Intro to R and R Studio

October 1, 2019


Cortney Hanna-Benson, PhD


@cortHB


linkedin.com/in/corthb/


corthb.wordpress.com

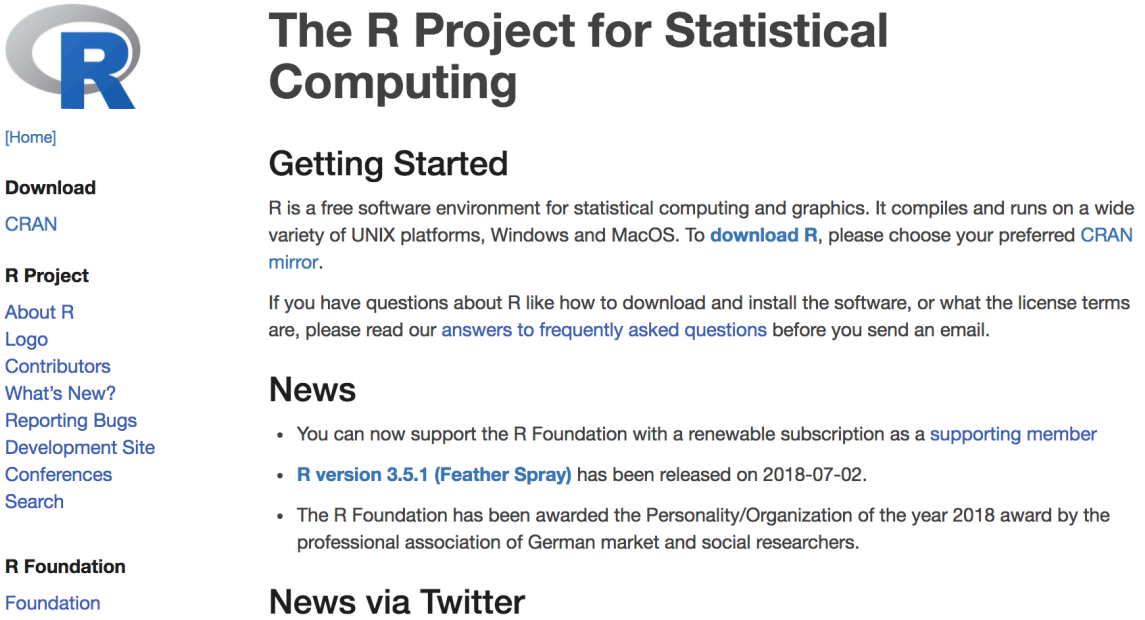

channa27@uwo.ca

Plan of attack

- What is R?
- Why bother?
- Downloading R and R Studio
- An exploration of the R Studio environment
- Basic functions
  - Introduction to the language
  - Descriptive Statistics
- T-test in R
  - With dataset import
- THIS IS AWESOME! Where do I go next?

What is R?

- R is a free, open-source **software** and **programming language**
  - Software environment for statistics, graphics, programming, calculator and MORE!
  - Language to explore, summarize, and model data, where functions are verbs and objects are nouns



[Home]

Download

CRAN

R Project

About R
Logo
Contributors
What's New?
Reporting Bugs
Development Site
Conferences
Search

R Foundation

Foundation

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.

If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

News

- You can now support the R Foundation with a renewable subscription as a supporting member
- **R version 3.5.1 (Feather Spray)** has been released on 2018-07-02.
- The R Foundation has been awarded the Personality/Organization of the year 2018 award by the professional association of German market and social researchers.

News via Twitter

Source: Roecker, S. et al. (2018)

Why learn R?

- There are a few reasons why you may choose to use R:

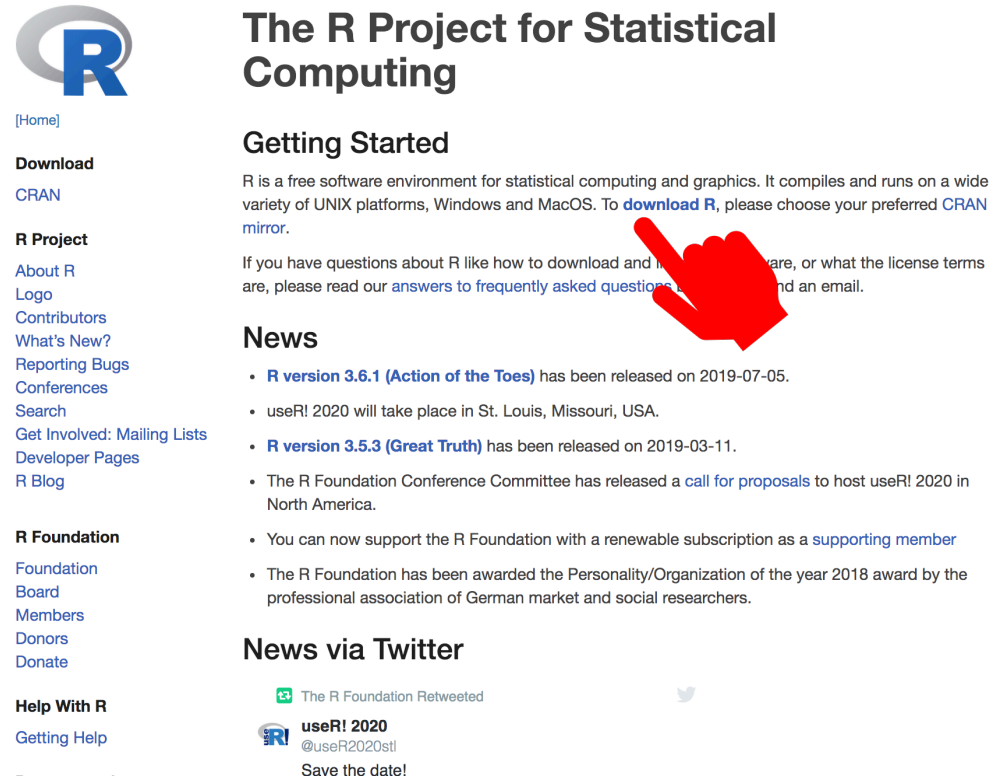
Reproducible research	Extensive user community	Learning Resources	Free!
• The code and output is a single document • Reduce potential errors in re-running analyses • E.g. adding data	• There are both online (e.g. Stack Overflow) and face-to-face communities (e.g. R Ladies) that support R use	• Many R users are also invested in the development and promotion of R use • Free resources created by R enthusiasts (e.g. bookdown.org/)	

Source: Roecker, S. et al. (2018)

Sounds great!
How do I begin?

- Step 1:
  - Download R

https://www.r-project.org/



Sounds great! How do I begin?

- Step 1:
  - Download R

https://www.r-project.org/

CRAN Mirrors

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. Some statistics on the status of the mirrors can be found here: main page, windows release, windows old release.

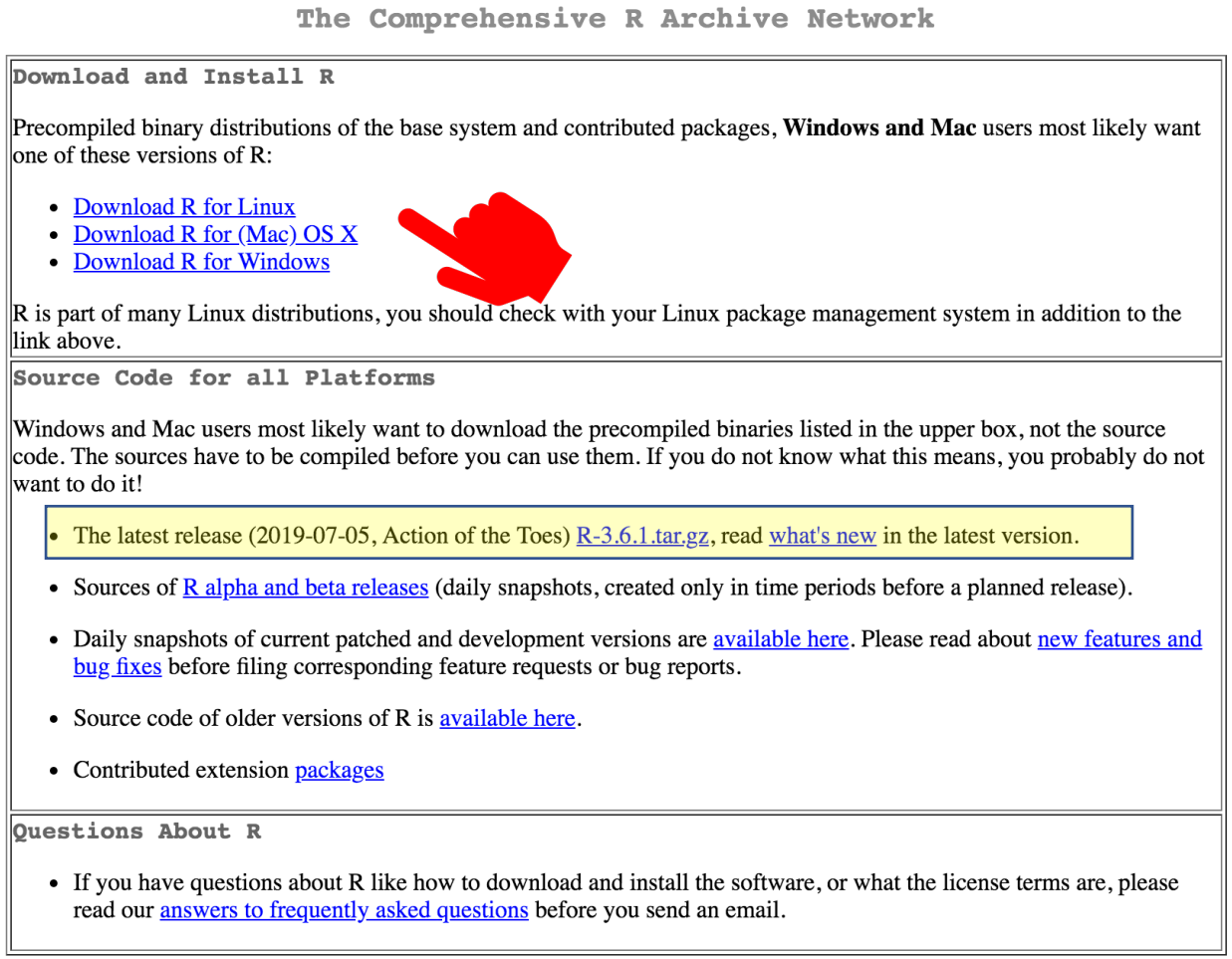
If you want to host a new mirror at your institution, please have a look at the CRAN Mirror HOWTO.

0-Cloud	
https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
http://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Algeria	
https://cran.usthb.dz/	University of Science and Technology Houari Boumediene
http://cran.usthb.dz/	University of Science and Technology Houari Boumediene
Argentina	
http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	
https://cran.csiro.au/	CSIRO
http://cran.csiro.au/	CSIRO
https://mirror.aarnet.edu.au/pub/CRAN/	AARNET
https://cran.ms.unimelb.edu.au/	School of Mathematics and Statistics, University of Melbourne
https://cran.curtin.edu.au/	Curtin University of Technology
Austria	
https://cran.wu.ac.at/	Wirtschaftsuniversität Wien
http://cran.wu.ac.at/	Wirtschaftsuniversität Wien
Belgium	
http://www.freestatistics.org/cran/	K.U.Leuven Association
https://lib.ugent.be/CRAN/	Ghent University Library
http://lib.ugent.be/CRAN/	Ghent University Library
Brazil	
http://nbcgib.uesc.br/mirrors/cran/	Center for Comp. Biol. at Universidade Estadual de Santa Cruz
https://cran-r.c3sl.ufpr.br/	Universidade Federal do Parana
http://cran-r.c3sl.ufpr.br/	Universidade Federal do Parana
https://cran.fiocruz.br/	Oswaldo Cruz Foundation, Rio de Janeiro
http://cran.fiocruz.br/	Oswaldo Cruz Foundation, Rio de Janeiro
https://vps.fmvz.usp.br/CRAN/	University of Sao Paulo, Sao Paulo
http://vps.fmvz.usp.br/CRAN/	University of Sao Paulo, Sao Paulo
https://brieger.esalq.usp.br/CRAN/	University of Sao Paulo, Piracicaba
http://brieger.esalq.usp.br/CRAN/	University of Sao Paulo, Piracicaba
Bulgaria	
https://ftp.uni-sofia.bg/CRAN/	Sofia University
http://ftp.uni-sofia.bg/CRAN/	Sofia University
Canada	
https://mirror.its.sfu.ca/mirror/CRAN/	Simon Fraser University, Burnaby
http://cran.stat.sfu.ca/	Simon Fraser University, Burnaby
https://muug.ca/mirror/cran/	Manitoba Unix User Group
http://muug.ca/mirror/cran/	Manitoba Unix User Group
https://mirror.its.dal.ca/cran/	Dalhousie University, Halifax
http://mirror.its.dal.ca/cran/	Dalhousie University, Halifax
http://cran.utstat.utoronto.ca/	University of Toronto
Chile	

Sounds great!
How do I begin?

