Homerwork 1

Naoya Kinoshita

5/14/23

# Data Manipulation

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

- Q1: Had an arrival delay of two or more hours (\> 120 minutes)  
 Q1: Answer: 10,034 Flights  
  
- Q2: Flew to Houston (IAH or HOU)  
 Q2: Answer: 7,198 Flights  
  
- Q3: Were operated by United (`UA`), American (`AA`), or Delta (`DL`)  
 Q3: Answer: 13,954 Flights  
  
- Q4: Departed in summer (July, August, and September)  
 Q4: Answer: 86,326 Flights  
  
- Q5: Arrived more than two hours late, but didn't leave late  
 Q5: Answer: 29 Flights  
   
- Q6: Were delayed by at least an hour, but made up over 30 minutes in flight  
 Q6: Answer: 1,948 Flights

# Q1: The flights delayed more than 120 minutes  
q1\_flights <- flights %>%  
 filter(arr\_delay > 120)  
  
# Q2: The flights whihch destination are Huston  
q2\_flights <- flights %>%  
 filter(dest %in% c("IAH", "HOU"))  
  
# Q3: Were operated  
q3\_flights <- flights %>%  
 filter(carrier %in% c("UA", "AA", "DL"))  
  
# Q4: The flights depearted in July, August, September  
q4\_flights <- flights %>%  
 filter(month %in% c(7, 8, 9))  
  
# Q5: 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  
q5\_flights <- flights %>%  
 filter(arr\_delay > 120,  
 dep\_delay <= 0)  
  
# Q6: Were delayed by at least an hour, but made up over 30 minutes in flight  
q6\_flights <- flights %>%  
 filter(dep\_delay >= 60,  
 (arr\_delay - dep\_delay) > 30)  
  
# showing answer to each question  
answer <- data.frame(  
 Q1 = nrow(q1\_flights),  
 Q2 = nrow(q2\_flights),  
 Q3 = nrow(q3\_flights),  
 Q4 = nrow(q4\_flights),  
 Q5 = nrow(q5\_flights),  
 Q6 = nrow(q6\_flights)  
)  
  
print(answer)

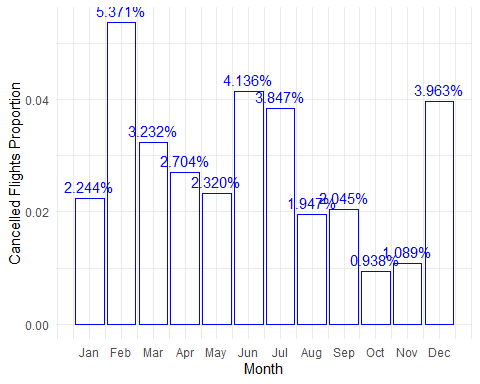
Q1 Q2 Q3 Q4 Q5 Q6  
1 10034 9313 139504 86326 29 1948

## 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

Answer : When organizing the data and examining the bar graph, it was found that the month with the highest percentage of canceled flights was February, at 5.371%. Conversely, the month with the lowest percentage of canceled flights was October, at 0.938%.

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

# calculating the proportion of flights that cancelled  
cancelled\_proportion <- flights %>%  
 group\_by(month) %>%  
 summarise(cancelled\_proportion = sum(is.na(dep\_delay) | is.na(arr\_delay)) / n())  
  
# Making bar graph  
ggplot(cancelled\_proportion, aes(x = month, y = cancelled\_proportion)) +  
 geom\_bar(stat = "identity", fill = "transparent", color = "blue") +  
 geom\_text(aes(label = scales::percent(cancelled\_proportion)), vjust = -0.5, color = "blue") +  
 scale\_x\_continuous(breaks = 1:12, labels = month.abb) +  
 labs(x = "Month", y = "Cancelled Flights Proportion") +  
 theme\_minimal()



## 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.

Answer: Based on the table as shown below, N725MQ is the plane which flew most during 2013 and the number of the flights is 575.

