

Foundations of Data Science for Biologists

Class Orientation and Introduction to R

BIO 724D

28-AUG-2023

Instructors: Greg Wray, Paul Magwene, and Jesse Granger

Orientation to BIO724D

What is BIO 724D?

A class that introduces how to work with data to biologists over two semesters

- Practical skills for wrangling, processing, filtering, and visualizing data

- Best practices for analysis, interpretation, reproducibility, and reusability of data

Assumes no prior programming experience

A class that builds community

- Drawing on data sets generated by current and former biology grad students

- Fostering discussions with current biology trainees, staff, and faculty

What is BIO 724D?

A class that introduces how to work with data to biologists over two semesters

- Practical skills for wrangling, processing, filtering, and visualizing data

- Best practices for analysis, interpretation, reproducibility, and reusability of data

Assumes no prior programming experience

A class that builds community

- Drawing on data sets generated by current and former biology grad students

- Fostering discussions with current biology trainees, staff, and faculty

Not

- An introduction to statistics (though we will often use statistical methods)

- An introduction to computer science (though it might feel like it occasionally)

Course structure

Lecture / lab: Mondays 3:05-5:05pm, 2237 FFSC

Semi-flipped format: complete reading / video assignments *before* class

Class sessions a mix of traditional lecture and hands-on engagement

Hands-on component a mix of individual follow-along and group exercises

Data lunch: Thursdays 12:00-1:00pm, 248 BioSci (“Cogitron”)

Presentations by trainees, staff, and faculty over lunch

Discussions / questions / interaction encouraged

Course structure

Lecture / lab: Mondays 3:05-5:05pm, 2237 FFSC

Semi-flipped format: complete reading / video assignments *before* class

Class sessions a mix of traditional lecture and hands-on engagement

Hands-on component a mix of individual follow-along and group exercises

Data lunch: Thursdays 12:00-1:00pm, 248 BioSci (“Cogitron”)

Presentations by trainees, staff, and faculty over lunch

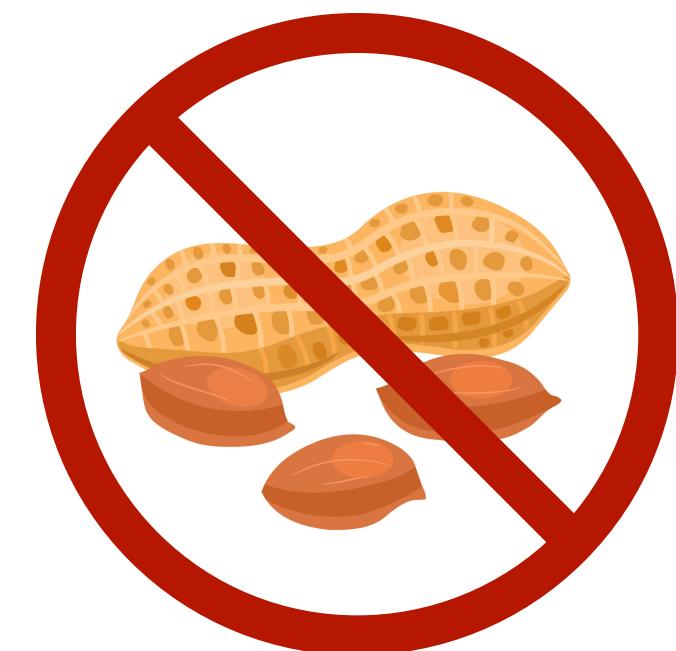
Discussions / questions / interaction encouraged



be respectful



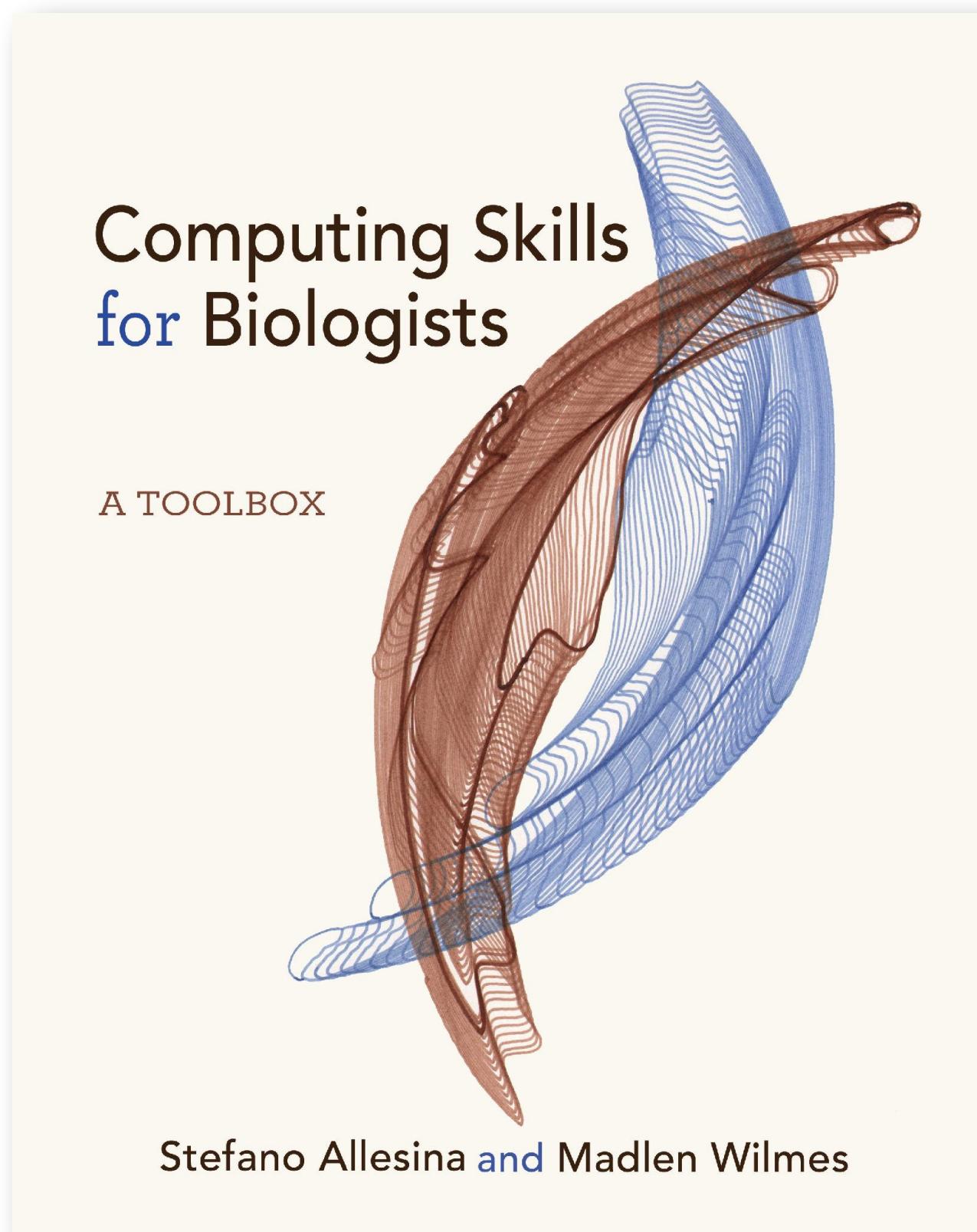
be polite



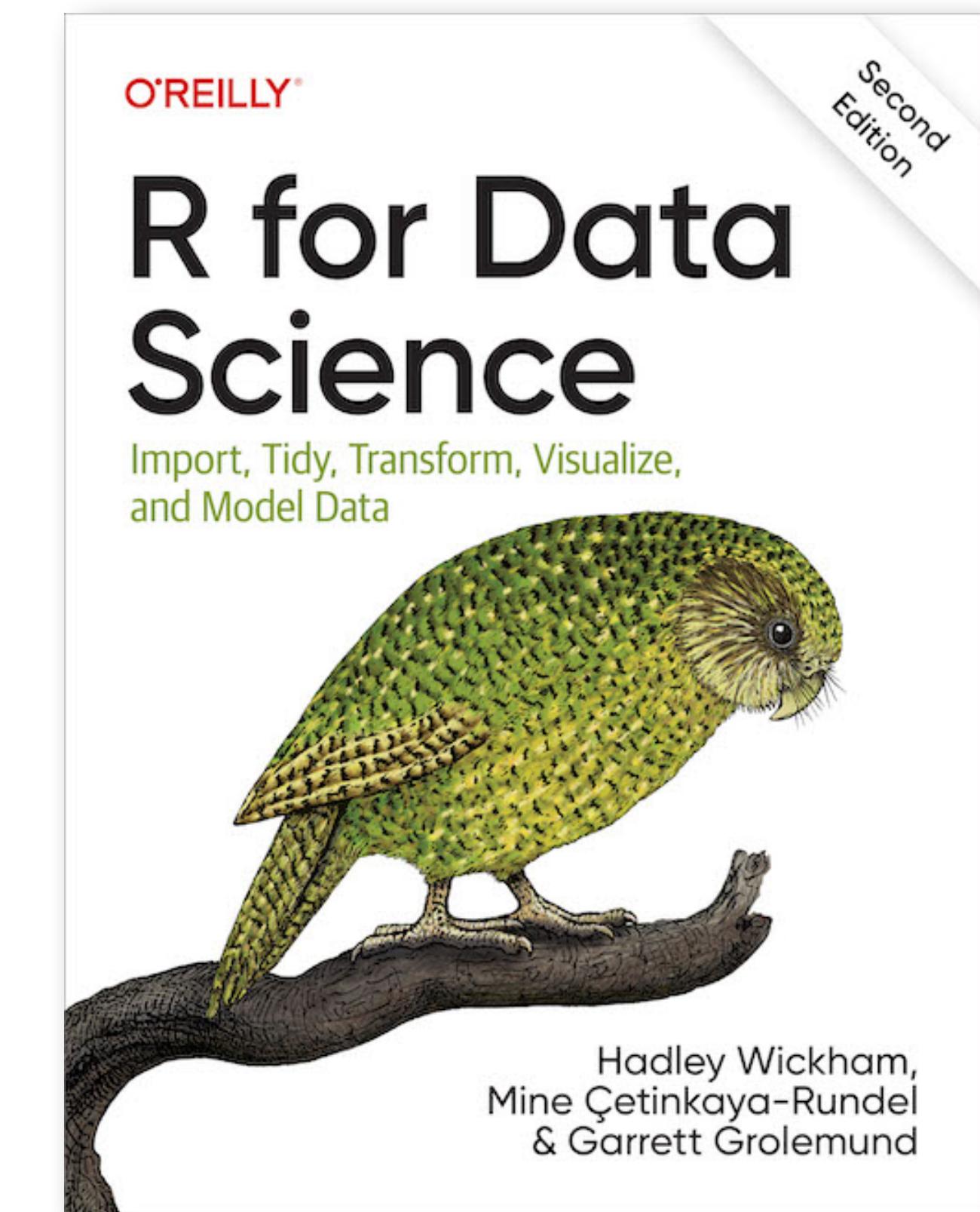
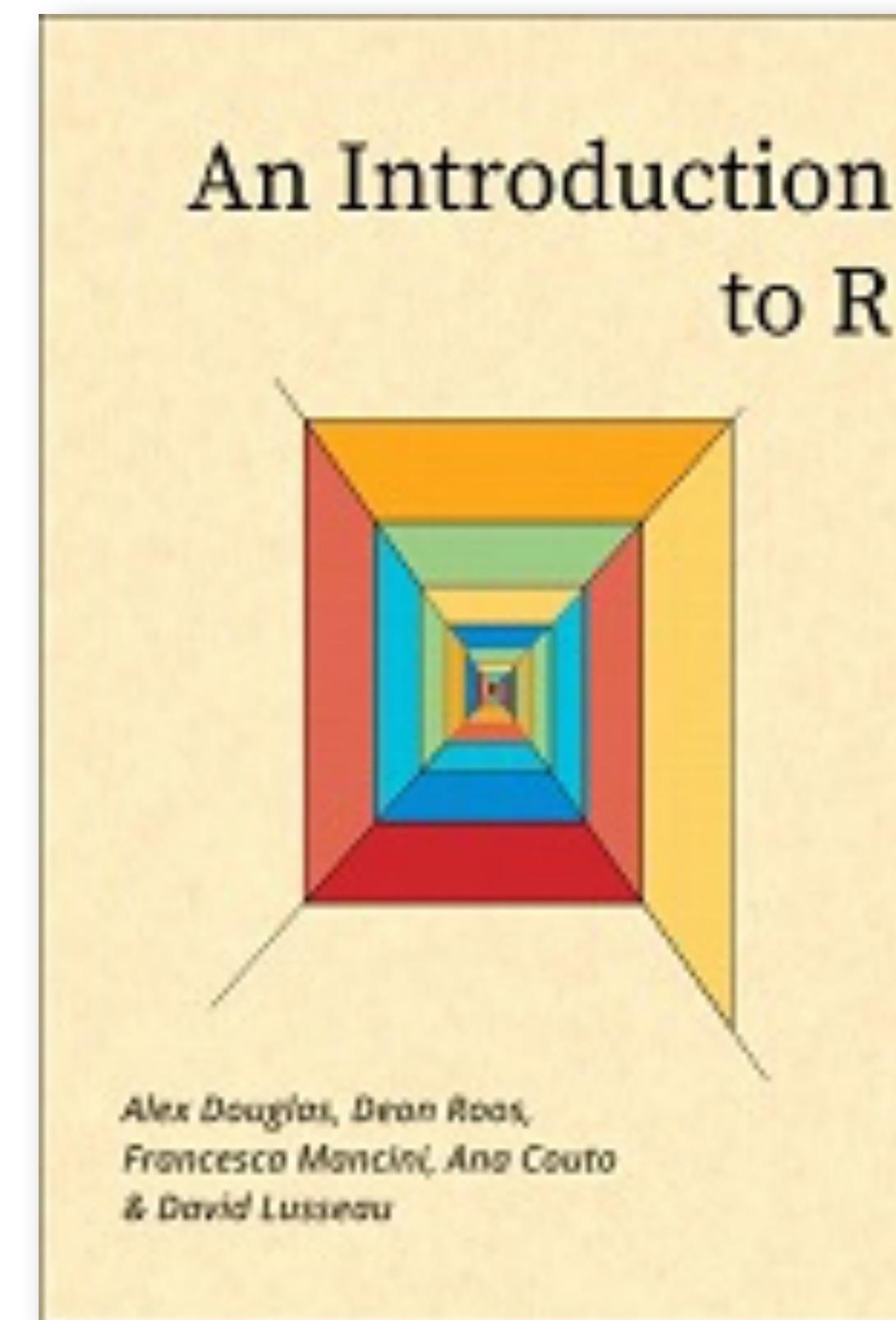
no peanuts

Course materials

Throughout the year:



For the R portion of the course:



All are available electronically for free: see the course wiki

Practicalities

Wiki: the hub for most things related to the course

- Schedule of lecture and lunch topics

- Information about course logistics, grading, and policies

- Links to preparation materials: readings and videos

- Links to slide decks

- Assignments: weekly problem sets and project

- Links to resources: help with learning, reference material, deeper dives

Sakai: where you turn in assignments

- Upload files into your folder under the “Assignments” tab

- Note the file naming convention in assignment instructions

Expectations

Adhere to the Duke Community Standard

Attend every class session and actively engage

Come to Monday sessions prepared

Submit homework and projects on time

Stay on top of the material and ask questions

Familiarize yourself with the information on the course wiki

Assignments

Data lunch “exit tickets” (25%)

Short description of new concept/method/application you learned

Due at midnight of same day; graded for completion

Problem sets (75%)

Posted by class time each Monday; due the following Sunday at midnight

Graded for completion: code must run or include an explanation

Projects

Posted on 16 Nov, due 3 Dec

Graded for completion; counts as 2 problem sets

Learning notebook (5%)

Markdown file of notes, code snippets, links, etc.; due at end of the semester

Seeking help

Seek help when you are stuck!

Textbooks, Google, YouTube, LinkedIn Learning (free!)

Built-in help resources, R vignettes, code snippets from official documentation

Classmates (class Slack channel), friends, lab-mates, other grad students

On-line forums: StackOverflow and others

AI resources: ChatGPT, BingGPT, GPT4, etc.

When turning in your homework, cite your sources and explain how they helped you

E.g., Code for removing empty rows modified from a post on StackOverflow

E.g., Use of Itertools library for permutation suggested by a lab mate

For humans, use generic attributions rather than personal names

For non-humans, cite specific sources

Code that isn't working

There may be occasions when you can't get your code to work properly: no problem!!

You will still get **full** credit if:

- You turn your homework in on time

- Identify specifically what is and is not working correctly

- Explain what you tried to do to fix the problems

You will **not** get full credit if:

- You turn your assignment in late without prior arrangement

- You neglect to reveal and explain what isn't working properly (see above)

Test your code carefully and be transparent about any issues you can't fix!

What can you expect from BIO 724D?

You will learn concepts and skills that will:

- Be useful throughout your research career
- Save you *lots* of time and frustration
- Reduce errors in your work
- Make it easier for you to collaborate

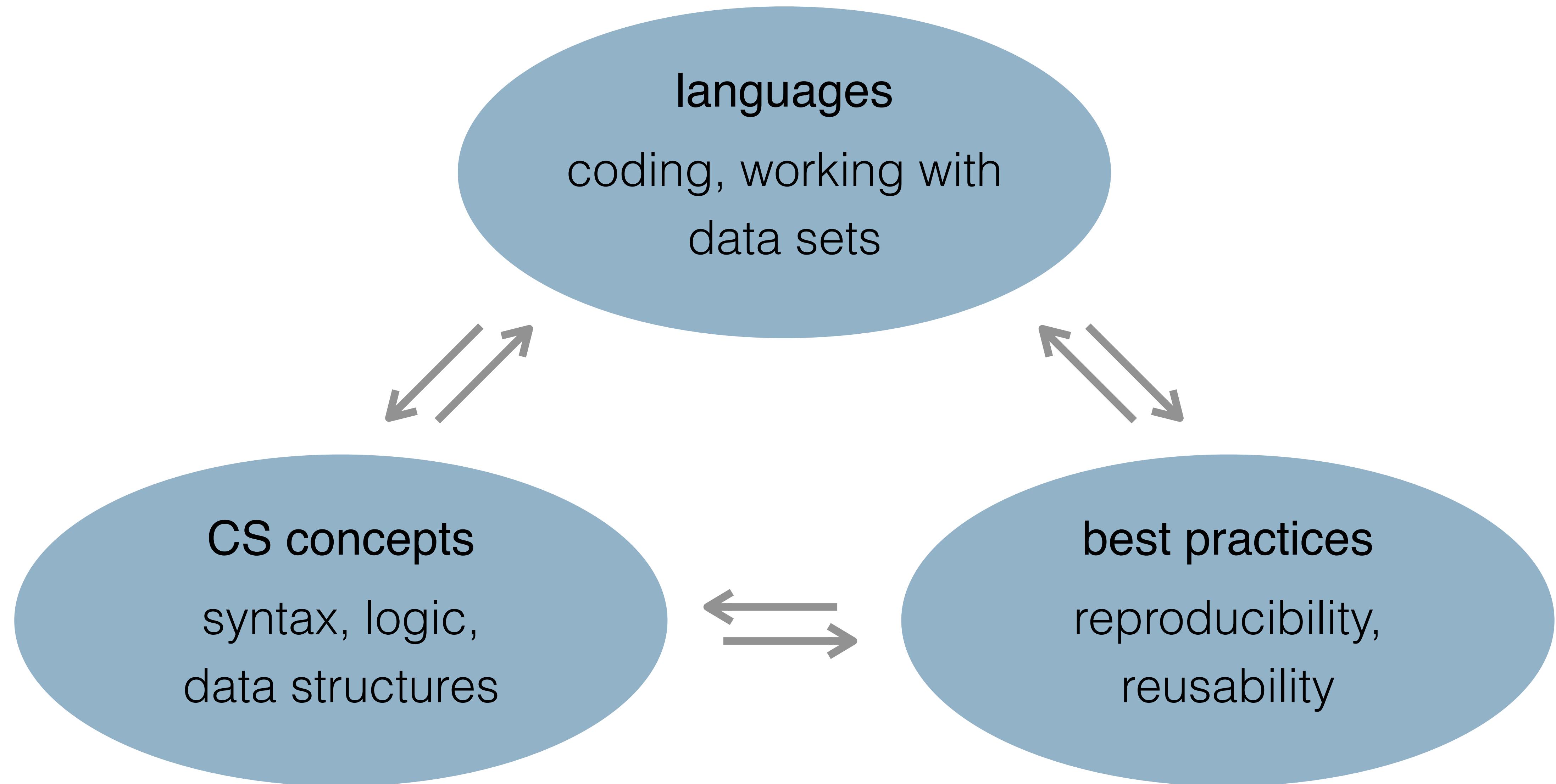
This comes at a cost

- You will need to practice, practice, practice
- You may feel frustrated at times, but persevere and it will be worth it!

We will strive to make this process as smooth as possible

- Maintain a safe, respectful, and constructive learning environment
- Encourage questions and foster curiosity (no “dumb” questions)
- Provide support and resources to help you thrive and learn
- 2-semester format is intended to allow you to absorb information and reduce stress

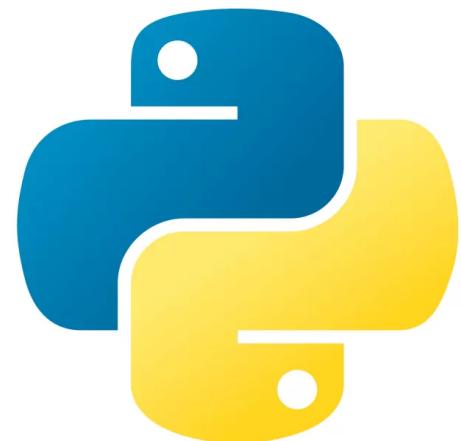
Learning objectives



Languages we will work with



UNIX



	<i>strengths</i>	<i>weaknesses</i>
R	<p>Designed for statistics and visualization Good for numerical modeling Widely used in biological research</p>	<p>Awkward for tasks outside its niche Size of data sets limited by RAM Library ecosystem can be confusing</p>
UNIX	<p>Nearly universal computing environment Extremely powerful, uniquely capable Most future-proof way to code</p>	<p>Terse syntax, not easily readable Does not provide much feedback Designed to work primarily with text</p>
Python	<p>General-purpose high-level language Readable code, consistent syntax Among most widely used languages</p>	<p>Relative to some other GPLs: Slower execution Less memory-efficient</p>
SQL	<p>Designed for relational databases Extraordinarily powerful in this role Dominant database environment</p>	<p>Nearly useless outside its niche Procedural programming is awkward Not as flexible as some other QLs</p>

Learning objectives: the fine print

How to work with data:

Format data to make it is easy for you (and for others!) to work with

Identify missing data, outliers, improperly and inconsistently formatted data

Fix all of the above gracefully, accurately, and reproducibly

How to write code that:

Does what you want it to do — and how to verify this

Saves you time and effort, reduces errors, and improves reproducibility

Can be re-purposed for other tasks, even unrelated ones

Makes sense to future-you and to others

Learning objectives: more fine print

How to produce visuals that:

- Allow you to explore and understand complex data sets with ease

- Portray your results to others with integrity, clarity, and accuracy

- Are customized to your precise specifications

How to adhere to FAIR data standards for dissemination of results:

- Findable: store data and metadata in a repository with a persistent identifier

- Accessible: adhere to free and open data standards

- Interoperable: metadata allow datasets to be combined

- Reproducible: code, data, and metadata are richly described

Preview of topics

session	topics
1 - 7	R: importing, transforming, querying, and displaying data
8 - 11	Unix: working with files at the command line; regular expressions
12	fall semester wrap-up
13 - 18	Python: general-purpose programming
19 - 20	SQL: designing and querying relational databases
21 - 22	best practices: using git, graphical integrity, FAIR principles
23 - 24	domain-specific modules
25	course wrap-up

File systems: A brief overview

Why discuss file systems?

