Homerwork 1

Nianyu Li

2023-05-18

# Data Manipulation

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

- Had an arrival delay of two or more hours (\> 120 minutes)  
- Flew to Houston (IAH or HOU)  
- Were operated by United (`UA`), American (`AA`), or Delta (`DL`)  
- Departed in summer (July, August, and September)  
- Arrived more than two hours late, but didn't leave late  
- Were delayed by at least an hour, but made up over 30 minutes in flight

# Had an arrival delay of two or more hours (> 120 minutes)  
flights %>%   
 filter(arr\_delay>=120)

# A tibble: 10,200 × 19  
 year month day dep\_time sched\_de…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
 1 2013 1 1 811 630 101 1047 830 137 MQ   
 2 2013 1 1 848 1835 853 1001 1950 851 MQ   
 3 2013 1 1 957 733 144 1056 853 123 UA   
 4 2013 1 1 1114 900 134 1447 1222 145 UA   
 5 2013 1 1 1505 1310 115 1638 1431 127 EV   
 6 2013 1 1 1525 1340 105 1831 1626 125 B6   
 7 2013 1 1 1549 1445 64 1912 1656 136 EV   
 8 2013 1 1 1558 1359 119 1718 1515 123 EV   
 9 2013 1 1 1732 1630 62 2028 1825 123 EV   
10 2013 1 1 1803 1620 103 2008 1750 138 MQ   
# … with 10,190 more rows, 9 more variables: flight <int>, tailnum <chr>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>, and abbreviated variable names  
# ¹​sched\_dep\_time, ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

# Flew to Houston (IAH or HOU)  
flights %>%   
 filter(dest == "IAH"| dest == "HOU")

# A tibble: 9,313 × 19  
 year month day dep\_time sched\_de…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
 1 2013 1 1 517 515 2 830 819 11 UA   
 2 2013 1 1 533 529 4 850 830 20 UA   
 3 2013 1 1 623 627 -4 933 932 1 UA   
 4 2013 1 1 728 732 -4 1041 1038 3 UA   
 5 2013 1 1 739 739 0 1104 1038 26 UA   
 6 2013 1 1 908 908 0 1228 1219 9 UA   
 7 2013 1 1 1028 1026 2 1350 1339 11 UA   
 8 2013 1 1 1044 1045 -1 1352 1351 1 UA   
 9 2013 1 1 1114 900 134 1447 1222 145 UA   
10 2013 1 1 1205 1200 5 1503 1505 -2 UA   
# … with 9,303 more rows, 9 more variables: flight <int>, tailnum <chr>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>, and abbreviated variable names  
# ¹​sched\_dep\_time, ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

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

# A tibble: 139,504 × 19  
 year month day dep\_time sched\_de…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
 1 2013 1 1 517 515 2 830 819 11 UA   
 2 2013 1 1 533 529 4 850 830 20 UA   
 3 2013 1 1 542 540 2 923 850 33 AA   
 4 2013 1 1 554 600 -6 812 837 -25 DL   
 5 2013 1 1 554 558 -4 740 728 12 UA   
 6 2013 1 1 558 600 -2 753 745 8 AA   
 7 2013 1 1 558 600 -2 924 917 7 UA   
 8 2013 1 1 558 600 -2 923 937 -14 UA   
 9 2013 1 1 559 600 -1 941 910 31 AA   
10 2013 1 1 559 600 -1 854 902 -8 UA   
# … with 139,494 more rows, 9 more variables: flight <int>, tailnum <chr>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>, and abbreviated variable names  
# ¹​sched\_dep\_time, ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

#Can also use  
flights %>%   
 filter(carrier %in% c("UA", "AA", "DL"))

# A tibble: 139,504 × 19  
 year month day dep\_time sched\_de…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
 1 2013 1 1 517 515 2 830 819 11 UA   
 2 2013 1 1 533 529 4 850 830 20 UA   
 3 2013 1 1 542 540 2 923 850 33 AA   
 4 2013 1 1 554 600 -6 812 837 -25 DL   
 5 2013 1 1 554 558 -4 740 728 12 UA   
 6 2013 1 1 558 600 -2 753 745 8 AA   
 7 2013 1 1 558 600 -2 924 917 7 UA   
 8 2013 1 1 558 600 -2 923 937 -14 UA   
 9 2013 1 1 559 600 -1 941 910 31 AA   
10 2013 1 1 559 600 -1 854 902 -8 UA   
# … with 139,494 more rows, 9 more variables: flight <int>, tailnum <chr>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>, and abbreviated variable names  
# ¹​sched\_dep\_time, ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

