Homework 2

Jacqui Cole

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

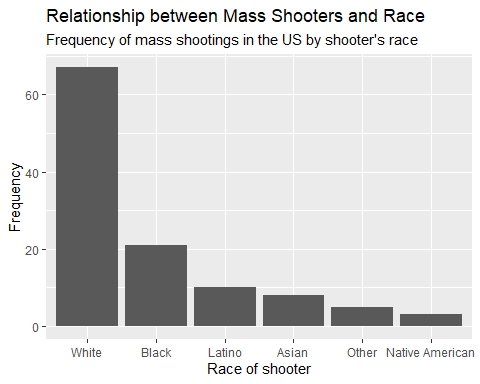
### Specific questions

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

#creating a dataframe of the frequency (n) of mass shootings per year and saving it as msperyear.  
df\_msperyear <- mass\_shootings %>%  
count(year)

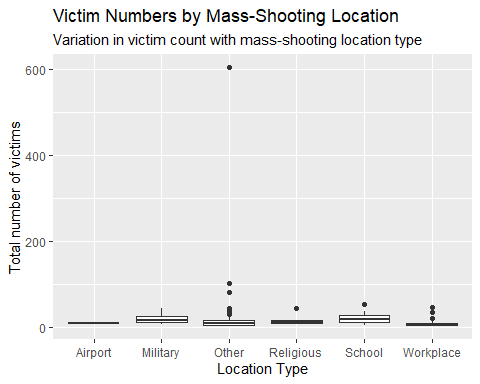
* Generate a bar chart that identifies the number of mass shooters associated with each race category. The bars should be sorted from highest to lowest and each bar should show its number.

#Read in data and filter out entries for race that are not applicable (NA)  
mass\_shootings %>%  
 filter(!is.na(race)) %>%  
#Produce a frequency metric, n, for the number of times a mass shooter is of a specific race  
 count(race) %>%  
#Plot the result, ordering the race from highest to lowest frequency of occurrence.  
ggplot(mapping = aes(reorder(race, -n, sum),n)) + geom\_col() +labs(title = "Relationship between Mass Shooters and Race", subtitle = "Frequency of mass shootings in the US by shooter's race", x = "Race of shooter", y = "Frequency")



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

# I produce a boxplot to visualise the number of total victims of mass shootings in the US by type of location. The code and plot are given below.  
 ggplot(data = mass\_shootings, mapping = aes(location\_type,total\_victims)) + geom\_boxplot() + labs(x = "Location Type", y = "Total number of victims", title = "Victim Numbers by Mass-Shooting Location", subtitle = "Variation in victim count with mass-shooting location type")

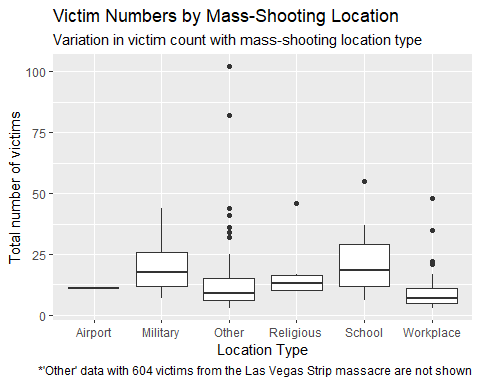


* Redraw the same plot, but remove the Las Vegas Strip massacre from the dataset.

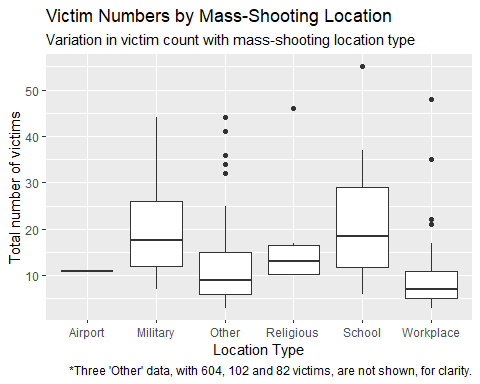
#I note that there is a extreme data outlier in the location\_type 'other' category and that its particularly high total number of victims skews the presentation of the plot. There are also two other much larger data points in other. In order that the rest of the data show up in good proportioning, I identify these three data points in order to consider removing them and then replot the trend:  
 mass\_shootings %>%  
 count(total\_victims) %>%  
 arrange(desc(total\_victims))

# A tibble: 38 × 2  
 total\_victims n  
 <dbl> <int>  
 1 604 1  
 2 102 1  
 3 82 1  
 4 55 1  
 5 48 1  
 6 46 1  
 7 44 2  
 8 41 1  
 9 37 1  
10 36 1  
# ℹ 28 more rows

#This produces a table where the three highest values can be easily identified. I first filter out the largest data outlier and replot the trend:  
   
 mass\_shootings %>%  
 filter(total\_victims != 604) %>%  
 ggplot(mapping = aes(location\_type,total\_victims)) + geom\_boxplot() + labs(x = "Location Type", y = "Total number of victims", title = "Victim Numbers by Mass-Shooting Location", subtitle = "Variation in victim count with mass-shooting location type", caption = "\*'Other' data with 604 victims from the Las Vegas Strip massacre are not shown")



#I then filter out the other two large 'Other' data outliers and then replot the trend in order to obtain even better data proportioning of the results:  
 mass\_shootings %>%  
 filter(total\_victims != 604) %>%  
 filter(total\_victims != 102) %>%  
 filter(total\_victims != 82) %>%  
 ggplot(mapping = aes(location\_type,total\_victims)) + geom\_boxplot() + labs(x = "Location Type", y = "Total number of victims", title = "Victim Numbers by Mass-Shooting Location", subtitle = "Variation in victim count with mass-shooting location type", caption = "\*Three 'Other' data, with 604, 102 and 82 victims, are not shown, for clarity.")



### 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?

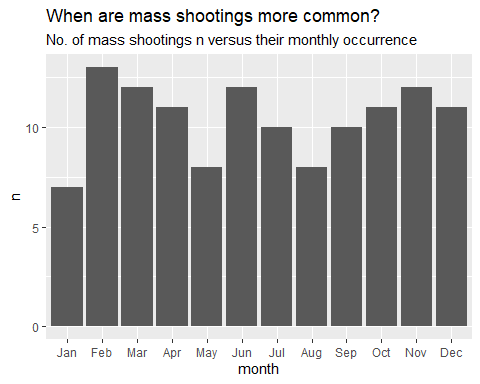
# There are 125 entries of mass shootings in this database. I filter these down to those that occurred after 2000, where the shooter is white and had shown signs of prior mental illness. This produces a tibble of 23 entries.  
  
mass\_shootings %>%  
 filter(year > 2000, race == "White", prior\_mental\_illness == "Yes")

# A tibble: 23 × 14  
 case year month day location summary fatalities injured total\_victims  
 <chr> <dbl> <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
 1 FedEx wa… 2021 Apr 15 Indiana… "Brand… 8 7 15  
 2 Odessa-M… 2019 Aug 31 Odessa,… "Seth … 7 25 32  
 3 SunTrust… 2019 Jan 23 Sebring… "Zephe… 5 0 5  
 4 Waffle H… 2018 Apr 22 Nashvil… "Travi… 4 4 8  
 5 Marjory … 2018 Feb 14 Parklan… "Nikol… 17 17 34  
 6 Texas Fi… 2017 Nov 5 Sutherl… "Devin… 26 20 46  
 7 Rural Oh… 2017 May 12 Kirkers… "Thoma… 3 0 3  
 8 Isla Vis… 2014 May 23 Santa B… "Ellio… 6 13 19  
 9 Santa Mo… 2013 Jun 7 Santa M… "John … 6 3 9  
10 Sandy Ho… 2012 Dec 14 Newtown… "Adam … 27 2 29  
# ℹ 13 more rows  
# ℹ 5 more variables: location\_type <chr>, male <lgl>, age\_of\_shooter <dbl>,  
# race <chr>, prior\_mental\_illness <chr>

#i.e. There are 23 white males with prior signs of mental illness initiated a mass shooting after 2000.

* 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 %>%  
 count(month) %>%  
 ggplot(mapping = aes(x = month, y = n)) + geom\_col() + scale\_x\_discrete(limits = month.abb) + labs(title = "When are mass shootings more common?", subtitle = "No. of mass shootings n versus their monthly occurrence")

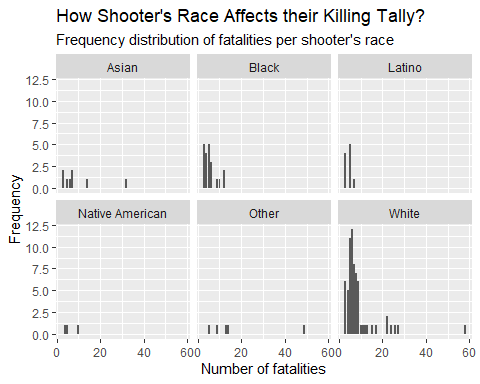


#This bar plot reveals that mass shootings in the US most commonly occur during the month of February. I now produce a data fram that shows the explicit number that occur per month (revealing that this is 13 in February):  
  
monthlyvariation <- mass\_shootings %>%  
 count(month) %>%  
 arrange(desc(n))   
monthlyvariation

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

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

mass\_shootings %>%  
 filter (!is.na(race)) %>%  
 count(fatalities,race) %>%  
 group\_by(fatalities,n,race) %>%  
 ggplot() + (aes(x = fatalities, y = n)) + geom\_col() + facet\_wrap(vars(race)) + labs(title = "How Shooter's Race Affects their Killing Tally?", subtitle = "Frequency distribution of fatalities per shooter's race", x = "Number of fatalities", y = "Frequency")



#This set of plots reveal that white, black and Latino shooters track a similar killing distribution profile. There are many fewer cases of black and Latino shooters, but the max point in the distribution appears to be similar between all three cases while the distribution for white shooters seems to have a longer tail at the upper boundary; this suggests that a small proportion of white shooters may shoot many more than those from another race (although it is difficult to make any concrete conclusions as the sample is small). We can calculate the mean to assist although these values not surprisingly vary because of the small samples involved (so we cannot regard these as representative except for the overall mean and the mean for white shooters - which are similar):  
  
#Determining the overall mean (= 8)  
  
mass\_shootings %>%  
 filter(!is.na(fatalities)) %>%  
 summarise(meanfat = mean(fatalities))

# A tibble: 1 × 1  
 meanfat  
 <dbl>  
1 8

#Determining the mean for white shooters (= 8.8)  
  
mass\_shootings %>%  
 filter(!is.na(fatalities)) %>%  
 filter(race == "White") %>%  
 summarise(meanfat = mean(fatalities))

# A tibble: 1 × 1  
 meanfat  
 <dbl>  
1 8.78

#Determining the mean for black shooters (= 5.6)  
mass\_shootings %>%  
 filter(!is.na(fatalities)) %>%  
 filter(race == "Black") %>%  
 summarise(meanfat = mean(fatalities))

# A tibble: 1 × 1  
 meanfat  
 <dbl>  
1 5.57

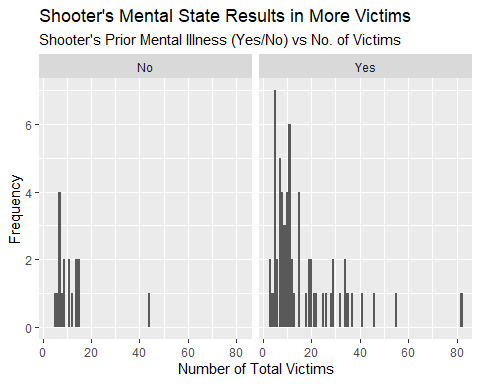
#Determining the mean for latino shooters (= 4.4)  
mass\_shootings %>%  
 filter(!is.na(fatalities)) %>%  
 filter(race == "Latino") %>%  
 summarise(meanfat = mean(fatalities))

# A tibble: 1 × 1  
 meanfat  
 <dbl>  
1 4.4

### 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?

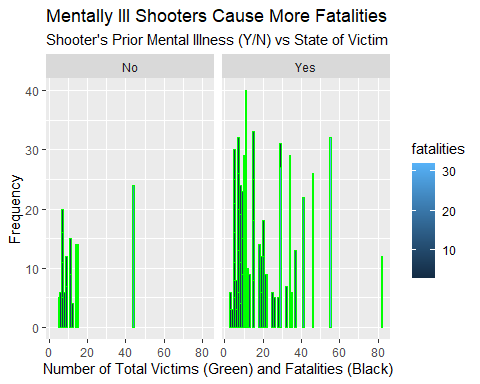
mass\_shootings %>%  
 filter (!is.na(prior\_mental\_illness)) %>%  
 group\_by(total\_victims,prior\_mental\_illness) %>%  
 ggplot(mapping = aes(total\_victims)) + geom\_bar() + facet\_wrap(vars(prior\_mental\_illness)) + labs(x = "Number of Total Victims", y = "Frequency", title = "Shooter's Mental State Results in More Victims", subtitle = "Shooter's Prior Mental Illness (Yes/No) vs No. of Victims")



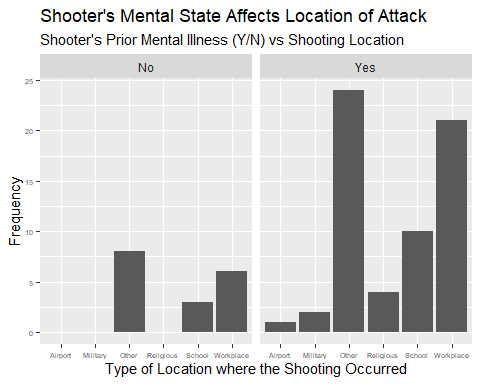
# This plot shows that there are many more shooters who have a priori known signs of mental illness than there are otherwise. The max number of victims that they affect is the same in each case. There is a greater tail of more victims in the case where a mental illness is known in the shooter prior to the shooting.

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

#I then explore the relative distribution of total victims versus fatalities in shootings and see if there is a difference between those who are killed or injured as a function of whether or not the shooter showed any prior signs of mental illness. I plot the total victims (green) versus fatalities (dark blue) on each of the two faceted plots that partitions those showing prior signs of mental illness ("Yes") and those that did not. The code and graph are below. The results show that the relative proportions of fatalities and injured victims are similar. However, there is a slightly higher proportion of fatalities if the shooter has no priori mental health issues, suggesting that they kill fewer people but that their victims are more targeted (and thus die); contrast this with the more sparse profile of victims when a mental health issue is known prior to the shooting - where there are more victims but the attack is more indiscriminate as there are more gaps in the histogram between fatalities (as opposed to injured). This said, it is difficult to judge entirely because the statistics of the 'No' data are not so many.  
  
mass\_shootings %>%  
 filter (!is.na(prior\_mental\_illness)) %>%  
 group\_by(total\_victims,fatalities,prior\_mental\_illness) %>%  
 ggplot(mapping = aes(x = total\_victims, y = fatalities, fill = fatalities)) + geom\_bar(stat="identity", colour = "green") + facet\_wrap(vars(prior\_mental\_illness)) + labs(x = "Number of Total Victims (Green) and Fatalities (Black)", y = "Frequency", title = "Mentally Ill Shooters Cause More Fatalities", subtitle = "Shooter's Prior Mental Illness (Y/N) vs State of Victim")

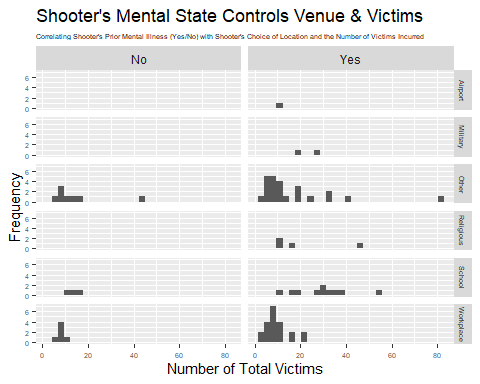


#I now explore possible correlation between mental health and location type using similar code and visualisation:  
  
mass\_shootings %>%  
 filter (!is.na(prior\_mental\_illness)) %>%  
 group\_by(location\_type,prior\_mental\_illness) %>%  
 ggplot(mapping = aes(location\_type)) + geom\_bar() + facet\_wrap(vars(prior\_mental\_illness)) + theme(axis.text = element\_text(size=5)) + labs(x = "Type of Location where the Shooting Occurred", y = "Frequency", title = "Shooter's Mental State Affects Location of Attack", subtitle = "Shooter's Prior Mental Illness (Y/N) vs Shooting Location")



#This plot shows that the school, workplace or other location types are shooting venues with the same proportion, while only shooters with prior signs of mental illness appear to target religious, military and airport locations.  
  
