Homerwork 1

SEAN O’DOHERTY

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: 10,034 × 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 10,024 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

# A tibble: 7,198 × 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 7,188 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: 29 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 27 1419 1420 -1 1754 1550  
 2 2013 10 7 1350 1350 0 1736 1526  
 3 2013 10 7 1357 1359 -2 1858 1654  
 4 2013 10 16 657 700 -3 1258 1056  
 5 2013 11 1 658 700 -2 1329 1015  
 6 2013 3 18 1844 1847 -3 39 2219  
 7 2013 4 17 1635 1640 -5 2049 1845  
 8 2013 4 18 558 600 -2 1149 850  
 9 2013 4 18 655 700 -5 1213 950  
10 2013 5 22 1827 1830 -3 2217 2010  
# ℹ 19 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

# 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>

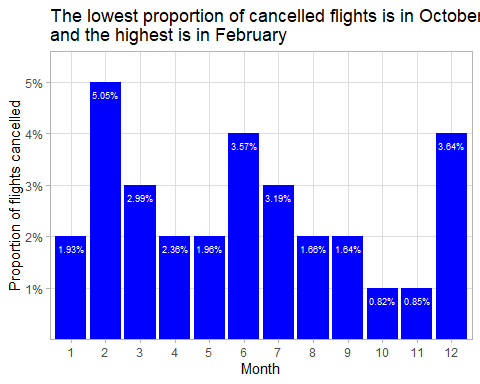
## Problem 2: What months had the highest and lowest proportion of cancelled flights? Interpret any seasonal patterns. To determine if a flight was cancelled use the following code

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

# // What months had the highest and lowest % of cancelled flights?  
  
# Assign to new table  
flights\_month\_prop\_cancelled <-  
   
 flights %>%  
   
 # Group by month  
 group\_by(month) %>%   
   
 # calculate the sum of non-na flights divided by count of flights  
 summarize("prop\_cancelled" = sum(is.na(dep\_time)) / n())  
  
  
# Print results sorted by prop\_cancelled  
flights\_month\_prop\_cancelled %>%   
   
 # sort by prop\_cancelled to get low to high  
 arrange(prop\_cancelled)

# A tibble: 12 × 2  
 month prop\_cancelled  
 <int> <dbl>  
 1 10 0.00817  
 2 11 0.00854  
 3 9 0.0164   
 4 8 0.0166   
 5 1 0.0193   
 6 5 0.0196   
 7 4 0.0236   
 8 3 0.0299   
 9 7 0.0319   
10 6 0.0357   
11 12 0.0364   
12 2 0.0505

# Plot line graph  
ggplot(  
   
 # use dataset from above  
 flights\_month\_prop\_cancelled  
   
 # set x and y axes, use factor(to make month discreet)  
 ,aes(  
 x = factor(month)  
 ,y = percent(prop\_cancelled, 1)  
 ,group=1  
 )  
  
) +  
   
 # add line to graph  
 geom\_col(  
 fill = "blue"  
 ) +  
   
 # Add value labels  
 geom\_text(  
 aes(  
 label = percent(prop\_cancelled, 0.01)  
 )  
 ,color = "white"  
 ,hjust = 0.5  
 ,vjust = 2  
 ,size = 2.5  
 ) +  
   
 # Add Axis labels  
 labs(  
 title = "The lowest proportion of cancelled flights is in October \nand the highest is in February"  
 ,x = "Month"  
 ,y = "Proportion of flights cancelled"  
 ) +  
   
 # Apply theme  
 theme\_light()



The lowest proportion of cancelled flights is in October and the highest is in February.

## 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.

# store to new table  
flights\_planes\_50p <-  
   
 # Left join flights to planes to get the number of seats  
 left\_join(x = flights, y = planes, by = "tailnum") %>%   
   
 # Filter to only planes with > 50 seats / flights which have a dep\_time  
 filter(seats > 50, !is.na(dep\_time))  
  
  
# store to new table  
planes\_50p\_flight\_count\_top1 <-  
   
 # use top\_n to return only the top row   
 top\_n(  
 flights\_planes\_50p %>%  
   
 # Group by plane and count flights  
 group\_by(tailnum) %>%  
   
 # Sort descending  
 summarise(flight\_count = n()) %>%  
 arrange(desc(flight\_count))  
   
 ,1  
 )

Selecting by flight\_count

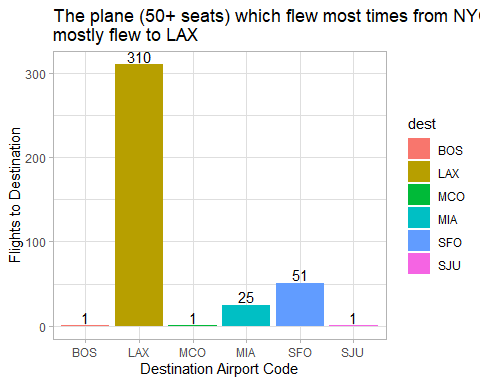
# Display results  
planes\_50p\_flight\_count\_top1

# A tibble: 1 × 2  
 tailnum flight\_count  
 <chr> <int>  
1 N328AA 389

# store to new table  
flights\_planes\_50p\_flight\_count\_top1 <-  
  
 # join the flights of planes with 50+ seats to table with top 1 by flight count   
 right\_join(  
 flights\_planes\_50p  
 ,planes\_50p\_flight\_count\_top1  
 ,by = "tailnum"  
 ) %>%   
   
 # Group by destination  
 group\_by(dest) %>%  
   
 #summarise to get count of flights (with dep\_time)  
 summarise(dest\_flight\_count = sum(!is.na(dep\_time)))  
   
# print table  
flights\_planes\_50p\_flight\_count\_top1

# A tibble: 6 × 2  
 dest dest\_flight\_count  
 <chr> <int>  
1 BOS 1  
2 LAX 310  
3 MCO 1  
4 MIA 25  
5 SFO 51  
6 SJU 1

# Create ggplot  
ggplot(  
   
 # call previous table  
 flights\_planes\_50p\_flight\_count\_top1  
   
 # define plot aesthetics  
 ,aes(  
 x = dest  
 ,y = dest\_flight\_count  
 ,fill = dest  
 )  
  
) +  
   
 # Add bar to chart  
 geom\_bar(stat = "identity") +  
   
 # Add value labels  
 geom\_text(aes(label = dest\_flight\_count), vjust = -0.2) +  
   
 # Add title and axis labels  
 labs(  
 x = "Destination Airport Code"  
 ,y = "Flights to Destination"  
 ,title = "The plane (50+ seats) which flew most times from NYC in 2013 \nmostly flew to LAX"  
 ) +  
   
 # Apply theme  
 theme\_light()



The plane (which has 50 or more seats) which flew the most times from NYC is tailnum N328AA.

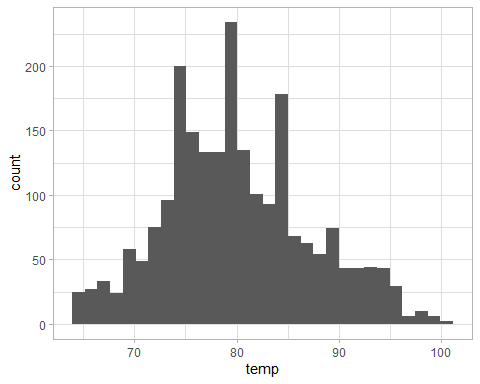
The breakdown of where this plane flew to is included in table and bar chart form. There were many flights to LAX and some to SFO and MIA. There was only 1 flight to each of BOS, MCO and SJU.

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

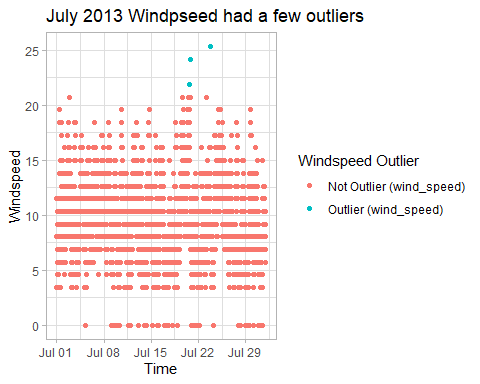
- What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
- What is the relationship between `dewp` and `humid`?  
- What is the relationship between `precip` and `visib`?

# // What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
  
# define function to find outliers  
findoutlier <-  
 function(x) {  
 return(x < quantile(x, .25) - 1.5\*IQR(x) | x > quantile(x, .75) + 1.5\*IQR(x))  
}  
  
# store to new table  
weather\_july <-  
   
 # filter weather to july  
 filter(weather, month == 7)  
  
# histogram of temperature  
ggplot(  
 weather\_july  
 ,aes(x = temp)  
) +  
 geom\_histogram() +  
 theme\_light()

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



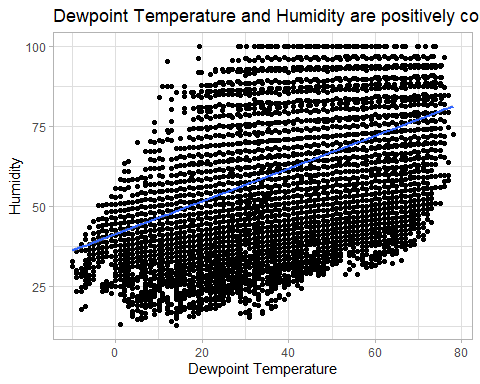
# store to table  
weather\_july\_windspeed <-  
  
 # filter out na values of windspeed  
 weather\_july %>%  
 filter(!is.na(wind\_speed)) %>%   
   
 # Create new col which identifies wind\_speed outliers  
 mutate(  
 wind\_speed\_outlier =  
 ifelse(  
 findoutlier(wind\_speed)  
 ,"Outlier (wind\_speed)"  
 ,"Not Outlier (wind\_speed)"  
 )  
 )  
  
# Create plot of wind speed through time with color identifying outliers  
ggplot(  
 weather\_july\_windspeed  
 ,aes(  
 x = time\_hour  
 ,y = wind\_speed  
 ,color = wind\_speed\_outlier  
 )  
) +  
   
 # Add scatter  
 geom\_point() +  
   
 # Add labels and theme  
 labs(  
 x = "Time"  
 ,y = "Windspeed"  
 ,title = "July 2013 Windpseed had a few outliers"  
 ,colour = "Windspeed Outlier"  
 ) +  
 theme\_light()



# // What is the relationship between `dewp` and `humid`?  
   
# Create plot of dewp vs humid  
ggplot(  
 weather  
 ,aes(  
 x = dewp  
 ,y = humid  
 )  
) +  
   
 # make scatter plot  
 geom\_point() +  
   
 # Use stat\_smooth to add a regression line  
 stat\_smooth(  
 method = "lm",  
 formula = y ~ x,  
 geom = "smooth"  
 ) +  
   
 # Add labels and theme  
 labs(  
 x = "Dewpoint Temperature"  
 ,y = "Humidity"  
 ,title = "Dewpoint Temperature and Humidity are positively correlated"  
 ) +  
 theme\_light()

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

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

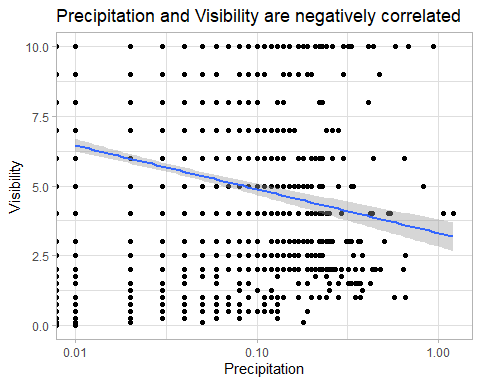


# // What is the relationship between `precip` and `visib`?  
  
# Scatter plot of precip vs visib  
# Use stat\_smooth to add a regression line  
# Convert x axis to log scale to make easier to read  
  
# Create plot of precip vs visib  
ggplot(  
 weather  
 ,aes(x = precip, y = visib)  
) +  
   
 # add scatter  
 geom\_point() +  
   
 # add regression line  
 stat\_smooth(  
 method = "lm",  
 formula = y ~ x,  
 geom = "smooth"  
 ) +  
   
 # put x on a log scale to aid leibility  
 scale\_x\_log10(  
 ) +  
   
 # add labels and theme  
 labs(  
 x = "Precipitation"  
 ,y = "Visibility"  
 ,title = "Precipitation and Visibility are negatively correlated"  
 ) +  
 theme\_light()

Warning: Transformation introduced infinite values in continuous x-axis

Warning: Transformation introduced infinite values in continuous x-axis

Warning: Removed 24366 rows containing non-finite values (`stat\_smooth()`).



## 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 to planes which do not have a manufacture year  
 filter(is.na(year)) %>%   
   
 # return count  
 summarise(planecount = n())

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

# // What are the five most common manufacturers?  
  
# store to new table  
manuf\_flightcount\_planecount <-  
   
 # join flights which have dep\_time to planes by tailnum  
 inner\_join(  
 x = filter(flights, !is.na(dep\_time))  
 ,y = planes  
 ,by = "tailnum"  
 ) %>%  
   
 # group by manufacturer  
 group\_by(manufacturer) %>%   
   
 # use summarise to get plane\_count and flight count  
 summarise(plane\_count = n\_distinct(tailnum), flight\_count = n())  
  
# store results top 5 by plane\_count and flight\_count  
manuf\_planecount\_top5 <-  
 top\_n(  
 manuf\_flightcount\_planecount %>%   
 arrange(desc(plane\_count))  
 ,5  
 )

Selecting by flight\_count

# store results top 5 by plane\_count and flight\_count  
manuf\_flightcount\_top5 <-  
 top\_n(  
 manuf\_flightcount\_planecount %>%   
 arrange(desc(flight\_count))  
 ,5  
 )

Selecting by flight\_count

#display results  
manuf\_planecount\_top5

# A tibble: 5 × 3  
 manufacturer plane\_count flight\_count  
 <chr> <int> <int>  
1 BOEING 1627 82524  
2 AIRBUS INDUSTRIE 400 40753  
3 BOMBARDIER INC 366 27588  
4 AIRBUS 336 47009  
5 EMBRAER 299 63783

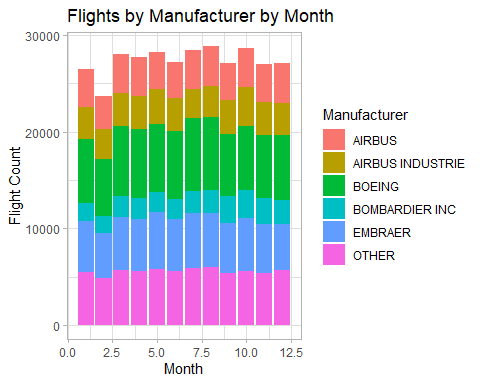
manuf\_flightcount\_top5

# A tibble: 5 × 3  
 manufacturer plane\_count flight\_count  
 <chr> <int> <int>  
1 BOEING 1627 82524  
2 EMBRAER 299 63783  
3 AIRBUS 336 47009  
4 AIRBUS INDUSTRIE 400 40753  
5 BOMBARDIER INC 366 27588

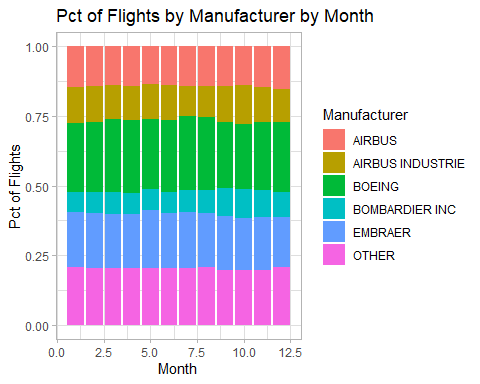
# // Has the distribution of manufacturer changed over time as reflected by the airplanes flying from NYC in 2013? (Hint: you may need to use case\_when() to recode the manufacturer name and collapse rare vendors into a category called Other.)  
  
# Left join flights (with dep\_time) to planes on tailnumber to get manufacturer  
# Left join to manuf\_planecount\_top5 to identify top 5 (use keep = TRUE)  
# Mutate NAs to "OTHER"  
# Group by Manufacturer and month and count flights  
  
# store to new table  
flights\_manuf\_month <-  
   
 # Join the flights with manufacturer to the top 5 manufacturers by flight count  
 left\_join(  
 x =  
   
 # join flights which have dep\_time to planes by tailnum to get manufacturer  
 left\_join(  
 x = filter(flights, !is.na(dep\_time))  
 ,y = planes  
 ,by = "tailnum"  
 )  
 ,y = manuf\_flightcount\_top5  
 ,by = "manufacturer"  
 ,keep = TRUE  
 ) %>%  
   
 # Replace the manufacturer col from the top5 table with "OTHER" where there is no match  
 mutate(manufacturer = replace\_na(manufacturer.y, "OTHER")) %>%  
   
 # group by manufacturer (and other) and month  
 group\_by(manufacturer, month) %>%  
   
 # get flight counts  
 summarise(flight\_count = n())

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

# Create plot of flight count by month with manufacturers in fill  
ggplot(  
 flights\_manuf\_month  
 ,aes(x = month, y = flight\_count, fill = manufacturer)  
) +  
   
 # Add stacked bars  
 geom\_bar(position = "stack", stat = "identity"  
) +  
   
 # Add labels and theme  
 labs(  
 x = "Month"  
 ,y = "Flight Count"  
 ,fill = "Manufacturer"  
 ,title = "Flights by Manufacturer by Month"  
 ) +  
 theme\_light()



# Create plot of flight count by month with manufacturers in fill  
ggplot(  
 flights\_manuf\_month  
 ,aes(x = month, y = flight\_count, fill = manufacturer)  
) +  
   
 # add pct of total bars  
 geom\_bar(position = "fill", stat = "identity"  
) +  
   
 # Add labels and theme  
 labs(  
 x = "Month"  
 ,y = "Pct of Flights"  
 ,fill = "Manufacturer"  
 ,title = "Pct of Flights by Manufacturer by Month"  
 ) +  
 theme\_light()



## 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?  
  
# join tailnums with dep\_time to planes to get manufacture year  
left\_join(  
   
 # lsit of tailnums with dep\_time  
 x = flights %>%   
 filter(!is.na(dep\_time)) %>%   
 group\_by(tailnum) %>%   
 summarise()  
 ,y = planes  
 ,by = "tailnum"  
) %>%  
   
 # sort by year ascending  
 arrange(year)

# A tibble: 4,037 × 9  
 tailnum year type manufacturer model engines seats speed engine  
 <chr> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 N381AA 1956 Fixed wing multi… DOUGLAS DC-7… 4 102 232 Recip…  
 2 N201AA 1959 Fixed wing singl… CESSNA 150 1 2 90 Recip…  
 3 N567AA 1959 Fixed wing singl… DEHAVILLAND OTTE… 1 16 95 Recip…  
 4 N378AA 1963 Fixed wing singl… CESSNA 172E 1 4 105 Recip…  
 5 N575AA 1963 Fixed wing singl… CESSNA 210-… 1 6 NA Recip…  
 6 N14629 1965 Fixed wing multi… BOEING 737-… 2 149 NA Turbo…  
 7 N615AA 1967 Fixed wing multi… BEECH 65-A… 2 9 202 Turbo…  
 8 N425AA 1968 Fixed wing singl… PIPER PA-2… 1 4 107 Recip…  
 9 N383AA 1972 Fixed wing multi… BEECH E-90 2 10 NA Turbo…  
10 N364AA 1973 Fixed wing multi… CESSNA 310Q 2 6 167 Recip…  
# ℹ 4,027 more rows

