Doing Your Work Properly

Modern Plain Text Computing

Week 01

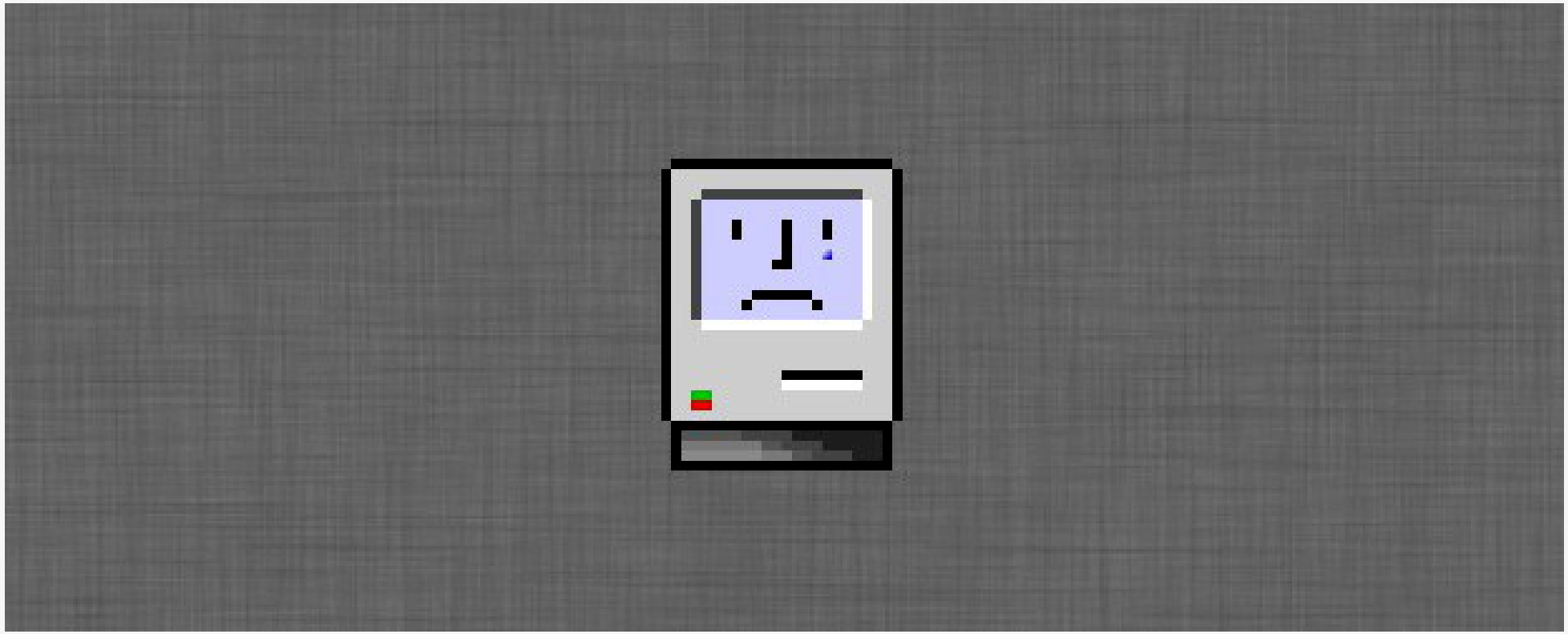
Kieran Healy
kieran.healy@duke.edu

August 2025

Motivation

We depend on our computers

Technical computing is frustrating



Can we make it fun?

