

First Steps with R and RStudio

Matt Steele

Welcome

- Matt Steele
- Data Cleaning and Analysis with R and RStudio
- Data Visualization and Presentation with R and RStudio
- Data Services Workshops

Resources

- [Official Site for R and Comprehensive R Archive Network \(CRAN\)](#)
- [Official Site for POSIT/RSTUDIO](#)
- [O'Reilly Learning Platform](#)
 - R for Data Science, 2nd Edition
 - R Programming for Statistics and Data Science
- [Postit/RStudio Youtube Page](#)
- [R for Data Science Youtube Page](#)

Why Use R and RStudio

- Open-source
 - Free
 - Platform independent
 - Reproducible
 - Shareable
 - Contains add-on packages
- Created for data statistical computation and graphic export

File Types

R Script: create code based text file that allows you to s

File > New File > R Script

R Markdown or Quarto Document: allows you to comb

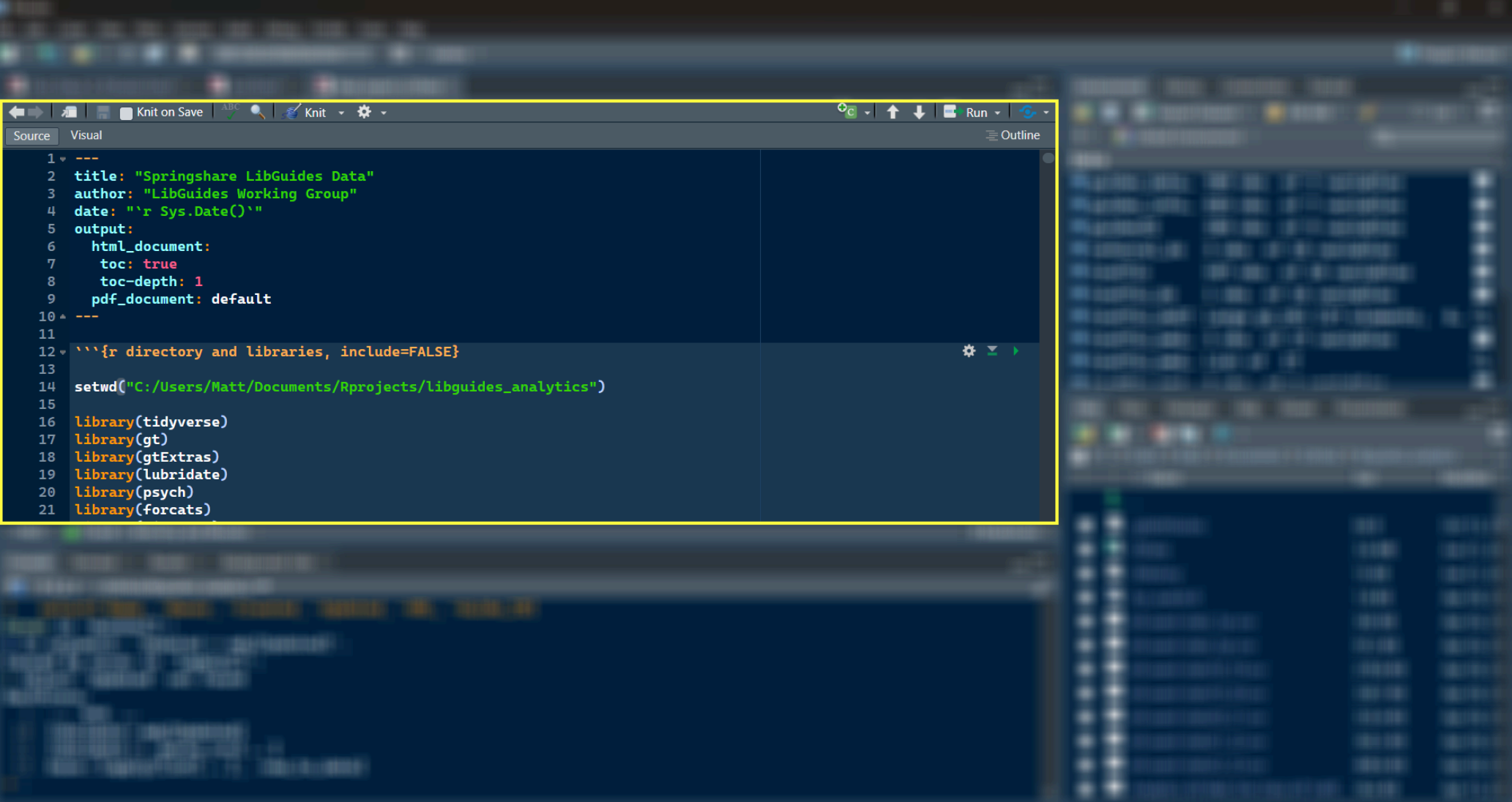
File > New File > R Markdown

File > New File > Quarto Document

RStudio

RStudio is an Integrated Development Environment (IDE) that allows you to save your code, store your variables and environments and view outputs.

Source Pane



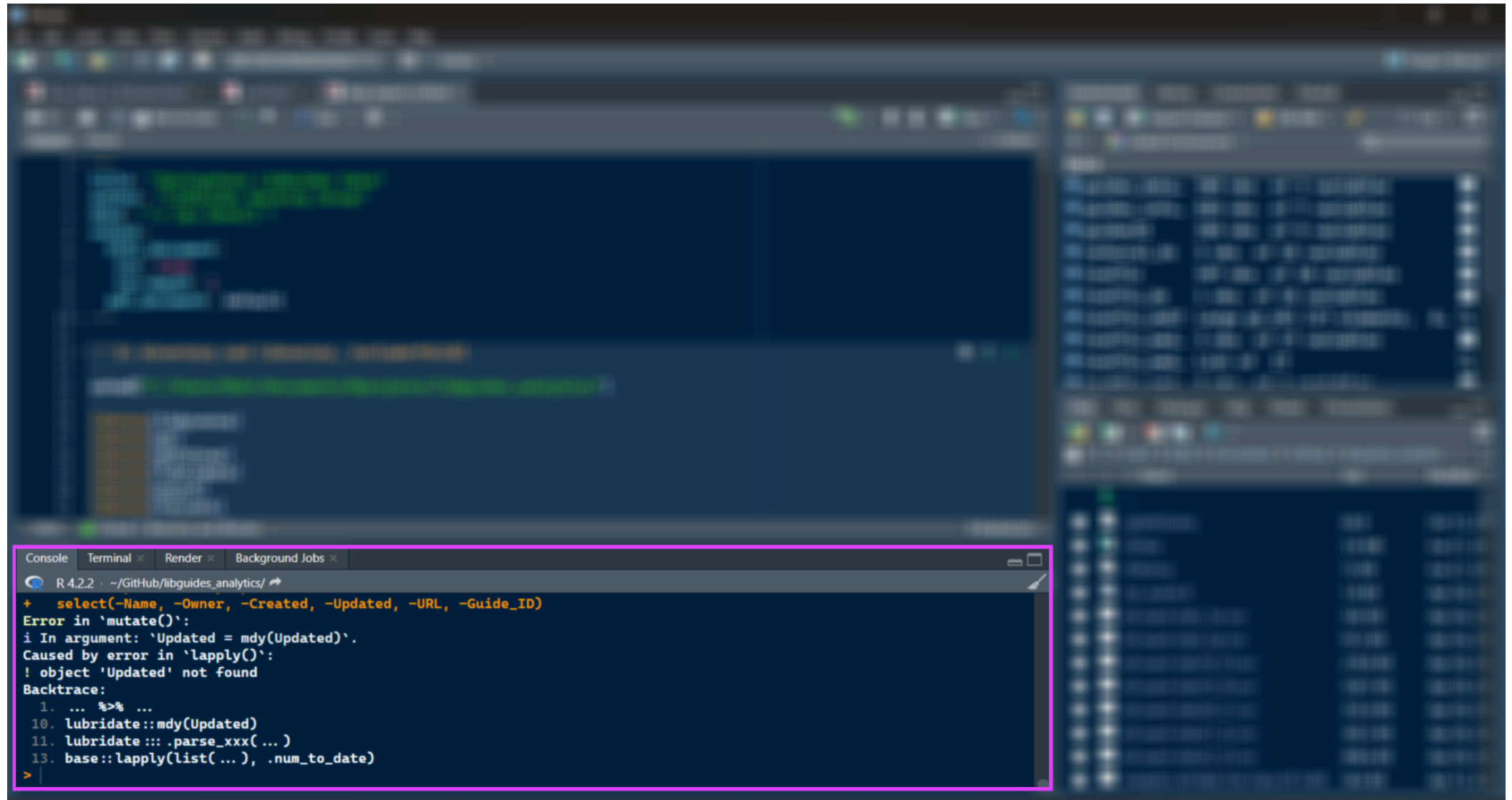
```
1 ---
2 title: "Springshare LibGuides Data"
3 author: "LibGuides Working Group"
4 date: "`r Sys.Date()`"
5 output:
6   html_document:
7     toc: true
8     toc-depth: 1
9   pdf_document: default
10 ---
11
12 ```{r directory and libraries, include=FALSE}
13
14 setwd("C:/Users/Matt/Documents/Rprojects/libguides_analytics")
15
16 library(tidyverse)
17 library(gt)
18 library(gtExtras)
19 library(lubridate)
20 library(psych)
21 library(forcats)
```

