Basic Statistics_Level 1- Assignment ¶

In [34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats
from scipy.stats import norm
import statsmodels.api as sma
%matplotlib inline

import warnings
warnings.filterwarnings("ignore")
```

Question-7

In [51]:
```
car_new=pd.read_csv('Q7.csv')
car_new.head()
```

Out[51]:

	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 [18]:
```
car_new.isnull().sum()
```

Out[18]:
```
Unnamed: 0    2
Points        3
Score         3
Weigh         3
dtype: int64
```

In [52]:
```
car_new.dropna()
car_new.head()
```

Out[52]:

	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 [20]:
```
# mean
car_new.mean()
```

Out[20]:
```
Points     3.435733
Score      3.104597
Weigh     16.992402
dtype: float64
```

In [21]:
```
car_new.median()   # median
```

Out[21]:
```
Points     3.69000
Score      3.21725
Weigh     17.60000
dtype: float64
```

In [22]:
```
car_new.Points.mode() # Mode
```

Out[22]:
```
0    3.92
dtype: float64
```

In [23]:
```
car_new.Score.mode()
```

Out[23]:
```
0    3.44
dtype: float64
```

In [24]:
```
car_new.Weigh.mode()
```

Out[24]:
```
0    17.02
dtype: float64
```

In [25]:
```
# # Variance
car_new.var()
```

Out[25]:
```
Points     0.787648
Score      1.094156
Weigh     15.148019
dtype: float64
```

In [26]:
```
# Satndard Deviation
car_new.std()
```

Out[26]:
```
Points    0.887495
Score     1.046019
Weigh     3.892046
dtype: float64
```

In [27]:
```
# Range
car_new.describe()
```

Out[27]:

	Points	Score	Weigh
count	37.000000	37.000000	37.000000
mean	3.435733	3.104597	16.992402
std	0.887495	1.046019	3.892046
min	0.285881	0.957379	1.786943
25%	3.080000	2.465000	16.870000
50%	3.690000	3.217250	17.600000
75%	3.920000	3.570000	18.610000
max	4.930000	5.424000	22.900000

In [28]:
```
point_range=car_new.Points.max()-car_new.Points.min()
point_range
```

Out[28]:
```
4.644118648999999
```

In [29]:
```
score_range=car_new.Score.max()-car_new.Score.min()
score_range
```

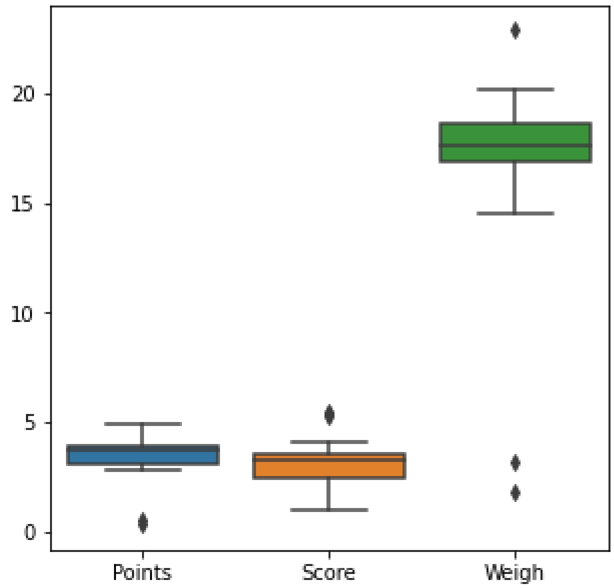
Out[29]:
```
4.466621032000001
```

In [30]:
```
weigh_range=car_new.Weigh.max()-car_new.Weigh.min()
weigh_range
```

Out[30]:
```
21.113056764
```

In [48]:

```
fig, ax = plt.subplots(figsize = (5,5))
sns.boxplot(data = car_new,ax = ax)
plt.show()
```



Question-8

In [6]:

```
q_8 = [108, 110, 123, 134, 135, 145, 167, 187, 199]
q_8 = pd.DataFrame(q_8)
q_8.mean()
```

Out[6]:

```
0    145.333333
dtype: float64
```

Question-9.a)

In [8]:

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

speed_new.head()
```

Out[8]:

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

In [9]:

```
speed_new.skew()
```

Out[9]:

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

In [10]:

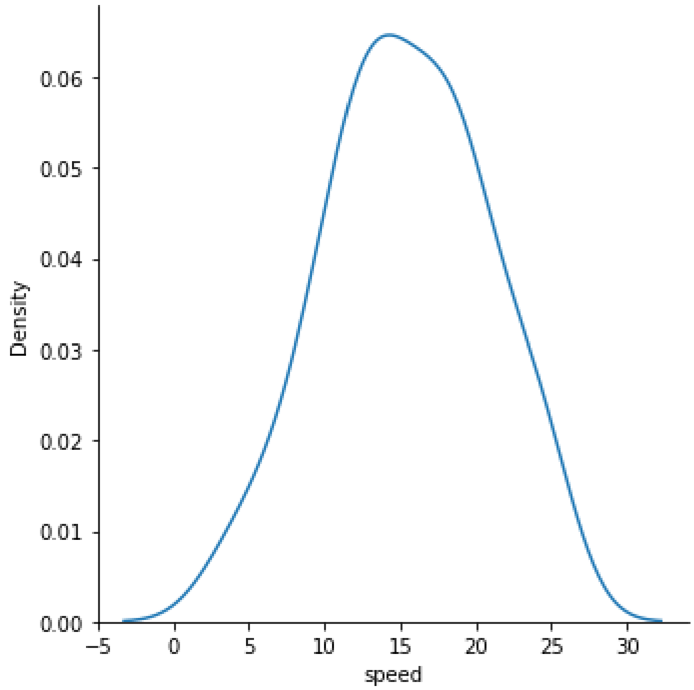
```
speed_new.kurtosis()
```

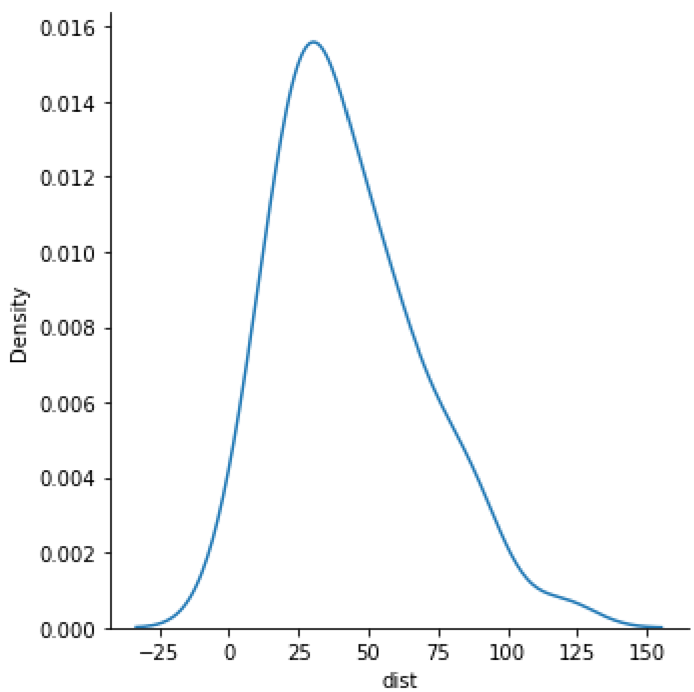
Out[10]:

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

In [11]:
```
sns.displot(data=speed_new['speed'],kind='kde')
sns.displot(data=speed_new['dist'],kind='kde')
```

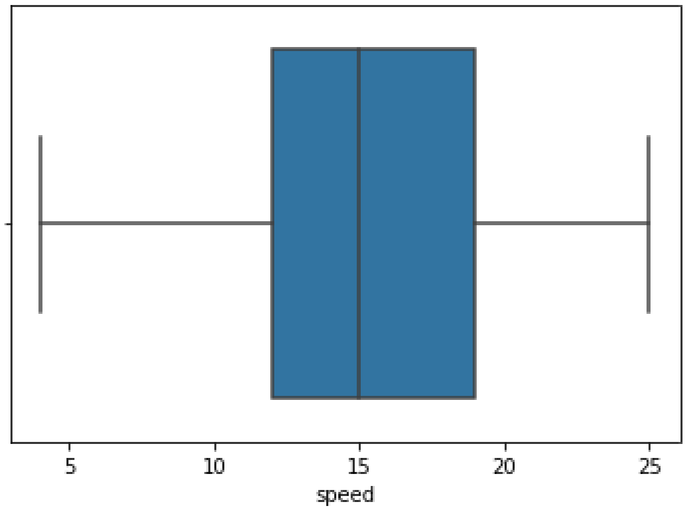
Out[11]: <seaborn.axisgrid.FacetGrid at 0x293d7291df0>





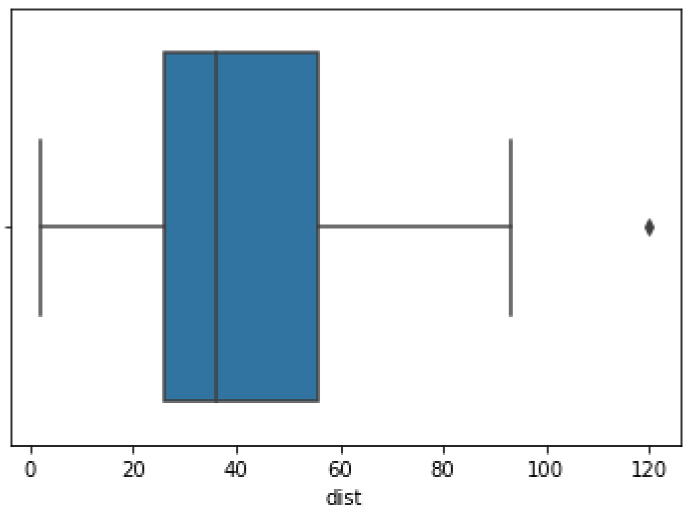
In [12]:
```
sns.boxplot(x=speed_new['speed'],orient='v')
```

Out[12]: <AxesSubplot:xlabel='speed'>

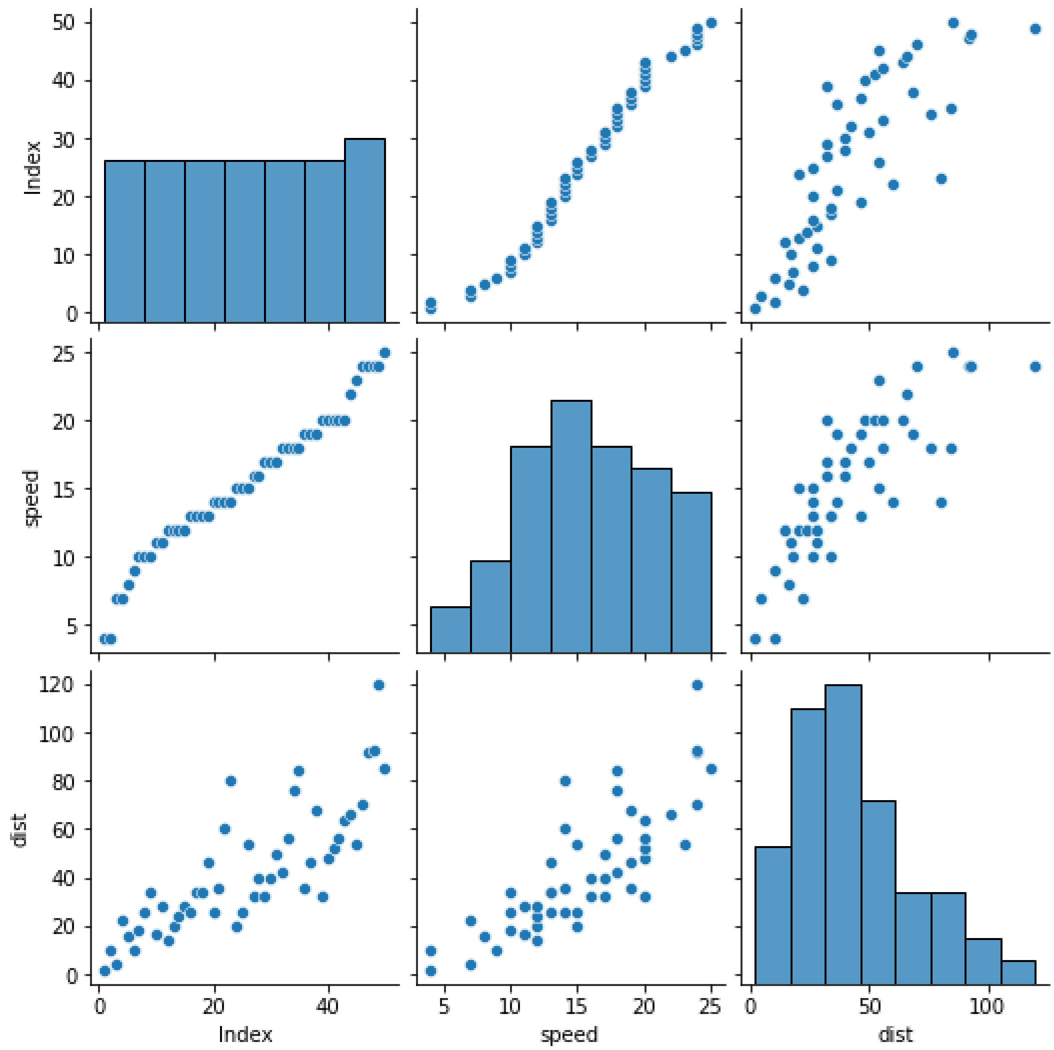


In [13]:
```
sns.boxplot(x=speed_new['dist'],orient='v')
```

Out[13]: <AxesSubplot:xlabel='dist'>



In [15]:
```
sns.pairplot(speed_new)
plt.show()
```



Question-9 b)

In [16]:
```
b_new=pd.read_csv('Q9_b.csv')

b_new.head()
```

Out[16]:

	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

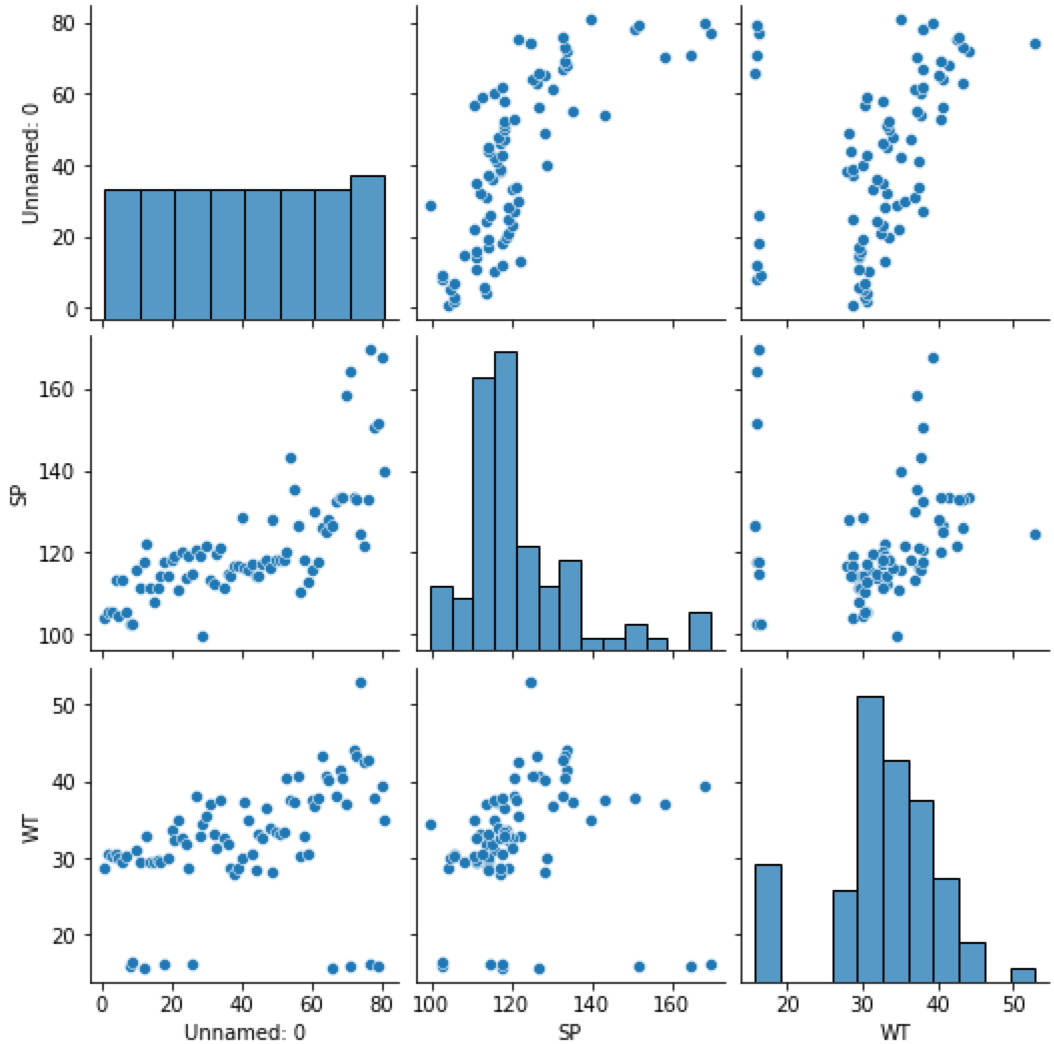
In [66]:
```
b_new.skew()
```

Out[66]:
```
Unnamed: 0    0.000000
SP            1.611450
WT           -0.614753
dtype: float64
```

In [67]:
```
b_new.kurtosis()
```

Out[67]:
```
Unnamed: 0   -1.200000
SP            2.977329
WT            0.950291
dtype: float64
```

In [17]:
```
sns.pairplot(b_new)
plt.show()
```



Question-11

In [18]:
```
# To estimate the average weight of an adult male in Mexico with confidence i
stats.norm.interval(0.94,200,(30/2000**0.5))
```

Out[18]:
```
(198.738325292158, 201.261674707842)
```

In [19]:
```
# To estimate the average weight of an adult male in Mexico with confidence i
stats.norm.interval(0.96,200,(30/2000**0.5))
```

Out[19]:
```
(198.62230334813333, 201.37769665186667)
```

In [70]:
```
# To estimate the average weight of an adult male in Mexico with confidence i
stats.norm.interval(0.98,200,(30/2000**0.5))
```

Out[70]:
```
(198.43943840429978, 201.56056159570022)
```

Question-12

In [77]:
```
a=np.array([34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56])
```

In [81]:
```
print('Mean:',a.mean())
print('Median:',np.median(a))
print('Variance:',a.var())
print('std deviation:',a.std())
```

```
Mean: 41.0
Median: 40.5
Variance: 24.11111111111111
std deviation: 4.910306620885412
```

Question-20

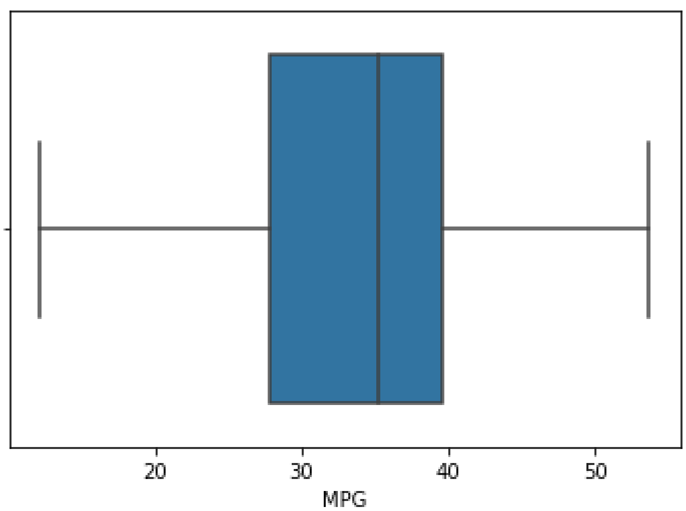
In [36]:
```
cars=pd.read_csv('Cars.csv')
cars.head()
```

Out[36]:

	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 [37]:
```
sns.boxplot(cars.MPG)
```

Out[37]: <AxesSubplot:xlabel='MPG'>



In [38]:
```
# P(MPG>38)
1-stats.norm.cdf(38,cars.MPG.mean(),cars.MPG.std())
```

Out[38]: 0.3475939251582705

In [39]:
```
# P(MPG<40)
stats.norm.cdf(40,cars.MPG.mean(),cars.MPG.std())
```

Out[39]: 0.7293498762151616

In [40]:
```
 #P (20<MPG<50)
#stats.norm.cdf(0.50,cars.MPG.mean(),cars.MPG.std())-stats.norm.cdf(0.20,cars
stats.norm.cdf(50,cars.MPG.mean(),cars.MPG.std())  - stats.norm.cdf(20,cars.M
```

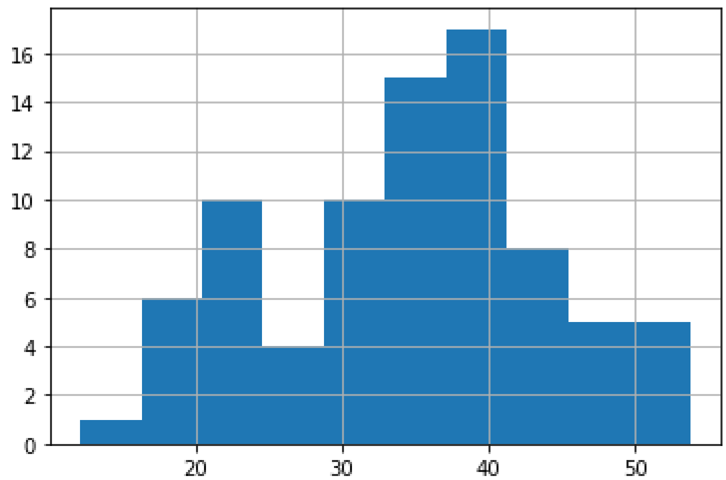
Out[40]: 0.8988689169682046

In [41]:
```
cars.MPG.mode()
```

Out[41]:
```
0    29.629936
dtype: float64
```

In [42]:
```
cars.MPG.hist()
```

Out[42]: <AxesSubplot:>



In [43]:
```
cars.MPG.skew()
```

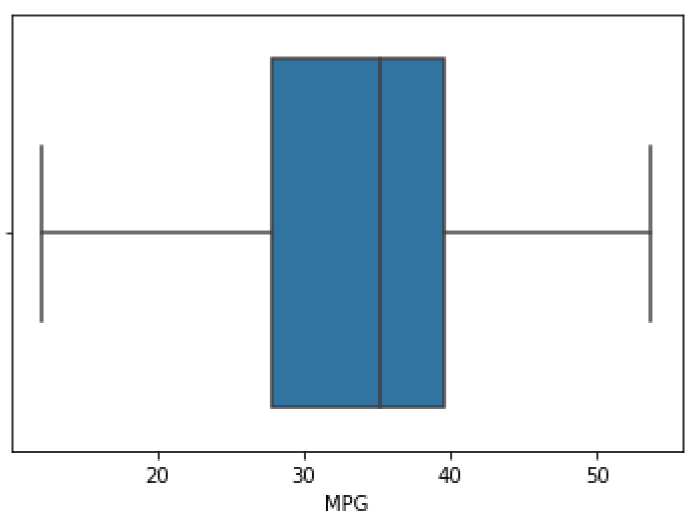
Out[43]: -0.17794674747025727

In [44]:
```
cars.MPG.kurt()
```

Out[44]: -0.6116786559430913

In [45]:
```
sns.boxplot(cars.MPG)
```

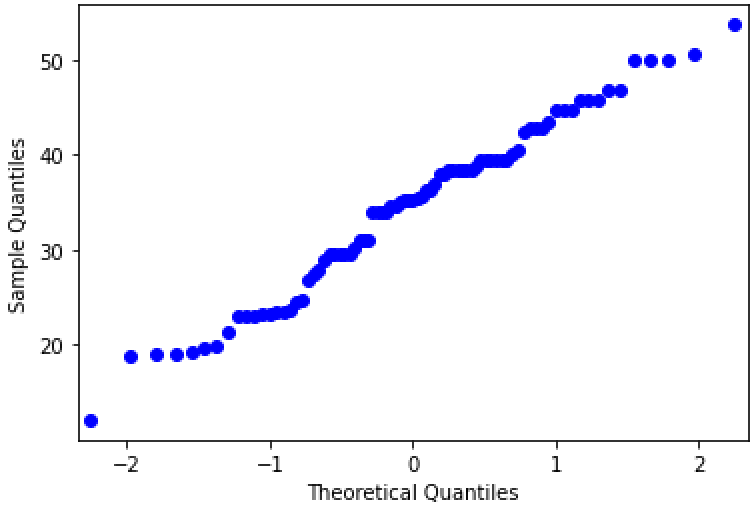
Out[45]: <AxesSubplot:xlabel='MPG'>

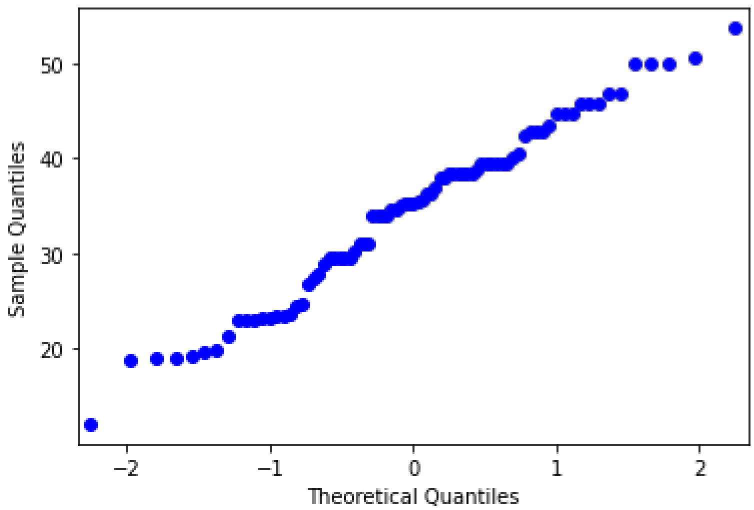


In [46]:

```
sma.qqplot(cars.MPG,dist=stats.norm)
```

Out[46]:



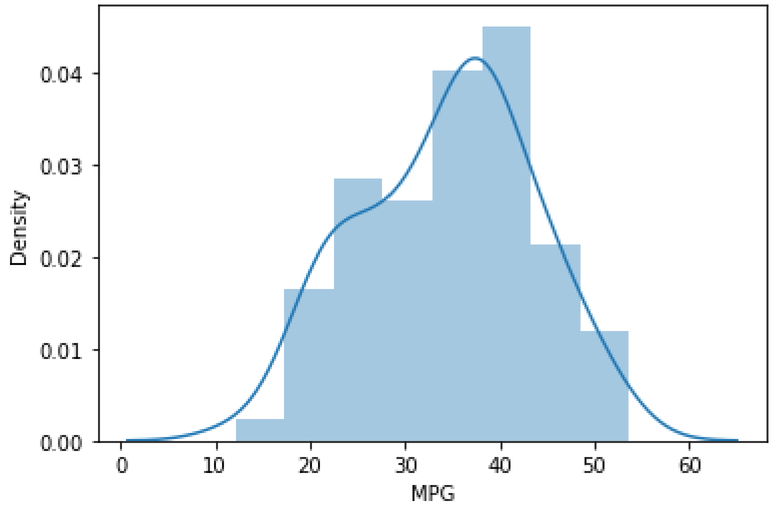


Question-21.a

a) To check whether the MPG of Cars follows Normal Distribution

In [48]:

```
sns.distplot(cars.MPG)
plt.show()
```



In [49]:

```
print("To Check whether the MPG of Cars follows Normal Distribution:")
print("Mean :",cars['MPG'].mean())
print("Median :",cars['MPG'].median())
print("Mode :",cars['MPG'].mode())
print("Skewness :",cars['MPG'].skew())
```

```
To Check whether the MPG of Cars follows Normal Distribution:
Mean : 34.422075728024666
Median : 35.15272697
Mode : 0    29.629936
dtype: float64
Skewness : -0.17794674747025727
```

Question-21.b

b. To check Whether the Adipose Tissue (AT) and Waist Circumference(Waist) follows Normal Distribution

In [50]:
```
df=pd.read_csv('wc-at.csv')

df.head()
```

Out[50]:

	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 [51]:
```
df.mean()
```

Out[51]:
```
Waist     91.901835
AT       101.894037
dtype: float64
```

In [52]:
```
df.median()
```

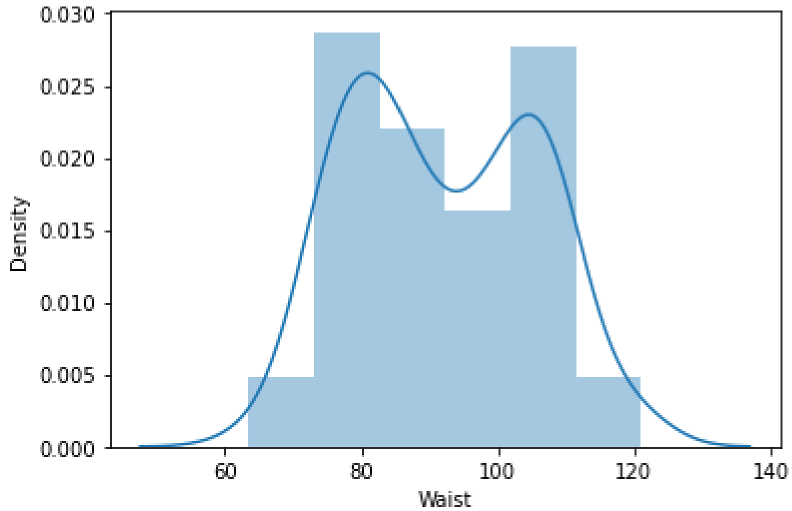
Out[52]:
```
Waist    90.80
AT       96.54
dtype: float64
```

In [53]:
```
df.mode()
```

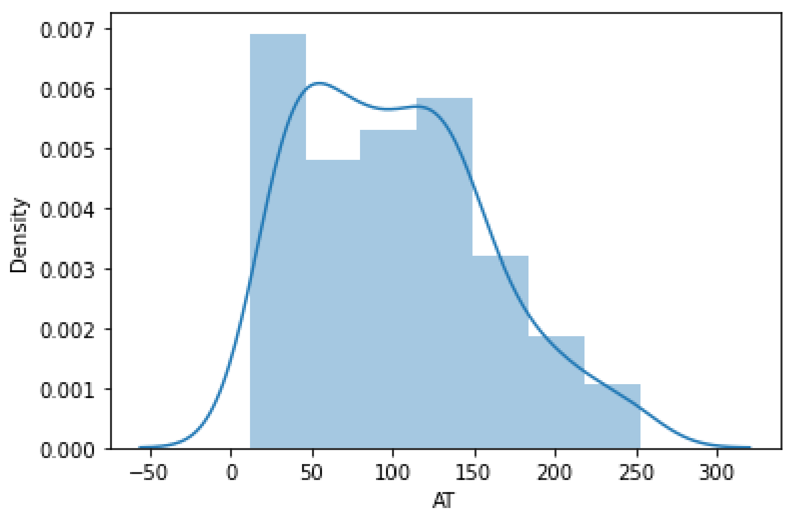
Out[53]:

	Waist	AT
0	94.5	121.0
1	106.0	123.0
2	108.5	NaN

In [54]:
```
sns.distplot(df['Waist'])
plt.show()
```

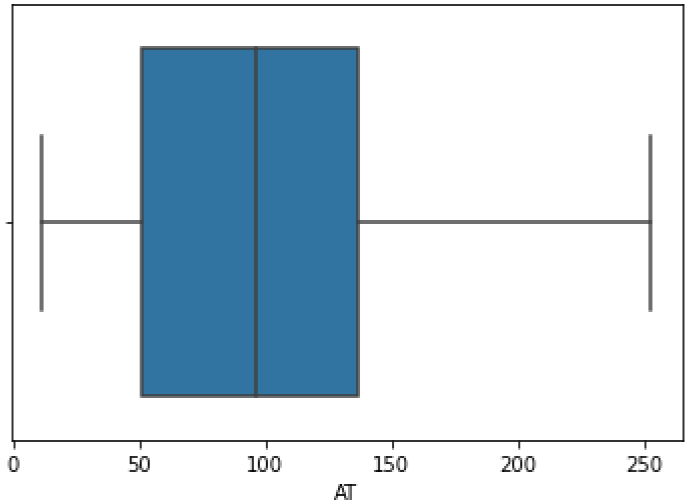


In [55]:
```
sns.distplot(df['AT'])
plt.show()
```



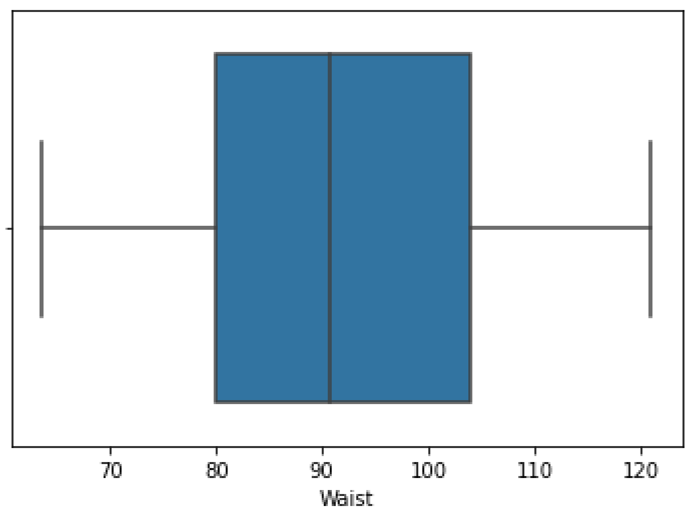
In [56]:

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



In [57]:

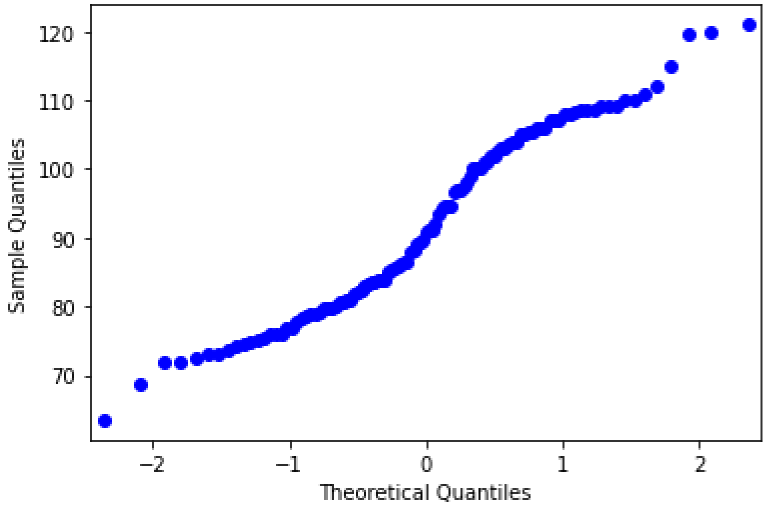
```
sns.boxplot(df['Waist'])
plt.show()
```

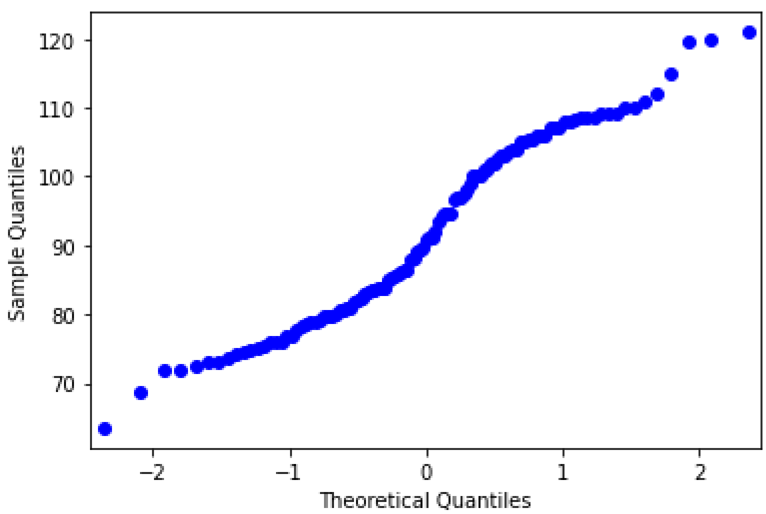


In [58]:

```
sma.qqplot(df.Waist)#,dist=stats.norm)
```

Out[58]:





Question-22

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

In [59]:
```
# To calculate the Z-score of 90% confidence interval
stats.norm.ppf(0.95)
```
Out[59]: 1.6448536269514722

In [60]:
```
# To calculate the Z-score of 94% confidence interval
stats.norm.ppf(0.93)
```
Out[60]: 1.475791028179171

In [61]:
```
# To calculate the Z-score of 60% confidence interval
stats.norm.ppf(0.8)
```
Out[61]: 0.8416212335729143

Question-23

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

In [62]:
```
# To calculate the T-score of 95% confidence interval with sample size of 25
stats.t.ppf(0.975,24)
```
Out[62]: 2.0638985616280205

In [63]:
```
# To calculate the T-score of 96% confidence interval with sample size of 25
stats.t.ppf(0.98,24)
```
Out[63]: 2.1715446760080677

In [64]:
```
# To calculate the T-score of 99% confidence interval with sample size of 25
stats.t.ppf(0.995,24)
```
Out[64]: 2.796939504772804

Question-24

Null Hypothesis - Average Life of light bulb >= 260

Alternate Hypothesis = Averageage Life of Light bulb <260

To calculate T score T = (X – μ) / [σ/√(n)].

In [117]:
```
sample_mean = 260
population_mean = 270
Sample_std_deviation = 90
Tscore = (260 -270)/(90/(18**0.5))
```

In [118]:
```
Tscore
```

Out[118]:
```
-0.4714045207910317
```

In [119]:
```
stats.t.cdf((Tscore),17)
```

Out[119]:
```
0.32167253567098364
```