HW 5

110077443

3/14/2022

Please note that all code in this document is presented in a grey box and the output reflected below each box

- The below code allows lengthy lines of code to display neatly within the grey box (wrapping it)

```
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

1) Confirming from the visualization that we cannot reject the null hypothesis.

- With n = 50, diff = 0.3, sd = 2.9, alph a= 5%, We **cannot reject** the hypothesis.

a) Data from a pool of young consumers:

- i) **Neither** because it's a coverage error in which the sample does not properly represent the population measured and the error is not caused by measurement.
- ii) **diff** and **sd** would be affected
- iii) Power would **decrease**
- iv) Type **II** error

b) 20 of the respondents are reporting data from the wrong wearable device:

- i) **Random** error
- ii) **n** would be affected
- iii) Power would **decrease**
- iv) Type **II** error

c) 90% CI:

- i) **Neither**
- ii) **alpha** would be affected
- iii) Power would **increase**
- iv) Type **I** error

d) Usage times on five weekday only:

- i) **Systemic** error
- ii) **sd** would be affected
- iii) Power would **decrease**
- iv) Type **II** error

2) Verify the claim that Verizon takes no more than 7.6 minutes on average (single-tail test):

- Hypothesized mean claim:

```r
# H0: mu <= 7.6
verizon_claim <- 7.6
```

- Import the data for our sample:

```r
verizon <- read.csv("verizon.csv", header = TRUE)
str(verizon)  # Checking structure for possible formatting
```

```
## ’data.frame':    1687 obs. of  2 variables:
##  $ Time : num  17.5 2.4 0 0.65 22.23 ...
##  $ Group: chr  "ILEC" "ILEC" "ILEC" "ILEC" ...
```

```r
table(verizon$Group)  # Checking how many observations in each 'Group'
```

```
##
## CLEC ILEC
##   23 1664
```

```r
verizon_sample <- verizon$Time  # Removing 'Group' variable since we only need time
sample_size <- length(verizon_sample)  # 1687
sample_mean <- mean(verizon_sample)  # 8.522009
sample_sd <- sd(verizon_sample)  # 14.78848
```

a) Traditional hypotheis test

i) Using the t.test() function to conduct a one-sample, one-tailed t-test

```r
t_test_greater <- t.test(verizon_sample, conf.level = 0.99, alternative = "greater",
    mu = 7.6)
t_test_greater  # Print all
```

```
##
##  One Sample t-test
##
## data:  verizon_sample
## 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
```

```
t_test_greater$conf.int  # 7.683604 - Infinity (99% CI of mean one-sided 'greater')
```

```
## [1] 7.683604      Inf
## attr(,"conf.level")
## [1] 0.99
```

```
t_test_greater$statistic  # 2.560762 (t-value)
```

```
##        t
## 2.560762
```

```
t_test_greater$p.value  # 0.005265342 (p-value)
```

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

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

```
power_test <- power.t.test(n = length(verizon_sample), delta = sample_mean -
    verizon_claim, sd = sample_sd, alternative = "one.sided")
power_test  # Print
```

```
##
##      Two-sample t test power calculation
##
##               n = 1687
##           delta = 0.9220095
##              sd = 14.78848
##       sig.level = 0.05
##           power = 0.5657309
##     alternative = one.sided
##
## NOTE: n is number in *each* group
```

b) Let's use bootstrapped hypothesis testing to re-examine this problem:

i) Traditional statistics: 99% CI of mean, t-value, and p-value.

```
# Population mean
sample_mean <- mean(verizon_sample)  # 8.522009
sample_mean  # Print
```

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

```
# Compute Standard Error
sample_se <- sample_sd/(sqrt(sample_size))  # 0.3600527
sample_se  # Print
```

```
## [1] 0.3600527
```

```
# Compute 99% confidence interval for this estimate
verizon_ci99 <- sample_mean + c(-2.576) * sample_se  # 99% CI
verizon_ci99  # Print
```

```
## [1] 7.594514
```

- The estimated population mean is **8.522009**, and we are 99% confident that this estimate is between **7.594514 and Infinity** since it is a one sided test.

```
# t-statistic
t_stat <- (sample_mean - verizon_claim)/sample_se  # 2.560762
t_stat  # Print
```

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

```
# p-value
df <- sample_size - 1  # Degrees of freedom
p_value <- pt(t_stat, df, lower.tail = FALSE)  # 0.005265342
p_value  # Print
```

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

ii) Bootstrap the null and alternative t-distributions

```
# Adding function for bootstrapping
bootstrap_null_alt <- function(sample0, hyp_mean) {
    resample <- sample(sample0, length(sample0), replace = TRUE)
    resample_se <- sd(resample)/sqrt(length(resample))

    t_stat_alt <- (mean(resample) - hyp_mean)/resample_se  # alt value of t
    t_stat_null <- (mean(resample) - mean(sample0))/resample_se  # null value of t
    return(c(t_stat_alt, t_stat_null))
}

# Bootstrap the t-statistics (Null and Alternative)
boot_t_stats <- replicate(10000, bootstrap_null_alt(verizon_sample,
    verizon_claim))
t_alt <- boot_t_stats[1, ]
t_null <- boot_t_stats[2, ]
```

iii) Find the 95% cutoff value for critical null values of t (from the bootstrapped null)

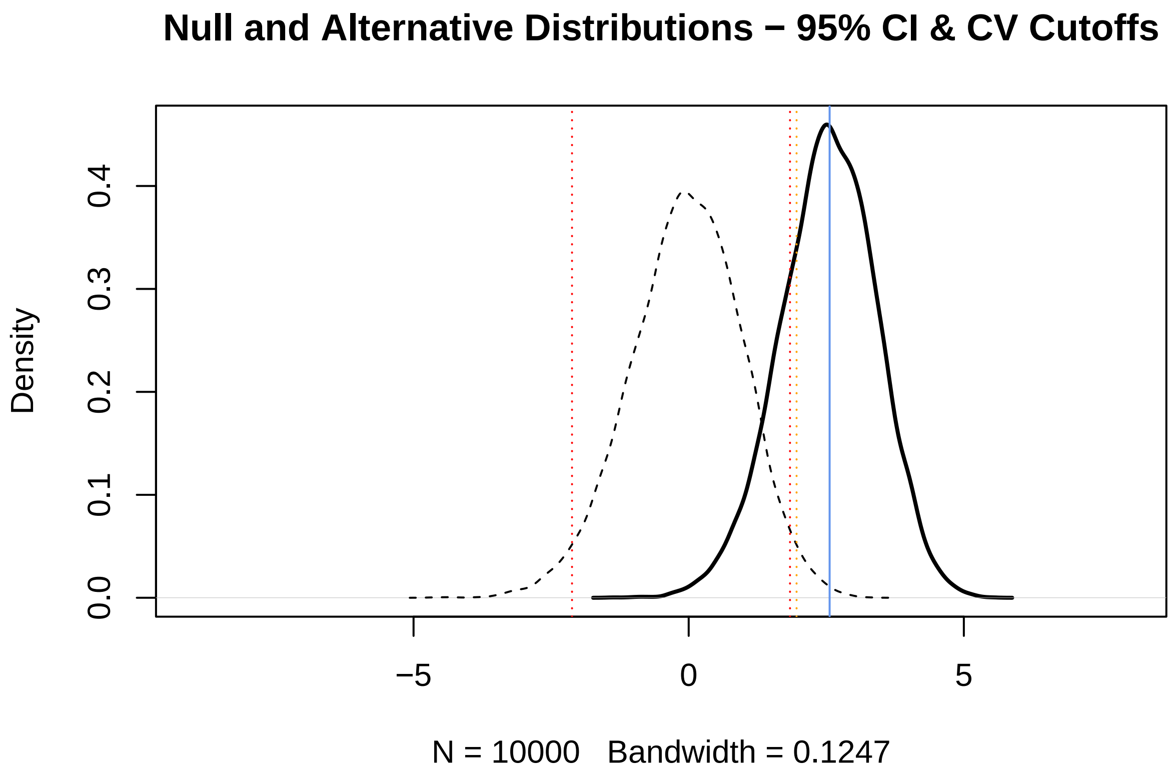
```
# 95% cutoff value for critical null values of t
null_t_cutoff95 <- abs(qt(0.025, df = length(t_null) - 1))  # 1.960201
null_t_cutoff95  # Print
```

```
## [1] 1.960201
```

```
# 95% CI of t-distribution
ci_95 <- quantile(t_null, probs = c(0.025, 0.975))
ci_95  # Print
```

```
##      2.5%     97.5%
## -2.120070  1.840994
```

```
# Visualizing null and alternative t-distributions with 95%
# CI Cutoff
plot(density(t_alt), xlim = c(-9, 8), lwd = 2, main = "Null and Alternative Distributions - 95% CI & CV
lines(density(t_null), lty = "dashed")
abline(v = t_stat, col = "cornflowerblue")
abline(v = null_t_cutoff95, lty = "dotted", col = "orange")
abline(v = ci_95, lty = "dotted", col = "red")
```



- **What should our test conclude when comparing the original t-value to the 99% cutoff value?**

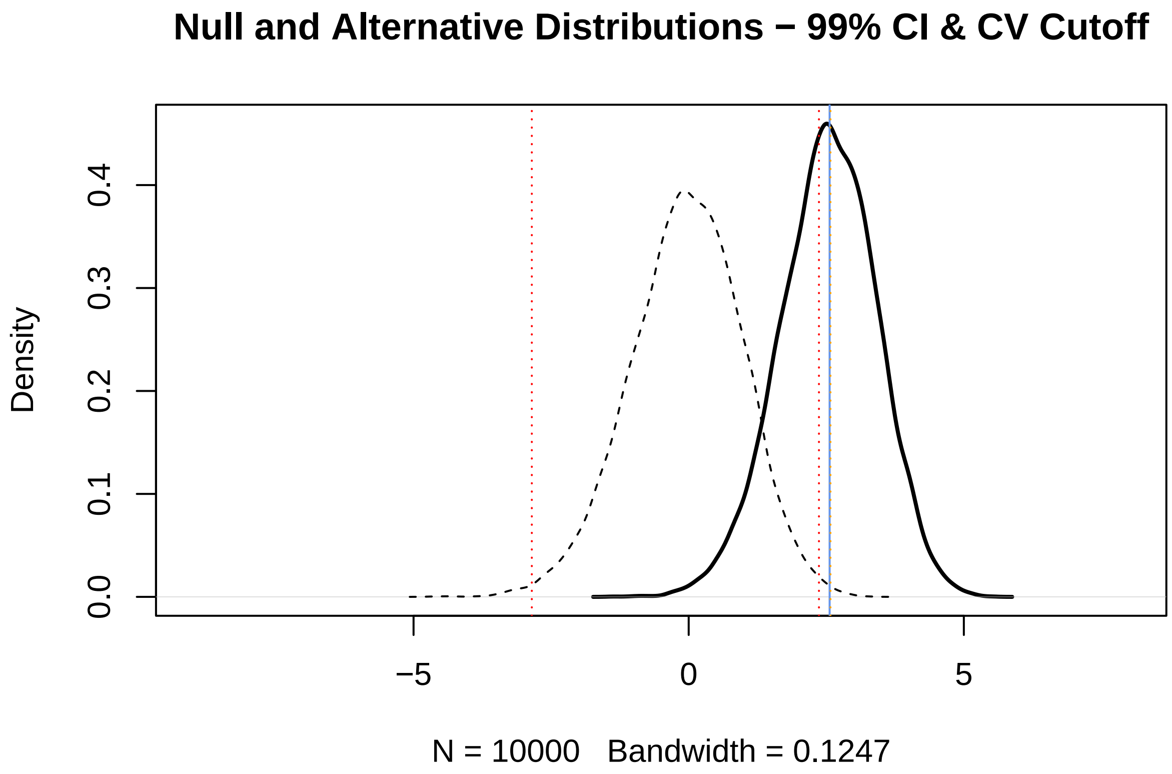
```
# 95% cutoff value for critical null values of t
null_t_cutoff99 <- abs(qt(0.005, df = length(t_null) - 1))  # 2.576321
null_t_cutoff99  # Print
```

```
## [1] 2.576321
```

```
# 99% CI of t-distribution
ci_99 <- quantile(t_null, probs = c(0.005, 0.995))
ci_99  # Print
```

```
##      0.5%     99.5%
## -2.850962  2.367069
```

```
# Visualizing null and alternative t-distributions with 99%
# CI Cutoff
plot(density(t_alt), xlim = c(-9, 8), lwd = 2, main = "Null and Alternative Distributions - 99% CI & CV
lines(density(t_null), lty = "dashed")
abline(v = t_stat, col = "cornflowerblue")
abline(v = null_t_cutoff99, lty = "dotted", col = "orange")
abline(v = ci_99, lty = "dotted", col = "red")
```



- **Conclusion:**
  - Our t-value of *2.560762* lies outside both the 95% and 99% *(between -2.897240 and 2.397879)* CI cutoffs of the bootstrapped null distribution, but within the 99% CV cutoff.

iv) Compute the p-value and power of our bootstrapped test

```
# P-value of bootstrapped test
null_probs <- ecdf(t_null)
one_tailed_pvalue <- 1 - null_probs(t_stat)  # 0.0028 for right-tailed
one_tailed_pvalue  # Print
```

```
## [1] 0.0023
```

```
two_tailed_pvalue <- 2 * one_tailed_pvalue  # 0.0056 for two-tailed
two_tailed_pvalue  # Print
```

```
## [1] 0.0046
```

```
# Power of our bootstrapped test
alt_probs <- ecdf(t_alt)
alt_probs(ci_95[1]) + (1 - alt_probs(ci_95[2]))  # 0.7897 for two-tailed power at 95%
```

```
## [1] 0.7887
```

```
alt_probs <- ecdf(t_alt)
alt_probs(ci_99[1]) + (1 - alt_probs(ci_99[2]))  # 0.5703 for two-tailed power at 99%
```

```
## [1] 0.5879
```