Data Input

R Projects

R Projects are a feature of RStudio that can help you stay organized. They are pretty straightforward to set up, but are not required. You can learn more about R Projects here:

https://jhudatascience.org/intro_to_r/resources/R_Projects.html

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')

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."

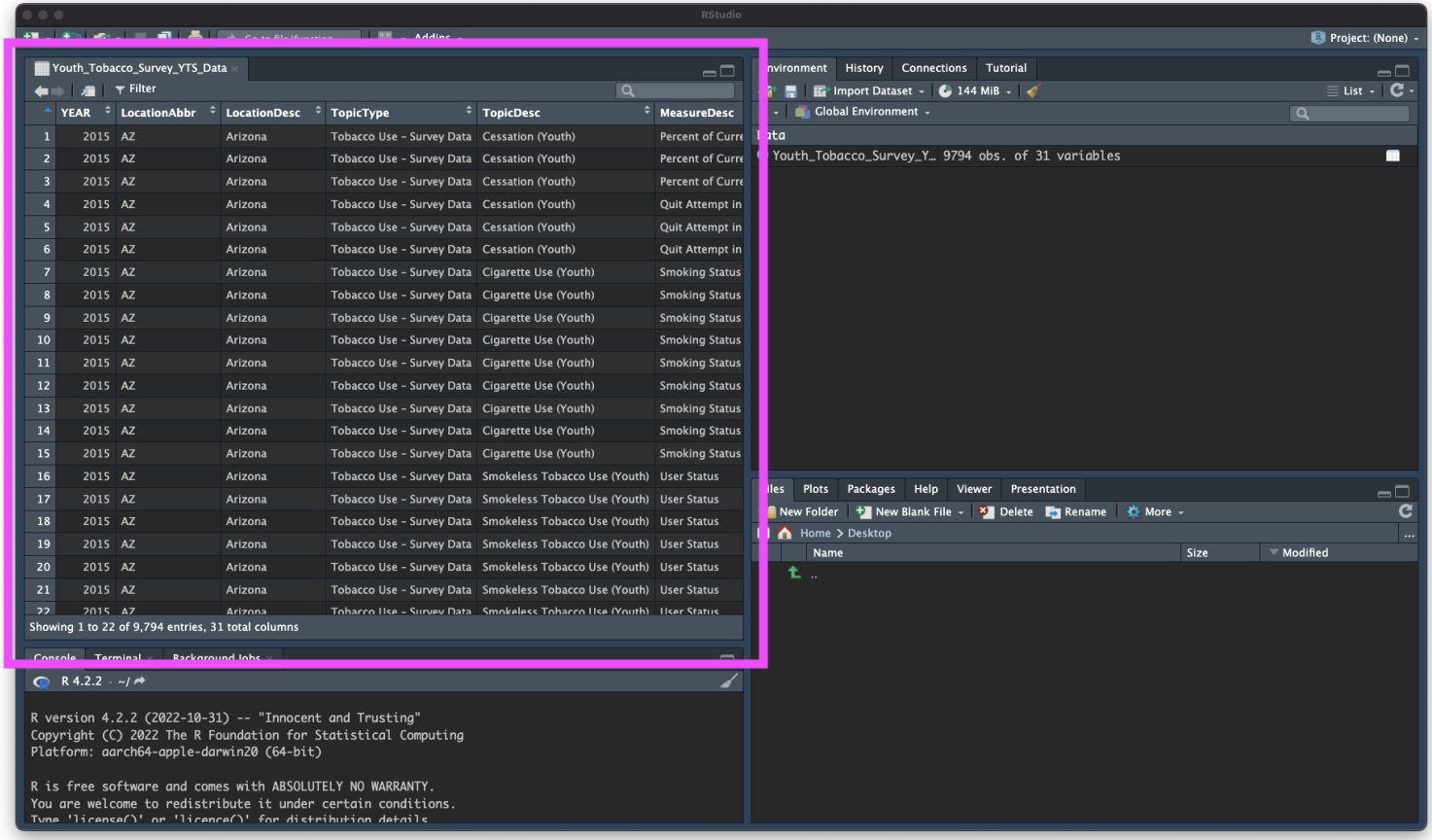
- Check out the data at: https://catalog.data.gov/dataset/youth-tobacco-survey-yts-data

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”

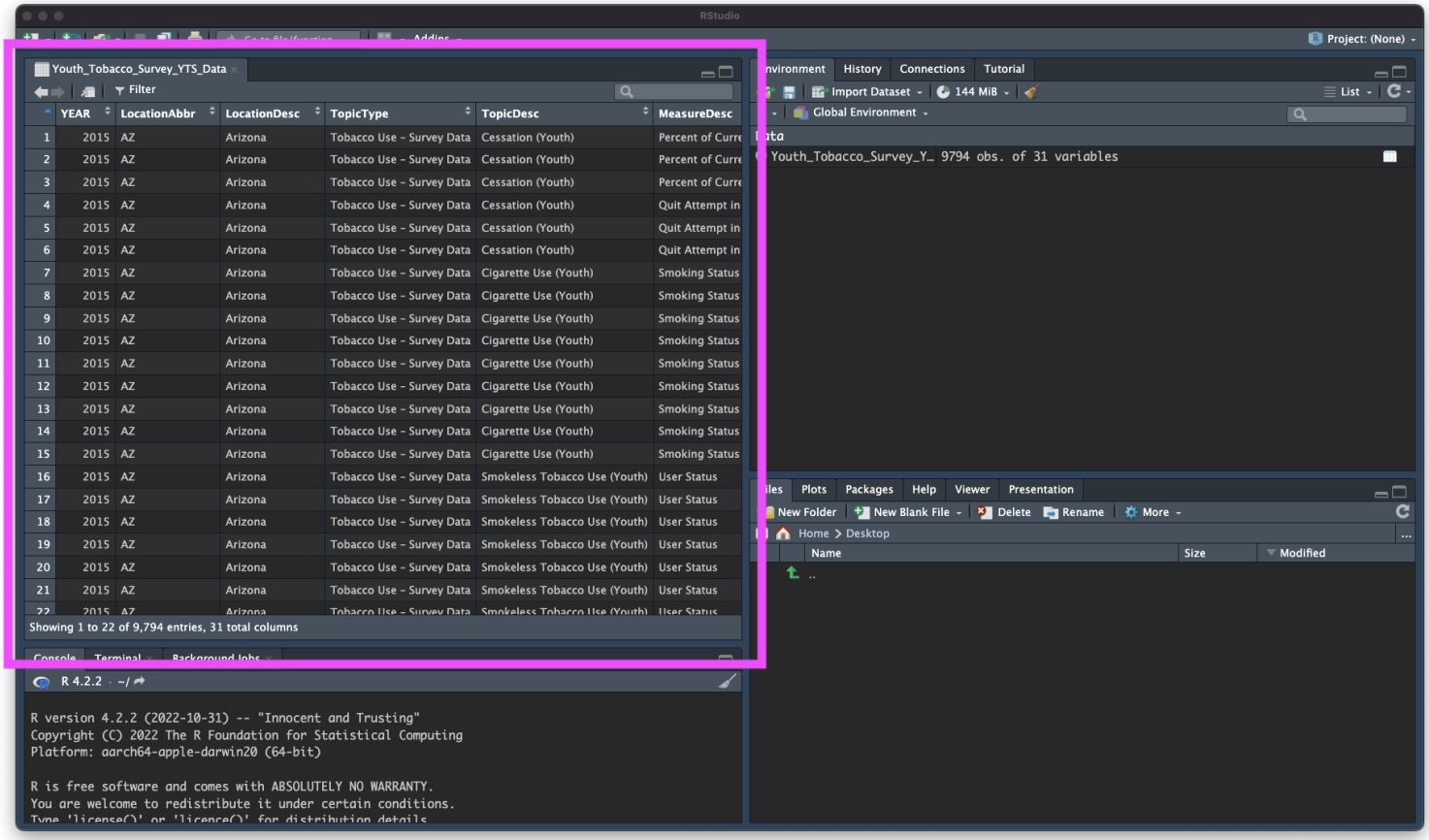
What Just Happened?

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



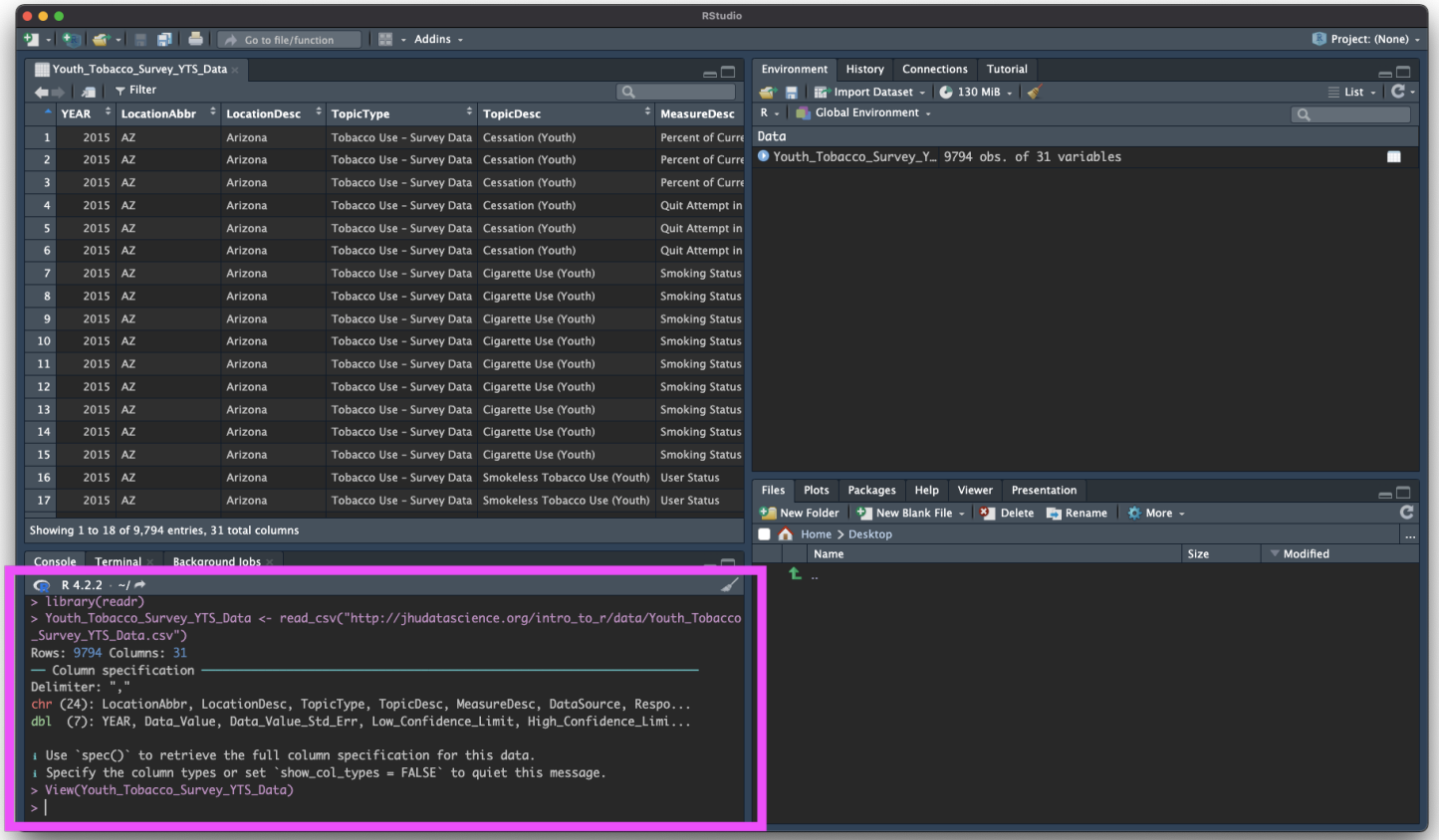
What Just Happened?

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.

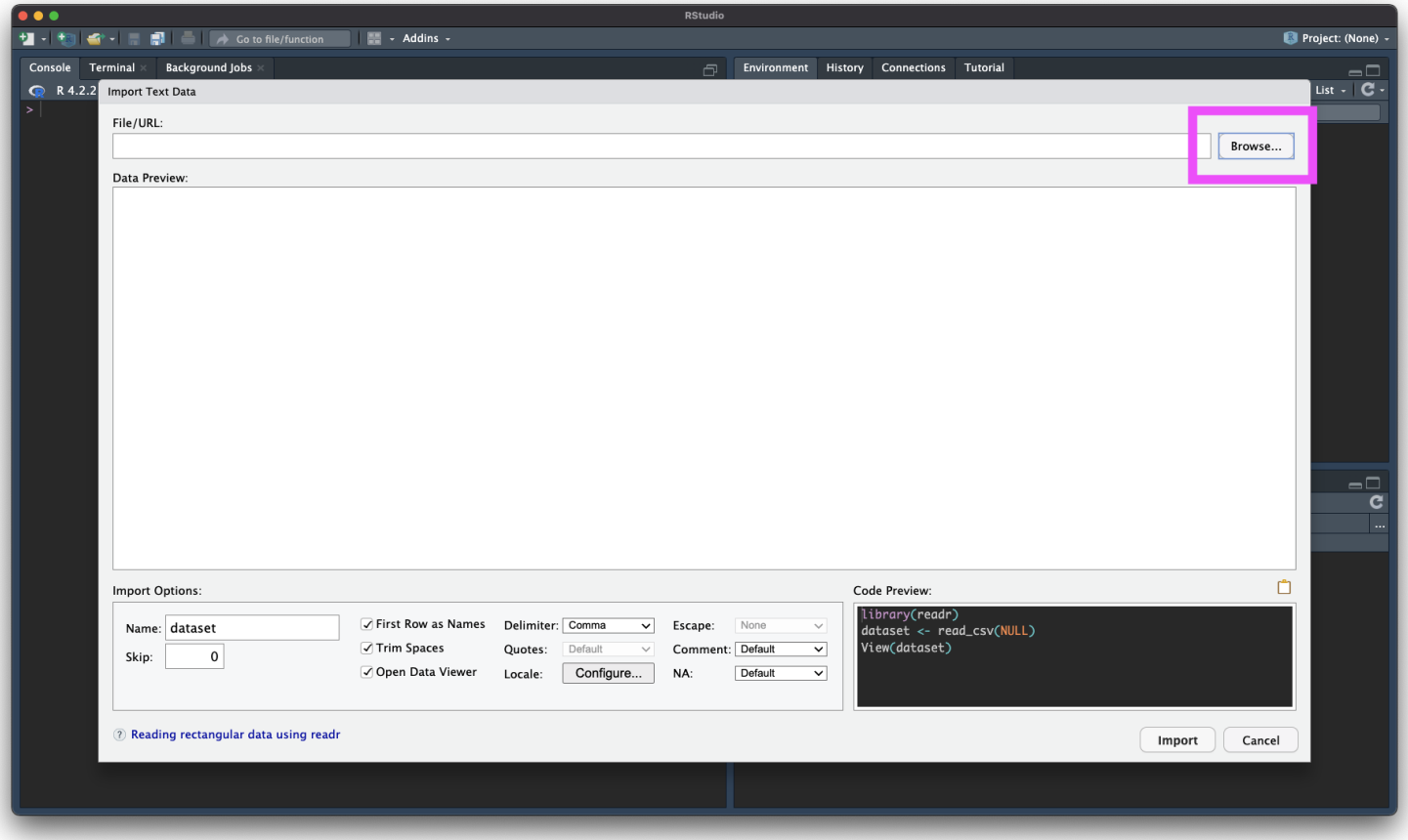


What Just Happened?

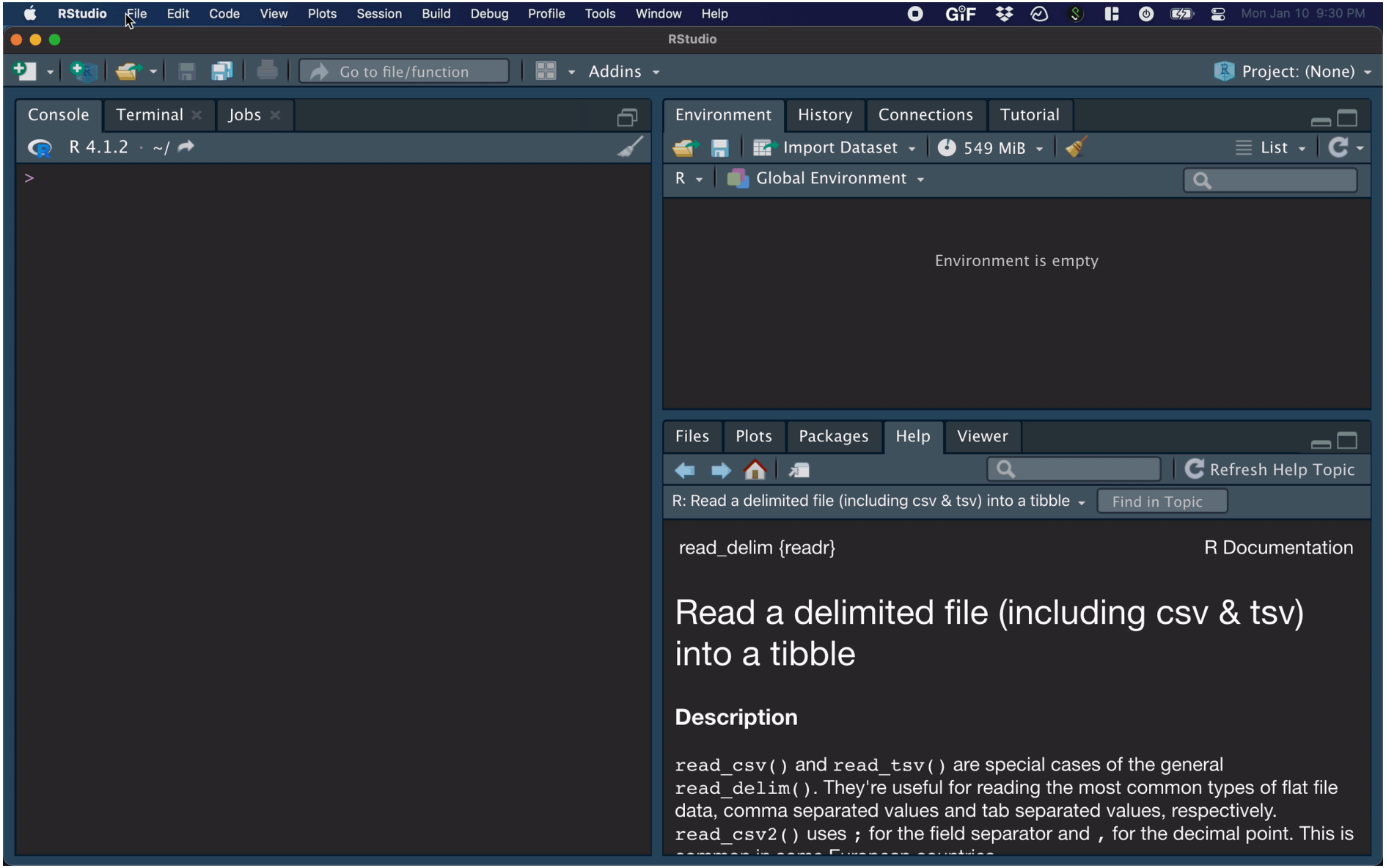
R ran some code in the console (bottom left).



Browsing for Data on Your Machine



Import Dataset



Manual Import: Pros and Cons

Pros: easy!!

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

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
# ℹ 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

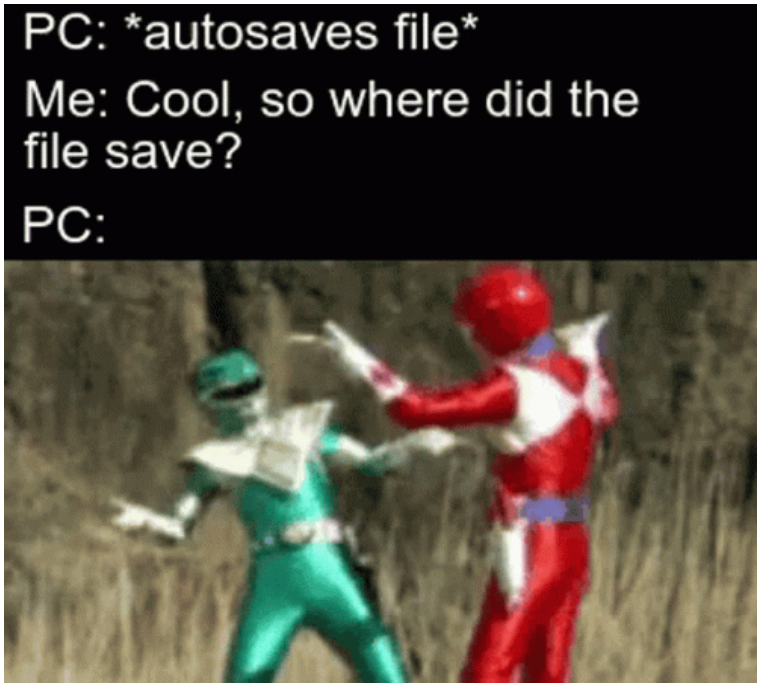
`read_csv()` needs an argument `file =`.

- `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 = "www.someurl.com/table1.csv")

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

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

Data Input: File paths

What is a file path ????

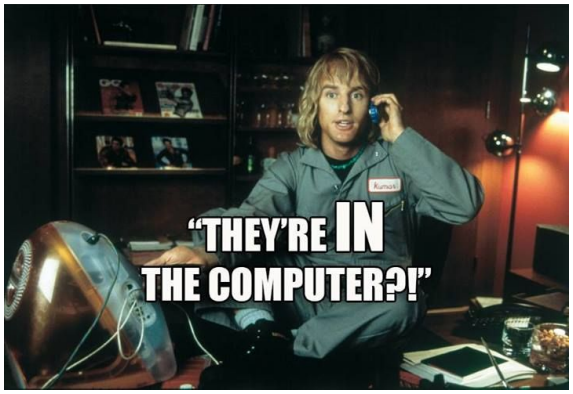


The working directory

When we work in R, we automatically have a **working directory**.

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

It’s where R looks for files.



Getting the working directory

Run the `getwd()` function to determine your working directory.

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

Relative path

Let's say my data is in a folder called "data" in my working directory.

`data/my_data.csv` would be the **relative path**. It's relative to the working directory.

The whole address, for example `/Users/avahoffman/Downloads/data/my_data.csv` is the **absolute path**.

Setting the working directory

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

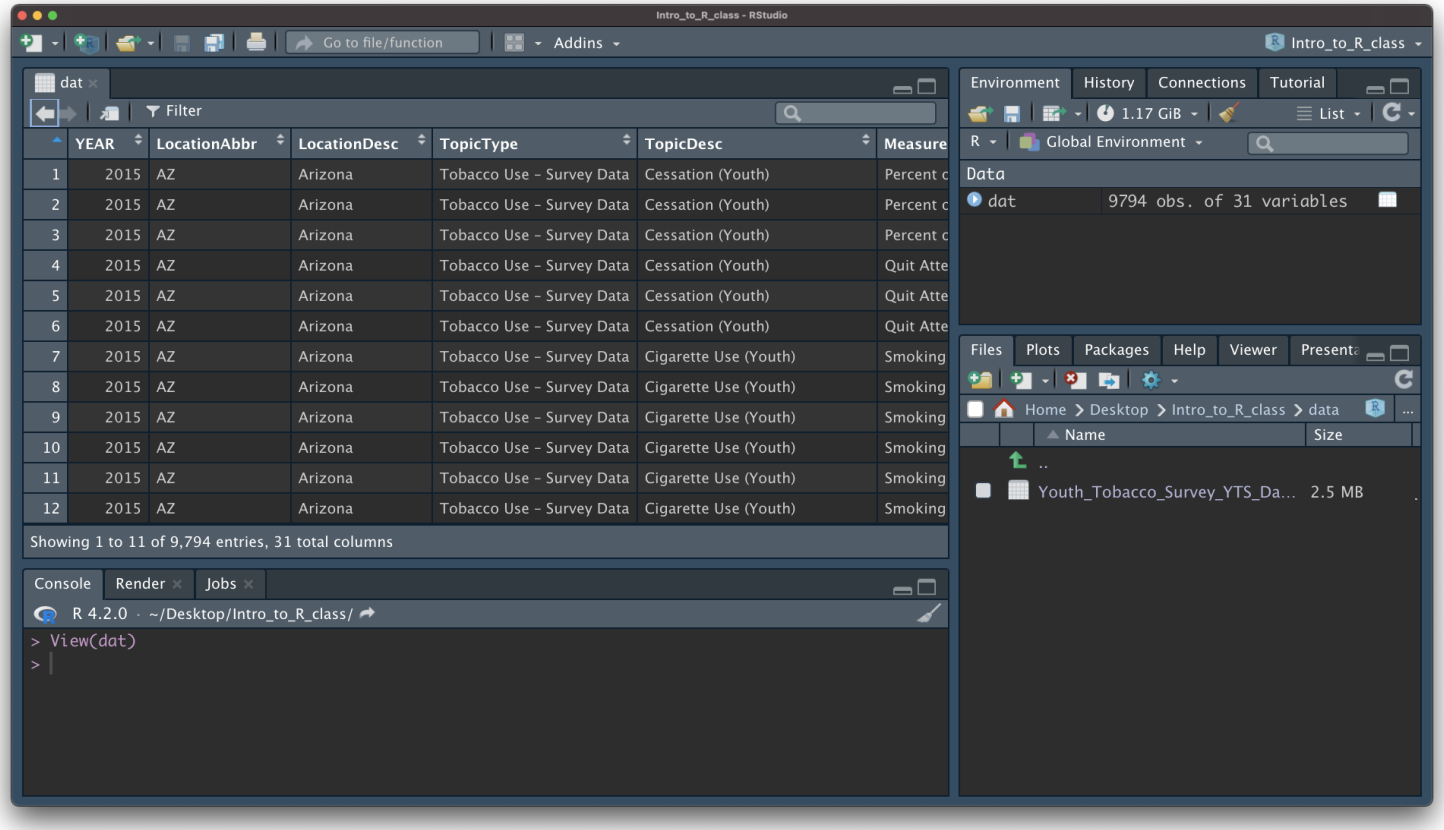
```
# set the working directory
setwd("/Users/avahoffman/Desktop")
```

Now what? 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: 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
- There are also resources for REDCap : `REDCapR`

WARNING! `read.csv` is * base R *

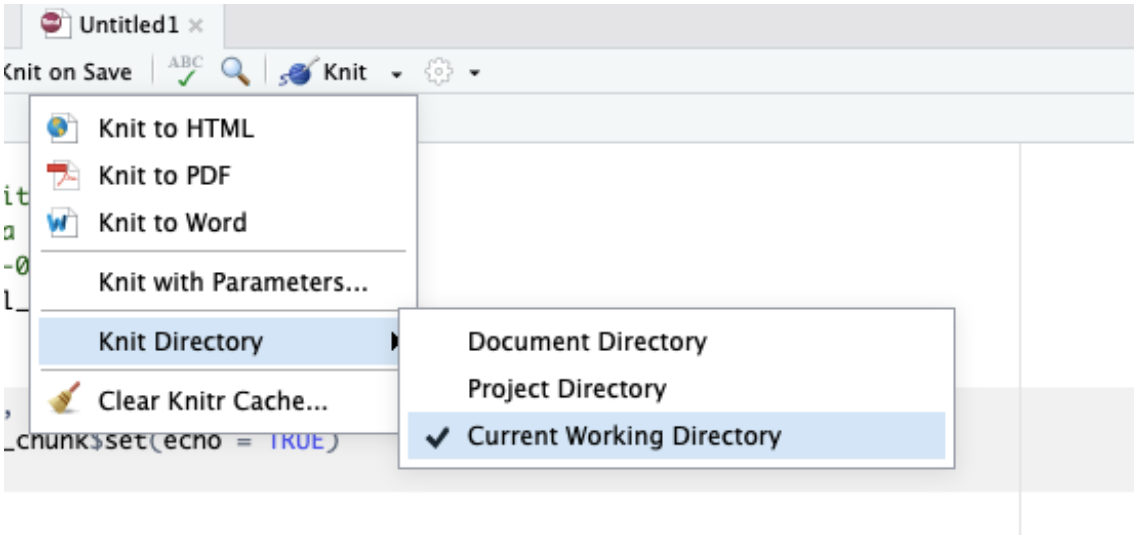
There are also data importing functions provided in base R (rather than the `readr` package), like `read.delim()` and `read.csv()`.

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: 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":



Other Useful Functions

- The `str()` function can tell you about data/objects.
- We will also discuss the `glimpse()` function later, which does something very similar.
- `head()` shows first few rows
- `tail()` shows the last few rows

Summary

R Projects can make it easier to find files. Check out this resource.

Importing data manually:

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

Importing data programmatically:

- `read_csv()` function from `readr` package
- Use `getwd()` to check your working directory, where R looks for your data files

Summary - Part 2

Look at your data!

- Check the environment for a data object
- `View()` gives you a preview of the data in a new tab

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

- Class Website
- Data Input Lab

Image by Gerd Altmann from Pixabay