Housing Prices Statistical Analysis

S P Jain School of Global Management
DUBAI • MUMBAI • SINGAPORE • SYDNEY

Naima Dzhunushova

BS23DSY045

Table of Contents:

Introduction:

Predicting housing prices is crucial for homeowners, investors, and policymakers. This project aims to develop a statistical model for predicting housing prices using various factors like area, number of bathrooms, and number of bedrooms. By accurately forecasting prices, this model can aid decision-making for buyers, sellers, and policymakers. We'll detail our methodology, present results, and discuss implications for stakeholders. This project contributes to understanding and navigating the complexities of the housing market. We will be working with the Housing Prices dataset from Kaggle.

Objectives:

- **Understanding the Dataset:** We'll analyze the distribution of housing prices and feature characteristics using visualizations like histograms, boxplots, and correlation matrices. We will perform Exploratory Data Analysis (EDA) and conduct a cleanup if required.
- **Predictive Modeling**: Using multiple linear regression, we'll construct predictive models to estimate the price of houses based on the provided features.
- **Project Outcomes & Insights:** We'll interpret our analysis findings, pinpointing key factors that notably influence housing prices.

Methodology:

We use linear regression – the equation of this line is what we use to make predictions. The equation for the line in regression modeling takes the form:

$$y_i = \beta_0 + \beta_1 x_i + e_i$$

β0 is the intercept also called the constant– this is where the line crosses the *y* axis of the graph.
β1 is the slope of the line – this is how much the value of *y* increases, for a one-unit increase in *x*.
ei is the error term for the *i th* attribute. The error is the amount by which the predicted value is different from the actual value.

In linear regression, we assume that if we calculate the error terms for every person in the sample, and take the mean, the mean value will be **zero**. The error term is also referred to as the residual.

$$e_i = y_i - \hat{y}_i$$

Multiple linear regression extends simple linear regression to include more than one explanatory variable. We still use the term 'linear' because we assume that the response variable is directly related to a linear combination of the descriptive variables (Tranmer, Murphy, Elliot & Pampaka, 2020).

The equation for multiple linear regression:

$$y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_p x_{pi} + e_i$$

β0 is the constant – which will be the predicted value of y when all explanatory variables are 0. In a model with *p* explanatory variables, each explanatory variable has its **β_coefficient.**

Hypothesis testing:

$$H_0\colon\ \beta_1 = 0$$
$$H_1\colon\ \beta_1 \neq 0$$

The null hypothesis is that there is no association, so the slope of the line, denoted $\beta 1$, would be zero. If there is a relationship, then the slope is not zero – our alternative hypothesis.

Correctly categorizing and measuring skewness provides insights into how values are spread around the mean and influences the choices of statistical techniques and data transformations. There are three types of skewness: positive, negative, and zero skewness

$$Skewness = \frac{3(mean - median)}{Standard\ deviation}$$

Distribution based on skewness value:

Skewness = 0: normally distributed.

Skewness > 0: more weight in the left tail of the distribution.

Skewness < 0: more weight in the right tail of the distribution.

While skewness focuses on the spread (tails) of normal distribution, kurtosis focuses more on the height. It tells us how peaked or flat our normal (or normal-like) distribution is.

$$Kurtosis = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{x_i - mean}{Stdev}\right)^4 - 3$$

kurtosis for normal distribution is equal to 3.

For a distribution having kurtosis < 3: It is called playkurtic, flat peak, and light tails.

For a distribution having kurtosis > 3, It is called leptokurtic, sharp peak, heavy tails, and it signifies that it tries to produce more outliers rather than the normal distribution.

Data Analysis:

Data Exploration:

Let's take a look at the sample data of five:

```
df=pd.read_csv('Housing.csv')
df.sample(5)
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
159	5460000	3150	3	2	1	yes	yes	yes	no	yes	0	no	furnished
124	5950000	6525	3	2	4	yes	no	no	no	no	1	no	furnished
501	2660000	2430	3	1	1	no	no	no	no	no	0	no	unfurnished
373	3640000	3000	2	1	2	yes	no	no	no	yes	0	no	furnished
31	8400000	7000	3	1	4	yes	no	no	no	yes	2	no	semi-furnished

We can also check how many rows and columns there are in the dataset.

```
rows,columns = df.shape
print("num of rows in data :",rows)
print("the number of column in the data:",columns)
```

```
num of rows in data : 545
the number of column in the data: 13
```

The data consists of 545 entries and 13 attributes: price, area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hotwaterheating, airconditioning, parking, prefarea, and furnishingstatus.

price: This column indicates the price of a house and it is the dependent variable to be studied based on other features.

area: This attribute refers to an area of the house. It has a numerical data type.

bedrooms: This column specifies how many bedrooms a house has. The houses in this dataset have 1 to 6 bedrooms.

bathrooms: This attribute denotes the number of bathrooms in a house, with values 1, 2, 3, and 4.

stories: It indicates the number of stories a house has. The values are from 1 to 4.

mainroad: This feature signifies whether the house is located on the main road or not, with values 'yes' and 'no'.

guestroom: It specifies whether the house has a guest room or not, with ‘yes’and ‘no’ values.

basement: This attribute indicates the presence of a basement in the house, taking values of ‘yes’ and ‘no’.

hotwaterheating: It signifies whether the house has a hot water heating system or not (‘yes’, ‘no’).

airconditioning: This attribute indicates whether the house is equipped with an air conditioning system, having values such as ‘‘yes’ and ‘no’’.

parking: It denotes the parking facilities available with the house (‘yes’, ‘no’).

prefarea: This column informs us whether the house is situated in the preferred neighborhood of the city, with values 'yes' and 'no'.

furnishingstatus: This attribute describes the furnishing status of the house. It has values such as ‘unfurnished’, ‘semi-furnished’, and ‘furnished’.

After counting how many unique values each attribute has, we can say there are 11 categorical & 1 numerical feature in this dataset.

Here is the overall description of our data.

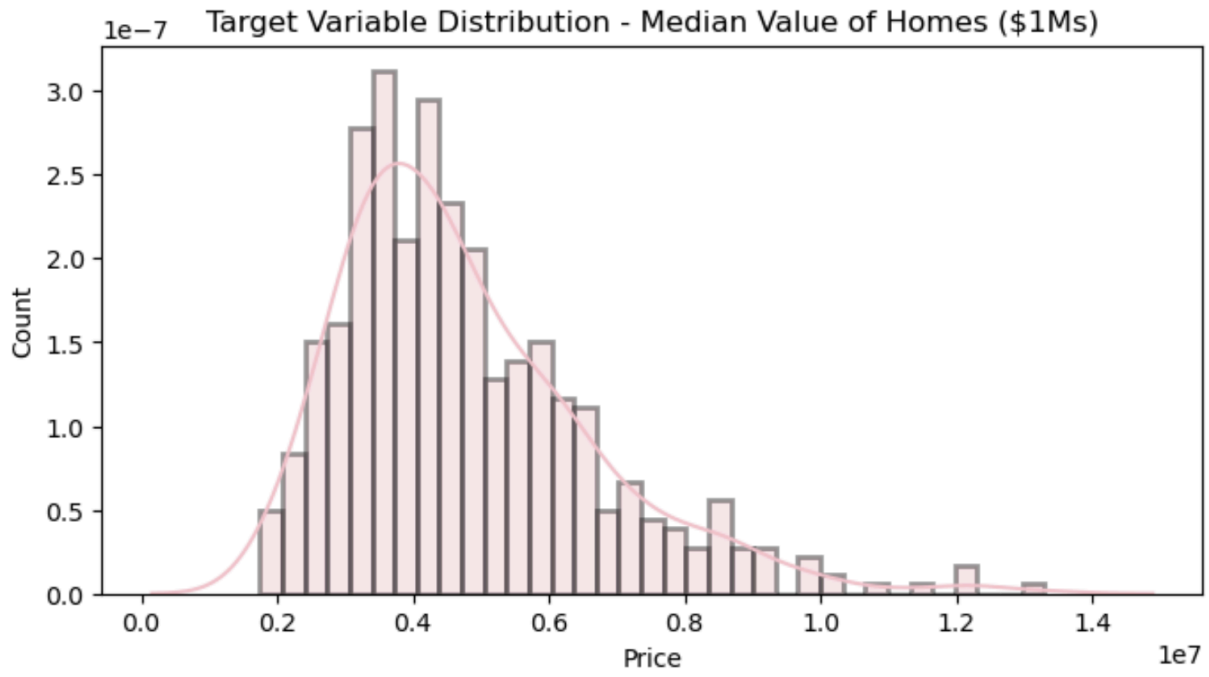
```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
price	545.0	4.766729e+06	1.870440e+06	1750000.0	3430000.0	4340000.0	5740000.0	13300000.0
area	545.0	5.150541e+03	2.170141e+03	1650.0	3600.0	4600.0	6360.0	16200.0
bedrooms	545.0	2.965138e+00	7.380639e-01	1.0	2.0	3.0	3.0	6.0
bathrooms	545.0	1.286239e+00	5.024696e-01	1.0	1.0	1.0	2.0	4.0
stories	545.0	1.805505e+00	8.674925e-01	1.0	1.0	2.0	2.0	4.0
parking	545.0	6.935780e-01	8.615858e-01	0.0	0.0	0.0	1.0	3.0

The stats seem to be fine, let us do further analysis on the Dataset.

Exploratory Data Analysis (EDA):

Let us first analyze the distribution of the target variable, starting with drawing its histogram.

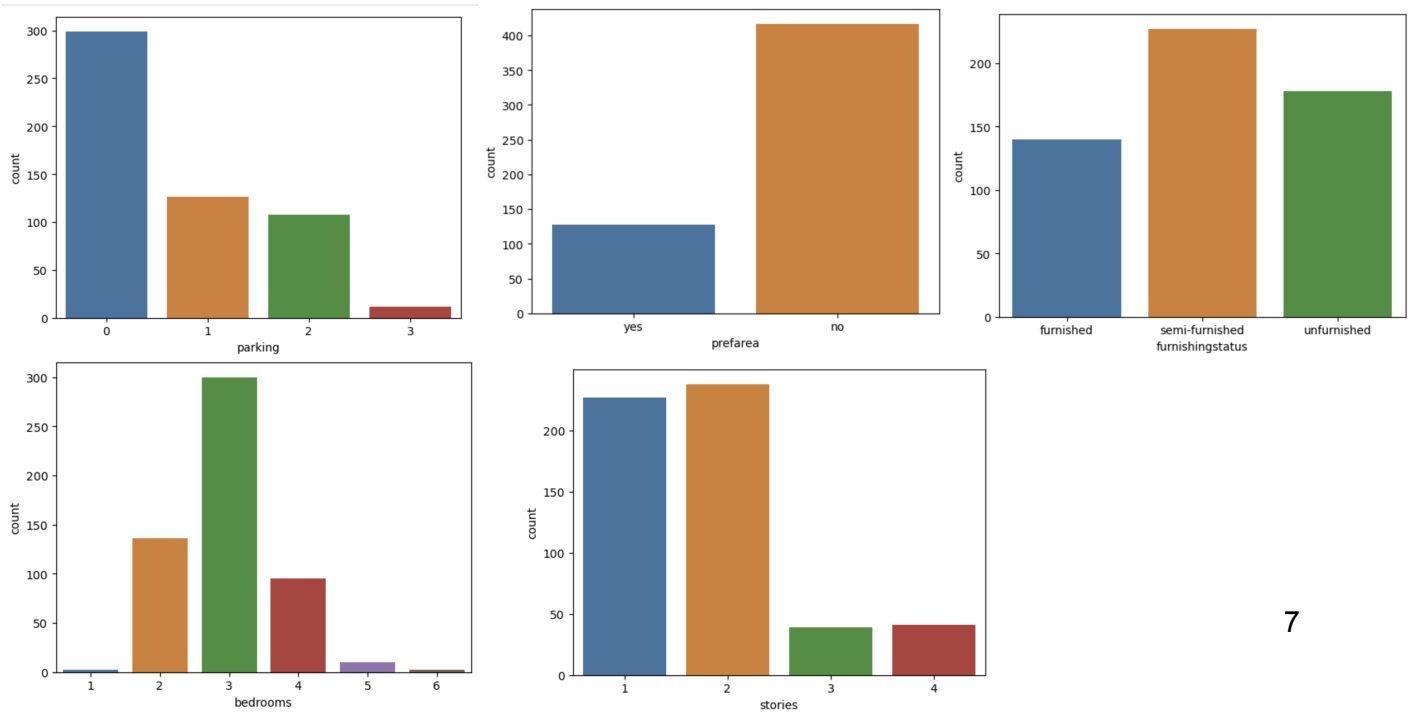


As we see it follows a normal distribution, however, it is positively skewed (right-skewed).

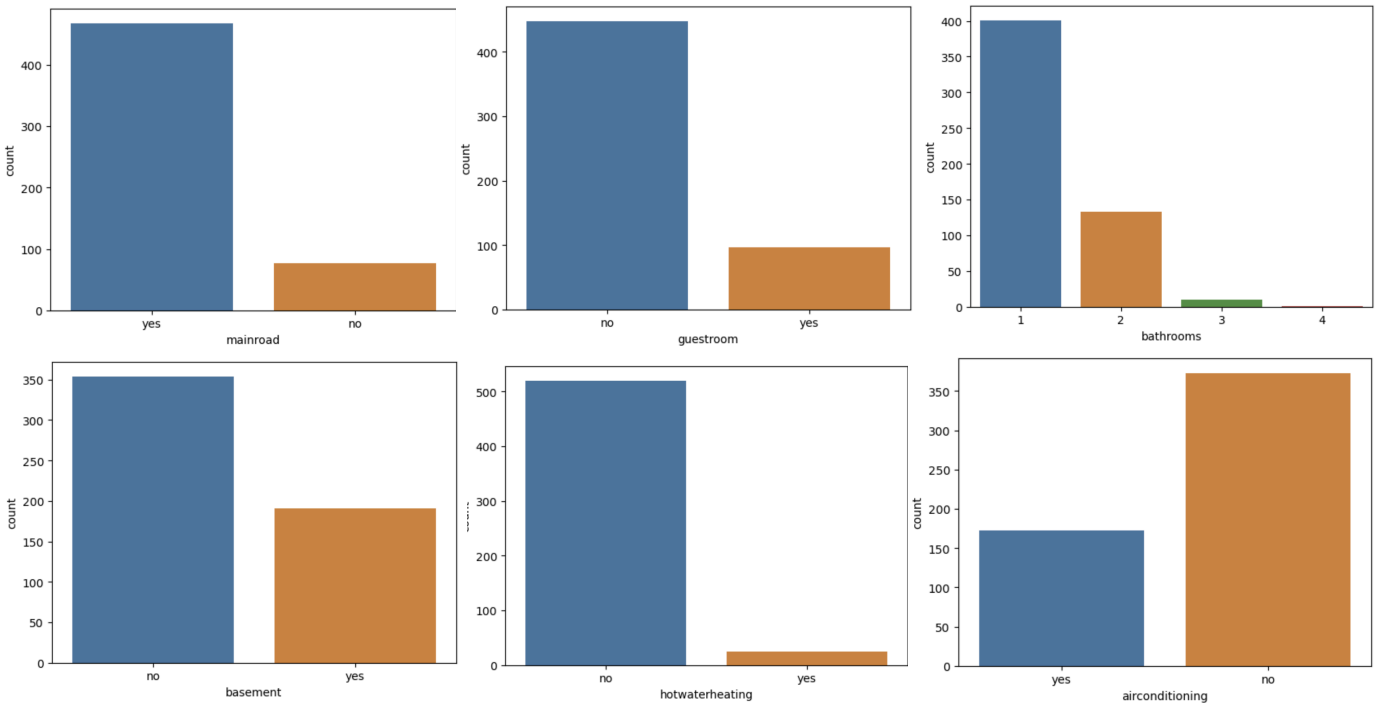
```
skewness: 1.2088998457878217
kurtosis: 1.9312045083457265
```

We see that calculated skewness of 1.21 and excess kurtosis of 1.93 confirm that when plotted housing price is right-skewed and has a sharper peak than a normal distribution.

Visualising Categorical Features:

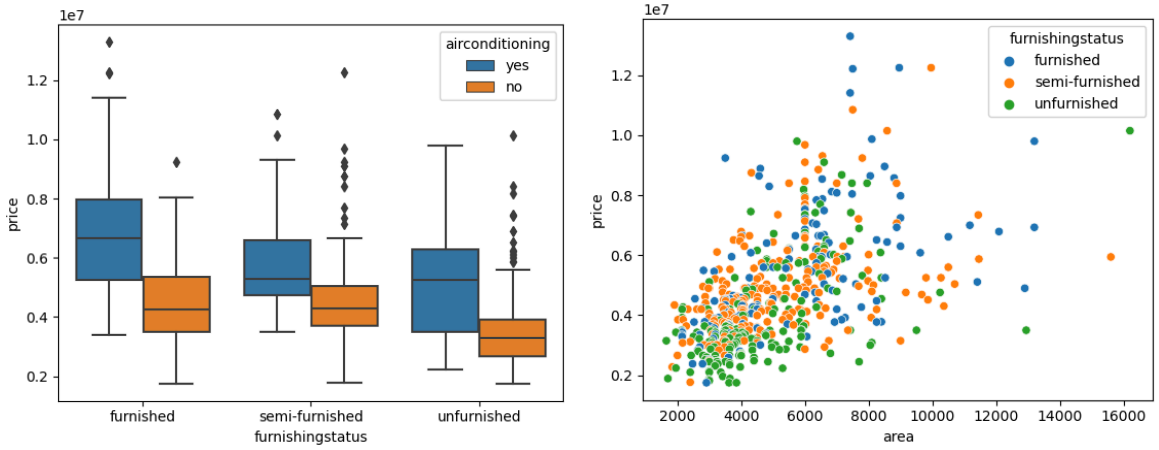


Let’s draw count plots for every categorical feature.



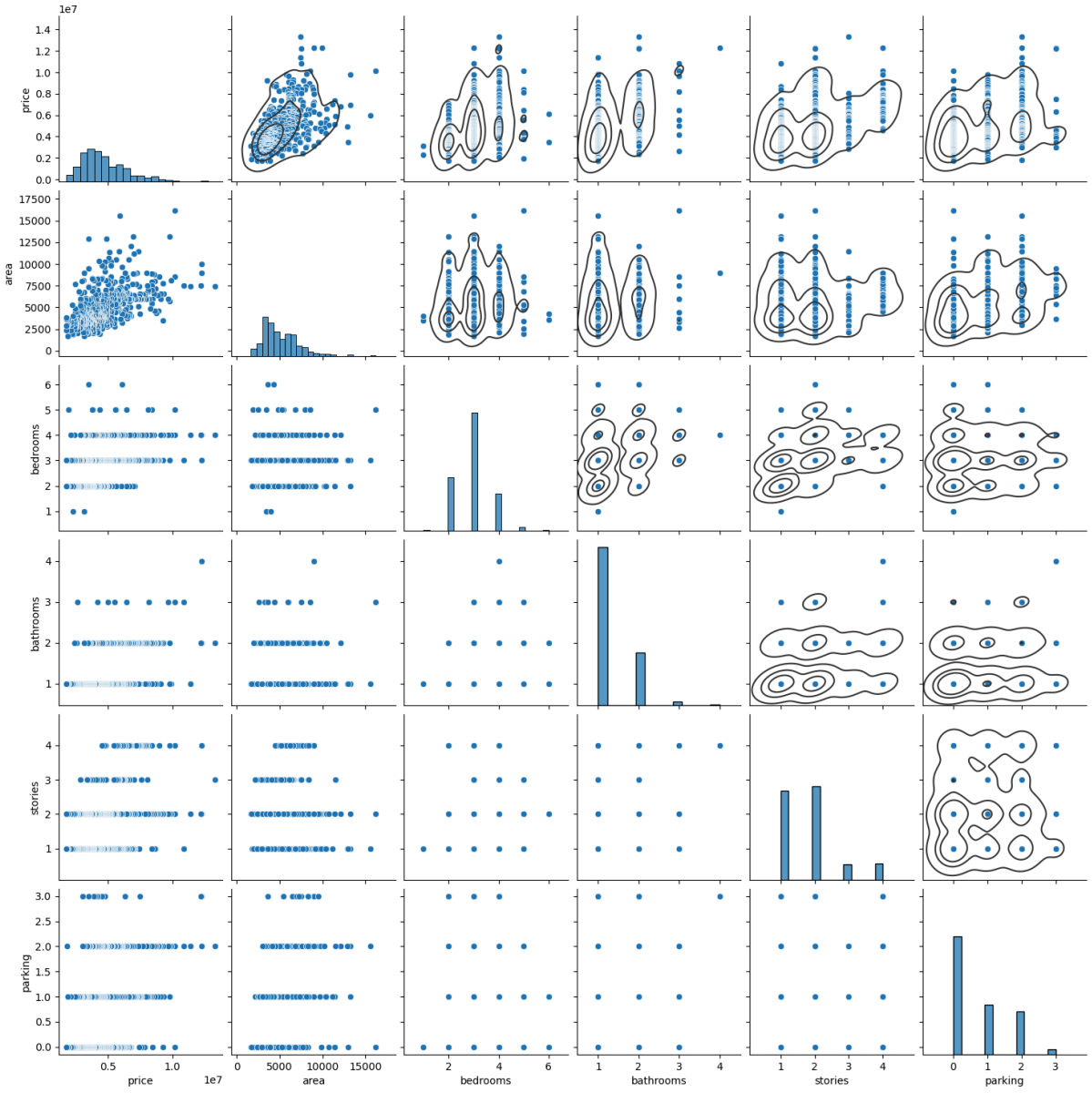
By looking at these plots of categorical features, we conclude that there is a small number of records with 3, or 4 bathrooms, 1, 5, or 6 bedrooms, 3 parking spots, and hot water heating. Still, we can’t judge their influence on a price as less significant just yet. We will check their significance again in the modeling process.

Two other visualizations based on furnishing status relationship to the house price.



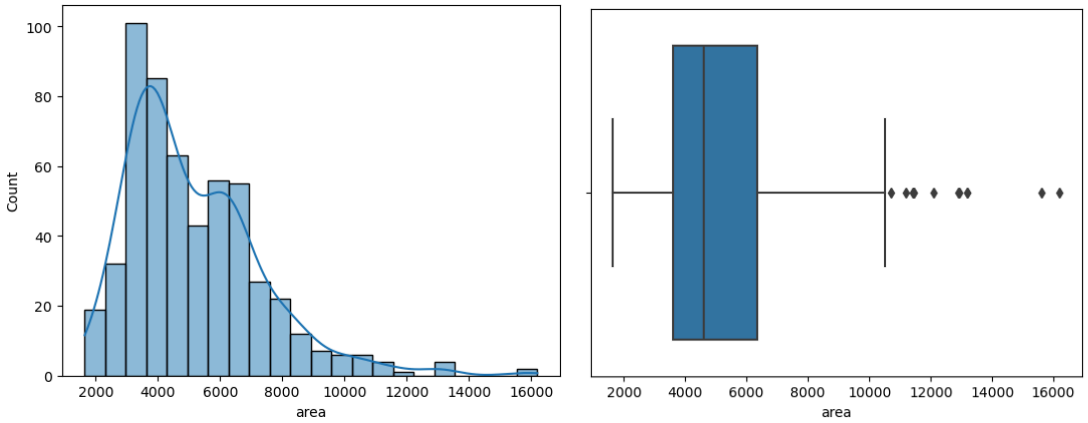
We can see that different furnishing statuses don’t show big differences in affecting our target value.

Let’s draw pair plots for all features to better understand their relationships with each other.



We can notice that some features have a linear relationship.

Let’s visualize our one numeric feature.



We can see that we have got some outliers. Let’s fix this.

Data Pre-processing:

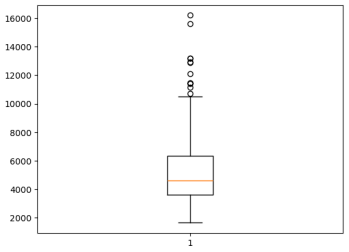
The data doesn’t have inconsistent values or duplicates.

```
df.isnull().sum() # null values check
```

```
price             0
area              0
bedrooms          0
bathrooms         0
stories           0
mainroad          0
guestroom         0
basement          0
hotwaterheating   0
airconditioning   0
parking           0
prefarea          0
furnishingstatus  0
dtype: int64
```

```
df.duplicated().sum() # duplicate values check
```

```
0
```



```
# Dealing with outliers in area

print('Before removal of outliers, The dataset had {} samples.'.format(df.shape[0]))
Q1 = df.area.quantile(0.25)
Q3 = df.area.quantile(0.75)
IQR = Q3 - Q1
df2 = df[(df.area >= Q1 - 1.5*IQR) & (df.area <= Q3 + 1.5*IQR)]
plt.boxplot(df.area)

print('After removal of outliers, The dataset now has {} samples.'.format(df2.shape[0]))
```

```
Before removal of outliers, The dataset had 545 samples.
After removal of outliers, The dataset now has 533 samples.
```

After the cleanup process, 12 samples were dropped, retaining 97.8% of the data.

Feature Engineering

Let's try 2 different approaches to getting dummies.

For the first instance, we change values in columns: main road, guestroom, basement, hot water heating, air conditioning, and prefarea. So values 'yes' to 1, and 'no' to 0. And in the furnishing status column, we transform 'unfurnished' to 0, 'semi-furnished' to 1, and 'furnished' to 2.

Here is what a sample of six looks like after we have performed our feature engineering.

```
df2.sample(6)
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
530	2240000	1950	3	1	1	0	0	0	1	0	0	0	0
397	3500000	5900	2	1	1	1	0	0	0	0	1	0	2
154	5530000	3650	3	2	2	1	0	0	0	0	2	0	1
62	7070000	6240	4	2	2	1	0	0	0	1	1	0	2
307	4165000	4080	3	1	2	1	0	0	0	0	2	0	1
388	3500000	3650	3	1	2	1	0	0	0	0	0	0	0

And let's check the shape of our data.

```
rows,columns = df2.shape
print("num of rows in data :",rows)
print("the number of column in the data:",columns)
```

```
num of rows in data : 533
the number of column in the data: 13
```

And here is a description of our transformed data.

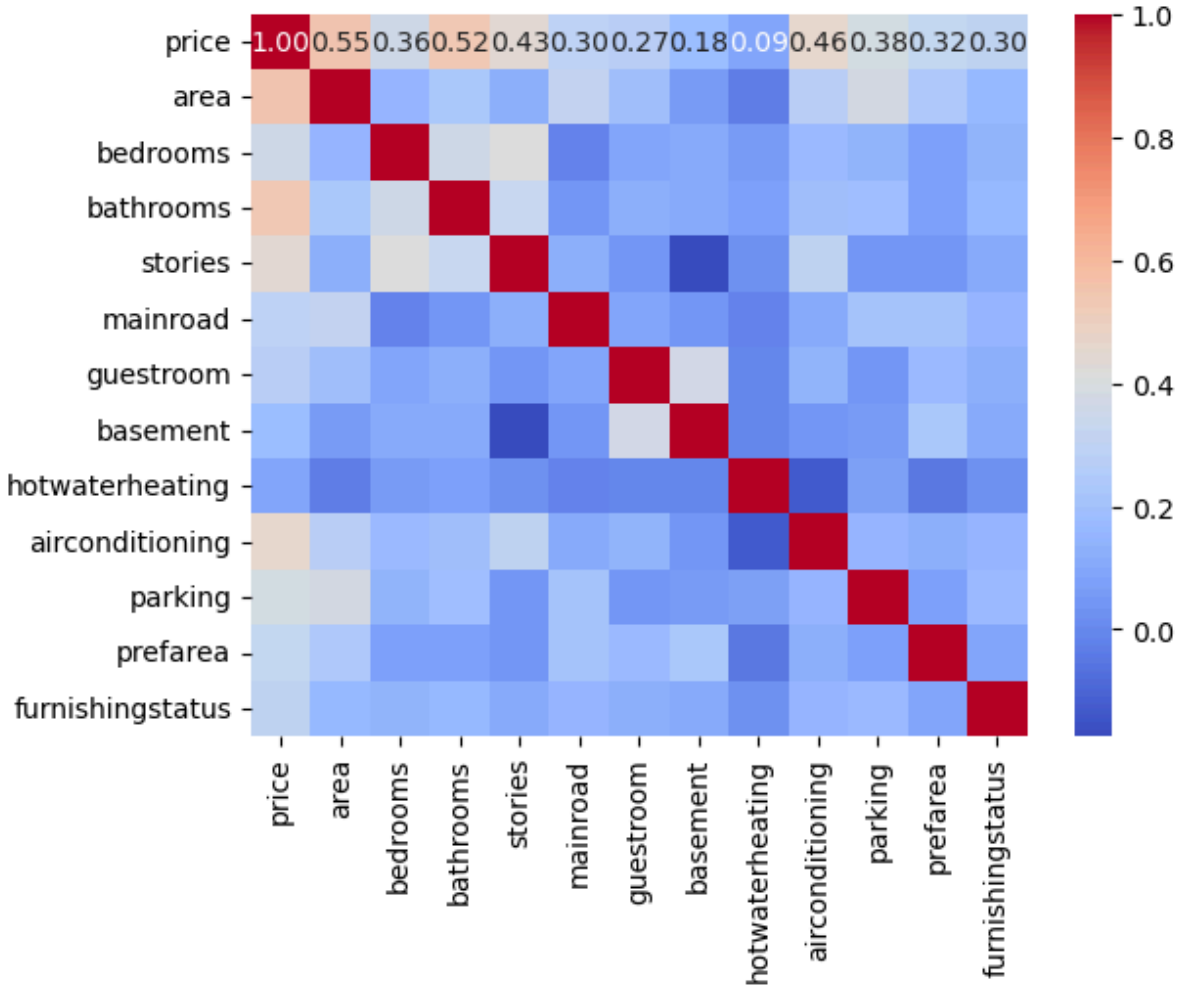
```
df2.describe().T
```

	count	mean	std	min	25%	50%	75%	max
price	533.0	4.726995e+06	1.851251e+06	1750000.0	3430000.0	4305000.0	5652500.0	13300000.0
area	533.0	4.980724e+03	1.855265e+03	1650.0	3540.0	4500.0	6300.0	10500.0
bedrooms	533.0	2.960600e+00	7.359877e-01	1.0	2.0	3.0	3.0	6.0
bathrooms	533.0	1.287054e+00	5.001516e-01	1.0	1.0	1.0	2.0	4.0
stories	533.0	1.808630e+00	8.719526e-01	1.0	1.0	2.0	2.0	4.0
mainroad	533.0	8.555347e-01	3.518912e-01	0.0	1.0	1.0	1.0	1.0
guestroom	533.0	1.801126e-01	3.846422e-01	0.0	0.0	0.0	0.0	1.0
basement	533.0	3.489681e-01	4.770916e-01	0.0	0.0	0.0	1.0	1.0
hotwaterheating	533.0	4.502814e-02	2.075607e-01	0.0	0.0	0.0	0.0	1.0
airconditioning	533.0	3.170732e-01	4.657733e-01	0.0	0.0	0.0	1.0	1.0
parking	533.0	6.848030e-01	8.595406e-01	0.0	0.0	0.0	1.0	3.0
prefarea	533.0	2.288931e-01	4.205149e-01	0.0	0.0	0.0	0.0	1.0
furnishingstatus	533.0	9.212008e-01	7.592666e-01	0.0	0.0	1.0	2.0	2.0

For the second instance, we get dummy variables for every value, except the first, of every categorical data. Let's explore our second example a bit later. Let's build a model using our first transformed dataset, then the second, and compare the two.

Data Manipulation:

Checking the correlation by using a heatmap.



Feature Scaling

We see that the correlation is very moderate. Keeping in mind that values in our attributes are very different with some from 0 to 1 and others in thousands, numerical, which is the area column, we should scale our data.

But first, let's split it into our training and testing datasets, so our model would train itself on the training dataset and predict values for the test dataset.

```
#Splitting the data intro training & testing sets
target = 'price'
X = df2.drop([target],axis=1)
Y = df2[target]
Train_X, Test_X, Train_Y, Test_Y = train_test_split(X, Y, train_size=0.8, test_size=0.2, random_state=100)
Train_X.reset_index(drop=True,inplace=True)

print('Original set  ---> ',X.shape,Y.shape,'\nTraining set  ---> ',
      Train_X.shape,Train_Y.shape,'\nTesting set   ---> ', Test_X.shape,'', Test_Y.shape)
```

```
Original set  --->  (533, 12) (533,)
Training set  --->  (426, 12) (426,)
Testing set   --->  (107, 12)  (107,)
```

```
#Feature Scaling (Standardization)

std = StandardScaler()

Train_X_std = std.fit_transform(Train_X)
Train_X_std = pd.DataFrame(Train_X_std, columns=X.columns)

Test_X_std = std.transform(Test_X)
Test_X_std = pd.DataFrame(Test_X_std, columns=X.columns)
```

Here is the description of our test data after standardization.

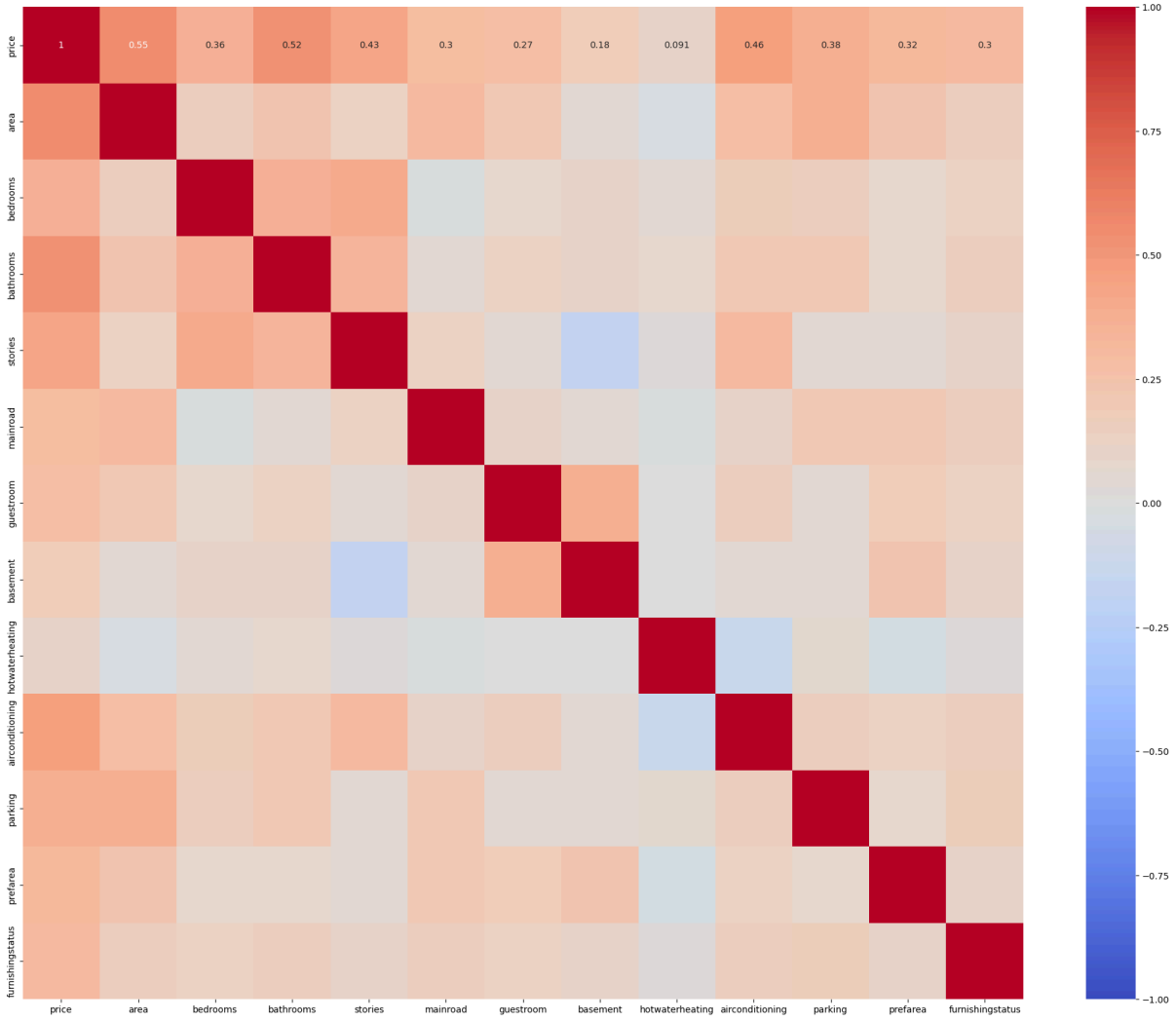
```
print('\033[1mStandardardization on Training set'.center(120))
display(Train_X_std.describe().T)
```

Standardardization on Training set

	count	mean	std	min	25%	50%	75%	max
area	426.0	-6.463270e-17	1.001176	-1.797874	-0.748789	-0.255102	0.611016	2.992840
bedrooms	426.0	-1.709639e-16	1.001176	-2.637489	-1.288665	0.060159	0.060159	4.106631
bathrooms	426.0	1.209257e-16	1.001176	-0.576679	-0.576679	-0.576679	1.436971	5.464271
stories	426.0	-6.984502e-17	1.001176	-0.915315	-0.915315	0.255629	0.255629	2.597517
mainroad	426.0	9.799152e-17	1.001176	-2.335497	0.428174	0.428174	0.428174	0.428174
guestroom	426.0	-2.501911e-17	1.001176	-0.465986	-0.465986	-0.465986	-0.465986	2.145988
basement	426.0	1.250956e-17	1.001176	-0.756278	-0.756278	-0.756278	1.322266	1.322266
hotwaterheating	426.0	-3.544374e-17	1.001176	-0.203874	-0.203874	-0.203874	-0.203874	4.904979
airconditioning	426.0	1.355202e-17	1.001176	-0.681115	-0.681115	-0.681115	1.468181	1.468181
parking	426.0	6.671763e-17	1.001176	-0.797919	-0.797919	-0.797919	0.370169	2.706345
prefarea	426.0	-2.918896e-17	1.001176	-0.539360	-0.539360	-0.539360	-0.539360	1.854050
furnishingstatus	426.0	1.000764e-16	1.001176	-1.218341	-1.218341	0.095617	1.409574	1.409574

Feature Selection:

Checking the correlation after scaling.



Let's look at the p-values of coefficients, testing a Linear Regression model with statsmodels.

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.717e+06	5.29e+04	89.232	0.000	4.61e+06	4.82e+06
area	4.564e+05	6.28e+04	7.265	0.000	3.33e+05	5.8e+05
bedrooms	1.005e+05	6.13e+04	1.641	0.102	-1.99e+04	2.21e+05
bathrooms	4.62e+05	5.98e+04	7.727	0.000	3.44e+05	5.79e+05
stories	4.043e+05	6.37e+04	6.343	0.000	2.79e+05	5.3e+05
mainroad	1.8e+05	5.75e+04	3.132	0.002	6.7e+04	2.93e+05
guestroom	1.26e+05	5.83e+04	2.162	0.031	1.15e+04	2.41e+05
basement	1.595e+05	6.07e+04	2.627	0.009	4.01e+04	2.79e+05
hotwaterheating	2.057e+05	5.42e+04	3.794	0.000	9.91e+04	3.12e+05
airconditioning	3.743e+05	5.87e+04	6.376	0.000	2.59e+05	4.9e+05
parking	2.345e+05	5.83e+04	4.020	0.000	1.2e+05	3.49e+05
prefarea	2.756e+05	5.65e+04	4.877	0.000	1.65e+05	3.87e+05
furnishingstatus	1.5e+05	5.51e+04	2.723	0.007	4.17e+04	2.58e+05

We see that the p-value of the attribute bedrooms is insignificant, so we can drop it.

```
Train_X_std.drop(['bedrooms'], axis = 1, inplace=True)
Test_X_std.drop(['bedrooms'], axis = 1, inplace=True)
```

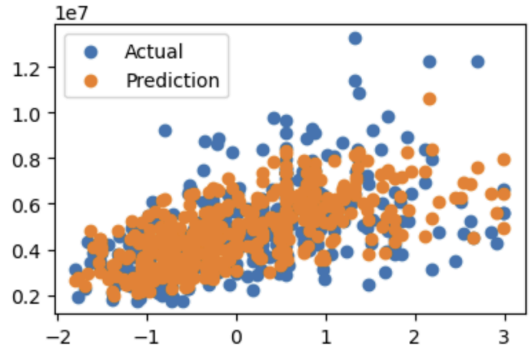
Predictive Modelling:

Now we create an instance of a multiple linear regression model and train it.

```
MLR = LinearRegression().fit(Train_X_std,Train_Y)
pred1 = MLR.predict(Train_X_std)
pred2 = MLR.predict(Test_X_std)
```

Let's see the coefficients of our model with the intercept, and plot predicted points alongside our actual points.

```
The Coeffecient of the Regresion Model was found to be  [462471.96815259 481863.02456962 441139.14753463 170158.46015036
 122291.99826536 173489.29800323 206477.09587415 374478.47929478
 240488.45873628 275454.08751257 155400.1980238 ]
The Intercept of the Regresion Model was found to be  4716708.779342723
```



From the above summary for our model, we can derive the Multiple Regression Equation to be:

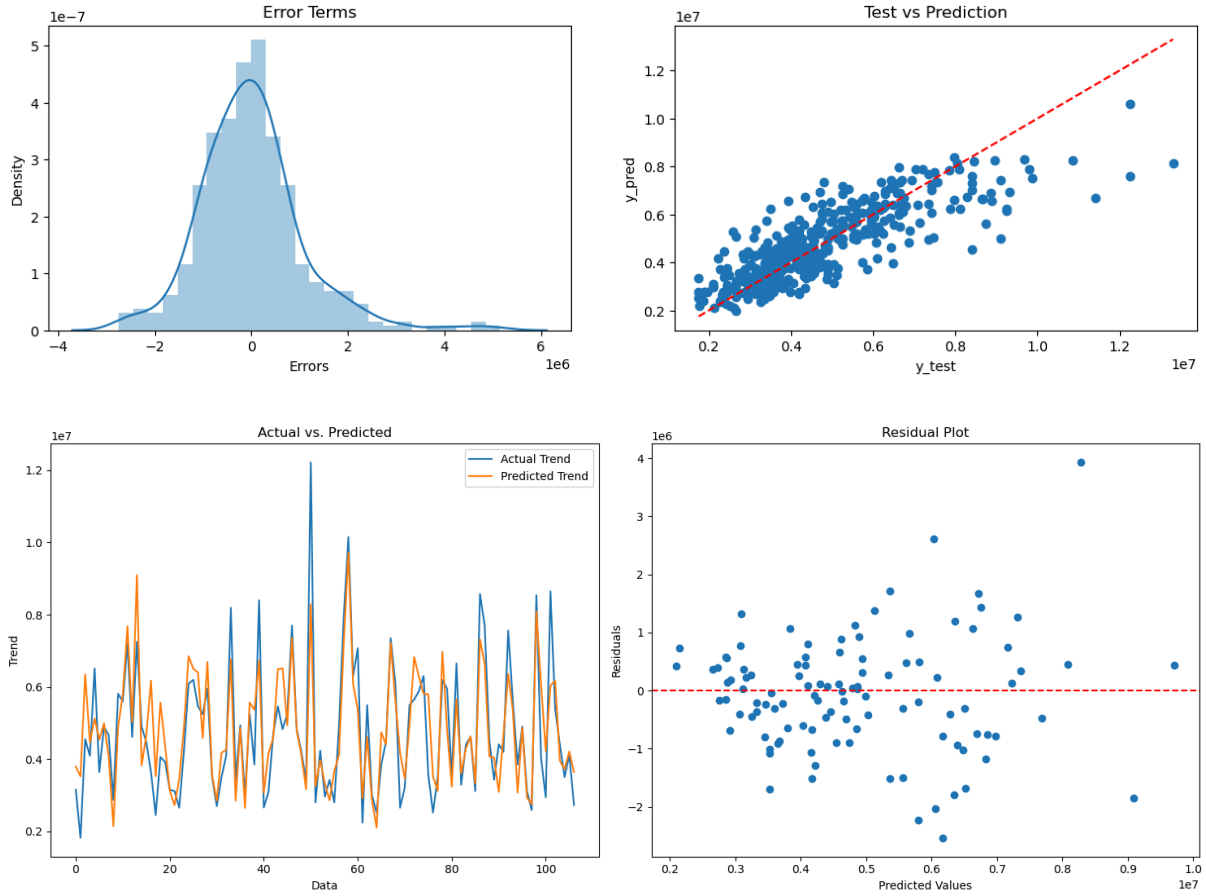
price = 4716708.779342723 + 462471.96815259*area + 481863.02456962*bathrooms + 441139.14753463*stories + 170158.46015036*mainroad + 122291.99826536*guestroom + 173489.29800323*basement + 206477.09587415*hotwaterheating + 374478.4792947*airconditioning + 240488.45873628*parking + 275454.08751257*prefarea + 155400.1980238*furnishingstatus

Let's evaluate our Multiple Linear Regression model by checking the R-squared score, Residual sum of squares, Mean squared error, and Root mean squared error on training and testing datasets.

```
--------------------Training Set Metrics--------------------

R2-Score on Training set ---> 0.659391304615061
Residual Sum of Squares (RSS) on Training set  ---> 494783098725147.1
Mean Squared Error (MSE) on Training set       ---> 1161462673063.7256
Root Mean Squared Error (RMSE) on Training set ---> 1077711.7764336278

--------------------Testing Set Metrics--------------------

R2-Score on Testing set ---> 0.7218542129195782
Residual Sum of Squares (RSS) on Training set  ---> 103015340369921.17
Mean Squared Error (MSE) on Training set       ---> 962760190373.0951
Root Mean Squared Error (RMSE) on Training set ---> 981203.4398498076
```

Looking at the R2-Score of 0.722 on our Testing dataset we can say our model did good.

And lastly, let's take a look at our residual plots and plots comparing our actual VS predicted values.



In this residual plot we don't see any patterns forming, which is good.

Feature Engineering 2:

Let us come back to the feature engineering step and try our second method, where we get dummy variables for every categorical data. So we change the column bedrooms to 5 columns like bedrooms_2, bedrooms_3, bedrooms_4, bedrooms_5, and bedrooms_6, each taking values 1 or 0. So if there are zeroes in every 'bedrooms_x' column in a data entry, it indicates that the house has only 1 bedroom. Similarly, we transform columns: bathrooms, stories, parking, and furnishing status to attributes: bathrooms_2, bathrooms_3, bathrooms_4, stories_2, stories_3, stories_4, parking_1, parking_2, parking_3, furnishingstatus_semi_furnished, furnishingstatus_unfurnished.

This is how a sample of five of our new data looks like:

```
df3.sample(5)
```

	price	area	mainroad	guestroom	basement	hotwaterheating	airconditioning	prefarea	furnishingstatus_semi-furnished	furnishingstatus_unfurnished	...
254	4480000	4510	1	0	0	0	1	0	1	0	...
504	2653000	3185	1	0	0	0	1	0	0	1	...
247	4550000	8400	1	0	0	0	0	0	0	1	...
456	3118850	2398	1	0	0	0	0	1	1	0	...
16	9100000	6600	1	1	1	0	1	1	0	1	...

5 rows × 24 columns

...	bedrooms_6	stories_2	stories_3	stories_4	bathrooms_2	bathrooms_3	bathrooms_4	parking_1	parking_2	parking_3
...	0	1	0	0	0	0	0	0	1	0
...	0	0	0	0	0	0	0	0	0	0
...	0	0	0	1	0	0	0	0	0	1
...	0	0	0	0	0	0	0	0	0	0
...	0	1	0	0	1	0	0	1	0	0

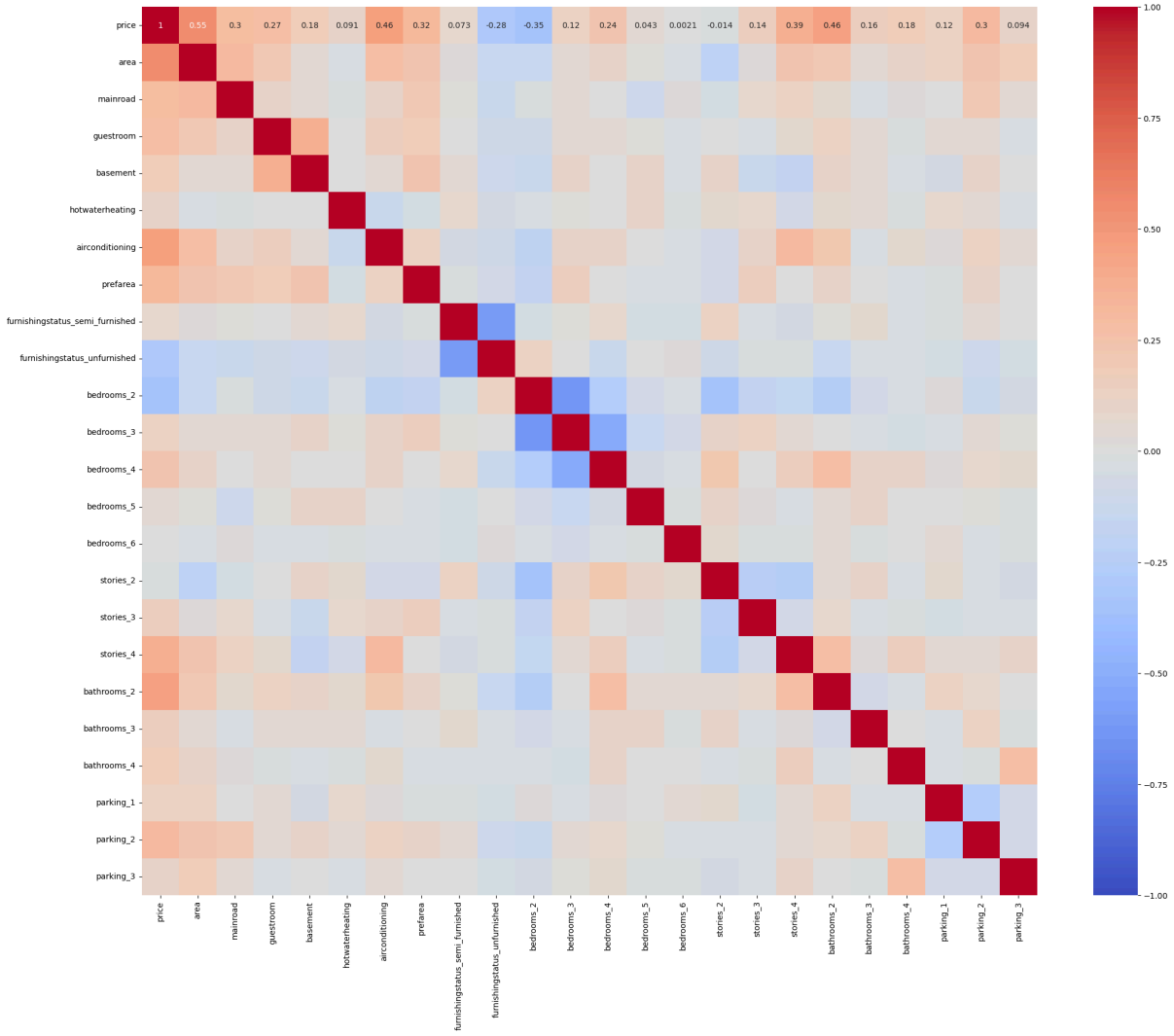
We can't see all of the columns in this representation. Here they are.

```
for col in df3.columns:
    print(col)
```

```
price
area
mainroad
guestroom
basement
hotwaterheating
airconditioning
prefarea
furnishingstatus_semi-furnished
furnishingstatus_unfurnished
bedrooms_2
bedrooms_3
bedrooms_4
bedrooms_5
bedrooms_6
stories_2
stories_3
stories_4
bathrooms_2
bathrooms_3
bathrooms_4
parking_1
parking_2
parking_3
```

We will perform the same steps as we did with the first manipulated dataset.

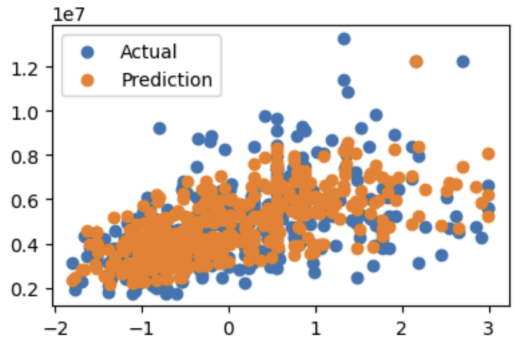
Correlation heatmap:



After looking at the statistics table for this data we drop columns whose p-value of coefficients is more than 0.05, hence, not statistically significant. Those columns are: 'bedrooms_2', 'bedrooms_3', 'bedrooms_4', 'bedrooms_5', 'bedrooms_6', 'parking_3', 'stories_2', 'furnishingstatus_semi_furnished'.

We create a new instance of multiple linear regression, train it, and test it on our new dataset. Here are the results.

```
<<<---------------------------------- Evaluating Multiple Linear Regression Model ----------------------------------->>>

The Coeffecient of the Regresion Model was found to be  [ 403158.2795346   175413.80956248  118392.31252757  178029.33790283
  207047.85845141  382521.59287567  274353.38218844 -200055.32690275
  213617.18847502  341709.54884133  435889.14577506  216328.57553405
  271743.19668313  177233.55343012  293962.8157676 ]
The Intercept of the Regresion Model was found to be  4716708.779342723
```



Corresponding Multiple Regression Equation to be:

price = 4716708.779342723 + 403158.2795346*area + 175413.80956248*mainroad

+ 118392.31252757*guestroom + 178029.33790283*basement

+ 207047.85845141*hotwaterheating + 382521.59287567*airconditioning

+ 274353.38218844*prefarea - 200055.32690275*furnishingstatus_unfurnished

+ 213617.18847502*stories_3 +341709.54884133*stories_4

+ 435889.14577506*bathrooms_2 + 216328.57553405*bathrooms_3

+ 271743.1966831*bathrooms_4 + 177233.55343012*parking_1

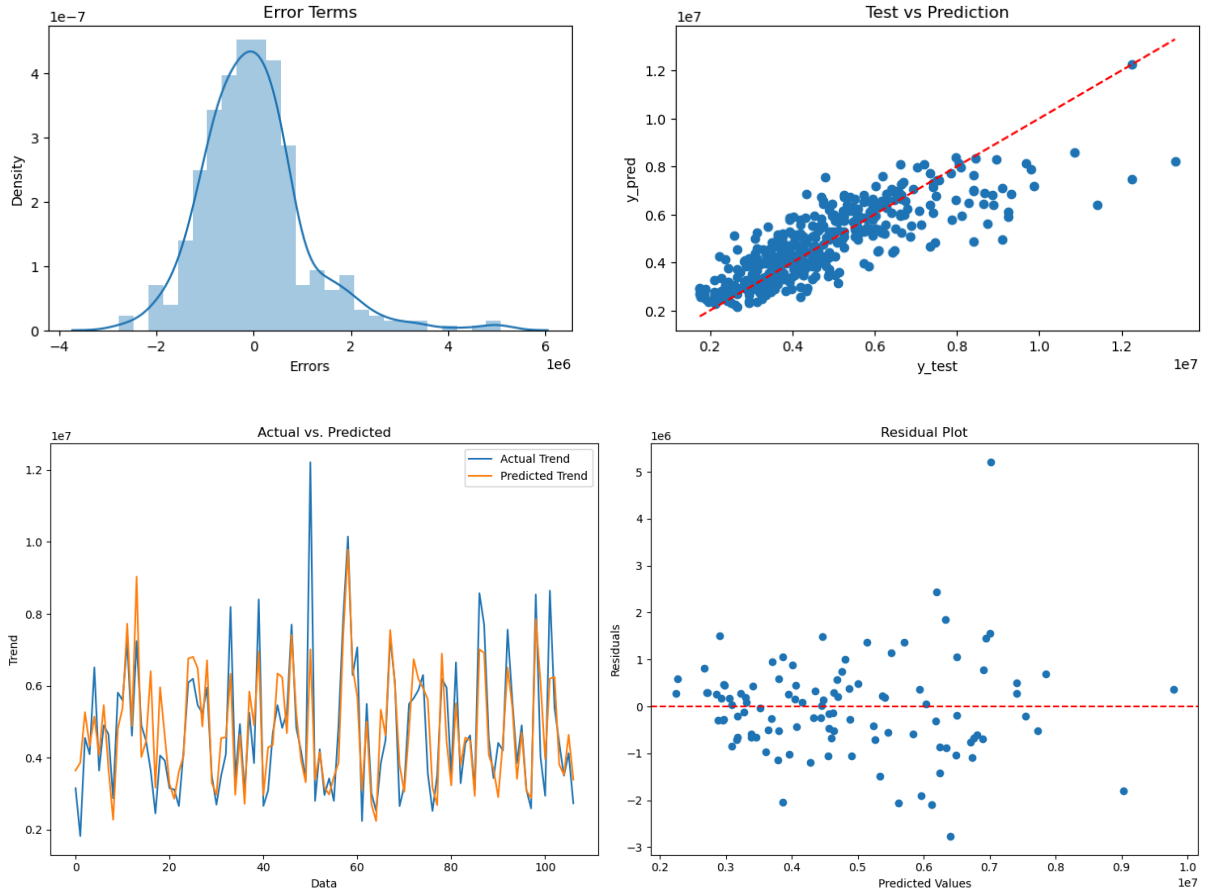
+ 293962.8157676*parking_2

This and the previous equation have the same intercept, while the coefficients are different.

Let's further evaluate our Multiple Linear Regression model.

```
--------------------Training Set Metrics--------------------

R2-Score on Training set ---> 0.6616846377926279
Residual Sum of Squares (RSS) on Training set  ---> 491451702576486.56
Mean Squared Error (MSE) on Training set       ---> 1153642494311.0012
Root Mean Squared Error (RMSE) on Training set ---> 1074077.50852115

--------------------Testing Set Metrics--------------------

R2-Score on Testing set ---> 0.6966862009554988
Residual Sum of Squares (RSS) on Training set  ---> 112336679895240.83
Mean Squared Error (MSE) on Training set       ---> 1049875513039.6339
Root Mean Squared Error (RMSE) on Training set ---> 1024634.3313785822
```

The results are a bit worse but our data is more correct this time.



In this residual plot too, we don't see any patterns forming, which is good.

Hypothesis testing:

Null hypothesis: Our model can't be used to predict our housing price based on specified features.
Alternative: Our model can be used to predict housing prices based on specified features.

As we know we have already selected only attributes that had a p-value less than 0.05. So we reject our null hypothesis and claim that we can use it to predict housing prices.

Project Outcomes:

- ➢ The multiple linear regression model we trained has the moderate explainability power to understand the dataset.
- ➢ The Dataset was quite small with just 545 samples & after preprocessing 2.2% of the data samples were dropped.

- ➢ Visualizing the distribution of data & their relationships, helped us to get some insights.
- ➢ it is safe to use both models we built, just the second one has a better approach to categorical variables as their scores were quite comparable

Conclusion:

- In this analysis, we performed exploratory data analysis, created a multiple linear regression model, and interpreted its results.
- We searched for the influence of area, bedrooms, bathrooms, stories, mainroad, guestroom, basement, hot water heating, air conditioning, parking, preferred area, and furnishing status on the price of a house.
- We have encoded categorical columns. We tried to interpret the coefficients of our first model, which performed better on the test dataset than the second model.
- As our dummy variable columns were not the best fit, we created a second model, which did not do as well as the first one, but it is easier to interpret the result.
- Our analysis revealed key factors significantly impacting housing prices, including area, number of bedrooms, and presence of amenities like air conditioning and 2 bathrooms. Understanding these factors is crucial for stakeholders in the real estate market to make informed decisions. Further research and refinement of predictive models could enhance accuracy and provide deeper insights into the dynamics of housing prices.

It is also important to recognize the profound impact that housing prices can have on society, affecting accessibility, affordability, and equity. By integrating social responsibility into analytical frameworks, the aim is to not only enhance predictive accuracy but also to foster positive societal impact and equitable outcomes in housing markets.

Reference:

Tranmer, M., Murphy, J., Elliot, M., and Pampaka, M. (2020) Multiple Linear Regression (2nd Edition); Cathie Marsh Institute Working Paper 2020-01. https://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/2020-1- multiple-linear-regression.pdf

Kaggle https://www.kaggle.com/datasets/yasserh/housing-prices-dataset

Performed using Python libraries