Source Pane

this pane is opened when you create or open a markdown or script file.

- This area is where you can create code in script or markdown files

Console Pane

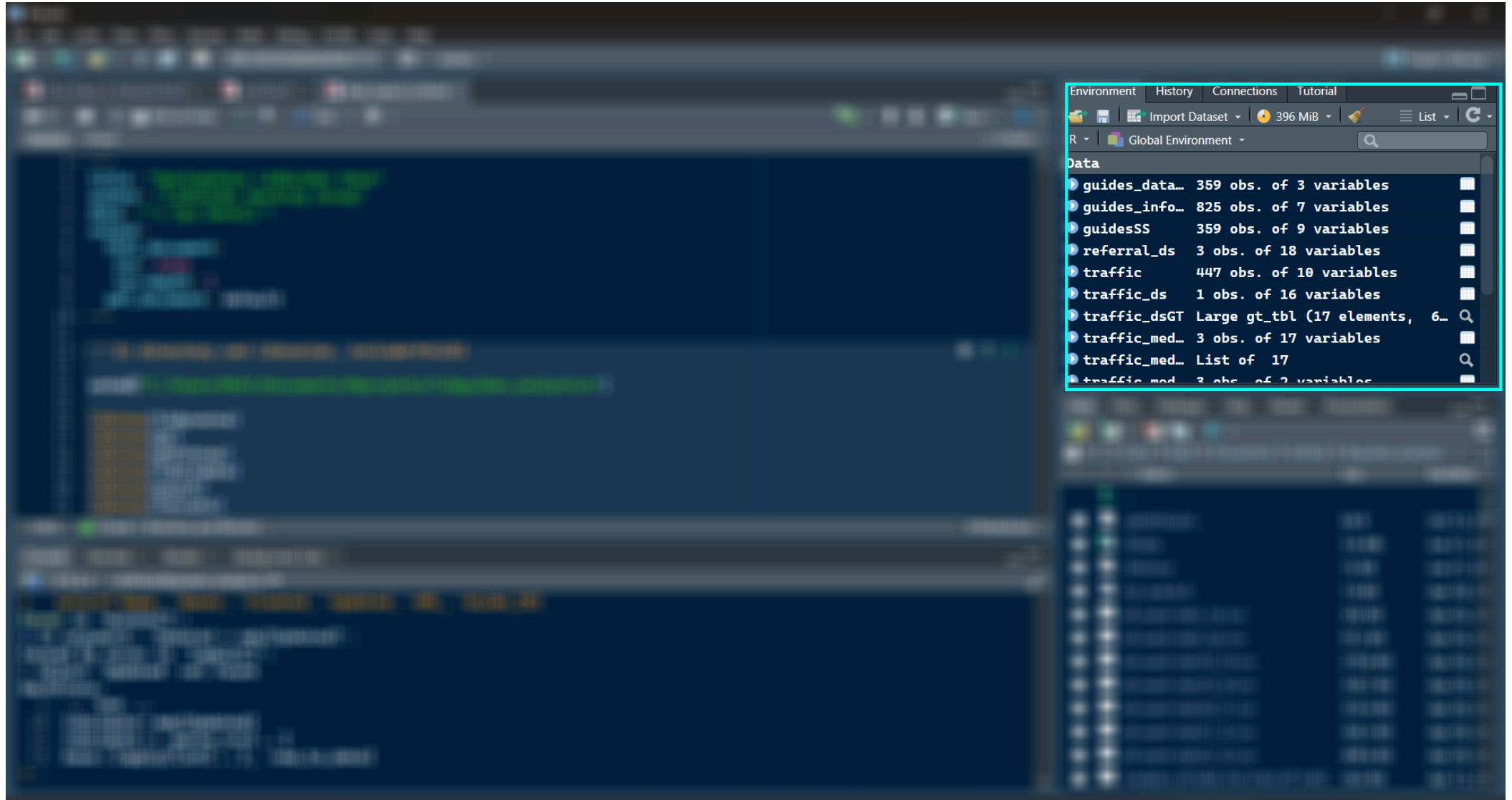


Console Pane

This is where you interact with the R. The results of your commands are displayed in this pane.

