Topic 1: Introduction to R language

In this topic in week 1, you will learn about :

- basic features of R programming
- Logical vectors and relational operators

Basic Features

EXERCISES

Introduction to basic features of R programming

Basic Features of R Programming

R is a powerful and popular programming language widely used for data analysis, statistical computing, and graphical representation. It has several essential features that make it a favorite choice among data scientists, statisticians, and researchers. Below are some of the key features of R programming:

1. **Open Source**: R is an open-source language, which means it is freely available for anyone to use, modify, and distribute. The open-source nature encourages collaboration and continuous improvement of the language.
2. **Data Handling**: R provides efficient data handling capabilities, making it easy to import, manipulate, and transform data. It supports various data structures, such as vectors, matrices, data frames, and lists.
3. **Statistical Analysis**: R is primarily known for its extensive statistical capabilities. It offers a vast collection of built-in functions and packages for conducting various statistical analyses, including regression, hypothesis testing, ANOVA, and time series analysis.
4. **Graphics and Visualization**: R excels in producing high-quality visualizations and graphs. It offers powerful plotting functions and packages like ggplot2, lattice, and base graphics, enabling users to create insightful and publication-quality plots.
5. **Extensible**: R is an extensible language, allowing users to create their functions and packages to extend its functionality. The Comprehensive R Archive Network (CRAN) hosts thousands of user-contributed packages for various purposes.
6. **Reproducibility**: R promotes reproducibility in data analysis and research. Scripts written in R can be easily shared and rerun on different datasets, ensuring that others can replicate and verify the results. Documentation and Community Support: R has comprehensive documentation available through manuals, guides, and tutorials. Additionally, it has a vibrant and active community of users and developers who provide support through forums and mailing lists.
7. **Interactive Environment**: R offers an interactive environment with an interactive console and script editor, allowing users to execute commands interactively and see immediate results.
8. **Integration with Other Tools**: R can be integrated with other programming languages like Python, C++, and Java, enabling users to leverage their existing code and tools seamlessly.

Summary: R programming offers a wide range of features for data analysis, statistics, visualization, and reproducibility. Its open-source nature, extensive statistical libraries, and active community support make it a preferred choice for data analysis and research in various domains. Whether you are a beginner or an experienced data analyst, R provides a powerful and flexible environment to explore, analyze, and visualize data effectively.

Step-by-Step Guide to Install and Use R

Installing and using R is a straightforward process. Here's a step-by-step guide to help you get started:

1. **Download R**

Go to the R Project website: **https://www.r-project.org/** Click on "CRAN" (Comprehensive R Archive Network) under "Download and Install R." Select a CRAN mirror location near you.

Choose the appropriate operating system (Windows, macOS, or Linux). Download the R installer for your operating system.

2. **Install R**

For **Windows**: Double-click the downloaded installer file and follow the on-screen instructions to install R.

For **macOS**: Double-click the downloaded package file, and the installation process will begin.

3. **Launch R**

For **Windows**: You can find the R program in the Start Menu or on your desktop. Double-click the R icon to launch the R console.

For **macOS**: You can find the R application in the Applications folder. Double-click the R icon to launch the R console.

4. **Using R Console**

After launching R, you will see the R console where you can type commands and interact with R. To execute a command, type it in the console and press Enter.

For example, try typing **'print("Hello, R!")'** and press Enter. R will print the message "Hello, R!".

5. **Installing R Packages**

R has a vast collection of packages contributed by the community. To install packages, you can use the **'install.packages() function'**.

For example, to install the **'ggplot2'** package for data visualization, type **'install.packages("ggplot2")'** and press Enter.

6. **Loading and Using Packages**

Once a package is installed, you need to load it into your R session before using its functions. To load a package, use the **library( )** or **require( )** function.

For example, to load the ggplot2 package, type **library(ggplot2)** and press Enter.

7. **Using R Scripts**

Writing code directly in the console is suitable for small tasks, but for more complex analyses or projects, it's best to use R scripts.

Create a new plain text file with a **.R** extension, e.g., **myscript.R**. Write your R code in the script file using a text editor or an integrated development environment (IDE).

Save the file and then open it in the R console using the **source( )** function. For example, **source("myscript.R")**.

8. **Interacting with Plots**

When you create plots in R, they will typically appear in a separate window or plot pane. You can interact with the plots using buttons or options in the plot pane (for GUI-based interfaces) or by typing additional commands to modify or save the plots.

Step 9: Exiting R

To exit R, type **q( )** or **quit( )** in the R console and press Enter. Congratulations! You have successfully installed and started using R. Now, you can explore R's vast capabilities for data analysis, visualization, and statistical modeling to suit your needs.

Operators in R

Operators in R

Type of operators in R

- Assignment operators
- Arithmetic operators
- Relational operators
- Logical operators

1. ASSIGNMENT OPERATORS

The assignment operators in R allows you to assign data to a named object in order to store the data.

```
##    Operator                                          Description
## 1:       <-                                 Leftwards assignment
## 2:        = Left assignment (not recommended) and argument assignment
## 3:      <<-        Left lexicographic assignment (for advanced users)
## 4:       ->                                Rightwards assignment
## 5:      ->>       Right lexicographic assignment (for advanced users)
```

Example

```
x <- 2
y = 6
3 -> z
w <<- 7
8 ->> s

x
```

```
## [1] 2
```

```
y
```

```
## [1] 6
```

```
z
```

```
## [1] 3
```

```
w
```

```
## [1] 7
```

```
s
```

```
## [1] 8
```

2. ARITHMETIC OPERATORS

These operators are used to carry out mathematical operations like addition and multiplication. Here is a list of arithmetic operators available in R.

```
##   Operator                   Description
## 1        +                      Addition
## 2        -                   Subtraction
## 3        *                Multiplication
## 4        /                      Division
## 5        ^                      Exponent
## 6       %% Modulus (Remainder from division)
## 7      %/%              Integer Division
```

Example

```
3 + 5
```

```
## [1] 8
```

```
8 - 3
```

```
## [1] 5
```

```
7 * 5
```

```
## [1] 35
```

```
1/2
```

```
## [1] 0.5
```

```
4 ^ 4
```

```
## [1] 256
```

```
4 ** 4
```

```
## [1] 256
```

```
5 %% 3
```

```
## [1] 2
```

```
5 %/% 3
```

```
## [1] 1
```

3. RELATIONAL/COMPARISON OPERATORS

Comparison or relational operators are designed to compare objects and the output of these comparisons are of type boolean. To clarify, the following table summarizes the R relational operators.

```
##    Operator             Description
## 1:        <               Less than
## 2:        >            Greater than
## 3:       <=     Less than or equal to
## 4:       >= Greater than or equal to
## 5:       ==                Equal to
## 6:       !=            Not equal to
```

Example

```
x <- 3
y <- 15

x<y
```

```
## [1] TRUE
```

```
x>y
```

```
## [1] FALSE
```

```
x<=5
```

```
## [1] TRUE
```

```
y>=20
```

```
## [1] FALSE
```

```
y == 16
```

```
## [1] FALSE
```

```
x != 5
```

```
## [1] TRUE
```

4. LOGICAL/BOOLEAN OPERATORS

```
##    Operator              Description
## 1:        !              Logical NOT
## 2:        & Element-wise logical AND
## 3:       &&              Logical AND
## 4:        |  Element-wise logical OR
## 5:       ||               Logical OR
```

Example

```
x<-c(TRUE, FALSE, 0,6)
x
```

```
## [1] 1 0 0 6
```

```
x<-c(TRUE, FALSE, 0,6)
y <- c(FALSE,TRUE,FALSE,TRUE)

!x
```

```
## [1] FALSE  TRUE  TRUE FALSE
```

Operators & and | perform element-wise operation producing result having length of the longer operand.

```
x<-c(TRUE, FALSE, 0,6)
y <- c(FALSE,TRUE,FALSE,TRUE)

x&y
```

```
## [1] FALSE FALSE FALSE  TRUE
```

```
x<-c(TRUE, FALSE, 0,6)
y <- c(FALSE,TRUE,FALSE,TRUE)

x|y
```

```
## [1]  TRUE  TRUE FALSE  TRUE
```

But && and || examines only the first element of the operands resulting into a single length logical vector.

x<-c(TRUE, FALSE, 0,6) y <- c(FALSE,TRUE,FALSE,TRUE)

x&&y

Zero is considered FALSE and non-zero numbers are taken as TRUE.

x<-c(TRUE, FALSE, 0,6) y <- c(FALSE,TRUE,FALSE,TRUE)

x||y

Let's summarise the main points of this tutorial.

- The three most important logical operators are NOT, AND and OR.
- In R, the NOT operator is the exclamation mark.
- The AND operator is the ampersand.
- The OR operator is the vertical bar.
- Logical operators are often used to subset vectors or data frames.