**ECO 274 LAB: Application of Statistics in Economics**

**Getting started with R and RStudio**

## 1.1 Why R?

R is a statistical software. R is not a programming language like C, C++, Python, Julia, Go, or Java. It was not created by software engineers for software development. Instead, it was developed by statisticians as an interactive environment for data analysis. The interactivity is an indispensable feature in data science because, as you will soon learn, the ability to quickly explore data is a necessity for success in this field. In R and R-studio (**Posit** will be the very soon future name), you can save your work as R-scripts (.R file) that can be easily executed at any moment. These R-scripts serve as a record of the analysis you performed, a key feature that facilitates reproducible work and open source (data) science research. If you are patient, you will come to appreciate the unequal power of R when it comes to data analysis, econometric model estimation and, specifically, data visualization and presentation.

*Some attractive features of R are:*

R is an open source and free to install, use, update, clone, modify, redistribute

R can handle complex and large data

R has technical merits in data science/machine learning

Getting help from the R community is super easy

R is popular and the standard language of choice for academics/econometrcians

R is popular with employers

Scripts and data objects can be shared seamlessly across platforms.

There is a large, growing, and active community of R users and, as a result, there are numerous resources for learning and asking questions.

It is easy for others to contribute add-ons which enables developers to share software implementations of new data science methodologies. This gives R users early access to the latest methods and to tools which are developed for a wide variety of disciplines, including economics, finance, ecology, molecular biology, social sciences, and geography, just to name a few examples.

Worldwide, Aug 2022 compared to a year ago:

| Rank | Change | Language | Share | Trend |
|---|---|---|---|---|
| 1 | | Python | 28.11 % | -2.6 % |
| 2 | | Java | 17.35 % | -0.9 % |
| 3 | | JavaScript | 9.48 % | +0.2 % |
| 4 | | C# | 7.08 % | +0.1 % |
| 5 | | C/C++ | 6.19 % | -0.3 % |
| 6 | | PHP | 5.47 % | -0.8 % |
| 7 | | R | 4.35 % | +0.6 % |
| 8 | ↑↑ | TypeScript | 2.79 % | +1.1 % |
| 9 | ↑↑ | Swift | 2.09 % | +0.5 % |
| 10 | ↓↓ | Objective-C | 2.03 % | +0.2 % |
| 11 | ↑ | Go | 2.03 % | +0.5 % |
| 12 | ↓↓↓ | Kotlin | 1.78 % | -0.0 % |
| 13 | ↑↑↑↑ | Rust | 1.58 % | +0.8 % |
| 14 | ↓ | Matlab | 1.52 % | +0.1 % |
| 15 | | Ruby | 1.15 % | +0.1 % |
| 16 | ↓↓ | VBA | 1.02 % | -0.2 % |
| 17 | ↑ | Dart | 0.83 % | +0.2 % |
| 18 | ↑↑ | Ada | 0.78 % | +0.2 % |
| 19 | | Scala | 0.73 % | +0.2 % |
| 20 | ↑ | Lua | 0.64 % | +0.1 % |
| 21 | ↓↓↓↓ | Visual Basic | 0.6 % | -0.2 % |
| 22 | | Abap | 0.48 % | -0.0 % |
| 23 | ↑↑ | Julia | 0.43 % | +0.1 % |
| 24 | ↓ | Groovy | 0.41 % | -0.1 % |
| 25 | ↓ | Perl | 0.32 % | -0.1 % |

**Module 1**

In this tutorial we'll learn how to begin programming with R using RStudio. We'll install R, and RStudio R-Studio, an extremely popular development environment for R. We'll learn the key RStudio features in order to start programming in R on our own.

*1. Install R*

R is available to download from the official R website (CRAN-comprehensive R archive network). Look for this section of the web page:

**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.

Search R download for windows/mac/Linux on your browser

Pick one of the links associated with your operating system

Click on the (**Download R4.1.3 for windows** if you are windows user) and Download R for your system

**Download R 4.1.3 for Windows** (87 megabytes, 32/64 bit)

Installation and other instructions
New features in this version

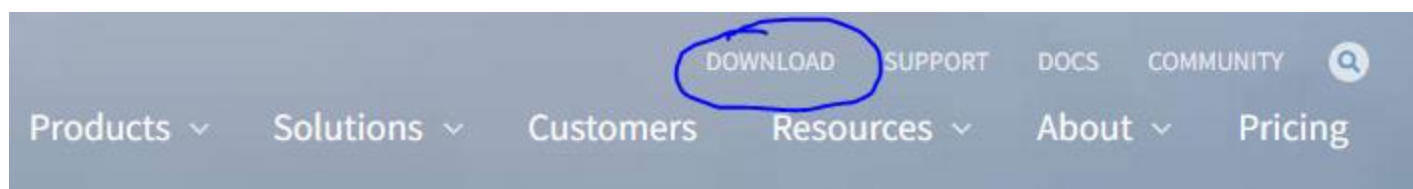Follow the typical downloading and installation prompts associated with your computer system

### 2. Install RStudio

Already have R the program on your computer, if you don't do that first.

Now that R is installed, we can install RStudio. Search R-studio download on google. Navigate to the RStudio downloads page. When we reach the RStudio downloads page, let's click the "Download" button of the RStudio Desktop Open-Source License Free option:

Go to https://www.rstudio.com/

In the right hand corner click download

DOWNLOAD    SUPPORT    DOCS    COMMUNITY

Products ⌄    Solutions ⌄    Customers    Resources ⌄    About ⌄    Pricing

Next, 4. Pick a version. Free has all of the tools you will need for this course and most likely all of the classes you will take in Undergrad, and research in Honours, Masters, PhD, PostDoc, etc

| | RStudio Desktop | RStudio Desktop | 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 |
| Integrated Tools for R | ✓ | ✓ | ✓ | ✓ |
| Priority Support | | ✓ | | ✓ |
| Access via Web Browser | | | ✓ | ✓ |
| Enterprise Security | | | | ✓ |
| Project Sharing | | | | ✓ |
| Manage Multiple R Sessions & Versions | | | | ✓ |
| Admin Dashboard | | | | ✓ |
| Load Balancing | | | | ✓ |
| Auditing and Monitoring | | | | ✓ |
| Data Connectivity | | | | ✓ |
| Launcher | | | | ✓ |
| Tutorial API | | | | ✓ |
| License | AGPL | Commercial | AGPL | Commercial |

Our operating system is usually detected automatically and so we can directly download the correct version for our computer by clicking the "Download RStudio" button. Follow your computer prompts to download and install

## 3. The first look at R console

Interactive data analysis usually occurs on the R console that executes commands as you type them. There are several ways to gain access to an R console. One way is to simply start R on your computer. The console looks something like this:

As a quick example, try using the console to calculate a multiplication of 15 times 5:

15 * 5

#> [1] 75

Note that in this book, grey boxes are used to show R code typed into the R console. The symbol #> is used to denote what the R console outputs.


### 4. Getting Started with RStudio

RStudio is an open-source tool for programming in R. RStudio is a flexible tool that helps you create readable analyses, and keeps your code, images, comments, and plots together in one place. It's worth knowing about the capabilities of RStudio for data analysis and programming in R.

### 4.1. First Look at RStudio

When we open R-Studio for the first time, we'll probably see a layout like this:
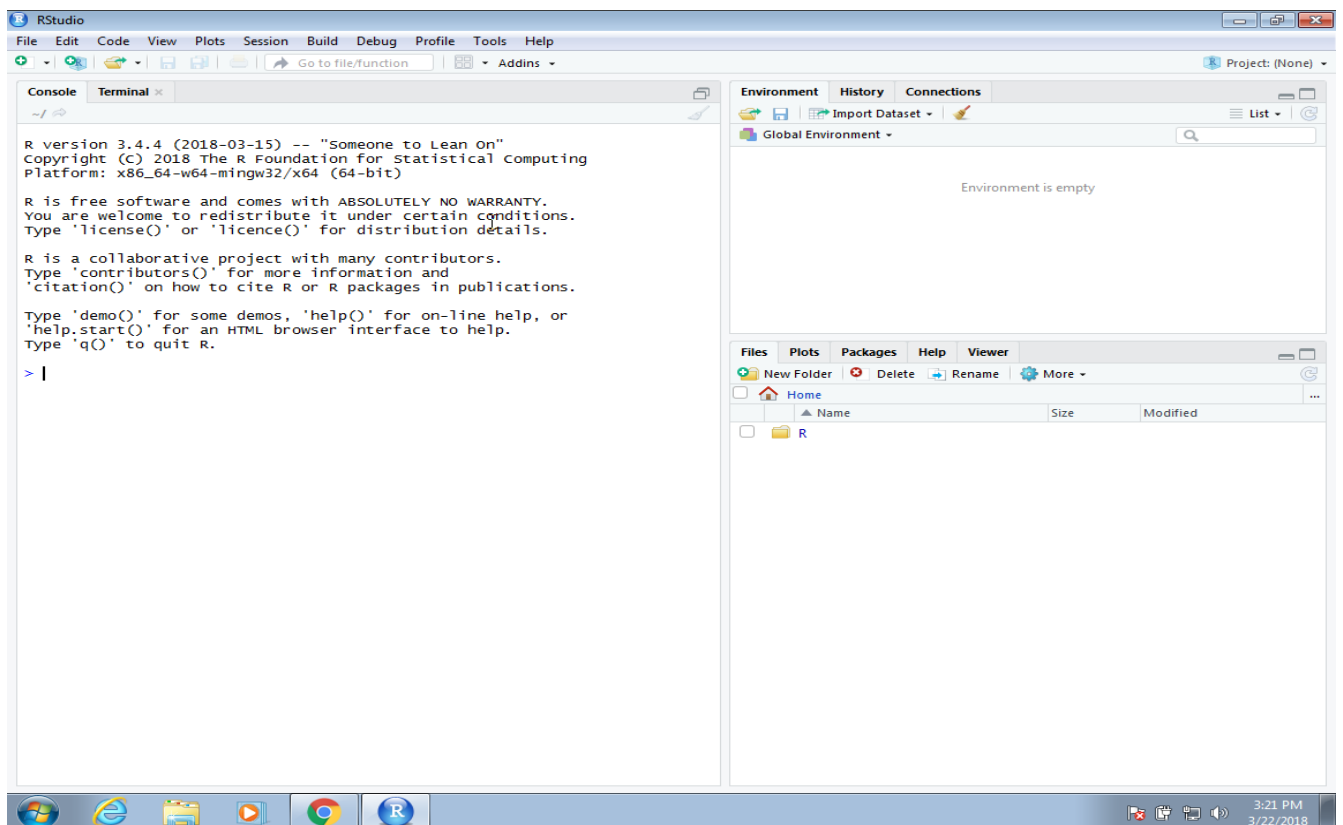


**The panes**

When you start RStudio for the first time, you will see three panes. The left pane shows the R console. On the right, the top pane includes tabs such as Environment and History, while the bottom pane shows

five tabs: File, Plots, Packages, Help, and Viewer (these tabs may change in new versions). You can click on each tab to move across the different features.

When we open RStudio, R is launched as well. A common mistake by new users is to open R instead of RStudio. To open RStudio, search for RStudio on the desktop, and pin the RStudio icon to the preferred location (e.g. Desktop or toolbar).

### The Console

Let's start off by introducing some features of the Console. The Console is a tab in RStudio where we can run R code. Notice that the window pane where the console is located contains three tabs: Console, Terminal and Jobs (this may vary depending on the version of RStudio in use). We'll focus on the Console for now.

When we open RStudio, the console contains information about the version of R we're working with. Scroll down, and try typing a few expressions like this one. Press the enter key to see the result.

we can use the console to test code immediately. When we type an expression like 1 + 2, we'll see the output below after hitting the enter key.

### The Global Environment

We can think of the global environment as our workspace. During a programming session in R, any variables we define, or data we import and save in a dataframe, are stored in our global environment. In RStudio, we can see the objects in our global environment in the Environment tab at the top right of the interface:
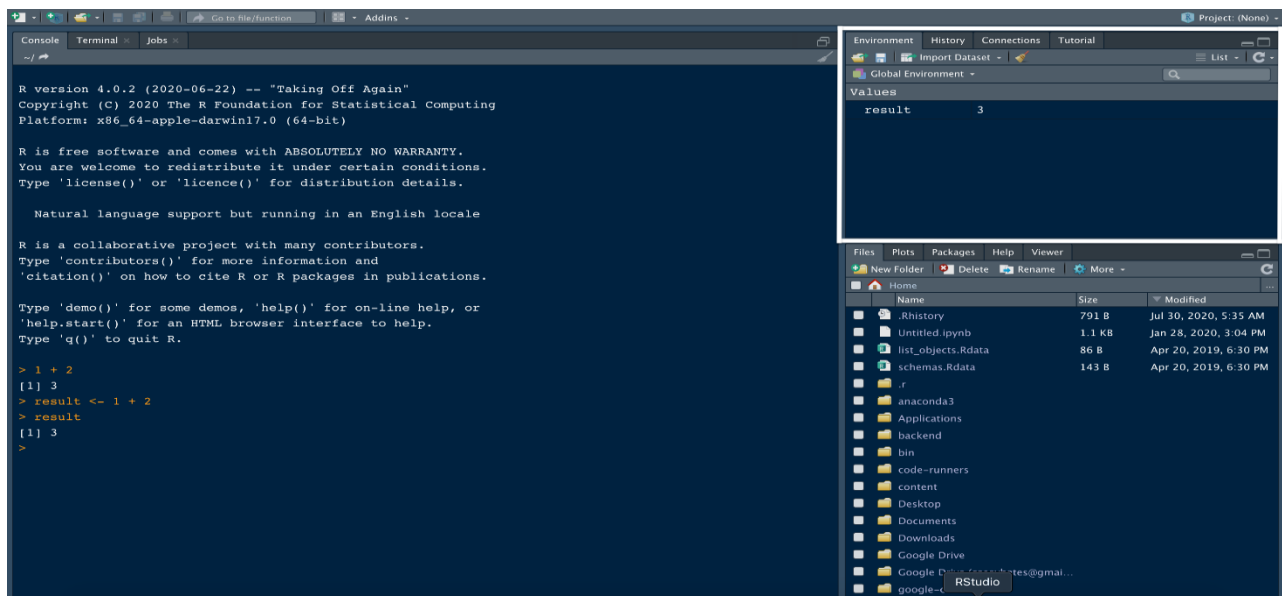


Sometimes, having too many named objects in the global environment creates confusion. Maybe we'd like to remove all or some of the objects. To remove all objects, click the broom icon at the top of the window. Finally, we will create an R-script (how? See the next section) and that will bring us to the following 4 panes:
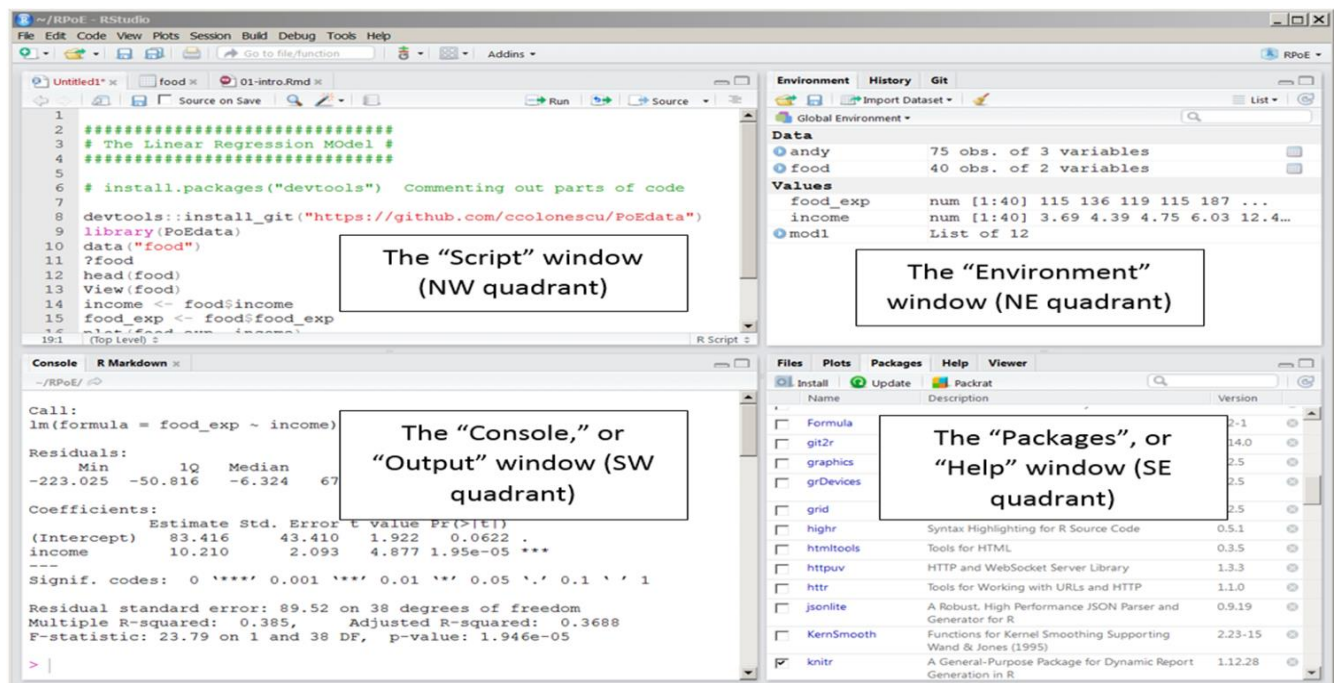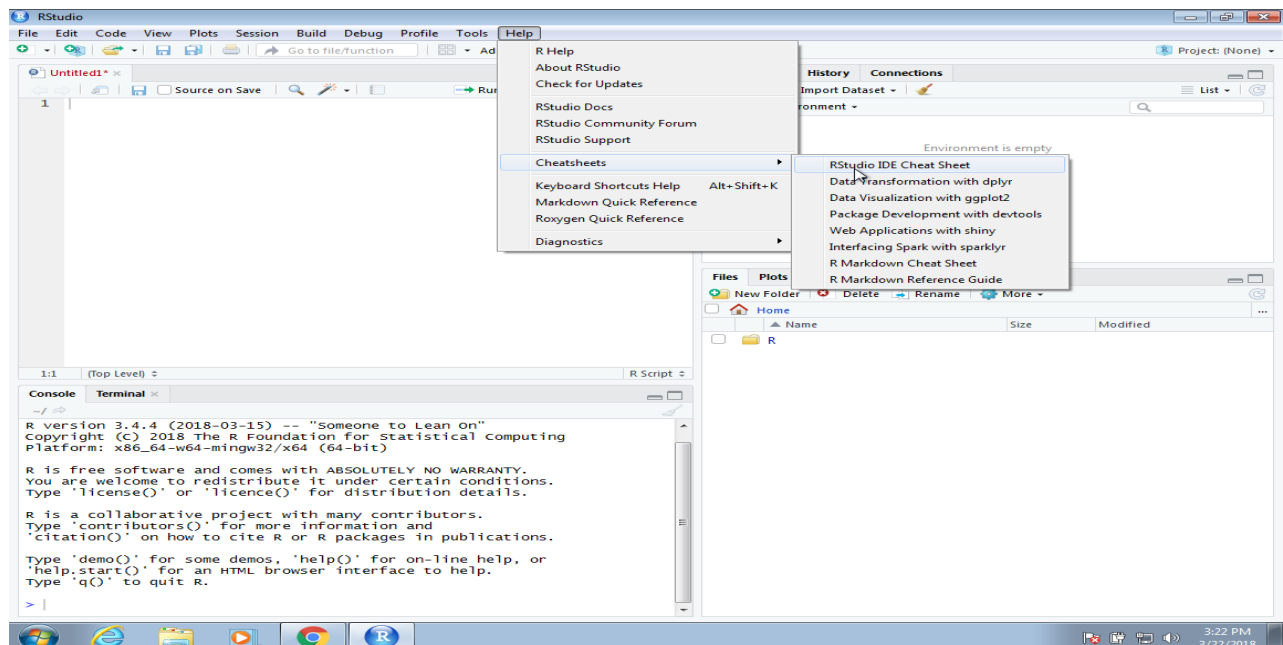
## Scripts (R-Scripts)

One of the great advantages of R over point-and-click analysis software is that you can save your work as scripts. You can edit and save these scripts using a text editor. The material in this note was developed using the interactive integrated development environment (IDE) RStudio. RStudio includes an editor with many R specific features, a console to execute your code, and other useful panes, including one to show figures. To start a new script, you can click on File, then New File, then R Script.
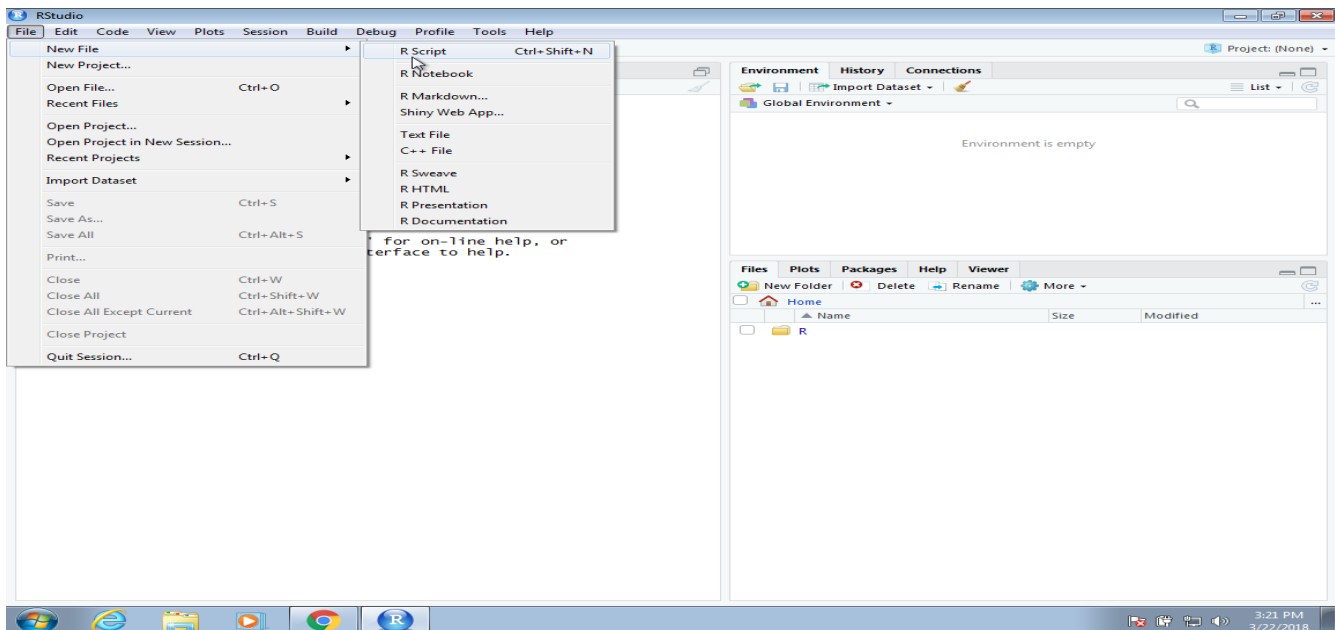


This starts a new pane on the left and it is here where you can start writing your script.
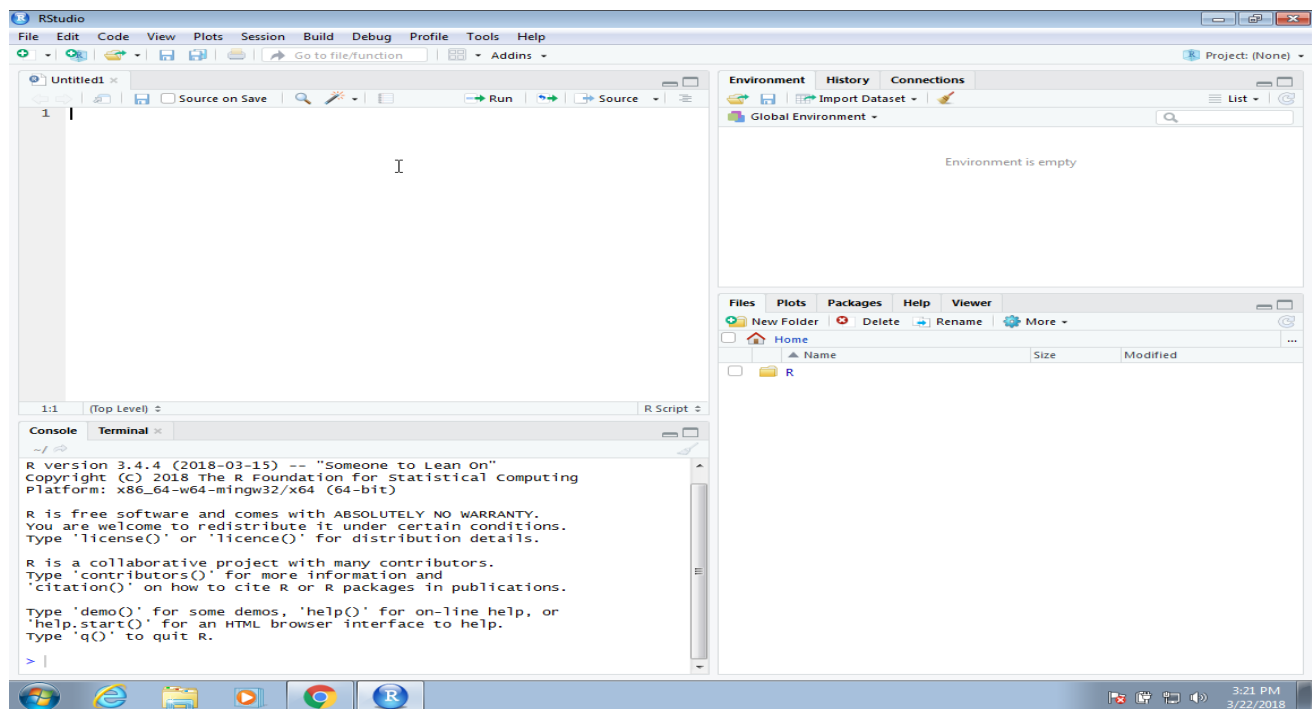
we just showed how to use the mouse to start a new script, but you can also use a key binding (key board short cut): Ctrl+Shift+N on Windows and command+shift+N on the Mac. Although in this tutorial we often show how to use the mouse, we highly recommend that you memorize key bindings for the operations you use most. RStudio provides a useful cheat sheet with the most widely used commands. You can get it from RStudio directly:



**Running commands while editing scripts and saving you work file**

Let's start by opening a new script as we did before. A next step is to give the script a name. We can do this through the editor by saving the current new unnamed script. To do this, click on the save icon or

use the key binding Ctrl+S on Windows and command+S on the Mac. When you ask for the document to be saved for the first time, RStudio will prompt you for a name. A good convention is to use a descriptive name, with lower case letters, no spaces, only hyphens to separate words, and then followed by the suffix .R. We will call this script **lab_476_d1**.R.
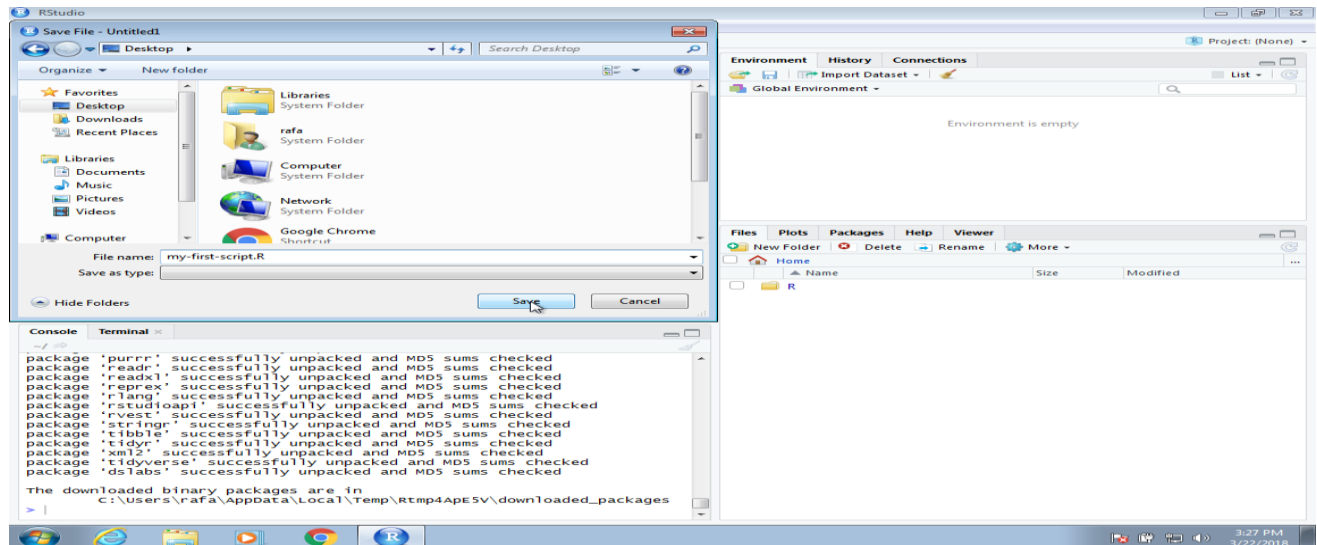


Now we are ready to start our coding and editing our first R-script.

**Ref:** This document is prepared from various sources, and sources that need to be acknowledged are not limited to books, workshop materials, and R-tutorials but include various internet materials.