Data Science with R

Lesson 2— Introduction to R Programming

Learning Objectives

- Outline the importance of R and its significance in data analytics
- Explain the data types and variables in R
- Discuss the types of R operators
- Describe the different types of conditional statements in R
- Examine the different types of loops in R
- Analyze the methods to run an R script
- List the commonly used R functions

Introduction to R Programming

Topic 1— Importance of R

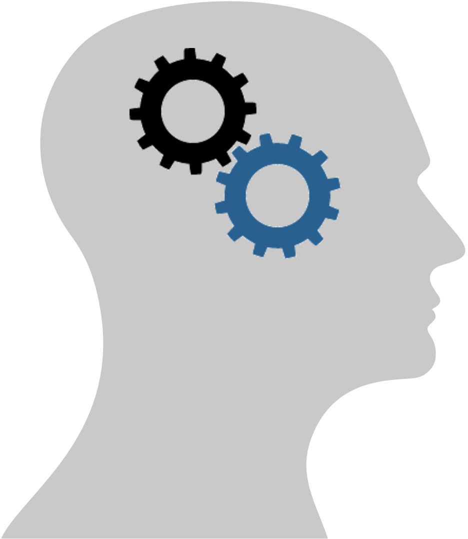
The Big Supermarket War

The UK's big four supermarkets have been engaged in a price war as German discounters expand rapidly in the UK.

One of the leading supermarkets in United Kingdom, Sainsbury’s, was facing a business problem with declining sales during early 2016. The company had two straight years of profit decline, and its share value had fallen by 7% over the past month.

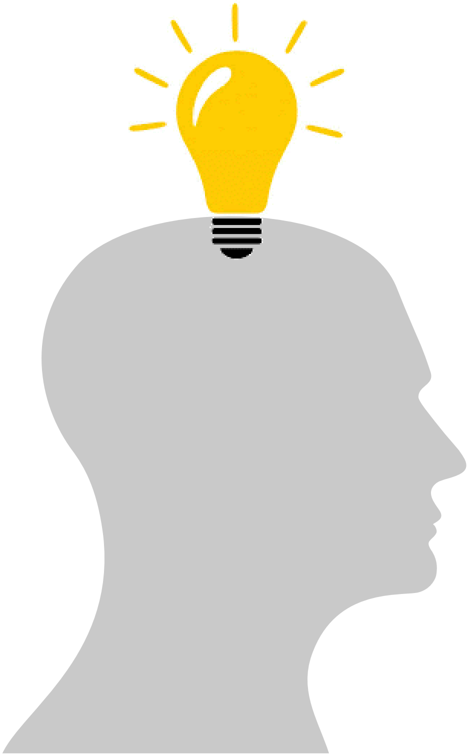
The Comeback

In a few months, Sainsbury's was able to regain its hold in the market and be the **biggest riser** of Financial Times Stock Exchange 100 index.



How did the supermarket achieve this?

