Homerwork 2

Edward Davies

5/21/23

# Data Visualisation - Exploration

My git repo can be found at the following link:  
https://github.com/edwardhdavies/mydsb2023

# 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, …

## Explore the data

### Specific questions

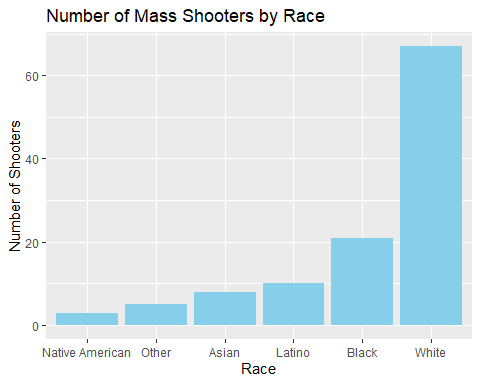
* Generate a data frame that summarizes the number of mass shootings per year.

shootings\_per\_year <- mass\_shootings %>%  
 group\_by(year) %>%  
 summarize(num\_incidents = n())  
  
# View the resulting data frame  
shootings\_per\_year

# A tibble: 37 × 2  
 year num\_incidents  
 <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

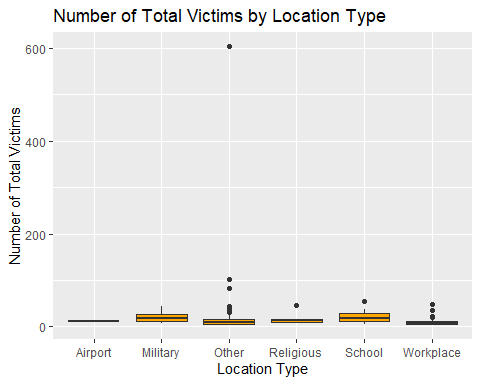
* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

# Filter out missing or unknown race values  
filtered\_data <- mass\_shootings %>%  
 filter(!is.na(race) & race != "Unknown")  
  
# Count the number of shooters for each race category  
shooters\_by\_race <- filtered\_data %>%  
 group\_by(race) %>%  
 summarize(num\_shooters = n()) %>%  
 arrange(desc(num\_shooters))  
  
ggplot(shooters\_by\_race, aes(x = reorder(race, num\_shooters), y = num\_shooters)) +  
 geom\_bar(stat = "identity", fill = "skyblue") +  
 labs(title = "Number of Mass Shooters by Race",  
 x = "Race",  
 y = "Number of Shooters")

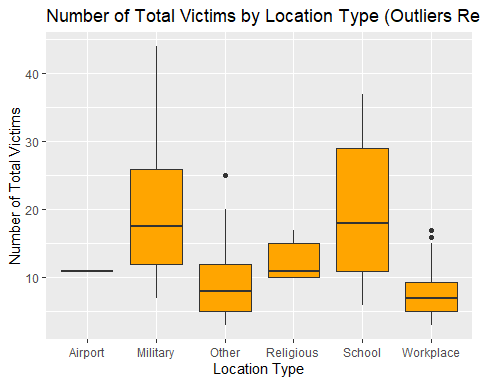


* Generate a boxplot visualizing the number of total victims, by type of location.

# Create the boxplot  
ggplot(mass\_shootings, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "orange") +  
 labs(title = "Number of Total Victims by Location Type",  
 x = "Location Type",  
 y = "Number of Total Victims")

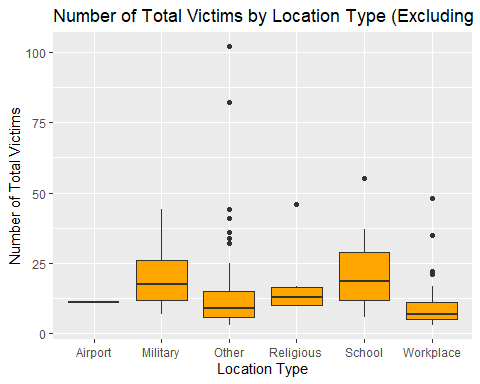


#This is an unreadable boxplot due to the outliers, particularly when one incident has over 600 total victims. I have redone the boxplot to remove these outliers.  
  
# Calculate the upper and lower bounds for outliers using the interquartile range (IQR)  
outlier\_threshold <- 1.5  
outlier\_bounds <- mass\_shootings %>%  
 group\_by(location\_type) %>%  
 summarize(  
 upper\_bound = quantile(total\_victims, 0.75) + outlier\_threshold \* IQR(total\_victims),  
 lower\_bound = quantile(total\_victims, 0.25) - outlier\_threshold \* IQR(total\_victims)  
 )  
  
# Filter out the outliers  
filtered\_data <- mass\_shootings %>%  
 inner\_join(outlier\_bounds, by = "location\_type") %>%  
 filter(total\_victims >= lower\_bound, total\_victims <= upper\_bound)  
  
# Create the boxplot  
ggplot(filtered\_data, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "orange") +  
 labs(title = "Number of Total Victims by Location Type (Outliers Removed)",  
 x = "Location Type",  
 y = "Number of Total Victims")



