Homerwork 2

GAING IGNACIO

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

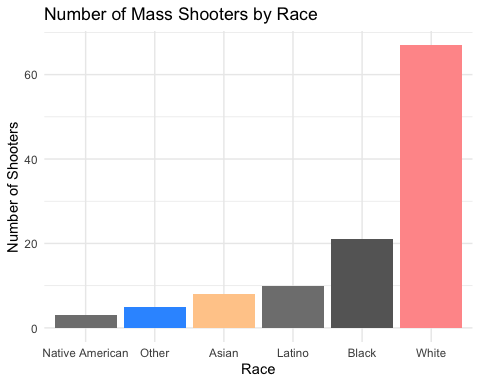
* Generate a data frame that summarizes the number of mass shootings per year.

# Group the data by year and count the number of shootings  
shootings\_per\_year <- mass\_shootings %>%  
 group\_by(year) %>%  
 summarise(Shootings = n())  
  
# View the resulting data frame  
print(shootings\_per\_year)

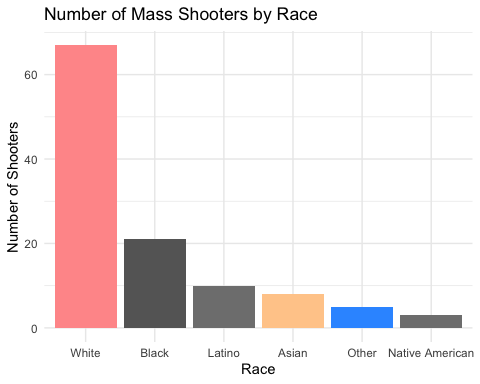
# A tibble: 37 × 2  
 year Shootings  
 <dbl> <int>  
 1 1982 1  
 2 1984 2  
 3 1986 1  
 4 1987 1  
 5 1988 1  
 6 1989 2  
 7 1990 1  
 8 1991 3  
 9 1992 2  
10 1993 4  
# ℹ 27 more rows

* 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.

library(ggplot2)  
  
# Create a custom color palette  
my\_colors <- c("White" = "#FF9999", "Black" = "#666666", "Asian" = "#FFCC99", "Other" = "#3399FF")  
  
# Reorder the data frame by the number of shooters in descending order and eliminate the NA  
mass\_shootings %>%   
 count(race, sort=TRUE) %>%   
 drop\_na(race) %>%   
 mutate(race = fct\_reorder(race, n)) %>%   
   
   
# Create the bar chart with ordered bars and custom colors  
 ggplot( aes(x = race, y = n, fill = race)) +  
 geom\_col()+  
 scale\_fill\_manual(values = my\_colors) +  
 labs(x = "Race", y = "Number of Shooters") +  
 ggtitle("Number of Mass Shooters by Race") +  
 theme\_minimal()+  
 theme(legend.position = "none")

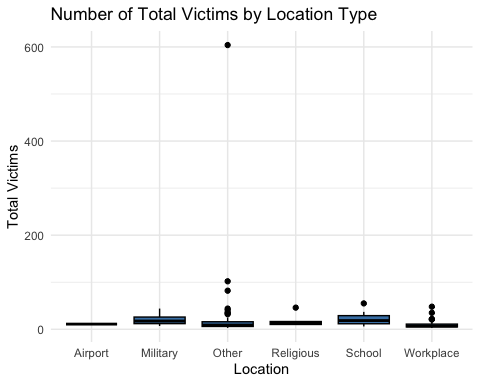


# Create the bar chart with ordered bars and custom colors in the other way from max to min  
ggplot(mass\_shootings %>%   
 count(race, sort = TRUE) %>%   
 drop\_na(race) %>%   
 mutate(race = fct\_reorder(race, n)),  
 aes(x = reorder(race, -n), y = n, fill = race)) +  
 geom\_col() +  
 scale\_fill\_manual(values = my\_colors) +  
 labs(x = "Race", y = "Number of Shooters") +  
 ggtitle("Number of Mass Shooters by Race") +  
 theme\_minimal() +  
 theme(legend.position = "none")



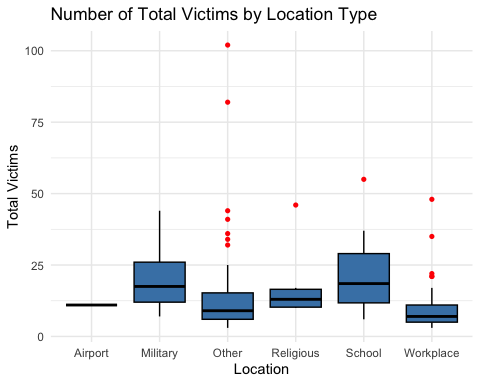
* Generate a boxplot visualizing the number of total victims, by type of location.

library(ggplot2)  
  
# Create the boxplot  
ggplot(mass\_shootings, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "steelblue", color = "black") +  
 labs(x = "Location", y = "Total Victims") +  
 ggtitle("Number of Total Victims by Location Type") +  
 theme\_minimal()



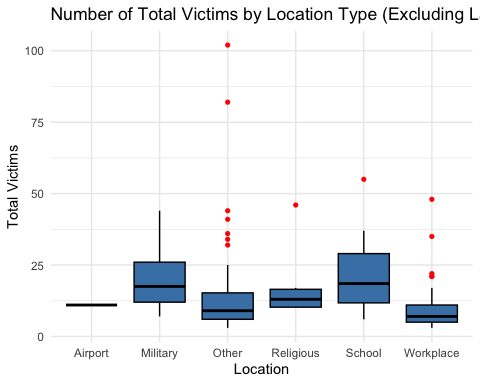
# Preprocess the data to remove the outlier in the "Others" location type to make a better chart   
processed\_data <- mass\_shootings  
processed\_data$total\_victims[processed\_data$location\_type == "Other" & processed\_data$total\_victims == 604] <- NA  
  
# Create the boxplot with outliers, excluding the outlier in the "Others" location type  
ggplot(processed\_data, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "steelblue", color = "black", outlier.colour = "red", outlier.shape = 16) +  
 labs(x = "Location", y = "Total Victims") +  
 ggtitle("Number of Total Victims by Location Type") +  
 theme\_minimal()

