Introduction to R

Lecture 1

January 6th, 2020

Dr. Kristin Eccles

kristin.eccles@utoronto.ca

@kristineccles

https://github.com/kristineccles/Introduction_to_R

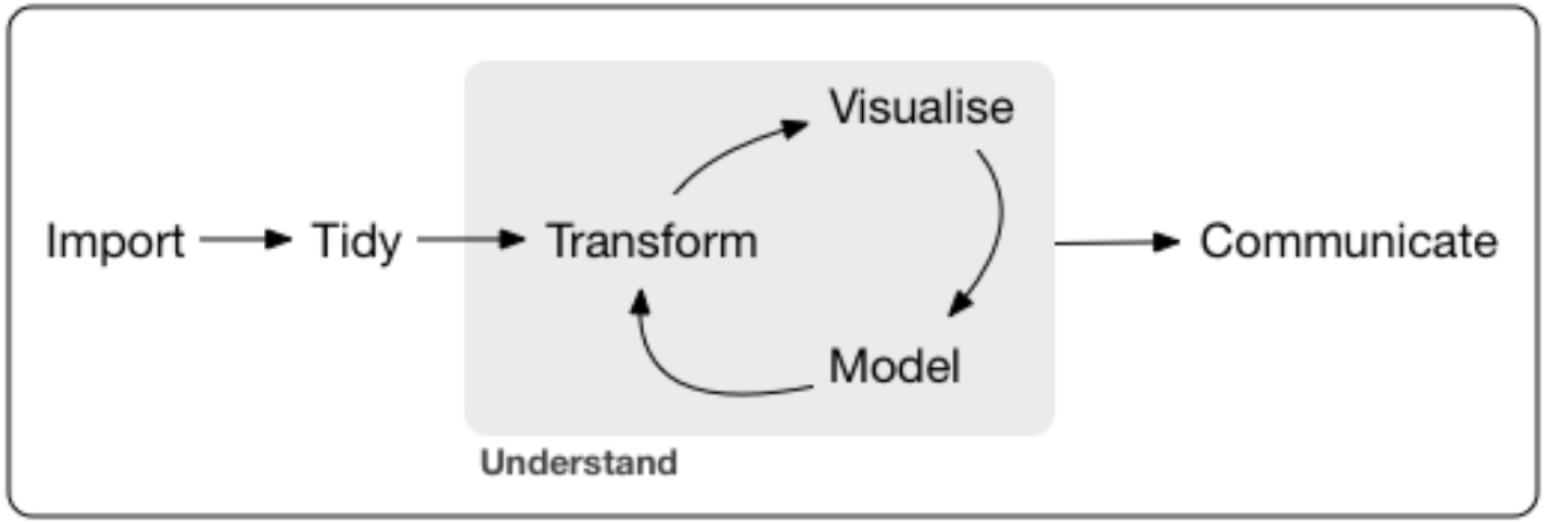
UNIVERSITY OF TORONTO MISSISSAUGA

Overview

- January 6th:
  - What can R do?
  - Why and when would you want to use R?
  - How to get help; Working with RStudio
  - Core Language
    - Data types, functions, operations, loading data, saving data
    - R Packages (installing, loading, using)
  - Exploratory Data Analysis
    - Graphs (base, ggplot2)
- January 13th:
  - Standard statistical functions: Descriptive statistics, correlations, linear regression
  - Overview other modelling possibilities (i.e. Generalised linear models, multilevel modelling, structural equation modelling, Bayesian analysis, bootstrapping, meta-analysis)
  - Introduction to mapping in R

What is R?

- "R is a free software environment for statistical computing and graphics" - http://www.r-project.org



- Why the name "R"?
  - First letter of two originators: Ross Ihaka and Robert Gentleman
  - Built on a earlier language called "S"
  - (S-Plus)

Why use R?

- R is free to use
- R is **open source**
  - "denoting software for which the original source code is made freely available and may be redistributed and modified."
- Runs on all operating systems (Windows, OSX, Linux)
- R is very versatile
  - huge library of user-contributed packages (over 6,000 on Comprehensive R Archive Network (CRAN))
- Facilitates reproducible research
- Popular in academia and industry
  - A lot of free online resources (stack overflow, r stats, etc.)

What is used in academia?

