BACS HW5

106070020

2021年3月27日

Problem 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? (report a precise decimal fraction, not a percentage)

```
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?

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

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

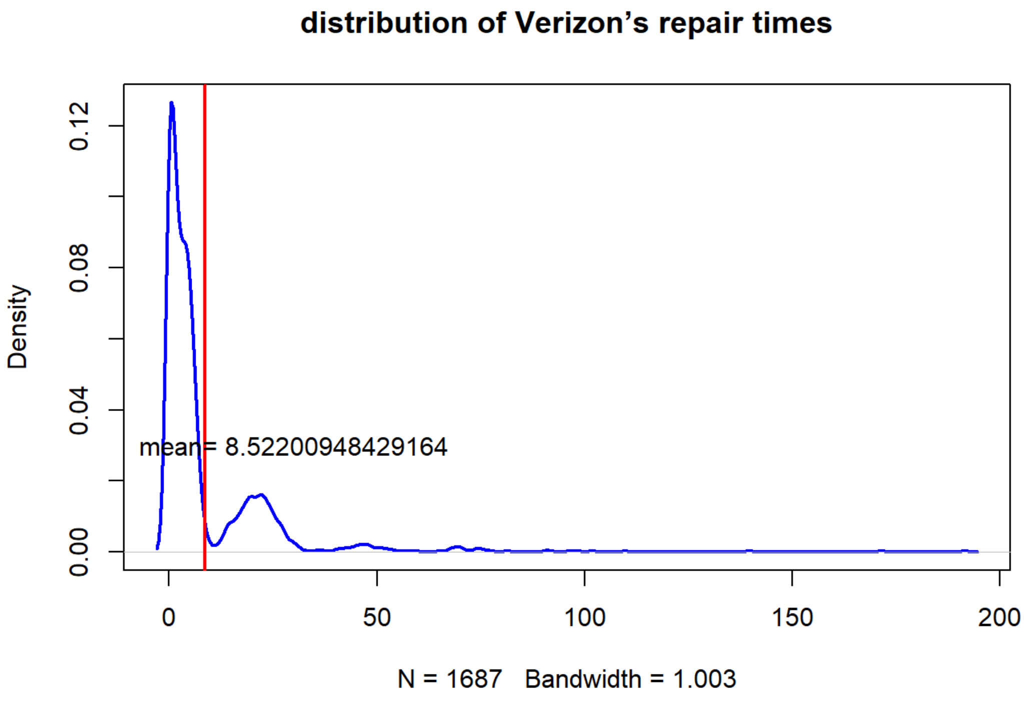
Problem 2

```
ver <- read.csv("C:/Users/eva/Downloads/verizon.csv", header = T)
```

(a) The Null distribution of t-values:

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

```
vert <- ver$Time
a<-paste("mean=",mean(vert))
{plot(density(ver$Time), lwd=2, col="blue", main="distribution of Verizon’s repair times")
abline(v=mean(vert), lwd=2, col="red")
text(x=30, y=0.03,a)}
```



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

> PUC want to test that the claims of Verizon that they take 7.6 minutes to repair phone services for its customers on average, and they seek to verify this claim at 99% confidence. H0: mean=7.6 min, H1: mean!= 7.6 min.

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

```
x<-mean(vert) #sample mean
s<-sd(vert)/sqrt(length(vert)) #standard_error
CI_99<-c(x-2.58*s,x+2.58*s)
x # The estimation of population mean, as sample mean is the unbiased estimator of population mean.
```

```
## [1] 8.522009
```

```
CI_99 #95% C.I
```

```
## [1] 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.

```
s<-sd(vert)/sqrt(length(vert))
t<-(mean(vert)-7.6)/s
t # t statistic
```

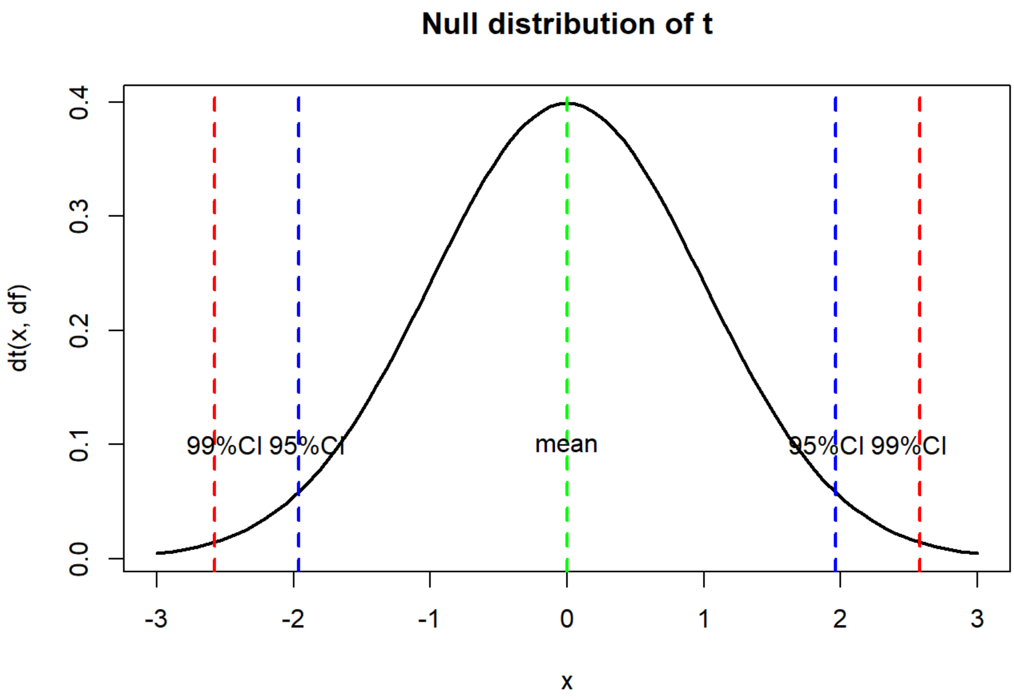
```
## [1] 2.560762
```

```
df=length(vert)-1
p_value<- 1-pt(t,df)
p_value #p-value of the test
```

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

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

```
{curve(dt(x,df),xlim = c(-3,3), lwd=2, main="Null distribution of t")
abline(v=qt(0.005,df), lty=2, lwd=2, col="red")
abline(v=qt(0.995,df), lty=2, lwd=2, col="red")
abline(v=qt(0.025,df), lty=2, lwd=2, col="blue")
abline(v=qt(0.975,df), lty=2, lwd=2, col="blue")
abline(v=qt(0.5,df), lty=2, lwd=2, col="green")
text(x=0, y=0.1, "mean", lwd=3)
text(x=-2.5, y=0.1, "99%CI", lwd=3)
text(x= 2.5, y=0.1, "99%CI", lwd=3)
text(x=-1.9, y=0.1, "95%CI", lwd=3)
text(x= 1.9, y=0.1, "95%CI", lwd=3)
}
```



> In the graph, I show the two -tail test. We can see the range of the 95% CI and the 99% CI. The p-value is the area under the curve that exceed the range of confidence interval. Thus, if alpha=0.05, then its p-value is the sum of the area under the curve where exceed the 95% CI range.

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

```
reject<-function(t,df){
  if(t<qt(0.005,df)){
    return(TRUE)
  }else if(t>qt(0.995,df)){
    return (TRUE)
  }else{
    return (FALSE)
  }
}
reject_p<-function(p_value){
  if(p_value<0.005){
    return(TRUE)
  }else if(p_value>0.995){
    return (TRUE)
  }else{
    return (FALSE)
  }
}
reject(t,df) # Whether to reject H0 by using t statistic under the 99% C.I.
```

```
## [1] FALSE
```

```
reject_p(p_value)# Whether to reject H0 by using p value under the 99% C.I.
```

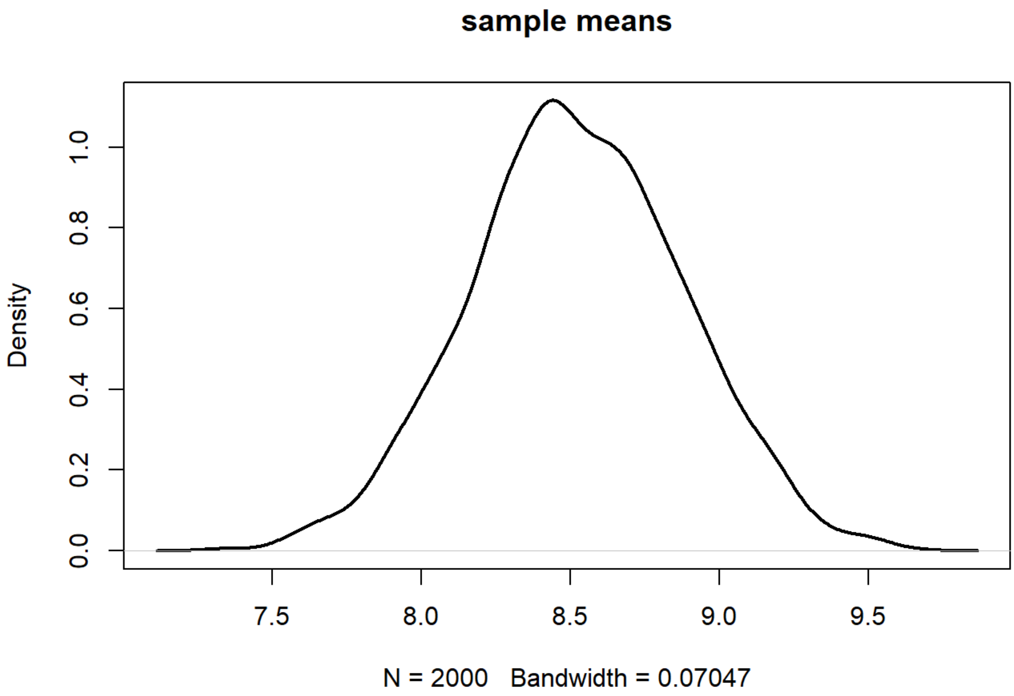
```
## [1] FALSE
```

> My conclusion about the advertising claim from this t-statistic is do not reject H0. According to the result of t statistic and p-value in this claim, those values are not enough for us to reject the null hypothesis (mean time = 7.6) under the 99% cl.

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

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

```
set.seed(112128)
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, vert))
plot(density(sample_means), lwd=2, main="sample means")
```



```
CI99_2 <- quantile(sample_means, probs = c(0.005, 0.995))
CI99_2 # 99% CI interval
```

```
##     0.5%    99.5%
## 7.608253 9.442853
```

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

```
set.seed(012321888)
boot_mean_diffs <- function(sample0, mean_hyp){
resample <- sample(sample0, length(sample0), replace=TRUE)
return( mean(resample) - mean_hyp )}
set.seed(1912321299)
num_boots <- 2000
mean_diffs <- replicate(
  num_boots,
  boot_mean_diffs(vert, 7.6) )
diff_ci_99 <- quantile(mean_diffs, probs=c(0.005, 0.995))
diff_ci_99
```

```
##        0.5%       99.5% 
## 0.008724363 1.846821310
```

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

```
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(08005)
num_boots <- 2000
t_boots <- replicate(num_boots, boot_t_stat(vert, 7.6))
mean(t_boots)
```

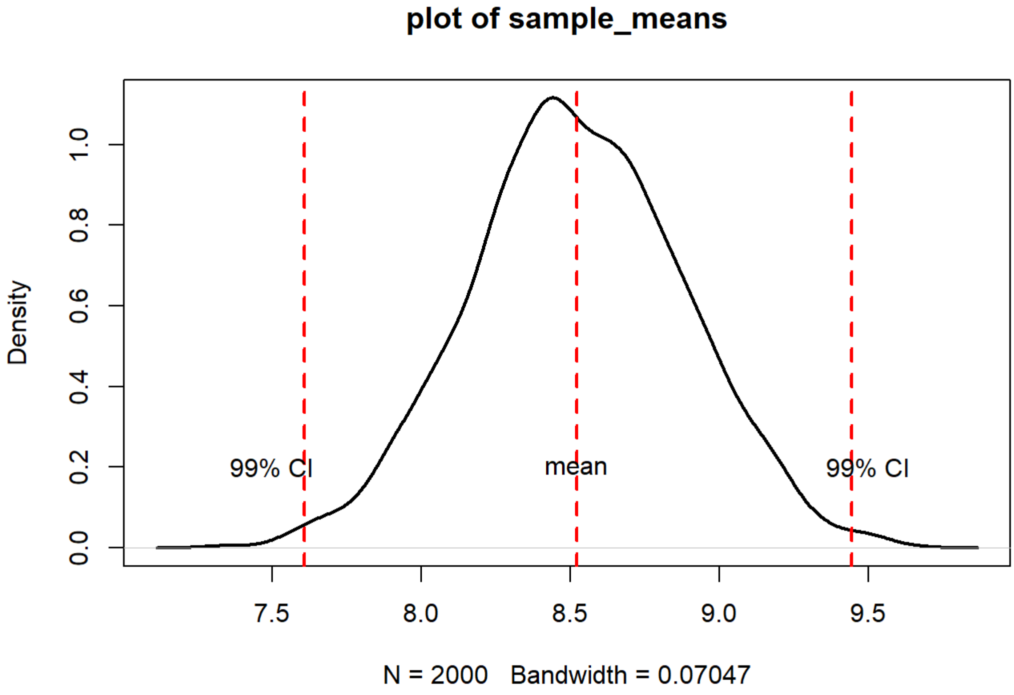
```
## [1] 2.511169
```

```
t99CI <- quantile(t_boots, probs=c(0.005, 0.995))
t99CI
```

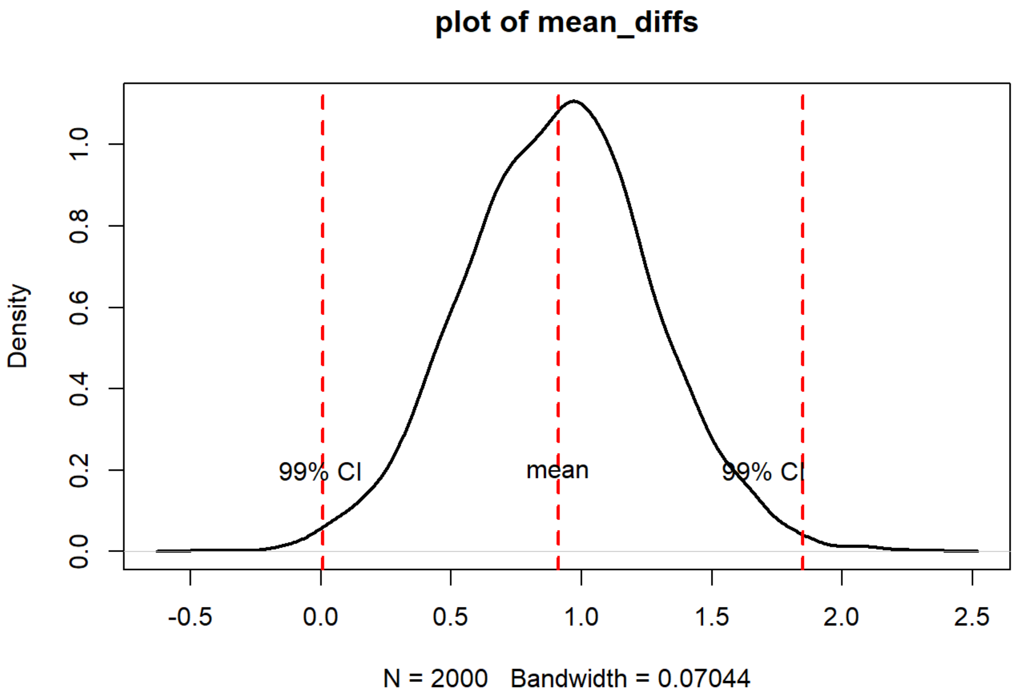
```
##       0.5%      99.5% 
## -0.1344704  4.6451525
```

(iv) Plot separate distributions of all three bootstraps above

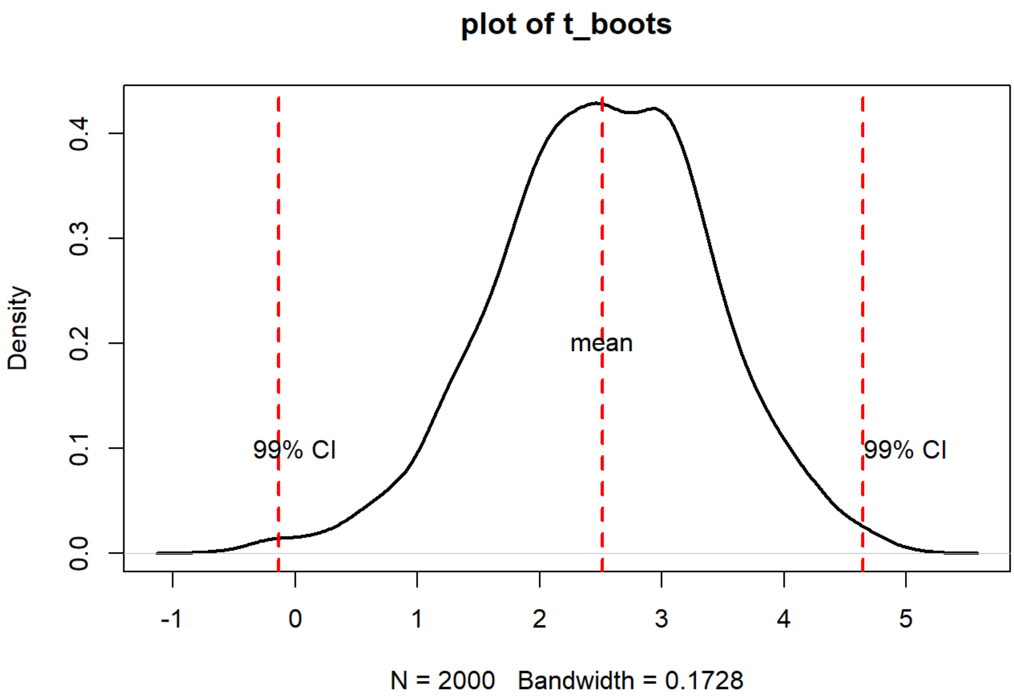
```
{plot(density(sample_means),lwd=2,main = "plot of sample_means")
  abline(v=CI99_2[1], lty=2, lwd=2, col="red")
  abline(v=CI99_2[2], lty=2, lwd=2, col="red")
  abline(v=mean(sample_means), lty=2, lwd=2, col="red")
  text(x=mean(sample_means), y=0.2, "mean", lwd=3)
  text(x= 7.5, y=0.2, "99% CI", lwd=3)
  text(x= 9.5, y=0.2, "99% CI", lwd=3)
}
```



```
{plot(density(mean_diffs),lwd=2,main = "plot of mean_diffs")
  abline(v=diff_ci_99[1], lty=2, lwd=2, col="red")
  abline(v=diff_ci_99[2], lty=2, lwd=2, col="red")
  abline(v=mean(mean_diffs), lty=2, lwd=2, col="red")
  text(x=mean(mean_diffs), y=0.2, "mean", lwd=3)
  text(x= 0, y=0.2, "99% CI", lwd=3)
  text(x= 1.7, y=0.2, "99% CI", lwd=3)
}
```



```
{plot(density(t_boots),lwd=2,main = "plot of t_boots")
  abline(v=t99CI[1], lty=2, lwd=2, col="red")
  abline(v=t99CI[2], lty=2, lwd=2, col="red")
  abline(v=mean(t_boots), lty=2, lwd=2, col="red")
  text(x=mean(t_boots), y=0.2, "mean", lwd=3)
  text(x= 0, y=0.1, "99% CI", lwd=3)
  text(x= 5, y=0.1, "99% CI", lwd=3)
  }
```



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

```
reject<-function(t,df){
  if(t<qt(0.005,df)){
    return(TRUE)
  }else if(t>qt(0.995,df)){
    return (TRUE)
  }else{
    return (FALSE)
  }
}
reject(t,df)
```

```
## [1] FALSE
```

```
reject_a <- function(m, CI){
  if(m < CI[1] | m > CI[2]){
    return(TRUE)
  }else{
    return (FALSE)
  }
}
reject_a(7.6, CI99_2)
```

```
## [1] TRUE
```

```
reject_b <- function(CI){
  if(0 < CI[1] | 0 > CI[2]){
    return(TRUE)
  }else{
    return (FALSE)
  }
}
reject_b(diff_ci_99)
```

```
## [1] TRUE
```

```r
reject_c <- function(CI){
  if(0 < CI[1] | 0 > CI[2]){
    return(TRUE)
  }else{
    return (FALSE)
  }
}
reject_c(t99CI)
```

```
## [1] FALSE
```

> According to the results above, we will not reject H0 base on traditional test and bootstrapped t-Interval, while we will reject H0 when using bootstrapped percentile and bootstrapped difference of means, so they are not agree with each other on the test. According to the confidence interval in b(i) and b(ii), their lower bound are both very close to 7.6 and 0, while 7.6 and 0 are not included in the confidence interval, so the H0 is rejected in these two situation. 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.