Homework 2

Gustavo Passebon Mendonca

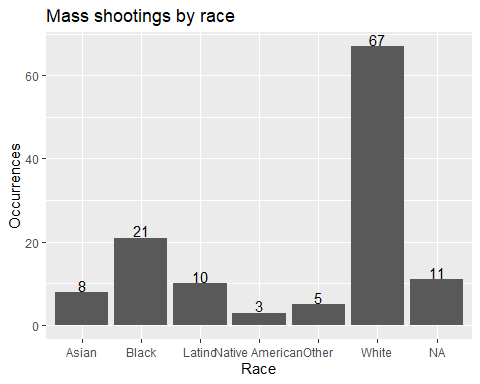
2023-05-22

Table of Contents

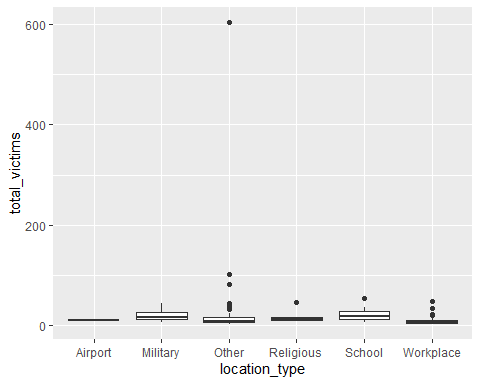
## Rows: 125  
## Columns: 14  
## $ case <chr> "Oxford High School shooting", "San Jose VTA shoo…  
## $ year <dbl> 2021, 2021, 2021, 2021, 2021, 2021, 2020, 2020, 2…  
## $ month <chr> "Nov", "May", "Apr", "Mar", "Mar", "Mar", "Mar", …  
## $ day <dbl> 30, 26, 15, 31, 22, 16, 16, 26, 10, 6, 31, 4, 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, …

cases\_year=mass\_shootings %>%   
 group\_by(year) %>%   
 count(year,sort=TRUE)

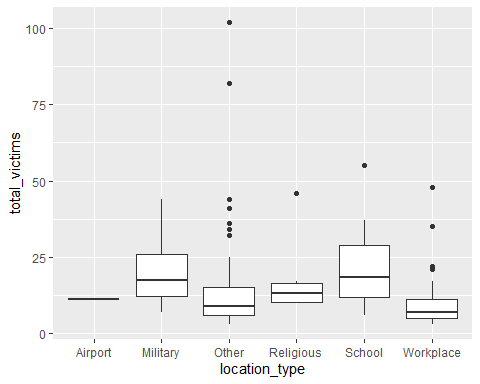
cases\_race = mass\_shootings %>%   
 group\_by(race) %>%   
 count(race,sort=TRUE)  
  
ggplot(cases\_race, aes(x=race, y = n))+  
 geom\_col()+  
 labs(title="Mass shootings by race",  
 x ="Race", y = "Occurrences")+  
 geom\_text(aes(label = n), vjust = -0.1)



#Note - do not group the data in the dataframe creation phase when building a boxplot  
cases\_location = mass\_shootings %>%   
 group\_by(location\_type)   
  
ggplot(cases\_location, aes(x=location\_type, y = total\_victims))+  
 geom\_boxplot()



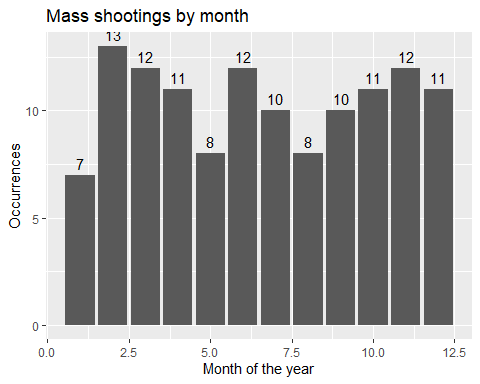
#Removing the Strip Massacre from the "Case" column  
cases\_location\_vegas = mass\_shootings %>%  
 filter(case!="Las Vegas Strip massacre") %>%   
 group\_by(location\_type)  
#Redoing the box plot  
ggplot(cases\_location\_vegas, aes(x=location\_type, y = total\_victims))+  
 geom\_boxplot()



