Homerwork 1

IGNACIO GAING

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 >= 2)

# A tibble: 127,929 × 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 1 554 558 -4 740 728  
 5 2013 1 1 555 600 -5 913 854  
 6 2013 1 1 558 600 -2 753 745  
 7 2013 1 1 558 600 -2 924 917  
 8 2013 1 1 559 600 -1 941 910  
 9 2013 1 1 600 600 0 837 825  
10 2013 1 1 602 605 -3 821 805  
# ℹ 127,919 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

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

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

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

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

# A tibble: 86,326 × 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(dep\_delay <= 0,arr\_delay >120)

# 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,arr\_delay <30)

# 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))

siome text

#library(dplyr)  
  
# What months had the highest and lowest % of cancelled flights?  
  
# Calculate the percentage of cancelled flights per month  
cancelled\_flights\_per\_month <- flights %>%   
 group\_by(month) %>%  
 summarise(total\_flights = n(),   
 cancelled\_flights = sum (is.na(dep\_time))) %>%   
 mutate(percentage\_cancelled = round(cancelled\_flights / total\_flights \* 100,2))  
  
# Display the results  
cancelled\_flights\_per\_month

# A tibble: 12 × 4  
 month total\_flights cancelled\_flights percentage\_cancelled  
 <int> <int> <int> <dbl>  
 1 1 27004 521 1.93  
 2 2 24951 1261 5.05  
 3 3 28834 861 2.99  
 4 4 28330 668 2.36  
 5 5 28796 563 1.96  
 6 6 28243 1009 3.57  
 7 7 29425 940 3.19  
 8 8 29327 486 1.66  
 9 9 27574 452 1.64  
10 10 28889 236 0.82  
11 11 27268 233 0.85  
12 12 28135 1025 3.64

# Find the month with the maximum and minimum percentage of cancelled flights  
month\_with\_max\_cancelled <- which.max(cancelled\_flights\_per\_month$percentage\_cancelled)  
month\_with\_min\_cancelled <- which.min(cancelled\_flights\_per\_month$percentage\_cancelled)  
  
# Display the month with the maximum and minimum percentage of cancelled flights  
cat("The month with the maximum percentage of cancelled flights was:", month.name[month\_with\_max\_cancelled], "\n")

The month with the maximum percentage of cancelled flights was: February

cat("The month with the minimum percentage of cancelled flights was:", month.name[month\_with\_min\_cancelled], "\n")

The month with the minimum percentage of cancelled flights was: October

## 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.

# Load necessary packages  
library(dplyr)  
library(nycflights13)  
  
# Filter flights for New York City airports and the year 2013 and that   
filtered\_flights <- flights %>%  
 filter(substr(year, 1, 4) == "2013" & !is.na(tailnum))  
  
# Count the number of occurrences of each tail number  
tailnum\_counts <- filtered\_flights %>%  
 count(tailnum)  
  
# Sort the counts in descending order  
sorted\_tailnum\_counts <- tailnum\_counts %>%  
 arrange(desc(n))  
  
#Join with planes to get planes >50   
  
sorted\_tailnum\_50\_counts <- sorted\_tailnum\_counts %>%  
 left\_join(planes, by = "tailnum", na\_matches = "never") %>%  
 filter(seats > 50)  
  
  
# Retrieve the tail number with the highest count  
most\_frequent\_plane <- sorted\_tailnum\_50\_counts$tailnum[1]  
  
# Display the results  
most\_frequent\_plane

[1] "N328AA"

# Get the plane with the greatest number of flights and more than 50 seats  
selected\_plane <- planes %>%  
 filter(seats > 50 & tailnum == most\_frequent\_plane)  
  
# Filter flights for the selected plane in 2013  
selected\_flights <- filtered\_flights %>%  
 filter(tailnum == most\_frequent\_plane)  
  
