PG Certificate Program

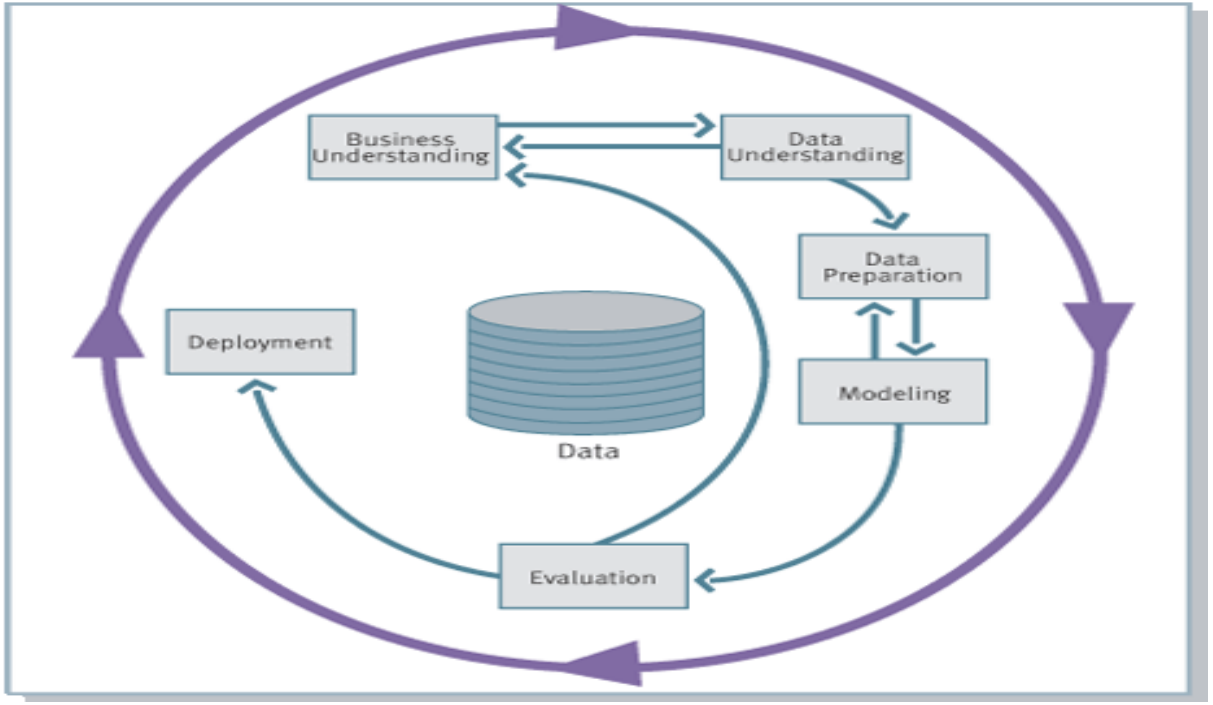
Data Science and Business Analytics

Introduction to R and Exploratory Data Analysis

Course instructor: Prof. Mahima Gupta

E-mail id: mahima.gupta@iimamritsar.ac.in

Crisp DM framework



Data Mining Tasks:
Description
Estimation
Prediction
Classification
Clustering
Association

Cross Industry Standard Process: CRISP-DM (*cont'd*)

(1) Business/Research Understanding Phase

- Define project requirements and objectives
- Translate objectives into data mining problem definition
- Prepare preliminary strategy to meet objectives

(2) Data Understanding Phase

- Collect data
- Perform exploratory data analysis (EDA)
- Assess data quality
- Optionally, select interesting subsets

(3) Data Preparation Phase

- Prepares for modeling in subsequent phases
- Select cases and variables appropriate for analysis
- Cleanse and prepare data so it is ready for modeling tools
- Perform transformation of certain variables, if needed

Cross Industry Standard Process: CRISP-DM (*cont'd*)

(4) Modeling Phase

- Select and apply one or more modeling techniques
- Calibrate model settings to optimize results
- If necessary, additional data preparation may be required for supporting a particular technique

(5) Evaluation Phase

- Evaluate one or more models for effectiveness
- Determine whether defined objectives achieved
- Establish whether some important facet of the problem has not been sufficiently accounted for
- Make decision regarding data mining results before deploying to field

Cross Industry Standard Process: CRISP-DM (*cont'd*)

(6) Deployment Phase

- Make use of models created
- Simple deployment example: generate report
- Complex deployment example: implement parallel data mining effort in another department
- In businesses, customer often carries out deployment based on your model

Software

- Microsoft Excel or other spreadsheet programs like Google Sheets
- Proprietary Statistical Software: SAS, Stata or SPSS

Limitations:

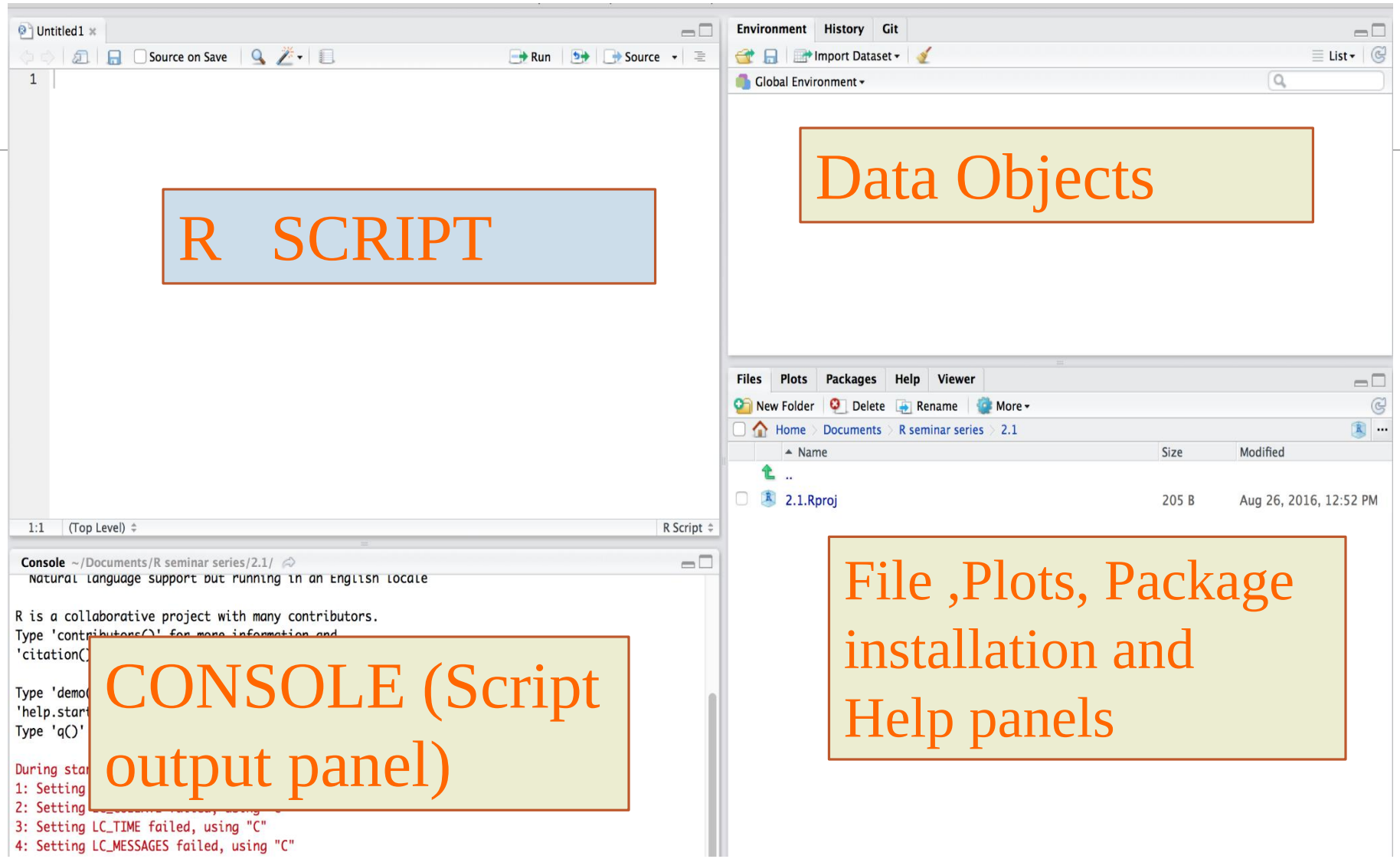
- Excel cannot handle datasets above a certain size.
- Reproducing previously conducted analyses on new datasets is challenging.
- Programs like SAS were developed for very specific uses.
- Don’t have a large community of contributors constantly adding new tools.

Next Step

R or Python

- Both are free and open source, and were developed in the early 1990s.
- R for statistical analysis and Python as a general-purpose programming language.
- But for data analysis, the differences between R and Python are starting to break down.
- https://www.guru99.com/r-vs-python.html

RStudio


Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
R SCRIPT
Environment
History
Git
Import Dataset
List
Global Environment
Data Objects
Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Home
Documents
R seminar series
2.1
Name
Size
Modified
..
2.1.Rproj
205 B
Aug 26, 2016, 12:52 PM
Console ~/Documents/R seminar series/2.1/
Natural language support but running in an English locale
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()
Type 'demo(
'help.start
Type 'q()'
During star
1: Setting
2: Setting
3: Setting LC_TIME failed, using "C"
4: Setting LC_MESSAGES failed, using "C"
CONSOLE (Script output panel)
File ,Plots, Package installation and Help panels

Name	Age	Occupation					
Mahima							
Rajan							
Vikas							

Get started with R

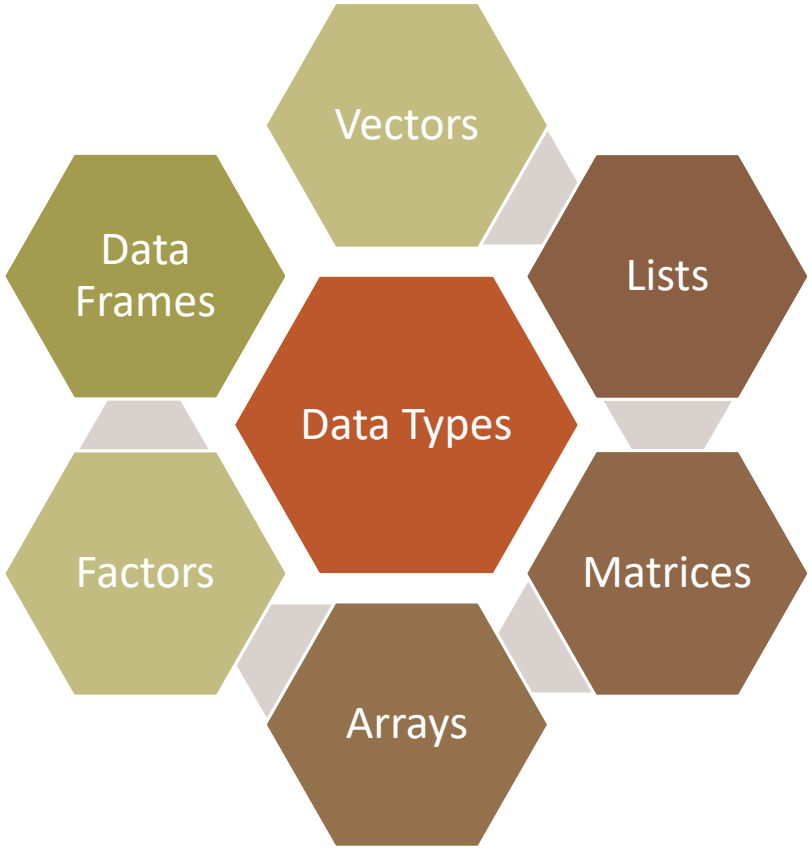
- Variables in R
- Operators in R
- Data Types in R
- Graphs in R

https://www.tutorialspoint.com/r/index.htm

Variables in R

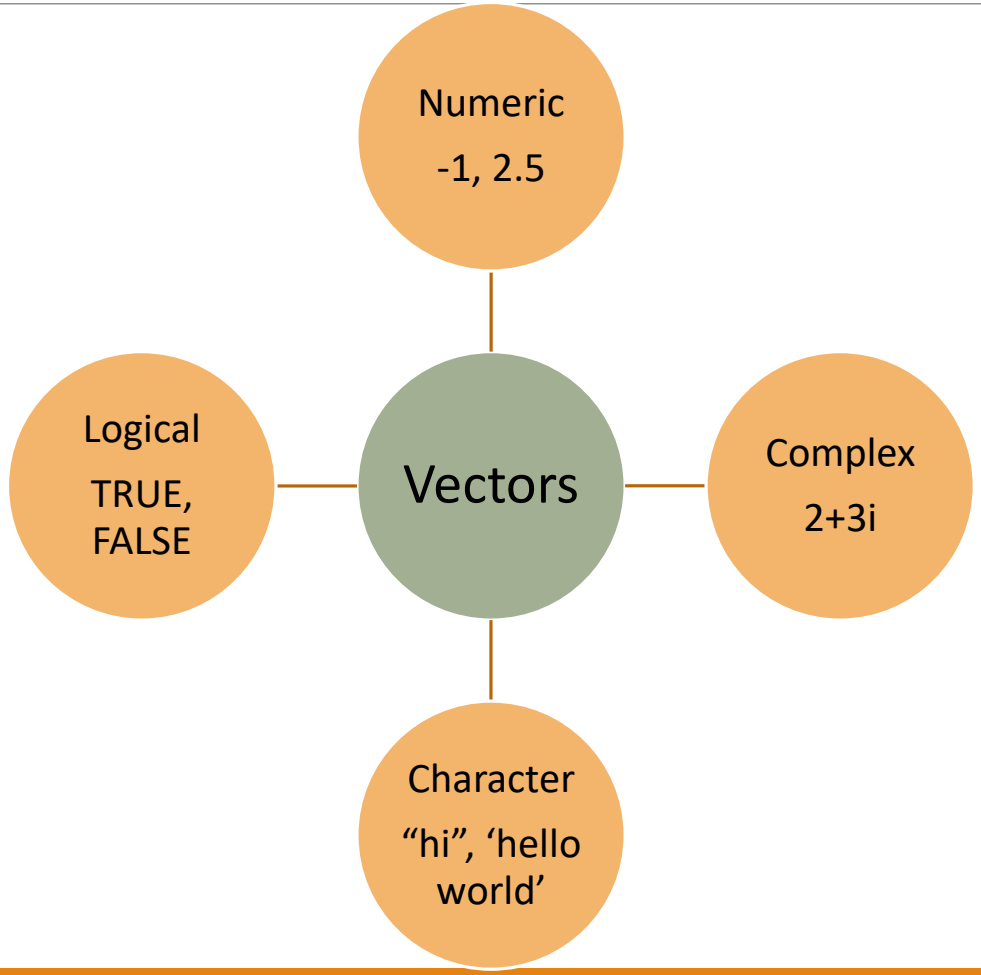
- Variables are reserved memory locations to store values.
- A valid variable name consists of letters, numbers and the dot or underline characters. The variable name starts with a letter or the dot not followed by a number.
- The variables can be assigned values using leftward (commonly used), rightward and equal to operator.
- Subject.1 <- “Maths”
- X_1 = 5
- TRUE -> .abc
- Some invalid names: .123, A%1,_class
- The variables are assigned with R-Objects and the data type of the R-object becomes the data type of the variable.

Data Types in R



Vectors

- A vector is a sequence of data elements of the same data type.
- Types of vectors: logical, numeric, complex, character
- If vector is defined of different basic types, the lower ranking type will be *coerced* into the higher ranking type.
- In general, The hierarchy for coercion is: logical < numeric < character
- Logicals are coerced a bit differently depending on what the highest data type is.



Operations in Vectors

Indexing

starts with 1; Accessed through []; Negative index is used for dropping the element

Age <- c(12,14,15,16)	Age [1] : 12	Age[-2]: 12, 15, 16
Age[Age>14]: 15, 16	Age[c(1,3)]: 12,15	Age[2:4]: 14, 15, 16

Replacing

Age[2] <- 16

Other functions

length, class

Operators in R

Arithmetic Operators	+ - * / ^ %%(Remainder) %/%(integer quotient)
Relational Operators	Give Boolean value as output < > == != >= <=
Logical Operators	&	! (Element wise) && || (first element comparison)
Assignment Operators	<- <<- > >>- =
Miscellaneous Operators	: (It creates the series of numbers in sequence for a vector)

https://excelquick.com/r-programming/assignment-operators-in-r/
https://renkun.me/2014/01/28/difference-between-assignment-operators-in-r/

Data Frames

- A data frame is a table or a two-dimensional array-like structure.
- Each column contains values of one variable.
- Each row contains one set of values from each column.
- The data stored in a data frame can be of numeric, factor or character type.
- Each column should contain same number of data items.

```
emp.data <- data.frame( emp_id = c (1:5), emp_name =
c("Rahul",“Rohan","Michelle","Ryan","Gaurav"), salary =
c(623.3,515.2,611.0,729.0,843.25))
```

- Some important commands : nrow(),ncol(),dim(),names(), rownames(),colnames(), head(), tail(), rbind(), cbind(), summary()

- Matrix : Same atomic type elements are arranged in a two-dimensional rectangular layout.

  A <- matrix(data, nrow, ncol)

  Indexing: A[1,2]; A[c(1,3),];A[c(1,3),-1]

- Arrays: Store data in more than two dimensions.

  vector1 <- rep(c(2,5),5) ; vector2 <- c(10,15,13,16,11,12)

  a<-array(c(vector1,vector2),dim=c(2,2,4))

- Lists contain elements of different types − numbers, strings, vectors and another list inside it.

