R COOKBOOK – R STUDIO BASICS

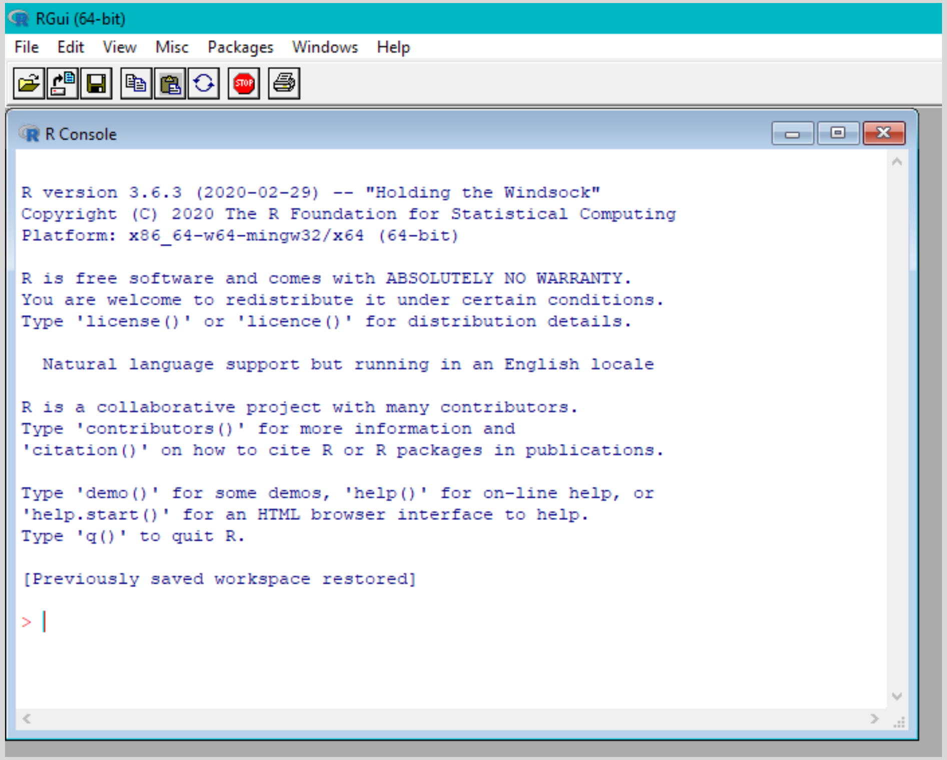
Robbie Stancil – CISER GRA

Session Info

- Today we’ll cover some of the basics in R
- Want to learn even more? Join in on another session:
  - 3/10 – Tidy Data in R
  - 3/24 – Data Visualization using ggplot2
  - 3/31 – Reproducible Reports using RMarkdown
- Have questions? Feel free to interrupt and ask!

R vs RStudio

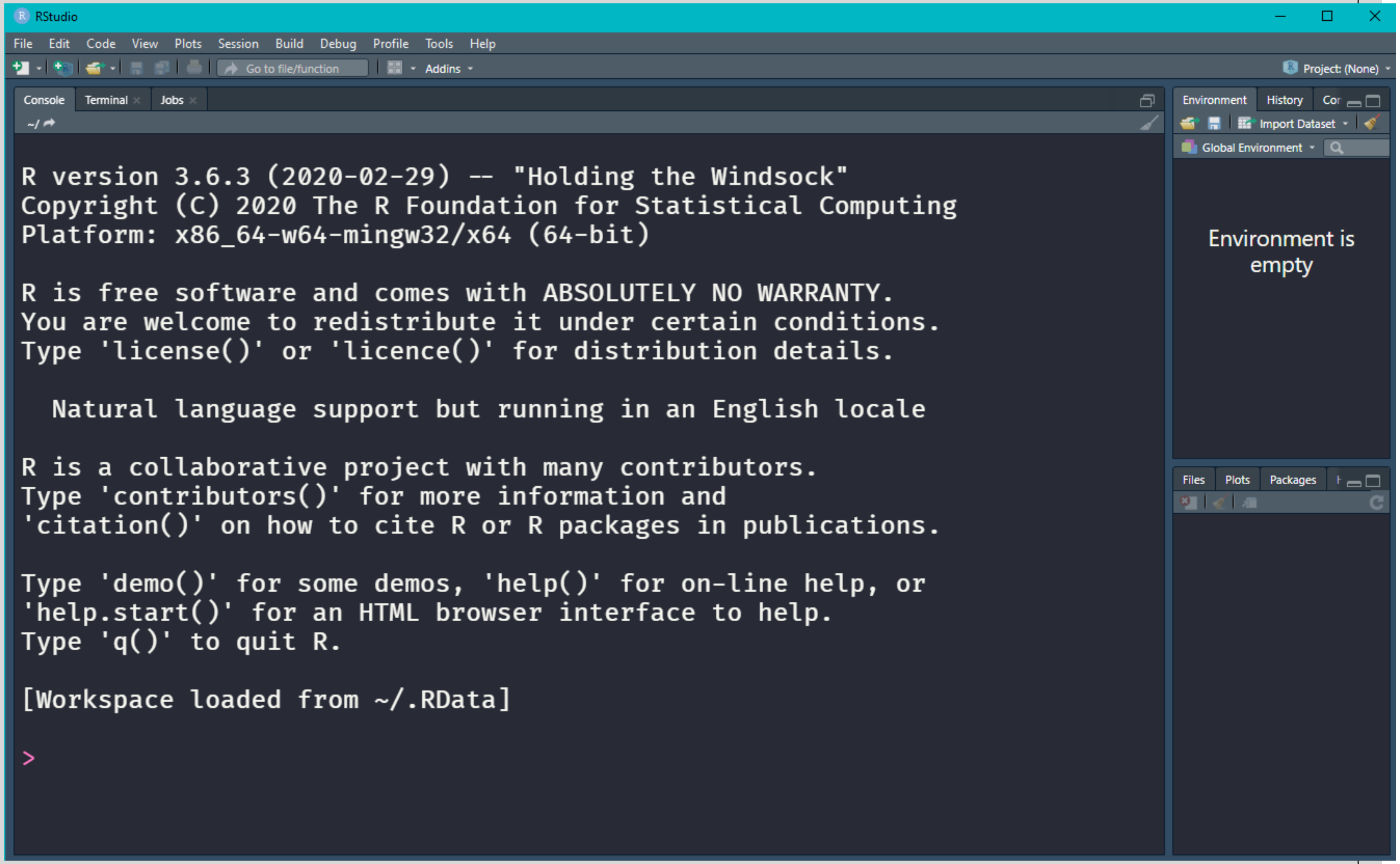
- R is the programming language we will be using
- R comes with an environment we can use to program



```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 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.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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

[Previously saved workspace restored]

> 
```

