



Introduction to R- Basics

By Sarah Ingabire



What is R?

- R was created by Ross Ihaka and Robert Gentleman in the 1990s.
- R is a comprehensive programming language used for statistical analysis and graphics
- R is considered as a dialect of the S language which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) around 1976.
 - Key limitation of S: only available in a commercial package, S-PLUS
- R is available in several forms:
 - The sources (written mainly in C and some routines in Fortran) for Unix and Linux machines
 - Some pre-compiled binaries for Windows, Linux and Macintosh



Why R?

- Free and open source.
- Most widely used software environments in data science
- an effective data handling and storage facility,
- R has many functions that provide a large variety of statistical analyses and graphical facilities for data analysis
- Functions can be visualized immediately in their own window and be saved in various formats (jpg, png, pdf, etc...)
- In R, different statistical functions can be combined in a single program to perform more complex analyses



How to install R?

Three Basic Steps:

1. Install R

Program that runs all
your code

2. Install R studio

Program that allows to control R
in a friendly way (text editor +
console)

3. Install R packages

1. Installing R



[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Search](#)
[CRAN Team](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Task Views](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)

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 \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [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 (2022-06-23, Funny-Looking Kid) [R-4.2.1.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

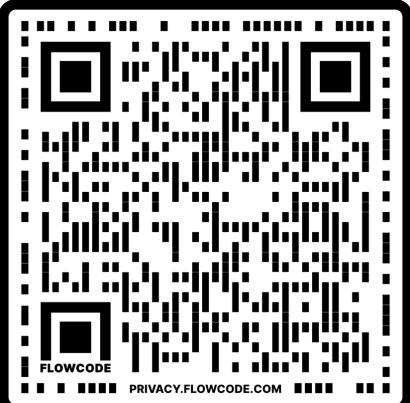
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

A. Installing R for MacoS

- Open the following website or QR code:

<https://cran.r-project.org/bin/macosx/>

- Click on the latest version of the .pkg file
- Open the downloaded .pkg file and open R



R 4.2.1 "Funny-Looking Kid" released on 2022/06/23

Please check the integrity of the downloaded package by checking the signature:

`pkgutil --check-signature R-4.2.1.pkg`

in the *Terminal* application. If Apple tools are not available you can check the SHA1 checksum of the downloaded image:

`openssl sha1 R-4.2.1.pkg`

Latest release:

R-4.2.1.pkg (notarized and signed)

SHA1-hash: f83a6c96cedd19193255f94cb01381a273073a3a
(ca. 90MB) for Intel Macs

R 4.2.1 binary for macOS 10.13 (**High Sierra**) and higher, **Intel 64-bit** build, signed and notarized package.

Contains R 4.2.1 framework, R.app GUI 1.79 in 64-bit for Intel Macs, Tcl/Tk 8.6.6 X11 libraries and Texinfo 6.7. The latter two components are optional and can be omitted 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 (version 2.7.11 or later) since it is no longer part of macOS. Always re-install XQuartz when upgrading your macOS to a new major version.

This release supports Intel Macs, but it is also known to work using Rosetta2 on M1-based Macs. For native Apple silicon arm64 binary see below.

Important: this release uses Xcode 12.4 and GNU Fortran 8.2. If you wish to compile R packages from sources, you may need to download GNU Fortran 8.2 - see the [tools](#) directory.

R-4.2.1-arm64.pkg (notarized and signed)

SHA1-hash: 05370dd000f0fded68594fc95334808ee25a8e91
(ca. 89MB) for M1 Macs only!

R 4.2.1 binary for macOS 11 (**Big Sur**) and higher, **Apple silicon arm64** build, signed and notarized package.

Contains R 4.2.1 framework, R.app GUI 1.79 for Apple silicon Macs (M1 and higher), Tcl/Tk 8.6.12 X11 libraries and Texinfo 6.8. **Important:** this version does NOT work on older Intel-based Macs.

Note: the use of X11 (including `tcltk`) requires [XQuartz](#) (version 2.8.1 or later). Always re-install XQuartz when upgrading your macOS to a new major version.

This release uses Xcode 13.1 and experimental GNU Fortran 12 arm64 fork. If you wish to compile R packages which contain Fortran code, you may need to download GNU Fortran for arm64 from <https://mac.R-project.org/tools>. Any external libraries and tools are expected to live in `/opt/R/arm64` to not conflict with Intel-based software and this build will not use `/usr/local` to avoid such conflicts (see the [tools page](#) for more details).

[NEWS](#) (for Mac GUI)

News features and changes in the R.app Mac GUI

[Mac-GUI-1.78.tar.gz](#)

SHA1-hash: 23b3c41b7eb771640fd504a75e5782792dddb2bc

Sources for the R.app GUI 1.78 for macOS. This file is only needed if you want to join the development of the GUI (see also [Mac-GUI repository](#)), it is not intended for regular users. Read the INSTALL file for further instructions.

Note: Previous R versions for El Capitan can be found in the [el-capitan/base](#) directory.

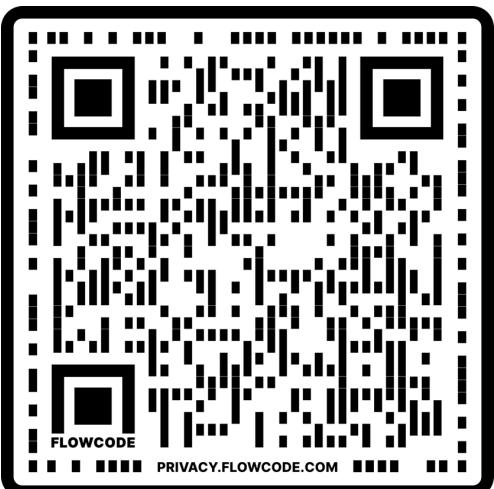
R-4.2.1 for Windows

B. Installing R for Windows

- Open the following website:

<https://cran.r-project.org/bin/windows/base/>

- Click on the “Download R -4.2.1 for Windows”



[Download R-4.2.1 for Windows](#) (79 megabytes, 64 bit) ←

[README on the Windows binary distribution](#)

[New features in this version](#)

This build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from [here](#).

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server.

Frequently asked questions

- [Does R run under my version of Windows?](#)
- [How do I update packages in my previous version of R?](#)

Please see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information.

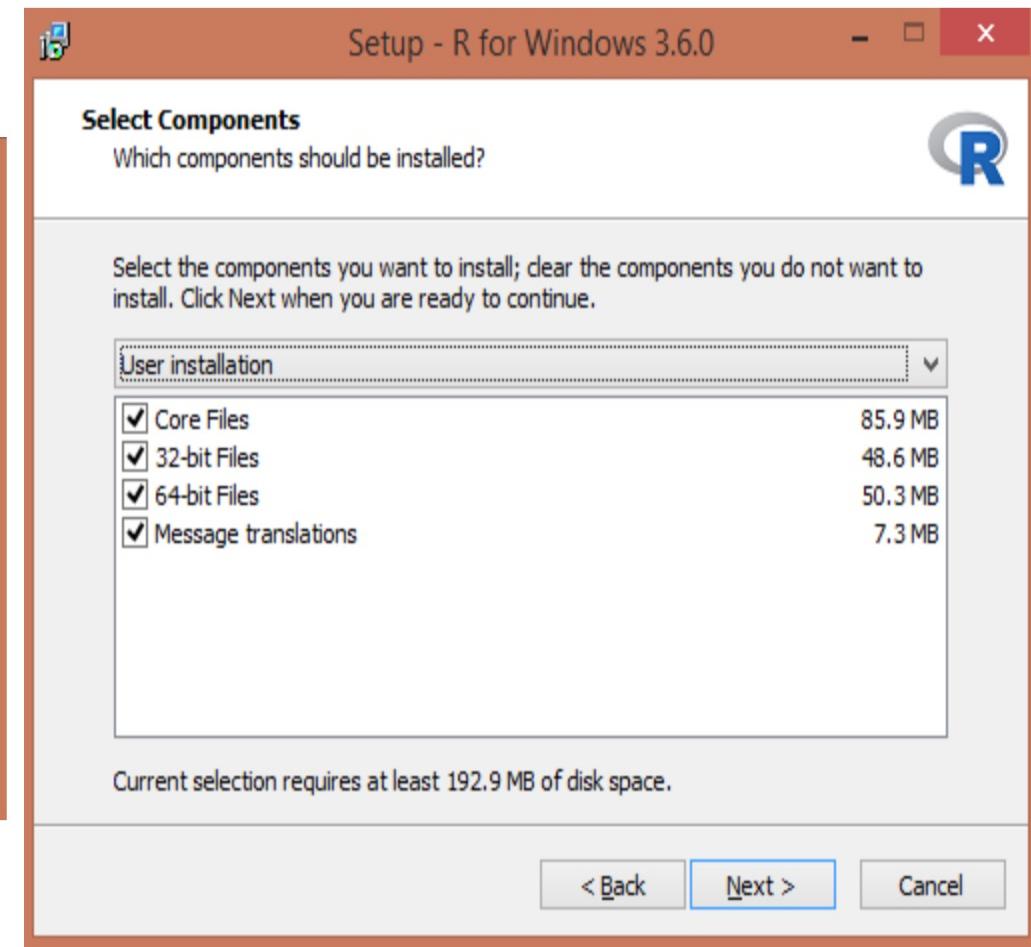
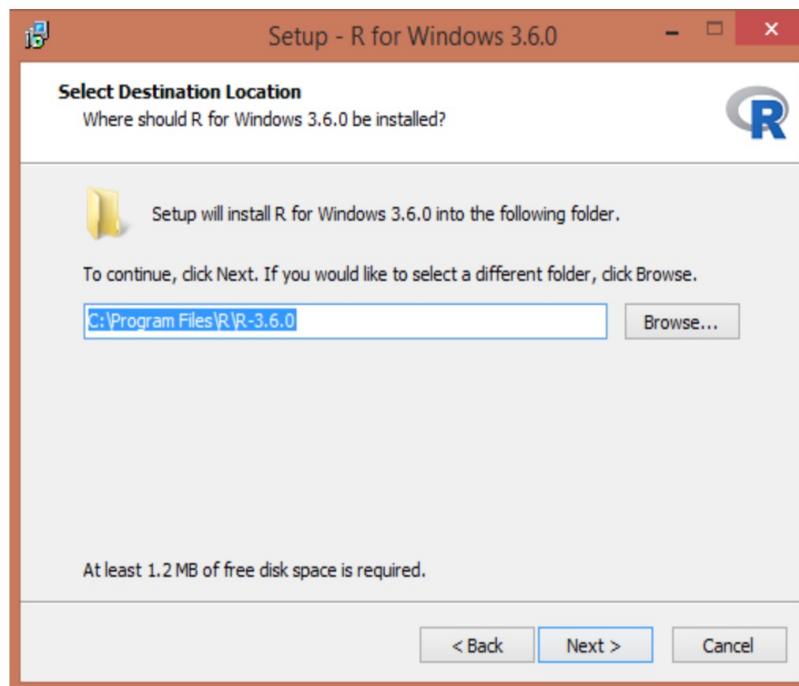
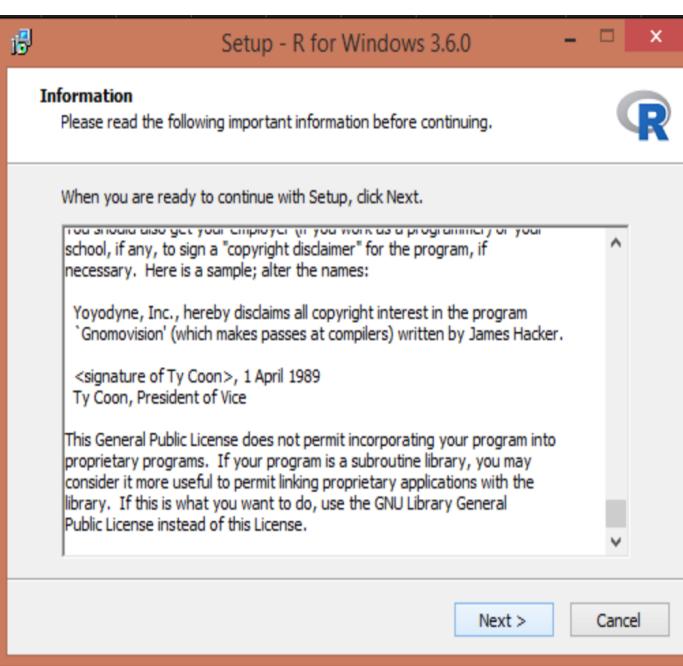
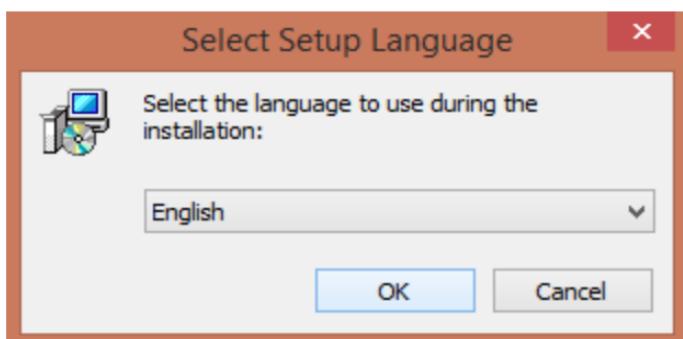
Other builds

- Patches to this release are incorporated in the [r-patched snapshot build](#).
- A build of the development version (which will eventually become the next major release of R) is available in the [r-devel snapshot build](#).
- [Previous releases](#)

Note to webmasters: A stable link which will redirect to the current Windows binary release is
<CRAN MIRROR>/bin/windows/base/release.html.

Last change: 2022-06-23

C. Installing R for Windows



Setup - R for Windows 3.6.0

Startup options

Do you want to customize the startup options?



Please specify yes or no, then click Next.

- Yes (customized startup)
 No (accept defaults)

< Back **Next >** Cancel

C. Installing R for Windows

Setup - R for Windows 3.6.0

Select Start Menu Folder

Where should Setup place the program's shortcuts?



Setup will create the program's shortcuts in the following Start Menu folder.

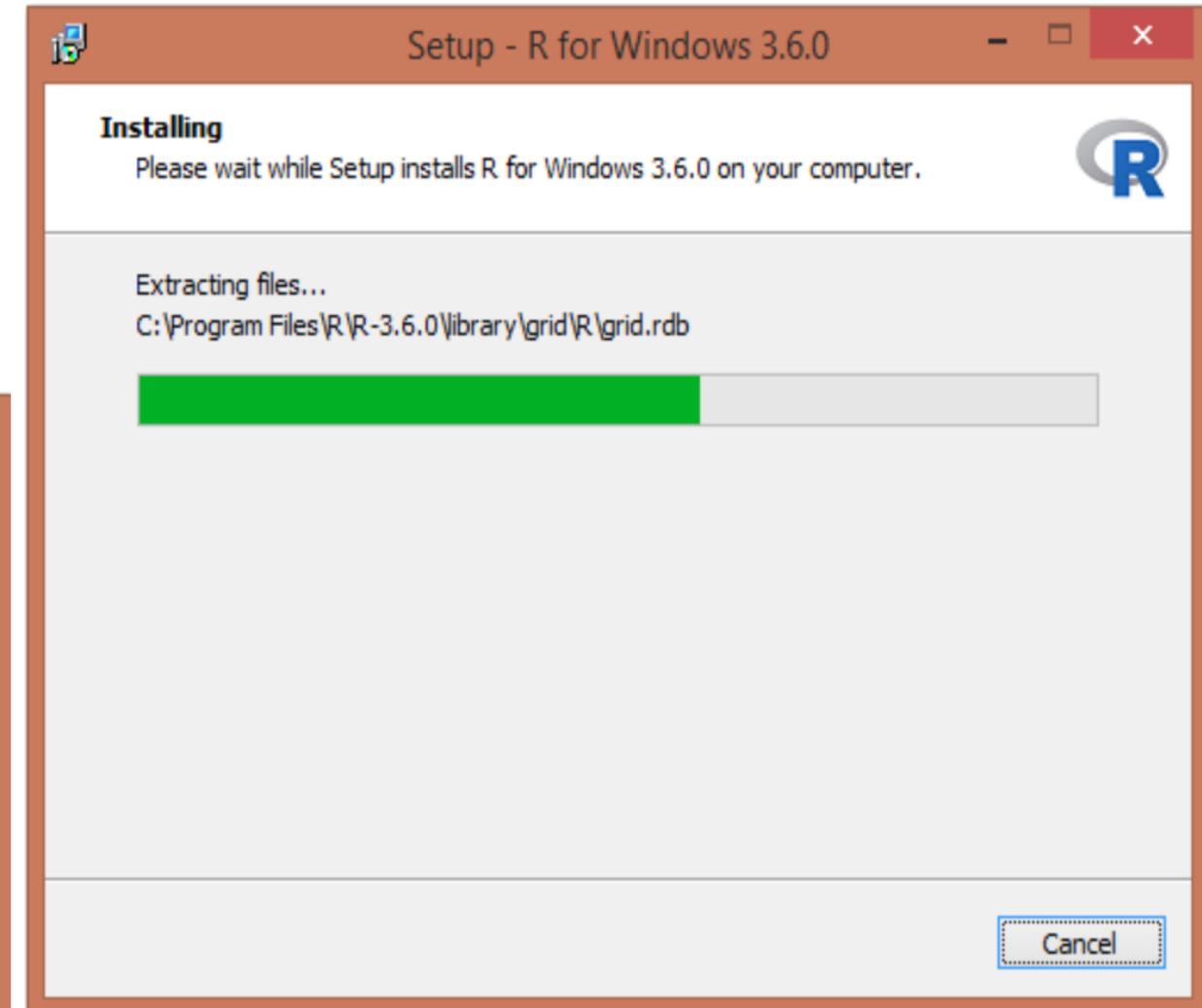
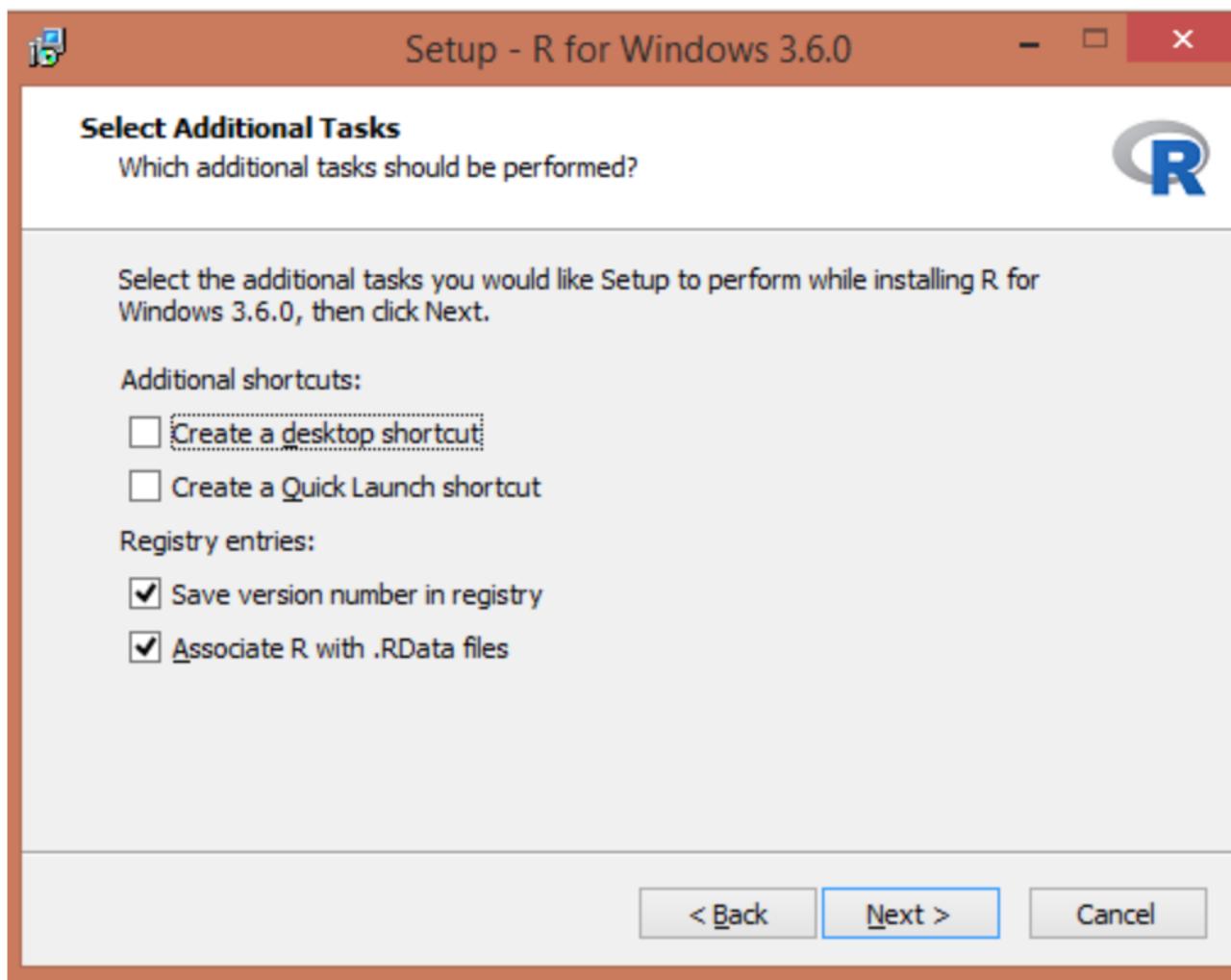
To continue, click Next. If you would like to select a different folder, click Browse.

R

Don't create a Start Menu folder

< Back **Next >** Cancel

C. Installing R for Windows

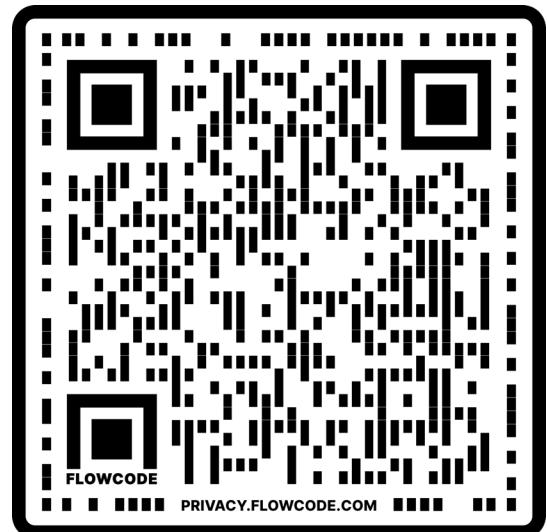


2. Installing R studio

Installing R studio

→ Open the following website:

<https://www.rstudio.com/products/rstudio/download/>



RStudio Desktop	RStudio Desktop Pro	RStudio Server	RStudio Workbench <small>i</small>
Open Source License	Commercial License	Open Source License	Commercial License
Free	\$995 <small>/year</small>	Free	\$4,975 <small>/year</small> <small>(5 Named Users)</small>
DOWNLOAD <small>Learn more</small>	BUY <small>Learn more</small>	DOWNLOAD <small>Learn more</small>	BUY <small>Evaluation Learn more</small>
Integrated Tools for R	✓	✓	✓
Priority Support	✓		✓
Access via Web Browser		✓	✓
RStudio Professional Drivers	✓		✓
Connect to RStudio Workbench <small>i</small> remotely	✓		
Enterprise Security			✓
Project Sharing			✓
Manage Multiple R Sessions & Versions			✓
Admin Dashboard			✓

Installing R studio

RStudio Desktop 2022.07.2+576 - [Release Notes](#)

1. Install R. RStudio requires [R 3.3.0+](#).
2. Download RStudio Desktop. Recommended for your system:



Requires macOS 10.15+ (64-bit)



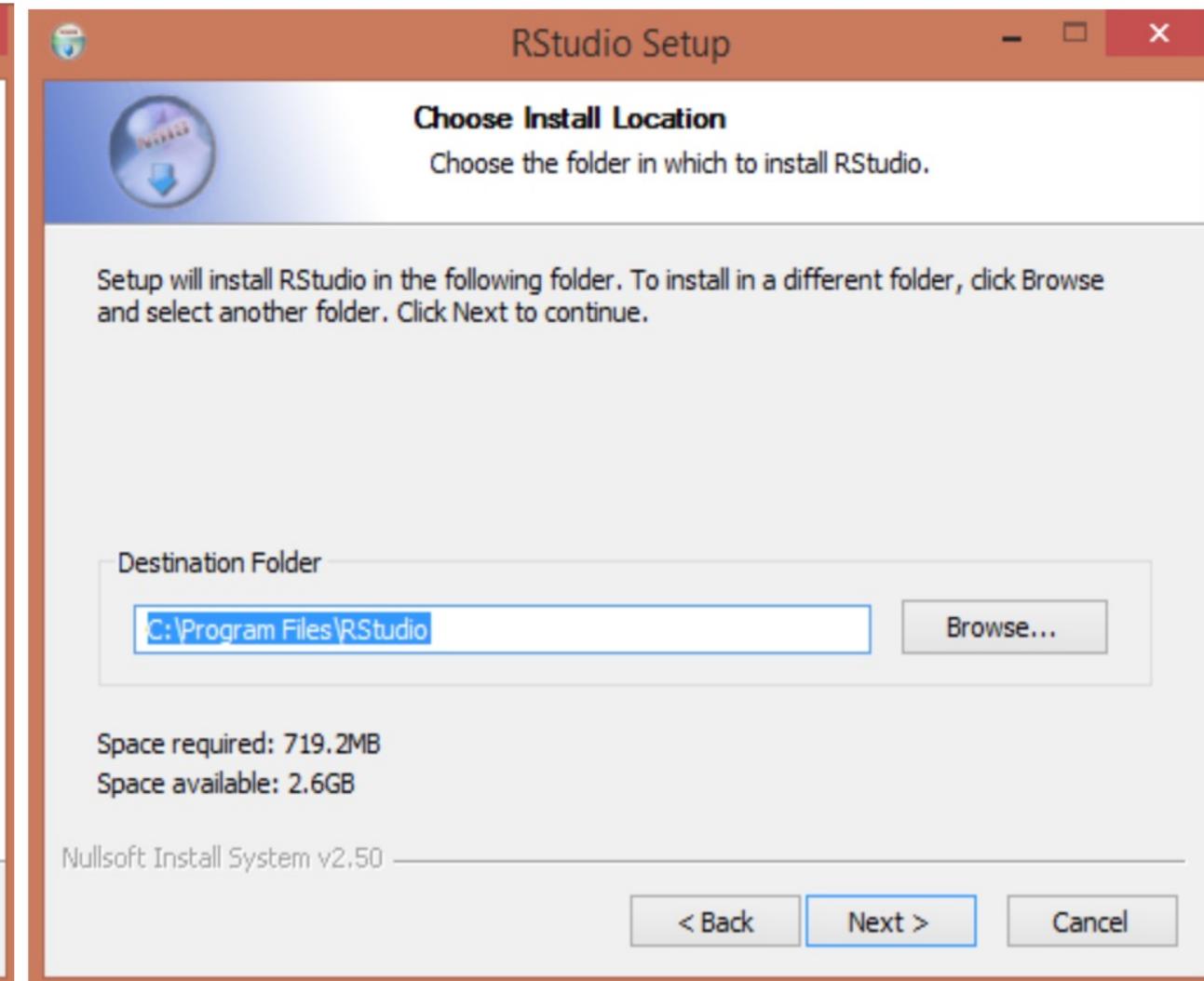
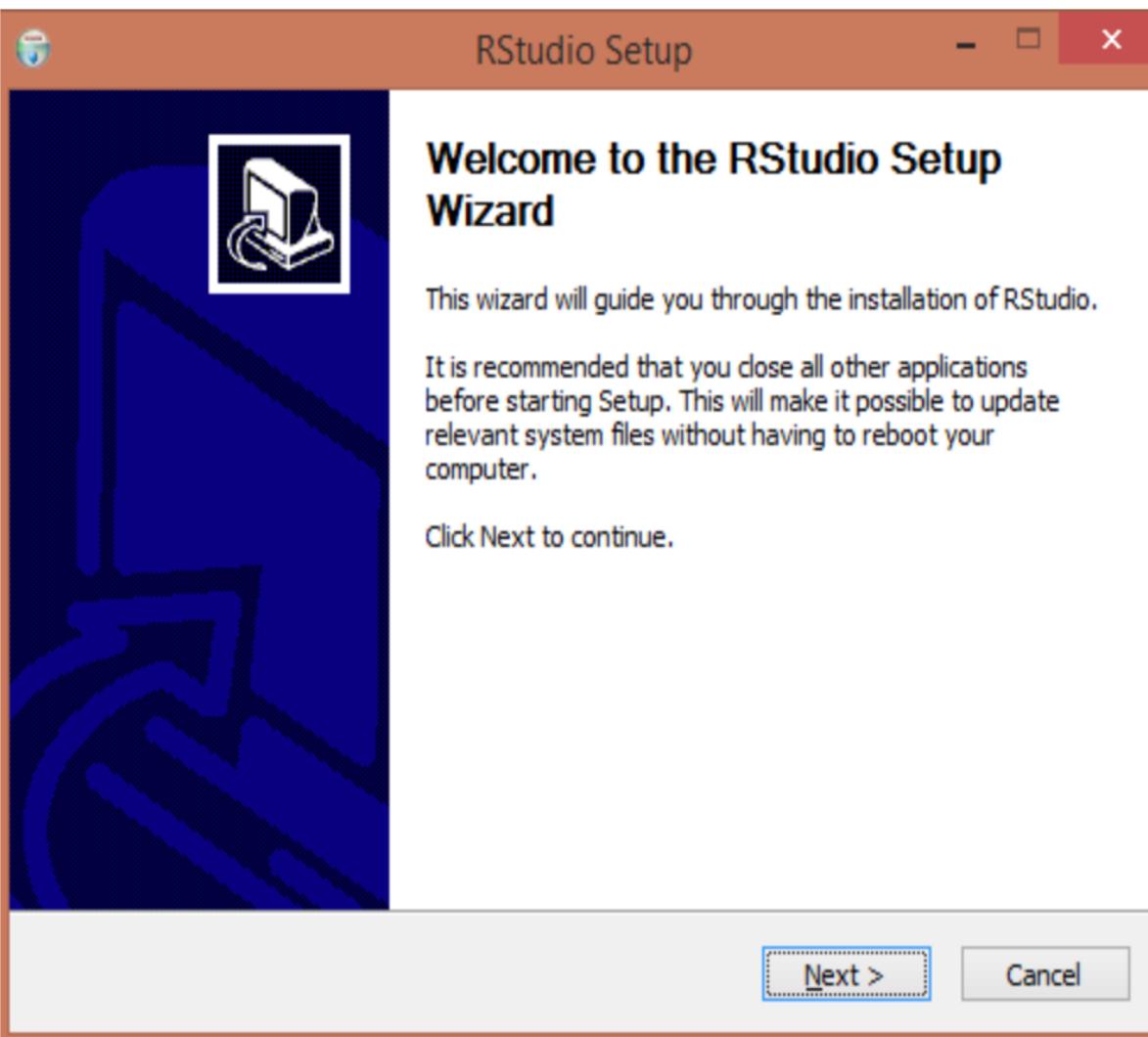
All Installers

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

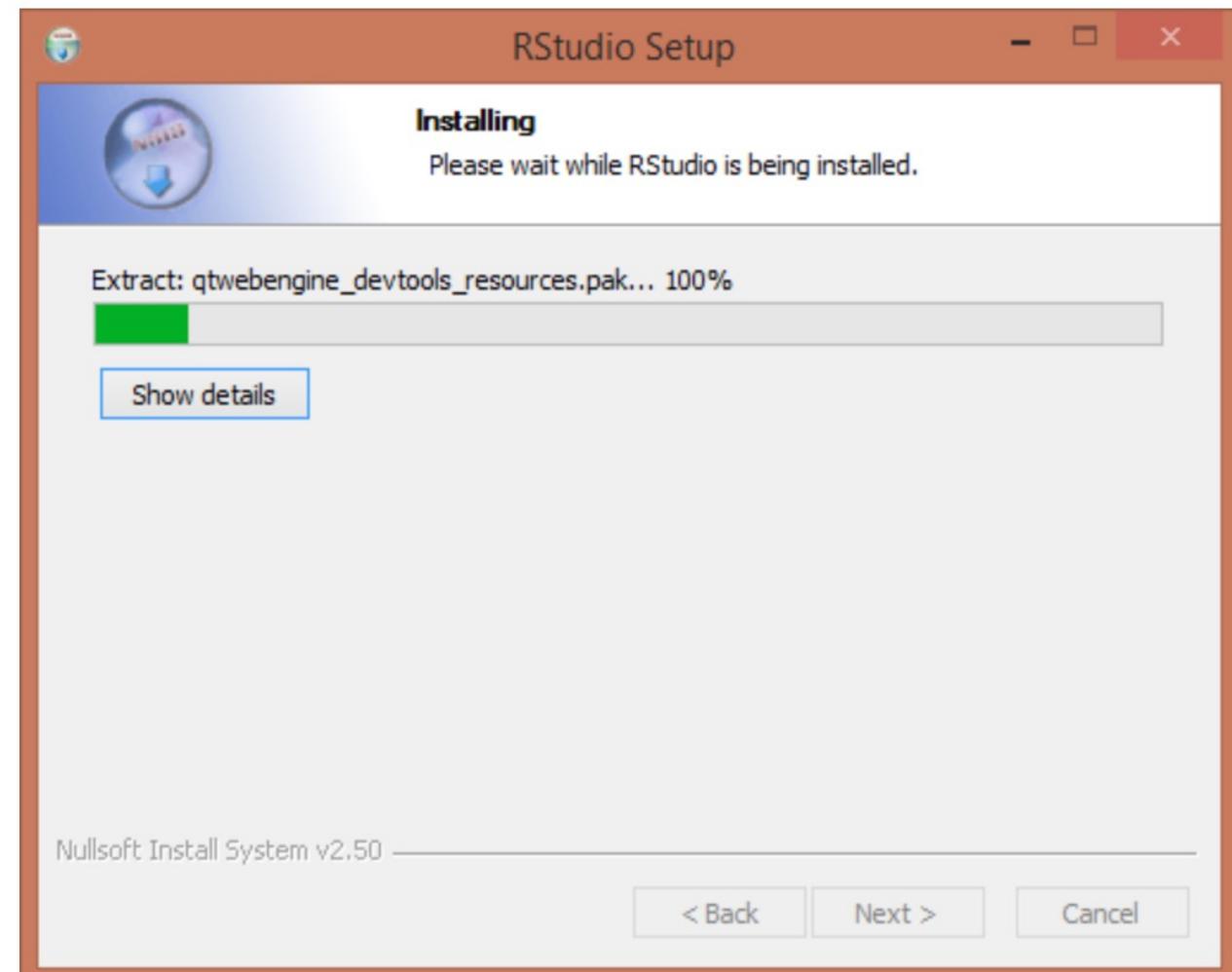
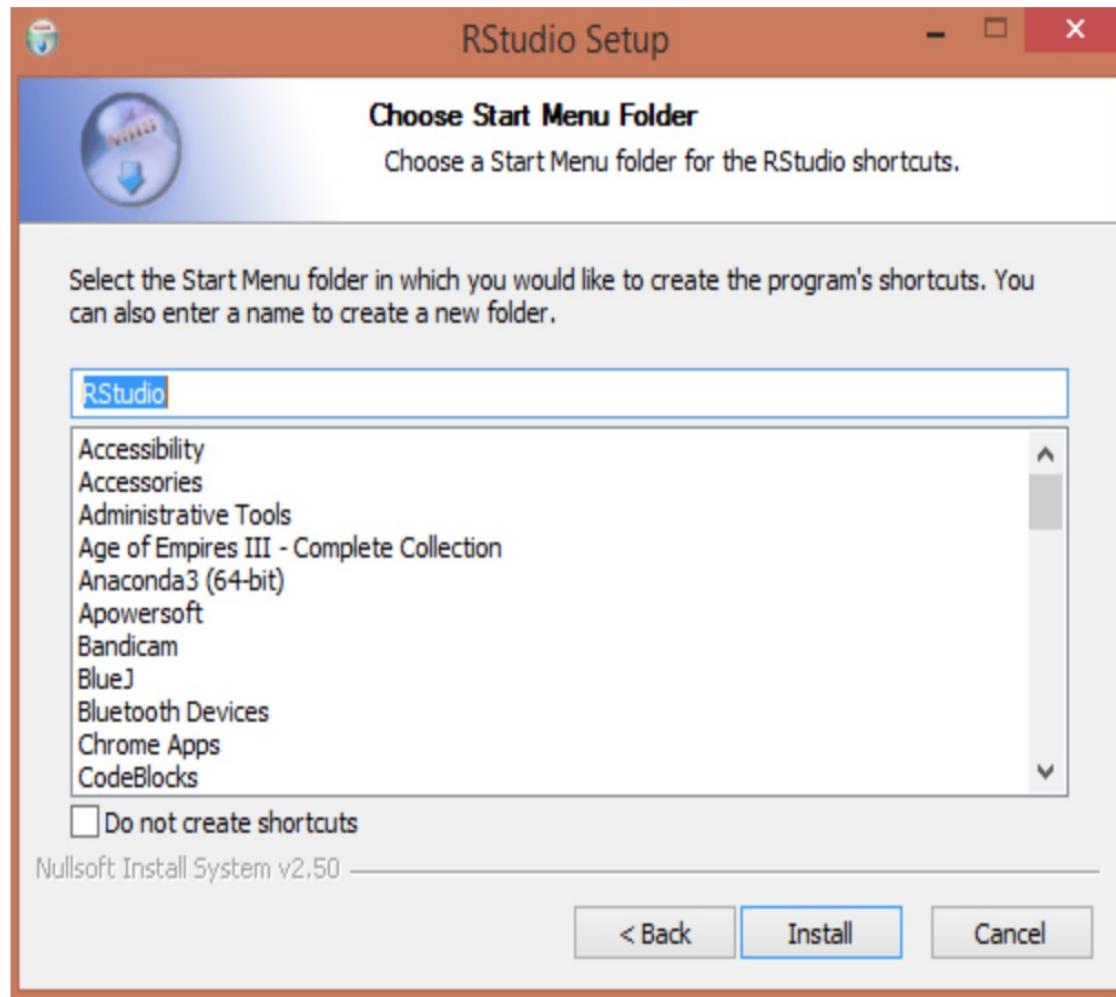
RStudio requires a 64-bit operating system. If you are on a 32 bit system, you can use an [older version of RStudio](#).

OS	Download	Size	SHA-256
Windows 10/11	 RStudio-2022.07.2-576.exe	190.49 MB	b38bf925
macOS 10.15+	 RStudio-2022.07.2-576.dmg	224.49 MB	35028d02
Ubuntu 18+/Debian 10+	 rstudio-2022.07.2-576-amd64.deb	133.19 MB	b7d0c386
Ubuntu 22	 rstudio-2022.07.2-576-amd64.deb	134.06 MB	e1c51003

Installing R studio for Windows

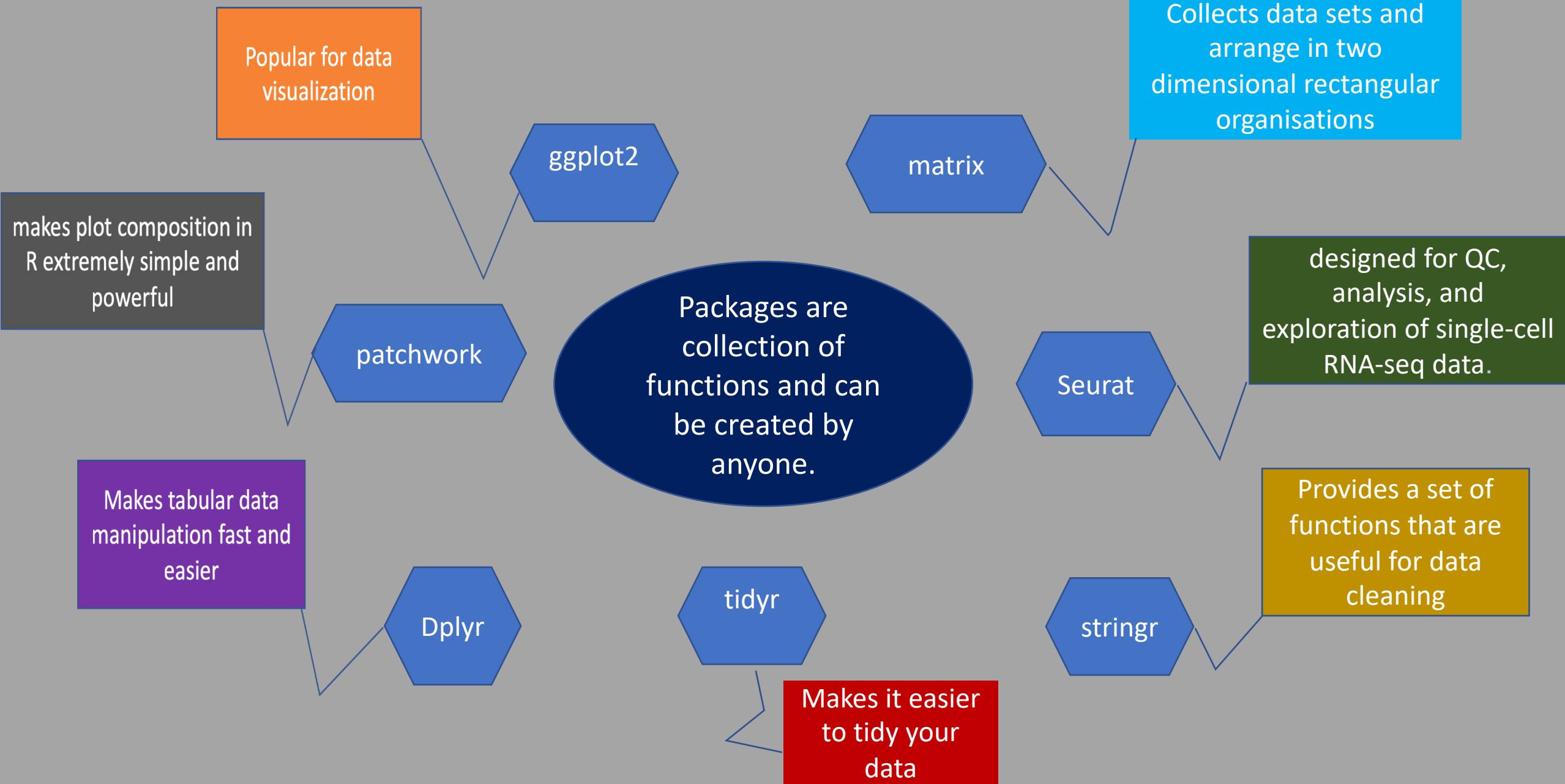


Installing R studio for Windows



3. Installing Packages in R

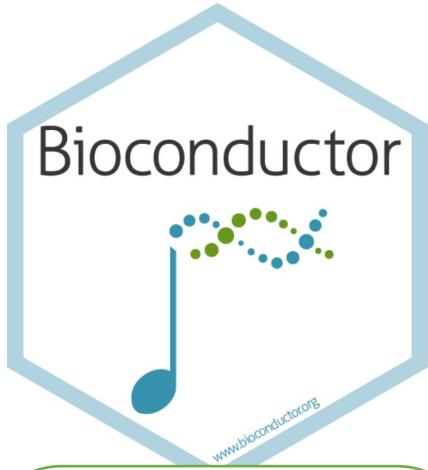
What are packages?



How to install packages in R?



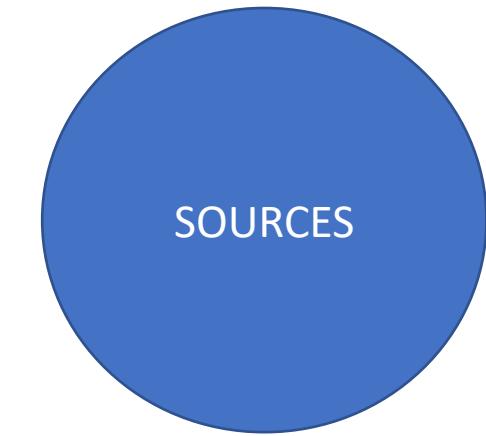
- The *Comprehensive R Archive Network* (CRAN) Packages is the main repository for software packages in R.
- Over 18,000 packages



- *Bioconductor* is an open software development for computational biology and bioinformatics.
- Over 1800 packages

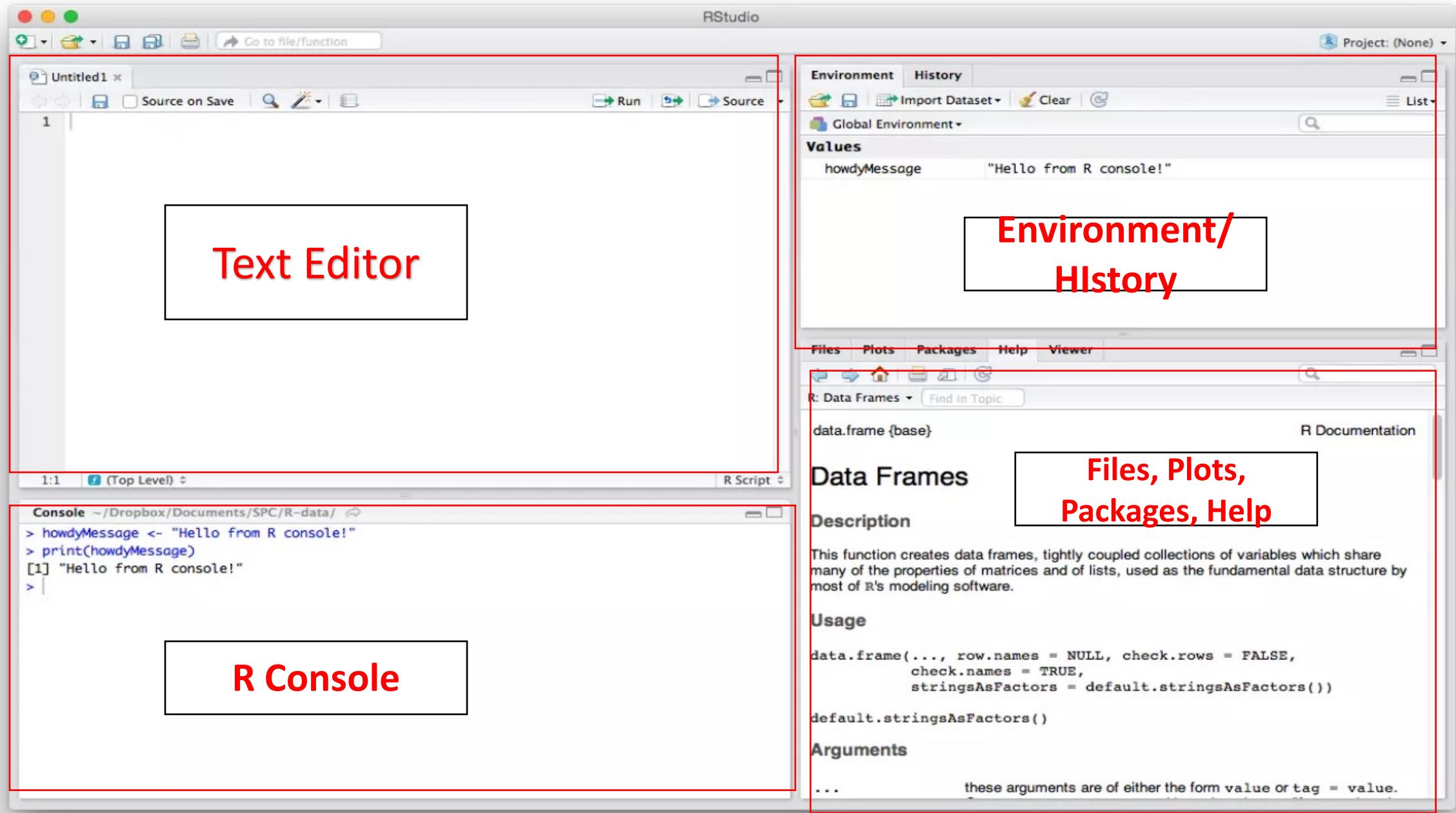


- R packages can be downloaded directly from the Github online software by using the “devtools” package



- Download the source package, open Terminal.app, navigate to the directory where you currently have the file, and then execute: usually
“packagename.tar.gz
> R CMD INSTALL
nameofpackage.tar.gz

Getting started with R studio



Creating and deleting objects from memory

- R directly execute commands without requiring to build a complete program
- The console can be used to test the code immediately
- To store the output of this command, we need to assign the value to a variable:

- The object is created using the assignment operators:

This can either be an arrow written as “`<-`” or the equal character “`=`”

Example: `> n <- 1 + 2`

- The object gets saved in the R global environment

Example: `> n <- 1 + 2`

```
> n
```

```
[1] 3
```

- To delete objects in memory, we use the function “`rm`”

Example: `> rm(n)`- deletes the object `n`

`> rm(m,n)`- deletes both `m` and `n`

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)

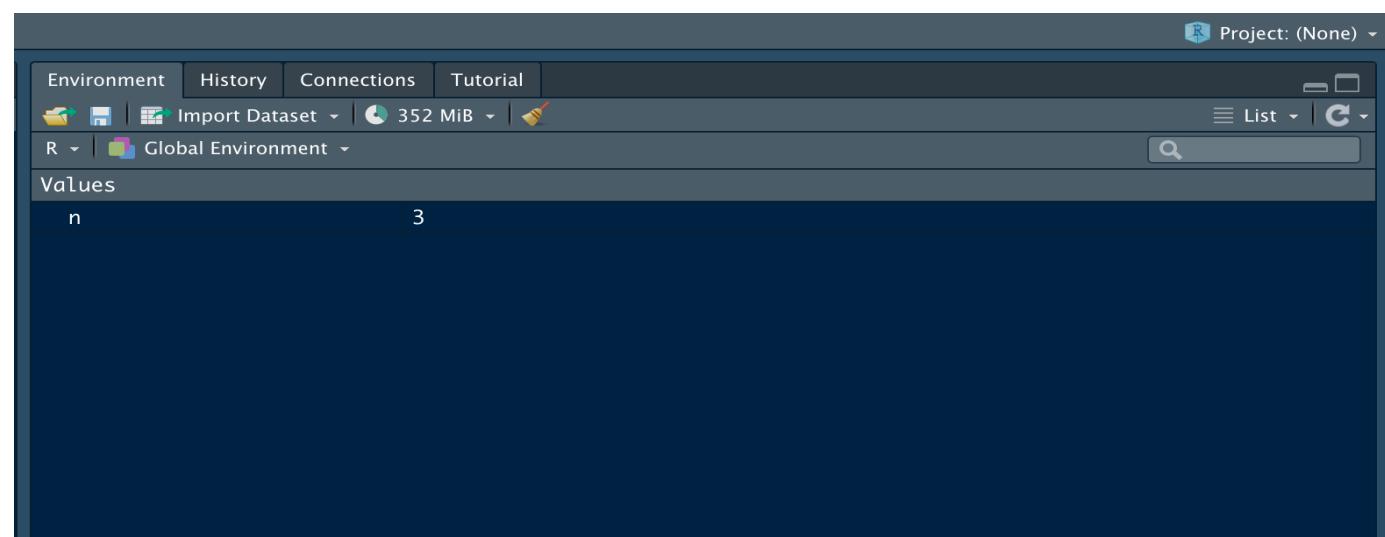
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

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
> |
```



A few things to keep in mind

- R is a case sensitive language. “N” and “n” are two different objects as well as “numbers” and “Numbers”.
- Space does not matter in R
- The object names cannot contain strange symbols such as: “+”, “-”, “!” or “#”. However, a dot “.” and an underscore “_” are allowed to use.
- The object names can contain a number but cannot start with a number.

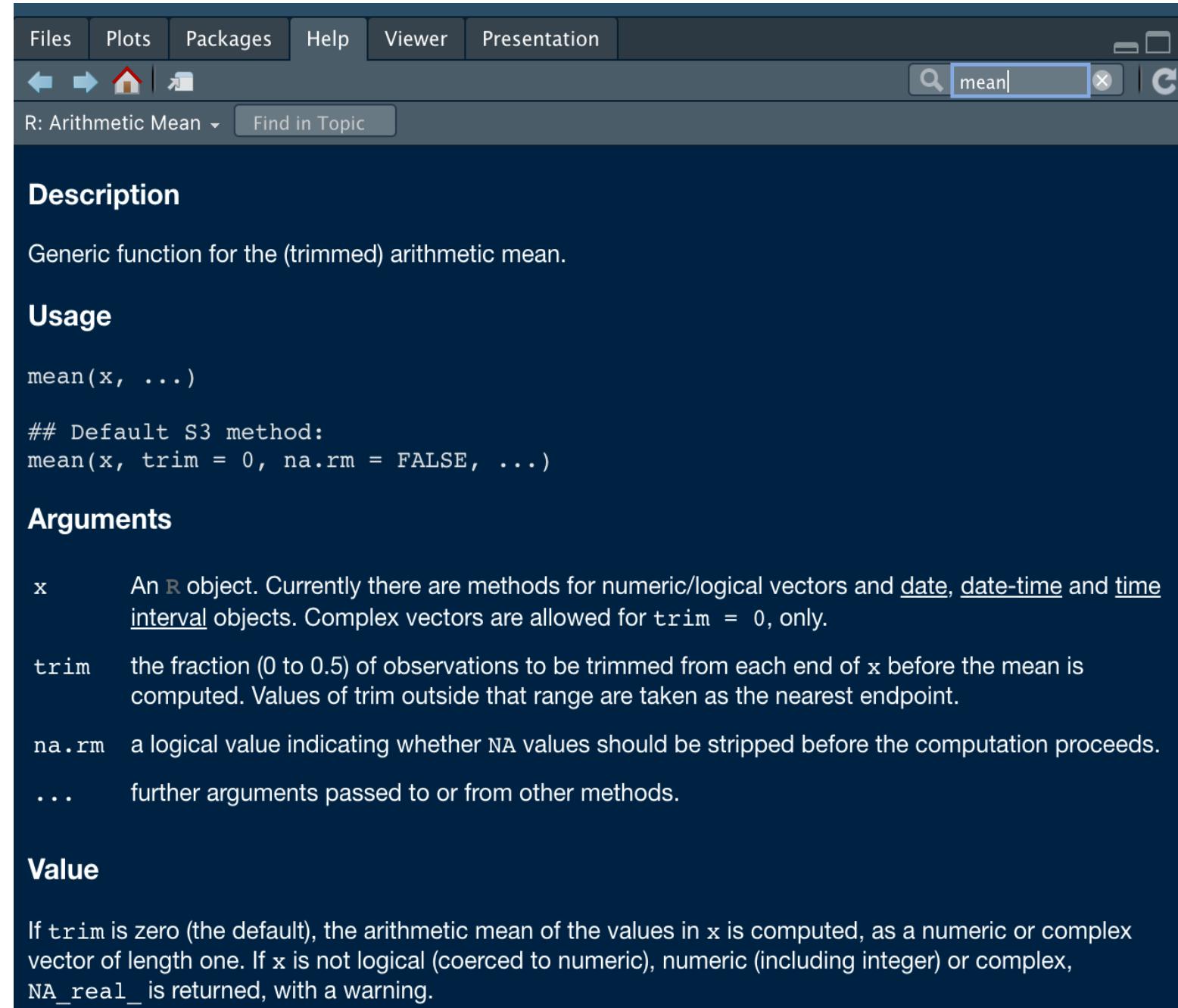
The R help

- R has a comprehensive built-in help system.
- The on-line help tool gives very useful information on how use the functions

Examples

- > ?mean
- > help ("mean")
- > help (mean)

Will display, the help page for the function mean() and how to use it.



The screenshot shows the R Help interface with the following details:

- Menu Bar:** Files, Plots, Packages, Help, Viewer, Presentation.
- Toolbar:** Back, Forward, Home, Stop, Refresh.
- Search Bar:** A search field containing "mean".
- Topic Bar:** R: Arithmetic Mean, Find in Topic.
- Description:** Generic function for the (trimmed) arithmetic mean.
- Usage:**

```
mean(x, ...)  
## Default S3 method:  
mean(x, trim = 0, na.rm = FALSE, ...)
```
- Arguments:**
 - x**: An R object. Currently there are methods for numeric/logical vectors and `date`, `date-time` and `time interval` objects. Complex vectors are allowed for `trim = 0`, only.
 - trim**: the fraction (0 to 0.5) of observations to be trimmed from each end of `x` before the mean is computed. Values of `trim` outside that range are taken as the nearest endpoint.
 - na.rm**: a logical value indicating whether `NA` values should be stripped before the computation proceeds.
 - ...**: further arguments passed to or from other methods.
- Value:**

If `trim` is zero (the default), the arithmetic mean of the values in `x` is computed, as a numeric or complex vector of length one. If `x` is not logical (coerced to numeric), numeric (including integer) or complex, `NA_real_` is returned, with a warning.

Reading Data into R

R is capable of reading data from most formats, including files created in other statistical packages.

Before loading or saving a dataset or object, you must set the R working directory to the location of the data.

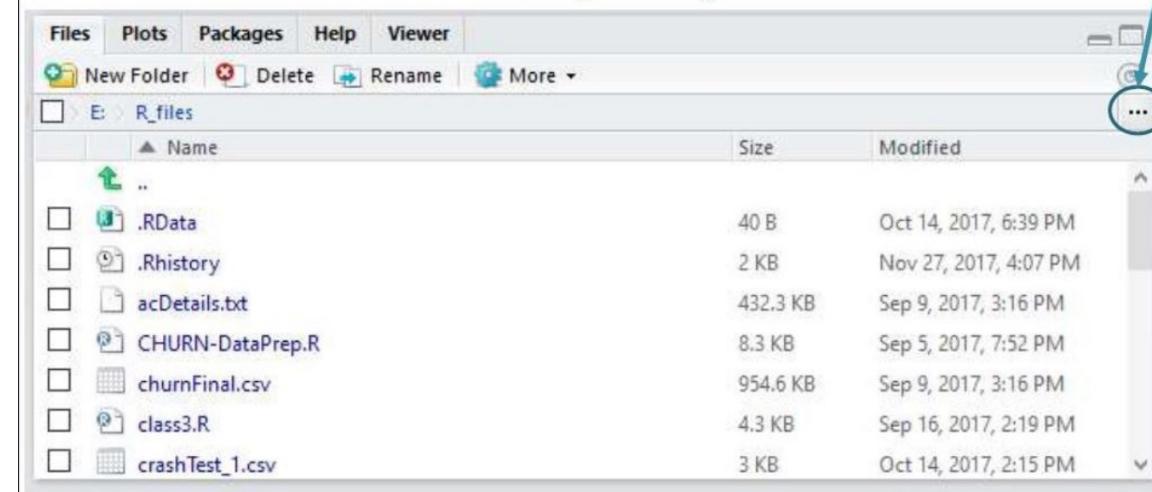
- Use command `getwd()` (get working directory) to find the directory

```
> getwd()
```

- Set working directory to directory name

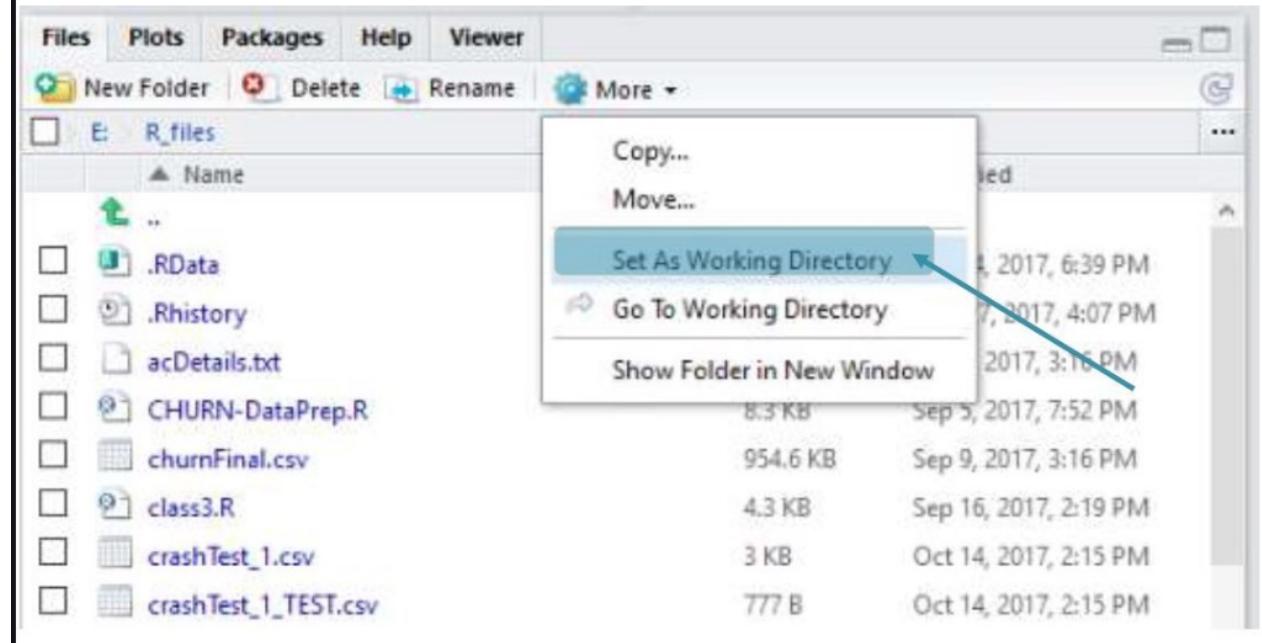
```
> setwd("C:\\\\Users\\\\s\\\\Desktop")
```

STEP 1: Choose a suitable location by clicking on the indicated icon



To choose working directory

STEP 2: Once directory is chosen, select the more icon and choose "Set as Working Directory"



Reading Data in a file

Let's suppose we want to load the Titanic dataset.

If the dataset is in

- an existing R package, load the package using the function “`data()`” → > **data(Titanic)**
- .RData format → **load(Titanic)**
- .txt or other text formats → **read.table("Titanic.txt")**
- .csv format → **read.csv("Titanic.txt")**
- For Stata format, load the foreign library using “`read.data()`” function → **read.data("Titanic")**

To save objects into these formats, we use
→ `write.table()`, `write.csv()`, etc. commands.

R objects

Data types

1. Numeric
2. Integers
3. Logical- Boolean
4. Characters

1. Numeric Data type

- The numeric data type is for numerical values.
e.g. 1, 2, 3, 4, -20, -21, 3.5 etc...
- We use the *as.numerical* function to convert the values of other data types into numerical values.

Code

Console Terminal × Background Jobs ×

```
R 4.1.1 · ~/Dropbox (MIT)/Sarah Ingabire's files/Ghana_workshop/ ↵
> num <- 1
> class(num)
[1] "numeric"
> is.numeric(num)
[1] TRUE
> |
```

Output

Environment History Connections Tutorial

Import Dataset 285 MiB

R Global Environment

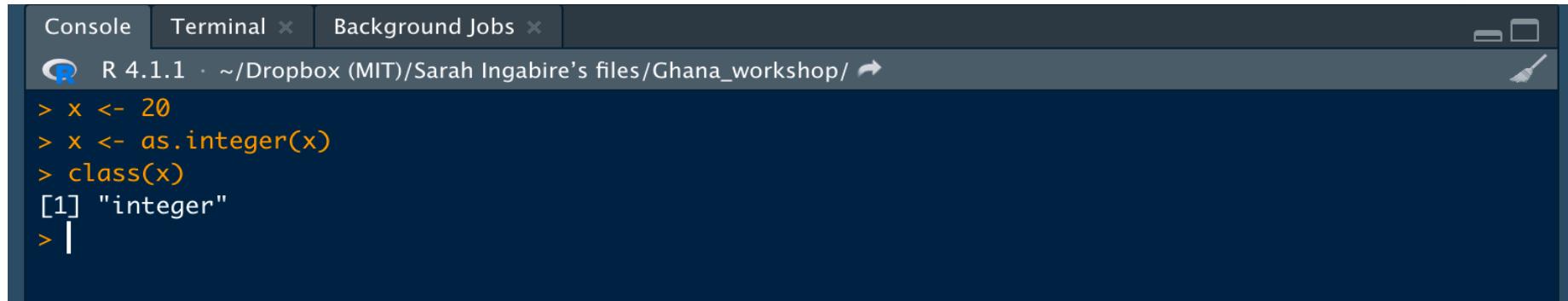
Name	Type	Length	Size	Value
num	numeric	1	56 B	1

2. Integers

- The integer data type is used for integer values.
- To store a value as an integer, we need to specify as such.

as.integer function is used to convert the values into an integer

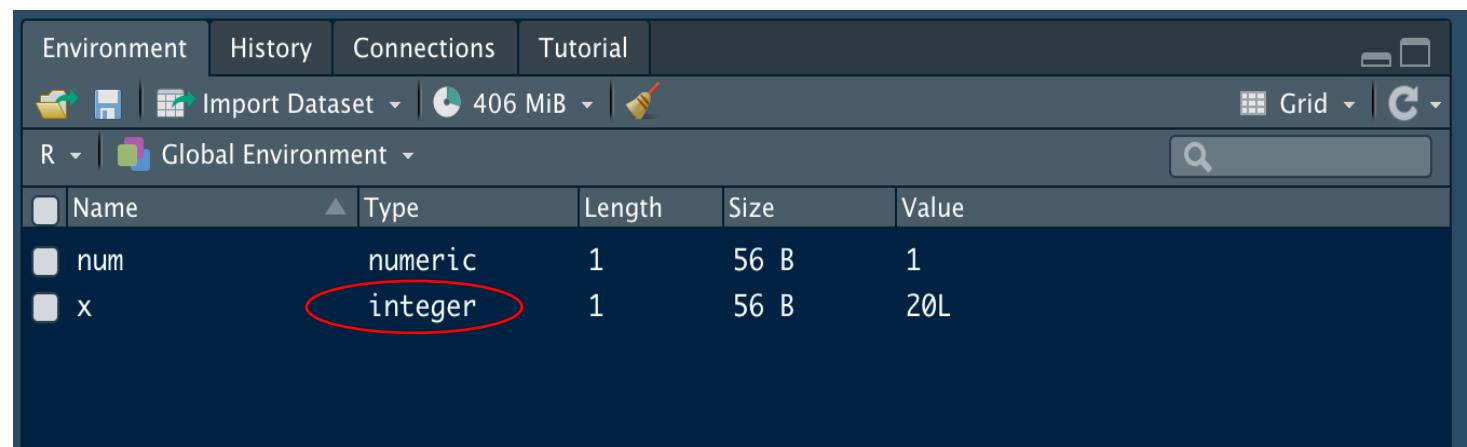
Code



R 4.1.1 · ~/Dropbox (MIT)/Sarah Ingabire's files/Ghana_workshop/ ↗

```
> x <- 20
> x <- as.integer(x)
> class(x)
[1] "integer"
> |
```

Output



Environment History Connections Tutorial

Import Dataset 406 MiB Grid C

R Global Environment

Name	Type	Length	Size	Value
num	numeric	1	56 B	1
x	integer	1	56 B	20L

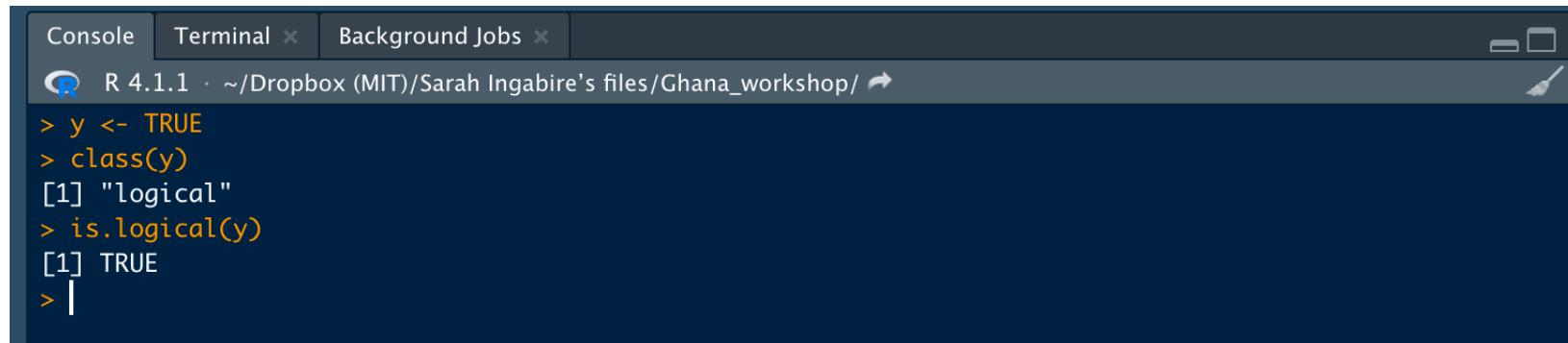
3. Logical Data Type

- The `logical` class stores boolean truth values,

i.e. TRUE and FALSE.

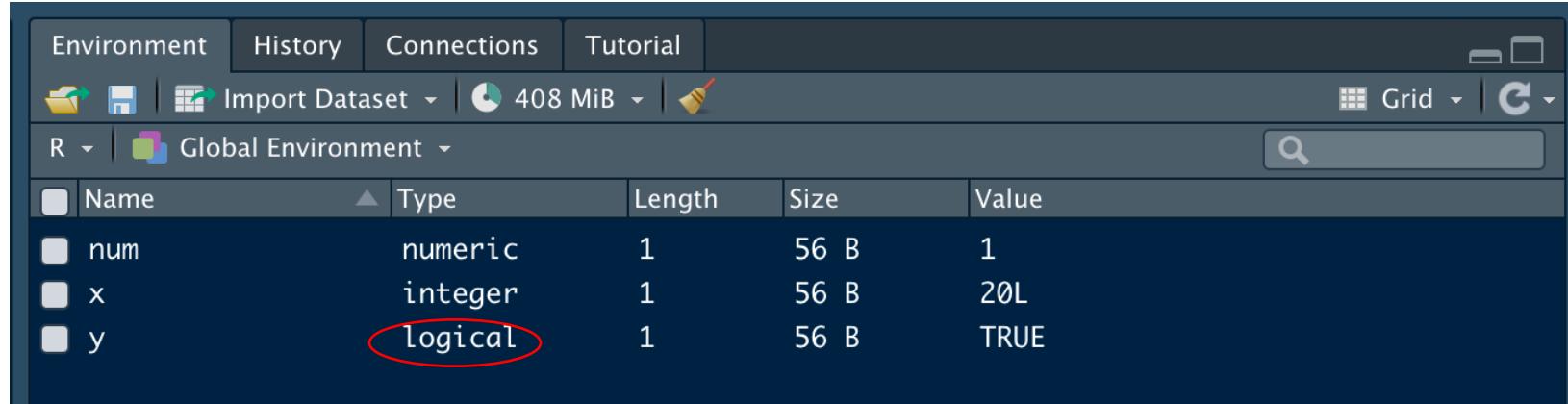
→ the *as.logical* function
function is used to convert the
values into an integer

Code



```
Console Terminal x Background Jobs x
R 4.1.1 · ~/Dropbox (MIT)/Sarah Ingabire's files/Ghana_workshop/ ↵
> y <- TRUE
> class(y)
[1] "logical"
> is.logical(y)
[1] TRUE
> |
```

Output



The screenshot shows the RStudio interface with the Environment tab selected. The Global Environment table lists variables:

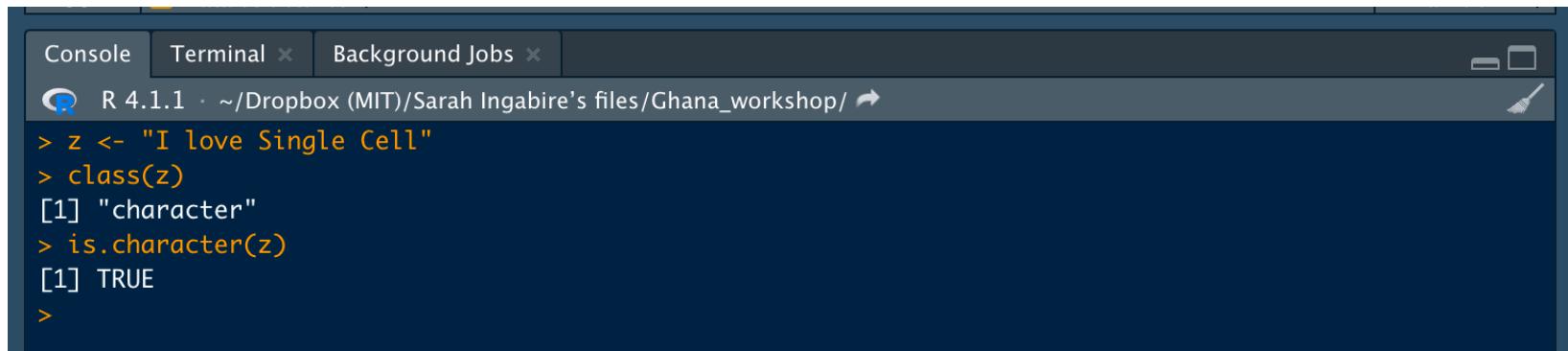
Name	Type	Length	Size	Value
num	numeric	1	56 B	1
x	integer	1	56 B	20L
y	logical	1	56 B	TRUE

The 'logical' entry for variable 'y' is circled in red.

4. Characters

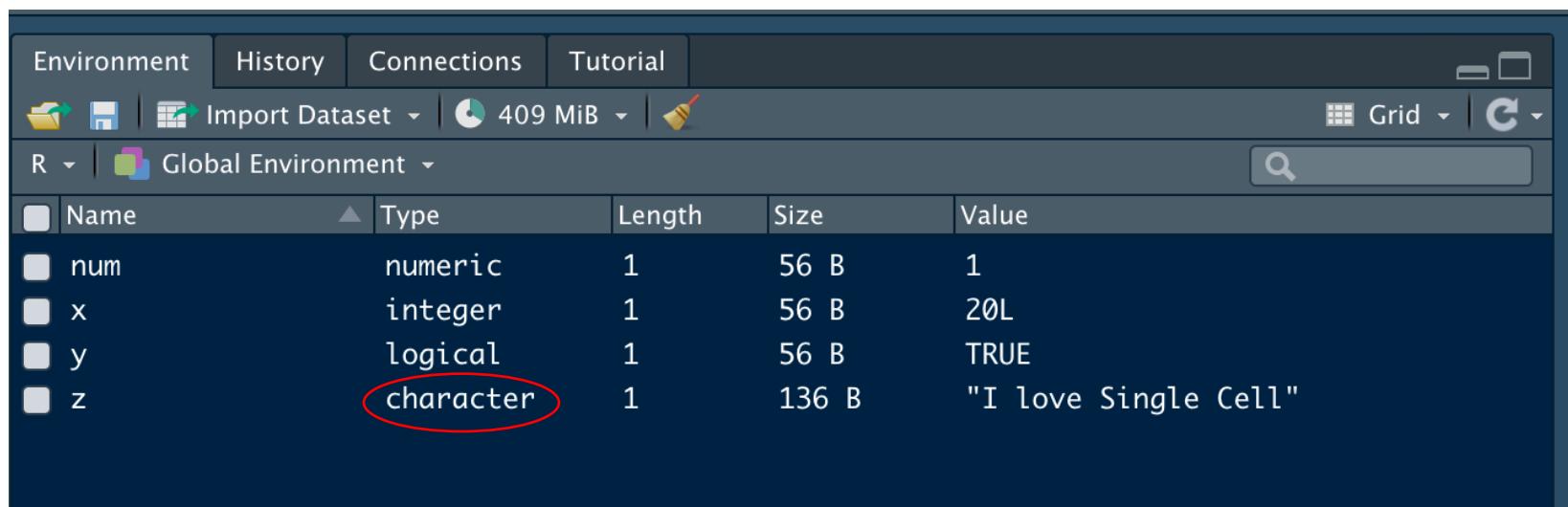
- The character data type stores character values or strings.
- Strings in R can contain the alphabet, numbers, and symbols. The easiest way to denote that a value is of character type in R is to wrap the value inside single or double inverted commas.

Code



```
Console Terminal × Background Jobs ×
R 4.1.1 · ~/Dropbox (MIT)/Sarah Ingabire's files/Ghana_workshop/ ↗
> z <- "I love Single Cell"
> class(z)
[1] "character"
> is.character(z)
[1] TRUE
>
```

Output



The screenshot shows the RStudio interface with the Global Environment tab selected. A table lists variables and their properties:

Name	Type	Length	Size	Value
num	numeric	1	56 B	1
x	integer	1	56 B	20L
y	logical	1	56 B	TRUE
z	character	1	136 B	"I love Single Cell"

Data Structures

Data structures are the objects that are manipulated regularly in R.

- Factors
- Vectors
- Lists
- Data Frames
- Matrices

Factors

A factor in R is also known as a categorical variable that stores both string and integer data values as levels.

i.e.

- `myfactor <- as.factor("male")` → This is a factor of 1 level

Vectors

Vector is one of the basic data structures in R. It is homogenous, which means that it only contains elements of the same data type.

To create a vector:

- assign the vector with c() operator. # c stands for concatenate
- check class using “class(nameofdatatype_vector)”

Data types can be:

- Numeric Vectors
i.e. numeric_vector <- c(1, 5, 3, 4, 6)
- Character Vectors
i.e. character_vector <- c("x", "y", "z", "27", "10", "my name is Laura")
- Factor Vectors- Same as writing a vector of character but give it "as.factor" function
i.e. factor_vector <- as.factor(c("A", "B", "C", "D", "E")). #Factor of 5 levels
- Logical Vectors
i.e. logical_vector <- c(F,F,TRUE,FALSE,T)

Vectors

Certain rules apply if different data types are put together in one vector

- Logical + numerical = numerical
- Logical + numerical + character = characters

Example

- `logic_num_vector <- c(1, FALSE)` → Numerical vector
- `logic_num_charact_vector <- c(2, TRUE, "x")` → Character Vector

List

List is a non-homogeneous data structure, which implies that it can contain elements of different data types.

List can be:

- numbers,
- characters,
- lists,
- matrices
- functions inside it.

To create a list, we use the “list()” function

i.e. `mylist <- list(numeric_vector, logical_vector, character_vector, 10, list(numeric_vector, logical_vector, character_vector), mean)`

Data Frame

Data frame is a list of equal-length vectors. Each element of the list can be thought of as a column and the length of each element of the list is the number of rows.

Example

```
> numeric_vector <- c(1, 5, 3, 4, 6)
> logical_vector <- c(F,F,TRUE,FALSE,T)
> mydataframe <- data.frame(numeric_vector, logical_vector)
> mydataframe2 <- data.frame(x = numeric_vector, y = logical_vector)
```

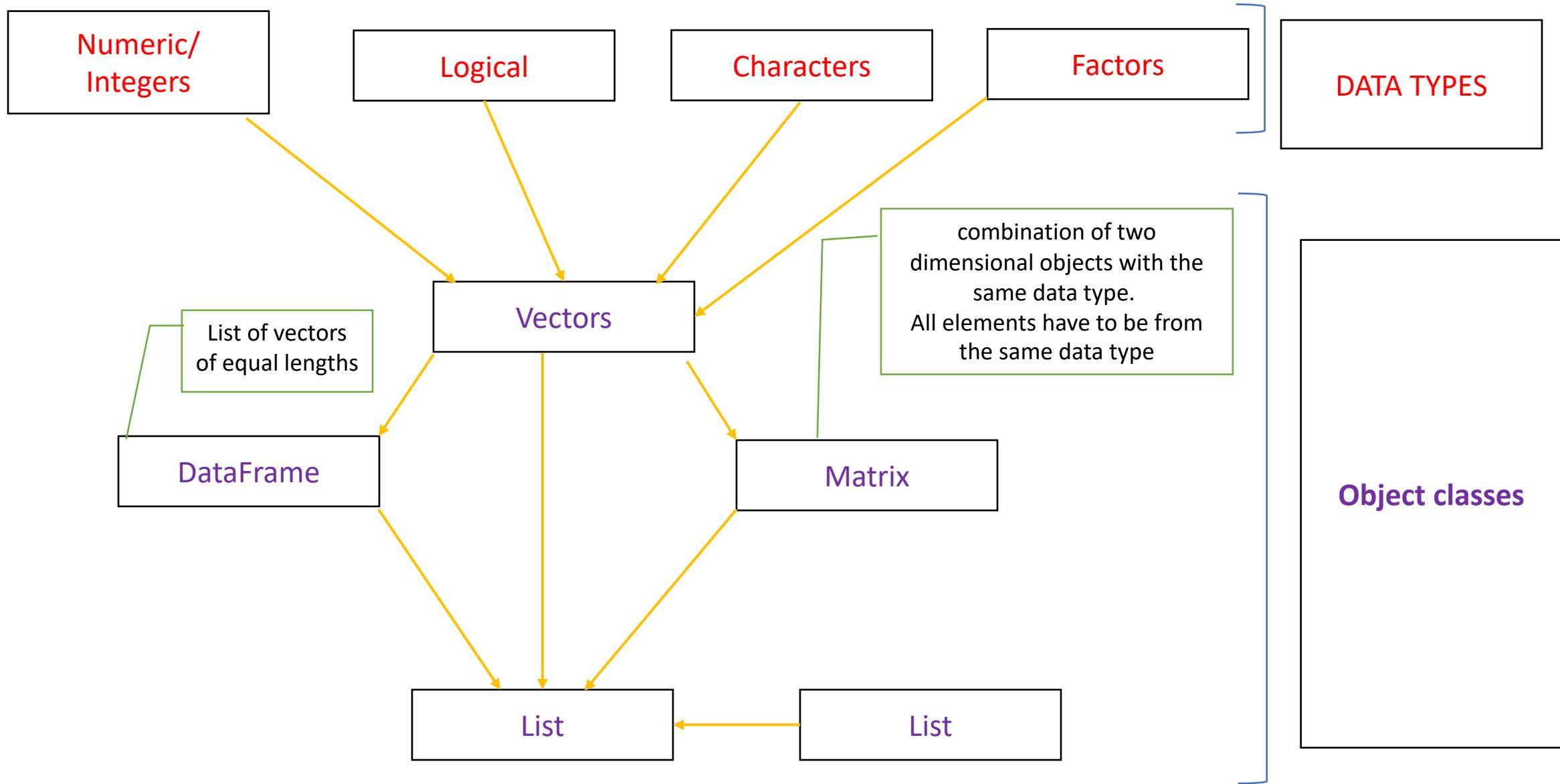
Matrix

Matrix is a combination of two dimensional objects with the same data type.

Let's create a character matrix

```
> character_vector <- c("x", "y", "z", "27", "10", "my name is Laura")
> mymatrix <- matrix(character_vector, nrow = 2, ncol = 2)
> class(mymatrix) # Check class
```

Review



References

- Journal of Computational and Graphical Statistics, Volume 19, Number 1, Pages 3–28 DOI: 10.1198/jcgs.2009.07098
 - Matthias Kohl. *Introduction to statistical data analysis with R*. bookboon.com, London, 2015.
 - Sarah Stowell. *Using R for Statistics*. Apress, 2014.
- Wickham, Hadley, and Garrett Grolemund. 2017. *R for Data Science: Visualize, Model, Transform, Tidy, and Import Data*. Sebastopol, California: O'Reilly Media, Inc. <https://r4ds.had.co.nz/n>.
- R Development Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL: <http://www.R-project.org>.