Intro to R

Data Input

Outline

- Part 0: A little bit of set up!
- Part 1: reading in manually (point and click)
- Part 2: reading in directly & working directories
- Part 3: checking data & multiple file formats

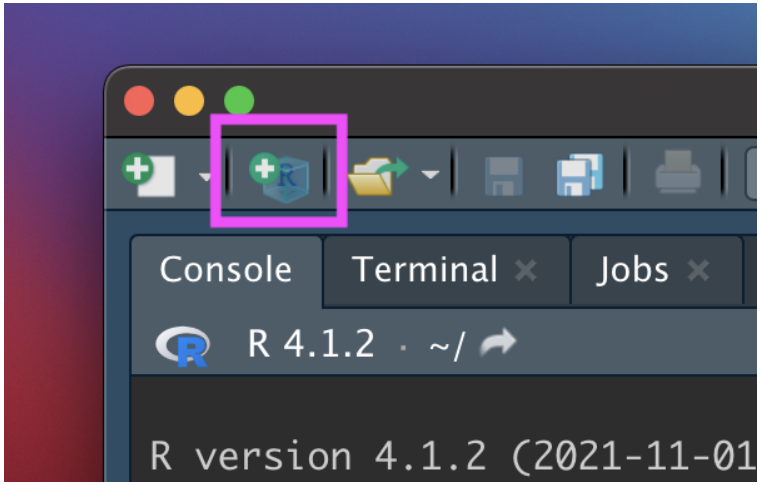
We will cover Output a bit later!

Part 0: Setup - R Project

New R Project

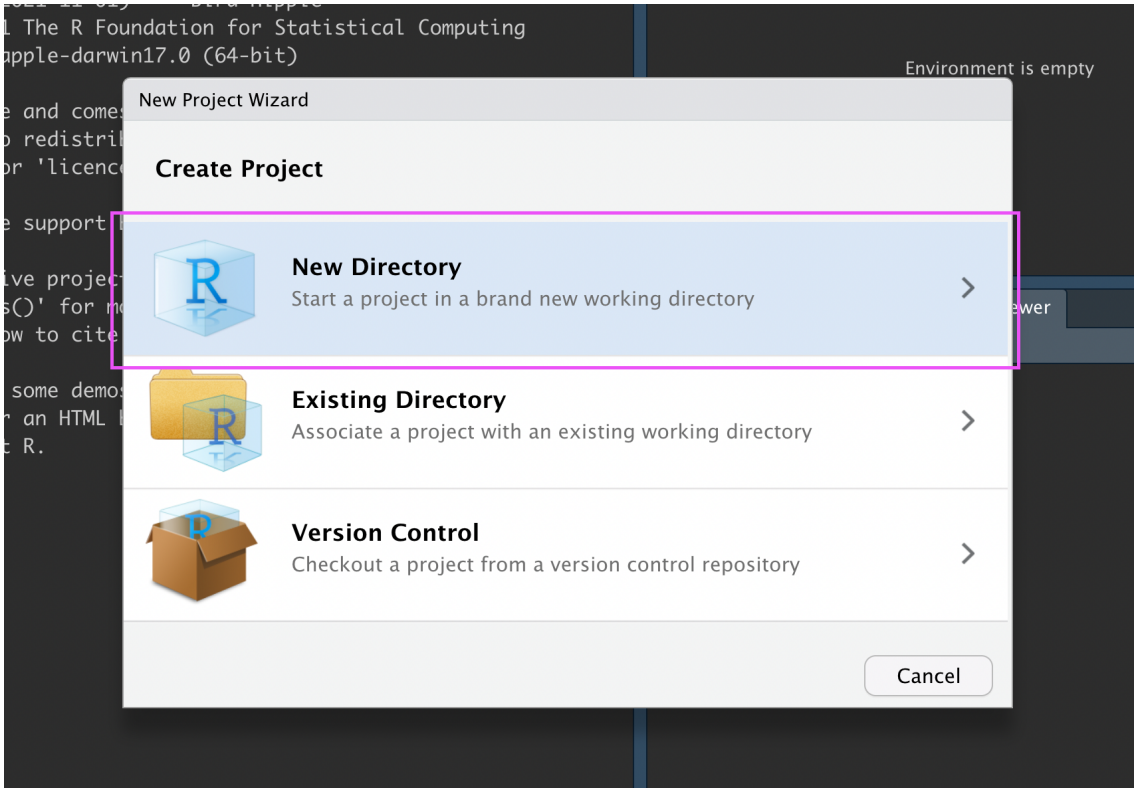
Let's make an R Project so we can stay organized in the next steps.

Click the new R Project button at the top left of RStudio:



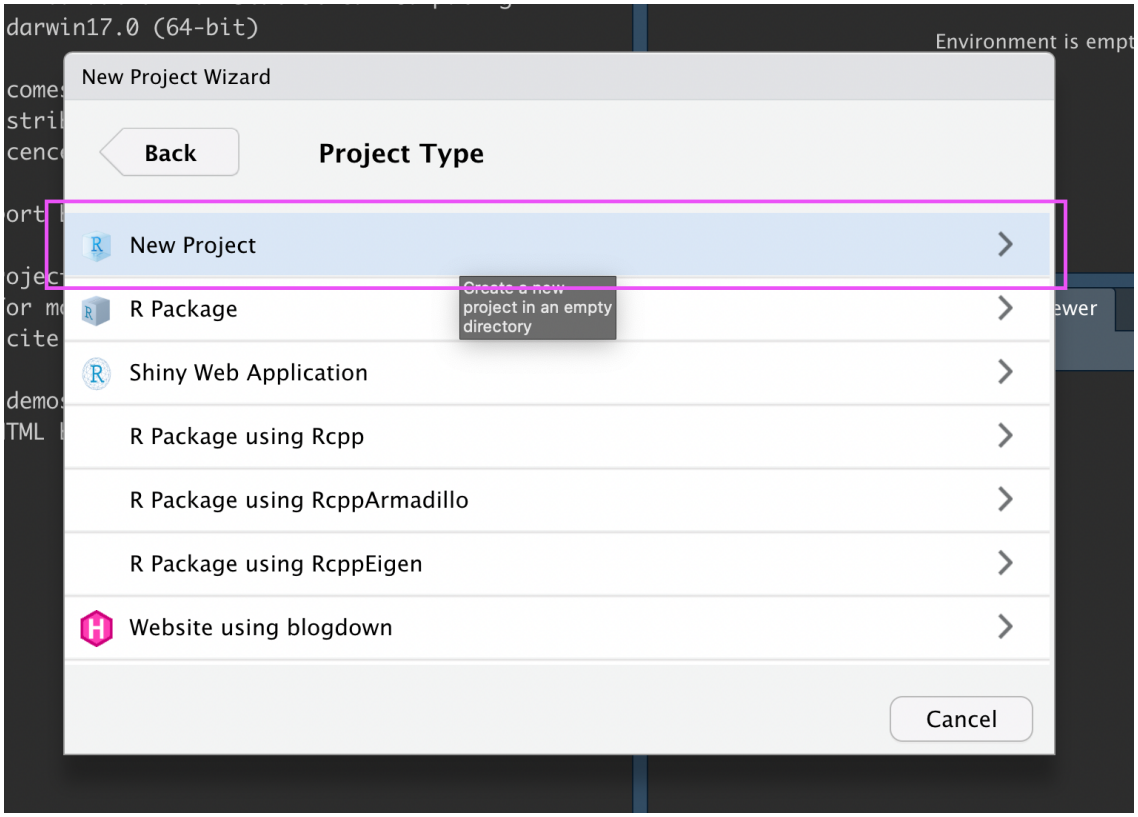
New R Project

In the New Project Wizard, click “New Directory”:



New R Project

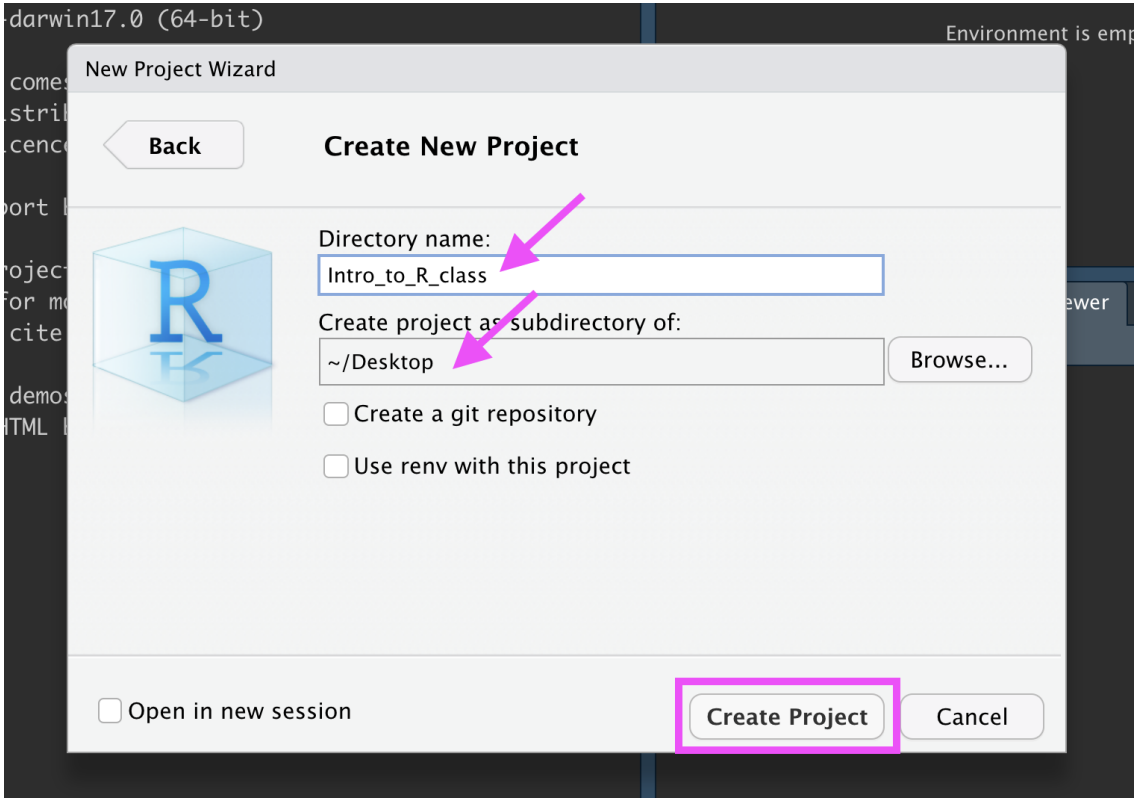
Click “New Project”:



