Getting to know the Basics of R

Edson Kambeu @ekambeu

4/09/2021

Introduction-What is R?

- Statistical computing programming language
- Free- Open source software
- Used by statisticians and data scientists
- Flexible (enables easy data manipulation and analysis)
- Reproducible data analysis
- Popular in academia and industry

Setting up

- Make sure your R environment is set up
- Download R from CRAN
- Download R Studio

R Studio

- An IDE(Integrated Development Environment) for R
- Code editing
- Code execution
- Code debugging
- Workspace management

Data types in R

- Numeric/ Double-numbers with decimal value/floating point numbers
- Integer-whole numbers
- Character-letters/numbers enclosed in quotes
- Logical - True/False(also called boolean values)
- Factor-data which is categorized
- Character- used to represent string values

Data Structures in R

- Vector
- Matrix
- Lists
- Data frame

Vector

- This is the basic data structure in R
- It is used to store a sequence of data of the same type
- For example can store a sequence of numbers or strings
- Assign variable names to a vector

```
#Vector of numbers 1,2,3
my_num <- c(1, 2, 3, 4)
my_num
```

```
## [1] 1 2 3 4
```

```
#Vector of names
my_names <- c("John", "Peter", "James", "Ed")
my_names
```

```
## [1] "John"  "Peter" "James" "Ed"
```

Vectors-Continued

- More vectors of different data types

```
#vector of logical values
heart_disease <- c(TRUE, TRUE, FALSE, FALSE)
heart_disease
```

```
## [1]  TRUE  TRUE FALSE FALSE
```

```
#Vector of factor
gender <- c("male", "female", "male", "female")
gender_factor <- factor(gender)
gender_factor
```

```
## [1] male   female male   female
## Levels: female male
```

Checking type of data contained in a vector

- Use the typeof() function to check type of data in a vector
- You can also use is.datatype() function to check data type

```
#What is the data type for my_names?
typeof(my_names)
```

```
## [1] "character"
```

```
#Is my my_names a character
is.character(my_names)
```

```
## [1] TRUE
```

Matrix

- Is a collection of elements of the same data type
- The data elements are arranges in rows and columns

```
# Arranging my numbers as a matrix
my_num_matrix <- matrix(my_num, nrow = 2, byrow = TRUE)
my_num_matrix
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

Data Frame

- It is used to store a list of vectors
- The vectors must be of equal length
- We can store vectors of different data type

```
#Creating a data frame
my_df <- data.frame(my_names, my_num)
my_df
```

```
##   my_names my_num
## 1     John      1
## 2    Peter      2
## 3    James      3
## 4       Ed      4
```

Lists

- A list can contain elements of different data types, vectors

```
#Creating a list of my data objects
my_list <- list(my_num, my_names, my_num_matrix, my_df)
my_list
```

```
## [[1]]
## [1] 1 2 3 4
##
## [[2]]
## [1] "John"  "Peter" "James" "Ed"
##
## [[3]]
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
##
## [[4]]
##   my_names my_num
```

Class of data structure

- To check of the class of data object use the class function
- Yo can use is function to find out if its a particular data structure

```
#Checking the type of data structure for my_df
class(my_df)
```

```
## [1] "data.frame"
```

```
#Is my_df a data frame?
is.data.frame(my_df)
```

```
## [1] TRUE
```

Selecting data elements from a vector

- Each element in a vector has a position/index
- Locate an element in a vector by using: variable_name[index]

```
#What is the first data element in my_names?
my_names[1]
```

```
## [1] "John"
```

Selecting data elements from a matrix

-Locate an element by indexing the row and column -Use matrix_name[row_index, column_index]

```
#What is the value of an element in the second row in colu
my_num_matrix[2,2]
```

```
## [1] 4
```

Selecting data elements from a data frame

- Use dataframe_name[row_index, column_index]
- You can use the dataframe_name$variable_name[index]

```
# What is the first element of my_names in my my_names dat
my_df$my_names[1]
```

```
## [1] "John"
```

```
#What are the elements in my_names?
my_df[,1]
```

```
## [1] "John"  "Peter" "James" "Ed"
```

Selecting data elements from a list

- Use double brackets to access a component in a list
- Then use same concepts as learnt before
- Example my_list_name[[list_index]][element_index]

```
#What is the first element in first component of my_list?
my_list[[1]][1]
```

```
## [1] 1
```

```
#What is the element in row 1 and column 2 in third compon
my_list[[3]][1, 2]
```

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

Base R Functions

- R uses functions in_built to perform tasks
- In R Studio hover a function to see its arguments
- To see the complete documentation of a function:
- Use help(function_name)

```
#Checking documentation of the help function
help("mean")
```

```
## starting httpd help server ... done
```

External functions from packages

- Packages developed by the R community
- Use install.packages() to install
- Use library(package name) to load a package/library

```
#Loading dplyr package
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Example: Using Select function from dplyr package

- After loading a library we can use its functions
- Example:
- We can use select() function in dplyr to select elements

```
#Selecting an element using select function in dplyr
select(my_df, my_names)
```

```
##   my_names
## 1     John
## 2    Peter
## 3    James
## 4       Ed
```

Function Errors

- A error will occur if a function has not been used properly
- An error message is usually displayed
- Read error message to find out what caused the error
- Fix the error -If it fails get help from the R community

```
mean(my_names)
```

```
## Warning in mean.default(my_names): argument is not numer
## NA

## [1] NA
```

Where to get help?

- Stackoverflow
- Twitter (use #rstats to ask a question)
- Facebook R groups
- Meet up groups(like ours)

Resources to learn R

- Books
- Youtube videos
- Blogs
- Twitter
- Meetup groups

Conclusion

- We discussed some of the basics of R
- Foundation to start learning R
- You will enjoy using R
- Follow Botswana R Users on Twitter @Botswana_r