Q1) Identify the Data type for the Following:

Activity	Data Type
Number of beatings from Wife	Discrete
Results of rolling a dice	Discrete
Weight of a person	Continuous
Weight of Gold	Continuous
Distance between two places	Continuous
Length of a leaf	Continuous
Dog's weight	Continuous
Blue Color	Discrete
Number of kids	Discrete
Number of tickets in Indian railways	Discrete
Number of times married	Discrete
Gender (Male or Female)	Discrete

Q2) Identify the Data types, which were among the following

Nominal, Ordinal, Interval, Ratio.

Data	Data Type
Gender	Nominal
High School Class Ranking	Ordinal
Celsius Temperature	Interval
Weight	Ratio
Hair Color	Nominal
Socioeconomic Status	Ordinal
Fahrenheit Temperature	Interval
Height	Ratio
Type of living accommodation	Ordinal
Level of Agreement	Ordinal
IQ(Intelligence Scale)	Interval
Sales Figures	Ratio
Blood Group	Nominal
Time Of Day	Interval
Time on a Clock with Hands	Interval
Number of Children	Ordinal
Religious Preference	Nominal
Barometer Pressure	Interval
SAT Scores	Interval
Years of Education	Ratio

Q3) Three Coins are tossed, find the probability that two heads and one tail are obtained?

Soln: 3 coins tossed

Possible outcomes: {HHH,THH,TTH,TTT,HTT,HHT,THT,HTH}

=3/8=0.375

Q4) Two Dice are rolled, find the probability that sum is

a) Equal to 1
b) Less than or equal to 4
c) Sum is divisible by 2 and 3

Soln:

a) Sum is equal to 1

= 0

b) Less than equal to 4

= {(1,1)(2,1),(3,1),(1,2).(2,2),(1,3)}

= 6/36

= 0.17

c) Sum is divisible by 2 and 3

= {(1,5)(2,4),(3,3),(4,2).(5,2),(6,6)}

= 6/36

=0.17

Q5) A bag contains 2 red, 3 green and 2 blue balls. Two balls are drawn at random. What is the probability that none of the balls drawn is blue?

Soln:

= {R,R,G,G,G,B,B}

= Probability of first ball being not blue is 5/7

= Probability of second ball being not blue is 4/6

= 5/7 and 4/6 = 5/7*4/6

= 20/42 = 10/21

Q6) Calculate the Expected number of candies for a randomly selected child

Below are the probabilities of count of candies for children (ignoring the nature of the child-Generalized view)

CHILD	Candies count	Probability
A	1	0.015
B	4	0.20
C	3	0.65
D	5	0.005
E	6	0.01

F	2	0.120

Child A – probability of having 1 candy = 0.015.

Child B – probability of having 4 candies = 0.20

Soln:

= Expected number of candies = Sum of (X* p(x))

= Sum of (Candies count * Probability)

= 1*0.015 + 4*0.20 + 3*0.65 + 5*0.005 + 6*0.01 + 2*0.120

= 3.09

Q7) Calculate Mean, Median, Mode, Variance, Standard Deviation, Range & comment about the values / draw inferences, for the given dataset

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

Use Q7.csv file

	Points	**Score**	**Weight**
Mean	**3.596**	**3.217**	**17.84**
Median	**3.695**	**3.325**	**17.710**
Mode	**3.07 & 3.92**	**3.44**	**17.02 & 18.90**
Std	**.5346**	**.9784**	**1.786**
Vairance	**.2858**	**.9573**	**3.19**
Range	**2.17**	**3.910**	**8.399**

Inference : 1.Mean & Median approx equal therefore data is normally distributed

2. Variance is less , all data points are equal to each other

In [1]:
```
import pandas as pd
```

In [14]:
```
df=pd.read_csv('Q7.csv')
df.head()
```

Out[14]:

	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

In [10]:
```
print("Points-Mean   : ",df.Points.mean())
print("Points-Std    : ",df.Points.std())
print("Points-Mode   : ",df.Points.mode()[0])
print("Points-Var    : ",df.Points.var())
print("Points-Range  : ",df.Points.max()-df.Points.min())
```

```
Points-Mean   :  3.5965625000000006
Points-Std    :  0.5346787360709716
Points-Mode   :  3.07
Points-Var    :  0.28588135080645166
Points-Range  :  2.17
```

In [11]:
```
print("Score-Mean   : ",df.Score.mean())
print("Score-Std    : ",df.Score.std())
print("Score-Mode   : ",df.Score.mode()[0])
print("Score-Var    : ",df.Score.var())
print("Score-Range  : ",df.Score.max()-df.Score.min())
```

```
Score-Mean   :  3.2172499999999995
Score-Std    :  0.9784574429896966
Score-Mode   :  3.44
Score-Var    :  0.9573789677419354
Score-Range  :  3.9109999999999996
```

In [12]:
```
print("Weigh-Mean   : ",df.Weigh.mean())
print("Weigh-Std    : ",df.Weigh.std())
print("Weigh-Mode   : ",df.Weigh.mode()[0])
print("Weigh-Var    : ",df.Weigh.var())
print("Weigh-Range  : ",df.Weigh.max()-df.Score.min())
```

```
Weigh-Mean   :  17.848750000000003
Weigh-Std    :  1.7869432360968431
Weigh-Mode   :  17.02
Weigh-Var    :  3.193166129032258
Weigh-Range  :  21.386999999999997
```

Q8) Calculate Expected Value for the problem below

a) The weights (X) of patients at a clinic (in pounds), are
108, 110, 123, 134, 135, 145, 167, 187, 199

Assume one of the patients is chosen at random. What is the Expected Value of the Weight of that patient?

Soln:

Expected Value = 145.33(Its nothing but avg of the given data)

(108+110+123+134+135+145+167+187+199)/9= 1308/9= 145.333

Q9) Calculate Skewness, Kurtosis & draw inferences on the following data

Cars speed and distance

Use Q9_a.csv

	Skewness	Kurtosis
Car Speed	-0.1775(-ve Skew)	-1.2(-ve Kurt)
Distance	0.8068(+ve Skew)	-0.508(-ve Kurt)

In [1]:
```
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df=pd.read_csv("Q9_a.csv")
df.head()
```

Out[2]:

	Index	speed	dist
0	1	4	2
1	2	4	10
2	3	7	4
3	4	7	22
4	5	8	16

In [10]:
```
print("Car Speed Skew : ",round(df.speed.skew(),4))
print("Car Speed Kurt : ",round(df.speed.kurt(),4))
```

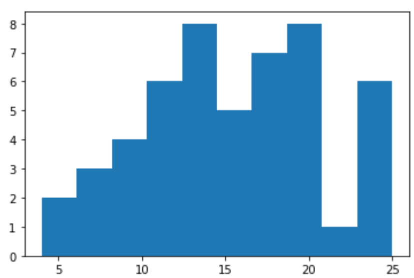
```
Car Speed Skew :  -0.1175
Car Speed Kurt :  -0.509
```

In [11]:
```
print("Distance Skew : ",round(df.dist.skew(),4))
print("Distance Kurt : ",round(df.dist.kurt(),4))
```

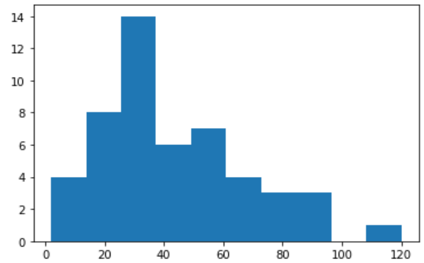
```
Distance Skew :  0.8069
Distance Kurt :  0.4051
```

