HW4

110078509

20220309

```
rm(list=ls())
#remove the random variable to fresh the working environment
```

Q1. Given the critical DOI score that Google uses to detect malicious apps

(a)

```
pnorm(-3.7)
```

```
## [1] 0.0001077997
```

(b)

```
2.2*1000000*pnorm(-3.7)
```

```
## [1] 237.1594
```

Question 2

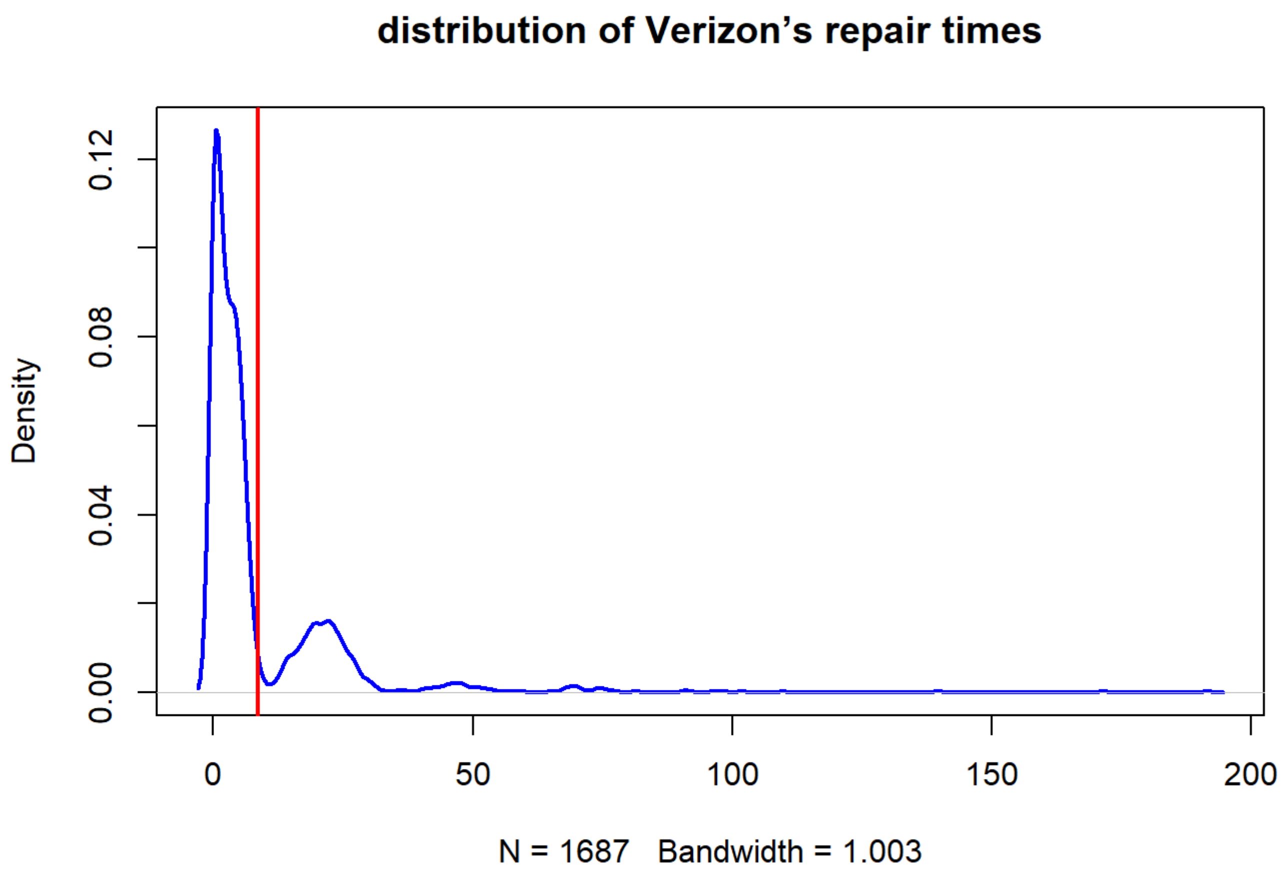
a. The Null distribution of t-values:

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

```
raw <- read.csv("verizon.csv", header = T)$Time

plot(density(raw), lwd=2, col="blue",
     main="distribution of Verizon’s repair times")

abline(v=mean(raw), lwd=2, col="red")
```



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

PUC wants to test whether Verizontake average 7.6 minutes to repair phone services for its clients. And they intend to verify this claim at 99% confidence. H0: mean = 7.6 minutes H1: mean != 7.6 minutes

---

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

population mean

```
x<-mean(raw) #sample mean
SE<-sd(raw)/sqrt(length(raw)) #standard_error
CI.99 <- c(x-2.58*SE , x+2.58*SE)

# Estimate the population mean
sprintf("Estimate the population mean: %f", x)
```

```
## [1] "Estimate the population mean: 8.522009"
```

```
#Because sample mean is the unbiased estimator of population's, x is the estimation of population mean,

#99% confidence interval
CI.99
```

```
## [1] 7.593073 9.450946
```

Ans:

As the markdown above, the 99% confidence interval is [7.593073 , 9.450946]. And the Estimate the population mean is 8.522009

iv. Using the traditional statistical testing methods we saw in class,

find the t-statistic and p-value of the test

```
# Let 7.6 = μ0
manager_hyp <- 7.6

SE <- sd(raw)/sqrt(length(raw)) #standard_error
t  <- (mean(time)-manager_hyp)/SE
```

```
## Warning in mean.default(time): 引數不是數值也不是邏輯值：回覆 NA
```

```
t # t statistic
```

```
## [1] NA
```

```
#caculation p-value of the test t
df=length(raw)-1 #degree of freedom-1 to get unbiaed estimator
p_value<- 1-pt(t,df)
p_value
```

```
## [1] NA
```

Ans:

t-value = 2.560762 p_value = 0.005265342

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

Ans:

T-distributions assume that us draw repeated random samples from a population where the null hypothesis is true. For t-value, as the sample data become progressively dissimilar from the null hypothesis, the absolute value of the t-value increases. And the p-value is the probability under H0 of observing a test statistic at least as extreme as what was observed.

vi. What is your conclusion about the advertising claim from this t-statistic, and why?

Ans:

Conclusion: Fail to reject H0. Because, t-value = 2.560762 & p_value = 0.005265342 >0.005, And the hypothesized condition is rejected if the p-value < 0.005 as 99% confident level.

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

Basic setting for question b first

```
# sample_size <- length(raw)# 1687
# sample_mean <- mean(raw) # 8.522009
# sample_sd <- sd(raw) # 14.78848
hypmanager_hyp <- 7.6
num_boots <- 2000
```

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

```
set.seed(53151)
Simple_boost <- function(func, sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  func(resample)
  }

sample_means <- replicate(num_boots, Simple_boost(mean, raw))

# plot(density(sample_means), lwd=2, main="The Bootstrapped 99% CI of the mean")

BootCI99 <- quantile(sample_means, probs = c(0.005, 0.995))
BootCI99 # 99% CI interval
```

```
##     0.5%    99.5% 
## 7.662153 9.568937
```

ii. Bootstrapped Difference of Means:

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

```
#Bootstrapping the 95% CI of the Difference of Means
# I paste the code from the pdf
boot_mean_diffs <- function(sample0, mean_hyp) {
resample <- sample(sample0, length(sample0), replace=TRUE)
return( mean(resample) - mean_hyp )
}

set.seed(64264)
num_boots <- 2000
mean_diffs <- replicate(
num_boots,
boot_mean_diffs(raw, manager_hyp)
)
diff_ci_99 <- quantile(mean_diffs, probs=c(0.005, 0.995))
diff_ci_99
```

```
##       0.5%      99.5% 
## 0.02777012 1.83839069
```

Ans:

The 99% CI of the bootstrapped mean difference is [0.02777012 , 1.83839069] under my seed setting as 64264.

iii. Bootstrapped t-Interval:

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

```
# i pasted the code fro the slides
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 )
}

set.seed(2346786)
num_boots <- 2000
t_boots <- replicate(num_boots, boot_t_stat(raw, manager_hyp))
# plot(density(t_boots), xlim=c(0,12), col="blue", lwd=2)

t_ci_99 <- quantile(t_boots, probs=c(0.005, 0.995))
t_ci_99
```

```
##      0.5%     99.5%
## 0.2434266 4.6637516
```

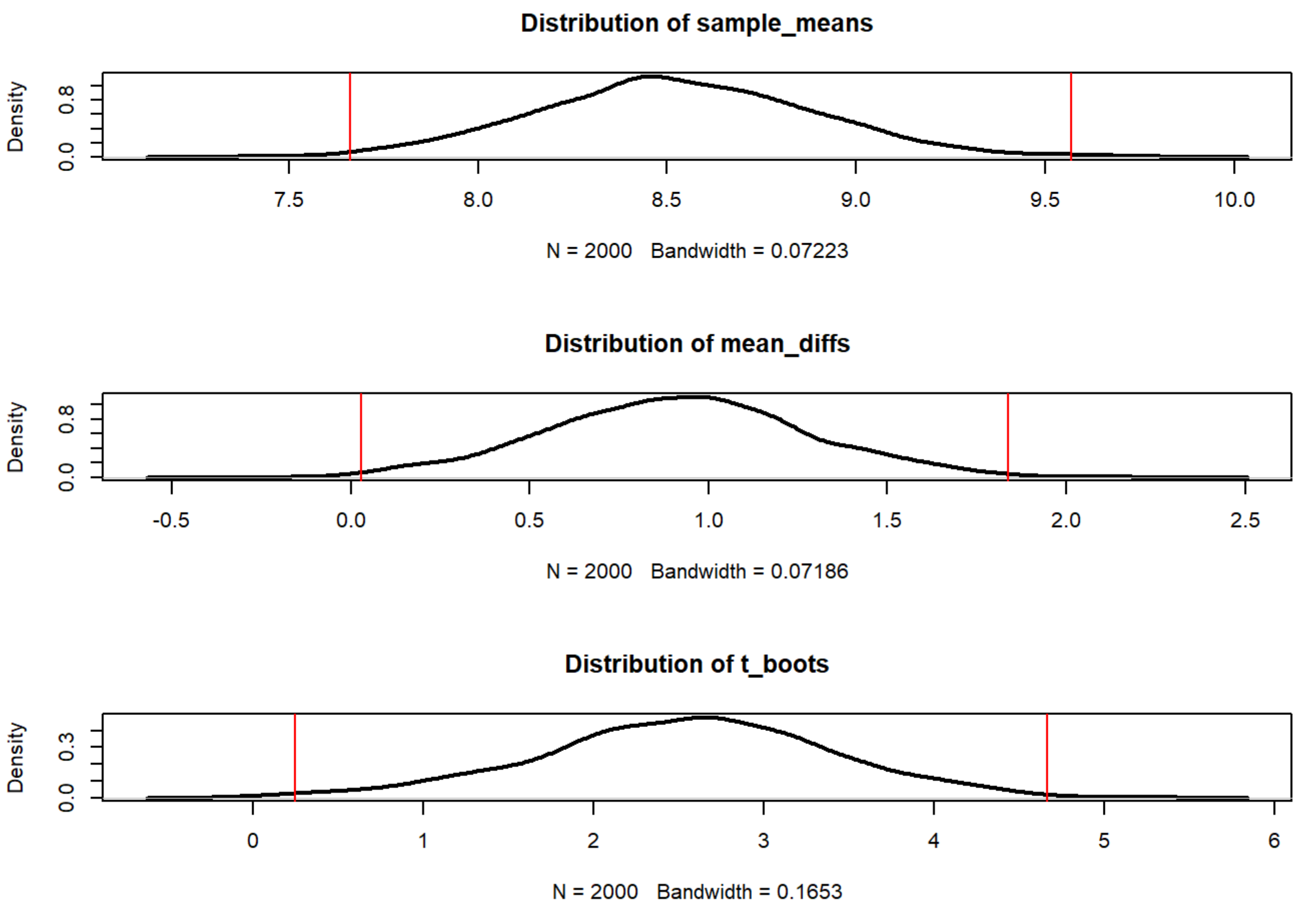
Ans:

the 99% CI of the bootstrapped t-statistic is [0.2434266 , 4.6637516] under my seed setting as 2346786.

iv. Plot separate distributions of all three bootstraps above

```
par(mfrow = c(3,1))

pop_the_sh_out <- function(boost_data, ci, title) {
plot(density(boost_data),lwd=2,main = title)
abline(v=ci, lty=1, lwd=1, col="red")
}
pop_the_sh_out(sample_means,BootCI99, "Distribution of sample_means" )
pop_the_sh_out(mean_diffs,diff_ci_99, "Distribution of mean_diffs" )
pop_the_sh_out(t_boots,t_ci_99, "Distribution of t_boots" )
```



Explain:

As the plot shown above, the red vertical lines are the 99% CI of each plot.And this plot can make us earlier to compare the distributions of all three bootstraps above.

---

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

Ans.

Kindly notice that all of my statements are answered under my seed setting.

For this question, I will do the calculation separately, then, summarize afterward.

For traditional test,

Because, t-value = 2.560762 & p_value = 0.005265342 > 0.005, And the hypothesized condition is rejected if the p-value < 0.005 as 99% confident level(Two-tails). Therefore, we can not reject H0.

```
print("For traditional test")
```

```
## [1] "For traditional test"
```

```
sprintf("t: %f",t)
```

```
## [1] "t: NA"
```

```
sprintf("p_value: %f",p_value)
```

```
## [1] "p_value: NA"
```

For bootstrapped percentile,

the hypothesis mean (7.6) is not includes in 99% C.I range[7.662153 ,9.568937]. Therefore, we reject H0.

```
print("For bootstrapped percentile")
```

```
## [1] "For bootstrapped percentile"
```

```
print(BootCI99)
```

```
##     0.5%    99.5% 
## 7.662153 9.568937
```

For bootstrapped difference of means

As the plot shown in IV, the 0 is not includes in the 99% range [0.02777012 ,1.83839069] Therefore, we reject H0.

```
print("bootstrapped difference of means")
```

```
## [1] "bootstrapped difference of means"
```

```
print(diff_ci_99)
```

```
##       0.5%      99.5% 
## 0.02777012 1.83839069
```

For the T interval

As the plot shown in IV, the 0 is not includes in the 99% range [0.02777012 ,1.83839069] Therefore, we reject H0.

```
print("t interval")
```

```
## [1] "t interval"
```

```
print(t_ci_99)
```

```
##      0.5%     99.5% 
## 0.2434266 4.6637516
```

However, because of the random seed, the outcome will be different sometime, as the lower bounds are very close to 7.6 and 0 in this case, so it will sometime accept the H0 because of the random seed.

Summary

According to the calculation and the detailed explanation mentioned above, we could not reject H0 base on traditional test. However, we could reject H0 based on the rest of method. Hence, they are not agree with each other. Lastly,I wished to stress again that the ‘seed’ would affect the result. Therefore, my answer only valid under the current setting of seed.