Homework6

Adyan Rahman

Problem 1:

(a) (ox) = Population standard deviation / Squareroot(sample size) = 1 minute / Squareroot(60) = 0.129 minutes

So, ux = 8.2 minutes and ox = 0.129 minutes.

(b)

X = u + (Z * o)

- X is the raw score (sample mean)
- u is the population mean
- Z is the z-score
- o is the population standard deviation

X = 8.2 + (1.28 * 1) = 9.48 minutes.

The 90th percentile for the sample mean time for app engagement for a tablet user is approximately 9.48 minutes. This means that 90% of the sample means fall below 9.48 minutes.

(c)

```
# Define parameters
mu <- 8.2  # Population mean
sigma <- 1  # Population standard deviation
sample_size <- 60  # Sample size

# (c) Find the probabilities that the sample mean is between ±1 standard deviation, ±2 standard deviati

# ±1 standard deviation
prob_1sd <- pnorm(mu + sigma / sqrt(sample_size), mean = mu, sd = sigma) -
  pnorm(mu - sigma / sqrt(sample_size), mean = mu, sd = sigma)

# ±2 standard deviations
prob_2sd <- pnorm(mu + 2 * sigma / sqrt(sample_size), mean = mu, sd = sigma) -
  pnorm(mu - 2 * sigma / sqrt(sample_size), mean = mu, sd = sigma)

# ±3 standard deviations
prob_3sd <- pnorm(mu + 3 * sigma / sqrt(sample_size), mean = mu, sd = sigma) -
  pnorm(mu - 3 * sigma / sqrt(sample_size), mean = mu, sd = sigma)

# Print probabilities
cat("Probability that the sample mean is within ±1 standard deviation:", prob_1sd, "\n")
```

```
## Probability that the sample mean is within ±1 standard deviation: 0.102721
```

```
cat("Probability that the sample mean is within ±2 standard deviations:", prob_2sd, "\n")
```

```
## Probability that the sample mean is within ±2 standard deviations: 0.2037466
```

```
cat("Probability that the sample mean is within ±3 standard deviations:", prob_3sd, "\n")
```

```
## Probability that the sample mean is within ±3 standard deviations: 0.3014646
```

(d) Yes, there is another way to approach part (c) without using calculations. It involves using the empirical rule, also known as the 68-95-99.7 rule, which states that:

Approximately 68% of the data falls within one standard deviation of the mean. Approximately 95% of the data falls within two standard deviations of the mean. Approximately 99.7% of the data falls within three standard deviations of the mean.

For ±1 standard deviation: Probability = 68.27% (from the empirical rule)

For ±2 standard deviations: Probability = 95.45% (from the empirical rule)

For ±3 standard deviations: Probability = 99.73% (from the empirical rule)

Problem 2:

(a)

```
# Load the stats package for exponential distribution functions
library(stats)

# Define parameters
mean_excess_data <- 2  # Mean excess data in Gigabytes (Gb)
sample_size <- 80  # Sample size
excess_data_threshold <- 2.5  # Excess data threshold in Gigabytes (Gb)

# (a) Probability that one customer's excess data use is larger than 2.5 Gb
prob_exceed_threshold <- 1 - pexp(excess_data_threshold, rate = 1 / mean_excess_data)
cat("Probability that one customer's excess data use is larger than 2.5 Gb:",
    prob_exceed_threshold, "\n")
```

```
## Probability that one customer's excess data use is larger than 2.5 Gb: 0.2865048
```

(b)

```
# For the sampling distribution of the sample mean, we use the central limit theorem.
# The mean of the sampling distribution will still be 2 Gb, but the standard deviation will be (mean /
sd_sampling_distribution <- mean_excess_data / sqrt(sample_size)
prob_sample_mean_exceed_threshold <- 1 - pnorm(excess_data_threshold, mean =
                          mean_excess_data, sd = sd_sampling_distribution)
cat("Probability that the average excess data used by the 80 customers in
    the sample is larger than 2.5 Gb:", prob_sample_mean_exceed_threshold, "\n")
```

```
## Probability that the average excess data used by the 80 customers in
##     the sample is larger than 2.5 Gb: 0.01267366
```

(c) The probabilities in a and b are different because a deals with a single customer's excess data use, while b deals with the average excess data used by a sample of 80 customers. The central limit theorem states that as sample size increases, the sampling distribution of the sample mean approaches a normal distribution. Hence, b involves using the normal distribution to approximate the probability of the sample mean exceeding a certain threshold, which can be different from the probability of a single observation exceeding the same threshold.

