ECO 476 LAB VI: Application of Econometrics II

This lab course (ECO476) and the following short lecture note are about learning how to use the statistical software R for econometric applications. It will provide you an understanding of the basic structure of the program and the basic econometric/statistical functions, which are needed to excel introductory courses in econometrics. The main focus is on the empirical implementation of econometric theory and is designed for a seamless transition from theory to practice. In many areas of applied statistics, R is the most widely used software package. In other areas, such as econometrics and finance, it is quickly catching up to commercial packages.

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:

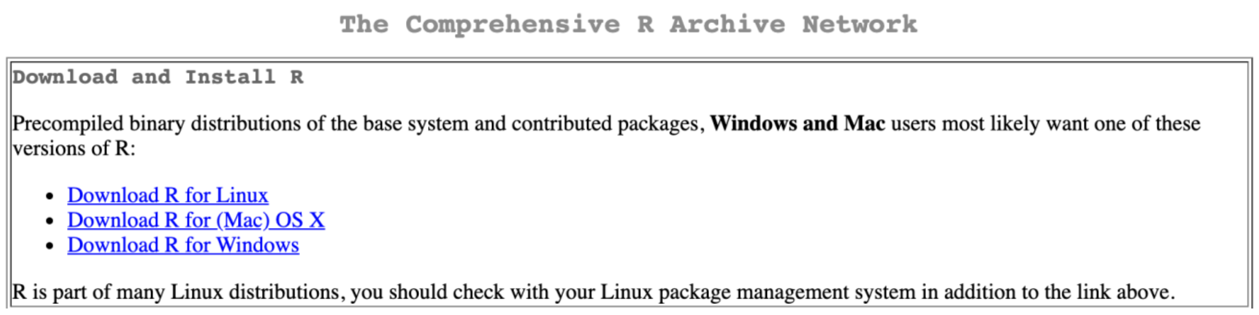
Rank	Change	Language	Share	Trend
1		Python	28.11 %	-2.6 %
2		Java	17.35 %	-0.9 %
3		JavaScript	9.48 %	+0.2 %
4		C#	7.08 %	+0.1 %
5		C/C++	6.19 %	-0.3 %
6		PHP	5.47 %	-0.8 %
7		R	4.35 %	+0.6 %
8	↑↑	TypeScript	2.79 %	+1.1 %
9	↑↑	Swift	2.09 %	+0.5 %
10	↓↓	Objective-C	2.03 %	+0.2 %
11	↑	Go	2.03 %	+0.5 %
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

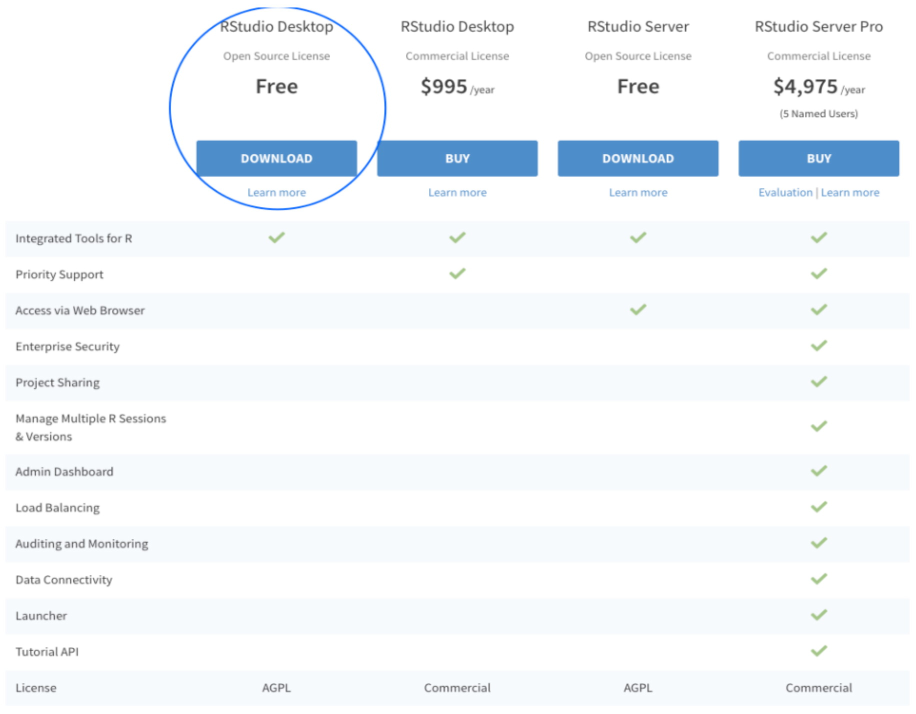
Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- 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.

2. Install RStudio

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:

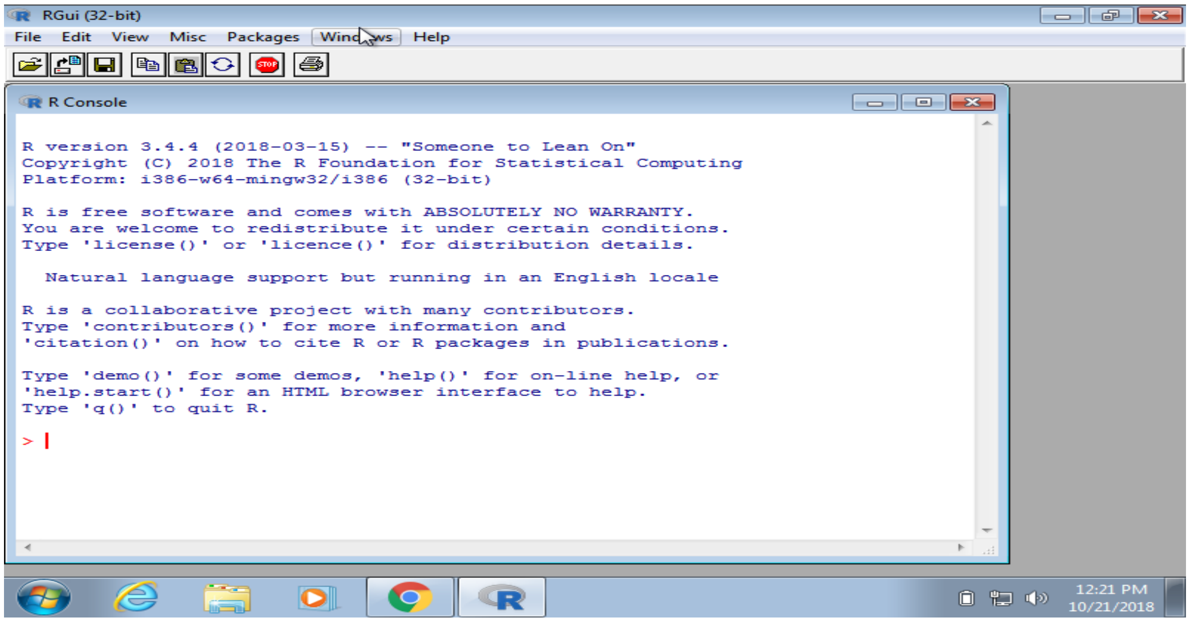


	RStudio Desktop Open Source License Free	RStudio Desktop Commercial License $995 /year	RStudio Server Open Source License Free	RStudio Server Pro Commercial License $4,975 /year (5 Named Users)
	DOWNLOAD Learn more	BUY Learn more	DOWNLOAD Learn more	BUY Evaluation | Learn more
Integrated Tools for R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

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:



```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: i386-w64-mingw32/i386 (32-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.

> |
```

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

```
15 * 5
```

```
#> [1] 75
```

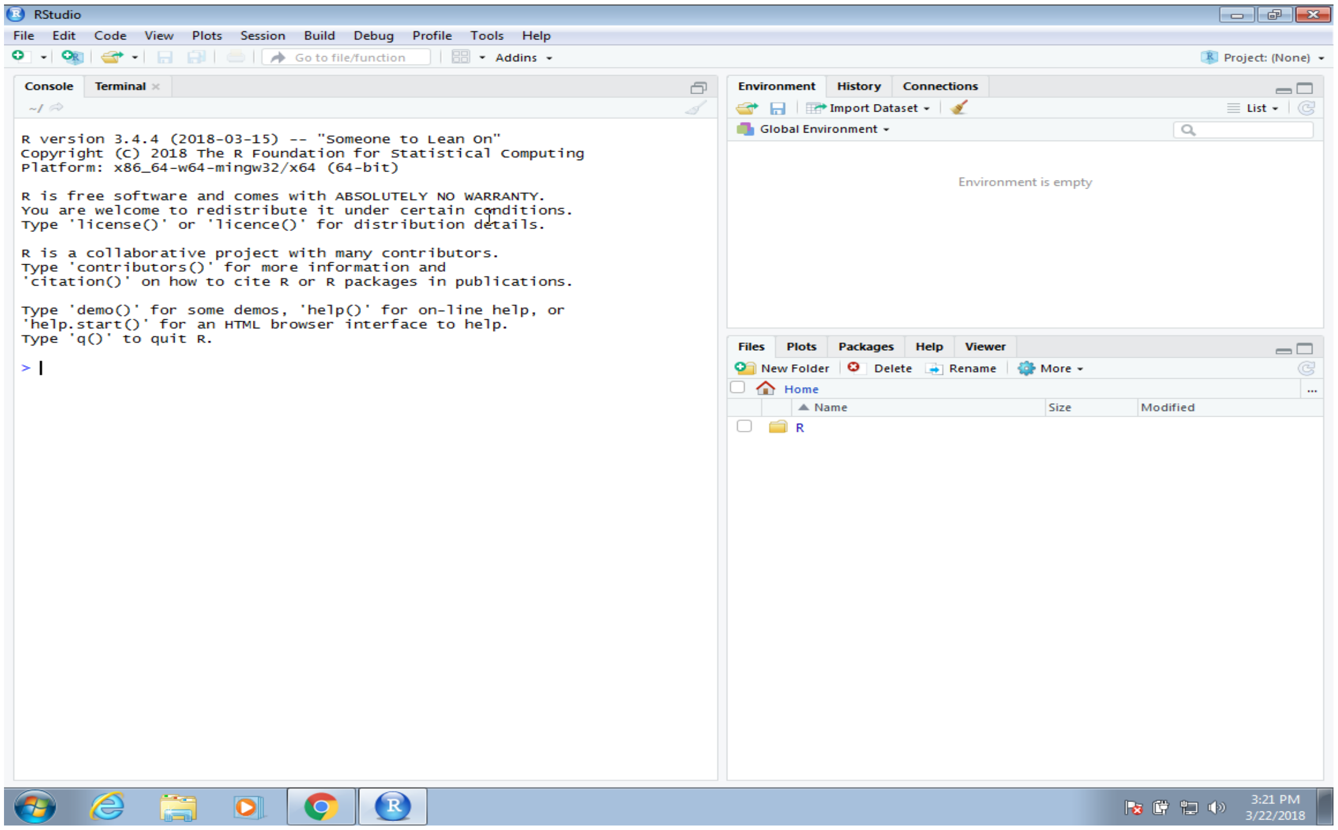
Note that in this book, grey boxes are used to show R code typed into the R console. The symbol #> is used to denote what the R console outputs.

