Homerwork 1

VICTOR RIOS FARIA

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\_delay>=120) #filter flights with arr\_delay greater than or equal to 120 miutes

# 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)  
  
flights %>%  
 filter(dest == "IAH" | dest == "HOU") #filter flights that flew to either IAH or 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`)  
  
flights %>%  
 filter(carrier == "UA" | carrier == "AA" | carrier =="DL") #filter flights operated by the assigned carriers

# 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)  
  
flights %>%  
 filter(month == 7| month == 8 | month == 9) #filter flights that departed in the months of July, August or September

# 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  
  
flights %>%  
 filter(arr\_delay>120) %>% #filter flights that arrived more than 2h late and pass this datastream through  
 filter(dep\_delay<=0) #filter flights that didn't leave late among those that were more than 2 hours late

# 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  
  
flights %>%  
 filter(dep\_delay>=60) %>% #filter flights that departed at least 1h late  
 filter(dep\_delay-arr\_delay >= 30) #filter flights that made up over 30 minutes in flight

# A tibble: 2,074 × 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 1716 1545 91 2140 2039  
 2 2013 1 1 2205 1720 285 46 2040  
 3 2013 1 1 2326 2130 116 131 18  
 4 2013 1 3 1503 1221 162 1803 1555  
 5 2013 1 3 1821 1530 171 2131 1910  
 6 2013 1 3 1839 1700 99 2056 1950  
 7 2013 1 3 1850 1745 65 2148 2120  
 8 2013 1 3 1923 1815 68 2036 1958  
 9 2013 1 3 1941 1759 102 2246 2139  
10 2013 1 3 1950 1845 65 2228 2227  
# ℹ 2,064 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?  
  
flights %>%  
 mutate(cancelled = ifelse(is.na(dep\_time),1,0)) %>% #creates a new field that hold a value of 1 if the flight was cancelled or 0 if it was not  
 group\_by(month) %>% #group the data by the month to get aggregate statistics after  
 summarise(cancelled\_flights\_perc = mean(cancelled)) %>% #gets the average of the cancelled field, which is equal the % of flights cancelled  
 arrange(cancelled\_flights\_perc) %>% #sort the months in ascending order of % of flights cancelled  
 slice(1,12) #selects the months with the least and the most % of flights cancelled, respectivelly

# A tibble: 2 × 2  
 month cancelled\_flights\_perc  
 <int> <dbl>  
1 10 0.00817  
2 2 0.0505

## 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.

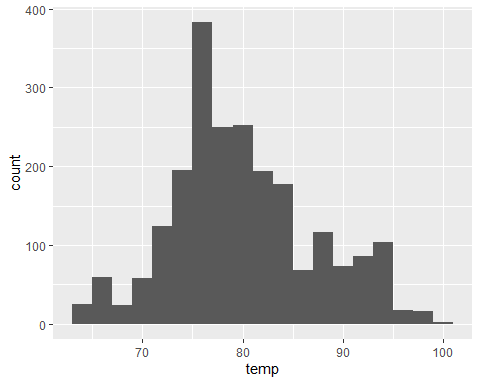
#| label: problem-3  
  
# What plane traveled the most times from New York City airports in 2013?  
  
  
flights %>%  
 filter(!is.na(tailnum) & dep\_time>0)%>% #filters flights that have an assigned tailnum and were not cancelled  
 group\_by(tailnum) %>% #groups the data by the tailnum  
 summarise(flights=n()) %>% #counts the number of flights departing from New York City per tailnum  
 arrange(desc(flights)) %>% #orders the data on descending order of number of flights  
 slice(1) %>% #gets the airplane that traveled the most times from New York City airports in 2013  
 left\_join(planes, by = "tailnum") #joins the two tables to get the airplane information from the table "planes". However, we observe that this specific airplane has no information stored in the "planes" table

# A tibble: 1 × 10  
 tailnum flights year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
1 N725MQ 546 NA <NA> <NA> <NA> NA NA NA <NA>

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

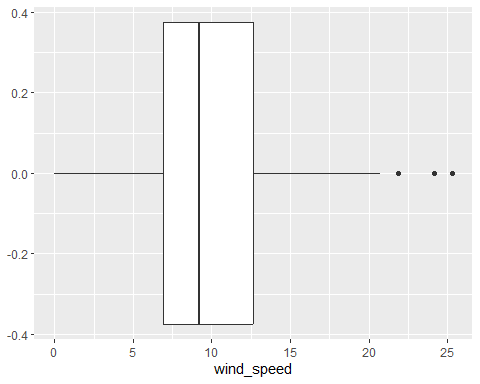
- What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
- What is the relationship between `dewp` and `humid`?  
- What is the relationship between `precip` and `visib`?

#| label: problem-4  
  
# What is the distribution of temperature (`temp`) in July 2013?   
  
weather\_july <- weather %>% filter(month == 7) # filters the data for the month of July and creates a new dataset with it  
ggplot(weather\_july, aes(x = temp)) + # creates a plot using the temperature variable  
 geom\_histogram(binwidth = 2) # sets the plot as a histogram with the given bin width, in which one can observe the temperature distribution to have its peak in the range of 75-77 degrees F



# Identify any important outliers in terms of the `wind\_speed` variable.  
ggplot(weather\_july, aes(x = wind\_speed)) + # creates a plot using the wind speed variable  
 geom\_boxplot() # sets the plot as a boxplot, from which we can observe there are 3 outliers in terms of wind speed

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

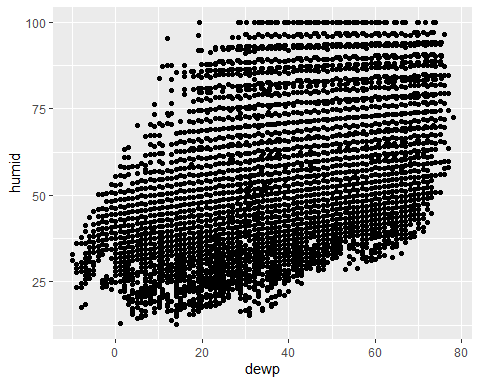


weather\_july %>%  
 arrange(desc(wind\_speed)) %>% # sort the data in descending order of wind speed  
 slice(1:3) # get the first three entries, that represent the 3 outliers found in the boxplot

# A tibble: 3 × 15  
 origin year month day hour temp dewp humid wind\_dir wind\_speed wind\_gust  
 <chr> <int> <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 JFK 2013 7 23 18 82.0 73.0 74.2 310 25.3 66.7  
2 JFK 2013 7 20 19 84.2 71.1 71.9 300 24.2 32.2  
3 JFK 2013 7 20 17 93.9 66.9 41.3 260 21.9 27.6  
# ℹ 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,  
# time\_hour <dttm>

# What is the relationship between `dewp` and `humid`?  
  
ggplot(weather,aes(x=dewp, y=humid)) +  
 geom\_point() # creates a scatter plot with dewp in the x axis and humid in the y axis, where one can observe a positive correlation between the two variables, suggesting that an increase in humidity leads to an increased dew point

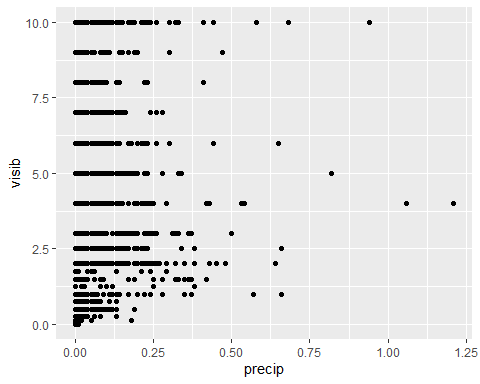
Warning: Removed 1 rows containing missing values (`geom\_point()`).



weather %>%  
 filter(!is.na(dewp) & !is.na(humid)) %>% # filters only observations with valid values for both dewp and humid  
 summarise(correlation = cor(dewp, humid)) # calculates the correlation between the two variables, confirming what has been observed in the scatter plot before

# A tibble: 1 × 1  
 correlation  
 <dbl>  
1 0.512

# What is the relationship between `precip` and `visib`?  
  
ggplot(weather,aes(x=precip, y=visib)) +  
 geom\_point() # creates a scatter plot with precip in the x axis and visib in the y axis, where one cannnot observe a clear correlation between the two variables, suggesting that preciptation is not a factor that affects visibility



weather %>%  
 filter(!is.na(precip) & !is.na(visib)) %>% # filters only observations with valid values for both precip and visib  
 summarise(correlation = cor(precip, visib)) # calculates the correlation between the two variables, pointing out a slightly negative correlation, suggesting that preciptation reduces visibility

# A tibble: 1 × 1  
 correlation  
 <dbl>  
1 -0.320

## 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.)

#| label: problem-5  
  
# How many planes have a missing date of manufacture?  
  
planes %>%  
 filter(is.na(year)) %>% # filters all observations with no information about the year of manufacture  
 summarise(planes\_missing\_date=n\_distinct(tailnum)) # counts how many different planes have the date of manufacture information missing

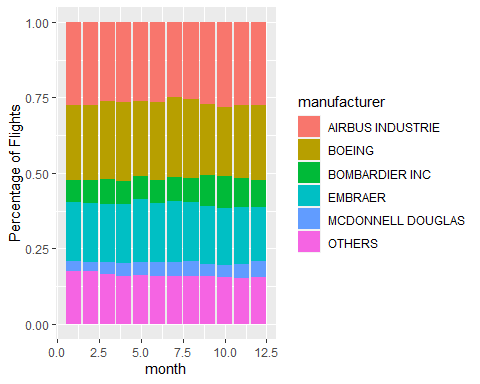
# A tibble: 1 × 1  
 planes\_missing\_date  
 <int>  
1 70

# What are the five most common manufacturers?  
  
planes\_corrected\_manufacturer <- planes %>%  
 mutate(manufacturer=if\_else(manufacturer=="AIRBUS","AIRBUS INDUSTRIE",manufacturer)) %>% # consolidates 2 different spellings for airbus into a single one  
 mutate(manufacturer=if\_else(manufacturer=="MCDONNELL DOUGLAS AIRCRAFT CO" | manufacturer=="MCDONNELL DOUGLAS CORPORATION","MCDONNELL DOUGLAS",manufacturer)) #consolidates 3 different spellings for McDonnell Douglas into a single one  
  
planes\_corrected\_manufacturer %>%  
 group\_by(manufacturer) %>% # groups the data by manufacturers  
 summarise(planesnumber=n\_distinct(tailnum)) %>% # counts how many different planes each manufactures has manufactured among those in the datastream  
 arrange(desc(planesnumber)) %>% # sorts the data in descending order of number of planes manufactured  
 slice(1:5) #gets the five most common manufacturers and the number of planes manufactured by each of them

# A tibble: 5 × 2  
 manufacturer planesnumber  
 <chr> <int>  
1 BOEING 1630  
2 AIRBUS INDUSTRIE 736  
3 BOMBARDIER INC 368  
4 EMBRAER 299  
5 MCDONNELL DOUGLAS 237

# Has the distribution of manufacturer changed over time as reflected by the airplanes flying from NYC in 2013?  
  
flightplanes <- left\_join(flights, planes\_corrected\_manufacturer, by = "tailnum") #creates a new datastream by joining flights and planes data  
  
flightplanes %>%  
 filter(!is.na(dep\_time)) %>% # filters out any cancelled flight  
 mutate(manufacturer = if\_else(manufacturer %in% c("AIRBUS INDUSTRIE", "EMBRAER", "BOEING", "MCDONNELL DOUGLAS", "BOMBARDIER INC", "NA"), manufacturer, "OTHERS")) %>% #aggregates all the smaller manufacturers into "OTHERS"  
 group\_by(manufacturer, month) %>% # groups the data by manufacturers  
 summarise(flightsnumber = n()) %>% # counts how many flights were made with planes manufactured by each manufacturer  
 ggplot(aes(x = month, y = flightsnumber, fill = manufacturer)) + # creates a plot showing the number of flights per month per manufacturer  
 geom\_bar(stat = "identity", position = "fill") + # sets the plot type as a bar plot that sums up to 100%  
 labs(y = "Percentage of Flights") # renames the y-axis to show the proper measure

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



#the final graphic shows the percentage of flights per manufacturer each month, and one can observe that the number of "OTHERS" decreased over time, with a slight increase in both "BOMBARDIER INC" and "MCDONNELL DOUGLAS". The other 3 players oscilated from one month to another, but no significant trend over time can be observed

## 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?

#| label: problem-6  
  
# What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
  
flightplanes %>%  
 distinct(tailnum, .keep\_all = TRUE) %>% # keeps only one observation per tailnum  
 select(tailnum, year.y, type, manufacturer, model, engines, seats, speed, engine) %>% # drops all features that are not related to the plane  
 arrange(year.y) %>% # sort all planes by ascending year of manufature  
 slice(1) # get the oldest plane that flew from New York City airports in 2013

# A tibble: 1 × 9  
 tailnum year.y type manufacturer model engines seats speed engine  
 <chr> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
1 N381AA 1956 Fixed wing multi… DOUGLAS DC-7… 4 102 232 Recip…

# How many airplanes that flew from New York City are included in the planes table?  
  
flightplanes %>%  
 semi\_join(planes, by = "tailnum") %>% # filters only travels made with with planes that has information in the planes table  
 summarise(number\_of\_planes = n\_distinct(tailnum)) # counts how many distinct airplanes flew from New York City that are included in the planes table

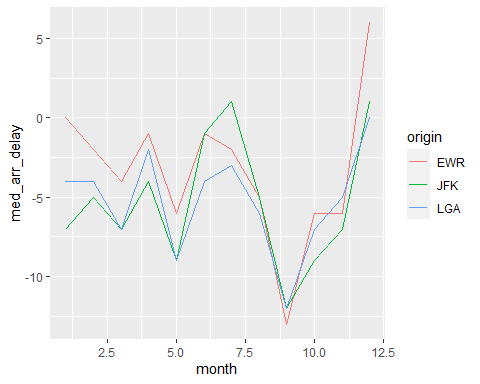
# A tibble: 1 × 1  
 number\_of\_planes  
 <int>  
1 3322

## 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.

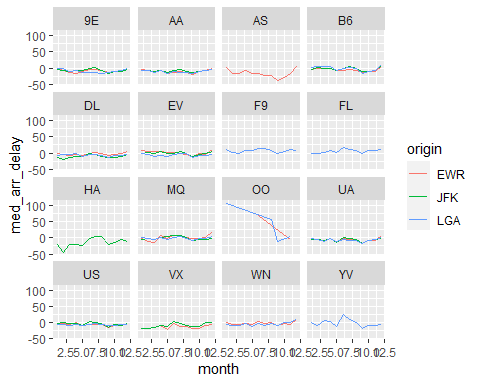
#| label: problem-7  
  
# What is the median arrival delay on a month-by-month basis in each airport?  
  
flights %>%  
 filter(!is.na(arr\_delay)) %>% # get rid of flights that have no information on the arrival delay  
 group\_by(month, origin) %>% # groups the data by month and origin to get the associated values after  
 summarise(med\_arr\_delay=median(arr\_delay)) %>% #gets the median arrival delay for each airport each month  
 ggplot(aes(x=month, y=med\_arr\_delay, color=origin)) + # plot the results  
 geom\_line() # as a line plot

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



# For each airline, plot the median arrival delay for each month and origin airport  
  
flights %>%  
 filter(!is.na(arr\_delay)) %>% # get rid of flights that have no information on the arrival delay  
 group\_by(month, origin, carrier) %>% # groups the data by month, origin and carrier to get the associated values after  
 summarise(med\_arr\_delay=median(arr\_delay)) %>% #gets the median arrival delay for each carrier in each airport each month  
 ggplot(aes(x=month, y=med\_arr\_delay, color=origin)) + # plot the results  
 geom\_line() + # as a line plot  
 facet\_wrap(~ carrier) # with a separete plot for each carrier

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

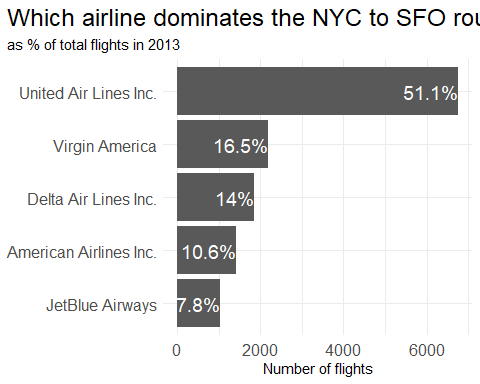


## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United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

#| label: problem-8  
  
# Which airlines flew the most to SFO?  
  
flightsairlines <- left\_join(flights, airlines, by = "carrier") # creates a new dataframe by joining the flights and airlines tables  
  
fly\_into\_sfo <- flightsairlines %>% # creates the fly\_into\_sfo dataframe that will contain the information about which carriers flew to SFO, how many times and whaat % does it represent  
 filter(!is.na(dep\_time) & dest == "SFO") %>% # filters the original data to keep only flights to SFO that were not cancelled  
 group\_by(name) %>% # groups the data by the name of the airline  
 summarise(number\_of\_flights = n()) %>% # counts how many times each airline flew to SFO  
 mutate(percentage\_of\_flights = number\_of\_flights / sum(number\_of\_flights)) # calculates what is their market share on flights from New York City to 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, number\_of\_flights)) %>%   
   
 ggplot() +  
   
 aes(x = number\_of\_flights,   
 y = name) +  
   
 # a simple bar/column plot  
 geom\_col() +  
   
 # add labels, so each bar shows the % of total flights   
 geom\_text(aes(label = paste0(round(percentage\_of\_flights\*100, 1), "%")),  
 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.

{r}  
  
#| label: problem-9  
  
# To reproduce the image above, I would go through the following steps:  
  
#i) Use a left join to merge the cancellations dataframe with the airlines dataframe to get the airlines names used as a tag in the horizontal strip  
  
#ii) Group the data by month, name and origin  
  
#iii) Summarize the data to get a count of how many observations there is for each combination of month, name and origin  
  
#iv) Call ggplot to create a plot, setting x=month and y=count as the variables in the aesthetic (aes)  
  
#v) Define the plot time as a bar plot using geom\_bar()  
  
#vi) Define a grid of plots using facet\_grid(name ~ origin)  
  
#vii) Use the labs function to add title and tags to the axes  
  
#viii) Use the geom\_text function to add the label to each bar  
  
#ix) Use the theme function to adjust the axis and strip texts and the strip background  
  
  
############################################################################  
  
cancellations %>%  
 left\_join(airlines, by="carrier") %>%  
 group\_by(month, name, origin) %>%  
 summarise(count = n())%>%  
 ggplot(aes(x = month, y = count)) +  
 geom\_bar(stat = "identity") +  
 facet\_grid(name ~ origin) +  
 labs(title = "Cancellations of flights to SFO by month carrier and airport origin",  
 x = "Month",  
 y = "Count of Cancelled Flights") +  
 geom\_text(aes(label = count), vjust = 1, color = "white", size = 1) +  
 theme\_minimal() +  
 theme(axis.text = element\_text(size = 6),  
 strip.text = element\_text(size = 4),  
 strip.background = element\_rect(fill = "lightgray", color = "gray", size = 1))

## 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!

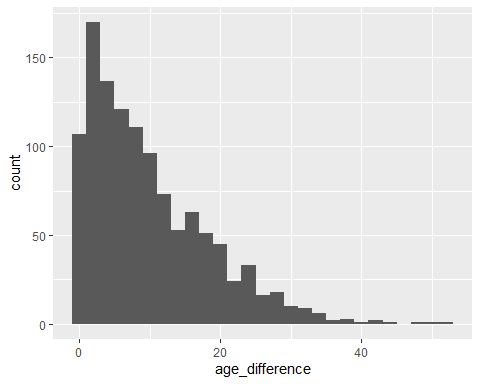
The following is a data dictionary of the variables used

| 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 |

#| label: problem-10  
  
library(DescTools)  
  
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.

# How is age\_difference distributed? What's the 'typical' age\_difference in movies?  
  
age\_gaps %>%  
 ggplot(aes(x=age\_difference)) +  
 geom\_histogram(binwidth = 2) # Through an histogram of age\_difference, one can investigate its distribution and observe that it's concentrated in the region between 0 and 10, with a long tail to higher values. The typical age\_difference, as per the highest bar in the histogram, is around 2 or 3 years.



age\_gaps %>%  
 summarise(upto5years = mean(age\_difference <= 5)) # with this summarise call we can check that 35.8% of the age difference observations are equal to or less than 5 years

# A tibble: 1 × 1  
 upto5years  
 <dbl>  
1 0.358

age\_gaps %>%  
 summarise(upto10years = mean(age\_difference <= 10)) # with this summarise call we can check that 59.8% of the age difference observations are equal to or less than 10 years

# A tibble: 1 × 1  
 upto10years  
 <dbl>  
1 0.598

age\_gaps %>%  
 summarise(upto20years = mean(age\_difference <= 20)) # with this summarise call we can check that 86.8% of the age difference observations are equal to or less than 20 years

# A tibble: 1 × 1  
 upto20years  
 <dbl>  
1 0.868

# How frequently does the "half plus seven" rule apply in this dataset?  
  
age\_gaps %>%  
 mutate(half\_plus\_seven = if\_else (actor\_1\_age/2 + 7 < actor\_2\_age,1,0)) %>% # uses the mutate function to create a dummy variable that takes the value of 1 if the half plus seven rule applies and 0 otherwise  
 summarise(half\_plus\_seven\_frequency = mean(half\_plus\_seven)) # takes the mean of the created dummy variable to get the % of observations to which the half plus seven rule applies - 68.8%

# A tibble: 1 × 1  
 half\_plus\_seven\_frequency  
 <dbl>  
1 0.688

# Which movie has the greatest number of love interests?  
  
age\_gaps %>%  
 group\_by (movie\_name) %>% # groups the data by the name of the movie  
 summarise(love\_interests = n()) %>% # counts how many love interests each movie has  
 arrange(desc(love\_interests)) %>% # sorts the data on descending order of number of love interests  
 slice(1) # selects the movie with the most number of love interests, which is Love Actually

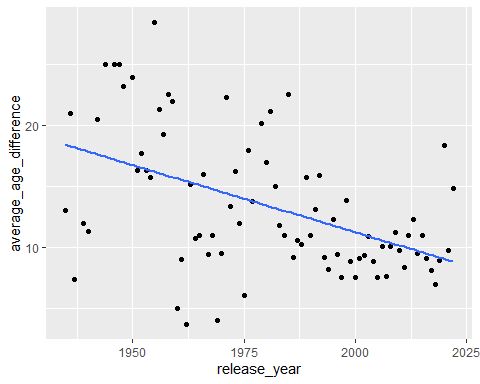
# A tibble: 1 × 2  
 movie\_name love\_interests  
 <chr> <int>  
1 Love Actually 7

# Which actors/ actresses have the greatest number of love interests in this dataset?  
  
love\_interests\_actor\_1 <- age\_gaps %>% # creates a new dataframe for the number of love interests for the actors/actrices in slot 1  
 group\_by (actor\_1\_name) %>% # groups the data by actor 1's name  
 summarise (love\_interests\_1 = n()) # counts how many love interests each of those actors/actrices has  
  
love\_interests\_actor\_2 <- age\_gaps %>% # creates a new dataframe for the number of love interests for the actors/actrices in slot 2  
 group\_by (actor\_2\_name) %>% # groups the data by actor 2's name  
 summarise (love\_interests\_2 = n()) # counts how many love interests each of those actors/actrices has  
  
full\_join(love\_interests\_actor\_1, love\_interests\_actor\_2, by = c("actor\_1\_name" = "actor\_2\_name")) %>% # joins together the data from the two individual actors/actrices dataframes  
 mutate\_at(vars(love\_interests\_1, love\_interests\_2), ~replace(., is.na(.), 0)) %>% # replace any NA value (because the actor/actrice was absent in one of the previous dataframe) with 0  
 mutate(total\_love\_interests = love\_interests\_1 + love\_interests\_2) %>% # calculates the total number of love interests, considering apparitions in both positions  
 arrange(desc(total\_love\_interests)) %>% # sorts the data on descending order of number of total love interests  
 slice(1:5) # selects the top 5 actor/actrice with the most number of love interests. Keanu Reeves, with 27 total love interests, leads the ranking, followed by Adam Sandler (20), Leonardo DiCaprio, Roger Moore and Sean Connery (17)

# A tibble: 5 × 4  
 actor\_1\_name love\_interests\_1 love\_interests\_2 total\_love\_interests  
 <chr> <dbl> <dbl> <dbl>  
1 Keanu Reeves 24 3 27  
2 Adam Sandler 18 2 20  
3 Leonardo DiCaprio 11 6 17  
4 Roger Moore 17 0 17  
5 Sean Connery 15 2 17

# Is the mean/median age difference staying constant over the years (1935 - 2022)?  
  
age\_gaps %>%  
 group\_by(release\_year) %>% # groups the data by release year  
 summarise(average\_age\_difference=mean(age\_difference)) %>% # computes the average age difference for each year  
 ggplot(aes(x=release\_year,y=average\_age\_difference)) + # creates a plot of the average age difference x release year of the movie  
 geom\_point() + # defines the plot as a scatter plot  
 geom\_smooth(method = "lm" ,se = FALSE) # adds a linear trend line to the plot, from which we can see there is a downward slope, meaning that the average age difference went down throught years (got reduced by about 10 years over the 87-year range)

`geom\_smooth()` using formula = 'y ~ x'



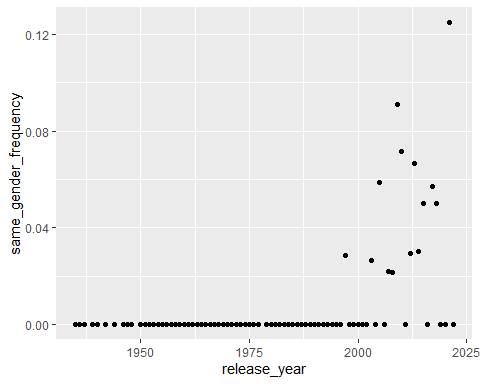
age\_gaps %>%  
 group\_by(release\_year) %>% # groups the data by release year  
 summarise(average\_age\_difference=mean(age\_difference)) %>% # computes the average age difference for each year  
 summarise(correlation = cor(average\_age\_difference, release\_year)) # double checks the existence of a negative correlation betwen release year and average age difference (-0.486, equivalent to an R2 of 0.236 in the trend line observed in the previous plot)

# A tibble: 1 × 1  
 correlation  
 <dbl>  
1 -0.486

# How frequently does Hollywood depict same-gender love interests?  
  
age\_gaps %>%  
 mutate(same\_gender=if\_else(character\_1\_gender==character\_2\_gender,1,0)) %>% # creates a dummy variable that takes the value of 1 if the love interest was between characters of the same gender and 0 otherwise  
 summarise(same\_gender\_frequency=mean(same\_gender)) # gets the frequency of same gender love interest among the dataset by computing the mean of the previously defined dummy variable. We can conclude that same gender love interests represents less than 2% of the observations

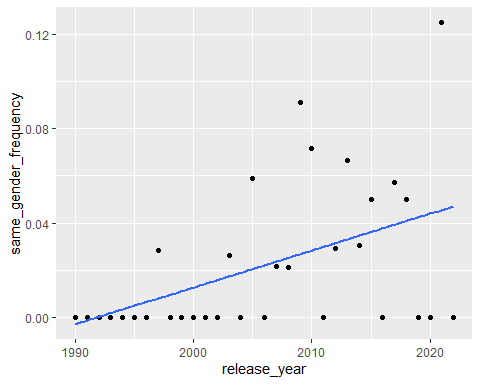
# A tibble: 1 × 1  
 same\_gender\_frequency  
 <dbl>  
1 0.0199

age\_gaps %>%  
 mutate(same\_gender=if\_else(character\_1\_gender==character\_2\_gender,1,0)) %>% # creates a dummy variable that takes the value of 1 if the love interest was between characters of the same gender and 0 otherwise  
 group\_by(release\_year) %>% # groups the data by release year  
 summarise(same\_gender\_frequency=mean(same\_gender)) %>% # gets the frequency of same gender love interest each year  
 ggplot(aes(x=release\_year,y=same\_gender\_frequency)) + # creates a plot of the same gender love interest frequency x release year of the movie  
 geom\_point() # defines the plot as a scatter plot. From this plot, we can observe that same gender love interest appeared for the first time only in 1997, 62 years after the oldest record in the dataset



age\_gaps %>%  
 mutate(same\_gender=if\_else(character\_1\_gender==character\_2\_gender,1,0)) %>% # creates a dummy variable that takes the value of 1 if the love interest was between characters of the same gender and 0 otherwise  
 group\_by(release\_year) %>% # groups the data by release year  
 summarise(same\_gender\_frequency=mean(same\_gender)) %>% # gets the frequency of same gender love interest each year  
 slice(50:82) %>% # slices the dataset so it starts in 1990  
 ggplot(aes(x=release\_year,y=same\_gender\_frequency)) + # creates a plot of the same gender love interest frequency x release year of the movie  
 geom\_point() + # defines the plot as a scatter plot.  
 geom\_smooth(method = "lm" ,se = FALSE) # adds a linear trend line to the plot, from which we can see there is a upward slope, meaning that the same gender love interest has become for frequent since 1990

`geom\_smooth()` using formula = 'y ~ x'



age\_gaps %>%  
 mutate(same\_gender=if\_else(character\_1\_gender==character\_2\_gender,1,0)) %>% # creates a dummy variable that takes the value of 1 if the love interest was between characters of the same gender and 0 otherwise  
 group\_by(character\_1\_gender) %>% # groups the data by character 1 gender  
 summarise(same\_gender\_observations=sum(same\_gender)) # gets the number of observations of same gender love interest per gender. From this table, we can observe that there is no significant difference between man-man and woman-woman love interest observations

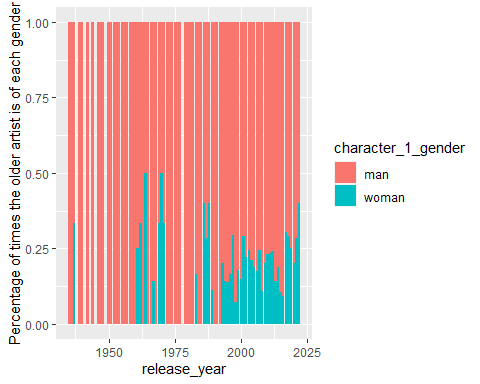
# A tibble: 2 × 2  
 character\_1\_gender same\_gender\_observations  
 <chr> <dbl>  
1 man 12  
2 woman 11

# Extra: which gender is played by an older actor/actress more often?  
  
age\_gaps %>%  
 mutate(same\_gender=if\_else(character\_1\_gender==character\_2\_gender,1,0)) %>% # creates a dummy variable that takes the value of 1 if the love interest was between characters of the same gender and 0 otherwise  
 filter(same\_gender==0) %>% # filters the data to keep only observations of different gender love interests  
 group\_by(character\_1\_gender) %>% # groups the data by character 1 gender  
 summarise(older\_partner\_observations=n()) # creates a table showing how many times each gender played the oldest partner in the love interest. From it, we can observe that man are more than 4.5 times more likely to be played by an older actor

# A tibble: 2 × 2  
 character\_1\_gender older\_partner\_observations  
 <chr> <int>  
1 man 929  
2 woman 203

age\_gaps %>%  
 mutate(same\_gender=if\_else(character\_1\_gender==character\_2\_gender,1,0)) %>% # creates a dummy variable that takes the value of 1 if the love interest was between characters of the same gender and 0 otherwise  
 filter(same\_gender==0) %>% # filters the data to keep only observations of different gender love interests  
 group\_by(character\_1\_gender, release\_year) %>% # groups the data by character 1 gender and year  
 summarise(older\_partner\_observations=n()) %>% # creates a table showing how many times each gender played the oldest partner in the love interest.  
 ggplot(aes(x = release\_year, y = older\_partner\_observations, fill = character\_1\_gender)) + # creates a plot of how often each gender is played by an older character over the years  
 geom\_bar(stat = "identity", position = "fill") + # sets the plot type as a bar plot that sums up to 100%  
 labs(y = "Percentage of times the older artist is of each gender") # once again, it appear that in the 1990's it started to become more often that female characters are played by artists that are older than their male character love interest pair

`summarise()` has grouped output by 'character\_1\_gender'. You can override  
using the `.groups` argument.



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: -
* Approximately how much time did you spend on this problem set: 6 hours
* What, if anything, gave you the most trouble: getting right the graphic on Question 5 (the percentage bar plots)

**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.