Ryan Andrews

Some Statistics on Sold Price/SqFt Study Retrospective of 2006-2023

Introduction

In the realm of real estate, understanding market trends is vital for strategic decision-making. As a budding college student delving into the intricacies of housing dynamics, I present an in-depth statistical analysis focused on Trilogy La Quinta, a gem nestled in the heart of California's La Quinta. Bennion Deville Homes, a prominent brokerage, serves as a leading real estate service into this vibrant community.

Through meticulous examination of property prices, days on market, and housing inventory, we uncover nuanced insights into Trilogy La Quinta's real estate landscape. By applying advanced statistical techniques and visualization tools, we aim to equip Bennion Deville Homes with insights to navigate the dynamic market.

This report reflects my dedication to empirical exploration and data-driven decision-making, empowering real-estate agents to seize opportunities and address challenges in Trilogy La Quinta's ever-evolving real estate terrain.

Data Collection

Data was collected from Software on JohnKevinMiller.com and presented to me by Mark Miller. The data, while not perfect, it still allowed me to delve into its intricacies and do analysis on variables that were collected properly. Unfortunately lots of data had to be removed from the analysis due to incomplete entries but in some cases I was able to impute values from statistical techniques that predict entries with maximum accuracy.

The data itself had 1200 entries, and by the time I had filtered and cleaned the data it had 350. It's important to remember that computational statistics only works if you have data that can be interpreted by software. As the old saying goes “garbage in, garbage out”, which is why it’s paramount that the data is cleaned and managed properly.

Figures and Findings

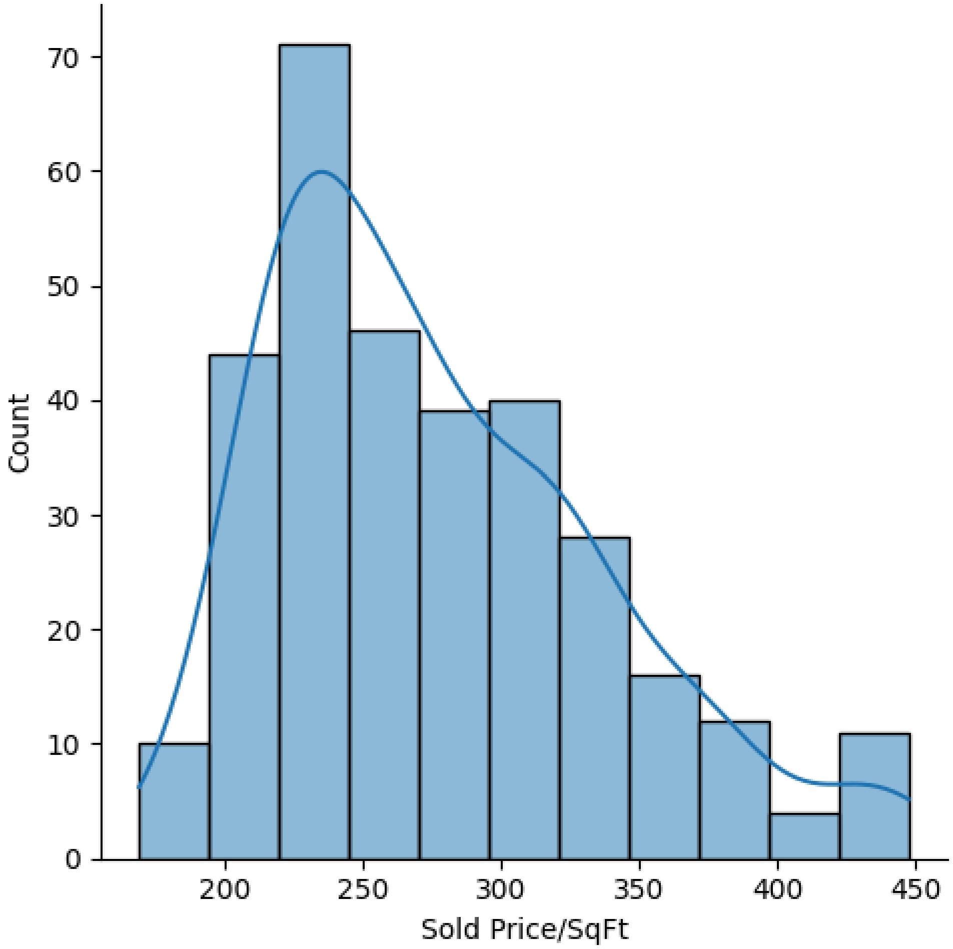


Figure 1.0: Sold Price/Sqft Count data

Here in the graph we can observe the highest amount of properties sold have an average of around $250 Per Sqft.

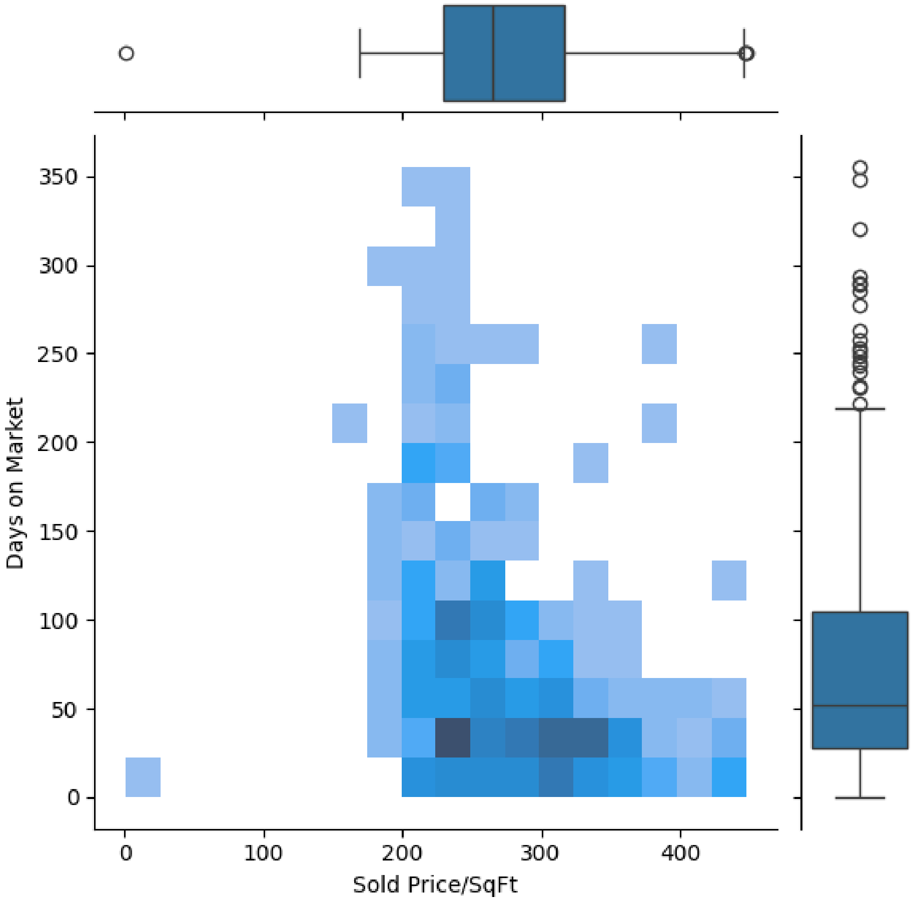


Figure 1.1: Sold Price/SqFt vs. Days on Market

Its clear that if you know what box plots are you can observe the amount of outliers, particularly in the Days on Market Axis. However we should take note that the majority of data says that Homes with about $225/sqft stay on the market for an average of 50 days (Darker values indicate higher frequency)

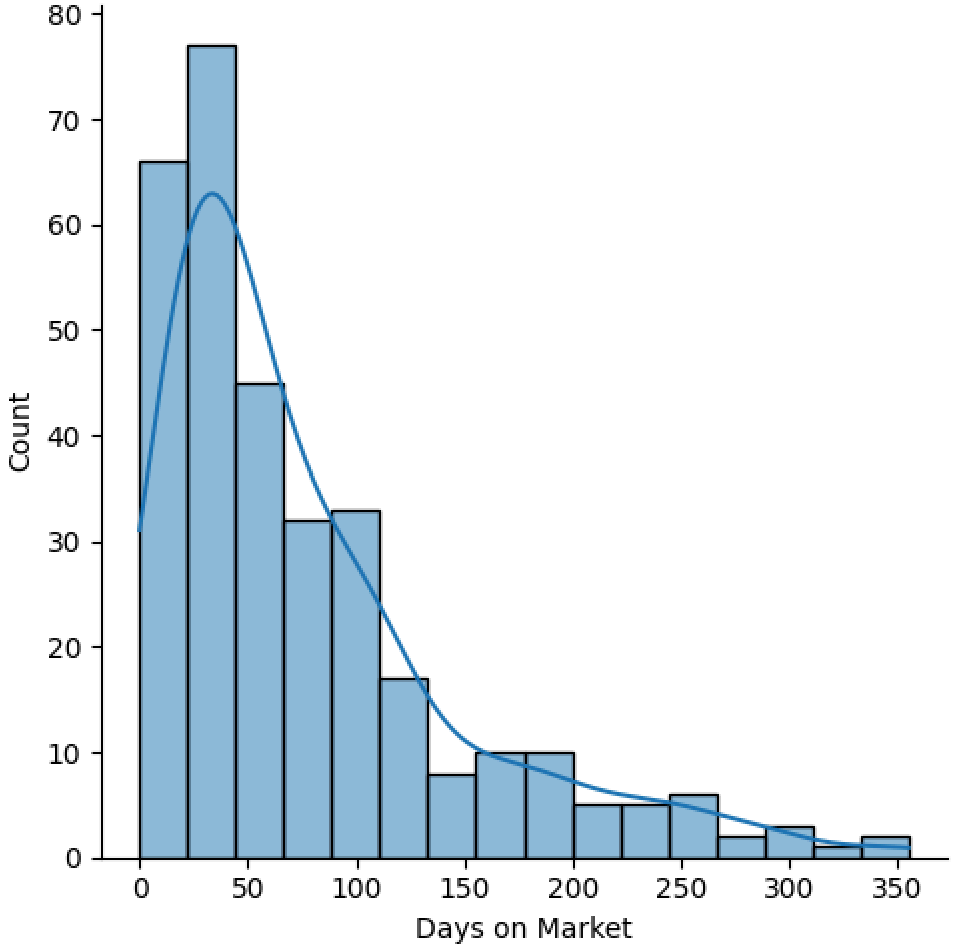


Figure 1.3: Days on Market Frequency

It should be no surprise that this graph follows a Right skewed distribution of some sort. This graph states that it is rare that homes stay on the market for intervals of above 200 days.

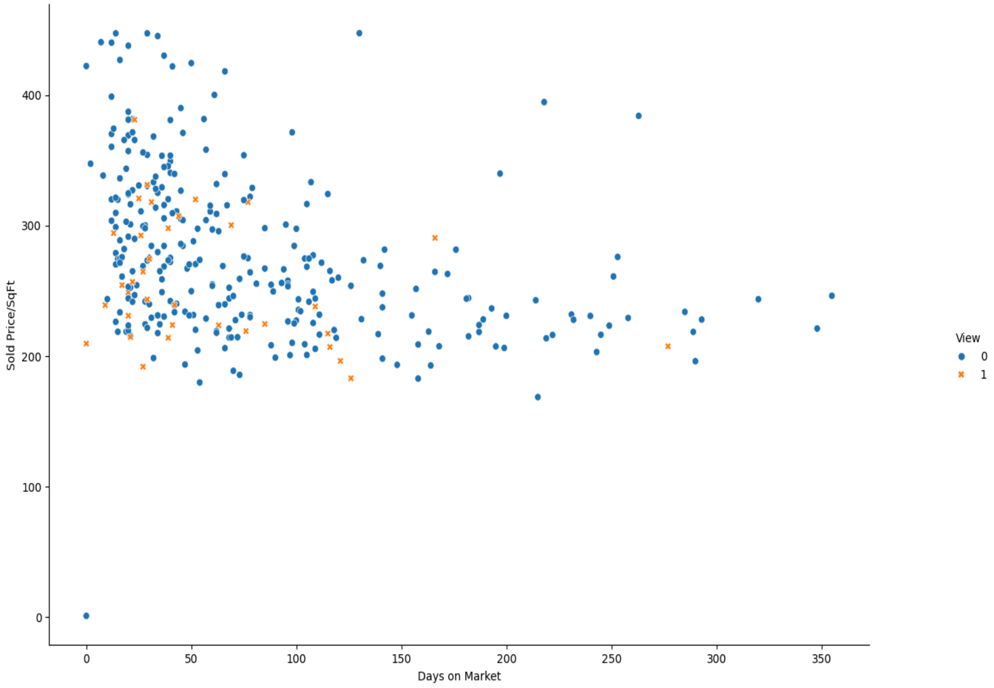


Figure 1.4 Days on market vs Sold Price per sqft with View as its Hue:

In this plot, its indicating that, The higher the sold price/sqft, the shorter days on average it is on the market. It also indicated that majority of the sales happen within 100 days of the property being on the market.

*note: The orange labels indicate if the property has a view. On the right side of the graph we had a categorical variable that had to be converted into a binary value so equivalently,

0= no, 1= yes. Im also not sure What view means here, but it seems the majority of properties that have a view are statistically Biased around $300/sqft

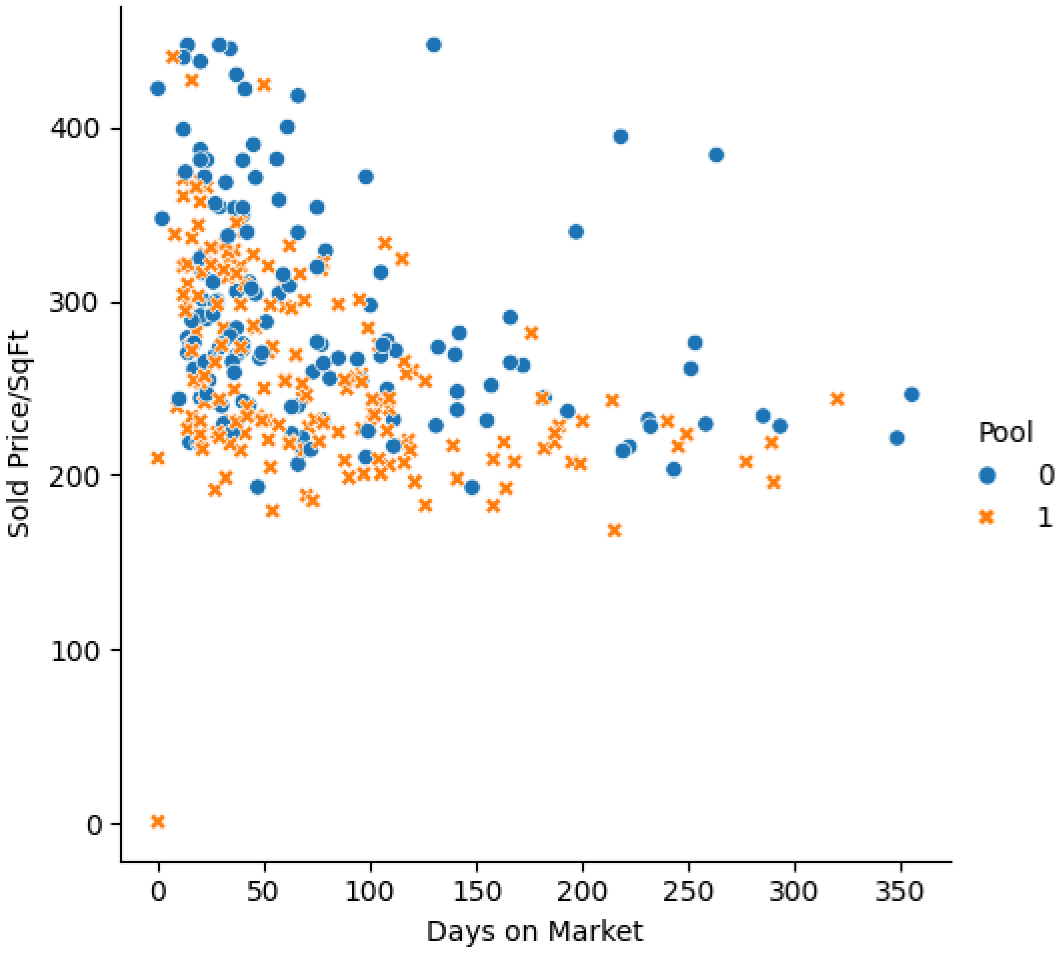


Figure 1.5: Days on Market vs Sold Price/sqft with a pool as its hue.

We can see 1 outlier on the bottom. Its the same plot as before but now we're paying attention to Pool as the Hue. As before data is skewed right however it seems that theres alot less bias with the pool variable But with the trade off of more variance. As stated before, Pool is a categorical Variable so it needed to be factored into a binary variable and thus 0= No, 1= Yes

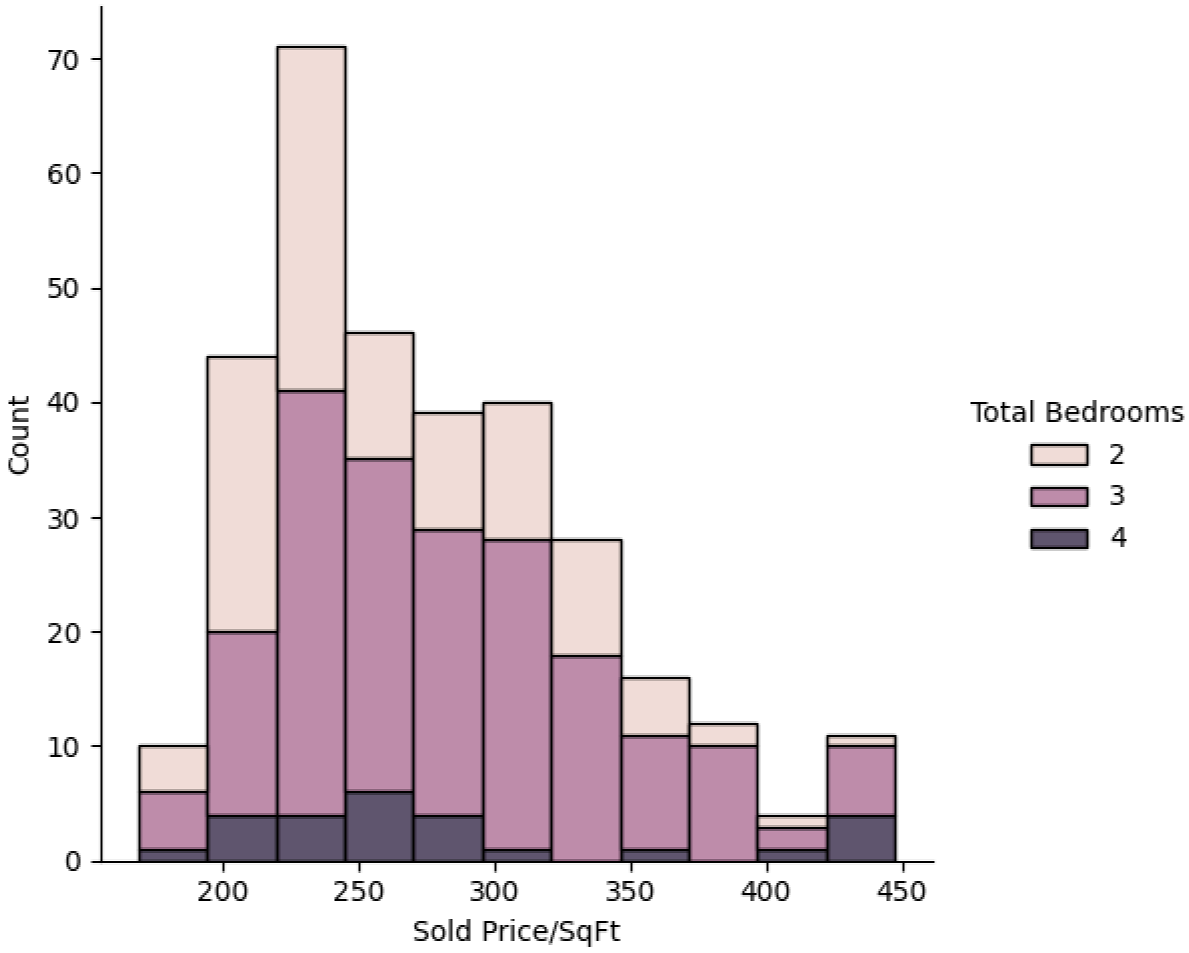


Figure 1.6: Sold Price/SqFt frequency with Bedroom as the hue

By observing the graph we can deduce that across all instances of Price/SqFt that 2 bedrooms is the most frequent.

Notes:

With the current data it is near impossible to build more complex statistical analysis. Things such as predictive models, Principle component Analysis, Machine Learning models And computing Error Analysis. Due to the limitations of the data given. This is about as far as you can go unless of course you want to see the other metrics. Metrics Such as Summary Statistics to computer Means, Variance, Quantiles, Quartiles, etc. (Please Inquire if you want them, i can promise you won't learn much but still may be useful to know)

Acknowledgements

I want to thank Mark Miller for the data set and explaining real estate jargon that i don't understand

Special Thanks to John and Jim for guiding me all these years and putting up with my shenanigans. May you guys find fortune and exceptional happiness in the coming years.

References:

Python documentation

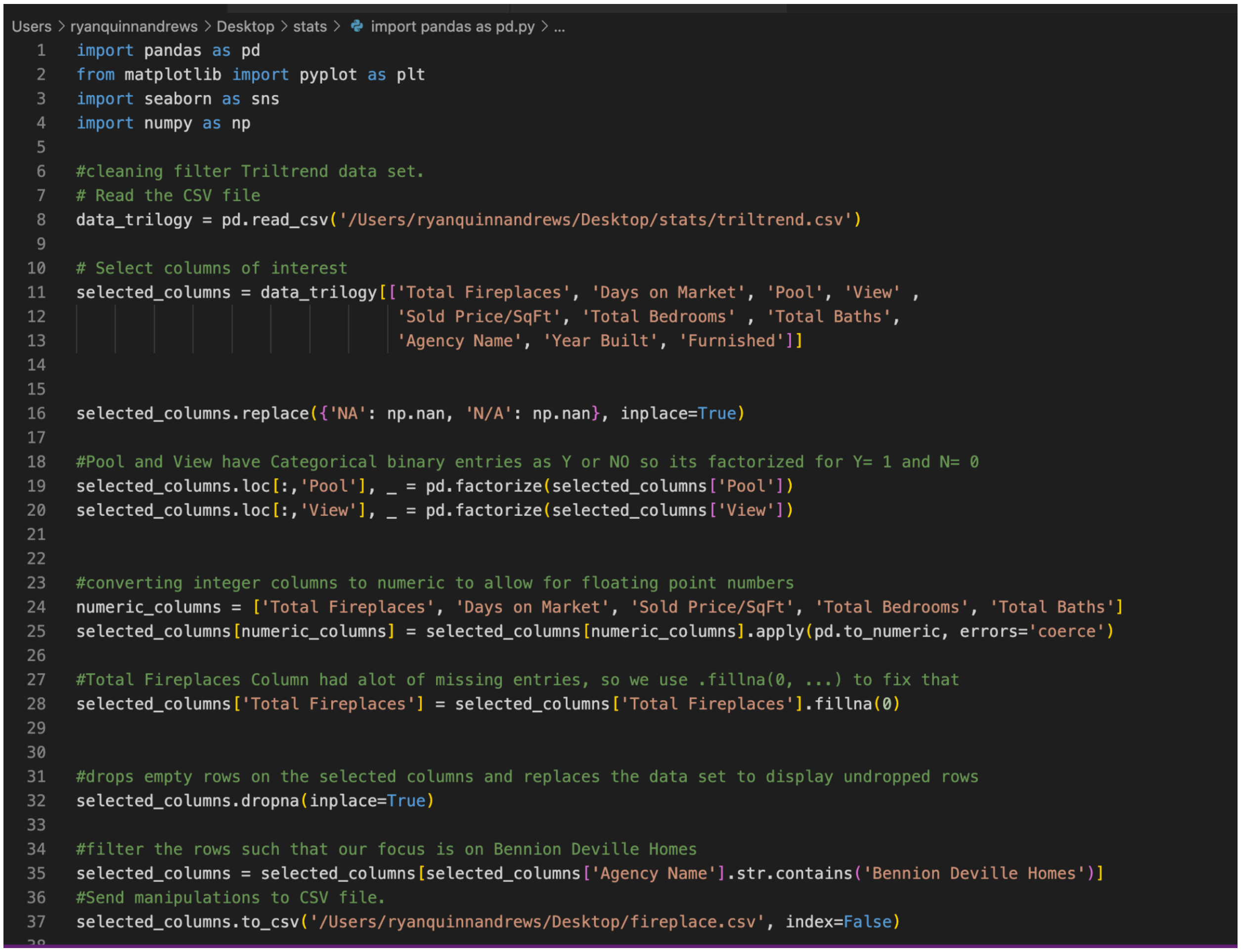
1.https://docs.python.org/3/library/stdtypes.html

Graphical Interface

2.https://seaborn.pydata.org/api.html

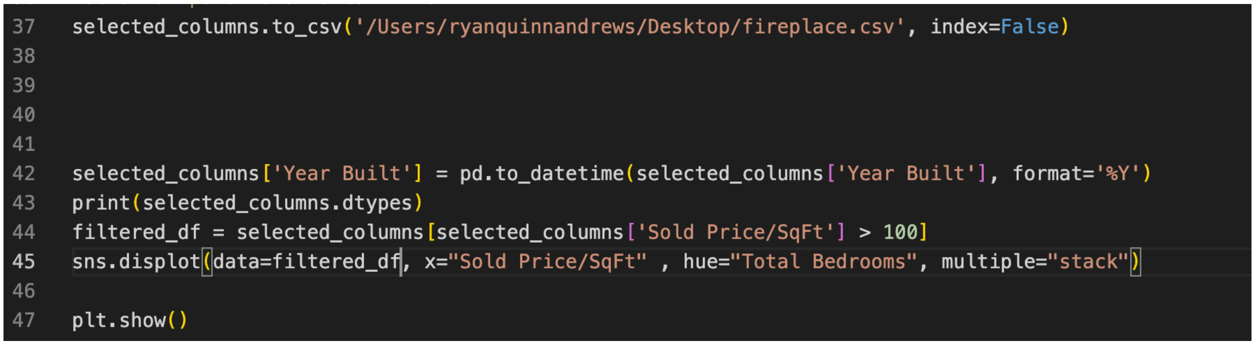
Packages used: Pandas, Numpy,Seaborn, Matplotlib

Code:



Users > ryanquinnandrews > Desktop > stats > import pandas as pd.py > ...

```python
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import numpy as np

#cleaning filter Triltrend data set.
# Read the CSV file
data_trilogy = pd.read_csv('/Users/ryanquinnandrews/Desktop/stats/triltrend.csv')

# Select columns of interest
selected_columns = data_trilogy[['Total Fireplaces', 'Days on Market', 'Pool', 'View' ,
                                 'Sold Price/SqFt', 'Total Bedrooms' , 'Total Baths',
                                 'Agency Name', 'Year Built', 'Furnished']]


selected_columns.replace({'NA': np.nan, 'N/A': np.nan}, inplace=True)

#Pool and View have Categorical binary entries as Y or NO so its factorized for Y= 1 and N= 0
selected_columns.loc[:,'Pool'], _ = pd.factorize(selected_columns['Pool'])
selected_columns.loc[:,'View'], _ = pd.factorize(selected_columns['View'])


#converting integer columns to numeric to allow for floating point numbers
numeric_columns = ['Total Fireplaces', 'Days on Market', 'Sold Price/SqFt', 'Total Bedrooms', 'Total Baths']
selected_columns[numeric_columns] = selected_columns[numeric_columns].apply(pd.to_numeric, errors='coerce')

#Total Fireplaces Column had alot of missing entries, so we use .fillna(0, ...) to fix that
selected_columns['Total Fireplaces'] = selected_columns['Total Fireplaces'].fillna(0)


#drops empty rows on the selected columns and replaces the data set to display undropped rows
selected_columns.dropna(inplace=True)

#filter the rows such that our focus is on Bennion Deville Homes
selected_columns = selected_columns[selected_columns['Agency Name'].str.contains('Bennion Deville Homes')]
#Send manipulations to CSV file.
selected_columns.to_csv('/Users/ryanquinnandrews/Desktop/fireplace.csv', index=False)
```



```python
selected_columns.to_csv('/Users/ryanquinnandrews/Desktop/fireplace.csv', index=False)




selected_columns['Year Built'] = pd.to_datetime(selected_columns['Year Built'], format='%Y')
print(selected_columns.dtypes)
filtered_df = selected_columns[selected_columns['Sold Price/SqFt'] > 100]
sns.displot(data=filtered_df, x="Sold Price/SqFt" , hue="Total Bedrooms", multiple="stack")

plt.show()
```

Note* i recreated each figure (ie. 1.0,1.1,1.2,etc) on the fly so the code wouldnt be 200+ lines long. This particular code displays figure 1.6 if youre curious.