HW5

108048110

3/16/2022

Question1

Recall the form of t-tests, where

$t = (\bar{x} - \mu_o) * s/\sqrt{n}$

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

$$H0 : \bar{x} \ <= \mu_0$$
$$H1 : \bar{x} > \mu_0$$

After collecting data from just $n = 50$ customers, he informs you that he has found $diff = 0.3$ and $sd = 2.9$. Your colleague believes that we cannot reject the null hypothesis at $alpha = 0.05$.

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 $diff$ and sd will increase.

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

Ans.

By left shifting the alternative hypothesis, it will decrease 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.

n, 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' *alpha* will be affected.

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

Your power to reject the null hypothesis increase because of the increase value of the critical value (0.01 ~ 0.1).

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

Type I 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 both systematic and random errors.

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

It would affect the $diff$, sd and n of the t-statistics. Since more data is about to be included in your sample.

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

Therefore, it increases your 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 will be more likely to occur.

Question2

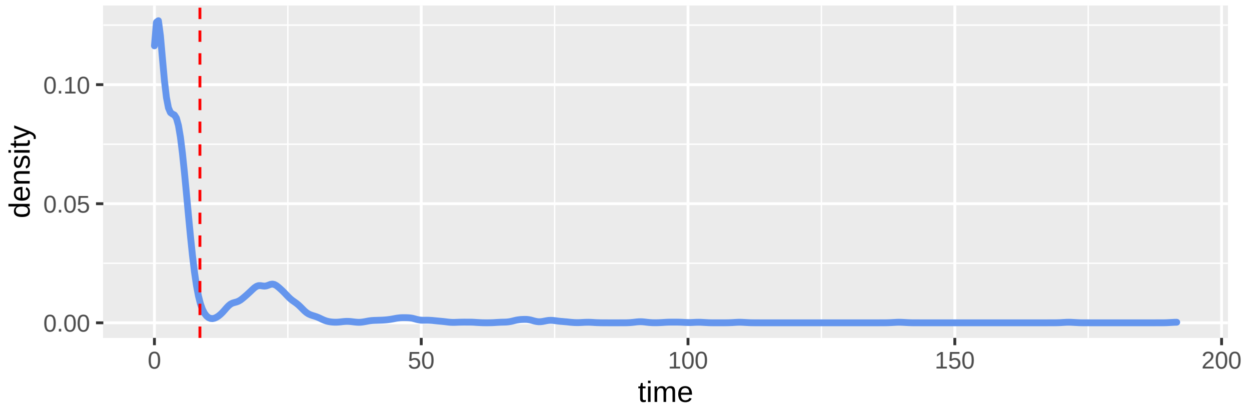
Verizon claims that they take no more than `7.6 minutes on average` to repair phone services for its customers. A recent sample of repair times was seeked to verify this claim at 99% confidence.

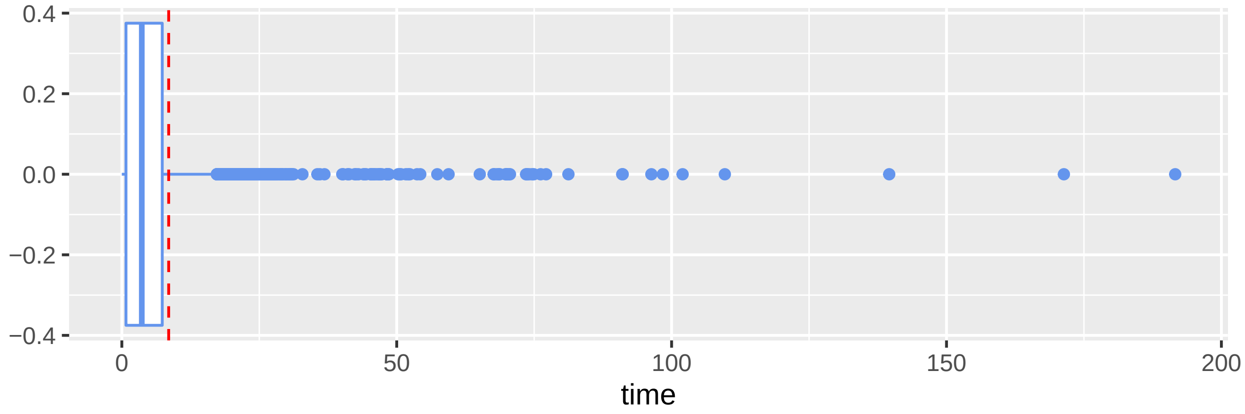
$H0 : \mu_0 <= 7.6$

$H1 : \mu_0 > 7.6$

```
library(ggplot2)
library(ggpubr)
```

```
# 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 do not have enough evidence to reject the Null hypothesis!")}
  else
    {cat("We 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)), col="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))
```

```r
  {
    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

```r
ci_99 <- CI(time, 0.99)
cat("99% CI: ", ci_99)
```

```
## 99% CI:  7.593525 9.450494
```

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

```
## t:  2.560762
## p:  0.005265317
```

```r
p_test(p, 0.005)
```

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

Bootstrapping Method

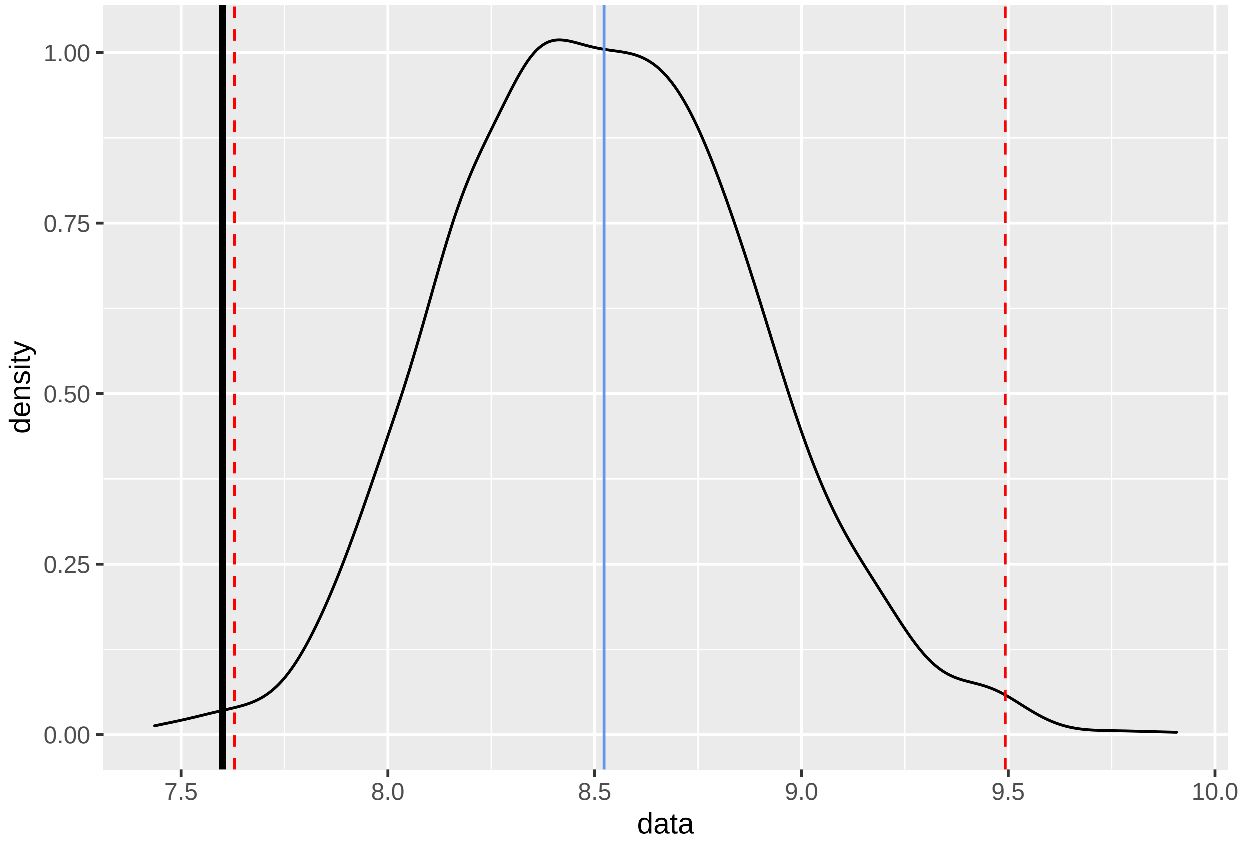
```r
seed = round(runif(1)*10^9)
seed
```

```
## [1] 285431985
```

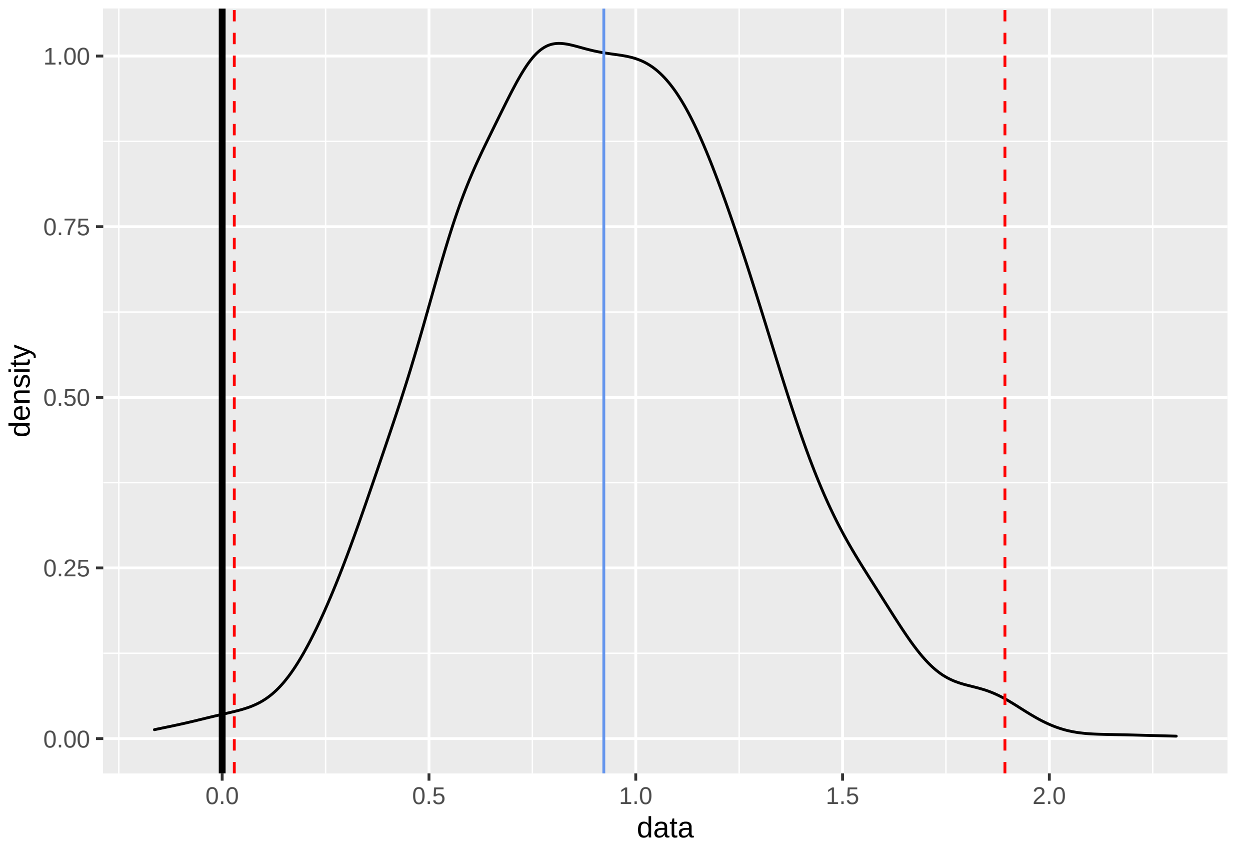
```
set.seed(seed)

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] 8.501766 8.543679
```

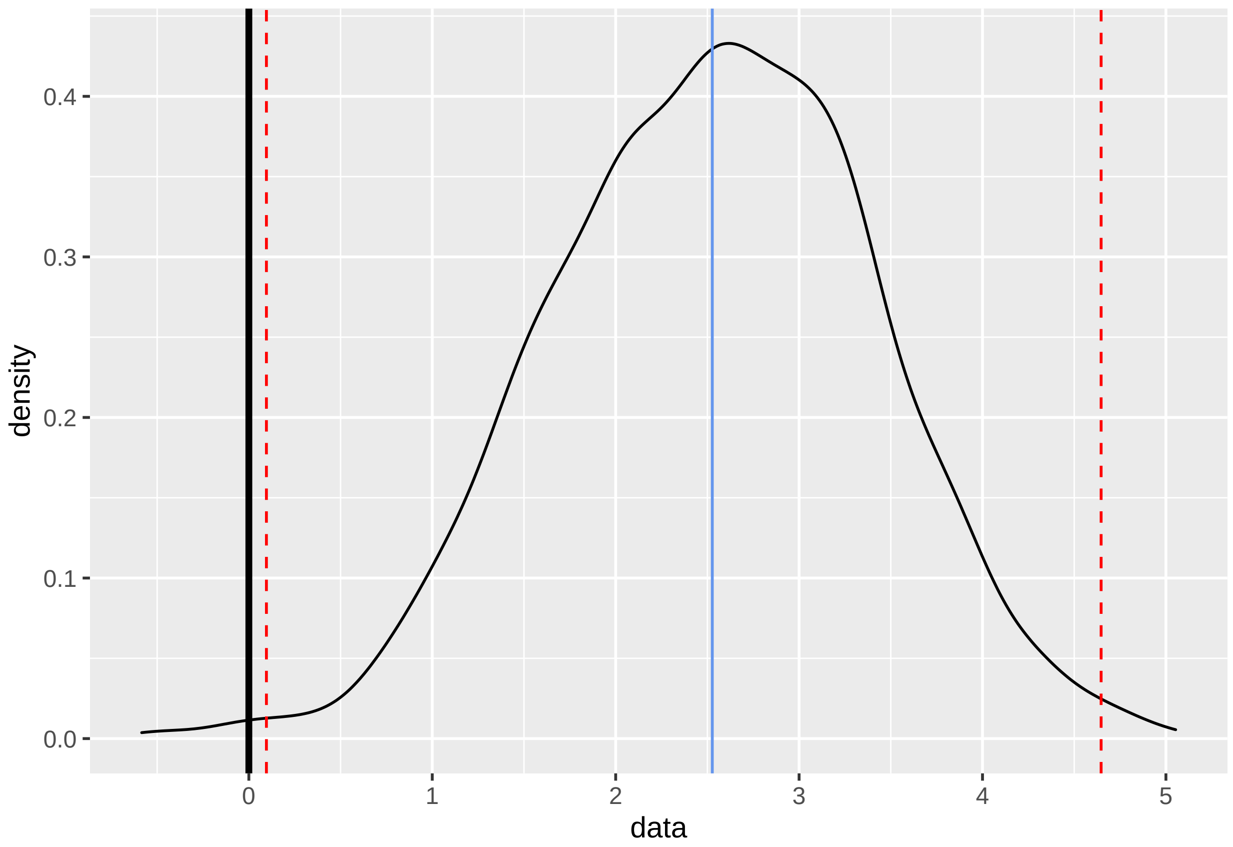
```
plt(boot_mean, claims, 0.99)
```



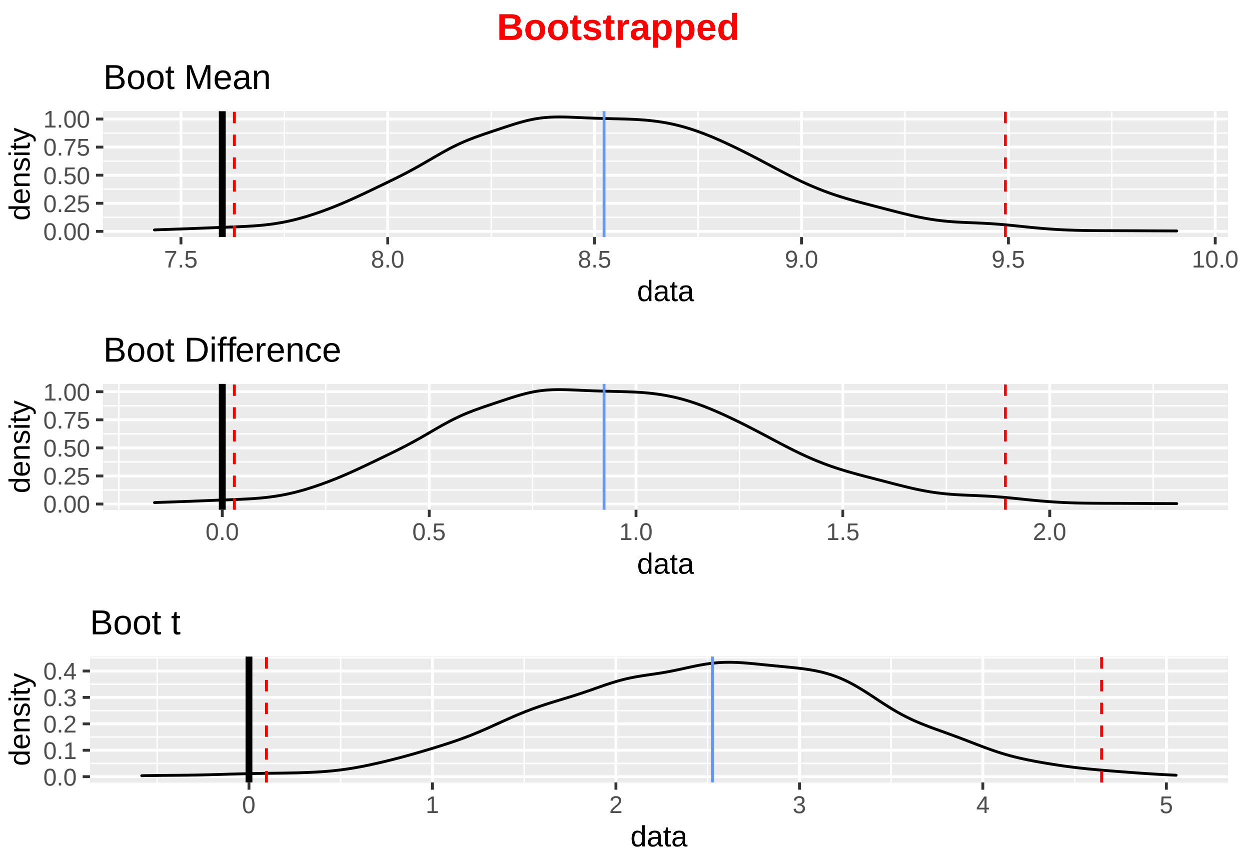
```
boot_mean_diff = boot_statistics[2, ]
plt(boot_mean_diff, 0)
```



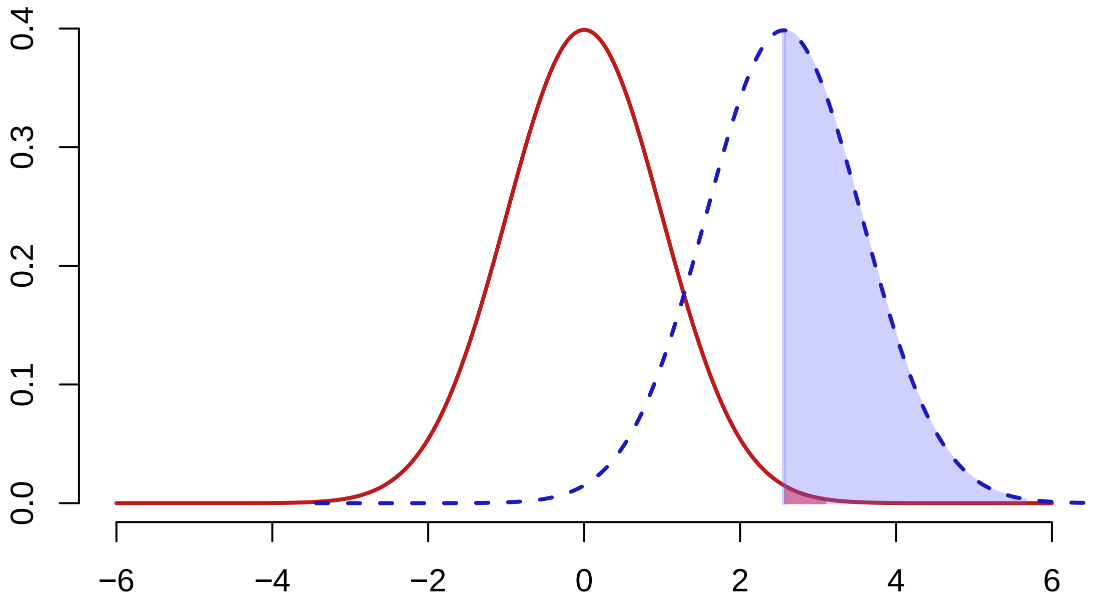
```
boot_t = boot_statistics[3, ]
plt(boot_t, 0)
```



```
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", col="red", face="bold", size=14))
```



```
t_test_plot(diff=mean(time)-claims, sd=sd(time), n=length(time), alpha=0.005)
```



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 99% 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
```

2. 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
```

2. 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,]
```

3. 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?

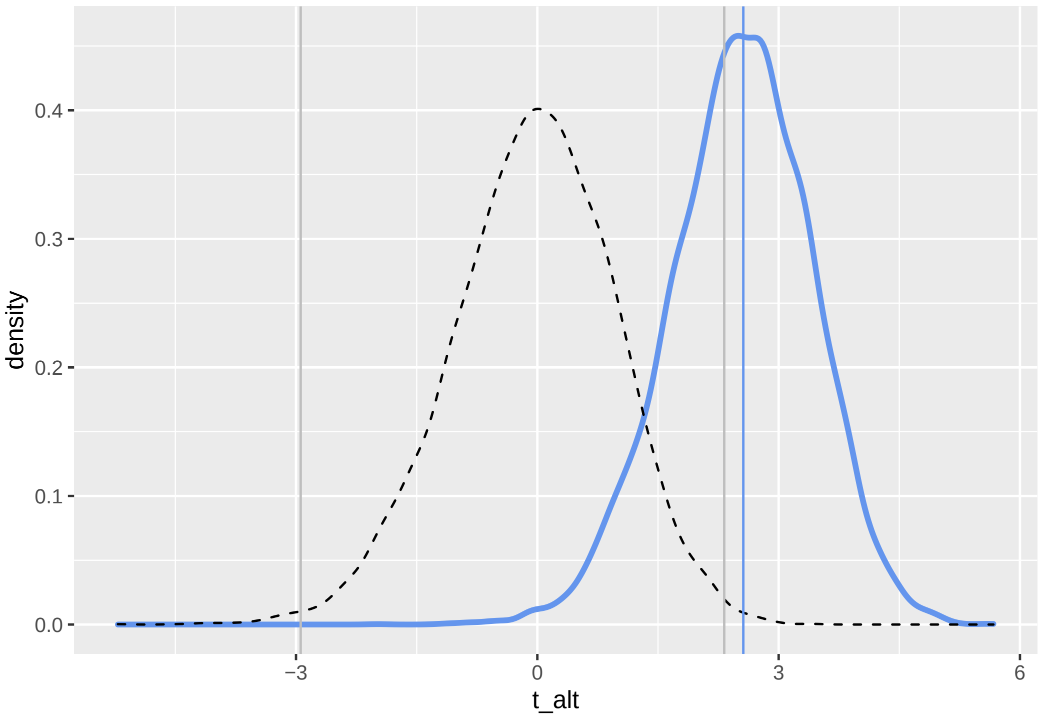
```
cutoff <- quantile(t_null, c(0.005, 0.995))
t > cutoff[1] && t < cutoff[2]
```

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

```
t_test(t, cutoff)
```

```
## We 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")
```



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

   p

```
null_prob <- ecdf(t_null)
p <- 1-null_prob(t)
p_test(p, 0.01)
```

```
## We 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
```