Homerwork 1

Assad Ahmed

5/14/23

# Data Manipulation

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

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

#Glimpse data set to view clumn names easily  
dplyr::glimpse(flights)

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

#Utilising the table flights from the nycflights 13 library  
  
# Had an arrival delay of two or more hours (> 120 minutes)  
 #Create new dataframe to filter delays >=120mins  
arr\_delay\_120mins <- flights %>%   
 #fliter for delays on arrival greater than or equal to 120 mins  
 filter(arr\_delay>=120)  
  
#View new data frame top entries  
head(arr\_delay\_120mins)

# A tibble: 6 × 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  
# 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)  
# create new data frame for the subset of arrivals into IAH and HOU  
IAH\_HOU <- flights %>%   
 filter(dest == "IAH" | dest== "HOU")  
  
#view new data frame  
head(IAH\_HOU)

# A tibble: 6 × 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  
# 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`)  
#New data frame for the fights operated by the 3 airlines  
UA\_AA\_DL <- flights %>%   
 filter(carrier == "UA" | carrier== "AA"| carrier== "DL")  
  
#view new data frame  
head(UA\_AA\_DL)

# A tibble: 6 × 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  
# 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)  
# July, August, September are months 7-9 only   
#Define new data frame for summer flights  
Summer <- flights %>%   
 filter(month %in% c(7,8,9))  
  
#view new data frame  
head(Summer)

# A tibble: 6 × 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  
# 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  
left\_ontime\_ari\_late <- flights %>%  
 filter(arr\_delay>=120 & dep\_delay<=0)  
  
#view new data frame  
head(left\_ontime\_ari\_late)

# A tibble: 6 × 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  
# 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  
madeup\_30min\_delay <- flights %>%  
 filter(dep\_delay>=60 & (arr\_delay-dep\_delay)<=-30)%>%  
 mutate(arr\_minus\_dep\_delays =arr\_delay-dep\_delay)  
  
#view new data frame  
head(madeup\_30min\_delay)

# A tibble: 6 × 20  
 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 1716 1545 91 2140 2039  
2 2013 1 1 2205 1720 285 46 2040  
3 2013 1 1 2326 2130 116 131 18  
4 2013 1 3 1503 1221 162 1803 1555  
5 2013 1 3 1821 1530 171 2131 1910  
6 2013 1 3 1839 1700 99 2056 1950  
# ℹ 12 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>, arr\_minus\_dep\_delays <dbl>

## Problem 2: What months had the highest and lowest proportion of cancelled flights? Interpret any seasonal patterns.

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

# What months had the highest and lowest % of cancelled flights?  
  
high\_low\_cancel <- flights %>%   
 #Group by month  
 group\_by(month) %>%   
 #proportion of cancelled flights calculated by converting dep\_time to True/false  
 # Taking the total true values and dividng by the lenght of the created vector  
 summarise(prop\_Cancelled = sum(is.na(dep\_time)) / length(dep\_time),  
 NumberofFLights = length(dep\_time))  
  
#view top entries  
#highest % canceled  
  
highest <- high\_low\_cancel %>%   
 #filter for max month only  
 filter(prop\_Cancelled==max(prop\_Cancelled))  
#view highest month only  
highest

# A tibble: 1 × 3  
 month prop\_Cancelled NumberofFLights  
 <int> <dbl> <int>  
1 2 0.0505 24951

lowest <- high\_low\_cancel %>%   
 #filter for max month only  
 filter(prop\_Cancelled==min(prop\_Cancelled))  
#view lowest month only  
lowest

# A tibble: 1 × 3  
 month prop\_Cancelled NumberofFLights  
 <int> <dbl> <int>  
1 10 0.00817 28889

**Interpreting the data:**

From the above summarized data we can see that between months 8-11 (assumed August-November) there are the lowest proportions of cancellations. Month 2 (February) has the larges proportion of cancellations (~5%), closely followed by Months 12/6 at ( 3.6% each). This implies that over the busy holiday periods in June/December cancellations are more likely. February appears to be an outlier in terms of seasonality, with potentially the weather conditions in that month leading towards the increased cancellations.

## 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.

#Glimpse the tables to identify concurrent columns and datatypes  
dplyr::glimpse(flights)

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

dplyr::glimpse(planes)

Rows: 3,322  
Columns: 9  
$ tailnum <chr> "N10156", "N102UW", "N103US", "N104UW", "N10575", "N105UW…  
$ year <int> 2004, 1998, 1999, 1999, 2002, 1999, 1999, 1999, 1999, 199…  
$ type <chr> "Fixed wing multi engine", "Fixed wing multi engine", "Fi…  
$ manufacturer <chr> "EMBRAER", "AIRBUS INDUSTRIE", "AIRBUS INDUSTRIE", "AIRBU…  
$ model <chr> "EMB-145XR", "A320-214", "A320-214", "A320-214", "EMB-145…  
$ engines <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
$ seats <int> 55, 182, 182, 182, 55, 182, 182, 182, 182, 182, 55, 55, 5…  
$ speed <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  
$ engine <chr> "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turb…

#Talinumber is chr type in both tables, implies direct join possible  
flights\_2013 <-flights %>%   
 #Filter for 2013 only, and remove cancelled flights as the plane did not leave in those instances  
 filter( year == 2013 & !is.na(dep\_time)) %>%   
 #Group by plane ID  
 group\_by(tailnum) %>%   
 #count number of flights by tailnumbers  
 summarise(count = n()) %>%   
 #Order largest to smallest number of flights  
 arrange(desc(count))  
#view tail numbers for the most active planes  
head(flights\_2013)

# A tibble: 6 × 2  
 tailnum count  
 <chr> <int>  
1 N725MQ 546  
2 N722MQ 487  
3 N723MQ 480  
4 N711MQ 467  
5 N713MQ 455  
6 N258JB 422

#N725MQ made the most flights  
  
#Finding plane with the most flights in 2015 with more than 50 seats  
#Joining the flights and planes tables  
Flights\_Planes <-left\_join(x = flights, y = planes, by = "tailnum")  
#viewing to ensure columns added  
head(Flights\_Planes)

# A tibble: 6 × 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  
# 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>

#Creating a join table to flights\_2013  
Flights\_Planes\_2013 <-left\_join(x = flights\_2013, y = planes, by = "tailnum")  
#viewing to ensure columns added  
head(Flights\_Planes)

# A tibble: 6 × 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  
# 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>

#Finding plane with most number of flights in 2013 and more than 50 seats  
  
#Finding the plane with >50 seats  
#utiliseb the sorted and filtered dataframe created perviously  
Seats\_50 <- Flights\_Planes\_2013 %>%   
 #Filter for planes with >50 seats  
 filter(seats >50 & !is.na(seats))   
#view tail numbers for the most active planes with >50 seats  
head(Seats\_50)

# A tibble: 6 × 10  
 tailnum count year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
1 N328AA 389 1986 Fixed wing … BOEING 767-… 2 255 NA Turbo…  
2 N327AA 385 1986 Fixed wing … BOEING 767-… 2 255 NA Turbo…  
3 N338AA 385 1987 Fixed wing … BOEING 767-… 2 255 NA Turbo…  
4 N335AA 384 1987 Fixed wing … BOEING 767-… 2 255 NA Turbo…  
5 N323AA 356 1986 Fixed wing … BOEING 767-… 2 255 NA Turbo…  
6 N319AA 353 1985 Fixed wing … BOEING 767-… 2 255 NA Turbo…

#Select the most active  
Most\_Act <- Seats\_50 %>%   
 #filter for count=max  
 filter(count == max(count))  
  
Most\_Act

# A tibble: 1 × 10  
 tailnum count year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
1 N328AA 389 1986 Fixed wing … BOEING 767-… 2 255 NA Turbo…

#we know that plane:N328AA was the most active with more than 50 seats  
#Create a table of all 2013 flights by N328AA  
#Create a reference for the tail number pulled directly from the data frame for the plane  
PlaneRef <- Most\_Act[1,1]  
#View tailnum for the plane  
PlaneRef

# A tibble: 1 × 1  
 tailnum  
 <chr>   
1 N328AA

#show all flights in 2013 from the most active plane with >50 seats  
Flights\_Planes %>%   
 #filter for 2013 and N328AAflights only, also excluding cancelled flights  
 filter(tailnum == as.character(PlaneRef) & year.x == 2013 & !is.na(dep\_time))

# A tibble: 389 × 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 1026 1030 -4 1351 1340  
 2 2013 1 2 1038 1030 8 1347 1340  
 3 2013 1 3 1152 1200 -8 1446 1510  
 4 2013 1 4 858 900 -2 1210 1220  
 5 2013 1 5 851 900 -9 1206 1220  
 6 2013 1 6 1027 1030 -3 1335 1340  
 7 2013 1 7 724 730 -6 1008 1100  
 8 2013 1 7 2134 2135 -1 19 50  
 9 2013 1 8 2130 2135 -5 114 50  
10 2013 1 9 1701 1645 16 1958 2005  
# ℹ 379 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>

#Table shows the 389 flights in 2013 as expected by the count shown earlier

## 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`?

#Glipmse weather data set to view colums and data types  
dplyr::glimpse(weather)

Rows: 26,115  
Columns: 15  
$ origin <chr> "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EWR", "EW…  
$ year <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013,…  
$ month <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
$ day <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,…  
$ hour <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 14, 15, 16, 17, 18, …  
$ temp <dbl> 39.02, 39.02, 39.02, 39.92, 39.02, 37.94, 39.02, 39.92, 39.…  
$ dewp <dbl> 26.06, 26.96, 28.04, 28.04, 28.04, 28.04, 28.04, 28.04, 28.…  
$ humid <dbl> 59.37, 61.63, 64.43, 62.21, 64.43, 67.21, 64.43, 62.21, 62.…  
$ wind\_dir <dbl> 270, 250, 240, 250, 260, 240, 240, 250, 260, 260, 260, 330,…  
$ wind\_speed <dbl> 10.35702, 8.05546, 11.50780, 12.65858, 12.65858, 11.50780, …  
$ wind\_gust <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 20.…  
$ precip <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…  
$ pressure <dbl> 1012.0, 1012.3, 1012.5, 1012.2, 1011.9, 1012.4, 1012.2, 101…  
$ visib <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10,…  
$ time\_hour <dttm> 2013-01-01 01:00:00, 2013-01-01 02:00:00, 2013-01-01 03:00…

#View distribution of temp  
skimr::skim(weather)

Data summary

|  |  |
| --- | --- |
| Name | weather |
| Number of rows | 26115 |
| Number of columns | 15 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 1 |
| numeric | 13 |
| POSIXct | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| origin | 0 | 1 | 3 | 3 | 0 | 3 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| year | 0 | 1.00 | 2013.00 | 0.00 | 2013.00 | 2013.00 | 2013.00 | 2013.00 | 2013.00 | ▁▁▇▁▁ |
| month | 0 | 1.00 | 6.50 | 3.44 | 1.00 | 4.00 | 7.00 | 9.00 | 12.00 | ▇▆▆▆▇ |
| day | 0 | 1.00 | 15.68 | 8.76 | 1.00 | 8.00 | 16.00 | 23.00 | 31.00 | ▇▇▇▇▆ |
| hour | 0 | 1.00 | 11.49 | 6.91 | 0.00 | 6.00 | 11.00 | 17.00 | 23.00 | ▇▇▆▇▇ |
| temp | 1 | 1.00 | 55.26 | 17.79 | 10.94 | 39.92 | 55.40 | 69.98 | 100.04 | ▂▇▇▇▁ |
| dewp | 1 | 1.00 | 41.44 | 19.39 | -9.94 | 26.06 | 42.08 | 57.92 | 78.08 | ▁▆▇▇▆ |
| humid | 1 | 1.00 | 62.53 | 19.40 | 12.74 | 47.05 | 61.79 | 78.79 | 100.00 | ▁▆▇▇▆ |
| wind\_dir | 460 | 0.98 | 199.76 | 107.31 | 0.00 | 120.00 | 220.00 | 290.00 | 360.00 | ▆▂▆▇▇ |
| wind\_speed | 4 | 1.00 | 10.52 | 8.54 | 0.00 | 6.90 | 10.36 | 13.81 | 1048.36 | ▇▁▁▁▁ |
| wind\_gust | 20778 | 0.20 | 25.49 | 5.95 | 16.11 | 20.71 | 24.17 | 28.77 | 66.75 | ▇▅▁▁▁ |
| precip | 0 | 1.00 | 0.00 | 0.03 | 0.00 | 0.00 | 0.00 | 0.00 | 1.21 | ▇▁▁▁▁ |
| pressure | 2729 | 0.90 | 1017.90 | 7.42 | 983.80 | 1012.90 | 1017.60 | 1023.00 | 1042.10 | ▁▁▇▆▁ |
| visib | 0 | 1.00 | 9.26 | 2.06 | 0.00 | 10.00 | 10.00 | 10.00 | 10.00 | ▁▁▁▁▇ |

