# Getting Started with R

R Intro

Gaston Sanchez

STAT 33B, Fall 2025

## About

In this slides we'll talk about some of aspects for getting started with R.

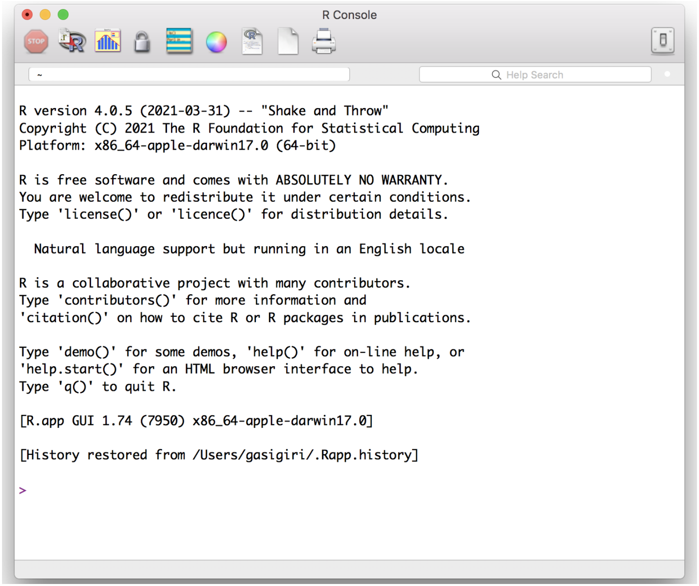
## Using R

## Using R Interactively

There are three main ways to use R interactively:

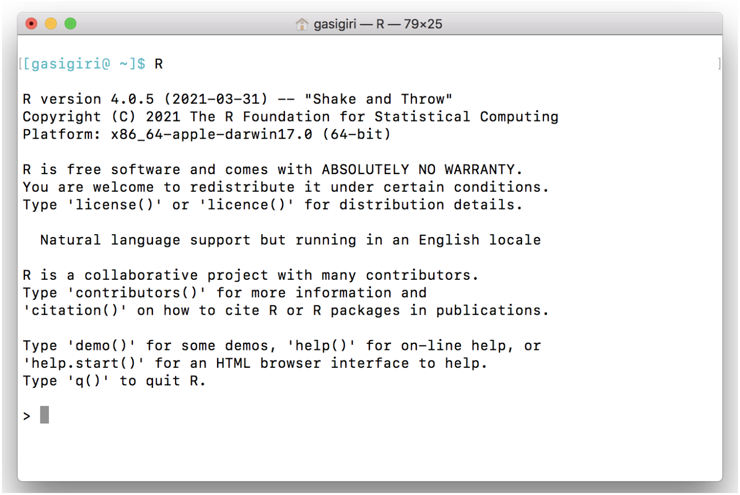
- R's built-in Graphical User Interface (GUI)
- R's console via a command line interface (terminal, shell)
- Via an Integrated Development Environment (IDE) such as RStudio (this is what we'll use in this course)

## R’s built-in GUI



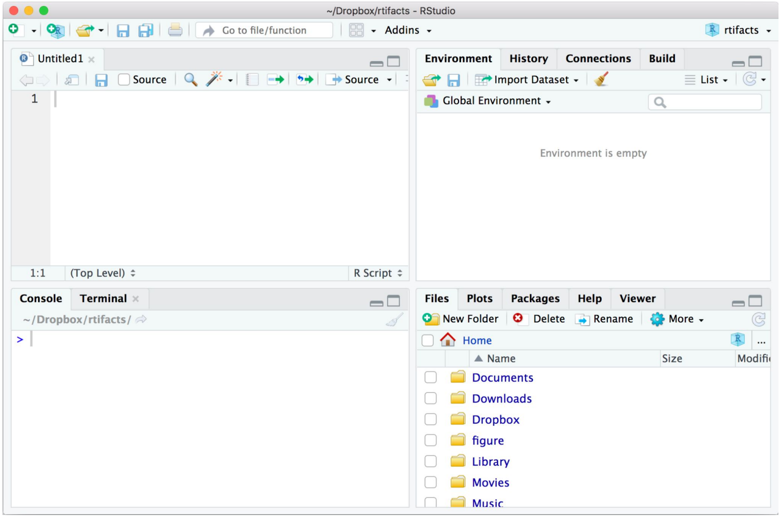
```
R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

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.

[R.app GUI 1.74 (7950) x86_64-apple-darwin17.0]

[History restored from /Users/gasigiri/.Rapp.history]

>
```

## R from Terminal



```
[gasigiri@ ~]$ R

R version 4.0.5 (2021-03-31) -- "Shake and Throw"
Copyright (C) 2021 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.

>
```

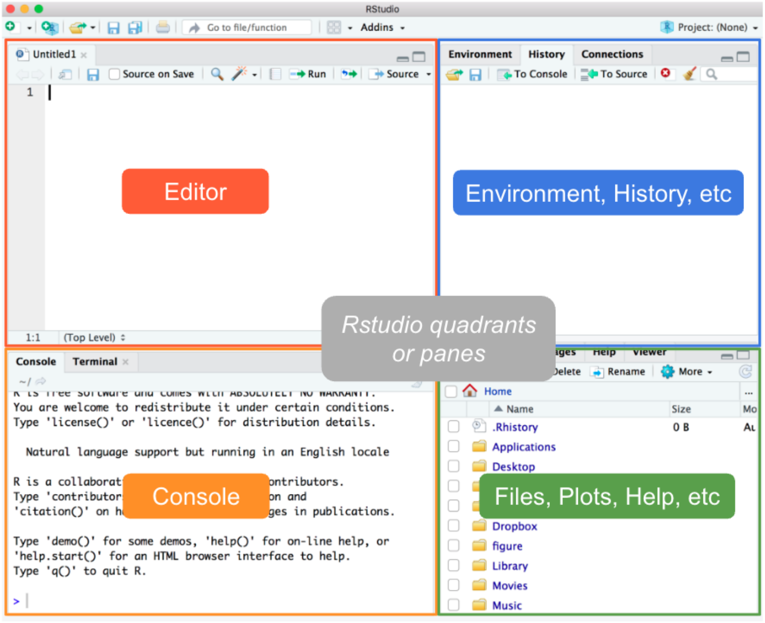
# R from RStudio



## R and RStudio

R $\neq$ RStudio

## RStudio Panes



```
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 collaborat                ontributors.
Type 'contributor                on and
'citation()' on h                ges 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.

>
```

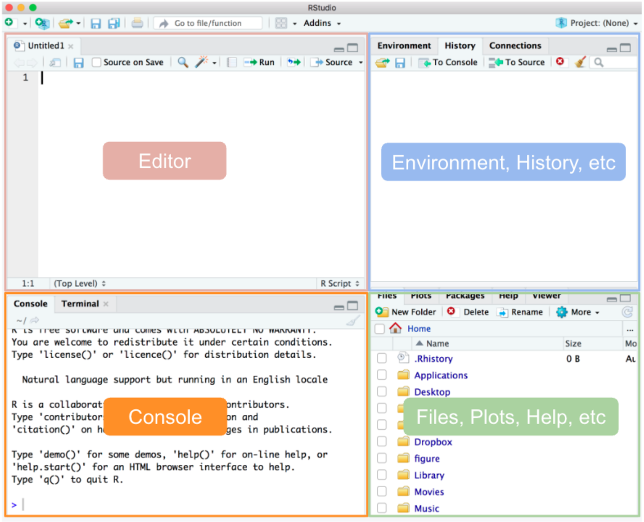
## Working with RStudio

RStudio provides an *Integrated Development Environment* (IDE) that makes it really easy to work with R (everything in a single window).

RStudio layout has four quadrants or *panes*:

- console pane
- editor pane
- environment & history pane
- files, plots, and help pane

## Console Pane

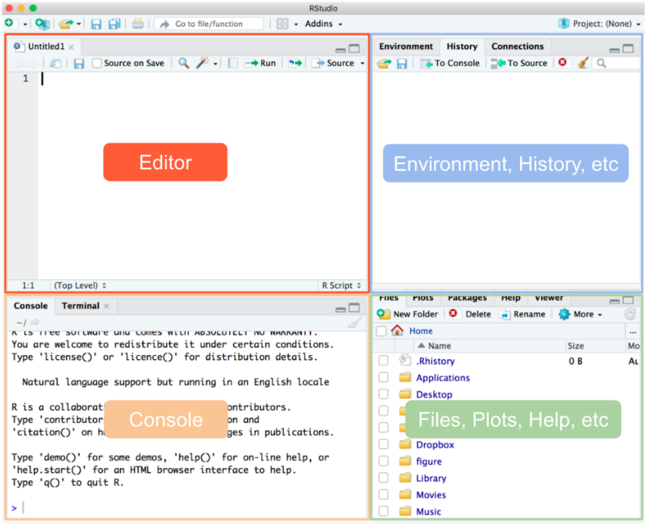


```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborat                   ontributors.
Type 'contributor                   on and
'citation()' on h                   ges 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.

> 
```

## Console Pane

- This is where you can directly interact with R.
- You type commands, and get numeric/text output.
- Good for running short-and-simple commands for exploratory purposes, or for trial-and-error commands.
- In reality, you won't be using this pane that much (instead, you'll use the Editor pane).
- The console is always associated to a working directory.
- By default, the working directory is your home directory, but you can customize it temporarily or permanently.

## Editor Pane



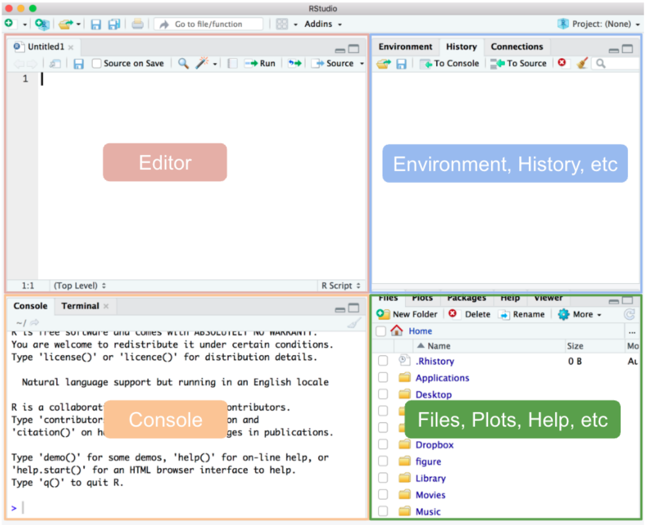
```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborat                ontributors.
Type 'contributor                on and
'citation()' on h                ges 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.

>
```

| | Name | Size | Mo |
|---|---|---|---|
| | .Rhistory | 0 B | Au |
| | Applications | | |
| | Desktop | | |
| | Dropbox | | |
| | figure | | |
| | Library | | |
| | Movies | | |
| | Music | | |

## Editor Pane

- Where you work with source documents (e.g. R script files, markdown files, etc)
- This is where you'll be working most of the time.
- Depending on the source file, you may write only R commands, or other text that follows a different syntax (e.g. markdown, latex, html)

## Files, Plots, Help, etc Pane



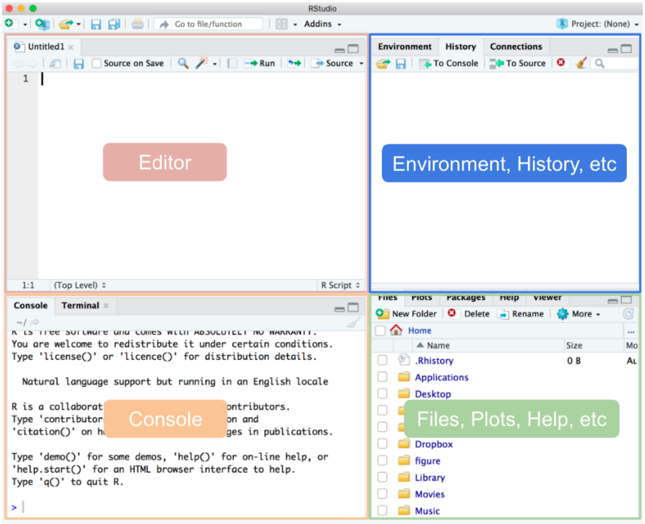
```
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborat          ontributors.
Type 'contributor          on and
'citation()' on h          ges 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.

>
```

| Name | Size | Mo |
|---|---|---|
| .Rhistory | 0 B | Au |
| Applications | | |
| Desktop | | |
| Dropbox | | |
| figure | | |
| Library | | |
| Movies | | |
| Music | | |

## Files, Plots, Help, etc Pane

- **Files tab:** allows you to interact with your file-system.
- **Plots:** graphics device to display graphics.
- **Packages**: to manage R packages.
- **Help:** to read supporting (help) documentation.
- **Viewer:** additional window to visualize certain outputs.

# Environment, History, etc Pane



```
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 collaborat                    ontributors.
Type 'contributor                    on and
'citation()' on h                    ges 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.

>
```

## Environment, History, etc Pane

- **Environment tab:** allows you to see the objects created in your workspace.
- **History tab:** lets you see all the commands that you've used (unless you delete such log).
- **Connections tab:** to manage connections (advanced topic for working with files; you will very rarely use this in practice).

## Quarto Markdown Files

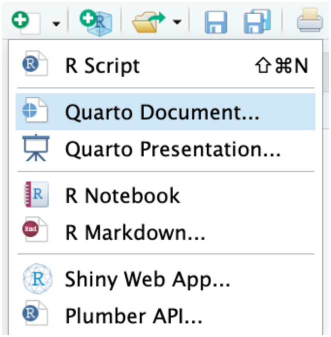
## About quarto and markdown files

- Quarto notebooks or `qmd` (quarto markdown) files are known as dynamic documents or computational documents.
- There are also `Rmd` (R markdown) files, which are the predecessors of `qmd` files.
- This type of document allows you to combine code and narrative in a single place.
- You can use markdown syntax, HTML, latex (for math equations), R code, and other programming langs.
- Most of the work done in this course will involve working with a `qmd` file.

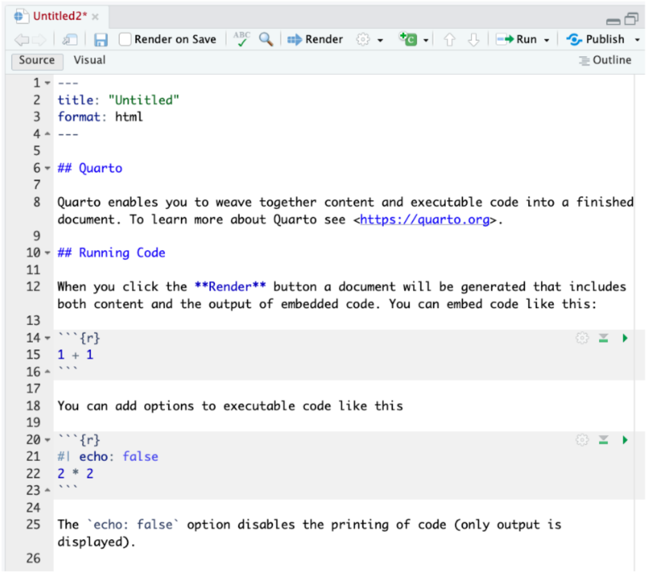
https://quarto.org/docs/get-started/hello/rstudio.html

## Qmd Files

One way to open a `qmd` file is by clicking on the *new file* icon (located in the upper left corner of Rstudio), and then selecting the `Quarto Document` option:



# Qmd Files



~~~
---
title: "Untitled"
format: html
---

## Quarto

Quarto enables you to weave together content and executable code into a finished
document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes
both content and the output of embedded code. You can embed code like this:

```{r}
1 + 1
~~~

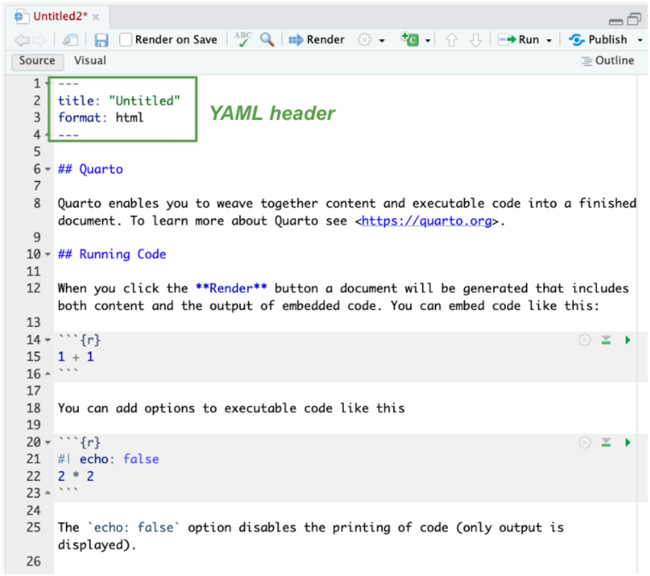
You can add options to executable code like this

```{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is
displayed).

```
```

## Qmd Files



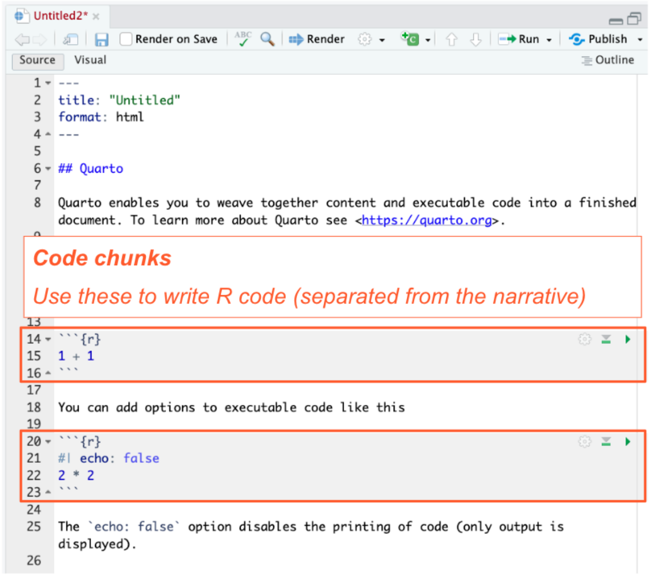
~~~
---
title: "Untitled"
format: html
---

## Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```{r}
1 + 1
~~~

You can add options to executable code like this

```{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).
```
```

## Qmd Files



```
---
title: "Untitled"
format: html
---

## Quarto

Quarto enables you to weave together content and executable code into a finished
document. To learn more about Quarto see <https://quarto.org>.
```

***Code chunks***

*Use these to write R code (separated from the narrative)*

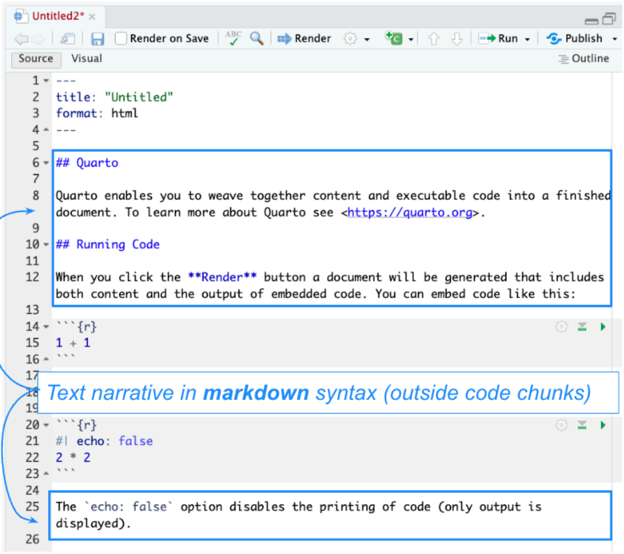
~~~
```{r}
1 + 1
~~~

You can add options to executable code like this

```{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is
displayed).
```
```

## Qmd Files

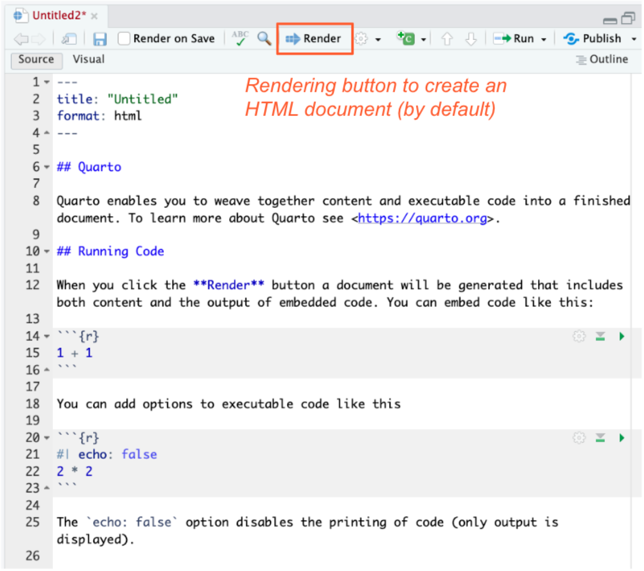


~~~
---
title: "Untitled"
format: html
---

## Quarto

Quarto enables you to weave together content and executable code into a finished
document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes
both content and the output of embedded code. You can embed code like this:

```{r}
1 + 1
~~~

```{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is
displayed).
```

*Text narrative in **markdown** syntax (outside code chunks)*
```

# Qmd Files



~~~
---
title: "Untitled"
format: html
---

## Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

## Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```{r}
1 + 1
```

You can add options to executable code like this

```{r}
#| echo: false
2 * 2
```

The `echo: false` option disables the printing of code (only output is displayed).

~~~

## Some Basic Concepts

## R Console

- Minimal GUI
- The console is OK for short expressions.
- The console is good as a calculator.
- But very limited for longer expressions.
- It's better to alternate with **source scripts** such as `R` script file(s), `qmd` or `Rmd` notebooks.

## Entering Input in R's Console

At the R prompt, > , we type *expressions*.

```
> 5 + 3
>
> "some text"
>
> 3^2
```

## CalculatoR

You can use R as a calculator; this is perhaps the simplest way to "break the ice" with R (especially if you are new to programming)

```r
2 + 3
4 - 1
3 * 4
10 / 2
3^3
```

## Using functions

You'll be constantly using functions. You type the name of the function, immediately followed by parenthesis. Inside parenthesis, you typically specify one or more inputs, separated by commas.

```
sqrt(9)

exp(1)

log(5, base = 10)

toupper("go bears!")
```

## Assignments

You can assign values to objects (i.e. variables) using the assignment operator **<-** or the equal sign **=** :

```
# assignment with 'arrow'
a <- 2 + 3

# assignment with 'equal'
b = 2 * 3
```

## Comments

The hash symbol # (or number sign) indicates a comment. Anything to the right of # is ignored.

```
# this is a comment
txt <- 'this is some text'

sqrt(9)  # example of square root

# ------------
# more comments
# ------------
```

## R basics

R is case sensitive!

```r
# Z different from z
Z <- 1
z <- 2
Z + z
```

```
## [1] 3
```

## R basics

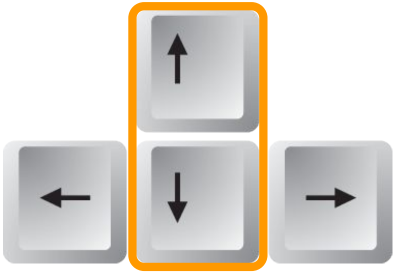
Case sensitive: this means that `"hello"` is not the same as `"Hello"` or `"HELLO"`

```r
hello <- "hello"
Hello <- "Hello"

# are they equal?
hello == Hello
```

```
## [1] FALSE
```

## R basics

When working in the R's console, use the up and down arrows to navigate through previous commands or instructions:



## R basics

Many languages use semicolons after each line. But in R there's (almost) no need to use semicolons

```r
# no need for semicolons
2 + 4
2 + 4;

# except in this case (which I don't recommend)
# to include various statements in the same line
2 + 4; A <- 2 * 5; B <- 'abc'
```

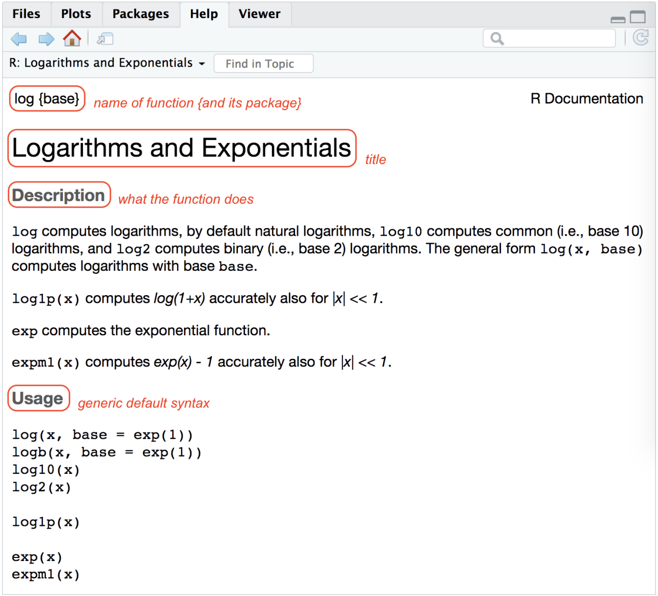
## Help (manual)
## Documentation

## Help

To know about a function, use the `help()` or preappend the question mark `?`

For example, say you want to know about the `log()` function:

```r
help("log")

# equivalent
?log
```



log {base} *name of function {and its package}* R Documentation

## Logarithms and Exponentials *title*

### Description *what the function does*

`log` computes logarithms, by default natural logarithms, `log10` computes common (i.e., base 10) logarithms, and `log2` computes binary (i.e., base 2) logarithms. The general form `log(x, base)` computes logarithms with base `base`.

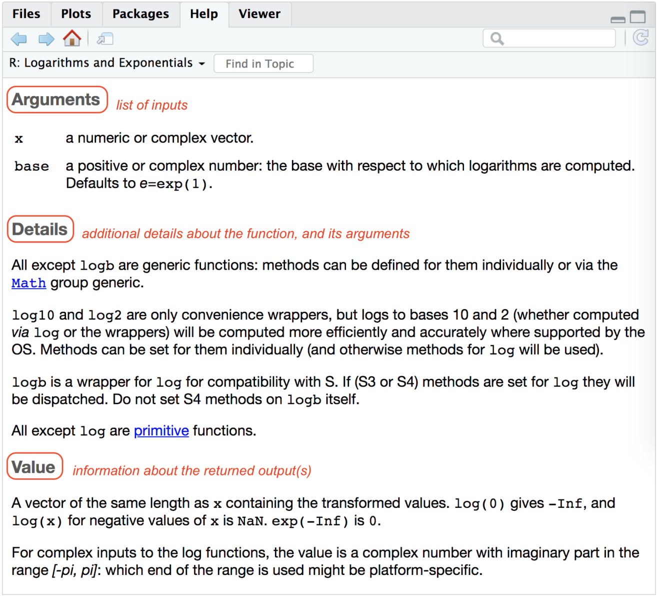
`log1p(x)` computes *log(1+x)* accurately also for |*x*| << *1*.

`exp` computes the exponential function.

`expm1(x)` computes *exp(x) - 1* accurately also for |*x*| << *1*.

### Usage *generic default syntax*

```
log(x, base = exp(1))
logb(x, base = exp(1))
log10(x)
log2(x)

log1p(x)

exp(x)
expm1(x)
```



**Arguments** *list of inputs*

| | |
|---|---|
| `x` | a numeric or complex vector. |
| `base` | a positive or complex number: the base with respect to which logarithms are computed. Defaults to *e*=`exp(1)`. |

**Details** *additional details about the function, and its arguments*

All except `logb` are generic functions: methods can be defined for them individually or via the `Math` group generic.

`log10` and `log2` are only convenience wrappers, but logs to bases 10 and 2 (whether computed *via* `log` or the wrappers) will be computed more efficiently and accurately where supported by the OS. Methods can be set for them individually (and otherwise methods for `log` will be used).

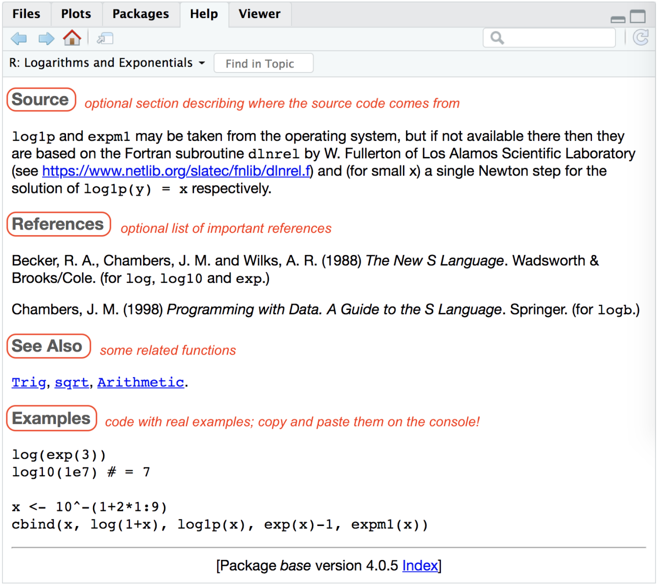
`logb` is a wrapper for `log` for compatibility with S. If (S3 or S4) methods are set for `log` they will be dispatched. Do not set S4 methods on `logb` itself.

All except `log` are primitive functions.

**Value** *information about the returned output(s)*

A vector of the same length as `x` containing the transformed values. `log(0)` gives `-Inf`, and `log(x)` for negative values of `x` is `NaN`. `exp(-Inf)` is `0`.

For complex inputs to the log functions, the value is a complex number with imaginary part in the range *[-pi, pi]*: which end of the range is used might be platform-specific.



**Source** *optional section describing where the source code comes from*

`log1p` and `expm1` may be taken from the operating system, but if not available there then they are based on the Fortran subroutine `dlnrel` by W. Fullerton of Los Alamos Scientific Laboratory (see https://www.netlib.org/slatec/fnlib/dlnrel.f) and (for small x) a single Newton step for the solution of `log1p(y) = x` respectively.

**References** *optional list of important references*

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole. (for `log`, `log10` and `exp`.)

Chambers, J. M. (1998) *Programming with Data. A Guide to the S Language*. Springer. (for `logb`.)

**See Also** *some related functions*

`Trig`, `sqrt`, `Arithmetic`.

**Examples** *code with real examples; copy and paste them on the console!*

```
log(exp(3))
log10(1e7) # = 7

x <- 10^-(1+2*1:9)
cbind(x, log(1+x), log1p(x), exp(x)-1, expm1(x))
```

[Package *base* version 4.0.5 Index]

## Session Management

## Opening a new session

When you start a new session, the following welcome message is displayed on the console:

```
R version 4.5.0 (2025-04-11) -- "How About a Twenty-Six"
Copyright (C) 2025 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20

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 Version

- Usually, about 2 or 3 versions of R released per year
- Each version has its own name
- e.g. `R version 4.5.0 -- "How About a Twenty-Six"`

## R's Starting Message

```r
# GNU GPL2 license
license()

# humans behind R
contributors()

# citing R
citation()

# some demos
demo()

# on-line help
help()
```

## History tab, and `.Rhistory` file

- All the commands that you execute in a session will be displayed in your **History** tab.
- The very first time you launch R (via Rstudio), it will create a (hidden) text file called `.Rhistory`
- This file is located in the working directory (by default this is your home directory).
- When you close a session, all the commands in your **History** tab will be saved in the `.Rhistory` file.
- It's good to know that you can always check the history file to "go back in time" and see what commands you used.

## Environment tab and `.RData` file

- When you create objects; e.g. `x = 10` , they become part of the so-called **Global Environment**.
- These objects are listed in the **Environment** tab.
- The set of objects created (in the Global Environment) during a session are part of your **workspace**.
- When you quit R, it will ask you whether you want to save all the objects (your workspace) in a (binary) file called **`.RData`**

## Environment tab and `.RData` file

- If you save your workspace in the **`.RData`** file, next time you open a session those objects will be available in your **Environment**.
- This can be good or bad depending on what you are doing or what you are working on.
- Personally, I tend to not save my workspace (to avoid hoarding multiple objects that can cluttered my sessions).
- Instead, I prefer to create my own `.RData` files when I need them.

## Terminate an R session

To quit a session simply type `quit()` or `q()` in R's console:

```r
# saves your workspace
quit(save = "yes")
```

```r
# doesn't save your workspace
quit(save = "no")
```

You can also click the `File` tab in the menu bar, and then click on the option `Quit Session`

## Saved Workspace

If you previously typed `q("yes")`, open a new R session and inspect what objects do you have:

```r
# list objectst in your workspace
ls()
```

## One option to record your work

- In addition to `quit(save = "yes")`, there's also the function `savehistory()`
- You can use `savehistory()` to save everything you did
- It may be useful to call `savehistory()` at the end of a session
- By default, the commands-history will be saved in a file called `.Rhistory` (you can use other extension)
- You can open this file in any text editor

## Recording your work

Type some expressions, save your commands-history, and then quit R (without saving workspace)

```
2 * 2
2^10

# first comment
course <- "stat133"

# converting units
height_ft <- 5.9
height_in <- height_ft * 12
height_m <- height_ft * 0.3048

savehistory(file = 'test-session.R')
quit(save = "no")
```

Open the file `"test-session.R"` and see what's in it

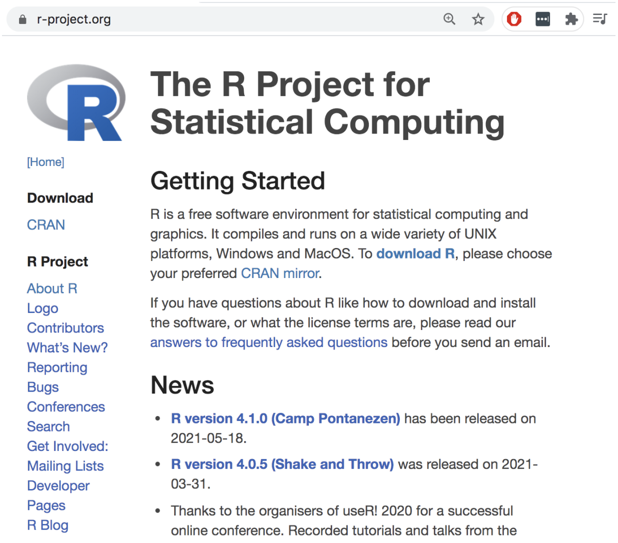
## Learning R (or any other language) reminder

While learning R (or any programming language), keep in mind:

- You'll get frustrated.
- It takes time to become fluent.
- Lots of trials and errors.
- Be patient.
- Practice, practice, practice.

## Additional Resources

## R Website



r-project.org

[Home]

**Download**

CRAN

**R Project**

About R
Logo
Contributors
What's New?
Reporting Bugs
Conferences
Search
Get Involved: Mailing Lists
Developer Pages
R Blog

# 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.

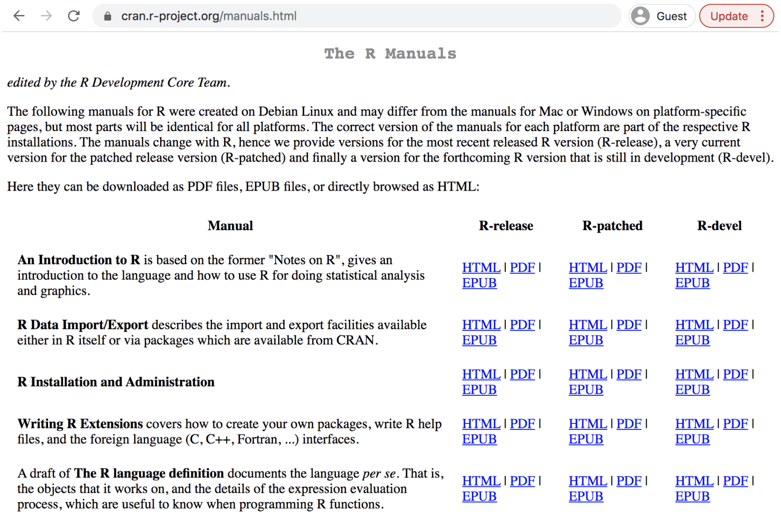
If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

## News

- **R version 4.1.0 (Camp Pontanezen)** has been released on 2021-05-18.
- **R version 4.0.5 (Shake and Throw)** was released on 2021-03-31.
- Thanks to the organisers of useR! 2020 for a successful online conference. Recorded tutorials and talks from the

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

## R Technical Manuals



**The R Manuals**

*edited by the R Development Core Team.*

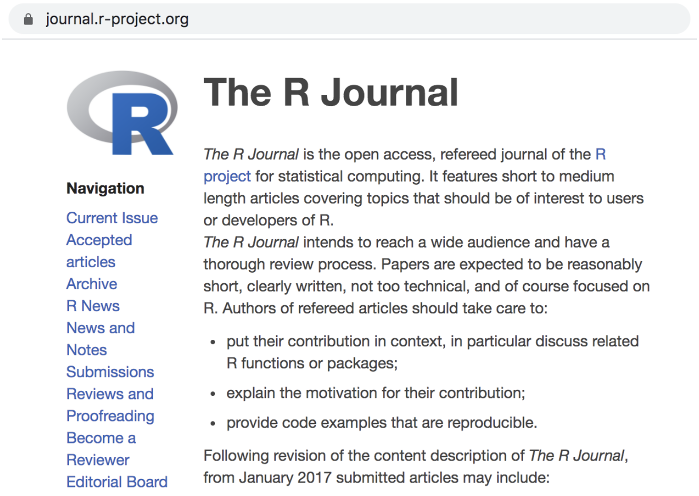
The following manuals for R were created on Debian Linux and may differ from the manuals for Mac or Windows on platform-specific pages, but most parts will be identical for all platforms. The correct version of the manuals for each platform are part of the respective R installations. The manuals change with R, hence we provide versions for the most recent released R version (R-release), a very current version for the patched release version (R-patched) and finally a version for the forthcoming R version that is still in development (R-devel).

Here they can be downloaded as PDF files, EPUB files, or directly browsed as HTML:

| Manual | R-release | R-patched | R-devel |
|---|---|---|---|
| **An Introduction to R** is based on the former "Notes on R", gives an introduction to the language and how to use R for doing statistical analysis and graphics. | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB |
| **R Data Import/Export** describes the import and export facilities available either in R itself or via packages which are available from CRAN. | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB |
| **R Installation and Administration** | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB |
| **Writing R Extensions** covers how to create your own packages, write R help files, and the foreign language (C, C++, Fortran, ...) interfaces. | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB |
| A draft of **The R language definition** documents the language *per se*. That is, the objects that it works on, and the details of the expression evaluation process, which are useful to know when programming R functions. | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB | HTML \| PDF \| EPUB |

https://cran.r-project.org/manuals.html

## R Journal



journal.r-project.org

# The R Journal

**Navigation**

- Current Issue
- Accepted articles
- Archive
- R News
- News and Notes
- Submissions
- Reviews and Proofreading
- Become a Reviewer
- Editorial Board

*The R Journal* is the open access, refereed journal of the R project for statistical computing. It features short to medium length articles covering topics that should be of interest to users or developers of R.
*The R Journal* intends to reach a wide audience and have a thorough review process. Papers are expected to be reasonably short, clearly written, not too technical, and of course focused on R. Authors of refereed articles should take care to:

- put their contribution in context, in particular discuss related R functions or packages;
- explain the motivation for their contribution;
- provide code examples that are reproducible.

Following revision of the content description of *The R Journal*, from January 2017 submitted articles may include:

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