

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

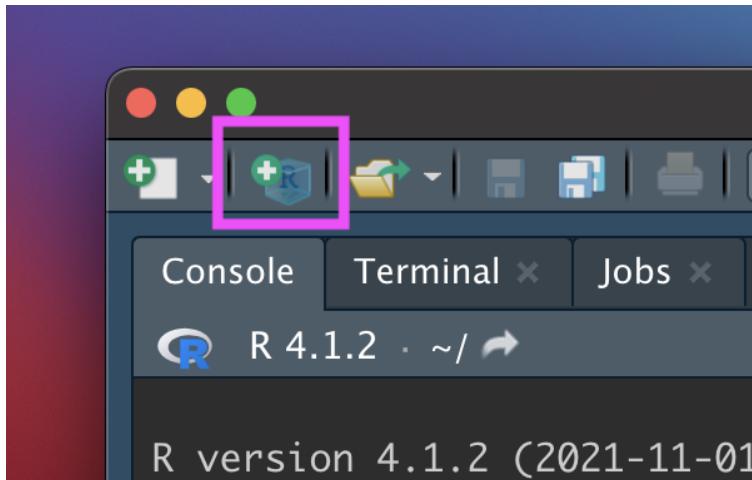
We will cover Output a bit later!

Part 0: Setup - R Project

New R Project

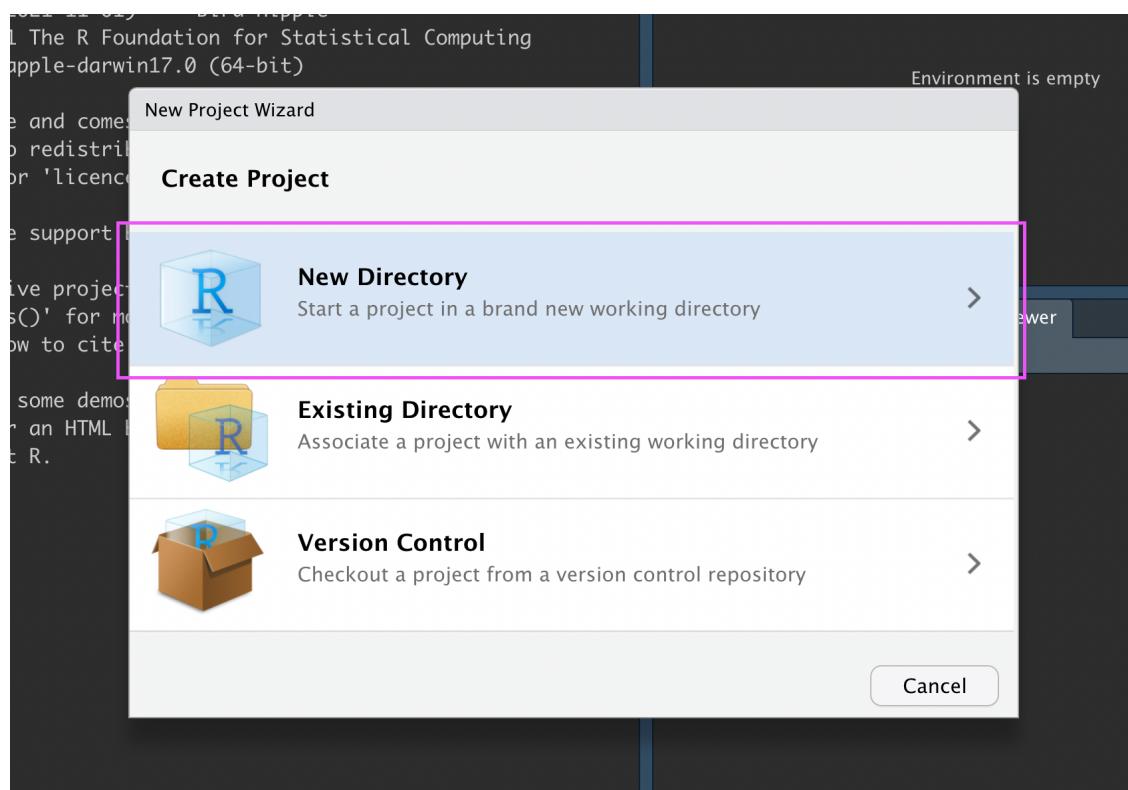
Let's make an R Project so we can stay organized in the next steps.

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



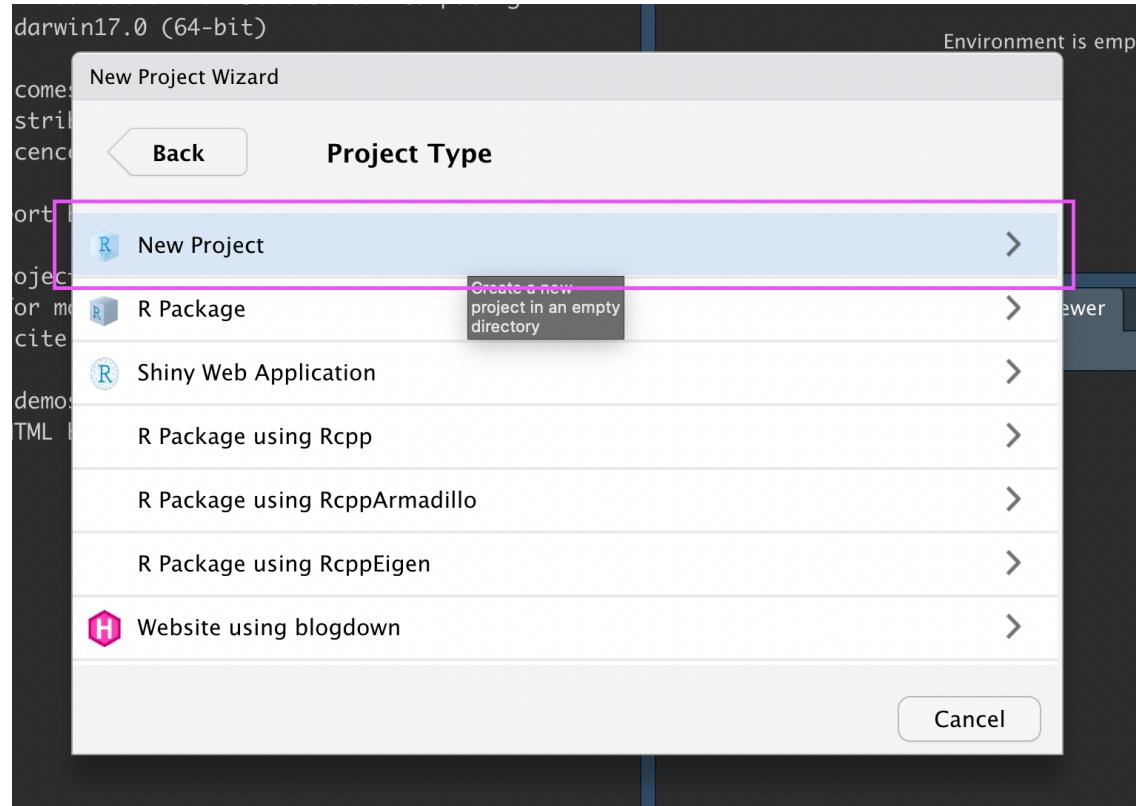
New R Project

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



New R Project

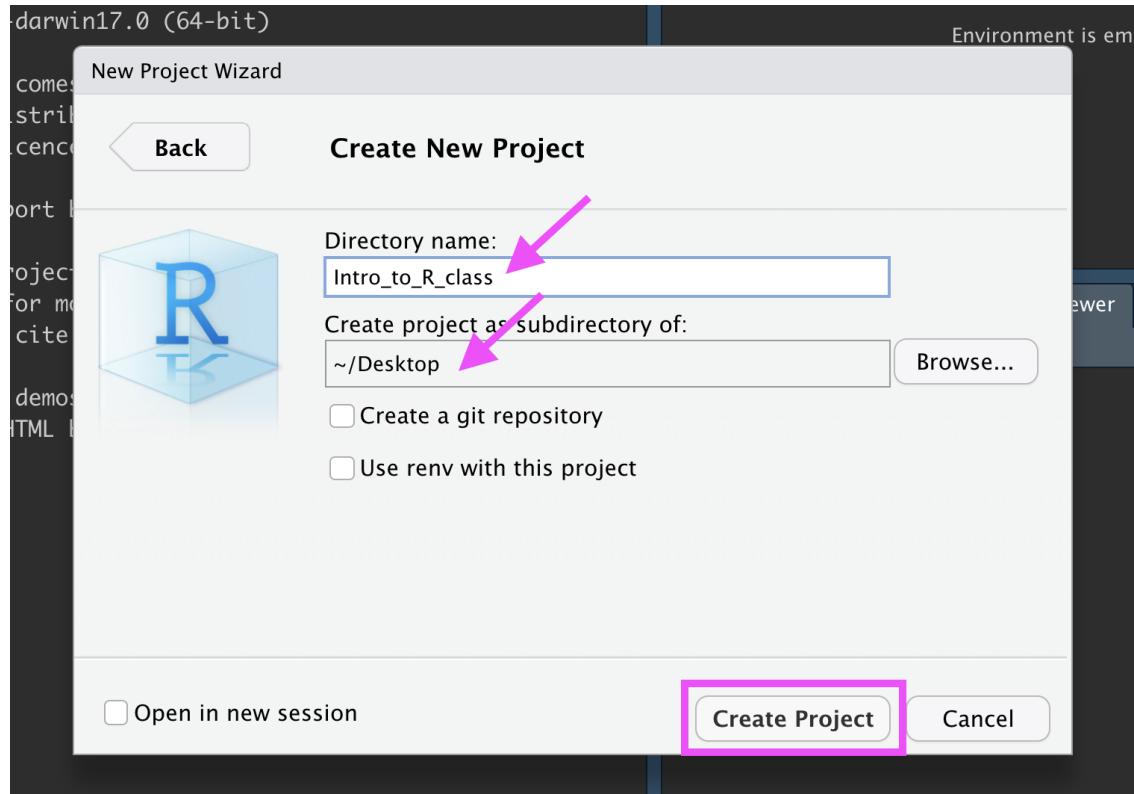
Click “New Project”:



New R Project

Type in a name for your new folder.

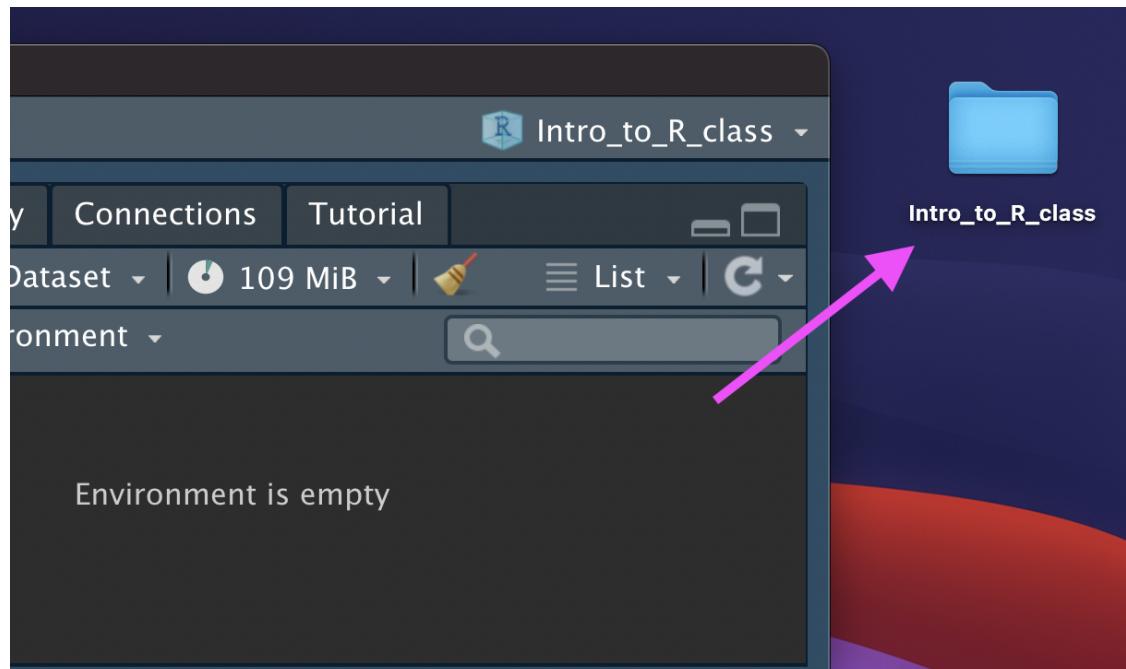
Store it somewhere easy to find, such as your Desktop:



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://jhubdatascience.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

The screenshot shows the RStudio interface with a dark theme. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, and Help. The status bar at the top right shows "Mon Jan 10 9:30 PM". The main window has tabs for Console, Terminal, and Jobs. The Environment pane is open, showing tabs for Environment, History, Connections, and Tutorial. The Environment tab displays the message "Environment is empty". The Global Environment section shows an R icon and a search bar. Below the Environment pane is a help viewer window. The title bar of the help viewer says "Files Plots Packages Help Viewer". The main content area shows the topic "R: Read a delimited file (including csv & tsv) into a tibble". The topic name is "read_delim {readr}" and it is categorized under "R Documentation". The description explains that `read_csv()` and `read_tsv()` are special cases of the general `read_delim()`. It notes that they are useful for reading common flat file types like CSV and TSV. The text also mentions that `read_csv2()` uses ; for the field separator and , for the decimal point, which is common in some European countries.

R 4.1.2 · ~/

>

Environment is empty

Environment History Connections Tutorial

Import Dataset 549 MiB

Global Environment

Files Plots Packages Help Viewer

R: Read a delimited file (including csv & tsv) into a tibble

read_delim {readr}

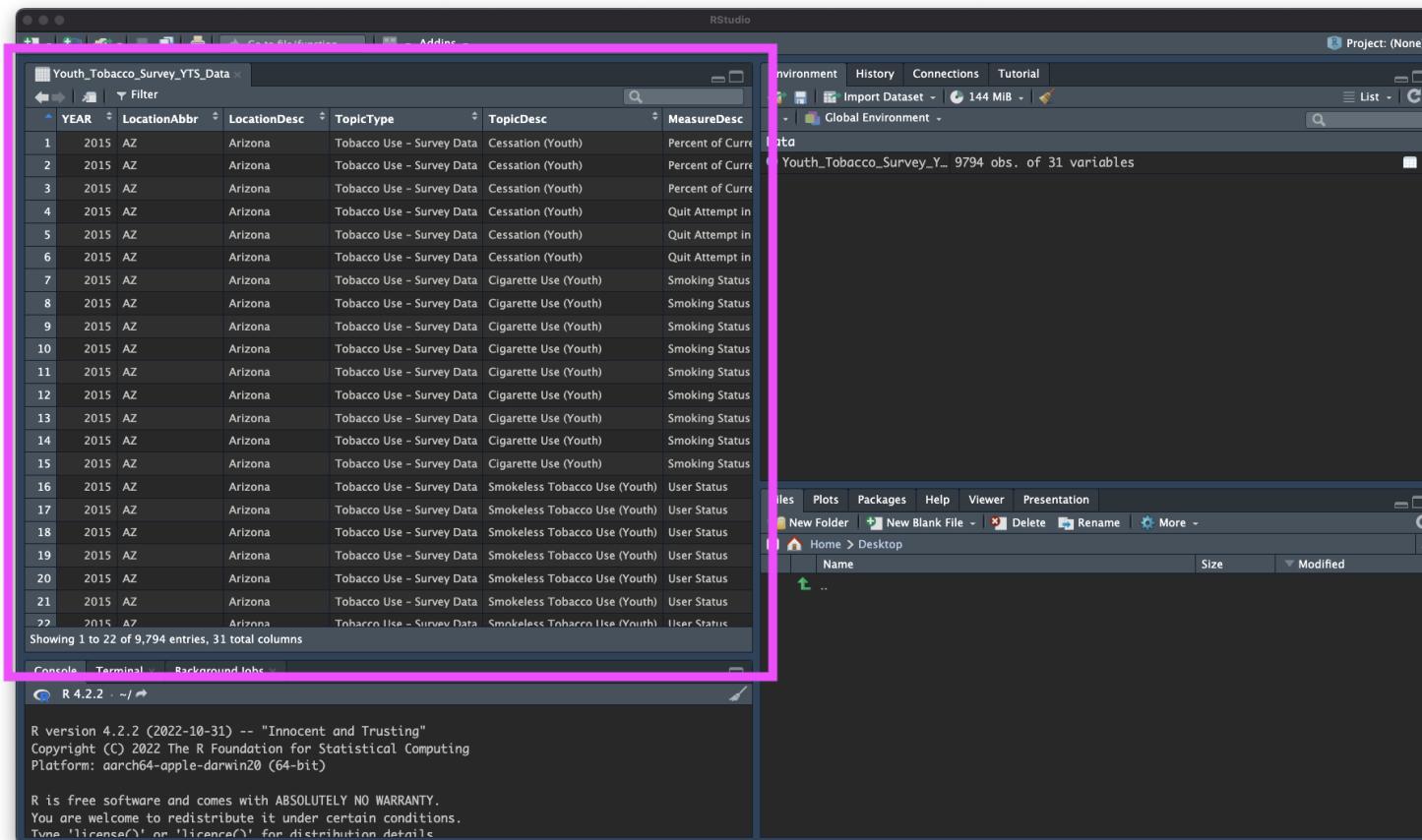
Read a delimited file (including csv & tsv) into a tibble

Description

`read_csv()` and `read_tsv()` are special cases of the general `read_delim()`. They're useful for reading the most common types of flat file data, comma separated values and tab separated values, respectively. `read_csv2()` uses ; for the field separator and , for the decimal point. This is common in some European countries.

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.

The screenshot shows the RStudio interface. On the left, a table titled "Youth_Tobacco_Survey_YTS_Data" is displayed, showing 22 rows of data. The columns are: YEAR, LocationAbbr, LocationDesc, TopicType, TopicDesc, and MeasureDesc. The data primarily consists of rows where YEAR is 2015 and LocationAbbr is AZ (Arizona), with various TopicType and TopicDesc entries like "Tobacco Use - Survey Data" and "Cessation (Youth)". On the right, the "Environment" pane shows the object "Youth_Tobacco_Survey_YTS_Data" with the description "9794 obs. of 31 variables". At the bottom, the R console window displays the R version information and the copyright notice.

```
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (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.
```

What Just Happened?

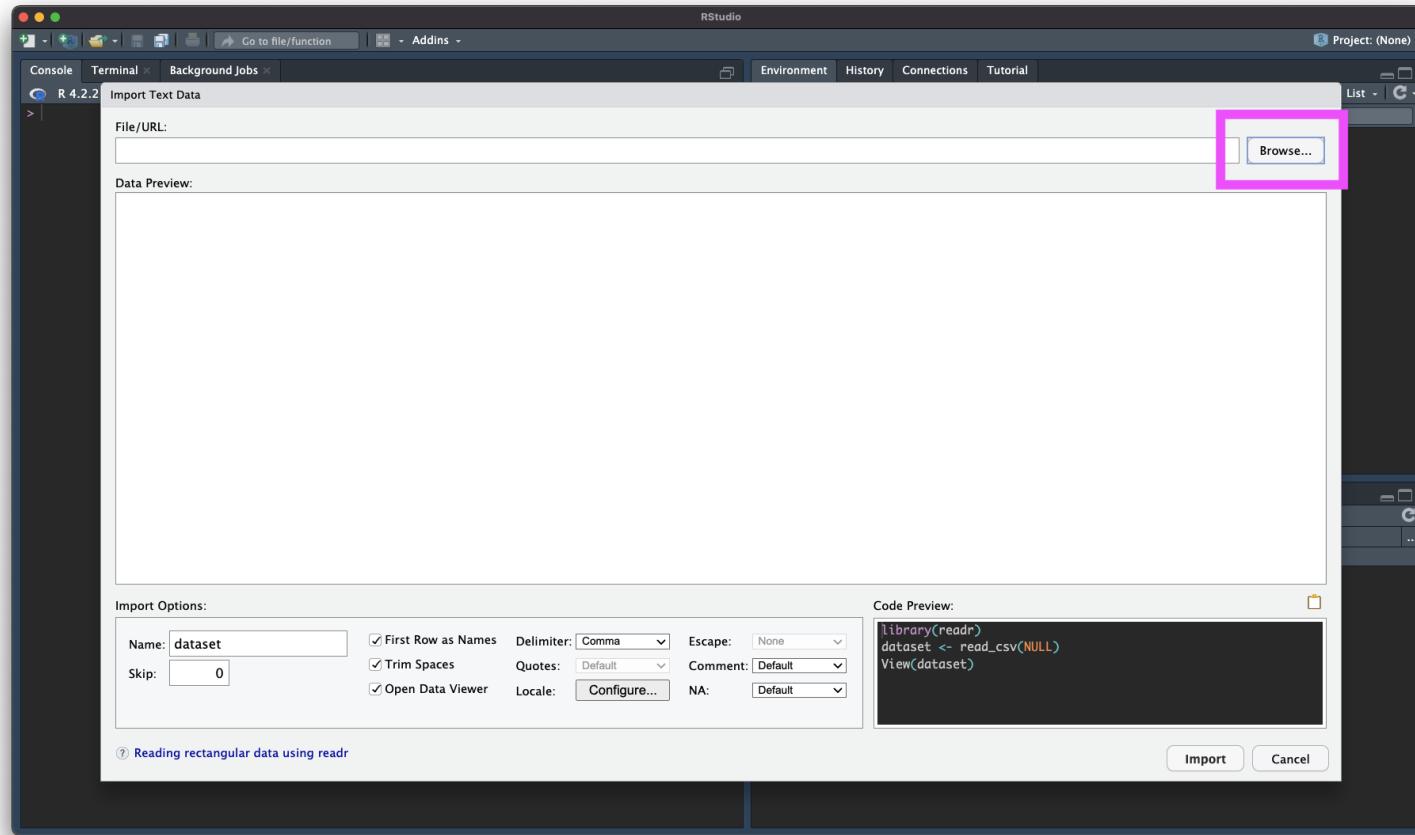
R ran some code in the console (bottom left).

The screenshot shows the RStudio interface. At the top, there's a menu bar with options like File, Edit, View, Insert, Tools, Help, and Addins. Below the menu is a toolbar with various icons. The main area is divided into several panes:

- Data View:** A large pane on the left displaying a table titled "Youth_Tobacco_Survey_YTS_Data". The table has columns: YEAR, LocationAbbr, LocationDesc, TopicType, TopicDesc, and MeasureDesc. The data shows rows from 1 to 17, all corresponding to the year 2015 and location Arizona.
- Environment:** A pane at the top right showing the global environment. It lists "Youth_Tobacco_Survey_YTS_Data" with 9794 observations and 31 variables.
- Files:** A pane at the bottom right showing a file tree with "Home > Desktop".
- Console:** The bottom-left pane where R code is run. It has tabs for "Console", "Terminal", and "Background jobs". The "Console" tab is active, showing the following R session:

```
R 4.2.2 - /~>
> library(readr)
> Youth_Tobacco_Survey_YTS_Data <- read_csv("http://jhubdatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv")
Rows: 9794 Columns: 31
-- Column specification --
Delimiter: ","
chr (24): LocationAbbr, LocationDesc, TopicType, TopicDesc, MeasureDesc, DataSource, Respo...
dbl (7): YEAR, Data_Value, Data_Value_Std_Err, Low_Confidence_Limit, High_Confidence_Limi...
# Use `spec()` to retrieve the full column specification for this data.
# Specify the column types or set `show_col_types = FALSE` to quiet this message.
> View(Youth_Tobacco_Survey_YTS_Data)
> |
```

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://jhubdatascience.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)
```

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”?

PC: *autosaves file*

Me: Cool, so where did the
file save?

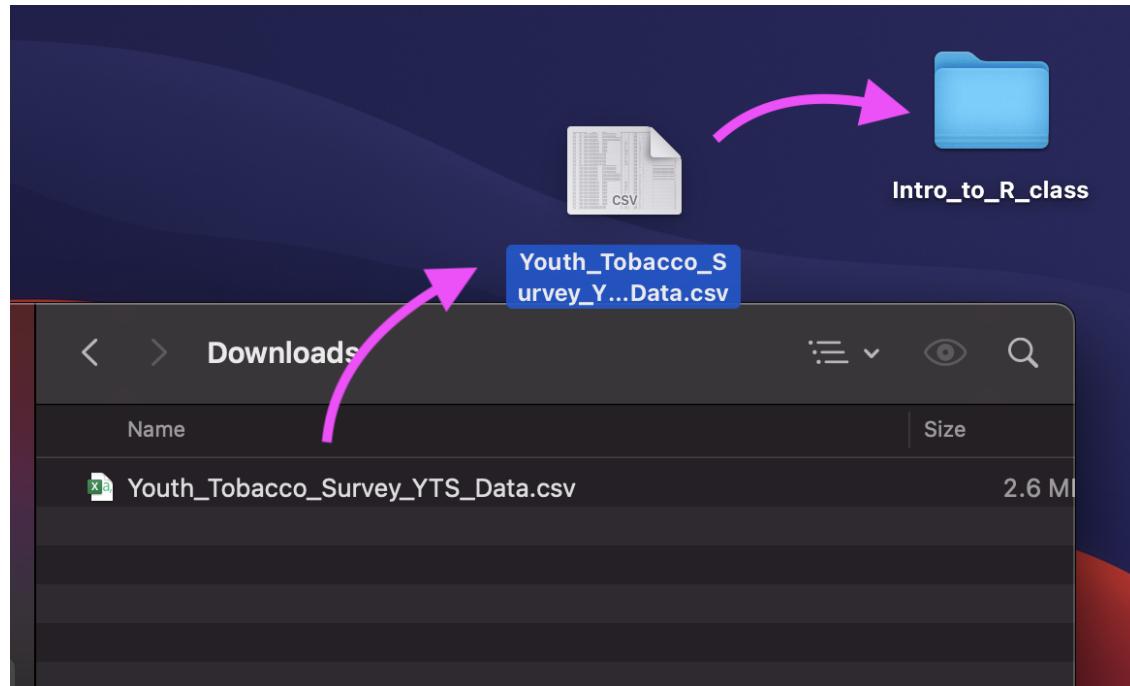
PC:



Data Input: File paths

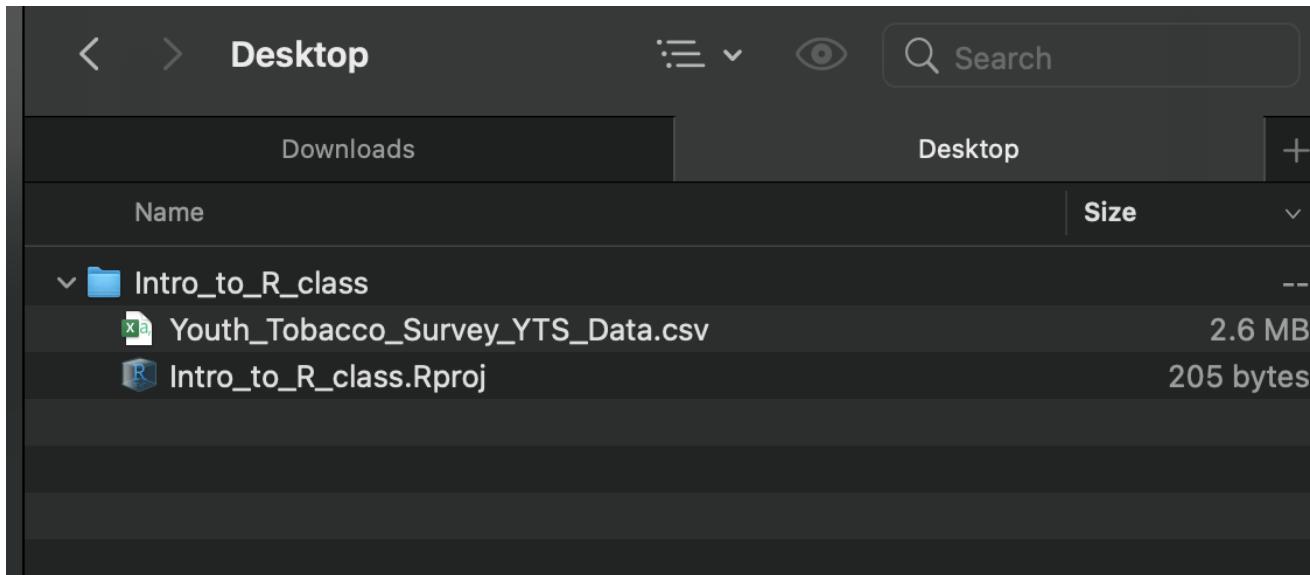
Luckily, we already set up an R Project!

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:

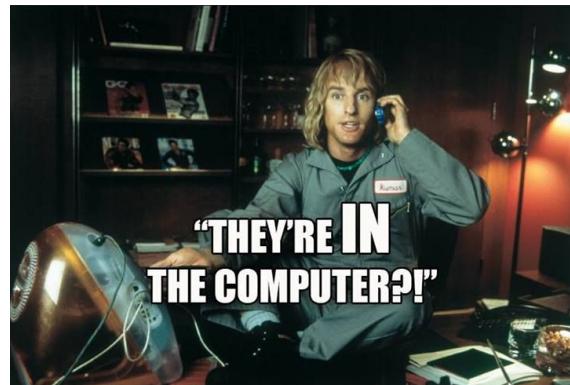
```
dat <- read_csv(file = "Youth_Tobacco_Survey_YTS_Data.csv")