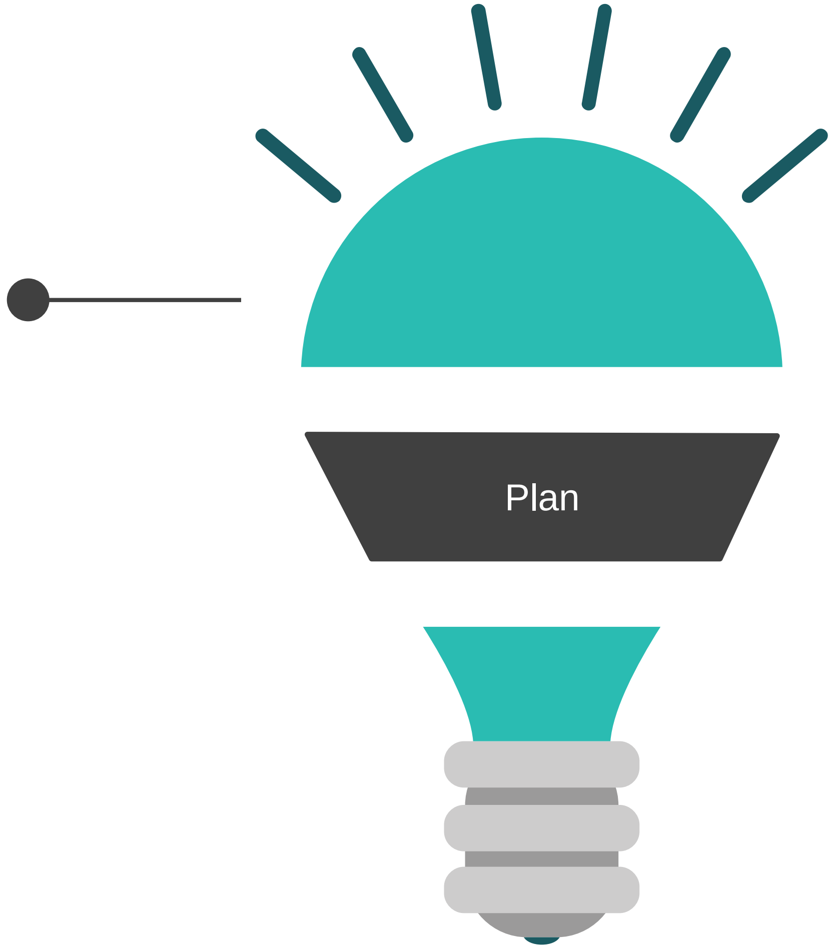
The Comeback



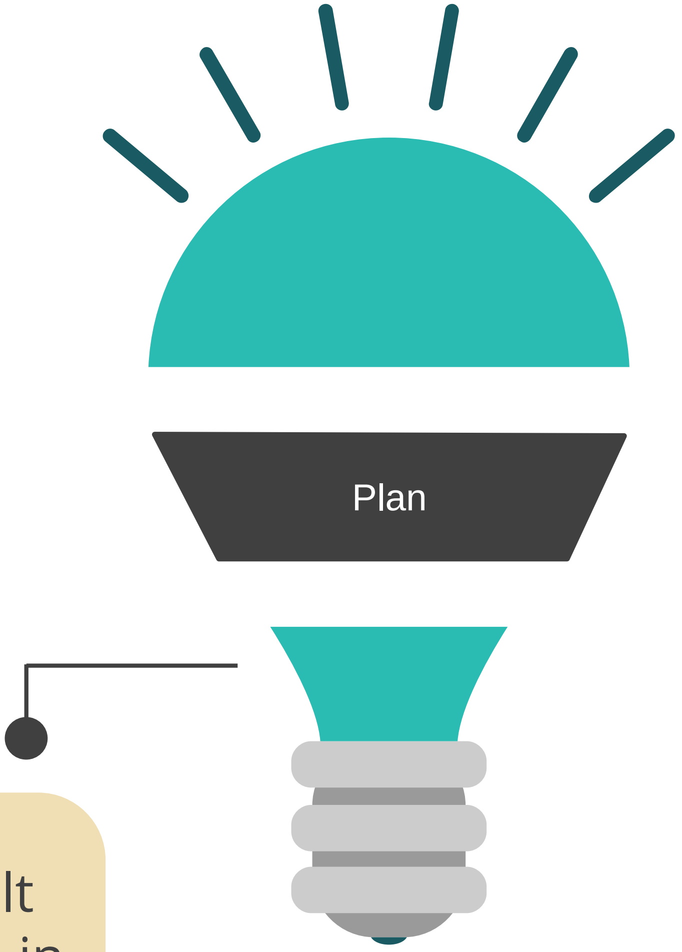
Along with its many other strategies, Sainsbury’s deployed **R software** to its system.

How R Is Implemented at Sainsbury's

The retail store's daily sales are updated and refreshed every 30 minutes from local retail store machines to big data server. R system is connected to the big data server to analyze the real-time data.

