

```
import pandas as pd

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

Double-click (or enter) to edit

+ Code

+ Text

```
df.head(10)
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
0	13300000	7420	4	2	3	yes	no	no	no	yes	2	yes
1	12250000	8960	4	4	4	yes	no	no	no	yes	3	no
2	12250000	9960	3	2	2	yes	no	yes	no	no	2	yes
3	12215000	7500	4	2	2	yes	no	yes	no	yes	3	yes
4	11410000	7420	4	1	2	yes	yes	yes	no	yes	2	no
5	10850000	7500	3	3	1	yes	no	yes	no	yes	2	yes
6	10150000	8580	4	3	4	yes	no	no	no	yes	2	yes
7	10150000	16200	5	3	2	yes	no	no	no	no	0	no
8	9870000	8100	4	1	2	yes	yes	yes	no	yes	2	yes
9	9800000	5750	3	2	4	yes	yes	no	no	yes	1	yes

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```
df.tail()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	no
541	1767150	2400	3	1	1	no	no	no	no	no	0	no
542	1750000	3620	2	1	1	yes	no	no	no	no	0	no
543	1750000	2910	3	1	1	no	no	no	no	no	0	no
544	1750000	3850	3	1	2	yes	no	no	no	no	0	no

```
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

Double-click (or enter) to edit

```
df.info()
```

<class 'pandas.core.frame.
RangeIndex: 545 entries, 0
Data columns (total 13 columns):
Column Non-Null Count Dtype

What can I help you build?

+

▶

```

0  price          545 non-null  int64
1  area           545 non-null  int64
2  bedrooms       545 non-null  int64
3  bathrooms      545 non-null  int64
4  stories        545 non-null  int64
5  mainroad       545 non-null  object
6  guestroom      545 non-null  object
7  basement       545 non-null  object
8  hotwaterheating 545 non-null  object
9  airconditioning 545 non-null  object
10 parking        545 non-null  int64
11 prefarea       545 non-null  object
12 furnishingstatus 545 non-null  object
dtypes: int64(6), object(7)
memory usage: 55.5+ KB

```

```
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.head()
```



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

Next steps:

[Generate code with df](#)
[View recommended plots](#)
[New interactive sheet](#)

```
list1 = ['mainroad','guestroom','basement','hotwaterheating','airconditioning','prefarea']
```

```
df[list1] = df[list1].replace({'yes':1,'no':0})
```



```
<ipython-input-12-8538a97291bb>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version
df[list1] = df[list1].replace({'yes':1,'no':0})
```

```
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
0	13300000	7420	4	2	3	1	0	0	0	1	2	1
1	12250000	8960	4	4	4	1	0	0	0	1	3	0
2	12250000	9960	3	2	2	1	0	1	0	0	2	1
3	12215000	7500	4	2	2	1	0	1	0	1	3	1
4	11410000	7420	4	1	2	1	1	1	0	1	2	0

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

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

3

```
df.tail()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
540	1820000	3000	2	1	1	1	0	1	0	0	2	0
541	1767150	2400	3	1	1	0	0	0	0	0	0	0
542	1750000	3620	2	1	1	1	0	0	0	0	0	0
543	1750000	2910	3	1	1	0	0	0	0	0	0	0
544	1750000	3850	3	1	2	1	0	0	0	0	0	0

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

```
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
0	13300000	7420	4	2	3	1	0	0	0	1	2	1
1	12250000	8960	4	4	4	1	0	0	0	1	3	0
2	12250000	9960	3	2	2	1	0	1	0	0	2	1
3	12215000	7500	4	2	2	1	0	1	0	1	3	1
4	11410000	7420	4	1	2	1	1	1	0	1	2	0

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

```
from sklearn.preprocessing import StandardScaler
```

```
scaler = StandardScaler()
```

```
list2 = ['price','area']
```

```
df[list2] = scaler.fit_transform(df[list2])
```

```
df.head()
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea
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	

Next steps: [Generate code with df](#) [View recommended plots](#) [New interactive sheet](#)

df.dtypes



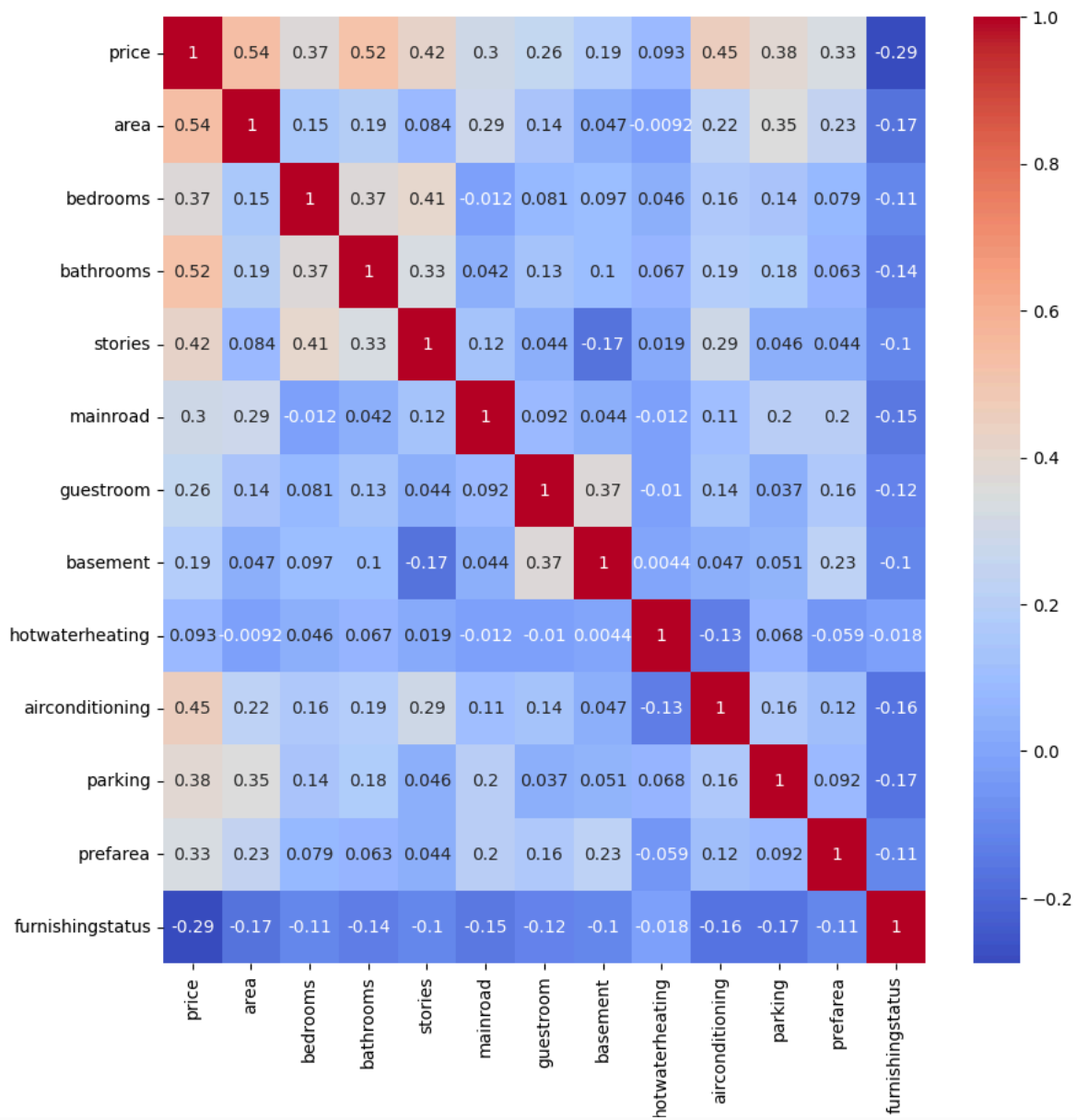
	0
price	float64
area	float64
bedrooms	int64
bathrooms	int64
stories	int64
mainroad	int64
guestroom	int64
basement	int64
hotwaterheating	int64
airconditioning	int64
parking	int64
prefarea	int64
furnishingstatus	int64

df is a object

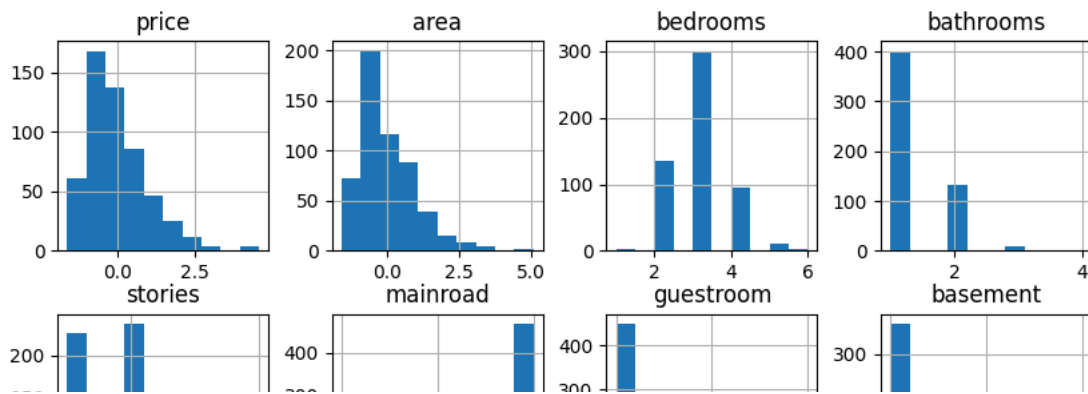
```
core = df.corr()

import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(10,10))
sns.heatmap(core,annot=True,cmap='coolwarm')
plt.show()
```

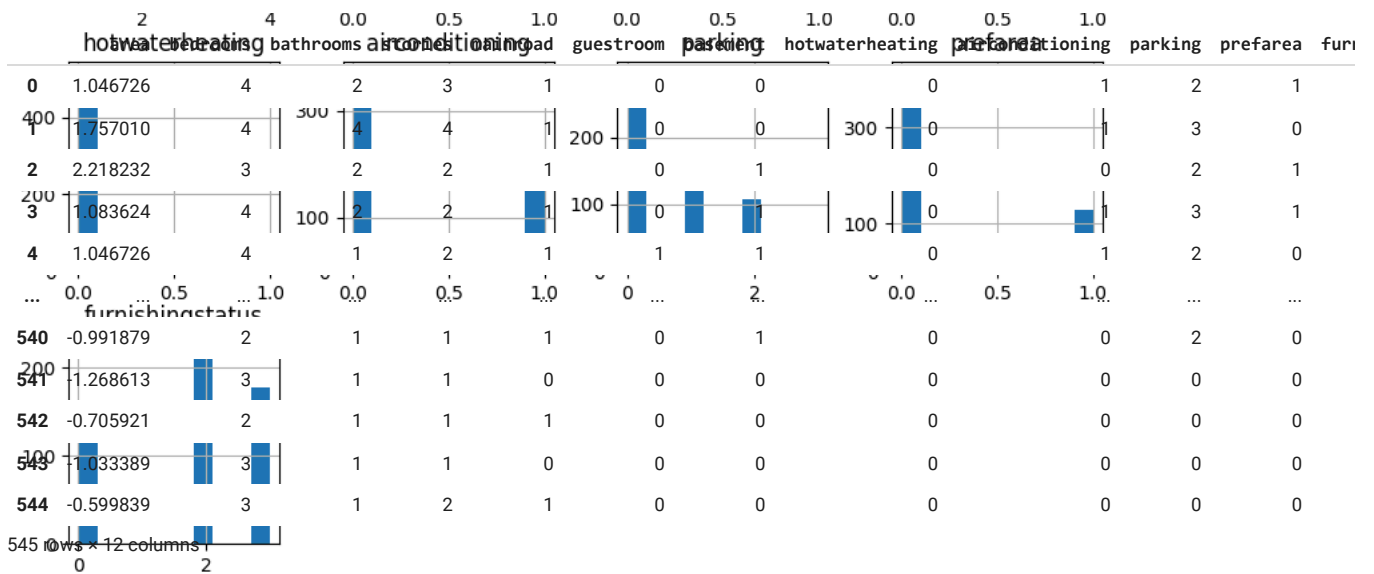


```
df.hist(figsize=(10,10),bins=10)
plt.show()
```



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

x



Next steps:

[Generate code with x](#)

[View recommended plots](#)

[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 x 1 columns

df.yes: float64