1--

```
import pandas as pd
df=pd.read_csv('kc_house_data.csv')
df.head()
```

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080

```
df.info()
```

```
<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 
                                ull  float64
                                ull  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
```

Le notebook a bien été enregistré. ×

```
df.columns
```

```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
            'lat', 'long', 'sqft_living15', 'sqft_lot15'],
           dtype='object')
```

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

```
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
```

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

```
0
```

Le notebook a bien été enregistré. ×

```
3     9824
4     6882
2     2760
5     1601
6      272
1      199
7       38
8       13
0       13
9        6
10       3
11       1
33       1
Name: bedrooms, dtype: int64
```

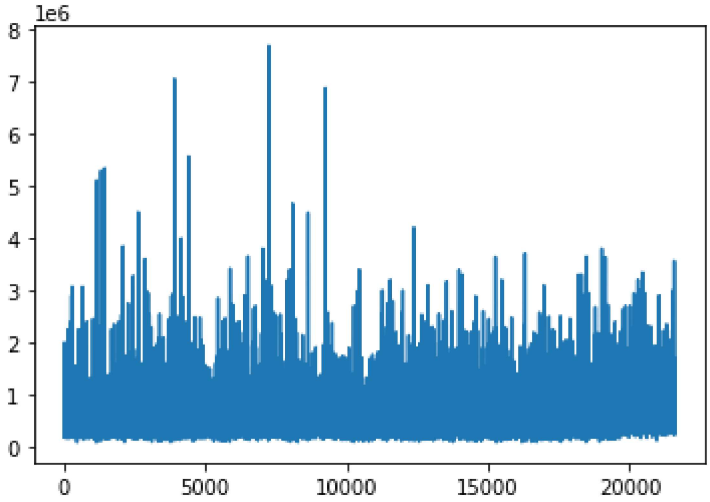
```
df['grade'].value_counts()
```

```
7     8981
8     6068
9     2615
6     2038
10    1134
11     399
5      242
```

```
        12      90
        4       29
        13      13
        3        3
        1        1
        Name: grade, dtype: int64
```

```
import matplotlib.pyplot as plt
plt.plot(df['price'])
```

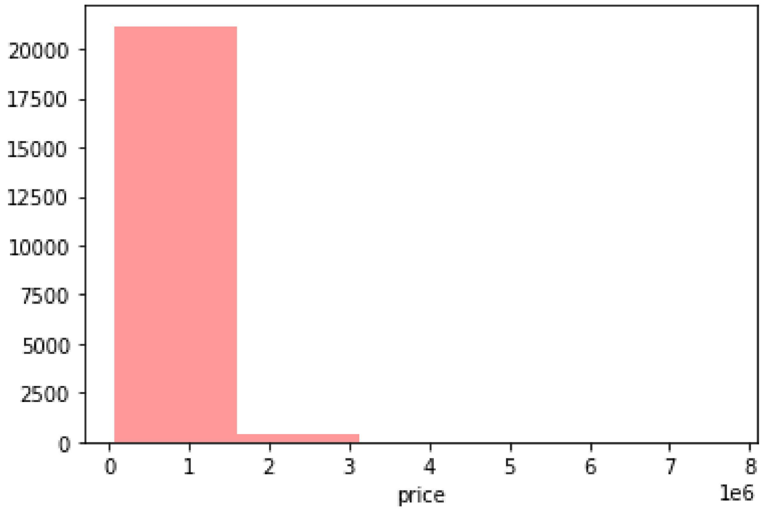
```
        [<matplotlib.lines.Line2D at 0x7fc953c53790>]
```



```
import seaborn as sns
sns.distplot(df['price'],bins=5,hist=True,kde=False,color='red')
```

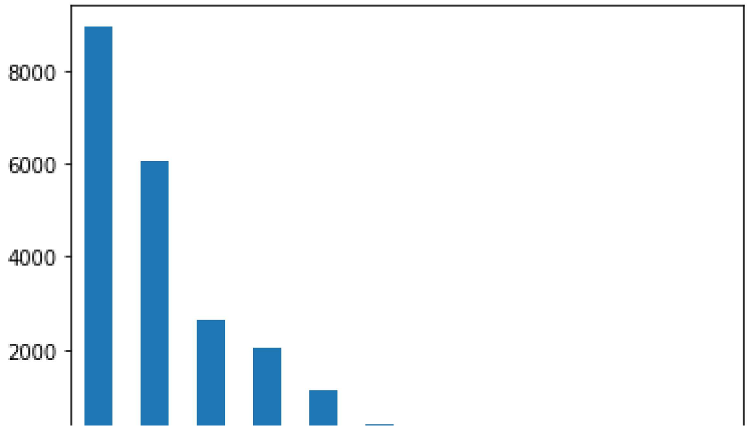
Le notebook a bien été enregistré.

```
packages/seaborn/distributions.py:2557: FutureWarning:
ning)
        <matplotlib.axes._subplots.AxesSubplot at 0x7fc94fc3ae10>
```



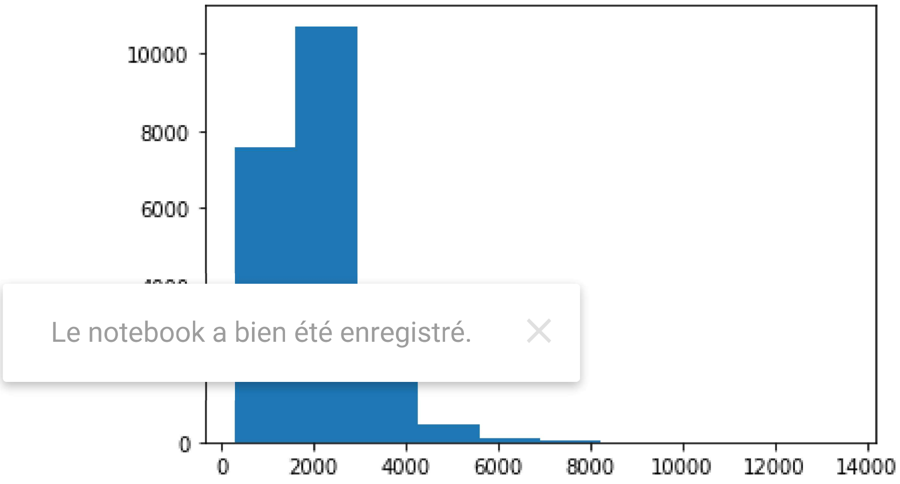
```
import matplotlib.pyplot as plt
v=df['grade'].value_counts()
v.plot.bar(rot=45)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc94fa80b10>
```



```
import matplotlib.pyplot as plt
plt.hist('sqft_living',data=df)
```

```
(array([7.5690e+03, 1.0681e+04, 2.8140e+03, 4.4100e+02, 7.7000e+01,
        2.4000e+01, 2.0000e+00, 3.0000e+00, 1.0000e+00, 1.0000e+00]),
 array([  290.,  1615.,  2940.,  4265.,  5590.,  6915.,  8240.,  9565.,
        10890., 12215., 13540.]),
 <a list of 10 Patch objects>)
```



Le notebook a bien été enregistré.

```
3--
```

```
from sklearn.model_selection import train_test_split
x=df[["id","grade","bathrooms","bedrooms"]]
y=df["price"].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=40)
```

```
4--
```

```
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
x=df[["grade"]]
y=df["price"].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=40)
model=LinearRegression()
model.fit(x_train,y_train)
```

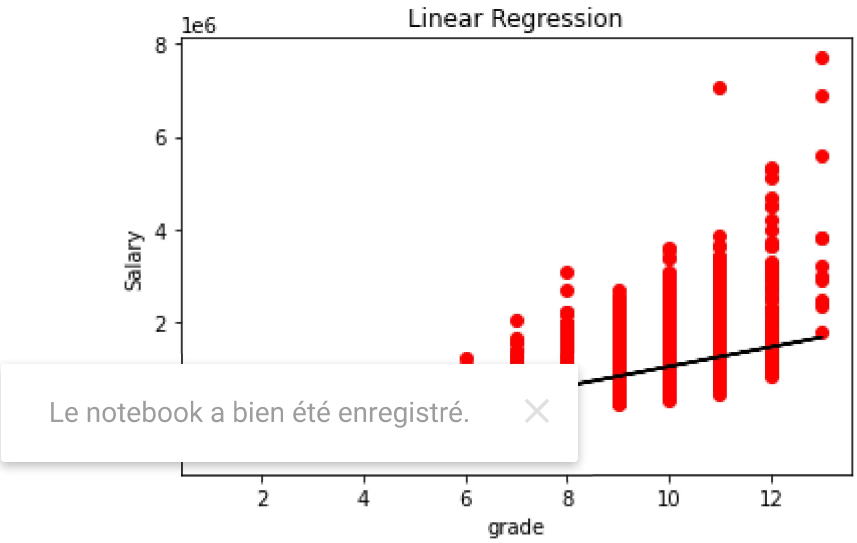
```
predicted=model.predict(x_test)

print("MSE", mean_squared_error(y_test,predicted))
print("R squared", metrics.r2_score(y_test,predicted))
```

```
    MSE 68533946794.312935
    R squared 0.4552042311532961
```

5--

```
plt.scatter(x,y,color="r")
plt.title("Linear Regression")
plt.ylabel("Salary")
plt.xlabel("grade")
plt.plot(x,model.predict(x),color="k")
plt.show()
```



Le notebook a bien été enregistré.

6-- R-squared=0.455<0.5 the half of the output can be explained by the model's inputs donc la correlation est faible

```
x=df[['grade','bedrooms','bathrooms']]  #we have more than one input
y=df["price"].values
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.35,random_state=40) #splitt

model=LinearRegression()
model.fit(x_train,y_train)
predicted=model.predict(x_test)

print("MSE", mean_squared_error(y_test,predicted))
print("R squared", metrics.r2_score(y_test,predicted))
```

```
    MSE 66370581890.284134
    R squared 0.4720658164992009
```

7--

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
x =df[['sqft_living','sqft_lot']]
y = df['price'].values
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.35, random_state=40)
lg=LinearRegression()
poly=PolynomialFeatures(degree=2)

x_train_fit = poly.fit_transform(x_train) #transforming our input data
lg.fit(x_train_fit, y_train)
x_test_ = poly.fit_transform(x_test)
predicted = lg.predict(x_test_)

print("MSE: ", metrics.mean_squared_error(y_test, predicted))
print("R squared: ", metrics.r2_score(y_test,predicted))
```

```
MSE:  56768005841.851654
R squared:  0.5484479725877804
```

Le notebook a bien été enregistré. ✕

✓ 0 s terminée à 10:42 ● ✕