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

# A tibble: 86,326 × 19  
 year month day dep\_time sched\_de…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
 1 2013 7 1 1 2029 212 236 2359 157 B6   
 2 2013 7 1 2 2359 3 344 344 0 B6   
 3 2013 7 1 29 2245 104 151 1 110 B6   
 4 2013 7 1 43 2130 193 322 14 188 B6   
 5 2013 7 1 44 2150 174 300 100 120 AA   
 6 2013 7 1 46 2051 235 304 2358 186 B6   
 7 2013 7 1 48 2001 287 308 2305 243 VX   
 8 2013 7 1 58 2155 183 335 43 172 B6   
 9 2013 7 1 100 2146 194 327 30 177 B6   
10 2013 7 1 100 2245 135 337 135 122 B6   
# … with 86,316 more rows, 9 more variables: flight <int>, tailnum <chr>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>, and abbreviated variable names  
# ¹​sched\_dep\_time, ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

# Arrived more than two hours late, but didn't leave late  
flights %>%   
 filter(arr\_delay>120 & dep\_delay==0)

# A tibble: 3 × 19  
 year month day dep\_time sched\_dep…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
1 2013 10 7 1350 1350 0 1736 1526 130 EV   
2 2013 5 23 1810 1810 0 2208 2000 128 MQ   
3 2013 7 1 905 905 0 1443 1223 140 DL   
# … with 9 more variables: flight <int>, tailnum <chr>, origin <chr>,  
# dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,  
# time\_hour <dttm>, and abbreviated variable names ¹​sched\_dep\_time,  
# ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

# 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\_de…¹ dep\_d…² arr\_t…³ sched…⁴ arr\_d…⁵ carrier  
 <int> <int> <int> <int> <int> <dbl> <int> <int> <dbl> <chr>   
 1 2013 1 3 1850 1745 65 2148 2120 28 AA   
 2 2013 1 3 1950 1845 65 2228 2227 1 B6   
 3 2013 1 3 2015 1915 60 2135 2111 24 9E   
 4 2013 1 6 1019 900 79 1558 1530 28 HA   
 5 2013 1 7 1543 1430 73 1758 1735 23 AA   
 6 2013 1 11 1020 920 60 1311 1245 26 AA   
 7 2013 1 12 1706 1600 66 1949 1927 22 DL   
 8 2013 1 12 1953 1845 68 2154 2137 17 9E   
 9 2013 1 19 1456 1355 61 1636 1615 21 EV   
10 2013 1 21 1531 1430 61 1843 1815 28 DL   
# … with 196 more rows, 9 more variables: flight <int>, tailnum <chr>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>, and abbreviated variable names  
# ¹​sched\_dep\_time, ²​dep\_delay, ³​arr\_time, ⁴​sched\_arr\_time, ⁵​arr\_delay

## Problem 2: What months had the highest and lowest proportion of cancelled flights? Interpret any seasonal patterns. To determine if a flight was cancelled use the following code

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

# What months had the highest and lowest % of cancelled flights?  
  
#Calculate total # of flights each month  
total\_flights <- flights %>%  
 group\_by(month) %>%   
 count(name = "total\_flights")  
  
#Calculate total # of flights each month that is cancelled  
cancelled\_flights <- flights %>%  
 filter(is.na(dep\_time)) %>%   
 group\_by(month) %>%  
 count(name = "cancelled\_flights")  
  
#Join the two new tables and calculate the cancel rate, arrange the column in descending order.  
cancel\_rate <-  
 inner\_join(x = total\_flights,  
 y = cancelled\_flights,  
 by = "month") %>%   
 mutate(cancel\_rate = cancelled\_flights/total\_flights) %>% arrange(desc(cancel\_rate))  
  
print(cancel\_rate)

# A tibble: 12 × 4  
# Groups: month [12]  
 month total\_flights cancelled\_flights cancel\_rate  
 <int> <int> <int> <dbl>  
 1 2 24951 1261 0.0505   
 2 12 28135 1025 0.0364   
 3 6 28243 1009 0.0357   
 4 7 29425 940 0.0319   
 5 3 28834 861 0.0299   
 6 4 28330 668 0.0236   
 7 5 28796 563 0.0196   
 8 1 27004 521 0.0193   
 9 8 29327 486 0.0166   
10 9 27574 452 0.0164   
11 11 27268 233 0.00854  
12 10 28889 236 0.00817

#From the output we can tell that Feburary has the highest proportion of cancelled flights (5.05%), and October has the lowest (0.82%)

## 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.

#Count the travel times for each flights  
travel\_times <- flights %>%  
 filter(!is.na(tailnum)) %>%   
 group\_by(tailnum) %>%  
 count(name = "travel\_times")  
  
#Left join it with the table planes  
planes\_travel\_times <- left\_join(x = travel\_times, y = planes, by = "tailnum") %>%  
 filter(seats > 50) %>%   
 arrange(desc(travel\_times))  
  
print(planes\_travel\_times)

# A tibble: 3,200 × 10  
# Groups: tailnum [3,200]  
 tailnum travel\_times year type manuf…¹ model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 N328AA 393 1986 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 2 N338AA 388 1987 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 3 N327AA 387 1986 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 4 N335AA 385 1987 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 5 N323AA 357 1986 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 6 N319AA 354 1985 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 7 N336AA 353 1987 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 8 N329AA 344 1987 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
 9 N789JB 332 2011 Fixed wi… AIRBUS A320… 2 200 NA Turbo…  
10 N324AA 328 1986 Fixed wi… BOEING 767-… 2 255 NA Turbo…  
# … with 3,190 more rows, and abbreviated variable name ¹​manufacturer

#We can tell that the most travelled plane's tailnumber is N328AA  
  
#Create a table for the most travelled plane  
most\_travelled\_flight <- flights %>%   
 filter(tailnum == "N328AA") %>%   
 group\_by(dest) %>%   
 count() %>%   
 arrange(desc(n))  
  
print(most\_travelled\_flight)

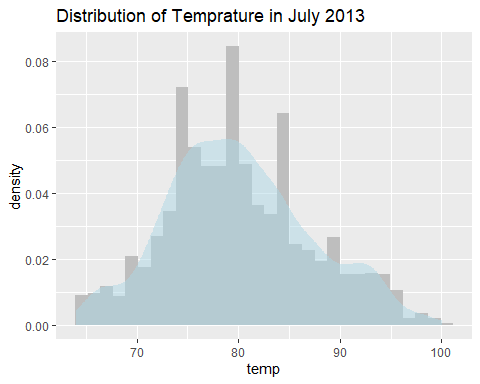
# A tibble: 6 × 2  
# Groups: dest [6]  
 dest n  
 <chr> <int>  
1 LAX 313  
2 SFO 52  
3 MIA 25  
4 BOS 1  
5 MCO 1  
6 SJU 1

## 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`?

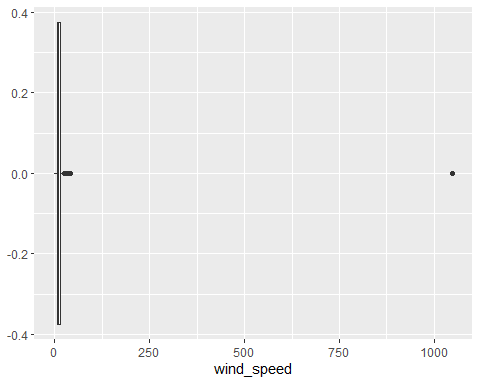
#What is the distribution of temperature (`temp`) in July 2013?  
  
#Filter data for July 2013  
temp\_July <- weather %>%   
 filter(month == 7) %>%   
 select(temp)  
  
#Distribution  
ggplot(temp\_July, aes(x=temp)) +  
 geom\_histogram(aes(y = ..density..), fill = "grey")+  
 geom\_density(color = NA, fill = "lightblue", alpha = 0.5)+  
 labs(title = "Distribution of Temprature in July 2013")

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



#Temperature in July 2013 is roughly normally distributed.  
  