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:



```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

> |
```

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

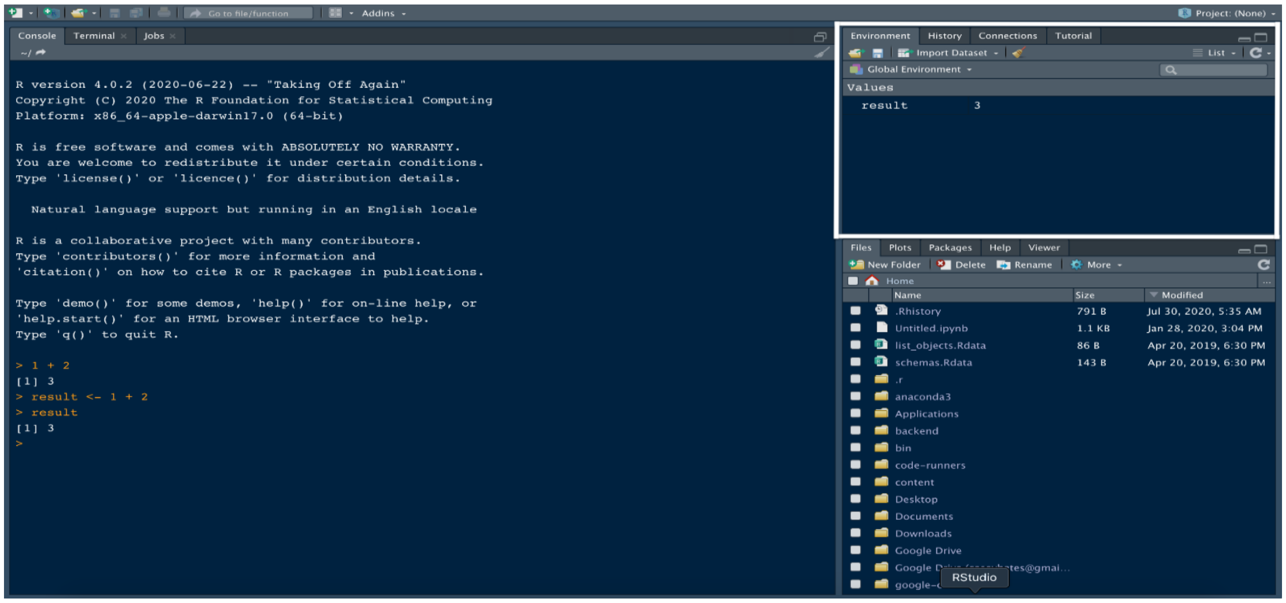
Let’s start off by introducing some features of the Console. The Console is a tab in RStudio where we can run R code. Notice that the window pane where the console is located contains three tabs: Console, Terminal and Jobs (this may vary depending on the version of RStudio in use). We’ll focus on the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

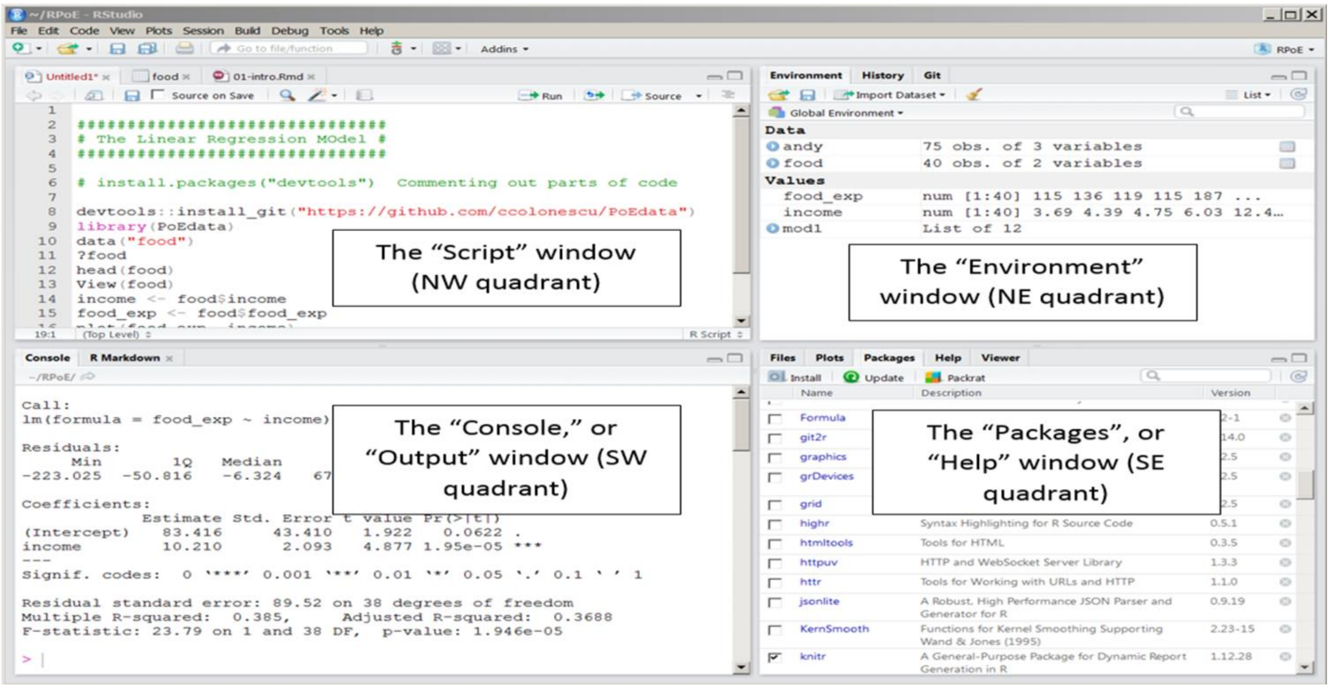


```
R version 4.0.2 (2020-06-22) -- "Taking Off Again"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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.

> 1 + 2
[1] 3
> result <- 1 + 2
> result
[1] 3
>
```

```
Values
  result        3
```

Name	Size	Modified
.Rhistory	791 B	Jul 30, 2020, 5:35 AM
Untitled.ipynb	1.1 KB	Jan 28, 2020, 3:04 PM
list_objects.Rdata	86 B	Apr 20, 2019, 6:30 PM
schemas.Rdata	143 B	Apr 20, 2019, 6:30 PM
.r		
anaconda3		
Applications		
backend		
bin		
code-runners		
content		
Desktop		
Documents		
Downloads		
Google Drive		
Google D[illegible]tes@gmai...		
google-[illegible]		

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:



```
#################################
# The Linear Regression MOdel #
#################################

# install.packages("devtools")  Commenting out parts of code

devtools::install_git("https://github.com/ccolonescu/PoEdata")
library(PoEdata)
data("food")
?food
head(food)
View(food)
income <- food$income
food_exp <- food$food_exp
```

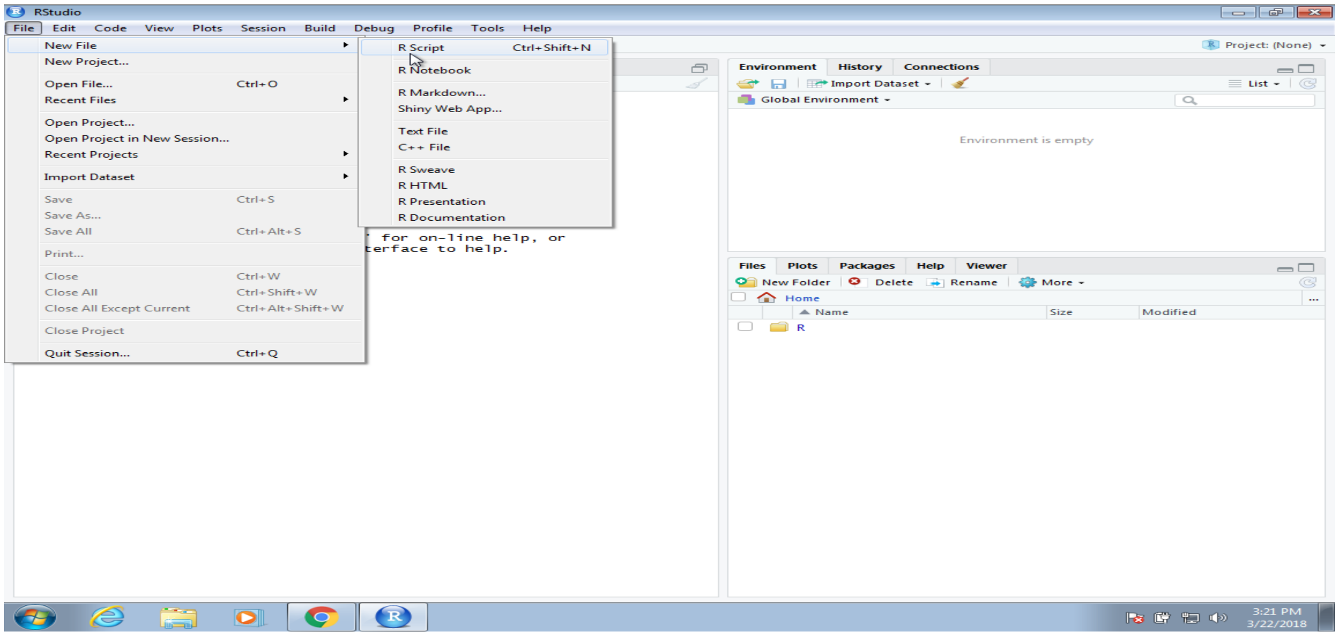
```
Data
  andy        75 obs. of 3 variables
  food        40 obs. of 2 variables
Values
  food_exp    num [1:40] 115 136 119 115 187 ...
  income      num [1:40] 3.69 4.39 4.75 6.03 12.4…
  mod1        List of 12
```

```
Call:
lm(formula = food_exp ~ income)

Residuals:
     Min       1Q   Median
-223.025  -50.816   -6.324    67

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   83.416     43.410   1.922   0.0622 .
income        10.210      2.093   4.877 1.95e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 89.52 on 38 degrees of freedom
Multiple R-squared:  0.385,     Adjusted R-squared:  0.3688
F-statistic: 23.79 on 1 and 38 DF,  p-value: 1.946e-05

> |
```

	Name	Description	Version
☐	Formula		2-1
☐	git2r		14.0
☐	graphics		2.5
☐	grDevices		2.5
☐	grid		2.5
☐	highr	Syntax Highlighting for R Source Code	0.5.1
☐	htmltools	Tools for HTML	0.3.5
☐	httpuv	HTTP and WebSocket Server Library	1.3.3
☐	httr	Tools for Working with URLs and HTTP	1.1.0
☐	jsonlite	A Robust, High Performance JSON Parser and Generator for R	0.9.19
☐	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-15
✓	knitr	A General-Purpose Package for Dynamic Report Generation in R	1.12.28

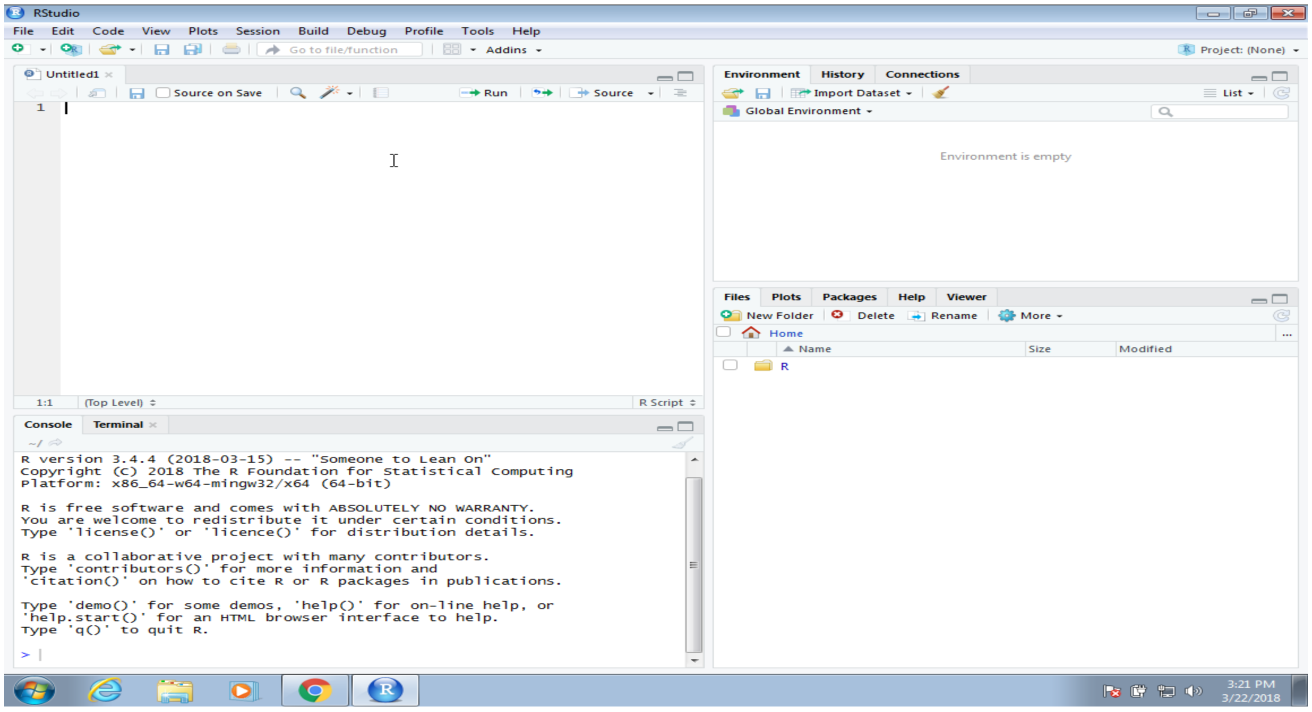
Scripts (R-Scripts)

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.



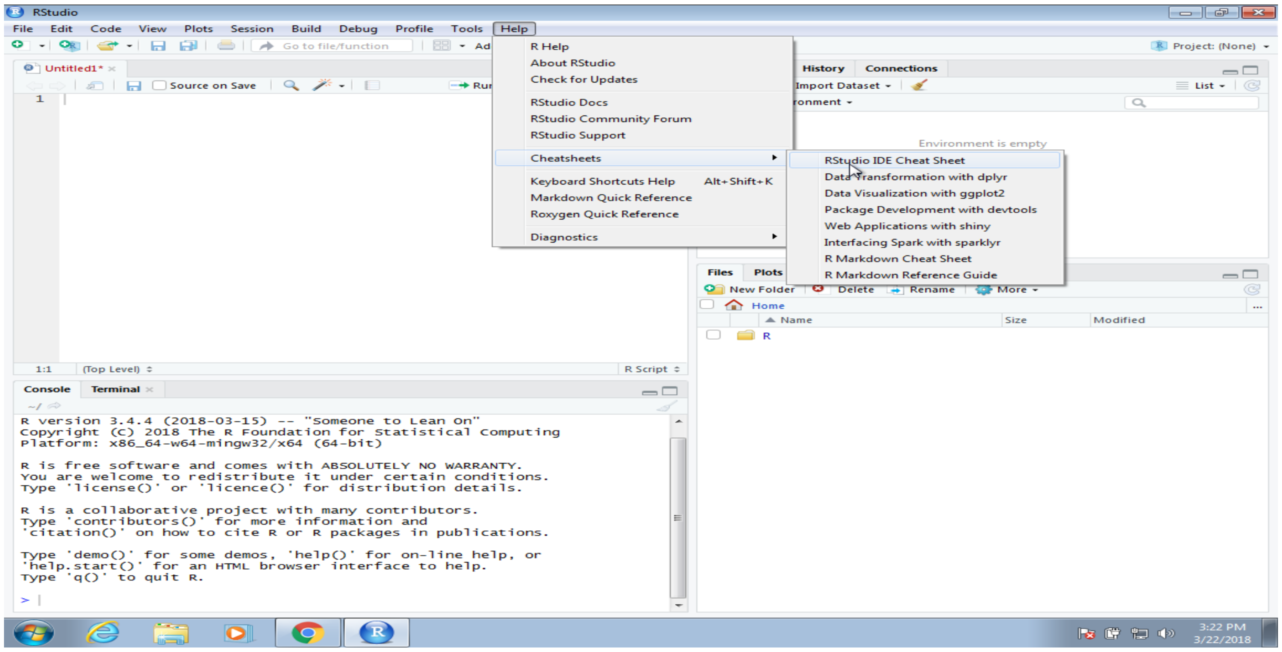
```
' for on-line help, or
terface to help.
```

This starts a new pane on the left and it is here where you can start writing your script.



```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

> |
```

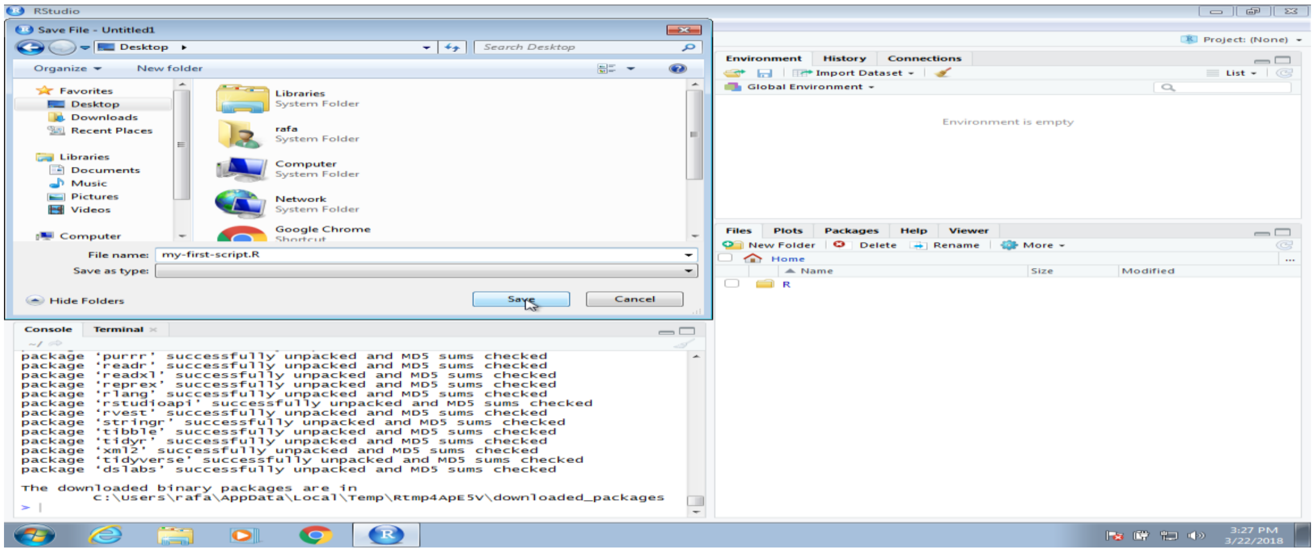
we just showed how to use the mouse to start a new script, but you can also use a key binding (key board short cut): Ctrl+Shift+N on Windows and command+shift+N on the Mac. Although in this tutorial we often show how to use the mouse, we highly recommend that you memorize key bindings for the operations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



```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

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.

>
```

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.



File name: my-first-script.R
Save as type:

```
package 'purrr' successfully unpacked and MD5 sums checked
package 'readr' successfully unpacked and MD5 sums checked
package 'readxl' successfully unpacked and MD5 sums checked
package 'reprex' successfully unpacked and MD5 sums checked
package 'rlang' successfully unpacked and MD5 sums checked
package 'rstudioapi' successfully unpacked and MD5 sums checked
package 'rvest' successfully unpacked and MD5 sums checked
package 'stringr' successfully unpacked and MD5 sums checked
package 'tibble' successfully unpacked and MD5 sums checked
package 'tidyr' successfully unpacked and MD5 sums checked
package 'xml2' successfully unpacked and MD5 sums checked
package 'tidyverse' successfully unpacked and MD5 sums checked
package 'dslabs' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\rafa\AppData\Local\Temp\Rtmp4ApE5V\downloaded_packages
>
```

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

Module 2

You should provide/write detail about this R-script using hashtags so that anyone (even you) can recognize what this script is for. R treats the hashtag character, # , in a special way; R will not run anything that follows a hashtag on a line. This makes hashtags very useful for adding comments and annotations to your code. Humans will be able to read the comments, but your computer will pass over them.

```
 The Absolute Basics
 #add
 1 + 1
 #subtract them
 8 – 4
 #divide
 13/2
 #multiply
 4*pi
 #exponentiation
 2^10
Strings (text)
'Econometrics is awesome‘
#R delimits strings with EITHER double or single quotes.
#  There is only a very minimal difference
"Econometrics is still awesome“

Variables
Creating object
x = 42
x / 2
#if we assign something else to x,
#  the old value is deleted
```

```
x = "Melody to Funkytown!"

X

x = 5

x == 5

foo = 3

bar = 5

foo.bar = foo + bar

foo.bar
```

Logical Comparison

```
3 < 4
3 > 4
#contrast with 3 = 4; see section about variables below
3 == 4
#!= means "not equal to"
3 != 4
4 >= 5
4 <= 5
2 + 2 == 5
10 - 6 == 4
```

Lists

```
x = list(TRUE, 1, "Frank")

x = list(c(1, 2), c("a", "b"), c(TRUE, FALSE), c(5L, 6L))

y = list(list(1, 2, 3), list(4:5), 6)
```

Matrices

```
# Generating matrix A from one vector with all values

v <- c(2,-4,-1,5,7,0)

## The <- is called the assignment operator. This operator assigns
values to variables. The command above is translated into a sentence
as: The result variable gets the value of one plus two. C stand for
concatenation.


( A <- matrix(v,nrow=2) )

# Generating matrix A from two vectors corresponding to rows:

row1 <- c(2,-1,7); row2 <- c(-4,5,0)
```

```
( A <- rbind(row1, row2) )

# Generating matrix A from three vectors corresponding to columns:

 col1 <- c(1,6); col2 <- c(2,3); col3 <- c(7,2)

( US40Chart <- cbind(col1, col2, col3) )


# Giving names to rows and columns:
colnames(US40Chart) <- c("Drake","Maroon5","Dua Lipa")
rownames(US40Chart) <- c("weekTop","totPeak")
US40Chart
# Diaginal and identity matrices:
diag( c(4,2,6) )
diag( 3 )
# Indexing for extracting elements (still using US40Chart from above):
US40Chart[2,1]
US40Chart[,2]
US40Chart[,c(1,3)]
US40Chart[2,c(1,2)]
US40Chart[2,]
```

Matrices

```
# Generating matrix A and B

A <- matrix( c(2,-4,-1,5,7,0), nrow=2)

B <- matrix( c(2,1,0,3,-1,5), nrow=2)

A

B

A*B

# Transpose:

(C <- t(B) )

# Matrix multiplication:

 (D <- A %*% C )

# Inverse:

 solve(D)


# Giving names to rows and columns:
B = matrix(c(2, 4, 3, 1, 5, 7), nrow=3,ncol=2)
C = matrix(c(7, 4, 2),nrow=3,ncol=1)
```

```
# combine the columns of B and C with cbind
cbind(B, C)
# combine the rows of two matrices if they have the same number of
columns
D = matrix(c(6, 2),nrow=1,ncol=2)
rbind(B, D)
```

#Data frame

# A data frame is a kind of list. More precisely, a data frame is a list of items with

# the same structure; e.g., all vectors or matrices included in a data frame have the

# same dimensions (i.e., number of rows and columns).

topHit <- c(1, 3, 5)

s <- c("Drake", "Swift", "Selina")

at40 <- c(TRUE, FALSE, TRUE)

df <- data.frame(topHit, s, at40)

# Define one x vector for all:

year   <- c(2008,2009,2010,2011,2012,2013)

# Define a matrix of y values:

product1<-c(0,3,6,9,7,8); product2<-c(1,2,3,5,9,6); product3<-c(2,4,4,2,3,2)

sales_mat <- cbind(product1,product2,product3)

rownames(sales_mat) <- year

# The matrix looks like this:

sales_mat

# Create a data frame and display it:

sales <- as.data.frame(sales_mat)

sales

```
# Accessing a single variable:

sales$product2

# Generating a new  variable in the data frame:

sales$totalv1 <- sales$product1 + sales$product2 + sales$product3

# Subset: all years in which sales of product 3 were >=3

subset(sales, product3>=3)

# Note: "sales" is defined in Data-frames.R, so it has to be run first!

# save data frame as RData file (in the current working directory)

save(sales, file = "oursalesdata.RData")

write.csv(sales,"sales.csv")
```

Now close and restart R-studio from your original lab_476 folder and import sales csv data file

```
df_sales <- read.csv(file = "sales.csv", header = TRUE, stringsAsFactors=FALSE)
```

Module 3

Getting Help & Packages

R truly exceptional with its vast library of user-contributed packages

All of the packages available to R users (through CRAN) are completely free of charge

install.packages("readxl")

library("readxl")

library("xlsx")

Importing Data in R Script

# Import the sales csv dataset

Data_sales <- read.csv(file.choose(), header=T)

df_sales <- read.csv(file = "sales.csv", header = TRUE, stringsAsFactors=FALSE)

# Import the flower xls|xlsx dataset

my_data <- read_excel("flower.xls")

# Import the flower.txt dataset

flowers <- read.table(file = 'flower.txt', header = TRUE, sep = "\t", stringsAsFactors = TRUE)

use the str() function to return a compact and informative summary of your data frame.

If we just wanted to see the names of our variables (columns) in the data frame we can use the names() function which will return a character vector of the variable names.

To remove everything from the environment: rm(list=ls())

Import data from online

df <- read.csv("https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data",header = FALSE)

View(df)

# Activate the `foreign` library

library(foreign)

# Read the SPSS data

```
mySPSSData <- read.spss("example.sav", to.data.frame=TRUE, use.value.labels=FALSE)

# Read Stata data into R

mydata <- read.dta("<Path to file>")


# Activate the `sas7bdat` library

library(sas7bdat)

# Read in the SAS data

mySASData <- read.sas7bdat("example.sas7bdat")
```

Economics students new to both econometrics and R may find the introduction to both challenging. However, if their text is "Introductory Econometrics: A Modern Approach, 6e" by Jeffrey M. Wooldridge, they are in luck! The wooldridge data package aims to lighten the task by easily loading any data set from the text. The package contains full documentation for every data set and all data have been compressed to a fraction of their original size. Just install the package, load it, and call the data you wish to work with.

```
library(wooldridge)

data("wage1")
```

For Data Sets from "Basic Econometrics, 5ed" by Damodar N. Gujarati and Dawn Porter

```
install.packages('devtools')
devtools::install_github('https://github.com/brunoruas2/gujarati')

require(gujarati)

data('Table1_1')

View(Table1_1)

df11 <- gujarati::Table1_1

##https://rdrr.io/github/brunoruas2/gujarati/man/
```

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.