Homerwork 2

GUIDO BOZZANO

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.

#Create a dataframe based on mass\_shootings that groups by year and counts the number of shootings per year  
  
dfmass\_shootings <- mass\_shootings %>%   
 group\_by(year) %>%   
 summarise(count=n())  
  
dfmass\_shootings

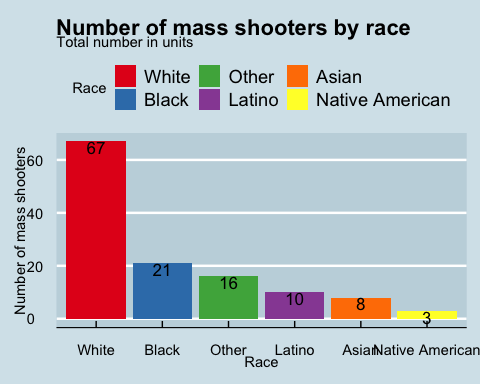
# A tibble: 37 × 2  
 year count  
 <dbl> <int>  
 1 1982 1  
 2 1984 2  
 3 1986 1  
 4 1987 1  
 5 1988 1  
 6 1989 2  
 7 1990 1  
 8 1991 3  
 9 1992 2  
10 1993 4  
# ℹ 27 more rows

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

#Create a dataframe that groups by race and I renamed the duplicated race categories so that the information is collapsed into one variable per race category. Then I collapse all the others into Other. Afterwards, I reorder the information using fct\_reorder and get it ready to plot it  
  
dfmass\_shootings <- mass\_shootings %>%   
 group\_by(race) %>%   
 mutate(race = case\_when(  
 race %in% c("White", "white") ~ "White",   
 race %in% c("Black", "black") ~ "Black",   
 race %in% c("Asian", "Latino", "Native American", "") ~ race,  
 TRUE ~ "Other" # Collapse other races into "Other" category  
 )) %>%   
 summarise(count=n()) %>%   
 mutate(race = fct\_reorder(race, count, .desc = TRUE))  
  
dfmass\_shootings

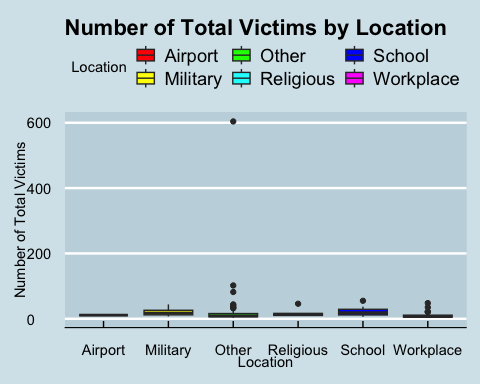
# A tibble: 6 × 2  
 race count  
 <fct> <int>  
1 Asian 8  
2 Black 21  
3 Latino 10  
4 Native American 3  
5 Other 16  
6 White 67

# Get unique race categories  
unique\_races <- unique(dfmass\_shootings$race) #define a unique list of categories  
  
# Generate a color palette for each race category  
num\_races <- length(unique\_races)  
custom\_colors <- brewer.pal(num\_races, "Set1")  
  
# Generate the bar chart  
chart <- ggplot(dfmass\_shootings, aes(x = race, y = count, fill = race)) +  
 geom\_bar(stat = "identity") +  
 theme\_economist(base\_family = "ITC Officina Sans", dkpanel = TRUE) +  
 scale\_fill\_manual(values = custom\_colors) +  
 geom\_text(aes(label = count), vjust = 1, colour = "black", size = 4.5) +  
 labs(  
 title = "Number of mass shooters by race",  
 subtitle = "Total number in units",  
 x = "Race",  
 y = "Number of mass shooters",  
 fill = "Race"  
 )  
  
chart



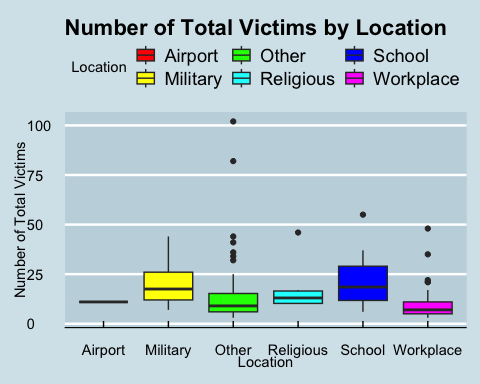
* Generate a boxplot visualizing the number of total victims, by type of location.

# Get unique race categories  
unique\_location <- unique(mass\_shootings$location\_type)  
  
# Generate a color palette for each race category  
num\_location <- length(unique\_location)  
custom\_colors <- rainbow(num\_location)  
  
# Generate the bar chart  
chart <- ggplot(mass\_shootings, aes(x = location\_type, y = total\_victims, fill = location\_type)) +  
 geom\_boxplot() +  
 theme\_economist(base\_family = "ITC Officina Sans", dkpanel = TRUE) +  
 scale\_colour\_economist() +   
 scale\_fill\_manual(values = custom\_colors) +  
 labs(  
 title = "Number of Total Victims by Location",  
 x = "Location",  
 y = "Number of Total Victims",  
 fill = "Location"  
 )  
  
chart



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

#I create dfmass\_shootings and exclude the Las Vegas Strip Massacre and re plot the data using the same code as the question above  
  
dfmass\_shootings <- mass\_shootings %>%   
 filter(case !="Las Vegas Strip massacre")  
  
# Get unique race categories  
unique\_location <- unique(dfmass\_shootings$location\_type)  
  
# Generate a color palette for each race category  
num\_location <- length(unique\_location)  
custom\_colors <- rainbow(num\_location)  
  
# Generate the bar chart  
chart <- ggplot(dfmass\_shootings, aes(x = location\_type, y = total\_victims, fill = location\_type)) +  
 geom\_boxplot() +  
 theme\_economist(base\_family = "ITC Officina Sans", dkpanel = TRUE) +  
 scale\_colour\_economist() +   
 scale\_fill\_manual(values = custom\_colors) +  
 labs(  
 title = "Number of Total Victims by Location",  
 x = "Location",  
 y = "Number of Total Victims",  
 fill = "Location"  
 )  
  
chart



### 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?

dfmass\_shootings1 <- mass\_shootings %>%   
 filter(race %in% c("White", "white")) %>% #Filter race white and White  
 filter(prior\_mental\_illness == "Yes") %>% #Filter prior mental illness as Yes  
 filter(year > 2000) %>% #Filter year greater than 2000 (not inclusive of the year 2000)  
 filter(male == "TRUE") %>% #Filter male = TRUE to include only males  
 summarise(count=n()) #Count the information  
  
dfmass\_shootings1

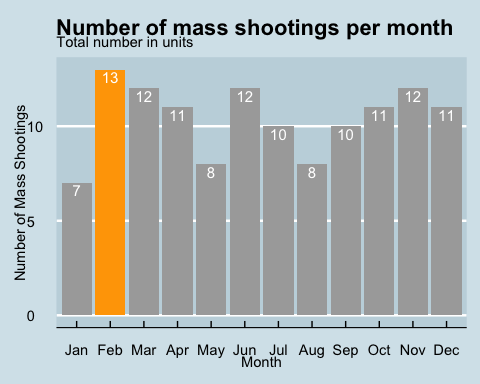
# A tibble: 1 × 1  
 count  
 <int>  
1 22

#Answer: 22 white males

* 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.

# Calculate the count of mass shootings per month  
month\_counts <- mass\_shootings %>%  
 count(month)  
  
# Specify the desired order of month names  
month\_order <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")  
  
# Reorder the levels of month variable  
month\_counts <- month\_counts %>%  
 mutate(month = factor(month, levels = month\_order, ordered = TRUE))  
  
# Generate the bar chart  
ggplot(month\_counts, aes(x = month, y = n, fill = month == "Feb")) +  
 geom\_bar(stat = "identity") +  
 theme\_economist(base\_family = "ITC Officina Sans", dkpanel = TRUE) +  
 scale\_fill\_manual(values = c("darkgrey", "orange"), guide = FALSE) +  
 scale\_colour\_economist() +  
 geom\_text(aes(label = n), vjust = 1.2, colour = "white", size = 4) +  
 labs(  
 title = "Number of mass shootings per month",  
 subtitle = "Total number in units",  
 x = "Month",  
 y = "Number of Mass Shootings"  
 )

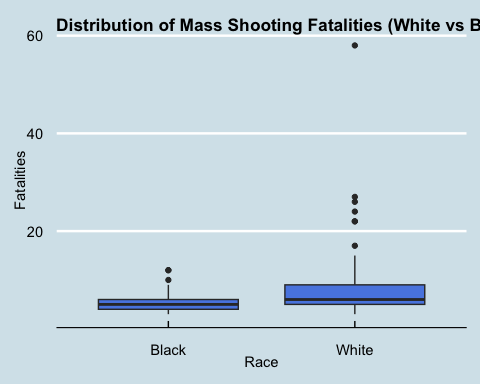
Warning: The `guide` argument in `scale\_\*()` cannot be `FALSE`. This was deprecated in  
ggplot2 3.3.4.  
ℹ Please use "none" instead.



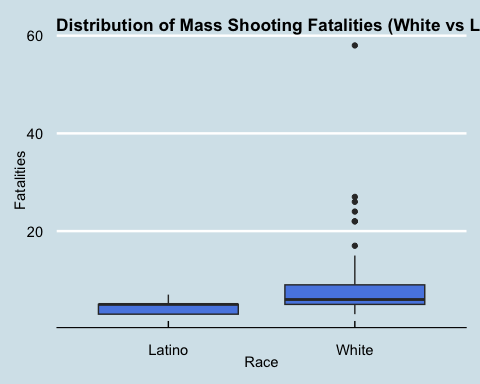
#Answer: Based on the chart, the month with greater number of mass shootings in the year is February

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

# Filter the data for White and Black shooters  
white\_black\_data <- mass\_shootings %>%  
 mutate(race = case\_when(  
 race %in% c("White", "white") ~ "White",   
 race %in% c("Black", "black") ~ "Black",   
 race %in% c("Asian", "Latino", "Native American", "") ~ race,  
 TRUE ~ "Other" # Collapse other races into "Other" category  
 )) %>%   
 filter(race %in% c("White", "Black"))  
  
# Filter the data for White and Latino shooters  
white\_latino\_data <- mass\_shootings %>%  
 mutate(race = case\_when(  
 race %in% c("White", "white") ~ "White",   
 race %in% c("Black", "black") ~ "Black",   
 race %in% c("Asian", "Latino", "Native American", "") ~ race,  
 TRUE ~ "Other" # Collapse other races into "Other" category  
 )) %>%   
 filter(race %in% c("White", "Latino"))  
  
# Create boxplots for White and Black shooters  
boxplot\_white\_black <- ggplot(white\_black\_data, aes(x = race, y = fatalities)) +  
 geom\_boxplot(fill = "#5989e3") +  
 theme\_economist() +  
 labs(  
 title = "Distribution of Mass Shooting Fatalities (White vs Black Shooters)",  
 x = "Race",  
 y = "Fatalities"  
 ) +  
 theme(plot.title = element\_text(size = 13))  
  
# Create boxplots for White and Latino shooters  
boxplot\_white\_latino <- ggplot(white\_latino\_data, aes(x = race, y = fatalities)) +  
 geom\_boxplot(fill = "#5989e3") +  
 theme\_economist() +  
 labs(  
 title = "Distribution of Mass Shooting Fatalities (White vs Latino Shooters)",  
 x = "Race",  
 y = "Fatalities"  
 ) +  
 theme(plot.title = element\_text(size = 13))  
  
# Display the boxplots  
boxplot\_white\_black



boxplot\_white\_latino



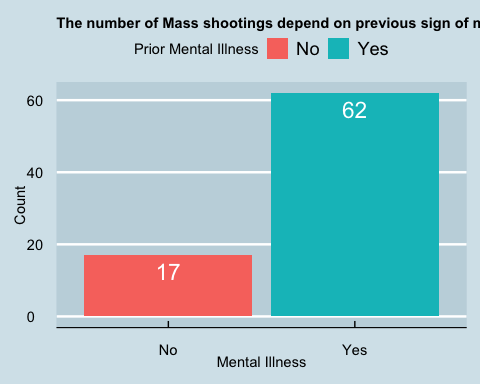
### Very open-ended

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

#To answer this question I propose comparing the number of mass shootings depending on whether there is a prior signal of mental illness in the shooter  
dfmass\_shootings <- mass\_shootings %>%   
 filter(!is.na(prior\_mental\_illness))  
  
label\_data <- dfmass\_shootings %>%  
 count(prior\_mental\_illness)  
  
dfmass\_shootings

# A tibble: 79 × 14  
 case year month day location summary fatalities injured total\_victims  
 <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
 1 San Jose… 2021 May 26 San Jos… "Samue… 9 0 9  
 2 FedEx wa… 2021 Apr 15 Indiana… "Brand… 8 7 15  
 3 Boulder … 2021 Mar 22 Boulder… "Ahmad… 10 0 10  
 4 Odessa-M… 2019 Aug 31 Odessa,… "Seth … 7 25 32  
 5 Harry Pr… 2019 Feb 15 Aurora,… "Gary … 5 6 11  
 6 SunTrust… 2019 Jan 23 Sebring… "Zephe… 5 0 5  
 7 Thousand… 2018 Nov 7 Thousan… "Ian D… 12 22 34  
 8 Fifth Th… 2018 Sep 6 Cincinn… "Omar … 3 2 5  
 9 Waffle H… 2018 Apr 22 Nashvil… "Travi… 4 4 8  
10 Yountvil… 2018 Mar 9 Yountvi… "Army … 3 0 3  
# ℹ 69 more rows  
# ℹ 5 more variables: location\_type <chr>, male <lgl>, age\_of\_shooter <dbl>,  
# race <chr>, prior\_mental\_illness <chr>

# Create bar chart comparing variables for shootings with mental illness vs. no mental illness  
ggplot(dfmass\_shootings, aes(x = prior\_mental\_illness, fill = prior\_mental\_illness)) +  
 geom\_bar() +  
 theme\_economist(base\_family = "ITC Officina Sans", dkpanel = TRUE) +  
 geom\_text(data = label\_data, aes(label = n, y = n), vjust = 1.5, colour = "white", size = 6) +  
 labs(  
 title = "The number of Mass shootings depend on previous sign of mental illness in the shooter ",  
 x = "Mental Illness",  
 y = "Count",   
 fill = "Prior Mental Illness"  
 ) +  
 theme(  
 plot.title = element\_text(size = 11)  
 )



#Answer: the number of mass shootings seems to depend on whether there's a sign of mentall illness in the shooter

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

dfmass\_shootingsment <- mass\_shootings %>%   
 filter(location\_type != "Other") %>% #Exclude location type "Other" from the analysis for clarity  
 filter(!is.na(prior\_mental\_illness)) %>% #Filter for mental ilness  
 group\_by(location\_type) %>%   
 summarise(SumVic=sum(total\_victims)) #Sum the number of total victims  
  
dfmass\_shootingsment

# A tibble: 5 × 2  
 location\_type SumVic  
 <chr> <dbl>  
1 Airport 11  
2 Military 48  
3 Religious 82  
4 School 329  
5 Workplace 243

dfmass\_shootingsnoment <- mass\_shootings %>%   
 filter(location\_type != "Other") %>% #Exclude location type "Other" from the analysis for clarity  
 filter(is.na(prior\_mental\_illness)) %>% #Filter for no mental illness  
 group\_by(location\_type) %>%   
 summarise(SumVic=sum(total\_victims)) #Sum the number of total victims  
  
dfmass\_shootingsnoment

# A tibble: 4 × 2  
 location\_type SumVic  
 <chr> <dbl>  
1 Military 77  
2 Religious 27  
3 School 65  
4 Workplace 212

#To begin with, there seems to be more shootings if the shooter shows signs of previous mental illness. On top of that, the shootings seem to be more heavily focused in Schools and Workplace compared to only Workplace in cases where shooter displays no sign of mental illness

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.

fraud\_transactions <- card\_fraud %>%  
 group\_by(trans\_year) %>%  
 summarise(total\_transactions = n(),  
 fraudulent\_transactions = sum(is\_fraud),  
 fraud\_percentage = (fraudulent\_transactions / total\_transactions)\*100)  
  
fraud\_transactions

# A tibble: 2 × 4  
 trans\_year total\_transactions fraudulent\_transactions fraud\_percentage  
 <dbl> <int> <dbl> <dbl>  
1 2019 478646 2721 0.568  
2 2020 192382 1215 0.632

#Answer: Fraud transactions are very unlikely in this dataset. The reason behind this is that the fraud percentage is particularly low for the years 2019 and 2020 where it's around 0.56% and 0.63% respectively

* 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.

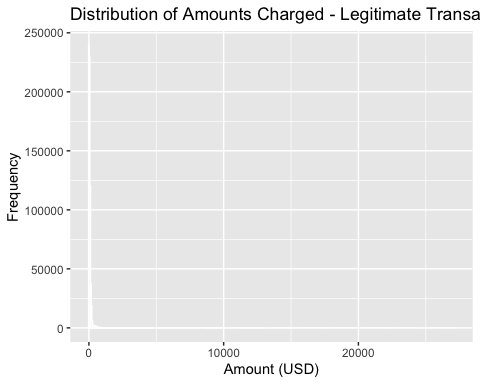
fraud\_cost\_summary <- card\_fraud %>% #Generate a fraud cost summary table  
 group\_by(trans\_year) %>% #Group by transaction year  
 summarise(total\_legitimate\_amount = sum(amt[!is\_fraud]), #Summarise the amount that is legitimate by filtering the transactions that aren't fraud  
 total\_fraudulent\_amount = sum(amt[is\_fraud]), #Summarise the amount that is legitimate by filtering the transactions that are fraud  
 total\_amount = sum(amt), #Sum the total amout  
 fraud\_percentage = (total\_fraudulent\_amount / total\_amount) \* 100) #Calculate the percentage of fraud per year  
  
fraud\_cost\_summary

# A tibble: 2 × 5  
 trans\_year total\_legitimate\_amount total\_fraudulent\_amount total\_amount  
 <dbl> <dbl> <dbl> <dbl>  
1 2019 32182901. 21197. 33606041.  
2 2020 12925914. 4143. 13577863.  
# ℹ 1 more variable: fraud\_percentage <dbl>

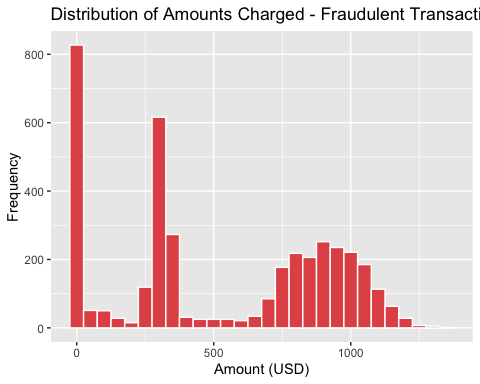
#The fraudulent transactions account for $21.196 for 2019 and $4143 for 2020

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

# Summary statistics for legitimate transactions  
legitimate\_summary <- card\_fraud %>%  
 filter(is\_fraud == 0) %>%  
 summarise(min\_amount = min(amt),  
 max\_amount = max(amt),  
 mean\_amount = mean(amt),  
 median\_amount = median(amt),  
 sd\_amount = sd(amt),  
 count=n())  
  
legitimate\_hist <- card\_fraud %>%  
 filter(is\_fraud == 0) %>%  
 ggplot(aes(x = amt)) +  
 geom\_histogram(binwidth = 50, fill = "#5989e3", color = "white") +  
 labs(title = "Distribution of Amounts Charged - Legitimate Transactions",  
 x = "Amount (USD)",  
 y = "Frequency")  
  
legitimate\_hist



# Summary statistics for fraudulent transactions  
fraudulent\_summary <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 summarise(min\_amount = min(amt),  
 max\_amount = max(amt),  
 mean\_amount = mean(amt),  
 median\_amount = median(amt),  
 sd\_amount = sd(amt))  
  
fraudulent\_hist <- card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 ggplot(aes(x = amt)) +  
 geom\_histogram(binwidth = 50, fill = "#e35656", color = "white") +  
 labs(title = "Distribution of Amounts Charged - Fraudulent Transactions",  
 x = "Amount (USD)",  
 y = "Frequency")  
  
fraudulent\_hist



legitimate\_summary #Display the table with summary statistics for the legitimate transactions

# A tibble: 1 × 6  
 min\_amount max\_amount mean\_amount median\_amount sd\_amount count  
 <dbl> <dbl> <dbl> <dbl> <dbl> <int>  
1 1 27120. 67.6 47.2 155. 667092

fraudulent\_summary #Display the table with summary statistics for the fraudulent transactions

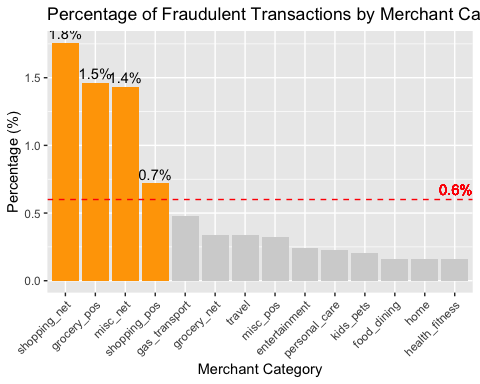
# A tibble: 1 × 5  
 min\_amount max\_amount mean\_amount median\_amount sd\_amount  
 <dbl> <dbl> <dbl> <dbl> <dbl>  
1 1.06 1334. 527. 369. 391.

* 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.

fraud\_by\_category <- card\_fraud %>%  
 group\_by(category) %>%  
 summarise(total\_fraud = sum(is\_fraud == 1), total\_transactions = n()) %>%  
 mutate(percentage = total\_fraud / total\_transactions \* 100) %>%  
 arrange(percentage)  
  
# Add a column for coloring the bars  
fraud\_by\_category <- fraud\_by\_category %>%  
 mutate(color = ifelse(row\_number() >= 11, "Top 4", "Other")) %>%   
 mutate(label = ifelse(row\_number() >= 11, paste0(round(percentage, 1), "%"), ""))  
  
# Define the predefined value for the horizontal line  
predefined\_value <- 0.6  
  
# Generate the bar chart with colored bars and horizontal line  
chart <- ggplot(fraud\_by\_category, aes(x = reorder(category, -percentage), y = percentage, fill = color)) +  
 geom\_bar(stat = "identity") +  
 geom\_hline(yintercept = predefined\_value, linetype = "dashed", color = "red") +  
 geom\_text(aes(label = label), vjust = -0.4) +  
 geom\_text(aes(x = Inf, y = predefined\_value, label = paste0(predefined\_value, "%")),  
 hjust = 1, vjust = -0.5, color = "red") +  
 labs(title = "Percentage of Fraudulent Transactions by Merchant Category",  
 x = "Merchant Category",  
 y = "Percentage (%)") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values = c("Top 4" = "orange", "Other" = "lightgrey")) +  
 guides(fill = FALSE)

Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as  
of ggplot2 3.3.4.

print(chart)



#Answer: the most common merchant categories for fraud are shopping\_net, grocery\_pos, misc\_net and shopping\_pos. Having said that, fraudulent transactions tend to happen more online and in shopping stores such as clothing or multi purpose shops for example. Also, there are fraudulent transactions in grocery stores

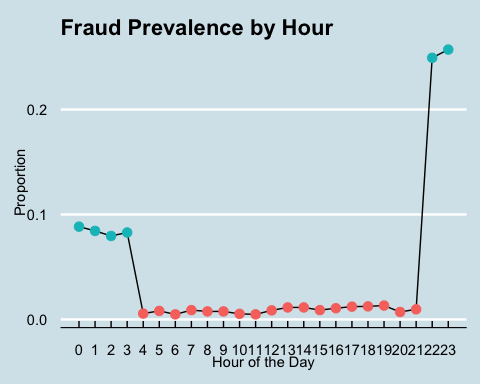
* When is fraud more prevalent? Which days, months, hours? To create new variables to help you in your analysis, we use the lubridate package and the following code

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)  
 )

