Homerwork 1

PATRICK COOK

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

glimpse(flights)

Rows: 336,776  
Columns: 19  
$ year <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…  
$ month <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
$ day <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
$ dep\_time <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 558, …  
$ sched\_dep\_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 600, …  
$ dep\_delay <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1…  
$ arr\_time <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 849,…  
$ sched\_arr\_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 851,…  
$ arr\_delay <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1…  
$ carrier <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6", "…  
$ flight <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301, 4…  
$ tailnum <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N394…  
$ origin <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LGA",…  
$ dest <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IAD",…  
$ air\_time <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149, 1…  
$ distance <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 733, …  
$ hour <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6…  
$ minute <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0…  
$ time\_hour <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-01 0…

# Had an arrival delay of two or more hours (> 120 minutes)  
problem1A <- flights %>%  
   
 filter(arr\_delay >= 120,   
  
# Flew to Houston (IAH or HOU)  
 dest %in% c("IAH", "HOU"),   
  
# Were operated by United (`UA`), American (`AA`), or Delta (`DL`)  
 carrier %in% c("UA", "AA", "DL"),   
  
# Departed in summer (July, August, and September)  
 month %in% c(7, 8, 9),  
  
# Arrived more than two hours late, but didn't leave late (there are no flights that did this, every flight that arrived late by at least 2 hours was already delayed on departure)  
   
 dep\_delay <= 0)  
  
  
  
# Were delayed by at least an hour, but made up over 30 minutes in flight (I have put a # in front of)  
problem1B <- flights %>%  
 filter(  
 dest %in% c("IAH", "HOU"),   
 carrier %in% c("UA", "AA", "DL"),   
 month %in% c(7, 8, 9),  
 dep\_delay >= 60,  
 arr\_delay <= 30  
 )  
   
  
problem1A

# A tibble: 0 × 19  
# ℹ 19 variables: year <int>, month <int>, day <int>, dep\_time <int>,  
# sched\_dep\_time <int>, dep\_delay <dbl>, arr\_time <int>,  
# sched\_arr\_time <int>, 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>

problem1B

# A tibble: 2 × 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 8 28 2122 2005 77 2318 2255  
2 2013 9 6 2006 1859 67 2220 2153  
# 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))

# A tibble: 8,255 × 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 NA 1630 NA NA 1815  
 2 2013 1 1 NA 1935 NA NA 2240  
 3 2013 1 1 NA 1500 NA NA 1825  
 4 2013 1 1 NA 600 NA NA 901  
 5 2013 1 2 NA 1540 NA NA 1747  
 6 2013 1 2 NA 1620 NA NA 1746  
 7 2013 1 2 NA 1355 NA NA 1459  
 8 2013 1 2 NA 1420 NA NA 1644  
 9 2013 1 2 NA 1321 NA NA 1536  
10 2013 1 2 NA 1545 NA NA 1910  
# ℹ 8,245 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>

# What months had the highest and lowest % of cancelled flights?  
  
problem2 <- flights %>%  
 group\_by(month) %>%  
 summarise(total\_flights = n(),  
 number\_cancelled = sum((is.na(dep\_time))),  
 proportion\_cancelled = number\_cancelled/total\_flights\*100)  
 #Calculating the total flights, total cancelled flights, and proportion of flights cancelled for each of the 12 months in 2013  
problem2

# A tibble: 12 × 4  
 month total\_flights number\_cancelled proportion\_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 0.817  
11 11 27268 233 0.854  
12 12 28135 1025 3.64

highest\_cancelled\_month <- problem2 %>%   
 filter(proportion\_cancelled == max(proportion\_cancelled)) %>%   
 pull(month, proportion\_cancelled)  
 #Pulling the month with the highest proportion of cancelled flights, which is `month 2 (February)  
highest\_cancelled\_month

5.05390565508396   
 2

lowest\_cancelled\_month <- problem2 %>%   
 filter(proportion\_cancelled == min(proportion\_cancelled)) %>%   
 pull(month, proportion\_cancelled)  
 #Pulling the month with the lowest proportion of cancelled flights, which is month 10 (November)  
lowest\_cancelled\_month

0.816919934923327   
 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.

