Homework 2

Nianyu Li

2023-05-22

Table of Contents

[Data Visualisation - Exploration 2](#_Toc135691779)

[Mass shootings in the US 2](#_Toc135691780)

[Obtain the data 2](#_Toc135691781)

[Explore the data 3](#_Toc135691782)

[Specific questions 3](#_Toc135691783)

[More open-ended questions 9](#_Toc135691784)

[Very open-ended 13](#_Toc135691785)

[Exploring credit card fraud 17](#_Toc135691786)

[Obtain the data 17](#_Toc135691787)

[Exploring sources of electricity production, CO2 emissions, and GDP per capita. 30](#_Toc135691788)

[1. A stacked area chart that shows how your own country generated its electricity since 2000. 30](#_Toc135691789)

[2. A scatter plot that looks at how CO2 per capita and GDP per capita are related 30](#_Toc135691790)

[3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related 30](#_Toc135691791)

[1. A stacked area chart that shows how your own country generated its electricity since 2000. 32](#_Toc135691792)

[Deliverables 34](#_Toc135691793)

[Details 34](#_Toc135691794)

[Rubric 34](#_Toc135691795)

# 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.

mass\_shootings %>%   
 group\_by(year) %>%   
 summarize(count = n())

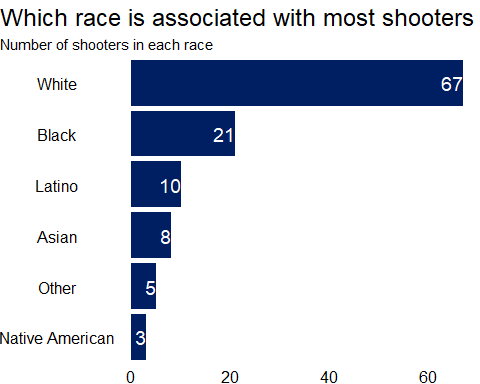
## # A tibble: 37 × 2  
## year count  
## <dbl> <int>  
## 1 1982 1  
## 2 1984 2  
## 3 1986 1  
## 4 1987 1  
## 5 1988 1  
## 6 1989 2  
## 7 1990 1  
## 8 1991 3  
## 9 1992 2  
## 10 1993 4  
## # … with 27 more rows

* 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 table that shows the number of shooters in each race category  
shooters\_race <- mass\_shootings %>%  
 #Remove the NA in race  
 filter(!is.na(race)) %>%   
 group\_by(race) %>%   
 summarize(count = n())  
  
print(shooters\_race)

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

#Plot the table above.  
shooters\_race %>%   
   
 #sort the bars from highest to lowest  
 mutate(race = fct\_reorder(race, count)) %>%   
   
 ggplot()+  
   
 aes(x = count,  
 y = race) +  
   
 geom\_col(fill = "#001E62") +  
   
 #add labels to show # of shooters on each bar  
 geom\_text(aes(label = count),  
 hjust = 1,  
 color = "#FFFFFF",  
 size = 5) +  
   
 #add titles  
 labs(title = "Which race is associated with most shooters",  
 subtitle = "Number of shooters in each race",  
 x = "Number of shooter",  
 y = NULL) +  
   
 theme\_void() +  
 #adjust the positions and sizes  
 theme(plot.title.position = "plot",  
 axis.text = element\_text(size=12),  
 plot.title = element\_text(size=18))+  
   
 NULL



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

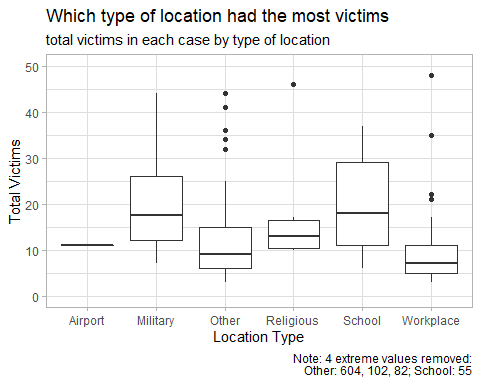
#Create a table that shows the number of total victims in each case by type of location  
victims\_location\_type <-   
 mass\_shootings %>%  
 select(location\_type,  
 total\_victims)  
  
print(victims\_location\_type)

## # A tibble: 125 × 2  
## location\_type total\_victims  
## <chr> <dbl>  
## 1 School 11  
## 2 Workplace 9  
## 3 Workplace 15  
## 4 Workplace 5  
## 5 Workplace 10  
## 6 Workplace 9  
## 7 Workplace 4  
## 8 Workplace 5  
## 9 Other 7  
## 10 Military 11  
## # … with 115 more rows

#Note that 95% of the data is below 44:  
quantile(mass\_shootings$total\_victims, probs = .95)

## 95%   
## 44

#Plot the table above in a box plot  
victims\_location\_type %>%   
   
 ggplot()+  
   
 aes(x = location\_type,  
 y = total\_victims)+  
   
 #Create the box plot and remove outliers  
 geom\_boxplot()+  
 scale\_y\_continuous(limits = c(0,50))+  
   
 #add titles  
 labs(title = "Which type of location had the most victims",  
 subtitle = "total victims in each case by type of location",  
 x = "Location Type",  
 y = "Total Victims",  
 caption = "Note: 4 extreme values removed:  
 Other: 604, 102, 82; School: 55") +  
   
 theme\_light()+  
   
 NULL



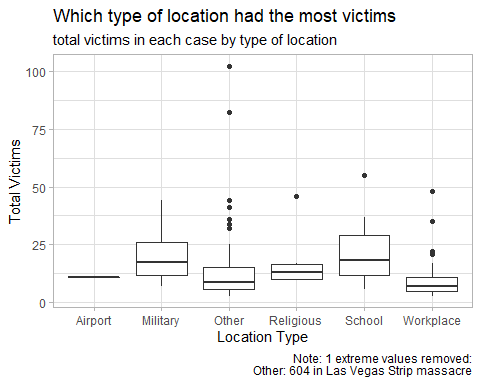
* Redraw the same plot, but remove the Las Vegas Strip massacre from the dataset.

It can be done using the method in the question above, i.e.setting a limit on y axis. We can also filter out the outlier in the first place.

victims\_location\_type\_1 <-   
 mass\_shootings %>%  
 select(location\_type,  
 total\_victims) %>%   
 filter(total\_victims!=604)  
  
print(victims\_location\_type\_1)

## # A tibble: 124 × 2  
## location\_type total\_victims  
## <chr> <dbl>  
## 1 School 11  
## 2 Workplace 9  
## 3 Workplace 15  
## 4 Workplace 5  
## 5 Workplace 10  
## 6 Workplace 9  
## 7 Workplace 4  
## 8 Workplace 5  
## 9 Other 7  
## 10 Military 11  
## # … with 114 more rows

#Plot the table above in a box plot  
victims\_location\_type\_1 %>%   
   
 ggplot()+  
   
 aes(x = location\_type,  
 y = total\_victims)+  
   
 geom\_boxplot()+  
   
 #add titles  
 labs(title = "Which type of location had the most victims",  
 subtitle = "total victims in each case by type of location",  
 x = "Location Type",  
 y = "Total Victims",  
 caption = "Note: 1 extreme values removed:  
 Other: 604 in Las Vegas Strip massacre") +  
   
 theme\_light()+  
   
 NULL



### More open-ended questions

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

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

#Filter out the data using the criteria above  
mass\_shootings %>%   
 filter(race == "White",  
 male == TRUE,  
 prior\_mental\_illness == "Yes",  
 year > 2000) %>%   
 count()

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

#Filtered data shows that there are 22 of them.

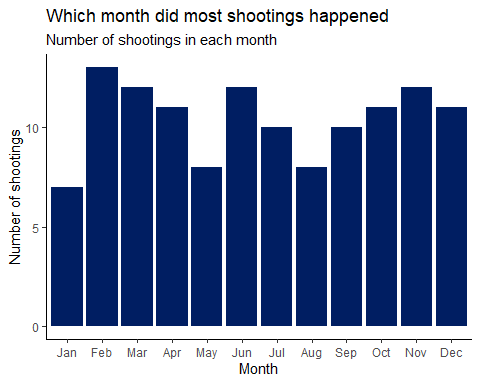
* Which month of the year has the most mass shootings? Generate a bar chart sorted in chronological (natural) order (Jan-Feb-Mar- etc) to provide evidence of your answer.

Feburary has the most mass shootings. See bar chart below.

#Create a table to show mass shootings in each month of the year  
shootings\_by\_month <-   
 mass\_shootings %>%  
   
 #convert month to a factor  
 mutate(month = factor(month, levels = month.abb)) %>%   
   
 #group by month and count # of shootings in each month  
 group\_by(month) %>%   
 summarise(number = n())  
  
print(shootings\_by\_month)

## # A tibble: 12 × 2  
## month number  
## <fct> <int>  
## 1 Jan 7  
## 2 Feb 13  
## 3 Mar 12  
## 4 Apr 11  
## 5 May 8  
## 6 Jun 12  
## 7 Jul 10  
## 8 Aug 8  
## 9 Sep 10  
## 10 Oct 11  
## 11 Nov 12  
## 12 Dec 11

#Plot the table  
  
shootings\_by\_month %>%   
   
 ggplot()+  
   
 aes(x = month,  
 y = number)+  
   
 geom\_bar(stat = "identity",  
 fill = "#001E62")+  
   
 #add titles  
 labs(title = "Which month did most shootings happened",  
 subtitle = "Number of shootings in each month",  
 x = "Month",  
 y = "Number of shootings")+  
   
 theme\_classic()+  
   
 NULL



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

Black and white shooter: in most cases, fatalities caused by black shooters are smaller than that of white shooters. As the density plot indicates, fatalities by black shooters are concentrated on the left hand side of fatalities by white shooters. In the white shooters case, the density plot has a longer tail, which means fatalities are more disperse.

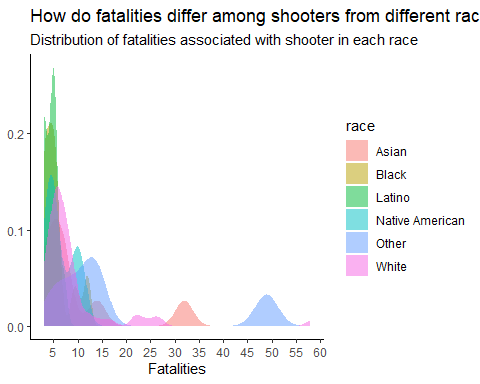
White and Latino shooters: in most cases, fatalities caused by latino shooters are also smaller than that of white shooters. They are also more concentrated on the left side of the plot, compared to that caused by white shooters.

See density plot below.

#Create a table to show shootings fatalities by shooters race  
fatalities\_race <-   
 mass\_shootings %>%  
 #filter out data that is not NA in race  
 filter(!is.na(race)) %>%  
 select(fatalities, race)  
  
print(fatalities\_race)

## # A tibble: 114 × 2  
## fatalities race   
## <dbl> <chr>  
## 1 8 White  
## 2 8 White  
## 3 5 Black  
## 4 4 Black  
## 5 7 White  
## 6 9 White  
## 7 22 White  
## 8 12 Black  
## 9 5 Black  
## 10 3 White  
## # … with 104 more rows

#Plot the distribution of fatalities, using geom\_density  
fatalities\_race %>%   
   
 ggplot()+  
   
 aes(x = fatalities,  
 fill = race)+  
   
 geom\_density(color = NA,alpha = 0.5)+  
   
 scale\_x\_continuous(breaks = seq(0,60,5))+  
   
 theme\_classic()+  
   
 #add titles  
 labs(title = "How do fatalities differ among shooters from different races",  
 subtitle = "Distribution of fatalities associated with shooter in each race",  
 x = "Fatalities",  
 y = NULL)+  
   
 NULL



### 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?

Shootings were more likely to be conducted by shooters with mental illness. Location types are also different. Shooters with mental illness have a wider choices of locations and they can shoot in public spaces like airports and religious places.

See pie chart below for percentages of type of locations for shooters with and without mental illness.

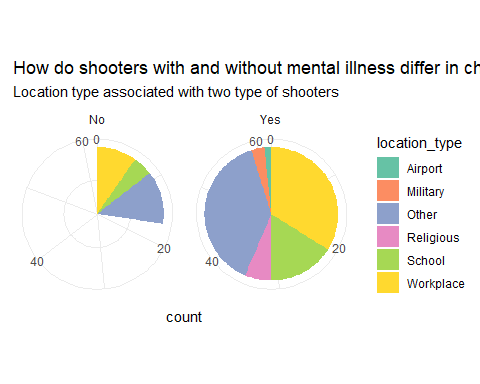
#Create a table to show percent of type of locations for the two type of shooters.  
shooters\_mental\_location <-   
 mass\_shootings %>%   
 #filter out NAs  
 filter(!is.na(prior\_mental\_illness)) %>%   
 group\_by(prior\_mental\_illness,  
 location\_type) %>%   
 summarize(count = n())

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

print(shooters\_mental\_location)

## # A tibble: 9 × 3  
## # Groups: prior\_mental\_illness [2]  
## prior\_mental\_illness location\_type count  
## <chr> <chr> <int>  
## 1 No Other 8  
## 2 No School 3  
## 3 No Workplace 6  
## 4 Yes Airport 1  
## 5 Yes Military 2  
## 6 Yes Other 24  
## 7 Yes Religious 4  
## 8 Yes School 10  
## 9 Yes Workplace 21

#Create a pie chart to show differences in location type by the two type of shooters  
  
shooters\_mental\_location %>%   
   
 ggplot()+  
   
 aes(x="",  
 y = count,  
 fill = location\_type)+  
   
 geom\_bar(stat="identity", width=1)+  
 coord\_polar("y", start=0)+  
   
 facet\_wrap(~prior\_mental\_illness)+  
   
 scale\_fill\_brewer(palette="Set2")+  
   
 theme\_minimal()+  
   
 #add titles  
 labs(title = "How do shooters with and without mental illness differ in choices of locations",  
 subtitle = "Location type associated with two type of shooters",  
 x = NULL)+  
   
 NULL



* 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: shooters with mental illness were causing more victims than those without.

Mental illness and location type: shooters with mental illness chose locations more randomly, while those without mainly focused on others, schools and workplaces.

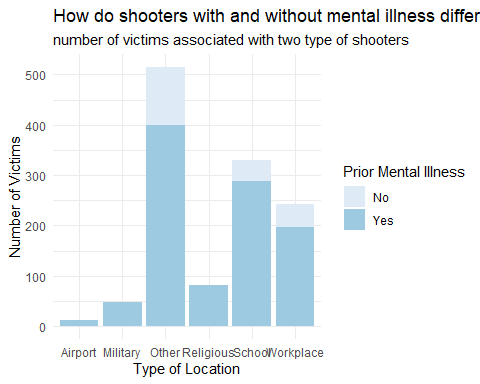
Intersection: shooters with mental illness shooting at school and workplaces were causing most of the victims.

See bar plots below.

