In [19]:
```
import pandas as pd
import numpy as np
```

In [20]:
```
db = pd.read_csv('lec06.csv')
db.head()
```

Out[20]:

	Color	Quality	Price
0	7	5	58
1	3	7	11
2	5	8	24
3	8	1	11
4	9	3	31

In [21]:
```
db['Price'] = np.log(db['Price'])
db.head()
```

Out[21]:

	Color	Quality	Price
0	7	5	4.060443
1	3	7	2.397895
2	5	8	3.178054
3	8	1	2.397895
4	9	3	3.433987

In [37]:
```
db = pd.concat([pd.Series(1, index=db.index, name='00'), db], axis=1)
X = db.drop(columns='Price')
y = db.iloc[:, 3]
```

Out[37]:
```
0     4.060443
1     2.397895
2     3.178054
3     2.397895
4     3.433987
5     2.708050
6     1.609438
7     2.079442
8     4.430817
9     3.178054
10    3.044522
Name: Price, dtype: float64
```

In [38]:
```
for i in X.columns:
    X[i] = X[i]/np.max(X[i])
X.head()
```

Out[38]:

	00	Color	Quality
0	1.0	0.777778	0.625
1	1.0	0.333333	0.875
2	1.0	0.555556	1.000
3	1.0	0.888889	0.125
4	1.0	1.000000	0.375

In [50]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(X, y)
reg.score(X, y)
```

Out[50]:
```
0.8454824413307531
```

In [51]:
```
reg.coef_
```

Out[51]:
```
array([0.        , 2.69491251, 1.84009513])
```

In [52]:
```
reg.intercept_
```

Out[52]:
```
0.17684970200766736
```

In [53]:
```
db = pd.read_csv('lec06.csv')
db.head()
```

Out[53]:

	Color	Quality	Price
0	7	5	58
1	3	7	11
2	5	8	24
3	8	1	11
4	9	3	31

In [54]:
```
X = db.drop(columns='Price')
y = db.iloc[:, 2]
```

In [55]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
reg = LinearRegression().fit(X, y)
reg.score(X, y)
```

Out[55]:
```
0.6096344612876989
```

In [57]:
```
#importing libararies
import pandas as pd
import seaborn as sns
import numpy as np
import statsmodels.api  as sm
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
import matplotlib.pyplot as plt
```

In [58]:
```
df = pd.read_csv('lec06.csv')
df
```

Out[58]:

	Color	Quality	Price
0	7	5	58
1	3	7	11
2	5	8	24
3	8	1	11
4	9	3	31
5	5	4	15
6	4	0	5
7	2	6	8
8	8	7	84
9	6	4	24
10	9	2	21

In [59]:
```
df.describe
```

Out[59]:
```
<bound method NDFrame.describe of      Color  Quality  Price
0        7        5     58
1        3        7     11
2        5        8     24
3        8        1     11
4        9        3     31
5        5        4     15
6        4        0      5
7        2        6      8
8        8        7     84
9        6        4     24
10       9        2     21>
```

In [60]:
```
Xk = df[['Color', 'Quality']]
Yk = df['Price']
Xm = sm.add_constant(Xk)

Xm = sm.OLS(Yk,Xm ).fit()
Xm.summary()
```

```
/Users/wendellwang/Developer/ML/lib/python3.9/site-packages/scipy/stats/stats.py:1603: User
Warning: kurtosistest only valid for n>=20 ... continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

Out[60]:

OLS Regression Results

Dep. Variable:	Price	**R-squared:**	0.610
Model:	OLS	**Adj. R-squared:**	0.512
Method:	Least Squares	**F-statistic:**	6.247
Date:	Thu, 16 Sep 2021	**Prob (F-statistic):**	0.0232
Time:	13:52:30	**Log-Likelihood:**	-44.887
No. Observations:	11	**AIC:**	95.77
Df Residuals:	8	**BIC:**	96.97
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-41.1565	19.850	-2.073	0.072	-86.930	4.617
Color	7.1180	2.339	3.043	0.016	1.724	12.512
Quality	5.8497	2.157	2.711	0.027	0.875	10.825

Omnibus:	1.855	**Durbin-Watson:**	1.552
Prob(Omnibus):	0.396	**Jarque-Bera (JB):**	1.321
Skew:	0.717	**Prob(JB):**	0.517
Kurtosis:	2.090	**Cond. No.**	30.3

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [71]:
```
def printModelResult(db):
    Xk = db[['Color', 'Quality']]
    Ym = db['Price']
    Xm = sm.add_constant(Xk)

    Xm = sm.OLS(Ym,Xm ).fit()
    print(Xm.summary())
```

In [72]:
```
db = pd.read_csv('lec06.csv')
db.head()
```

Out[72]:

	Color	Quality	Price
0	7	5	58
1	3	7	11
2	5	8	24
3	8	1	11
4	9	3	31

In [73]:
```
printModelResult(db)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.610
Model:                            OLS   Adj. R-squared:                  0.512
Method:                 Least Squares   F-statistic:                     6.247
Date:                Thu, 16 Sep 2021   Prob (F-statistic):             0.0232
Time:                        13:56:15   Log-Likelihood:                -44.887
No. Observations:                  11   AIC:                             95.77
Df Residuals:                       8   BIC:                             96.97
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -41.1565     19.850     -2.073      0.072     -86.930       4.617
Color          7.1180      2.339      3.043      0.016       1.724      12.512
Quality        5.8497      2.157      2.711      0.027       0.875      10.825
==============================================================================
Omnibus:                        1.855   Durbin-Watson:                   1.552
Prob(Omnibus):                  0.396   Jarque-Bera (JB):                1.321
Skew:                           0.717   Prob(JB):                        0.517
Kurtosis:                       2.090   Cond. No.                        30.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
/Users/wendellwang/Developer/ML/lib/python3.9/site-packages/scipy/stats/stats.py:1603: User
Warning: kurtosistest only valid for n>=20 ... continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

In [74]:
```
db['Price'] = np.log(db['Price'])
db.head()
```

Out[74]:

	Color	Quality	Price
0	7	5	4.060443
1	3	7	2.397895
2	5	8	3.178054
3	8	1	2.397895
4	9	3	3.433987

In [75]:
```
printModelResult(db)
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Price   R-squared:                       0.845
Model:                            OLS   Adj. R-squared:                  0.807
Method:                 Least Squares   F-statistic:                     21.89
Date:                Thu, 16 Sep 2021   Prob (F-statistic):           0.000570
Time:                        13:56:18   Log-Likelihood:                -2.8482
No. Observations:                  11   AIC:                             11.70
Df Residuals:                       8   BIC:                             12.89
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1768      0.435      0.407      0.695      -0.825       1.179
Color          0.2994      0.051      5.848      0.000       0.181       0.418
Quality        0.2300      0.047      4.870      0.001       0.121       0.339
==============================================================================
Omnibus:                        0.795   Durbin-Watson:                   1.427
Prob(Omnibus):                  0.672   Jarque-Bera (JB):                0.702
Skew:                           0.466   Prob(JB):                        0.704
Kurtosis:                       2.186   Cond. No.                        30.3
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
/Users/wendellwang/Developer/ML/lib/python3.9/site-packages/scipy/stats/stats.py:1603: User
Warning: kurtosistest only valid for n>=20 ... continuing anyway, n=11
  warnings.warn("kurtosistest only valid for n>=20 ... continuing "
```

In [ ]:
```

```