# Overview

The online directory contains software to get you started with R, this document and a document R (programming language).pdf, downloaded from Wikipedia, which gives the history of this language. This document is divided into two parts: a technical part having to do with connecting to an online repository and a brief introduction to R and RStudio.

# Part 1: Atlassian, Bitbucket and SourceTree

The following steps describe how to download all online files associated with the book (including RJafroc) onto the user's computer, where they may be modified at will without disrupting the online files, and the user always has the option of reverting to the original files, i.e., discarding all local changes. Software (SourceTree) will be installed that serves as an interface to BitBucket (an Atlassian website that hosts such scientific projects). SourceTree alerts the user to any subsequent updates to the software and/or the physical book[[1]](#footnote-1).

Corrections to the book, as well as new features to the software and improvements to algorithms will be posted to the online repository. It is the author's hope that users will post questions and comments to this forum, and make suggestions for improvements. Any improvements resulting from user-input will be acknowledged.

This section is "geeky", but is necessary before one can get to the science. Persons already familiar with using online repositories like GitHub and Bitbucket probably do not need the following detailed directions. But for those who do, read on ...

The reader should have Internet access and administrative privileges and reasonable familiarity with working with software in one's preferred computing environment. While the screen shots and directions are for a MAC, the software runs on practically any platform.

## Create an Atlassian/Bitbucket account

To get to the following screen, type Create an Atlassian account into the Google search bar and hit Enter, Figure 1.

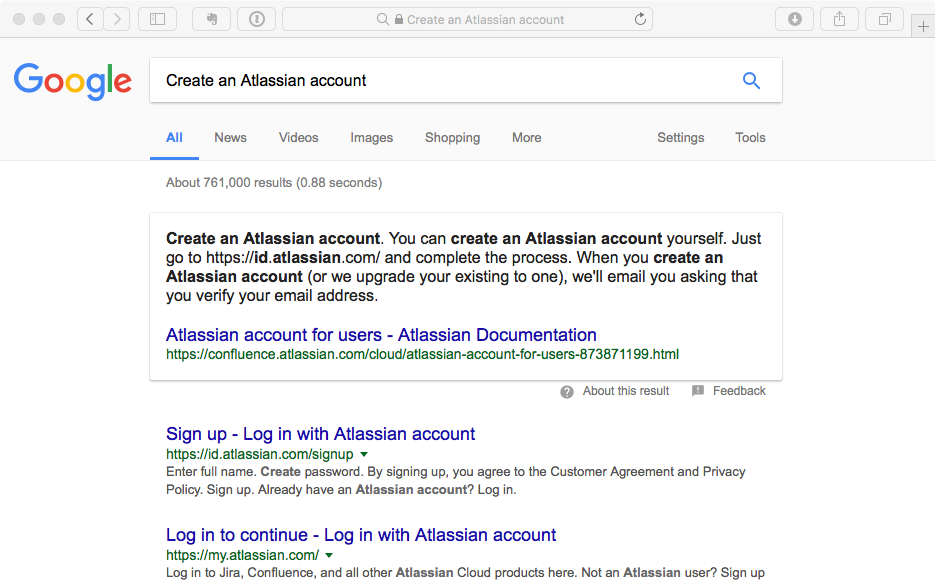


Figure 1

Follow the directions in the above screen-shot to create an Atlassian account. Record your username and password. To illustrate the steps, I created a user with email [beadevchakraborty@gmail.com](mailto:beadevchakraborty@gmail.com) and a password.

## Generate a password to allow communication between SourceTree and Bitbucket

SourceTree is an application (app) that facilitates working with Bitbucket. Type in the following link: <https://bitbucket.org/account/signin/> to sign in to your new Bitbucket account, Figure 2.

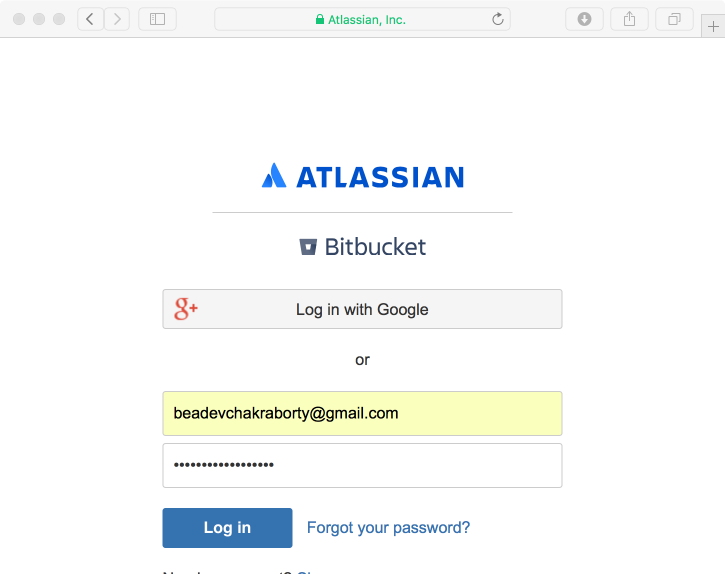


Figure 2

After clicking on Log in one sees Figure 3.

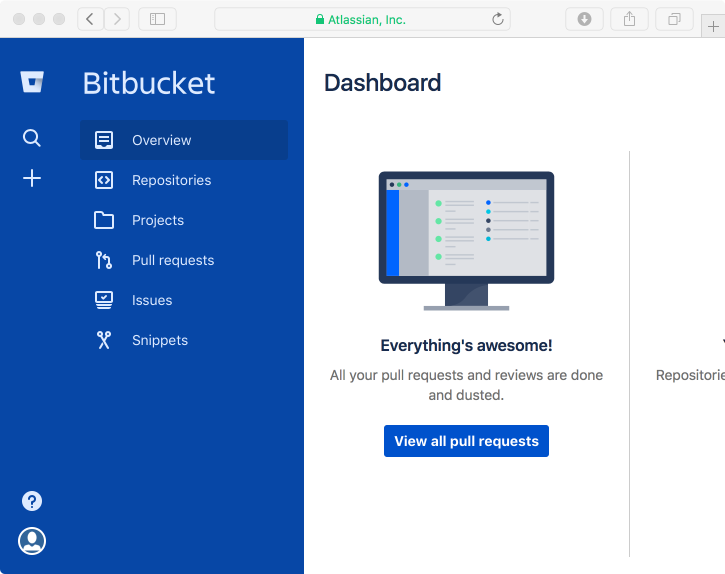


Figure 3

Click on the bottom-left icon (the face) and select Bitbucket settings, Figure 4.

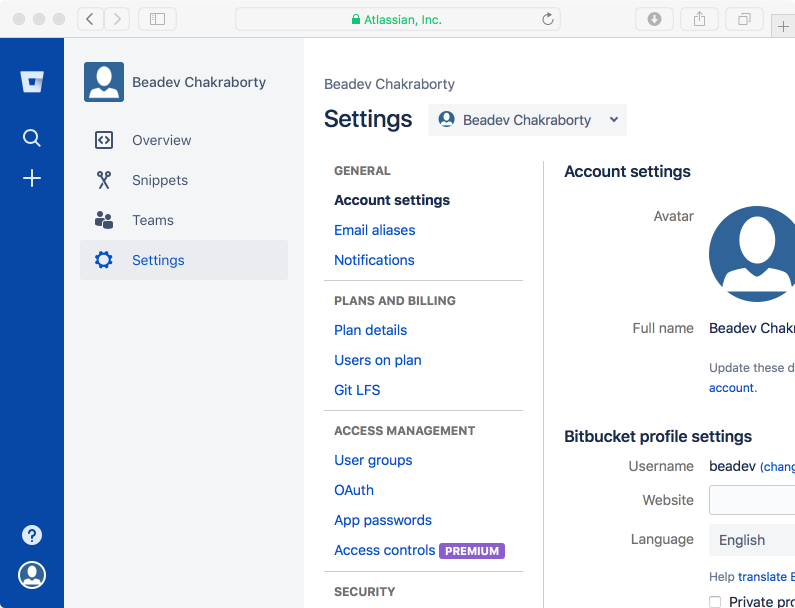


Figure 4

Click on App passwords. On the following screen click on Create app password, provide a label, which will be your username for SourceTree, and click Create, Figure 5. In the example I entered beadev as the username. This yields Figure 6, containing the automatically generated password.

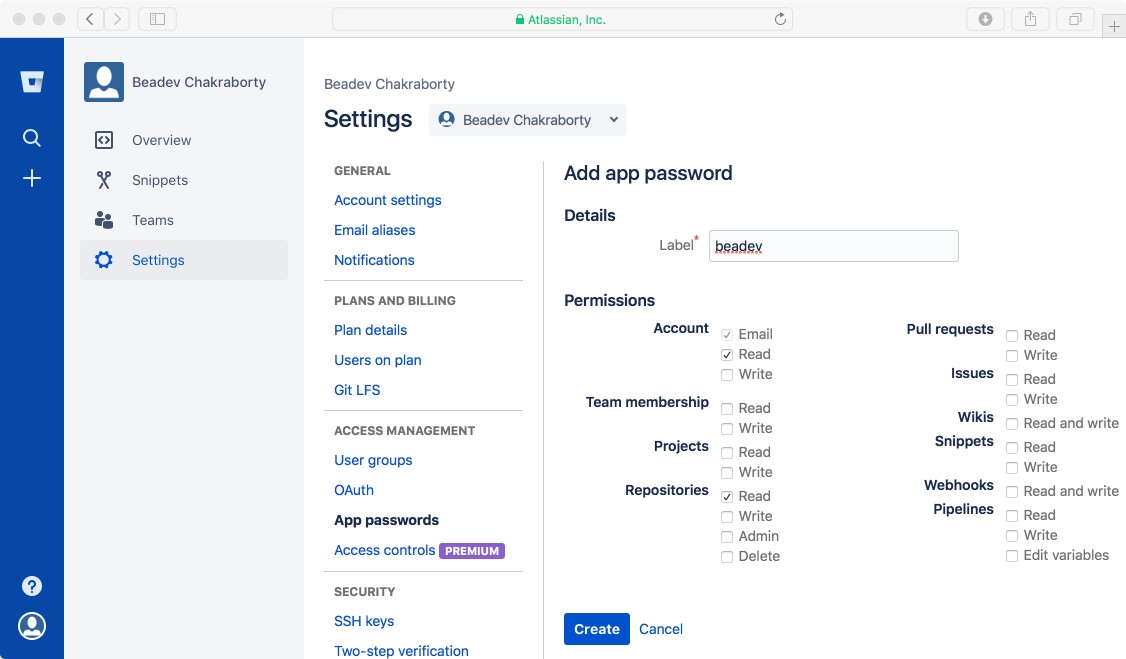


Figure 5

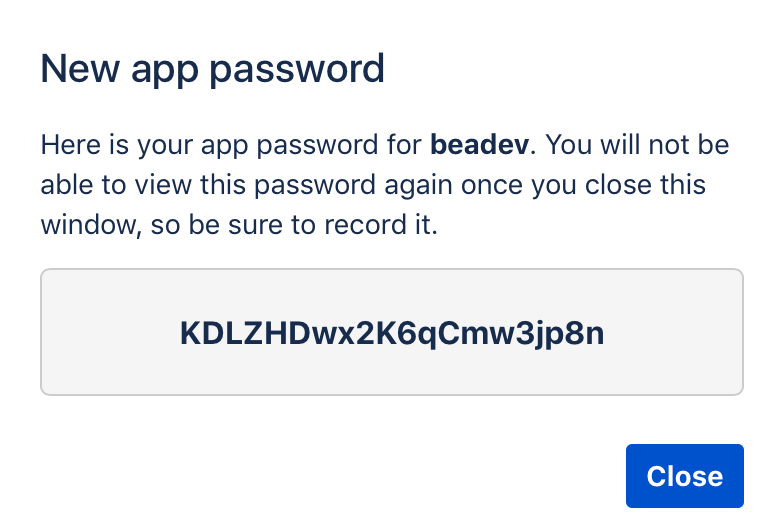


Figure 6

Copy the automatically generated password and save it. You will need it to allow SourceTree (the app) to communicate with Bitbucket.

## Download SourceTree

The next step is to download SourceTree, which is software that synchronizes the online book, and any subsequent changes made to it by me (Dr. Chakraborty), with the files stored on your computer.

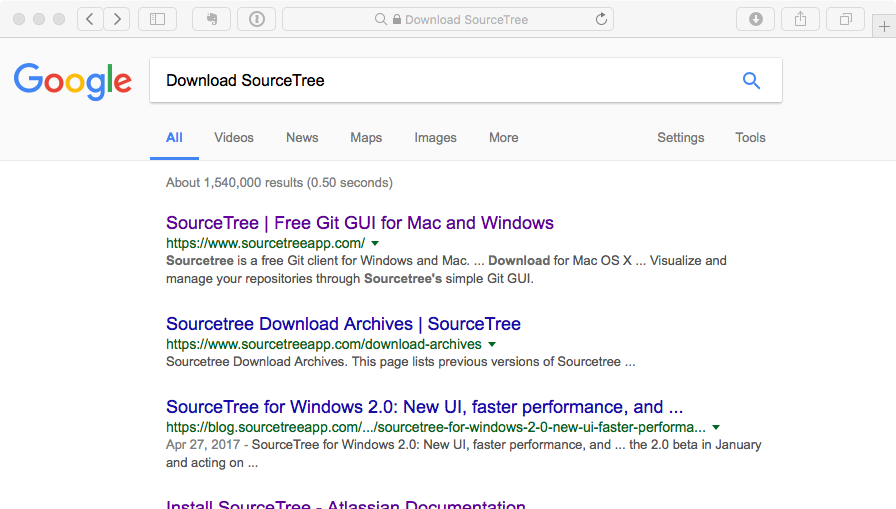


Figure 7

Click on the first link and install the software. Open SourceTree, Figure 8.

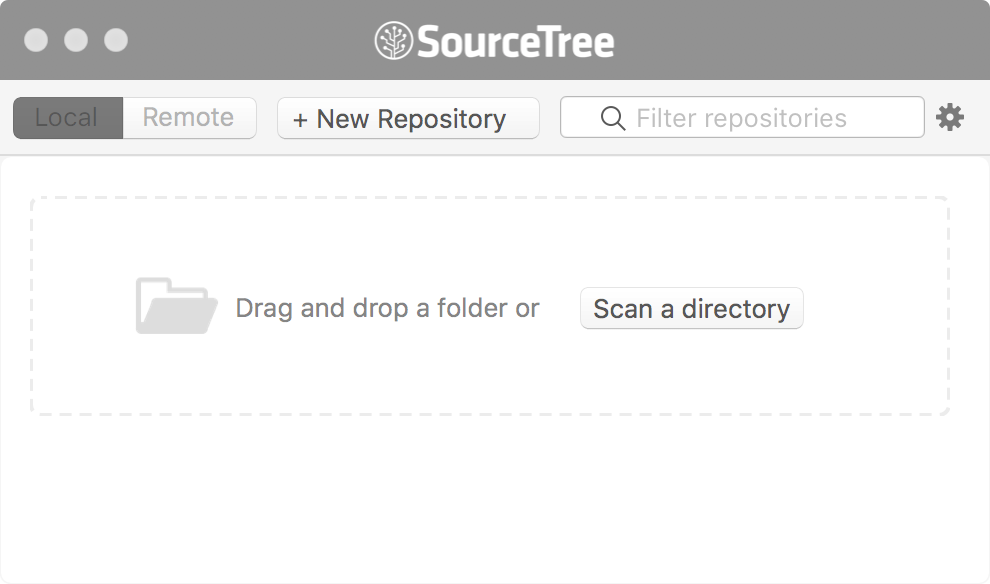


Figure 8

Click on the gear shaped icon at top-right, select Settings, Figure 9. [devtemp is coming from a new user, devtemp, that I created on my Mac to test the directions]

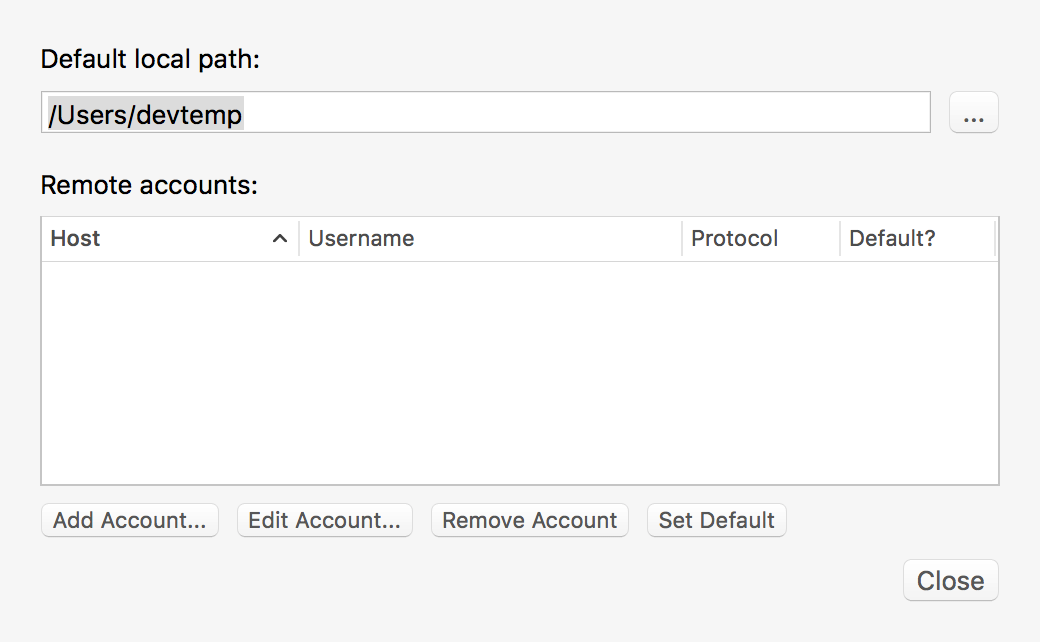


Figure 9

Click on Add Account, Figure 10.

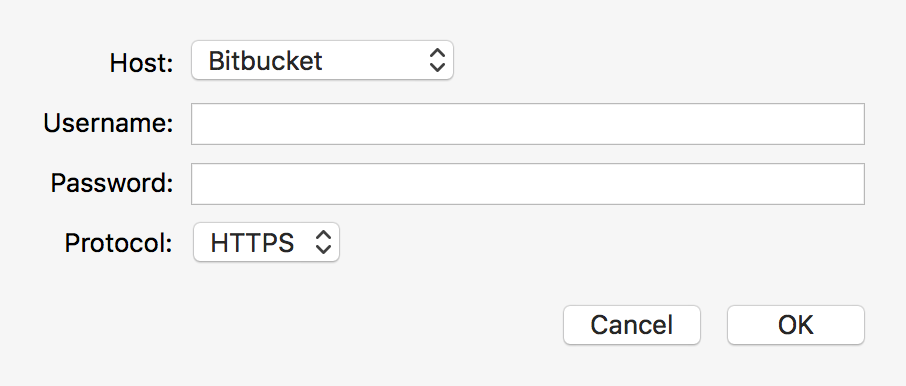


Figure 10

For Username enter the Atlassian/BitBucket user name, so in the current example it would be beadev, and for Password paste the value copied from the screen in Figure 6. Click OK followed by Close.

## Getting the book software

1. Type this link into your browser: <https://bitbucket.org/repo/all>?
2. Type OnlineBookK21778 under Search
3. Click the Search button, Figure 11.
4. [To download RJafroc[[2]](#footnote-2), in step 2 replace OnlineBookK21778 with RJafroc.]

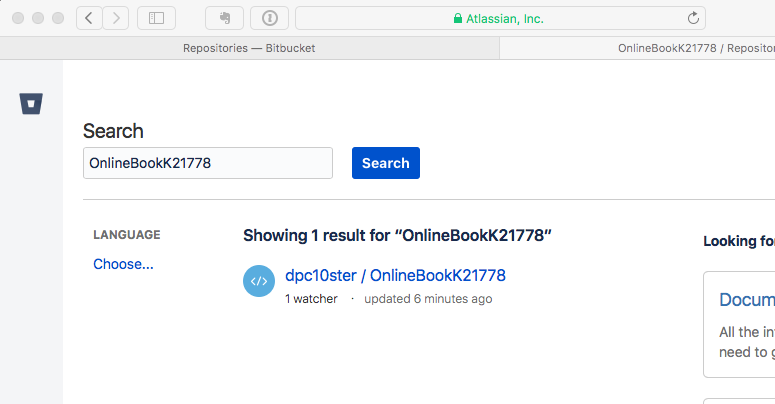


Figure 11

Click on the highlighted link: dpc10ster / OnlineBookK21778. You should see Figure 12.

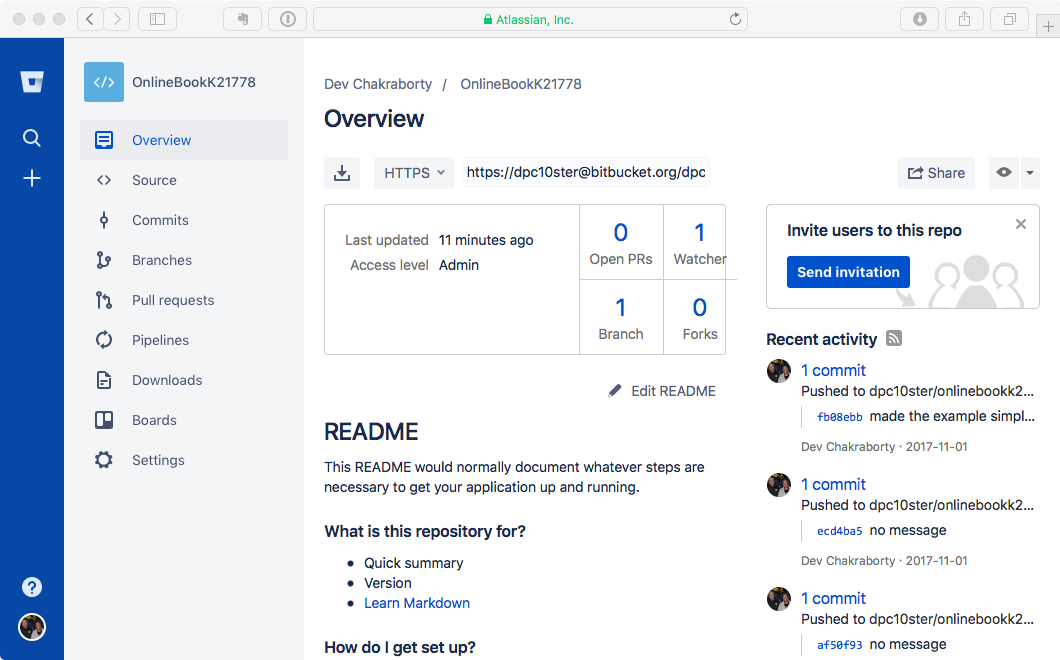


Figure 12

Under Overview there is a link perhaps described as an arrow pointed down into a shallow bucket. Hovering the cursor over it reveals the message: Clone in Sourcetree. Click on it and click Allow on the next screen, accept the default destination path and click on Clone, Figure 13.

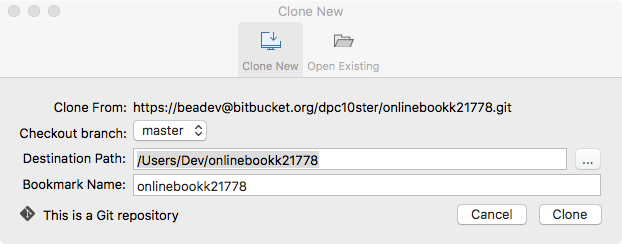


Figure 13

You should see Figure 14.

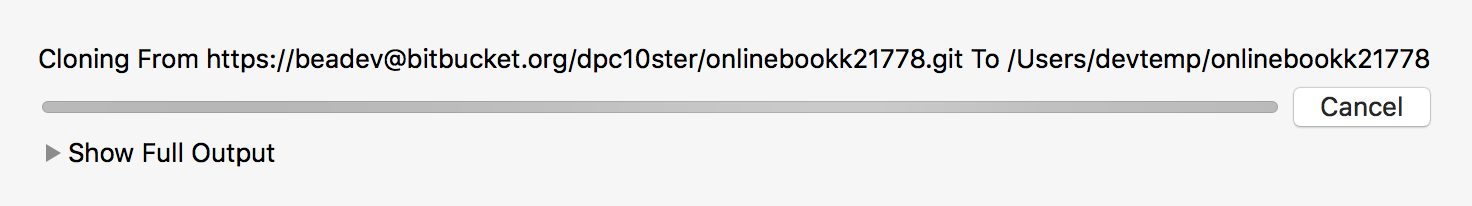


Figure 14

It may take a while, but finally you should see screens like Figure 15 and Figure 16 (in case you are wondering, "book" has not been misspelled. K21778 is the number assigned to this book by the publisher; also, you may see messages asking if confidential login information should be allowed access by the SourceTree software; you can safely allow it):

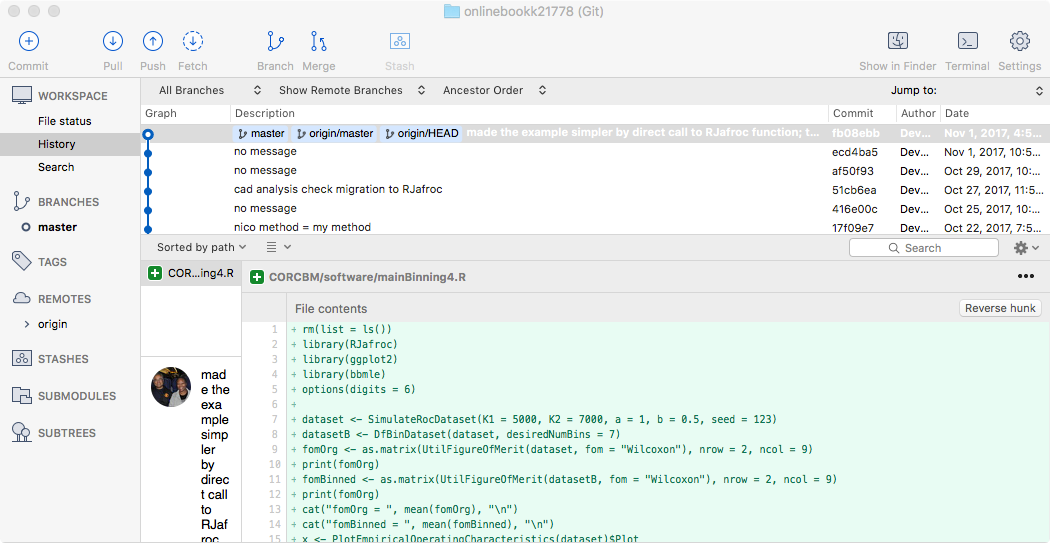


Figure 15

The various lines in the upper panel correspond to commits made by me during the writing of the book and the associated software. For example, the line which says nico method = my method, dated October 22, 2017, is a commit I made showing that a method of analyzing CAD data used by Prof. Nico Karssemeijer was equivalent to the method described in Chapter 22 of the book.

By simply double clicking on that line you can revert all your files to that date. To get back to the most current date, simply double click on the most recent commit.

Figure 16 is the SourceTree window showing that a repository onlinebookk21778 has been added.

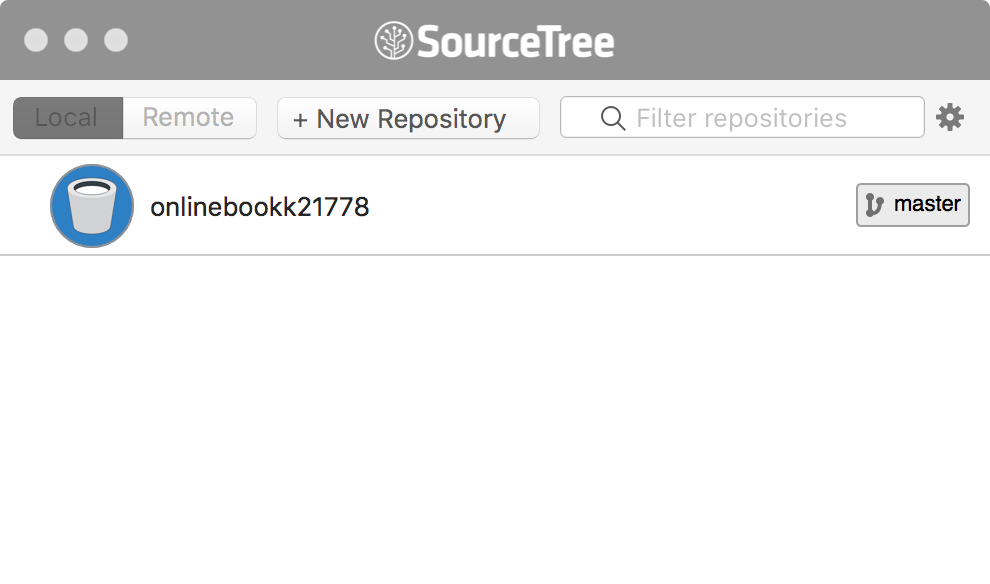


Figure 16

When one or more updates are available, and provided SourceTree is running on your machine, you will receive an alert and a number, 1 in the example below, Figure 17, tells you that an update has been posted to the repository and that you may need to synchronize your copy of the files with the most recent ones on the website (just click on master and click on pull on the next screen). There is much more but this should at least get you started.

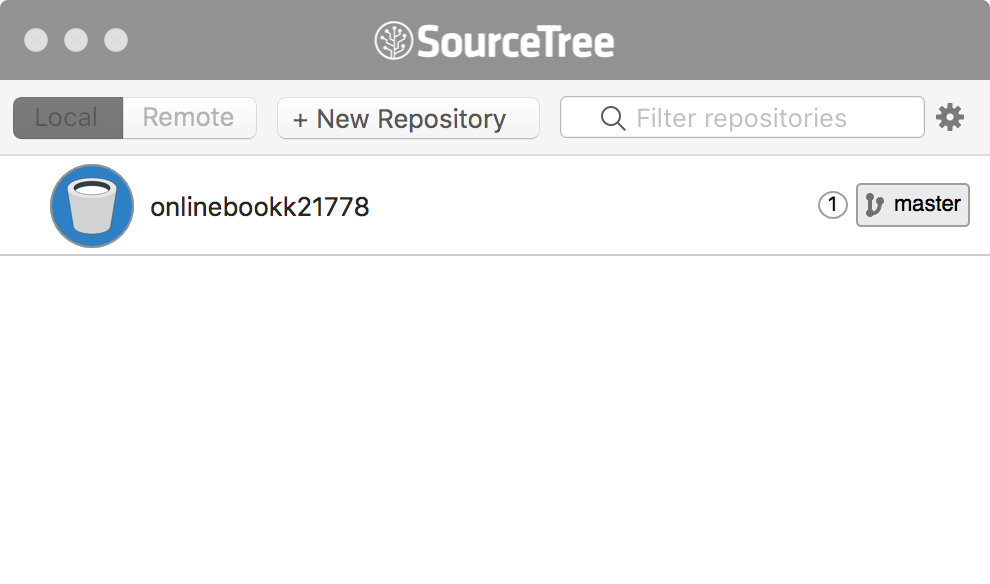


Figure 17

When the files have been synchronized, by clicking on master followed by pull, the "circled one" symbol will disappear.

## So where is the book software?

The files are located in the path named in Figure 9, which happens to be Users/devtemp/onlinebookk21778 in this example. You may wish to create an alias to this directory on your desktop. Figure 18.

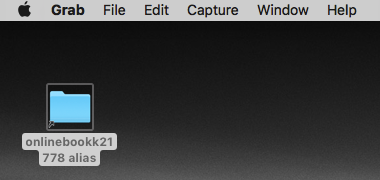


Figure 18

Click the desktop alias to reveal its contents, Figure 19.

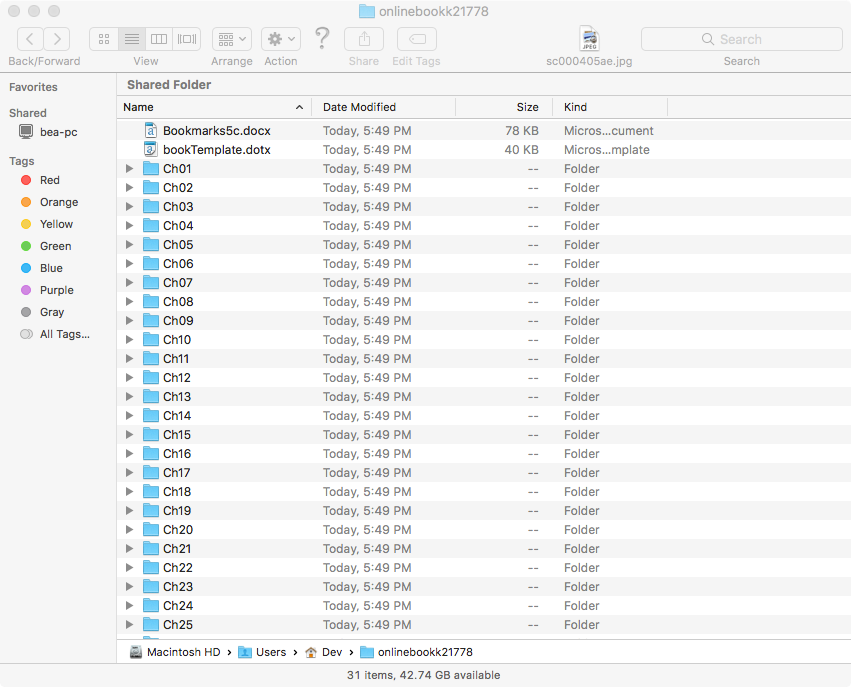


Figure 19

Click on Ch01, corresponding to Chapter 1, Figure 20.

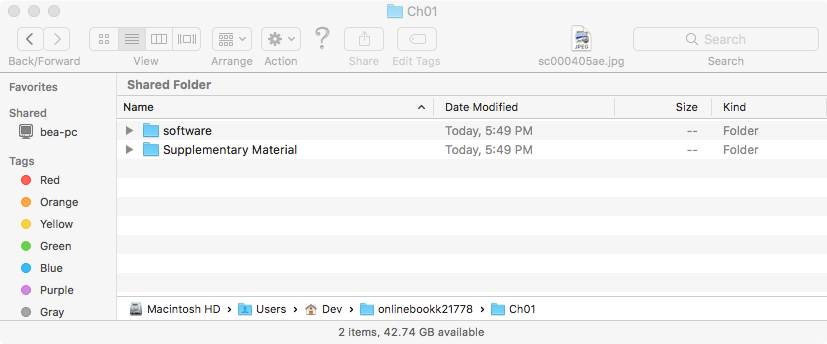


Figure 20

All chapters have a common structure: a folder containing software and a folder containing Supplementary Material (the folder may be empty in a few cases, e.g., Ch02) In this document we will work with Chapter 1 online material. But first we need to download two important R software related items. And this is where the geeky stuff ends and the science begins.

# Part 2: Online Appendix 1.A: Introduction to R/RStudio, Part I

Download R and the helper software RStudio. The code in this book should run on almost any computing environment: e.g., Windows, OSX, Linux, etc.). That is one reason for choosing it. The other reason is that R is open source platform, so one has access to the *entire* code; nothing is hidden from the user; with sufficient knowledge one can modify and improve the code; if so, the author would appreciate hearing from the user so that he may incorporate improvements in future versions of the book/code.

## Online Appendix 1.A.1: Windows computer

On a Windows PC do a Google search for "Install R Windows", and one should be able to find the appropriate link. Download the software and follow the installation instructions. One needs to also install RStudio after installing R. This too is free software; think of it as a user interface for R (R already has a basic user interface; RStudio adds many powerful features which make life a lot easier). Do a Google search for "Install RStudio Windows" and find the link. Download the software and follow the installation instructions.

## Online Appendix 1.A.2: OS X (MAC) machine

(This is the author's platform.) Search for "Install R MAC". Click on the downloaded software and follow the standard OS X installation procedure. After the R installation is complete install RStudio for the Macintosh operating system. Do a search for "Install RStudio Mac" and find the link. Download the software and follow the installation instructions.

## Online Appendix 1.A.3: Book software

Feel free to play with the code. Do not worry about "messing it up": if one does "mess up" one can always download the files again. *The author cannot emphasize strongly enough that the only way to really learn is by doing.*

## Online Appendix 1.A.4: A simple example using R/RStudio

This section is intended as a gentle introduction to the software. The screen-shots that follow are for the author's iMAC installation (the author's operating system was OS X Yosemite 10.10.5). If your screen shots look different these are due to differences in platforms (Windows, OSX, Linux, etc.) and version numbers. With a little faith and inquisitiveness one should be able to follow the brief tutorial below. *Follow these directions carefully:* find the folder corresponding to the chapter heading of this chapter and open it; in the author's computer it is /Users/devtemp/onlinebookk21778/Ch01. You should see a folder named software: open this folder, Figure 21. [devtemp is specific to this example; your value would be different]

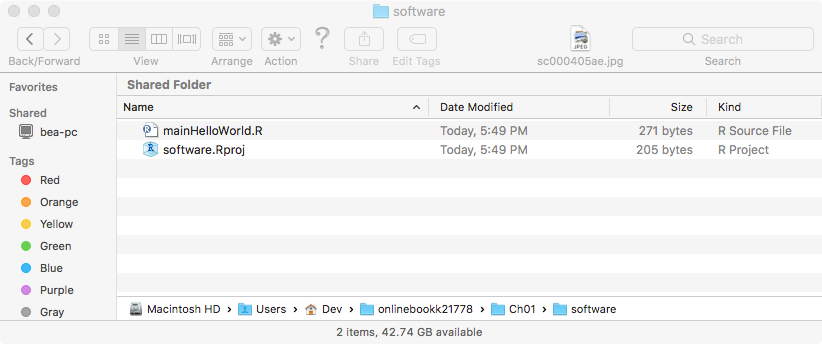


Figure 21

You should see a file named software.Rproj. The extension means that this is an R project file. A project file organizes all the files in your project for easy viewing, editing and compilation, but one does not need to know its contents. Open this file by clicking (or double clicking, depending on your computer settings) its icon. Your screen should resemble Figure 22.

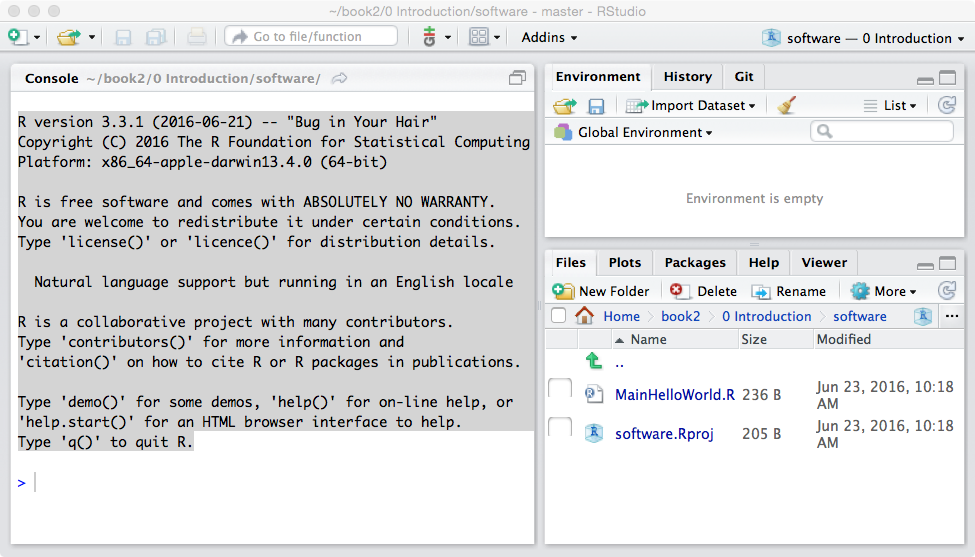


Figure 22

## Online Appendix 1.A.5 Introduction to the RStudio interface

Notice the organization of the display into 3 major panels (one may see 4 panels if any of the source code files - i.e., those ending in .R. - are open; simply close all such files to get a display similar, but perhaps not identical to Figure 23). The left panel is labeled Console and the right consists of two panels: the top-right panel consists of two sub-panels labeled Environment, History and Git, and the Environment sub-panel is in focus (notice the brighter background of the text Environment compared to History or Git. The bottom-right panel consists of five sub-panels labeled Files, Plots, Packages, Help and Viewer and the Files sub-panel is in focus (if it is not in focus, click on it to bring it into focus). The actual number of menu items may have changed by the time this book is published, because the developers of RStudio and R are hard at work improving the software: ignore any extra ones for now.

The left window, termed Console, ends with a ">" symbol. This is an invitation to the user to enter commands, so this symbol is the command prompt for R. Sometimes, when debugging a program, one might see this symbol preceded by a Browse[2], or Browse[3], as in , Browse[2]>. This indicates that the software is being run in debug mode. Debugging capabilities have improved dramatically in the author's four-year experience with the RStudio software.

The Console panel is currently simply showing the output screen one gets from starting R. Try it! Locate R using Spotlight (the magnifying glass symbol in the very top-right of the iMAC display) and open it (i.e., click on it). The R application is in the Applications folder. This is what I see, Figure 23. [On Windows go to Program and find the R application.]

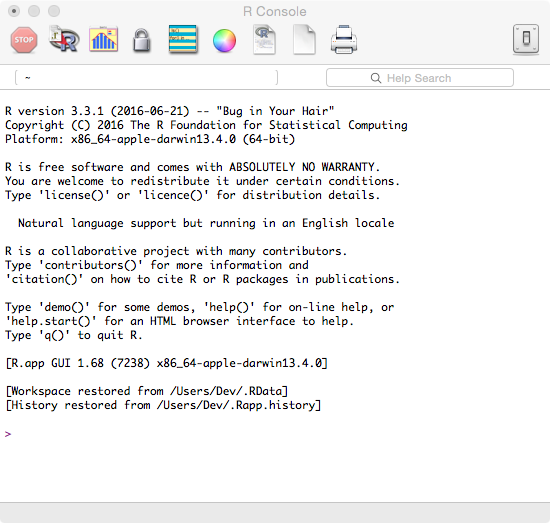


Figure 23

Screen-shot obtained by opening the R application (i.e., independent of RStudio). Note that it is practically identical to the shaded text in Figure 22. Think of RStudio as a helper application that makes it easier to work with R, analogous to SourceTree making it easier to work with Bitbucket. Excepting for the last few lines, the two screens (the left panel of Figure 22 and Figure 23) are identical.

Now quit R - in this book we will always use RStudio and never open R directly.

Returning to Figure 22, the Files sub-panel contains all the files in this project. Below the Files sub-panel one sees New Folder, Delete, Rename and More. The next row shows the complete directory path: Home/onlinebookk21778/software. The next row is labeled Name, Size and Modified, clicking on any of which sorts the files according to the chosen label. Click on the file named mainHelloWorld.R in this panel.

*In this book any file name that starts with main contains directly executable code; all the other files with the .R extension are functions (called subroutines in some programming languages) that may be called by directly executable code or by other functions.* This is what I see, Figure 24. Note the additional window showing the code file in addition to the Console window. Note the locations of the Run and Source buttons. The former button executes the line the cursor happens to be on, or a highlighted section of code. The latter executes the entire code in the file.

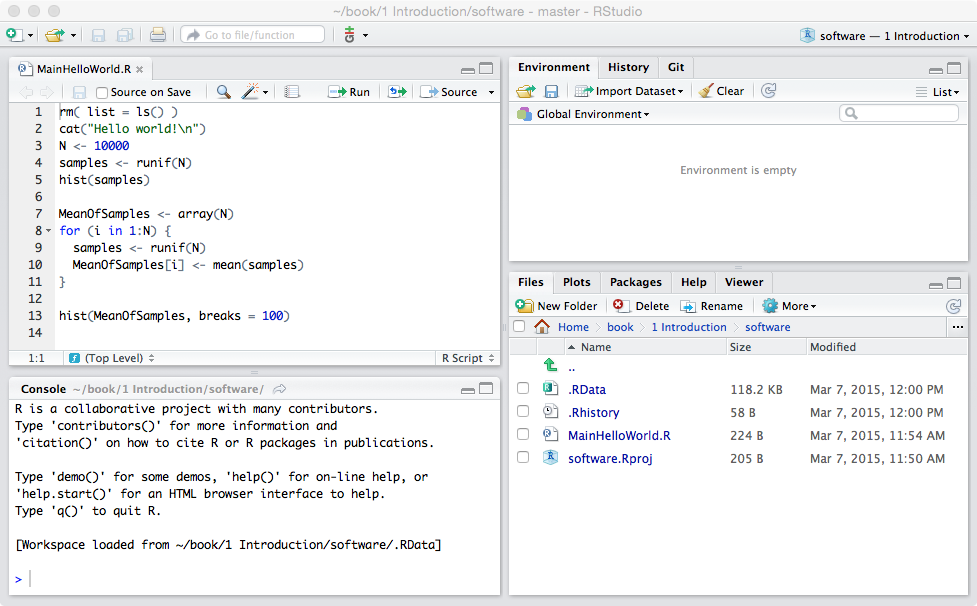


Figure 24

Notice the additional MainHelloWorld.R window occupying the top-left RStudio window, in addition to the Console window, which now occupies the bottom-left window. The additional window organizes your *source* code files (also called *script* files, these are all files with the .R extension) into one window for convenient viewing and editing; we will refer to it as the "*source*" panel. Whenever one creates new code for this project one should save it to the directory indicated on the bottom-right panel. To distinguish it easily from regular text, all code shown in this book will look like the one shown below (i.e., using the same font type and size, background, shading and borders). A listing of the file MainHelloWorld.R follows:

## Online Appendix 1.A.5.1 Code Listing

rm( list = ls() ) #MainHelloWorld.R

seed <- 1;set.seed(seed)

cat("Hello world!\n")

N <- 10000

samples <- runif(N)

hist(samples)

MeansOfSamples <- array(N)

for (i in 1:N) {

samples <- runif(N)

MeansOfSamples[i] <- mean(samples)

}

hist(MeansOfSamples, breaks = 100)

First, notice that the existence of helpful line numbers, generated by RStudio, not shown in the above code listing. The hash symbol (#) is used in R to insert comments: the compiler[[3]](#footnote-3) ignores anything appearing on a line after the hash symbol. This is a convenient way to insert explanations of what we are doing and/or why we are doing it this way. Ignoring the description that follows the hash symbol, the command rm(list = ls()) on line 1 says (in effect) that we are deleting ("removing") all existing variables so as to start with a "clean slate". It will be shown later how one uses the Help system to find out exactly what the code does. Blank lines of code, which are ignored by the compiler, are inserted for improved readability. Multiple commands, separated by semi-colons (;) can be put on one line to preserve "vertical real estate". R does not have a continuation symbol for long lines - we will see later how to split up long lines across several lines without using any special continuation symbol. Also, R commands are case sensitive: as one example, the command Rm(list = ls())will not work [Try it! Type Rm(list = ls())into the Console window (bottom-left panel in Figure 24) and press the return/enter key]:

### Online Appendix 1.A.5.1.1 Code Snippets

> Rm(list = ls())

Error: could not find function "Rm"}

Unlike some other programming languages (e.g., IDL and C), R uses the <- symbol to *assign the right hand side to the left hand side* (keyboard shortcut: use Alt– to enter this quickly in the Console window; the shortcut even inserts a space before the value that follows). Line 2 uses the set.seed() function to set the seed of the random number generator to unity, to force repeatable "random" sequences[[4]](#footnote-4). Line 3 uses the cat() function, which stands for *concatenate and print*. In the present example the cat() function simply prints the string variable "Hello world!". Line 4 initializes the variable N to 10,000. Line 5 obtains 10,000 samples from the standard random uniform distribution (i.e., uniformly distributed in the closed range 0 to 1). The function used is called runif(N) for "N random samples from the random uniform distribution". The 10,000 samples are assigned to the variable samples. Line 6 uses the hist() function to display a histogram of these samples. Line 8 allocates an array variable MeansOfSamples to hold 10,000 numbers. The for‑loop beginning on line 9 and ending on line 12 means, effectively, "repeatedly execute the statements in the enclosed curly brackets for all values of the integer variable i in the range 1 to 10,000". On each pass through this loop, at line 10, a fresh sample of N random values from the uniform distribution are generated which overwrites the existing variable samples, and line 11 uses the mean() function to calculate the average of these samples, which is saved to MeansOfSamples[i]. The square brackets denote array indexing: the first time through the loop the average is saved to MeansOfSamples[1], the second time it is saved to MeansOfSamples[2], and so on. Unlike C, R arrays start with 1. Line 14 displays a histogram of the array MeansOfSamples. Click on the Source button on the top-right corner of the source-file window; clicking this button causes all lines in the current file to be executed sequentially. You should get the following output in the Console window:

### Online Appendix 1.A.5.1.2 Code Output

> source(…)

Hello world!

The first line source(…) has been abbreviated as explained in the book, as its contents depend on the installation, which would vary from user to user.

The reassuring message came from line 3. Your RStudio plots window should look like that shown below:

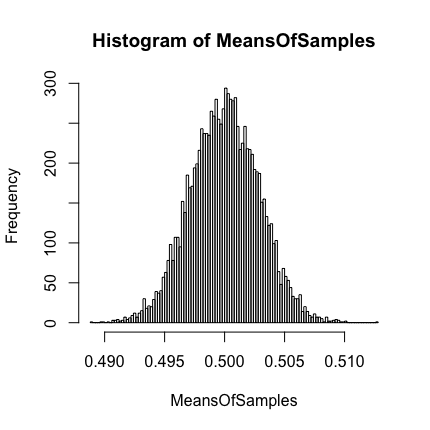


Figure 25

Screen-shot obtained by sourcing the file mainHelloWorld.R. Note the histogram plot, resulting from line 14.

Notice that the lower-right window is focused on the Plots tab, and a histogram of the variable MeansOfSamples is shown, which came from line 14, that looks like a normal distribution centered at 0.5, a result following from the central limit theorem of statistics, which states that the mean of a sufficiently large number of independent random samples, will be approximately normally distributed, regardless of the underlying distribution.

Notice that the top-right window is focused on the Environment tab, which is filled in with values (later we will demonstrate many uses for this window, basically a convenient way of examining variables without having to print them out). For example, one sees that i is 10,000 (the last value in the for loop), MeansOfSamples is an array with 10,000 values the first few are 0.5, 0.496, 0.501, 0.504, etc. all values being near 0.5. The variable N is 10,000 (from line 4) and finally the variable samples is an array with 10,000 values, the first few being 0.593, 0.586, 0.753, 0.964, etc., which correspond to the last execution of line 10.

If one looks carefully at the top-left end of the Plots window one sees that the blue *left* arrow button is active, indicating that there is another plot *hiding* behind the one being shown; clicking on this button Clicking on the blue *left* arrow should yield the following screen-shot, Figure 26.

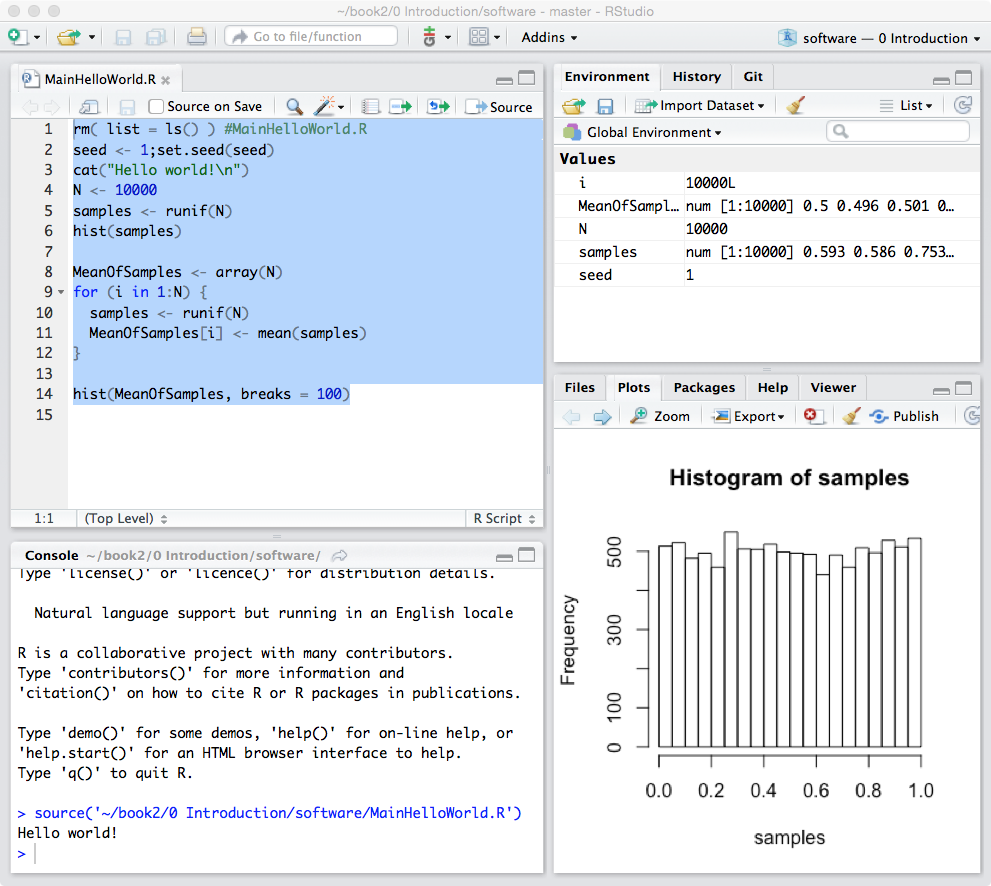


Figure 26

The histogram, which is due to line 6, is nearly flat, as one expects with a uniform distribution. This time the *right* blue arrow is active, clicking on which will bring up the previous figure. So one can switch between all the plots (not just two) generated by one's code. Click on the Clear All (with the "broom" symbol next to it) to delete all plots.

This ends the first tutorial. But before moving on, look at the Supplementary Material folder. It contains this document and a pdf file giving an overview of the R programming language.

1. At the time of writing, 12/7/2017, there is a bug in Sourcetree that is preventing this feature; so until this is fixed, the sure way of seeing the true state of the repository (i.e., is the local version identical to the web version?) is to quit Sourcetree and restart it. Clumsy, but it works! [↑](#footnote-ref-1)
2. This includes the entire development process of this package, i.e., intermediate versions; the final (i.e., most recent) version has not yet (as of December 17, 2017) been uploaded to CRAN. [↑](#footnote-ref-2)
3. A compiler is software the converts the more-or-less English-like instructions that make some sense to the programmer to sequences of binary instructions that the computer understands, but which would be gibberish to the programmer. When the program is run, the binary instructions are sequentially executed by the computer. [↑](#footnote-ref-3)
4. Allowing the period (.) as part of a variable name causes the author to cringe; but in life one cannot have everything one's way. [↑](#footnote-ref-4)