



Introduction to R, RStudio, and RStudio Server

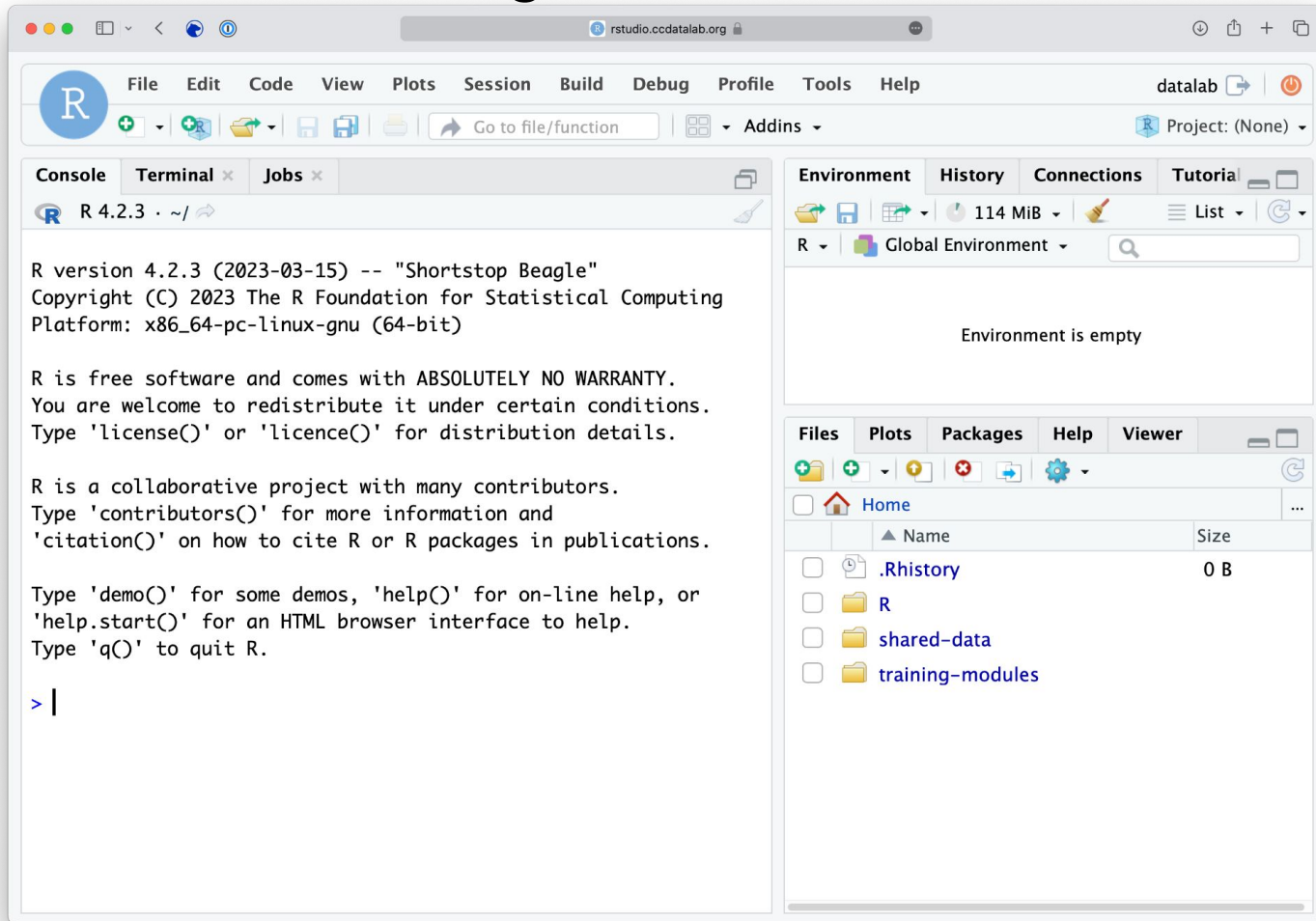
The Data Lab

Powered by Alex's Lemonade Stand Foundation

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



The screenshot displays the RStudio Server web interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The console pane shows the R version 4.2.3 (2023-03-15) and the license information. The environment pane shows the Global Environment is empty. The file pane shows the Home directory with files .Rhistory, R, shared-data, and training-modules.

```
R 4.2.3 . ~/
```

R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

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.

```
> |
```

Environment: 114 MiB, Global Environment

Files: Home

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

(FYI, we're now
at R 4.4.0)

The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The top right shows 'datalab' and 'Project: (None)'. The main area is divided into several panes: Console, Terminal, Jobs, Environment, History, Connections, Tutorial, Files, Plots, Packages, Help, and Viewer. The Console pane is highlighted with a red border and contains the following text:

```
R 4.2.3 . ~/ ↵
```

R version 4.2.3 (2023-03-15) -- "Shortstop Beagle"
Copyright (C) 2023 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (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.

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.

