Homerwork 2

Juan Aranguren

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 data frame that summarizes the number of mass shootings per year  
Data\_frame\_mass\_shootings <- mass\_shootings %>%   
 group\_by(year) %>%   
 summarise(count = n())  
  
#Show results  
Data\_frame\_mass\_shootings

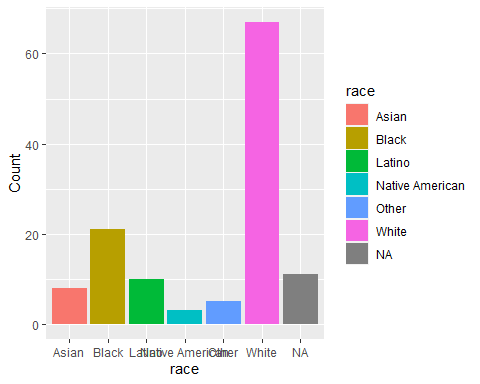
# A tibble: 37 × 2  
 year count  
 <dbl> <int>  
 1 1982 1  
 2 1984 2  
 3 1986 1  
 4 1987 1  
 5 1988 1  
 6 1989 2  
 7 1990 1  
 8 1991 3  
 9 1992 2  
10 1993 4  
# ℹ 27 more rows

* 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 the bar chat  
bar\_chart <- mass\_shootings %>%   
 group\_by(race) %>%   
 summarise(Count = n()) %>%   
 arrange(desc(Count)) %>%   
 ggplot(aes(x = race, y = Count, fill = race )) +  
 geom\_bar(stat = "identity")  
 labs(x = "Race", y = "Number of mass shooters", title = "Number of mass shooters by Race") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1))

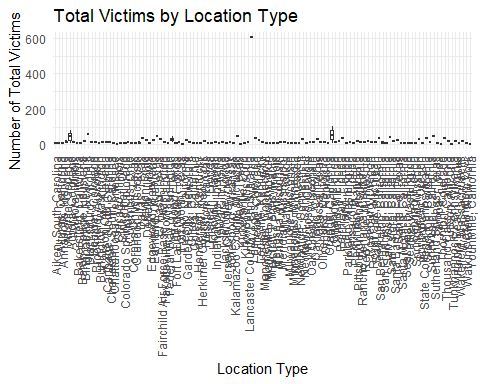
NULL

#Show the result  
bar\_chart



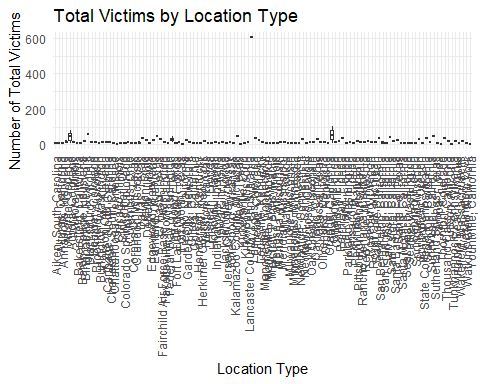
* Generate a boxplot visualizing the number of total victims, by type of location.

#Generate boxplot visualizing the number of total victims by type of location. In the X-asis I will put the location and in the Y x-sis I will put the total victims.  
boxplot <- ggplot(mass\_shootings, aes(x = location, y = total\_victims)) +  
 geom\_boxplot() +  
 labs(x = "Location Type", y = "Number of Total Victims", title = "Total Victims by Location Type") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1))  
  
# Show the results  
boxplot



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

#filter the data base by taking out the Las Vegas Strip masacre  
filter\_las\_vegas <- mass\_shootings %>%   
 filter(!grepl("Las Vegas Strip", location))  
  
#Execute the same code as before for the box plot  
boxplot\_no\_las\_vegas <- ggplot(filter\_las\_vegas, aes(x = location, y = total\_victims)) +  
 geom\_boxplot() +  
 labs(x = "Location Type", y = "Number of Total Victims", title = "Total Victims by Location Type") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1))  
  
#Show the results  
boxplot\_no\_las\_vegas



### More open-ended questions

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

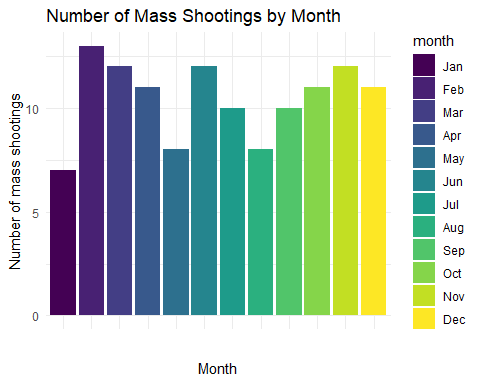
* How many white males with prior signs of mental illness initiated a mass shooting after 2000?

# Frist, filter the data for white males with prior signs of mental illness after 2000  
WM\_Mental\_Illness <- mass\_shootings %>%  
 filter(race == "White", male == TRUE, prior\_mental\_illness == "Yes", year >= 2000)  
  
# Calculate the count  
count <- nrow(WM\_Mental\_Illness)  
  
#Show Result  
count

[1] 23

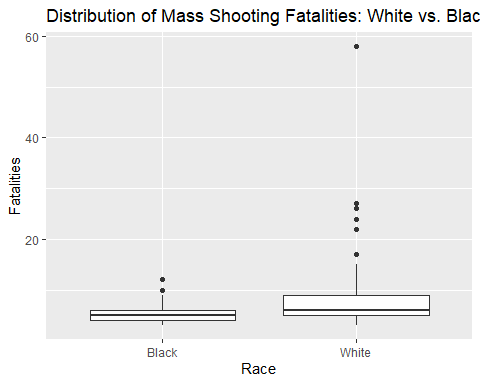
* Which month of the year has the most mass shootings? Generate a bar chart sorted in chronological (natural) order (Jan-Feb-Mar- etc) to provide evidence of your answer.

#First, let's count the number of shootings by month  
shootings\_month <- mass\_shootings %>%  
 group\_by(month) %>%   
 summarize(count = n())  
  
# 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\_order <- shootings\_month %>%  
 mutate(month = factor(month, levels = month\_order, ordered = TRUE))  
  
#Then we generate the bar chart  
bar\_chart <- ggplot(month\_order, aes(x = month, y = count, fill = month)) +  
 geom\_bar(stat = "identity") +  
 labs(x = "Month", y = "Number of mass shootings", title = "Number of Mass Shootings by Month") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 50, hjust = 1))  
  
#Show the result  
bar\_chart

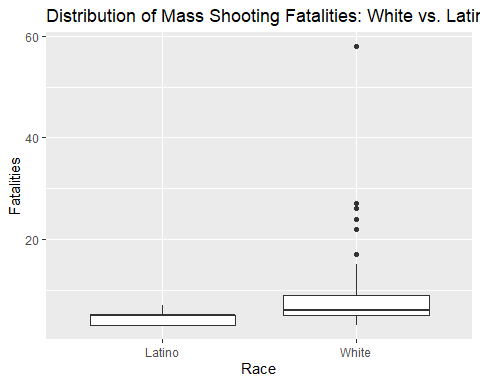


* How does the distribution of mass shooting fatalities differ between White and Black shooters? What about White and Latino shooters?

#To analyze how the mass shooting fatalities differ between white and Black people and white and Latino, we first filter the data for bouth groups  
WB <- mass\_shootings %>%   
 filter(race %in% c("White","Black"))  
  
WL <- mass\_shootings %>%   
 filter(race %in% c("White","Latino"))  
  
#Then we create the boxplots for both combinations  
Boxplot\_WB <- ggplot(WB, aes(x = race, y = fatalities)) +  
 geom\_boxplot() +  
 labs(x = "Race", y = "Fatalities", title = "Distribution of Mass Shooting Fatalities: White vs. Black Shooters")  
   
Boxplot\_WL <- ggplot(WL, aes(x = race, y = fatalities)) +  
 geom\_boxplot() +  
 labs(x = "Race", y = "Fatalities", title = "Distribution of Mass Shooting Fatalities: White vs. Latino Shooters")  
  
#finally, we display both results  
Boxplot\_WB



Boxplot\_WL



### 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?

#Let's first filtrate those with prior mental illness from those who did not  
Mental\_Illness <- mass\_shootings %>%   
 mutate(Mental\_Illness\_X = if\_else(!is.na(prior\_mental\_illness), "Yes", "No"))  
  
#Count the amount of people with and without  
MI\_Count <- Mental\_Illness %>%   
 count(Mental\_Illness\_X)  
  
#Show the result  
MI\_Count

# A tibble: 2 × 2  
 Mental\_Illness\_X n  
 <chr> <int>  
1 No 46  
2 Yes 79

#Create a bar chart to show the differences  
Bar\_Chart\_MI <- ggplot(MI\_Count, aes(x = Mental\_Illness\_X, y = n, fill = Mental\_Illness\_X)) +  
 geom\_bar(stat = "identity") +  
 labs(x = "Mental Illness", y = "Number of Mass Shootings", title = "Mass Shootings by Mental Illness") +  
 theme\_minimal()  
  
#Analyzing the numbers only we can see that there are much more people that have a prior mental illness involve in mass shootings. Even though we see this trend, we should consider for other reasons than the prior mental illness to decided whether this is a relevant variable or not

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

#Let's first filtrate those with prior mental illness from those who did not  
Mental\_Illness <- mass\_shootings %>%   
 mutate(Mental\_Illness\_X = if\_else(!is.na(prior\_mental\_illness), "Yes", "No"))  
  
#Let's then understand the relationship between mental illness and the total number of victims  
MI\_TV <- Mental\_Illness %>%   
 group\_by(Mental\_Illness\_X) %>%   
 summarize(Victims\_AVG = mean(total\_victims))  
  
#Show the result  
MI\_TV

# A tibble: 2 × 2  
 Mental\_Illness\_X Victims\_AVG  
 <chr> <dbl>  
1 No 26.8  
2 Yes 15.5

#Let's then understand the relationship between mental illness and the location type  
MI\_LT <- Mental\_Illness %>%   
 group\_by(Mental\_Illness\_X, location\_type) %>%   
 summarize(Count = n())

`summarise()` has grouped output by 'Mental\_Illness\_X'. You can override using  
the `.groups` argument.

#Show the result  
MI\_LT

# A tibble: 11 × 3  
# Groups: Mental\_Illness\_X [2]  
 Mental\_Illness\_X location\_type Count  
 <chr> <chr> <int>  
 1 No Military 4  
 2 No Other 17  
 3 No Religious 2  
 4 No School 5  
 5 No Workplace 18  
 6 Yes Airport 1  
 7 Yes Military 2  
 8 Yes Other 32  
 9 Yes Religious 4  
10 Yes School 13  
11 Yes Workplace 27

#Now let's understand the instersection of all 3 of them  
Intersection <- mass\_shootings %>%   
 group\_by(prior\_mental\_illness, location\_type) %>%   
 summarize(Victims\_AVG = mean(total\_victims))

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

#Show Result  
Intersection

# A tibble: 14 × 3  
# Groups: prior\_mental\_illness [3]  
 prior\_mental\_illness location\_type Victims\_AVG  
 <chr> <chr> <dbl>  
 1 No Other 14.2   
 2 No School 13.3   
 3 No Workplace 7.83  
 4 Yes Airport 11   
 5 Yes Military 24   
 6 Yes Other 16.7   
 7 Yes Religious 20.5   
 8 Yes School 28.9   
 9 Yes Workplace 9.33  
10 <NA> Military 19.2   
11 <NA> Other 50.1   
12 <NA> Religious 13.5   
13 <NA> School 13   
14 <NA> Workplace 11.8

#here we can also see that when there is a prior mental illness, the average number of victims increases. In addition, considering that there were much more people with prior illness involve in mass shootings in the dataset, we can see that they were involved in much more locations as well

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 <chr> "10/19/1959", "4/3/1946", "3/31/1985", "1/28/199…  
$ 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.

#First we are going to create the table grouped by transaction year that calculates de number of frauds they were and how frequent they occured. As sated by the text, the number should be very low  
Fraudulent\_Transactions <- card\_fraud %>%   
 group\_by(trans\_year) %>%   
 summarise(Number\_Fraud = sum(is\_fraud),Frequency = Number\_Fraud/n())  
  
#Show result  
Fraudulent\_Transactions

# A tibble: 2 × 3  
 trans\_year Number\_Fraud Frequency  
 <dbl> <dbl> <dbl>  
1 2019 2721 0.00568  
2 2020 1215 0.00632

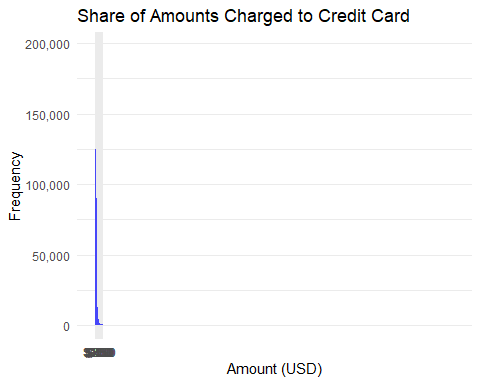
* How much money (in US$ terms) are fraudulent transactions costing the company? Generate a table that summarizes the total amount of legitimate and fraudulent transactions per year and calculate the % of fraudulent transactions, in US$ terms.

#We are going to create a table that summarises the amount of money that was fraudulent or legitimate and calculate its share   
  
F\_L\_Transaction <- card\_fraud %>%   
 group\_by(trans\_year) %>%   
 summarise(Leigitimate\_Money = sum(amt\*(1-is\_fraud)),  
 Fraudulent\_Money = sum(amt\*is\_fraud),  
 Share = (Fraudulent\_Money/(Leigitimate\_Money + Fraudulent\_Money) \* 100))  
#Show result  
F\_L\_Transaction

# A tibble: 2 × 4  
 trans\_year Leigitimate\_Money Fraudulent\_Money Share  
 <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.

# First we are going to clasify the dataset for legitimate and fraudulent transactions  
Legitimate <- card\_fraud %>%   
 filter(is\_fraud == 0)  
Fraudulent <- card\_fraud %>%   
 filter(is\_fraud == 1)  
  
# Then we are going to create the histograms  
ggplot() +  
 geom\_histogram(data = Legitimate, aes(x = amt), fill = "blue", alpha = 0.7, binwidth = 30) +  
 geom\_histogram(data = Fraudulent, aes(x = amt), fill = "red", alpha = 0.7, binwidth = 30) +  
 labs(x = "Amount (USD)", y = "Frequency", title = "Share of Amounts Charged to Credit Card") +  
 scale\_x\_continuous(labels = scales::dollar\_format(prefix = "$"), breaks = seq(0, 500, by = 50)) +  
 scale\_y\_continuous(labels = scales::comma\_format()) +  
 theme\_minimal()



# Finally, we are going to calculate summary statistics for each  
Legitimate\_summary <- summary(Legitimate$amt)  
Fraudulent\_summary <- summary(Fraudulent$amt)  
  
#Show results  
Legitimate\_summary

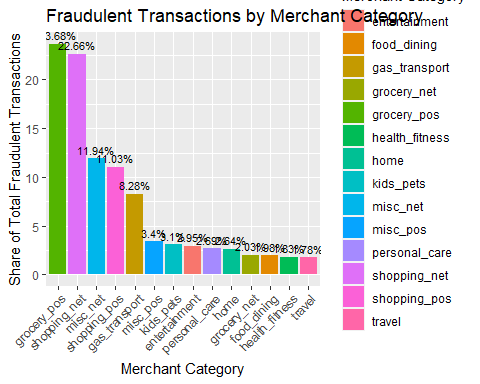
Min. 1st Qu. Median Mean 3rd Qu. Max.   
 1.00 9.60 47.17 67.62 82.41 27119.77

Fraudulent\_summary

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 1.06 240.49 368.83 527.21 900.95 1334.07

* 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.

# First we are gonna filter the data for fraudulent transactions  
Fraudulent <- card\_fraud %>%   
 filter(is\_fraud == 1)  
  
# Then we are going to calculate the percentage of fraudulent transactions for each merchant category  
category\_summary <- Fraudulent %>%  
 group\_by(category) %>%  
 summarise(Share\_Fraudulent = (sum(is\_fraud) / sum(Fraudulent$is\_fraud)) \* 100,  
 cumulative\_percent = cumsum(Share\_Fraudulent))  
  
# Then, we are going to create a sorted bar chart  
ggplot(category\_summary, aes(x = reorder(category, -Share\_Fraudulent), y = Share\_Fraudulent, fill = category)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = paste0(round(Share\_Fraudulent, 2), "%")), vjust = -0.5, color = "black", size = 3) +  
 labs(x = "Merchant Category", y = "Share of Total Fraudulent Transactions", title = "Fraudulent Transactions by Merchant Category") +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_discrete(name = "Merchant Category")



* 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),  
 )

# First we are going to create new variables for analysis with the provided code  
card\_fraud <- 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)  
 )  
  
# Then we are going to analyse whether fraud has a prevalence in any day of the week  
Day\_week <- card\_fraud %>%  
 group\_by(weekday) %>%  
 summarise(fraud\_count = sum(is\_fraud),  
 total\_count = n(),  
 share\_fraudulent = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(share\_fraudulent))  
  
# Then we are going to analyse whether fraud has a prevalence in any month  
Month\_analysis <- card\_fraud %>%  
 group\_by(month\_name) %>%  
 summarise(fraud\_count = sum(is\_fraud),  
 total\_count = n(),  
 share\_fraudulent = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(share\_fraudulent))  
  
# Then we are going to analyse whether fraud has a prevalence in any hour  
Hour\_analysis <- card\_fraud %>%  
 group\_by(hour) %>%  
 summarise(fraud\_count = sum(is\_fraud),  
 total\_count = n(),  
 share\_fraudulent = (fraud\_count / total\_count) \* 100) %>%  
 arrange(desc(share\_fraudulent))  
  
# Then we are using the provided code to analyse whether older customers are more likely to be part of credit fraud  
card\_fraud <- card\_fraud %>%  
 mutate(  
 age = interval(dob, trans\_date\_trans\_time) / years(1)  
 )

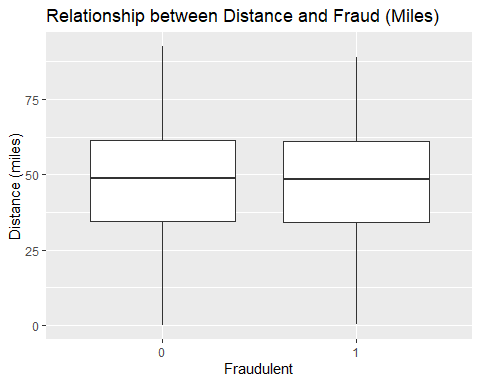
Warning: There was 1 warning in `mutate()`.  
ℹ In argument: `age = interval(dob, trans\_date\_trans\_time)/years(1)`.  
Caused by warning:  
! All formats failed to parse. No formats found.

# We will compare the age of every transaction and see if the fraudulent ones are more commmon in older people  
average\_age\_summary <- card\_fraud %>%  
 group\_by(is\_fraud) %>%  
 summarise(average\_age = mean(age, na.rm = TRUE))  
  
#Show results  
average\_age\_summary

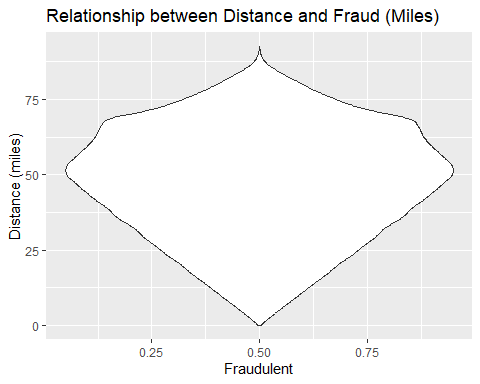
# A tibble: 2 × 2  
 is\_fraud average\_age  
 <dbl> <dbl>  
1 0 NaN  
2 1 NaN

* 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))  
  
 )  
  
# Boxplot  
boxplot\_plot <- ggplot(fraud, aes(x = factor(is\_fraud), y = distance\_miles, group = factor(is\_fraud))) +  
 geom\_boxplot() +  
 labs(x = "Fraudulent", y = "Distance (miles)") +  
 ggtitle("Relationship between Distance and Fraud (Miles)")  
  
# Violin plot  
violin\_plot <- ggplot(fraud, aes(x = is\_fraud, y = distance\_miles)) +  
 geom\_violin() +  
 labs(x = "Fraudulent", y = "Distance (miles)") +  
 ggtitle("Relationship between Distance and Fraud (Miles)")  
  
# Plot the boxplot and violin plot  
boxplot\_plot



violin\_plot



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

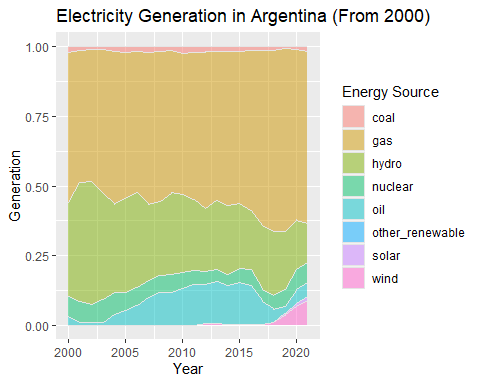
geom\_area(colour="grey90", alpha = 0.5, position = "fill")

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

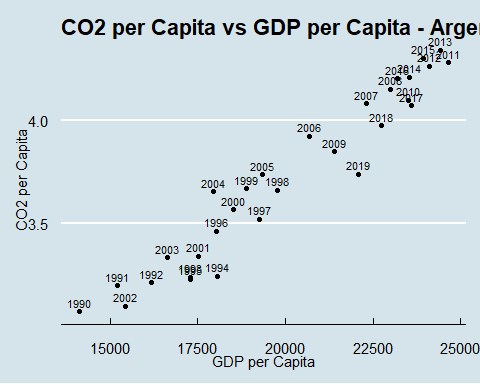
## 3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related

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.

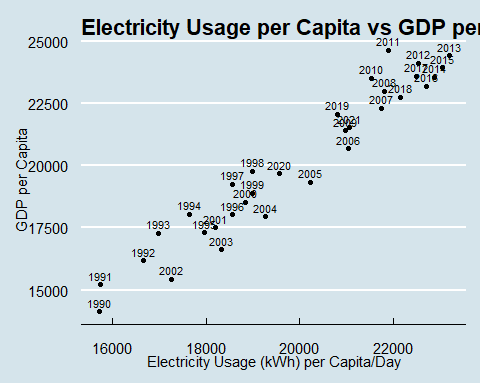
# Download electricity data  
url <- "https://nyc3.digitaloceanspaces.com/owid-public/data/energy/owid-energy-data.csv"  
  
energy <- read\_csv(url) %>%   
 filter(year >= 1990) %>%   
 drop\_na(iso\_code) %>%   
 select(1:3,  
 biofuel = biofuel\_electricity,  
 coal = coal\_electricity,  
 gas = gas\_electricity,  
 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 = "Argentina",   
 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 = "Argentina",   
 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)  
  
#Question 1  
# First, we are going to filter data for Argentina since 2000  
arg\_energy <- energy %>%  
 filter(iso\_code == "ARG", year >= 2000)  
  
# Then we are going to select the needed columns for electricity generation  
arg\_elec\_generation <- arg\_energy %>%  
 select(year, coal, gas, hydro, nuclear, oil, other\_renewable, solar, wind)  
  
# Then we are going to convert the data from wide to long format  
arg\_format <- arg\_elec\_generation %>%  
 pivot\_longer(cols = -year, names\_to = "source", values\_to = "generation")  
  
