Homework 3: Databases, web scraping, and a basic Shiny app

Qi Ren

2023-06-01

Table of Contents

# Money in UK politics

[The Westminster Accounts](https://news.sky.com/story/the-westminster-accounts-12786091), a recent collaboration between Sky News and Tortoise Media, examines the flow of money through UK politics. It does so by combining data from three key sources:

1. [Register of Members’ Financial Interests](https://www.parliament.uk/mps-lords-and-offices/standards-and-financial-interests/parliamentary-commissioner-for-standards/registers-of-interests/register-of-members-financial-interests/),
2. [Electoral Commission records of donations to parties](http://search.electoralcommission.org.uk/English/Search/Donations), and
3. [Register of All-Party Parliamentary Groups](https://www.parliament.uk/mps-lords-and-offices/standards-and-financial-interests/parliamentary-commissioner-for-standards/registers-of-interests/register-of-all-party-party-parliamentary-groups/).

You can [search and explore the results](https://news.sky.com/story/westminster-accounts-search-for-your-mp-or-enter-your-full-postcode-12771627) through the collaboration’s interactive database. Simon Willison [has extracted a database](https://til.simonwillison.net/shot-scraper/scraping-flourish) and this is what we will be working with. If you want to read more about [the project’s methodology](https://www.tortoisemedia.com/2023/01/08/the-westminster-accounts-methodology/).

## Open a connection to the database

The database made available by Simon Willison is an SQLite database

sky\_westminster <- DBI::dbConnect(  
 drv = RSQLite::SQLite(),  
 dbname = here::here("data", "sky-westminster-files.db")  
)

How many tables does the database have? ANS: it has 7 tables: appg\_donations, appgs, member\_appgs, members, parties, party\_donations, and payments

DBI::dbListTables(sky\_westminster)

## [1] "appg\_donations" "appgs" "member\_appgs" "members"   
## [5] "parties" "party\_donations" "payments"

## Which MP has received the most amount of money?

ANS: Theresa May

You need to work with the payments and members tables and for now we just want the total among all years. To insert a new, blank chunk of code where you can write your beautiful code (and comments!), please use the following shortcut: Ctrl + Alt + I (Windows) or cmd + option + I (mac)

#set-up table in original database as database objects to be manipulated in R  
payments\_db <- dplyr::tbl(sky\_westminster, "payments")   
members\_db <- dplyr::tbl(sky\_westminster, "members")  
#this extracts an entity, need to use glimpse() to see the categories within  
#from payments\_db, extract member\_id and value, and then sum value for each member\_id  
payments\_db %>%   
 group\_by(member\_id) %>%   
 summarise(total = sum(value, na.rm = TRUE)) %>% #removed NAs while summing   
 #no need to use mutate, summarise is more efficient  
 #left join using payments\_db's category name (member\_id)  
 left\_join(members\_db, by = c("member\_id"="id")) %>%   
 arrange(desc(total)) %>%   
collect()

## # A tibble: 595 × 8  
## member\_id total name gender constituency party\_id short\_name status  
## <chr> <dbl> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 m8 2809765. Theresa May F Maidenhead p4 Mrs May active  
## 2 m1508 2191387. Sir Geoffr… M Torridge an… p4 Sir Geoff… active  
## 3 m1423 1282402 Boris John… M Uxbridge an… p4 Mr Johnson active  
## 4 m4514 799936. Keir Starm… M Holborn and… p15 Mr Starmer active  
## 5 m1211 769373. Andrew Mit… M Sutton Cold… p4 Mr Mitche… active  
## 6 m3958 712321. Fiona Bruce F Congleton p4 Ms Bruce active  
## 7 m14 692438. John Redwo… M Wokingham p4 Mr Redwood active  
## 8 m4483 546043 Rishi Sunak M Richmond (Y… p4 Mr Sunak active  
## 9 m4097 538678. Liz Truss F South West … p4 Ms Truss active  
## 10 m188 441681. Ed Davey M Kingston an… p17 Mr Davey active  
## # ℹ 585 more rows

## Any entity that accounts for more than 5% of all donations?

Is there any entity whose donations account for more than 5% of the total payments given to MPs over the 2020-2022 interval? Who are they and who did they give money to?

ANS: Withers LLP, donated to Geoffrey Cox

payments\_db %>%   
 #add a "total" column for the sum of donations  
 mutate(total = sum(value)) %>%   
 group\_by(entity) %>%   
 #summarise - member id, and proportion of donation in total donations  
 summarise(member\_id, props = (sum(value, na.rm = TRUE))/total) %>%   
 left\_join(members\_db, by = c("member\_id"="id")) %>%   
 arrange(desc(props)) %>%   
collect()

## Warning: Missing values are always removed in SQL aggregation functions.  
## Use `na.rm = TRUE` to silence this warning  
## This warning is displayed once every 8 hours.

## # A tibble: 2,213 × 9  
## entity member\_id props name gender constituency party\_id short\_name status  
## <chr> <chr> <dbl> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 Withe… m1508 0.0525 Sir … M Torridge an… p4 Sir Geoff… active  
## 2 Fiona… m3958 0.0206 Fion… F Congleton p4 Ms Bruce active  
## 3 Charl… m14 0.0195 John… M Wokingham p4 Mr Redwood active  
## 4 Unite m523 0.0184 Nich… M Newcastle u… p8 Mr Brown active  
## 5 Cambr… m8 0.0118 Ther… F Maidenhead p4 Mrs May active  
## 6 GMB U… m523 0.0115 Nich… M Newcastle u… p8 Mr Brown active  
## 7 MPM C… m420 0.0100 Yvet… F Normanton, … p15 Ms Cooper active  
## 8 Cente… m1423 0.00805 Bori… M Uxbridge an… p4 Mr Johnson active  
## 9 Counc… m1423 0.00800 Bori… M Uxbridge an… p4 Mr Johnson active  
## 10 Hindu… m1423 0.00758 Bori… M Uxbridge an… p4 Mr Johnson active  
## # ℹ 2,203 more rows

## Do entity donors give to a single party or not?

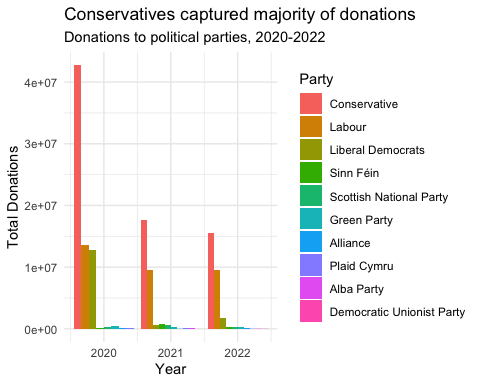
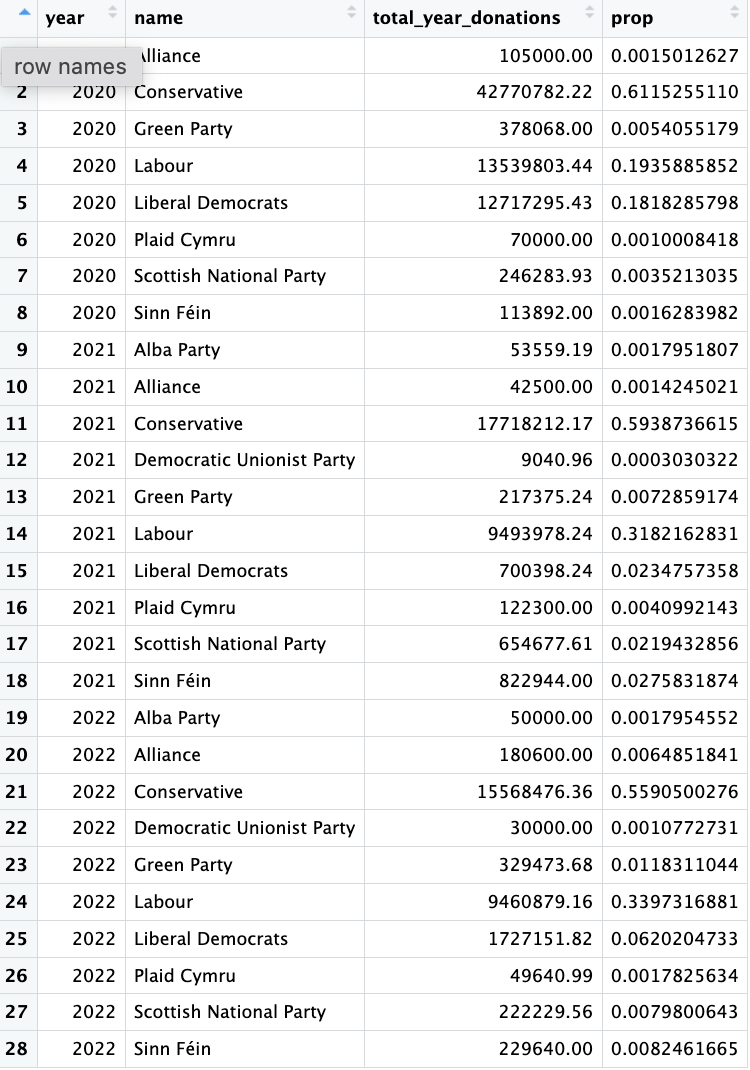
* How many distinct entities who paid money to MPS are there? ANS: 2213 distinct entities
* How many (as a number and %) donated to MPs belonging to a single party only?

all\_entities\_count <- payments\_db %>%   
 group\_by(entity) %>%   
 #summarise member, total donations received, and number of distinct entities  
 summarise(member\_id, total = sum(value, na.rm = TRUE)) %>%  
 filter(total >= 0) %>%   
 summarise(count = n()) %>%   
collect()  
  
single\_parties <- left\_join(payments\_db, members\_db, by = c("member\_id"="id")) %>%   
 #count no. of party\_id per entity  
 group\_by(entity, party\_id) %>%  
 summarise(count = n()) %>%   
 #select entity again to filter out those donated to more than 1 party  
 select(entity) %>%   
 #no. of entities that donated to 1 party only  
 summarise(count = n()) %>%  
 filter(count == 1 ) %>%   
 summarise(single\_party = n()) %>%   
 collect() %>%   
 #proportion of entities donated to 1 party only  
 summarise(prop\_single = single\_party/all\_entities\_count) %>%   
 collect()

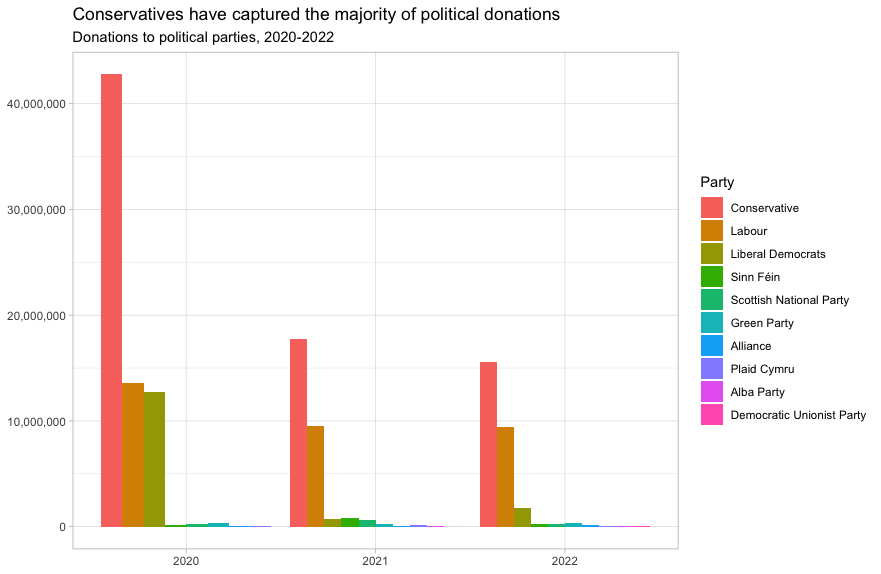
## `summarise()` has grouped output by "entity". You can override using the  
## `.groups` argument.

## Which party has raised the greatest amount of money in each of the years 2020-2022?

I would like you to write code that generates the following table.



… and then, based on this data, plot the following graph.



This uses the default ggplot colour pallete, as I dont want you to worry about using the [official colours for each party](https://en.wikipedia.org/wiki/Wikipedia:Index_of_United_Kingdom_political_parties_meta_attributes). However, I would like you to ensure the parties are sorted according to total donations and not alphabetically. You may even want to remove some of the smaller parties that hardly register on the graph. Would facetting help you?

Finally, when you are done working with the databse, make sure you close the connection, or disconnect from the database.

dbDisconnect(sky\_westminster)

# Anonymised Covid patient data from the CDC

We will be using a dataset with [anonymous Covid-19 patient data that the CDC publishes every month](https://data.cdc.gov/Case-Surveillance/COVID-19-Case-Surveillance-Public-Use-Data-with-Ge/n8mc-b4w4). The file we will use was released on April 11, 2023, and has data on 98 million of patients, with 19 features. This file cannot be loaded in memory, but luckily we have the data in parquet format and we will use the {arrow} package.

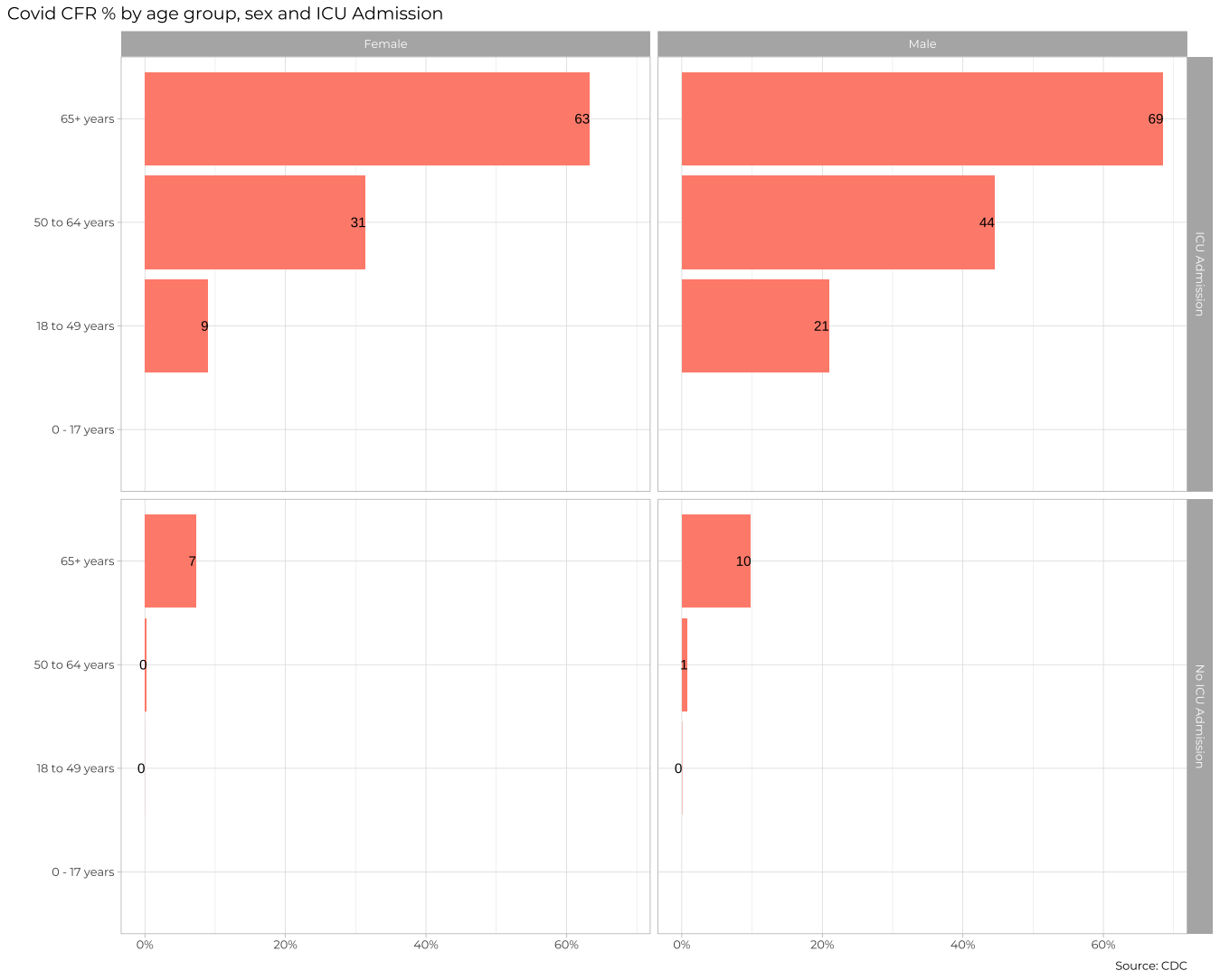
## Obtain the data

The dataset cdc-covid-geography in in parquet format that {arrow}can handle. It is > 600Mb and too large to be hosted on Canvas or Github, so please download it from dropbox <https://www.dropbox.com/sh/q1yk8mmnbbrzavl/AAAxzRtIhag9Nc_hODafGV2ka?dl=0> and save it in your dsb repo, under the data folder

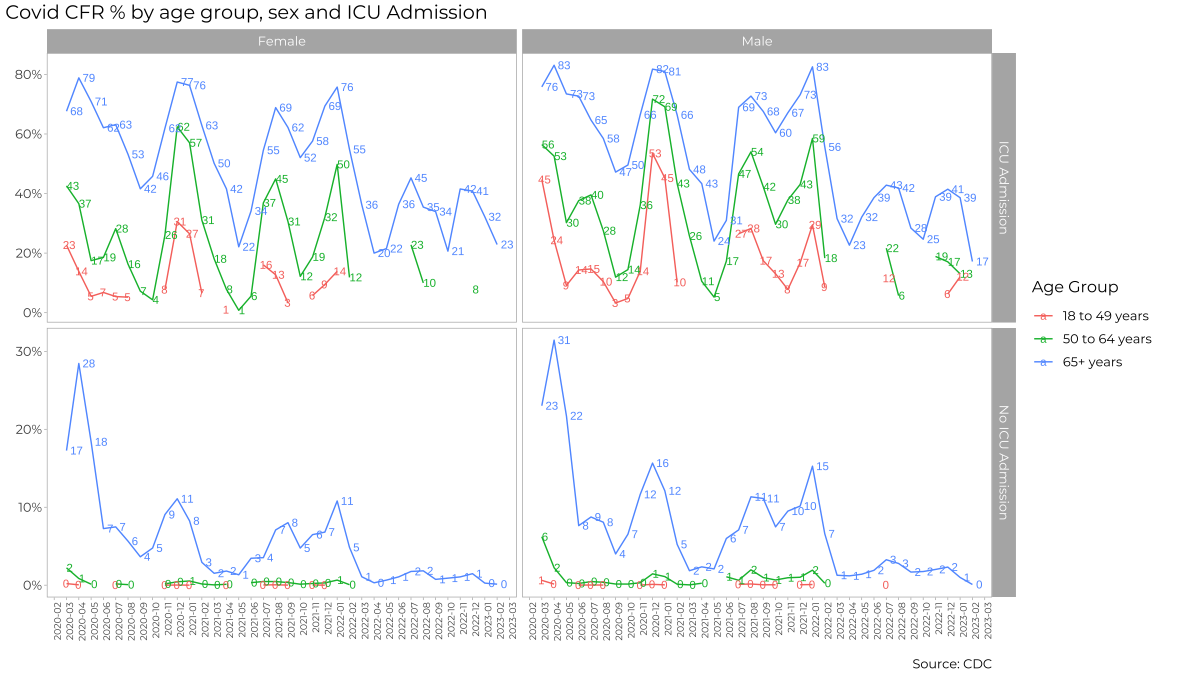
## 0.016 sec elapsed

## FileSystemDataset with 1 Parquet file  
## 97,799,772 rows x 19 columns  
## $ case\_month <string> "2021-09", "2022-09", "2022-01", "2020…  
## $ res\_state <string> "TX", "TX", "TX", "CA", "IL", "CA", "N…  
## $ state\_fips\_code <int32> 48, 48, 48, 6, 17, 6, 36, 36, 36, 53, …  
## $ res\_county <string> "TARRANT", NA, "HARRIS", "SAN BERNARDI…  
## $ county\_fips\_code <int32> 48439, NA, 48201, 6071, 17031, 6085, 3…  
## $ age\_group <string> "18 to 49 years", "18 to 49 years", "1…  
## $ sex <string> "Male", "Male", "Female", "Female", "F…  
## $ race <string> "White", "White", "Unknown", "Asian", …  
## $ ethnicity <string> "Non-Hispanic/Latino", "Non-Hispanic/L…  
## $ case\_positive\_specimen\_interval <int32> NA, NA, NA, NA, 0, NA, 0, 0, 0, 0, 0, …  
## $ case\_onset\_interval <int32> NA, NA, -1, NA, 0, NA, NA, NA, NA, 0, …  
## $ process <string> "Missing", "Missing", "Missing", "Miss…  
## $ exposure\_yn <string> "Missing", "Missing", "Missing", "Miss…  
## $ current\_status <string> "Laboratory-confirmed case", "Probable…  
## $ symptom\_status <string> "Missing", "Missing", "Symptomatic", "…  
## $ hosp\_yn <string> "Missing", "Missing", "No", "No", "No"…  
## $ icu\_yn <string> "Missing", "Missing", "Missing", "Miss…  
## $ death\_yn <string> "Missing", "Missing", "Missing", "Miss…  
## $ underlying\_conditions\_yn <string> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA…

Can you query the database and replicate the following plot?



The previous plot is an aggregate plot for all three years of data. What if we wanted to plot Case Fatality Ratio (CFR) over time? Write code that collects the relevant data from the database and plots the following



For each patient, the dataframe also lists the patient’s states and county [FIPS code](https://en.wikipedia.org/wiki/Federal_Information_Processing_Standard_state_code). The CDC also has information on the [NCHS Urban-Rural classification scheme for counties](https://www.cdc.gov/nchs/data_access/urban_rural.htm)

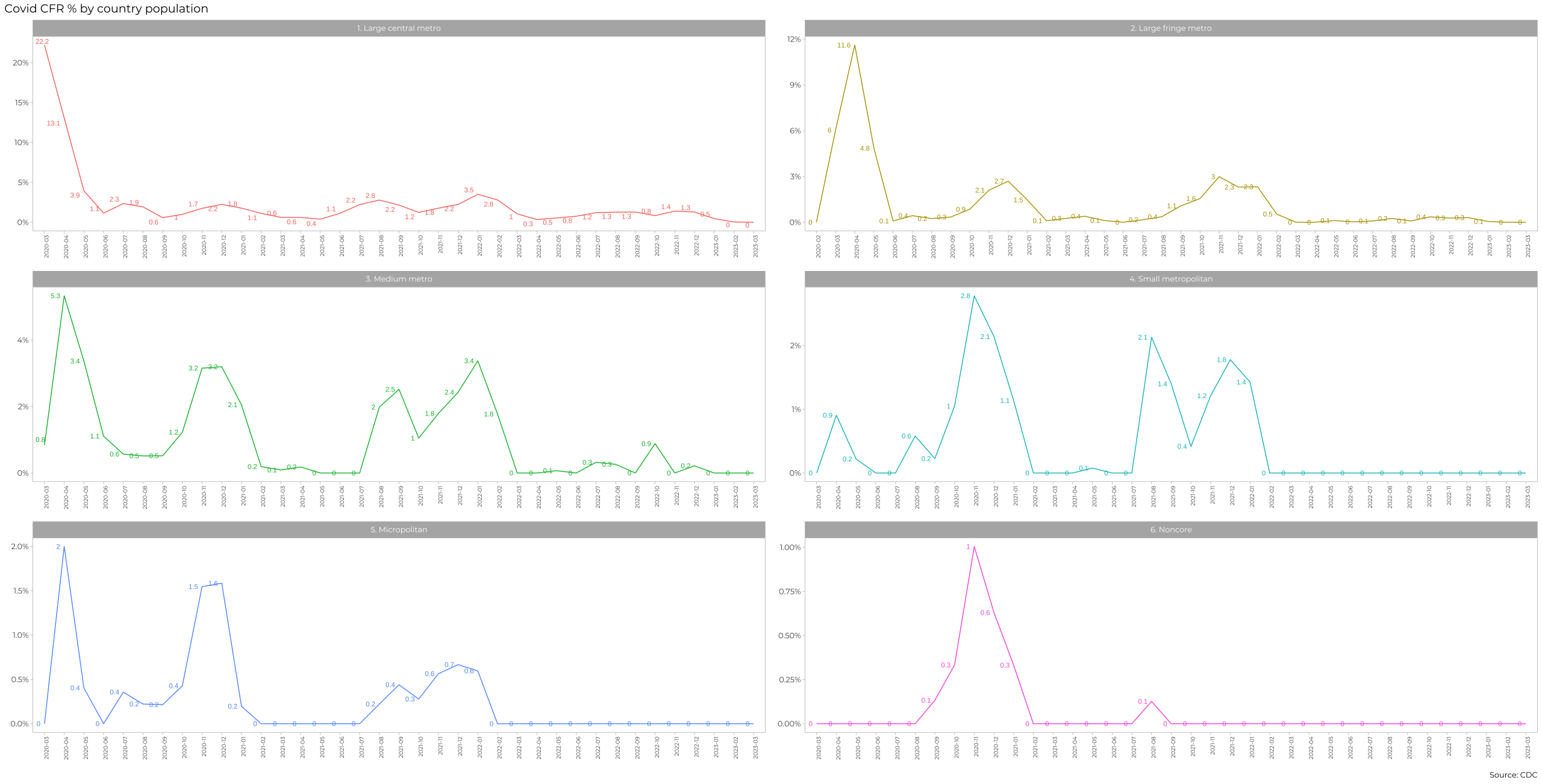
urban\_rural <- read\_xlsx(here::here("data", "NCHSURCodes2013.xlsx")) %>%   
 janitor::clean\_names()

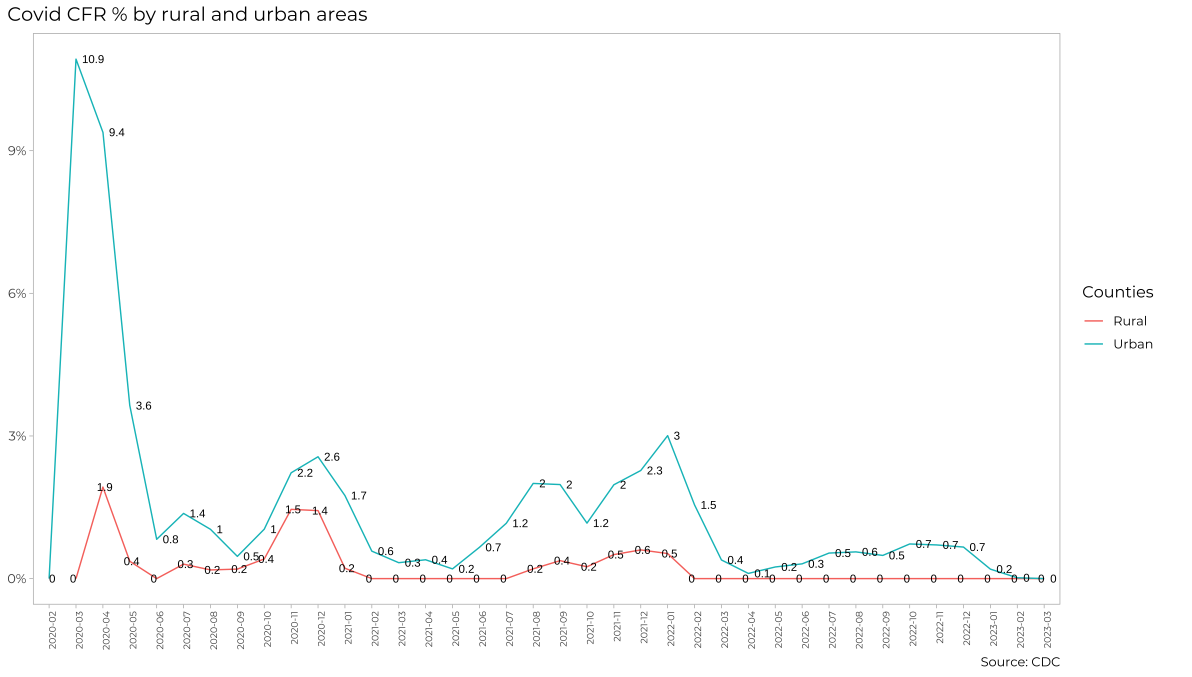
Each county belongs in seix diffent categoreis, with categories 1-4 being urban areas and categories 5-6 being rural, according to the following criteria captured in x2013\_code