#Create a table that shows all three variables above  
mental\_victims\_location <-   
 mass\_shootings %>%   
 #filter out NAs  
 filter(!is.na(prior\_mental\_illness)) %>%   
 select(prior\_mental\_illness,  
 total\_victims,  
 location\_type)  
  
print(mental\_victims\_location)

## # A tibble: 79 × 3  
## prior\_mental\_illness total\_victims location\_type  
## <chr> <dbl> <chr>   
## 1 Yes 9 Workplace   
## 2 Yes 15 Workplace   
## 3 Yes 10 Workplace   
## 4 Yes 32 Other   
## 5 Yes 11 Workplace   
## 6 Yes 5 Workplace   
## 7 Yes 34 Other   
## 8 Yes 5 Workplace   
## 9 Yes 8 Other   
## 10 Yes 3 Workplace   
## # … with 69 more rows

#Bar plots  
mental\_victims\_location %>%   
   
 ggplot()+  
   
 aes(x = location\_type,  
 y = total\_victims,  
 fill = prior\_mental\_illness)+  
   
 geom\_bar(stat = "identity")+  
   
 scale\_fill\_brewer(palette="Blues",  
 name ="Prior Mental Illness")+  
   
 theme\_minimal()+  
   
 #add titles  
 labs(title = "How do shooters with and without mental illness differ in number of victims",  
 subtitle = "number of victims associated with two type of shooters",  
 x = "Type of Location",  
 y = "Number of Victims")+  
   
 NULL



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.

In 2019, there were 2721 fraudulent transactions, occurring at a frequency of 0.57%. In 2020, there were 1215 fraudulent transactions, occurring at a frequency of 0.63%.

fraud\_per\_year <-   
 card\_fraud %>%   
 group\_by(trans\_year) %>%   
 summarise(number = sum(is\_fraud),  
 frequency\_in\_percent = round(number/n()\*100,2)) %>%   
 ungroup()  
  
print(fraud\_per\_year)

## # A tibble: 2 × 3  
## trans\_year number frequency\_in\_percent  
## <dbl> <dbl> <dbl>  
## 1 2019 2721 0.57  
## 2 2020 1215 0.63

* 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.

See table for totl amount of legitimate and fraudulent transactions per year. Calculate % of fraudulent transactions percent\_fraud\_amount\_2019:4.23% percent\_fraud\_amount\_2020:4.8%

#Create the table to show the amount of legitimate and fraudulent transactions per year  
amount\_fraud\_per\_year <-   
 card\_fraud %>%   
 group\_by(trans\_year, is\_fraud) %>%   
 summarize(amount = sum(amt)) %>%   
 ungroup()

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

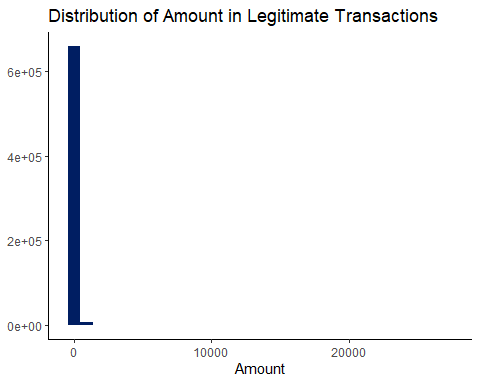
print(amount\_fraud\_per\_year)

## # A tibble: 4 × 3  
## trans\_year is\_fraud amount  
## <dbl> <dbl> <dbl>  
## 1 2019 0 32182901.  
## 2 2019 1 1423140.  
## 3 2020 0 12925914.  
## 4 2020 1 651949.

* 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.

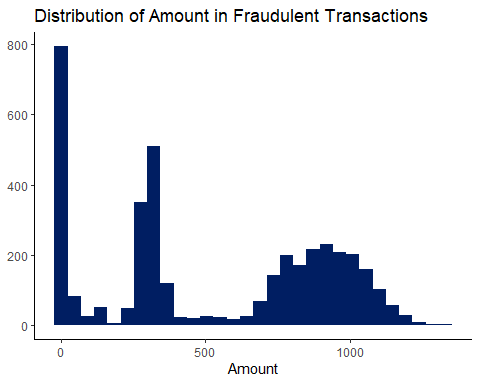
#Create a table to show amounts in each transaction  
amount\_charged\_legit <-   
 card\_fraud %>%  
 filter(is\_fraud == 0) %>%   
 select(amt)  
  
amount\_charged\_fraud <-   
 card\_fraud %>%  
 filter(is\_fraud == 1) %>%   
 select(amt)  
  
