

Homework 1

Please complete the following assignment in R. To turn this in, please submit your R file on Canvas.

1. 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.

1.1 Exercises

1. Practice using R by writing code to do the following:
 - a. Store the sum of 5 and 6 in a variable named `bob`, then print the continence of `bob`
 - b. 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
<code>==</code>	Equal to	<code>2 == 2</code>	<code>TRUE</code>
<code>!=</code>	Not equal to	<code>2 != 2</code>	<code>FALSE</code>
<code><</code>	Less than	<code>2 < 3</code>	<code>TRUE</code>
<code><=</code>	Less than or equal to	<code>3 <= 3</code>	<code>TRUE</code>
<code>></code>	Greater than	<code>3 > 2</code>	<code>TRUE</code>
<code>>=</code>	Greater than or equal to	<code>2 >= 3</code>	<code>FALSE</code>
<code>&</code>	Element-wise logical AND	<code>TRUE & FALSE</code>	<code>FALSE</code>
<code>\ </code>	Element-wise logical OR	<code>TRUE \ FALSE</code>	<code>TRUE</code>

Operator	Description	Example	Result
----------	-------------	---------	--------

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

2. Loading and Manipulating Data in R

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

2.1 Loading a Dataset

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.

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

1. Installing the **psych** package:

```
install.packages("psych")
```

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

2. Loading the **psych** package:

```
library(psych)
```

This command loads the **psych** package into the R session, making its functions available for use. Upon loading, it typically prints a message indicating which of the core **psych** 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** package to work with the **sleep** dataset. The **sleep** dataset contains data on the effect of two drugs on 10 patients' sleep improvements.

First, we will 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 will 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

The two sample 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 is 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:
 - a. **Data Import:** Use the `read.csv()` function to import the dataset into R. Assign the imported data to a variable named `sleep`.
 - b. **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
 - a. **Install Packages:** Utilize the `install.packages()` function to install the `reshape2` and `psych` packages.
 - b. **Load Packages:** Utilize the `library()` function to load the the `reshape2` and `psych` packages.
3. Use 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, naming the new dataframe as `sleep_long`. 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 hypothesis, 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.