DSCC/CSC/STAT 462 Assignment 3

Due October 20, 2022 by 11:59 p.m.

Please complete this assignment using `RMarkdown`, and submit the knitted PDF. *For all hypothesis tests, state the hypotheses, report the test statistic and p-value, and comment on the results in the context of the problem.*

In order to run hypothesis tests and construct confidence intervals, you may find the `z.test` and/or `t.test` functions in `R` to be useful. For documentation, run `?z.test` and/or `?t.test` in the console.

1. Recently there has been much concern regarding fatal police shootings, particularly in relation to a victim's race (with "victim" being used generally to describe the person who was fatally shot). Since the start of 2015, the Washington Post has been collecting data on every fatal shooting in America by a police office who was on duty. A subset of that data is presented in the dataset "shootings.csv."

```r
shoot <- read.csv("shootings.csv")
head(shoot)
```

```
##                  name armed age minority
## 1          Tim Elliot   gun  53      yes
## 2 Sylasone Ackhavong   gun  41      yes
## 3        Mario Jordan   gun  34      yes
## 4      Douglas Harris   gun  77      yes
## 5      Jeffrey Adkins   gun  53      yes
## 6   Trepierre Hummons   gun  21      yes
```

```r
len <- nrow(shoot)
len
```

```
## [1] 180
```

a. Construct a two-sided 85% confidence interval "by-hand" (i.e. do not use the `t.test()` function, but still use `R`) on the mean age of victims. Interpret the result.

```r
age = shoot$age
mean_age = mean(shoot$age)
sd_age = sd(shoot$age)
mean_age
```

```
## [1] 41.72778
```

```r
sd_age
```

```
## [1] 14.30312
```

```r
y = qt(0.925, df = 179)
y
```

```
## [1] 1.445735
```

```r
ci = y * sd_age/sqrt(len)
ci
```

```
## [1] 1.541286
```

```r
l_ci = mean_age - ci
u_ci = mean_age + ci
print(paste("(", l_ci, ",", u_ci, ")", "is confidence interval"))
```

```
## [1] "( 40.1864919311615 , 43.269063624394 ) is confidence interval"
```

(40.1864919311615 , 43.269063624394) is confidence interval

```r
age
```

```
##   [1] 53 41 34 77 53 21 67 46 29 38 30 29 43 26 38 25 29 43 33 28 33 39 17 24 25
##  [26] 69 31 32 64 31 39 22 33 25 50 24 27 24 56 36 39 24 32 54 61 68 31 42 45 57
##  [51] 47 43 31 53 28 52 20 49 60 57 56 49 27 23 31 72 40 26 27 53 46 58 53 41 18
##  [76] 62 56 58 69 15 48 29 51 17 50 50 53 41 44 19 57 36 27 37 39 33 55 47 28 51
## [101] 47 39 20 21 50 58 60 40 33 39 45 26 58 44 34 51 49 50 40 46 33 51 52 24 30
## [126] 28 50 63 31 59 69 59 59 37 55 56 22 26 43 72 61 41 39 56 30 50 40 58 38 56
## [151] 79 32 24 67 30 48 66 58 59 45 36 25 28 25 32 50 38 57 26 22 28 47 31 35 30
## [176] 29 51 46 26 34
```

```r
# t.test(age, y = NULL, alternative = c('two.sided',
# 'less', 'greater'),mu = 40, paired = FALSE, var.equal =
# FALSE, conf.level = 0.95)
```

b. A recent census study indicates that the average age of Americans is 40 years old. Conduct a hypothesis test "by-hand" (i.e. do not use the ‘t.test()‘ function, but still use ‘R‘) at the $\alpha=0.05$ significance level to see if the average age of victims is significantly different from 40 years old.

H0:
H1:

```r
alpha = 0.05
mean = 40
t = ((mean_age - mean) * (sqrt(len)))/sd_age
t
```

```
## [1] 1.620666
```

```r
p_value_shoot = 2 * (1 - pt(t, len - 1))
p_value_shoot
```

```
## [1] 0.1068496
```

P value is 0.1068496

c. At the $\alpha=0.01$ significance level, test "by-hand" (i.e. do not use the ‘t.test()‘ function, but still use ‘R‘) whether the average age of minority victims is different than the average age of non-minority victims. Assume equal variances.
\vspace{5pt}

H0:
H1: Average age

```r
minority <- shoot[shoot$minority == "yes", ]
mean_minority = mean(minority$age)
sd_minority = sd(minority$age)
mean_minority
```

```
## [1] 36.72917
```

```
sd_minority
```

```
## [1] 13.5407
```

```
smin = sd_minority^2
lenmin = nrow(minority)

nonminority <- shoot[shoot$minority == "no", ]
mean_nonminority = mean(nonminority$age)
sd_nonminority = sd(nonminority$age)
mean_nonminority
```

```
## [1] 43.54545
```

```
sd_nonminority
```

```
## [1] 14.18706
```

```
snonmin = sd_nonminority^2
lennonmin = nrow(nonminority)

numer = (lenmin - 1) * smin + (lennonmin - 1) * snonmin
denom = (lenmin - 1) + (lennonmin - 1)
sp2 = numer/denom
tnum = (mean_minority - mean_nonminority)
tdeno = sqrt(sp2 * (1/lenmin + 1/lennonmin))
teq = tnum/tdeno
teq
```

```
## [1] -2.884651
```

```
df = lennonmin + lenmin - 2
p_value = 2 * (pt(teq, df))
p_value
```

```
## [1] 0.00440256
```

P Value is 0.00440256

2. In the dataset named “blackfriday.csv,” there is information relating to the amount of money that a sample of $n = 31$ consumers spent shopping on Black Friday in 2017.

a. A company is interested in determining an upper-bound on the mean amount of money spent on Black Friday in order to determine maximum effects on the economy. Construct a one-sided upper-bound 99% lower confidence interval "by-hand" (i.e. do not use the `t.test()` function, but still use `R`) for the mean amount of money spent on Black Friday. Interpret the results.

```
bf <- read.csv("blackfriday.csv")
amountbf = bf$Amount
mean_bf = mean(bf$Amount)
sd_bf = sd(bf$Amount)
mean_bf
```

```
## [1] 11087.65
```

```
sd_bf
```

```
## [1] 5959.942
```

```
lenbf <- nrow(bf)
lenbf
```

```
## [1] 31
```

```
y = qt(0.99, lenbf - 1)
y
```

```
## [1] 2.457262
```

```
ci_bf = y * sd_bf/sqrt(lenbf)
ci_bf
```

```
## [1] 2630.344
```

```
up_bf = mean_bf + ci_bf
up_bf
```

```
## [1] 13717.99
```

One-sided upper-bound 99% lower confidence interval is (-infinity,13717.99)

```r
# t.test(amountbf, y = NULL, alternative = c('two.sided',
# 'less', 'greater'), mu = 0, paired = FALSE, var.equal =
# FALSE, conf.level = 0.99)
```

```
b. Suppose that in 2018, the average amount spent shopping on Black Friday
was \$12000. Based on your sample, is there evidence to conclude that the
mean amount spent shopping on Black Friday is 2017 is less than \$12000?
Conduct an appropriate hypothesis test "by-hand" (i.e. do not use the
`t.test()` function, but still use `R`) at the $\alpha=0.05$ significance level.
\vspace{5pt}

 H0: mean >= 12000
 H1: mean amount <12000
```

```r
alpha = 0.05
mean2 = 12000
tbf = ((mean_bf - mean2) * (sqrt(lenbf)))/sd_bf
tbf
```

```
## [1] -0.8523199
```

```r
pt = pt(tbf, lenbf - 1)
pt
```

```
## [1] 0.2003949
```

```
Answer) P Value is 0.2003949
```

3. The Duke Chronicle collected data on all 1739 students listed in the Class of 2018's "Freshmen Picture Book." In particular, the Duke Chronicle examined hometowns, details about the students' high schools, whether they won a merit scholarship, and their sports team involvement. Ultimately, the goal was to determine trends between those who do and do not join Greek life at the university. A subset of this data is contained in the file named "greek.csv." The variable `greek` is an indicator that equals 1 if the student is involved in Greek life and 0 otherwise. The variable `hstuition` gives the amount of money spent on the student's high school tuition.

```r
# Read Dataset
greek <- read.csv("greek.csv.")
head(greek)
```

```
##      X greek_council organization       city       state        country
## 1 1575          None         None  New Delhi        <NA>          India
## 2 1584          None         None  Singapore        <NA>      Singapore
## 3 1582          None         None  Singapore        <NA>      Singapore
## 4   10          None         None       Wuxi        <NA>          China
## 5   11          None         None     Coogee        <NA>      Australia
## 6   14          None         None Yazoo City Mississippi United States
##   percent_Frlunch hspubpriv        domint hsboardday                hsreligion
## 1              NA    public International        Day               Unafilliated
## 2              NA    public International        Day               Unafilliated
## 3              NA    public International        Day               Unafilliated
## 4              NA   private International        Day               Unafilliated
## 5              NA   private International        Day                   Catholic
## 6              NA   private      Domestic        Day Inter-/Non-denominational
##   hsgender hstuition sports scholarship greek
## 1    Co-Ed    992.25   None        None     0
## 2    Co-Ed   2708.82   None        None     0
## 3    Co-Ed   2708.82   None        None     0
## 4 All-Boys   3165.54   None        None     0
## 5 All-Boys   4699.46   None   Robertson     0
## 6    Co-Ed   5508.00   None        None     0
```

```
# Find mean, standard deviation of money column and length
# of dataset
mean_greek_money = mean(greek$hstuition)
sd_greek_money = sd(greek$hstuition)
mean_greek_money
```

```
## [1] 27923.25
```

```
sd_greek_money
```

```
## [1] 17817.32
```

```
lengreek <- nrow(greek)
lengreek
```

```
## [1] 81
```

```
a. At the $\alpha=0.1$ significance level, test whether the average high
school tuition for a student who does not partake in Greek life is less than
the average high school tuition for a student who does partake in Greek life.
```

```
Assume unequal variances.

H0:
H1:
```

```r
greek_0 <- greek[greek$greek == "0", ]
greek_1 <- greek[greek$greek == "1", ]

mean_0 = mean(greek_0$hstuition)
mean_0
```

```
## [1] 23477
```

```r
sd_0 = sd(greek_0$hstuition)
sd_0
```

```
## [1] 14674.84
```

```r
mean_1 = mean(greek_1$hstuition)
mean_1
```

```
## [1] 34731.57
```

```r
sd_1 = sd(greek_1$hstuition)
sd_1
```

```
## [1] 20166.82
```

```r
len0 <- nrow(greek_0)
len0
```

```
## [1] 49
```

```r
len1 <- nrow(greek_1)
len1
```

```
## [1] 32
```

```r
t1 = (mean_0 - mean_1)/(sqrt(((sd_0^2)/len0) + ((sd_1^2)/len1)))
t1
```

```
## [1] -2.721299
```

```r
numerator = (((sd_0^2)/len0) + ((sd_1^2)/len1))^2
numerator
```

```
## [1] 2.925572e+14
```

```r
denominator = ((((sd_0^2)/len0)^2)/(len0 - 1) + (((sd_1^2)/len1)^2)/(len1 -
    1))
denominator
```

```
## [1] 5.613e+12
```

```r
degree = numerator/denominator
degree
```

```
## [1] 52.12135
```

```r
# welsh ttest

pval = pt(t1, degree)
pval
```

```
## [1] 0.004409615
```

```
P Value = 0.004409615
pval < alpha
Reject the null hypothesis
```

```r
# t.test(greek_0$hstuition,greek_1$hstuition)
```

```
b. Construct a one-sided, lower-bound 90% confidence interval on the mean
amount of high school tuition paid by Duke students. Interpret the result.
\vspace{5pt}
```

```r
# Calculating
z = qt(0.9, lengreek - 1)
# Finding interval size
ci = z * sd_greek_money/sqrt(lengreek)
ci
```

```
## [1] 2558.217
```

```r
lower_90 = mean_greek_money - ci
lower_90
```

```
## [1] 25365.03
```