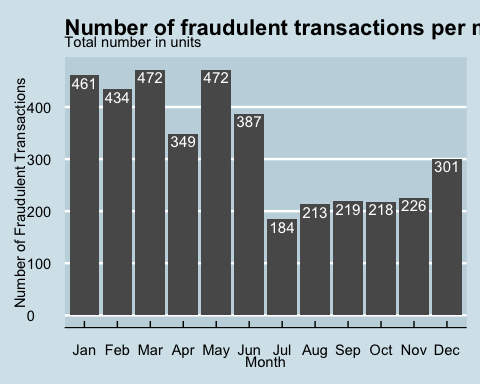
* Are older customers significantly more likely to be victims of credit card fraud? To calculate a customer’s age, we use the lubridate package and the following code

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

dfcardfraud <- card\_fraud %>%   
 mutate(  
 date\_only = lubridate::date(trans\_date\_trans\_time),  
 month\_name = lubridate::month(trans\_date\_trans\_time, label=TRUE),  
 hour = lubridate::hour(trans\_date\_trans\_time),  
 weekday = lubridate::wday(trans\_date\_trans\_time, label = TRUE)  
 )  
  
df\_cardfraud <- dfcardfraud %>%   
 filter(is\_fraud == 1) %>%   
 group\_by(hour) %>%   
 summarise(count=n()) %>%   
 mutate(prop = count/sum(count))  
  
chart <- ggplot(df\_cardfraud, aes(x = hour, y = prop)) +  
 geom\_line() +  
 geom\_point(aes(color = prop > 0.079), size = 3) +  
 theme\_economist() +  
 labs(title = "Fraud Prevalence by Hour",  
 x = "Hour of the Day",  
 y = "Proportion") +  
 scale\_x\_continuous(breaks = 0:23) +  
 guides(color = FALSE)  
  
chart



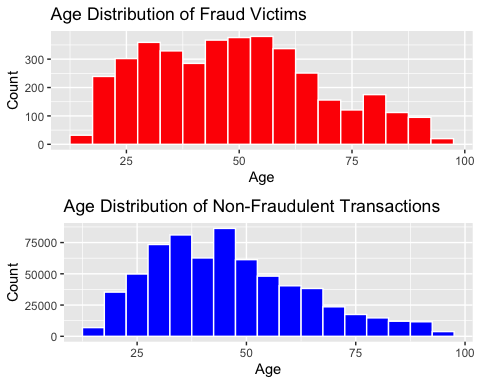
df\_cardfraud2 <- dfcardfraud %>%   
 filter(is\_fraud == 1) %>%   
 group\_by(month\_name) %>%   
 summarise(count=n()) %>%   
 mutate(prop = count/sum(count))  
  
# Specify the desired order of month names  
month\_order <- c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec")  
  
# Reorder the levels of month variable  
month\_counts <- month\_counts %>%  
 mutate(month = factor(month, levels = month\_order, ordered = TRUE))  
  
# Generate the bar chart  
chart2 <- ggplot(df\_cardfraud2, aes(x = month\_name, y = count)) +  
 geom\_bar(stat = "identity") +  
 theme\_economist(base\_family = "ITC Officina Sans", dkpanel = TRUE) +  
 scale\_fill\_manual(values = c("darkgrey", "orange"), guide = FALSE) +  
 scale\_colour\_economist() +  
 geom\_text(aes(label = count), vjust = 1.2, colour = "white", size = 4) +  
 labs(  
 title = "Number of fraudulent transactions per month",  
 subtitle = "Total number in units",  
 x = "Month",  
 y = "Number of Fraudulent Transactions"  
 )  
  
chart2



df\_cardfraud3 <- dfcardfraud %>%   
 filter(is\_fraud == 1) %>%   
 group\_by(weekday) %>%   
 summarise(count=n()) %>%   
 mutate(prop = count/sum(count))  
  
df\_cardfraud3

# A tibble: 7 × 3  
 weekday count prop  
 <ord> <int> <dbl>  
1 Sun 608 0.154  
2 Mon 639 0.162  
3 Tue 496 0.126  
4 Wed 468 0.119  
5 Thu 542 0.138  
6 Fri 557 0.142  
7 Sat 626 0.159

dfcardfraudage <- card\_fraud %>%   
 mutate(age = interval(dob, trans\_date\_trans\_time) / years(1),)  
  
# Filter the data for fraudulent transactions  
dfcardfraudagef <- dfcardfraudage %>%   
 filter(is\_fraud == 1)  
  
# Create a histogram of age for fraud victims  
fraud\_hist <- ggplot(dfcardfraudagef, aes(x = age)) +  
 geom\_histogram(binwidth = 5, fill = "red", color = "white") +  
 labs(title = "Age Distribution of Fraud Victims",  
 x = "Age",  
 y = "Count")  
  
# Create a histogram of age for non-fraudulent transactions  
dfcardfraudagenf <- dfcardfraudage %>%   
 filter(is\_fraud == 0)  
  
non\_fraud\_hist <- ggplot(dfcardfraudagenf, aes(x = age)) +  
 geom\_histogram(binwidth = 5, fill = "blue", color = "white") +  
 labs(title = "Age Distribution of Non-Fraudulent Transactions",  
 x = "Age",  
 y = "Count")  
  
# Combine the histograms into a single chart  
combined\_chart <- cowplot::plot\_grid(fraud\_hist, non\_fraud\_hist, ncol = 1)  
  
# Display the combined chart  
print(combined\_chart)



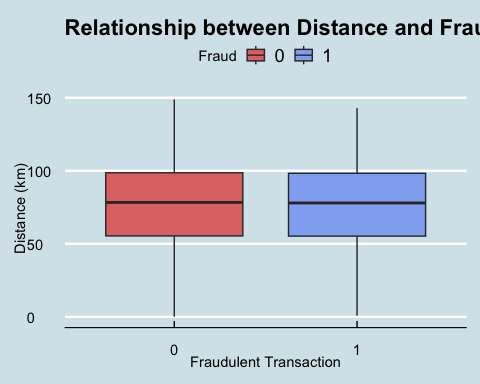
#Answer: When analyzing the information daily, I can conclude that fraud prevalence is clearly higher at late hours at night and early hours of the day. The correct range is from 22 up to 4.  
#On a yearly analysis, the first six months of the year tend to include more fraudulent transactions than the second half of the year  
#By weekday the proportion is quite stable ranging from 11% to 16% which means that fraudulent transactions happen regularly every day of the week.  
#Last, within the fraud victims, older customers do not suffer more fraudulent transactions than younger ones, meaning that the younger segment of the population is clearly more vulnerable than the older one, this could be because the older do not regularly shop online or use credit cards and rely mostly on cash to perform their daily shopping activities

* 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

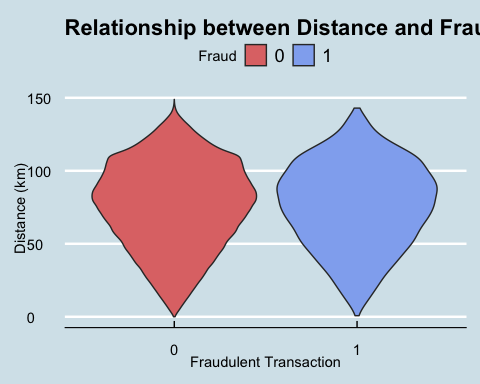
# distance between card holder's home and transaction  
# code adapted from https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/  
  
  
fraud <- card\_fraud %>%  
 mutate(  
   
 # convert latitude/longitude to radians  
 lat1\_radians = lat / 57.29577951,  
 lat2\_radians = merch\_lat / 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))  
  
 )  
  
fraud

# A tibble: 671,028 × 20  
 trans\_date\_trans\_time trans\_year category amt city state lat long  
 <dttm> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl>  
 1 2019-02-22 07:32:58 2019 entertainment 7.79 Veed… IN 40.1 -87.3  
 2 2019-02-16 15:07:20 2019 kids\_pets 3.89 Holl… OH 40.0 -81.0  
 3 2019-12-27 22:25:34 2019 personal\_care 8.43 Arno… MO 38.4 -90.4  
 4 2019-03-03 10:11:39 2019 grocery\_net 40 Apis… TN 35.0 -85.0  
 5 2019-02-09 17:14:54 2019 food\_dining 54.0 Red … CO 39.5 -106.   
 6 2019-09-09 01:19:59 2019 shopping\_net 95.6 Irwi… GA 32.8 -83.2  
 7 2019-12-15 09:11:53 2019 gas\_transport 65.0 Rani… MN 48.6 -93.3  
 8 2020-06-02 17:31:41 2020 home 3.41 Hunt… AL 34.6 -86.6  
 9 2019-12-28 16:58:06 2019 entertainment 69.1 Lore… TX 33.7 -102.   
10 2019-12-15 12:14:45 2019 entertainment 7.23 Bowd… ME 44.1 -70.0  
# ℹ 671,018 more rows  
# ℹ 12 more variables: city\_pop <dbl>, job <chr>, dob <date>, merch\_lat <dbl>,  
# merch\_long <dbl>, is\_fraud <dbl>, lat1\_radians <dbl>, lat2\_radians <dbl>,  
# long1\_radians <dbl>, long2\_radians <dbl>, distance\_miles <dbl>,  
# distance\_km <dbl>

# Boxplot with color differentiation  
ggplot(fraud, aes(x = factor(is\_fraud), y = distance\_km, fill = factor(is\_fraud))) +  
 geom\_boxplot() +  
 scale\_fill\_manual(values = c("#E07675", "#91AFF0")) +  
 theme\_economist() +  
 labs(title = "Relationship between Distance and Fraud",  
 x = "Fraudulent Transaction",  
 y = "Distance (km)",  
 fill = "Fraud")



# Violin plot with color differentiation  
ggplot(fraud, aes(x = factor(is\_fraud), y = distance\_km, fill = factor(is\_fraud))) +  
 geom\_violin() +  
 scale\_fill\_manual(values = c("#E07675", "#91AFF0")) +  
 theme\_economist() +  
 labs(title = "Relationship between Distance and Fraud",  
 x = "Fraudulent Transaction",  
 y = "Distance (km)",  
 fill= "Fraud")



#Answer: there's no clear relationship between fraud and distance since both fraudulent and non fraudulent transactions occur at relatively same distances

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

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

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

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

You will use

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

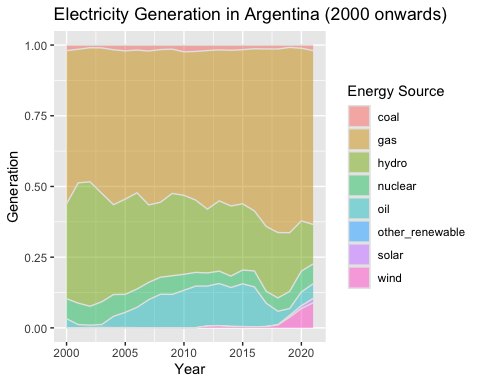
## 2. A scatter plot that looks at how CO2 per capita and GDP per capita are related

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

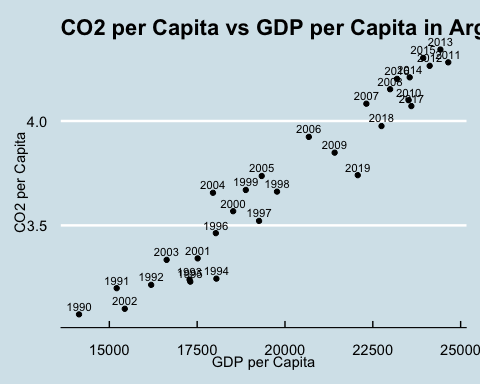
We will get energy data from the Our World in Data website, and CO2 and GDP per capita emissions from the World Bank, using the wbstatspackage.

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

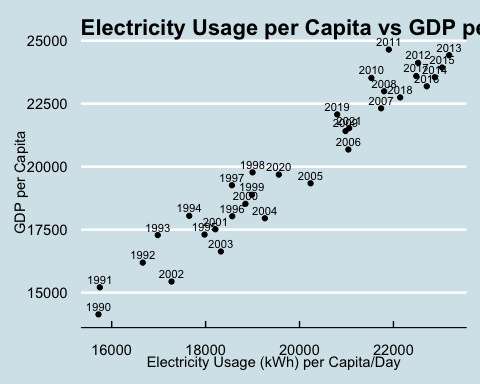
##1  
  
# Filter data for Argentina since 2000  
argentina\_energy <- energy %>%  
 filter(iso\_code == "ARG", year >= 2000)  
  
# Select the relevant columns for electricity generation  
argentina\_generation <- argentina\_energy %>%  
 select(year, coal, gas, hydro, nuclear, oil, other\_renewable, solar, wind)  
  
# Convert data from wide to long format  
argentina\_long <- argentina\_generation %>%  
 pivot\_longer(cols = -year, names\_to = "source", values\_to = "generation")  
  
# Create a stacked area chart for electricity generation  
ggplot(argentina\_long, aes(x = year, y = generation, fill = source)) +  
 geom\_area(colour = "grey90", alpha = 0.5, position = "fill") +  
 labs(title = "Electricity Generation in Argentina (2000 onwards)",  
 x = "Year",  
 y = "Generation",  
 fill = "Energy Source") +  
 scale\_fill\_discrete()



##2  
  
# Merge CO2 per capita and GDP per capita data using ISO code as the key  
co2\_gdp <- left\_join(co2\_percap, gdp\_percap, by = c("iso2c","iso3c","country","year"))  
  
co2\_gdpf <- co2\_gdp %>%   
 filter(!is.na(co2percap),!is.na(GDPpercap)) #Exclude missing values from the co2percap and GDPpercap variables  
  
argentina\_data <- co2\_gdpf %>%  
 filter(country == "Argentina") %>%  
 mutate(year = as.factor(year))  
  
ggplot(data = argentina\_data, aes(x = GDPpercap, y = co2percap)) +  
 geom\_point() +  
 geom\_text(aes(label = year), vjust = -0.5, hjust = 0.5, size = 3) +  
 labs(title = "CO2 per Capita vs GDP per Capita in Argentina",  
 x = "GDP per Capita",  
 y = "CO2 per Capita") +  
 theme\_economist()

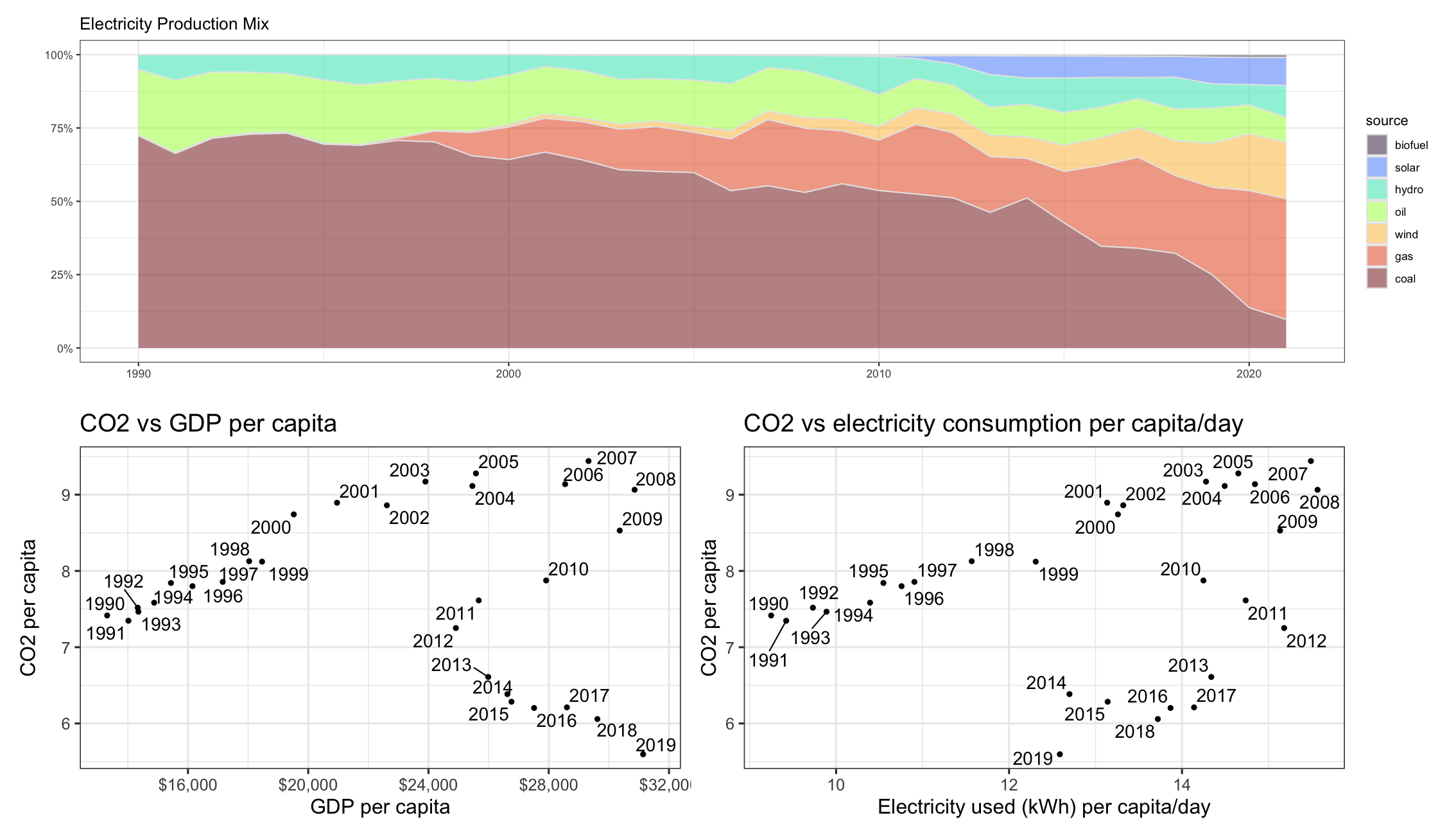


##3  
  
# Merge CO2 per capita and GDP per capita data using ISO code as the key  
energ\_gdp <- left\_join(energy, gdp\_percap, by = c("country","year"))  
  
energ\_gdpf <- energ\_gdp %>%   
 filter(!is.na(energy\_per\_capita),!is.na(GDPpercap)) #Exclude missing values from the energy\_per\_capita and GDPpercap variables  
  
argentina\_data <- energ\_gdpf %>%  
 filter(country == "Argentina") %>%  
 mutate(year = as.factor(year))  
  
# Create the scatter plot  
ggplot(data = argentina\_data, aes(x = energy\_per\_capita, y = GDPpercap)) +  
 geom\_point() +  
 geom\_text(aes(label = year), vjust = -0.5, hjust = 0.5, size = 3) +  
 labs(title = "Electricity Usage per Capita vs GDP per Capita",  
 x = "Electricity Usage (kWh) per Capita/Day",  
 y = "GDP per Capita") +  
 theme\_economist()



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



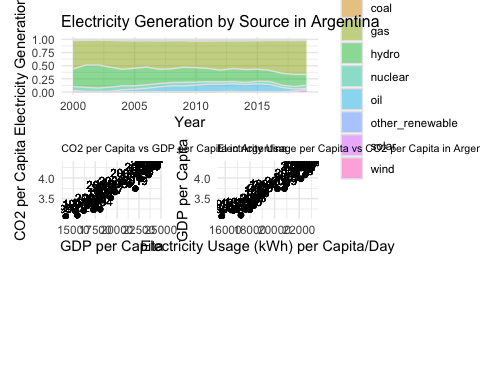
energy\_tidy <- energy %>%  
 pivot\_longer(cols = starts\_with(c("biofuel", "coal", "gas", "hydro", "nuclear", "oil", "other\_renewable", "solar", "wind")), #Pivoting the data set to make it long tidy format  
 names\_to = "source",  
 values\_to = "electricity") %>%   
 rename(iso3c = iso\_code) #renaming the iso\_code column to iso3c to merge it later  
  
  
merged\_data <- left\_join(gdp\_percap, co2\_percap, by = c("iso2c","iso3c","country","year")) %>% #Carry out a merge based on the by variables  
 left\_join(energy\_tidy, by = c("iso3c", "year")) #Merge again with the energy\_tidy table based on iso3c and year  
  
merged\_data

# A tibble: 51,874 × 19  
 indicator\_id.x indicator.x iso2c iso3c country.x year GDPpercap  
 <chr> <chr> <chr> <chr> <chr> <dbl> <dbl>  
 1 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 2 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 3 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 4 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 5 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 6 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 7 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 8 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
 9 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2021 1516.  
10 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AF AFG Afghanis… 2020 1968.  
# ℹ 51,864 more rows  
# ℹ 12 more variables: indicator\_id.y <chr>, indicator.y <chr>,  
# co2percap <dbl>, country.y <chr>, electricity\_demand <dbl>,  
# electricity\_generation <dbl>, net\_elec\_imports <dbl>,  
# energy\_per\_capita <dbl>, energy\_per\_gdp <dbl>,  
# per\_capita\_electricity <dbl>, source <chr>, electricity <dbl>

create\_country\_plots <- function(country\_name) {  
 iso\_code <- countrycode::countrycode(country\_name, "country.name", "iso3c") #Create a function whose input is the country name  
   
 country\_data <- merged\_data %>%  
 filter(iso3c == iso\_code, !is.na(GDPpercap), !is.na(co2percap)) #filter the missing values and set the iso3c equal to iso\_code  
   
 scatter\_plot <- ggplot(data = country\_data, aes(x = GDPpercap, y = co2percap)) + #plot the scatter plot from above inside the function  
 geom\_point() +  
 geom\_text(aes(label = year), vjust = -0.5, hjust = 0.5, size = 3) +  
 labs(  
 title = paste("CO2 per Capita vs GDP per Capita in", country\_name),  
 x = "GDP per Capita",  
 y = "CO2 per Capita"  
 ) +  
 theme\_minimal() +  
 theme(plot.title = element\_text(size = 8))  
   
 scatter\_plot2 <- ggplot(data = country\_data, aes(x = energy\_per\_capita, y = co2percap)) + #plot the second scatter plot  
 geom\_point() +  
 geom\_text(aes(label = year), vjust = -0.5, hjust = 0.5, size = 3) +  
 labs(title = paste("Electricity Usage per Capita vs CO2 per Capita in", country\_name),  
 x = "Electricity Usage (kWh) per Capita/Day",  
 y = "GDP per Capita") +  
 theme\_minimal() +  
 theme(plot.title = element\_text(size = 8))  
   
 stacked\_area\_chart <- country\_data %>% #plot the initial stacked are chart  
 filter(year >= 2000) %>%  
 group\_by(year,source) %>%  
 summarise(electricity = sum(electricity)) %>%  
 ggplot(aes(x = year, y = electricity, fill = source)) +  
 geom\_area(colour = "grey90", alpha = 0.5, position = "fill") +  
 labs(  
 title = paste("Electricity Generation by Source in", country\_name),  
 x = "Year",  
 y = "Electricity Generation",  
 fill = "Source"  
 ) +  
 theme\_minimal() +  
 theme(legend.position = "right", plot.title = element\_text(size = 12))  
   
 # Arrange the plots using patchwork  
 all\_plots <- (stacked\_area\_chart / ( scatter\_plot + scatter\_plot2 )) # Arrange them so that the order is similar to the image of the homework, make it so that the stacked area chart comes first and then the two smaller scatter plots come below it  
   
 all\_plots <- all\_plots + plot\_layout(ncol=1 ,nrow = 4, heights = c(2, 2, 2))  
}  
  
argentina\_plots <- create\_country\_plots("Argentina") #Test drive the function above with Argentina

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

argentina\_plots



# Deliverables

There is a lot of explanatory text, comments, etc. You do not need these, so delete them and produce a stand-alone document that you could share with someone. Knit the edited and completed Quarto Markdown (qmd) file as a Word document (use the “Render” button at the top of the script editor window) and upload it to Canvas. You must be commiting and pushing tour changes to your own Github repo as you go along.

# Details

* Who did you collaborate with: TYPE NAMES HERE
* Approximately how much time did you spend on this problem set: 10 hours
* What, if anything, gave you the most trouble: The last exercise was really troublesome, especially the triple merge with different keys

**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.