- Step 1:
  - Download R

https://www.r-project.org/



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.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-07-05, Action of the Toes) R-3.6.1.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Sounds great! How do I begin?

- Step 1:
  - Download R

https://www.r-project.org/

R for Mac OS X

This directory contains binaries for a base distribution and packages to run on Mac OS X (release 10.6 and above). Mac OS 8.6 to 9.2 (and Mac OS X 10.1) are no longer supported but you can find the last supported release of R for these systems (which is R 1.7.1) here. Releases for old Mac OS X systems (through Mac OS X 10.5) and PowerPC Macs can be found in the old directory.

Note: CRAN does not have Mac OS X systems and cannot check these binaries for viruses. Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

As of 2016/03/01 package binaries for R versions older than 2.12.0 are only available from the CRAN archive so users of such versions should adjust the CRAN mirror setting accordingly.

R 3.6.1 "Action of the Toes" released on 2019/07/05

Important: since R 3.4.0 release we are now providing binaries for OS X 10.11 (El Capitan) and higher using non-Apple toolkit to provide support for OpenMP and C++17 standard features. To compile packages you may have to download tools from the tools directory and read the corresponding note below.

Please check the MD5 checksum of the downloaded image to ensure that it has not been tampered with or corrupted during the mirroring process. For example type
`md5 R-3.6.1.pkg`
in the *Terminal* application to print the MD5 checksum for the R-3.6.1.pkg image. On Mac OS X 10.7 and later you can also validate the signature using
`pkgutil --check-signature R-3.6.1.pkg`

Latest release:

R-3.6.1.pkg
MD5-hash: 27 662103dfe6a625b4573143cb995
SHA1-hash: 32f8e5013870d2a9179b54eaee277f41657b0
(ca 6M)

R 3.6.1 binary for OS X 10.11 (El Capitan) and higher, signed package. Contains R 3.6.1 framework, R.app GUI 1.70 in 64-bit for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 5.2. The latter two components are optional and can be ommitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

Note: the use of X11 (including `tcltk`) requires XQuartz to be installed since it is no longer part of OS X. Always re-install XQuartz when upgrading your macOS to a new major version.

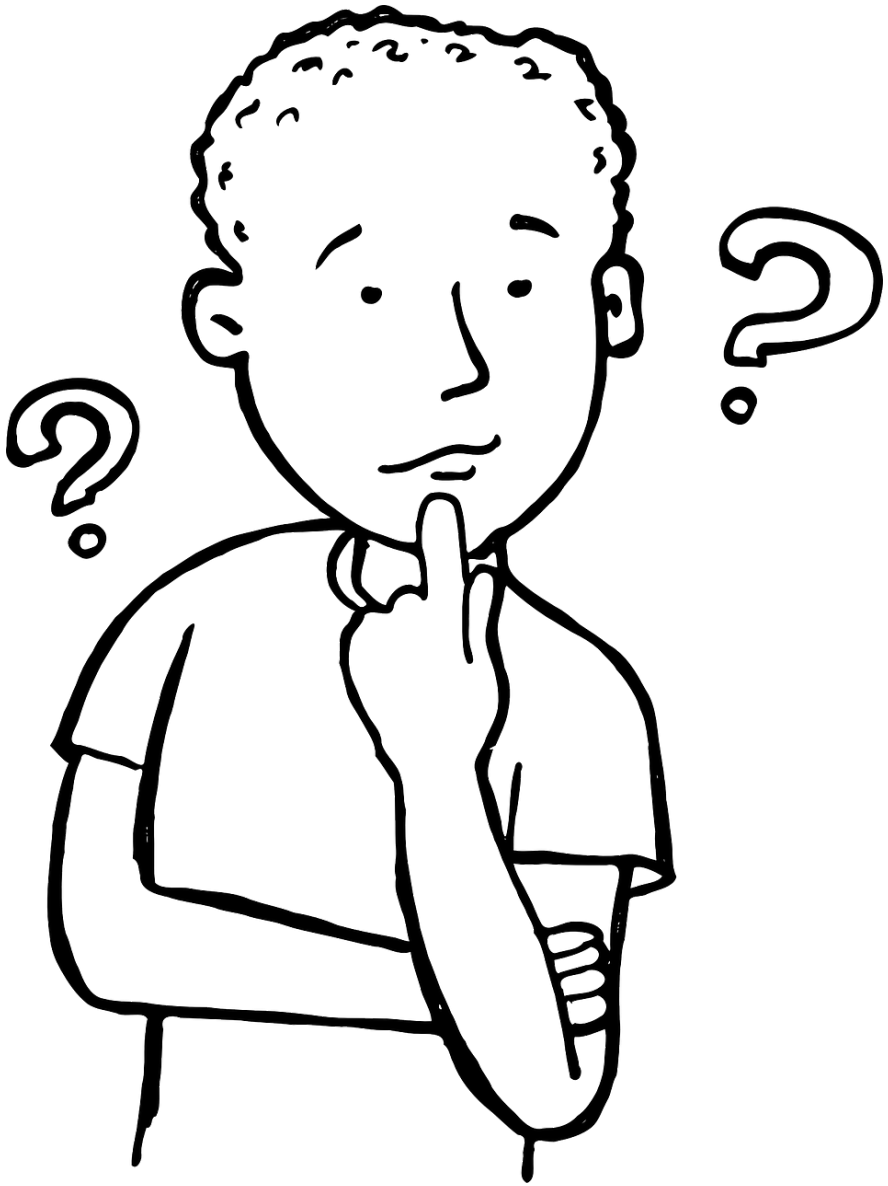
Important: this release uses Clang 7.0.0 and GNU Fortran 6.1, neither of which is supplied by

R Console

~

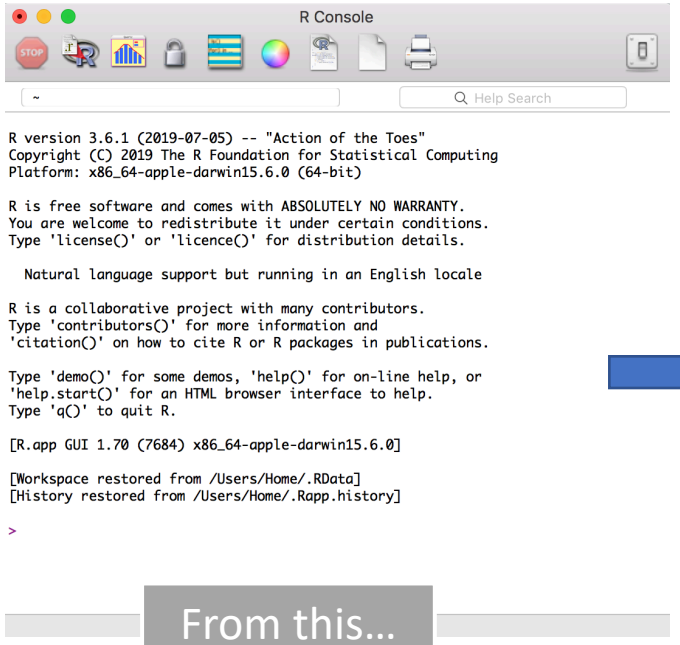
Help Search

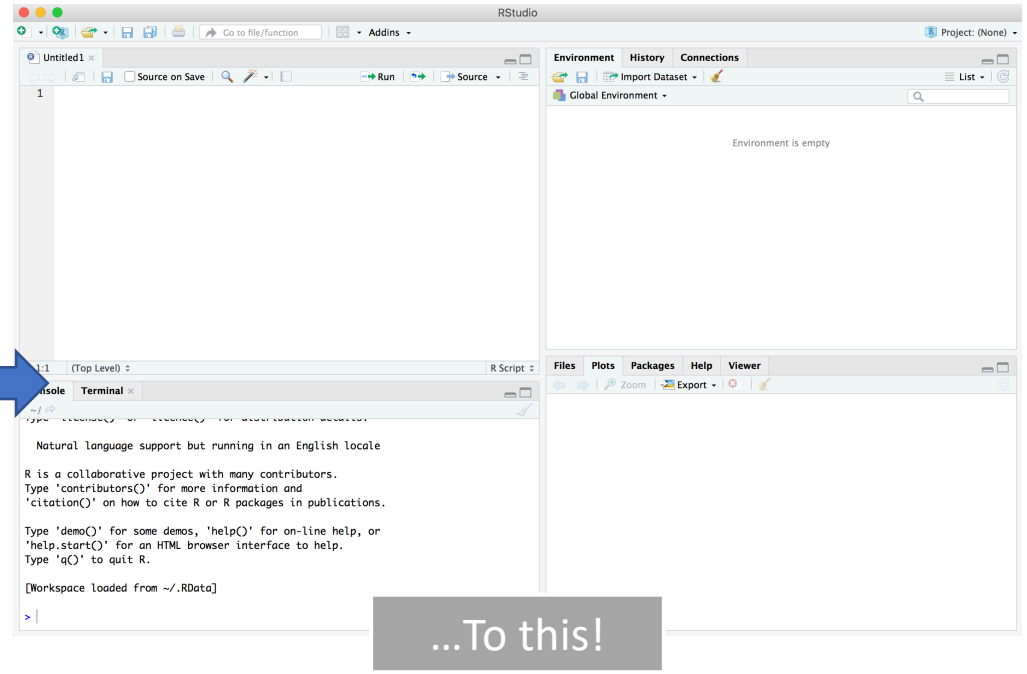
```
R version 3.6.1 (2019-07-05) -- "Action of the Toes"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

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.

[R.app GUI 1.70 (7684) x86_64-apple-darwin15.6.0]

[Workspace restored from /Users/Home/.RData]
[History restored from /Users/Home/.Rapp.history]

>
```



R Studio… R's best friend

- R Studio is an environment for running R. It makes running your analyses in R much more user friendly

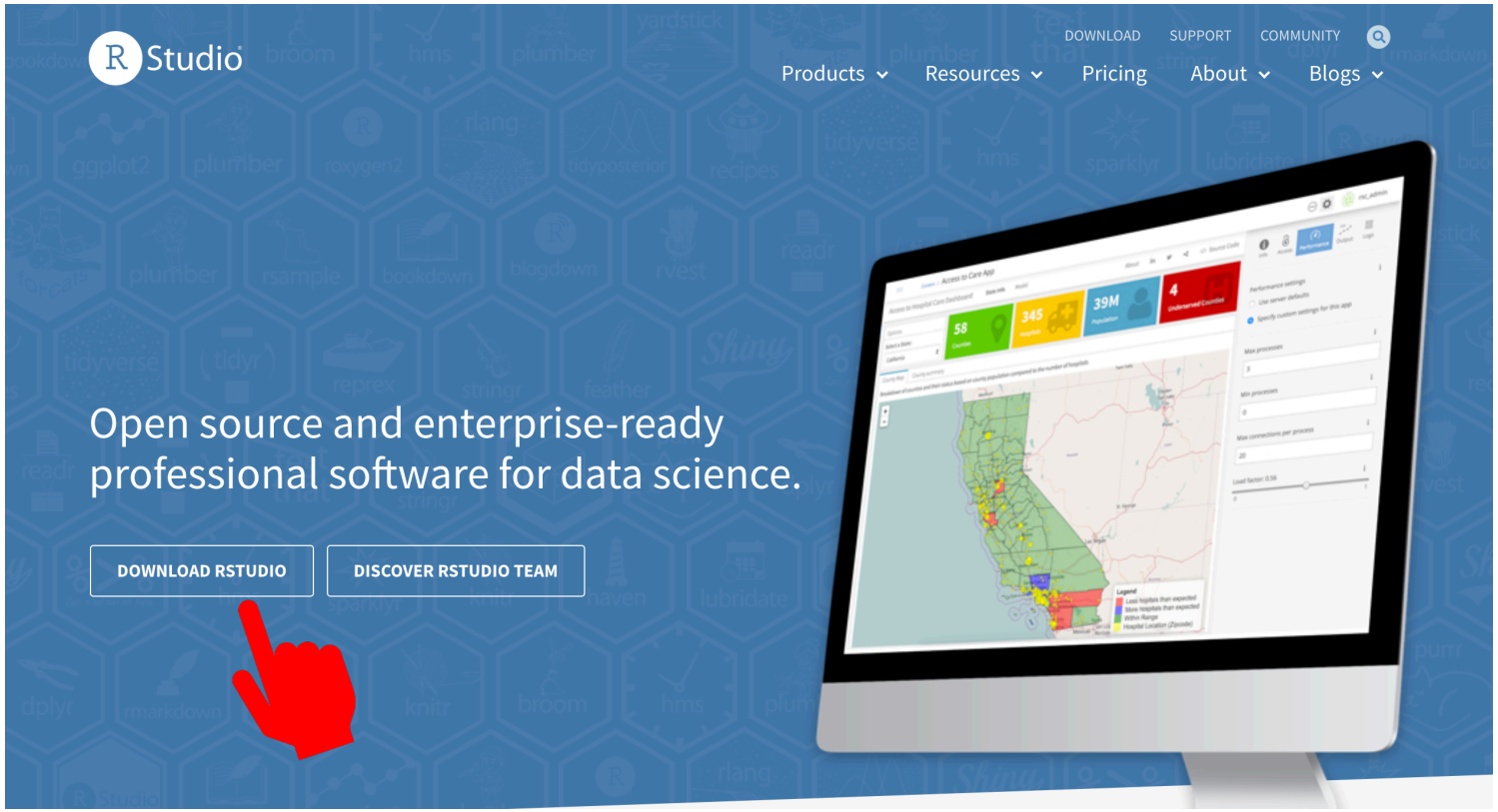




Sounds great! How do I begin?

- Step 1:
  - Download R
- Step 2:
  - Download R Studio

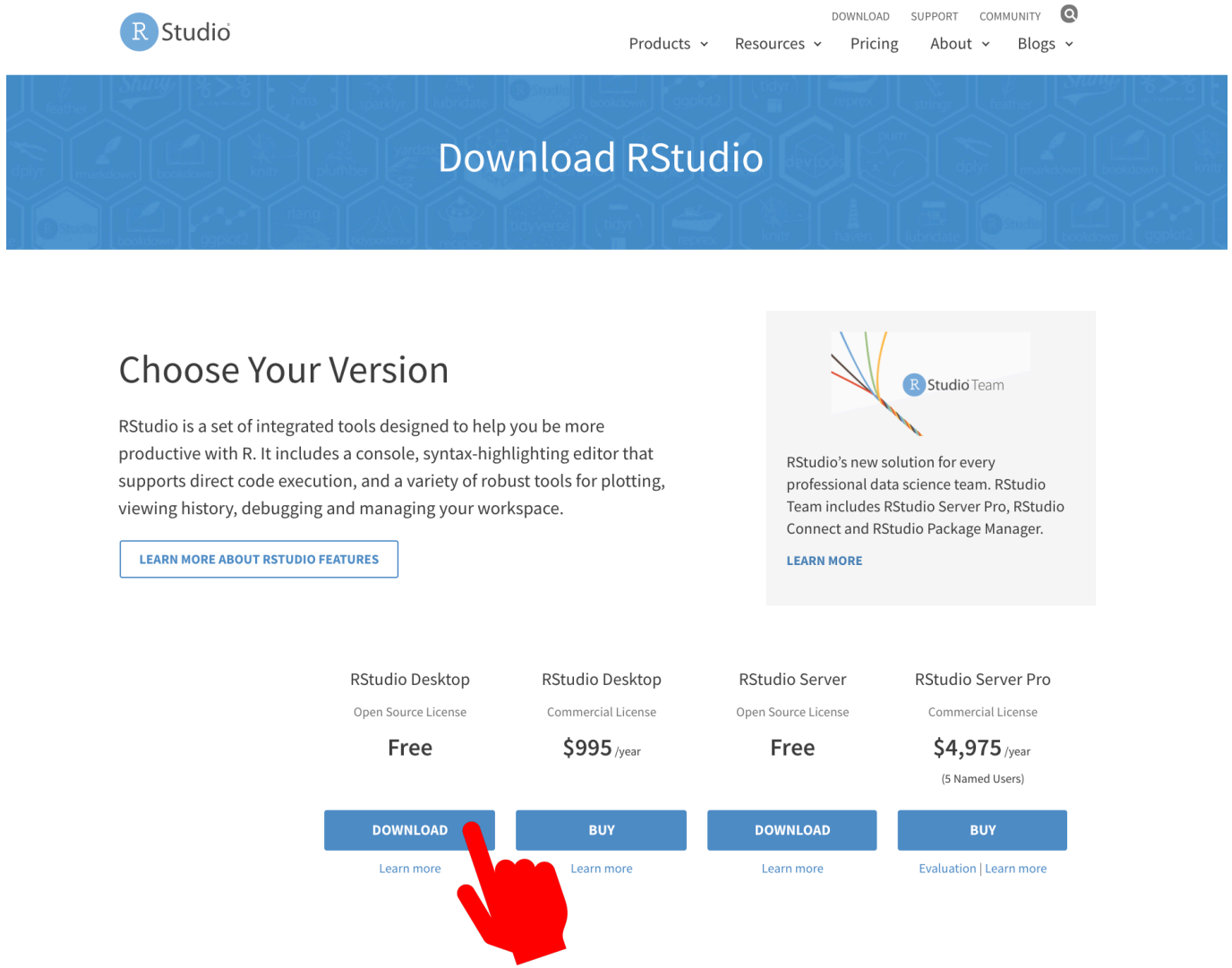
https://www.rstudio.com/



Sounds great! How do I begin?

- Step 1:
  - Download R
- Step 2:
  - Download R Studio

https://www.rstudio.com/



Sounds great! How do I begin?

- Step 1:
  - Download R
- Step 2:
  - Download R Studio

https://www.rstudio.com/

RStudio Desktop 1.2.5001 - Release Notes

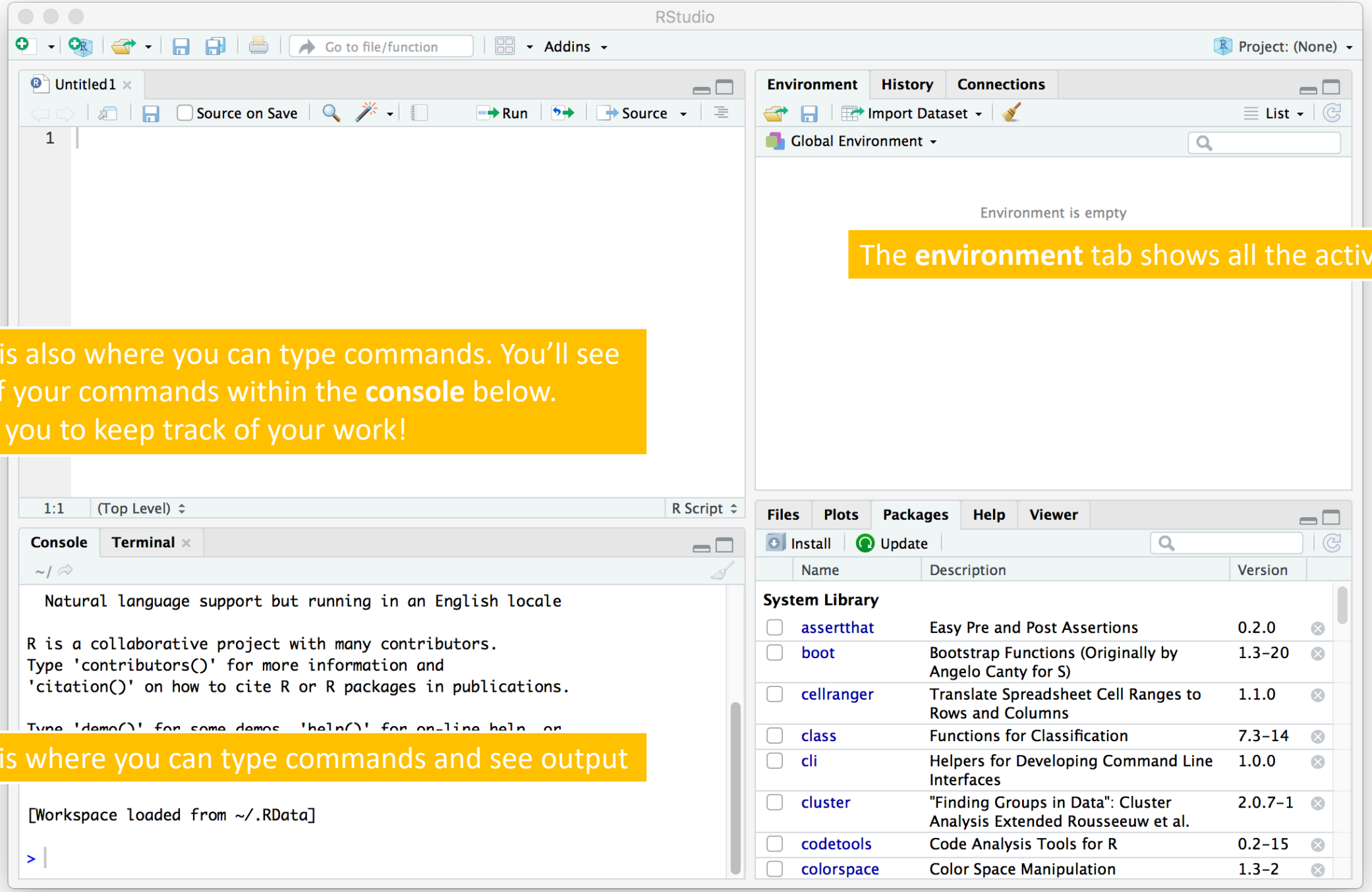
RStudio requires R 3.0.1+. If you don't already have R, download it here.

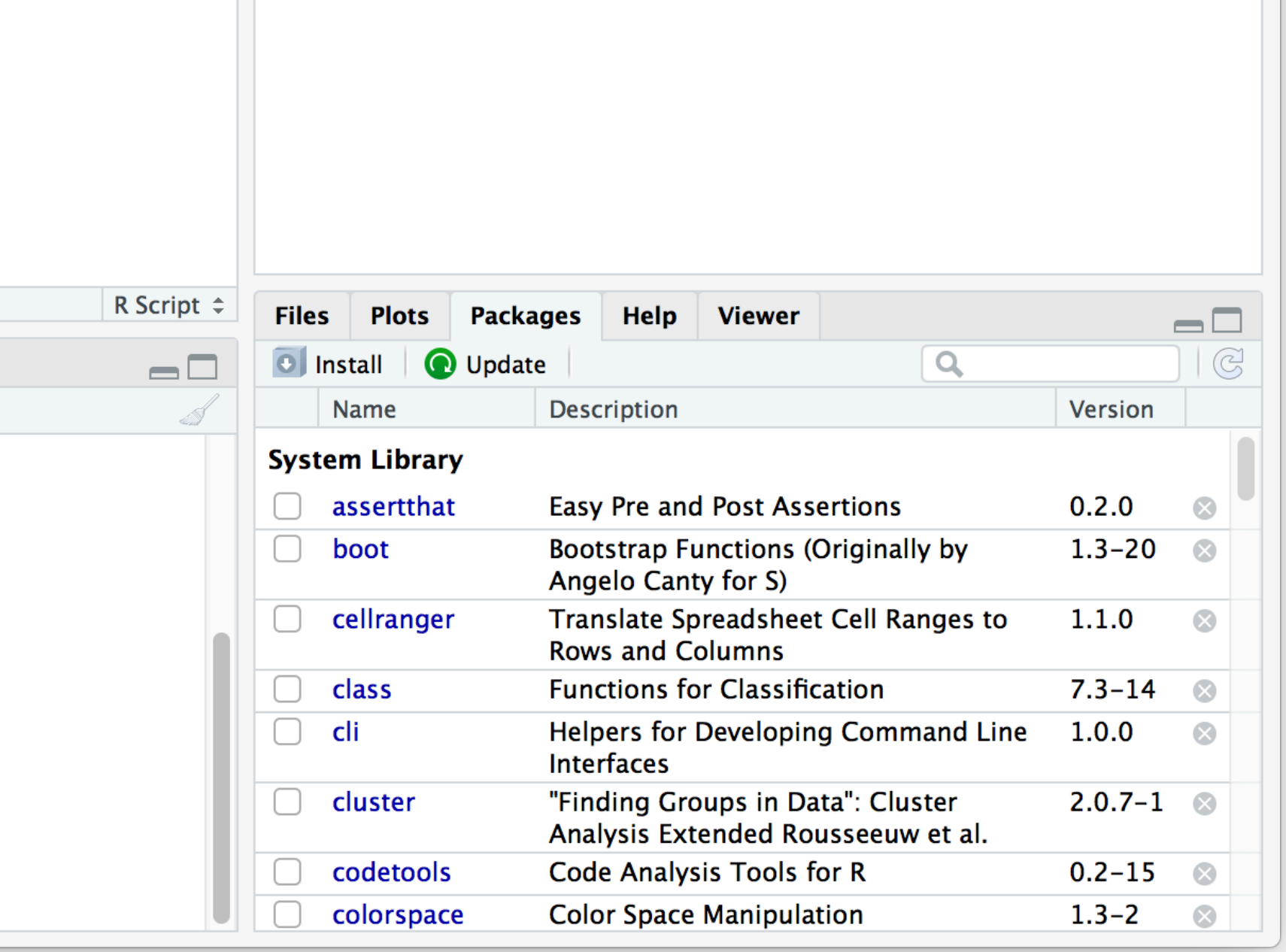
Linux users may need to import RStudio's public code-signing key prior to installation, depending on the operating system's security policy.

RStudio 1.2 requires a 64-bit operating system, and works exclusively with the 64 bit version of R. If you are on a 32 bit system or need the 32 bit version of R, you can use an older version of RStudio.

Installers for Supported Platforms

Installers	Size	Date	MD5
RStudio 1.2.5001 - Ubuntu 18/Debian 10 (64-bit)	105.43 MB	2019-09-19	f108e4d5c1b6c19690378b3ca0990249
RStudio 1.2.5001 - Debian 9 (64-bit)	105.70 MB	2019-09-19	23dca12a5e0a0849522f05b4a8600ce8
RStudio 1.2.5001 - Fedora 28/Red Hat 8 (64-bit)	120.90 MB	2019-09-19	45eab0baf8d0504d183f09c8d40ae704
RStudio 1.2.5001 - macOS 10.12+ (64-bit)	126.86 MB	2019-09-19	a4d8ee737818158b272450eaad4bdc4f
RStudio 1.2.5001 - SLES/OpenSUSE 12 (64-bit)	[illegible]	2019-09-19	ca94a9bb7e7f5474eedd233ddeef14d6
RStudio 1.2.5001 - OpenSUSE 15 (64-bit)	[illegible]	2019-09-19	2a40768fdc5c5f97cd1d40628bf8aaad
RStudio 1.2.5001 - Fedora 19/Red Hat 7 (64-bit)	120.27 MB	2019-09-19	14bc52d6f78bc4ee22abff2298be919f
RStudio 1.2.5001 - Ubuntu 14/Debian 8 (64-bit)	96.93 MB	2019-09-19	c19b0ece90130bed7248c1bf6001c647
RStudio 1.2.5001 - Windows 10/8/7 (64-bit)	149.83 MB	2019-09-19	c54d8779f363ec9636c7831e577521bd
RStudio 1.2.5001 - Ubuntu 16 (64-bit)	104.88 MB	2019-09-19	4ff58a08c305207729feb65eb145a5a2


RStudio
Go to file/function
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
1
Environment
History
Connections
Import Dataset
List
Global Environment
Environment is empty
The environment tab shows all the active objects
The R Script is also where you can type commands. You'll see the results of your commands within the console below. Scripts allow you to keep track of your work!
1:1
(Top Level)
R Script
Console
Terminal
~/
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.
The console is where you can type commands and see output
[Workspace loaded from ~/.RData]
>
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
System Library
assertthat
Easy Pre and Post Assertions
0.2.0
boot
Bootstrap Functions (Originally by Angelo Canty for S)
1.3-20
cellranger
Translate Spreadsheet Cell Ranges to Rows and Columns
1.1.0
class
Functions for Classification
7.3-14
cli
Helpers for Developing Command Line Interfaces
1.0.0
cluster
"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.
2.0.7-1
codetools
Code Analysis Tools for R
0.2-15
colorspace
Color Space Manipulation
1.3-2



The **files** tab shows all the files and folders in your default workspace as if you were searching your own computer.

The **plots** tab will show all of your graphs as you generate them.

The **packages** tab will list packages you have (or added-on) needed to run specific processes.

The **help** tab gives you extra info about functions, packages, you name it!

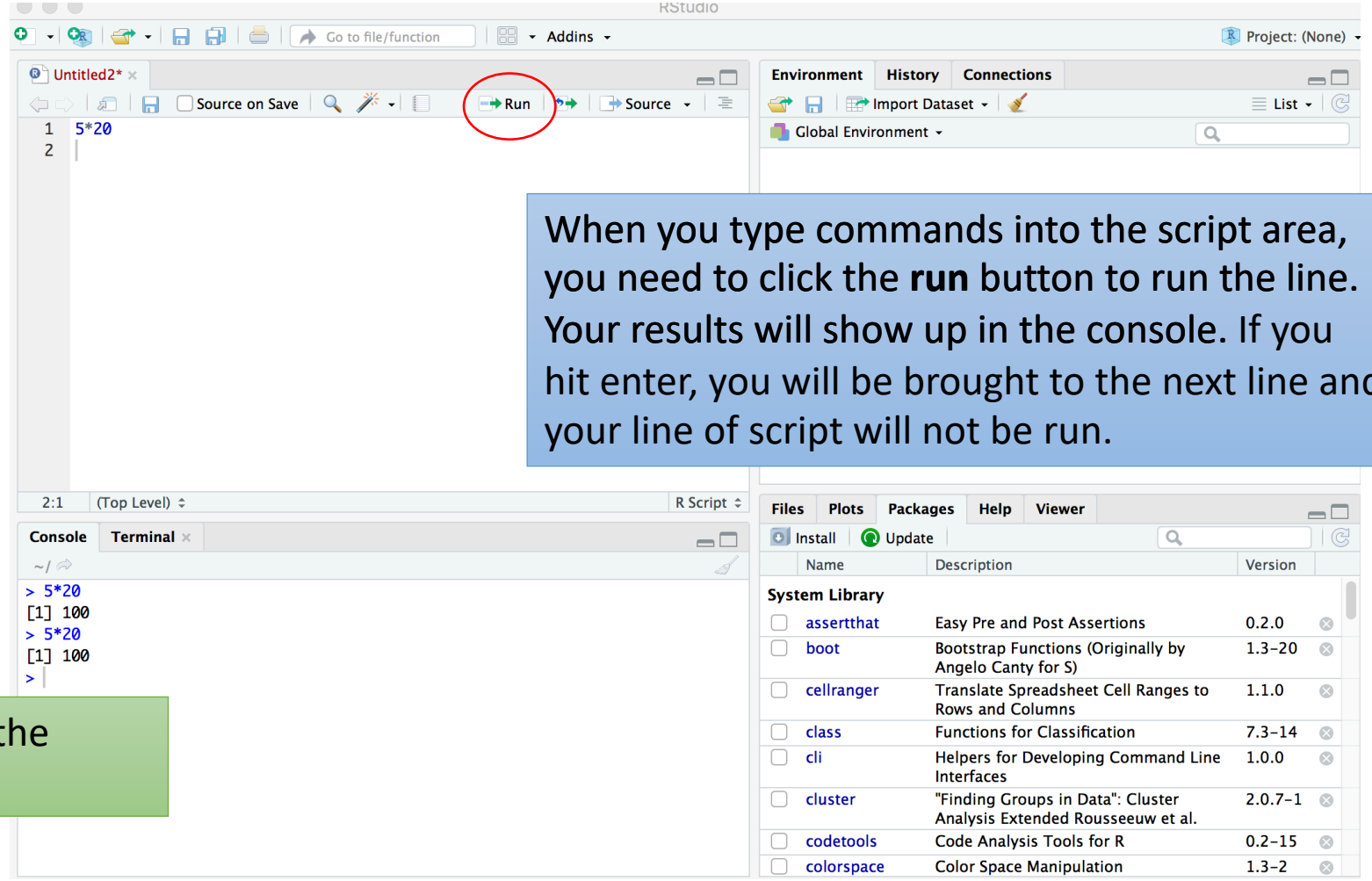
Let’s give it a try!

Basic functions explored!

Source material: (Smyth and Johnson, 2015)

Using the console and script areas

- As previously mentioned, one of the uses of R is a calculator
- Try multiplying 5 by 20



When you type commands into the script area, you need to click the **run** button to run the line. Your results will show up in the console. If you hit enter, you will be brought to the next line and your line of script will not be run.

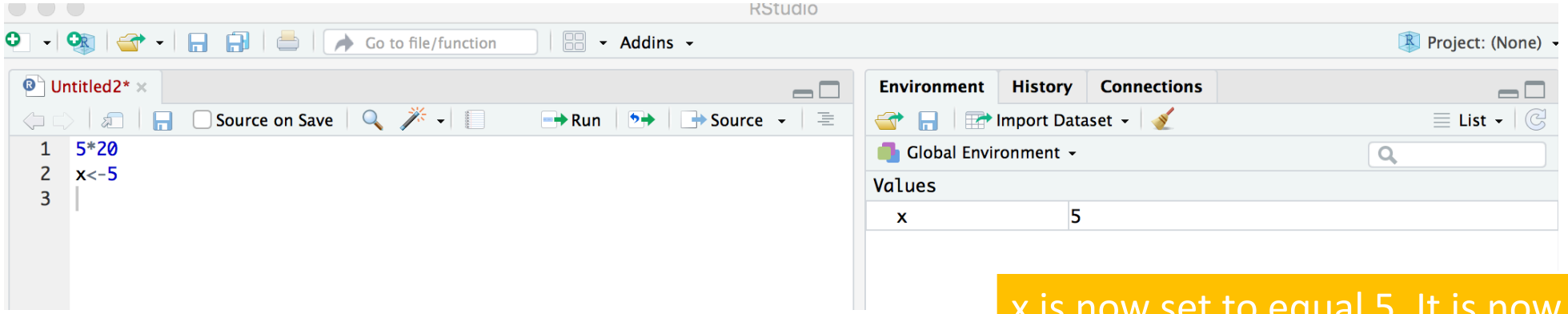
When you type commands directly into the **console**, hitting enter will run your line.