R vs RStudio

- RStudio adds a more user-friendly graphical interface
- RStudio requires R to work



```
R version 3.6.3 (2020-02-29) -- "Holding the Windsock"
Copyright (C) 2020 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.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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

[Workspace loaded from ~/.RData]

>
```

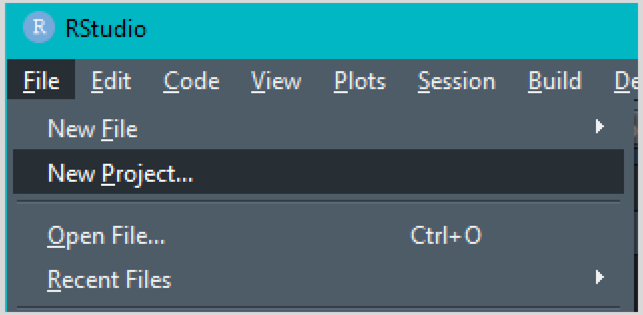
Environment is empty

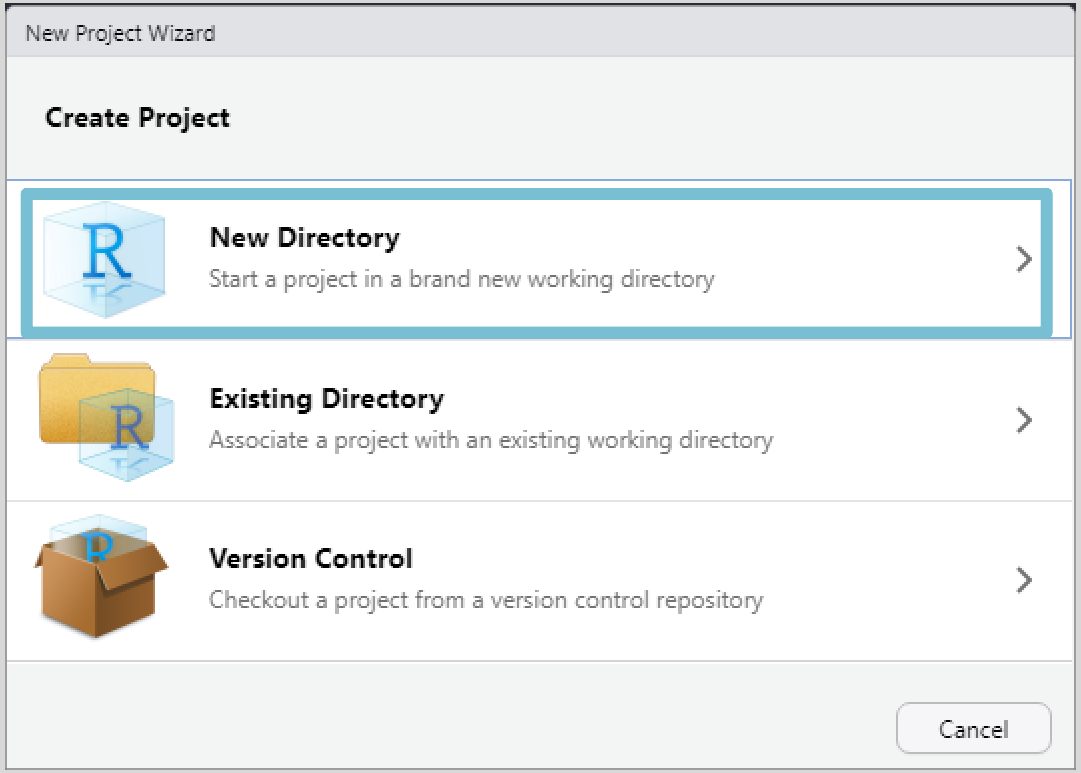
Why Choose R?

- Free and Open Source
- Wide range of packages
- Easy graphics
- Presenting results in RMarkdown

Setting Up Our Project

- Open RStudio




New Project Wizard

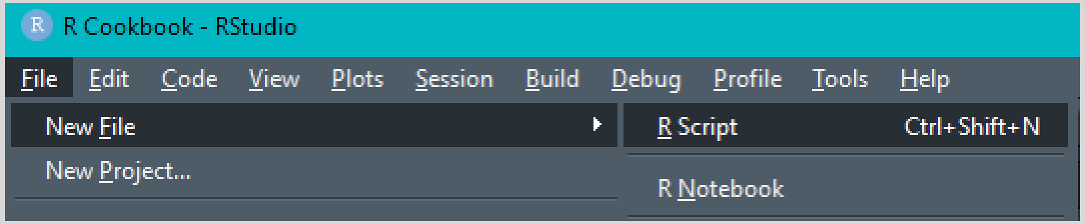
Create Project

New Directory
Start a project in a brand new working directory

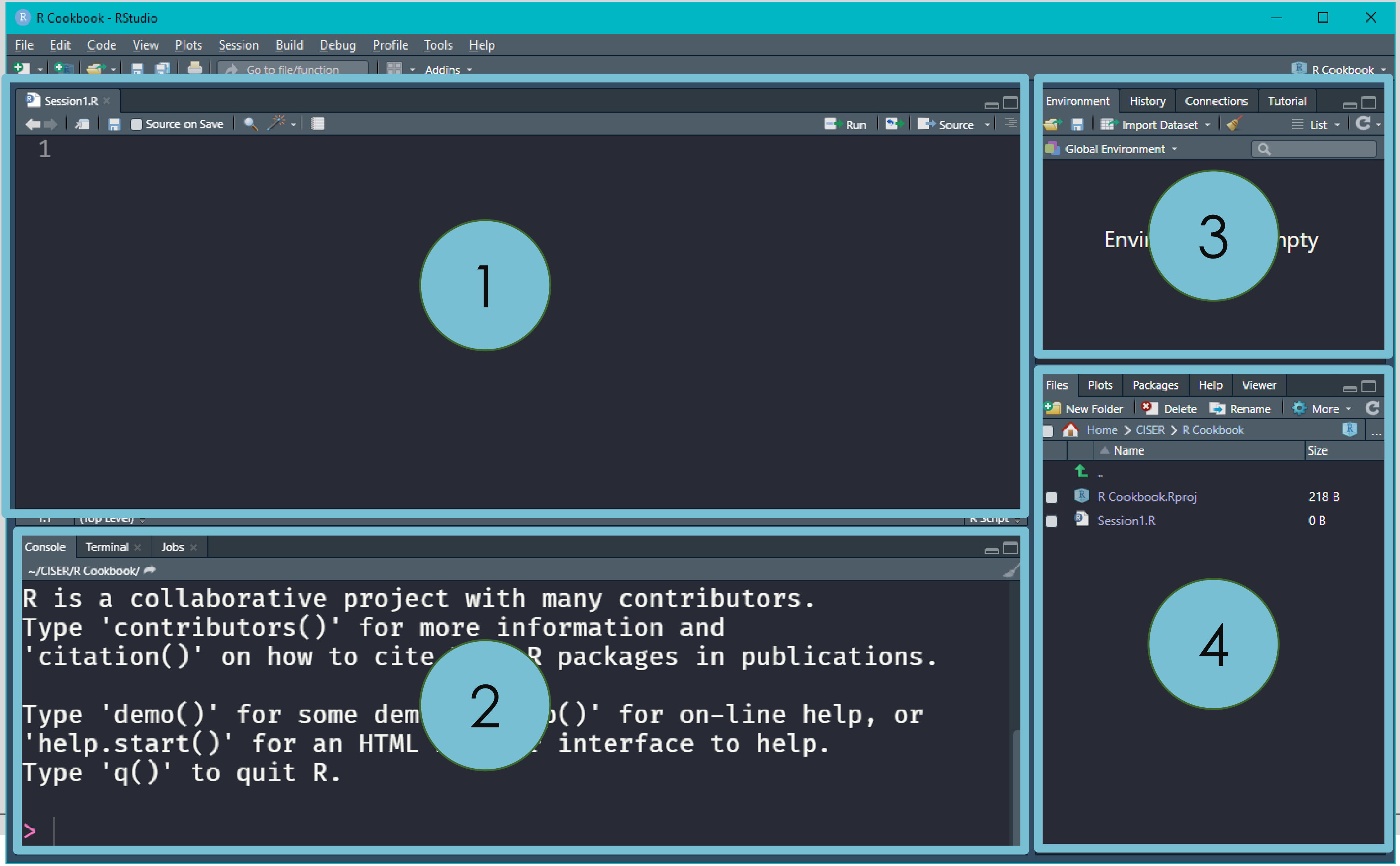
Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Create an R Script



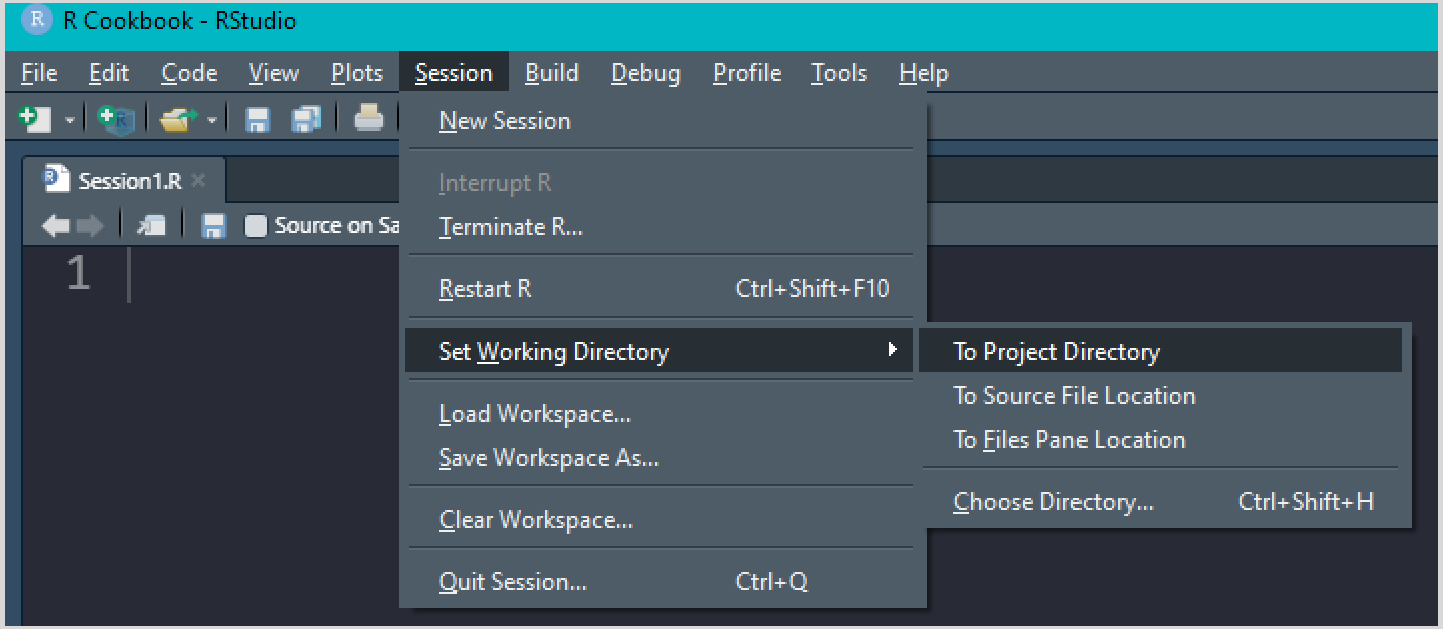
Rundown of the Interface



```
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite           R packages in publications.

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

>
```

Current Working Directory

- Always make sure RStudio is accessing the directory you want!



Operators in R

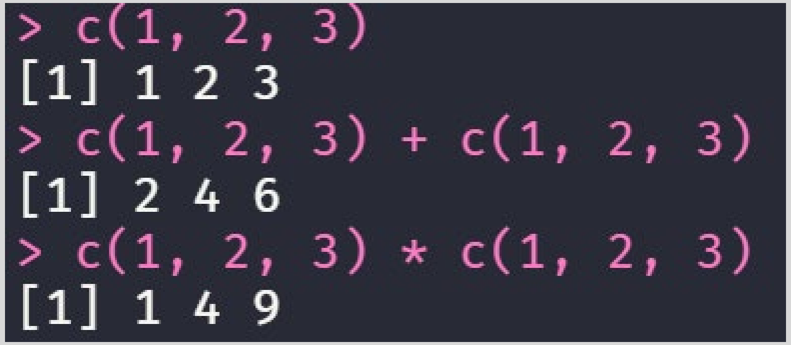
- Operator: Symbol that tells R what to do
- We'll focus on Arithmetic Operators, Assignment Operators, and Extraction Operators
- More info: https://www.tutorialspoint.com/r/r_operators.htm

Arithmetic Operators

- + : Addition
- - : Subtraction
- * : Multiplication
- / : Division
- ^ : Exponentiation

Vector Notation

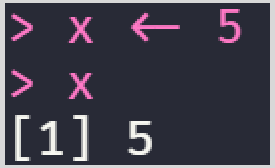
- We can create a vector by surrounding the values we want in the vector by c()



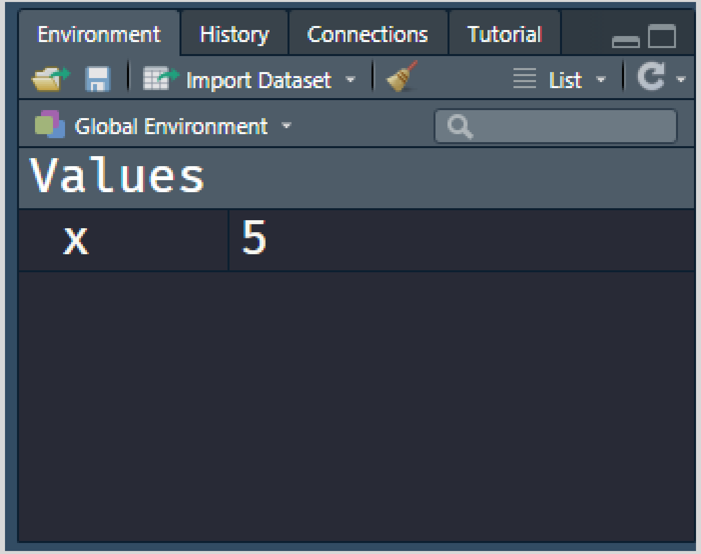
```
> c(1, 2, 3)
[1] 1 2 3
> c(1, 2, 3) + c(1, 2, 3)
[1] 2 4 6
> c(1, 2, 3) * c(1, 2, 3)
[1] 1 4 9
```

Assignment Operator

- To store a value, we save it in a variable by assigning it
- This is done using ‘<-’ (shortcut alt-dash)



```
> x ← 5
> x
[1] 5
```



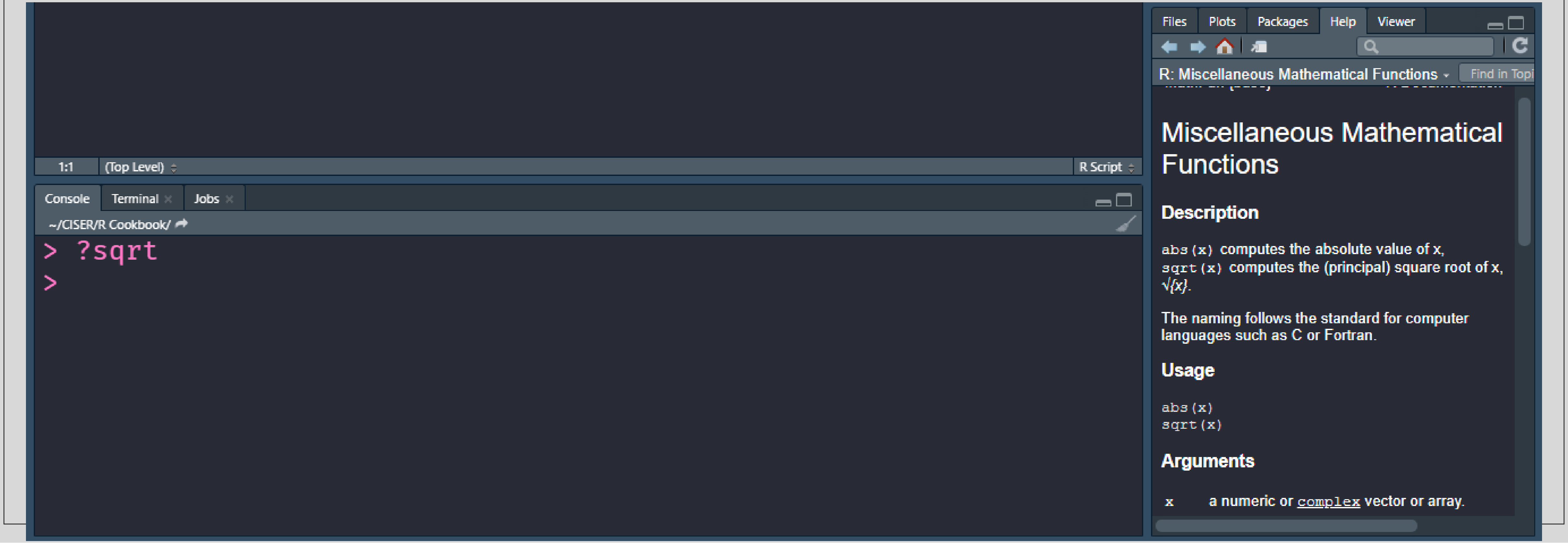
```
Values
 x        5
```

Functions

- A function is basically a piece of code that performs a specific task
- Can take input through arguments
- Can return values as output
- Examples: sum, sqrt

Function Help

- Not sure what a function does or how it works?



```
> ?sqrt
>
```

Miscellaneous Mathematical Functions

Description

`abs(x)` computes the absolute value of x, `sqrt(x)` computes the (principal) square root of x, √{x}.

The naming follows the standard for computer languages such as C or Fortran.

Usage

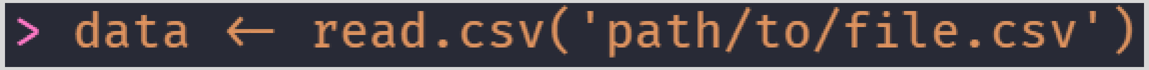
```
abs(x)
sqrt(x)
```

Arguments

`x` a numeric or `complex` vector or array.

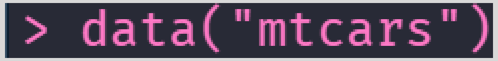
Reading in Data

- Loading from a csv file:

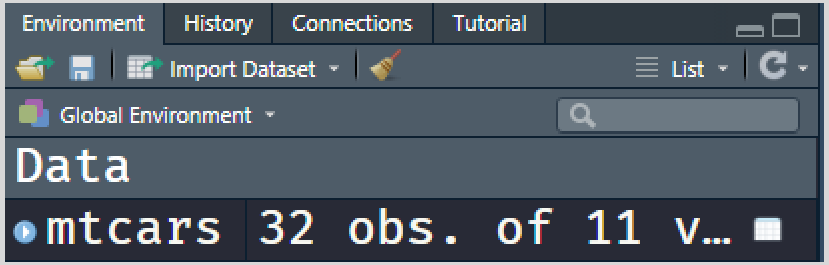


```
> data ← read.csv('path/to/file.csv')
```

- Other examples: http://www.r-tutor.com/r-introduction/data-frame/data-import



```
> data("mtcars")
```



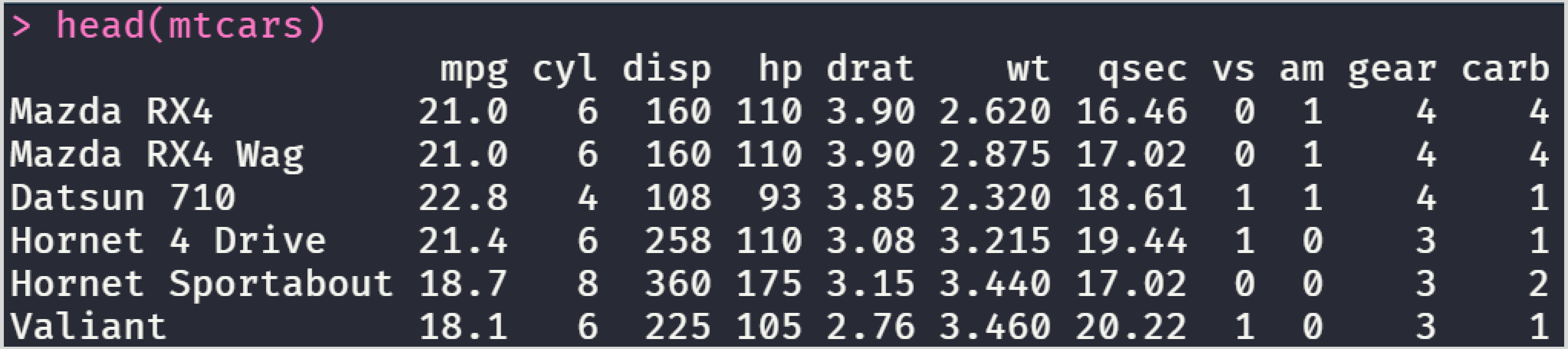
Data

```
mtcars 32 obs. of 11 v…
```

Data Frames in R

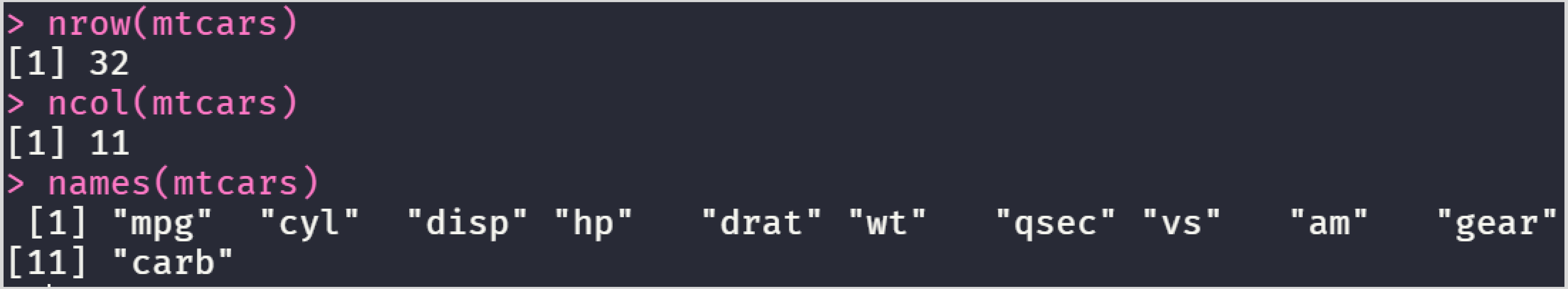
- Data in R is usually stored in a Data Frame
- These consist of
  - Rows: these are the separate observations
  - Columns: these are the variables that were recorded

Examining Data Frames



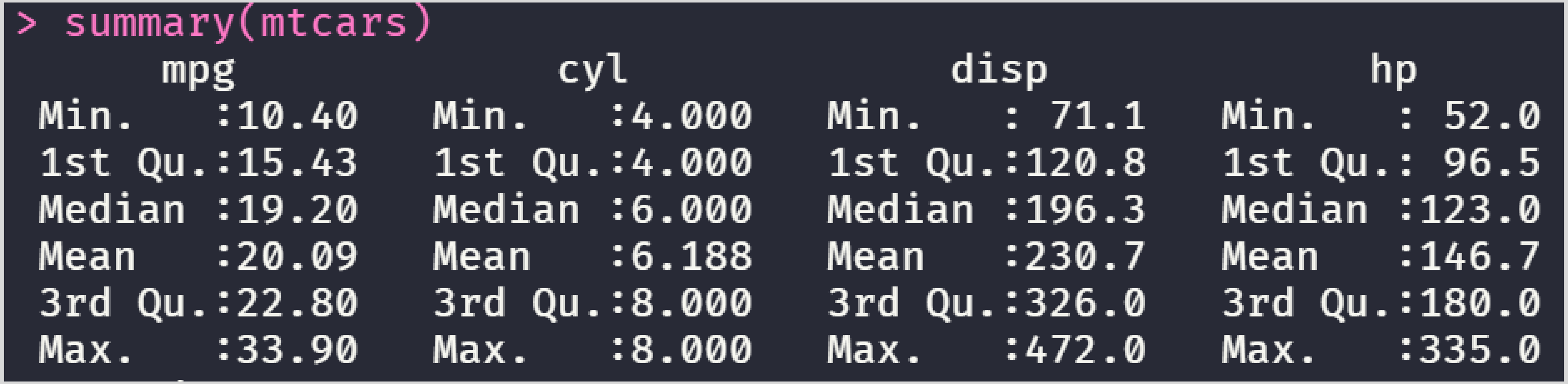
```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Examining Data Frames



```
> nrow(mtcars)
[1] 32
> ncol(mtcars)
[1] 11
> names(mtcars)
 [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
[11] "carb"
```

Examining Data Frames



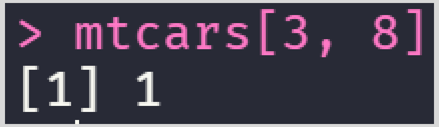
```
> summary(mtcars)
      mpg             cyl             disp             hp       
 Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
 1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
 Median :19.20   Median :6.000   Median :196.3   Median :123.0  
 Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
 3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
 Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
```

Accessing Data

- How do we access specific data?
- Use the extraction operator: []
- Format: dataframe[row, column]

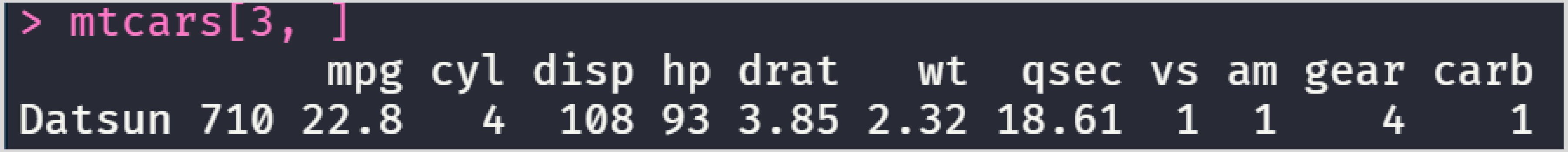
Accessing Data

- Specific Cell:



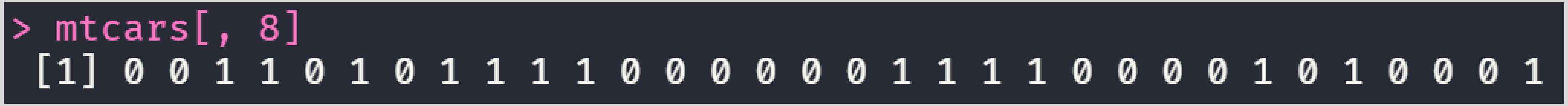
```
> mtcars[3, 8]
[1] 1
```

- Entire Row:



```
> mtcars[3, ]
            mpg cyl disp hp drat   wt  qsec vs am gear carb
Datsun 710 22.8   4  108 93 3.85 2.32 18.61  1  1    4    1
```

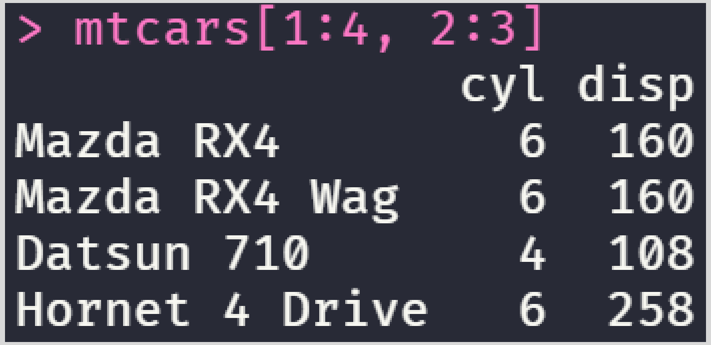
- Entire Column



```
> mtcars[, 8]
 [1] 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 0 1 0 0 0 1
```

Accessing Data

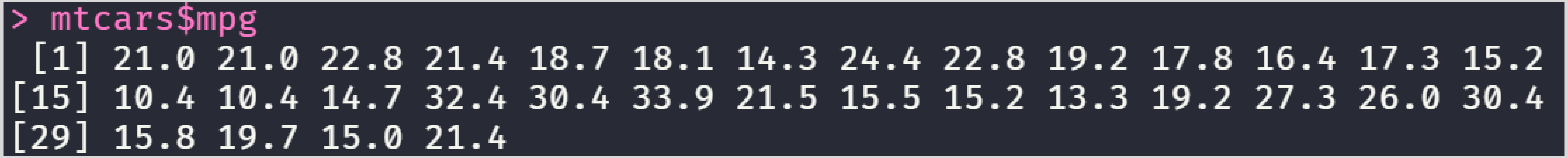
- Ranges of Data: use ‘:’



```
> mtcars[1:4, 2:3]
               cyl disp
Mazda RX4        6  160
Mazda RX4 Wag    6  160
Datsun 710       4  108
Hornet 4 Drive   6  258
```

Accessing Data

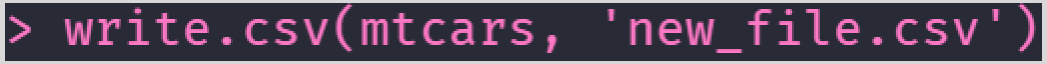
- Accessing via Column Name
- Dataframe[‘column_name’]
- Dataframe$column_name



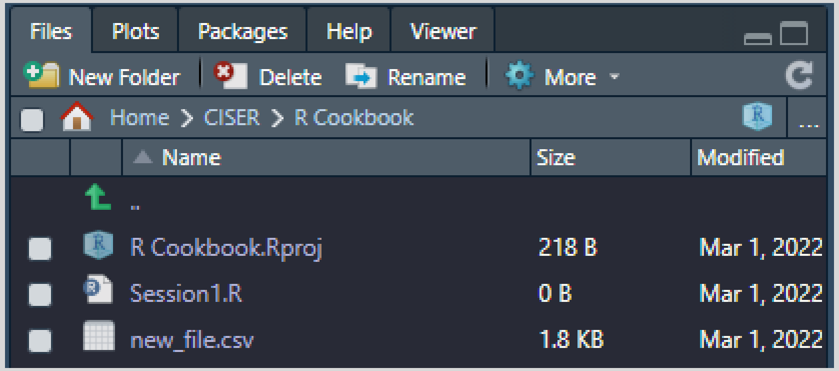
```
> mtcars$mpg
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
[15] 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4
[29] 15.8 19.7 15.0 21.4
```

Saving Data

- What if we want to reuse our data?
- Export it to a csv file!



```
> write.csv(mtcars, 'new_file.csv')
```



Name	Size	Modified
..		
R Cookbook.Rproj	218 B	Mar 1, 2022
Session1.R	0 B	Mar 1, 2022
new_file.csv	1.8 KB	Mar 1, 2022

Ending Notes

- Thanks for joining in – if you have any questions please ask!
- 3/10 – Tidy Data in R
- 3/24 – Data Visualization using ggplot2
- 3/31 – Reproducible Reports using RMarkdown