R - Basics

Shah Ayub Quadri

Ayub.quadri89@gmail.com

R Course curriculum

Introduction to R
- History of R
- Installation & R Studio IDE

Variables & identifiers
- What is a variable?
- What is Identifiers
- Rules to declare a variable
- Variables vs Identifiers

Data Types
- Numeric
- Integer
- Complex
- Logical
- Characters

Operations in R
- Arithmetic operators
- Relational operators
- Logical operators
- Assignment operation

Data Structures
- Vectors
- Matrix
- Lists
- Data Frame
- Factor
- Strings

Importing data into R
- Build-in data
- Excel-csv file
- Text file
- ODBC data

Functions
- Basic function
- In build functions
- Parameters
- Return values
- Variable scope
- Exception handling

Apply family
- Apply function
- Lapply
- Rapply
- Mapply
- Sapply
- Tapply

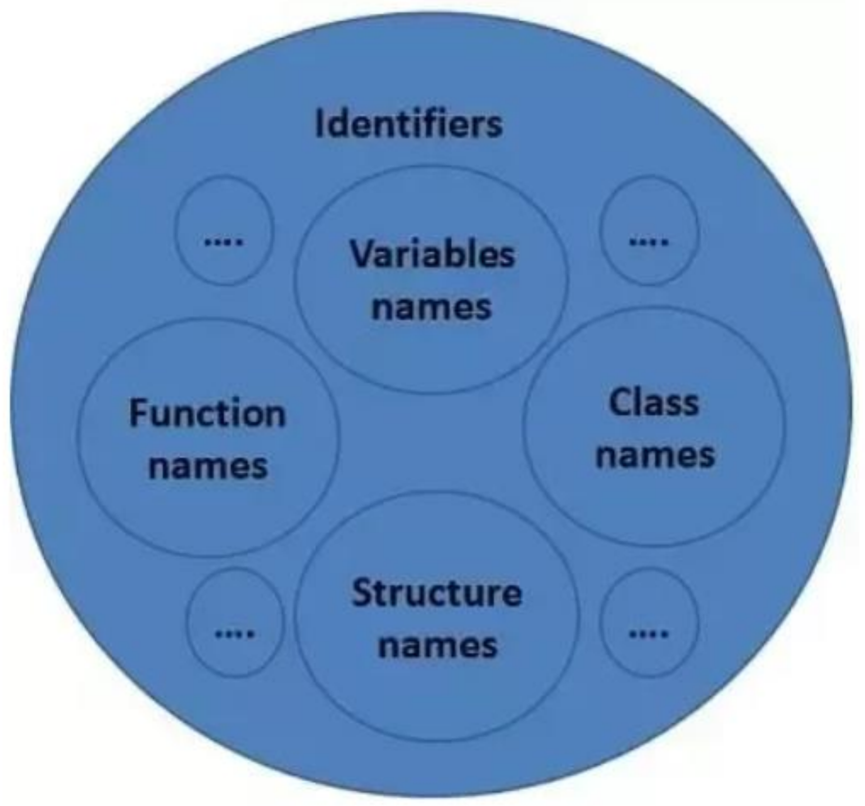
Graphs & Plot
- Histogram, Box plot
- Line graph, ggplot, ggplot2
- Importance of colours & shapes in graphs

Identifier

Identifier is a name given to an entity, which distantly identifies an entity in a program

Here the entity can be

- Variable Name
- Function Name
- Class name
- Structure name



Variables

Variables:

- This are the placeholders which can hold some data, this data can be altered as & when required

  Eg: x = 5

     _y = 10

Rules for variables

- Variables names should starts with letter or period, in the later period case it should not be followed by a digit.

- Variable name can be a combination of letters, numbers and underscore.

- Reserved word cannot be used for identifiers.

Valid & Invalid variables

Valid Variables: those that follow the thumb rule of variable declaration.

Eg: a = 4, .b= 1, a10 = 20, a_20 = 30

Invalid variable: those variables that don’t flow the rules

Eg: .1 =2, _abc = 12, @a=10, if = 33, else = 24

Constants

- Numeric constant

Hexa decimal value	Numeric value
0x1	1
0x2	2
0x9	9
0xa	10
0xb	11
0xc	12
0xd	13

- Character constants

eg: ‘example’

typeof(‘example’)

- Build-in constants: those constants which are inbuild with R package

eg: month.abb, month.names, pi, LETTERS

Data Types in R

- There are 5 data types in R
  - **Numeric**: Decimal values are referred as Numeric in R
    - Eg: x = 1.78

  - **Integer**: it includes integer number
    - Eg: x =10

  - **Complex**: it consists of combination of real number part and imaginer part
    - Eg: x = 21+8i

  - **Logical**: this data type holds logical values such as TRUE or FALSE
    - Eg: x = c(TRUE, FALSE, FALSE, TRUE)
      x = a < b

  - **Character**: this data type holds sequence of characters or string elements
    - Eg: x = ‘Welcome to R programming’

Operators in R

Arithmetic operators – These are the operators that perform arithmetic operations.

Addition	+
Subtraction	–
Multiplication	*
Division	/
Exponent	^
Modulus	%%
Integer division	%/%

Relational operators – These operators compare two values.

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

Logical operators – These operators are used to carry-out logical operations.

Logical AND	&&
Element-wise Logical AND	&
Logical OR	II
Element-wise logical OR	I
Logical NOT	!

Assignment operators – These operators assign values.

Rightwards assignment	=>, ->>
Leftwards assignment	<=, <<-

Data Structures in R

Data Structure defines the specific form of *organizing & storing* the data.

R has 6 types of data structures

Vectors: It contains the similar type of data

- Eg: x = 1: 5, y = c(10,20,30,40,50);

Matrix: it's a two-dimensional data structure consisting of rows and column.

- Eg: x = matrix(1:6, nrow =2, ncol = 3);

List: this data structure includes data of different types

- Similar to vectors but vector contains similar type of data.
- Eg: x = list("a" = 2.5, "b" = True, "c" = 1:5)

  str(x)

Data Frame: Special case of list where each component is of same length

- Created using data.frame()
- Eg: x = data.frame("SN" = 1:2, "Age" = c(12,15), "Name" = c("R" ,"Programming"))

Factor: this are used to store categorical data

- Eg: x = factor("Software Engg", "Data Scientist", "Data Engg", "Data Scientist")

Strings: Any value written in single or double quotes are strings

- Eg: x = 'Welcome to R programming'; y = "New style of writing strings";

Install Packages in R

To install package:

One can install using two methods

- Command line approach
- GUI based approach

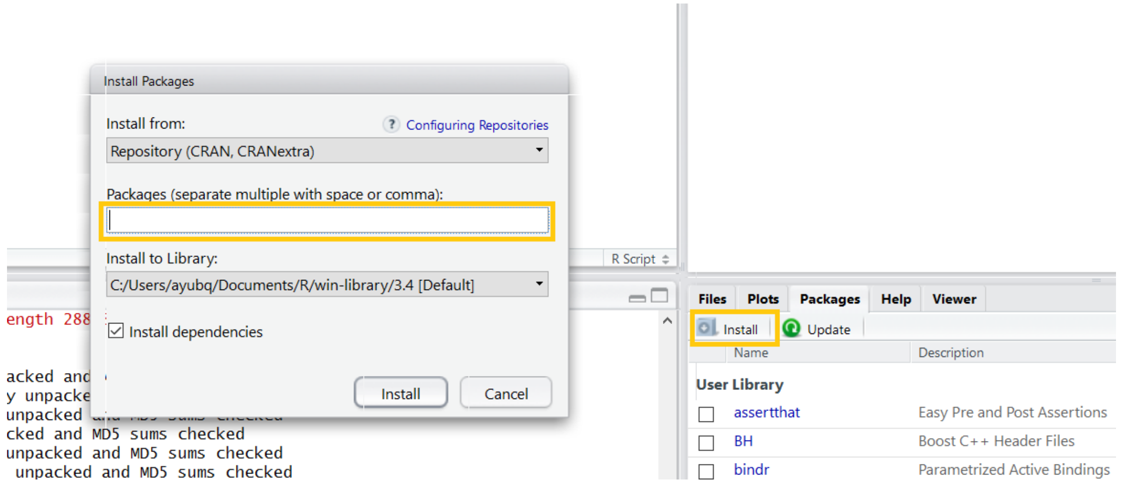
Command Line Based Approach:

Syntax:

install.package('package_name')

Eg: install.packages('dplyr')

GUI Based Approach:



Install from: Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma):

Install to Library: C:/Users/ayubq/Documents/R/win-library/3.4 [Default]

☑ Install dependencies

```
ength 288

acked and
y unpacke
unpacked
cked and MD5 sums checked
unpacked and MD5 sums checked
 unpacked and MD5 sums checked
```

	Name	Description
	User Library	
☐	assertthat	Easy Pre and Post Assertions
☐	BH	Boost C++ Header Files
☐	bindr	Parametrized Active Bindings

Import file to R

Sources:

- Inbuild data set

  Eg: data_frame = load(iris)

- Local: csv, text file

  Eg: data_frame = read.csv(‘test.csv’, header=T,sep=“,”)

- URL: specify the url which points to CSV file

  Eg: df <- read.csv("https://s3.amazonaws.com/assets.datacamp.com/blog_assets/scores_timed.csv", header = TRUE, sep =“,”)

Apply family in R

Apply:

- When you want to apply a function to the rows or columns of a matrix
  - Returns a **vector** or **array** or **list** of values obtained by performing an operation to margins of an array or matrix
  - Margins of matrices 1 – Rows, 2 – column, 1:2 – Both

L apply:

  - When you want to apply a function to each element of a list
  - Returns a list which is of same size as that of input list(X) by applying a function or operation

S apply:

  - When you want to apply a function to each element of a list in turn, but you want a vector back, rather than a list.

M apply:

- when you have several data structures (e.g. vectors, lists) and you want to apply a function to the 1st elements of each, and then the 2nd elements of each, etc.
  - Multiplicative version of sapply
  - It applies function to the first element of each argument, then second elements, then 3rd and so on.

T apply

- when you want to apply a function to subsets of a vector
  - applies a function or operation on subset of the vector broken down by a given factor variable

Working with data in R

- Import the data
- Understanding the data variables-- what are their types
- Split the data into test and train
- Data type conversions
- Descriptive stats for distribution of data and for outlier detection
- Looking at the missing values --either removing or imputing them
- Standardizing the data
  - a. Using Standardization
  - b. Using range
- Converting the variables
  - From Categorical to numeric --Dummy
  - From Numeric to categorical -- Discretizing

References

Apply family functions

- https://stackoverflow.com/questions/3505701/r-grouping-functions-sapply-vs-lapply-vs-apply-vs-tapply-vs-by-vs-aggrega

Introduction to R Course

- https://www.datacamp.com/courses/free-introduction-to-r