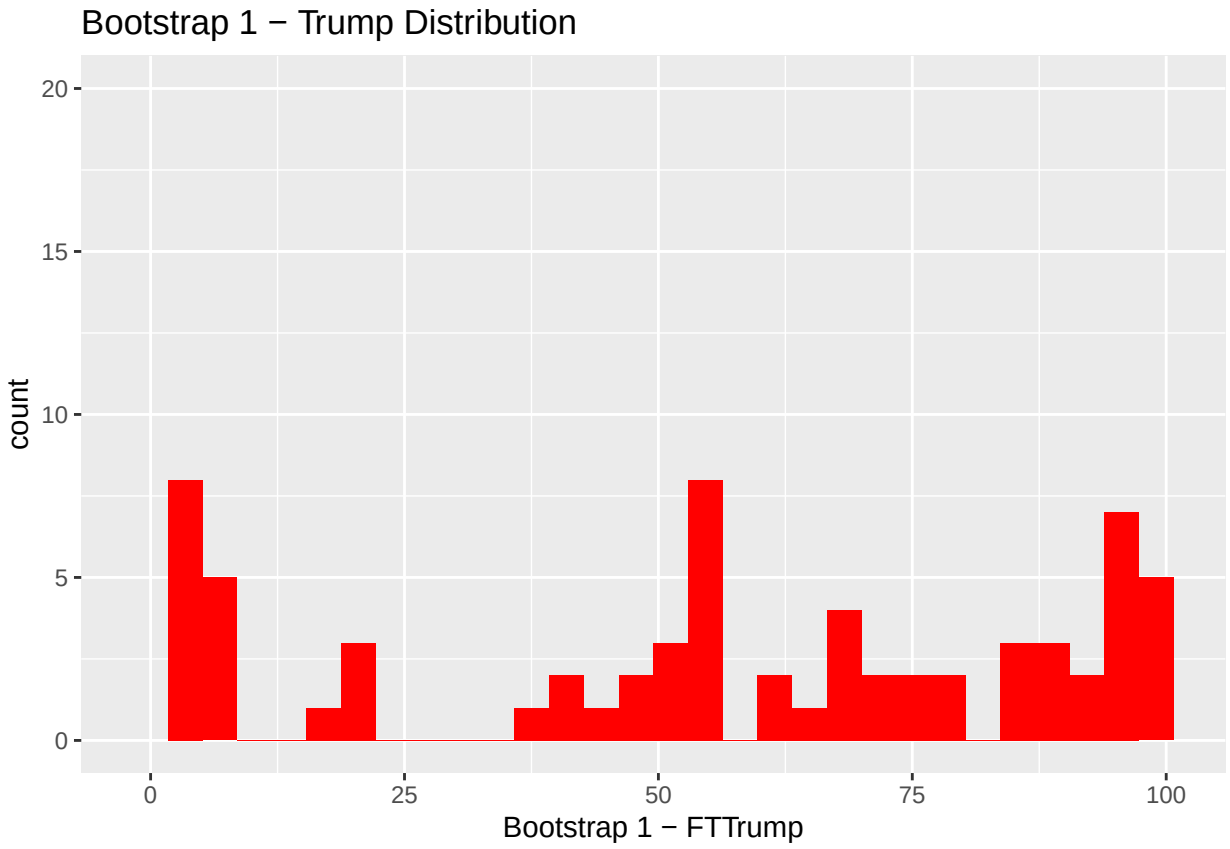
**You can minimize the training MSE by increasing the model flexibility, which means you can closely fit the model to the observations in the training sample. But when applying this estimated function with the minimized training MSE to the test set, the function might be tuned too specifically to the unique characteristics of the training set (i.e., "overfit") and it will generate a much larger *test* MSE than the smaller *training* MSE. Basically, the model only does well with the training set and because it too closely models those specific observations (*overfitting*) it can no longer do well in samples that are different from the training set.**

7. (10 points) Recall bootstrapping involves a process of drawing random samples of size $n$ through sampling *with* replacement. Using the 2016 ANES data we've used a few times already, create and plot two bootstrapped samples manually (i.e., *not using a function like* `boot()` ). For reference, also plot the original data set to compare distributions of feelings toward Trump (`fttrump`) versus feelings toward Obama (`ftobama`). *Note:* When loading the ANES data, you may simply drop the `NA`s, rather than impute, for ease (though in practice, this strategy isn't recommended).

