```
import pandas as pd
import numpy as np


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


df
```

	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	
...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	yes	no	yes	no	no	2	
541	1767150	2400	3	1	1	no	no	no	no	no	0	
542	1750000	3620	2	1	1	yes	no	no	no	no	0	
543	1750000	2910	3	1	1	no	no	no	no	no	0	
544	1750000	3850	3	1	2	yes	no	no	no	no	0	

545 rows × 13 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
for col in binary_cols:
    print(f"{col}: {df[col].unique()}")
```

```
mainroad: ['yes' 'no']
guestroom: ['no' 'yes']
basement: ['no' 'yes']
hotwaterheating: ['no' 'yes']
airconditioning: ['yes' 'no']
prefarea: ['yes' 'no']
```

```
binary_cols = ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'prefarea']
df[binary_cols] = df[binary_cols].apply(lambda x: x.map({'yes': 1, 'no': 0}))


df.head()
```

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

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df = pd.get_dummies(df, columns=['furnishingstatus'], drop_first=True)


df
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	pref
0	13300000	7420	4	2	3	1	0	0	0	1	2	
1	12250000	8960	4	4	4	1	0	0	0	1	3	
2	12250000	9960	3	2	2	1	0	1	0	0	2	
3	12215000	7500	4	2	2	1	0	1	0	1	3	
4	11410000	7420	4	1	2	1	1	1	0	1	2	
...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	1	0	1	0	0	2	
541	1767150	2400	3	1	1	0	0	0	0	0	0	
542	1750000	3620	2	1	1	1	0	0	0	0	0	
543	1750000	2910	3	1	1	0	0	0	0	0	0	
544	1750000	3850	3	1	2	1	0	0	0	0	0	

545 rows × 14 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df['furnishingstatus_semi-furnished'] = df['furnishingstatus_semi-furnished'].astype(int)
df['furnishingstatus_unfurnished'] = df['furnishingstatus_unfurnished'].astype(int)
```

```
df
```

	price	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	pref
0	13300000	7420	4	2	3	1	0	0	0	1	2	
1	12250000	8960	4	4	4	1	0	0	0	1	3	
2	12250000	9960	3	2	2	1	0	1	0	0	2	
3	12215000	7500	4	2	2	1	0	1	0	1	3	
4	11410000	7420	4	1	2	1	1	1	0	1	2	
...	...	...	...	...	...	...	...	...	...	...	...	...
540	1820000	3000	2	1	1	1	0	1	0	0	2	
541	1767150	2400	3	1	1	0	0	0	0	0	0	
542	1750000	3620	2	1	1	1	0	0	0	0	0	
543	1750000	2910	3	1	1	0	0	0	0	0	0	
544	1750000	3850	3	1	2	1	0	0	0	0	0	

545 rows × 14 columns

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.isnull().any()
```

	0
price	False
area	False
bedrooms	False
bathrooms	False
stories	False
mainroad	False
guestroom	False
basement	False
hotwaterheating	False
airconditioning	False
parking	False
prefarea	False
furnishingstatus_semi-furnished	False
furnishingstatus_unfurnished	False

dtype: bool

```
from sklearn.model_selection import train_test_split


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


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


print("Training features shape:", X_train.shape)
print("Testing features shape:", X_test.shape)
print("Training labels shape:", y_train.shape)
print("Testing labels shape:", y_test.shape)
```

```
Training features shape: (436, 13)
Testing features shape: (109, 13)
Training labels shape: (436,)
Testing labels shape: (109,)
```

```
X_train.head()
```

	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furn
46	6000	3	2	4	1	0	0	0	1	1	0	
93	7200	3	2	1	1	0	1	0	1	3	0	
335	3816	2	1	1	1	0	1	0	1	2	0	
412	2610	3	1	2	1	0	1	0	0	0	1	
471	3750	3	1	2	1	0	0	0	0	0	0	

Next steps: Generate code with `X_train` View recommended plots New interactive sheet

```
X_test.head()
```

	area	bedrooms	bathrooms	stories	mainroad	guestroom	basement	hotwaterheating	airconditioning	parking	prefarea	furn
316	5900	4	2	2	0	0	1	0	0	1	0	
77	6500	3	2	3	1	0	0	0	1	0	1	
360	4040	2	1	1	1	0	0	0	0	0	0	
90	5000	3	1	2	1	0	0	0	1	0	0	
493	3960	3	1	1	1	0	0	0	0	0	0	

Next steps: Generate code with X_test | View recommended plots | New interactive sheet

```
y_train.head()
```

	price
46	7525000
93	6300000
335	3920000
412	3430000
471	3010000

dtype: int64

```
from sklearn.linear_model import LinearRegression


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

```
LinearRegression
LinearRegression()
```

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score


y_pred = lr_model.predict(X_test)


mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MAE: {mae:.2f}")
print(f"MSE: {mse:.2f}")
print(f"R²: {r2:.2f}")
```

```
MAE: 970043.40
MSE: 1754318687330.66
R²: 0.65
```

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


plt.figure(figsize=(8,6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y.min(), y.max()], [y.min(), y.max()], color='red', linestyle='--')
plt.xlabel("Actual Prices")
plt.ylabel("Predicted Prices")
plt.title("Actual vs Predicted House Prices")
plt.show()

# Coefficient interpretation
coef_df = pd.DataFrame({
    'Feature': X.columns,
    'Coefficient': lr_model.coef_
}).sort_values(by='Coefficient', ascending=False)

print(coef_df)
```

```
print(coef_df)
```