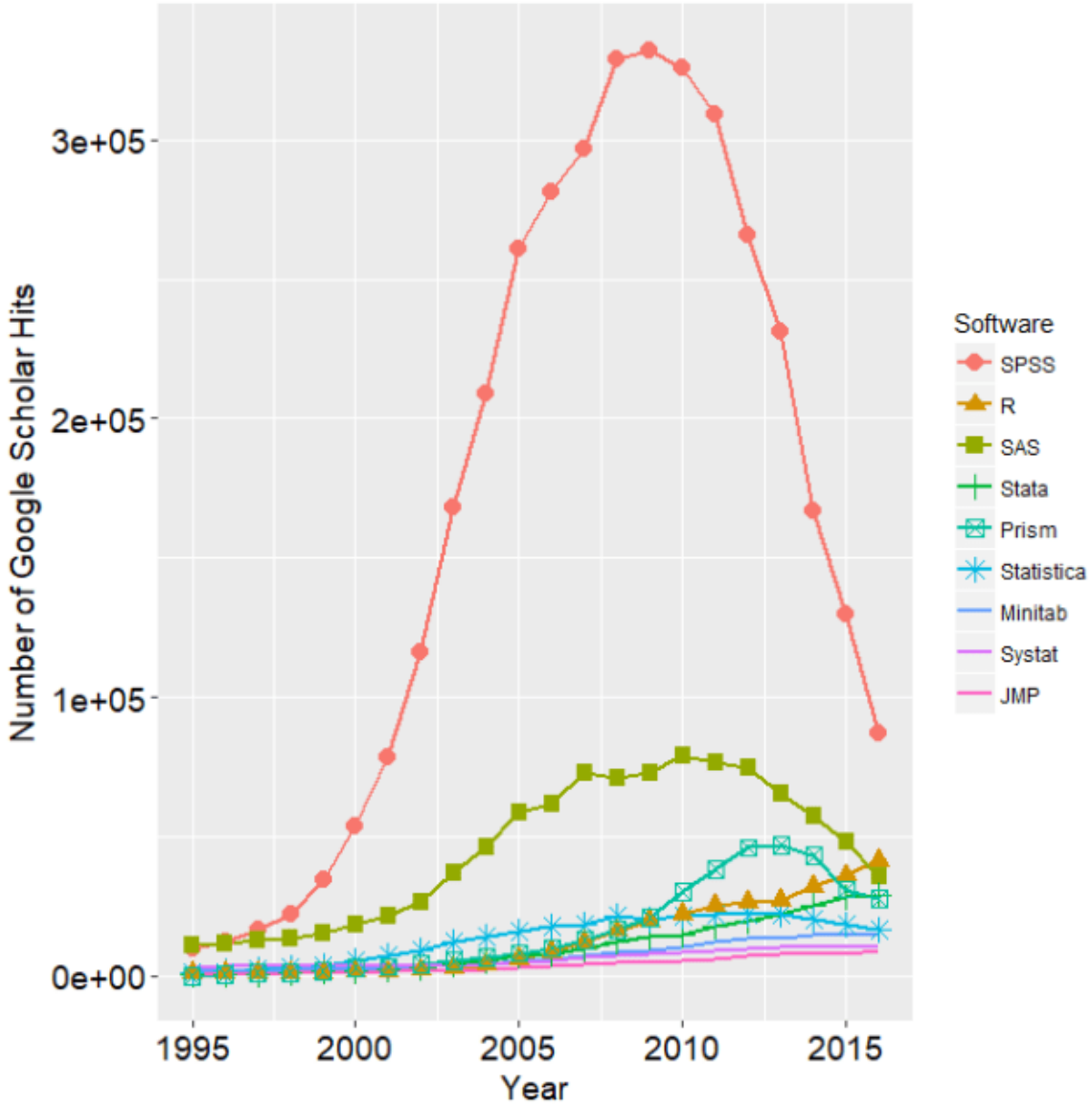


Figure 2d. The number of scholarly articles found in each year by Google Scholar. Only the top six “classic" statistics packages are shown Source: http://r4stats.com/articles/popularity.

