Activity 5 - [Jawadul Chowdhury]

Due Date: April 18, 2025

Problem 1:

Contained in the file "Activity5PGA.csv" (found in Canvas) is data for the average driving distance for each PGA Tour golf from the 2022-2023 season. Make sure that file is in the same folder as this .Rmd file, and run the chunk of code below so that you can use it. I've called the dataset PGA, but you can certainly rename if it you'd like.

```r
# reading the .csv file as a data frame
pga_df <- read.csv("Activity5PGA.csv")
```

Your goal for this problem is to use a random sample of size 25 to create confidence intervals for the mean and standard deviation.

Part (a)

First, we need a random sample of size 25. Use the set.seed() function first, so that your specific results are reproducible. Then use the sample() function to create a random sample of size 25 from the "AVG" column of the data. I would recommend assigning your sample a name (such as "my_sample") for ease of use in future parts.

```r
# setting the seed to ensure reproducibility
set.seed(45)

# obtaining a random sample of size 25 from "AVG"
avg_sample <- sample(pga_df$AVG, 25)

# seeing the results of the sampling
avg_sample
```

```
##  [1] 308.6 303.4 289.3 308.1 305.0 302.2 314.0 304.5 300.7 299.1 299.6 292.1
## [13] 304.1 310.6 302.3 313.5 296.7 310.4 304.9 293.4 299.5 297.7 300.4 302.7
## [25] 321.7
```

Part (b)

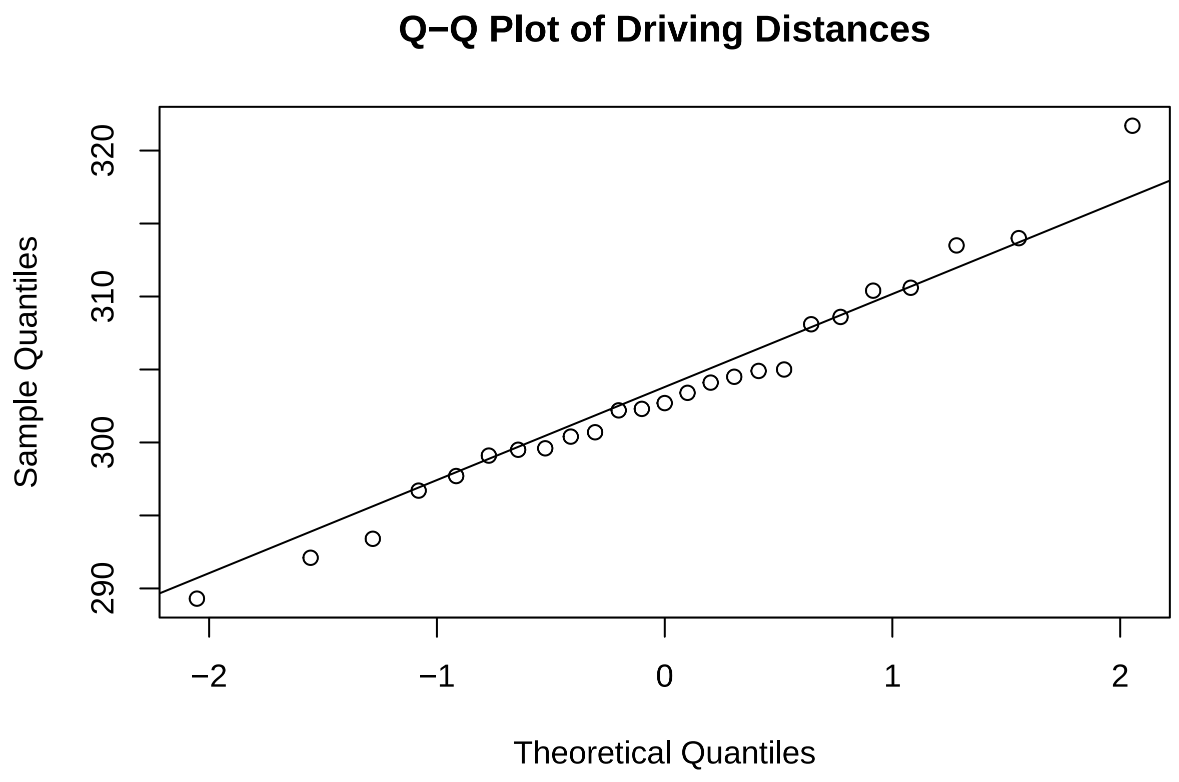
Since we are in the "small sample" case, we need to assume our population distribution is (at least approximately) normal. One way to see if this is reasonable is by using a quantile-quantile (Q-Q) plot, which plots theoretical quantiles on the x-axis and sample quantiles on the y-axis.

If these points fairly closely follow a straight diagonal line, the normality assumption is considered reasonable.

Good news: R has built-in functions for such plots: qqnorm() will create the Q-Q plot, and qqline() will overlay the diagonal line. Use these functions to create the plot and line for your sample. For this problem, don't worry too much about if your plot doesn't seem to follow the line very closely - I just want you to know how to create such a plot for now.

```r
# creating the qq plot
qqnorm(avg_sample,  main = "Q-Q Plot of Driving Distances")
```

```r
# creating the qqline() plot
qqline(avg_sample)
```



Part (c)

Based on your sample, create a 95% (two-sided) confidence interval for the average driving distance for PGA Tour players during the 2022-2023 season. Use the code chunk below for any calculations, and give your interval as well as an interpretation sentence where indicated.

Be sure to be appropriately descriptive/specific/clear with your interpretation (note that driving distance here are measured in yards).

```r
# finding the sample mean
sample_mean <- mean(avg_sample)
print(paste("Sample Mean: ", sample_mean))
```

```
## [1] "Sample Mean:  303.38"
```

```r
# finding the sample standard deviation
sample_sd <- sd(avg_sample)
print(paste("Sample Standard Deviation: ", sample_sd))
```

```
## [1] "Sample Standard Deviation:  7.29491832259507"
```

```r
# finding the number of samples
sample_number <- length(avg_sample)
print(paste("Number of Samples: ", sample_number))
```

```
## [1] "Number of Samples:  25"
```

```r
# finding degrees of freedom
degrees_of_freedom <- sample_number - 1
print(paste("Degree of Freedom: ", degrees_of_freedom))
```

```
## [1] "Degree of Freedom:  24"
```

```r
# calculating the t-value
t_value <- qt(0.975, degrees_of_freedom)
print(paste("t-value: ", t_value))
```

```
## [1] "t-value:  2.06389856162803"
```

```r
# finding the lower and upper bounds for average driving distance
upper_bound = sample_mean + (t_value * (sample_sd / sqrt(sample_number)))
lower_bound = sample_mean - (t_value * (sample_sd / sqrt(sample_number)))

# printing out the upper and lower bounds
sprintf("95%% Confidence Interval: [%.2f, %.2f]", lower_bound, upper_bound)
```

```
## [1] "95% Confidence Interval: [300.37, 306.39]"
```

Answers for part (c): Since the number of samples that we have is less than 40 and that the population standard deviation is unknown, this means we use the t distribution. With regards to the lower and upper bound that we found, we can say that we are 95% confident that the average driving distance for PGA Tour players is between 300.37 and 306.39

Part (d)

If you repeated this process 50 times, how many of those 50 confidence intervals would you expect to contain the true mean? Does the interval you created contain the true mean? Note that we would not usually know this, but here we can check just for fun :) (Type your answers where below as indicated.)

```r
# calculaating number of true means
num_true_mean <- 0.95 * 50
print(paste("Number of True Mean: ", num_true_mean))
```

```
## [1] "Number of True Mean:  47.5"
```

```r
# capturing the true mean
true_mean <- mean(pga_df$AVG)

# checking if the true mean is between the confidence interval
if(true_mean >= lower_bound && true_mean <= upper_bound) {
    print("True Mean is between the Confidence Interval")
} else {
    print("True Mean isn't between the Confidence Interval")
}
```

```
## [1] "True Mean is between the Confidence Interval"
```

Answers for part (d): The number of true mean if 47.5, which means it is either 47 to 48. To check whether the true mean is between the confidence intervals, I used a if statement to check, and it confirmed that the true mean is between the confidence interval.

Part (e)

Based on your sample, create a 95% upper confidence bound for the standard deviation of driving distance for PGA Tour players during the 2022-2023 season. Use the code chunk below for any calculations, and give your interval as well as an interpretation sentence where indicated. Be sure to be appropriately descriptive/specific/clear with your interpretation.

```r
# finding the sample standard deviation
standard_deviation <- sd(avg_sample)

# finding the number of elements
elements <- length(avg_sample)

# finding the degrees of freedom
freedom <- elements - 1

# finding the chi-squared value
chi_sq <- qchisq(0.05, freedom)

# finding the upper confidence bound
chi_upper_bound <- ((elements - 1) * (standard_deviation^2)) / chi_sq
upper_sd_bound <- sqrt(chi_upper_bound)

# the output of the result
print(paste("The Upper Bound: ", upper_sd_bound))
```

```
## [1] "The Upper Bound:  9.60341866359974"
```

Answers for part (e): Since the sample is normally distributed and we are trying to find the upper bound of the standard deviation, it would make sense to use the chi-squared distribution. Given the results, we can say that we are 95% confident that the standard deviation of driving distance for PGA Tour players was less than 9.6

Problem 2

Here you will be creating a 90% confidence interval for a proportion based on data you collected. For each part, type your answers in the appropriate places.

Part (a)

What is the "topic" you chose? In other words, what exactly are you creating a confidence interval for?

Answer for part (a): I am creating a confidence interval for the proportion of sixes that I got after sitting on my chair and rolling a dice 100 times.

Part (b)

Briefly describe how your data was collected, and what the results were. (Be sure to include things like sample size and sample proportion.)

Answer for part (b): The data was collected by rolling a dice and then recording the number that was received from the die itself onto a piece of paper. The results were as follows:

Dice Face	Frequency
1	15
2	17
3	14
4	18
5	19
6	17

The sample size for the number of sixes I got is 17 / 100 which is 0.17

Part (c)

State your confidence interval, along with an appropriately specific/descriptive/clear interpretation sentence. You may use the code chunk below for any calculations (or you may use a calculator and z-table instead if you prefer - just make sure to indicate this if that is your choice).

```r
#Use this code chunk for any calculation for Problem 2

# stating the confidence interval and alpha value
confidence_interval <- 0.95
alpha <- (1 - confidence_interval) /  2
print(paste("Alpha Value: ", alpha))
```

```
## [1] "Alpha Value:  0.025"
```

```r
# stating the sample size
die_sample_size = 100

# the sample probability
sample_probability <- 17 / 100

# caluclating the z value
z_value <- qnorm(1 - alpha)
z_squared <- z_value^2

# finding the denominator
denominator <-  1 + (z_squared / die_sample_size)

# finding value of part one
part_one <- (sample_probability + (z_squared / (2 * die_sample_size))) / denominator

# finding the value of part two
part_two <- ((((sample_probability) * (1 - sample_probability)) / die_sample_size) +
(z_squared / (4 * die_sample_size^2))) / denominator

# finding the upper and lower bounds
die_upper_bound <- part_one + (z_value * part_two)
die_lower_bound <-  part_one - (z_value * part_two)

# printing out the upper and lower bounds
sprintf("95%% Confidence Interval: [%.2f, %.2f]", die_lower_bound, die_upper_bound)
```

```
## [1] "95% Confidence Interval: [0.18, 0.19]"
```

Answer for part (c): Our confidence interval is from 0.18 to 0.19. Based on the rolls that have been recorded, what I can say is that I am 95% confident that the true proportion of getting a 6 after rolling a die is between 0.18 to 0.19.

Problem 3:

According to the city of Tulsa's municipal website, the rate it charges per 5 CCF of residential was $22.62 in 2020. We are going to investigate how the residential water rates of other US public utilities compare to Tulsa's rate.

To do this, we will conduct a hypothesis test, with significance level 0.05 to try to determine whether the population average rate of 5 CCF of residential water in the US in 2020 was less than the rate charged by Tulsa.

The file "Activity5Water.csv"' (found in Canvas) contains the rate per 5 CCF of residential water for 42 US cities. If you make sure that the file is in the same folder as this file, you should be able to use it with no problems as long as you run the following chunk of code.

I've just called it "water" for ease of use, but you can of course change that if you want to.

```
# reading in a data frame
water_df <- read.csv("Activity5Water.csv")
```

Include any calculations used in parts (a)-(e) in the code chunk below, and give an answer/response for each of the parts in the appropriate space.

```
#Use this chunk of code for any calculations in parts (a)-(e)



# Question 3B

rate_values <- water_df$Rate
print(paste("The Rate Values: ", rate_values))
```

```
##  [1] "The Rate Values:  10.48" "The Rate Values:  9.18"
##  [3] "The Rate Values:  11.8"  "The Rate Values:  6.5"
##  [5] "The Rate Values:  12.42" "The Rate Values:  14.53"
##  [7] "The Rate Values:  15.56" "The Rate Values:  10.12"
##  [9] "The Rate Values:  14.5"  "The Rate Values:  16.18"
## [11] "The Rate Values:  17.6"  "The Rate Values:  19.18"
## [13] "The Rate Values:  17.98" "The Rate Values:  12.85"
## [15] "The Rate Values:  16.8"  "The Rate Values:  17.35"
## [17] "The Rate Values:  15.64" "The Rate Values:  14.8"
## [19] "The Rate Values:  18.91" "The Rate Values:  17.99"
## [21] "The Rate Values:  14.9"  "The Rate Values:  18.42"
## [23] "The Rate Values:  16.05" "The Rate Values:  26.85"
## [25] "The Rate Values:  22.32" "The Rate Values:  22.76"
## [27] "The Rate Values:  20.98" "The Rate Values:  23.45"
## [29] "The Rate Values:  19.05" "The Rate Values:  23.7"
## [31] "The Rate Values:  19.26" "The Rate Values:  23.75"
## [33] "The Rate Values:  27.8"  "The Rate Values:  27.05"
## [35] "The Rate Values:  27.14" "The Rate Values:  26.99"
## [37] "The Rate Values:  24.68" "The Rate Values:  37.86"
## [39] "The Rate Values:  26.51" "The Rate Values:  39.01"
## [41] "The Rate Values:  29.46" "The Rate Values:  41.65"
```

```
# finding the mean
mean_rate <- mean(rate_values)
standard_deviation_rate <- sd(rate_values)
number_rate <- length(rate_values)

# finding the z-value
z_value <- (mean_rate - 22.62) / (standard_deviation_rate / sqrt(number_rate))
print(paste("The z-value: ", z_value))
```

```
## [1] "The z-value:  -1.97906935349608"
```

```
# Question 3C
```

```r
# finding the p-value
p_value <- pnorm(z_value)
print(paste("P-value:", p_value))
```

```
## [1] "P-value: 0.0239040991819248"
```

Part (a)

First, determine the null and alternative hypotheses.

Answer for part (a): - Null Hypothesis: $H_0 : \mu = 22.62$ - Alternative Hypothesis: $H_a : \mu < 22.62$

Part (b)

Next, calculate our test statistic value.

Answer for part (b): The test statistic value is -1.97906935349608

Part (c)

Now use the value from part (b) to find the P-value.

Answer for part (c): The p-value is 0.0239040991819248

Part (d)

Is the null hypothesis rejected at our chosen significance level?

Answer for part (d): As the signifigance level is 0.05, since our p-value is 0.02, making it less than the signifigance leve, we choose to reject the null hypothesis

Part (e)

Give an interpretation of the result in the context of this problem.

Answer for (e): In the context of this problem, what this means is that there is enough statistical evidence to show that the population average rate of 5 CCF of residential water in the US in 2020 is less than the rate charged by Tulsa.

Part (f)

In practice, even with a large sample size, a t-test is actually preferred (instead of a z-test) if the tools to do so are available and the normality assumption is reasonable. In the R Console, type in "?t.test" and hit enter to view the documentation for the t.test() function. Then, in the code chunk below, use this function to "redo" the hypothesis test we just did (except now we are using the t-distribution) with just a single line of code. Make sure you specify the appropriate arguments in the t.test() function.

```r
#This code just is just for using the t.test() function for part (f)
result <- t.test(water_df$Rate, mu = 22.62, alternative = "less")

# printing out the results
print(result)
```

```
##
##  One Sample t-test
##
## data:  water_df$Rate
## t = -1.9791, df = 41, p-value = 0.02728
## alternative hypothesis: true mean is less than 22.62
## 95 percent confidence interval:
##      -Inf 22.26356
## sample estimates:
## mean of x
##  20.23833
```