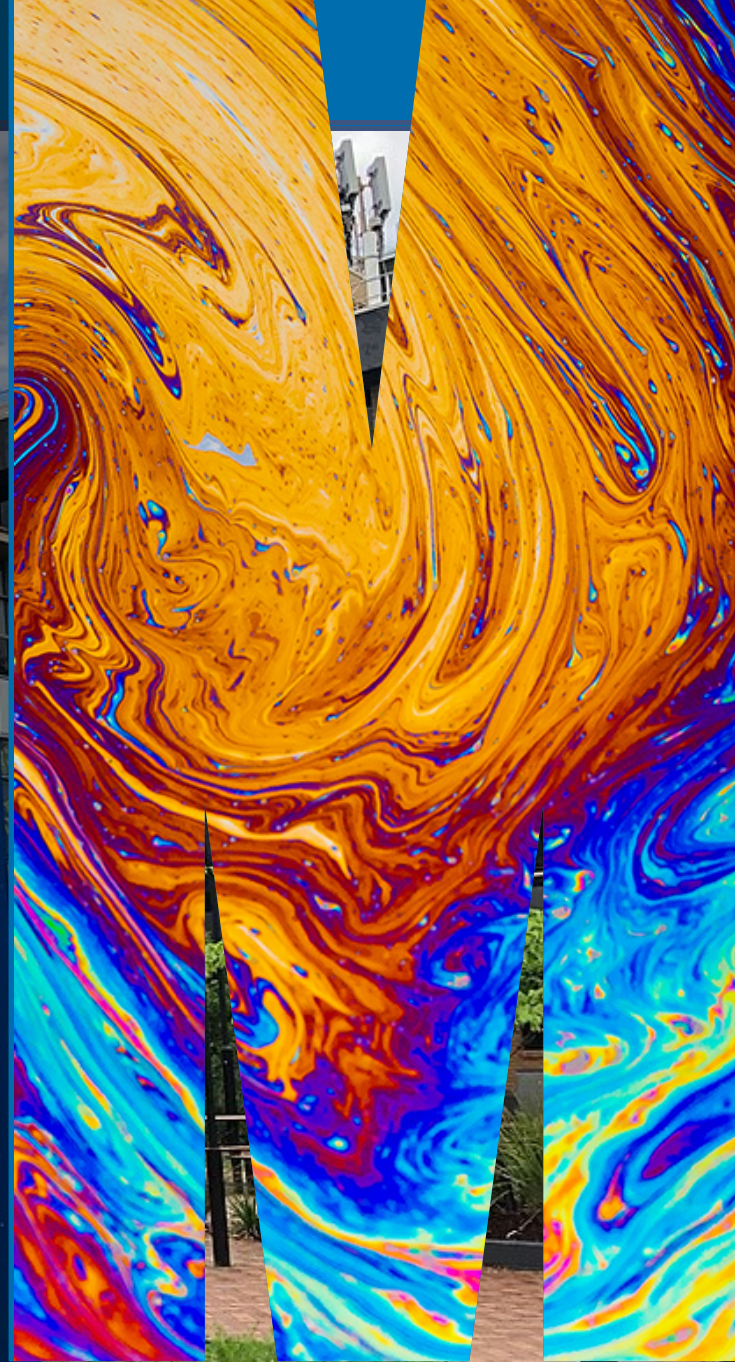
ETC5510: Introduction to Data Analysis

Week 6, part A

Web Scraping

Style, file paths, & functions



Lecturer: *Nicholas Tierney & Stuart Lee*

Department of Econometrics and Business Statistics

ETC5510.Clayton-x@monash.edu

April 2020

Upcoming due dates

- Midsemester test: Opens after class on the 29th April
- Assignment 2: 13th May (Released this week)
- Practical Exam: 3rd June
- Project: 8th June (See examples of past projects in assessments)

Midsemester test

- Completed on Moodle as an online MCQ
- Will be available from 29th April 8pm until 1st of May 11.59pm
- Once started you will have 1 hour and 10 minutes to complete
- Based on materials from weeks 1 - 5

How do I study for midsemester?

- Take practice midsem available on course site
- Revise lecture slides and draw mental models for core concepts
- Look over the lab exercises
- Look over relevant chapters in R4DS and complete exercises

Practical Exam?

- A live data analysis
- ~ 1 Hour to complete

Project?

- Collect / find your own data
- Clean the data
- Determine interesting questions to answer about the data
- Plan how to execute analysis of the data
- Communicate the idea, data cleaning, and analysis (oral presentation)
- Further details are on the course website

Lecture Overview

- Organising your own folders
- File paths and Rstudio projects
- (Intro to) Using functions

File Paths and organising yourself

- It's important when you start working on your own machine that you understand *file storage hygiene*.
- It helps prevent unexpected problems and makes you more productive
- You'll spend less time fighting against strange file paths.
- Not sure what a file path is? We will explain that as well!

Your Turn

1. What your normal "workflow" is for starting a new project / assessment
2. Possible challenges that might arise when maintaining your project / assessment
3. What is a file path?

What even is a file path?

- This all might be a bit confusing if you don't know what a file path is.
- A file path is: "the machine-readable directions to where files on your computer live."
- So, this file path:

```
/Users/njtierney/rmd4sci-materials/demo-gapminder.Rmd
```

Describes the location of the file "demo-gapminder.Rmd".

What even is a file path

We could visualise this path:

```
/Users/njtierney/rmd4sci-materials/demo-gapminder.Rmd
```

as:

```
users
   └── njtierney
         └──  rmd4sci-materials
                └── demo-gapminder.Rmd
```

What even is a file path

- To read in the `gapminder.csv` file, you might need to write code like this:

```
gapminder <- read_csv("/Users/njtierney/Desktop/rmd4sci-materials/data/gapminder.csv
```

This is a problem, because this is not portable code.

A Mantra: Start a new project - start an RStudio project

- This section is heavily influenced by Jenny Bryan's great blog post on project based workflows.
- Sometimes this is the first line of an R Script or R markdown file.

```
setwd("c:/really/long/file/path/to/this/directory)
```

- What do you think the `setwd` code does?

What does `setwd()` do?

- "set my working directory to this specific working directory".
- It means that you can read in data and other things like this:

```
data <- read_csv("data/mydata.csv")
```

- Instead of

```
data <- read_csv("c:/really/long/file/path/to/this/directory/data/mydata.csv")
```

Using `setwd()`

- This has the effect of **making the file paths work in your file**
- This is a problem because, among other things, using `setwd()`:
  - Has 0% chance of working on someone else's machine (**this includes you in >6 months**)
  - Your file is not self-contained and portable. (Think: *"What if this folder moved to /Downloads, or onto another machine?"*)
- To get this to work, you need to hand edit the file path to your machine.
- This is painful. And when you do this all the time, it gets old, fast.

If you have an RStudio project file inside the `rmd4sci-materials` folder, you can instead write the following:

```
gapminder <- read_csv("data/gapminder.csv")
```

Your Turn: Think about this before discussion

- (1-2 minutes) What folders are above the `health.csv` file in the following given file path?

`"/Users/miles/etc5510/week1/data/health.csv"`

- and the result of using the below code in `demo-gapminder.Rmd`, then using the code, and then moving this to another location, say inside your C drive?

```
setwd("Downloads/etc5510/week1/week1.Rmd)
```

Is there an answer to the madness?

- This file path situation is a real pain.
- Is there an answer to the madness?

The answer is yes!

I highly recommend when you start on a new idea, new research project, paper. Anything that is new. It should start its life as an **rstudio project**.

Rstudio projects

An rstudio project helps keep related work together in the same place. Amongst other things, they:

- Keep all your files together
- Set the working directory to the project directory
- Starts a new session of R
- Restore previously edited files into the editor tabs
- Restore other rstudio settings
- Allow for multiple R projects open at the same time.

Rstudio projects

This helps keep you sane, because:

- Your projects are each independent.
- You can work on different projects at the same time.
- Objects and functions you create and run from project idea won't impact one another.
- You can refer to your data and other projects in a consistent way.

And finally, the big one

RStudio projects help resolve file path problems, because they automatically set the working directory to the location of the rstudio project.

The "here" package

- RStudio projects help resolve file path problems
- In some cases you might have many folders in your r project. To help navigate them appropriately, you can use the `here` package to provide the full path directory, in a compact way.

```
here::here("data")
```

returns

```
[1] "/Users/njtierney/Desktop/rmd4sci-materials/data"
```

The `here` package

```
here::here("data", "gapminder.csv")
```

returns

```
[1] "/Users/njtierney/Desktop/rmd4sci-materials/data/gapminder.csv"
```

You can read the above `here` code as:

> *In the folder `data`, there is a file called `gapminder.csv`, can you please give me the full path to that file?*

The `here` package

This is really handy for a few reasons:

1. It makes things *completely* portable
2. Rmarkdown documents have a special way of looking for files, this helps eliminate file path pain.
3. If you decide to not use RStudio projects, you have code that will work on *any machine*

Remember

If the first line of your R script is

```
setwd("C:\Users\jenny\path\that\only\I\have")
```

I will come into your office and SET YOUR COMPUTER ON FIRE 🔥.

-- Jenny Bryan

Aside: How to create an RStudio project

- Go to section 5.12 of rmarkdown for scientists

Summary of file paths and rstudio projects

In this lesson we've:

- Learnt what file paths are
- How to setup an rstudio project
- How to construct full file paths with the `here` package

Recommendations on how to file structure in ETC5510

File structures for class

Approach 1: Folder per week

```
/Users/njtierney/etc5510/week_1/

users
   └── njtierney
       └──  etc5510
            └── etc5510.Rproj
            └── week_1
                └── lecture_1.html
                └── lecture_1.pdf
                └── ida-exercise-1.Rmd
                └── data
                    └── file.csv
            └── week_2
                └── lecture_2.html
                └── lecture_2.pdf
                └── ida-exercise-2.Rmd
                └── data
                    └── file.csv
```

File structures for class

Approach 2: flater structure

```
/Users/njtierney/etc5510/

users
   └── njtierney
       └──  etc5510
            └── etc5510.Rproj
            └── lecture_1.html
            └── lecture_1.pdf
            └── ida-exercise-1.Rmd
            └── data
                └── data.csv
```

Remember: There is no one true "correct" file format

It's just important to have a system

Motivating Functions

Do you see any problems with this code?

```
st_episode <- st %>%
  html_nodes(".np_right_arrow .bp_sub_heading") %>%
  html_text() %>%
  str_replace(" episodes", "") %>%
  as.numeric()

got_episode <- got %>%
  html_nodes(".np_right_arrow .bp_sub_heading") %>%
  html_text() %>%
  str_replace(" episodes", "") %>%
  as.numeric()

twd_episode <- got %>%
  html_nodes(".np_right_arrow .bp_sub_heading") %>%
  html_text() %>%
  str_replace(" episodes", "") %>%
  as.numeric()
```

Next Lecture: Why functions?

- Automate common tasks in a power powerful and general way than copy-and-pasting:
  - You can give a function an evocative name that makes your code easier to understand.
  - As requirements change, you only need to update code in one place, instead of many.
  - You eliminate the chance of making incidental mistakes when you copy and paste (i.e. updating a variable name in one place, but not in another).
- Down the line: Improve your reach as a data scientist by writing functions (and packages!) that others use

Take the lab quiz!