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



* 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 without the Las Vegas Strip massacre  
ggplot(filtered\_data, aes(x = location\_type, y = total\_victims)) +  
 geom\_boxplot(fill = "steelblue", color = "black", outlier.colour = "red", outlier.shape = 16) +  
 labs(x = "Location", y = "Total Victims") +  
 ggtitle("Number of Total Victims by Location Type (Excluding Las Vegas Strip massacre)") +  
 theme\_minimal()



### More open-ended questions

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

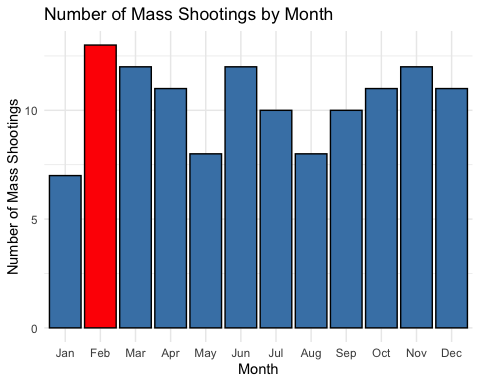
* How many white males with prior signs of mental illness initiated a mass shooting after 2000?

# Filter the dataset based on the specified criteria  
filtered\_data <- mass\_shootings %>%  
 filter(race == "White", male == "TRUE", `prior\_mental\_illness` == "Yes", year > 2000)  
  
# Count the number of filtered cases  
num\_cases <- nrow(filtered\_data)  
  
# Display the result as text  
cat("The number of white males with prior signs of mental illness who initiated a mass shooting after 2000 is:", num\_cases)

The number of white males with prior signs of mental illness who initiated a mass shooting after 2000 is: 22

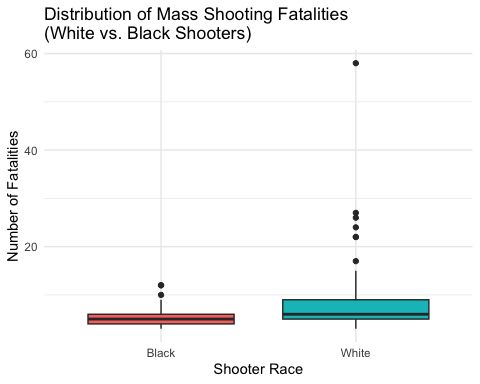
* Which month of the year has the most mass shootings? Generate a bar chart sorted in chronological (natural) order (Jan-Feb-Mar- etc) to provide evidence of your answer.

# Define the order of the months  
month\_order <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun",  
 "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")  
  
# Calculate the count of mass shootings for each month  
shootings\_by\_month <- mass\_shootings %>%  
 count(month) %>%  
 arrange(match(month, month\_order))  
  
# Convert the month column to a factor with the desired order  
shootings\_by\_month$month <- factor(shootings\_by\_month$month, levels = month\_order)  
  
# Determine the month with the maximum number of mass shootings  
max\_month <- shootings\_by\_month$month[which.max(shootings\_by\_month$n)]  
max\_count <- max(shootings\_by\_month$n)  
  
# Create the bar chart  
ggplot(shootings\_by\_month, aes(x = month, y = n)) +  
 geom\_bar(stat = "identity", fill = ifelse(shootings\_by\_month$month == max\_month, "red", "steelblue"), color = "black") +  
 labs(x = "Month", y = "Number of Mass Shootings") +  
 ggtitle("Number of Mass Shootings by Month") +  
 theme\_minimal() +  
 theme(legend.position = "none")

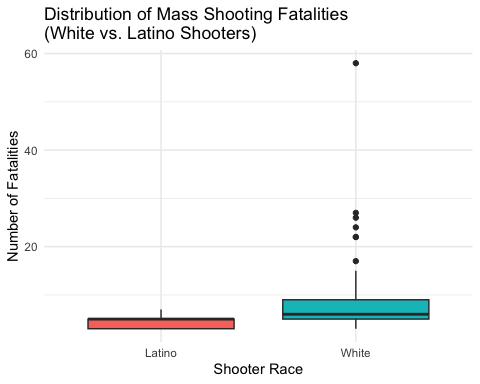


* How does the distribution of mass shooting fatalities differ between White and Black shooters? What about White and Latino shooters?

# Filter the data for relevant groups (White and Black shooters)  
filtered\_data <- mass\_shootings %>%  
 filter(race %in% c("White", "Black"))  
  
# Create a boxplot to compare the distribution of fatalities between White and Black shooters  
ggplot(filtered\_data, aes(x = race, y = fatalities, fill = race)) +  
 geom\_boxplot() +  
 labs(x = "Shooter Race", y = "Number of Fatalities") +  
 ggtitle("Distribution of Mass Shooting Fatalities\n(White vs. Black Shooters)") +  
 theme\_minimal() +  
 theme(legend.position = "none")



# Filter the data for relevant groups (White and Latino shooters)  
filtered\_data <- mass\_shootings %>%  
 filter(race %in% c("White", "Latino"))  
  
# Create a boxplot to compare the distribution of fatalities between White and Latino shooters  
ggplot(filtered\_data, aes(x = race, y = fatalities, fill = race)) +  
 geom\_boxplot() +  
 labs(x = "Shooter Race", y = "Number of Fatalities") +  
 ggtitle("Distribution of Mass Shooting Fatalities\n(White vs. Latino Shooters)") +  
 theme\_minimal() +  
 theme(legend.position = "none")



# Calculate summary statistics  
summary\_white <- summary(mass\_shootings$fatalities[mass\_shootings$race == "White"])  
summary\_black <- summary(mass\_shootings$fatalities[mass\_shootings$race == "Black"])  
summary\_latino <- summary(mass\_shootings$fatalities[mass\_shootings$race == "Latino"])  
  
# Conduct statistical tests  
ttest\_white\_black <- t.test(mass\_shootings$fatalities[mass\_shootings$race == "White"],  
 mass\_shootings$fatalities[mass\_shootings$race == "Black"])  
ttest\_white\_latino <- t.test(mass\_shootings$fatalities[mass\_shootings$race == "White"],  
 mass\_shootings$fatalities[mass\_shootings$race == "Latino"])  
  
# Calculate effect sizes  
effect\_size\_white\_black <- abs(ttest\_white\_black$estimate / sqrt(ttest\_white\_black$parameter))  
effect\_size\_white\_latino <- abs(ttest\_white\_latino$estimate / sqrt(ttest\_white\_latino$parameter))  
  