Creating variables

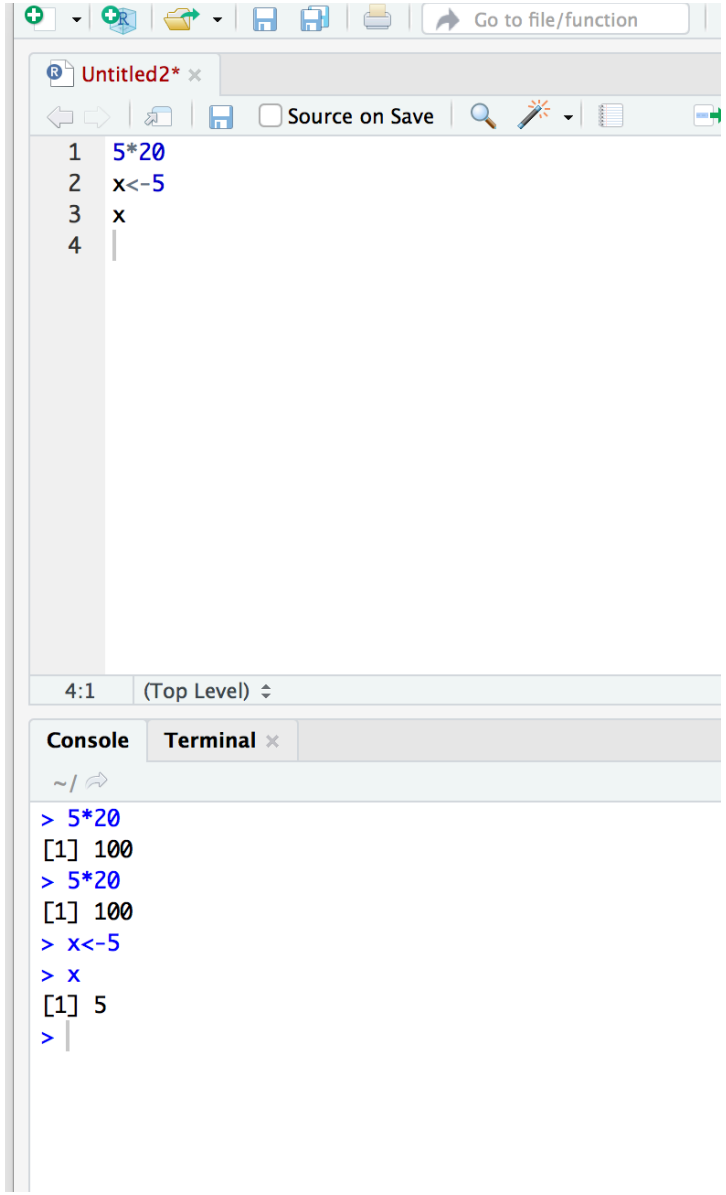
- A variable is a symbol that represents a number.

- In R, you need to assign a number to the symbol. A generally accepted way to make assignments is by using “<-”, with the variable on the left and the number (or the variable’s value) on the right.

- For example, in creating a new variable “x”, type the following into the script window: **x<-5**

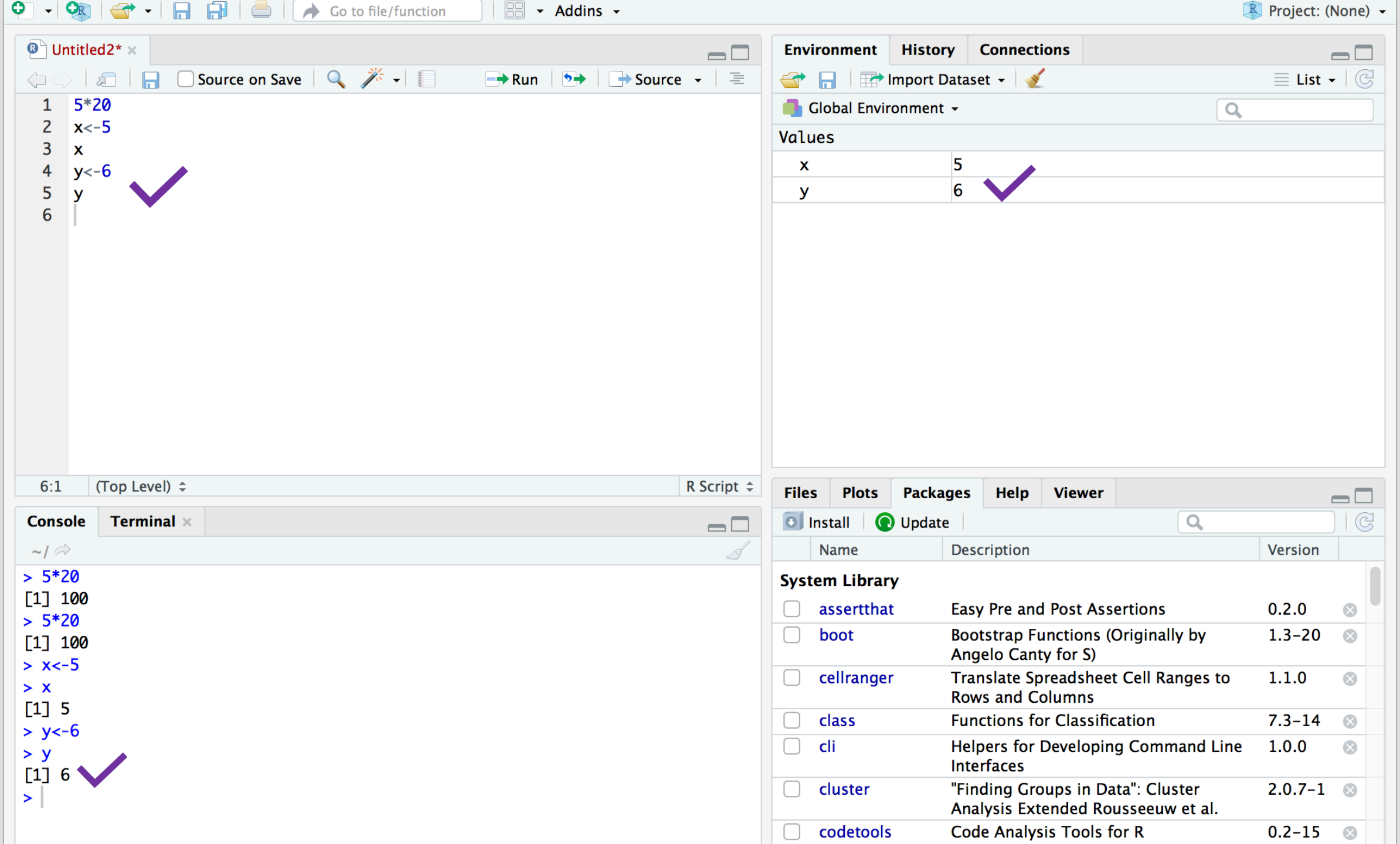


x is now set to equal 5. It is now an object in the workspace.



Creating variables

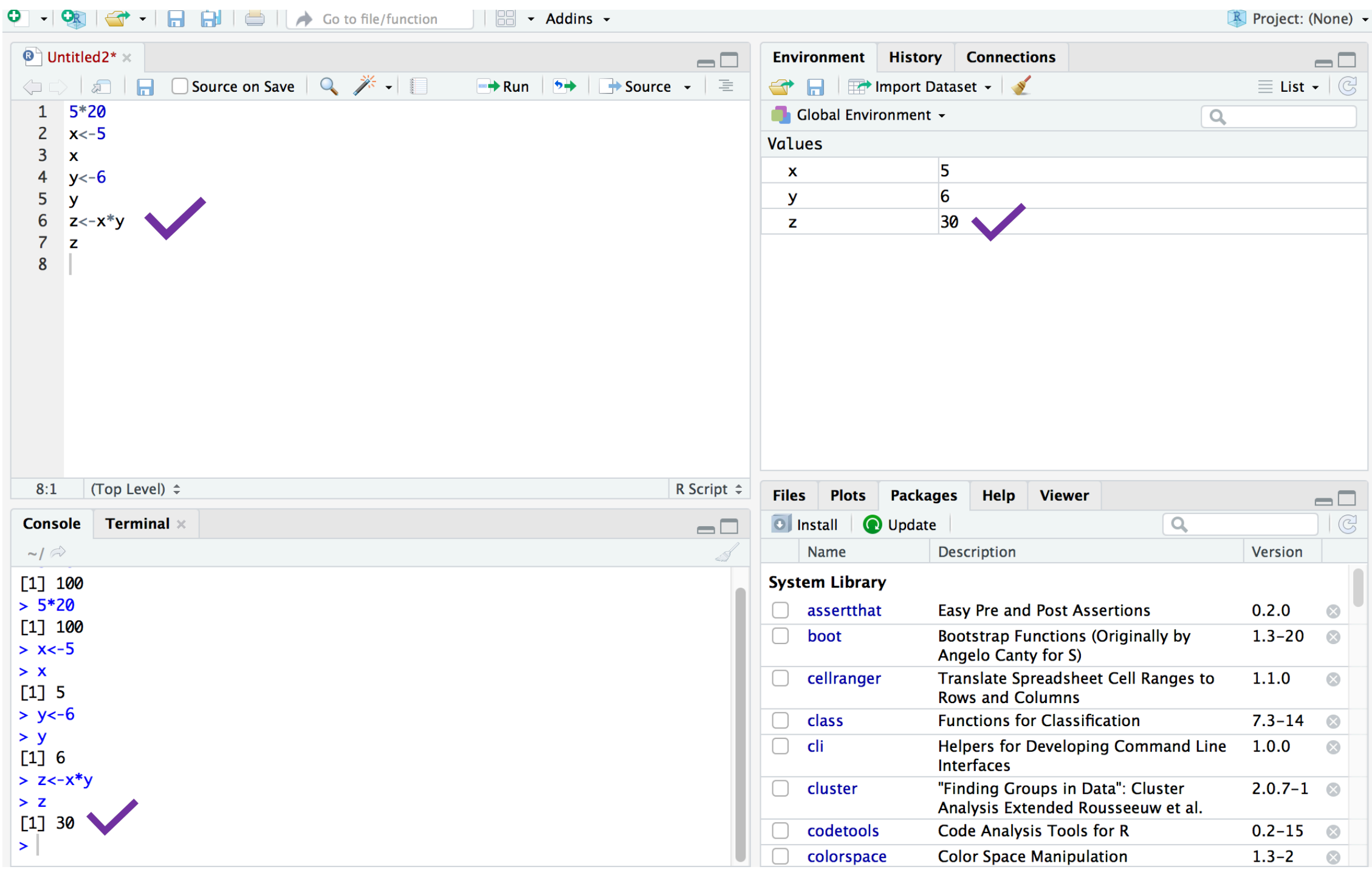
- Every time you call on “x”, it will return the value as 5.
- For example, if you run “**x**” in the script, in the console it will produce “5”
- **Now, create a new variable, ”y”, that equals to 6.**


Go to file/function
Addins
Project: (None)
Untitled2*
Source on Save
Run
Source
1 5*20
2 x<-5
3 x
4 y<-6
5 y
6
6:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Values
x 5
y 6
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
System Library
assertthat Easy Pre and Post Assertions 0.2.0
boot Bootstrap Functions (Originally by Angelo Canty for S) 1.3-20
cellranger Translate Spreadsheet Cell Ranges to Rows and Columns 1.1.0
class Functions for Classification 7.3-14
cli Helpers for Developing Command Line Interfaces 1.0.0
cluster "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al. 2.0.7-1
codetools Code Analysis Tools for R 0.2-15
Console
Terminal
~/
> 5*20
[1] 100
> 5*20
[1] 100
> x<-5
> x
[1] 5
> y<-6
> y
[1] 6
>

Creating variables

- We can now use these variables to create new variables. For example, type the following into your script to create a new variable, z:

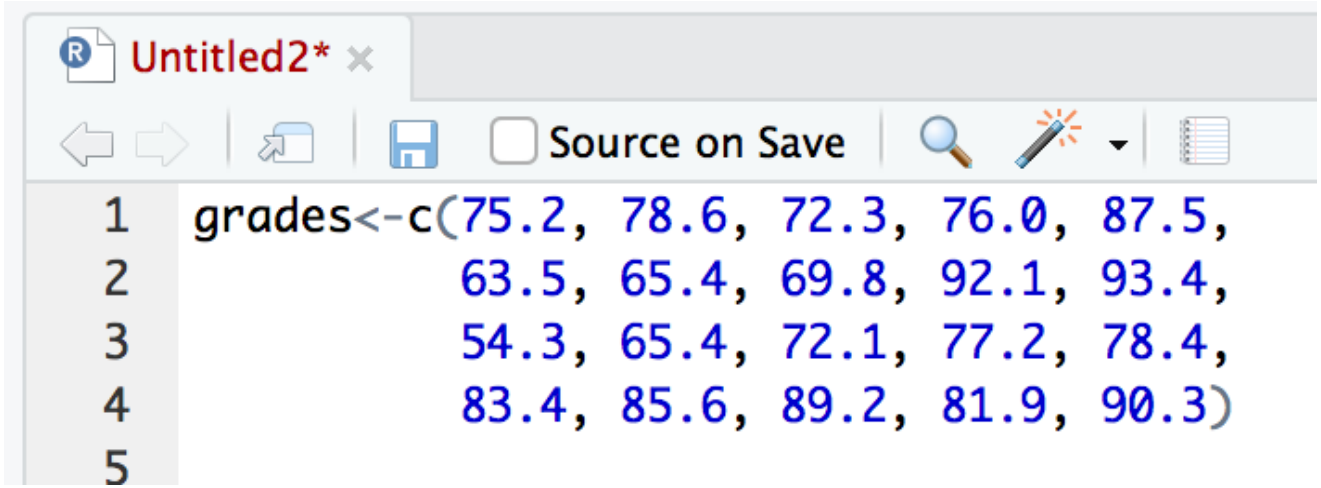
```
z<-x*y
```


Go to file/function
Addins
Project: (None)
Untitled2*
Source on Save
Run
Source
1 5*20
2 x<-5
3 x
4 y<-6
5 y
6 z<-x*y
7 z
8
8:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global Environment
Values
x 5
y 6
z 30
Console
Terminal
~/
[1] 100
> 5*20
[1] 100
> x<-5
> x
[1] 5
> y<-6
> y
[1] 6
> z<-x*y
> z
[1] 30
>
Files
Plots
Packages
Help
Viewer
Install
Update
Name
Description
Version
System Library
assertthat Easy Pre and Post Assertions 0.2.0
boot Bootstrap Functions (Originally by Angelo Canty for S) 1.3-20
cellranger Translate Spreadsheet Cell Ranges to Rows and Columns 1.1.0
class Functions for Classification 7.3-14
cli Helpers for Developing Command Line Interfaces 1.0.0
cluster "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al. 2.0.7-1
codetools Code Analysis Tools for R 0.2-15
colorspace Color Space Manipulation 1.3-2

Vectors, Factors, and Objects in R

- **Vectors** are the most basic data structure in R. If you have more than one number in a variable, you can create an **object** that creates a vector.

- For example, if you had biology grades for 20 different students, you can combine them into one object. Let's call that object "grades".



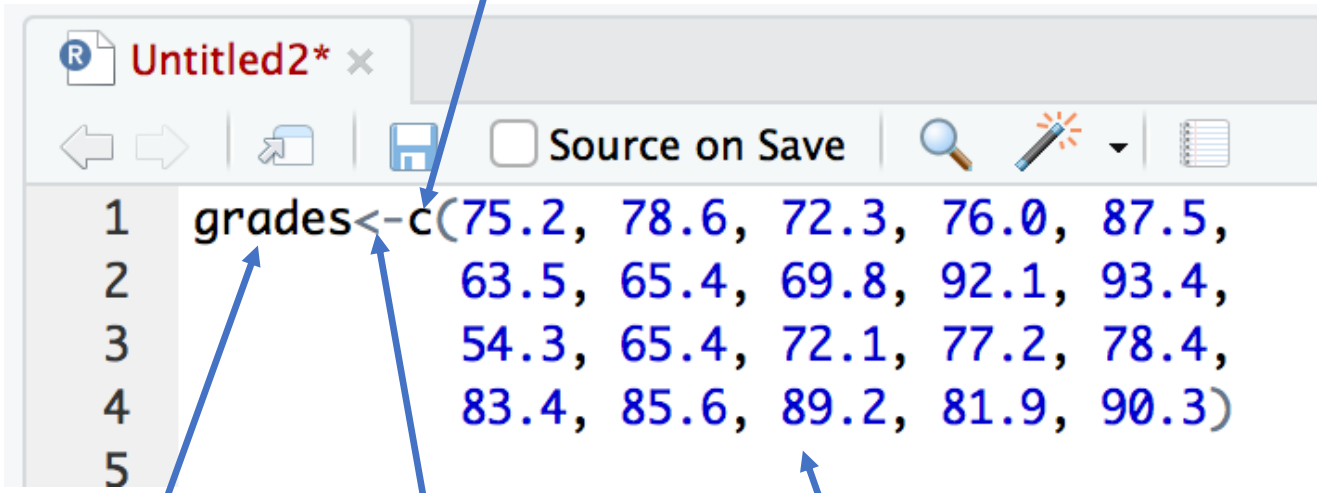
Type this directly into the script area. For ease of reading, I have hit enter after a set of five students. If you were to type this directly into the console, you should not hit enter until you're done typing everything in the parentheses.

Vectors, Factors, and Objects in R

- **Vectors** are the most basic data structure in R. If you have more than one number in a variable, you can create an **object** that creates a vector.

- For example, if you had biology grades for 20 different students, you can combine them into one object. Let's call that object "grades":

"c" is a **function** in R. Functions describe what you want done with the specified data. This function stands for *combine*. To learn more about specific functions, type the function into the search window of the help tab



The new **object** name is "grades". These are case sensitive! "**G**rades" is not the same as "**g**rades"

The grades object is **assigned** as the combined data points.

This is the raw data that will be *combined* into the new object, "grades", using the "c" function. Data used in the function is contained within parentheses, and individual data points are separated by commas.

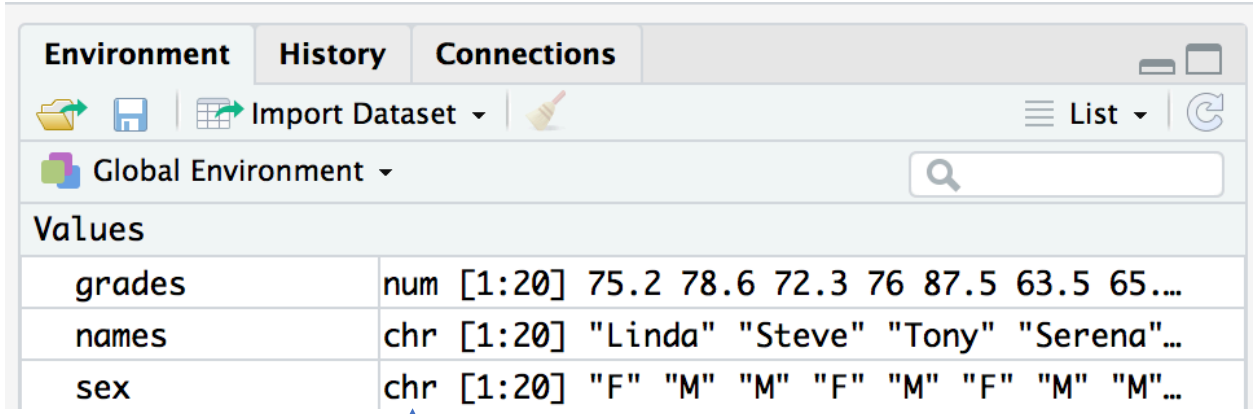
Vectors, Factors, and Objects in R

- **R can also handle non-numeric data. Let's create a vector of student names and sex to go along with the grade data we just entered.**

- **We'll call the first object "names" and the second object "sex".**

```
5  names<-c("Linda", "Steve", "Tony", "Serena", "Will",
6           "Maggie", "John", "Jason", "Charlie", "Raj",
7           "Becca", "Russ", "Kevin", "Chelsea", "Astrid",
8           "Mary", "Priya", "Sanders", "Daniel", "Tess")
9  sex<-c("F", "M", "M", "F", "M",
10        "F", "M", "M", "M", "M",
11        "F", "M", "M", "F", "F",
12        "F", "F", "M", "M", "F")
```

Non-numeric data should be surrounded by quotation marks, separated by commas.



- Objects “names” and “sex” are now **character vectors**, because they contain non-numeric information.
- The “names” vector probably won’t be used for any actual analyses, so we can leave it as is.
- However, if we wanted to use the “sex” variable as a grouping variable in a statistical analysis (such as a t-test), we might want to turn it into a **factor.**

Converting an object to a factor

- This is very easily done using the “factor” function.

- Your line of command would look something like this:

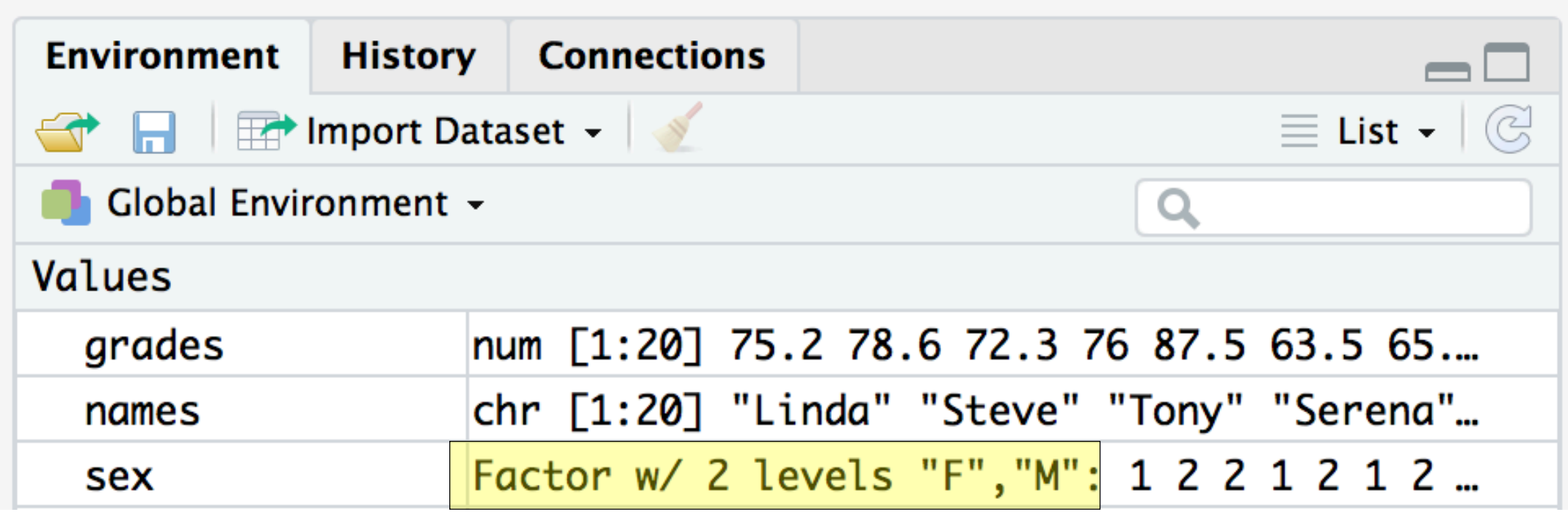
```
13   sex<-factor(sex)
```

New object name.
Note: we’re using the same name as before because we’re intending to replace the previous object named “sex” with the new version.

Assigning the results of the “factor” function on the previous object labeled “sex” to a new object called “sex”.

The function “factor” is applied to the object called “sex”, indicated in parentheses.

Converting an object to a factor



This object is now a “factor” with two levels (“F”, and “M”), and so it will behave differently in analyses.

Why do you think female is coded as 1, and male as 2?

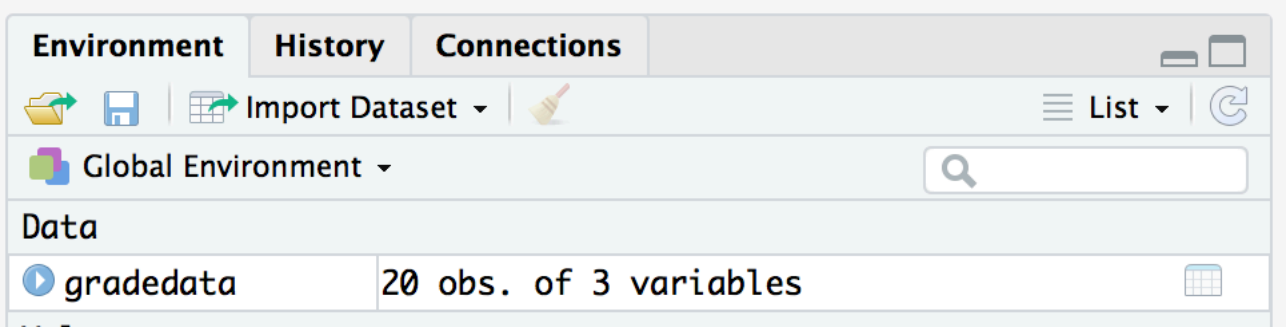
Creating datasets in R

- It's common to see groupings of objects contained within a **dataset**. For your own research purposes, you're likely to import your own .csv file containing your dataset as opposed to entering each variable separately.
- Let's create a new dataset from our three objects. R calls datasets **dataframes**, and dataframes can be created using the "data.frames" function:

```
16  gradedata<-data.frame(names,sex,grades)
```

List your objects in the order in which you want them to appear in your dataframe. Object names are separated by commas.

