Data Input

Outline

- Part 0: A little bit of set up!
- Part 1: reading in manually (point and click)
- Part 2: reading in directly, reading XLSX file (Excel file), other data inputs
- Part 3: working directories, relative vs. absolute paths

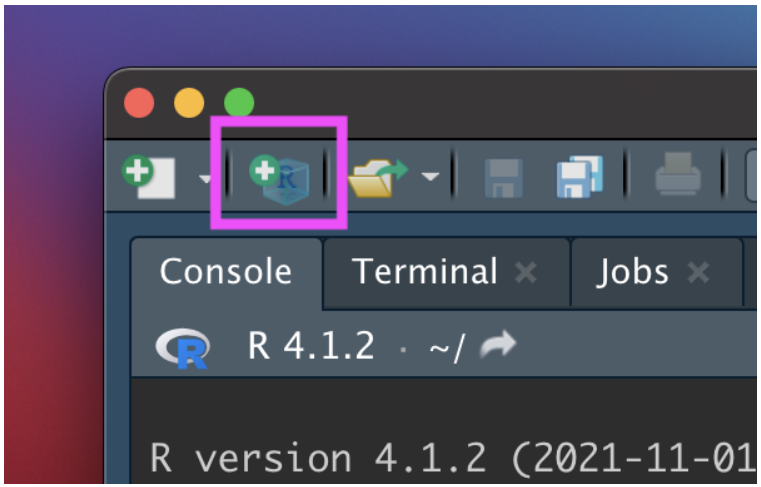
We will cover Output a bit later!

Part 0: Setup - R Project

New R Project

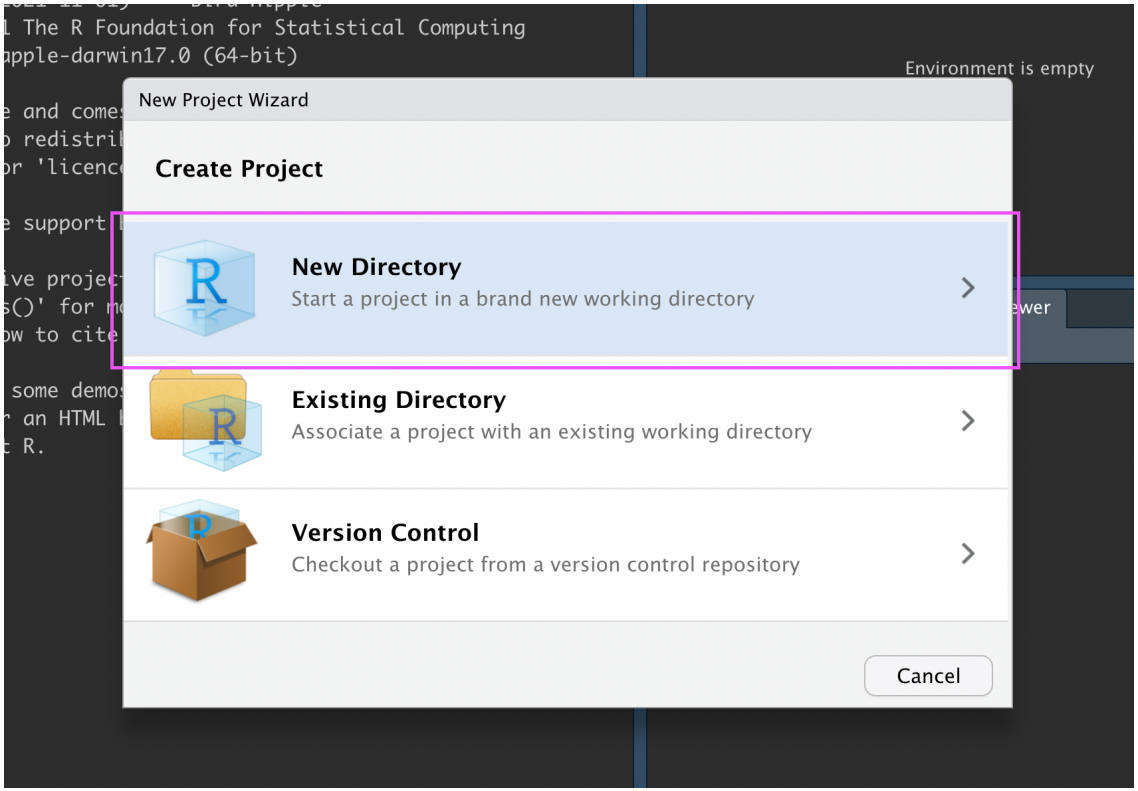
Let's make an R Project so we can stay organized in the next steps.

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



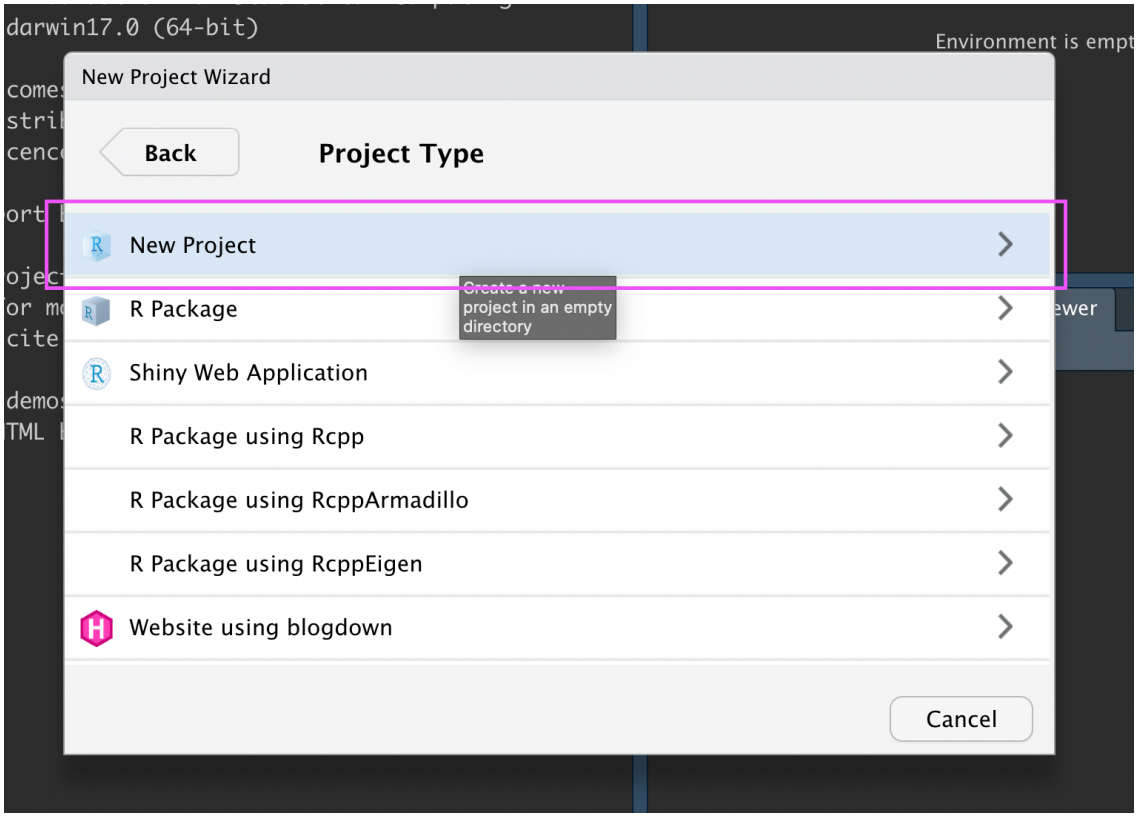
New R Project

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



New R Project

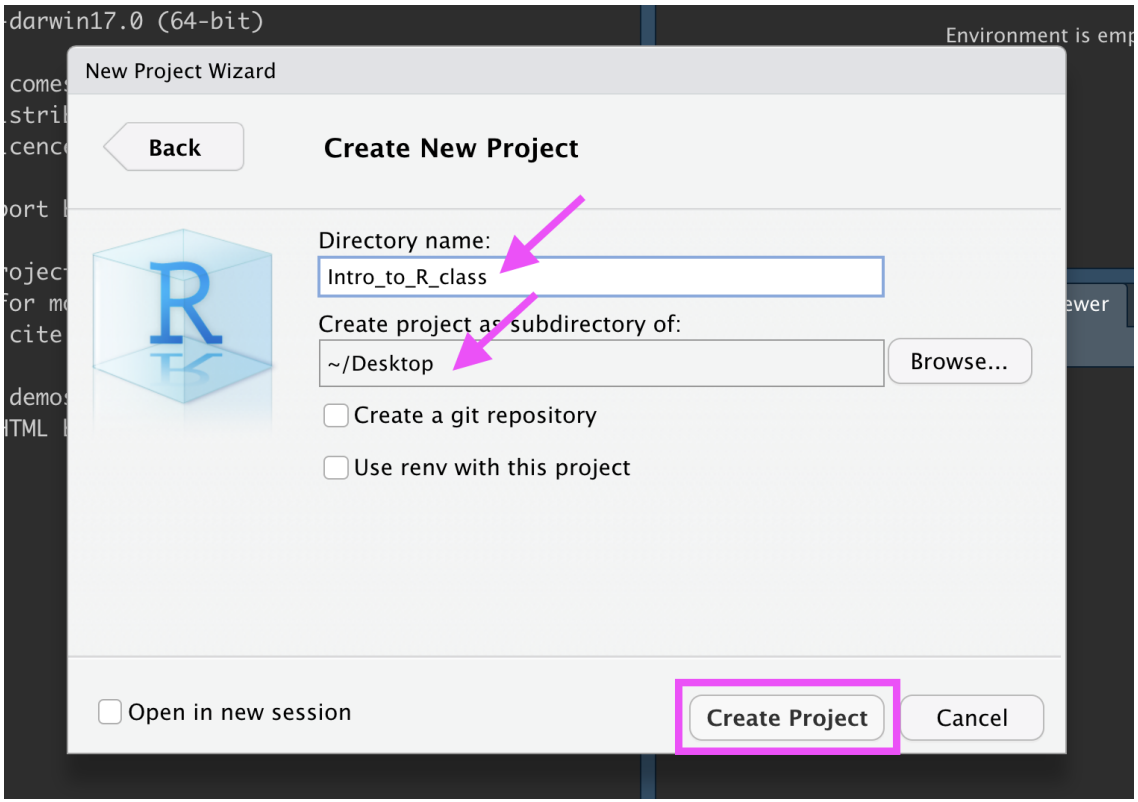
Click “New Project”:



New R Project

Type in a name for your new folder.

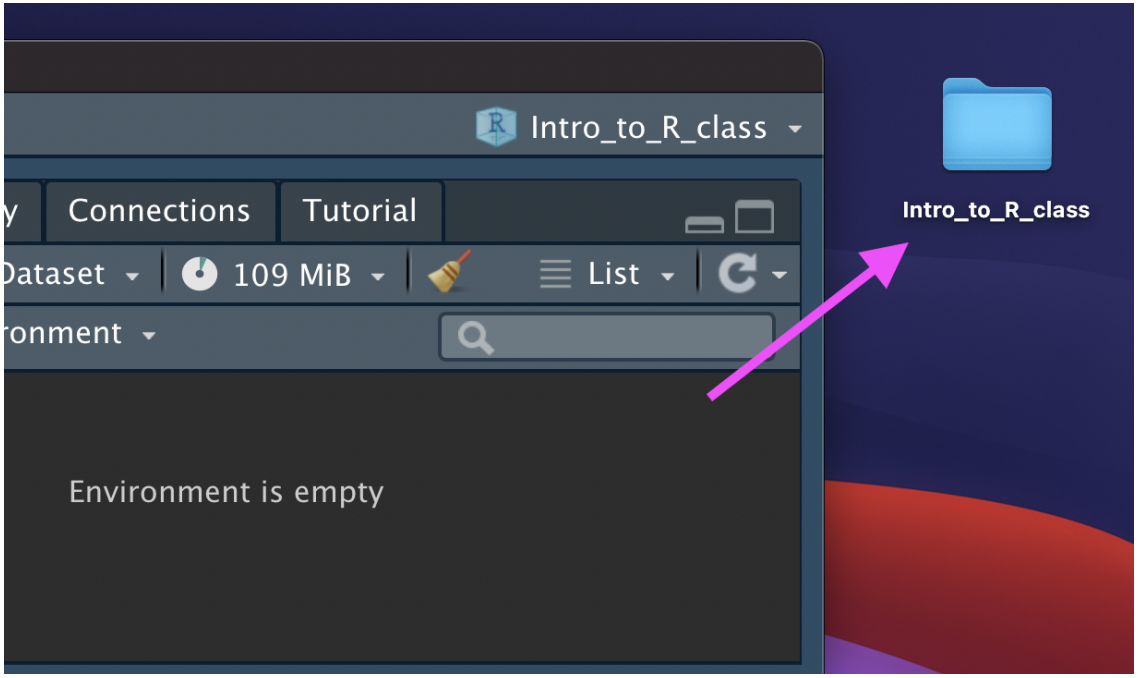
Store it somewhere easy to find, such as your Desktop:



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

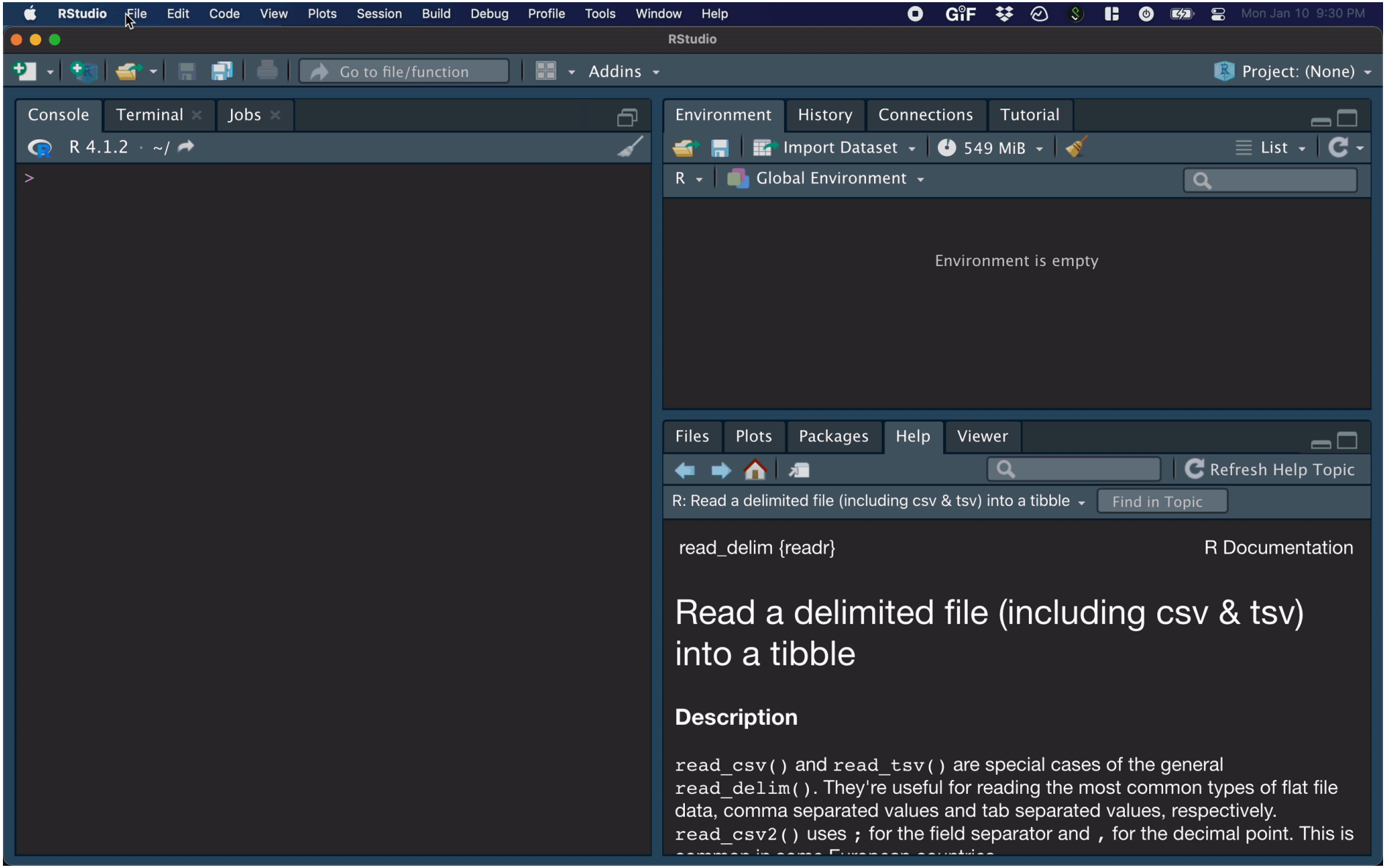
Dataset is located at
http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv

- 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

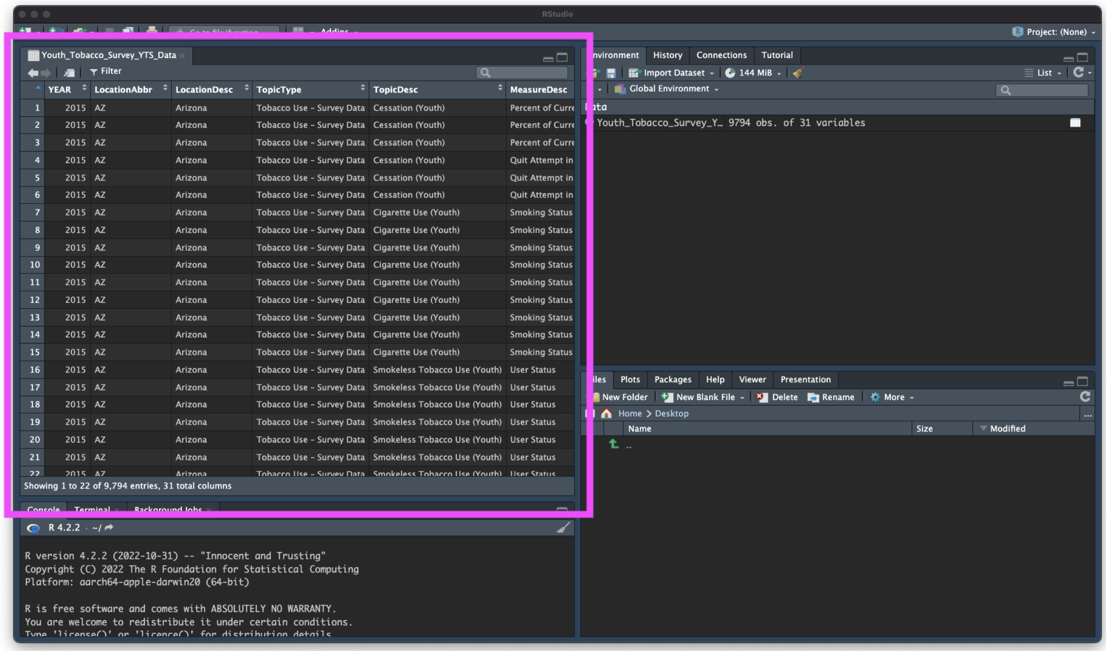
File Import Dataset From Text (`readr`) paste the url (http://jhudatascience.org/intro_to_r/data/Youth_Tobacco_Survey_YTS_Data.csv) click “Update” and “Import”

Import Dataset



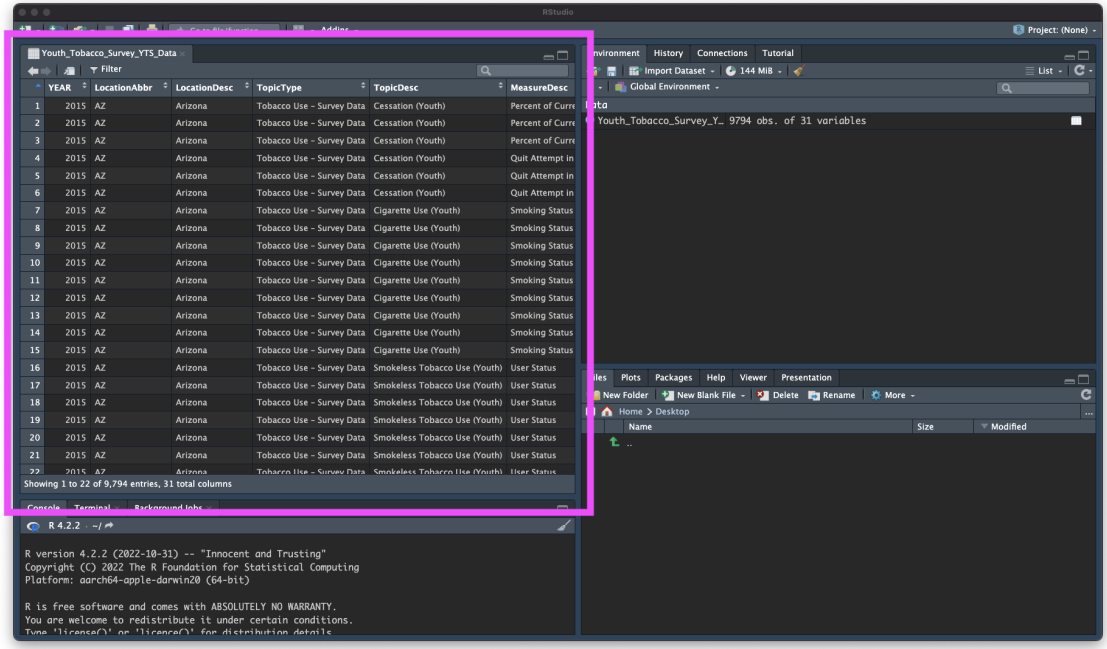
What Just Happened?

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



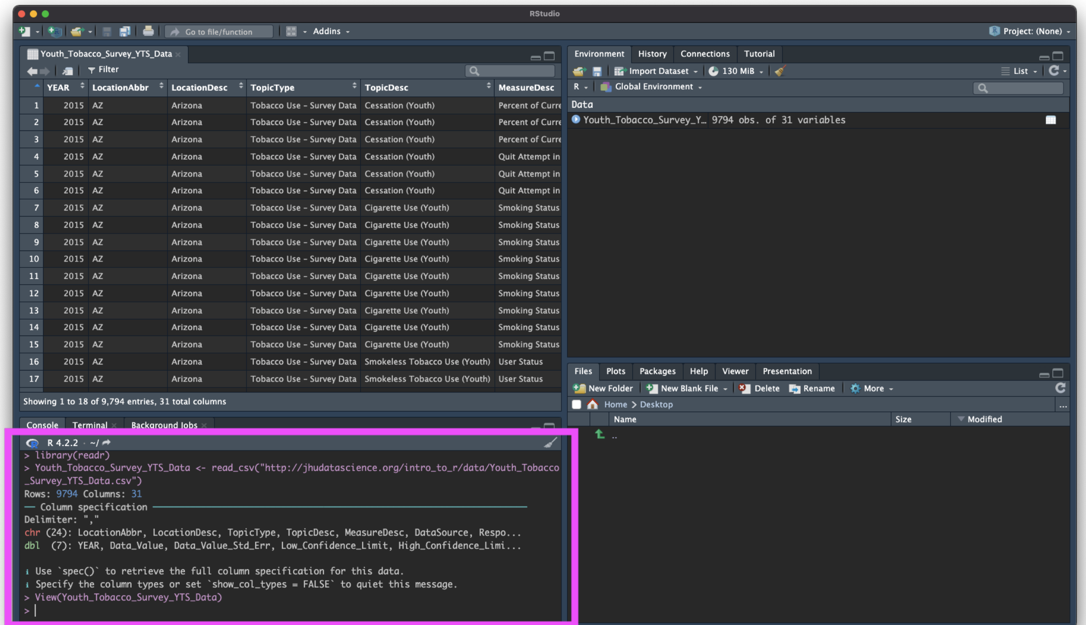
What Just Happened?

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.

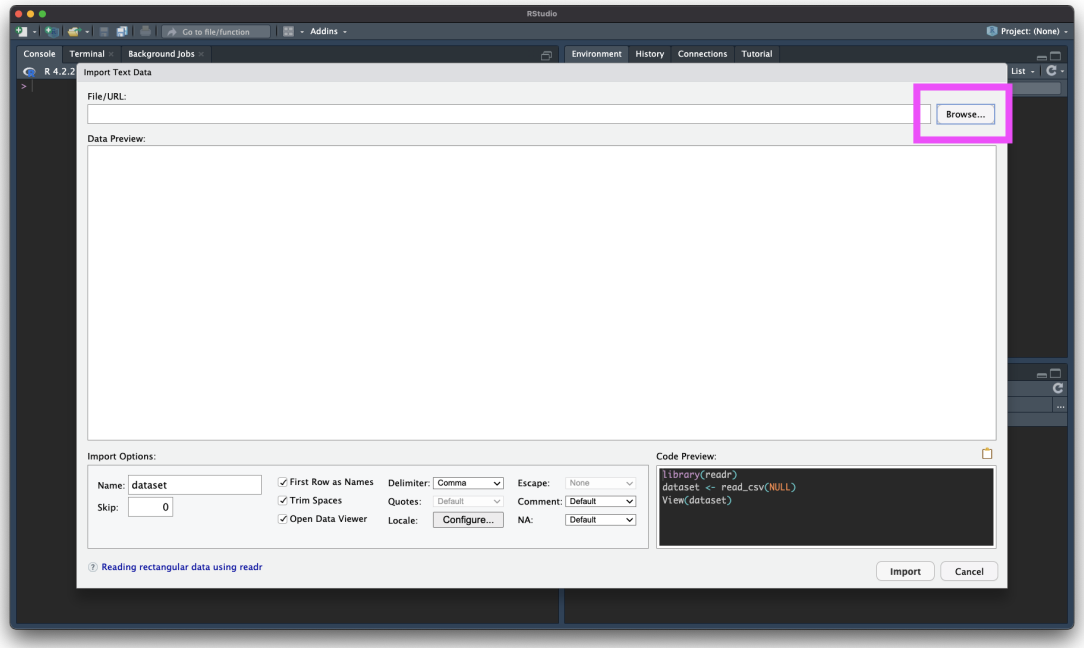


What Just Happened?

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 - 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”

Lab Part 1

- Class Website
- Data I/O 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

# `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
# 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 the path to your file. It will return a tibble

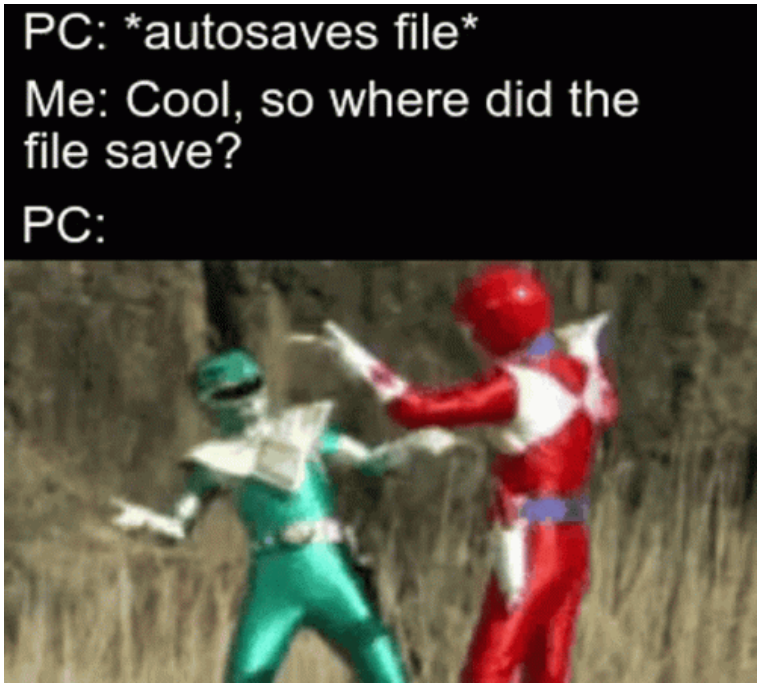
```
read_csv(file, col_names = TRUE, col_types = NULL,
  locale = default_locale(), na = c("", "NA"),
  quoted_na = TRUE, quote = "\"", comment = "", trim_ws = TRUE,
  skip = 0, n_max = Inf, guess_max = min(1000, n_max),
  progress = show_progress(), skip_empty_rows = TRUE
)
```

- `file` is the path to your file, in quotes
- can be path in your local computer – absolute file path or relative file path
- can be path to a file on a website

```
## 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")
```

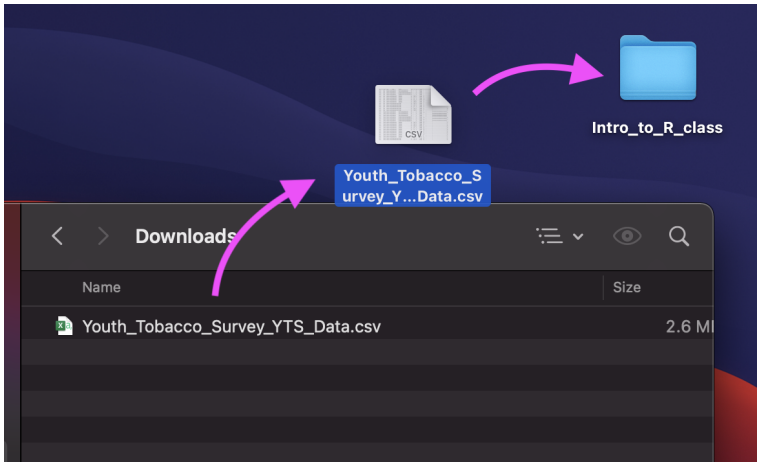
Data Input: Read in Directly

Great, but what is my “path”?



Data Input: Read in Directly

Luckily, we already set up an R Project!



If we add the Youth_Tobacco_Survey_YTS_Data.csv file to the intro_to_r folder, we can use the relative path:

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

Data Input: Read in Directly

`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

```
read_delim(file, delim, quote = "\"", escape_backslash = FALSE,
  escape_double = TRUE, col_names = TRUE,  col_types = NULL,
  locale = default_locale(),na = c("", "NA"),  quoted_na = TRUE,
  comment = "", trim_ws = FALSE, skip = 0,
  n_max = Inf,  guess_max = min(1000, n_max),
  progress = show_progress(), skip_empty_rows = TRUE
)
```

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

```
## Examples
dat <- read_delim(file = "Youth_Tobacco_Survey_YTS_Data.csv", delim = ",")

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

Data Input: Read in Directly From File Path

Move the data to the data folder and change the relative path:

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

The data is now successfully read into your R environment. You can confirm this by checking the “Environment” pane (top right). Column specification of first few columns is printed to the console.

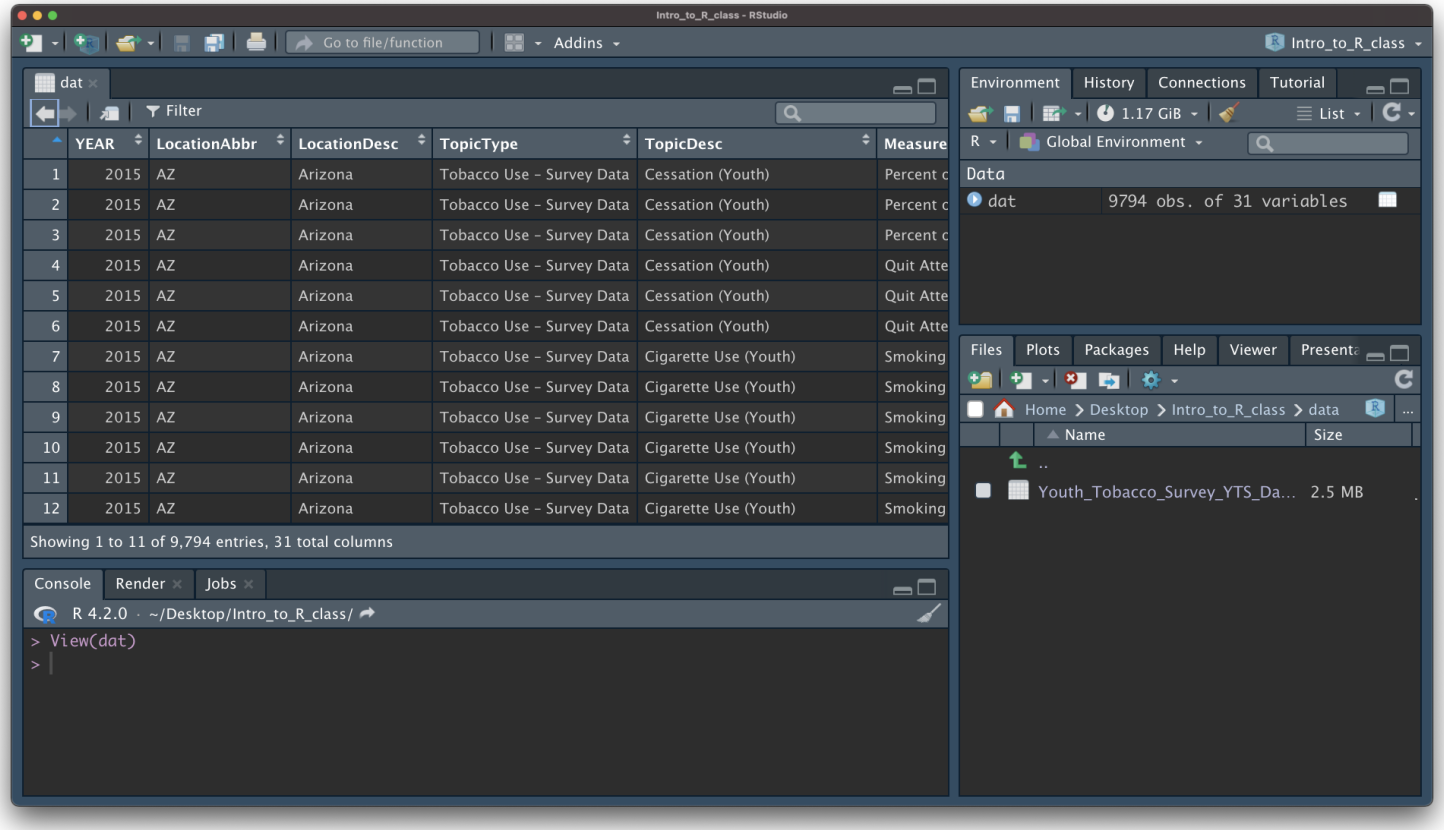
Common new user mistakes we have seen

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

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 shows you the structure of the data (different variables and their classes - more on this later).

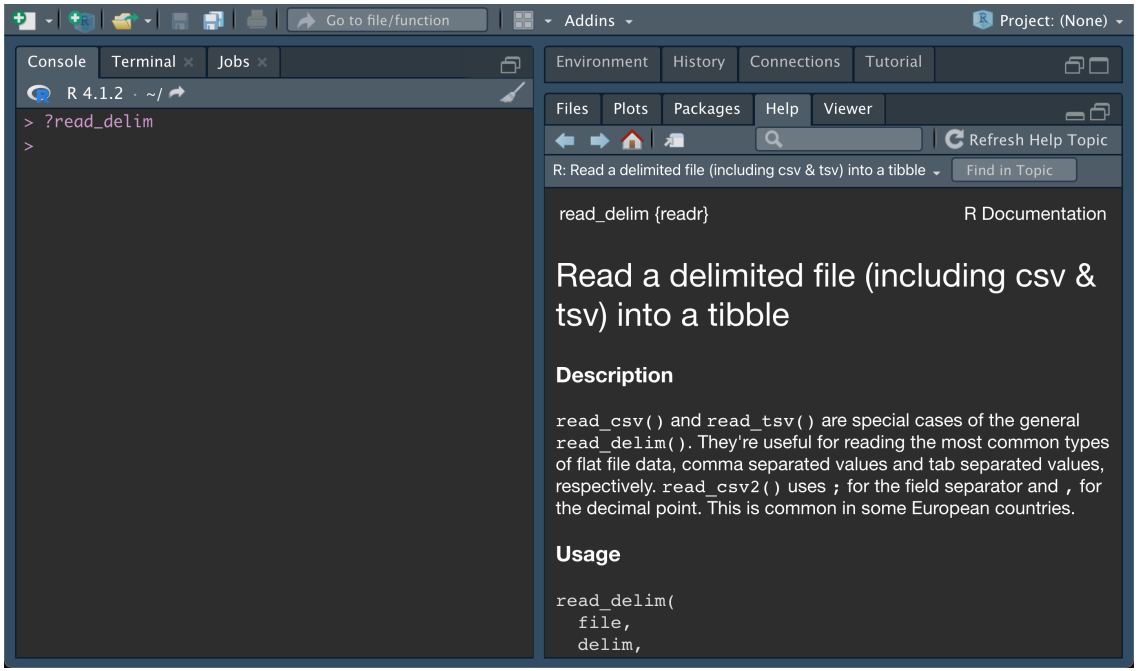
```
str(dat)
```

```
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 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)" "(34.86
 $ TopicTypeId                : chr [1:9794] "BEH" "BEH" "BEH" "BEH" ...
```

Help

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

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



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.

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")
```

Summary: `readr` highlights - Part 2

- Modern, improved tools from `readr` R package: `read_delim()`, `read_csv()`
  - needs a file path to be provided
  - parses the file into rows/columns, determines column type
  - returns a tibble (data frame)
- Some functions to look at a data frame:
  - `head()` shows first few rows
  - `tail()` shows the last few rows
  - `View()` shows the data as a spreadsheet
  - `str()` tells you about column types

Summary: other file types - Part 2

- From `readr` package:
  - `read_delim()`: general delimited files
  - `read_csv()`: comma separated (CSV) files
  - `read_tsv()`: tab separated files
  - others
- For reading Excel files, you can do one of:
  - use `read_excel()` function from `readxl` package
  - use other packages: `xlsx`, `openxlsx`

Lab Part 2

- Class Website
- Data I/O Lab

Working Directories

Working directory is a directory that R assumes "you are working in". It's where R looks for files.

"Setting working directory" means specifying the path to the directory.

```
# get the working directory
getwd()

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

R uses working directory as a starting place when searching for files.

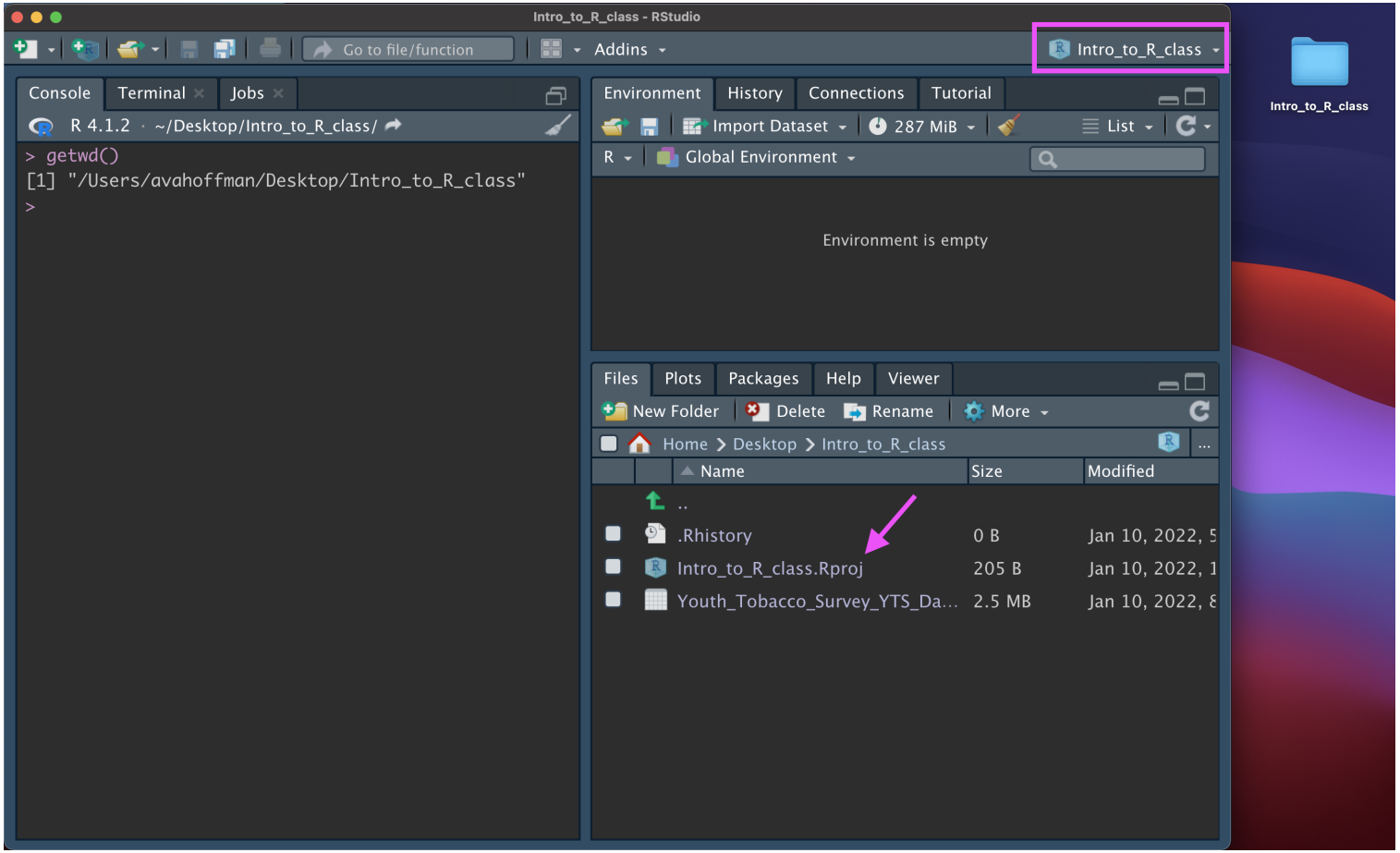
Working Directories

R uses working directory as a starting place when searching for files:

- if you use `read_csv("Bike_Lanes_Long.csv")`, R assumes that the file is **in** the working directory
- if you use `read_csv("data/Bike_Lanes_Long.csv")`, R assumes that `data` directory is **in** the working directory
- if you use an absolute path, e.g. `read_csv("/Users/avahoffman/data/Bike_Lanes_Long.csv")`, the working directory information is not used

Working Directories

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



Summary

- R Projects are a good way to keep your files organized and reduce headaches
- Use `read_csv()` and `read_delim()` from the `readr` package to read in your data
- Don't forget to use `<-` to assign your data to an object!
- Use `str()` to understand objects
- Use `head()` and `tail()` to preview the first and last lines of the data

- Class Website
- Data I/O Lab