Homerwork 2

Sakurako Mizutani

5/21/23

# 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

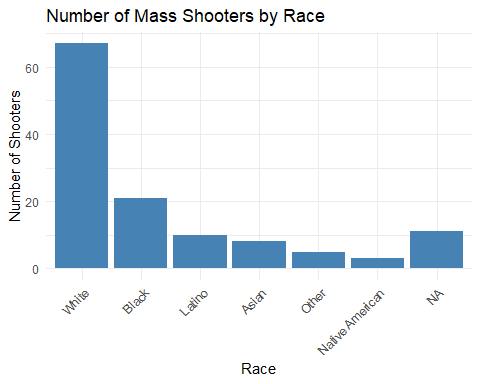
* Generate a data frame that summarizes the number of mass shootings per year.

# Convert the date columns to Date type  
mass\_shootings$year <- as.Date(paste(mass\_shootings$year, "01", "01", sep = "-"))  
  
# Group the data by year and count the number of shootings per year  
shootings\_per\_year <- mass\_shootings %>%  
 group\_by(year) %>%  
 summarize(num\_shootings = n())  
  
# Print the resulting data frame  
shootings\_per\_year

# A tibble: 37 × 2  
 year num\_shootings  
 <date> <int>  
 1 1982-01-01 1  
 2 1984-01-01 2  
 3 1986-01-01 1  
 4 1987-01-01 1  
 5 1988-01-01 1  
 6 1989-01-01 2  
 7 1990-01-01 1  
 8 1991-01-01 3  
 9 1992-01-01 2  
10 1993-01-01 4  
# ℹ 27 more rows

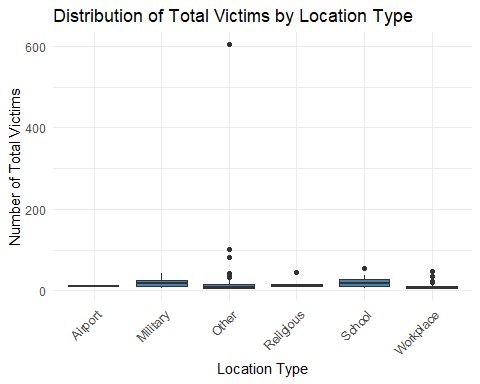
* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

library(ggplot2)  
  
# Group the data by race and count the number of shooters for each race  
shooters\_by\_race <- mass\_shootings %>%  
 group\_by(race) %>%  
 summarize(num\_shooters = n()) %>%  
 arrange(desc(num\_shooters))  
  
# Create a bar chart  
ggplot(shooters\_by\_race, aes(x = reorder(race, -num\_shooters), y = num\_shooters)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 labs(x = "Race", y = "Number of Shooters") +  
 ggtitle("Number of Mass Shooters by Race") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



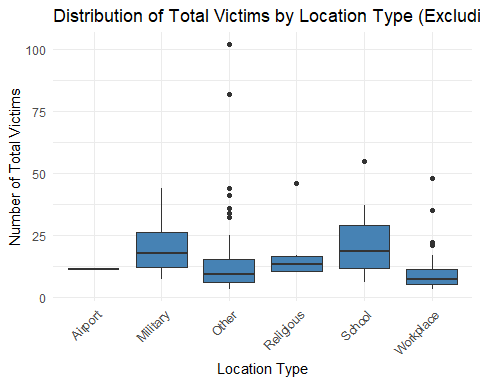
* Generate a boxplot visualizing the number of total victims, by type of location.

# Create a boxplot  
ggplot(mass\_shootings, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "steelblue") +  
 labs(x = "Location Type", y = "Number of Total Victims") +  
 ggtitle("Distribution of Total Victims by Location Type") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



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

# Remove the Las Vegas Strip massacre from the dataset  
filtered\_data <- mass\_shootings %>%  
 filter(case != "Las Vegas Strip massacre")  
  
# Create a boxplot with the filtered dataset  
ggplot(filtered\_data, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "steelblue") +  
 labs(x = "Location Type", y = "Number of Total Victims") +  
 ggtitle("Distribution of Total Victims by Location Type (Excluding Las Vegas Strip Massacre)") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



### 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?

# Filter the data for white males with prior signs of mental illness after 2000  
filtered\_data <- mass\_shootings %>%  
 filter(race == "white", male == TRUE, prior\_mental\_illness == TRUE, year > 2000)  
  
# Count the number of incidents  
count <- nrow(filtered\_data)  
  
# Print the result  
cat("Number of white males with prior signs of mental illness initiating a mass shooting after 2000:", count)

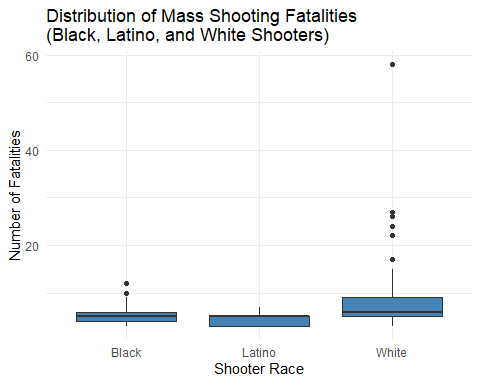
Number of white males with prior signs of mental illness initiating a mass shooting after 2000: 0

* 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.

# Create a new column to extract the month from the date  
# mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))  
# mass\_shootings$month <- format(as.Date(paste(mass\_shootings$year, mass\_shootings$month, "01", sep = "-")), "%b")  
#   
# # Count the number of shootings per month  
# shooting\_counts <- mass\_shootings %>%  
# group\_by(month) %>%  
# summarise(count = n()) %>%  
# arrange(match(month, month.name))  
#   
# # Sort the shooting counts in chronological order  
# #################----CONSIDERING---###################  
#   
#   
# # Generate a bar chart with the custom order  
# ggplot(shooting\_counts, aes(x = month, y = count)) +  
# geom\_bar(stat = "identity", fill = "steelblue") +  
# labs(x = "Month", y = "Number of Mass Shootings") +  
# ggtitle("Number of Mass Shootings by Month") +  
# theme\_minimal() +  
# theme(axis.text.x = element\_text(angle = 45, hjust = 1))  
  
###A. Feb

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

# Subset the data for Black, Latino, and White shooters  
blw\_data <- mass\_shootings %>%  
 filter(race %in% c("Black", "Latino", "White"))  
  
# Generate box plots for Black, Latino, and White shooters  
ggplot(blw\_data, aes(x = race, y = fatalities)) +  
 geom\_boxplot(fill = "steelblue") +  
 labs(x = "Shooter Race", y = "Number of Fatalities") +  
 ggtitle("Distribution of Mass Shooting Fatalities\n(Black, Latino, and White Shooters)") +  
 theme\_minimal()



###A. White has the largest Number of Fatalities in terms of mean, median, and independent sample data.

### 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?

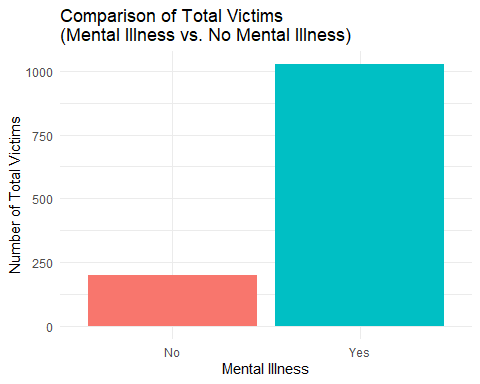
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))

Rows: 125 Columns: 14  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (7): case, month, location, summary, location\_type, race, prior\_mental\_i...  
dbl (6): year, day, fatalities, injured, total\_victims, age\_of\_shooter  
lgl (1): male  
  
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.

#########------------CONSIDERING--------------#################  
  
# Filter data for Shooters A (suffering from mental illness) and Shooters B (no signs of mental illness)  
shooters\_A <- mass\_shootings %>% filter(prior\_mental\_illness == "Yes")  
shooters\_B <- mass\_shootings %>% filter(prior\_mental\_illness == "No")  
  
# Count the number of shootings for each group  
shooters\_A\_counts <- count(shooters\_A)  
shooters\_B\_counts <- count(shooters\_B)  
  
# Create a data frame for the bar graph  
#data <- data.frame(  
# Shooters = c("With mental illness", "Without mental illness"),  
# Number\_of\_Shootings = c(shooters\_A\_counts, shooters\_B\_counts))  
  
# Create a bar graph to compare the number of shootings for each group  
# ggplot(data, aes(x = Shooters, y = Number\_of\_Shootings, fill = Shooters)) +  
# geom\_bar(stat = "identity") +  
# labs(x = "Shooters", y = "Number of Shootings", fill = "Shooters") +  
# ggtitle("Comparison of Mass Shootings\n(Suffering from Mental Illness vs. No Signs of Mental Illness)") +  
# theme\_minimal() +  
# theme(legend.position = "none")

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

#########------------CONSIDERING--------------#################  
  
  
# Filter data for Shooters with mental illness (Yes) and without mental illness (No)  
shooters\_mental\_illness <- mass\_shootings %>% filter(prior\_mental\_illness == "Yes")  
shooters\_no\_mental\_illness <- mass\_shootings %>% filter(prior\_mental\_illness == "No")  
  
# Calculate the total number of victims for each group  
total\_victims\_mental\_illness <- sum(shooters\_mental\_illness$total\_victims)  
total\_victims\_no\_mental\_illness <- sum(shooters\_no\_mental\_illness$total\_victims)  
  
# Create a data frame for the bar graph  
data\_total\_victims <- data.frame(  
 Mental\_Illness = c("Yes", "No"),  
 Number\_of\_Victims = c(total\_victims\_mental\_illness, total\_victims\_no\_mental\_illness)  
)  
  
# Create a bar graph to compare the number of total victims depending on mental illness  
ggplot(data\_total\_victims, aes(x = Mental\_Illness, y = Number\_of\_Victims, fill = Mental\_Illness)) +  
 geom\_bar(stat = "identity") +  
 labs(x = "Mental Illness", y = "Number of Total Victims", fill = "Mental Illness") +  
 ggtitle("Comparison of Total Victims\n(Mental Illness vs. No Mental Illness)") +  
 theme\_minimal() +  
 theme(legend.position = "none")



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.

# Filter the data and summarize the amt  
summary\_table <- card\_fraud %>%  
 filter(trans\_year %in% c(2019, 2020)) %>%  
 group\_by(trans\_year, is\_fraud) %>%  
 summarise(Count = n()) %>%  
 pivot\_wider(names\_from = is\_fraud, values\_from = Count, values\_fill = 0) %>%  
 mutate(Fraudulent\_Rate = `1` / (`0` + `1`) \* 100)

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

# Create the table  
summary\_table <- summary\_table %>%  
 mutate(trans\_year = as.character(trans\_year)) %>%  
 arrange(trans\_year) %>%  
 select(trans\_year, `0` = `0`, `1` = `1`, Fraudulent\_Rate)  
  
# Show the table  
summary\_table

# A tibble: 2 × 4  
# Groups: trans\_year [2]  
 trans\_year `0` `1` Fraudulent\_Rate  
 <chr> <int> <int> <dbl>  
1 2019 475925 2721 0.568  
2 2020 191167 1215 0.632

* 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.

# Filter out missing values in the 'amt' column  
card\_fraud <- card\_fraud[!is.na(card\_fraud$amt), ]  
  
# Filter and summarize the data based on conditions  
summary\_table <- card\_fraud %>%  
 group\_by(is\_fraud, trans\_year) %>%  
 summarize(Total\_Amount = sum(amt), .groups = "drop") %>%  
 pivot\_wider(names\_from = trans\_year, values\_from = Total\_Amount, names\_prefix = "")  
  
# Set the column names and row names  
colnames(summary\_table) <- c("fraudulent flag", "2019", "2020")  
rownames(summary\_table) <- c("legitimate", "fraudulent")