# Display the results  
selected\_flights

# A tibble: 393 × 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 1026 1030 -4 1351 1340  
 2 2013 1 2 1038 1030 8 1347 1340  
 3 2013 1 3 1152 1200 -8 1446 1510  
 4 2013 1 4 858 900 -2 1210 1220  
 5 2013 1 5 851 900 -9 1206 1220  
 6 2013 1 6 1027 1030 -3 1335 1340  
 7 2013 1 7 724 730 -6 1008 1100  
 8 2013 1 7 2134 2135 -1 19 50  
 9 2013 1 8 2130 2135 -5 114 50  
10 2013 1 9 1701 1645 16 1958 2005  
# ℹ 383 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>

# Count flights by destination  
flight\_counts\_by\_dest <- selected\_flights %>%  
 count(dest) %>%  
 arrange(desc(n))  
  
# Output the result  
flight\_counts\_by\_dest

# A tibble: 6 × 2  
 dest n  
 <chr> <int>  
1 LAX 313  
2 SFO 52  
3 MIA 25  
4 BOS 1  
5 MCO 1  
6 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`?

# Load necessary packages  
library(dplyr)  
library(nycflights13)  
  
# Filter weather data for July 2013  
weather\_july\_2013 <- weather %>%  
 filter(month == 7,   
 year == 2013)  
  
# Summary statistics of temperature  
summary\_temp <- summary(weather\_july\_2013$temp)  
  
# Identify outliers in wind speed  
outliers\_wind\_speed <- boxplot.stats(weather\_july\_2013$wind\_speed)$out  
  
# Filter weather data for outliers in wind speed  
outliers\_weather <- weather\_july\_2013 %>%  
 filter(wind\_speed %in% outliers\_wind\_speed)  
  
# Output the result  
summary\_temp

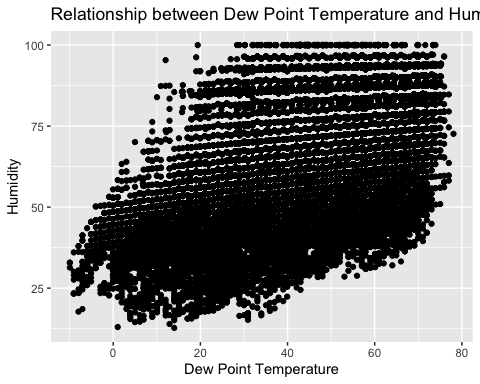
Min. 1st Qu. Median Mean 3rd Qu. Max.   
 64.04 75.02 78.98 80.07 84.20 100.04

outliers\_weather

# 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 20 17 93.9 66.9 41.3 260 21.9 27.6  
2 JFK 2013 7 20 19 84.2 71.1 71.9 300 24.2 32.2  
3 JFK 2013 7 23 18 82.0 73.0 74.2 310 25.3 66.7  
# ℹ 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,  
# time\_hour <dttm>

# Result explanation: Amongst the data, we found three standout values that we consider outliers. These outliers indicate instances of unusually high wind speeds that deviate from the typical range. These outliers provide interesting insights into the range of wind conditions that can occur in the area.   
  
  
#| label: problem-4B  
  
# Load necessary packages  
library(dplyr)  
library(ggplot2)  
library(nycflights13)  
  
# Filter weather data  
filtered\_weather <- weather %>%  
 select(dewp,   
 humid)  
  
# Create scatter plot  
ggplot(filtered\_weather, aes(x = dewp,   
 y = humid)) +  
 geom\_point() +  
 labs(x = "Dew Point Temperature",   
 y = "Humidity") +  
 ggtitle("Relationship between Dew Point Temperature and Humidity")

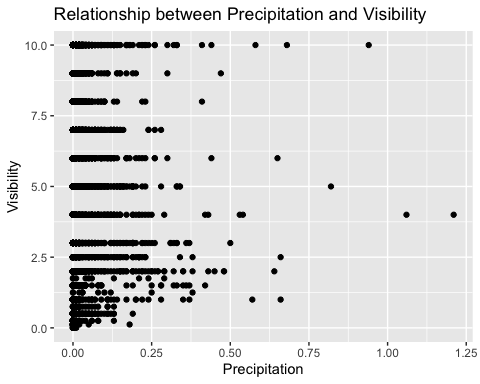
Warning: Removed 1 rows containing missing values (`geom\_point()`).