**Variable type: POSIXct**

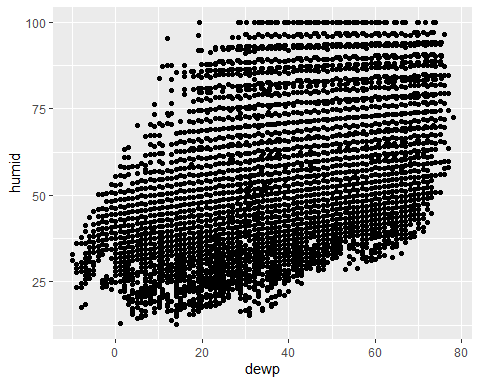
| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| time\_hour | 0 | 1 | 2013-01-01 01:00:00 | 2013-12-30 18:00:00 | 2013-07-01 14:00:00 | 8714 |

#Utilise the skim output to identify distribution of temps  
#Mean temerature= 55.3 degrees with standard deviation of 17.78 degrees  
  
#Identifying outliers in wind\_speed  
#Skim output shows clear outliers are present based on p100 being>>>2SD from mean  
  
weather %>%  
 #Remove any rows with NA windspeed  
 filter (!is.na(wind\_speed)) %>%   
 #Filter for wind speed values +/- 2 SD from mean Wind\_Speed  
 #Assuming a normal distribution  
 filter(wind\_speed <= (mean(wind\_speed) - 2\*sd(wind\_speed))| wind\_speed >= (mean(wind\_speed) + 2\*sd(wind\_speed))) %>%   
 #arrange largest to smallest  
 arrange(desc(wind\_speed))

# A tibble: 219 × 15  
 origin year month day hour temp dewp humid wind\_dir wind\_speed  
 <chr> <int> <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl>  
 1 EWR 2013 2 12 3 39.0 27.0 61.6 260 1048.   
 2 EWR 2013 1 31 6 57.2 53.6 87.7 270 42.6  
 3 JFK 2013 1 31 4 53.6 53.1 100 200 42.6  
 4 EWR 2013 1 31 4 60.8 59 93.8 230 40.3  
 5 LGA 2013 1 31 4 59 55.4 93.7 230 40.3  
 6 EWR 2013 1 31 8 46.0 30.0 53.3 270 39.1  
 7 JFK 2013 3 6 14 41 28.9 61.9 50 38.0  
 8 JFK 2013 1 31 3 53.1 52.0 100 180 36.8  
 9 JFK 2013 1 31 7 51.8 46.4 81.7 270 36.8  
10 JFK 2013 11 24 10 28.0 -0.04 29.2 310 36.8  
# ℹ 209 more rows  
# ℹ 5 more variables: wind\_gust <dbl>, precip <dbl>, pressure <dbl>,  
# visib <dbl>, time\_hour <dttm>

#the above shoes the windspeeds >2SD from the mean in the data which we can conclude as outliers.  
  
#Identifying relationship between dewp and humid  
#Utilise a plot to see clearly the relationship  
ggplot(weather,aes(x=dewp,y=humid))+  
 geom\_point()

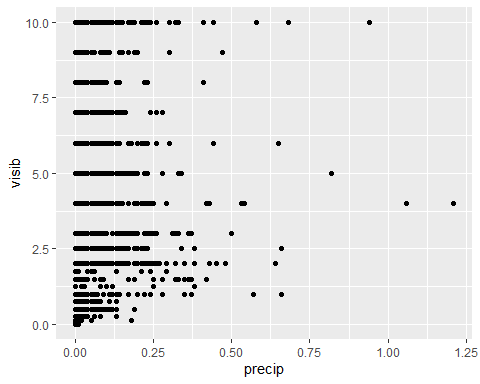
Warning: Removed 1 rows containing missing values (`geom\_point()`).



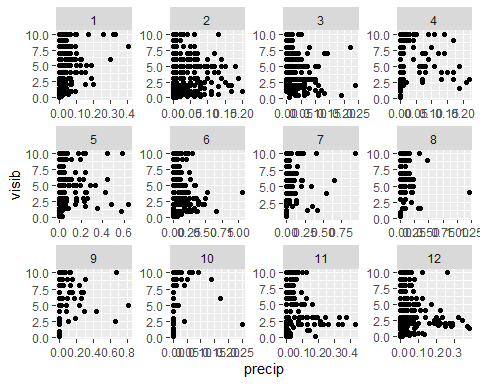
#lets confirm by calculating correlation between the 2 variables  
weather %>%   
 #Remove NA's  
 filter(!is.na(dewp) & !is.na(humid)) %>%   
 #Calculate correlation  
 summarize(correlation=cor(dewp,humid))

# A tibble: 1 × 1  
 correlation  
 <dbl>  
1 0.512

#Correlation of 0.5 confirms the obervation above  
  
#psotive correlation between dewp and humidity visible from chart, as humid increases, so does dewp  
  
#Identifying relationship between precip and visib  
#Utilise a plot to see clearly the relationship  
ggplot(weather,aes(x=precip,y=visib))+  
 geom\_point()



#unclear from the above  
weather %>%   
 group\_by(month) %>%   
 ggplot(aes(x=precip,y=visib))+  
 facet\_wrap(~month, scales = "free")+  
 geom\_point()



#appears to be a varying relationship by month  
#Calculate correlation by month  
  
weather %>%   
 #Remove NA's  
 filter(!is.na(dewp) & !is.na(humid)) %>%   
 #group\_by month  
 group\_by(month) %>%   
 #Calculate correlation  
 summarize(correlation=cor(dewp,humid))

# A tibble: 12 × 2  
 month correlation  
 <int> <dbl>  
 1 1 0.793  
 2 2 0.824  
 3 3 0.786  
 4 4 0.787  
 5 5 0.651  
 6 6 0.581  
 7 7 0.535  
 8 8 0.812  
 9 9 0.665  
10 10 0.711  
11 11 0.815  
12 12 0.735

#Dewp and humid appear to be positvely correlated, with the strength of this correlaiton varying by month

## 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.)

#Planes with missing date of manafacture  
#join the planes and flights table  
Flights\_Planes <-left\_join(x = flights, y = planes, by = "tailnum")  
#Glimpse the planes table   
dplyr::glimpse(planes)

Rows: 3,322  
Columns: 9  
$ tailnum <chr> "N10156", "N102UW", "N103US", "N104UW", "N10575", "N105UW…  
$ year <int> 2004, 1998, 1999, 1999, 2002, 1999, 1999, 1999, 1999, 199…  
$ type <chr> "Fixed wing multi engine", "Fixed wing multi engine", "Fi…  
$ manufacturer <chr> "EMBRAER", "AIRBUS INDUSTRIE", "AIRBUS INDUSTRIE", "AIRBU…  
$ model <chr> "EMB-145XR", "A320-214", "A320-214", "A320-214", "EMB-145…  
$ engines <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, …  
$ seats <int> 55, 182, 182, 182, 55, 182, 182, 182, 182, 182, 55, 55, 5…  
$ speed <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N…  
$ engine <chr> "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turbo-fan", "Turb…

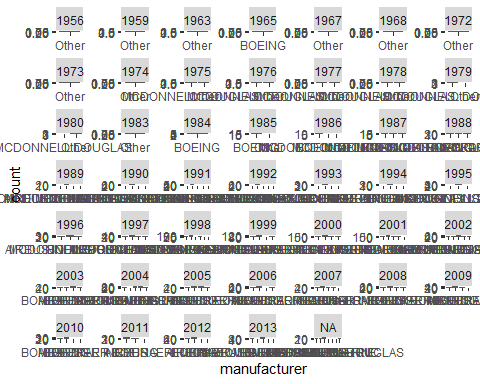
#View planes table  
#view(planes)  
#Missing date of manafacture from planes table  
planes %>%   
 #Filter for entries with no manafacturer  
 #assuming year is the year of manafacture  
 filter(is.na(year)) %>%   
 #Count the number of planes with no manafacturer  
 summarise(count =n())

