Introduction to R

Data analysis and statistical computing for quantitative social science research

Laboratorio de Investigación para el Desarrollo del Ecuador

Instructor: Daniel Sánchez, MA

Module length: 15 hours

Course level: Intermediate

Prerequisite knowledge: Basic computer skills, Git, shell.

Corequisite knowledge: **Introduction to Statistics** module.

GitHub repository: https://github.com/laboratoriolide/intro-to-r

1 Module overview

This module introduces the R programming language basics for data analysis and statistical computing. This is a companion module to the *Introduction to Statistics* module, which provides the foundations of statistics for social science research. The material is designed for those who aim to use R as a tool for quantitative research methods, such as statistics, econometrics, and (to a lesser degree), data science/machine learning.

I assume little prior coding experience: familiarity with basic computer skills, Git and the command line. Further, the course will be relatively fast-paced, combining self-study with the DataCamp platform and live coding. We will cover many important concepts in the language, however, the focus is not on technical programming but rather on understanding the language as a tool for statistical analysis.

2 Module contents

The following is a planned outline of the course. This may change depending on the pace of the class. It is sometimes expected that students complete tasks before each lecture to make the most of the class time.

- **Lecture 1**: Intro
  - Introduction to the R language: history, uses, advantages and applications in the social sciences.
  - The development environment: R, RStudio, IDEs, and R Packages.
  - The RStudio interface: R scripts, file viewer, console, and environment.
  - Ensuring reproducibility: Filepaths, working directories, and project organization; using R Projects with RStudio.
  - Installing packages, loading libraries; the `install.packages()` and `library()` functions.
  - Basic R syntax: variables, assigners, data types, structures, and (logical) operators.
  - Data frames: definition and basic manipulation with base R syntax.
  - Looking at data: `head()`, `tail()`, `str()`, `summary()`
  - Base R graphics: `plot()`, `hist()`, `boxplot()`, `barplot()`.
  - Getting help in R/RStudio, the internet and AI tools.
  - **Complete before class**: installation of R and RStudio, cloning of the course's GitHub repository and completion of the first DataCamp course, *Introduction to R for Finance.*
- **Lecture 2**: Introduction to the tidyverse and importing data
  - Base R and the *tidyverse*
  - Introduction to the *tidyverse*: the tidy process, tidy data, the pipe (`%>%`) operator
  - Importing data with base R functions
  - Importing data with `readr`, `haven` from different file formats
  - Importing Excel files and preliminary manipulation with `readxl`
  - Downloading data from R packages, GitHub and the web/URLs
  - Tibbles: the tidy data frame; using `glimpse()` to understand data
  - dplyr: Selecting and renaming columns, pipe workflows, `transmute`.

  - Complete before class: DataCamp course *Introduction to importing data in R* and *Introduction to the Tidyverse* (select chapters) and make sure you have access to the module's `data` folder in the GitHub repository.
- **Lecture 3**: More on data manipulation
  - Filtering rows, sorting, grouping, and summarizing data.
  - Using `mutate()` to create new columns: `if_else`, `case_when`, `case_match()`.
  - Joins, semijoins and antijoins: applications for researchers.

  - The `bind_rows()` function for appending data frames.
  - The `across()` function for applying functions to multiple columns.

- **Lecture 4**: Data visualization with ggplot2
  - Introduction to the `ggplot2` package: the grammar of graphics.
  - The `ggplot()` function: aesthetics, geoms, and layers.
  - Scatter plots, line plots, bar plots, histograms, and box plots.
  - Faceting: `facet_wrap()` and `facet_grid()`.
  - Themes and customization: `theme()`, `labs()`, `scale_x_continuous()`, `scale_fill_manual()`.
  - Saving plots, changing plot size, and exporting to different formats.

- **Lecture 5**: Advanced topics in data cleaning
  - Understanding long vs. wide data formats; reshaping data with tidyr.
  - Expanding data: `expand()` and `complete()` functions.
  - Missing data: identifying, handling, and imputing missing values.

- **Lecture 6**: Reporting and reproducibility
  - The concept of reproducible research.
  - Introduction to R Markdown: the R Markdown document, YAML header, code chunks.
  - Knitting documents: HTML, PDF, Word, and slides.
  - Customizing R Markdown documents: themes, templates, and CSS.
  - Quarto: the next generation of R Markdown.
  - Math formulas: incorporating LaTeX into R Markdown.
  - Time-permitting: `.Rnw` files and Knitr/Sweave.
  - **Complete before class**: DataCamp course *Reporting with R Markdown.* Making sure you have LaTeXinstalled in your computer, or TinyTeX as a package in R[1].

- **Lecture 7**: Select topics in R programming
  - Control structures, conditional statements, and logical operators; relationships to prepackaged functions.
  - Functions: when to use, how to define functions, arguments, return values.
  - The `apply` family of functions: `apply()`, `lapply()`, `sapply()`, `mapply()`.
  - Loops: while and for loops, the `map` family of functions.

- **Lecture 8**: Further advanced topics with R
  - Time and date management: `lubridate`.
  - String manipulation with `stringr`.
  - Fuzzy joins: string distance and the `fuzzyjoin` package.

[1] TinyTeX is a lightweight, portable, cross-platform, and easy-to-maintain LaTeX distribution. It is available as a package in R, and can be installed with `tinytex::install_tinytex()`

    - Time-permitting: regex, text analysis.

- **Lecture 9**: High-performance computing in R
    - Why high performancec? The need for speed in data analysis.
    - Loading very large datasets: `data.table` and `fread()`.
    - Reading many files at once: `purrr` and `map()` with `fread()`.
    - Grouping and summarizing large datasets: `dplyr` and `data.table` comparison
    - Other alternatives: bigmemoryr, ff, and SQL databases.
    - Time-permitting: Introduction to parallel computing: the `parallel` package.
- **Lecture 10**: Advanced data access with R
    - Web scraping: the `rvest` package.
    - Accessing twitter data: the `rtweet` package.
    - Accessing financial data: the `quantmod` package.
    - Accessing spatial data: the `sf` package.
    - Time-permiting: APIs, applications with open data.

3 Evaluation

The final grade will be calculated as follows:

Component	Percentage
Attendance	15%
DataCamp assignments	50%
Participation	5%
Task Completion	30%

3.1 Attendance

I do not have any special requirements for attendance other than the requirements set by the program. Consult the program's regulation handbook for more information.

3.2 Assignments

There will be weekly DataCamp assignments that reinforce the material taught in this module as well as that of the companion module *Intro to Statistics* module. These will be either DataCamp courses, projects or exercises, and are graded on a pass/fail basis. Please check the DataCamp platform for the due dates of each assignment.

Apart from the DataCamp assignments, you may need to complete some self-study or autonomous practice work **before** each lecture. This is necessary to make the most of the class time, as programming is a skill that requires practice. Please see above for all tasks to be completed before each lecture.

3.2.1 Tasks

Tasks are small exercises or activities that are due before each lecture. They may be small coding exercises, readings or installations that you **must** complete before class. These will be graded on a pass/fail basis. We may discuss these tasks in class or in the Slack channel, so it is important that you complete them on time.

3.2.2 Tentative DataCamp assignments

This is a list of the DataCamp courses that will be assigned as part of the module. This list may change, so please check the DataCamp platform for the most up-to-date information.

- Introduction to R for Finance
- Introduction to Importing data in R
- Introduction to the Tidyverse
- Data Manipulation with dplyr
- Data Visualization with ggplot2
- Reporting with R Markdown
- Reshaping Data with tidyr (select chapters)
- Intermediate R for Finance

4 Course materials

All course materials will be provided in the course's GitHub repository. This includes lecture slides, readings, datasets, assignments and any other relevant material. The repository will be updated regularly, so please check it often for new material. I recommend using a Git client, such as GitHub Desktop or GitKraken, to keep your local repository up-to-date.

4.1 Textbook

There is no required textbook for this course, as I will provide slides for all lectures. However, I recommend the following books for those who want to delve deeper into the material. These were used as references for the course.

- *R for Data Science*, @wickham-grolemund[2]
- The Effect, @huntington-klein22[3].

5 Software

This course requires the installation of R and RStudio. Both are free and open-source software, and are available for Windows, macOS, and Linux. You will need to install these **before** the first lecture.

- **R**: Download the latest version of R from the Comprehensive R Archive Network (CRAN) at https://cran.r-project.org/. Follow the instructions for your operating system.
- **RStudio**: Download the latest version of RStudio Desktop from here. Follow the instructions for your operating system.

If you are having trouble installing, please post your query in the Slack channel.

You will also need Git to be able to access the course's GitHub repository. Follow the instructions from the Git & GitHub short module to make sure this software is configured correctly in your computer. As mentioned, a Git client is recommended to keep your local repository up-to-date and easily accesible to not miss any updates. For Windows users, we may use Git Bash sometimes in the course, so make sure you are familiar with the command line. Mac or Linux users can use the native terminal.

6 Communication

All communications to the instructor should be made through the course's Slack channel. We hope to respond to questions within 72 hours, but please be patient if we take longer. I do not monitor email regularly, so please use Slack for all communications if you need a timely response.

7 References

[2] This book is a comprehensive introduction to the R language, focusing on the tidyverse packages.0

[3] This textbook, while focused on causality, describes much of the modern R development environment for statistics and econometrics. It also contains Stata and Python code.