Homework 2

VICTOR RIOS FARIA

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: 144  
## Columns: 24  
## $ case <chr> "New Mexico neighborhood shooting", "…  
## $ location...2 <chr> "Farmington, New Mexico", "Allen, Tex…  
## $ date <chr> "5/15/23", "5/6/23", "4/10/23", "3/27…  
## $ summary <chr> "Beau Wilson, 18, opened fire on the …  
## $ fatalities <dbl> 3, 8, 5, 6, 3, 7, 11, 6, 5, 3, 5, 3, …  
## $ injured <dbl> 6, 7, 8, 6, 5, 1, 10, 6, 25, 2, 2, 2,…  
## $ total\_victims <dbl> 9, 15, 13, 12, 8, 8, 21, 12, 30, 5, 7…  
## $ location...8 <chr> "Other", "Other", "workplace", "Schoo…  
## $ age\_of\_shooter <chr> "18", "33", "25", "28", "43", "67", "…  
## $ prior\_signs\_mental\_health\_issues <chr> "-", "yes", "yes", "-", "-", "-", "ye…  
## $ mental\_health\_details <chr> "-", "Reportedly had a history of men…  
## $ weapons\_obtained\_legally <chr> "yes", "yes", "yes", "yes", "yes", "-…  
## $ where\_obtained <chr> "-", "-", "gun dealership in Louisvil…  
## $ weapon\_type <chr> "semiautiomatic rifle; semiautomatic …  
## $ weapon\_details <chr> "AR-15-style rifle", "AR-15-style rif…  
## $ race <chr> "-", "Latino", "White", "White", "Bla…  
## $ gender <chr> "M", "M", "M", "F (\"identifies as tr…  
## $ sources <chr> "https://www.cbsnews.com/news/farming…  
## $ mental\_health\_sources <chr> "-", "-", "-", "-", "-", "-", "https:…  
## $ sources\_additional\_age <chr> "-", "-", "-", "-", "-", "-", "-", "-…  
## $ latitude <chr> "-", "-", "-", "-", "-", "-", "-", "-…  
## $ longitude <chr> "-", "-", "-", "-", "-", "-", "-", "-…  
## $ type <chr> "mass", "Mass", "Mass", "Mass", "Mass…  
## $ year <dbl> 2023, 2023, 2023, 2023, 2023, 2023, 2…

| 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

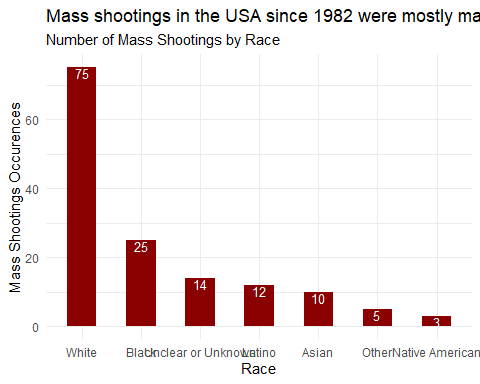
### Specific questions

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

#| label: Question 1  
  
yearly\_mass\_shootings <- mass\_shootings %>% # creates a new data frame to store the information about the number of mass shootings per year, which will be derived from the original mass\_shootings data  
 group\_by(year) %>% # group the data from the original data by year  
 summarise(n()) # counts how many mass shootings there was per year

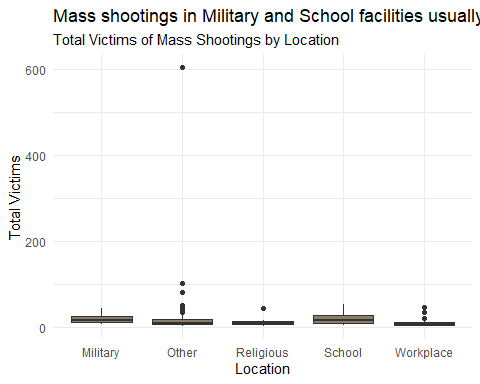
* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

#| label: Question 2  
  
mass\_shootings\_race <- mass\_shootings %>% # defines a new datastream for corrected/consolidated race information  
   
 mutate(race = case\_when(  
 race == "white" ~ "White", # unifies the spelling for White  
 race == "black" ~ "Black", # unifies the spelling for Black  
 race %in% c("unclear", "-") ~ "Unclear or Unknown", # consolidates "unclear" and "-" into a single category  
 TRUE ~ race # handles other cases that don't match the conditions  
 ))  
  
mass\_shootings\_race %>%  
   
 group\_by(race) %>% # groups the data by race  
 summarise(count = n()) %>% # count the number of mass shootings per race  
 ggplot(aes(x = reorder(race, -count), y = count)) + # creates a plot that shows the number of mass shootings per race  
 geom\_bar(stat = "identity", fill = "red4", width = 0.5) + # sets the plot as a bar plot  
 geom\_text(aes(label = count), vjust = 1, size = 3.5, color = "white") + # adds the numbers to the bars  
 labs(x = "Race", y = "Mass Shootings Occurences", title = "Mass shootings in the USA since 1982 were mostly made by white people ", subtitle = "Number of Mass Shootings by Race") + # adds labels to the axis, title and subtitle  
 theme\_minimal()



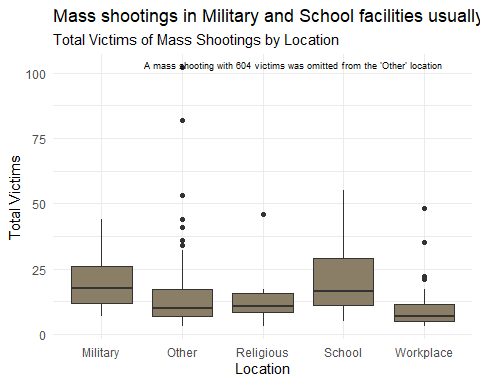
* Generate a boxplot visualizing the number of total victims, by type of location.

#| label: Question 3  
  
mass\_shootings %>%  
 mutate(location...8 = case\_when(  
 location...8 == "religious" ~ "Religious", # unifies the spelling for Religious  
 location...8 %in% c("Other\n", "Airport") ~ "Other", # unifies the spelling for Other and adds Airport to this grouping because it has only one observation  
 location...8 %in% c("\nWorkplace", "workplace") ~ "Workplace", # unifies the spelling for Workplace  
 TRUE ~ location...8 # handles other cases that don't match the conditions  
 )) %>%  
 ggplot(aes(x = location...8, y = total\_victims)) + # creates a plot for the total victims per location  
 geom\_boxplot(fill = "wheat4") + # defines the plot as a boxplot  
 labs(x = "Location", y = "Total Victims", title = "Mass shootings in Military and School facilities usually have more victims", subtitle = "Total Victims of Mass Shootings by Location") + # adds the axis labels, title and subtitle  
 theme\_minimal()



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

#| label: Question 4  
  
mass\_shootings %>%  
 filter(total\_victims<200) %>% # remove the Las Vegas Strip massacre  
 mutate(location...8 = case\_when(  
 location...8 == "religious" ~ "Religious", # unifies the spelling for Religious  
 location...8 %in% c("Other\n", "Airport") ~ "Other", # unifies the spelling for Other and adds Airport to this grouping because it has only one observation  
 location...8 %in% c("\nWorkplace", "workplace") ~ "Workplace", # unifies the spelling for Workplace  
 TRUE ~ location...8 # handles other cases that don't match the conditions  
 )) %>%  
 ggplot(aes(x = location...8, y = total\_victims)) + # creates a plot for the total victims per location  
 geom\_boxplot(fill = "wheat4") + # defines the plot as a boxplot  
 labs(x = "Location", y = "Total Victims", title = "Mass shootings in Military and School facilities usually have more victims", subtitle = "Total Victims of Mass Shootings by Location") + # adds the axis labels, title and subtitle  
 theme\_minimal() +  
 annotate("text", x = Inf, y = Inf, label = "A mass shooting with 604 victims was omitted from the 'Other' location", hjust = 1.1, vjust = 1.7, size = 2.5) # adds an annotation to tell that the Las Vegas Strip massacre has been removed from the datastream



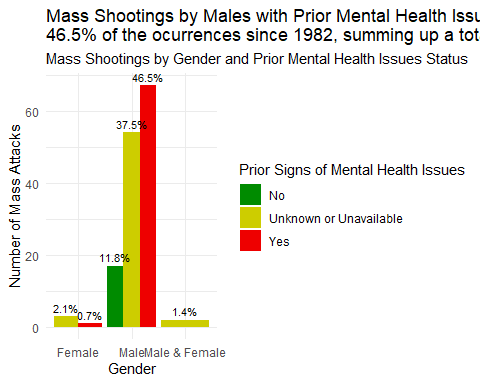
### More open-ended questions

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?

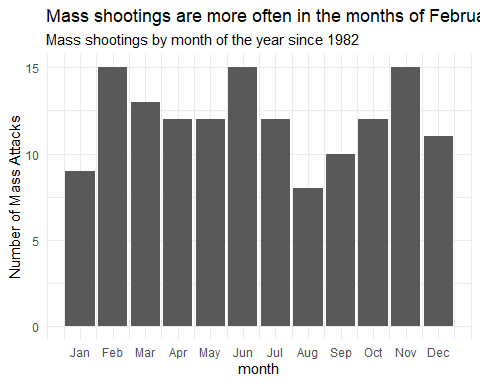
#| label: Question 5  
  
mass\_shootings\_mh <- mass\_shootings %>% # defines a new datastream consolidating the information on prior signs mental health issues  
   
 mutate(prior\_signs\_mental\_health\_issues = case\_when(   
 prior\_signs\_mental\_health\_issues %in% c("yes", "Yes") ~ "Yes", # consolidates all spellings of Yes into a single category  
 prior\_signs\_mental\_health\_issues %in% c("no", "No") ~ "No", # consolidates all spellings of No into a single category  
 TRUE ~ "Unknown or Unavailable" # consolidates all other cases as Unknown or Unavailable  
 ))  
  
mass\_shootings\_mh %>%  
 mutate(gender = case\_when(  
 gender == "M" ~ "Male", # consolidates M and Male into a single category  
 gender == "F" ~ "Female", # consolidates F and Female into a single category  
 substring(gender,1,1) == "F" ~ "Female", # sets the category for the long string starting with an F as Female  
 TRUE ~ gender # handles other cases that don't match the conditions   
 )) %>%  
   
