Assignment-1-Q7 (Basic Statistics Level-1)

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:

```
cars=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/Q7.csv")
```

In [3]:

```
cars
```

Out[3]:

	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 [4]:

```
# mean
cars.mean()
```

Out[4]:

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

In [5]:

```
# Median
cars.median()
```

Out[5]:

```
Points     3.695
Score      3.325
Weigh     17.710
dtype: float64
```

In [7]:

```
# Mode
cars.Points.mode()
```

Out[7]:

```
0    3.07
1    3.92
dtype: float64
```

In [8]:

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

Out[8]:

```
0    3.44
dtype: float64
```

In [9]:

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

Out[9]:

```
0    17.02
1    18.90
dtype: float64
```

In [10]:

```
# variance
cars.var()
```

Out[10]:

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

In [11]:

```
# Standard Deviation
cars.std()
```

Out[11]:

```
Points    0.534679
Score     0.978457
Weigh     1.786943
dtype: float64
```

In [12]:

```
# Range
cars.describe()
```

Out[12]:

	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 [16]:

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

Out[16]:

```
2.17
```

In [17]:

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

Out[17]:

```
3.9110000000000005
```

In [18]:

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

Out[18]:

```
8.399999999999999
```

Assignment-1-Q9_a

In [19]:

```
data1=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/Q9_a.csv")
```

In [20]:

```
data1
```

Out[20]:

	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

	Index	speed	dist
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 [21]:

```
# Skewness
data1.skew()
```

Out[21]:

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

In [22]:

```
# Kurtosis
data1.kurt()
```

Out[22]:

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

Assignment-1-Q9_b

In [23]:

```
data2=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/Q9_b.csv")
```

In [25]:

```
data2
```

Out[25]:

	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 [27]:

```
data3=data2.iloc[:,1:]
data3
```

Out[27]:

	SP	WT
0	104.185353	28.762059
1	105.461264	30.466833
2	105.461264	30.193597
3	113.461264	30.632114
4	104.461264	29.889149
...	...	...
76	169.598513	16.132947
77	150.576579	37.923113
78	151.598513	15.769625
79	167.944460	39.423099
80	139.840817	34.948615

81 rows × 2 columns

In [28]:

```
# Skewness
data3.skew()
```

Out[28]:

```
SP    1.611450
WT   -0.614753
dtype: float64
```

In [29]:

```
data3.kurt()
```

Out[29]:

```
SP    2.977329
WT    0.950291
dtype: float64
```

Assignment-1-Q11

In [30]:

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

In [31]:

```
# Avg. weight of Adult in Mexico with 94% confidence interval
stats.norm.interval(0.94,200,30/(2000**0.5))
```

Out[31]:

```
(198.738325292158, 201.261674707842)
```

In [32]:

```
# Avg.  weight of Adult in Mexico with 98% confidence interval
stats.norm.interval(0.98,200,30/(2000**0.5))
```

Out[32]:

```
(198.43943840429978, 201.56056159570022)
```

In [35]:

```
# Avg.  weight of Adult in Mexico with 96% confidence interval
stats.norm.interval(0.96,200,30/(2000**0.5))
```

Out[35]:

```
(198.62230334813333, 201.37769665186667)
```

In [ ]:

```

```

Assignment-1-Q12

In [36]:

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

In [37]:

```
# Mean
x.mean()
```

Out[37]:

```
41.0
```

In [38]:

```
# Median
x.median()
```

Out[38]:

```
40.5
```

In [39]:

```
# Varience
x.var()
```

Out[39]:

```
25.529411764705884
```

In [40]:

```
# Standard Deviation
x.std()
```

Out[40]:

```
5.05266382858645
```

In [ ]:

```

```

Assignment-1-Q20

In [41]:

```
cars=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/Cars.csv")
cars
```

Out[41]:

	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

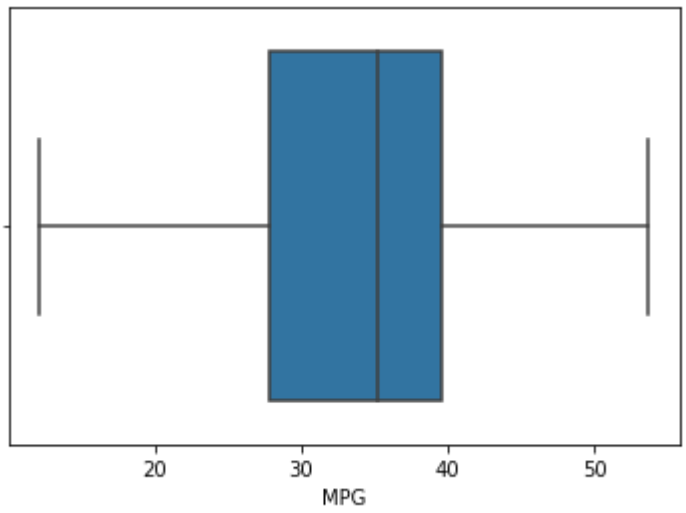
In [42]:

```
sns.boxplot(cars.MPG)
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\_decorators.py:36: Futur
eWarning: Pass the following variable as a keyword arg: x. From version 0.1
2, the only valid positional argument will be `data`, and passing other argu
ments without an explicit keyword will result in an error or misinterpretati
on.
  warnings.warn(
```

Out[42]:

```
<AxesSubplot:xlabel='MPG'>
```



In [43]:

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

Out[43]:

```
0.3475939251582705
```

In [45]:

```
# P(MPG<40)
stats.norm.cdf(40,cars.MPG.mean(),cars.MPG.std())
```

Out[45]:

```
0.7293498762151616
```

In [48]:

```
# P(20<MPG<50)
stats.norm.cdf(50,cars.MPG.mean(),cars.MPG.std())-stats.norm.cdf(20,cars.MPG.mean(),cars.MP
```

Out[48]:

```
0.8988689169682046
```

In [ ]:

```

```

Assignment-1-Q21_a

In [49]:

```
cars2=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/Cars.csv")
cars2
```

Out[49]:

	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

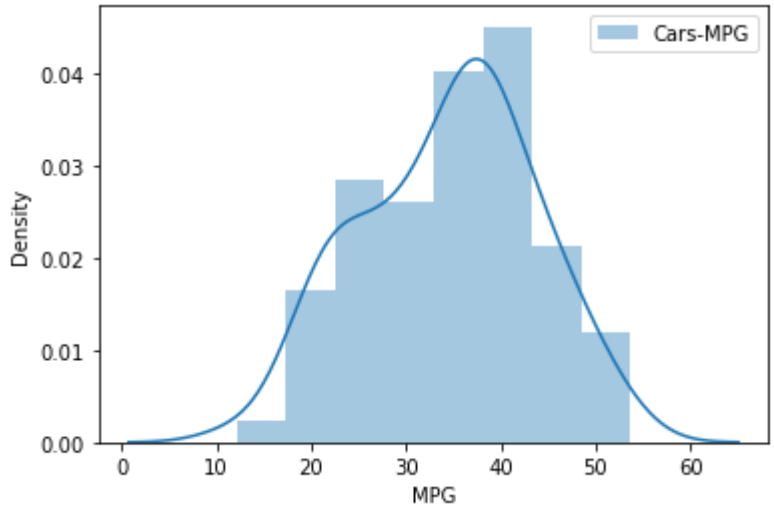
In [50]:

```
sns.distplot(cars2.MPG, label='Cars-MPG')
plt.xlabel('MPG')
plt.ylabel('Density')
plt.legend()
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\distributions.py:2557: F
utureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function f
or histograms).
  warnings.warn(msg, FutureWarning)
```

Out[50]:

```
<matplotlib.legend.Legend at 0x4d90e076a0>
```



In [51]:

```
cars2.MPG.mean()
```

Out[51]:

```
34.422075728024666
```

In [52]:

```
cars2.MPG.median()
```

Out[52]:

```
35.15272697
```

Assignment-1-Q21_b

In [54]:

```
wcat=pd.read_csv("C:/Users/LENOVO/Documents/Custom Office Templates/wc-at.csv")
wcat
```

Out[54]:

	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

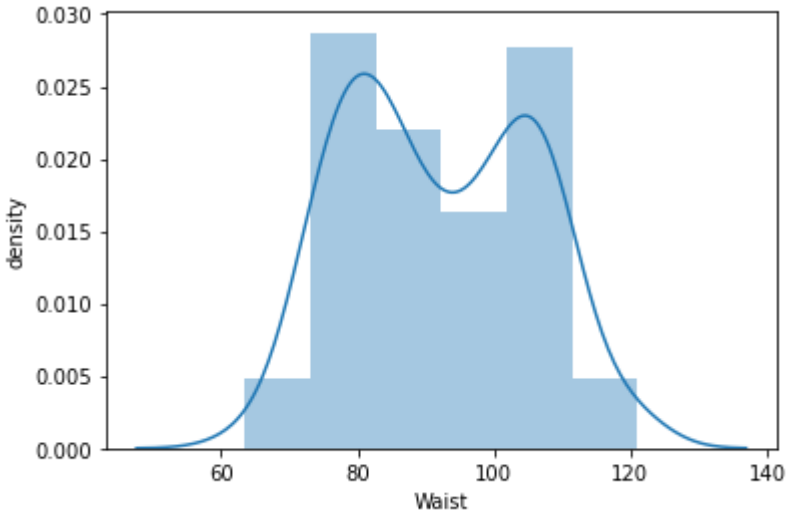
In [55]:

```
# plotting distribution for waist circumference (waist)
sns.distplot(wcat.Waist)
plt.ylabel('density')
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\distributions.py:2557: F
utureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function f
or histograms).
  warnings.warn(msg, FutureWarning)
```

Out[55]:

```
Text(0, 0.5, 'density')
```



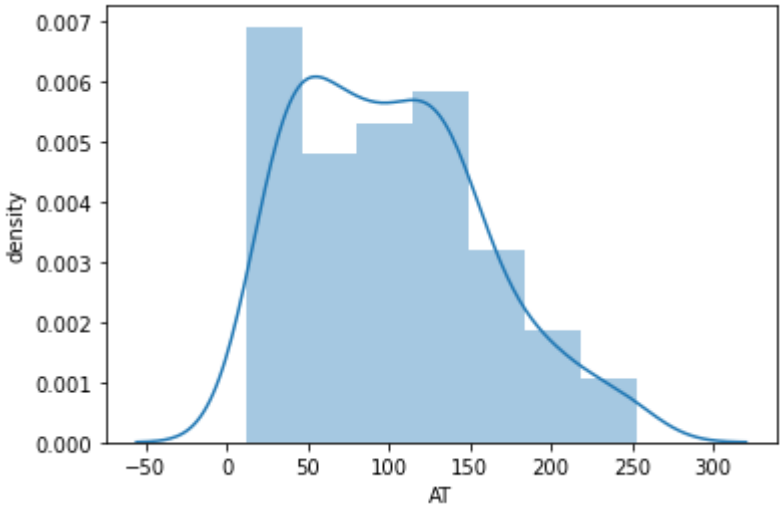
In [56]:

```
# plotting distribution for Adipose Tissue (AT)
sns.distplot(wcat.AT)
plt.ylabel('density')
```

```
C:\Users\LENOVO\anaconda3\lib\site-packages\seaborn\distributions.py:2557: F
utureWarning: `distplot` is a deprecated function and will be removed in a f
uture version. Please adapt your code to use either `displot` (a figure-leve
l function with similar flexibility) or `histplot` (an axes-level function f
or histograms).
  warnings.warn(msg, FutureWarning)
```

Out[56]:

```
Text(0, 0.5, 'density')
```



In [57]:

```
# WC
wcat.Waist.mean(), wcat.Waist.median()
```

Out[57]:

```
(91.90183486238533, 90.8)
```

In [58]:

```
wcat.AT.mean(), wcat.AT.median()
```

Out[58]:

```
(101.89403669724771, 96.54)
```

In [ ]:

```

```

Assignment-1-Q22

In [59]:

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

Out[59]:

```
1.6448536269514722
```

In [60]:

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

Out[60]:

```
1.8807936081512509
```

In [62]:

```
# Z score for 60% confidence interval
stats.norm.ppf(0.80)
```

Out[62]:

```
0.8416212335729143
```

Assignment-1-Q23

In [63]:

```
# t Score of 95% confidence interval for sample size of 25
stats.t.ppf(0.975,24) # df=n-1=24
```

Out[63]:

```
2.0638985616280205
```

In [64]:

```
# t Score of 96% confidence interval for sample size of 25
stats.t.ppf(0.98,24)
```

Out[64]:

```
2.1715446760080677
```

In [65]:

```
# t Score of 99% confidence interval for sample size of 25
stats.t.ppf(0.995,24)
```

Out[65]:

```
2.796939504772804
```

In [ ]:

```

```

Assignment-1-Q24

In [66]:

```
# Assume Null Hypothesis is H0:=Avg life of bulb >=260 days
# Alternative Hypothesis is Ha:=Avg life of bulb < 260 days
```

In [67]:

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

Out[67]:

```
-0.4714045207910317
```

In [68]:

```
# Find the p(X>=260) for null hypothesis
```

In [70]:

```
# p_value=1-stats.t.cdf(abs(t_scores),df=n-1)....Using cdf function
p_value=1-stats.t.cdf(abs(-0.4714),df=17)
p_value
```

Out[70]:

```
0.32167411684460556
```

In [ ]: