Project 3
STAT 355

Sabbir Ahmed

May 14, 2017

1 Part 1

1.1 Question

An oceanographer wants to test, on the basis of a random sample of size 35, whether the average depth of the ocean in a certain area is 72.4 fathoms. At the 0.05 level of significance, what will the oceanographer decide if she gets a sample mean of 73.2? Assume the population standard deviation is 2.1.

1.2 Answer

The null hypothesis, H_0, claims the mean depth of the ocean in a certain area is 72.4, while the alternative hypothesis, H_a, says otherwise.

$$H_0 : \mu = 72.4 \; vs \; H_a : \mu \neq 72.4$$

Since the population mean and standard deviation are known with a sample size of $n > 30$, the Z-score was calculated as follows:

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{73.2 - 72.4}{2.10/\sqrt{35}} = 2.2537$$

The following snippet was used to generate the Z-test and its probability:

```
X <- 73.2
mu <- 72.4
sigma <- 2.1
n <- 35
alpha <- 0.05

dumpComputation(X=X, mu=mu, sigma=sigma, n=n,
    alpha=alpha, distType="Z", twoSided=TRUE, "part1")
```

The test statistic was computed to be:

$$Z_{1-0.05/2} = Z_{0.975} = 1.96 < 2.2537$$

The p-value was computed with the following snippet:

```
pScore <- 2 * (1 - (pnorm(score)))
# 0.0242
```

Since $Z_{\alpha/2} < Z$ and the p-score was under 0.05, the null hypothesis is rejected.

2 Part 2

2.1 Question

A random sample of 12 graduates of a secretarial school averaged 73.2 words per minute with a standard deviation of 7.9 words per minute on a typing test. What can we conclude, at the 0.05 level, regarding the claim that secretaries at this school average less than 75 words per minute on the typing test?

2.2 Answer

The null hypothesis, H_0, claims the school averaged greater or equal to 75, while the alternative hypothesis, H_a, says otherwise.

$$H_0 : \mu < 75 \; vs \; H_a : \geq 75$$

Since the population standard deviation is unknown, and the sample was $n < 30$, the t-score was calculated as follows:

$$t = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{73.2 - 75.0}{7.90/\sqrt{12}} = -0.7893$$

The following snippet was used to generate the t-test and its probability:

```
X <- 73.2
mu <- 75
s <- 7.9
n <- 12

dumpComputation(X=X, mu=mu, sigma=s, n=n,
    alpha=alpha, distType="t", twoSided=FALSE, "part2")
```

The test statistic was computed to be:

$$\because t_{1-0.05,11} = t_{0.95,11} = 1.7958 > -0.7893$$

The p-value was computed with the following snippet:

```
pScore <- 1-pt(score, df=n-1)
# 0.7767
```

Since $t_{1-\alpha,df} > t$ and the very high p-value, there is strong evidence to not reject the null hypothesis.

3 Part 3

3.1 Question

The weights of mature dogs of a certain breed approximately follow a normal distribution. Five dogs selected at random weighed 66, 63, 64, 62 and 65 pounds. A kennel club claims that the average weight for this breed is 60 pounds. Using the 0.05 level of significance, do we have reason to doubt this claim?

3.2 Answer

The null hypothesis, H_0, claims the mean weight of the breed is 60 pounds, while the alternative hypothesis, H_a, says otherwise.

$$H_0 : \mu = 60 \; vs \; H_a : \mu \neq 60$$

Since the population standard deviation is unknown, and the sample was $n < 30$, the t-score was calculated as follows:
The sample mean and standard deviation were calculated:

```
weights <- c(66, 63, 64, 62, 65)
X <- mean(weights)
s <- sd(weights)
```

$$t = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{64.0 - 60.0}{1.58/\sqrt{5}} = 5.6569$$

The following snippet was used to generate the t-test and its probability:

```
weights <- c(66, 63, 64, 62, 65)
X <- mean(weights)
s <- sd(weights)
n <- length(weights)
mu <- 60

weightsSeq <- seq(1,length(weights))

df <- data.frame(weightsSeq, weights)

dumpComputation(X=X, mu=mu, sigma=s, n=n,
    alpha=alpha, distType="t", twoSided=TRUE, "part3")
```

The test statistic was computed to be:

$$\because t_{1-0.025,4} = t_{0.975,4} = 2.7764 < 5.6569$$

The p-value was computed with the following snippet:

```
pScore <- 2 * (1 - pt(score, df=n-1))
# 0.004812
```

Since $t_{\alpha,df} < t$ and the p-score was well under 0.05, the null hypothesis is rejected.

The box plot and normal probability plot were generated:

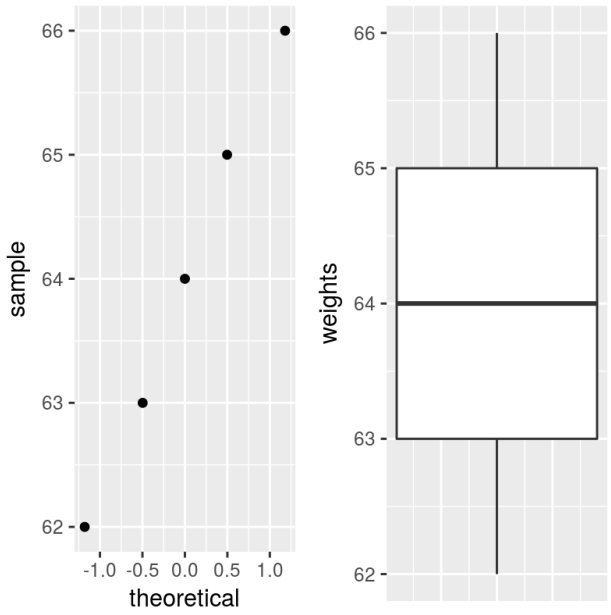


Figure 1: Normal Probability Plot and Box Plot of the Distribution

4 Part 4

4.1 Question

Suppose the lengths in millimeters of metal fibers produced by a certain process have a normal distributions for which the mean was 5.2 and the variance 0.8. After a system upgrade of the process, the engineer wants to test whether the mean length and the variance have changed or not. He took a sample of 15 fibers and measured. The sample mean was 5.4 and the sample variance was 1.0. Based on these measurements, state H_0 and H_a and conduct a test to verify the engineer's concern with $\alpha = 0.05$.

4.2 Answer

The null hypothesis, H_0, claims the mean of the lengths of the metal fibers after the upgrade is still 5.2, while the alternative hypothesis, H_a, says otherwise.

$$H_0 : \mu = 5.2 \; vs \; H_a : \mu \neq 5.2$$

Since the population mean and standard deviation are known with a sample size of $n < 30$, the Z-score was calculated as follows:

$$t = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{5.4 - 5.2}{0.89/\sqrt{15}} = 0.8660$$

The following snippet was used to generate the t-test and its probability:

```
X <- 5.4
mu <- 5.2
sigma <- sqrt(0.8)
```

```
n <- 15

dumpComputation(X=X, mu=mu, sigma=sigma, n=n,
    alpha=alpha, distType="t", twoSided=TRUE, "part4")
```

The test statistic was computed to be:

$$t_{0.025,14} = -2.14 < 0.8660 < t_{1-0.025,14} = 2.14$$

The p-value was computed with the following snippet:

```
pScore <- 2 * (1 - (pt(score, df=n-1)))
# 0.4011
```

Since $t_{\alpha/2} < t < t_{1-\alpha/2}$ and the p-score was well over 0.05, there is strong evidence to not reject the null hypothesis.

The null hypothesis for the variance claims the population variance is 0.8, while the alternative hypothesis proposes otherwise.

$$H_0 : \sigma^2 = 0.8 \; vs \; H_a : \sigma^2 \neq 0.8$$

As a separate test, the chi-squared hypothesis test is used:

$$T = (n-1)\frac{s^2}{\sigma^2} = (14)\frac{1}{0.64} = 21.8750$$

The test statistic was computed to be:

$$\chi^2_{0.025,14} = 5.63 < 21.8750$$

The p-value for the Chi-squared test was computed with the following snippet:

```
pScore <- 2 * (1 - (pchisq(score, df=n-1)))
# 0.1624
```

The high p-value also suggests sufficient evidence to not reject the null hypothesis based on the variance testing.

5 Part 5

Example 8.11 was simulated to plot the power curve for one sample t testing. Figure 2 was reproduced below with the following snippet:

```
sigma <- 0.1 # standard deviation
alpha <- qnorm(1-0.05) # P(Z > alpha)
xSeq <- c(0, 0.2) # x bounds

png(
    filename="figures/part5.png",
    units="in", width=6, height=4, res=200
)
ggplot(NULL, aes(x=x, colour=n, fill=n)) +
    stat_function(data=data.frame(x=xSeq, n=factor(5)),
        fun=function(x) { pnorm(sqrt(5)*x/sigma - alpha) }) +
    stat_function(data=data.frame(x=xSeq, n=factor(10)),
        fun=function(x) { pnorm(sqrt(10)*x/sigma - alpha) }) +
```

```
    stat_function(data=data.frame(x=xSeq, n=factor(15)),
        fun=function(x) { pnorm(sqrt(15)*x/sigma - alpha) }) +
    ylab("Power") + xlab("Difference") + labs(colour="Sample Size")
dev.off()
```

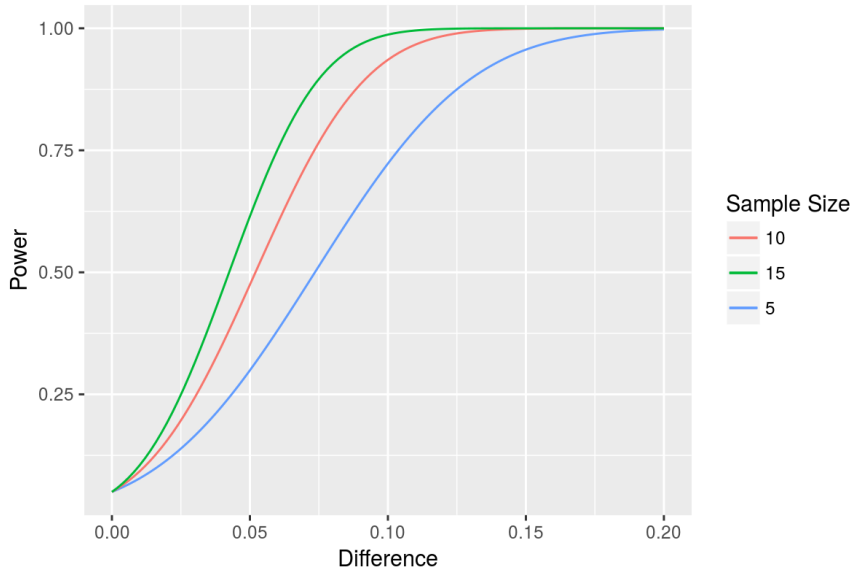


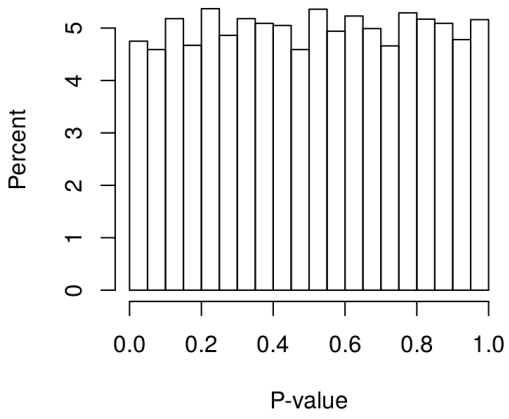
Figure 2: Power Curves for the t test Simulated from Example 8.11

