```
# import major libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

# additional libraries
import warnings
warnings.filterwarnings('ignore')

# importing dataset
df = pd.read_csv('Magicbricks.csv')
```

## Data Assessing

```
# overview data

df.head()
```

```
     Area  BHK  Bathroom Furnishing    Location       District  \
0   950.0    2       2.0  Furnished  Karol Bagh  Central Delhi
1   535.0    2       1.0  Furnished  Karol Bagh  Central Delhi
2  1280.0    3       3.0  Furnished  Karol Bagh  Central Delhi
3  1135.0    3       3.0  Furnished  Karol Bagh  Central Delhi
4  1135.0    3       3.0  Furnished  Karol Bagh  Central Delhi

                                             Locality  Parking
Status  \
0  DDA MIG Flats Prasad Nagar Phase 2, Prasad Nag...      1.0
Ready_to_move
1                               Dev Nagar, Karol Bagh      1.0
Ready_to_move
2                                         Karol Bagh      2.0
Ready_to_move
3                           The Amaryllis, Karol Bagh      2.0
Almost_ready
4                           The Amaryllis, Karol Bagh      2.0
Almost_ready

    Transaction           Type  Per_Sqft     Price
0        Resale      Apartment    8761.0  12500000
1  New_Property      Apartment    7290.0   3900000
2        Resale  Builder_Floor   14092.0  15000000
3        Resale      Apartment   22222.0  25000000
4        Resale      Apartment   22222.0  25000000
```

```
df.shape
```

```
(1214, 13)
```

# Data Card

A dataset contain 1214 rows and have columns

## Types of Error:

- **completeness**
- **Validity** --> dtypes(example: no of chirldren in float), duplicacy issue(example:patient_id), salary(-10000), age in negative
- **Accuracy** --> body weight (13kg of adult), age 190
- **Inconsistency** --> new york city, nyc

## Types of Data

- **Dirty data** --> completeness validdity accuracy inconsistency
- **Messy Data** --> structural issues --> exapmle : pivot tables, sparsity issue

```
# seeking information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1214 entries, 0 to 1213
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Area         1214 non-null   float64
 1   BHK          1214 non-null   int64  
 2   Bathroom     1212 non-null   float64
 3   Furnishing   1214 non-null   object 
 4   Location     1214 non-null   object 
 5   District     1214 non-null   object 
 6   Locality     1214 non-null   object 
 7   Parking      1182 non-null   float64
 8   Status       1214 non-null   object 
 9   Transaction  1214 non-null   object 
 10  Type         1209 non-null   object 
 11  Per_Sqft     973 non-null    float64
 12  Price        1214 non-null   int64  
dtypes: float64(4), int64(2), object(7)
memory usage: 123.4+ KB
```

- the dataset contains null values indicating completeness issue.
- bathroom and parking columns are stored at float thought they represent counts -- a validity error.
- the dataset includes 6 numerical columns and 7 categorical (object-type) column requring proper preprocessing before analysis.

```
# seeking description
df.describe()
```

```
                Area           BHK     Bathroom      Parking
Per_Sqft  \
count   1214.000000  1214.000000  1212.000000  1182.000000
973.000000
mean    1451.850751     2.778418     2.523927     1.708122
15574.885920
std     1586.472855     0.946811     1.017723     5.717177
21574.389007
min       28.000000     1.000000     1.000000     1.000000
1259.000000
25%      800.000000     2.000000     2.000000     1.000000
6154.000000
50%     1150.000000     3.000000     2.000000     1.000000
10838.000000
75%     1620.000000     3.000000     3.000000     2.000000
17647.000000
max    24300.000000    10.000000     7.000000   114.000000
183333.000000

               Price
count  1.214000e+03
mean   2.079898e+07
std    2.561308e+07
min    1.000000e+06
25%    5.600000e+06
50%    1.400000e+07
75%    2.500000e+07
max    2.400000e+08
```

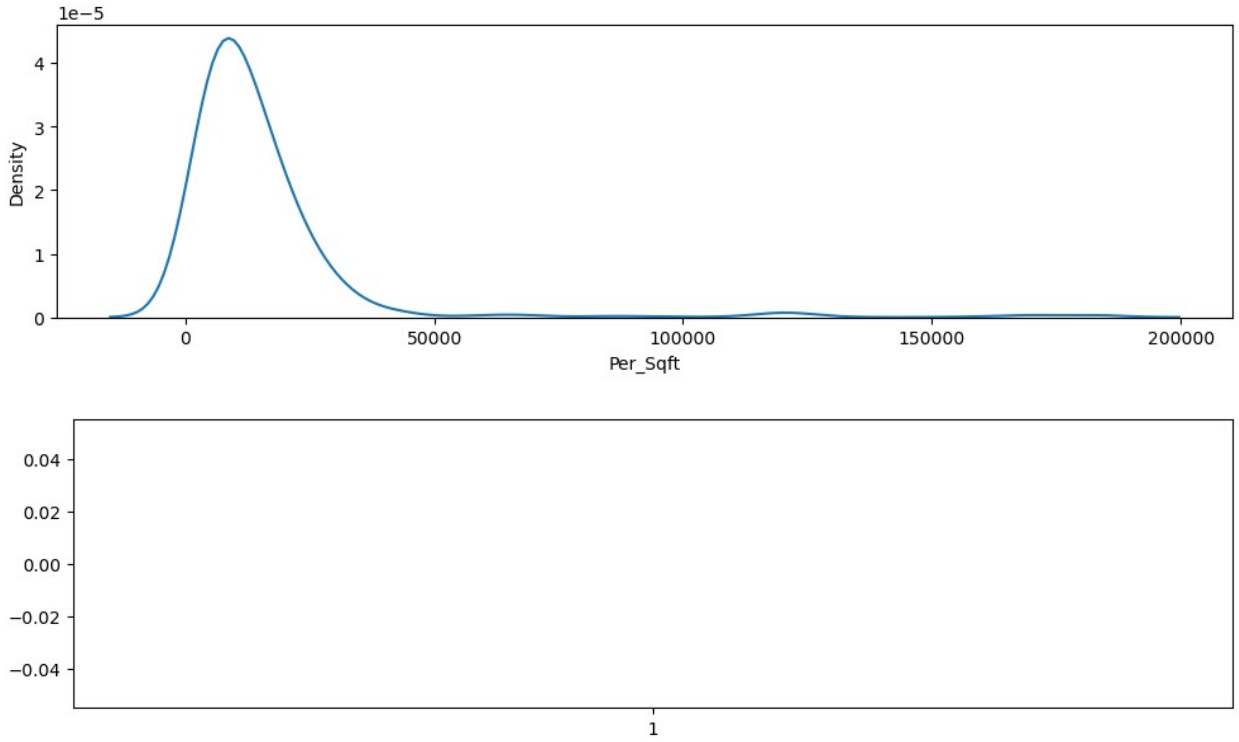
- points
- Accuracy --> min-value of area 28 sqft dicy and max 24300 sqft

```
# COMPLETENESS
df.isnull().sum().sum()
# percentage
df.isnull().mean()*100
```

```
Area             0.000000
BHK              0.000000
Bathroom         0.164745
Furnishing       0.000000
Location         0.000000
District         0.000000
Locality         0.000000
Parking          2.635914
Status           0.000000
Transaction      0.000000
Type             0.411862
Per_Sqft        19.851730
```

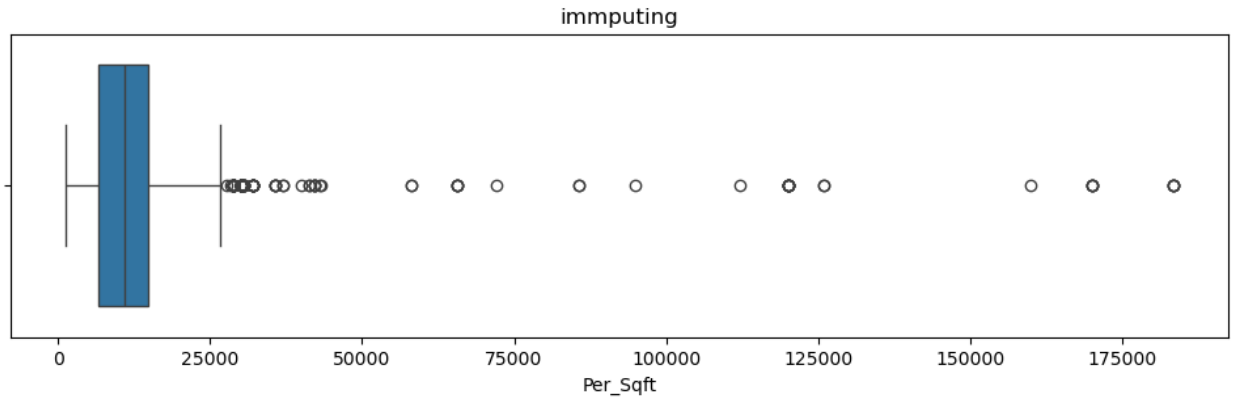
```
Price            0.000000
dtype: float64
```

```
df.Per_Sqft.describe()
# VISUALIZATION
plt.figure(figsize=(12,3))
sns.kdeplot(data=df, x='Per_Sqft')
plt.show()
plt.figure(figsize=(12,3))
plt.boxplot(data=df, x='Per_Sqft')
plt.show()

print('skewness',df.Per_Sqft.skew())
```

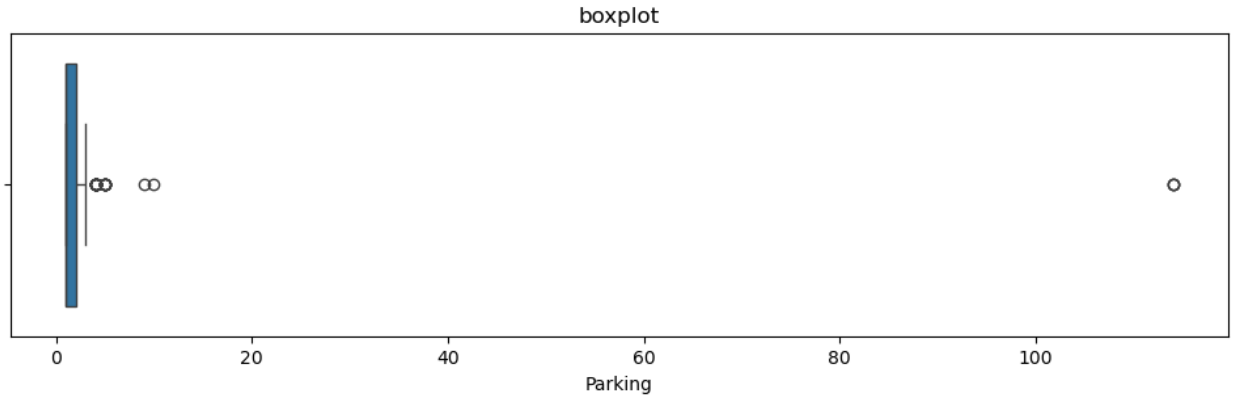



```
skewness 5.264871991245157
```

```
plt.figure(figsize=(12,3))
plt.title('immputing')
sns.boxplot(x=df.Per_Sqft.fillna(df.Per_Sqft.median()))

<Axes: title={'center': 'immputing'}, xlabel='Per_Sqft'>
```