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

# 70 planes have no date of manafacture  
  
#Fnding the most common manafacturers  
  
Pop\_Man <- planes %>%   
 #Group by manafacturer  
 group\_by(manufacturer) %>%   
 #Count the number of planes for each manafacturer  
 summarise(Count = n()) %>%   
 #Order most to least  
 arrange(desc(Count))  
#selecting the top 5  
head(Pop\_Man,5)

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

#Manafacturer distribition over time  
#collapsing manafacturers with <100 planes into other catagory  
#assmining the 2 similarly named MCDONNELL manafacturers  
#Define list of manafactures to keep seperate  
M <- c('BOEING', 'AIRBUS INDUSTRIE', 'BOMBARDIER INC', 'AIRBUS', 'EMBRAER','MCDONNELL DOUGLAS','MCDONNELL DOUGLAS AIRCRAFT CO')  
  
simplified\_mana <-planes %>%   
 #replace small manafacturer names with other  
 mutate(manufacturer= case\_when(manufacturer %in% M ~ manufacturer, TRUE ~ 'Other')) %>%   
 mutate(count=n())  
#plot manafacturer count over time  
  
simplified\_mana %>%   
 #Group by manafacturer  
 group\_by(manufacturer,year) %>%   
 #plot charts  
 ggplot(aes(x=manufacturer))+  
 facet\_wrap(~year, scales = "free")+  
 geom\_bar()



# the plots showcase how much the manafacturer split has changed over time, with significant differences visible year to year in recent times.  
#This may be due to the manafacturing/lifespan cycles being longdated, with certain manafactures producing more inventory in some years Vs others.

## 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?

#join the planes and flights table  
Flights\_Planes <-left\_join(x = flights, y = planes, by = "tailnum")  
  
#oldest plane to fly from NY  
oldest <- Flights\_Planes %>%   
 #filter for 2013  
 filter(year.x== 2013 & !is.na(year.y)) %>%   
 #Sort max to min on year/y  
 arrange(year.y)  
#View thetop 5 entries  
head(oldest)

# A tibble: 6 × 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  
# 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>

#oldest plane is N381AA  
#Oldest plane and its flights in 2013  
oldest %>%  
 filter(year.y == min(year.y))

# A tibble: 22 × 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 12 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>

#Planes that flew from NY that are in the planes table, implies tailnumbers that are in both tables.  
#manafacturer field is fully populated in lanes, implies use for count basis  
  
Flights\_Planes %>%  
 #select manafacturecolumn from thejoined table  
 select(manufacturer,tailnum) %>%   
 #filter out NA values  
 filter(!is.na(manufacturer)) %>%   
 #Count distinct entries in the right join  
 summarise(count= n\_distinct(tailnum))

# A tibble: 1 × 1  
 count  
 <int>  
1 3322

#implies 3322 planes from the planes table flew from NYC

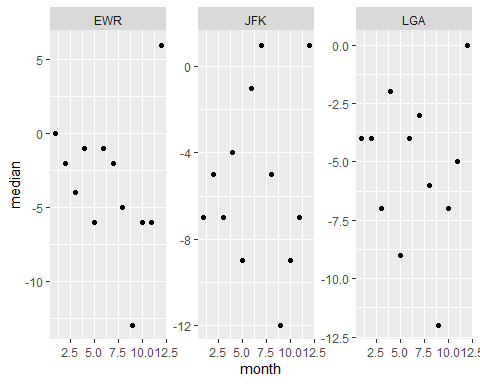
## Problem 7: Use the nycflights13 to answer the following questions:

- 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.

#glimpse(flights)  
#median arrival delay by airport  
# arrival delay & origin airport assumed as variables from questions  
#assumed month by month basis means across the same month in all years.  
  
medians <- flights %>%   
 #Groupby airport & month  
 group\_by(origin,month) %>%   
 #Filter out NA's in the arr\_delay field  
 filter(!is.na(arr\_delay)) %>%   
 #summarize to calculate the median for each month  
 summarise(median= median(arr\_delay)) %>%   
 #order by airport and by month  
 arrange(origin,month)

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

#plot charts  
  
#plot charts by origin, dot plot  
ggplot(medians, aes(x=month,y=median))+  
 facet\_wrap(~origin, scales = "free")+  
 geom\_point()



## 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.

#glimpse airlines table  
#identify the join column  
glimpse(airlines)

Rows: 16  
Columns: 2  
$ carrier <chr> "9E", "AA", "AS", "B6", "DL", "EV", "F9", "FL", "HA", "MQ", "O…  
$ name <chr> "Endeavor Air Inc.", "American Airlines Inc.", "Alaska Airline…

glimpse(flights)

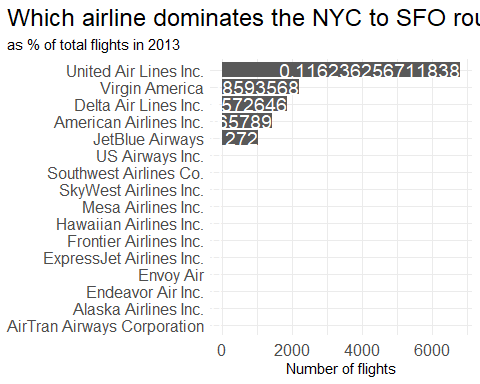
Rows: 336,776  
Columns: 19  
$ year <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2…  
$ month <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
$ day <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1…  
$ dep\_time <int> 517, 533, 542, 544, 554, 554, 555, 557,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

#join the 2 tables  
fly\_into\_sfo <- left\_join(x=flights,y=airlines, by = "carrier") %>%   
 #add column that =1 if dest=SFO, 0 otherwise  
 mutate(Fly\_SFO = case\_when(dest == 'SFO' ~ 1, TRUE ~ 0)) %>%   
 #Group by airline name  
 group\_by(name) %>%   
 #Summarize to find the count of SFO and % to SFO for each airline name  
 summarise(count= sum(Fly\_SFO),  
 percent = sum(Fly\_SFO)/length(Fly\_SFO))  
  
#glimpse new dataframe  
glimpse(fly\_into\_sfo)

Rows: 16  
Columns: 3  
$ name <chr> "AirTran Airways Corporation", "Alaska Airlines Inc.", "Americ…  
$ count <dbl> 0, 0, 1422, 1858, 0, 0, 0, 0, 0, 1035, 0, 0, 0, 0, 6819, 2197  
$ percent <dbl> 0.00000000, 0.00000000, 0.04344771, 0.03861983, 0.00000000, 0.…

Plotting the new dataframe with given formats:

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))

Explanation for how to produce the plot below:

* Step 1: group data by airport, origin, carrier, and ,month.
* Step2: filter for any NA’s across any of the above fields. Filter for WER& JFK airports only.Filter out for any airlines with a count =0 so that only the airlines with cancellations in SFO populate (avoid blank graphs)
* Step 3: plot using a ggplot 2 and geom\_bar() bar chart to show counts, and utilize facet wrap by carrier and origin to populate the charts as below



## 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.

#initially skim the dataset to get an idea of trends and variables  
skim(age\_gaps)

Data summary

|  |  |
| --- | --- |
| Name | age\_gaps |
| Number of rows | 1155 |
| Number of columns | 13 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 6 |
| Date | 2 |
| numeric | 5 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| movie\_name | 0 | 1 | 2 | 43 | 0 | 830 | 0 |
| director | 0 | 1 | 3 | 31 | 0 | 510 | 0 |
| actor\_1\_name | 0 | 1 | 6 | 22 | 0 | 567 | 0 |
| actor\_2\_name | 0 | 1 | 7 | 27 | 0 | 647 | 0 |
| character\_1\_gender | 0 | 1 | 3 | 5 | 0 | 2 | 0 |
| character\_2\_gender | 0 | 1 | 3 | 5 | 0 | 2 | 0 |

**Variable type: Date**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| actor\_1\_birthdate | 0 | 1 | 1889-04-16 | 1996-06-01 | 1964-10-03 | 562 |
| actor\_2\_birthdate | 0 | 1 | 1906-10-06 | 1996-11-11 | 1974-07-30 | 640 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| release\_year | 0 | 1 | 2000.80 | 16.37 | 1935 | 1997 | 2004 | 2012 | 2022 | ▁▁▁▆▇ |
| age\_difference | 0 | 1 | 10.42 | 8.51 | 0 | 4 | 8 | 15 | 52 | ▇▃▂▁▁ |
| couple\_number | 0 | 1 | 1.40 | 0.75 | 1 | 1 | 1 | 2 | 7 | ▇▁▁▁▁ |
| actor\_1\_age | 0 | 1 | 40.64 | 10.42 | 18 | 33 | 39 | 47 | 81 | ▂▇▅▂▁ |
| actor\_2\_age | 0 | 1 | 30.21 | 7.50 | 17 | 25 | 29 | 34 | 68 | ▇▇▂▁▁ |

#Calculate number of releases per year  
  
age\_gaps %>%   
 #Filter out any NA's  
 filter(!is.na(release\_year)) %>%   
 #Group by release year  
 group\_by(release\_year) %>%   
 summarize(count=n()) %>%   
 #Arrange by release year  
 arrange(release\_year)

# A tibble: 82 × 2  
 release\_year count  
 <dbl> <int>  
 1 1935 2  
 2 1936 1  
 3 1937 3  
 4 1939 1  
 5 1940 3  
 6 1942 2  
 7 1944 1  
 8 1946 1  
 9 1947 1  
10 1948 4  
# ℹ 72 more rows

#Check if actor age 1 or 2 is <17 absed on age field  
age\_gaps %>%   
 #Filter condition if 1 of the 2 ages is <17  
 filter(actor\_1\_age<17 | actor\_2\_age<17)

# A tibble: 0 × 13  
# ℹ 13 variables: movie\_name <chr>, release\_year <dbl>, director <chr>,  
# age\_difference <dbl>, couple\_number <dbl>, actor\_1\_name <chr>,  
# actor\_2\_name <chr>, character\_1\_gender <chr>, character\_2\_gender <chr>,  
# actor\_1\_birthdate <date>, actor\_2\_birthdate <date>, actor\_1\_age <dbl>,  
# actor\_2\_age <dbl>

#None appear to be below age 17  
  
#check again using release year-birth year  
age\_gaps %>%   
 #Filter condition if 1 of the 2 ages is <17  
 filter((release\_year- year(actor\_1\_birthdate))<17 | (release\_year- year(actor\_1\_birthdate))<17)

# A tibble: 0 × 13  
# ℹ 13 variables: movie\_name <chr>, release\_year <dbl>, director <chr>,  
# age\_difference <dbl>, couple\_number <dbl>, actor\_1\_name <chr>,  
# actor\_2\_name <chr>, character\_1\_gender <chr>, character\_2\_gender <chr>,  
# actor\_1\_birthdate <date>, actor\_2\_birthdate <date>, actor\_1\_age <dbl>,  
# actor\_2\_age <dbl>

#None appear to be below age 17 again  
  
#Count distinct number of couples by movie and order by most to elast couples  
  
top\_5\_couples <- age\_gaps %>%   
 #Filter for NA couple numbers  
 filter(!is.na(couple\_number)) %>%   
 #group by movie name  
 group\_by(movie\_name) %>%   
 #Summarize a distinct count  
 summarise(CoupleCount= n\_distinct(couple\_number)) %>%   
 #Order my most to least couples  
 arrange(desc(CoupleCount))  
  
#view top 5  
head(top\_5\_couples,5)

# A tibble: 5 × 2  
 movie\_name CoupleCount  
 <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

#loveactually has the most couples with 7!

# Details

* Who did you collaborate with: NA
* Approximately how much time did you spend on this problem set: 4 hours - mainly trying to get github to work!
* What, if anything, gave you the most trouble: replacing names in the manufacturers field in a coherent and scaleable way, think i found a solution although i didn’t quite have time to update some of the code above. (i.e- i would replace the list i typed out with the top seven manufacturers with a vector selection the top X distinct manufactures base don count)