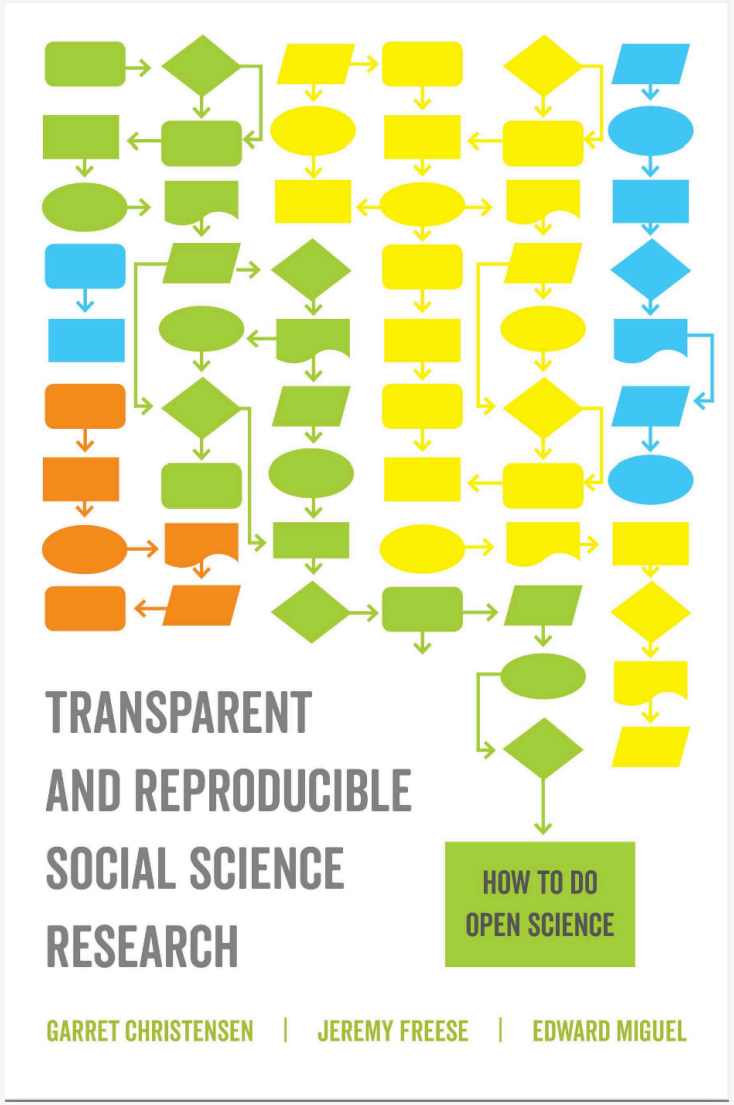
OK but can we eliminate frustration?

But we can make it work

Also it’s weirdly satisfying once you get into it.

Bigger picture

Science is hard



Scientific research is difficult.

It depends on norms that entail a lot of trust.

Doing it badly or fraudulently is too easy.

We have tools to help us do the right thing.

(But our tools can't *make* us do the right thing.)

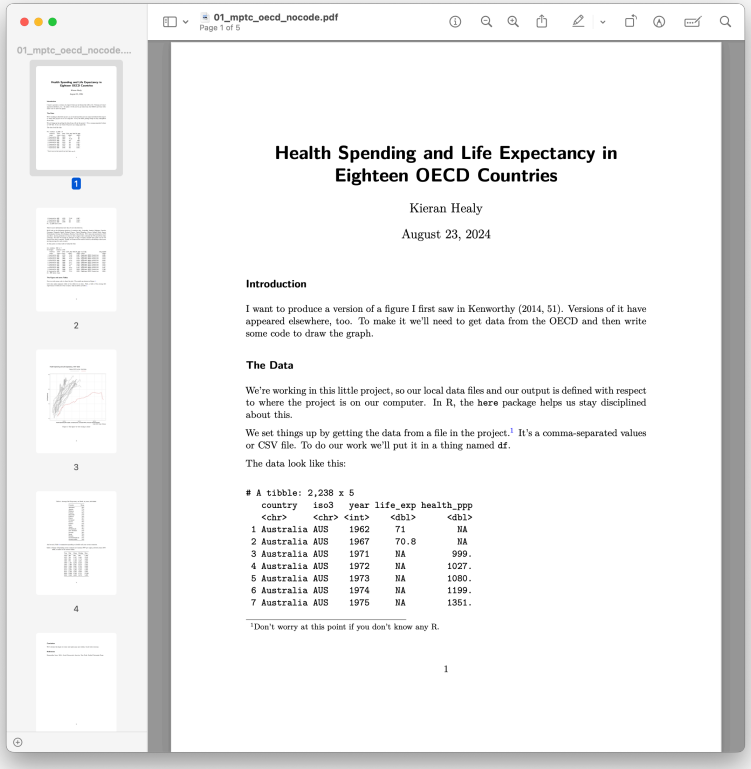
This course

What it’s not

What it is

The Whole Game

A tiny research project



Health Spending and Life Expectancy in Eighteen OECD Countries

Kieran Healy

August 23, 2024

Introduction

I want to produce a version of a figure I first saw in Kenworthy (2014, 51). Versions of it have appeared elsewhere, too. To make it we'll need to get data from the OECD and then write some code to draw the graph.

