Homework 2

Your name goes here

2023-05-22

Table of Contents

# Data Visualisation - Exploration

Now that you’ve demonstrated your software is setup, and you have the basics of data manipulation, the goal of this assignment is to practice transforming, visualising, and exploring data.

# Mass shootings in the US

In July 2012, in the aftermath of a mass shooting in a movie theater in Aurora, Colorado, [Mother Jones](https://www.motherjones.com/politics/2012/07/mass-shootings-map/) published a report on mass shootings in the United States since 1982. Importantly, they provided the underlying data set as [an open-source database](https://www.motherjones.com/politics/2012/12/mass-shootings-mother-jones-full-data/) for anyone interested in studying and understanding this criminal behavior.

## Obtain the data

## Rows: 125  
## Columns: 14  
## $ case <chr> "Oxford High School shooting", "San Jose VTA shoo…  
## $ year <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2020, 2020, 2…  
## $ month <chr> "Nov", "May", "Apr", "Mar", "Mar", "Mar", "Mar", …  
## $ day <dbl> 30, 26, 15, 31, 22, 16, 16, 26, 10, 6, 31, 4, 3, …  
## $ location <chr> "Oxford, Michigan", "San Jose, California", "Indi…  
## $ summary <chr> "Ethan Crumbley, a 15-year-old student at Oxford …  
## $ fatalities <dbl> 4, 9, 8, 4, 10, 8, 4, 5, 4, 3, 7, 9, 22, 3, 12, 5…  
## $ injured <dbl> 7, 0, 7, 1, 0, 1, 0, 0, 3, 8, 25, 27, 26, 12, 4, …  
## $ total\_victims <dbl> 11, 9, 15, 5, 10, 9, 4, 5, 7, 11, 32, 36, 48, 15,…  
## $ location\_type <chr> "School", "Workplace", "Workplace", "Workplace", …  
## $ male <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, T…  
## $ age\_of\_shooter <dbl> 15, 57, 19, NA, 21, 21, 31, 51, NA, NA, 36, 24, 2…  
## $ race <chr> NA, NA, "White", NA, NA, "White", NA, "Black", "B…  
## $ prior\_mental\_illness <chr> NA, "Yes", "Yes", NA, "Yes", NA, NA, NA, NA, NA, …

| column(variable) | description |
| --- | --- |
| case | short name of incident |
| year, month, day | year, month, day in which the shooting occurred |
| location | city and state where the shooting occcurred |
| summary | brief description of the incident |
| fatalities | Number of fatalities in the incident, excluding the shooter |
| injured | Number of injured, non-fatal victims in the incident, excluding the shooter |
| total\_victims | number of total victims in the incident, excluding the shooter |
| location\_type | generic location in which the shooting took place |
| male | logical value, indicating whether the shooter was male |
| age\_of\_shooter | age of the shooter when the incident occured |
| race | race of the shooter |
| prior\_mental\_illness | did the shooter show evidence of mental illness prior to the incident? |

## Explore the data

### Specific questions

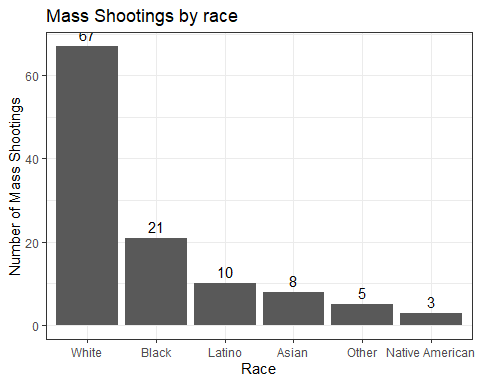
* Generate a data frame that summarizes the number of mass shootings per year.

mass\_shootings %>%  
 group\_by(year) %>%  
 summarise(count=n())

## # A tibble: 37 × 2  
## year count  
## <dbl> <int>  
## 1 1982 1  
## 2 1984 2  
## 3 1986 1  
## 4 1987 1  
## 5 1988 1  
## 6 1989 2  
## 7 1990 1  
## 8 1991 3  
## 9 1992 2  
## 10 1993 4  
## # ℹ 27 more rows

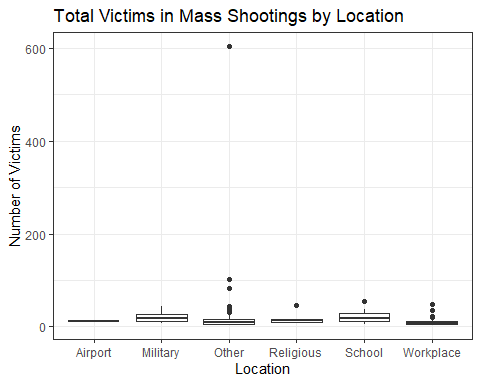
* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

race\_mass\_shootings<-mass\_shootings %>%  
 group\_by(race) %>%   
 summarise(count=n()) %>%   
 arrange(desc(count)) %>%   
 drop\_na() %>%   
 ggplot(aes(x=reorder(race,-count),y=count))+  
 geom\_col()+   
 geom\_text(aes(label=count),vjust=-0.5) +  
 labs(title = "Mass Shootings by race",  
 x = "Race", y="Number of Mass Shootings") +  
 theme\_bw()  
  
print(race\_mass\_shootings)



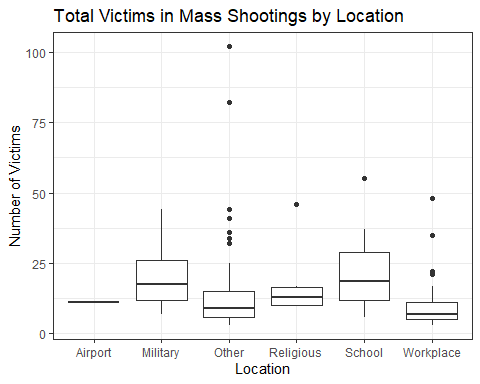
* Generate a boxplot visualizing the number of total victims, by type of location.

victims\_location\_mass\_shootings<-mass\_shootings %>%   
 group\_by(location\_type) %>%   
 ggplot(aes(x=location\_type,y=total\_victims))+geom\_boxplot()+  
labs(title = "Total Victims in Mass Shootings by Location ",  
 x = "Location", y="Number of Victims") +  
 theme\_bw()   
  
print(victims\_location\_mass\_shootings)



* Redraw the same plot, but remove the Las Vegas Strip massacre from the dataset.

victims\_location\_mass\_shootings\_excl<-  
 mass\_shootings %>%   
 filter(case != "Las Vegas Strip massacre") %>%   
 group\_by(location\_type) %>%   
 ggplot(aes(x=location\_type,y=total\_victims))+geom\_boxplot()+  
labs(title = "Total Victims in Mass Shootings by Location ",  
 x = "Location", y="Number of Victims") +  
 theme\_bw()  
  
print(victims\_location\_mass\_shootings\_excl)



### More open-ended questions

Address the following questions. Generate appropriate figures/tables to support your conclusions.

* How many white males with prior signs of mental illness initiated a mass shooting after 2000?

mass\_shootings %>%  
 filter(year > 2000, race == "White", prior\_mental\_illness =="Yes") %>%   
 summarise(count=n())

## # A tibble: 1 × 1  
## count  
## <int>  
## 1 23

#The number of white males with prior signs of mental illness initiated a mass shooting after 2000 is 23

* Which month of the year has the most mass shootings? Generate a bar chart sorted in chronological (natural) order (Jan-Feb-Mar- etc) to provide evidence of your answer.

month\_order<-c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")  
  
mass\_shootings%>%   
 mutate(month = factor(month, levels=month\_order)) %>%   
 group\_by(month) %>%   
 summarise(count=n())

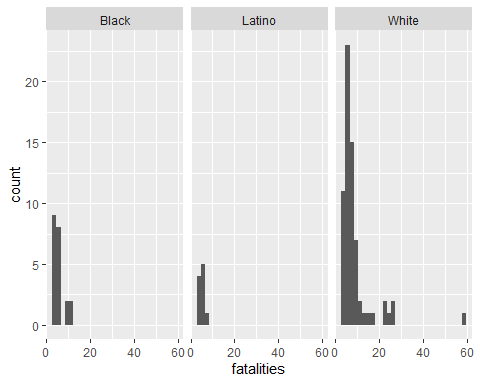
## # A tibble: 12 × 2  
## month count  
## <fct> <int>  
## 1 Jan 7  
## 2 Feb 13  
## 3 Mar 12  
## 4 Apr 11  
## 5 May 8  
## 6 Jun 12  
## 7 Jul 10  
## 8 Aug 8  
## 9 Sep 10  
## 10 Oct 11  
## 11 Nov 12  
## 12 Dec 11

#convert month into factor first which has category and then order

* How does the distribution of mass shooting fatalities differ between White and Black shooters? What about White and Latino shooters?

mass\_shootings%>%   
 filter(race %in% c("White","Black","Latino")) %>%   
 group\_by(race) %>%   
 ggplot(aes(x=fatalities)) +  
 geom\_histogram() +  
 facet\_wrap(~race)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



### Very open-ended

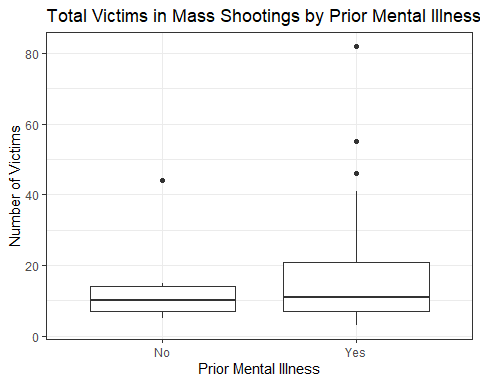
* Are mass shootings with shooters suffering from mental illness different from mass shootings with no signs of mental illness in the shooter?

mass\_shootings%>%   
 group\_by(prior\_mental\_illness) %>%   
 summarize(count=n())

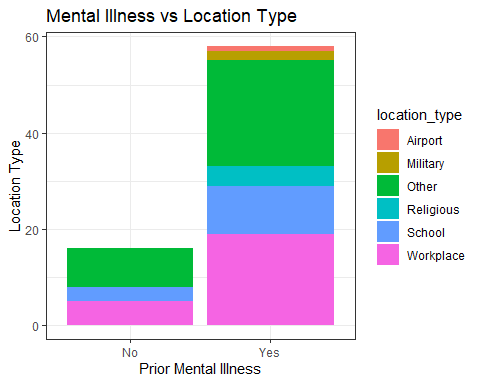
## # A tibble: 3 × 2  
## prior\_mental\_illness count  
## <chr> <int>  
## 1 No 17  
## 2 Yes 62  
## 3 <NA> 46

* Assess the relationship between mental illness and total victims, mental illness and location type, and the intersection of all three variables.

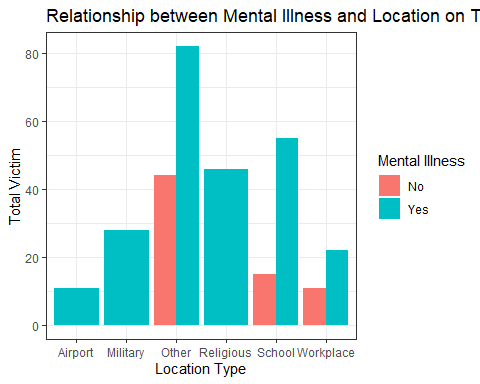
mass\_shootings%>%   
 group\_by(prior\_mental\_illness) %>%   
 drop\_na() %>%   
 ggplot(aes(x=prior\_mental\_illness,y=total\_victims))+geom\_boxplot()+  
labs(title = "Total Victims in Mass Shootings by Prior Mental Illness ",  
 x = "Prior Mental Illness", y="Number of Victims") +  
 theme\_bw()



mass\_shootings%>%   
 group\_by(prior\_mental\_illness) %>%   
 drop\_na() %>%   
 ggplot(aes(x=prior\_mental\_illness,fill=location\_type))+geom\_bar()+  
labs(title = "Mental Illness vs Location Type ",  
 x = "Prior Mental Illness", y="Location Type") +  
 theme\_bw()



mass\_shootings%>%   
 group\_by(prior\_mental\_illness,location\_type) %>%   
 drop\_na() %>%   
 ggplot(aes(x=location\_type,y=total\_victims,fill=prior\_mental\_illness))+geom\_bar(stat="identity",position="dodge")+  
labs(title = "Relationship between Mental Illness and Location on Total Number of Victims ",  
 x = "Location Type", y="Total Victim", fill="Mental Illness") +  
 theme\_bw()



Make sure to provide a couple of sentences of written interpretation of your tables/figures. Graphs and tables alone will not be sufficient to answer this question.

# Exploring credit card fraud

We will be using a dataset with credit card transactions containing legitimate and fraud transactions. Fraud is typically well below 1% of all transactions, so a naive model that predicts that all transactions are legitimate and not fraudulent would have an accuracy of well over 99%– pretty good, no? (well, not quite as we will see later in the course)

You can read more on credit card fraud on [Credit Card Fraud Detection Using Weighted Support Vector Machine](https://www.scirp.org/journal/paperinformation.aspx?paperid=105944)

The dataset we will use consists of credit card transactions and it includes information about each transaction including customer details, the merchant and category of purchase, and whether or not the transaction was a fraud.

## Obtain the data

The dataset is 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

## Rows: 671,028  
## Columns: 14  
## $ trans\_date\_trans\_time <dttm> 2019-02-22 07:32:58, 2019-02-16 15:07:20, 2019-…  
## $ trans\_year <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2020, …  
## $ category <chr> "entertainment", "kids\_pets", "personal\_care", "…  
## $ amt <dbl> 7.79, 3.89, 8.43, 40.00, 54.04, 95.61, 64.95, 3.…  
## $ city <chr> "Veedersburg", "Holloway", "Arnold", "Apison", "…  
## $ state <chr> "IN", "OH", "MO", "TN", "CO", "GA", "MN", "AL", …  
## $ lat <dbl> 40.1186, 40.0113, 38.4305, 35.0149, 39.4584, 32.…  
## $ long <dbl> -87.2602, -80.9701, -90.3870, -85.0164, -106.385…  
## $ city\_pop <dbl> 4049, 128, 35439, 3730, 277, 1841, 136, 190178, …  
## $ job <chr> "Development worker, community", "Child psychoth…  
## $ dob <chr> "19/10/1959", "3/4/1946", "31/3/1985", "28/1/199…  
## $ merch\_lat <dbl> 39.41679, 39.74585, 37.73078, 34.53277, 39.95244…  
## $ merch\_long <dbl> -87.52619, -81.52477, -91.36875, -84.10676, -106…  
## $ is\_fraud <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

The data dictionary is as follows

| column(variable) | description |
| --- | --- |
| trans\_date\_trans\_time | Transaction DateTime |
| trans\_year | Transaction year |
| category | category of merchant |
| amt | amount of transaction |
| city | City of card holder |
| state | State of card holder |
| lat | Latitude location of purchase |
| long | Longitude location of purchase |
| city\_pop | card holder’s city population |
| job | job of card holder |
| dob | date of birth of card holder |
| merch\_lat | Latitude Location of Merchant |
| merch\_long | Longitude Location of Merchant |
| is\_fraud | Whether Transaction is Fraud (1) or Not (0) |

* In this dataset, how likely are fraudulent transactions? Generate a table that summarizes the number and frequency of fraudulent transactions per year.

card\_fraud %>%  
 group\_by(trans\_year) %>%   
 summarise(count=n())

## # A tibble: 2 × 2  
## trans\_year count  
## <dbl> <int>  
## 1 2019 478646  
## 2 2020 192382

* How much money (in US$ terms) are fraudulent transactions costing the company? Generate a table that summarizes the total amount of legitimate and fraudulent transactions per year and calculate the % of fraudulent transactions, in US$ terms.

fraudulent\_transaction\_cost<-card\_fraud %>%  
 mutate(is\_fraud = case\_when(is\_fraud == 0 ~ "Legitimate", is\_fraud == 1 ~ "Fraudulent")) %>%  
 group\_by(trans\_year, is\_fraud) %>%  
 summarise(total\_amount = sum(amt),.groups='keep') %>%  
 pivot\_wider(names\_from = is\_fraud, values\_from = total\_amount, values\_fill = 0) %>%  
 mutate(fraud\_percentage = Fraudulent / (Legitimate + Fraudulent) \* 100)  
  
print(fraudulent\_transaction\_cost)

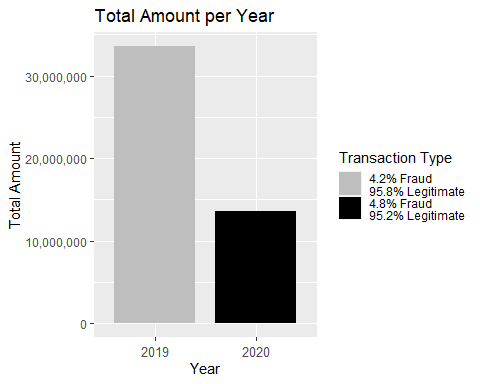
## # A tibble: 2 × 4  
## # Groups: trans\_year [2]  
## trans\_year Fraudulent Legitimate fraud\_percentage  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 1423140. 32182901. 4.23  
## 2 2020 651949. 12925914. 4.80

* Generate a histogram that shows the distribution of amounts charged to credit card, both for legitimate and fraudulent accounts. Also, for both types of transactions, calculate some quick summary statistics.

summary\_table <- card\_fraud %>%  
 group\_by(trans\_year, is\_fraud) %>%  
 summarize(total\_amt = sum(amt))

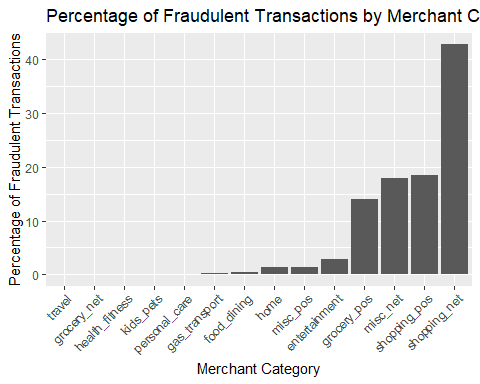
## `summarise()` has grouped output by 'trans\_year'. You can override using the  
## `.groups` argument.

# total fraud  
fraud\_sum <- summary\_table %>%  
 filter(is\_fraud == 1) %>%  
 mutate(fraud\_amt = total\_amt) %>%  
 select(trans\_year, fraud\_amt)  
  
# total legit  
legitimate\_sum <- summary\_table %>%  
 filter(is\_fraud == 0) %>%  
 mutate(legitimate\_amt = total\_amt) %>%  
 select(trans\_year, legitimate\_amt)  
   
fraud\_percentage <- fraud\_sum %>%  
 left\_join(legitimate\_sum, by = "trans\_year") %>%  
 mutate(percentage\_fraud = (fraud\_amt / (fraud\_amt + legitimate\_amt)) \* 100, percentage\_legitimate = (legitimate\_amt / (fraud\_amt + legitimate\_amt)) \* 100) %>%  
 select(trans\_year, fraud\_amt, legitimate\_amt, percentage\_fraud, percentage\_legitimate) %>%   
   
ggplot(aes(x = factor(trans\_year), y = fraud\_amt + legitimate\_amt)) +  
 geom\_col(aes(fill = paste0(sprintf("%.1f", round(percentage\_fraud, 1)), "% Fraud\n", sprintf("%.1f", round(percentage\_legitimate, 1)), "% Legitimate")), width = 0.8) +  
 labs(title = "Total Amount per Year", x = "Year", y = "Total Amount") +  
 scale\_fill\_manual(values = c("grey", "black"), name = "Transaction Type") +  
 scale\_x\_discrete(labels = c("2019", "2020")) +  
 scale\_y\_continuous(labels = scales::comma) +  
 theme(axis.text.x = element\_text(size = 10, vjust = 0.5))  
  
print(fraud\_percentage)



* What types of purchases are most likely to be instances of fraud? Consider category of merchants and produce a bar chart that shows % of total fraudulent transactions sorted in order.

# Calculate the percentage of fraudulent transactions by merchant category  
fraud\_by\_category <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 count(category, wt = amt) %>%  
 mutate(percentage = (n / sum(n)) \* 100) %>%  
 arrange(desc(percentage))  
  
# Create a bar chart  
bar\_chart <- ggplot(fraud\_by\_category, aes(x = reorder(category, percentage), y = percentage)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Percentage of Fraudulent Transactions by Merchant Category",  
 x = "Merchant Category", y = "Percentage of Fraudulent Transactions") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
  
# Print the bar chart  
bar\_chart



* When is fraud more prevalent? Which days, months, hours? To create new variables to help you in your analysis, we use the lubridate package and the following code

mutate(  
 date\_only = lubridate::date(trans\_date\_trans\_time),  
 month\_name = lubridate::month(trans\_date\_trans\_time, label=TRUE),  
 hour = lubridate::hour(trans\_date\_trans\_time),  
 weekday = lubridate::wday(trans\_date\_trans\_time, label = TRUE)  
 )

* Are older customers significantly more likely to be victims of credit card fraud? To calculate a customer’s age, we use the lubridate package and the following code

card\_fraud <- card\_fraud %>%   
 mutate(  
 date\_only = lubridate::date(trans\_date\_trans\_time),  
 month\_name = lubridate::month(trans\_date\_trans\_time, label=TRUE),  
 hour = lubridate::hour(trans\_date\_trans\_time),  
 weekday = lubridate::wday(trans\_date\_trans\_time, label = TRUE)  
 )  
  
#When is fraud more prevalent?   
#by date #2019-01-18 has the most fraud  
card\_fraud %>%   
 filter(is\_fraud == 1) %>%   
 group\_by(date\_only) %>%  
 select(date\_only, amt) %>%   
 arrange(desc(amt))

## # A tibble: 3,936 × 2  
## # Groups: date\_only [493]  
## date\_only amt  
## <date> <dbl>  
## 1 2019-01-18 1334.  
## 2 2019-12-08 1313.  
## 3 2020-06-07 1313.  
## 4 2019-12-20 1295.  
## 5 2019-06-03 1288.  
## 6 2019-08-31 1282.  
## 7 2019-05-20 1262.  
## 8 2019-02-15 1259.  
## 9 2020-03-10 1258.  
## 10 2019-10-05 1246.  
## # ℹ 3,926 more rows

#by months #Mar and May have the most fraud card\_fraud %>% filter(is\_fraud == 1) %>% group\_by(month\_name) %>% summarise(count = n()) %>% arrange(desc(count))

#by hours #Fraud is likely to happen at hour 23 card\_fraud %>% filter(is\_fraud == 1) %>% group\_by(hour) %>% summarise(count = n()) %>% arrange(desc(count)) #by day #Monday has the most fraud card\_fraud %>% filter(is\_fraud == 1) %>% group\_by(weekday) %>% summarise(count = n()) %>% arrange(desc(count))

’’’

* Is fraud related to distance? The distance between a card holder’s home and the location of the transaction can be a feature that is related to fraud. To calculate distance, we need the latidue/longitude of card holders’s home and the latitude/longitude of the transaction, and we will use the [Haversine formula](https://en.wikipedia.org/wiki/Haversine_formula) to calculate distance. I adapted code to [calculate distance between two points on earth](https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/) which you can find below

# distance between card holder's home and transaction  
# code adapted from https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/  
  
  
card\_fraud <- card\_fraud %>%  
 mutate(  
 # convert latitude/longitude to radians  
 lat1\_radians = lat / 57.29577951,  
 lat2\_radians = merch\_lat / 57.29577951,  
 long1\_radians = long / 57.29577951,  
 long2\_radians = merch\_long / 57.29577951,  
   
 # calculate distance in miles  
 distance\_miles = 3963.0 \* acos((sin(lat1\_radians) \* sin(lat2\_radians)) + cos(lat1\_radians) \* cos(lat2\_radians) \* cos(long2\_radians - long1\_radians)),  
  
 # calculate distance in km  
 distance\_km = 6377.830272 \* acos((sin(lat1\_radians) \* sin(lat2\_radians)) + cos(lat1\_radians) \* cos(lat2\_radians) \* cos(long2\_radians - long1\_radians))  
 )

Plot a boxplot or a violin plot that looks at the relationship of distance and is\_fraud. Does distance seem to be a useful feature in explaining fraud?

# Exploring sources of electricity production, CO2 emissions, and GDP per capita.

There are many sources of data on how countries generate their electricity and their CO2 emissions. I would like you to create three graphs:

## 1. A stacked area chart that shows how your own country generated its electricity since 2000.

You will use

geom\_area(colour="grey90", alpha = 0.5, position = "fill")

## 2. A scatter plot that looks at how CO2 per capita and GDP per capita are related

## 3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related

We will get energy data from the Our World in Data website, and CO2 and GDP per capita emissions from the World Bank, using the wbstatspackage.

# Download electricity data  
url <- "https://nyc3.digitaloceanspaces.com/owid-public/data/energy/owid-energy-data.csv"  
  
energy <- read\_csv(url) %>%   
 filter(year >= 1990) %>%   
 drop\_na(iso\_code) %>%   
 select(1:3,  
 biofuel = biofuel\_electricity,  
 coal = coal\_electricity,  
 gas = gas\_electricity,  
 hydro = hydro\_electricity,  
 nuclear = nuclear\_electricity,  
 oil = oil\_electricity,  
 other\_renewable = other\_renewable\_exc\_biofuel\_electricity,  
 solar = solar\_electricity,  
 wind = wind\_electricity,   
 electricity\_demand,  
 electricity\_generation,  
 net\_elec\_imports, # Net electricity imports, measured in terawatt-hours  
 energy\_per\_capita, # Primary energy consumption per capita, measured in kilowatt-hours Calculated by Our World in Data based on BP Statistical Review of World Energy and EIA International Energy Data  
 energy\_per\_gdp, # Energy consumption per unit of GDP. This is measured in kilowatt-hours per 2011 international-$.  
 per\_capita\_electricity, # Electricity generation per capita, measured in kilowatt-hours  
 )   
  
# Download data for C02 emissions per capita https://data.worldbank.org/indicator/EN.ATM.CO2E.PC  
co2\_percap <- wb\_data(country = "countries\_only",   
 indicator = "EN.ATM.CO2E.PC",   
 start\_date = 1990,   
 end\_date = 2022,  
 return\_wide=FALSE) %>%   
 filter(!is.na(value)) %>%   
 #drop unwanted variables  
 select(-c(unit, obs\_status, footnote, last\_updated)) %>%   
 rename(year = date,  
 co2percap = value)  
  
  
# Download data for GDP per capita https://data.worldbank.org/indicator/NY.GDP.PCAP.PP.KD  
gdp\_percap <- wb\_data(country = "countries\_only",   
 indicator = "NY.GDP.PCAP.PP.KD",   
 start\_date = 1990,   
 end\_date = 2022,  
 return\_wide=FALSE) %>%   
 filter(!is.na(value)) %>%   
 #drop unwanted variables  
 select(-c(unit, obs\_status, footnote, last\_updated)) %>%   
 rename(year = date,  
 GDPpercap = value)

Specific questions:

1. How would you turn energy to long, tidy format?
2. You may need to join these data frames
   * Use left\_join from dplyr to [join the tables](http://r4ds.had.co.nz/relational-data.html)
   * To complete the merge, you need a unique *key* to match observations between the data frames. Country names may not be consistent among the three dataframes, so please use the 3-digit ISO code for each country
   * An aside: There is a great package called [countrycode](https://github.com/vincentarelbundock/countrycode) that helps solve the problem of inconsistent country names (Is it UK? United Kingdom? Great Britain?). countrycode() takes as an input a country’s name in a specific format and outputs it using whatever format you specify.
3. Write a function that takes as input any country’s name and returns all three graphs. You can use the patchwork package to arrange the three graphs as shown below

# 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 (qmd) 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 comitting 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: 8 hours
* What, if anything, gave you the most trouble: trying to solve the errors

**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.