#Applying the filters  
white\_males = mass\_shootings %>%   
 filter(year>2000) %>%   
 filter(prior\_mental\_illness=="Yes") %>%   
 filter(race=="White") %>%   
 count(prior\_mental\_illness)  
  
#Showing the result  
white\_males

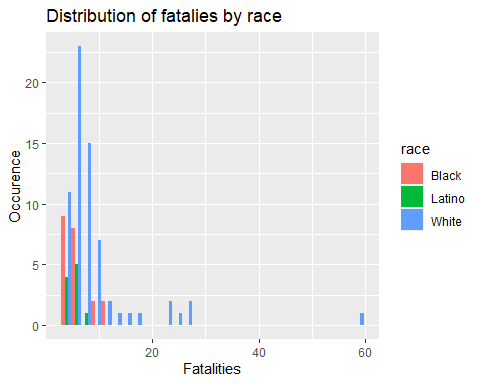
## # A tibble: 1 × 2  
## prior\_mental\_illness n  
## <chr> <int>  
## 1 Yes 23

#Dataframe to organize the months  
months = data.frame(c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec"),  
 c(1,2,3,4,5,6,7,8,9,10,11,12))  
colnames(months) = c("month","Number")  
#Filtering and counting the information  
cases\_month = mass\_shootings %>%   
 group\_by(month) %>%   
 count(month)  
  
#Joining the two tables  
cases\_month = left\_join(cases\_month,months,by="month")  
  
#Ordering the joined table by number of the month  
cases\_month = cases\_month %>%   
 arrange(Number,by\_group="Number")  
  
#Plotting  
ggplot(cases\_month, aes(x=Number, y = n))+  
 geom\_col()+  
 labs(title="Mass shootings by month",  
 x ="Month of the year", y = "Occurrences")+  
 geom\_text(aes(label = n), vjust = -0.5)

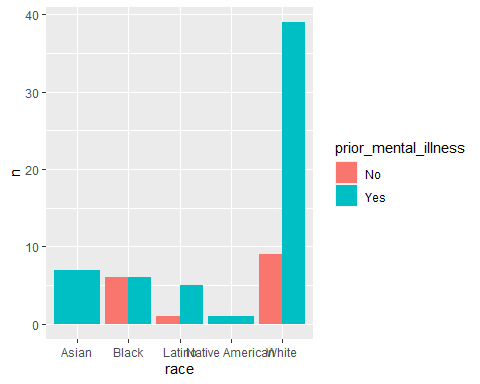


cases\_distrib = mass\_shootings %>%  
 filter(race %in% c("White","Black","Latino")) %>%   
 group\_by(race)  
  
ggplot(cases\_distrib,aes(x=fatalities,fill=race))+  
 geom\_histogram(alpha=1,position="dodge")+  
 labs(title="Distribution of fatalies by race",  
 x ="Fatalities", y = "Occurence")

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

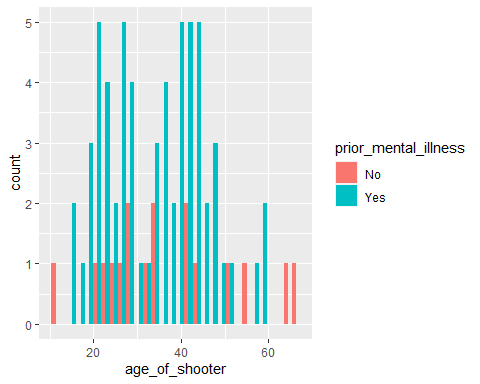


mental\_distrib = mass\_shootings %>%   
 filter(is.na(prior\_mental\_illness)==FALSE) %>%   
 group\_by(prior\_mental\_illness)  
   
mental\_location = mass\_shootings %>%  
 filter(is.na(prior\_mental\_illness)==FALSE) %>%   
 group\_by(prior\_mental\_illness) %>%   
 count(location\_type)  
  
mental\_race = mass\_shootings %>%  
 filter(is.na(prior\_mental\_illness)==FALSE) %>%   
 filter(is.na(race)==FALSE) %>%   
 group\_by(prior\_mental\_illness) %>%   
 count(race)  
  
ggplot(mental\_race,aes(x=race,y=n,fill=prior\_mental\_illness))+  
 geom\_col(position = "dodge")