# Finally, we are going to create a stacked area chart for electricity generation  
ggplot(arg\_format, aes(x = year, y = generation, fill = source)) +  
 geom\_area(colour = "grey90", alpha = 0.5, position = "fill") +  
 labs(title = "Electricity Generation in Argentina (From 2000)",  
 x = "Year",  
 y = "Generation",  
 fill = "Energy Source") +  
 scale\_fill\_discrete()



#Question 2  
library("ggthemes")  
# First we are going to merge CO2 per capita and GDP per capita data  
co2\_gdp <- left\_join(co2\_percap, gdp\_percap, by = c("iso2c","iso3c","country","year"))  
  
# Then, we are going to exlcude missing values  
co2\_gdpf <- co2\_gdp %>%   
 filter(!is.na(co2percap),!is.na(GDPpercap))  
  
#Then, we will filter the information for Argentina (Even though I already filtered from the original statement provided just to be sure)  
ARG <- co2\_gdpf %>%  
 filter(country == "Argentina") %>%  
 mutate(year = as.factor(year))  
  
#Finally, we are going to do the plot of CO2 vs GDP in Argentina  
ggplot(data = ARG, 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 - Argentina",  
 x = "GDP per Capita",  
 y = "CO2 per Capita") +  
 theme\_economist()

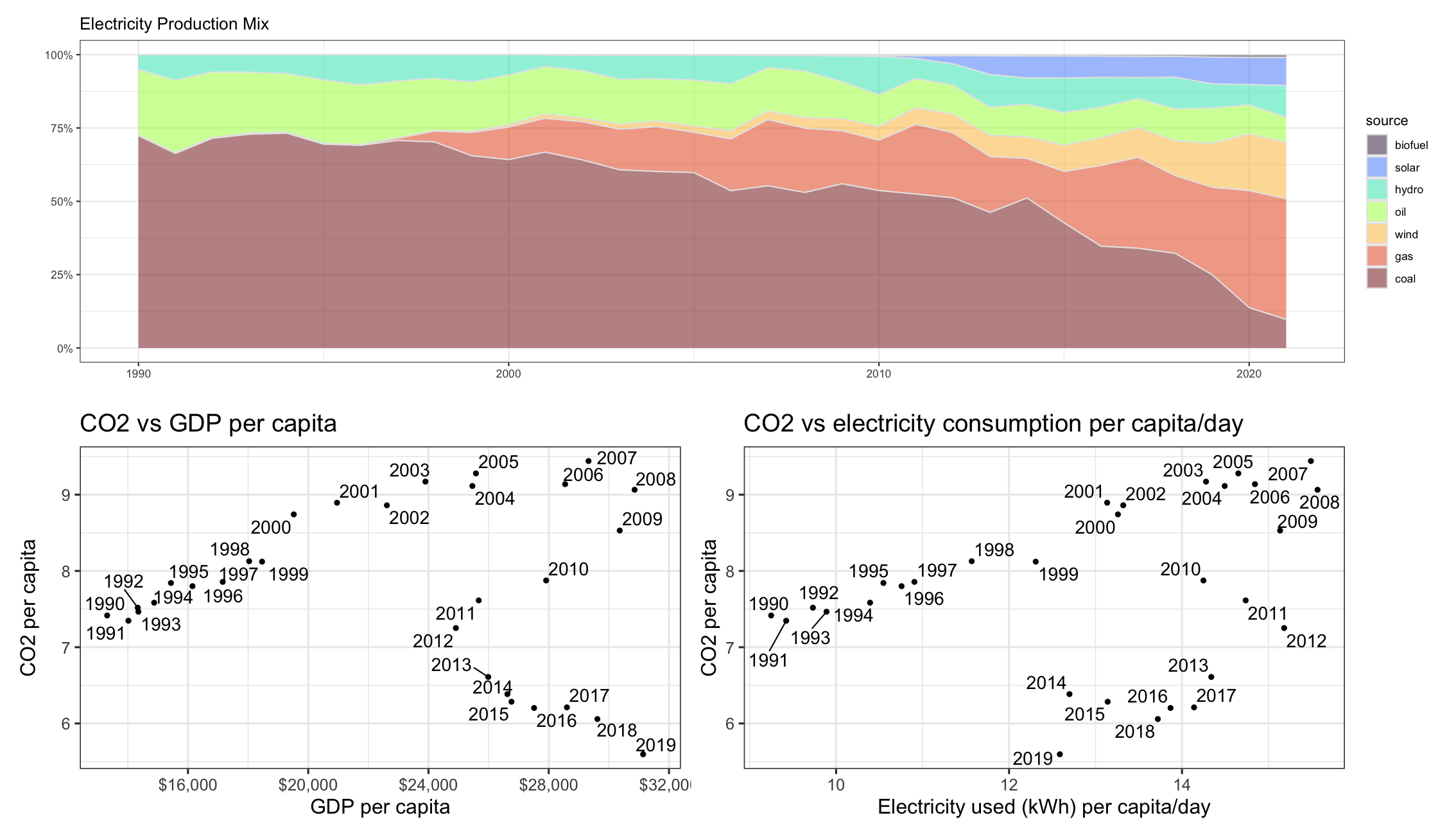


#Question 3  
# First, we are going to merge CO2 per capita and GDP per capita data  
energ\_gdp <- left\_join(energy, gdp\_percap, by = c("country","year"))  
  
# Then, we are going to exlcude missing values  
energ\_gdpf <- energ\_gdp %>%   
 filter(!is.na(energy\_per\_capita),!is.na(GDPpercap))  
  
#Then, we will filter the information for Argentina (Even though I already filtered from the original statement provided just to be sure)  
ARG <- energ\_gdpf %>%  
 filter(country == "Argentina") %>%  
 mutate(year = as.factor(year))  
  
# Finally, we are going to create the scatter plot  
ggplot(data = ARG, 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



# First, we are going to pivot the data set to make it long tidy format and renamed the iso\_code column for future use as a key  
energy\_tidy <- energy %>%  
 pivot\_longer(cols = starts\_with(c("biofuel", "coal", "gas", "hydro", "nuclear", "oil", "other\_renewable", "solar", "wind")),   
 names\_to = "source",  
 values\_to = "electricity") %>%   
 rename(iso3c = iso\_code)  
  
#Then we are going to merge the information  
merged\_data <- left\_join(gdp\_percap, co2\_percap, by = c("iso2c","iso3c","country","year")) %>%  
 left\_join(energy\_tidy, by = c("iso3c", "year"))  
  
merged\_data

# A tibble: 288 × 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 … AR ARG Argentina 2021 21527.  
 2 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 3 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 4 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 5 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 6 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 7 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 8 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
 9 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2021 21527.  
10 NY.GDP.PCAP.PP.KD GDP per capita, PPP … AR ARG Argentina 2020 19685.  
# ℹ 278 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>

# Then we re going to create a function with country\_name as the input  
create\_country\_plots <- function(country\_name) {  
 iso\_code <- countrycode::countrycode(country\_name, "country.name", "iso3c")  
   
# Then we are going to filter missing values  
 country\_data <- merged\_data %>%  
 filter(iso3c == iso\_code, !is.na(GDPpercap), !is.na(co2percap))   
   
# Then, we are going to do the scatter plot   
 scatter\_plot <- ggplot(data = country\_data, aes(x = GDPpercap, y = co2percap)) +  
 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))  
  
# Then, plot the other one   
 scatter\_plot2 <- ggplot(data = country\_data, aes(x = energy\_per\_capita, y = co2percap)) +   
 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))  
   
# Then plot the initial stacked area chart  
 stacked\_area\_chart <- country\_data %>%   
 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))  
   
# Finally, we are going to arrange the plots using patchwork  
 all\_plots <- (stacked\_area\_chart / ( scatter\_plot + scatter\_plot2 ))  
   
 all\_plots <- all\_plots + plot\_layout(ncol=1 ,nrow = 4, heights = c(2, 2, 2))  
}

# 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: Guido Bozzano in the electricty excercise for Argentina
* Approximately how much time did you spend on this problem set: 2 days
* What, if anything, gave you the most trouble: Arranging one of the graphs by order of month in written format. Finally, the electricity excercise. Completely

**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.