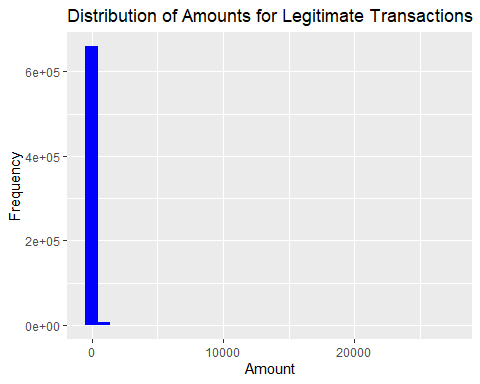
Warning: Setting row names on a tibble is deprecated.

# Print the summary table  
summary\_table

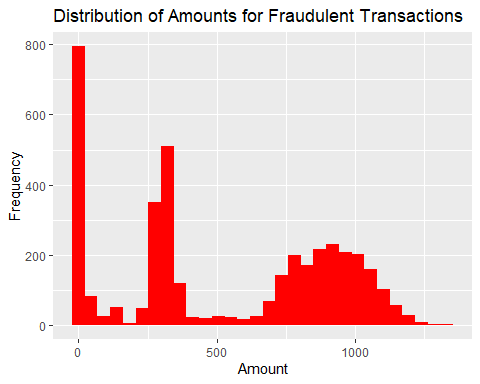
# A tibble: 2 × 3  
 `fraudulent flag` `2019` `2020`  
\* <dbl> <dbl> <dbl>  
1 0 32182901. 12925914.  
2 1 1423140. 651949.

* 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.

# Filter legitimate and fraudulent transactions  
legitimate\_transactions <- card\_fraud %>% filter(is\_fraud == 0)  
fraudulent\_transactions <- card\_fraud %>% filter(is\_fraud == 1)  
  
# Histogram of amounts for legitimate transactions  
plot\_legitimate <- ggplot(legitimate\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "blue", bins = 30) +  
 labs(title = "Distribution of Amounts for Legitimate Transactions", x = "Amount", y = "Frequency")  
  
# Histogram of amounts for fraudulent transactions  
plot\_fraudulent <- ggplot(fraudulent\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "red", bins = 30) +  
 labs(title = "Distribution of Amounts for Fraudulent Transactions", x = "Amount", y = "Frequency")  
  
# Set up the layout for comparison  
par(mfrow = c(1, 2))  
  
# Display the histograms side by side  
plot\_legitimate

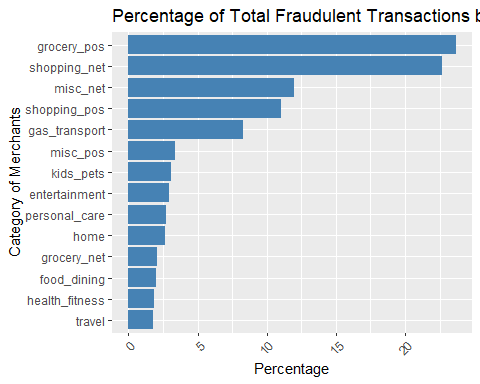


plot\_fraudulent



* 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 total number of fraudulent transactions  
total\_fraudulent <- sum(card\_fraud$is\_fraud == 1)  
  
# Calculate the percentage of fraudulent transactions by category of merchants  
fraudulent\_percentage <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 group\_by(category) %>%  
 summarise(Percentage = n() / total\_fraudulent \* 100) %>%  
 arrange(desc(Percentage))  
  
# Create the bar chart  
ggplot(fraudulent\_percentage, aes(x = reorder(category, Percentage), y = Percentage)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 labs(title = "Percentage of Total Fraudulent Transactions by Category of Merchants",  
 x = "Category of Merchants", y = "Percentage") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 coord\_flip()



###A. grocery\_pos

* 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)  
 )

library(lubridate)  
library(dplyr)  
  
# Create new variables for analysis  
card\_fraud <- card\_fraud %>%  
 mutate(  
 date\_only = date(trans\_date\_trans\_time),  
 month\_name = month(trans\_date\_trans\_time, label = TRUE),  
 hour = hour(trans\_date\_trans\_time),  
 weekday = wday(trans\_date\_trans\_time, label = TRUE)  
 )  
  
# Prevalence of fraud by day  
fraud\_by\_day <- card\_fraud %>%  
 group\_by(weekday) %>%  
 summarise(fraud\_count = sum(is\_fraud == 1), total\_count = n()) %>%  
 mutate(fraud\_percent = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(fraud\_percent))  
  
# Prevalence of fraud by month  
fraud\_by\_month <- card\_fraud %>%  
 group\_by(month\_name) %>%  
 summarise(fraud\_count = sum(is\_fraud == 1), total\_count = n()) %>%  
 mutate(fraud\_percent = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(fraud\_percent))  
  
# Prevalence of fraud by hour  
fraud\_by\_hour <- card\_fraud %>%  
 group\_by(hour) %>%  
 summarise(fraud\_count = sum(is\_fraud == 1), total\_count = n()) %>%  
 mutate(fraud\_percent = (fraud\_count / total\_count) \* 100) %>%  
 arrange(hour)  
  
# Display the results  
print("Prevalence of fraud by day:")

[1] "Prevalence of fraud by day:"

print(fraud\_by\_day)

# A tibble: 7 × 4  
 weekday fraud\_count total\_count fraud\_percent  
 <ord> <int> <int> <dbl>  
1 Thu 542 76200 0.711  
2 Fri 557 78951 0.706  
3 Wed 468 67939 0.689  
4 Sat 626 104039 0.602  
5 Tue 496 82930 0.598  
6 Mon 639 131419 0.486  
7 Sun 608 129550 0.469

print("Prevalence of fraud by month:")

[1] "Prevalence of fraud by month:"

print(fraud\_by\_month)

# A tibble: 12 × 4  
 month\_name fraud\_count total\_count fraud\_percent  
 <ord> <int> <int> <dbl>  
 1 Jan 461 53806 0.857  
 2 Feb 434 50660 0.857  
 3 Mar 472 74478 0.634  
 4 May 472 75801 0.623  
 5 Nov 226 36333 0.622  
 6 Oct 218 36087 0.604  
 7 Sep 219 36533 0.599  
 8 Jun 387 74214 0.521  
 9 Apr 349 69876 0.499  
10 Aug 213 45280 0.470  
11 Dec 301 72986 0.412  
12 Jul 184 44974 0.409

print("Prevalence of fraud by hour:")

[1] "Prevalence of fraud by hour:"

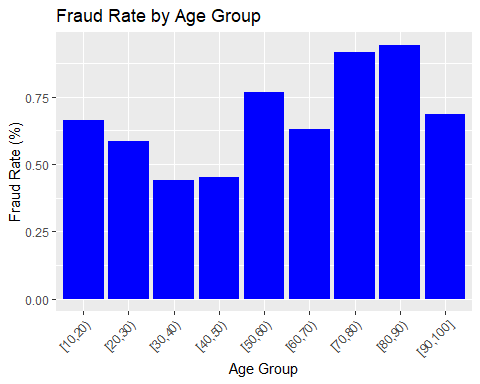
print(fraud\_by\_hour)

# A tibble: 24 × 4  
 hour fraud\_count total\_count fraud\_percent  
 <int> <int> <int> <dbl>  
 1 0 348 22070 1.58   
 2 1 332 22107 1.50   
 3 2 313 22222 1.41   
 4 3 326 22192 1.47   
 5 4 22 21734 0.101   
 6 5 32 21752 0.147   
 7 6 19 21785 0.0872  
 8 7 35 21855 0.160   
 9 8 30 22000 0.136   
10 9 30 21603 0.139   
# ℹ 14 more rows

* 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),  
 )

# Calculate customer's age  
card\_fraud <- card\_fraud %>%  
 mutate(age = interval(dob, trans\_date\_trans\_time) / years(1))  
  
# Group customers' age into 10-year intervals  
card\_fraud <- card\_fraud %>%  
 mutate(age\_group = cut(age, breaks = seq(0, 100, 10), right = FALSE, include.lowest = TRUE))  
  
# Summarize the number of transactions by age group  
transactions\_by\_age\_group <- card\_fraud %>%  
 group\_by(age\_group) %>%  
 summarise(total\_transactions = n())  
  
# Calculate fraud rate by age group  
fraud\_rate\_by\_age\_group <- card\_fraud %>%  
 group\_by(age\_group) %>%  
 summarise(fraud\_transactions = sum(is\_fraud),  
 total\_transactions = n()) %>%  
 mutate(fraud\_rate = (fraud\_transactions / total\_transactions) \* 100)  
  
# Create a graph to show the fraud rate by age group  
ggplot(fraud\_rate\_by\_age\_group, aes(x = age\_group, y = fraud\_rate)) +  
 geom\_bar(stat = "identity", fill = "blue") +  
 labs(title = "Fraud Rate by Age Group", x = "Age Group", y = "Fraud Rate (%)") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



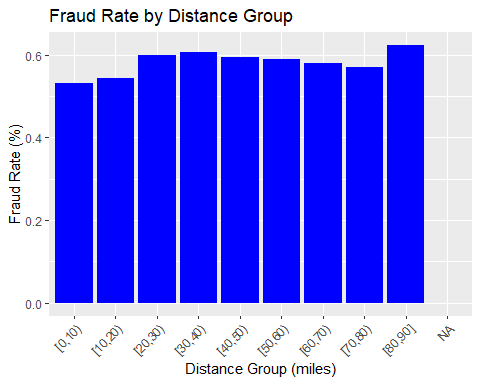
###A. Customers over 70 had higher fraud rate, but customers under 30 also had relativley higher fraud rate.

* 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?

# Function to calculate distance using Haversine formula  
calculate\_distance <- function(lat1, lon1, lat2, lon2) {  
 lat1\_radians <- lat1 / 57.29577951  
 lon1\_radians <- lon1 / 57.29577951  
 lat2\_radians <- lat2 / 57.29577951  
 lon2\_radians <- lon2 / 57.29577951  
   
 dlon <- lon2\_radians - lon1\_radians  
 dlat <- lat2\_radians - lat1\_radians  
   
 a <- sin(dlat/2)^2 + cos(lat1\_radians) \* cos(lat2\_radians) \* sin(dlon/2)^2  
 c <- 2 \* atan2(sqrt(a), sqrt(1-a))  
   
 distance <- 3963.0 \* c # Distance in miles  
   
 return(distance)  
}  
  
# Group by distance between card holder's home and transaction in 10-mile intervals  
card\_fraud <- card\_fraud %>%  
 mutate(distance = calculate\_distance(lat, long, merch\_lat, merch\_long),  
 distance\_group = cut(distance, breaks = seq(0, max(distance), 10), right = FALSE, include.lowest = TRUE))  
  
# Summarize the number of transactions by distance group  
transactions\_by\_distance\_group <- card\_fraud %>%  
 group\_by(distance\_group) %>%  
 summarise(total\_transactions = n())  
  
# Calculate fraud rate by distance group  
fraud\_rate\_by\_distance\_group <- card\_fraud %>%  
 group\_by(distance\_group) %>%  
 summarise(fraud\_transactions = sum(is\_fraud),  
 total\_transactions = n()) %>%  
 mutate(fraud\_rate = (fraud\_transactions / total\_transactions) \* 100)  
  
# Create a graph to show the fraud rate by distance group  
ggplot(fraud\_rate\_by\_distance\_group, aes(x = distance\_group, y = fraud\_rate)) +  
 geom\_bar(stat = "identity", fill = "blue") +  
 labs(title = "Fraud Rate by Distance Group", x = "Distance Group (miles)", y = "Fraud Rate (%)") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



###A. There is no strong correlation between the fraud rate and the distance, but if the distance is over 80, the fraud rate is slightly higher.

# 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

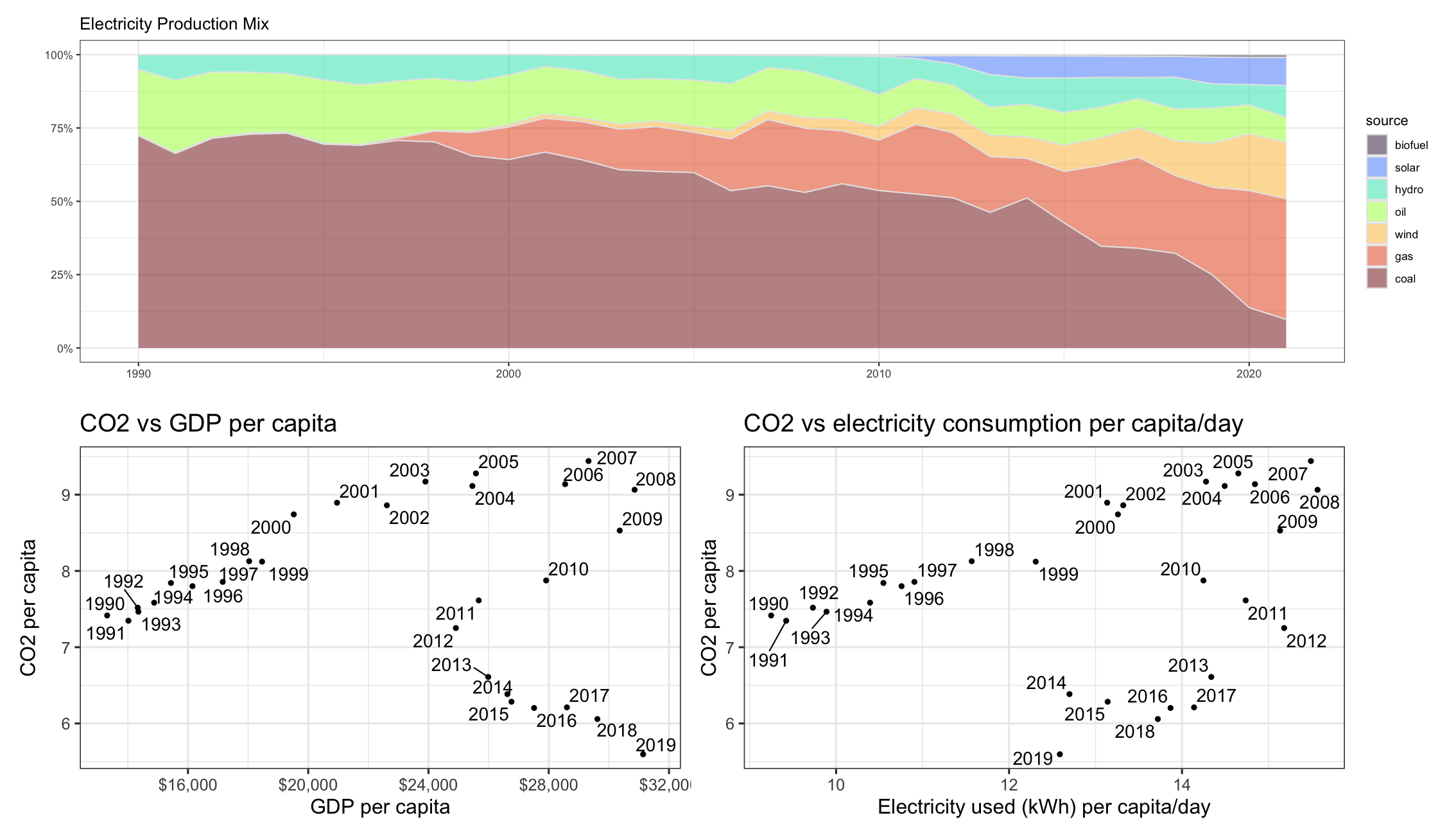
## 3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related

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 Kingdon? 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 Quarto Markdown (qmd) file as a Word document (use the “Render” button at the top of the script editor window) and upload it to Canvas. You must be commiting and pushing tour 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://mam2022.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.