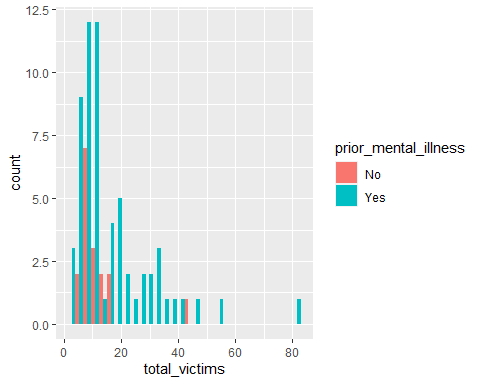
ggplot(mental\_distrib,aes(x=age\_of\_shooter,fill=prior\_mental\_illness))+  
 geom\_histogram(alpha=1,position="dodge")

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

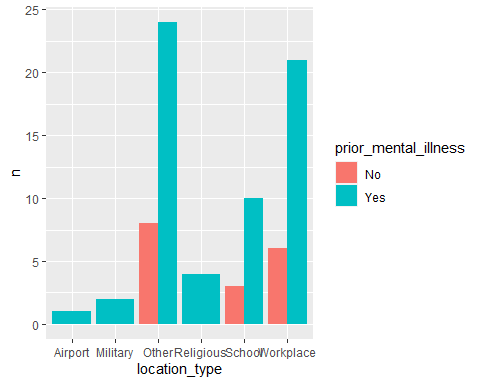


#Relationship Mental Illness vs Victims  
ggplot(mental\_distrib,aes(x=total\_victims,fill=prior\_mental\_illness))+  
 geom\_histogram(alpha=1,position="dodge")

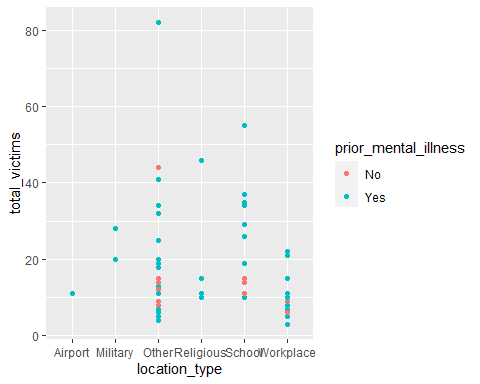
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#Relationship Location type vs Mental Illness  
ggplot(mental\_location,aes(x=location\_type,y=n,fill=prior\_mental\_illness))+  
 geom\_col(position = "dodge")



#Relationship between the 3 variables  
ggplot(mental\_distrib,aes(x=location\_type,y=total\_victims,color=prior\_mental\_illness))+  
 geom\_point()



## 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, …

total\_transactions = card\_fraud %>%   
 group\_by(trans\_year) %>%  
 count(trans\_year,name="transactions")  
total\_frauds = card\_fraud %>%  
 filter(is\_fraud==1) %>%   
 group\_by(trans\_year) %>%  
 count(is\_fraud,name="frauds")  
total\_frauds = left\_join(total\_frauds,total\_transactions,by="trans\_year")  
total\_frauds %>%   
 mutate(fraud\_rate = frauds/transactions)

## # A tibble: 2 × 5  
## # Groups: trans\_year [2]  
## trans\_year is\_fraud frauds transactions fraud\_rate  
## <dbl> <dbl> <int> <int> <dbl>  
## 1 2019 1 2721 478646 0.00568  
## 2 2020 1 1215 192382 0.00632

card\_fraud %>%  
 group\_by(trans\_year,is\_fraud) %>%   
 summarize(count = sum(amt))%>%  
 ungroup() %>%   
 mutate(is\_fraud = ifelse(is\_fraud == 1, "fraud", "valid")) %>%   
 pivot\_wider(names\_from = "is\_fraud",values\_from = "count") %>%   
 mutate(ratio=fraud/(valid+fraud))