Category name

1. Large central metro - 1 million or more population and contains the entire population of the largest principal city
2. large fringe metro - 1 million or more poulation, but does not qualify as 1
3. Medium metro - 250K - 1 million population
4. Small metropolitan population < 250K
5. Micropolitan
6. Noncore

Can you query the database, extract the relevant information, and reproduce the following two graphs that look at the Case Fatality ratio (CFR) in different counties, according to their population?





# Money in US politics

In the United States, [*“only American citizens (and immigrants with green cards) can contribute to federal politics, but the American divisions of foreign companies can form political action committees (PACs) and collect contributions from their American employees.”*](https://www.opensecrets.org/political-action-committees-pacs/foreign-connected-pacs)

We will scrape and work with data foreign connected PACs that donate to US political campaigns. The data for foreign connected PAC contributions in the 2022 election cycle can be found at <https://www.opensecrets.org/political-action-committees-pacs/foreign-connected-pacs/2022>. Then, we will use a similar approach to get data such contributions from previous years so that we can examine trends over time.

All data come from [OpenSecrets.org](https://www.opensecrets.org), a *“website tracking the influence of money on U.S. politics, and how that money affects policy and citizens’ lives”*.

library(robotstxt)  
paths\_allowed("https://www.opensecrets.org")

## [1] TRUE

base\_url <- "https://www.opensecrets.org/political-action-committees-pacs/foreign-connected-pacs/2022"  
  
contributions\_tables <- base\_url %>%  
 read\_html()

* First, make sure you can scrape the data for 2022. Use janitor::clean\_names() to rename variables scraped using snake\_case naming.
* Clean the data:
  + Write a function that converts contribution amounts in total, dems, and repubs from character strings to numeric values.
  + Separate the country\_of\_origin\_parent\_company into two such that country and parent company appear in different columns for country-level analysis.
* Write a function called scrape\_pac() that scrapes information from the Open Secrets webpage for foreign-connected PAC contributions in a given year. This function should
  + have one input: the URL of the webpage and should return a data frame.
  + add a new column to the data frame for year. We will want this information when we ultimately have data from all years, so this is a good time to keep track of it. Our function doesn’t take a year argument, but the year is embedded in the URL, so we can extract it out of there, and add it as a new column. Use the str\_sub() function to extract the last 4 characters from the URL. You will probably want to look at the help for this function to figure out how to specify “last 4 characters”.
* Define the URLs for 2022, 2020, and 2000 contributions. Then, test your function using these URLs as inputs. Does the function seem to do what you expected it to do?
* Construct a vector called urls that contains the URLs for each webpage that contains information on foreign-connected PAC contributions for a given year.
* Map the scrape\_pac() function over urls in a way that will result in a data frame called contributions\_all.
* Write the data frame to a csv file called contributions-all.csv in the data folder.

# Scraping consulting jobs

The website [https://www.consultancy.uk/jobs/](https://www.consultancy.uk/jobs) lists job openings for consulting jobs.

library(robotstxt)  
paths\_allowed("https://www.consultancy.uk") #is it ok to scrape?  
  
base\_url <- "https://www.consultancy.uk/jobs/page/1"  
  
listings\_html <- base\_url %>%  
 read\_html()

Identify the CSS selectors in order to extract the relevant information from this page, namely

1. job
2. firm
3. functional area
4. type

Can you get all pages of ads, and not just the first one, https://www.consultancy.uk/jobs/page/1 into a dataframe?

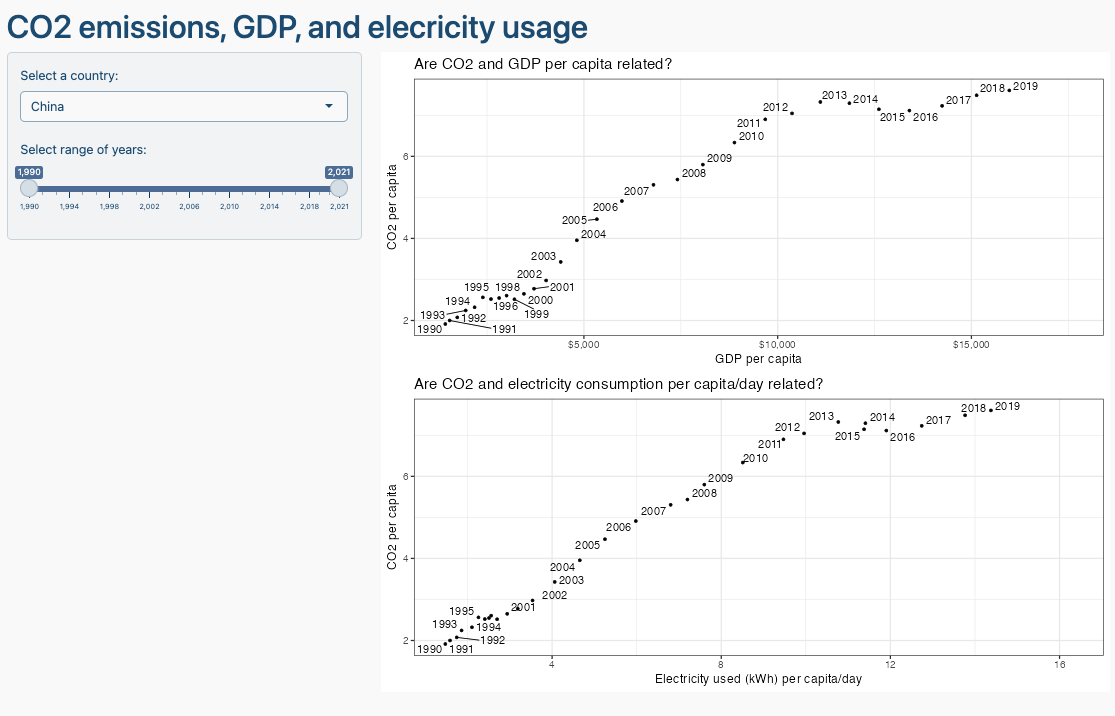
* Write a function called scrape\_jobs() that scrapes information from the webpage for consulting positions. This function should
  + have one input: the URL of the webpage and should return a data frame with four columns (variables): job, firm, functional area, and type
  + Test your function works with other pages too, e.g., <https://www.consultancy.uk/jobs/page/2>. Does the function seem to do what you expected it to do?
  + Given that you have to scrape ...jobs/page/1, ...jobs/page/2, etc., define your URL so you can join multiple stings into one string, using str\_c(). For instnace, if page is 5, what do you expect the following code to produce?

base\_url <- "https://www.consultancy.uk/jobs/page/1"  
url <- str\_c(base\_url, page)

* Construct a vector called pages that contains the numbers for each page available
* Map the scrape\_jobs() function over pages in a way that will result in a data frame called all\_consulting\_jobs.
* Write the data frame to a csv file called all\_consulting\_jobs.csv in the data folder.

# Create a shiny app - OPTIONAL

We have already worked with the data on electricity production and usage, GDP/capita and CO2/capita since 1990. You have to create a simple Shiny app, where a user chooses a country from a drop down list and a time interval between 1990 and 2020 and shiny outputs the following



You can use chatGPT to get the basic layout of Shiny app, but you need to adjust the code it gives you. Ask chatGPT to create the Shiny app using the gapminder data and make up similar requests for the inputs/outpus you are thinking of deploying.

# Deliverables

There is a lot of explanatory text, comments, etc. You do not need these, so delete them and produce a stand-alone document that you could share with someone. Knit the edited and completed R Markdown (Rmd) file as a Word or HTML document (use the “Knit” button at the top of the script editor window) and upload it to Canvas. You must be commiting and pushing your changes to your own Github repo as you go along.

# Details

* Who did you collaborate with: TYPE NAMES HERE
* Approximately how much time did you spend on this problem set: ANSWER HERE
* What, if anything, gave you the most trouble: ANSWER HERE

**Please seek out help when you need it,** and remember the [15-minute rule](https://dsb2023.netlify.app/syllabus/#the-15-minute-rule). You know enough R (and have enough examples of code from class and your readings) to be able to do this. If you get stuck, ask for help from others, post a question on Slack– and remember that I am here to help too!

As a true test to yourself, do you understand the code you submitted and are you able to explain it to someone else?

# Rubric

13/13: Problem set is 100% completed. Every question was attempted and answered, and most answers are correct. Code is well-documented (both self-documented and with additional comments as necessary). Used tidyverse, instead of base R. Graphs and tables are properly labelled. Analysis is clear and easy to follow, either because graphs are labeled clearly or you’ve written additional text to describe how you interpret the output. Multiple Github commits. Work is exceptional. I will not assign these often.

8/13: Problem set is 60–80% complete and most answers are correct. This is the expected level of performance. Solid effort. Hits all the elements. No clear mistakes. Easy to follow (both the code and the output). A few Github commits.

5/13: Problem set is less than 60% complete and/or most answers are incorrect. This indicates that you need to improve next time. I will hopefully not assign these often. Displays minimal effort. Doesn’t complete all components. Code is poorly written and not documented. Uses the same type of plot for each graph, or doesn’t use plots appropriate for the variables being analyzed. No Github commits.