

# Doing Your Work Properly

*Modern Plain Text Computing*

*Week 01*

Kieran Healy

kieran.healy@duke.edu

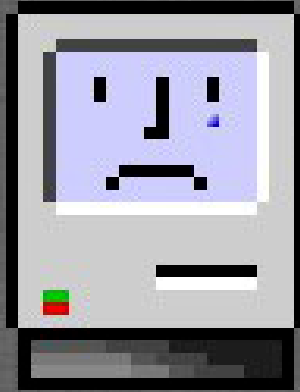
August 2025

# 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

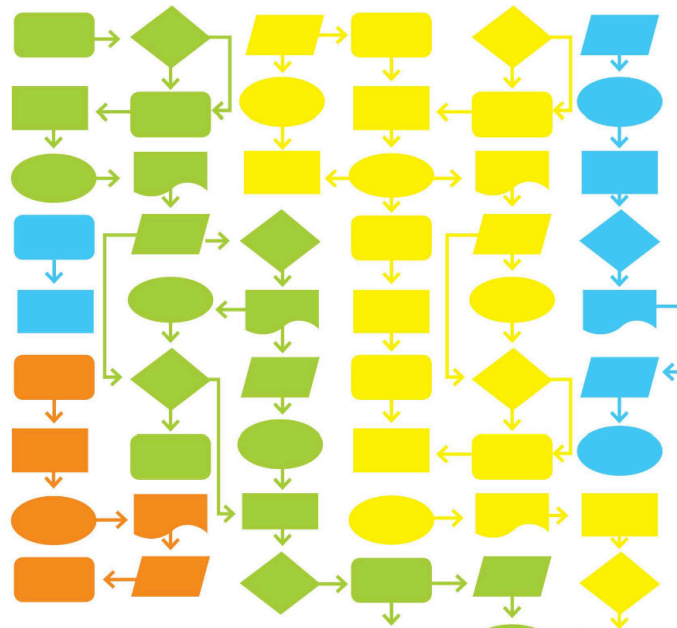
Scientific research is difficult.

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.)



