R Bootcamp: Interacting with R/RStudio

August 23-24, 2021

UBC MFRE | Master of Food and Resource Economics

Learning Objectives

- Become familiar with RStudio’s interface
- Know how to run codes in R using scripts or directly in console
- Understand what a package is
- Install and load a package
- Use the {pacman} package to install and load packages
- Create objects using the assignment operator
- Add comments

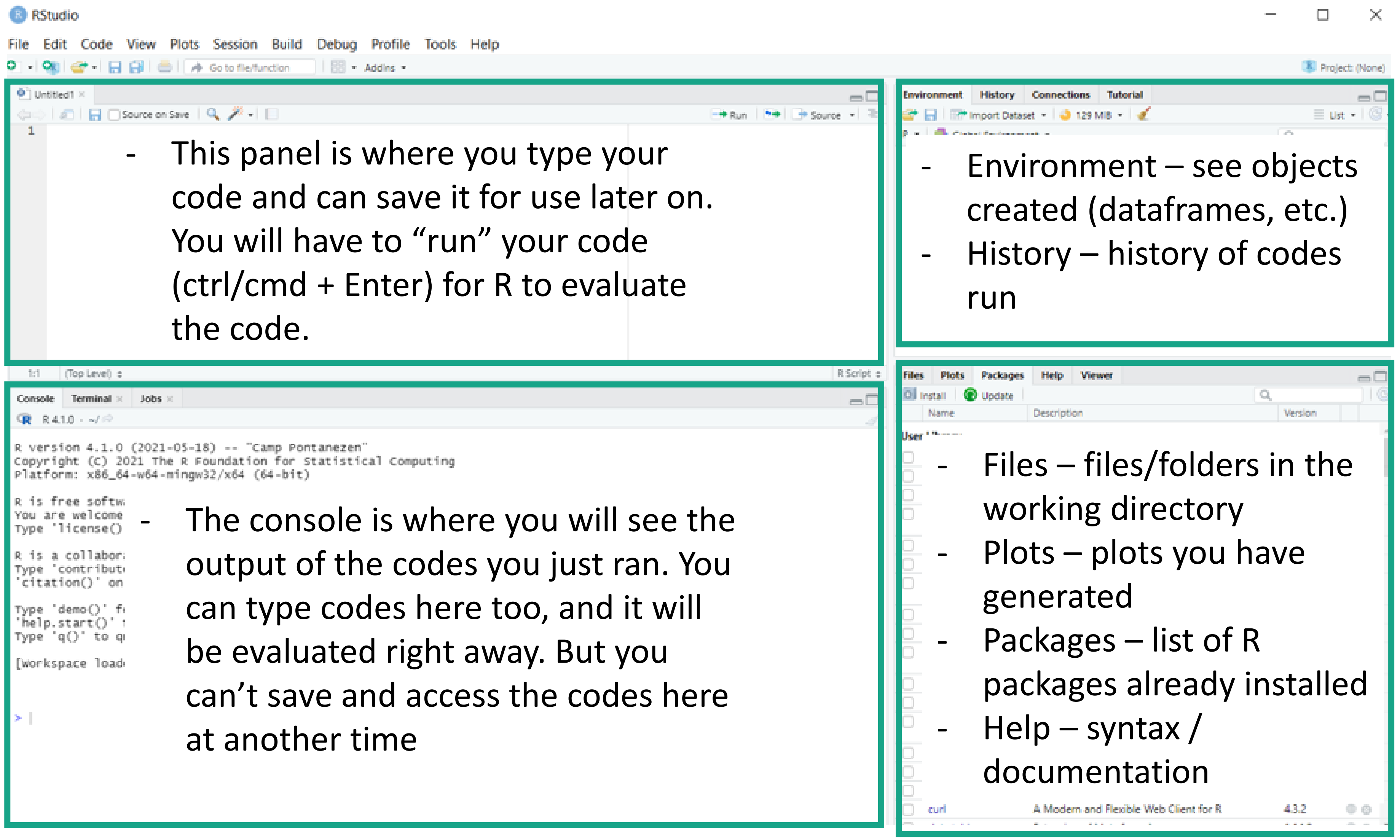
UBC
MFRE

R and RStudio

- R is a programming language and a free statistical software environment
- RStudio is an integrated development environment (IDE) for R programming.
- When I say R, I mean we will code using R in RStudio.
- **Always remember: R is the programming language. RStudio is the IDE.**

UBC
MFRE

RStudio Interface



- This panel is where you type your code and can save it for use later on. You will have to “run” your code (ctrl/cmd + Enter) for R to evaluate the code.

- Environment – see objects created (dataframes, etc.)
- History – history of codes run

```
R version 4.1.0 (2021-05-18) -- "Camp Pontanezen"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free softw
You are welcome
Type 'license()

R is a collabor
Type 'contribut
'citation()' on

Type 'demo()' f
'help.start()' 
Type 'q()' to q

[Workspace load

> |
```

- The console is where you will see the output of the codes you just ran. You can type codes here too, and it will be evaluated right away. But you can’t save and access the codes here at another time

- Files – files/folders in the working directory
- Plots – plots you have generated
- Packages – list of R packages already installed
- Help – syntax / documentation

curl	A Modern and Flexible Web Client for R	4.3.2

UBC
MFRE

RStudio Interface

- You can also change some settings (Tools > Global Options)
  - Theme (Appearance)
  - Parentheses (Code > Display tab > check rainbow parentheses)
  - Code wrapping (Code > Editing tab > check Soft-wrap R source files)

UBC
MFRE

Interacting with R

- Two ways to code
  - In console (can’t save codes for access later on)
  - In Script files (source editor)

- To execute/run our code from the source editor
  - ctrl + enter in Windows
  - cmd + return in Mac

- Note: .R scripts is analogous to .do file in Stata

UBC
MFRE

Interacting with R

- If you see the “>” prompt in the source editor, that means R is ready to accept new commands

- If you see the “+” prompt, that means R is waiting for you to enter more text, i.e. your code is still incomplete
  - Missing a parenthesis or quotation
  - To cancel this incomplete command, press *Esc* on your keyboard while inside the console window

UBC
MFRE

Packages

- “A package bundles together code, data, documentation, and tests, and is easy to share with others”
  - If you are struggling to accomplish a certain task, it is very possible that someone has already solved your problem and created a package for it.

UBC
MFRE

https://r-pkgs.org/intro.html

Packages

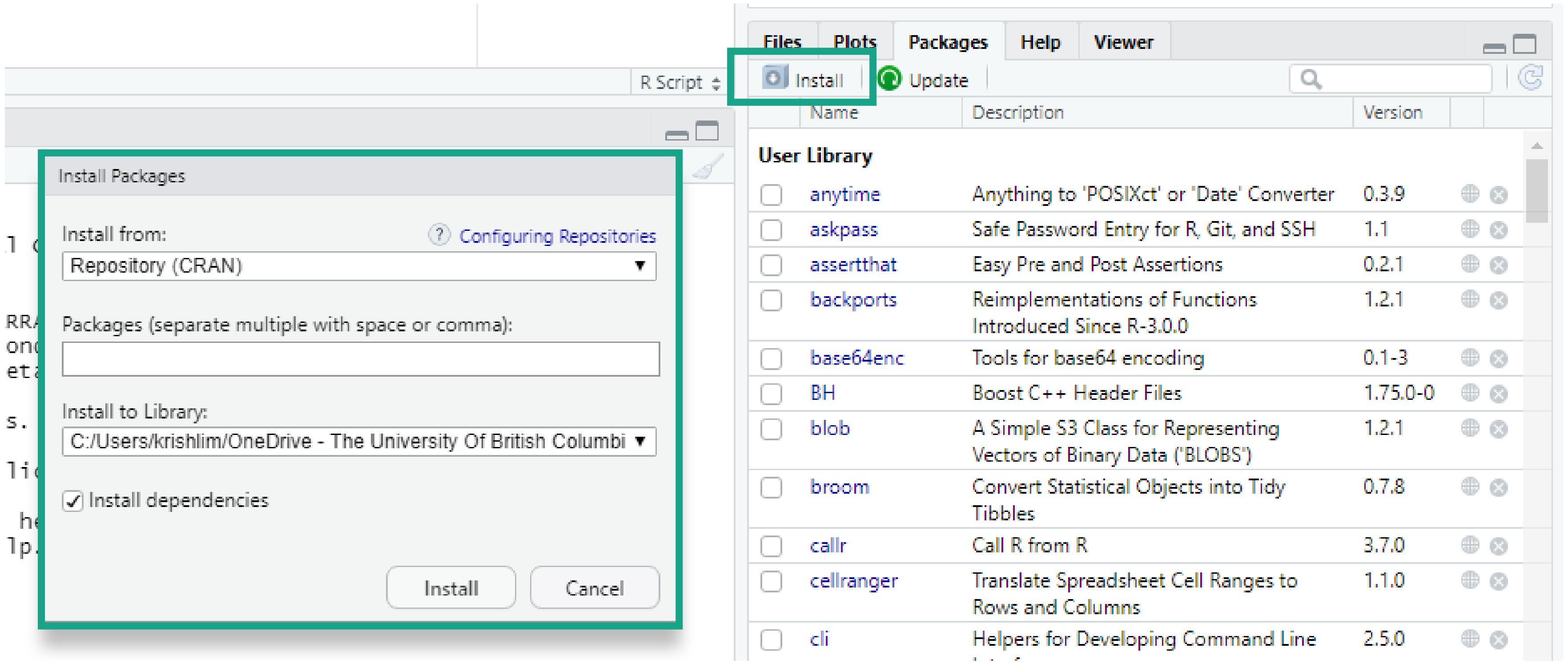
- As of June 2019, there are 14,000 packages on the Comprehensive R Archive Network (CRAN)

- Because there are so many packages out there, there are many ways to perform the same task.
  - In MFRE, different course/lab instructors may have different ways of coding or prefer different packages/functions.

UBC
MFRE

Packages

- You install packages with the command ***install.packages("package_name")***
  - Don't forget the quotation marks!



Install Packages

Install from: Repository (CRAN)

Configuring Repositories

Packages (separate multiple with space or comma):

Install to Library: C:/Users/krishlim/OneDrive - The University Of British Columbi

✓ Install dependencies

Install | Cancel

Name	Description	Version
User Library		
anytime	Anything to 'POSIXct' or 'Date' Converter	0.3.9
askpass	Safe Password Entry for R, Git, and SSH	1.1
assertthat	Easy Pre and Post Assertions	0.2.1
backports	Reimplementations of Functions Introduced Since R-3.0.0	1.2.1
base64enc	Tools for base64 encoding	0.1-3
BH	Boost C++ Header Files	1.75.0-0
blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.1
broom	Convert Statistical Objects into Tidy Tibbles	0.7.8
callr	Call R from R	3.7.0
cellranger	Translate Spreadsheet Cell Ranges to Rows and Columns	1.1.0
cli	Helpers for Developing Command Line	2.5.0

Packages

- To install multiple packages at once, you can use the code

```
install.packages(c('package1', 'package2',...))
```

- Try it now!
  - Install the packages **tidyverse** and **readxl**

UBC MFRE

Packages

- We always have to load libraries every time to start a new R session – key difference with Stata

- A few ways to load a library
  - Click the checkbox beside the package you want in the package tab on lower left panel in RStudio
  - Type in ***library(package_name)*** in the console
    - Quotation marks are now optional.
  - You may also see some people use **require(package_name)**

UBC
MFRE

A more efficient way

- Google it – “How to load a bunch of libraries in R”

- We can use the ***p_load()*** function in the ***pacman*** package to check if a certain package is already installed, and if not, install and then load it.
  - The **p_load()** function can be used to load multiple packages at the same time

UBC
MFRE

https://statsandr.com/blog/an-efficient-way-to-install-and-load-r-packages/

A more efficient way

install.packages(“pacman”) - ***do this only once***

pacman::p_load(tidyverse, readxl, googlesheets4, readstata13,here, magrittr, cansim, stats, broom, modelsummary, flextable)

- The pacman::p_load() syntax means that R loads the p_load() function only from the {pacman} package.
    - You can use this syntax even without doing library(pacman).
    - The packages inside the parentheses of p_load() change depending on your needs.

UBC MFRE

Interacting with R

- You can get output from R by typing math expressions in the console

```
1+2

2+4*1^3

100 %/% 60        # How many whole hours in 100 min

100 %% 60         # How many minutes are left over?
```

UBC MFRE

Interacting with R

- You can get output from R by typing logic statements in the console

```
1 > 2 & 1 > 0.5          # The “&” stands for “and”

1 > 2 | 1 > 0.5          # The “|” stands for “or”

3 + 4 == 4 + 3           # The “==“ stands for “equal to”

3 + 4 != 4 + 3           #The “!” is negation
```

Creating objects in R

- To do useful and interesting things, we assign *values* to *objects*

- In R, we use <- or = as assignment operators

- Which assignment operator to use?[1]
  - You will find most R users use <- because = have a particular use within functions.
  - You can read more about this issue here and here
  - **Bottom line:** Use whichever you prefer. Just be consistent.

UBC MFRE

[1] https://raw.githack.com/uo-ec607/lectures/master/04-rlang/04-rlang.html#28

Creating objects in R

- You can name your objects anything you want, but they cannot start with a number and cannot use R reserved words (i.e. if, else, for)

- R is also case sensitive (age is different from Age)
    - snake_case
    - PascalCase
    - camelCase

UBC
MFRE

https://colinfay.me/r-assignment/

Creating objects in R

- When we assign values to an object, R does not automatically print the object.

```
x <- 3 # creates an object x; does not print anything
(x <- 3) # creates an object x; prints the value of x
x # prints the value of x

2 * x   #x is now stored in the memory; we can use it
```

UBC
MFRE

Creating objects in R

- We can also overwrite the value of an existing object

```
x <- 10  # we can also change the value of an object

x * 3
```

UBC
MFRE

Commenting your code

- It is very important to comment on your code

- In R, we use the **#** character
  - Anything after **#** up to the end of the line will be treated as a comment, and R will not evaluate it as code

  - To comment/uncomment a chunk of code, highlight the lines then press *ctrl* + *shift* + *c* on your keyboard

Recap

- We use R/RStudio interchangeably
- There are 4 panels in R/Rstudio.
- We can run codes directly in the console. Or we can type the codes in the upper left panel to save for future use.
- There are multiple ways to install packages in R – install.packages(), pacman::p_load(), or manually through the packages tab
- <- and = are both assignment operators. Can use either as long as you are consistent.
- # indicates the start of the comment

UBC
MFRE

UBC
MFRE

mfre.landfood.ubc.ca