```
import pandas as pd
```

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

```
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	pref
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	

Next steps: Generate code with df New interactive sheet

```
df.describe()
```

	price	area	bedrooms	bathrooms	stories	parking
count	5.450000e+02	545.000000	545.000000	545.000000	545.000000	545.000000
mean	4.766729e+06	5150.541284	2.965138	1.286239	1.805505	0.693578
std	1.870440e+06	2170.141023	0.738064	0.502470	0.867492	0.861586
min	1.750000e+06	1650.000000	1.000000	1.000000	1.000000	0.000000
25%	3.430000e+06	3600.000000	2.000000	1.000000	1.000000	0.000000
50%	4.340000e+06	4600.000000	3.000000	1.000000	2.000000	0.000000
75%	5.740000e+06	6360.000000	3.000000	2.000000	2.000000	1.000000
max	1.330000e+07	16200.000000	6.000000	4.000000	4.000000	3.000000

```
df.isna().sum()
```

	0
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['furnishingstatus'].nunique()
```

```
3
```

```
columns_to_transform = ['mainroad', 'guestroom', 'basement','hotwaterheating','airconditioning','prefarea']
df[columns_to_transform] = df[columns_to_transform].replace({'yes': 1, 'no': 0})

df['furnishingstatus'] = df['furnishingstatus'].replace({'unfurnished': 0, 'semi-furnished': 1, 'furnished': 2})
```

```
/tmp/ipython-input-2182552935.py:2: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a fu
  df[columns_to_transform] = df[columns_to_transform].replace({'yes': 1, 'no': 0})
/tmp/ipython-input-2182552935.py:4: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a fu
  df['furnishingstatus'] = df['furnishingstatus'].replace({'unfurnished': 0, 'semi-furnished': 1, 'furnished': 2})
```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
sc = ['price', 'area']
df[sc] = scaler.fit_transform(df[sc])
```

```
df
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking
0	4.566365	1.046726	4	2	3	1	0	0	0	1	2
1	4.004484	1.757010	4	4	4	1	0	0	0	1	3
2	4.004484	2.218232	3	2	2	1	0	1	0	0	2
3	3.985755	1.083624	4	2	2	1	0	1	0	1	3
4	3.554979	1.046726	4	1	2	1	1	1	0	1	2
...	...	...	...	...	...	...	...	...	...	...	..
540	-1.576868	-0.991879	2	1	1	1	0	1	0	0	2
541	-1.605149	-1.268613	3	1	1	0	0	0	0	0	0
542	-1.614327	-0.705921	2	1	1	1	0	0	0	0	0
543	-1.614327	-1.033389	3	1	1	0	0	0	0	0	0
544	-1.614327	-0.599839	3	1	2	1	0	0	0	0	0

545 rows × 13 columns

Next steps: Generate code with df New interactive sheet

```
df.dtypes
```

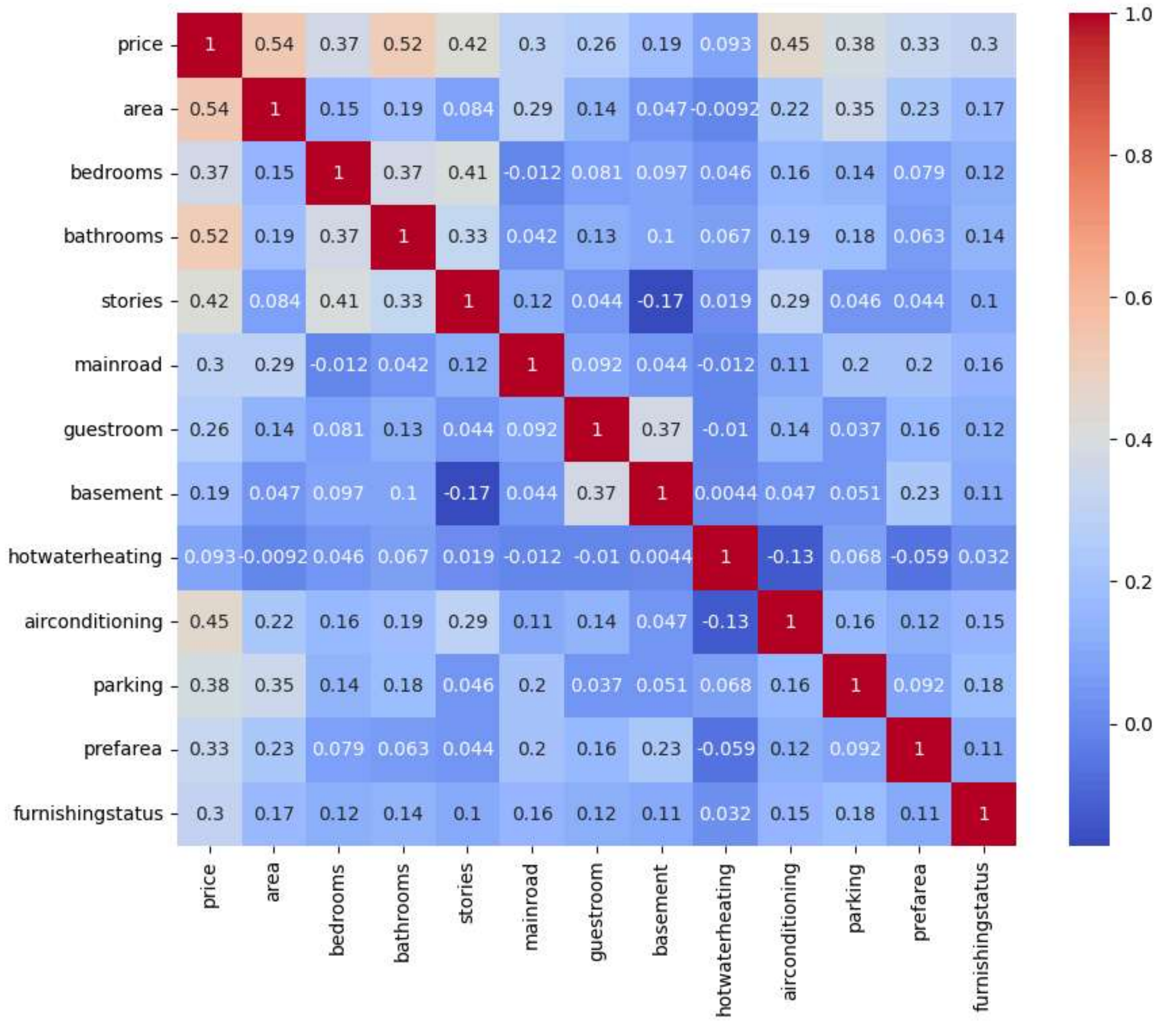
	0
price	float64
area	float64
bedrooms	int64
bathrooms	int64
stories	int64
mainroad	int64
guestroom	int64
basement	int64
hotwaterheating	int64
airconditioning	int64
parking	int64
prefarea	int64
furnishingstatus	int64

dtype: object

```
corr_matrix = df.corr()
```

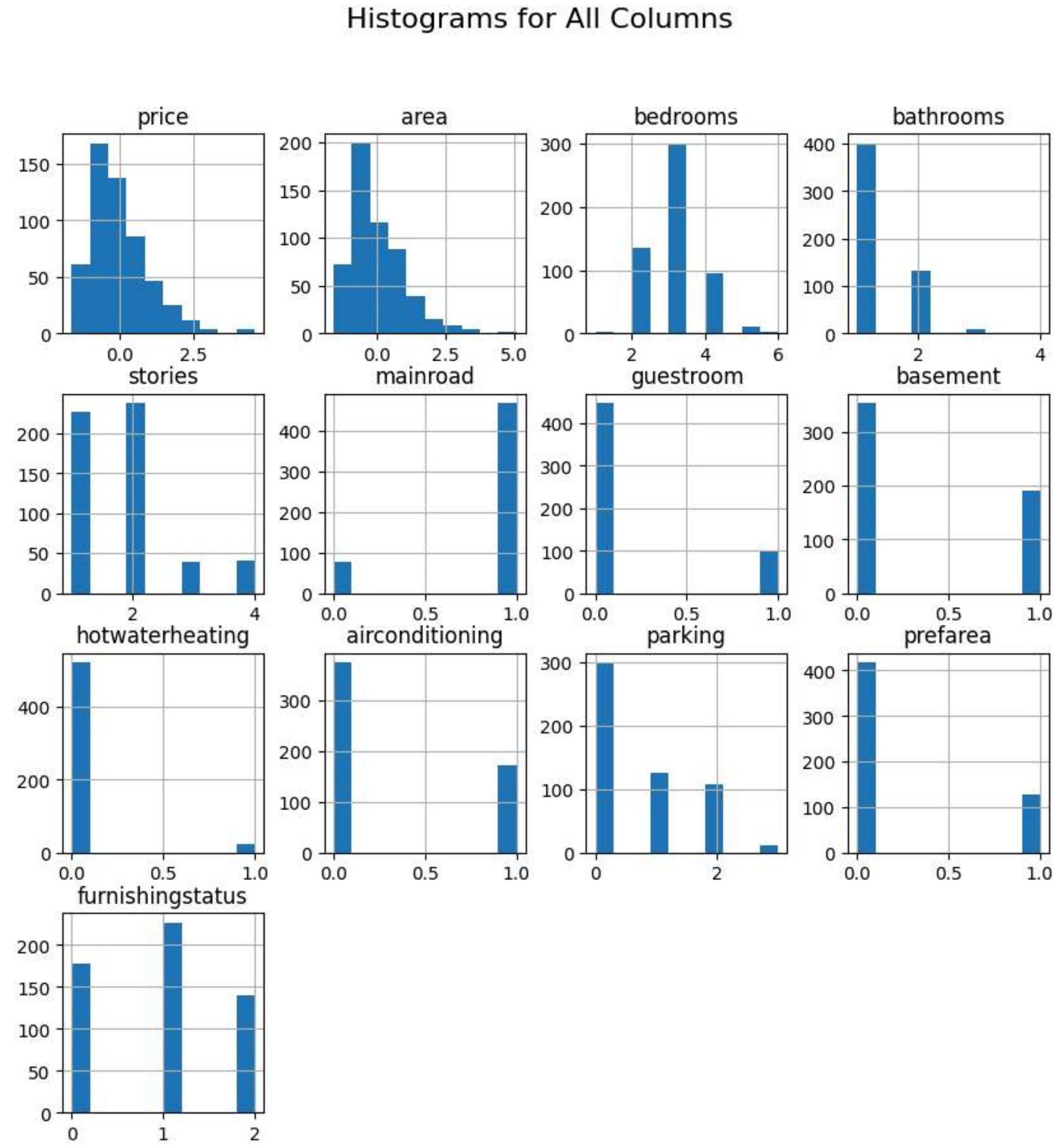
```
import matplotlib.pyplot as plt
import seaborn as sns
```

```
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm")
plt.show()
```



	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furnishingstatus
price	1	0.54	0.37	0.52	0.42	0.3	0.26	0.19	0.093	0.45	0.38	0.33	0.3
area	0.54	1	0.15	0.19	0.084	0.29	0.14	0.047	-0.0092	0.22	0.35	0.23	0.17
bedrooms	0.37	0.15	1	0.37	0.41	-0.012	0.081	0.097	0.046	0.16	0.14	0.079	0.12
bathrooms	0.52	0.19	0.37	1	0.33	0.042	0.13	0.1	0.067	0.19	0.18	0.063	0.14
stories	0.42	0.084	0.41	0.33	1	0.12	0.044	-0.17	0.019	0.29	0.046	0.044	0.1
mainroad	0.3	0.29	-0.012	0.042	0.12	1	0.092	0.044	-0.012	0.11	0.2	0.2	0.16
guestroom	0.26	0.14	0.081	0.13	0.044	0.092	1	0.37	-0.01	0.14	0.037	0.16	0.12
basement	0.19	0.047	0.097	0.1	-0.17	0.044	0.37	1	0.0044	0.047	0.051	0.23	0.11
hotwaterheating	0.093	-0.0092	0.046	0.067	0.019	-0.012	-0.01	0.0044	1	-0.13	0.068	-0.059	0.032
airconditioning	0.45	0.22	0.16	0.19	0.29	0.11	0.14	0.047	-0.13	1	0.16	0.12	0.15
parking	0.38	0.35	0.14	0.18	0.046	0.2	0.037	0.051	0.068	0.16	1	0.092	0.18
prefarea	0.33	0.23	0.079	0.063	0.044	0.2	0.16	0.23	-0.059	0.12	0.092	1	0.11
furnishingstatus	0.3	0.17	0.12	0.14	0.1	0.16	0.12	0.11	0.032	0.15	0.18	0.11	1

```
df.hist(figsize=(10, 10), bins=10)
plt.suptitle("Histograms for All Columns", fontsize=16)
plt.show()
```



```
X = df.drop('price', axis=1)
y = df['price']
```

```
X
```

	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
0	1.046726	4	2	3	1	0	0	0	1	2	1
1	1.757010	4	4	4	1	0	0	0	1	3	0
3	1.083624	4	2	2	1	0	1	0	1	3	1
4	1.046726	4	1	2	1	1	1	0	1	2	0
...	...	...	...	...	...	...	...	...	...	...	...
540	-0.991879	2	1	1	1	0	1	0	0	2	0
541	-1.268613	3	1	1	0	0	0	0	0	0	0
542	-0.705921	2	1	1	1	0	0	0	0	0	0
543	-1.033389	3	1	1	0	0	0	0	0	0	0
544	-0.599839	3	1	2	1	0	0	0	0	0	0

545 rows × 12 columns

Next steps: Generate code with X New interactive sheet

```
y
```

	price
0	4.566365
1	4.004484
2	4.004484
3	3.985755
4	3.554979
...	...
540	-1.576868
541	-1.605149
542	-1.614327
543	-1.614327
544	-1.614327

545 rows × 1 columns

dtype: float64

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.linear_model import LinearRegression

lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

y_pred = lr_model.predict(X_test)
```

```
lr_y_pred = lr_model.predict(X_test)
```

```
from sklearn.metrics import r2_score
lr_accuracy = r2_score(y_test, lr_y_pred)*100
```

```
lr_accuracy
```

```
64.94754192267794
```