HW4

108048110

3/9/2022

Used packages

```
#install.packages(c("ggplot2", "gapminder", "gridExtra", "ggpubr"))
library(ggplot2)
library(gapminder)
library(gridExtra)
library(ggpubr)
```

```
seed <- runif(1)*10^9
paste(seed)
```

```
## [1] "381303698.988631"
```

Question 1) DOI score

The DOI score indicates an app has a statistically significant lower retention rate if the Z-score is much less than -3.7. `Retention Rate == DOI score == Z score`

a. Given the critical DOI score (-3.7), what is the probability that a randomly chosen app will turn off the Verify security feature? (decimal)

```
pnorm(-3.7)
```

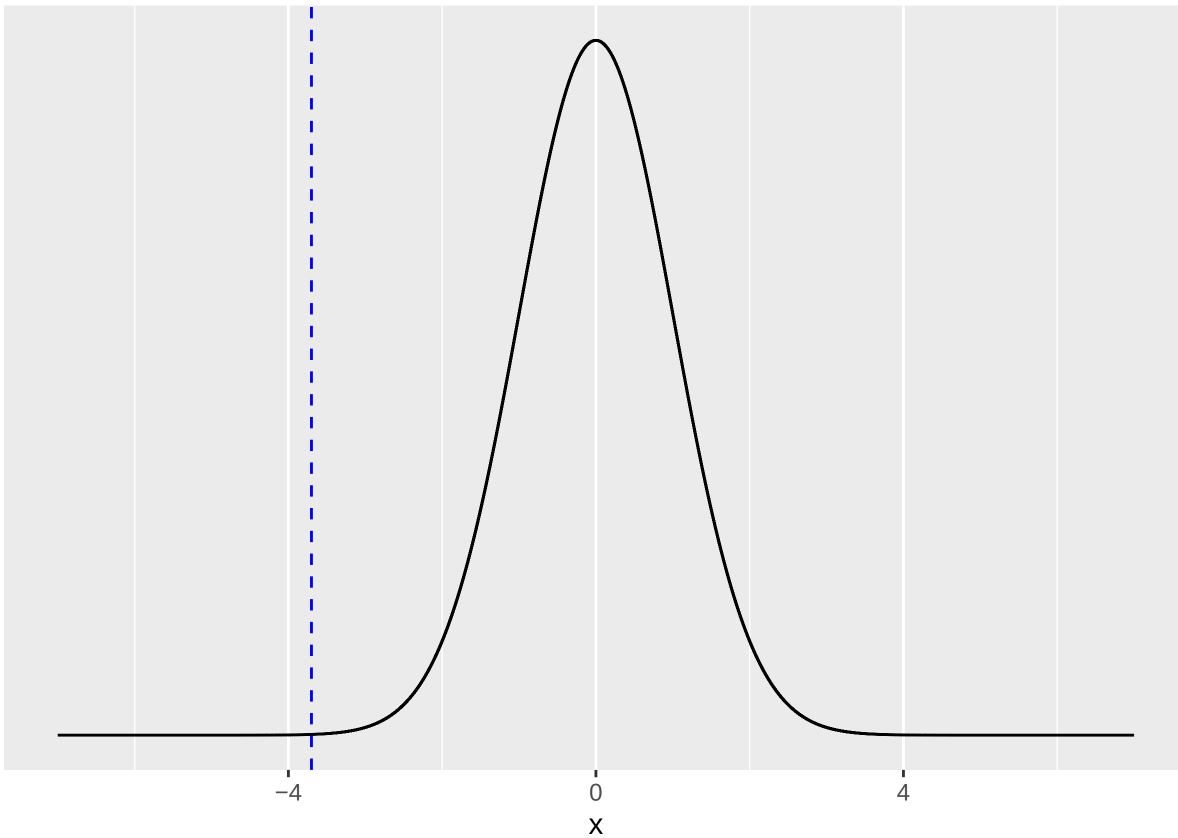
```
## [1] 0.0001077997
```

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

```
round(pnorm(-3.7)*220000)
```

```
## [1] 24
```

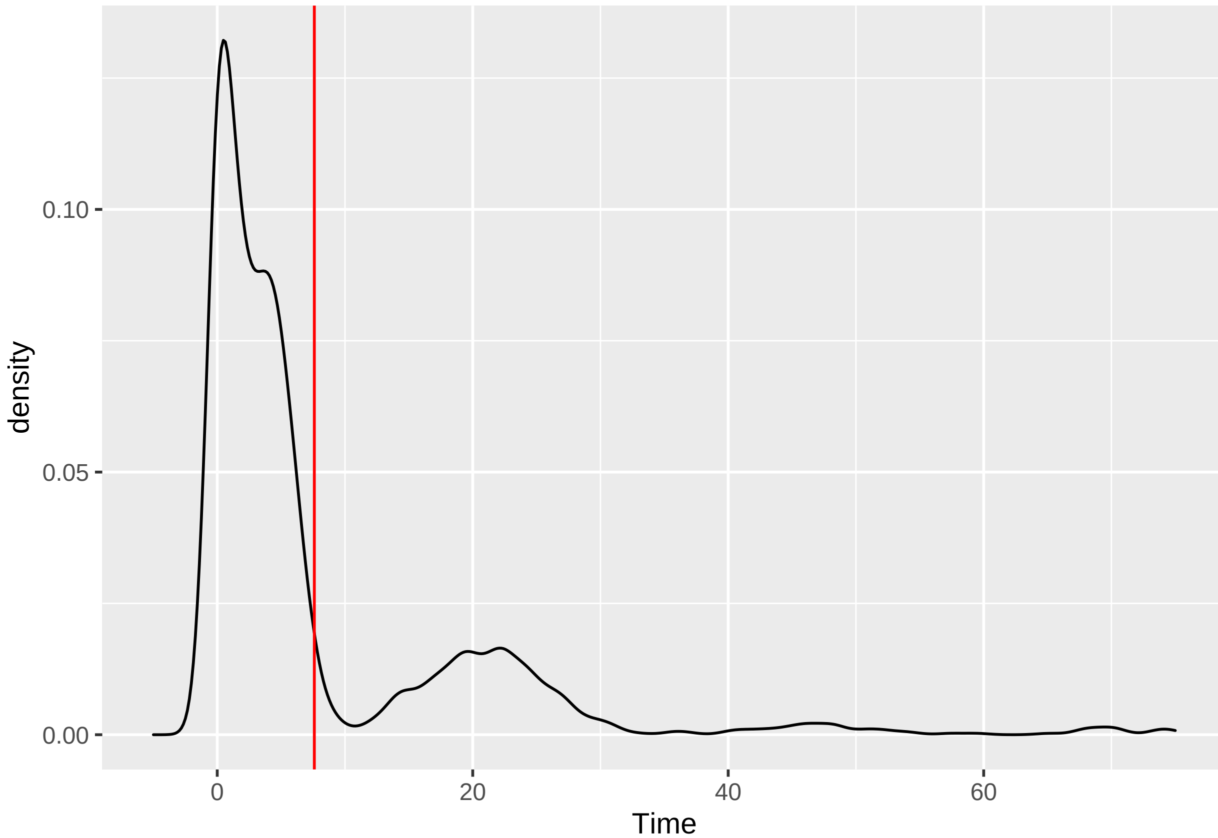
```
ggplot(data.frame(x=c(-7, 7)), aes(x))+
  stat_function(fun=dnorm, n=220000, args=list(mean=0, sd=1))+
  ylab("")+
  scale_y_continuous(breaks = NULL)+
  geom_vline(xintercept = -3.7, color="blue", linetype="dashed")
```



Question 2) Verizon

Verizon claims that they take 7.6 minutes to repair phone services for its customers on average.PUC needs to verify the quality of Verizon's services.Recent sample of repair times collected by PUC, who seeks to verify this claim at 99% confidence are stored in a variable, `repair_times`.

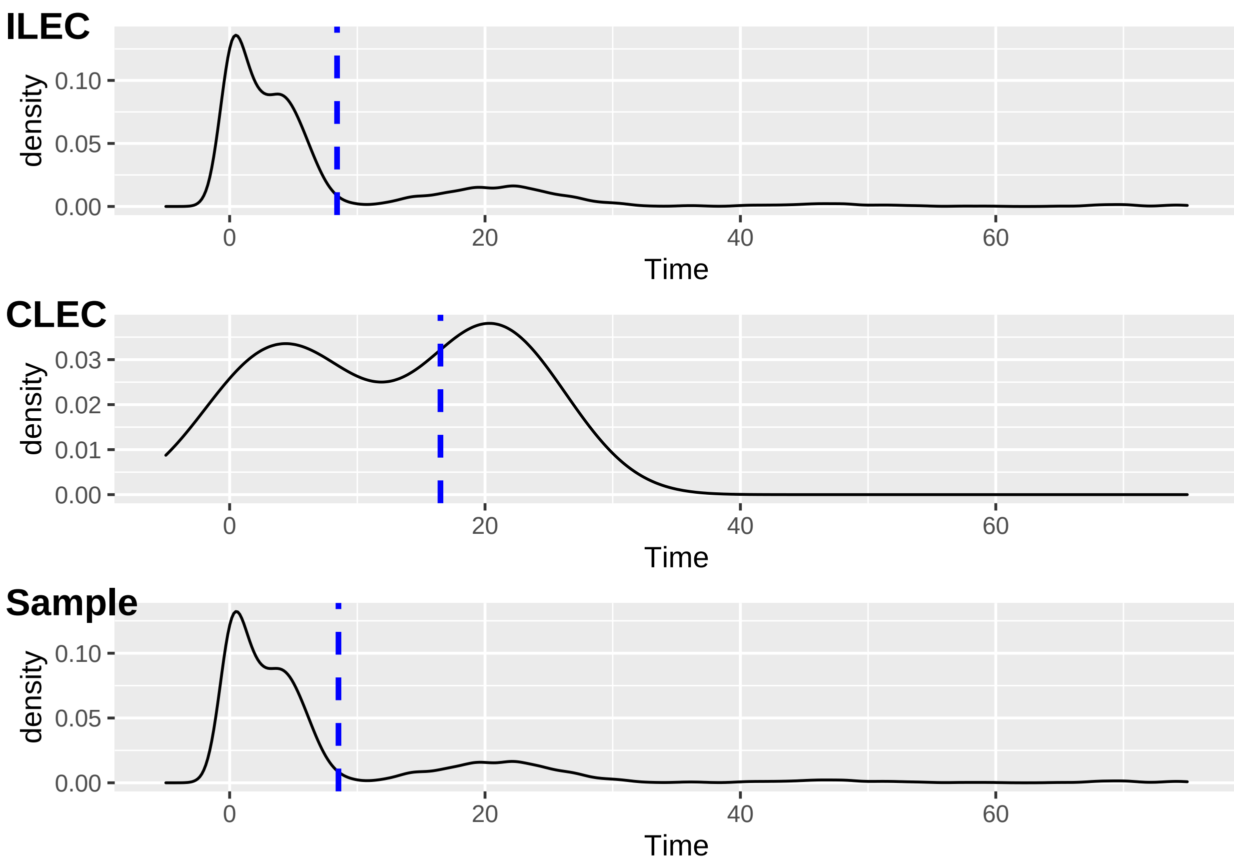
```
claims <- 7.6
repair_times <- read.csv("verizon.csv")
TIME <- repair_times$Time
Group <- repair_times$Group
ILEC <- repair_times[repair_times$Group=="ILEC",]
CLEC <- repair_times[repair_times$Group=="CLEC",]

ggplot(repair_times, aes(x=Time))+
  geom_density()+
  xlim(-5,75)+
  geom_vline(xintercept = 7.6, color="red", linewidth=3)
```



a. The Null distribution of t-values:

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

```
first_plot <- function(data){
  p <- ggplot(data, aes(x=Time))+
    geom_density()+
    xlim(-5, 75)+
    geom_vline(aes(xintercept = mean(Time)), color="blue", linetype="dashed", size=1)

  return(p)
}

p1 <- first_plot(repair_times)
p2 <- first_plot(ILEC)
p3 <- first_plot(CLEC)

ggarrange(p2, p3, p1,
          labels=c("ILEC", "CLEC", "Sample"),
          ncol=1,
          nrow=3,
          vjust=1,
          hjust=0)
```



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

Ans. The null hypothesis statement in this question is Verizon's claims an average repair time of 7.6 minutes; the alternative hypothesis happens when the null hypothesis statement is wrong; consequently, it can be written in mathmetic terms as: H0 : mu = 7.6 H1 : mu != 7.6

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

```
sample_sd <- function(data){
  v <- sum((data-mean(data))^2)/(length(data)-1)
  sqrt(v)
}

property <- function(data, type="p", CI=0.95){
  Mean <- mean(data)
  Median <- median(data)
  if(type=="p"){Std <- sd(data)}
  else{Std <- sample_sd(data)}
  Stderr <- Std/sqrt(length(data))
  if(CI==0.9){CI <- Mean+c(-1, 1)*1.645*Stderr}
  else if (CI==0.95) { CI <- Mean+c(-1, 1)*1.96*Stderr}
  else if (CI==0.99) {CI <- Mean+c(-1, 1)*2.57*Stderr}

  cat("Mean: ", Mean,
      "\nMedian: ", Median,
      "\nStd: ", Std,
      "\nStderr: ", Stderr,
```

```r
      "\nCI: ", CI)
}

property(TIME,type="s", CI=0.99)
```

```
## Mean:  8.522009
## Median:  3.63
## Std:  14.78848
## Stderr:  0.3600527
## CI:  7.596674 9.447345
```

*(iv) Use the **`traditional statistical testing methods`** to find the t-statistic and p-value of the test.*

```r
# If H0 is true...
sample_size <- length(TIME)
sample_mean <- mean(TIME)
sample_sd <- sample_sd(TIME)
sderr <- sample_sd/sqrt(sample_size)
t <- (sample_mean-claims)/sderr
t
```

```
## [1] 2.560762
```

```r
p <- (1-pt(t, sample_size-1))
p
```

```
## [1] 0.005265342
```

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

Ans. The P-value approach involves determining "likely" or "unlikely" by determining the probability of observing a more extreme test statistic in the direction of the alternative hypothesis (H1) than the one observed.At first, I assumed a null hypothesis (H0) is true, and used the p-value approach to determine whether or not to reject the assumption. Using the known distribution of the test statistics, I then calculated the p-value.'

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

Ans. A small p-value indicates strong evidence against the null hypothesis, so the PUC should probably reject Verizon's claims.

b. Bootstraped on the sample data to examine this problem: `for ii and iii make sure to include zero on the x-axis`

```r
set.seed(round(seed))
# return mean of sample
boot_mean <- function(data, func){
  resample <- sample(data, length(data), replace=TRUE)
  func(resample)
}

# return mean diff of sample and claim
boot_mean_diff <- function(data, claim){
```

```r
  resample <- sample(data, length(data), replace = TRUE)
  return(mean(resample)-claim)
}

# return t stat of sample
boot_t_stat <- function(data, claim){
  resample <- sample(data, length(data), replace=TRUE)
  t <- (mean(resample)-claim)/(sd(resample)/sqrt(length(resample)))
  return(t)
}
```

Function that create CI graph

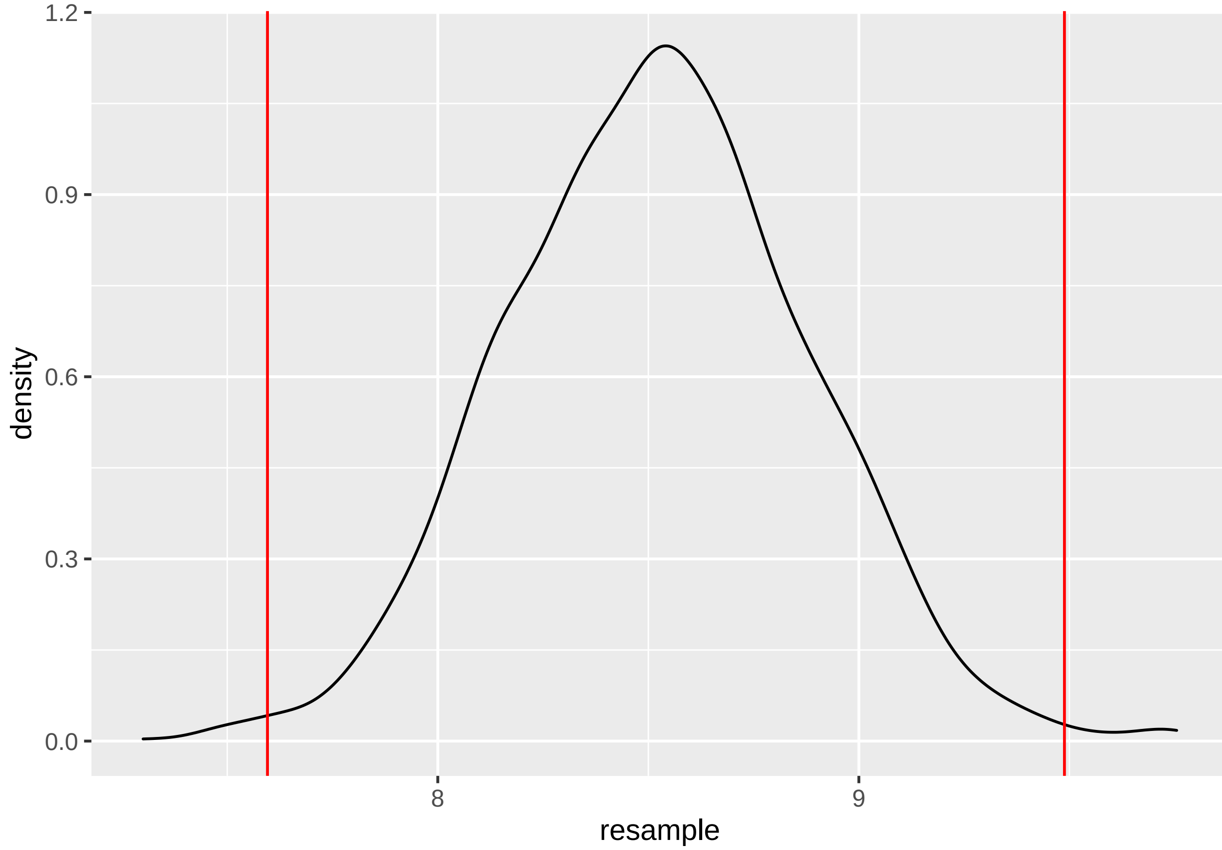
```r
draw_ci <- function(data, title="99% CI"){
  c_mean <- mean(data)
  c_sd <- sd(data)
  n <- length(data)

  plot(x=c(c_mean-0.125*c_sd,
           c_mean+0.125*c_sd),
       y=c(0,110), type = "n", main=title)

  abline(v = c_mean, lty = 2, col = "red")


  for(i in 1:100){
    x = rnorm(2000 , c_mean , c_sd)
    width = qt(0.995 , n-1)*sd(x)/sqrt(n)

    left = mean(x)-width
    right = mean(x)+width

    if (c_mean >= left && c_mean <= right){
      lines(c(left,right), c(i,i),lty = 1)
      }else{
        lines(c(left,right), c(i,i), lwd = 2, col="green")
      }
  }
}
```

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

```r
set.seed(round(seed))
resample <- replicate(2000, boot_mean(TIME, mean))
quantile(resample, probs=c(0.005, 0.995))
```

```
##     0.5%    99.5%
## 7.595476 9.488254
```

```r
ggplot()+
  geom_density(aes(resample))+
  geom_vline(xintercept=quantile(resample, c(0.005, 0.995)), color="red")
```



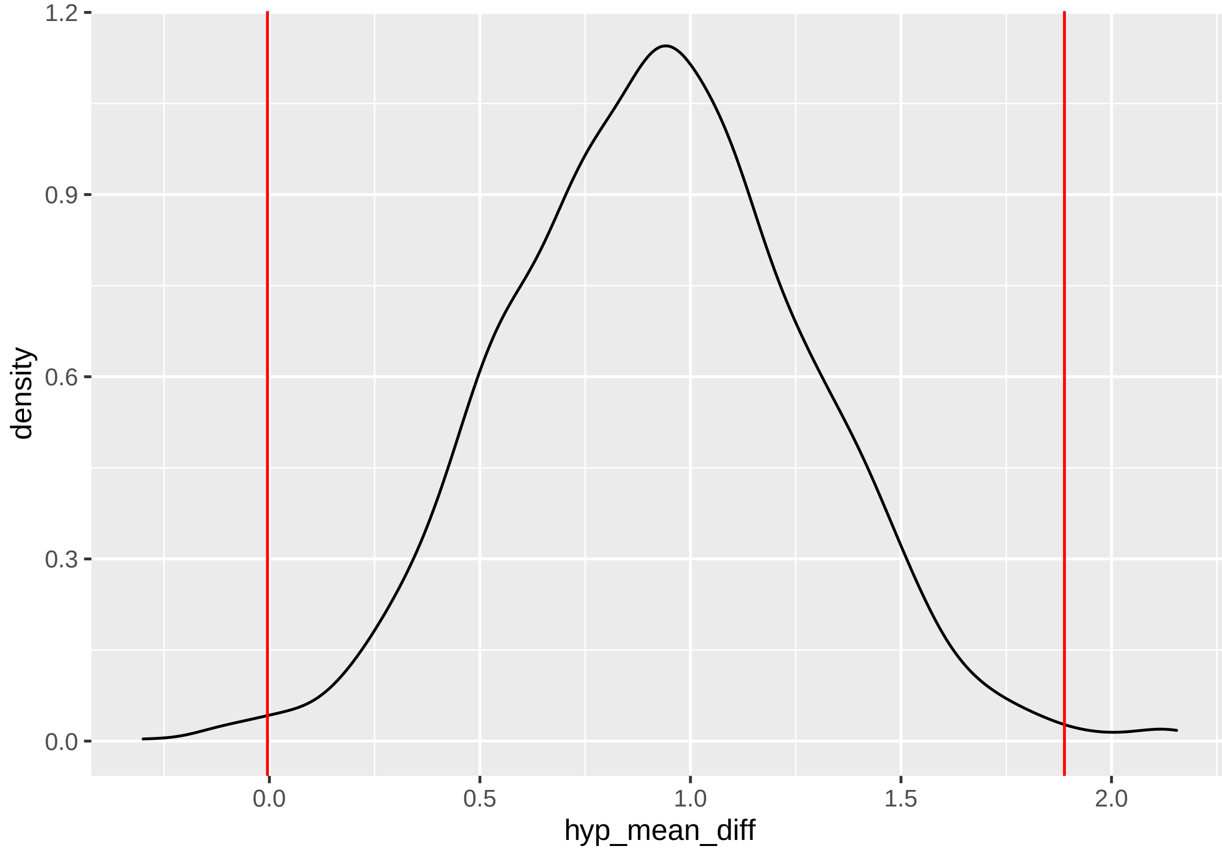
Note. As we can observe from the graph above, two red line indicates the 99% confidence interval of the sample bootstrapped data, and the claim (7.6) is not included, therefore, we reject the null hypothesis.

```
(ii)What is the 99% CI of the bootstrapped difference?
```

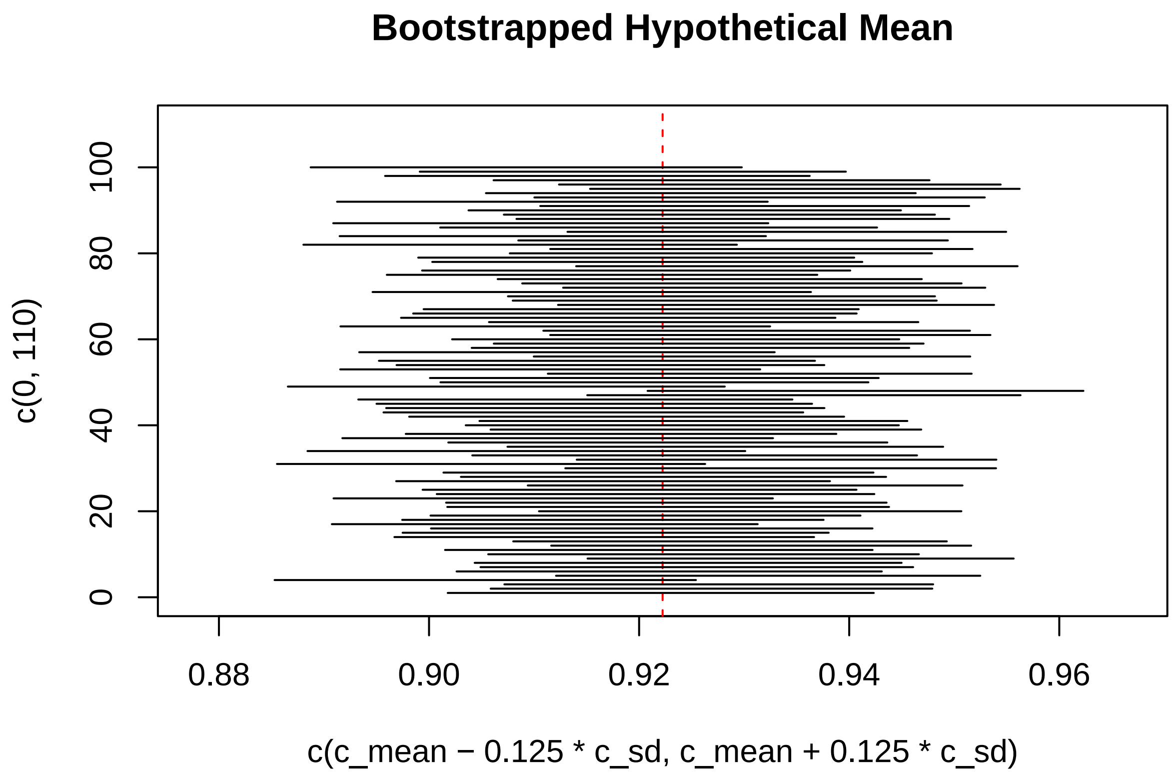
```
set.seed(round(seed))
hyp_mean_diff <- replicate(2000, boot_mean_diff(TIME, claims))
hyp_ci_99 <- quantile(hyp_mean_diff, probs=c(0.005, 0.995))

cat("hyp: ", hyp_ci_99)
```

```
## hyp:  -0.004523563 1.888254
```

```
ggplot()+
  aes(hyp_mean_diff)+
  geom_density()+
  geom_vline(xintercept = quantile(hyp_mean_diff, c(0.005, 0.995)), color="red")
```




```
draw_ci(hyp_mean_diff, "Bootstrapped Hypothetical Mean")
```



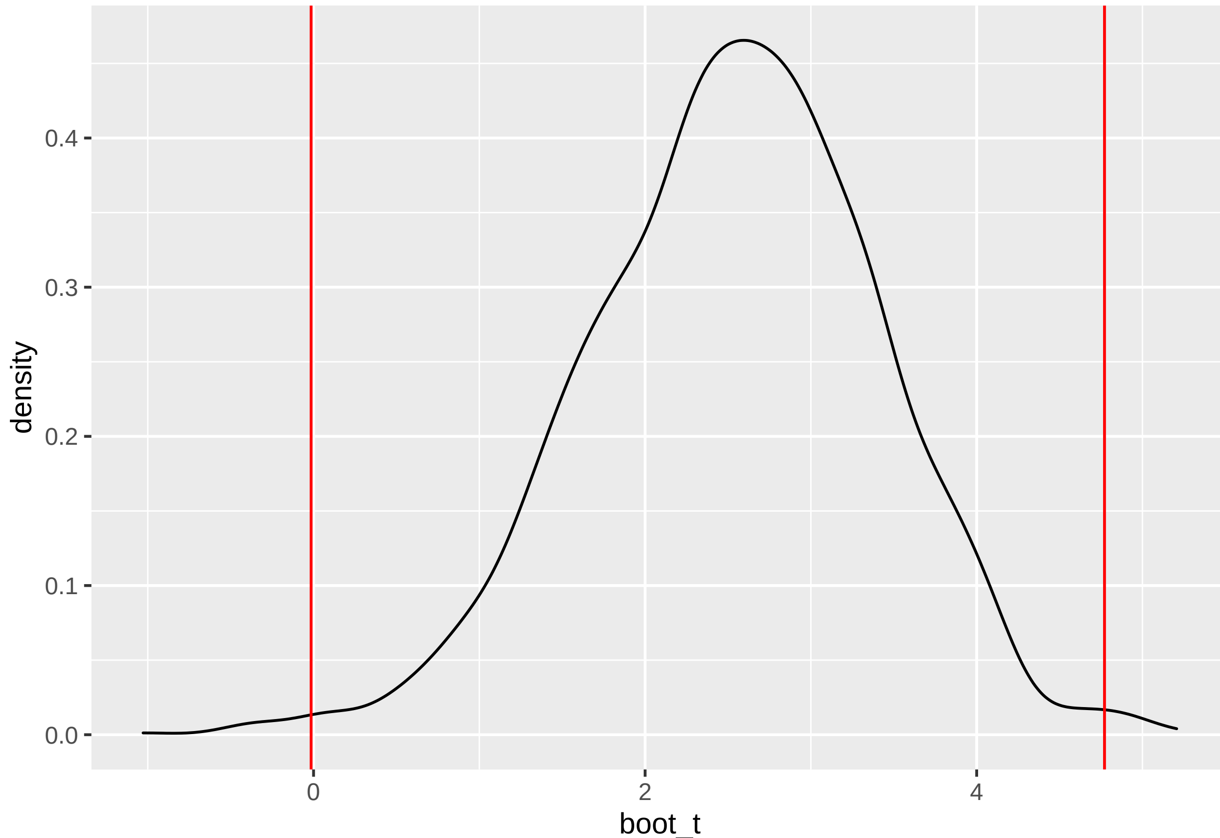
Note. As we can observe from the graph above, 0 clearly is not included in the 99% confidence interval, thus reject the null hypothesis.

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

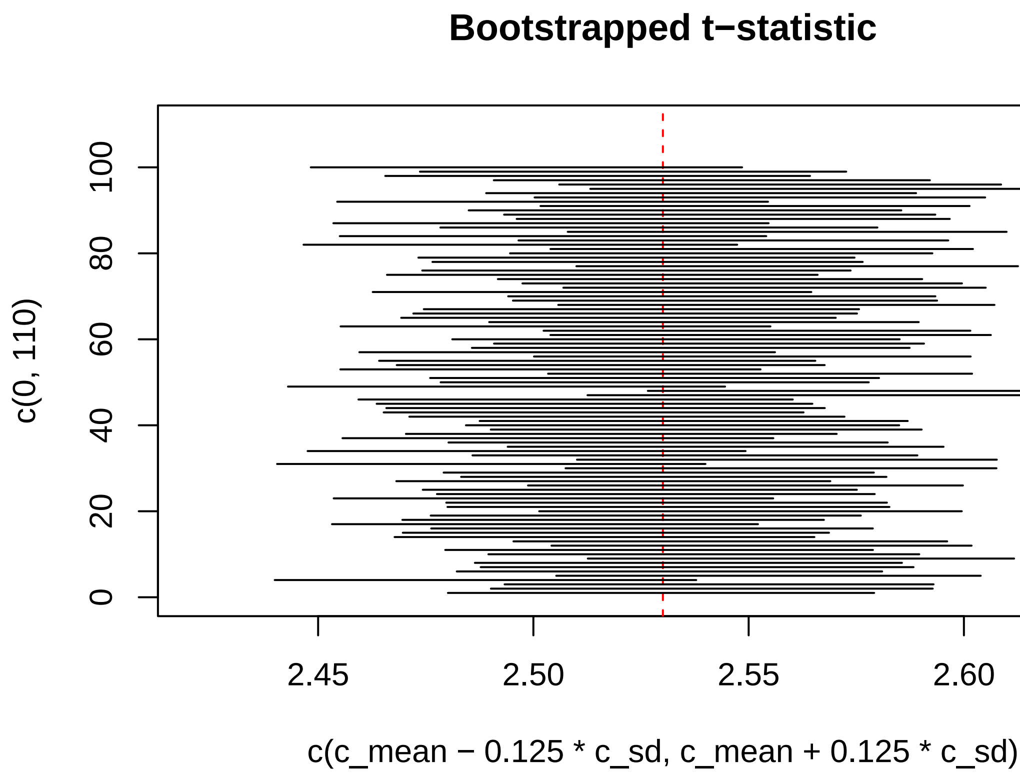
```
set.seed(round(seed))
boot_t <- replicate(2000, boot_t_stat(TIME, claims))
print(quantile(boot_t, c(0.005, 0.995)))
```

```
##         0.5%       99.5%
## -0.01465784  4.77127445
```

```
ggplot()+
  aes(boot_t)+
  geom_density()+
  geom_vline(xintercept = quantile(boot_t, c(0.005, 0.995)), color="red")
```



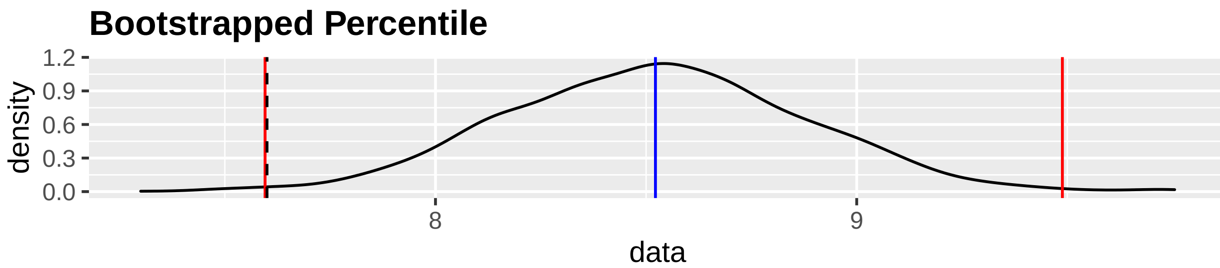

```
draw_ci(boot_t, title="Bootstrapped t-statistic")
```

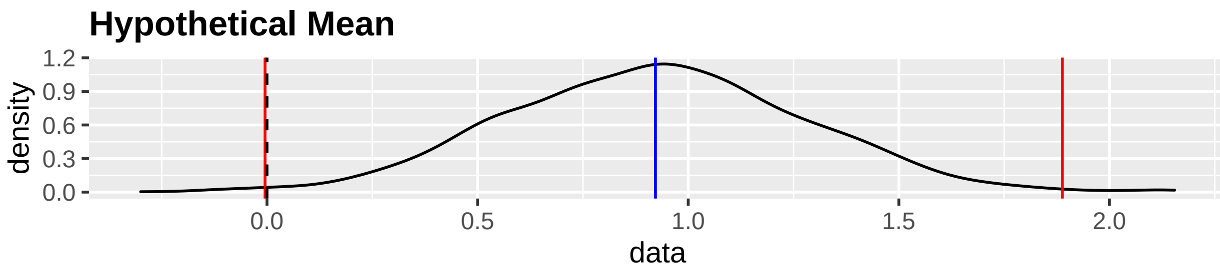


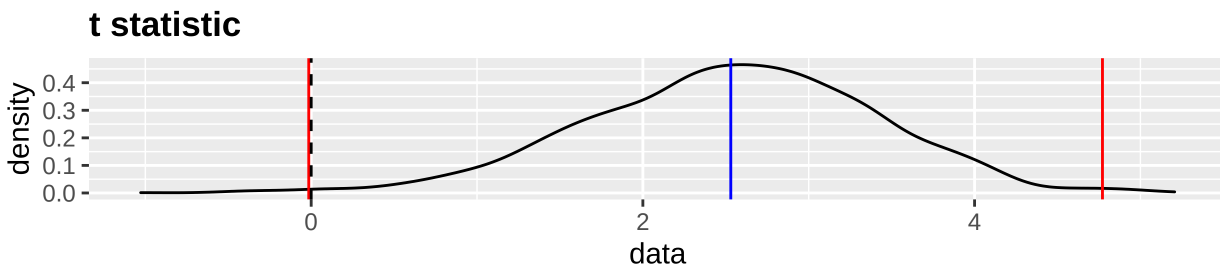
Note. As we can observe from the above graph, 0 does not included in the 99% CI range, in other words, by bootstrapping t-statisic from the sample the result reject the null hypothesis.

`(iv) Plot separate distributions of all three bootstraps above.`

```
set.seed(round(seed))
gggplot <- function(data, title){
  p <- ggplot()+
    aes(data)+
    geom_density()+
    geom_vline(xintercept = mean(data), color="blue")+
    ggtitle(title)+
    theme(plot.title=element_text(hjust=0, face="bold"))+
    geom_vline(xintercept = quantile(data, c(0.005, 0.995)), color="red")
  return(p)
}

p1 <- gggplot(resample, "Bootstrapped Percentile")+geom_vline(xintercept = claims, linetype="dashed")
p2 <- gggplot(hyp_mean_diff, "Hypothetical Mean")+geom_vline(xintercept = 0, linetype="dashed")
p3 <- gggplot(boot_t, "t statistic")+geom_vline(xintercept = 0, linetype="dashed")

ggarrange(p1, p2, p3, ncol=1, nrow=3)
```







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

Ans. The four methods all agree with each other, they all rejected the null hypothesis. This indicates that the claim from Verizon of 7.6 of service time is an inaccurate statement.