Homerwork 1

GUIDO BOZZANO

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)  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to the arr\_delay variable of >120 minutes  
flights %>%   
 filter(arr\_delay >= 120)

# A tibble: 10,200 × 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 1 811 630 101 1047 830  
 2 2013 1 1 848 1835 853 1001 1950  
 3 2013 1 1 957 733 144 1056 853  
 4 2013 1 1 1114 900 134 1447 1222  
 5 2013 1 1 1505 1310 115 1638 1431  
 6 2013 1 1 1525 1340 105 1831 1626  
 7 2013 1 1 1549 1445 64 1912 1656  
 8 2013 1 1 1558 1359 119 1718 1515  
 9 2013 1 1 1732 1630 62 2028 1825  
10 2013 1 1 1803 1620 103 2008 1750  
# ℹ 10,190 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)  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to the dest variable to look for destination Houston (IAH or HOU)  
 filter(flights, dest == "IAH"| dest == "HOU")

# A tibble: 9,313 × 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`)  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to the carrier variable to look for flights from United, American and Delta  
 filter(flights, carrier == "UA"| carrier == "AA" | carrier == "DL")

# A tibble: 139,504 × 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)  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to the carrier month variable to look for flights departed in the summer months  
 filter(flights, month == 7 | month == 8 | month == 9)

# A tibble: 86,326 × 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  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to the arr\_delay to indicate more than 120 minutes of delay but the departure delay is below 0  
 filter(flights, arr\_delay >=120 , dep\_delay <=0)

# A tibble: 29 × 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 27 1419 1420 -1 1754 1550  
 2 2013 10 7 1350 1350 0 1736 1526  
 3 2013 10 7 1357 1359 -2 1858 1654  
 4 2013 10 16 657 700 -3 1258 1056  
 5 2013 11 1 658 700 -2 1329 1015  
 6 2013 3 18 1844 1847 -3 39 2219  
 7 2013 4 17 1635 1640 -5 2049 1845  
 8 2013 4 18 558 600 -2 1149 850  
 9 2013 4 18 655 700 -5 1213 950  
10 2013 5 22 1827 1830 -3 2217 2010  
# ℹ 19 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 delayed by at least an hour, but made up over 30 minutes in flight  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to the arr\_delay to indicate that they arrived less than 30 minutes and the departure delay was at least 60 minutes  
 filter(flights, arr\_delay <30 , dep\_delay >=60)

# A tibble: 206 × 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 3 1850 1745 65 2148 2120  
 2 2013 1 3 1950 1845 65 2228 2227  
 3 2013 1 3 2015 1915 60 2135 2111  
 4 2013 1 6 1019 900 79 1558 1530  
 5 2013 1 7 1543 1430 73 1758 1735  
 6 2013 1 11 1020 920 60 1311 1245  
 7 2013 1 12 1706 1600 66 1949 1927  
 8 2013 1 12 1953 1845 68 2154 2137  
 9 2013 1 19 1456 1355 61 1636 1615  
10 2013 1 21 1531 1430 61 1843 1815  
# ℹ 196 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>

## 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?  
#For this problem I begin using the flights table included in the nycflights13 and apply a filter to understand how many flights were cancelled (is.na(dep\_time)). Then, I group by month and then I count per month and mutate a new column to understand the proportion of each month  
flights %>%  
 filter(is.na(dep\_time)) %>%   
 group\_by (month) %>%   
 summarise(count = n()) %>%   
 mutate(prop = count/sum(count))

# A tibble: 12 × 3  
 month count prop  
 <int> <int> <dbl>  
 1 1 521 0.0631  
 2 2 1261 0.153   
 3 3 861 0.104   
 4 4 668 0.0809  
 5 5 563 0.0682  
 6 6 1009 0.122   
 7 7 940 0.114   
 8 8 486 0.0589  
 9 9 452 0.0548  
10 10 236 0.0286  
11 11 233 0.0282  
12 12 1025 0.124

#Answer: Highest is February with 15.2% and lowest is November with 2.8%

## 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.

#For this problem I create a new table called mostflights based on the flights table included in the nycflights13 and apply a filter to the year equal to 2013. Also, I apply a filter to bring me the results of those observations in which I have a tailnum.  
#Afterwards, I group by tailnum and origin and count the number of flights (When I ran the code without the .groups= "drop" I had a warning, thus I included that piece of code inside the summarise). Last, I arrange in descending order  
#This table I created counts me the flights by tailnum and origin  
mostflights <- flights %>%  
 filter(year == 2013) %>%   
 filter(!is.na(tailnum)) %>%   
 group\_by(tailnum,origin) %>%   
 summarise(total\_flights=n(), .groups = "drop") %>%   
 arrange(desc(total\_flights))  
mostflights

# A tibble: 7,941 × 3  
 tailnum origin total\_flights  
 <chr> <chr> <int>  
 1 N725MQ LGA 567  
 2 N722MQ LGA 507  
 3 N723MQ LGA 502  
 4 N711MQ LGA 484  
 5 N713MQ LGA 478  
 6 N735MQ LGA 396  
 7 N328AA JFK 393  
 8 N258JB JFK 391  
 9 N338AA JFK 388  
10 N327AA JFK 387  
# ℹ 7,931 more rows

#Now I do a left\_join between the table defined above - mostflights - and planes by "tailnum". Also, I make sure to exclude the N/A matches by applying na\_matches = "never" in the left\_join.  
#Then I apply the seats > 50 filter to understand which planes have the most number of flights and have seats > 50. I save this under a new object called answer  
answer <- left\_join(mostflights, planes, by="tailnum", na\_matches = "never") %>%   
 filter(seats>50)  
answer

# A tibble: 6,202 × 11  
 tailnum origin total\_flights year type manufacturer model engines seats  
 <chr> <chr> <int> <int> <chr> <chr> <chr> <int> <int>  
 1 N328AA JFK 393 1986 Fixed wi… BOEING 767-… 2 255  
 2 N338AA JFK 388 1987 Fixed wi… BOEING 767-… 2 255  
 3 N327AA JFK 387 1986 Fixed wi… BOEING 767-… 2 255  
 4 N335AA JFK 385 1987 Fixed wi… BOEING 767-… 2 255  
 5 N323AA JFK 357 1986 Fixed wi… BOEING 767-… 2 255  
 6 N319AA JFK 354 1985 Fixed wi… BOEING 767-… 2 255  
 7 N336AA JFK 353 1987 Fixed wi… BOEING 767-… 2 255  
 8 N329AA JFK 344 1987 Fixed wi… BOEING 767-… 2 255  
 9 N324AA JFK 328 1986 Fixed wi… BOEING 767-… 2 255  
10 N332AA JFK 328 1987 Fixed wi… BOEING 767-… 2 255  
# ℹ 6,192 more rows  
# ℹ 2 more variables: speed <int>, engine <chr>

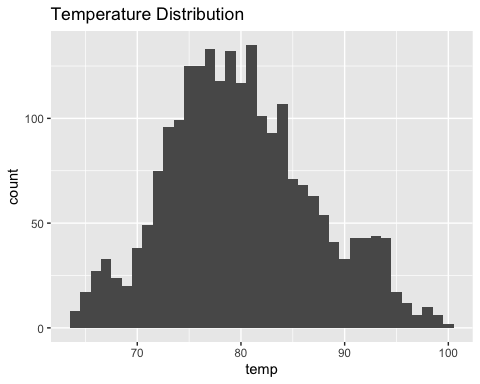
#Finally, I look for the tailnum that the answer table give me as the most flights and apply that information to filter the flights table.  
#I create a new object called table\_answer where I apply this filter and count the number of flights the "N328AA" tailnum did by destination  
table\_answer <- flights %>%   
 filter(tailnum=="N328AA") %>%   
 group\_by(tailnum,dest) %>%   
 summarise(total\_flights=n(), .groups="drop") %>%   
 arrange(desc(total\_flights))  
table\_answer

# A tibble: 6 × 3  
 tailnum dest total\_flights  
 <chr> <chr> <int>  
1 N328AA LAX 313  
2 N328AA SFO 52  
3 N328AA MIA 25  
4 N328AA BOS 1  
5 N328AA MCO 1  
6 N328AA SJU 1

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

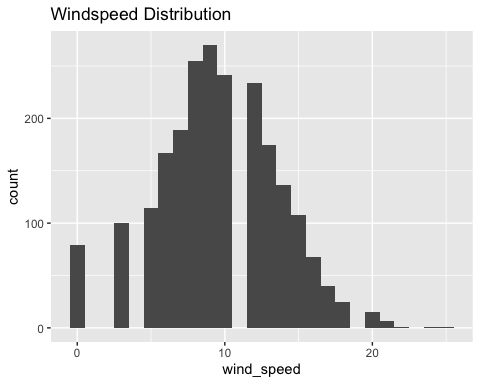
- What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
- What is the relationship between `dewp` and `humid`?  
- What is the relationship between `precip` and `visib`?

#Based on the weather table, I create a new object called july13 which filters the weather table by year == 2013 and month == 7  
july13 <- weather %>%  
 filter(year==2013,month==7)  
   
#To understand the temperature distribution I do a histogram using ggplot and geom\_histogram and the july13 table defined above and map the x variable as "temp". I set the title using labels (labs function)  
  
ggplot(july13, aes(x=temp)) +  
 geom\_histogram(binwidth = 1) +  
 labs(title = "Temperature Distribution")



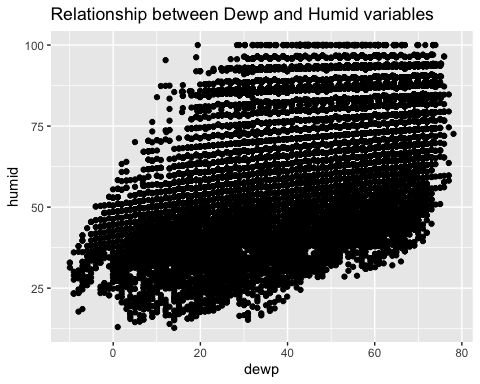
#To understand the wind speed distribution I do a histogram using ggplot and geom\_histogram and the july13 table defined above and map the x variable as "wind\_speed". I set the title using labels (labs function)  
  
ggplot(july13, aes(x=wind\_speed)) +  
 geom\_histogram(binwidth = 1) +  
 labs(title = "Windspeed Distribution")

Warning: Removed 2 rows containing non-finite values (`stat\_bin()`).



#Answer 2: Based on the wind speed distribution I can observe outliers when the wind speed is exactly 11 as there are no observations with that information. The same happens when the wind speed is 1 or 2.  
  
#To understand relationships between two variables I plot a scatter plot using geom\_point and also calculate the correlation between the two variables for a clear indication of the relationship. To do so, I use the table july13 and apply a summarise with the cor function to the variables dewp and humid  
  
ggplot(weather, aes(x=dewp, y=humid)) +  
 geom\_point() +  
 labs(title = "Relationship between Dewp and Humid variables")

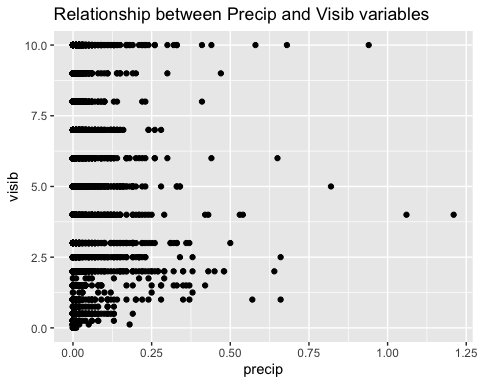
Warning: Removed 1 rows containing missing values (`geom\_point()`).



weather %>%   
 summarise(corr\_dh=cor(dewp,humid, use="complete.obs"))

# A tibble: 1 × 1  
 corr\_dh  
 <dbl>  
1 0.512

#Answer 3: There seems to be a positive relationship between humidity and dew point temperature since the coefficient of correlation is 0.52. Therefore, as the humidity in the air increases the dew point temperature also increases  
  
#To understand relationships between two variables I plot a scatter plot using geom\_point and also calculate the correlation between the two variables for a clear indication of the relationship. To do so, I use the table july13 and apply a summarise with the cor function to the variables precip and visib  
  
ggplot(weather, aes(x=precip, y=visib)) +  
 geom\_point() +  
 labs(title = "Relationship between Precip and Visib variables")



weather %>%   
 summarise(corr\_pv=cor(precip,visib, use="complete.obs"))

# A tibble: 1 × 1  
 corr\_pv  
 <dbl>  
1 -0.320

#Answer 4:On the other hand the relationship between visibility and precipitation is not that clear as the coefficient of correlation is -0.32. This indicates that there is not clear linear relationship between the precip and visib variables

## 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.)

#To answer the first question I begin by using the planes table, then group it by year and filter which years have missing values (i.e: N/As). Then I count them to find how many there are  
planes %>%   
 group\_by(year) %>%   
 filter(is.na(year)) %>%   
 summarise(total\_na=n())

# A tibble: 1 × 2  
 year total\_na  
 <int> <int>  
1 NA 70

#Answer: There are 70 planes with missing date of manufacture  
  
#For the second question I also use the planes table and group them by manufacturer. Then, I create a new variable called total\_manuf that counts the amount of manufacturers there are in the table and arrange them in descending order. Last, I select the top 5 total manufacturers   
planes %>%   
 group\_by(manufacturer) %>%   
 summarise(total\_manuf=n()) %>%   
 arrange(desc(total\_manuf)) %>%   
 top\_n(5,total\_manuf)

# A tibble: 5 × 2  
 manufacturer total\_manuf  
 <chr> <int>  
1 BOEING 1630  
2 AIRBUS INDUSTRIE 400  
3 BOMBARDIER INC 368  
4 AIRBUS 336  
5 EMBRAER 299

#To answer the third question I begin by creating a table called flights\_subset that contains the origin NYC and year equal to 2013.  
flights\_subset <- filter(flights, year == 2013, origin %in% c("JFK", "LGA", "EWR"))  
  
#Then I do a left join between the table before and planes by tailnum. Then I mutate the manufacturer column and apply a case when to collapse the multiple AIRBUS manufacturers into 1 and then case when to group the others  
answer <- left\_join(flights\_subset, planes, by="tailnum", na\_matches = "never") %>%   
 mutate(manufacturer = case\_when(  
 manufacturer %in% c("AIRBUS INDUSTRIE", "AIRBUS") ~ "AIRBUS", # Keep Boeing and Airbus as they are  
 manufacturer %in% c("BOEING", "EMBRAER", "BOMBARDIER INC") ~ manufacturer,  
 TRUE ~ "Other" # Collapse other manufacturers into "Other" category  
 ))  
answer

# A tibble: 336,776 × 27  
 year.x 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 336,766 more rows  
# ℹ 19 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>, year.y <int>, type <chr>,  
# manufacturer <chr>, model <chr>, engines <int>, seats <int>, speed <int>,  
# engine <chr>

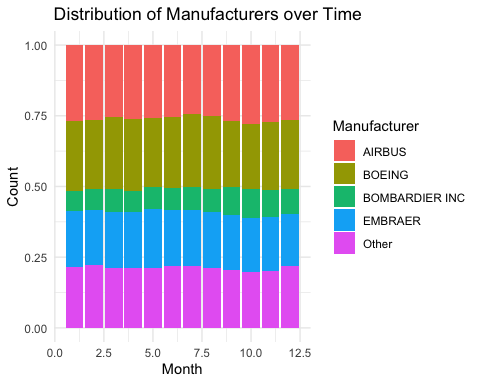
#Then I define a manufacturers over time table and group it by month and manufacturer and count  
manufacturers\_over\_time <- answer %>%  
 group\_by(month,manufacturer) %>%  
 summarise(count = n())

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

manufacturers\_over\_time

# A tibble: 60 × 3  
# Groups: month [12]  
 month manufacturer count  
 <int> <chr> <int>  
 1 1 AIRBUS 7283  
 2 1 BOEING 6623  
 3 1 BOMBARDIER INC 1925  
 4 1 EMBRAER 5364  
 5 1 Other 5809  
 6 2 AIRBUS 6654  
 7 2 BOEING 6048  
 8 2 BOMBARDIER INC 1809  
 9 2 EMBRAER 4908  
10 2 Other 5532  
# ℹ 50 more rows

#Last I plot the distribution of manufacturers over time for the year 2013 using ggplot  
ggplot(manufacturers\_over\_time, aes(x = month, y = count, fill = manufacturer)) + #Set up x as month and y as the count where the colour fill is the manufacturer  
 geom\_bar(stat = "identity", position = "fill") + #Set up the chart as a bar chart  
 labs(title = "Distribution of Manufacturers over Time",  
 x = "Month",  
 y = "Count",  
 fill = "Manufacturer") +  
 theme\_minimal()



#From the chart I can conclude that the distribution of manufacturers hasn't changed much in the year 2013 for planes departing NYC

## 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?

#First I filter the flights from NYC airports in 2013  
nyc\_flights\_2013 <- flights %>%  
 filter(year == 2013)  
  
#Then I make sure to join with planes table to get the aircraft details  
flights\_with\_planes <- left\_join(nyc\_flights\_2013, planes %>%   
 rename(year\_plane = year), by = "tailnum")  
  
#Find the oldest plane  
oldest\_plane <- flights\_with\_planes %>%  
 filter(!is.na(year\_plane)) %>%  
 arrange(year\_plane) %>%  
 select(tailnum, year\_plane) %>%  
 slice(1)  
  
oldest\_plane #Display the oldest plane

# A tibble: 1 × 2  
 tailnum year\_plane  
 <chr> <int>  
1 N381AA 1956

flightsnyc <- flights %>% #Define a flights from NYC table  
 filter(origin %in% c("JFK", "LGA", "EWR")) #Only contains NYC origins  
  
answer <- flightsnyc %>%   
 inner\_join(planes, by = "tailnum") %>% #Create an innerjoin with the planes table by tailnum  
 distinct(tailnum) %>%   
 summarise(count=n())  
  
answer #Display the result

# A tibble: 1 × 1  
 count  
 <int>  
1 3322

#Answer A: N381AA - 1956  
#Answer B: 3322

## 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.

#First I define a new table called arrivdelay that groups by month and destination, then I calculate the median arrival delay excluding N/As  
arrivdelay <- flights %>%   
 group\_by(month,dest) %>%   
 summarise(med\_arrdelay = median(arr\_delay, na.rm = TRUE))

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

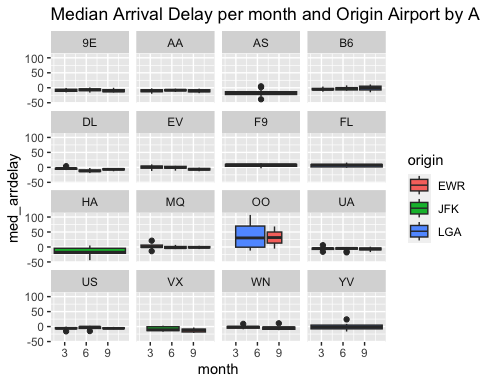
arrivdelay #Display the results

# A tibble: 1,113 × 3  
# Groups: month [12]  
 month dest med\_arrdelay  
 <int> <chr> <dbl>  
 1 1 ALB 6   
 2 1 ATL -2   
 3 1 AUS -2   
 4 1 AVL 23.5  
 5 1 BDL -10   
 6 1 BHM -11   
 7 1 BNA 1   
 8 1 BOS -10   
 9 1 BQN -5   
10 1 BTV -6   
# ℹ 1,103 more rows

#First I define a new table called arrivdelay that groups by month, origin and carrier, then I calculate the median arrival delay excluding N/As. After this I use ggplot to make a boxplot chart facet wrapped by carrier with the x-axis equal to month and y equal to med\_arrdelay  
arrivd <- flights %>%   
 group\_by(month,origin,carrier) %>%   
 summarise(med\_arrdelay = median(arr\_delay, na.rm = TRUE))

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

ggplot(arrivd, aes(x=month, y=med\_arrdelay, fill=origin)) +  
 geom\_boxplot() +  
 facet\_wrap(vars(carrier)) +  
 labs(title="Median Arrival Delay per month and Origin Airport by Airline")



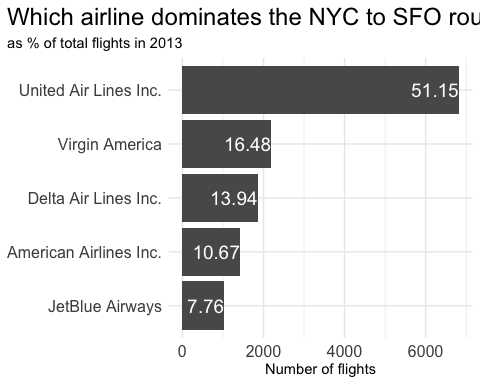
## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United 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.