Creating datasets in R



The dataset contains 20 observations of 3 variables. It is now an object contained in the environment.

The “head” function can be used to get a quick snapshot of the data. It will produce the column (variable) names and the first six rows of data.

Now each participant has a linked sex and grade within a single object.

```
> head(gradedata)
   names sex grades
1  Linda   F   75.2
2  Steve   M   78.6
3   Tony   M   72.3
4 Serena   F   76.0
5   Will   M   87.5
6 Maggie   F   63.5
```

Pulling information from the dataframe

- You can access specific values in a dataframe using the data point's coordinates in square brackets:

  Dataframe[row,column]

- So in our example, if we wanted the grade (found in the third column) for the fourth student (each row corresponds to a student), our command line would look like this:

```
> gradedata[4,3]
[1] 76
```

- Or if we wanted all the data on participant "Steve" (second student):

```
> gradedata[2,]
  names sex grades
2 Steve   M   78.6
```

- Or suppose I wanted all of the data points of one variable in the dataset. I can pull out single variables by using the “$” sign:

```
> gradedata$grades
 [1] 75.2 78.6 72.3 76.0 87.5 63.5 65.4 69.8 92.1 93.4 54.3 65.4 72.1 77.2
[15] 78.4 83.4 85.6 89.2 81.9 90.3
```

Note: in our case, our variables are still in our environment. But this function becomes especially useful when we are importing datasets and want to pull out variables not already in the environment.

- I can also pull out the grade data of all females in the dataset using the “subset” function. By using two = signs, we are telling R that we want the subset of the data where “sex is exactly equal to F”:

```
> subset(gradedata, sex=="F")
     names sex grades
1    Linda   F   75.2
4   Serena   F   76.0
6   Maggie   F   63.5
11   Becca   F   54.3
14 Chelsea   F   77.2
15  Astrid   F   78.4
16    Mary   F   83.4
17   Priya   F   85.6
20    Tess   F   90.3
```

This command line effectively reads:

“I want a **subset** of the **gradedata** dataframe consisting of cases where sex is exactly equal to F”

Descriptive statistics

- Find the mean with the “mean” function:

```
> mean(grades)
[1] 77.58
```

or

```
> mean(gradedata$grades)
[1] 77.58
```

- Find the median with the “median” function:

```
> median(grades)
[1] 77.8
```

or

```
> median(gradedata$grades)
[1] 77.8
```

- Find the standard deviation with the “sd” function:

```
> sd(grades)
[1] 10.60921
```

or

```
> sd(gradedata$grades)
[1] 10.60921
```

t-test in R

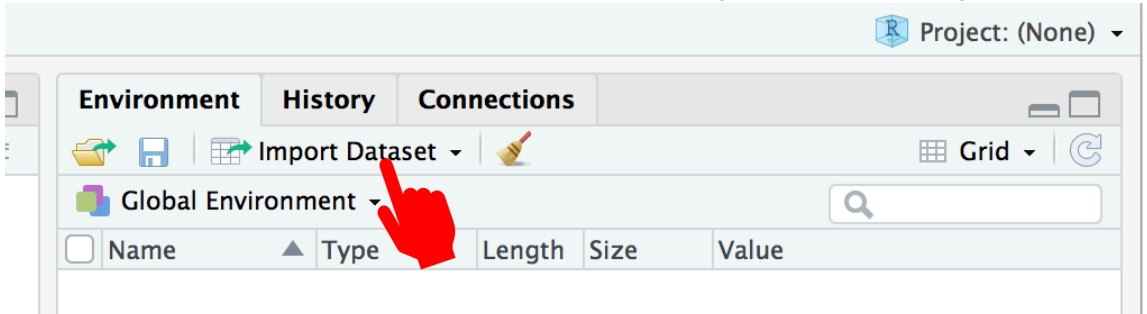
Simple independent two-group comparison

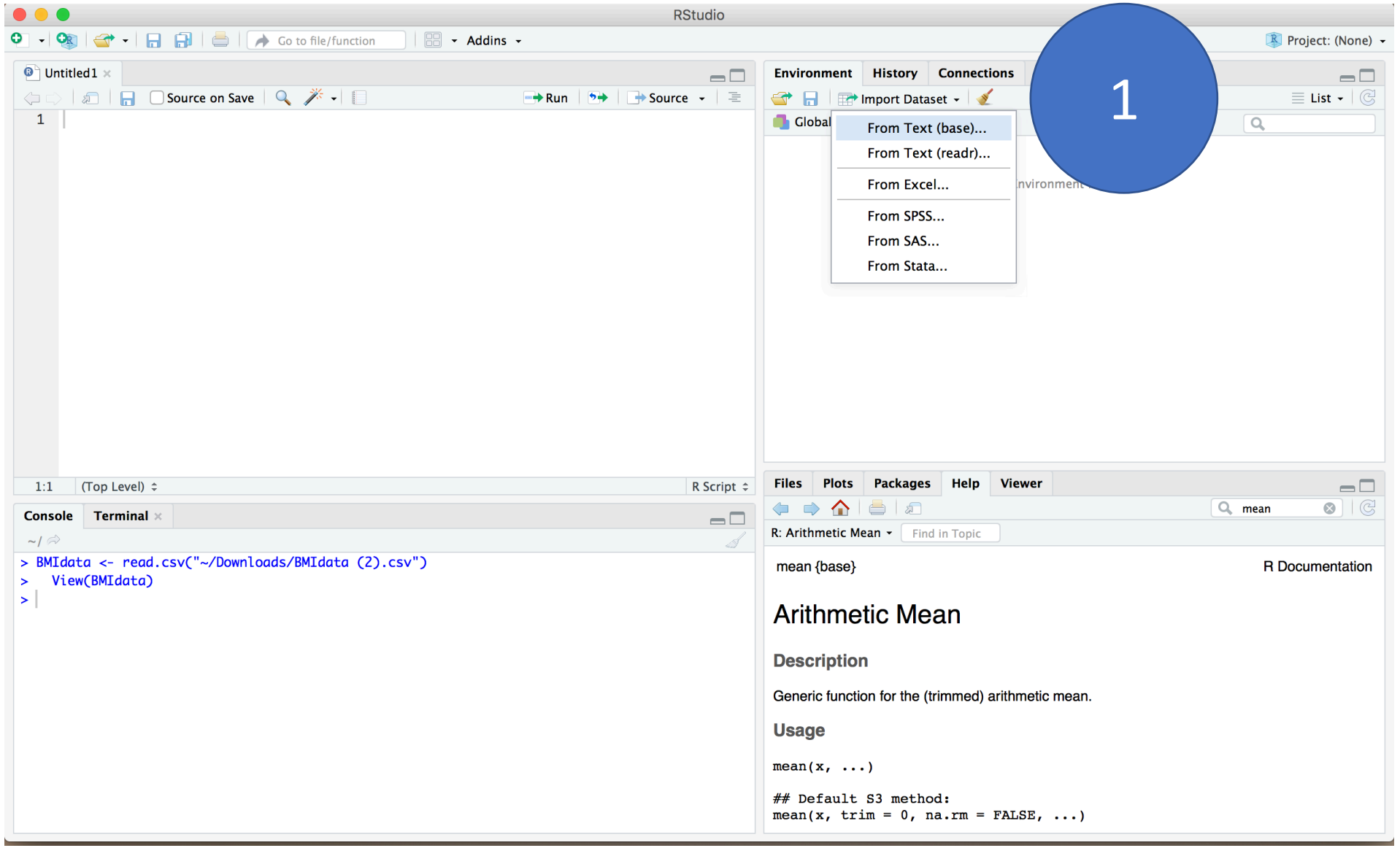
Source material: (Johnson, 2015)

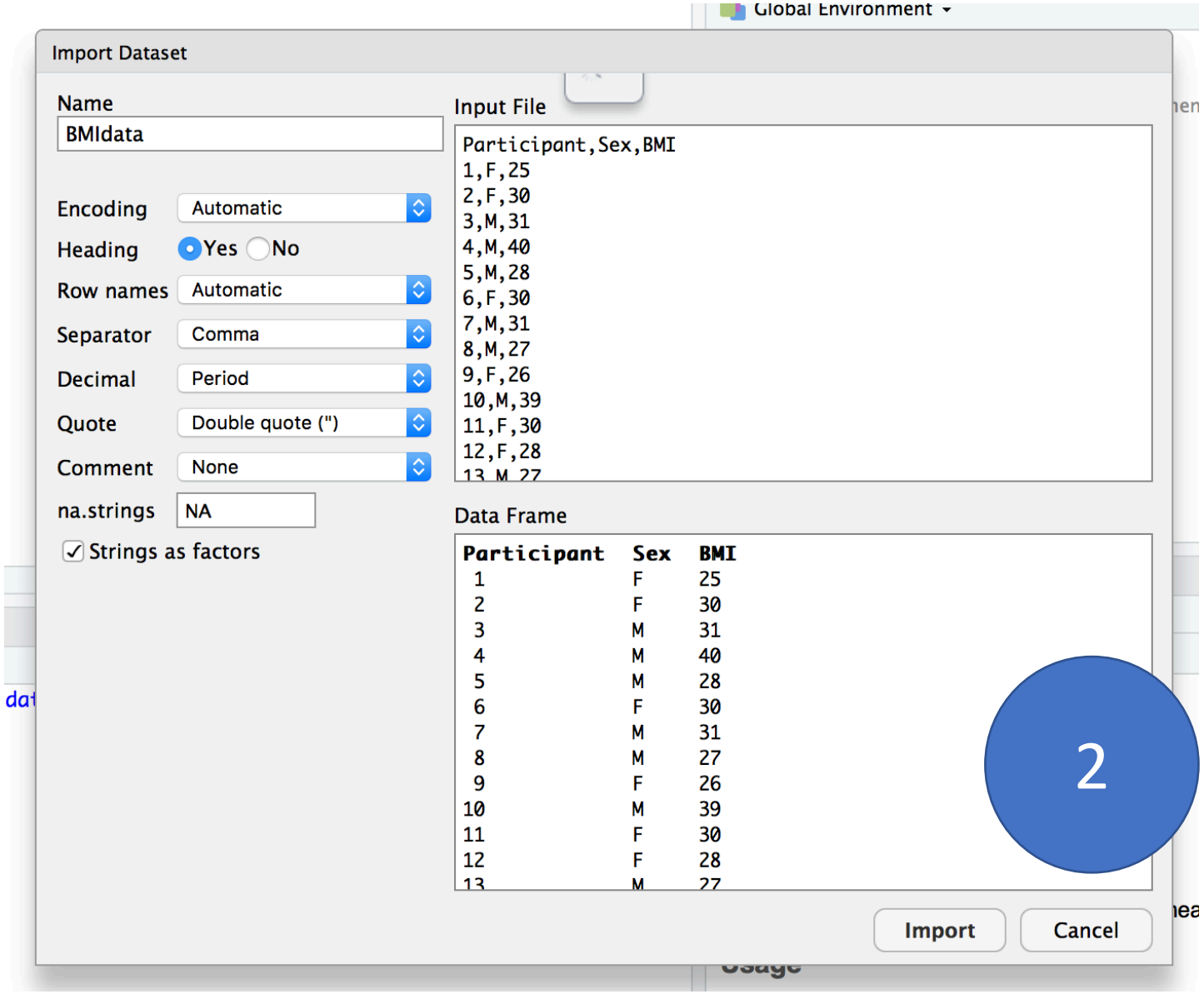
Step 1: Download the Data and Import into R Studio

- Head to: https://tinyurl.com/BMIdata2019
- Dropbox link to a fake dataset I created
  - If you're not a Dropbox user, click "Download" at the top right, then "Direct download"

