Homerwork 1

Ignacio Landerreche

5/14/23

# Data Manipulation

## Problem 1: Use logical operators to find flights that:

- Had an arrival delay of two or more hours (\> 120 minutes)  
- Flew to Houston (IAH or HOU)  
- Were operated by United (`UA`), American (`AA`), or Delta (`DL`)  
- Departed in summer (July, August, and September)  
- Arrived more than two hours late, but didn't leave late  
- Were delayed by at least an hour, but made up over 30 minutes in flight

# Had an arrival delay of two or more hours (> 120 minutes)  
flights %>%   
 filter(arr\_time>120)

# A tibble: 319,950 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 517 515 2 830 819  
 2 2013 1 1 533 529 4 850 830  
 3 2013 1 1 542 540 2 923 850  
 4 2013 1 1 544 545 -1 1004 1022  
 5 2013 1 1 554 600 -6 812 837  
 6 2013 1 1 554 558 -4 740 728  
 7 2013 1 1 555 600 -5 913 854  
 8 2013 1 1 557 600 -3 709 723  
 9 2013 1 1 557 600 -3 838 846  
10 2013 1 1 558 600 -2 753 745  
# i 319,940 more rows  
# i 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

# Flew to Houston (IAH or HOU)  
flights %>%   
 filter(dest %in% c("IAH","HOU"))

# A tibble: 9,313 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 517 515 2 830 819  
 2 2013 1 1 533 529 4 850 830  
 3 2013 1 1 623 627 -4 933 932  
 4 2013 1 1 728 732 -4 1041 1038  
 5 2013 1 1 739 739 0 1104 1038  
 6 2013 1 1 908 908 0 1228 1219  
 7 2013 1 1 1028 1026 2 1350 1339  
 8 2013 1 1 1044 1045 -1 1352 1351  
 9 2013 1 1 1114 900 134 1447 1222  
10 2013 1 1 1205 1200 5 1503 1505  
# i 9,303 more rows  
# i 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

# Were operated by United (`UA`), American (`AA`), or Delta (`DL`)  
flights %>%   
 filter(carrier %in% c("UA","AA","DL"))

# A tibble: 139,504 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 517 515 2 830 819  
 2 2013 1 1 533 529 4 850 830  
 3 2013 1 1 542 540 2 923 850  
 4 2013 1 1 554 600 -6 812 837  
 5 2013 1 1 554 558 -4 740 728  
 6 2013 1 1 558 600 -2 753 745  
 7 2013 1 1 558 600 -2 924 917  
 8 2013 1 1 558 600 -2 923 937  
 9 2013 1 1 559 600 -1 941 910  
10 2013 1 1 559 600 -1 854 902  
# i 139,494 more rows  
# i 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

# Departed in summer (July, August, and September)  
flights %>%   
 filter(month %in% c(7,8,9))

# A tibble: 86,326 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 7 1 1 2029 212 236 2359  
 2 2013 7 1 2 2359 3 344 344  
 3 2013 7 1 29 2245 104 151 1  
 4 2013 7 1 43 2130 193 322 14  
 5 2013 7 1 44 2150 174 300 100  
 6 2013 7 1 46 2051 235 304 2358  
 7 2013 7 1 48 2001 287 308 2305  
 8 2013 7 1 58 2155 183 335 43  
 9 2013 7 1 100 2146 194 327 30  
10 2013 7 1 100 2245 135 337 135  
# i 86,316 more rows  
# i 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

# Arrived more than two hours late, but didn't leave late  
flights %>%   
 filter(arr\_delay>120 & dep\_delay==0)

# A tibble: 3 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
1 2013 10 7 1350 1350 0 1736 1526  
2 2013 5 23 1810 1810 0 2208 2000  
3 2013 7 1 905 905 0 1443 1223  
# i 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

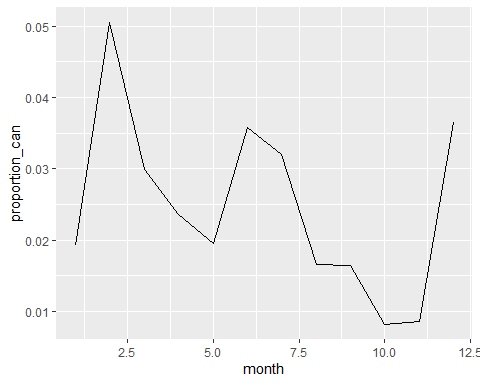
# Were delayed by at least an hour, but made up over 30 minutes in flight  
flights %>%   
 filter(arr\_delay>=-30 & arr\_delay<0 & dep\_delay>=60)

# A tibble: 3 x 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
1 2013 2 26 1000 900 60 1513 1540  
2 2013 4 5 932 831 61 1149 1151  
3 2013 7 11 2018 1915 63 2210 2211  
# i 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

## Problem 2: What months had the highest and lowest proportion of cancelled flights? Interpret any seasonal patterns. To determine if a flight was cancelled use the following code

flights %>%   
 filter(is.na(dep\_time))

# What months had the highest and lowest % of cancelled flights?  
cancelled\_flights<-flights %>%   
 group\_by(month) %>% #groupping by month  
 summarise(total\_flights\_m=n(),cancelled=sum(is.na(dep\_time)),proportion\_can=cancelled/total\_flights\_m) #calculating total number of flights, the cancelled number of flights and the proportion  
  
  
#Creating a plot to see the relationship  
ggplot(cancelled\_flights,aes(x=month,y=proportion\_can))+geom\_line()



#I see that there are three peaks: February, June and December. One could infer that June and December are high seasonality because of school vacations and another assumption would be that flights are cancelled because of cold weather and ergo Jan-March are months with a high proportion of cancelled flights. However without a deeper analysis one can't conclude much from this.

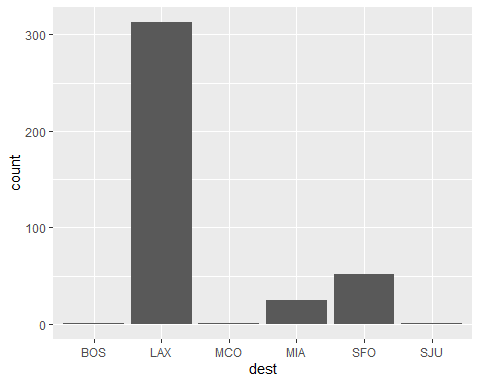
## Problem 3: What plane (specified by the tailnum variable) traveled the most times from New York City airports in 2013? Please left\_join() the resulting table with the table planes (also included in the nycflights13 package).

For the plane with the greatest number of flights and that had more than 50 seats, please create a table where it flew to during 2013.

problem3<-flights %>%   
 filter(year==2013) %>%   
 left\_join(planes,by="tailnum") %>% #Left joining so we can have the number of seats by each flight   
 filter(seats>50)%>% #filtering by flights > 50 seats  
 group\_by(tailnum)%>% #groupping by flight number and counting it  
 summarise(count=n()) %>%   
 arrange(desc(count))  
  
head(problem3,1)

# A tibble: 1 x 2  
 tailnum count  
 <chr> <int>  
1 N328AA 393

#We know that N328AA was the plane with more than 50 seats  
  
planemoreflown<-flights %>%  
 filter(year==2013) %>%   
 left\_join(planes,by="tailnum")%>%  
 filter(tailnum=="N328AA")%>% #Having only the flight >50 seats that flew the most  
 group\_by(dest) %>% #groupping by destination and counting them  
 summarise(count=n()) %>%   
 arrange(desc(count))  
  
  
ggplot(planemoreflown,aes(x=dest,y=count))+geom\_bar(stat="identity")

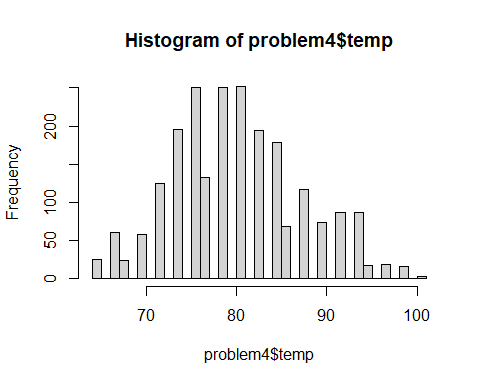


#The city the flight N3288AA flights the most is from NY to Los Angeles with 313 flights.

## Problem 4: The nycflights13 package includes a table (weather) that describes the weather during 2013. Use that table to answer the following questions:

- What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
- What is the relationship between `dewp` and `humid`?  
- What is the relationship between `precip` and `visib`?

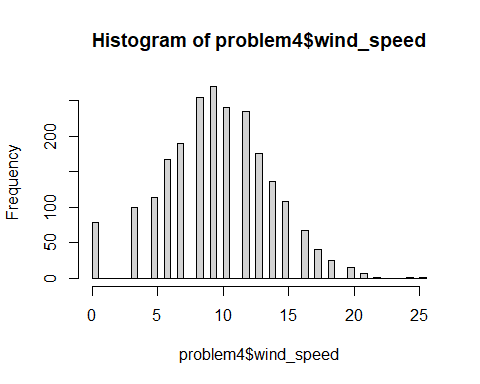
##Working on the data base  
problem4<-weather %>%   
 filter(year==2013 & month==7) %>%   
 arrange(wind\_speed)  
  
##Analyzing temperature  
hist(problem4$temp,breaks=50) #Distribution of temperature in July, on a bare eye it is difficult to see which distribution it might look like but it seems like it is a tri-modal distribution



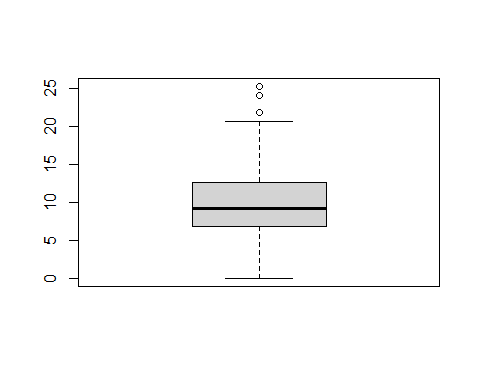
summary(problem4$temp) #basic stats from the sample of the temperature

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 64.04 75.02 78.98 80.07 84.20 100.04

##Analyzing wind\_speed  
hist(problem4$wind\_speed,breaks=50) #distribution of wind\_speed, looks more normal-like



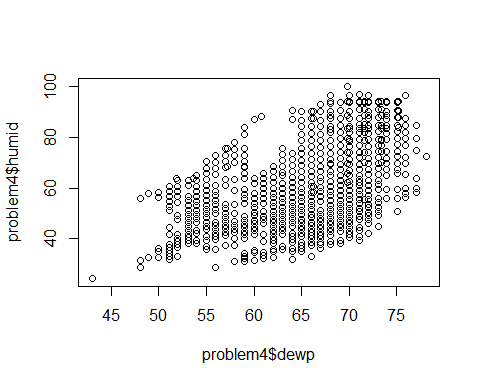
boxplot(problem4$wind\_speed) #with the boxplot we can identify the IQ range and outliers (specifically 3)



tail(problem4$wind\_speed,10)# these might be the outliers from the data distribution, everything above 20, basing everything also with the boxplot

[1] 20.71404 20.71404 20.71404 20.71404 20.71404 21.86482 24.16638 25.31716  
 [9] NA NA

##Relationship between dewp, and humid  
  
plot(problem4$dewp,problem4$humid) #there is no clear linear trend nor any quadratic trend based on the plot.



cor(problem4$dewp,problem4$humid) #based on the plot and the correlation, we see it is positive

[1] 0.5354781

plot(problem4$visib,problem4$precip) #looks like there is a relation between the two variables. Additionally, it looks like they are both discrete variables.

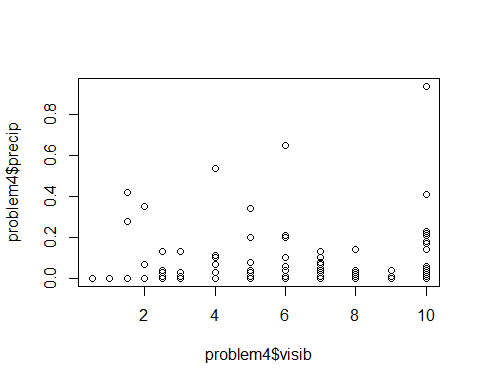
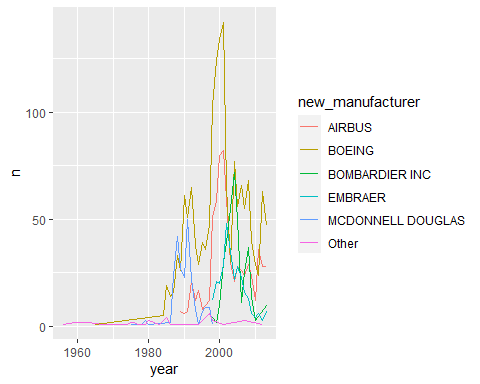


tabla1<-table(problem4$visib,problem4$precip) #based on this table, we see the relationship between these variables, but no further information can be gotten from this  
view(tabla1)

## Problem 5: Use the flights and planes tables to answer the following questions:

- How many planes have a missing date of manufacture?  
- What are the five most common manufacturers?  
- Has the distribution of manufacturer changed over time as reflected by the airplanes flying from NYC in 2013? (Hint: you may need to use case\_when() to recode the manufacturer name and collapse rare vendors into a category called Other.)

#Creating the database with manufacturer as #NA  
missing\_manufacturer<- planes %>%   
 filter(is.na(manufacturer))  
view(missing\_manufacturer) #there are no planes without a manufacturer  
  
#Creating database groupped by manufacturer, counting them and arranging them  
manufacturer<-planes %>%   
 group\_by(manufacturer) %>%   
 summarise(count=n()) %>%   
 arrange(desc(count))  
view(manufacturer)# It seems that there are couple of manufacturers that are the same but written differently, I will manually change them.  
total\_planes<- manufacturer%>% summarise(sum(count))  
#Changing the duplicate names  
  
manufacturer<-manufacturer %>%   
 mutate(manufacturer=case\_when(manufacturer=="AIRBUS INDUSTRIE" ~ "AIRBUS",  
 manufacturer %in% c("MCDONNELL DOUGLAS AIRCRAFT CO","MCDONNELL DOUGLAS AIRCRAFT CO","MCDONNELL DOUGLAS CORPORATION")~"MCDONNELL DOUGLAS",  
 .default=manufacturer  
   
 )) %>% group\_by(manufacturer) %>%   
 summarise(count=sum(count)) %>%   
 arrange(desc(count))  
  
view(manufacturer)  
  
#Validation:  
validation<- manufacturer %>% summarise(sum(count))-total\_planes  
  
topmanufacturer<-head(manufacturer$manufacturer,5) #These are the top 5 manufacturers: "BOEING" "AIRBUS" "BOMBARDIER INC" "EMBRAER" "MCDONNELL DOUGLAS"  
  
#Creating a new database with new classification of manufacturer only mentioning the first 5 and then having "Other"  
planes\_modified<-planes %>%   
 mutate(manufacturer=case\_when(manufacturer=="AIRBUS INDUSTRIE" ~ "AIRBUS",  
 manufacturer %in% c("MCDONNELL DOUGLAS AIRCRAFT CO","MCDONNELL DOUGLAS AIRCRAFT CO","MCDONNELL DOUGLAS CORPORATION")~"MCDONNELL DOUGLAS",  
 .default=manufacturer  
   
 )) %>%   
 mutate(new\_manufacturer=case\_when(  
 manufacturer %in% topmanufacturer ~ manufacturer,  
 .default="Other"  
 ))  
  
  
#Groupping the newly created data base  
groupped\_manufacturer<- planes\_modified %>%   
 filter(!is.na(year))%>%   
 group\_by(new\_manufacturer,year)%>%   
 count(new\_manufacturer,year)  
  
#Plotting by year the new manufacturers and count  
ggplot(groupped\_manufacturer,aes(x=year,y=n,group=new\_manufacturer))+geom\_line(aes(color=new\_manufacturer))



## Problem 6: Use the flights and planes tables to answer the following questions:

- What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
- How many airplanes that flew from New York City are included in the planes table?

problem6<-flights %>%   
 filter(year==2013) %>%   
 left\_join(planes,by="tailnum") %>%   
 arrange(year.y)  
  
problem6$tailnum[1]

[1] "N381AA"

problem6$year.y[1]

[1] 1956

problem6$dest[1]

[1] "SFO"

problem6$origin[1]

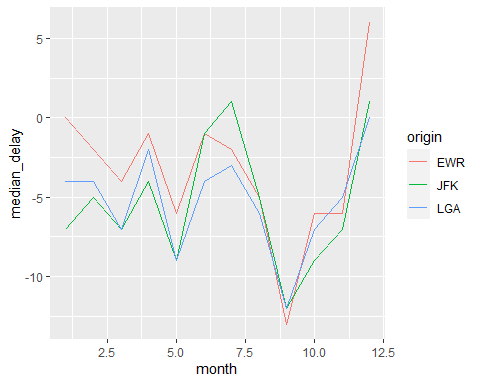
[1] "JFK"

# the N381AA is the oldest plane built in 1956 and travelling from JFK to SFO  
  
#Anti\_joining the planes table to the flights  
problem6\_2<-planes %>%   
 anti\_join(flights,by="tailnum")  
view(problem6\_2)  
  
#Since the database is empty, and the "flights" database had only flights from NY, one may conclude that all of the planes in the "planes" table came from New York.

## Problem 7: Use the nycflights13 to answer the following questions:

- What is the median arrival delay on a month-by-month basis in each airport?  
- For each airline, plot the median arrival delay for each month and origin airport.

#Creating the data set groupped by month, origin and median delay  
problem7<-flights %>%   
 filter(year==2013) %>%   
 group\_by(month,origin)%>%   
 summarise(median\_delay=median(arr\_delay,na.rm=TRUE),.groups="drop")  
  
  
#Creatting the plot with the above information- it seems that August is the month with less delays and december with most delays. The graph is very volatile within the months  
ggplot(problem7,aes(x=month,y=median\_delay,group=origin))+geom\_line(aes(color=origin))



## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United Air Lines Inc. not UA, the count (number) of times it flew to SFO, and the percent of the trips that that particular airline flew to SFO.

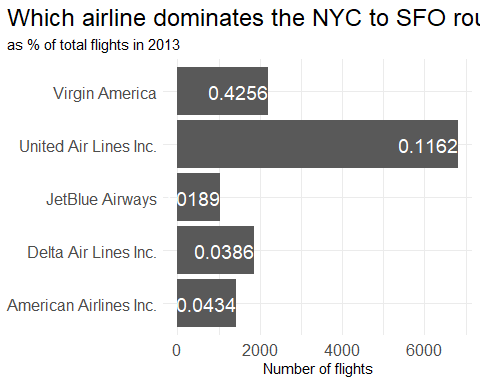
fly\_into\_sfo<- flights %>%  
 left\_join(airlines,by="carrier") %>%   
 select(name,dest) %>%  
 group\_by(name,dest) %>%   
 summarise(count=n()) %>%   
 mutate(percent=round((count/sum(count)),4))%>%   
 filter(dest=="SFO")

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

view(fly\_into\_sfo)

And here is some bonus ggplot code to plot your dataframe

fly\_into\_sfo %>%   
   
 # sort 'name' of airline by the numbers it times to flew to SFO  
 mutate(name = fct\_reorder(name, count)) %>%   
   
 ggplot() +  
   
 aes(x = count,   
 y = name) +  
   
 # a simple bar/column plot  
 geom\_col() +  
   
 # add labels, so each bar shows the % of total flights   
 geom\_text(aes(label = percent),  
 hjust = 1,   
 colour = "white",   
 size = 5)+  
   
 # add labels to help our audience   
 labs(title="Which airline dominates the NYC to SFO route?",   
 subtitle = "as % of total flights in 2013",  
 x= "Number of flights",  
 y= NULL) +  
   
 theme\_minimal() +   
   
 # change the theme-- i just googled those , but you can use the ggThemeAssist add-in  
 # https://cran.r-project.org/web/packages/ggThemeAssist/index.html  
   
 theme(#  
 # so title is left-aligned  
 plot.title.position = "plot",  
   
 # text in axes appears larger   
 axis.text = element\_text(size=12),  
   
 # title text is bigger  
 plot.title = element\_text(size=18)  
 ) +  
  
 # add one final layer of NULL, so if you comment out any lines  
 # you never end up with a hanging `+` that awaits another ggplot layer  
 NULL



## Problem 9: Let’s take a look at cancellations of flights to SFO. We create a new dataframe cancellations as follows

cancellations <- flights %>%   
   
 # just filter for destination == 'SFO'  
 filter(dest == 'SFO') %>%   
   
 # a cancelled flight is one with no `dep\_time`   
 filter(is.na(dep\_time))

I want you to think how we would organise our data manipulation to create the following plot. No need to write the code, just explain in words how you would go about it.



## Problem 10: On your own – Hollywood Age Gap

The website <https://hollywoodagegap.com> is a record of *THE AGE DIFFERENCE IN YEARS BETWEEN MOVIE LOVE INTERESTS*. This is an informational site showing the age gap between movie love interests and the data follows certain rules:

* The two (or more) actors play actual love interests (not just friends, coworkers, or some other non-romantic type of relationship)
* The youngest of the two actors is at least 17 years old
* No animated characters

The age gaps dataset includes “gender” columns, which always contain the values “man” or “woman”. These values appear to indicate how the characters in each film identify and some of these values do not match how the actor identifies. We apologize if any characters are misgendered in the data!

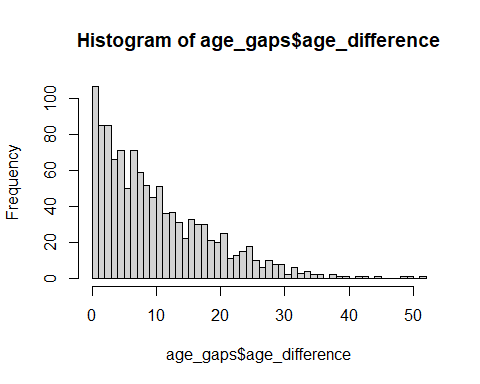
The following is a data dictionary of the variables used

| variable | class | description |
| --- | --- | --- |
| movie\_name | character | Name of the film |
| release\_year | integer | Release year |
| director | character | Director of the film |
| age\_difference | integer | Age difference between the characters in whole years |
| couple\_number | integer | An identifier for the couple in case multiple couples are listed for this film |
| actor\_1\_name | character | The name of the older actor in this couple |
| actor\_2\_name | character | The name of the younger actor in this couple |
| character\_1\_gender | character | The gender of the older character, as identified by the person who submitted the data for this couple |
| character\_2\_gender | character | The gender of the younger character, as identified by the person who submitted the data for this couple |
| actor\_1\_birthdate | date | The birthdate of the older member of the couple |
| actor\_2\_birthdate | date | The birthdate of the younger member of the couple |
| actor\_1\_age | integer | The age of the older actor when the film was released |
| actor\_2\_age | integer | The age of the younger actor when the film was released |

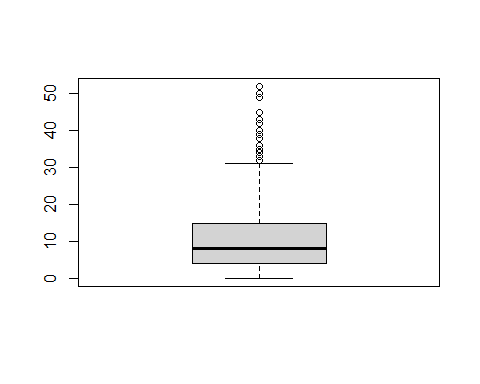
age\_gaps <- readr::read\_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2023/2023-02-14/age\_gaps.csv')

Rows: 1155 Columns: 13  
-- Column specification --------------------------------------------------------  
Delimiter: ","  
chr (6): movie\_name, director, actor\_1\_name, actor\_2\_name, character\_1\_gend...  
dbl (5): release\_year, age\_difference, couple\_number, actor\_1\_age, actor\_2\_age  
date (2): actor\_1\_birthdate, actor\_2\_birthdate  
  
i Use `spec()` to retrieve the full column specification for this data.  
i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

##Distribution of "Age\_difference"  
hist(age\_gaps$age\_difference,breaks=50) #It is a very right-skewed distribution, meaning that most of the age differences are low



boxplot(age\_gaps$age\_difference) #Additionally we can se the boxplot, IQ Range and the many outliers that exist with this difference



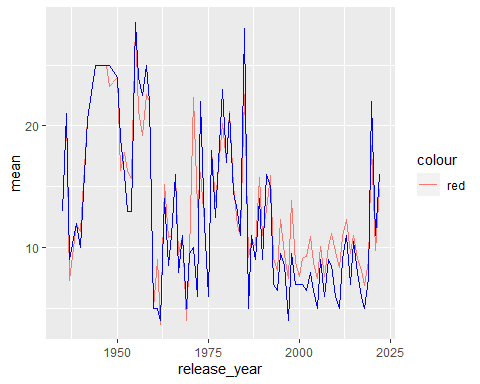
summary(age\_gaps$age\_difference) #one can see that the mean age difference is around 10 years, and half of the actors have at least 8 years of difference. The Maximum is 52 years which looks very like an outlier, as the 3rd quartile is just 15

Min. 1st Qu. Median Mean 3rd Qu. Max.   
 0.00 4.00 8.00 10.42 15.00 52.00

##Rule of half plus seven  
  
age\_gaps\_new<-age\_gaps %>%   
 mutate(rule=case\_when(  
 (actor\_2\_age>(actor\_1\_age/2)+7&(actor\_1\_age-7)\*2>actor\_2\_age)~TRUE,  
 .default=FALSE  
   
 ))   
#Creating a table by groupping and calculating relative frequency  
test<-age\_gaps\_new%>% group\_by(rule) %>%   
 summarise(count=n()) %>%   
 mutate(freq=count/sum(count))  
view(test)  
  
#There are 795 actors who follow the rule (69% approximately)  
  
##Movie with most love interests  
movie\_love<-age\_gaps %>%  
 group\_by(movie\_name) %>%   
 summarise(count=n()) %>%   
 arrange(desc(count))  
  
#Movie with most love interests is called "Love actually" with 7 love interests  
  
## Actors with most love interests  
#Creating a table only with actor 1 and actor 2  
actor1<-age\_gaps %>% select(actor=actor\_1\_name)  
actor2<-age\_gaps %>% select(actor=actor\_2\_name)  
#binding both tables into a single one  
list\_actors<-bind\_rows(actor1,actor2)  
  
#Groupping by name and arranging it descending order  
list\_actors %>% group\_by(actor) %>% summarise(count=n()) %>% arrange(desc(count))

# A tibble: 1,031 x 2  
 actor count  
 <chr> <int>  
 1 Keanu Reeves 27  
 2 Adam Sandler 20  
 3 Leonardo DiCaprio 17  
 4 Roger Moore 17  
 5 Sean Connery 17  
 6 Keira Knightley 14  
 7 Pierce Brosnan 14  
 8 Harrison Ford 13  
 9 Reese Witherspoon 13  
10 Scarlett Johansson 13  
# i 1,021 more rows

#Answer- Keanu Reeves is the actor that has had more love interests  
  
##Mean/Median age by year  
mean\_median<-age\_gaps %>%   
 group\_by(release\_year) %>%   
 summarise(mean=round(mean(age\_difference),1),median=round(median(age\_difference),1))  
  
ggplot(mean\_median,aes(x=release\_year))+geom\_line(aes(y=mean,color="red"))+geom\_line(aes(y=median),color="blue")



#Conclusion: the mean/median ages have not been static throughout the years  
  
  
#LGBTQ romances  
age\_gaps\_new<-age\_gaps\_new %>%   
 mutate(orientation=case\_when(  
 (character\_1\_gender=="man" & character\_1\_gender==character\_2\_gender)~"gay",  
 (character\_1\_gender=="woman" & character\_1\_gender==character\_2\_gender)~"lesbian",  
 .default="heterosexual"  
 ))  
  
summary\_orientation<-age\_gaps\_new %>%   
 group\_by(orientation) %>%   
 summarise(count=n()) %>%   
 mutate(frequency=count/sum(count))  
  
view(summary\_orientation)  
  
  
"Straight couples are:98% of total, gay relationships are 1.04% of total and lesbian are 0.952% of total"

[1] "Straight couples are:98% of total, gay relationships are 1.04% of total and lesbian are 0.952% of total"

How would you explore this data set? Here are some ideas of tables/ graphs to help you with your analysis

* How is age\_difference distributed? What’s the ‘typical’ age\_difference in movies?
* The half plus seven\ rule. Large age disparities in relationships carry certain stigmas. One popular rule of thumb is the [half-your-age-plus-seven](https://en.wikipedia.org/wiki/Age_disparity_in_sexual_relationships#The_.22half-your-age-plus-seven.22_rule) rule. This rule states you should never date anyone under half your age plus seven, establishing a minimum boundary on whom one can date. In order for a dating relationship to be acceptable under this rule, your partner’s age must be:

$$\frac{\text{Your age}}{2} + 7 \< \text{Partner Age} \< (\text{Your age} - 7) \\* 2$$

How frequently does this rule apply in this dataset?

* Which movie has the greatest number of love interests?
* Which actors/ actresses have the greatest number of love interests in this dataset?
* Is the mean/median age difference staying constant over the years (1935 - 2022)?
* How frequently does Hollywood depict same-gender love interests?

# 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. Render 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: Only me
* Approximately how much time did you spend on this problem set: did not count, but at least 4 hours
* What, if anything, gave you the most trouble: groupping by 2 variables

**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.