# // How many airplanes that flew from New York City are included in the planes table?  
  
#Inner join flights to planes to get matches  
inner\_join(  
   
 x = flights %>%  
   
 # filter to flights with a dep\_time  
 filter(!is.na(dep\_time))  
   
 ,y = planes  
 ,by = "tailnum"  
) %>%   
  
 # get count of distinct tailnumbers  
 summarise(n\_distinct(tailnum))

# A tibble: 1 × 1  
 `n\_distinct(tailnum)`  
 <int>  
1 3316

## 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.

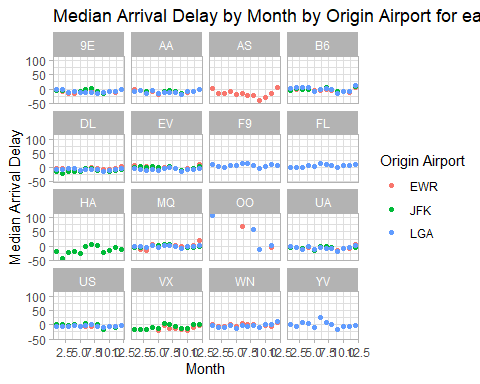
# //What is the median arrival delay on a month-by-month basis in each airport?  
  
# filter on flights with dep\_time  
filter(flights, !is.na(dep\_time)) %>%   
   
 # group by origin and month  
 group\_by(origin, month) %>%   
   
 # calculate median\_arr\_delay ignoring na using summarise  
 summarise(median\_arr\_delay = median(arr\_delay,na.rm=TRUE))

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

# 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

# // For each airline, plot the median arrival delay for each month and origin airport.  
  
# create plot of median\_arr\_delay by month with origin as color  
ggplot(  
   
 # filter on flights with dep\_time  
 filter(flights, !is.na(dep\_time)) %>%   
   
 # group by carrier, origin and month  
 group\_by(carrier, origin, month) %>%   
   
 # calculate median\_arr\_delay ignoring na using summarise  
 summarise(median\_arr\_delay = median(arr\_delay,na.rm=TRUE))  
   
 ,aes(  
 x = month  
 ,y = median\_arr\_delay  
 ,color = origin  
 )  
) +  
   
 # Add scatter  
 geom\_point(  
) +  
   
 # Facet wrap to dispay each carrier separately  
 facet\_wrap(~carrier  
) +  
   
 # Add labels and theme  
 labs(  
 x = "Month"  
 ,y = "Median Arrival Delay"  
 ,color = "Origin Airport"  
 ,title = "Median Arrival Delay by Month by Origin Airport for each Carrier"  
 ) +  
 theme\_light()

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



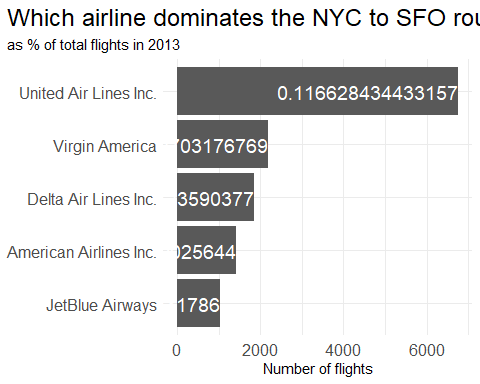
## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United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

# store to new table  
fly\_into\_sfo <-  
   
 # join flights with dep\_time to airlines to get full name  
 left\_join(  
 x =filter(flights, !is.na(dep\_time))  
 ,y = airlines  
 ,by = "carrier"  
 ) %>%   
   
 # group by full name  
 group\_by(name) %>%   
   
 # get SFO count and PCT (divide by total count)  
 summarise(  
 count = sum(dest == "SFO")  
 ,percent = sum(dest == "SFO") / n()  
 ) %>%   
   
 # filter to only include those which fly to SFO  
 filter(count > 0)  
  
  
# display results  
fly\_into\_sfo

# A tibble: 5 × 3  
 name count percent  
 <chr> <int> <dbl>  
1 American Airlines Inc. 1402 0.0437  
2 Delta Air Lines Inc. 1850 0.0387  
3 JetBlue Airways 1029 0.0190  
4 United Air Lines Inc. 6762 0.117   
5 Virgin America 2187 0.426

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))

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 obtain the plot below I would first group the data by month, carrier and origin and summarise the number of cancelled flights.*

*I would then create month\_mmm by mutating the month field using month.abb.*

*I would then create a box plot with month\_mmm on the x axis and cancelled flights on the y axis.*

*I would then use facet\_grid to break the plot out by origin (cols) and carrier (rows).*

## 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!

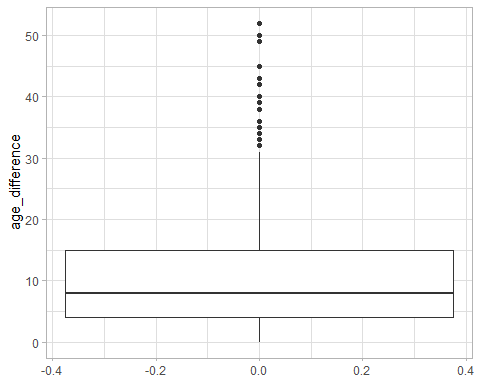
The following is a data dictionary of the variables used

| 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')

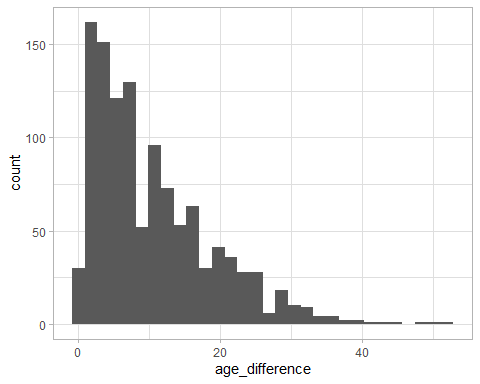
Rows: 1155 Columns: 13  
── Column specification ────────────────────────────────────────────────────────  
Delimiter: ","  
chr (6): movie\_name, director, actor\_1\_name, actor\_2\_name, character\_1\_gend...  
dbl (5): release\_year, age\_difference, couple\_number, actor\_1\_age, actor\_2\_age  
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?  
  
# Use boxplot to explore age\_difference distribution  
ggplot(  
 age\_gaps  
 ,aes(y = age\_difference)  
) +  
 geom\_boxplot() +  
 theme\_light()



# Use histogram to explore age\_difference distribution  
ggplot(  
 age\_gaps  
 ,aes(x = age\_difference)  
) +  
 geom\_histogram() +  
 theme\_light()

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



# // How frequently does this rule apply in this dataset?  
  
age\_gaps %>%   
   
 # create fields for the older and younger actors ages from the two age fields  
 mutate(  
 older = if\_else(actor\_1\_age > actor\_2\_age, actor\_1\_age, actor\_2\_age)  
 ,younger = if\_else(actor\_1\_age < actor\_2\_age, actor\_1\_age, actor\_2\_age)  
 ) %>%   
   
 # Check if half age plus 7 holds  
 mutate(half\_age\_plus\_7 = if\_else(younger >= ((older / 2) + 7), TRUE, FALSE)  
 ) %>%   
   
 # group by whether the rule holds  
 group\_by(half\_age\_plus\_7) %>%   
   
 # count instances  
 summarise(n())

# A tibble: 2 × 2  
 half\_age\_plus\_7 `n()`  
 <lgl> <int>  
1 FALSE 326  
2 TRUE 829

# // Which movie has the greatest number of love interests?  
  
# union the two lists together  
union(  
   
 # get lists of actors in each movie  
 age\_gaps %>%   
 group\_by(movie\_name, actor\_name = actor\_1\_name) %>%   
 summarise()  
 ,age\_gaps %>%   
 group\_by(movie\_name, actor\_name = actor\_2\_name) %>%   
 summarise()  
) %>%   
   