Understanding where your data lives and what type of data it is

Modern operating systems obscure or hide important information about files, folders, and their structure

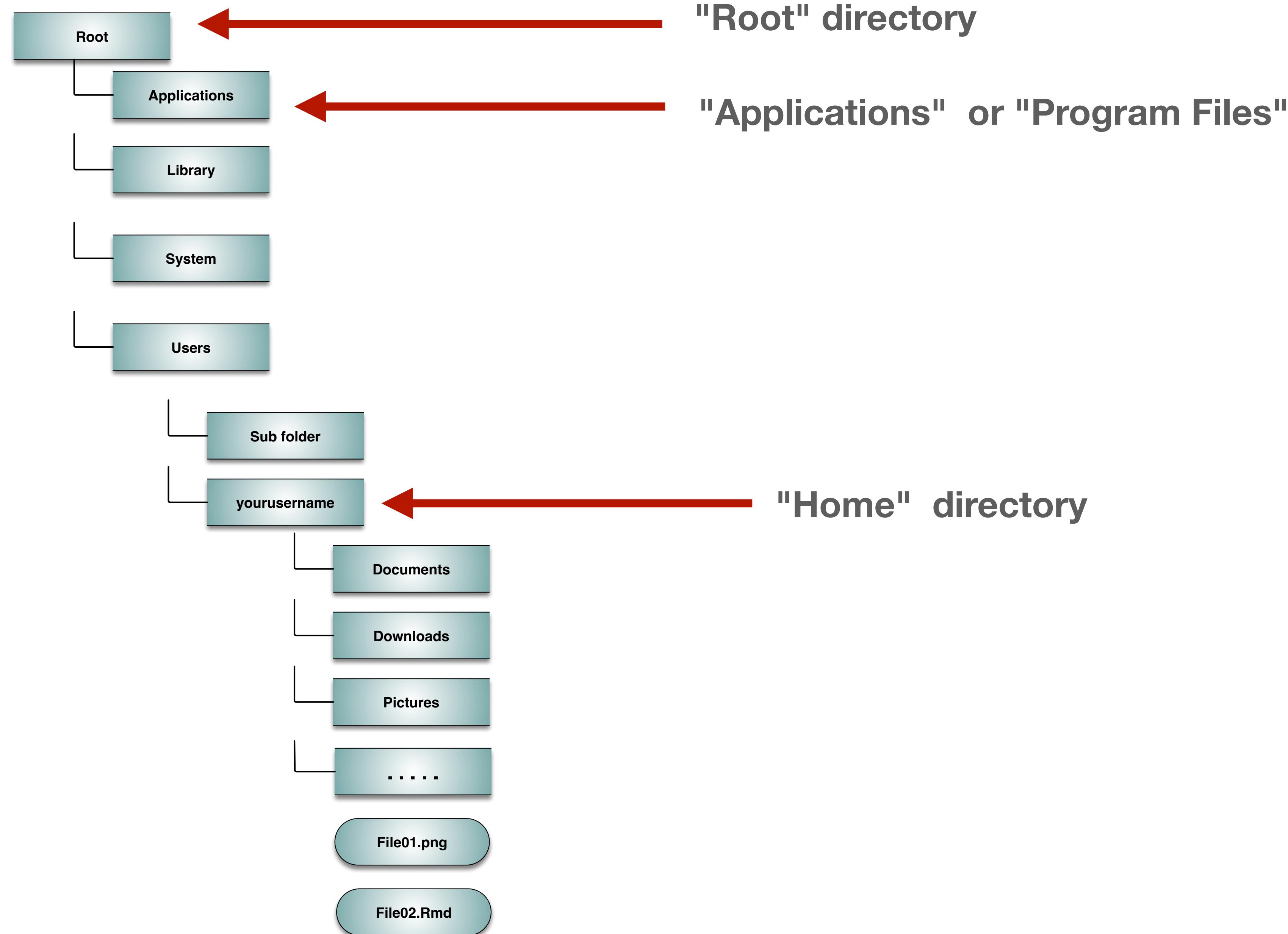
Access

Most of the tools we will use in this course, and that you will use in your research careers, assume you know the conventions for specifying file paths for the data you are analyzing

Data backup

Knowing where your data lives, and the form it is stored in, is critical for data backup and retention

Heirarchical file systems

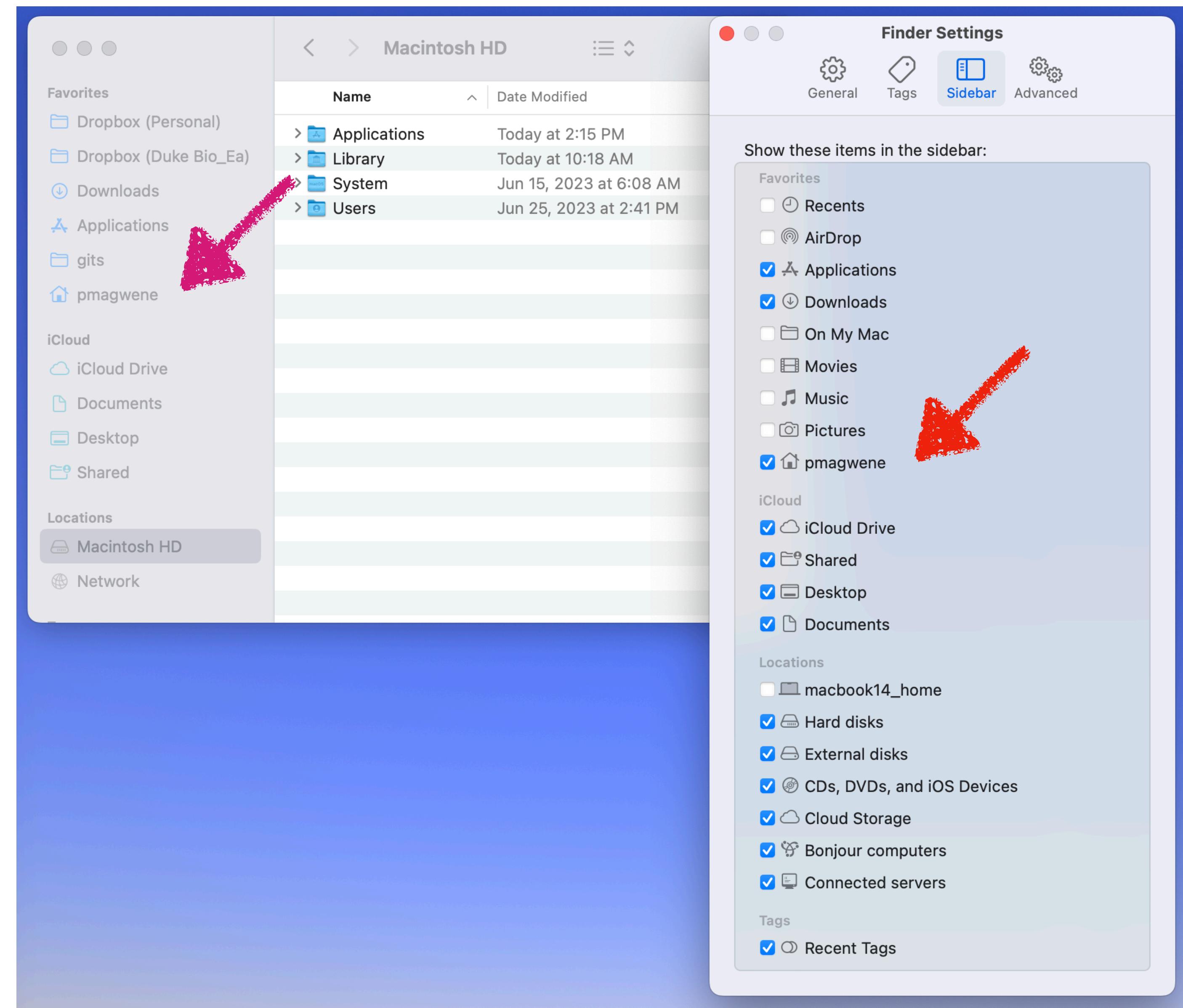


MacOS and Windows File System Conventions

	MacOS	Windows
Root directory	/	C:\
Applications directory	/Applications	C:\Program Files C:\Program Files (x86)
User (home) directory	/Users/yourusername	c:\Users\yourusername
Desktop, Downloads, Documents	Live in home directory	Live in home directory

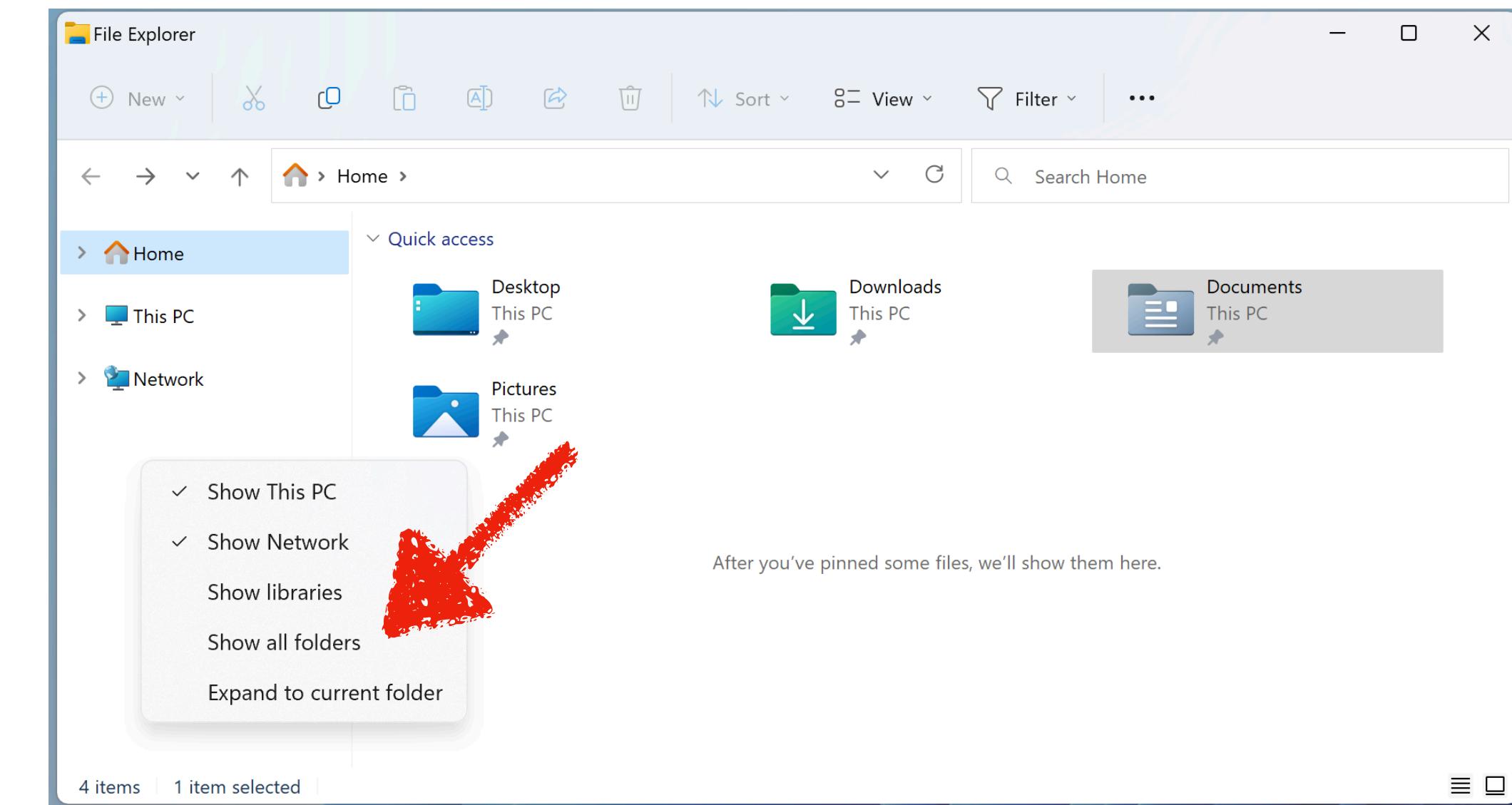
Usability recommendation: Show your user directory in the sidebar of Finder

Finder >
Settings... >
Sidebar >
Click the item with the
Home icon and your username

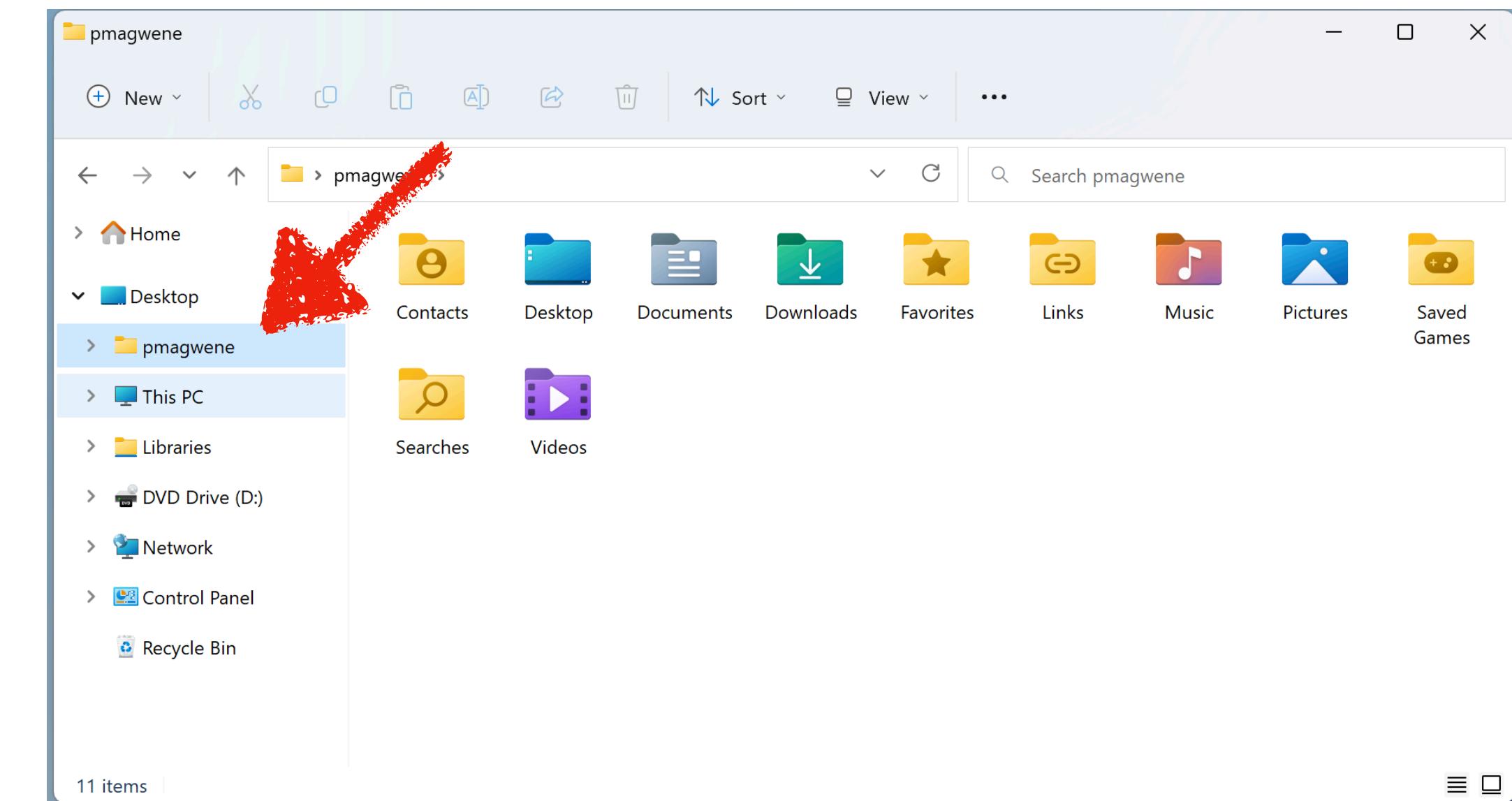


Usability recommendation: Show your user directory in the sidebar (navigation pane) of File explorer

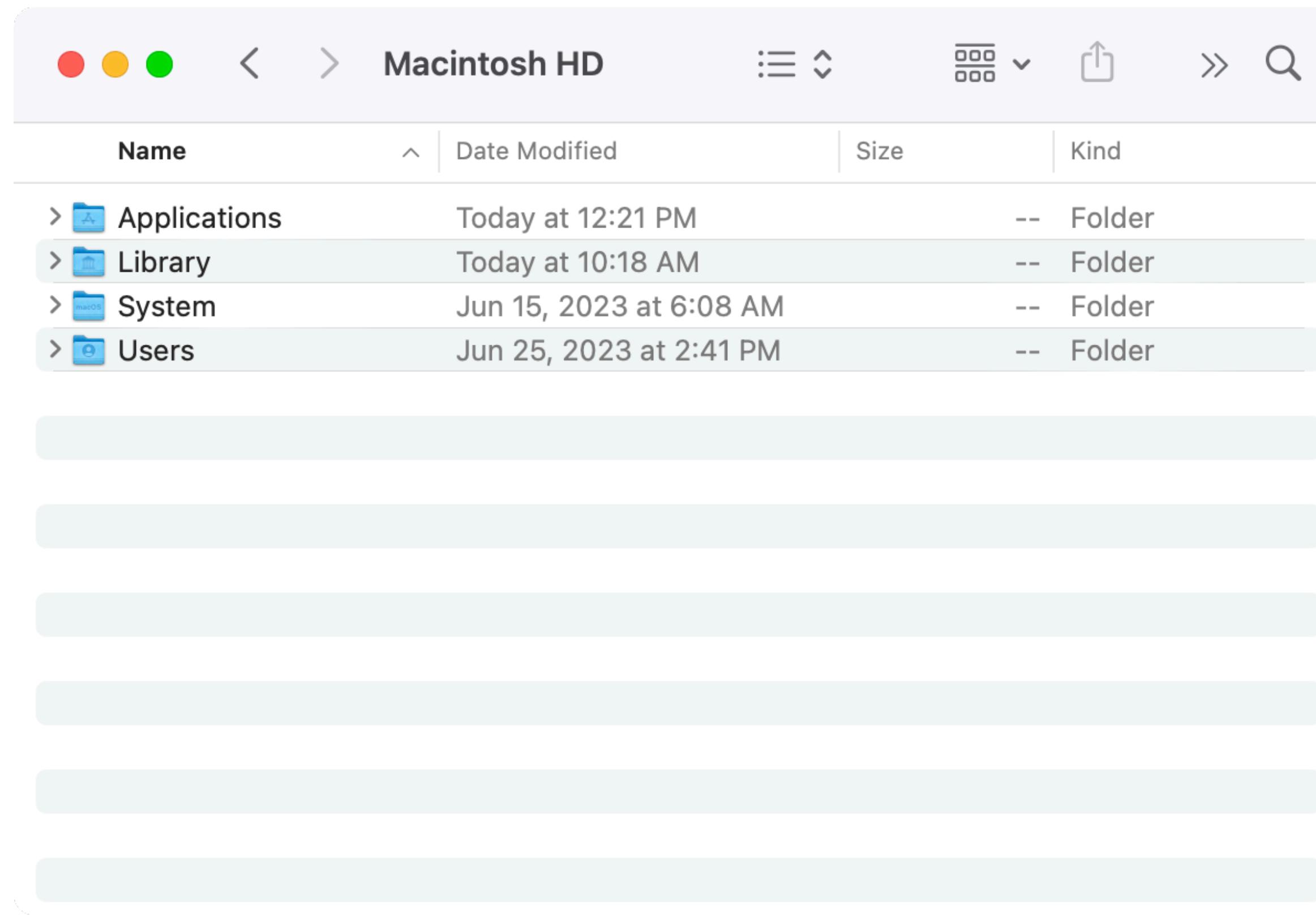
Right click in empty region
on sidebar >
Choose "Show all folders"



A shortcut to your user
directory will show up in the
sidebar



Hidden files and folders: The Finder (MacOS) and File Explorer (Windows) are lying to you



What the Finder tells me is in my Root directory

Name	Date Modified	Size	Kind
.file	Jun 15, 2023 at 6:08 AM	Zero bytes	Document
> .vol	Jun 15, 2023 at 6:08 AM	--	Folder
> .VolumeIcon.icns	Jun 15, 2023 at 6:08 AM	36 bytes	Alias
> Applications	Today at 2:15 PM	--	Folder
> bin	Jun 15, 2023 at 6:08 AM	--	Folder
> cores	Feb 9, 2023 at 4:39 AM	--	Folder
> etc	Jun 15, 2023 at 6:08 AM	11 bytes	Alias
> home	Today at 2:28 PM	25 bytes	Alias
> Library	Today at 10:18 AM	--	Folder
> opt	May 18, 2023 at 2:14 PM	--	Folder
> private	Today at 2:28 PM	--	Folder
> sbin	Jun 15, 2023 at 6:08 AM	--	Folder
> System	Jun 15, 2023 at 6:08 AM	--	Folder
> tmp	Jun 15, 2023 at 6:08 AM	11 bytes	Alias
> Users	Jun 25, 2023 at 2:41 PM	--	Folder
> usr	Jun 15, 2023 at 6:08 AM	--	Folder
> var	Jun 15, 2023 at 6:08 AM	11 bytes	Alias
> Volumes	Today at 2:31 PM	--	Folder

What's actually in the Root directory
(Toggle with command(⌘)-shift-period(.))

Hidden file extensions: The Finder (MacOS) is lying to you

See demonstration in class

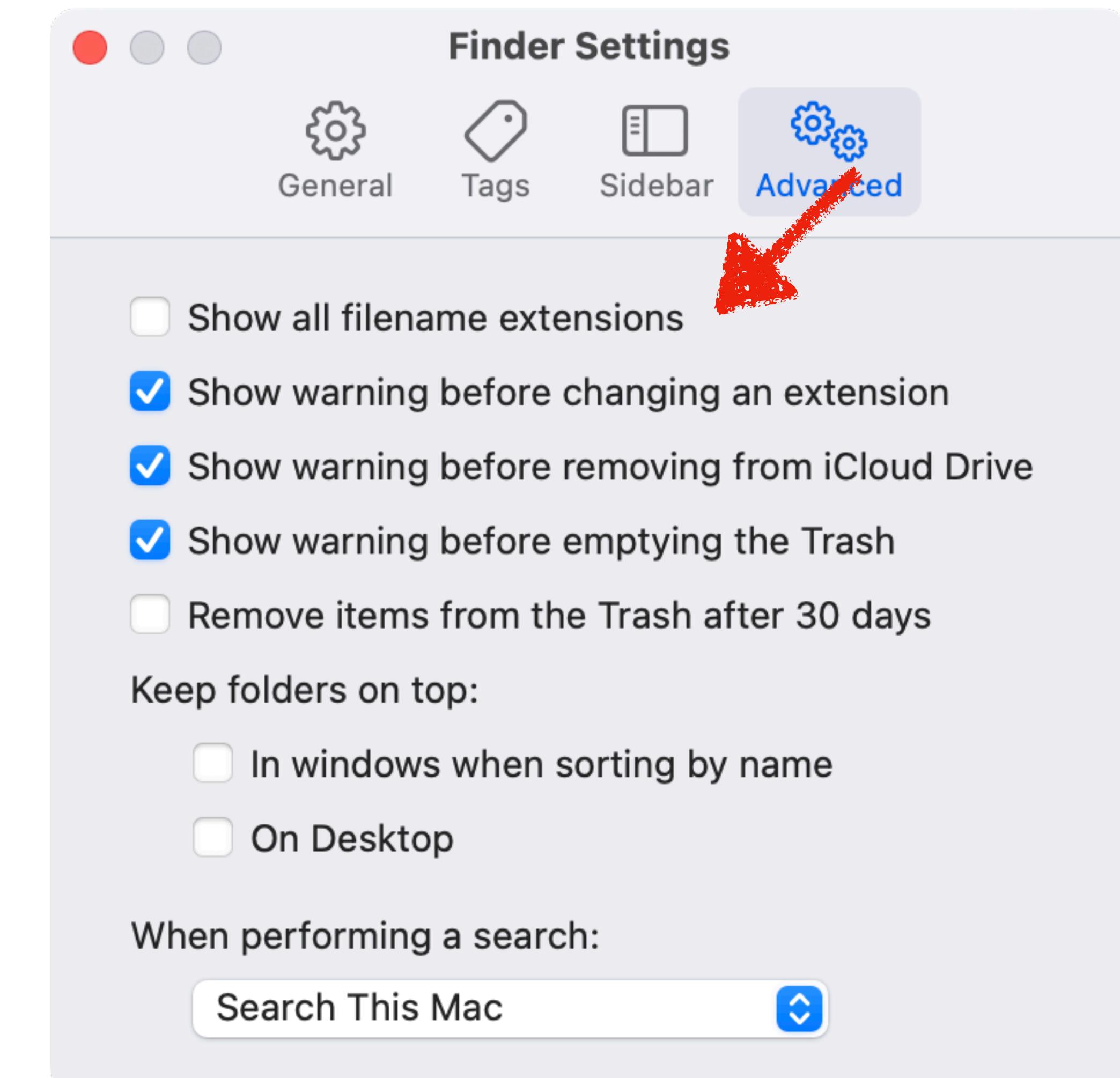
To show file extensions, MacOS

Finder >

Settings... >

Advanced >

Click the "Show all Filenames" box



Hidden file extensions: The File Explorer (Windows) is lying to you

See demonstration in class

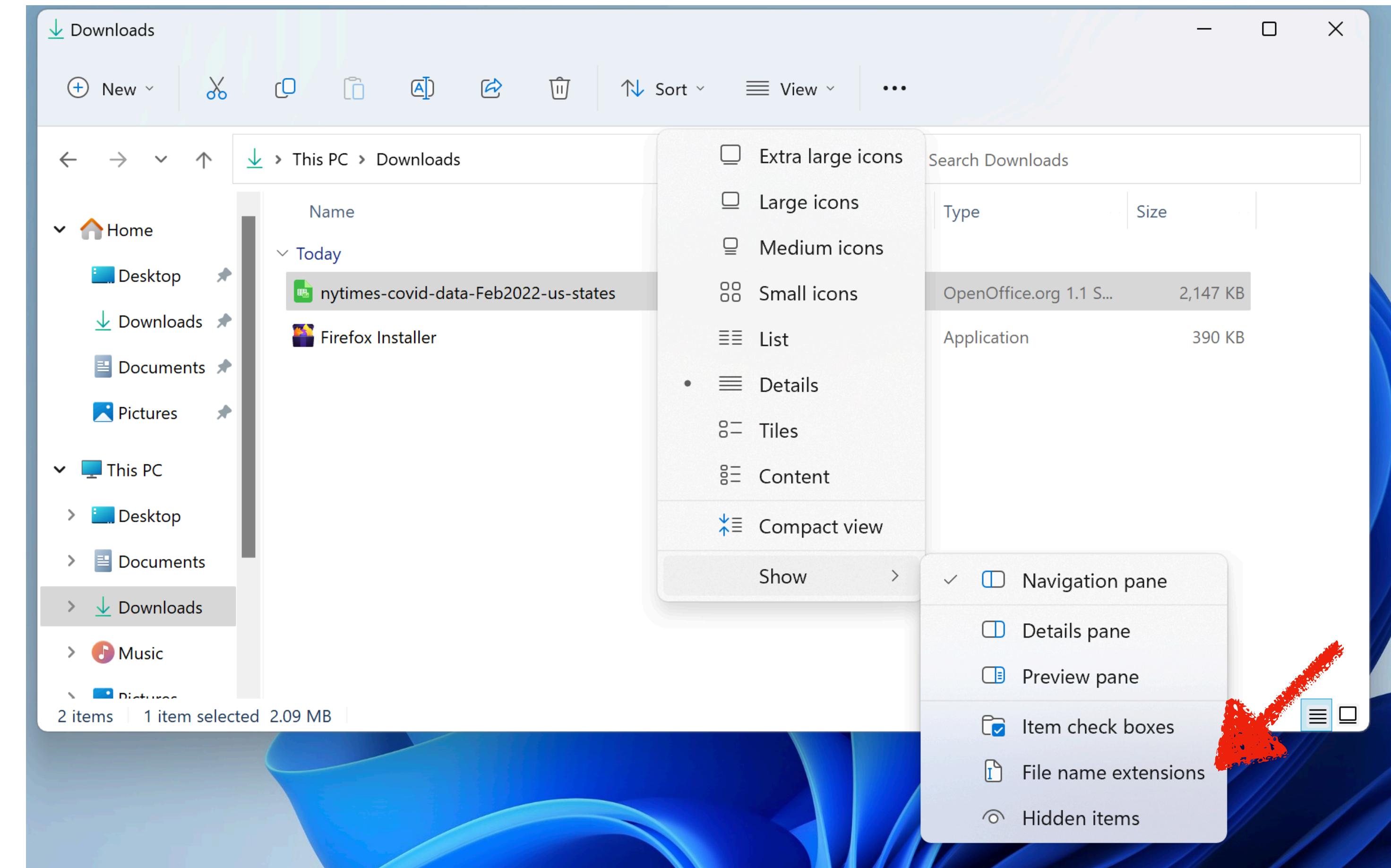
To show file extensions, Windows

File Explorer >

View >

Show >

Click "File Name Extensions"



File system locations to path names

Home directory:

Users > username

MacOS: /users/username/

Windows: c:\Users\username\my_resume.docx

A file in a subdirectory of your home directory:

Users > username > Downloads > shoe_size.csv

MacOS: /users/username/Downloads/shoe_size.csv

Windows: c:\Users\username\shoe_size.csv

The Root directory of your operating system

MacOS: /

Windows: c:\

Do the following

Create a "Bio724" directory in your user (home) directory

Make sure you can navigate to this directory quickly from the
Finder / File Explorer

Download the test file `nytimes-covid-data-us-states_2022-02.csv` (see class wiki)
and save it or move it to the Bio724 directory

What is the full textual path of the data file?

Introduction to programming with R

Why R?

Why R?

- Data Analysis and Visualization

Why R?

- Data Analysis and Visualization
- Statistical Analysis

Why R?

- Data Analysis and Visualization
- Statistical Analysis
- Data Science

Why R?

- Data Analysis and Visualization
- Statistical Analysis
- Data Science
- Reproducibility

Why R?

- Data Analysis and Visualization
- Statistical Analysis
- Data Science
- Reproducibility
- Community

Why R?

- Data Analysis and Visualization
- Statistical Analysis
- Data Science
- Reproducibility
- Community
- Integration with Biology Tools and Packages

Why R?

- Data Analysis and Visualization
- Statistical Analysis
- Data Science
- Reproducibility
- Community
- Integration with Biology Tools and Packages
- Example: <https://shiny.posit.co/r/gallery/life-sciences/covid19-tracker/>

What is the Difference between R and RStudio?

What is the Difference between R and RStudio?

R

The programming
language

RStudio:

A program to help you write
and run R code more easily

What is the Difference between R and RStudio?

R

The programming
language

Engine



<http://moderndive.com/>

RStudio:

A program to help you write
and run R code more easily

Dashboard



What is the Difference between R and RStudio?

R

The programming
language

Engine



<http://moderndive.com/>



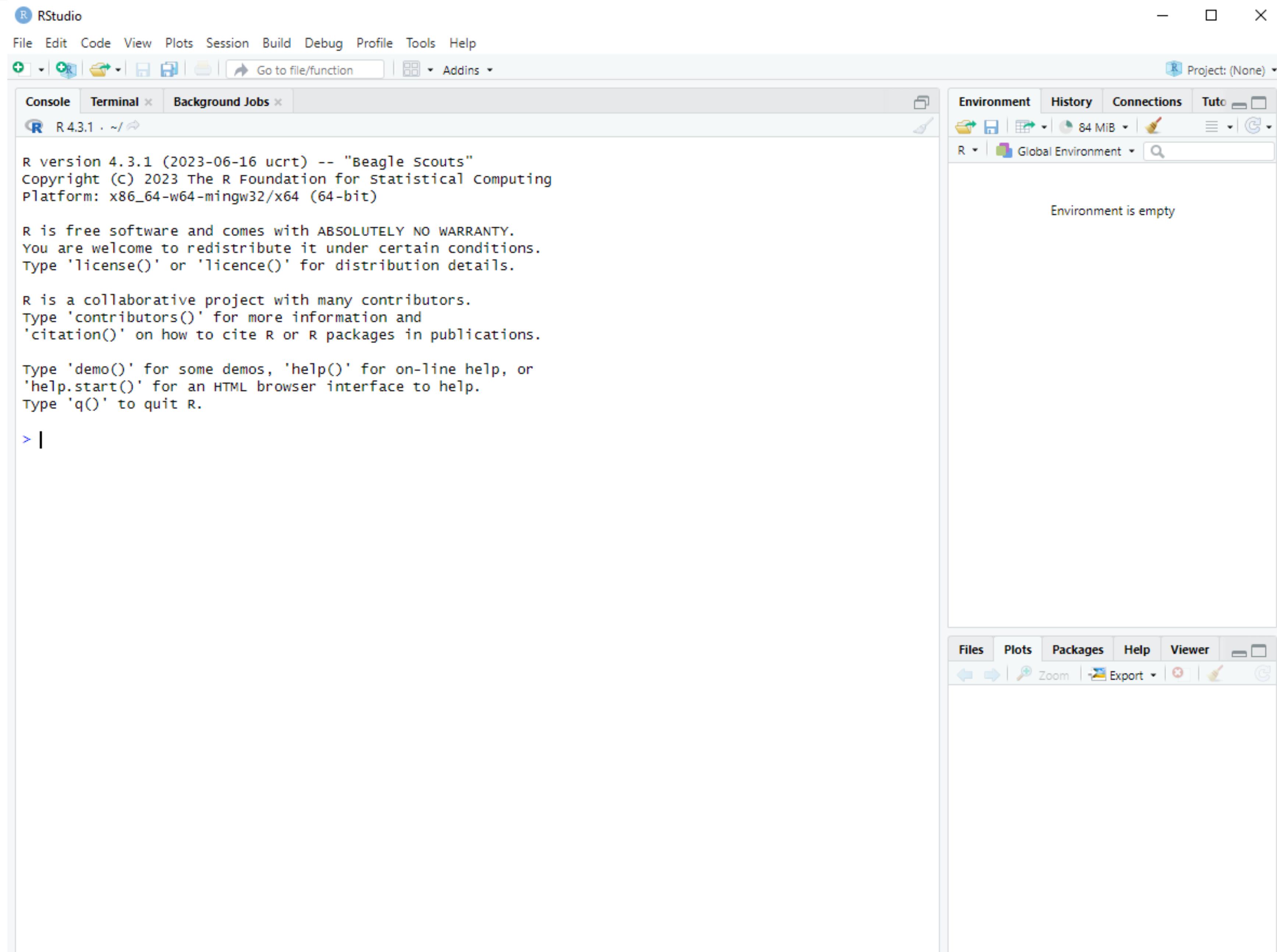
RStudio:

A program to help you write
and run R code more easily

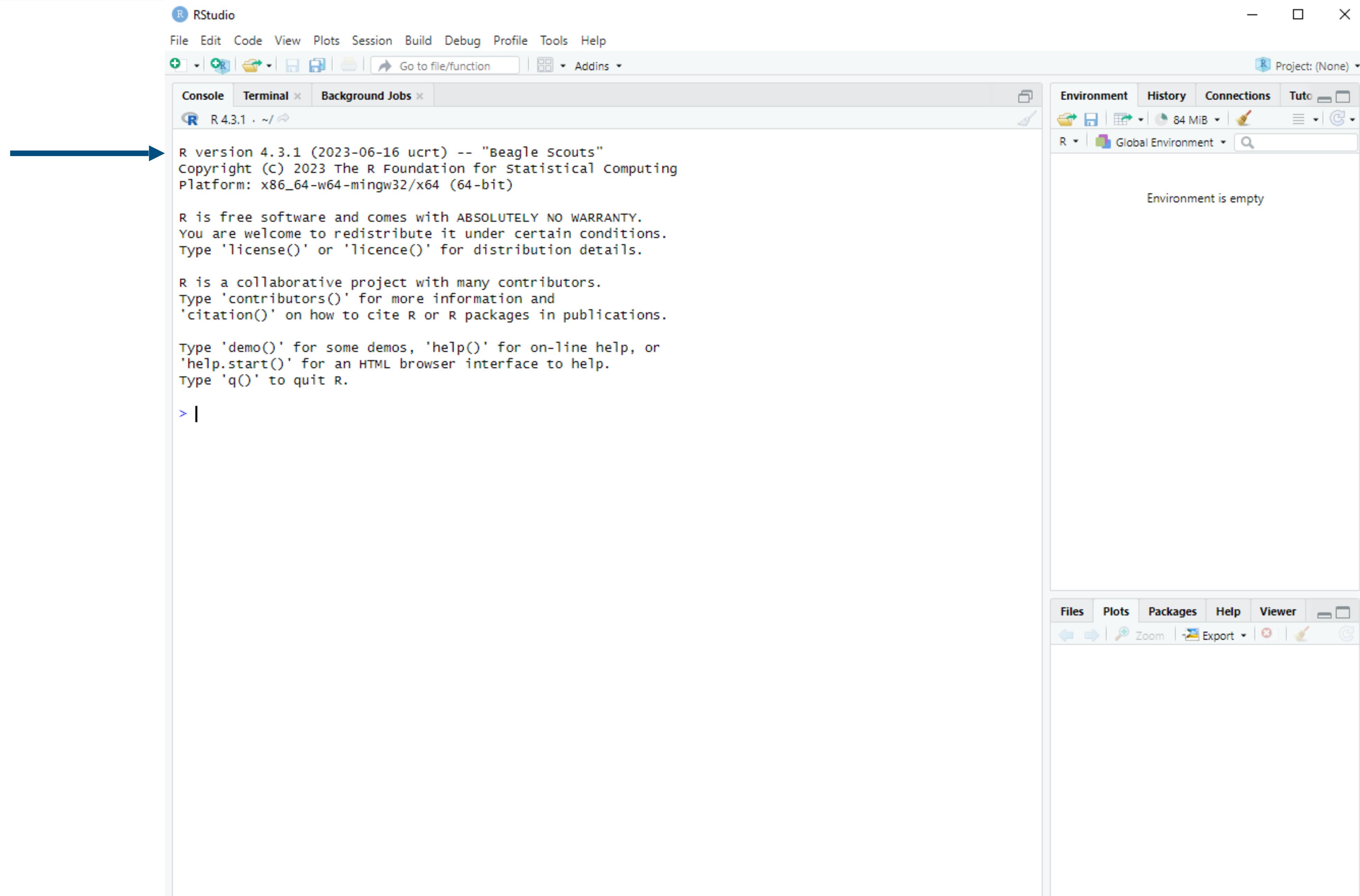
Dashboard



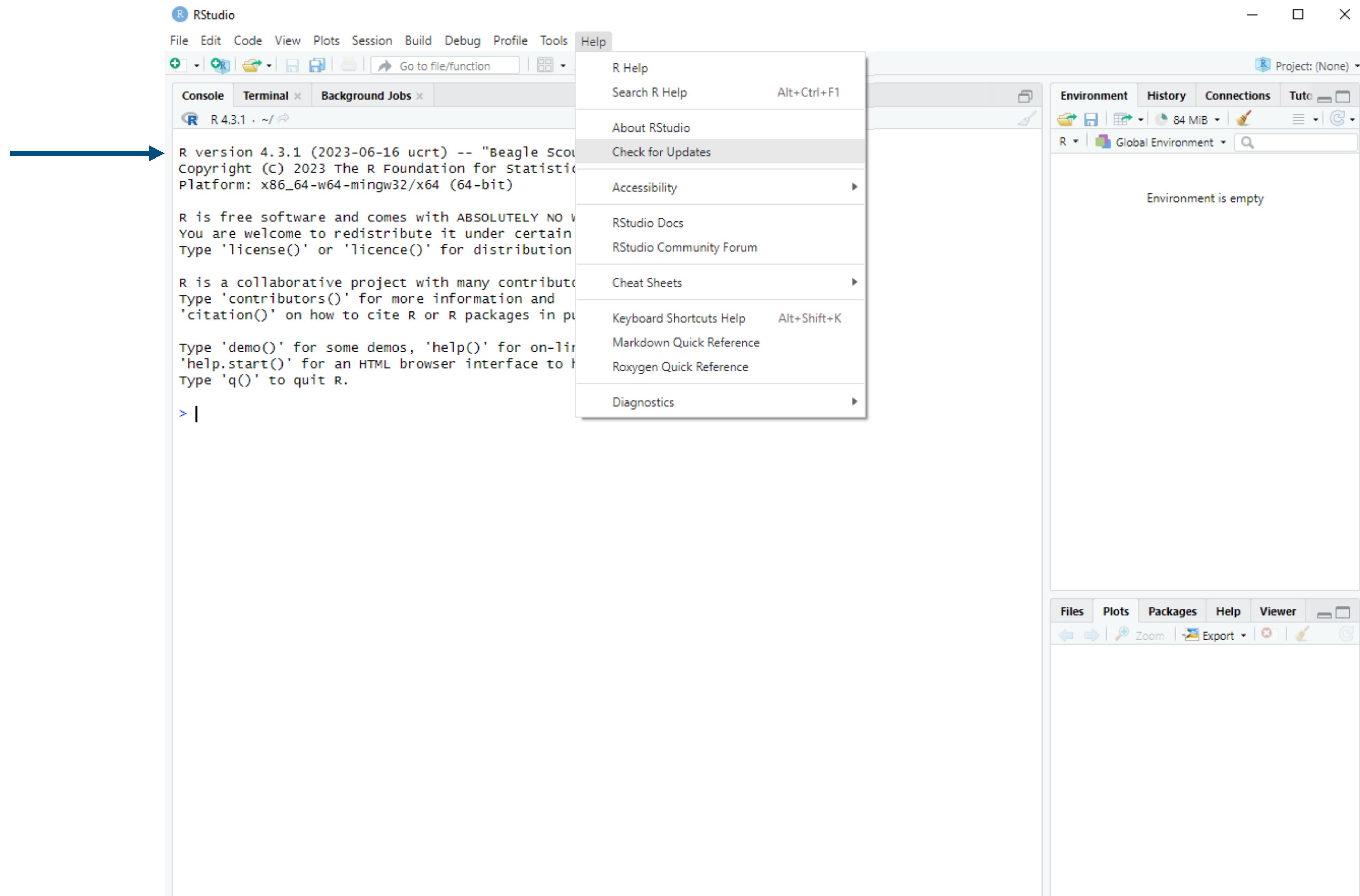
RStudio—Version



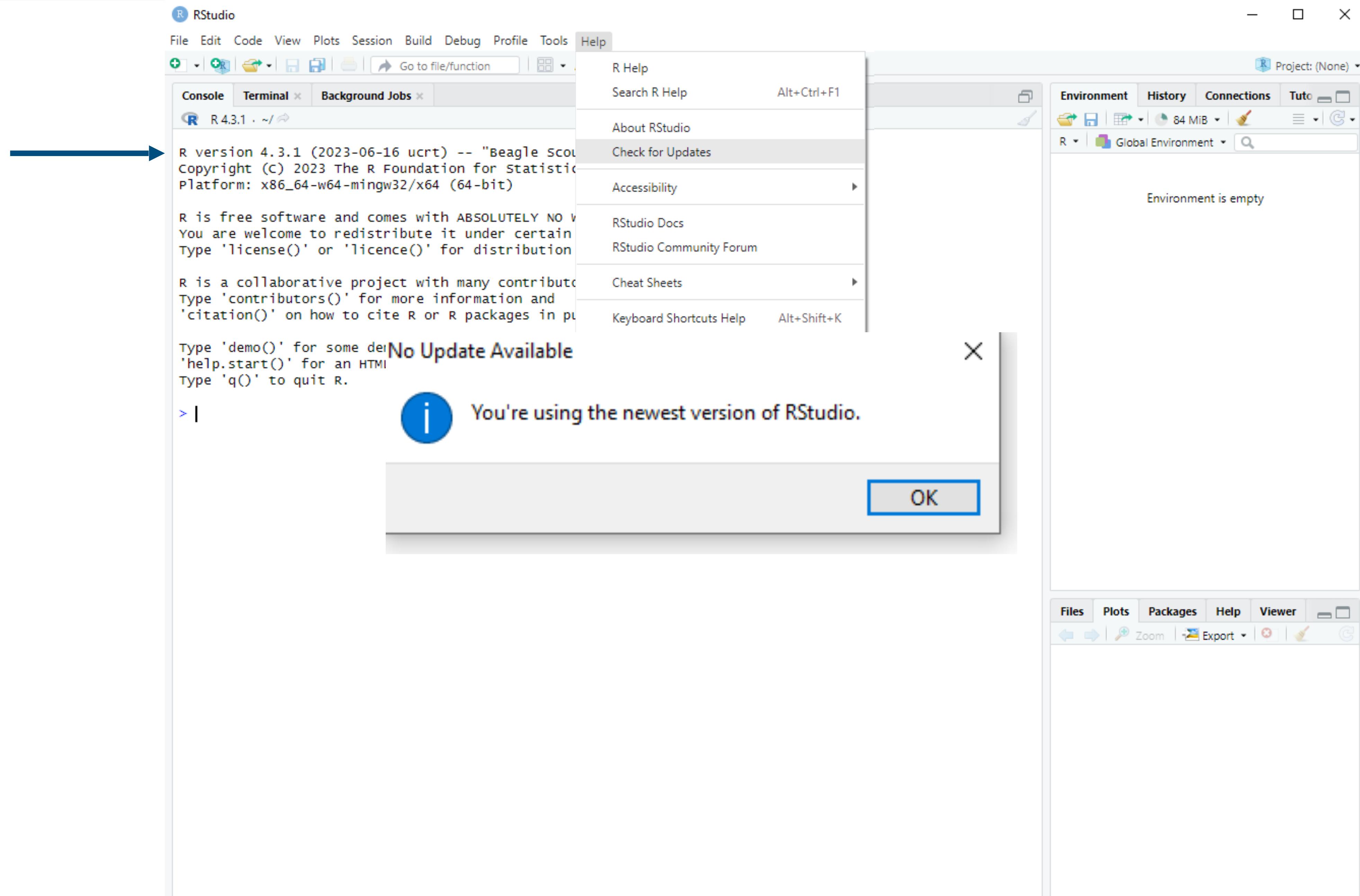
RStudio—Version



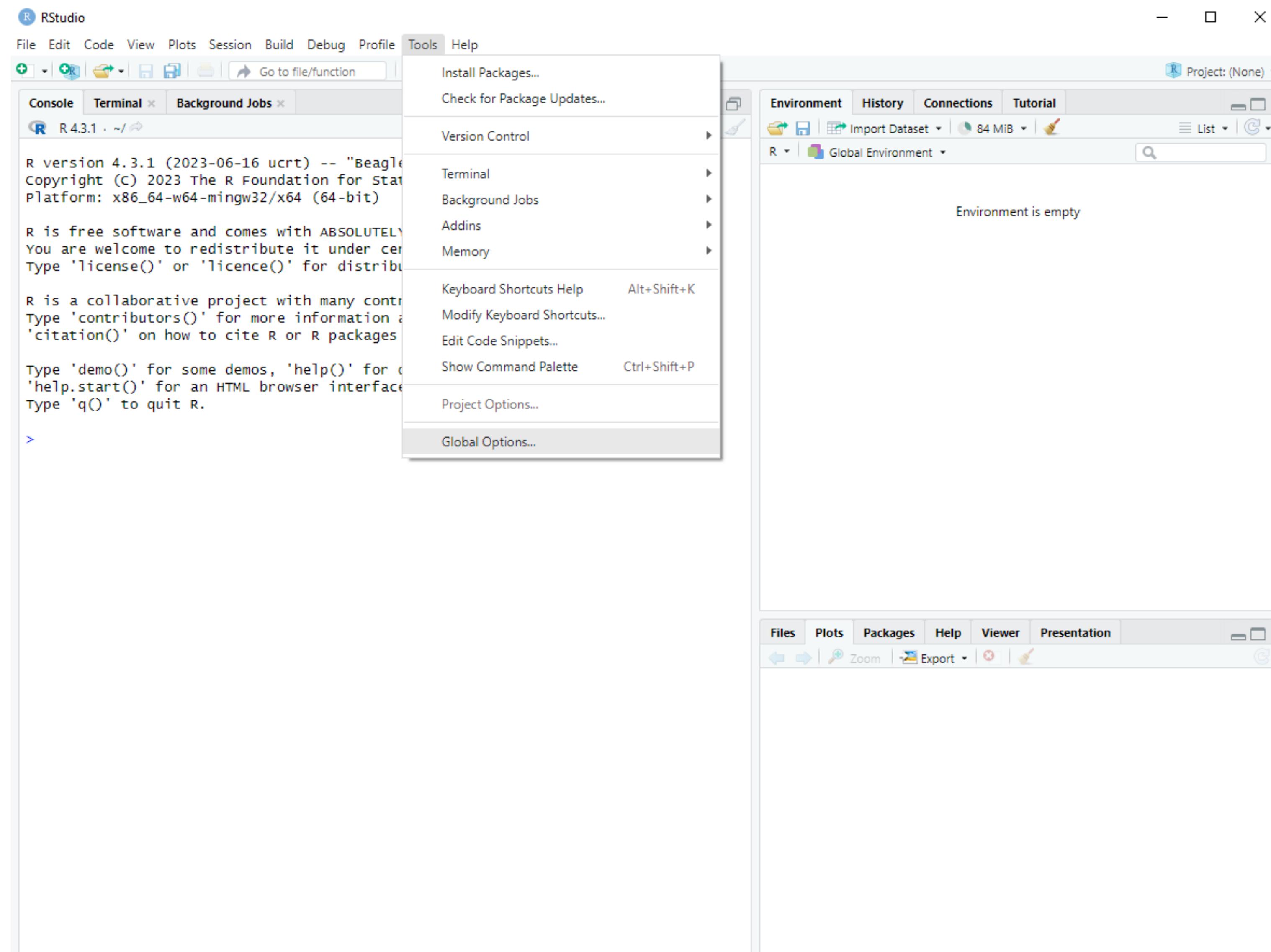
RStudio—Version



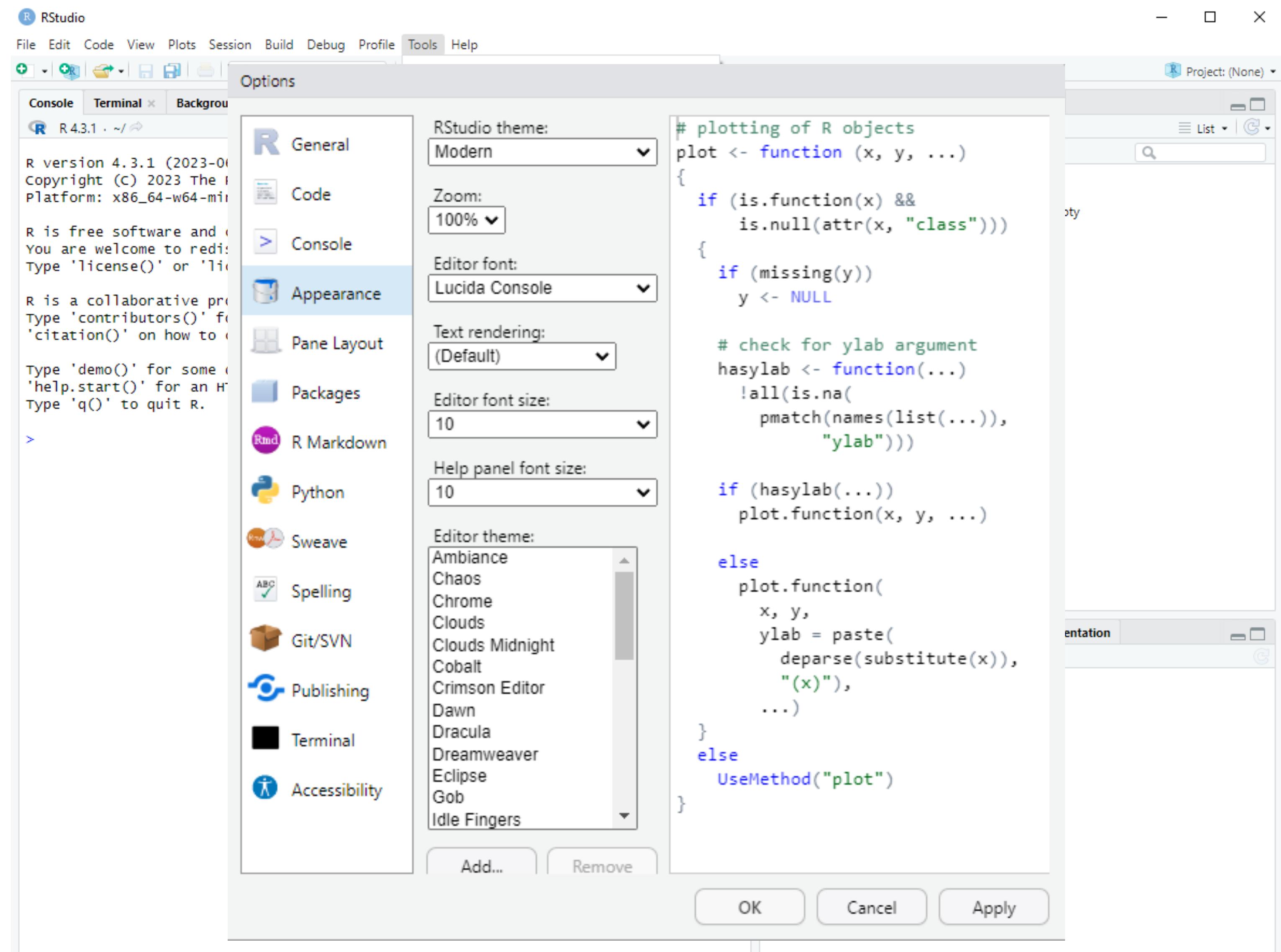
RStudio – Version



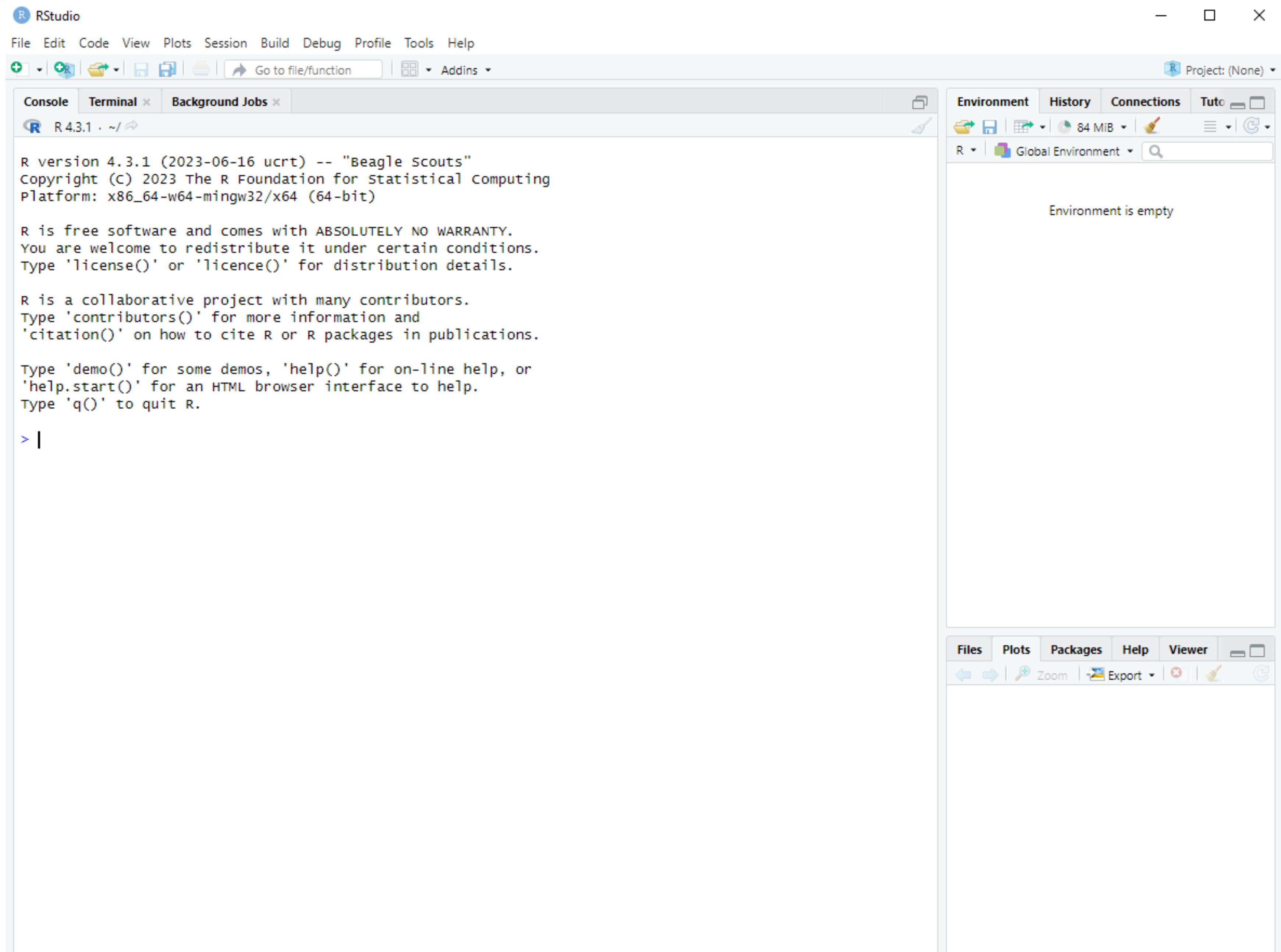
RStudio—Appearance



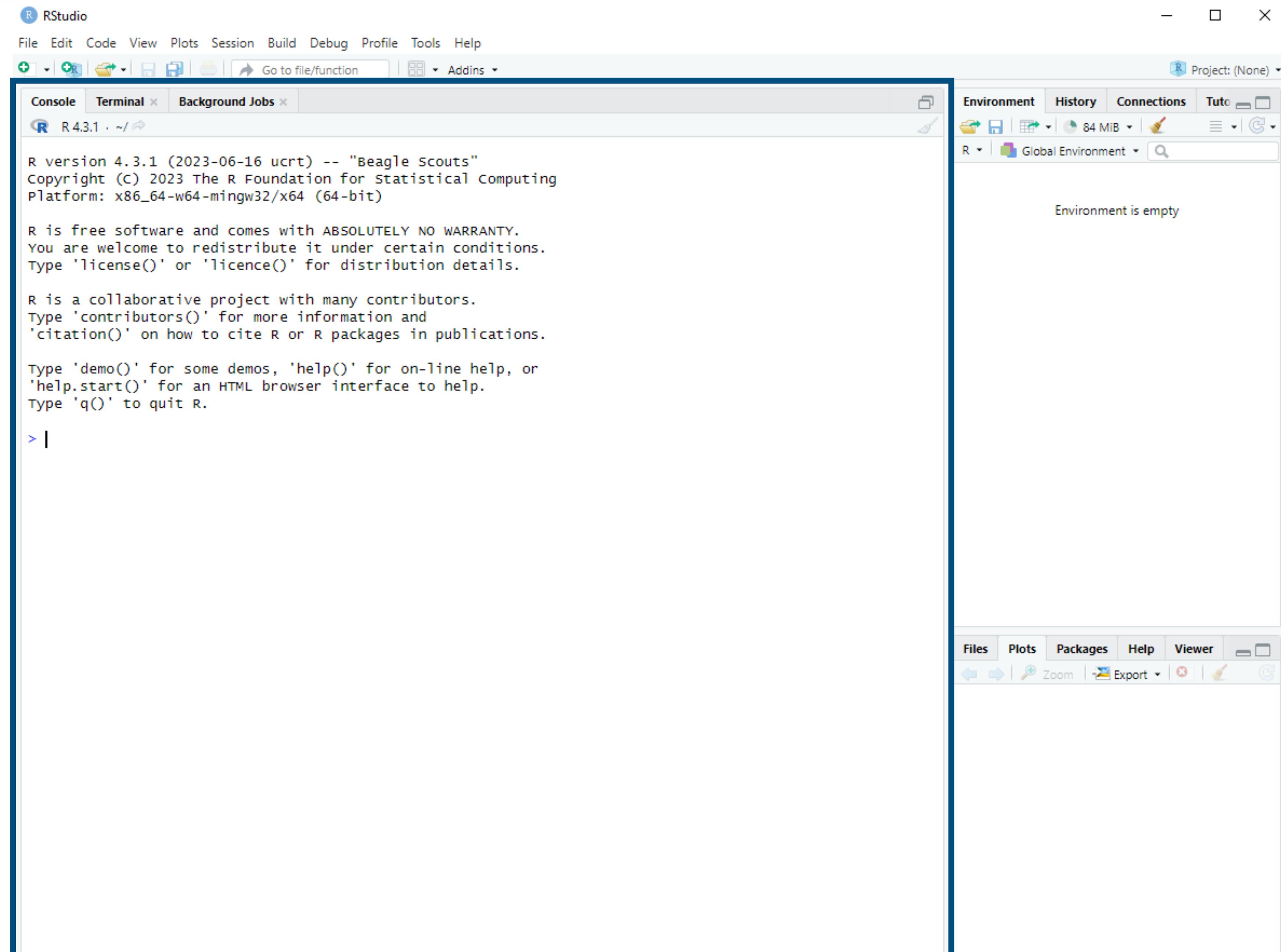
RStudio—Appearance



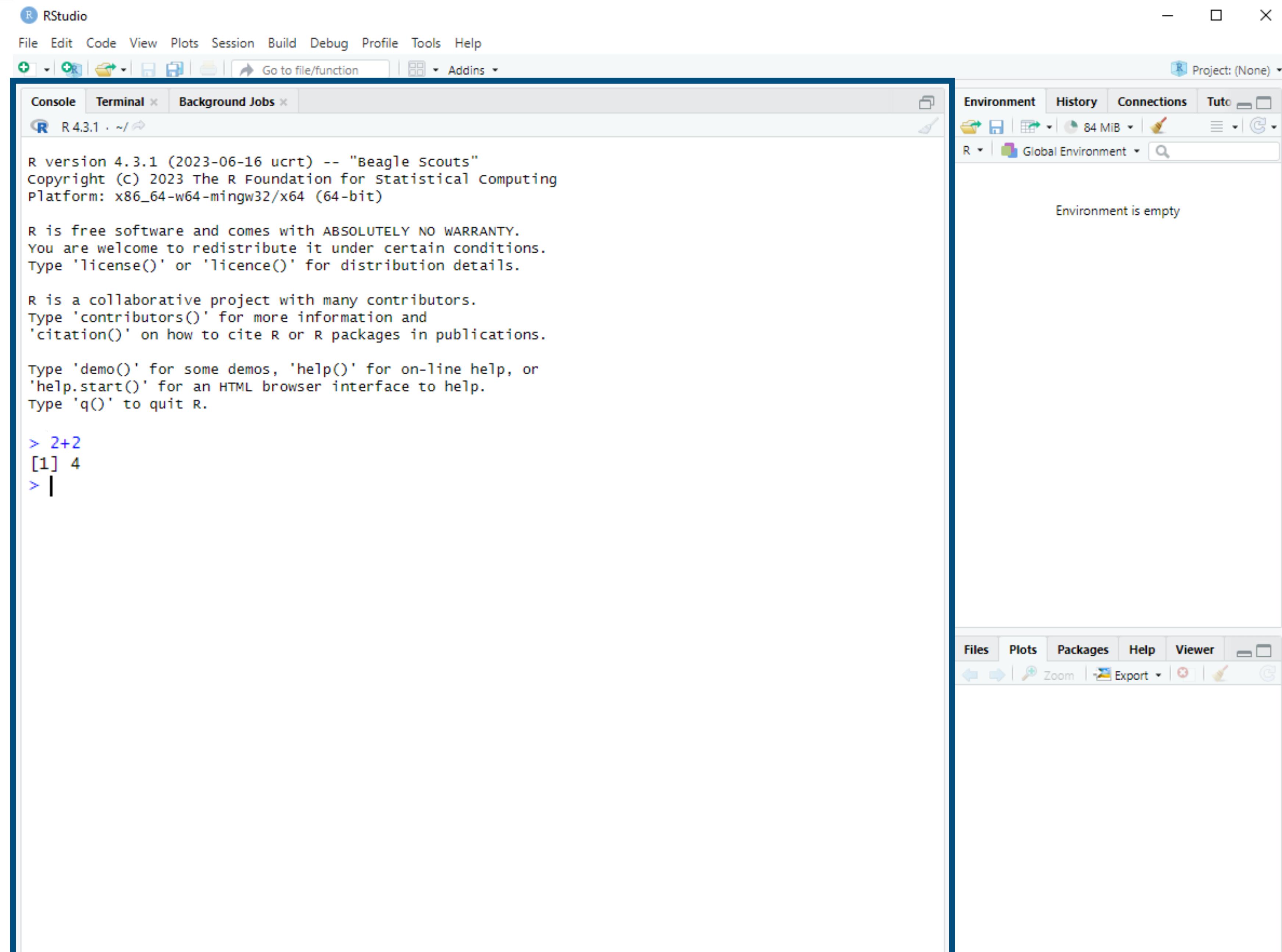
RStudio—Console



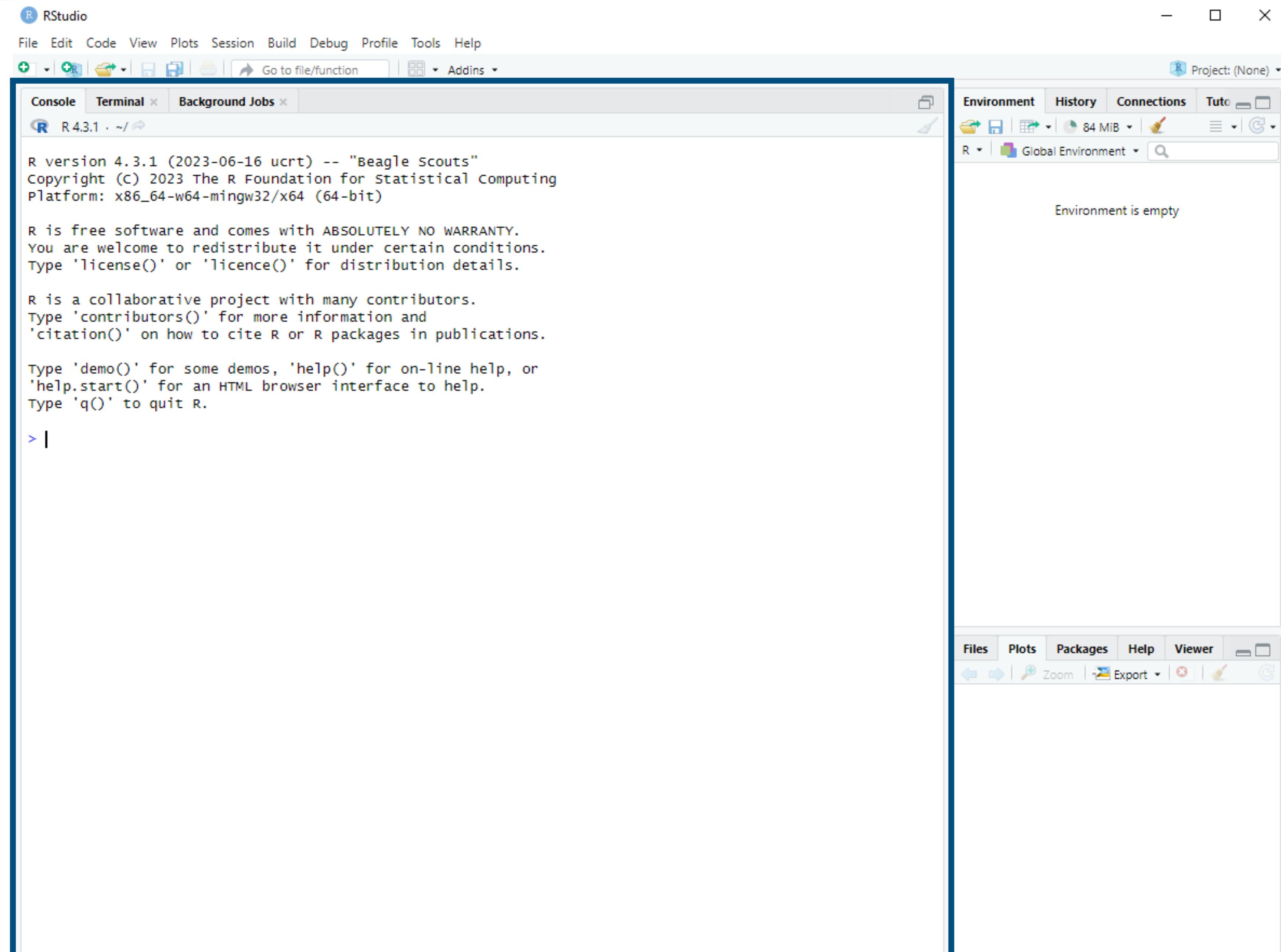
RStudio—Console



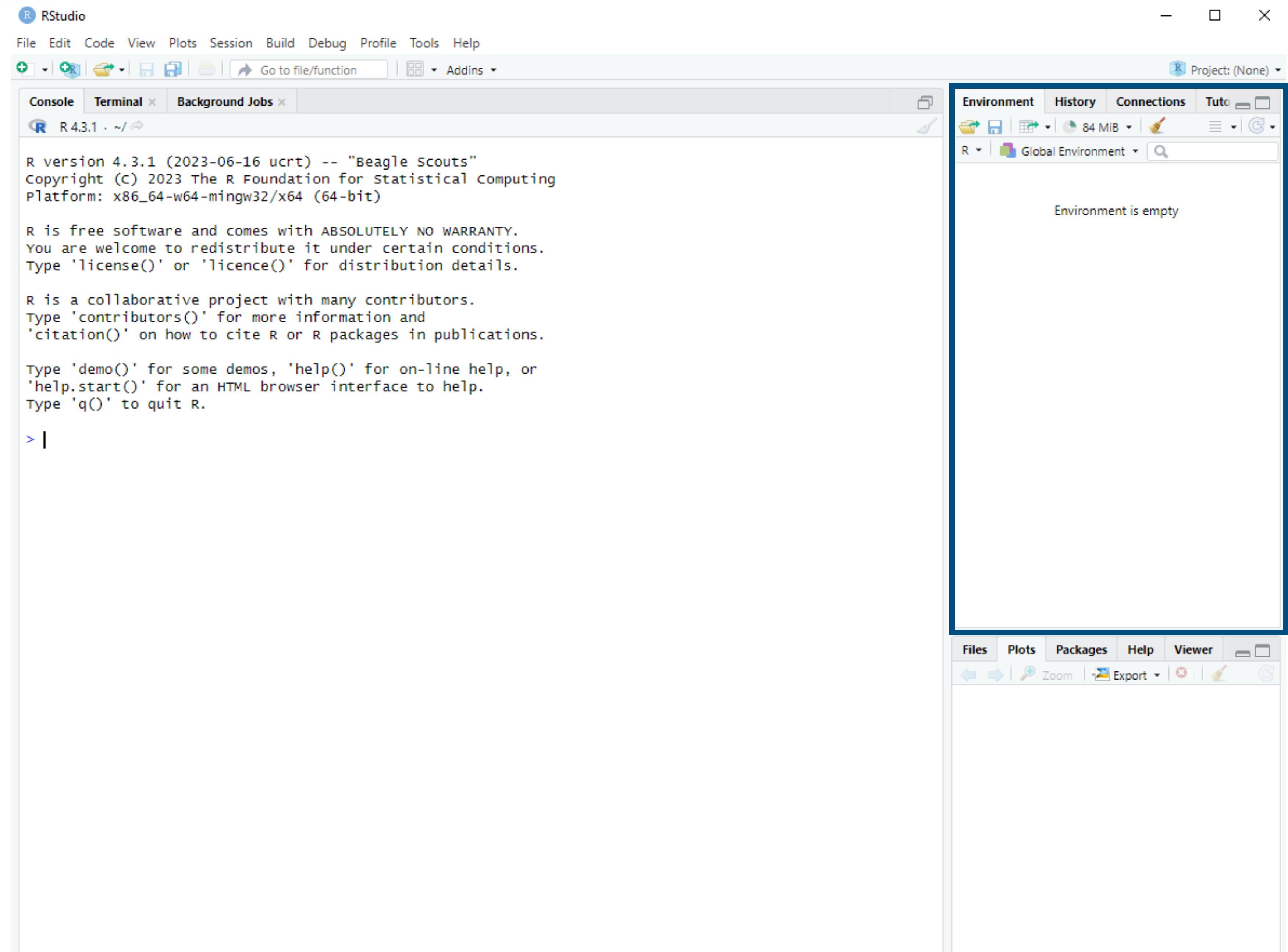
RStudio—Console



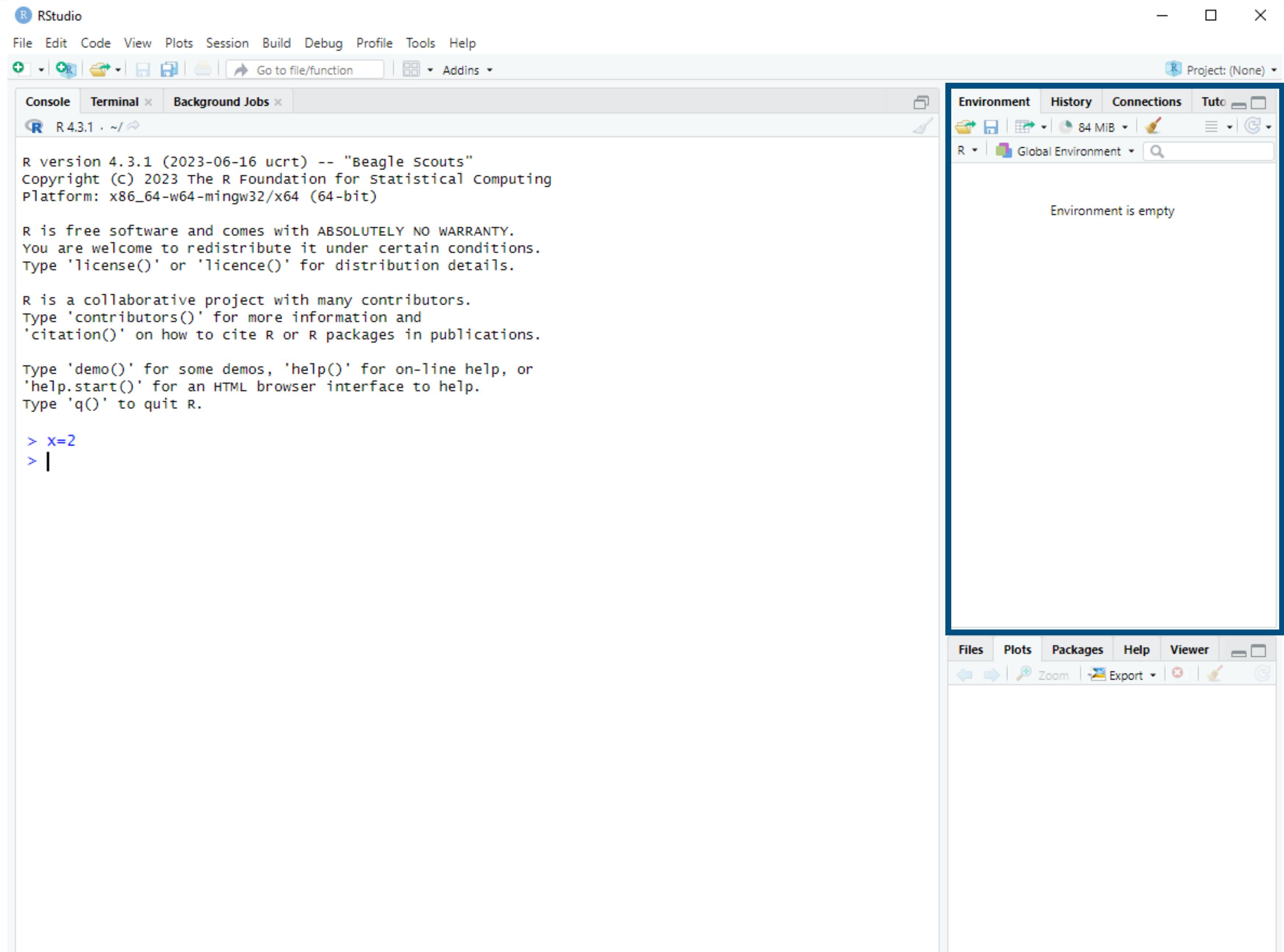
RStudio—Console



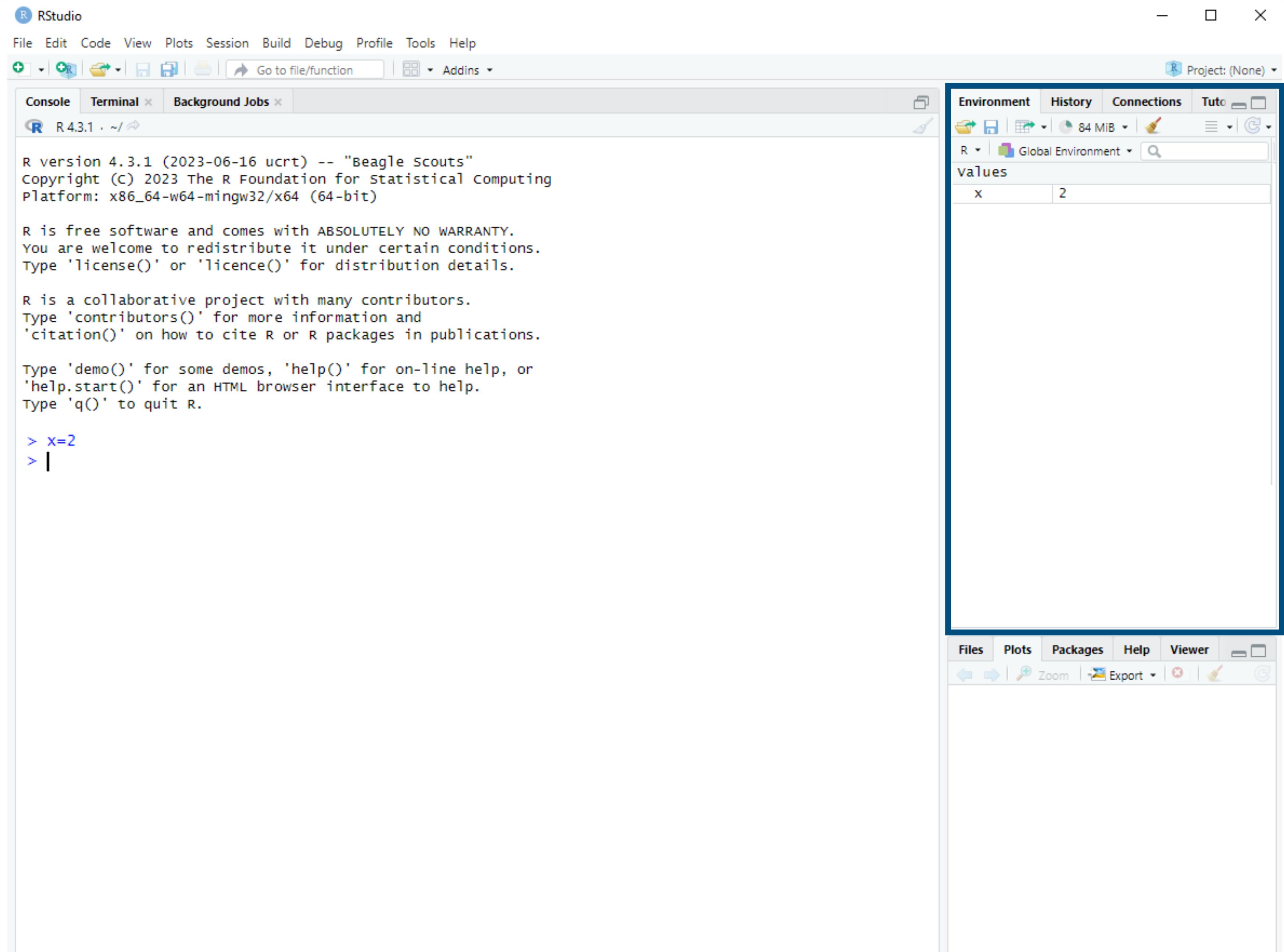
RStudio—Environment



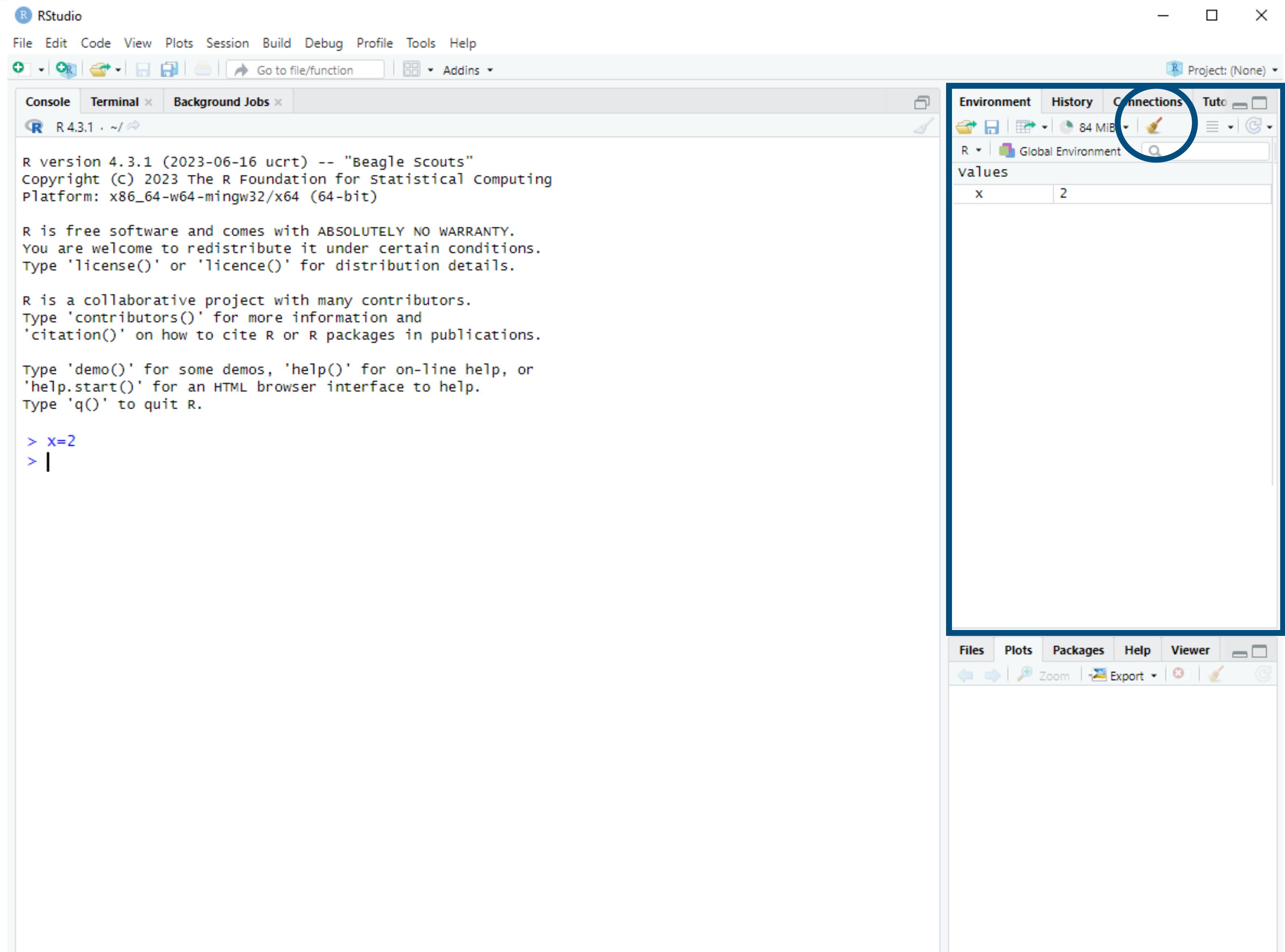
RStudio—Environment



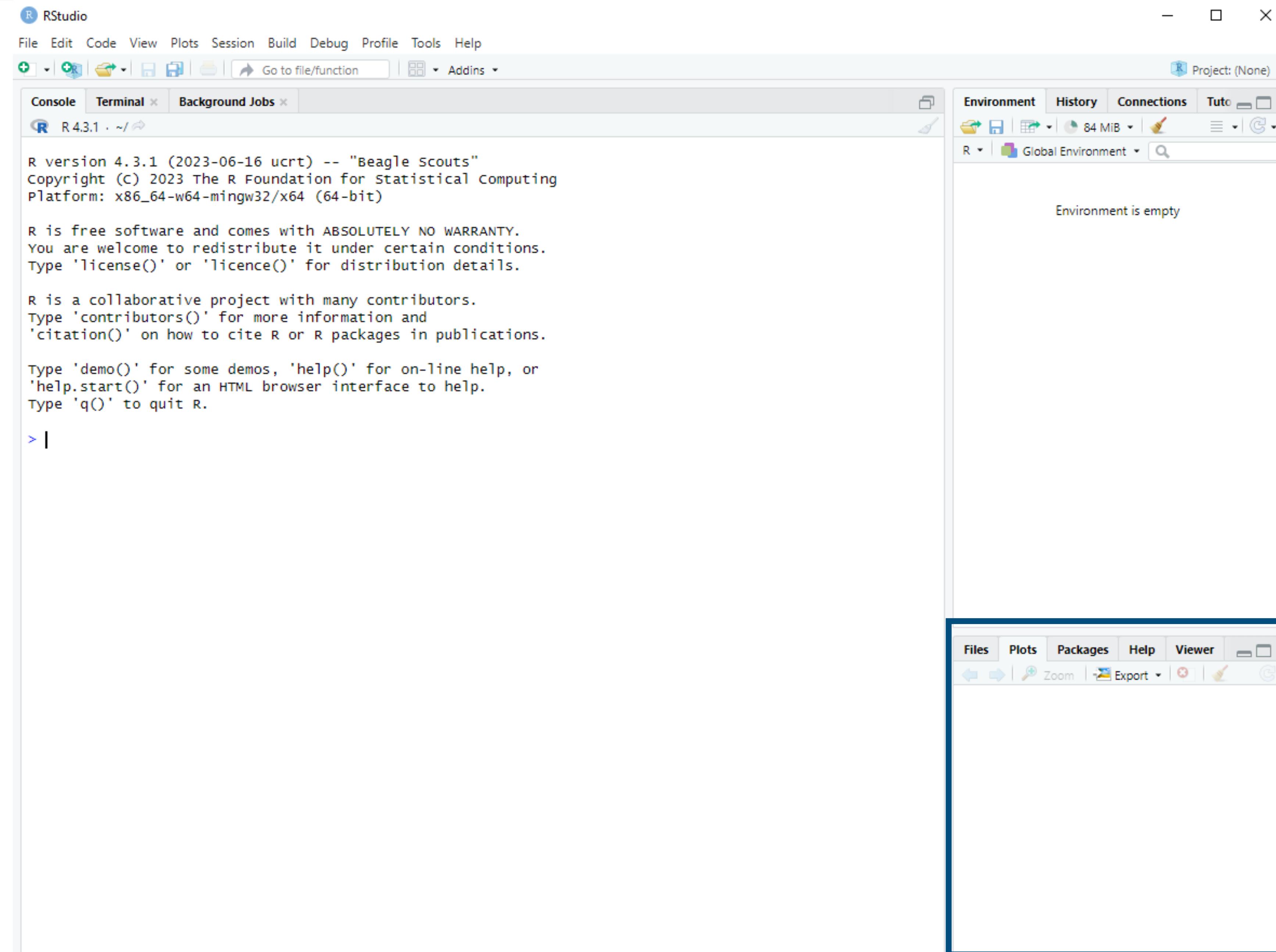
RStudio—Environment



RStudio—Environment

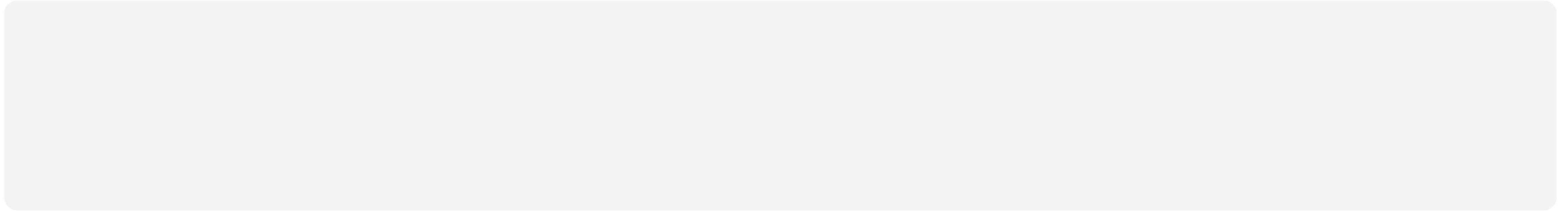


RStudio—Plot and Help Panel



RStudio—some basic tools

Save a value (case sensitive)



|

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
x <- 2
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
x <- 2
```

```
y <- "Word"
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
z = c(1,2,3,4)
```

```
x <- 2
```

```
y <- "Word"
```

```
z <- c(1,2,3,4)
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
z = c(1,2,3,4)
```

```
x <- 2
```

```
y <- "Word"
```

```
z <- c(1,2,3,4)
```

You can also can now use values you saved

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2  
y = "Word"  
z = c(1,2,3,4)
```

```
x <- 2  
y <- "Word"  
z <- c(1,2,3,4)
```

You can also use values you saved

```
x + x #Should output 4
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2  
y = "Word"  
z = c(1,2,3,4)
```

```
x <- 2  
y <- "Word"  
z <- c(1,2,3,4)
```

You can also use values you saved

```
x + x #Should output 4
```

Functions

```
mean(z) #Should output 2.5
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
z = c(1,2,3,4)
```

```
x <- 2
```

```
y <- "Word"
```

```
z <- c(1,2,3,4)
```

You can also use values you saved

```
x + x #Should output 4
```

Functions

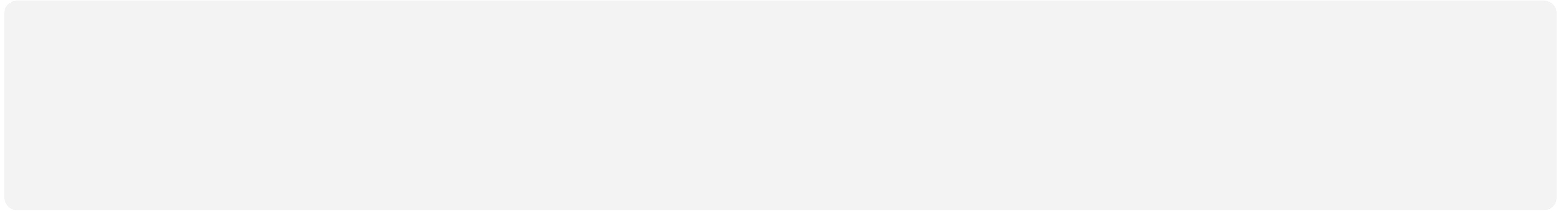
```
mean(z) #Should output 2.5
```

The screenshot shows the RStudio interface with the help documentation for the 'mean' function. The top navigation bar includes 'Files', 'Plots', 'Packages', 'Help', 'Viewer', and 'Presentation'. The main content area displays the 'Arithmetic Mean' page for the 'mean' function from the 'base' package. Key sections include 'Description' (generic function for the (trimmed) arithmetic mean), 'Usage' (the function signature: `mean(x, ...)`), 'Arguments' (descriptions for `x`, `trim`, `na.rm`, and `...`), 'Value' (description of the returned value), 'See Also' (links to related functions like `weighted.mean`), 'Examples' (a link to examples), and 'Run examples' (a button to run the examples). Below the examples section, there is R code:

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

RStudio—some basic tools

Save a value (case sensitive)



RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
x <- 2
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
x <- 2
```

```
y <- "Word"
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
z = c(1,2,3,4)
```

```
x <- 2
```

```
y <- "Word"
```

```
z <- c(1,2,3,4)
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
z = c(1,2,3,4)
```

```
x <- 2
```

```
y <- "Word"
```

```
z <- c(1,2,3,4)
```

You can also use values you saved

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2
```

```
y = "Word"
```

```
z = c(1,2,3,4)
```

```
x <- 2
```

```
y <- "Word"
```

```
z <- c(1,2,3,4)
```

You can also use values you saved

```
x + x #Should output 4
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2  
y = "Word"  
z = c(1,2,3,4)
```

```
x <- 2  
y <- "Word"  
z <- c(1,2,3,4)
```

You can also use values you saved

```
x + x #Should output 4
```

Functions

```
mean(z) #Should output 2.5  
?mean
```

RStudio—some basic tools

Save a value (case sensitive)

```
x = 2  
y = "Word"  
z = c(1,2,3,4)
```

```
x <- 2  
y <- "Word"  
z <- c(1,2,3,4)
```

You can also use values you saved

```
x + x #Should output 4
```

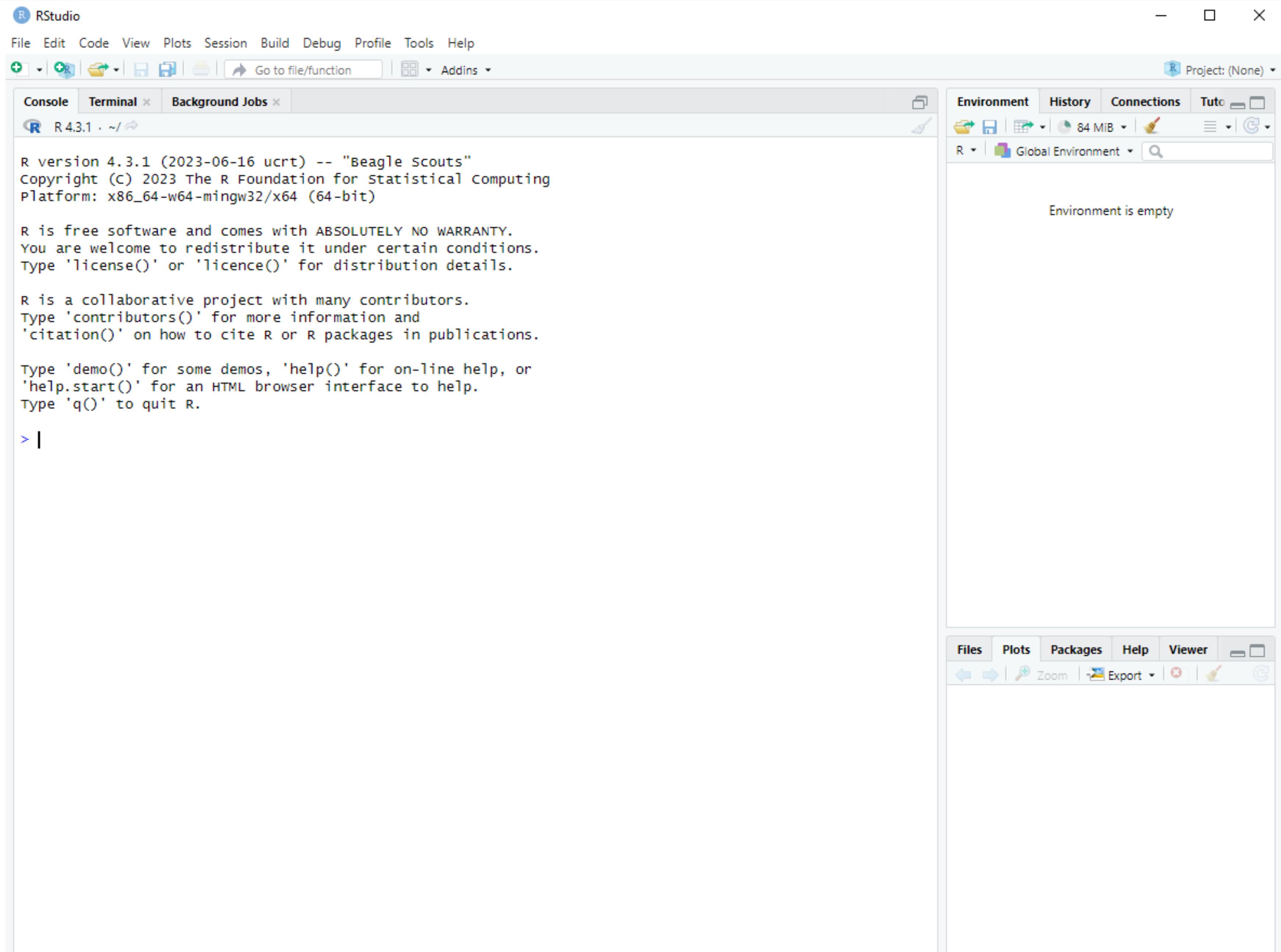
Functions

```
mean(z) #Should output 2.5
```

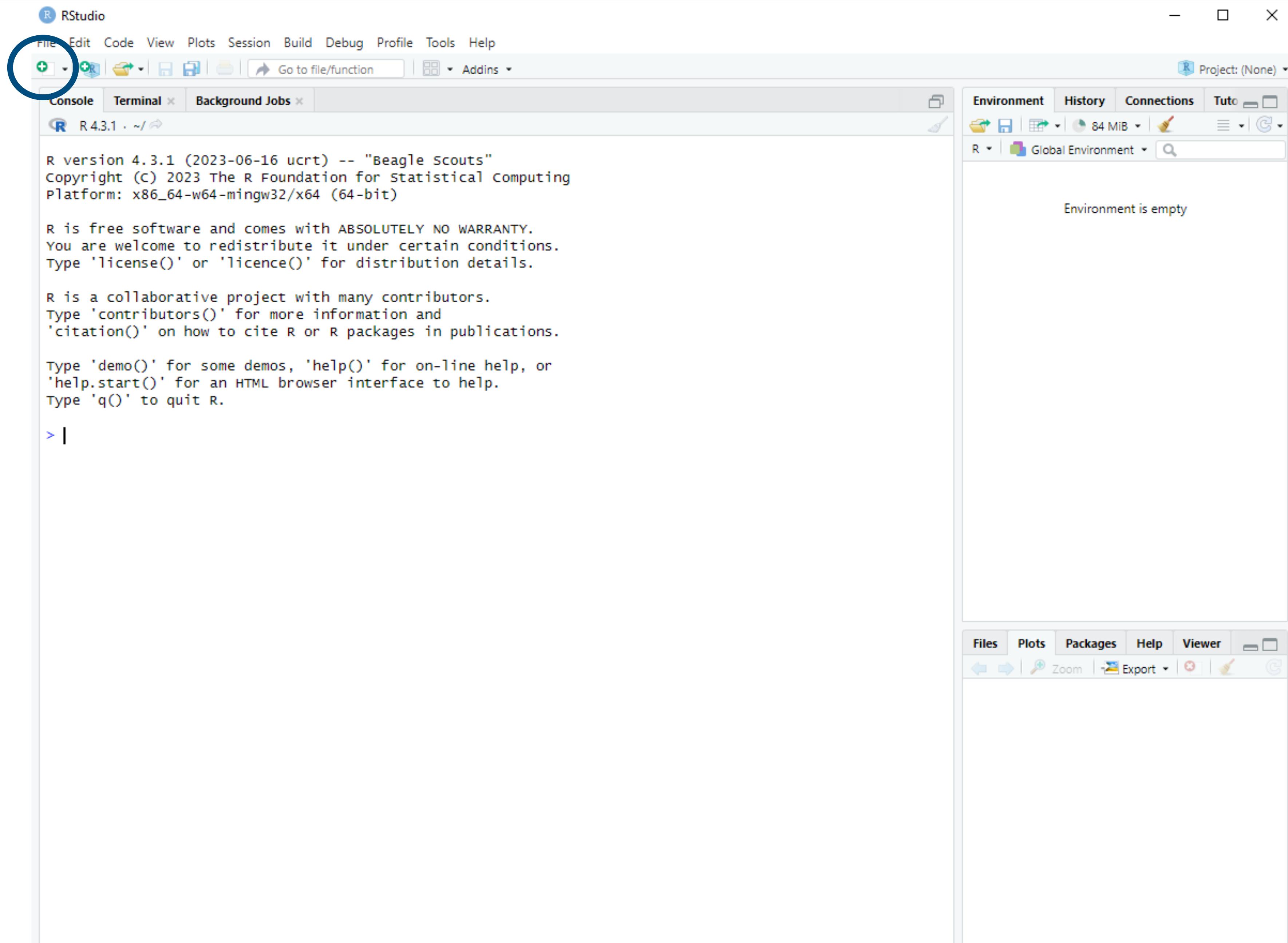
```
?mean
```

The screenshot shows the RStudio help interface for the 'mean' function. The top navigation bar includes 'Files', 'Plots', 'Packages', 'Help', 'Viewer', and 'Presentation'. The main search bar shows 'R: Arithmetic Mean'. The help page for 'mean {base}' is displayed, with sections for 'Description', 'Usage', 'Arguments', 'Value', and 'See Also'. The 'Usage' section shows the command `mean(x, ...)`. The 'Arguments' section details parameters: `x` (An R object), `trim` (the fraction of observations to be trimmed from each end of `x` before the mean is computed), `na.rm` (a logical indicating whether NA values should be stripped before the computation proceeds), and `...` (further arguments passed to or from other methods). The 'Value' section states that if `trim` is zero, 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. If `trim` is non-zero, a symmetrically trimmed mean is computed with a fraction of `trim` observations deleted from each end before the mean is computed. The 'See Also' section lists related functions: `weighted.mean`, `mean.POSIXct`, and `colMeans`. The 'Examples' section includes a code snippet: `x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))`. The 'Run examples' button is also visible.

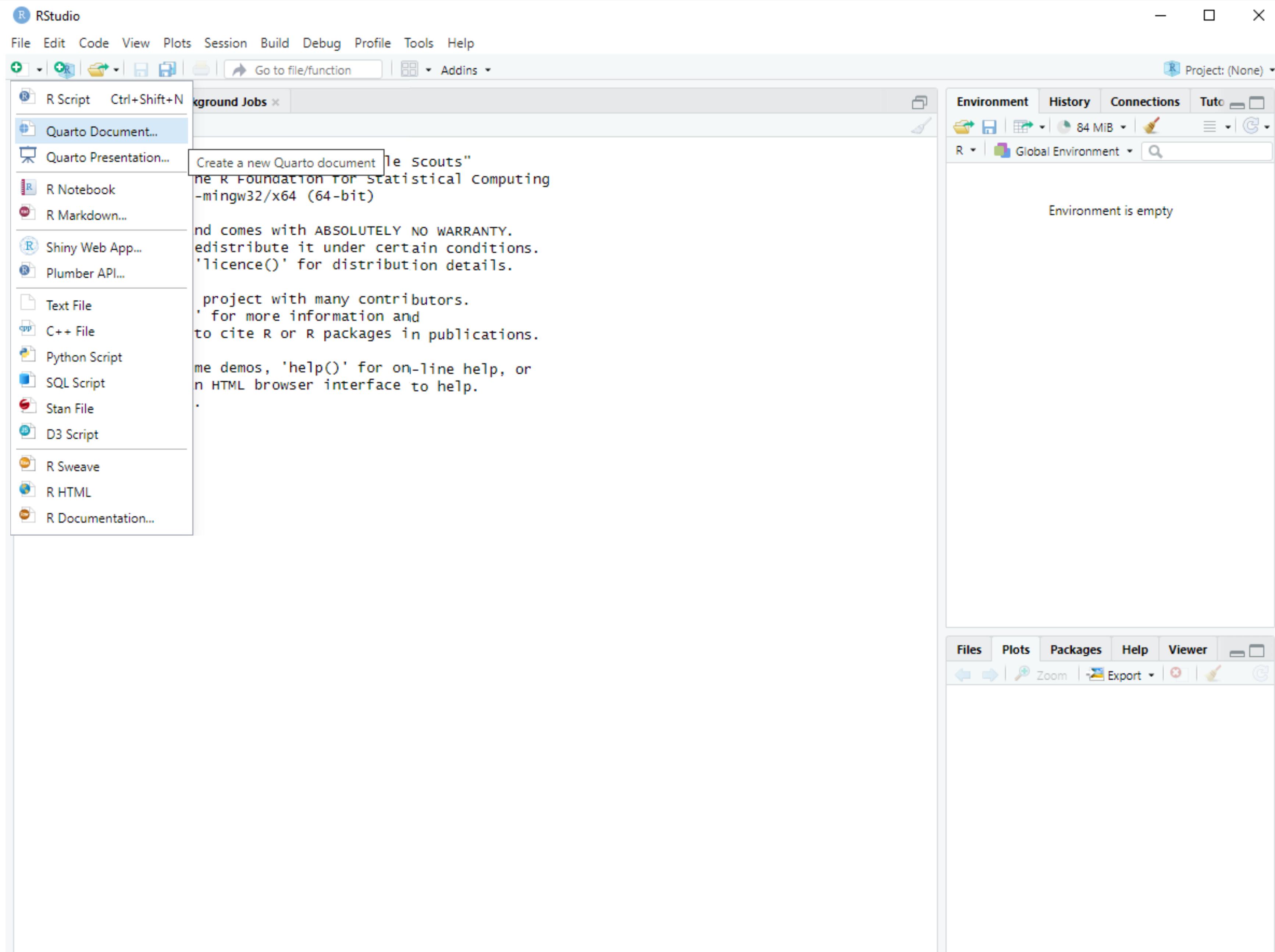
Quarto—Making a document



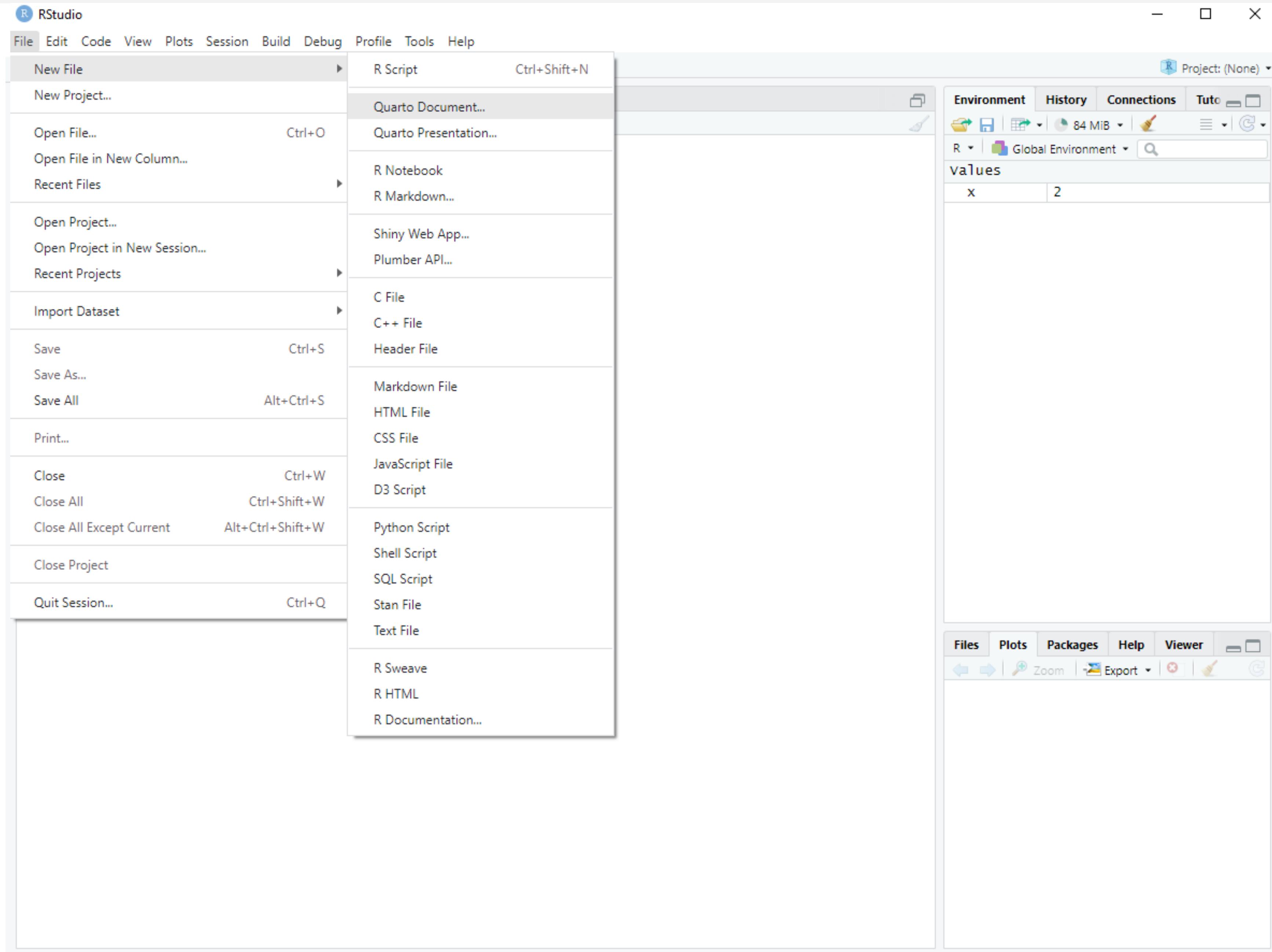
Quarto—Making a document



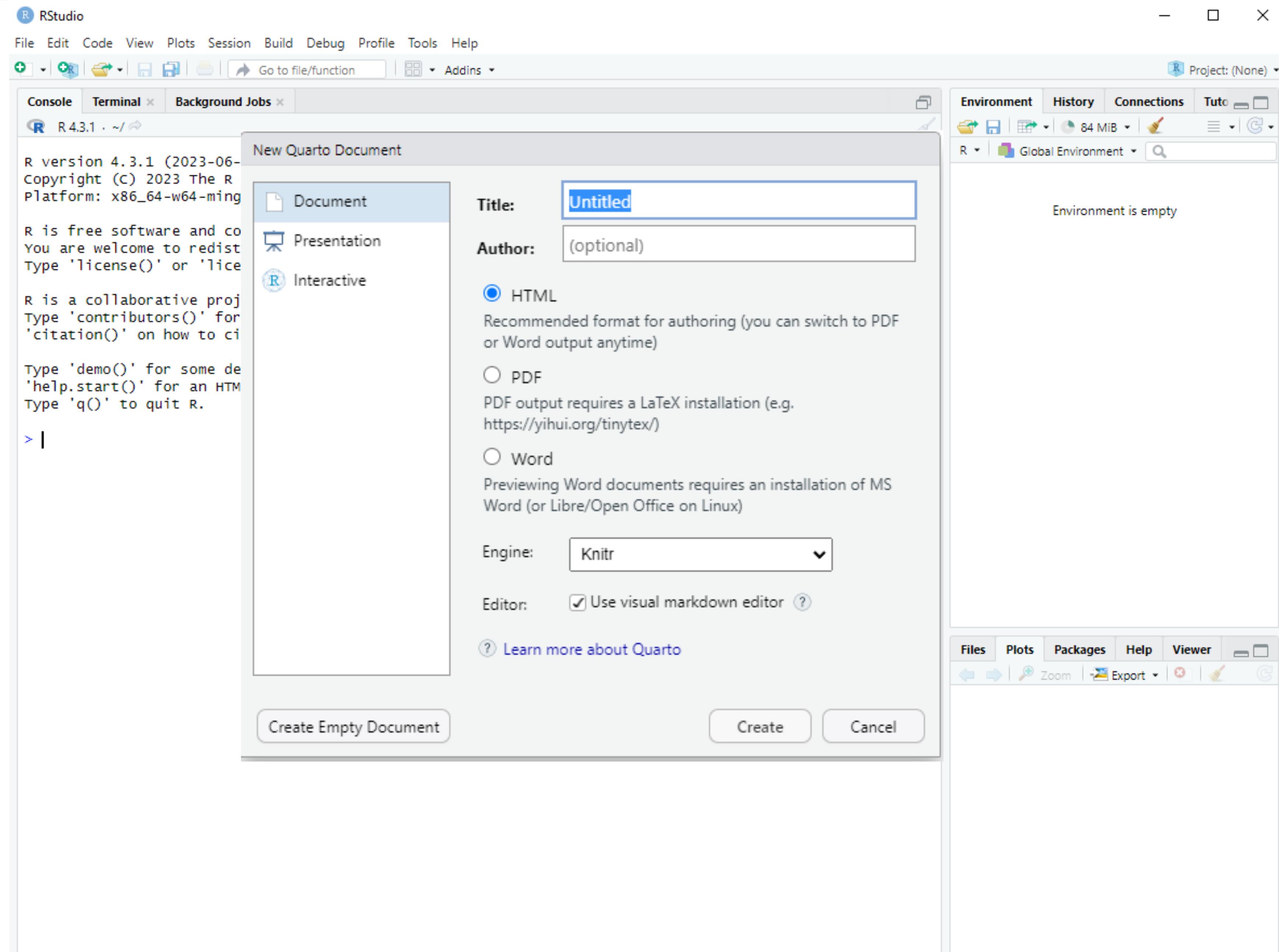
Quarto—Making a document



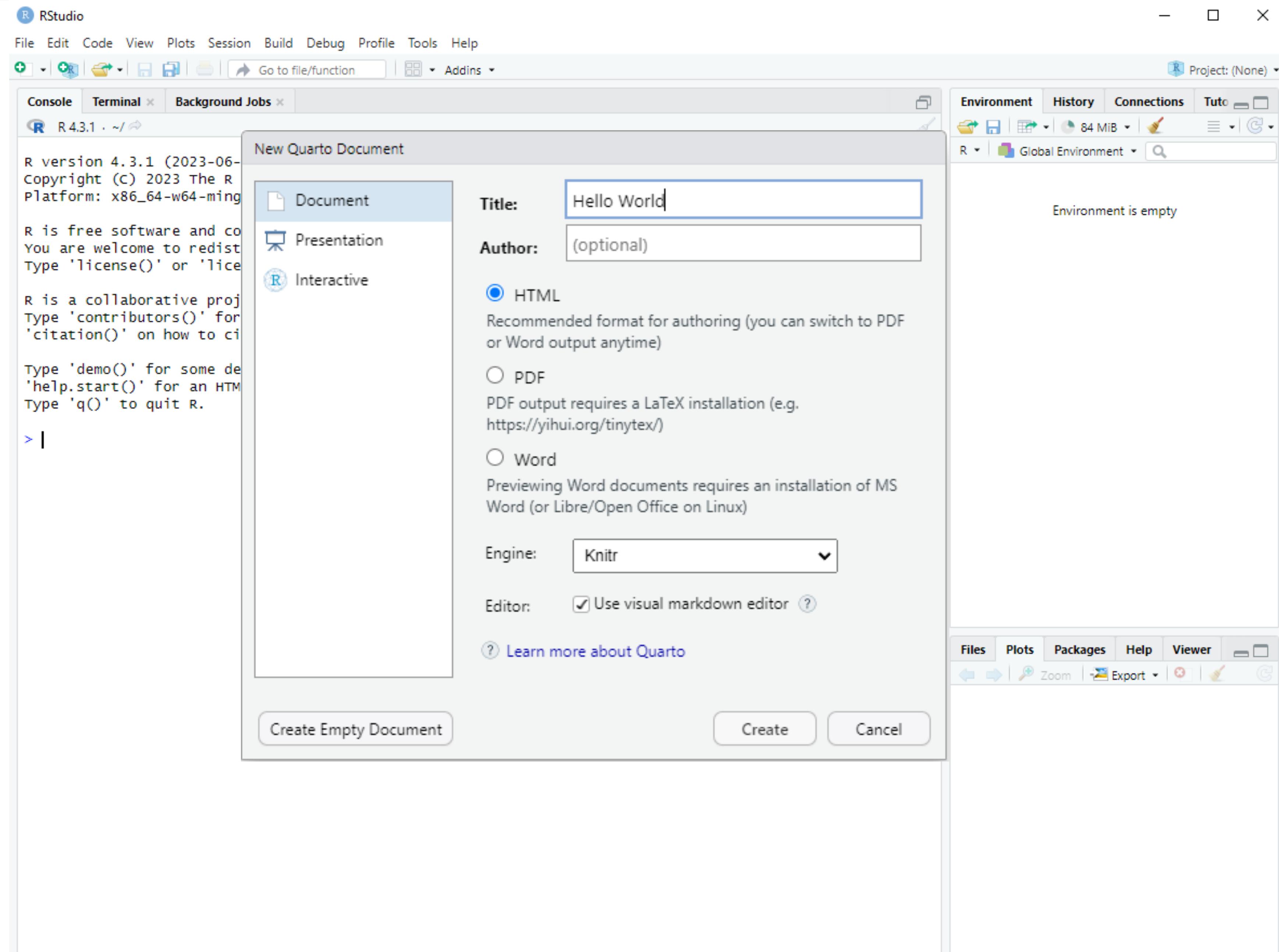
Quarto—Making a document



Quarto—Making a document



Quarto—Making a document



Quarto—Making a document

The screenshot shows the RStudio interface with a Quarto document open. The main pane displays the Quarto configuration header and two sections of content: "Quarto" and "Running Code". The "Running Code" section contains executable R code examples. The bottom pane shows the R console output.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Untitled1* x Go to file/function Addins

Source Visual B I Normal Format Insert Table Outline

Package markdown required but is not installed. [Install](#) [Don't Show Again](#)

```
title: "Hello world"
format: html
editor: visual
```

Quarto Running Code

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).

(Top Level) Quarto

Console Terminal Background Jobs

R 4.3.1 ~/

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (c) 2023 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
```

Project: (None)

Environment History Connections Tuto

Global Environment values

x	2
---	---

Files Plots Packages Help Viewer

Zoom Export

Quarto—Making a document

A screenshot of the RStudio interface demonstrating the Quarto document editor. A blue arrow points from the left towards the Quarto tab bar.

The main area shows a Quarto document titled "Untitled1" with the following YAML front matter:

```
---  
title: "Hello world"  
format: html  
editor: visual  
---
```

The document content includes:

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).

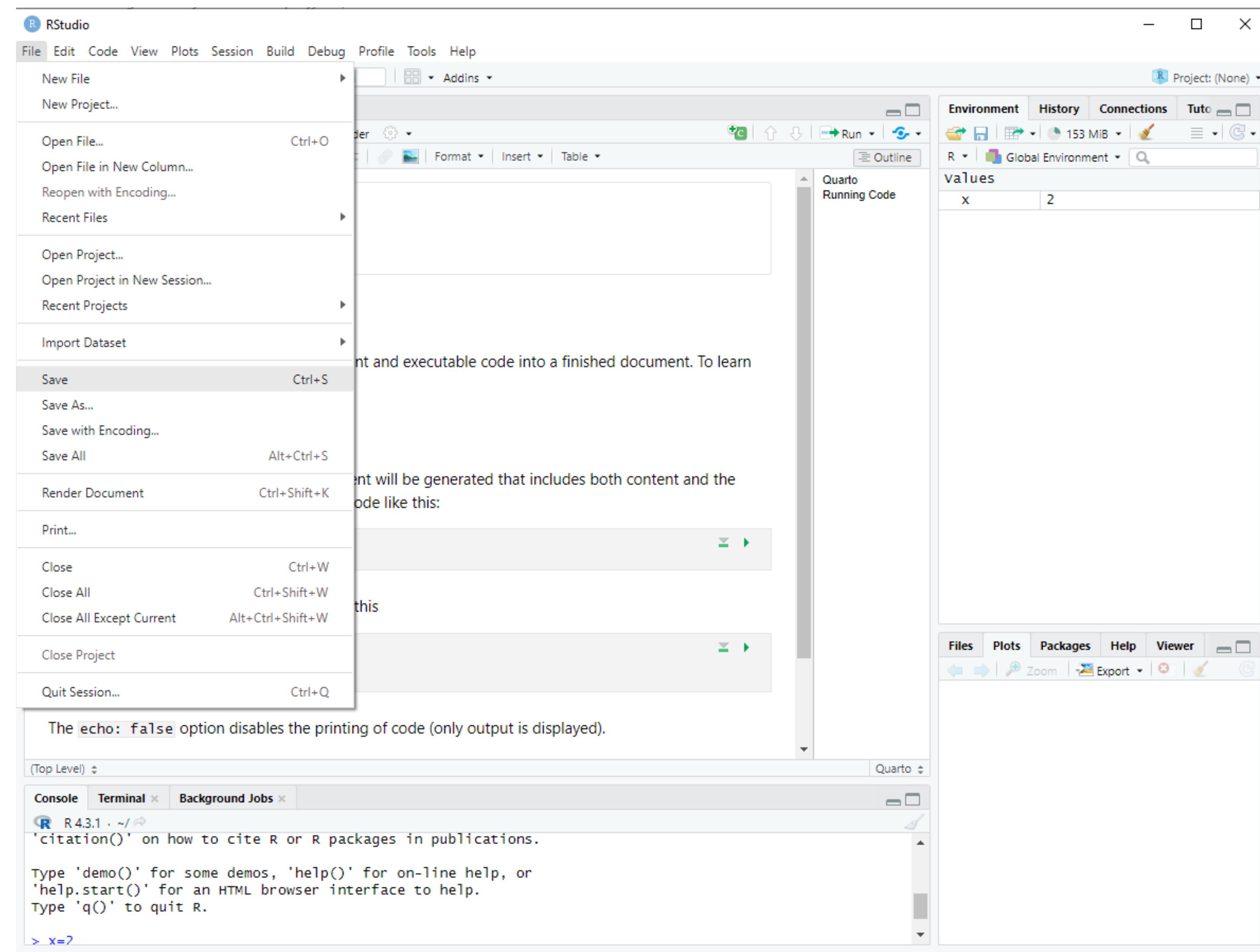
The bottom pane shows the R console output:

```
(Top Level) Quarto  
Console Terminal Background Jobs  
R 4.3.1 ~/  
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (c) 2023 The R Foundation for Statistical Computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions
```

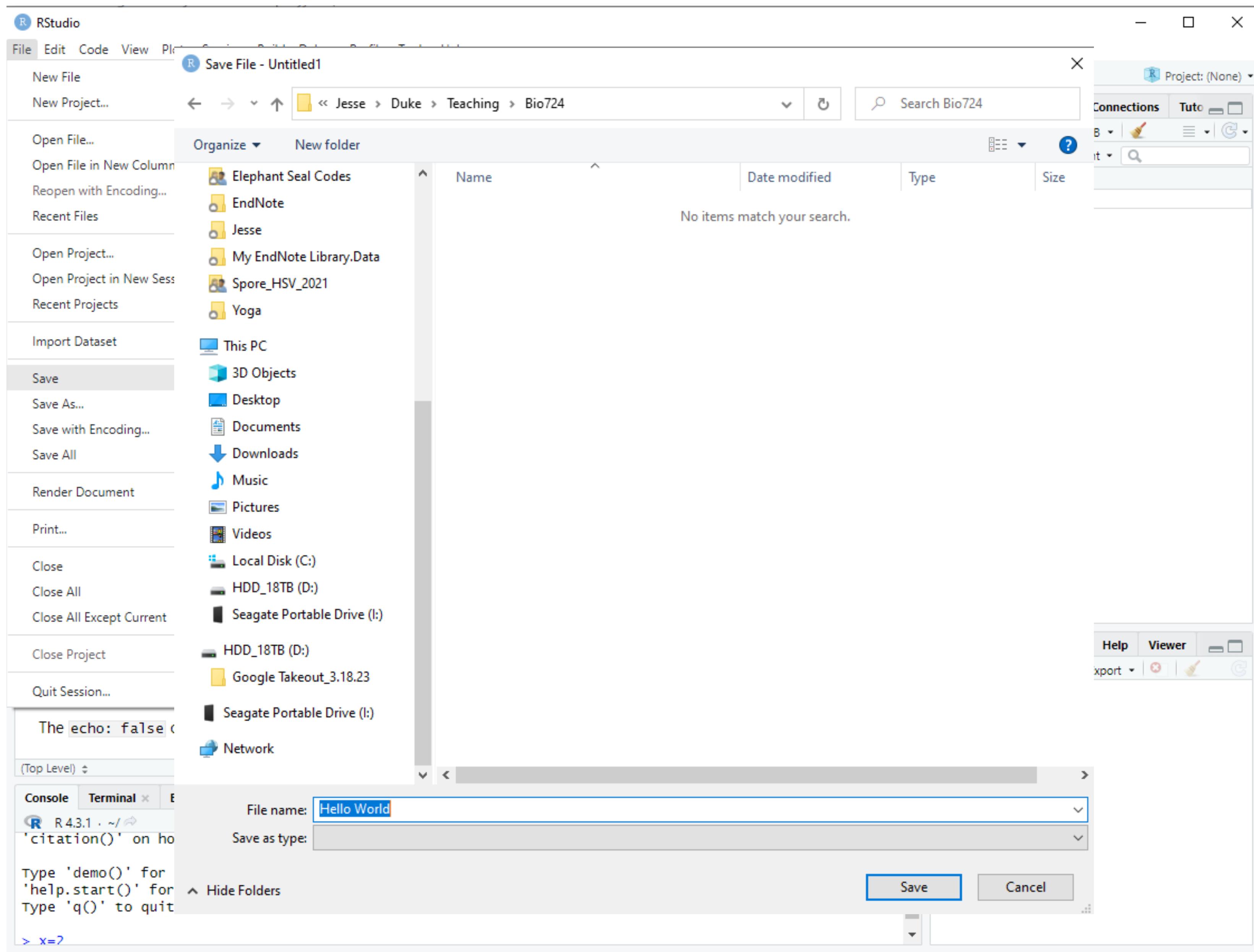
The right side of the interface shows the Environment, History, and Connections panes, and the Global Environment table which contains a single entry:

x	2
x	2

Quarto—Saving your document



Quarto—Saving your document



Quarto—Saving your document

Quarto—Saving your document

Class Exercise: Exit RStudio. Open up your file explorer, and locate the Quarto document you just saved.

What type of file is it (what letters appear after Hello World.)?
Once you've located it, click the quarto document to open it.

Quarto—Source vs Visual Editor

The screenshot shows the RStudio interface with the Quarto add-on installed. The main window displays a Quarto document titled "Hello World.qmd". The left pane shows the document's front matter:

```
---  
title: "Hello world"  
format: html  
editor: visual  
---
```

The right pane shows the "Quarto Running Code" environment, which includes a Global Environment viewer showing a variable "x" with value 2, and a "Files" panel.

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).

(Top Level) Quarto

Console Terminal Background Jobs

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.
```

Quarto—Source vs Visual Editor

The screenshot shows the RStudio interface with a Quarto document open. The main window displays the source code for a Quarto document named "Hello World.qmd". The code includes YAML front matter, a title, and content sections for Quarto and Running Code. It also shows examples of executable code blocks with and without the `echo: false` option. The right side of the interface features the Quarto visual editor, which shows the rendered content of the document, including a table of values and a sidebar with navigation links. The bottom of the screen shows the R console output, confirming the R version and platform.

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## Quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running Code  
12  
13 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:  
14  
15 ```{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ```{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

1:1 # Hello World

R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for Statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions

Quarto—What's in your document?

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the Quarto code in the 'Source' tab:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 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:  
14  
15 ```{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ```{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

The right pane shows the rendered output in the 'Quarto' tab, which includes the rendered text and the result of the code blocks:

values

x	2
---	---

Files Plots Packages Help Viewer

R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.

Quarto—What's in your document?

Text

The screenshot shows the RStudio interface with a Quarto document titled "Hello World.qmd". The document contains the following content:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
10 document. To learn more about Quarto see <https://quarto.org>.  
11  
12 ## Running code  
13  
14 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:  
15 `r`  
16 1 + 1  
17 `r`  
18  
19 You can add options to executable code like this  
20  
21 `r`  
22 #| echo: false  
23 2 * 2  
24 `r`  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

The code block from line 11 to 26 is highlighted with a blue rectangle.

The RStudio interface includes the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Source Editor:** Shows the Quarto document code.
- Quarto Status:** Running Code
- Environment Tab:** Shows the Global Environment with a value for 'x' set to 2.
- Console Tab:** Shows the R version information and the R command prompt.
- Plots, Packages, Help, Viewer Tabs:** Standard RStudio tabs.

Quarto—What's in your document?

Text
Chunks (Code)

The screenshot shows the RStudio interface with a Quarto document titled "Hello World.qmd". The document contains the following content:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
10 document. To learn more about Quarto see <https://quarto.org>.  
11  
12 ## Running code  
13  
14 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:  
15  
16 ```{r}  
17 1 + 1  
18 ```  
19 You can add options to executable code like this  
20  
21 ```{r}  
22 #| echo: false  
23 2 * 2  
24 ```  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

The code chunks are highlighted with a blue box. The first chunk contains explanatory text and a render button. The second chunk contains executable R code (1 + 1) and a render button. The third chunk contains executable R code (2 * 2) and a render button.

The RStudio interface includes a top menu bar, a toolbar with various icons, and several panes on the right side for Environment, History, Connections, and Tutorials. At the bottom, there are tabs for Console, Terminal, and Background Jobs, along with an R version status bar.

Quarto—Text

Text

The screenshot shows the RStudio interface with a Quarto document titled "Hello World.qmd". The "Source" tab is selected, displaying the following code:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
10 document. To learn more about Quarto see <https://quarto.org>.  
11  
12 ## Running code  
13  
14 when you click the **Render** button a document will be generated that includes both  
15 content and the output of embedded code. You can embed code like this:  
16  
17 `r 1 + 1`  
18  
19 You can add options to executable code like this  
20  
21 `r #| echo: false`  
22 `r 2 * 2`  
23  
24  
25 The `echo: false` option disables the printing of code (only output is displayed).  
26  
27
```

A blue rectangular box highlights the explanatory text from line 9 to line 14. The "Quarto" status bar at the bottom right of the code editor indicates "Running Code".

The "Console" tab at the bottom shows the R version information:

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

The "Global Environment" pane on the right shows a variable "x" with the value "2".

Quarto—Text

Text

The screenshot shows the RStudio interface with the Quarto welcome page open. The title "Welcome to Quarto" is displayed prominently. Below it, the text "An open-source scientific and technical publishing system" is followed by a bulleted list of features:

- Author using Jupyter notebooks or with plain text markdown in your favorite editor.
- Create dynamic content with Python, R, Julia, and Observable.
- Publish reproducible, production quality articles, presentations, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
- Share knowledge and insights organization-wide by publishing to Posit Connect, Confluence, or other publishing systems.
- Write using Pandoc markdown, including equations, citations, crossrefs, figure panels, callouts, advanced layout, and more.

Below the list, the text "Analyze. Share. Reproduce. You have a story to tell with data—tell it with Quarto." is displayed. At the bottom, there are two buttons: "Get Started" and "Guide".

The RStudio interface includes a top menu bar with File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. A toolbar below the menu contains icons for file operations like New, Open, Save, and Print. The bottom of the screen shows the R console output:

```
1:1 # Hello World
Console Terminal × Background Jobs ×
R 4.3.1 · ~/ ...
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
```

A sidebar on the right lists publishing options: Websites / Blogs, Articles, Presentations, Books, and Knowledge Repos. The "Websites / Blogs" option is highlighted with a blue border.

Quarto—Text

Text

The screenshot shows the RStudio interface with the Quarto Guide page open. The title "Guide" is at the top. Below it is a paragraph about the guide. The main content is organized into four columns:

Authoring	Computations	Tools	Documents
Create content with markdown	Execute code and display its output	Use your favorite tools with Quarto	Generate output in many formats
Markdown Basics	Using Python	JupyterLab	HTML
Figures	Using R	RStudio IDE	PDF
Tables	Using Julia	VS Code	MS Word
Diagrams	Using Observable	Neovim	Markdown
Citations & Footnotes	Execution Options	Text Editors	All Formats
Cross References	Parameters	Visual Editor	
Article Layout			
Presentations	Websites	Books	Interactivity
Present code and technical content	Create websites and blogs	Create books and manuscripts	Engage readers with interactivity
Presentation Basics	Creating a Website	Creating a Book	Overview
Revealjs (HTML)	Website Navigation	Book Structure	Observable JS
PowerPoint (Office)	Creating a Blog	Book Crossrefs	Shiny
Beamer (PDF)	Website Search	Customizing Output	Widgets
	Website Listings		Component Layout
Publishing	Projects	Advanced	
Publishing documents and sites	Scale up your work with projects	Refine documents with advanced tools	
Publishing Basics	Project Basics	Includes	

The RStudio interface includes a sidebar with file navigation, a top menu bar with File, Edit, Code, and a bottom console window showing R version 4.3.1.

Quarto—Text

The screenshot shows the Quarto interface within the RStudio environment. The main content area displays the 'Markdown Basics' guide, which includes an 'Overview' section and a quote from John Gruber about the readability of Markdown. It also links to Pandoc's full documentation. Below this, the 'Text Formatting' section provides examples of various Markdown syntaxes and their outputs.

Text

RStudio
File Edit Code
Guide > Authoring > Markdown Basics

Markdown Basics

Overview

Quarto is based on Pandoc and uses its variation of markdown as its underlying document syntax. Pandoc markdown is an extended and slightly revised version of John Gruber's [Markdown](#) syntax.

Markdown is a plain text format that is designed to be easy to write, and, even more importantly, easy to read:

A Markdown-formatted document should be publishable as-is, as plain text, without looking like it's been marked up with tags or formatting instructions. – [John Gruber](#)

This document provides examples of the most commonly used markdown syntax. See the [full documentation of Pandoc's Markdown](#) for more in-depth documentation.

Text Formatting

Markdown Syntax	Output
italics, **bold**, ***bold italics***	<i>italics</i> , bold , <i>bold italics</i>
superscript ² / subscript ₂	superscript ² / subscript ₂
~~strikethrough~~	strikethrough

1:1 # Hello
Console Terminal
R 4.3.1 ~/
R version 4.
Copyright (C)
Platform: x86_64
R is free software:
You are welcome to use, copy, and
distribute it, but please give credit
to the authors.

Project: (None) Tuto

Quarto—Text

The screenshot shows the RStudio interface with a Quarto document open. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains various icons for file operations like New, Open, Save, and Print, along with Go to file/function, Addins, and Render buttons.

The main workspace displays a Quarto configuration header:

```
---
```

```
title: "Hello world"
format: html
editor: visual
---
```

Below the header, the text "Quarto" is displayed in bold, followed by a descriptive paragraph:

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).

The bottom pane shows the R console output:

```
(Top Level) Quarto
Console Terminal Background Jobs
R 4.3.1 · ~/
```

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
```

Quarto—Text

The screenshot shows the RStudio interface with a Quarto document open. The main pane displays the following content:

```
---  
title: "Hello world"  
format: html  
editor: visual  
---
```

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).

The bottom pane shows the R console output:

```
(Top Level) Quarto  
Console Terminal Background Jobs  
R 4.3.1 · ~/  
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions
```

Quarto—Text

The screenshot shows the RStudio interface with a Quarto document open. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains various icons for file operations like New, Open, Save, and Render. A Quarto document titled "Hello World.qmd" is displayed in the main pane. The document content includes a YAML front matter section:

```
---  
title: "Hello world"  
format: html  
editor: visual  
---
```

Below the front matter, the document contains two sections: "Quarto" and "Running Code".

Quarto

Quarto enables you to weave together content and executable 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).

The bottom pane shows the R console output:

```
(Top Level) Quarto  
Console Terminal Background Jobs  
R 4.3.1 ~/  
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions
```

On the right side of the interface, there are several panes: Environment, History, Connections, Global Environment, Files, Plots, Packages, Help, and Viewer.

Quarto—Chunks

Chunks (Code)

The screenshot shows the RStudio interface with a Quarto document titled "Hello World.qmd". The document contains the following content:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 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:  
14  
15 ````{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ````{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

A code chunk starting at line 15 is highlighted with a blue rectangle. The RStudio interface includes the following components:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Source Editor:** Shows the Quarto code with syntax highlighting.
- Quarto Status:** Running Code
- Environment Tab:** Shows the Global Environment with a variable "x" set to 2.
- Console Tab:** Shows the R version information: R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts".
- Terminal Tab:** Shows the command "R 4.3.1 . ~/".
- Background Jobs Tab:** Shows no active jobs.
- Bottom Status:** Shows the R license: "R is free software and comes with ABSOLUTELY NO WARRANTY. You are welcome to redistribute it under certain conditions".

Quarto—Chunks

Chunks (Code)

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the Quarto code:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code in a single document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 when you click the **Render** button a document will be generated with both the rendered content and the output of embedded code. You can embed code like this:  
14  
15 ````{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ````{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

A context menu is open over the code at line 16, listing various execution options:

- Run Selected Line(s) (Ctrl+Enter)
- Run Current Chunk (Ctrl+Shift+Enter)
- Run Next Chunk (Ctrl+Alt+N)
- Run Setup Chunk Automatically (checked)
- Run All Chunks Above (Ctrl+Alt+Shift+P)
- Run All Chunks Below
- Restart R and Run All Chunks
- Restart R and Clear Output
- Run All (Ctrl+Alt+R)

The R console at the bottom shows the output of the code execution:

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.
```

Quarto—Chunks

Chunks (Code)

The screenshot shows the RStudio interface with a Quarto document titled "Hello World.qmd". The document contains the following content:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 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:  
14  
15 ````{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ````{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

A code chunk starting at line 15 is highlighted with a blue rectangle. The RStudio interface includes the following panels:

- Source**: The main code editor panel.
- Quarto**: A status bar at the top right of the Source panel.
- Environment**: Shows the variable `x` with value `2`.
- Files**, **Plots**, **Packages**, **Help**, **Viewer**: Standard RStudio navigation tabs.
- Console**: Shows the R version information and the command `R 4.3.1`.
- Terminal**: Shows the command `~/`.
- Background Jobs**: Shows no active jobs.

Quarto—Chunks

Chunks (Code)

The screenshot shows the RStudio interface with a Quarto document titled "Hello World.qmd". The "Source" tab is selected, displaying the following content:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 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:  
14  
15 ````{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ````{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

A blue box highlights the code block starting at line 15, and a blue circle highlights the "Run" button in the toolbar.

The "Console" tab at the bottom shows the R version information:

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)
```

The "Global Environment" pane on the right shows a variable "x" with the value "2".

Quarto—Rendering

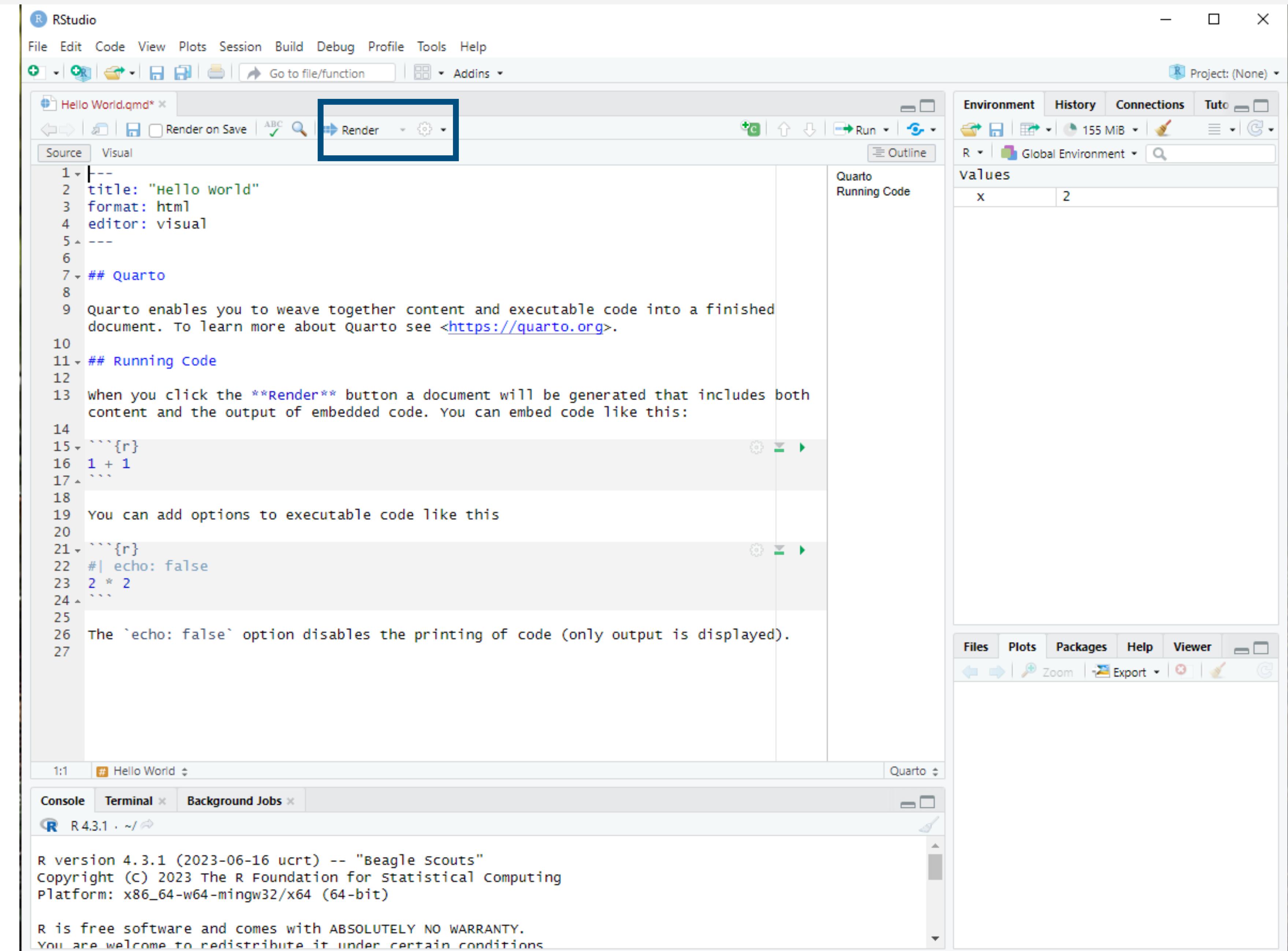
The screenshot shows the RStudio interface with a Quarto document open. The main window displays the Quarto code for a "Hello World" document, including sections for configuration, a header, and two code blocks demonstrating rendering and options.

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 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:  
14  
15 ```{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ```{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

The RStudio interface includes:

- Top Bar:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Go to file/function, Addins.
- Source Editor:** Shows the Quarto code with syntax highlighting.
- Quarto Status:** Shows "Quarto Running Code".
- Environment:** Global Environment table showing values x = 2.
- Files:** Files, Plots, Packages, Help, Viewer.
- Console:** Displays the R version information and the R command prompt.
- Bottom Status:** Shows the R version (R 4.3.1), the current working directory (~), and the license information.

Quarto—Rendering



The screenshot shows the RStudio interface with a Quarto document open. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar contains icons for file operations like New, Open, Save, and Print, along with Go to file/function and Addins. A blue box highlights the 'Render' button in the toolbar. The main workspace shows the Quarto code for 'Hello World.qmd'. The code includes a YAML front matter section and several code blocks demonstrating rendering options. The right pane displays the R Environment, showing a variable 'x' with a value of 2. The bottom pane shows the R Console output:

```
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"  
Copyright (C) 2023 The R Foundation for statistical computing  
Platform: x86_64-w64-mingw32/x64 (64-bit)  
  
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.
```

Quarto—Rendering

The screenshot shows the RStudio interface with a Quarto document open. The main window displays the Quarto code for a "Hello World" document, including sections for configuration, a header, and content. The code includes executable R code blocks and a note about the `echo: false` option. The RStudio environment pane shows a global variable `x` with a value of 2. The bottom console pane shows the R version information and the Quarto rendering command.

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together content and executable code into a finished  
document. To learn more about Quarto see <https://quarto.org>.  
10  
11 ## Running code  
12  
13 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:  
14  
15 ```{r}  
16 1 + 1  
17 ````  
18  
19 You can add options to executable code like this  
20  
21 ```{r}  
22 #| echo: false  
23 2 * 2  
24 ````  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

Environment History Connections Tuto

values

x	2
---	---

Files Plots Packages Help Viewer

R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions

Quarto—Rendering

The screenshot shows the RStudio interface with a Quarto document open. The main pane displays the following code:

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ## quarto  
8  
9 Quarto enables you to weave together text and data, creating a finished  
document. To learn more about how Quarto works, see our documentation.  
10  
11 ## Running code  
12  
13 when you click the **Render** button, Quarto will weave together the  
content and the output of the chunks.  
14  
15 ```{r}  
16 1 + 1  
17 ```  
18  
19 You can add options to executable code like this  
20  
21 ```{r}  
22 #| echo: false  
23 2 * 2  
24 ```  
25  
26 The `echo: false` option disables the printing of code (only output is displayed).  
27
```

A context menu is open over the code at line 13, specifically over the chunk starting with `## Running code`. The menu is titled "Use Visual Editor" and includes the following items:

- Preview in Window
- ✓ Preview in Viewer Pane (No Preview)
- ✓ Preview Images and Equations
- ✓ Show Previews Inline
- ✓ Chunk Output Inline
- Chunk Output in Console
- Expand All Output
- Collapse All Output
- Clear Output
- Clear All Output

The "Quarto" tab in the top right corner of the RStudio interface is highlighted. The "Global Environment" pane shows a variable `x` with the value `2`.

Quarto—Rendering

The screenshot shows the RStudio interface with a Quarto document open. The left pane displays the Quarto source code for "Hello World.qmd". A context menu is open over the code, specifically over the line containing the R code `1 + 1`. The menu is titled "Use Visual Editor" and includes options like "Preview in Window", "Preview in Viewer Pane", "Preview Images and Equations", "Show Previews Inline", "Chunk Output Inline", "Expand All Output", "Collapse All Output", "Clear Output", and "Clear All Output". The "Preview in Viewer Pane" option is checked.

The right pane shows the rendered output. It features a title "Hello World" and a section "Quarto" with the text: "Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>". Below this is a section "Running Code" with the text: "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:" followed by a code block containing the R command `1 + 1` and its output [1] 2. There is also a note about the `echo: false` option.

At the bottom, the R console output shows the R version information and the result of the `1 + 1` calculation.

```
1:1 # Hello World
Console Terminal × Background Jobs ×
R 4.3.1 · ~/ ...
R version 4.3.1 (2023-06-16 ucrt) -- "Beagle Scouts"
Copyright (C) 2023 The R Foundation for statistical computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
```

In Class Exercise 1 – groups of 4

- Take your Quarto Document and delete everything except the bit at the top encased in ---
- Make a header that says “Some basic math”
- Make a chunk in which you assign $x = 2$ and run the chunk.
- Make a new chunk and type in `x/2` and `x^2` and run the chunk
- Now change the value of x and re run both chunks
- Make a header that says “Mean height”
- Add some text in this section with the name of everyone in your group
- Make a vector that contains the height of everyone in your group in inches and name it “height”
- Calculate the mean of that vector
- Now use the following functions to calculate the Maximum height, minimum height, the sum, and the length of that vector: `max(height)` `min(height)` `sum(height)` `length(height)`
- Calculate the mean by taking the sum / the length

Data

The screenshot shows the RStudio interface with the following components:

- Quarto Document:** The left pane displays a Quarto document titled "Hello World.qmd". The code includes a YAML front matter block and an R code block that loads the "iris" dataset.
- Global Environment:** The top right pane shows the "Global Environment" tab, which lists the "iris" dataset as a 150 obs. of 5 variables.
- Console:** The bottom right pane shows the R console output:

```
R 4.3.1 · ~/↳
> mean(c(1,2,3,4))
[1] 2.5
> ?mean
> data(iris)
> data(iris)
> |
```
- Navigation:** The top bar includes tabs for Environment, History, Connections, and Tutorial.
- File Management:** The bottom bar includes tabs for Files, Plots, Packages, Help, Viewer, and Presentation.

Data

The screenshot shows the RStudio interface with the following components:

- Quarto Document:** The left pane displays a Quarto document titled "Hello World.qmd". The code includes a YAML front matter block and an R code block that loads the "iris" dataset.
- Global Environment:** The top-right pane shows the "Global Environment" tab with the "Data" section. It lists the "iris" dataset, which contains 150 observations and 5 variables.
- Console:** The bottom-right pane shows the R console output. It includes a session header for R 4.3.1, a command to calculate the mean of the numbers 1 through 4, and a help call for the "mean" function.

A blue circle highlights the "iris" dataset entry in the Global Environment Data section.

```
1 ---  
2 title: "Hello world"  
3 format: html  
4 editor: visual  
5 ---  
6  
7 ```{r}  
8 data(iris)  
9 ```  
10  
11 |
```

```
R 4.3.1 · ~/ ↗  
> mean(c(1,2,3,4))  
[1] 2.5  
> ?mean  
> data(iris)  
> data(iris)  
> |
```

Data

The screenshot shows the RStudio interface with the following components:

- Quarto Document:** A file named "Hello World.qmd" is open in the Source tab. The code includes a title, format, and editor specification, followed by an R block that loads the iris dataset.
- Console:** The R console shows a simple mean calculation: `> mean(c(1,2,3,4))` resulting in `[1] 2.5`.
- Global Environment:** The Data pane displays the "iris" dataset, which contains 150 observations and 5 variables.
- Annotations:** A blue circle highlights the "iris" entry in the Global Environment pane, with the label "View(iris)" positioned nearby.

View(iris)

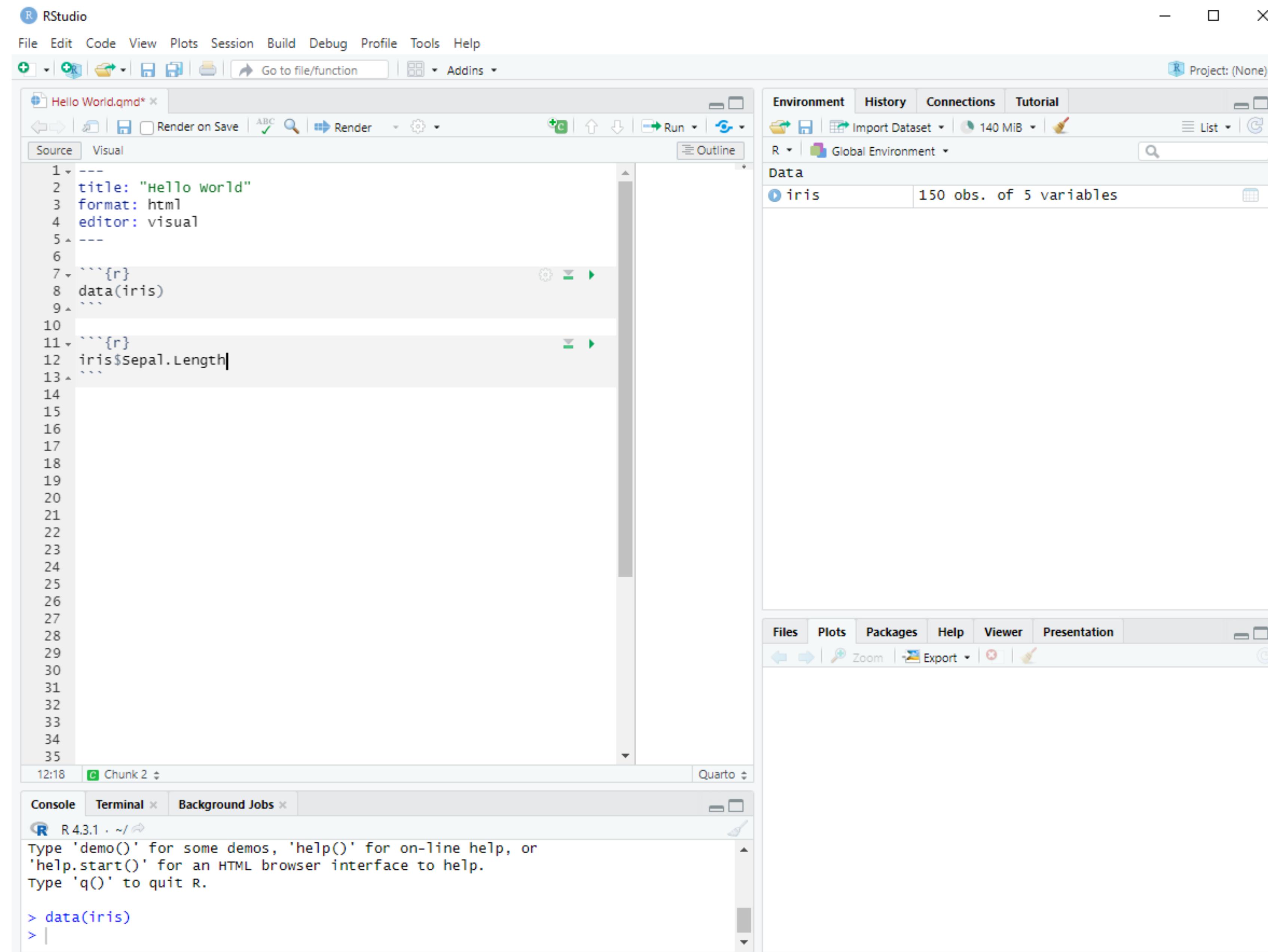
Data

The screenshot shows the RStudio interface with the following components:

- File Menu:** File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help.
- Toolbar:** Includes icons for New, Open, Save, Print, Go to file/function, and Addins.
- Environment Tab:** Shows the Global Environment with the iris dataset selected.
- Global Environment Tab:** Shows the iris dataset details: 150 obs. of 5 variables.
- Data View:** A data grid showing the first 23 rows of the iris dataset. The columns are Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species (setosa).
- Console Tab:** Displays R session history:

```
R 4.3.1 · ~/ 
[1] 2.5
> ?mean
> data(iris)
> data(iris)
> view(iris)
> |
```
- Plots, Packages, Help, Viewer, Presentation Tabs:** Located at the bottom of the interface.

Plots (A quick intro)



Plots (A quick intro)

The screenshot shows the RStudio interface with the following components:

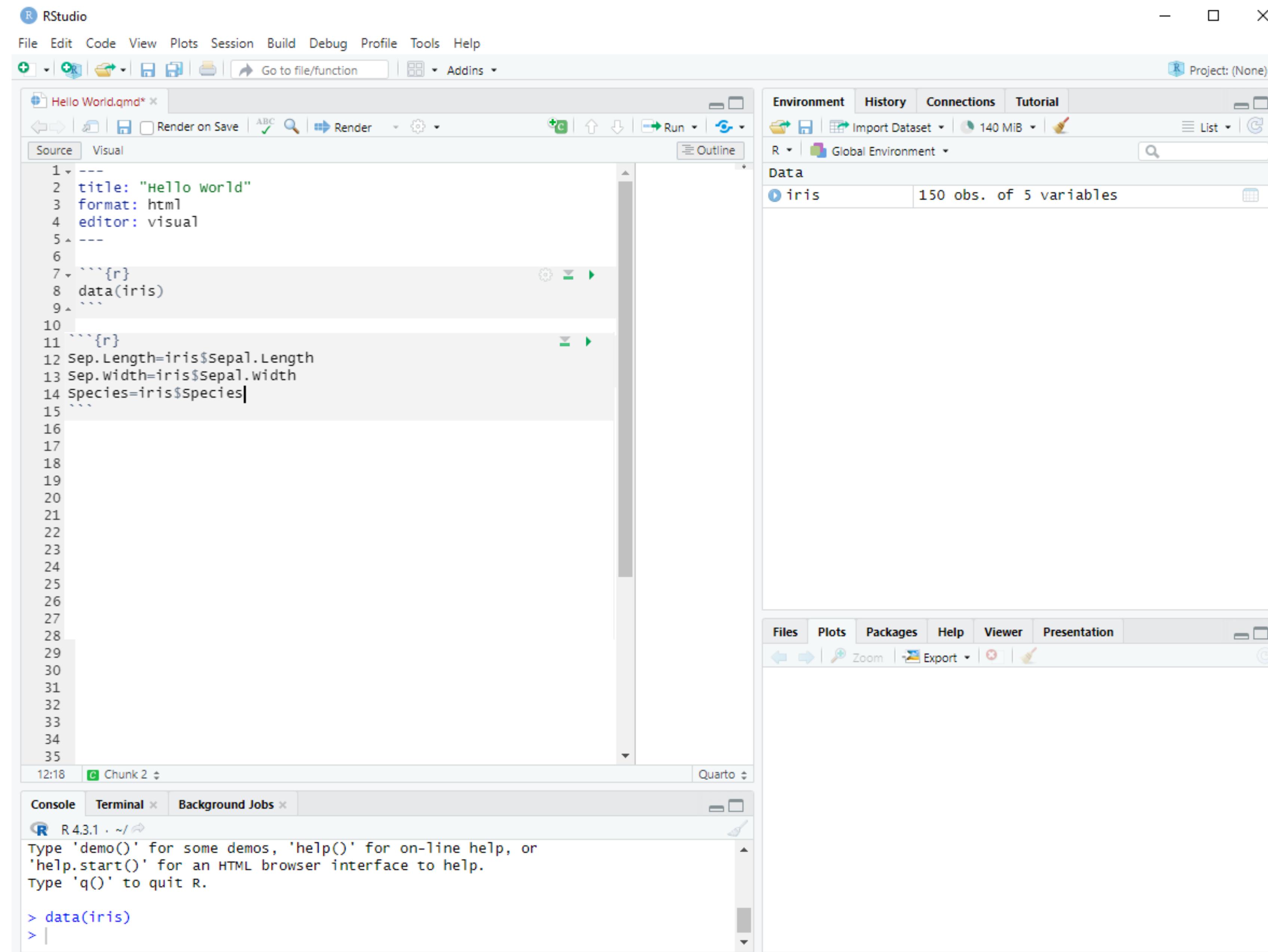
- Quarto Document:** The left pane displays a Quarto document titled "Hello World.qmd". It contains code for generating an HTML file with a title of "Hello world" and a visual representation of the "iris\$Sepal.Length" variable.
- Global Environment:** The top-right pane shows the "Global Environment" tab, which lists the "iris" dataset. The "iris" dataset is described as having 150 observations and 5 variables.
- Console:** The bottom-right pane is the "Console" tab, showing the R environment and the command `> data(iris)`.

Plots (A quick intro)

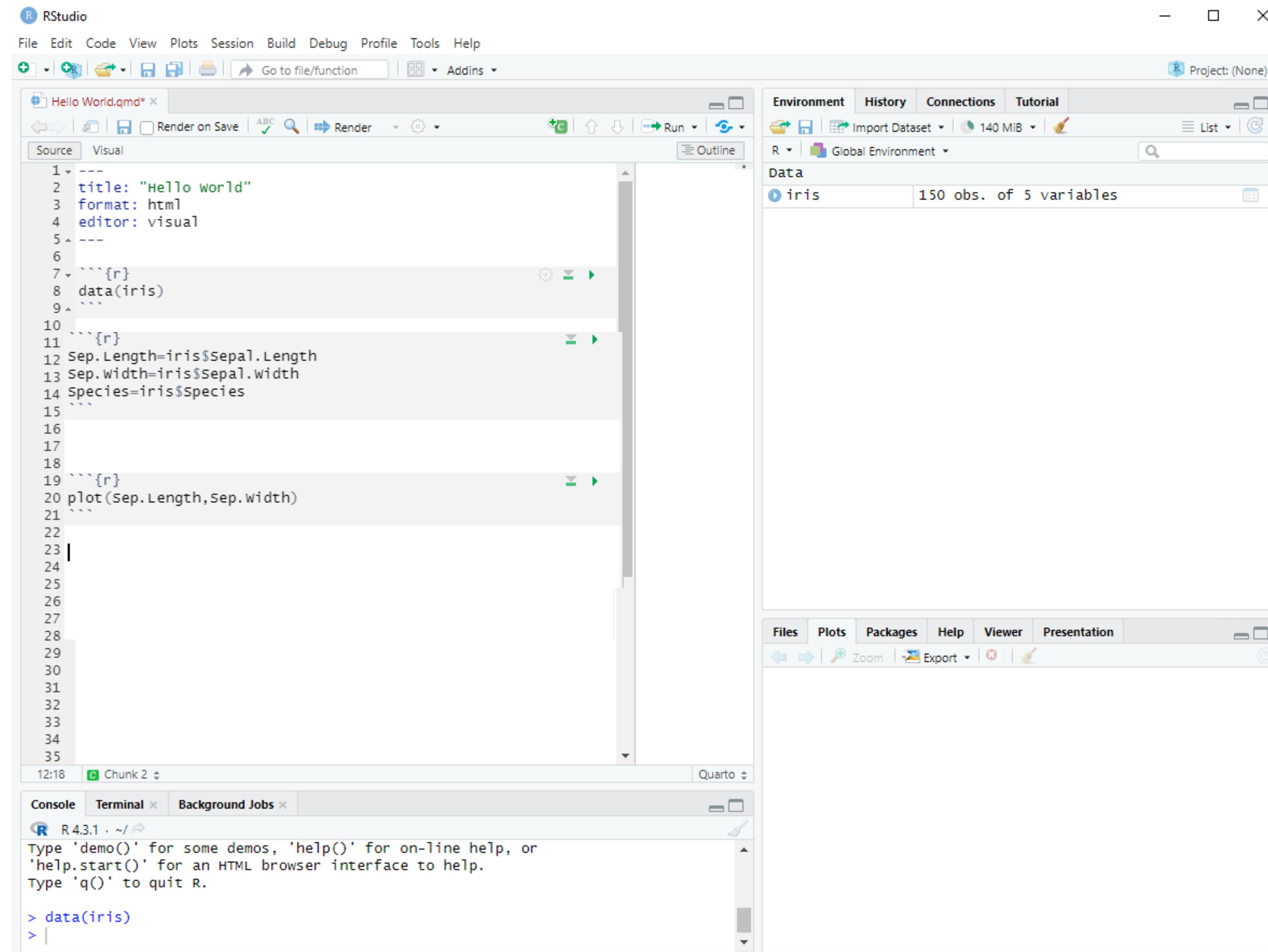
The screenshot shows the RStudio interface with the following components:

- Quarto Document:** The left pane displays a Quarto document titled "Hello World.qmd". The code includes a title, format, editor, and two code chunks. The second chunk contains the command `Sep.Length=iris\$Sepal.Length`.
- R Console:** The bottom pane shows the R console output for the `Sep.Length` assignment. It lists 150 observations of Sepal.Length from the iris dataset, ranging from 4.3 to 7.9.
- Data View:** The right pane shows the "Data" view with the "iris" dataset selected, displaying 150 observations of 5 variables.
- Environment View:** The top-right pane shows the "Environment" view with the "Global Environment" tab selected, listing the "iris" dataset.

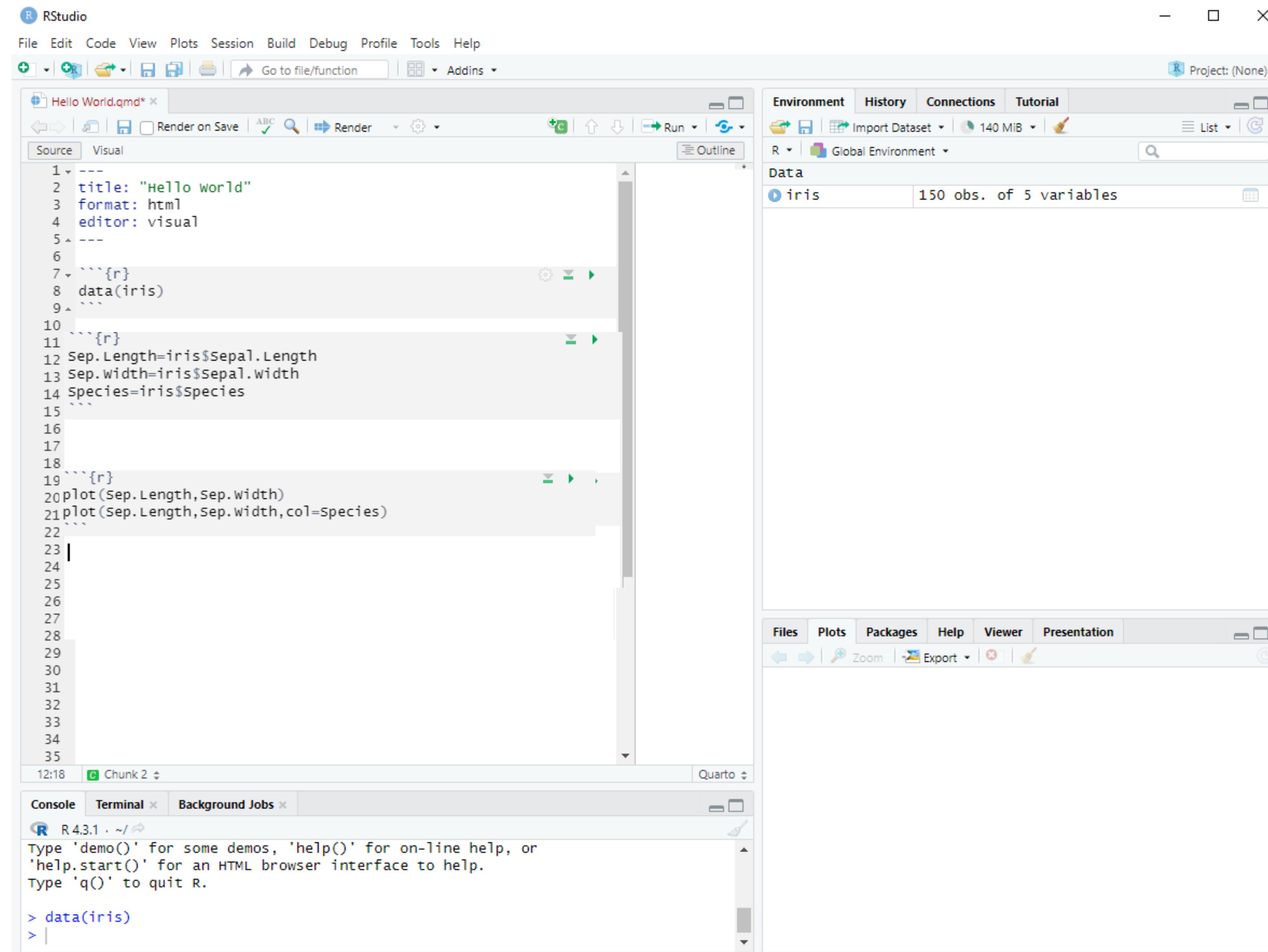
Plots (A quick intro)



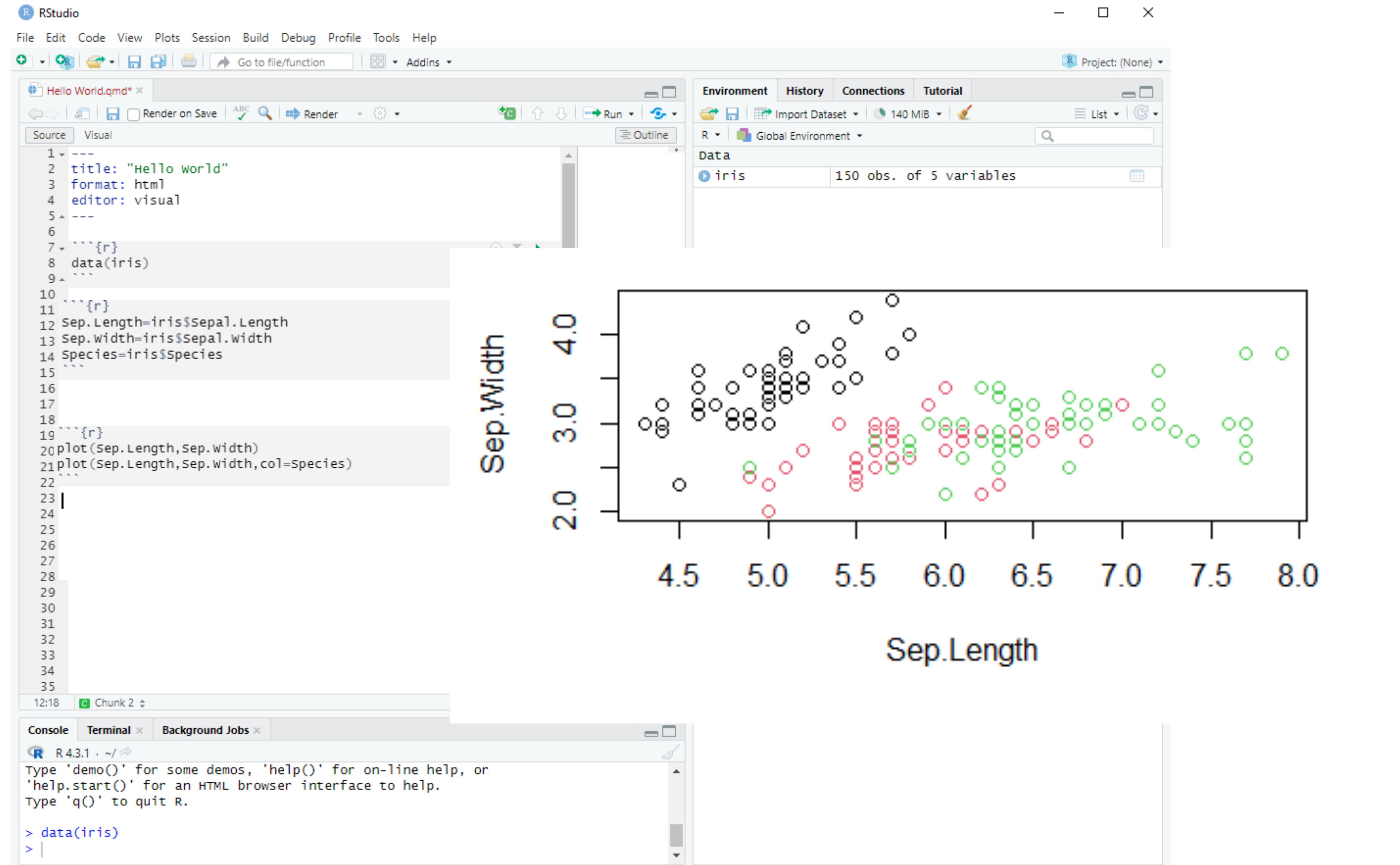
Plots (A quick intro)



Plots (A quick intro)



Plots (A quick intro)



In Class Exercise 2—groups of 4

- Make a new section with a header that says “Plots of class height and shoe size”
- Make a new vector with everyone’s shoe size (make sure it’s in the same order as the height vector from before. So, if the order in the height vector was Bob,Grant,Sue then the shoe size vector should also be Bob,Grant, Sue
- Make another vector where you catalog whether that shoe size was in MENS shoe size (“M”) or WOMENS shoe size (“W”) (or other “X”). Again, make sure the order is the same as before. Ex: `shoesizing=c("W","M","M","X")`
- Make a plot of your group’s heights versus their shoe size and color the shoe size by the type of sizing

More Reading

More reading/tutorial on Quarto: <https://quarto.org/docs/get-started/hello/rstudio.html>

Markdown Basics: <https://quarto.org/docs/authoring/markdown-basics.html>