# Print the results  
cat("Summary Statistics for White Shooters:\n")

Summary Statistics for White Shooters:

print(summary\_white)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
 3.000 5.000 6.000 8.776 9.000 58.000 11

cat("\nSummary Statistics for Black Shooters:\n")

Summary Statistics for Black Shooters:

print(summary\_black)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
 3.000 4.000 5.000 5.571 6.000 12.000 11

cat("\nSummary Statistics for Latino Shooters:\n")

Summary Statistics for Latino Shooters:

print(summary\_latino)

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
 3.0 3.0 5.0 4.4 5.0 7.0 11

cat("\nT-Test Results (White vs. Black Shooters):\n")

T-Test Results (White vs. Black Shooters):

print(ttest\_white\_black)

Welch Two Sample t-test  
  
data: mass\_shootings$fatalities[mass\_shootings$race == "White"] and mass\_shootings$fatalities[mass\_shootings$race == "Black"]  
t = 2.7413, df = 85.048, p-value = 0.007457  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 0.8803752 5.5290064  
sample estimates:  
mean of x mean of y   
 8.776119 5.571429

cat("\nT-Test Results (White vs. Latino Shooters):\n")

T-Test Results (White vs. Latino Shooters):

print(ttest\_white\_latino)

Welch Two Sample t-test  
  
data: mass\_shootings$fatalities[mass\_shootings$race == "White"] and mass\_shootings$fatalities[mass\_shootings$race == "Latino"]  
t = 4.0458, df = 74.103, p-value = 0.0001266  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 2.220969 6.531270  
sample estimates:  
mean of x mean of y   
 8.776119 4.400000

cat("\nEffect Size (White vs. Black Shooters):\n")

Effect Size (White vs. Black Shooters):

print(effect\_size\_white\_black)

mean of x mean of y   
0.9516369 0.6041368

cat("\nEffect Size (White vs. Latino Shooters):\n")

Effect Size (White vs. Latino Shooters):

print(effect\_size\_white\_latino)

mean of x mean of y   
1.0194946 0.5111344

Answer <-"Answer: The distribution of mass shooting fatalities differs between White and Black shooters. The summary statistics show that White shooters have a slightly higher mean (8.776) compared to Black shooters (5.571), indicating a potential difference in the average number of fatalities. The Welch's t-test between White and Black shooters reveals a statistically significant difference in means (t = 2.7413, p = 0.007457), suggesting that the distributions are likely different. The effect size (Cohen's d) for this comparison is 0.9516, indicating a moderate effect.  
  
Regarding the distribution of mass shooting fatalities between White and Latino shooters, the summary statistics show that White shooters have a higher mean (8.776) compared to Latino shooters (4.4). The Welch's t-test between White and Latino shooters reveals a statistically significant difference in means (t = 4.0458, p = 0.0001266), suggesting distinct distributions. The effect size (Cohen's d) for this comparison is 1.0195, indicating a substantial effect.  
  
In summary, based on the available information, there are differences in the distribution of mass shooting fatalities between White and Black shooters as well as White and Latino shooters. White shooters tend to have higher average fatalities compared to both Black and Latino shooters, and statistical tests confirm these differences as statistically significant. Effect sizes indicate the practical significance of the observed differences, with moderate to substantial effects in both comparisons."  
  
# Print the answer  
cat("\n", Answer)

Answer: The distribution of mass shooting fatalities differs between White and Black shooters. The summary statistics show that White shooters have a slightly higher mean (8.776) compared to Black shooters (5.571), indicating a potential difference in the average number of fatalities. The Welch's t-test between White and Black shooters reveals a statistically significant difference in means (t = 2.7413, p = 0.007457), suggesting that the distributions are likely different. The effect size (Cohen's d) for this comparison is 0.9516, indicating a moderate effect.  
  
Regarding the distribution of mass shooting fatalities between White and Latino shooters, the summary statistics show that White shooters have a higher mean (8.776) compared to Latino shooters (4.4). The Welch's t-test between White and Latino shooters reveals a statistically significant difference in means (t = 4.0458, p = 0.0001266), suggesting distinct distributions. The effect size (Cohen's d) for this comparison is 1.0195, indicating a substantial effect.  
  
In summary, based on the available information, there are differences in the distribution of mass shooting fatalities between White and Black shooters as well as White and Latino shooters. White shooters tend to have higher average fatalities compared to both Black and Latino shooters, and statistical tests confirm these differences as statistically significant. Effect sizes indicate the practical significance of the observed differences, with moderate to substantial effects in both comparisons.

### 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?

# Subset data for mass shootings with mental illness and without signs of mental illness  
mental\_illness <- mass\_shootings$fatalities[mass\_shootings$prior\_mental\_illness == "Yes"]  
no\_mental\_illness <- mass\_shootings$fatalities[mass\_shootings$prior\_mental\_illness == "No"]  
  
# Perform t-test  
ttest\_mental\_illness <- t.test(mental\_illness, no\_mental\_illness)  
  
# Create a string representation of the t-test results  
ttest\_results <- paste("T-test results:\n",  
 "Statistic:", ttest\_mental\_illness$statistic, "\n",  
 "Degrees of Freedom:", ttest\_mental\_illness$parameter, "\n",  
 "P-value:", ttest\_mental\_illness$p.value, "\n")  
  
# Print the t-test results  
cat(ttest\_results)

T-test results:  
 Statistic: 0.775830936868875   
 Degrees of Freedom: 30.3424131726434   
 P-value: 0.443854255977963

# Check if there is a significant difference  
if (ttest\_mental\_illness$p.value < 0.05) {  
 cat("There is a significant difference in the number of fatalities between mass shootings with mental illness and those without signs of mental illness.\n")  
} else {  
 cat("There is no significant difference in the number of fatalities between mass shootings with mental illness and those without signs of mental illness.\n")  
}

There is no significant difference in the number of fatalities between mass shootings with mental illness and those without signs of mental illness.

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

# Relationship between mental illness and total victims  
ttest\_mental\_illness\_victims <- t.test(mass\_shootings$total\_victims[mass\_shootings$prior\_mental\_illness == "Yes"],  
 mass\_shootings$total\_victims[mass\_shootings$prior\_mental\_illness == "No"])  
  
# Relationship between mental illness and location type  
test\_mental\_illness\_location <- chisq.test(table(mass\_shootings$prior\_mental\_illness, mass\_shootings$location\_type))

Warning in chisq.test(table(mass\_shootings$prior\_mental\_illness,  
mass\_shootings$location\_type)): Chi-squared approximation may be incorrect

# Intersection of mental illness, location type, and total victims  
subset\_mental\_illness\_location <- mass\_shootings[mass\_shootings$prior\_mental\_illness == "Yes" & mass\_shootings$location\_type == "Specific Location", ]  
summary\_victims\_mental\_illness\_location <- summary(subset\_mental\_illness\_location$total\_victims)  
  
# Print the results  
cat("Relationship between mental illness and total victims:\n")

Relationship between mental illness and total victims:

print(ttest\_mental\_illness\_victims)

Welch Two Sample t-test  
  
data: mass\_shootings$total\_victims[mass\_shootings$prior\_mental\_illness == "Yes"] and mass\_shootings$total\_victims[mass\_shootings$prior\_mental\_illness == "No"]  
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

cat("\n")

cat("Relationship between mental illness and location type:\n")

Relationship between mental illness and location type:

cat("Contingency table:\n")

Contingency table:

print(table(mass\_shootings$prior\_mental\_illness, mass\_shootings$location\_type))

Airport Military Other Religious School Workplace  
 No 0 0 8 0 3 6  
 Yes 1 2 24 4 10 21

cat("\n")

cat("Chi-square test results:\n")

Chi-square test results:

print(test\_mental\_illness\_location)

Pearson's Chi-squared test  
  
data: table(mass\_shootings$prior\_mental\_illness, mass\_shootings$location\_type)  
X-squared = 2.1755, df = 5, p-value = 0.8244

cat("\n")

cat("Intersection of mental illness, location type, and total victims:\n")

Intersection of mental illness, location type, and total victims:

cat("Summary statistics of total victims:\n")

Summary statistics of total victims:

print(summary\_victims\_mental\_illness\_location)

Min. 1st Qu. Median Mean 3rd Qu. Max.

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.

# Group the data by year and calculate the count and frequency of fraudulent transactions  
fraud\_summary <- card\_fraud %>%  
 group\_by(trans\_year) %>%  
 summarise(fraud\_count = sum(is\_fraud),  
 fraud\_frequency = mean(is\_fraud))  
  
# Print the fraud summary table  
print(fraud\_summary)

# A tibble: 2 × 3  
 trans\_year fraud\_count fraud\_frequency  
 <dbl> <dbl> <dbl>  
1 2019 2721 0.00568  
2 2020 1215 0.00632

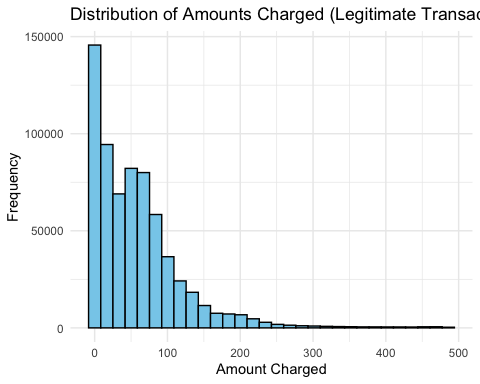
* How much money (in US$ terms) are fraudulent transactions costing the company? Generate a table that summarizes the total amount of legitimate and fraudulent transactions per year and calculate the % of fraudulent transactions, in US$ terms.

# Group the data by year and calculate the total amount of legitimate and fraudulent transactions  
transaction\_summary <- card\_fraud %>%  
 group\_by(trans\_year) %>%  
 summarise(total\_legitimate\_amount = sum(amt[is\_fraud == 0]),  
 total\_fraudulent\_amount = sum(amt[is\_fraud == 1]))  
  
# Calculate the percentage of fraudulent transactions in US dollar terms  
transaction\_summary <- transaction\_summary %>%  
 mutate(fraud\_percentage = (total\_fraudulent\_amount / (total\_legitimate\_amount + total\_fraudulent\_amount)) \* 100)  
  
# Print the transaction summary table  
print(transaction\_summary)

# A tibble: 2 × 4  
 trans\_year total\_legitimate\_amount total\_fraudulent\_amount fraud\_percentage  
 <dbl> <dbl> <dbl> <dbl>  
1 2019 32182901. 1423140. 4.23  
2 2020 12925914. 651949. 4.80

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

library(dplyr)  
library(ggplot2)  
  
# Create separate data frames for legitimate and fraudulent transactions  
legitimate\_transactions <- card\_fraud %>% filter(is\_fraud == 0)  
fraudulent\_transactions <- card\_fraud %>% filter(is\_fraud == 1)  
  
  
# Remove outliers from legitimate transactions  
legitimate\_transactions <- legitimate\_transactions %>%  
 filter(amt <= quantile(amt, 0.99)) # Adjust the quantile value as desired  
  
# Remove outliers from fraudulent transactions  
fraudulent\_transactions <- fraudulent\_transactions %>%  
 filter(amt <= quantile(amt, 0.99)) # Adjust the quantile value as desired  
  
# Generate histogram for legitimate transactions with "smooth (calm)" color  
ggplot(legitimate\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "#87CEEB", color = "black", bins = 30) +  
 labs(title = "Distribution of Amounts Charged (Legitimate Transactions)",  
 x = "Amount Charged", y = "Frequency") +  
 theme\_minimal()



# Calculate summary statistics for legitimate transactions  
legitimate\_summary <- summary(legitimate\_transactions$amt)  
cat("Summary Statistics for Legitimate Transactions:\n")

Summary Statistics for Legitimate Transactions:

cat("Min. :", legitimate\_summary[1], "\n")

Min. : 1

cat("1st Qu. :", legitimate\_summary[2], "\n")

1st Qu. : 9.5

cat("Median :", legitimate\_summary[3], "\n")

Median : 46.48

cat("Mean :", legitimate\_summary[4], "\n")

Mean : 58.67505

cat("3rd Qu. :", legitimate\_summary[5], "\n")

3rd Qu. : 80.99

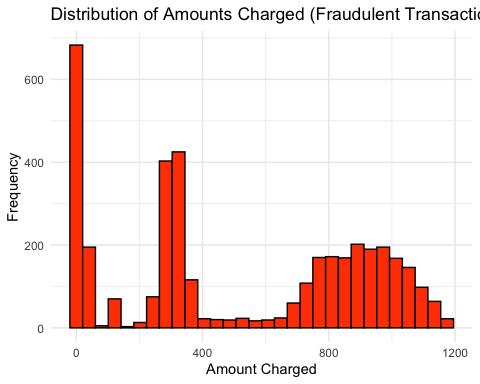
cat("Max. :", legitimate\_summary[6], "\n")

Max. : 486.8

cat("NA's :", legitimate\_summary[7], "\n\n")

NA's : NA

# Generate histogram for fraudulent transactions with "alarm" color  
ggplot(fraudulent\_transactions, aes(x = amt)) +  
 geom\_histogram(fill = "#FF4500", color = "black", bins = 30) +  
 labs(title = "Distribution of Amounts Charged (Fraudulent Transactions)",  
 x = "Amount Charged", y = "Frequency") +  
 theme\_minimal()



# Calculate summary statistics for fraudulent transactions  
fraudulent\_summary <- summary(fraudulent\_transactions$amt)  
cat("Summary Statistics for Fraudulent Transactions:\n")

Summary Statistics for Fraudulent Transactions:

cat("Min. :", fraudulent\_summary[1], "\n")

Min. : 1.06

cat("1st Qu. :", fraudulent\_summary[2], "\n")

1st Qu. : 230.4625

cat("Median :", fraudulent\_summary[3], "\n")

Median : 361.415

cat("Mean :", fraudulent\_summary[4], "\n")

Mean : 520.0701

cat("3rd Qu. :", fraudulent\_summary[5], "\n")

3rd Qu. : 895.0525

cat("Max. :", fraudulent\_summary[6], "\n")

Max. : 1175.88

cat("NA's :", fraudulent\_summary[7], "\n")

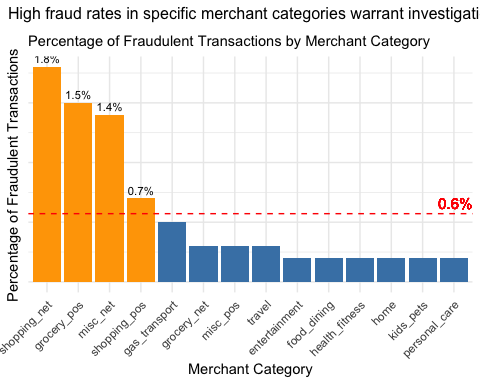
NA's : NA

Answer <- "The summary statistics reveal that fraudulent transactions tend to involve larger amounts compared to legitimate transactions. This poses a significant problem as higher transaction amounts in fraudulent activities can lead to substantial financial losses for the company. Detecting and preventing fraud is crucial to mitigate these risks, safeguard the company's financial health, and maintain customer trust."  
cat(Answer)

The summary statistics reveal that fraudulent transactions tend to involve larger amounts compared to legitimate transactions. This poses a significant problem as higher transaction amounts in fraudulent activities can lead to substantial financial losses for the company. Detecting and preventing fraud is crucial to mitigate these risks, safeguard the company's financial health, and maintain customer trust.

* 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 count and percentage of fraudulent transactions per category  
fraud\_percentage\_df <- card\_fraud %>%  
 group\_by(category) %>%  
 summarise(total\_fraudulent\_transactions = sum(is\_fraud),  
 total\_transactions = n(),  
 percentage = round((total\_fraudulent\_transactions / total\_transactions) \* 100, 1))  
  
# Calculate the average percentage of fraudulent transactions  
avg\_percentage <- mean(fraud\_percentage\_df$percentage)  
  
# Add a column indicating if the category is above or below average  
fraud\_percentage\_df <- fraud\_percentage\_df %>%  
 mutate(above\_average = ifelse(percentage > avg\_percentage, "Above Average", "Below Average"))  
  
# Create the bar chart with percentages, average line, and average label  
ggplot(fraud\_percentage\_df, aes(x = reorder(category, -percentage), y = percentage, fill = above\_average)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = ifelse(percentage > avg\_percentage, paste0(round(percentage, 1), "%"), "")), vjust = -0.5, size = 3) +  
 geom\_hline(yintercept = avg\_percentage, linetype = "dashed", color = "red") +  
 geom\_text(aes(label = paste0(round(avg\_percentage, 1), "%")), x = Inf, y = avg\_percentage, vjust = -0.5, color = "red", hjust = 1, size = 4) +  
 labs(x = "Merchant Category", y = "Percentage of Fraudulent Transactions") +  
 ggtitle(" High fraud rates in specific merchant categories warrant investigation") +  
 labs(subtitle = "Percentage of Fraudulent Transactions by Merchant Category") +  
 scale\_fill\_manual(values = c("Below Average" = "steelblue", "Above Average" = "orange")) +  
 theme\_minimal() +  
 theme(legend.position = "none",  
 axis.text.x = element\_text(angle = 45, hjust = 1),  
 axis.text.y = element\_blank(),  
 axis.ticks.y = element\_blank(),  
 axis.text.y.right = element\_text(color = "red", size = 6),  
 plot.title = element\_text(size = 12, hjust = 0.5, margin = margin(b = 10)))



Answer <- "After analyzing the data, we found that there are four categories of merchants with percentages of fraudulent transactions that are above the average. However, upon closer inspection, three of these categories stand out even more. The category Shopping\_Net has an exact value of three times the average percentage of fraudulent transactions, while Grocery\_POS and Misc\_Net are both more than two times the average. These three categories in particular warrant further investigation as they exhibit significantly higher rates of fraudulent transactions compared to the average"  
cat(Answer)

After analyzing the data, we found that there are four categories of merchants with percentages of fraudulent transactions that are above the average. However, upon closer inspection, three of these categories stand out even more. The category Shopping\_Net has an exact value of three times the average percentage of fraudulent transactions, while Grocery\_POS and Misc\_Net are both more than two times the average. These three categories in particular warrant further investigation as they exhibit significantly higher rates of fraudulent transactions compared to the average

* 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),  
 )

library(dplyr)  
library(lubridate)  
  
# Create new variables for analysis  
data <- card\_fraud %>%  
 mutate(  
 day\_of\_year = yday(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)  
 )  
  
# Analyze fraud prevalence by days  
fraud\_by\_day <- data %>%  
 group\_by(day\_of\_year) %>%  
 summarise(fraud\_count = sum(is\_fraud == 1),  
 total\_count = n()) %>%  
 mutate(fraud\_percentage = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(fraud\_percentage))  
  
# Analyze fraud prevalence by months  
fraud\_by\_month <- data %>%  
 group\_by(month\_name) %>%  
 summarise(fraud\_count = sum(is\_fraud == 1),  
 total\_count = n()) %>%  
 mutate(fraud\_percentage = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(fraud\_percentage))  
  
# Analyze fraud prevalence by hours  
fraud\_by\_hour <- data %>%  
 group\_by(hour) %>%  
 summarise(fraud\_count = sum(is\_fraud == 1),  
 total\_count = n()) %>%  
 mutate(fraud\_percentage = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(fraud\_percentage))  
  
# Print the results  
cat("Fraud prevalence by days (ordered by descending fraud percentage):\n")

Fraud prevalence by days (ordered by descending fraud percentage):

print(fraud\_by\_day)

# A tibble: 365 × 4  
 day\_of\_year fraud\_count total\_count fraud\_percentage  
 <dbl> <int> <int> <dbl>  
 1 254 24 838 2.86  
 2 31 29 1372 2.11  
 3 56 34 1790 1.90  
 4 324 15 817 1.84  
 5 30 23 1263 1.82  
 6 305 16 992 1.61  
 7 253 12 750 1.6   
 8 169 30 1910 1.57  
 9 306 22 1419 1.55  
10 79 27 1747 1.55  
# ℹ 355 more rows

cat("\nFraud prevalence by months (ordered by descending fraud percentage):\n")

Fraud prevalence by months (ordered by descending fraud percentage):

print(fraud\_by\_month)

# A tibble: 12 × 4  
 month\_name fraud\_count total\_count fraud\_percentage  
 <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

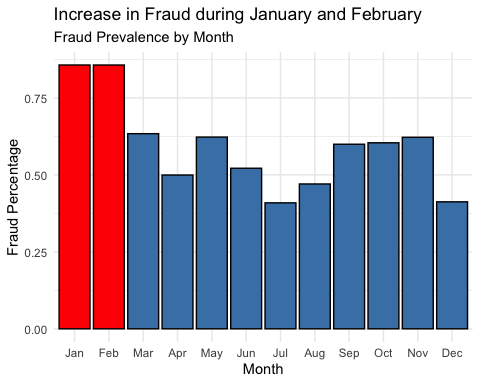
cat("\nFraud prevalence by hours (ordered by descending fraud percentage):\n")

Fraud prevalence by hours (ordered by descending fraud percentage):

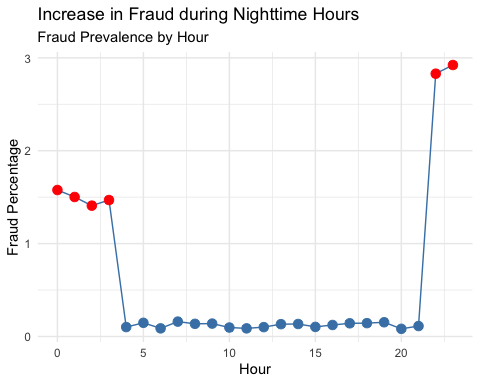
print(fraud\_by\_hour)

# A tibble: 24 × 4  
 hour fraud\_count total\_count fraud\_percentage  
 <int> <int> <int> <dbl>  
 1 23 1012 34625 2.92   
 2 22 981 34674 2.83   
 3 0 348 22070 1.58   
 4 1 332 22107 1.50   
 5 3 326 22192 1.47   
 6 2 313 22222 1.41   
 7 7 35 21855 0.160  
 8 19 52 33987 0.153  
 9 5 32 21752 0.147  
10 18 49 34131 0.144  
# ℹ 14 more rows

# Filter data for months of January and February  
fraud\_jan\_feb <- data %>%  
 filter(month\_name %in% c("Jan", "Feb"))  
  
  
# Convert hour to numeric  
fraud\_by\_hour$hour <- as.numeric(fraud\_by\_hour$hour)  
  
# Create plot for fraud prevalence by hour  
plot\_hour <- ggplot(fraud\_by\_hour, aes(x = hour, y = fraud\_percentage)) +  
 geom\_line(color = "steelblue") +  
 geom\_point(aes(color = ifelse(hour >= 22 | hour <= 3, "Nighttime", "Daytime")),  
 size = 3) +  
 scale\_color\_manual(values = c("Daytime" = "steelblue", "Nighttime" = "red")) +  
 labs(x = "Hour", y = "Fraud Percentage") +  
 ggtitle("Increase in Fraud during Nighttime Hours") +  
 theme\_minimal() +  
 labs(subtitle = "Fraud Prevalence by Hour") +  
 theme(legend.position = "none")  
  
# Create plot for fraud prevalence by month  
plot\_month <- ggplot(fraud\_by\_month, aes(x = month\_name, y = fraud\_percentage, fill = month\_name)) +  
 geom\_bar(stat = "identity", color = "black") +  
 labs(x = "Month", y = "Fraud Percentage") +  
 ggtitle("Increase in Fraud during January and February") +  
 theme\_minimal() +  
 scale\_fill\_manual(values = ifelse(fraud\_by\_month$month\_name %in% c("Jan", "Feb"), "red", "steelblue")) +  
 labs(subtitle = "Fraud Prevalence by Month") +  
 theme(legend.position = "none")  
  
# Display the plots  
plot\_month

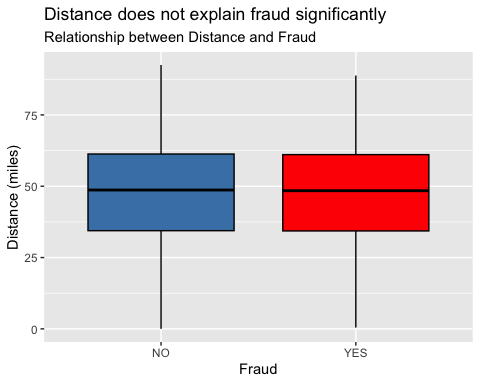


plot\_hour



* 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))  
  
 )  
  
# Convert is\_fraud to factor  
fraud$is\_fraud <- factor(fraud$is\_fraud, labels = c("NO", "YES"))  
  
# Boxplot  
ggplot(fraud, aes(x = is\_fraud, y = distance\_miles, fill = is\_fraud)) +  
 geom\_boxplot(color = "black") +  
 scale\_fill\_manual(values = c("steelblue", "red")) +  
 labs(x = "Fraud", y = "Distance (miles)") +  
 ggtitle("Distance does not explain fraud significantly") +  
 labs(subtitle = "Relationship between Distance and Fraud") +  
 theme(legend.position = "none")



# Violin plot  
ggplot(fraud, aes(x = factor(is\_fraud), y = distance\_miles, fill = factor(is\_fraud))) +  
 geom\_violin(color = "black") +  
 scale\_fill\_manual(values = c("steelblue", "red"), labels = c("Non-Fraud", "Fraud")) +  
 labs(x = "Fraud", y = "Distance (miles)") +  
 ggtitle("Distance does not explain fraud significantly") +  
 labs(subtitle = "Relationship between Distance and Fraud") +  
 theme(legend.position = "none")



Answer <- "Based on the boxplot and violin plot, there doesn't seem to be a clear relationship between distance and fraud. The distribution of distances for both fraudulent and non-fraudulent transactions appears to be similar, with no significant difference in the median or spread. Therefore, it can be concluded that distance alone may not be a useful feature in explaining fraud in this dataset."  
cat(Answer)

Based on the boxplot and violin plot, there doesn't seem to be a clear relationship between distance and fraud. The distribution of distances for both fraudulent and non-fraudulent transactions appears to be similar, with no significant difference in the median or spread. Therefore, it can be concluded that distance alone may not be a useful feature in explaining fraud in this dataset.

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

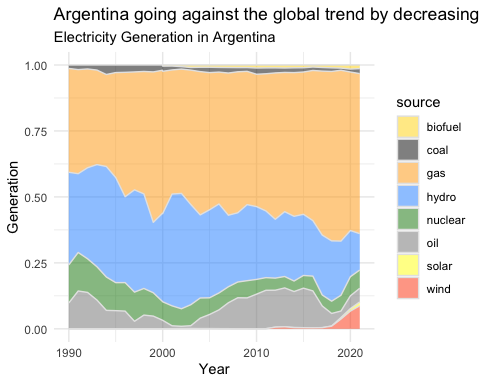
geom\_area(colour="grey90", alpha = 0.5, position = "fill")

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

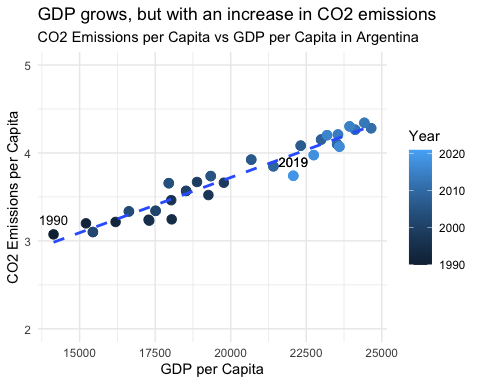
## 3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related

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.

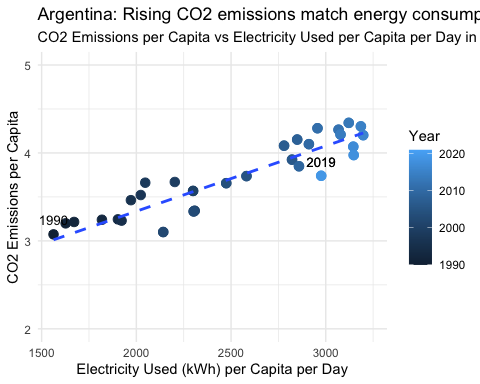
#install.packages("cowplot")  
library(cowplot)  
  
# Download electricity data  
url <- "https://nyc3.digitaloceanspaces.com/owid-public/data/energy/owid-energy-data.csv"  
  
energy <- read\_csv(url) %>%   
 filter(year >= 1990) %>%   
 drop\_na(iso\_code) %>%   
 select(1:3,  
 biofuel = biofuel\_electricity,  
 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)  
  
# Make the table tidy  
energy\_tidy <- energy %>%  
 pivot\_longer(cols = c(biofuel, coal, gas, hydro, nuclear, oil, other\_renewable, solar, wind),  
 names\_to = c(".value", "source"),  
 names\_sep = "\_",  
 values\_drop\_na = TRUE) %>%  
 select(-iso\_code)  
  
# Join the data frames  
merged\_data <- left\_join(energy\_tidy, co2\_percap, by = c("country", "year")) %>%  
 left\_join(., gdp\_percap, by = c("country", "year"))  
  
# Filter data for Argentina  
argentina\_data <- merged\_data %>% filter(country == "Argentina")  
  
# Rename the duplicate column names  
names(argentina\_data)[9] <- "renewable\_2"  
names(argentina\_data)[21] <- "wind\_2"  
  
# Define a custom color palette for electricity generation  
generation\_colors <- c(biofuel = "#FFD700",  
 coal = "#000000",  
 gas = "#FFA500",  
 hydro = "#1E90FF",  
 nuclear = "#008000",  
 oil = "#808080",  
 solar = "#FFFF00",  
 wind = "#FF4500")  
  
# Create stacked area chart for electricity generation  
electricity\_generation\_chart <- argentina\_data %>%  
 pivot\_longer(cols = c(biofuel, coal, gas, hydro, nuclear, oil, solar, wind),  
 names\_to = "source",  
 values\_to = "generation",  
 names\_repair = "unique") %>%  
 ggplot(aes(x = year, y = generation, fill = source)) +  
 geom\_area(colour = "grey90", alpha = 0.5, position = "fill") +  
 scale\_fill\_manual(values = generation\_colors) +  
 labs(title = "Argentina going against the global trend by decreasing renewable energy share",  
 x = "Year", y = "Generation") +  
 labs(subtitle = "Electricity Generation in Argentina") +  
 theme\_minimal()  
  
# Display the electricity generation chart  
electricity\_generation\_chart



# Get the first and last years  
first\_year <- min(argentina\_data$year)  
last\_year <- max(argentina\_data$year)  
second\_last\_year <- max(argentina\_data$year) - 2   
  
# Create scatter plot of CO2 per capita vs GDP per capita for Argentina with trend line and annotations  
co2\_gdp\_scatter <- ggplot(argentina\_data, aes(x = GDPpercap, y = co2percap)) +  
 geom\_point(aes(color = year), size = 3) +  
 geom\_smooth(method = "lm", se = FALSE, linetype = "dashed") + # Add trend line  
 geom\_text(data = argentina\_data %>% filter(year %in% c(first\_year, second\_last\_year, last\_year)),  
 aes(label = as.character(year)), vjust = -1, size = 3.5) + # Add annotations for selected years  
 labs(title = "GDP grows, but with an increase in CO2 emissions",  
 x = "GDP per Capita",  
 y = "CO2 Emissions per Capita",  
 color = "Year") +  
 labs(subtitle = "CO2 Emissions per Capita vs GDP per Capita in Argentina") +  
 theme\_minimal() +  
 scale\_x\_continuous(expand = c(0.05, 0)) + # Increase x-axis expansion  
 scale\_y\_continuous(expand = c(0.05, 0), limits = c(2, 5)) # Set limits of y-axis  
  
# Display the scatter plot  
co2\_gdp\_scatter

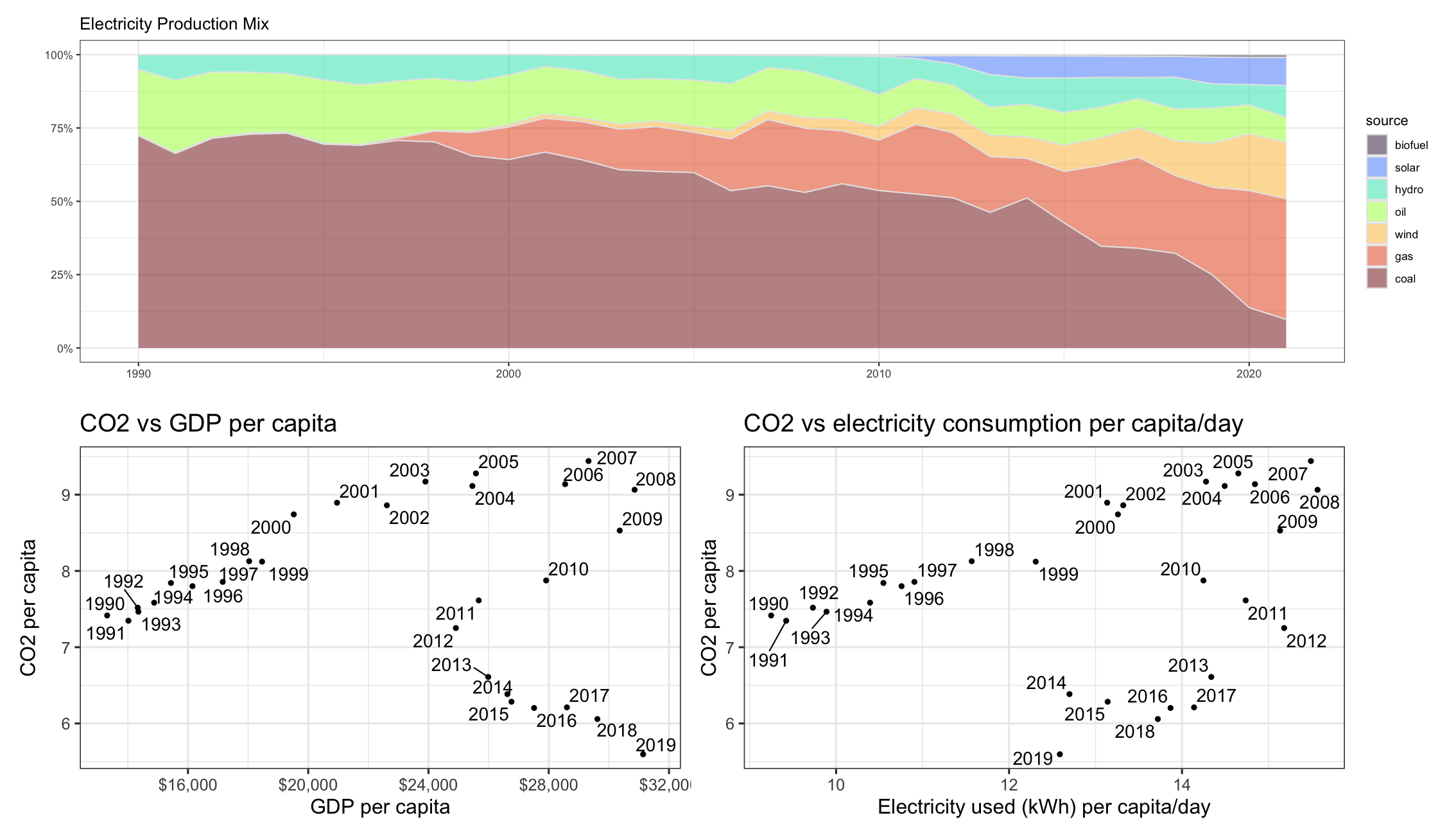


# Create scatter plot of CO2 per capita vs Electricity used per capita per day for Argentina  
co2\_electricity\_scatter <- ggplot(argentina\_data, aes(x = per\_capita\_electricity, y = co2percap)) +  
 geom\_point(aes(color = year), size = 3) +  
 geom\_smooth(method = "lm", se = FALSE, linetype = "dashed") + # Add trend line  
 geom\_text(data = argentina\_data %>% filter(year %in% c(first\_year, second\_last\_year, last\_year)),  
 aes(label = as.character(year)), vjust = -1, size = 3.5) + # Add annotations for selected years  
 labs(title = "Argentina: Rising CO2 emissions match energy consumption",  
 x = "Electricity Used (kWh) per Capita per Day",  
 y = "CO2 Emissions per Capita",  
 color = "Year") +  
 labs(subtitle = "CO2 Emissions per Capita vs Electricity Used per Capita per Day in Argentina") +  
 theme\_minimal() +  
 scale\_x\_continuous(expand = c(0.05, 0)) + # Increase x-axis expansion  
 scale\_y\_continuous(expand = c(0.05, 0), limits = c(2, 5)) # Set limits of y-axis  
  
# Display the scatter plot  
co2\_electricity\_scatter



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: -
* Approximately how much time did you spend on this problem set: 15 HOURS
* What, if anything, gave you the most trouble: THE LAST ONE

**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.