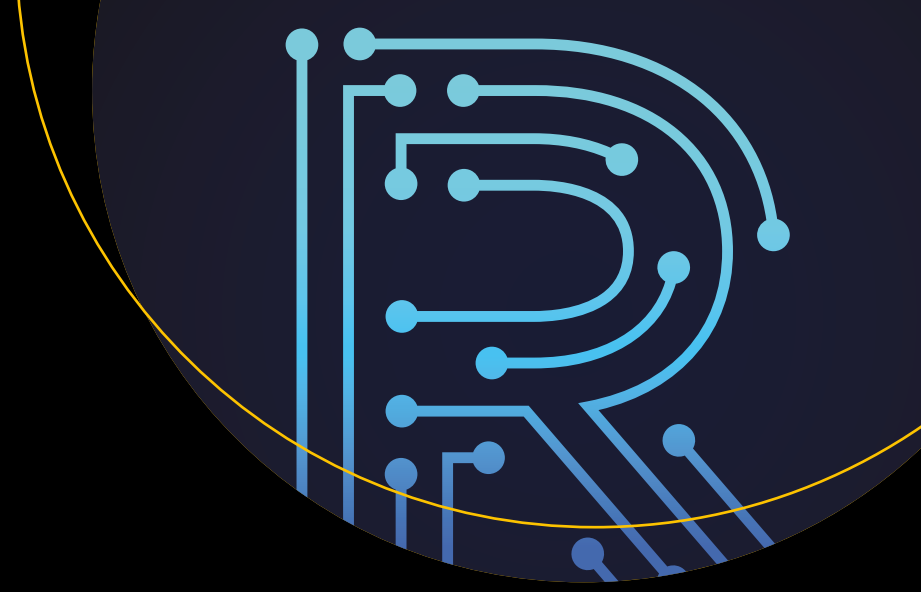

Beginning R 4

From Beginner to Pro

—

Matt Wiley
Joshua F. Wiley

Apress®

Beginning R 4

From Beginner to Pro

Matt Wiley
Joshua F. Wiley

Apress®

Beginning R 4: From Beginner to Pro

Matt Wiley
Victoria College,
Victoria, TX, USA

Joshua F. Wiley
Turner Institute for Brain and Mental Health,
Monash University, Melbourne, Australia

ISBN-13 (pbk): 978-1-4842-6052-4
ISBN-13 (electronic): 978-1-4842-6053-1
https://doi.org/10.1007/978-1-4842-6053-1

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Steve Anglin
Development Editor: Matthew Moodie
Coordinating Editor: Mark Powers

Cover designed by eStudioCalamar

Cover image by Shutterstock

Distributed to the book trade worldwide by Apress Media, LLC, 1 New York Plaza, New York, NY 10004, U.S.A. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/9781484260524. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

Table of Contents

About the Authors

Matt Wiley leads institutional effectiveness, research, and assessment at Victoria College, facilitating strategic and unit planning, data-informed decision making, and state/regional/federal accountability. As a tenured, associate professor of mathematics, he won awards in both mathematics education (California) and student engagement (Texas). Matt holds degrees in computer science, business, and pure mathematics from the University of California and Texas A&M systems.

Outside academia, he has co-authored three books about the popular R programming language and was managing partner of a statistical consultancy for almost a decade. His programming experience is with `R`, `SQL`, `C++`, `Ruby`, `Fortran`, and `JavaScript`.

A programmer, a published author, a mathematician, and a transformational leader, Matt has always melded his passion for writing with his joy of logical problem solving and data science. From the boardroom to the classroom, he enjoys finding dynamic ways to partner with interdisciplinary and diverse teams to make complex ideas and projects understandable and solvable. Matt enjoys being found online via Twitter (`@matt_math`) or `http://mattwiley.org/`.

Joshua F. Wiley is a lecturer in the Turner Institute for Brain and Mental Health and School of Psychological Sciences at Monash University. He earned his PhD from the University of California, Los Angeles, and completed his postdoctoral training in primary care and prevention. His research uses advanced quantitative methods to understand the dynamics between psychosocial factors, sleep, and other health behaviors in relation to psychological and physical health. He develops or co-develops a number of R packages including `varian`, a package to conduct Bayesian scale-location structural equation models; `MplusAutomation`, a popular package that links R to the commercial

Mplus software; `extraoperators` for faster logical operations; `multilevelTools` for diagnostics, effect sizes, and easy display of multilevel/mixed effects model results; and miscellaneous functions to explore data or speed up analysis in `JWileymisc`. Joshua enjoys being found online via Twitter (`@WileyResearch`) or http://joshuawiley.com/.

About the Technical Reviewer

Rachel Winkenwerder is an associate professor of mathematics at Victoria College and serves as assistant director of institutional effectiveness, research, and assessment. With extensive mathematical teaching experience in China and Texas for both secondary and tertiary education, Rachel understands the practical development of a contextualized curriculum. Her most recent work in higher education includes co-chairing her institution's curriculum and instructional council, reviewing regional accreditation narratives, and leading academic assessment. Her proven classroom track record teaching statistical methods courses combines with her current work in the R programming language and her earned degrees in computer science, mathematics, and education.

Acknowledgments

We would like to sincerely thank all our students through and across the years, who collectively taught us A Great Many Important Things.

Foreword

This book was, in part, written to be rigorously suitable for a sole mathematics course for undergraduates for what may be called "core." In that respect, it may be suited to fit into a statistical methods course (usually with either an elementary or introductory modifier at the front). Additionally, depending on how far through the last chapters one goes, it may also be suitable for an upper-division undergraduate course for the social sciences (e.g., psychology or sociology).

Beyond these objectives, however, this book is meant to empower learners to physically experience statistical thinking through the hands-on use of the R programming language. While theory will not be ignored, techniques and theory will be first motivated through models, visuals, and an intuitive approach. The true objective is to share a language for communicating complex numerical facts in understandable terms. This practical application – sometimes called empirical and quantitative skills – is designed to enable students to critically think and explore increasingly complex data sets, successfully describe and summarize large quantities of information, and accurately analyze, model, and communicate results to both technical and lay audiences.

To make this happen, this text has two distinct parts. Part 1 is designed to efficiently walk the reader step by step through installing and understanding the essential minimums of the R programming language's computer environment. We will do our best to avoid "techno speak," maintain an everyday language that jump-starts the reader as fast as possible through the initial stages, and move as smartly as possible to studying actual statistics. Part 2 is a methods approach to introductory statistics. Populations, samples, descriptive statistics, probability, distributions, correlation, regression, confidence intervals, hypothesis testing, and *an*alysis *o*f *va*riance (ANOVA) are all treated in turn. While not shying away from technical mathematical theory, all statistical ideas are first introduced conceptually. From visualizations to hands-on activities to help "get your hands dirty," the goal is to build a solid, real-world, contextual intuition that makes the theory more relatable. Your authors' goal is to help learners "live a statistical life" – and completely avoid writing a book used to pass one course and then quickly forgotten.

Lastly, while we have said this text is suitable for an undergraduate course, it will without a doubt increase a learner's ability to use a highly popular programming language and lay the foundation for using R for research, data science, machine learning, dynamic reporting, and bespoke visualizations. As such, it is also suitable for practiced data analysts looking to make the transition to R, for graduate students looking to both gain a powerful skill and refresh their knowledge of statistics, and for anyone who enjoys learning about data and statistics.

Thank you for spending your time and attention on our book. Please be sure to download source code and engage with this learning hands-on, and do not hesitate to reach out to us should anything not be working.

CHAPTER 1

Installing R

You are here to learn statistics, and the first step is *installing* some *program*. It can almost feel like a magician's trick. You need to know some applied statistics, and somehow you need to learn how to program instead. While this can look scary, never fear! This chapter will walk you through step by step what you need to do to have a pain-free installation. Regardless of whether you are learning on your personal computer (be that a PC or Mac) or on your company/institution's networked machine, this guide will let you know in everyday language what to do and yet stay complete enough to give to your IT department if needed (for corporate/institutional learners).

It is worth mentioning that by learning how to think statistically using the R [16] language, you not only learn on *free*, open source software, you also start *using* statistics from day 1 the way it is used in "real" life. In addition to statistical knowledge, you gain some quite useful applied skills.

This chapter helps you learn to be able to

- Download the latest version of R from CRAN [2] and install on either a Windows or Mac computer.
- Download and install RStudio on either a Windows or Mac computer.
- Understand the RStudio project environment.
- Evaluate if your installations and understanding work through applying some basic R code principles.

If you already know something about R and are eager to get into learning statistics, we start by listing the software required for this book to work in Table 1-1. On the other hand, if you wish to journey through the walk with us, please proceed through the chapter.

M. Wiley and J. F. Wiley, *Beginning R 4*, https://doi.org/10.1007/978-1-4842-6053-1_1

Table 1-1. *Beginning R Tech Stack*

Software	URL
R 4.0.2	https://cran.r-project.org/
RStudio 1.2.x	https://rstudio.com/products/rstudio/download/
Windows 10	www.microsoft.com/
MacOS	www.apple.com/macos

1.1 Your Tech Stack

Because the software in Table 1-1 tend to all require each other in a certain order and are technologies, the list of software required for a project to work/run is sometimes referred to as a *Tech Stack*. If you find yourself having trouble installing R, one of the first things you would want to share with anyone helping you is your Tech Stack. In particular, your computer's operating system will make a difference as may the version of R you are running. While this book is written for `R.Version()[["version.string"]]`, most likely future versions will not change so very much that this book will not work.

1.2 Updating Your Operating System

While it is not an absolute requirement, your computer's operating system (e.g., Windows or macOS) should be up to date. An up-to-date operating system helps for a few reasons. Firstly, we are going to walk you through installing the latest version of R and RStudio. Those are for sure tested on the current operating system versions. Secondly, most "tech" types tend to keep their software updated; thus, it helps to have a system similar to what they are currently familiar with should you run into trouble.

Windows

To ensure Windows 10 is up to date is not difficult at all. In the operating system search bar (a magnifying glass most often to the right of the Windows logo), type `check for updates` and select the option titled *Check for updates System settings* with your mouse cursor. This should open Windows Update, and there will be a box titled `Check for updates` to click. If successful, the *Last checked: Today* text will include the current time.

Before you move on to actually installing R, now is a good time to verify your computer is a 64-bit machine. It is possible, although not as likely these days, that you may have a 32-bit computer. In that same *Type here to search* Windows search box, go ahead and type About your PC. This should bring up an option titled *About your PC System settings*. Selecting that will open up a screen with some information about your system. In particular, under Device specifications, you're looking to see if your *System type* is either 64-bit or 32-bit. In either case, keep that in mind later on when you install R.

MacOS

Getting R running on *macOS* requires ensuring you have the latest version of macOS. This is important as the installation process differs across versions. At the time of writing, the latest version is *macOS Catalina*, with *macOS Big Sur* or later likely to be released by the time you are reading this. You can get help with upgrading from Apple: www.apple.com/au/macos/how-to-upgrade/.

1.3 Downloading and Installing R from CRAN

With your operating system up to date, it is time to download the latest version of R. As of this writing, it is R version 4.0.2 (2020-06-22) and can be found at https://cran.r-project.org/. Depending on your operating system, you will install a slightly different version of R. Thus, we ask you choose your own adventure.

Windows

To install R on Windows 10, use your favorite web browser (e.g., Chrome, Firefox, or Edge) to visit https://cran.r-project.org/. There should be a box on that web page titled *Download and Install R*, and you will want to select the *Download R for Windows* link https://cran.r-project.org/bin/windows/. From there, your goal is to install base R, so go ahead and select the *base* link https://cran.r-project.org/bin/windows/base/. The first link at the top should be *Download R 4.0.2 for Windows (XX megabytes, 32/64 bit)*.

It is both likely and possible that by the time you read this book, the version will be a larger number than 4.0.2. The megabytes may also likely be slightly different. Go ahead and download that latest version. Most likely, as long as the major version is still 4.x.x, everything in this book should stay fully or almost fully true.

After downloading, go ahead and navigate to your Downloads folder where the file `R-4.0.2-win.exe` is saved. Most often, your web browser will have a pop-up reminding you where the file is. Otherwise, you may press and hold the windows key on your keyboard followed by the E key which is sometimes called the *`Win + e`*. Not capital E, just the lowercase e key while still holding down the windows key button. This will open Windows Explorer, and most often there is a folder on the left titled *Downloads*.

Once you have found your `R-4.0.2-win.exe` file, you will want to double-click it to start the install process. For now, go ahead and accept all the default options, clicking Next and OK as needed.

MacOS

To get R running most flexibly on macOS, several additional tools are suggested.

Before you can install R, you need to install Xcode from App Store. After installation, **open it** to accept terms; otherwise, it may not work.

Once you have Xcode installed, you also need to install command-line tools. Open `Terminal` (if you cannot find it, try spotlight search) and type in `xcode-select --install` and then press *Enter* to run. If you run into any access issues, you may need to enable root. You can do that by typing in Terminal the following: `dsenableroot`, and pressing *Enter/Return* to enable root user.

Note At the terminal, if you are asked to enter a password, type the password you use to log in to your Mac and press Enter/Return. When typing your password in the terminal, no characters will appear, but it is still being entered.

Install `XQuartz/X11`. Visit `www.xquartz.org/`, download and run the file, and follow any on-screen instructions.

Go to `https://mac.r-project.org/tools/` and follow their instructions to get the mandatory tools and libraries. An example of the instructions includes, install `gfortran` by going to `https://github.com/fxcoudert/gfortran-for-macOS/releases` and download gfortran 8.2. Download and install following any instructions.

Although not needed for R itself, for many add-on packages that extend R's functionality, you will want some additional tools.

Install homebrew for macOS at https://brew.sh/ and follow the "Install Homebrew" steps. If you run into any access issues, you may need to enable root. You can do that by typing in Terminal the following: dsenableroot, and pressing *Enter/Return* to enable root user.

Note At the terminal, if you are asked to enter a password, type the password you use to log in to your Mac and press Enter/Return. When typing your password in the terminal, no characters will appear, but it is still being entered.

Install openssl which allows R to securely download files and packages from the Internet. Do this by opening the terminal (you can search for "terminal" or look in the launchpad) and type this code once the terminal opens and press *Enter*: `brew install openssl`

Install libgit2 which is needed for one of our graphing packages. Do this by opening the terminal (you can search for "terminal" or look in the launchpad) and type this code once the terminal opens and press *Enter*: `brew install libgit2`

Finally, download R by visiting https://cloud.r-project.org/ and clicking *Download R for (Mac) OS X*. Then download version 4.0.2. Once you have downloaded R, please be sure to install it to your apps.

1.4 Downloading and Installing RStudio

Congratulations! You have now installed R. However, there is still one more thing to install. While R by itself is powerful, it is not a convenient working environment. In programming, one often works in something called an IDE (*Integrated Development Environment*). In our case, we want to install RStudio Desktop, which is going to add a lot of easy-to-use visuals to help us "see" R better. RStudio will make it easier to focus on learning statistics.

Please visit https://rstudio.com/products/rstudio/download/ and select the *RStudio Desktop Free* option. As before, please catch up with us in the section that matches your operating system.

Windows

Having visited https://rstudio.com/products/rstudio/download/ and selected the *RStudio Desktop Free* option, you are now ready to install RStudio Desktop for Windows. As of this writing, the most up-to-date version is 1.3.1056.exe. Again as before, you will want to simply download the most recent version which should be step 2 on this website https://rstudio.com/products/rstudio/download/#download.

Again, simply click the download button, and note where your browser is saving the file. To install, double-click the textttRStudio-1.3.1056.exe file. Also, as with R itself, you will want to go with all the default options as they are selected, clicking *Next* and *OK* as needed.

Please note we are supposing you have a 64-bit operating system. If that is not the case, you will need one of the older copies of RStudio which may be found at this link: https://rstudio.com/products/rstudio/older-versions/.

MacOS

Having visited https://rstudio.com/products/rstudio/download/#download and selected the *RStudio Desktop Free* option, you are now ready to install RStudio Desktop for macOS. As of this writing, the most up-to-date version is 1.3.1056.dmg. Make sure R.app and RStudio.app are able to access disk resources needed.

Follow this guide www.r-bloggers.com/escaping-the-macos-10-14-mojave-filesystem-sandbox-with-r-rstudio to give them the necessary permissions.

1.5 Using RStudio

Now that you have R and RStudio installed, it is time to open and run RStudio for the first time. Go ahead and click the RStudio icon to open RStudio. What you should see is a small strip of icons along the top along with three large tiles or boxes that make up the bulk of the program screen. One large window pane on the left should be called *Console*, and inside that pane, there should be some text that reads *R version 4.0.2* (or whatever the version of R you just downloaded and installed is).

On the right side, there should be two smaller window panes. On top is the *Environment* which should be empty. On the lower-right side should be the *Files/Plots/Packages/Help/Viewer* pane. It should be showing you a file directory.

Your first chore (and you will only do this once) is to set some default options to make sure we share the same settings.

On the very top menu ribbon, find and click the *Tools* link, and then select *Global Options* from the drop-down menu. The Options menu should open, and you should be already in the *General* tab as in Figure 1-1. You will want to make sure the following options are **not** selected in that *General* tab ➤ *Basic*:

- **Restore most recently opened project at startup**: **Not checked**
- **Restore previously open source documents at startup**: **Not checked**
- **Restore .RData into workspace at startup**: **Not checked**
- **Save workspace to .RData on exit**: **Never**

Go ahead and click *Apply*.