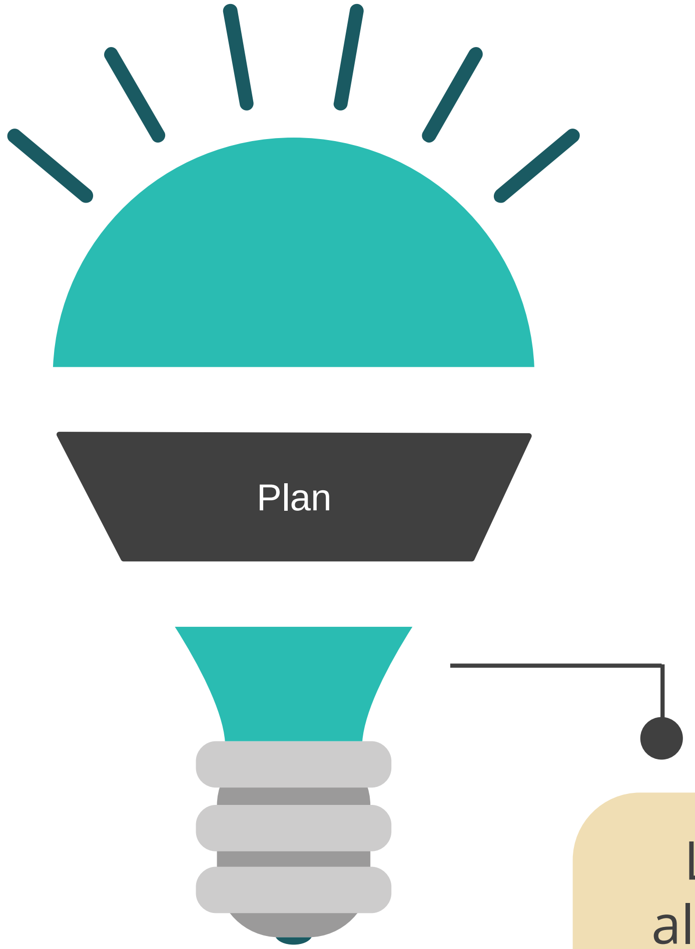


How R Is Implemented at Sainsbury’s



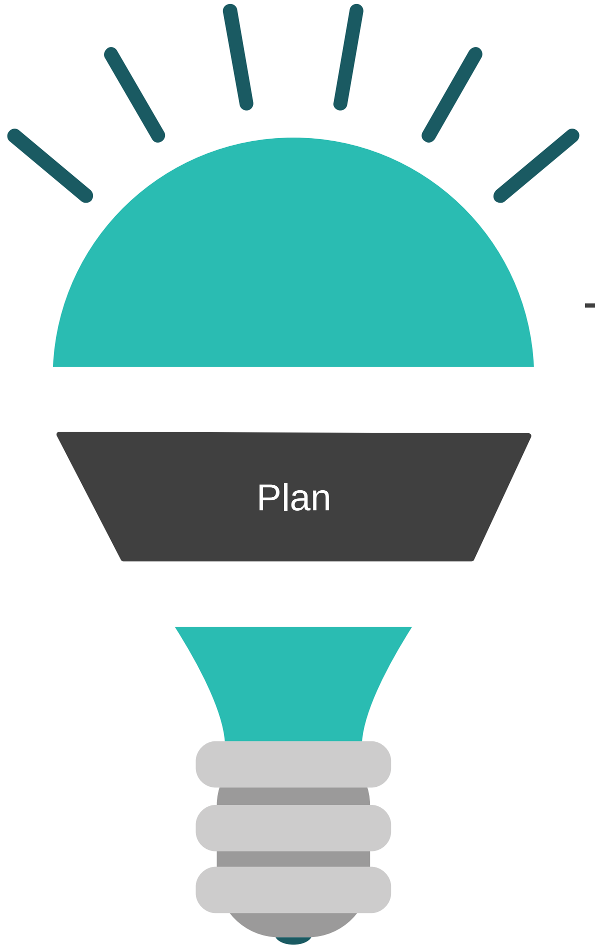
Models and codes are built up in R to trace any decline in sales as per the forecasting model prediction.

How R Is Implemented at Sainsbury's



Live R system will throw alerts if the sales go below the bottom cut-off limit defined in the time series forecasting model.

How R Is Implemented at Sainsbury's



Apriori algorithm is used to give offers on associated products, and time series forecasting modeling technique is deployed to trigger any decline in sales.

What Is R?

R is a free, open-source language used as a statistical and visualization software. It can handle structured as well as semi-structured data.

Features of R

- Is available across all platforms, such as Linux, Mac, and Windows
- Has the ability to integrate with the procedures written in the C, C++, .Net, Python, or FORTRAN languages
- Has an effective data handling and storage facility
- Provides a wide variety of integrated collection of tools for data analytics

R has a worldwide repository system – CRAN. The Comprehensive R Archive Network (CRAN) is a network of sites that acts as the primary web service distributing R sources and binaries, extension packages, and documentation.