In [13]:
```
plt.hist(df.speed)
```

Out[13]:
```
(array([2., 3., 4., 6., 8., 5., 7., 8., 1., 6.]),
 array([ 4. ,  6.1,  8.2, 10.3, 12.4, 14.5, 16.6, 18.7, 20.8, 22.9, 25. ]),
 <a list of 10 Patch objects>)
```



```
In [14]: plt.hist(df.dist)

Out[14]: (array([ 4.,  8., 14.,  6.,  7.,  4.,  3.,  3.,  0.,  1.]),
          array([  2. ,  13.8,  25.6,  37.4,  49.2,  61. ,  72.8,  84.6,  96.4,
                 108.2, 120. ]),
          <a list of 10 Patch objects>)
```



SP and Weight(WT)

Use Q9_b.csv

	Skewness	Kurtosis
SP	1.611(+ve Skew)	2.977(-ve kurtosis)
Weight	-.614(-ve Skew)	.950(-ve kurtosis)

```
In [1]: import pandas as pd
        from scipy import stats
        import matplotlib.pyplot as plt
        %matplotlib inline
```

```
In [2]: df=pd.read_csv('Q9_b.csv')
        df.head()
```

Out[2]:

	Unnamed: 0	SP	WT
0	1	104.185353	28.762059
1	2	105.461264	30.466833
2	3	105.461264	30.193597
3	4	113.461264	30.632114
4	5	104.461264	29.889149

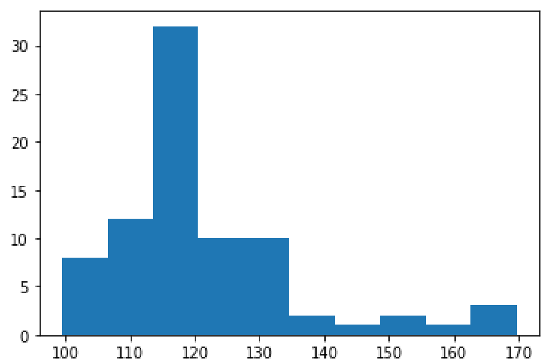
```
In [3]: print("SP Skew : ",round(df.SP.skew(),4))
        print("SP Kurt : ",round(df.SP.kurt(),4))
```

```
SP Skew :  1.6115
SP Kurt :  2.9773
```

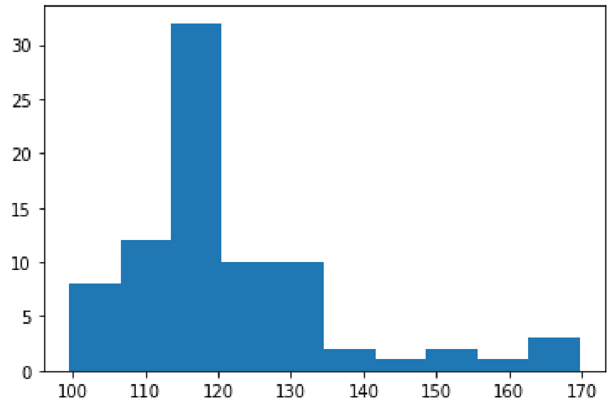
```
In [4]: print("WT Skew : ",round(df.WT.skew(),4))
        print("WT Kurt : ",round(df.WT.kurt(),4))
```

```
WT Skew :  -0.6148
WT Kurt :  0.9503
```

```
n [11]: plt.hist(df.SP)
        plt.show()
```

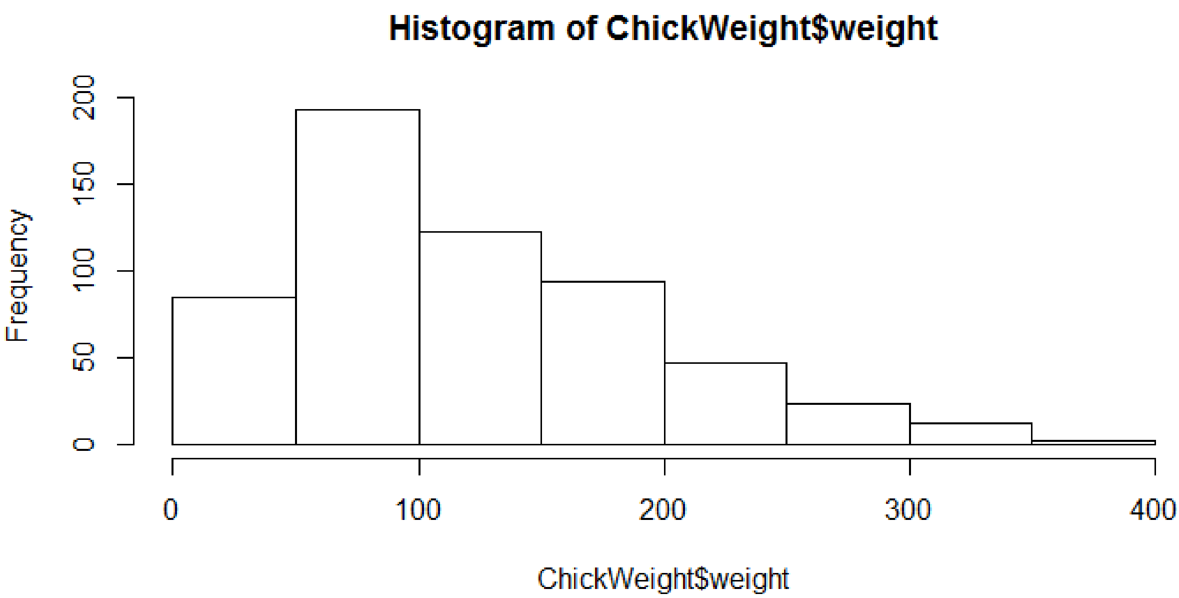


```
In [6]: plt.hist(df.SP)
        plt.show

Out[6]: <function matplotlib.pyplot.show(*args, **kw)>
```

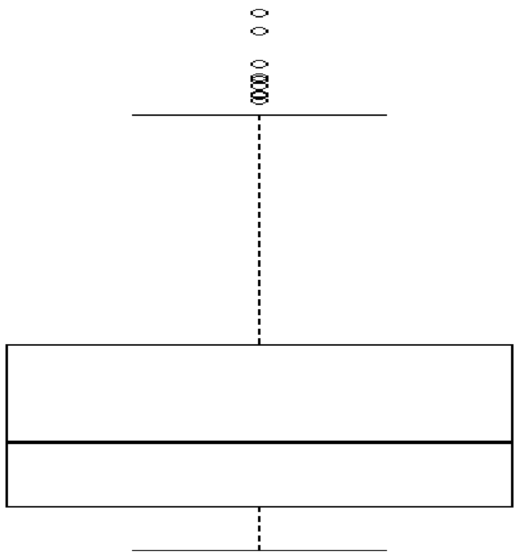



```
In [ ]:
```

Q10) Draw inferences about the following boxplot & histogram



Ans : 1.Data distribution is +ve skew 2. Most of data lies in btn 50-100



Ans : 1.Outilers prenent on upperextreme   2.there are 7 outliers on upper extreme 3. Data is +ve Skewed

Q11) 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?

Soln : stats.norm.interval(CI,sample mean,SD/sqrt(n))
Sample mean = 200
SD=30
n=2000

```
In [2]: from scipy import stats
        from scipy.stats import norm
```

From Que

- Sample mean = 200
- SD=30
- n=2000

```
n [19]: print("Average CI @ 94% :",stats.norm.interval(0.94,200,30/(2000**0.5)))
        print("Average CI @ 96% :",stats.norm.interval(0.96,200,30/(2000**0.5)))
        print("Average CI @ 98% :",stats.norm.interval(0.98,200,30/(2000**0.5)))

        Average CI @ 94% : (198.738325292158, 201.261674707842)
        Average CI @ 96% : (198.62230334813333, 201.37769665186667)
        Average CI @ 98% : (198.43943840429978, 201.56056159570022)
```

Q12) 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?

   Ans : mean = 41.0 median= 40.5 Variance =24.11 std=4.91

   Since variance is more ,score of students are not equal to each other

```
In [1]: import numpy as np

In [2]: df=[34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56]

In [10]: print("Mean     : ",np.mean(df))
         print("Median   : ",np.median(df))
         print("Variance : ",round(np.var(df),4))
         print("STD      : ",round(np.std(df),4))

         Mean     :  41.0
         Median   :  40.5
         Variance :  24.1111
         STD      :  4.9103
```

Q13) What is the nature of skewness when mean, median of data are equal?

Ans : Skewness will be zero

Data distribution is symmetry

Q14) What is the nature of skewness when mean > median ?
Ans : Data distribution is positively skewed

Q15) What is the nature of skewness when median > mean?

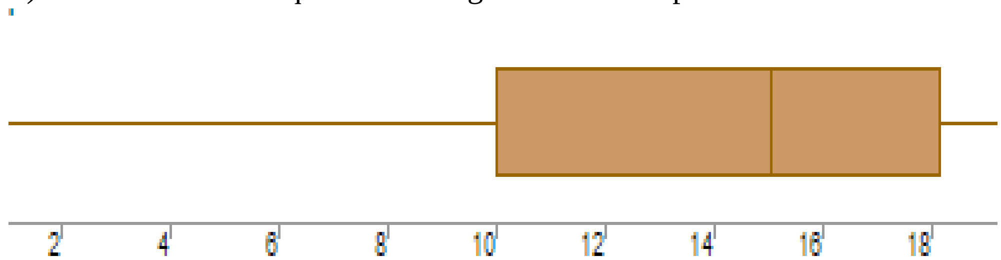
Ans : Data distribution is negatively skewed

Q16) What does positive kurtosis value indicates for a data ?
Ans: When compared to Normal Distribution graph, in positive kurtosis the distribution of data will be heavier at tails and sharp peak will be there

Q17) What does negative kurtosis value indicates for a data?

Ans : When compared to Normal Distribution graph, in positive kurtosis the distribution of data will be flat and thin tail will be there

Q18) Answer the below questions using the below boxplot visualization.



What can we say about the distribution of the data?

1. Distribution of the data is towards the tail
2. 1st 25% data lies between = 2 -10
3. 50% data lies between = 10-18
4. 2nd 25% data lies between = 18-

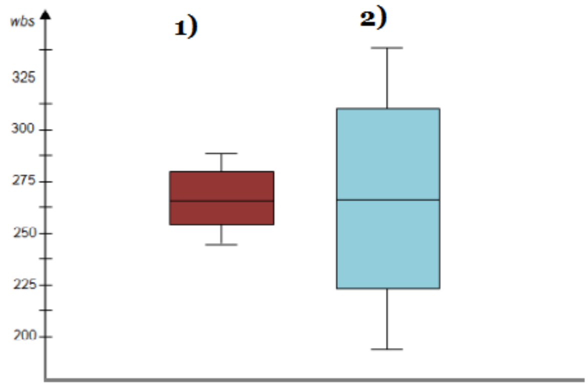
What is nature of skewness of the data?

1. Left whisker length is more therefore data is leftskewed

What will be the IQR of the data (approximately)?

Soln: Q1=10
Q2 = 14.6 approx
Q3 = 18
IQR= Q3-Q1=18-10= 8 Apporox

Q19) Comment on the below Boxplot visualizations?



Draw an Inference from the distribution of data for Boxplot 1 with respect Boxplot 2.

Ans :1. Both have similar median 2. Both are normally distributed

Q 20) Calculate probability from the given dataset for the below cases
Data _set: Cars.csv

Calculate the probability of MPG of Cars for the below cases.

MPG <- Cars$MPG

a. P(MPG>38)

1-stats.norm.cdf(38,df.MPG.mean(),df.MPG.std()) =.3475=34.75%

b. P(MPG<40)

stats.norm.cdf(40,df.MPG.mean(),df.MPG.std()) = .7293=72.93%

c. P (20<MPG<50)

stats.norm.cdf(50,df.MPG.mean(),df.MPG.std()) -
stats.norm.cdf(20,df.MPG.mean(),df.MPG.std()) = .8988=89.88%

```
In [2]: import pandas as pd
        from scipy import stats
```

```
In [3]: df=pd.read_csv('Cars.csv')
        df.head()
```

Out[3]:

	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

```
In [23]: print("P(MPG>38)     :",round(1-stats.norm.cdf(38,df.MPG.mean(),df.MPG.std()),4))
         print("P(MPG<40)     :",round(stats.norm.cdf(40,df.MPG.mean(),df.MPG.std()),4))
         print("P(20<MPG<50) :",round(stats.norm.cdf(50,df.MPG.mean(),df.MPG.std()) -stats.norm.cdf(20,df.MPG.mean(),df.MPG.std()),4))

P(MPG>38)     : 0.3476
P(MPG<40)     : 0.7293
P(20<MPG<50) : 0.8989
```

Q 21) Check whether the data follows normal distribution

a) Check whether the MPG of Cars follows Normal Distribution
Dataset: Cars.csv

Ans : Since mean&median are approximately equal thats why MPG is aprroximately Normal Distribution

```
In [2]: import pandas as pd
        import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns
```

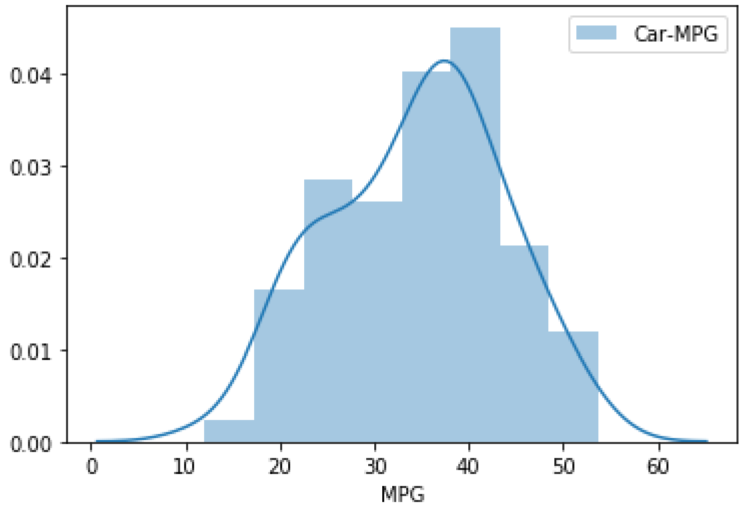
```
In [3]: df=pd.read_csv('Cars.csv')
        df.head()
```

Out[3]:

	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

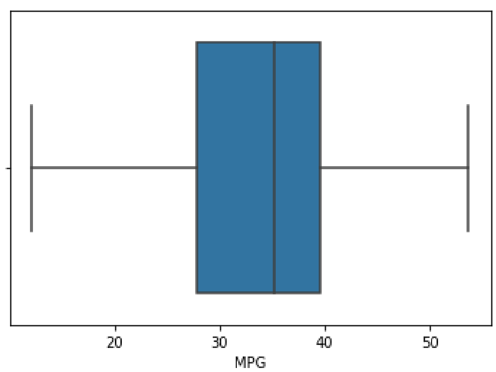
```
In [7]: print("MPG-Mean   : ",round(df.MPG.mean(),4))
        print("MPG-Median : ",round(df.MPG.median(),4))

MPG-Mean   :  34.4221
MPG-Median :  35.1527
```

In [10]:
```
sns.distplot(df.MPG, label='Car-MPG')
plt.xlabel('MPG')
plt.legend();
```



In [16]:
```
sns.boxplot(df.MPG)
```

Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x26a971e4788>



In [ ]:
```
# Since mean&median are approximately equal thats why MPG is aprroximately Normal Distribution
```

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

Ans: From above graph we infer that
AT: mean>median and right whisker is larger than left whisker , therefore AT is +ve skewed
Waist : mean approx= median and both whiskers are same length , therefore Waist is approx Normal Distribution

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

In [2]:
```
df=pd.read_csv('wc-at.csv')
df.head()
```

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

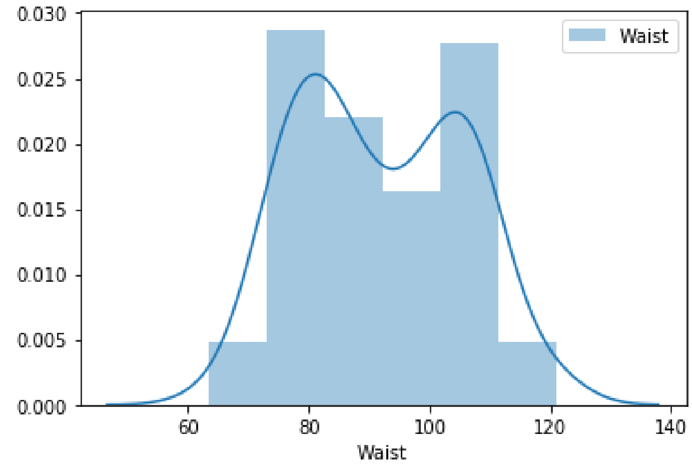
In [9]:
```
print("Waist Mean   :",round(df.Waist.mean(),2))
print("Waist Median :",df.Waist.median())
print("Waist Mode   :",df.Waist.mode()[0])
```

```
Waist Mean   : 91.9
Waist Median : 90.8
Waist Mode   : 94.5
```

In [11]:
```
print("AT Mean   :",round(df.AT.mean(),2))
print("AT Median :",df.AT.median())
print("AT Mode   :",df.AT.mode()[0])
```

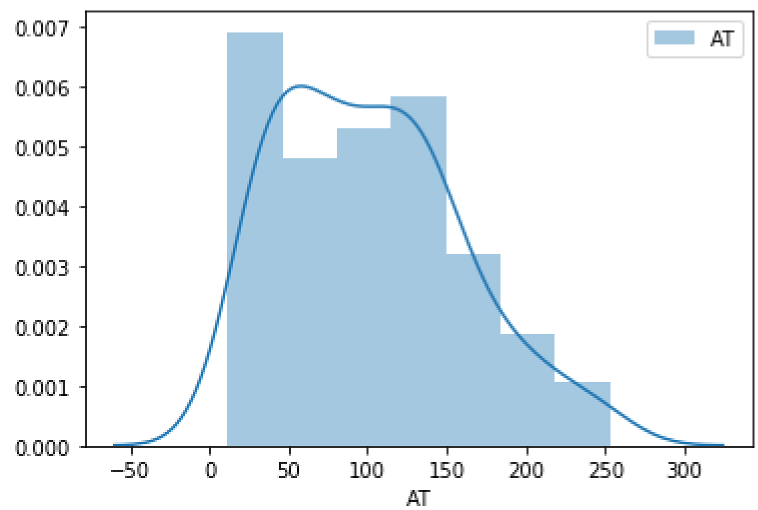
```
AT Mean   : 101.89
AT Median : 96.54
AT Mode   : 121.0
```

In [15]:
```
sns.distplot(df.Waist,label='Waist')
plt.xlabel('Waist')
plt.legend();
```

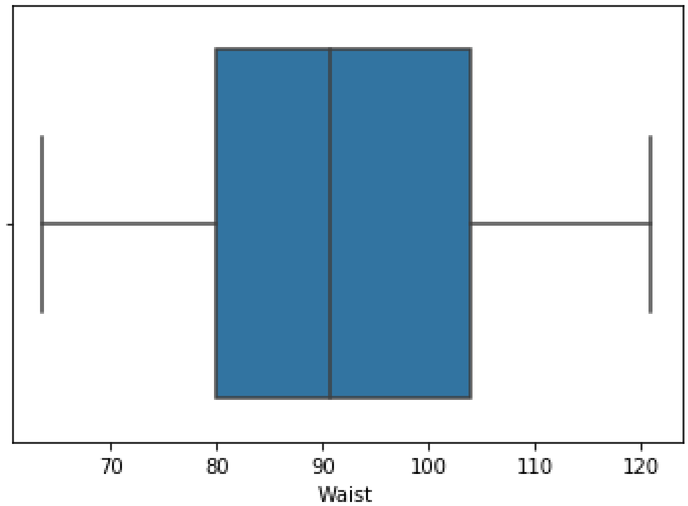


Waist

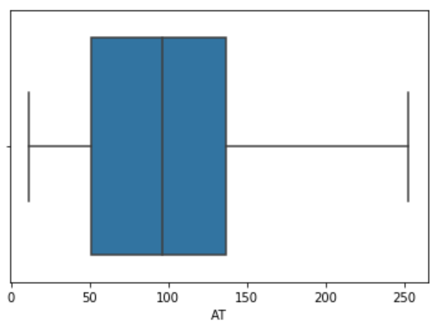
In [16]:
```
sns.distplot(df.AT,label='AT')
plt.xlabel('AT')
plt.legend();
```



```
In [13]: sns.boxplot(df.Waist)
Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1c8915d6408>
```




```
In [14]: sns.boxplot(df.AT)
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x1c891e47208>
```




```
In [2]: # From above graph we infer that
        # AT: mean>median and right whisker is larger than left whisker , therefore AT is +ve skewed
        # Waist : mean approx= median and both whiskers are same lenght , therefore Waist is approx Normal Distribution
```

Q 22) Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval

Ans: @90%

$(1-\alpha)$=90%=.9

α=.1

since we consider $\alpha//2$= .05

(1-.05)= .95

therefore Z=1.65 from z table @ $\alpha/2$ & (1- $\alpha/2$)

Z @ 94% = 1.89

Z @ 60% = 0.85

Q 23) Calculate the t scores of 95% confidence interval, 96% confidence interval, 99% confidence interval for sample size of 25

Ans : stat.t.ppf(CI,SampleSize) - Python

t @ 95% = 1.708
t @ 96% = 1.824
t @ 99% = 2.485

```
In [1]: from scipy import stats
```

From Que

CI : 95%,96%,99%

Sample Size : 25

```
In [9]: print("t-score @ 95% ",round(stats.t.ppf(.95,25),4))
        print("t-score @ 96% ",round(stats.t.ppf(.96,25),4))
        print("t-score @ 99% ",round(stats.t.ppf(.99,25),4))

t-score @ 95%  1.7081
t-score @ 96%  1.8248
t-score @ 99%  2.4851
```

Q 24) 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

Hint:

rcode → pt(tscore,df)

df → degrees of freedom

Ans : 0.3216 by using code stats.t.cdf(t,n) t = -.471
t score is cal using t score formula

$$t = \frac{x - \mu}{\frac{s}{\sqrt{n}}}$$

X= mean of sample =260 Mue = total ppln = 270 n= sample = 18 s = sd=90

In [1]:
```
from scipy import stats
```

- X = mean of sample = 260
- Mue = total ppln = 270
- n = sample = 18
- s = sd = 90
- t = using t-score formula = -0.471

In [5]:
```
print("Probability : ",round(stats.t.cdf(-.471,18),4)*100)
```

```
Probability :  32.16
```