Table of Contents

title: “Homework 2”

author: “Assad Ahmed”

date: “2023-05-22”

output:

word\_document:

toc: yes

pdf\_document:

toc: yes

html\_document:

theme: flatly  
  
highlight: zenburn  
  
number\_sections: yes  
  
toc: yes  
  
toc\_float: yes  
  
code\_folding: show

#| label: load-libraries  
  
#| echo: false # This option disables the printing of code (only output is displayed).  
  
#| message: false  
  
#| warning: false  
  
  
  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.2 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(wbstats)  
  
library(skimr)  
  
library(countrycode)  
  
library(here)

## here() starts at C:/Users/assad/OneDrive/Documents/dsb2023

library(ggforce)  
  
library(ggthemes)  
  
#added 2 new libraries for utilisation later on

#loading in the data

#| echo: false  
  
#| message: false  
  
#| warning: false  
  
  
  
  
  
mass\_shootings <- read\_csv(here::here("data", "mass\_shootings.csv"))

## Rows: 125 Columns: 14  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (7): case, month, location, summary, location\_type, race, prior\_mental\_i...  
## dbl (6): year, day, fatalities, injured, total\_victims, age\_of\_shooter  
## lgl (1): male  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(mass\_shootings)

## 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, …

head(mass\_shootings)

## # A tibble: 6 × 14  
## case year month day location summary fatalities injured total\_victims  
## <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Oxford Hi… 2021 Nov 30 Oxford,… Ethan … 4 7 11  
## 2 San Jose … 2021 May 26 San Jos… Samuel… 9 0 9  
## 3 FedEx war… 2021 Apr 15 Indiana… Brando… 8 7 15  
## 4 Orange of… 2021 Mar 31 Orange,… Aminad… 4 1 5  
## 5 Boulder s… 2021 Mar 22 Boulder… Ahmad … 10 0 10  
## 6 Atlanta m… 2021 Mar 16 Atlanta… Robert… 8 1 9  
## # ℹ 5 more variables: location\_type <chr>, male <lgl>, age\_of\_shooter <dbl>,  
## # race <chr>, prior\_mental\_illness <chr>

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 occurred |

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 occurred |

race | race of the shooter |

prior\_mental\_illness | did the shooter show evidence of mental illness prior to the incident? |

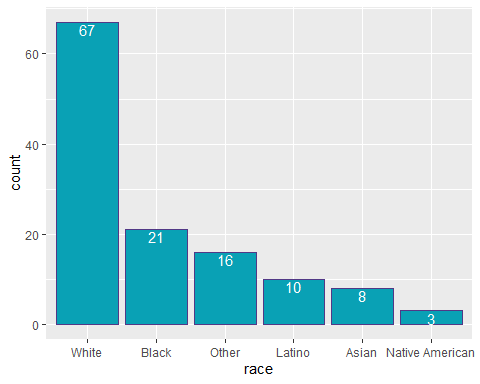
* Generate a data frame that summarizes the number of mass shootings per year.

#Create a new data frame for mass shootings per year called mass\_shootings\_yearly  
  
  
  
mass\_shootings\_yearly<- mass\_shootings %>%   
  
 #Filter out any Na's  
  
 filter(!is.na(year)) %>%   
  
 #group by year and summarize the count of mass shootings  
  
 group\_by(year) %>%   
  
 summarize(Count = n())  
  
  
  
mass\_shootings\_yearly

## # 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

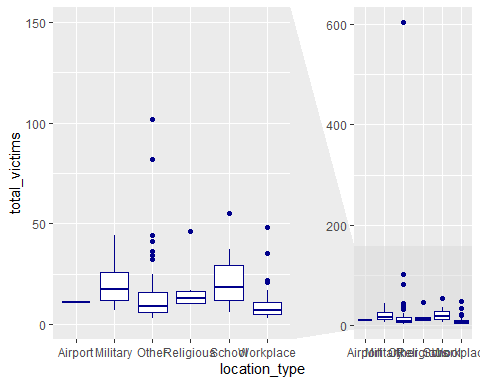
* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

#Creating a bar chart for number of mass shooters with each race category  
  
mass\_shootings %>%   
  
 #replace NA race entries with 'Unknown'  
  
 #reorder from most to least frequent race  
  
 mutate(race= case\_when( !is.na(race) ~ race, TRUE ~ 'Other'),race=fct\_infreq(race)) %>%  
  
 #use GGplot to produce a bar char, initialise x variable and aes  
  
 ggplot(aes(x=race)) +  
  
 #plot on a barchart  
  
 geom\_bar(fill='#09A1B5',color="#523886") +  
  
 #add aesthetic label  
  
 geom\_text(aes(label = after\_stat(count)), stat = "count", vjust = 1.1, colour = "white")



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

#assume total victims means to exclude the shooter from the total  
  
#manipulate data for the boxplot  
  
mass\_shootings %>%   
  
 #Filter out for victim count NA  
  
 filter(!is.na(total\_victims)) %>%   
  
 #replace NA race entries with 'Unknown'  
  
 mutate(location= case\_when( !is.na(location\_type) ~ location\_type, TRUE ~ 'Other')) %>%  
  
 #Create a plot  
  
 ggplot(aes(x=location\_type,y=total\_victims)) +  
  
 geom\_boxplot(color='darkblue') +  
  
 #facet zoom to better see the plots on a more reasonable scale due to the outlier  
  
 facet\_zoom(ylim = c(0, 150))

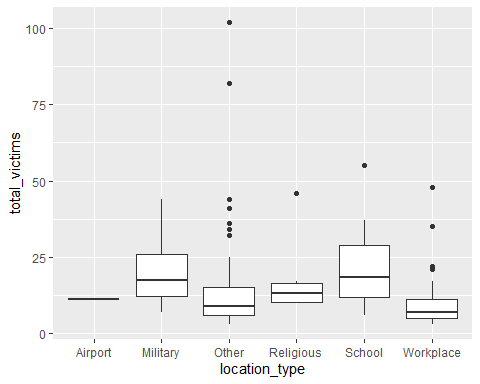


* Redraw the same plot, but remove the Las Vegas Strip massacre from the dataset.

#View data set for las vegas strip massacre only  
  
mass\_shootings %>%   
  
 #only filter for las vegas strip  
  
 filter(grepl("Las Vegas Strip",case))

## # A tibble: 1 × 14  
## case year month day location summary fatalities injured total\_victims  
## <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Las Vegas… 2017 Oct 1 Las Veg… "Steph… 58 546 604  
## # ℹ 5 more variables: location\_type <chr>, male <lgl>, age\_of\_shooter <dbl>,  
## # race <chr>, prior\_mental\_illness <chr>

#only shows the 1 entry as required  
  
  
  
#assume total victims means to exclude the shooter from the total  
  
#manipulate data for the boxplot  
  
mass\_shootings %>%   
  
 #Filter out for victim count NA and for the las vegas strip massacre  
  
 filter(!is.na(total\_victims) & !grepl("Las Vegas Strip",case)) %>%   
  
 #replace NA race entries with 'Unknown'  
  
 mutate(location= case\_when( !is.na(location\_type) ~ location\_type, TRUE ~ 'Other')) %>%  
  
 #Create a plot  
  
 ggplot(aes(x=location\_type,y=total\_victims)) +  
  
 geom\_boxplot()



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

Prior\_Mental\_2000 <- mass\_shootings %>%   
  
 #Filter for year>= 2000, for white males only  
  
 filter (year>=2000 & race=='White' & male==TRUE) %>%   
  
 #Convert NA's to a new category 'Unknown'  
  
 #this will allow us to see what proportion of mass shooters have a hidden history in terms of mental illness which will help to validate/invalidate any conclusions  
  
 mutate(prior\_mental\_illness= case\_when( !is.na(prior\_mental\_illness)   
  
 ~ prior\_mental\_illness, TRUE ~ 'Unknown'),  
  
 #add an arbitrary weighting column  
  
 n=1,   
  
 #add column calculating the % share each case has to the total based on current equal weighting  
  
 percent=n/sum(n)) %>%  
  
 #group by prior\_mental\_illness  
  
 group\_by(prior\_mental\_illness) %>%   
  
 #summaries data for count of each  
  
 summarize(Count = n(),  
  
 percent = sum(percent))  
  
  
  
Prior\_Mental\_2000

## # A tibble: 3 × 3  
## prior\_mental\_illness Count percent  
## <chr> <int> <dbl>  
## 1 No 4 0.0889  
## 2 Unknown 18 0.4   
## 3 Yes 23 0.511

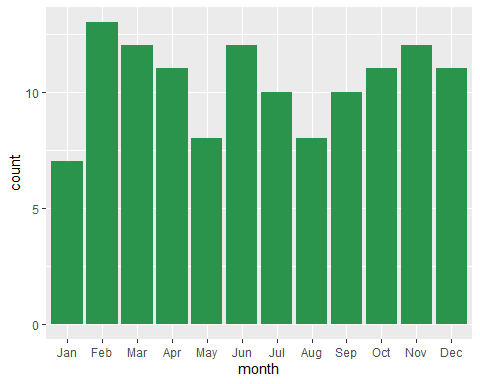
#>50% of white males had a history of prior mental illness since 2000, however >40% also had an unknown history  
  
#hence it would be be unreliable to draw a conclusion from the data, given the unknown portion represents a significant portion of the data.  
  
  
  
#instead lets exclude the unknowns and see of the cases with known mental health history what the proportions are if we assume the unknown data behaviors can be assimilated by the behavior of the known dataset  
  
  
  
Prior\_Mental\_2000\_v2 <- mass\_shootings %>%   
  
 #Filter for year>= 2000, for white males only & exclude NA values in prior mental illness  
  
 filter (year>=2000 & race=='White' & male==TRUE & !is.na(prior\_mental\_illness)) %>%   
  
 mutate( #add an arbitrary weighting column  
  
 n=1,   
  
 #add column calculating the % share each case has to the total based on current equal weighting  
  
 percent=n/sum(n)) %>%  
  
 #group by prior\_mental\_illness  
  
 group\_by(prior\_mental\_illness) %>%   
  
 #summaries data for count of each  
  
 summarize(Count = n(),  
  
 percent = sum(percent))  
  
  
  
Prior\_Mental\_2000\_v2

## # A tibble: 2 × 3  
## prior\_mental\_illness Count percent  
## <chr> <int> <dbl>  
## 1 No 4 0.148  
## 2 Yes 23 0.852

#of cases with a known history of mental illness for white males since 2000, almost 85% were perpetrated by individuals with a history of mental health issues.  
  
#We could extrapolate this relation ship and assume the 85% of cases that have a history of prior mental illness holds  
  
  
  
#overall i believe the prior\_mental\_illness field has too many NA values to produce a strong conclusion on the relationship between cases and mental illness. However, we can clearly see that a large proportion of cases (>50% at the minimum) show a history of mental illness which shows it is a significant factor, but a valid conclusion on the full extent of the relationship cannot be understood due to the limited data.

* 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.

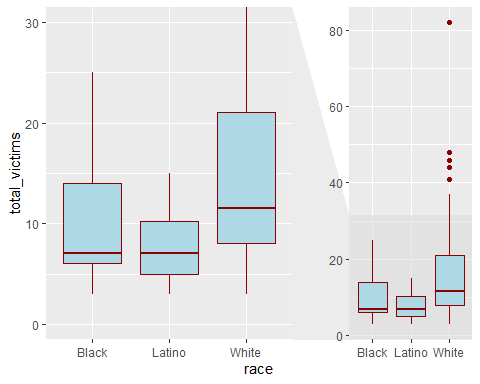
mass\_shootings %>%   
  
 #Filter for any month data missing  
  
 filter(!is.na(month)) %>%   
  
 #Group by month  
  
 group\_by(month) %>%   
  
 #create bar chart  
  
 ggplot(aes(x=month)) +  
  
 geom\_bar(fill="#2A944D") +   
  
 #order chronologically  
  
 scale\_x\_discrete(limits = month.abb)



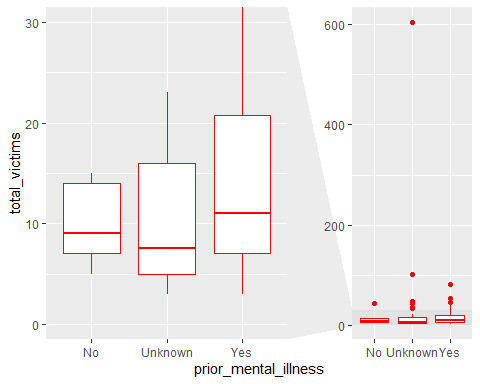
#It can clearly be seen from the bar chart that February is the most dangerous month in term of number of shootings

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

#Create list of races to be compared  
  
R <- c('Latino','White', 'Black')  
  
  
  
#Create a range of bar charts to compare distribution by month  
  
mass\_shootings %>%   
  
 #Filter for any total victims data missing  
  
 #Exclude las vegas strip case as it is a large outlier which will inflate the mean  
  
 filter(!is.na(total\_victims) & race %in% R & !grepl("Las Vegas Strip",case)) %>%   
  
 #create box plot as it is easy to compare IQR and means side by side  
  
 ggplot(aes(x=race,y=total\_victims)) +  
  
 geom\_boxplot(color="darkred",fill="lightblue") +  
  
 facet\_zoom(ylim = c(0, 30))



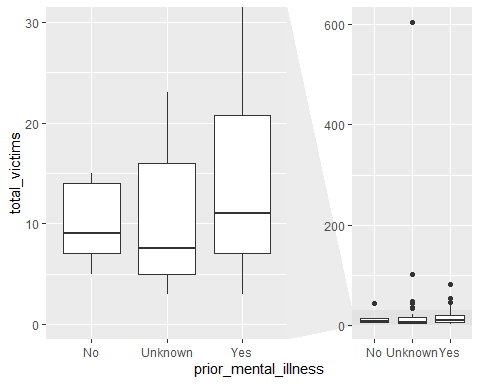
#Black shooters have a lower median and narrower distribution of victims as compared to white shooters  
  
#Latino shooters also have a lower median number of victims than white shooters, and a significantly narrower IQR vs white shooters.  
  
#Overall white shooters appear to have a larger number of victims on average, with mass shootings from white shooters covering a wider range of victim totals as compared to Black/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?  
  
   
  
#utilize data frame created earlier comparing Yes/No mental health issues, excusing cases where mental health history was unknown  
  
  
  
mass\_shootings %>%   
  
 #create unknown category for NA's  
  
 mutate(prior\_mental\_illness= case\_when(!is.na(prior\_mental\_illness) ~ prior\_mental\_illness, TRUE ~ 'Unknown')) %>%   
  
 #group by prior\_mental\_illness  
  
 group\_by(prior\_mental\_illness) %>%   
  
 ggplot(aes(x=prior\_mental\_illness,y=total\_victims)) +  
  
 geom\_boxplot(fill="white",color="red") +  
  
 facet\_zoom(ylim = c(0, 30))



#From the limited data we have on prior mental illness, we can clearly see that shootings with a history of mental illness for the shooter on average have a higher total victim count, and a significantly wider range/IQR in the total number of victims Vs those without a history of mental illness.  
  
#Again however,we can see from the plot of the cases with an unknown history of mental illness that they exhibit a lower median and intermediate IQR when compared to Yes/No, which could skew the conclusion depending on how those unknown data points are catagorised.

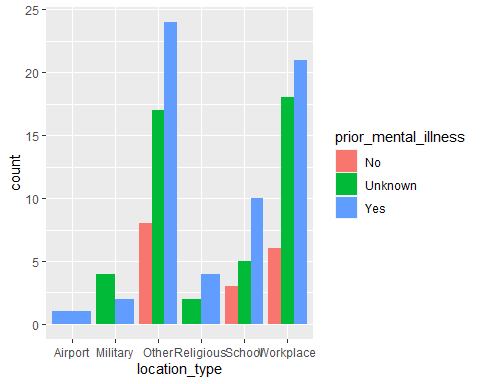
* Assess the relationship between mental illness and total victims, mental illness and location type, and the intersection of all three variables.

#Mental illness and total victims comparison  
  
mass\_shootings %>%   
  
 #create unknown category for NA's  
  
 mutate(prior\_mental\_illness= case\_when(!is.na(prior\_mental\_illness) ~ prior\_mental\_illness, TRUE ~ 'Unknown')) %>%   
  
 #group by prior\_mental\_illness  
  
 group\_by(prior\_mental\_illness) %>%   
  
 ggplot(aes(x=prior\_mental\_illness,y=total\_victims)) +  
  
 geom\_boxplot() +  
  
 facet\_zoom(ylim = c(0, 30))

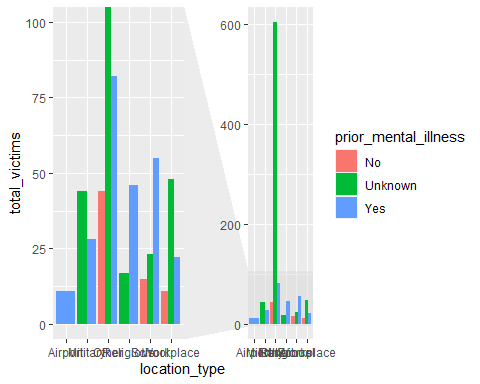


#Explanation from previous  
  
#From the limited data we have on prior mental illness, we can clearly see that shootings with a history of mental illness for the shooter on average have a higher total victim count, and a significantly wider range/IQR in the total number of victims Vs those without a history of mental illness.  
  
#Again however,we can see from the plot of the cases with an unknown history of mental illness that they exhibit a lower median and intermediate IQR when compared to Yes/No, which could skew the conclusion depending on how those unknown data points are catagorised.   
  
  
  
  
  
#Mental illness and location type comparison  
  
mass\_shootings %>%   
  
 #create unknown category for NA's  
  
 mutate(prior\_mental\_illness= case\_when(!is.na(prior\_mental\_illness) ~ prior\_mental\_illness, TRUE ~ 'Unknown')) %>%   
  
 #group by prior\_mental\_illness  
  
 group\_by(location\_type, prior\_mental\_illness) %>%   
  
 #Summarise count for each category  
  
 summarise(count=n()) %>%   
  
 #multiple bar chart  
  
 ggplot(aes(location\_type, count, fill = prior\_mental\_illness)) +  
  
 geom\_bar(stat="identity", position = "dodge")

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



#consistent trend with prior mental illness producing more cases in each location, except in the military where interestingly unknown history of mental illness was the most cases.  
  
  
  
#intersection of mental illness, location, total victims comparison  
  
mass\_shootings %>%   
  
 #create unknown category for NA's  
  
 mutate(prior\_mental\_illness= case\_when(!is.na(prior\_mental\_illness) ~ prior\_mental\_illness, TRUE ~ 'Unknown')) %>%   
  
 #group by prior\_mental\_illness  
  
 group\_by(location\_type, prior\_mental\_illness) %>%   
  
 #multiple bar chart  
  
 ggplot(aes(location\_type, total\_victims, fill = prior\_mental\_illness)) +  
  
 geom\_bar(stat="identity", position = "dodge") +  
  
 facet\_zoom(ylim = c(0, 100))



#a consistent trend across all location with a history of mental illness leading to a higher total victim count Vs no history of mental illness.  
  
#Additionally, cases where prior mental health history is unknown also appear to have a high total victim count which could imply a significant portion of that population may also have a history of mental illness. Although, there is not enough data/information on the data to draw such a conclusion currently.

## 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

#| echo: false  
  
#| message: false  
  
#| warning: false  
  
  
  
card\_fraud <- read\_csv(here::here("data", "card\_fraud.csv"))

## Rows: 671028 Columns: 14  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (4): category, city, state, job  
## dbl (8): trans\_year, amt, lat, long, city\_pop, merch\_lat, merch\_long, is\_fraud  
## dttm (1): trans\_date\_trans\_time  
## date (1): dob  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

glimpse(card\_fraud)

## 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, …

head(card\_fraud)

## # A tibble: 6 × 14  
## 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 Veede… IN 40.1 -87.3  
## 2 2019-02-16 15:07:20 2019 kids\_pets 3.89 Hollo… OH 40.0 -81.0  
## 3 2019-12-27 22:25:34 2019 personal\_care 8.43 Arnold MO 38.4 -90.4  
## 4 2019-03-03 10:11:39 2019 grocery\_net 40 Apison TN 35.0 -85.0  
## 5 2019-02-09 17:14:54 2019 food\_dining 54.0 Red C… CO 39.5 -106.   
## 6 2019-09-09 01:19:59 2019 shopping\_net 95.6 Irwin… GA 32.8 -83.2  
## # ℹ 6 more variables: city\_pop <dbl>, job <chr>, dob <date>, merch\_lat <dbl>,  
## # merch\_long <dbl>, is\_fraud <dbl>

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.

card\_fraud %>%   
  
 #Filter out transactions without a fraud flag  
  
 filter(!is.na(is\_fraud)) %>%   
  
 #Group by transaction year  
  
 group\_by(trans\_year) %>%   
  
 #add a count of transactions and calculate frequency as total count/365 days  
  
 summarise(No\_Frauds = n(),  
  
 Freq\_Fraud\_per\_day = No\_Frauds/365,  
  
 Freq\_per\_mins= No\_Frauds/(365\*24\*60))

## # A tibble: 2 × 4  
## trans\_year No\_Frauds Freq\_Fraud\_per\_day Freq\_per\_mins  
## <dbl> <int> <dbl> <dbl>  
## 1 2019 478646 1311. 0.911  
## 2 2020 192382 527. 0.366

* 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.

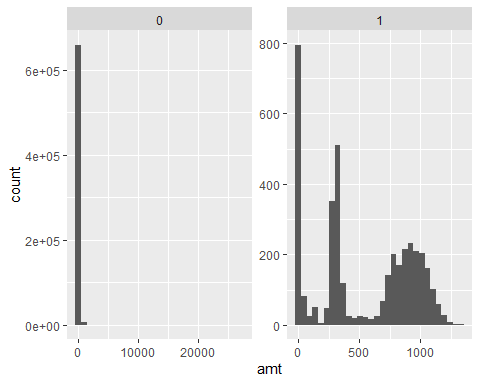
Summary\_One <- card\_fraud %>%   
  
 #Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt)) %>%   
  
 #mutate 2 new columns for fraudulent/none fraudulent transaction amt in USD  
  
 mutate(Amt\_Non\_Fraudulent = case\_when(is\_fraud==1 ~ 0, TRUE ~ amt),  
  
 Amt\_Fraudulent = case\_when(is\_fraud == 0 ~ 0, TRUE ~ amt)) %>%   
  
 #Group by transaction year  
  
 group\_by(trans\_year) %>%   
  
 #summaries total amount for non-fraudulent, fraudulent transactions, and the % share for fraudulent transactions in term of value  
  
 summarise(Total\_Non\_Fraudulent = sum(Amt\_Non\_Fraudulent),  
  
 Total\_Fraudulent = sum(Amt\_Fraudulent),  
  
 percent\_fraud= Total\_Fraudulent/(Total\_Non\_Fraudulent+Total\_Fraudulent))  
  
  
  
Summary\_One

## # A tibble: 2 × 4  
## trans\_year Total\_Non\_Fraudulent Total\_Fraudulent percent\_fraud  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 32182901. 1423140. 0.0423  
## 2 2020 12925914. 651949. 0.0480

* 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.

card\_fraud %>%   
  
 #Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt) ) %>%   
  
 #Group by is fraud  
  
 group\_by(is\_fraud) %>%   
  
 #Create histogram plot  
  
 ggplot(aes(x=amt)) +  
  
 geom\_histogram() +  
  
 facet\_wrap(~is\_fraud, scales = "free")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#Summary statistics  
  
  
  
#Summary for fraudulent  
  
skim(card\_fraud %>%   
  
 #Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt) & is\_fraud==1 ))

Data summary

|  |  |
| --- | --- |
| Name | %>%(…) |
| Number of rows | 3936 |
| Number of columns | 14 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 4 |
| Date | 1 |
| numeric | 8 |
| POSIXct | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| category | 0 | 1 | 4 | 14 | 0 | 14 | 0 |
| city | 0 | 1 | 3 | 25 | 0 | 699 | 0 |
| state | 0 | 1 | 2 | 2 | 0 | 51 | 0 |
| job | 0 | 1 | 3 | 53 | 0 | 441 | 0 |

**Variable type: Date**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| dob | 0 | 1 | 1925-08-29 | 2005-01-29 | 1971-08-20 | 749 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| trans\_year | 0 | 1 | 2019.31 | 0.46 | 2019.00 | 2019.00 | 2019.00 | 2020.00 | 2020.00 | ▇▁▁▁▃ |
| amt | 0 | 1 | 527.21 | 391.29 | 1.06 | 240.49 | 368.83 | 900.94 | 1334.07 | ▇▇▃▇▂ |
| lat | 0 | 1 | 38.65 | 5.10 | 20.03 | 35.06 | 39.43 | 41.84 | 66.69 | ▁▆▇▁▁ |
| long | 0 | 1 | -89.91 | 14.29 | -165.67 | -96.70 | -86.69 | -79.99 | -68.56 | ▁▁▂▆▇ |
| city\_pop | 0 | 1 | 94096.13 | 328508.72 | 23.00 | 741.00 | 2526.00 | 19803.00 | 2906700.00 | ▇▁▁▁▁ |
| merch\_lat | 0 | 1 | 38.64 | 5.16 | 19.53 | 35.12 | 39.42 | 41.92 | 67.44 | ▁▆▇▁▁ |
| merch\_long | 0 | 1 | -89.91 | 14.30 | -166.40 | -96.72 | -86.88 | -79.91 | -67.57 | ▁▁▂▆▇ |
| is\_fraud | 0 | 1 | 1.00 | 0.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | ▁▁▇▁▁ |

**Variable type: POSIXct**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| trans\_date\_trans\_time | 0 | 1 | 2019-01-02 01:06:37 | 2020-06-21 03:59:46 | 2019-09-29 05:54:09 | 3936 |

#Summary for Non-fraudulent  
  
skim(card\_fraud %>%   
  
 #Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt) & is\_fraud==0 ))

Data summary

|  |  |
| --- | --- |
| Name | %>%(…) |
| Number of rows | 667092 |
| Number of columns | 14 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 4 |
| Date | 1 |
| numeric | 8 |
| POSIXct | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| category | 0 | 1 | 4 | 14 | 0 | 14 | 0 |
| city | 0 | 1 | 3 | 25 | 0 | 836 | 0 |
| state | 0 | 1 | 2 | 2 | 0 | 50 | 0 |
| job | 0 | 1 | 3 | 59 | 0 | 475 | 0 |

**Variable type: Date**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| dob | 0 | 1 | 1924-10-30 | 2005-01-29 | 1975-11-30 | 894 |

**Variable type: numeric**

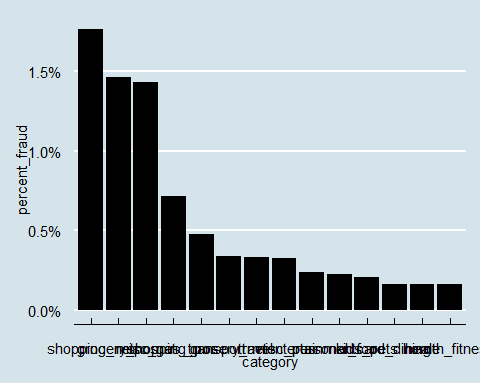
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| trans\_year | 0 | 1 | 2019.29 | 0.45 | 2019.00 | 2019.00 | 2019.00 | 2020.00 | 2020.00 | ▇▁▁▁▃ |
| amt | 0 | 1 | 67.62 | 155.29 | 1.00 | 9.60 | 47.17 | 82.41 | 27119.77 | ▇▁▁▁▁ |
| lat | 0 | 1 | 38.54 | 5.08 | 20.03 | 34.62 | 39.35 | 41.89 | 65.69 | ▁▅▇▁▁ |
| long | 0 | 1 | -90.23 | 13.75 | -165.67 | -96.80 | -87.48 | -80.16 | -67.95 | ▁▁▂▆▇ |
| city\_pop | 0 | 1 | 88875.59 | 302715.77 | 23.00 | 741.00 | 2456.00 | 20328.00 | 2906700.00 | ▇▁▁▁▁ |
| merch\_lat | 0 | 1 | 38.53 | 5.11 | 19.03 | 34.73 | 39.37 | 41.95 | 66.68 | ▁▅▇▁▁ |
| merch\_long | 0 | 1 | -90.23 | 13.77 | -166.67 | -96.90 | -87.44 | -80.23 | -66.95 | ▁▁▂▆▇ |
| is\_fraud | 0 | 1 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | ▁▁▇▁▁ |

**Variable type: POSIXct**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| trans\_date\_trans\_time | 0 | 1 | 2019-01-01 00:00:51 | 2020-06-21 12:12:32 | 2019-10-03 16:40:52 | 661249 |

* 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.

card\_fraud %>%   
  
 #Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt)) %>%   
  
 #Group by transaction year  
  
 group\_by(category ) %>%   
  
 #summaries total amount for non-fraudulent, fraudulent transactions, and the % share for fraudulent transactions in term of value  
  
 summarise(Total\_Fraudulent = sum(is\_fraud),  
  
 Total\_Non\_Fraudulent = length(is\_fraud) -sum(is\_fraud),  
  
 percent\_fraud= Total\_Fraudulent/(Total\_Non\_Fraudulent+Total\_Fraudulent)) %>%   
  
 #Order by most to least likely for fraudulent transaction  
  
 mutate(category =fct\_rev(fct\_reorder(category,percent\_fraud))) %>%   
  
 #bar chart plot  
  
 ggplot(aes(x=category, percent\_fraud)) +  
  
 #plot on a barchart  
  
 geom\_col(fill="black") +  
  
 scale\_y\_continuous(labels = scales::percent\_format()) +  
  
 theme\_economist()



#When is fraud most prevalent - day,month,year  
  
  
  
card\_fraud %>%   
  
#Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt) & is\_fraud==1) %>%   
  
#Separate transaction date into individual fields  
  
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)  
  
 ) %>%   
  
#Group by Day & summaries count  
  
 group\_by(weekday) %>%   
  
 summarise(count=n()) %>%   
  
#arrange descending  
  
 arrange(desc(count))

## # A tibble: 7 × 2  
## weekday count  
## <ord> <int>  
## 1 Mon 639  
## 2 Sat 626  
## 3 Sun 608  
## 4 Fri 557  
## 5 Thu 542  
## 6 Tue 496  
## 7 Wed 468

#monday the day with most fraud cases  
  
  
  
card\_fraud %>%   
  
#Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt) & is\_fraud==1) %>%   
  
#Separate transaction date into individual fields  
  
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)  
  
 ) %>%   
  
#Group by Day & summaries count  
  
 group\_by(month\_name) %>%   
  
 summarise(count=n()) %>%   
  
#arrange descending to see the most fraudulent months  
  
 arrange(desc(count))

## # A tibble: 12 × 2  
## month\_name count  
## <ord> <int>  
## 1 Mar 472  
## 2 May 472  
## 3 Jan 461  
## 4 Feb 434  
## 5 Jun 387  
## 6 Apr 349  
## 7 Dec 301  
## 8 Nov 226  
## 9 Sep 219  
## 10 Oct 218  
## 11 Aug 213  
## 12 Jul 184

#March and may are tied for the most fraudulent months with 472 cases  
  
  
  
card\_fraud %>%   
  
#Filter out transactions without a fraud flag or amt  
  
 filter(!is.na(is\_fraud) & !is.na(amt) & is\_fraud==1) %>%   
  
#Separate transaction date into individual fields  
  
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)  
  
 ) %>%   
  
#Group by Day & summaries count  
  
 group\_by(trans\_year) %>%   
  
 summarise(count=n()) %>%   
  
#arrange descending to see the most fraudulent year  
  
 arrange(desc(count))

## # A tibble: 2 × 2  
## trans\_year count  
## <dbl> <int>  
## 1 2019 2721  
## 2 2020 1215

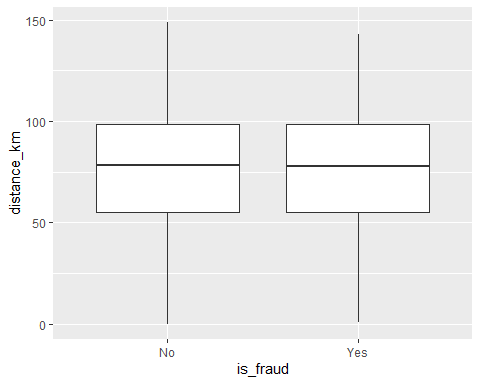
#2019 had the most fraud cases

* Is fraud related to distance? The distance between a card holder’s home and the location of the transaction can be a feature that is related to fraud. To calculate distance, we need the latidue/longitude of card holders’s home and the latitude/longitude of the transaction, and we will use the [Haversine formula](https://en.wikipedia.org/wiki/Haversine_formula) to calculate distance. I adapted code to [calculate distance between two points on earth](https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/) which you can find below

# distance between card holder's home and transaction  
  
# code adapted from https://www.geeksforgeeks.org/program-distance-two-points-earth/amp/  
  
  
  
  
  
card\_fraud <- card\_fraud %>%  
  
 mutate(  
  
   
  
 # convert latitude/longitude to radians  
  
 lat1\_radians = lat / 57.29577951,  
  
 lat2\_radians = merch\_lat / 57.29577951,  
  
 long1\_radians = long / 57.29577951,  
  
 long2\_radians = merch\_long / 57.29577951,  
  
   
  
 # calculate distance in miles  
  
 distance\_miles = 3963.0 \* acos((sin(lat1\_radians) \* sin(lat2\_radians)) + cos(lat1\_radians) \* cos(lat2\_radians) \* cos(long2\_radians - long1\_radians)),  
  
 # calculate distance in km  
  
 distance\_km = 6377.830272 \* acos((sin(lat1\_radians) \* sin(lat2\_radians)) + cos(lat1\_radians) \* cos(lat2\_radians) \* cos(long2\_radians - long1\_radians))  
  
   
  
 )

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

#Explore relationship for distance of fraud by plotting count of fraud cases by distance\_km  
  
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 yes, no  
  
 mutate(is\_fraud= case\_when( !is.na(is\_fraud) & is\_fraud==1 ~ "Yes", TRUE ~ case\_when( !is.na(is\_fraud)~ "No", TRUE ~ "Unknown")))%>%  
  
 #Group by is\_fraud  
  
 group\_by(is\_fraud) %>%   
  
 #create scatter plot to view distribution  
  
 ggplot(aes(x=is\_fraud,y=distance\_km)) +  
  
 geom\_boxplot()



# 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  
  
 )

## Rows: 21890 Columns: 129  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): country, iso\_code  
## dbl (127): year, population, gdp, biofuel\_cons\_change\_pct, biofuel\_cons\_chan...  
##   
## ℹ Use `spec()` to retrieve the full column specification for this data.  
## ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

# 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)  
  
   
  
#glimpse the new dataframes  
  
glimpse(energy)

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

glimpse(gdp\_percap)

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

glimpse(co2\_percap)

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

#Join the tables  
  
energy\_GDP <- left\_join(energy,gdp\_percap,by= c("iso\_code"="iso3c","year"))  
  
energy\_gdp\_co <- left\_join(energy\_GDP,co2\_percap,by= c("iso\_code"="iso3c","year"))  
  
  
  
#view top rows of joined table to ensure columns added  
  
head(energy\_gdp\_co)

## # A tibble: 6 × 28  
## country.x year iso\_code biofuel coal gas hydro nuclear oil  
## <chr> <dbl> <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Afghanistan 1990 AFG NA NA NA NA NA NA  
## 2 Afghanistan 1991 AFG NA NA NA NA NA NA  
## 3 Afghanistan 1992 AFG NA NA NA NA NA NA  
## 4 Afghanistan 1993 AFG NA NA NA NA NA NA  
## 5 Afghanistan 1994 AFG NA NA NA NA NA NA  
## 6 Afghanistan 1995 AFG NA NA NA NA NA NA  
## # ℹ 19 more variables: other\_renewable <dbl>, solar <dbl>, wind <dbl>,  
## # electricity\_demand <dbl>, electricity\_generation <dbl>,  
## # net\_elec\_imports <dbl>, energy\_per\_capita <dbl>, energy\_per\_gdp <dbl>,  
## # per\_capita\_electricity <dbl>, indicator\_id.x <chr>, indicator.x <chr>,  
## # iso2c.x <chr>, country.y <chr>, GDPpercap <dbl>, indicator\_id.y <chr>,  
## # indicator.y <chr>, iso2c.y <chr>, country <chr>, co2percap <dbl>

Specific questions:

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

Utilize Pivot Longer and create an electricity type filed to collapse the multiple columns down to 2 (electricity type and then amount)

1. You may need to join these data frames
   * Use left\_join from dplyr to [join the tables](http://r4ds.had.co.nz/relational-data.html)
   * To complete the merge, you need a unique *key* to match observations between the data frames. Country names may not be consistent among the three data frames, so please use the 3-digit ISO code for each country
2. 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

# Details

* Who did you collaborate with: NA
* Approximately how much time did you spend on this problem set: 4.5H
* What, if anything, gave you the most trouble: Visual editor stopped working partway through, unable to see previews of code run.
* Ran out of time to try the graphs on energy data, initially the code would not work to populate the CO2/GDP data frames (unsure why - EDIT: FOUND THE issue, it was some of the descriptive code used to help with the mutation in one of the questions running on too long). i ended up reopening everything and managed to get it to work,