#Left joining tables  
joined <- left\_join(planes, flights, by = 'tailnum') %>%   
 group\_by(tailnum)  
  
joined

# A tibble: 284,170 × 27  
# Groups: tailnum [3,322]  
 tailnum year.x type manufacturer model engines seats speed engine year.y  
 <chr> <int> <chr> <chr> <chr> <int> <int> <int> <chr> <int>  
 1 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 2 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 3 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 4 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 5 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 6 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 7 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 8 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
 9 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
10 N10156 2004 Fixed wi… EMBRAER EMB-… 2 55 NA Turbo… 2013  
# ℹ 284,160 more rows  
# ℹ 17 more variables: month <int>, day <int>, dep\_time <int>,  
# sched\_dep\_time <int>, dep\_delay <dbl>, arr\_time <int>,  
# sched\_arr\_time <int>, arr\_delay <dbl>, carrier <chr>, flight <int>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>

#Finding the plane with largest number of flights from NYC and with more than 50 seats  
number\_of\_flights <- joined %>%   
 filter(origin %in% c("EWR", "LGA", "JFK"),  
 seats >= 50  
 ) %>%   
 count(tailnum) %>%   
 arrange(desc(n))%>%  
 head(1)   
 #We can see that it is plane N328AA with 393 flights  
number\_of\_flights

# A tibble: 1 × 2  
# Groups: tailnum [1]  
 tailnum n  
 <chr> <int>  
1 N328AA 393

#Isolating tail number  
number\_of\_flights$tailnum

[1] "N328AA"

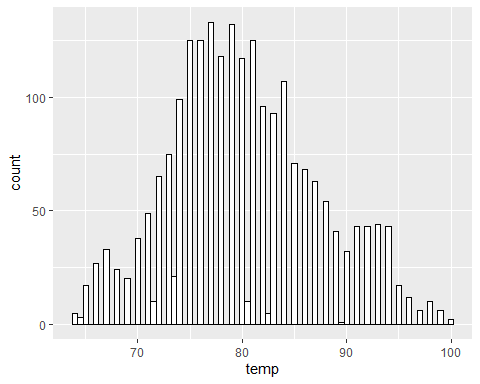
#Finding unique destinations for this plane  
table <- joined %>%   
 filter(tailnum == number\_of\_flights$tailnum) %>%   
 group\_by(dest) %>%   
 select(dest)  
  
unique(table)

# A tibble: 6 × 1  
# Groups: dest [6]  
 dest   
 <chr>  
1 LAX   
2 SFO   
3 SJU   
4 MIA   
5 MCO   
6 BOS

## 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`?

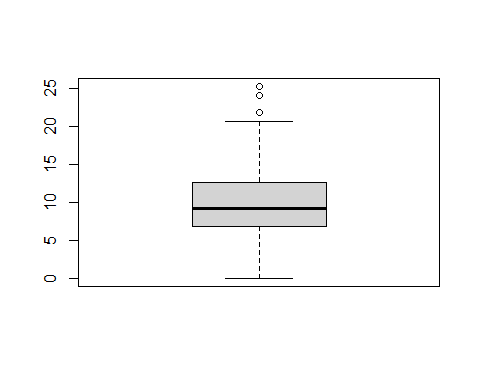
#A: Finding distribution of temperature in July 2013  
july\_weather <- weather %>%   
 filter(month == 7) %>%   
 select(temp) %>%  
 drop\_na()  
  
ggplot(july\_weather, aes(x=temp)) +  
 geom\_histogram(binwidth=.5, colour="black", fill="white")



summary(july\_weather)

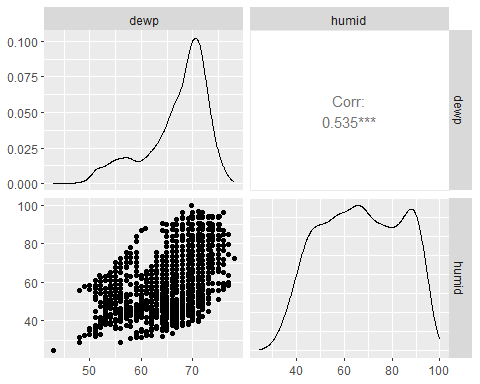
temp   
 Min. : 64.04   
 1st Qu.: 75.02   
 Median : 78.98   
 Mean : 80.07   
 3rd Qu.: 84.20   
 Max. :100.04

#To find the distribution of temperature I have drawn a histogram of the temperature in July, which roughly shows a normal distribution. I also have shown the summary statistics, which shows that the minimum temperature was 64.04, the maximum was 100.04, and the mean was 80.07  
  
  
#B: Finding outliers in wind\_speed (in July)  
  
wind\_speed\_outliers <- weather %>%   
 filter(month == 7) %>%   
 select(wind\_speed)  
  
boxplot(wind\_speed\_outliers)

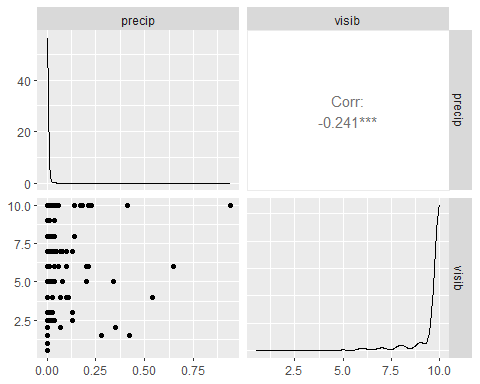


#To find outliers I have drawn a boxplot of the wind speed in July, this shows that there are 3 outliers where the wind speed was significantly high  
  
  
#C: Finding relationship between 'dewp' and 'humid'  
  
weather %>%  
 filter(month == 7) %>%   
 select(dewp,humid) %>%   
 GGally::ggpairs()

Registered S3 method overwritten by 'GGally':  
 method from   
 +.gg ggplot2



#We can see that in July, there is a positive correlation between dewp and humid of 0.535 which is ` statistically significant  
  
  
#D: Finding relationship between 'precip' and 'visib'  
  
weather %>%   
 filter(month == 7) %>%   
 select(precip, visib) %>%   
 GGally::ggpairs()



#We see that in July, there is a negative relationship between dewp and humid of -0.241, suggesting that typically when there is precipitation, the level of visibility will be low, this makes sense!

## 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.)

#A: Number of planes with missing date of manufacture  
  
planes %>%   
 filter(is.na(year)) %>%   
 nrow()

[1] 70

#B: Five most common manufacturers  
most\_common\_planes <- planes %>%   
 group\_by(manufacturer) %>%   
 count(manufacturer) %>%   
 arrange(desc(n)) %>%   
 head(5)  
  
most\_common\_planes

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

#B: Has the distribution of manufacturer changed over time  
common\_manufacturer <- planes %>%  
 group\_by(manufacturer) %>%  
 count(manufacturer) %>%  
 mutate(manufacturer = case\_when(  
 n >= 2 ~ as.character(manufacturer),  
 TRUE ~ "Other"  
 )) %>%   
 group\_by(manufacturer)  
  
total <- sum(common\_manufacturer$n)  
  
common\_manufacturer %>%  
 group\_by(manufacturer) %>%   
 summarize(prop = n/total) %>%   
 arrange(desc(prop))

Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in  
dplyr 1.1.0.  
ℹ Please use `reframe()` instead.  
ℹ When switching from `summarise()` to `reframe()`, remember that `reframe()`  
 always returns an ungrouped data frame and adjust accordingly.

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

# A tibble: 35 × 2  
# Groups: manufacturer [17]  
 manufacturer prop  
 <chr> <dbl>  
 1 BOEING 0.491   
 2 AIRBUS INDUSTRIE 0.120   
 3 BOMBARDIER INC 0.111   
 4 AIRBUS 0.101   
 5 EMBRAER 0.0900   
 6 MCDONNELL DOUGLAS 0.0361   
 7 MCDONNELL DOUGLAS AIRCRAFT CO 0.0310   
 8 MCDONNELL DOUGLAS CORPORATION 0.00421  
 9 CANADAIR 0.00271  
10 CESSNA 0.00271  
# ℹ 25 more rows

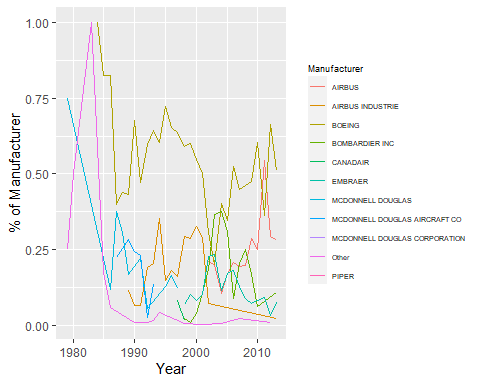
#The above code shows me trying to calculate the proportion of each manufacturer for all planes flown in 2013, I struggled to regroup the 'Other' variables together, but I can see that for flights in 2013, Boeing was by far the most common manufacturer with nearly 50% of planes, followed by Airbus  
  
  
common\_manufacturer\_over\_time <- planes %>%  
 group\_by(year) %>%   
 count(manufacturer, sort=TRUE) %>%   
 mutate(prop = n/sum(n)) %>%   
 mutate(manufacturer = case\_when(  
 n >= 2 ~ as.character(manufacturer),  
 TRUE ~ "Other"  
 )) %>%   
 group\_by(manufacturer, year) %>%   
 summarize(prop = sum(prop)) %>%   
 arrange(year) %>%   
 filter(year >= 1979)

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

#This data set now shows the proportion of the planes from each year that were made by each manufacturer  
common\_manufacturer\_over\_time

# A tibble: 123 × 3  
# Groups: manufacturer [11]  
 manufacturer year prop  
 <chr> <int> <dbl>  
 1 MCDONNELL DOUGLAS 1979 0.75   
 2 Other 1979 0.25   
 3 Other 1980 0.5   
 4 PIPER 1980 0.5   
 5 Other 1983 1   
 6 BOEING 1984 1   
 7 BOEING 1985 0.826  
 8 Other 1985 0.174  
 9 BOEING 1986 0.824  
10 MCDONNELL DOUGLAS 1986 0.118  
# ℹ 113 more rows

ggplot(common\_manufacturer\_over\_time, aes(x = year, y = prop, color = manufacturer)) +  
 geom\_line() +  
 labs(x = "Year", y = "% of Manufacturer") +  
 scale\_color\_discrete(name = "Manufacturer") +  
 theme(legend.text = element\_text(size = 5),  
 legend.title = element\_text(size = 7),  
 legend.key.size = unit(0.5, "cm"))



#The graph shows how the proportion of planes made by each manufacturer has changed over time. In some cases it has changed significantly from year to year, this could reflect that the plane manufacturers produce a number of planes in a year, then take a break whilst designing the next model of plane

## 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?

#A: Finding oldest plane that flew from NYC airports in 2013  
  
oldest\_plane <- left\_join(planes, flights, by = 'tailnum') %>%   
 arrange(year.x) %>%   
 filter(origin %in% c("EWR", "LGA", "JFK")) %>%   
 #isolating tailnum  
 select(tailnum) %>%   
 head(1)  
 #We can see that the oldest plane is plane N381AA  
oldest\_plane

# A tibble: 1 × 1  
 tailnum  
 <chr>   
1 N381AA

#B: How many airplanes that flew from NYC are in the table  
  
number\_of\_planes\_that\_flew\_from\_nyc <- flights %>%   
 filter(origin %in% c("EWR", "LGA", "JFK")) %>%   
 select(tailnum) %>%  
 n\_distinct('tailnum', na.rm = TRUE)  
  
number\_of\_planes\_that\_flew\_from\_nyc

[1] 4043

#We can see that 4043 unique planes flew from NYC in 2013

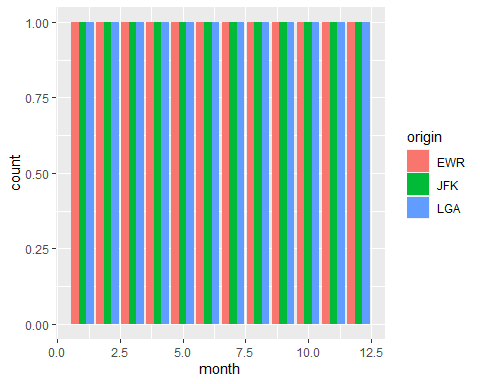
## Problem 7: Use the nycflights13 to answer the following questions:

- 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.

#A: This graph shows the median arrival delay on a month-by-month basis for each airport. I grouped the data by month and origin, and then used the summary function to find the median arrival delay  
median\_arrival\_delay <- flights %>%   
 group\_by(month, origin) %>%   
 summarize(median\_arr\_delay = median(arr\_delay, na.rm = TRUE))

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

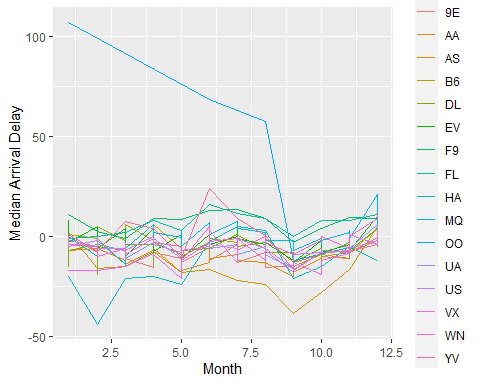
ggplot(median\_arrival\_delay, aes(x = month, fill = origin)) +   
 geom\_bar(stat = "count", position = "dodge")



#B: To plot the median per month for each airline, I grouped by carrier, month, and origin. I then used the summary function to calculate the median arrival delay.   
  
carriers\_month <- flights %>%   
 group\_by(carrier, month, origin) %>%  
 summarise(median\_arr\_delay = median(arr\_delay, na.rm = TRUE))

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

#Then to plot this data, I used a line graph   
  
ggplot(carriers\_month, aes(x = month, y = median\_arr\_delay, color = carrier)) +  
 geom\_line() +  
 labs(x = "Month", y = "Median Arrival Delay")



## 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.

#finding the number of flights into SFO by each airline  
fly\_into\_sfo <- left\_join(planes, flights, by = 'tailnum') %>%   
 filter(dest == 'SFO') %>%   
 group\_by(carrier) %>%   
 count(carrier)  
  
fly\_into\_sfo

# A tibble: 5 × 2  
# Groups: carrier [5]  
 carrier n  
 <chr> <int>  
1 AA 1209  
2 B6 1028  
3 DL 1858  
4 UA 6475  
5 VX 2197

#finding the total number of flights into SFO  
sum <- sum(fly\_into\_sfo$n)  
  
sum

[1] 12767

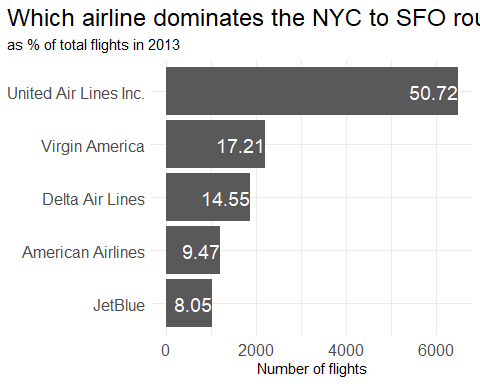
#renaming the values for carrier and calculating percentage of flights, also rounding the percentage to 2 decimal places  
fly\_into\_sfo <- fly\_into\_sfo %>%  
 summarize(name = recode(carrier, "AA" = "American Airlines", "B6" = "JetBlue", "DL" = "Delta Air Lines", "UA" = "United Air Lines Inc.", "VX" = "Virgin America"), count = n, percent = (count/sum)\*100) %>%   
 mutate(across(where(is.numeric), ~ round(., 2)))  
  
fly\_into\_sfo

# A tibble: 5 × 4  
 carrier name count percent  
 <chr> <chr> <dbl> <dbl>  
1 AA American Airlines 1209 9.47  
2 B6 JetBlue 1028 8.05  
3 DL Delta Air Lines 1858 14.6   
4 UA United Air Lines Inc. 6475 50.7   
5 VX Virgin America 2197 17.2

#The resulting table now shows the proportion of flights into SFO by each airline

And here is some bonus ggplot code to plot your dataframe

fly\_into\_sfo %>%   
   
 # sort 'name' of airline by the numbers it times to flew to SFO  
 mutate(name = fct\_reorder(name, count)) %>%   
   
 ggplot() +  
   
 aes(x = count,   
 y = name) +  
   
 # a simple bar/column plot  
 geom\_col() +  
   
 # add labels, so each bar shows the % of total flights   
 geom\_text(aes(label = percent),  
 hjust = 1,   
 colour = "white",   
 size = 5)+  
   
 # add labels to help our audience   
 labs(title="Which airline dominates the NYC to SFO route?",   
 subtitle = "as % of total flights in 2013",  
 x= "Number of flights",  
 y= NULL) +  
   
 theme\_minimal() +   
   
 # change the theme-- i just googled those , but you can use the ggThemeAssist add-in  
 # https://cran.r-project.org/web/packages/ggThemeAssist/index.html  
   
 theme(#  
 # so title is left-aligned  
 plot.title.position = "plot",  
   
 # text in axes appears larger   
 axis.text = element\_text(size=12),  
   
 # title text is bigger  
 plot.title = element\_text(size=18)  
 ) +  
  
 # add one final layer of NULL, so if you comment out any lines  
 # you never end up with a hanging `+` that awaits another ggplot layer  
 NULL



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

cancellations <- flights %>%   
   
 # just filter for destination == 'SFO'  
 filter(dest == 'SFO') %>%   
   
 # a cancelled flight is one with no `dep\_time`   
 filter(is.na(dep\_time))  
  
cancellations

# A tibble: 101 × 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 18 NA 1100 NA NA 1425  
 2 2013 10 2 NA 730 NA NA 1045  
 3 2013 10 17 NA 1830 NA NA 2157  
 4 2013 10 23 NA 1025 NA NA 1340  
 5 2013 10 24 NA 1025 NA NA 1340  
 6 2013 10 25 NA 1025 NA NA 1340  
 7 2013 10 26 NA 825 NA NA 1148  
 8 2013 10 27 NA 600 NA NA 923  
 9 2013 11 27 NA 1025 NA NA 1400  
10 2013 12 6 NA 1000 NA NA 1321  
# ℹ 91 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>

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 this plot we would first group the data by month and carrier (which we need to rename), and then filter to ensure that origin = EWR or JFK.

We then create a box plot using ggplot with month as the label variable, doing a facet\_wrap by both carrier and origin to ensure that there is a new plot for each specific combination of those variables.

We need to add the appropriate title, and adjust the theme to how we want 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 |

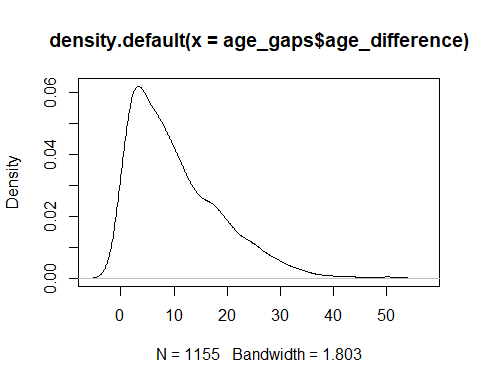
age\_gaps <- readr::read\_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2023/2023-02-14/age\_gaps.csv')

Rows: 1155 Columns: 13  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (6): movie\_name, director, actor\_1\_name, actor\_2\_name, character\_1\_gend...  
dbl (5): release\_year, age\_difference, couple\_number, actor\_1\_age, actor\_2\_age  
date (2): actor\_1\_birthdate, actor\_2\_birthdate  
  
ℹ Use `spec()` to retrieve the full column specification for this data.  
ℹ Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

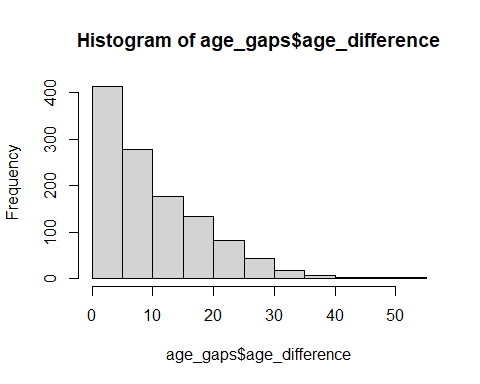
age\_gaps

# A tibble: 1,155 × 13  
 movie\_name release\_year director age\_difference couple\_number actor\_1\_name  
 <chr> <dbl> <chr> <dbl> <dbl> <chr>   
 1 Harold and M… 1971 Hal Ash… 52 1 Ruth Gordon   
 2 Venus 2006 Roger M… 50 1 Peter O'Too…  
 3 The Quiet Am… 2002 Phillip… 49 1 Michael Cai…  
 4 The Big Lebo… 1998 Joel Co… 45 1 David Huddl…  
 5 Beginners 2010 Mike Mi… 43 1 Christopher…  
 6 Poison Ivy 1992 Katt Sh… 42 1 Tom Skerritt  
 7 Whatever Wor… 2009 Woody A… 40 1 Larry David   
 8 Entrapment 1999 Jon Ami… 39 1 Sean Connery  
 9 Husbands and… 1992 Woody A… 38 1 Woody Allen   
10 Magnolia 1999 Paul Th… 38 1 Jason Robar…  
# ℹ 1,145 more rows  
# ℹ 7 more variables: actor\_2\_name <chr>, character\_1\_gender <chr>,  
# character\_2\_gender <chr>, actor\_1\_birthdate <date>,  
# actor\_2\_birthdate <date>, actor\_1\_age <dbl>, actor\_2\_age <dbl>

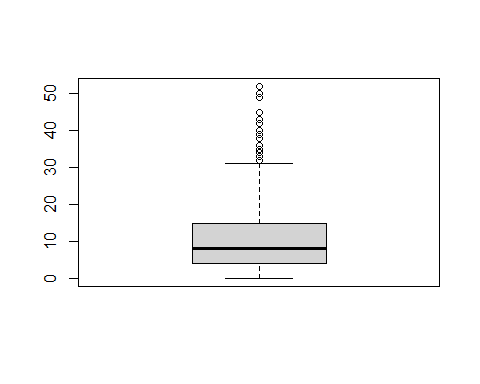
#A: Examining the distribution of age\_difference and finding the mean ('typical') age difference  
  
plot(density(age\_gaps$age\_difference))



#This density plot shows that the most common age gap is around 4 years  
  
hist(age\_gaps$age\_difference)



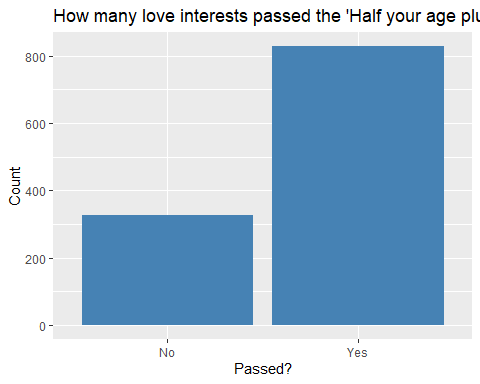
#This histogram shows that the vast majority of age gaps are between 0 and 20 years  
  
boxplot(age\_gaps$age\_difference)



#This boxplot shows that there are a number of outliers with very large age gaps of more than 30 years  
   
age\_gaps %>%   
 summarize(  
 typical\_age\_gap = mean(age\_difference),  
 maximum = max(age\_difference),  
 minimum = min(age\_difference),  
 median = median(age\_difference),  
 sd = sd(age\_difference)  
 ) %>%   
 mutate(  
 across(where(is.numeric), ~ round(., 2))  
 )

# A tibble: 1 × 5  
 typical\_age\_gap maximum minimum median sd  
 <dbl> <dbl> <dbl> <dbl> <dbl>  
1 10.4 52 0 8 8.51

#This table shows some summary statistics, including the 'typical'/mean age gap which is 10.42, the maximum age gap which is 52, and the minimum age gap which is 0  
  
  
#B: The half plus 7 rule  
half\_plus\_seven\_rule <- age\_gaps %>%   
 select(movie\_name, actor\_1\_age, actor\_2\_age) %>%   
 #In order for the 'half plus 7' rule to apply, the 'actor\_2\_age' variable must be more than 'half plus 7' of the actor 1 age  
 mutate(half\_plus\_seven = ((actor\_1\_age/2) + 7),  
 test = actor\_2\_age - half\_plus\_seven  
 #The 'test' is positive if the test is passed, and negative if not  
 ) %>%   
 arrange(test) %>%   
 mutate(passed = case\_when(  
 test >= 0 ~ "Yes",  
 TRUE ~ "No"  
 )) %>%   
 #Using the mutate and case\_when function I have created 2 outcomes, either 'yes' or 'no' to whether the test is passed  
 group\_by(passed) %>%   
 #Finally I count the number of times the test is either passed or not  
 count(passed)  
  
  
ggplot(data = half\_plus\_seven\_rule, aes(x = passed, y = n)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 labs(x = "Passed?", y = "Count") +  
 ggtitle("How many love interests passed the 'Half your age plus 7' rule?")



#I used ggplot to show the number of occasions in which the rule is passed or not  
  
  
  
  
#C: Which movie has the greatest number of love interests?  
number\_of\_love\_interests <- age\_gaps %>%   
 group\_by(movie\_name) %>%   
 count() %>%   
 arrange(desc(n)) %>%   
 head(1)  
 #To find which movie has the greatest number of love interests, I first grouped by movie name, then counted the number of times each movie appeared, then sorted by n, then took the film with the most appearances - which was (unsurprisingly), Love Actually with 7 love interests  
  
  
#D: Which actor/actresses has the greatest number of love interests?  
  
 #The challenge here is that some actors may appear as both actor\_1 and actor\_2 across different films, so we need to collate the data  
pivot\_longer(  
 data = age\_gaps, cols = c("actor\_1\_name", "actor\_2\_name"), names\_to = "Actor", values\_to = "Name") %>%   
 #I pivot longer to get all of the names of actors in one column, allowing me to find the most numerous names regardless of whether they originally appeared as 'Actor 1' or 'Actor 2'  
 select(Name) %>%  
 group\_by(Name) %>%   
 count() %>%   
 arrange(desc(n)) %>%   
 head(5)

# A tibble: 5 × 2  
# Groups: Name [5]  
 Name n  
 <chr> <int>  
1 Keanu Reeves 27  
2 Adam Sandler 20  
3 Leonardo DiCaprio 17  
4 Roger Moore 17  
5 Sean Connery 17

#The 5 actors/actresses that appear the most can be seen in this table, all are male and the lowest number of appearances is 17  
  
  
  
#E: Is the mean/median age difference staying constant over the years?  
age\_gaps\_over\_time <- age\_gaps %>%   
 mutate(age\_gap = actor\_1\_age - actor\_2\_age) %>%   
 select(age\_gap, release\_year) %>%   
 #cor() %>%   
 GGally::ggpairs()  
 #This correlation test shows that there is a negative correlation between age gap and release year, suggesting that the general age gap in films has come down over time  
  
age\_gaps\_over\_time\_mean <- age\_gaps %>%   
 mutate(age\_gap = actor\_1\_age - actor\_2\_age) %>%   
 select(age\_gap, release\_year) %>%  
 group\_by(release\_year) %>%   
 arrange(release\_year) %>%   
 summarize(mean\_age\_gap = mean(age\_gap),   
 median\_age\_gap = median(age\_gap)) %>%   
 cor()  
  
#This correlation table shows that there is a negative correlation between release year and both mean age gap (-0.486) and median age gap (-0.484)  
   
  
  
#F: How often does Hollywood depict same gender love interests?  
same\_gender\_love\_interests <- age\_gaps %>%   
 filter(character\_1\_gender == character\_2\_gender) %>%   
 count()  
 #I found the number of times that a film has depicted a same gender couple (where character 1 gender = character 2 gender)  
  
total\_love\_interests <- age\_gaps %>%   
 count()  
 #I found the total number of love interests   
  
proportion\_of\_same\_gender\_love\_interests <- (same\_gender\_love\_interests$n/total\_love\_interests$n)  
 #I calculate the proportion of love interests that are same gender  
  
proportion\_of\_same\_gender\_love\_interests

[1] 0.01991342

#We can see that only 1.99% of love interests depicted by Hollywood show same gender couples

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: TYPE NAMES HERE
* Approximately how much time did you spend on this problem set: ANSWER HERE
* 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

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.