Launching R Studio and Creating a Project

JO 521

DATA JOURNALISM

Open R Studio

When you open **RStudio**, you'll see the **CONSOLE**

We'll be creating a **PROJECT** in **RStudio** to do our work

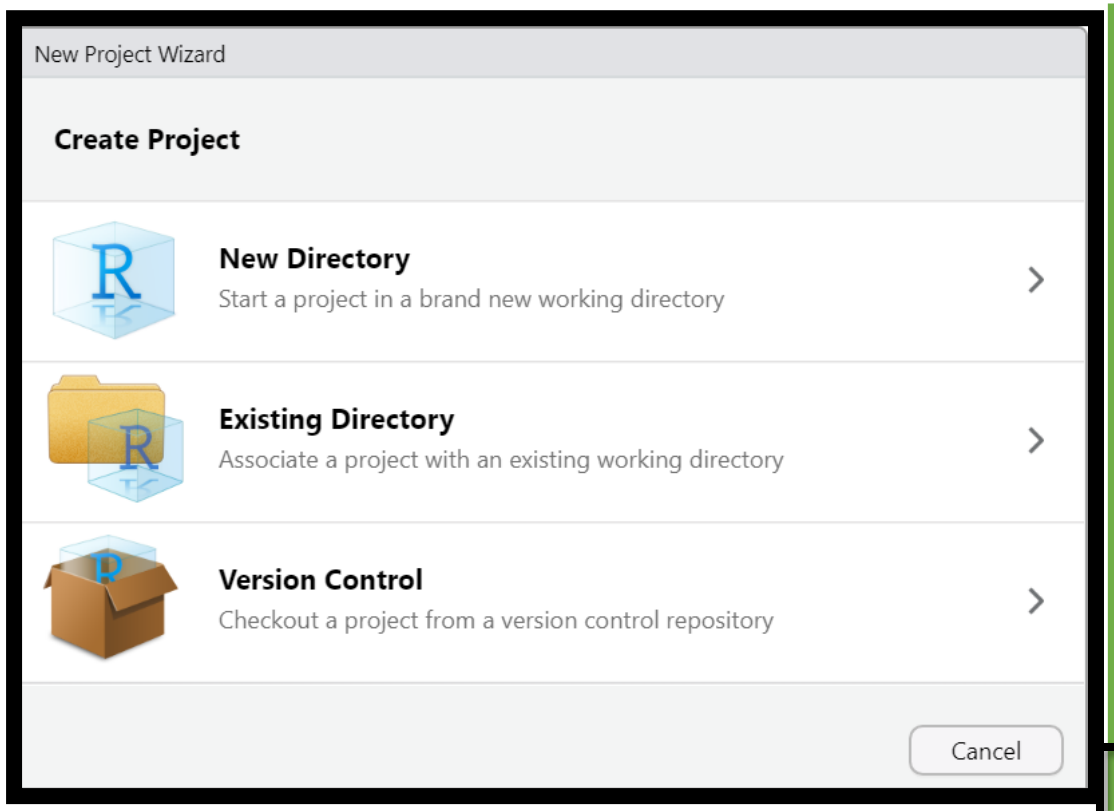
Go to **FILE**

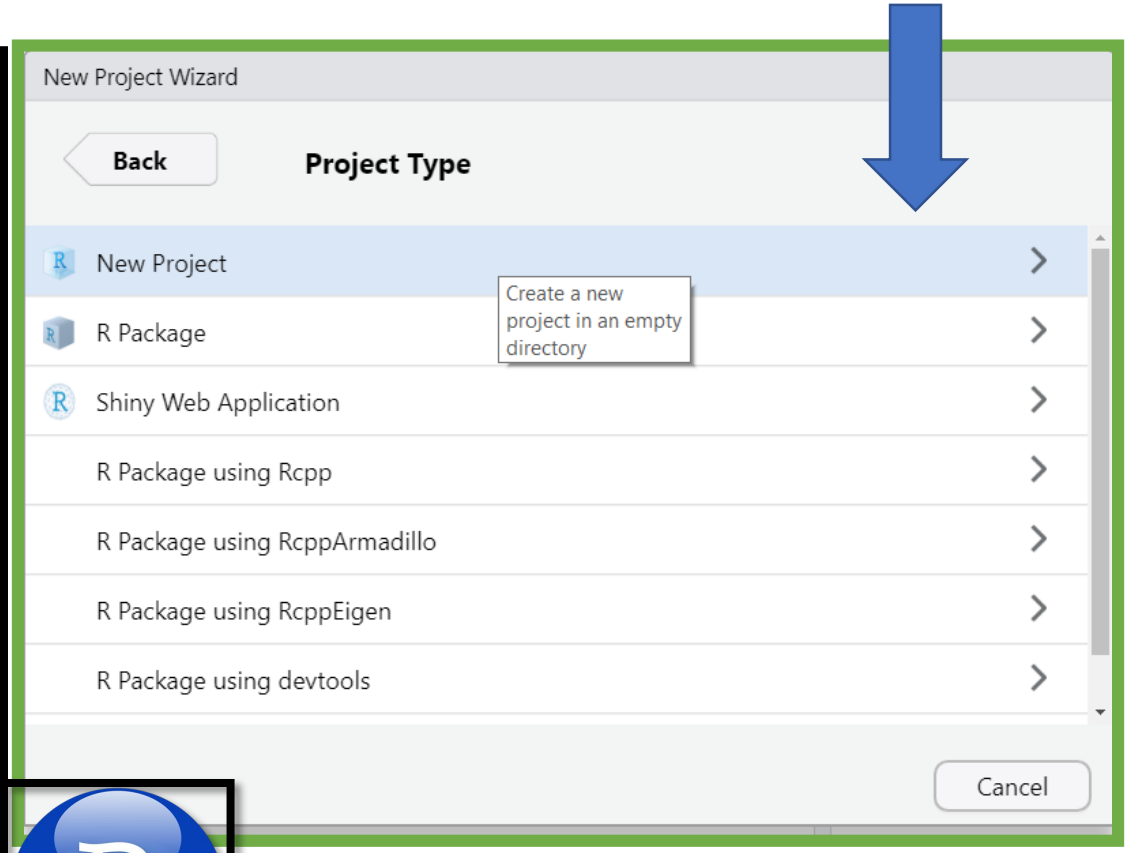
Choose **NEW PROJECT**

We'll choose **NEW DIRECTORY**

Find the files I sent to you and click **CREATE PROJECT**

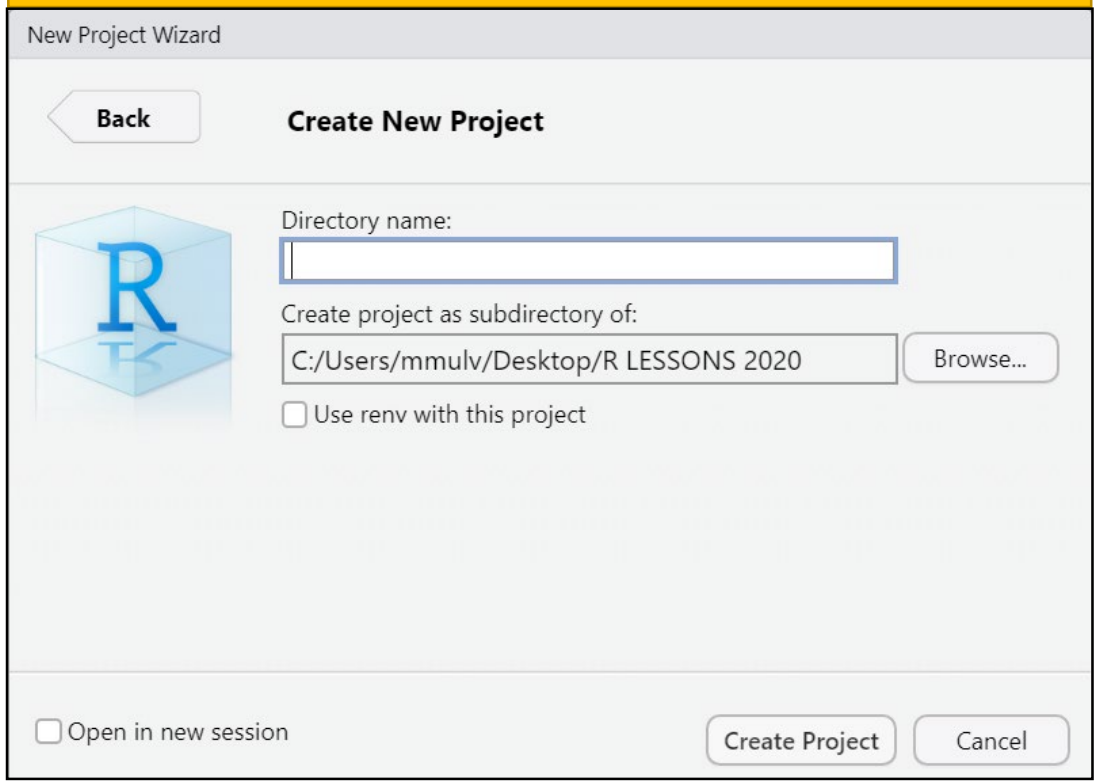
Choose **New Directory** And Then **New Project**





