

```
1 from google.colab import drive
```

```
1 drive.mount('/content/gdrive')
```

```
Mounted at /content/gdrive
```

```
1 import pandas as pd
2 df=pd.read_csv
3 ('content/gdrive/mydrive/data.csv')
```

```
'content/gdrive/mydrive/data.csv'
```

```
1 import numpy as np
2 import pandas as pd
```

```
1 df = pd.read_csv('/content/gdrive/MyDrive/data.csv')
```

```
1 #task1--
2 df.head()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basemen
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	28
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	100
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	80

```
1 #task 2--
2 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 18 columns):
#   Column                Non-Null Count  Dtype
---  -
0   date                  4600 non-null  object
1   price                 4600 non-null  float64
2   bedrooms              4600 non-null  float64
3   bathrooms             4600 non-null  float64
4   sqft_living           4600 non-null  int64
5   sqft_lot              4600 non-null  int64
6   floors                4600 non-null  float64
7   waterfront            4600 non-null  int64
8   view                  4600 non-null  int64
9   condition             4600 non-null  int64
10  sqft_above            4600 non-null  int64
11  sqft_basement         4600 non-null  int64
12  yr_built               4600 non-null  int64
13  yr_renovated           4600 non-null  int64
14  street                 4600 non-null  object
15  city                   4600 non-null  object
16  statezip               4600 non-null  object
17  country                4600 non-null  object
dtypes: float64(4), int64(9), object(5)
memory usage: 647.0+ KB
```

```
1 #task 3---
2 df.describe(include='all')
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	s
count	4600	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000
unique	70	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	2014-06-23 00:00:00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	142	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	1.512065
std	NaN	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	0.538288
min	NaN	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	0.000000
25%	NaN	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	1.000000
50%	NaN	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	1.500000
75%	NaN	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	2.000000
max	NaN	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	9.000000

```
1 #task 4--( find the null values)
2 df.isnull()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...
4595	False	False	False	False	False	False	False	False	False	False	False	False	False
4596	False	False	False	False	False	False	False	False	False	False	False	False	False
4597	False	False	False	False	False	False	False	False	False	False	False	False	False
4598	False	False	False	False	False	False	False	False	False	False	False	False	False
4599	False	False	False	False	False	False	False	False	False	False	False	False	False

4600 rows × 18 columns

```
1 #finding null values
2 df.isnull().any()
```

date	False
price	False
bedrooms	False
bathrooms	False
sqft_living	False
sqft_lot	False
floors	False
waterfront	False
view	False
condition	False
sqft_above	False
sqft_basement	False
yr_built	False
yr_renovated	False
street	False
city	False
statezip	False
country	False
dtype:	bool

```
1 #finding null values in numarical
2 df.isnull().sum()
```

date	0
price	0
bedrooms	0
bathrooms	0
sqft_living	0
sqft_lot	0
floors	0
waterfront	0
view	0
condition	0
sqft_above	0
sqft_basement	0
yr_built	0
yr_renovated	0
street	0
city	0
statezip	0
country	0
dtype: int64	

```
1 #selectinga specific data type
2 df.select_dtypes(exclude='object')
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built
0	3.130000e+05	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955
1	2.384000e+06	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921
2	3.420000e+05	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966
3	4.200000e+05	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963
4	5.500000e+05	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976
...
4595	3.081667e+05	3.0	1.75	1510	6360	1.0	0	0	4	1510	0	1954
4596	5.343333e+05	3.0	2.50	1460	7573	2.0	0	0	3	1460	0	1983
4597	4.169042e+05	3.0	2.50	3010	7014	2.0	0	0	3	3010	0	2009
4598	2.034000e+05	4.0	2.00	2090	6630	1.0	0	0	3	1070	1020	1974
4599	2.206000e+05	3.0	2.50	1490	8102	2.0	0	0	4	1490	0	1990

4600 rows × 13 columns

```
1 # selecting a specific data types
2
3 df.select_dtypes(include='object')
```

	date	street	city	statezip	country
0	2014-05-02 00:00:00	18810 Densmore Ave N	Shoreline	WA 98133	USA
1	2014-05-02 00:00:00	709 W Blaine St	Seattle	WA 98119	USA
2	2014-05-02 00:00:00	26206-26214 143rd Ave SE	Kent	WA 98042	USA
3	2014-05-02 00:00:00	857 170th PI NE	Bellevue	WA 98008	USA
4	2014-05-02 00:00:00	9105 170th Ave NE	Redmond	WA 98052	USA
...
4595	2014-07-09 00:00:00	501 N 143rd St	Seattle	WA 98133	USA
4596	2014-07-09 00:00:00	14855 SE 10th PI	Bellevue	WA 98007	USA
4597	2014-07-09 00:00:00	759 Ilwaco PI NE	Renton	WA 98059	USA
4598	2014-07-10 00:00:00	5148 S Creston St	Seattle	WA 98178	USA
4599	2014-07-10 00:00:00	18717 SE 258th St	Covington	WA 98042	USA

4600 rows × 5 columns

```
1 df['city'].unique()
```

```
array(['Shoreline', 'Seattle', 'Kent', 'Bellevue', 'Redmond',
      'Maple Valley', 'North Bend', 'Lake Forest Park', 'Sammamish',
      'Auburn', 'Des Moines', 'Bothell', 'Federal Way', 'Kirkland',
      'Issaquah', 'Woodinville', 'Normandy Park', 'Fall City', 'Renton',
      'Carnation', 'Snoqualmie', 'Duvall', 'Burien', 'Covington',
      'Inglewood-Finn Hill', 'Kenmore', 'Newcastle', 'Mercer Island',
      'Black Diamond', 'Ravensdale', 'Clyde Hill', 'Algona', 'Skykomish',
      'Tukwila', 'Vashon', 'Yarrow Point', 'SeaTac', 'Medina',
      'Enumclaw', 'Snoqualmie Pass', 'Pacific', 'Beaux Arts Village',
      'Preston', 'Milton'], dtype=object)
```

```
1 df['city'].value_counts()
```

```
Seattle      1573
Renton       293
Bellevue     286
Redmond      235
Issaquah     187
Kirkland     187
Kent         185
Auburn       176
Sammamish    175
Federal Way  148
Shoreline    123
Woodinville  115
Maple Valley  96
Mercer Island 86
Burien       74
Snoqualmie   71
Kenmore      66
Des Moines   58
North Bend   50
Covington    43
Duvall       42
Lake Forest Park 36
Bothell      33
Newcastle    33
SeaTac       29
Tukwila      29
Vashon       29
Enumclaw     28
Carnation    22
Normandy Park 18
Clyde Hill   11
Medina       11
Fall City    11
Black Diamond 9
Ravensdale   7
Pacific      6
Algona       5
Yarrow Point 4
Skykomish    3
Preston      2
Milton       2
Inglewood-Finn Hill 1
Snoqualmie Pass 1
Beaux Arts Village 1
Name: city, dtype: int64
```

```
1 df['statezip'].value_counts()
```

```
WA 98103      148
WA 98052      135
WA 98117      132
WA 98115      130
WA 98006      110
...
WA 98047       6
WA 98288       3
WA 98050       2
WA 98354       2
WA 98068       1
Name: statezip, Length: 77, dtype: int64
```

```
1 df.head()
```

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basemen
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	28
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	100
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	80

```
1 # Removing the unwanted columns
2
3 df = df.drop('date',axis=1)
4 df.head()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_ren
0	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	
1	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	
2	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	
3	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	
4	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	

```
1 pd.get_dummies(df['city'])
```

	Algona	Auburn	Beaux Arts Village	Bellevue	Black Diamond	Bothell	Burien	Carnation	Clyde Hill	Covington	...	SeaTac	Seattle	Sh
0	0	0	0	0	0	0	0	0	0	0	...	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	1	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	
3	0	0	0	1	0	0	0	0	0	0	...	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	
...	
4595	0	0	0	0	0	0	0	0	0	0	...	0	1	
4596	0	0	0	1	0	0	0	0	0	0	...	0	0	
4597	0	0	0	0	0	0	0	0	0	0	...	0	0	
4598	0	0	0	0	0	0	0	0	0	0	...	0	1	
4599	0	0	0	0	0	0	0	0	0	1	...	0	0	

4600 rows × 44 columns

```
1 from sklearn.preprocessing import LabelEncoder
```

Double-click (or enter) to edit

```
1 from sklearn.preprocessing import LabelEncoder
```

```
1 le=LabelEncoder()
```

```
1 df['city']=le.fit_transform(df['city'])
```

```
1 le2=LabelEncoder()
```

```
1 df['street']=le2.fit_transform(df['street'])
```

```
1 le3=LabelEncoder()
```

```
1 df['statezip']=le3.fit_transform(df['statezip'])
```

```
1 df.head()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_ren
0	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	
1	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	
2	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	
3	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	
4	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	

Double-click (or enter) to edit

```
1 df['country'].unique()
```

```
array(['USA'], dtype=object)
```

```
1 df['country']=df['country'].replace({'USA':1})
```

```
2 df.head()
```

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_ren
0	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	1955	
1	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	1921	
2	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	1966	
3	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	1963	
4	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	1976	

Double-click (or enter) to edit

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

```
2 y = df['price']
```

```
1 from sklearn.model_selection import train_test_split
```

```
1 x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

```
1 x_train.shape,x_test.shape
```

```
((3680, 16), (920, 16))
```

```
1 x_train.head()
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built	yr_renovated
2258	3.0	1.75	1760	8400	1.0	0	0	4	1460	300	1960	2001
2090	4.0	2.50	2380	13435	2.0	0	0	3	2380	0	1992	0
4256	4.0	1.00	1600	5500	1.5	0	0	4	1600	0	1946	1989
176	4.0	2.50	2000	5390	2.0	0	0	3	2000	0	2003	0
3283	3.0	4.50	4380	47044	2.0	1	3	3	3720	660	1968	1990

Double-click (or enter) to edit

```
1 from sklearn.preprocessing import MinMaxScaler,StandardScaler
```

Double-click (or enter) to edit

```
1 s = StandardScaler()
```

```
1 xtrainscaled = s.fit_transform(x_train)
```

```
1 xtrainscaled
```

```
array([[ -0.44015885, -0.52528171, -0.38824689, ...,  0.77153227,
         1.1529124 ,  0.          ],
       [  0.66098633,  0.43499484,  0.25186372, ..., -0.81809339,
         0.19623571,  0.          ],
       [  0.66098633, -1.48555825, -0.55343672, ...,  0.77153227,
         0.81807556,  0.          ],
       ...,
       [ -0.44015885, -0.52528171, -1.05933059, ...,  0.43687424,
        -0.37777031,  0.          ],
       [ -0.44015885, -0.52528171, -0.47084181, ...,  0.77153227,
         0.38757104,  0.          ],
       [  0.66098633,  0.43499484,  0.11764698, ..., -2.07306102,
         0.29190338,  0.          ]])
```

```
1 xtestscaled = s.transform(x_test)
```

```
1 xtestscaled
```

```
array([[ 0.66098633,  0.43499484,  0.04537642, ...,  0.43687424,
        -0.37777031,  0.          ],
       [ -0.44015885,  0.43499484, -0.24370579, ...,  0.60420326,
         0.14840187,  0.          ],
       [  0.66098633,  1.71536356,  2.07927626, ..., -2.07306102,
         0.29190338,  0.          ],
       ...,
       [  0.66098633,  0.43499484,  0.42737792, ..., -0.39977085,
        -0.76044098,  0.          ],
       [  1.76213151,  0.43499484,  0.29316118, ...,  0.77153227,
         0.86590939,  0.          ],
       [  0.66098633, -1.48555825, -0.22305706, ...,  0.68786776,
         1.58341691,  0.          ]])
```

```
1 #Task 5-- build ml model with linear regression(target colum is price)
```

Linear Regression model is in form $Y=MX+C$

```
1 df=df[['price','sqft_lot']]
```

```
1 df
```

	price	sqft_lot
0	3.130000e+05	7912
1	2.384000e+06	9050
2	3.420000e+05	11947
3	4.200000e+05	8030
4	5.500000e+05	10500
...
4595	3.081667e+05	6360

▾ INDEPENDENT VARIABLE

```

4595 3.081667e+05 6360
1 X=df['sqft_lot']

```

▾ Dependent variable

```
1 Y=df['price']
```

▾ Main process

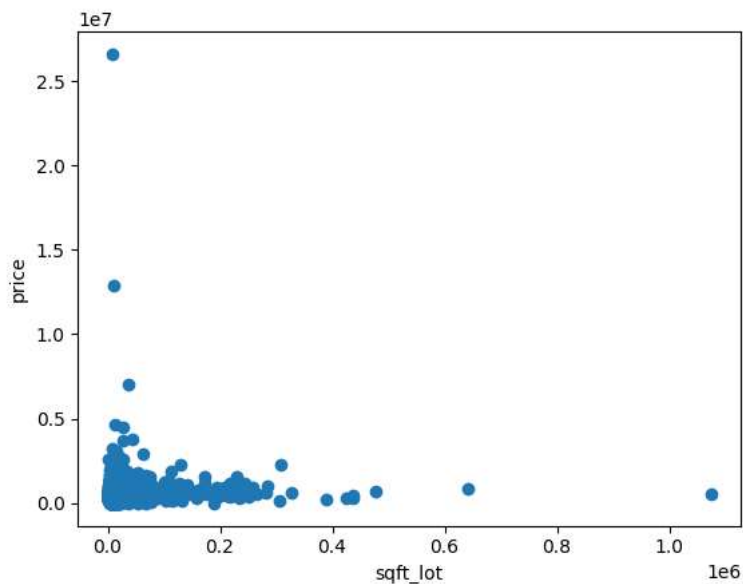
```
1 import matplotlib.pyplot as plt
```

```

1 plt.scatter(X,Y)
2 plt.xlabel('sqft_lot')
3 plt.ylabel('price')

```

```
Text(0, 0.5, 'price')
```



```
1 from sklearn.model_selection import train_test_split
```

```
1 x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.4,random_state=23)
```

```
1 x_train=np.array(x_train).reshape(-1,1)
```

```
1 x_train
```

```

array([[4.00e+00],
       [2.75e+00],

```



```
[3.27e+03],
...,
[3.10e+01],
[3.10e+01],
[1.00e+00]])
```

```
1 x_test=np.array(x_test).reshape(-1,1)
```

```
1 x_test
```

```
array([[3.00e+00],
       [1.50e+00],
       [1.82e+03],
       ...,
       [3.50e+01],
       [4.70e+01],
       [1.00e+00]])
```

FITTING OF LINEAR REGRESSION

```
1 from sklearn.linear_model import LinearRegression
```

```
1 lr=LinearRegression()
```

```
1 lr.fit(x_train,y_train)
```

```
1 c=lr.intercept_
2 c
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-69-bd493491d028> in <cell line: 1>()
----> 1 c=lr.intercept_
      2 c
```

AttributeError: 'LinearRegression' object has no attribute 'intercept_'

SEARCH STACK OVERFLOW

```
1 m=lr.coef_
2 m
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-127-e8f7b8051abf> in <cell line: 1>()
----> 1 m=lr.coef_
      2 m
```

AttributeError: 'LinearRegression' object has no attribute 'coef_'

SEARCH STACK OVERFLOW

```
1 y_pred_train=m*x_train+c
2 y_pred_train.flatten()
```

```
-----
NameError                                    Traceback (most recent call last)
<ipython-input-128-78c892379b55> in <cell line: 1>()
----> 1 y_pred_train=m*x_train+c
      2 y_pred_train.flatten()
```

NameError: name 'm' is not defined

SEARCH STACK OVERFLOW

```
1 plt.scatter(x_train,y_train)
2 plt.scatter(x_train,y_pred_train,color='red')
3 plt.xlabel('sqft_lot')
4 plt.ylabel('price')
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-62-7b000c875324> in <cell line: 1>()  
----> 1 plt.scatter(x_train,y_train)  
      2 plt.scatter(x_train,y_pred_train1,color='red')  
      3 plt.xlabel('sqft_lot')  
      4 plt.ylabel('price')
```

⬆ 2 frames

```
/usr/local/lib/python3.10/dist-packages/matplotlib/axes/_axes.py in scatter(self, x, y,  
s, c, marker, cmap, norm, vmin, vmax, alpha, linewidths, edgecolors, plotnonfinite,  
**kwargs)  
    4582         y = np.ma.ravel(y)  
    4583         if x.size != y.size:  
-> 4584             raise ValueError("x and y must be the same size")  
    4585  
    4586         if s is None:
```

ValueError: x and y must be the same size

SEARCH STACK OVERFLOW