TRANSPARENT  
AND REPRODUCIBLE  
SOCIAL SCIENCE  
RESEARCH

HOW TO DO  
OPEN SCIENCE

GARRET CHRISTENSEN | JEREMY FREESE | EDWARD MIGUEL

# This course

**What it's not**

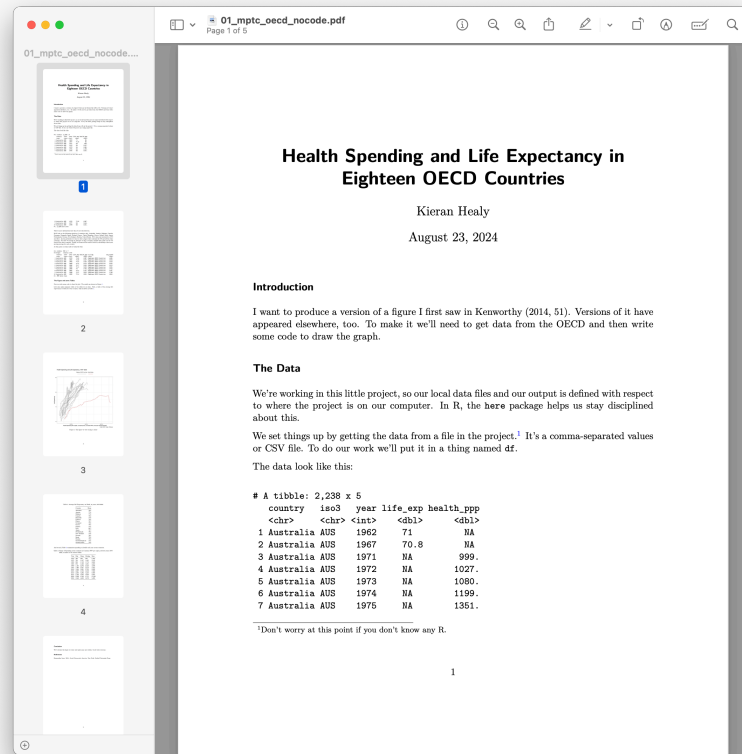
**What it is**



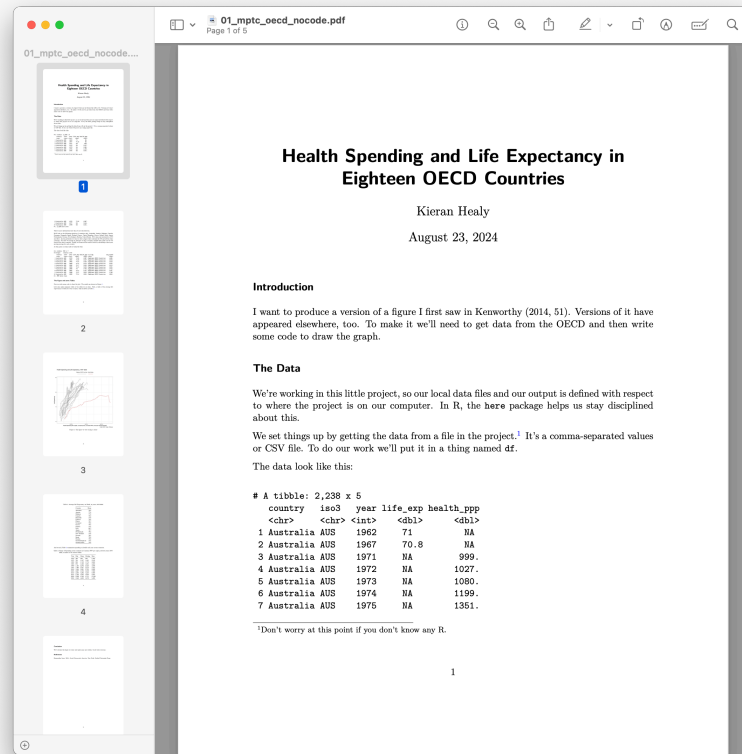
# The Whole Game

# A tiny research project

[Click here for a PDF of the tiny paper](#)



# Features of our tiny paper



It's a file!

Of a particular type (a PDF)

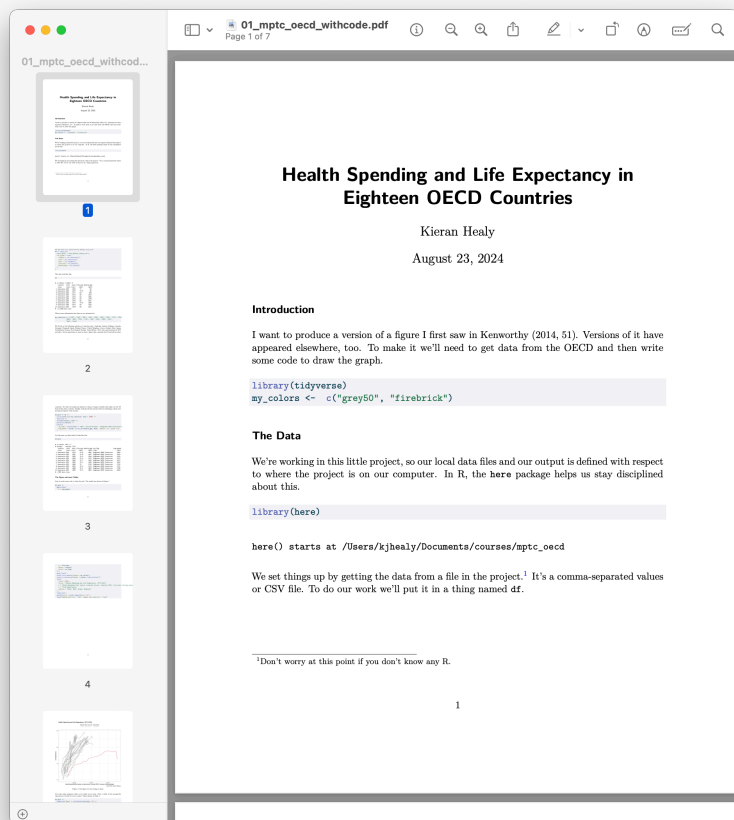
You can't easily edit it

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

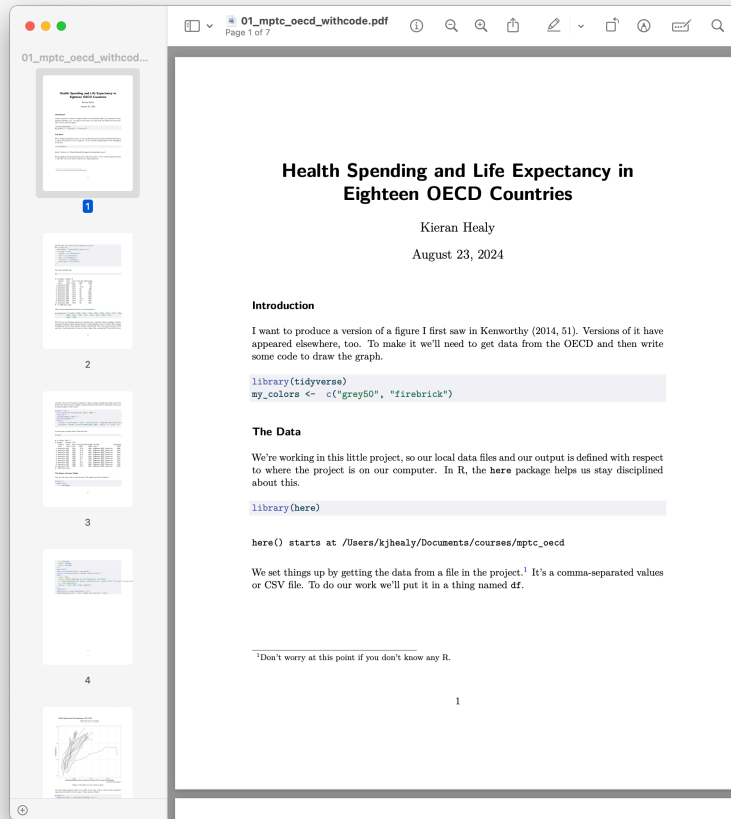
That is, it has many of the elements of a scholarly paper, in miniature.

# 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

The screenshot shows the GitHub repository page for `mptc_oecd` by user `kjhealy`. The repository is public and has 1 branch (main) and 0 tags. The commit history shows 9 commits, with the latest commit being `c07e24d` from last week. The file list includes:

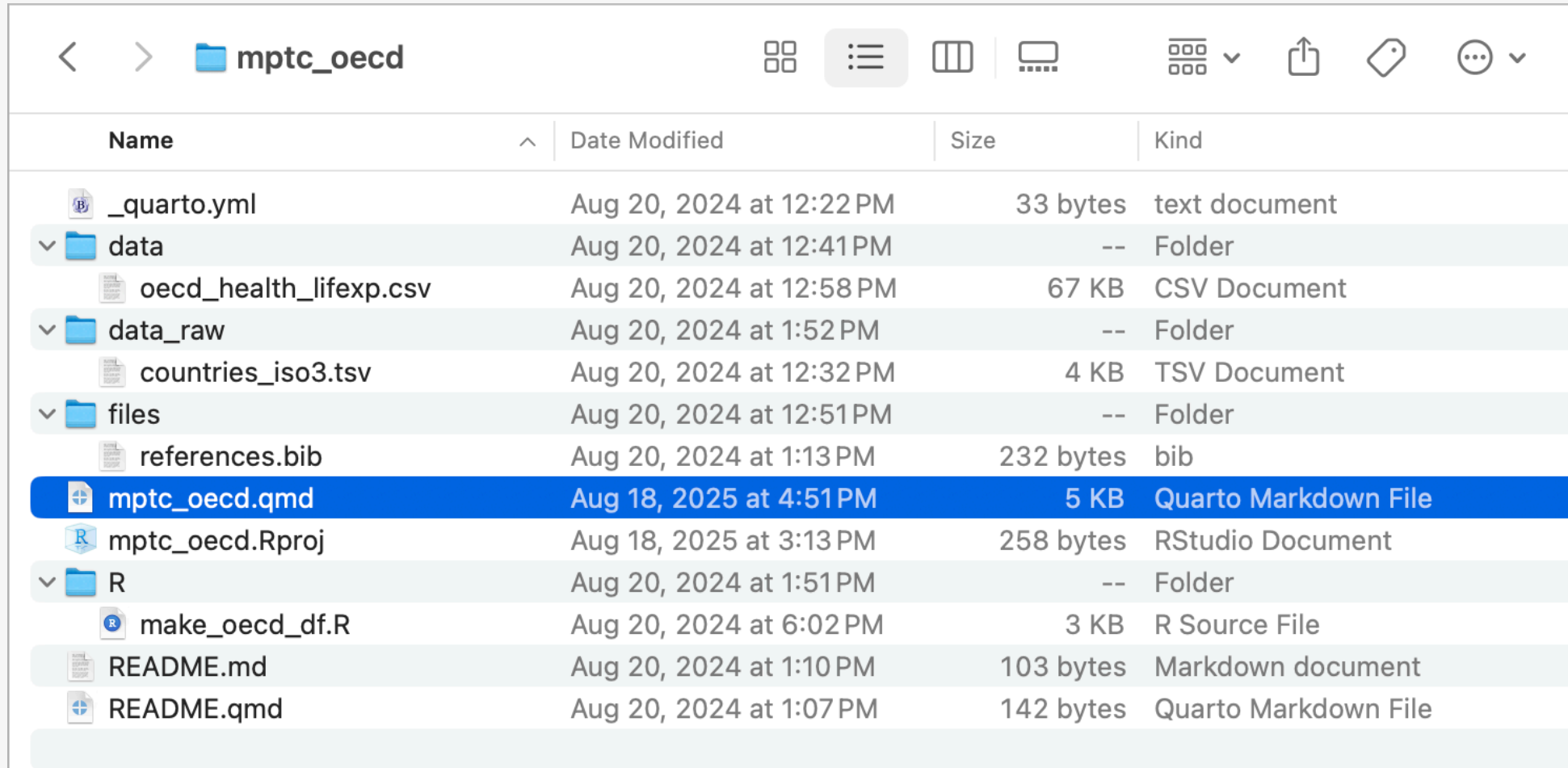
File	Commit Message	Commit Date
R	Reformat the code	last year
data	Initial commit	last year
data_raw	Initial commit	last year
files	Initial commit	last year
.gitignore	Initial commit	last year
README.md	Initial commit	last year
README.qmd	Initial commit	last year
_quarto.yml	Initial commit	last year
mptc_oecd.Rproj	Initial commit	last year
mptc_oecd.qmd	Just HTML for now	last week

The README section is titled "OECD Graph Sample Project" and is authored by Kieran Healy. It describes the project as "A small sample project to kick things off in the MPTC class."














[https://github.com/kjhealy/mptc\\_oecd](https://github.com/kjhealy/mptc_oecd)



# Once you've downloaded it



The screenshot shows a file explorer window with the address bar displaying the path to a folder named 'mptc\_oecd'. The window contains a table of files and folders. The file 'mptc\_oecd.qmd' is highlighted in blue. The table has four columns: Name, Date Modified, Size, and Kind. The files listed include configuration files, data folders, raw data, references, and source code files.

Name	Date Modified	Size	Kind
 _quarto.yml	Aug 20, 2024 at 12:22 PM	33 bytes	text document
▼  data	Aug 20, 2024 at 12:41 PM	--	Folder
 oecd_health_lifexp.csv	Aug 20, 2024 at 12:58 PM	67 KB	CSV Document
▼  data_raw	Aug 20, 2024 at 1:52 PM	--	Folder
 countries_iso3.tsv	Aug 20, 2024 at 12:32 PM	4 KB	TSV Document
▼  files	Aug 20, 2024 at 12:51 PM	--	Folder
 references.bib	Aug 20, 2024 at 1:13 PM	232 bytes	bib
 mptc_oecd.qmd	Aug 18, 2025 at 4:51 PM	5 KB	Quarto Markdown File
 mptc_oecd.Rproj	Aug 18, 2025 at 3:13 PM	258 bytes	RStudio Document
▼  R	Aug 20, 2024 at 1:51 PM	--	Folder
 make_oecd_df.R	Aug 20, 2024 at 6:02 PM	3 KB	R Source File
 README.md	Aug 20, 2024 at 1:10 PM	103 bytes	Markdown document
 README.qmd	Aug 20, 2024 at 1:07 PM	142 bytes	Quarto Markdown File

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

Modern Plain Text Computing

ScheduleContentExamplesAssignments

Q

Overview

Assignments

Weekly Work

00: Software Setup

01: Quarto Notes

02: Find Some Data

Weekly Work > 00: Software Setup

Assignment 00: Installation

Due by Monday, August 25, 2025

Getting Started

This is as it were the zeroth assignment for the course. It has several parts. The first part is to install R and RStudio. The second is to set up your GitHub account and connect it to RStudio. The third part is checking to see if you successfully did the first two parts. Finally, I suggest some other, related software you should also install. You should complete this assignment before the first class meeting. I *strongly* encourage you to talk to your fellow classmates about the assignment and to help each other out.

On this page

Getting Started

Task 1: Install R and RStudio

Task 2: Get set up on GitHub

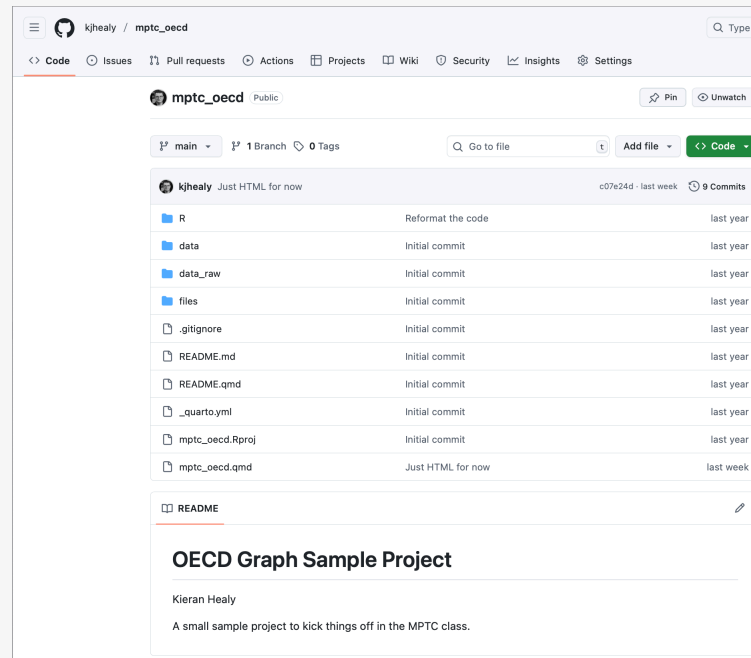
Task 3: See if everything works

Task 4: Install some other software

macOS

Windows

# The project's GitHub page



Go here and download it: [https://github.com/kjhealy/mptc\\_oecd](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

Main Idea

## 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?

The background is a monochromatic red with a fine grid overlay. In the center, there is a silhouette of a person standing. To the left and right of the person are silhouettes of industrial structures, possibly cranes or control towers. At the top center, there is a bright white rectangular light source. The overall composition is symmetrical and has a high-contrast, industrial feel.

# 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**