Introduction to R

Saeed Saffari Saeed.Saffari@dal.ca

Winter 2025

Contents

1 Introduction

This is an R Markdown Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter* (on Mac) or *Ctrl+Shift+Enter* (on Windows).

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I* (on Mac) or *Ctrl+Alt+I* (on Windows).

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

You can download and install packages with `install.packages("The name of package")`.

1.1 Markdown:

2 Basic of learning

In this part we talk about basic elements of R programming.

2.1 How to Print

2.2 Data Types (Classes)

In R, data types (or classes) define the kind of data stored in variables. Here are the most common data types in R:

Class	**Description**	**Example**	**Code Example**
`numeric`	Represents decimal or whole numbers	3.14, 42, -7.8	`x <- 3.14; class(x)`
`character`	Represents text strings	"Hello", "R Programming"	`z <- "Hello"; class(z)`
`logical`	Represents Boolean values (TRUE or FALSE)	TRUE, FALSE	`is_valid <- TRUE; class(is_valid)`
`complex`	Represents complex numbers	2+3i, 1-4i	`c <- 2 + 3i; class(c)`
`list`	Represents a collection of different types	A list of numbers, text	`lst <- list(1, "Hello", TRUE); class(lst)`

2.3 Arithmetic Operators

Symbol	Task Performed
+	Addition
-	Subtraction
/	division
*	multiplication
**	to the power of
^	to the power of
%%	modulus
%/%	floor division

We can save values in variables:

In R programming, code runs line by line, with only the last assignment determining the final value of a variable.

2.3.1 Practice:

Convert a given temperature X degrees Celsius to Fahrenheit.

$$F = C \cdot \frac{9}{5} + 32$$

2.4 Relational Operators

Symbol	Task Performed
<-	Assignment
=	Assignment
assign()	Assignment

Symbol	Task Performed
==	True, if it is equal
!=	True, if not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to

you can use below command to get special values:

you can write comment with # :

you can creat sequence numbers with below command:

This work like *arange* in numpy pakage in Python

This work like *linspace* in numpy pakage in Python

Replicate function:

2.5 Practice:

Simulate GDP growth from the year 2000 to 2025 with an annual growth rate of 3% starting from 1000 units.

$$GDP_t = 1000 \cdot (1 + r)^n$$

2.6 Loops

2.6.1 if , elif

2.6.2 for loops

2.6.3 while loops

Use break and next in loops

2.6.4 Practice:

Write a conditional statement to check whether the variable is positive, negative, or zero and print the appropriate message.

2.7 function

2.7.1 Practice I:

Write a function `temp()` that converts a temperature in Celsius to Fahrenheit.

2.7.2 Practice II:

Create a function `calculate_mean()` that takes a vector of numbers and returns their mean. Test the function with the vector `c(1, 2, 3, 4, 5)`.

3 Vetors

The most common way to creat vectors is to use function `c()`.

Join vectors

You can find *lenght* of vectors with *lenght()* function:

3.1 Vector Indexing

Use for loops for access to elemets of vectors

Use index:

3.2 Matching Operator

3.3 Vector Arithmetic's

3.4 Vector Methods

3.5 Logical Vector

3.6 Factors

- Used to represent categorical data
- Treated as integer vector, having a label
- Factors are self describing

```
x <- c('Male', "Female", "Male", 'Male', "Female")
```

3.7 Mathematical Function in R

3.8 Random Number in R

3.9 Practice:

Given a list of students `("Alice", "Bob", "Charlie", "David", "Eve")` and their corresponding scores `(85, 92, 78, 55, 88)`, extract the names and scores of students who passed (`score >= 60`). Also, calculate the mean score of the students who passed.

```
students <- c("Alice", "Bob", "Charlie", "David", "Eve")
scores <- c(85, 92, 78, 55, 88)
```

4 Matrix

4.1 Creat Matrix

Matrix are 2-dimentsional vectors and Dimensianal attribute is of lenght 2 (rows and columns). We should to know that Matrix contain elemts of same type.

4.2 Matrix diag

like `numpy.full` in python

like `numpy.diag` in python

for find the elements of diagonal of matrix:

4.3 Matrix: Naming Rows & Columns

4.4 Matrix Indexing

Indexing in R programming is similar to Python.

You can change values in matrix.

4.5 Matrix: `rbine()` and `cbind()` functions

You can combine matrices with `rbine()` and `cbind()` functions.

At first, we want to combine the matrices from the row.

After that, we want to combine the matrices from the columns.

Relational Operators in Matrics:

```
A <- matrix(c(1,2,3,4,5,6,8,9,1) , nrow=3, ncol=3, byrow=TRUE)
B <- matrix(c(3,1,2,4,2,1,5,1,2), nrow=3, ncol=3, byrow=TRUE)
```

Like `numpy.transpose()` or `.T` in python

```
A <- matrix(c(1,2,3,4,5,6,8,9,1,4,2,3) , nrow=3, ncol=4, byrow=TRUE)
```

4.6 Matrix Specific Functions

4.7 Practice I

Create a 4x4 matrix of random integers between 1 and 100.

- Print the matrix.
- Calculate the row-wise sum and column-wise mean.
- Check if the matrix is symmetric by comparing it to its transpose.

```
set.seed(123)
mat <- matrix(sample(1:100, 16), nrow = 4)
```

4.8 Practice II

Given a 3x3 matrix A of integers and a vector b = (30, 20, 15), solve the linear equation $A \cdot X = b$.

```
set.seed(123)
A <- matrix(sample(1:10, 9, replace = TRUE), nrow = 3)
b <- c(30, 20, 15)
```

5 Lists

5.1 Creat list

`Lists` are also collecting of data and another kind of data storage. Lists can contain elemnts of any type of R object and these elements of list don't need be same type. You can creat list by using `list()` function.

Creat list with vectors

5.2 List subset Operator

5.3 Lists Concatenation