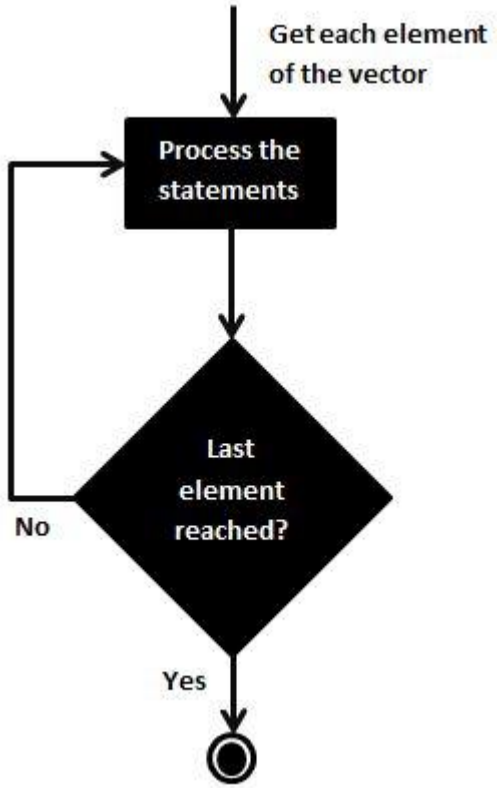
  Student <- list(c(“Arjun”, ”Ram”), 25, TRUE)

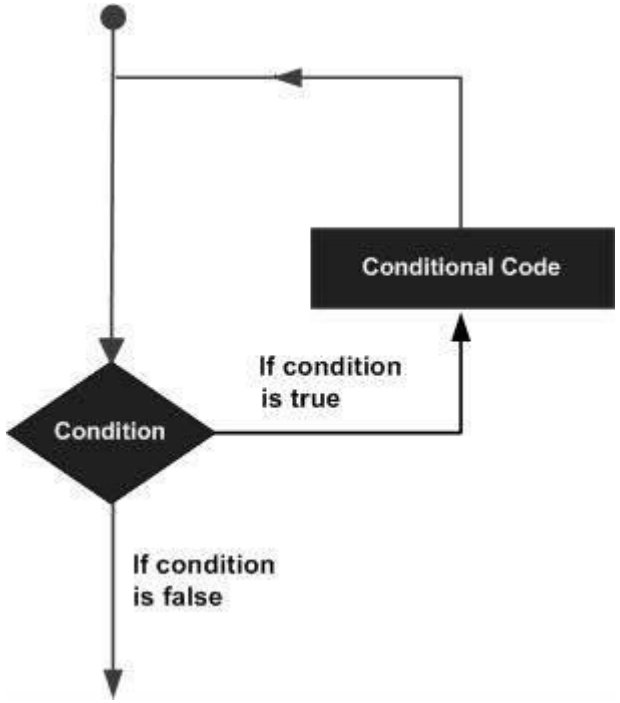
Factors

- Factors are used to categorize the data and store it as levels.
- They can store both strings and integers.
- They are useful in the columns which have a limited number of unique values. Like "Male, "Female" and True, False etc.
- They are useful in data analysis for statistical modeling.
- Factors are created using the **factor ()** function by taking a vector as input.

```
input. data <-
c("East","West","East","North","North","East","West","West","West","East","North")
factor_data <- factor(input.data)
```

Control structure in R





For loop

for (*variable* in *sequence*)
{ *expression expression expression* }

Example 1:

```
v <- c(1,2,3,4)
for ( i in v) {
  print(i)
}
```

```
for (j in 1:5) { print(j^2) }
```

If Loop

The keyword if

A single logical value between parentheses (or an expression that leads to a single logical value)

A block of code between braces that has to be executed when the logical value is TRUE

```
if(val %% 2 == 0) {count = count+1 }
```

```
# ignore the curls if it is only one statement

ifelse(test_expression, x, y)

a = c(5,7,2,9)

 ifelse(a %% 2 == 0,"even","odd")

[1] "odd" "odd" "even" "odd
```