Problem 3:

```r
# Define parameters
n <- 30  # Sample size
p_hat <- 22 / n  # Sample proportion
population_proportion <- 0.65  # Population proportion
confidence_level <- 0.95  # Confidence level

# Calculate the Z-score for the confidence level
Z <- qnorm(1 - (1 - confidence_level) / 2)

# Calculate the standard error
SE <- sqrt((p_hat * (1 - p_hat)) / n)

# Calculate the margin of error
margin_of_error <- Z * SE

# Calculate the confidence interval
lower_bound <- p_hat - margin_of_error
upper_bound <- p_hat + margin_of_error

# Check if 65% falls within the confidence interval
contains_65_percent <- lower_bound <= population_proportion && population_proportion <= upper_bound

# Print the results
cat("Confidence interval:", lower_bound, "-", upper_bound, "\n")
```

```
## Confidence interval: 0.575091 - 0.8915756
```

```r
if (contains_65_percent) {
  cat("Yes, the confidence interval contains the value of 65%.\n")
} else {
  cat("No, the confidence interval does not contain the value of 65%.\n")
}
```

```
## Yes, the confidence interval contains the value of 65%.
```

Problem 4:

```r
# Define parameters
n <- 30  # Sample size
x <- 2   # Number of left-handed students
```

```r
p_hat <- x / n  # Sample proportion
population_proportion <- 0.131  # Population proportion
confidence_level <- 0.95  # Confidence level

# Calculate the Z-score for the confidence level
Z <- qnorm(1 - (1 - confidence_level) / 2)

# Calculate the standard error
SE <- sqrt((p_hat * (1 - p_hat)) / n)

# Calculate the margin of error
margin_of_error <- Z * SE

# Calculate the confidence interval
lower_bound <- p_hat - margin_of_error
upper_bound <- p_hat + margin_of_error

# Check if 13.1% falls within the confidence interval
contains_13_1_percent <- lower_bound <= population_proportion && population_proportion <= upper_bound

# Print the results
cat("Confidence interval:", lower_bound, "-", upper_bound, "\n")
```

```
## Confidence interval: -0.02259402 - 0.1559274
```

```r
if (contains_13_1_percent) {
  cat("Yes, the confidence interval contains the value of 13.1%.\n")
} else {
  cat("No, the confidence interval does not contain the value of 13.1%.\n")
}
```

```
## Yes, the confidence interval contains the value of 13.1%.
```

Problem 5:

```r
library(UsingR)
```

```
## Warning: package 'UsingR' was built under R version 4.3.3
```

```
## Loading required package: MASS
```

```
## Loading required package: HistData
```

```
## Warning: package 'HistData' was built under R version 4.3.3
```

```
## Loading required package: Hmisc
```

```
## Warning: package 'Hmisc' was built under R version 4.3.3
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```r
# Load the babies dataset
data(babies)

# Calculate mean ages for mothers and fathers
mean_age_mothers <- mean(babies$age)
mean_age_fathers <- mean(babies$dage)

# Calculate standard deviations for mothers and fathers
sd_age_mothers <- sd(babies$age)
sd_age_fathers <- sd(babies$dage)

# Sample sizes for mothers and fathers
n_mothers <- length(babies$age)
n_fathers <- length(babies$dage)

# Calculate the standard error of the difference in means
SE_diff_means <- sqrt((sd_age_mothers^2 / n_mothers) + (sd_age_fathers^2 / n_fathers))

# Calculate the t-statistic
t_statistic <- (mean_age_mothers - mean_age_fathers) / SE_diff_means

# Degrees of freedom
df <- n_mothers + n_fathers - 2

# Calculate the critical t-value for a 95% confidence interval
t_critical <- qt(0.975, df)

# Calculate the margin of error
margin_of_error <- t_critical * SE_diff_means

# Calculate the confidence interval
lower_bound <- (mean_age_mothers - mean_age_fathers) - margin_of_error
upper_bound <- (mean_age_mothers - mean_age_fathers) + margin_of_error

# Check if the confidence interval contains 0
contains_zero <- lower_bound <= 0 && upper_bound >= 0

# Print the results
cat("95% Confidence Interval for Difference in Mean Age (Mothers - Fathers):",
    lower_bound, "to", upper_bound, "\n")
```

```
## 95% Confidence Interval for Difference in Mean Age (Mothers - Fathers): -3.962047 to -2.769345
```

```r
if (contains_zero) {
  cat("Yes, the confidence interval contains 0.\n")
} else {
  cat("No, the confidence interval does not contain 0.\n")
}
```

```
## No, the confidence interval does not contain 0.
```