Before you leave this Options menu, in order to have a little fun while you learn, please click *Appearance* as in Figure 1-2. Now, select the following options:

- **RStudio theme**: **Modern**.
- **Zoom**: **Set however you like**.
- **Editor font**: **Choose your favorite font**.
- **Editor font size**: **Big enough to read, small enough to optimize monitor space**.
- **Editor theme**: **Vibrant Ink** (we disagree on this – truly choose your favorite).

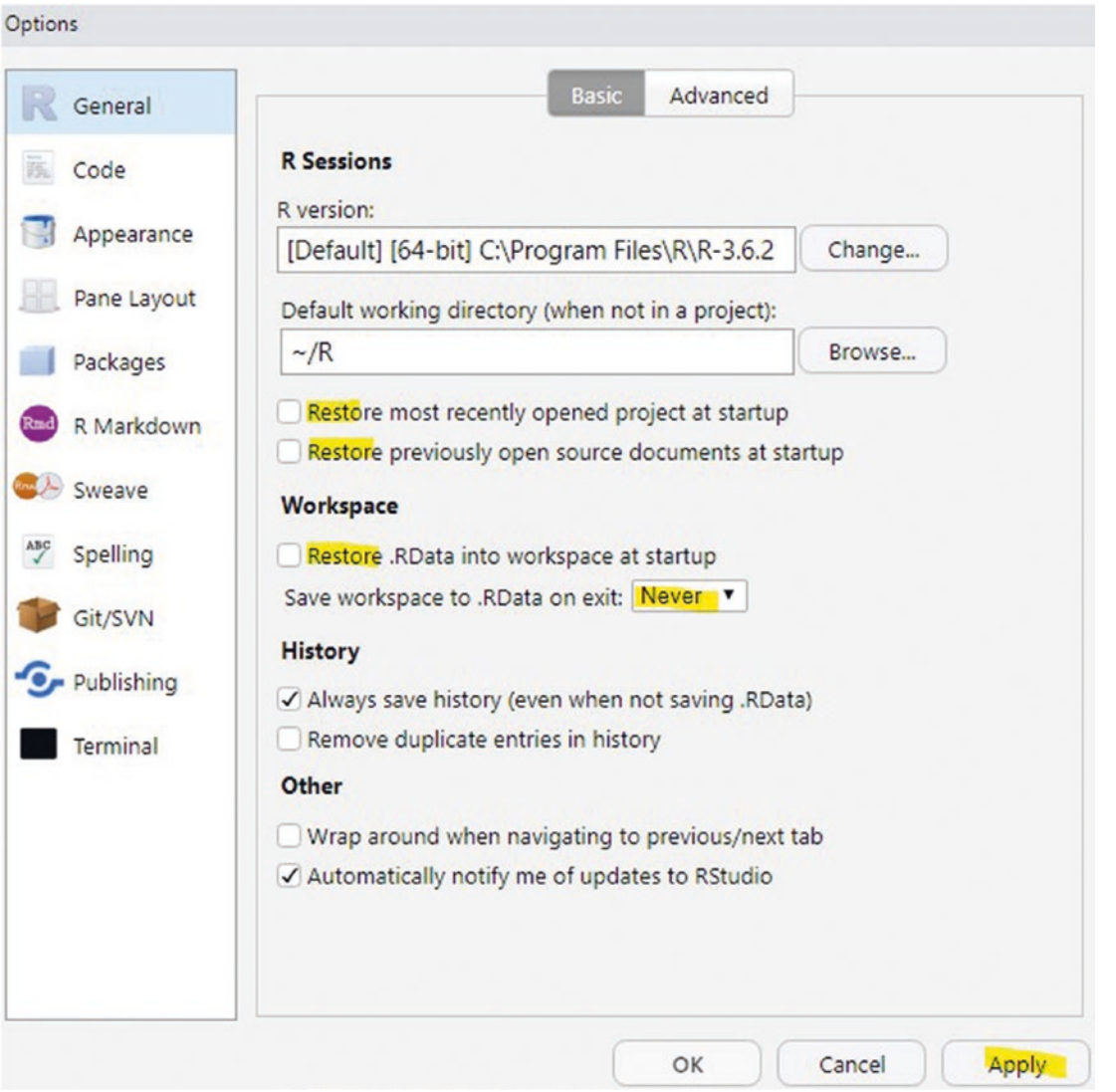

Figure 1-1. *RStudio general options*

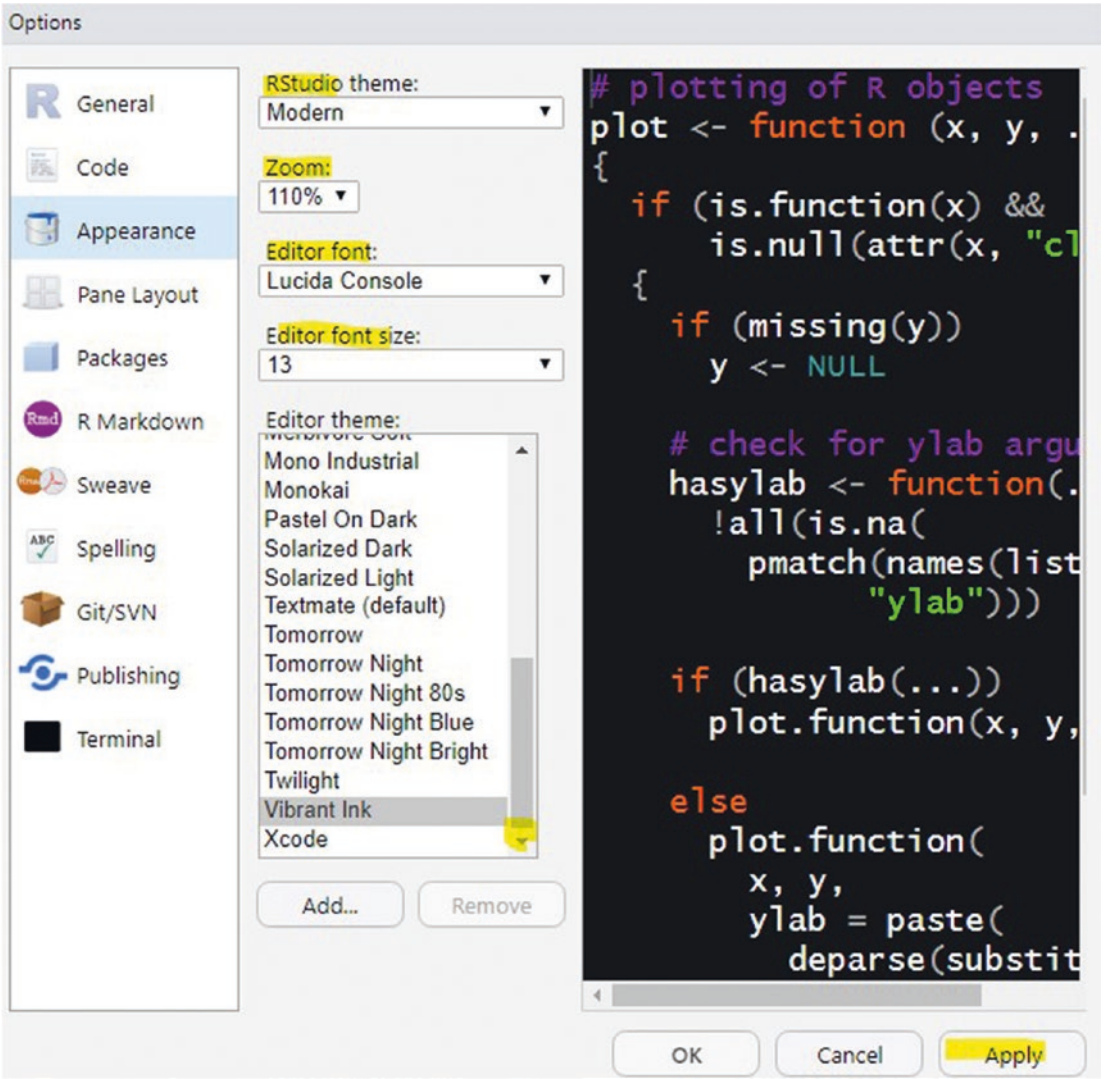

Figure 1-2. *RStudio Appearance*

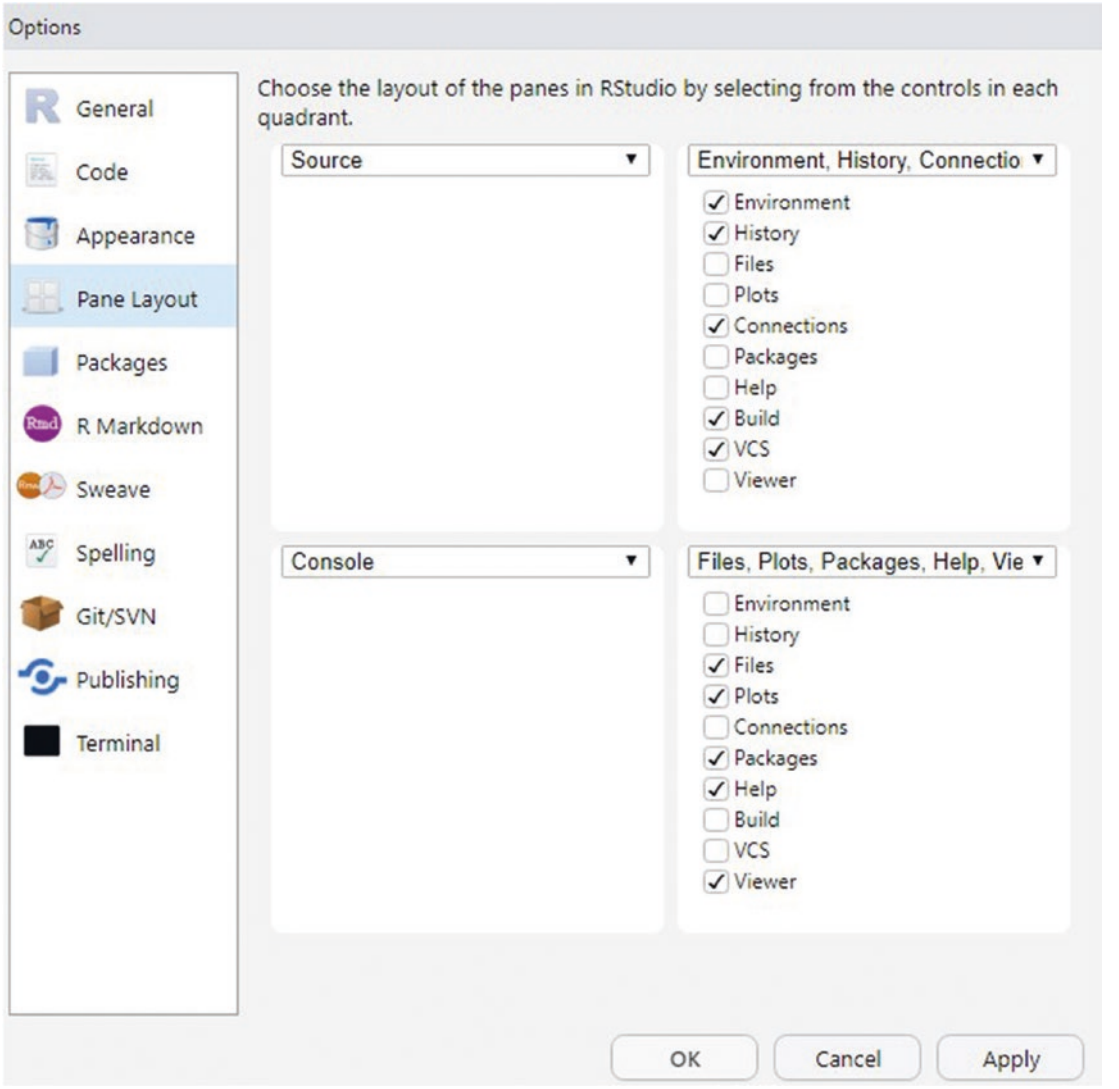

Figure 1-3. *RStudio Pane Layout*

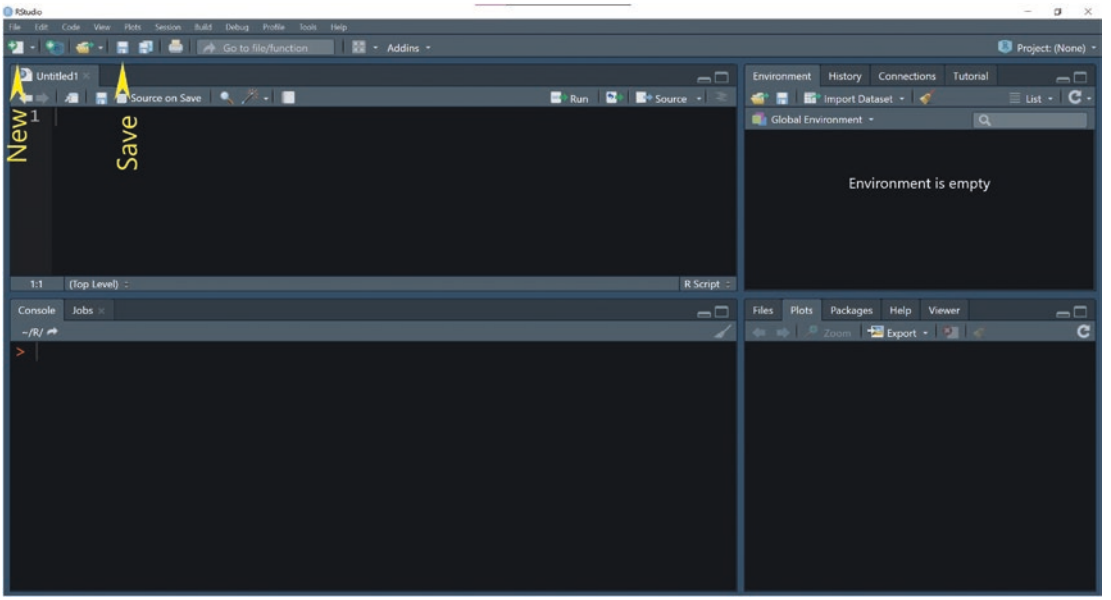

Figure 1-4. *RStudio New and Save example*

Lastly, please click *Pane Layout* and ensure your options match ours as in Figure 1-3.

Now, at last, you are done with setup. Please click *OK*, and welcome to a world where we spend a great deal of time.

Lastly, if you want to learn some more about using RStudio as an interface, the following two-page "cheat sheet" is a good place to go. It diagrams all the many parts of RStudio and how they can help you do your work: `https://github.com/rstudio/cheatsheets/raw/master/rstudio-ide.pdf`.

New Projects

Now that RStudio has some good defaults, it is time to start your first **project**. Creating a project builds a folder that will hold your work, and it has some files unique to RStudio which will allow you to easily work on a related set of ideas. There is no right or wrong answer for how many R files should live in a project. If you are taking this book as part of a class, it may make sense to build one project for each chapter. In that case, a good project name would be the title of the chapter. On the other hand, this entire book was written in a single project titled "BeginningR_2020."

For now, let us go ahead and build one project for each chapter. At the very least, you will get comfortable starting, closing, and opening projects.

To start a new project, on the upper-left menu ribbon, select *File* and click *New Project*. You will want to then use the New Project wizard to choose *New Directory* ➤ *New Project* and then pick a *Directory name*. For now, we recommend `01Installing`. Once that is typed into the *Directory name* field, select *Create Project*.

You are now in your first project!

Your first task is to create a new, blank `R` file. To do that, on the top ribbon, right under the word *File*, there is a small icon with a plus sign on top of a blank bit of paper. Click that icon, and then, from the menu of new files, pick the first one titled *R Script*. This blank file is now visible because you should have a tab in that top-left pane titled *Untitled1*. Go ahead and click the floppy disk–shaped *save* icon. Name this file `MyFirstRScript.R` and click *Save*. A screenshot showing what the new script and save look like is in Figure 1-4.

If all has gone well, you now have four panes as we described earlier in the setup. Your top-left pane is titled *MyFirstRScript.R* and is a script or code pane. This is the area where we will do most of our work. Moving clockwise, your top-right pane is the Environment pane; it is still empty. The lower-right pane is your *Files/Plots/Packages/Help/Viewer* pane. There should now be three files in your *project directory* which is in particular the *Files* tab. Those three files should be *.Rhistory*, *01Installing.Rproj*, and *MyFirstRScript.R*. Lastly, on the lower left is your *Console* pane, and right now it should still be telling you about your current `R` version. That is all about to change. Let us briefly discuss the purpose of each of these panes in turn.

The script pane, which currently holds your *MyFirstRScript.R*, is where you will type `R` code. Think of this just like a Word file or a PowerPoint slide. It will store and save any code you write there, and until you activate or *run* your code, nothing happens. If you like, this script area is your program. At present, your program is blank. We will change that very soon. You will write your first program in `R`, and become a programmer.

The *Environment* pane shows the data or *objects* or *variables* your program has stored in memory. Right now, there is nothing there. Most commonly in this book, you will find some data in that spot.

The *Files* tab shows your project's *working directory*. Any files you want to use for this project need to live in this directory. This is important! `R` needs to know where to look for any Excel files or data files you use to get information for statistical analysis. Those files all need to live in the directory folder. However, this pane does more than just show you files. Notice there are tabs for *Plots*, *Packages*, *Help*, and *Viewer*. Go ahead and click through each of those in turn. For now, *Plots* and *Files* and *Help* will be your most common tabs here. Go ahead and get back to *Files*.