108070017

BACS_HomeWork4_Report

1-(a)

```
#convert Z-score to probability
pnorm(-3.7)
```

0.0001077997

The probability that a randomly chosen app from Google's app store will turn off the Verify security feature is 0.0001077997.

1-(b)

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

237.1594

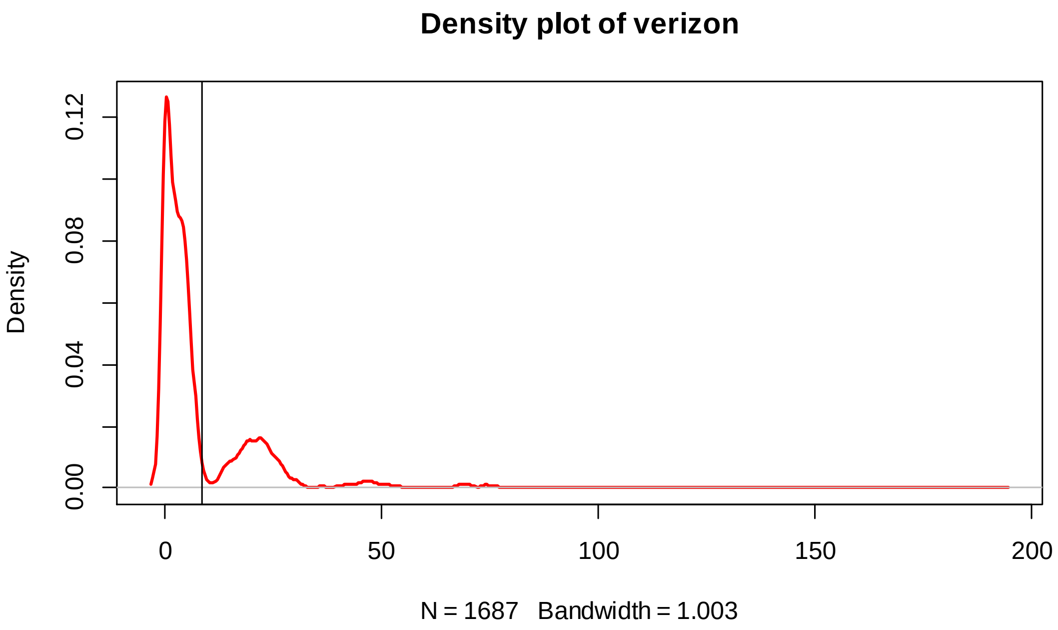
About 237 apps on the Play Store Google expected would maliciously turn off the Verify feature once installed.

2-(a)-i

```
#load data
verizon = read.csv("C:/Users/USER/Desktop/verizon.csv", header = TRUE)

#the density plot of data
plot(density(verizon$Time), col="red", lwd=2,
        main = "Density plot of verizon")

#add vertical line at mean
abline(v=mean(verizon$Time), lwd=1)
```



2-(a)-ii

H0: mu = 7.6 minutes

H1: mu $\neq$ 7.6 minutes

2-(a)-iii

```
#population mean
mean(verizon$Time)

#99% confidence interval
left <- 8.522009 - 2.58*(sd(verizon$Time)/(1687^(0.5)))
right <- 8.522009 + 2.58*(sd(verizon$Time)/(1687^(0.5)))
cat ("99% confidence:",left,right)
```

Population mean = 8.522009
99% CI = (7.593073, 9.450945)

2-(a)-iv

```
#t-statistic
t = (8.522009-7.6)/(sd(verizon$Time)/(1687^(0.5)))
t

#p-value
df = 1687-1
p = 1 - pt(t, df)
p
```

t-statistic = 2.560761
p-value = 0.005265363

2-(a)-v

t-statistic is used to determine that whether the difference of population mean and hypothetical mean is significant or not.
p-value is a measure of the probability that an observed difference could have occurred by random chance.

2-(a)-vi

By setting the significant level to 0.05, we can find that the p-value is smaller than the significant level. In this way, we reject the null hypothesis, it means the advertising claim is not correct.

2-(b)-i

```
set.seed(66)
number_boots <- 2500

#bootstrap function
boot_mean<-function(sample0) {
   resample <-sample(sample0, length(sample0), replace=TRUE)
   return(mean(resample))}
mean2 <- replicate(number_boots, boot_mean(verizon$Time))
cat("99% CI of estimated mean:",quantile(mean2,c(0.005,0.995)))
```

The bootstrapped 99% CI of the mean = (7.660614, 9.457096)

2-(b)-ii

```
set.seed(66)
boot_mean_diff<-function(sample0, hyp_mean) {
   resample <-sample(sample0, length(sample0), replace=TRUE)
   return(mean(resample) - hyp_mean)}
manager_hyp = 7.6
mean_diff<-replicate(number_boots, boot_mean_diff(verizon$Time, manager_hyp))
cat("99% CI of estimated mean difference:",quantile(mean_diff,c(0.005,0.995)))
```

The 99% CI of the bootstrapped difference between the population mean and the hypothesized mean = (0.04670074, 1.853749)

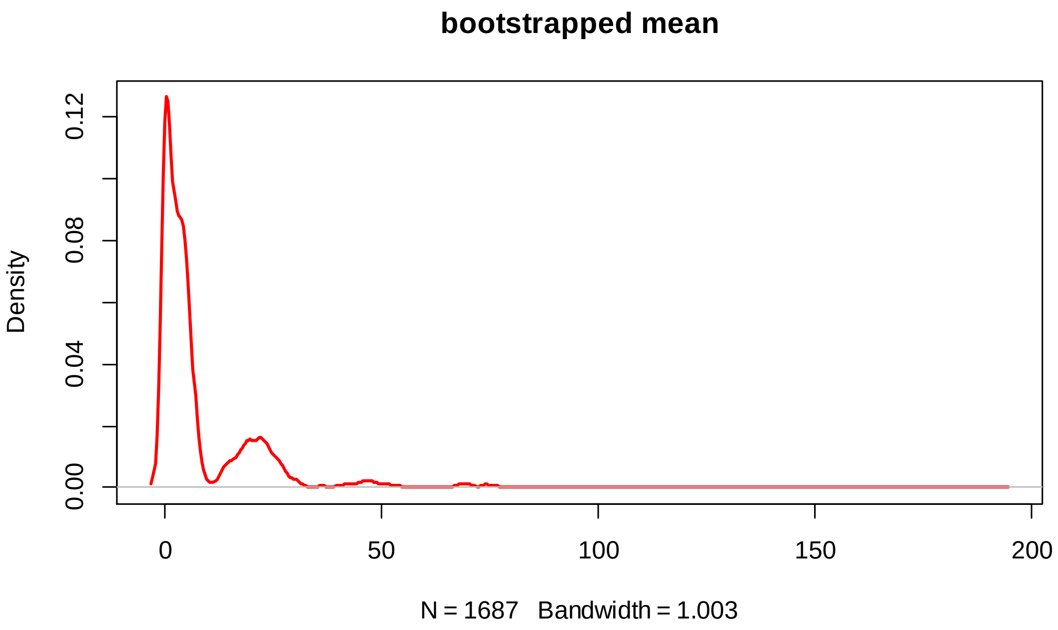
2-(b)-iii

```
boot_t_stat<-function(sample0, mean_hyp) {
   resample <- sample(sample0, length(sample0), replace=TRUE)
   diff <- mean(resample) - mean_hyp
   resample_se<-sd(resample)/sqrt(length(resample))
   return( diff/resample_se)}
t_boots <- replicate(number_boots,boot_t_stat(verizon$Time, manager_hyp))
cat("99% CI of mean bootstrapped t-statistic:",quantile(t_boots,c(0.005,0.995)))
```

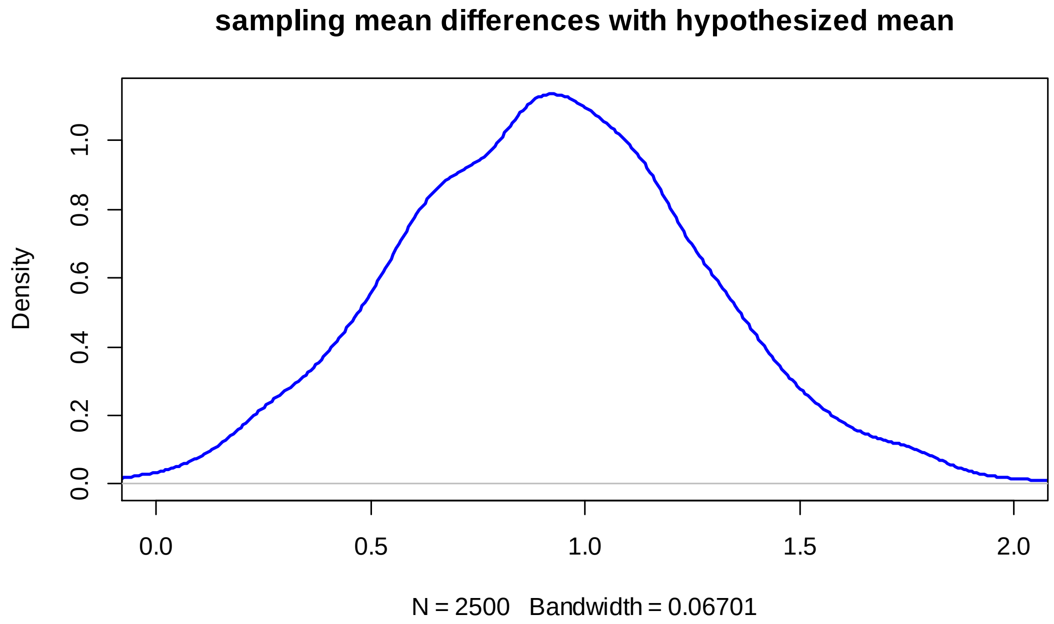
The 99% CI of the bootstrapped t-statistic = (0.2002863, 4.643956)

2-(b)-iv

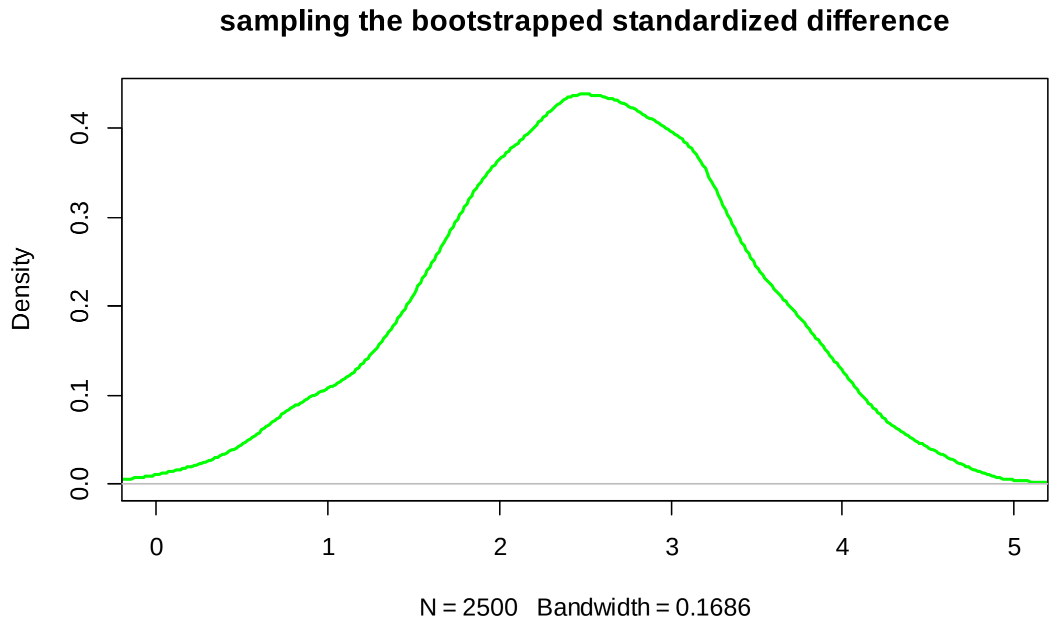
```
#plot for i
set.seed(66)
plot(density(verizon$Time), col="red", main="bootstrapped mean", lwd = 2)
resamples <-replicate(number_boots, sample(verizon$Time,
length(verizon$Time), replace=TRUE))
plot_resample_mean<-function(sample_i)
sample_means<-apply(resamples, 2, FUN=plot_resample_mean)
sample_means_mean <- mean(sample_means)
```




```
#plot for ii
set.seed(66)
plot(density(mean_diff),xlim=c(0, 2),
        main = "sampling mean differences with hypothesized mean",
        lwd = 2, col="blue")
```




```
#plot for iii
set.seed(66)
plot(density(t_boots),
        main = "sampling the bootstrapped standardized difference",
        xlim=c(0, 5), lwd = 2, col="green")
```



2-(c)

Yes, no matter we use which method to test the data, it gets similar answers.