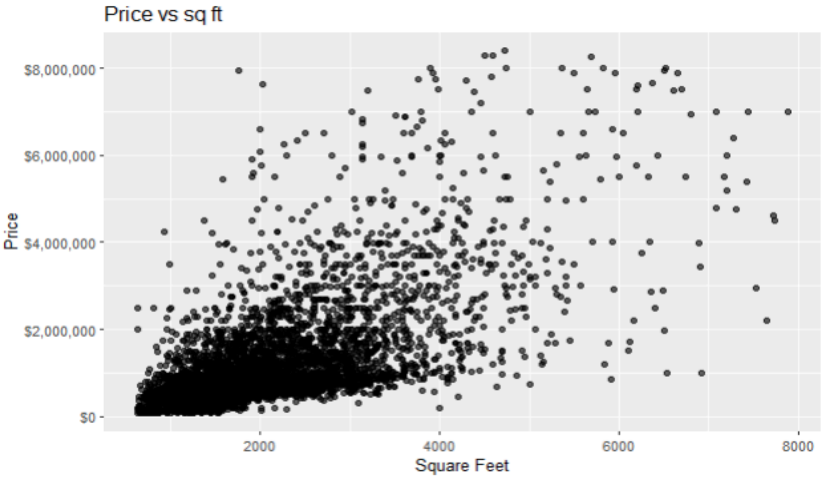
Rohan Pillay
920674736
STA 141B A01
Due: June 12, 10pm

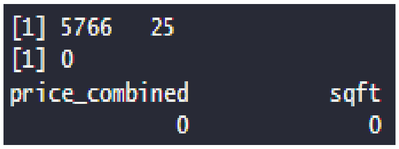
Assignment 6 Interactive Data Visualization

Our main goal this assignment is to build an interactive HTML page with linked visualizations of real estate data from 8 counties that are near Davis, CA. The data is stored on a RDS file, which we need to clean first.

The raw dataset contains multiple property listings scraped from numerous counties near Davis, and since the data could contain inconsistencies, outliers, and NA or missing values, we first need to clean it. By doing this, the data we use won't negatively affect the visualizations and analysis. To avoid double-counting, I first removed duplicated rows by using distinct(), which makes sure that each property is only represented once. I used %>% (and did multiple times during the cleaning process), which is the pipe operator in R, which made my code easier to write, more readable, and just meant taking the thing on the right, and passing it to the function on the right. I created a singular column that was the price_combined since there were two columns related to price. This new column used price and then when needed to, used price.1. I then removed outliers by using percentiles. I first used a strict IQR rule, but I changed it to only remove the top and bottom 1% of listings based on price_combined and sqft. By doing this, I was able to reduce the influence that extreme value or listings would have on our future plots and visualizations. I decided to split into subsets for visualizations. Using full_data, which is all cleaned listings, I only chose the listings with valid geographical coordinates. I also made sure that all the labels were unified. I used trimws() to remove any extra spaces, tolower() to ensure capitalization was uniform throughout, and I then used toTitleCase() to capitalize the first letter of each word. I checked the data rows and map rows, and saved the two .rds files. I created a scatterplot of price vs. square footage to see the relationship of the cleaned data. I also did a quick check to see the dimensions of what I was working with, and if there were any missing or NA values.

Figure 1 and 2:





```
[1] 5766   25
[1] 0
price_combined           sqft
             0              0
```

From the scatterplot, we see that there is a clear upward trend, that as the square footage of a house increases. Along with this, we also observe a considerable spread, notably among the larger homes, which shows the location, property type, amenities, and market in the area among other factors heavily influencing price. This checks out with how the housing market is as price normally increases with the bigger the house, and also there is always variability. The percentile-based filtering removed unrealistic outliers, but was still able to show the variation in the housing market. Along with that we see that there weren't any missing or NA values.

I then moved on to creating the interactive dashboard. I first added an introduction section to go at the top of the page containing information about what the user is seeing, how to interact with the plots and data table, and also a description of an insight they could look into. I used the plotly, crosstalk, DT, and dplyr libraries in R. Our main goal was to create an interactive exploration dashboard of the housing data by connecting geographic locations with price and property size.

I first loaded the cleaned property data. I then made sure to only select the main main columns that I wanted to see in the final plot. Along with that I made sure the filtering was done, and handled potential missing values in type.1 or property type by assigning unknown to NA values. By assigning unknown instead of dropping these missing values, it preserves as much of the data as possible. Along with this, even if a property was missing its type, it still has valuable info such as price, size, or location. By labeling this as unknown, it ensures consistent plot behavior. I then made the property type column into a factor, which I defined custom colors for so it was easily distinguishable in both plots.

The map I built using plotly with type = "scattermapbox". Each property is displayed as a point on an interactive map of Northern California and is color-coded by the property type (single-family, townhouse, mobile, etc.). Hovering over a point shows the address, price, and a link to the property listing as well. When creating the text, for both plots, <b> starts the bold text while </b> ends the bold text. <br> creates a line break and <a href = 'URL'> starts the beginning of the clickable link while </a> is the end of the clickable link. Then, target = '_blank' opens the link in a new tab. The listing link when you hover over a point is quite hard to click. This is due to plotly hover tooltips being dynamic and disappearing when the cursor leaves the plot area. Due to this, users should mainly use the interactive table to open the listings more reliably.

The interactive scatter plot shows users to explore how the square footage of a house relates to the overall price. Each point represents a property and similar to the map not only are the points color-coded, but hovering over one, provides the same info (address, price, and link to property listing). For both the map and the scatterplot, the user can click a point on the legend, and that property type won't show on the map. This is useful as when looking at both the plots, we see that a majority of the dataset includes single-family homes. What this means is that it takes over quite a bit and you mostly see only blue points (single-family). If a user wants to explore other property types, they can just click single family on the legend, and viewing other properties will be much more visible. Along with this I made sure for both plots that when a user clicks a different point/property the previous point becomes unhighlighted. Both plots retain previously dimmed points after a user clicks one due to plotly with crosstalk highlighting the selected point without fully clearing the previous section visually. This still ensures though that the linked table updates with each click, and that the previous point is unhighlighted.

Next, using the DT library, this displayed the full property dataset allowing the user to search and scroll through. It is made in a way so that when a point is clicked on either of the two plots it filters in the datatable to show that listing. It is synchronized with the two plots through the SharedData object. The table shows details including price, size, address, type, etc. You can choose how many entries you want to see and you can search for a property as well. The number of entries/pages, etc are shown at the bottom of the table as well.

I listed certain things such as listing links, the unhighlighted points, and made sure to explain why these occur the way they do. I also made sure to give the users instructions, an overview, and something to look for when viewing. Overall, the dashboard is made to be simple, user friendly, and distinguishable. It makes it much easier to look for homes keeping in mind many factors. One can compare price with sq ft, find the actual listing of the house, see its geographical location, the type of house, all with a single click.

References

I first tried creating the interactive plot without creating the SharedData object and was overcomplicating it, so claude.ai suggested creating shared interactivity using a shared object.

One technique that claude.ai suggested was using %>% which is the pipe operator which pretty much means taking the thing on the left to pass to the function on the right. This made my code much more readable.

To create the labels, I used claude.ai to see how to bold, start a clickable link text, etc. They give me suggestions such as <b>, <a href='", and target='_blank', which I explained in my write up.