In [162…
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import mean_absolute_error
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn import svm
from sklearn.svm import SVC
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.linear_model import LinearRegression
```

In [138…
```python
dataset = pd.read_csv('data.csv')
```

In [139…
```python
dataset
```

Out[139]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip	country
0	2014-05-02 00:00:00	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	18810 Densmore Ave N	Shoreline	WA 98133	USA
1	2014-05-02 00:00:00	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	WA 98119	USA
2	2014-05-02 00:00:00	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	26206-26214 143rd Ave SE	Kent	WA 98042	USA
3	2014-05-02 00:00:00	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	857 170th Pl NE	Bellevue	WA 98008	USA
4	2014-05-02 00:00:00	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	9105 170th Ave NE	Redmond	WA 98052	USA
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	2014-07-09 00:00:00	3.081667e+05	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954	1979	501 N 143rd St	Seattle	WA 98133	USA
4596	2014-07-09 00:00:00	5.343333e+05	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	1983	2009	14855 SE 10th Pl	Bellevue	WA 98007	USA
4597	2014-07-09 00:00:00	4.169042e+05	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2009	0	759 Ilwaco Pl NE	Renton	WA 98059	USA
4598	2014-07-10 00:00:00	2.034000e+05	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974	0	5148 S Creston St	Seattle	WA 98178	USA
4599	2014-07-10 00:00:00	2.206000e+05	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990	0	18717 SE 258th St	Covington	WA 98042	USA

4600 rows × 18 columns

In [140…
```python
dataset.head(10)
```

Out[140]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip	country
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	2005	18810 Densmore Ave N	Shoreline	WA 98133	USA
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	0	709 W Blaine St	Seattle	WA 98119	USA
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	0	26206-26214 143rd Ave SE	Kent	WA 98042	USA
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	0	857 170th Pl NE	Bellevue	WA 98008	USA
