

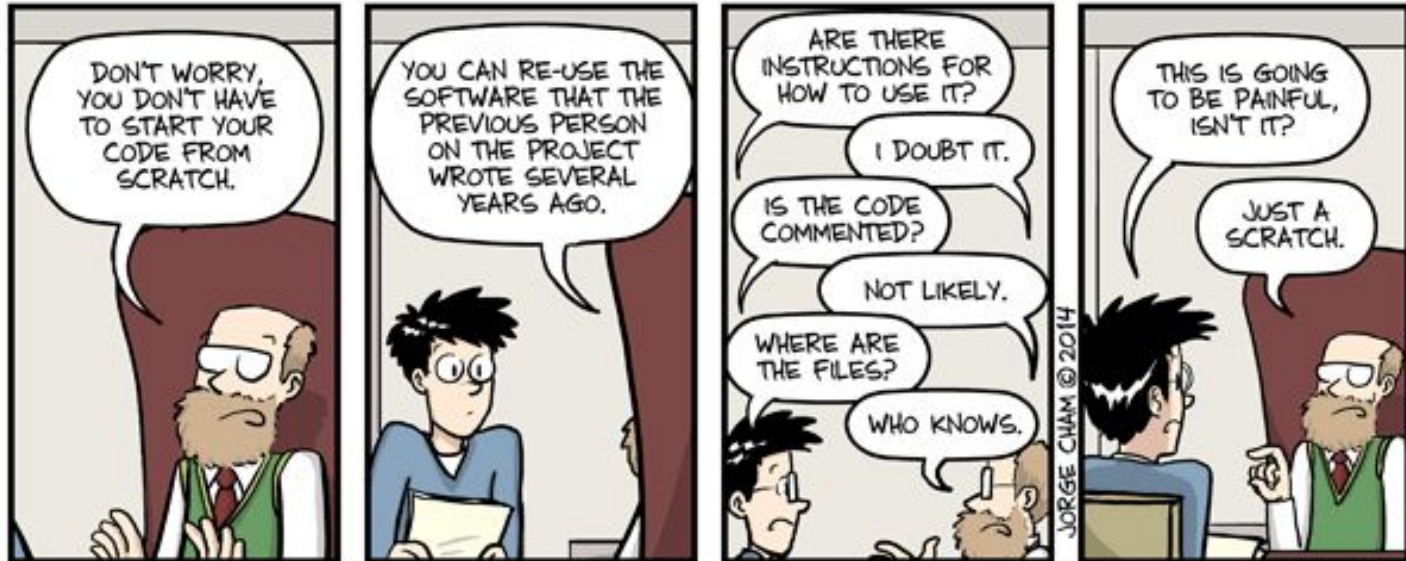


# Introduction to R, RStudio, and RStudio Server

The Data Lab

Powered by Alex's Lemonade Stand Foundation

# Who's been here before?



One of our goals is to help you write code that makes this process much, much less painful!

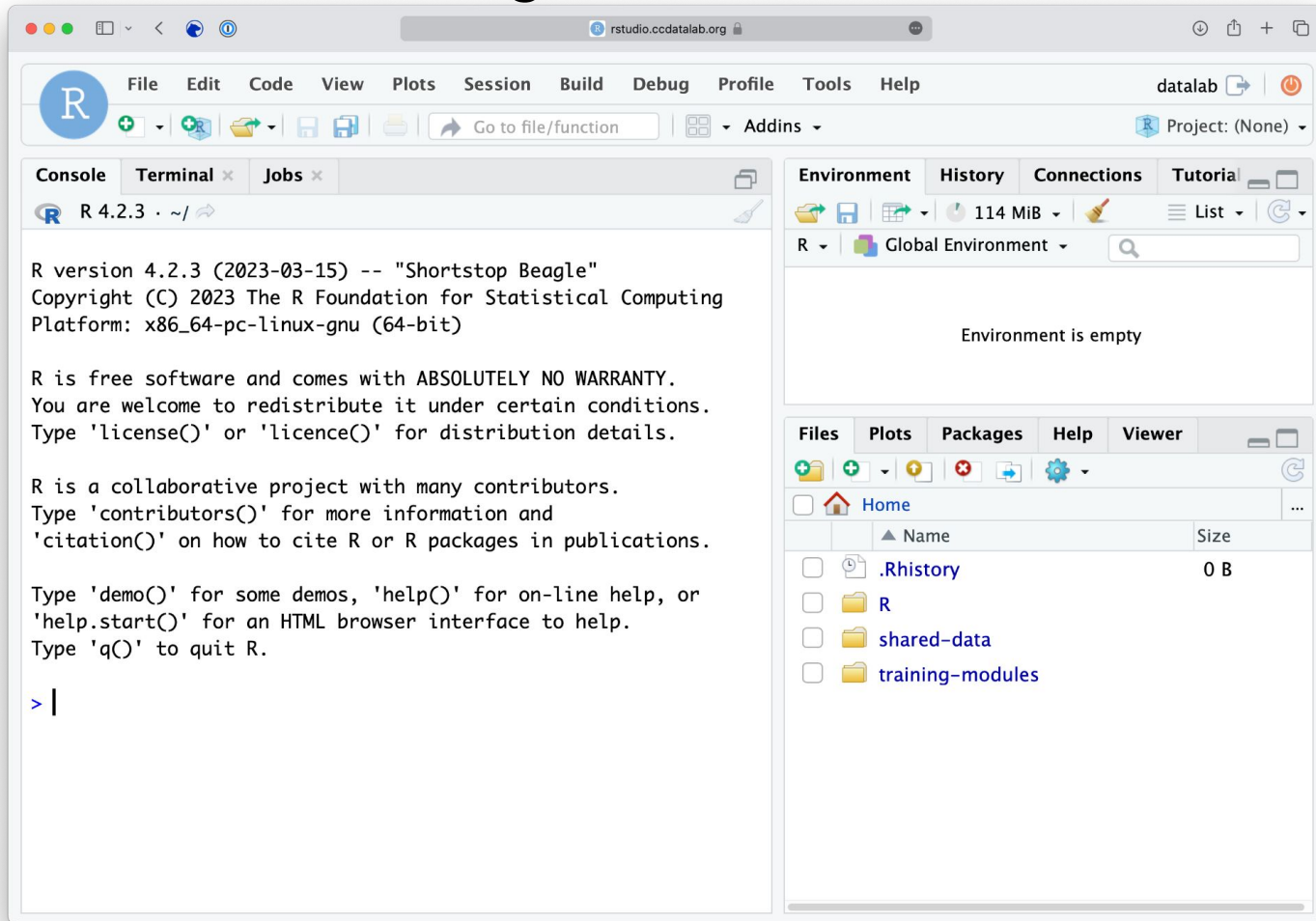
# Command line vs GUI (graphics user interface)

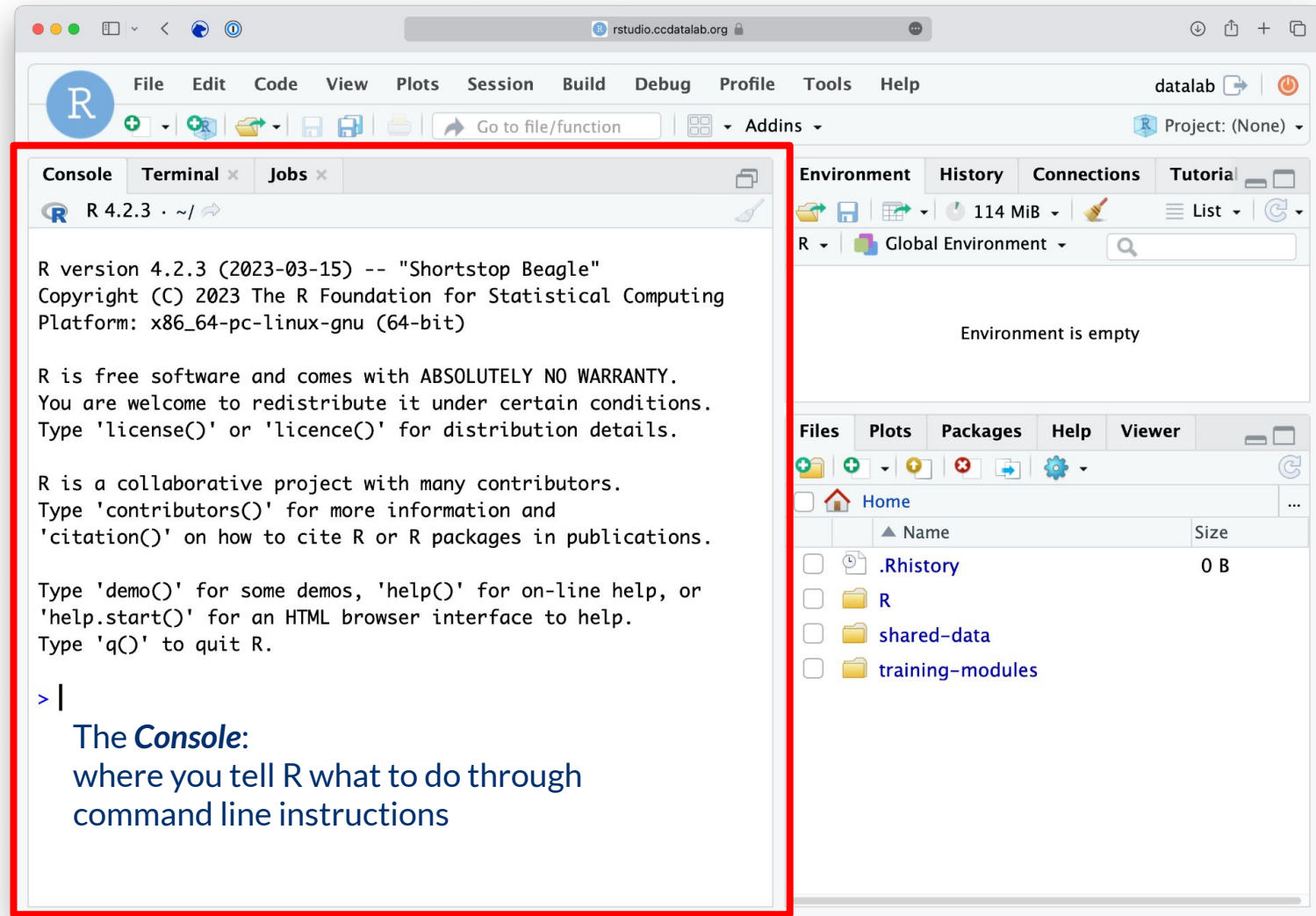
- An interface is how you interact with a program
- GUI's have buttons you can *click* to do things, but...
- Command-line interfaces (CLI) have you *type* out things to do them

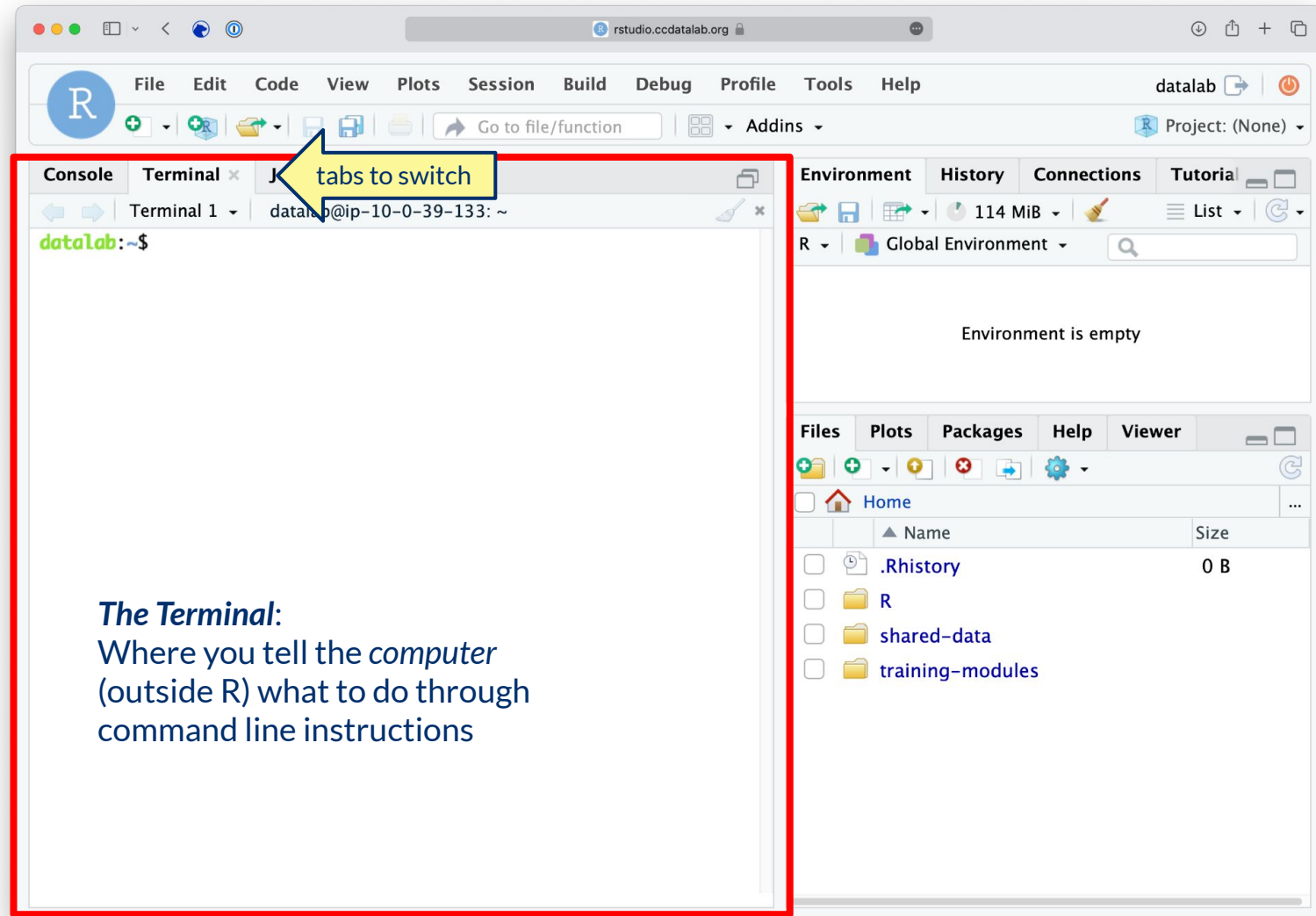


# RStudio Server: A basic guide

(FYI, we're now  
at [R 4.4.0](#))







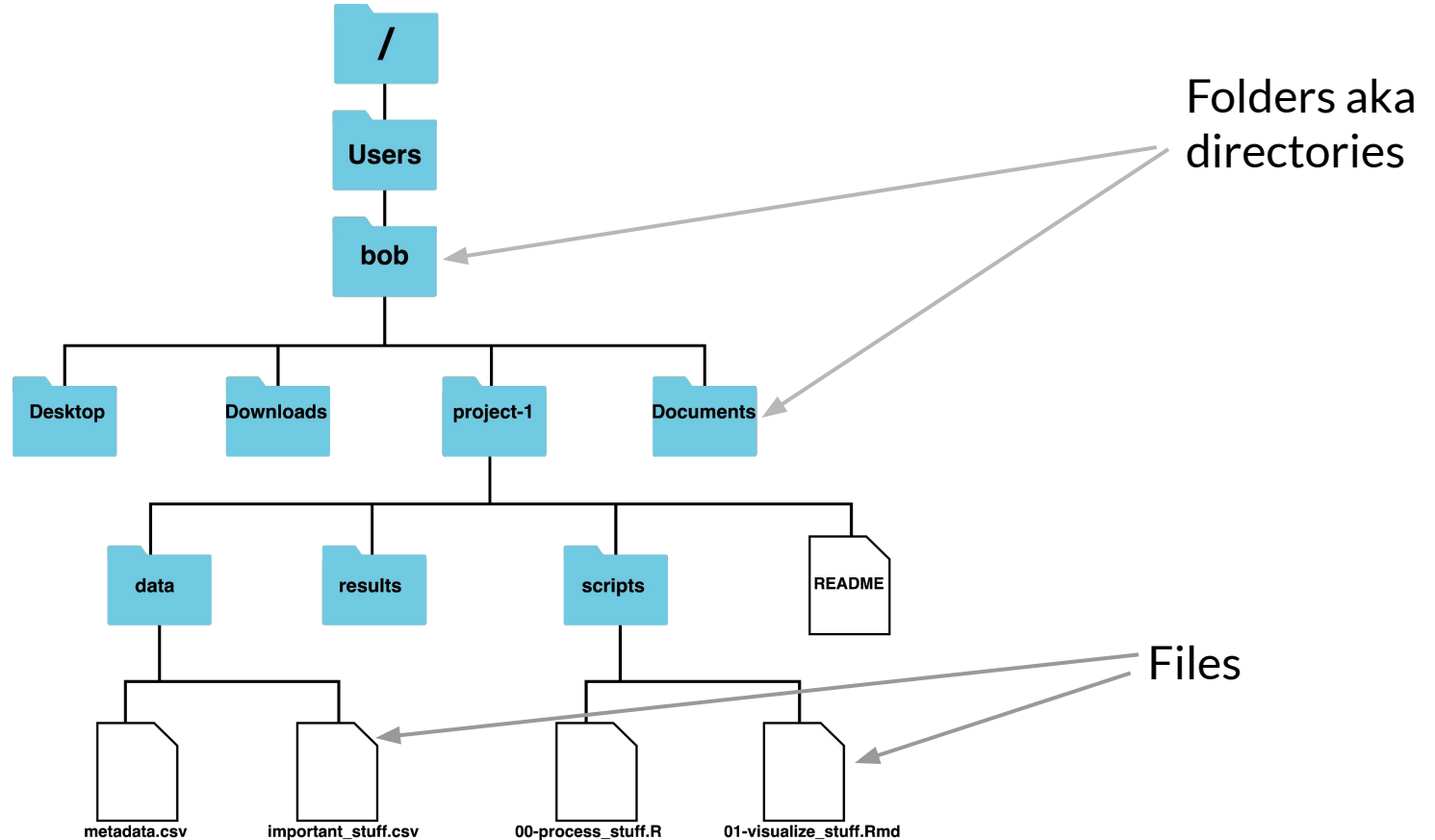
The image shows a screenshot of the RStudio web interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top right shows 'datalab' and a power icon. Below the menu bar is a toolbar with icons for adding files, saving, and navigating. The main interface is divided into several panes. On the left, the 'Console' pane is active, showing a terminal window with the prompt 'datalab:~\$'. Two arrows point from the text below to the '~' and '\$' in the prompt. To the right of the terminal is the 'Environment' pane, which is empty. Below the environment pane is the 'Files' pane, showing a file explorer view of the 'Home' directory with files like '.Rhistory', 'R', 'shared-data', and 'training-modules'. The bottom right pane is labeled 'Viewer'.

These indicate what **directory** you are **currently** carrying out a command in

This is called your "**current directory**"

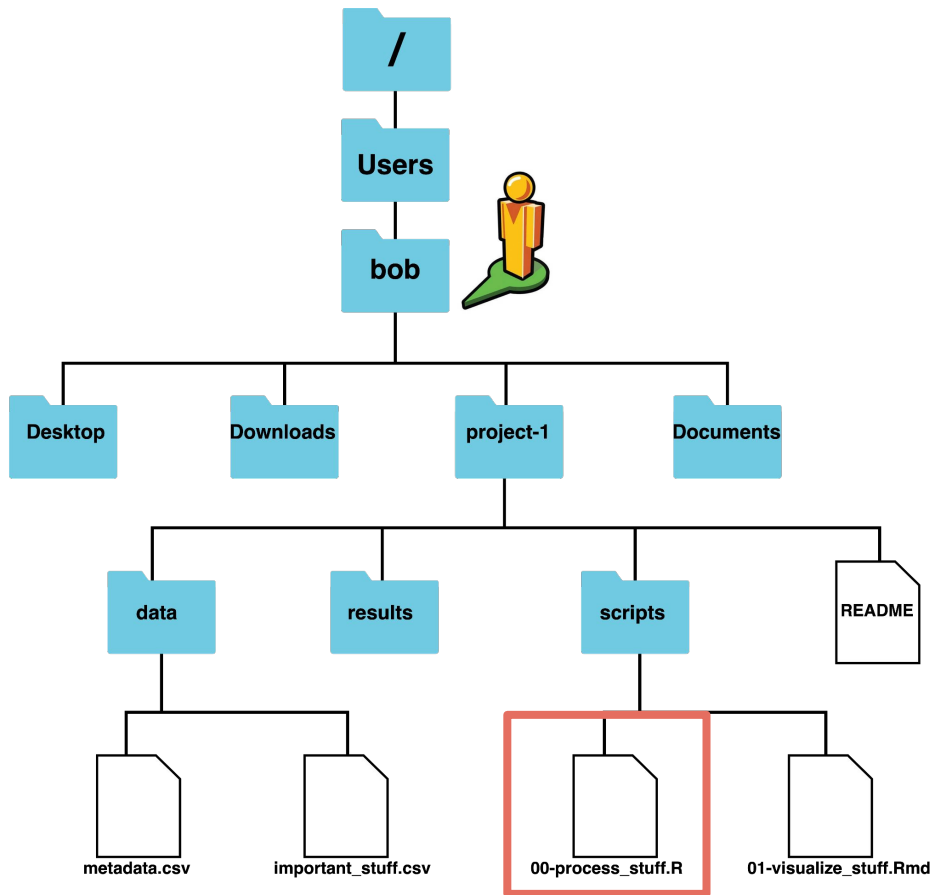
"~" is a shortcut for your "**Home**" directory, so these mean the same thing.

# Example of a filesystem hierarchy





# We are always working somewhere!



Assume we are working “from” the **bob** directory.  
This means **bob** is the *current (working) directory*

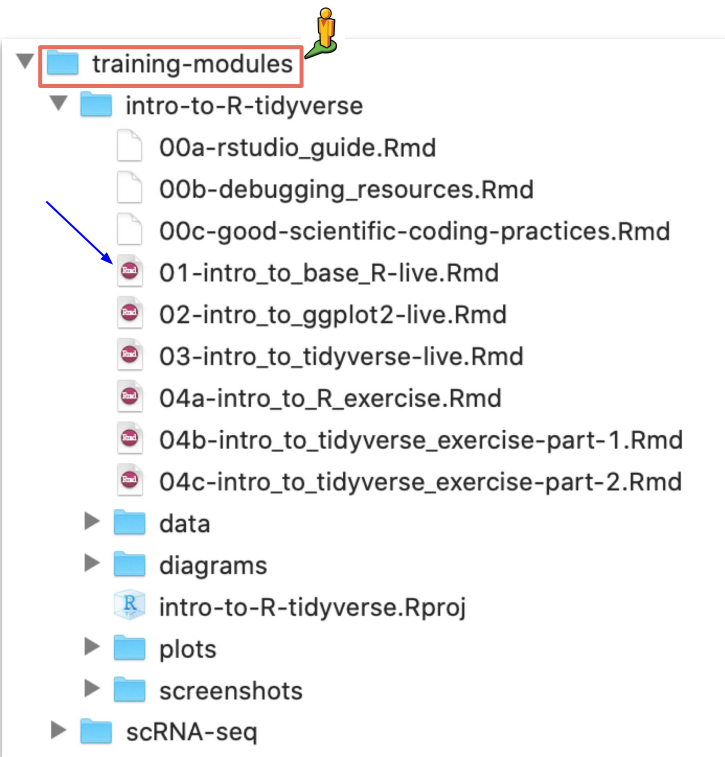
We therefore need to know the **paths** to files we are working with, relative to our working directory, to be able to use those files in our code.

The file we are working on

Relative path: **project-1/scripts/00-process\_stuff.R**

# Let's look at our workshop files

Let's say we want access to `01-intro_to_base_R-live.Rmd`

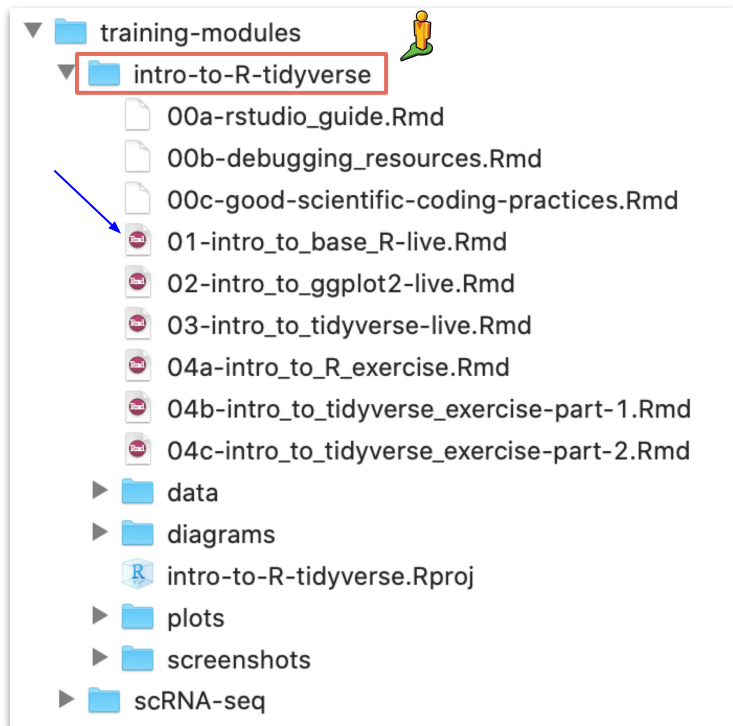


Current/working directory = `training-modules`

File path = `intro-to-R-tidyverse/01-intro_to_base_R-live.Rmd`

# Relative paths depend on your working directory

Let's say we want access to `01-intro_to_base_R-live.Rmd`



Current/working directory = `training-modules/intro-to-R-tidyverse`

File path = `01-intro_to_base_R-live.Rmd`

The screenshot shows the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top toolbar contains icons for creating a new file, opening a file, saving, and other standard RStudio functions. The main workspace is divided into four panes: Console, Terminal, Jobs, and Environment. The Console pane shows the following commands and output:

```
datalab:~$ ls
R shared-data training-modules
datalab:~$ cd training-modules/
```

Below the Console pane, there is a text box with the following text:

Some common **Terminal** commands:

- ls** - list the files and folders in a directory (files that start with a '.' are not shown by default)
- cd** - change directories

The Environment pane shows the Global Environment, which is currently empty. The bottom pane shows the Files view, displaying the directory structure:

	Name	Size	Modified
<input type="checkbox"/>	.Rhistory	0 B	Jan 31, 2023
<input type="checkbox"/>	R		
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		

The screenshot displays the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar below the menu contains various icons for file operations and navigation. The main workspace is divided into four panes: Console, Terminal, Jobs, and Environment. The Console pane shows a terminal session with the following commands and output:

```
datalab:~$ ls
R shared-data training-modules
datalab:~$ cd training-modules/
datalab:~/training-modules$ ls
LICENSE.md README.md intro-to-R-tidyverse
module-cheatsheets scrna-seq
datalab:~/training-modules$
```

The words "training-modules" in the last command are highlighted with a red box. Below the terminal output, a text box contains the following explanation:

The words in front of our cursor have changed because we are now "in" the `training-modules` directory

The Environment pane on the right shows the current environment is empty. Below it, the Files pane displays the file structure of the current directory:

	Name	Size	Modified
<input type="checkbox"/>	.Rhistory	0 B	Jan 31, 2023
<input type="checkbox"/>	R		
<input type="checkbox"/>	shared-data		
<input type="checkbox"/>	training-modules		

The screenshot shows the RStudio application window. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top toolbar contains icons for creating a new file, opening a file, saving, and other standard RStudio functions. The top right shows the 'datalab' project name and a power button icon.

The main interface is divided into four panes:

- Console:** Shows the terminal output. The prompt is `datalab:~$`. The commands entered are `ls`, `cd training-modules/`, and `ls`. The output of the second `ls` command is `LICENSE.md README.md intro-to-R-tidyverse module-cheatsheets scrna-seq`. The current directory is `~/training-modules`, which is highlighted with a red box.
- Environment:** Shows the current environment. It is empty, with the text "Environment is empty".
- Files:** Shows the file explorer. The current directory is `Home`. The files listed are `.Rhistory`, `R`, `shared-data`, and `training-modules`. The `training-modules` directory is highlighted with a red box.
- Plots:** Empty.
- Packages:** Empty.
- Help:** Empty.
- Viewer:** Empty.

The text "The words in front of our cursor have changed because we are now 'in' the `training-modules` directory" is overlaid on the console output.

The text "The files tab over here does **NOT** reflect your current directory or any changes within it" is overlaid on the files panel.



# Introduction to R

The Data Lab

# R programming

Programming: making executable scripts for accomplishing a task  
(in this case, data analysis is our task)

Scripts allow others to see, step-by-step, what you did.

## Why we use R:

- It's free and open-source
- People make cool packages that do stuff for us
- Many researchers in genomics use it (as well as Python)





# R, RStudio, and RStudio Server

R is a statistical programming language.



RStudio is an IDE for working in R

- IDE: Integrated Development Environment
- We write R code using the (free!) RStudio IDE



**RStudio Server** allows us to run the **RStudio IDE** from a browser

The screenshot displays the RStudio IDE interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top toolbar contains icons for saving, running, and other functions. The main editor window shows a live R notebook with the following content:

```
84 | Divide | ` / ` |
85 | Exponentiate | ` ^ ` or ` ** ` |
86
87 For example, we can do some simple multiplication like this.
88 When you execute code within the notebook, the results appear beneath
89 the code.
90 Try executing this chunk by clicking the *Run* button within the chunk
91 or by
92 placing your cursor inside it and pressing *Cmd+Shift+Enter*.
93
94 ```{r calculator}
95 5 * 6
96 ```
```

The output of the code chunk is displayed below the code:

```
[1] 30
```

Below the output, there is a text block:

```
97
98 Use the console to calculate other expressions. Standard order of
99 operations applies (mostly), and you can use parentheses `()` as you
100 might expect (but not brackets `[]` or braces `{}`, which have special
101 meanings). Note however, that you must **always** specify multiplication
102 with `*`; implicit multiplication such as `10(3 + 4)` or `10x` will not
103 work and will generate an error, or worse.
104
105 ```{r expressions, live = TRUE}
106
107 ```
```

The bottom status bar shows the current file is "105:193" and the mode is "R Markdown".

On the right side, the "Environment" pane shows "Environment is empty". The "Files" pane shows a file explorer view with the following table:

Name	Size	Modified
00a-intro_to_tidyverse	13.6 KB	Apr 7, 2023
00b-debugging_resources.md	3.9 KB	Apr 7, 2023
00c-good-scientific-coding-pr...	19 KB	Apr 7, 2023
01-intro_to_base_R-live.Rmd	11.5 KB	Apr 7, 2023
02-intro_to_ggplot2-live.Rmd	21.3 KB	Apr 7, 2023
03-intro_to_tidyverse-live.Rmd		
exercise_03a-intro_to_tidyvers...	6.2 KB	Apr 7, 2023
exercise_03b-intro_to_tidyvers...	1.8 KB	Apr 7, 2023
README.md		

A red box highlights the "Console" icon in the bottom right corner of the RStudio interface, with a red arrow pointing to it. A text box next to the arrow says "Click here to show the Console".

The screenshot displays the RStudio IDE interface. The main editor window shows an R script file named `01-intro_to_base_R-live.Rmd`. The script contains a code chunk with the following content:

```
[1] 30

95
96 Use the console to calculate other expressions. Standard order of
   operations applies (mostly), and you can use parentheses `()` as you
   might expect (but not brackets `[]` or braces `{}`, which have special
   meanings). Note however, that you must **always** specify multiplication
   with `*`; implicit multiplication such as `10(3 + 4)` or `10x` will not
   work and will generate an error, or worse.
97
98 ```{r expressions, live = TRUE}
99 x <- 5.5
100
101 x
102 ```
```

The console output for the code chunk is:

```
[1] 5.5
```

The Environment pane on the right shows the Global Environment with a variable `x` having a value of `5.5`.

The Files pane on the right shows the file explorer view, displaying a list of files and folders in the `training-modules > intro-to-R-tidyverse` directory. The files include:

- `00a-tidyverse_garden.md` (3.9 KB)
- `00b-debugging_resources.md` (13.6 KB)
- `00c-good-scientific-coding-pr...` (3.9 KB)
- `01-intro_to_base_R-live.Rmd` (19 KB)
- `02-intro_to_ggplot2-live.Rmd` (11.5 KB)
- `03-intro_to_tidyverse-live.Rmd` (21.3 KB)
- `data` (folder)
- `diagrams` (folder)
- `exercise_01-intro_to_base_R.R...` (4.8 KB)
- `exercise_02-intro_to_R.Rmd` (4.7 KB)
- `exercise_03a-intro_to_tidyvers...` (7.7 KB)
- `exercise_03b-intro_to_tidyvers...` (6.2 KB)
- `README.md` (1.8 KB)

The Console pane at the bottom shows the R prompt and the execution of the code chunk:

```
R 4.2.3 ~ /
> x <- 5.5
>
> x
[1] 5.5
> |
```

**R Console:**  
What you are actually telling R to do

The screenshot shows the RStudio IDE interface. A red box highlights the Source editor, which contains the following R code:

```
95  
96 Use the console to calculate other expressions. Standard order of  
   operations applies (mostly), and you can use parentheses `()` as you  
   might expect (but not brackets `[]` or braces `{}`, which have special  
   meanings). Note however, that you must **always** specify multiplication  
   with `*`; implicit multiplication such as `10(3 + 4)` or `10x` will not  
   work and will generate an error, or worse.  
97  
98 ```{r expressions, live = TRUE}  
99 x <- 5.5  
100  
101 x  
102 ```
```

Below the code, the output of the last line is shown:

```
[1] 5.5
```

A text overlay is present in the Source editor:

**R Script or Notebook:**  
Where you are writing and editing  
what you will tell R or Terminal

The Environment pane on the right shows the following values:

Variable	Value
x	5.5

The Files pane on the right shows a directory listing:

Name	Size	Modified
00a-creating_a_github_repo	1.8 KB	Apr 7, 2023
00b-debugging_resources.md	13.6 KB	Apr 7, 2023
00c-good-scientific-coding-pr...	3.9 KB	Apr 7, 2023
01-intro_to_base_R-live.Rmd	19 KB	Apr 7, 2023
02-intro_to_ggplot2-live.Rmd	11.5 KB	Apr 7, 2023
03-intro_to_tidyverse-live.Rmd	21.3 KB	Apr 7, 2023
data		
diagrams		
exercise_01-intro_to_base_R.R...	4.8 KB	Apr 7, 2023
exercise_02-intro_to_R.Rmd	4.7 KB	Apr 7, 2023
exercise_03a-intro_to_tidyvers...	7.7 KB	Apr 7, 2023
exercise_03b-intro_to_tidyvers...	6.2 KB	Apr 7, 2023
README.md	1.8 KB	Apr 7, 2023





rstudio.ccdatalab.org

File Edit Code View Plots Session Build Debug Profile Tools Help

datalab Project: (None)

01-intro\_to\_base\_R-live.Rmd\*

Source Visual [1] 30 Outline

```
95
96 Use the console to calculate other expressions. Standard order of
    operations applies (mostly), and you can use parentheses `()` as you
    might expect (but not brackets `[]` or braces `{}`, which have special
    meanings). Note however, that you must **always** specify multiplication
    with `*`; implicit multiplication such as `10(3 + 4)` or `10x` will not
    work and will generate an error, or worse.
97
98 ```{r expressions, live = TRUE}
99 x <- 5.5
100
101 x
102 ```
```

[1] 5.5

103
104

118:91 # Defining and using variables R Markdown

Console Terminal Jobs

```
R 4.2.3 ~ /
> x <- 5.5
>
> x
[1] 5.5
> |
```

Environment History Connections Tutorial

Import Dataset 172 MiB List

R Global Environment

Values

x	5.5
---	-----

Other Assistance Tabs:  
Things that help you in your coding

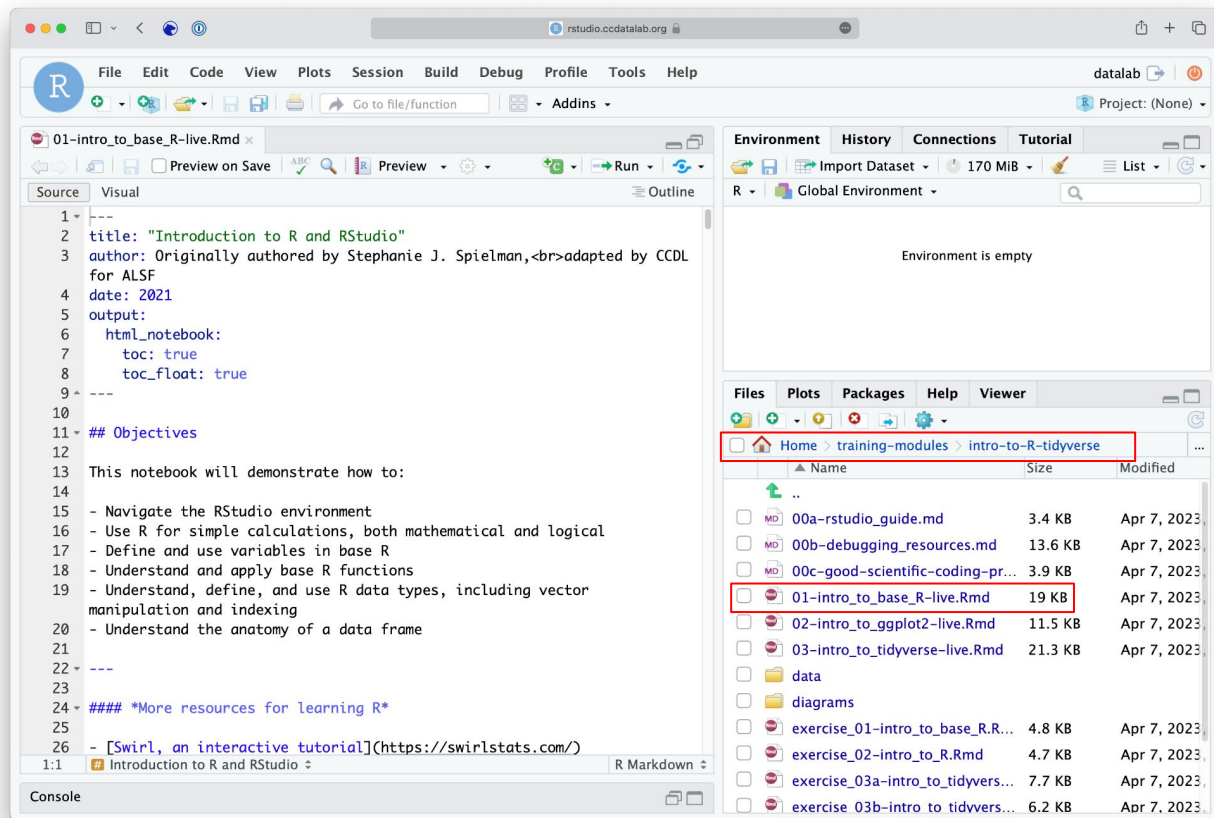
Files Plots Packages Help Viewer

Home > training-modules > intro-to-R-tidyverse

	Name	Size	Modified
	00a-tidyverse_garden.md	3.9 KB	Apr 7, 2023
	00b-debugging_resources.md	13.6 KB	Apr 7, 2023
	00c-good-scientific-coding-pr...	3.9 KB	Apr 7, 2023
	01-intro_to_base_R-live.Rmd	19 KB	Apr 7, 2023
	02-intro_to_ggplot2-live.Rmd	11.5 KB	Apr 7, 2023
	03-intro_to_tidyverse-live.Rmd	21.3 KB	Apr 7, 2023
	data		
	diagrams		
	exercise_01-intro_to_base_R.R...	4.8 KB	Apr 7, 2023
	exercise_02-intro_to_R.Rmd	4.7 KB	Apr 7, 2023
	exercise_03a-intro_to_tidyvers...	7.7 KB	Apr 7, 2023
	exercise_03b-intro_to_tidyvers...	6.2 KB	Apr 7, 2023
	README.md	1.8 KB	Apr 7, 2023

# R Notebooks

Use the "Files" tab to open: `training-modules/intro-to-R-tidyverse/01-intro_to_base_R-live.Rmd`

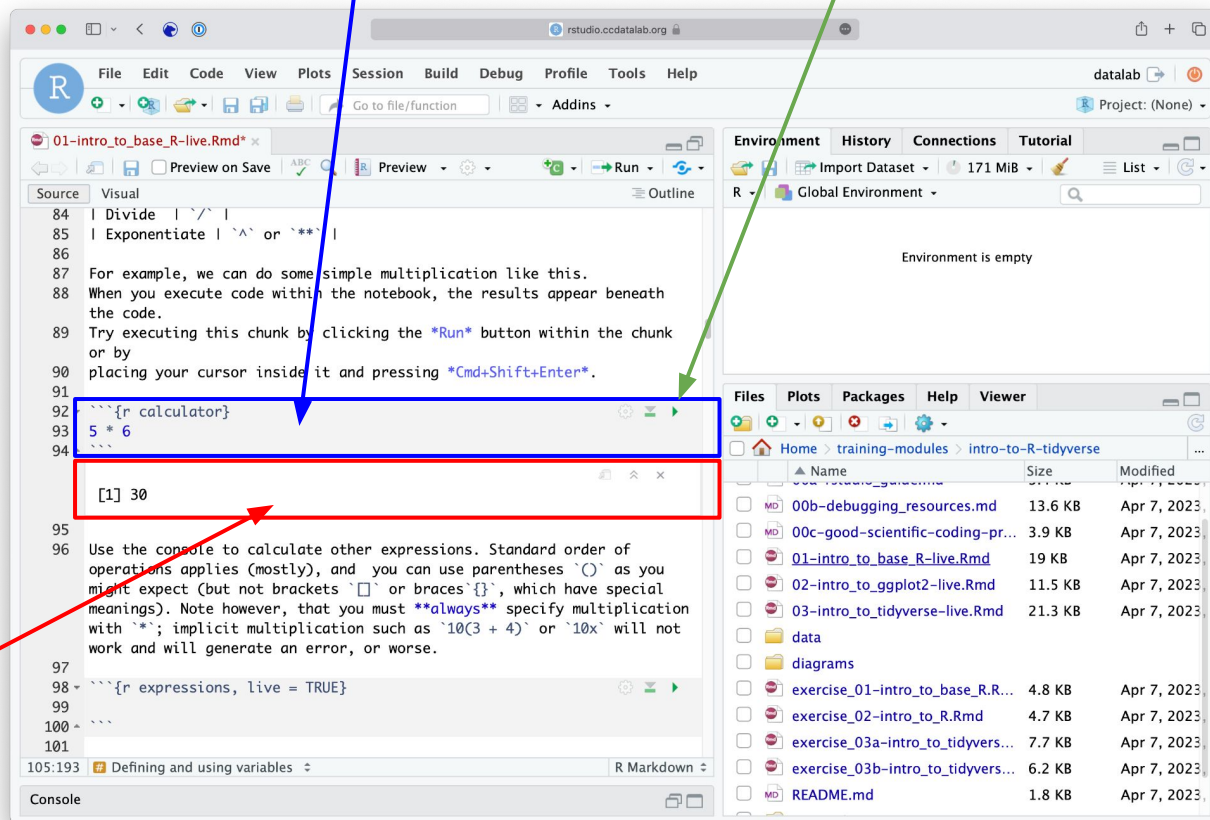


# R Notebooks

R Notebooks allow you to have files that show both your code and results

Executable code chunk

Can click here to run a code chunk

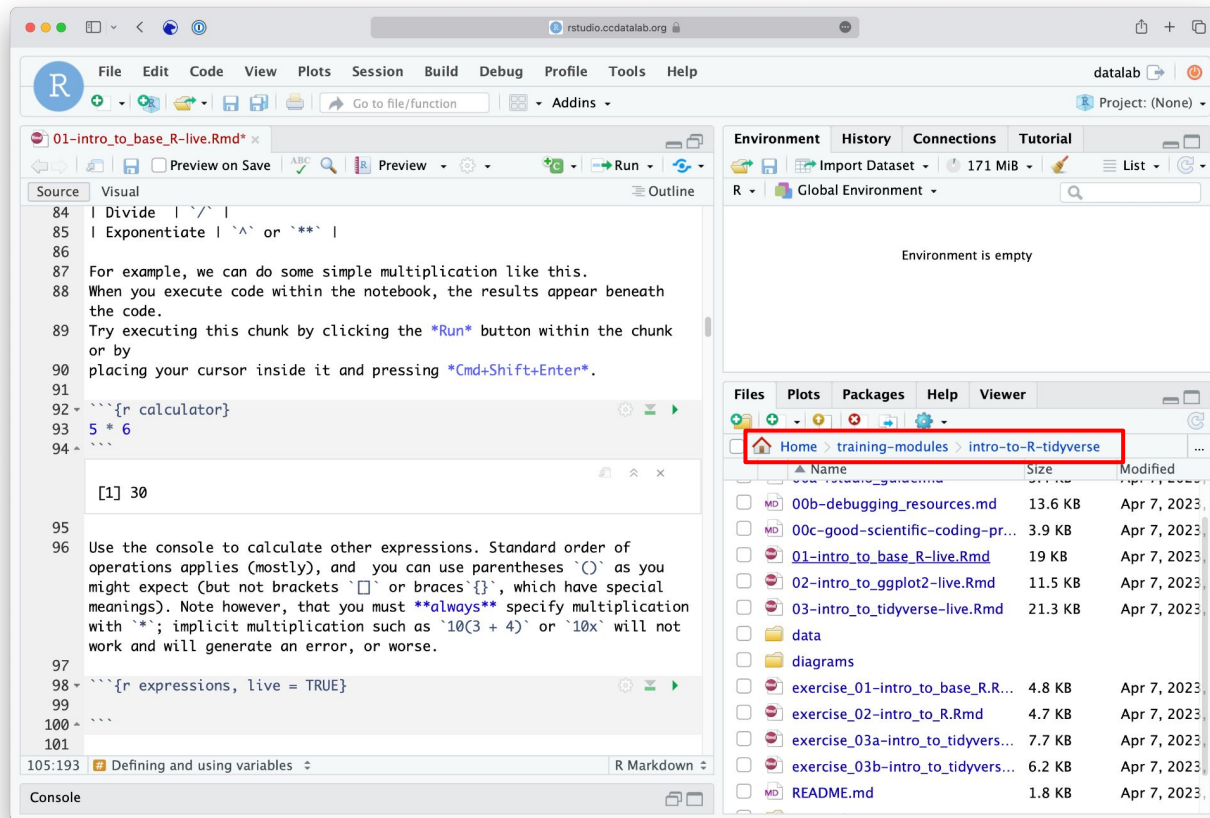


Output from above code chunk



# R Notebooks

- Code that runs in R Notebooks uses wherever the file is saved as its *current directory*
- **Warning!** That may not be the directory shown in the files pane or the console!



# RStudio Sessions

- On the server, R is running many times at once
  - Each user has their own “**Session**” running, with its own memory and processes
- We will usually want to start new sessions between notebooks to keep the environment clean

Log out of website



End the current session and start new session