6 Part 6

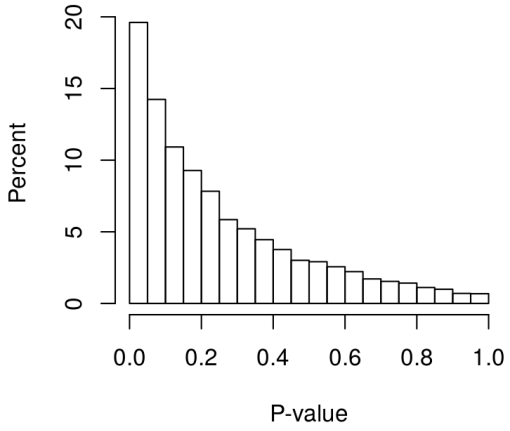
Example 8.12 was simulated to plot the distribution of the p-values generated from the one sample t testing. Figure 6 was reproduced below with the following snippet:

```
NUMSAMPS <- 10000

# initialize distribution variables
generateHist <- function(mu, filename) {

    # initialize empty arrays
    sampPScores <- generatedData <- rep(0, times=NUMSAMPS)

    # generate 10000 samples
    for (i in 1:NUMSAMPS) {

        generatedData <- rnorm(4, 20, 2)

        # store the sample means in vector
        xbar <- mean(generatedData)
        s <- sd(generatedData)

        tScore <- (xbar - mu)/(s/2)
        sampPScores[i] = pt(tScore, df=3)

    }
```

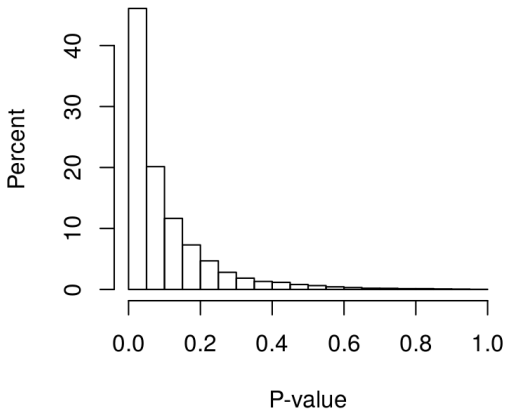
```
    # generate density histograms to display percentage
    h <- hist(
        sampPScores
    )
    h$density = h$counts/sum(h$counts)*100

    # dump plots into PNG
    png(
        filename=paste0("figures/", filename),
        units="in", width=4, height=4, res=200
    )
    plot(
        h, freq=FALSE,
        main=NULL,
        xlab="P-value",
        ylab="Percent"
    )
    dev.off()

}

generateHist(mu=20, filename="part6a.png")
generateHist(mu=21, filename="part6b.png")
generateHist(mu=22, filename="part6c.png")
```



(a) $\mu = 20$



(b) $\mu = 21$



(c) $\mu = 22$

Figure 3: P-value Simulations for Example 8.12

main.R

```
# main.R
# This file contains the implementation of the functions in the Project 3
# NOTE: THIS SCRIPT WAS COMPILED ON A LINUX MACHINE - SOME STATEMENTS MAY THROW
# WARNINGS OR ERRORS IN OTHER SYSTEMS

library(ggplot2)
library(gridExtra)
library(scales)
set.seed(0)  # seed the random generators

scoreTemplate <-
    "\\begin{equation*}
    %s=\\frac{\\overline{X}-\\mu}{\\sfrac{\\sigma}{\\sqrt{n}}}
    =\\frac{%0.1f-%0.1f}{\\sfrac{%0.2f}{\\sqrt{%d}}}=%0.4f
    \\end{equation*}"


dumpComputation <- function(X, mu, sigma, n, alpha,
    distType, twoSided, outputFile) {

    score <- (X - mu)/(sigma/sqrt(n))

    tableVal <- 0
    pScore <- 0

    if (distType == "Z") {

        tableVal <- qnorm(alpha/2)
        pScore <- 2 * (1 - (pnorm(score)))

    } else if (distType == "t") {

        if (twoSided) {
            tableVal <- qt(alpha/2, df=n-1)
            pScore <- 2 * (1 - pt(score,df=n-1))
        } else {
            tableVal <- qt(alpha, df=n-1)
            pScore <- 1 - pt(score, df=n-1)
        }

    }

    # dump output to LaTex modules
    sink(
        paste0("latex_mods/", outputFile, "_out.tex"),
        append=FALSE, split=FALSE
    )
    cat(
        sprintf(scoreTemplate,
            distType, X, mu, sigma, n,
            score, distType)
    )
    sink()  # return stdout to console
    print(outputFile)
    print(paste("Score:", tableVal))
    print(paste("P-value:", pScore))

}



# ------------------------------ Part 1 ------------------------------

X <- 73.2
mu <- 72.4
```

```
sigma <- 2.1
n <- 35
alpha <- 0.05

dumpComputation(X=X, mu=mu, sigma=sigma, n=n,
    alpha=alpha, distType="Z", twoSided=TRUE, "part1")


# ------------------------------ Part 2 ------------------------------

X <- 73.2
mu <- 75
s <- 7.9
n <- 12

dumpComputation(X=X, mu=mu, sigma=s, n=n,
    alpha=alpha, distType="t", twoSided=FALSE, "part2")


# ------------------------------ Part 3 ------------------------------

weights <- c(66, 63, 64, 62, 65)
X <- mean(weights)
s <- sd(weights)
n <- length(weights)
mu <- 60

weightsSeq <- seq(1,length(weights))

df <- data.frame(weightsSeq, weights)

dumpComputation(X=X, mu=mu, sigma=s, n=n,
    alpha=alpha, distType="t", twoSided=TRUE, "part3")

probNormPlt <- ggplot(df, aes(sample=weights)) + stat_qq()
boxPlt <- ggplot(df, aes(x=weightsSeq, y=weights)) + geom_boxplot() +
    theme(
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()
    )

# save plot to filename
png(filename="figures/part3.png", units="in", width=4, height=4, res=200)
grid.arrange(probNormPlt, boxPlt, ncol=2)
dev.off()


# ------------------------------ Part 4 ------------------------------

X <- 5.4
mu <- 5.2
sigma <- sqrt(0.8)
n <- 15

dumpComputation(X=X, mu=mu, sigma=sigma, n=n,
    alpha=alpha, distType="t", twoSided=TRUE, "part4")


# ------------------------------ Part 5 ------------------------------

sigma <- 0.1  # standard deviation
alpha <- qnorm(1-0.05)  # P(Z > alpha)
xSeq <- c(0, 0.2)  # x bounds

png(
    filename="figures/part5.png",
    units="in", width=6, height=4, res=200
```

```
)
ggplot(NULL, aes(x=x, colour=n, fill=n)) +
    stat_function(data=data.frame(x=xSeq, n=factor(5)),
        fun=function(x) { pnorm(sqrt(5)*x/sigma - alpha) }) +
    stat_function(data=data.frame(x=xSeq, n=factor(10)),
        fun=function(x) { pnorm(sqrt(10)*x/sigma - alpha) }) +
    stat_function(data=data.frame(x=xSeq, n=factor(15)),
        fun=function(x) { pnorm(sqrt(15)*x/sigma - alpha) }) +
    ylab("Power") + xlab("Difference") + labs(colour="Sample Size")
dev.off()

# ------------------------------ Part 6 ------------------------------

NUMSAMPS <- 10000

# initialize distribution variables
generateHist <- function(mu, filename) {

    # initialize empty arrays
    sampPScores <- generatedData <- rep(0, times=NUMSAMPS)

    # generate 10000 samples
    for (i in 1:NUMSAMPS) {

        generatedData <- rnorm(4, 20, 2)

        # store the sample means in vector
        xbar <- mean(generatedData)
        s <- sd(generatedData)

        tScore <- (xbar - mu)/(s/2)
        sampPScores[i] = pt(tScore, df=3)

    }

    # dump plots into PNG
    png(
        filename=paste0("figures/", filename),
        units="in", width=4, height=4, res=200
    )

    # generate density histograms to display percentage
    h <- hist(
        sampPScores
    )
    h$density = h$counts/sum(h$counts)*100
    plot(
        h, freq=FALSE,
        main=NULL, xlab="P-value", ylab="Percent"
    )

    dev.off()

}

generateHist(mu=20, filename="part6a.png")
generateHist(mu=21, filename="part6b.png")
generateHist(mu=22, filename="part6c.png")
```