- Useful for testing code and exploring data

Environment Pane

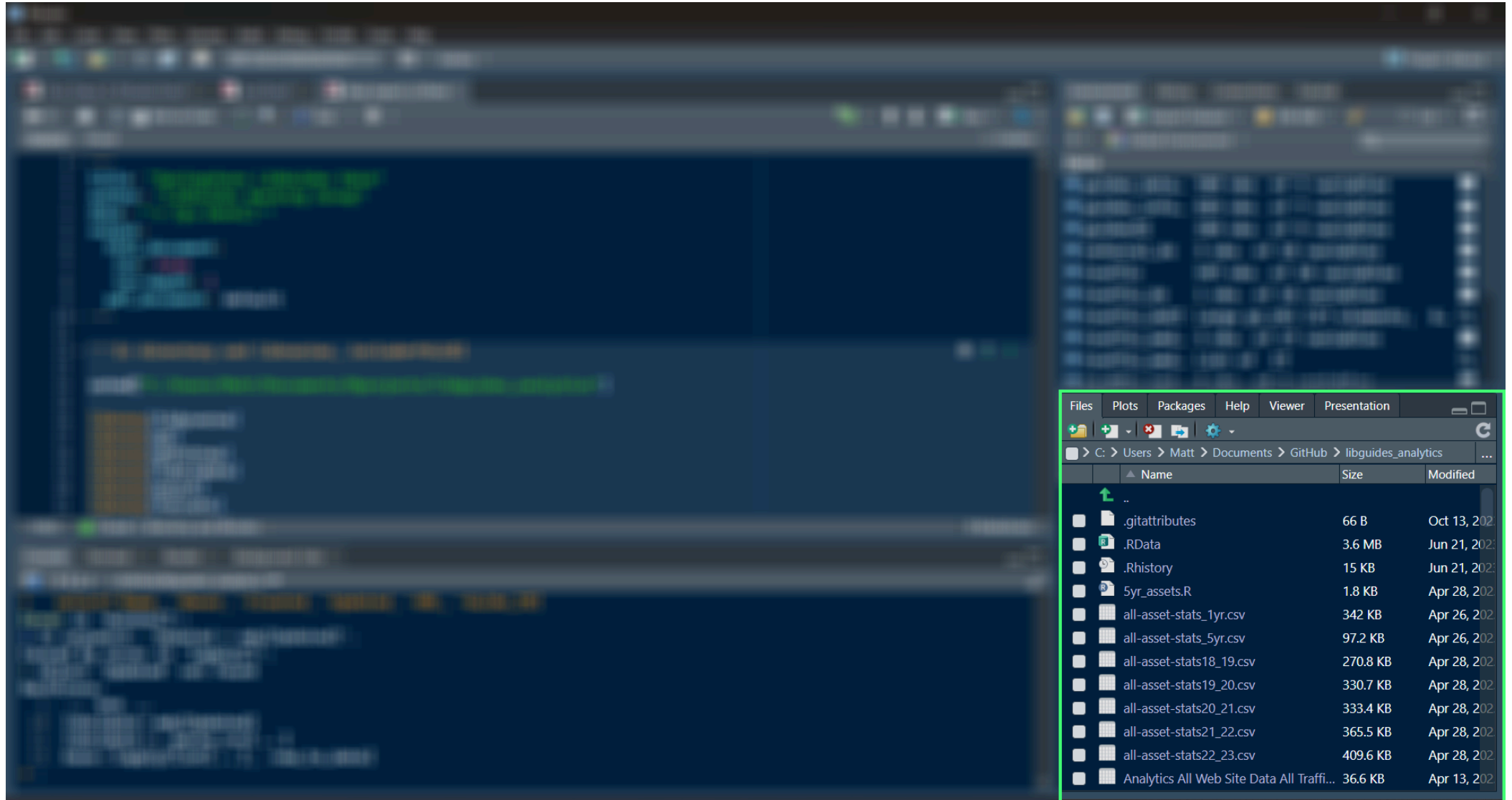


Environment Pane

view functions, objects, and data sets that are stored here

- Your environment can be saved and accessed at any point
- Save your environment to your working directory

Misc Pane



Misc Pane

view files, plots, packages, and get help

Set Your Preferences

Tools > Global Options

Some suggested Preferences to set:

- Code > Editing > Use Native Pipe Operator
- Code > Editing > Soft wrap source R files

Working Directory

the folder that you are working in, where you store your files, and where you load your files.

Session > Set Working Directory > Choose Directory

```
1 getwd() # show current directory that you are in
2
3
4 setwd("path/to/your/directory") # sets the working directory
```


Keyboard Shortcuts

Tools > Keyboard Shortcuts Help

	PC	MAC
Run Code	CTRL + ENTER	CMD + RETURN
Assignment Operator	ALT + -	OPTION + -
Pipe Operator	CTRL + SHIFT + M	CMD + SHIFT + M

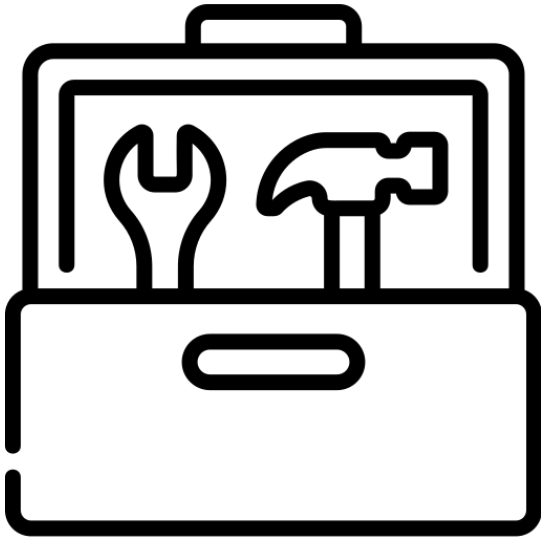
Comments

used to provide context, documentation, and explanations for the code.

```
1 mean(mtcars$mpg) # get the mean of the mpg variable
```

Functions

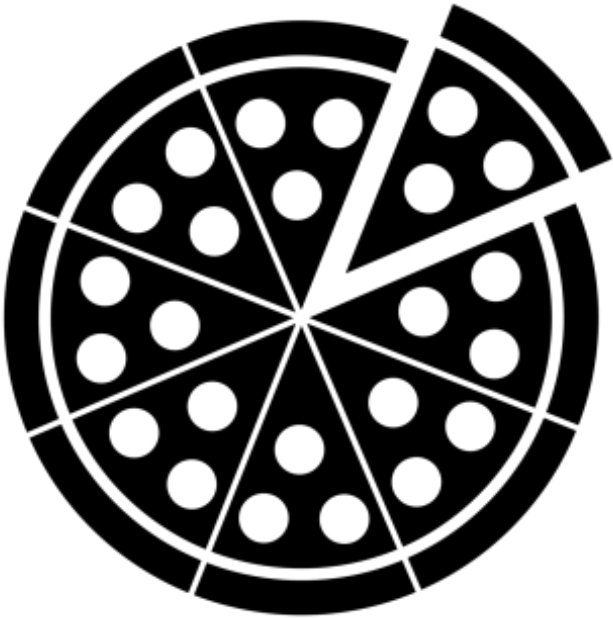
used to give commands to perform a tasks.



```
1 sample(1:5000)
2
3 mean(c(1, 3, 500))
4
5 str(400)
```

Arguments

used to provide the details about how your function operates



```
1 sample(1:5000, size = 50, replace = TRUE)
2
3 sample(1:5000, 50, TRUE)
```

Documentation

provides the necessary information, explanations, examples, and guidance to help you learn, understand, and effectively use R functions and packages.

- [Base R Function Documentation](#)
- [Dplyr Documentation](#)
- Get help with a package, function, or argument within R.
Using (?) or **help**("name of function")
- The **TAB** button will also provide help with suggestions and auto-filling options

Objects

used to store and work on data (numbers, words, tables, and more).

Assignment Operator (<-)

used to create an object

	PC	MAC
Assignment Operator	ALT + -	CMD + -

```
1 a <- 35
2 b <- 45
3
4 a
5 a + b
```

Naming Objects

- Use descriptive and meaningful names that indicate the purpose of the object
- Use lowercase letters.
- Use underscores to separate words (e.g., `my_variable_name`).
- Avoid using reserved words or functions (e.g., “if,” “else,” “for,” “function”).

Data Types

type	values
Double or Numeric	used for numbers which can be integers (
Character	used for text, words, and strings of chara
Factor	used to represent categorical data with p
Date	used for handling dates, times, and time
Boolean	used for decision-making and represente

Packages

R packages are like toolkits or collections of pre-built functions, data sets, and tools that extend the capabilities of the R programming language.

Packages

Install

You must install a package before you can load it. *But you only need to install it one time.*

```
1 install.packages("tidyverse")
```

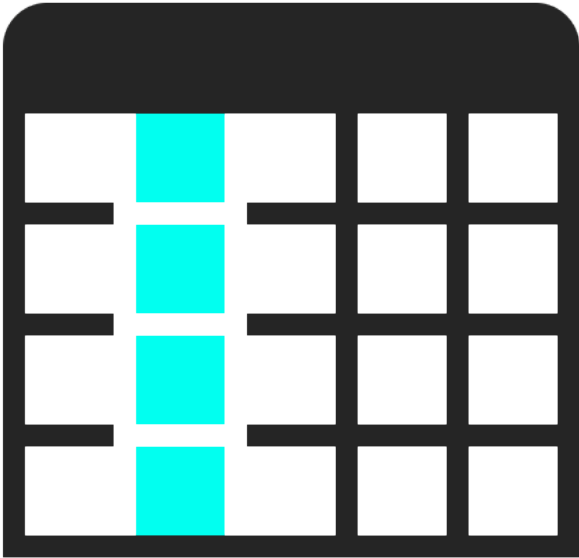
Load

For every new session, you must load the package to use the it's functions.

```
1 library(tidyverse)
```

Vectors

ordered collections of data items of the same type.



```
1 vec_one <- c(1, 2, 3)
2 vec_two <- c(4:6)
3
4 vec_two
```

Data Frame

two-dimensional sequence of data variables (columns) and observations (rows). While each variable in a data frame typically contains data of the same type, different variable can contain different data types.

```
1 # create vectors
2 title <- c("Star Wars", "The Empire Strikes Back", "Return of the Jedi")
3 year <- c(1977, 1980, 1983)
4 length.min <- c(121, 124, 133)
5 box.office.mil <- c(787, 534, 572)
6
7 # combine these vectors with the data.frame() function
8
9 starWars.data <- data.frame(title, year, length.min, box.office.mil)
10 starWars.data
```

Subsetting Variables

used to select and work with specific variables (columns) from a data frame.

```
1 starWars.data$year
```

Export / Save Data

Once you are done entering your data, you can export it to your working directory. The function without built-in arguments is **write.table()** but if are saving it as a csv, you are better using **write.csv()**.

```
1 write.csv(starWars.data, "starwars.csv")
```

Load Data

load data from a file in your working directory using the **read_csv()** function from tidyverse. There is also the **read.csv()** function in base R.

```
1 fight_songs <- read_csv("fight-songs.csv")
```


Explore Data

```
1 view(fight_songs) # view in a new tab
2
3 nrow(fight_songs) # number of rows
4
5 ncol(fight_songs) # number of columns
6
7 str(fight_songs) # structure of data frame
```

Descriptive Statistics

```
1 summary(fight_songs) # various descriptive statistics
2
3 sum(fight_songs$number_fights) # sum a variable
4
5 mean(fight_songs$number_fights) # average a variable
6
7 median(fight_songs$number_fights) # middle value of a variable
8
9 sd(fight_songs$number_fights) # standard deviation of a variable
10
11 min(fight_songs$number_fights) # minimum value of a variable
12
13 max(fight_songs$number_fights) # maximum value of a variable
```

Conclusion

- Matt Steele
- Data Cleaning and Analysis with R and RStudio
- Data Visualization and Presentation with R and RStudio
- Data Services Workshops