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

# Filter out the Las Vegas Strip massacre  
filtered\_data <- mass\_shootings %>%  
 filter(case != "Las Vegas Strip massacre")  
  
# Create the boxplot  
ggplot(filtered\_data, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "orange") +  
 labs(title = "Number of Total Victims by Location Type (Excluding Las Vegas Strip massacre)",  
 x = "Location Type",  
 y = "Number of Total Victims")



### 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?

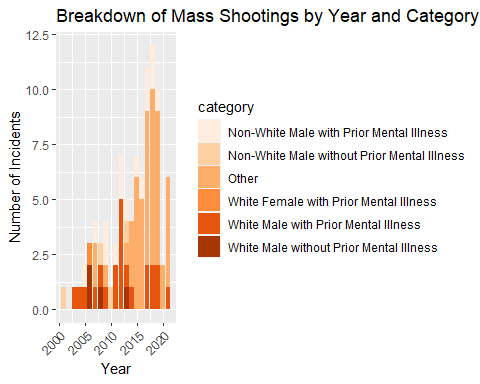
filtered\_data <- mass\_shootings %>%  
 filter(race == "White", male == TRUE, prior\_mental\_illness == "Yes", year > 2000)  
  
# Count the number of incidents  
num\_incidents <- nrow(filtered\_data)  
  
# Display the result  
num\_incidents

[1] 22

#22 mass shooting were initated by a white male with prior signs of mental illness after 2000.  
  
  
# Filter the mass shootings dataset  
filtered\_data <- mass\_shootings %>%  
 filter(year > 2000)  
  
# Create a new variable for the category based on race, gender, and mental illness  
filtered\_data <- filtered\_data %>%  
 mutate(category = case\_when(  
 race == "White" & male == TRUE & prior\_mental\_illness == "Yes" ~ "White Male with Prior Mental Illness",  
 race == "White" & male == TRUE & prior\_mental\_illness == "No" ~ "White Male without Prior Mental Illness",  
 race == "White" & male == FALSE & prior\_mental\_illness == "Yes" ~ "White Female with Prior Mental Illness",  
 race == "White" & male == FALSE & prior\_mental\_illness == "No" ~ "White Female without Prior Mental Illness",  
 race != "White" & male == TRUE & prior\_mental\_illness == "Yes" ~ "Non-White Male with Prior Mental Illness",  
 race != "White" & male == TRUE & prior\_mental\_illness == "No" ~ "Non-White Male without Prior Mental Illness",  
 race != "White" & male == FALSE & prior\_mental\_illness == "Yes" ~ "Non-White Female with Prior Mental Illness",  
 race != "White" & male == FALSE & prior\_mental\_illness == "No" ~ "Non-White Female without Prior Mental Illness",  
 TRUE ~ "Other"  
 ))  
  
# Group the data by year and category and calculate the count of incidents  
grouped\_data <- filtered\_data %>%  
 group\_by(year, category) %>%  
 summarize(count = n())

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

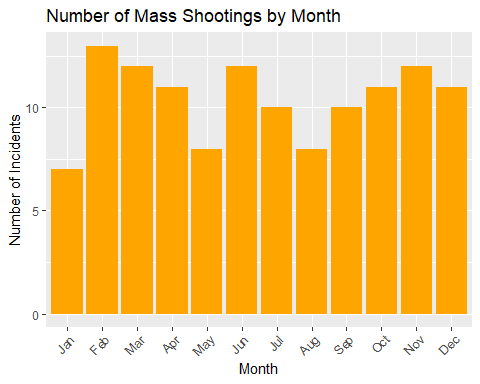
library(RColorBrewer)  
  
# Create the stacked bar chart  
ggplot(grouped\_data, aes(x = year, y = count, fill = category)) +  
 geom\_bar(stat = "identity") +  
 labs(title = "Breakdown of Mass Shootings by Year and Category",  
 x = "Year",  
 y = "Number of Incidents") +  
 scale\_fill\_brewer(palette = "Oranges") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



When focusing primarily on white males, since 2000, the proportion of white males without prior mental illness committing mass shootings has decreased. The reason for this decrease is likely due to improved mental health services, therefore identifying mental illnesses more efficiently. Despite this identification, mass shootings continue to occur.

* 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.

# Group the data by month and calculate the count of incidents  
grouped\_data <- mass\_shootings %>%  
 mutate(month = factor(month, levels = month.abb)) %>%  
 group\_by(month) %>%  
 summarize(count = n())  
  
# Sort the data in chronological order  
grouped\_data <- grouped\_data[order(as.integer(grouped\_data$month)), ]  
  
# Create the bar chart  
ggplot(grouped\_data, aes(x = month, y = count)) +  
 geom\_bar(stat = "identity", fill = "orange") +  
 labs(title = "Number of Mass Shootings by Month",  
 x = "Month",  
 y = "Number of Incidents") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

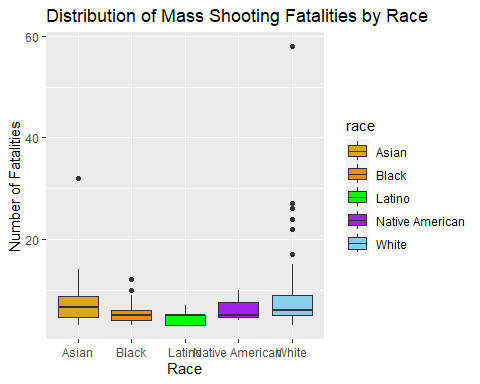


The month with the most mass shootings is February.

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

To analyse the distribution in mass shooting fatalities, we can generate a boxplot for white, latino and black shooters simultaneously:

# Filter out "Other" and NA values  
filtered\_data <- mass\_shootings %>%  
 filter(!is.na(race), race %in% c("White", "Black", "Latino", "Native American", "Asian"))  
  
# Create a boxplot of mass shooting fatalities by race  
ggplot(filtered\_data, aes(x = race, y = fatalities, fill = race)) +  
 geom\_boxplot() +  
 labs(title = "Distribution of Mass Shooting Fatalities by Race",  
 x = "Race",  
 y = "Number of Fatalities") +  
 scale\_fill\_manual(values = c("White" = "skyblue", "Black" = "darkorange", "Latino" = "green",  
 "Native American" = "purple", "Asian" = "goldenrod"))

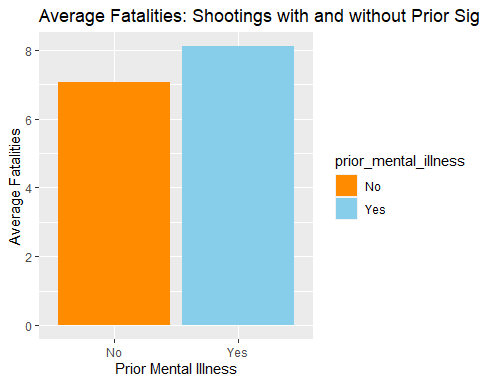


Rather than just analyse white shooters independently with black and latino shooters, we can analyse shooters from a variety of races with one another. Regarding fatality distribution, white shooters have more outliers on the higher end of fatalities, whereas black shooters have a more concentrated fatality distribution, with latino individuals having the most condensed fatality distribution. One paper [https://www.sciencedirect.com/science/article/abs/pii/S0091743522002250] suggests that the reason white shooters often cause more fatalities is due to their increased probability to be utlising a legally owned weapon, which is more often a highly lethal weapon such as an assault rifle. This is opposed to black or latino shooters more often utilising a weapon sourced through other means, which often is of a smaller calibre/magazine.

### 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?

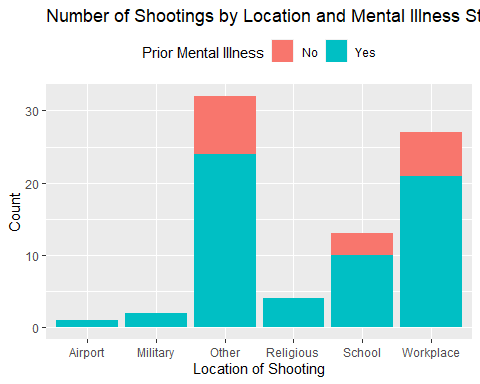
# Filter the dataset for mass shootings with information on mental illness  
filtered\_data <- mass\_shootings %>%  
 filter(!is.na(prior\_mental\_illness))  
  
# Descriptive Analysis  
summary\_stats <- filtered\_data %>%  
 group\_by(prior\_mental\_illness) %>%  
 summarize(  
 avg\_fatalities = mean(fatalities),  
 avg\_injured = mean(injured),  
 total\_victims = sum(total\_victims)  
 )  
  
  
# Visualizations  
# Bar chart of average fatalities by mental illness status  
ggplot(summary\_stats, aes(x = prior\_mental\_illness, y = avg\_fatalities, fill = prior\_mental\_illness)) +  
 geom\_bar(stat = "identity") +  
 labs(  
 title = "Average Fatalities: Shootings with and without Prior Signs of Mental Illness",  
 x = "Prior Mental Illness",  
 y = "Average Fatalities"  
 ) +  
 scale\_fill\_manual(values = c("Yes" = "skyblue", "No" = "darkorange"))



# Create a summary table with counts by location and mental illness status  
summary\_data <- filtered\_data %>%  
 group\_by(location\_type, prior\_mental\_illness) %>%  
 summarize(count = n())

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

# Create a stacked bar chart  
ggplot(summary\_data, aes(x = location\_type, y = count, fill = prior\_mental\_illness)) +  
 geom\_bar(stat = "identity") +  
 labs(  
 title = "Number of Shootings by Location and Mental Illness Status",  
 x = "Location of Shooting",  
 y = "Count",  
 fill = "Prior Mental Illness"  
 ) +  
 theme(legend.position = "top")



When looking at the visualisations above, there are clear trends we can determine from the behaviour of mass shooters who have or do not have prior mental illness.

Those with prior mental illness on average are more lethal in their mass shootings, with a slightly higher average number of fatalities.

More interestingly, locations for mass shootings such as Airports, Military areas and religious places have only been attacked by those with prior mental illness.

