HW04

107070008

Question 1)

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?

```
pnorm(-3.7)
```

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

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?

```
2200000*pnorm(-3.7)
```

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

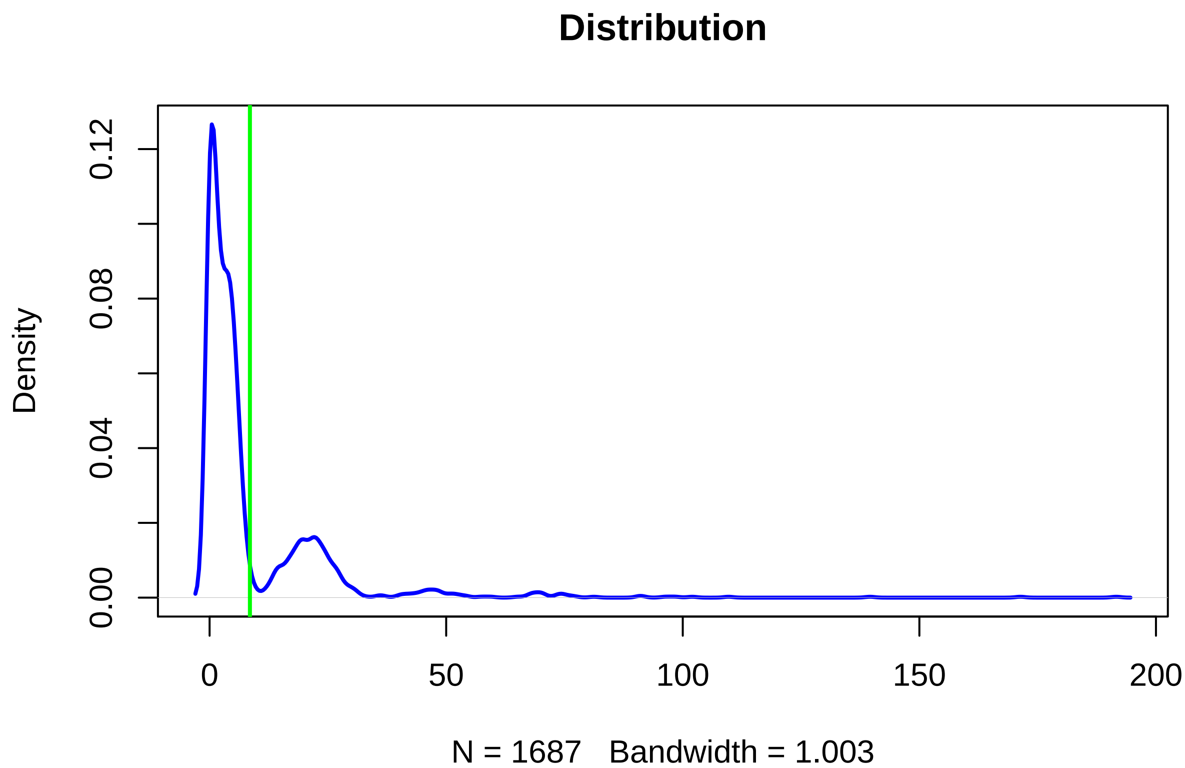
Question2

a. The Null distribution of t-values:

```
verizon <- read.csv("verizon.csv")
time <- verizon$Time
group <- verizon$Group
```

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

```
plot(density(time), lwd = 2, col = "blue", main = "Distribution")
mean_t <- mean(time)
abline(v= mean_t, lwd = 2, col = "green")
```



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

H0: $\mu = 7.6$

H1: $\mu \neq 7.6$

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

```
population_mean <- t.test(time, conf.level = 0.99)
population_mean
```

```
##
##  One Sample t-test
##
## data:  time
## t = 23.669, df = 1686, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 99 percent confidence interval:
##  7.593524 9.450495
## sample estimates:
## mean of x
##  8.522009
```

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

```r
hyp <- 7.6
sample_size <- length(time)
sample_mean <- mean(time)
sample_sd <- sd(time)
se <- (sample_sd/sqrt(sample_size))
t <- (sample_mean - hyp)/se
t
```

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

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

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

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

`For each test, the t-value is a way to quantify the difference between the population means and the p-value is the probability of obtaining a t-value with an absolute value at least as large as the one we actually observed in the sample data if the null hypothesis is actually true.`

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

`We will reject null hypothesis, because our p-value is smaller than the significant level.`

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

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

```r
compute_sample_mean <- function(sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  mean(resample)
}
sample_means <- replicate(2000, compute_sample_mean(time))
per_ci_99 <- quantile(sample_means, probs=c(0.005, 0.995))
per_ci_99
```

```
##     0.5%    99.5%
## 7.640527 9.562081
```

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

```r
sample0 = sample(time, sample_size)
boot_mean_diffs <- function(smaple0, hyp) { #mean_hyp??
  resample <- sample(smaple0, length(sample0), replace = TRUE)
  return(mean(resample) - hyp)
}
set.seed(42379878)
```

```r
num_boots <- 2000
mean_diffs <- replicate(
  num_boots,
  boot_mean_diffs(time, hyp)
)
diff_ci_99 <- quantile(mean_diffs, probs=c(0.005, 0.995))
diff_ci_99
```

```
##        0.5%       99.5%
## -0.01417365  1.89941769
```

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

```r
boot_t_stat <- function(sample0, hyp) {
    resample <- sample(sample0, length(sample0), replace=TRUE)
    diff <- mean(resample) - hyp
    se <- sd(resample)/sqrt(length(resample))
    return(diff/se)
}
set.seed(2346786)
num_boots <- 2000
t_boots <-replicate(num_boots, boot_t_stat(time, hyp))
mean(t_boots)
```

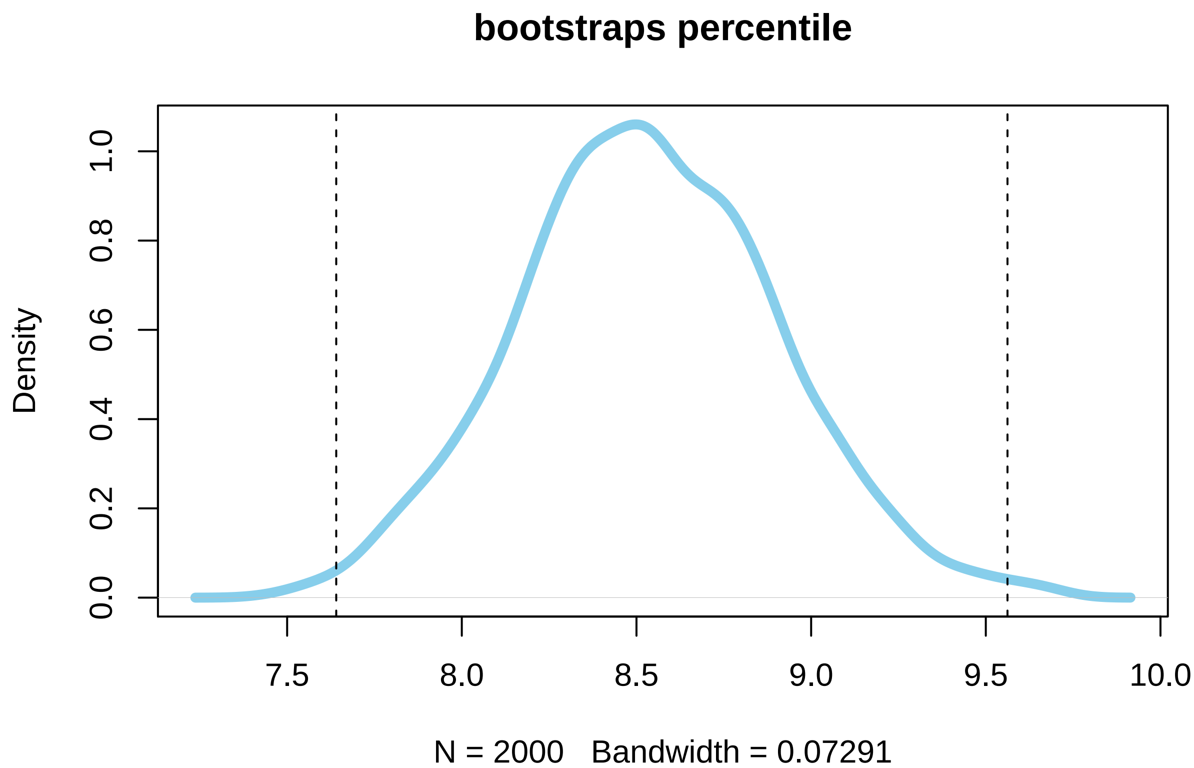
```
## [1] 2.536774
```

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

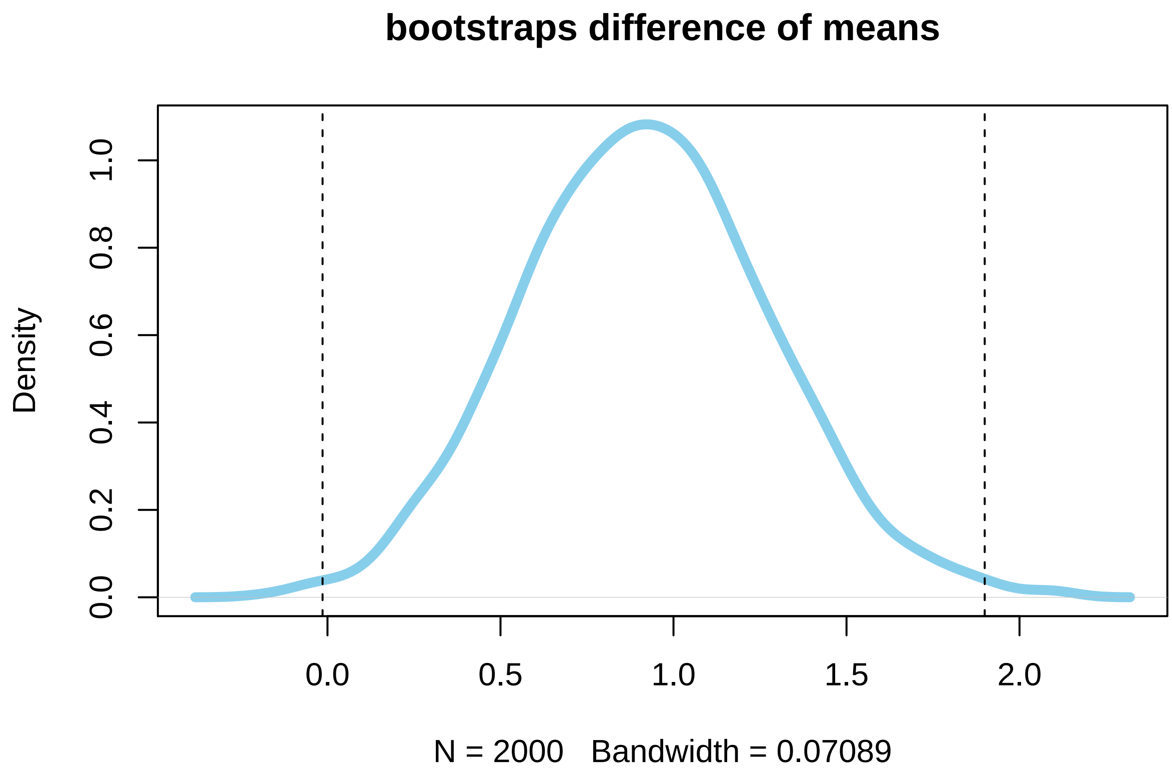
```
##      0.5%     99.5%
## 0.2434266 4.6637516
```

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

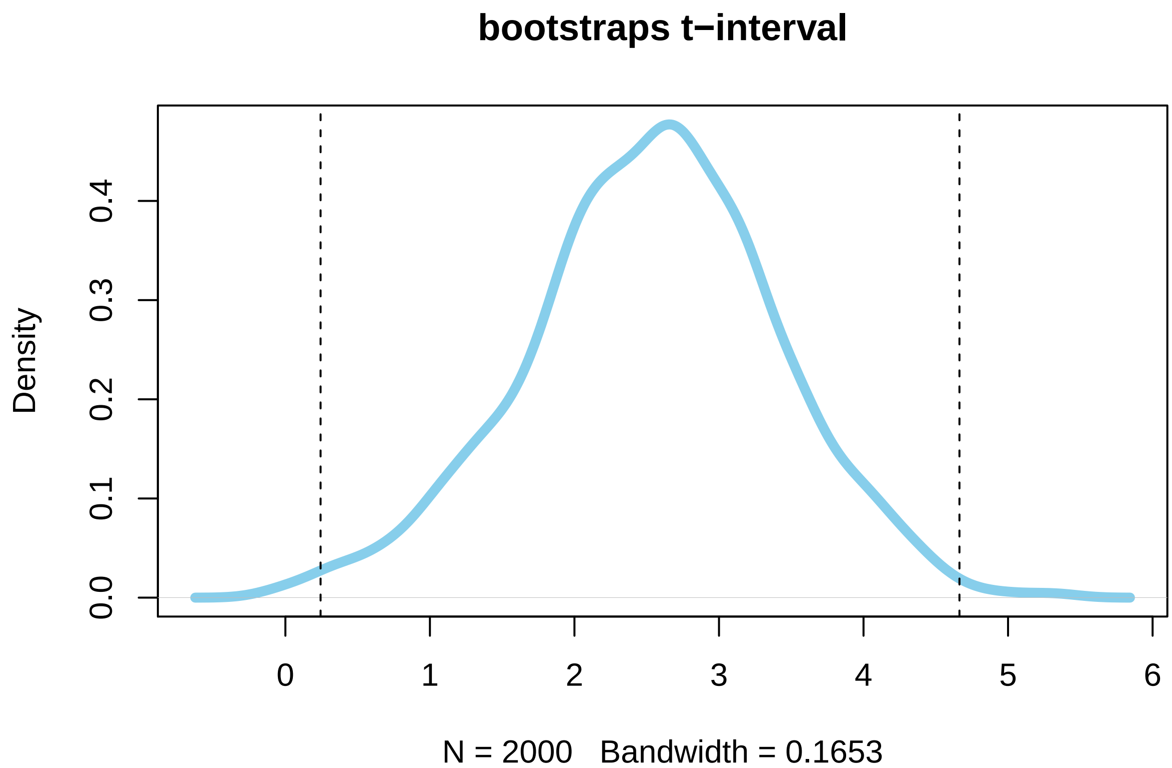
```r
plot(density(sample_means), lwd = 5, col = "skyblue", main = "bootstraps percentile")
abline(v=per_ci_99, lty="dashed")
```




```
plot(density(mean_diffs),lwd = 5, col="skyblue", main = "bootstraps difference of means")
abline(v=diff_ci_99, lty="dashed")
```




```
plot(density(t_boots),lwd = 5, col="skyblue", main = "bootstraps t-interval")
abline(v=t_ci_99, lty="dashed")
```



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

```
In traditional test, bootstrapped percentile, and bootstraped t-interval, 99% ci does
not contain zero, so we can reject the Verizon's claim. On the other hand, bootstapped
difference of means contain zero in 99% ci. Therefore, those four method do not agree
with each other on test.
```