Homerwork 1

Shashvat Somany

5/14/23

# Data Manipulation

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

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

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

# A tibble: 10,200 × 19  
 year month day dep\_time sched\_dep\_time dep\_delay arr\_time sched\_arr\_time  
 <int> <int> <int> <int> <int> <dbl> <int> <int>  
 1 2013 1 1 811 630 101 1047 830  
 2 2013 1 1 848 1835 853 1001 1950  
 3 2013 1 1 957 733 144 1056 853  
 4 2013 1 1 1114 900 134 1447 1222  
 5 2013 1 1 1505 1310 115 1638 1431  
 6 2013 1 1 1525 1340 105 1831 1626  
 7 2013 1 1 1549 1445 64 1912 1656  
 8 2013 1 1 1558 1359 119 1718 1515  
 9 2013 1 1 1732 1630 62 2028 1825  
10 2013 1 1 1803 1620 103 2008 1750  
# ℹ 10,190 more rows  
# ℹ 11 more variables: arr\_delay <dbl>, carrier <chr>, flight <int>,  
# tailnum <chr>, origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>,  
# hour <dbl>, minute <dbl>, time\_hour <dttm>

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

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

# A tibble: 139,504 × 19  
 year month day dep\_time sched\_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 == 7 | month == 8 | month ==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?  
 #First, I created new vraibles using the mutate function to find the flighst delayed and not delayed. I then divided the cancelled flights by the total  
cancprop <- flights %>%  
 mutate(  
 canc = is.na(dep\_time),  
 notc = !is.na(dep\_time),  
 canp = canc / (canc + notc)  
 )  
#To then figure out which months had a higher proportion of cancelled flights, I grouped it by month and then used the summarize function and arranged it by descending order  
cancprop %>%  
 group\_by(month) %>%   
 summarize(mcanp = mean(canp)) %>%   
 arrange(desc(mcanp))

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

## 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.

skim(planes)

Data summary

|  |  |
| --- | --- |
| Name | planes |
| Number of rows | 3322 |
| Number of columns | 9 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 5 |
| numeric | 4 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| tailnum | 0 | 1 | 5 | 6 | 0 | 3322 | 0 |
| type | 0 | 1 | 10 | 24 | 0 | 3 | 0 |
| manufacturer | 0 | 1 | 4 | 29 | 0 | 35 | 0 |
| model | 0 | 1 | 2 | 18 | 0 | 127 | 0 |
| engine | 0 | 1 | 7 | 13 | 0 | 6 | 0 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| year | 70 | 0.98 | 2000.48 | 7.19 | 1956 | 1997.0 | 2001 | 2005 | 2013 | ▁▁▂▇▇ |
| engines | 0 | 1.00 | 2.00 | 0.12 | 1 | 2.0 | 2 | 2 | 4 | ▁▇▁▁▁ |
| seats | 0 | 1.00 | 154.32 | 73.65 | 2 | 140.0 | 149 | 182 | 450 | ▂▇▃▁▁ |
| speed | 3299 | 0.01 | 236.78 | 149.76 | 90 | 107.5 | 162 | 432 | 432 | ▇▃▁▁▆ |

view(planes)  
view(flights)  
  
# find the right plane   
flights %>%  
 count(tailnum, sort=TRUE) %>%   
 # this returns 'n' variable for number of flights by plane  
 # left\_join(planes, by = join\_by(tailnum)) %>%  
 filter(n > 50) %>%  
 drop\_na(tailnum) %>%   
 slice(1)

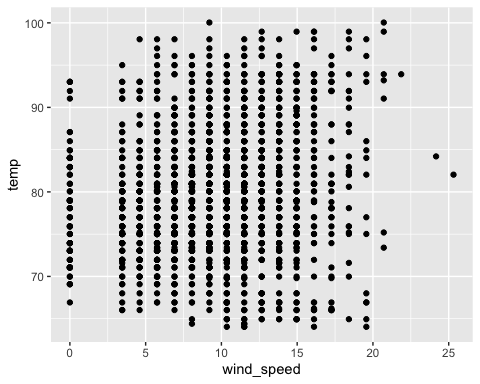
# A tibble: 1 × 2  
 tailnum n  
 <chr> <int>  
1 N725MQ 575

## 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`?

#distribution of temperature in July 2013  
julyweather <- weather %>%   
 filter(month == 7)  
  
#outliers in wind\_speed  
ggplot(julyweather, aes(x=wind\_speed, y=temp)) + geom\_point()

Warning: Removed 2 rows containing missing values (`geom\_point()`).



#temperature range in July  
julyweather %>%   
 arrange(temp) %>%   
 summarise(max(temp)-min(temp))

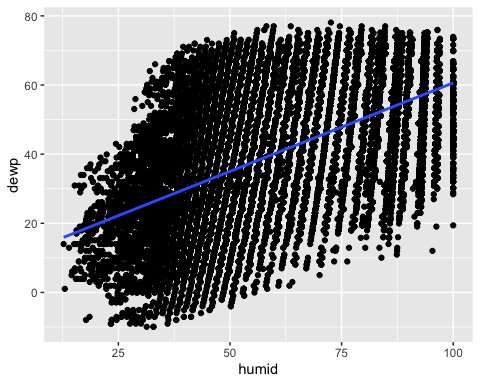
# A tibble: 1 × 1  
 `max(temp) - min(temp)`  
 <dbl>  
1 36

#relationship between humidity and dew  
ggplot(weather, aes(x =humid, y =dewp)) +   
 geom\_point() +  
 geom\_smooth(method=lm, se=FALSE)

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

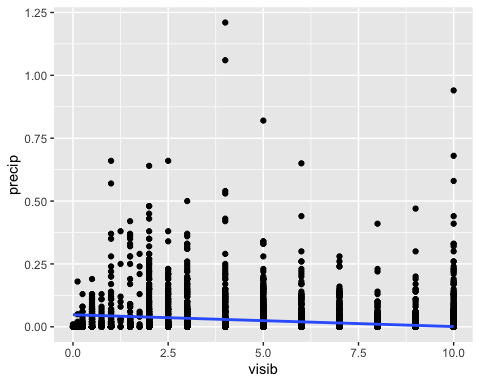
Warning: Removed 1 rows containing non-finite values (`stat\_smooth()`).

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



#relationship between visibility and precipitation  
ggplot(weather, aes(x =visib, y=precip)) +  
 geom\_point() +  
 geom\_smooth(method=lm, se=FALSE)

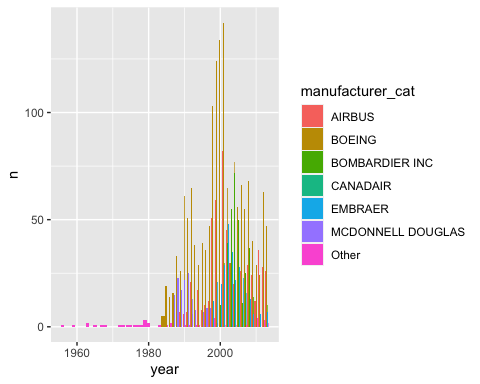
`geom\_smooth()` using formula = 'y ~ x'



## Problem 5: Use the flights and planes tables to answer the following questions:

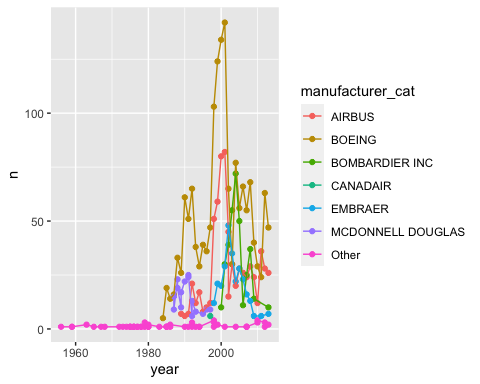
- How many planes have a missing date of manufacture?  
- What are the five most common manufacturers?  
- Has the distribution of manufacturer changed over time as reflected by the airplanes flying from NYC in 2013? (Hint: you may need to use case\_when() to recode the manufacturer name and collapse rare vendors into a category called Other.)

#How many planes have a missing date of manufacture?  
miss\_plane <- planes %>%   
summarize(sum(is.na(year) == TRUE))  
# The answer is 70  
  
#What are the five most common manufacturers?  
q5.2 <- planes %>%  
 add\_count(manufacturer) %>%   
 group\_by(manufacturer) %>%  
 summarise(models = mean(n)) %>%  
 arrange(desc(models)) %>%  
 top\_n(models,n=5)  
#The five most common manufacturers are Boeing, Airbus Industries, Bombardier, Airbus, and Embraer  
  
#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.)  
#First, I counted the number of planes by manufacturer and year  
planes1 <- planes %>%   
 count(manufacturer,year) %>%   
#I then used the mutate function to create a new category called "Other" which were all manufacturers that were not in the Top 5  
 mutate(manufacturer\_cat = ifelse(n >= 5, manufacturer, "Other")) %>%  
#I then filtered out all flights that did not have a date of manufacture  
 filter(is.na(year) == FALSE)  
#I then recategorised some manufacturer names   
  
planes1$manufacturer\_cat <- recode(planes1$manufacturer\_cat, "AIRBUS INDUSTRIE" = "AIRBUS",   
 "MCDONNELL DOUGLAS AIRCRAFT CO" = "MCDONNELL DOUGLAS",  
 "MCDONNELL DOUGLAS CORPORATION" = "MCDONNELL DOUGLAS",  
 "CANADAIR LTD" = "CANADAIR LTD")  
  
#I then just plotted the manufacturers over time and added a color function too   
planes1 %>% ggplot(aes(x = year, y = n)) + geom\_col(aes(fill = manufacturer\_cat), position = "dodge")



planes1 %>% ggplot(aes(x = year, y = n, group = manufacturer\_cat)) + geom\_point(aes(colour = manufacturer\_cat), position = "dodge") + geom\_line(aes(colour = manufacturer\_cat))

Warning: Width not defined  
ℹ Set with `position\_dodge(width = ...)`



## 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?  
planes %>%  
#First, I renamed year to year built  
 rename(year\_built = year) %>%  
#I then right joined, to merge all rows in the second data frame with the first  
 right\_join(flights, by = "tailnum") %>%  
 arrange(year\_built) %>%  
 select(tailnum, year\_built) %>%  
 head(1)

# A tibble: 1 × 2  
 tailnum year\_built  
 <chr> <int>  
1 N381AA 1956

#How many airplanes that flew from New York City are included in the planes table?  
flights%>%  
 group\_by(tailnum) %>%  
 slice(1L) %>%  
 inner\_join(planes, by = "tailnum")%>%  
 ungroup() %>%  
 summarize(num\_included = n(), missing\_date = sum(is.na(year.y)))

# A tibble: 1 × 2  
 num\_included missing\_date  
 <int> <int>  
1 3322 70

## 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?  
flights %>%   
group\_by(month, origin) %>%  
 summarize(median\_delay = median(arr\_delay, na.rm = TRUE))

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

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

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

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

# A tibble: 399 × 4  
# Groups: month, origin [36]  
 month origin carrier median\_delay  
 <int> <chr> <chr> <dbl>  
 1 1 EWR 9E -1  
 2 1 EWR AA -3  
 3 1 EWR AS 2  
 4 1 EWR B6 -5  
 5 1 EWR DL -3  
 6 1 EWR EV 8  
 7 1 EWR MQ -1  
 8 1 EWR UA -4  
 9 1 EWR US -4  
10 1 EWR WN 0  
# ℹ 389 more rows

#notworking

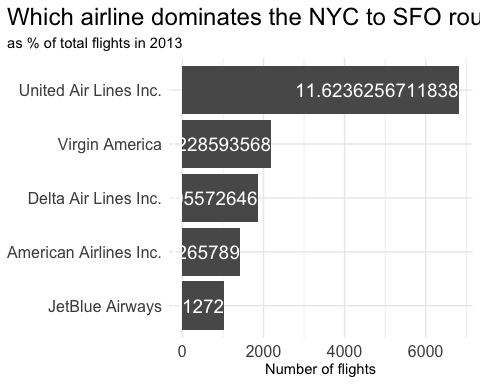
## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United 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 %>%  
 left\_join(airlines, by = join\_by(carrier)) %>% # get carrier names  
 add\_count(carrier) %>% # n = total flights by that carrier  
 filter(dest == "SFO") %>%  
 add\_count(carrier) %>% # nn = SFO flights  
 mutate(percent\_sfo = nn/n \*100) %>%  
 group\_by(name) %>%  
 summarise(count = mean(nn),percent = mean(percent\_sfo))

Storing counts in `nn`, as `n` already present in input  
ℹ Use `name = "new\_name"` to pick a new name.

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.

#



## 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

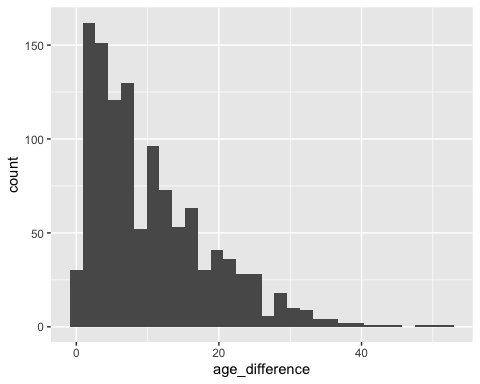
| variable | class | description |
| --- | --- | --- |
| movie\_name | character | Name of the film |
| release\_year | integer | Release year |
| director | character | Director of the film |
| age\_difference | integer | Age difference between the characters in whole years |
| couple\_number | integer | 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 and what is the typical age difference?  
age\_gaps %>%   
 ggplot()+  
 aes(x=age\_difference)+  
 geom\_histogram()

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



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

* How is age\_difference distributed? What’s the ‘typical’ age\_difference in movies?
* The half plus seven\ rule. Large age disparities in relationships carry certain stigmas. One popular rule of thumb is the [half-your-age-plus-seven](https://en.wikipedia.org/wiki/Age_disparity_in_sexual_relationships#The_.22half-your-age-plus-seven.22_rule) rule. This rule states you should never date anyone under half your age plus seven, establishing a minimum boundary on whom one can date. In order for a dating relationship to be acceptable under this rule, your partner’s age must be:

How frequently does this rule apply in this dataset?

* Which movie has the greatest number of love interests?
* Which actors/ actresses have the greatest number of love interests in this dataset?
* Is the mean/median age difference staying constant over the years (1935 - 2022)?
* How frequently does Hollywood depict same-gender love interests?

# Deliverables

There is a lot of explanatory text, comments, etc. You do not need these, so delete them and produce a stand-alone document that you could share with someone. Render the edited and completed Quarto Markdown (qmd) file as a Word document (use the “Render” button at the top of the script editor window) and upload it to Canvas. You must be commiting and pushing tour changes to your own Github repo as you go along.

# Details

* Who did you collaborate with: TYPE NAMES HERE
* Approximately how much time did you spend on this problem set: ANSWER HERE
* What, if anything, gave you the most trouble: ANSWER HERE

**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.