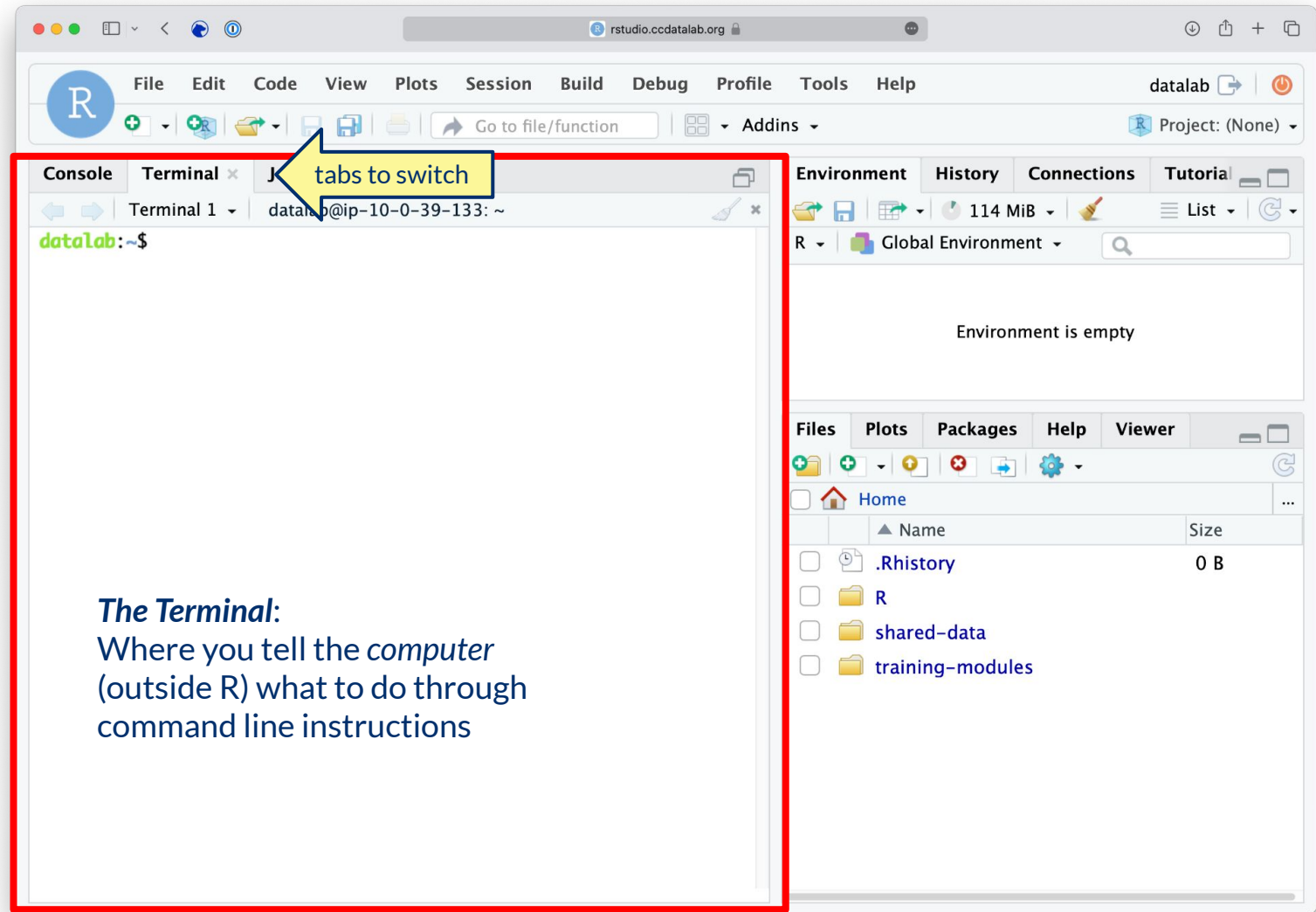
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

```
> |
```

The Console:
where you tell R what to do through
command line instructions

The Environment pane shows 'Global Environment' and 'Environment is empty'. The Files pane shows a file browser with a table of files:

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



The Terminal:

Where you tell the *computer* (outside R) what to do through command line instructions