Why R for Data Analytics?

Open Source	Statistical algorithm	Visualizations	Machine learning algorithms	Recognized by other software	Customized algorithms

R is a free software.

Why R for Data Analytics?

Open Source | Statistical algorithm | Visualizations | Machine learning algorithms | Recognized by other software | Customized algorithms

R is a statistical software where complex stats models like linear regression, logistic regression, hypothesis testing, ANOVA(Analysis Of Variance), GLM(Generalized Linear Model), etc., can be run.

Why R for Data Analytics?

Open Source

Statistical algorithm

Visualizations

Machine learning algorithms

Recognized by other software

Customized algorithms

R has some great tools to aid data visualization to create graphs, bar charts, multi-panel lattice charts, scatter plots, and new custom-designed graphics.

Why R for Data Analytics?

Open Source	Statistical algorithm	Visualizations	Machine learning algorithms	Recognized by other software	Customized algorithms

ML algorithms like SVM, Naives Bayes theorem, XGboost, Decision tree, and Random forest are available in R readily. These algorithms have proven to be better over time and provide good accuracy of results.

Why R for Data Analytics?

Open Source

Statistical algorithm

Visualizations

Machine learning algorithms

Recognized by other software

Customized algorithms

- You can now write R codes in SAS as R codes are widely used, and programmers are getting familiar with these.
- R can handle semi-structured data and has algorithms built.

Why R for Data Analytics?

Open Source | Statistical algorithm | Visualizations | Machine learning algorithms | Recognized by other software | Customized algorithms

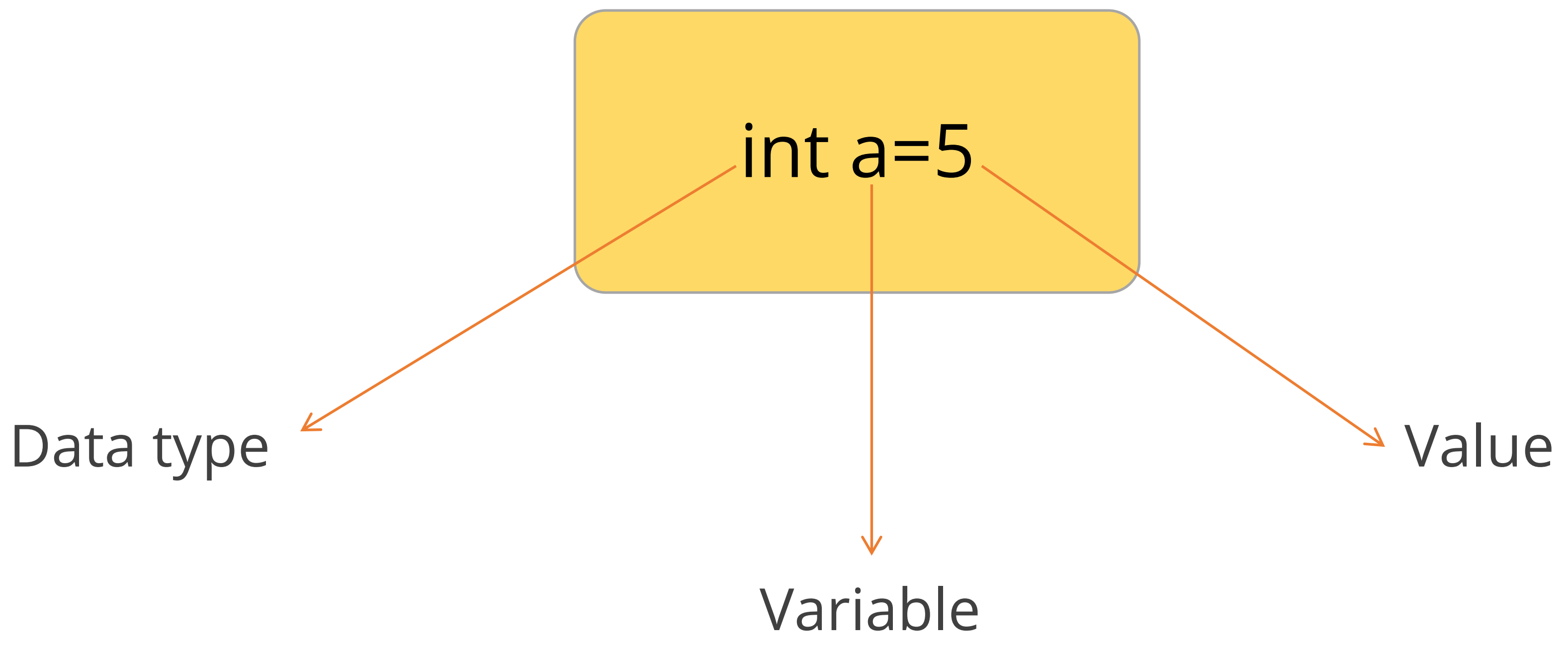
Programmers can define their customized algorithms in R and develop their own algorithms and packages.