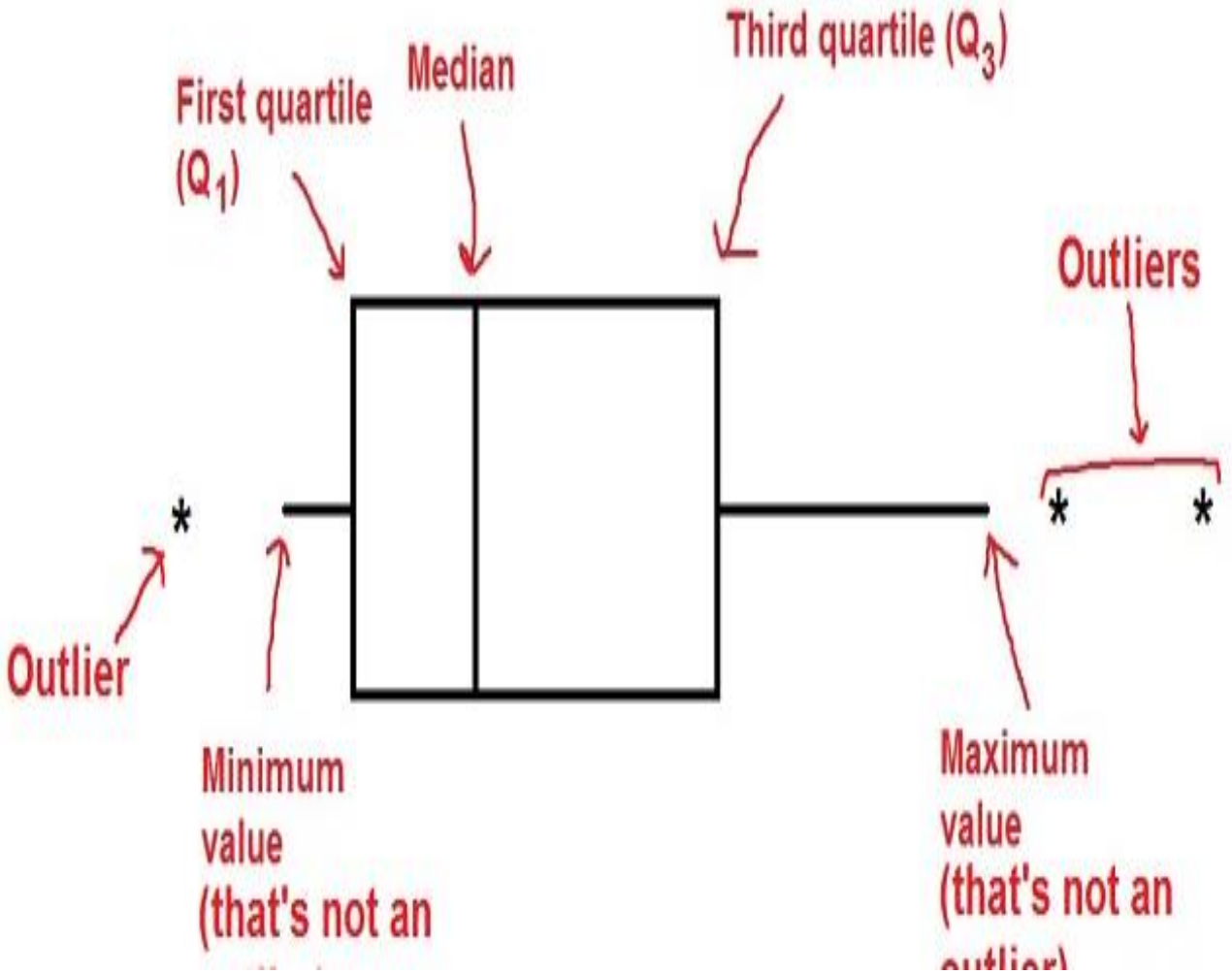
If else

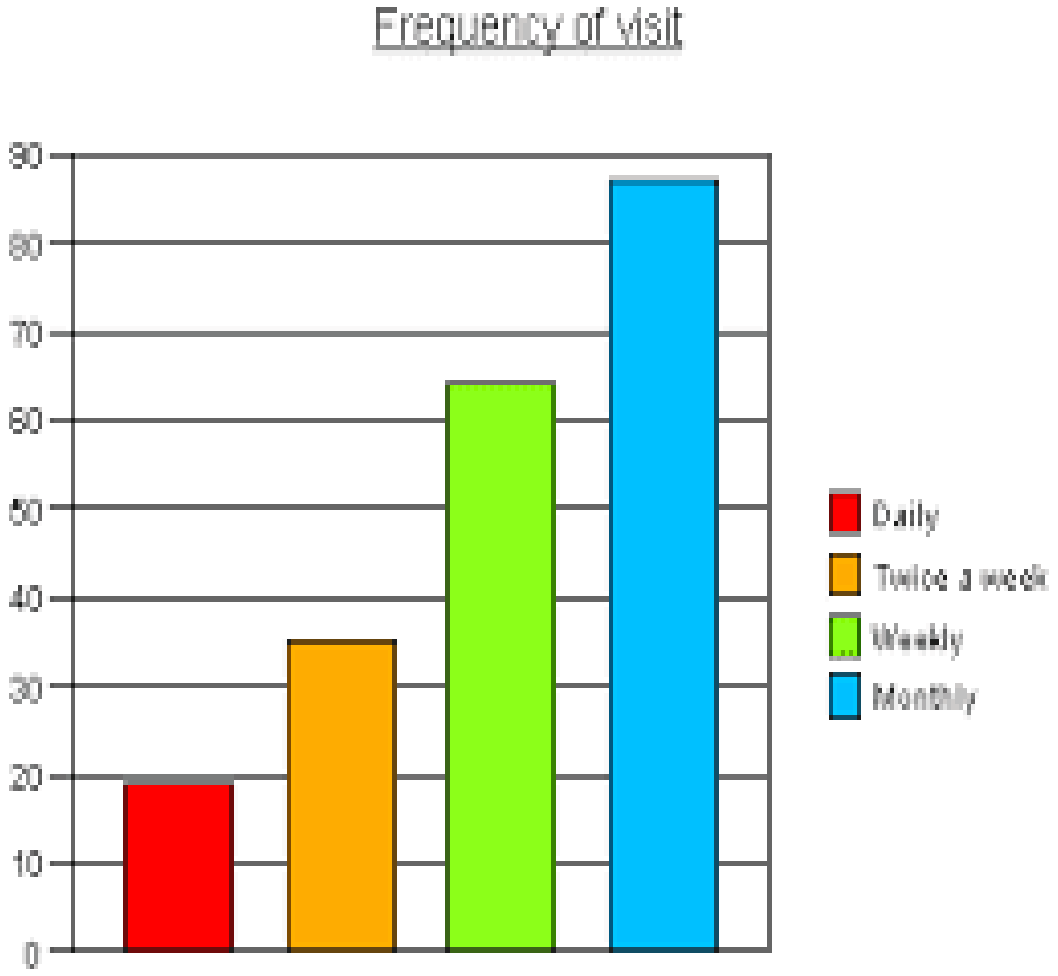
if(boolean_expression 1) { // Executes when the boolean expression 1 is true. }

else if(boolean_expression 2) { // Executes when the boolean expression 2 is true. }

else if(boolean_expression 3) { // Executes when the boolean expression 3 is true. }

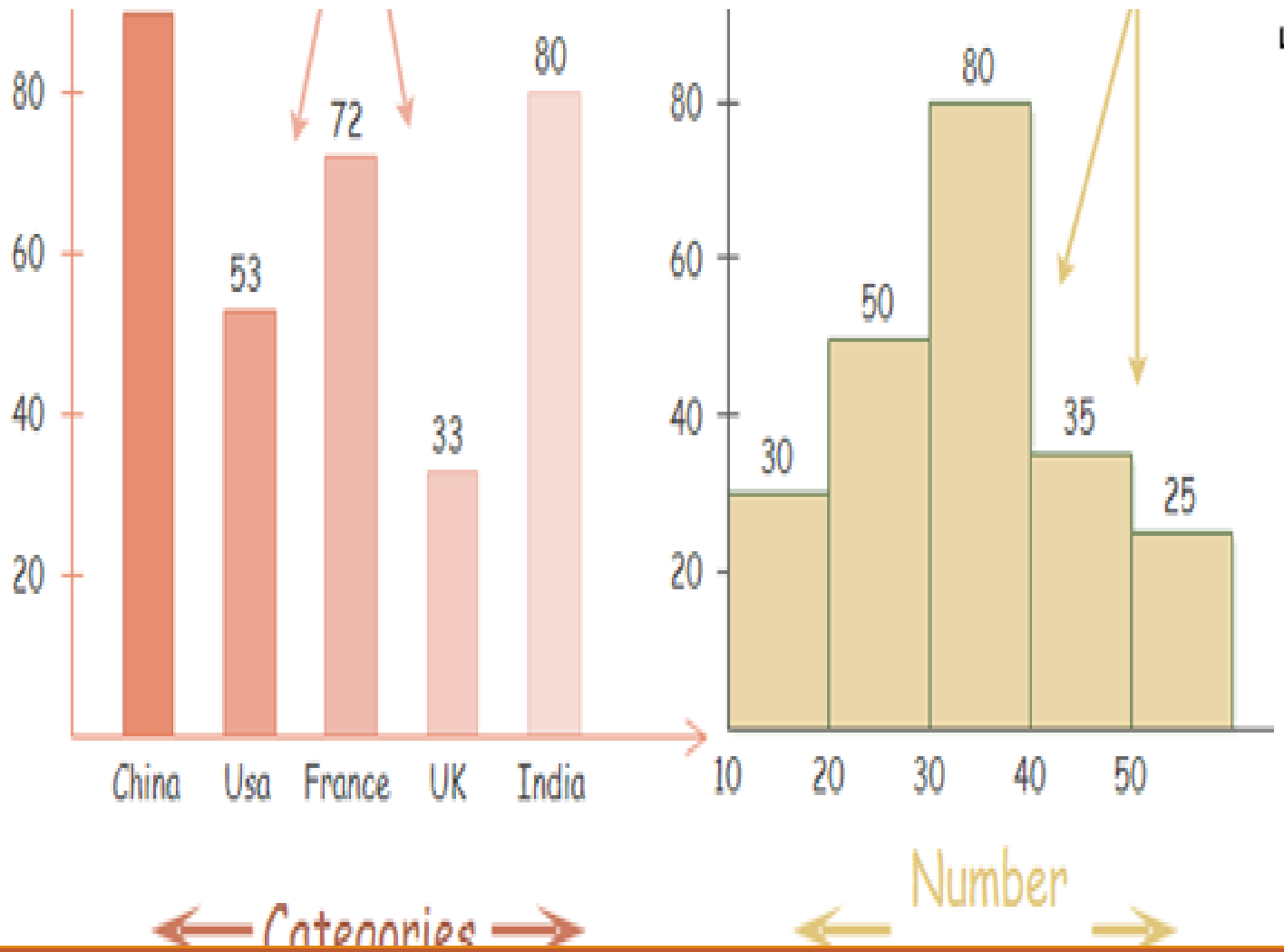
else { // executes when none of the above condition is true. }

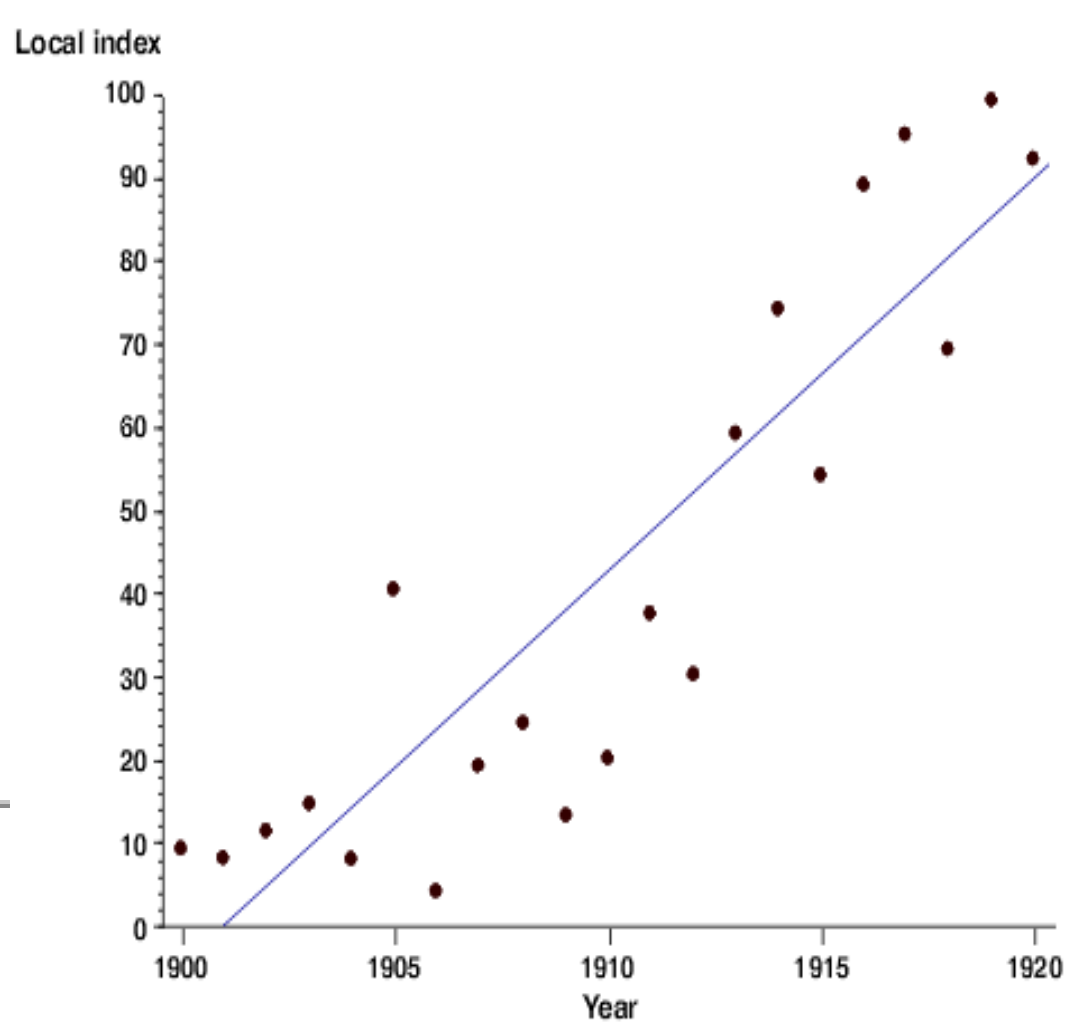




Data Visualization in R

Boxplot, Barchart





Data Visualization in R

Histogram, Line Graph, Scatterplot