In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
wcat=pd.read_csv("C:/Users/USER/Desktop/Datasets/wc.at.csv")
wcat
```

Out[2]:

	Waist	AT
0	74.75	25.72
1	72.60	25.89
2	81.80	42.60
3	83.95	42.80
4	74.65	29.84
...	...	...
104	100.10	124.00
105	93.30	62.20
106	101.80	133.00
107	107.90	208.00
108	108.50	208.00

109 rows × 2 columns

In [3]:
```
wcat.describe()
```

Out[3]:

	Waist	AT
count	109.000000	109.000000
mean	91.901835	101.894037
std	13.559116	57.294763
min	63.500000	11.440000
25%	80.000000	50.880000
50%	90.800000	96.540000
75%	104.000000	137.000000
max	121.000000	253.000000

In [4]:
```
wcat.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 109 entries, 0 to 108
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Waist   109 non-null    float64
 1   AT      109 non-null    float64
dtypes: float64(2)
memory usage: 1.8 KB
```

In [5]:
```
wcat.median()
```

Out[5]:
```
Waist    90.80
AT       96.54
dtype: float64
```

In [6]:
```
wcat.mode()
```

Out[6]:

	Waist	AT
0	94.5	121.0
1	106.0	123.0
2	108.5	NaN

In [7]:
```
wcat.isnull().sum()
```

Out[7]:
```
Waist    0
AT       0
dtype: int64
```

In [8]:
```
wcat.std()
```

Out[8]:
```
Waist    13.559116
AT       57.294763
dtype: float64
```

In [9]:
```
wcat.var()
```

Out[9]:
```
Waist     183.849626
AT       3282.689835
dtype: float64
```

In [10]:
```
range_AT=wcat["AT"].max()-wcat["AT"].min()
range_AT
```

Out[10]:
```
241.56
```

In [11]:
```
range_Waist=wcat["Waist"].max()-wcat["Waist"].min()
range_Waist
```

Out[11]:
```
57.5
```

In [12]:
```
wcat.kurtosis()
```

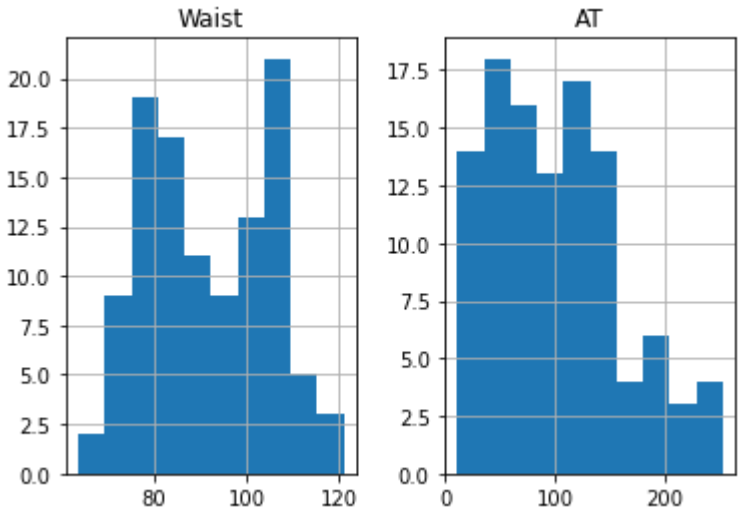
Out[12]:
```
Waist   -1.102667
AT      -0.285576
dtype: float64
```

In [13]:
```
wcat.skew()
```

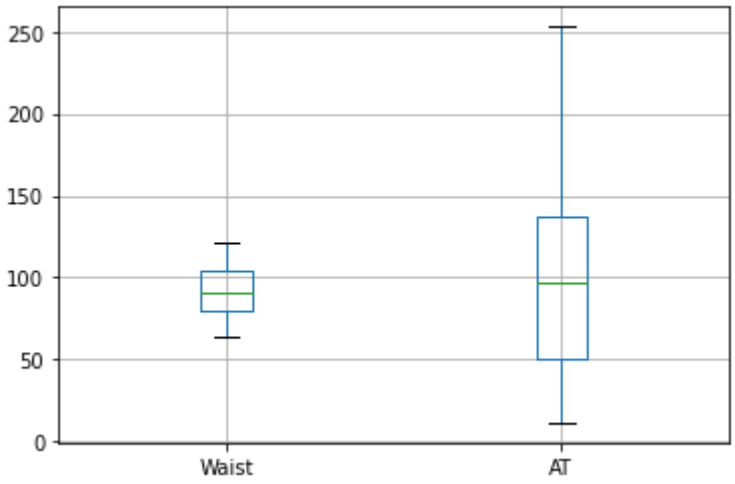
Out[13]:
```
Waist    0.134056
AT       0.584869
dtype: float64
```

In [14]:
```
wcat.hist()
```

Out[14]:
```
array([[<AxesSubplot:title={'center':'Waist'}>,
        <AxesSubplot:title={'center':'AT'}>]], dtype=object)
```

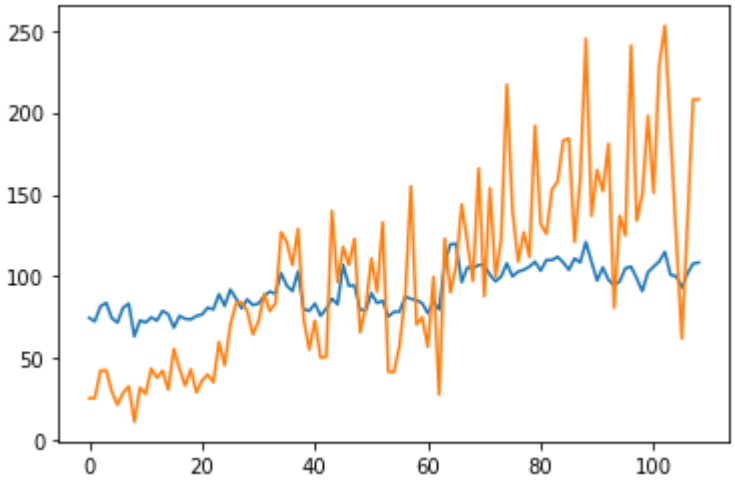


In [15]:
```
boxpot=wcat.boxplot()
```



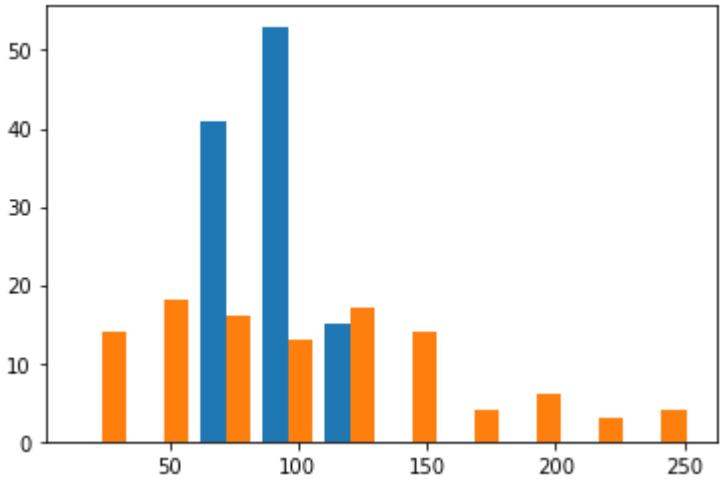
In [16]:
```
plt.plot(wcat)
```

Out[16]:
```
[<matplotlib.lines.Line2D at 0x233cad3e460>,
 <matplotlib.lines.Line2D at 0x233cad3e490>]
```



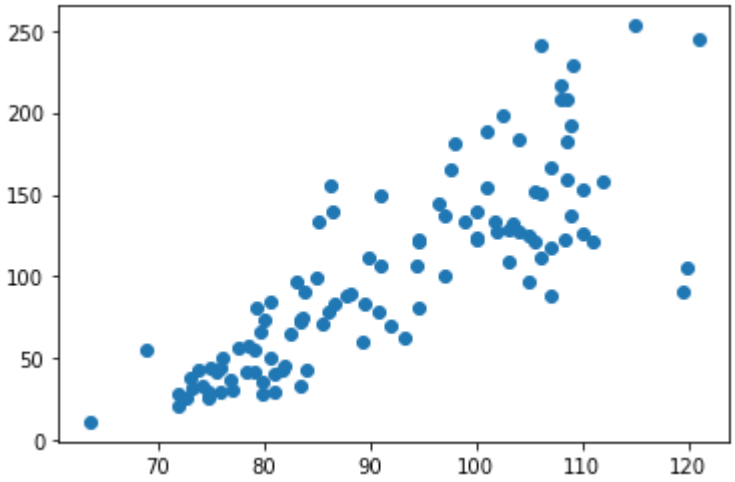
In [17]:
```
plt.hist(wcat)
```

Out[17]:
```
(array([[ 0.,  0., 41., 53., 15.,  0.,  0.,  0.,  0.,  0.],
        [14., 18., 16., 13., 17., 14.,  4.,  6.,  3.,  4.]]),
 array([ 11.44 ,  35.596,  59.752,  83.908, 108.064, 132.22 , 156.376,
        180.532, 204.688, 228.844, 253.   ]),
 <a list of 2 BarContainer objects>)
```



In [18]:
```
plt.scatter(x=wcat["Waist"],y=wcat["AT"])
```

Out[18]:
```
<matplotlib.collections.PathCollection at 0x233cae2eeb0>
```



In [19]:
```
np.corrcoef(x=wcat["Waist"],y=wcat['AT'])
```

Out[19]:
```
array([[1.        , 0.81855781],
       [0.81855781, 1.        ]])
```

In [20]:
```
import statsmodels.formula.api as smf
mdl=smf.ols('AT~Waist',data=wcat).fit()
mdl.params
```

Out[20]:
```
Intercept   -215.981488
Waist          3.458859
dtype: float64
```

In [21]:
```
mdl.summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	AT	**R-squared:**	0.670
Model:	OLS	**Adj. R-squared:**	0.667
Method:	Least Squares	**F-statistic:**	217.3
Date:	Mon, 28 Feb 2022	**Prob (F-statistic):**	1.62e-27
Time:	14:52:36	**Log-Likelihood:**	-534.99
No. Observations:	109	**AIC:**	1074.
Df Residuals:	107	**BIC:**	1079.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-215.9815	21.796	-9.909	0.000	-259.190	-172.773
Waist	3.4589	0.235	14.740	0.000	2.994	3.924

Omnibus:	3.960	**Durbin-Watson:**	1.560
Prob(Omnibus):	0.138	**Jarque-Bera (JB):**	4.596

Skew:	0.104	**Prob(JB):**	0.100
Kurtosis:	3.984	**Cond. No.**	639.

Notes:

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

In [22]:
```
mdl.conf_int(0.05)
```

Out[22]:

	0	1
Intercept	-259.190053	-172.772923
Waist	2.993689	3.924030

In [23]:
```
mdl_1 = smf.ols('AT~np.log(Waist)',data = wcat).fit()
mdl_1.params
```

Out[23]:
```
Intercept       -1328.341989
np.log(Waist)     317.135564
dtype: float64
```

In [24]:
```
mdl_1.summary()
```

Out[24]:

OLS Regression Results

Dep. Variable:	AT	**R-squared:**	0.675
Model:	OLS	**Adj. R-squared:**	0.672
Method:	Least Squares	**F-statistic:**	222.6
Date:	Mon, 28 Feb 2022	**Prob (F-statistic):**	6.80e-28
Time:	14:52:45	**Log-Likelihood:**	-534.11
No. Observations:	109	**AIC:**	1072.
Df Residuals:	107	**BIC:**	1078.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1328.3420	95.923	-13.848	0.000	-1518.498	-1138.186
np.log(Waist)	317.1356	21.258	14.918	0.000	274.994	359.277

Omnibus:	3.317	**Durbin-Watson:**	1.599
Prob(Omnibus):	0.190	**Jarque-Bera (JB):**	2.908
Skew:	0.235	**Prob(JB):**	0.234

Kurtosis: 3.647 **Cond. No.** 145.

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

In [25]:
```
print(mdl_1.conf_int(0.05))
```

```
                          0           1
Intercept     -1518.498012 -1138.185966
np.log(Waist)   274.993649   359.277480
```

In [26]:
```
x=wcat.iloc[:,:1]
x
```

Out[26]:

	Waist
0	74.75
1	72.60
2	81.80
3	83.95
4	74.65
...	...
104	100.10
105	93.30
106	101.80
107	107.90
108	108.50

109 rows × 1 columns

In [27]:
```
y = wcat.iloc[:,1:2]
y
```

Out[27]:

	AT
0	25.72
1	25.89
2	42.60
3	42.80
4	29.84
...	...

	AT
104	124.00
105	62.20
106	133.00
107	208.00
108	208.00

109 rows × 1 columns

In [28]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.02,random_state = 1)
```

In [ ]:
```
from sklearn.preprocessing import MinMaxScaler
scale=MinMaxScaler()
S_x=scale.fit(x)
S_x=scale.transform(x)

S_x=pd.DataFrame(S_x, columns=names)
S_x
```

In [54]:
```
from sklearn.linear_model import LinearRegression
from sklearn import metrics
LR= LinearRegression()
LR.fit(x_train,y_train)
```

Out[54]:
```
LinearRegression()
```

In [55]:
```
y_pred = LR.predict(x_test)
y_pred
```

Out[55]:
```
array([[149.23424574],
       [120.10667673],
       [121.82006314]])
```

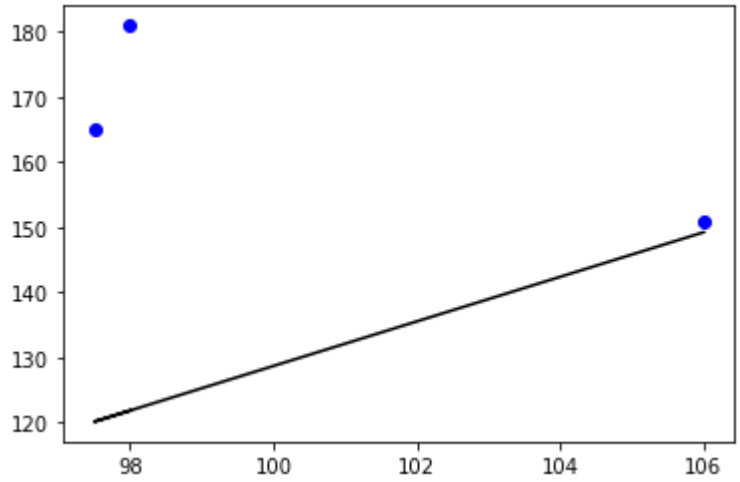
In [49]:
```
print('mean_square_error',metrics.mean_squared_error(ytest,ypred))
```

```
mean_square_error 1840.2644296349736
```

In [50]:
```
print('rmse',np.sqrt(metrics.mean_squared_error(ytest,ypred)))
```

```
rmse 42.89830334214832
```

In [64]:
```
plt.scatter(x_test, y_test, color='b')
plt.plot(x_test, y_pred, color='k')
plt.show()
```



In [65]:
```
N_wcat=wcat.iloc[0:50,:]
```

In [66]:
```
N_wcat
```

Out[66]:

	Waist	AT
0	74.75	25.72
1	72.60	25.89
2	81.80	42.60
3	83.95	42.80
4	74.65	29.84
5	71.85	21.68
6	80.90	29.08
7	83.40	32.98
8	63.50	11.44
9	73.20	32.22
10	71.90	28.32
11	75.00	43.86
12	73.10	38.21
13	79.00	42.48
14	77.00	30.96
15	68.85	55.78
16	75.95	43.78
17	74.15	33.41
18	73.80	43.35
19	75.90	29.31

	Waist	AT
20	76.85	36.60
21	80.90	40.25
22	79.90	35.43
23	89.20	60.09
24	82.00	45.84
25	92.00	70.40
26	86.60	83.45
27	80.50	84.30
28	86.00	78.89
29	82.50	64.75
30	83.50	72.56
31	88.10	89.31
32	90.80	78.94
33	89.40	83.55
34	102.00	127.00
35	94.50	121.00
36	91.00	107.00
37	103.00	129.00
38	80.00	74.02
39	79.00	55.48
40	83.50	73.13
41	76.00	50.50
42	80.50	50.88
43	86.50	140.00
44	83.00	96.54
45	107.10	118.00
46	94.30	107.00
47	94.50	123.00
48	79.70	65.92
49	79.30	81.29

In [67]:

```
input=np.array(N_wcat["Waist"]).reshape(-1,1)
output=np.array(N_wcat["AT"]).reshape(-1,1)
```

In [68]:
```
X_train, X_test, Y_train, Y_test=train_test_split(input, output, test_size=0.25)
```

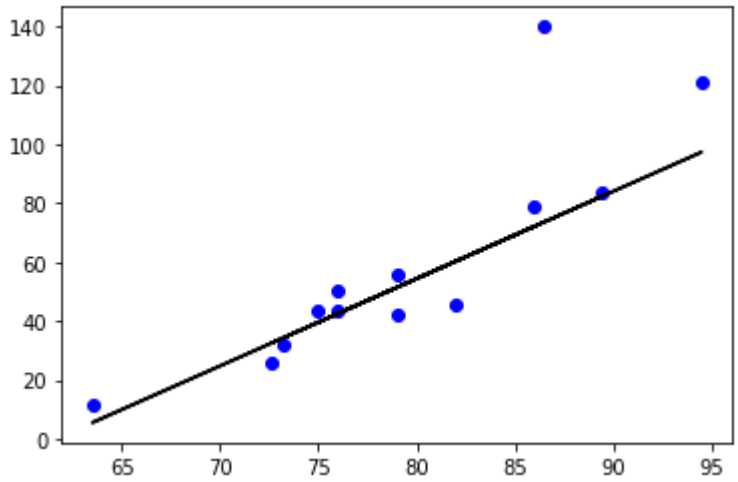
In [71]:
```
regr=LinearRegression()
regr.fit(X_train, Y_train)
print(regr.score(X_test, Y_test))
```

```
0.6673727245502035
```

In [73]:
```
Y_pred = regr.predict(X_test)
Y_pred
```

Out[73]:
```
array([[34.23556367],
       [51.43187331],
       [39.57234942],
       [82.26663542],
       [51.43187331],
       [97.38752839],
       [60.32651623],
       [42.53723039],
       [42.38898634],
       [ 5.47621823],
       [72.18604012],
       [32.45663508],
       [73.6684806 ]])
```

In [74]:
```
plt.scatter(X_test, Y_test, color='b')
plt.plot(X_test, Y_pred, color='k')
plt.show()
```



In [ ]:
```
#done :)
```