 group\_by(gender, prior\_signs\_mental\_health\_issues) %>% # groups the data by both the gender and prior mental health issues  
 summarise(count=n()) %>% # counts how many mass attacks were initiated by each combination of gender and prior mental health issues  
 ungroup %>% # ungroups the data so total percentages can be calculated later  
   
 mutate(perc\_cases = count / sum(count)) %>% # creates a new field with the percentage of total cases per combination of gender and prior mental health issues  
   
 ggplot(aes(x=gender, y=count, fill=prior\_signs\_mental\_health\_issues)) + # creates a plot of the number of mass attacks per gender and prior mental health issues status  
 geom\_bar(stat = "identity", position = "dodge") + # configures the display of the bars in the chart  
 geom\_text(aes(label = paste0(round(perc\_cases\*100, 1), "%")), # defines the labels in the bars as a percentage  
 position = position\_dodge(width = 0.9), # and its position in regard to the bars,  
 hjust = 0.5, # horizontally  
 vjust = -0.5, # and vertically  
 colour = "black", # defines the label text as black  
 size = 3)+ # and font size 3  
 scale\_fill\_manual(values = c("green4", "yellow3", "red2")) + # defines the colour of the bars for each prior mental health issues status  
 labs(x = "Gender", #creates labels for the x-axis...  
 y = "Number of Mass Attacks", # and the y-axis...  
 fill = "Prior Signs of Mental Health Issues", # changes the colour legend's title  
 title = "Mass Shootings by Males with Prior Mental Health Issues represents \n46.5% of the ocurrences since 1982, summing up a total of 67 occurences",# sets the chart title...  
 subtitle = "Mass Shootings by Gender and Prior Mental Health Issues Status") + # and its subtitle  
 theme\_minimal()

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



* 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.

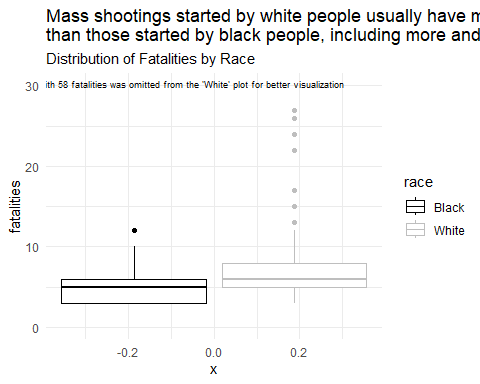
#| label: Question 6  
  
mass\_shootings %>%  
 separate(date, into = c("month", "day", "year"), sep = "/") %>% # breaks the information in the data column into three different variables: month, day and year  
 mutate(month = as.numeric(month)) %>% # transform the month variable from char to numeric so that later it gets sorted by ascending numerical order  
 group\_by(month) %>% # group the data by month  
 summarise(count = n()) %>% # count the number of mass attacks per month of the year  
   
 ggplot(aes(x = month, y = count)) + # creates a plot of the number of mass attacks per month  
 geom\_bar(stat = "identity") + # defines it as a bar plot  
 scale\_x\_continuous(breaks = 1:12, labels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))+ # tags the x-axis with the months abbreviation instead of the numbers  
 labs(y = "Number of Mass Attacks", # adds the title to the y-axis  
 title = "Mass shootings are more often in the months of February, June and November", # adds a title to the chart  
 subtitle = "Mass shootings by month of the year since 1982") + # adds a subtitle to the chart  
 theme\_minimal()



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

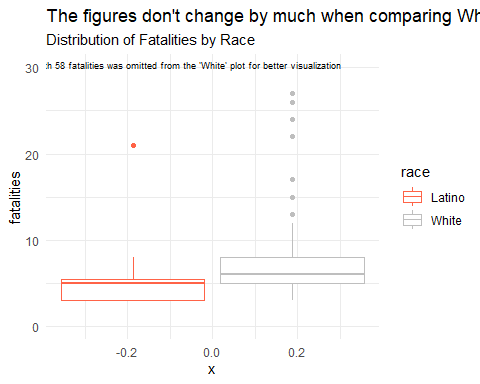
#| label: Question 7  
  
mass\_shootings\_race %>% # recalls the datastream created in Question 2 with corrected/consolidated race information  
   
 filter(race == "White" | race == "Black") %>% # filters the data to keep only the 2 races we want to compare  
 group\_by(race) %>% # groups the data by race  
   
 ggplot(aes(y=fatalities, color=race)) + # creates a plot of the number of fatalities for each race  
 geom\_boxplot() + # defines the chart s a boxplot  
 scale\_color\_manual(values = c("White" = "grey", "Black" = "black")) + # sets different colours for the two races  
 labs(subtitle = "Distribution of Fatalities by Race", title = "Mass shootings started by white people usually have more fatalities \nthan those started by black people, including more and higher outliers") + # adds a title and a subtitle to the chart  
 theme\_minimal() +  
 ylim(0, 30) + # changes the limit of the y-axis in the plot  
 annotate("text", x = Inf, y = Inf, label = "A mass shooting with 58 fatalities was omitted from the 'White' plot for better visualization", hjust = 1.1, vjust = 1.7, size = 2.5) # adds an annotation to the chart to cite the removed observation

## Warning: Removed 1 rows containing non-finite values (`stat\_boxplot()`).



##########################################################################################  
  
# repeats the process replacing black for latino  
  
mass\_shootings\_race %>%  
   
 filter(race == "White" | race == "Latino") %>% # filters the data to keep only the 2 races we want to compare  
 group\_by(race) %>% # groups the data by race  
   
 ggplot(aes(y=fatalities, color=race)) + # creates a plot of the number of fatalities for each race  
 geom\_boxplot() + # defines the chart s a boxplot  
 scale\_color\_manual(values = c("White" = "grey", "Latino" = "tomato")) +  
 labs(subtitle = "Distribution of Fatalities by Race", title = "The figures don't change by much when comparing White to Latinos") + # sets different colours for the two races  
 theme\_minimal() +  
 ylim(0, 30) + # changes the limit of the y-axis in the plot  
 annotate("text", x = Inf, y = Inf, label = "A mass shooting with 58 fatalities was omitted from the 'White' plot for better visualization", hjust = 1.1, vjust = 1.7, size = 2.5) # adds an annotation to the chart to cite the removed observation

## Warning: Removed 1 rows containing non-finite values (`stat\_boxplot()`).



### 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?

#| label: Question 8  
  
mass\_shootings\_mh %>% # makes reference to the datastream with consolidated mental health information  
   
 filter(prior\_signs\_mental\_health\_issues != "Unknown or Unavailable") %>% # filters the data to contain only cases where the shooter had a conclusive record  
 group\_by(prior\_signs\_mental\_health\_issues) %>% # groups the data by the mental health status  
 summarise(count = n(), # counts how many mass shootings there has been per group  
 avg\_fatalities = mean(fatalities), # gets the average of fatalities per group  
 avg\_victims = mean(total\_victims), # gets the average number of victims per group  
 med\_fatalities = median(fatalities), # gets the median of fatalities per group  
 med\_victims = median(total\_victims)) # gets the median number of victims per group

## # A tibble: 2 × 6  
## prior\_signs\_mental\_health\_is…¹ count avg\_fatalities avg\_victims med\_fatalities  
## <chr> <int> <dbl> <dbl> <dbl>  
## 1 No 17 7.06 11.8 6  
## 2 Yes 68 8.28 17 7  
## # ℹ abbreviated name: ¹​prior\_signs\_mental\_health\_issues  
## # ℹ 1 more variable: med\_victims <dbl>

# From some simple statistics, we are leaned to conclude that mass shootings by people with prior signs of mental health issues are not only more frequent, but also lead to more fatalities and victims  
  
# Filter the data for the two groups of interest  
group\_yes <- mass\_shootings\_mh %>% filter(prior\_signs\_mental\_health\_issues == "Yes")  
group\_no <- mass\_shootings\_mh %>% filter(prior\_signs\_mental\_health\_issues == "No")  
  
# Perform t-test on fatalities  
t.test(group\_yes$fatalities, group\_no$fatalities)

##   
## Welch Two Sample t-test  
##   
## data: group\_yes$fatalities and group\_no$fatalities  
## t = 0.91024, df = 29.089, p-value = 0.3702  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -1.521613 3.962789  
## sample estimates:  
## mean of x mean of y   
## 8.279412 7.058824

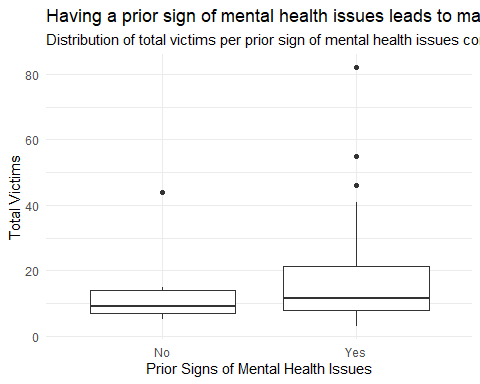
# Perform t-test on total victims  
t.test(group\_yes$total\_victims, group\_no$total\_victims)

##   
## Welch Two Sample t-test  
##   
## data: group\_yes$total\_victims and group\_no$total\_victims  
## t = 1.8812, df = 38.502, p-value = 0.06753  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -0.3917532 10.7446944  
## sample estimates:  
## mean of x mean of y   
## 17.00000 11.82353

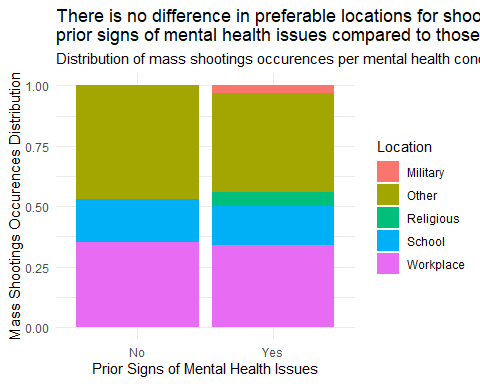
# However, a t-test on the number of fatalities and victims does not show statistically significant difference at 5% level between the two groups

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

#| label: Question 9  
  
  
mass\_shootings\_mh %>% # makes reference to the datastream with consolidated mental health information  
   
 filter(prior\_signs\_mental\_health\_issues != "Unknown or Unavailable") %>% # filters the data to contain only cases where the shooter had a conclusive record  
   
 ggplot(aes(y = total\_victims, x = prior\_signs\_mental\_health\_issues)) + # plots the total victims of the mass shooting for different mental health conditions  
 geom\_boxplot() + # defines the plot as a boxplot  
 labs(  
 x = "Prior Signs of Mental Health Issues", # adds a title to the x-axis  
 y = "Total Victims", # adds a title to the y-axis  
 title = "Having a prior sign of mental health issues leads to mass attacks with more victims", # adds a title to the chart  
 subtitle = "Distribution of total victims per prior sign of mental health issues condition", # adds a subtitle to the chart  
 ) +  
 theme\_minimal()

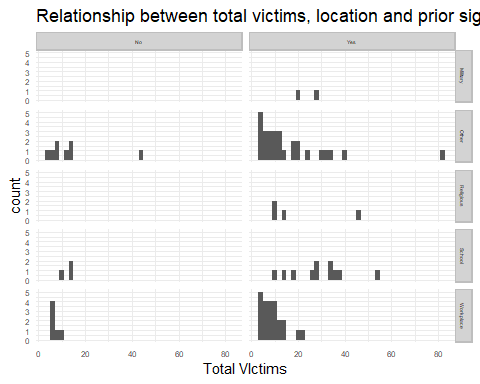


# From the plot, one may think there are more victims in mass shootings initiated by people with prior signs of mental health issues. However, the t-test performed in the previous question showed that we can not state that the average number of victims is different between the two groups in a 5% significance level  
  
##########################################################################################################  
  
mass\_shootings\_mh %>%  
 filter(total\_victims<200) %>% # remove the Las Vegas Strip massacre  
 mutate(location...8 = case\_when(  
 location...8 == "religious" ~ "Religious", # unifies the spelling for Religious  
 location...8 %in% c("Other\n", "Airport") ~ "Other", # unifies the spelling for Other and adds Airport to this grouping because it has only one observation  
 location...8 %in% c("\nWorkplace", "workplace") ~ "Workplace", # unifies the spelling for Workplace  
 TRUE ~ location...8 # handles other cases that don't match the conditions  
 )) %>%  
   
 filter(prior\_signs\_mental\_health\_issues != "Unknown or Unavailable") %>% # filters the data to contain only cases where the shooter had a conclusive record  
   
 ggplot(aes(x = prior\_signs\_mental\_health\_issues, fill = location...8)) + # plots the occurences per location and mental health state  
 geom\_bar(position = "fill") + # defines the plot as a bar chart  
 labs(x = "Prior Signs of Mental Health Issues", # adds a name to the x-axis  
 y = "Mass Shootings Occurences Distribution", # adds a name to the y-axis  
 fill = "Location", # adds a name to the colour mapping  
 title = "There is no difference in preferable locations for shoothers with \nprior signs of mental health issues compared to those with no signs", # adds a title to the chart  
 subtitle = "Distribution of mass shootings occurences per mental health condition and shooting location"  
 ) + # adds a subtitle to the chart  
 theme\_minimal()



# The plot shows that the distribuition of mass shootings accross the different location types is very similar between the two groups, meaning that people with prior signs of mental health issues do not have a different behaviour in terms of where they start the mass shootings  
  
##########################################################################################################  
  
mass\_shootings\_mh %>%  
 filter(total\_victims<200) %>% # remove the Las Vegas Strip massacre  
 mutate(location = case\_when(  
 location...8 == "religious" ~ "Religious", # unifies the spelling for Religious  
 location...8 %in% c("Other\n", "Airport") ~ "Other", # unifies the spelling for Other and adds Airport to this grouping because it has only one observation  
 location...8 %in% c("\nWorkplace", "workplace") ~ "Workplace", # unifies the spelling for Workplace  
 TRUE ~ location...8 # handles other cases that don't match the conditions  
 )) %>%  
   
 filter(prior\_signs\_mental\_health\_issues != "Unknown or Unavailable") %>% # filters the data to contain only cases where the shooter had a conclusive record  
  
 ggplot(aes(x = total\_victims)) + # creates a plot of the total number of victims  
 geom\_histogram(binwidth = 2) + # sets the plot as a histogram with the given binwidth  
 facet\_grid(location ~ prior\_signs\_mental\_health\_issues) + # creates a facet grid to set a different histogram for each combination of location and prior signs of mental health issues  
 labs(title = "Relationship between total victims, location and prior signs of mental health issues", # adds a title to the chart  
 x = "Total VIctims") + # adds a title to the x-axis  
 theme\_minimal() +  
 theme(axis.text = element\_text(size = 6),  
 strip.text = element\_text(size = 4),  
 strip.background = element\_rect(fill = "lightgray", color = "gray", size = 1))

## Warning: The `size` argument of `element\_rect()` is deprecated as of ggplot2 3.4.0.  
## ℹ Please use the `linewidth` argument instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# From this series of plot, it appears that victims have a stronger dependency on the location than on the mental health condition of the shooter. However, there is to few data to perform a more conclusive analysis

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 <date> 1959-10-19, 1946-04-03, 1985-03-31, 1991-01-28,…  
## $ 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.

#| label: Question 10  
  
card\_fraud %>%  
 group\_by(is\_fraud, trans\_year) %>% # groups the data by the fraudulent status and year  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 mutate(is\_fraud = ifelse(is\_fraud == 0, "not\_fraudulent", "fraudulent")) %>% # replaces the values 0 and 1 for values that can be easily understood when they are used as a variable name  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) # calculates the frequency of fraudulent transactions per year

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

## # A tibble: 2 × 4  
## trans\_year not\_fraudulent fraudulent frequency\_fraud  
## <dbl> <int> <int> <dbl>  
## 1 2019 475925 2721 0.00568  
## 2 2020 191167 1215 0.00632

* 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.

#| label: Question 11  
  
card\_fraud %>%  
 group\_by(is\_fraud, trans\_year) %>% # groups the data by the fraudulent status and year  
 summarise(total\_transactions\_amt = sum(amt))%>% # summarises the data summing how many dollars were spent in fraudulent and non fraudulent transactions  
 ungroup() %>% # ungroup the data to manipulate it and get the percentages  
 mutate(is\_fraud = ifelse(is\_fraud == 0, "not\_fraudulent", "fraudulent")) %>% # replaces the values 0 and 1 for values that can be easily understood when they are used as a variable name  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "total\_transactions\_amt") %>% # ... and the values from the count variable  
 mutate(perc\_amt\_fraud = fraudulent/(fraudulent+not\_fraudulent)) # calculates the frequency of fraudulent transactions per year

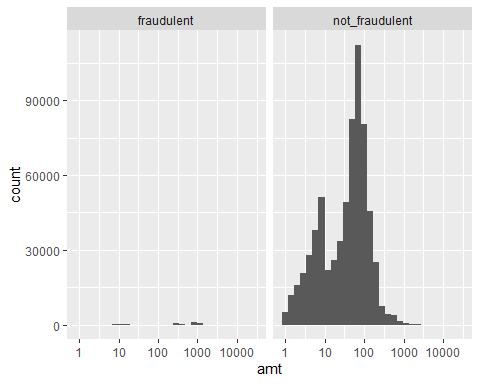
## `summarise()` has grouped output by 'is\_fraud'. You can override using the  
## `.groups` argument.

## # A tibble: 2 × 4  
## trans\_year not\_fraudulent fraudulent perc\_amt\_fraud  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 32182901. 1423140. 0.0423  
## 2 2020 12925914. 651949. 0.0480

* 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.

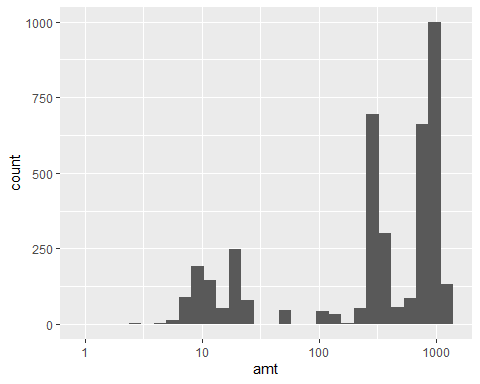
#| label: Question 12  
  
card\_fraud %>%  
 mutate(is\_fraud = ifelse(is\_fraud == 0, "not\_fraudulent", "fraudulent")) %>% # replaces the values 0 and 1 for values that can be easily understood when they are used as a variable name  
 ggplot(aes(x=amt)) + # creates a plot for the value of the transaction  
 geom\_histogram() + # sets the chart as a histogram  
 scale\_x\_log10() + # changes the scale to log so that the chart has a better shape for visualization  
 facet\_wrap(~ is\_fraud) # creates two different histograms based on the fraud status

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



# However, the visualization for the fraudulent category is not great, so let's plot it separately  
  
card\_fraud %>%  
 filter(is\_fraud==1) %>% # filters only the fraudulent transactions  
 ggplot(aes(x=amt)) + # creates a plot for the value of the transaction  
 geom\_histogram() + # sets the chart as a histogram  
 scale\_x\_log10() # changes the scale to log so that the chart has a better shape for visualization

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



# From the histograms, we can see that fraudulent transactions are concentrated around 10 dollars and 1,000 dollars, while not fraudulent transactions are more concentrated around 10 dollars and 100 dollars  
  
card\_fraud %>%  
 mutate(amt\_range = round(log10(amt),1)) %>% # applies a transformation to make the analysis more consistent to the previous histograms  
 group\_by(is\_fraud, amt\_range) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by amt\_range  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 mutate(is\_fraud = ifelse(is\_fraud == 0, "not\_fraudulent", "fraudulent")) %>% # replaces the values 0 and 1 for values that can be easily understood when they are used as a variable name  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(fraudulent=if\_else(is.na(fraudulent),0,fraudulent)) %>% # replaces na for 0 to avoid numerical errors  
 mutate(amt\_range = 10^amt\_range) %>% # undoes the log transformation  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per amt range  
 filter(frequency\_fraud>=0.05) # filters approximate transaction values that have more than 5% of chances of being a fraud, which is about 8 to 10 times more likely than the average fraud frequency. In the range of 794 to 1000 dollars value, frauds frequency can get as high as 50%

