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?

Code:

```
pnorm(-3.7)
```

Result:

```
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?

Code:

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

Result:

```
237.1594
```

Question 2)

a. The Null distribution of t-values:

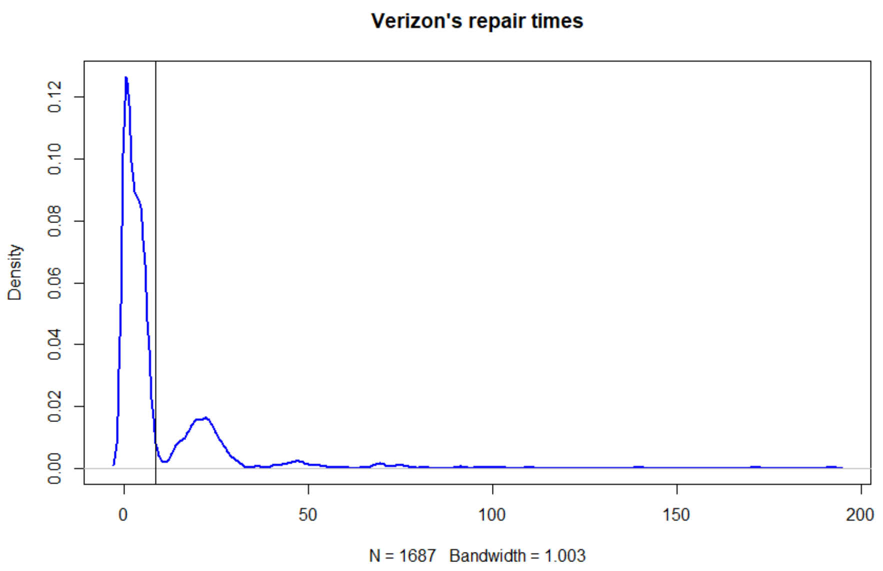
i. Visualize the distribution of Verizon's repair times, marking the mean with a vertical line

Code:

```
#read data
data<-read.csv("verizon.csv",header = TRUE)
#get variable from data
Time<-data$Time
Group<-data$Group

plot_centrality <- function(distr, title) {
  # Plot the distribution
  plot(density(distr), col="blue", lwd=2, main = title)
  # Add vertical lines showing mean
 abline(v=mean(distr))
}
plot_centrality(Time,title="Verizon's repair times")
```

Result:



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

Description:

H0:μ=7.6

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

Code:

```
mean(Time)
Time_se <- sd(Time)/sqrt(length(Time))
CI99 <- mean(Time)+ c(-2.58,2.58)* Time_se
```

Result:

mean:

8.522009

99% CI:

7.593073 9.450946

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

Code:

```
t<- (mean(Time)-7.6)/Time_se
df <- length(Time)-1
p <- 1- pt(t,df)
```

Result:

t:

2.560762

p-value:

0.005265342

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

Description:

t=2.56, means that 2.56 standard errors the sample mean is away from the hypothesized population mean (7.6).

p=0.0053, means there is only a 0.53% probability that the results(μ<7.6) happened by chance.

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

Description:

I think the advertising claim from Verizon is exaggerated (**rejected**), because according to the t-test, there's only **0.53%** probability that the population mean is lower than 7.6, it way too rare to happen in reality.

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

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

Code:

```
num_boot <- 2000
sample_statistic <- function(stat_function, sample0) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  stat_function(resample)
}
sample_means <- replicate(num_boot, sample_statistic(mean,Time))
quantile(sample_means, probs = c(0.005, 0.995))
```

Result:

```
    0.5%    99.5%
7.658457 9.462797
```

ii. Bootstrapped Difference of Means:

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

Code:

```
boot_mean_diffs <- function(sample0, mean_hyp) {
  resample <- sample(sample0, length(sample0), replace=TRUE)
  return( mean(resample) - mean_hyp )
}
mean_diffs <- replicate(
  num_boot,
  boot_mean_diffs(Time, 7.6)
)
diff_ci_99 <- quantile(mean_diffs, probs= c(0.005, 0.995))
```

Result:

```
    0.5%    99.5%
0.031215 1.855009
```

iii. Bootstrapped t-Interval: What is 99% CI of the bootstrapped t-statistic?

Code:

```
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_boots <- replicate(num_boot, boot_t_stat(Time, 7.6))
mean(t_boots)
t_ci_99 <- quantile(t_boots, probs=c(0.005, 0.995))
```

Result:

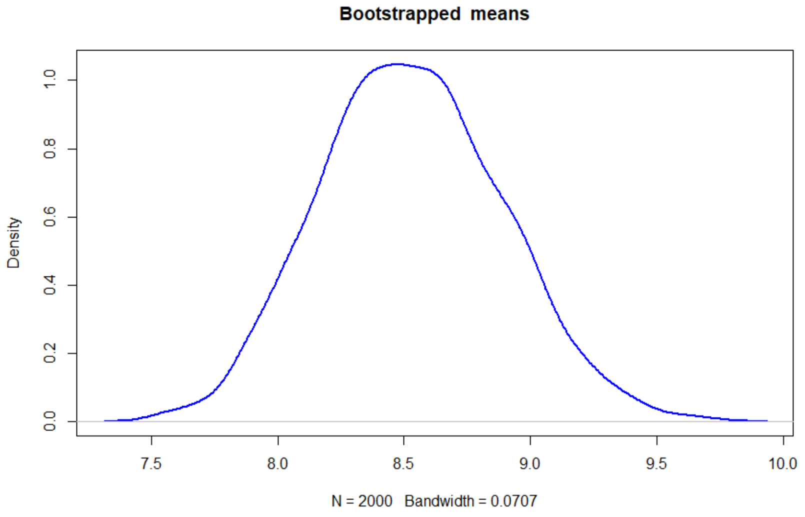
```
    0.5%    99.5%
0.185224 4.706900
```

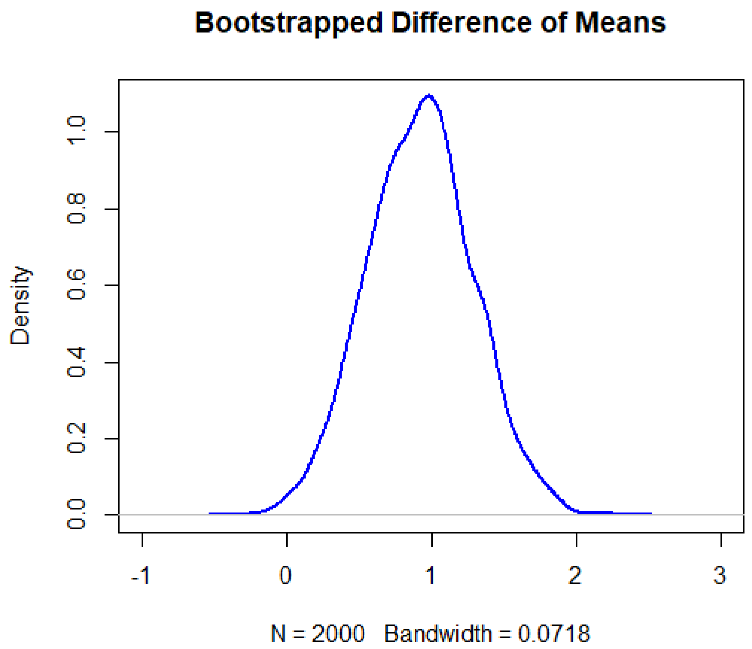
iv. Plot separate distributions of all three bootstraps above (for ii and iii make sure to include zero on the x-axis)

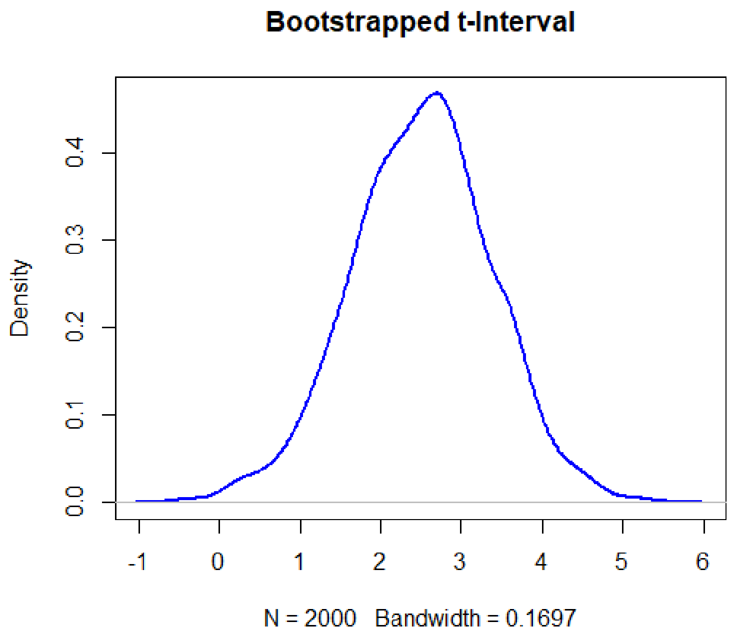
Code:

```
plot(density(sample_means), lwd=2, col="blue", main="Bootstrapped  means")
plot(density(mean_diffs), xlim=c(-1,3), col="blue", lwd=2,
main="Bootstrapped Difference of Means")
plot(density(t_boots), xlim=c(-1,6), col="blue", lwd=2, main="Bootstrapped
t-Interval")
```

Result:







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

Description:

traditional test:

t=2.560762 (p=0.0053), the possibility is lower than 0.05, so we can **reject** the Verizon claim.

Bootstrapping:

99% CI of the mean:[7.658457,9.462797], it doesn't contain 7.6 so we can **reject** the Verizon claim.

99% CI of the mean difference:[0.031215,1.855009], it doesn't contain 0 so we can **reject** the Verizon claim.

99% CI of t-Interval:[0.185224,4.706900], it doesn't contain 0 so we can **reject** the Verizon claim.

Conclusion: four methods all agree with each other.