Calculate Mean, Median, Mode, Variance, Standard Deviation, Range & comment about the values / draw inferences, for the given dataset

```
1.For Points,Score,Weigh>
2.Find Mean, Median, Mode, Variance, Standard Deviation,
and Range and also Comment about the values/ Draw some inferences.
```

In [9]:

```
cars=pd.read_csv('Q7.csv')
cars
```

Out[9]:

	Unnamed: 0	Points	Score	Weigh
0	Mazda RX4	3.90	2.620	16.46
1	Mazda RX4 Wag	3.90	2.875	17.02
2	Datsun 710	3.85	2.320	18.61
3	Hornet 4 Drive	3.08	3.215	19.44
4	Hornet Sportabout	3.15	3.440	17.02
5	Valiant	2.76	3.460	20.22
6	Duster 360	3.21	3.570	15.84
7	Merc 240D	3.69	3.190	20.00
8	Merc 230	3.92	3.150	22.90
9	Merc 280	3.92	3.440	18.30
10	Merc 280C	3.92	3.440	18.90
11	Merc 450SE	3.07	4.070	17.40
12	Merc 450SL	3.07	3.730	17.60
13	Merc 450SLC	3.07	3.780	18.00
14	Cadillac Fleetwood	2.93	5.250	17.98
15	Lincoln Continental	3.00	5.424	17.82
16	Chrysler Imperial	3.23	5.345	17.42
17	Fiat 128	4.08	2.200	19.47
18	Honda Civic	4.93	1.615	18.52
19	Toyota Corolla	4.22	1.835	19.90
20	Toyota Corona	3.70	2.465	20.01
21	Dodge Challenger	2.76	3.520	16.87
22	AMC Javelin	3.15	3.435	17.30
23	Camaro Z28	3.73	3.840	15.41
24	Pontiac Firebird	3.08	3.845	17.05
25	Fiat X1-9	4.08	1.935	18.90
26	Porsche 914-2	4.43	2.140	16.70
27	Lotus Europa	3.77	1.513	16.90
28	Ford Pantera L	4.22	3.170	14.50
29	Ferrari Dino	3.62	2.770	15.50
30	Maserati Bora	3.54	3.570	14.60
31	Volvo 142E	4.11	2.780	18.60

In [10]:

```
cars.describe()
```

Out[10]:

	Points	Score	Weigh
count	32.000000	32.000000	32.000000
mean	3.596563	3.217250	17.848750
std	0.534679	0.978457	1.786943
min	2.760000	1.513000	14.500000
25%	3.080000	2.581250	16.892500
50%	3.695000	3.325000	17.710000
75%	3.920000	3.610000	18.900000
max	4.930000	5.424000	22.900000

In [11]:

```
cars.mean()
```

Out[11]:

```
Points     3.596563
Score      3.217250
Weigh     17.848750
dtype: float64
```

In [16]:

```
cars.Points.mode()
```

Out[16]:

```
0    3.07
1    3.92
Name: Points, dtype: float64
```

In [18]:

```
cars.Score.mode()
```

Out[18]:

```
0    3.44
Name: Score, dtype: float64
```

In [19]:

```
cars.Weigh.mode()
```

Out[19]:

```
0    17.02
1    18.90
Name: Weigh, dtype: float64
```

In [20]:

```
cars.var()
```

Out[20]:

```
Points    0.285881
Score     0.957379
Weigh     3.193166
dtype: float64
```

In [24]:

```
Points_range=cars.Points.max()-cars.Points.min()
Points_range
```

Out[24]:

```
2.17
```

In [25]:

```
Score_range=cars.Score.max()-cars.Score.min()
Score_range
```

Out[25]:

```
3.9110000000000005
```

In [28]:

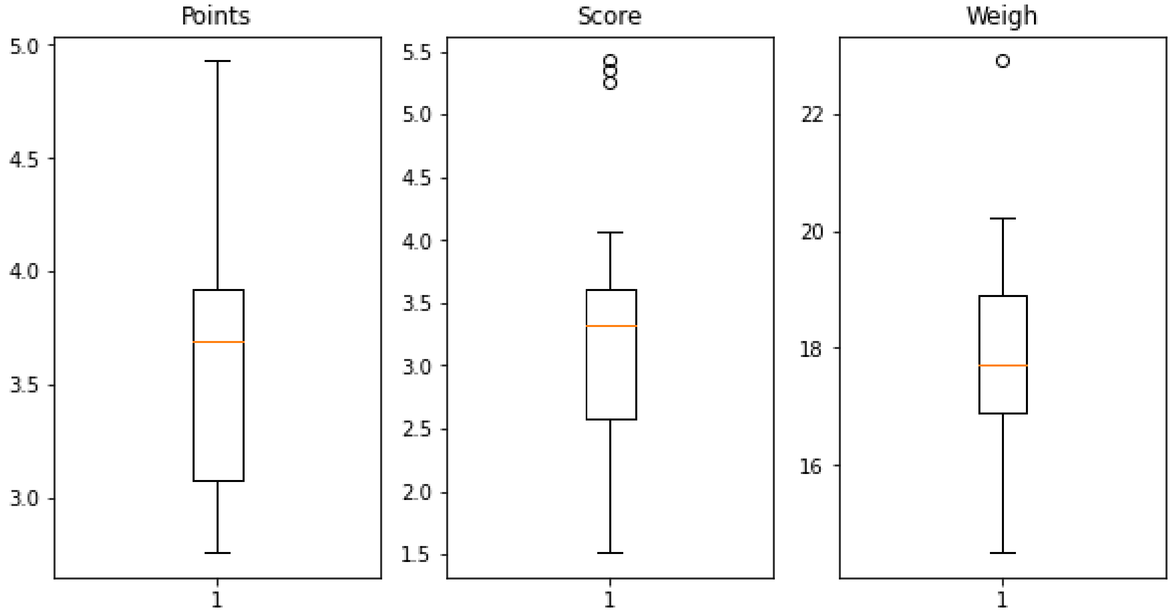
```
Weigh_range=cars.Weigh.max()-cars.Weigh.min()
Weigh_range
```

Out[28]:

```
8.399999999999999
```

In [44]:

```
f,ax=plt.subplots(figsize=(10,5))
plt.subplot(1,3,1)
plt.boxplot(cars.Points)
plt.title('Points')
plt.subplot(1,3,2)
plt.boxplot(cars.Score)
plt.title('Score')
plt.subplot(1,3,3)
plt.boxplot(cars.Weigh)
plt.title('Weigh')
plt.show()
```



In [ ]:

```
#Inferences:
a) For Points dataset:
    1) we see that data is concentrated aroound Median
    2) There are no outliars
    3) The distribution is Right skewed

 b) For Score dataset:
        1) The data is concentrated around Median
        2) There are 3 Outliars:
        3) The distribution is Left skewed

 c) For Weigh dataset:
      1) The data is concentrated around Median
      2) There is 1 Outliar:
      3) The distribution is Left skewed
```

Calculate Skewness, Kurtosis & draw inferences on the following data

Cars speed and distance

In [46]:

```
data=pd.read_csv('Q9_a.csv')
data
```

Out[46]:

	Index	speed	dist
0	1	4	2
1	2	4	10
2	3	7	4
3	4	7	22
4	5	8	16
5	6	9	10
6	7	10	18
7	8	10	26
8	9	10	34
9	10	11	17
10	11	11	28
11	12	12	14
12	13	12	20
13	14	12	24
14	15	12	28
15	16	13	26
16	17	13	34
17	18	13	34
18	19	13	46
19	20	14	26
20	21	14	36
21	22	14	60
22	23	14	80
23	24	15	20
24	25	15	26
25	26	15	54
26	27	16	32
27	28	16	40
28	29	17	32
29	30	17	40
30	31	17	50
31	32	18	42
32	33	18	56

