Kingsley Okoye
Samira Hosseini

R Programming

Statistical Data Analysis in Research

Springer

R Programming

Kingsley Okoye · Samira Hosseini

R Programming

Statistical Data Analysis in Research

Springer

Kingsley Okoye
Department of Computer Science
School of Engineering and Sciences
and Institute for the Future of Education
Tecnológico de Monterrey
Monterrey, Nuevo Leon, Mexico

Samira Hosseini
School of Engineering and Sciences
and Institute for the Future of Education
Tecnológico de Monterrey
Monterrey, Nuevo Leon, Mexico

ISBN 978-981-97-3384-2 ISBN 978-981-97-3385-9 (eBook)
https://doi.org/10.1007/978-981-97-3385-9

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

The goal of scientific research is often to investigate a specific phenomenon or topic and find relationships that exist between the underlying variables or factors within the subject population to be able to draw conclusions. The *statistical analysis* is an important part of the scientific research, particularly in the social sciences. It involves the process of collecting and analyzing different data samples in order to identify patterns or trends that can be used to provide valuable insights into the research or draw conclusions. The procedure for performing the various statistical operations and data scrutiny is called Statistical Data Analysis. The researchers can use this procedure to test different hypotheses and make estimations about the studied populations.

This book is written for statisticians, data analysts, programmers, researchers, teachers, students, professionals, and general consumers on how to perform different types of statistical data analysis for research purposes using the R programming language. R is an open-source software and object-oriented programming language with an integrated development environment (IDE) called RStudio for computing statistics and graphical displays through data manipulation, modeling, and calculation. R packages and supported libraries provide the users with a wide range of functions for programming and analyzing of data. Unlike many of the existing statistical softwares, R has the added benefit of allowing the users to write more efficient codes by using command-line scripting and vectors. It has several built-in functions and libraries that are extensible and allow the users to define their own (customized) functions on how they expect the program to behave while handling the data, which can also be stored in the simple object system.

For all intents and purposes, this book serves as both textbook and manual for R statistics, particularly in academic research, data analytics, and computer programming targeted to help inform and guide the work of the R users or statisticians. It provides information about different types of statistical data analysis and methods, and the best scenarios for use of each case in R. It gives a hands-on step-by-step practical guide on how to identify and conduct the different parametric and non-parametric procedures vastly used in social science research. This includes a description of the different conditions or assumptions that are necessary for performing the various statistical tests, and how to understand the results of the different methods. The book also covers different data formats and sources, and how to test for reliability and validity of the available datasets used for research purposes. Different research experiments, case scenarios, and examples are explained in this book. It is the first book to provide a comprehensive description and step-by-step practical hands-on guide on how to carry out the different types of statistical analysis in R, particularly for research purposes with examples. Ranging from how to import and store datasets in R as objects, how to code and call the R methods and functions for manipulating the datasets and objects, factorization, and vectorization, to better reasoning, interpretation, and storage of the results for future use, and graphical visualizations and representations. Thus, the congruence of Statistics and Computer programming for Scientific Research purposes.

Structure and Organization

The content of this book is organized into 13 chapters that are subdivided (classified) into two parts: Part I and Part II.

The chapters in Part I—which covers Chaps. 1–6 focus on introducing the readers to the basic concepts of R programming and how to use data in R, particularly for research purposes. This includes introducing the readers to the norm of scientific research and the different elements or components that form a typical research process. This part of the book is especially intended for readers who are new to the topic or require a refresher on their knowledge of the R topic or understanding of the different statistical methods and analysis in scientific research. The chapters in Part I prepare the users for easy and efficient use and application of the different statistical methods and analysis that are written in the remainder of chapters in Part II of the book.

The chapters in Part II—which covers Chaps. 7–13 focus on the practical implementation of the different types of statistical data analysis for research using R. This part is meant for readers who have gained advanced knowledge of the topic, and may have the need for applying the various illustrated methods for their research use or data analysis problems.

The content of the book has been structured in a way that it gives a hands-on step-by-step practical guide to the users on how to identify and conduct the different statistical tests and procedures for their research purpose. The users can make references or choose to skip to the specific methods or chapters of interest based on their expert or individual needs.

The following is a brief description of each of the chapters in this book:

Part I

Chapter 1 presents an introduction to the R programming language and RStudio software particularly for conducting statistical data analysis, graphical displays, modeling, and calculations. It covers the basic concept of R programming, and how the readers can be able to install and run their first R project.

Chapter 2 explores the basic principles and concepts of data management and manipulation in R by discussing what are R objects, vectors, packages, and libraries, including graphs and data visualization methods using the RStudio IDE. It introduces the users to the different functions and methods for working with data in R.

Chapter 3 introduces the readers to the main tests of data normality and reliability using R. The most commonly used and frequently applied type of methods for research are described and illustrated in detail in this chapter.

Chapter 4 explains the Parametric and Non-Parametric Tests for statistical data analysis, and the best scenarios for the use of each test. The chapter provides a guide for the readers on how to choose which test is most suitable for their specific research, including a description of the differences, advantages, and disadvantages of using the two types of tests.

Chapter 5 describes what are *dependent* and *independent* variables for conducting research experiments. It introduces the readers to the different conditions for the use of the two types of variables in scientific research. The differences between the two variables (independent versus dependent) and examples of each use case scenario are also provided in this chapter.

Chapter 6 provides the readers with information about the various types of statistical data analysis methods used in scientific research and examples of best scenarios for the use of each method.

Part II

Chapter 7 introduces and explains to the readers how to run a *linear* and *logistic* regression analysis in R using RStudio. This statistical technique helps to estimate the association or dependency of the relationship between two variables.

Chapter 8 provides the readers with guidelines on how to run the T-tests for evaluating the *mean* of one or two groups of variables using R. The Independent samples, Paired sample, and One sample t-tests are explained and practically illustrated in this chapter.

Chapter 9 provides detailed information on how to run the analysis of variance (ANOVA) test in R. This test helps to determine the *mean differences* that may exist in data samples. The One-way and Two-way ANOVA are explained and practically illustrated in this chapter.

Chapter 10 explains and illustrates how to apply a Chi-squared (X^2) analysis in R. The test is used to compare how expectations are linked with the actual observed experimental data.

Chapter 11 explains and demonstrates how to analyze the effect of a variable over another using the "Mann-Whitney U" and "Kruskal-Wallis H" tests. These are non-parametric equivalents and alternatives to the Independent t-tests and ANOVA used for non-normally distributed dataset in the nominal or ordinal scales, otherwise referred to as *distribution-free* tests.

Chapter 12 explains and illustrates how to conduct the three primary correlational analyses in R which includes: Pearson cor, Kendall's tau, and Spearman's rho correlation tests.

Chapter 13 explains and demonstrates how to perform the Wilcoxon test in R. This distribution-free test which is of two types (Signed-Rank and Sum-rank) and assumes that the data comes from two matched or dependent populations, are practically illustrated in this chapter.

Data Availability: Link to the different example datasets used in the practical illustrations and statistical data analysis and computations in this book have been provided in the individual chapters where each of the specific datasets is used. In addition, the authors have uploaded the list of example datasets to the following repository: https://doi.org/10.6084/m9.figshare.24728073 for easy access and download for use and practice by the readers.

Monterrey, Mexico

Kingsley Okoye
Samira Hosseini

Acknowledgments

This work would not have become a successful project without the contributions from different people. Writing this book required sacrificing hundreds of hours of both learning and teaching, away from family and friends, and the authors would like to thank each and every person, colleagues, and departments for their wonderful support in the publication of this book.

First and foremost, we would like to thank the Writing Lab, Institute for Future of Education, Tecnologico de Monterrey for the technical and financial support in the publication of this book. Most especially, we would like to thank the editor, Daniel Wetta and the Research Assistants for their professional work in editing the different chapters that are contained in this book and for taking the time to read through the series of chapters and the entire manuscripts for proof-reading, constructive comments, and suggestions.

We would like to also acknowledge the expert contributions from colleagues, mentees, and professors both within and outside the Tecnologico de Monterrey with whom we have discussed some of the draft chapters and series of handouts during the workshops and seminars.

We would also like to express our sincere gratitude to the Institute for Future of Education and the staffs who have provided the authors with the facilities and sophisticated support in both learning and working space and sourcing of some of the materials the authors have used in this book.

The author's gratitude and acknowledgment will not be complete without expressing my sincere thanks for the understanding, support, and encouragement I have received from both family and friends. I would most especially like to thank my wife, Justina for her patience and unwavering assistance during the preparation of this book and the final manuscripts.

Contents

Part I Fundamental Concepts of R Programming and Statistical Data Analysis in Research

1 Introduction to R Programming and RStudio Integrated Development Environment (IDE) . . . 3
1.1 What is R Programming Language? . . . 3
1.2 RStudio Integrated Development Environment (IDE) . . . 5
1.3 Installing and Configuring R and RStudio Software . . . 6
1.3.1 Downloading and Installing R Language . . . 6
1.3.2 Downloading and Installing RStudio Software . . . 9
1.4 Running Your First R Project in R Using RStudio . . . 14
1.5 Tips and Technical Guidelines . . . 20
1.5.1 Tips About a New R Project . . . 21
1.5.2 Opening Existing R Projects and R Scripts . . . 22
1.5.3 Working with Multiple R Projects . . . 23
1.5.4 Closing or Quitting R . . . 23
1.6 Summary . . . 23
References . . . 24

2 Working with Data in R: Objects, Vectors, Factors, Packages and Libraries, and Data Visualization . . . 25
2.1 Introduction . . . 25
2.2 Preparing RStudio and Script for Working with Data in R . . . 25
2.3 Working with Data in R . . . 28
2.3.1 Pre-loaded Sample Data in R . . . 28
2.3.2 Creating Your Own Data in R . . . 30
2.3.3 Import and Using External Data in R . . . 35
2.4 R Objects . . . 44
2.5 R Vectors: Vectorization and Factorization . . . 46
2.5.1 Creating and Working with Vectors in R . . . 47
2.5.2 Understanding Sequence in Vectors . . . 49
2.5.3 Extracting and Replacing Elements in Vectors . . . 50

2.5.4 Vectorization in R 53
2.5.5 Factorization in R 54
2.6 R Packages and Libraries 56
2.7 Plots and Data Visualization 60
2.8 Summary 65
References 66

3 Test of Normality and Reliability of Data in R 67
3.1 Introduction 67
3.2 Test of Data Normality in R: *Kolmogorov–Smirnov (K-S) and Shapiro–Wilk (S-W) Test* 68
3.3 Test of Data Reliability in R: Cronbach's Alpha Test 79
3.4 Summary 83
References 85

4 Choosing Between Parametric and Non-parametric Tests in Statistical Data Analysis 87
4.1 Introduction 87
4.2 Parametric Versus Non-parametric Tests 88
4.2.1 Parametric Test 88
4.2.2 Non-parametric Tests 90
4.3 Choosing Between Parametric and Non-parametric Test 91
4.3.1 Types of Parametric Versus Non-parametric Tests in Statistical Analysis 93
4.3.2 Examples and Use Case Scenarios: Parametric Versus Non-parametric Tests 93
4.4 Differences Between Parametric Versus Non-parametric Tests 95
4.5 Advantages and Disadvantages of Parametric Versus Non-parametric Tests 95
4.6 Summary 95
References 97

5 Understanding Dependent and Independent Variables in Research Experiments and Hypothesis Testing 99
5.1 What Are Variables in Scientific Research? 99
5.1.1 Types of Variables in Scientific Research 101
5.1.2 Examples and Use Case Scenarios of Independent Versus Dependent Variables 104
5.2 Summary 105
References 106

6 Understanding the Different Types of Statistical Data Analysis and Methods 109
6.1 Introduction to Statistical Data Analysis 109
6.2 Statistical Data Analysis and Methods in Scientific Research . . . 109
6.2.1 Linear Regression 110

6.2.2 Logistic Regression 110
6.2.3 Linear-Log Model 112
6.2.4 T-test 113
6.2.5 Analysis of Variance—ANOVA (F-test) 114
6.2.6 Mann–Whitney U Test 117
6.2.7 Chi-Squared (χ^2) 118
6.2.8 Kruskal–Wallis H Test 119
6.2.9 Correlation 120
6.2.10 Wilcoxon Test (Signed-Rank and Rank-Sum) 122
6.3 Summary 124
References 126

Part II Application and Implementation of Advanced Methods for Statistical Data Analysis in Research Using R

7 Regression Analysis in R: Linear Regression and Logistic Regression 131
7.1 Introduction to Regression Analysis 131
7.2 Linear Regression Analysis in R: Simple Regression and Multiple Regression 133
7.3 Logistic Regression Analysis in R 143
7.4 Summary 157
References 158

8 T-test Statistics in R: Independent Samples, Paired Sample, and One Sample T-tests 159
8.1 Introduction 159
8.2 Independent Samples T-test in R 160
8.3 Paired (Dependent) Sample T-test in R 171
8.4 One Sample T-test in R 178
8.5 Summary 185
References 186

9 Analysis of Variance (ANOVA) in R: One-Way and Two-Way ANOVA 187
9.1 Introduction 187
9.2 One-Way ANOVA Test in R 189
9.3 Two-Way ANOVA Test in R 198
9.4 Summary 207
References 208

10 Chi-Squared (X^2) Statistical Test in R 211
10.1 Introduction 211
10.2 Chi-Squared (X^2) Test in R 212
10.3 Conclusion 222
References 222

11 Mann–Whitney U Test and Kruskal–Wallis H Test Statistics in R 225
11.1 Introduction 225
11.2 Mann–Whitney U Test in R 228
11.3 Kruskal–Wallis H Test in R 235
11.4 Summary 244
References 245

12 Correlation Tests in R: Pearson Cor, Kendall's Tau, and Spearman's Rho 247
12.1 Introduction 247
12.2 Pearson Correlation Test in R 250
12.3 Kendall's Tau and Spearman's Rho Correlation Tests in R 261
12.4 Summary 276
References 277

13 Wilcoxon Statistics in R: Signed-Rank Test and Rank-Sum Test . . . 279
13.1 Introduction 279
13.2 Signed-Rank Wilcoxon Test in R 281
13.3 Rank-Sum Wilcoxon Test in R 292
13.4 Summary 302
References 303

Epilogue and Conclusion 305

Index 307

Abbreviations

ANCOVA	Analysis of Co-variance
ANOVA	Analysis of Variance
Chr	Character
Cor	Correlation
CSV	Comma-Separated Values
DV	Dependent Variable
EFA	Exploratory Factor Analysis
FTP	File Transfer Protocol
GG	Grammar of Graphics
ggplot	Grammar of Graphics Plot
GUI	Graphical User Interface
IDE	Integrated Development Environment
IV	Independent Variable
K-S	Kolmogorov-Smirnov
MANCOVA	Multivariate Analysis of Co-variance
MANOVA	Multivariate Analysis of Variance
MSE	Mean Sum of squares due to Error
MST	Mean Sum of squares due to Treatment
NCA	Necessary Condition Analysis
Num	Numeric
OLS	Ordinary Least Square
OS	Operating System
PCA	Principal Component Factor Analysis
Q-Q plot	Quantile-Quantile Plot
S-W	Shapiro-Wilk

Part I
Fundamental Concepts of R Programming and Statistical Data Analysis in Research

Chapter 1
Introduction to R Programming and RStudio Integrated Development Environment (IDE)

1.1 What is R Programming Language?

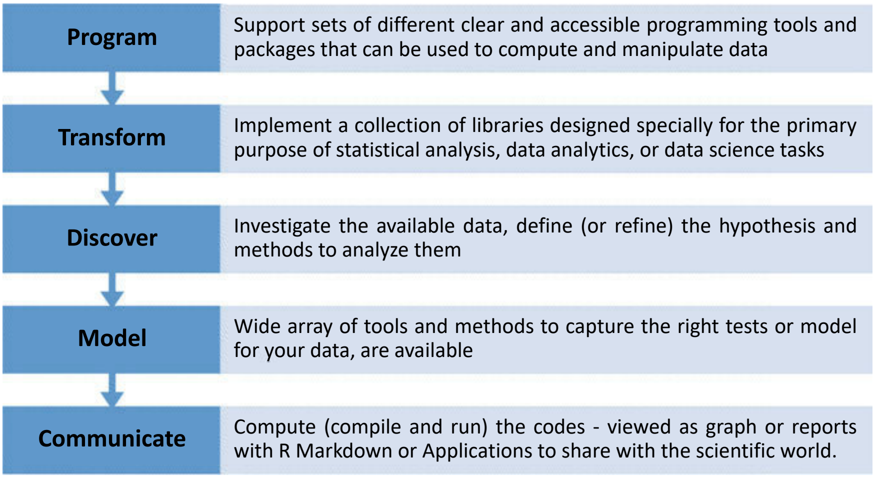
R is an open-source software or programming language for computing statistics and graphical displays through methods such as data manipulation, modeling, and calculation (Ihaka & Gentleman, 1996; Venables et al., 2020). In theory, it is a programming language developed by Ross Ihaka and Robert Gentleman in 1993 (Ihaka & Gentleman, 1996), and is regarded as an implementation of the S and S-Plus language that was originally developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks (Becker et al., 1988). Practically, R packages and the several supported methods are available and are implemented using integrated developments environment (IDE) such as the RStudio (see Sect. 1.2). Technically, R provides a wide range of statistical and graphical techniques for programming and modeling: ranging from linear and nonlinear modeling techniques to statistical tests and analysis, predictive modeling such as clustering and classification, and time series analysis, etc.

For research and data analytics purposes (see Fig. 1.1), R programming and statistics (R Project, 2023) are performed and used in a series of steps that includes programming, transforming, discovering, modeling, and communication of the outputs or results (Wickham & Grolemund, 2017).

Whilst many existing statistical software, such as SAS (SAS Institute, 2023) and SPSS (IBM, 2022), provide the researchers or data scientists with a bounteous output from conducting the different statistical analyses or methods, R tends to give the analysts a minimal output by storing the results of the methods in an apt "object" for further functions or interrogations.

Therefore, with R being an "object-oriented programming language" (Mailund, 2017), the intermediate results are stored in *objects* that can be recalled, re-used, or manipulated by running other pre-defined functions or codes by pointing to the "object name". In other terms, R is a functional and object-oriented programming

K. Okoye and S. Hosseini, *R Programming*,
https://doi.org/10.1007/978-981-97-3385-9_1



Support sets of different clear and accessible programming tools and packages that can be used to compute and manipulate data

Implement a collection of libraries designed specially for the primary purpose of statistical analysis, data analytics, or data science tasks

Investigate the available data, define (or refine) the hypothesis and methods to analyze them

Wide array of tools and methods to capture the right tests or model for your data, are available

Compute (compile and run) the codes - viewed as graph or reports with R Markdown or Applications to share with the scientific world.

Fig. 1.1 Conceptual overview of steps to using R programming for statistical data analysis in research

used to analyze datasets by running mathematical simulations, rearranging complex datasets into simpler and more useful formats and functions, etc. (Matloff, 2011).

Among the many benefits of R in comparison to the other statistical tools and software includes (Douglas et al., 2020; Matloff, 2011):

1. The capacity to write more efficient code using parallel method or vectorization, because of its programmable integrated environment that uses command-line scripting.
2. The capability to define and customize the functions or codes (e.g., how the analysts expect the resultant models to behave) upon handling of the data. R has several built-in functions and libraries that are extendable (extensible) and allow the users to define their own (customized) functions or methods that can be stored in the simple object system.
3. The capability to create artful (illustrative) graphs to visualize or have a conceptual overview of complex data characteristics and functions.
4. The capacity to interface R language with other programs or softwares (e.g., C/C++, Tableau, Python) for improved and better functionality or speed of data analysis.
5. The capacity to find different packages that can be used to perform image manipulation, textual data analysis or natural language processing, machine learning and classifications, etc.
6. Troubleshooting of bugs (code) with an advanced level of debugging performance.

Other advantages of using R particularly as it concerns its technicality or conducting the different statistical data analysis discussed in this book, include (Venables et al., 2020):

- It is built on a well-developed, simple, and effective programming language called "S and S-Plus" that supports user-defined recursive functions and conditionals loops.
- It consists of an effective data handling and storage facility with a wide range of coherent and integrated collections of intermediate tools (packages) and suite operators for statistical data analysis and calculating arrays and matrices.
- It supports different graphical facilities or functions for data manipulation and displays, either directly on the computer screen or storage as soft copy and hard copy on the machine.

However, just like every other programming language, aside from the statistical power of the software, R has its own sets of limitations. R can be daunting for first-time users and people who do not have prior programming knowledge or experience may find it difficult to use the software. Not necessarily because it is more difficult than other programming languages, but because the syntax is different from that of the many other existing languages. Also, R-supported methods or algorithms are spread across different packages, and in consequence, users with no prior knowledge of some of the packages might find it hard to implement the specific methods or algorithms. Thus, the authors has provided in the first part (PART I) of the book, the fundamental concepts of the R programming and statistical data analysis to guide the work of the readers. In addition, R commands give minimal consideration to the computer memory and management and use a lot of the computer physical memory to store the results of the methods as objects, which may be different from some of the other programming languages like Python (Python Software Foundation, 2023). Thus, it uses more computational power and memory. But, R is a continuously evolving language with newer versions and functionalities being developed, and therefore, much of the limitations will eventually fade away with fresher versions and future updates.

1.2 RStudio Integrated Development Environment (IDE)

RStudio is a "friendly front end to R language". It allows the researchers and data scientists to practically implement the R packages, methods, and run the several lines of codes. By definition, "RStudio" is an Integrated development environment (IDE) designed to help researchers and analysts to be more efficient and productive with R (Rstudio, 2023). Typically, RStudio consists of a console, syntax-highlighting editor for direct code execution, and several sophisticated tools and functions for viewing the data history, visualization, troubleshooting of codes, and managing of the project workspace as the authors discuss more in detail later in this chapter.

Just like R programming language, RStudio is free and open source (Rstudio, 2023). Its graphical user interface (GUI) is logically systematized in a way that allows the users to clearly view the data tables and graphs, the source codes, and output/results of the codes, simultaneously.

The RStudio IDE offers the users with Import-Wizard features for importing files of different formats into the environment, e.g., comma-separated values (*.csv), Excel (*.xlsl), SAS (*.sas), SPSS (*.sav), and Stata (*.dta) file formats without having to write the codes. Also, just like many of the existing IDEs or GUIs that are used to execute different programming languages, RStudio has windows with multiple tabs, drop-down menus, and many customization options. And, it is available for Windows, Macintosh, and Linux operating systems (OS) (Rstudio, 2023).

Among the many features and functionalities of RStudio IDE includes:

- a window that allows the user to write codes and view the results in real time.
- navigate through the files on the local machine or computer.
- check the details and history of the imported/analyzed data and variables.
- visualization of the results and plots (graphs, models) that are generated.

The RStudio IDE can also be used for developing packages, modeling and writing of executable applications, and natural language and machine learning techniques, etc.

It is important when working with R to know the difference between the "R language" and "RStudio" as covered already in this chapter (see Sects. 1.1 and 1.2). It is noteworthy, to always keep in mind that while "R" is a programming language that can practically be used to statistically compute and manipulate the different variables (data) and models. On the other hand, "RStudio" makes use of R language to develop and show the statistical programs/outputs. Thus, "RStudio allows the users to develop and edit programs using R". Interestingly, R can be used without RStudio, but RStudio cannot be used without R. The user or researcher must first install R before they can install RStudio on their computer, as the authors cover in the next section of this chapter.

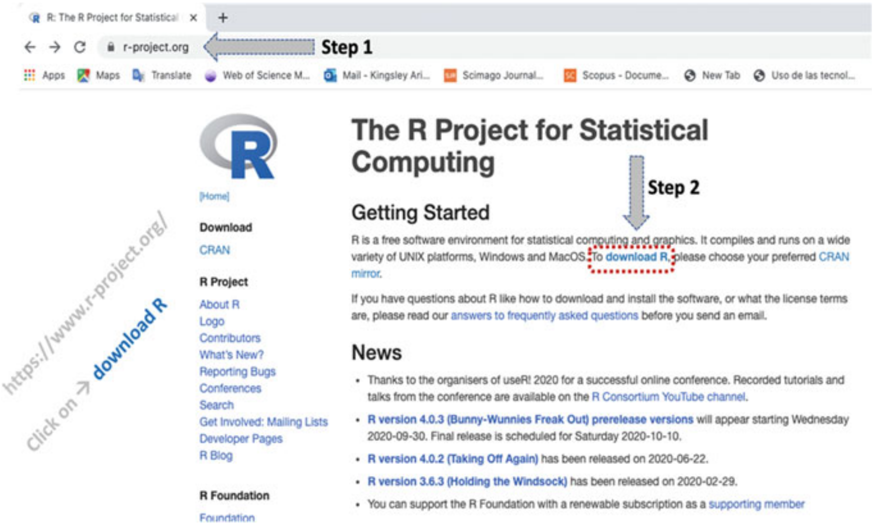
1.3 Installing and Configuring R and RStudio Software

This section of the chapter covers the different steps on how to download, install, and configure R and RStudio before using it for statistical data analysis or research purpose.

1.3.1 Downloading and Installing R Language

Installing R on the computer is very simple and easy. All the user need is to know which operating system (OS) they are using so that they can download the right software for installation on the computer system.

The official site for downloading the R free software is via the following link: https://www.r-project.org/.



The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To download R, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- Thanks to the organisers of useR! 2020 for a successful online conference. Recorded tutorials and talks from the conference are available on the R Consortium YouTube channel.
- **R version 4.0.3 (Bunny-Wunnies Freak Out) prerelease versions** will appear starting Wednesday 2020-09-30. Final release is scheduled for Saturday 2020-10-10.
- **R version 4.0.2 (Taking Off Again)** has been released on 2020-06-22.
- **R version 3.6.3 (Holding the Windsock)** has been released on 2020-02-29.
- You can support the R Foundation with a renewable subscription as a supporting member

Fig. 1.2 Downloading R software

When you visit the site, you will find different binary files for the different types of operating systems (OS) that the R software support, particularly the most common: **Windows**, **Mac OS**, and **Linux**. The latest versions of Linux distributions come with R by default. But for Windows and Mac OS, the user will need to download and install the software as follows:

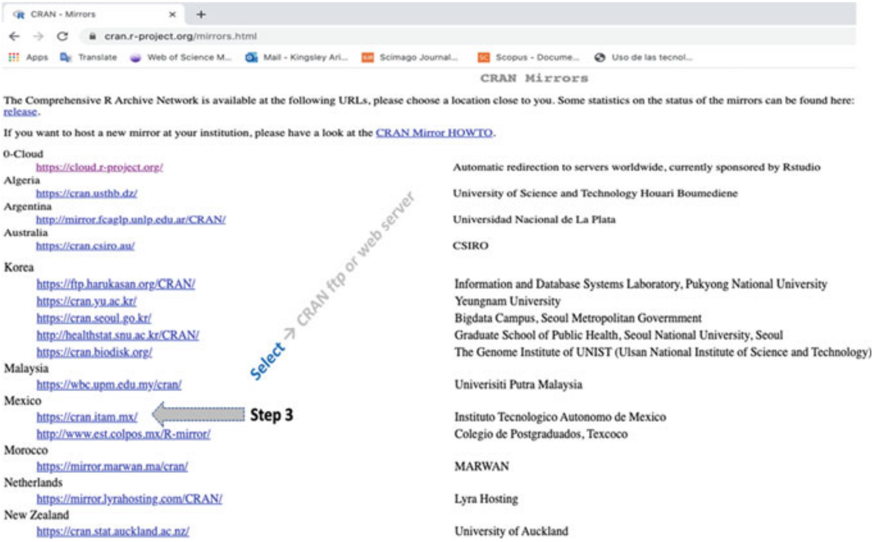
Go to https://www.r-project.org/ by entering the URL on the web browser and click on “**download R”** as shown in Fig. 1.2.

When the user selects **“download R”**, they will be automatically directed to another page where they will be asked to select the **Country** from which they will be using R software, or yet the closest location to you if the country of location is not listed on the site. Navigate to the country of choice and click on any of the **CRAN links** under the country to proceed with the download process. CRAN is a network of *ftp* (file transfer protocol) and web servers around the world that store identical, up-to-date, versions of code and documentation for R.

For example, as shown in Fig. 1.3, the user can navigate to Mexico (e.g., the authors of this book are affiliated with the country at the time of writing this book) and select https://cran.itam.mx/ to proceed with the downloading process. ***Note, it is recommended to always choose the closest CRAN link to you upon downloading the R software.

When the user clicks on the R CRAN binary distribution of their choice, they will be directed to a page where they can download the ***right version of R*** for the specific **operating system (OS)** they are using on the computer (see Fig. 1.4).

For example, as shown in Fig. 1.5 (or step 5), when the users click on **Download R for (Mac) OS X**, they will be directed to where they can then download the **latest**



CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: release.

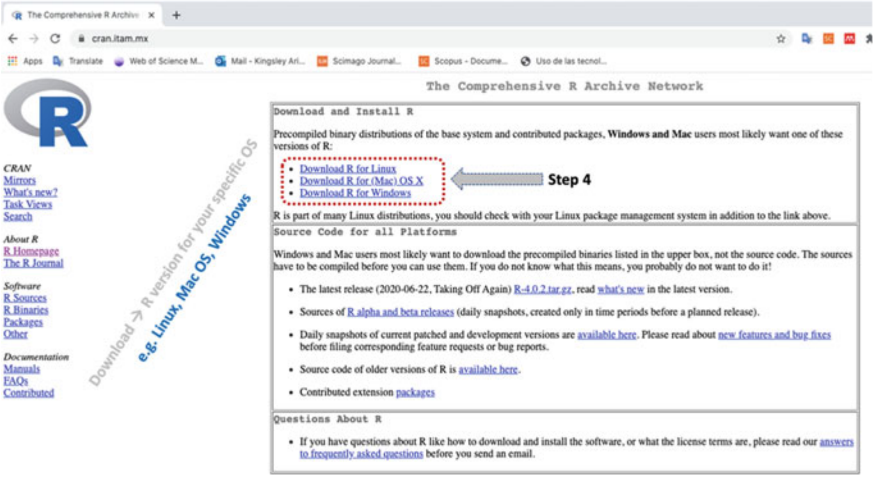
If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

0-Cloud	https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria	https://cran.usthb.dz/	University of Science and Technology Houari Boumediene
Argentina	http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	https://cran.csiro.au/	CSIRO
Korea	https://ftp.harukasan.org/CRAN/	Information and Database Systems Laboratory, Pukyong National University
	https://cran.yu.ac.kr/	Yeungnam University
	https://cran.seoul.go.kr/	Bigdata Campus, Seoul Metropolitan Govermment
	http://healthstat.snu.ac.kr/CRAN/	Graduate School of Public Health, Seoul National University, Seoul
	https://cran.biodisk.org/	The Genome Institute of UNIST (Ulsan National Institute of Science and Technology)
Malaysia	https://wbc.upm.edu.my/cran/	Univerisiti Putra Malaysia
Mexico	https://cran.itam.mx/	Instituto Tecnologico Autonomo de Mexico
	http://www.est.colpos.mx/R-mirror/	Colegio de Postgraduados, Texcoco
Morocco	https://mirror.marwan.ma/cran/	MARWAN
Netherlands	https://mirror.lyrahosting.com/CRAN/	Lyra Hosting
New Zealand	https://cran.stat.auckland.ac.nz/	University of Auckland

Select → CRAN ftp or web server

Step 3

Fig. 1.3 Country of location and nearest CRAN network for R download



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux
- Download R for (Mac) OS X
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2020-06-22, Taking Off Again) R-4.0.2.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Download → R version for your specific OS

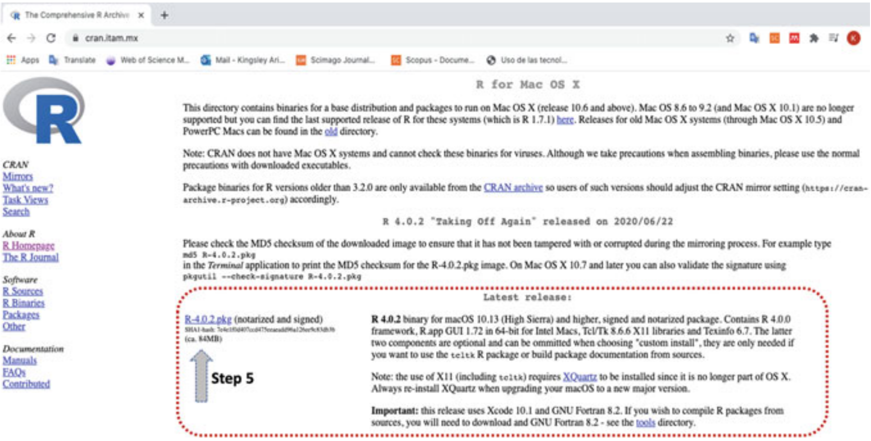
e.g. Linux, Mac OS, Windows

Step 4

Fig. 1.4 Downloading the right version of R for your operating system (OS)

version of R for Mac OS X. Same applies to the other types of operating systems (OS) such as Windows, if you are using the Windows operating system.

When the user clicks on the download link, the executable program (i.e., installation file) will be automatically downloaded on the computer. Navigate to the location where the downloaded file is stored on the computer and install it as every other application program, e.g., by double-clicking on the downloaded file. When you



R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) here. Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the old directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

Package binaries for R versions older than 3.2.0 are only available from the CRAN archive so users of such versions should adjust the CRAN mirror setting (`https://cran-archive.r-project.org`) accordingly.

R 4.0.2 "Taking Off Again" released on 2020/06/22

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type
`md5 R-4.0.2.pkg`
in the *Terminal* application to print the MD5 checksum for the R-4.0.2.pkg image. On Mac OS X 10.7 and later you can also validate the signature using
`pkgutil --check-signature R-4.0.2.pkg`

CRAN
Mirrors
What's new?
Task Views
Search

About R
R Homepage
The R Journal

Software
R Sources
R Binaries
Packages
Other

Documentation
Manuals
FAQs
Contributed

Latest release:

R-4.0.2.pkg (notarized and signed)
SHA1-hash: [illegible]
(ca. 84MB)

R 4.0.2 binary for macOS 10.13 (High Sierra) and higher, signed and notarized package. Contains R 4.0.0 framework, R.app GUI 1.72 in 64-bit for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 6.7. The latter two components are optional and can be ommitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

Note: the use of X11 (including `tcltk`) requires XQuartz to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your macOS to a new major version.

Important: this release uses Xcode 10.1 and GNU Fortran 8.2. If you wish to compile R packages from sources, you will need to download and GNU Fortran 8.2 - see the tools directory.

Fig. 1.5 Downloading the latest release of R for your operating system (OS)

double-click on the file, you will get a pop-up window as shown in Fig. 1.6a. Follow the steps illustrated in the figures (Fig. 1.6a, b, and c) by Clicking on **Continue** until you see the window that says you have **successfully installed R software**.

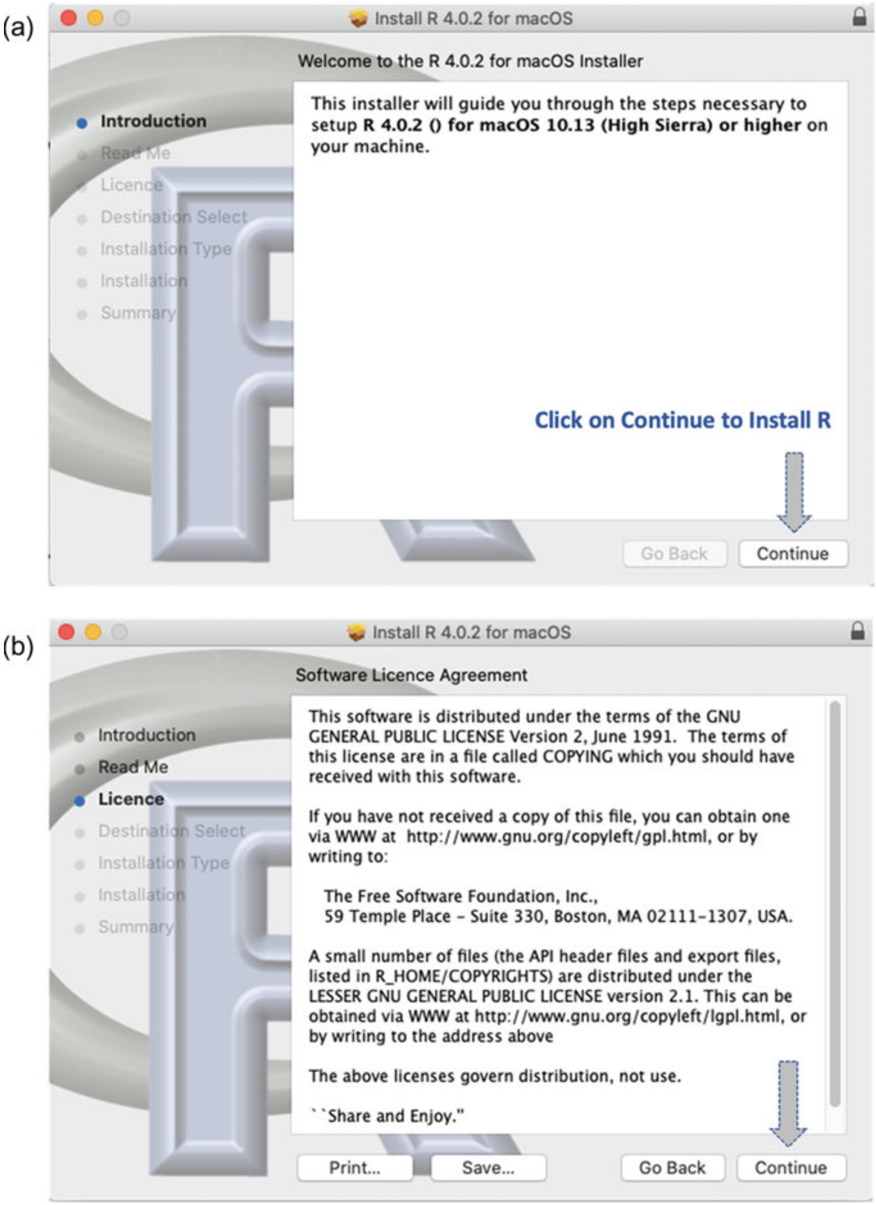
1.3.2 Downloading and Installing RStudio Software

The next step after installing R on your computer, is to download and install the **RStudio IDE** that allows the users to use R.

The official site for downloading the RStudio free software is via the following link: https://rstudio.com/ or https://posit.co/.

As shown in Fig. 1.7, when you visit the RStudio website, you will find the **download** link where the user can download the latest version of RStudio for their computer operating system (OS). ***Note that all the companies update their websites every now and then, and therefore, it may be likely that you find a different front-end display different from the one in Fig. 1.7, which the company uses at the time of writing this book. If you happen to find an updated website depending on when the reader is reading or using this book guide, just simply find where the *download* link is located on the website and follow the same steps or procedure discussed in this current chapter.

Click on the **"DOWNLOAD"** menu (Fig. 1.7), and you will be directed to a page where you can download the RStudio software. Select the "**Download"** link for the "**free version of RStudio Desktop"** as shown in Fig. 1.8. Again, it is important to note that the web display is based on the time of writing this book. As you can see in the figure (Fig. 1.8), there are also paid versions of the software, but those are not covered in this topic.



This installer will guide you through the steps necessary to setup **R 4.0.2 () for macOS 10.13 (High Sierra) or higher** on your machine.

This software is distributed under the terms of the GNU GENERAL PUBLIC LICENSE Version 2, June 1991. The terms of this license are in a file called COPYING which you should have received with this software.

If you have not received a copy of this file, you can obtain one via WWW at http://www.gnu.org/copyleft/gpl.html, or by writing to:

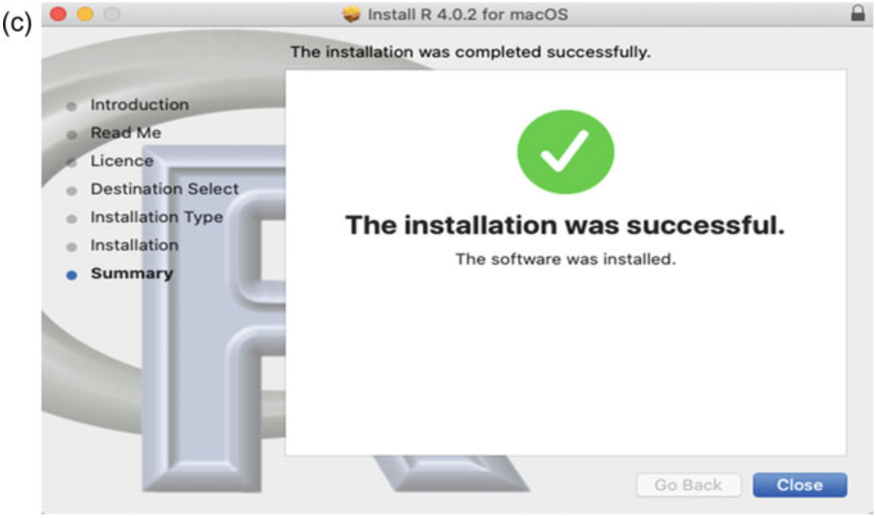
The Free Software Foundation, Inc.,
59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

A small number of files (the API header files and export files, listed in R_HOME/COPYRIGHTS) are distributed under the LESSER GNU GENERAL PUBLIC LICENSE version 2.1. This can be obtained via WWW at http://www.gnu.org/copyleft/lgpl.html, or by writing to the address above

The above licenses govern distribution, not use.

``Share and Enjoy."

Fig. 1.6 **a** Installing R on the computer or local machine (step 1). **b** Installing R on the computer or local machine (step 2). **c** Successfully installing R on the computer or local machine (step 3)

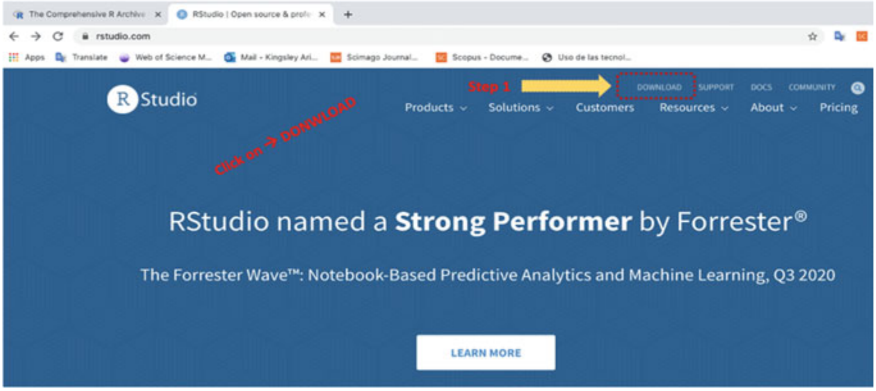


The installation was completed successfully.

The installation was successful.

The software was installed.

Fig. 1.6 (continued)



RStudio named a **Strong Performer** by Forrester®

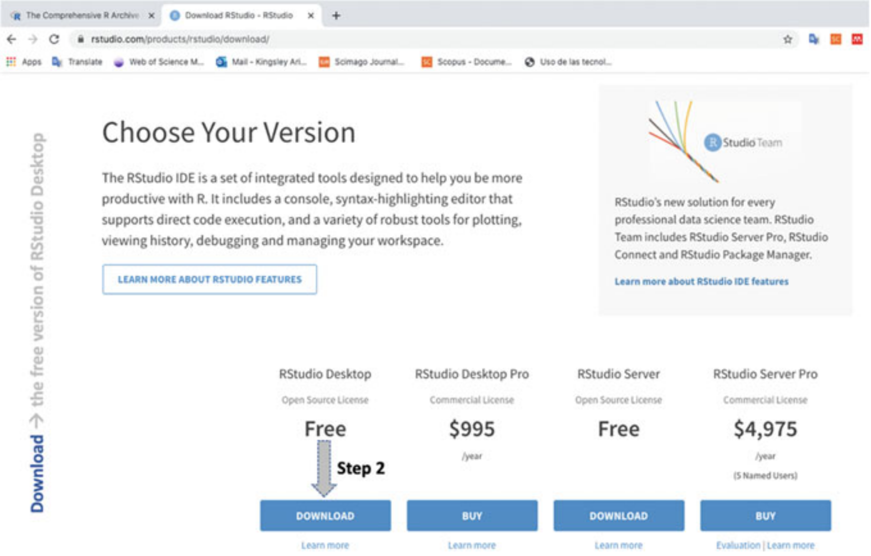
The Forrester Wave™: Notebook-Based Predictive Analytics and Machine Learning, Q3 2020

Fig. 1.7 Downloading RStudio software

When you have selected the free version of the software, then select the "**right version of the Installer for your Operating System (OS)"** as shown in Fig. 1.9.

For instance, as shown in Fig. 1.9, when the user clicks on the download link for the file "**RStudio.dmg"** (which is for the MacOS latest version at the time of writing this book), the executable program **(Installer)** will be automatically downloaded on the computer system. Navigate to the location where the downloaded (Installer program) file is stored on your computer or local machine, and install it as every other application program, e.g., by double-clicking on the downloaded file.

When you double-click or run the file, you will get a pop-up window as shown in Fig. 1.10.



Download → the free version of RStudio Desktop

Choose Your Version

The RStudio IDE is a set of integrated tools designed to help you be more productive with R. It includes a console, syntax-highlighting editor that supports direct code execution, and a variety of robust tools for plotting, viewing history, debugging and managing your workspace.

LEARN MORE ABOUT RSTUDIO FEATURES

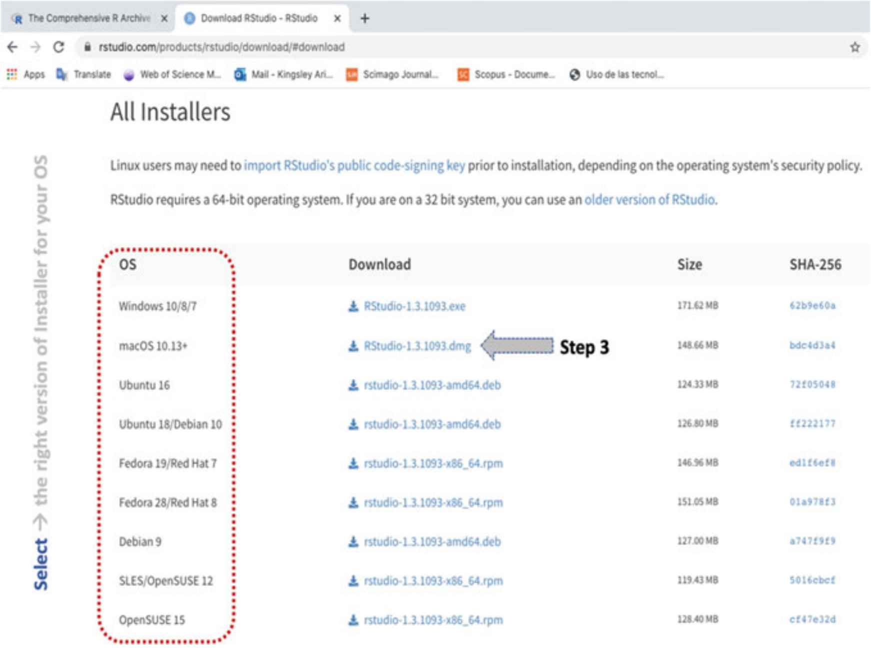
RStudio's new solution for every professional data science team. RStudio Team includes RStudio Server Pro, RStudio Connect and RStudio Package Manager.

Learn more about RStudio IDE features

RStudio Desktop	RStudio Desktop Pro	RStudio Server	RStudio Server Pro
Open Source License	Commercial License	Open Source License	Commercial License
Free	$995 /year	Free	$4,975 /year (5 Named Users)
DOWNLOAD	BUY	DOWNLOAD	BUY
Learn more	Learn more	Learn more	Evaluation | Learn more

Step 2

Fig. 1.8 Downloading free version of RStudio software



Select → the right version of Installer for your OS

All Installers

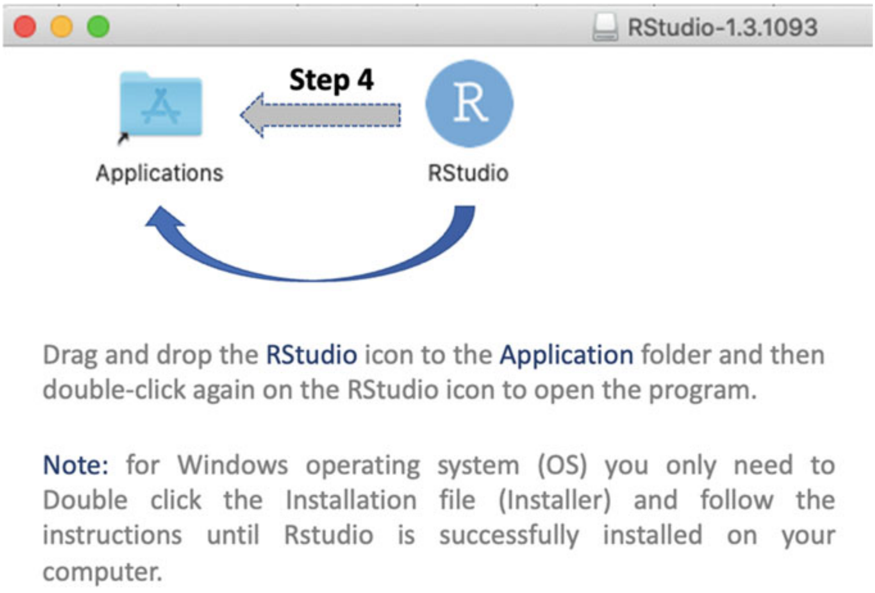
Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an older version of RStudio.

OS	Download	Size	SHA-256
Windows 10/8/7	RStudio-1.3.1093.exe	171.62 MB	62b9e60a
macOS 10.13+	RStudio-1.3.1093.dmg	148.66 MB	bdc4d3a4
Ubuntu 16	rstudio-1.3.1093-amd64.deb	124.33 MB	72f05048
Ubuntu 18/Debian 10	rstudio-1.3.1093-amd64.deb	126.80 MB	ff222177
Fedora 19/Red Hat 7	rstudio-1.3.1093-x86_64.rpm	146.96 MB	ed1f6ef8
Fedora 28/Red Hat 8	rstudio-1.3.1093-x86_64.rpm	151.05 MB	01a978f3
Debian 9	rstudio-1.3.1093-amd64.deb	127.00 MB	a747f9f9
SLES/OpenSUSE 12	rstudio-1.3.1093-x86_64.rpm	119.43 MB	5016cbcf
OpenSUSE 15	rstudio-1.3.1093-x86_64.rpm	128.40 MB	cf47e32d

Step 3

Fig. 1.9 Downloading the right version of RStudio for your operating system (OS)



Drag and drop the RStudio icon to the Application folder and then double-click again on the RStudio icon to open the program.

Note: for Windows operating system (OS) you only need to Double click the Installation file (Installer) and follow the instructions until Rstudio is successfully installed on your computer.

Fig. 1.10 Installing RStudio on your computer (e.g., for MacOS)

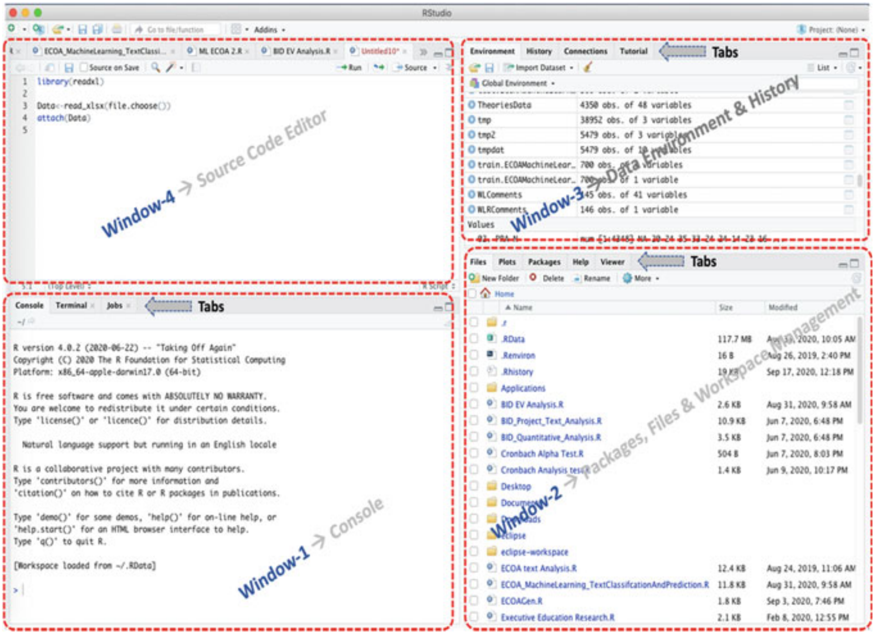
As illustrated in the figure (Fig. 1.10), same installation process applies to other types of OS (operating system) such as Windows if you are using the Windows OS. Double-click the Installation file to start up the executable file (Installer program). Then, click on **Continue** until you see the window that says you have **successfully installed the RStudio software**.

Once you have completed the installation process, **start the RStudio IDE** by either opening the application from the list of programs on your computer or clicking on the desktop shortcut icon. You will be presented with a Window as shown in Fig. 1.11.

Congratulations! You are now set to run and execute your first R project in RStudio. Welcome to using R programming for statistical data analysis in research covered as the main objective of this book.

The first time the users open RStudio, they will be presented with three windows by default, i.e., **Window-1**, **Window-2**, and **Window-3** (see Fig. 1.11). The fourth window (**Window-4**) is hidden by default and is only displayed when the user executes a program or run a command, but the users can also open it by selecting the **"File" drop-down menu**, then **New File**, and then **R Script** or simply by importing a dataset into the environment, which the authors will cover in detail in the next section (Sect. 1.4) and chapter (Chap. 2) of this book.

In Table 1.1, the authors outline the description and functions of the different tabs (component) of the R window or integrated development environment (IDE) (see Windows 1, 2, 3, and 4 in Fig. 1.11).



```
library(readxl)

Data<-read_xlsx(file.choose())
attach(Data)
```

```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

>
```

TheoriesData	4350 obs. of 48 variables
tmp	38952 obs. of 3 variables
tmp2	5479 obs. of 3 variables
tmpdat	5479 obs. of 10 variables
train.ECOAMachineLear…	700 obs. of 3 variables
train.ECOAMachineLear…	700 obs. of 1 variable
WLComments	145 obs. of 41 variables
WLRComments	146 obs. of 1 variable
Values	

Name	Size	Modified
..		
.RData	117.7 MB	Aug 31, 2020, 10:05 AM
.Renviron	16 B	Aug 26, 2019, 2:40 PM
.Rhistory	19 KB	Sep 17, 2020, 12:18 PM
Applications		
BID EV Analysis.R	2.6 KB	Aug 31, 2020, 9:58 AM
BID_Project_Text_Analysis.R	10.9 KB	Jun 7, 2020, 6:48 PM
BID_Quantitative_Analysis.R	3.5 KB	Jun 7, 2020, 6:48 PM
Cronbach Alpha Test.R	504 B	Jun 7, 2020, 8:03 PM
Cronbach Analysis test.R	1.4 KB	Jun 9, 2020, 10:17 PM
Desktop		
Documents		
Downloads		
Eclipse		
eclipse-workspace		
ECOA text Analysis.R	12.4 KB	Aug 24, 2019, 11:06 AM
ECOA_MachineLearning_TextClassifcationAndPrediction.R	11.8 KB	Aug 31, 2020, 9:58 AM
ECOAGen.R	1.8 KB	Sep 3, 2020, 7:46 PM
Executive Education Research.R	2.1 KB	Feb 8, 2020, 12:55 PM

Fig. 1.11 RStudio integrated development environment (IDE)

1.4 Running Your First R Project in R Using RStudio

In this section of the chapter, the authors introduces to the readers steps on how to create or start an R Project in RStudio. RStudio Projects make it easier and straightforward for the users to distribute their work into different categories or contexts, with each having their own working *directory* in the *workspace* including the history and source code documents. It is important to keep in mind that R projects are associated with a "working directory" where the users can save their new or running projects, and also retrieve existing projects.

Users can create an RStudio project in either a (i) **brand-new directory**, (ii) an **existing directory** where they already have R code and data, (iii) or by **cloning a version control** repository, e.g., from Git, GitHub or Subversion (see Fig. 1.13).

To create a **new R Project in RStudio**, start RStudio (see description in the previous section—Sect. 1.3). Once you are logged in and have the RStudio window open (see Fig. 1.11), click on the "**File**" menu at the top left corner of the RStudio window and select the "**New Project**" button as shown in Fig. 1.12. You will be presented with a pop-up window as shown in Fig. 1.13.

Select the "**New Directory**" option and fill in the pop-up with your chosen preferred **project_directory_name** by following the steps illustrated in Figs. 1.14 and 1.15.

Table 1.1 Description of function of the different tabs (component) of R window (IDE)

RStudio window	Component/ menu	Description/function
Window-1 (console)	Console tab	Results/output of the executed codes are displayed (printed) here. Also, further commands can be entered via the window
	Terminal tab	Command line system that allows the user to quickly control or access the operating system and make changes
	Jobs tab	Contains a list of open job(s) the system has currently running or pending in R
Window-2 (file explorer)	Files tab	File Explorer that allows the user to access the different files and folders stored on the hard drive (C: drive)
	Plots tab	Plots/graphs visualizations are displayed (outputted) at this location
	Packages tab	Contains a list of packages (libraries) that are installed in R on your system. Users can also install new packages or update the existing ones through the tab
	Help tab	Search window for help on different R topics, functions, or packages, and output location for the help commands
	Viewer tab	Advanced tab for local web content
Window-3 (file environment or history tab, e.g., objects, data)	Environment tab	Shows the list of interactive R objects that are loaded in the IDE
	History tab	Contains a list of key codes that are entered and executed unto the Console
	Connection tab	Tab through which the user can connect R to other external/ existing data sources or database
	Tutorial tab	Location where users can access different tutorials and learn about the several R functions, data types/variables, and commands
Window-4 (code editor)	Source/code tabs	A built-in text editor where the user enters the codes and commands, and can also find the shortcuts to run and save the commands/codes

When finished click the "**Create Project**" button and you're done! Congratulations once more! You have created your new project in R.

With the window and a console open, for instance, the authors named our project "**MyFirstR_Project**" in the directory as shown in Fig. 1.16, we are ready to run the R script, therein we can code and run the programs.

To create a "**new R script"** and run your new/written codes, select the "**File**" drop-down menu, then "**New File**", and then "**R Script**" as shown in Fig. 1.17.

You will be presented with a new working window or editor where you can start writing your code (see Fig. 1.18).

Now let's run some simple lines of code. As shown in Fig. 1.19 (see Steps 1 and 2), write the **example codes** from **Line 1** to **Line 4** in the Editor and execute the codes using the "**Run**" button (see Fig. 1.19). Example R code: Line 1: x <- 3 + 5

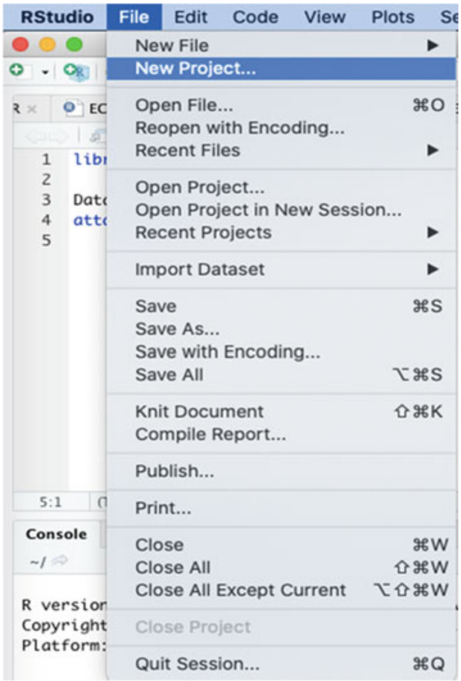


Fig. 1.12 Creating a new project in RStudio

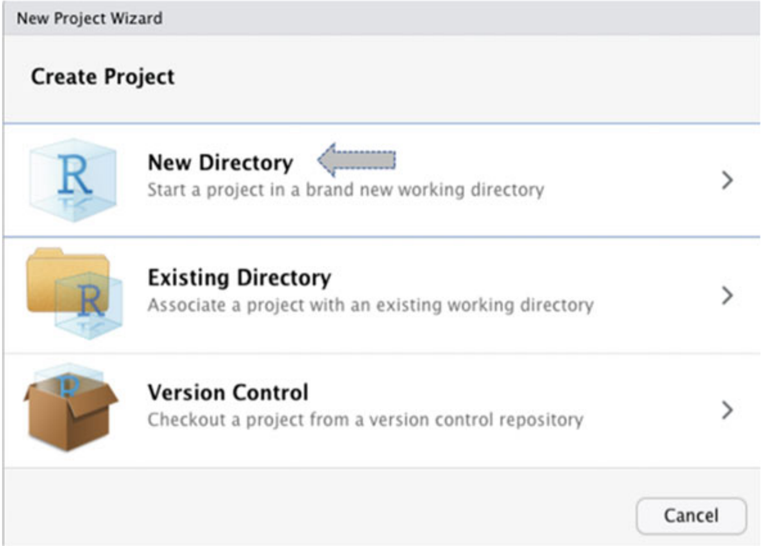

New Project Wizard
Create Project
New Directory
Start a project in a brand new working directory
Existing Directory
Associate a project with an existing working directory
Version Control
Checkout a project from a version control repository
Cancel

Fig. 1.13 New project wizard pop-up window

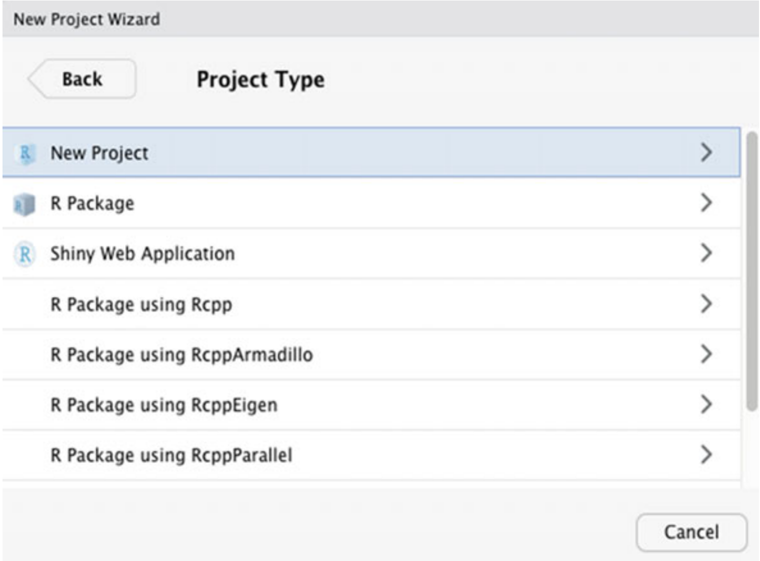
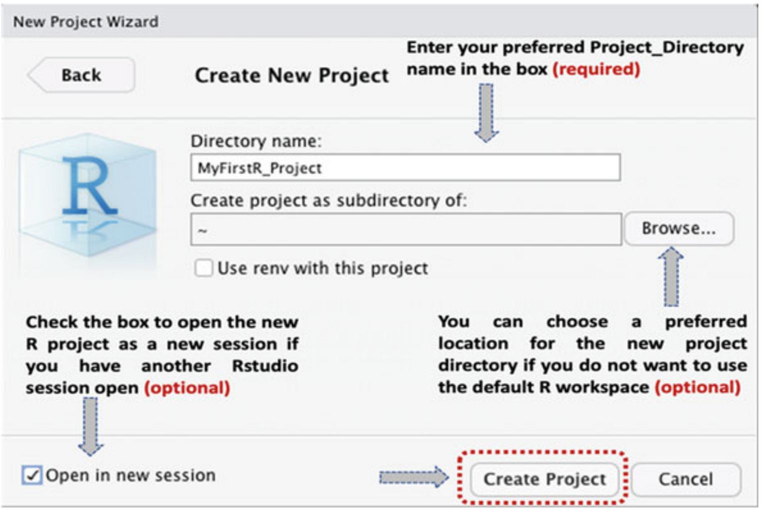


Fig. 1.14 Selecting the project type



Enter your preferred Project_Directory name in the box (required)

Directory name: MyFirstR_Project

Create project as subdirectory of: ~

□ Use renv with this project

Check the box to open the new R project as a new session if you have another Rstudio session open (optional)

You can choose a preferred location for the new project directory if you do not want to use the default R workspace (optional)

☑ Open in new session

Fig. 1.15 Creating a new R project and directory name

Line 2: x Line 3: print(x) Line 4: print("I am ready to work with data in R and start conducting the different statistical analysis for my research")

*Remember, start from Line 1 (e.g., by clicking anywhere in the line) before running the codes, or alternatively, follow the steps illustrated in Fig. 1.20 to run (execute) all the codes at once.