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	1992	9105 170th Ave NE	Redmond	WA 98052	USA
5	2014-05-02 00:00:00	490000.0	2.0	1.00	880	6380	1.0	0	0	3	880	0	1938	1994	522 NE 88th St	Seattle	WA 98115	USA
6	2014-05-02 00:00:00	335000.0	2.0	2.00	1350	2560	1.0	0	0	3	1350	0	1976	0	2616 174th Ave NE	Redmond	WA 98052	USA
7	2014-05-02 00:00:00	482000.0	4.0	2.50	2710	35868	2.0	0	0	3	2710	0	1989	0	23762 SE 253rd Pl	Maple Valley	WA 98038	USA
8	2014-05-02 00:00:00	452500.0	3.0	2.50	2430	88426	1.0	0	0	4	1570	860	1985	0	46611-46625 SE 129th St	North Bend	WA 98045	USA
9	2014-05-02 00:00:00	640000.0	4.0	2.00	1520	6200	1.5	0	0	3	1520	0	1945	2010	6811 55th Ave NE	Seattle	WA 98115	USA

In [141…
```python
dataset.tail(10)
```

Out[141]:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated	street	city	statezip	country
4590	2014-07-08 00:00:00	380680.555556	4.0	2.50	2620	8331	2.0	0	0	3	2620	0	1991	0	13602 SE 186th Pl	Renton	WA 98058	USA
4591	2014-07-08 00:00:00	396166.666667	3.0	1.75	1880	5752	1.0	0	0	4	940	940	1945	0	3529 SW Webster St	Seattle	WA 98126	USA
4592	2014-07-08 00:00:00	252980.000000	4.0	2.50	2530	8169	2.0	0	0	3	2530	0	1993	0	37654 18th Pl S	Federal Way	WA 98003	USA
4593	2014-07-08 00:00:00	289373.307692	3.0	2.50	2538	4600	2.0	0	0	3	2538	0	2013	1923	5703 Charlotte Ave SE	Auburn	WA 98092	USA
4594	2014-07-09 00:00:00	210614.285714	3.0	2.50	1610	7223	2.0	0	0	3	1610	0	1994	0	26306 127th Ave SE	Kent	WA 98030	USA
4595	2014-07-09 00:00:00	308166.666667	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954	1979	501 N 143rd St	Seattle	WA 98133	USA
4596	2014-07-09 00:00:00	534333.333333	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	1983	2009	14855 SE 10th Pl	Bellevue	WA 98007	USA
4597	2014-07-09 00:00:00	416904.166667	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2009	0	759 Ilwaco Pl NE	Renton	WA 98059	USA
4598	2014-07-10 00:00:00	203400.000000	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974	0	5148 S Creston St	Seattle	WA 98178	USA
4599	2014-07-10 00:00:00	220600.000000	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990	0	18717 SE 258th St	Covington	WA 98042	USA

In [142…
```python
dataset.shape
```

Out[142]:
```
(4600, 18)
```

In [143…
```python
obj = (dataset.dtypes == 'object')
object_cols = list(obj[obj].index)
print("Categorical variables:",len(object_cols))
```

```
Categorical variables: 5
```

In [144…
```python
int_ = (dataset.dtypes == 'int')
num_cols = list(int_[int_].index)
print("Integer variables:",len(num_cols))
```

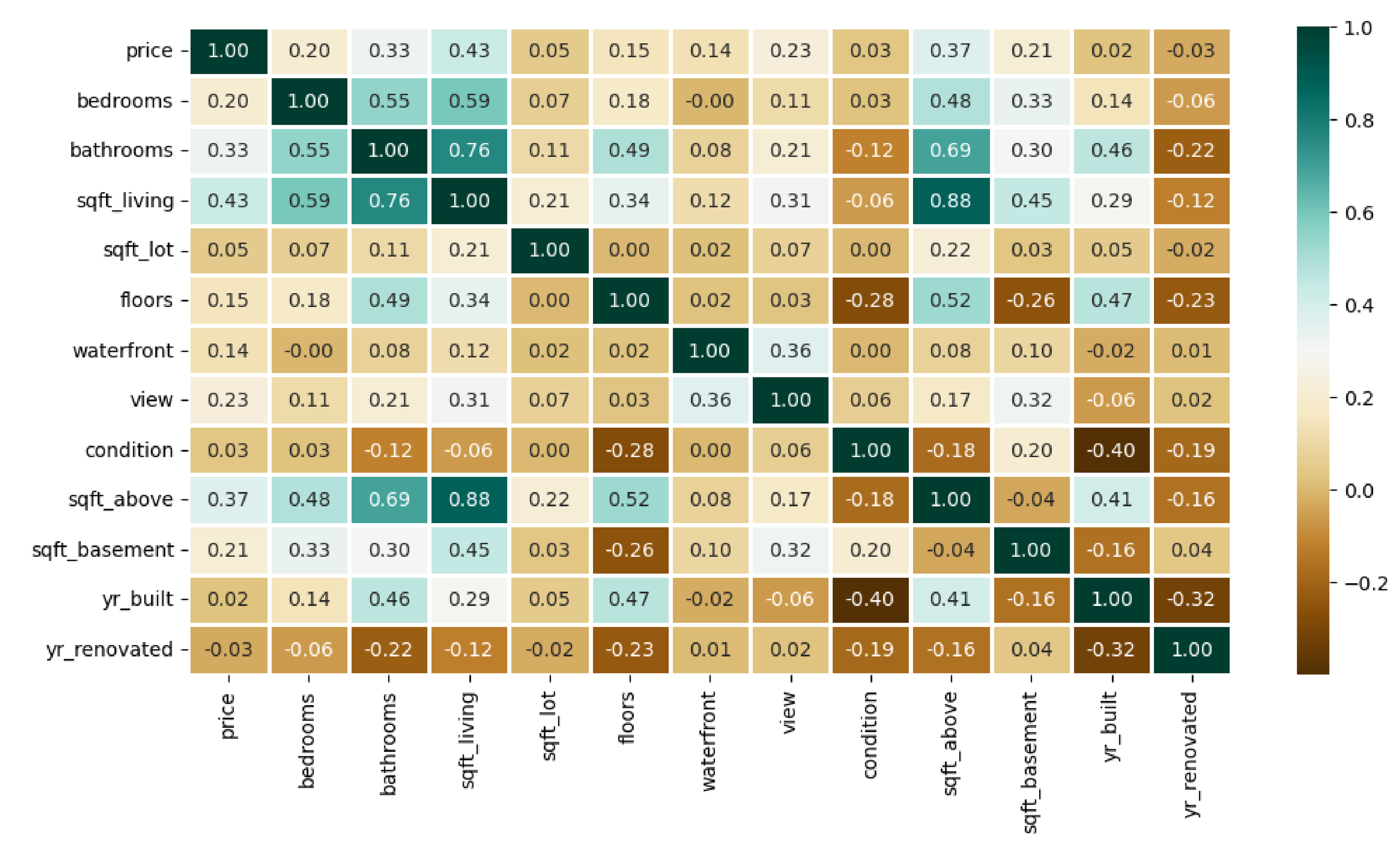
```
Integer variables: 0
```

In [145…
```python
fl = (dataset.dtypes == 'float')
fl_cols = list(fl[fl].index)
print("Float variables:",len(fl_cols))
```

```
Float variables: 4
```

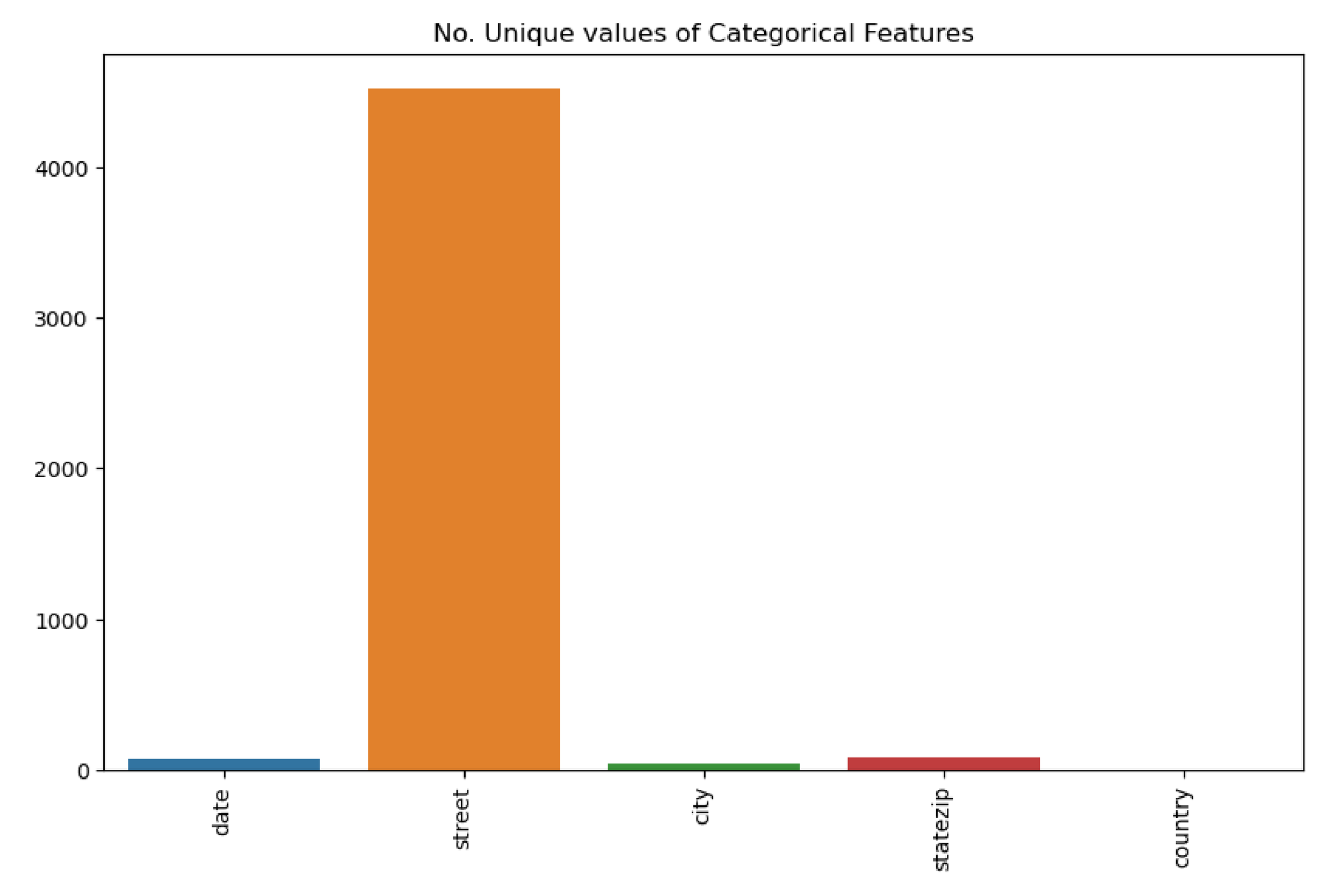
In [146…
```python
plt.figure(figsize=(12, 6))
sns.heatmap(dataset.corr(),
            cmap = 'BrBG',
            fmt = '.2f',
            linewidths = 2,
            annot = True)
```

Out[146]:
```
<AxesSubplot:>
```



In [147…
```python
unique_values = []
for col in object_cols:
  unique_values.append(dataset[col].unique().size)
plt.figure(figsize=(10,6))
plt.title('No. Unique values of Categorical Features')
plt.xticks(rotation=90)
sns.barplot(x=object_cols,y=unique_values)
```

Out[147]:
```
<AxesSubplot:title={'center':'No. Unique values of Categorical Features'}>
```



In [148…
```python
dataset.drop(['date'],
             axis=1,
             inplace=True)
```

In [149…
```python
dataset['price'] = dataset['price'].fillna(
  dataset['price'].mean())
```

In [150…
```python
new_dataset = dataset.dropna()
```

In [151…
```python
new_dataset.isnull().sum()
```

Out[151]:
```
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
street           0
city             0
statezip         0
country          0
dtype: int64
```

In [152…
```python
from sklearn.preprocessing import OneHotEncoder

s = (new_dataset.dtypes == 'object')
object_cols = list(s[s].index)
print("Categorical variables:")
print(object_cols)
print('No. of. categorical features: ',
      len(object_cols))
```

```
Categorical variables:
['street', 'city', 'statezip', 'country']
No. of. categorical features:  4
```

In [153…
```python
OH_encoder = OneHotEncoder(sparse=False)
OH_cols = pd.DataFrame(OH_encoder.fit_transform(new_dataset[object_cols]))
OH_cols.index = new_dataset.index
OH_cols.columns = OH_encoder.get_feature_names()
df_final = new_dataset.drop(object_cols, axis=1)
df_final = pd.concat([df_final, OH_cols], axis=1)
```

```
C:\Users\KONDAPAKA RISHIKA\anaconda3\lib\site-packages\sklearn\utils\deprecation.py:87: FutureWarning: Function get_feature_names is deprecated; get_feature_names is deprecated in 1.0 and will be removed in 1.2. Please use get_feature_names_out instead.
  warnings.warn(msg, category=FutureWarning)
```

In [154…
```python
X = df_final.drop(['price'], axis=1)
Y = df_final['price']
```

In [155…
```python
X
```

Out[155]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	...	x2_WA 98166	x2_WA 98168	x2_WA 98177	x2_WA 98178	x2_WA 98188	x2_WA 98198	x2_WA 98199	x2_WA 98288	x2_WA 98354	x3_USA
0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
1	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
2	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
3	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
4	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
4596	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
4597	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
4598	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	...	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
4599	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

4600 rows × 4659 columns

In [156…
```python
Y
```

Out[156]:
```
0       3.130000e+05
1       2.384000e+06
2       3.420000e+05
3       4.200000e+05
4       5.500000e+05
            ...
4595    3.081667e+05
4596    5.343333e+05
4597    4.169042e+05
4598    2.034000e+05
4599    2.206000e+05
Name: price, Length: 4600, dtype: float64
```

In [157…
```python
X_train, X_test, y_train, y_test = train_test_split(X1, y1, test_size=0.5, random_state=42)
```

In [158…
```python
Features1 = ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors']
target = 'price'
X1 = dataset[Features1]
y1 = dataset[target]
```

In [159…
```python
X1
```

Out[159]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors
0	3.0	1.50	1340	7912	1.5
1	5.0	2.50	3650	9050	2.0
2	3.0	2.00	1930	11947	1.0
3	3.0	2.25	2000	8030	1.0
4	4.0	2.50	1940	10500	1.0
...	...	...	...	...	...
4595	3.0	1.75	1510	6360	1.0
4596	3.0	2.50	1460	7573	2.0
4597	3.0	2.50	3010	7014	2.0
4598	4.0	2.00	2090	6630	1.0
4599	3.0	2.50	1490	8102	2.0

4600 rows × 5 columns

In [160…
```python
y1
```

Out[160]:
```
0       3.130000e+05
1       2.384000e+06
2       3.420000e+05
3       4.200000e+05
4       5.500000e+05
            ...
