Homework 1

Jenna Thomas

5/14/23

<https://github.com/jennalit/dsb2023_jt/blob/7560f3a57699ffd180a81e969c193944cdeeed7d/homework1.qmd>

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…

Data summary

|  |  |
| --- | --- |
| Name | flights |
| Number of rows | 336776 |
| Number of columns | 19 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 4 |
| numeric | 14 |
| POSIXct | 1 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| carrier | 0 | 1.00 | 2 | 2 | 0 | 16 | 0 |
| tailnum | 2512 | 0.99 | 5 | 6 | 0 | 4043 | 0 |
| origin | 0 | 1.00 | 3 | 3 | 0 | 3 | 0 |
| dest | 0 | 1.00 | 3 | 3 | 0 | 105 | 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 | 2013 | 2013 | 2013 | 2013 | ▁▁▇▁▁ |
| month | 0 | 1.00 | 6.55 | 3.41 | 1 | 4 | 7 | 10 | 12 | ▇▆▆▆▇ |
| day | 0 | 1.00 | 15.71 | 8.77 | 1 | 8 | 16 | 23 | 31 | ▇▇▇▇▆ |
| dep\_time | 8255 | 0.98 | 1349.11 | 488.28 | 1 | 907 | 1401 | 1744 | 2400 | ▁▇▆▇▃ |
| sched\_dep\_time | 0 | 1.00 | 1344.25 | 467.34 | 106 | 906 | 1359 | 1729 | 2359 | ▁▇▇▇▃ |
| dep\_delay | 8255 | 0.98 | 12.64 | 40.21 | -43 | -5 | -2 | 11 | 1301 | ▇▁▁▁▁ |
| arr\_time | 8713 | 0.97 | 1502.05 | 533.26 | 1 | 1104 | 1535 | 1940 | 2400 | ▁▃▇▇▇ |
| sched\_arr\_time | 0 | 1.00 | 1536.38 | 497.46 | 1 | 1124 | 1556 | 1945 | 2359 | ▁▃▇▇▇ |
| arr\_delay | 9430 | 0.97 | 6.90 | 44.63 | -86 | -17 | -5 | 14 | 1272 | ▇▁▁▁▁ |
| flight | 0 | 1.00 | 1971.92 | 1632.47 | 1 | 553 | 1496 | 3465 | 8500 | ▇▃▃▁▁ |
| air\_time | 9430 | 0.97 | 150.69 | 93.69 | 20 | 82 | 129 | 192 | 695 | ▇▂▂▁▁ |
| distance | 0 | 1.00 | 1039.91 | 733.23 | 17 | 502 | 872 | 1389 | 4983 | ▇▃▂▁▁ |
| hour | 0 | 1.00 | 13.18 | 4.66 | 1 | 9 | 13 | 17 | 23 | ▁▇▇▇▅ |
| minute | 0 | 1.00 | 26.23 | 19.30 | 0 | 8 | 29 | 44 | 59 | ▇▃▆▃▅ |

**Variable type: POSIXct**

| skim\_variable | n\_missing | complete\_rate | min | max | median | n\_unique |
| --- | --- | --- | --- | --- | --- | --- |
| time\_hour | 0 | 1 | 2013-01-01 05:00:00 | 2013-12-31 23:00:00 | 2013-07-03 10:00:00 | 6936 |

