Homerwork 1

Isabel Gonzalez

5/14/23

# Data Manipulation

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

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

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

# A tibble: 10,200 × 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 1 811 630 101 1047 830  
 2 2013 1 1 848 1835 853 1001 1950  
 3 2013 1 1 957 733 144 1056 853  
 4 2013 1 1 1114 900 134 1447 1222  
 5 2013 1 1 1505 1310 115 1638 1431  
 6 2013 1 1 1525 1340 105 1831 1626  
 7 2013 1 1 1549 1445 64 1912 1656  
 8 2013 1 1 1558 1359 119 1718 1515  
 9 2013 1 1 1732 1630 62 2028 1825  
10 2013 1 1 1803 1620 103 2008 1750  
# ℹ 10,190 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

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

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

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

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

# A tibble: 86,326 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 7 1 1 2029 212 236 2359  
 2 2013 7 1 2 2359 3 344 344  
 3 2013 7 1 29 2245 104 151 1  
 4 2013 7 1 43 2130 193 322 14  
 5 2013 7 1 44 2150 174 300 100  
 6 2013 7 1 46 2051 235 304 2358  
 7 2013 7 1 48 2001 287 308 2305  
 8 2013 7 1 58 2155 183 335 43  
 9 2013 7 1 100 2146 194 327 30  
10 2013 7 1 100 2245 135 337 135  
# ℹ 86,316 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

# A tibble: 3 × 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 10 7 1350 1350 0 1736 1526  
2 2013 5 23 1810 1810 0 2208 2000  
3 2013 7 1 905 905 0 1443 1223  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

# Were delayed by at least an hour, but made up over 30 minutes in flight  
  
flights %>%  
 filter(dep\_delay-arr\_delay> 30 , dep\_delay >= 60)

# A tibble: 1,844 × 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 2205 1720 285 46 2040  
 2 2013 1 1 2326 2130 116 131 18  
 3 2013 1 3 1503 1221 162 1803 1555  
 4 2013 1 3 1839 1700 99 2056 1950  
 5 2013 1 3 1850 1745 65 2148 2120  
 6 2013 1 3 1941 1759 102 2246 2139  
 7 2013 1 3 1950 1845 65 2228 2227  
 8 2013 1 3 2015 1915 60 2135 2111  
 9 2013 1 3 2257 2000 177 45 2224  
10 2013 1 4 1917 1700 137 2135 1950  
# ℹ 1,834 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

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

# What months had the highest and lowest % of cancelled flights?  
  
#create a new table (flights\_cancelled) with a line for each month and three columns (# of cancelled flights, # of total flights and % of cancelled flights)  
  
flights\_c <- flights %>%  
 group\_by(month) %>%  
 summarize(cancelled = sum(is.na(dep\_time)),  
 total = n()) %>%   
 mutate(percent\_c = cancelled / total)   
   
#months with highest and lowest % of cancelled flights  
   
highest\_cancelled <- filter(flights\_c, percent\_c == max(percent\_c))$month  
lowest\_cancelled <- filter(flights\_c, percent\_c== min(percent\_c))$month  
  
#print  
flights\_c

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

highest\_cancelled

[1] 2

lowest\_cancelled

[1] 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.

# filter planes that departed from New York City airports, group them by tailnumber and count the number of flights for each plane that flew from that airport  
  
num\_flights\_nyc <- flights %>%  
 filter(origin %in% c("JFK", "EWR", "LGA")) %>%  
 group\_by(tailnum) %>%  
 summarise(num\_flights = n())   
   
# join tables  
   
joint\_flights\_planes <- num\_flights\_nyc %>%  
 left\_join(planes, by = "tailnum") %>%  
arrange(desc(num\_flights))  
  
# plane that travel the most   
  
planethattraveledthemost <- filter(joint\_flights\_planes, num\_flights== max(num\_flights))$tailnum  
  
# print   
planethattraveledthemost

[1] NA

joint\_flights\_planes

# A tibble: 4,044 × 10  
 tailnum num\_flights year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 <NA> 2512 NA <NA> <NA> <NA> NA NA NA <NA>   
 2 N725MQ 575 NA <NA> <NA> <NA> NA NA NA <NA>   
 3 N722MQ 513 NA <NA> <NA> <NA> NA NA NA <NA>   
 4 N723MQ 507 NA <NA> <NA> <NA> NA NA NA <NA>   
 5 N711MQ 486 1976 Fixe… GULFSTREAM … G115… 2 22 NA Turbo…  
 6 N713MQ 483 NA <NA> <NA> <NA> NA NA NA <NA>   
 7 N258JB 427 2006 Fixe… EMBRAER ERJ … 2 20 NA Turbo…  
 8 N298JB 407 2009 Fixe… EMBRAER ERJ … 2 20 NA Turbo…  
 9 N353JB 404 2012 Fixe… EMBRAER ERJ … 2 20 NA Turbo…  
10 N351JB 402 2012 Fixe… EMBRAER ERJ … 2 20 NA Turbo…  
# ℹ 4,034 more rows

# plane with more than 50 seats  
  
fiftyseats <- joint\_flights\_planes %>%  
 filter(seats > 50)  
  
plane\_fifty <- filter(fiftyseats, num\_flights== max(num\_flights))$tailnum  
  
# print   
  
fiftyseats

# A tibble: 3,200 × 10  
 tailnum num\_flights year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 N328AA 393 1986 Fixe… BOEING 767-… 2 255 NA Turbo…  
 2 N338AA 388 1987 Fixe… BOEING 767-… 2 255 NA Turbo…  
 3 N327AA 387 1986 Fixe… BOEING 767-… 2 255 NA Turbo…  
 4 N335AA 385 1987 Fixe… BOEING 767-… 2 255 NA Turbo…  
 5 N323AA 357 1986 Fixe… BOEING 767-… 2 255 NA Turbo…  
 6 N319AA 354 1985 Fixe… BOEING 767-… 2 255 NA Turbo…  
 7 N336AA 353 1987 Fixe… BOEING 767-… 2 255 NA Turbo…  
 8 N329AA 344 1987 Fixe… BOEING 767-… 2 255 NA Turbo…  
 9 N789JB 332 2011 Fixe… AIRBUS A320… 2 200 NA Turbo…  
10 N324AA 328 1986 Fixe… BOEING 767-… 2 255 NA Turbo…  
# ℹ 3,190 more rows

plane\_fifty

[1] "N328AA"

#create table of destinations for plane that traveled the most with more than 50 seats ("plane\_fifty")  
  
plane\_destinations <- flights %>%  
 filter(tailnum == plane\_fifty) %>%  
 distinct(dest)  
  
  
plane\_destinations

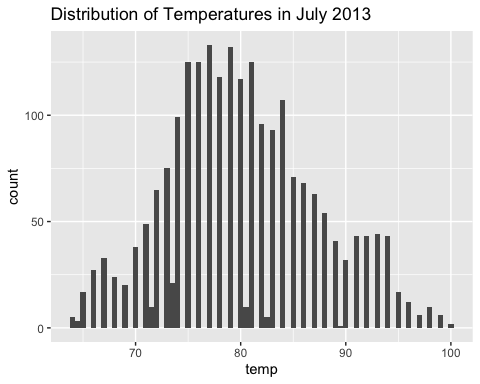
# A tibble: 6 × 1  
 dest   
 <chr>  
1 LAX   
2 SFO   
3 SJU   
4 MIA   
5 MCO   
6 BOS

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

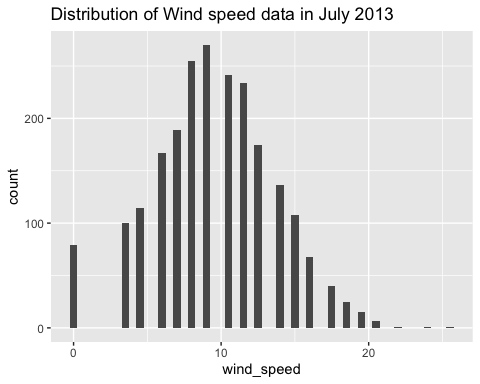
According to the histogram of the historical data of temperatures in July 2013, the data is normally distributed with a slight right skew After plotting the data of “dewp” vs. “humid” we can see the blue line that represents a positive relation between the two variables After plotting the data of “precip” vs. “visib” we can see that there is no relation between the two variables

- 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`?

# filter and plot only the temperatures from July 2013, I used the histogram to identify the distribution of the data  
weather\_july <- weather %>%  
 filter(month == 7)   
  
temp\_plot <-  
ggplot(weather\_july,aes(x=temp))+geom\_histogram(binwidth = 0.5)+ggtitle("Distribution of Temperatures in July 2013")  
  
temp\_plot



# filter and plot only the wind\_speed from July 2013  
   
wind\_speed\_plot <-  
ggplot(weather\_july,aes(x=wind\_speed))+geom\_histogram(binwidth = 0.5, na.rm = TRUE)+ggtitle("Distribution of Wind speed data in July 2013")  
wind\_speed\_plot

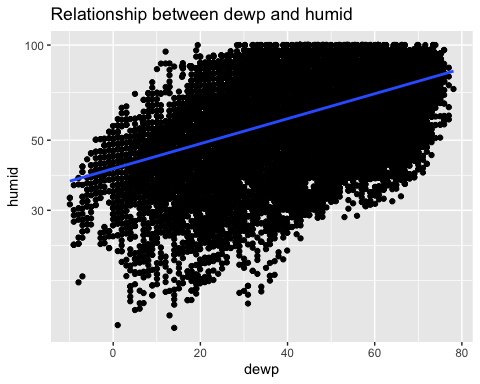


# Extract the wind\_speed column  
wind\_speed <- weather\_july$wind\_speed  
  
# Calculate the mean and standard deviation  
mean\_wind\_speed <- mean(wind\_speed, na.rm = TRUE)  
sd\_wind\_speed <- sd(wind\_speed, na.rm = TRUE)  
  
# Calculate the lower and upper thresholds  
lower\_threshold <- mean\_wind\_speed - 3 \* sd\_wind\_speed  
upper\_threshold <- mean\_wind\_speed + 3 \* sd\_wind\_speed  
  
# Identify outliers  
outliers <- wind\_speed[wind\_speed < lower\_threshold | wind\_speed > upper\_threshold]  
#delete NAs from the answer  
outliers <- outliers[!is.na(outliers)]  
  
# Create a data frame with outliers as separate rows  
outliers\_table <- data.frame(Outliers = outliers)  
  
# Print the table  
print(outliers\_table)

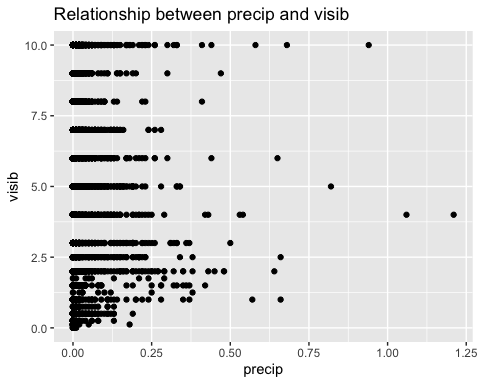
Outliers  
1 21.86482  
2 24.16638  
3 25.31716

#Plot to observe the relationship between `dewp` and `humid`  
dewp\_humid\_plot <-  
ggplot(weather,aes(x=dewp, y=humid))+geom\_point(na.rm = TRUE)+scale\_y\_log10()+ggtitle("Relationship between dewp and humid") + geom\_smooth(method = "lm", se = FALSE,na.rm = TRUE)  
dewp\_humid\_plot

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



# Plot to observe the relationship between `precip` and `visib`  
precip\_visib\_plot <- ggplot(weather, aes(x = precip, y = visib)) +  
 geom\_point(na.rm = TRUE) +  
 ggtitle("Relationship between precip and visib")   
  
precip\_visib\_plot



## 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?  
# Count the number of planes that have NA in the year column  
non\_manufacture\_date <- planes %>%   
 filter(is.na(year)) %>%  
 count()  
  
non\_manufacture\_date

# A tibble: 1 × 1  
 n  
 <int>  
1 70

# What are the five most common manufacturers?  
#Create a new table with the columns manufacturer and number\_of\_planes, arranged in descending order  
top\_manufacturers <- planes %>%   
 group\_by(manufacturer)%>%  
 summarise(number\_of\_planes=n())%>%  
arrange(,desc(number\_of\_planes)) %>%  
#slect the first five rows and all of the columns in the table  
 top\_n(5, number\_of\_planes)  
  
top\_manufacturers

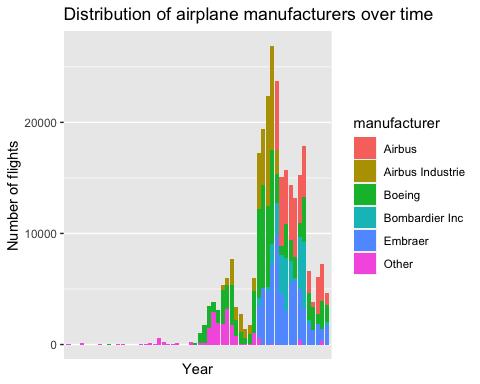
# A tibble: 5 × 2  
 manufacturer number\_of\_planes  
 <chr> <int>  
1 BOEING 1630  
2 AIRBUS INDUSTRIE 400  
3 BOMBARDIER INC 368  
4 AIRBUS 336  
5 EMBRAER 299

# Has the distribution of manufacturer changed over time as reflected by the airplanes flying from NYC in 2013? (Hint: you may need to use case\_when() to recode the manufacturer name and collapse rare vendors into a category called Other.)  
  
flights\_man <- flights %>%  
 inner\_join(planes, by = 'tailnum') %>%   
 mutate(manufacturer = case\_when(  
 grepl("BOEING", manufacturer) ~ "Boeing",  
 grepl("AIRBUS INDUSTRIE", manufacturer) ~ "Airbus Industrie",  
 grepl("EMBRAER", manufacturer) ~ "Embraer",  
 grepl("BOMBARDIER INC", manufacturer) ~ "Bombardier Inc",  
 grepl("AIRBUS", manufacturer) ~ "Airbus",  
 TRUE ~ "Other"  
 ))  
  
flights\_man

# A tibble: 284,170 × 27  
 year.x month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 517 515 2 830 819  
 2 2013 1 1 533 529 4 850 830  
 3 2013 1 1 542 540 2 923 850  
 4 2013 1 1 544 545 -1 1004 1022  
 5 2013 1 1 554 600 -6 812 837  
 6 2013 1 1 554 558 -4 740 728  
 7 2013 1 1 555 600 -5 913 854  
 8 2013 1 1 557 600 -3 709 723  
 9 2013 1 1 557 600 -3 838 846  
10 2013 1 1 558 600 -2 849 851  
# ℹ 284,160 more rows  
# ℹ 19 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>, year.y <int>, type <chr>,  
# manufacturer <chr>, model <chr>, engines <int>, seats <int>, speed <int>,  
# engine <chr>

ggplot(flights\_man, aes(x= year.y, fill = manufacturer)) +  
 geom\_bar() +  
 labs(title = "Distribution of airplane manufacturers over time",  
 x = "Year",  
 y = "Number of flights",  
 color = "Manufacturer") +  
 scale\_x\_discrete(breaks = seq(1930, 2020, 5)) +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

Warning: Removed 5306 rows containing non-finite values (`stat\_count()`).



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

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

#What is the oldest plane (specified by the tailnum variable) that flew from New York City airports in 2013?  
# Filter flights from New York airports  
flights\_nyc <- flights %>%  
 filter(origin %in% c("JFK", "LGA", "EWR"))  
  
# Left join to planes table to know the details on the planes  
flights\_nyc\_planes <- flights\_nyc %>%  
 left\_join(planes, by = "tailnum") %>%  
 arrange(year.y)   
flights\_nyc\_planes

# A tibble: 336,776 × 27  
 year.x 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 30 741 745 -4 1059 1125  
 2 2013 10 7 1525 1530 -5 1915 1845  
 3 2013 10 8 1737 1735 2 2052 2055  
 4 2013 11 7 817 745 32 1140 1100  
 5 2013 11 12 1528 1530 -2 1837 1845  
 6 2013 12 17 1043 1030 13 1416 1355  
 7 2013 12 18 808 800 8 1146 1135  
 8 2013 2 1 1526 1530 -4 1915 1910  
 9 2013 2 3 1036 1030 6 1411 1355  
10 2013 2 7 742 745 -3 1114 1125  
# ℹ 336,766 more rows  
# ℹ 19 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>, year.y <int>, type <chr>,  
# manufacturer <chr>, model <chr>, engines <int>, seats <int>, speed <int>,  
# engine <chr>

#find the oldest plane  
oldest\_plane <- filter(flights\_nyc\_planes, year.y == min(year.y, na.rm = TRUE)) %>%  
 distinct(tailnum)  
  
oldest\_plane

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

#How many airplanes that flew from New York City are included in the planes table?  
#remove lines with NA values and remove repeated tailnum  
planes\_clean <- na.omit(flights\_nyc\_planes) %>%  
 distinct(tailnum)  
  
planes\_clean

# A tibble: 23 × 1  
 tailnum  
 <chr>   
 1 N381AA   
 2 N201AA   
 3 N567AA   
 4 N378AA   
 5 N615AA   
 6 N425AA   
 7 N364AA   
 8 N621AA   
 9 N508AA   
10 N675MC   
# ℹ 13 more rows

#count the number of lines on the table  
count\_planes <- length(planes\_clean$tailnum)  
  
count\_planes

[1] 23

## 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.

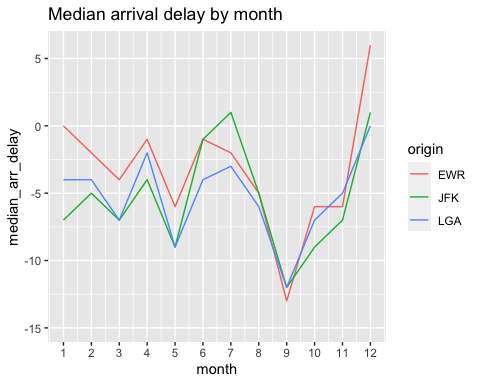
# What is the median arrival delay on a month-by-month basis in each airport?  
  
median\_delay <- flights %>%  
 group\_by(origin, month) %>%  
 summarize(median\_arr\_delay=median(arr\_delay, na.rm = TRUE))

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

median\_delay

# A tibble: 36 × 3  
# Groups: origin [3]  
 origin month median\_arr\_delay  
 <chr> <int> <dbl>  
 1 EWR 1 0  
 2 EWR 2 -2  
 3 EWR 3 -4  
 4 EWR 4 -1  
 5 EWR 5 -6  
 6 EWR 6 -1  
 7 EWR 7 -2  
 8 EWR 8 -5  
 9 EWR 9 -13  
10 EWR 10 -6  
# ℹ 26 more rows

ggplot(median\_delay, aes(x = month, y = median\_arr\_delay, color = origin)) +  
 geom\_line() +  
 ggtitle("Median arrival delay by month") +  
 expand\_limits(y = -15) +   
 scale\_x\_continuous(breaks = seq(1, 12, 1))



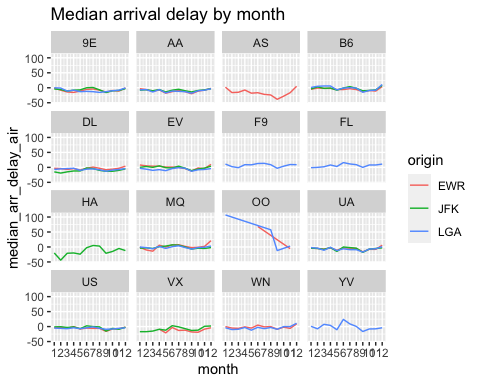
#For each airline, plot the median arrival delay for each month and origin airport.  
  
median\_delay\_airline <- flights %>%  
 group\_by(carrier, origin,month) %>%  
 summarize(median\_arr\_delay\_air=median(arr\_delay, na.rm = TRUE))

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

median\_delay\_airline

# A tibble: 399 × 4  
# Groups: carrier, origin [35]  
 carrier origin month median\_arr\_delay\_air  
 <chr> <chr> <int> <dbl>  
 1 9E EWR 1 -1   
 2 9E EWR 2 -8   
 3 9E EWR 3 -14   
 4 9E EWR 4 -15.5  
 5 9E EWR 5 -11   
 6 9E EWR 6 -5.5  
 7 9E EWR 7 -4   
 8 9E EWR 8 -8   
 9 9E EWR 9 -14.5  
10 9E EWR 10 -9   
# ℹ 389 more rows

ggplot(median\_delay\_airline, aes(x = month, y = median\_arr\_delay\_air, color = origin)) +  
 geom\_line() +  
 ggtitle("Median arrival delay by month") +  
 facet\_wrap(~ carrier) +  
 ylim(min(median\_delay\_airline$median\_arr\_delay\_air),max(median\_delay\_airline$median\_arr\_delay\_air))+  
 scale\_x\_continuous(breaks = seq(1, 12, 1))



## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United Air Lines Inc. not UA, the count (number) of times it flew to SFO, and the percent of the trips that that particular airline flew to SFO.

# join tables  
joint\_flights\_airlines <- airlines %>%  
 left\_join(flights, by = "carrier")   
  
joint\_flights\_airlines

# A tibble: 336,776 × 20  
 carrier name year month day dep\_time sched\_dep\_time dep\_delay arr\_time  
 <chr> <chr> <int> <int> <int> <int> <int> <dbl> <int>  
 1 9E Endeavo… 2013 1 1 810 810 0 1048  
 2 9E Endeavo… 2013 1 1 1451 1500 -9 1634  
 3 9E Endeavo… 2013 1 1 1452 1455 -3 1637  
 4 9E Endeavo… 2013 1 1 1454 1500 -6 1635  
 5 9E Endeavo… 2013 1 1 1507 1515 -8 1651  
 6 9E Endeavo… 2013 1 1 1530 1530 0 1650  
 7 9E Endeavo… 2013 1 1 1546 1540 6 1753  
 8 9E Endeavo… 2013 1 1 1550 1550 0 1844  
 9 9E Endeavo… 2013 1 1 1552 1600 -8 1749  
10 9E Endeavo… 2013 1 1 1554 1600 -6 1701  
# ℹ 336,766 more rows  
# ℹ 11 more variables: sched\_arr\_time <int>, arr\_delay <dbl>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

#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.  
  
#Planes that traveled to San Francisco   
  
fly\_into\_sfo <- joint\_flights\_airlines %>%  
 group\_by(name)%>%  
 summarize(total\_flights = n(), number=sum(dest == "SFO", na.rm = TRUE), percent = sum(dest == "SFO", na.rm = TRUE) / total\_flights \*100) %>%  
 filter(percent > 0) %>%  
 arrange(desc(percent))%>%  
 select(name, number, percent)  
fly\_into\_sfo

# A tibble: 5 × 3  
 name number percent  
 <chr> <int> <dbl>  
1 Virgin America 2197 42.6   
2 United Air Lines Inc. 6819 11.6   
3 American Airlines Inc. 1422 4.34  
4 Delta Air Lines Inc. 1858 3.86  
5 JetBlue Airways 1035 1.89

And here is some bonus ggplot code to plot your dataframe

# #| label: ggplot-flights-toSFO  
# #| message: false  
# #| warning: false  
#   
# 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 organize our data manipulation to create the following plot. No need to write the code, just explain in words how you would go about it.

Answer:

1. Left join flights table with airlines table to have te full name of the airlines
2. I would group by : name, origin, month
3. with a column that summarizes count = n()
4. Plot in a bar graph with x = month and y = count
5. Finally, use facet\_wrap(~ origin) + facet\_grid(origin ~ name)



## 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!

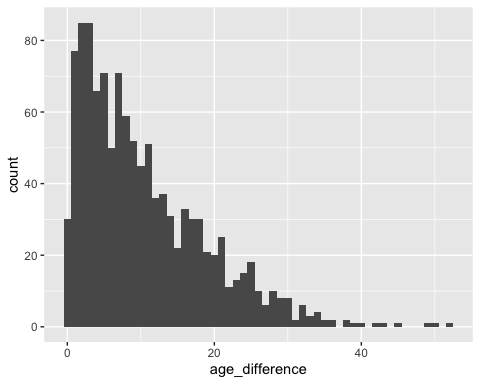
The following is a data dictionary of the variables used

| 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 is `age\_difference` distributed? What's the 'typical' `age\_difference` in movies?  
  
ggplot(age\_gaps, aes(x=age\_difference)) +   
 geom\_histogram(binwidth = 1)



#How frequently does this rule apply in this dataset?  
  
half\_plus\_seven <- age\_gaps %>%  
 mutate(lower\_bound = actor\_1\_age/2+7, upper\_bound = (actor\_1\_age-7)\*2) %>%  
 mutate(rule\_ok = ifelse(actor\_2\_age >= lower\_bound & actor\_2\_age <= upper\_bound, "Yes", "No"))   
  
frequency\_rule\_yes <- half\_plus\_seven %>%  
 summarise(total\_movies = n(), percent\_rule\_yes = sum(rule\_ok == "Yes", na.rm = TRUE)/total\_movies\*100) %>%  
 select(percent\_rule\_yes)  
  
frequency\_rule\_yes

# A tibble: 1 × 1  
 percent\_rule\_yes  
 <dbl>  
1 71.8

#Which movie has the greatest number of love interests?  
  
love\_interests\_movies <- age\_gaps %>%  
 group\_by(movie\_name)%>%   
 summarize(couple\_number = n())%>%  
 arrange(desc(couple\_number))  
  
love\_interests\_movies

# A tibble: 830 × 2  
 movie\_name couple\_number  
 <chr> <int>  
 1 Love Actually 7  
 2 The Family Stone 6  
 3 A View to a Kill 5  
 4 He's Just Not That Into You 5  
 5 Mona Lisa Smile 5  
 6 A Star Is Born 4  
 7 American Pie 4  
 8 Boogie Nights 4  
 9 Closer 4  
10 Pushing Tin 4  
# ℹ 820 more rows

#Which actors/ actresses have the greatest number of love interests in this dataset?  
   
love\_interests\_actors1 <- age\_gaps %>%   
 group\_by(actor\_name = actor\_1\_name)%>%   
 summarize(couple\_number = n())%>%  
 arrange(desc(couple\_number))  
  
love\_interests\_actors2 <- age\_gaps %>%   
 group\_by(actor\_name = actor\_2\_name)%>%   
 summarize(couple\_number = n())%>%  
 arrange(desc(couple\_number))   
  
love\_interests\_actors<- rbind(love\_interests\_actors1, love\_interests\_actors2) %>%  
 arrange(desc(couple\_number))  
  
love\_interests\_actors

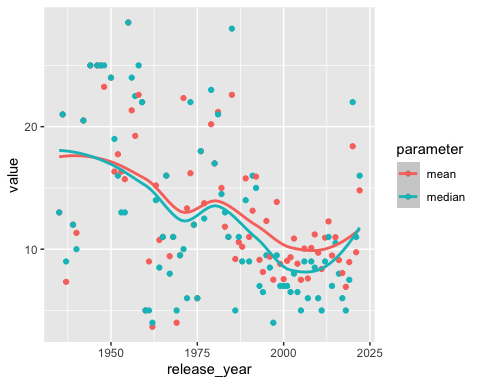
# A tibble: 1,214 × 2  
 actor\_name couple\_number  
 <chr> <int>  
 1 Keanu Reeves 24  
 2 Adam Sandler 18  
 3 Roger Moore 17  
 4 Sean Connery 15  
 5 Harrison Ford 13  
 6 Keira Knightley 13  
 7 Scarlett Johansson 13  
 8 Johnny Depp 12  
 9 Pierce Brosnan 12  
10 Leonardo DiCaprio 11  
# ℹ 1,204 more rows

#Is the mean/median age difference staying constant over the years (1935 - 2022)?  
  
  
mm\_age\_diff <- age\_gaps %>%  
 group\_by(release\_year) %>%  
 summarise(mean=mean(age\_difference), median = median(age\_difference)) %>%  
 pivot\_longer(  
 cols = 2:3,  
 names\_to = "parameter",  
 values\_to = "value"  
 )  
  
mm\_age\_diff

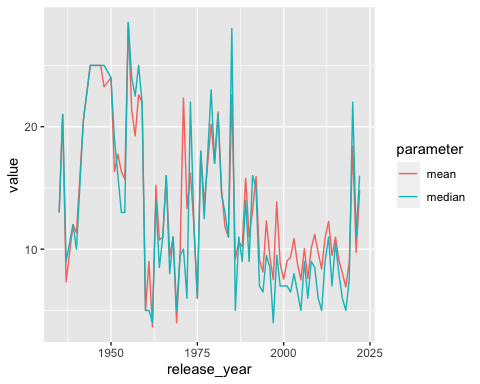
# A tibble: 164 × 3  
 release\_year parameter value  
 <dbl> <chr> <dbl>  
 1 1935 mean 13   
 2 1935 median 13   
 3 1936 mean 21   
 4 1936 median 21   
 5 1937 mean 7.33  
 6 1937 median 9   
 7 1939 mean 12   
 8 1939 median 12   
 9 1940 mean 11.3   
10 1940 median 10   
# ℹ 154 more rows

ggplot(mm\_age\_diff, aes(x=release\_year, y = value, color = parameter)) + geom\_point() + geom\_smooth(aes(y = value), level = 0)

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



ggplot(mm\_age\_diff, aes(x=release\_year, y = value, color = parameter)) + geom\_line()



#How frequently does Hollywood depict same-gender love interests?  
  
same\_gender <- age\_gaps %>%  
 mutate(gender\_equal = ifelse(character\_1\_gender == character\_2\_gender, "Yes", "No"))%>%  
 summarise(total\_movies = n(), percent\_same\_gender = sum(gender\_equal == "Yes", na.rm = TRUE)/total\_movies\*100) %>%  
 select(percent\_same\_gender)  
  
same\_gender

# A tibble: 1 × 1  
 percent\_same\_gender  
 <dbl>  
1 1.99

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 committing and pushing tour changes to your own Github repo as you go along.

# Details

* Who did you collaborate with: Nicholas Arnovitz, Angela Zhong, Harry Heo, Deven Jonbanputra
* Approximately how much time did you spend on this problem set: 10 Hours
* What, if anything, gave you the most trouble: group\_by

**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.