The Data

We're working in this little project, so our local data files and our output is defined with respect to where the project is on our computer. In R, the `here` package helps us stay disciplined about this.

We set things up by getting the data from a file in the project.[1] It's a comma-separated values or CSV file. To do our work we'll put it in a thing named `df`.

The data look like this:

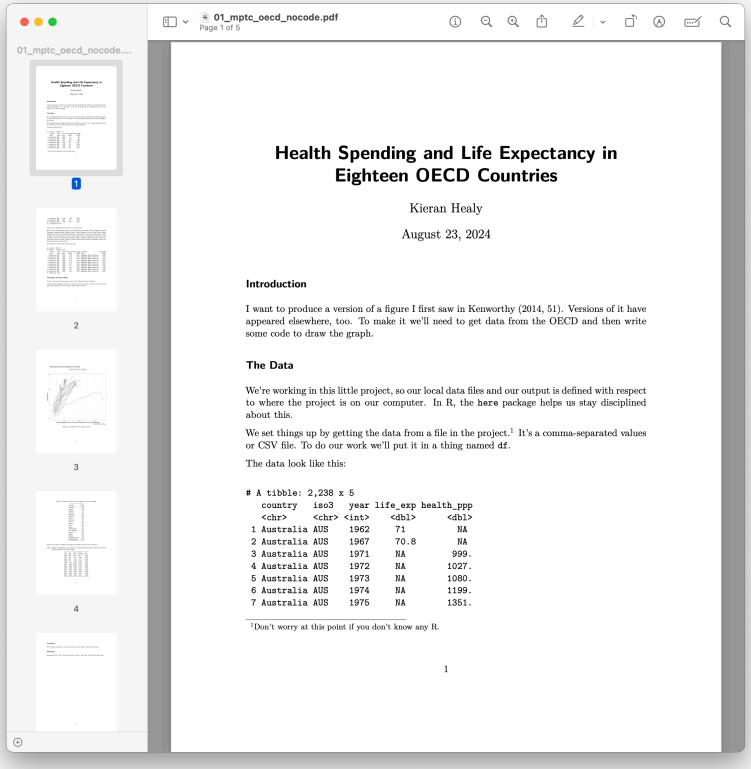
```
# A tibble: 2,238 x 5
   country   iso3   year life_exp health_ppp
   <chr>     <chr> <int>    <dbl>      <dbl>
 1 Australia AUS    1962     71          NA
 2 Australia AUS    1967     70.8        NA
 3 Australia AUS    1971     NA         999.
 4 Australia AUS    1972     NA        1027.
 5 Australia AUS    1973     NA        1080.
 6 Australia AUS    1974     NA        1199.
 7 Australia AUS    1975     NA        1351.
```

[1] Don't worry at this point if you don't know any R.

1

Click here for a PDF of the tiny paper

Features of our tiny paper



It's a file!

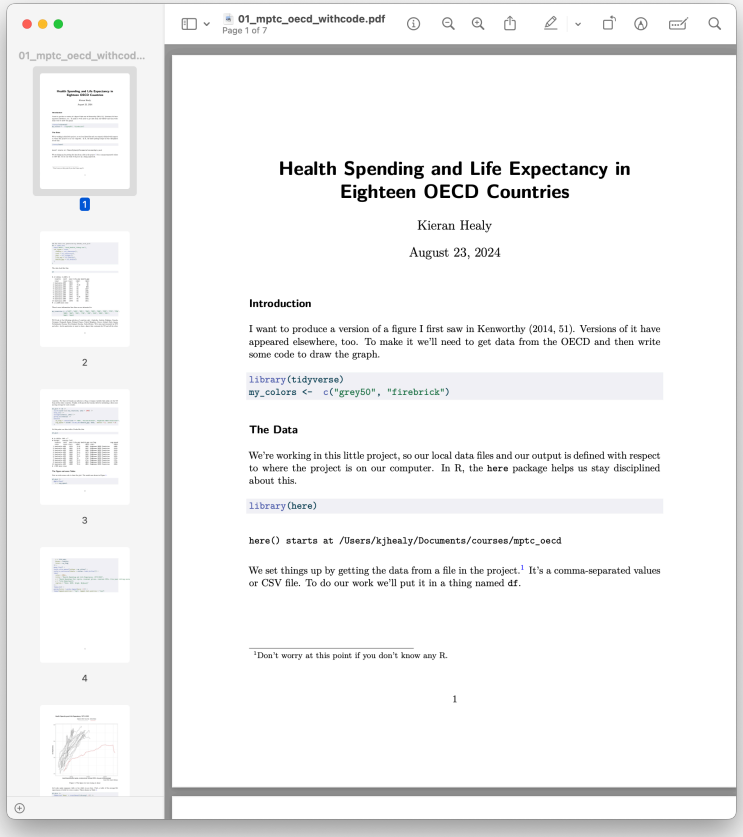
Of a particular type (a PDF)

You can't easily edit it

It has a title, author, sections, a figure, tables, references, a footnote, and a bibliography.

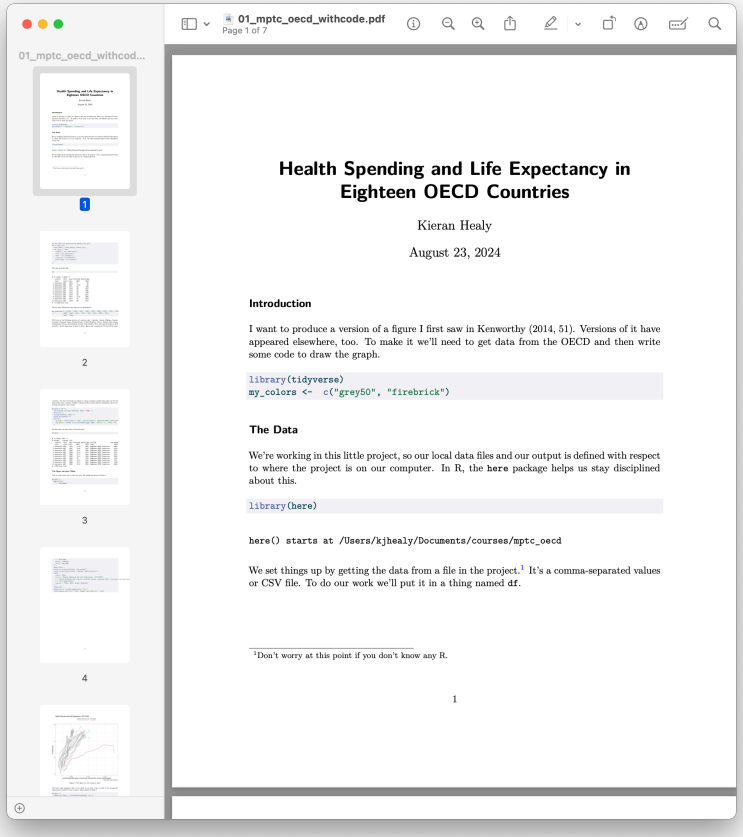
That is, it has many of the elements of a scholarly paper, in miniature.

A slightly different version



Click here for a PDF of this version of the tiny paper

Features of this version



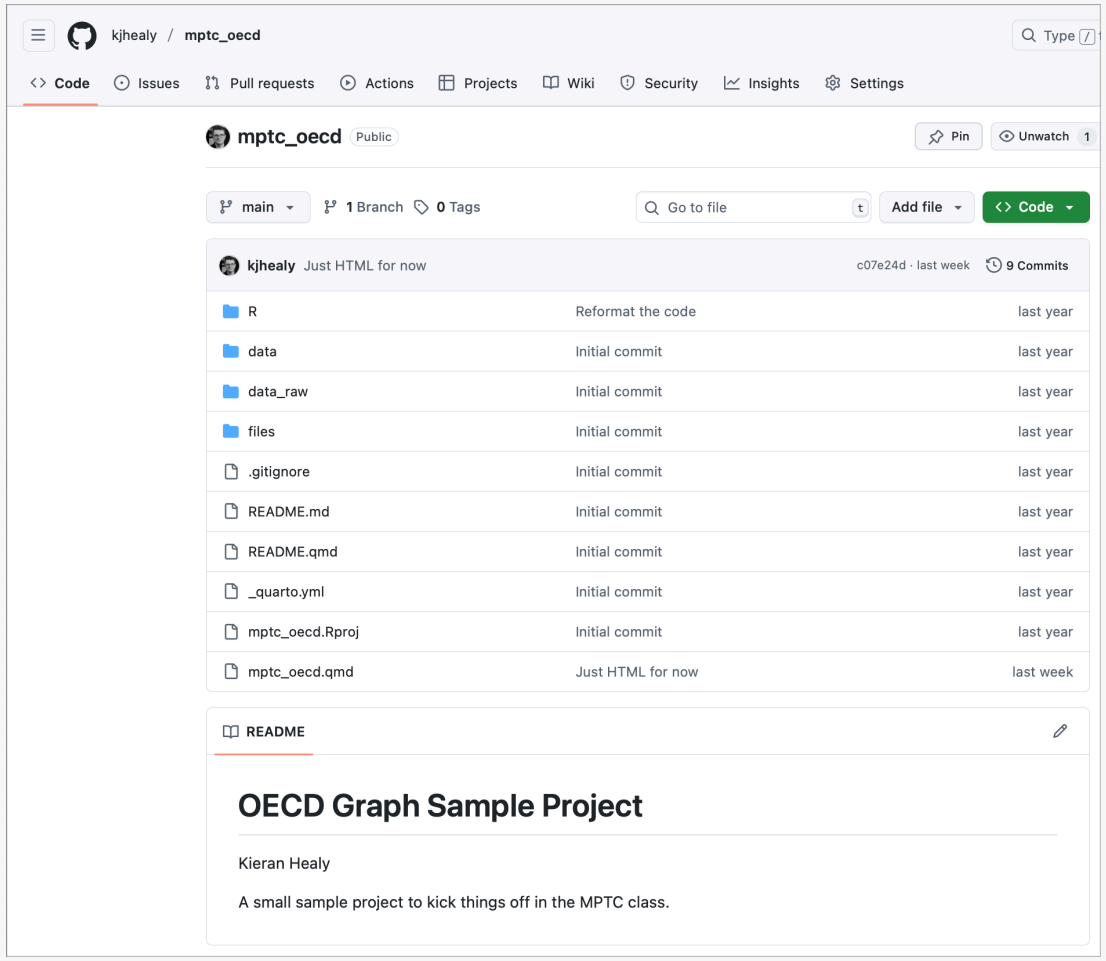
It's also a file!

Of a particular type (still a PDF)

You still can't easily edit it

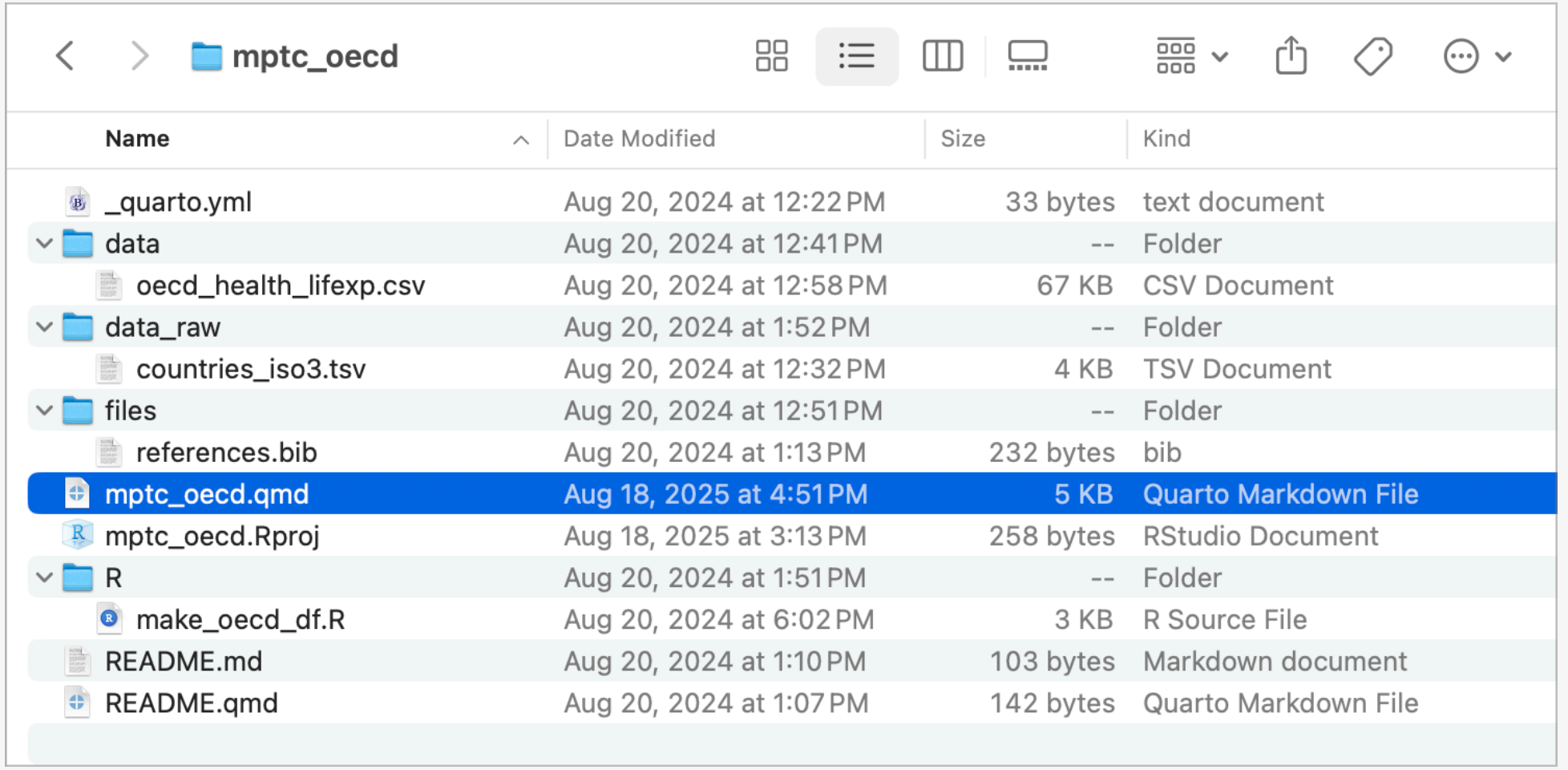
It has all the stuff in the original version *plus* a bunch of code that we can now see that was not shown before.

The source for both versions



https://github.com/kjhealy/mptc_oecd

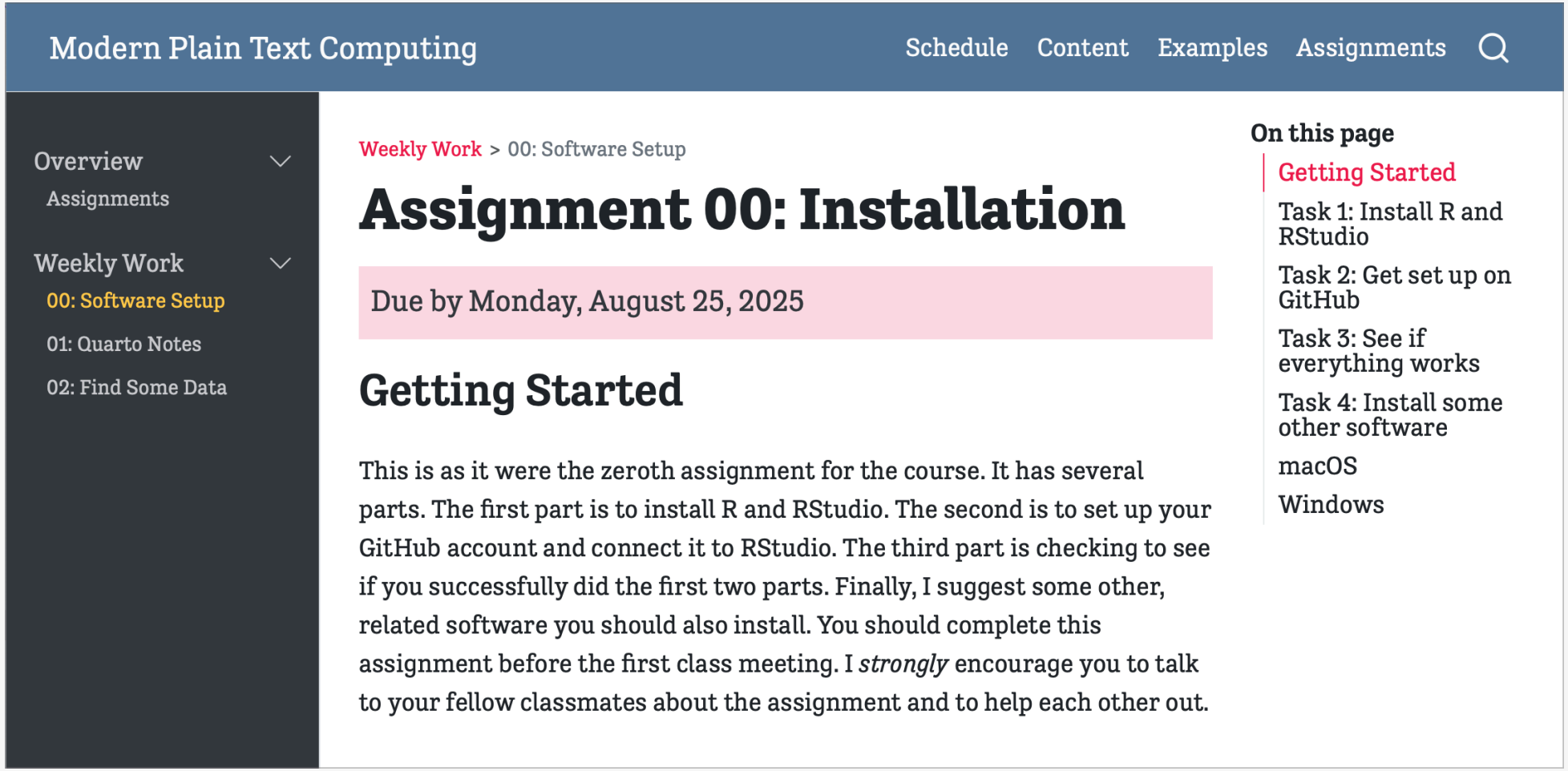
Once you've downloaded it



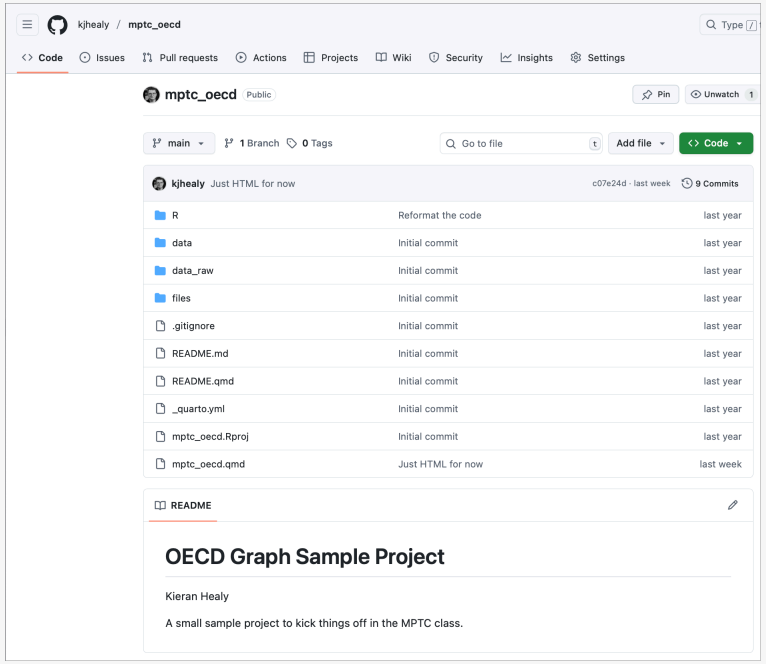
It's a folder on your computer with a bunch of files in it.

How did this go?

Week 00 Assignment: Install R and RStudio



The project's GitHub page



Go here and download it: https://github.com/kjhealy/mptc_oecd

We're going to open it in RStudio, take a look at the various parts of the project, and turn it into a document.

Don't worry at this point if you're not sure what's happening, or what GitHub or RStudio are.

(Sound of Engine Trying to Start)

This slide is up because we are installing the required software by following the instructions handed out earlier. In other words, we are discovering the inevitable idiosyncrasies of everyone's individual setup, the vagaries of various operating systems, the intrinsic difficulty of following documented steps in a procedure, the hidden bits of implicit knowledge or not-fully-articulated steps that are nevertheless necessary, the high prevalence of ordinary error and failure in everyday life, and the awful grip of chance on human affairs in general.

Main Ideas

That’s a lot all at once

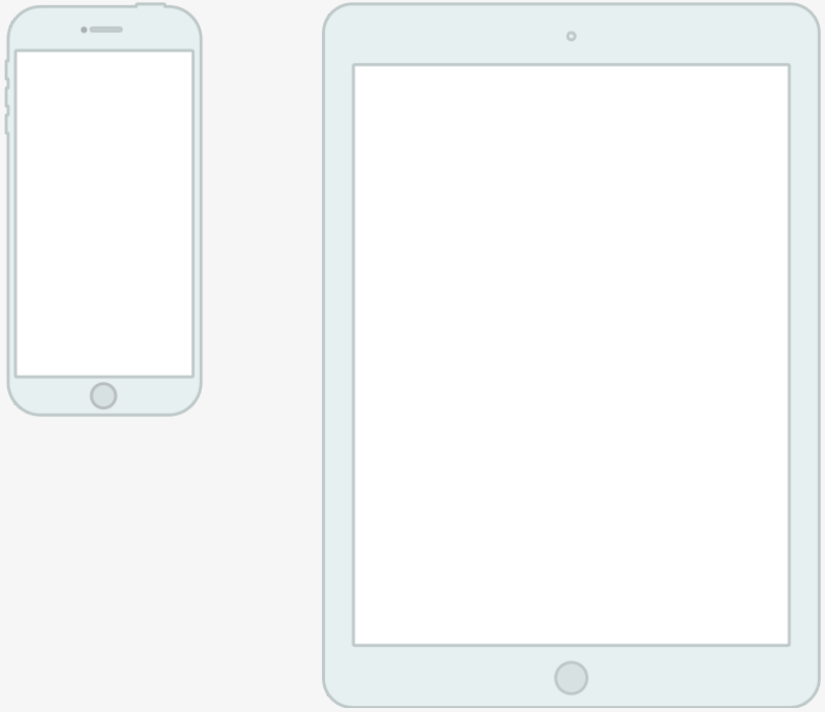
We’re going to back up and go through these pieces slowly.

Keep in mind why we’re doing it. (We want to reliably produce a scholarly paper).

Also start thinking about why the tools we’re using might look like this.

Two Revolutions in Computing

What everyday computing is now



Touch-based user interface.

Foregrounds a single application.

Dislikes multi-tasking.*

Hides the file system.

Very far underneath, it's often the 1970s, Unix, and the command-line. But usually you don't get to see this.

Multi-tasking: I mean, “Making different specialized applications and resources work together in the service of a single but multi-dimensional project”, not “Checking social media while also listening to a talk and waiting for an update from the school nurse.”

Where technical computing lives

Most interaction via windows, icons, a pointer, and the keyboard.

Multi-tasking via multiple windows at once is standard.

Exposes and leverages the file system.

Using several specialized applications in concert is common.

Underneath, it's often the 1970s, Unix, and the command-line. You get to see and often need to interact with this directly.

Where technical computing lives

For technical computing in the sense of doing “statistics” this toolset is by now really good.

It’s also very good for computing in the sense that *all* scholarly work is technical, i.e. it must meet specific procedural demands.

But these tools are grounded in a paradigm that is increasingly far away from the everyday use of our most common computing devices.

So why do we continue to use and develop them?


CONTROL

Control, not Productivity

Productivity is great and everything, but not why we do all this.

The most important thing is to be able to *confidently know and clearly show what it was that you did* in the service of doing your work properly.

“Office” vs “Engineering” approaches

The challenge of “Knowing and Showing” gives rise to questions like these:

What is “real” in your project?

What is the final output?

How is it produced?

How are changes managed?

Can you do it again?

Different Answers

Office model

Formatted documents are real.

Intermediate outputs are cut and pasted into documents.

Changes are tracked inside files.

Final output is often in the same format you've been working in, e.g. a Word file, or that file converted to a PDF.

Different strengths and weaknesses

Office model

Documents look like documents.

Everyone knows Word, Excel, or Google Docs.

“Track changes” is powerful and easy in a single document.

Hm, I can’t remember how I made this figure.

Where did this table of results come from?

`Paper_edits_FINAL_kh-1.docx`

Each approach generates solutions to its own problems