New R Project

Type in a name for your new folder.

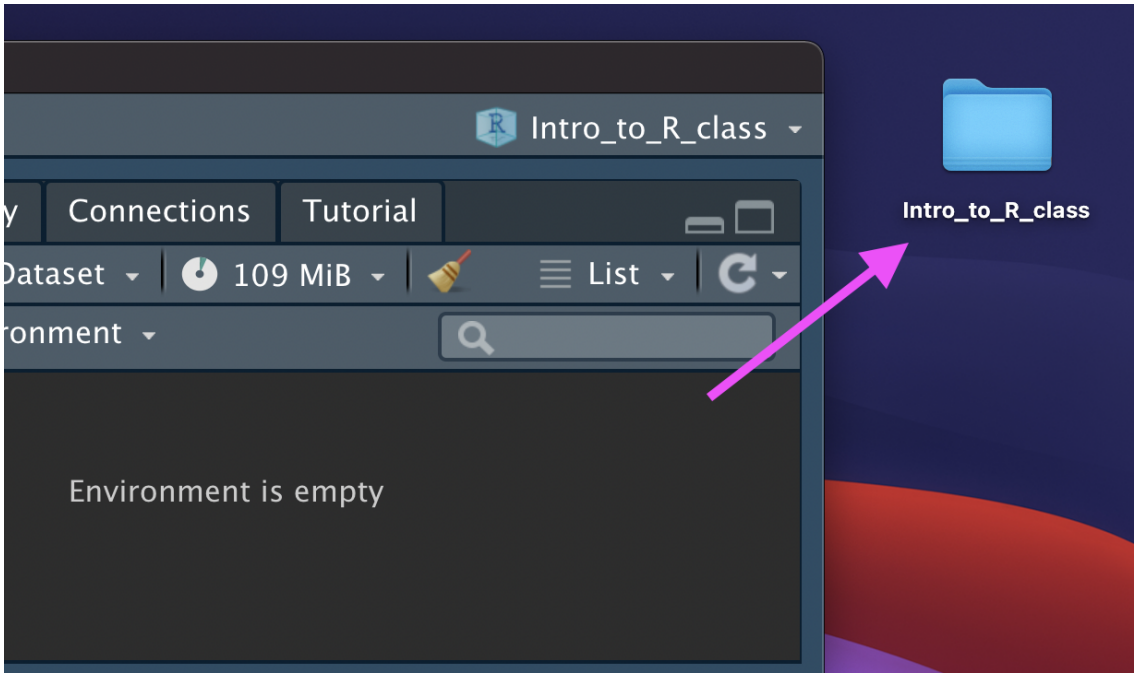
Store it somewhere easy to find, such as your Desktop:



New R Project

You now have a new R Project folder on your Desktop!

Make sure you add any scripts or data files to this folder as we go through today's lesson. This will make sure R is able to "find" your files.



Part 1: Getting data into R (manual/point and click)

Data Input

- 'Reading in' data is the first step of any real project/analysis
- R can read almost any file format, especially via add-on packages
- We are going to focus on simple delimited files first
  - comma separated (e.g. '.csv')
  - tab delimited (e.g. '.txt')
  - Microsoft Excel (e.g. '.xlsx')

Note: data for demonstration

- We have added functionality to load some datasets directly in the `jhur` package

Data Input

Youth Tobacco Survey (YTS) dataset:

"The YTS was developed to provide states with comprehensive data on both middle school and high school students regarding tobacco use, exposure to environmental tobacco smoke, smoking cessation, school curriculum, minors' ability to purchase or otherwise obtain tobacco products, knowledge and attitudes about tobacco, and familiarity with pro-tobacco and anti-tobacco media messages."

- Check out the data at: https://catalog.data.gov/dataset/youth-tobacco-survey-yts-data

Data Input: Dataset Location

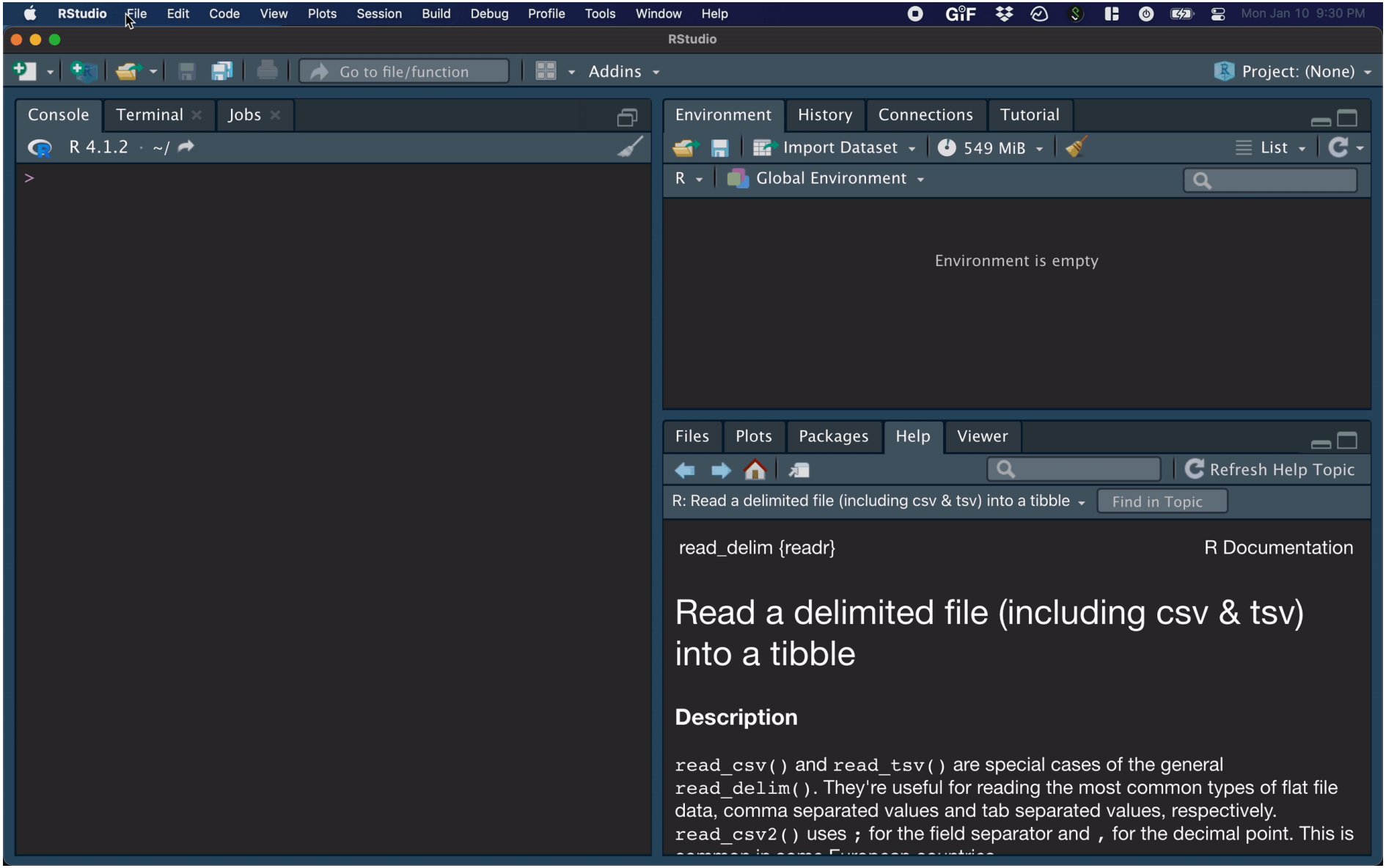
Dataset is located at
http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv

- Download data by clicking the above link
    - Safari - if a file loads in your browser, choose File –> Save As, select, Format “Page Source” and save

Import Dataset

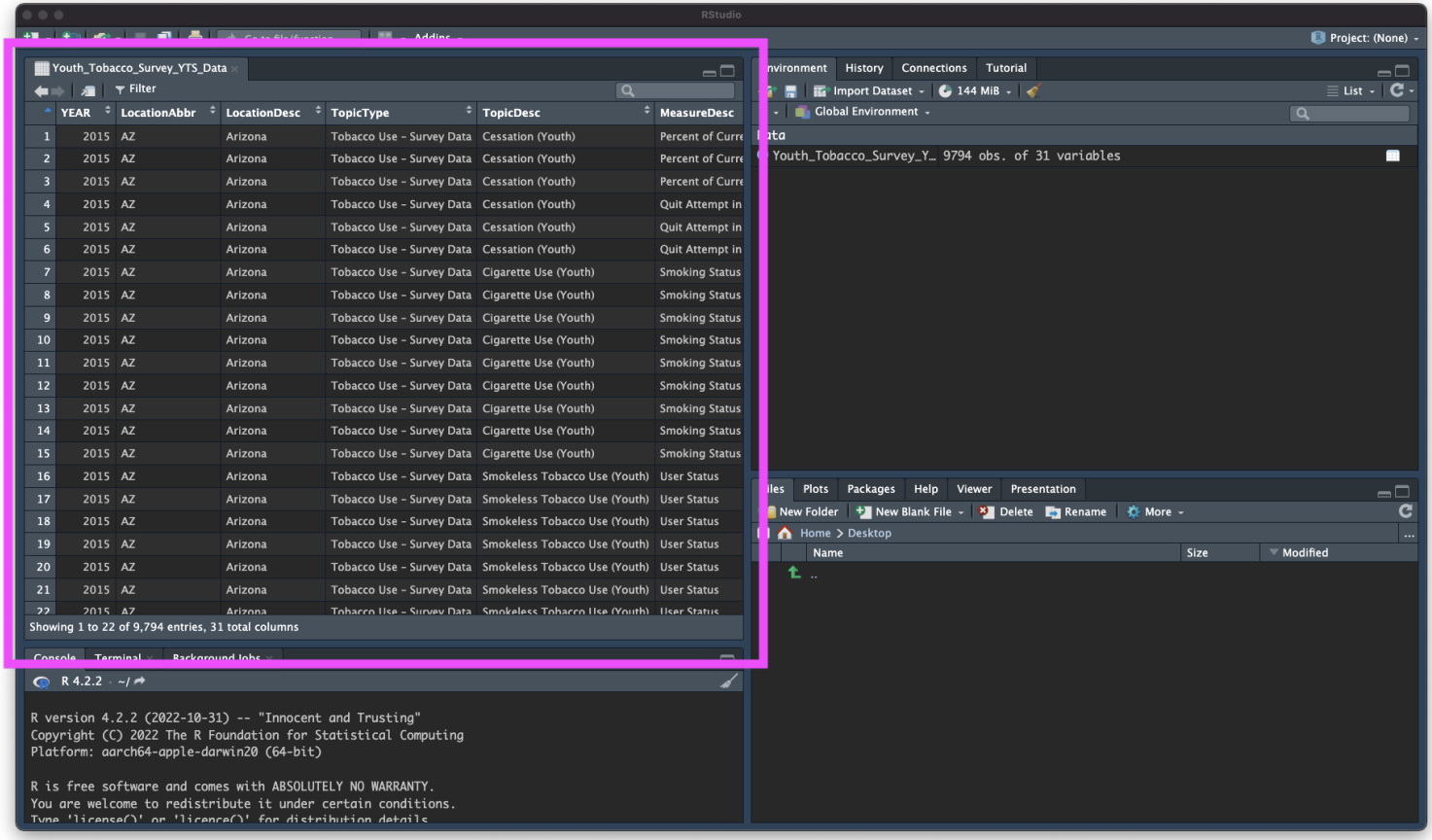
- > File
- > Import Dataset
- > From Text (`readr`)
- > paste the url (http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv)
- > click “Update” and “Import”

Import Dataset



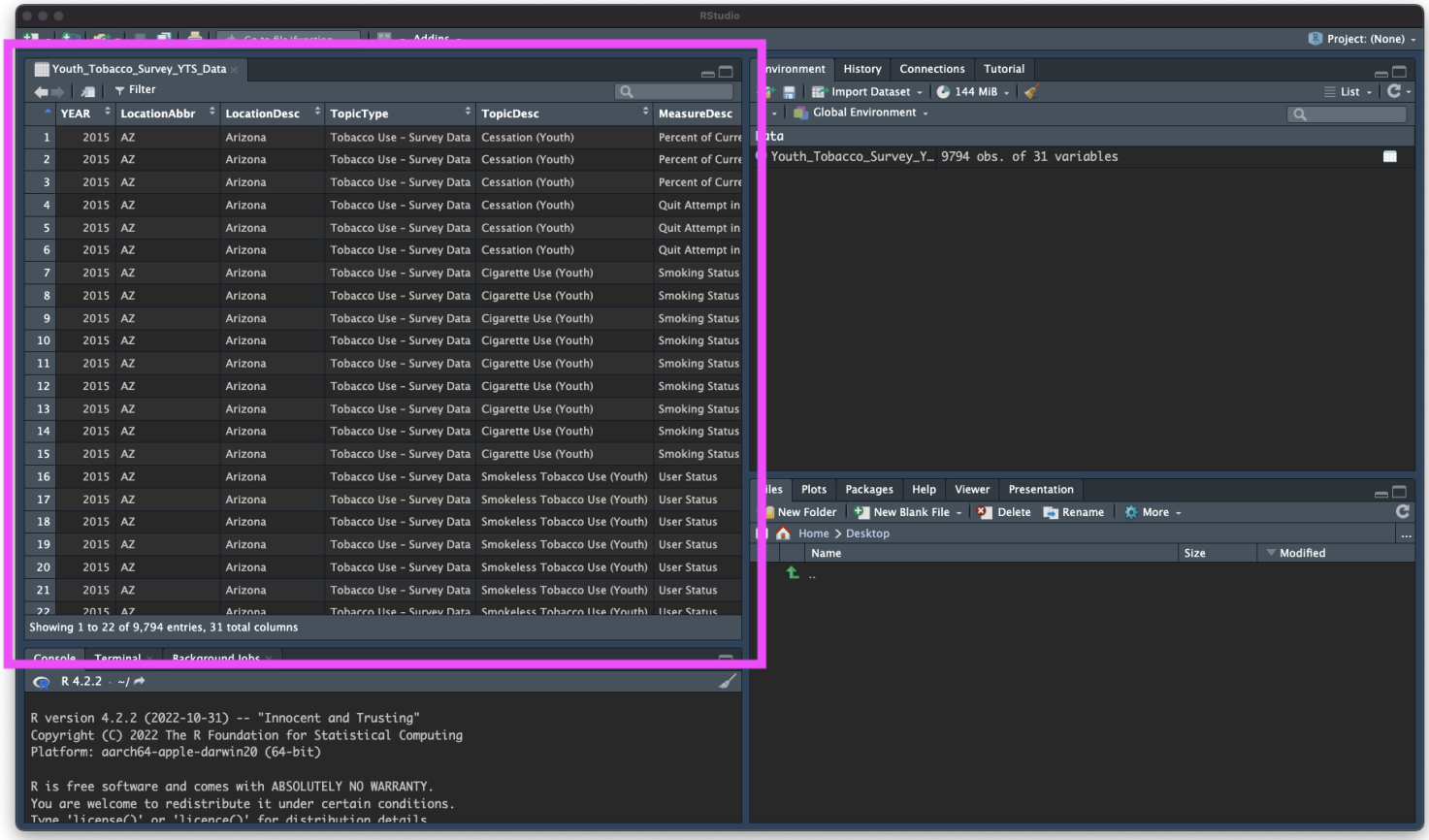
What Just Happened?

You see a preview of the data on the top left pane.



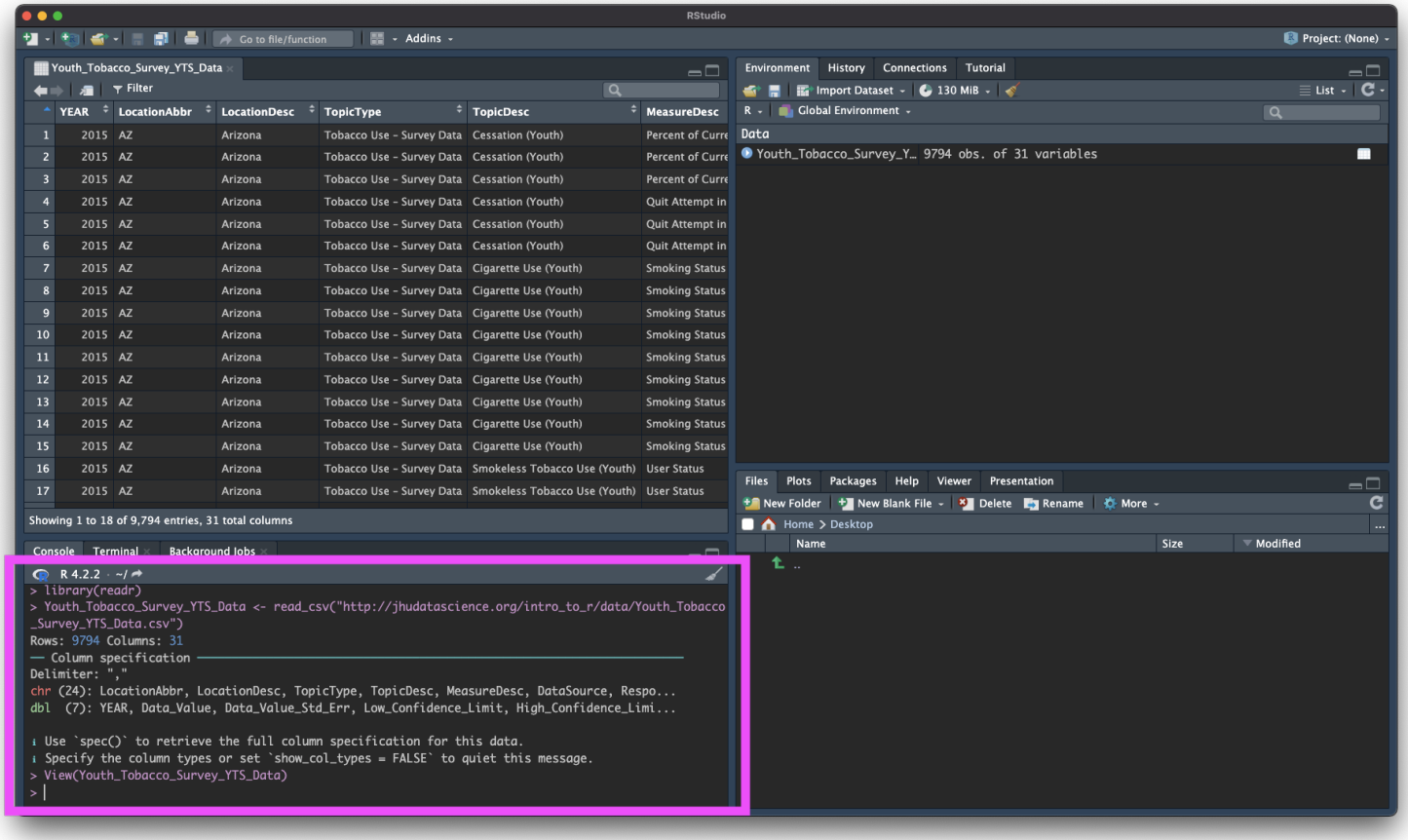
What Just Happened?

You see a new object called `Youth_Tobacco_Survey_YTS_Data` in your environment pane (top right). The table button opens the data for you to view.

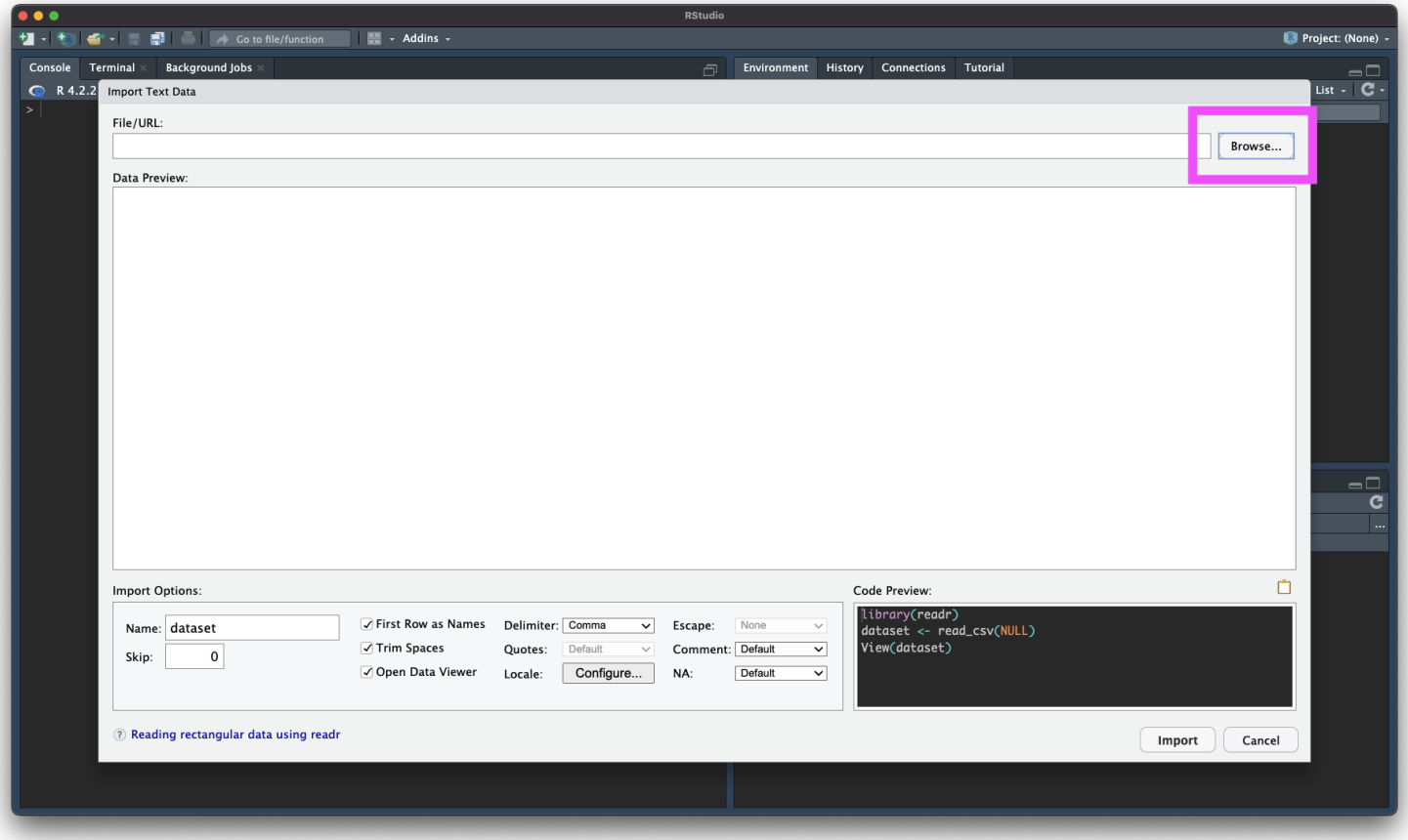


What Just Happened?

R ran some code in the console (bottom left).



Browsing for Data on Your Machine



Manual Import: Pros and Cons

Pros: easy!!

Cons: obscures some of what's happening, others will have difficulty running your code

Summary & Lab Part 1

Review the process: `https://youtu.be/LEkNfJgpunQ`

- > File
- > Import Dataset
- > From Text (`readr`)
- > paste the url (http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv)
- > click “Update” and “Import”

Class Website

Data Input Lab

Part 2: Getting data into R (directly)

Data Input: Read in Directly

```
# load library `readr` that contains function `read_csv`
library(readr)
dat <- read_csv(
  file = "http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv"
)

# `head` displays first few rows of a data frame. `tail()` works the same way.
head(dat, n = 5)
```

```
# A tibble: 5 × 31
   YEAR LocationAbbr LocationDesc TopicType     TopicDesc MeasureDesc DataSource
  <dbl> <chr>        <chr>        <chr>         <chr>     <chr>       <chr>
1  2015 AZ           Arizona      Tobacco Use … Cessatio… Percent of… YTS
2  2015 AZ           Arizona      Tobacco Use … Cessatio… Percent of… YTS
3  2015 AZ           Arizona      Tobacco Use … Cessatio… Percent of… YTS
4  2015 AZ           Arizona      Tobacco Use … Cessatio… Quit Attem… YTS
5  2015 AZ           Arizona      Tobacco Use … Cessatio… Quit Attem… YTS
# … with 24 more variables: Response <chr>, Data_Value_Unit <chr>,
#   Data_Value_Type <chr>, Data_Value <dbl>, Data_Value_Footnote_Symbol <chr>,
#   Data_Value_Footnote <chr>, Data_Value_Std_Err <dbl>,
#   Low_Confidence_Limit <dbl>, High_Confidence_Limit <dbl>, Sample_Size <dbl>,
#   Gender <chr>, Race <chr>, Age <chr>, Education <chr>, GeoLocation <chr>,
#   TopicTypeId <chr>, TopicId <chr>, MeasureId <chr>, StratificationID1 <chr>,
#   StratificationID2 <chr>, StratificationID3 <chr>, …
```

Data Input: Declaring Arguments

```
dat <- read_csv(
  file = "http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv"
)
# EQUIVALENT TO
dat <- read_csv(
  "http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv"
)
```

Data Input: Read in Directly

So what is going on “behind the scenes”?

`read_csv()` parses a “flat” text file (.csv) and turns it into a **tibble** – a rectangular data frame, where data are split into rows and columns

- First, a flat file is parsed into a rectangular matrix of strings
- Second, the type of each column is determined (heuristic-based guess)

Data Input: Read in Directly

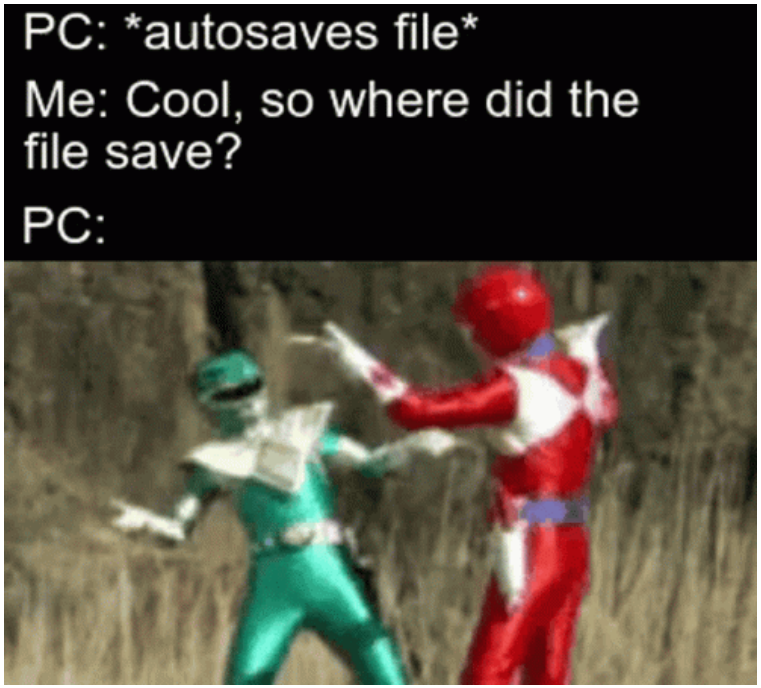
`read_csv()` needs an argument `file =`.

- `file` is the path to your file, **in quotation marks**
- can be path to a file on a website (URL)
- can be path in your local computer – absolute file path or relative file path

```
# Examples

dat <- read_csv(file = "/Users/avahoffman/Downloads/Youth_Tobacco_Survey_YTS_Data.csv")

dat <- read_csv(file = "Youth_Tobacco_Survey_YTS_Data.csv")

dat <- read_csv(file = "www.someurl.com/table1.csv")
```

Data Input: File paths

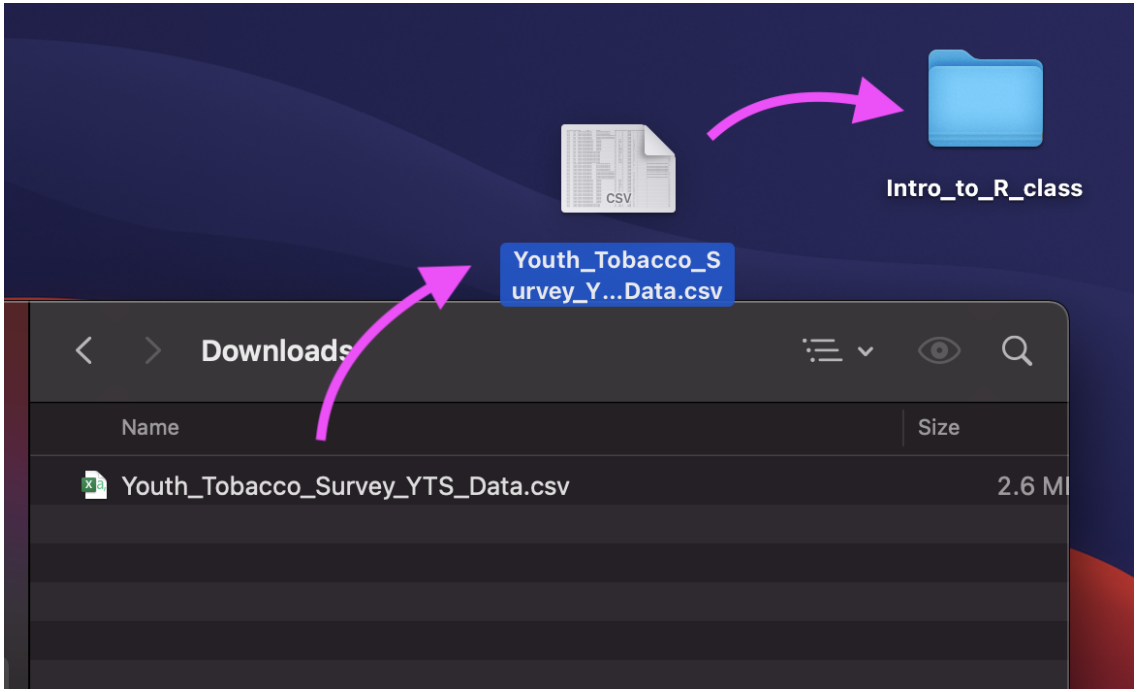
Reading from your computer.. What is my “path”?



Data Input: File paths

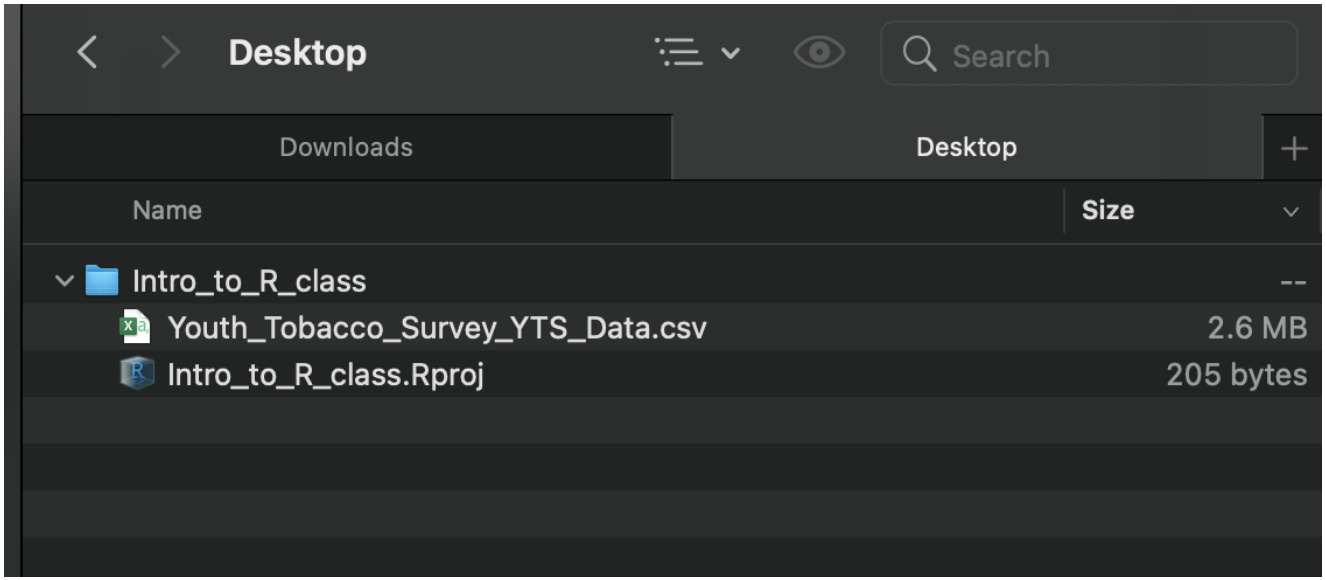
Luckily, we already set up an R Project!

First, we need to move the downloaded file into our R Project folder.



Data Input: File paths

Confirm the data is in the R Project folder.



Data Input: File paths

If we add the Youth_Tobacco_Survey_YTS_Data.csv file to the R Project folder, we can use the file name for the `file` argument:

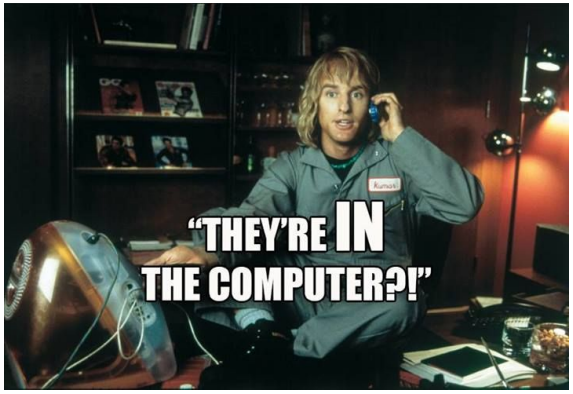
```
dat <- read_csv(file = "Youth_Tobacco_Survey_YTS_Data.csv")
```

Why does this work?

The Youth_Tobacco_Survey_YTS_Data.csv file is in the **working directory**.

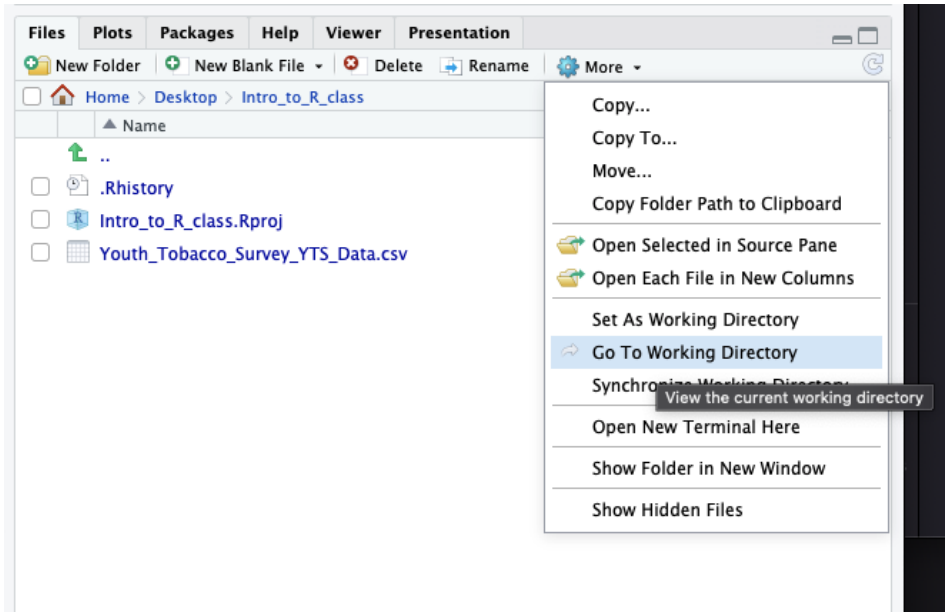
Working directory is a folder (directory) that RStudio assumes “you are working in”.

It’s where R looks for files.



The Working Directory

Click “More” > “Go To Working Directory” to locate your working directory.

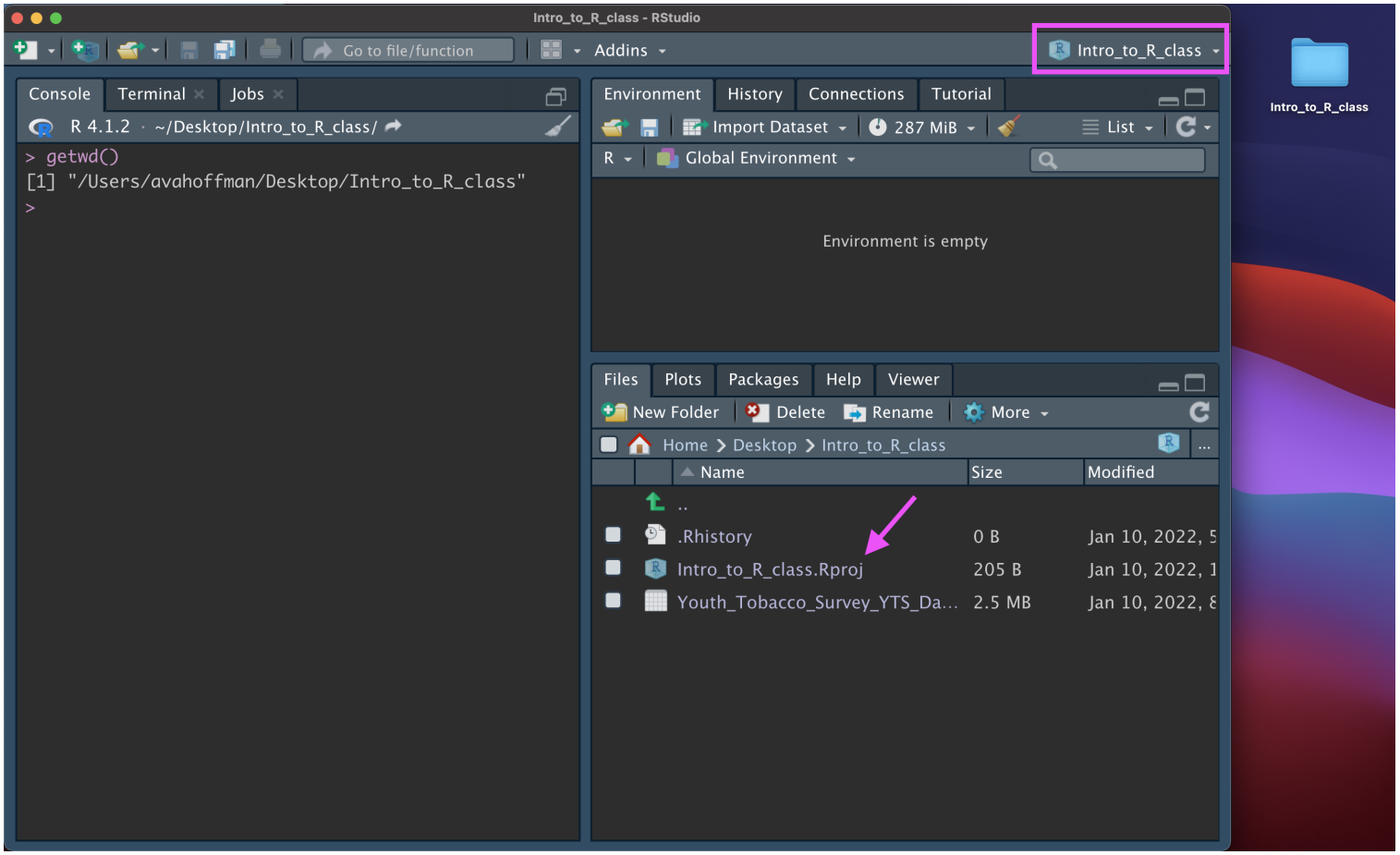


You can also run the `getwd()` function.

```
# Get the working directory
getwd()
```

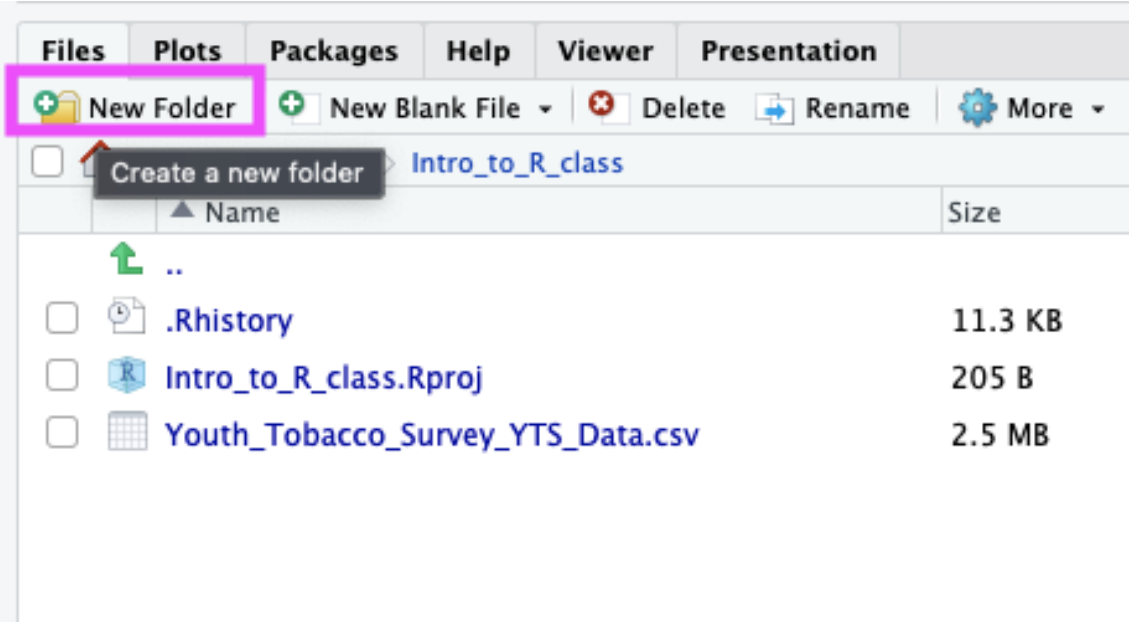
The Working Directory

Setting up an R Project can avoid headaches by telling R that the working directory is wherever the `.Rproj` file is.



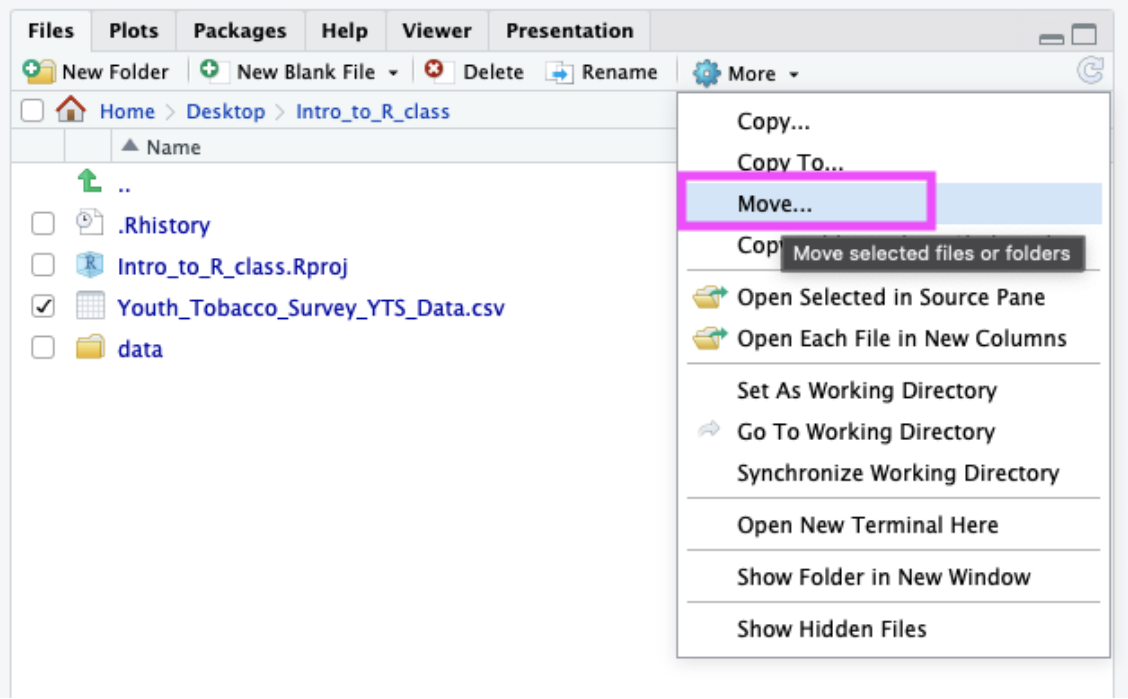
Data Input: Getting Organized!

Create a “data” folder in the Files pane (bottom right).



Data Input: Getting Organized

- Select the Youth_Tobacco_Survey_YTS_Data.csv file
- Select “More” > “Move”
- Select the data folder you just created



Data Input: Modify the path

If you move the file, **you must update the path!**

```
# No longer works!
dat <- read_csv(file = "Youth_Tobacco_Survey_YTS_Data.csv")
```

```
# Works!
dat <- read_csv(file = "data/Youth_Tobacco_Survey_YTS_Data.csv")
```

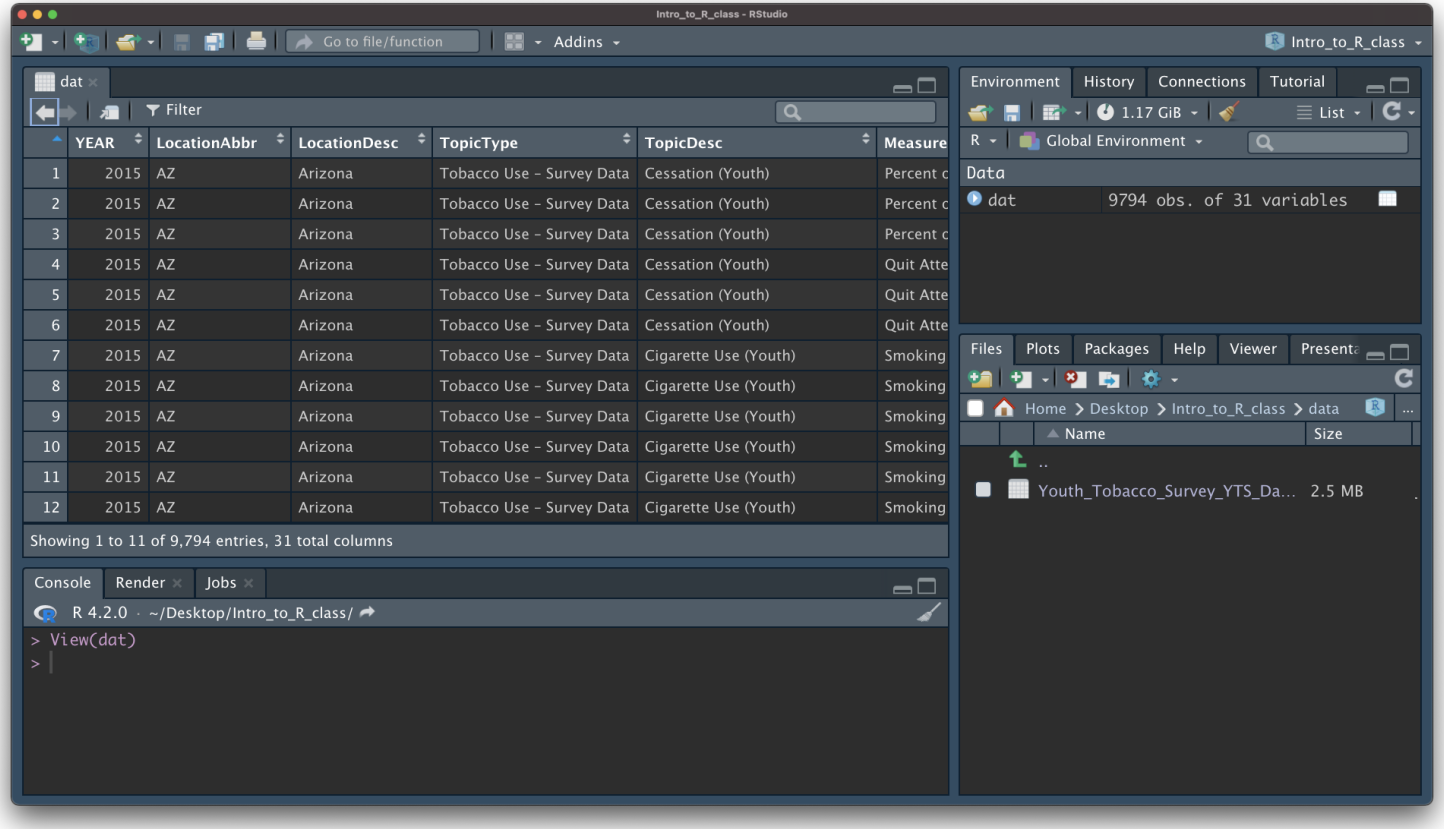
Confirm you read in the data by checking the “Environment” pane (top right).

Part 3: Checking data & Other formats

Data Input: Checking the data

- the `View()` function shows your data in a new tab, in spreadsheet format
- be careful if your data is big!

```
View(dat)
```



Data Input: Checking the data

The `str()` function can tell you about data/objects(different variables and their classes - more on this later). We will also discuss the `glimpse()` function later, which does something very similar.

```
str(dat)
```

```
spec_tbl_df [9,794 × 31] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ YEAR                      : num [1:9794] 2015 2015 2015 2015 2015 ...
 $ LocationAbbr              : chr [1:9794] "AZ" "AZ" "AZ" "AZ" ...
 $ LocationDesc              : chr [1:9794] "Arizona" "Arizona" "Arizona" "Arizona" ...
 $ TopicType                 : chr [1:9794] "Tobacco Use – Survey Data" "Tobacco Use – Survey D
 $ TopicDesc                 : chr [1:9794] "Cessation (Youth)" "Cessation (Youth)" "Cessation
 $ MeasureDesc               : chr [1:9794] "Percent of Current Smokers Who Want to Quit" "Perc
 $ DataSource                : chr [1:9794] "YTS" "YTS" "YTS" "YTS" ...
 $ Response                  : chr [1:9794] NA NA NA NA ...
 $ Data_Value_Unit           : chr [1:9794] "%" "%" "%" "%" ...
 $ Data_Value_Type           : chr [1:9794] "Percentage" "Percentage" "Percentage" "Percentage"
 $ Data_Value                : num [1:9794] NA NA NA NA NA NA 3.2 3.2 3.1 12.5 ...
 $ Data_Value_Footnote_Symbol: chr [1:9794] "*" "*" "*" "*" ...
 $ Data_Value_Footnote       : chr [1:9794] "Data in these cells have been suppressed because c
 $ Data_Value_Std_Err        : num [1:9794] NA NA NA NA NA NA 1.5 1.5 1.6 2.7 ...
 $ Low_Confidence_Limit      : num [1:9794] NA NA NA NA NA NA 0.3 0.3 0.1 7.2 ...
 $ High_Confidence_Limit     : num [1:9794] NA NA NA NA NA NA 6.1 6.2 6.1 17.9 ...
 $ Sample_Size               : num [1:9794] NA NA NA NA NA ...
 $ Gender                    : chr [1:9794] "Overall" "Male" "Female" "Overall" ...
 $ Race                      : chr [1:9794] "All Races" "All Races" "All Races" "All Races" ...
 $ Age                       : chr [1:9794] "All Ages" "All Ages" "All Ages" "All Ages" ...
 $ Education                 : chr [1:9794] "Middle School" "Middle School" "Middle School" "Mi
 $ GeoLocation               : chr [1:9794] "(34.865970280000454, -111.76381127699972)" "(34.86
```

Data Input: Other delimiters with `read_delim()`

`read_csv()` is a special case of `read_delim()` – a general function to read a delimited file into a data frame

`read_delim()` needs path to your file and **file's delimiter**, will return a tibble

- `file` is the path to your file, in quotes
- `delim` is what separates the fields within a record

```
## Examples
dat <- read_delim(file = "www.someurl.com/table1.tsv", delim = "\t")

dat <- read_delim(file = "data.txt", delim = "|")
```

Data Input: Excel files

- You **cannot** read in an excel file from a URL.
- Need to load the `readxl` package with `library()`.
- The argument is `path` (not `file`).

```
library(readxl)

read_excel(path = "asthma.xlsx")
```

Data input: other file types

- haven package has functions to read SAS, SPSS, Stata formats

```
library(haven)

# SAS
read_sas(file = "mtcars.sas7bdat")

# SPSS
read_sav(file = "mtcars.sav")

# Stata
read_dta(file = "mtcars.dta")
```

Data Input: base R

There are also data importing functions provided in base R (rather than the `readr` package), like `read.delim()` and `read.csv()`.

These functions have slightly different syntax for reading in data (e.g. `header` argument).

However, while many online resources use the base R tools, the latest version of RStudio switched to use these new `readr` data import tools, so we will use them in the class for slides. They are also up to two times faster for reading in large datasets, and have a progress bar which is nice.

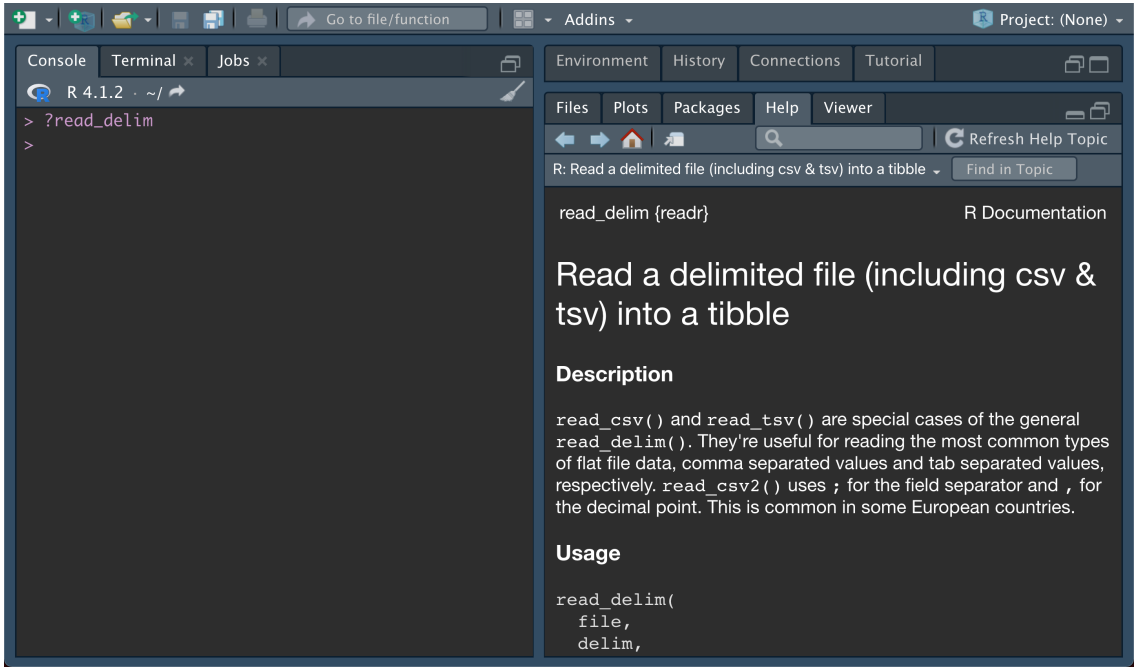
TROUBLESHOOTING: Common new user mistakes we have seen

1. **Working directory problems: trying to read files that R "can't find"**
   - Path misspecification
   - more on this shortly!
2. Typos (R is **case sensitive**, `x` and `X` are different)
   - RStudio helps with "tab completion"
3. Open ended quotes, parentheses, and brackets
4. Different versions of software
5. Deleting part of the code chunk

TROUBLESHOOTING: Help

For any function, you can write `?FUNCTION_NAME`, or `help("FUNCTION_NAME")` to look at the help file:

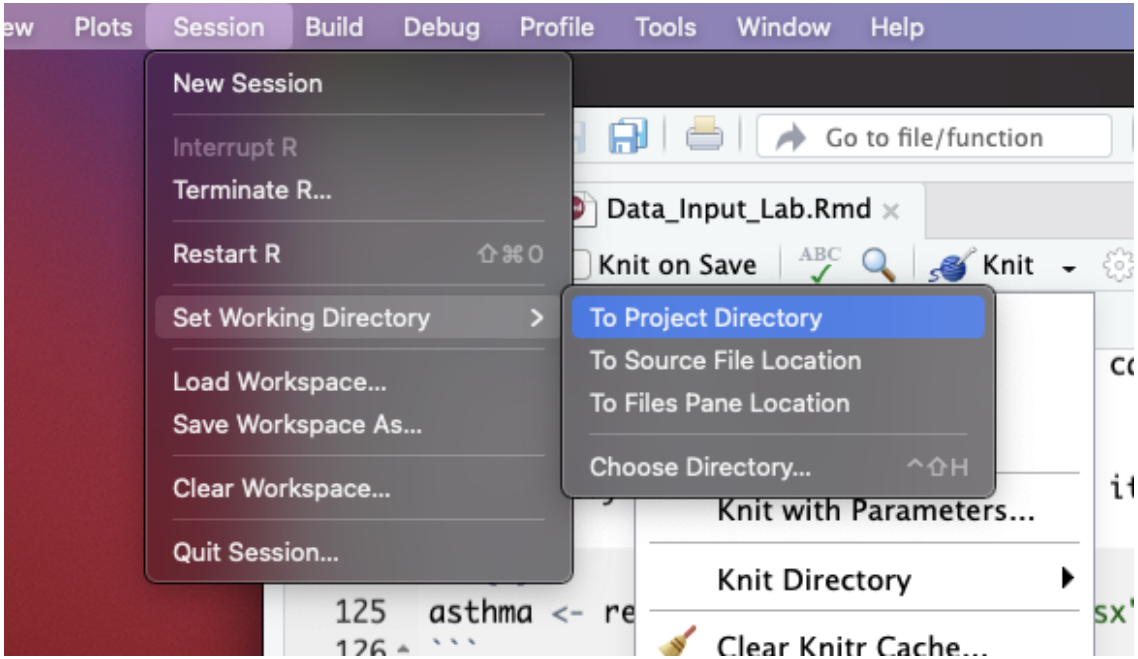
```
?read_delim
help("read_delim")
```



TROUBLESHOOTING: Setting the working directory

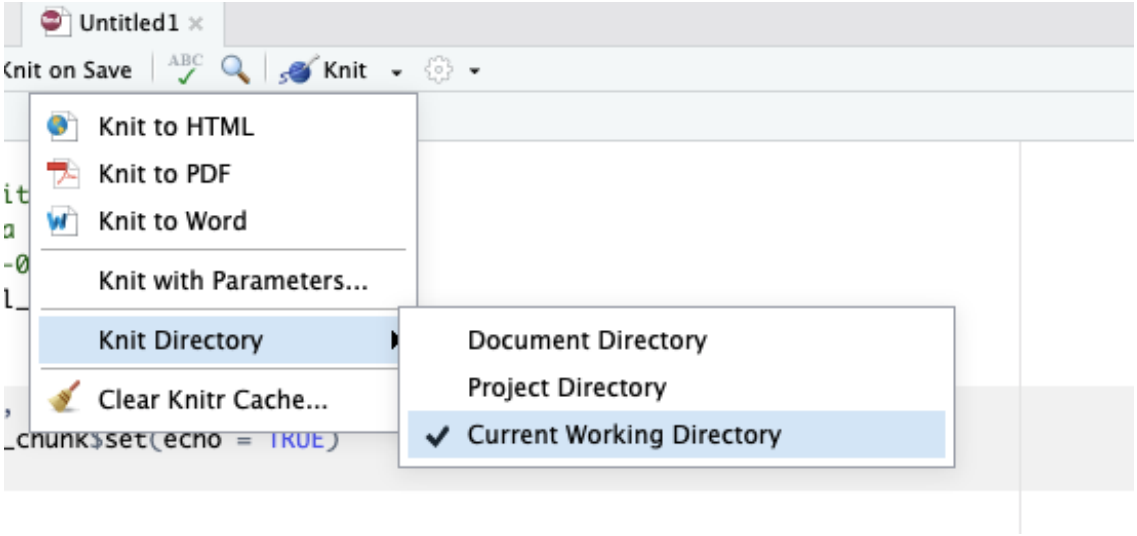
If your R project directory and working directory do not match:

- Session > Set Working Directory > To Project Directory



TROUBLESHOOTING: Setting the working directory

If you are trying to knit your work, it might help to set the knit directory to the "Current Working Directory":



TROUBLESHOOTING: Setting the working directory

You can also set the working directory manually with the `setwd()` function:

```
# set the working directory
setwd("/Users/avahoffman/Desktop")
```

Summary - Part 2

Functions from `readr` R package: `read_csv()` and `read_delim()`

- comma delimited data
- needs a file path to be provided
- returns a tibble (data frame)

R Projects are a good way to keep your files organized and reduce headaches

- Use `getwd()` to check your working directory, where R looks for your data files

Summary - Part 3

Look at your data!

- `glimpse()` gets a quick look at the column structure
- `View()` gives you a preview of the data in a new tab
- `head()` shows first few rows
- `tail()` shows the last few rows

Other file types

- `readr` package: `read_delim()` for general delimited files
- `readxl` package: `read_excel()` for Excel files

Don't forget to use `<-` to assign your data to an object!

Lab Part 2

Class Website

Data Input Lab

Image by Gerd Altmann from Pixabay