

# Managing Data in R

## Importing Data into R

Glenn Williams

University of Sunderland

2021-10-26 (updated: 2021-10-27)

# Understanding File Systems

Before we read in any data, we need to understand file paths on your computer.

- Computers are often split up into different **drives** where you can store data: e.g. a C drive on a PC, a Macintosh HD drive on newer Macs.
- The more drives you have, the more places data can live.
- Usually we organise files into folders on these drives.
- Any file is saved at a unique **paths** (places) on these drives.

# Understanding File Systems

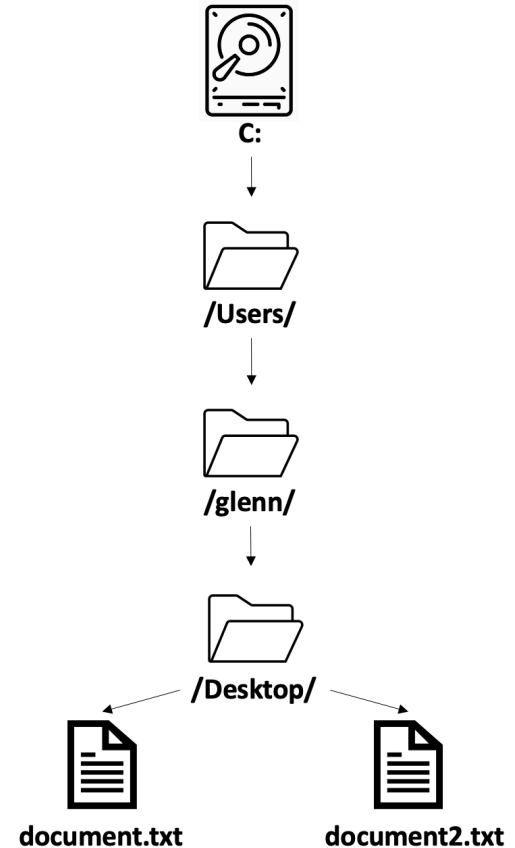
We have a file on our Desktop called "document.txt". This is a text file (.txt extension).

How does the computer know where it is?

- On a PC, it might live at  
C:\Users\glenn\Desktop\document.txt.
- Another file, "document2.txt",  
could live at  
C:\Users\glenn\Desktop\document2.txt.

You need to be able to **uniquely identify** files within file paths.

They either need unique names in a given location, or can have the same name in a different location.



A flow chart of these files on the system.

# How R Reads Files

By default, R will either ask you to:

- Specify the **exact (absolute) file path** to access a file (e.g. `C:\Users\glenn\desktop\document.txt`), or...
- Specify a file path **relative** to the **working directory**.

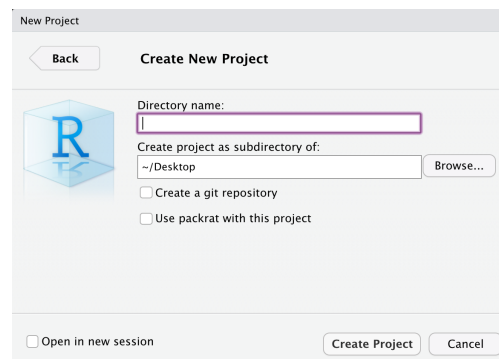
Using a relative file path is best if you want your code to work on any computer besides the one you wrote your code in. That's because other computers aren't likely to have a `C:\Users\glenn\desktop` path.

Where's the **working directory** though?

# The Working Directory

- The working directory in R by default is wherever you open up an R file, or wherever R is installed if you open RStudio without first opening a file.
- **Depending on how you start R, it could be anywhere!**
- RStudio Projects and the here package to the rescue!
- If you open an **RStudio Project**, your working directory is wherever the project is.
- If you read files in with here, no matter where your working directory is right now, it reads files **relative** to where the RStudio Project lives.

Having your work associated with an **RStudio project ensures the working directory is always the same** in any session and on any computer.



Making an RStudio Project.

# Projects and Here: An Example

Read a file called **raw\_data.csv** from a folder on our **desktop** called **my\_data**.

## Method

1. Using an **absolute file path** requires you to state exactly where the file is on your computer. This will only work on your computer.
2. Using a **relative file path** is better, but only works if the working directory is the **my\_data** folder.
3. Using an **RStudio Project** fixes the working directory to the **my\_data** folder. You can then read files relative to that folder using the **here function**.

## Code

```
# using absolute paths
read_csv(
  "C:/Users/glenn/Desktop/
  my_data/raw_data.csv"
)

# using relative paths
read_csv("my_data/raw_data.csv")

# using the here function
read_csv(here("raw_data.csv"))
```

*(3) is best for reproducibility.*

# Reading and Writing Data in R

There are a few inbuilt functions in R that allow us to read in data from a file and to write data to a file, but they sometimes take arguments in different orders and can behave unexpectedly.

- The `tidyverse` has a number of functions for both tasks that are consistent and tell you important information.
- All `tidyverse` functions that read data in from different formats start with **read\_** and end in the file format after the underscore; e.g. `read_csv()`, `read_delim()`, `read_tsv()`.
- All `tidyverse` functions that write data to file in different formats start with **write\_** and end in the name of the file format; e.g. `write_csv()`, `write_delim()`, `write_tsv()`.

# Reading Data into R

## An Example

- We already have some data stored in the /data/ folder. It's stored as a .csv file.
- Values in the file are separated by commas (hence, **comma-seperated values; csv**).
- It's a good idea to save data (e.g. from Excel) in this format for sharing; it's lightweight and can be read for free by most programs.



# Reading Data into R

## An Example

- We read the data in using `read_csv()`
- We tell R where to find the data using `here()`.
- We then assign the data to an object called **raw\_data** so we can work with it in R.
- By default, `read_csv()` tells us how R parsed each column (i.e. what the data is stored as).

```
raw_data <- read_csv(here("data", "factorial_data.csv"))
```

```
## Parsed with column specification:  
## cols(  
##   subj_id = col_character(),  
##   list_id = col_double(),  
##   item_id = col_double(),  
##   A = col_character(),  
##   B = col_character(),
```

# Recap

We know...

- how **file paths** work and that these paths can be very different depending upon your computer set up.
- why **relative file paths** set up in a **working directory** is often the most user-friendly method of accessing and writing files.
- why using **RStudio Projects** and the `here` package makes working with files easier.
- how to read different file types into R.