Homerwork 1

PAULA LAGOS

5/14/23

library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.2 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.2 ✔ tibble 3.2.1  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.1   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(nycflights13)  
library(skimr)  
data(flights)

# 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

#looking at the data with VIEW, and we identify the important variables.   
view(flights)  
# Had an arrival delay of two or more hours (> 120 minutes)  
  
#I create Problem 1. Looking for lines with the variable ARR\_DELAY that show more than 120 minutes.Sort by that variable.   
flights %>%   
 filter(arr\_delay>120) %>%   
 arrange(arr\_delay)

# A tibble: 10,034 × 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 6 2051 1820 151 2206 2005  
 2 2013 1 17 801 600 121 1002 801  
 3 2013 1 17 2004 1828 96 2227 2026  
 4 2013 1 20 2259 2102 117 118 2317  
 5 2013 1 22 1206 1006 120 1322 1121  
 6 2013 1 24 1848 1703 105 2011 1810  
 7 2013 1 26 1342 1129 133 1639 1438  
 8 2013 1 26 1513 1315 118 1824 1623  
 9 2013 10 1 1338 1105 153 1446 1245  
10 2013 10 3 1034 829 125 1323 1122  
# ℹ 10,024 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) - We are looking at the arrival airport (DEST) to be IAH or HOU  
Question1.2<-flights %>%   
 filter(dest=="IAH"|dest=="HOU")  
  
  
# Were operated by United (`UA`), American (`AA`), or Delta (`DL`)- This means the carrier (carrier) is one of those  
Question1.3<-flights %>%   
 filter(carrier=="UA"|carrier=="AA"|carrier=="DL")  
  
  
# Departed in summer (July, August, and September). In here we look for the variable month to be 7, 8 or 9  
Question1.4<-flights %>%   
 filter(month==7|month==8|month==9)  
   
   
# Arrived more than two hours late, but didn't leave late. Has to have a arr\_delay of 120 or more, and a dep\_delay of 0 or less  
Question1.5<-flights %>%   
 filter(arr\_delay>120) %>%  
 filter(dep\_delay<=0)  
   
  
  
# Were delayed by at least an hour, but made up over 30 minutes in flight. This means dep\_delay is bigger than arr\_delay by more than 30  
Question1.6<-flights %>%  
 filter(dep\_delay>arr\_delay+30)

## 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?  
#If cancelled flights the ones with no information on dep\_time, I need to count by month those with na, and then, calculate the division of said number over total flights for that month  
  
  
# Calculating the number of cancelled flights per month on cancelled\_flights column  
  
Question2.1cancelled<- flights %>%  
 group\_by(month) %>%  
 summarise(cancelled\_flights = sum(is.na(dep\_delay)))  
   
  
# Calculate the total number of flights by month con total\_flights column  
Q2.1flights\_per\_month <- flights %>%   
 group\_by(month) %>%   
 summarise(total\_flights = n())  
  
# Join the two data frames (left by month) and calculate the percentage of cancelled flights on a new table perc\_cancelled\_per\_month  
perc\_cancelled\_per\_month <- left\_join(Question2.1cancelled, Q2.1flights\_per\_month, by = "month") %>%   
 mutate(cancelled\_flights\_percentage= cancelled\_flights/ total\_flights) %>%   
 arrange(desc(cancelled\_flights\_percentage))  
#by looking at the table, the months 8 to 11 seem to have the lower percentage of cancelled flights, but the ones on the higher end don't seem to follow a pattern  
  
  
# Find the month with the highest percentage of cancelled flights  
max\_cancelled <- perc\_cancelled\_per\_month %>%   
 filter(cancelled\_flights\_percentage == max(cancelled\_flights\_percentage))  
  
# Find the month with the lowest percentage of cancelled flights  
min\_cancelled<- perc\_cancelled\_per\_month %>%   
 filter(cancelled\_flights\_percentage == min(cancelled\_flights\_percentage))  
  
# Print the results  
cat("Month with the highest percentage of cancelled flights:", max\_cancelled$month, "\n")

Month with the highest percentage of cancelled flights: 2

cat("Month with the lowest percentage of cancelled flights:", min\_cancelled$month, "\n")

Month with the lowest percentage of cancelled flights: 10

## 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.

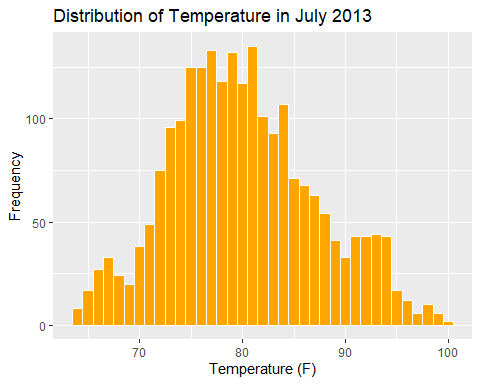
#Identify NYC airport codes (EWR, JFK, LGA) and count the lines any time the origin is one of those, groupy by tail number  
  
flights\_from\_nyc<-flights %>%   
 filter(year==2013,origin %in% c("EWR","JFK","LGA")) %>%   
 group\_by(tailnum) %>%   
 summarise(Flights\_from\_NYC=n()) %>%   
 arrange(desc(Flights\_from\_NYC))  
  
#Filter only for the highest at this variable  
max\_flightsNYC <- flights\_from\_nyc%>%  
 slice(2)#WE SEE THAT THE ERROR LINE WITH NA IS THE ONE WITH MORE COINCIDENCES, WE WANT TO FILTER ONLY FOR VALUES DIFFERENT FROM NA.  
max\_flightsNYC %>%   
#LEftjoin with planes table to show information on that plane  
 left\_join(planes,by="tailnum")

# A tibble: 1 × 10  
 tailnum Flights\_from\_NYC year type manufacturer model engines seats speed  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int>  
1 N725MQ 575 NA <NA> <NA> <NA> NA NA NA  
# ℹ 1 more variable: engine <chr>

## 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`?

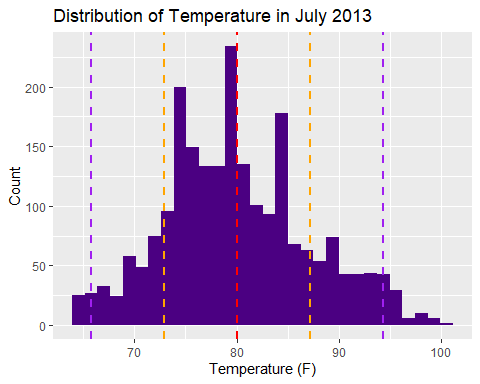
#distribution of temperature in July 2013: identify if they   
#Identify outliers of wind\_speed  
#relationship betweer dewp and humid  
#relationship betweer precip and visib  
#From ChatGPT:  
  
  
# Filter for weather data in July 2013  
weather\_jul13 <- weather %>%  
 filter(month == 7, year == 2013)  
  
# Plot a histogram of temperature  
ggplot(weather\_jul13, aes(x = temp)) +   
 geom\_histogram(binwidth = 1, color = "white", fill = "orange") +  
 labs(title = "Distribution of Temperature in July 2013",  
 x = "Temperature (F)",  
 y = "Frequency")



#[\*\*\*\*]Graphing to identify outliers depending on standard deviations  
weather %>%  
 filter(month == 7) %>%  
 ggplot(aes(x = temp)) +  
 geom\_histogram(fill = "#4B0082") +  
 geom\_vline(aes(xintercept = mean(temp)), color = "red", linetype = "dashed", size = 1) +  
 geom\_vline(aes(xintercept = mean(temp) + sd(temp)), color = "orange", linetype = "dashed", size = 1) +  
 geom\_vline(aes(xintercept = mean(temp) - sd(temp)), color = "orange", linetype = "dashed", size = 1) +  
 geom\_vline(aes(xintercept = mean(temp) + 2\*sd(temp)), color = "purple", linetype = "dashed", size = 1) +  
 geom\_vline(aes(xintercept = mean(temp) - 2\*sd(temp)), color = "purple", linetype = "dashed", size = 1) +  
 labs(title = "Distribution of Temperature in July 2013",  
 x = "Temperature (F)",  
 y = "Count")

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

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

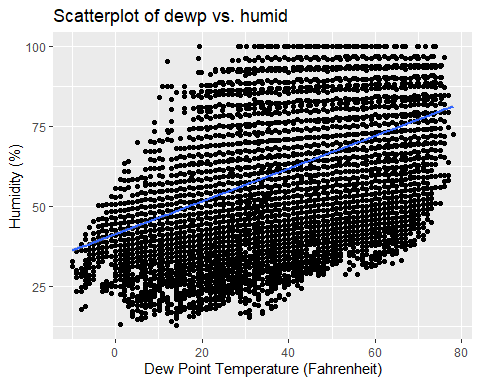


#WE SEE THAT THE THERE ARE OUTLIERS with more than 2 st dev away from mean  
  
#To evaluate a relationship betweer dewp and humid, will build a scatter plot to look for correlation. Another option could be to calculate correlation betweeen variables.   
  
  
# Create a scatterplot of dewp vs. humid with trend line  
ggplot(weather, aes(x = dewp, y = humid)) +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 labs(title = "Scatterplot of dewp vs. humid",  
 x = "Dew Point Temperature (Fahrenheit)",  
 y = "Humidity (%)")

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

Warning: Removed 1 rows containing non-finite values (`stat\_smooth()`).

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



#We see a positive correlation, now I will Evaluate strength of correlation  
  
# Fit a linear regression model for dewp vs. humid  
model <- lm(humid ~ dewp, data = weather)  
  
# Print the summary of the model  
summary(model)

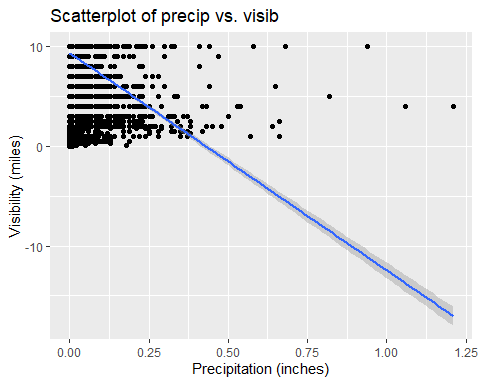
Call:  
lm(formula = humid ~ dewp, data = weather)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-42.097 -11.839 -0.215 11.626 48.764   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 41.294097 0.243286 169.74 <2e-16 \*\*\*  
dewp 0.512451 0.005318 96.37 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 16.66 on 26112 degrees of freedom  
 (1 observation deleted due to missingness)  
Multiple R-squared: 0.2623, Adjusted R-squared: 0.2623   
F-statistic: 9287 on 1 and 26112 DF, p-value: < 2.2e-16

# Extract the R-squared value from the summary of the model  
r\_squared <- summary(model)$r.squared  
cat("R-squared value:", r\_squared)

R-squared value: 0.2623439

#With an R-squared of 0.2623 we can see that the fit is not very strong.   
  
# Load required libraries  
  
  
# Create a scatterplot of precip vs. visib with trend line  
ggplot(weather, aes(x = precip, y = visib)) +  
 geom\_point() +  
 geom\_smooth(method = "lm") +  
 labs(title = "Scatterplot of precip vs. visib",  
 x = "Precipitation (inches)",  
 y = "Visibility (miles)")

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



# Calculate the Pearson correlation coefficient between precip and visib  
correlation <- cor(weather$precip, weather$visib)  
  
cat("Pearson correlation coefficient:", correlation)

Pearson correlation coefficient: -0.3199118

#with a correlacion coefficient of -0.3199 there is not much evidence of a strong correlation, but there is a negative relationship.

## 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.)

# Count the number of planes with missing date of manufacture  
missing\_manufacture\_count <- planes %>%  
 filter(is.na(year))%>%  
 nrow()  
  
missing\_manufacture\_count

[1] 70

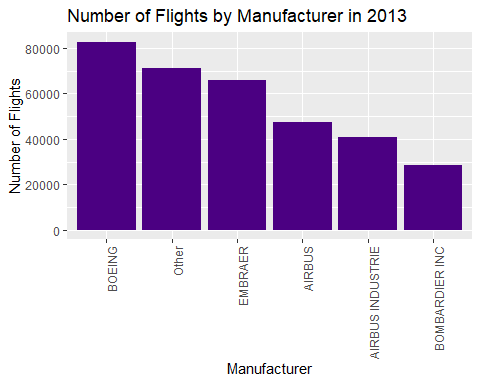
#We see that there are 70 planes without information on the year of manufacturing  
  
  
# Count the number of planes for each manufacturer  
manufacturer\_counts <- planes %>%   
 count(manufacturer, sort = TRUE)  
  
# Get the top 5 manufacturers  
top\_manufacturers <- head(manufacturer\_counts$manufacturer, 5)  
  
top\_manufacturers

[1] "BOEING" "AIRBUS INDUSTRIE" "BOMBARDIER INC" "AIRBUS"   
[5] "EMBRAER"

# Filter flights for the year 2013  
flights\_2013 <- flights %>%  
 filter(year == 2013)  
  
# Join flights with planes using tailnum  
flights\_planes\_2013 <- flights\_2013 %>%  
 left\_join(planes, by = "tailnum")  
  
# Recode rare vendors into "Other" category using case\_when()  
flights\_planes\_2013 <- flights\_planes\_2013 %>%  
 mutate(manufacturer = case\_when(  
 manufacturer %in% top\_manufacturers ~ manufacturer,  
 TRUE ~ "Other"  
 ))  
  
# Count the number of flights for each manufacturer  
manufacturer\_counts\_2013 <- flights\_planes\_2013 %>%  
 count(manufacturer, sort = TRUE)  
  
manufacturer\_counts\_2013

# A tibble: 6 × 2  
 manufacturer n  
 <chr> <int>  
1 BOEING 82912  
2 Other 71331  
3 EMBRAER 66068  
4 AIRBUS 47302  
5 AIRBUS INDUSTRIE 40891  
6 BOMBARDIER INC 28272

#ggplot  
ggplot(manufacturer\_counts\_2013, aes(x = reorder(manufacturer, -n), y = n)) +  
 geom\_bar(stat = "identity", fill = "#4B0082") +  
 labs(title = "Number of Flights by Manufacturer in 2013",  
 x = "Manufacturer",  
 y = "Number of Flights") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1))



## 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?

#What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
   
# Filter flights for the year 2013  
flights\_2013 <- flights %>%  
 filter(year == 2013)  
  
# Join flights with planes using tailnum  
flights\_planes\_2013 <- flights\_2013 %>%  
 left\_join(planes, by = "tailnum")  
#fIlter flights departing from NYC  
flights\_nyc\_planes13<-flights\_planes\_2013 %>%   
 filter(origin %in% c("EWR","JFK","LGA"))  
  
# Find the oldest plane based on the year of manufacture  
oldest\_plane <- flights\_nyc\_planes13 %>%  
 filter(!is.na(year.y)) %>%  
 arrange(year.y) %>%  
 slice(1)  
  
oldest\_plane$tailnum

[1] "N381AA"

#How many airplanes that flew from New York City are included in the planes table?  
# Count the number of unique airplanes in the planes table  
unique\_airplanes <- flights\_nyc\_planes13 %>%  
 distinct(tailnum) %>%  
 nrow()  
  
unique\_airplanes

[1] 4044

## 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.

# Calculate median arrival delay on a month-by-month basis in each airport  
median\_arrival\_delay <- flights %>%  
 group\_by(month, dest) %>%  
 summarize(median\_arr\_delay = median(arr\_delay, na.rm = TRUE))

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

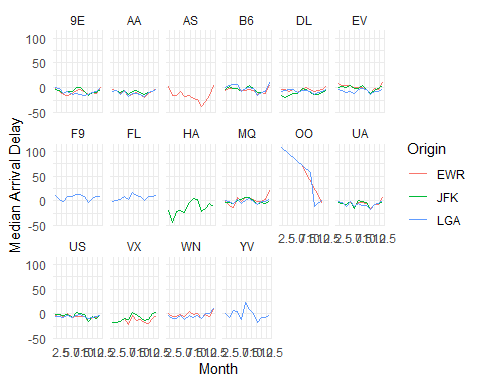
median\_arrival\_delay

# A tibble: 1,113 × 3  
# Groups: month [12]  
 month dest median\_arr\_delay  
 <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

#For each airline, plot the median arrival delay for each month and origin airport.  
library(ggplot2)  
  
# Calculate median arrival delay for each airline, month, and origin airport  
median\_arrival\_delay\_airline <- flights %>%  
 group\_by(month, carrier, origin) %>%  
 summarize(median\_arr\_delay = median(arr\_delay, na.rm = TRUE))

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

# Plot the median arrival delay for each airline  
ggplot(median\_arrival\_delay\_airline, aes(x = month, y = median\_arr\_delay, color = origin)) +  
 geom\_line() +  
 facet\_wrap(~ carrier, nrow = 3) +  
 labs(x = "Month", y = "Median Arrival Delay", color = "Origin") +  
 theme\_minimal()



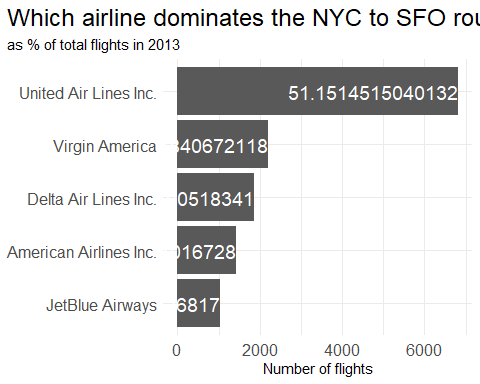
## Problem 8: route to (SFO).three variables: the name, count of times it flew to SFO, and the percent of the trips that that particular airline flew to SFO.

#With the help of Microsoft's Bing Chat answer, after iterating +5 times with chatGPT:  
  
fly\_into\_sfo <- flights %>%  
 #filtering landing at SFO  
 filter(dest == "SFO") %>%   
 #showwing info by carrier  
 group\_by(carrier) %>%  
 #calculating total flights per carrier  
 summarise(count = n()) %>%  
 #creating new column of percentage of that number over total  
 mutate(percent = count / sum(count) \* 100) %>%  
 #Adding information of names by the key carrier  
 left\_join(airlines, by = "carrier") %>%  
 #selecting columns to show, no code, only name and calculations  
 select(name, count, percent) %>%   
 #arranging by count  
 arrange(percent)  
  
  
fly\_into\_sfo

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

And here is some bonus ggplot code to plot your dataframe

library(forcats)  
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

To organise data:

* filter for desitnation airport SFO
* filter not cancelled by filtering out those with no informtion on depart time
* we calculate cancellations per airline name and origin airport and month(summarising for those variables)
* we arrange per month

To plot:

* First ask ggplot and select data
* define X axis as months, and Y axis as count, or number of cancelled flights
* Then define the type of graph as bars
* show the information for every airline name and takeoff airport pair with wrap
* eliminate labels on axis

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

Answer:

* HOW AGE DIFFERENCE IS DISTRIBUTED Mean, median, desv est., modew
* HALF PLUS SEVEN;how many how many couples match the age/2+7 rule, by calculating a new column with the calculation of the older actor, and checking if that calculation is lower or equal to the age of the youngest
* Which movie has the greatest number of love interests?
* MOVIES WITH GREATEST LOVE INTERESTS: count love interest and summarise by movie
* ACTORWITH MOST LOVE INTERESTS count love interest, summarise by actor, arrange descending, pick first
* MEAN OVER THE YEARS: Calculate the mean of the age difference,s ummarise by year, arrange ascending, plot line (smooth)
* SAME GENDER: create column of type of love interest: different or same, and add to every line accordign to the genders of each. then count different and same summarising by type of love interest

# 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: Just me, with the help of Chat GPT and Microsoft Bing Chat
* Approximately how much time did you spend on this problem set: ANSWER HERE
* What, if anything, gave you the most trouble: multiple lines to answer one question, such as calculations, filter, etc. Remembering to checck the data for NA, or errors, validating my answers

**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.