Homework 1

Max Cheatle

2023-05-14

# Data Manipulation

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

# Flights with 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>

# Flights that 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>

# Flights 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>

# Flights that departed in summer (July, August, and September)  
   
flights %>%   
 filter(month %in% c(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>

# Flights that arrived more than two hours late, but didn't leave late  
  
flights %>%   
 filter(arr\_delay > 120 & dep\_delay <= 0) %>%   
 select(year:day, carrier, flight, arr\_delay, dep\_delay)

# A tibble: 29 × 7  
 year month day carrier flight arr\_delay dep\_delay  
 <int> <int> <int> <chr> <int> <dbl> <dbl>  
 1 2013 1 27 MQ 3728 124 -1  
 2 2013 10 7 EV 5181 130 0  
 3 2013 10 7 AA 1151 124 -2  
 4 2013 10 16 B6 3 122 -3  
 5 2013 11 1 VX 399 194 -2  
 6 2013 3 18 UA 389 140 -3  
 7 2013 4 17 MQ 4540 124 -5  
 8 2013 4 18 AA 707 179 -2  
 9 2013 4 18 AA 2083 143 -5  
10 2013 5 22 MQ 4674 127 -3  
# ℹ 19 more rows

# Flights delayed by at least an hour, but made up over 30 minutes in flight  
  
flights %>%  
 filter(dep\_delay >= 60 & arr\_delay <= dep\_delay + 30) %>%   
 select(year:day, carrier, flight, arr\_delay, dep\_delay)

# A tibble: 24,854 × 7  
 year month day carrier flight arr\_delay dep\_delay  
 <int> <int> <int> <chr> <int> <dbl> <dbl>  
 1 2013 1 1 AA 443 51 71  
 2 2013 1 1 MQ 3944 851 853  
 3 2013 1 1 UA 856 123 144  
 4 2013 1 1 UA 1086 145 134  
 5 2013 1 1 EV 4495 78 96  
 6 2013 1 1 MQ 4646 93 71  
 7 2013 1 1 B6 673 78 77  
 8 2013 1 1 EV 4497 127 115  
 9 2013 1 1 B6 525 125 105  
10 2013 1 1 B6 705 115 122  
# ℹ 24,844 more rows

## 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

# First, I calculate the number of canclled flights, and the total number of flights. Then, I filter for only the minimum/maximum value, and select the appropriate values.   
  
# Finding the minimum number of cancelled flights  
  
flights %>%   
 group\_by(month) %>%  
 summarise(count\_total = n(), count\_cancelled = sum(is.na(dep\_time)), pct\_cancelled = (count\_cancelled/count\_total)\*100) %>%  
 filter(pct\_cancelled == min(pct\_cancelled)) %>%  
 select(month, pct\_cancelled)

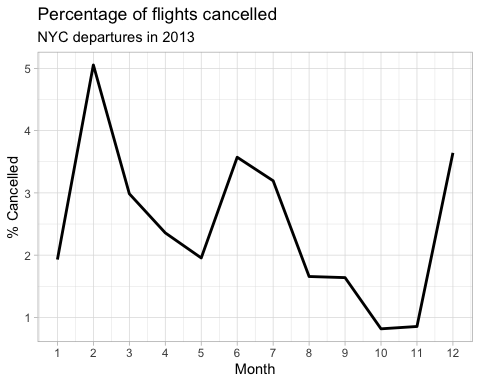
# A tibble: 1 × 2  
 month pct\_cancelled  
 <int> <dbl>  
1 10 0.817

# Finding the maximum number of cancelled flights  
  
flights %>%   
 group\_by(month) %>%  
 summarise(count\_total = n(), count\_cancelled = sum(is.na(dep\_time)), pct\_cancelled = (count\_cancelled/count\_total)\*100) %>%  
 filter(pct\_cancelled == max(pct\_cancelled)) %>%  
 select(month, pct\_cancelled)

# A tibble: 1 × 2  
 month pct\_cancelled  
 <int> <dbl>  
1 2 5.05

# Combined dataframe for plotting  
  
flights %>%   
 group\_by(month) %>%  
 summarise(count\_total = n(), count\_cancelled = sum(is.na(dep\_time)), pct\_cancelled = (count\_cancelled/count\_total)\*100) %>%  
 ggplot(aes(x=month, y=pct\_cancelled)) +  
 geom\_line(size=1) +  
 theme\_light() +  
 scale\_x\_continuous(limit = c(1,12), breaks = 1:12) +  
 labs(title = "Percentage of flights cancelled", subtitle = "NYC departures in 2013", x = "Month", y = "% Cancelled") +  
 NULL

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



* Lowest cancellation month: 10 (October), 0.82%
* Highest cancellation month: 2 (February), 5.05%
* Cancellations appear to be highest in February, the summer months, and December. This may coincide with peak travel seasons, where the flight network is strained by volume generating increased cancellation rates.

## 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.

# Counting flights by tailnum, and joining to planes table  
  
tailnum\_depts <- flights %>%  
 group\_by(tailnum) %>%  
 summarise(count = n(), na.rm = TRUE) %>%  
 arrange(desc(count)) %>%  
 select(tailnum, count)  
  
view(tailnum\_depts)  
  
plane\_depts <- left\_join(tailnum\_depts, planes, "tailnum") %>%  
 filter(!is.na(tailnum))  
  
plane\_depts %>%  
 arrange(desc(count))

# A tibble: 4,043 × 10  
 tailnum count 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

# Finding the plane with greatest number of flights, and more than 50 seats  
  
plane\_depts %>%   
 filter(seats > 50, na.rm = TRUE) %>%  
 arrange(desc(count))

# A tibble: 3,200 × 10  
 tailnum count year type manufacturer model engines seats speed engine  
 <chr> <int> <int> <chr> <chr> <chr> <int> <int> <int> <chr>   
 1 N328AA 393 1986 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 2 N338AA 388 1987 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 3 N327AA 387 1986 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 4 N335AA 385 1987 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 5 N323AA 357 1986 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 6 N319AA 354 1985 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 7 N336AA 353 1987 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 8 N329AA 344 1987 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
 9 N789JB 332 2011 Fixed wing… AIRBUS A320… 2 200 NA Turbo…  
10 N324AA 328 1986 Fixed wing… BOEING 767-… 2 255 NA Turbo…  
# ℹ 3,190 more rows

# It's plane N328AA, a Boeing 767-223. Let's plot all of its flights  
  
flights %>%  
 filter(tailnum == "N328AA") %>% # Finding all flights with our tailnumber  
 group\_by(dest) %>%   
 summarise(flights\_to = n()) %>% # Counting flights per destination  
 arrange(desc(flights\_to))

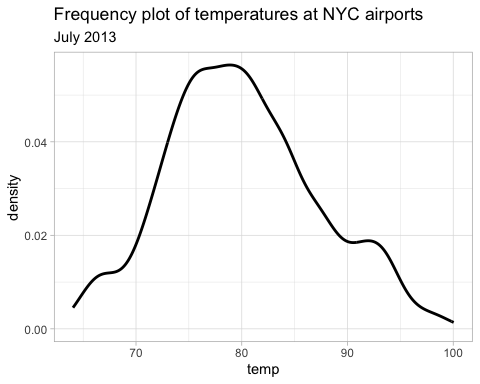
# A tibble: 6 × 2  
 dest flights\_to  
 <chr> <int>  
1 LAX 313  
2 SFO 52  
3 MIA 25  
4 BOS 1  
5 MCO 1  
6 SJU 1

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

# Distribution of temperature in July 2013.  
  
weather %>%  
 filter(month == '7') %>% # Gathering July data only to perform a visual inspection  
 select(temp) %>%   
 summary()

temp   
 Min. : 64.04   
 1st Qu.: 75.02   
 Median : 78.98   
 Mean : 80.07   
 3rd Qu.: 84.20   
 Max. :100.04

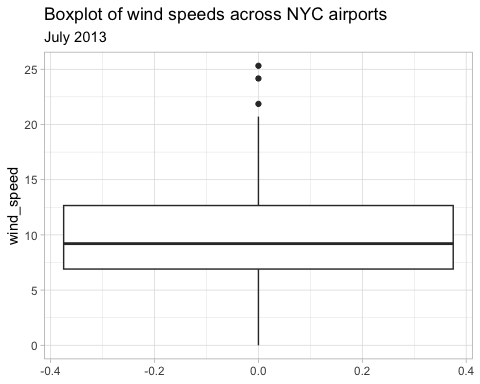
weather %>%  
 filter(month == '7') %>%  
 ggplot(aes(x=temp)) + # Plotting as density plot  
 geom\_density(size = 1) +  
 labs(title = "Frequency plot of temperatures at NYC airports", subtitle = "July 2013") +  
 theme\_light() +  
 NULL



The above distribution is *somewhat* normally distributed, with a slightly longer right tail. (positive skew). The mean is 80.07, and the median is 70.98, confirming the positive skew observed in the frequency plot.

# Outliers in wind\_speed, opted for a boxplot to highlight outliers and manual identification thereafter.   
  
weather %>%  
 filter(month == '7') %>%  
 ggplot(aes(y=wind\_speed)) +  
 geom\_boxplot() +  
 labs(title = "Boxplot of wind speeds across NYC airports", subtitle = "July 2013") +  
 theme\_light() +  
 NULL

Warning: Removed 2 rows containing non-finite values (`stat\_boxplot()`).



The outliers are:

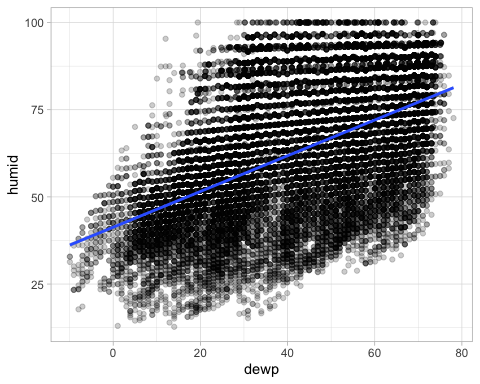
* Hour 18, Day 23, Wind Speed 25.3
* Hour 19, Day 20, Wind Speed 24.2
* Hour 17, Day 20, Wind Speed 21.9

## Relationship between `dewp` and `humid`  
  
# First, I plot a scatterplot to check the overall trend.  
  
weather %>%  
 ggplot(aes(x=dewp, y=humid)) +  
 geom\_point(alpha = 0.2) +  
 geom\_smooth(method = lm) +  
 theme\_light() +  
 NULL

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

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

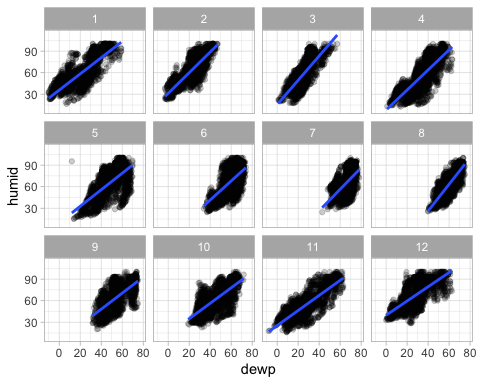
Warning: Removed 1 rows containing missing values (`geom\_point()`).



# Does this apply to all months?  
  
weather %>%  
 ggplot(aes(x=dewp, y=humid)) +  
 geom\_point(alpha = 0.2) +  
 facet\_wrap(~ month) +  
 geom\_smooth(method = lm) +  
 theme\_light() +  
 NULL

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

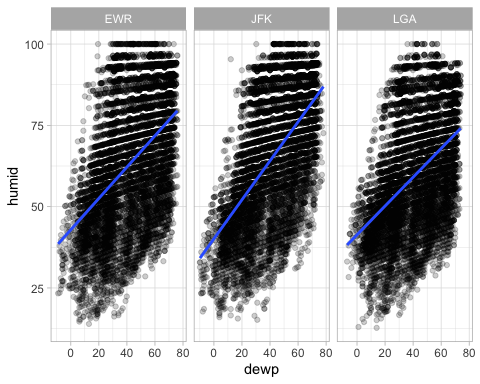
Warning: Removed 1 rows containing non-finite values (`stat\_smooth()`).  
Removed 1 rows containing missing values (`geom\_point()`).



# Does this apply to all origin airports?  
  
weather %>%  
 ggplot(aes(x=dewp, y=humid)) +  
 geom\_point(alpha = 0.2) +  
 facet\_wrap(~ origin) +  
 geom\_smooth(method = lm) +  
 theme\_light() +  
 NULL

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

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



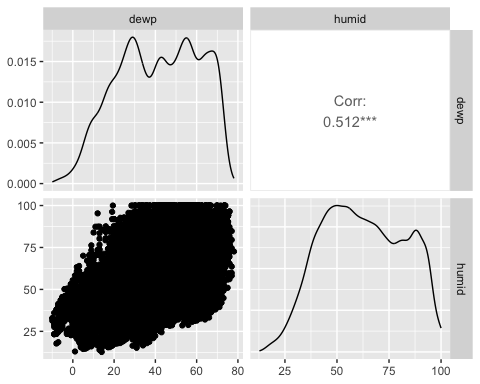
weather %>%  
 select(dewp, humid) %>%  
 ggpairs()

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

Warning in ggally\_statistic(data = data, mapping = mapping, na.rm = na.rm, :  
Removing 1 row that contained a missing value

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

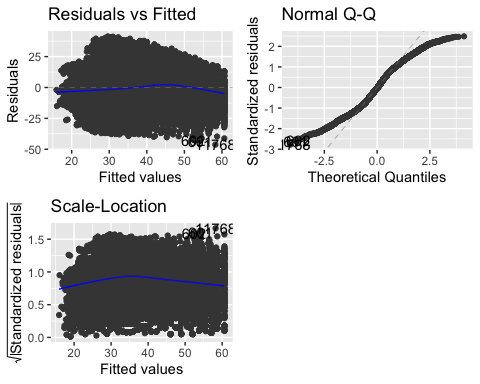
Warning: Removed 1 rows containing non-finite values (`stat\_density()`).



reg\_dewp\_humid <- lm(dewp ~ humid, data = weather)  
summary(reg\_dewp\_humid)

Call:  
lm(formula = dewp ~ humid, data = weather)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-46.217 -14.034 -0.588 13.881 41.405   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 9.428376 0.347797 27.11 <2e-16 \*\*\*  
humid 0.511940 0.005312 96.37 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 16.65 on 26112 degrees of freedom  
 (1 observation deleted due to missingness)  
Multiple R-squared: 0.2623, Adjusted R-squared: 0.2623   
F-statistic: 9287 on 1 and 26112 DF, p-value: < 2.2e-16

autoplot(reg\_dewp\_humid, 1:3)



The first scatter plot shows a positive correlation between dewp and humid. To be precise, the correlation coefficient is 0.512, a relatively significant positive correlation.

The following 2 plots show that this trend applies across all months and origin airports. This defends the robustness of said correlation.

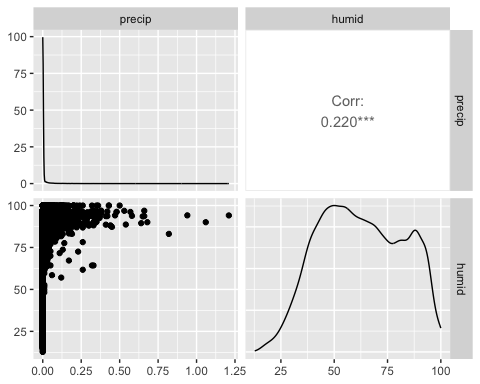
Finally, a quick linear regression shows, again, a positive relationship. Though, our R-Squared value is small since the model is rather unsophisticated. However, the correlations and regression show that there is a positive relationship between the variables, of which is *partly* causal.

# Relationship between `precip` and `visib`?  
  
weather %>%  
 select(precip, humid) %>%  
 ggpairs()

Warning in ggally\_statistic(data = data, mapping = mapping, na.rm = na.rm, :  
Removing 1 row that contained a missing value

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

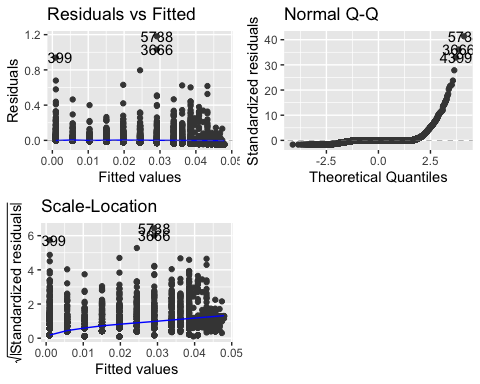
Warning: Removed 1 rows containing non-finite values (`stat\_density()`).



reg\_precip\_visib <- lm(precip ~ visib, data = weather)  
summary(reg\_precip\_visib)

Call:  
lm(formula = precip ~ visib, data = weather)  
  
Residuals:  
 Min 1Q Median 3Q Max   
-0.04791 -0.00097 -0.00097 -0.00097 1.18086   
  
Coefficients:  
 Estimate Std. Error t value Pr(>|t|)   
(Intercept) 4.791e-02 8.156e-04 58.75 <2e-16 \*\*\*  
visib -4.694e-03 8.603e-05 -54.56 <2e-16 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
Residual standard error: 0.02857 on 26113 degrees of freedom  
Multiple R-squared: 0.1023, Adjusted R-squared: 0.1023   
F-statistic: 2977 on 1 and 26113 DF, p-value: < 2.2e-16

autoplot(reg\_precip\_visib, 1:3)



There appears to be little to no trend correlation between precip and humid. The correlation plot shows that when precipitation is low, humidity can vary hugely. However, in order for precipitation to be high, there must be a high (above 50%) level of humidity. Hence, to an extent, one could say these are positively correlated in a non-linear fashion.

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

# Planes missing manufacturing dates: 70  
  
planes %>%   
 summarise(missing\_years = sum(is.na(year)))

# A tibble: 1 × 1  
 missing\_years  
 <int>  
1 70

# Five most common manufacturers: Boeing, Airbus, Bombadier, Embraer, McDonnell  
  
planes %>%  
 group\_by(manufacturer) %>%   
 summarise(number\_of\_planes = n()) %>%   
 arrange(desc(number\_of\_planes))

# A tibble: 35 × 2  
 manufacturer number\_of\_planes  
 <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

# Generating a list of all the planes, their repsective manufacturer, and manufactured year that departed from NYC in 2013.  
  
tailnums\_dept\_nyc <- as.tibble(c(unique(flights$tailnum))) %>%   
   
 # Creating this as a tibble so that it can be joined later  
   
 rename(tailnum = value)

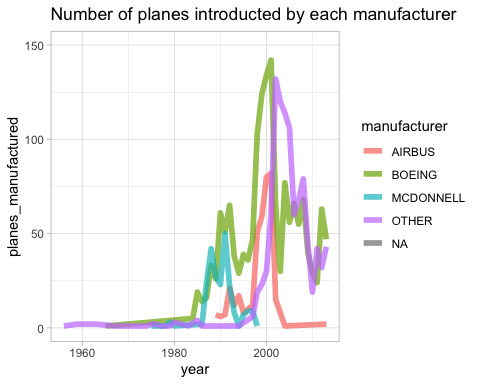
Warning: `as.tibble()` was deprecated in tibble 2.0.0.  
ℹ Please use `as\_tibble()` instead.  
ℹ The signature and semantics have changed, see `?as\_tibble`.

planes\_dept\_nyc <- left\_join(tailnums\_dept\_nyc, planes, "tailnum") %>%  
 select(tailnum, manufacturer, year)  
   
# Ditribution of manufacturer changed over time  
  
planes\_dept\_nyc %>%   
 mutate(manufacturer = case\_when(   
 manufacturer == "AIRBUS INDUSTRIE" ~ "AIRBUS",  
 manufacturer == "MCDONNELL DOUGLAS" ~ "MCDONNELL",  
 manufacturer == "MCDONNELL DOUGLAS AIRCRAFT CO" ~ "MCDONNELL",  
 manufacturer == "MCDONNELL DOUGLAS CORPORATION" ~ "MCDONNELL",  
 manufacturer == "BOEING" ~ "BOEING",  
 manufacturer != c("AIRBUS INDUSTRIE",   
 "MCDONNELL DOUGLAS",   
 "MCDONNELL DOUGLAS AIRCRAFT CO",   
 "MCDONNELL DOUGLAS CORPORATION",   
 "BOEING") ~ "OTHER"  
 )) %>%   
   
 # The code above renames the manufacturer variants into their generic names.  
   
 group\_by(year, manufacturer) %>%   
 summarise(planes\_manufactured = n()) %>%   
 arrange(desc(planes\_manufactured)) %>%   
   
 # Next step is to plot this  
   
 ggplot(aes(x=year, y=planes\_manufactured, color = manufacturer)) +  
 geom\_line(size = 2, alpha = 0.7) +  
 scale\_y\_continuous(limits = c(0,150)) +  
 theme\_light() +  
 labs(title = "Number of planes introducted by each manufacturer") +  
 NULL

Warning: There was 1 warning in `mutate()`.  
ℹ In argument: `manufacturer = case\_when(...)`.  
Caused by warning in `manufacturer != c("AIRBUS INDUSTRIE", "MCDONNELL DOUGLAS",  
 "MCDONNELL DOUGLAS AIRCRAFT CO", "MCDONNELL DOUGLAS CORPORATION", "BOEING")`:  
! longer object length is not a multiple of shorter object length

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

Warning: Removed 5 rows containing missing values (`geom\_line()`).



* 70 planes are missing their manufacturing year on the planes table
* The five most common manufacturers are Boeing, Airbus, Bombadier, Embraer and McDonnell
* Boeing and McDonnell were the early market leaders, with essentially zero competition from any other manufacturer. That said, Airbus would soon enter as a major player, but not for very long - the data appears to show a decline in the European manufacturer’s deliveries in the early 2000s. In the late 2000s, and into the 2010s, Boeing remains the strongest single manufacturer, but does face competition from other manufacturers - most likely for smaller aircraft.

## Problem 6: Use the flights and planes tables to answer the following questions:

# Using the previous tibble, I filter out nulls, then select only the minimum year value  
  
planes\_dept\_nyc %>%   
 filter(!is.na(year)) %>%   
 filter(year == min(year)) %>%   
 select(tailnum, year)

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

The oldest plane to depart from NYC in 2013 was manufactured in 1956!

# How many airplanes that flew from New York City are included in the planes table?  
  
planes\_dept\_nyc # Values with nulls here were not included in the planes table, hence when they were joined from the flights table, they didn't receive values for manufacturer and year. So, we simply count them.

# A tibble: 4,044 × 3  
 tailnum manufacturer year  
 <chr> <chr> <int>  
 1 N14228 BOEING 1999  
 2 N24211 BOEING 1998  
 3 N619AA BOEING 1990  
 4 N804JB AIRBUS 2012  
 5 N668DN BOEING 1991  
 6 N39463 BOEING 2012  
 7 N516JB AIRBUS INDUSTRIE 2000  
 8 N829AS CANADAIR 1998  
 9 N593JB AIRBUS 2004  
10 N3ALAA <NA> NA  
# ℹ 4,034 more rows

planes\_dept\_nyc %>%   
 summarise(null\_manf = sum(is.na(manufacturer)), null\_year = sum(is.na(year)), total\_planes = n()) # This shows us that some planes do exist in the planes database without a year, so we need to check if all of them have manufacturer values in the next line of code.

# A tibble: 1 × 3  
 null\_manf null\_year total\_planes  
 <int> <int> <int>  
1 722 792 4044

planes %>%   
 summarise(null\_manf = sum(is.na(manufacturer))) # None of them are missing a manufacturer, so we are safe to use the null manufacturing values to count the number of planes on the flights database, but not on the planes database.

# A tibble: 1 × 1  
 null\_manf  
 <int>  
1 0

planes\_dept\_nyc %>%   
 summarise(missing\_from\_planes = sum(is.na(manufacturer)), total = n(), pct\_missing\_from\_planes = (sum(is.na(manufacturer))/n())\*100)

# A tibble: 1 × 3  
 missing\_from\_planes total pct\_missing\_from\_planes  
 <int> <int> <dbl>  
1 722 4044 17.9

722 planes that flew from NYC are missing from the planes table. That’s 17.89%, not great!

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

# First, it's important to understand if NA arr\_delay values = 0, or cancelled flights.   
  
flights %>%   
 group\_by(origin, month) %>%   
 filter(is.na(arr\_delay)) # Looks like they're cancelled, I will remove them in the following steps.

# A tibble: 9,430 × 19  
# Groups: origin, month [36]  
 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 1525 1530 -5 1934 1805  
 2 2013 1 1 1528 1459 29 2002 1647  
 3 2013 1 1 1740 1745 -5 2158 2020  
 4 2013 1 1 1807 1738 29 2251 2103  
 5 2013 1 1 1939 1840 59 29 2151  
 6 2013 1 1 1952 1930 22 2358 2207  
 7 2013 1 1 2016 1930 46 NA 2220  
 8 2013 1 1 NA 1630 NA NA 1815  
 9 2013 1 1 NA 1935 NA NA 2240  
10 2013 1 1 NA 1500 NA NA 1825  
# ℹ 9,420 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>

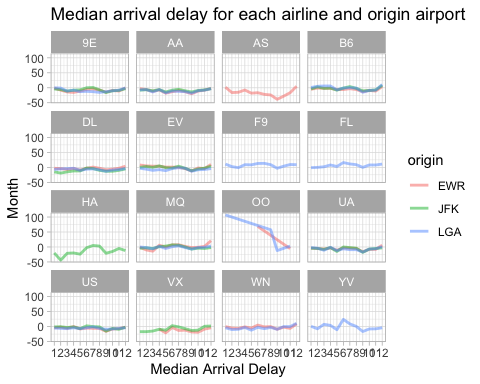
# Here, I have caluclated the median arrival delay for each origin airport.  
  
flights %>%   
 group\_by(origin, month) %>%   
 filter(!is.na(arr\_delay)) %>% # Removing nulls as discussed  
 summarise(median\_arr\_delay = median(arr\_delay))

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

# A tibble: 36 × 3  
# Groups: origin [3]  
 origin month median\_arr\_delay  
 <chr> <int> <dbl>  
 1 EWR 1 0  
 2 EWR 2 -2  
 3 EWR 3 -4  
 4 EWR 4 -1  
 5 EWR 5 -6  
 6 EWR 6 -1  
 7 EWR 7 -2  
 8 EWR 8 -5  
 9 EWR 9 -13  
10 EWR 10 -6  
# ℹ 26 more rows

# Now let's look at an airline-by-airline basis, and plot for each month.   
  
flights %>%   
 group\_by(origin, month, carrier) %>%   
 filter(!is.na(arr\_delay)) %>% # Removing nulls as discussed  
 summarise(median\_arr\_delay = median(arr\_delay)) %>%  
 ggplot(aes(x=month, y=median\_arr\_delay, colour = origin)) + # Now lets plot these on a line chart  
 geom\_line(size = 1, alpha = 0.5) +  
 facet\_wrap(~ carrier) +  
 theme\_light() +  
 labs(title = "Median arrival delay for each airline and origin airport", x = "Median Arrival Delay", y = "Month") +  
 scale\_x\_continuous(limit = c(1,12), breaks = 1:12, ) +  
 NULL

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

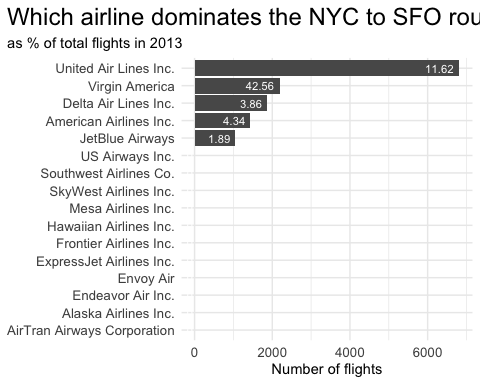


## Problem 8: Let’s take a closer look at what carriers service the route to San Francisco International (SFO). Join the flights and airlines tables and count which airlines flew the most to SFO. Produce a new dataframe, fly\_into\_sfo that contains three variables: the name of the airline, e.g., United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

# First, lets join the two tables so that we have all the neccessary data avaialable  
  
carriers\_to\_sfo <- left\_join(flights, airlines, "carrier") %>%   
   
 # Now let's group by carrier name for the upcoming analysis  
   
 group\_by(name) %>%   
  
 # Filtering out non-SFO flights to count trips by airline  
   
 filter(dest == "SFO") %>%   
 summarise(count = n())  
  
 # Now lets take a step back to count total flights by each airline  
  
carriers\_total <- left\_join(flights, airlines, "carrier") %>%   
 group\_by(name) %>%   
 summarise(total\_flights = n())  
  
# Building the final table  
  
fly\_into\_sfo <- left\_join(carriers\_total, carriers\_to\_sfo, "name") %>%   
 mutate(count = case\_when(  
 is.na(count) ~ 0,   
   
 # Changing NAs to zero for clarity  
   
 TRUE ~ count  
 ),  
 percent = count/total\_flights\*100,  
 percent = round(percent, digits = 2)) %>%   
 select(name, count, percent) %>%   
 arrange(desc(percent))

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 = 1.2,   
 colour = "white",   
 size = 3)+  
   
 # 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=10),  
   
 # 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



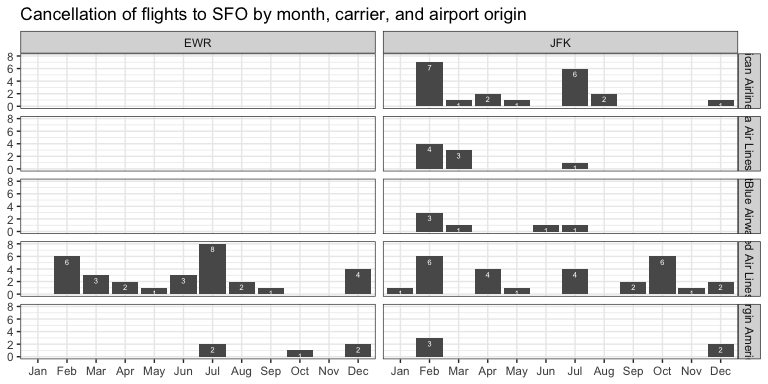
## Problem 9: Let’s take a look at cancellations of flights to SFO. We create a new dataframe cancellations as follows

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.



# Here's the code for the above plot:  
  
# First, let's group the data by month, carrier, and origin and count.  
  
cancellations %>%   
 group\_by(month, carrier, origin) %>%   
 summarise(count = n(), .groups = "drop") %>%   
   
 # Let's get the airline names and month names to replicate the plot axes.   
   
 left\_join(y=airlines,by = "carrier") %>%   
 mutate(month = case\_when(  
 month == 1 ~ "Jan",  
 month == 2 ~ "Feb",  
 month == 3 ~ "Mar",  
 month == 4 ~ "Apr",  
 month == 5 ~ "May",  
 month == 6 ~ "Jun",  
 month == 7 ~ "Jul",  
 month == 8 ~ "Aug",  
 month == 9 ~ "Sep",  
 month == 10 ~ "Oct",  
 month == 11 ~ "Nov",  
 month == 12 ~ "Dec",  
 ),  
 month = factor(month, levels = c("Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"))) %>% # Ordering the monhts for the x-axis. This gave me some trouble, I think R doesn't recognise abbreviated months?  
   
 # Now let's plot it. Trying to replicate the above plot as accurately as possible.   
  
 ggplot(aes(x=month, y=count)) +  
 geom\_bar(stat = "identity") +  
 geom\_text(aes(label = count), vjust = 1.5, size = 2, color = "white") +   
 facet\_grid(row = vars(name), col = vars(origin)) +  
 labs(title = "Cancellation of flights to SFO by month, carrier, and airport origin",x = NULL, y = NULL) +  
 theme\_bw() +  
 NULL



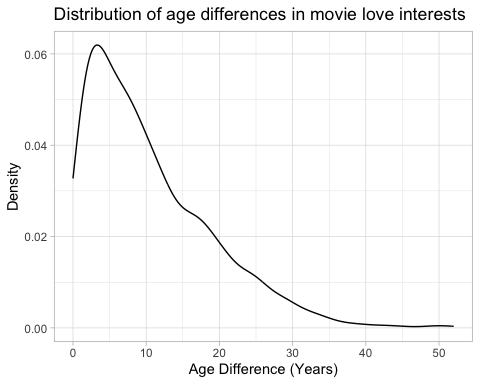
# To explain the data organisation:  
# - First we group by month, carrier, and origin so that our counting of cancellation is executed under the appropriate conditions.  
# - We then need to order the months correctly, to preserve chronology.

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

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.

view(age\_gaps)  
  
# First checking the distribution of age\_difference to get an overview of the variable of interest  
  
age\_gaps %>%   
 ggplot(aes(x=age\_difference)) +  
 geom\_density() +  
 labs(title = "Distribution of age differences in movie love interests", y = "Density", x = "Age Difference (Years)") +  
 theme\_light() +  
 NULL



# Let's see the summary statistics too  
  
age\_gaps %>%  
 select(age\_difference) %>%   
 summary()

age\_difference   
 Min. : 0.00   
 1st Qu.: 4.00   
 Median : 8.00   
 Mean :10.42   
 3rd Qu.:15.00   
 Max. :52.00

Age difference disitribution is positively skewed, with a minimum of 0 and maximum of 52. The mean is 10.42 years.

# To explore how many of these love interests violate the half plus seven rule, we first need to check if the actor\_1\_age column is always larger than the actor\_2\_age column  
  
age\_gaps %>%   
 select(actor\_1\_age, actor\_2\_age) %>%   
 mutate(positive\_delta = actor\_1\_age - actor\_2\_age) %>%  
 mutate(positive\_delta = case\_when(  
 positive\_delta >= 0 ~ "T",  
 TRUE ~ "F"  
 )) %>%   
 group\_by(positive\_delta) %>%   
 summarise(n())

# A tibble: 1 × 2  
 positive\_delta `n()`  
 <chr> <int>  
1 T 1155

# The data is already nice a tidy for our use! Let's see how many violate the half plus seven now...  
  
age\_gaps %>%   
 mutate(half\_plus\_seven = (actor\_1\_age/2)+7) %>%   
 mutate(half\_plus\_seven = case\_when(  
 actor\_2\_age < half\_plus\_seven ~ 'FAIL',  
 TRUE ~ "PASS"  
 )) %>%   
 group\_by(half\_plus\_seven) %>%   
 summarise(n())

# A tibble: 2 × 2  
 half\_plus\_seven `n()`  
 <chr> <int>  
1 FAIL 326  
2 PASS 829

The half plus seven rule is violated in 326 of the movie love interests in our dataset. Those ones probably didn’t last beyond the scope of the movie!

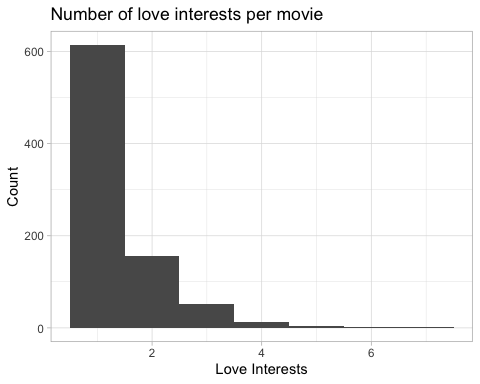
# Now we have an overview of movie love interests, which movies had the most love interests squeezed into their runtime?  
  
age\_gaps %>%   
 group\_by(movie\_name, release\_year) %>% # I also grouped by release year, just in case there are repeated movie names  
 summarise(love\_interests = n()) %>%   
 arrange(desc(love\_interests))

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

# A tibble: 838 × 3  
# Groups: movie\_name [830]  
 movie\_name release\_year love\_interests  
 <chr> <dbl> <int>  
 1 Love Actually 2003 7  
 2 The Family Stone 2007 6  
 3 A View to a Kill 1985 5  
 4 He's Just Not That Into You 2009 5  
 5 Mona Lisa Smile 2003 5  
 6 American Pie 1999 4  
 7 Boogie Nights 1997 4  
 8 Closer 2004 4  
 9 Pushing Tin 1999 4  
10 Sex and the City 2008 4  
# ℹ 828 more rows

# Out of interest, let's plot this in a density plot  
  
age\_gaps %>%   
 group\_by(movie\_name, release\_year) %>%  
 summarise(love\_interests = n()) %>%   
 arrange(desc(love\_interests)) %>%   
 ggplot(aes(x=love\_interests)) +  
 geom\_histogram(binwidth = 1) +  
 labs(title = "Number of love interests per movie", x = 'Love Interests', y = "Count") +  
 theme\_light() +  
 NULL

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



The movie ‘Love Actually’ had the most love interests, with 7. Appropriately named.

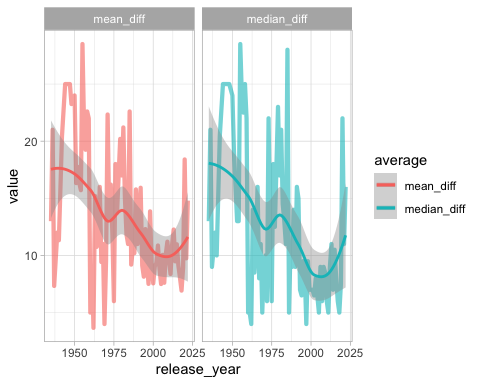
# Now let's check which actors/actresses are popular in the movie relationship market. We need a single list of all the protagonists first.  
  
long\_names <- data.frame(c(age\_gaps$actor\_1\_name, age\_gaps$actor\_2\_name))  
  
colnames(long\_names) <- "names"  
  
# Now I count for the frequency of each name  
  
long\_names %>%  
 group\_by(names) %>%   
 summarise(count = n()) %>%   
 arrange(desc(count))

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

Keanu Reeves and Adam Sandler, popular! Kiera Knightly the most popular woman.

# It's always good to check trends over time. Let's have a look  
  
age\_gaps %>%   
 group\_by(release\_year) %>%   
 summarise(mean\_diff = mean(age\_difference), median\_diff = median(age\_difference)) %>%  
 select(release\_year, mean\_diff, median\_diff) %>%   
 pivot\_longer(cols = c("mean\_diff", 'median\_diff'), names\_to = "average", values\_to = "value") %>%   
   
# Have to long the data for ggplot() to perform properly  
  
 ggplot(aes(x=release\_year, y=value, color=average)) +  
 geom\_line(size = 1.5, alpha = 0.6) +  
 geom\_smooth(method = loess) +  
 facet\_wrap(~ average) +  
 theme\_light()

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



Age differences in relationships seem to have been trending downwards over the years, both in mean and median terms. However, recent years seem to have returned to some much larger age differences than seen in the past 20 years or so.

# Let's take a look at same gender love interests in Hollywood  
  
age\_gaps %>%   
 select(release\_year, character\_1\_gender, character\_2\_gender) %>%   
 mutate(gender\_match = case\_when(  
 character\_1\_gender == character\_2\_gender ~ "TRUE",  
 TRUE ~ "FALSE"  
 )) %>%   
 group\_by(gender\_match) %>%   
 summarise(n())

# A tibble: 2 × 2  
 gender\_match `n()`  
 <chr> <int>  
1 FALSE 1132  
2 TRUE 23

Only 23 of the 1155 movie love interests are the same gender. Let’s hope for more in the future.