

<https://www.kaggle.com/datasets/hosammhmdali/house-price-dataset?resource>

```
from google.colab import drive
drive.mount('/content/drive')
```

Mounted at /content/drive

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
```

```
df = pd.read_csv('/content/drive/MyDrive/project1/housing.csv')
```

```
df.head()
```

	longitude	latitude	housing_median_age	total_rooms	total_bedrooms	population	households	median_income	median_house_value
0	-122.23	37.88	41.0	880.0	129.0	322.0	126.0	8.3252	452600
1	-122.22	37.86	21.0	7099.0	1106.0	2401.0	1138.0	8.3014	358500
2	-122.24	37.85	52.0	1467.0	190.0	496.0	177.0	7.2574	352100
3	-122.25	37.85	52.0	1274.0	235.0	558.0	219.0	5.6431	341300
4	-122.25	37.85	52.0	1627.0	280.0	565.0	259.0	3.8462	342200

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

```
df.dtypes
```

	0
longitude	float64
latitude	float64
housing_median_age	float64
total_rooms	float64
total_bedrooms	float64
population	float64
households	float64
median_income	float64
median_house_value	float64
ocean_proximity	object
dtype:	object

```
df.columns
```

```
Index(['longitude', 'latitude', 'housing_median_age', 'total_rooms',
      'total_bedrooms', 'population', 'households', 'median_income',
      'median_house_value', 'ocean_proximity'],
      dtype='object')
```

```
df.isnull().sum()
```

	0
longitude	0
latitude	0
housing_median_age	0
total_rooms	0
total_bedrooms	207
population	0
households	0
median_income	0
median_house_value	0
ocean_proximity	0

dtype: int64

```
df['total_bedrooms'].fillna(df['total_bedrooms'].median(), inplace=True)
```

The dataset contained missing values only in the total_bedrooms column (207 values). Since the number of missing values is relatively small, they were handled using median imputation instead of dropping the column

```
print(df.isnull().sum())
```

```
longitude      0
latitude       0
housing_median_age  0
total_rooms    0
total_bedrooms 0
population     0
households     0
median_income  0
median_house_value  0
ocean_proximity 0
dtype: int64
```

```
import numpy as np
```

```
numeric_df = df.select_dtypes(include=np.number)
```

```
Q1 = numeric_df.quantile(0.25)
```

```
Q3 = numeric_df.quantile(0.75)
```

```
IQR = Q3 - Q1
```

```
df_out = df[~((numeric_df < (Q1 - 1.5 * IQR)) | (numeric_df > (Q3 + 1.5 * IQR))).any(axis=1)]
```

```
print("Shape before:", df.shape)
```

```
print("Shape after:", df_out.shape)
```

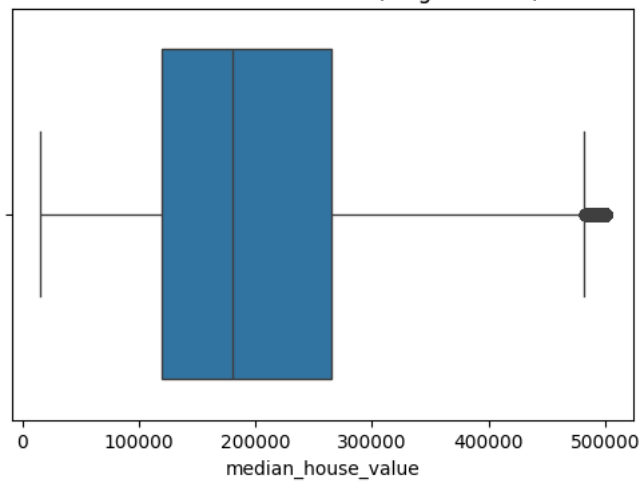
```
Shape before: (20640, 10)
```

```
Shape after: (17609, 10)
```

Rows containing outliers in numeric columns such as median_house_value and total_rooms were removed. The dataset size decreased from 20640 to 17609 rows after removing 3031 rows.

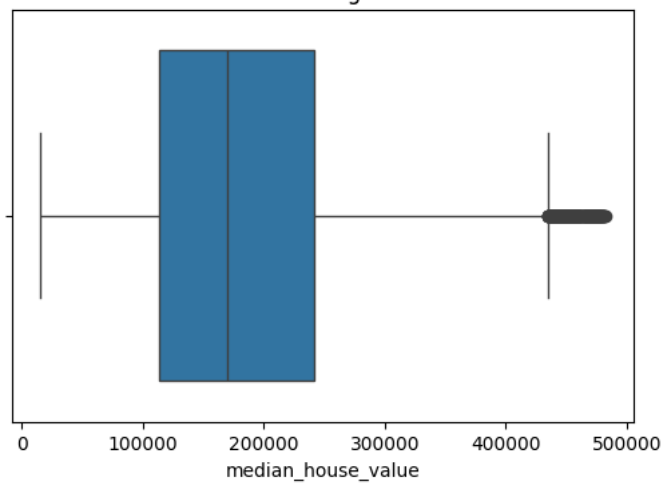
```
plt.figure(figsize=(6,4))
sns.boxplot(x=df['median_house_value'])
plt.title("Outliers in House Prices (Original Data)")
plt.show()
```

Outliers in House Prices (Original Data)



```
plt.figure(figsize=(6,4))
sns.boxplot(x=df_out['median_house_value'])
plt.title("After Removing Outliers")
plt.show()
```

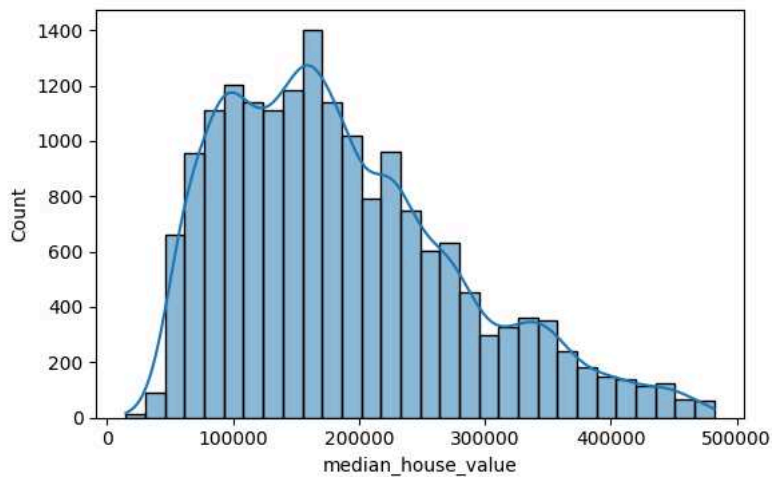
After Removing Outliers



The boxplot above shows the distribution of the median_house_value before and after removing outliers. After removing the outliers, the data distribution becomes more balanced and extreme values are reduced.

```
plt.figure(figsize=(6,4))
sns.histplot(df_out['median_house_value'], bins=30, kde=True)
plt.title("Distribution of House Prices")
plt.tight_layout()
plt.show()
```

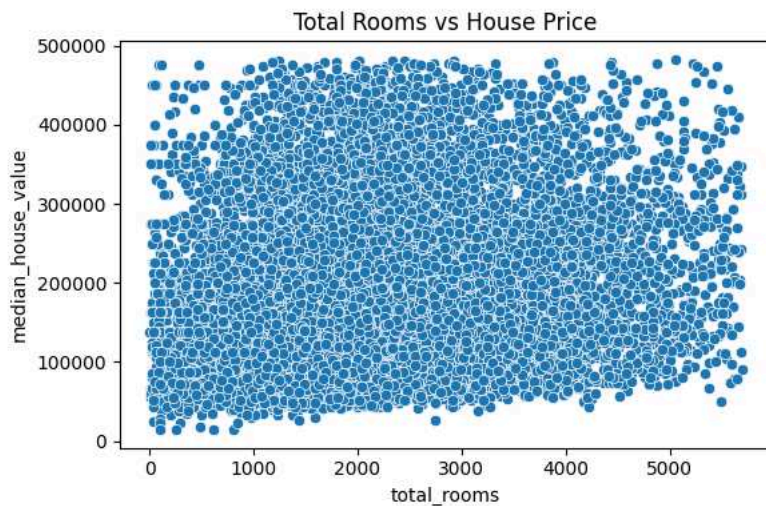
Distribution of House Prices



Shows the distribution of house prices after removing outliers. Most houses are priced in the mid_range, with fewer houses at the higher end.

```
plt.figure(figsize=(6,4))
sns.scatterplot(x=df_out['total_rooms'], y=df_out['median_house_value'])
```

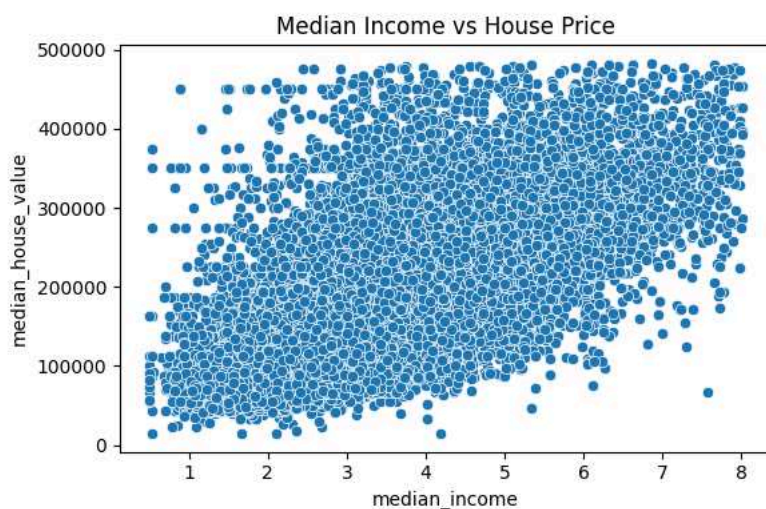
```
plt.title("Total Rooms vs House Price")
plt.tight_layout()
plt.show()
```



Shows how house size number of rooms affects price. Houses with more rooms generally have higher prices, but there is some variation due to family size and location.

Double-click (or enter) to edit

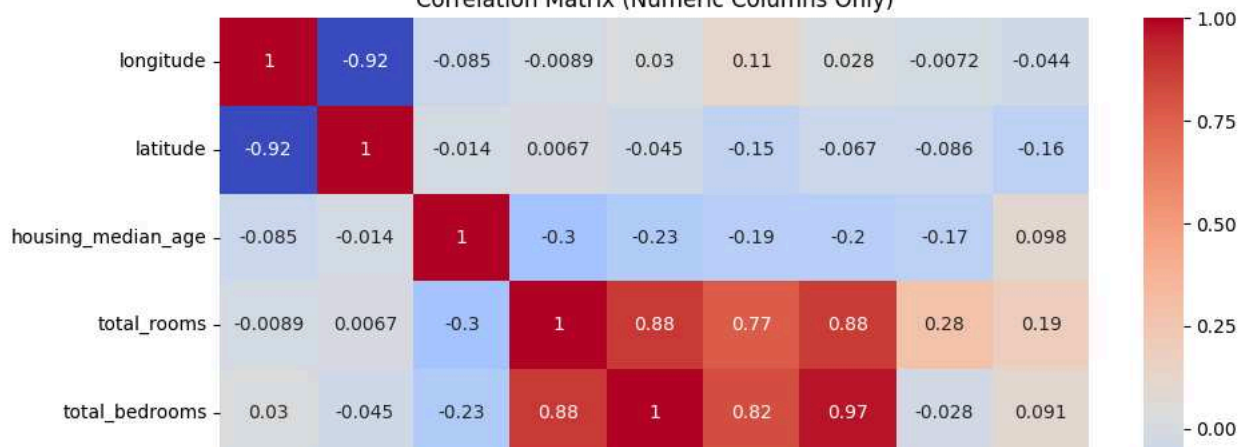
```
plt.figure(figsize=(6,4))
sns.scatterplot(x=df_out['median_income'], y=df_out['median_house_value'])
plt.title("Median Income vs House Price")
plt.tight_layout()
plt.show()
```

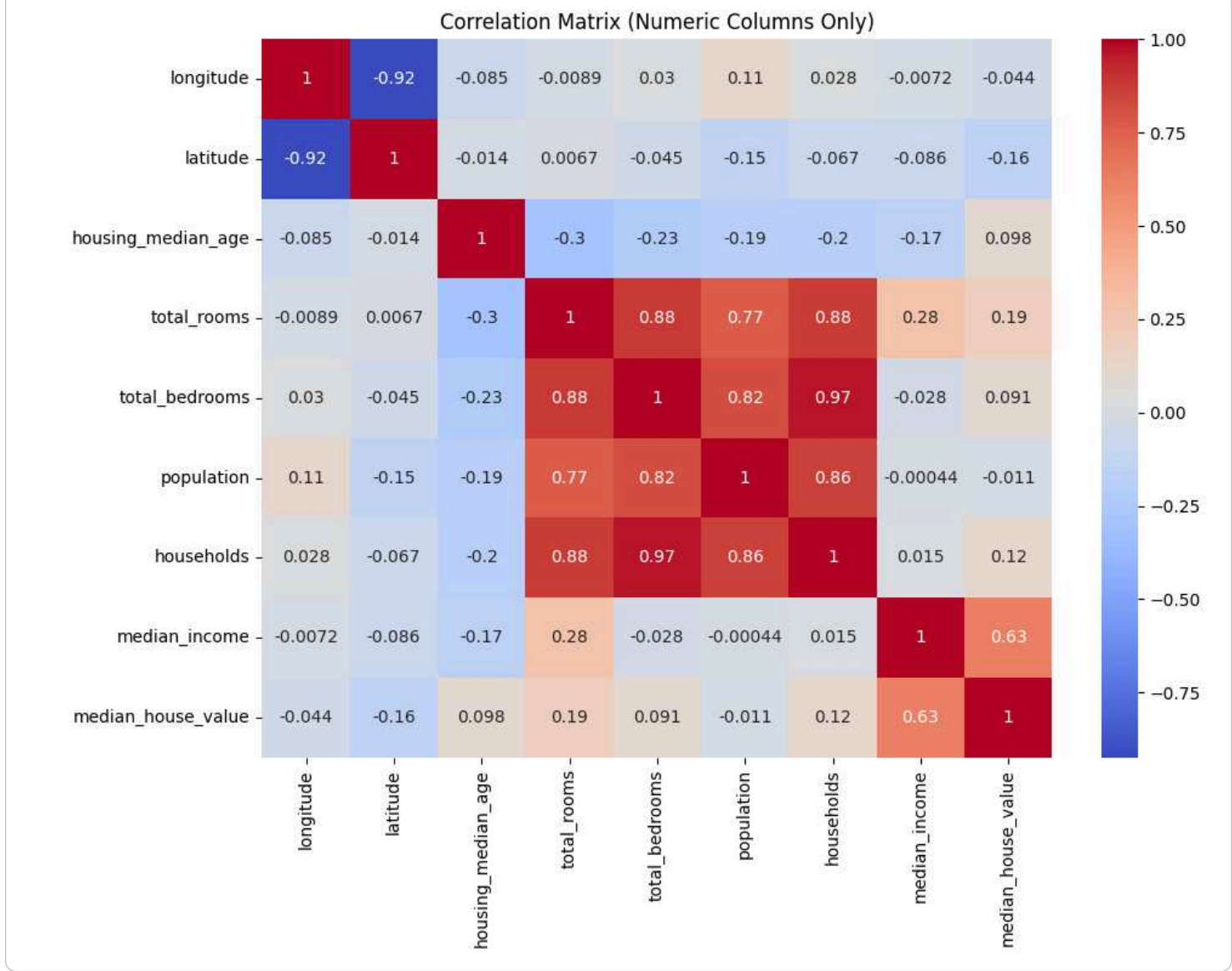


There is a positive correlation between median_income and median_house_value. Higher income areas generally have more expensive houses.

```
numeric_df = df_out.select_dtypes(include=np.number)
plt.figure(figsize=(10,8))
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix (Numeric Columns Only)")
plt.tight_layout()
plt.show()
```

Correlation Matrix (Numeric Columns Only)





Displays correlations between numeric variables only. median_income and total_rooms show the strongest positive correlation with median_house_value. Non-numeric columns like ocean_proximity are excluded from the heatmap.