Assignment 1

Please complete the following assignment in R. To turn this in, please submit either your R file, or if you feel comfortable trying Quarto, a file created using Quarto (see the first section). We will go over Quarto in class for files for future assignments.

# 1. Extra: Quarto

Quarto is a comprehensive and versatile document creation package designed to facilitate dynamic and reproducible research, reports, presentations, and more. Built with the needs of data scientists, researchers, and authors in mind, Quarto supports a wide array of programming languages including R, Python, Julia, and Observable JavaScript, allowing for seamless integration of code, results, and narratives in a single document. This open-source tool extends the capabilities of Markdown, enabling users to create high-quality outputs in various formats such as HTML, PDF, slides, and even interactive websites or applications, all from a single source. With its emphasis on reproducibility, collaboration, and flexibility, Quarto aims to enhance the way knowledge is shared and communicated in the digital age. To learn more about Quarto see <https://quarto.org>.

## 1.1 Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

1 + 1

[1] 2

You can add options to executable code like this

[1] 4

The echo: false option disables the printing of code (only output is displayed).

# 2. A First Look at R

R is a prominent programming language tailored for statistical computing and graphics, celebrated for its open-source nature and a vast library of packages. Originating from the S language, R facilitates advanced data manipulation, analysis, and visualization, making it a preferred tool among statisticians, data scientists, and researchers. Its extensive community support and comprehensive tools for various statistical techniques have solidified R’s position as a fundamental resource in data-driven fields.

## 1.2 Exercises

1. Practice using R by writing code to do the following:
   1. Store the sum of 5 and 6 in a variable named bob, then print the continence of bob
   2. Store the product of 25 and 5 in a variable named bill, then print the continence of bill
2. Use the logical operators to instruct R to complete the following true false test:

| Operator | Description | Example | Result |
| --- | --- | --- | --- |
| == | Equal to | 2 == 2 | TRUE |
| != | Not equal to | 2 != 2 | FALSE |
| < | Less than | 2 < 3 | TRUE |
| <= | Less than or equal to | 3 <= 3 | TRUE |
| > | Greater than | 3 > 2 | TRUE |
| >= | Greater than or equal to | 2 >= 3 | FALSE |
| & | Element-wise logical AND | TRUE & FALSE | FALSE |
| \| | Element-wise logical OR | TRUE \| FALSE | TRUE |

* 1. **Direct Equality Test**: Verify if 100 divided by 2 equals 50.
  2. **Inequality Check**: Determine if 7 times 3 is not equal to 20.
  3. **Simple Less Than Comparison**: Check if 15 minus 10 is less than 8.
  4. **Less Than or Equal Verification**: Assess if half of 100 is less than or equal to 50.
  5. **Greater Than or Equal to Confirmation**: Confirm if 200 minus 150 is greater than or equal to 50.
  6. **Inequality Test**: Check if the the string “data” is not equal to the length of “science”.
  7. **Logical AND Operation**: Determine if 20 is greater than 10 and 30 is less than 40.
  8. **Logical OR Operation**: Check if 5 is less than 3 or 8 is greater than 6.
  9. **Logical AND Test**: Test if 10 is greater than 5 and 8 is less than 15.
  10. **Logical OR Test**: Determine if either 20 is less than 15 or 25 is greater than 20.

## 3. Loading and Manipulating Data in R

Before running any statistics, you need to import a dataset to use and format it to best fit the statistic you are going to run.

## 2.1 Loading Data

To import a dataset from a given URL into R, you can use the read.csv() function, which is designed to read a file in CSV (Comma Separated Values) format and convert it into a data frame in R. In this case, the dataset is located at "https://ljkelly3141.github.io/mont820/data/sleep.csv". To perform the import, you would assign the result of the read.csv() function to a variable, in this case, named sleep. The complete command would be

sleep <- read.csv("https://ljkelly3141.github.io/mont820/data/sleep.csv")

This line of code tells R to fetch the CSV file from the specified URL and load its contents into the variable sleep, effectively storing the dataset in your R environment as a data frame. This process allows you to then work with the data using various R functions, starting with basic data inspection commands like head(sleep), which displays the first few rows of the dataset to give you a quick glimpse of its structure and contents.

## 2.2 Packages

In R, a package is a collection of functions, data, and compiled code in a well-defined format. Packages extend the functionality of R by providing additional routines for specific tasks, such as statistical techniques, graphical devices, import/export capabilities, and more. The Comprehensive R Archive Network (CRAN) hosts thousands of packages covering a wide range of disciplines. Users can also access packages from GitHub and other repositories.

The tidyverse is a popular collection of R packages designed for data science that share an underlying design philosophy, grammar, and data structures. It includes packages like ggplot2 for data visualization, dplyr for data manipulation, and readr for data import, among others.

To demonstrate installing and loading the tidyverse package in R, you can follow these steps:

1. **Installing the tidyverse package:**

install.packages("tidyverse")

This command downloads and installs the tidyverse package and its dependencies from CRAN.

1. **Loading the tidyverse package:**

library(tidyverse)

This command loads the tidyverse package into the R session, making its functions available for use. Upon loading, it typically prints a message indicating which of the core tidyverse packages are being attached to the session.

## 2.3 Summarizing and Shaping Data

To demonstrate data analysis and manipulation in R, we can utilize the describe function from the psych package and the melt function from the reshape2 to work with the sleep dataset from the datasets package. The sleep dataset contains data on the effect of two drugs on 10 patients’ sleep improvements.

First, we’ll summarize the sleep dataset using the describe function, which provides a detailed statistical summary including measures like mean, standard deviation, and range.

describe(sleep)

Then, we’ll reshape the data using melt to convert it from wide to long format, making it easier for certain types of analysis.

sleep\_long <- melt(sleep,

id.vars = “X”,

measure.vars = c(“Group1”, “Group2”),

variable.name = “group”,

value.name = “change.in.sleep”)

The melt() function is used to reshape the dataset, where id.vars = “X” specifies the identification variables that remain the same and will not be lengthened, measure.vars = c(“Group1”, “Group2”) specifies the columns to be lengthened, variable.name = "group" defines the new column that will hold the name of the original columns, and value.name = "change.in.sleep" defines the new column that will contain the values from the lengthened columns. This transformation is particularly useful for tidying the data and preparing it for analyses that require a long format, such as a t-test.

# 3. Difference of means test

The t-test is a statistical test used to compare the means of two groups to determine if there is a significant difference between them. It’s based on the t-distribution, a type of probability distribution that arises when estimating the mean of a normally distributed population in situations where the sample size is small and the population standard deviation is unknown.

In the context of the t.test function in R, the test can be applied in various scenarios, including comparing the means of two independent groups or the mean of a single group against a known value. The syntax

t.test(change.in.sleep ~ group, data = sleep\_long, alternative = "two.sided")

specifies a formula-based interface where change.in.sleep ~ group indicates that the mean difference in sleep change is being compared across different groups defined in the sleep dataset.

* change.in.sleep: The variable representing the amount of change in sleep duration, which is the numeric variable being tested.
* group: The grouping variable, which categorizes the data into different groups for comparison.
* data = sleep: Specifies the dataset containing the variables.
* alternative = "two.sided": Indicates that the test is two-sided, meaning the hypothesis is testing for any difference in means, regardless of direction. The alternative could also be “greater” or “less” for one-sided tests, depending on the hypothesis.

This particular use of t.test aims to determine whether there is a statistically significant difference in the mean change in sleep duration between the two groups defined in the sleep dataset. The result includes the t-statistic value, degrees of freedom, p-value, and confidence interval for the mean difference, providing insights into whether the observed difference in means could be due to chance.

## 3.1 Exercises

1. In this exercise, you are provided with a dataset on sleep patterns, available at the following URL: **"https://ljkelly3141.github.io/mont820/data/sleep.csv"**. Your task is to write a code chunk in R that accomplishes the following objectives:
   1. **Data Import**: Use the **read.csv()** function to import the dataset into R. Assign the imported data to a variable named **sleep**.
   2. **Data Inspection**: Utilize the **head()** function to display the first few rows of the dataset. This step is crucial for an initial inspection of the data structure and to verify the successful import.
2. In this exercise, you will write code to preform the following actions
   1. **Install Packages**: Utilize the install.packages() function to install the reshape2 and psych packages.
   2. **Load Packages**: Utilize the library() function to load the the tidyverse and psych packages.
3. Utilize the describe function from the psych package to provide a statistical summary of the sleep dataset. Following that, employ the melt function from the reshape2 package to reformat the dataset into a long structure. Document your process and findings, explaining the statistical summary provided by the describe function and how the data’s structure changed after pivoting.
4. Formulate a hypothesis and conduct a t-test to evaluate if the average change in sleep duration for Group 1 is significantly less than that for Group 2 using the sleep\_long dataset. Write a short paragraph in APA style to present your findings, including the t-test statistics and your interpretation of the results, describing if there is a statistically significant difference in sleep duration changes between the two groups.