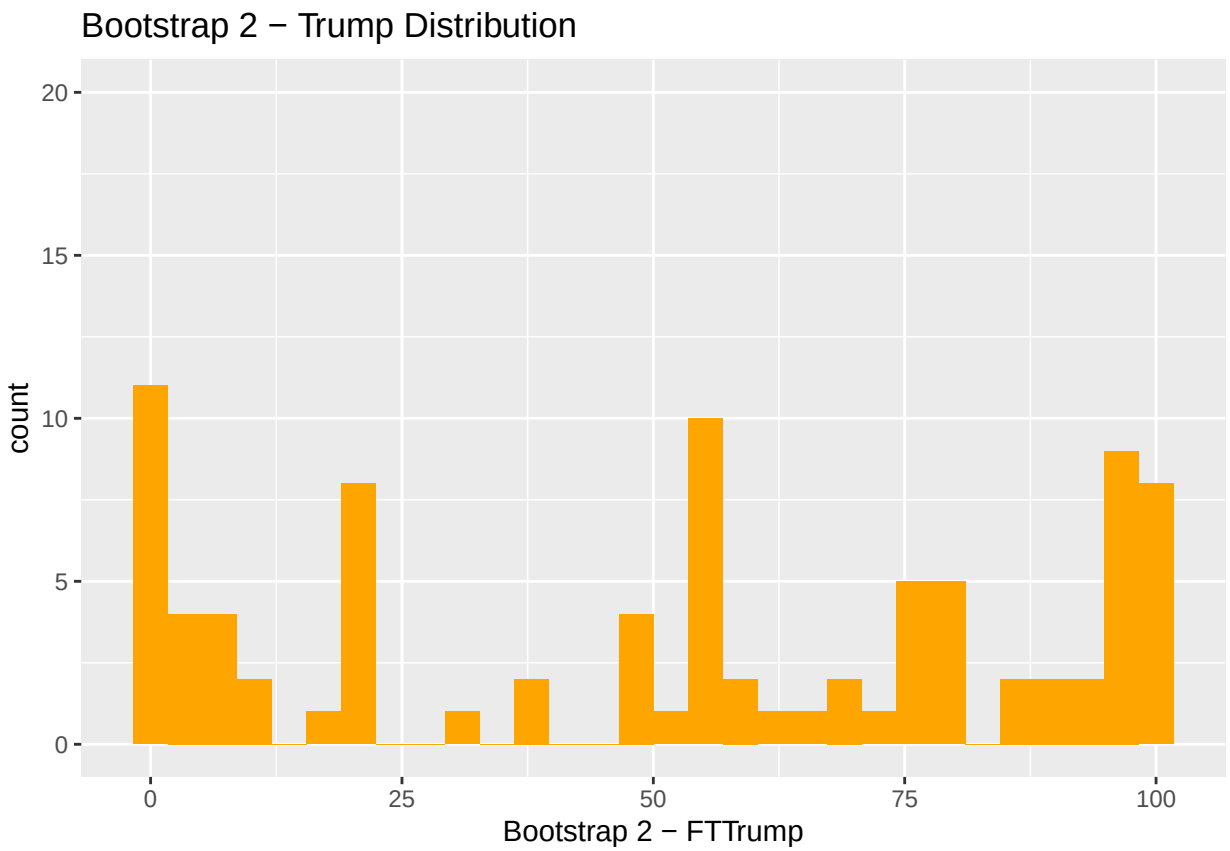
```r
library(tidyverse)
library(here)
library(ggplot2)

anes <- read_csv(here("data", "anes_pilot_2016.csv")) %>%
  tibble::as_tibble() %>%
  na.omit() %>%
  select(fttrump, ftobama)

#creating two bootstrapped samples without using 'boot' function
b1trump <- sample(anes$fttrump, replace = T)
b1obama <- sample(anes$ftobama, replace = T)
b1 <- data.frame(b1trump, b1obama)
b2trump <- sample(anes$fttrump, replace = T)
b2obama <- sample(anes$ftobama, replace = T)
b2 <- data.frame(b2trump, b2obama)

#plotting trump
ggplot(anes, aes(fttrump)) +
  geom_histogram(fill = "black")+
  labs(x= "Original - FTTrump",
       title = "Original - Trump Distribution")+
  ylim(0,20)
```