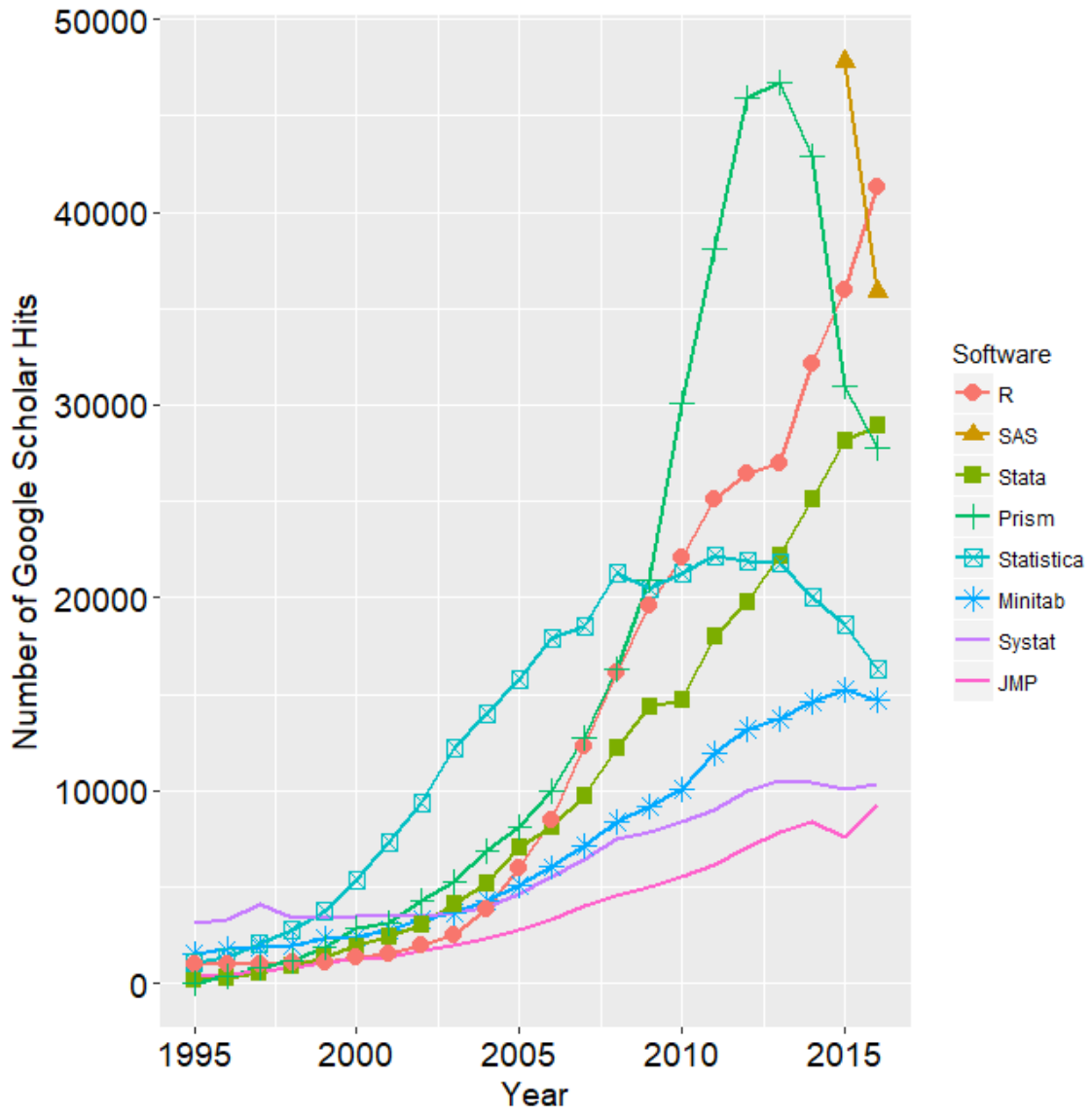
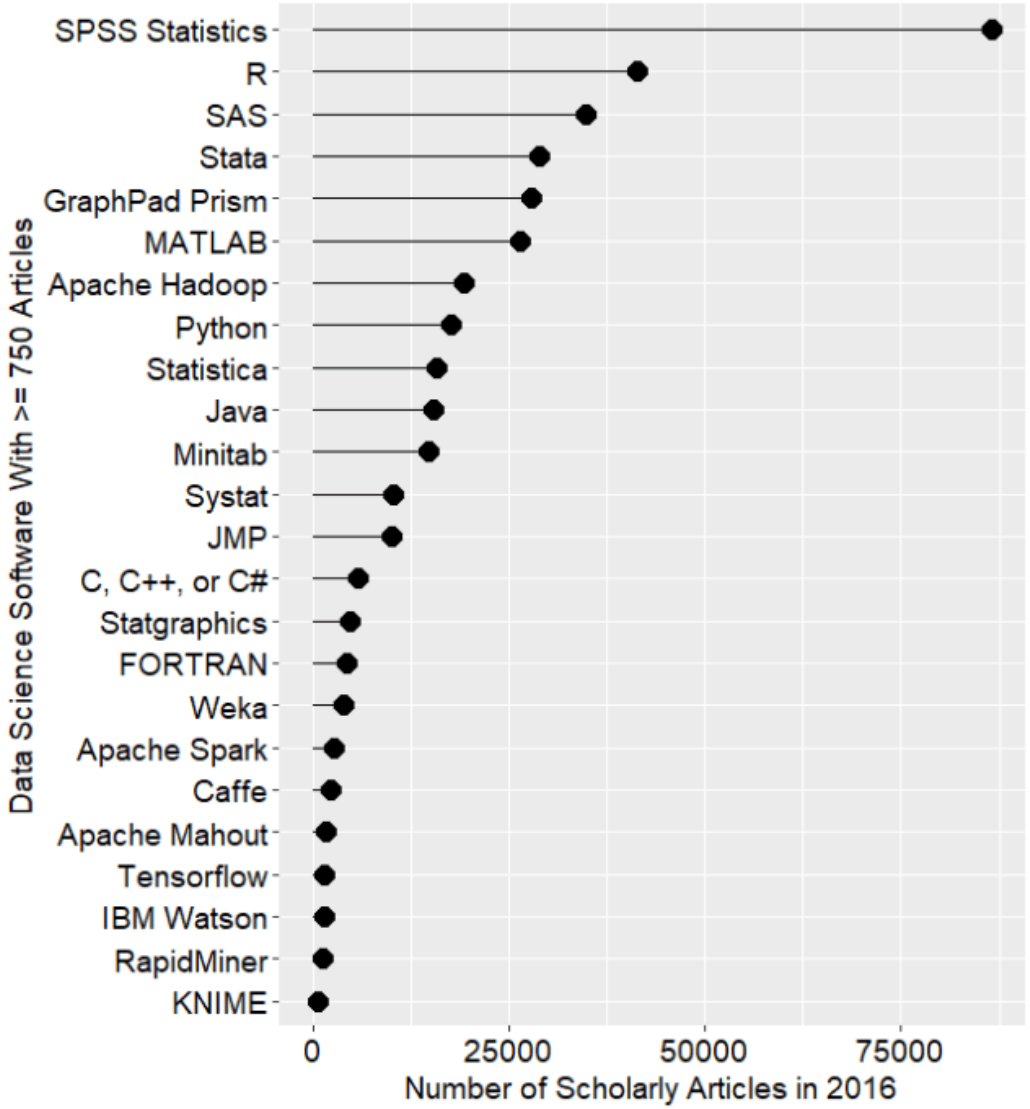
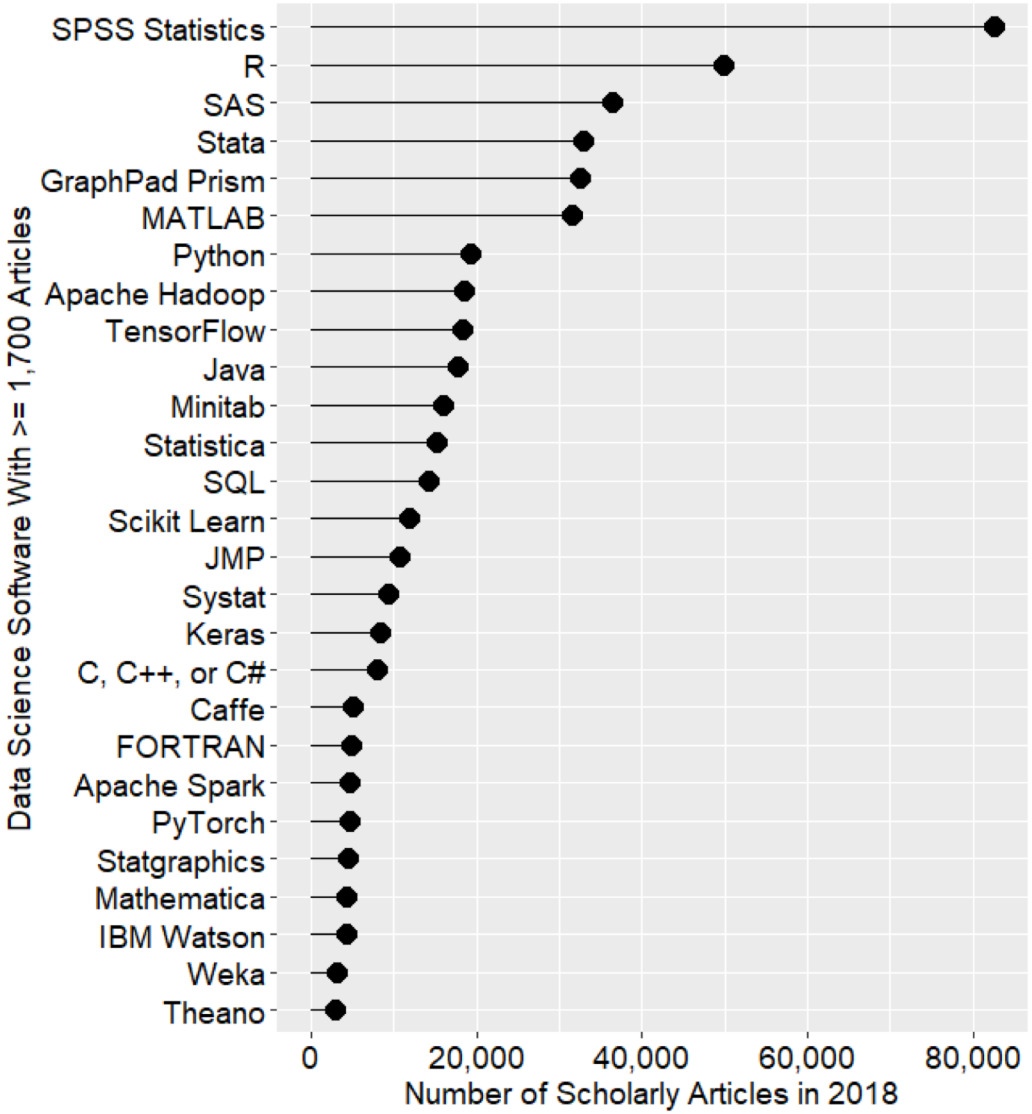


Figure 2e. The number of scholarly articles found in each year by Google Scholar for classic statistics packages after the curves for SPSS and SAS have been removed.Source: http://r4stats.com/articles/popularity.

What is Used in Academic Articles?

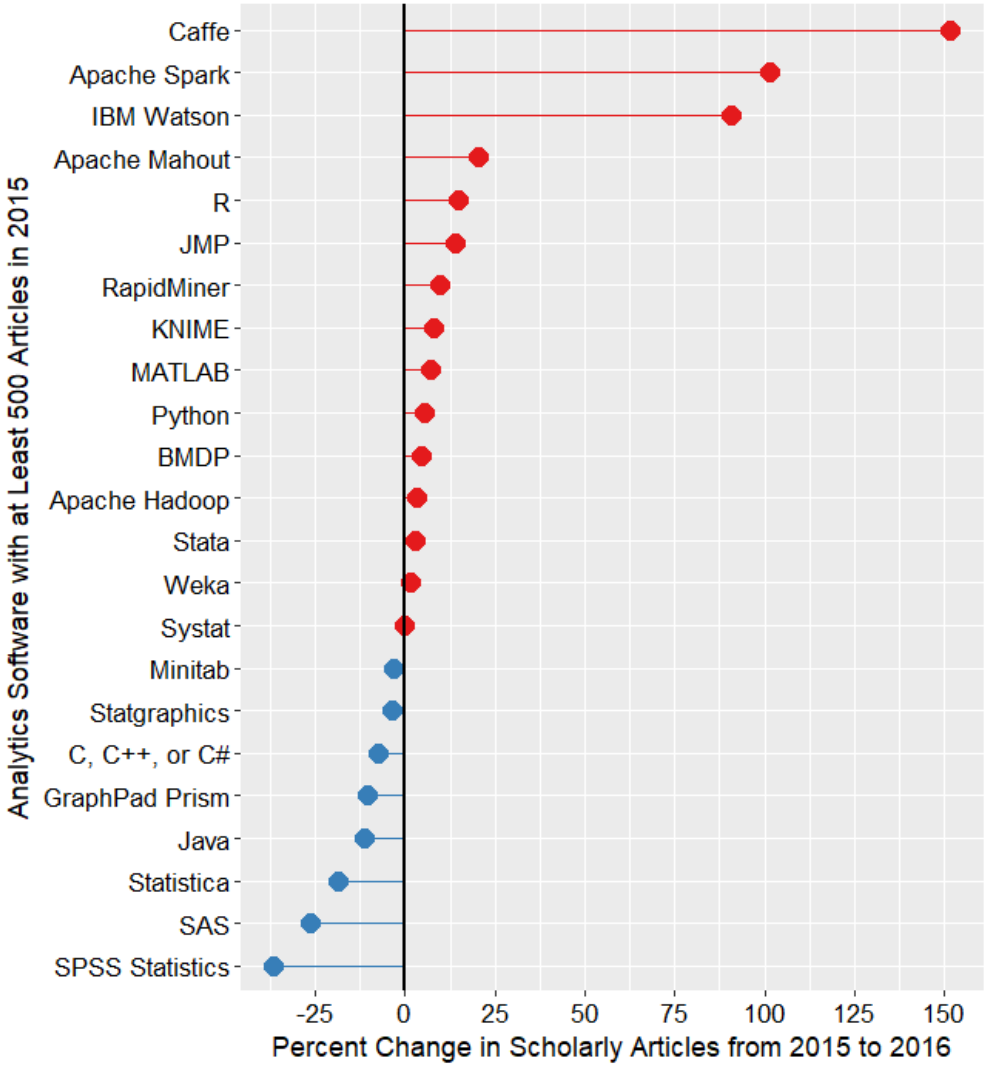


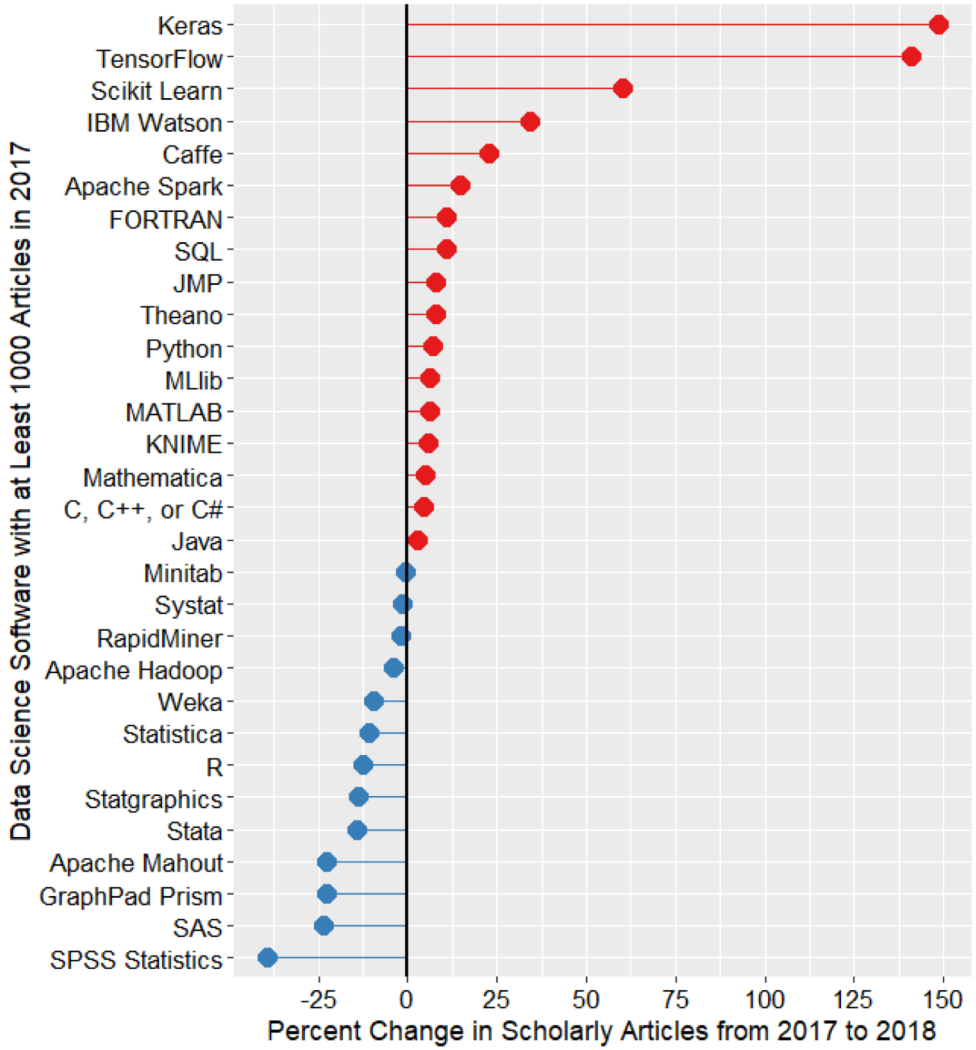


UNIVERSITY OF TORONTO MISSISSAUGA

Figure 2a. The number of scholarly articles found on Google Scholar, for data science software. Source: http://r4stats.com/2019/04/01/scholarly-datasci-popularity-2019/

Software with the most academic growth





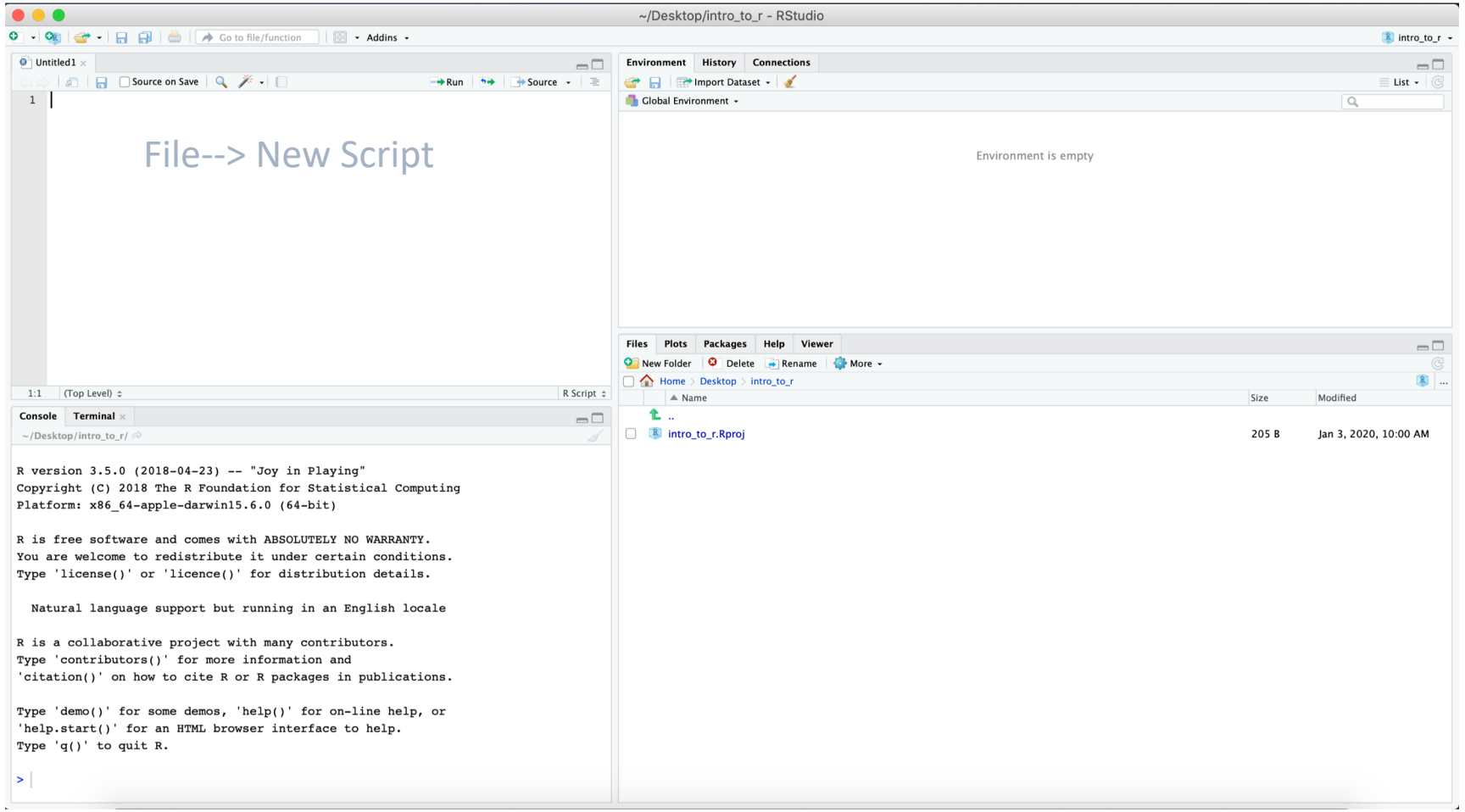
UNIVERSITY OF TORONTO MISSISSAUGA

Figure 2c. Change in the number of scholarly articles using each software in the most recent two complete years (2015 to 2016 and 2017-2018. Packages shown in red are “hot" and growing, while those shown in blue are “cooling down" or declining. Source: http://r4stats.com/articles/popularity

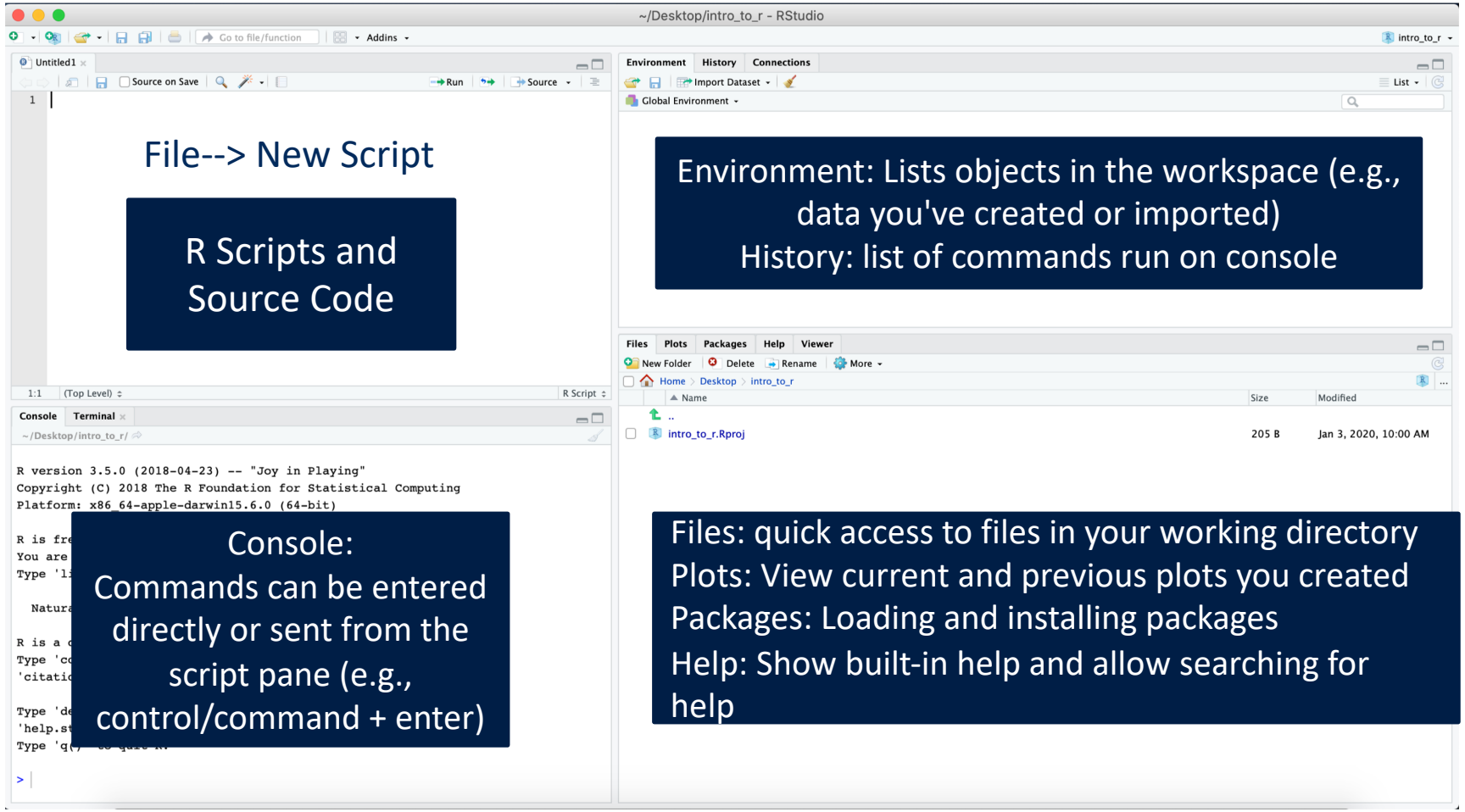
Challenges of using R

- R involves writing scripts
- It does not have a GUI like SPSS, SAS, Stata, etc.
  - But RStudio is a user friendly Integrated Developer Environment (IDE)
- R is more interactive
  - In SPSS and SAS you choose a command and get piles of output which you wade through
  - R is a conversation: You interactively request relevant output

A Guided Tour of RStudio



A Guided Tour of RStudio



RStudio Projects

- It is good practice to store all files related to a particular analysis project in a single directory on your computer
  - I.e. scripts, data files, configuration files, figures, exported tables, etc.
- Rstudio makes this easy to do
  - (Go to: `File → New Project → New Directory → New Project → Create Project`)
  - The directory name: want to call it
  - Create project of a subdirectory of: where on your computer you want it stored
- This generates a folder and a file with an "Rproj" extension (e.g., `projectname.Rproj`)
  - In the future, double click on this file to open the project
  - R studio will open the previous working environment

Overview of common file extensions

- R Specific file formats
  - .r : R script files
  - .rmd : RMarkdown files
  - .Rproj : RStudio project files
  - .rdata : Native format in r for saving R objects
- Other relevant formats
  - .md : Markdown file
  - .csv : comma separated value data file

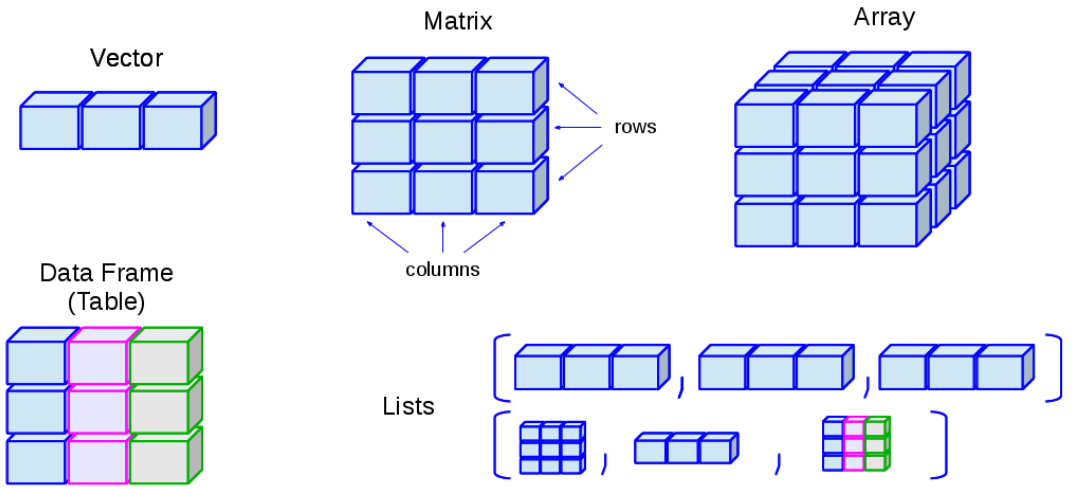
Objects and Classes

- R is an object oriented language
- Everything in R is an object: functions, symbols, and even R expressions.
- Objects may have attributes, such as name, dimension, and class R is an object-oriented language
  - Every object in R has a type
  - Every object in R is a member of a class
  - i.e. vectors, numeric vectors, dataframes, lists, and arrays
- All R code manipulates objects