#Plot the distribution for legit transactions  
amount\_charged\_legit %>%   
   
 ggplot()+  
   
 aes(x = amt)+  
   
 geom\_histogram(fill = "#001E62")+  
   
 theme\_classic()+  
   
 #add titles  
 labs(title = "Distribution of Amount in Legitimate Transactions",  
 x = "Amount",  
 y = NULL)+  
   
 NULL

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



#Plot the distribution for fraud transactions  
amount\_charged\_fraud %>%   
   
 ggplot()+  
   
 aes(x = amt)+  
   
 geom\_histogram(fill = "#001E62")+  
   
 theme\_classic()+  
   
 #add titles  
 labs(title = "Distribution of Amount in Fraudulent Transactions",  
 x = "Amount",  
 y = NULL)+  
   
 NULL

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



#Summary of statistics  
summary(amount\_charged\_legit)

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

summary(amount\_charged\_fraud)

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

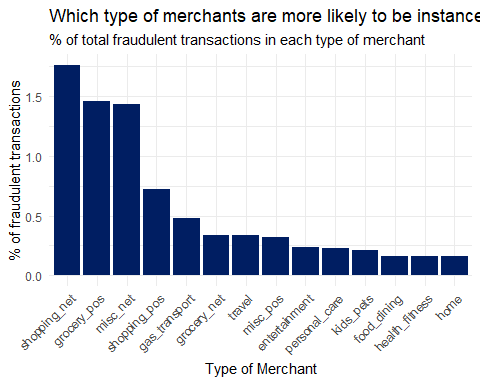
* What types of purchases are most likely to be instances of fraud? Consider category of merchants and produce a bar chart that shows % of total fraudulent transactions sorted in order.

#Create a table to show # of transactions by merchants.  
merchant\_type <-   
 card\_fraud %>%  
 group\_by(category, is\_fraud) %>%  
   
 #Calculate the percent of fraudulent transactions in each category  
 summarise(count = n()) %>%   
 mutate(pct = round(count/sum(count)\*100, 2)) %>%  
   
 #Remove unnecessary data  
 filter(is\_fraud == 1) %>%   
 select(category, pct) %>%  
   
 #Sort in descending order by pct  
 arrange(desc(pct)) %>%   
 print()

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

## # A tibble: 14 × 2  
## # Groups: category [14]  
## category pct  
## <chr> <dbl>  
## 1 shopping\_net 1.76  
## 2 grocery\_pos 1.46  
## 3 misc\_net 1.43  
## 4 shopping\_pos 0.72  
## 5 gas\_transport 0.48  
## 6 grocery\_net 0.34  
## 7 travel 0.34  
## 8 misc\_pos 0.32  
## 9 entertainment 0.24  
## 10 personal\_care 0.23  
## 11 kids\_pets 0.21  
## 12 food\_dining 0.16  
## 13 health\_fitness 0.16  
## 14 home 0.16

#Bar chart  
merchant\_type %>%   
   
 ggplot()+  
   
 aes(x= reorder(category,-pct),  
 y = pct)+  
   
 geom\_bar(stat = "identity",  
 fill = "#001E62")+  
   
 theme\_minimal()+  
 theme(axis.text.x = element\_text(angle = 45, vjust = 1, hjust=1))+  
   
 #add titles  
 labs(title = "Which type of merchants are more likely to be instances of fraud",  
 subtitle = "% of total fraudulent transactions in each type of merchant",  
 x = "Type of Merchant",  
 y = "% of fraudulent transactions")+  
   
 NULL



* 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

We can tell from the table below that fraud is most prevalent: 1) at 23:00 2) on 2019-02-25 3) in March 4) on Mondays

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)  
 )

#Create new variables  
time\_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)  
 )  
  
#See which hours are frauds most prevalent  
time\_fraud %>%   
 group\_by(hour) %>%   
 summarise(count = sum(is\_fraud)) %>%   
 arrange(desc(count)) %>%   
 print()

## # A tibble: 24 × 2  
## hour count  
## <int> <dbl>  
## 1 23 1012  
## 2 22 981  
## 3 0 348  
## 4 1 332  
## 5 3 326  
## 6 2 313  
## 7 19 52  
## 8 18 49  
## 9 17 48  
## 10 13 45  
## # … with 14 more rows

#See which days are frauds most prevalent  
time\_fraud %>%   
 group\_by(date\_only) %>%   
 summarise(count = sum(is\_fraud)) %>%   
 arrange(desc(count)) %>%   
 print()

## # A tibble: 537 × 2  
## date\_only count  
## <date> <dbl>  
## 1 2019-02-25 30  
## 2 2019-02-12 28  
## 3 2019-08-31 27  
## 4 2019-05-20 26  
## 5 2019-10-20 25  
## 6 2019-12-12 25  
## 7 2020-06-17 25  
## 8 2019-01-14 24  
## 9 2019-09-11 24  
## 10 2020-03-17 23  
## # … with 527 more rows

#See which month are frauds most prevalent  
time\_fraud %>%   
 group\_by(month\_name) %>%   
 summarise(count = sum(is\_fraud)) %>%   
 arrange(desc(count)) %>%   
 print()

## # A tibble: 12 × 2  
## month\_name count  
## <ord> <dbl>  
## 1 3月 472  
## 2 5月 472  
## 3 1月 461  
## 4 2月 434  
## 5 6月 387  
## 6 4月 349  
## 7 12月 301  
## 8 11月 226  
## 9 9月 219  
## 10 10月 218  
## 11 8月 213  
## 12 7月 184

#See which weekdays are frauds most prevalent  
time\_fraud %>%   
 group\_by(weekday) %>%   
 summarise(count = sum(is\_fraud)) %>%   
 arrange(desc(count)) %>%   
 print()

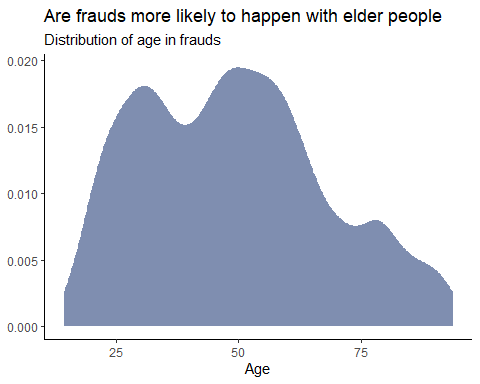
## # A tibble: 7 × 2  
## weekday count  
## <ord> <dbl>  
## 1 周一 639  
## 2 周六 626  
## 3 周日 608  
## 4 周五 557  
## 5 周四 542  
## 6 周二 496  
## 7 周三 468

* 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

From the density plot, we can tell that it is tilted towards the left hand side, which means that young people are actually more likely to be victims of frauds.

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

age\_fraud <-   
 card\_fraud %>%  
 #Create new variable that shows customer age  
 mutate(  
 age = interval(dob, trans\_date\_trans\_time) / years(1),  
 ) %>%   
 filter(is\_fraud==1)  
   
#See number of fraud in different ages  
age\_fraud %>%   
   
 ggplot()+  
   
 aes(x = age)+  
   
 geom\_density(color = NA, fill = "#001E62", alpha = 0.5)+  
   
 theme\_classic()+  
   
 #add titles  
 labs(title = "Are frauds more likely to happen with elder people",  
 subtitle = "Distribution of age in frauds",  
 x = "Age",  
 y = NULL)+  
   
 NULL



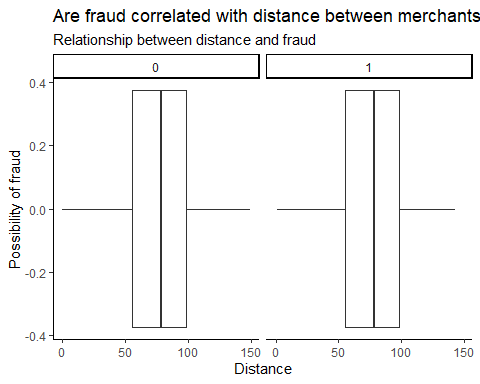
* Is fraud related to distance? The distance between a card holder’s home and the location of the transaction can be a feature that is related to fraud. To calculate distance, we need the latidue/longitude of card holders’s home and the latitude/longitude of the transaction, and we will use the [Haversine 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?

It turned out that two box plots looks more or less the same. Therefore, distance seems not a good explanatory feature for fraud.

#Create boxplot  
card\_fraud %>%   
   
 ggplot()+  
 aes(x = distance\_km)+  
   
 geom\_boxplot()+  
   
 facet\_wrap(~is\_fraud)+  
   
 theme\_classic()+  
   
 #add titles  
 labs(title = "Are fraud correlated with distance between merchants and buyers",  
 subtitle = "Relationship between distance and fraud",  
 x = "Distance",  
 y = "Possibility of fraud")+  
   
 NULL



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

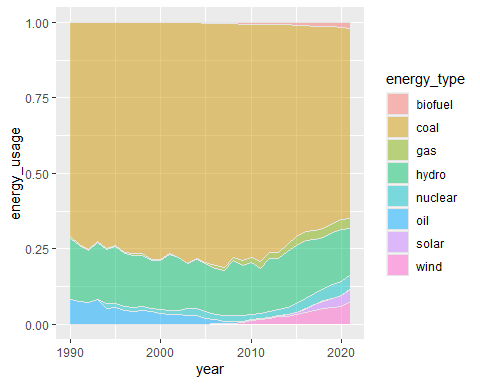
You will use

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

energy\_china <- energy %>%   
   
 #Filter data for China  
 filter(country == "China") %>%   
   
 #select lines we need  
 select(year,   
 biofuel, coal, gas,   
 hydro, nuclear, oil,  
 other\_renewable, solar, wind) %>%   
   
 #transform the table in to long format  
 pivot\_longer(cols = c("biofuel", "coal", "gas",   
 "hydro", "nuclear", "oil",  
 "other\_renewable", "solar", "wind"),  
 names\_to="energy\_type",  
 values\_to = "energy\_usage")  
  
print(energy\_china)

## # A tibble: 288 × 3  
## year energy\_type energy\_usage  
## <dbl> <chr> <dbl>  
## 1 1990 biofuel NA   
## 2 1990 coal 441.   
## 3 1990 gas 2.77   
## 4 1990 hydro 127.   
## 5 1990 nuclear 0   
## 6 1990 oil 50.4   
## 7 1990 other\_renewable NA   
## 8 1990 solar 0.002  
## 9 1990 wind 0.002  
## 10 1991 biofuel NA   
## # … with 278 more rows

#Create the stacked area chart  
energy\_china %>%   
   
 ggplot()+  
   
 aes(x = year,  
 y = energy\_usage,  
 fill = energy\_type)+  
   
 geom\_area(colour="grey90", alpha = 0.5, position = "fill")



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 Kingdom? Great Britain?). countrycode() takes as an input a country’s name in a specific format and outputs it using whatever format you specify.
3. Write a function that takes as input any country’s name and returns all three graphs. You can use the patchwork package to arrange the three graphs as shown below

# Deliverables

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

# Details

* Who did you collaborate with: I worked on myself.
* Approximately how much time did you spend on this problem set: 8hrs
* What, if anything, gave you the most trouble: Not enough time.

**Please seek out help when you need it,** and remember the [15-minute rule](https://dsb2023.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?

Yes!

# 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.