Divij Nandan Sharma

2024-05-16

Table of Contents

# Data Manipulation nycflights13

## Problem 1: 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

# What months had the highest and lowest % of cancelled flights?  
  
pct\_share <- flights %>%   
 filter(is.na(dep\_time)) %>%   
 group\_by(month) %>%   
 summarise(Count = n()) %>%   
 mutate(pct\_share = Count \* 100 / sum(Count)) %>%   
 arrange(desc(pct\_share))  
  
print(pct\_share)

## # A tibble: 12 × 3  
## month Count pct\_share  
## <int> <int> <dbl>  
## 1 2 1261 15.3   
## 2 12 1025 12.4   
## 3 6 1009 12.2   
## 4 7 940 11.4   
## 5 3 861 10.4   
## 6 4 668 8.09  
## 7 5 563 6.82  
## 8 1 521 6.31  
## 9 8 486 5.89  
## 10 9 452 5.48  
## 11 10 236 2.86  
## 12 11 233 2.82

## Problem 2: What plane (specified by the tailnum variable) traveled the most times from New York City airports in 2013?

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.

most\_flown <- flights %>%  
 group\_by(tailnum) %>%   
 summarise(count = n()) %>%   
 arrange(desc(count)) %>%   
 left\_join(planes, by = "tailnum") %>%   
 filter(!is.na(tailnum))  
   
print(most\_flown)

## # 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

most\_flown\_dest\_2013 <- most\_flown %>%   
 filter(seats > 50) %>%   
 head(1) %>%   
 left\_join(flights, by = "tailnum") %>%   
 group\_by(carrier, dest) %>%   
 count()  
  
print(most\_flown\_dest\_2013)

## # A tibble: 6 × 3  
## # Groups: carrier, dest [6]  
## carrier dest n  
## <chr> <chr> <int>  
## 1 AA BOS 1  
## 2 AA LAX 313  
## 3 AA MCO 1  
## 4 AA MIA 25  
## 5 AA SFO 52  
## 6 AA SJU 1

## Problem 3: 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?

# Missing date of manufacture  
missing\_year <- planes %>%   
 filter(is.na(year)) %>%   
 count()  
print(missing\_year)

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

# Cleaning the dataset - combining same manufacturers, adding an others category  
planes\_clean <- planes %>%  
 mutate(  
 manufacturer = case\_when(  
 manufacturer == 'BOEING' ~ 'BOEING',   
 manufacturer == 'AIRBUS' ~ 'AIRBUS',   
 manufacturer == 'AIRBUS INDUSTRIE' ~ 'AIRBUS',   
 manufacturer == 'EMBRAER' ~ 'EMBRAER',   
 manufacturer == 'BOMBARDIER INC' ~ 'BOMBARDIER INC',   
 manufacturer == 'MCDONNELL DOUGLAS' ~ 'MCDONNELL DOUGLAS',   
 manufacturer == 'MCDONNELL DOUGLAS AIRCRAFT CO' ~ 'MCDONNELL DOUGLAS',   
 manufacturer == 'MCDONNELL DOUGLAS CORPORATION' ~ 'MCDONNELL DOUGLAS',  
 TRUE ~ 'OTHER'  
 )  
 )  
  
# Most common manufacturers  
common\_manuf <- planes\_clean %>%   
 count(manufacturer, sort = TRUE) %>%   
 head(5)  
print(common\_manuf)

## # A tibble: 5 × 2  
## manufacturer n  
## <chr> <int>  
## 1 BOEING 1630  
## 2 AIRBUS 736  
## 3 BOMBARDIER INC 368  
## 4 EMBRAER 299  
## 5 MCDONNELL DOUGLAS 237

# Manufacturer distribution  
manuf\_dist <- planes\_clean %>%   
 left\_join(flights, by = 'tailnum') %>%   
 group\_by(month, manufacturer) %>%   
 summarise(count = n())

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

print(manuf\_dist)

## # A tibble: 72 × 3  
## # Groups: month [12]  
## month manufacturer count  
## <int> <chr> <int>  
## 1 1 AIRBUS 7283  
## 2 1 BOEING 6623  
## 3 1 BOMBARDIER INC 1925  
## 4 1 EMBRAER 5364  
## 5 1 MCDONNELL DOUGLAS 872  
## 6 1 OTHER 458  
## 7 2 AIRBUS 6654  
## 8 2 BOEING 6048  
## 9 2 BOMBARDIER INC 1809  
## 10 2 EMBRAER 4908  
## # ℹ 62 more rows

## Problem 4: Which carriers service the route to San Francisco International (SFO).

fly\_into\_sfo <- flights %>%   
 filter(dest == 'SFO') %>%   
 count(carrier, sort = TRUE) %>%   
 left\_join(airlines, by = 'carrier') %>%   
 mutate( pct = n / sum(n)) %>%   
 select(name, n, pct)  
print(fly\_into\_sfo)

## # A tibble: 5 × 3  
## name n pct  
## <chr> <int> <dbl>  
## 1 United Air Lines Inc. 6819 0.512   
## 2 Virgin America 2197 0.165   
## 3 Delta Air Lines Inc. 1858 0.139   
## 4 American Airlines Inc. 1422 0.107   
## 5 JetBlue Airways 1035 0.0776

And here is some bonus ggplot code to plot your dataframe

sfo\_plot <- fly\_into\_sfo %>%   
   
 # sort 'name' of airline by the numbers it times to flew to SFO  
 mutate(name = fct\_reorder(name, n)) %>%   
   
 ggplot() +   
   
 aes(x = n,   
 y = name) +  
   
 # a simple bar/column plot  
 geom\_col() +  
   
 # add labels, so each bar shows the % of total flights   
 geom\_text(aes(label = pct),  
 hjust = 1,   
 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\_bw() +   
   
 # 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  
   
print(sfo\_plot)

## Problem 5: 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.



To recreate the graph starting with the cancellations dataset, I would:

1. **Group and Summarize**:
   * Group the filtered data by **month**, **carrier**, and **origin**.
   * Summarize the grouped data to count the number of cancellations within each group.
2. **Prepare Data for Plotting**:
   * Convert **month** to a factor to ensure the months are plotted in order.
3. **Plot the Data**:
   * Use **ggplot** to create a bar plot.
   * Map **month** to the x-axis and the count of cancellations to the y-axis.
   * Use **facet\_grid** to create separate panels for each origin airport.
   * Use **geom\_bar** to create the bar plot

# Rents in San Francsisco 2000-2018

# download directly off tidytuesdaygithub repo  
  
rent <- readr::read\_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2022/2022-07-05/rent.csv')

## Rows: 200796 Columns: 17  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (8): post\_id, nhood, city, county, address, title, descr, details  
## dbl (9): date, year, price, beds, baths, sqft, room\_in\_apt, lat, lon  
##   
## 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 are the variable types? Do they all correspond to what they really are? Which variables have most missing values?

# YOUR CODE GOES HERE  
skimr::skim(rent)

Data summary

|  |  |
| --- | --- |
| Name | rent |
| Number of rows | 200796 |
| Number of columns | 17 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| character | 8 |
| numeric | 9 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: character**

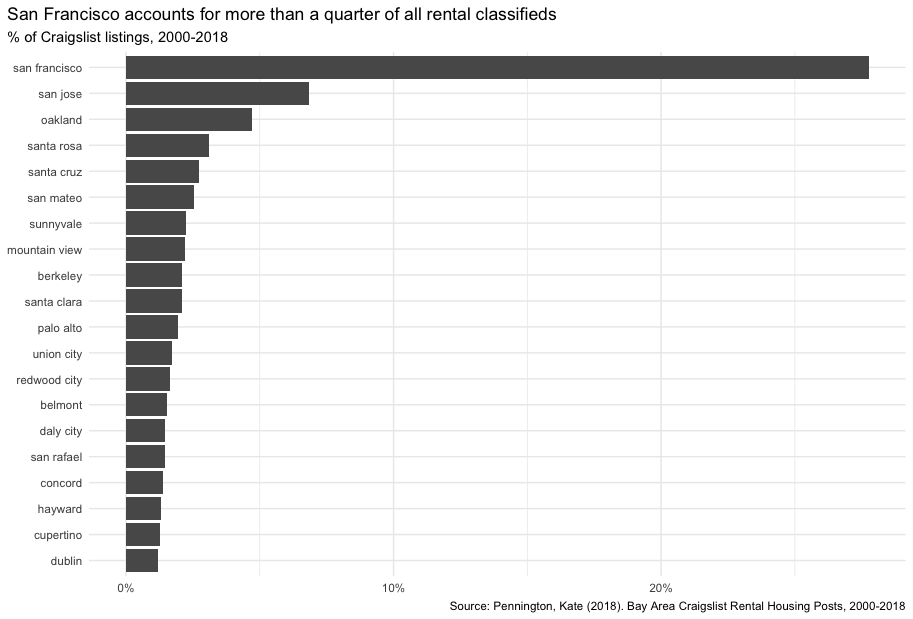
| skim\_variable | n\_missing | complete\_rate | min | max | empty | n\_unique | whitespace |
| --- | --- | --- | --- | --- | --- | --- | --- |
| post\_id | 0 | 1.00 | 9 | 14 | 0 | 200796 | 0 |
| nhood | 0 | 1.00 | 4 | 43 | 0 | 167 | 0 |
| city | 0 | 1.00 | 5 | 19 | 0 | 104 | 0 |
| county | 1394 | 0.99 | 4 | 13 | 0 | 10 | 0 |
| address | 196888 | 0.02 | 1 | 38 | 0 | 2869 | 0 |
| title | 2517 | 0.99 | 2 | 298 | 0 | 184961 | 0 |
| descr | 197542 | 0.02 | 13 | 16975 | 0 | 3025 | 0 |
| details | 192780 | 0.04 | 4 | 595 | 0 | 7667 | 0 |

**Variable type: numeric**

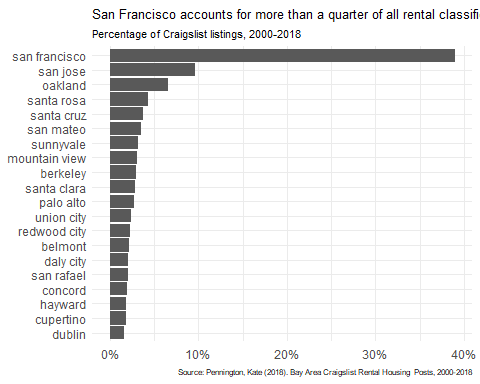
| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| date | 0 | 1.00 | 20095718.38 | 44694.07 | 20000902.00 | 20050227.00 | 20110924.00 | 20120805.0 | 20180717.00 | ▁▇▁▆▃ |
| year | 0 | 1.00 | 2009.51 | 4.48 | 2000.00 | 2005.00 | 2011.00 | 2012.0 | 2018.00 | ▁▇▁▆▃ |
| price | 0 | 1.00 | 2135.36 | 1427.75 | 220.00 | 1295.00 | 1800.00 | 2505.0 | 40000.00 | ▇▁▁▁▁ |
| beds | 6608 | 0.97 | 1.89 | 1.08 | 0.00 | 1.00 | 2.00 | 3.0 | 12.00 | ▇▂▁▁▁ |
| baths | 158121 | 0.21 | 1.68 | 0.69 | 1.00 | 1.00 | 2.00 | 2.0 | 8.00 | ▇▁▁▁▁ |
| sqft | 136117 | 0.32 | 1201.83 | 5000.22 | 80.00 | 750.00 | 1000.00 | 1360.0 | 900000.00 | ▇▁▁▁▁ |
| room\_in\_apt | 0 | 1.00 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 | 0.0 | 1.00 | ▇▁▁▁▁ |
| lat | 193145 | 0.04 | 37.67 | 0.35 | 33.57 | 37.40 | 37.76 | 37.8 | 40.43 | ▁▁▅▇▁ |
| lon | 196484 | 0.02 | -122.21 | 0.78 | -123.20 | -122.42 | -122.26 | -122.0 | -74.20 | ▇▁▁▁▁ |

Make a plot that shows the top 20 cities in terms of % of classifieds between 2000-2018. You need to calculate the number of listings by city, and then convert that number to a %.

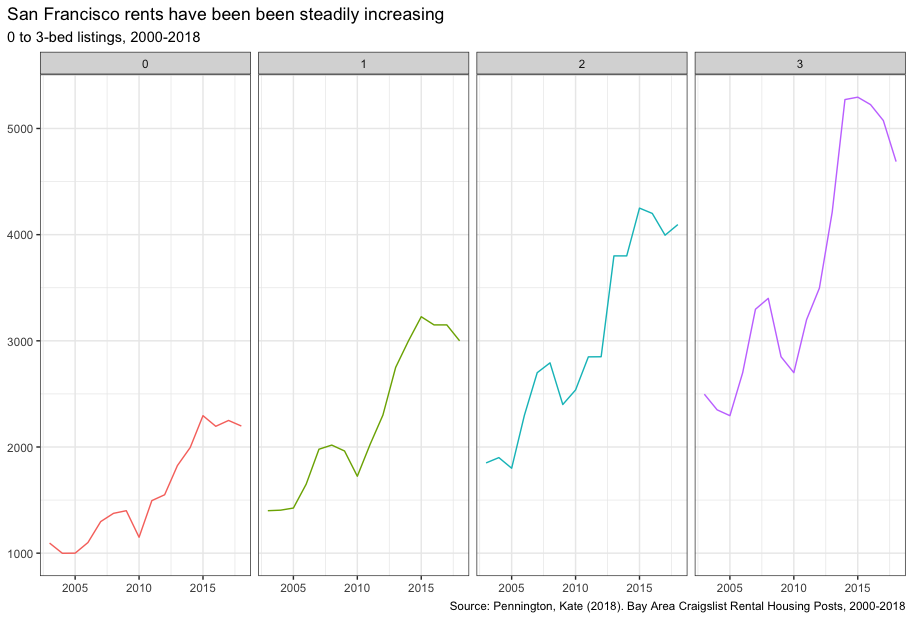
The final graph should look like this



# YOUR CODE GOES HERE  
rent %>%   
 count(city, sort = TRUE) %>%   
 head(20) %>%   
 mutate(  
 pct = n / sum(n),  
 city = fct\_reorder(city, pct)  
 ) %>%   
   
 ggplot() +  
 aes(x = pct, y = city) +  
 geom\_col() +  
 scale\_x\_continuous( labels = scales::percent\_format()) +  
 labs(  
 title = "San Francisco accounts for more than a quarter of all rental classifieds",  
 subtitle = "Percentage of Craigslist listings, 2000-2018",  
 caption = "Source: Pennington, Kate (2018). Bay Area Craigslist Rental Housing Posts, 2000-2018",  
 x = NULL, # Removes the 'pct' label on x-axis  
 y = NULL # Removes the 'city' label on y-axis  
 ) +  
 theme\_minimal() +  
 theme(  
 plot.title = element\_text(size = 10, hjust = 0),  
 plot.subtitle = element\_text(size = 8, hjust = 0),  
 plot.caption = element\_text(size = 6)  
 )

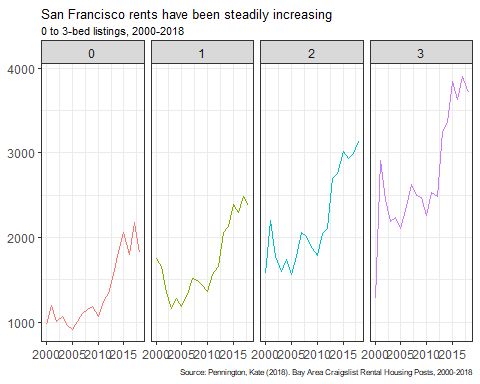


Make a plot that shows the evolution of median prices in San Francisco for 0, 1, 2, and 3 bedrooms listings. The final graph should look like this

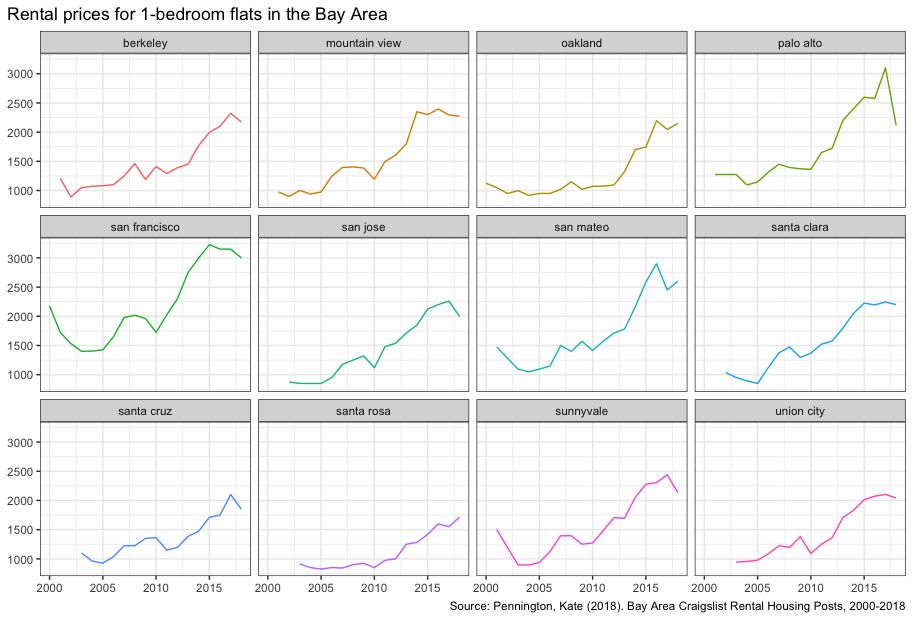


# YOUR CODE GOES HERE  
rent %>%   
 group\_by(beds, year) %>%   
 summarise(median\_price = mean(price, na.rm = TRUE)) %>%   
 filter(beds %in% c(0,1,2,3)) %>%   
 ggplot() +  
 aes(x = year, y = median\_price, color = as.factor(beds)) +  
 geom\_line() +  
 facet\_wrap(~beds, ncol = 4) +  
 theme\_bw() +  
 labs(  
 title = 'San Francisco rents have been steadily increasing',  
 subtitle = '0 to 3-bed listings, 2000-2018',   
 caption = 'Source: Pennington, Kate (2018). Bay Area Craigslist Rental Housing Posts, 2000-2018',   
 x = NULL,   
 y = NULL  
 ) +  
 theme(  
 legend.position = "none",  
 plot.title = element\_text(size = 10, hjust = 0, margin = margin(t = 0, b = 3, l = 0)),  
 plot.subtitle = element\_text(size = 8, hjust = 0, margin = margin(t = 0, b = 3, l = 0)),  
 plot.caption = element\_text(size = 6)  
 )

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

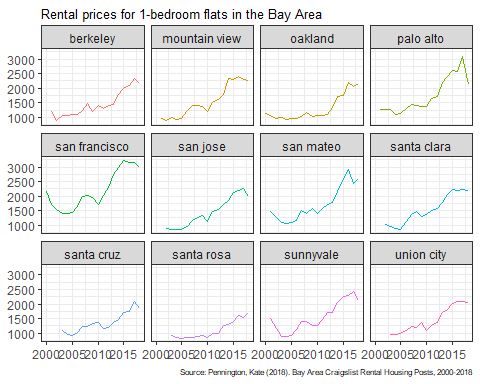


Finally, make a plot that shows median rental prices for one-bed flats in the top 12 cities in the Bay area. Your final graph should look like this



# YOUR CODE GOES HERE  
bay\_area = c('berkeley', 'mountain view', 'oakland', 'palo alto', 'san francisco', 'san jose', 'san mateo', 'santa clara', 'santa cruz', 'santa rosa', 'sunnyvale', 'union city')  
  
rent %>%   
 filter(city %in% bay\_area, beds == 1) %>%   
 group\_by(city, year) %>%   
 summarise(median\_price = median(price)) %>%   
 ggplot() +  
 aes(x = year, y = median\_price, color = city) +   
 geom\_line() +  
 facet\_wrap(~city, ncol = 4) +  
 theme\_bw() +  
 labs(  
 title = 'Rental prices for 1-bedroom flats in the Bay Area',  
 caption = 'Source: Pennington, Kate (2018). Bay Area Craigslist Rental Housing Posts, 2000-2018',   
 x = NULL,   
 y = NULL  
 ) +  
 theme(  
 legend.position = "none",  
 plot.title = element\_text(size = 10, hjust = 0, margin = margin(t = 0, b = 3, l = 0)),  
 plot.subtitle = element\_text(size = 8, hjust = 0, margin = margin(t = 0, b = 3, l = 0)),  
 plot.caption = element\_text(size = 6)  
 )

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



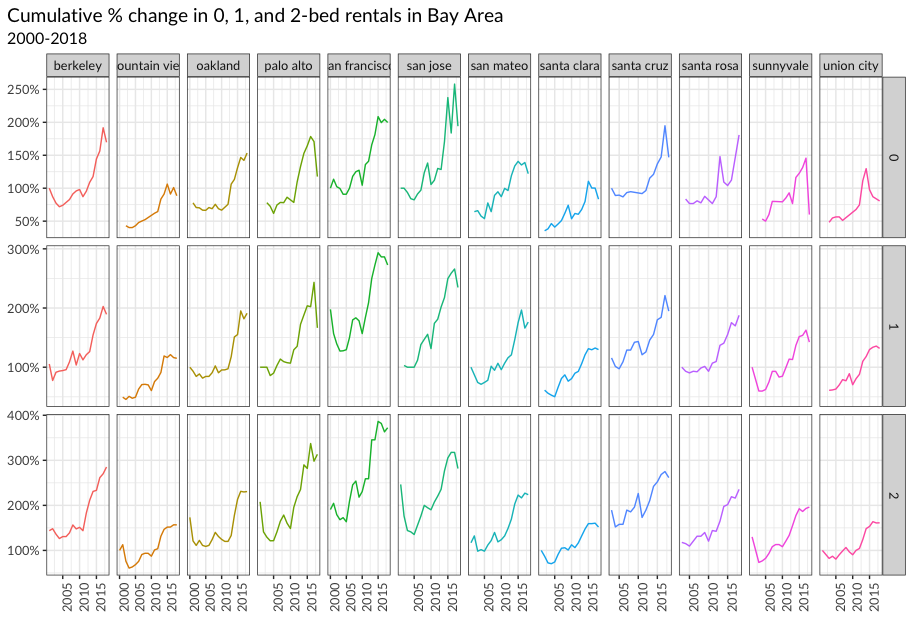
What can you infer from these plots? Don’t just explain what’s in the graph, but speculate or tell a short story (1-2 paragraphs max).

With over 25% of all Craigslist rental listings between 2000 and 2018, San Francisco is evidently the dominant rental market in the Bay Area, according to the visualizations. This significant presence points to a very active rental market, probably caused by strong demand and a steady stream of people looking for property in the city—possibly as a result of its reputation as a center of culture and technology.  
The second plot emphasizes a clear rising trend from 2000 to 2018, showing a consistent rise in rental prices across different numbers of bedrooms. This pattern highlights San Francisco’s rising cost of living and reflects the city’s larger problems with gentrification and economic expansion.

The final plot, which compares rental costs for one-bedroom apartments in various locations, shows that although growth occurred in all of the cities, it did so at different rates and with variable consistency. Certain cities, like Oakland and Palo Alto, exhibit greater volatility, whereas cities like San Francisco and San Jose show steady and dramatic increases, indicating ongoing demand. This suggests that economic pressures and housing demand vary by geography, with certain cities experiencing more acute pressures from the housing market than others.

## Challenge

How would you go about creating this?

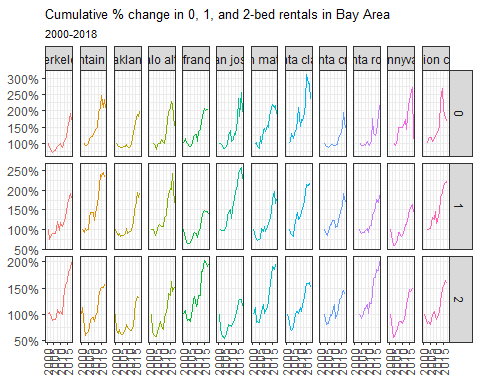


TYPE YOUR ANSWER AFTER (AND OUTSIDE!) THIS BLOCKQUOTE.

median\_rent <- rent %>%   
 filter(city %in% bay\_area, beds %in% c(0,1,2)) %>%   
 group\_by(city, beds, year) %>%  
 summarise(median\_price = median(price, na.rm = TRUE)) %>%   
 mutate(  
 pct\_change = (median\_price / first(median\_price))  
 )

## `summarise()` has grouped output by 'city', 'beds'. You can override using the  
## `.groups` argument.

ggplot(median\_rent, aes(x = year, y = pct\_change, color = city)) +  
 geom\_line() + # Draw lines  
 facet\_grid(beds ~ city, scales = "free\_y") + # Facet by city and beds, free y-axis scaling  
 scale\_y\_continuous(labels = scales::percent\_format()) +  
 labs(  
 title = "Cumulative % change in 0, 1, and 2-bed rentals in Bay Area",  
 subtitle = "2000-2018",  
 x = NULL,  
 y = NULL  
 ) +  
 theme\_bw() + # Use a minimal theme  
 theme(  
 plot.title = element\_text(hjust = 0, size = 10),   
 plot.subtitle = element\_text(hjust = 0, size = 8),  
 axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust = 1),  
 legend.position = "none" # Hide legend  
 )



# 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')

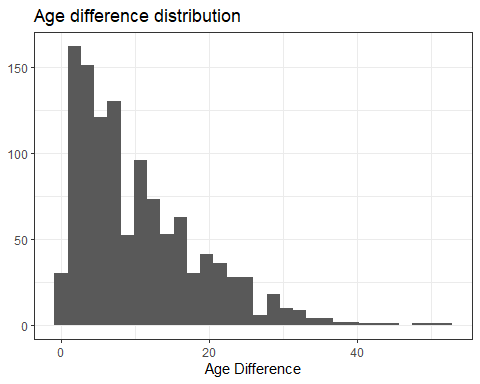
## Rows: 1155 Columns: 13  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): movie\_name, director, actor\_1\_name, actor\_2\_name, character\_1\_gend...  
## dbl (5): release\_year, age\_difference, couple\_number, actor\_1\_age, actor\_2\_age  
## 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.

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?

age\_gaps %>%   
 ggplot() +  
 aes(x = age\_difference) +  
 geom\_histogram() +  
 theme\_bw() +  
 labs(  
 x = 'Age Difference',  
 y = NULL,  
 title = 'Age difference distribution'  
 )

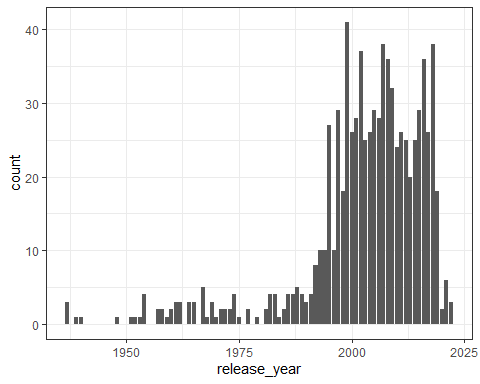
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



* 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?

age\_diff\_followed <- age\_gaps %>%  
 mutate(  
 min\_acceptable\_age = actor\_1\_age / 2 + 7,  
 max\_acceptable\_age = (actor\_1\_age - 7) \* 2,  
 follows\_rule = actor\_2\_age >= min\_acceptable\_age & actor\_2\_age <= max\_acceptable\_age  
 )  
age\_diff\_followed %>%   
 filter(follows\_rule == TRUE) %>%   
 group\_by(release\_year) %>%   
 ggplot() +  
 aes(x = release\_year) +  
 geom\_bar() +   
 theme\_bw()



# Summarize the results  
summary <- age\_diff\_followed %>%  
 summarise(  
 total\_couples = n(),  
 couples\_following\_rule = sum(follows\_rule),  
 proportion\_following\_rule = mean(follows\_rule)  
 )  
  
print(summary)

## # A tibble: 1 × 3  
## total\_couples couples\_following\_rule proportion\_following\_rule  
## <int> <int> <dbl>  
## 1 1155 829 0.718

* Which movie has the greatest number of love interests?

movie\_couples <- age\_gaps %>%  
 group\_by(movie\_name) %>%  
 summarise(love\_interests = n())  
  
# Find the movie with the greatest number of love interests  
movie\_with\_most\_love\_interests <- movie\_couples %>%  
 arrange(desc(love\_interests)) %>%  
 slice(1)  
  
# Print the result  
print(movie\_with\_most\_love\_interests)

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

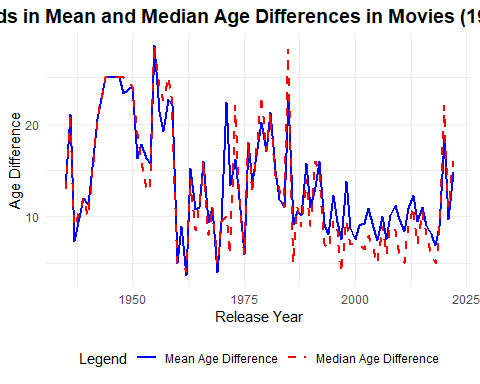
* Which actors/ actresses have the greatest number of love interests in this dataset?

# Combine actor\_1 and actor\_2 into a unified format for counting love interests  
love\_interests <- age\_gaps %>%  
 select(actor\_1\_name, actor\_2\_name, movie\_name, couple\_number) %>%  
 gather(key = "actor\_role", value = "actor\_name", actor\_1\_name, actor\_2\_name) %>%  
 distinct(actor\_name, movie\_name, couple\_number)  
  
# Group by actor\_name and count the number of unique love interests  
actor\_love\_interests <- love\_interests %>%  
 group\_by(actor\_name) %>%  
 summarise(love\_interests = n()) %>%  
 arrange(desc(love\_interests))  
  
# Find the actors with the greatest number of love interests  
actors\_with\_most\_love\_interests <- actor\_love\_interests %>%  
 filter(love\_interests == max(love\_interests))  
  
# Print the result  
print(actors\_with\_most\_love\_interests)

## # A tibble: 1 × 2  
## actor\_name love\_interests  
## <chr> <int>  
## 1 Keanu Reeves 27

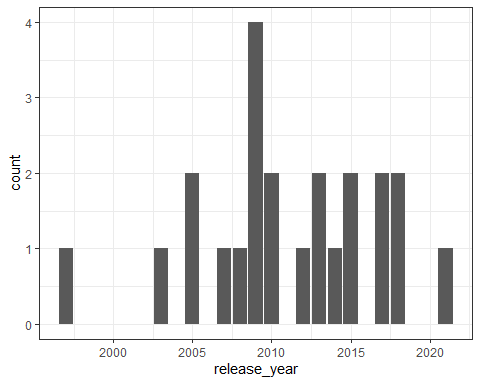
* Is the mean/median age difference staying constant over the years (1935 - 2022)?

# Filter the data for the years 1935 to 2022  
age\_gaps\_filtered <- age\_gaps %>%  
 filter(release\_year >= 1935 & release\_year <= 2022)  
  
# Calculate mean and median age differences for each year  
age\_difference\_trends <- age\_gaps\_filtered %>%  
 group\_by(release\_year) %>%  
 summarise(  
 mean\_age\_difference = mean(age\_difference, na.rm = TRUE),  
 median\_age\_difference = median(age\_difference, na.rm = TRUE)  
 )  
  
# Plot the results  
ggplot(age\_difference\_trends, aes(x = release\_year)) +  
 geom\_line(aes(y = mean\_age\_difference, color = "Mean Age Difference"), linewidth = 1) +  
 geom\_line(aes(y = median\_age\_difference, color = "Median Age Difference"), linewidth = 1, linetype = "dashed") +  
 labs(  
 title = "Trends in Mean and Median Age Differences in Movies (1935-2022)",  
 x = "Release Year",  
 y = "Age Difference",  
 color = "Legend"  
 ) +  
 theme\_minimal() +  
 scale\_color\_manual(values = c("Mean Age Difference" = "blue", "Median Age Difference" = "red")) +  
 theme(  
 plot.title = element\_text(hjust = 0.5, size = 14, face = "bold"),  
 legend.position = "bottom"  
 )



* How frequently does Hollywood depict same-gender love interests?

age\_gaps\_same\_gender <- age\_gaps %>%  
 mutate(  
 same\_gender = character\_1\_gender == character\_2\_gender  
 )  
age\_gaps\_same\_gender %>%   
 group\_by(release\_year) %>%  
 filter(same\_gender == TRUE) %>%   
 ggplot() +  
 aes(x = release\_year) +  
 geom\_bar() +  
 theme\_bw()



# Calculate the frequency of same-gender couples  
same\_gender\_summary <- age\_gaps\_same\_gender %>%  
 summarise(  
 total\_couples = n(),  
 same\_gender\_couples = sum(same\_gender, na.rm = TRUE),  
 proportion\_same\_gender = mean(same\_gender, na.rm = TRUE)  
 )  
  
# Print the summary  
print(same\_gender\_summary)

## # A tibble: 1 × 3  
## total\_couples same\_gender\_couples proportion\_same\_gender  
## <int> <int> <dbl>  
## 1 1155 23 0.0199

# 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: N/A
* Approximately how much time did you spend on this problem set: ~5 hours
* What, if anything, gave you the most trouble: Recreating the graphs, mainly formatting the graphs as per the image. I could not get the title and subtitle to align correctly.

**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?

Yes

# 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.