

# Introduction to R

## Lecture 1

September 14<sup>th</sup>, 2020

Dr. Kristin Eccles

kristin.eccles@utoronto.ca

 @kristineccles

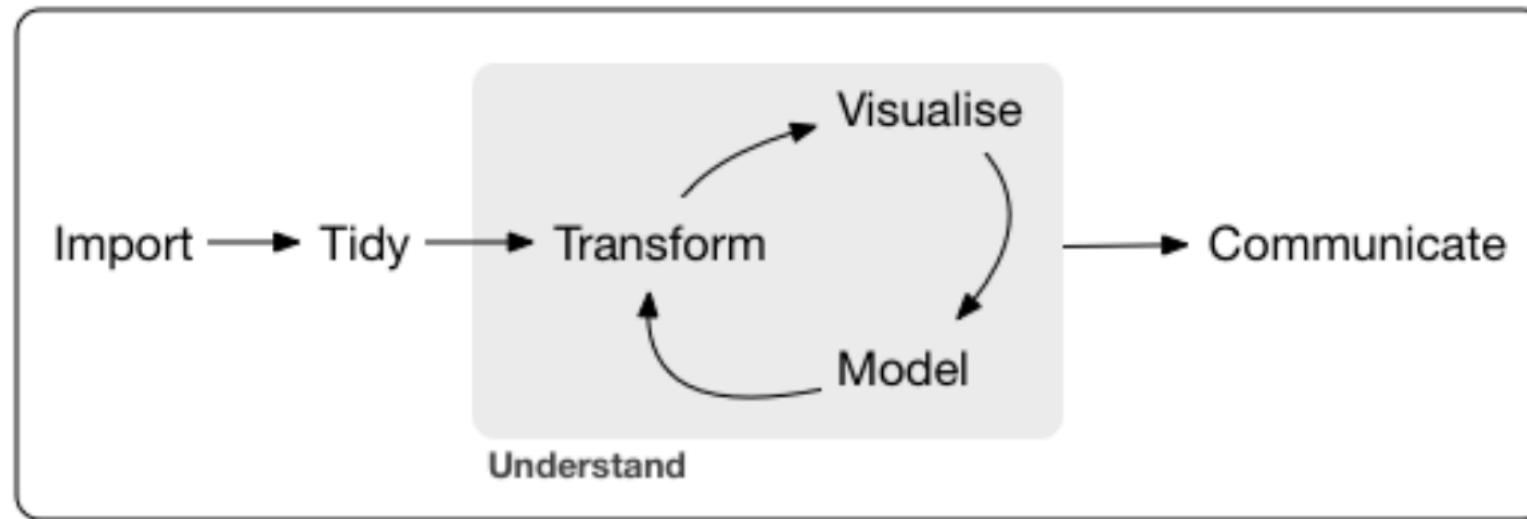
[https://github.com/kristineccles/Introduction\\_to\\_R](https://github.com/kristineccles/Introduction_to_R)

# Overview

- September 14<sup>th</sup>:
  - 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)
- September 21<sup>st</sup>:
  - 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?

