

# Project\_1

March 9, 2025

Housing prices in King County vary widely, and buyers, sellers, and real estate investors want to understand the key factors driving these prices. This project aims to analyze historical sales data to uncover patterns and build predictive models to estimate house prices. In the process, SQL will be used to extract useful data used for visualization in Tableau, and further data exploration on the correlation of prices of different variables such as the number of bedrooms, bathrooms, or the square footage of living space in the house, will be done in R and Python.

```
[2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn import linear_model
```

```
[3]: housing_data = pd.read_csv(r"C:\Users\15157\Desktop\Work\project1\kc_house_data.
↪CSV")
```

```
[4]: housing_data
```

```
[4]:
```

	id	date	price	bedrooms	bathrooms	\
0	7129300520	20141013T000000	221900.0	3	1.00	
1	6414100192	20141209T000000	538000.0	3	2.25	
2	5631500400	20150225T000000	180000.0	2	1.00	
3	2487200875	20141209T000000	604000.0	4	3.00	
4	1954400510	20150218T000000	510000.0	3	2.00	
...	...	...	...	...	...	
21608	263000018	20140521T000000	360000.0	3	2.50	
21609	6600060120	20150223T000000	400000.0	4	2.50	
21610	1523300141	20140623T000000	402101.0	2	0.75	
21611	291310100	20150116T000000	400000.0	3	2.50	
21612	1523300157	20141015T000000	325000.0	2	0.75	

	sqft_living	sqft_lot	floors	waterfront	view	...	grade	\
0	1180	5650	1.0	0	0	...	7	
1	2570	7242	2.0	0	0	...	7	
2	770	10000	1.0	0	0	...	6	
3	1960	5000	1.0	0	0	...	7	
4	1680	8080	1.0	0	0	...	8	

...	...	...	...	...	...	...	...	...
21608	1530	1131	3.0		0	0	...	8
21609	2310	5813	2.0		0	0	...	8
21610	1020	1350	2.0		0	0	...	7
21611	1600	2388	2.0		0	0	...	8
21612	1020	1076	2.0		0	0	...	7

	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	\
0	1180	0	1955	0	98178	47.5112	
1	2170	400	1951	1991	98125	47.7210	
2	770	0	1933	0	98028	47.7379	
3	1050	910	1965	0	98136	47.5208	
4	1680	0	1987	0	98074	47.6168	

...	...	...	...	...	...	...	...
21608	1530	0	2009	0	98103	47.6993	
21609	2310	0	2014	0	98146	47.5107	
21610	1020	0	2009	0	98144	47.5944	
21611	1600	0	2004	0	98027	47.5345	
21612	1020	0	2008	0	98144	47.5941	

	long	sqft_living15	sqft_lot15
0	-122.257	1340	5650
1	-122.319	1690	7639
2	-122.233	2720	8062
3	-122.393	1360	5000
4	-122.045	1800	7503

...	...	...	...
21608	-122.346	1530	1509
21609	-122.362	1830	7200
21610	-122.299	1020	2007
21611	-122.069	1410	1287
21612	-122.299	1020	1357

[21613 rows x 21 columns]

The data originates from housing sales dataset from Kaggle(<https://www.kaggle.com/datasets/harlfoxem/housesalesprediction/data>), which has the sales statistics of all the transactions between the year 2014 and 2015, along with detail information on each of the houses. The dataset contains information on 21,613 house sales, including attributes such as price, number of bedrooms, bathrooms, living area, and condition. Initially, I identified and removed duplicate listings, as well as checked for missing values, ensuring clean and reliable data for analysis.

```
[6]: #clean up

missing_values = housing_data.isnull().sum()
print(missing_values)
```

```
housing_data = housing_data.drop_duplicates()
```

```
id          0
date        0
price       0
bedrooms    0
bathrooms   0
sqft_living  0
sqft_lot    0
floors       0
waterfront  0
view         0
condition   0
grade        0
sqft_above  0
sqft_basement 0
yr_built     0
yr_renovated 0
zipcode      0
lat          0
long         0
sqft_living15 0
sqft_lot15   0
dtype: int64
```

```
[7]: housing_data.info()
housing_data.head()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   id              21613 non-null  int64
1   date            21613 non-null  object
2   price           21613 non-null  float64
3   bedrooms        21613 non-null  int64
4   bathrooms       21613 non-null  float64
5   sqft_living     21613 non-null  int64
6   sqft_lot        21613 non-null  int64
7   floors          21613 non-null  float64
8   waterfront      21613 non-null  int64
9   view            21613 non-null  int64
10  condition       21613 non-null  int64
11  grade           21613 non-null  int64
12  sqft_above      21613 non-null  int64
13  sqft_basement   21613 non-null  int64
14  yr_built        21613 non-null  int64
```

```

15  yr_renovated    21613 non-null  int64
16  zipcode         21613 non-null  int64
17  lat             21613 non-null  float64
18  long            21613 non-null  float64
19  sqft_living15   21613 non-null  int64
20  sqft_lot15      21613 non-null  int64
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB

```

```

[7]:
      id      date      price  bedrooms  bathrooms  sqft_living  \
0  7129300520  20141013T000000  221900.0         3         1.00         1180
1  6414100192  20141209T000000  538000.0         3         2.25         2570
2  5631500400  20150225T000000  180000.0         2         1.00         770
3  2487200875  20141209T000000  604000.0         4         3.00         1960
4  1954400510  20150218T000000  510000.0         3         2.00         1680

```

```

      sqft_lot  floors  waterfront  view  ...  grade  sqft_above  sqft_basement  \
0         5650     1.0           0     0  ...     7         1180           0
1         7242     2.0           0     0  ...     7         2170          400
2        10000     1.0           0     0  ...     6          770           0
3         5000     1.0           0     0  ...     7         1050          910
4         8080     1.0           0     0  ...     8         1680           0

```

```

      yr_built  yr_renovated  zipcode      lat      long  sqft_living15  \
0         1955           0     98178  47.5112 -122.257         1340
1         1951          1991     98125  47.7210 -122.319         1690
2         1933           0     98028  47.7379 -122.233         2720
3         1965           0     98136  47.5208 -122.393         1360
4         1987           0     98074  47.6168 -122.045         1800

```

```

      sqft_lot15
0         5650
1         7639
2         8062
3         5000
4         7503

```

[5 rows x 21 columns]

```

[8]: #summary statistics
housing_data.describe()

```

```

[8]:
      count      id      price      bedrooms      bathrooms  sqft_living  \
count  2.161300e+04  2.161300e+04  21613.000000  21613.000000  21613.000000
mean    4.580302e+09  5.400881e+05    3.370842    2.114757  2079.899736
std     2.876566e+09  3.671272e+05    0.930062    0.770163   918.440897
min     1.000102e+06  7.500000e+04    0.000000    0.000000   290.000000
25%     2.123049e+09  3.219500e+05    3.000000    1.750000  1427.000000

```

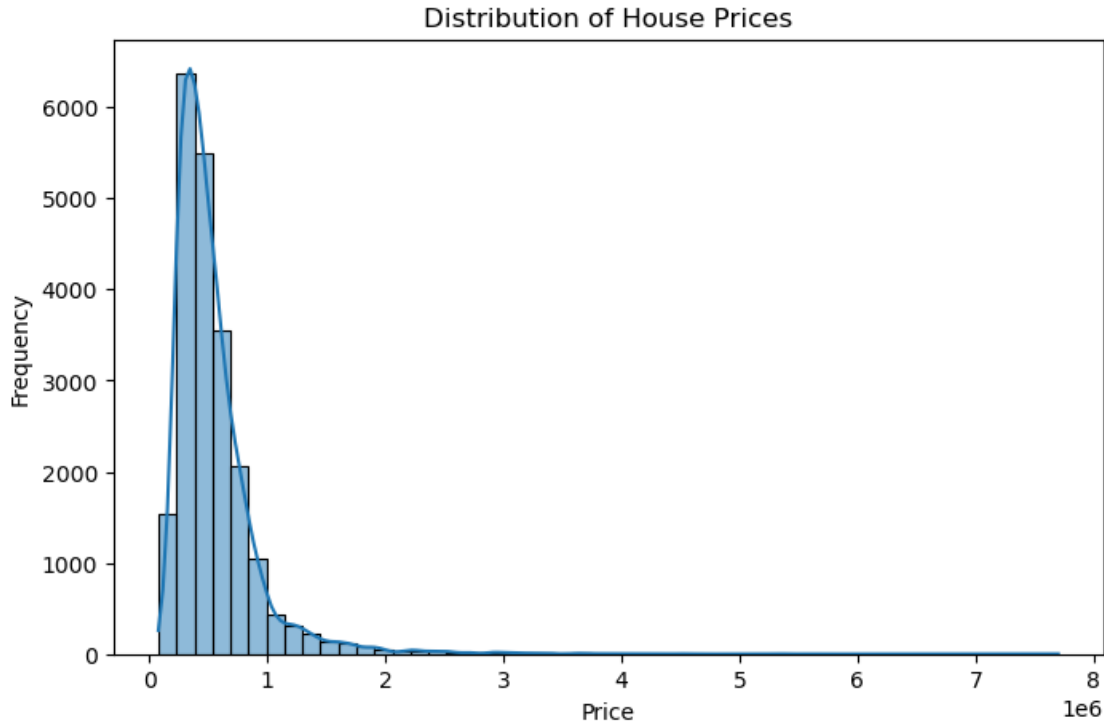
50%	3.904930e+09	4.500000e+05	3.000000	2.250000	1910.000000
75%	7.308900e+09	6.450000e+05	4.000000	2.500000	2550.000000
max	9.900000e+09	7.700000e+06	33.000000	8.000000	13540.000000

	sqft_lot	floors	waterfront	view	condition \
count	2.161300e+04	21613.000000	21613.000000	21613.000000	21613.000000
mean	1.510697e+04	1.494309	0.007542	0.234303	3.409430
std	4.142051e+04	0.539989	0.086517	0.766318	0.650743
min	5.200000e+02	1.000000	0.000000	0.000000	1.000000
25%	5.040000e+03	1.000000	0.000000	0.000000	3.000000
50%	7.618000e+03	1.500000	0.000000	0.000000	3.000000
75%	1.068800e+04	2.000000	0.000000	0.000000	4.000000
max	1.651359e+06	3.500000	1.000000	4.000000	5.000000

	grade	sqft_above	sqft_basement	yr_built	yr_renovated \
count	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000
mean	7.656873	1788.390691	291.509045	1971.005136	84.402258
std	1.175459	828.090978	442.575043	29.373411	401.679240
min	1.000000	290.000000	0.000000	1900.000000	0.000000
25%	7.000000	1190.000000	0.000000	1951.000000	0.000000
50%	7.000000	1560.000000	0.000000	1975.000000	0.000000
75%	8.000000	2210.000000	560.000000	1997.000000	0.000000
max	13.000000	9410.000000	4820.000000	2015.000000	2015.000000

	zipcode	lat	long	sqft_living15	sqft_lot15
count	21613.000000	21613.000000	21613.000000	21613.000000	21613.000000
mean	98077.939805	47.560053	-122.213896	1986.552492	12768.455652
std	53.505026	0.138564	0.140828	685.391304	27304.179631
min	98001.000000	47.155900	-122.519000	399.000000	651.000000
25%	98033.000000	47.471000	-122.328000	1490.000000	5100.000000
50%	98065.000000	47.571800	-122.230000	1840.000000	7620.000000
75%	98118.000000	47.678000	-122.125000	2360.000000	10083.000000
max	98199.000000	47.777600	-121.315000	6210.000000	871200.000000

```
[9]: #Histogram
plt.figure(figsize=(8,5))
sns.histplot(housing_data['price'], bins=50, kde=True)
plt.title('Distribution of House Prices')
plt.xlabel('Price')
plt.ylabel('Frequency')
plt.show()
```



```
[10]: percentage_under_1M = (housing_data[housing_data['price'] <= 1_000_000].
      ↪shape[0] / len(housing_data)) * 100
      print(f"Percentage of houses priced under $1M: {percentage_under_1M:.2f}%")
```

Percentage of houses priced under \$1M: 93.22%

The King's County housing market is dominated by homes under \$1M, making up 93.22% of transactions.

Through exploratory analysis, I found that 93.22% of houses sold for under \$1M, but the most expensive sales reached as high as \$7.7M. The top five zip codes with the highest house prices had a strong correlation with waterfront properties and higher property grades. Additionally, homes with more bedrooms and larger living spaces tend to have higher selling prices, but the relationship is nonlinear.

```
[13]: X = housing_data[['bathrooms', 'bedrooms', 'sqft_living']]
      y = housing_data['price']
      regr = linear_model.LinearRegression()
      regr.fit(X, y)
      print("Coefficients:", regr.coef_)
      print("Intercept:", regr.intercept_)
      print("R² score:", regr.score(X, y))
```

Coefficients: [ 7932.71222265 -57860.8943206 309.39239013]

Intercept: 74847.14080126339

R<sup>2</sup> score: 0.5069197973159074

```
[14]: for column in X.columns:
      r = np.corrcoef(housing_data[column], y)[0, 1] # Pearson correlation
      print(f"Correlation between {column} and price: {r:.4f}")
```

Correlation between bathrooms and price: 0.5251  
Correlation between bedrooms and price: 0.3083  
Correlation between sqft\_living and price: 0.7020