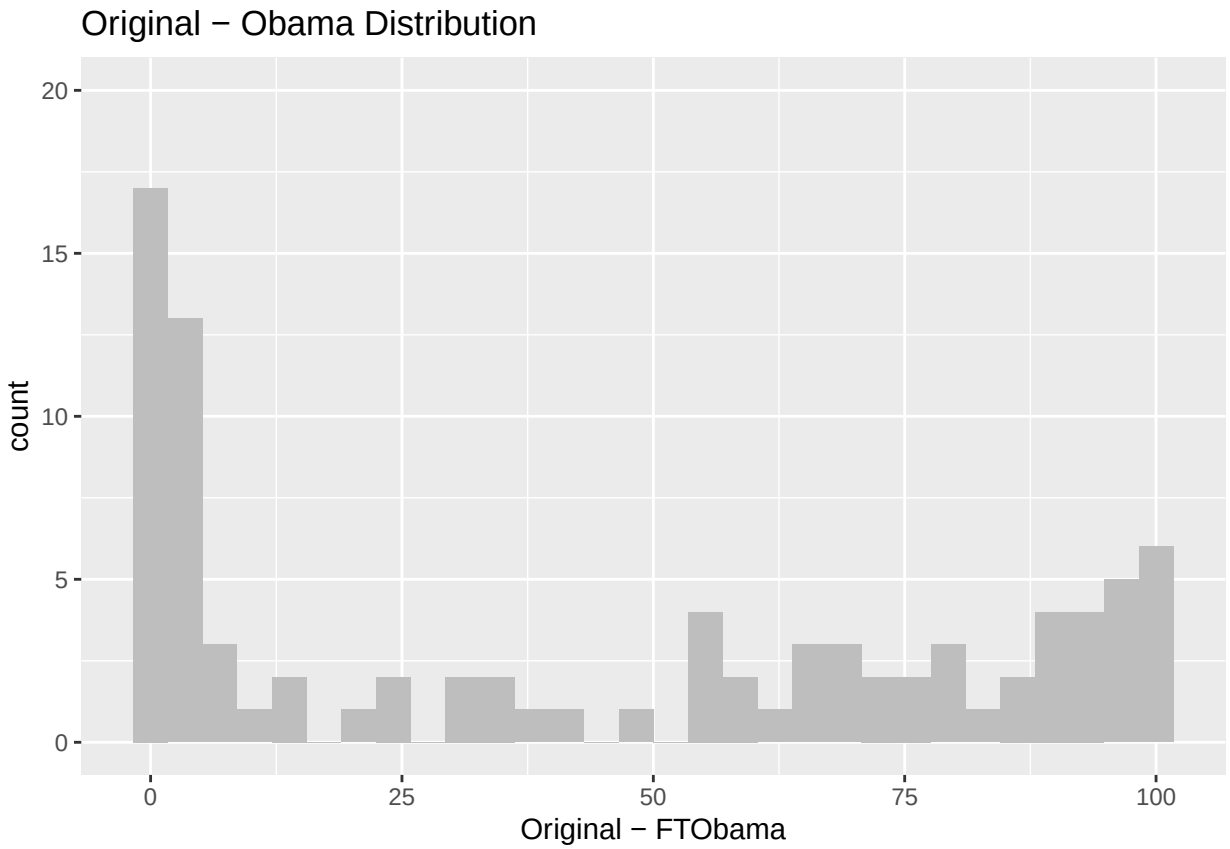
```
ggplot(b1, aes(b1trump)) +
  geom_histogram(fill = "red")+
  labs(x= "Bootstrap 1 - FTTrump",
       title = "Bootstrap 1 - Trump Distribution")+
  ylim(0,20)
```