4595    3.081667e+05
4596    5.343333e+05
4597    4.169042e+05
4598    2.034000e+05
4599    2.206000e+05
Name: price, Length: 4600, dtype: float64
```

In [163…
```python
model = LinearRegression()
```

In [164…
```python
model.fit(X_train, y_train)
```

Out[164]:
```
LinearRegression()
```

In [165…
```python
y_pred = model.predict(X_test)
```

In [166…
```python
y_pred
```

Out[166]:
```
array([363715.81566702, 399446.44586549, 834230.23979668, ...,
       748123.84167884, 569569.52453247, 658083.4182335 ])
```

In [167…
```python
score = model.score(X_test, y_test)
print("Model R^2 Score:", score)
```

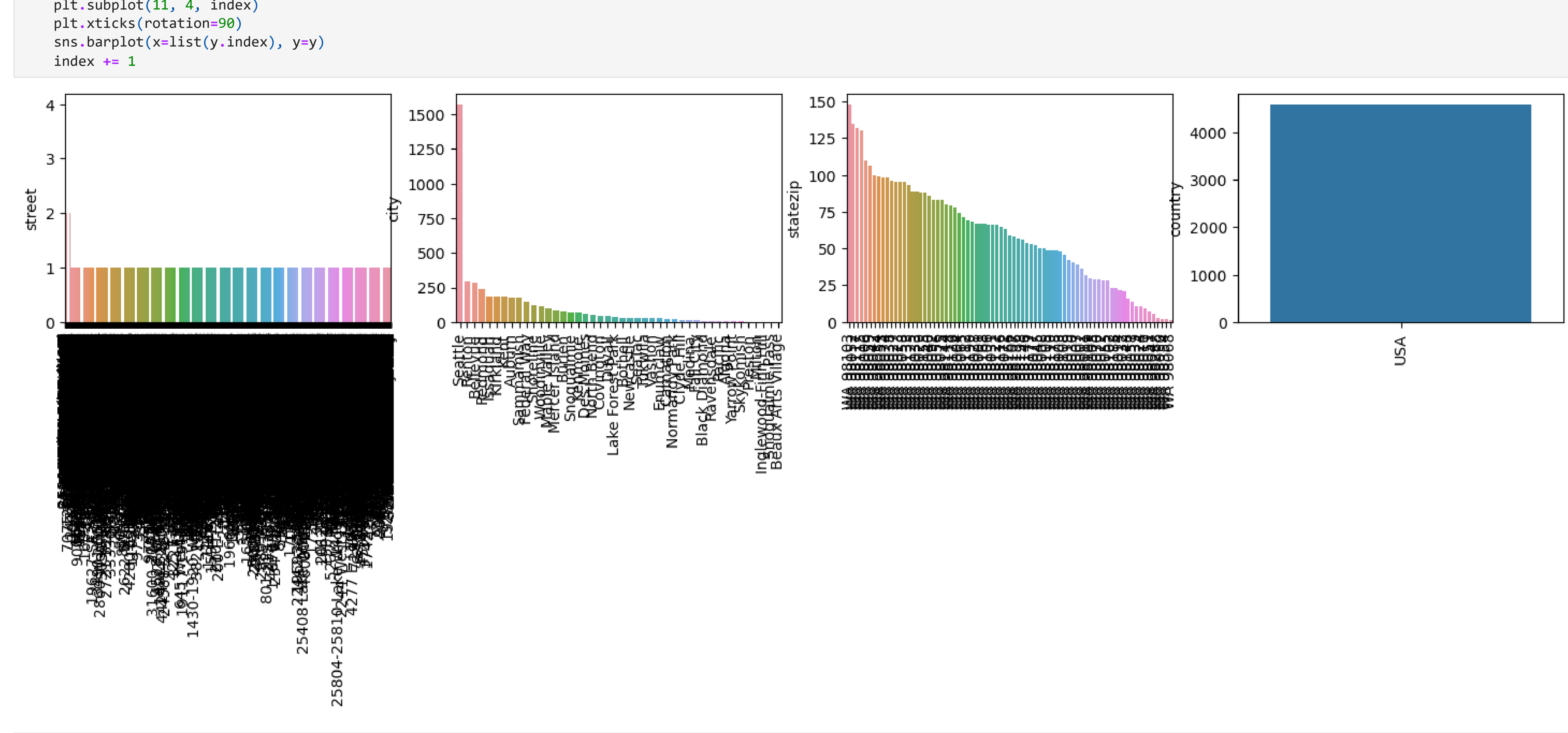
```
Model R^2 Score: 0.10933671026238057
```

In [169…
```python
new_house = pd.DataFrame({'bedrooms': [2], 'bathrooms': [2.5], 'sqft_living': [600], 'sqft_lot': [600], 'floors': [2]})
predicted_price = model.predict(new_house)
print("Predicted Price:", predicted_price[0])
```

```
Predicted Price: 146977.0030232838
```

In [170…
```python
plt.figure(figsize=(18, 36))
plt.title('Categorical Features: Distribution')
plt.xticks(rotation=90)
index = 1

for col in object_cols:
    y = dataset[col].value_counts()
    plt.subplot(11, 4, index)
    plt.xticks(rotation=90)
    sns.barplot(x=list(y.index), y=y)
    index += 1
```



In [ ]: