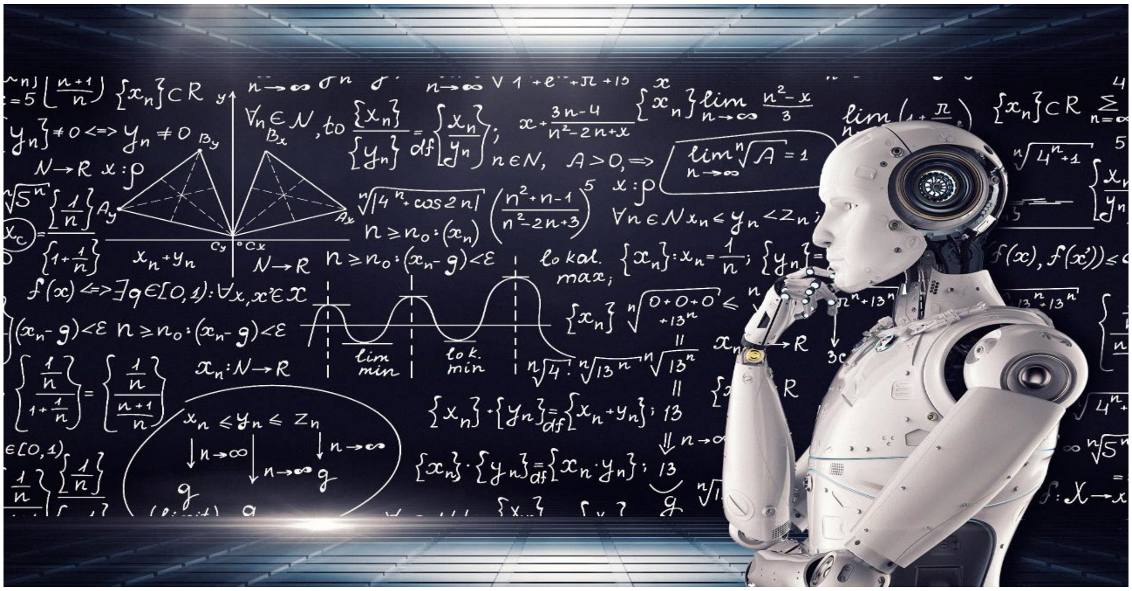

AAI-500: San Diego Home Sales

10.6.2022

Team 4

Trevor McGirr

Ike Ugwu

Adam Graves

(The Triumph Real Estate Brokerage-2022)

Links to Documentation:

Github Link to Model: https://github.com/tmcgirr/AAI-500-FA22-Project/blob/main/Multiple_Linear_Model_A%20.ipynb

Github Link to Model B: https://github.com/tmcgirr/AAI-500-FA22-Project/blob/main/Multiple_Linear_Model_B_Zipcode.ipynb

YouTube Presentation: https://youtu.be/-pS-U1ZES68

Table of Contents

Overview

Using a database of 300 San Diego Home listings we will:

- Load a dataset of 300 rows that have San Diego residential real estate listings
- Prepare the Extract, Transform, Load (ETL) process of the dataset
- Normalize the data (Identify the fields, their types, issues, fill in as needed, remove as needed)
- Identify the Independent Variables
- Identify the Dependent Variables
- Create a 70% Trainer set from data
- Categorize into area: Zip Codes
- Correlate the Zip to Price Per SF ($/SQUARE FEET)
- Run calculations of Mean, Standard Deviation, Coefficients, Lower and Upper Tails for all fields
- Identify the Outliers, and decide on impact to the dataset at whole
- Make decision if to remove outliers or keep them as legit values
- Create Graphs and Plots to visualize and verify data and patterns
- Based on information, evaluate any obstacles, discuss as needed. Plan on overcoming obstacles if any
- Calculate the Mean for property price
- Run a Predicted Price vs. Actual Price and evaluate the findings. If any issues loop back to dealing with obstacles
- Create a 95% Confidence Interval for property prices
- Test 20 listings from the 30% Test data set
- Document results, Acceptances or Rejects of CI
- Load a fresh set of data and validate if the numbers are in-line

Optional Phase 2

- Categorize Means and Standard Deviations by Zip Codes
- Create 95% Confidence Intervals per Zip Code
- Calculate the 25% lower tail per Zip Code

Goals/Strategize

1. How are we going to import this data?
    a. The dataset is in the form of a csv file. We need to normalize the data (ETL Process)
        i. Identify the fields
        ii. Categorize the fields
        iii. Drop any fields not used
        iv. Correct/normalize any findings
        v. Import csv file via Python
2. What are we looking to do with the data?
    a. Create the Independent Variables: LOCATION, BEDS, BATH, LOT SIZE, YEAR BUILT.
    b. Create the Dependent Variable: $/SQUARE FEET
    c. Create Curve and Scattered plots for visual verification of patterns and values
    d. Show $/SQUARE FEET by ZIP OR POSTAL CODE distribution
    e. Find the Mean for each field
    f. Find the Standard Deviation for each field
    g. Find the Lower and Upper tail values for each field
    h. Find the Mean for property price per square foot
    i. Find the Margin of Error for a 95% Confidence Interval
    j. Establish the range. (This is to be used by the brokers to base their BOV on)
    k. For optional Phase 2 create the same calculation patterns but breakdown to individual Zip codes. This will allow brokers to have a 95% Confidence Interval per Zip code, and the lower 25% value as a default price per square foot in case the potential property is rejected by the 95% CI.
3. How are we going to do this?
    a. We will import the Dataset
    b. We will then normalize the data based on the work done prior to identify requirements
    c. We will run Distribution formulas to set the data and test the data as we have identified
    d. We will create Plots to visualize the findings
    e. We will run regression testing on the trainer dataset
    f. We will run regression verification on the test dataset
    g. Complete documentation

   h. All will be done in Python code

4. What are the resources we will use?
   a. The Team 4 people
   b. Python
   c. Excel
   d. Web research as needed
5. Brainstorming on all issues
   a. (Adam): I suggest playing with the 'Price per SF'. There is a lot we can do with this: a. Find the zip code with the highest mean for this field. b. Find the sd for this zip code. c. Do an H0 hypothesis about any new houses coming to a new listing in this zip code will be at a certain per SF price range. We can create a second data set with newer data from what we have to reject or not the H0.
   b. This can all be done pretty easy with Python…I hope… and I can get a new data set for the H0 if we go that route.
   c. We will need to think about the regression testing …any thoughts? I was thinking of the Multi Linear, since we have multiple IV that impact the DV. ?
   d. Thinking we might bring in another Dataset with a newer home sales listing and that can be the dataset we run the H0 against to prove.???
6. Set timeline
   a. Twice to Three times a week meeting
   b. Start the Project Scope documentation: 10/6/2022
   c. Start meetings: 10/10/2022
   d. Dataset import: 10/12/2022
   e. Run Distribution codes: 10/16/2022
   f. Test Results and plots: 10/18/2022
   g. Finalize the code: 10/20/2022
   h. Test: 10/20/2022
   i. Finalize documentation and presentation: 10/22/2022
   j. Final submission: 10/24/2022
7. Set Tasks to team members
   a. Trevor: Coding and Team representative
   b. Ike: Research and Testing
   c. Adam: Strategizing, Documentation, and Research

Business Need

Triumph is a Southern California Real Estate Agency with seven offices in various cities. We are looking to open a new office in San Diego and to capture at least 20% of the current Market listings in the residential brokerage.

We have a successful working system which pairs types of properties with real estate agents based on their expertise and skill level.

Normally from our past experiences in different cities we work in, we have seen a normal distribution for home pricing based on their location, however there are clearly more expensive areas based on the location, as they say "Location, Location, Location", with some outliers.

Our system is based on a few logics:

1. Any lower tail outlier we assign to an analyzing team that specializes in distress properties to evaluate the reason for the low price, and then based on the finding the properties are paired to the agents that specialize in distress properties if that is the case, or to juniors in case it's just a low worth property.
2. The higher tail outliers are normally large to mansion style homes in which we have agents that specialize in these types of homes and they are immediately assigned to them. (These outliers are not many and are not our main focus)
3. The rest is where most of our work is concentrated on and where we need to work on our hypothesis which is: *"The price of a house per square foot has a direct relation to the zip code it is in, the bedroom, baths, lot size, and year it was built in."*
    a. Based on these findings we assign the homes to the appropriate real estate agent. The more expensive areas require a more experienced agent. Note that this is important for us to list the more expensive areas as this helps create a more exclusive reputation and assists the brokerage in getting listings in all areas.
4. We have a team of engineers that load datasets and run the required calculations and Regression testing as defined in the project.
    a. First, we need to verify the dataset and its accuracy.
    b. Second, we need to identify the Mean and Standard Deviation for all IV and DV.
    c. Third, we run Scattered and Curved plots to go through the patterns and numbers of the dataset

d. Fourth, We evaluate the results based on the visualized patterns of plots, the numbers, the error numbers, predicted vs. actual, and test results.
e. Fifth, we establish a base 95% Confidence Interval for the price per square foot.
f. Six, if needed we will continue to filter lower into individual Zip code areas, and establish the 95% Confidence Intervals for each area. In addition we will establish a minimum value based on the %25 lower tail values.

Note: The lower tail base price is a good starting point to evaluate the property price per zip code, while the confidence intervals is a more exact price evaluation for the majority of the properties that fall in the mean range of Bedrooms-Baths-YearBuilt combo.

Design

1. Data Importing and Pre-processing

The process of the Extract, Transform, Load (ETL) for the San Diego MLS listing csv file with 300 rows.

(1.1) Importing Libraries

Importing the necessary libraries to read the dataset

(See Python code)

(1.2) Importing Dataset

Actual Importing the dataset from the csv file and storing it in a dataframe

- redfin_2022_san_diego-all.csv

(1.3) Initial Data Summary

Review data field attributes and data types, ensure proper import and verify the field types

(1.4) Data Cleaning

Normalization of the data by removing unnecessary columns and replacing missing values

2. Statistical Analysis and Visualization

Running the calculations to produce the required values, i.e. Means, Standard Deviations, lower and upper tails

(2.1) Final Dataframe Information

Filtering in on the final set of data used

(2.2) Dataframe Description

Descriptive statistics of the dataframe, table of values

(2.3) Dataframe Visualization (General)

Visualizing the dataframe using histograms; Curvature and Plot

(2.4) Dataframe Insights

Grouping Dataframe by Neighborhood/Zipcode and calculating the mean of each column sorted by '$/SQUARE FEET': Better presentation of the landscape of pricing in the San Diego area broken down by Zip Codes.

(2.5) Dataframe Visualization (Grouped)

Boxplot of the grouped dataframe by Neighborhood/Zipcode. From the boxplot we can see that the Zipcode 92037 has the widest spread in price per square foot.

(2.6) Convert Categorical Data to Numerical Data

Using One Hot Encoding to convert categorical data to numerical data

(2.7) Setting the Independent and Dependent Variables

Set the independent variables (X): Zip Codes, and dependent variable (y): Price Per Square Foot

(2.8) Dataframe Visualization (Correlation)

- Probability Density Function (PDF) of the Independent Variables
- Q-Q Plot of the Independent Variables. The goodness of fit is determined by the closeness of the points to the line and currently displays a normal fit.
- Heatmap for Correlation Matrix of the Independent Variables and Dependent Variable

3. Regression Model

(3.1) Importing Libraries

Importing the necessary libraries to split the dataframe (sklearn)

(3.2) Splitting the Dataframe

Splitting the dataframe into training and testing sets (70/30)

(3.3) Dataframe Modeling

Importing the necessary libraries to model the dataframe and Setting the model

(3.4) Fitting the Model

Fit the model to the training set

4. Model Evaluation

(4.1) Model Score

Scoring the model on the training set (R^2)

Model Score using OLS

(4.2) Visualization (Actual vs Predicted)

Setting the scatterplot of the actual vs predicted values

(4.3) Mean Absolute Error (MAE) and Mean Squared Error (MSE)

5. Model Prediction Test (User Input)

Selecting a random row from the testing set and predicting the price

6. Conclusion

Hypothesis Testing

1. H0: The predicted price is less than or equal to the actual price
2. H1: The predicted price is greater than the actual price

This Hypothesis will enable us to understand the landscape of the current listings in San Diego, and set expectations for client listings.

The Confidence Intervals will be used as a gauge range for brokers to use in calculating their BOV for potential clients.

Go Live

1. Activate the code
2. Regression testing: Fit a model
    a. Take a video of this: Let’s individually video what we do (short clips) and later we can put together to show the entire flow of work.
    b. Explain the regression process: Show the check for correlation, the verification of what we chose.
    c. Plot the regression model with the R^2 table etc.
3. Create a presentation for this, if PP or/and Video:
    a. Explain the Project
        i. The Dataset
        ii. The Fields in the data and the ETL process
        iii. The Goal of testing: The Prediction
        iv. The Data tested
        v. The Hypothesis H0 vs. H1
        vi. The Regression used and the description of the Analysis
        vii. Finalize the findings
4. (Modifications as needed)

Final Submission

1. Team approval-Review
2. Final Submission to the Professors: By 10/24/2022

Scope of Work

10/6/2022:

- We began to plan the scope of the project and introduce ourselves

10/7/2022:

- We began the project plan build out:
    - Define the resources required
    - Start a Project Plan documentation
    - Identify the Team Rep
    - Identify the tasks roles within the team
- Download the Dataset: San Diego Home Listing from Redfin (300)
- Setup a shared GitHub repository
- Setup a Google Docs: Start the Project Management and documentation
- Setup a Slack Team
- Setup a Video Conferencing schedule

10/10/2022:

- First team call:
    - Strategize the business need
    - Strategize the project scope
    - First action plan: Real Estate Agency opening in San Diego

10/11/2022:

- Work on Python code
- Work on Project Scope documentation
- R&D work

10/11/2022:

- Second team meeting:
    - Review current code progress: Trevor presenting
    - Evaluate alignment with the business requirements
    - Discuss obstacles
    - Continue to assign team tasks as needed

10/12/2022:

- Third team meeting:

  - Review current code progress
  - Refine the reports: Highlight the IV and DV sections: Trevor
  - Verify the results: Ike
  - Update reports and documentation: Adam
  - Set a verification sheet in Excel: Adam
  - Set the DV base values per Zip Code: Trevor

10/14/2022:

- Fourth team meeting:
  - Review current code progress
  - Approve the graphs
  - Approve the numbers
  - Discuss the final presentation method and brainstorm ideas
  - Research outlier

10/16/2022:

- Fifth team meeting:
  - Review current code progress:
    - 1.1: Importing the Python libraries: Done
    - 1.2: Importing The Dataset: Done
    - 1.3: Initial Data Summary: Done
    - 1.4: Data Cleaning: Done
    - 2.1: Dataframe Information: Done
    - 2.2: Dataframe Description: 95% In the comments add more info about what the results mean
    - 2.3: Dataframe Visualization: Done
    - 2.4: Dataframe Insights: Done
    - 2.5: Dataframe Visualization (Grouped): Done
    - 2.6: Convert Categorical Data to Numeric: 95% Explain why?
    - 2.7: Setting IV and DV: Done
    - 2.8: P.D.F: What does it mean?, Q-Q Plot: Done
    - 3.1: Importing Libraries: Done
    - 3.2: Splitting the Dataframe: Done
    - 3.3: Dataframe Modeling: Done
    - 3.4: Fitting Model: Done
    - 4.1: Model Score: 95%: 0.89 is a good score?
    - 4.2: Visualization (Actual vs. Predicted): Done

    - 4.3: MAE, MSE: What does do for us, explain connection
    - 5: Model Prediction Test: Done
    - 6: Conclusion: We need to organize the information along with the need, and create the Hypothesis
        - H0: for Xi=92122 x>=828, H1: < 828
        - H0: for Xi=9206H0: for Xi=92122 y>=828, H1: y< 828
        - H0: for Xi=92067 y>=840, H1: y< 840
        - H0: for Xi=92037 y>=889, H1: y< 889
        - H0: for Xi=91942 y>=485, H1: y< 485
        - H0: for Xi=92127 y>=579, H1: y< 579

- Discuss outlier: Completed; Decision to leave it as it shows the TRUE to the mansion style properties that can show up and need to be delegated to our exclusive agent services.
- Approve completed codes
- Schedule final presentation reports

10/18/2022:

- Six team meeting:
- Discuss the latest code
- Discuss the large outlier: impact
- Go over the regression testing

10/20/2022:

- Seventh team meeting:
- Ike working on documentation of formulas
- Adam to update the Project Management
- Trevor trying code without large outlier.
- Deciding to simplify the goal and regression dataframe

10/22/2022:

- Eighth team meeting:

- Simplify to a 95% price per foot for entire dataframe
- Keep the phase 2 optional by zip code as showing the option
- Adam to modify the Project Management and related documentation to reflect phases.
- Trevor to redo code to reflect new CI
- Ike to finish up his documentation and add to the Project documentation
- Adam to calculate the CI and verify against the code

10/23/2022:

- Ninth team meeting:
- Go over code:
  - Size: 290
  - Mean Price: 988.7172
  - Standard Dev: 635.5966
  - Margin of Error: 73.46
  - 95% Confidence Interval: 915.26 to 1062.18

Hypothesis Testing

1. H0: The predicted price is less than or equal to the actual price
2. H1: The predicted price is greater than the actual price

- Questions on plots: correct the Y axis text to In 10 Millions
- Verify the Confidence Intervals
- Verify the Hypothesis
- Discuss the Video presentations, review the current ones by Ike and Adam
- Review the Project Document, Adam add-ons, Ike add-ons
- Plan final presentation; Adam-complete the Project document, Ike complete documentation, Trevor: complete the testing (20)

Project - Triumph Real Estate Brokerage - San Diego Branch

1. Data Importing and Pre-processing

The process of the Extract, Transform, Load (ETL) for the San Diego MLS listing csv file with 300 rows.

- Sample file from Redfin including 300 listings.
- Name: redfin_2022_san_diego-all.csv

(1.1) Importing Libraries

Importing the necessary libraries to read the dataset

(See Python code)

(1.2) Importing Dataset

Actual Importing the dataset from the csv file and storing it in a dataframe

- redfin_2022_san_diego-all.csv

(1.3) Initial Data Summary

Review data field attributes and data types, ensure proper import and verify the field types

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 300 entries, 0 to 299

Data columns (total 27 columns):

 #   Column                                   Non-Null Count  Dtype

---  ------                                 --------------  -----

 0   SALE TYPE                                  300 non-null    object

 1   SOLD DATE                                  0 non-null      float64

 2   PROPERTY TYPE                                 300 non-null    object

 3   ADDRESS                                    297 non-null    object

 4   CITY                                     300 non-null    object

 5   STATE OR PROVINCE                               300 non-null    object

 6   ZIP OR POSTAL CODE                              300 non-null    object

 7   PRICE                                     300 non-null    int64

 8   BEDS                                      295 non-null    float64

 9   BATHS                                     290 non-null    float64

 10  LOCATION                                   300 non-null    object

 11  SQUARE FEET                                  290 non-null    float64

 12  LOT SIZE                                   199 non-null    float64
 13  YEAR BUILT                                           286 non-null    float64
 14  DAYS ON MARKET                                         296 non-null    float64
 15  $/SQUARE FEET                                         290 non-null    float64
 16  HOA/MONTH                                            173 non-null    float64
 17  STATUS                                              300 non-null    object
 18  NEXT OPEN HOUSE START TIME                                 33 non-null    object
 19  NEXT OPEN HOUSE END TIME                                  33 non-null    object
 20  URL (SEE https://www.redfin.com/buy-a-home/comparative-market-analysis FOR INFO ON PRICING)  300 non-null
object
 21  SOURCE                                               300 non-null    object
 22  MLS#                                                300 non-null    object
 23  FAVORITE                                             300 non-null    object
 24  INTERESTED                                            300 non-null    object
 25  LATITUDE                                             300 non-null    float64
 26  LONGITUDE                                             300 non-null    float64
dtypes: float64(11), int64(1), object(15)
memory usage: 63.4+ KB
```

Categorize Field Type and usage:

SALE TYPE	use = as is / text	Filter MLS Listing	Nominal
SOLD DATE	No data		
PROPERTY TYPE	use = as is / text	Filter Single Family Residential	Nominal
ADDRESS	use = as is / text		Nominal
CITY	use = as is / text		Nominal
STATE OR PROVINCE	use = as is / text		Nominal
ZIP OR POSTAL CODE	use = as is / number		Nominal
PRICE	use = as is / number		Interval
BEDS	use = as is / number		Interval
BATHS	use = as is / number		Interval
LOCATION	use = as is / text	sublevel to CITY	Ordinal
SQUARE FEET	use = as is / number		Ratio

LOT SIZE	use = as is / number		Ratio
YEAR BUILT	use = as is / number		Ratio
DAYS ON MARKET	use = as is / number		Ratio
$/SQUARE FEET	use = as is / number	Cal	Ratio
HOA/MONTH	use = as is / number	Not used	
STATUS	use = as is / text	Filter Active	Nominal
NEXT OPEN HOUSE START TIME	No need		
NEXT OPEN HOUSE END TIME	No need		
URL (SEE https://www.redfin.com/buy-a-home/comparative-market-analysis FOR INFO ON PRICING)	No need		
SOURCE	No need		
MLS#	No need		
FAVORITE	No need		
INTERESTED	No need		
LATITUDE	No need		
LONGITUDE	No need		

(1.4) Data Cleaning

Normalization of the data by removing unnecessary columns and replacing missing values

- Drop not used fields:

  'SALE TYPE', 'CITY', 'STATUS', 'STATE OR PROVINCE', 'SOLD DATE', 'ADDRESS', 'NEXT OPEN HOUSE START TIME', 'NEXT OPEN HOUSE END TIME'

2. Statistical Analysis and Visualization

Running the calculations to produce the required values, i.e. Means, Standard Deviations, lower and upper tails

(2.1) Final Dataframe Information

Filtering in on the final set of data used

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 290 entries, 0 to 299
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   PROPERTY TYPE       290 non-null    object
 1   ZIP OR POSTAL CODE  290 non-null    object
 2   PRICE               290 non-null    int64
 3   BEDS                290 non-null    float64
 4   BATHS               290 non-null    float64
 5   LOCATION            290 non-null    object
 6   SQUARE FEET         290 non-null    float64
 7   LOT SIZE            290 non-null    float64
 8   YEAR BUILT          290 non-null    float64
 9   DAYS ON MARKET      290 non-null    float64
 10  $/SQUARE FEET       290 non-null    float64
 11  HOA/MONTH           290 non-null    float64
dtypes: float64(8), int64(1), object(3)
memory usage: 29.5+ KB
```

(2.2) Dataframe Description

Descriptive statistics of the dataframe, table of values

Statistical description/summary of the data frame

The standard deviation of the price is more than the mean. This is because of extreme outliers, as is known in statistics. Once there is an extreme outlier, it will affect the mean (average) thus the mean is no longer the appropriate measure of centrality but the median. The lot size shows that there is a house that is on a large square foot, which correlates with a huge price that caused the outlier. Since the data set is a true random sample, the outlier is not removed.

(2.3) Dataframe Visualization (General)

Visualizing the dataframe using histograms; Curvature and Plot

The scatter plot of each column in the data frame against each other to show the correlations.

(2.4) Dataframe Insights

Grouping Dataframe by Neighborhood/Zipcode and calculating the mean of each column sorted by '$/SQUARE FEET': Better presentation of the landscape of pricing in the San Diego area broken down by Zip Codes.

(2.5) Dataframe Visualization (Grouped)

Boxplot of the grouped dataframe by Neighborhood/Zipcode. From the boxplot we can see that the Zipcode 92037 has the widest spread in price per square foot.

- Box plot to identify outliers. **Most outliers are from zip code 92037 can be seen outside the upper outer fence of the box plot**

(2.6) Convert Categorical Data to Numerical Data

Using One Hot Encoding to convert categorical data to numerical data

(2.7) Setting the Independent and Dependent Variables

Set the independent variables (X): Zip Codes, and dependent variable (y): Price Per Square Foot

(2.8) Dataframe Visualization (Correlation)

- Probability Density Function (PDF) of the Independent Variables
- Q-Q Plot of the Independent Variables. The goodness of fit is determined by the closeness of the points to the line and currently displays a normal fit.
- Heatmap for Correlation Matrix of the Independent Variables and Dependent Variable

Shows the plot of the Probability Density Function of the $/Square Feet. It shows the probability of $/Square Feet at the interval of each independent variable. (Example bed, bath, etc.). **It is right skewed because the data frame has data sets smaller than the median, occurring with relatively higher frequency than those values that are greater than the median.**

The next plot is the Q-Q plot. The quantile-quantile (q-q) plot is a graphical representation to show whether two data sets come from populations with a common distribution such as normal or exponential. A Q-Q plot is a scatterplot created by plotting two sets of quantiles against one another. If both sets of quantiles came from the same distribution, the Points closely form or follow a straight line. It is one way to check if the residuals are normally distributed. **The plot shows that the points relatively follow the straight line at the Interquartile range (IQR) but curve away from the line at points less than the first quartile(Q1) and points more than the third quartile (Q3).**

The third plot shows the Correlation Matrix of the Independent Variables and Dependent Variable. It shows the measure of association by the correlation between the columns in the data frame and indicates the correlation coefficient. The correlation coefficient measure can attain any value in the interval [-1, +1]. As the correlation or association between the variables increases, the absolute value of the correlation coefficient approaches 1. For perfect correlation/association between variables, the correlation coefficient could be 1 or -1, depending on if the correlation or association is positive or negative. **From the correlation matrix of the independent variables and the dependent variable, beds, baths, and square feet have about +0.75 correlation coefficient whereas the Property type_condo/ co-op has -0.5 correlation coefficient to Dependent variable $/square feet.**

3. Regression Model

(3.1) Importing Libraries

Importing the necessary libraries to split the dataframe (sklearn)

(3.2) Splitting the Dataframe

Splitting the dataframe into training and testing sets (70/30)

(3.3) Dataframe Modeling

Importing the necessary libraries to model the dataframe and Setting the model

(3.4) Fitting the Model

Fit the model to the training set

4. Model Evaluation

(4.1) Model Score

Scoring the model on the training set (R^2)

Model Score using OLS

Is the diagnostics results that shows the model evaluation statistics highlighting various quality indicators like the R-squared value and Adj. R-squared value. **Both the R-squared and the Adj. R-squared values are close to 1, which means that the model is good.**

(4.2) Visualization (Actual vs Predicted)

Setting the scatterplot of the actual vs predicted values

(4.3) Mean Absolute Error (MAE) and Mean Squared Error (MSE)

5. Model Prediction Test (User Input)

Selecting 20 random rows from the testing set and predicting the price

Test 1: H0: XiMean (Predicted)<25th lower quantile vs. XiMean (Predicted)> 25th lower quantile

Results:

```
T-Test of the Sample 20 Houses
T-Test:  Ttest_1sampResult(statistic=2.112344959833596,
pvalue=0.048121966633116645)
Sample Mean of the Predicted Price:  829.0138127085495
Sample SD of the Predicted Price:  448.3316038302377
25th Quantile:  611.75
```

Conclusion: The predicted price is greater than the 25th Quantile

Reject the Null Hypothesis

Test 2: H0: XiMean = 989 vs. XiMean not=989 (Two Tail)-(Price Per Square Foot)(Are the test values in the critical region of a %5 level of significance)

We reject H0 if either XiMean<915.26 or XiMean>1062.18

Results:

```
Sample Mean (actual 20):  1034.75
Sample SD:  402.1564
```

Conclusion: At 95% Confidence Interval Calculations: The observed value does not fall into the critical region and we do not reject H0.

The model was used to perform a multi linear regression on a user input data against the result output:

Mean Price: 988.7172

Standard Dev: 635.5966

Margin of Error: 73.46

95% Confidence Interval: 915.26 to 1062.18

Extra:

We downloaded a new dataset to run a larger test region and to verify the issues of a very scattered price range.

The regression testing was good; however we did find a closer range between the Predicted price vs. Actual price, and a lower mean for $/SQUARE FEET. This was due to the lack of many outliers of the mansion style properties. We did see the La Jolla being the most expensive area confirming our conclusion on zip code values.

This verifies our conclusions that multiple datasets are required to get more accurate values.

6. Conclusion

Hypothesis Testing 1: (Predicted Price vs. 25% quantile)(Good indication as to where the market is)

H0: The predicted price is less than the 25th quantile

H1: The predicted price is greater than the 25th quantile

This Hypothesis will enable us to understand the landscape of the current listings in San Diego and set expectations for client listings.

The results have shown that we reject the null hypothesis (H0). However at a closer look it is borderline, This indicates that the market is moving from a Seller's Market to a Buyer's Market.

Hypothesis Testing 2: (Price Per Square Foot)(Are the test values in the critical region of a %5 level of significance)

H0: XiMean = 989 vs. H1: XiMean not= 989

If not rejected then this means that the Confidence Intervals will be used as a gauge range for brokers to use in calculating their BOV for potential clients.

San Diego Real Estate Landscape Review:

1. Outliers are part of the Landscape
2. La Jolla Zip code 92037, is by far the most expensive area, with multiple listings above the 10mm
3. Prices are scattered, multiple testing for Hypothesis are required
4. Due to the null hypothesis of the 25% lower tail being rejected and it is on the borderline, it seems the city is in a transition from a Seller's Market to a Buyer's Market.
5. The Second Hypothesis shows that we have a good critical region to work with, and the 95% Confidence Intervals are good to use for doing the Brokers Opinion of Value for potential clients.
6. The lower 25% tail Price Per Square Foot is a good default for brokers to estimate their BOV on.

Optional Phase 2 Notes:

As we have a better breakdown of the property prices per SF by the Zip codes, we can better understand:

- The area and its related Zip Codes
- The differences between the zip codes in prices per SF
- The percent differences in price per Zip code
- Able to pair potential listings by area to dominant broker

- Keep up-to-date on prices by Zip Code by doing periodic Regression Testing with current code, and verifying the numbers and relating to the Hypothesis.