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

## # A tibble: 5 × 4  
## amt\_range not\_fraudulent fraudulent frequency\_fraud  
## <dbl> <int> <dbl> <dbl>  
## 1 316. 3515 791 0.184   
## 2 631. 1589 112 0.0658  
## 3 794. 1008 708 0.413   
## 4 1000 693 925 0.572   
## 5 1259. 462 112 0.195

amt\_summary <- summary(card\_fraud$amt) # gets a quick statistical summary of the amt variable  
amt\_summary$sd <- sd(card\_fraud$amt) # adds the standard deviation to the statistic summary

## Warning in amt\_summary$sd <- sd(card\_fraud$amt): Coercing LHS to a list

print(amt\_summary)

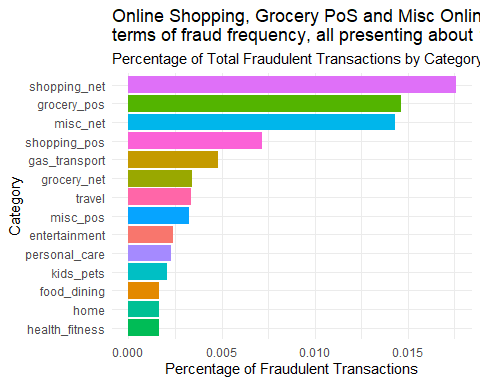
## $Min.  
## [1] 1  
##   
## $`1st Qu.`  
## [1] 9.64  
##   
## $Median  
## [1] 47.41  
##   
## $Mean  
## [1] 70.31585  
##   
## $`3rd Qu.`  
## [1] 83.01  
##   
## $Max.  
## [1] 27119.77  
##   
## $sd  
## [1] 161.5691

* 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.

#| label: Question 13  
  
card\_fraud %>%  
 mutate(is\_fraud = ifelse(is\_fraud == 0, "not\_fraudulent", "fraudulent")) %>% # replaces the values 0 and 1 for values that can be easily understood when they are used as a variable name  
 group\_by(category, is\_fraud) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by category  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per category  
   
 ggplot(aes(x = reorder(category, frequency\_fraud), y = frequency\_fraud, fill = category)) +  
 geom\_bar(stat = "identity") +  
 coord\_flip() +  
 labs(subtitle = "Percentage of Total Fraudulent Transactions by Category",  
 title = "Online Shopping, Grocery PoS and Misc Online are the top-3 categories in \nterms of fraud frequency, all presenting about 15% of fraudulent transactions",  
 x = "Category",  
 y = "Percentage of Fraudulent Transactions") +  
 scale\_fill\_discrete(guide = FALSE) +  
 theme\_minimal()

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

## Warning: The `guide` argument in `scale\_\*()` cannot be `FALSE`. This was deprecated in  
## ggplot2 3.3.4.  
## ℹ Please use "none" instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



* 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

mutate(  
 age = interval(dob, trans\_date\_trans\_time) / years(1),  
 )

#| label: Question 14  
  
# Setting up a new datastream with the suggested data manipulation  
  
card\_fraud\_lubridate <- card\_fraud %>%  
 mutate(is\_fraud = ifelse(is\_fraud == 0, "not\_fraudulent", "fraudulent")) %>% # replaces the values 0 and 1 for values that can be easily understood when they are used as a variable name  
 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)  
 ) %>%  
 mutate(  
 age = interval(dob, trans\_date\_trans\_time) / years(1),  
 )  
###############################################################################################################  
  
# Evaluating days of the week  
  
card\_fraud\_lubridate %>%  
 group\_by(weekday, is\_fraud) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by weekday  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per weekday  
 arrange(desc(frequency\_fraud))

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

## # A tibble: 7 × 4  
## weekday fraudulent not\_fraudulent frequency\_fraud  
## <ord> <int> <int> <dbl>  
## 1 Thu 542 75658 0.00711  
## 2 Fri 557 78394 0.00706  
## 3 Wed 468 67471 0.00689  
## 4 Sat 626 103413 0.00602  
## 5 Tue 496 82434 0.00598  
## 6 Mon 639 130780 0.00486  
## 7 Sun 608 128942 0.00469

# Thursday and Friday are the days with more frequent frauds with about 0.7%, but the difference is not as significant as the transaction value previously evaluated  
   
###############################################################################################################  
  
# Evaluating months  
  
card\_fraud\_lubridate %>%  
 group\_by(month\_name, is\_fraud) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by month  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per month  
 arrange(desc(frequency\_fraud))

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

## # A tibble: 12 × 4  
## month\_name fraudulent not\_fraudulent frequency\_fraud  
## <ord> <int> <int> <dbl>  
## 1 Jan 461 53345 0.00857  
## 2 Feb 434 50226 0.00857  
## 3 Mar 472 74006 0.00634  
## 4 May 472 75329 0.00623  
## 5 Nov 226 36107 0.00622  
## 6 Oct 218 35869 0.00604  
## 7 Sep 219 36314 0.00599  
## 8 Jun 387 73827 0.00521  
## 9 Apr 349 69527 0.00499  
## 10 Aug 213 45067 0.00470  
## 11 Dec 301 72685 0.00412  
## 12 Jul 184 44790 0.00409

# January and February are the days with more frequent frauds with about 0.85% , but the difference is not as significant as the transaction value previously evaluated  
  
###############################################################################################################  
  
# Evaluating hours of the day  
  
card\_fraud\_lubridate %>%  
 group\_by(hour, is\_fraud) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by hour of the day  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per hour of the day  
 arrange(desc(frequency\_fraud))

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

## # A tibble: 24 × 4  
## hour fraudulent not\_fraudulent frequency\_fraud  
## <int> <int> <int> <dbl>  
## 1 23 1012 33613 0.0292   
## 2 22 981 33693 0.0283   
## 3 0 348 21722 0.0158   
## 4 1 332 21775 0.0150   
## 5 3 326 21866 0.0147   
## 6 2 313 21909 0.0141   
## 7 7 35 21820 0.00160  
## 8 19 52 33935 0.00153  
## 9 5 32 21720 0.00147  
## 10 18 49 34082 0.00144  
## # ℹ 14 more rows

# The vast majority of fraudulent transactions occur between 11 PM and 3 AM, all with more than 1.4% incidence ratio. Moreover, transactions happening at 10 PM and 11 PM have incidences as high as 2.9% approximately  
  
###############################################################################################################  
  
# Evaluating age  
  
card\_fraud\_lubridate %>%  
 mutate(age = 5\* round(age/5)) %>% # rounds the age to the nearest multiple of 5  
 group\_by(age, is\_fraud) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by age  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per age  
 arrange(desc(frequency\_fraud))

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

## # A tibble: 17 × 4  
## age fraudulent not\_fraudulent frequency\_fraud  
## <dbl> <int> <int> <dbl>  
## 1 80 175 14741 0.0117   
## 2 85 112 12065 0.00920  
## 3 60 337 40417 0.00827  
## 4 90 95 11627 0.00810  
## 5 55 380 48147 0.00783  
## 6 75 121 17461 0.00688  
## 7 20 239 35354 0.00671  
## 8 70 156 23599 0.00657  
## 9 65 251 38262 0.00652  
## 10 50 376 61317 0.00609  
## 11 25 302 49840 0.00602  
## 12 95 20 3770 0.00528  
## 13 30 359 73436 0.00486  
## 14 15 32 6958 0.00458  
## 15 40 285 62687 0.00453  
## 16 45 367 86314 0.00423  
## 17 35 329 81097 0.00404

# Fraudulent transactions are more frequent among people 80 years old or older, as well as among people around their 60's  
  
###############################################################################################################  
  
# Bonus: Evaluating state  
  
card\_fraud\_lubridate %>%  
 group\_by(state, is\_fraud) %>%  
 summarise(count = n())%>% # summarises the data counting how many fraudulent and non fraudulent transactions there is in the dataset by state  
 ungroup() %>% # ungroup the data to manipulate it and get the frequencies  
 pivot\_wider(names\_from = "is\_fraud", # pivots the data to the wider format taking column names from the is\_fraud variable...  
 values\_from = "count") %>% # ... and the values from the count variable  
 mutate(frequency\_fraud = fraudulent/(fraudulent+not\_fraudulent)) %>% # calculates the frequency of fraudulent transactions per state  
 arrange(desc(frequency\_fraud))

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

## # A tibble: 51 × 4  
## state fraudulent not\_fraudulent frequency\_fraud  
## <chr> <int> <int> <dbl>  
## 1 RI 6 267 0.0220   
## 2 AK 17 1084 0.0154   
## 3 ME 73 8541 0.00847  
## 4 NV 24 2894 0.00822  
## 5 OR 79 9541 0.00821  
## 6 CO 58 7154 0.00804  
## 7 TN 71 9021 0.00781  
## 8 NE 97 12372 0.00778  
## 9 NH 32 4302 0.00738  
## 10 OH 176 23919 0.00730  
## # ℹ 41 more rows

# RI and AK show overwhelming high fraud frequencies if compared to other states (22% and 15% respectively)  
  
###############################################################################################################

* 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))  
  
 )  
  
############################################################################################################  
  
#| label: Question 15  
  
# Load the required library (if not already loaded)  
library(stats)  
  
# Perform logistic regression  
model <- glm(is\_fraud ~ distance\_km, data = card\_fraud, family = binomial)  
  
# Print the summary of the logistic regression model  
summary(model)

##   
## Call:  
## glm(formula = is\_fraud ~ distance\_km, family = binomial, data = card\_fraud)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -5.134e+00 4.475e-02 -114.728 <2e-16 \*\*\*  
## distance\_km 2.111e-05 5.484e-04 0.038 0.969   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 48300 on 671027 degrees of freedom  
## Residual deviance: 48300 on 671026 degrees of freedom  
## AIC: 48304  
##   
## Number of Fisher Scoring iterations: 8

# The p-value of the logistic regression is very high, meaning that the distance\_km variable is not statistic significant to differentiate the fraud status of a transaction

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.

#| label: Question 16  
  
#| message: false  
#| warning: false  
  
# 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  
 )

## Rows: 21890 Columns: 129  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): country, iso\_code  
## dbl (127): year, population, gdp, biofuel\_cons\_change\_pct, biofuel\_cons\_chan...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# 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: GUSTAVO MENDONCA
* Approximately how much time did you spend on this problem set: 15 hours
* What, if anything, gave you the most trouble: The exercises were not that difficult but the amount of work was overwhelming for only 5 days between classes

**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.