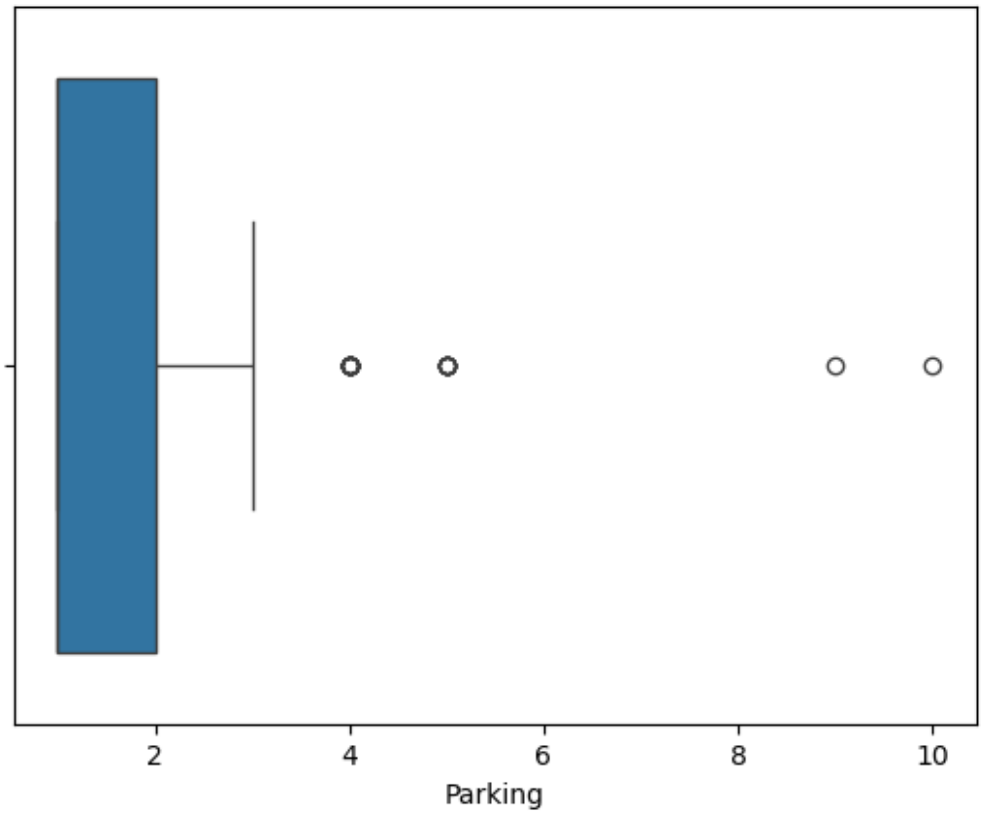


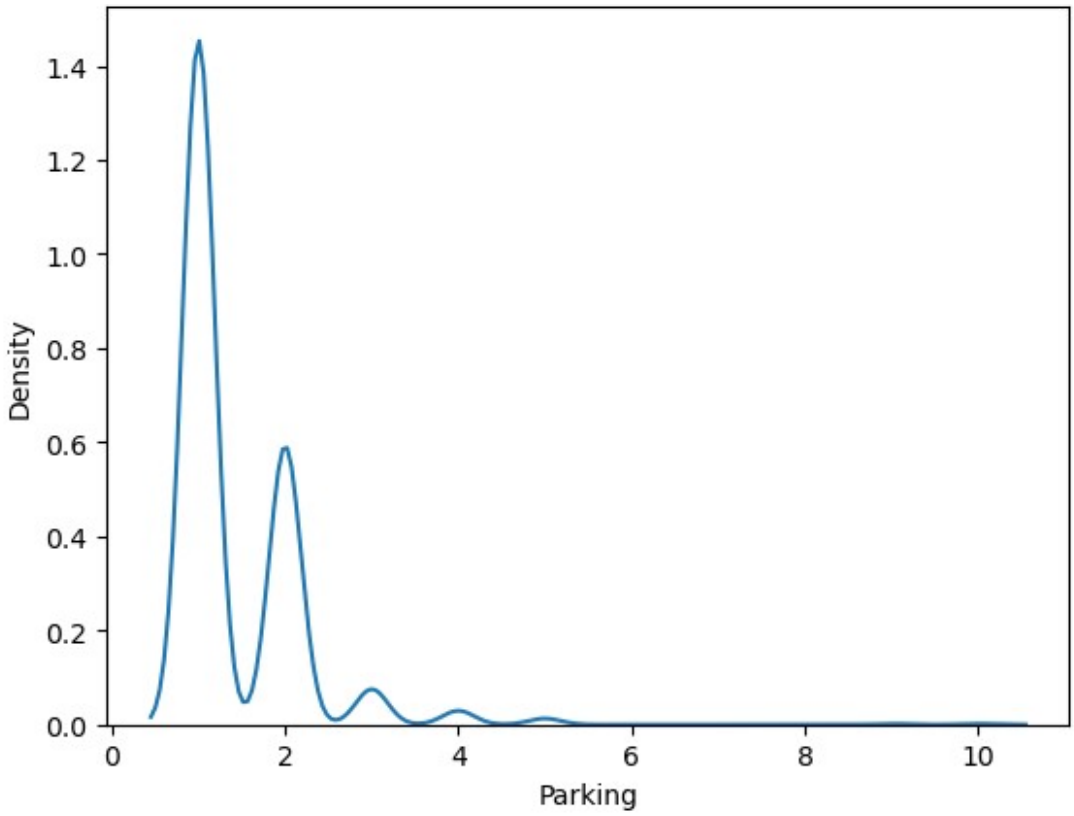

```
df.columns

Index(['Area', 'BHK', 'Bathroom', 'Furnishing', 'Location',
'District',
       'Locality', 'Parking', 'Status', 'Transaction', 'Type',
'Per_Sqft',
       'Price'],
      dtype='object')

df.Per_Sqft = df.Per_Sqft.fillna(df.Price/df.Area)

df.Per_Sqft.isnull().sum()

np.int64(0)

# outlier
plt.figure(figsize=(12,3))
plt.title('boxplot')
sns.boxplot(x=df.Parking)
plt.show()
```




```
df[df.Parking>100]

df.Parking = np.where(df.Parking>100,1,df.Parking)
```

```
sns.boxplot(x=df.Parking)
plt.show()
sns.kdeplot(x=df.Parking)
plt.show()
```

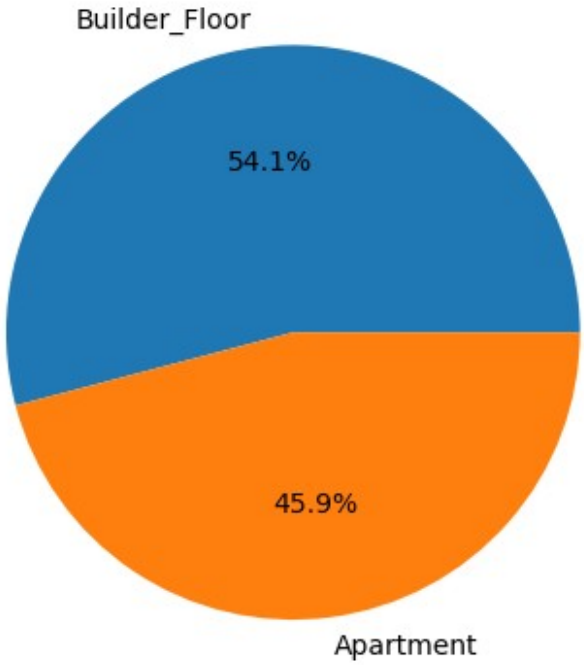





```
df.Parking = df.Parking.fillna(1)
df.Parking = df.Parking.astype(int)
df.Parking.sample(5)
```

```
233    1
220    1
845    1
475    1
180    1
Name: Parking, dtype: int64
```

```
# Type
df.isnull().sum()
temp = df.Type.value_counts().reset_index()
plt.pie(temp['count'],labels=temp.Type,autopct='%1.1f%%')
plt.show()
temp
```




```
            Type  count
0  Builder_Floor    654
1      Apartment    555
```

```
df.Type.fillna('Builder_Floor',inplace=True)
df.Type.mode().values
```

```
array(['Builder_Floor'], dtype=object)
```

```
df.isnull().sum().sum()
```

```
np.int64(2)
```

```
df.dropna(inplace=True)
```

```
df.shape
```

```
(1212, 13)
```

```
# numerical columns
# categorical columns
df.columns
num = list(df.describe().columns)
cat = []
for i in df.columns:
    if i not in num:
        cat.append(i)
```

```
print('numerical\t',num)
print('categorical\t',cat)
```

```
numerical   ['Area', 'BHK', 'Bathroom', 'Parking', 'Per_Sqft',
'Price']
categorical      ['Furnishing', 'Location', 'District', 'Locality',
'Status', 'Transaction', 'Type']
```

```
# univariate analysis
```

## Potential issue

```
df[df.Area>10000]
```

```
        Area  BHK  Bathroom      Furnishing           Location
District  \
429  22050.0    4       4.0  Semi-Furnished    Greater Kailash  South
Delhi
431  22050.0    4       4.0  Semi-Furnished    Greater Kailash  South
Delhi
515  10350.0    4       7.0  Semi-Furnished     Friends Colony  South
Delhi
603  24300.0    4       5.0  Semi-Furnished              Saket  South
Delhi
806  14220.0    3       3.0  Semi-Furnished      Paschim Vihar   West
Delhi
835  17010.0    3       3.0  Semi-Furnished       Punjabi Bagh   West
Delhi
978  11050.0    3       3.0     Unfurnished  Chittaranjan Park  South
Delhi

                              Locality  Parking         Status
Transaction  \
429                  Greater Kailash 1        2   Almost_ready
New_Property
431                  Greater Kailash 1        2   Almost_ready
New_Property
515  Maharani Bagh, New Friends Colony        3  Ready_to_move
New_Property
603                              Saket        2  Ready_to_move
Resale
806             Paschim Vihar Block B4        1  Ready_to_move
New_Property
835                  Punjabi Bagh West        2  Ready_to_move
Resale
978                  Chittaranjan Park        1  Ready_to_move
New_Property
```

```
                Type  Per_Sqft      Price
429  Builder_Floor    30556.0   51000000
431  Builder_Floor    30556.0   51000000
515      Apartment    15459.0  160000000
603  Builder_Floor    12500.0   51000000
806  Builder_Floor    10943.0   27500000
835  Builder_Floor    15278.0   25000000
978  Builder_Floor    12916.0   18500000
```

```
22050.0*30556.0
```

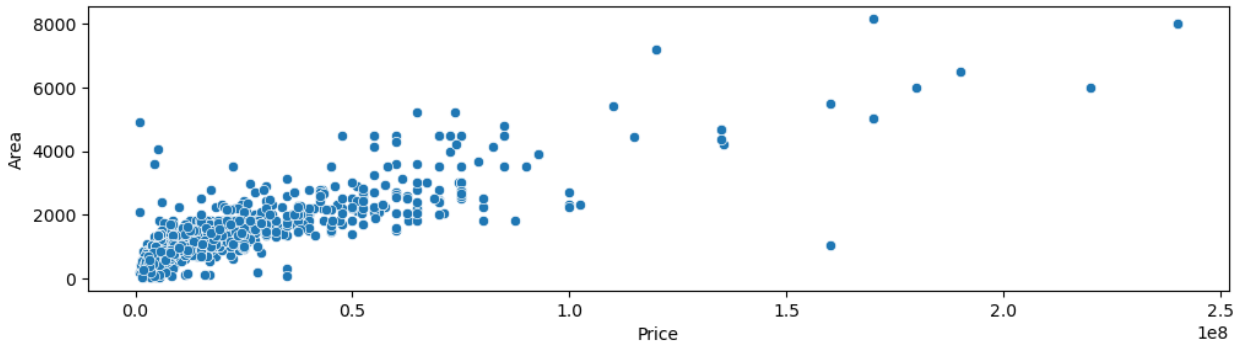
```
673759800.0
```

```
(673759800.0/51000000)/10
```

```
1.3210976470588236
```

```
df.Area = np.where(df.Area>10000,df.Area/10,df.Area)
```

```
# bivariate analysis
plt.figure(figsize=(12,3))
sns.scatterplot(data=df,x='Price',y='Area')
plt.show()
```




```
# plotly

px.scatter(df,x='Price',y='Area',hover_data=['Area','Location','Price'
],height=500)
```


```
# plotly

px.scatter(df,x='Price',y='Area',

hover_data=['Area','Location','Price'],height=500,color='Price',title=
'Area')
```

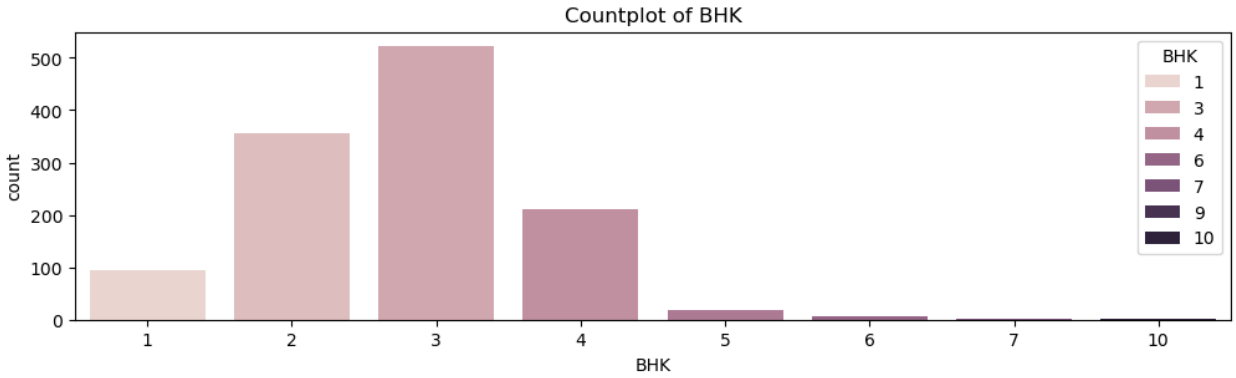

```
# univariate analysis

# bhk
plt.figure(figsize=(12,3))
sns.countplot(data=df,x='BHK',hue='BHK')
plt.title('Countplot of BHK')
temp = df.BHK.value_counts().reset_index().head()
```

```
temp
px.pie(temp,names='BHK',values='count',
       color_discrete_sequence = px.colors.sequential.Blues,
       height=400,title='Distribution of BHK')
```


```
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"domain":
{"x":[0,1],"y":[0,1]},"hovertemplate":"BHK=%{label}<br>count=%
{value}<extra></extra>","labels":
[3,2,4,1,5],"legendgroup":"","name":"","showlegend":true,"type":"pie",
"values":[523,356,212,94,19]}],"layout":{"height":400,"legend":
{"tracegroupgap":0},"piecolorway":
["rgb(247,251,255)","rgb(222,235,247)","rgb(198,219,239)","rgb(158,202
,225)","rgb(107,174,214)","rgb(66,146,198)","rgb(33,113,181)","rgb(8,8
1,156)","rgb(8,48,107)"],"template":{"data":{"bar":[{"error_x":
{"color":"#2a3f5f"},"error_y":{"color":"#2a3f5f"},"marker":{"line":
{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpo
lar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"
carpet":[{"aaxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":
{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","min
orgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"ch
oropleth":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":
{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],
[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],
[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],
[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],
[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":
{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],
"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
```

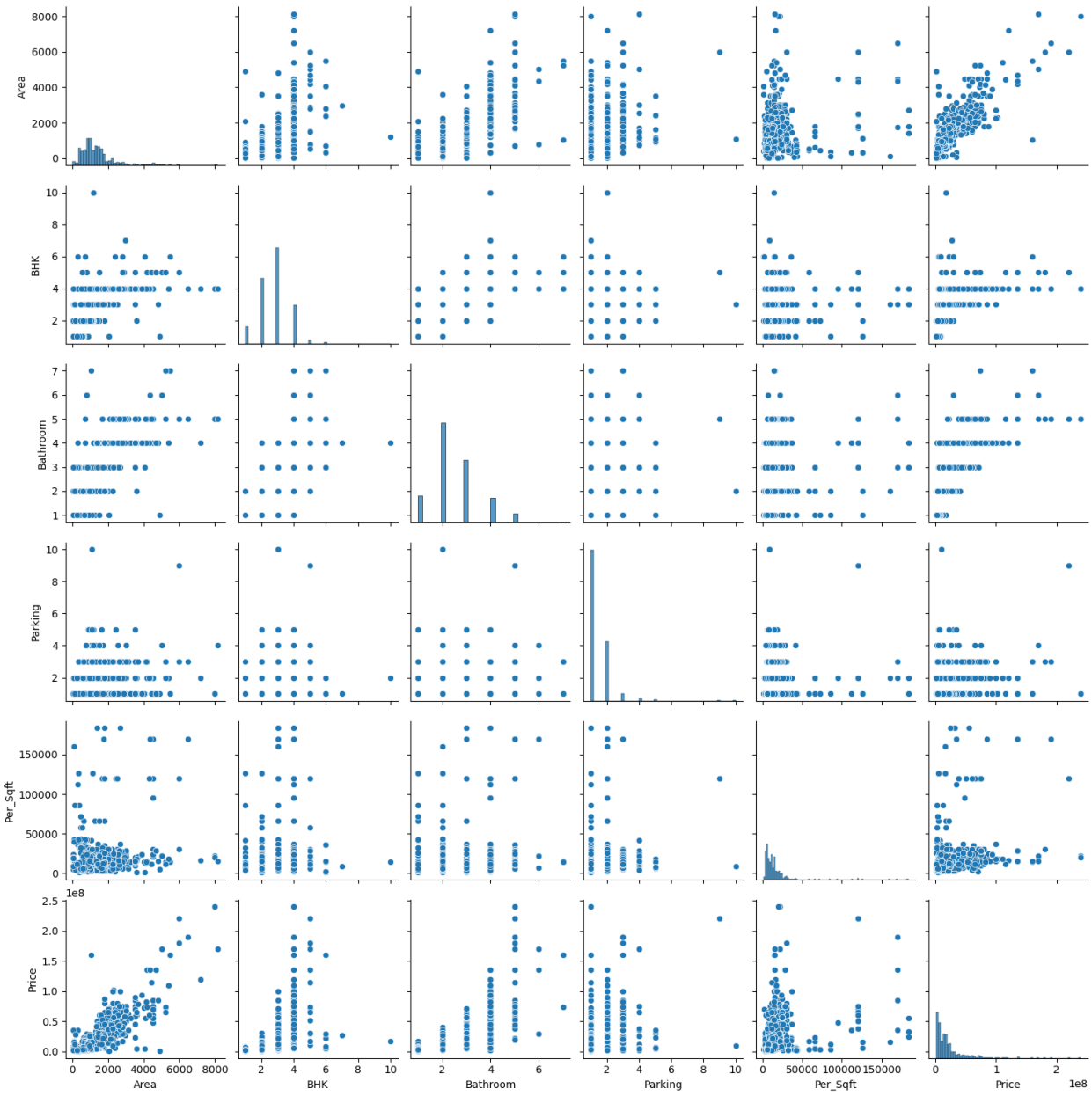
```
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":
{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":
{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":
[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":
{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"sc
atter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":
{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":
[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar"
:[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl
":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterterna
ry":[{"marker":{"colorbar":
{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":
[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],
[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":
{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":
{"color":"#C8D4E3"},"line":
{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":
{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers
":"strict","coloraxis":{"colorbar":
{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":
[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],
[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],
[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":
```

```
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":
[[0,"#0d0887"],[0.1111111111111111,"#46039f"],
[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],
[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],
[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],
[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":
["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692"
,"#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":
{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlake
s":true,"showland":true,"subunitcolor":"white"},"hoverlabel":
{"align":"left"},"hovermode":"closest","mapbox":
{"style":"light"},"paper_bgcolor":"white","plot_bgcolor":"#E5ECF6","po
lar":{"angularaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","radialaxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":
{"xaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"yaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
,"zaxis":
{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"lineco
lor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}
},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":
{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF
6","caxis":
{"gridcolor":"white","linecolor":"white","ticks":""}},"title":
{"x":5.0e-2},"xaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":
{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"",
"title":
{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}},"title":
{"text":"Distribution of BHK"}}}
```




```
# multivariate analysis

sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x29ce4096f90>
```



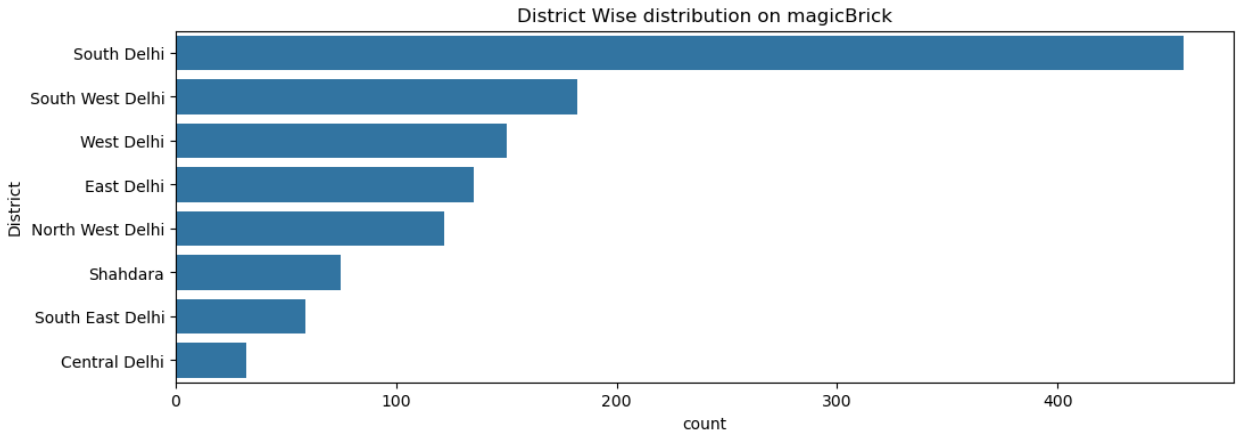

```
# corr
# heat map
plt.figure(figsize=(10,4))
sns.heatmap(df.corr(numeric_only=True),cmap='Blues',annot=True)
```

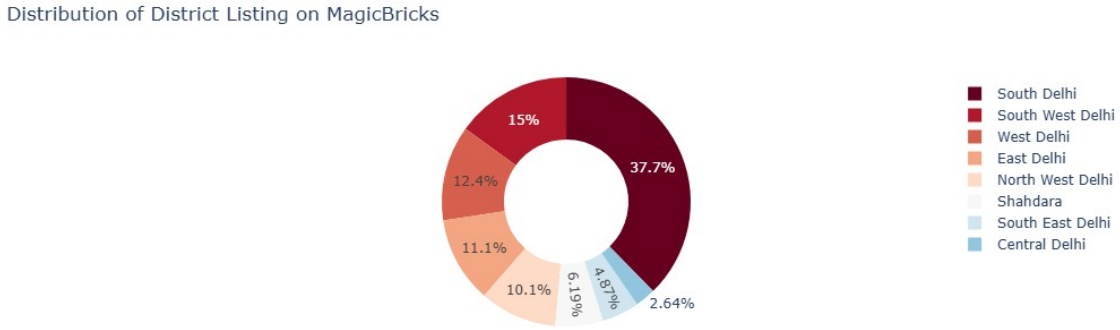
```
<Axes: >
```


```
df.District.value_counts().reset_index()
```

| | District | count |
|---|---|---|
| 0 | South Delhi | 457 |
| 1 | West Delhi | 150 |
| 2 | East Delhi | 135 |
| 3 | North West Delhi | 122 |
| 4 | South West Delhi | 95 |
| 5 | South West Delhi | 87 |
| 6 | Shahdara | 75 |
| 7 | South East Delhi | 59 |
| 8 | Central Delhi | 32 |

```
df.District = df.District.str.strip()

temp = df.District.value_counts().reset_index()
plt.figure(figsize=(12,4))
sns.countplot(data=df,y='District',order=temp.District)
plt.title('District Wise distribution on magicBrick')
plt.show()

px.pie(temp,values='count',names='District',height=400,
       color_discrete_sequence=px.colors.sequential.RdBu,hole=0.5,
       title='Distribution of District Listing on MagicBricks').show()
temp
```



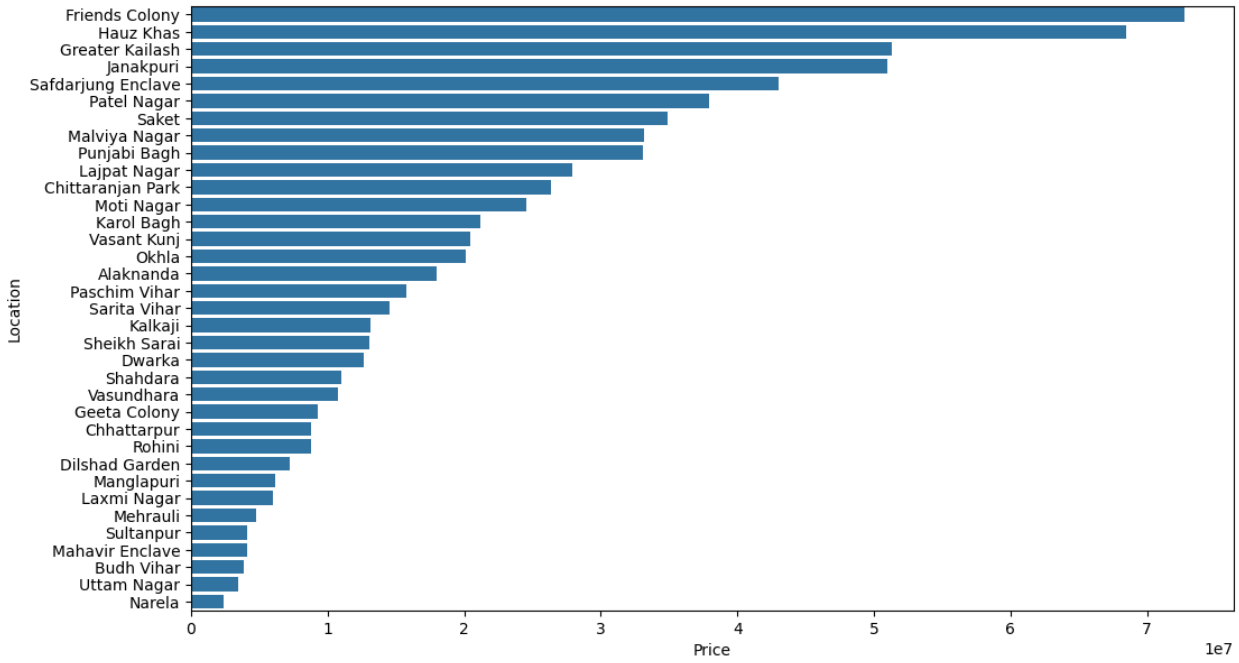




| | District | count |
|---|---|---|
| 0 | South Delhi | 457 |
| 1 | South West Delhi | 182 |
| 2 | West Delhi | 150 |
| 3 | East Delhi | 135 |
| 4 | North West Delhi | 122 |
| 5 | Shahdara | 75 |
| 6 | South East Delhi | 59 |
| 7 | Central Delhi | 32 |

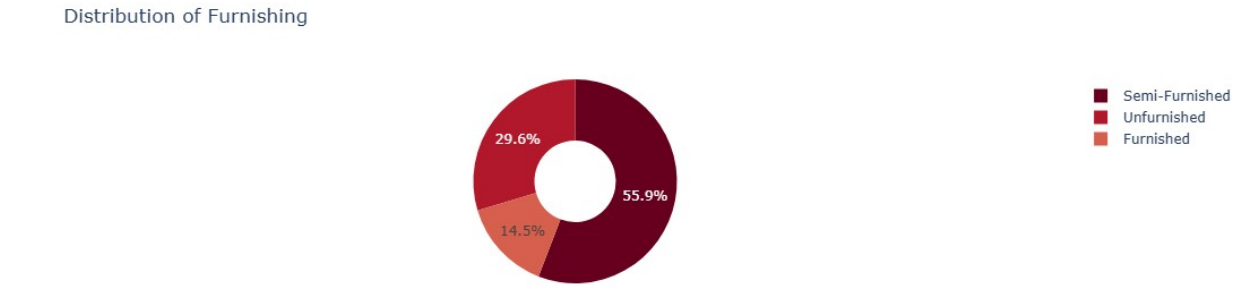
```
# location
# price
# bivariate analysis

df.loc[:,['Location','Price']].sample(5)
```

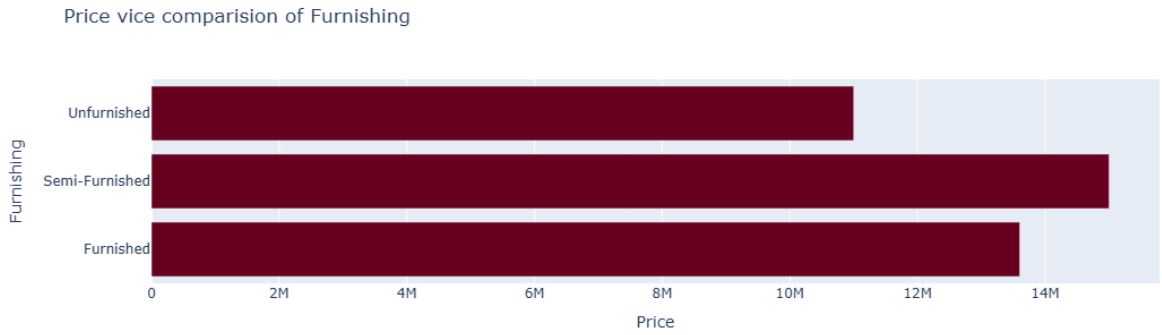
| | Location | Price |
|---|---|---|
| 183 | Karol Bagh | 24000000 |
| 1131 | Sultanpur | 4500000 |
| 428 | Malviya Nagar | 57500000 |
| 766 | Dwarka | 6800000 |
| 234 | Vasundhara | 16500000 |

```
temp = df.groupby('Location')
['Price'].mean().sort_values(ascending=False).reset_index()
plt.figure(figsize=(12,7))
sns.barplot(data=df,y='Location',x='Price',
            ci=False,order = temp.Location)
plt.show()
```




```
temp = df.Furnishing.value_counts().reset_index()

px.pie(temp,names = 'Furnishing'
       ,values='count',title='Distribution of Furnishing',hole=0.4,
       color_discrete_sequence=px.colors.sequential.RdBu).show()

temp2 = df.groupby('Furnishing')['Price'].median().reset_index()

px.bar(temp2, x='Price',y='Furnishing',title='Price vice comparision
of Furnishing',
       color_discrete_sequence=px.colors.sequential.RdBu).show()
```




# Conclusion

The MagicBrick data analysis conducted using Python highlights the effectiveness of data-driven techniques in extracting meaningful insights from complex datasets. Through systematic data cleaning, transformation, and exploratory analysis, the dataset was refined to ensure accuracy, consistency, and reliability. Visualization and statistical summaries enabled the identification of key patterns, trends, and anomalies, supporting clearer interpretation and informed decision-making. Overall, the Python-based analysis establishes a robust analytical foundation for the MagicBrick project, enabling scalable modeling, performance optimization, and future integration of advanced techniques such as predictive analytics and machine learning.