AfricaCDC
Centres for Disease Control and Prevention

Safeguarding Africa's Health

Module B05

Bioinformatics Foundational Course

Introduction to R Programming for Data Analysis

NGS Academy for the Africa CDC

ASLM
AFRICAN SOCIETY FOR LABORATORY MEDICINE

H3ABioNet
Pan African Bioinformatics Network for H3Africa

ASLM ACADEMY

Module B05

Introduction to R Programming for Data Analysis

back to the table of modules

Module last updated:
December 2024

Suggested or approximate number of sessions	3-4
Suggested or approximate total learning time	8-10 hours (based on the data carpentry course)
Target audience	Bioinformaticians and IT personnel
Delivery format	Lectures, videos, practicals with data examples
Level of the module	Introductory

Contributors

Hocine Bendou, George Githinji, Shahiid Kiyaga, Tony Yiqun Li, Perceval Maturure and Kennedy Mwai.

Suggested prerequisite module(s)

- Module B01. Introduction to Unix/Linux, Command Line, and Shell Scripting
- Module B02. Introduction to Version Control
- Module B04. Introduction to Programming (for advanced R programming)

Module description

This module is optional and is for bioinformatics users who wish to use R for programming, data analysis and visualization. The module introduces RStudio, a widely used platform for writing R scripts and interacting with the R software. To ensure proper functionality, both R and RStudio need to be installed on the computer. One of the advantages of working with R is the ability to analyze results relying on a series of written commands rather than a sequence of clicks and selections. This aspect proves beneficial when one needs to replicate one's analysis with additional data. Instead of recalling which buttons one clicked in a specific order to obtain one's results, one can simply rerun the stored series of commands in an R script. R will process the new dataset in the same consistent manner as before.

Utilizing scripts in one's analysis offers clarity in the steps undertaken, and the code one writes becomes accessible for review by others who can provide feedback and identify any errors. Working with scripts promotes a deeper understanding of one's analytical processes and enhances one's learning and comprehension of the methods employed. By engaging with scripts, one will gain better insight into the workings of analyses and foster overall expertise. In this module, participants are introduced to the following topics and/or concepts:

- Installing R, Installing RStudio
- Introducing R and RStudio IDE.
- R Basics -introduction to an example dataset and file type.
- R Basics continued -factors and data frames.
- Creating an RStudio project, and features of working within a project
- Customizing the RStudio layout, locating and changing the current working directory with getwd() and setwd()
- Composing an R script file containing comments and commands.
- Understanding what an R function is
- Locating help for an R function using ?, ??, and args()
- Loading a tabular dataset using base R functions
- Determining the structure of a data frame including its dimensions and the datatypes of variables.
- Subsetting/retrieving values from a data frame
- Describing what the Bioconductor repository is and what it is used for.
- Describing how Bioconductor differs from CRAN. Search Bioconductor for relevant packages
- Installing and using packages from Bioconductor.
- Basic programming of instruction blocks in R.
- Data wrangling and analysis with Tidyverse
- Data visualization with ggplot2
- Manipulating a factor, including subsetting and reordering
- Applying an arithmetic function to a data frame
- Coercing the class of an object (including variables in a data frame)
- Importing data from Excel, saving a data frame as a delimited file

Module learning outcomes

On completion of this module, participants will have a basic knowledge of, or will be able to:

- Write effective and efficient R programmes
- Understand the advantages of analyzing data in R and the advantages of using RStudio
- Use the RStudio IDE
- Explain the basic principle of tidy datasets
- Understand how R may coerce data into different modes
- Change the mode of an object
- Extract useful descriptive statistics from real biomedical data
- Use data import functions, create data visualizations with ggplot2
- Perform descriptive statistics in R (summary statistics, probabilities, and random variables)

Module assessments

Module practical: Practical available on the ASLM platform
Module quiz: Assessment questions available on the ASLM platform

Module resources

- The Carpentry | Webpage - Data Analysis and Visualization in R for Ecologists
- Datacarpentry | GitHub - R ecology lesson
- Testbook | Webpage - R Programming MCQ Quiz
- Sanfoundry | Webpage - R Programming MCQ
- Data Flair | Webpage - R Multiple Choice Questions and Answers
- Free Time Learning | Webpage - R Language - Quiz (MCQ)
- The Carpentry | Webpage - Using packages from Bioconductor
- The Carpentry | Webpage - Data Wrangling and Analyses with Tidyverse
- Swirl Stats | Webpage - An R programming teaching platform
- The Carpentry | Webpage - Intro to R and RStudio for Genomics
- Keniajin | GitHub - Introduction to R: Short course
- Keniajin | GitHub: I-StaR
- Applied Epi | Handbook - The Epidemiologist R Handbook

Additional information on RStudio

RStudio is divided into 4 “panes”:
1. The Source for your scripts and documents (top-left, in the default layout)
2. Your Environment/History (top-right) which shows all the objects in your working space (Environment) and your command history (History)
3. Your Files/Plots/Packages/Help/Viewer (bottom-right)
4. The R Console (bottom-left)

The essence of programming lies in providing instructions for the computer to follow, and subsequently instructing the computer to execute those instructions. We use the R language to write these instructions since it serves as a common language that both computers and humans can comprehend. The instructions we write in R are referred to as commands, and we execute (or run) these commands to direct the computer's actions. When working with R, there are two primary methods of interaction: using the console or utilizing script files, which are plain text files containing our code. The console pane, typically located in the bottom left panel of RStudio, serves as a space where R commands can be entered and promptly executed by the computer. It is also the area where the results of executed commands are displayed. While it is possible to directly type commands into the console and press Enter to execute them, it's important to note that these commands will be forgotten once the session is closed.

To ensure reproducibility and maintain a record of our code and workflow, it is advisable to type the desired commands in the script editor and save the script. By doing so, we create a comprehensive record of our actions, enabling anyone (including our future selves) to easily replicate the results on their own computers.

Acknowledgements

We would like to thank the following individuals, in alphabetical order of last name, for their valuable time and effort spent in designing (i.e., drafting, reviewing, and refining) this module: **Hocine Bendou, George Githinji, Shahiid Kiyaga, Tony Yiqun Li, Perceval Maturure and Kennedy Mwai**.

Furthermore, we would like to thank the following institutions, societies, journals and individuals from whom we sourced open-access resources, used in this module:
Applied Epi, Data Flair, Datacarpentry, Free Time Learning, Keniajin GitHub, Sanfoundry, The Carpentry, Testbook; Akamau, Chris Bailey, Neale Batra, Isha Berry, Paula Blomquist, Emma Buajitti, Finlay Campbell, Liza Coyer, Isaac Florence, Natalie Fischer, Sara Hollis, Toby Hodges and co-contributors (of the Datacarpentry GitHub), Yurie Izawa, Henry Laurenson-Schafer, Wen Lin, Daniel Molling, Mathilde Mousset, Ken Mwai, Aminata Ndiaye, Moses Ngari, Dr. Osman, Mark Otiende, Jonathan Polonsky, Alex Spina.