Importing Required Libraries

In [18]:

```
import pandas as pd
import seaborn as sns
from matplotlib import pyplot as plt
import statsmodels.formula.api as smf
import warnings
warnings.filterwarnings('ignore')
```

Importing Data

In [3]:

```
df = pd.read_csv("wc-at.csv")
df
```

Out[3]:

	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

Data Understanding

In [4]:

```
df.head()
```

Out[4]:

	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

In [5]:

```
df.shape
```

Out[5]:

```
(109, 2)
```

In [6]:

```
df.isna().sum()
```

Out[6]:

```
Waist    0
AT       0
dtype: int64
```

In [7]:

```
df.describe()
```

Out[7]:

	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 [8]:

```
df.dtypes
```

Out[8]:

```
Waist    float64
AT       float64
dtype: object
```

In [9]:

```
df.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 [10]:

```
df.corr()
```

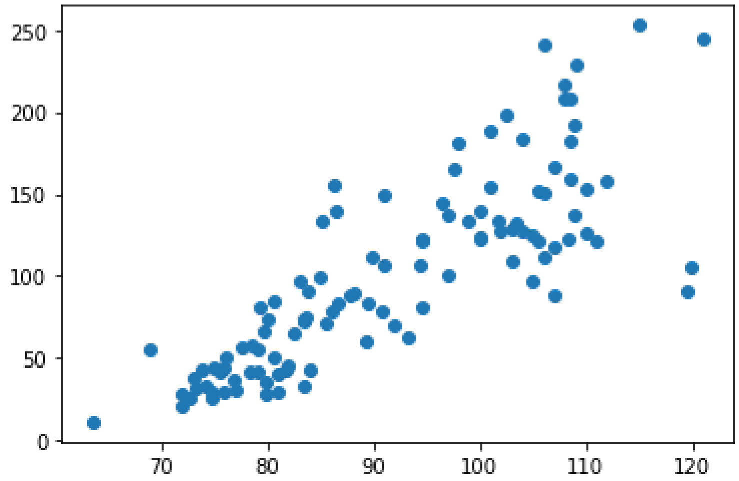
Out[10]:

	Waist	AT
Waist	1.000000	0.818558
AT	0.818558	1.000000

Checking the wheather the given assumption are correct or not

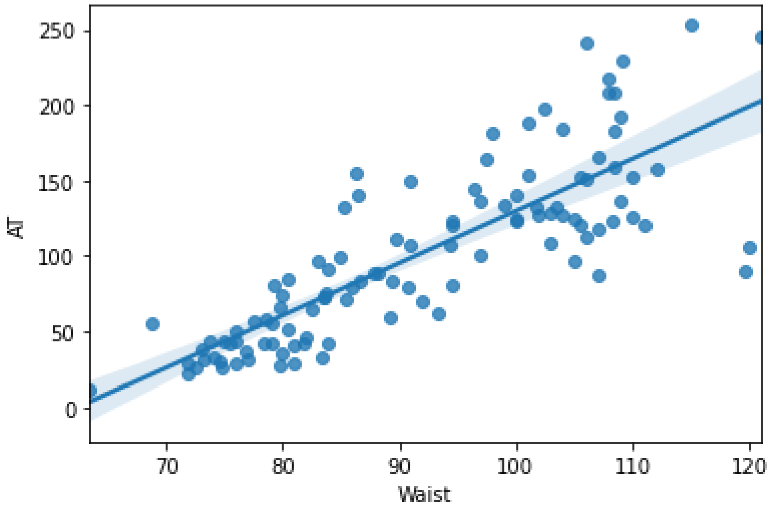
In [11]:

```
plt.scatter(x = 'Waist', y = 'AT',data = df)
plt.show()
```



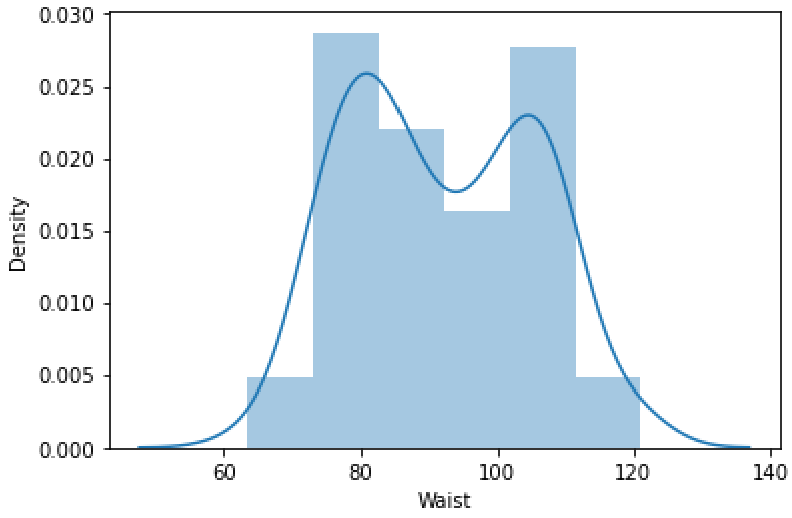
In [13]:

```
sns.regplot(x = 'Waist' , y = 'AT' , data = df)
plt.show()
```



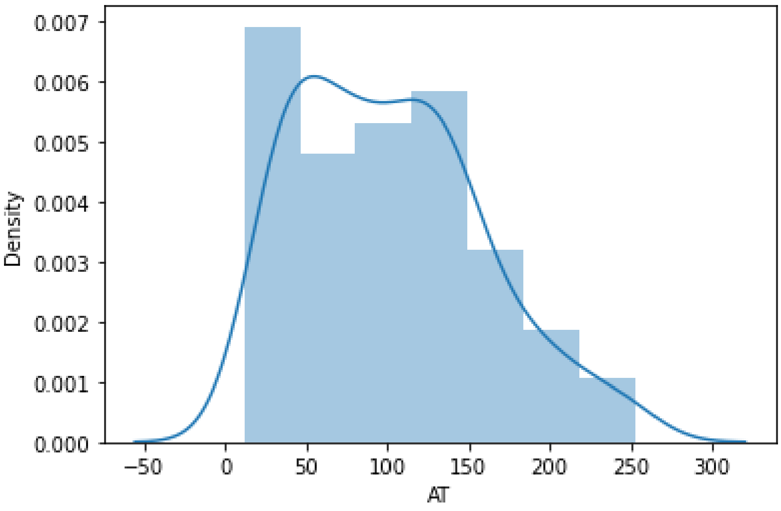
In [19]:

```
sns.distplot(df['Waist'])
plt.show()
```



In [20]:

```
sns.distplot(df['AT'])
plt.show()
```



Model Building

In [24]:

```
linear_model = smf.ols(formula = 'Waist~AT' , data = df).fit()
linear_model
```

Out[24]:

```
<statsmodels.regression.linear_model.RegressionResultsWrapper at 0x1c95b1687c0>
```

Model Testing

In [26]:

```
linear_model.params
```

Out[26]:

```
Intercept    72.163315
AT            0.193716
dtype: float64
```

In [27]:

```
linear_model.tvalues,linear_model.pvalues
```

Out[27]:

```
(Intercept    47.025363
 AT           14.740376
 dtype: float64,
 Intercept    2.697170e-73
 AT           1.618607e-27
 dtype: float64)
```

In [28]:

```
linear_model.rsquared,linear_model.rsquared_adj
```

Out[28]:

```
(0.6700368930528431, 0.6669531256981969)
```

Model Prediction

In [30]:

```
##sample calculation
##y = mx+c
df = (72.163315 + 0.193716)*(5)
df
```

Out[30]:

```
361.785155
```

In [34]:

```
##machine prediction
pred_data = {'AT':[10,20,30,40]}
pred_data
```

Out[34]:

```
{'AT': [10, 20, 30, 40]}
```

In [35]:

```
new_data = pd.DataFrame(pred_data)
new_data
```

Out[35]:

	AT
0	10
1	20
2	30
3	40

In [36]:

```
linear_model.predict(new_data)
```

Out[36]:

```
0    74.100477
1    76.037638
2    77.974800
3    79.911961
dtype: float64
```

In [ ]: