ASSIGNMENT 1- BASIC STATISTICS LEVEL 1

IMPORT NECESSARY LIBRARIES

In [178…
```
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt

import numpy as np
from scipy import stats
from scipy.stats import norm
import warnings
warnings.filterwarnings('ignore')
```

QUESTION - 7

In [53]:
```
car_data=pd.read_csv(r'D:\Downloads\Q7.csv')
car_data
```

Out[53]:

	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 [54]:
```
car_data.dtypes
```

Out[54]:
```
Unnamed: 0     object
Points        float64
Score         float64
Weigh         float64
dtype: object
```

In [55]:
```
car_data.mean()
```

Out[55]:
```
Points     3.596563
Score      3.217250
Weigh     17.848750
dtype: float64
```

In [56]:
```
car_data.median()
```

Out[56]:
```
Points     3.695
Score      3.325
Weigh     17.710
dtype: float64
```

In [75]:
```
car_data.Points.mode()
```

Out[75]:
```
0    3.07
1    3.92
Name: Points, dtype: float64
```

In [58]:
```
car_data.Score.mode()
```

Out[58]:
```
0    3.44
Name: Score, dtype: float64
```

In [59]:
```
car_data.Weigh.mode()
```

Out[59]:
```
0    17.02
1    18.90
Name: Weigh, dtype: float64
```

In [60]:
```
car_data.var()
```

Out[60]:
```
Points    0.285881
Score     0.957379
Weigh     3.193166
dtype: float64
```

In [61]:
```
car_data.std()
```

Out[61]:
```
Points    0.534679
Score     0.978457
Weigh     1.786943
dtype: float64
```

In [66]:
```
car_data.describe()
```

Out[66]:

	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 [68]:
```
points_range = car_data.Points.max() - car_data.Points.min()
points_range
```

Out[68]:
```
2.17
```

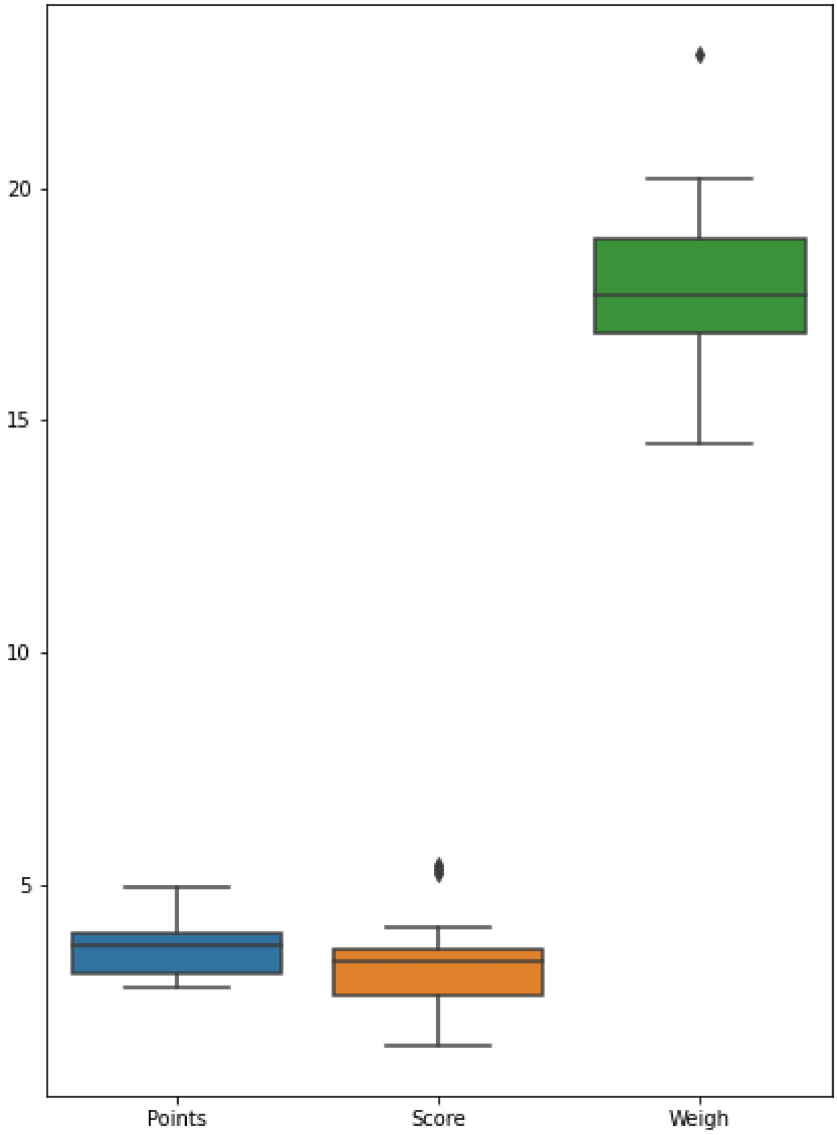
In [69]:
```
score_range = car_data.Score.max() - car_data.Score.min()
score_range
```

Out[69]:
```
3.9110000000000005
```

In [73]:
```
weigh_range = car_data.Weigh.max() - car_data.Weigh.min()
weigh_range
```

Out[73]:
```
8.399999999999999
```

In [86]:
```
plt.figure(figsize=(7,10))
sns.boxplot(data=car_data)
plt.show()
```



QUESTION - 9

A) Cars speed and distance

In [91]:
```
car_speed = pd.read_csv(r'D:\Downloads\Q9_a.csv')
car_speed
```

Out[91]:

	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
33	34	18	76
34	35	18	84
35	36	19	36

	Index	speed	dist
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 [92]:
```
car_speed.skew()
```

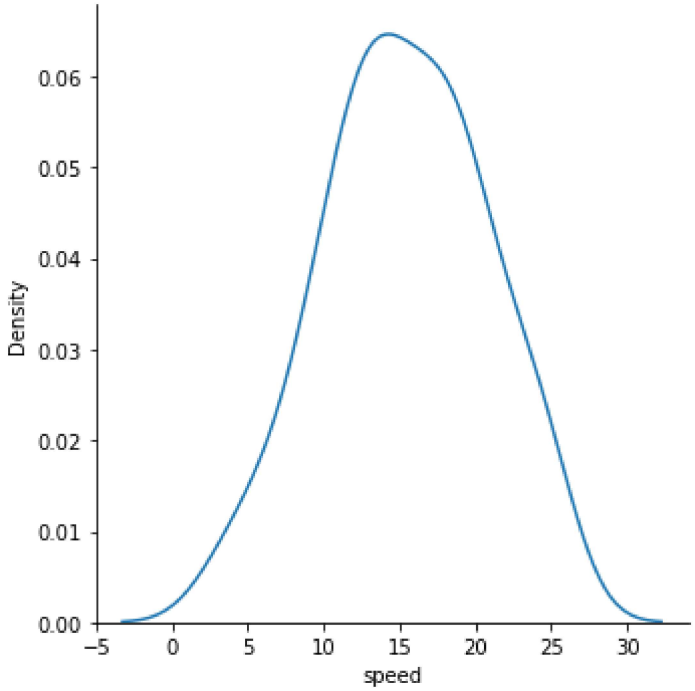
Out[92]:
```
Index    0.000000
speed   -0.117510
dist     0.806895
dtype: float64
```

In [94]:
```
car_speed.kurtosis()
```

Out[94]:
```
Index   -1.200000
speed   -0.508994
dist     0.405053
dtype: float64
```

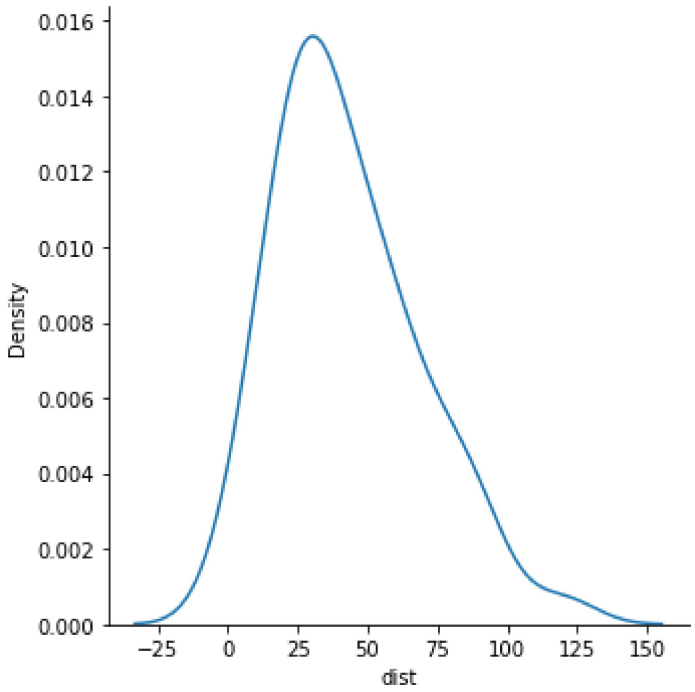
In [134…
```
sns.displot(data = car_speed['speed'],kind='kde')
```

Out[134]:
```
<seaborn.axisgrid.FacetGrid at 0x2e1e306a700>
```



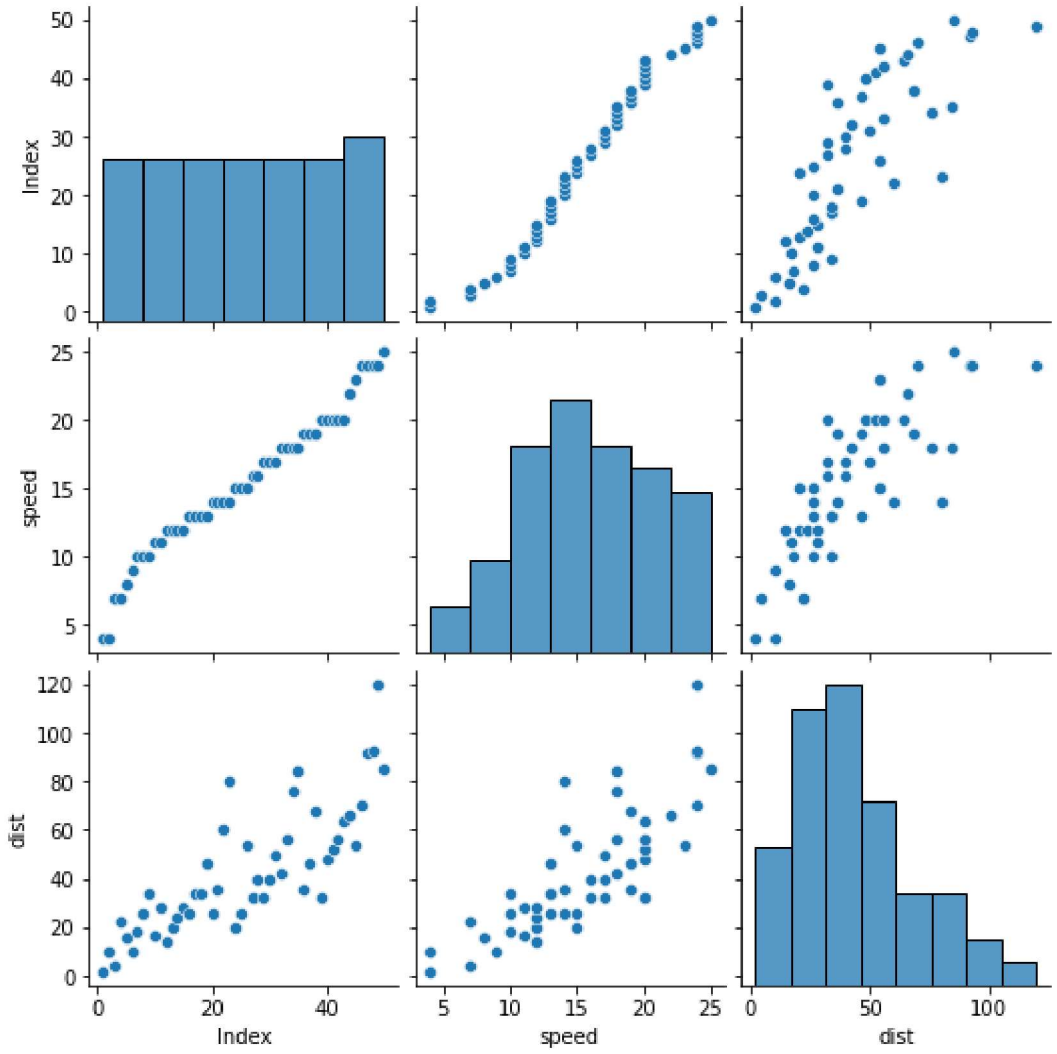
In [135…
```
sns.displot(data=car_speed['dist'],kind='kde')
```

Out[135]:
```
<seaborn.axisgrid.FacetGrid at 0x2e1e28c69d0>
```



In [136…
```
sns.pairplot(data = car_speed)
```

Out[136]:
```
<seaborn.axisgrid.PairGrid at 0x2e1e2760910>
```



B) SP and Weight(WT)

In [138…

```
car_speed1=pd.read_csv(r'D:\Downloads\Q9_b.csv')
car_speed1
```

Out[138]:

	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
...	...	...	...
76	77	169.598513	16.132947
77	78	150.576579	37.923113
78	79	151.598513	15.769625
79	80	167.944460	39.423099
80	81	139.840817	34.948615

81 rows × 3 columns

In [139…
```
car_speed1.skew()
```

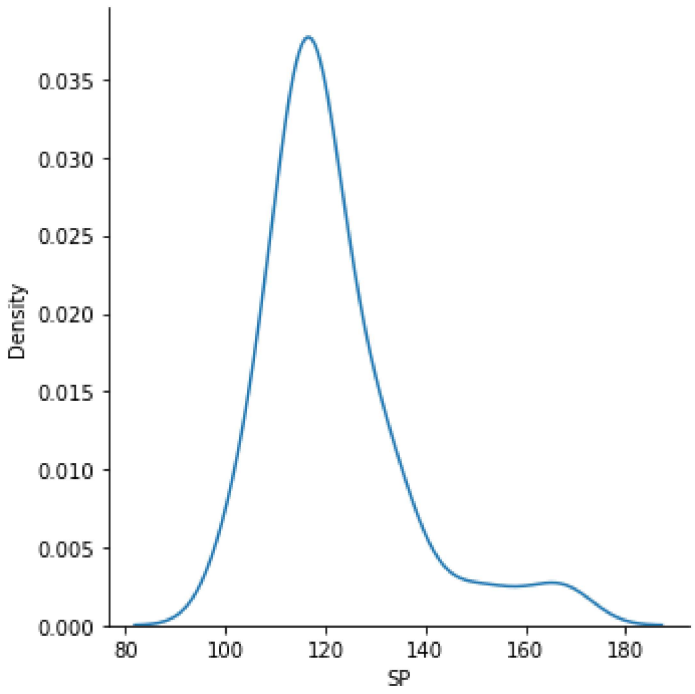
Out[139]:
```
Unnamed: 0    0.000000
SP            1.611450
WT           -0.614753
dtype: float64
```

In [140…
```
car_speed1.kurtosis()
```

Out[140]:
```
Unnamed: 0   -1.200000
SP            2.977329
WT            0.950291
dtype: float64
```

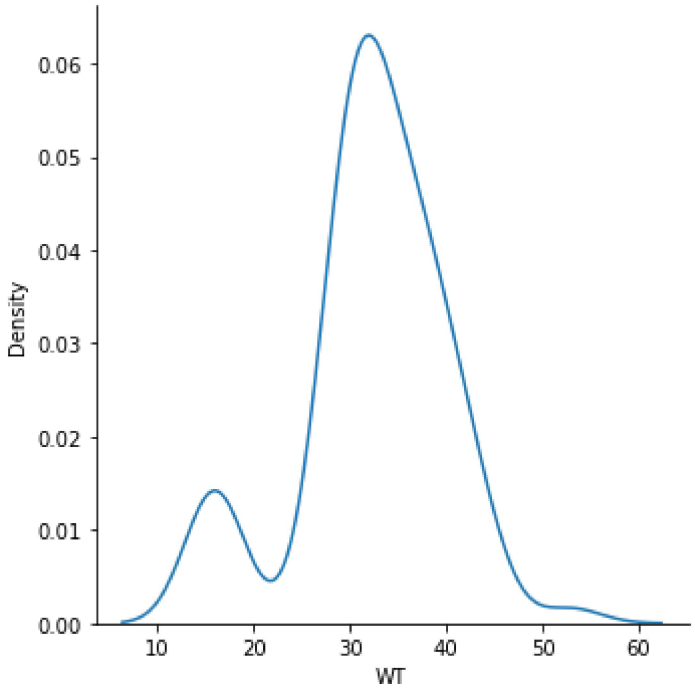
In [142…
```
sns.displot(data = car_speed1 ['SP'], kind='kde')
```

Out[142]:
```
<seaborn.axisgrid.FacetGrid at 0x2e1e2a9f460>
```



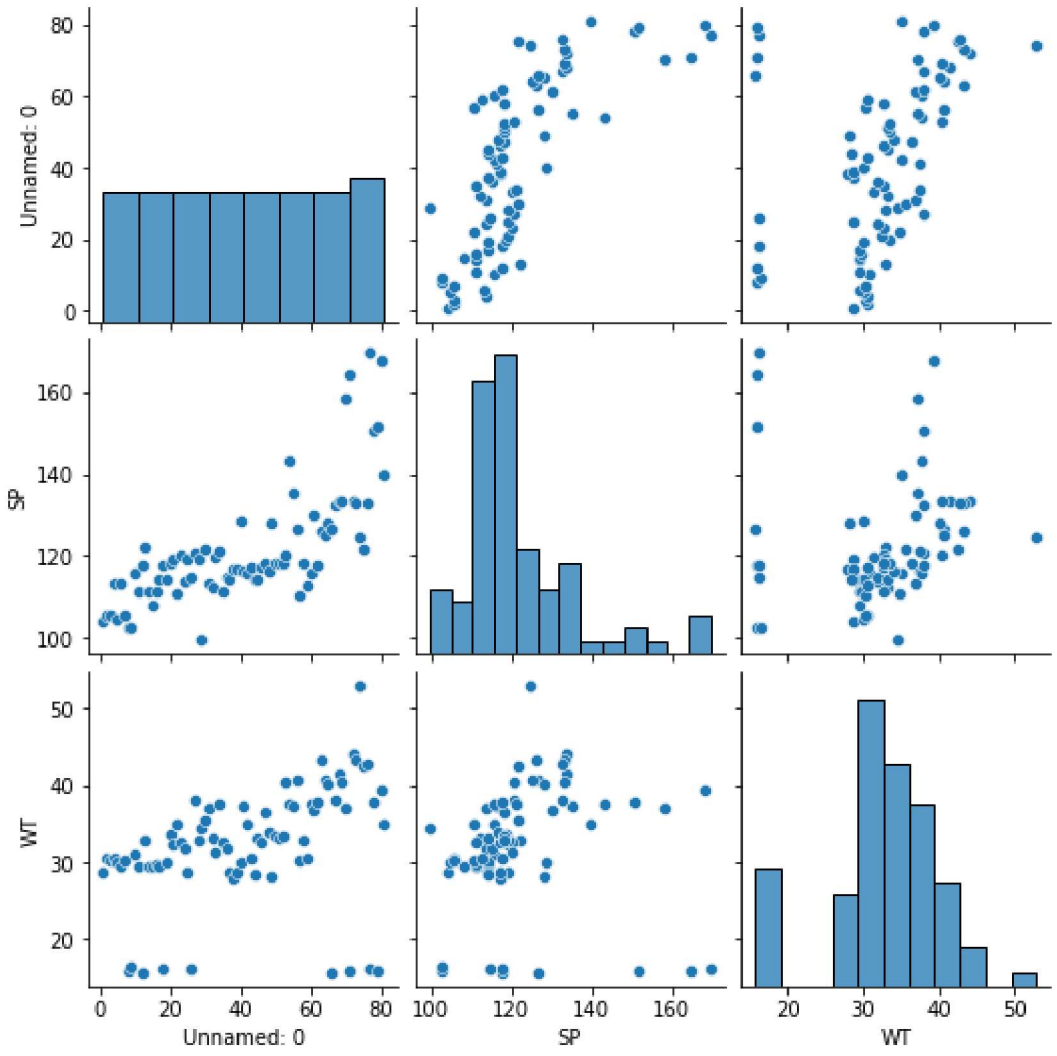
In [143…
```
sns.displot(data = car_speed1['WT'], kind= 'kde')
```

Out[143]:
```
<seaborn.axisgrid.FacetGrid at 0x2e1e3045400>
```



In [144…
```
sns.pairplot(data = car_speed1)
```

Out[144]:
```
<seaborn.axisgrid.PairGrid at 0x2e1e2ba93d0>
```



QUESTION - 11

In [232…
```
#  94% confidence interval
stats.norm.interval(0.94,200,(30/2000**0.5))
```

Out[232]:
```
(198.738325292158, 201.261674707842)
```

In [234…
```
# 98% confidence interval
stats.norm.interval(0.98,200,(30/2000**0.5))
```

Out[234]:
```
(198.43943840429978, 201.56056159570022)
```

In [235…
```
# 96% confidence interval
stats.norm.interval(0.96,200,(30/2000**0.5))
```

Out[235]:
```
(198.62230334813333, 201.37769665186667)
```

QUESTION - 12

In [149…
```
scores = np.array([34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56 ])
scores
```

Out[149]:
```
array([34, 36, 36, 38, 38, 39, 39, 40, 40, 41, 41, 41, 41, 42, 42, 45, 49,
       56])
```

In [171…
```
print('Mean = ',scores.mean())
print('median = ',np.median(scores))
print('Variance = ',scores.var().round(3))
print('Standard deviation = ',scores.std().round(3))
```

```
Mean =  41.0
median =  40.5
Variance =  24.111
Standard deviation =  4.91
```

QUESTION - 20

In [161…
```
cars_data=pd.read_csv(r'D:\Downloads\cars (2).csv')
cars_data
```

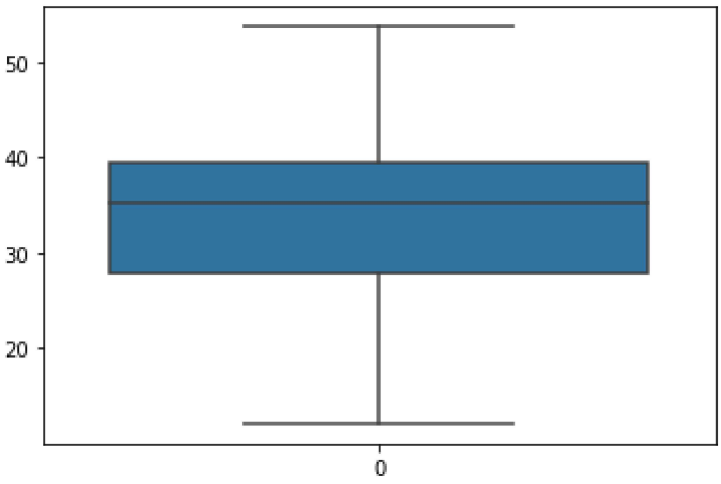
Out[161]:

	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

In [163…
```
sns.boxplot(data=cars_data['MPG'])
```

Out[163]:
```
<AxesSubplot:>
```



In [182…
```
# P(MPG>38)
1-stats.norm.cdf(38,cars_data.MPG.mean(),cars_data.MPG.std()).round(3)
```

Out[182]:
```
0.348
```

In [183…
```
# p(MPG<40)
stats.norm.cdf(40,cars_data.MPG.mean(),cars_data.MPG.std()).round(3)
```

Out[183]:
```
0.729
```

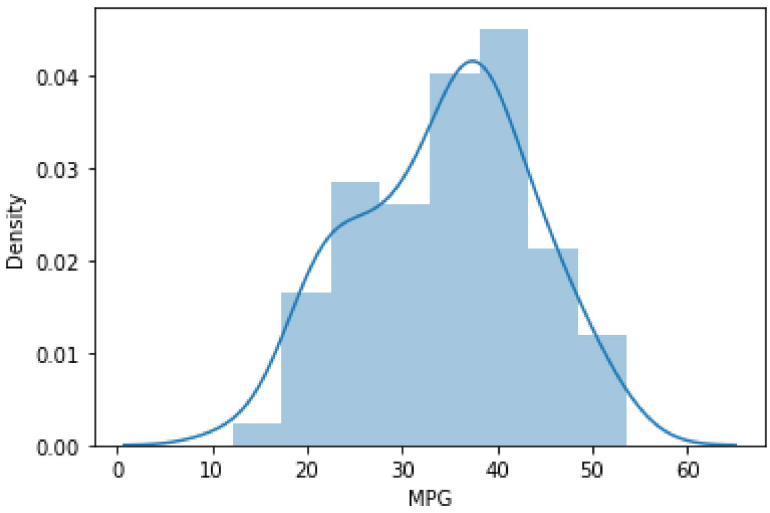
In [191…
```
#P (20<MPG<50)
(stats.norm.cdf(50,cars_data.MPG.mean(),cars_data.MPG.std())-stats.norm.cdf(20,cars
```

Out[191]:
```
0.899
```

QUESTION - 21

21) A- Find the normal distribution of cars

In [209…
```
sns.distplot(cars_data.MPG)
plt.show()
```



In [213…
```
print('Check whether the data follows normal distribution')
print('Mean = ',cars_data['MPG'].mean())
print('Median = ',cars_data['MPG'].median())
print('Mode = ',cars_data['MPG'].mode())
print('skewness = ',cars_data['MPG'].skew())
```

```
Check whether the data follows normal distribution
Mean =  34.422075728024666
Median =  35.15272697
Mode =  0    29.629936
Name: MPG, dtype: float64
skewness =  -0.17794674747025727
```

21) B - Find the normal distribution of Adipose Tissue (AT) and Waist Circumference(Waist)

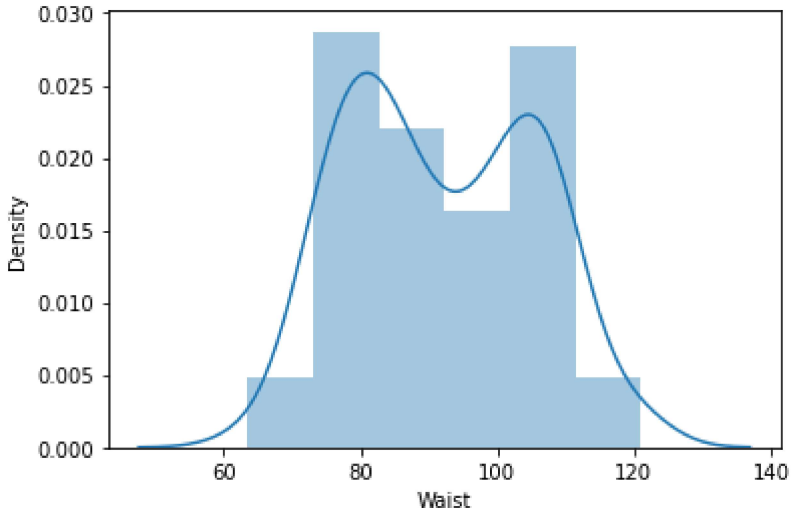
In [211…
```
type_data = pd.read_csv(r'D:\Downloads\wc-at.csv')
type_data
```

Out[211]:

	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

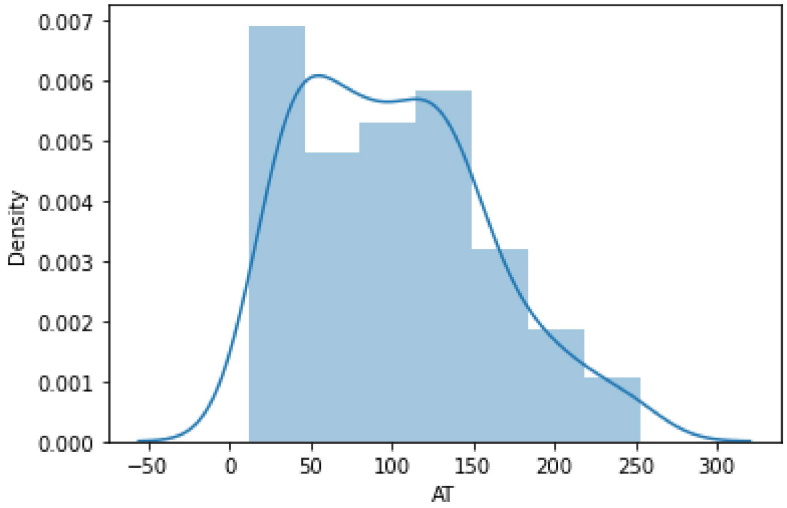
In [218…
```
sns.distplot(type_data.Waist)
plt.show()
```



In [220…
```
print('Check whether the data follows normal distribution')
print('Mean = ',type_data['Waist'].mean())
print('Median = ',type_data['Waist'].median())
print('Mode = ',type_data['Waist'].mode())
print('skewness = ',type_data['Waist'].skew())
```

```
Check whether the data follows normal distribution
Mean =  91.90183486238533
Median =  90.8
Mode =  0     94.5
1    106.0
2    108.5
Name: Waist, dtype: float64
skewness =  0.1340560824786468
```

In [219…
```
sns.distplot(type_data.AT)
plt.show()
```



In [221…
```
print('Check whether the data follows normal distribution')
print('Mean = ',type_data['AT'].mean())
print('Median = ',type_data['AT'].median())
print('Mode = ',type_data['AT'].mode())
print('skewness = ',type_data['AT'].skew())
```

```
Check whether the data follows normal distribution
Mean =  101.89403669724771
Median =  96.54
Mode =  0    121.0
1    123.0
Name: AT, dtype: float64
skewness =  0.584869324127853
```

QUESTION - 22

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

In [226…
```
# Calculate the Z scores of  90% confidence interval
stats.norm.ppf(0.95)
```

Out[226]:
```
1.6448536269514722
```

In [227…
```
# Calculate the Z scores of 94% confidence interval
stats.norm.ppf(0.93)
```

Out[227]:
```
1.475791028179171
```

In [228…
```
# Calculate the Z scores of 60% confidence interval
stats.norm.ppf(0.8)
```

Out[228]:
```
0.8416212335729143
```

QUESTION - 23

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

In [229…
```
#  Calculate the t scores of 95% confidence interval for sample size of 25
stats.t.ppf(0.975,24)
```

Out[229]:
```
2.0638985616280205
```

In [230…
```
# Calculate the t scores of 96% confidence interval for sample size of 25
stats.t.ppf(0.98,24)
```

Out[230]:
```
2.1715446760080677
```

In [231…
```
#  Calculate the t scores of 99% confidence interval for sample size of 25
stats.t.ppf(0.995,24)
```

Out[231]:
```
2.796939504772804
```

QUESTION - 24

In [237…
```
sample_mean = 260
population_mean = 270
Sample_std_deviation = 90
test_data = (260 -270)/(90/(18**0.5))
```

In [238…
```
test_data
```

Out[238]:
```
-0.4714045207910317
```

In [239…
```
stats.t.cdf((Tscore),17)
```

Out[239]:
```
0.32167253567098364
```