For more reading: Chapter 7 and 10 of R in a Nutshell

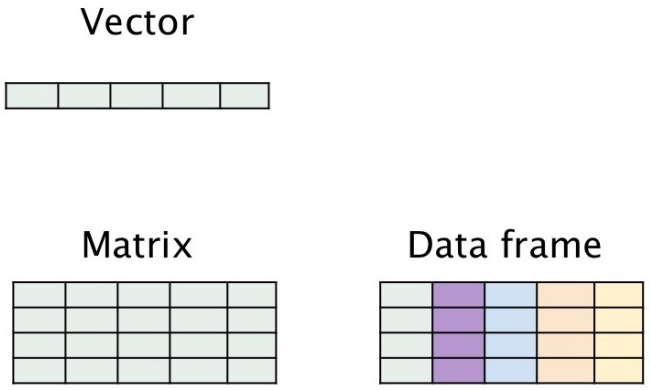
Data structure and types in R

- Data structures: vectors, matrices, arrays, data frames (similar to tables, and lists



- Data types: integer, numeric, logical, and factor

Variables	Example
integer	100
numeric	0.05
character	“hello”
logical	TRUE
factor	“Green”



Introducing R Commands

OPEN R

- R is an interpreted language
- Accessed through a command-line interpreter
  - This requires the user knowledge of commands and their parameters, and the syntax of the language
- Upon starting there is a “>” in the console. R is prompting you to type something, so this is called a prompt.
- The commands that you type into the console are called expressions

Vectors

- Vectors are one dimensional sequences of values
- In R, any number that you enter in the console is interpreted as a vector (numeric or character).
- A vector is an ordered collection of numbers.
  - The “[1]” means that the index of the first item displayed in the row is 1.

```
> # Basic Operations
> 2 + 2 # addition
[1] 4
> 3 - 5 # subtraction
[1] -2
> 3 * 2 # multiplication
[1] 6
> (2 + 2)^(3 / 3.5) # exponents and brackets
[1] 3.281341
```

```
> "Hello world."
[1] "Hello world."
> #This is called a character vector in R.
> c("Hello world", "Hello R interpreter")
[1] "Hello world"          "Hello R interpreter"
```

Functions

- Functions are the workhorses of R
- They take arguments as inputs and return objects as outputs.
- May modify objects in the environment or cause effects outside the R environment
  - I.e. plotting graphics, saving files, or sending data over the network.
- Functions provide information about vectors
- There are probably hundreds of thousands of functions in R.
- E.g.
  - length(x), mean(x), sd(x)

For more reading: Chapter 5 R in a Nutshell

Indexing

- The $ sign is used to reference a column by name
  - df$teams
- Reference a column
  - df[,2:3]
- Reference a row
  - df[2:3,]
- Reference rows and columns
  - df[1:2,1:2]
- R functions work better on columns than rows
  - Try calculating the average of a column
  - How would calculate the average of a row?

```
> df
  teams wins loses
1   PHI   92    70
2   NYM   89    73
3   FLA   94    77
4   ATL   72    90
5   WSN   59   102
```

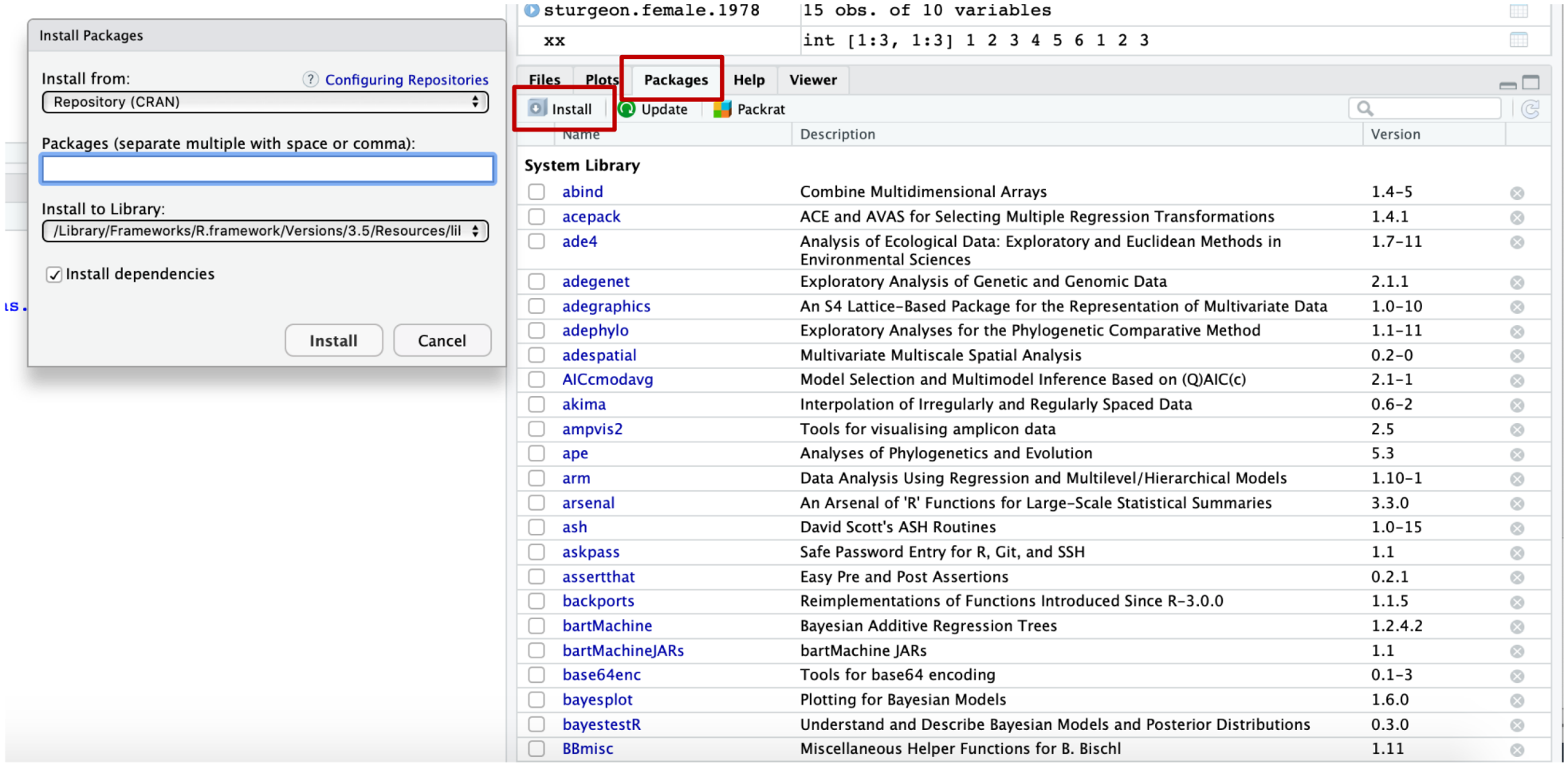
Loading Packages

- A package is a related set of functions, help files, and data files that have been bundled together.
- Typically, all the functions in the package are related
  - i.e. the stats package contains functions for doing statistical analysis
- You first need to make sure that it has been installed into a local library
  - R comes with a number of different packages

Table 4-1. Packages included with R

Package name	Loaded by default	Description
base	✓	Basic functions of the R language, including arithmetic, I/O, programming support
boot		Bootstrap resampling
class		Classification algorithms, including nearest neighbors, self-organizing maps, and learning vector quantization
cluster		Clustering algorithms
codetools		Tools for analyzing R code
compiler		Byte code compiler for R
datasets	✓	Some famous data sets
foreign		Tools for reading data from other formats, including Stata, SAS, and SPSS files
graphics	✓	Functions for base graphics
grDevices	✓	Device support for base and grid graphics, including system-specific functions
grid		Tools for building more sophisticated graphics than the base graphics
KernSmooth		Functions for kernel smoothing
lattice		An implementation of Trellis graphics for R: prettier graphics than the default graphics
MASS		Functions and data used in the book *Modern Applied Statistics with S* by Venables and Ripley; contains a lot of useful

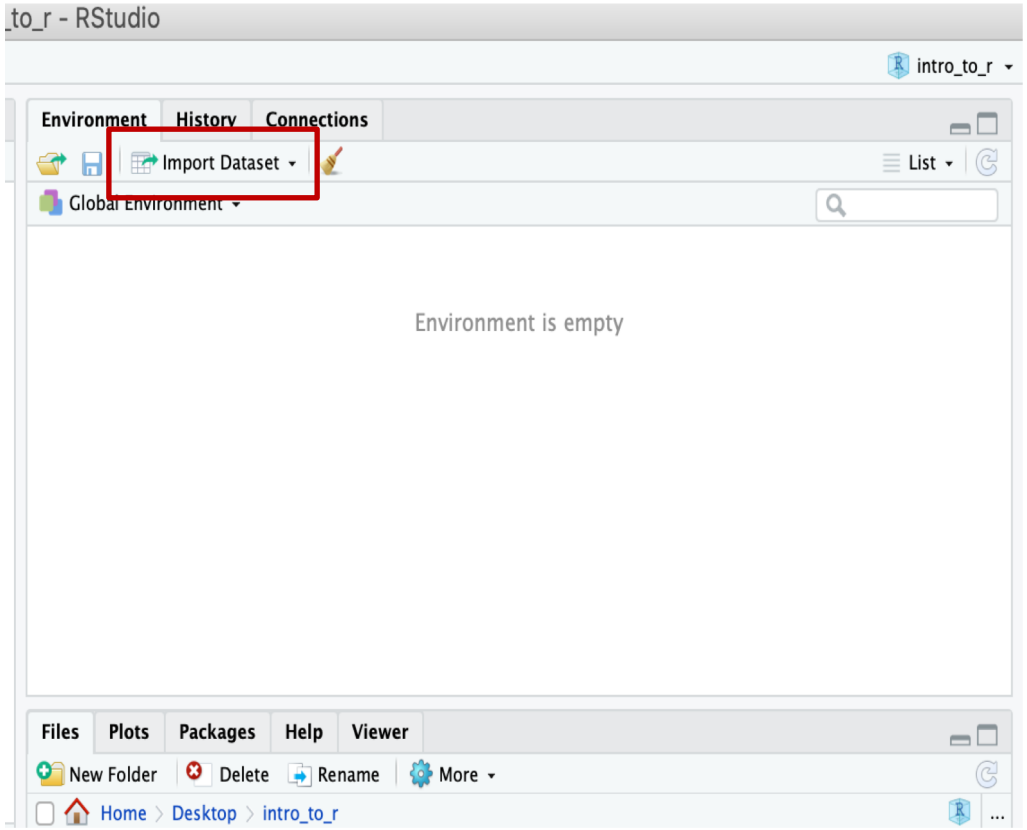
For more info see chapter 4 of R in a Nutshell



Or you can use the command install.packages(package)
Then you must call the library using library(package)

Read data into R

- You can import a variety of data file types, including from other statistics programs like SPSS, Stata, SAS, Minitab
- Common file formats like .xlsx and .csv
- The easiest way to import data is using the Import Dataset button in the Environment window.
- Better to use the command read.csv()



Useful Commands/ Tips

- To bring up help file, in the command line type:
  - ?commandname (searches only installed packages) OR help(commandname)
  - ??commandname (searches whole CRAN repository)
  - i.e. ?ggplot or help(ggplot)
- R is always case sensitive
- Always use a script and work from the editor
- Save your own annotated copy of the script.
- The # symbol means that the line will not be executed in R (useful to annotate scripts)