```

Why does this work?

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

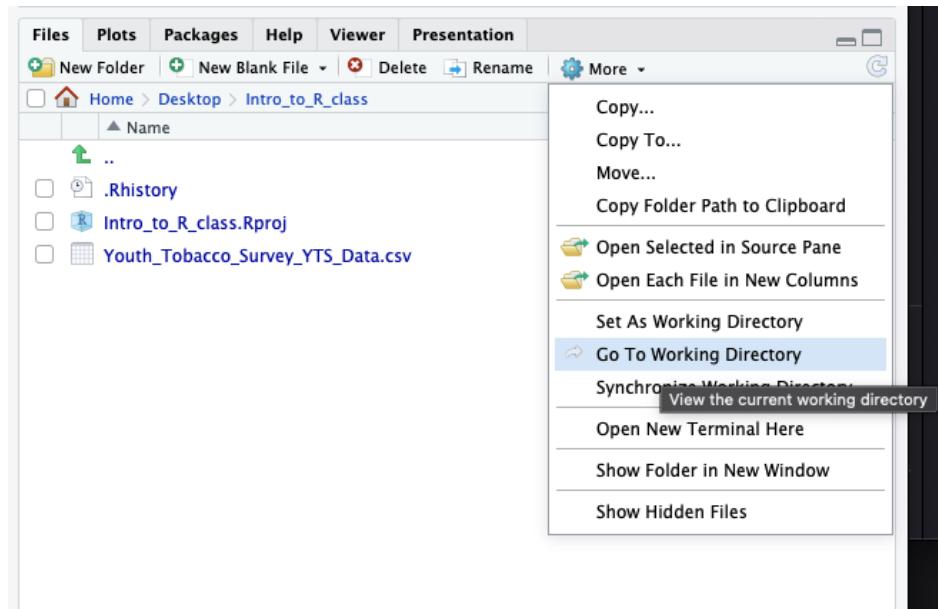
Working directory is a folder (directory) that RStudio assumes “you are working in”.

It's where R looks for files.



The Working Directory

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

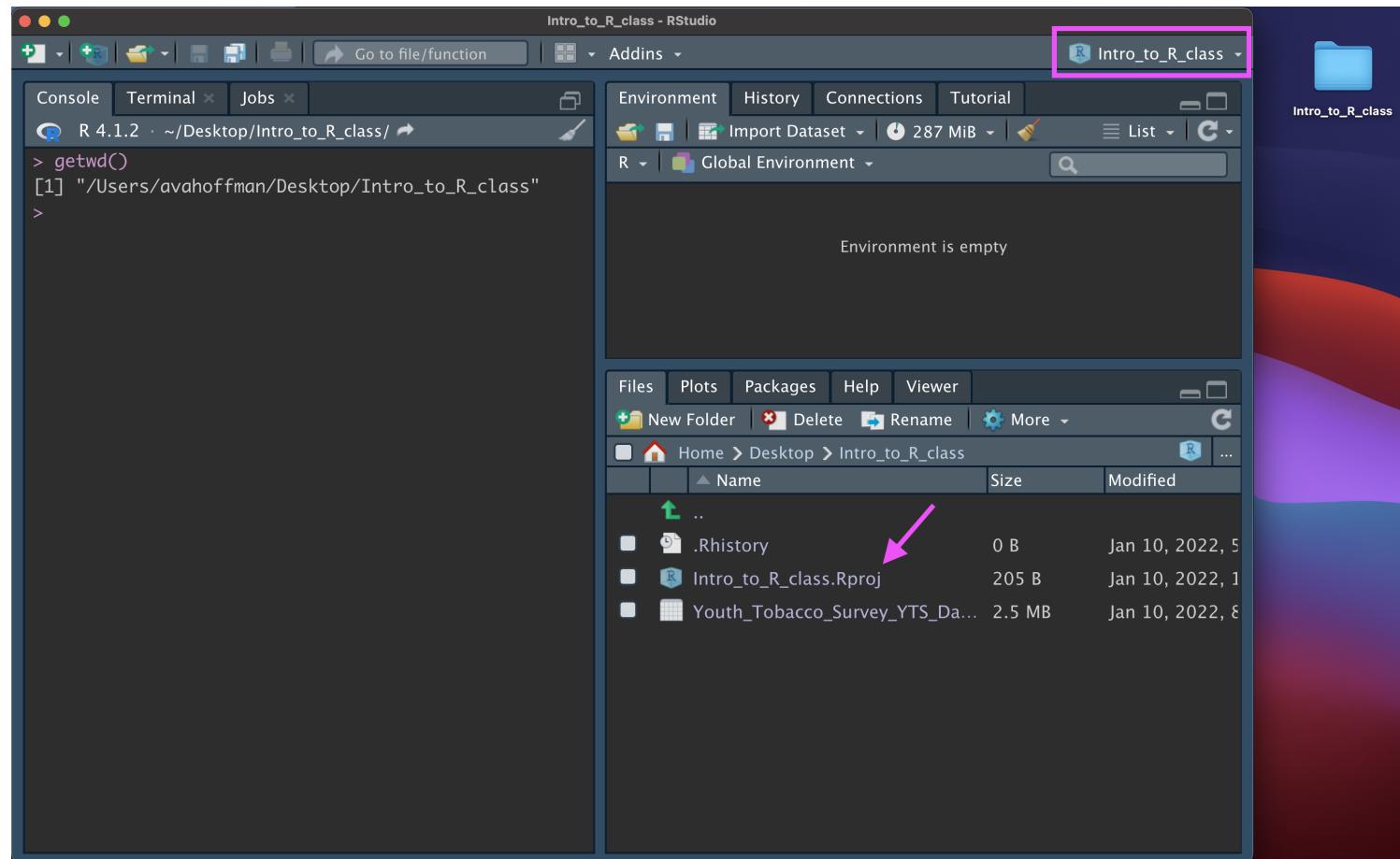


You can also run the `getwd()` function.

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

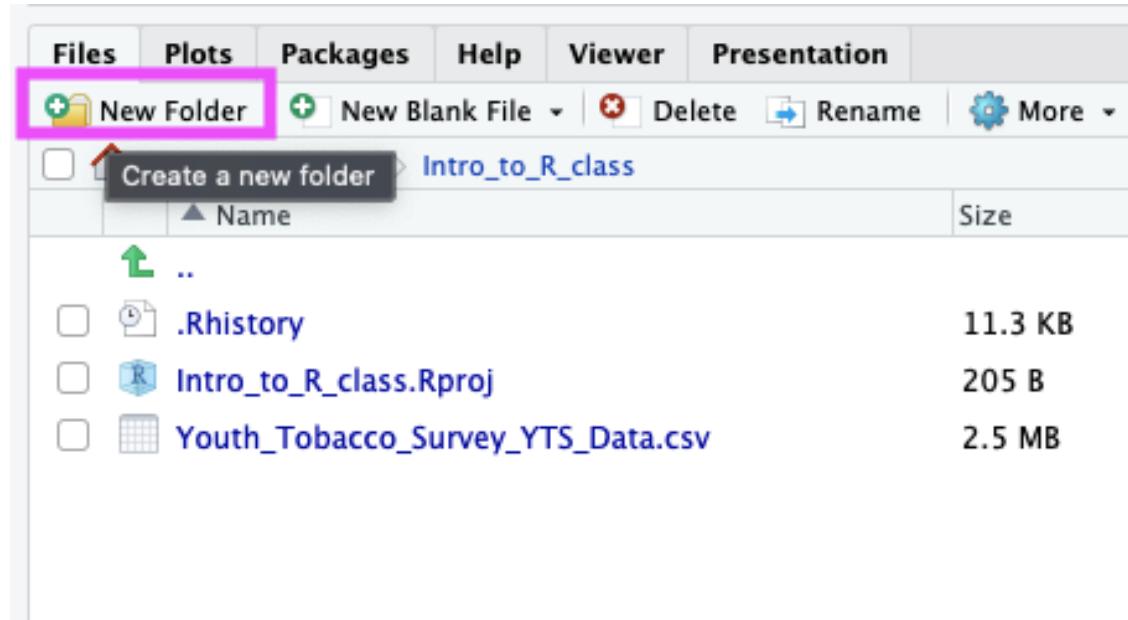
The Working Directory

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).

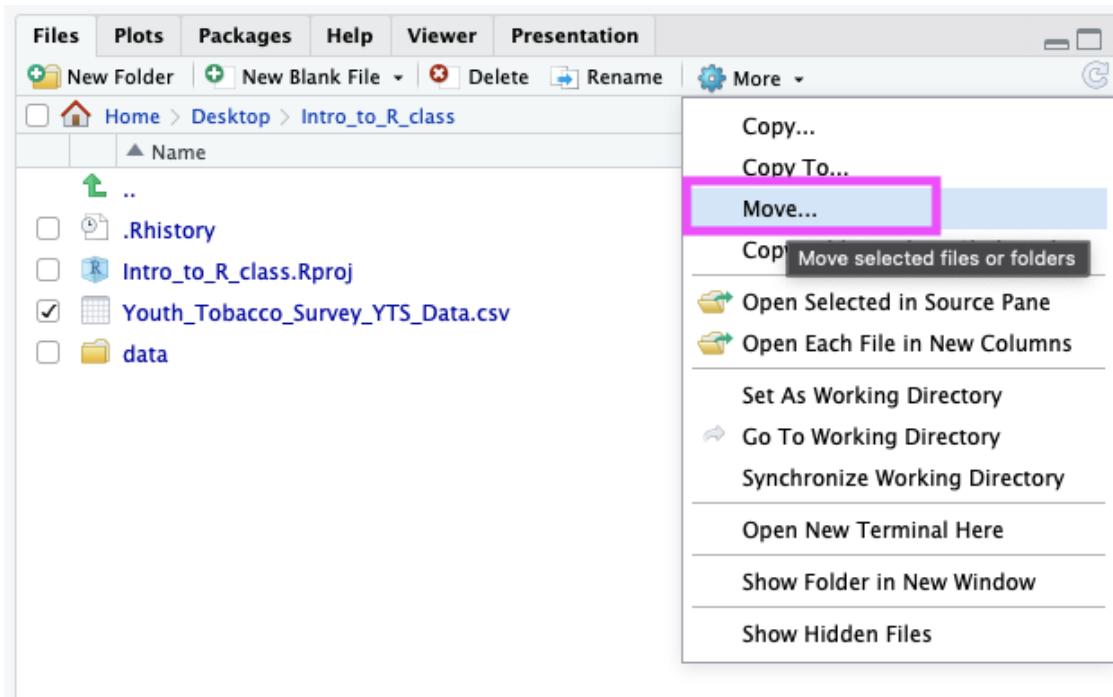


The screenshot shows the RStudio interface with the 'Files' tab selected in the top navigation bar. A pink box highlights the 'New Folder' button in the toolbar. A tooltip 'Create a new folder' is visible over the button. The main pane displays a file list with the following entries:

Name	Size
..	
.Rhistory	11.3 KB
Intro_to_R_class.Rproj	205 B
Youth_Tobacco_Survey_YTS_Data.csv	2.5 MB

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)
```

The screenshot shows the RStudio interface with the following components:

- Top Bar:** Intro_to_R_class - RStudio
- Left Panel:** A data viewer panel titled "dat" showing a subset of the data. The columns are: YEAR, LocationAbbr, LocationDesc, TopicType, TopicDesc, and Measure. Rows 1 through 12 are displayed, all corresponding to the year 2015 and location AZ.
- Right Panel:** An "Environment" tab showing the global environment. It lists "dat" as 9794 obs. of 31 variables.
- Bottom Left:** A "Console" tab showing the command `> View(dat)` and its execution.
- Bottom Right:** A "Files" tab showing the file structure: Home > Desktop > Intro_to_R_class > data. A file named "Youth_Tobacco_Survey_YTS_Da..." is listed with a size of 2.5 MB.

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" "Perce...
$ 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 o...
$ 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)" "(34.865970280000454, -111.76381127699972)"
```

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:

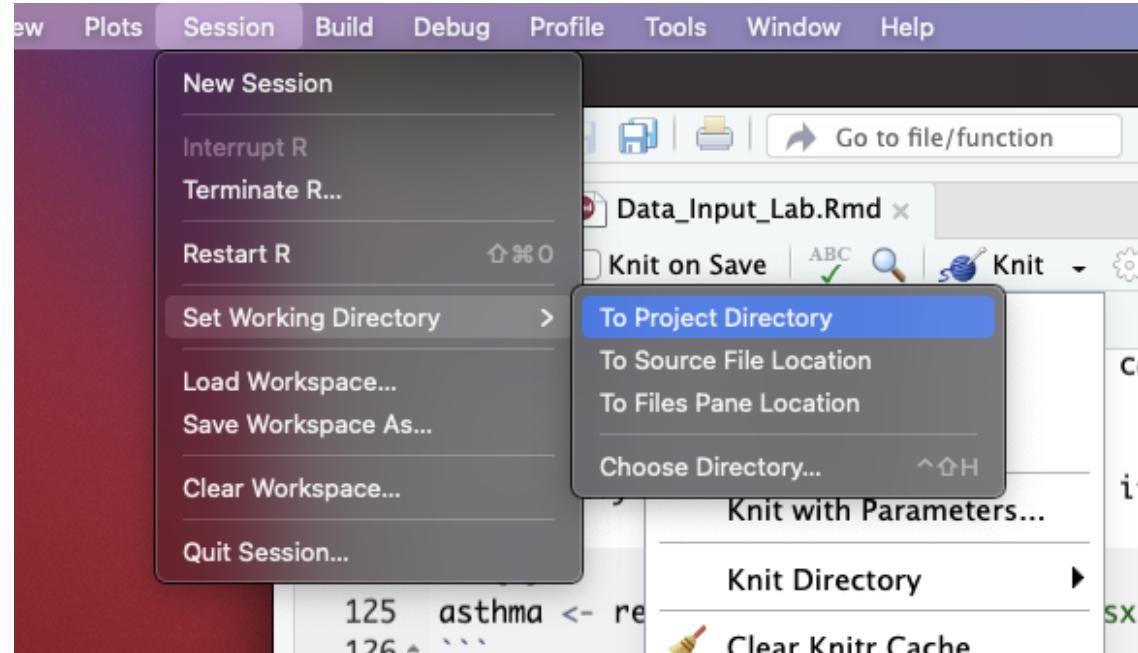
```
?read_delim  
help("read_delim")
```

The screenshot shows the RStudio interface with the help viewer open. The console tab has the command `?read_delim` entered. The help viewer shows the documentation for `read_delim`. The title is "R: Read a delimited file (including csv & tsv) into a tibble". The description states that `read_csv()` and `read_tsv()` are special cases of the general `read_delim()`. It explains that they're useful for reading the most common types of flat file data, comma separated values and tab separated values, respectively. The usage section shows the function signature: `read_delim(file, delim, ...)`.

TROUBLESHOOTING: Setting the working directory

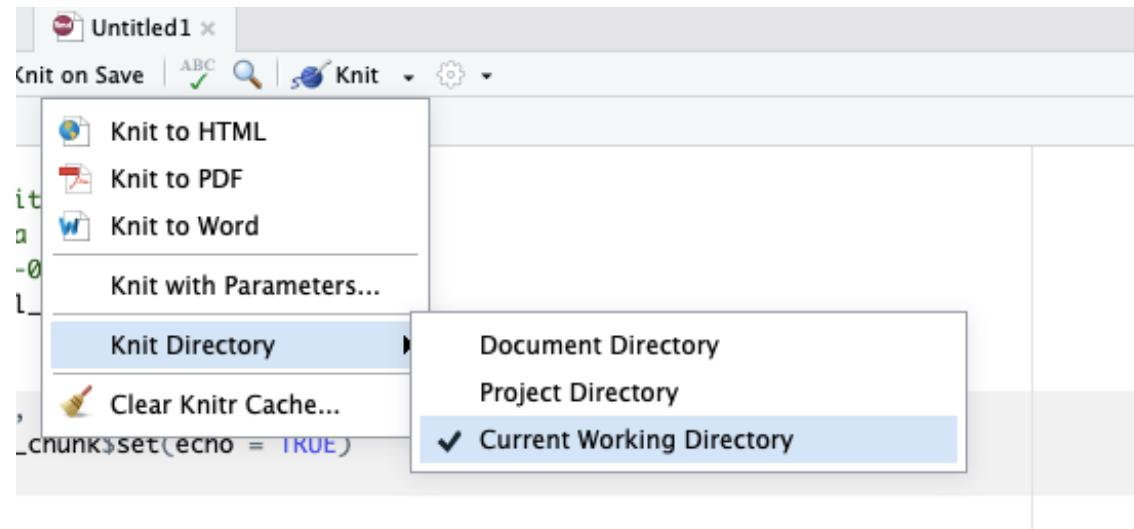
If your R project directory and working directory do not match:

- 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](#)