#Find the outliers for "windspeed" variable  
ggplot(weather, aes(x=wind\_speed)) +  
 geom\_boxplot()

Warning: Removed 4 rows containing non-finite values (stat\_boxplot).



skimr::skim(weather$wind\_speed)

Data summary

|  |  |
| --- | --- |
| Name | weather$wind\_speed |
| Number of rows | 26115 |
| Number of columns | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| numeric | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: numeric**

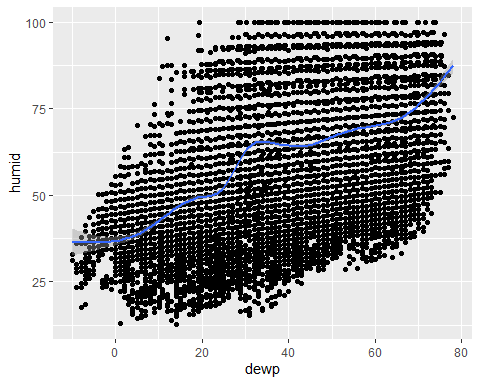
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| data | 4 | 1 | 10.52 | 8.54 | 0 | 6.9 | 10.36 | 13.81 | 1048.36 | ▇▁▁▁▁ |

#The outlier is 1048.361, which is also the 100 percentile.  
  
#What is the relationship between `dewp` and `humid`?  
ggplot(weather, aes(x=dewp, y=humid))+  
 geom\_point()+  
 geom\_smooth()

`geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'

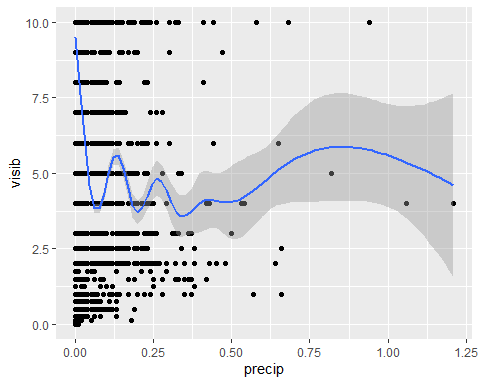
Warning: Removed 1 rows containing non-finite values (stat\_smooth).

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



#Positively correlated: when dewp level raises, it becomes more humid.  
  
#What is the relationship between `precip` and `visib`?  
ggplot(weather, aes(x=precip, y=visib))+  
 geom\_point()+  
 geom\_smooth()

`geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



#Pattern not obvious: there is no relationship between `precip` and `visib`.

## 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.)

#How many planes have a missing date of manufacture?  
planes %>%   
 filter(is.na(year)) %>%   
 count(name = "plane\_missing\_manu\_date")

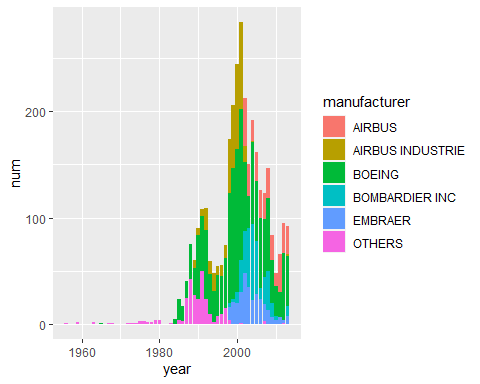
# A tibble: 1 × 1  
 plane\_missing\_manu\_date  
 <int>  
1 70

#What are the five most common manufacturers?  
common\_manufacturers <- planes %>%  
 group\_by(manufacturer) %>%   
 count() %>%   
 arrange(desc(n)) %>%   
 head(., 5)  
print(common\_manufacturers)

# 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

#Has the distribution of manufacturer changed over time as reflected by the airplanes flying from NYC in 2013?  
  