A one-sided, lower-bound 90% confidence interval on the mean amount of high school tuition paid by Duke students is (25365.03,infinity)

4. Seven trumpet players are given a new breathing exercise to help with their breath support. The trumpet players are asked to play a C note for as long as they can both before and after the breathing exercise. The time (in seconds) that they can hold the note for are presented below. Assume times are normally distributed.

Subject	1	2	3	4	5	6	7
Before	9.1	11.2	11.9	14.7	11.7	9.5	14.2
After	10.7	14.2	12.4	14.6	16.4	10.1	19.2

```r
# Creating dataframe
ind <- c(1, 2, 3, 4, 5, 6, 7)
val <- c(1.6, 3, 0.5, -0.1, 4.7, 0.6, 5)
bef <- c(9.1, 11.2, 11.9, 14.7, 11.7, 9.5, 14.2)
aft <- c(10.7, 14.2, 12.4, 14.6, 16.4, 10.1, 19.2)
df <- data.frame(ind, bef, aft, val)
df
```

```
##   ind  bef  aft  val
## 1   1  9.1 10.7  1.6
## 2   2 11.2 14.2  3.0
## 3   3 11.9 12.4  0.5
## 4   4 14.7 14.6 -0.1
## 5   5 11.7 16.4  4.7
## 6   6  9.5 10.1  0.6
## 7   7 14.2 19.2  5.0
```

```r
# val column has the difference values
after = df$aft
before = df$bef
difference = df$val
```

```
a. Construct a one-sided lower-bound 95\% confidence interval for the mean
after-before change time holding a note. Interpret your interval.
```

```r
meandf = mean(difference)
meandf
```

```
## [1] 2.185714
```

```r
sddf = sd(difference)
sddf
```

```
## [1] 2.074792
```

```r
# Finding Lower Bound confidence interval
n <- nrow(df)
z = qt(0.95, n - 1)
z
```

```
## [1] 1.94318
```

```r
ci = z * sddf/sqrt(n)
ci
```

```
## [1] 1.523837
```

```r
low_95 = meandf - ci  # TO BE DONE
low_95
```

```
## [1] 0.6618768
```

One-sided lower-bound 95% confidence interval for the mean is (0.6618768,infinity)

```r
# t.test(difference, y = NULL, alternative = c('two.sided',
# 'less', 'greater'), paired = FALSE, var.equal = FALSE,
# conf.level = 0.95)
```

b. Perform an appropriate test at the $\alpha=0.1$ significance level to determine if the mean time holding a note is greater after the exercise than before.
\vspace{5pt}
After- Before = Difference
H0: Difference less than or equal to zero
H1: Difference greater than zero

```
meanbef = mean(before)
meanbef
```

```
## [1] 11.75714
```

```
sdbef = sd(before)
sdbef
```

```
## [1] 2.125917
```

```
meanaft = mean(after)
meanaft
```

```
## [1] 13.94286
```

```
sdaft = sd(after)
sdaft
```

```
## [1] 3.210326
```

```
tdif = meandf * sqrt(n)/(sddf)
tdif
```

```
## [1] 2.787198
```

```
deg_fred = n - 1
pdif = 1 - pt(tdif, deg_fred)
pdif
```

```
## [1] 0.01584723
```

Answer) P Value is 0.01584723
P value < Alpha
Reject null hypothesis

5. Let μ be the average amount of time in minutes spent on social media apps each day. Based on an earlier study, it is hypothesized that $\mu = 124$ minutes. It is believed, though, that people are spending increasingly more time on social media apps during the pandemic. We sample n people and determine the average amount of time spent on social media apps per day in order to test the hypotheses $H_0 : \mu \leq 124$ vs. $H_1 : \mu > 124$, at the $\alpha = 0.01$ significance level. Suppose we know that $\sigma = 26$ minutes.

    a. Create a sequence of reasonable alternative values for μ. Take $\mu_1 \in (124, 190)$, using `seq(124,190, by=0.001)` in `R`.

```r
# Sequence of reasonable alternative values
mu = seq(124, 190, by = 0.001)

# Not printing all of them as there are more than 60k
# values Printing only first 20 values
for (x in mu) {
    if (x == 124.02) {
        break
    }
    print(x)
}
```

```
## [1] 124
## [1] 124.001
## [1] 124.002
## [1] 124.003
## [1] 124.004
## [1] 124.005
## [1] 124.006
## [1] 124.007
## [1] 124.008
## [1] 124.009
## [1] 124.01
## [1] 124.011
## [1] 124.012
## [1] 124.013
## [1] 124.014
## [1] 124.015
## [1] 124.016
## [1] 124.017
## [1] 124.018
## [1] 124.019
```

```r
# power
```

b. Use ‘R‘ to draw a power curve for when $n=5$. You may find the ‘plot()‘ function useful. In particular, ‘plot(mu1, __, type = "l", ylab = "Power", xlab = expression(mu[1]))‘ could be a useful starting point for formatting.

```r
n = 5
sigma = 26
u = 124
zold = qnorm(0.99)
zold
```

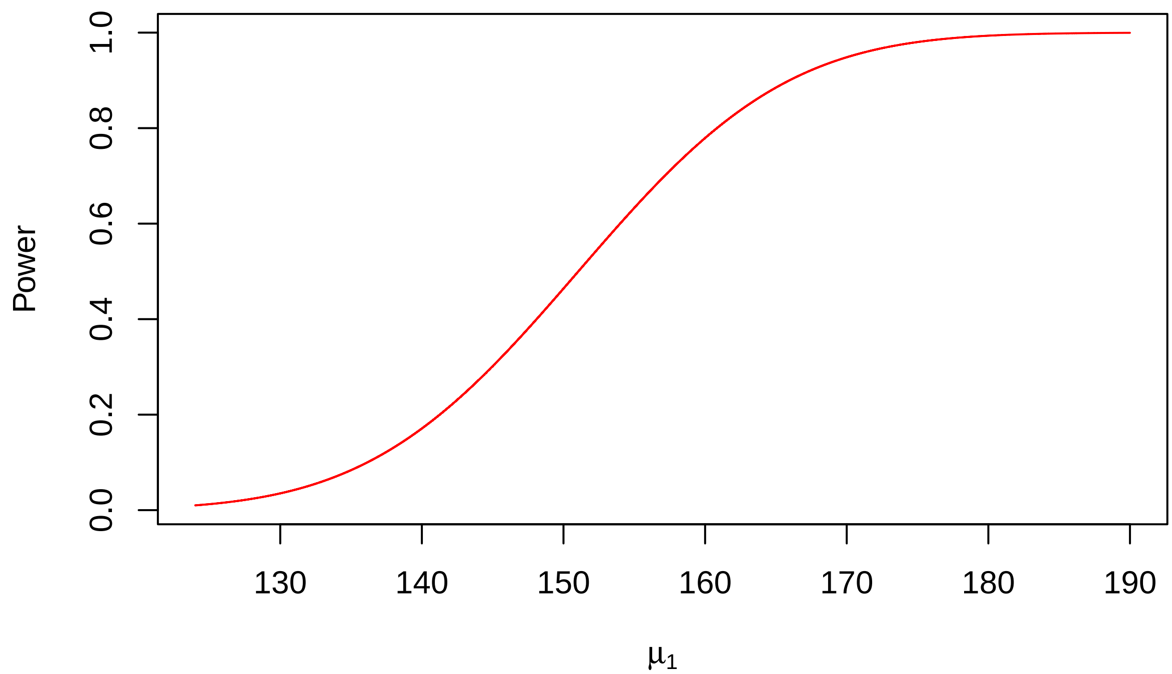
```
## [1] 2.326348
```

```r
xbar = (zold * sigma)/sqrt(n) + u
xbar
```

```
## [1] 151.0497
```

```r
znew = (xbar - mu) * sqrt(n)/sigma
Beta = pnorm(znew)
powernew = 1 - Beta

plot(mu, powernew, type = "l", ylab = "Power", xlab = expression(mu[1]),
    col = "red")
```




```
c. Using the same general plot as part b, draw power curves for when the
sample size equals $n=5,15,25,50$. You can do this using the ‘lines()‘
function in place of when you used ‘plot()‘ in part b. Make the curve for
each of these a different color, and add a legend to distinguish these curves.
```

```r
mu = seq(124, 190, by = 0.001)

sigma = 26
u = 124
n1 = 5
n2 = 15
n3 = 25
n4 = 50


xbar15 = (zold * sigma)/sqrt(n2) + u
xbar15
```

```
## [1] 139.6172
```

```r
xbar25 = (zold * sigma)/sqrt(n3) + u
xbar25
```

```
## [1] 136.097
```

```r
xbar50 = (zold * sigma)/sqrt(n4) + u
xbar50
```

```
## [1] 132.5539
```

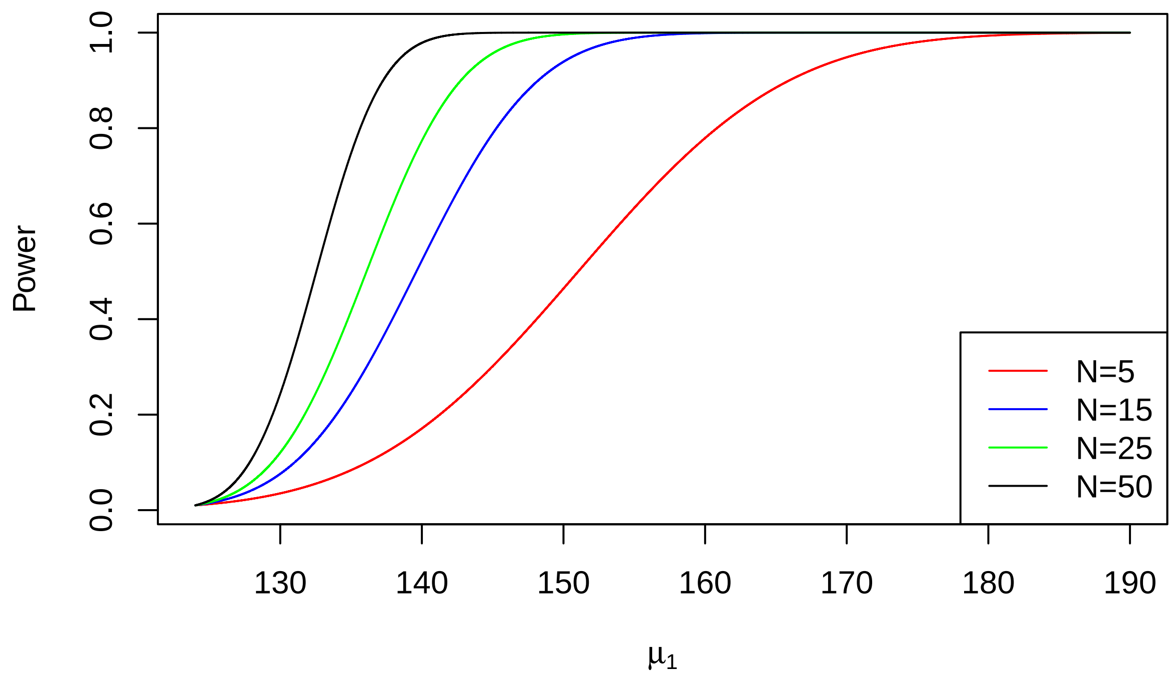
```r
z1new = ((xbar - mu) * sqrt(n1))/sigma
powernew1 = 1 - pnorm(z1new)
plot(mu, powernew1, type = "l", ylab = "Power", xlab = expression(mu[1]),
    col = "red")

z2new = ((xbar15 - mu) * sqrt(n2))/(sigma)
powernew2 = 1 - pnorm(z2new)
lines(mu, powernew2, type = "l", ylab = "Power", xlab = expression(mu[1]),
    col = "blue")

z3new = ((xbar25 - mu) * sqrt(n3))/sigma
powernew3 = 1 - pnorm(z3new)
lines(mu, powernew3, type = "l", ylab = "Power", xlab = expression(mu[1]),
    col = "green")

z4new = ((xbar50 - mu) * sqrt(n4))/sigma
powernew4 = 1 - pnorm(z4new)
lines(mu, powernew4, type = "l", ylab = "Power", xlab = expression(mu[1]),
    col = "black")

legend("bottomright", c("N=5", "N=15", "N=25", "N=50"), lty = 1,
    col = c("red", "blue", "green", "black"))
```



d. What is the power of this test when $\mu_1=141$ and $n=28$?

```
u = 124

ngiven = 28
sigma = 26
mean = 141

xbard = (zold * sigma)/sqrt(ngiven) + u
xbard
```

```
## [1] 135.4306
```

```
z = (mean - xbard) * sqrt(ngiven)/sigma
z
```

```
## [1] 1.133481
```

```r
b = 1 - pnorm(z)
b
```

```
## [1] 0.1285062
```

```r
power = 1 - b
power
```

```
## [1] 0.8714938
```

```
Power is 0.8714938

e. How large of a sample size is needed to attain a power of $0.95$ when
the true mean amount of time on social media apps is $\mu_1=128$?
\vspace{5pt}
```

```r
alpha_e = 0.99
power_e = 0.95
beta_e = 1 - power_e
beta_e
```

```
## [1] 0.05
```

```r
zalp = qnorm(alpha_e)
zalp
```

```
## [1] 2.326348
```

```r
zbet = qnorm(power_e)
zbet
```

```
## [1] 1.644854
```

```r
z_e = zalp + zbet
z_e
```

```
## [1] 3.971202
```

```r
new_m = 128

n = ((z_e * sigma)/(new_m - u))^2
n
```

```
## [1] 666.3011
```

Sample Size = 667

```r
# Check
ucheck = 124

ncheck = 666.3011
sigmacheck = 26
meancheck = 128

xbarcheck = (zold * sigmacheck)/sqrt(ncheck) + ucheck
xbarcheck
```

```
## [1] 126.3432
```

```r
zcheck = (meancheck - xbarcheck) * sqrt(ncheck)/sigmacheck
zcheck
```

```
## [1] 1.644853
```

```r
bcheck = 1 - pnorm(zcheck)
bcheck
```

```
## [1] 0.05000001
```

```r
powercheck = 1 - bcheck
powercheck
```

```
## [1] 0.95
```

6. When it is time for vacation, many of us look to Air BnB for renting a room/house. Data collected on $n = 83$ Air BnB listings in New York City are contained in the file "airbnb.csv." Read this file into R.

    a. Create two new variables: one for the price of full house rentals and one for the price of private room rentals. You can use code such as this to subset:

Answer) Home variable: Full house rental Private variable: Private room rental price_home: Price of full house rental price_private: Price of private room rental

```
air <- read.csv("airbnb.csv.")
head(air)
```

```
##          id neighbourhood_group   room_type price minimum_nights
## 1   1803165           Manhattan Entire home   799              6
## 2  13410813              Queens Entire home   120              3
## 3    941179            Brooklyn Entire home   150              2
## 4   1256768            Brooklyn Entire home   147              7
## 5   7816449           Manhattan Entire home   500              7
## 6   3415102            Brooklyn Entire home   500              2
##   number_of_reviews reviews_per_month availability_365
## 1                40              0.58              365
## 2                40              1.45              365
## 3                42              0.72              365
## 4                42              0.61              365
## 5                44              0.94              365
## 6                48              0.80              365
```

```
home <- air[air$room_type == "Entire home", ]
head(home)
```

```
##          id neighbourhood_group   room_type price minimum_nights
## 1   1803165           Manhattan Entire home   799              6
## 2  13410813              Queens Entire home   120              3
## 3    941179            Brooklyn Entire home   150              2
## 4   1256768            Brooklyn Entire home   147              7
## 5   7816449           Manhattan Entire home   500              7
## 6   3415102            Brooklyn Entire home   500              2
##   number_of_reviews reviews_per_month availability_365
## 1                40              0.58              365
## 2                40              1.45              365
## 3                42              0.72              365
## 4                42              0.61              365
## 5                44              0.94              365
## 6                48              0.80              365
```

```
price_home = home$price
price_home
```

```
##  [1] 799 120 150 147 500 500 299 180 250 500 250 105 200 150 300  99 895 200 150
## [20] 165 150 105 200  60 125 249 125
```

```r
private <- air[air$room_type == "Private room", ]
head(private)
```

```
##          id neighbourhood_group    room_type price minimum_nights
## 28  2160591            Brooklyn Private room    70              3
## 29  4093399               Bronx Private room    68              2
## 30 26984883            Brooklyn Private room    95              2
## 31    94035              Queens Private room    80              1
## 32   158290            Brooklyn Private room    75              3
## 33 21139541              Queens Private room    43              2
##    number_of_reviews reviews_per_month availability_365
## 28                40              0.63              365
## 29                41              0.74              365
## 30                41              3.50              365
## 31                42              1.21              365
## 32                43              0.44              365
## 33                43              1.99              365
```

```r
price_private = private$price
price_private
```

```
##  [1]  70  68  95  80  75  43 100 109  70 150  85  39 120  89  65  55 100  68 150
## [20]  55 319 110  45  60  54  89  58  59  89  55  80  55  39 129 135 149 259  72
## [39]  75  80 135  50 150 119  70  69  80 125  69  80  77 150  80  48  50  99
```

```r
nhome <- nrow(home)
nprivate <- nrow(private)
nhome
```

```
## [1] 27
```

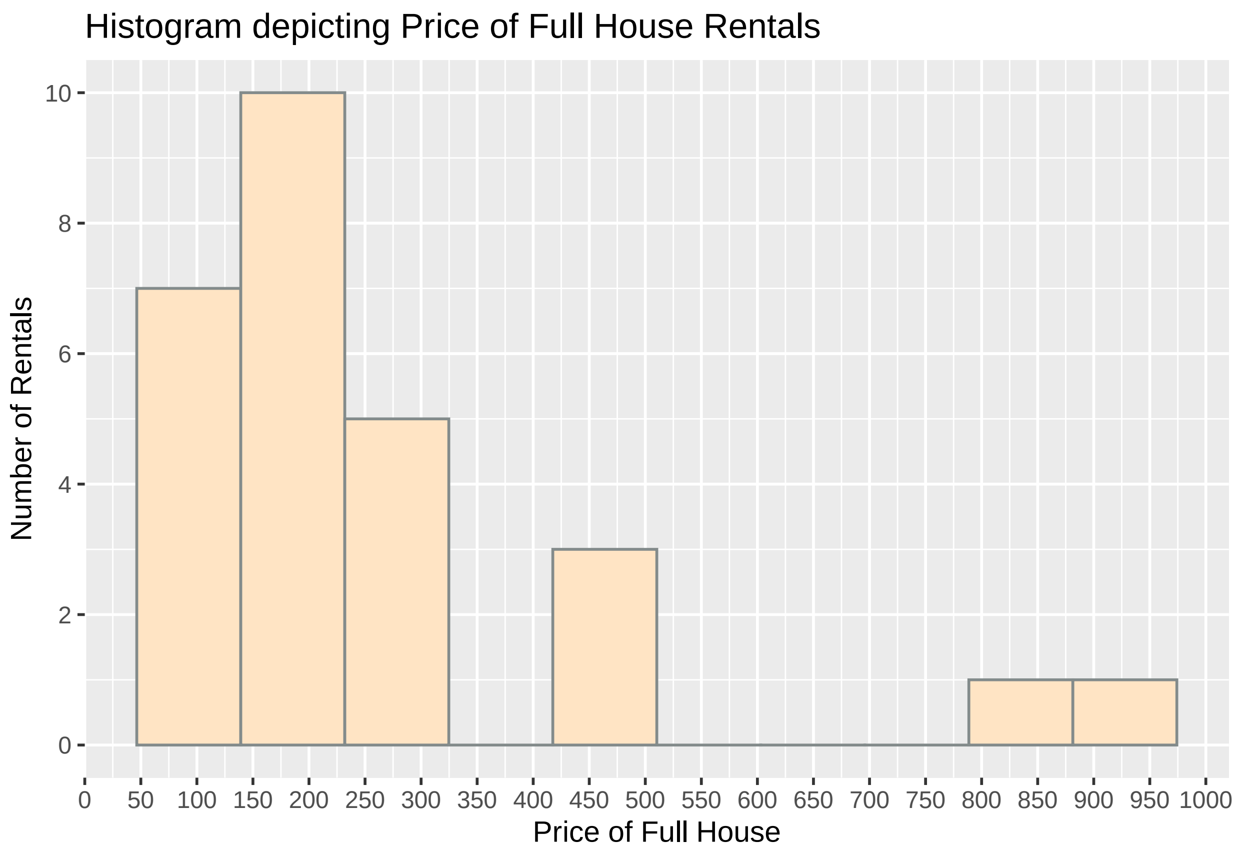
```r
nprivate
```

```
## [1] 56
```

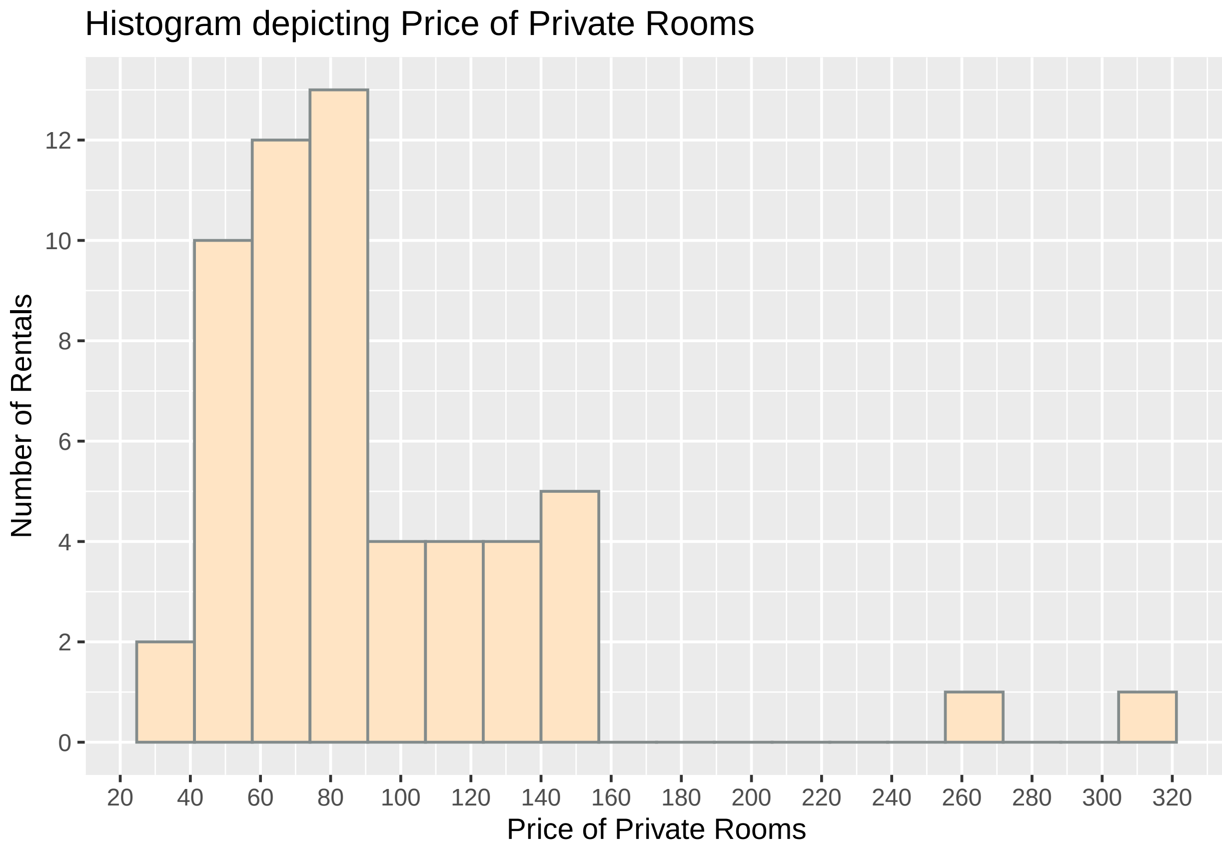
b. Make a histogram for each of the new variables from part a to visualize their distributions. You can use base R or ggplot2.

```r
# Histogram depicting Price of Full House Rentals
library(ggplot2)

histohome <- ggplot(home, aes(x = price_home)) + geom_histogram(bins = 10,
    color = "azure4", fill = "bisque") + ggtitle("Histogram depicting Price of Full Hous
    labs(y = "Number of Rentals", x = "Price of Full House") +
    scale_x_continuous(breaks = seq(0, 1000, by = 50)) + scale_y_continuous(breaks = seq
    10, by = 2))
# bins = 10 as n= 27 for private rooms, Used Sturges
# formula
histohome
```




```r
# Histogram depicting Price of Private Rooms
histoprivate <- ggplot(private, aes(x = price_private)) + geom_histogram(bins = 18,
    color = "azure4", fill = "bisque") + ggtitle("Histogram depicting Price of Private R
    labs(y = "Number of Rentals", x = "Price of Private Rooms") +
    scale_x_continuous(breaks = seq(0, 400, by = 20)) + scale_y_continuous(breaks = seq(
    20, by = 2))
# bins = 18 as n= 56 for private rooms, Used Sturges
# formula
histoprivate
```




```
c. Discuss why we generally can apply the central limit theorem to analyze
these two variables.
You should mention the histogram and the sample size, along with any potential
reservations you have about using the CLT here.

Answer)
Central limit theorem can't be applied to fll house rental, n is 27
whcih is less than 30, and for central theorem to be used this condition
has to be satisfied.
For private rooms, n=56, which is sufficiently large to apply CLT. Even the
graph is quite normal except few points on the right side.



d. Calculate the mean, standard deviation, and sample size for the price
of full home rentals.
```

```
meanhome = mean(price_home)
# Mean
meanhome
```

```
## [1] 258.2593
```

```r
sdhome = sd(price_home)
# Standard Deviation
sdhome
```

```
## [1] 208.2271
```

```r
n1 <- nrow(home)
# Sample Size
n1
```

```
## [1] 27
```

```
e. Calculate the mean, standard deviation, and sample size for the price of
private room rentals.
```

```r
meanprivate = mean(price_private)
# Mean
meanprivate
```

```
## [1] 91.92857
```

```r
sdprivate = sd(price_private)
# Standard Deviation
sdprivate
```

```
## [1] 49.91005
```

```r
n2 <- nrow(private)
# Sample Size
n2
```

```
## [1] 56
```

```
f. At the $\alpha=0.05$ significance level, test "by-hand" (i.e. do not
use the ‘t.test()‘ function, but still use ‘R‘) whether the average price of
renting an entire home in NYC is different from the average price of renting
a private room. Use unequal variances.

H0:
H1:
```

```r
s1 = sdhome^2
s2 = sdprivate^2
t = (meanhome - meanprivate)/(sqrt((s1/n1) + (s2/n2)))
t
```

```
## [1] 4.094341
```

```r
numerator1 = (((s1)/n1) + ((s2)/n2))^2
denominator1 = ((((s1)/n1)^2)/(n1 - 1) + (((s2)/n2)^2)/(n2 -
    1))
degree1 = numerator1/denominator1
degree1
```

```
## [1] 27.45038
```

```r
pval = 2 * (1 - pt(t, degree1))
pval
```

```
## [1] 0.0003360658
```

```r
# t.test(price_home,price_private)
```

```
Reject Null Hypothesis
```

Short Answers:

- About how long did this assignment take you? Did you feel it was too long, too short, or reasonable? 5-6 Hours, It was reasonable
- Who, if anyone, did you work with on this assignment? No one
- What questions do you have relating to any of the material we have covered so far in class?
  Hypothesis testing with two variables, Confidence Intervals, Power.