R Programming Notes

Introduction to R

What is R?

- **R is an open-source programming language designed for statistical analysis, data visualization, and data manipulation.**
- **It is both a language and an environment.**
- **Originally developed by Ross Ihaka and Robert Gentleman in the 1990s.**
- **It's widely used in statistics, data science, and machine learning.**

Why Learn R?

- **Specialized for data analysis & visualization**
- **Comes with powerful libraries/packages (like ggplot2, dplyr, tidyverse)**
- **Strong community support**
- **Used in academia, research, and industry**

R Environment

When you open RStudio, you'll see 4 main panes:

1. **Source Editor: for writing and saving R scripts (.R files)**
2. **Console: for running code line by line**
3. **Environment/History Pane: stores variables created**
4. **Plots/Files/Packages/Help Pane: for visual output, files, and documentation**

Getting Help

```
?mean      # Help for mean function
help(plot) # Another way to get help
example(sum) # Shows examples of sum()
```

Arithmetic Operators

Operator	Description	Example	Output
`+`	Addition	`5 + 3`	`8`
`-`	Subtraction	`10 - 4`	`6`
`*`	Multiplication	`2 * 3`	`6`
`/`	Division	`10 / 5`	`2`
`^` or `**`	Power	`2^3`	`8`
`%%`	Modulus	`10 %% 3`	`1`
`%/%`	Integer Division	`10 %/% 3`	`3`

Relational Operators

5 > 3 # TRUE

5 == 5 # TRUE

5 != 2 # TRUE

4 <= 3 # FALSE

Logical Operators

TRUE & FALSE # AND

TRUE | FALSE # OR

!TRUE # NOT

Assignment Operator

x <- 10 # Preferred

y = 20

10 -> z

🔹 Common Data Types

Type	Example	Description
Numeric	`x <- 10.5`	Decimal values
Integer	`x <- 5L`	Whole numbers
Character	`x <- "R Programming"`	Text data
Logical	`x <- TRUE`	Boolean
Complex	`x <- 3 + 2i`	Complex numbers
Raw	`x <- charToRaw("A")`	Binary data

Type Conversion

as.numeric("12") # 12

as.character(25.5) # "25.5"

as.logical(1) # TRUE

Creating Vectors

v1 <- c(1, 2, 3, 4, 5) # Numeric

v2 <- c("A", "B", "C") # Character

v3 <- c(TRUE, FALSE, TRUE) # Logical

v4 <- seq(1, 10, by = 2) # Sequence

v5 <- rep(3, times = 5) # Repeat

#indexing

v1[1] # First element

v1[2:4] # 2nd to 4th elements

v1[-3] # All except 3rd

v1[c(1,5)] # 1st and 5th

Understanding rm() in R

- **Purpose:**

rm() stands for **remove**.
It's used to **delete objects (variables, functions, data frames, etc.)** from the current **R environment (workspace)**.

When you run R (or RStudio), all objects you create are stored in memory — the workspace.
Sometimes, we want to clean it up — delete old variables, or clear everything to start fresh.

x <- 10

y <- 20

rm(x) #remove the single object

rm(x, y, z) #remove the multiple object

#Remove all objects starting with "temp"

rm(list = ls(pattern = "^temp"))

Remove all variables containing "data"

rm(list = ls(pattern = "data"))

What does cat() do?

cat() stands for **concatenate and print**.

It **displays output** directly on the console (not as a return value).
It joins multiple items together and prints them as text.

```
cat("Hello", "World")
```

What is \014?

\014 is a **special character** — an ASCII **form feed** character.

In older systems, this was used to tell the printer to move to a new page.
In RStudio and R console, it tells the system:

"Clear the console screen."

```
cat("\014")
```

In R:
<- is the **preferred assignment operator** (for assigning values to objects).
= can also assign values, but it's mainly used for **specifying arguments in functions**.

Operator	Main Use	Creates object in workspace?	Common in functions?
<-	Assignment	✅ Yes	Rarely used
=	Function argument specification	❌ No	✅ Yes

Expression	Meaning	Creates object x?
x <- 10	Assign 10 to x	✅
x = 10	Assign 10 to x	✅
print(x = 10)	Pass 10 as argument to print()	❌
print(x <- 10)	Assign 10 to x, then print it	✅

Operator	Description	Used for	Creates variable?
<-	Assignment operator (preferred)	Assign values	✅
->	Rightward assignment	Assign values	✅
=	Argument assignment in functions	Pass parameters	❌
==	Comparison operator	Check equality	❌

Built in vector functions

Function	Description	Example
length(x)	Number of elements	length(x)
sum(x)	Sum of elements	sum(x)
mean(x)	Mean of elements	mean(x)
max(x)	Maximum value	max(x)
min(x)	Minimum value	min(x)
sort(x)	Sort elements	sort(x, decreasing=TRUE)
rev(x)	Reverse elements	rev(x)
unique(x)	Remove duplicates	unique(x)

Rules & Restrictions for Naming Objects in R

Basic Rule of Object Naming

An **object name**:

- Must **start with a letter** (A–Z or a–z) or a **dot (.)** followed by a letter.
- Can contain **letters, digits (0–9), dots (.)**, and **underscores (_)**.
- **Cannot start with a number.**
- **Cannot contain spaces or special symbols** like @, $, %, #, !, -, etc.
- **Cannot be a reserved keyword** in R (e.g., if, else, TRUE, FALSE, for, etc.).

Example	Valid / Invalid	Explanation
x	✅ Valid	Simple name
x1	✅ Valid	Letter followed by number
.x	✅ Valid	Starts with dot + letter
x_value	✅ Valid	Underscore allowed
x.value	✅ Valid	Dot allowed
1x	❌ Invalid	Cannot start with number
_x	❌ Invalid	Cannot start with underscore
.1x	❌ Invalid	Dot cannot be followed by number
x-value	❌ Invalid	- is not allowed (it's the minus operator)
my name	❌ Invalid	Spaces not allowed
TRUE	❌ Invalid	Reserved constant
if	❌ Invalid	Reserved keyword

Style	Example	Recommendation
snake_case	student_marks	✅ Recommended for readability
camelCase	studentMarks	✅ Common in programming
dot.separated	student.marks	✅ Used in base R packages
single letter	x, y, z	⚠️ Fine for small examples, avoid in big projects
UPPERCASE	TOTAL_SUM	⚠️ Often used for constants

Escape Sequence	Meaning	Used For
\n	**New Line**	Moves the cursor to a new line
\t	**Tab Space**	Adds a horizontal tab (like pressing the TAB key)

Escape Sequence	Meaning	Example	Output
\n	New line	cat("Hi\nBye")	Hi Bye
\t	Tab	cat("A\tB\tC")	A B C
\\	Backslash	cat("C:\\Program Files")	C:\Program Files
\"	Double quote	cat("He said \"Hi\"")	He said "Hi"
\'	Single quote	cat('It\'s R language')	It's R language