	Index	speed	dist
33	34	18	76
34	35	18	84
35	36	19	36
36	37	19	46
37	38	19	68
38	39	20	32
39	40	20	48
40	41	20	52
41	42	20	56
42	43	20	64
43	44	22	66
44	45	23	54
45	46	24	70
46	47	24	92
47	48	24	93
48	49	24	120
49	50	25	85

In [49]:

```
#Skewness
data.skew()
```

Out[49]:

```
Index    0.000000
speed   -0.117510
dist     0.806895
dtype: float64
```

In [ ]:

```
#Inference:
    1.As we see Speed distribution is left skewed (negative skewness)
    2.As we see Distance distribution is right skewed (positive
```

In [48]:

```
#Kurtosis
data.kurtosis()
```

Out[48]:

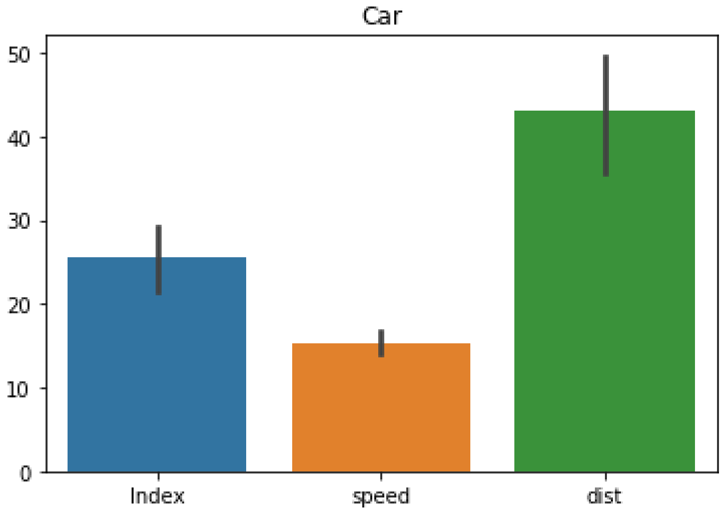
```
Index   -1.200000
speed   -0.508994
dist     0.405053
dtype: float64
```

In [ ]:

```
# Inference:
    1. Speed distribution is flatter than normal distribution which is negative kurtosis.
    2. Distance distributin is peaked than normal distribution which is positive kurtosis.
```

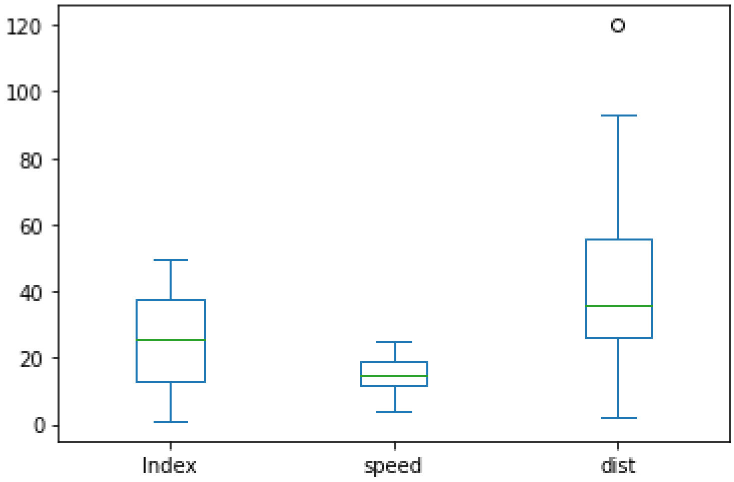
In [51]:

```
import seaborn as sns
sns.barplot(data=data)
plt.title('Car')
plt.show()
```



In [54]:

```
data.plot.box()
plt.show()
```



Suppose we want to estimate the average weight of an adult male in Mexico. We draw a random sample of 2,000 men from a population of 3,000,000 men and weigh them. We find that the average person in our sample weighs 200 pounds, and the standard deviation

# of the sample is 30 pounds. Calculate 94%,98%,96% confidence interval?

In [56]:

```
from scipy import stats
from scipy.stats import norm
```

In [58]:

```
#for 94% Confidence Interval
stats.norm.interval(0.94,200,30/(2000*0.5))
```

Out[58]:

```
(198.738325292158, 201.261674707842)
```

In [59]:

```
#for 98% Confidence Interval
stats.norm.interval(0.98,200,30/(2000*0.5))
```

Out[59]:

```
(199.9302095637788, 200.0697904362212)
```

In [60]:

```
#for 96% Confidence Interval
stats.norm.interval(0.96,200,30/(2000*0.5))
```

Out[60]:

```
(199.93838753268105, 200.06161246731895)
```

Below are the scores obtained by a student in tests

34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56

1)Find mean, median, variance, standard deviation. 2)What can we say about the student marks?

In [69]:

```
df=pd.Series([34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56])
df
```

Out[69]:

```
0     34
1     36
2     36
3     38
4     38
5     39
6     39
7     40
8     40
9     41
10    41
11    41
12    41
13    42
14    42
15    45
16    49
17    56
dtype: int64
```

In [62]:

```
df.mean()
```

Out[62]:

```
41.0
```

In [63]:

```
df.mode()
```

Out[63]:

```
0    41
dtype: int64
```

In [64]:

```
df.std()
```

Out[64]:

```
5.05266382858645
```

In [65]:

```
df.var()
```

Out[65]:

```
25.529411764705884
```

In [66]:

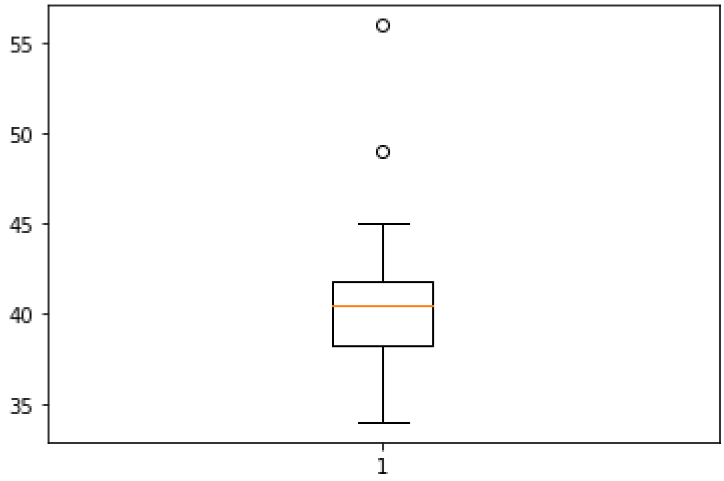
```
df.describe()
```

Out[66]:

```
count    18.000000
mean     41.000000
std       5.052664
min      34.000000
25%      38.250000
50%      40.500000
75%      41.750000
max      56.000000
dtype: float64
```

In [74]:

```
plt.boxplot(df)
plt.show()
```



In [75]:

```
#As we there are two outliers in following students marks between 4
```

Q20

In [82]:

```
cars_data=pd.read_csv('Cars.csv')
cars_data
```

Out[82]:

	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149
...	...	...	...	...	...
76	322	36.900000	50	169.598513	16.132947
77	238	19.197888	115	150.576579	37.923113
78	263	34.000000	50	151.598513	15.769625
79	295	19.833733	119	167.944460	39.423099
80	236	12.101263	107	139.840817	34.948615

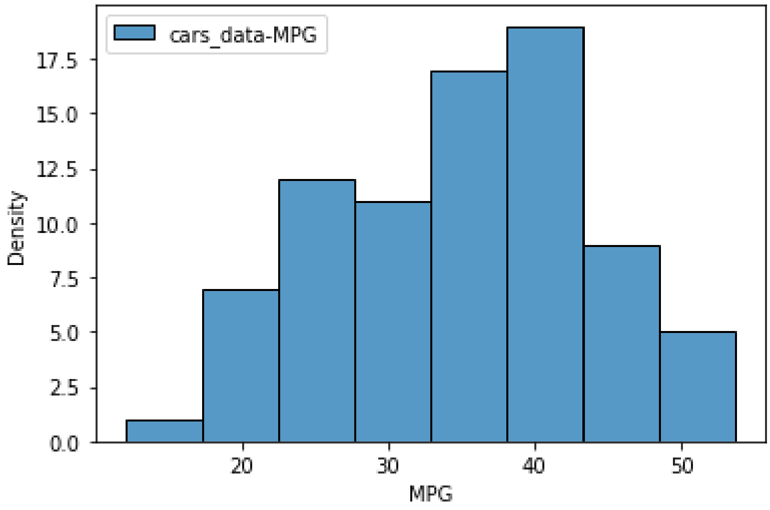
81 rows × 5 columns

Check whether the data follows normal distribution

a)Check whether the MPG of Cars follows Normal Distribution

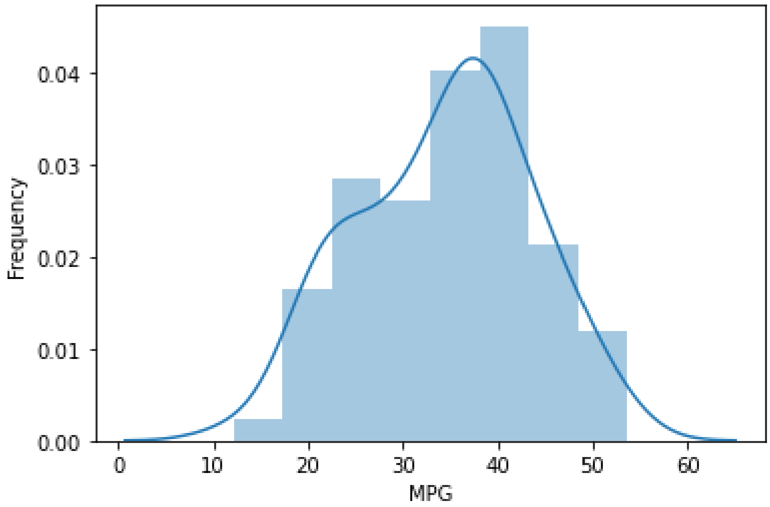
In [96]:

```
sns.histplot(cars_data.MPG,label='cars_data-MPG')
plt.xlabel('MPG')
plt.ylabel('Density')
plt.legend();
plt.show()
```



In [101]:

```
sns.distplot(cars_data.MPG,label='Cars_data-MPG')
plt.xlabel('MPG')
plt.ylabel('Frequency')
plt.show()
```



In [102]:

```
cars_data.MPG.mean()
```

Out[102]:

```
34.422075728024666
```

In [103]:

```
cars_data.MPG.mode()
```

Out[103]:

```
0    29.629936
Name: MPG, dtype: float64
```

In [104]:

```
cars_data.MPG.std()
```

Out[104]:

```
9.131444731795982
```

b)Check Whether the Adipose Tissue (AT) and Waist Circumference(Waist) from wc-at data set follows Normal Distribution

In [107]:

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

Out[107]:

	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 [109]:

```
df.tail()
```

Out[109]:

	Waist	AT
104	100.1	124.0
105	93.3	62.2
106	101.8	133.0
107	107.9	208.0
108	108.5	208.0

In [108]:

```
df.head()
```

Out[108]:

	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 [110]:

```
df.mode()
```

Out[110]:

	Waist	AT
0	94.5	121.0
1	106.0	123.0
2	108.5	NaN

In [112]:

```
df.mean()
```

Out[112]:

```
Waist     91.901835
AT       101.894037
dtype: float64
```

In [113]:

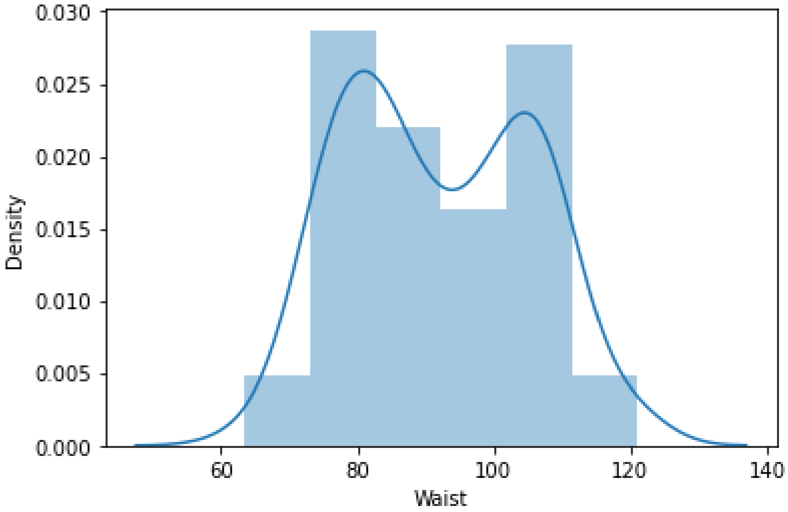
```
df.median()
```

Out[113]:

```
Waist    90.80
AT       96.54
dtype: float64
```

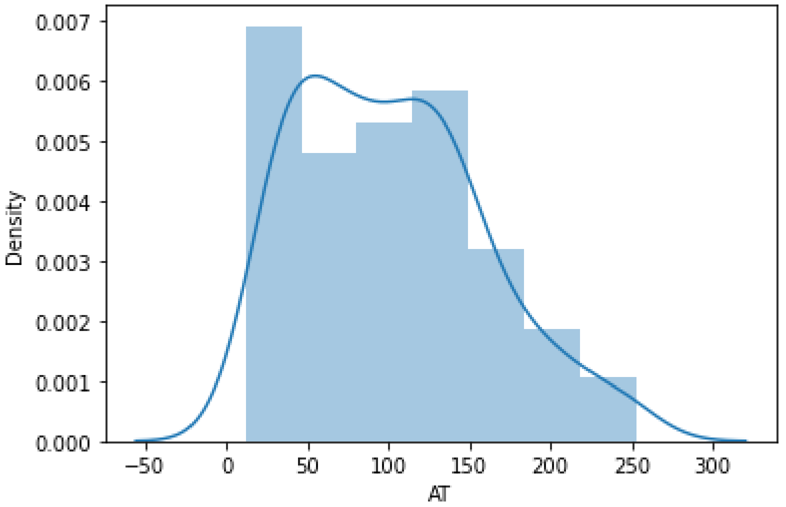
In [117]:

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



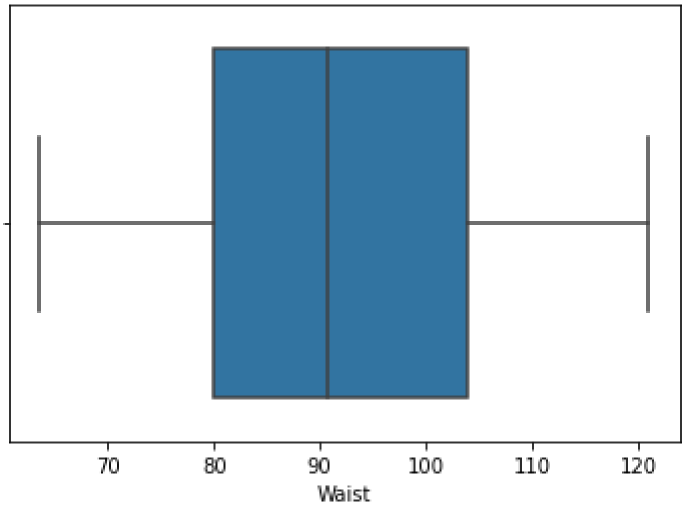
In [116]:

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



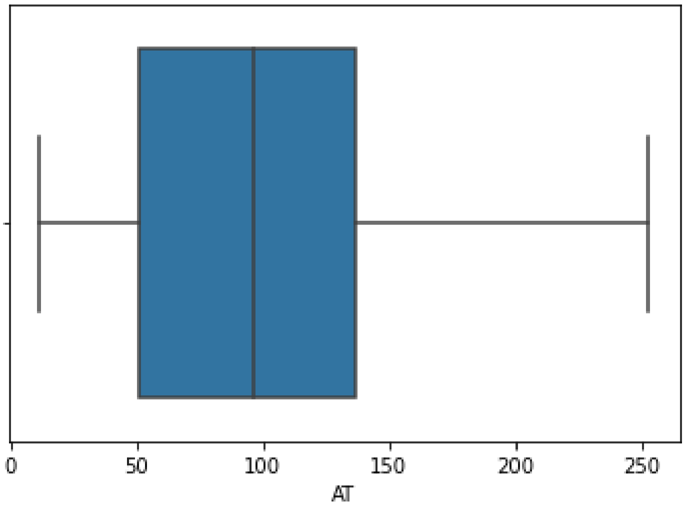
In [118]:

```
sns.boxplot(df['Waist'])
plt.show()
# mean> median, right whisker is larger than left whisker, data is positively skewed.
```



In [119]:

```
sns.boxplot(df['AT'])
plt.show()
# mean> median, both the whisker are of same lenght, median is slightly shifted towards lef
```



Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval

In [120]:

```
from scipy import stats
from scipy.stats import norm
```

In [122]:

```
# Z-score for  90% confidence interval is
stats.norm.ppf(0.95)
```

Out[122]:

```
1.6448536269514722
```

In [123]:

```
# Z-score for 94% confidence interval is
stats.norm.ppf(0.97)
```

Out[123]:

```
1.8807936081512509
```

In [124]:

```
# Z-score of 60% confidence interval is
stats.norm.ppf(0.80)
```

Out[124]:

```
0.8416212335729143
```

Calculate the t scores of 95% confidence interval, 96% confidence interval, 99% confidence interval for sample size of 25

In [148]:

```
#for 95 % confidence interval t score is
stats.t.ppf(0.975,24)
```

Out[148]:

```
2.0638985616280205
```

In [146]:

```
#for 96 % confidence interval t score is :
stats.t.ppf(0.98,24)
```

Out[146]:

```
2.1715446760080677
```

In [147]:

```
#for 99 % confidence interval t score is :
stats.t.ppf(0.99,24)
```

Out[147]:

```
2.4921594731575762
```

A Government company claims that an average light bulb lasts 270 days. A researcher randomly selects 18 bulbs for testing. The sampled bulbs last an average of 260 days, with a standard deviation of 90 days. If the CEO's claim were true, what is the probability that 18 randomly selected bulbs would have an average life of no more than 260 days

In [156]:

```
# find t-scores at x=260; t=(s_mean-P_mean)/(s_SD/sqrt(n))
t=(260-270)/(90/18**0.5)
t
```

Out[156]:

```
-0.4714045207910317
```

In [158]:

```
# p_value=1-stats.t.cdf(abs(t_scores),df=n-1)... Using cdf function.
p_value=1-stats.t.cdf(abs(-0.47140),df=17)
p_value
```

Out[158]:

```
0.32167411684460556
```

In [ ]:

In [ ]: