bacs_hw4

110071010

2024-03-14

110070011 has helped me find the slider bars of interactive_t_test(). Originally I didn't run the code in the console, so they didn't show up.

109048231 has helped me notice that I forgot to divide diff by se when computing t-statistic.

Question 1

The large American phone company Verizon had a monopoly on phone services in many areas of the US. The New York Public Utilities Commission (PUC) regularly monitors repair times with customers in New York to verify the quality of Verizon's services. The file verizon.csv has a recent sample of repair times collected by the PUC.

1a

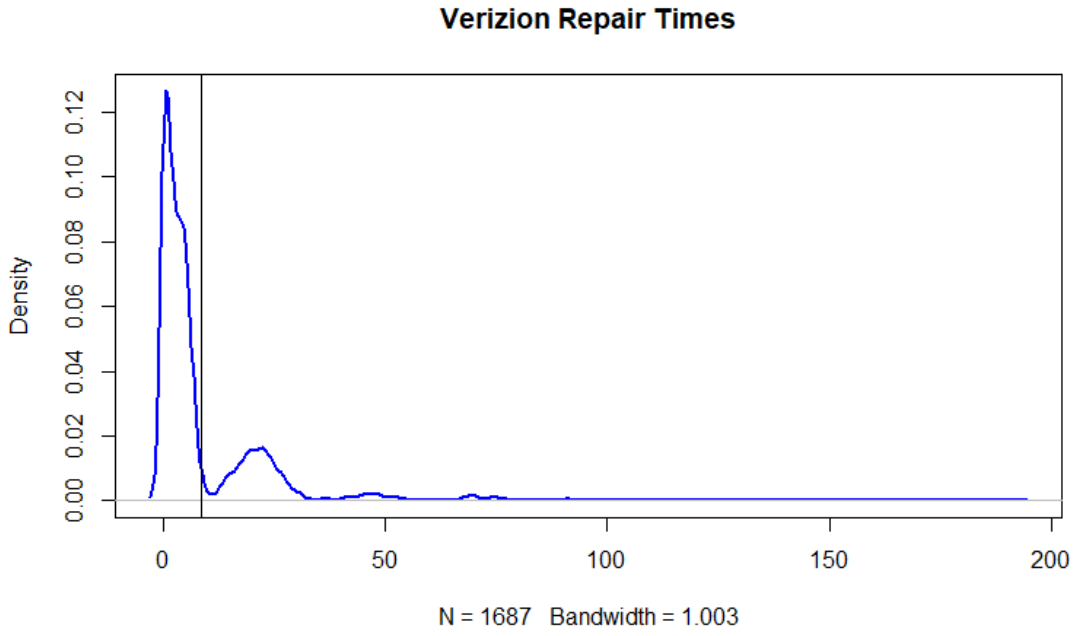
Instruction

Imagine that Verizon claims that they take 7.6 minutes to repair phone services for its customers on average. The PUC seeks to verify this claim at 99% confidence (i.e., significance alpha = 1%) using traditional statistical methods.

My Solution

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

```
# Read in the dataset
data <- read.csv("verizon.csv")
sample_vrt <- data$Time

# Get a glimpse into the distribution
plot(density(sample_vrt),lwd =2, col = "blue", main = "Verizion Repair
Times")
abline(v = mean(sample_vrt))
```



(ii) Given what the PUC wishes to test, how would you write the hypothesis?

Null Hypothesis H0 : Population mean is equal to 7.6 mins
Alternative Hypothesis H1 : Population mean is not equal to 7.6 mins

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

```
vrt_size <- length(sample_vrt)
vrt_mean <- mean(sample_vrt)
vrt_sd <- sd(sample_vrt)
vrt_se <- vrt_sd/sqrt(vrt_size)
vrt_ci99 <- vrt_mean + c(-2.58*vrt_se, 2.58*vrt_se) # 99% CI

# Getting sample mean and 99% CI
cat("Sample Mean:", vrt_mean, "\n")
#cat("Standard Error:", vrt_se, "\n")
cat("99% Confidence Interval:",
    "\n","Lower Bound =", vrt_ci99[1], "\n",
    "Upper Bound =", vrt_ci99[2], "\n")

## Sample Mean: 8.522009
## 99% Confidence Interval:
##  Lower Bound = 7.593073
##  Upper Bound = 9.450946
```

(iv) Find the t-statistic and p-value of the test

```
# Calculating t-stat and p-value
hyp_mean <- 7.6
t <- (vrt_mean - hyp_mean)/vrt_se
df <- vrt_size -1
```

```r
p <- 1- pt(t,df)

cat("t-statistic:", t, "\n")
cat("p-value:", p, "\n")

## t-statistic: 2.560762
## p-value: 0.005265342
```

(v) Briefly describe how these values relate to the Null distribution of t

: The null distribution of the t-statistic is a theoretical distribution that assumes the null hypothesis is true. It is centered around 0 (indicating no difference between the sample mean and the null hypothesis mean) and has a shape determined by the degrees of freedom.

(vi) What is your conclusion about the company's claim from this t-statistic, and why?

: Given that p-value is greater than 0.005 (alpha/2) , we can accept the company's claim (H0) that their average repair times equals 7.6.

1b

Instruction

Let's re-examine Verizon's claim that they take no more than 7.6 minutes on average, but this time using bootstrapped testing:

My Solution

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

```r
# Setting up function
sample_statistic <- function(stat_function,sample0){
  resample <- sample(sample0,length(sample0),replace = TRUE)
  stat_function(resample)
}

# Setting seed before bootstrapping
set.seed(123123)

# Do bootstrapping 2000 times
num_boot <- 2000
boot_sample_means <- replicate(num_boot,sample_statistic(mean,sample_vr
t))

# Print estimated 99% CI of sampling means
ci99_boot_means <- quantile(boot_sample_means, probs = c(0.005, 0.995))
cat("99% CI of sampling means","\n")
ci99_boot_means
```

```
## 99% CI of sampling means
##     0.5%    99.5%
## 7.638925 9.441972
```

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

```
# Setting up function
bootstrapping_mean_diff <- function(sample0, hypothesized_mean){
  resample <- sample(sample0,length(sample0),replace = TRUE)
  return(mean(resample) - hypothesized_mean)
}

# Setting seed before bootstrapping
set.seed(123123)

# Do bootstrapping 2000 times
num_boot <- 2000
boot_mean_diffs <- replicate(num_boot,bootstrapping_mean_diff(sample_vr
t,hyp_mean))

# Print estimated 99% CI of mean differences
ci99_boot_mean_diffs <- quantile(boot_mean_diffs, probs = c(0.005, 0.99
5))
cat("99% CI of mean differences","\n")
ci99_boot_mean_diffs
```

```
## 99% CI of mean differences
##       0.5%      99.5%
## 0.03892528 1.84197161
```

(iii) Bootstrapped t-statistic: What is the 99% CI of the bootstrapped t-statistic of the sample mean versus the hypothesized mean?

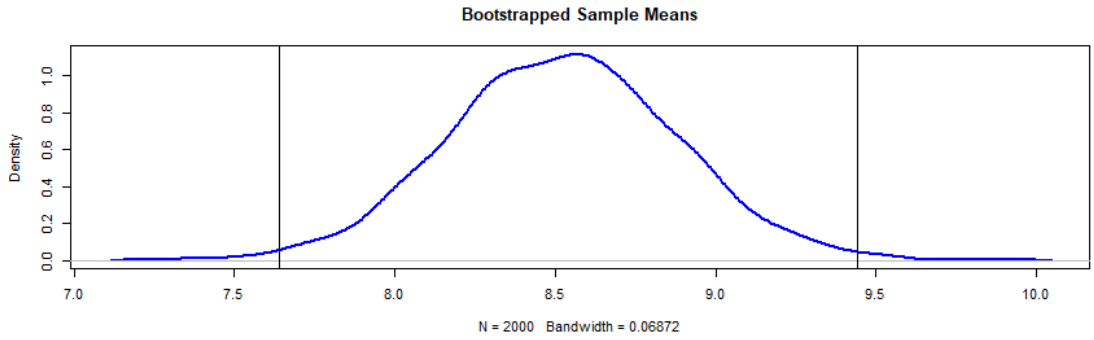
```
# Build the function for calculating t-stat
bootstrapping_t_stat <- function(sample0, hypothesized_mean){
  resample <- sample(sample0,length(sample0),replace = TRUE)
  diff <- mean(resample) - hypothesized_mean
  se <- sd(resample)/sqrt(length(resample))
  return(diff/se)
}

# Setting seed before bootstrapping
set.seed(123123)

# Do bootstrapping 2000 times
num_boot <- 2000
boot_t_stats <- replicate(num_boot,bootstrapping_t_stat(sample_vrt,hyp_
mean))

# Print estimated 99% CI of t_stats
```

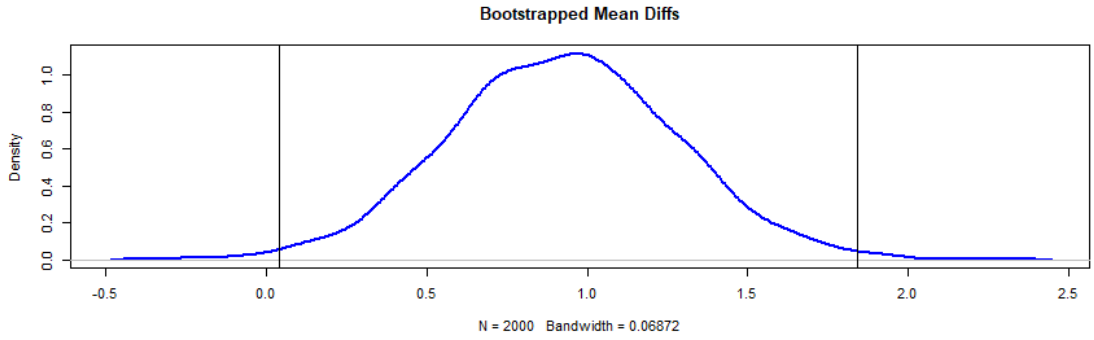
```r
ci99_boot_t_stats <- quantile(boot_t_stats, probs = c(0.005, 0.995))
cat("99% CI of tstats","\n")
ci99_boot_t_stats
```

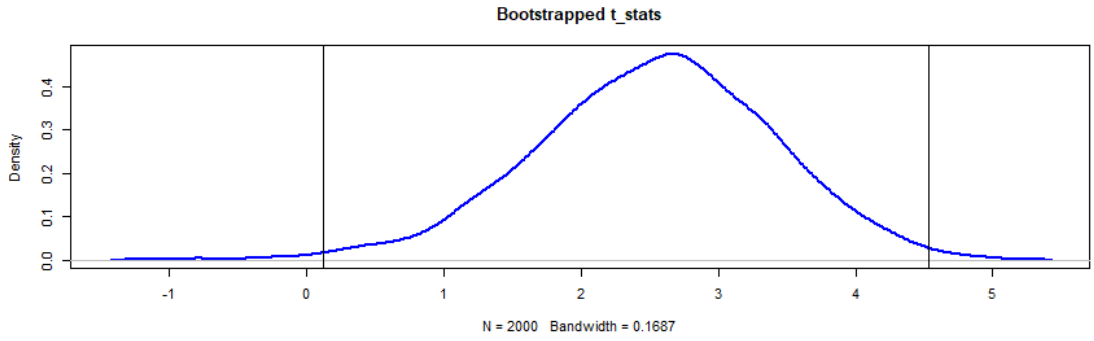
```
## 99% CI of tstats
##      0.5%     99.5%
## 0.1232184 4.5356472
```

(iv) Plot the distribution of the three bootstraps above on separate plots; draw vertical lines showing the lower/upper bounds of their respective 99% confidence intervals.

```r
# Display three plots at once
par(mfrow = c(3,1))

# bootstrapped sampling means
plot(density(boot_sample_means),lwd = 2, main = "Bootstrapped Sample Me
ans",col = "blue")
abline(v = quantile(boot_sample_means, probs = c(0.005, 0.995), names =
 FALSE))
# bootstrapped mean differences
plot(density(boot_mean_diffs),lwd = 2, main = "Bootstrapped Mean Diffs",
col = "blue")
abline(v = quantile(boot_mean_diffs, probs = c(0.005, 0.995), names = F
ALSE))
# bootstrapped t-statistics
plot(density(boot_t_stats),lwd = 2, main = "Bootstrapped t_stats",col =
 "blue")
abline(v = quantile(boot_t_stats, probs = c(0.005, 0.995), names = FALS
E))
```







(v) Does the bootstrapped approach agree with the traditional t-test in part [a]?

: No, it is evident that the claimed average repair times of 7.6 mins (or 0) lies outside the 99 % CIs of the bootstrapped statistics above.

1c

Instruction

Finally, imagine that Verizon notes that the distribution of repair times is highly skewed by outliers, and feel that testing the mean is not fair because the mean is sensitive to outliers. They argue that the median is a more fair test, and claim that the median repair time is no more than 3.5 minutes at 99% confidence (i.e., significance alpha = 1%).

My Solution

One-tailed Hypothesis Testing

Null Hypothesis H0 : population median <= 3.5
Alternative Hypothesis : population median > 3.5

(i) Bootstrapped Percentile: Estimate the bootstrapped 99% CI of the population median

```
# Get the function ready
sample_statistic <- function(stat_function,sample0){
  resample <- sample(sample0,length(sample0),replace = TRUE)
  stat_function(resample)
}

# Setting seed before bootstrapping
set.seed(123123)

# Do bootstrapping 2000 times
boot_sample_medians <- replicate(num_boot,sample_statistic(median,sampl
e_vrt))

# Print estimated 99% CI of sampling medians
ci99_boot_medians <- quantile(boot_sample_medians, probs = 0.99)
cat("99% CI of sampling medians","\n")
ci99_boot_medians
```

```
## 99% CI of sampling medians
##    99%
## 3.9002
```

(ii) Bootstrapped Difference of Medians: What is the 99% CI of the bootstrapped difference between the sample median and the hypothesized median?

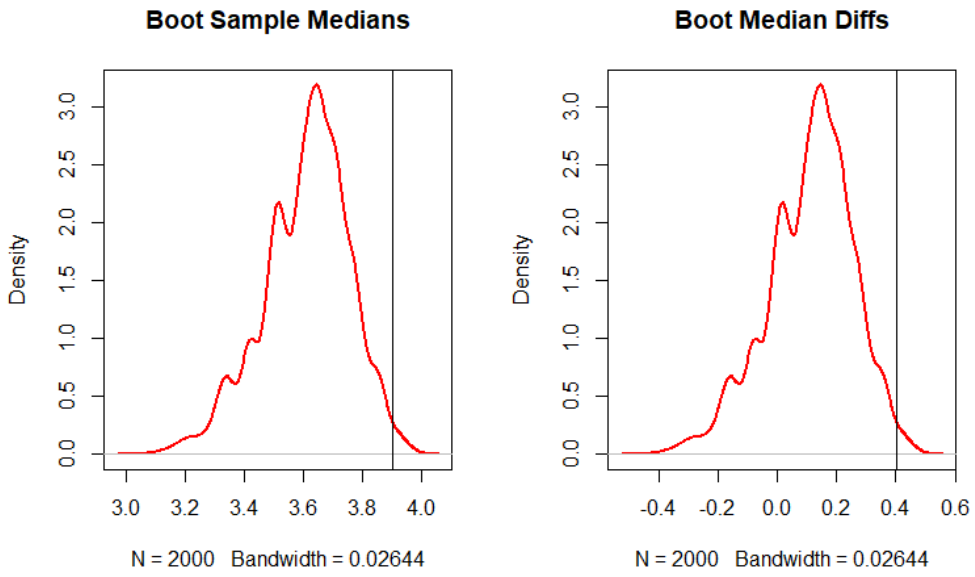
```
# Setting up function
bootstrapping_median_diff <- function(sample0, hypothesized_median){
  resample <- sample(sample0,length(sample0),replace = TRUE)
  return(median(resample) - hypothesized_median)
}

# Setting seed before bootstrapping
set.seed(123123)

# Do bootstrapping 2000 times
hyp_median <- 3.5
num_boot <- 2000
boot_median_diffs <- replicate(num_boot,bootstrapping_median_diff(sampl
e_vrt,hyp_median))

# Print estimated 99% CI of sampling median differences
ci99_boot_median_diffs <- quantile(boot_median_diffs, probs = 0.99)
```

```r
cat("99% CI of median differences","\n")
ci99_boot_median_diffs
```

```
## 99% CI of median differences
##    99%
## 0.4002
```

(iii) Plot distribution the two bootstraps above on two separate plots

```r
# Display two plots at once
par(mfrow = c(1,2))
# bootstrapped sampling medians
plot(density(boot_sample_medians),lwd = 2, main = "Boot Sample Medians",
col = "red")
abline(v = quantile(boot_sample_medians, probs = 0.99, names = FALSE))
# bootstrapped median differences
plot(density(boot_median_diffs),lwd = 2, main = "Boot Median Diffs",col
 = "red")
abline(v = quantile(boot_median_diffs, probs = 0.99, names = FALSE))
```



(iv) What is your conclusion about Verizon's claim about the median, and why?

We accept Null Hypothesis H0, since 3.5 clearly lies within the estimated 99 % CI.

Question 2

```r
# install.packages("remotes")
# remotes::install_github("soumyaray/compstatslib")
# library(compstatslib)
# compstatslib::interactive_t_test()
```

Instruction

Your colleague, a data analyst in your organization, is working on a hypothesis test where he has sampled product usage information from customers who are using a new smartwatch. He wishes to test whether the mean usage time is higher than the usage time of the company's previous smartwatch released two years ago:

H_null : The mean usage time of the new smartwatch is the same or less than for the previous smartwatch

H_alt : The mean usage time is greater than that of our previous smartwatch

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 of 5%.

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:

1. Would this scenario create systematic or random error (or both or neither)?
2. Which part of the t-statistic or significance (diff, sd, n, alpha) would be affected?
3. Will it increase or decrease our power to reject the null hypothesis?
4. Which kind of error (Type I or Type II) becomes more likely because of this scenario?

2a

Scenario

You discover that your colleague wanted to target the general population of Taiwanese users of the product. However, he only collected data from a pool of young consumers, and missed many older customers who you suspect might use the product much less every day.

Answer

1. systemic error
2. It appears that diff and sd would be affected.
3. increase
4. neither becomes more likely

2b

Scenario

You find that 20 of the respondents are reporting data from the wrong wearable device, and should not have been in the sample. These 20 people are just like the others in every other respect.

Answer

1. random error
2. The unwanted responses in the sample is likely to inflate the sample size n.
3. decrease
4. type II error rate (beta)

2c

Scenario

A very annoying professor visiting your company has criticized your colleague's "95% confidence" criteria, and has suggested relaxing it to just 90%.

Answer

1. neither
2. The significance level (alpha) is relaxed from 0.05 to 0.10.
3. increase
4. type I error rate (alpha)

2d

Scenario

Your colleague has measured usage times on five weekdays and taken a daily average. But you feel this will underreport usage for younger people who are very active on weekends, whereas it over-reports usage of older users.

Answer

1. systemic error
2. It appears that diff and sd would be affected.
3. increase
4. neither becomes more likely.