dataLab:~\$

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

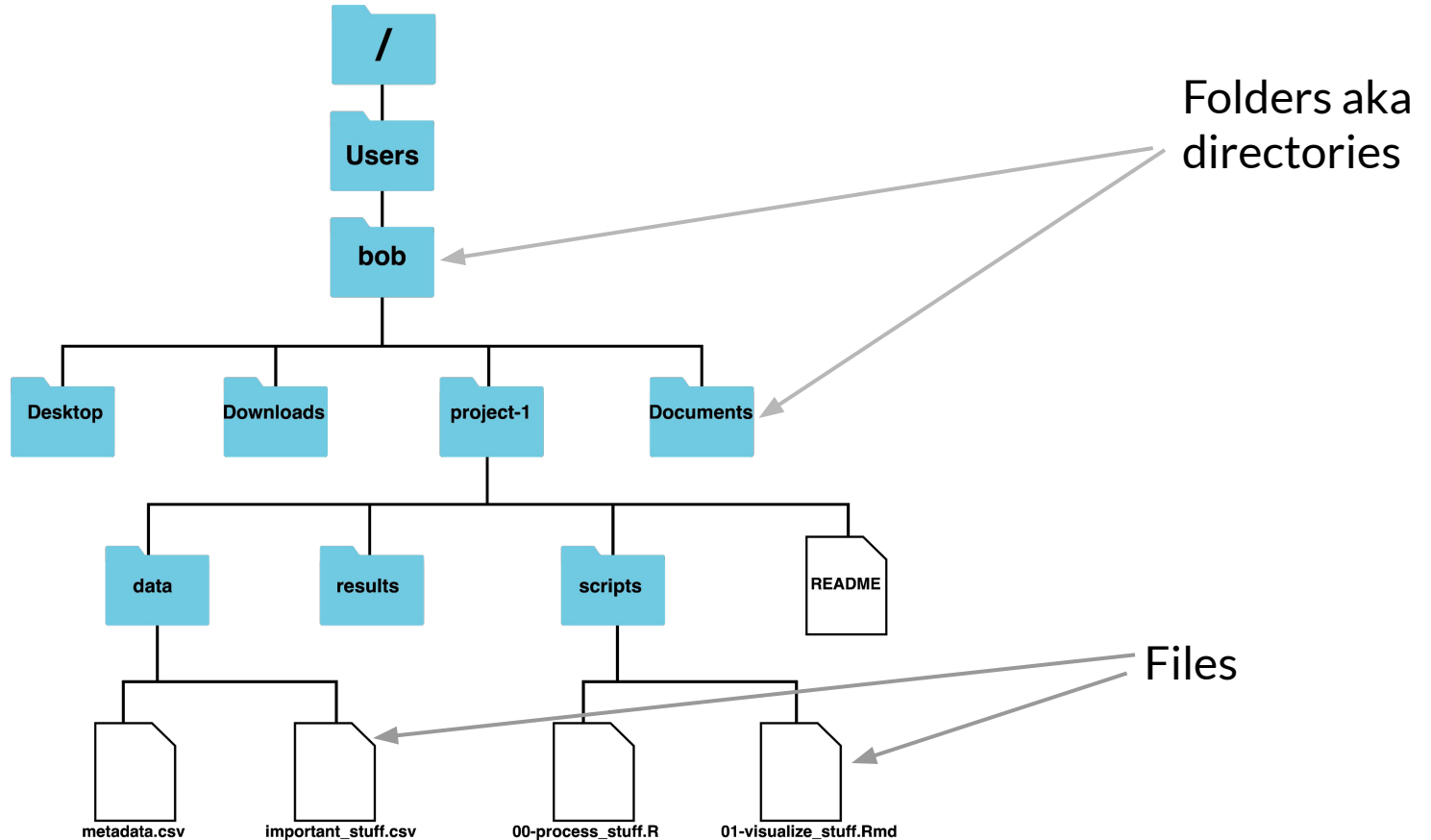
This is called your "**current directory**"

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

The screenshot shows the RStudio interface with the following components:

- Menu bar: File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help
- Toolbar: Go to file/function, Addins
- Project: (None)
- Environment pane: 114 MiB, Global Environment, Environment is empty
- Files pane: Home, .Rhistory (0 B), R, shared-data, training-modules

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.

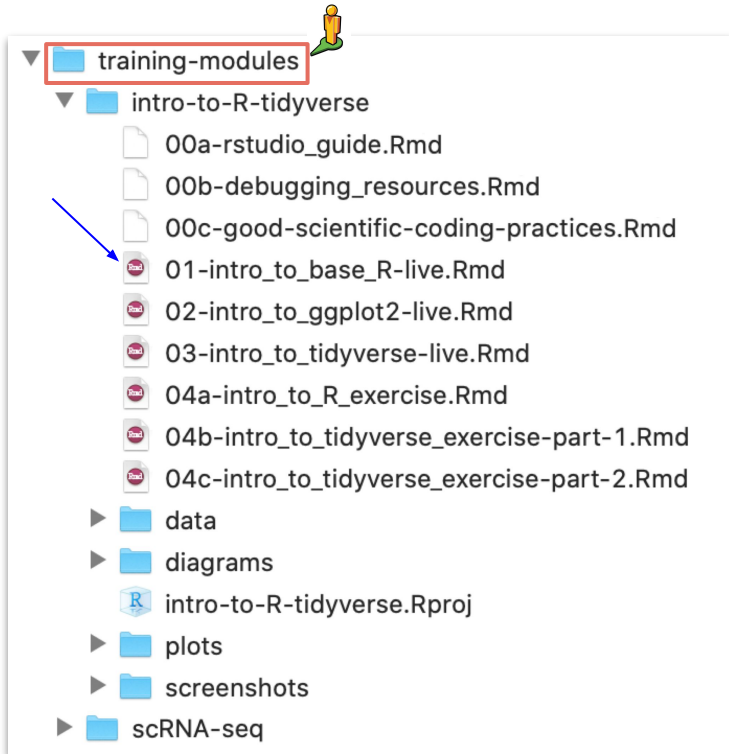
The file we are working on

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`

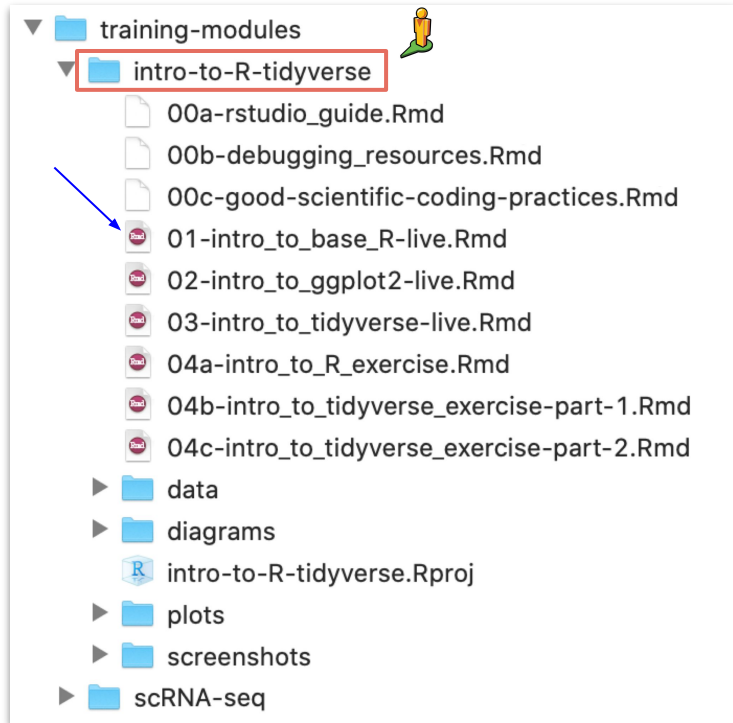


Current/working directory = `training-modules`

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`

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.

The screenshot shows the RStudio interface with the Terminal window highlighted in red. The Terminal prompt is `dataLab:~$`. An arrow points from the tilde character (~) to the text below. The Environment pane shows 'Global Environment' and 'Environment is empty'. The Files pane shows the Home directory with subdirectories: `.Rhistory` (0 B), `R`, `shared-data`, and `training-modules`.

The screenshot shows the RStudio interface with the following components:

- Terminal:** Shows the execution of `ls` and `cd training-modules/` commands. The output of `ls` is `R shared-data training-modules`.
- Environment:** Shows the Global Environment, which is currently empty.
- Files:** Shows the file explorer view of the Home directory, listing `.Rhistory`, `R`, `shared-data`, and `training-modules`.

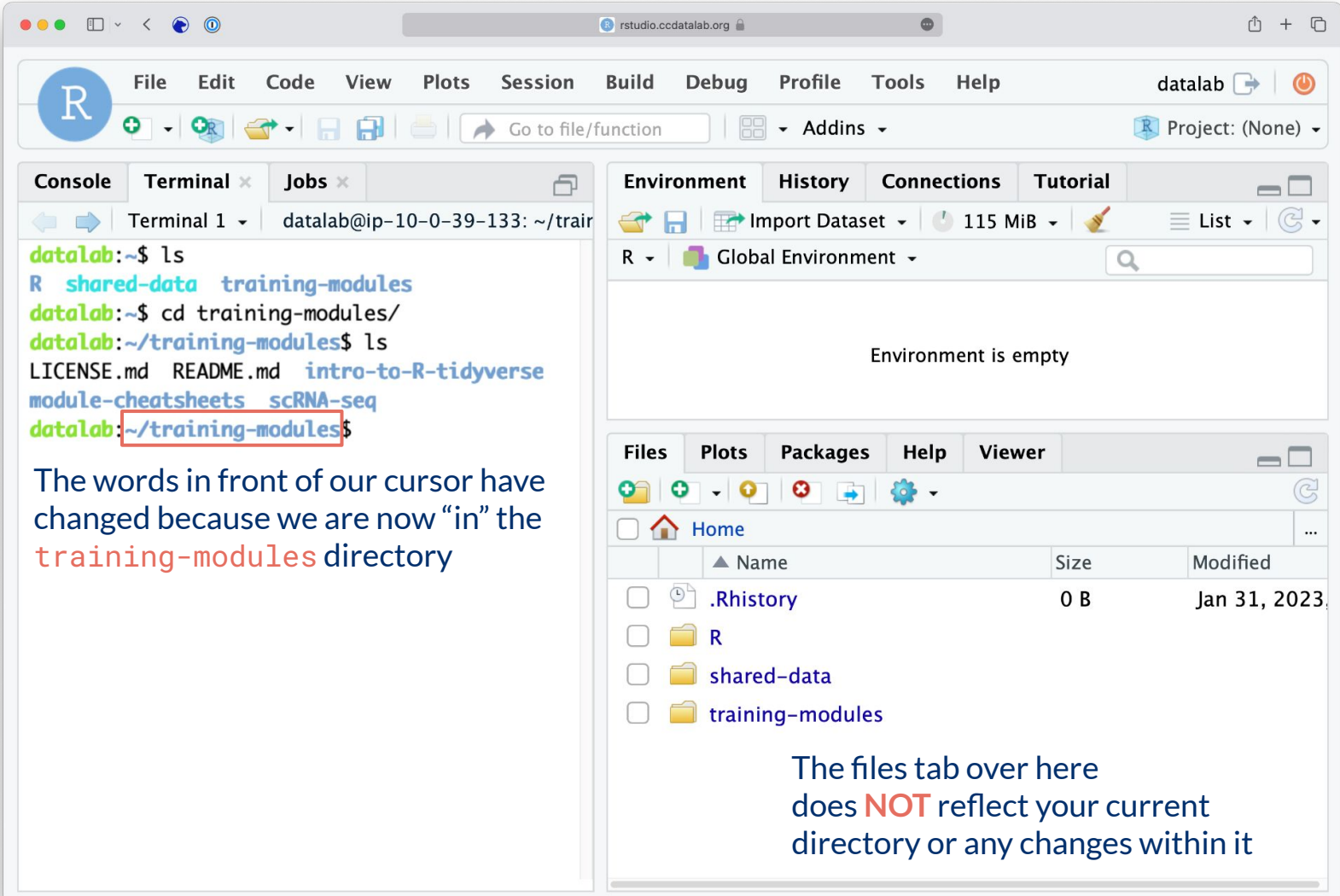
Some common **Terminal** commands:

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

The screenshot shows the RStudio interface with the following components:

- Terminal:** Shows the execution of commands: `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 path `~/training-modules` is highlighted with a red box.
- Environment Panel:** Shows "Global Environment" and "Environment is empty".
- Files Panel:** Shows the file explorer for the current directory, listing: `.Rhistory` (0 B, Jan 31, 2023), `R`, `shared-data`, and `training-modules`.

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





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 image shows the RStudio interface with a notebook titled "01-intro_to_base_R-live.Rmd". The code in the notebook includes instructions on how to run code chunks and a code chunk that calculates $5 * 6$. The console output shows $[1] 30$. The Environment pane is empty. The Files pane shows a directory structure with several R Markdown files. A red box highlights a button in the console area with the text "Click here to show the Console".

```
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 ```
97
98 [1] 30
99
100 Use the console to calculate other expressions. Standard order of
101 operations applies (mostly), and you can use parentheses `(`) as you
102 might expect (but not brackets `[` or braces `{}`, which have special
103 meanings). Note however, that you must **always** specify multiplication
104 with `*`; implicit multiplication such as `10(3 + 4)` or `10x` will not
105 work and will generate an error, or worse.
106
107 ```{r expressions, live = TRUE}
108
109
110
111
```

Environment: Global Environment

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

Name	Size	Modified
00a-intro_to_tidyverse-live.Rmd	19.7 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
exercise_01a-intro_to_tidyvers...	1.7 KB	Apr 7, 2023
exercise_03b-intro_to_tidyvers...	6.2 KB	Apr 7, 2023
README.md	1.8 KB	Apr 7, 2023

Console: Defining and using variables

Click here to show the Console

The screenshot displays the RStudio interface. The main editor shows a script with the following 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 ```
```

The Environment pane on the right shows the variable `x` with the value `5.5`.

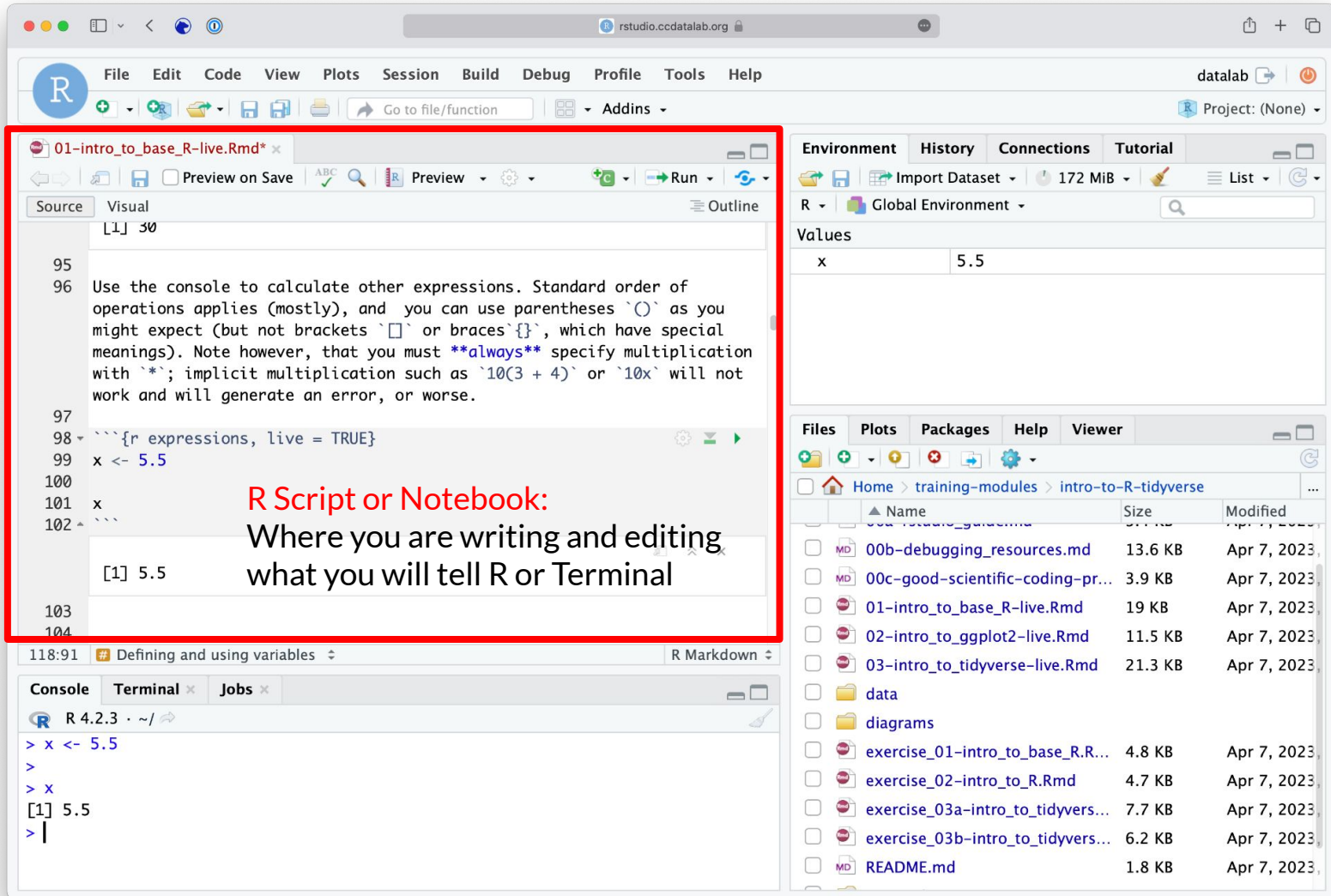
The Files pane on the right shows a directory listing for `intro-to-R-tidyverse`:

Name	Size	Modified
00a-creating_variables.md	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		
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

The Console pane at the bottom, highlighted with a red border, shows the following output:

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

R Console:
What you are actually telling R to do



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

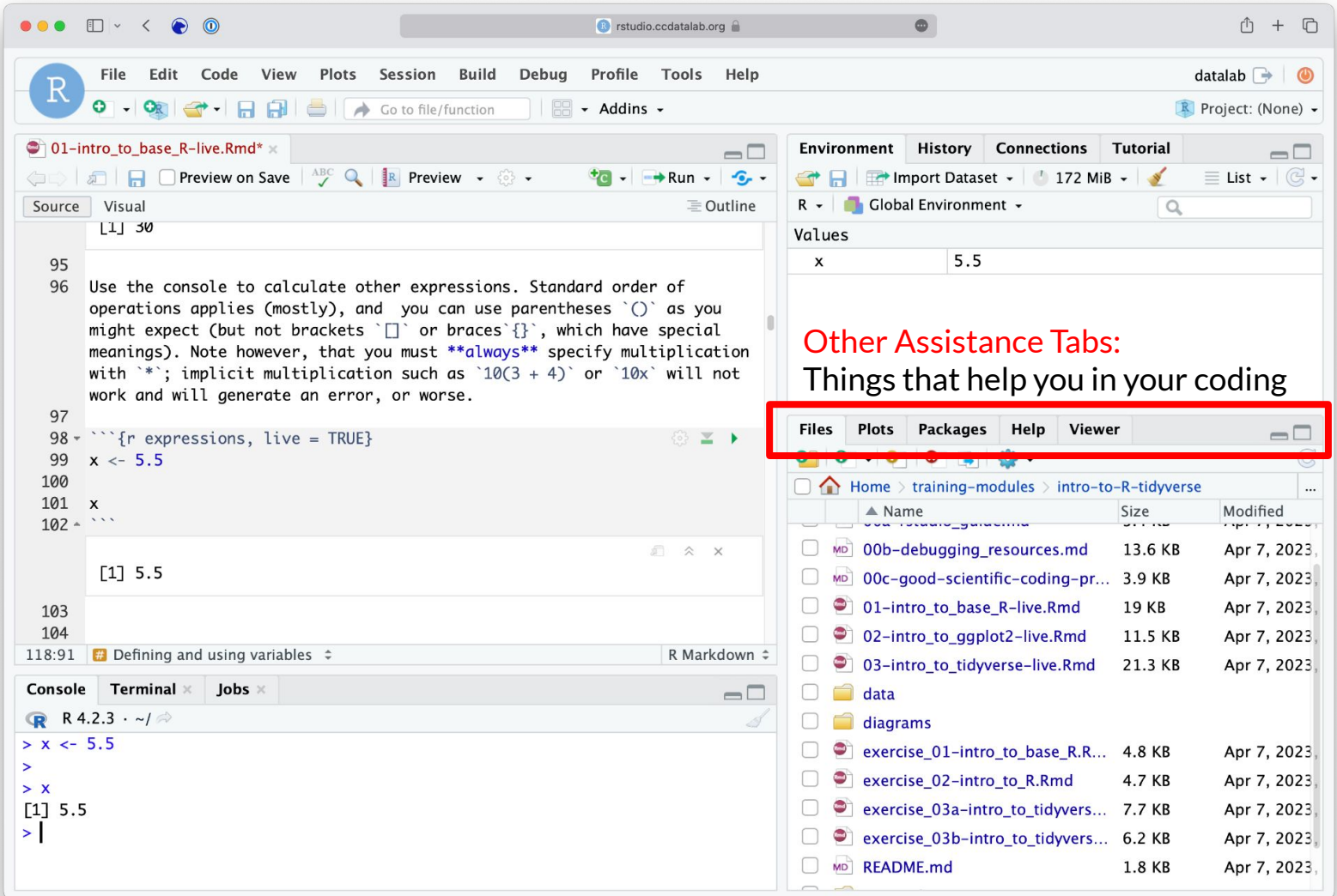
```
95  
96 Use the console to calculate other expressions. Standard order of  
operations applies (mostly), and you can use parentheses `()` as you  
might expect (but not brackets `[]` or braces `{}`, which have special  
meanings). Note however, that you must **always** specify multiplication  
with `*`; 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 4.2.3 ~/  
> x <- 5.5  
>  
> x  
[1] 5.5  
> |
```

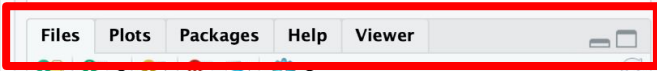
Environment History Connections Tutorial
Import Dataset 172 MiB
R Global Environment
Values
x 5.5

Files Plots Packages Help Viewer
Home > training-modules > intro-to-R-tidyverse

Name	Size	Modified
00a-tidyverse_guidance		
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

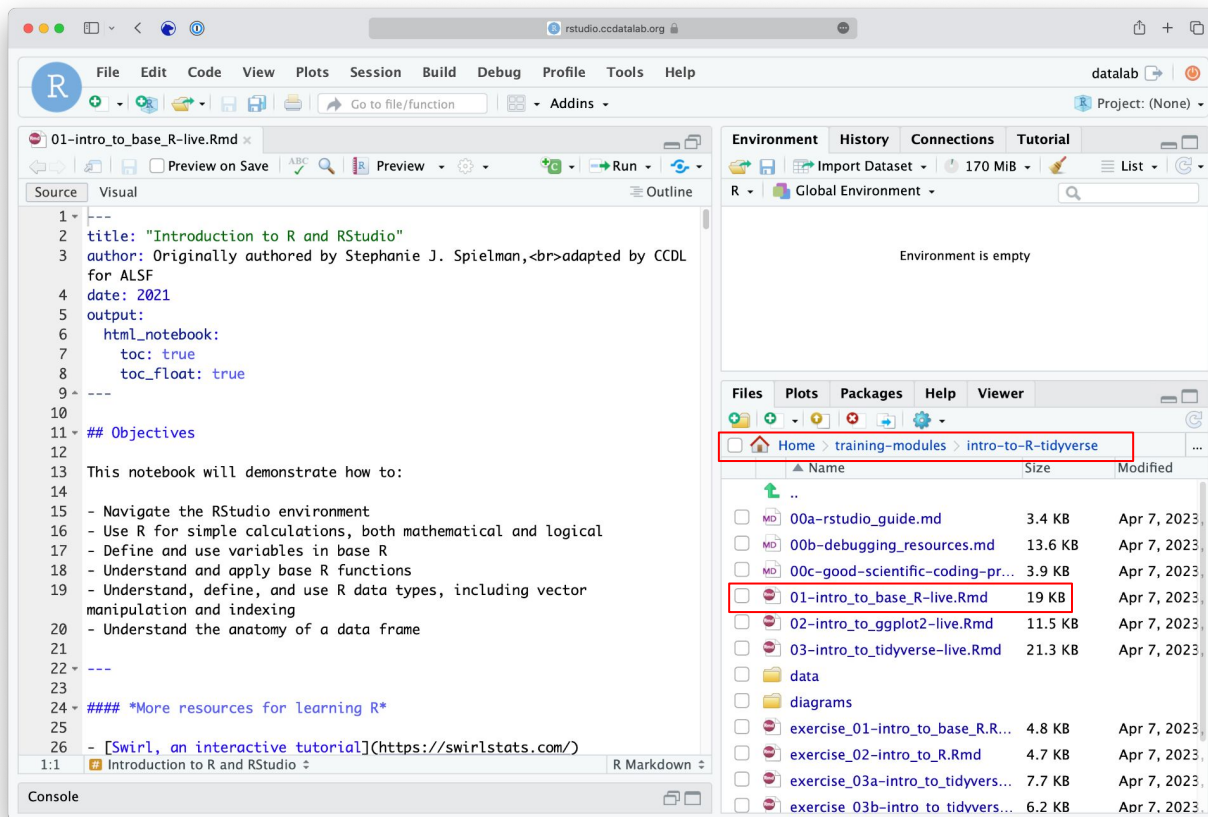


Other Assistance Tabs:
Things that help you in your coding



R Notebooks

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



The screenshot displays the RStudio interface. The main editor shows the following R Notebook content:

```
1 ---  
2 title: "Introduction to R and RStudio"  
3 author: Originally authored by Stephanie J. Spielman, <br> adapted by CC DL  
4 for ALSF  
5 date: 2021  
6 output:  
7   html_notebook:  
8     toc: true  
9     toc_float: true  
10 ---  
11 ## Objectives  
12  
13 This notebook will demonstrate how to:  
14  
15 - Navigate the RStudio environment  
16 - Use R for simple calculations, both mathematical and logical  
17 - Define and use variables in base R  
18 - Understand and apply base R functions  
19 - Understand, define, and use R data types, including vector  
20   manipulation and indexing  
21 - Understand the anatomy of a data frame  
22 ---  
23  
24 #### *More resources for learning R*  
25  
26 - [Swirl, an interactive tutorial](https://swirlstats.com/)
```

The right-hand pane shows the "Files" tab with the following file browser view:

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

Name	Size	Modified
..		
00a-rstudio_guide.md	3.4 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_tidvers...	7.7 KB	Apr 7, 2023
exercise_03b-intro_to tidvers...	6.2 KB	Apr 7, 2023

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

The screenshot shows the RStudio interface with an R Notebook open. The notebook content includes:

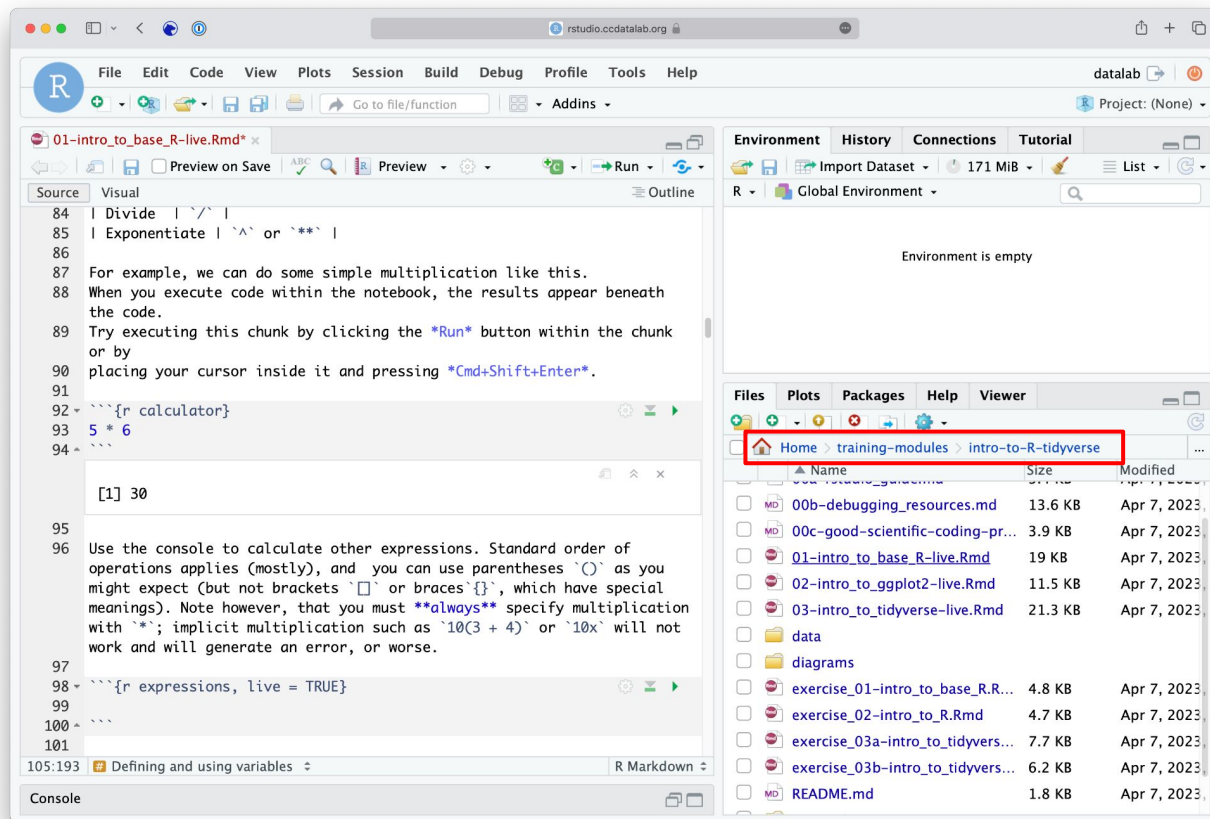
```
84 | Divide | \ / |
85 | Exponentiate | ^ or ** |
86
87 For example, we can do some simple multiplication like this.
88 When you execute code within the notebook, the results appear beneath
89 the code.
90 Try executing this chunk by clicking the *Run* button within the chunk
91 or by
92 placing your cursor inside it and pressing *Cmd+Shift+Enter*.
93
94 {r calculator}
95 5 * 6
96
97 [1] 30
98
99 Use the console to calculate other expressions. Standard order of
100 operations applies (mostly), and you can use parentheses `()` as you
101 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.
102
103 {r expressions, live = TRUE}
104
105:193 Defining and using variables
```

The code chunk containing `5 * 6` is highlighted with a blue box, and the output `[1] 30` is highlighted with a red box. A blue arrow points from the text "Executable code chunk" to the code chunk, and a green arrow points from "Can click here to run a code chunk" to the Run button in the code chunk toolbar. A red arrow points from "Output from above code chunk" to the output box.

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