Experiment 1: Exploratory Data Analysis

- Name: **Vinayak Tripathi**
- UID: **2019110067**
- Class: **BE ETRX**
- Subject: **DA Lab**

Objective: Perform EDA such as number of data samples, number of features, number of classes, number of data samples per class, removing missing values, conversion to numbers, using seaborn library to plot different graphs.

In [ ]:
```
# Importing the libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [ ]:
```
data = pd.read_csv('/content/drive/MyDrive/data/dataset/train.csv',index_col=[0])
data.head()
```

Out[ ]:

Property_ID	Property_Type	Property_Area	Number_of_Windows	Number_of_Doors	Furnishing
0x21e3	Apartment	106	NaN	1	Semi_Furnished
0x68d4	Apartment	733	2.0	2	Unfurnished
0x7d81	Apartment	737	4.0	2	Fully Furnished
0x7a57	Apartment	900	3.0	2	Unfurnished
0x9409	Bungalow	2238	14.0	6	Fully Furnished

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 39499 entries, 0x21e3 to 0x8cb4
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   Property_Type          39499 non-null  object
 1   Property_Area          39499 non-null  int64
 2   Number_of_Windows      37845 non-null  float64
 3   Number_of_Doors        39499 non-null  int64
 4   Furnishing             38457 non-null  object
 5   Frequency_of_Powercuts 38116 non-null  float64
```

```
 6   Power_Backup            39499 non-null  object
 7   Water_Supply            39499 non-null  object
 8   Traffic_Density_Score   39499 non-null  float64
 9   Crime_Rate              38712 non-null  object
 10  Dust_and_Noise          38280 non-null  object
 11  Air_Quality_Index       39499 non-null  float64
 12  Neighborhood_Review     39499 non-null  float64
 13  Habitability_score      39499 non-null  float64
dtypes: float64(6), int64(2), object(6)
memory usage: 4.5+ MB
```

In the above command we can say that there are 14 columns in the data set. The types of the data present are object, float and integer. Bty closely oberserving the non null count we can also infer that there are also null values in the dataset

In [ ]:
```
# Here using this command we will see the number of the
data.isnull().sum()
```

Out[ ]:
```
Property_Type               0
Property_Area               0
Number_of_Windows        1654
Number_of_Doors             0
Furnishing               1042
Frequency_of_Powercuts   1383
Power_Backup                0
Water_Supply                0
Traffic_Density_Score       0
Crime_Rate                787
Dust_and_Noise           1219
Air_Quality_Index           0
Neighborhood_Review         0
Habitability_score          0
dtype: int64
```

In [ ]:
```
print("The data size with null values",data.shape)
data.dropna(inplace=True) # this command is use to remove the null values of the dat
print("The data sizen after dropping null values",data.shape)
```

```
The data size with null values (39499, 14)
The data sizen after dropping null values (33772, 14)
```

In [ ]:
```
data.isnull().sum()
```

Out[ ]:
```
Property_Type            0
Property_Area            0
Number_of_Windows        0
Number_of_Doors          0
Furnishing               0
Frequency_of_Powercuts   0
Power_Backup             0
Water_Supply             0
Traffic_Density_Score    0
Crime_Rate               0
Dust_and_Noise           0
Air_Quality_Index        0
Neighborhood_Review      0
Habitability_score       0
dtype: int64
```

In [ ]:
```
print(data.nunique(),"\n") # getting the number unique values for each columns
print("Propety Type",data.Property_Type.unique().tolist())
```

```
print("Power Backup",data.Power_Backup.unique().tolist())
print("Water Supply",data.Water_Supply.unique().tolist())
print("Furnishing",data.Furnishing.unique().tolist())
print("CrimeRate",data.Crime_Rate.unique().tolist())
print("Dust and Noise",data.Dust_and_Noise.unique().tolist())
```

```
Property_Type               6
Property_Area            4278
Number_of_Windows          16
Number_of_Doors             6
Furnishing                  3
Frequency_of_Powercuts      4
Power_Backup                3
Water_Supply                5
Traffic_Density_Score     761
Crime_Rate                  4
Dust_and_Noise              3
Air_Quality_Index         460
Neighborhood_Review       412
Habitability_score       5858
dtype: int64

Propety Type ['Apartment', 'Bungalow', 'Single-family home', 'Duplex', '#R%$G&867',
'Container Home']
Power Backup ['No', 'Yes', 'NOT MENTIONED']
Water Supply ['Once in a day - Evening', 'Once in a day - Morning', 'All time', 'NOT
MENTIONED', 'Once in two days']
Furnishing ['Unfurnished', 'Fully Furnished', 'Semi_Furnished']
CrimeRate ['Well below average', 'Slightly below average', 'Well above average', 'Sl
ightly above average']
Dust and Noise ['Medium', 'High', 'Low']
```

Till now we have explored the data columns there data variety. We have also remove the null values. Now we will try to explore the data set using some grpahs. We have also explored the unique values in the data set

In [ ]:
```
# this command will give us the 5 point summary
data.describe()
```

Out[ ]:

	Property_Area	Number_of_Windows	Number_of_Doors	Frequency_of_Powercuts	Traffic_Dens
count	33772.000000	33772.000000	33772.000000	33772.000000	3377
mean	1396.554601	3.926152	2.336255	0.484247	
std	1968.196445	2.693232	1.182115	0.752413	
min	100.000000	0.000000	1.000000	0.000000	
25%	495.000000	2.000000	1.000000	0.000000	
50%	932.000000	4.000000	2.000000	0.000000	
75%	1776.000000	5.000000	3.000000	1.000000	
max	28057.000000	15.000000	6.000000	3.000000	

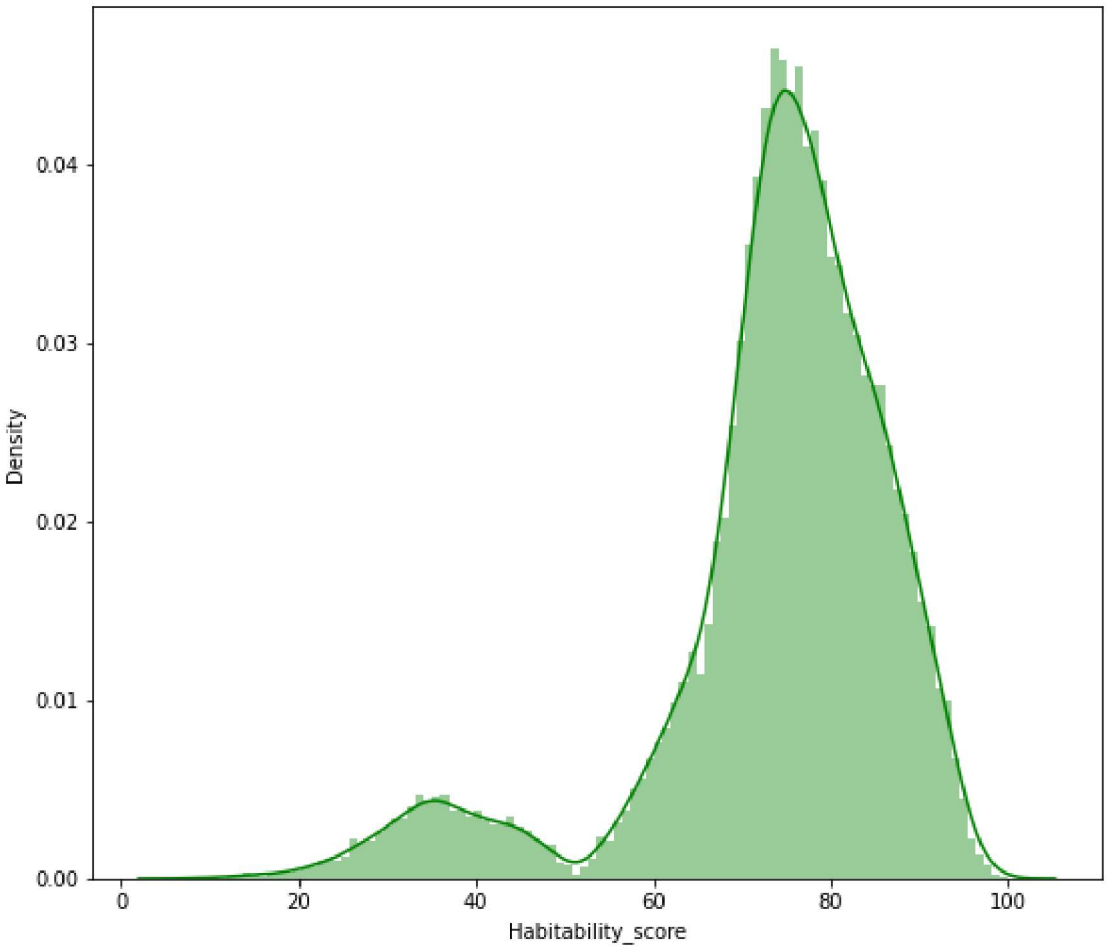
Now we will do the data visualization

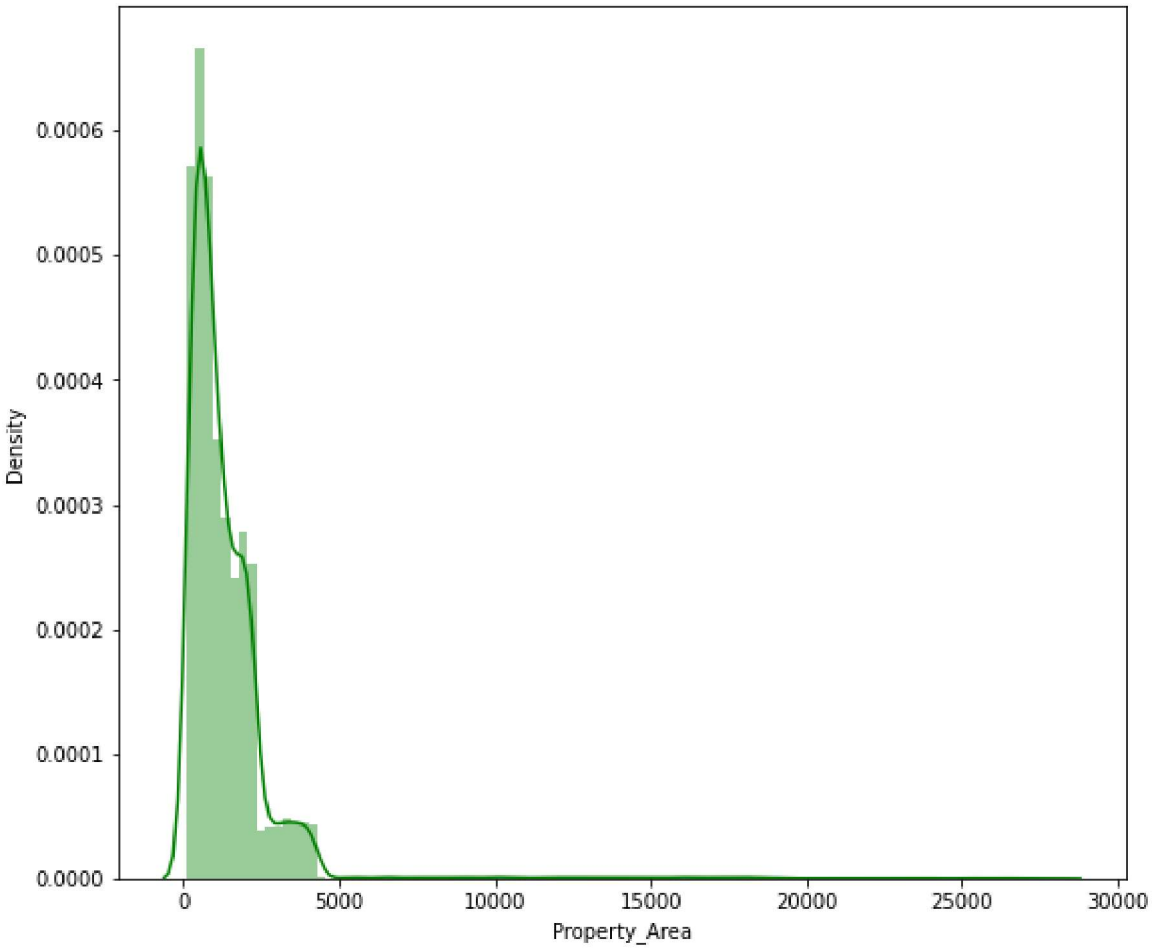
In [ ]:
```
# in this we are plotting the distribution of the Habitability score and Property ar
# THis is just to understand how the these two data spread around the mean and verif
```

```
# from the 5 point summary
plt.figure(figsize=(9, 8))
sns.distplot(data['Habitability_score'], color='g', bins=100, hist_kws={'alpha': 0.4
plt.figure(figsize=(9, 8))
sns.distplot(data['Property_Area'], color='g', bins=100, hist_kws={'alpha': 0.4})
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexib
ility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please
adapt your code to use either `displot` (a figure-level function with similar flexib
ility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f64ab2e6710>
```

Out[ ]:



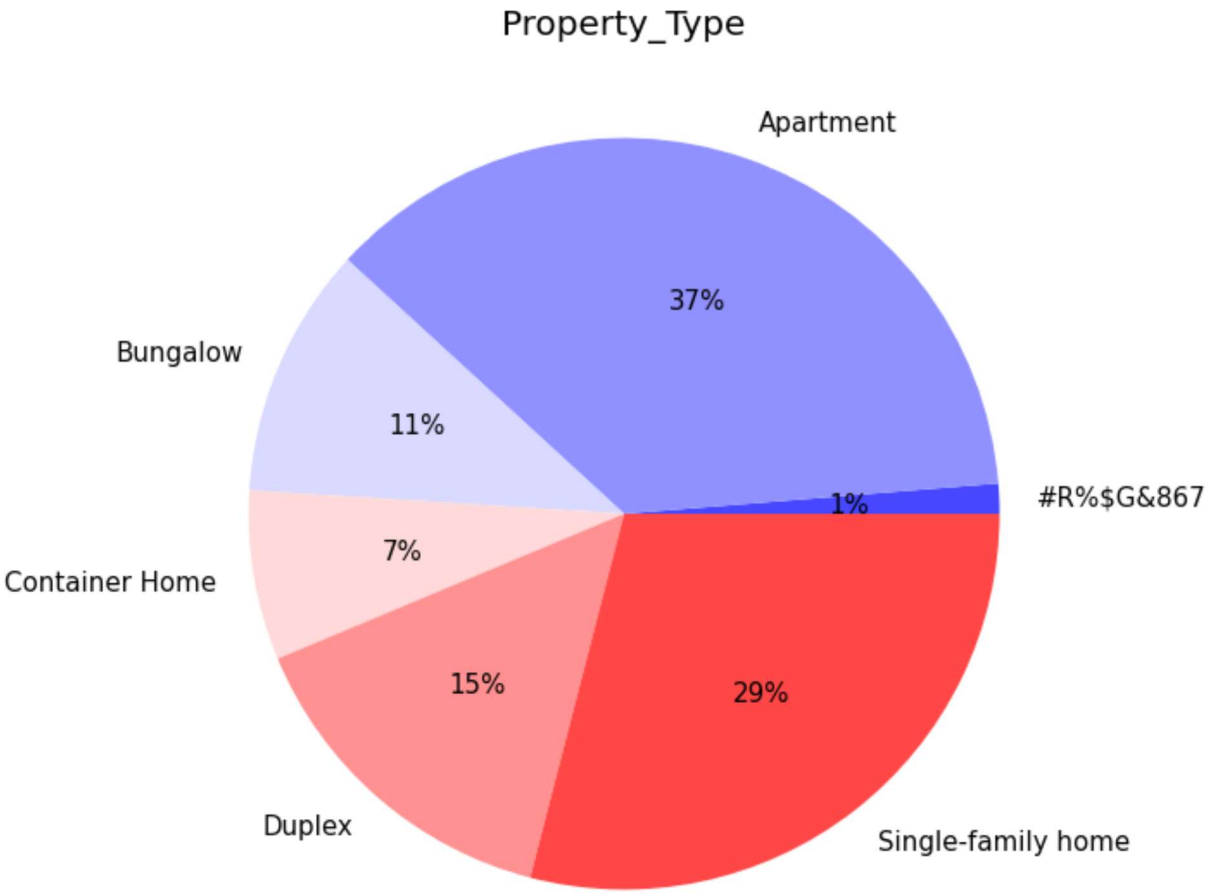


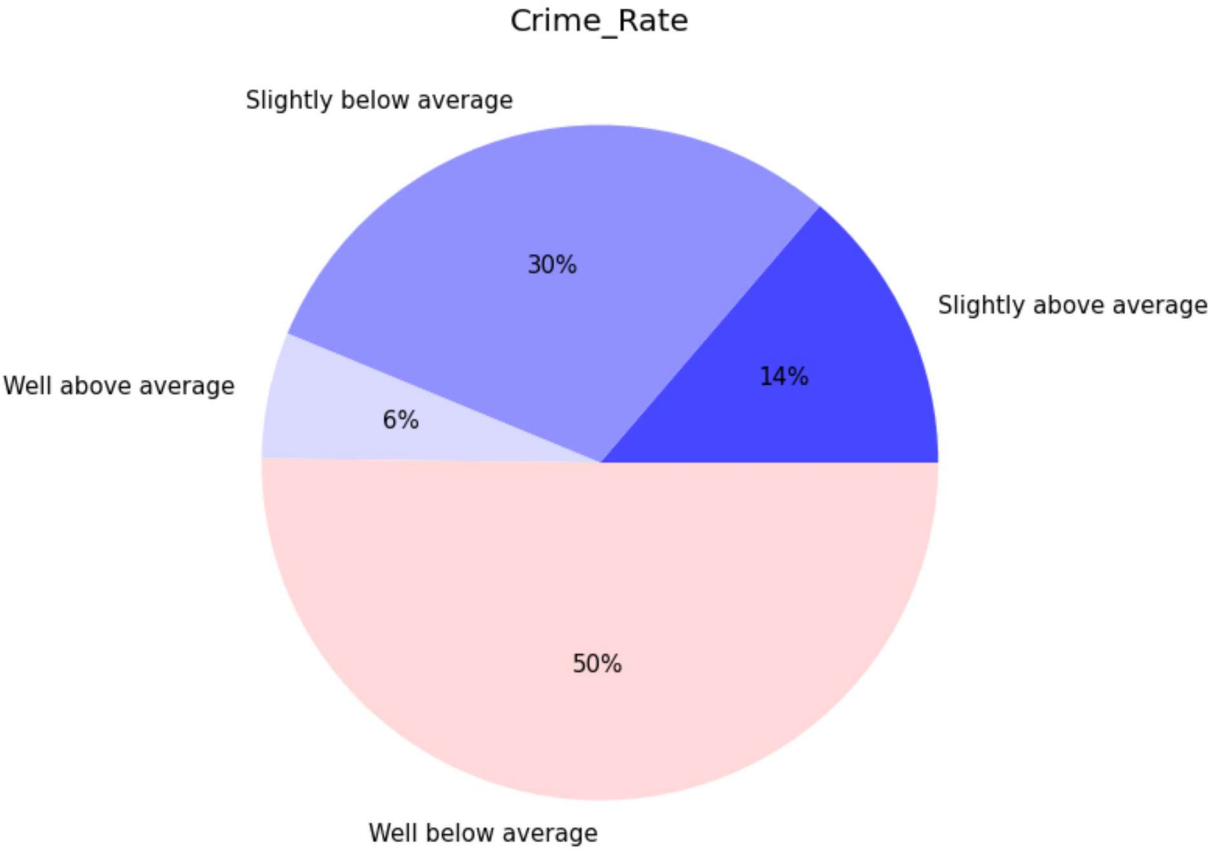
In [ ]:

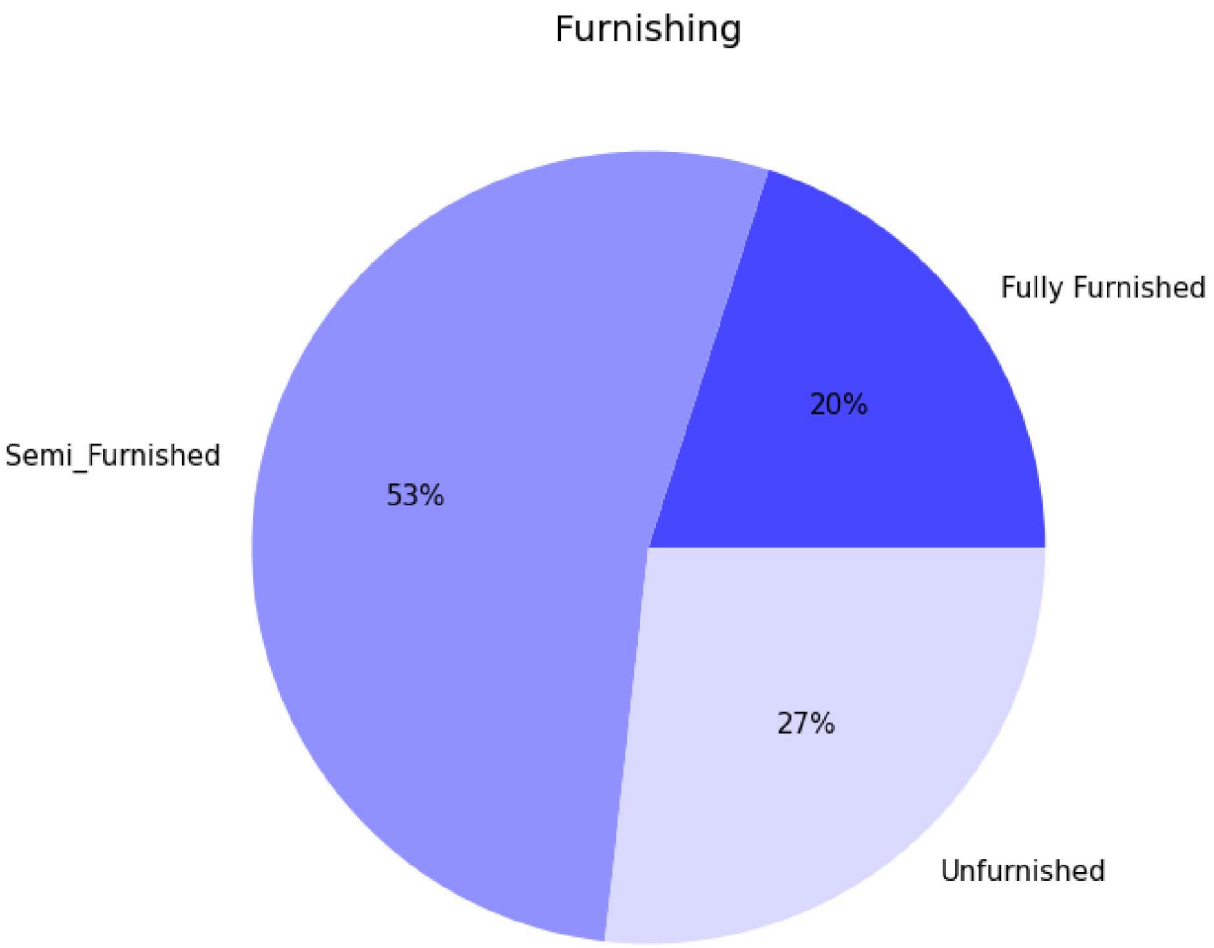
```
plt.figure(figsize=(10, 10))
palette_color = sns.color_palette('bwr')
plt.pie(data.groupby('Property_Type').size(),labels = data.groupby('Property_Type').
plt.title('Property_Type',fontsize=20)

plt.figure(figsize=(10, 10))
plt.pie(data.groupby('Crime_Rate').size(),labels = data.groupby('Crime_Rate').size()
plt.title('Crime_Rate',fontsize=20)

plt.figure(figsize=(10, 10))
plt.pie(data.groupby('Furnishing').size(),labels = data.groupby('Furnishing').size()
plt.title('Furnishing',fontsize=20)
```

Out[ ]:

```
Text(0.5, 1.0, 'Furnishing')
```


Property_Type
Apartment
37%
Bungalow
11%
#R%$G&867
1%
Container Home
7%
15%
29%
Duplex
Single-family home


Crime_Rate
Slightly below average
30%
Slightly above average
Well above average
14%
6%
50%
Well below average



In [ ]:

```
# We will be selecting only some of the numberical features of the dataset
df_num = data.loc[:,['Property_Area','Number_of_Windows',"Number_of_Doors",'Traffic_
# df_num = data.select_dtypes(include = 'number')
df_num.head()
```

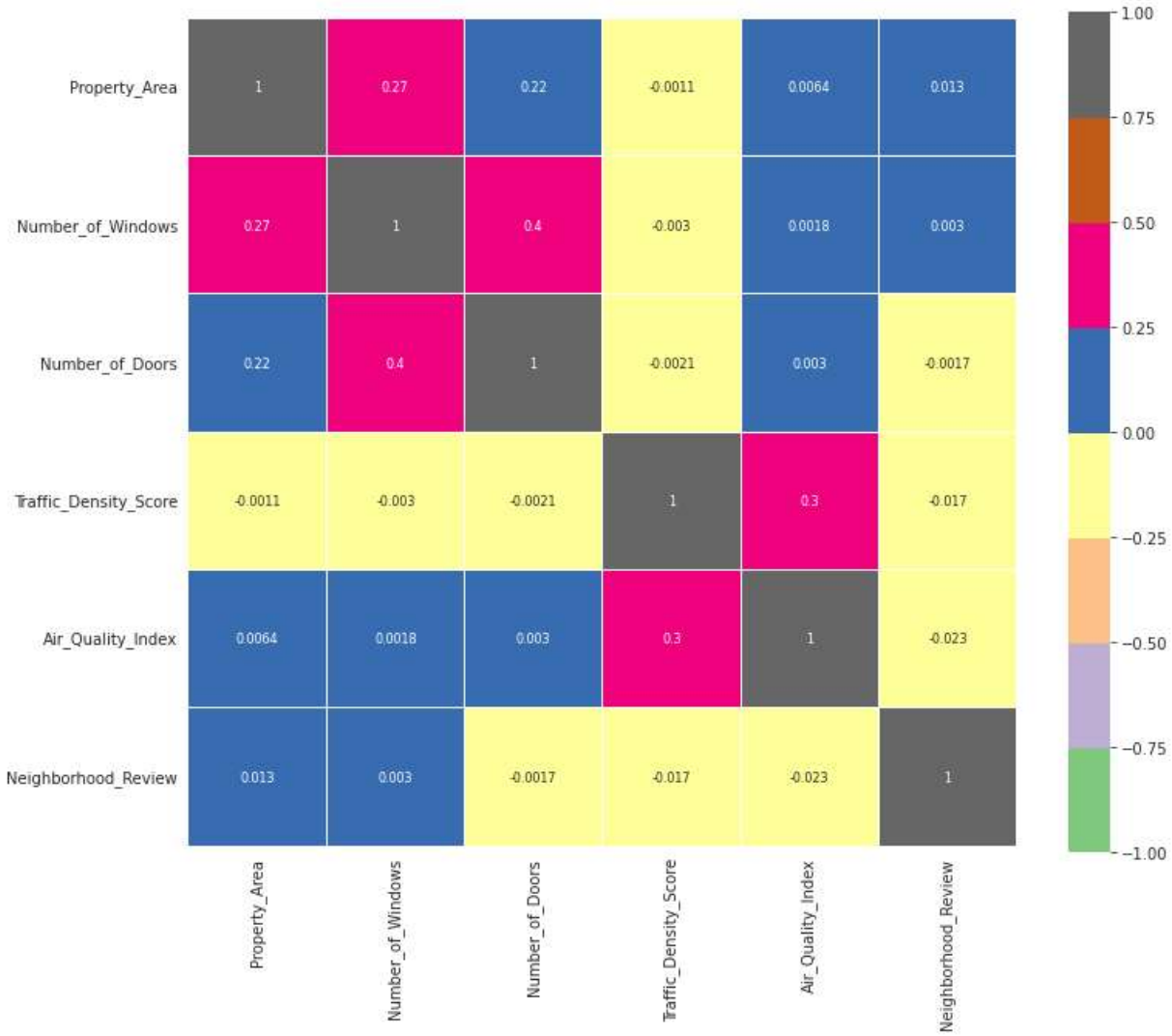
Out[ ]:

Property_ID	Property_Area	Number_of_Windows	Number_of_Doors	Traffic_Density_Score	Air_Quali
0x68d4	733	2.0	2	4.37	
0x7d81	737	4.0	2	7.45	
0x7a57	900	3.0	2	6.16	
0x9409	2238	14.0	6	5.46	
0xbe4e	1185	3.0	3	5.69	

In [ ]:

```
corr = df_num.drop('Habitability_score', axis=1).corr()
# We find the correlation of the various feature to find the relation among the othe

plt.figure(figsize=(12, 10))
# Here we will be plotting the heatmap of the correalation matrix obtaines
sns.heatmap(corr,
            cmap='Accent', vmax=1.0, vmin=-1.0, linewidths=0.1,
            annot=True, annot_kws={"size": 8}, square=True)
```

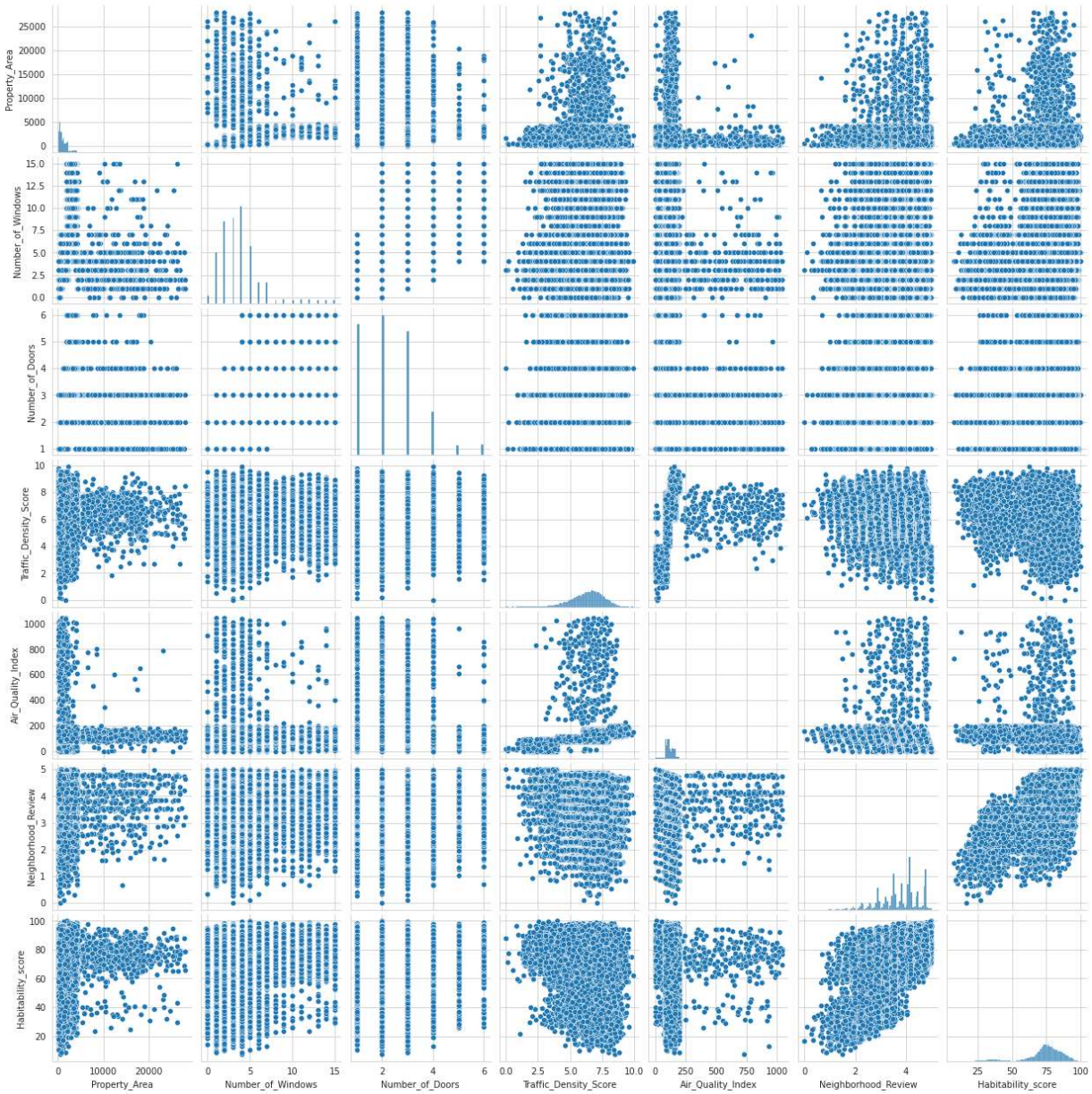
Out[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7f64a3f64410>



From above we can infer that non of the selected fratures show correlation with each other. We will confirm this inference usng scatterplot between the features.

In [ ]:
```
sns.pairplot(df_num)
```

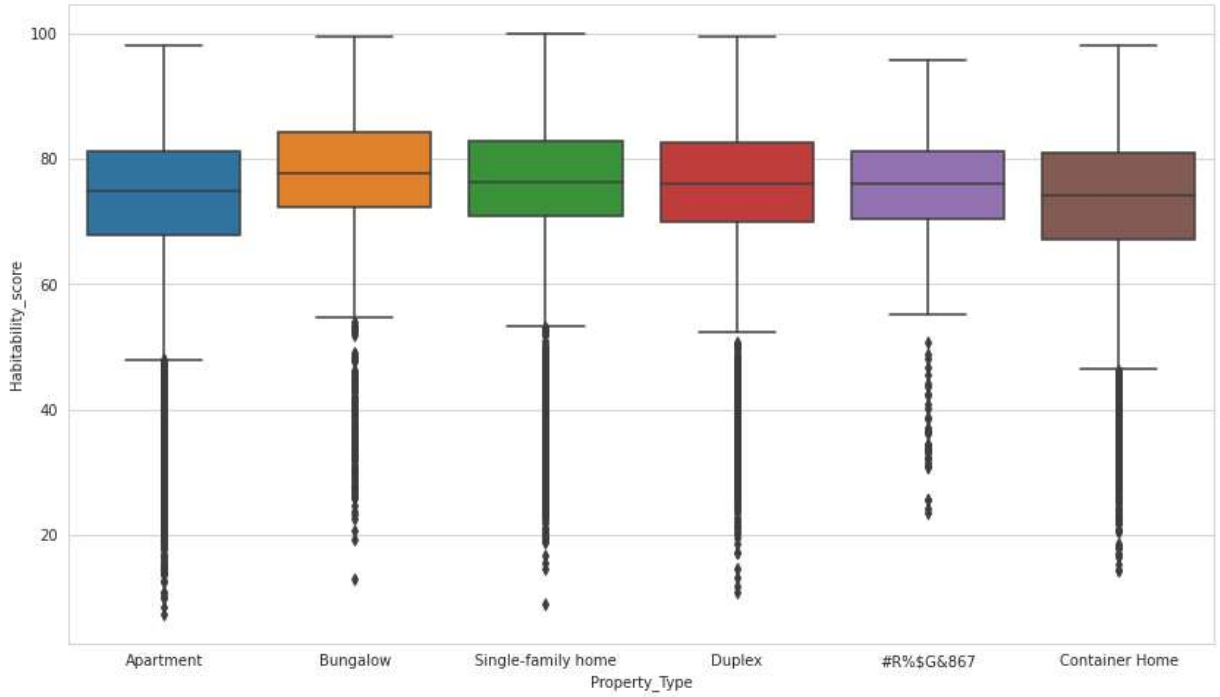
Out[ ]: <seaborn.axisgrid.PairGrid at 0x7f64a3ee5510>



Now lets find how there are outliers in the dataset for each type of the propety type vs hability score also property type and property are. This is done just to understand what are the outliers in each property type and their median. THis is achieved using boxplot

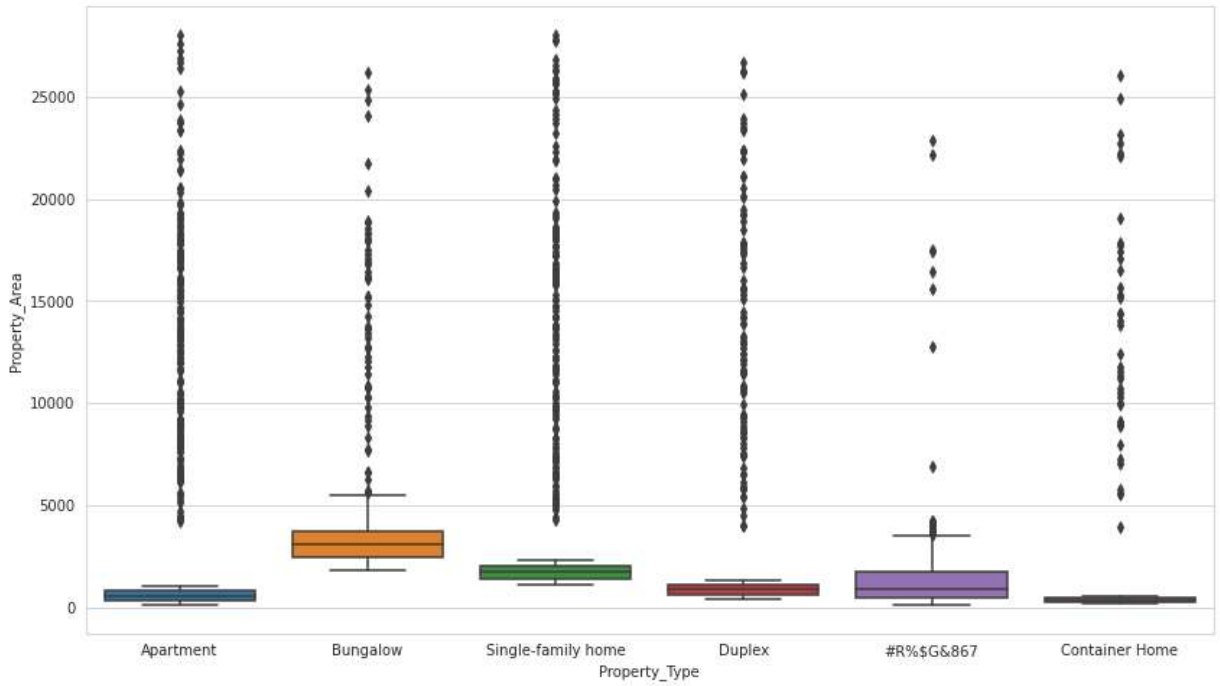
In [ ]:
```
plt.figure(figsize=(14, 8))
sns.set_style("whitegrid")
sns.boxplot(x = 'Property_Type', y = 'Habitability_score', data = data)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f64a2b4b350>
```



In [ ]:
```
plt.figure(figsize=(14, 8))
sns.set_style("whitegrid")
sns.boxplot(x = 'Property_Type', y = 'Property_Area', data = data)
```

Out[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f64a1e75490>
```



Summary of EDA:

1. We have done Exploratory data analytics on the data set which have the data attributes of the living area and then have the habitability score for each apartment.
2. We have clean the data for the null values using the pandas dropna() command
3. Then we have tried to get the no. of unique values in each column and getting the unique values for the columns which have the string as their data type.
4. We can also see the 5 point summary of the data set and can infer the max, min and the values statistical value of the columns.

5. We have tried to get the histogram plot of the habitability score and property area. By doing this we can see how much the spread of the data is and if it follows normal distribution or not. If it does not follows normal distribution then we might need to normalize it and then use in training the model
6. We can now see the relation ship of the various columns on each other where we try to plot the correlation matrix of the columns. In this we inferred that all columns have a weak correlation with each other and we might not need the dimensionality reduction.
7. Now using the boxplot we tried to find the outliers in the data set and we can plot the boxplot for each category and see the median represented by the line in the box, the 25% data points by the lower line and the 75% data point by the upper line. And the wishkers in the plot shows the IQR.

Conclusion: We successfully applied EDA procedures on the data set. I have included the snaps and the comments why we are doing the process in the code section.

In [ ]: