Project 3
STAT 355

Sabbir Ahmed

May 13, 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 was known with a sample size of $n > 30$, the Z-score was calculated as follows:

$$Z = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{73.2 - 72.4}{2.1/\sqrt{35}} = 2.2537$$

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

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

z <- (X - mu)/(sigma/sqrt(n))
print(z) # print the Z-score
print(pnorm(z)) # print the probability
```

The test statistic was computed to be:

$$\pm Z_{0.05} = \pm 1.96 < 2.2537$$

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:

$$t = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{73.2 - 75.0}{7.9/\sqrt{12}} = -0.7893$$

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

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

z <- (X - mu)/(sigma/sqrt(n))
print(z) # print the Z-score
print(pnorm(z)) # print the probability
```

The test statistic was computed to be:

$$\because t_{0.05,11} = -1.7958 < -0.7893$$

The p-value was computed with the following snippet:

```
pScore <- pt(score, df=n-1)
# 0.2233
```

Since $Z_{\alpha/2} < Z$, the null hypothesis is rejected.

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)
```

$$t = \frac{\overline{X} - \mu}{\sigma/\sqrt{n}} = \frac{64.0 - 60.0}{1.6/\sqrt{5}} = 5.6569$$

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

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

z <- (X - mu)/(sigma/sqrt(n))
print(z) # print the Z-score
print(pnorm(z)) # print the probability
```

The test statistic was computed to be:

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

The p-value was computed with the following snippet:

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

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

References

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)
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{%0.1f}{\\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 <- 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(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))

data <- data.frame(weightsSeq, weights)

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

probNormPlt <- ggplot(data, aes(sample=weights))+stat_qq()
boxPlt <- ggplot(data, 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")
grid.arrange(probNormPlt, boxPlt, ncol=2)
dev.off()
```