Name Your Project

Name Your Project



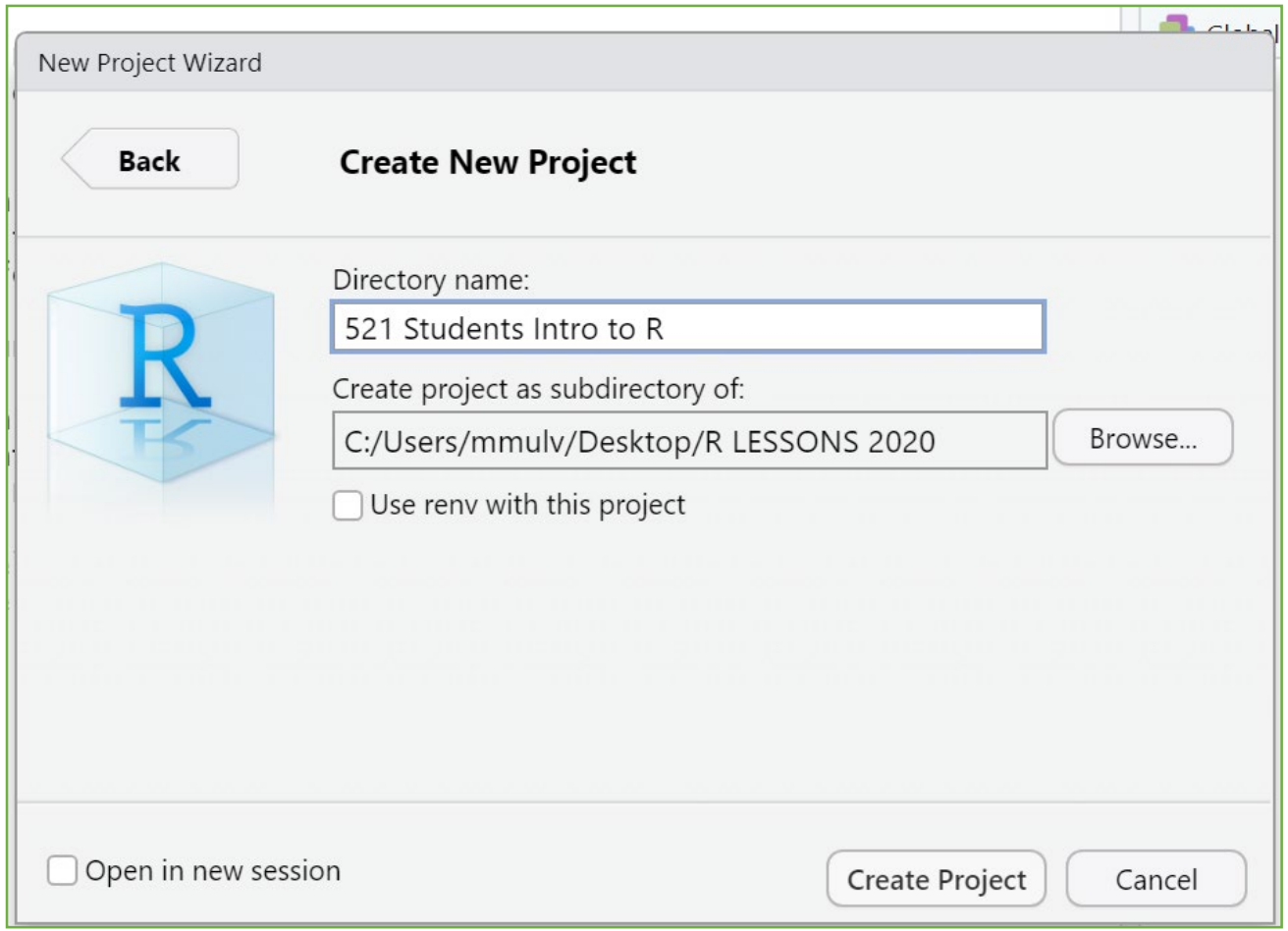
Choose Your Subdirectory

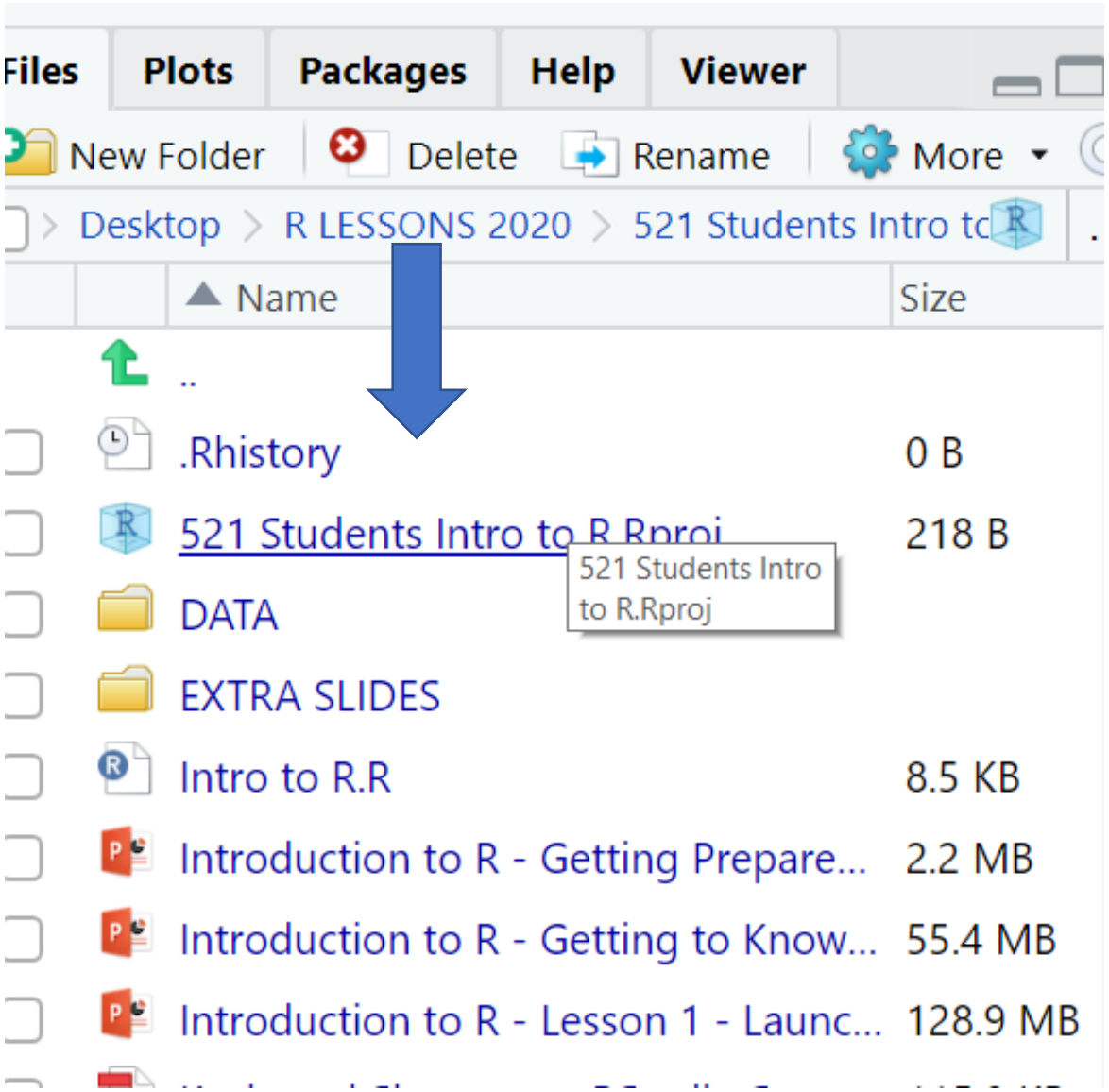
Mine is R Lessons 2020 which is on my desktop

Then click **CREATE PROJECT**

Name Your New Project

- Make sure the name matches the name of the folder where you put your files on your desktop
  - You are creating a **PROJECT**
- We will then bring **FILES** into the project including our script and our data




Files
Plots
Packages
Help
Viewer
New Folder
Delete
Rename
More
Desktop > R LESSONS 2020 > 521 Students Intro to
Name
Size
..
.Rhistory
0 B
521 Students Intro to R.Rproj
218 B
521 Students Intro to R.Rproj
DATA
EXTRA SLIDES
Intro to R.R
8.5 KB
Introduction to R - Getting Prepare...
2.2 MB
Introduction to R - Getting to Know...
55.4 MB
Introduction to R - Lesson 1 - Launc...
128.9 MB

Intro to R - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins | Intro to R

Console | Terminal | Jobs

C:/Users/mmulv/Desktop/R LESSONS 2020/Intro to R/

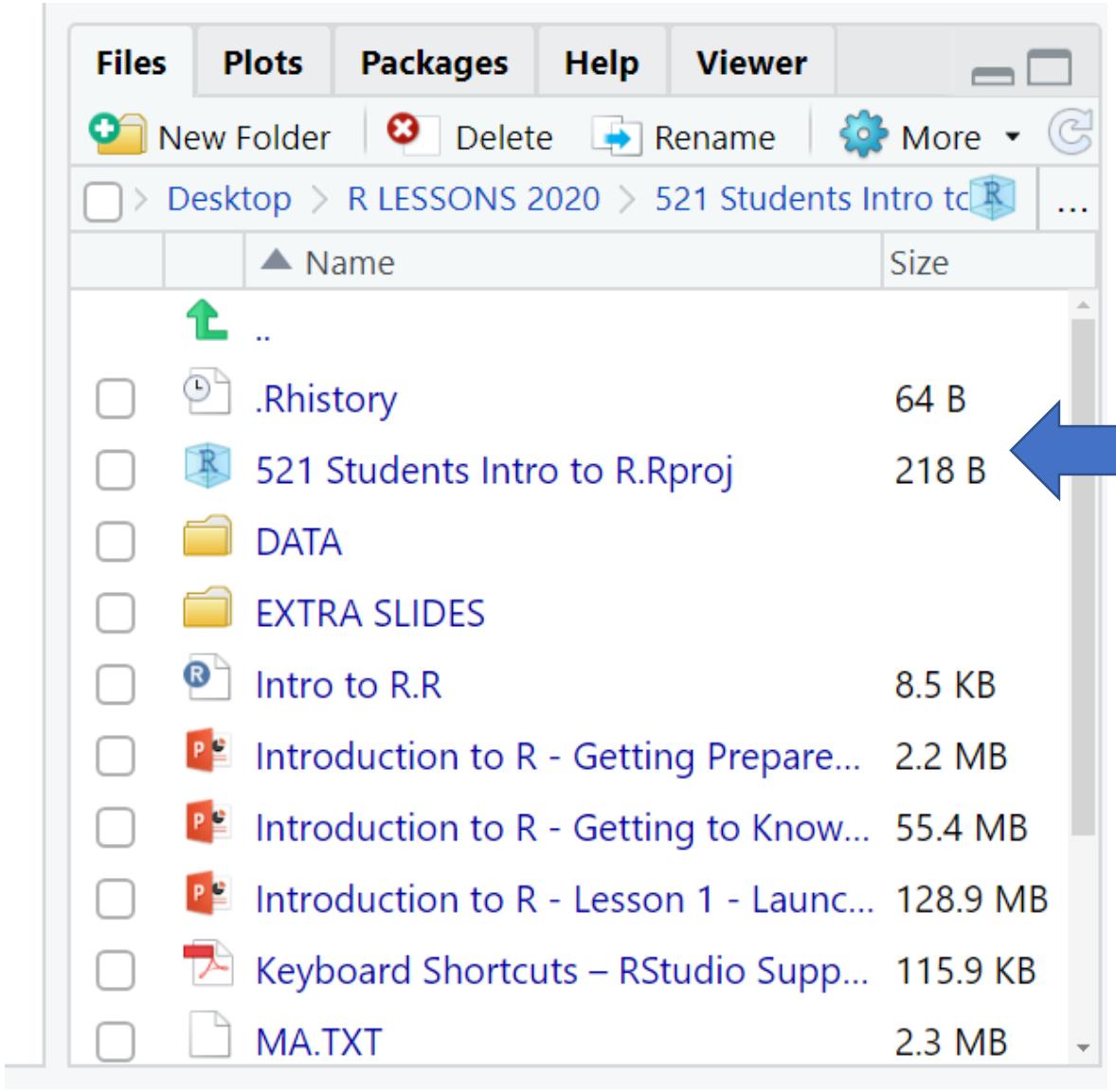
```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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.

>
```

Files | Plots | Packages | Help | Viewer

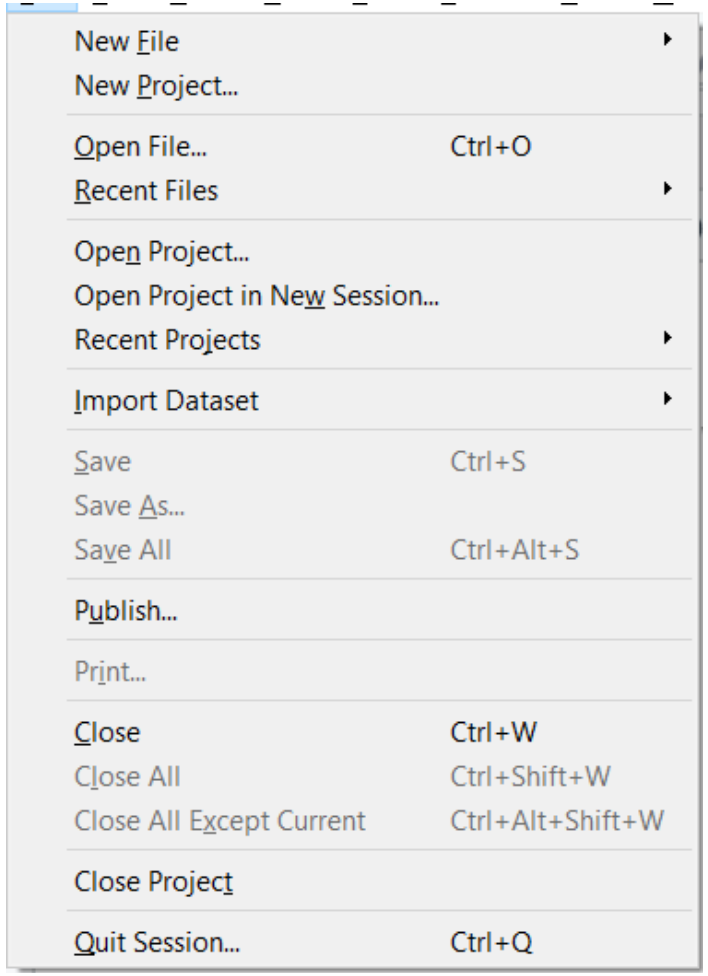
New Folder | Delete | Rename | More

esktop > R LESSONS 2020 > 521 Students Intro to R

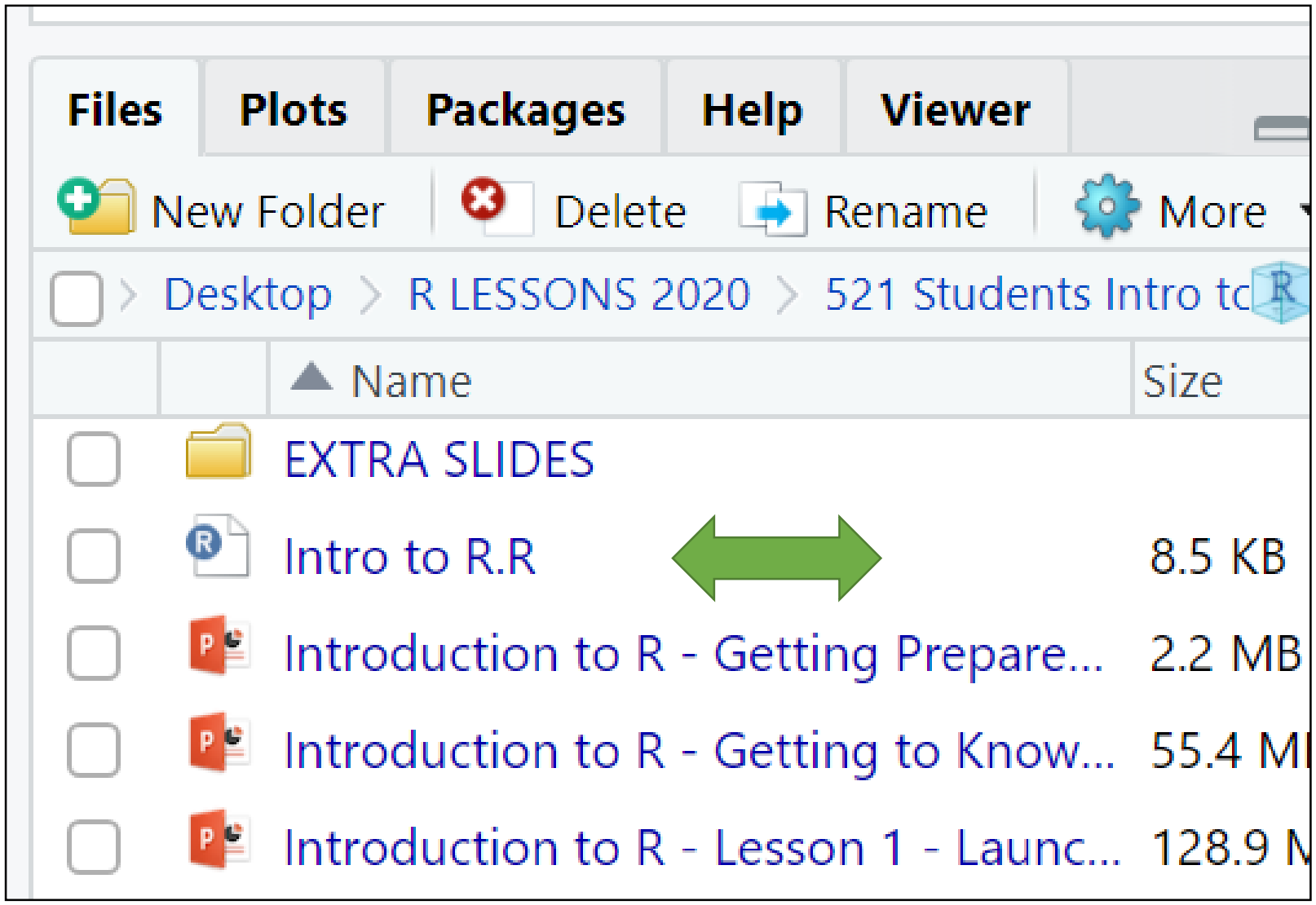
Name	Size
..	
.Rhistory	0 B
521 Students Intro to R.Rproj	218 B
DATA	
EXTRA SLIDES	
Intro to R.R	8.5 KB
Introduction to R - Getting Prepare...	2.2 MB
Introduction to R - Getting to Know...	55.4 MB
Introduction to R - Lesson 1 - Launc...	128.9 MB
Keyboard Shortcuts – RStudio Supp...	115.9 KB
MA.TXT	2.3 MB



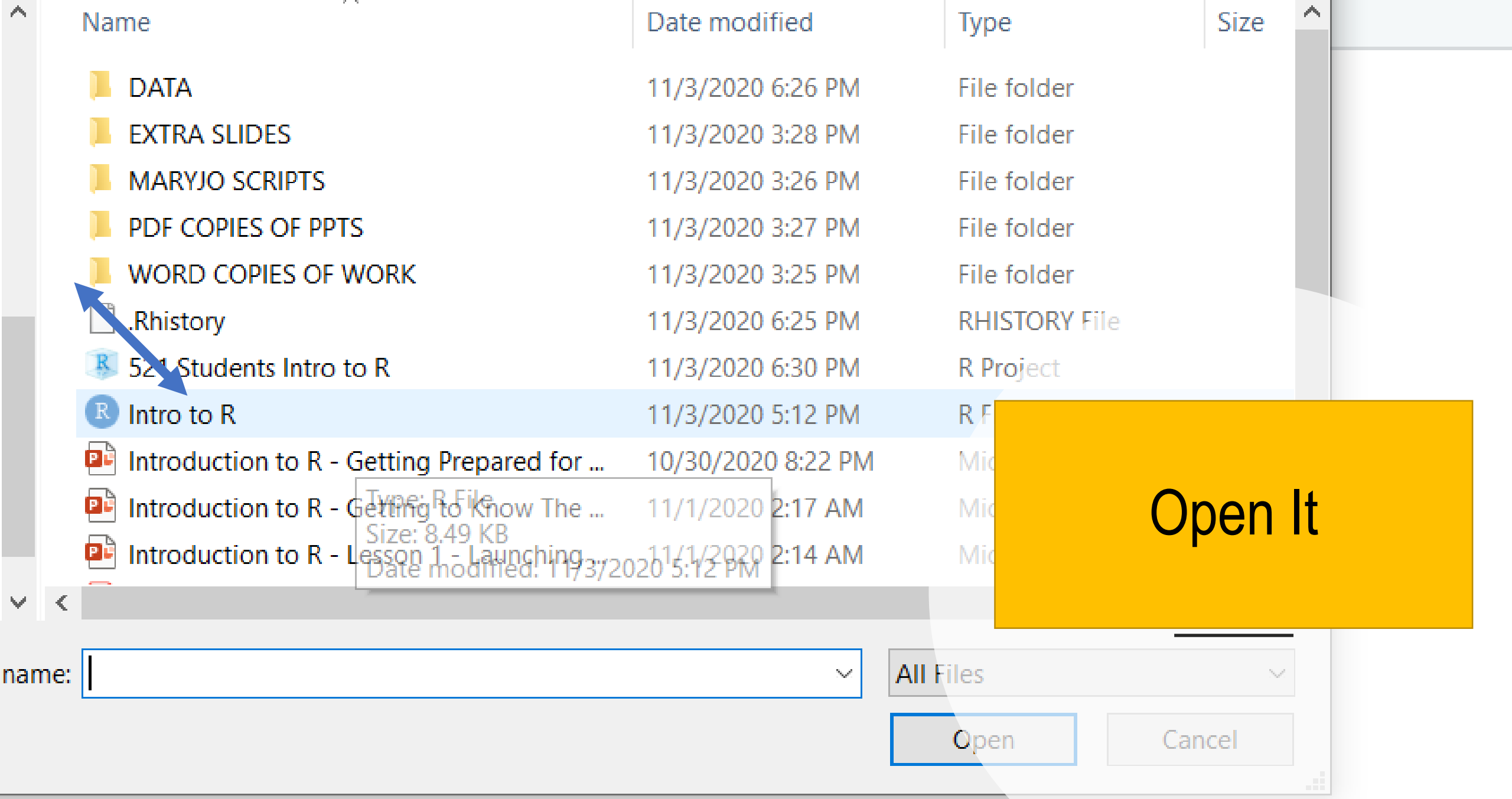
Your **NEW PROJECT** Will Show Up in Your **FILES** Window



Now In Rstudio, Go To **FILE**



Find The R Script File I Sent You: “Intro To R”


Name
Date modified
Type
Size
DATA 11/3/2020 6:26 PM File folder
EXTRA SLIDES 11/3/2020 3:28 PM File folder
MARYJO SCRIPTS 11/3/2020 3:26 PM File folder
PDF COPIES OF PPTS 11/3/2020 3:27 PM File folder
WORD COPIES OF WORK 11/3/2020 3:25 PM File folder
.Rhistory 11/3/2020 6:25 PM RHISTORY File
521 Students Intro to R 11/3/2020 6:30 PM R Project
Intro to R 11/3/2020 5:12 PM
Introduction to R - Getting Prepared for ... 10/30/2020 8:22 PM
Introduction to R - Getting to Know The ... 11/1/2020 2:17 AM
Introduction to R - Lesson 1 - Launching ... 11/1/2020 2:14 AM
Type: R File
Size: 8.49 KB
Date modified: 11/3/2020 5:12 PM
Open It
name:
All Files
Open
Cancel
()' for an HTML browser interface to help.

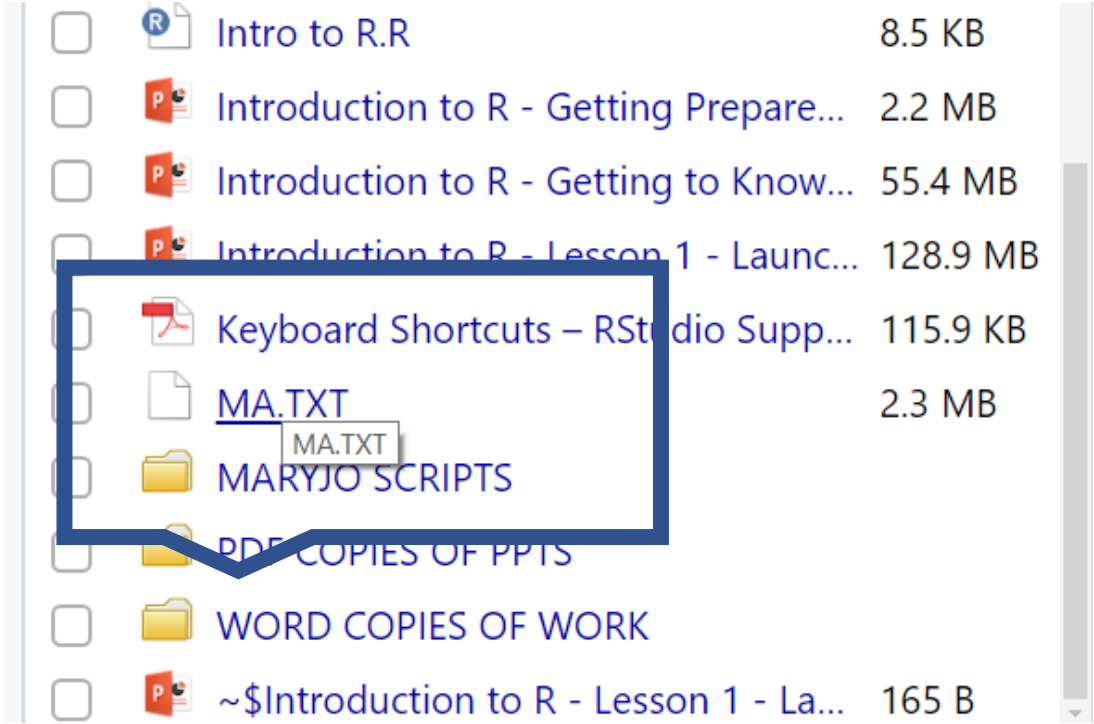
Intro to R.R

Source on Save | Run | Source

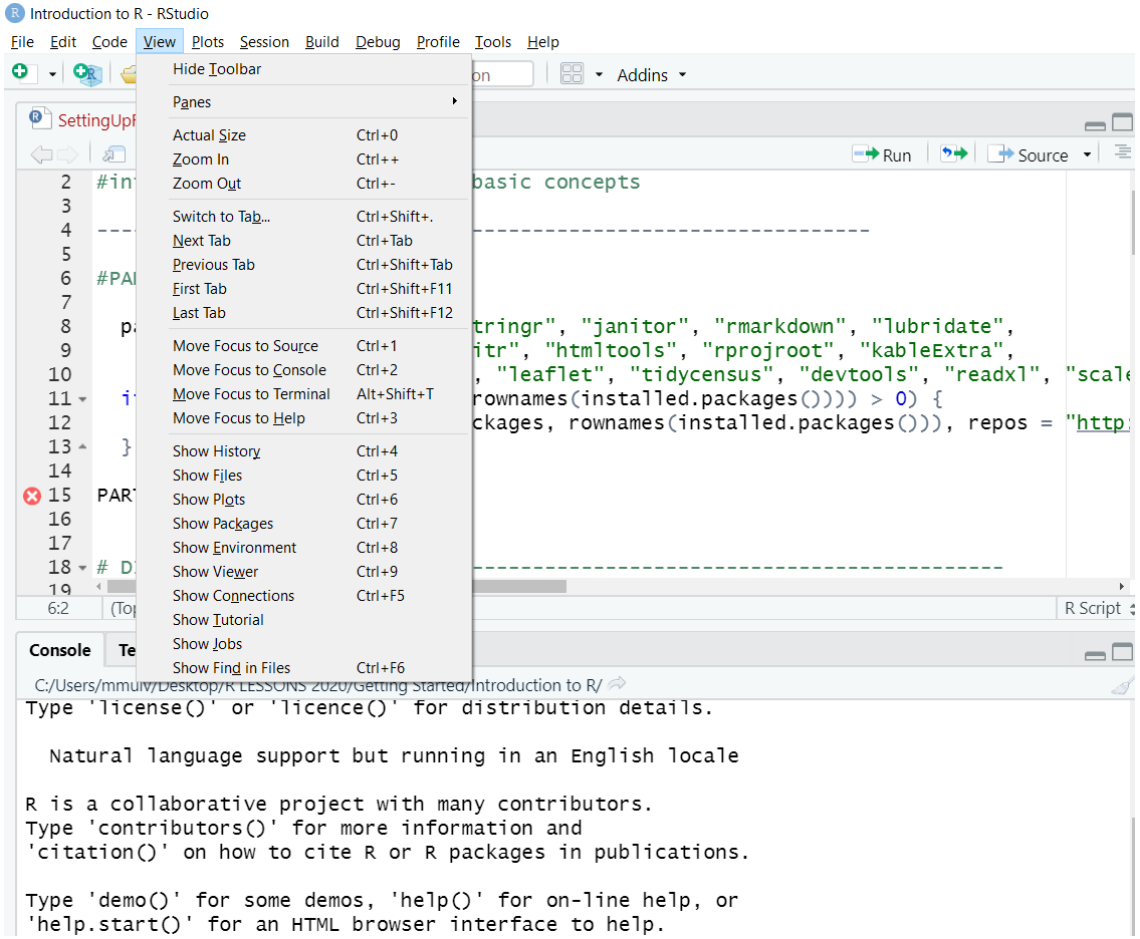
```
#Introduction to R: Basic Concepts and Scripts
#----------------

INSTRUCTIONS
#everything with a # in front of it is an instruction or annotation
#it is not code you run
#we are working in the SCRIPT window
#This script will allow us to run code and save it.
#But our results will not be displayed within the page
#They will be displayed in the CONSOLE area (below)
#Scripts are great for importing and cleaning data;
#less sufficient for analysis and making charts
#if you make an error, just hit CTRL or CMND Z and you'll go
#back to where you were
#-------------------------
```

Let's Check On Our .TXT File

We'll be "importing" or "reading it in" soon



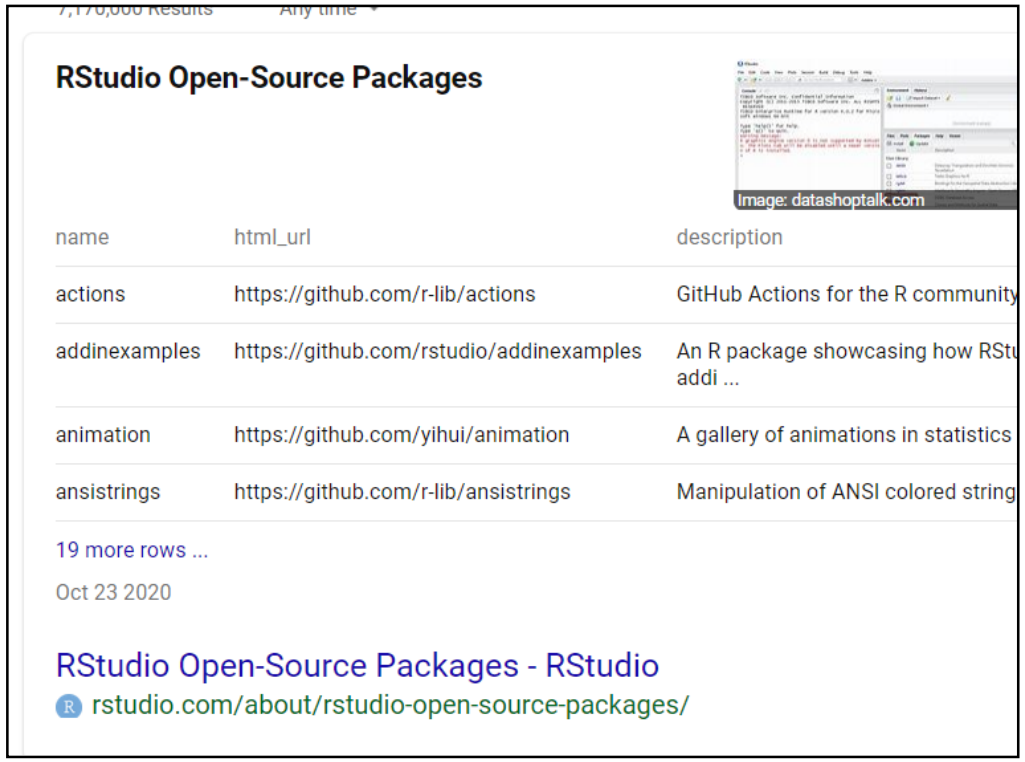
Changing the View in RStudio



Installing Packages

#You only need to install packages once

#Only time you would have to reinstall is if you got a new computer or to update a package



RStudio Open-Source Packages

Image: datashoptalk.com

name	html_url	description
actions	https://github.com/r-lib/actions	GitHub Actions for the R community
addinexamples	https://github.com/rstudio/addinexamples	An R package showcasing how RSt addi ...
animation	https://github.com/yihui/animation	A gallery of animations in statistics
ansistrings	https://github.com/r-lib/ansistrings	Manipulation of ANSI colored string

19 more rows ...

Oct 23 2020

RStudio Open-Source Packages - RStudio

rstudio.com/about/rstudio-open-source-packages/

Installing Packages

Packages are also known as libraries

Free code that can enhance **base R** and do specific tasks that make your data analysis more powerful

Sources of packages:

- CRAN – packages have to be vetted by **R** before they can be shared with users
- Can installed from the **RStudio** command console
- Write install.packages("NameOfPackage")
- Also available on **GitHub**, a platform where developers and journalists share code

Installing Packages

When you **RUN** code, you'll see the green line on the right-hand side

That tells you the code is running

You should see a **STOP** sign; that shows up when it is running

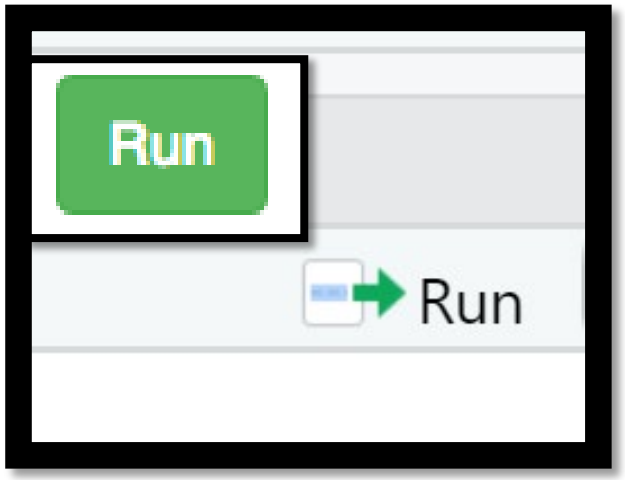
Installing Packages

#another option is to click **CTRL** or **CMND Enter**

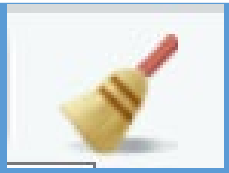
#that will run the lines of code you selected

#We are in a **SCRIPT**, so selecting the line of code and hitting the **RUN** button works

#You see the code is running below in the **CONSOLE**



Installing Packages



The **BROOM** clears out the **CONSOLE**

The **CONSOLE** will clear out whenever you close **RStudio**

You can **RUN** code in the **SCRIPT** or the **CONSOLE**

>

If you see a **caret** > at the bottom of your **CONSOLE**, you know the packages have installed

Another Way To Check Your Packages Are Installed

```
INSTRUCTIONS
#everything with a # in front of it is an instruction or annotation
#it is not code you run
#we are working in the SCRIPT window
#This script will allow us to run code and save it.
#But our results will not be displayed within the page
#They will be displayed in the CONSOLE area (below)
#Scripts are great for importing and cleaning data;
#less sufficient for analysis and making charts
#if you make an error, just hit CTRL or CMND Z and you'll go
#back to where you were
#-------------------------
```

1:1 (Top Level) R Script

Console Terminal Jobs

C:/Users/mmulv/Desktop/R LESSONS 2020/521 Students Intro to R/

```
  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.

>
```

Environment is empty

Files Plots Packages Help Viewer

Install Update

	Name	Description	Ve...
User Library			
☐	askpass	Safe Password Entry for R, Git, and SSH	1.1
☐	assertt...	Easy Pre and Post Assertions	0.2.1
☐	backpo...	Reimplementations of Functions Introduced Since R-3.0.0	1.1.10
☐	base64...	Tools for base64 encoding	0.1-3
☐	BH	Boost C++ Header Files	1.72.0-3
☐	blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.1

```
51
52
58:2 (Top Level) R Script
Console Terminal Jobs
C:/Users/mmulv/Desktop/R LESSONS 2020/Introduction to R/
package 'ggthemes' successfully unpacked and MD5 sums checked
package 'kableExtra' successfully unpacked and MD5 sums checked
package 'sf' successfully unpacked and MD5 sums checked
package 'mapview' successfully unpacked and MD5 sums checked
package 'leaflet' successfully unpacked and MD5 sums checked
package 'tidycensus' successfully unpacked and MD5 sums checked
package 'devtools' successfully unpacked and MD5 sums checked
package 'formattable' successfully unpacked and MD5 sums checked
package 'DT' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\mmulv\AppData\Local\Temp\RtmpmwaTIr\downloaded_packages
>
```

Minimizing Windows

Script vs. Markdown

R Script

- You write your code in the **SCRIPT** window
- Results appear in the **CONSOLE** below
- Works well with data that requires a lot of cleanup on the import

R Markdown

- Let's you see your code and results right in the **SCRIPT** window
- Let's you save your results as an .HTLM page
- You can then share that page with others

Installing v. Loading Libraries

#You install packages once in **RStudio**

#But every time you use **RStudio**, you need to load them

#It is like turning on a light switch

#RUN to load your package

$Your package name doesn't need to be in quotes

- Library(tidyverse)

#If you highlight the package and run the code, you'll see it has attached the packages to **RStudio**

```
#once you've installed the package, put a # in front of yo

library(tidyverse)
```

11:19 (Top Level)

Console Terminal Jobs

C:/Users/mmulv/Desktop/R LESSONS 2020/Getting Started/Introduction to R/

```
Content type 'application/zip' length 440022 bytes (429 KB)
downloaded 429 KB

package 'tidyverse' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\mmulv\AppData\Local\Temp\RtmpKC3CD5\downloaded_
> library(tidyverse)
-- Attaching packages ---------------------------------------
v ggplot2 3.3.2     v purrr   0.3.4
v tibble  3.0.4     v dplyr   1.0.2
v tidyr   1.1.2     v stringr 1.4.0
v readr   1.4.0     v forcats 0.5.0
-- Conflicts ------------------------------------------------
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
>
```

Loading Libraries

library(tidyverse)

#Now that the packages have installed, we'll start the code we need for this particular project

#You first need to load any of the packages (also referred to as libraries) that you will use for this script

#You do this every time you write R code (in scripts, RMarkdown, Notebooks, etc)

#for this script we're only going to use a compilation of packages called **tidyverse**

#to run code, highlight the line of code below and go to the RUN button above

#choose "run selected lines“

library(tidyverse)

Loading Libraries

library(tidyverse)

#Now that the packages have installed, we'll start the code we need for this particular project

#You first need to load any of the packages (also referred to as libraries) that you will use for this script

#You do this every time you write R code (in scripts, RMarkdown, Notebooks, etc)

#for this script we're only going to use a compilation of packages called **tidyverse**

#to run code, highlight the line of code below and go to the RUN button above

library(tidyverse)

Loading Libraries

Console Terminal Jobs

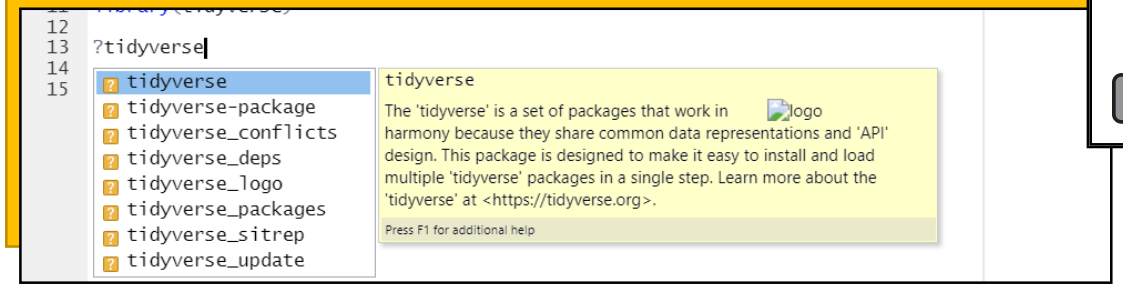
C:/Users/mmulv/Desktop/R LESSONS 2020/Introduction to R/

```
The downloaded binary packages are in
        C:\Users\mmulv\AppData\Local\Temp\RtmpmwaTIr\downloaded_
> library(tidyverse)
-- Attaching packages ---------------------------------------
v ggplot2 3.3.2     v purrr   0.3.4
v tibble  3.0.4     v dplyr   1.0.2
v tidyr   1.1.2     v stringr 1.4.0
v readr   1.4.0     v forcats 0.5.0
-- Conflicts ------------------------------------------ ti
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
> library(tidyverse)
>
```

Learn More About **tidyverse**

Getting Help With Packages

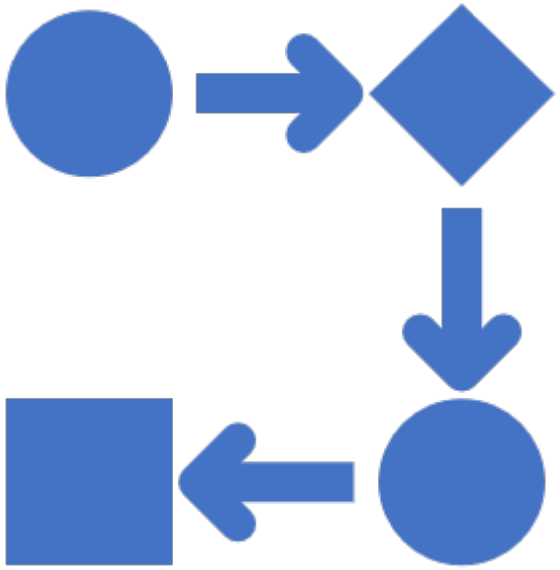
- On a new line, you can get more information about **tidyverse**
  - Type ?tidyverse
  - **RUN** the code



Working Directory

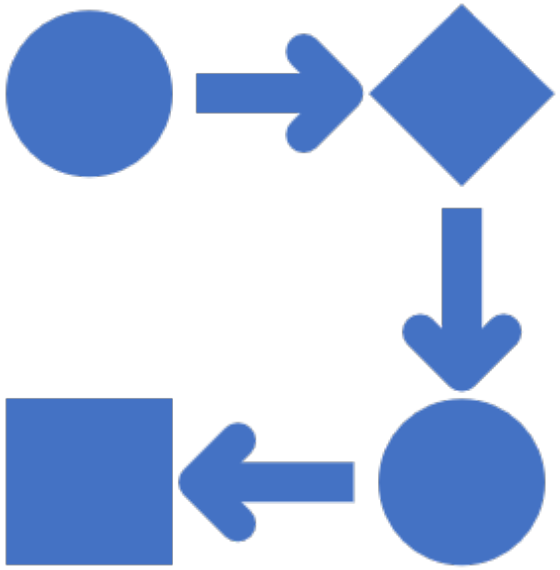
- **Getwd**
- #This tells you what directory you are working in
- #This is a base **R** command
- #Sometimes you need to find a file; this can tell you where it is
- #you can set a permanent default directory under **Tools/Global Options**
- #you need to close and reopen **R** to make it work
- #in the future when you make a new project, it will guess that you
- #want to save it in that directory

Understanding How to Import Data Files



- Every time we want to work with data in **R**, we have to bring it again when we start a new session of **RStudio**
- We will work with the baby names data we just used in **SQL**
- we are going to create a **Data frame**
- **Data frame** is the data you are going to work with

Understanding How to Import Data Files



- This is a **base R** label
  - Think of a spreadsheet in **EXCEL**
  - To bring a spreadsheet into **R**, you “read” it into the program
  - It will come in as a data frame
  - **RStudio** calls these a **“tibble”**
  - Its just rows and columns

Understanding How to Import Data Files

- This is a **base R** label
  - Think of a spreadsheet in **EXCEL**
  - To bring a spreadsheet into **R**, you “read” it into the program
  - It will come in as a data frame
  - R Studio calls these a **“tibble”**
  - Its just rows and columns

Assignment Operators

#we are going to use the **readr()** package

#and the **read_csv function**

#to bring in our **.csv** or **txt** file

#type:

rawdata <- read_csv('MA.TXT)

<- - assignment operator:

- This assigns data to a data frame
  - Shortcut on Windows - Alt +
  - Shortcut on MAC - Option+-

Reading In Data

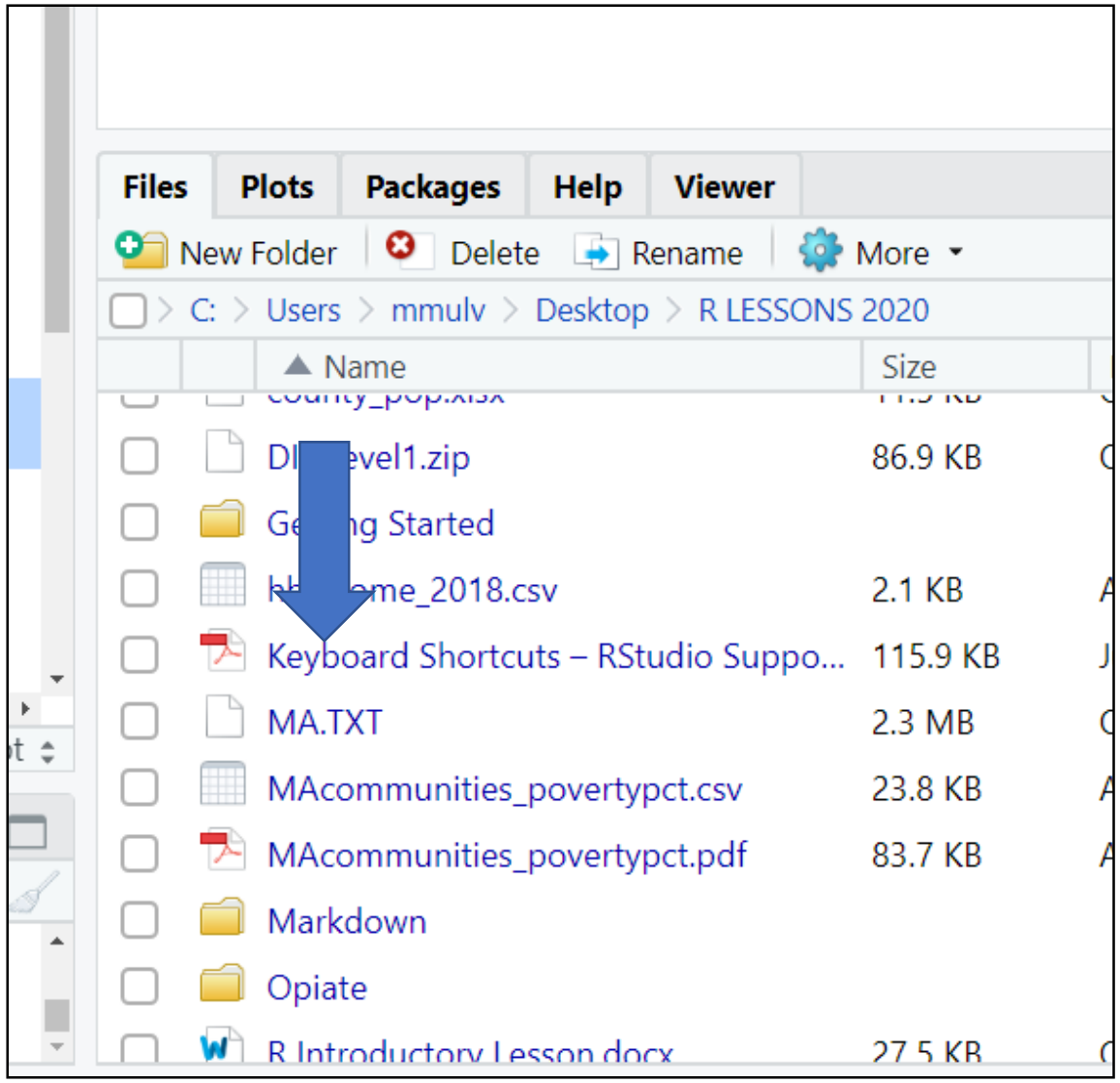
#If you go to **FILE,** you will see the **.csv** or **.txt** file you are going to bring in

MA.TXT

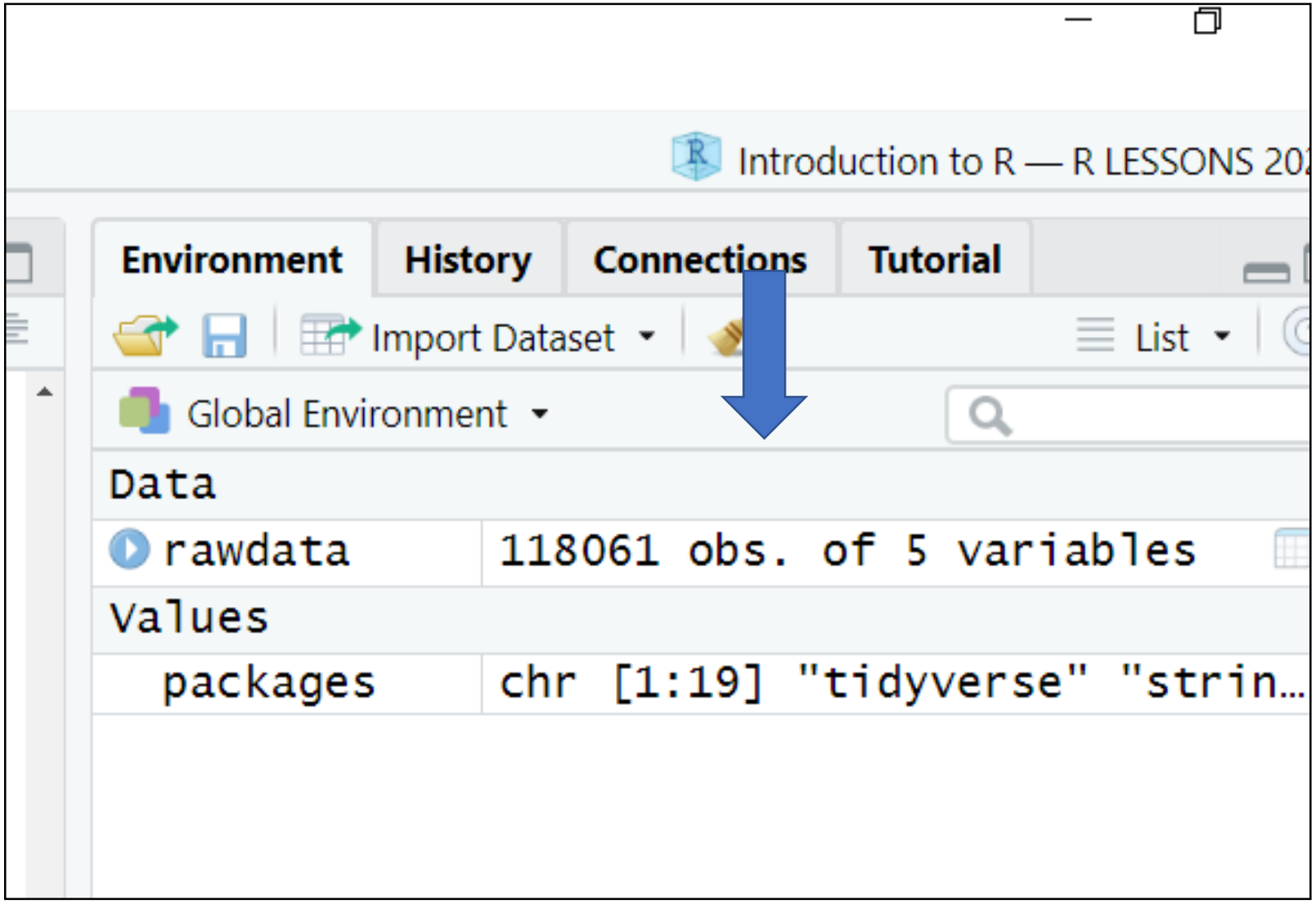
#type rawdata <- read_csv('MA.TXT')

#you need to type the file name exactly as it appears in your **FILES**

#don't forget the quote marks



Importing Data



Look In Your Environment On Upper Right

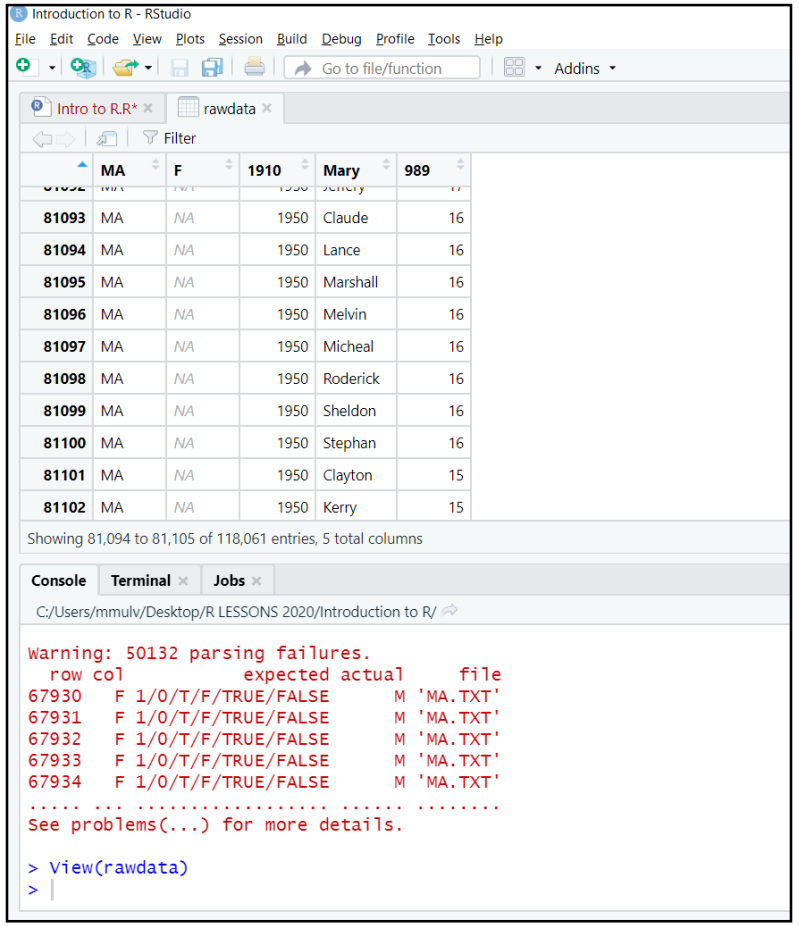
Click On **rawdata** – A Table Opens!

```
140:31   (Top Level)
Console   Terminal   Jobs
C:/Users/mmulv/Desktop/R LESSONS 2020/Introduction to R/
)

Warning: 50132 parsing failures.
  row col           expected actual     file
67930   F 1/0/T/F/TRUE/FALSE      M 'MA.TXT'
67931   F 1/0/T/F/TRUE/FALSE      M 'MA.TXT'
67932   F 1/0/T/F/TRUE/FALSE      M 'MA.TXT'
67933   F 1/0/T/F/TRUE/FALSE      M 'MA.TXT'
67934   F 1/0/T/F/TRUE/FALSE      M 'MA.TXT'
..... ... ................... ...... ........
See problems(...) for more details.
```

Our Code Ran But . . . Problems?

Problems

- In the **CONSOLE**
- We have parsing failures
- We see TRUE and FALSE
- Look at your table
- Does anything look odd
- Remember your record layout
- Scroll down
- What do you see?



Problems to Fix

- #If we were to open the text file, we'd see that the gender column had
- #either an M or an F
- #the girls were at the top, so F was the first thing **R** found
- #R guesses the format of each column
- #it looks at the first 2,000 rows
- #should it be text or a date or a number?
- #in this case, it guesses that F is FALSE
- #it didn't import it properly

Problems to Fix

- When it hit the M letters, it didn’t know what to do
- It switched all the “F” (female) values to “FALSE”
- It eliminated all the “M” values completely
- We need to tell it to store that value as character

Filter

	MA	F	1910	Mary	989
1	MA	FALSE	1910	Helen	473
2	MA	FALSE	1910	Margaret	374
3	MA	FALSE	1910	Dorothy	331
4	MA	FALSE	1910	Alice	313
5	MA	FALSE	1910	Anna	252
6	MA	FALSE	1910	Ruth	247
7	MA	FALSE	1910	Elizabeth	224
8	MA	FALSE	1910	Mildred	198
9	MA	FALSE	1910	Lillian	196
10	MA	FALSE	1910	Rose	187

Showing 1 to 11 of 118,061 entries, 5 total columns

- It also doesn't have a header
- It took the first row of data and made it a header
- Our data file doesn't have a header row in it

#we are going to change the import code to add a few arguments

#read_csv has multiple arguments you can give it for the import

#the only one it really needs is the name of the file

#we will add two more arguments to fix our problems

#all our arguments will be separated by commas

Adjusting Our Code

#our second argument, after our file name, is to tell **R** our data doesn't have any column names

#so we will add an argument that says ***col_names = FALSE***

#R will put its own column names in as x1, x2, x3, x4

Adding Arguments To Our Code

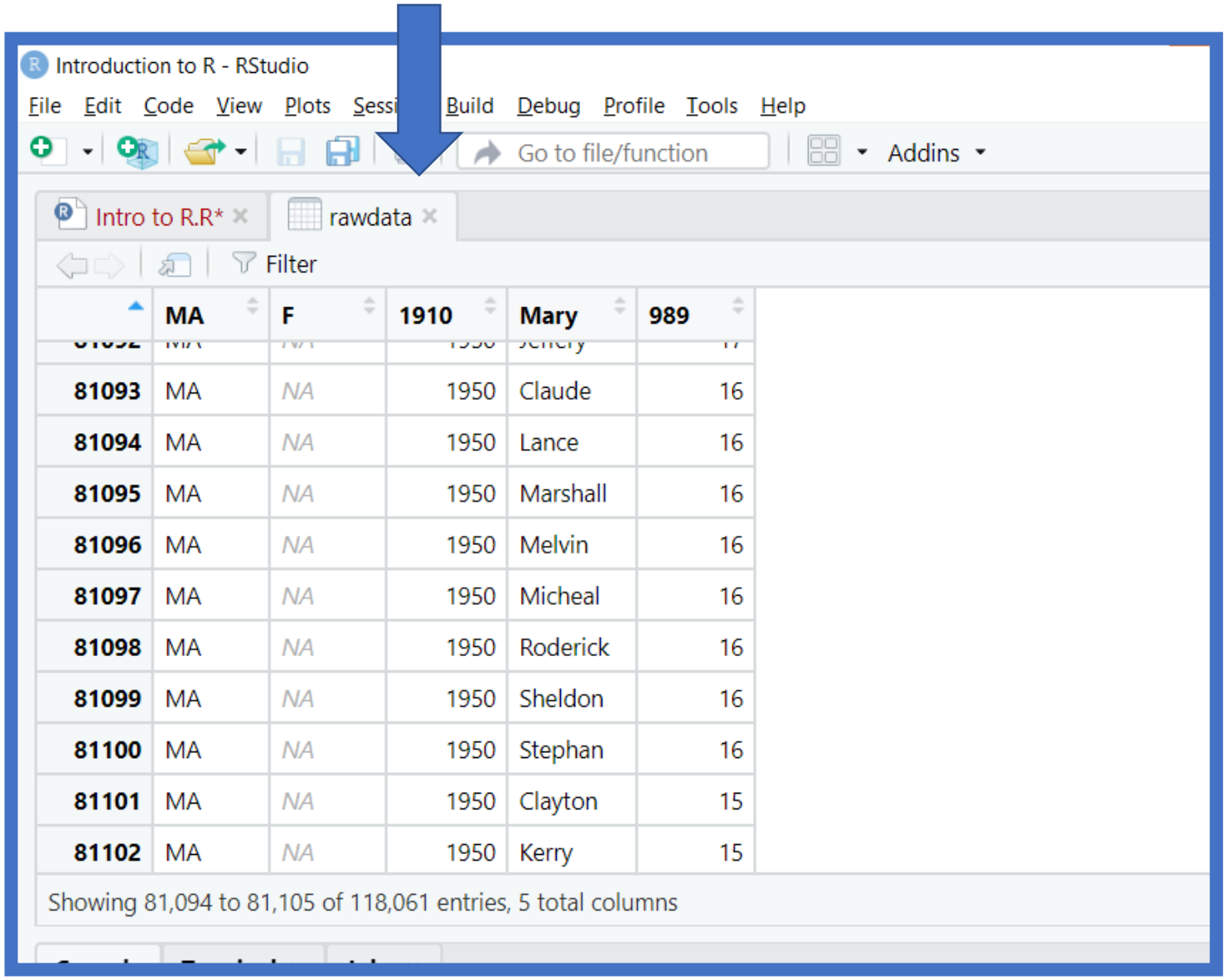
- We'll also add a third argument to our code
- We are going to tell **R** that our renamed columns need to have characters in them
- The third argument will be:
- ***col_types = (X2=col_character()))***
- Remember our gender column was the second column
- With our second argument, it was renamed as X2
- We need to tell **R** to make X2 a character

Our New Code

- Our new code will be:
- ***rawdata <- read_csv('MA.TXT', col_names=FALSE, col_types=cols(X2=col_character()))***
- Because we are assigning this to rawdata, it will overwrite our previous table
- Run this new code

Getting Out Of a Window

- You can click the small x to get out of a window
- Click that to close



Fixing Errors

Let’s Look At Our New Data Frame

- We can look at the structure
- This will show us format of each column and a sample of the data in each column
- ***str(rawdata)***
- It is like a record layout
- It let’s us preview our new data
- Run that code

Now Look At Our Data

Console Terminal Jobs

C:/Users/mmulv/Desktop/R LESSONS 2020/Introduction to R/

```
 $ X2: chr [1:118062] "F" "F" "F" "F" ...
 $ X3: num [1:118062] 1910 1910 1910 1910 1910 1910 1910 1910 1910 1910 ...
 $ X4: chr [1:118062] "Mary" "Helen" "Margaret" "Dorothy" ...
 $ X5: num [1:118062] 989 473 374 331 313 252 247 224 198 196 ...
 - attr(*, "spec")=
  .. cols(
  ..   X1 = col_character(),
  ..   X2 = col_character(),
  ..   X3 = col_double(),
  ..   X4 = col_character(),
  ..   X5 = col_double()
  .. )
>
```

This is like setting data types in EXCEL where you
Can set something as a date or a number or text
It has set column 3 – X3 – to double characters because it is a year
The same with column 5 – X5 – the number of babies

Checking Your Adjustments

- If you scroll up in your CONSOLE you can see some examples of the new data that is fixed
- R is telling you the first four records
- A glimpse of your data

```
awdata)
[118,062 x 5] (S3: spec_tbl_
chr [1:118062] "MA" "MA" "MA
chr [1:118062] "F" "F" "F" "
num [1:118062] 1910 1910 191
chr [1:118062] "Mary" "Helen
num [1:118062] 989 473 374 3
(*, "spec")=
ls(
X1 = col_character(),
X2 = col_character(),
X3 = col_double(),
X4 = col_character(),
```

Renaming Column Names

#this file comes out every year without headers

#so having this script will let you fix it each time

#let's rename our column names

#only remaining problem is that we don't want X1, X2, etc as our column names

#we'll use the **rename()** function to fix that

#but which package is this function in?

#here's how we can find out which package the **rename()** function comes from

?rename()

Introduction to R - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Intro to R.R*

```r
str(rawdata)


#We have one more thing we need to fix
#we don't want X1, X2, X3 as our column names
#Let's rename them
#We'll use the rename() function to fix that
#but which package is this function in?
#Here's how we can find out which package the rename() function comes from
?rename()
```

210:10 (Top Level) R Script

Console Terminal Jobs

C:/Users/mmulv/Desktop/R LESSONS 2020/Introduction to R/

```
 $ X3: num [1:118062] 1910 1910 1910 1910 1910 1910 1910 1910 1910 1910 ...
 $ X4: chr [1:118062] "Mary" "Helen" "Margaret" "Dorothy" ...
 $ X5: num [1:118062] 989 473 374 331 313 252 247 224 198 196 ...
 - attr(*, "spec")=
  .. cols(
  ..   X1 = col_character(),
  ..   X2 = col_character(),
  ..   X3 = col_double(),
  ..   X4 = col_character(),
  ..   X5 = col_double()
  .. )
> ?rename()
>
```

Environment History Connections Tutorial

Global Environment

Data	
rawdata	118062 obs. of 5 variables
Values	
packages	chr [1:19] "tidyverse" "strin…

Files Plots Packages Help Viewer

R: Rename columns

rename {dplyr} R Documentation

Rename columns

Description

`rename()` changes the names of individual variables using `new_name = old_name` syntax; `rename_with()` renames columns using a function.

Usage

```
rename(.data, ...)

rename_with(.data, .fn, .cols = everything(),
```

Arguments

The **Rename** Function

#We'll add **rename** function on the end of our import process

#We want to **rename** our column names from X1, X2, X3 to babynames

#We are going to add this argument to the end of our import

#we'll name our dataframe "babynames" by using the assignment operator (<-)

#keyboard shortcut on Windows (Alt+-)
#Keyboard shortcut on Mac (Option+-)

Our New Code

```
babynames <-  read_csv("MA.TXT", col_names=FALSE,
                    col_types=cols(X2="c"))%>%
rename(state=X1, gender=X2, yr= X3, name=X4, num_births=X5)
```

Piping Operations

#we are “piping” these two operations together

#they are separate operations but connect them with a **pipe %>%**

#now we can remove the old data frame – **rawdata** – from our **ENVIRONMENT**

Removing Data Frames

The Rename Function

- we've made some small adjustments to our code
- We are giving our data frame a new name - ***babynames***
- We'll get rid of our old data frame once this is imported
- We've also changed ***X2=col_character to "c"*** – which means character
- It is just another way of saying character
- R has different ways to do the same thing
- We've also added a "**pipe**" onto the end of that string - **%>%**

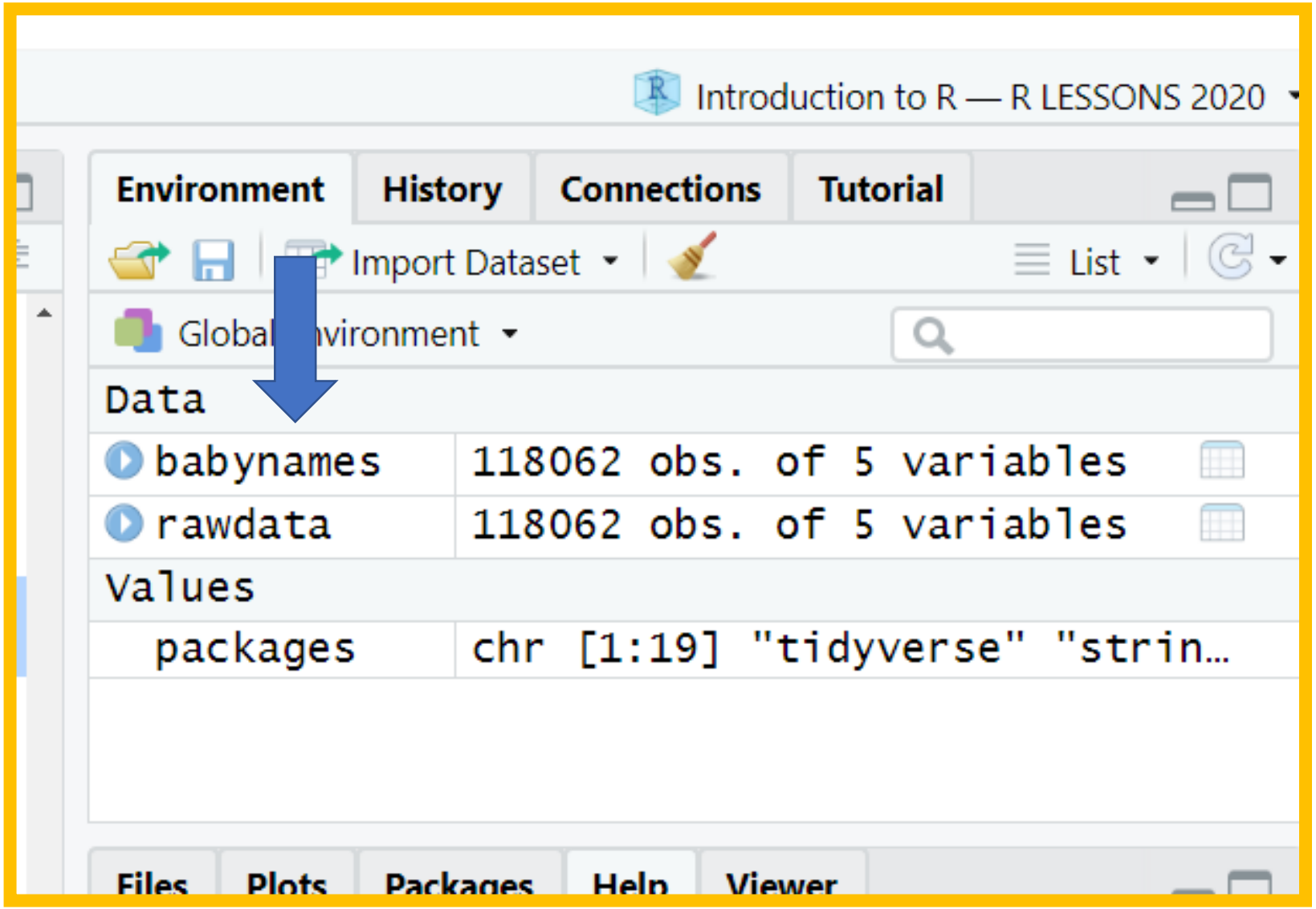
Using A Pipe

- A pipe is a way to string syntax in **R** together
  - The syntax is **%>%**
- You are saying import this data and these columns and **THEN** rename these columns with these new names, i.e. state=X1, gender=X2

Your New Code

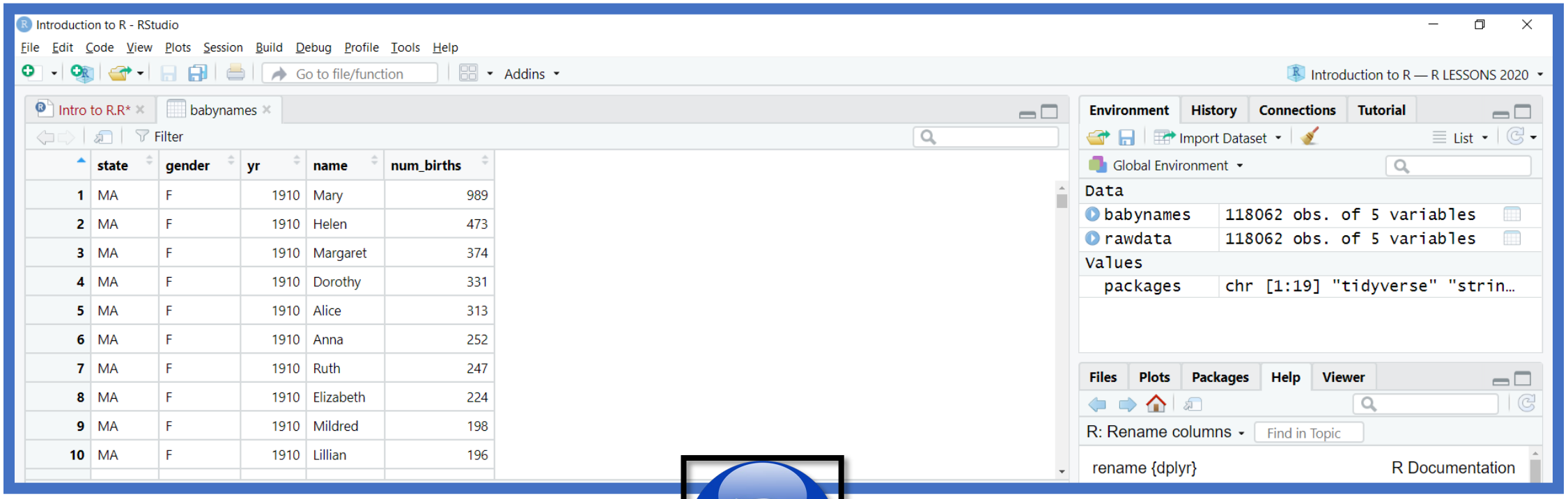
```
babynames <-  read_csv("MA.TXT", col_names=FALSE,
                  col_types=cols(X2="c"))%>%
      rename(state=X1, gender=X2, yr= X3, name=X4, num_births=X5)

#RUN that code
```

The Rename Function



Now You See Your New Data Frame In Your Environment

Click On Your New Data Frame



	state	gender	yr	name	num_births
1	MA	F	1910	Mary	989
2	MA	F	1910	Helen	473
3	MA	F	1910	Margaret	374
4	MA	F	1910	Dorothy	331
5	MA	F	1910	Alice	313
6	MA	F	1910	Anna	252
7	MA	F	1910	Ruth	247
8	MA	F	1910	Elizabeth	224
9	MA	F	1910	Mildred	198
10	MA	F	1910	Lillian	196

Removing A Data Frame

- #let's remove that old data frame from our environment
- #we don't need it
- #the function is ***rm(NameOfDataFrame)***
- #can you do it?

rm(rawdata)

Saving and Closing Your Project

- Periodically save your project by hitting the save icon at the top of your script window
- Now you can close your project
- It will be saved
- Now you can quit **RStudio**
- When you open **RStudio** next, you can **OPEN PROJECT** to get back into it