

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, reading XLSX file (Excel file), other data inputs
- Part 3: working directories, relative vs. absolute paths

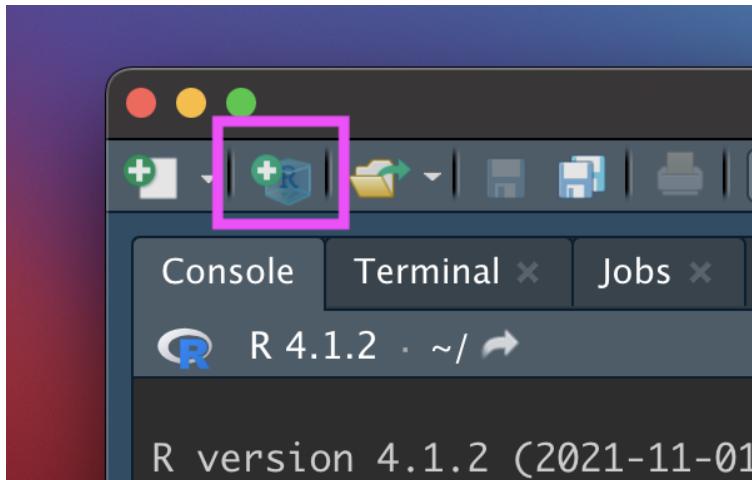
We will cover Output a bit later!

Part 0: Setup - R Project

New R Project

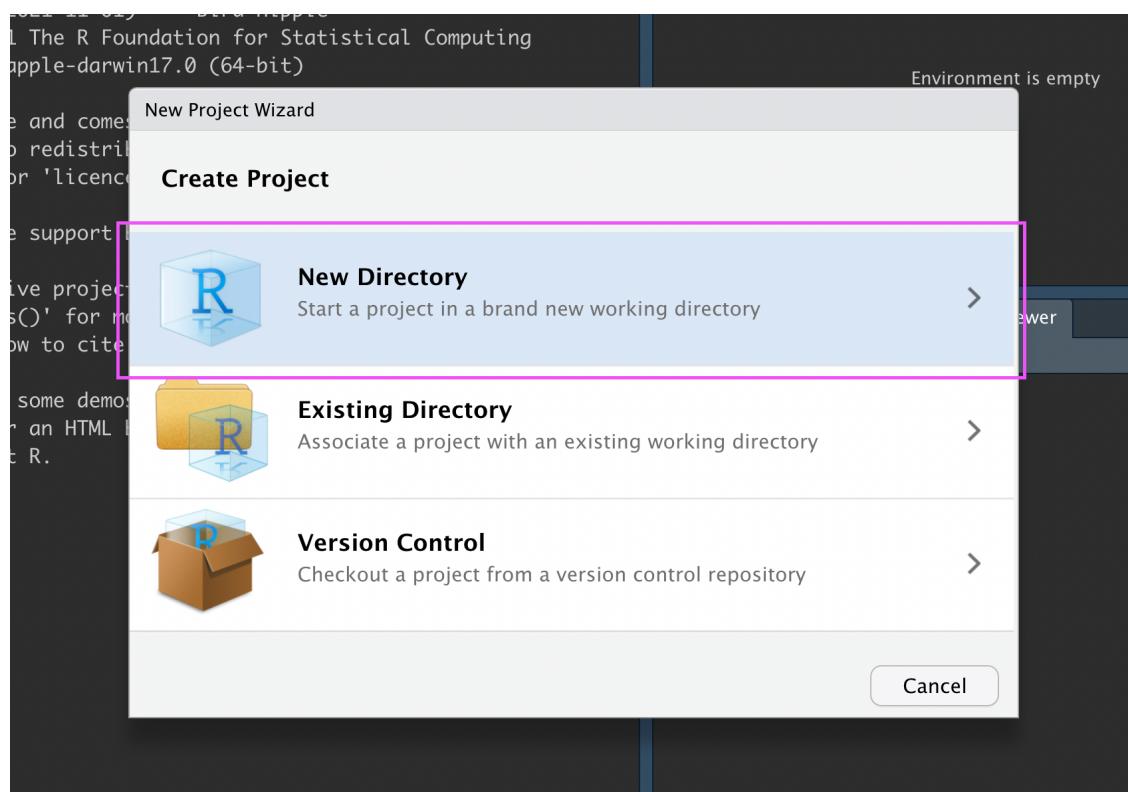
Let's make an R Project so we can stay organized in the next steps.

