DATA SCIENCE PROJECT

1] DATASET SELECTION

```
Dataset name: camera_dataset
Source: Kaggle
No. of observations: 1038
Percentage of missing values: 5%
```

2] EXPLORATORY DATA ANALYSIS

Describe the features of the dataset

```
MODEL:
This feature of the dataset describes about the various models of the different
 type(company) of cameras present in the choosen dataset.

RELEASE DATE:
This feature of the dataset describes about the various release dates(in years)o
f the different models of the cameras in the dataset.

MAX RESOLUTION:
This feature of the dataset describes about the maximum resolution of the differ
ent models of the cameras present in the dataset.

MIN RESOLUTION:
This feature of the dataset describes about the minimum resolution of the differ
ent models of the cameras present in the dataset.

EFFECTIVE PIXELS:
This feature of the dataset describes about the different number of effective pi
xels of each camera model in the dataset.

ZOOM WIDE:
This feature describes about the short focal length of the different cameras in
 the dataset.

ZOOM TELE:
This feature describes about the long focal length of the different cameras in t
he dataset.

STORAGE INCLUDED:
This feature of the dataset describes about the inbuilt storage that is included
in the various cameras of the dataset.

WEIGHT:
This feature of the dataset describes about the weight of the cameras(inclusive
 of the weight of the batteries)in the dataset.

DIMENSION:
This feature of the dataset describes about the dimensions of the cameras in the
dataset.

PRICE:
This feature of the dataset describes about the prices of the cameras in the dat
aset.
```

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.api as sm
```

In [2]:

```
dataset=pd.read_csv("C:/Users/Rahul Bhat/Desktop/SDS project/camera_dataset.csv")
dataset.describe()
```

Out[2]:

	Release date	Max resolution	Low resolution	Effective pixels	Zoom wide (W)	Zoom tele (T)	Norn foc ran
count	1038.000000	1037.000000	984.000000	1003.000000	953.000000	953.000000	901.0000
mean	2003.590559	2477.058824	1871.286585	4.756730	35.903463	132.364113	50.8579
std	2.724755	755.976743	738.892348	2.758156	3.261553	89.875022	18.1602
min	1994.000000	512.000000	320.000000	1.000000	23.000000	28.000000	1.0000
25%	2002.000000	2048.000000	1280.000000	3.000000	35.000000	102.000000	40.0000
50%	2004.000000	2560.000000	2048.000000	5.000000	36.000000	111.000000	50.0000
75%	2006.000000	3072.000000	2560.000000	7.000000	38.000000	117.000000	60.0000
max	2007.000000	5616.000000	4992.000000	21.000000	52.000000	518.000000	120.0000

Data cleaning and missing value treatment

    No.of missing values in each column

In [3]:

```
dataset.isnull().sum()
```

Out[3]:

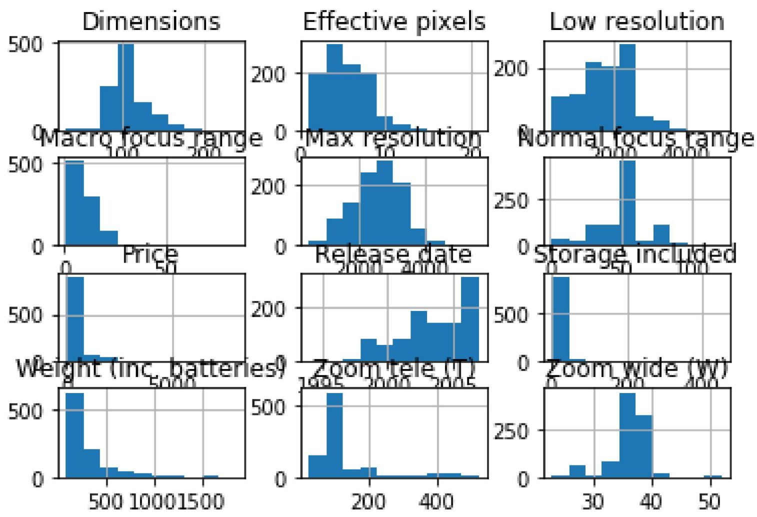
```
Model                        0
Release date                 0
Max resolution               1
Low resolution              54
Effective pixels            35
Zoom wide (W)               85
Zoom tele (T)               85
Normal focus range         137
Macro focus range          128
Storage included           125
Weight (inc. batteries)     23
Dimensions                  16
Price                        0
dtype: int64
```

In [4]:

```
dataset.hist(column=["Release date","Max resolution","Low resolution","Effective pixel
s","Zoom wide (W)","Zoom tele (T)","Normal focus range","Macro focus range","Storage in
cluded","Weight (inc. batteries)","Dimensions","Price"])
```

Out[4]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C73C6CD4
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C73F90A0
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C73FD054
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7400864
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7404070
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7407784
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C740AF94
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C740E7A4
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7411FBC
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C74157D0
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7418FD0
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C741C8E0
8>]],
      dtype=object)
```



Replace the missing values with the mean of that column

Print the original dataset

In [5]:

```
dataset
```

Out[5]:

	Model	Release date	Max resolution	Low resolution	Effective pixels	Zoom wide (W)	Zoom tele (T)	Normal focus range	Macro focus range	Stor inclu
0	Agfa ePhoto 1280	1997	1024.0	640.0	NaN	38.0	114.0	70.0	40.0	
1	Agfa ePhoto 1680	1998	1280.0	640.0	1.0	38.0	114.0	50.0	NaN	
2	Agfa ePhoto CL18	2000	640.0	NaN	NaN	45.0	45.0	NaN	NaN	
3	Agfa ePhoto CL30	1999	1152.0	640.0	NaN	35.0	35.0	NaN	NaN	
4	Agfa ePhoto CL30 Clik!	1999	1152.0	640.0	NaN	43.0	43.0	50.0	NaN	
...	...	...	...	...	...	...	...	...	...	
1033	Toshiba PDR-M65	2001	2048.0	1024.0	3.0	38.0	114.0	10.0	10.0	
1034	Toshiba PDR-M70	2000	2048.0	1024.0	3.0	35.0	105.0	80.0	9.0	
1035	Toshiba PDR-M71	2001	2048.0	1024.0	3.0	35.0	98.0	80.0	10.0	
1036	Toshiba PDR-M81	2001	2400.0	1200.0	3.0	35.0	98.0	80.0	10.0	
1037	Toshiba PDR-T10	2002	1600.0	800.0	1.0	38.0	38.0	40.0	20.0	

1038 rows × 13 columns

Replace the NaN value with mean

In [6]:

```
for i in dataset:
       if(i=="Model"):
            continue;
       if(i=='Price' or i=='Weight (inc. batteries)'):
           dataset[i].fillna(dataset[i].mean(),inplace=True)
       else:
           dataset[i].fillna(int(dataset[i].mean()),inplace=True)
```

Cleaned dataset

In [7]:

```
dataset
```

Out[7]:

	Model	Release date	Max resolution	Low resolution	Effective pixels	Zoom wide (W)	Zoom tele (T)	Normal focus range	Macro focus range	Stor inclu
0	Agfa ePhoto 1280	1997	1024.0	640.0	4.0	38.0	114.0	70.0	40.0	
1	Agfa ePhoto 1680	1998	1280.0	640.0	1.0	38.0	114.0	50.0	8.0	
2	Agfa ePhoto CL18	2000	640.0	1871.0	4.0	45.0	45.0	50.0	8.0	
3	Agfa ePhoto CL30	1999	1152.0	640.0	4.0	35.0	35.0	50.0	8.0	
4	Agfa ePhoto CL30 Clik!	1999	1152.0	640.0	4.0	43.0	43.0	50.0	8.0	
...	...	...	...	...	...	...	...	...	...	
1033	Toshiba PDR-M65	2001	2048.0	1024.0	3.0	38.0	114.0	10.0	10.0	
1034	Toshiba PDR-M70	2000	2048.0	1024.0	3.0	35.0	105.0	80.0	9.0	
1035	Toshiba PDR-M71	2001	2048.0	1024.0	3.0	35.0	98.0	80.0	10.0	
1036	Toshiba PDR-M81	2001	2400.0	1200.0	3.0	35.0	98.0	80.0	10.0	
1037	Toshiba PDR-T10	2002	1600.0	800.0	1.0	38.0	38.0	40.0	20.0	

1038 rows × 13 columns

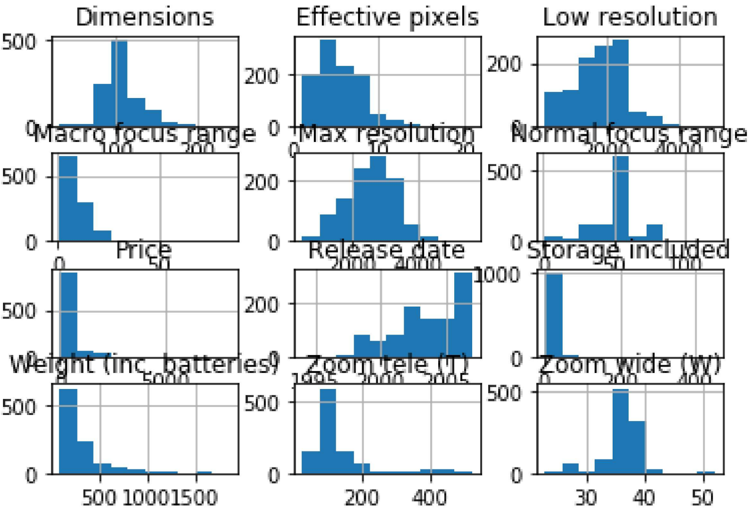
3] GRAPH VISUALIZATION

In [8]:

```
dataset.hist(column=["Release date","Max resolution","Low resolution","Effective pixel
s","Zoom wide (W)","Zoom tele (T)","Normal focus range","Macro focus range","Storage in
cluded","Weight (inc. batteries)","Dimensions","Price"])
```

Out[8]:

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7444628
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C744BB74
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C744E610
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7451E20
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7455630
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7458E44
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C745C850
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7460060
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7463874
8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000029C7467088
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C746A994
8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000029C746E1A0
8>]],
      dtype=object)
```



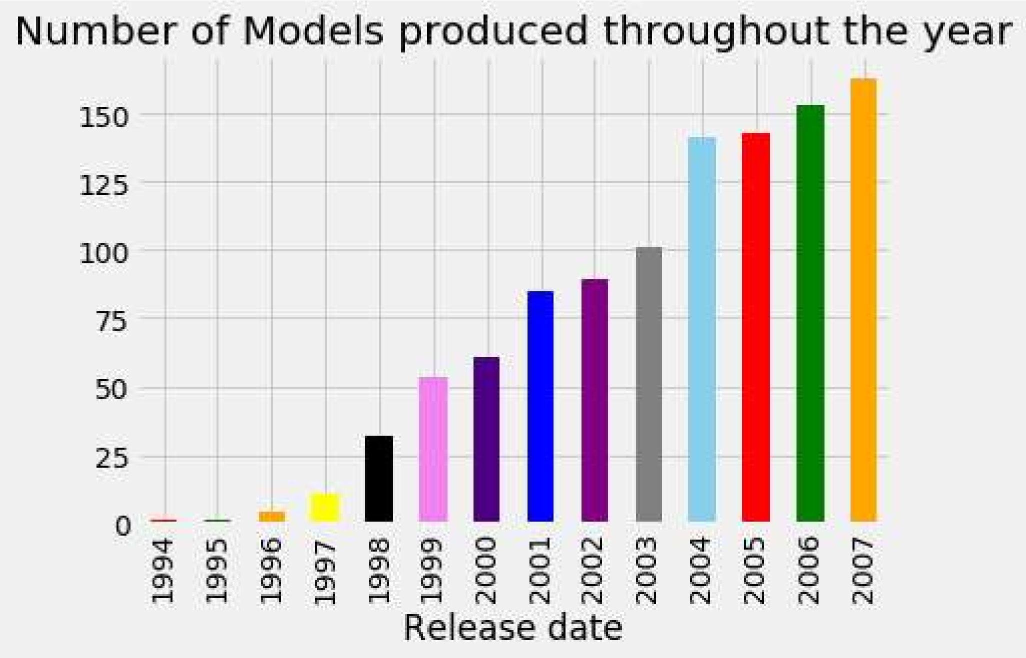
In [9]:

```
plt.style.use('fivethirtyeight')
print(dataset.pivot_table(index=['Release date'], aggfunc='size').plot(kind='bar',color
=['red','green','orange','yellow','black','violet','indigo','blue','purple','grey','sky
blue']))
plt.ylabel=["No. of cameras"]
plt.title("Number of Models produced throughout the year")
```

```
AxesSubplot(0.08,0.07;0.87x0.81)
```

Out[9]:

```
Text(0.5, 1.0, 'Number of Models produced throughout the year')
```



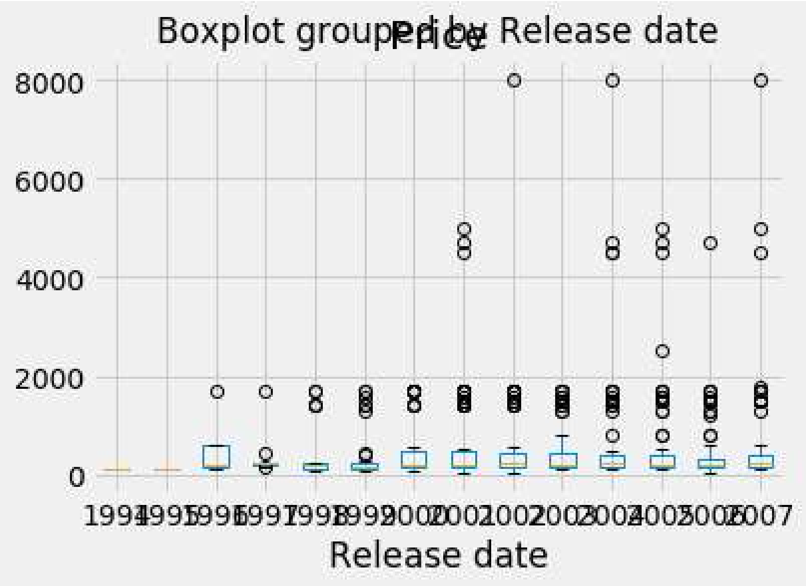
Check for outliers

In [10]:

```
df=dataset[['Release date','Price']]
df.boxplot(column='Price',by='Release date')
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29c74990b48>
```



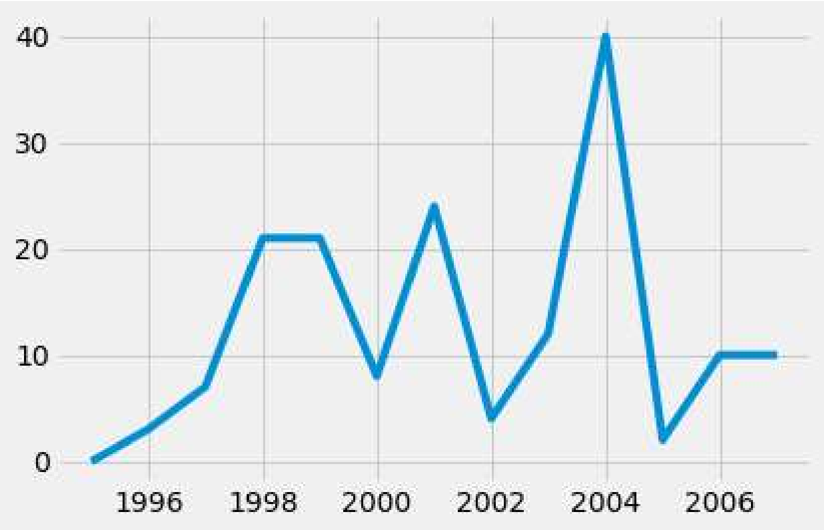
Growth in production

In [14]:

```
growth_rate=[]
growth=list(dataset["Release date"].value_counts())
dates=set(dataset["Release date"])
j=0
for i in growth:
    growth_rate.append(int(j-i))
    j=i
del(growth_rate[0])
growth.clear()
for i in growth_rate:
    growth.insert(0,i)
del(growth_rate)
dates=list(dates)
del(dates[0])
plt.plot(dates,growth)
```

Out[14]:

```
[<matplotlib.lines.Line2D at 0x29c74d0c8c8>]
```



Adding a new feature to the dataframe

In [15]:

```
Manufacturer=list()
release_dates=set(df["Release date"])
for i in dataset['Model']:
    if('Agfa' in i):
        Manufacturer.append('Agfa')
    elif('Canon' in i):
        Manufacturer.append('Canon')
    elif('Casio' in i):
        Manufacturer.append('Casio')
    elif('Contax' in i):
        Manufacturer.append('Contax')
    elif('Epson' in i):
        Manufacturer.append('Epson')
    elif('Fujifilm' in i):
        Manufacturer.append('Fujifilm')
    elif('HP' in i):
        Manufacturer.append('HP')
    elif('JVC' in i):
        Manufacturer.append('JVC')
    elif('Kodak' in i):
        Manufacturer.append('Kodak')
    elif('Kyocera' in i):
        Manufacturer.append('Kyocera')
    elif('Leica' in i):
        Manufacturer.append('Leica')
    elif('Nikon' in i):
        Manufacturer.append('Nikon')
    elif('Olympus' in i):
        Manufacturer.append('Olympus')
    elif('Panasonic' in i):
        Manufacturer.append('Panasonic')
    elif('Pentax' in i):
        Manufacturer.append('Pentax')
    elif('Ricoh' in i):
        Manufacturer.append('Ricoh')
    elif('Samsung' in i):
        Manufacturer.append('Samsung')
    elif('Sanyo' in i):
        Manufacturer.append('Sanyo')
    elif('Sigma' in i):
        Manufacturer.append('Sigma')
    elif('Sony' in i):
        Manufacturer.append('Sony')
    elif('Toshiba' in i):
        Manufacturer.append('Toshiba')
dataset['Manufacturer']=Manufacturer
```

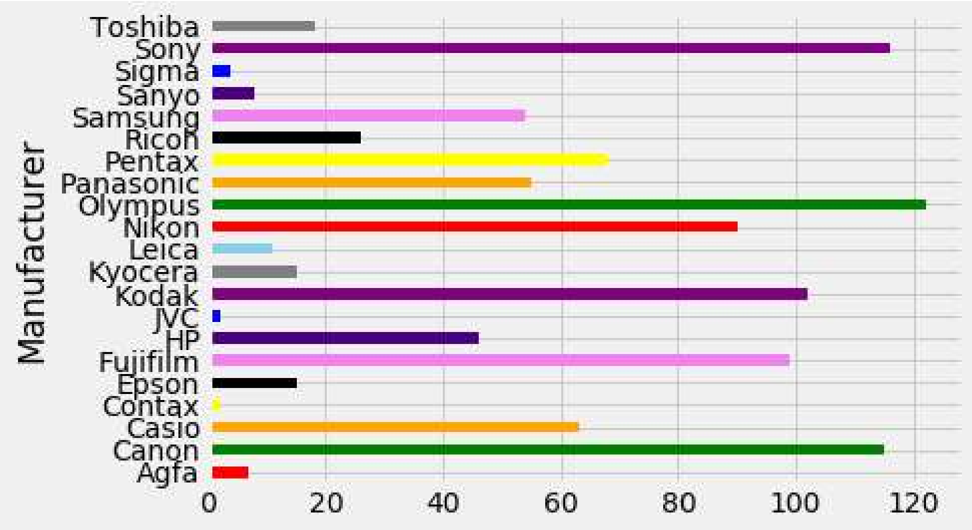
Plot representing no.of models manufactured by individual manufacturers

In [17]:

```
dataset.pivot_table(index=['Manufacturer'], aggfunc='size').plot(kind='barh',color=['re
d','green','orange','yellow','black','violet','indigo','blue','purple','grey','skyblue'
])
```

Out[17]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29c74b9da08>
```



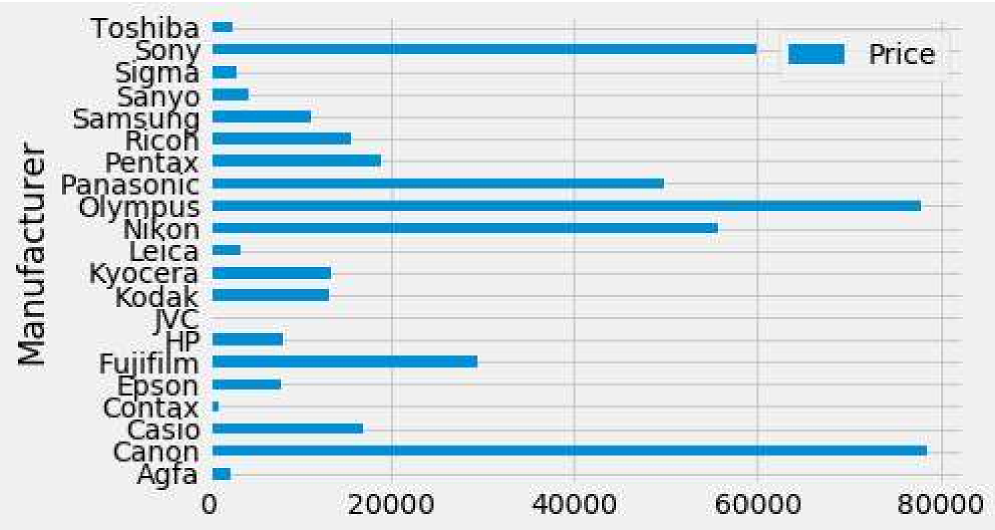
Accumulated price of all manufacturers throughout the year

In [18]:

```
turn_over=dataset[['Manufacturer','Price']]
turn_over.pivot_table(index=['Manufacturer'], aggfunc='sum').plot(kind='barh')
```

Out[18]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x29c74dff988>
```



4] NORMALIZATION

Mean and variance of each column

In [22]:

```
dataset.mean()
```

Out[22]:

```
Release date                2003.590559
Max resolution              2477.058767
Low resolution              1871.271676
Effective pixels               4.731214
Zoom wide (W)                 35.829480
Zoom tele (T)                132.334297
Normal focus range            50.744701
Macro focus range              8.766859
Storage included              19.702312
Weight (inc. batteries)      325.870936
Dimensions                   106.794316
Price                        457.384393
dtype: float64
```

In [23]:

```
dataset.std()**2
```

Out[23]:

```
Release date                     7.424289
Max resolution              570949.725569
Low resolution              517531.874048
Effective pixels                 7.369343
Zoom wide (W)                    9.827211
Zoom tele (T)                 7415.437800
Normal focus range             286.307952
Macro focus range               57.166422
Storage included               710.631644
Weight (inc. batteries)      65575.780224
Dimensions                     435.628097
Price                       578288.639949
dtype: float64
```

Make mean 0 and variance 1

In [24]:

```
numeric_dataset = dataset[["Release date","Max resolution","Low resolution","Effective pixels","Zoom wide (W)","Zoom tele (T)","Normal focus range","Macro focus range","Storage included","Weight (inc. batteries)","Dimensions","Price"]]
normalized_dataset=(numeric_dataset-numeric_dataset.mean())/numeric_dataset.std()
```

Print normalized dataset

In [25]:

```
normalized_dataset
```

Out[25]:

	Release date	Max resolution	Low resolution	Effective pixels	Zoom wide (W)	Zoom tele (T)	Normal focus range	Macro focus range
0	-2.418771	-1.923022	-1.711533	-0.269358	0.692387	-0.212910	1.137977	4.130904
1	-2.051766	-1.584224	-1.711533	-1.374472	0.692387	-0.212910	-0.044011	-0.101425
2	-1.317755	-2.431219	-0.000378	-0.269358	2.925357	-1.014183	-0.044011	-0.101425
3	-1.684760	-1.753623	-1.711533	-0.269358	-0.264600	-1.130310	-0.044011	-0.101425
4	-1.684760	-1.753623	-1.711533	-0.269358	2.287365	-1.037409	-0.044011	-0.101425
...	...	...	...	...	...	...	...	...
1033	-0.950749	-0.567829	-1.177753	-0.637729	0.692387	-0.212910	-2.407989	0.163096
1034	-1.317755	-0.567829	-1.177753	-0.637729	-0.264600	-0.317424	1.728971	0.030835
1035	-0.950749	-0.567829	-1.177753	-0.637729	-0.264600	-0.398712	1.728971	0.163096
1036	-0.950749	-0.101982	-0.933103	-0.637729	-0.264600	-0.398712	1.728971	0.163096
1037	-0.583744	-1.160726	-1.489125	-1.374472	0.692387	-1.095472	-0.635006	1.485699

1038 rows × 12 columns

Verify that the mean is 0 and variance is 1

In [26]:

```
normalized_dataset.mean().round(decimals=1)
```

Out[26]:

```
Release date              -0.0
Max resolution             0.0
Low resolution             0.0
Effective pixels           0.0
Zoom wide (W)             -0.0
Zoom tele (T)              0.0
Normal focus range         0.0
Macro focus range         -0.0
Storage included          -0.0
Weight (inc. batteries)   -0.0
Dimensions                 0.0
Price                     -0.0
dtype: float64
```

In [27]:

```
normalized_dataset.var()
```

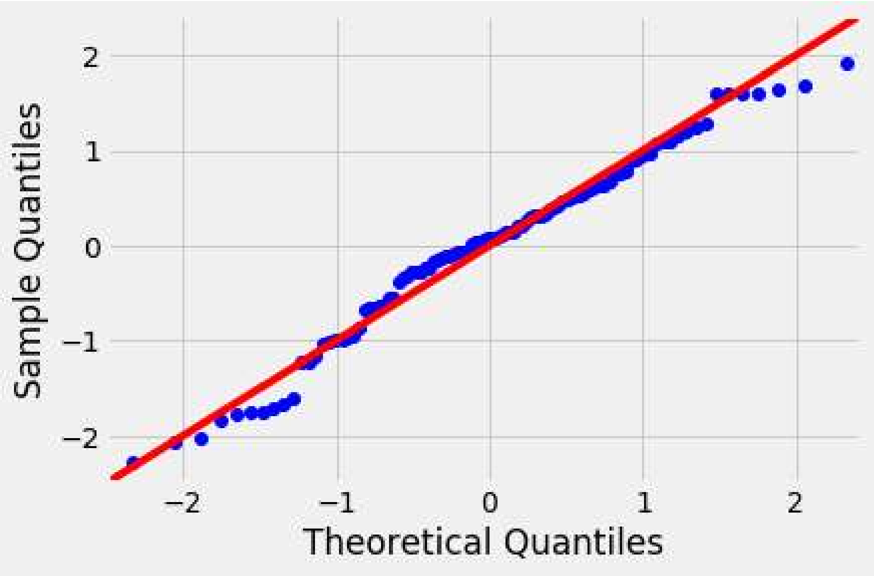
Out[27]:

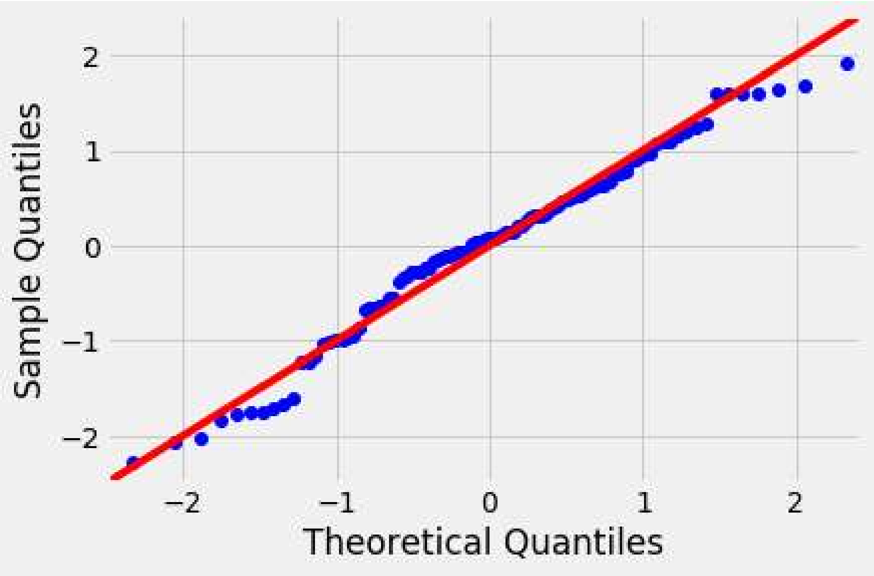
```
Release date               1.0
Max resolution             1.0
Low resolution             1.0
Effective pixels           1.0
Zoom wide (W)              1.0
Zoom tele (T)              1.0
Normal focus range         1.0
Macro focus range          1.0
Storage included           1.0
Weight (inc. batteries)    1.0
Dimensions                 1.0
Price                      1.0
dtype: float64
```

In [28]:

```
data_points = np.random.normal(0, 1, 100)
sm.qqplot(data_points, line ='45')
```

Out[28]:





5] HYPOTHESIS TESTING

HYPOTHESIS 1: It has been observed that the mean release date of a camera model is 2004.Discuss the validity of this statement. The standard deviation of the release date is 2.72. WE TEST THE ABOVE HYPOTHESIS BASED ON THE EVIDENCE TAKEN FROM THE BELOW SMALL SAMPLE FROM THE SAME POPULATION: SAMPLE:

```
SAMPLE Size(n)=5
SAMPLE VALUES={1997,1998,2000,1999,1999}
SAMPLE MEAN(X_bar)=1998.6
```

STEP1:FORMUALTING THE NULL HYPOTHESIS AND ALTERNATE HYPOTHESIS-> Null Hypothesis: The mean release date of the camera is not equal to 2004. EQ->H0:mean!=2004 Alternate Hypothesis: The mean release date of the camera is equal to 2004. EQ->H1:mean=2004

STEP2:ASSUMING THAT THE NULL HYPOTHESIS(H0) is TRUE

STEP3:CALCULATING THE VALUE OF THE TEST STATISTIC-> z-score=(X_bar-mean)/(std/sqrt(n)) =>z-score=(1998.6-2004)/(2.72/sqrt(5)) [std=2.72] =>z-score=-4.44

STEP4:CALCULATING THE P-VALUE OF THE TEST STATISTIC: P-value=P(z<-4.44)+P(z>4.44) =>P-value=P(z<-4.44)+1-P(z<4.44) =>P-value=(<0.0001)+1-(>0.9999)[Obtained from the table] =>P-value=approx(0)+1-approx(1) =>P-value=0+1-1 =>P-value=0+0=0 =>Therefore P-value=0

STEP5:FORMING THE CONCLUSION ON THE BASIS OF THE P-VALUE OBTAINED IN THE TEST-> =>Therefore the Null Hypothesis(H0) is rejected. Therefore the mean release date of the camera is equal to 2004.

HYPOTHESIS 2: It has been observed that the mean price of a camera model is less than 460.Discuss the validity of this statement.The standard deviation of the release date is 760.45. WE TEST THE ABOVE HYPOTHESIS BASED ON THE EVIDENCE TAKEN FROM THE BELOW SMALL SAMPLE FROM THE SAME POPULATION: SAMPLE:

```
SAMPLE Size(n)=5
SAMPLE VALUES={179,179,179,269,1299}
SAMPLE MEAN(X_bar)=421
```

STEP1:FORMUALTING THE NULL HYPOTHESIS AND ALTERNATE HYPOTHESIS-> Null Hypothesis: The mean price of the camera is greater than or equal to 460. EQ->H0:mean>=460 Alternate Hypothesis: The mean price of the camera is less than 460. EQ->H1:mean<460

STEP2:ASSUMING THAT THE NULL HYPOTHESIS(H0) is TRUE

STEP3:CALCULATING THE VALUE OF THE TEST STATISTIC-> NOTE:IN THIS CASE WE USE THE WILCOXIN SIGNED RANK TEST SINCE THE ABOVE CHOSEN SMALL SAMPLE FROM THE POPULATION IS NOT NORMAL AND HAS OUTLIERS.

RANK TABLE:

SAMPLE VALUES(X) X-mean RANK

```
179                 -281    -2
179                 -281    -2
179                 -281    -2
269                 -191    -1
1299                 839     3
```

Therefore=>Test statistic(S+)=3

STEP4:CALCULATING THE P-VALUE OF THE TEST STATISTIC: P-value=0.1562[Obtained from the table]

STEP5:FORMING THE CONCLUSION ON THE BASIS OF THE P-VALUE OBTAINED IN THE TEST-> =>Therefore,the Null Hypothesis(H0) is plausible and so is H1. Therefore,the mean price of the camera model may be less than 460.

HYPOTHESIS 3: It has been observed that the mean low resolution value of a camera model is greater than 1871.Discuss the validity of this statement.The standard deviation of the low resolution value is 719.39. WE TEST THE ABOVE HYPOTHESIS BASED ON THE EVIDENCE TAKEN FROM THE BELOW SMALL SAMPLE FROM THE SAME POPULATION: SAMPLE:

```
SAMPLE Size(n)=5
SAMPLE VALUES={640,640,0,640,640}
SAMPLE MEAN(X_bar)=886.25
```

STEP1:FORMUALTING THE NULL HYPOTHESIS AND ALTERNATE HYPOTHESIS-> Null Hypothesis: The mean low resolution value of the camera is less than or equal to 1871. EQ->H0:mean<=1871

Alternate Hypothesis: The mean low resolution value of the camera is greater than 1781. EQ->H1:mean>1781

STEP3:CALCULATING THE VALUE OF THE TEST STATISTIC-> NOTE:IN THIS CASE WE USE THE WILCOXIN SIGNED RANK TEST SINCE THE ABOVE CHOSEN SMALL SAMPLE FROM THE POPULATION IS NOT NORMAL AND HAS OUTLIERS.

RANK TABLE: SAMPLE VALUES(X) X-mean RANK

```
640             -246.25  -1
640             -246.25  -1
0               -886.25  -2
640             -246.25  -1
640             -246.25  -1
```

Therefore=>Test statistic(S-)=1+1+2+1+1=6 S-=6

STEP4:FORMING THE CONCLUSION ON THE BASIS OF THE P-VALUE OBTAINED IN THE TEST-> Since the sum value of 6 is not available in the table for the Wilcoxin signed rank test,we take the nearest value to (S-) in the table which is 3. Therefore,P-value(S-=6)>0.1562[Obtained from the table]

STEP5:FORMING THE CONCLUSION ON THE BASIS OF THE P-VALUE OBTAINED IN THE TEST-> Therefore,the Null Hypothesis(H0) is plausible and so is H1. The mean value of the low resolution of the camera may be greater than 1871.

6] CORRELATION

In [30]:

```
dataset.corr(method ='pearson')
```

Out[30]:

	Release date	Max resolution	Low resolution	Effective pixels	Zoom wide (W)	Zoom tele (T)	Normal focus range	M
Release date	1.000000	0.793229	0.788348	0.727437	-0.158559	0.219224	-0.036885	-0.28
Max resolution	0.793229	1.000000	0.903539	0.897375	-0.257676	0.207330	-0.016576	-0.28
Low resolution	0.788348	0.903539	1.000000	0.854653	-0.207279	0.228487	0.003808	-0.29
Effective pixels	0.727437	0.897375	0.854653	1.000000	-0.185677	0.198812	-0.039158	-0.27
Zoom wide (W)	-0.158559	-0.257676	-0.207279	-0.185677	1.000000	-0.020743	0.063771	0.07
Zoom tele (T)	0.219224	0.207330	0.228487	0.198812	-0.020743	1.000000	-0.095580	-0.23
Normal focus range	-0.036885	-0.016576	0.003808	-0.039158	0.063771	-0.095580	1.000000	0.26
Macro focus range	-0.280573	-0.286491	-0.293249	-0.277810	0.073385	-0.233013	0.261964	1.00
Storage included	0.228688	0.238868	0.243829	0.206505	-0.101012	0.062244	0.094715	-0.1
Weight (inc. batteries)	-0.277609	0.094312	0.054294	0.083540	-0.138453	0.225876	-0.030688	-0.02
Dimensions	-0.368209	-0.088711	-0.117810	-0.048487	-0.075149	0.087615	-0.023818	0.04
Price	-0.023249	0.182783	0.133804	0.199242	-0.096697	-0.010971	-0.058069	-0.0

Correlation of a variable with itself is 1 If correlation is higher, the quantities are highly related.