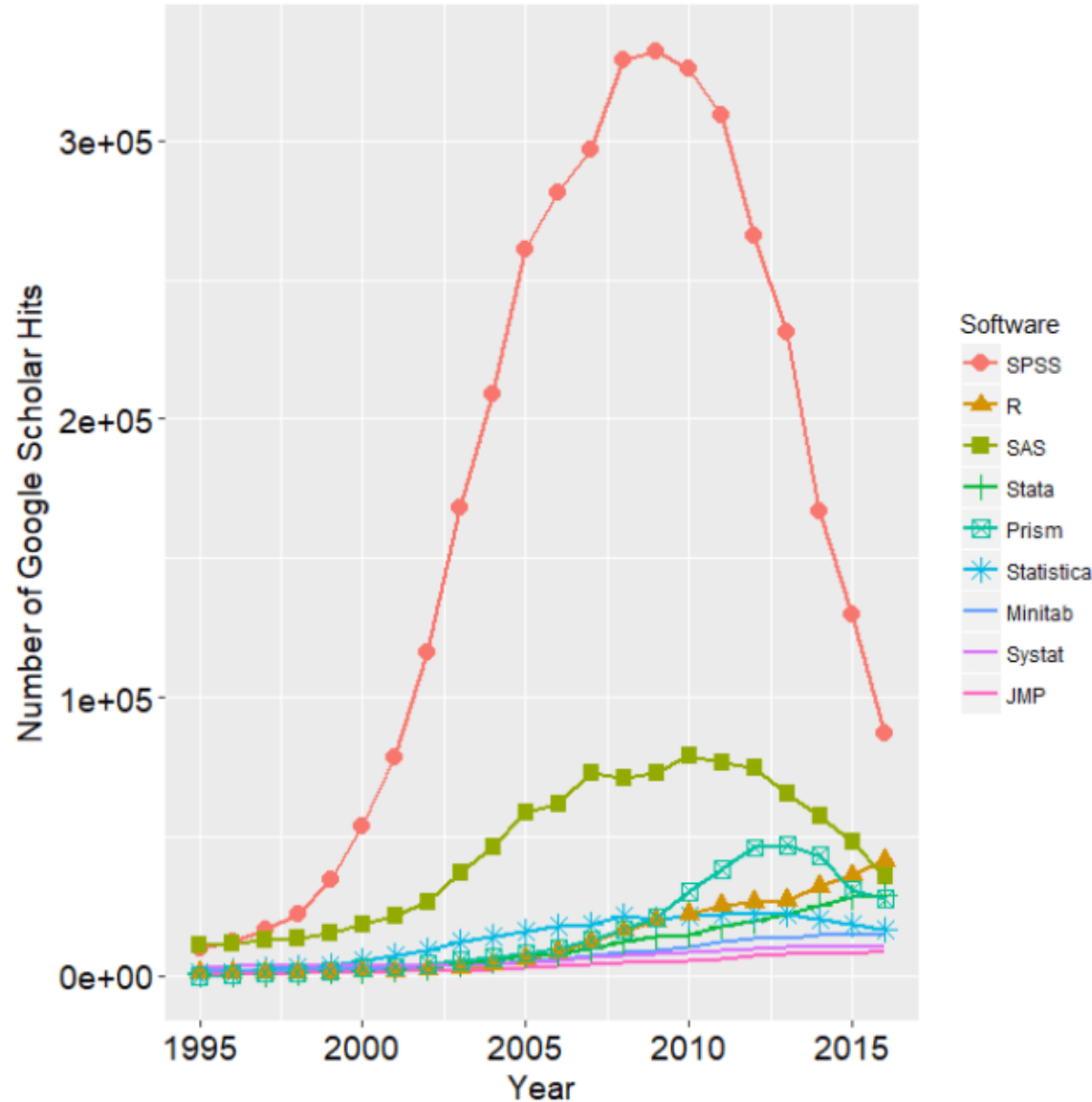


Figure 2d. The number of scholarly articles found in each year by Google Scholar. Only the top six “classic” statistics packages are shown Source: <http://r4stats.com/articles/popularity>.

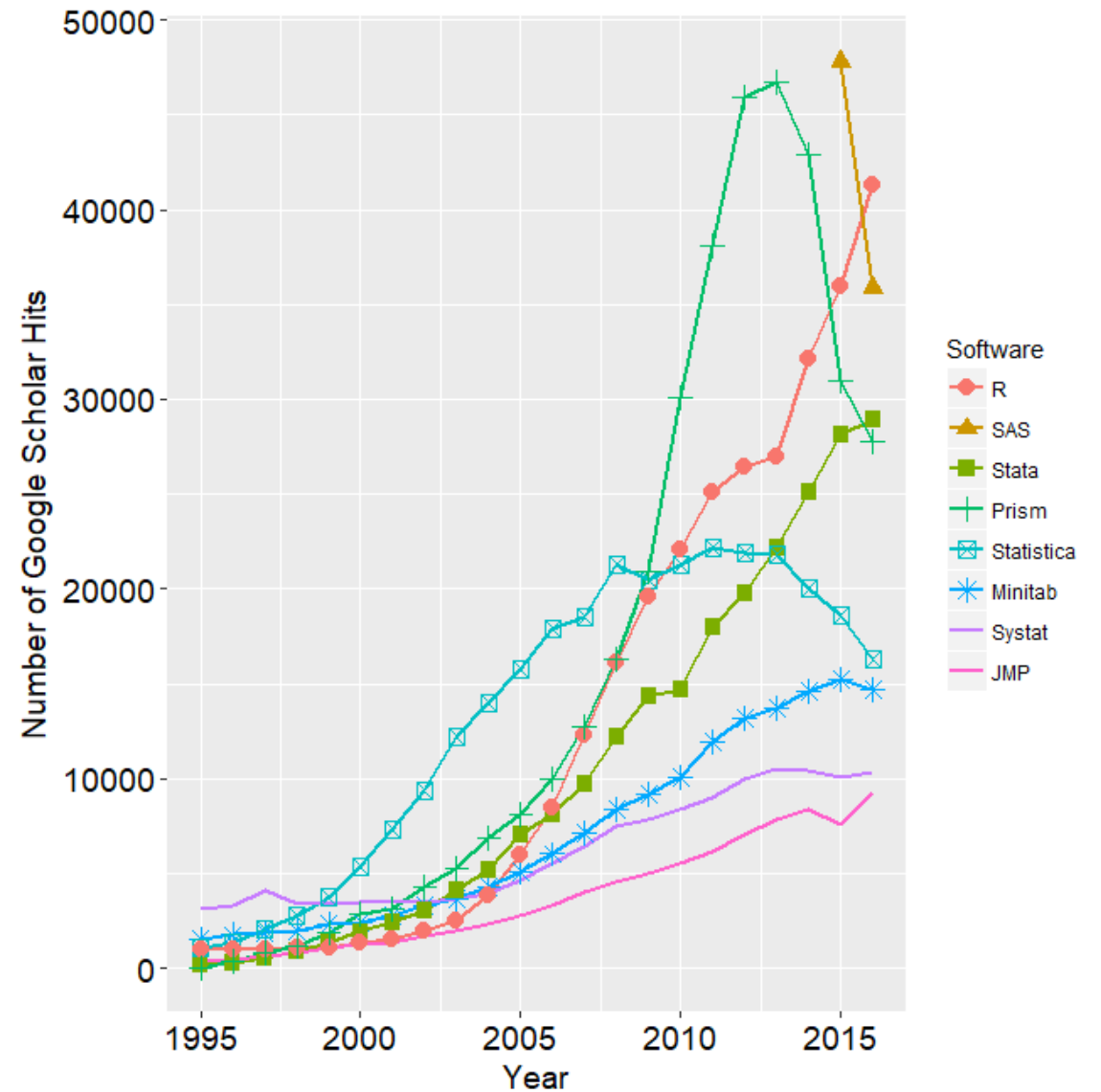
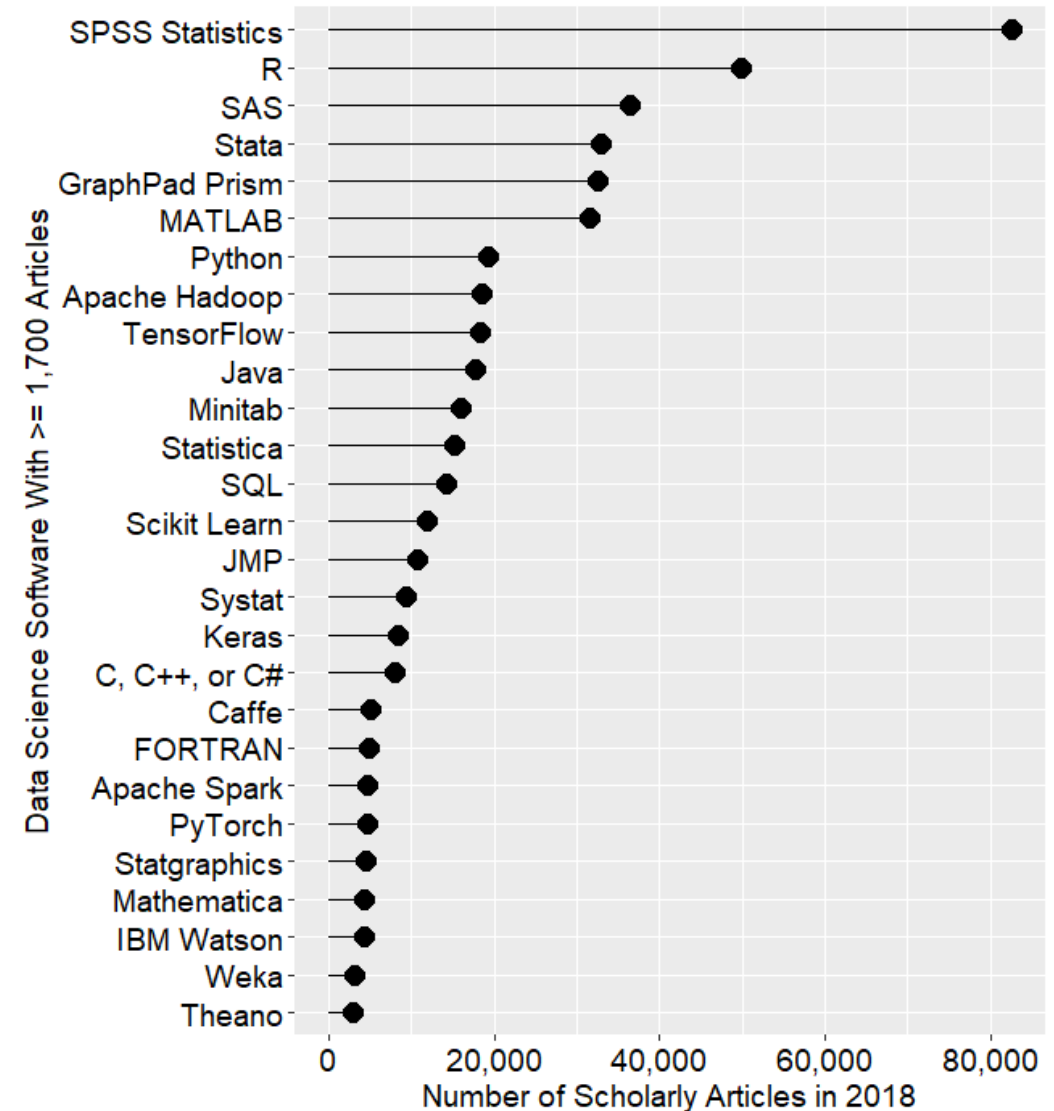
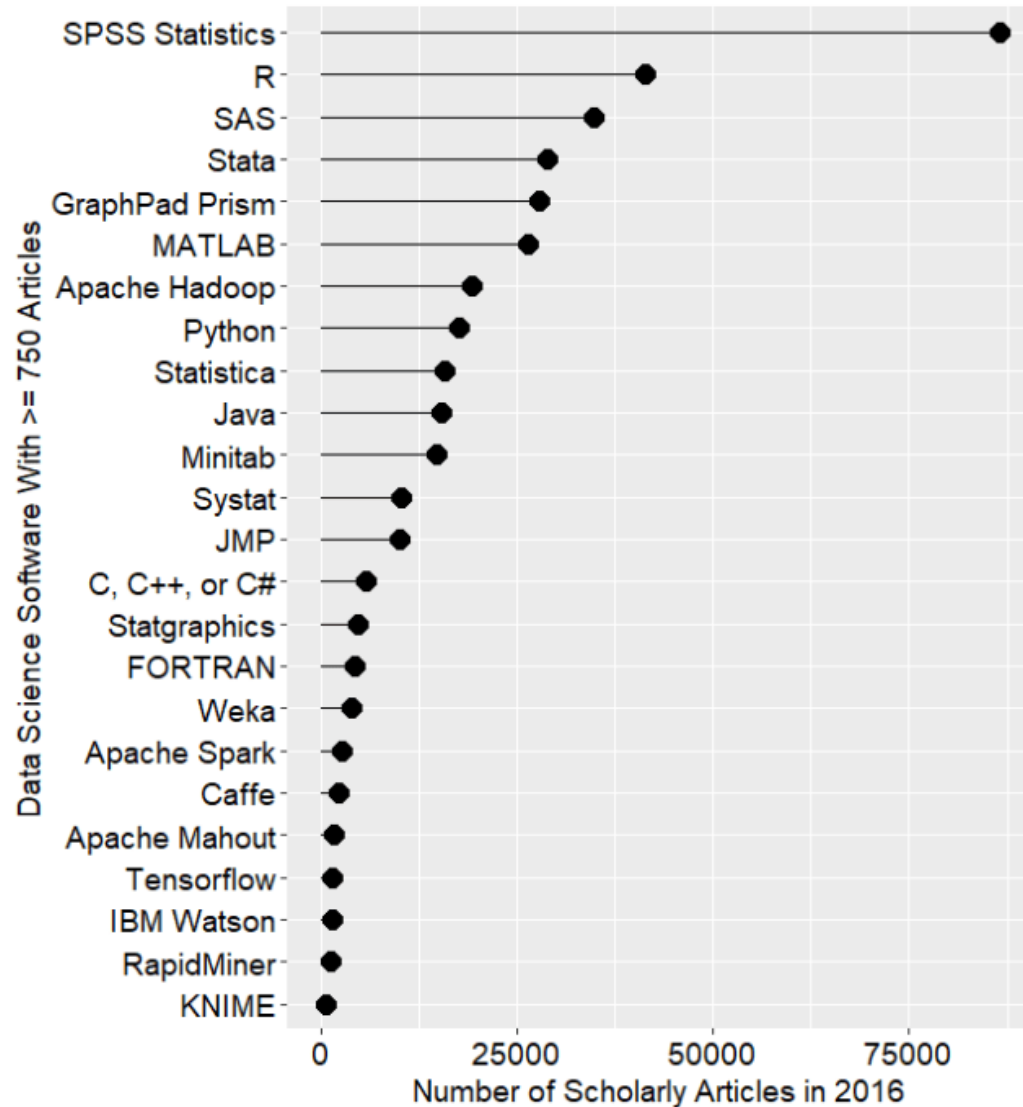


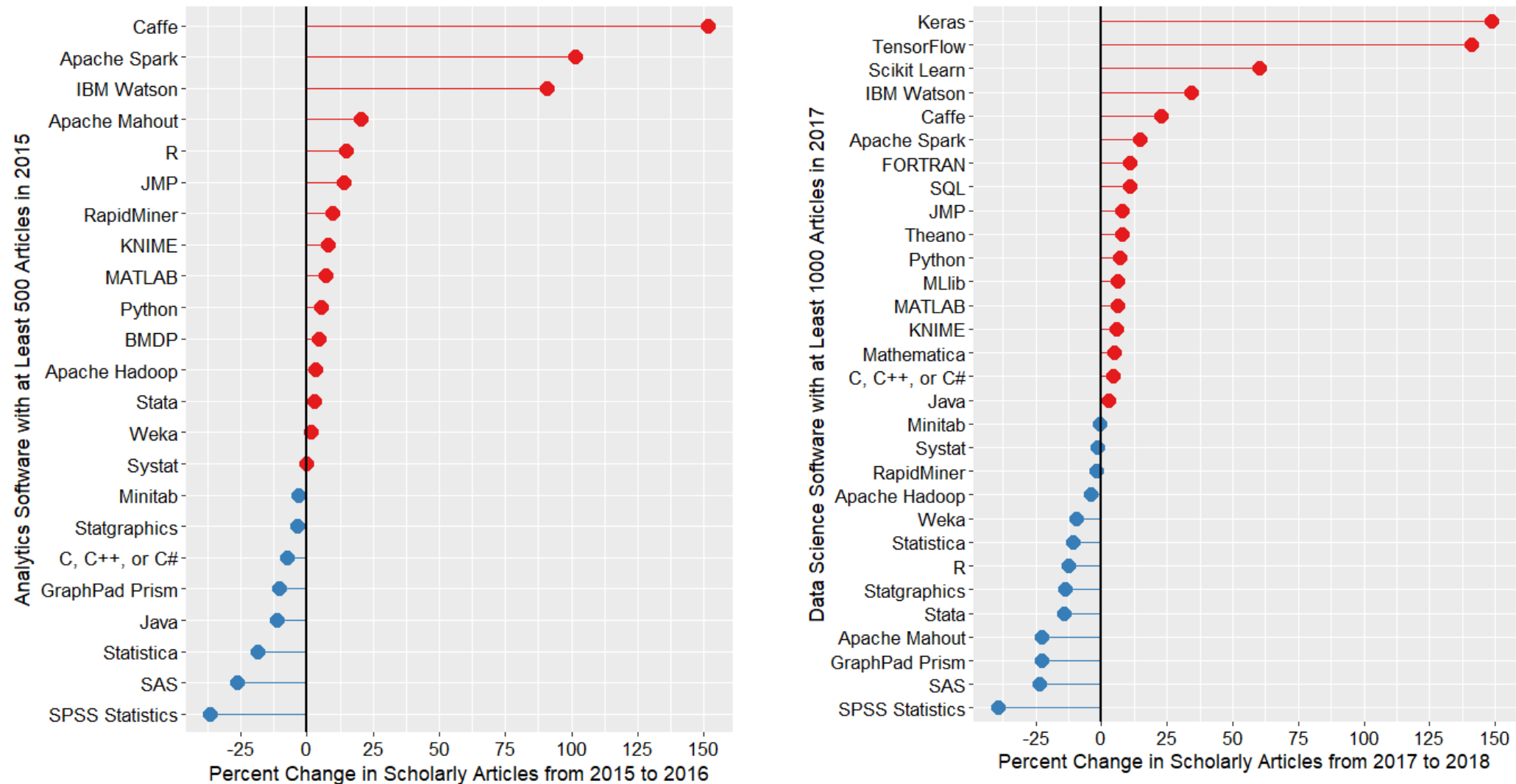
Figure 2e. The number of scholarly articles found in each year by Google Scholar for classic statistics packages after the curves for SPSS and SAS have been removed. Source: <http://r4stats.com/articles/popularity>.



# What is Used in Academic Articles?



# Software with the most academic growth

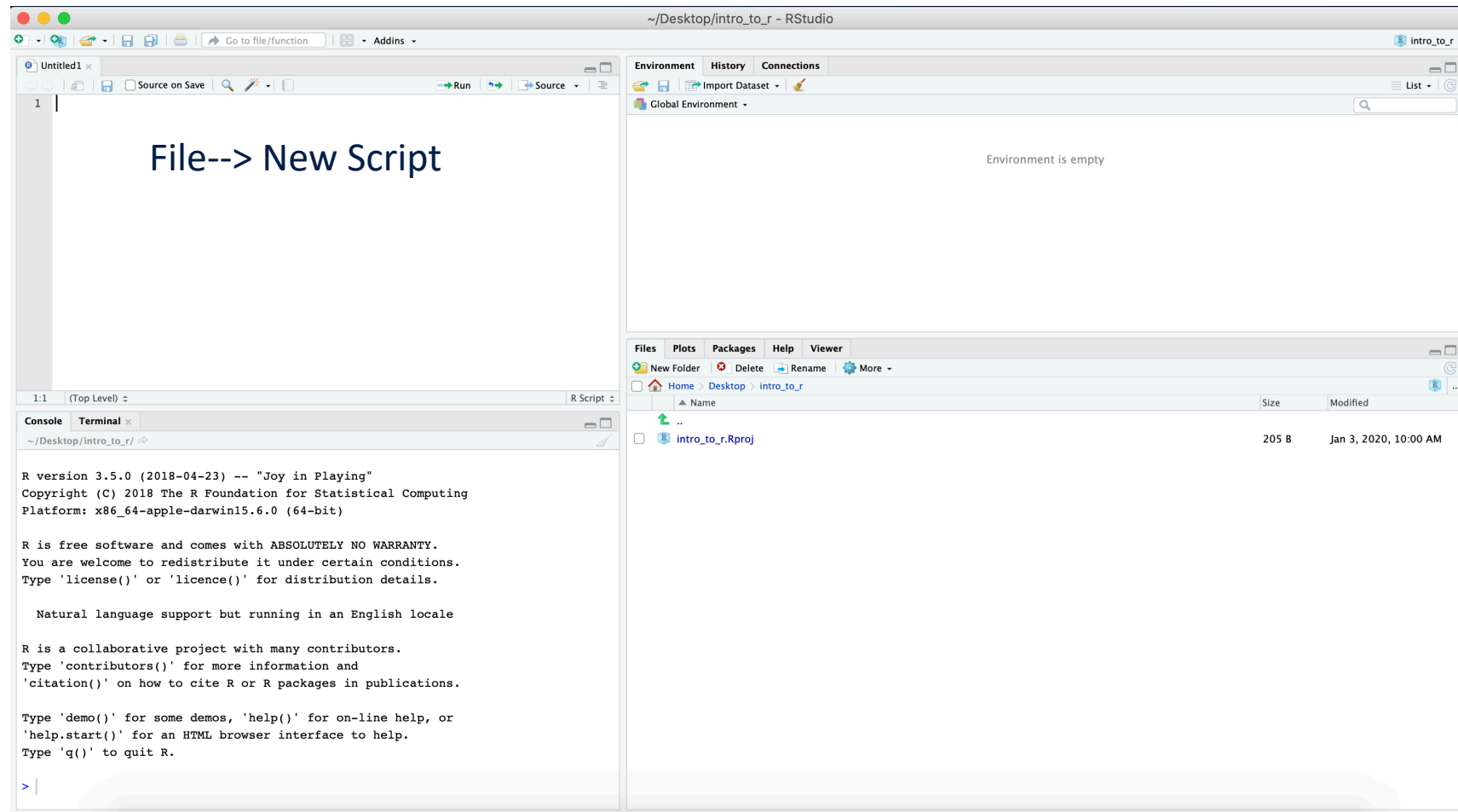


# Challenges of using R

- 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

The screenshot shows the RStudio desktop environment with the following components and annotations:

- Source Editor (Top Left):** Contains a text box with the instruction "File--> New Script" and a dark blue box labeled "R Scripts and Source Code".
- Environment and History (Top Right):** A dark blue box explains: "Environment: Lists objects in the workspace (e.g., data you've created or imported)" and "History: list of commands run on console".
- Console (Bottom Left):** A dark blue box explains: "Console: Commands can be entered directly or sent from the script pane (e.g., control/command + enter)". The console window shows the R startup message: "R version 3.5.0 (2018-04-23) -- 'Joy in Playing' Copyright (C) 2018 The R Foundation for Statistical Computing Platform: x86\_64-apple-darwin15.6.0 (64-bit)".
- Files Panel (Bottom Right):** A dark blue box explains: "Files: quick access to files in your working directory". The panel shows a file named "intro\_to\_r.Rproj" with a size of 205 B, modified on Jan 3, 2020, at 10:00 AM.
- Other Panels:** The "Plots" panel is visible but empty. The "Packages" panel shows a list of installed packages.

# 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
  - (Goto: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

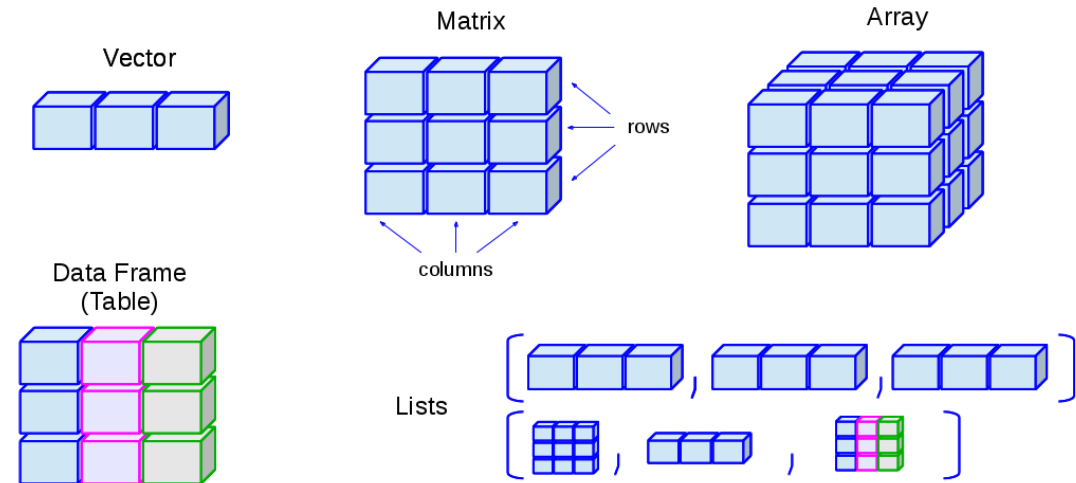
- R Specific file formats
  - .r : R script files
  - .rmd : RMarkdown files
  - .Rproj : RStudio project files
  - .rdata : Native format in r for saving R objects
- Other relevant formats
  - .md : Markdown file
  - .csv : comma separated value data file

# 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

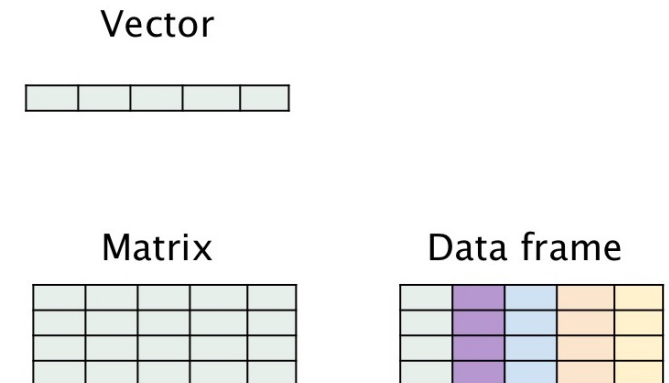
# Data structure and types in R

- 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)`

# 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 <i>Modern Applied Statistics with S</i> by Venables and Ripley; contains a lot of useful

For more info see chapter 4 of R in a Nutshell

The screenshot shows the RStudio interface. On the left, the 'Install Packages' dialog box is open. It has a tab 'Configuring Repositories' with a question mark icon. The 'Install from:' dropdown is set to 'Repository (CRAN)'. Below it, a text box for 'Packages (separate multiple with space or comma):' is empty. The 'Install to Library:' dropdown is set to '/Library/Frameworks/R.framework/Versions/3.5/Resources/lib'. The 'Install dependencies' checkbox is checked. 'Install' and 'Cancel' buttons are at the bottom.

On the right, the 'Packages' tab in the Environment pane is active. It shows a table of installed and available packages. The 'Install' button in the toolbar is highlighted with a red box. The 'Update' button is also highlighted with a red box. The 'Packrat' button is visible. The table lists packages with their names, descriptions, and versions.

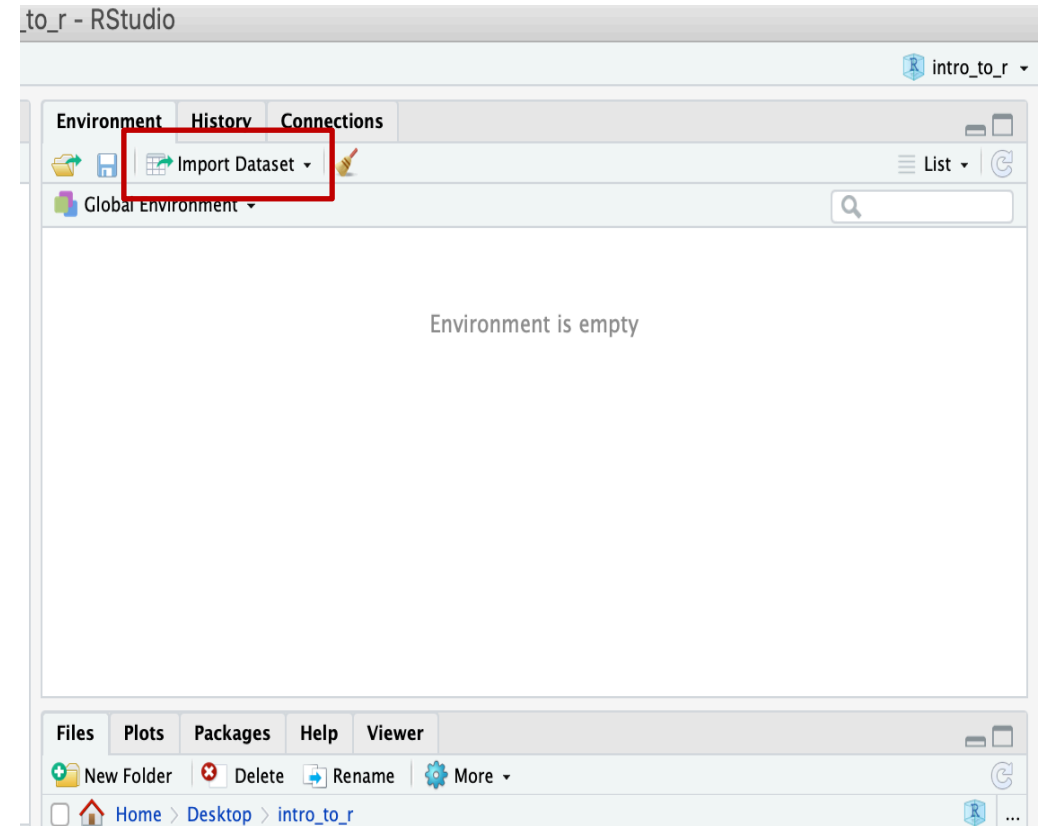
Name	Description	Version
<b>System Library</b>		
<input type="checkbox"/> abind	Combine Multidimensional Arrays	1.4-5
<input type="checkbox"/> acepack	ACE and AVAS for Selecting Multiple Regression Transformations	1.4.1
<input type="checkbox"/> ade4	Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences	1.7-11
<input type="checkbox"/> adegenet	Exploratory Analysis of Genetic and Genomic Data	2.1.1
<input type="checkbox"/> adegenet	An S4 Lattice-Based Package for the Representation of Multivariate Data	1.0-10
<input type="checkbox"/> adephylo	Exploratory Analyses for the Phylogenetic Comparative Method	1.1-11
<input type="checkbox"/> adespatial	Multivariate Multiscale Spatial Analysis	0.2-0
<input type="checkbox"/> AICcmodavg	Model Selection and Multimodel Inference Based on (Q)AIC(c)	2.1-1
<input type="checkbox"/> akima	Interpolation of Irregularly and Regularly Spaced Data	0.6-2
<input type="checkbox"/> ampvis2	Tools for visualising amplicon data	2.5
<input type="checkbox"/> ape	Analyses of Phylogenetics and Evolution	5.3
<input type="checkbox"/> arm	Data Analysis Using Regression and Multilevel/Hierarchical Models	1.10-1
<input type="checkbox"/> arsenal	An Arsenal of 'R' Functions for Large-Scale Statistical Summaries	3.3.0
<input type="checkbox"/> ash	David Scott's ASH Routines	1.0-15
<input type="checkbox"/> askpass	Safe Password Entry for R, Git, and SSH	1.1
<input type="checkbox"/> assertthat	Easy Pre and Post Assertions	0.2.1
<input type="checkbox"/> backports	Reimplementations of Functions Introduced Since R-3.0.0	1.1.5
<input type="checkbox"/> bartMachine	Bayesian Additive Regression Trees	1.2.4.2
<input type="checkbox"/> bartMachineJARs	bartMachine JARs	1.1
<input type="checkbox"/> base64enc	Tools for base64 encoding	0.1-3
<input type="checkbox"/> bayesplot	Plotting for Bayesian Models	1.6.0
<input type="checkbox"/> bayestestR	Understand and Describe Bayesian Models and Posterior Distributions	0.3.0
<input type="checkbox"/> BBmisc	Miscellaneous Helper Functions for B. Bischl	1.11

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)