HW5

108048110

3/16/2022

## **Question1**

Recall the form of t-tests, where

Your colleague, a data analyst, wishes to test whether the mean () usage time is higher than the usage time of the company’s previous smartwatch released two years ago ():

After collecting data from just customers, he informs you that he has found and .  
Your colleague believes that we cannot reject the null hypothesis at .

Consider the scenarios (a – d) independently using the simulation tool. For each scenario, start with the initial parameters above, then adjust them to answer the following questions:

#### **a.) Your colleague missed many older customers who you suspect might use the product much less every day.**

**1.** **Systematic or random error?**

*Ans.*

##### This scenario creates systematic error. Sampling bias occurs when some members of a population are more likely to be included in the study than others, it reduces the generalizability of your findings because your sample is not representative of the whole population.

**2.** **Which part of the t-statistic or significance would be affected?**

*Ans.*

##### The and will increase.

**3. Will it increase or decrease our power to reject the null hypothesis?**

*Ans.*

##### By left shifting the alternative hypothesis, it will increase power to reject the null hypothesis.

**4.** **Which kind of error (Type I or Type II) becomes more likely because of this scenario?**

*Ans.*

##### Type I error becomes more likely to happen because of this scenario.

#### **b.) You find that 20 of the respondents should be removed from the data. These 20 people are just like the others in every other respect.**

**1.** **Systematic or random error?**

*Ans.*

##### This scenario creates random error.

**2.** **Which part of the t-statistic or significance would be affected?**

*Ans.*

##### , the number of sample will decrease.

**3.** **Will it increase or decrease our power to reject the null hypothesis?**

*Ans.*

##### By right shifting the alternative hypothesis, your power to reject the null hypothesis will decrease.

**4.** **Which kind of error (Type I or Type II) becomes more likely because of this scenario?**

*Ans.*

##### Type II error is more likely to occur.

#### **c.) Relaxing confidence criteria to just 90%**

**1.** **Systematic or random error?**

*Ans.*

##### This scenario has neither of the errors.

**2.** **Which part of the t-statistic or significance would be affected?**

*Ans.*

##### The value of the t-statistics’ will be affected.

**3.** **Will it increase or decrease our power to reject the null hypothesis?**

*Ans.*

##### Your power to reject the null hypothesis decrease because of the increase value of the critical value.

**4.** **Which kind of error (Type I or Type II) becomes more likely because of this scenario?**

*Ans.*

##### Type II error will be more likely to occur.

#### **d.) You feel the method under-reports usage for samples who are active on weekends, whereas it over-reports usage of older users.**

#### **1.** Systematic or random error?

*Ans.*

##### This scenario creates systematic error. Sampling bias occurs when some members of a population are more likely to be included in the study than others, it reduces the generalizability of your findings because your sample is not representative of the whole population.

**2. Which part of the t-statistic or significance would be affected?**

*Ans.*

##### The and will increase.

**3.** **Will it increase or decrease our power to reject the null hypothesis?**

##### *Ans.*

##### By left shifting the alternative hypothesis, it will increase power to reject the null hypothesis.

##### **4.** **Which kind of error (Type I or Type II) becomes more likely because of this scenario?**

*Ans.*

##### Type I error becomes more likely to happen because of this scenario.

## **Question2**

Verizon claims that they take no more than 7.6 minutes on average to repair phone services for its customers. A recent sample of repair times was seeked to verify this claim at 99% confidence.

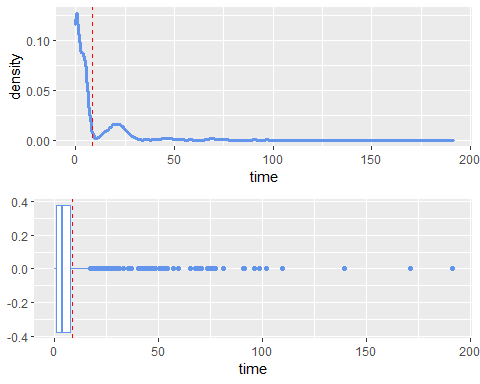
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.1.2

library(ggpubr)

## Warning: package 'ggpubr' was built under R version 4.1.2

# calculate the standard error  
sderr <- function(data){  
 return(sd(data)/sqrt(length(data)))  
}  
  
# Using p value to decide whether or not to reject the null hypothesis.  
p\_test <- function(p, sig\_level){  
 if(p > sig\_level) cat("We do not have enough evidence to reject the Null hypothesis!")  
 else cat("We have enough evidence to reject the Null hypothesis!")  
}  
  
# Using t value to decide whether or not to reject the null hypothesis.  
t\_test <- function(t, cutoff){  
 if(t>cutoff[1] && t<cutoff[2]) cat("We have enough evidence to reject the Null hypothesis!")  
 else cat("We do not have enough evidence to reject the Null hypothesis!")  
}  
  
# Computing confidence interval by formula.  
CI <- function(data, confidence){  
 v1 = (1-confidence)/2  
 v2 = confidence+v1  
 ci <- mean(data)+qt(c(v1, v2), length(data))\*sderr(data)  
 return(ci)  
}  
  
# Load the data.  
verizon <- read.csv('verizon.csv')  
time <- verizon$Time  
claims = 7.6  
  
# Visualizing it.  
p1 <- ggplot()+aes(time)+geom\_density(color="cornflowerblue", lwd=1.2)+geom\_vline(xintercept = mean(time), color="red", lty="dashed")  
p2 <- ggplot()+aes(time)+geom\_boxplot(color="cornflowerblue")+geom\_vline(xintercept = mean(time), color="red", lty="dashed")  
ggarrange(p1, p2, ncol=1, nrow=2)



#### Plotting functions

plt <- function(data, claims=claims, confidence=0.99){  
 v1 = (1-confidence)/2  
 v2 = confidence+v1  
 ggplot()+  
 aes(data)+  
 geom\_density()+  
 geom\_vline(xintercept = mean(data), color="cornflowerblue")+  
 geom\_vline(xintercept = quantile(data, c(v1, v2)), color="red", lty="dashed")+  
 geom\_vline(xintercept = claims, lwd=1.2)  
}  
  
  
plotdist <- function(xseq, xdens, col, xlim, type, lty, lwd, segments=NULL, qlty, qcol, polyfill=NULL) {  
 if (type == "plot") {  
 plot(xseq, xdens, type="l", lwd=0, col=col, frame=FALSE, xlim=xlim, lty=lty, ylab='', xlab='')  
 }  
   
 if (!is.null(polyfill)) {  
 polygon(polyfill[,1], polyfill[,2], col=qcol, border=qcol)  
 }  
   
 # draw quantile lines  
 if (!is.null(segments)) {  
 segments(x0=segments[,1], x1=segments[,1], y0=0, y1=segments[,2], lwd=lwd, col=qcol, lty=qlty)  
 }  
   
 lines(xseq, xdens, type="l", lwd=lwd, col=col, lty=lty)  
}  
  
# Plot the t distribution  
plott <- function(lwd=2, ncp=0, df=300, col=rgb(0.30,0.50,0.75), xlim=c(-3,3), type="plot", lty="solid", quants=NULL, qlty="solid", qcol=rgb(0.30,0.50,0.75, 0.5), fill\_quants=NULL) {  
 xseq = seq(ncp-6, ncp+6, length=1000)  
 xdens = dt(xseq, ncp=ncp, df=df)  
 if (length(xlim) == 0) {  
 xlim = c(ncp-3.5, ncp+3.5)  
 }  
   
 segments <- NULL  
 polyfill <- NULL  
   
 if (!is.null(quants)) {  
 xquants = qt(quants, ncp=ncp, df=df)  
 dquants = dt(xquants, ncp=ncp, df=df)  
 segments = cbind(xquants, dquants)  
 }  
  
 if(!is.null(fill\_quants)) {  
 polyq = qt(fill\_quants, ncp=ncp, df=df)  
 polyfill.x = seq(polyq[1], polyq[2], by=0.001)  
 polyfill.y = dt(polyfill.x, ncp=ncp, df=df)  
 polyfill.x = c(polyfill.x[1], polyfill.x, tail(polyfill.x,1))  
 polyfill.y = c(0, polyfill.y, 0)  
 polyfill <- cbind(polyfill.x, polyfill.y)  
 }  
   
 plotdist(xseq, xdens, col, xlim, type, lty, lwd, segments, qlty, qcol, polyfill)  
}  
  
t\_null\_plot <- function(df, alpha) {  
 plott(df=df, col=rgb(0.75, 0.1, 0.1), qcol=rgb(1, 0.5, 0.5), xlim=c(-6, 6), fill\_quants=c(1-alpha, 0.999))  
}  
  
t\_alt\_lines <- function(df, ncp=0, alpha) {  
 blue <- rgb(0.1, 0.1, 0.75)  
 lightblue <- rgb(0.4, 0.4, 1, 0.3)  
 quants <- c(0.5)  
 power\_quant <- pt(qt(1-alpha, df=df), df=df, ncp=ncp)  
 plott(df=df, ncp=ncp, type='lines', lty="dashed", col=blue, quants=quants, qcol=lightblue, xlim=c(-6, 6), fill\_quants=c(power\_quant, 0.999))  
}  
  
t\_test\_plot <- function(diff, sd, n, alpha) {  
 df=n-1  
 t = diff/(sd/sqrt(n))  
 t\_null\_plot(df, alpha)  
 t\_alt\_lines(df,t, alpha)  
}

### Classical Method

ci\_99 <- CI(time, 0.99)  
cat("99% CI: ", ci\_99)

## 99% CI: 7.593525 9.450494

t <- (mean(time)-claims)/sderr(time)  
p <- 1- pt(t, length(time))  
cat("t: ", t, "\np: ", p)

## t: 2.560762   
## p: 0.005265317

p\_test(p, 0.005)

## We do not have enough evidence to reject the Null hypothesis!

### Bootstrapping Method

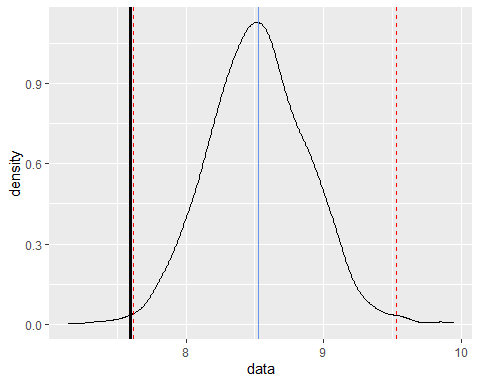
seed = round(runif(1)\*10^9)  
seed

## [1] 500590468

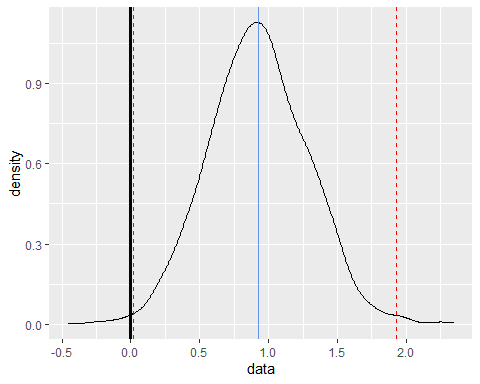
set.seed(seed)  
  
boot\_sample <- function(data, claims){  
 resample <- sample(data, length(data), replace=TRUE)  
 boot\_mean <- mean(resample)  
 boot\_mean\_diff <- mean(resample)-claims  
 boot\_t <- (mean(resample)-claims)/sderr(resample)  
   
 return(c(boot\_mean, boot\_mean\_diff, boot\_t))  
}  
  
boot\_statistics <- replicate(2000, boot\_sample(time, claims))  
  
boot\_mean = boot\_statistics[1, ]  
CI(boot\_mean, 0.99)

## [1] 8.502511 8.544310

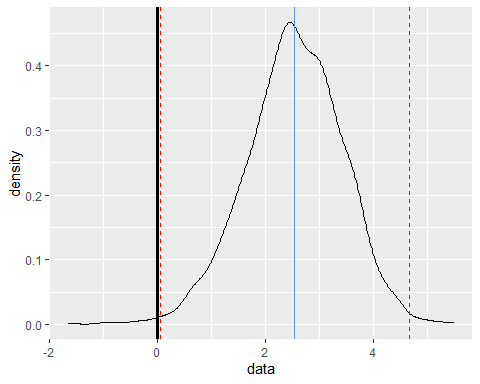
plt(boot\_mean, claims, 0.99)



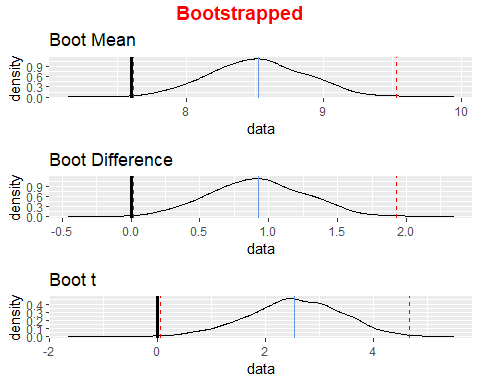
boot\_mean\_diff = boot\_statistics[2, ]  
plt(boot\_mean\_diff, 0)



boot\_t = boot\_statistics[3, ]  
plt(boot\_t, 0)

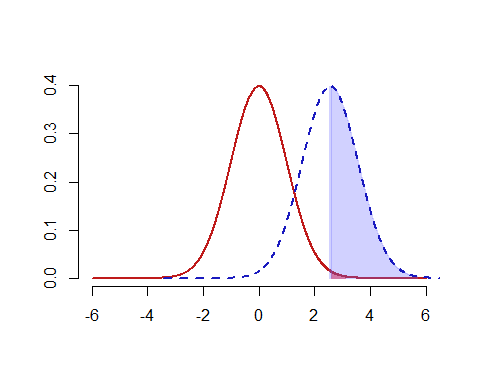


p1 <- plt(boot\_mean, claims, 0.99)+ggtitle('Boot Mean')  
p2 <- plt(boot\_mean\_diff, 0)+ggtitle('Boot Difference')  
p3 <- plt(boot\_t, 0)+ggtitle('Boot t')  
  
graphs <- ggarrange(p1, p2, p3, ncol=1,   
 font.label = c(size=10),  
 common.legend = TRUE,   
 legend="bottom")  
annotate\_figure(graphs, top = text\_grob("Bootstrapped", color="red", face="bold", size=14))



t\_test\_plot(diff=mean(time)-claims, sd=sd(time), n=length(time), alpha=0.005)

## Warning in qt(fill\_quants, ncp = ncp, df = df): full precision may not have been  
## achieved in 'pnt{final}'



**NOTE.** As we can observe from the plot above, it visualizes the classical method we conducted, and that the p-value is slightly larger than the significance level.

#### **a.)** Recreate the traditional hypothesis test

1. Use the t.test() function to conduct a one-sample, one-tailed t-test:  
   report % confidence interval of the mean, t-value, and p-value.

* t.test(time, mu=claims, alternative = "greater", conf.level = 0.99)
* ##   
  ## One Sample t-test  
  ##   
  ## data: time  
  ## t = 2.5608, df = 1686, p-value = 0.005265  
  ## alternative hypothesis: true mean is greater than 7.6  
  ## 99 percent confidence interval:  
  ## 7.683604 Inf  
  ## sample estimates:  
  ## mean of x   
  ## 8.522009

1. Use the power.t.test() function to tell us the power of the test.

* power.t.test(n=length(time),   
   delta = mean(time)-claims,  
   sd = sd(time),  
   alternative = "one.sided",  
   sig.level = 0.001)
* ##   
  ## Two-sample t test power calculation   
  ##   
  ## n = 1687  
  ## delta = 0.9220095  
  ## sd = 14.78848  
  ## sig.level = 0.001  
  ## power = 0.1001352  
  ## alternative = one.sided  
  ##   
  ## NOTE: n is number in \*each\* group

#### **b.)** Bootstrapped hypothesis testing to re-examine this problem.

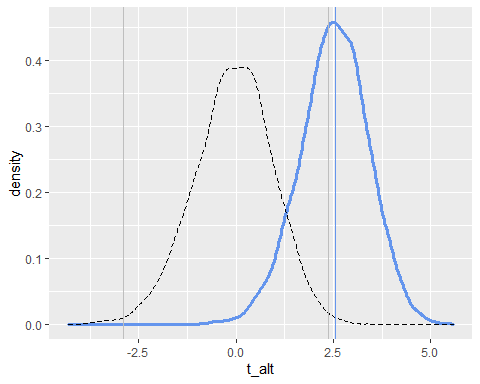
1. Retrieve the original t-value from traditional methods

* t <- (mean(time)-claims)/sderr(time)  
  t
* ## [1] 2.560762

1. Bootstrap the null and alternative t-distributions.

* boot\_null\_alt <- function(data, claims){  
   resample <- sample(data, length(data), replace=TRUE)  
   null <- (mean(resample)-mean(data))/sderr(resample)  
   alt <- (mean(resample)-claims)/sderr(resample)  
   return(c(null, alt))  
  }  
    
  boot\_t\_stat <- replicate(10000, boot\_null\_alt(time, claims))  
    
  t\_null <- boot\_t\_stat[1,]  
  t\_alt <- boot\_t\_stat[2,]

1. Find the 99% cutoff value for critical null values of t, what should our test conclude when comparing the original t-value to the 99% cutoff value?

* cutoff <- quantile(t\_null, c(0.005, 0.995))  
  t > cutoff[1] && t < cutoff[2]
* ## [1] FALSE
* t\_test(t, cutoff)
* ## We do not have enough evidence to reject the Null hypothesis!
* ggplot()+  
   geom\_density(aes(t\_alt), color="cornflowerblue", lwd=1.2)+  
   geom\_density(aes(t\_null), lty="dashed")+  
   geom\_vline(xintercept=t, color="cornflowerblue")+  
   geom\_vline(xintercept=cutoff, col="gray")
* 

1. Compute the p-value and power of our bootstrapped test.

* **p**
* null\_prob <- ecdf(t\_null)  
  p <- 1-null\_prob(t)  
  p\_test(p, 0.001)
* ## We do not have enough evidence to reject the Null hypothesis!
* **power**
* alt\_prob <- ecdf(t\_alt)  
  alt\_power <- 1-alt\_prob(cutoff[2])  
  alt\_power>0.5
* ## [1] TRUE