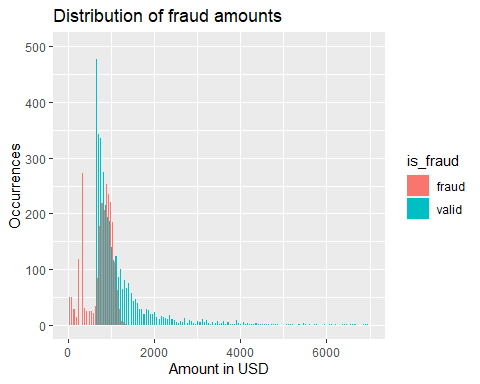
## `summarise()` has grouped output by 'trans\_year'. You can override using the  
## `.groups` argument.

## # A tibble: 2 × 4  
## trans\_year valid fraud ratio  
## <dbl> <dbl> <dbl> <dbl>  
## 1 2019 32182901. 1423140. 0.0423  
## 2 2020 12925914. 651949. 0.0480

fraud\_hist = card\_fraud %>%  
 mutate(is\_fraud = ifelse(is\_fraud == 1, "fraud", "valid"))  
   
ggplot(fraud\_hist,aes(x=amt,fill=is\_fraud))+  
 scale\_y\_continuous(limits=c(0,500))+  
 scale\_x\_continuous(limits=c(0,7000))+  
 geom\_histogram(alpha=1,position="dodge",binwidth = 50)+  
 labs(title="Distribution of fraud amounts",  
 x ="Amount in USD", y = "Occurrences")

## Warning: Removed 38 rows containing non-finite values (`stat\_bin()`).

## Warning: Removed 16 rows containing missing values (`geom\_bar()`).



fraud\_hist %>%   
 group\_by(is\_fraud) %>%   
 summarize(countAmt = n(),meanAmount=mean(amt),medianAmount=median(amt),sdAmt=sd(amt),minAmt=min(amt),maxAmt=max(amt))

## # A tibble: 2 × 7  
## is\_fraud countAmt meanAmount medianAmount sdAmt minAmt maxAmt  
## <chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 fraud 3936 527. 369. 391. 1.06 1334.  
## 2 valid 667092 67.6 47.2 155. 1 27120.

card\_fraud %>%   
 group\_by(category) %>%   
 count(is\_fraud) %>%   
 ungroup() %>%   
 mutate(is\_fraud = ifelse(is\_fraud == 1, "fraud", "valid")) %>%  
 pivot\_wider(names\_from = "is\_fraud",values\_from = "n") %>%   
 mutate(ratio=fraud/(valid+fraud)) %>%   
 arrange(desc(ratio))

## # A tibble: 14 × 4  
## category valid fraud ratio  
## <chr> <int> <int> <dbl>  
## 1 shopping\_net 49851 892 0.0176   
## 2 grocery\_pos 62859 932 0.0146   
## 3 misc\_net 32359 470 0.0143   
## 4 shopping\_pos 59982 434 0.00718  
## 5 gas\_transport 67720 326 0.00479  
## 6 grocery\_net 23405 80 0.00341  
## 7 travel 20803 70 0.00335  
## 8 misc\_pos 41110 134 0.00325  
## 9 entertainment 48405 116 0.00239  
## 10 personal\_care 46737 106 0.00226  
## 11 kids\_pets 58650 122 0.00208  
## 12 food\_dining 47449 78 0.00164  
## 13 home 63493 104 0.00164  
## 14 health\_fitness 44269 72 0.00162

fraud\_time = 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)  
 )  
#Most common day  
fraud\_time %>%   
 group\_by(weekday) %>%   
 count(weekday,sort=TRUE)

## # A tibble: 7 × 2  
## # Groups: weekday [7]  
## weekday n  
## <ord> <int>  
## 1 seg 131419  
## 2 dom 129550  
## 3 sáb 104039  
## 4 ter 82930  
## 5 sex 78951  
## 6 qui 76200  
## 7 qua 67939

#Most common hour  
fraud\_time %>%   
 group\_by(hour) %>%   
 count(hour,sort=TRUE)

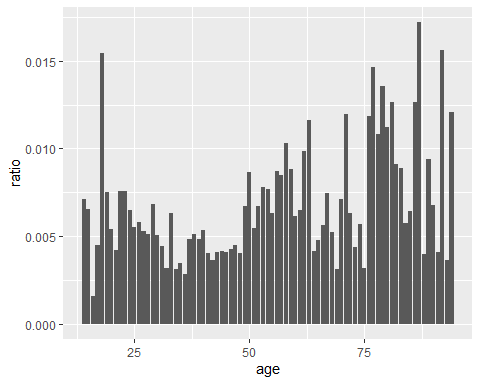
## # A tibble: 24 × 2  
## # Groups: hour [24]  
## hour n  
## <int> <int>  
## 1 22 34674  
## 2 23 34625  
## 3 18 34131  
## 4 19 33987  
## 5 20 33978  
## 6 13 33972  
## 7 16 33960  
## 8 15 33907  
## 9 17 33842  
## 10 21 33814  
## # ℹ 14 more rows

#Most common month  
fraud\_time %>%   
 group\_by(month\_name) %>%   
 count(month\_name,sort=TRUE)

## # A tibble: 12 × 2  
## # Groups: month\_name [12]  
## month\_name n  
## <ord> <int>  
## 1 mai 75801  
## 2 mar 74478  
## 3 jun 74214  
## 4 dez 72986  
## 5 abr 69876  
## 6 jan 53806  
## 7 fev 50660  
## 8 ago 45280  
## 9 jul 44974  
## 10 set 36533  
## 11 nov 36333  
## 12 out 36087

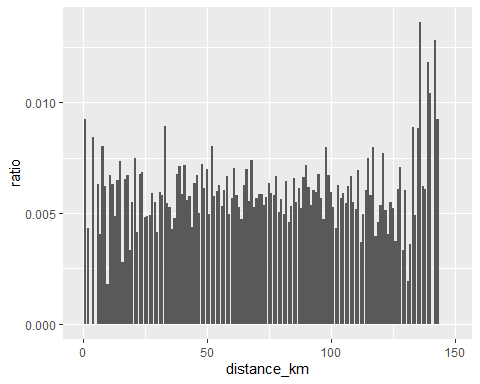
#Filtering the data  
fraud\_age = card\_fraud %>%  
 mutate(  
 age = round(interval(dob, trans\_date\_trans\_time) / years(1)),-1) %>%  
 group\_by(age) %>%   
 count(is\_fraud) %>%  
 ungroup() %>%   
 mutate(is\_fraud = ifelse(is\_fraud == 1, "fraud", "valid")) %>%  
 pivot\_wider(names\_from = "is\_fraud",values\_from = "n") %>%   
 mutate(ratio=fraud/(valid+fraud))  
   
#Ploting the relationship between age and the ratio of fraudulent transactions  
ggplot(fraud\_age,aes(x=age,y=ratio))+  
 geom\_col()

## Warning: Removed 2 rows containing missing values (`position\_stack()`).



#It appears that the occurence of fraud increases with age up to a certain point

# 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))  
  
 )  
  
#Filtering  
card\_fraud$distance\_km=round(card\_fraud$distance\_km,0)  
fraud\_distance = card\_fraud %>%  
 group\_by(distance\_km) %>%   
 count(is\_fraud) %>%  
 ungroup() %>%   
 mutate(is\_fraud = ifelse(is\_fraud == 1, "fraud", "valid")) %>%  
 pivot\_wider(names\_from = "is\_fraud",values\_from = "n") %>%   
 mutate(fraud = ifelse(is.na(fraud)==TRUE, 0, fraud)) %>%   
 mutate(ratio=fraud/(valid+fraud))  
  
#Plotting  
ggplot(fraud\_distance,aes(x=distance\_km,y=ratio))+  
 geom\_col()

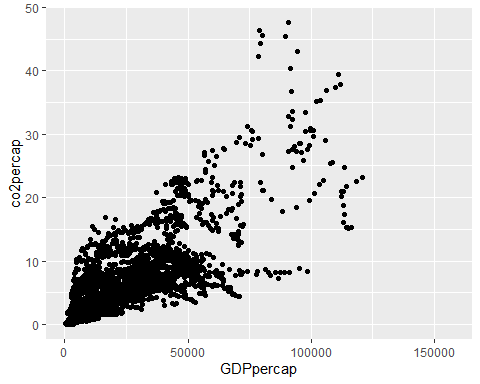


#As it is possible to see, there is a substantial increase in the % of fraudulent cases after 130km

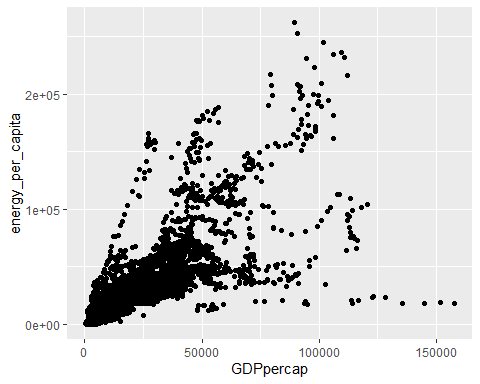
# Download electricity data  
url <- "https://nyc3.digitaloceanspaces.com/owid-public/data/energy/owid-energy-data.csv"  
  
energy <- read\_csv(url) %>%   
 filter(year >= 1990) %>%   
 drop\_na(iso\_code) %>%   
 select(1:3,  
 biofuel = biofuel\_electricity,  
 coal = coal\_electricity,  
 gas = gas\_electricity,  
 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)  
  
#Turning energy into tidy, long format  
energy %>%   
 pivot\_longer(cols=4:18,names\_to = "characteristics",values\_to="values")

## # A tibble: 104,265 × 5  
## country year iso\_code characteristics values  
## <chr> <dbl> <chr> <chr> <dbl>  
## 1 Afghanistan 1990 AFG biofuel NA  
## 2 Afghanistan 1990 AFG coal NA  
## 3 Afghanistan 1990 AFG gas NA  
## 4 Afghanistan 1990 AFG hydro NA  
## 5 Afghanistan 1990 AFG nuclear NA  
## 6 Afghanistan 1990 AFG oil NA  
## 7 Afghanistan 1990 AFG other\_renewable NA  
## 8 Afghanistan 1990 AFG solar NA  
## 9 Afghanistan 1990 AFG wind NA  
## 10 Afghanistan 1990 AFG electricity\_demand NA  
## # ℹ 104,255 more rows

#Merging the tables  
#Selecting the relevant info  
co2\_percap = co2\_percap %>%   
 select(iso3c,year,co2percap)  
gdp\_percap = gdp\_percap %>%   
 select(iso3c,year,GDPpercap)  
energy\_select = energy %>%   
 select(year,iso\_code,energy\_per\_capita)  
  
#Relationship GDP vs CO2 per capita  
joint\_table = left\_join(gdp\_percap,co2\_percap,by=c("iso3c"="iso3c","year"="year"))  
ggplot(joint\_table,aes(x=GDPpercap,y=co2percap))+  
 geom\_point()



#Relationship GDP vs Energy consumption per capita  
energy\_gdp = left\_join(gdp\_percap,energy\_select,by=c("iso3c"="iso\_code","year"="year"))  
ggplot(energy\_gdp,aes(x=GDPpercap,y=energy\_per\_capita))+  
 geom\_point()



# Details

* Who did you collaborate with: No one
* Approximately how much time did you spend on this problem set: 7 hours
* What, if anything, gave you the most trouble: -