Introduction to R Programming

Topic 2— Data Types and Variables in R

Example – Data Types And Variables

Consider the code given below:



```
int a=5
```

Data type

Variable

Value

Data Types in R

Data Type	Description	Example
Logical	Boolean values	TRUE, FALSE
Numeric	Numbers of all kinds	6, 18.5, 825
Integer	Explicit Integers	8L, 45L
Complex	Real Value + Complex Value	7+5i
Character	Characters and Strings	‘g’ “Smith”
Raw	Any data is stored as raw bytes	“Hello” is stored as 68 65 6c 6c 6f

Variables in R

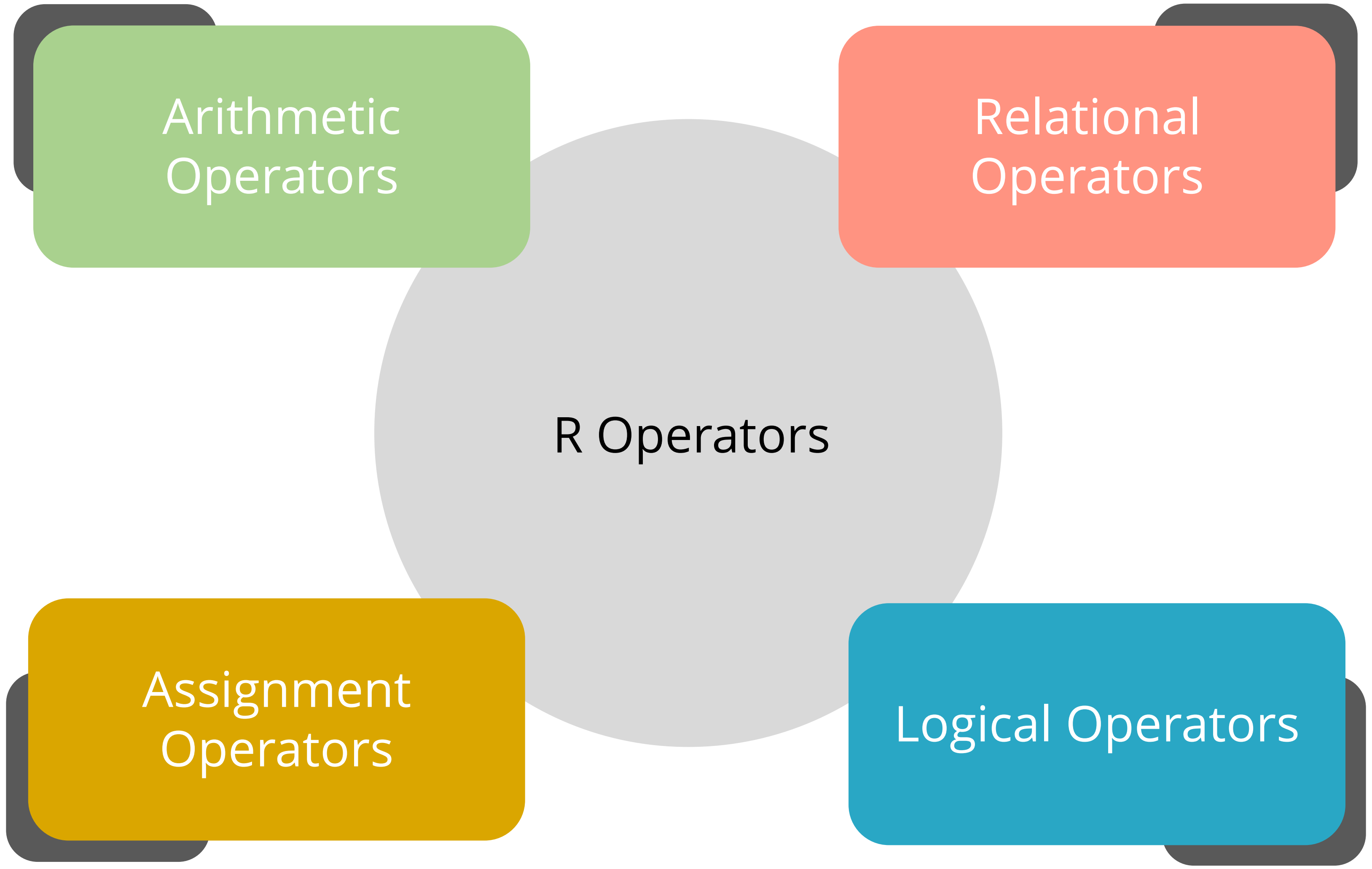
- Variables are logical names for pieces of data. They are used to store data, and their value can be changed.
- The unique name given to a variable is called an identifier.
- Rules for defining variable names are as follows:
  - Variable names can consist of a combination of letters, numbers, underscore, and period.
  - It should begin with a letter or a period. If it starts with a period, then it cannot be followed by a digit.
  - Reserved words in R cannot be used as identifiers.

