Filepaths and Projects

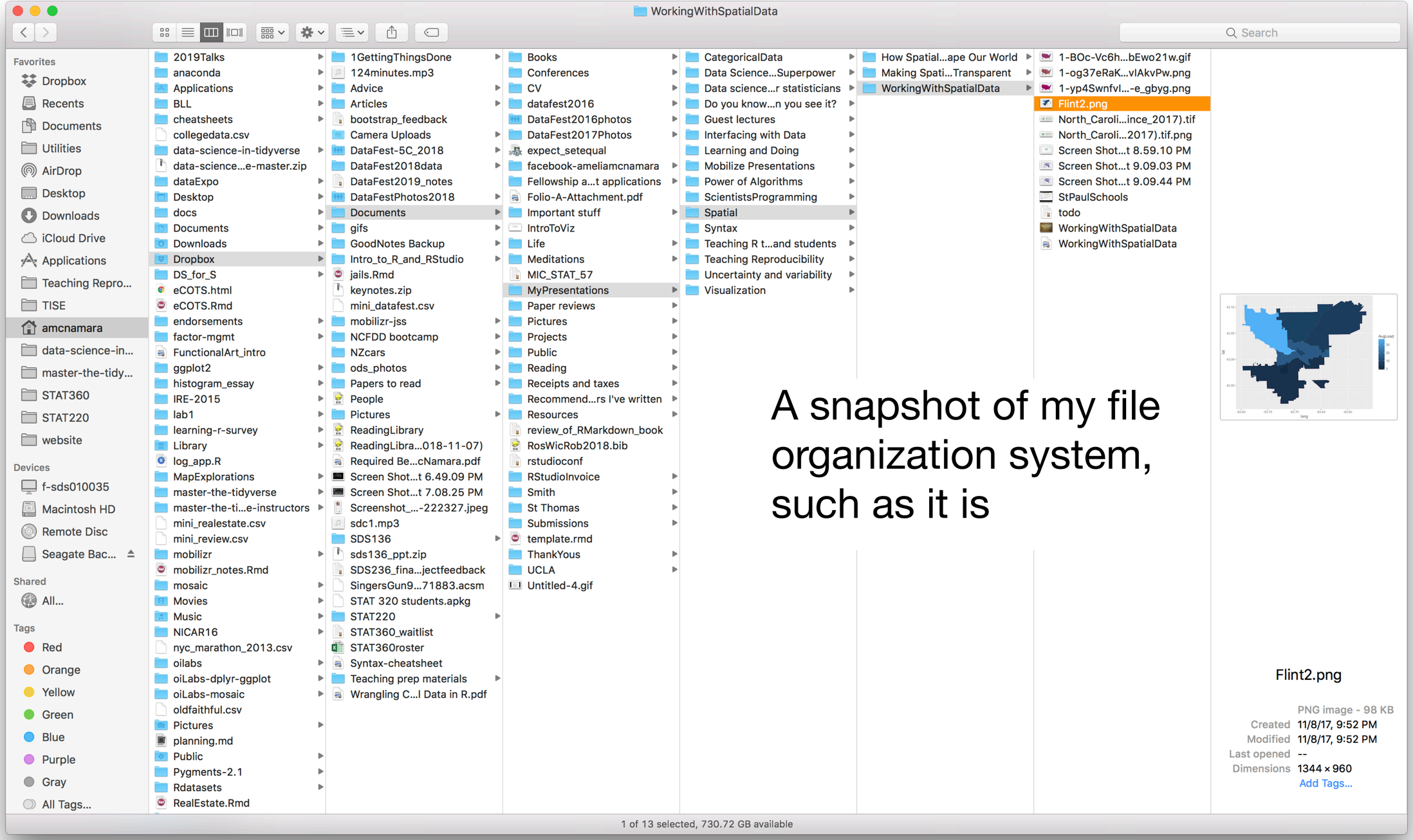
Filepaths are less important in today's computing landscape

If you have files stored in OneDrive, or GoogleDrive, or iCloud, they are findable with search functionality that rivals Google (or in some cases, **is** Google).

Computers have great search functions, like Apple's Spotlight and Windows' searchable File Explorer. iPads have actually removed the ability to walk through your filesystem.

But… it wasn't always like this. It used to be that you needed to know "where" your file was stored (scare quotes because it isn't a physical location) in order to find it. So, we used folders to keep everything organized…

/Users/amcnamara/Dropbox/Documents/MyPresentations/Spatial/WorkingWithSpatialData/Flint2.png



Computers and programming languages still need to use filepaths to locate things

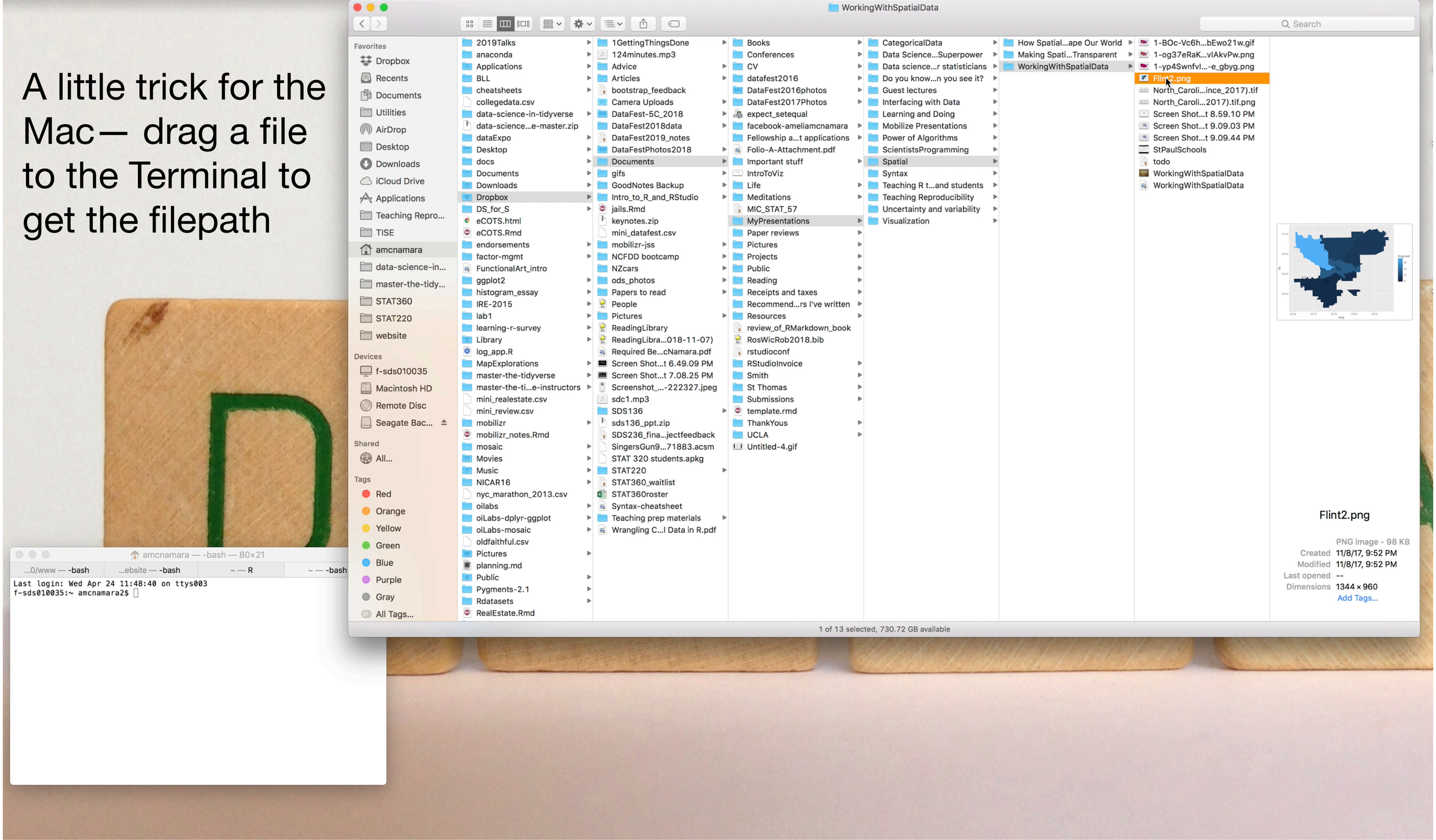
```
> read_csv("labike.csv")

Error: 'labike.csv' does not exist in current working
directory ('/Users/amcnamara').
```

One solution is to use **absolute paths**

```
> read_csv("/Users/amcnamara/Dropbox/Documents/Projects/data/labi
```

A little trick for the Mac— drag a file to the Terminal to get the filepath



It’s all about the slashes!

Mac: `/Users/amcnamara/Dropbox/Documents/MyPresentations/Spatial/WorkingWithSpatialData/Flint2.png`

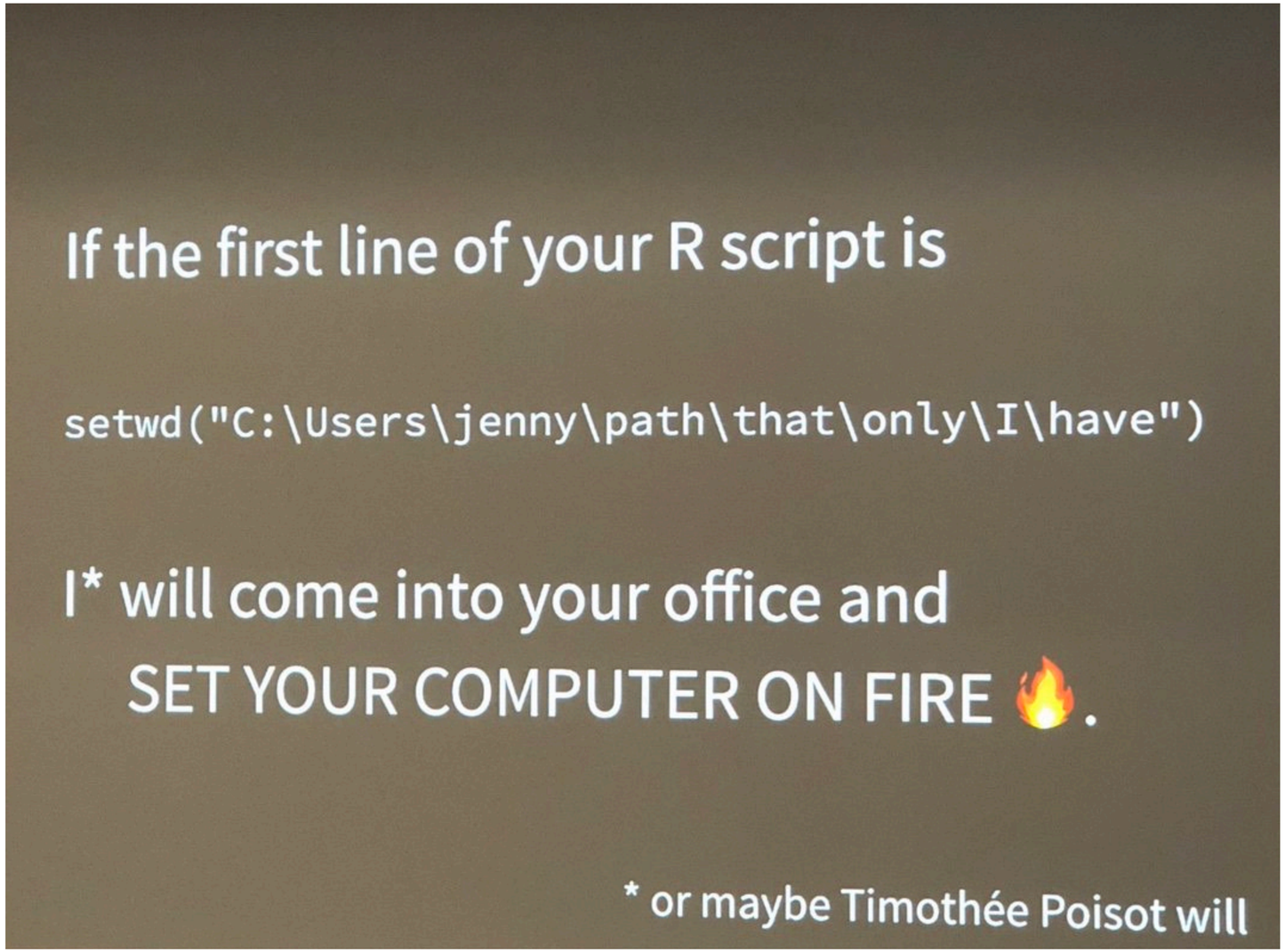
Windows (?): `C:\Dropbox\Documents\MyPresentations\Spatial\WorkingWithSpatialData\Flint2.png`

RStudio will help you with paths. If you start typing a path and hit Tab, it will show you the list of files and folders you can choose from.

“directory” = folder
“working directory” = the folder
your computer is looking in for stuff

```
getwd() # gets the working directory
setwd() # sets the working directory
```

Strong words from Jenny Bryan

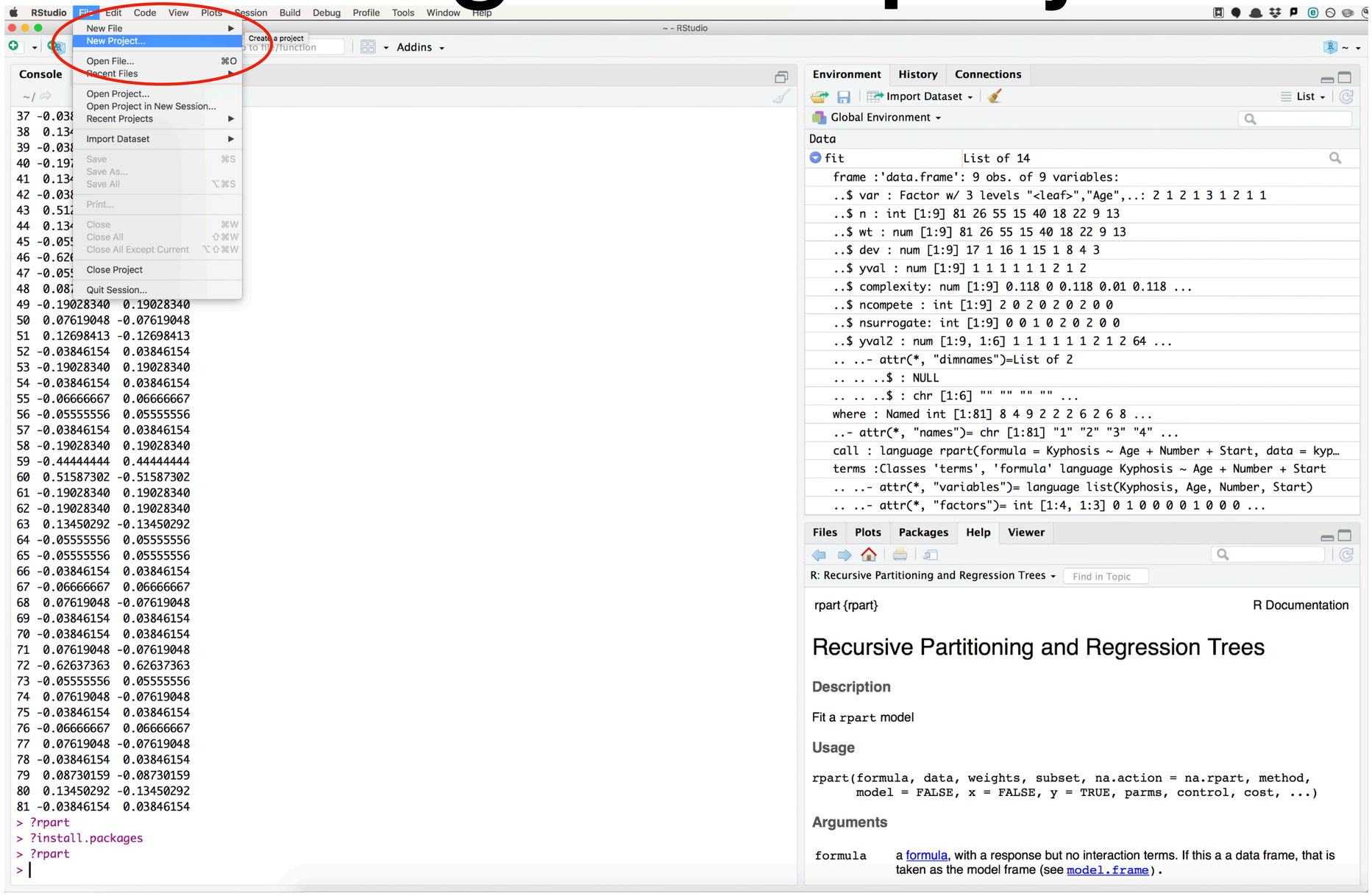


Relative paths are easier and better than absolute paths
That's why we use Projects in RStudio

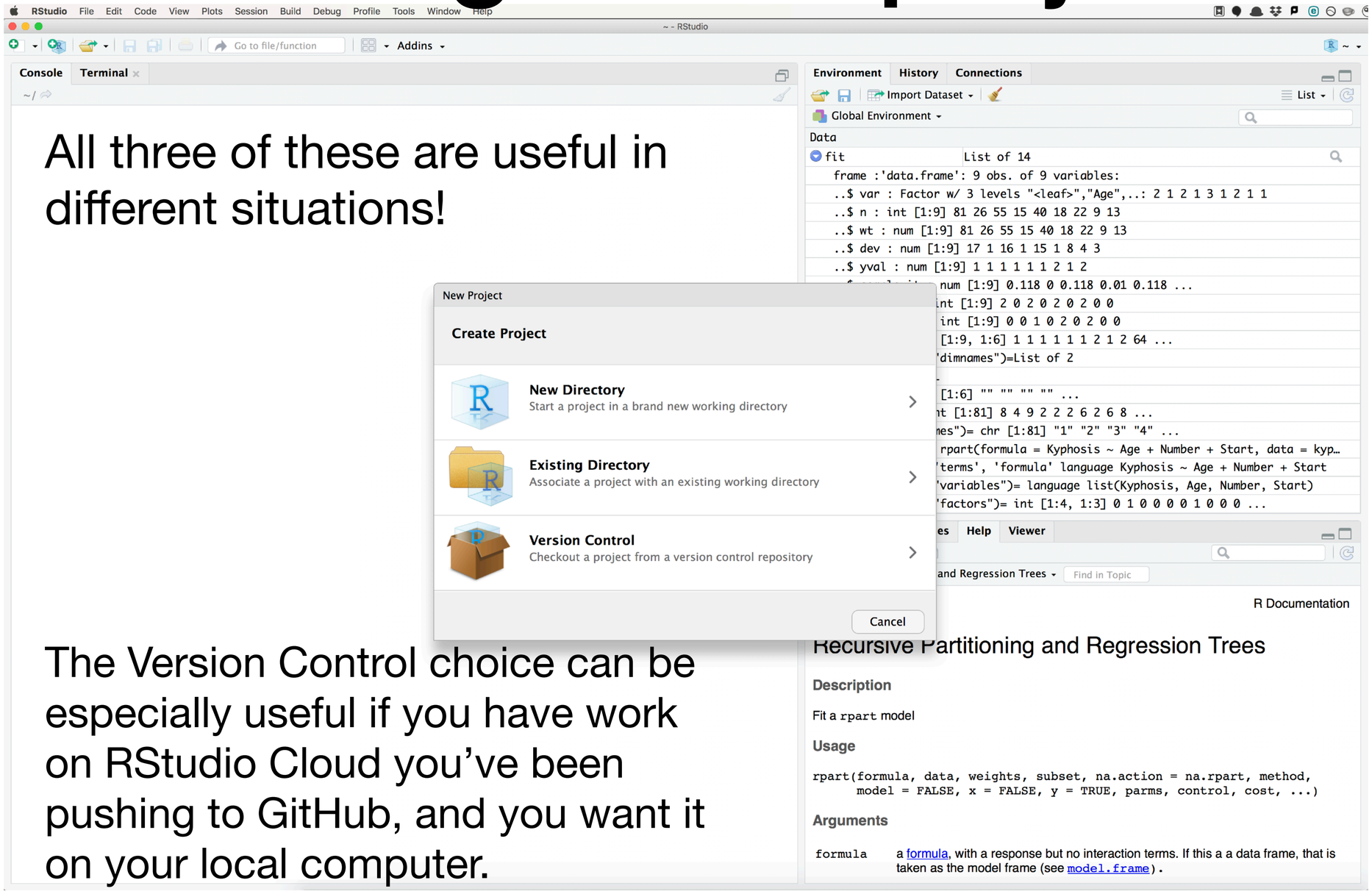
When you make a Project, you're essentially making a special directory that RStudio knows things about. For example, you can set Project-specific settings, and Projects can have different tabs (e.g., the Build tab we saw working with forcats and the git tab we often have)

When you open a Project, RStudio sets your working directory to the folder of the project. After that, you can use relative filepaths.

Making a new project



Making a new project



All three of these are useful in different situations!

The Version Control choice can be especially useful if you have work on RStudio Cloud you've been pushing to GitHub, and you want it on your local computer.

Relative paths are easier and better than absolute paths
That's why we use Projects in RStudio

When you open a Project, RStudio sets your working directory to the folder of the project. After that, you can use relative filepaths.

For example, if I have a Project set up in the directory "/Users/amcnamara/Dropbox/Documents/Projects", then I can use the following relative path

```
read_csv("data/labike.csv")
```

This is better than an absolute path, and if I share that code (e.g., via GitHub), other people with Macs will be able to use my code as-is. But, it still doesn't work for Windows users.

The here package

```
> library(here)

here() starts at /Users/amcnamara/Dropbox/Documents/Projects

> here()

[1] "/Users/amcnamara/Dropbox/Documents/Projects"

> here("data", "labike.csv")

[1] “/Users/amcnamara/Dropbox/Documents/Projects/data/labike.csv"

> read_csv(here("data", "labike.csv"))
```

This will work on Macs and Windows. It will work whether you’ve cloned the GitHub repo into your Downloads folder, your Documents folder, onto your Desktop, etc.

Artwork by @allison_horst

(Wanna be oldschool?)

Use `file.path()`

```
> file.path("data", “labike.csv")
```

```
[1] “data/labike.csv"
```

Not quite as good, but no dependencies because it comes from base R