Homerwork 1

Gustavo Passebon Mendonca

5/14/23

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

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

# Flew to Houston (IAH or HOU)  
flights %>%   
 filter(dest %in% c("IAH","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(month %in% c(7,8,9))

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

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

# A tibble: 10,171 × 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 10,161 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) %>%   
 filter(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>

# What months had the highest and lowest % of cancelled flights?  
flights %>%   
 filter(is.na(dep\_time)==FALSE) %>%   
 count(month,sort=TRUE) %>%   
 mutate(percentTotal = n/sum(n)) %>%   
 arrange(percentTotal) %>%   
 slice(c(which.min(percentTotal),which.max(percentTotal)))

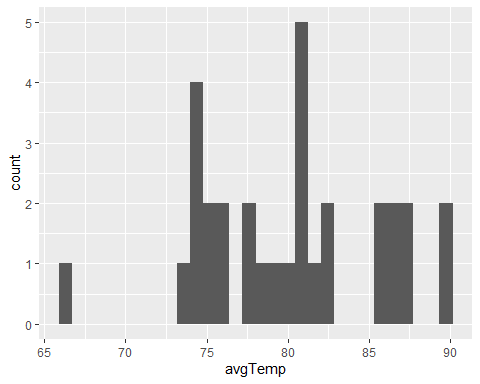
# A tibble: 2 × 3  
 month n percentTotal  
 <int> <int> <dbl>  
1 2 23690 0.0721  
2 8 28841 0.0878

#Creating data frame with the tailnums that flew the most  
mostFlights = flights %>%  
 filter(is.na(tailnum)==FALSE) %>%   
 group\_by(tailnum) %>%   
 count(tailnum,sort=TRUE)  
#Left joining with the plane characteristics  
left\_join(mostFlights,planes,by="tailnum")

# A tibble: 4,043 × 10  
# Groups: tailnum [4,043]  
 tailnum n year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 N725MQ 575 NA <NA> <NA> <NA> NA NA NA <NA>   
 2 N722MQ 513 NA <NA> <NA> <NA> NA NA NA <NA>   
 3 N723MQ 507 NA <NA> <NA> <NA> NA NA NA <NA>   
 4 N711MQ 486 1976 Fixed wing… GULFSTREAM … G115… 2 22 NA Turbo…  
 5 N713MQ 483 NA <NA> <NA> <NA> NA NA NA <NA>   
 6 N258JB 427 2006 Fixed wing… EMBRAER ERJ … 2 20 NA Turbo…  
 7 N298JB 407 2009 Fixed wing… EMBRAER ERJ … 2 20 NA Turbo…  
 8 N353JB 404 2012 Fixed wing… EMBRAER ERJ … 2 20 NA Turbo…  
 9 N351JB 402 2012 Fixed wing… EMBRAER ERJ … 2 20 NA Turbo…  
10 N735MQ 396 NA <NA> <NA> <NA> NA NA NA <NA>   
# ℹ 4,033 more rows

#Calculating average temperature  
average\_temperature=weather %>%   
 filter(is.na(temp)==FALSE) %>%   
 filter(month==07) %>%   
 filter(year==2013) %>%  
 group\_by(day) %>%   
 summarize(avgTemp=mean(temp))  
  
#Ploting a histogram of the average temperatures  
ggplot(average\_temperature, aes(x = avgTemp))+  
 geom\_histogram()

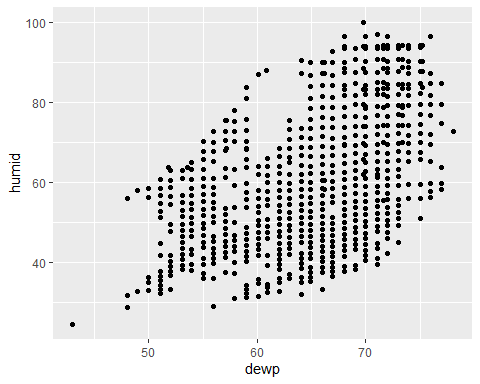
`stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



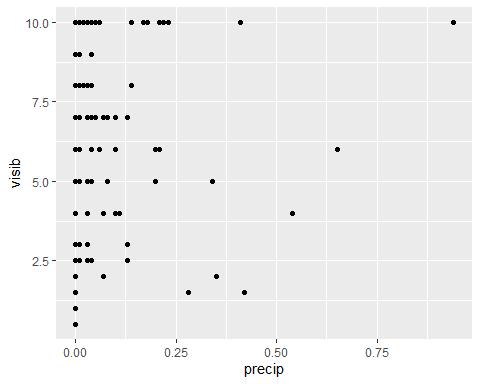
#Finding outliers in wind\_speed  
weather %>%   
 slice(c(which.min(wind\_speed),which.max(wind\_speed)))

# A tibble: 2 × 15  
 origin year month day hour temp dewp humid wind\_dir wind\_speed wind\_gust  
 <chr> <int> <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
1 EWR 2013 1 7 19 39.9 21.0 46.4 0 0 NA  
2 EWR 2013 2 12 3 39.0 27.0 61.6 260 1048. NA  
# ℹ 4 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,  
# time\_hour <dttm>

#Relationship between variables - setting up the dataframe  
weather\_stats = weather %>%   
 filter(is.na(temp)==FALSE) %>%   
 filter(month==07) %>%   
 filter(year==2013) %>%  
 group\_by(day)  
  
#Scatter plot between Dewp and Humid  
ggplot(weather\_stats, aes(x=dewp, y = humid))+  
 geom\_point()



#Scatter plot between Precip and Visib  
ggplot(weather\_stats, aes(x=precip, y = visib))+  
 geom\_point()



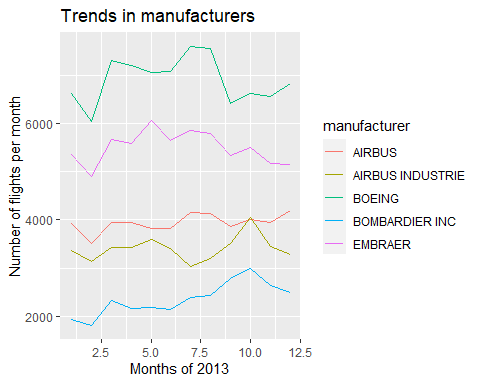
#Number of planes without manufacturing date = 70  
planes %>%   
 filter(is.na(year)==TRUE) %>%   
 count(year,sort=TRUE)

# A tibble: 1 × 2  
 year n  
 <int> <int>  
1 NA 70

#Five most common manufacturers  
common\_manufacturers = planes %>%   
 filter(is.na(manufacturer)==FALSE) %>%   
 group\_by(manufacturer) %>%   
 count(manufacturer,sort=TRUE)  
#Storing the top5 in a dataframe that will be used later  
most\_common = data.frame(common\_manufacturers[1:5,],stringsAsFactors = FALSE)  
most\_common

manufacturer n  
1 BOEING 1630  
2 AIRBUS INDUSTRIE 400  
3 BOMBARDIER INC 368  
4 AIRBUS 336  
5 EMBRAER 299

#Trends in manufacturers  
#Joining the manufacturing information into Flights df  
flights\_manuf = left\_join(flights,planes,by="tailnum")  
#Filtering by most common, grouping by month, counting manufacturers  
flights\_trends = flights\_manuf %>%  
 filter(is.na(manufacturer)==FALSE) %>%   
 group\_by(month) %>%   
 count(manufacturer,sort=TRUE) %>%   
 arrange(month) %>%   
 filter(manufacturer %in% most\_common[,1])  
#Ploting the results  
ggplot(flights\_trends, aes(x=month, y = n,color=manufacturer))+  
 geom\_line()+  
 labs(title="Trends in manufacturers",  
 x ="Months of 2013", y = "Number of flights per month")



#Oldest plane that flew from NYC in 2013  
flights\_manuf = left\_join(planes,flights,by="tailnum")  
flights\_manuf %>%   
 slice(which.min(year.x))

# A tibble: 1 × 27  
 tailnum year.x type manufacturer model engines seats speed engine year.y  
 <chr> <int> <chr> <chr> <chr> <int> <int> <int> <chr> <int>  
1 N381AA 1956 Fixed win… DOUGLAS DC-7… 4 102 232 Recip… 2013  
# ℹ 17 more variables: month <int>, day <int>, dep\_time <int>,  
# sched\_dep\_time <int>, dep\_delay <dbl>, arr\_time <int>,  
# sched\_arr\_time <int>, arr\_delay <dbl>, carrier <chr>, flight <int>,  
# origin <chr>, dest <chr>, air\_time <dbl>, distance <dbl>, hour <dbl>,  
# minute <dbl>, time\_hour <dttm>

#Number of airplanes that flew from NYC and are included in planes  
mostFlights = flights %>%   
 filter(is.na(tailnum)==FALSE) %>%   
 group\_by(tailnum) %>%   
 count(tailnum,sort=TRUE)  
  
sum(mostFlights$tailnum %in% planes$tailnum)

[1] 3322

#Median monthly arrival delay by airport  
flights %>%   
 filter(is.na(arr\_delay)==FALSE) %>%   
 group\_by(month,origin) %>%   
 summarize(medianDelay = median(arr\_delay))

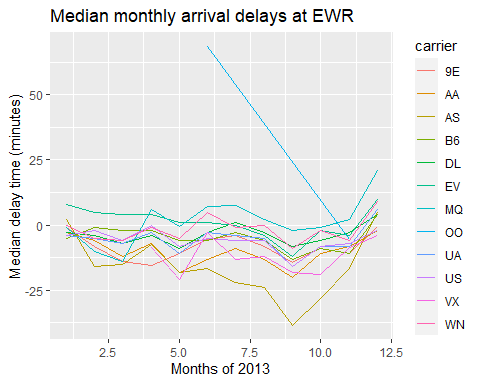
`summarise()` has grouped output by 'month'. You can override using the  
`.groups` argument.

# A tibble: 36 × 3  
# Groups: month [12]  
 month origin medianDelay  
 <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

#Median arrival delay by Airline and Airport  
median\_EWR = flights %>%   
 filter(is.na(arr\_delay)==FALSE) %>%  
 filter(origin=="EWR") %>%   
 group\_by(month,carrier) %>%   
 summarize(medianDelay = median(arr\_delay))

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

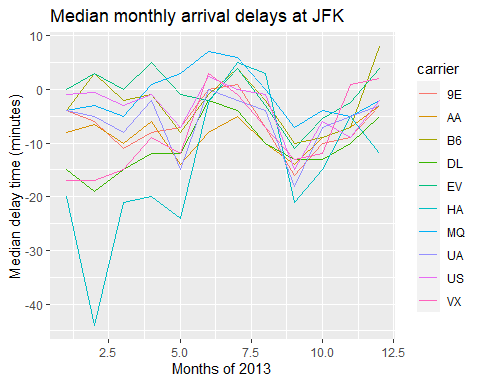
ggplot(median\_EWR, aes(x=month, y = medianDelay,color=carrier))+  
 geom\_line()+  
 labs(title="Median monthly arrival delays at EWR",  
 x ="Months of 2013", y = "Median delay time (minutes)")



median\_JFK = flights %>%   
 filter(is.na(arr\_delay)==FALSE) %>%  
 filter(origin=="JFK") %>%   
 group\_by(month,carrier) %>%   
 summarize(medianDelay = median(arr\_delay))

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

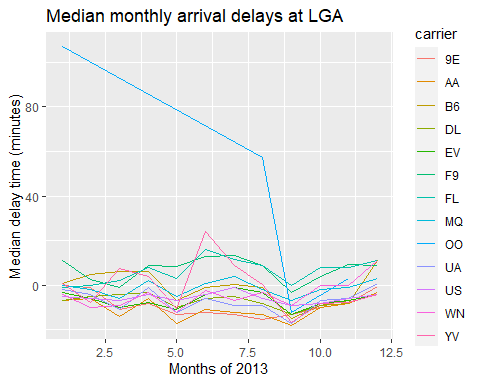
ggplot(median\_JFK, aes(x=month, y = medianDelay,color=carrier))+  
 geom\_line()+  
 labs(title="Median monthly arrival delays at JFK",  
 x ="Months of 2013", y = "Median delay time (minutes)")



median\_LGA = flights %>%   
 filter(is.na(arr\_delay)==FALSE) %>%  
 filter(origin=="LGA") %>%   
 group\_by(month,carrier) %>%   
 summarize(medianDelay = median(arr\_delay))

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

ggplot(median\_LGA, aes(x=month, y = medianDelay,color=carrier))+  
 geom\_line()+  
 labs(title="Median monthly arrival delays at LGA",  
 x ="Months of 2013", y = "Median delay time (minutes)")



flights\_airlines = left\_join(flights,airlines,by="carrier")  
flights\_sfo = flights\_airlines %>%   
 filter(dest=="SFO") %>%  
 group\_by(name) %>%   
 count(name,sort=TRUE,name="SFO")  
  
flights\_total = flights\_airlines %>%   
 filter(is.na(dest)==FALSE) %>%  
 group\_by(name) %>%   
 count(name,sort=TRUE,name="total\_flights")  
  
left\_join(flights\_sfo,flights\_total,by="name") %>%   
 mutate(percent\_flights = SFO/total\_flights)

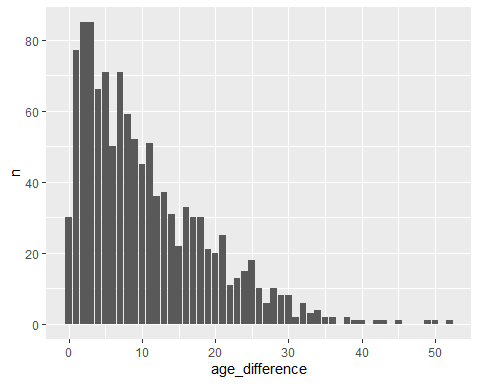
# A tibble: 5 × 4  
# Groups: name [5]  
 name SFO total\_flights percent\_flights  
 <chr> <int> <int> <dbl>  
1 United Air Lines Inc. 6819 58665 0.116   
2 Virgin America 2197 5162 0.426   
3 Delta Air Lines Inc. 1858 48110 0.0386  
4 American Airlines Inc. 1422 32729 0.0434  
5 JetBlue Airways 1035 54635 0.0189

cancellations <- flights %>%   
   
 # just filter for destination == 'SFO'  
 filter(dest == 'SFO') %>%   
   
 # a cancelled flight is one with no `dep\_time`   
 filter(is.na(dep\_time))  
  
#Given that the code provide already filtered the destination, the first step is to group\_by month and to count the number of cancellations. Then, we would call ggplot2 and use histograms where x = months and y = count of cancellations

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.

#Distribution of age gaps  
ageDifference = age\_gaps %>%   
 group\_by(age\_difference) %>%   
 count(age\_difference,sort=TRUE)  
  
ggplot(ageDifference, aes(x = age\_difference,y=n))+  
 geom\_col()



#Most common age gap  
ageDifference[1:2,]

# A tibble: 2 × 2  
# Groups: age\_difference [2]  
 age\_difference n  
 <dbl> <int>  
1 2 85  
2 3 85

#Half plus seven rule - Total occurrence  
sevenRule = age\_gaps %>%   
 #Calculating upper and lower boundaries  
 mutate(LB = actor\_1\_age>(actor\_2\_age/2+7)) %>%   
 mutate(UB = (actor\_2\_age-7)\*2 > actor\_1\_age)  
#Calculating how many times the rule does not apply  
sprintf("The rule does not apply on %s occasions",dim(sevenRule)[1]\*2 - sum(sevenRule$LB) - sum(sevenRule$UB))

[1] "The rule does not apply on 360 occasions"

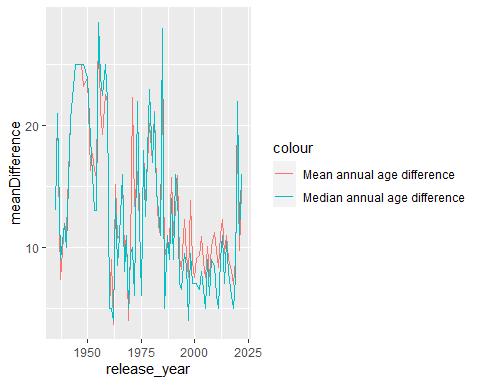
#Actors with the most love interests  
age\_gaps %>%  
 count(actor\_1\_name,sort=TRUE,name="count") %>%   
 slice(which.max(count))

# A tibble: 1 × 2  
 actor\_1\_name count  
 <chr> <int>  
1 Keanu Reeves 24

#Movies with most love interests  
age\_gaps %>%  
 count(movie\_name,sort=TRUE,name="count") %>%   
 slice(which.max(count))

# A tibble: 1 × 2  
 movie\_name count  
 <chr> <int>  
1 Love Actually 7

#Changes in trends of age difference  
age\_trends = age\_gaps %>%   
 group\_by(release\_year) %>%   
 summarize(meanDifference = mean(age\_difference),medianDifference = median(age\_difference))  
#Plot of this difference with time  
ggplot(age\_trends, aes(x = release\_year))+  
 geom\_line(aes(y = meanDifference, color="Mean annual age difference"))+  
 geom\_line(aes(y = medianDifference, color="Median annual age difference"))



#Same gender couples  
sameGender = age\_gaps %>%   
 mutate(SG = character\_1\_gender == character\_2\_gender)  
#Number of occasions  
sum(sameGender$SG)

[1] 23

# 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: No one
* Approximately how much time did you spend on this problem set: 10 hours
* 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.