Introduction to R Programming

Topic 3— Operators in R

Operators in R

R has many operators to perform different mathematical and logical operations.



R Operators

Arithmetic Operators

Relational Operators

Assignment Operators

Logical Operators

Operators in R

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

These operators are used for mathematical operations like addition and multiplication.

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponent
%%	Modulus
%/%	Integer Division

Operators in R

EXAMPLE

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

x=10 ; y =3

Operator	Code	Output
+	x+y	13
-	x-y	7
*	x*y	30
/	x/y	3.333
^	x^y	1000
%%	x%%y	1
%/%	x%/%y	3

Operators in R

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

- These operators are used to compare two values. The result is a logical vector.

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

- While single = assigns value, double == checks if the value of element in an R object is equal to defined value.

Operators in R

EXAMPLE

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

Operator	Code	Output
<	2<3	TRUE
>	2>3	FALSE
<=	2<=2	TRUE
>=	3>=3	TRUE
==	x = 3^4, x==81	TRUE
	x = 3^4, x==80	FALSE
!=	2!=3	TRUE

Operators in R

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

These operators are used to perform Boolean operations such as “AND” and “OR.”

Operator	Description
!	Logical NOT
&	Element-wise logical AND
&&	Logical AND
|	Element-wise logical OR
||	Logical OR

Operators in R

EXAMPLE

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

x=1 ; y=2

Operator	Code	Output
!	x!=10	TRUE
&	x==1 & y==2	TRUE
&&	x<1 && y>4	FALSE
|	x==1 | y==3	TRUE
||	x=2 || y=2	TRUE

Operators in R

Arithmetic Operators

Relational Operators

Logical Operators

Assignment Operators

- These operators are used to assign values to variables.
- Variables are assigned using “<-”, although “=” also works.

Example:

```
age <- 18 (left assignment )
18 -> age (right assignment)
```

Introduction to R Programming

Topic 4— Conditional Statements in R

Conditional Statements in R

- Conditional statements are used to check if certain conditions are met.
- R supports two types of conditional statements:
  - If…else
    - Other forms of if..else are ifelse function and switch statement
  - Nested if…else

Types of Conditional Statements

If…else

Nested if…else

When the test expression is true, the code in the “if” block executes; otherwise, the code in the “else” block executes.

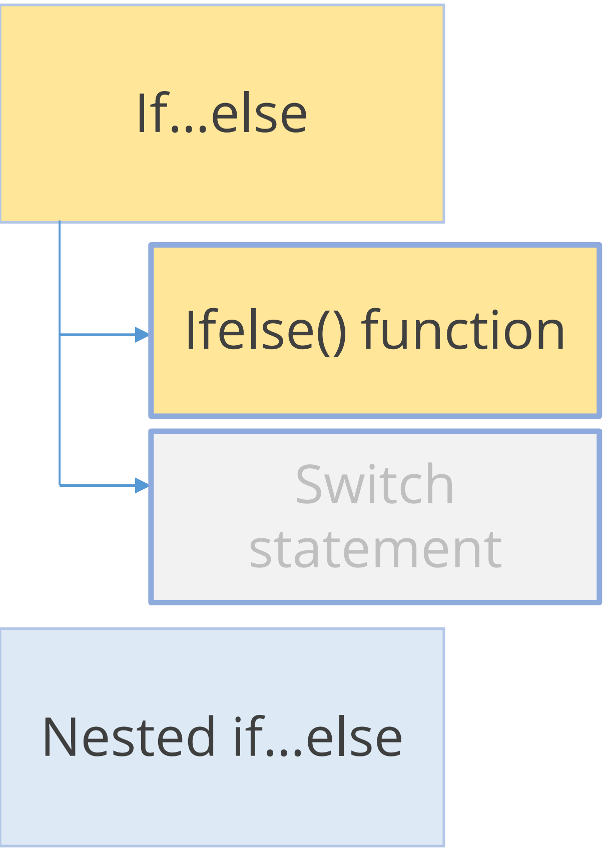
Example:

```
age <- 20
if(age > 18){
  print("Major")
} else {
  print(“Minor”)
}
```

Output:

```
[1] "Major"
```

Types of Conditional Statements



If...else

Ifelse() function

Switch statement

Nested if...else

This is a vector equivalent form of if...else.

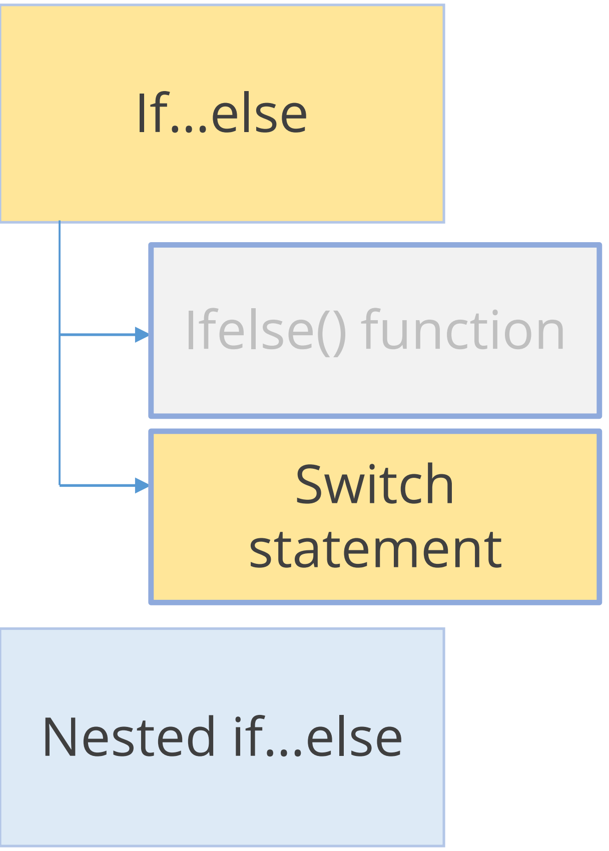
Example:

```
age <- 20
Ifelse(age>=18,
“Major”,”Minor”)
```

Output:

```
[1] "Major"
```

Types of Conditional Statements



- Switch is a control statement that allows a value to change control of execution.
- It is similar to a controlled branch of if...else statement.

Example:

```
age <- "Major"
switch(age,
Major={
    print("Age is
greater than 18")
},
Minor={
    print("Age is
less than 18")
}
)
```

Output:

```
[1] "Age is greater than 18"
```

Types of Conditional Statements

If…else

Nested if…else

In nested if…else statements, only one statement executes, depending on the test expressions in the ‘if’ blocks.

Example:

```
x <- 0
if (x < 0) {
  print("Negative
number")
} else if (x > 0)
{
  print("Positive
number")
} else
  print("Zero")
```

Output:

```
[1] "Zero"
```

Introduction to R Programming

Topic 5— Loops in R

Loop Statement

A loop is a control flow statement that executes a statement or a group of statements many times.

Types of Loops

For Loop

While Loop

Repeat Loop

The for loop executes a code a specific number of times.

Example:

```
vec <-
c(1,2,3,4,5)
for (val in vec)
{
    print(val)
}
```

Output:

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Types of Loops

For Loop

While Loop

Repeat Loop

In the while loop, while the test expression remains true, the code inside the loop keeps on executing.

Example:

```
i <- 1
while (i < 6)
{
  print(i)
  i = i+1
}
```

Output:

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Types of Loops

For Loop

While Loop

Repeat Loop

- A repeat loop iterates a code multiple times.
- Since there is no conditional check to exit the loop, you must specify it inside the body of the loop.

Example:

```
x <- 1
repeat
{
  print(x)
  x = x+1
  if (x == 6){
    break
  }
}
```

Output:

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Loop Control Statements

- Loop control statements exit the loop without executing all the statements.
- Loop control statements used in R:
    - Break statement
    - Next statement

Loop Control Statements

Break Statement

Next Statement

When present inside a loop, it stops the iterations from executing and forces the flow to exit the loop.

Example:

```
num <- 1:5
for (val in num)
{
  if (val == 3)
{
    break
  }
  print(val)
}
```

Output:

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

Loop Control Statements

Break Statement

Next Statement

It helps in skipping the current iteration of a loop.

Example:

```
num <- 1:5
for (val in num)
{
  if (val == 3)
{
    next
  }
  print(val)
}
```

Output:

```
[1] 1
[1] 2
[1] 4
[1] 5
```

Introduction to R Programming

Topic 6— R script

R Script

R script is a text file that contains a set of commands to be executed in R.

Features of R Script

- The commands are executed in the console.
- Script files allow you to execute long and multiple commands easily at once.
- The file is saved with an extension ".R"

Methods to Run an R Script

R script can be run by any of the following methods:

The “source()” function instructs R to read the text file and execute its contents.

It executes the script from the R console.

```
source("myScript.R")
```

“Rscript()” command can be invoked to run an R script from the command line.

```
$ Rscript myScript.R
```

The “R CMD BATCH” command can be used to run the code in batch mode from the command line.

```
$ R CMD BATCH myScript.R
```

Introduction to R Programming

Topic 7— Functions in R

R Functions

A function is a code used to execute a specific task.

Some Aspects of R Functions

- Functions are stored as R objects.
- There are over 1000 functions at the core of R, and new R functions are created all the time.
- Each R function comes with its own help page. To access a function’s help page, type a question mark followed by the function’s name in the console.

R Functions

Function	Description
append()	Add elements to a vector
c()	Combine values into a vector or list
identical()	Test if two objects are exactly equal
length()	Return the length of the R object
ls()	List objects in the current environment
range(x)	Return the minimum and maximum of the vector
rep(x,n)	Repeat the number x, n times
rev(x)	Provide the reversed version of an argument
seq(x,y,n)	Generate regular sequences from x to y, spaced by n
unique(x)	Remove duplicate entries from the vector

R Functions

Function	Description
tolower()	Convert a string to lower case letters
toupper()	Convert a string to upper case letters
grep()	Use for regular expressions
summary(x)	Return object summaries
str(x)	Compactly display the structure of an arbitrary R object
glimpse(x)	Compactly display the structure of an arbitrary R object (dplyr package)
class(x)	Return the class of an object
mode(x)	Get or set the type or storage mode of an object
summary(x)	Return object summaries
tolower()	Convert a string to lower case letters

Key Takeaways

- ✓ R is a free, open-source programming language that is most useful for statistical computation and visualization.
- ✓ There are six data types in R: logical, numeric, integer, complex, character, and raw.
- ✓ Variables are used to store data, and the unique name given to it is called identifier.
- ✓ There are four types of operators in R: arithmetic, relational, logical, and assignment.
- ✓ Conditional statements are used to verify if the required conditions are met. The two conditional statements used in R are if…else and nested if…else.
- ✓ A loop is a control flow statement that executes a statement or a group of statements many times. The loop statements used in R are for loop, while loop, and repeat loop.
- ✓ R script is a text file that contains a set of commands to be executed in R.
- ✓ A function is a code used to execute a specific task, and there are over 1000 functions available in R.