Welcome Back!

SES 5394: Travel Behavior and Forecasting (Day 5)

File management

Directory structures

REPORT

FILE NOT FOUND

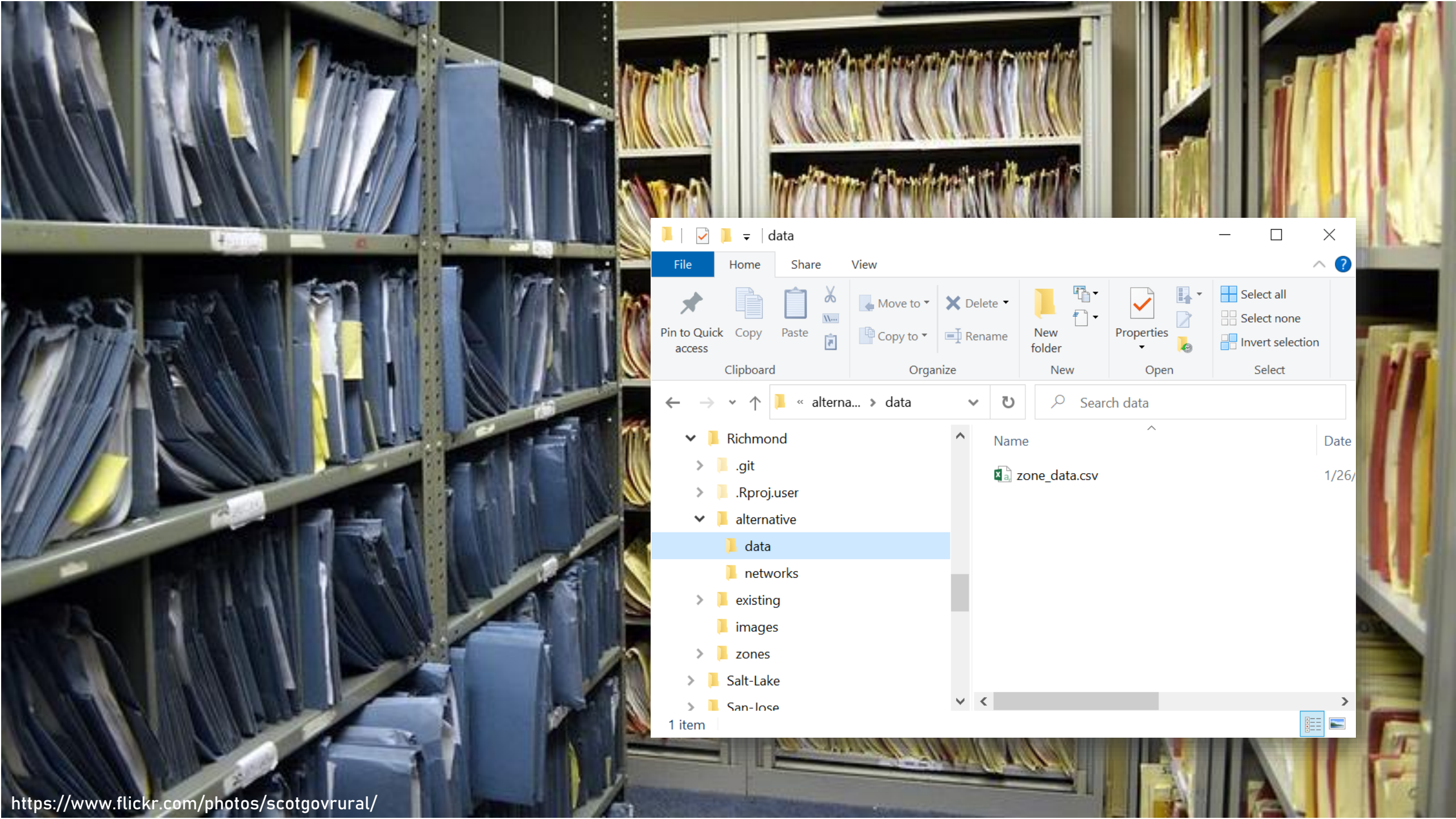
A generation that grew up with Google is forcing professors to rethink their lesson plans

By Monica Chin | @mcsquared96 | Sep 22, 2021, 8:00am EDT

Illustrations by Micha Huigen

https://www.theverge.com/22684730/students-file-folder-directory-structure-education-gen-z

Keep your project files where you can find them…



https://www.flickr.com/photos/scotgovrural/

Referring to file locations in your code

Bad practice: absolute paths

```
zone_data <- read_csv("C:/Users/cav223/Documents/GitHub/Richmond/existing/dat
a/zone_data.csv")
```

This will work on my computer but is more or less guaranteed not to work on anyone else's computer.

Also, long lines like this are hard (for humans) to read.

Good practice: relative paths

```
zone_data <- read_csv("existing/data/zone_data.csv")
```

This will probably work on most computers.

It will stop working if you move the file with the code in it to a different subfolder within your project directory.

Better practice: the here() function

```
zone_data <- read_csv(here("existing/data/zone_data.csv"))
```

This will work regardless of where your code is saved (as long as it's in your project directory).

Nested functions can be hard to read, and they set you up for errors if you can't keep track of the parentheses.

Best practice: Pipe the here() function

```
zone_data <- here("existing/data/zone_data.csv") %>%
  read_csv()
```

This is great.

One problem that could come up is that it's not totally platform independent.

Also, you could still end up with long line lengths if you use a lot of subdirectories with long names.

So beautiful

```
zone_data <- here("existing",
                  "data",
                  "zone_data.csv") %>%
  read_csv()
```

Simple, functional, and readable for both humans and computers.

Speaking of readable code

Documentation…

Documenting your project: README file

github.com/GSD-SES-5394-S2022/Richmond

Apps

a3_skills.html — Added to A3 skills page - added copies of osmconvert — 4 days ago

Environments 1

README.md

Richmond Travel De

This file contains code and data for a simp
as a term project for the course SES 5394
design.

The purpose of this project is to compare

- *Existing:* An approximation of the exis
- *Proposed:* A potential future in which
  housing.

All code is contained in the following RMa

- *Zones.Rmd:* Loads zone boundaries and demographic data from the United States Census Bureau API and

Richmond - main - RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function | Addins

a3_skills.Rmd | README.md

Preview on Save | Preview | Run

```
1  # Richmond Travel Demand model
2
3  This file contains code and data for a simplified travel demand model of the
   Richmond, Virginia region, completed as a term project for the course SES 5394
   (Travel Behavior and Forecasting) at the Harvard Graduate School of design.
4
5  The purpose of this project is to compare accessibility, transit ridership, and
   reagional VMT for two alternatives:
6
7  * *Existing:* An approximation of the existing condition
8  * *Proposed:* A potential future in which a segment of the I-95 freeway is removed
   and replaced with affordable housing.
9
10 All code is contained in the following RMarkdown scripts:
11
12 * *Zones.Rmd:* Loads zone boundaries and demographic data from the United States
```

12:116 Markdown

Richmond

Richmond Travel Demand model

This file contains code and data for a simplified travel demand model of the Richmond, Virginia region, completed as a term project for the course SES 5394 (Travel Behavior and Forecasting) at the Harvard Graduate School of design.

The purpose of this project is to compare accessibility, transit ridership, and reagional VMT for two alternatives:

- *Existing:* An approximation of the existing condition
- *Proposed:* A potential future in which a segment of the I-95 freeway is removed and replaced with affordable housing.

Documenting a file or code chunk: Markdown text

a3_skills.Rmd | Assignment 5.Rmd | example.Rmd

Knit on Save | Knit | Run

```
---
title: "Trip Gereration"
subtitle: "Estimating a Trip Production Model"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

The purpose of this analysis is to estimate a regression model to estimate daily trip productions for each Zone in the Buffalo, New York Metropolitan Statistical Area.

We do this in three steps:

1. Load the household survey data
2. Visualize the household survey data (technically optional, but still a good idea)
3. Estimate a regression model

We use the following libraries:

```{r, message=FALSE, warning = FALSE}
library(httr)
library(knitr)
library(jtools)
library(srvyr)
library(survey)
```
```

~/GitHub/Richmond/example.html

example.html | Open in Browser | Find | Publish

Trip Generation

Estimating a Trip Production Model

The purpose of this analysis is to estimate a regression model to estimate daily trip productions for each Zone in the Buffalo, New York Metropolitan Statistical Area.

We do this in three steps:

1. Load the household survey data
2. Visualize the household survey data (technically optional, but still a good idea)
3. Estimate a regression model

We use the following libraries:

```
library(httr)
library(knitr)
library(jtools)
library(srvyr)
```

Documenting a line of code: Comments

```
# Load the MSA boundaries
boundary <- core_based_statistical_areas() %>%
  filter(GEOID == "40060")

# Define a bounding box containing the MSA
richmond_bbox <- st_bbox(boundary)

q <- opq(bbox = richmond_bbox) %>% # create a query
  add_osm_feature(key = 'highway') %>% # request only road data
  osmdata_xml(file = 'existing/networks/streets.osm') # download osm file
```

When to do your documentation

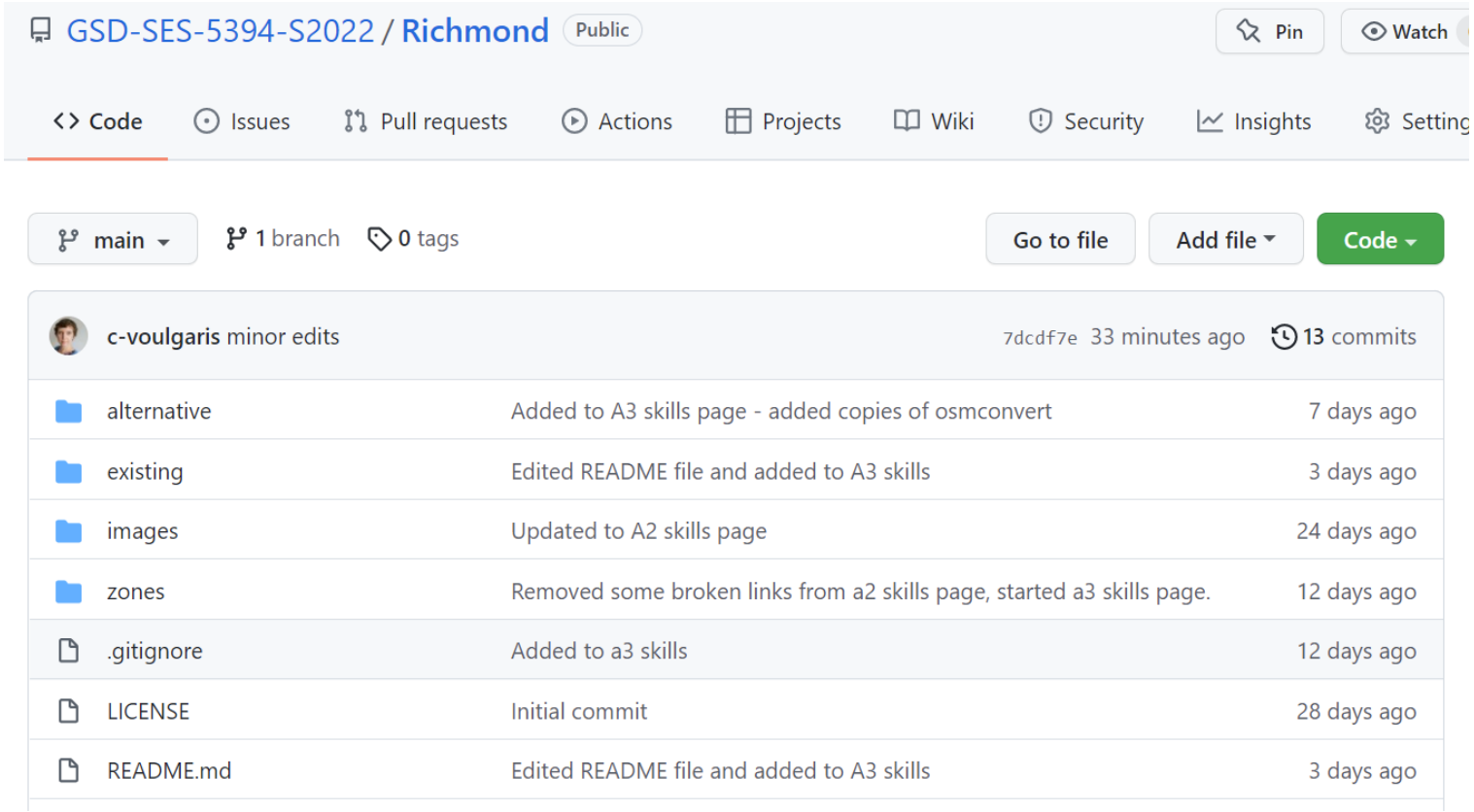
Go back and document your code once it's finished.

Document your code as you go along.

Document your code before you write it.

Documenting a GitHub commit

Commit often enough that you can easily describe the changes you made in a few words.



Speaking of GitHub…

Git workflow

- Pull: Get the most current version of everything from GitHub
- Edit: Do an amount of work you can describe in a few words
- Save: Save changes to your local computer
- Commit: Commit those changes to your local repo
- Push: Update GitHub with the changes you just made

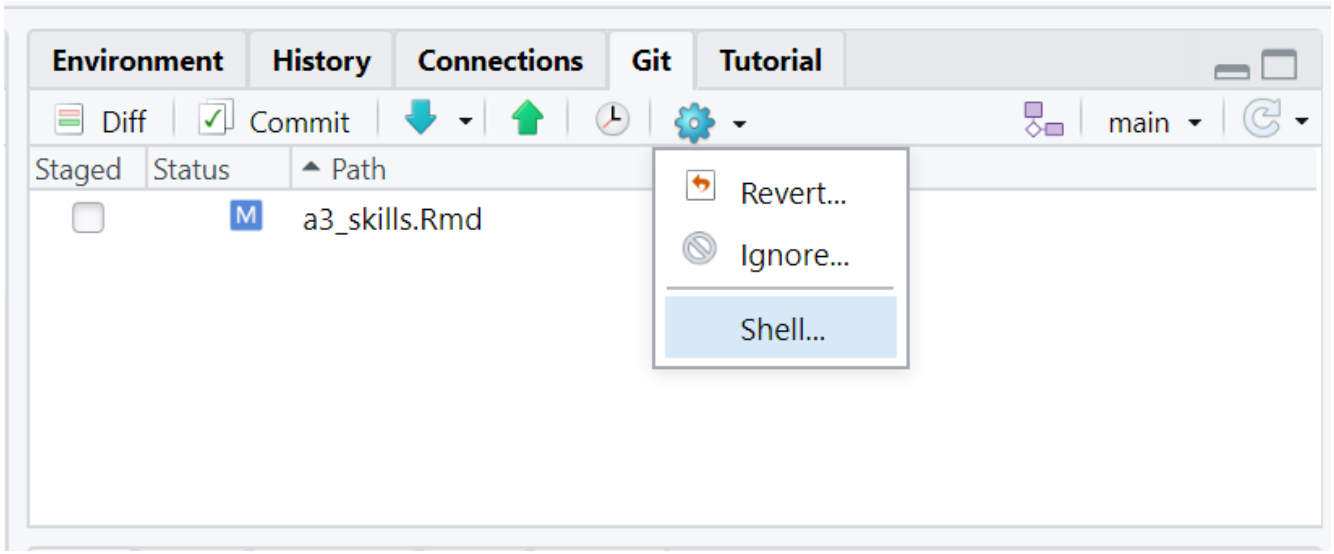
Some Git shell commands (unusual cases)

Un-commit your most recent commit:
`git reset head~`

Un-commit your most recent 3 commits:
`git reset head~3`

Un-commit everything back to a commit called “squiggles”:
`git reset squiggles`

Force the remote (GitHub repo) to update to match your local repo (even if your local is missing commits that are in the remote):
`git push --force`



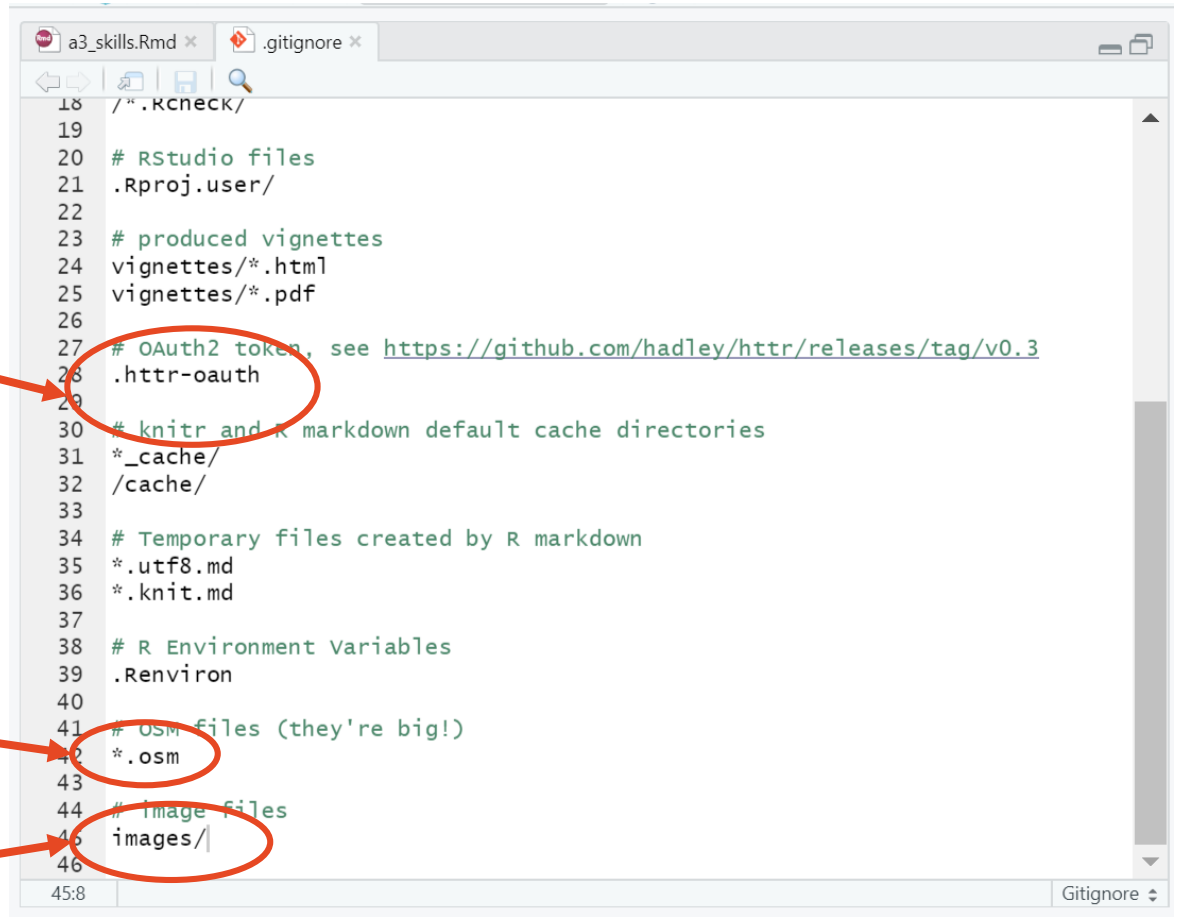
.gitignore

Lists files that Git should ignore

Ignore a file called ".httr-oath"

Ignore files with an osm extension

Ignore files in a folder called images



Analysis Assignment 3

- Create a document template (InDesign or Word) that looks looks attractive, distinctive, and readable.
  - Include headings/chapter titles for the full report and set up a table of contents (refer to the syllabus for chapter titles)
- Write an introduction chapter (it might repurpose most of what you wrote for the project definition, and might not go much further).
- Fill in the README file for your GitHub repo
- The end goal of this project is to determine how your proposal would change
  - Regional VMT
  - Route-level transit ridership
  - Accessibility

Speaking of file management...

If I told you lecture slides were posted on Canvas, where would you look for them?