Class Activity#9

Sunny Lee

2021-03-05

Permutation test and Bootstraping Use the music data to understand the permutation test.

```
music <- read.csv("C1 Music.csv")
attach(music)
```

1) Before they looked at the data, Anne and Anna decided to use a one-sided test to see whether fast music increased pulse rate more than slow music. Why is it important to determine the direction of the test before looking at the data?

```
fast_slow <- Fastdiff_Minus_Slowdiff
obs <- mean(fast_slow)
multiplier <- sample(c(1, -1), length(fast_slow), replace = TRUE)

mean(fast_slow * multiplier)
```

```
## [1] 0.8571429
```

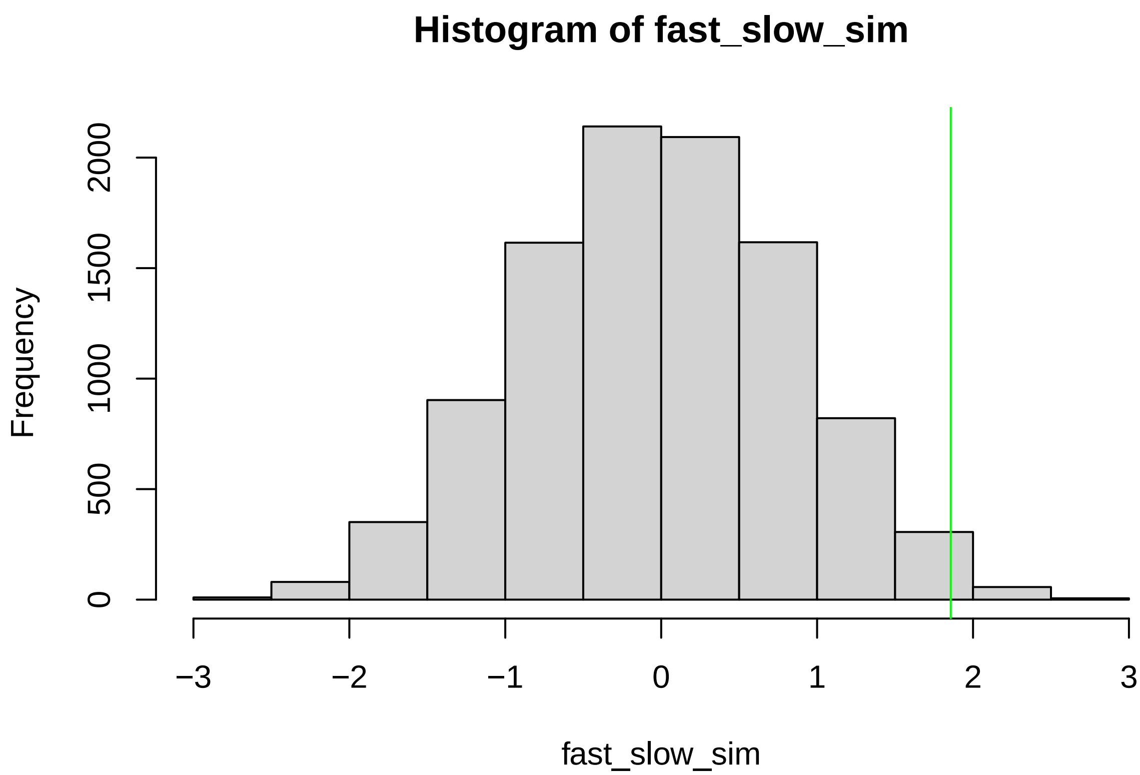
```
fast_slow
```

```
##  [1]  2  3 -8 -4 -6  2  1  2  1  2  9  3  6  8  3  0  2 -4  2 -2  1  2  3  2 13
## [26]  6  1  2
```

```
p <- 10000
fast_slow_sim <- rep(0, p)

for (i in 1:p){
  tempMult <- sample(c(1, -1), length(fast_slow), replace = TRUE)
  temp <- tempMult * fast_slow
  fast_slow_sim[i] <- mean(temp)
}

hist(fast_slow_sim)
abline(v = obs, col = "green")
```



```
p_value <- mean(fast_slow_sim > obs)
p_value
```

```
## [1] 0.0109
```

It is important to look into the data before we proceed because we need to know in our difference column whether we are subtracting the fast from teh slow or the slow from the fast.

2) Create a simulation to test the music data. Use the technology instructions provided to randomly multiply a 1 or a -1 by each observed difference. This randomly assigns an order (Fastdiff-Slowdiff or slowdiff-Fastdiff). Then, for each iteration, calculate the mean difference. The p-value is the proportion of times your simulation found a mean difference greater than or equal to 1.857

```
fast_slow <- Fastdiff_Minus_Slowdiff
obs <- mean(fast_slow)
multiplier <- sample(c(1, -1), length(fast_slow), replace = TRUE)

mean(fast_slow * multiplier)
```

```
## [1] 0.3571429
```

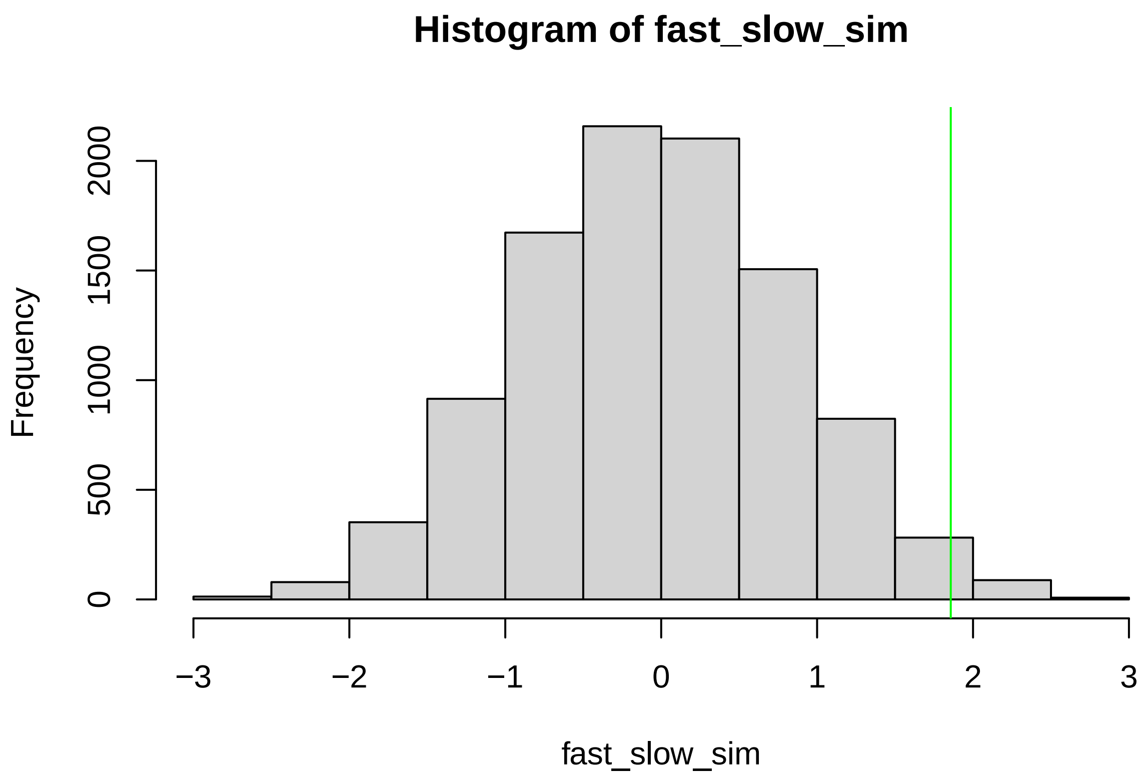
```
fast_slow
```

```
##  [1]  2  3 -8 -4 -6  2  1  2  1  2  9  3  6  8  3  0  2 -4  2 -2  1  2  3  2 13
## [26]  6  1  2
```

```
p <- 10000
fast_slow_sim <- rep(0, p)

for (i in 1:p){
  tempMult <- sample(c(1, -1), length(fast_slow), replace = TRUE)
  temp <- tempMult * fast_slow
  fast_slow_sim[i] <- mean(temp)
}

hist(fast_slow_sim)
abline(v = obs, col = "green")
```



```
p_value <- mean(fast_slow_sim > obs)
p_value
```

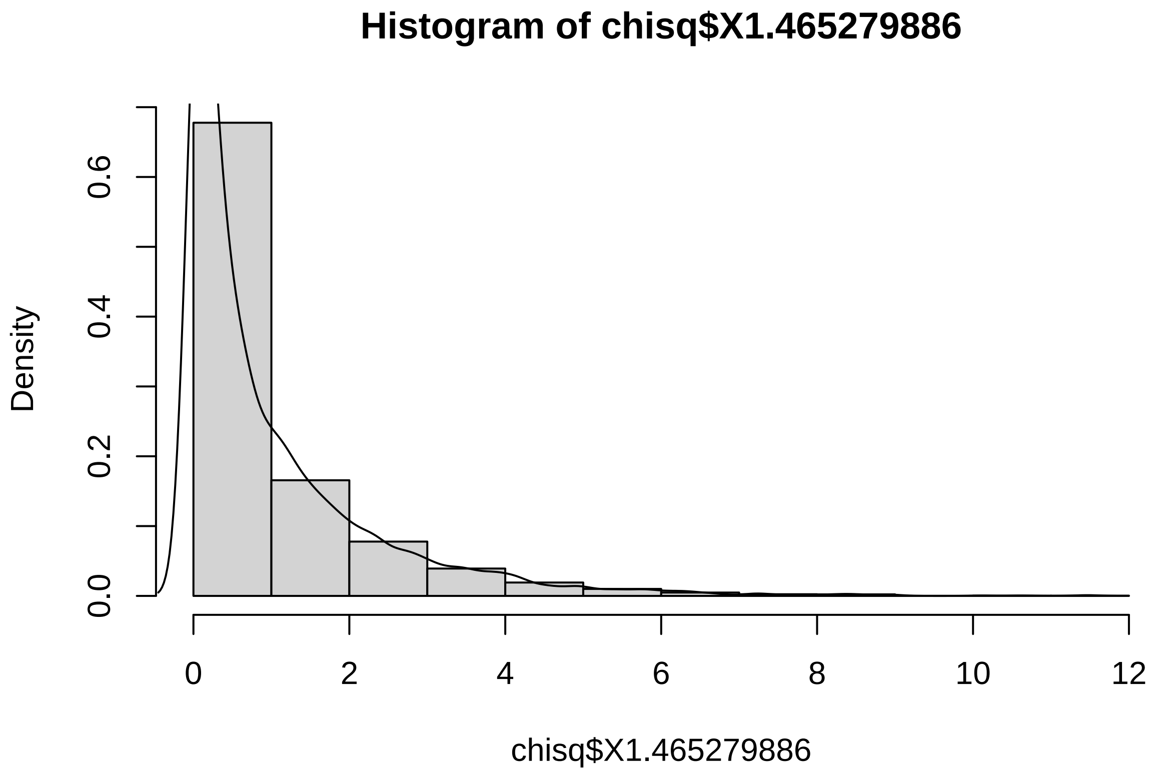
```
## [1] 0.0137
```

a) Create a histogram of the mean differences. Mark the area on the histogram that represents your p-value.
b) Use the p-value to state your conclusions in the context of the problem. Address random allocation and random sampling (or lack of either) when stating your conclusions.

Since we are working with paired data, the only randomness we can add to our data is changing which variable comes first when calculating the difference between the two variables. Thus, we randomly create an array of 1s and -1s, multiply this array to our difference array and then calculate the mean. By repeating this process multiple times, we can find a p value by calculating the amount of times our random mean is greater than our observed mean. Finally, from the p value we get above, we can reject the null hypothesis and conclude that the mean pulse rate after fast music is higher than the mean pulse rate of slow music.

3) The file Chisq contains data from a highly skewed population (with mean 0.9744 and standard deviation 1.3153)

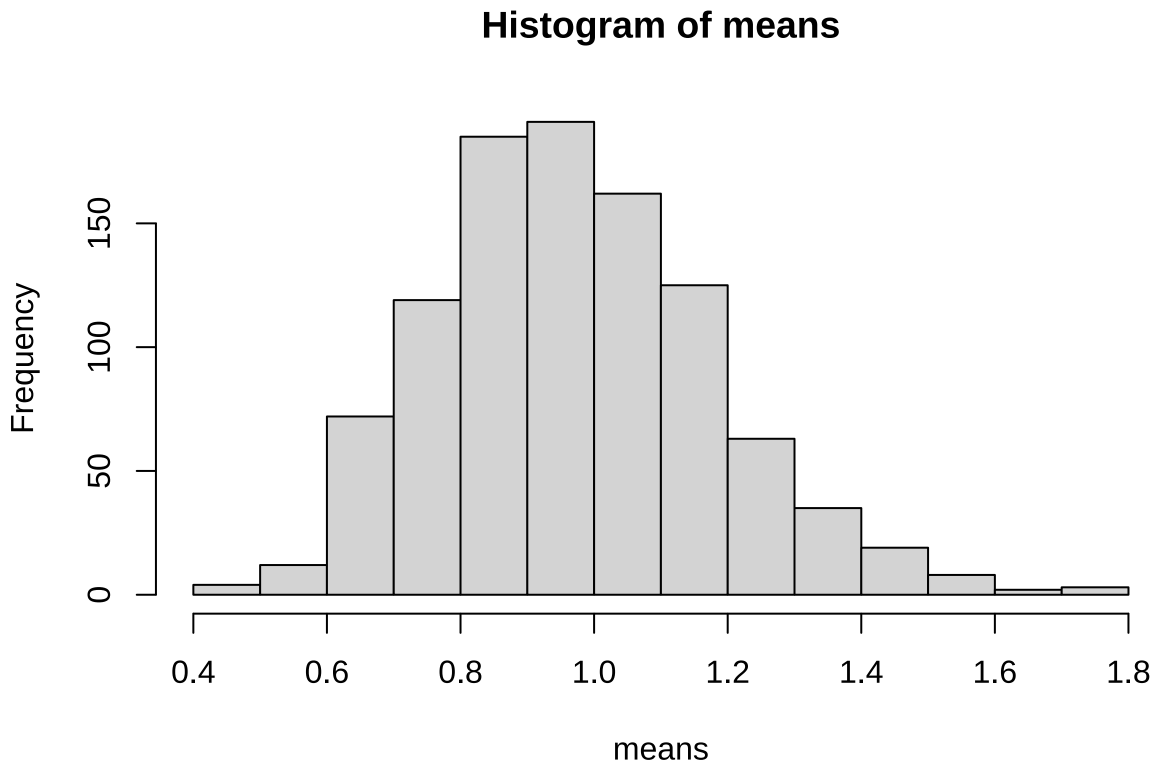
```
chisq <- read.csv("C1 Chisq.csv")
hist(chisq$X1.465279886, freq = FALSE)
points (density(chisq$X1.465279886), type = "l")
```



```
B <- 1000
n <- 40
means <- rep(0, B)

for (i in 1:B){
  means[i] <- mean(sample(chisq$X1.465279886, 40, replace = TRUE))
}

hist(means)
```



```
mean(means)
```

```
## [1] 0.9726667
```

```
1.3153 / sqrt(n)
```

```
## [1] 0.2079672
```

```
sd(means)
```

```
## [1] 0.206661
```

a) Take 1000 simple random samples of size 40 and calculate each mean ($\bar{X}$). Plot the histogram of the 1000 sample means. The distribution of the sample means is called the sampling distribution.

b) What does the central limit theorem tell us about the shape, center, and spread of the sampling distribution in this example?

The central limit theorem tells us since our sample size is greater than 30, we expect the shape of our sampling distribution to be normal. The center should also be quite close to the population mean and the spread should be about $\frac{\sigma}{\sqrt{n}}$.

c) Calculate the mean and standard deviation of the sampling distribution in Part A. Does the sampling distribution match what you would expect from the central limit theorem? Explain.

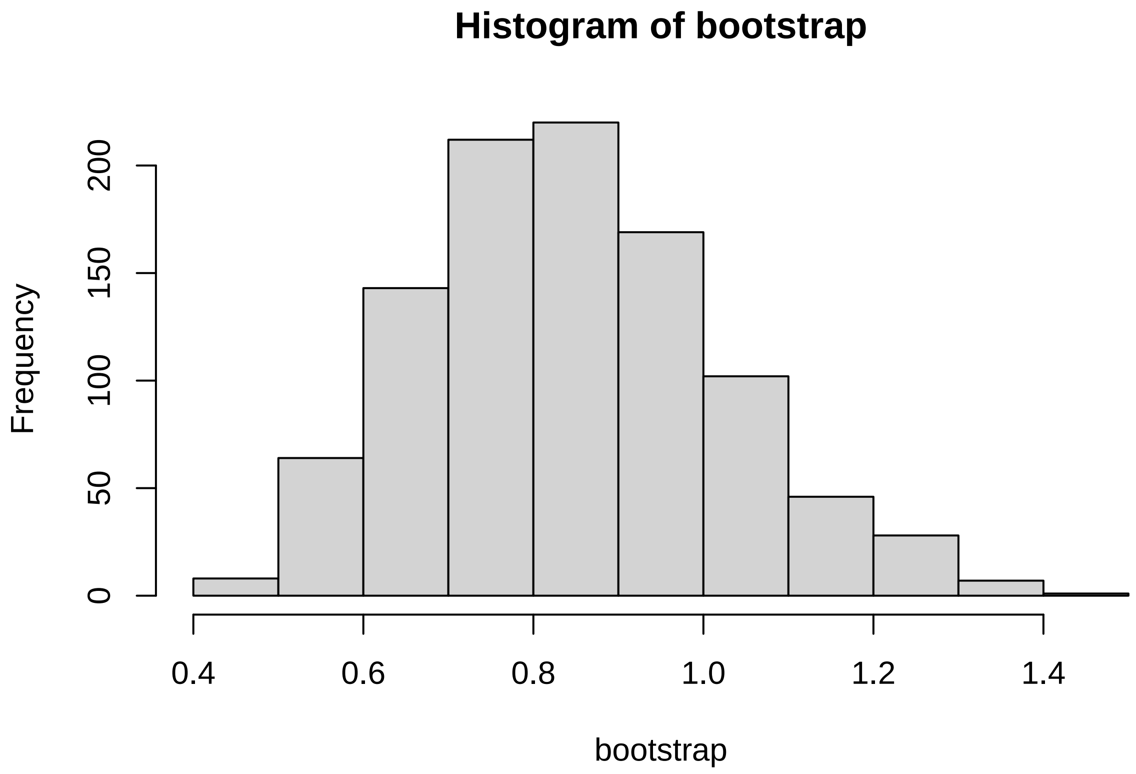
Calculating the mean and sd of the sampling distribution, we find those values are very similar to the population mean and the population sd over $sqrtn$. This makes sense, since the central limit theorem tells us that if we sample more than 30 times, our sampling distribution's mean and sd should approach the population values.

4) Take one simple random sample of size 40 from the Chisq data.

```
chisq_sm <- sample(chisq$X1.465279886, 40, replace = TRUE)
bootstrap <- rep(0, B)
B <- 1000

for (i in 1:B){
  bootstrap[i] <- mean(sample(chisq_sm, 40, replace = TRUE))
}

hist(bootstrap)
```



```
mean(bootstrap)
```

```
## [1] 0.8445166
```

```
sd(bootstrap)
```

```
## [1] 0.1751034
```

a) Take 1000 resampling (1000 samples of 40 observations with replacement from the one simple random sample).

b) Calculate the mean of each resample ($\bar{X}^*$) and plot the histogram of the 1000 resample means. This distribution of resample means is called the **bootstrap distribution**.

c) Compare the shape, center, and spread of the simulated histograms from Part B and Question 3 Part A. Are they similar?

Comparing the shape, we find that the shape is very similar to that of the histogram in Part A. As for the center and spread, we calculate the mean and sd of the new bootstrap distribution comparing those values to the values in Part A, we find they are quite similar.