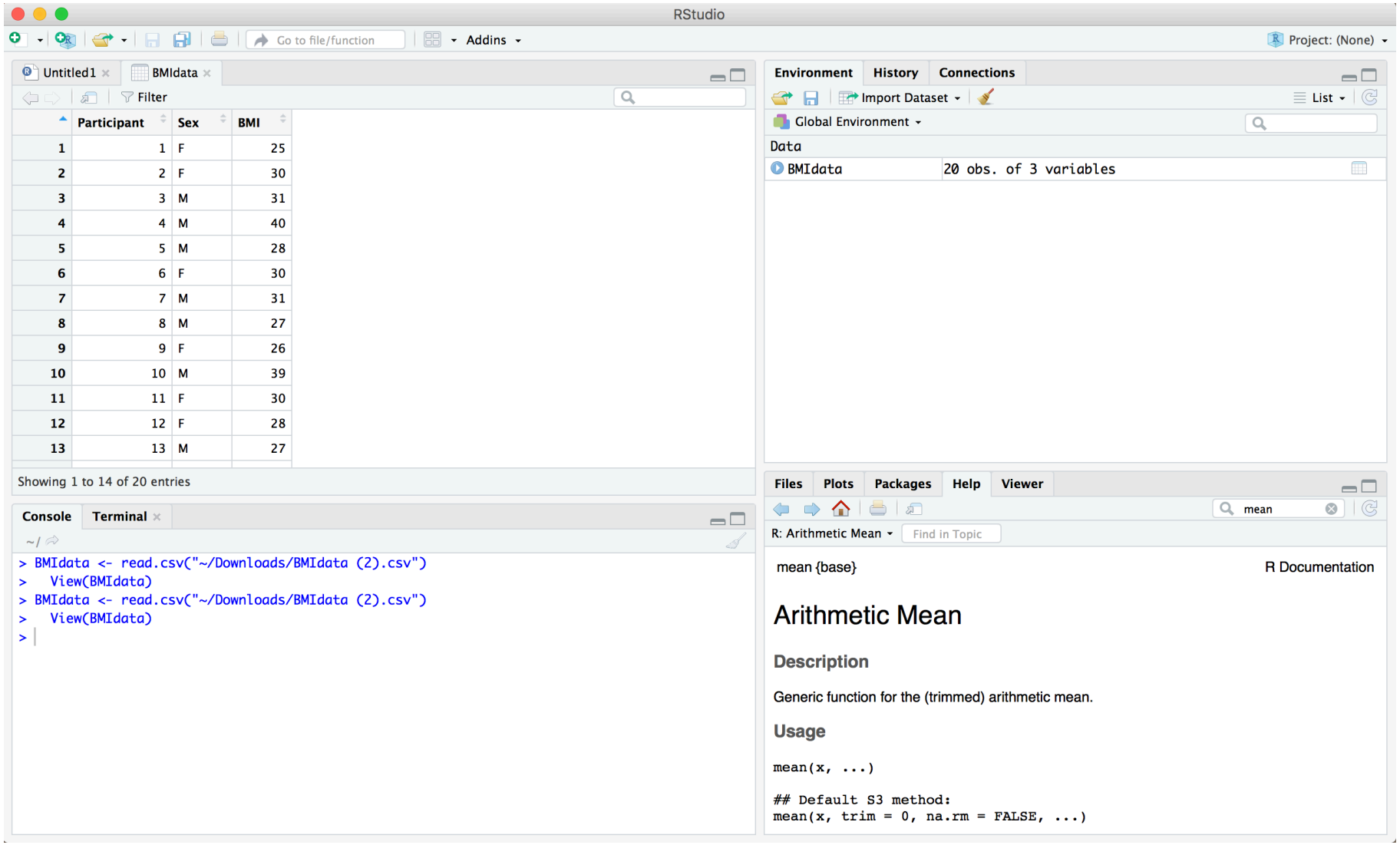
- Once downloaded, import the dataset into R Studio. It is a .csv file so import it as a Text file. Find where the file is saved on your computer.




RStudio
Go to file/function
Addins
Project: (None)
Untitled1
Source on Save
Run
Source
1
1:1
(Top Level)
R Script
Environment
History
Connections
Import Dataset
List
Global
From Text (base)...
From Text (readr)...
From Excel...
From SPSS...
From SAS...
From Stata...
1
Console
Terminal
~/
> BMIdata <- read.csv("~/Downloads/BMIdata (2).csv")
> View(BMIdata)
>
Files
Plots
Packages
Help
Viewer
mean
R: Arithmetic Mean
Find in Topic
mean {base}
R Documentation
Arithmetic Mean
Description
Generic function for the (trimmed) arithmetic mean.
Usage
mean(x, ...)
Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)


Global Environment
Import Dataset
Name
BMIdata
Encoding
Automatic
Heading
Yes
No
Row names
Automatic
Separator
Comma
Decimal
Period
Quote
Double quote (")
Comment
None
na.strings
NA
Strings as factors
Input File
Participant,Sex,BMI
1,F,25
2,F,30
3,M,31
4,M,40
5,M,28
6,F,30
7,M,31
8,M,27
9,F,26
10,M,39
11,F,30
12,F,28
Data Frame
Participant Sex BMI
1 F 25
2 F 30
3 M 31
4 M 40
5 M 28
6 F 30
7 M 31
8 M 27
9 F 26
10 M 39
11 F 30
12 F 28
Import
Cancel
2

3


RStudio
Go to file/function
Addins
Project: (None)
Untitled1
BMIdata
Filter
Participant Sex BMI
1 1 F 25
2 2 F 30
3 3 M 31
4 4 M 40
5 5 M 28
6 6 F 30
7 7 M 31
8 8 M 27
9 9 F 26
10 10 M 39
11 11 F 30
12 12 F 28
13 13 M 27
Showing 1 to 14 of 20 entries
Environment History Connections
Import Dataset
List
Global Environment
Data
BMIdata 20 obs. of 3 variables
Files Plots Packages Help Viewer
mean
R: Arithmetic Mean
Find in Topic
mean {base} R Documentation
Arithmetic Mean
Description
Generic function for the (trimmed) arithmetic mean.
Usage
mean(x, ...)
Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)
Console Terminal
~/
> BMIdata <- read.csv("~/Downloads/BMIdata (2).csv")
> View(BMIdata)
> BMIdata <- read.csv("~/Downloads/BMIdata (2).csv")
> View(BMIdata)
>

Descriptive statistics

- This dataset presents the BMI calculations for 20 participants, 10 female and 10 male.
- Suppose you were interested in seeing if there was a statistically significant effect of sex on BMI. Since this is an independent, two-group (male vs. female) comparison, we can approach the analysis with a t-test.
- Generally speaking, analyses are presented following the calculation of some simple descriptive statistics of the two groups compared.

```r
subsetF<-subset(BMIdata,Sex=="F")
subsetM<-subset(BMIdata,Sex=="M")

meanF<-mean(subsetF$BMI)
meanM<-mean(subsetM$BMI)

sdF<-sd(subsetF$BMI)
sdM<-sd(subsetM$BMI)

descript<-data.frame("Sex"=c("Female","Male"),
                     "Mean BMI"=c(meanF,meanM),
                     "Standard Deviation"=c(sdF,sdM))

descript
```

Creating a subset of the data for the two groups, male and female

Calculating the mean BMI for both male and female groups

Calculating the BMI standard deviation for both male and female groups

Putting together a dataframe with a column for mean and standard deviation by group

```
      Sex Mean.BMI Standard.Deviation
1 Female     28.6           5.168279
2   Male     30.8           5.028806
```

t-test in R

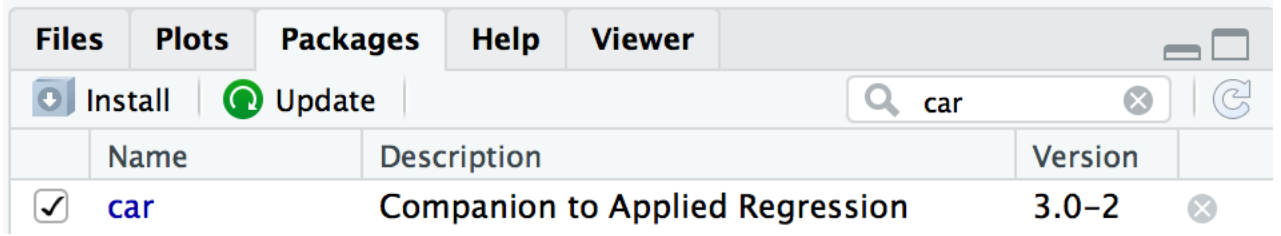
- One of the key assumptions of the independent groups t-test is homogeneity of variance.
- Many statisticians recommend that you always conduct a Welch's t-test (i.e., a t-test that does not assume equal variances), but you can easily test homogeneity of variance.
  - Let's test for homogeneity of variance using Levene's test
    - Recap: if Levene's test is significant, variances are heterogeneous

t-test in R

- The function ”leveneTest” in the “car” package makes testing assumptions easy and quick! First, download and run the ”car” package by running the following commands:

Step 1) install the package `install.packages(“car”)`

Step 2) run the package `library(car)`



t-test in R

- Next, let's run levene's test:

"BMI by sex", or "y by x"

```
> leveneTest(BMI~Sex, data=BMIdata)
Levene's Test for Homogeneity of Variance (center = median)
      Df F value Pr(>F)
group  1  0.0198 0.8897
      18
```

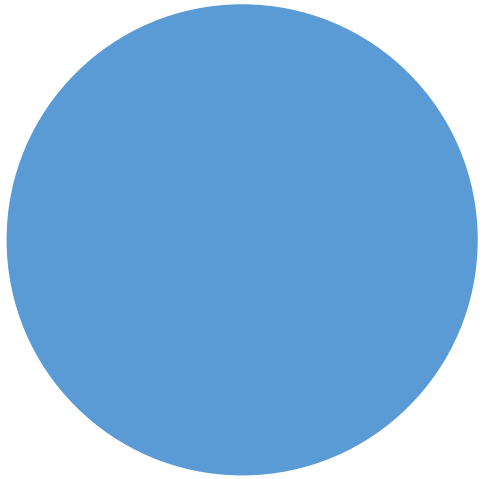
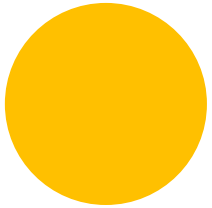
Homogeneity of variance assumption has not been violated ($p>0.05$), so we can proceed with a t-test that assumes equal variances.

t-test in R

```
> t.test(BMI~Sex, data=BMIdata, var.equal=TRUE, conf.level=0.95)

        Two Sample t-test

data:  BMI by Sex
t = -0.96476, df = 18, p-value = 0.3474
alternative hypothesis: true difference in means
95 percent confidence interval:
 -6.990839  2.590839
sample estimates:
mean in group F mean in group M
           28.6            30.8
```

The ”t.test” function defaults to unequal variances, which would provide Welch’s t-test instead of this. The “var.equal=TRUE” argument tells the function that the variances are equal. We’ve also specified a confidence level of 95%, which is generally acceptable.

The t-test is not significant, t(18)=-0.96, p>0.05, suggesting there is no significant difference between groups.

Wow that was awesome!

Where do I go next?
How can I learn more?

R Resources

- Quick R's introductory R tutorials: https://www.statmethods.net/r-tutorial/index.html
  - Great for a quick search
- Swirl package in R: https://swirlstats.com/
  - Learn R in R – prompted learning
- Data Camp's introduction to R: https://www.datacamp.com/courses/free-introduction-to-r
  - High production value and the introductory course is free!
- R Bootcamp: https://www.jaredknowles.com/r-bootcamp/
  - Basically an entire R course, complete with handouts and scripts
- Test construction modules: https://www.uwo.ca/fhs/tc/
  - Within each lesson there are R handouts to go about different test construction analyses in R

Thank you!

References

- Johnson, A. (2015). Independent Groups t-test. Retrieved from https://owl.uwo.ca/access/lessonbuilder/item/39536706/group/69b01c0c-1af0-4924-8261-529274ddf03c/R%20Handouts/independent-t.pdf
- Roecker, S., Yoast, K., Wills, S., & D'Avello, T. (2018). Chapter 1 Introduction to R and Rstudio. Retrieved from: http://ncss-tech.github.io/stats_for_soil_survey/chapters/1_introduction/1_introduction.html
- Smyth, R. & Johnson, A. (2015). Introduction to R. Retrieved from https://www.uwo.ca/fhs/tc/labs/01.Introduction.pdf