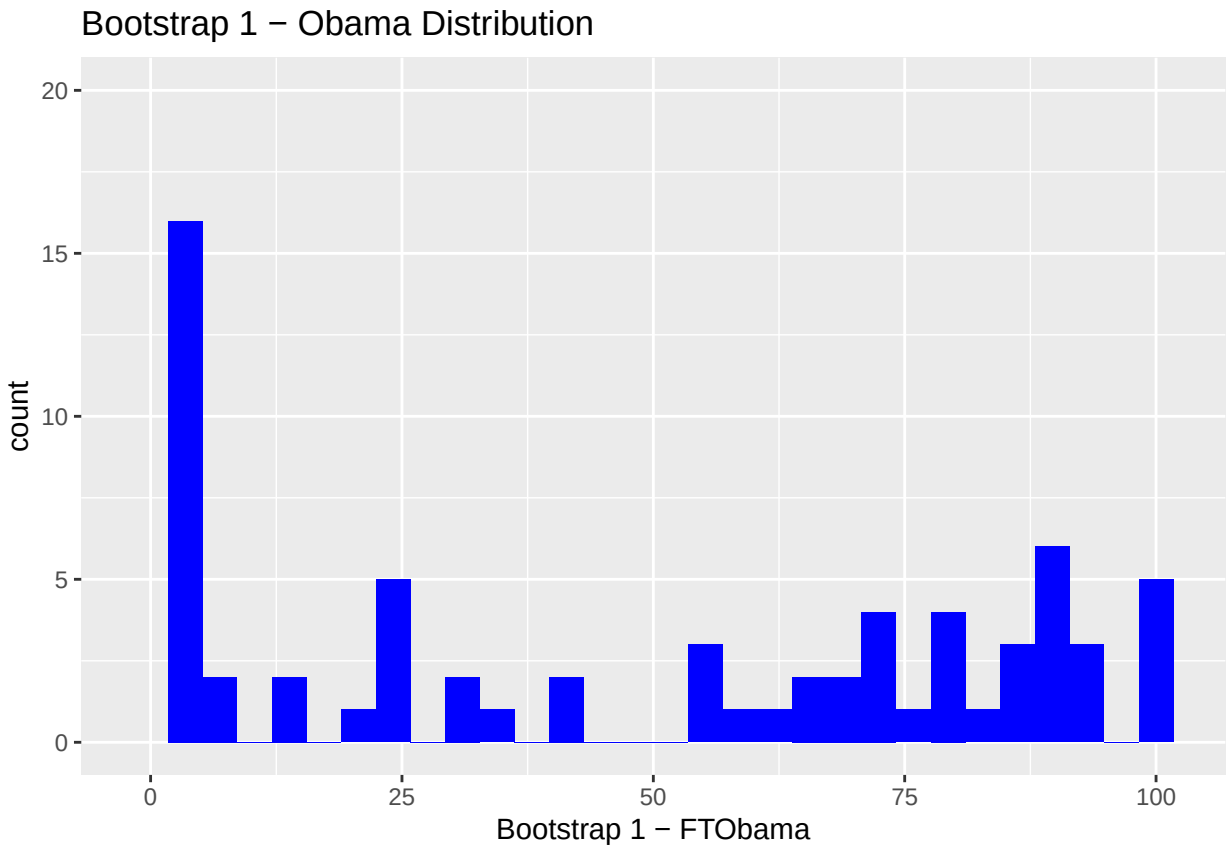

```
ggplot(b2, aes(b2trump)) +
  geom_histogram(fill = "orange")+
  labs(x= "Bootstrap 2 - FTTrump",
       title = "Bootstrap 2 - Trump Distribution")+
  ylim(0,20)
```



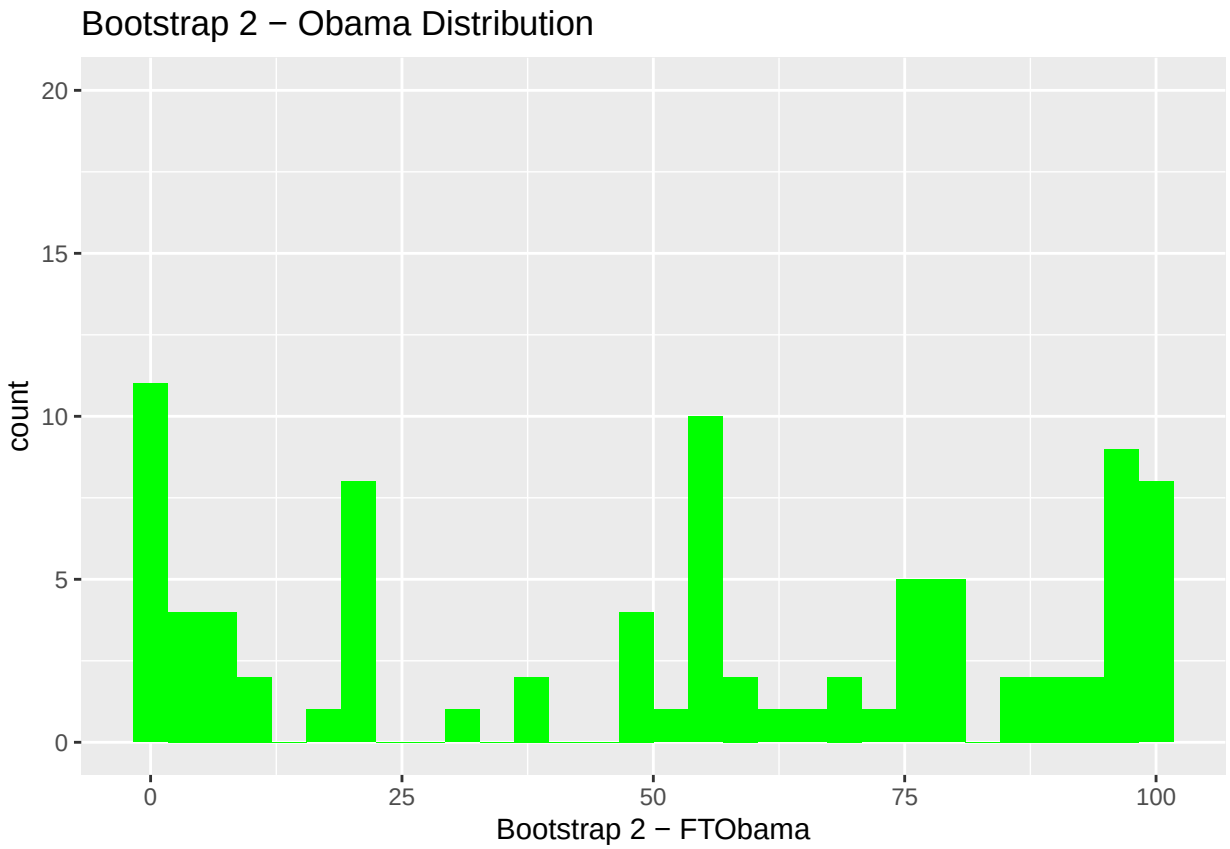

```
#plotting obama
ggplot(anes, aes(ftobama)) +
  geom_histogram(fill = "gray")+
  labs(x= "Original - FTObama",
       title = "Original - Obama Distribution")+
  ylim(0,20)
```




```
ggplot(b1, aes(b1obama)) +
  geom_histogram(fill = "blue")+
  labs(x= "Bootstrap 1 - FTObama",
       title = "Bootstrap 1 - Obama Distribution")+
  ylim(0,20)
```




```
ggplot(b2, aes(b2trump)) +
  geom_histogram(fill = "green")+
  labs(x= "Bootstrap 2 - FTObama",
       title = "Bootstrap 2 - Obama Distribution")+
  ylim(0,20)
```

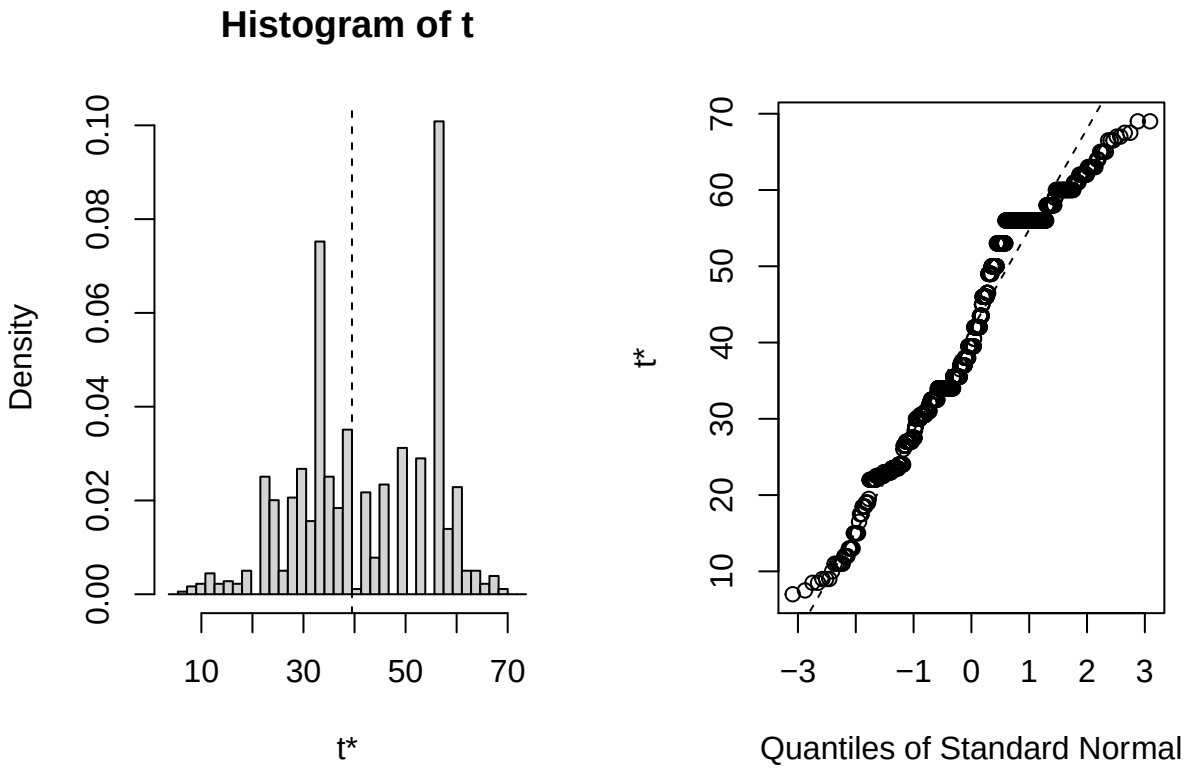



8. (5 points) Discuss the distributions from the previous question. Do they look mostly similar as is expected by sampling with replacement? Why or why not, do you think? *Respond with a few sentences.*

**They look mostly similar because the bootstrap samples are sampling from the original distribution in a small $n$ sample. This means that if in the orginal ANES sample of only 88 observations many people had low (near 0) feelings towards the presidents then when taking a bootstapped sample with replacement there is greater probability of generating the bootstrapped sample observation with that value. By the same logic, the values for each variables that had few people with that level of feeling in the original set will also have a smaller chance of being selected in the bootstrapped samples. This means the bootstrapped samples' distributions will closely resemble the original sample's distribution.**

9. (15 points) The median for feelings toward Obama (`ftobama`) is 39.5. Using a package or any function/method you'd like, bootstrap the standard error of our statistic of interest (which is the median in this case) based on 1000 draws from the data. You might consider writing a simple helper function to speed along the bootstrapping process, but this is up to you of course. Then, construct the 95% confidence interval around your bootstrapped estimate. Report your results and offer a few points of discussion. *Respond with a few sentences.*

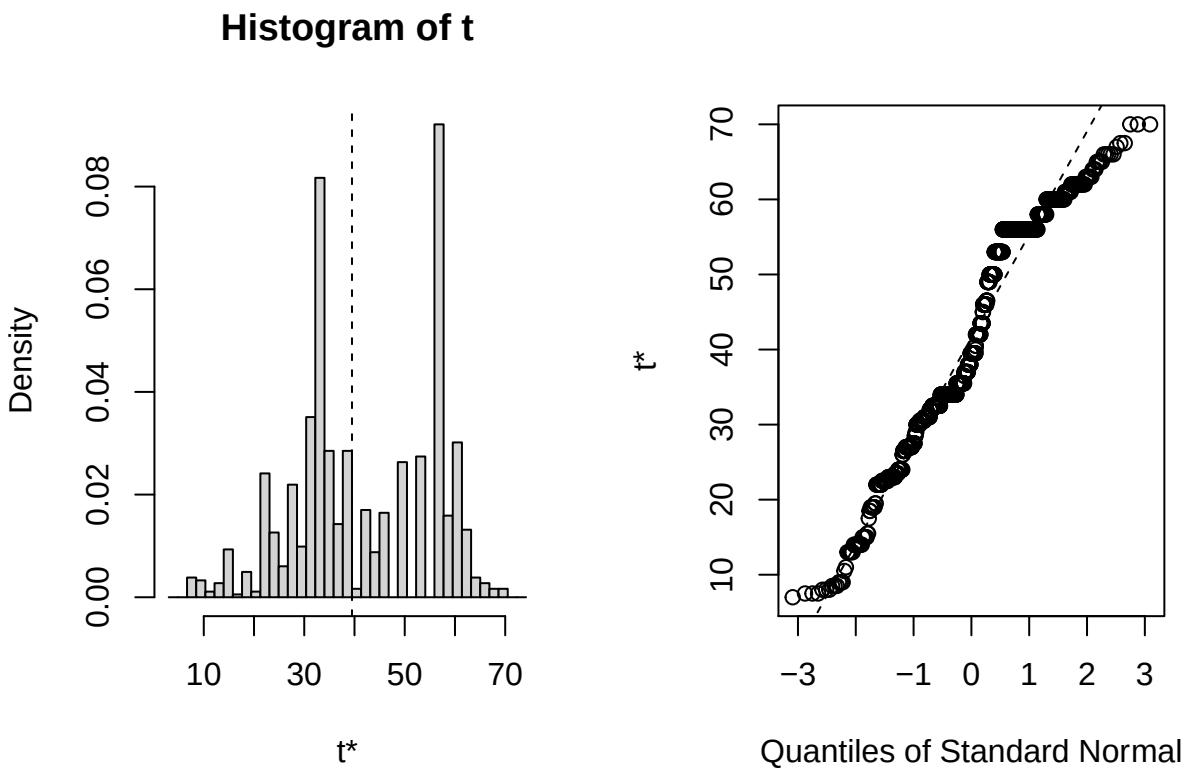
```
library(boot)

set.seed(1016)
medFunc <- function(x, i){median(x[i])} #helper function to find median of 1000 bootstrapped samples
bootMed <- boot(anes$ftobama, medFunc, 1000) #bootstrapped sample of 1000 medians
plot(bootMed) # visual check to see if this makes sense
```




```
boot.ci(bootMed)
```

```
## Warning in boot.ci(bootMed): bootstrap variances needed for studentized
## intervals

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootMed)
##
## Intervals :
## Level      Normal              Basic
## 95%   (11.51, 63.26 )   (17.00, 63.96 )
##
## Level     Percentile            BCa
## 95%   (15.04, 62.00 )   (13.00, 61.00 )
## Calculations and Intervals on Original Scale
```

```
#trying another way to code the same thing to see if I did it right
bootMed2 <- boot(anes$ftobama, statistic = function(x, i) median(x[i]), 1000)
plot(bootMed2)
```




```
bootMed2
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = anes$ftobama, statistic = function(x, i) median(x[i]),
##     R = 1000)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*     39.5  2.0685    13.74412
```

```
boot.ci(bootMed2)
```

```
## Warning in boot.ci(bootMed2): bootstrap variances needed for studentized
## intervals

## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 1000 bootstrap replicates
##
## CALL :
## boot.ci(boot.out = bootMed2)
```

```
##
## Intervals :
## Level      Normal              Basic
## 95%   (10.49, 64.37 )   (17.00, 65.00 )
##
## Level     Percentile            BCa
## 95%   (14, 62 )   (14, 62 )
## Calculations and Intervals on Original Scale
```

**For ease of viewing, I'll just describe and reference the last sample (*bootMed2*) printed directly above. The differences between the Normal and Basic 95% CIs indicate that the bootstrapped sample is not perfectly normally distributed. Both of the CIs indicate there Obama's FT true population median could be anywhere from approximately 10-65. The standard error allows us to calculate this range, and the beauty of the bootstrap method is that we can calculate this standard deviation of the distribution (standard error) from just our single original sample. Without this resampling technique we wouldn't be able to test with any statistically meaningful backing if a median of another single sample was likely to be representative of the true population.**

10. (10 points) How are bootstrapping and cross-validation approaches to resampling different? How are they similar? Why does any of this matter from both social science and computational modeling perspectives?

**Different**: Bootstrapping selects *with* replacement, meaning some of the observations in the original data sample will not be represented in the bootstrapped sample. This ends up reducing the variance, but increasing bias. Cross-validation selects *without* replacement, meaning all the observations in the original data sample will show up in the smaller sets derived from the original. This ends up reducing the bias, but increasing the variance (the opposite of bootstrapping). Cross-validation is typically used for evaluating a model's performance or for comparing flexibility, while bootstrapping is more commonly used for determining the uncertainty of a model.

**Similar**: Both resampling techniques are used to help test and evaluate different statistical models. They both use repeated sampling to help ascertain more information about a fitted model that we would otherwise be unable to determine without obtaining an entirely new sample.

**Why does this matter**: The social sciences study the social world, and as such, obtaining accurate and reliable measures of behaviors or thoughts of interest are often incredibly expensive (time, resources, etc.). Obtaining a full population measure of anything is impossible. Therefore, we have to rely on samples of the population, which as I mentioned, are very costly to collect for financial reasons but also because some variables of interest are difficult to measure and some populations are hard to reach. These resampling techniques make the samples social scientists ARE able to collect much more statistically useful, as both the cross-validation and bootstrapping methods extend our abilities to inch closer to a more accurate understanding of what might actually might happening in our true population of interest.