#Recode the manufacturer name and collapse rare vendors into a category called Other  
planes\_1 <- planes %>%   
 mutate(manufacturer = ifelse(manufacturer %in%   
 c("BOEING","AIRBUS INDUSTRIE",  
 "BOMBARDIER INC", "AIRBUS", "EMBRAER"),  
 manufacturer, "OTHERS"))  
  
#Create a table of the distribution in each year  
distribution\_long <- planes\_1 %>%   
 group\_by(year) %>%  
 count(manufacturer, name = "num")   
  
#Plot the distribution to see the trend  
ggplot(distribution\_long, aes(x=year, y=num, fill = manufacturer))+  
 geom\_bar(stat="identity")

Warning: Removed 6 rows containing missing values (position\_stack).



#We can tell that there are indeed changes in distribution of manufacturers.   
#For example, BOEING and AIRBUS INDUSTRIE raised around 2000.  
#Or "BOMBARDIER INC" and "EMBRAER" starting to take up market shares after 2000.

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

- What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
- How many airplanes that flew from New York City are included in the planes table?

#What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
#I am not sure how the tailnum indicates the age of the plane. I assume the smaller the number the older the plane?  
planes\_dist <-flights %>%   
 group\_by(tailnum) %>%  
 count() %>%   
 print()

# A tibble: 4,044 × 2  
# Groups: tailnum [4,044]  
 tailnum n  
 <chr> <int>  
 1 D942DN 4  
 2 N0EGMQ 371  
 3 N10156 153  
 4 N102UW 48  
 5 N103US 46  
 6 N104UW 47  
 7 N10575 289  
 8 N105UW 45  
 9 N107US 41  
10 N108UW 60  
# … with 4,034 more rows

#Then the oldest one is N0EGMQ.  
  
#How many airplanes that flew from New York City are included in the planes table?  
inner\_join(x=planes\_dist, y=planes, by = "tailnum")

# A tibble: 3,322 × 10  
# Groups: tailnum [3,322]  
 tailnum n year type manuf…¹ model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 N10156 153 2004 Fixed wing mult… EMBRAER EMB-… 2 55 NA Turbo…  
 2 N102UW 48 1998 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
 3 N103US 46 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
 4 N104UW 47 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
 5 N10575 289 2002 Fixed wing mult… EMBRAER EMB-… 2 55 NA Turbo…  
 6 N105UW 45 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
 7 N107US 41 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
 8 N108UW 60 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
 9 N109UW 48 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
10 N110UW 40 1999 Fixed wing mult… AIRBUS… A320… 2 182 NA Turbo…  
# … with 3,312 more rows, and abbreviated variable name ¹​manufacturer

#There are 3322 rows in the joint table.  
#Therefore, of all the 4044 planes that flew from New York City, 3322 of them are included in the planes table.

## 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.

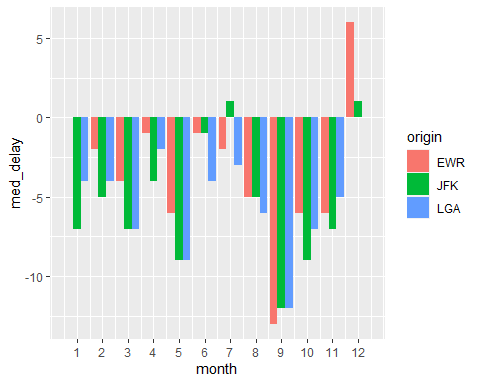
#What is the median arrival delay on a month-by-month basis in each airport?  
med\_delay <- flights %>%   
 group\_by(month,origin) %>%   
 summarize(med\_delay = median(arr\_delay, na.rm = TRUE))

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

print(med\_delay)

# A tibble: 36 × 3  
# Groups: month [12]  
 month origin med\_delay  
 <int> <chr> <dbl>  
 1 1 EWR 0  
 2 1 JFK -7  
 3 1 LGA -4  
 4 2 EWR -2  
 5 2 JFK -5  
 6 2 LGA -4  
 7 3 EWR -4  
 8 3 JFK -7  
 9 3 LGA -7  
10 4 EWR -1  
# … with 26 more rows

#For each airline, plot the median arrival delay for each month and origin airport.  
ggplot(med\_delay, aes(x=month, y=med\_delay, group = origin, fill = origin))+  
 geom\_col(position = "dodge")+  
 scale\_x\_continuous(breaks = 1:12)



## 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.

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

#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.



## 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.

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: 6hrs
* What, if anything, gave you the most trouble: Not enough time to go through all the questions.

**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.