# 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 == "IAH" | dest == "HOU")

# 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(between(month,7,9))

# 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: 26 × 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 1357 1359 -2 1858 1654  
 3 2013 10 16 657 700 -3 1258 1056  
 4 2013 11 1 658 700 -2 1329 1015  
 5 2013 3 18 1844 1847 -3 39 2219  
 6 2013 4 17 1635 1640 -5 2049 1845  
 7 2013 4 18 558 600 -2 1149 850  
 8 2013 4 18 655 700 -5 1213 950  
 9 2013 5 22 1827 1830 -3 2217 2010  
10 2013 6 5 1604 1615 -11 2041 1840  
# ℹ 16 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: 1,819 × 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 2257 2000 177 45 2224  
 9 2013 1 4 1917 1700 137 2135 1950  
10 2013 1 4 2010 1745 145 2257 2120  
# ℹ 1,809 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))

flights %>%  
 # total flights that month  
 add\_count(month, name = "total\_flights") %>%   
 # filter to only cancelled flights  
 filter(is.na(dep\_time)) %>%   
 # cancelled flights that month  
 add\_count(month, name = "cancelled\_flights") %>%   
 # calculate proportion cancelled  
 mutate(prop\_cancelled = cancelled\_flights / total\_flights) %>%  
 # return just the month and proportion  
 group\_by(month, prop\_cancelled) %>%  
 summarise() %>%  
 # sort by proportion  
 arrange(prop\_cancelled)

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

# A tibble: 12 × 2  
# Groups: month [12]  
 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

# Lowest proportion in October and highest in February. The fall (August - November) seems to be the least likely for cancellation but I don't see much seasonality otherwise.

## 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.

view(planes)  
view(flights)  
  
# find the right plane   
x <- flights %>%  
 # this returns 'n' variable for number of flights by plane  
 count(tailnum, name = "num\_flights") %>%   
 # joining in order to filter by plane size  
 left\_join(planes, by = join\_by(tailnum)) %>%  
 filter(seats > 50) %>%  
 # tells you which plane had the most flights and saves that  
 slice(which.max(num\_flights))   
  
# go back to the original table to find the other flights for that plane  
flights %>%  
 filter(tailnum == x$tailnum)

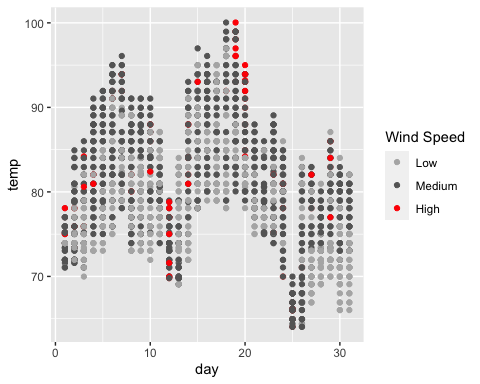
# A tibble: 393 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 1026 1030 -4 1351 1340  
 2 2013 1 2 1038 1030 8 1347 1340  
 3 2013 1 3 1152 1200 -8 1446 1510  
 4 2013 1 4 858 900 -2 1210 1220  
 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 383 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>

rm(x)

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

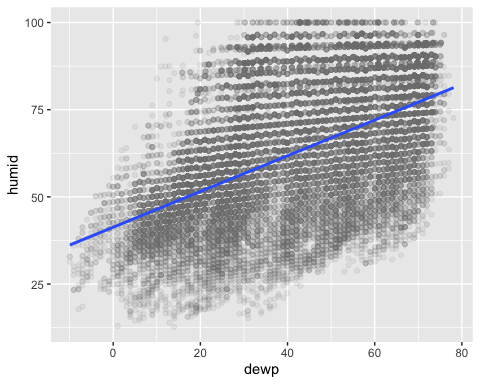
- What is the distribution of temperature (`temp`) in July 2013? Identify any important outliers in terms of the `wind\_speed` variable.  
- What is the relationship between `dewp` and `humid`?  
- What is the relationship between `precip` and `visib`?

view(weather)  
  
# distribution of temperature in July  
weather %>%  
 filter(month == 7) %>%  
 filter(!is.na(wind\_speed)) %>%  
 # I can't think of a better way to identify outliers that will be nice to visualise  
 # I don't feel that there are any significant wind speed outliers for July anyway  
 mutate(wind\_speed\_bin = cut(wind\_speed, 3)) %>%  
 # plot temperature by date and color by wind speed bins  
 ggplot(aes(x = day, y = temp, color = wind\_speed\_bin)) +  
 geom\_point() +  
 scale\_color\_manual(name = "Wind Speed", labels = c("Low", "Medium", "High"),  
 values = c("gray70","gray40","red"))



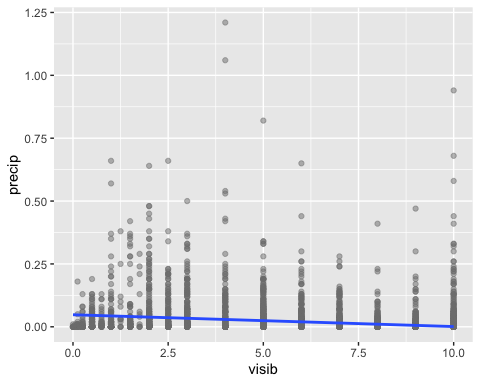
# relationship between dewpoint and humidity  
weather %>%  
 ggplot(aes(x = dewp, y = humid)) +  
 geom\_point(na.rm = TRUE, color = "gray50", alpha = 0.1) +  
 geom\_smooth(method = lm, na.rm = TRUE)

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



# relationship between precipitation and visibility  
weather %>%  
 ggplot(aes(y = precip, x = visib)) +  
 geom\_point(na.rm = TRUE, color = "gray50", alpha = 0.5) +  
 geom\_smooth(method = lm, na.rm = TRUE)

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



## 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.)

view(planes)  
  
planes %>%  
 count(is.na(year))

# A tibble: 2 × 2  
 `is.na(year)` n  
 <lgl> <int>  
1 FALSE 3252  
2 TRUE 70

# 70 planes  
  
planes %>%  
 # models = number of planes from each manufacturer  
 add\_count(manufacturer, name = "models") %>%   
 group\_by(manufacturer, models) %>%  
 summarise() %>%  
 # select top five manufacturers based on models  
 arrange(desc(models)) %>%  
 top\_n(models,n=5)

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

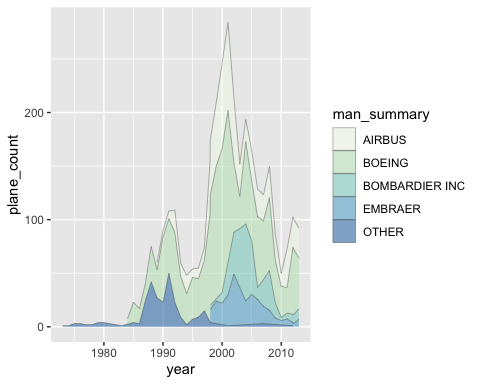
# A tibble: 35 × 2  
# Groups: manufacturer [35]  
 manufacturer models  
 <chr> <int>  
 1 BOEING 1630  
 2 AIRBUS INDUSTRIE 400  
 3 BOMBARDIER INC 368  
 4 AIRBUS 336  
 5 EMBRAER 299  
 6 MCDONNELL DOUGLAS 120  
 7 MCDONNELL DOUGLAS AIRCRAFT CO 103  
 8 MCDONNELL DOUGLAS CORPORATION 14  
 9 CANADAIR 9  
10 CESSNA 9  
# ℹ 25 more rows

planes %>%  
 # find total plane counts to filter out smaller manufacturers  
 add\_count(manufacturer, name = "tot\_plane\_count") %>%  
 # collapse smaller vendors into 'OTHER' and Airbus into one bucket  
 mutate(man\_summary = case\_when(tot\_plane\_count <200 ~ "OTHER",  
 grepl('AIRBUS\*', manufacturer) ~ "AIRBUS",  
 .default = manufacturer)) %>%  
 # find number of planes by year  
 group\_by(year, man\_summary) %>%  
 summarise(plane\_count = n()) %>%  
 # make a graph  
 ggplot(aes(x = year, y = plane\_count, fill = man\_summary)) +  
 geom\_area(color = "black", size = 0.1, alpha = 0.5) +  
 scale\_fill\_brewer(palette = "GnBu") +  
 # filtering to last 40 years for aesthetics  
 xlim(1973,2013)

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

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
ℹ Please use `linewidth` instead.

Warning: Removed 12 rows containing non-finite values (`stat\_align()`).



## 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?

flights %>%  
 # join the data with plane ages  
 left\_join(planes, by = join\_by(tailnum))%>%   
 # select only the oldest (noting that there are two 'year' variables now)  
 slice(which.min(year.y))

# A tibble: 1 × 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  
# 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>

# tailnum = N381AA  
  
# inner join to find shared   
inner\_join(flights,planes,by = join\_by(tailnum)) %>%  
 # find unique ids  
 group\_by(tailnum) %>%   
 summarise() %>%   
 count()

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

# 3322 planes

## 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.

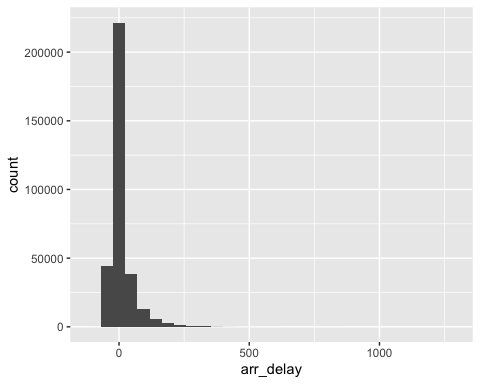
view(flights)  
  
flights %>%  
 group\_by(month) %>%  
 summarise(med\_arr\_delay = median(arr\_delay, na.rm = TRUE))

# A tibble: 12 × 2  
 month med\_arr\_delay  
 <int> <dbl>  
 1 1 -3  
 2 2 -3  
 3 3 -6  
 4 4 -2  
 5 5 -8  
 6 6 -2  
 7 7 -2  
 8 8 -5  
 9 9 -12  
10 10 -7  
11 11 -6  
12 12 2

# just getting a visual of where the bulk of data points lie to set y scale reasonably  
ggplot(flights, aes(x = arr\_delay)) +  
 geom\_histogram()

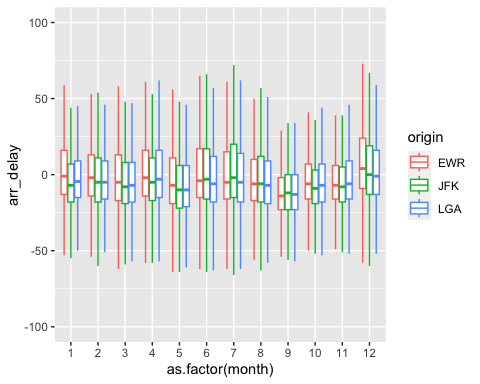
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

Warning: Removed 9430 rows containing non-finite values (`stat\_bin()`).



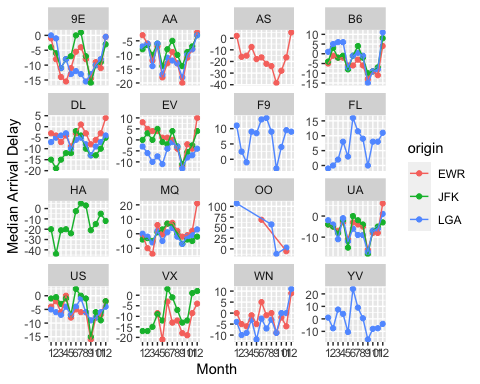
# misinterpreted the question but leaving it anyway  
flights %>%  
 ggplot(aes(x = as.factor(month), y = arr\_delay, color = origin)) +  
 geom\_boxplot(outlier.shape = NA) +  
 scale\_y\_continuous(limits = c(-100, 100))

Warning: Removed 23317 rows containing non-finite values (`stat\_boxplot()`).



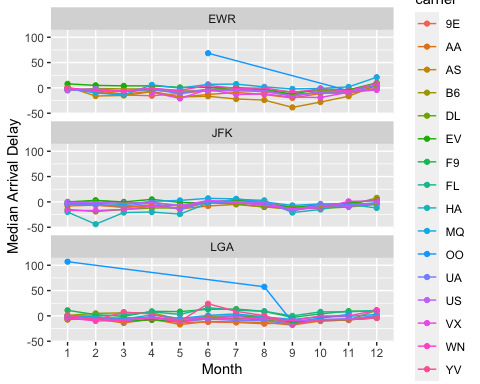
# carrier median arrival delay by month and origin   
flights %>%  
 group\_by(carrier, month, origin) %>%  
 summarise(med\_arr\_delay = median(arr\_delay, na.rm = TRUE)) %>%  
 ggplot(aes(x = as.factor(month), y = med\_arr\_delay, color = origin)) +  
 geom\_point() +   
 geom\_line(aes(x = month, y = med\_arr\_delay)) +  
 facet\_wrap(~carrier, scales = "free\_y") +  
 # theme(legend.position="none") +  
 xlab("Month") + ylab("Median Arrival Delay")

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



# origin median arrival delay by month and carrier  
flights %>%  
 group\_by(carrier, month, origin) %>%  
 summarise(med\_arr\_delay = median(arr\_delay, na.rm = TRUE)) %>%  
 ggplot(aes(x = as.factor(month), y = med\_arr\_delay, color = carrier)) +  
 geom\_point() +   
 geom\_line(aes(x = month, y = med\_arr\_delay)) +  
 facet\_wrap(~origin, ncol = 1) +  
 xlab("Month") + ylab("Median Arrival Delay")

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

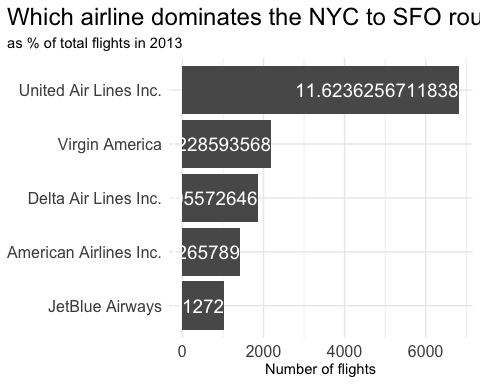


## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United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

view(airlines)  
  
fly\_into\_sfo <- flights %>%  
 # get carrier names  
 left\_join(airlines, by = join\_by(carrier)) %>%   
 # total number of flights for each carrier  
 add\_count(carrier, name = "total\_flights") %>%   
 # now filter by SFO and count flights  
 filter(dest == "SFO") %>%  
 add\_count(carrier, name = "sfo\_flights") %>%   
 # calculate percent sfo  
 mutate(percent\_sfo = sfo\_flights/total\_flights \*100) %>%  
 group\_by(name) %>%  
 # really need to figure out how to just return the shared variable  
 summarise(count = mean(sfo\_flights),percent = mean(percent\_sfo))

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



rm(list = ls())

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

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

I want you to think how we would organise our data manipulation to create the following plot. No need to write the code, just explain in words how you would go about it.

## Problem 10: On your own – Hollywood Age Gap

The website https://hollywoodagegap.com is a record of *THE AGE DIFFERENCE IN YEARS BETWEEN MOVIE LOVE INTERESTS*. This is an informational site showing the age gap between movie love interests and the data follows certain rules:

* The two (or more) actors play actual love interests (not just friends, coworkers, or some other non-romantic type of relationship)
* The youngest of the two actors is at least 17 years old
* No animated characters

The age gaps dataset includes “gender” columns, which always contain the values “man” or “woman”. These values appear to indicate how the characters in each film identify and some of these values do not match how the actor identifies. We apologize if any characters are misgendered in the data!

The following is a data dictionary of the variables used

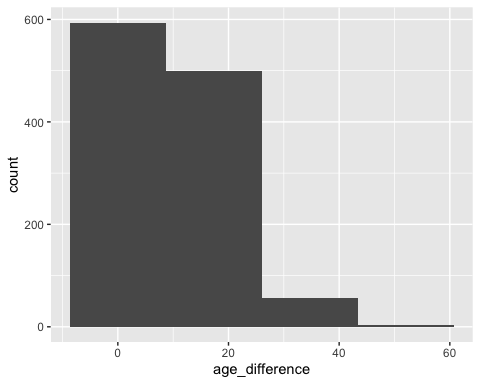
| variable | class | description |
| --- | --- | --- |
| movie\_name | character | Name of the film |
| release\_year | integer | Release year |
| director | character | Director of the film |
| age\_difference | integer | Age difference between the characters in whole years |
| couple\_number | integer | An identifier for the couple in case multiple couples are listed for this film |
| actor\_1\_name | character | The name of the older actor in this couple |
| actor\_2\_name | character | The name of the younger actor in this couple |
| character\_1\_gender | character | The gender of the older character, as identified by the person who submitted the data for this couple |
| character\_2\_gender | character | The gender of the younger character, as identified by the person who submitted the data for this couple |
| actor\_1\_birthdate | date | The birthdate of the older member of the couple |
| actor\_2\_birthdate | date | The birthdate of the younger member of the couple |
| actor\_1\_age | integer | The age of the older actor when the film was released |
| actor\_2\_age | integer | The age of the younger actor when the film was released |

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

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

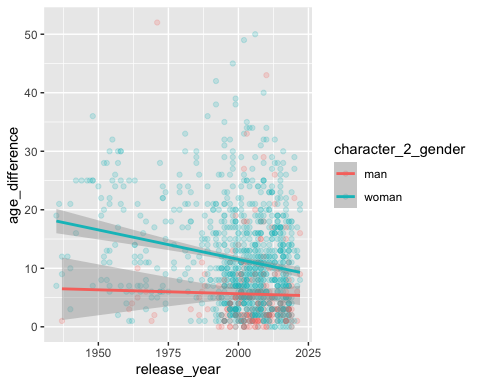
# what's the breakdown of age differences?  
age\_gaps %>%  
 ggplot(aes(x = age\_difference)) +  
 geom\_histogram() +   
 stat\_bin(bins = 4)

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

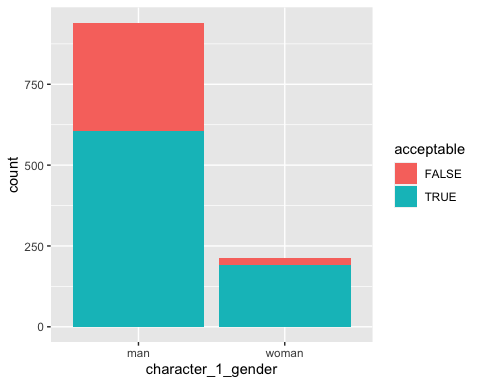


# vast majority are <25 years  
  
# let's look at how age difference has changed over time  
age\_gaps %>%  
 ggplot(aes(x = release\_year, y = age\_difference, color = character\_2\_gender)) +  
 geom\_point(alpha = 0.2) +  
 geom\_smooth(method = lm)

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



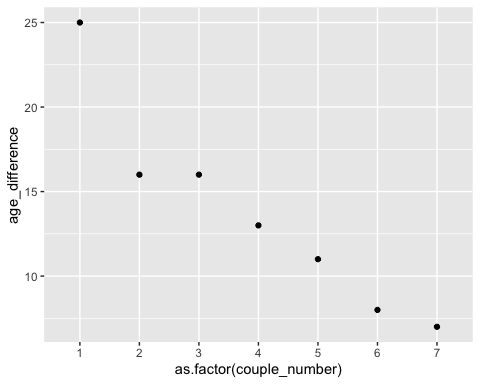
# I am a bit surprised that there isn't more correlation between release year and age\_diff  
# I am not surprised that female characters are more likely to be younger  
  
# now let's check the half plus seven rule  
halfrule <- age\_gaps %>%  
 mutate(minage = ((actor\_1\_age / 2) + 7),  
 maxage = ((actor\_1\_age - 7) \* 2),  
 acceptable = ifelse(actor\_2\_age > minage, TRUE, FALSE)) %>%  
 ggplot(aes(x = character\_1\_gender, fill = acceptable)) +  
 geom\_bar()  
halfrule



# if the older character is a woman, it's much more likely to be acceptable  
  
  
# which movie has the largest number of love interests and what are the gaps?  
age\_gaps %>%  
 filter(couple\_number == max(couple\_number))

# A tibble: 1 × 13  
 movie\_name release\_year director age\_difference couple\_number actor\_1\_name  
 <chr> <dbl> <chr> <dbl> <dbl> <chr>   
1 Love Actually 2003 Richard … 7 7 Martin Free…  
# ℹ 7 more variables: actor\_2\_name <chr>, character\_1\_gender <chr>,  
# character\_2\_gender <chr>, actor\_1\_birthdate <date>,  
# actor\_2\_birthdate <date>, actor\_1\_age <dbl>, actor\_2\_age <dbl>

age\_gaps %>%  
 filter(movie\_name == "Love Actually") %>%  
 ggplot(aes(x = as.factor(couple\_number), y = age\_difference)) +  
 geom\_point()

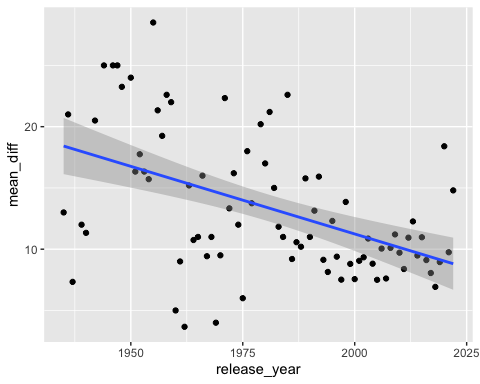


# the less significant the couple, the lower the age difference  
  
  
# percent of same gender  
age\_gaps %>%  
 mutate(same\_sex = case\_when(  
 character\_1\_gender == character\_2\_gender & character\_1\_gender == "woman" ~ "lesbian",  
 character\_1\_gender == character\_2\_gender & character\_1\_gender == "man" ~ "gay",  
 character\_1\_gender != character\_2\_gender ~ "hetero"  
 )) %>%  
 group\_by(same\_sex) %>%  
 summarise(no\_movies = n()) %>%  
 mutate(perc\_representation = no\_movies / sum(no\_movies)\*100)

# A tibble: 3 × 3  
 same\_sex no\_movies perc\_representation  
 <chr> <int> <dbl>  
1 gay 12 1.04   
2 hetero 1132 98.0   
3 lesbian 11 0.952

# change in mean age difference over time  
age\_gaps %>%  
 group\_by(release\_year) %>%  
 summarise(mean\_diff = mean(age\_difference), med\_diff = median(age\_difference)) %>%  
 ggplot(aes(x = release\_year, y = mean\_diff)) +  
 geom\_point() +  
 geom\_smooth(method = lm)

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



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?

# Details

* Who did you collaborate with: Colby Richardson, Shashvat Somany
* Approximately how much time did you spend on this problem set: ~3hrs
* What, if anything, gave you the most trouble: Understanding the logic of the actual question in order to come up with a solution (I had to think way too long about how to calculate a proportion)