HW5_106022103

106022103

2021/3/24

Set up

import libary

```
library(ggplot2)
```

Useful function

Before go on the question, just write some useful function to make the answer more pretty.

```
estimate_mean_CI <- function(samples, confidence_level = 0.95, population_sigma = -1){
  xbar <- mean(samples)
  s <- sd(samples)
  n <- length(samples)
  if (population_sigma != -1){
    # if we known the population sigma, use z-dist
    score <- qnorm(p = (1 - confidence_level)/2)
  }
  else
    # if we do not known the pupulation sigma, use t-dist
    score <- qt(p = (1 - confidence_level)/2, df = n-1)
  CI <- c(1, -1) * score * s / sqrt(n) + xbar
  return(CI)
}
```

```
plot_CI <- function(dist, confidence_level=0.95,title = ""){
  plot(density(dist), col="red",lwd=3,
       main=paste(title," (Confidence Level = ", confidence_level*100, "%)"))
  abline(v = mean(dist), lwd=1.5)
  abline(v=quantile(dist, probs = c((1-0.95)/2, (1+0.95)/2)),lty="dashed",col="blue")
}
```

Question 1)

> Some years ago, a Google engineer explained how they spot malicious apps (malware) that made it to their Android mobile apps store. Some malware apps deliberately turn off a security feature called Verify as soon as they are installed on an Android device. But there are also other, non-malicious, reasons why Verify might get turned off. So Google computes a "DOI score" for each app. The distribution of DOI scores is binomial, which Google approximates as a normal distribution (recall our reading on binomial distributions).

The equation for the Z-score to measure DOI is below,

- N = Number of devices that downloaded the app.

- x = Number of retained devices that downloaded the app.
- p = Probability of a device downloading any app will be retained.

a

Given the critical DOI score that Google uses to detect malicious apps (-3.7), what is the probability that a randomly chosen app from Google's app store will turn off the Verify security feature? (report a precise decimal fraction, not a percentage)

ANSWER:

```
Q_1a <- function(DOI)
  sprintf("The DOI score of %.1f has the probability of %.6f(= %.2e)
to turn off the Verify security feature.",
          DOI, pnorm(DOI), pnorm(DOI))
print(Q_1a(DOI=-3.7))
```

```
## [1] "The DOI score of -3.7 has the probability of 0.000108(= 1.08e-04)\nto turn off the Verify secur
```

b

Assuming there were ~2.2 million apps when the article was written, what number of apps on the Play Store did Google expect would maliciously turn off the Verify feature once installed?

ANSWER: According to the equation to calculate the malicious apps,

```
Q_1b <- function(DOI,N)
  sprintf("According to the DOI score=%.1f,
 there will be %d apps in all about %.1f million apps.",
          DOI, floor(N*pnorm(DOI)) , N/1e6)
print(Q_1b(DOI=-3.7, N=2.2e6))
```

```
## [1] "According to the DOI score=-3.7,\n there will be 237 apps in all about 2.2 million apps."
```

Question 2) This week's data is real but the scenario is imaginary

The large American phone company Verizon has a monopoly on phone services in many areas of the US. The New York Public Utilities Commission (PUC) regularly monitors repair times with customers in New York to verify the quality of Verizon's services. Verizon claims that they take 7.6 minutes to repair phone services for its customers on average. The file verizon.csv has a recent sample of repair times collected by PUC, who seeks to verify this claim at 99% confidence.

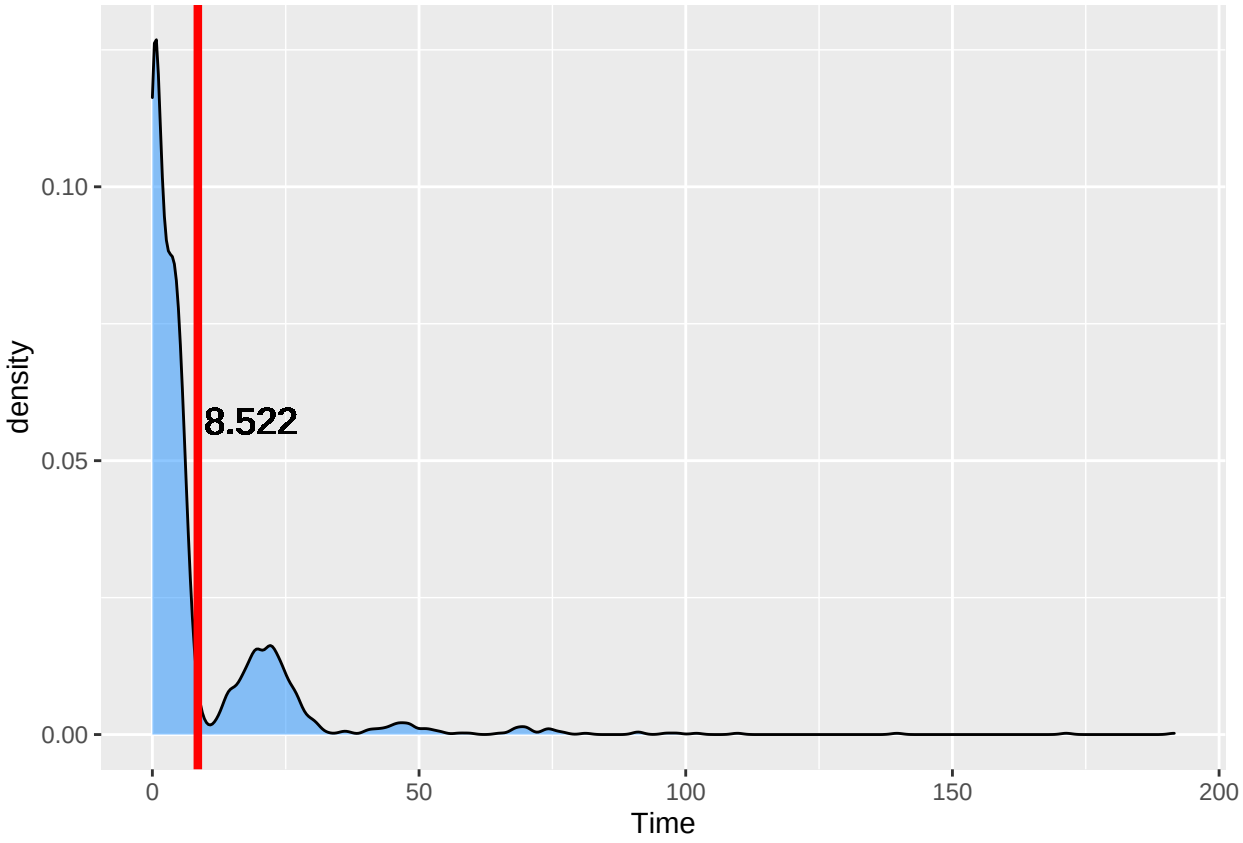
```
verizon <- read.csv("data/verizon.csv")
num_boots <- 2000
```

a. The Null distribution of t-values:

```
time_mean <- mean(verizon$Time)

ggplot(data = verizon, aes(x=Time)) +
  geom_density( fill="dodgerblue", alpha=0.5)+
  geom_vline(xintercept=time_mean, size=1.5, color="red") +
  geom_text(aes(label=round(time_mean,3), y=0.05,x=time_mean+10), vjust=-1,size=5)
```

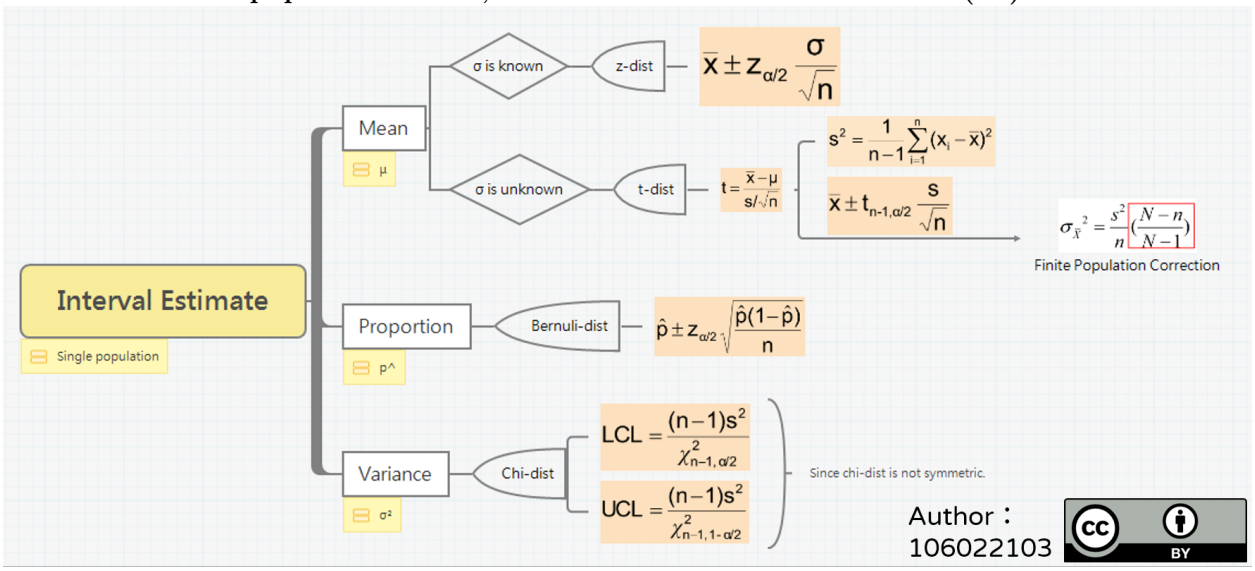
i. Visualize the distribution of Verizon's repair times, marking the mean with a vertical line



ii. Given what PUC wishes to test, how would you write the hypothesis? (not graded)
ANSWER: I would write the hypothesis like this,

- $H_0 : \mu = 7.6$
- $H_1 : \mu \neq 7.6$

iii. Estimate the population mean, and the 99% confidence interval (CI) of this estimate



ANSWER: Since the σ of population is unknown, we can choose t-dist to estimate the population mean.

```
CI99 <- estimate_mean_CI(verizon$Time,0.99)
sprintf("The CI of mean under 0.99 confidence level is (%.3f,%.3f)", CI99[1], CI99[2])
```

```
## [1] "The CI of mean under 0.99 confidence level is (7.594,9.450)"
```

```
t.test(verizon$Time, mu=7.6,conf.level = 0.99)
```

iv. Using the traditional statistical testing methods we saw in class, find the t-statistic and p-value of the test

```
##
##  One Sample t-test
##
## data:  verizon$Time
## t = 2.5608, df = 1686, p-value = 0.01053
## alternative hypothesis: true mean is not equal to 7.6
## 99 percent confidence interval:
##  7.593524 9.450495
## sample estimates:
## mean of x
##  8.522009
```

ANSWER: The t-statistic ≈ 2.5608 and $p - value \approx 0.01053$.

v. Briefly describe how these values relate to the Null distribution of t (not graded)

t-statistic **ANSWER:** The t-statistic is the ratio of the deviation of the parameter estimate from its hypothetical value to its standard error.

p-value **ANSWER:** The p-value is the probability of observing a sample with at least the same extremes as the actual observed sample, assuming that the null hypothesis is true in the hypothesis test.

vi. What is your conclusion about the advertising claim from this t-statistic, and why?
ANSWER: Since the p-value is 0.0153, so we would

- **reject** the Null hypothesis under the **95%** confidence interval.
- **not reject** the Null hypothesis under the **99%** confidence interval.

b. Let's use bootstrapping on the sample data to examine this problem:

```
boot_mean <- function(sample0, mean_hyp) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  return( mean(resample) )
}
means <- replicate(num_boots, boot_mean(sample0=verizon$Time, mean_hyp=mean(verizon$Time)))
means_CI <- quantile(means, probs = c(0.005, 0.995))
sprintf("The CI of means under 0.99 confidence level is (%.3f,%.3f)",
        means_CI[1], means_CI[2])
```

i. Bootstrapped Percentile: Estimate the bootstrapped 99% CI of the mean

```
## [1] "The CI of means under 0.99 confidence level is (7.659,9.498)"
```

ii. Bootstrapped Difference of Means:

What is the 99% CI of the bootstrapped difference between the population mean and the hypothesized mean?

```
boot_mean_diff <- function(sample0, mean_hyp) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  return( mean(resample) - mean_hyp )
}
mean_diffs <- replicate(num_boots, boot_mean_diff(sample0=verizon$Time,
                                                  mean_hyp=mean(verizon$Time)))
mean_diffs_CI <- quantile(mean_diffs, probs = c(0.005, 0.995))
sprintf("The CI of difference of means under 0.99 confidence level is (%.3f,%.3f)",
        mean_diffs_CI[1], mean_diffs_CI[2])
```

```
## [1] "The CI of difference of means under 0.99 confidence level is (-0.843,0.959)"
```

iii. Bootstrapped t-Interval:

What is 99% CI of the bootstrapped t-statistic?

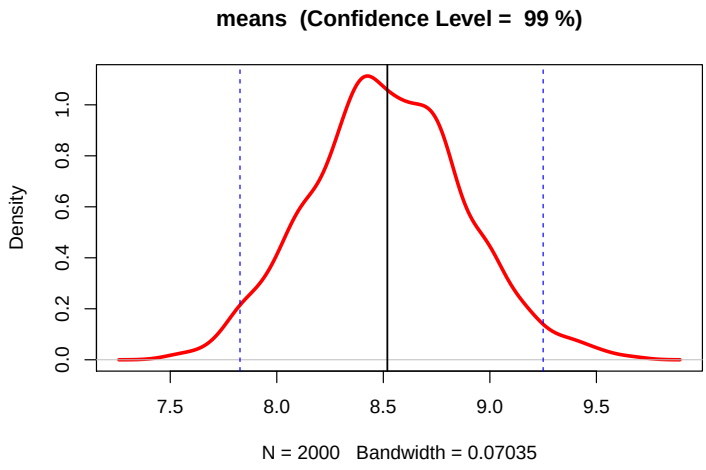
```
boot_t_stat <- function(sample0, mean_hyp) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  diff <- mean(resample) - mean_hyp
  se <- sd(resample)/sqrt(length(resample))
  return( diff / se )
}
t_stats <- replicate(num_boots, boot_t_stat(sample0=verizon$Time,
                                            mean_hyp=mean(verizon$Time)))
t_stats_CI <- quantile(t_stats, probs = c(0.005, 0.995))
sprintf("The CI of t-statistic under 0.99 confidence level is (%.3f,%.3f)",
        t_stats_CI[1], t_stats_CI[2])
```

```
## [1] "The CI of t-statistic under 0.99 confidence level is (-2.833,2.212)"
```

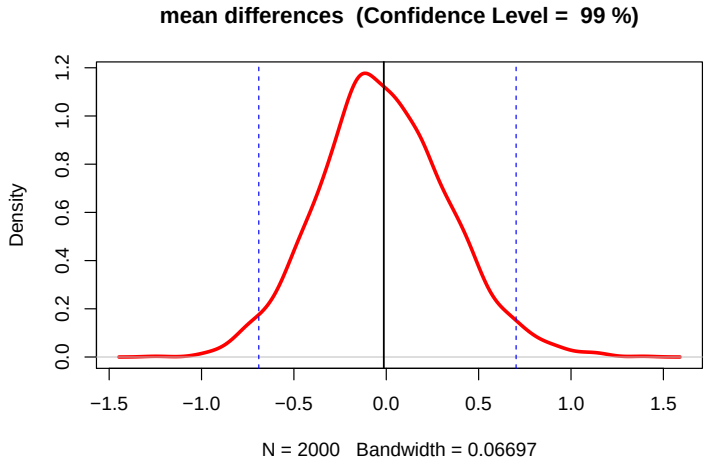
iv. Plot separate distributions of all three bootstraps above

(for ii and iii make sure to include zero on the x-axis)

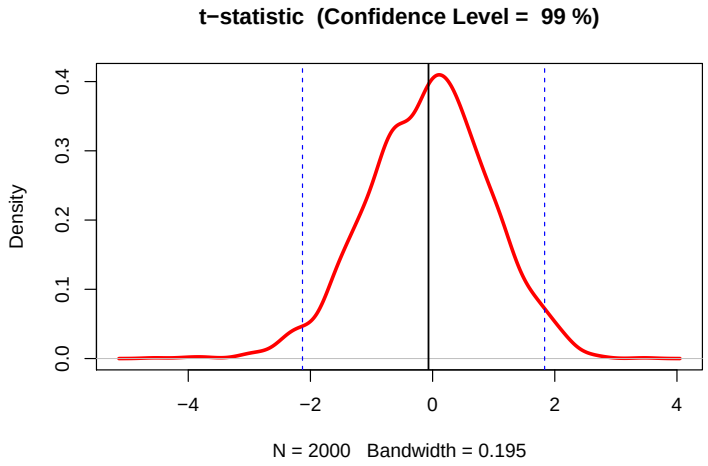
```
plot_CI(means,0.99,"means")
```




```
plot_CI(mean_diffs,0.99,"mean differences")
```




```
plot_CI(t_stats,0.99,"t-statistic")
```



c. Do the four methods (traditional test, bootstrapped percentile, bootstrapped difference of means, bootstrapped t-Interval) agree with each other on the test?

ANSWER:

- traditional test:
  - $CI_{99\%} = (7.594, 9.450)$
  - **not reject** the Null hypothesis under the **99%** confidence interval.
- bootstrapped percentile:
  - $CI_{99\%} = (7.596, 9.464)$
  - **not reject** the Null hypothesis under the **99%** confidence interval.
- bootstrapped difference of means:
  - $CI_{99\%} = (-0.847, 0.957)$
  - **not reject** the Null hypothesis under the **99%** confidence interval.
- bootstrapped t-Interval:
  - $CI_{99\%} = (-3.155, 2.318)$
  - **not reject** the Null hypothesis under the **99%** confidence interval.

All four methods agree with each other.