#First, I carry out a leftjoin between the flights and airlines table by tailnum. I filter destination SFO and group by name. Then, I count and mutate a new column to find the percent trips that each airline flew to SFO  
fly\_into\_sfo <- left\_join(flights,airlines, by="carrier") %>%   
 filter(dest=="SFO") %>%   
 group\_by (name) %>%   
 summarise(count = n()) %>%   
 mutate(percent = round(count/sum(count),4)\*100)  
   
fly\_into\_sfo #Display the results

# A tibble: 5 × 3  
 name count percent  
 <chr> <int> <dbl>  
1 American Airlines Inc. 1422 10.7   
2 Delta Air Lines Inc. 1858 13.9   
3 JetBlue Airways 1035 7.76  
4 United Air Lines Inc. 6819 51.2   
5 Virgin America 2197 16.5

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.



To make a graph that looks like the one above I believe I should create a ggplot with a geom\_bar that facet wraps by origin and carrier. The sub plot per each wrap would be number of cancellations (y-axis) by month (x-axis)

## 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!

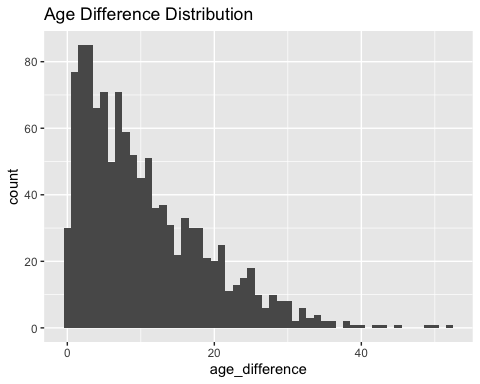
The following is a data dictionary of the variables used

| 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.

#To understand the age difference distribution I use age\_gaps to make a graph using ggplot. I make sure to apply geom\_histogram to make a histogram to understand the distribution  
ggplot(age\_gaps, aes(x=age\_difference)) +  
 geom\_histogram(binwidth = 1) +  
 labs(title = "Age Difference Distribution")



#Also, to understand what's the typical median age\_difference I calculate the median value for the age\_difference  
age\_gaps %>%   
 summarise(median\_agedif=median(age\_difference))

# A tibble: 1 × 1  
 median\_agedif  
 <dbl>  
1 8

#Answer A: From the chart I can conclude that the age difference is particularly skewed towards the lower end of the difference. In addition, from the medican calculation I know that the median age difference from the age\_gaps dataset is 8 which is in line with what the graph displays  
  
#To calculate the half plus seven rule I define a new table called agegapdata that has the upper and lower bounds mutate with the information provided by the formula  
  
#Then, I calculate the lower and upper age bounds based on the "half plus seven" rule  
agegapdata <- age\_gaps %>%  
 mutate(lower\_bound = floor(actor\_1\_age/2) + 7,  
 upper\_bound = (actor\_1\_age - 7) \* 2)  
  
#Afterwards, I make sure to count the number of actor/actress pairs that satisfy the "half plus seven" rule  
rule\_applies <- agegapdata %>%  
 filter(actor\_2\_age >= lower\_bound, actor\_2\_age <= upper\_bound) %>%  
 tally()  
  
#Last, I calculate the percentage of pairs that satisfy the rule  
percentage\_rule\_applies <- rule\_applies$n / nrow(agegapdata) \* 100  
  
percentage\_rule\_applies #To show the results

[1] 74.02597

#Answer is around 74% (74.02597)  
  
#To understand how many couples each movie has I make a group by movie\_name to the age\_gaps table I count them and arrange them descendingly  
age\_gaps %>%   
 group\_by(movie\_name) %>%   
 summarise(count=n()) %>%   
 arrange(desc(count))

# A tibble: 830 × 2  
 movie\_name count  
 <chr> <int>  
 1 Love Actually 7  
 2 The Family Stone 6  
 3 A View to a Kill 5  
 4 He's Just Not That Into You 5  
 5 Mona Lisa Smile 5  
 6 A Star Is Born 4  
 7 American Pie 4  
 8 Boogie Nights 4  
 9 Closer 4  
10 Pushing Tin 4  
# ℹ 820 more rows

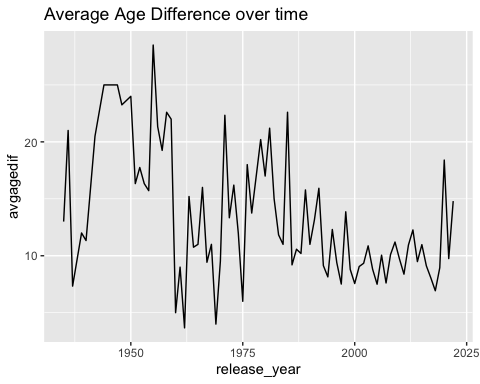
#Answer is Love Actually with 7  
  
#To understand which actors have the greatest number of love interests in each movie I summarise by actor\_1\_name and actor\_2\_name and then I define a new variable called loveint to count the distinct number of couples. Last I arrange them by descending order  
age\_gaps %>%   
 group\_by(actor\_1\_name) %>%   
 summarise(lovint=n\_distinct(couple\_number)) %>%   
 arrange(desc(lovint))

# A tibble: 567 × 2  
 actor\_1\_name lovint  
 <chr> <int>  
 1 Pierce Brosnan 4  
 2 Roger Moore 4  
 3 Bradley Cooper 3  
 4 Christian Bale 3  
 5 Colin Firth 3  
 6 Dermot Mulroney 3  
 7 Heath Ledger 3  
 8 Jesse Eisenberg 3  
 9 John Cusack 3  
10 Julia Roberts 3  
# ℹ 557 more rows

age\_gaps %>%   
 group\_by(actor\_2\_name) %>%   
 summarise(lovint=n\_distinct(couple\_number)) %>%   
 arrange(desc(lovint))

# A tibble: 647 × 2  
 actor\_2\_name lovint  
 <chr> <int>  
 1 Diane Keaton 4  
 2 Keira Knightley 4  
 3 Ali Wong 3  
 4 Amy Adams 3  
 5 Angelina Jolie 3  
 6 Annette Bening 3  
 7 Blake Lively 3  
 8 Cameron Diaz 3  
 9 Cate Blanchett 3  
10 Helena Bonham Carter 3  
# ℹ 637 more rows

#Answer Pierce Brosnan for the male actors and Diane Keaton for the female actors  
  
#To understand whether the age difference varies over time I create a new table called agedifplot which groups the data by release\_year and calculate the mean in agedifference. Afterwards I plot the information in a line chart using ggplot + geom\_line.  
#From the graph I can conclude the age difference has varied significantly over time and does not stay constant  
agedifplot <- age\_gaps %>%   
 group\_by(release\_year) %>%   
 summarise(avgagedif=mean(age\_difference))   
  
ggplot(agedifplot,aes(x=release\_year,y=avgagedif)) +  
 geom\_line() +  
 labs(title = "Average Age Difference over time")



#To find the number of movies with same gender love relationships I create a table called same\_gender and filter the information by how many characters gender are equal between 1 and 2 actors. Last, I count the information  
same\_gender <- age\_gaps %>%   
 filter(character\_1\_gender==character\_2\_gender) %>%   
 summarise(count=n())  
  
same\_gender

# A tibble: 1 × 1  
 count  
 <int>  
1 23

#Answer: There are 23 movies that display same gender love relationships in the age\_gaps dataset

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:

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: IGNACIO GAING
* Approximately how much time did you spend on this problem set: 10 hours or more
* What, if anything, gave you the most trouble: From problem 3 onwards, specially problem 10

**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.