 # group by movie  
 group\_by(movie\_name) %>%   
   
 # count distinct actors  
 summarise(actor\_count = n()) %>%   
   
 # sort descending  
 arrange(desc(actor\_count))

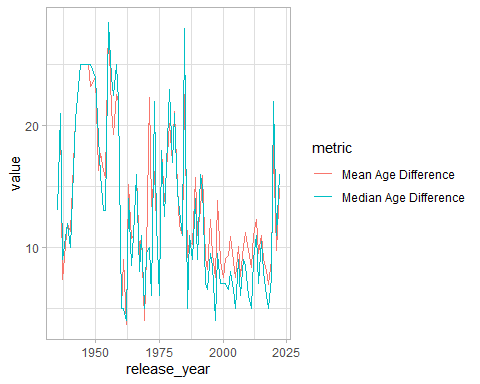
`summarise()` has grouped output by 'movie\_name'. You can override using the  
`.groups` argument.  
`summarise()` has grouped output by 'movie\_name'. You can override using the  
`.groups` argument.

# A tibble: 830 × 2  
 movie\_name actor\_count  
 <chr> <int>  
 1 Love Actually 13  
 2 The Family Stone 10  
 3 He's Just Not That Into You 9  
 4 Mona Lisa Smile 9  
 5 A Star Is Born 8  
 6 American Pie 8  
 7 Sex and the City 8  
 8 Twilight 8  
 9 Boogie Nights 7  
10 Soul Food 7  
# ℹ 820 more rows

# // Which actors/ actresses have the greatest number of love interests in this dataset?  
  
# union the lists together to and keep duplicates   
union\_all(  
   
 # get list of actors   
 select(age\_gaps, actor\_name = actor\_1\_name)  
 ,select(age\_gaps, actor\_name = actor\_2\_name)  
) %>%   
   
 # group by actor  
 group\_by(actor\_name) %>%   
   
 # count instances  
 summarise(love\_interests = n()) %>%   
   
 # sort descending  
 arrange(desc(love\_interests))

# A tibble: 1,031 × 2  
 actor\_name love\_interests  
 <chr> <int>  
 1 Keanu Reeves 27  
 2 Adam Sandler 20  
 3 Leonardo DiCaprio 17  
 4 Roger Moore 17  
 5 Sean Connery 17  
 6 Keira Knightley 14  
 7 Pierce Brosnan 14  
 8 Harrison Ford 13  
 9 Reese Witherspoon 13  
10 Scarlett Johansson 13  
# ℹ 1,021 more rows

# //Is the mean/median age difference staying constant over the years (1935 - 2022)?  
  
# create plot  
ggplot(  
   
 # create combined table with release\_year, metric and value to plot on same chart  
 union(  
   
 # get mean age difference by year  
 age\_gaps %>%  
 group\_by(release\_year) %>%   
 summarize(  
 metric = "Mean Age Difference"  
 ,value = mean(age\_difference)  
 )  
   
 # get median age difference by year  
 ,age\_gaps %>%   
 group\_by(release\_year) %>%   
 summarize(  
 metric = "Median Age Difference"  
 ,value = median(age\_difference)  
 )  
 )  
   
 # plot release year and value with metic as color  
 ,aes(x = release\_year, y = value, color = metric)  
) +  
   
 # add lines to chart  
 geom\_line() +  
 theme\_light()



# // How frequently does Hollywood depict same-gender love interests?  
  
age\_gaps %>%   
   
 # Create column which returns TRUE when both characters are the same gender  
 mutate(same\_sex = if\_else(character\_1\_gender == character\_2\_gender, TRUE, FALSE)) %>%  
   
 # group by same sex test  
 group\_by(same\_sex) %>%   
   
 # get counts  
 summarize(n())

# A tibble: 2 × 2  
 same\_sex `n()`  
 <lgl> <int>  
1 FALSE 1132  
2 TRUE 23

How would you explore this data set? Here are some ideas of tables/ graphs to help you with your analysis

* How is age\_difference distributed? What’s the ‘typical’ age\_difference in movies?
* 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: *na*
* Approximately how much time did you spend on this problem set: *a few hours*
* What, if anything, gave you the most trouble: *researching functions*

**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.