#Correlation  
filtered\_weather %>%   
 summarise(corr\_dh=cor(dewp,humid, use="complete.obs"))

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

# Result explanation: In the scatter plot, each point represents a specific observation. As we examine the plot, we can see that there is a general pattern. When the dew point temperature is low, the corresponding humidity tends to be relatively low as well. Similarly, when the dew point temperature is high, the humidity tends to be higher. The correlation is 0.51  
  
  
#| label: problem-4C  
  
# Load necessary packages  
library(dplyr)  
library(ggplot2)  
library(nycflights13)  
  
# Filter weather data  
filtered\_weather <- weather %>%  
 select(precip, visib) %>%  
 na.omit()  
  
# Create scatter plot  
ggplot(filtered\_weather, aes(x = precip,   
 y = visib)) +  
 geom\_point() +  
 labs(x = "Precipitation",   
 y = "Visibility") +  
 ggtitle("Relationship between Precipitation and Visibility")



#Correlation  
filtered\_weather %>%   
 summarise(corr\_dh=cor(precip,visib))

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

# Result explanation: In the scatter plot, we can observe several points where the visibility (visib) is 0 even when the precipitation (precip) values are lower than 0.25.This phenomenon suggests that there might be other factors impacting visibility besides precipitation alone. For example, fog, mist, or other atmospheric conditions can significantly reduce visibility even with relatively low levels of precipitation. The correlation is -0.32.

## 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.)

# Load necessary packages  
library(dplyr)  
library(nycflights13)  
  
# View columns using the names() function  
column\_names\_1 <- names(planes)  
print(column\_names\_1)

[1] "tailnum" "year" "type" "manufacturer" "model"   
[6] "engines" "seats" "speed" "engine"

# Filter planes with missing date of manufacture  
missing\_manufacturer <- planes %>%  
 filter(is.na(year))  
  
# Count the number of planes with missing date of manufacture  
count\_missing\_manufacturer <- nrow(missing\_manufacturer)  
  
# Output the result  
count\_missing\_manufacturer

[1] 70

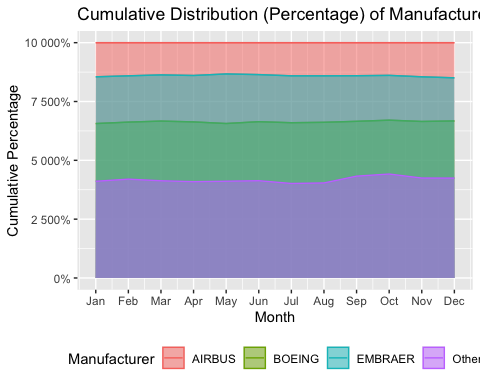
# View the final result result  
cat("There are", count\_missing\_manufacturer, "model(s) without a year.")

There are 70 model(s) without a year.

#| label: problem-5B  
  
# Count planes per manufacturer  
planes\_per\_manufacturer <- planes %>%  
 count(manufacturer) %>%  
 arrange(desc(n)) %>%   
 top\_n(5, n)  
  
# View the result  
print(planes\_per\_manufacturer)

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

# Result explanation: Clearly BOEING is the main manufacturer followed by AIRBUS  
  
  
#| label: problem-5C  
  
# Merge flights and planes datasets  
merged\_data <- flights %>%  
 left\_join(planes, by = "tailnum")  
  
# Filter data for flights from NYC in 2013  
filtered\_data <- merged\_data %>%  
 filter(year.x == 2013)  
  
# Recode manufacturer names and collapse rare vendors into "Other"  
recoded\_data <- filtered\_data %>%  
 mutate(manufacturer = case\_when(  
 manufacturer %in% c("BOEING", "AIRBUS", "EMBRAER") ~ manufacturer,  
 TRUE ~ "Other"  
 ))  
  
# Calculate the distribution of manufacturers by month  
manufacturer\_distribution <- recoded\_data %>%  
 group\_by(month, manufacturer) %>%  
 summarise(count = n(), .groups = "drop\_last") %>%  
 arrange(month, desc(count))  
  
# Calculate cumulative percentages by month  
manufacturer\_distribution <- manufacturer\_distribution %>%  
 group\_by(month) %>%  
 mutate(cumulative\_percentage = cumsum(count) / sum(count) \* 100)  
  
# Create cumulative line chart  
ggplot(manufacturer\_distribution, aes(x = month, y = cumulative\_percentage, group = manufacturer, color = manufacturer)) +  
 geom\_line() +  
 geom\_area(aes(fill = manufacturer), position = "identity", alpha = 0.5) +  
 labs(x = "Month", y = "Cumulative Percentage", fill = "Manufacturer", color = "Manufacturer") +  
 ggtitle("Cumulative Distribution (Percentage) of Manufacturers by Month in 2013") +  
 scale\_y\_continuous(labels = scales::percent) +  
 theme(legend.position = "bottom") +  
 scale\_x\_continuous(breaks = 1:12, labels = month.abb)



# Result explanation: Upon analyzing the chart, it appears that the distribution of manufacturers for airplanes flying from NYC in 2013 remained relatively stable over time, with minimal changes in market share among different manufacturers.

## 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?

# Filter flights from NYC airports in 2013  
nyc\_flights\_2013 <- flights %>%  
 filter(year == 2013)  
  
# 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)  
  
# Result of the plane and its year  
oldest\_plane

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

# Display the result  
cat("The oldest plane (specified by tailnum) that flew from New York City airports in 2013 is:", oldest\_plane$tailnum)

The oldest plane (specified by tailnum) that flew from New York City airports in 2013 is: N381AA

#| label: problem-6B  
  
# Filter flights from NYC airports in 2013  
nyc\_flights\_2013 <- flights %>%  
 filter(year == 2013)  
  
# Join with planes table to get the aircraft details  
flights\_with\_planes <- left\_join(nyc\_flights\_2013,   
 planes %>%   
 rename(year\_plane = year),   
 by = "tailnum")  
  
# Get unique tail numbers from flights  
unique\_tailnums <- unique(flights\_with\_planes$tailnum)  
  
# Filter planes for the unique tail numbers  
planes\_from\_nyc <- planes %>%  
 filter(tailnum %in% unique\_tailnums)  
  
# Count the number of airplanes  
num\_airplanes <- nrow(planes\_from\_nyc)  
  
# Display the result  
num\_airplanes

[1] 3322

# Display the result  
cat("The number of airplanes that flew from New York City and are included in the planes table is", num\_airplanes)

The number of airplanes that flew from New York City and are included in the planes table is 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.

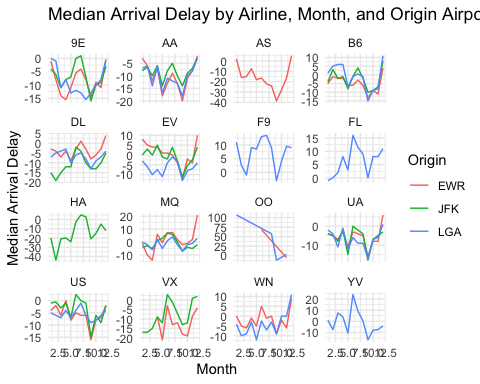
# Calculate median arrival delay on a month-by-month basis for each airport  
median\_arrival\_delay <- flights %>%  
 group\_by(month, dest) %>%  
 summarise(median\_delay = median(arr\_delay, na.rm = TRUE), .groups = "drop")  
  
# Calculate total median arrival delay per month  
total\_median\_delay <- median\_arrival\_delay %>%  
 group\_by(month) %>%  
 summarise(total\_median\_delay = sum(median\_delay, na.rm = TRUE))  
  
# Pivot the table to have airports as columns  
median\_arrival\_delay\_pivot <- pivot\_wider(median\_arrival\_delay, names\_from = dest, values\_from = median\_delay)  
  
# Add the total median delay column as the first column  
median\_arrival\_delay\_pivot <- median\_arrival\_delay\_pivot %>%  
 left\_join(total\_median\_delay, by = "month") %>%  
 select(month, Total = total\_median\_delay, everything())  
  
# Display the result  
median\_arrival\_delay\_pivot

# A tibble: 12 × 107  
 month Total ALB ATL AUS AVL BDL BHM BNA BOS BQN BTV  
 <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 1 91.5 6 -2 -2 23.5 -10 -11 1 -10 -5 -6  
 2 2 -106 -3 -1 2 NA -11 -10 1 -9 -3 0  
 3 3 -312. 0.5 -4 -4 NA -1 -1 -7 -8 -1 -1  
 4 4 -44 9 2 -1.5 -22 -3 -3 1 -9 9 -3  
 5 5 -578. 0 -4 -9 -10 -11 -23 -5 -8 -2 -3  
 6 6 -21.5 -14 3 -5.5 -2 -2 11 3 -8 1 -5  
 7 7 222. -13 7 1 2.5 49 21.5 2 -6 8 -3  
 8 8 -352 -19 1 -5.5 -2 -14 -3 -3 -10 -1 -5  
 9 9 -1098. -19 -6 -21 -3 -17 -17 -18 -11 -6 -4  
10 10 -390. 13 -4 -9 7 -11 4.5 -5 -10 -10 -9  
11 11 -426 -1 -1 -11 2.5 8 5 -5 -10 -8.5 -8  
12 12 514. 0 3 5 -2 5.5 31 4 -3 3 0  
# ℹ 95 more variables: BUF <dbl>, BUR <dbl>, BWI <dbl>, BZN <dbl>, CAE <dbl>,  
# CAK <dbl>, CHS <dbl>, CLE <dbl>, CLT <dbl>, CMH <dbl>, CRW <dbl>,  
# CVG <dbl>, DAY <dbl>, DCA <dbl>, DEN <dbl>, DFW <dbl>, DSM <dbl>,  
# DTW <dbl>, EGE <dbl>, EYW <dbl>, FLL <dbl>, GRR <dbl>, GSO <dbl>,  
# GSP <dbl>, HDN <dbl>, HNL <dbl>, HOU <dbl>, IAD <dbl>, IAH <dbl>,  
# IND <dbl>, JAC <dbl>, JAX <dbl>, LAS <dbl>, LAX <dbl>, LGB <dbl>,  
# MCI <dbl>, MCO <dbl>, MDW <dbl>, MEM <dbl>, MHT <dbl>, MIA <dbl>, …

#| label: problem-7B  
  
# Get the distinct origins from the flights table  
distinct\_origins <- distinct(flights, origin)  
  
# Display the results  
print(distinct\_origins)

# A tibble: 3 × 1  
 origin  
 <chr>   
1 EWR   
2 LGA   
3 JFK

# Calculate median arrival delay for each airline, month, and origin airport  
median\_arrival\_delay <- flights %>%  
 group\_by(carrier, month, origin) %>%  
 summarise(median\_delay = median(arr\_delay, na.rm = TRUE), .groups = "drop")  
  
# Plot median arrival delay for each airline  
ggplot(median\_arrival\_delay, aes(x = month, y = median\_delay, color = origin, group = interaction(carrier, origin))) +  
 geom\_line() +  
 facet\_wrap(vars(carrier), scales = "free\_y") +  
 labs(x = "Month", y = "Median Arrival Delay", color = "Origin", title = "Median Arrival Delay by Airline, Month, and Origin Airport") +  
 theme\_minimal()



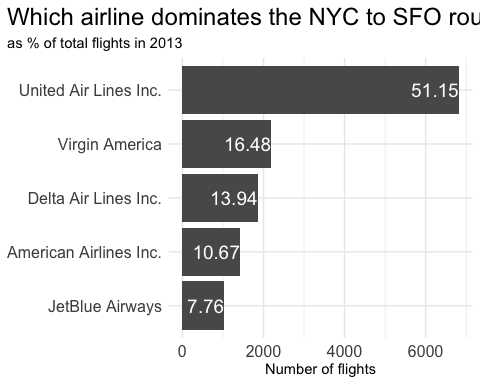
## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United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

# Join flights and airlines tables  
fly\_into\_sfo <- flights %>%  
 filter(dest == "SFO") %>%  
 group\_by(carrier) %>%  
 summarise(count = n(), .groups = "drop") %>%  
 inner\_join(airlines, by = c("carrier" = "carrier"))  
  
# Calculate percent of trips for each airline  
total\_trips <- sum(fly\_into\_sfo$count)  
fly\_into\_sfo <- fly\_into\_sfo %>%  
 mutate(percent\_trips = (count / total\_trips) \* 100)  
  
# Round the percentage values to two decimal places  
fly\_into\_sfo$percent\_trips <- round(fly\_into\_sfo$percent\_trips, 2)  
  
# Reorder the columns  
fly\_into\_sfo <- fly\_into\_sfo %>%  
 select(carrier, name, flights\_SFO = count, percent\_trips)  
  
# Sort by carrier in ascending order  
fly\_into\_sfo <- fly\_into\_sfo %>%  
 arrange(carrier)  
  
# Display the resulting dataframe  
fly\_into\_sfo

# A tibble: 5 × 4  
 carrier name flights\_SFO percent\_trips  
 <chr> <chr> <int> <dbl>  
1 AA American Airlines Inc. 1422 10.7   
2 B6 JetBlue Airways 1035 7.76  
3 DL Delta Air Lines Inc. 1858 13.9   
4 UA United Air Lines Inc. 6819 51.2   
5 VX 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, flights\_SFO)) %>%   
   
 ggplot() +  
   
 aes(x = flights\_SFO,   
 y = name) +  
   
 # a simple bar/column plot  
 geom\_col() +  
   
 # add labels, so each bar shows the % of total flights   
 geom\_text(aes(label = percent\_trips),  
 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 create a plot displaying cancellations of flights to SFO by month, carrier, and airport origin, we group the filtered dataset by month, carrier, and airport origin. Then, we calculate the count of cancelled flights for each combination of these variables. After arranging the dataset by month in ascending order, we proceed to plot the data using a suitable visualization method.

## 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-10A  
   
library(ggplot2)  
library(magrittr)

Attaching package: 'magrittr'

The following object is masked from 'package:purrr':  
  
 set\_names

The following object is masked from 'package:tidyr':  
  
 extract

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

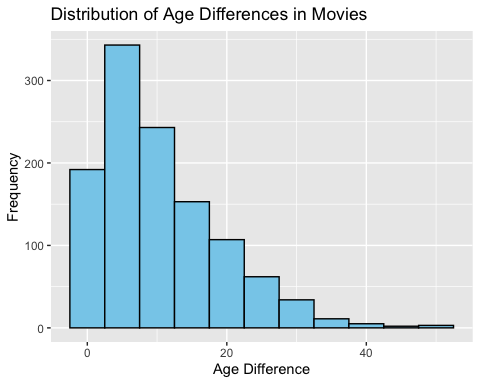
Rows: 1155 Columns: 13

── 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.

# Create a dataframe with age\_difference column  
age\_gap\_df <- data.frame(age\_difference = age\_gap$age\_difference)  
  
# Print the structure of age\_gap\_df  
str(age\_gap\_df)

'data.frame': 1155 obs. of 1 variable:  
 $ age\_difference: num 52 50 49 45 43 42 40 39 38 38 ...

# Filter out missing values  
age\_gap\_filtered <- age\_gap\_df %>%  
 filter(!is.na(age\_difference))  
  
# Plotting the histogram  
ggplot(data = age\_gap\_filtered, aes(x = age\_difference)) +  
 geom\_histogram(binwidth = 5, fill = "skyblue", color = "black") +  
 labs(x = "Age Difference", y = "Frequency") +  
 ggtitle("Distribution of Age Differences in Movies")



# Calculate the typical age difference (median)  
typical\_age\_difference <- median(age\_gap\_df$age\_difference)  
typical\_age\_difference

[1] 8

# Print the typical age difference in movies  
cat("The typical age difference in movies is approximately", typical\_age\_difference, "years.\n")

The typical age difference in movies is approximately 8 years.

#| label: problem-10B  
  
# Calculate the lower and upper age bounds based on the "half plus seven" rule  
age\_gap\_data <- age\_gap %>%  
 mutate(lower\_bound = floor(actor\_1\_age/2) + 7,  
 upper\_bound = (actor\_1\_age - 7) \* 2)  
  
# Count the number of actor/actress pairs that satisfy the "half plus seven" rule  
rule\_applies <- age\_gap\_data %>%  
 filter(actor\_2\_age >= lower\_bound, actor\_2\_age <= upper\_bound) %>%  
 tally()  
  
# Calculate the percentage of pairs that satisfy the rule  
percentage\_rule\_applies <- rule\_applies$n / nrow(age\_gap\_data) \* 100  
  
# Print the result  
cat("The 'half plus seven' rule applies in approximately", percentage\_rule\_applies, "% of the actor/actress pairs in the dataset.\n")

The 'half plus seven' rule applies in approximately 74.02597 % of the actor/actress pairs in the dataset.

#| label: problem-10C  
  
# Count the number of love interests per movie  
movie\_love\_interests <- age\_gap %>%  
 group\_by(movie\_name) %>%  
 summarise(num\_love\_interests = n\_distinct(couple\_number)) %>%  
 arrange(desc(num\_love\_interests))  
  
# Get the movie with the greatest number of love interests  
greatest\_love\_interests\_movie <- movie\_love\_interests$movie\_name[1]  
  
# Print the result  
cat("The movie with the greatest number of love interests is:", greatest\_love\_interests\_movie, "\n")

The movie with the greatest number of love interests is: Love Actually

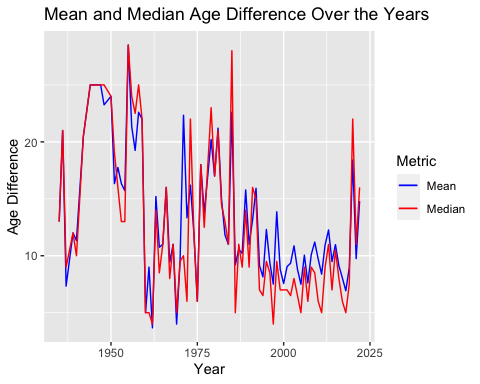
#| label: problem-10D  
  
#For Actor  
# Count the number of love interests per actor/actress  
actor\_love\_interests <- age\_gap %>%  
 group\_by(actor\_1\_name) %>%  
 summarise(num\_love\_interests = n\_distinct(couple\_number)) %>%  
 arrange(desc(num\_love\_interests))  
  
# Get the maximum number of love interests  
max\_love\_interests <- max(actor\_love\_interests$num\_love\_interests)  
  
# Get the actor/actress(es) with the greatest number of love interests  
greatest\_love\_interests\_actors <- actor\_love\_interests %>%  
 filter(num\_love\_interests == max\_love\_interests) %>%  
 pull(actor\_1\_name)  
  
# Print the result  
cat("The actor(es) with the greatest number of love interests is/are:", paste(greatest\_love\_interests\_actors, collapse = ", "), "\n")

The actor(es) with the greatest number of love interests is/are: Pierce Brosnan, Roger Moore

#For actress  
# Count the number of love interests per actor/actress  
actor\_love\_interests <- age\_gap %>%  
 group\_by(actor\_2\_name) %>%  
 summarise(num\_love\_interests = n\_distinct(couple\_number)) %>%  
 arrange(desc(num\_love\_interests))  
  
# Get the maximum number of love interests  
max\_love\_interests <- max(actor\_love\_interests$num\_love\_interests)  
  
# Get the actor/actress(es) with the greatest number of love interests  
greatest\_love\_interests\_actors <- actor\_love\_interests %>%  
 filter(num\_love\_interests == max\_love\_interests) %>%  
 pull(actor\_2\_name)  
  
# Print the result  
cat("The actress(es) with the greatest number of love interests is/are:", paste(greatest\_love\_interests\_actors, collapse = ", "), "\n")

The actress(es) with the greatest number of love interests is/are: Diane Keaton, Keira Knightley

#| label: problem-10E  
  
# Convert release\_year to numeric format  
age\_gap$release\_year <- as.numeric(age\_gap$release\_year)  
  
# Filter data for the years 1935 to 2022  
filtered\_data <- age\_gap %>% filter(release\_year >= 1935 & release\_year <= 2022)  
  
# Calculate the mean and median age difference for each year  
age\_diff\_by\_year <- filtered\_data %>%  
 group\_by(release\_year) %>%  
 summarise(mean\_age\_diff = mean(age\_difference),  
 median\_age\_diff = median(age\_difference))  
  
# Plot the trend of mean and median age difference over the years  
plot <- ggplot(age\_diff\_by\_year, aes(x = release\_year)) +  
 geom\_line(aes(y = mean\_age\_diff, color = "Mean")) +  
 geom\_line(aes(y = median\_age\_diff, color = "Median")) +  
 labs(x = "Year", y = "Age Difference", color = "Metric") +  
 ggtitle("Mean and Median Age Difference Over the Years") +  
 scale\_color\_manual(values = c("Mean" = "blue", "Median" = "red"))  
  
plot # Display the plot



comment\_text <- "In general, it can be observed that the mean and median age difference did not remain constant over the years. There are a couple of key trends that can be observed from the plot:  
  
1) Historically, there tended to be a larger age difference between movie love interests compared to more recent years. This suggests that in earlier times, movies portrayed relationships with larger age gaps between the characters.  
  
2) However, it is important to note that there is no clear downward pattern over time. In fact, in recent years, there appears to be an increase in the mean and median age difference, indicating that movies have depicted relationships with larger age disparities again.  
  
These observations suggest that while there was a general decrease in age differences between movie love interests in the past, there is no consistent downward trend. The age difference in movies seems to have varied over time, with recent years showing an upward trend in the mean and median age difference"  
  
cat(comment(comment\_text))  
  
#| label: problem-10F  
  
# Filter the data for same-gender love interests  
same\_gender\_love\_interests <- age\_gap %>%  
 filter(character\_1\_gender == character\_2\_gender)  
  
# Calculate the frequency of same-gender love interests  
frequency\_same\_gender <- nrow(same\_gender\_love\_interests) / nrow(age\_gap) \* 100  
  
# Print the result  
cat("Hollywood depicts same-gender love interests in approximately", frequency\_same\_gender, "% of the relationships.\n")

Hollywood depicts same-gender love interests in approximately 1.991342 % of the relationships.

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: Guido Bozzano
* Approximately how much time did you spend on this problem set: 12 or more
* What, if anything, gave you the most trouble: Che charts

**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.