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



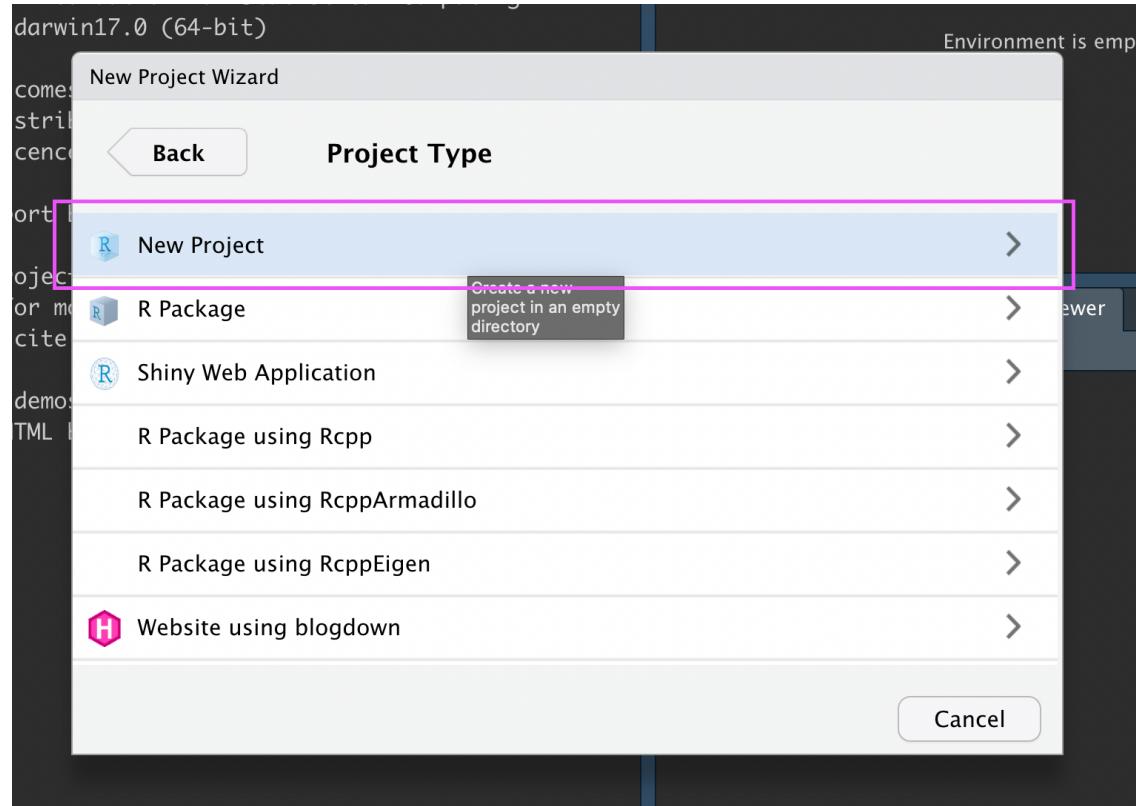
New R Project

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



New R Project

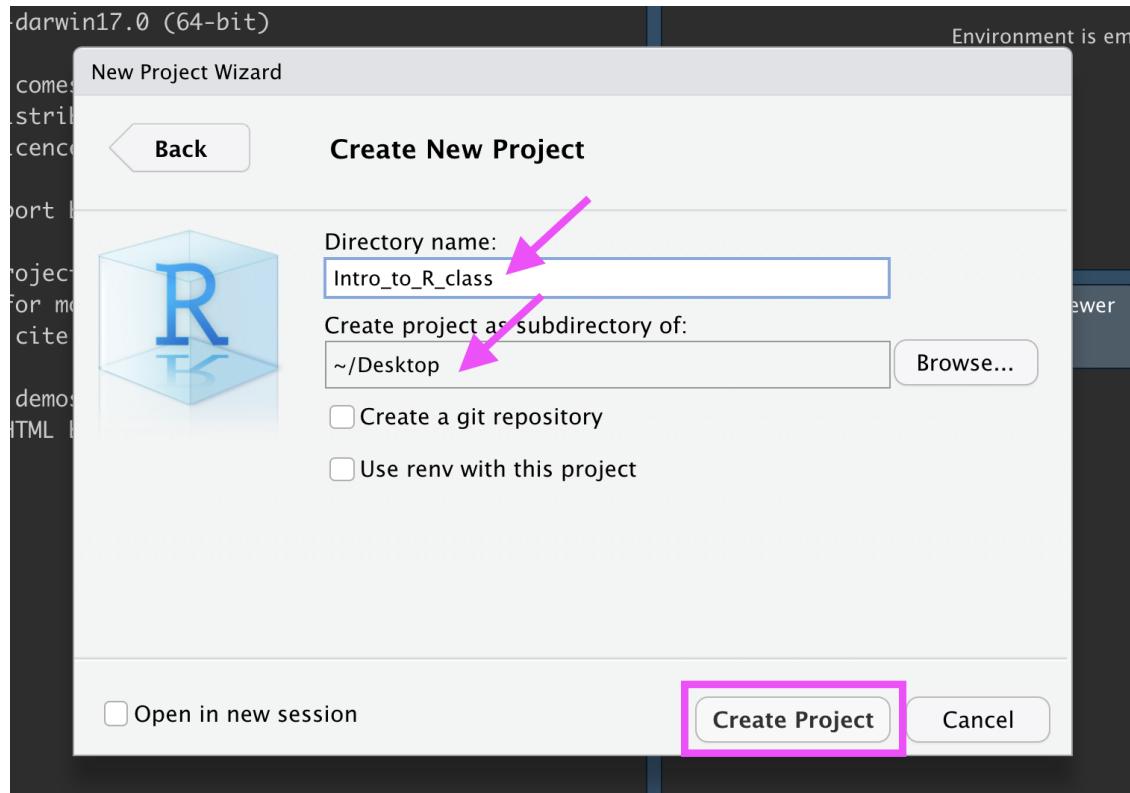
Click “New Project”:



New R Project

Type in a name for your new folder.

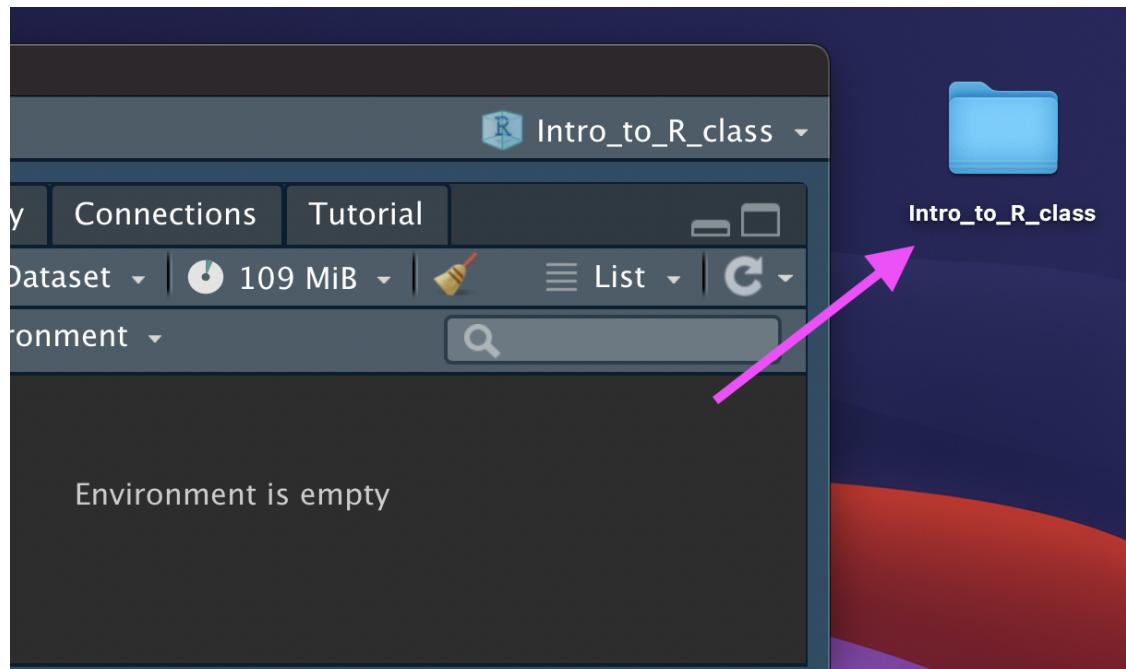
Store it somewhere easy to find, such as your Desktop:



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://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.

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Window Help

Mon Jan 10 9:30 PM

Project: (None)

Console Terminal × Jobs ×

R 4.1.2 ~/ ↗

>

Environment History Connections Tutorial

Import Dataset 549 MiB

Global Environment

Environment is empty

Files Plots Packages Help Viewer

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

read_delim {readr}

R Documentation

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.

The screenshot shows the RStudio interface. The top-left pane displays a data preview of the 'Youth_Tobacco_Survey_YTS_Data' dataset. The columns listed are YEAR, LocationAbbr, LocationDesc, TopicType, TopicDesc, and MeasureDesc. The data shows rows from 1 to 22 of 9,794 entries, with 31 total columns. The bottom-left pane shows the R console output, which includes the R version information (R 4.2.2), copyright notice, and a note about the software being free. The right-hand sidebar shows the file structure under 'Project: (None)'. The top menu bar includes Environment, History, Connections, Tutorial, and Project: (None).

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. The Environment pane on the right displays the `Youth_Tobacco_Survey_YTS_Data` object, which contains 9794 observations and 31 variables. The Data pane below it shows a preview of the first few rows of the dataset. The left pane shows the R console output, which includes the R version information and the command used to load the data. A pink rectangle highlights the Data pane where the dataset is previewed.

YEAR	LocationAbbr	LocationDesc	TopicType	TopicDesc	MeasureDesc
1	2015 AZ	Arizona	Tobacco Use - Survey Data	Cessation (Youth)	Percent of Current
2	2015 AZ	Arizona	Tobacco Use - Survey Data	Cessation (Youth)	Percent of Current
3	2015 AZ	Arizona	Tobacco Use - Survey Data	Cessation (Youth)	Percent of Current
4	2015 AZ	Arizona	Tobacco Use - Survey Data	Cessation (Youth)	Quit Attempt in
5	2015 AZ	Arizona	Tobacco Use - Survey Data	Cessation (Youth)	Quit Attempt in
6	2015 AZ	Arizona	Tobacco Use - Survey Data	Cessation (Youth)	Quit Attempt in
7	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
8	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
9	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
10	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
11	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
12	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
13	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
14	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
15	2015 AZ	Arizona	Tobacco Use - Survey Data	Cigarette Use (Youth)	Smoking Status
16	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status
17	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status
18	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status
19	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status
20	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status
21	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status
22	2015 AZ	Arizona	Tobacco Use - Survey Data	Smokeless Tobacco Use (Youth)	User Status

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?

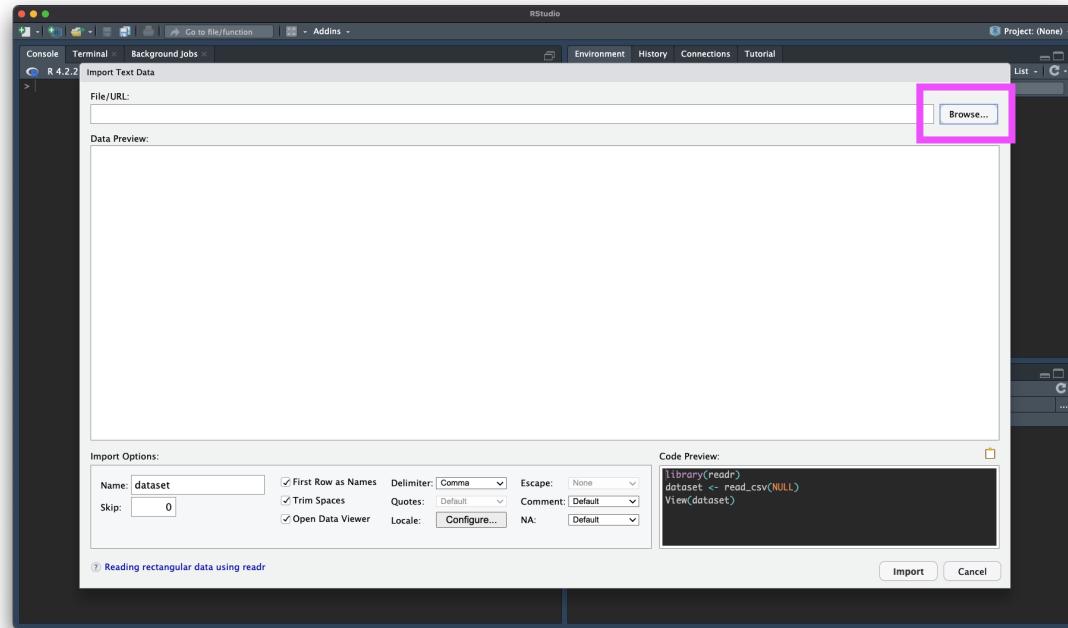
R ran some code in the console (bottom left).

The screenshot shows the RStudio interface. The top-left pane displays a data frame titled "Youth_Tobacco_Survey_YTS_Data" with 17 rows and 5 columns: YEAR, LocationAbbr, LocationDesc, TopicType, TopicDesc, and MeasureDesc. The data consists of survey entries from 2015, primarily for Arizona, regarding tobacco use and cessation. The top-right pane shows the Global Environment, listing "R" and "Global Environment". The bottom-left pane, highlighted with a pink border, is the Console, which contains the following R session history:

```
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: ","
  ...
```

The bottom-right pane is the File Explorer, showing a folder structure under "Home > Desktop".

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 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”?

PC: *autosaves file*

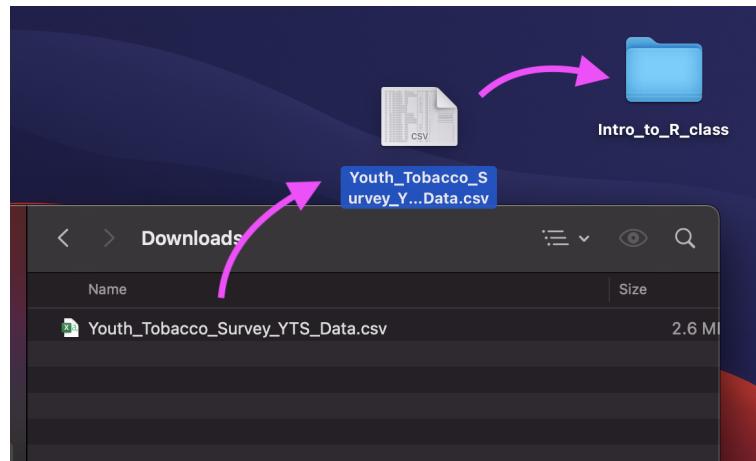
Me: Cool, so where did the
file save?

PC:



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)
```

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 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" "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)" "(34.86...
$ TopicTypeId                : chr [1:9794] "BEH" "BEH" "BEH" "BEH" ...
$ TopicId                   : chr [1:9794] "105BEH" "105BEH" "105BEH" "105BEH" ... 33/44
```

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 on the left shows the command `?read_delim` being run. The help viewer on the right displays the documentation for `read_delim`. The title is "read_delim {readr}" and the subtitle is "Read a delimited file (including csv & tsv) into a tibble". The "Description" section explains that `read_csv()` and `read_tsv()` are special cases of `read_delim()`. The "Usage" section shows the function signature: `read_delim(file, 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: Excel files

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

```
library(readxl)  
read_excel("mtcars.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")
```

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

- 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 Input 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.



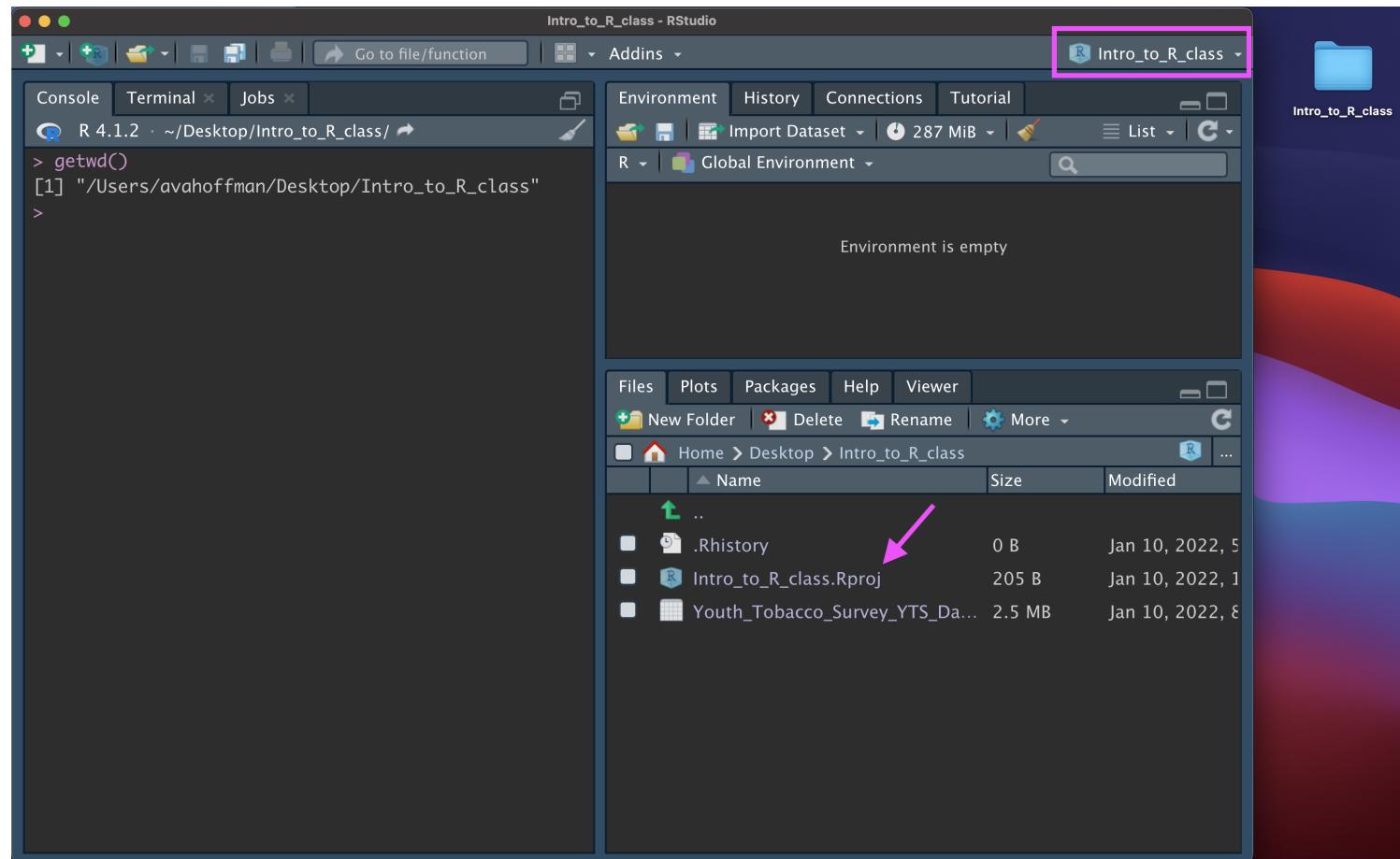
Working Directories

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 Input Lab](#)

