

## Procedures and Installation Guide for Open Source Statistical Software: A focus on R-Statistics and RStudio

*From the African Community of Practice on Management for  
Development Results at the African Capacity Building Foundation*



Guide

N°14

### SYNOPSIS

Understanding how to handle data is an important skill in today's world, especially as the cost of collecting data has fallen sharply. Yet data analysis could remain challenging because the cost of software licenses for data management is often prohibitive, which has led to widespread use of pirated software. To overcome this challenge, the Open Source Community has developed alternative tools. This guide provides a step-by-step procedure for setting up two of these tools: R-Statistics and RStudio.

An advantage of using the integrated environment of R-Statistics/RStudio is that the user can combine data analysis with report writing. RStudio makes it easy to use *rmarkdown* files and the *knitr* package to embed all analysis code and text and produce three report formats (MS Word, HTML, and PDF), aiding collaboration.

### Introduction

One of the greatest challenges confronting the African continent today is the weak capacity to collect, analyze, and use data for decision making. Despite global breakthroughs in data collection, analysis, and management, the story is different for Africa. The cost of procuring licenses for statistical software for data management is prohibitive for many of its organizations and individuals. For example, most social science analysis is done through packages such as Statistical Package for the Social Sciences (SPSS), Statistical Analysis Software (SAS), and Statistics and Data (Stata), but few institutions and individuals have official licenses, due to very high prices.

Their prohibitive costs have led to an emerging generation of pirated software that is often shared widely, especially in Africa.

To address this challenge, the Open Source Community has developed alternative tools that

can be used for data analysis and management at no additional licensing cost. All that is required is the ability to set it up and the skills to use it.

### Objective

The purpose of this guide is to provide a step-by-step approach to African Community of Practice (AfCoP) members and other potential users on how to transition to an Open Source platform for statistical analysis, called the R-Statistics. The document seeks to demonstrate how R-Statistics can help organizations and individuals gain control over data cleaning, analysis, and visualization with ease. It is intended to highlight some of the key packages of the software that can be leveraged by potential users for their data analysis. It is not intended to teach readers how to compute with R-Statistics.

Why do we need another guide on R-Statistics when there are several versions online? The answer is because with most Open Source

Software, the documentation tends to be scattered and it takes users a lot of time to figure out what tools users need to undertake their analysis. This guide contributes to the existing how-to guides on R-Statistics but uniquely puts together the key packages that users of proprietary software will need to transition to R-Statistics.

## Methodology

The drafting of this guide was informed by a desk assessment and a review of online literature. The guide delineates the key steps in installing and reading data files from R-Statistics. It makes full use of screen shots.

This guide is mostly suited for Windows users but comprehensive instructions are also available online for other platforms such as Linux and Macintosh Operating System (MAC OS).

## History of R-Statistics

The R language is a computer language developed by Ross Ihaka and Robert Clifford Gentleman at the University of Auckland, New Zealand, and is managed by the R Development Core Team (Baayen 2008). The R language comprises base and add-on packages.

The base part of R-Statistics can be referred to as the default standard for both graphical and statistical analysis. The add-on packages are user-developed extensions customized to accomplish various statistical and graphical tasks.

The use of R-Statistics has grown over the last decade and the most recent development is that Microsoft has embraced R-Statistics for its analytic solutions.<sup>1</sup> The positive trend in the adoption of R-Statistics among data scientists have made the software attractive, especially as it is distributed free of charge and allows users to

extend its capabilities through developing packages. In addition to its Open License, R-Statistics has excellent graphing and Geographic Information Systems mapping capabilities, as well as quantitative and qualitative data processing pipelines.

## Installing R-Statistics and RStudio

Installing R-Statistics is a two-step process that is better executed using a Graphical User Interface, especially for beginners. The philosophy of R, however, is to ensure that users have control over their analysis, including developing their own functions to solve users' unique needs.

**Step 1:** Download R-Statistics at the following link: <https://cran.r-project.org/bin/windows/base/R-3.2.3-win.exe> and then install it. This file is the engine that runs your analysis and is normally updated, and so it is good practice to always use the latest version of the software as subsequent releases often provide some fixes and enhancements to previous versions.

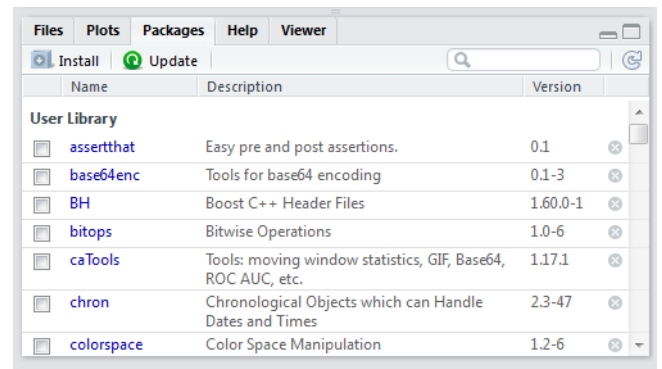
**Step 2:** After downloading the software, close the file and then install RStudio, an integrated development environment that makes it easy to manage files, scripts, and output files. The recent version of RStudio is located at <https://www.rstudio.com/products/rstudio/download/>. RStudio has four panes (figure 1).

The first pane shows the **Source** [1] and this is where you type your syntax, and save it, in case of further refinement or sharing with colleagues. R-Statistics can be used in an interactive mode, and the place to do so is in the **Console** [2] where you type commands and receive an answer. The user can first type arithmetic operations such as 1+1, 2\*2 and R-Statistics will give back the answers as shown in the **Console** (figure 1b). The bottom left pane shows the **Environment** where R-Statistics stores the objects. The **Files, Plots,**

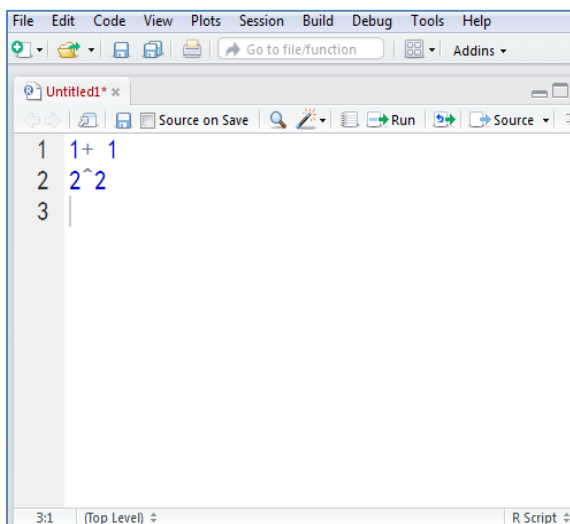
<sup>1</sup> [http://blogs.microsoft.com/blog/2015/01/23/microsoft-acquire-revolution-analytics-help-](http://blogs.microsoft.com/blog/2015/01/23/microsoft-acquire-revolution-analytics-help-customers-find-big-data-value-advanced-statistical-analysis/)

[customers-find-big data-value-advanced-statistical-analysis/](http://blogs.microsoft.com/blog/2015/01/23/microsoft-acquire-revolution-analytics-help-customers-find-big-data-value-advanced-statistical-analysis/).

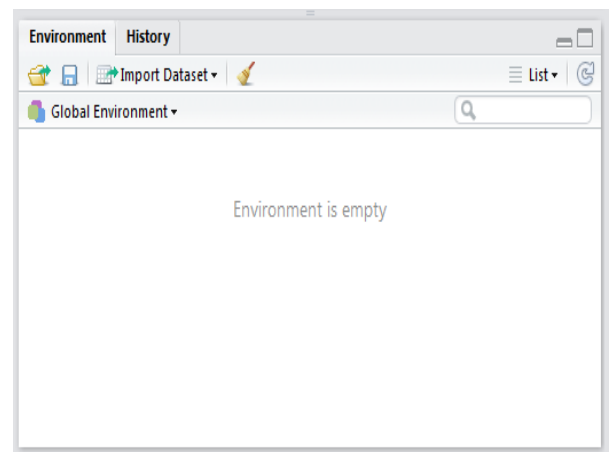
**Packages, Help, and Viewer** pane will show you the names of packages contained in R-Statistics (base packages and additional packages installed by the user, as well as others that can be written by the user—figure 1c). The **Plots** and **Help** window are displayed on the same pane. The fourth pane (figure 1d) will show objects stored in the **Environment** while the **History** keeps track of previous commands. We explain the concept of a working directory in detail as this often confuses non-programmers. Once users grasp where files are located, they will find it easy to read files into the **Environment**.



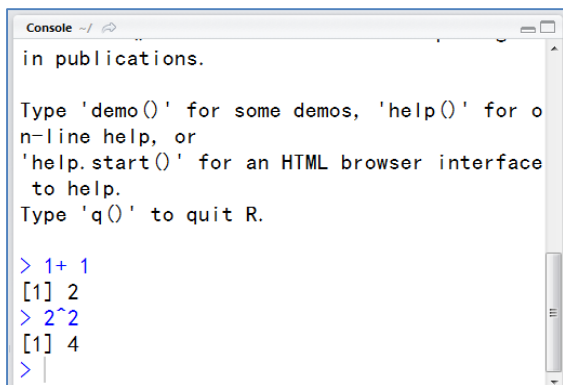
**Figure 1c: Files, Plots, Packages, Help, and Viewer pane**



**Figure 1a: Source window for typing and saving the analysis code**



**Figure 1d: Environment and History pane**



**Figure 1b: Console panel shows the output from the source window**

**A working directory**

First, create a folder on your desktop with the name 'RTraining'. The full path to your files will look like this: **C:\Users\SM\Desktop\RTraining**. In everyday interaction with our computers, we normally double click the mouse to navigate to the folders and then open a file that we need. However, in R-Statistics, we use code to do so, including specifying where we want to store our files. In the RStudio, "pane 4" shows that you are working in my folder named: **"C:/Users/SM/Box Sync/Rprogramming/paper"**. In short, "The

working directory points to a directory or folder on the computer where your data in R is stored.”<sup>2</sup>

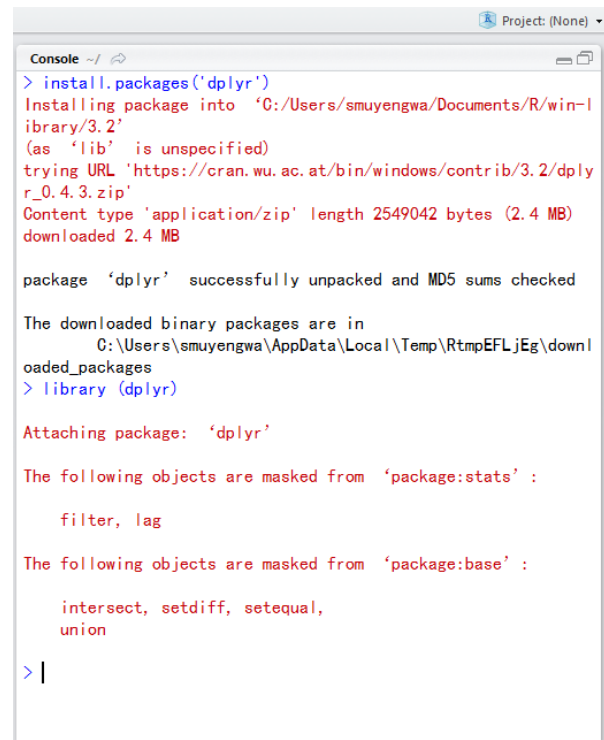
## Installing packages

As highlighted, R-Statistics comprises base and add-on packages. The base packages come with the default R-Statistics installation. The user has to install add-on packages depending on the functions of interest. Since R-Statistics is so specialized, there is at least one good package already developed to solve specific data analysis needs. Currently there are more than 6,760 R-Statistics packages. The next section demonstrates how to install some of the selected packages.

### R Commands

To install a package, you start by typing the following command `install.packages("packagename")` and then you hit the enter key in the Console window. Note that installing a package requires an active internet connection. Once a package has been installed, the user will be able to call it using the `library(packagename)`. It is advisable to use "quotation marks" when one is installing a package. These can be dropped when the user is loading a package into the workspace. Once a package is installed, the rest of the analysis does not require an active internet connection and you do not need to run the installation command each time you open R-Statistics/RStudio.

Figure 2 summarizes the process of installing packages in R-Statistics.



```

Project: (None)
Console ~/
> install.packages('dplyr')
Installing package into 'C:/Users/smuyengwa/Documents/R/win-library/3.2'
(as 'lib' is unspecified)
trying URL 'https://cran.wu.ac.at/bin/windows/contrib/3.2/dplyr_0.4.3.zip'
Content type 'application/zip' length 2549042 bytes (2.4 MB)
downloaded 2.4 MB

package 'dplyr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
  C:\Users\smuyengwa\AppData\Local\Temp\RtmpEFLjEg\download_packages
> 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

> |

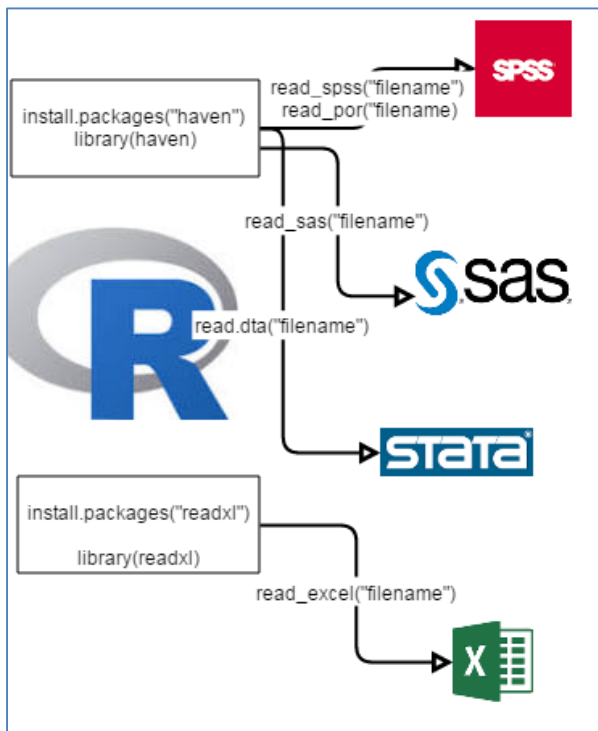
```

**Figure 2: Installing packages in R-Statistics/RStudio**

## Reading various data formats into R-Statistics

The purpose of this section is to demonstrate how R-Statistics can facilitate collaborations when working with people using different software for data analysis (often SPSS, SAS, and Stata). Figure 3 shows that one can use two packages to read files into the working environment to perform statistical data analysis.

<sup>2</sup> <http://neondatakills.org/R/Set-Working-Directory>.



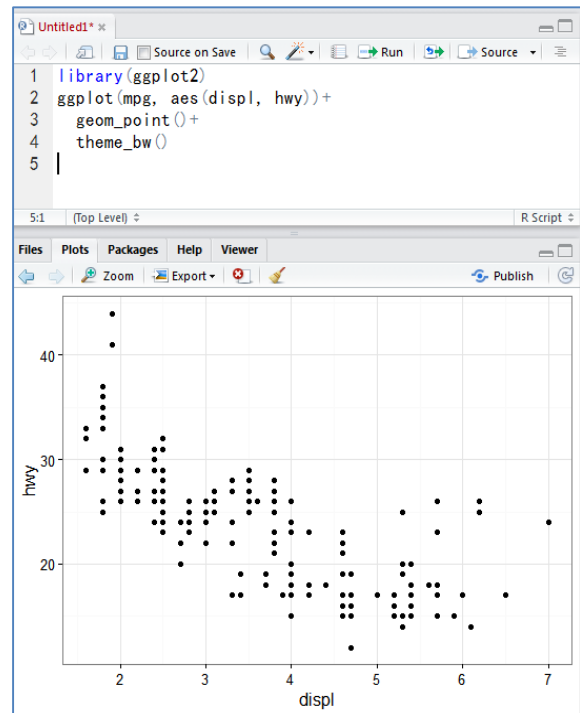
**Figure 3: Installing packages for reading SAS, SPSS, and STATA files.**

Using the “haven” package (Wickham and Miller 2015), we can read in the three major file formats. We can further use R-Statistics to interact with databases, and read Hierarchical Data and Geospatial Data. We can install an additional package called “readxl” (Wickham 2015a) to read in Excel files. Once the file is read into the Environment, the user will be able to undertake data janitorial work such as cleaning, analysis, modeling, and transformation. The haven package can write the files back into SPSS, SAS, or Stata. The user will be able to share cleaned files with colleagues in formats they are familiar with. The procedures for data cleaning and analysis are also package-specific within R-Statistics. Some of the key packages are:

- tidyr (Wickham 2016).
- dplyr performs complex summaries of grouped and non-grouped data (Wickham and Francois 2015).
- stringr effectively handles string data (Wickham 2015b).

### High-level plotting with R-Statistics

One of the strengths of R-Statistics is the capability to generate high-quality graphs. The software has several plotting devices, which include base plotting, lattice plotting, and ggplot2 (Wickham and Chang 2016). Each of the three packages has its own advantage. Figure 4 shows how graphing in R-Statistics looks. Several other packages can be installed for interactive plotting (*ggvis* (Chang and Wickham 2015), *googleVis* (Gesmann and Castillo 2015), *rbokeh* (Hafen and Continuum Analytics, Inc. 2016), and *ggmap* (Kahle and Wickham 2016)). With R-Statistics, it is possible to apply different graph themes as you would in other statistical packages.

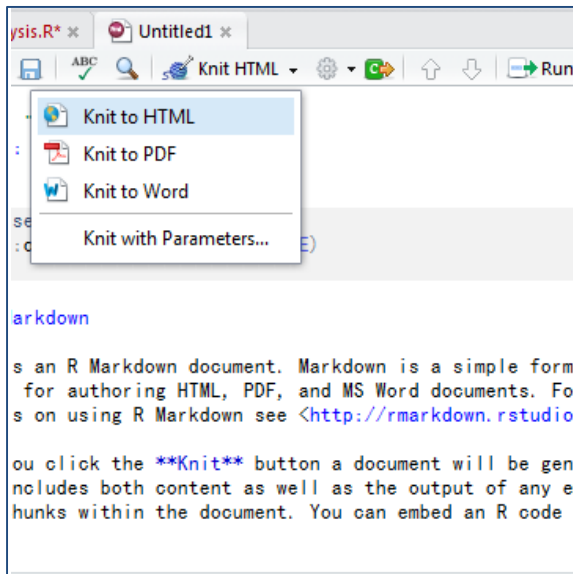


**Figure 4: Producing graphs in R-Statistics/RStudio**

### Outputs

An added advantage of using the R-Statistics/RStudio integrated environment is that the user can combine their data analysis with report writing. RStudio makes it easy to use *rmarkdown* (Allaire et al. 2016) files and the *knitr* (Xie 2016) package to embed all analysis code and text and produce three report formats as shown in figure 5 (MS Word, HTML, and PDF). This

framework is termed “literate programming” and saves time, especially when one needs to update reports daily or monthly. The *parameters* function allows users to subset their data by some variable and the analysis will automatically update and provide output relevant to the input parameters.



**Figure 5: Using rmarkdown to produce report in R**

## Conclusions and recommendations

The R-Statistics has immense capabilities that can improve the quality and user-friendliness of data analysis and reporting in Africa. Additional resources online through the Coursera,<sup>3</sup> DataCamp,<sup>4</sup> and EdX,<sup>5</sup> platforms provide free online courses on R programming—good for mastering R-Statistics and developing one’s own packages.

<sup>3</sup> [www.coursera.org](http://www.coursera.org).

<sup>4</sup> [www.datacamp.com](http://www.datacamp.com).

<sup>5</sup> [www.edx.org](http://www.edx.org)

## References

- Allaire, J. J., J. Cheng, Y. Xie, J. McPherson, W. Chang, J. Allen, and R. Hyndman. 2016. rmarkdown: Dynamic Documents for R. <https://CRAN.R-project.org/package=rmarkdown>.
- Baayen, R. H. 2008. Analyzing linguistic data: A practical introduction to statistics using R. Cambridge, UK: Cambridge University Press.
- Chang, W., and H. Wickham. 2015. ggvis: Interactive Grammar of Graphics. <https://CRAN.R-project.org/package=ggvis>.
- Gesmann, M., and D. de Castillo. 2015. googleVis: R Interface to Google Charts. <https://CRAN.R-project.org/package=googleVis>.
- Hafen, R., and Continuum Analytics, Inc. 2016. rbokeh: R Interface for Bokeh. <https://CRAN.R-project.org/package=rbokeh>.
- Kahle, D., and H. Wickham. 2016. ggmap: Spatial Visualization with ggplot2. <https://CRAN.R-project.org/package=ggmap>.
- Wickham, H. 2015a. readxl: Read Excel Files. <https://CRAN.R-project.org/package=readxl>.
- . 2015b. stringr: Simple, Consistent Wrappers for Common String Operations. <https://CRAN.R-project.org/package=stringr>.
- . 2016. tidyr: Easily Tidy Data with `spread()` and `gather()` Functions. <https://CRAN.R-project.org/package=tidyr>.
- Wickham, H., and E. Miller. 2015. haven: Import SPSS, Stata and SAS Files. <https://CRAN.R-project.org/package=haven>.
- Wickham, H., and R. Francois. 2015. dplyr: A Grammar of Data Manipulation. <https://CRAN.R-project.org/package=dplyr>.
- Wickham, H., and W. Chang. 2016. ggplot2: An Implementation of the Grammar of Graphics. <https://CRAN.R-project.org/package=ggplot2>.
- Xie, Y. 2016. knitr: A General-Purpose Package for Dynamic Report Generation in R. <https://CRAN.R-project.org/package=knitr>.



## Acknowledgments

This knowledge series intends to summarize good practices and key policy findings on managing for development results (MfDR). African Community of Practice (AfCoP) knowledge products are widely disseminated and are available on the website of the Africa for Results initiative, at: <http://afrik4r.org/en/ressources/>.

This AfCoP-MfDR knowledge product is a joint work by the African Capacity Building Foundation (ACBF) and the African Development Bank (AfDB). This is one of the knowledge products produced by ACBF under the leadership of its Executive Secretary, Professor Emmanuel Nnadozie.

The product was prepared by a team led by the ACBF's Knowledge and Learning Department (K&L), under the overall supervision of its Director, Dr. Thomas Munthali. Within the K&L Department, Ms. Aimtonga Makawia coordinated and managed production of the knowledge product while Dr. Barassou Diawara, Mr. Kwabena Boakye, Mr. Frejus Thoto and Ms. Anne François provided support with initial reviews of the manuscripts. Special thanks to colleagues from other departments of the Foundation who also supported and contributed to the production of this paper. ACBF is grateful to the African Development Bank which supported production of this MfDR case study under grant number 2100150023544.

The Foundation is also immensely grateful to Shylock Muyengwa, the main contributor, for sharing the research work contributing to the development of this publication. We also thank Professor G. Nhamo, Dr. Lyimo, and Dr. A. Kirenga whose insightful external reviews enriched this knowledge product. The Foundation also wishes to express its appreciation to AfCoP members, ACBF partner institutions, and all individuals who provided critical inputs to completing this product. The views and opinions expressed in this publication do not necessarily reflect the official position of ACBF, its Board of Governors, its Executive Board, or that of the AfDB management or board.