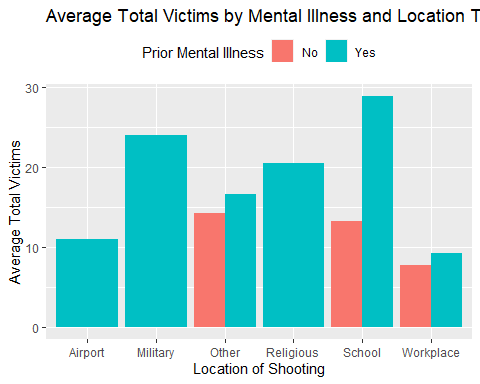
Although these are interesting trends to observe, we cannot make strong conclusions from this data. Firstly, a large proportion of the data does not possess data surrounding prior mental illness for many different incidents. Secondly, those that have not been identified to have prior mental illness did not necessairly not have prior mental illness, as this may not have been officially identified due to limited mental health support in some areas. Finally, the dataset is incomplete, as it does not possess all mass shootings that occurred within this time period.

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

# Filter the dataset for mass shootings with information on mental illness  
filtered\_data <- mass\_shootings %>%  
 filter(!is.na(prior\_mental\_illness))  
  
# Calculate average total victims by mental illness and location type  
summary\_data <- filtered\_data %>%  
 group\_by(prior\_mental\_illness, location\_type) %>%  
 summarize(avg\_total\_victims = mean(total\_victims))

`summarise()` has grouped output by 'prior\_mental\_illness'. You can override  
using the `.groups` argument.

# Create a grouped bar chart  
ggplot(summary\_data, aes(x = location\_type, y = avg\_total\_victims, fill = prior\_mental\_illness)) +  
 geom\_bar(stat = "identity", position = "dodge") +  
 labs(  
 title = "Average Total Victims by Mental Illness and Location Type",  
 x = "Location of Shooting",  
 y = "Average Total Victims",  
 fill = "Prior Mental Illness"  
 ) +  
 theme(legend.position = "top")



The prior graph gives us some interesting data to interpret.

As previously discussed, there is an interesting relationship between prior mental illness and location of shooting.

On a whole, when observing fatalities, as discussed previously, shooters within prior mental illness often carry out more fatal shootings. To determine whether this difference is statistically significant, we can carry out the following T-test.

# Filter the dataset for mass shootings with information on mental illness and total victims  
filtered\_data <- mass\_shootings %>%  
 filter(!is.na(prior\_mental\_illness) & !is.na(total\_victims))  
  
# Separate the data into two groups: shooters with and without prior mental illness  
group1 <- filtered\_data %>%  
 filter(prior\_mental\_illness == "Yes") %>%  
 pull(total\_victims)  
  
group2 <- filtered\_data %>%  
 filter(prior\_mental\_illness == "No") %>%  
 pull(total\_victims)  
  
# Perform t-test to compare the means of the two groups  
t\_test\_result <- t.test(group1, group2)  
  
t\_test\_result

Welch Two Sample t-test  
  
data: group1 and group2  
t = 1.6718, df = 41.324, p-value = 0.1021  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -0.9814223 10.4311377  
sample estimates:  
mean of x mean of y   
 16.54839 11.82353

With this returning a p-value of 0.1021, we can determine that this difference in fatalities between shooters with and without prior mental illness is not significant.

Finally, locations such as schools and military areas often have higher fatalities on average. For schools, I suggest that this is due to the high density of victims in a viscinity. For military bases, I would suggest that this is due to ready access to highly lethal weapons, alongside highly skilled shooters due to their previous military training.

# Exploring credit card fraud

## Obtain the data

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, …

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

# Filter the dataset for fraudulent transactions  
fraudulent\_transactions <- card\_fraud %>%  
 filter(is\_fraud == 1)  
  
# Summarize the number and frequency of fraudulent transactions per year  
fraud\_summary <- fraudulent\_transactions %>%  
 group\_by(trans\_year) %>%  
 summarize(  
 num\_fraudulent\_transactions = n(),  
 frequency = num\_fraudulent\_transactions / nrow(fraudulent\_transactions)  
 )  
  
fraud\_summary

# A tibble: 2 × 3  
 trans\_year num\_fraudulent\_transactions frequency  
 <dbl> <int> <dbl>  
1 2019 2721 0.691  
2 2020 1215 0.309

* 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.

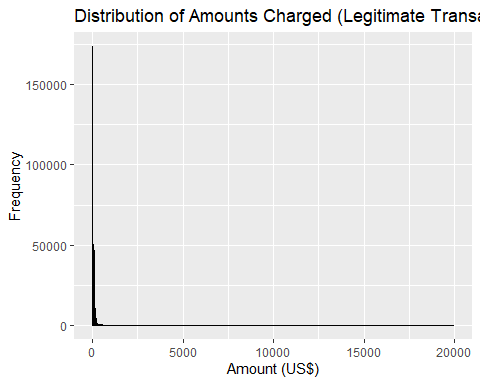
# Calculate the total amount of legitimate transactions per year  
legitimate\_transactions <- card\_fraud %>%  
 filter(is\_fraud == 0) %>%  
 group\_by(trans\_year) %>%  
 summarize(total\_legitimate\_amount = sum(amt))  
  
# Calculate the total amount of fraudulent transactions per year  
fraudulent\_transactions <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 group\_by(trans\_year) %>%  
 summarize(total\_fraudulent\_amount = sum(amt))  
  
# Merge the legitimate and fraudulent transactions data  
transaction\_summary <- merge(legitimate\_transactions, fraudulent\_transactions, by = "trans\_year", all = TRUE)  
  
# Calculate the percentage of fraudulent transactions in US dollar terms  
transaction\_summary <- transaction\_summary %>%  
 mutate(percentage\_fraudulent = (total\_fraudulent\_amount / (total\_legitimate\_amount + total\_fraudulent\_amount)) \* 100)  
  
transaction\_summary

trans\_year total\_legitimate\_amount total\_fraudulent\_amount  
1 2019 32182901 1423140.0  
2 2020 12925914 651949.2  
 percentage\_fraudulent  
1 4.234774  
2 4.801560

* 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.

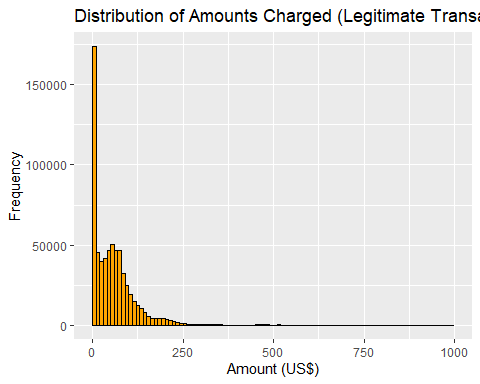
# Subset the data for legitimate and fraudulent transactions  
legitimate\_transactions <- card\_fraud %>%  
 filter(is\_fraud == 0)  
fraudulent\_transactions <- card\_fraud %>%  
 filter(is\_fraud == 1)  
  
# Plot histogram for legitimate transactions  
ggplot(legitimate\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "orange", color = "black", bins = 30, breaks = seq(0, 20000, 10)) +  
 scale\_x\_continuous(limits = c(0, 20000)) +  
 labs(title = "Distribution of Amounts Charged (Legitimate Transactions)",  
 x = "Amount (US$)", y = "Frequency")

Warning: Removed 3 rows containing non-finite values (`stat\_bin()`).



# Make the graph more visually appealing, through reducing the limits to 1000  
ggplot(legitimate\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "orange", color = "black", bins = 30, breaks = seq(0, 1000, 10)) +  
 scale\_x\_continuous(limits = c(0, 1000)) +  
 labs(title = "Distribution of Amounts Charged (Legitimate Transactions)",  
 x = "Amount (US$)", y = "Frequency")

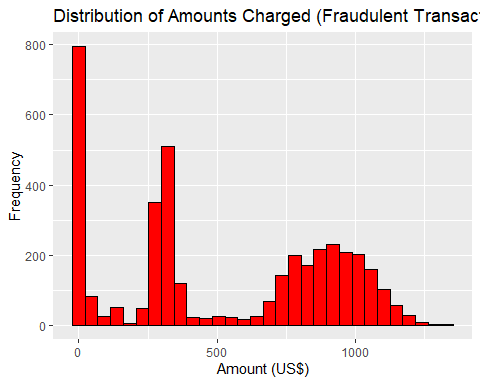
Warning: Removed 1535 rows containing non-finite values (`stat\_bin()`).



# Calculate summary statistics for legitimate transactions  
legitimate\_summary <- legitimate\_transactions %>%  
 summarise(  
 count = n(),  
 mean\_amount = mean(amt),  
 median\_amount = median(amt),  
 min\_amount = min(amt),  
 max\_amount = max(amt)  
 )  
print(legitimate\_summary)

# A tibble: 1 × 5  
 count mean\_amount median\_amount min\_amount max\_amount  
 <int> <dbl> <dbl> <dbl> <dbl>  
1 667092 67.6 47.2 1 27120.

# Plot histogram for fraudulent transactions  
ggplot(fraudulent\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "red", color = "black", bins = 30) +  
 labs(title = "Distribution of Amounts Charged (Fraudulent Transactions)",  
 x = "Amount (US$)", y = "Frequency")

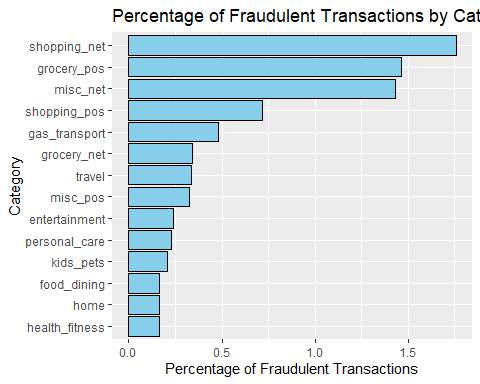


# Calculate summary statistics for fraudulent transactions  
fraudulent\_summary <- fraudulent\_transactions %>%  
 summarise(  
 count = n(),  
 mean\_amount = mean(amt),  
 median\_amount = median(amt),  
 min\_amount = min(amt),  
 max\_amount = max(amt)  
 )  
print(fraudulent\_summary)

# A tibble: 1 × 5  
 count mean\_amount median\_amount min\_amount max\_amount  
 <int> <dbl> <dbl> <dbl> <dbl>  
1 3936 527. 369. 1.06 1334.

* 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 category  
fraud\_percentage <- card\_fraud %>%  
 group\_by(category) %>%  
 summarize(fraud\_rate = sum(is\_fraud) / n()) %>%  
 arrange(desc(fraud\_rate))  
  
# Create a bar chart of the percentage of fraudulent transactions by category  
ggplot(fraud\_percentage, aes(x = reorder(category, fraud\_rate), y = fraud\_rate \* 100)) +  
 geom\_bar(stat = "identity", fill = "skyblue", color = "black") +  
 labs(title = "Percentage of Fraudulent Transactions by Category",  
 x = "Category", y = "Percentage of Fraudulent Transactions") +  
 coord\_flip()



Shopping and grocery are the most likely transactions to be fraudulent.

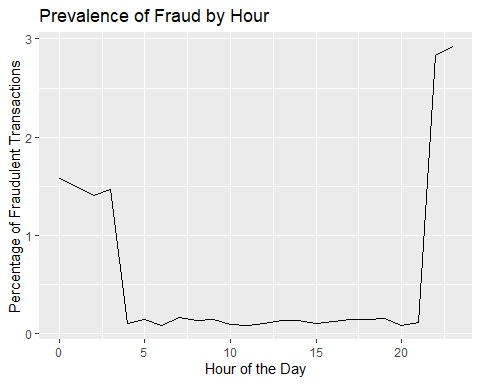
* When is fraud more prevalent? Which days, months, hours? To create new variables to help you in your analysis, we use the lubridate package and the following code

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)  
  
  
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)  
 )

Firstly, let us look at fraudulent transactions by hour.

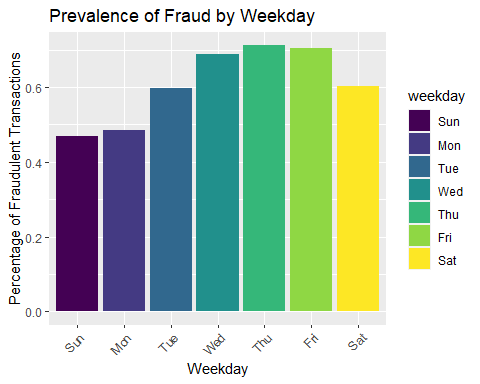
fraud\_by\_hour <- card\_fraud %>%  
 group\_by(hour) %>%  
 summarize(fraud\_count = sum(is\_fraud),  
 total\_count = n(),  
 fraud\_percentage = fraud\_count / total\_count \* 100)  
  
# Line plot  
ggplot(fraud\_by\_hour, aes(x = hour, y = fraud\_percentage)) +  
 geom\_line() +  
 labs(title = "Prevalence of Fraud by Hour",  
 x = "Hour of the Day",  
 y = "Percentage of Fraudulent Transactions")



On an hourly basis, fraudulent transactions are most common between 9pm and 4am. This makes sense as those committing fraud will do this so that their victims are likely asleep and therefore unable to stop the fraud in progress.

Secondly, let us look at fraudulent transactions by weekday.

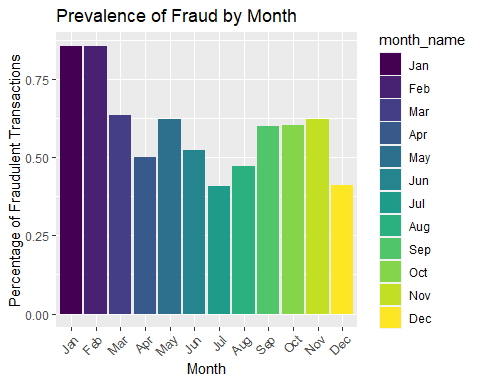
fraud\_by\_weekday <- card\_fraud %>%  
 group\_by(weekday) %>%  
 summarize(fraud\_count = sum(is\_fraud),  
 total\_count = n(),  
 fraud\_percentage = fraud\_count / total\_count \* 100) %>%  
 arrange(desc(fraud\_percentage))  
  
# Bar chart  
ggplot(fraud\_by\_weekday, aes(x = weekday, y = fraud\_percentage, fill = weekday)) +  
 geom\_col() +  
 labs(title = "Prevalence of Fraud by Weekday",  
 x = "Weekday",  
 y = "Percentage of Fraudulent Transactions") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



The most common days for fraudulent transactions are Wednesday, Thursday and Friday.

Finally, let us look at fraudulent transactions by month.

fraud\_by\_month <- card\_fraud %>%  
 group\_by(month\_name) %>%  
 summarize(fraud\_count = sum(is\_fraud),  
 total\_count = n(),  
 fraud\_percentage = fraud\_count / total\_count \* 100) %>%  
 arrange(desc(fraud\_percentage))  
  
# Bar chart  
ggplot(fraud\_by\_month, aes(x = month\_name, y = fraud\_percentage, fill = month\_name)) +  
 geom\_col() +  
 labs(title = "Prevalence of Fraud by Month",  
 x = "Month",  
 y = "Percentage of Fraudulent Transactions") +  
 scale\_x\_discrete(limits = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))



The most common months for fraud are January and February.

* 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),  
 )

card\_fraud <- card\_fraud %>%  
 mutate(age = interval(dob, trans\_date\_trans\_time) / years(1))  
  
# Create age groups  
card\_fraud <- card\_fraud %>%  
 mutate(age\_group = cut(age, breaks = c(0, 30, 40, 50, Inf),  
 labels = c("Under 30", "30-40", "40-50", "50+"),  
 include.lowest = TRUE))  
  
# Calculate fraud rates by age group  
fraud\_rates <- card\_fraud %>%  
 group\_by(age\_group) %>%  
 summarize(fraud\_rate = mean(is\_fraud) \* 100)  
  
# Conduct chi-square test  
chisq\_result <- chisq.test(card\_fraud$is\_fraud, card\_fraud$age\_group)  
  
# Print fraud rates and chi-square test result  
print(fraud\_rates)

# A tibble: 4 × 2  
 age\_group fraud\_rate  
 <fct> <dbl>  
1 Under 30 0.598  
2 30-40 0.441  
3 40-50 0.453  
4 50+ 0.763

print(chisq\_result)

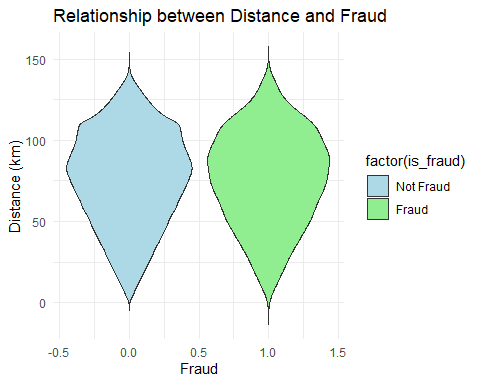
Pearson's Chi-squared test  
  
data: card\_fraud$is\_fraud and card\_fraud$age\_group  
X-squared = 229.01, df = 3, p-value < 2.2e-16

When observing the fraud rates directly, it is clear that both the age group of under 30 and age group of over 50 are the most likely to go through fraud. This explainable through those under 30 likely being less responsible with their security, leading to increased rate of fraud, while those over 50 are likely less familiar with the technology and therefore are more susceptible to fraud as well.

When viewing the results of our chi-squared test, it is clear that an increased age does correlate with increased risk of fraud.

* 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/  
  
  
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))  
  
 )  
  
# Plotting the violin plot  
ggplot(fraud, aes(x = is\_fraud, y = distance\_km, fill = factor(is\_fraud))) +  
 geom\_violin(trim = FALSE) +  
 labs(title = "Relationship between Distance and Fraud",  
 x = "Fraud",  
 y = "Distance (km)") +  
 scale\_fill\_manual(values = c("lightblue", "lightgreen"), labels = c("Not Fraud", "Fraud")) +  
 theme\_minimal()



This is not very useful in order to determine the relationship between distance of transcation and fraud. THis is seen above as the violin plots for distance are very similar whether the transaction has been identified as fraud or as a legitimate transaction. Due to this, it cannot be used to determine the probability of a transaction being fraudulent.

# 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.

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)  
  
glimpse(co2\_percap)

Rows: 5,729  
Columns: 7  
$ indicator\_id <chr> "EN.ATM.CO2E.PC", "EN.ATM.CO2E.PC", "EN.ATM.CO2E.PC", "EN…  
$ indicator <chr> "CO2 emissions (metric tons per capita)", "CO2 emissions …  
$ iso2c <chr> "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF…  
$ iso3c <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "…  
$ country <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan…  
$ year <dbl> 2019, 2018, 2017, 2016, 2015, 2014, 2013, 2012, 2011, 201…  
$ co2percap <dbl> 0.16085016, 0.16536691, 0.13418691, 0.15312098, 0.1761799…

# 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)  
  
glimpse(gdp\_percap)

Rows: 5,898  
Columns: 7  
$ indicator\_id <chr> "NY.GDP.PCAP.PP.KD", "NY.GDP.PCAP.PP.KD", "NY.GDP.PCAP.PP…  
$ indicator <chr> "GDP per capita, PPP (constant 2017 international $)", "G…  
$ iso2c <chr> "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF", "AF…  
$ iso3c <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "…  
$ country <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanistan…  
$ year <dbl> 2021, 2020, 2019, 2018, 2017, 2016, 2015, 2014, 2013, 201…  
$ GDPpercap <dbl> 1516.306, 1968.341, 2079.922, 2060.699, 2096.093, 2101.42…

Specific questions:

1. How would you turn energy to long, tidy format?

glimpse(energy)

Rows: 6,951  
Columns: 18  
$ country <chr> "Afghanistan", "Afghanistan", "Afghanistan", "A…  
$ year <dbl> 1990, 1991, 1992, 1993, 1994, 1995, 1996, 1997,…  
$ iso\_code <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG"…  
$ biofuel <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0…  
$ coal <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0…  
$ gas <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0…  
$ hydro <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.31, 0…  
$ nuclear <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0…  
$ oil <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.16, 0…  
$ other\_renewable <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0…  
$ solar <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.00, 0…  
$ wind <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0, 0, 0…  
$ electricity\_demand <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.57, 0…  
$ electricity\_generation <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.47, 0…  
$ net\_elec\_imports <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 0.10, 0…  
$ energy\_per\_capita <dbl> 2968.316, 1293.862, 760.342, 650.207, 570.403, …  
$ energy\_per\_gdp <dbl> 2.430, 1.154, 0.723, 0.926, 1.113, 0.604, 0.604…  
$ per\_capita\_electricity <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 24.050,…

energy\_long <- energy %>%  
 pivot\_longer(  
 cols = biofuel:per\_capita\_electricity,  
 names\_to = "energy\_source",  
 values\_to = "value",  
 names\_prefix = "energy\_"  
 )  
  
glimpse(energy\_long)

Rows: 104,265  
Columns: 5  
$ country <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afghanista…  
$ year <dbl> 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 1990, 19…  
$ iso\_code <chr> "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", "AFG", …  
$ energy\_source <chr> "biofuel", "coal", "gas", "hydro", "nuclear", "oil", "ot…  
$ value <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 2968.316…

1. 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.

# Rename iso\_code column in energy\_long table  
energy\_long <- energy\_long %>%  
 rename(iso3c = iso\_code)  
  
  
# Join the tables  
joined\_data <- left\_join(energy\_long, gdp\_percap, by = c("iso3c", "year")) %>%  
 left\_join(co2\_percap, by = c("iso3c", "year"))

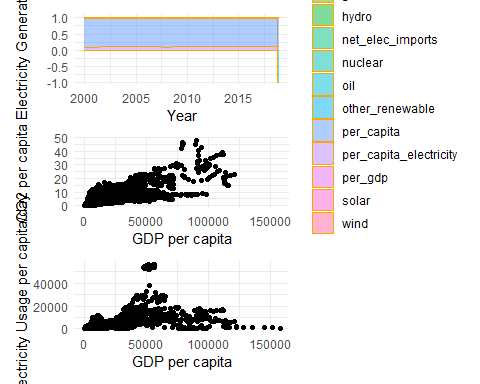
1. Write a function that takes as input any country’s name and returns all three graphs. You can use the patchwork package to arrange the three graphs as shown below

library(patchwork)  
  
generate\_country\_plots <- function(country\_name) {  
   
 # Filter data for the specified country  
 your\_country <- joined\_data %>% filter(country == country\_name, year >= 2000)  
   
 # Stacked Area Chart  
 area\_chart <- ggplot(your\_country, aes(x = year, y = value, fill = energy\_source)) +  
 geom\_area(colour = "orange", alpha = 0.5, position = "fill") +  
 labs(x = "Year", y = "Electricity Generation", fill = "Energy Source") +  
 theme\_minimal()  
   
 # Scatter Plot (CO2 per capita vs. GDP per capita)  
 scatter\_plot\_co2\_gdp <- ggplot(joined\_data, aes(x = GDPpercap, y = co2percap)) +  
 geom\_point() +  
 labs(x = "GDP per capita", y = "CO2 per capita") +  
 theme\_minimal()  
   
 # Scatter Plot (Electricity Usage per capita/day vs. GDP per capita)  
 scatter\_plot\_electricity <- ggplot(joined\_data %>% filter(energy\_source == "per\_capita\_electricity"), aes(x = GDPpercap, y = value)) +  
 geom\_point() +  
 labs(x = "GDP per capita", y = "Electricity Usage per capita/day") +  
 theme\_minimal()  
   
 # Arrange the plots using patchwork  
 all\_plots <- area\_chart + scatter\_plot\_co2\_gdp + scatter\_plot\_electricity  
 all\_plots <- all\_plots + plot\_layout(ncol = 1)  
   
 # Return the combined plots  
 return(all\_plots)  
}  
  
# Example usage: Generate plots for "Morocco"  
plots <- generate\_country\_plots("Morocco")  
plots

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

Warning: Removed 26535 rows containing missing values (`geom\_point()`).

Warning: Removed 2183 rows containing missing values (`geom\_point()`).



The data produced here is very interesting.When observing Morocco’s energy production, we can see that in recent years they have seen a dramatic increase in net electiricty imports, likely due to the steady increase in the manufacturing industry in the region.

When observing GDP per Capita compared to CO2 emissions, it is clear that countries with a higher GDP per capita have higher CO2 emissions. This makes sense as countries with higher GDP per capita are often more developed and therefore require far more energy usage per capita to maintain the lifestyles of their citizens, through services and infrastructure. As the majority of energy is still produced through fossil fuels, this explains the higher CO2 emissions.

When observing GDP per capita in comparison to energy usage per capita, there again appears to be a steady positive correlation. However, there are a collection of outliers with medium GDP per capitas which consume a large amount of energy. This is likely due to rapidly developing countries which have very strong domestic industries, such as India and China.

# 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: N/A
* Approximately how much time did you spend on this problem set: 7 hours
* What, if anything, gave you the most trouble: using git after adding card fraud dataset

**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.