#I then explore possible correlation between the three variables: prior\_mental\_illness, location\_type and total\_victims:  
  
mass\_shootings %>%  
 filter (!is.na(prior\_mental\_illness)) %>%  
 group\_by(total\_victims,location\_type,prior\_mental\_illness) %>%  
 ggplot(mapping = aes(x = total\_victims)) + geom\_histogram() + facet\_grid(rows = vars(location\_type), cols = vars(prior\_mental\_illness)) + theme(axis.text = element\_text(size=5),strip.text.y = element\_text (size = 5),plot.subtitle = element\_text(size=5)) + labs(x = "Number of Total Victims", y = "Frequency", title = "Shooter's Mental State Controls Venue & Victims", subtitle = "Correlating Shooter's Prior Mental Illness (Yes/No) with Shooter's Choice of Location and the Number of Victims Incurred")

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



#These results show that the Workplace and 'Other' are the worst types of locations for mass shootings that result in the greatest number of victims. The maximum number of victims in each case lies in the same bin of 8-10 in these histograms, irrespective of the mental state of the shooter, or the location type. Schools are the next most frequent type of location for a mass shooting in the US, with fewer victims overall but a much wider statistical variation of the number of victims for a given shooting than any other location type except for Other. One might interpret this in terms of US school staff being briefed in what to do in a mass shooting but the number of victims varying a lot because of the more unpredictable actions of a child when placed in the path of a shooter. However, we cannot make this interpretation with too much conviction as we don't have any corroboratory evidence beyond these distributions. There are few instances of shootings at religious, military and airport locations and those that exist occur exclusively when the shooter has shown prior signs of mental illness.

Make sure to provide a couple of sentences of written interpretation of your tables/figures. Graphs and tables alone will not be sufficient to answer this question.

# Exploring credit card fraud

We will be using a dataset with credit card transactions containing legitimate and fraud transactions. Fraud is typically well below 1% of all transactions, so a naive model that predicts that all transactions are legitimate and not fraudulent would have an accuracy of well over 99%– pretty good, no? (well, not quite as we will see later in the course)

You can read more on credit card fraud on [Credit Card Fraud Detection Using Weighted Support Vector Machine](https://www.scirp.org/journal/paperinformation.aspx?paperid=105944)

The dataset we will use consists of credit card transactions and it includes information about each transaction including customer details, the merchant and category of purchase, and whether or not the transaction was a fraud.

## Obtain the data

The dataset is too large to be hosted on Canvas or Github, so please download it from dropbox https://www.dropbox.com/sh/q1yk8mmnbbrzavl/AAAxzRtIhag9Nc\_hODafGV2ka?dl=0 and save it in your dsb repo, under the data folder

Rows: 671,028  
Columns: 14  
$ trans\_date\_trans\_time <dttm> 2019-02-22 07:32:58, 2019-02-16 15:07:20, 2019-…  
$ trans\_year <dbl> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2020, …  
$ category <chr> "entertainment", "kids\_pets", "personal\_care", "…  
$ amt <dbl> 7.79, 3.89, 8.43, 40.00, 54.04, 95.61, 64.95, 3.…  
$ city <chr> "Veedersburg", "Holloway", "Arnold", "Apison", "…  
$ state <chr> "IN", "OH", "MO", "TN", "CO", "GA", "MN", "AL", …  
$ lat <dbl> 40.1186, 40.0113, 38.4305, 35.0149, 39.4584, 32.…  
$ long <dbl> -87.2602, -80.9701, -90.3870, -85.0164, -106.385…  
$ city\_pop <dbl> 4049, 128, 35439, 3730, 277, 1841, 136, 190178, …  
$ job <chr> "Development worker, community", "Child psychoth…  
$ dob <date> 1959-10-19, 1946-04-03, 1985-03-31, 1991-01-28,…  
$ merch\_lat <dbl> 39.41679, 39.74585, 37.73078, 34.53277, 39.95244…  
$ merch\_long <dbl> -87.52619, -81.52477, -91.36875, -84.10676, -106…  
$ is\_fraud <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, …

The data dictionary is as follows

| column(variable) | description |
| --- | --- |
| trans\_date\_trans\_time | Transaction DateTime |
| trans\_year | Transaction year |
| category | category of merchant |
| amt | amount of transaction |
| city | City of card holder |
| state | State of card holder |
| lat | Latitude location of purchase |
| long | Longitude location of purchase |
| city\_pop | card holder’s city population |
| job | job of card holder |
| dob | date of birth of card holder |
| merch\_lat | Latitude Location of Merchant |
| merch\_long | Longitude Location of Merchant |
| is\_fraud | Whether Transaction is Fraud (1) or Not (0) |

* In this dataset, how likely are fraudulent transactions? Generate a table that summarizes the number and frequency of fraudulent transactions per year.

card\_fraud %>% #reading in dataset  
 count(is\_fraud) %>% #generating no. of fraud cases  
 mutate(freq\_as\_a\_percent = 100\*n/sum(n)) #calculate proportion of fraudulent cases by percent:

# A tibble: 2 × 3  
 is\_fraud n freq\_as\_a\_percent  
 <dbl> <int> <dbl>  
1 0 667092 99.4   
2 1 3936 0.587

* 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.

#Generating a table that summarizes the amount of legitimate and fraudulent transactions per year   
  
amt2019 <- card\_fraud %>%  
 filter(trans\_year == 2019) %>%  
 summarise(amt2019 = sum(amt))  
amt2019

# A tibble: 1 × 1  
 amt2019  
 <dbl>  
1 33606041.

amt2020 <- card\_fraud %>%  
 filter(trans\_year == 2020) %>%  
 summarise(amt2020 = sum(amt))  
  
transyear <- c("2019", "2020")   
yearlyamt <- c("33606041", "13577863")   
  
df <- data.frame(transyear, yearlyamt)   
df[] <- lapply(df, as.numeric)  
df

transyear yearlyamt  
1 2019 33606041  
2 2020 13577863

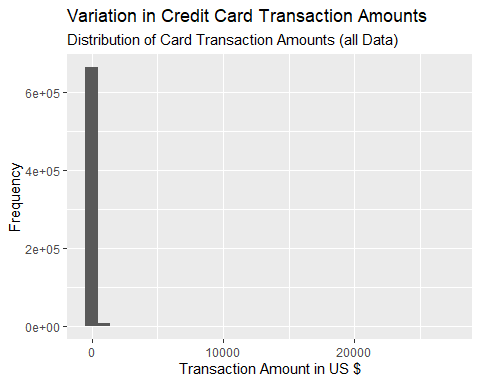
#calculate the % of fraudulent transactions, in US$ terms.  
  
df %>%  
 mutate(freq\_as\_percent = 100\*yearlyamt/sum(yearlyamt))

transyear yearlyamt freq\_as\_percent  
1 2019 33606041 71.22353  
2 2020 13577863 28.77647

* 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.

#Generating a histogram that shows the distribution of amounts charged to credit card  
  
card\_fraud %>%  
 ggplot(mapping = aes(amt)) + geom\_histogram() + labs(x = "Transaction Amount in US $", y = "Frequency", title = "Variation in Credit Card Transaction Amounts", subtitle ="Distribution of Card Transaction Amounts (all Data)")

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



#Checking the maximum value of transaction amounts, since the plot is heavily skewed to the low end.  
  
card\_fraud %>%  
 count(amt) %>%  
 arrange(desc(amt))

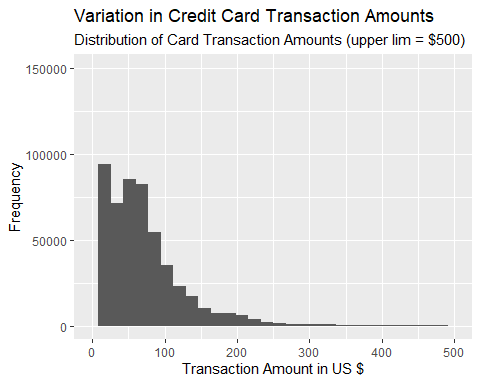
# A tibble: 40,945 × 2  
 amt n  
 <dbl> <int>  
 1 27120. 1  
 2 26544. 1  
 3 25087. 1  
 4 15047. 1  
 5 14850. 1  
 6 14468. 1  
 7 14238. 1  
 8 14113. 1  
 9 13537. 1  
10 12788. 1  
# ℹ 40,935 more rows

# summarise(max=max(amt))  
  
#Replotting the distribution of transaction amounts up to a $500 US limit.  
  
card\_fraud %>%  
 ggplot(mapping = aes(amt)) + geom\_histogram() + labs(x = "Transaction Amount in US $", y = "Frequency", title = "Variation in Credit Card Transaction Amounts", subtitle ="Distribution of Card Transaction Amounts (upper lim = $500)") + xlim(0,500)

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

Warning: Removed 8143 rows containing non-finite values (`stat\_bin()`).

Warning: Removed 2 rows containing missing values (`geom\_bar()`).

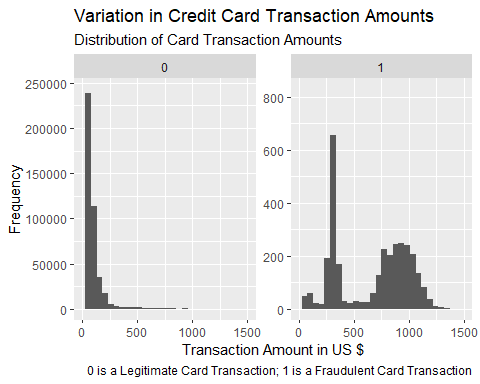


#This shows that most transactions are less than $300.   
  
#I now explore the distribution for legitimate and fraudulent transactions:  
  
card\_fraud %>%  
 ggplot(mapping = aes(amt)) + geom\_histogram() + labs(x = "Transaction Amount in US $", y = "Frequency", title = "Variation in Credit Card Transaction Amounts", subtitle ="Distribution of Card Transaction Amounts", caption = "0 is a Legitimate Card Transaction; 1 is a Fraudulent Card Transaction") + xlim(0,1500) + facet\_wrap(vars(is\_fraud), scales = "free\_y")

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

Warning: Removed 680 rows containing non-finite values (`stat\_bin()`).

Warning: Removed 4 rows containing missing values (`geom\_bar()`).



#Now calculating some summary statistics for both Fradulent and Legitimate Card Transactions:  
  
card\_fraud %>%  
 filter(!is.na(is\_fraud)) %>%  
 filter(is\_fraud == 0) %>%  
 summarise(mean\_amt = mean(amt))

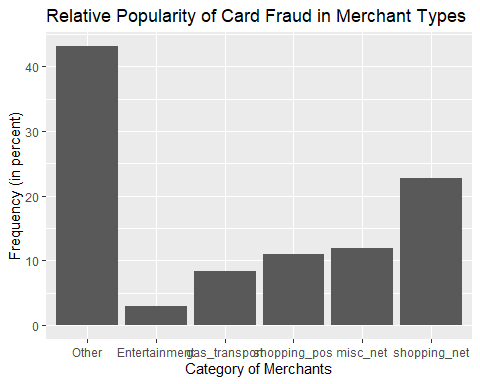
# A tibble: 1 × 1  
 mean\_amt  
 <dbl>  
1 67.6

card\_fraud %>%  
 filter(!is.na(is\_fraud)) %>%  
 filter(is\_fraud == 1) %>%  
 summarise(mean\_amt = mean(amt))

# A tibble: 1 × 1  
 mean\_amt  
 <dbl>  
1 527.

* 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.

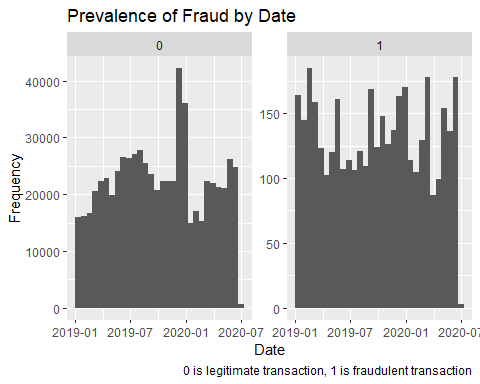
card\_fraud %>%  
 filter(is\_fraud == 1) %>%  
 count(category) %>% #generating no. of fraud cases  
 mutate(freq\_as\_a\_percent = 100\*n/sum(n)) %>% #calculating frequency as a percentage.  
 arrange(desc(freq\_as\_a\_percent)) %>%  
 mutate(recode\_category = case\_when(  
 category %in% c("entertainment") ~ "Entertainment",  
 category %in% c("shopping\_net") ~ "shopping\_net",  
 category %in% c("misc\_net") ~ "misc\_net",  
 category %in% c("shopping\_pos") ~ "shopping\_pos",  
 category %in% c("gas\_transport") ~ "gas\_transport",  
 TRUE ~ "Other"  
 )) %>%  
 arrange(desc(n)) %>%  
 mutate(recode\_category = fct\_reorder(recode\_category, n)) %>%  
 ggplot(mapping = aes(recode\_category,freq\_as\_a\_percent)) + geom\_col() + labs(x = "Category of Merchants", y = "Frequency (in percent)", title = "Relative Popularity of Card Fraud in Merchant Types")



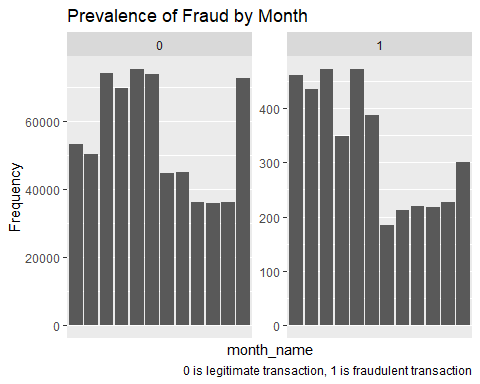
When is fraud more prevalent? Which days, months, hours? To create new variables to help you in your analysis, we use the lubridate package and the following code

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)  
 ) %>%  
  
 ggplot(mapping = aes(date\_only)) + geom\_histogram() + facet\_wrap(vars(is\_fraud), scales = "free\_y") + labs (x = "Date", y = "Frequency", title = "Prevalence of Fraud by Date", caption = "0 is legitimate transaction, 1 is fraudulent transaction")

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

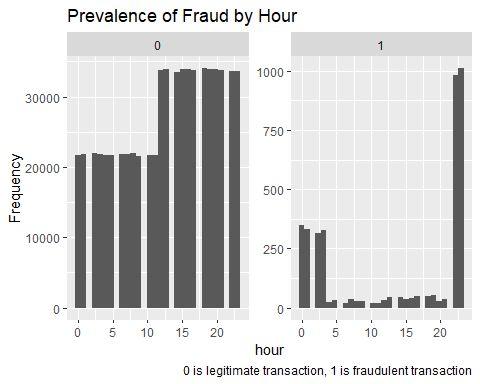


#This plot shows various spikes in frequency of fraudlent transaction activity. However, it is not entirely clear what is the nature of the pattern. So it is better to look at more fine grain time-based data as we see next (by month and then by hour).   
  
card\_fraud %>%  
mutate(  
 date\_only = lubridate::date(trans\_date\_trans\_time),  
 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)  
 ) %>%  
 ggplot(mapping = aes(month\_name)) + geom\_bar() + facet\_wrap(vars(is\_fraud), scales = "free\_y") + scale\_x\_discrete(breaks = 1:12, labels=c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")) + labs (x = "month\_name", y = "Frequency", title = "Prevalence of Fraud by Month", caption = "0 is legitimate transaction, 1 is fraudulent transaction")



#This plot shows that fraudulent transactions occur more often in winter months, especially from Jan-May. However, the effect is a lot more slight than other time types that we explore in this question.  
  
  
  
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)  
 ) %>%  
 ggplot(mapping = aes(hour)) + geom\_histogram() + facet\_wrap(vars(is\_fraud), scales = "free\_y") + labs (x = "hour", y = "Frequency", title = "Prevalence of Fraud by Hour", caption = "0 is legitimate transaction, 1 is fraudulent transaction")

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



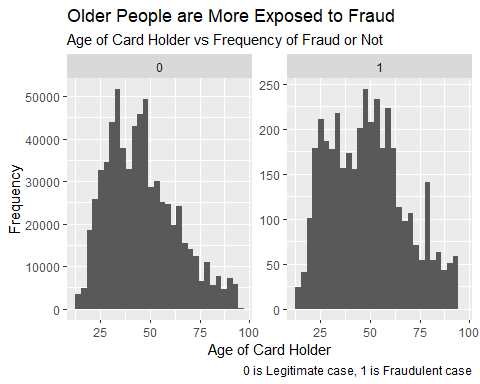
# This plot shows that fraudulent transactions mostly occur overnight, occurring by far more commonly at 22-23hr, and leading into the early hours of the new day.

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

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

library(lubridate)   
card\_fraud %>%   
 mutate(  
 age = interval(dob, trans\_date\_trans\_time)/years(1),  
 ) %>%   
 ggplot(mapping = aes(age)) + geom\_histogram() + facet\_wrap(vars(is\_fraud), scales = "free\_y") + labs(x = "Age of Card Holder", y = "Frequency", title = "Older People are More Exposed to Fraud", subtitle = "Age of Card Holder vs Frequency of Fraud or Not", caption = "0 is Legitimate case, 1 is Fraudulent case")

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

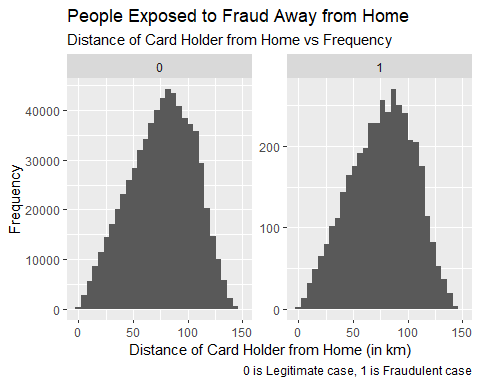


# This plot shows that there is a small but significant correlation between age of card holder and their exposure to fraudlent activity at the high age range, as the data are slightly more concentrated there. However, the overall distributions are broadly similar so it is not a massive effect.

* 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 %>%  
#fraud <- 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))  
  
 ) %>%  
ggplot(mapping = aes(distance\_km)) + geom\_histogram() + facet\_wrap(vars(is\_fraud), scales = "free\_y") + labs(x = "Distance of Card Holder from Home (in km)", y = "Frequency", title = "People Exposed to Fraud Away from Home", subtitle = "Distance of Card Holder from Home vs Frequency", caption = "0 is Legitimate case, 1 is Fraudulent case")

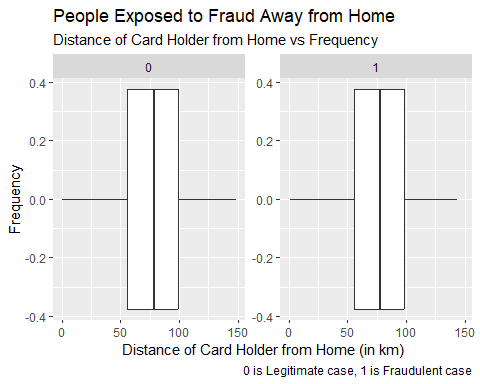
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# These two plots show that there is no discernible difference between fraudulent versus legitimate transactions as a function of distance of a card holder from their home.

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?

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))  
  
 ) %>%  
ggplot(mapping = aes(distance\_km)) + geom\_boxplot() + facet\_wrap(vars(is\_fraud), scales = "free\_y") + labs(x = "Distance of Card Holder from Home (in km)", y = "Frequency", title = "People Exposed to Fraud Away from Home", subtitle = "Distance of Card Holder from Home vs Frequency", caption = "0 is Legitimate case, 1 is Fraudulent case")



#These boxplots also show that there are no discernible statistical metrics which evidence an influence of distance of the card holder from home with the likelihood of fraudulent card-transaction activity.

# 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

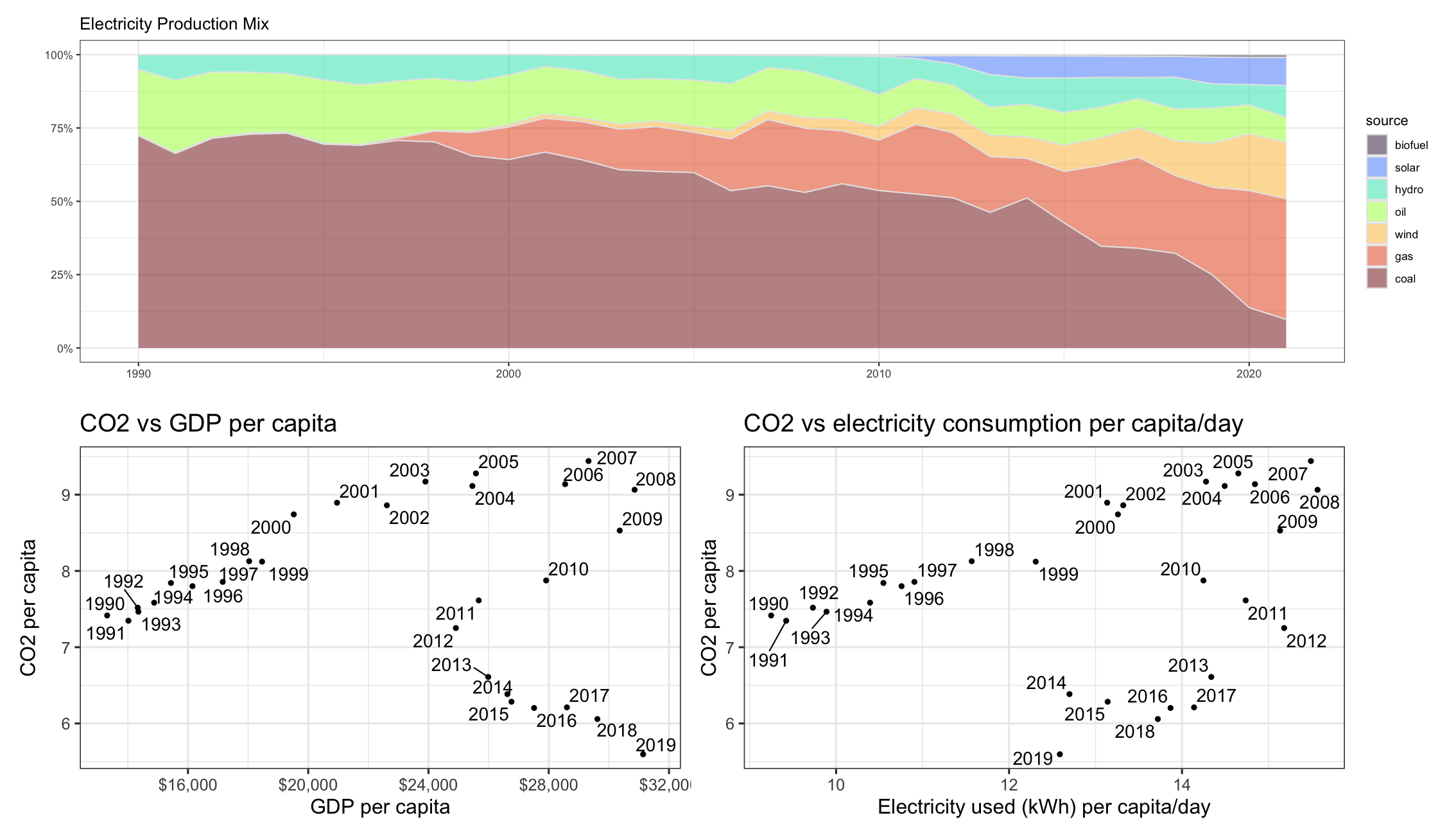
## 3. A scatter plot that looks at how electricity usage (kWh) per capita/day GDP per capita are related

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  
 )   
  
  
# For question 1. A stacked area chart that shows how your own country generated its electricity since 2000, where we are told:  
  
#You will use  
  
#geom\_area(colour="grey90", alpha = 0.5, position = "fill")  
  
#I begin with the following given that I am from the UK (so I monitor my own country as requested). I read in the energy dataset and execute the code below:  
  
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  
 )   
  
#I then open energy and remove NAs and filter on my home country and the year from 2000.  
  
#I then tried to plot this in various ways using the following code as well as using code to categorise each energy production type (e.g. biofuel, coal, etc) with commands such as case\_when etc. But nothing seems to work. So I have commmented out the code below so that I can still knit the code into a submittable document.  
  
# energy %>%  
# filter(!is.na(solar,wind)) %>%  
#filter(iso\_code == "GBR", year > 1999) %>%  
# ggplot (mapping = aes(x = year, y = #electricity\_generation) + geom\_area(colour = "solar", "wind") + geom\_area()  
 #(colour="grey90", alpha = 0.5, position = "fill")  
   
  
  
#I then tried question 2 and 3. I loaded in the datasets:  
  
# 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)  
  
  
#I then tried to left\_join these two datasets using the iso3c column as the common data. But it doesn't work. Maybe it is just late in the evening by now and I can't concentrate so well! Either way, what I would have done is left\_join these datasets and then create a scatterplot with the commented ggplot command below for question 2:  
  
#left\_join(co2\_percap,gdp\_percap, by "iso3c") %>%  
# ggplot (mapping = aes(x = gdp\_percap, y = co2\_percap)) + labs(x = "GDP per capita for the UK", y = "CO2 per capita for the UK", title = "CO2 vs GDP per capita") + geom\_text(stat = 'count', aes(label = after\_stat(count), vjust = 1,colour = "black", size = 4)) + theme(panel.grid = element\_line(color = "#8ccde3", size = 0.75, linetype = 2))   
  
#For question 3, I would plot a scatterplot with similar ggplot syntax to the above described for question 2.

Specific questions:

1. How would you turn energy to long, tidy format?
2. You may need to join these data frames
   * Use left\_join from dplyr to [join the tables](http://r4ds.had.co.nz/relational-data.html)
   * To complete the merge, you need a unique *key* to match observations between the data frames. Country names may not be consistent among the three dataframes, so please use the 3-digit ISO code for each country
   * An aside: There is a great package called [countrycode](https://github.com/vincentarelbundock/countrycode) that helps solve the problem of inconsistent country names (Is it UK? United Kingdon? Great Britain?). countrycode() takes as an input a country’s name in a specific format and outputs it using whatever format you specify.
3. Write a function that takes as input any country’s name and returns all three graphs. You can use the patchwork package to arrange the three graphs as shown below



# Deliverables

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

# Details

* Who did you collaborate with: No-one
* Approximately how much time did you spend on this problem set:
* What, if anything, gave you the most trouble: ANSWER HERE

**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.