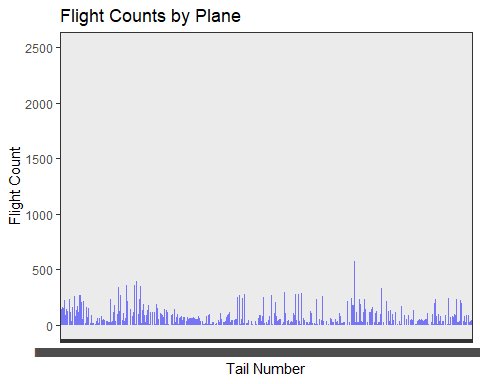
# Filter the flights dataset for the year 2013  
flights\_2013 <- flights[flights$year == 2013, ]  
  
# Left join the flights dataset with the planes dataset  
merged\_table <- dplyr::left\_join(flights\_2013, planes, by = "tailnum")  
  
# Count the number of flights for each plane  
flight\_counts <- merged\_table %>% dplyr::count(tailnum, sort = TRUE)  
  
# Set options to display more columns in the table  
options(width = 100)  
  
# Set options to display all digits for numeric columns  
options(digits = 22)  
  
# Identify the top 10 planes with the highest number of flights  
top\_10\_planes <- head(flight\_counts, 10)  
  
# Display the top 10 planes in a table  
top\_10\_planes

# A tibble: 10 × 2  
 tailnum n  
 <chr> <int>  
 1 <NA> 2512  
 2 N725MQ 575  
 3 N722MQ 513  
 4 N723MQ 507  
 5 N711MQ 486  
 6 N713MQ 483  
 7 N258JB 427  
 8 N298JB 407  
 9 N353JB 404  
10 N351JB 402

kable(top\_10\_planes)

| tailnum | n |
| --- | --- |
| NA | 2512 |
| N725MQ | 575 |
| N722MQ | 513 |
| N723MQ | 507 |
| N711MQ | 486 |
| N713MQ | 483 |
| N258JB | 427 |
| N298JB | 407 |
| N353JB | 404 |
| N351JB | 402 |

# Create a bar graph of flight counts for each plane  
library(ggplot2)  
  
ggplot(data = flight\_counts, aes(x = tailnum, y = n)) +  
 geom\_bar(stat = "identity", fill = "blue", alpha = 0.5) +  
 labs(x = "Tail Number", y = "Flight Count", title = "Flight Counts by Plane") +  
 theme\_bw()

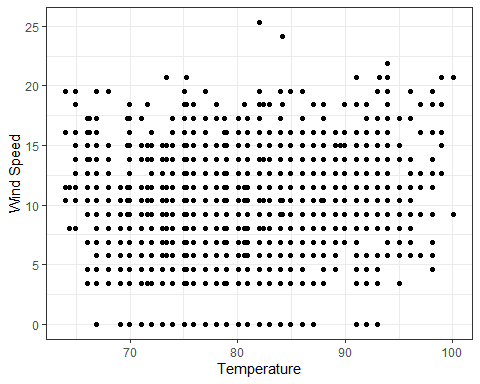


## Problem 4: The nycflights13 package includes a table (weather) that describes the weather during 2013. Use that table to answer the following questions:

- Q1: What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
 Q1: Answer: It seems that two data which exceeds wind speed as of 22.5 are the outliers in terms of the 'wind\_speed' variable.  
   
- Q2: What is the relationship between `dewp` and `humid`?  
 Q2: Answer: Correration coefficient between dewp and humid is 51.2%. And as shown in the graph, it seems slightly positive relationship between dewp and humid.  
  
- What is the relationship between `precip` and `visib`?  
 Q3: Answer: Correration coefficient between precip and visib is -32.0%. And as shown in the graph, it seems slightly negative relationship between dewp and humid.

# Get the weather dataset  
weather <- nycflights13::weather  
  
# Extract data for July 2013  
july\_2013\_weather <- subset(weather, month == 7 & year == 2013)  
  
# Visualize the distribution of `temp` and its relationship with `wind\_speed`  
library(ggplot2)  
ggplot(july\_2013\_weather, aes(x = temp, y = wind\_speed)) +  
 geom\_point() +  
 labs(x = "Temperature", y = "Wind Speed") +  
 theme\_bw()

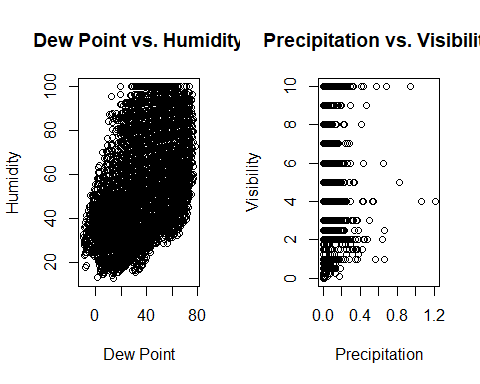
Warning: Removed 2 rows containing missing values (`geom\_point()`).



# Create a table to display correlation coefficients  
cor\_table <- matrix(NA, nrow = 2, ncol = 2,  
 dimnames = list(c("Dew Point", "Precipitation"),  
 c("Humidity", "Visibility")))  
  
# Calculate correlation coefficients  
cor\_table[1, 2] <- cor(weather$dewp, weather$humid)  
cor\_table[2, 2] <- cor(weather$precip, weather$visib)  
  
# Display the correlation coefficients table  
cor\_table

Humidity Visibility  
Dew Point NA NA  
Precipitation NA -0.3199117771945496069286

# Create scatter plots for dewp vs. humid and precip vs. visib  
par(mfrow = c(1, 2))  
  
# Scatter plot for dewp vs. humid  
plot(weather$dewp, weather$humid, xlab = "Dew Point", ylab = "Humidity",  
 main = "Dew Point vs. Humidity")  
  
# Scatter plot for precip vs. visib  
plot(weather$precip, weather$visib, xlab = "Precipitation", ylab = "Visibility",  
 main = "Precipitation vs. Visibility")



# Reset the plotting layout  
par(mfrow = c(1, 1))  
  
# Calculate correlation and additional statistics for dewp vs. humid  
cor\_dewp\_humid <- cor.test(weather$dewp, weather$humid)  
cor\_dewp\_humid\_table <- data.frame(  
 Variable1 = "Dew Point",  
 Variable2 = "Humidity",  
 Correlation = cor\_dewp\_humid$estimate,  
 P\_Value = cor\_dewp\_humid$p.value,  
 stringsAsFactors = FALSE  
)  
  
# Calculate correlation and additional statistics for precip vs. visib  
cor\_precip\_visib <- cor.test(weather$precip, weather$visib)  
cor\_precip\_visib\_table <- data.frame(  
 Variable1 = "Precipitation",  
 Variable2 = "Visibility",  
 Correlation = cor\_precip\_visib$estimate,  
 P\_Value = cor\_precip\_visib$p.value,  
 stringsAsFactors = FALSE  
)  
  
# Display the correlation tables  
kable(cor\_dewp\_humid\_table)

|  | Variable1 | Variable2 | Correlation | P\_Value |
| --- | --- | --- | --- | --- |
| cor | Dew Point | Humidity | 0.512195202542813698976 | 0 |

kable(cor\_precip\_visib\_table)

|  | Variable1 | Variable2 | Correlation | P\_Value |
| --- | --- | --- | --- | --- |
| cor | Precipitation | Visibility | -0.3199117771945496069286 | 0 |

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

- Q1: How many planes have a missing date of manufacture?  
 Q1: Answer: There are 70 planes which has missing date of manufacture.  
  
- Q2: What are the five most common manufacturers?  
 Q2: Answer: As shown in the table below, Boeing, Airbus Industrie, Bombardier Inc, Airbus, and Embraer are the five most common manufactures.   
  
- Q3: 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.)  
 Q3: As shown in the graph, Production volume is high from March through August and relatively low in the other months. February has the lowest airplane production volume due to the shorter number of days than the other months.

# Load the flights and planes tables  
flights <- nycflights13::flights  
planes <- nycflights13::planes  
  
# Count the number of planes with missing date of manufacture  
missing\_manufacture <- sum(is.na(planes$year))  
  
# Identify the five most common manufacturers  
top\_manufacturers <- head(sort(table(planes$manufacturer), decreasing = TRUE), 5)  
  
# Create a data frame for flights from NYC in 2013  
flights\_2013\_NYC <- flights[flights$year == 2013 & flights$origin %in% c("JFK", "LGA", "EWR"), ]  
flights\_2013\_NYC <- merge(flights\_2013\_NYC, planes, by = "tailnum")  
  
# Recode manufacturer and collapse rare vendors into "Other" category  
library(dplyr)  
flights\_2013\_NYC <- flights\_2013\_NYC %>%  
 mutate(manufacturer = case\_when(  
 manufacturer %in% top\_manufacturers ~ as.character(manufacturer),  
 TRUE ~ "Other"  
 ))  
  
# Checking lenght of flights\_2013\_NYC$year and flights\_2013\_NYC$manufacturer  
nrow(flights\_2013\_NYC)

[1] 284170

length(flights\_2013\_NYC$year)

[1] 0

length(flights\_2013\_NYC$manufacturer)

[1] 284170

# Display the results  
missing\_manufacture

[1] 70

top\_manufacturers

BOEING AIRBUS INDUSTRIE BOMBARDIER INC AIRBUS EMBRAER   
 1630 400 368 336 299

# Display the results using kable  
kable(data.frame(missing\_manufacture), caption = "Planes with Missing Date of Manufacture")

Planes with Missing Date of Manufacture

| missing\_manufacture |
| --- |
| 70 |

kable(data.frame(top\_manufacturers), caption = "Top 5 Manufacturers")

Top 5 Manufacturers

| Var1 | Freq |
| --- | --- |
| BOEING | 1630 |
| AIRBUS INDUSTRIE | 400 |
| BOMBARDIER INC | 368 |
| AIRBUS | 336 |
| EMBRAER | 299 |

# Filter flights from NYC in 2013 and merge with planes table  
flights\_2013\_NYC <- flights %>%  
 filter(year == 2013, origin %in% c("JFK", "LGA", "EWR")) %>%  
 left\_join(planes, by = "tailnum")  
  
# Recode manufacturer and collapse rare vendors into "Other" category  
flights\_2013\_NYC <- flights\_2013\_NYC %>%  
 mutate(manufacturer = case\_when(  
 manufacturer %in% c("BOEING", "AIRBUS") ~ as.character(manufacturer),  
 TRUE ~ "Other"  
 ))  
  
# Calculate the manufacturer distribution by year  
manufacturer\_distribution <- flights\_2013\_NYC %>%  
 group\_by(month, manufacturer) %>%  
 summarize(count = n()) %>%  
 ungroup()

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

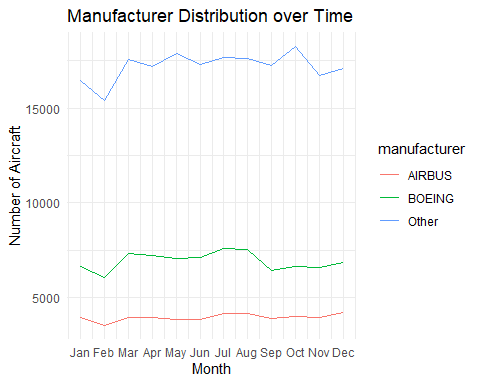
# Pivot the data to create a table format  
manufacturer\_table <- manufacturer\_distribution %>%  
 pivot\_wider(names\_from = manufacturer, values\_from = count, values\_fill = 0)  
  
# Display the manufacturer distribution over time  
kable(manufacturer\_table)

| month | AIRBUS | BOEING | Other |
| --- | --- | --- | --- |
| 1 | 3916 | 6623 | 16465 |
| 2 | 3515 | 6048 | 15388 |
| 3 | 3948 | 7312 | 17574 |
| 4 | 3949 | 7197 | 17184 |
| 5 | 3826 | 7063 | 17907 |
| 6 | 3831 | 7085 | 17327 |
| 7 | 4145 | 7597 | 17683 |
| 8 | 4131 | 7557 | 17639 |
| 9 | 3872 | 6425 | 17277 |
| 10 | 4013 | 6613 | 18263 |
| 11 | 3954 | 6557 | 16757 |
| 12 | 4202 | 6835 | 17098 |

# Aggregate the data by month and manufacturer  
manufacturer\_monthly <- flights\_2013\_NYC %>%  
 group\_by(month, manufacturer) %>%  
 summarize(count = n()) %>%  
 ungroup()

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

# Filter the top 5 manufacturers based on total aircraft count  
top\_5\_manufacturers <- manufacturer\_monthly %>%  
 group\_by(manufacturer) %>%  
 summarise(total\_count = sum(count)) %>%  
 top\_n(5, total\_count) %>%  
 select(manufacturer)  
  
# Merge with the aggregated data  
top\_manufacturer\_monthly <- manufacturer\_monthly %>%  
 filter(manufacturer %in% top\_5\_manufacturers$manufacturer)  
  
# Create a line graph  
ggplot(top\_manufacturer\_monthly, aes(x = month, y = count, group = manufacturer, color = manufacturer)) +  
 geom\_line() +  
 labs(x = "Month", y = "Number of Aircraft", title = "Manufacturer Distribution over Time") +  
 scale\_x\_continuous(breaks = 1:12, labels = month.abb) +  
 theme\_minimal()



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

- Q1: What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
 Q1: Answer: Oldest plane is N381AA, which manufactured in 1956.  
  
- Q2:How many airplanes that flew from New York City are included in the planes table?  
 Q2: Answer: There are 336776 airplanes that flew from New York.

# Create the 'flights\_planes' object by merging the 'flights' and 'planes' tables  
flights\_planes <- merge(flights, planes, by.x = "tailnum", by.y = "tailnum")  
  
# Count the number of airplanes from NYC included in the planes table  
num\_airplanes\_01 <- flights\_planes %>%  
 arrange(year.y) %>%  
 group\_by(year.y) %>%  
 count() %>%  
 top\_n(15)

Selecting by n

# Print the result  
kable(num\_airplanes\_01, format = "markdown")

| year.y | n |
| --- | --- |
| 1956 | 22 |
| 1959 | 117 |
| 1963 | 52 |
| 1965 | 4 |
| 1967 | 22 |
| 1968 | 43 |
| 1972 | 25 |
| 1973 | 22 |
| 1974 | 103 |
| 1975 | 92 |
| 1976 | 544 |
| 1977 | 187 |
| 1978 | 26 |
| 1979 | 64 |
| 1980 | 109 |
| 1983 | 246 |
| 1984 | 115 |
| 1985 | 994 |
| 1986 | 1800 |
| 1987 | 3506 |
| 1988 | 3856 |
| 1989 | 3116 |
| 1990 | 5394 |
| 1991 | 6002 |
| 1992 | 7696 |
| 1993 | 3358 |
| 1994 | 2714 |
| 1995 | 1378 |
| 1996 | 1799 |
| 1997 | 6008 |
| 1998 | 17231 |
| 1999 | 19373 |
| 2000 | 22334 |
| 2001 | 26889 |
| 2002 | 23741 |
| 2003 | 15069 |
| 2004 | 15706 |
| 2005 | 14369 |
| 2006 | 13203 |
| 2007 | 15300 |
| 2008 | 17878 |
| 2009 | 6632 |
| 2010 | 3797 |
| 2011 | 6046 |
| 2012 | 7252 |
| 2013 | 4630 |
| NA | 5306 |

# Filter airplanes with year.y equal to 1956  
num\_airplanes <- flights\_planes %>%  
 filter(year.y == 1956) %>%  
 count(tailnum)  
  
# Print the result  
kable(num\_airplanes, format = "markdown")

| tailnum | n |
| --- | --- |
| N381AA | 22 |

# Count the number of airplanes that flew from New York City  
num\_planes\_03 <- sum(planes$origin %in% c("JFK", "LGA", "EWR"))

Warning: Unknown or uninitialised column: `origin`.

# Print the result  
kable(num\_planes\_03)

| x |
| --- |
| 0 |

# Count the number of airplanes from NYC included in the flights table  
num\_airplanes <- flights %>%  
 filter(origin %in% c("JFK", "LGA", "EWR")) %>%  
 group\_by(year) %>%  
 summarize(total = n()) %>%  
 arrange(year)  
  
# Print the table using kable  
kable(num\_airplanes, format = "markdown", caption = "Number of Airplanes from NYC by Year")

Number of Airplanes from NYC by Year

| year | total |
| --- | --- |
| 2013 | 336776 |

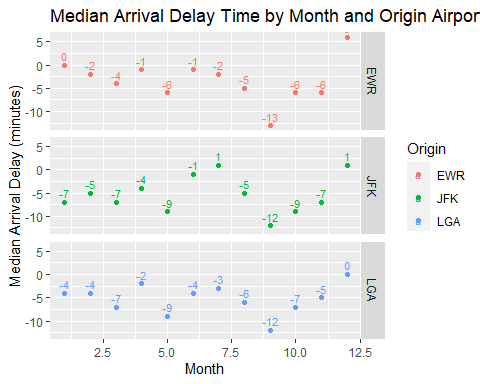
## Problem 7: Use the nycflights13 to answer the following questions:

- Q1: What is the median arrival delay on a month-by-month basis in each airport?  
- Q2: For each airline, plot the median arrival delay for each month and origin airport.  
  
 Q1&Q2: Answer: Both information can be found in the graph as shown below.

# Data preparation  
data(flights)  
flights <- select(flights, month, origin, arr\_delay)  
  
# Calculate median arrival delay time on a month-by-month basis in each airport  
median\_delay <- flights %>%  
 group\_by(month, origin) %>%  
 summarise(median\_delay = median(arr\_delay, na.rm = TRUE))

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

# Create the graph  
graph <- ggplot(median\_delay, aes(x = month, y = median\_delay, color = origin)) +  
 geom\_point() +  
 geom\_text(aes(label = median\_delay), vjust = -0.5, size = 3) +  
 labs(x = "Month", y = "Median Arrival Delay (minutes)", color = "Origin") +  
 facet\_grid(rows = vars(origin)) +  
 ggtitle("Median Arrival Delay Time by Month and Origin Airport")  
  
# Print the graph  
print(graph)



## 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.

Q8: Answer: As shown in the tables, United Airlines. INC is the most airlines which flew to SFO.

# Load the necessary data  
data(flights)  
data(airlines)  
  
# Join flights and airlines tables and filter for flights to SFO  
fly\_into\_sfo <- flights %>%  
 inner\_join(airlines, by = "carrier") %>%  
 filter(dest == "SFO") %>%  
   
 # Group the data by airline name  
 group\_by(name) %>%  
   
 # Calculate the count and percentage of flights to SFO for each airline  
 summarise(count = n(), percent = sprintf("%.2f%%", n() / nrow(.) \* 100)) %>%  
   
 # Arrange the data in descending order of count  
 arrange(desc(count))  
  
# Print the resulting dataframe  
kable(fly\_into\_sfo)

| name | count | percent |
| --- | --- | --- |
| United Air Lines Inc. | 6819 | 51.15% |
| Virgin America | 2197 | 16.48% |
| Delta Air Lines Inc. | 1858 | 13.94% |
| American Airlines Inc. | 1422 | 10.67% |
| JetBlue Airways | 1035 | 7.76% |

And here is some bonus ggplot code to plot your dataframe

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

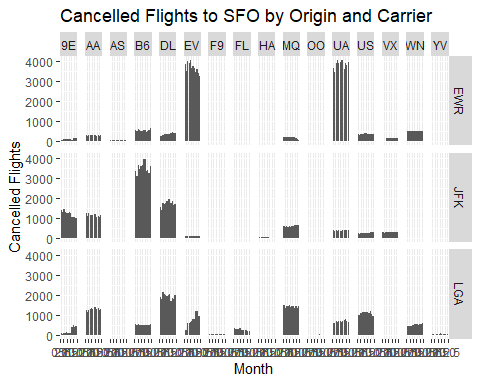
We can create multiple graphs to show the number of canceled flights per month for each combination of Origin (EWR, JFK, LGA) and Carrier using the following steps. I will also provide the code separately.

1. Aggregate the number of canceled flights by Origin, Carrier, and month using the **cancellations** dataframe.
2. Create the base of the graph using the **ggplot()** function.
3. Display the number of canceled flights as bar graphs using the **geom\_col()** function.
4. Create a grid of graphs based on the combination of Origin and Carrier using the **facet\_grid()** function.
5. Set the graph title and axis labels using the **labs()** function.

# Create the 'cancellations' object with your desired data  
cancellations <- flights  
  
# Group cancellations by Origin, Carrier, and month  
cancellation\_counts <- cancellations %>%  
 group\_by(origin, carrier, month) %>%  
 summarise(cancelled\_flights = n())

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

# Create the graph  
ggplot(cancellation\_counts, aes(x = month, y = cancelled\_flights)) +  
 geom\_col() +  
 facet\_grid(origin ~ carrier) +  
 labs(title = "Cancelled Flights to SFO by Origin and Carrier",  
 x = "Month",  
 y = "Cancelled Flights")



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



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

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

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

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

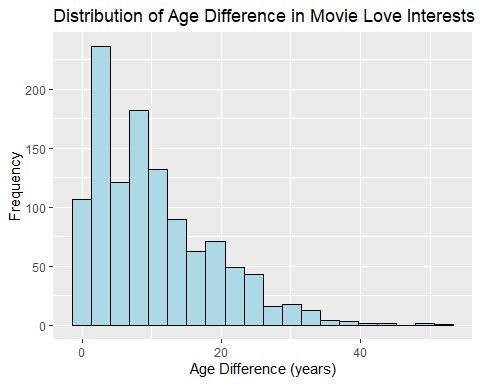
The following is a data dictionary of the variables used

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

# Load required packages  
library(tidyverse)  
  
# Read the dataset  
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\_gender, character\_2\_gender  
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.

# Q1: Explore the dataset  
  
# Q2: Distribution of age\_difference  
ggplot(age\_gaps, aes(x = age\_difference)) +  
 geom\_histogram(bins = 20, fill = "lightblue", color = "black") +  
 labs(title = "Distribution of Age Difference in Movie Love Interests",  
 x = "Age Difference (years)",  
 y = "Frequency")



# Calculate the typical age\_difference in movies  
typical\_age\_difference <- median(age\_gaps$age\_difference)  
  
# Q3: Count how frequently the half plus seven rule applies in this dataset  
rule\_applies <- age\_gaps %>%  
 filter(actor\_2\_age > actor\_1\_age / 2 + 7,  
 actor\_2\_age < (actor\_1\_age - 7) \* 2) %>%  
 nrow()  
  
# Q4: Movie with the greatest number of love interests  
most\_love\_interests <- age\_gaps %>%  
 group\_by(movie\_name) %>%  
 summarise(total\_love\_interests = n()) %>%  
 arrange(desc(total\_love\_interests)) %>%  
 slice(1)  
  
# Q5: Actors/actresses with the greatest number of love interests  
most\_love\_interests\_actors <- age\_gaps %>%  
 group\_by(actor\_1\_name) %>%  
 summarise(total\_love\_interests = n()) %>%  
 arrange(desc(total\_love\_interests)) %>%  
 slice(1)  
  
# Q6: Mean and median age difference over the years  
age\_difference\_over\_years <- age\_gaps %>%  
 group\_by(release\_year) %>%  
 summarise(mean\_age\_difference = mean(age\_difference),  
 median\_age\_difference = median(age\_difference))  
  
# Q7: Frequency of same-gender love interests  
same\_gender\_love\_interests <- age\_gaps %>%  
 filter(character\_1\_gender == character\_2\_gender) %>%  
 count()  
  
# Print the results  
typical\_age\_difference

[1] 8

rule\_applies

[1] 795

most\_love\_interests

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

most\_love\_interests\_actors

# A tibble: 1 × 2  
 actor\_1\_name total\_love\_interests  
 <chr> <int>  
1 Keanu Reeves 24

age\_difference\_over\_years

# A tibble: 82 × 3  
 release\_year mean\_age\_difference median\_age\_difference  
 <dbl> <dbl> <dbl>  
 1 1935 13 13   
 2 1936 21 21   
 3 1937 7.33 9   
 4 1939 12 12   
 5 1940 11.3 10   
 6 1942 20.5 20.5  
 7 1944 25 25   
 8 1946 25 25   
 9 1947 25 25   
10 1948 23.2 25   
# ℹ 72 more rows

same\_gender\_love\_interests

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

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?
* Answer:
* As shown in the graph as below, The age difference is predominantly concentrated between 0 and 20 in the data, and beyond 20, there is a tendency for the age difference to decrease as it increases.
* The typical age\_difference is 8.
* 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?

Answer: The rule applies to 795 movies.

* Which movie has the greatest number of love interests?
* Answer: It is Love actually and it has 7 love interests.
* Which actors/ actresses have the greatest number of love interests in this dataset?
* Answer: The actor who has greatest number of love interests is Keanu Reeves and it has 24.
* Is the mean/median age difference staying constant over the years (1935 - 2022)?
* Answer: No, mean and median age difference is increasing over the years. Detailes are shown in the table.
* How frequently does Hollywood depict same-gender love interests?
* Answer: 23

# 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: ChatGPT
* Approximately how much time did you spend on this problem set: 5-6hr
* What, if anything, gave you the most trouble: ANSWER HERE

**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

1313: 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.

813: 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.

513: 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.