In [61]:
```
import pandas as pd
import numpy as np
from scipy import stats
from scipy.stats import norm
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

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.**

In [62]:
```
df=pd.read_csv('Q7.csv')
df
```

Out[62]:

	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

	Unnamed: 0	Points	Score	Weigh
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 [63]:
```
df.describe()
```

Out[63]:

	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

Mean

In [64]:
```
df.mean()
```

Out[64]:
```
Points     3.596563
Score      3.217250
Weigh     17.848750
dtype: float64
```

Variance

In [65]:
```
df.var()
```

Out[65]:
```
Points    0.285881
Score     0.957379
Weigh     3.193166
dtype: float64
```

Standard Deviation

In [66]:
```
df.std()
```

Out[66]:
```
Points    0.534679
Score     0.978457
Weigh     1.786943
dtype: float64
```

Median

In [67]:
```
df.median()
```

Out[67]:
```
Points     3.695
Score      3.325
Weigh     17.710
dtype: float64
```

In [68]:
```
df.min()
```

Out[68]:
```
Unnamed: 0    AMC Javelin
Points              2.76
Score              1.513
Weigh               14.5
dtype: object
```

In [69]:
```
df.max()
```

Out[69]:
```
Unnamed: 0    Volvo 142E
Points              4.93
Score              5.424
Weigh               22.9
dtype: object
```

mode

In [70]:
```
df.Points.mode()
```

Out[70]:
```
0    3.07
1    3.92
Name: Points, dtype: float64
```

In [71]:
```
df.Score.mode()
```

Out[71]:
```
0    3.44
Name: Score, dtype: float64
```

In [72]:
```
df.Weigh.mode()
```

Out[72]:
```
0    17.02
1    18.90
Name: Weigh, dtype: float64
```

Range

In [73]:
```
df_range=df.Points.max()-df.Points.min()
df_range
```

Out[73]:
```
2.17
```

In [74]:
```
score_range=df.Score.max()-df.Score.min()
score_range
```

Out[74]:
```
3.9110000000000005
```

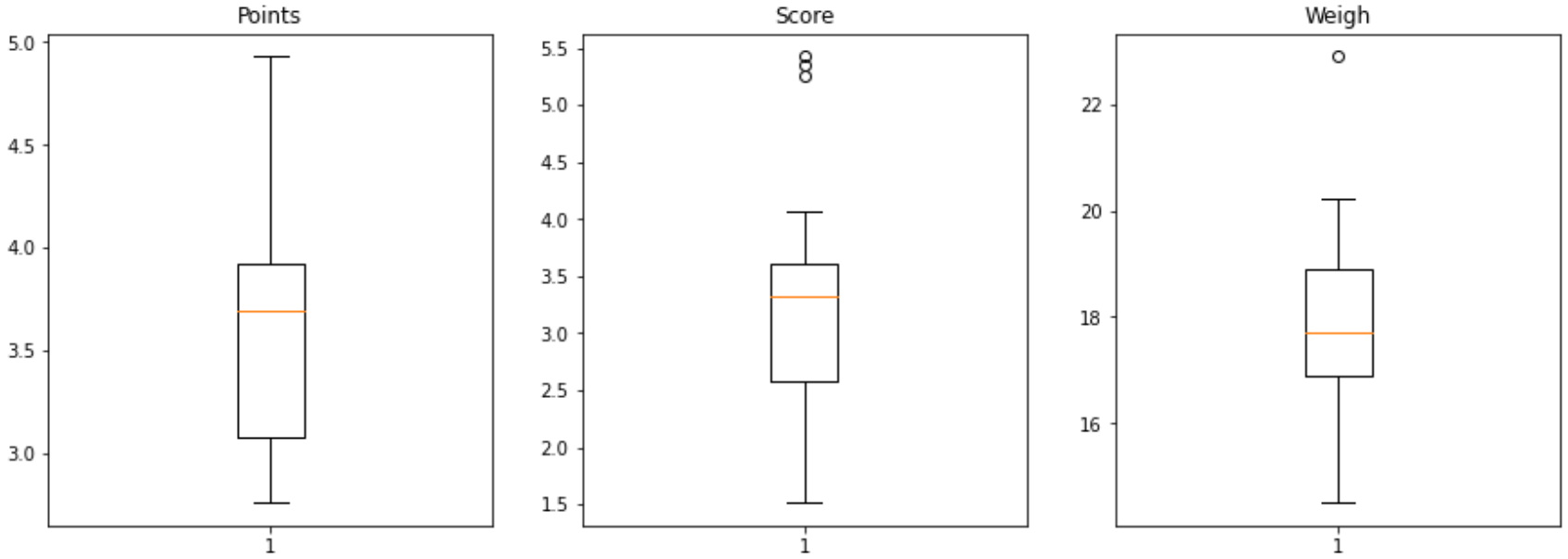
In [75]:
```
weigh_range=df.Weigh.max()-df.Weigh.min()
weigh_range
```

Out[75]:
```
8.399999999999999
```

infrances

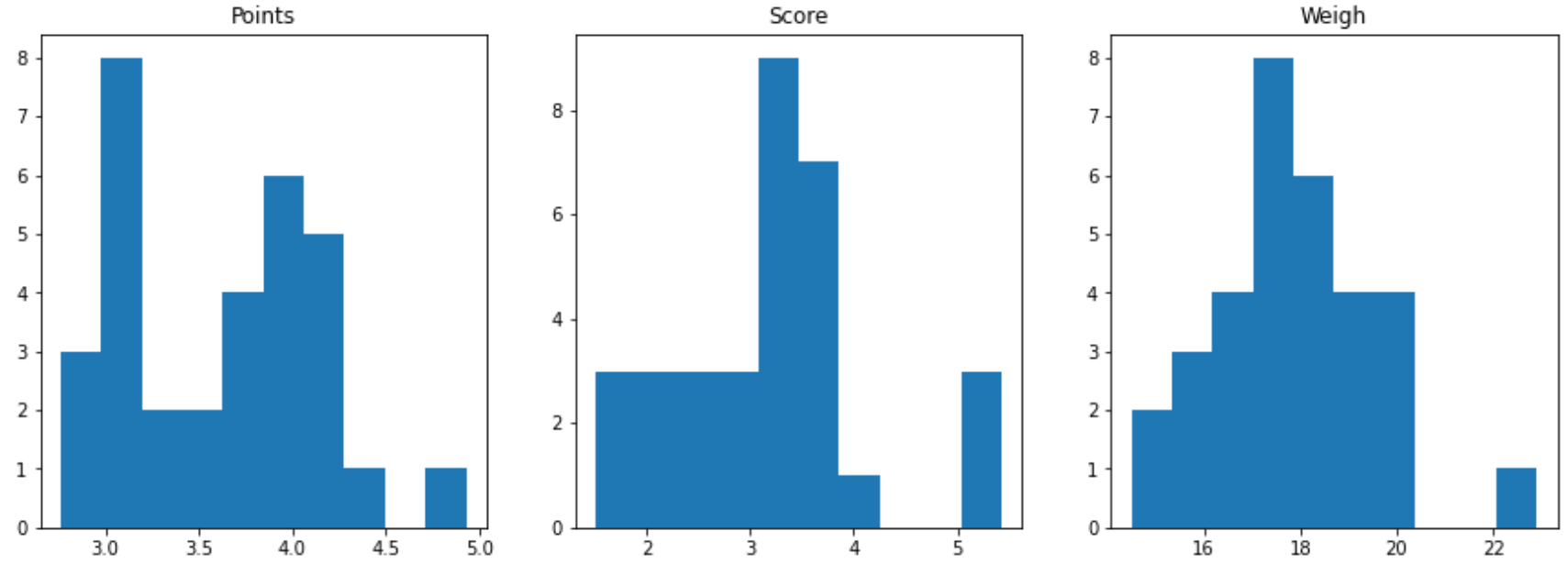
Boxplot

In [76]:
```
Fig=plt.subplots(figsize=(15,5))
plt.subplot(1,3,1)
plt.boxplot(df.Points)
plt.title('Points')
plt.subplot(1,3,2)
plt.boxplot(df.Score)
plt.title('Score')
plt.subplot(1,3,3)
plt.boxplot(df.Weigh)
plt.title('Weigh')
plt.show()
```



Histogram

In [77]:
```
Fig=plt.subplots(figsize=(15,5))
plt.subplot(1,3,1)
plt.hist(df.Points)
plt.title('Points')
plt.subplot(1,3,2)
plt.hist(df.Score)
plt.title('Score')
plt.subplot(1,3,3)
plt.hist(df.Weigh)
plt.title('Weigh')
plt.show()
```



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

- Cars speed and distance
- Use Q9_a.csv
- SP and Weight(WT)

In [78]:
```
Q_9=pd.read_csv('Q9_a.csv')
```

In [79]:
```
Q_9
```

Out[79]:

	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

	Index	speed	dist
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
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 [80]:
```
Q_9.iloc[:,1:]
```

Out[80]:

	speed	dist
0	4	2
1	4	10
2	7	4
3	7	22
4	8	16
5	9	10
6	10	18
7	10	26
8	10	34
9	11	17
10	11	28
11	12	14
12	12	20
13	12	24
14	12	28
15	13	26
16	13	34
17	13	34
18	13	46
19	14	26
20	14	36
21	14	60
22	14	80
23	15	20
24	15	26

	speed	dist
25	15	54
26	16	32
27	16	40
28	17	32
29	17	40
30	17	50
31	18	42
32	18	56
33	18	76
34	18	84
35	19	36
36	19	46
37	19	68
38	20	32
39	20	48
40	20	52
41	20	56
42	20	64
43	22	66
44	23	54
45	24	70
46	24	92
47	24	93
48	24	120
49	25	85

Skewnes

In [81]:
```
Q_9.skew()
```

Out[81]:
```
Index    0.000000
speed   -0.117510
dist     0.806895
dtype: float64
```

Kurtosis

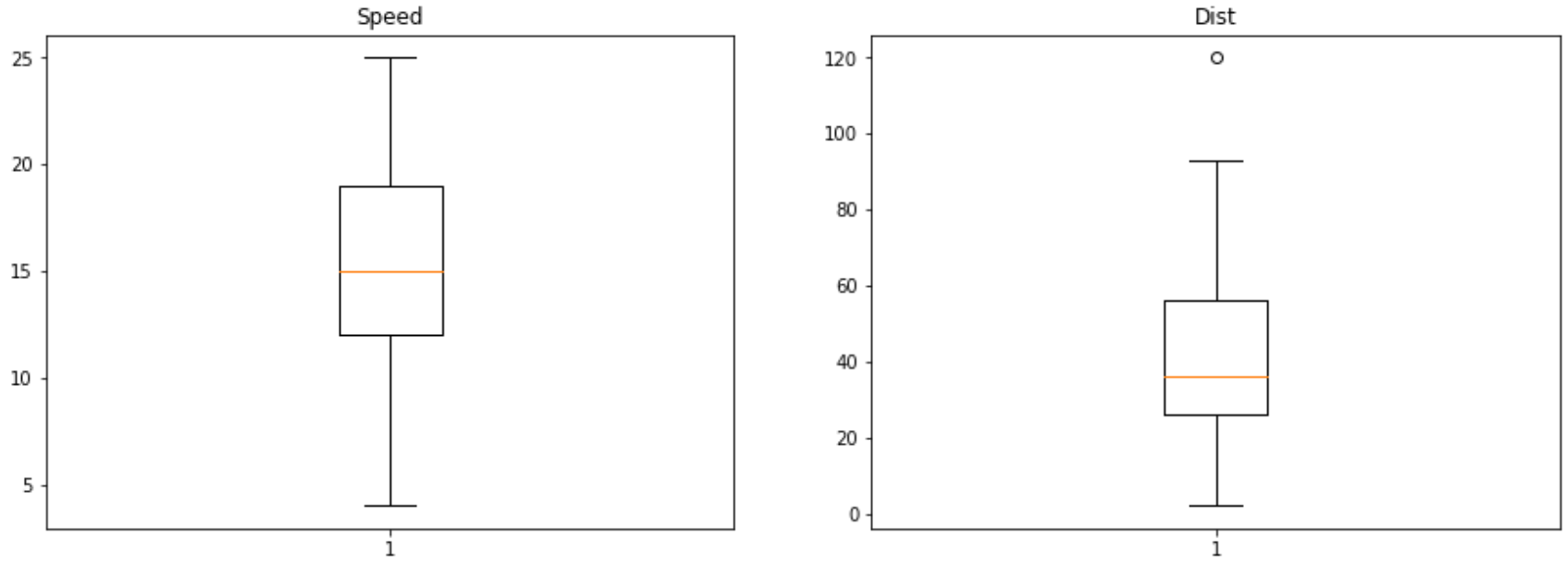
In [82]:
```
Q_9.kurt()
```

Out[82]:
```
Index   -1.200000
speed   -0.508994
dist     0.405053
dtype: float64
```

infrences

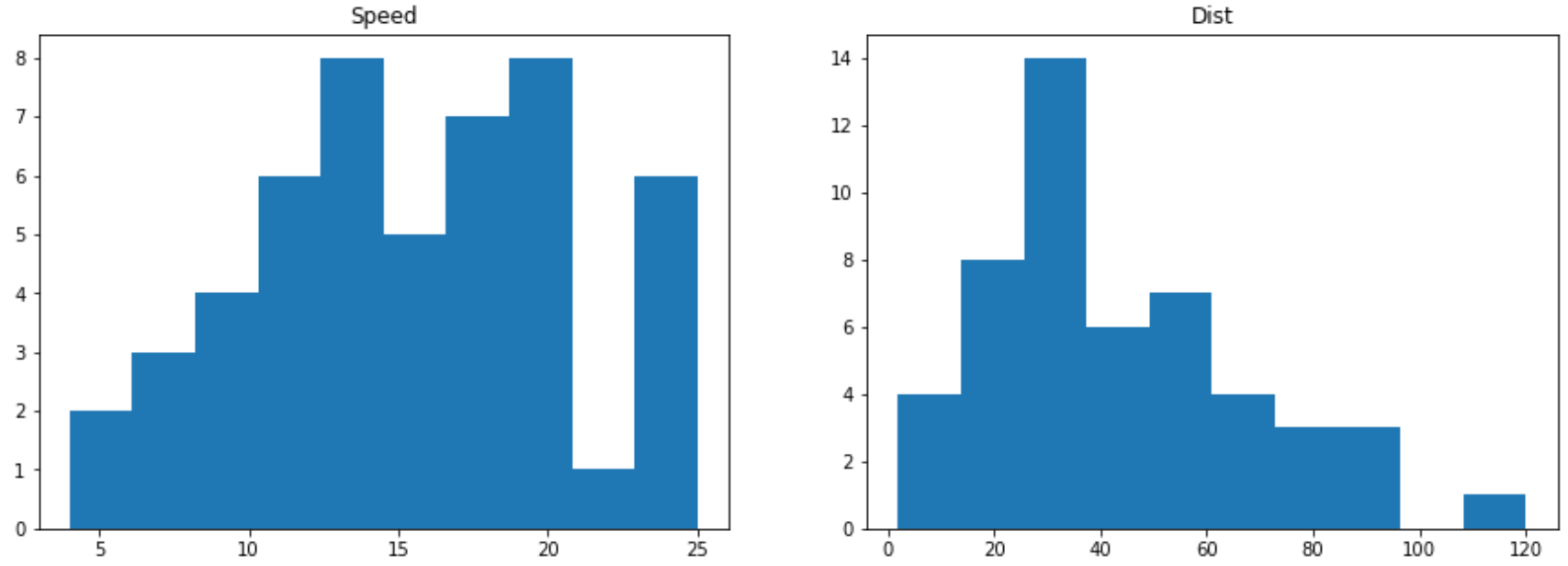
Boxplot

In [83]:
```
Fig=plt.subplots(figsize=(15,5))
plt.subplot(1,2,1)
plt.boxplot(Q_9.speed)
plt.title('Speed')
plt.subplot(1,2,2)
plt.boxplot(Q_9.dist)
plt.title('Dist')
plt.show()
```



Histogram

In [84]:
```
Fig=plt.subplots(figsize=(15,5))
plt.subplot(1,2,1)
plt.hist(Q_9.speed)
plt.title('Speed')
plt.subplot(1,2,2)
plt.hist(Q_9.dist)
plt.title('Dist')
plt.show()
```



==========

Q_9(b)

Import Dataset

In [85]:
```
Q_9b = pd.read_csv('Q9_b.csv')
Q_9b
```

Out[85]:

	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

Removing "Unnamed:0" Column

In [86]:
```
Q_9b = Q_9b.iloc[:,1:]
Q_9b
```

Out[86]:

	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

Skewness

In [87]:
```
Q_9b.skew()
```

Out[87]:
```
SP    1.611450
WT   -0.614753
dtype: float64
```

Kurtosis

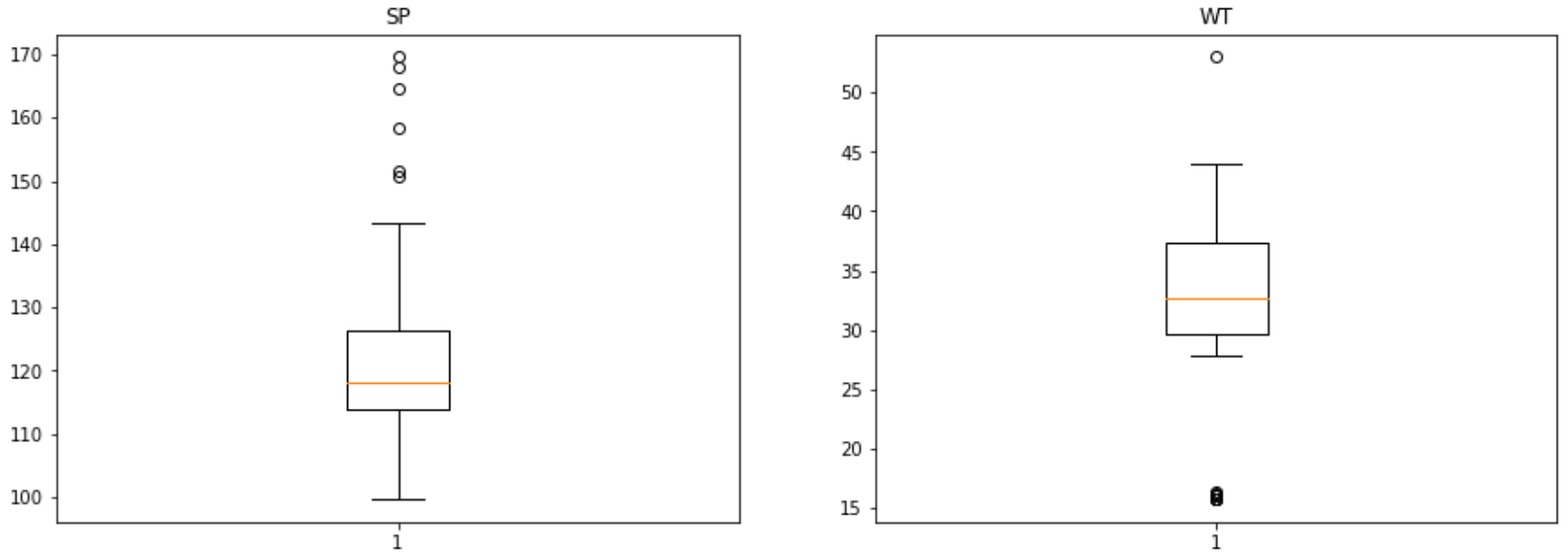
In [88]:
```
Q_9b.kurt()
```

Out[88]:
```
SP    2.977329
WT    0.950291
dtype: float64
```

Inferences

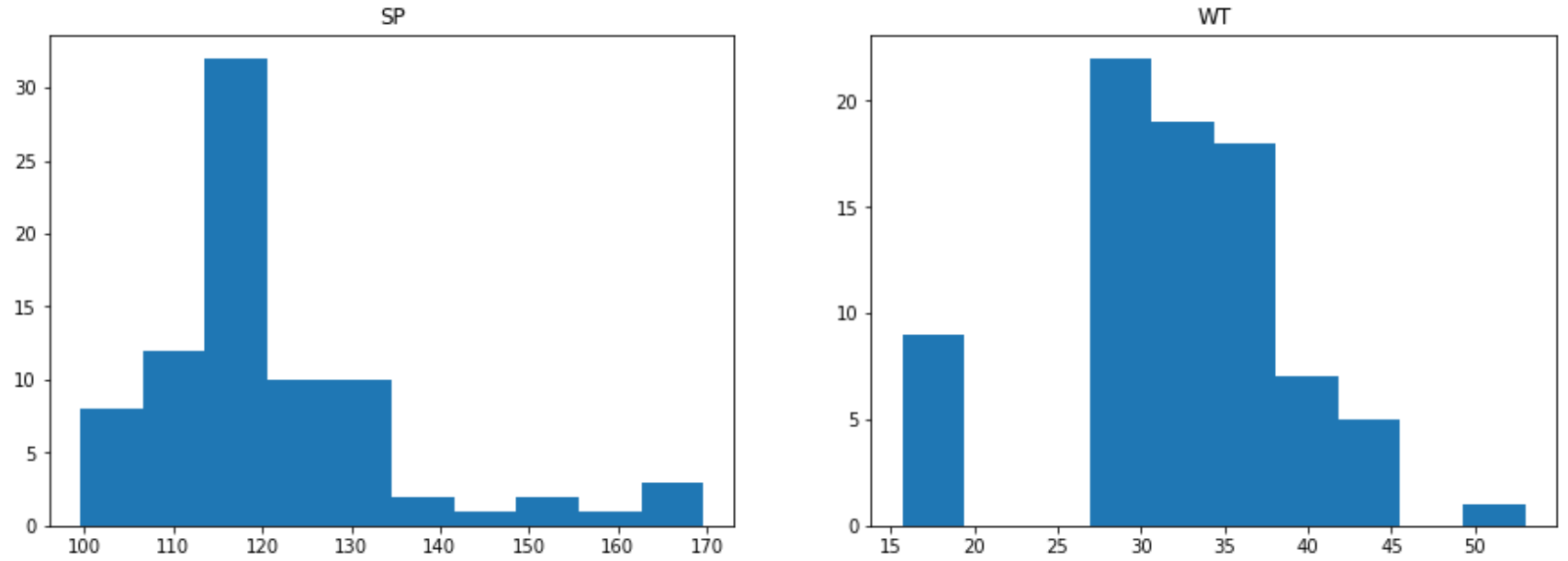
Boxplot

In [89]:
```
Fig=plt.subplots(figsize=(15,5))
plt.subplot(1,2,1)
plt.boxplot(Q_9b.SP)
plt.title('SP')
plt.subplot(1,2,2)
plt.boxplot(Q_9b.WT)
plt.title('WT')
plt.show()
```



Histogram

In [90]:
```
Fig=plt.subplots(figsize=(15,5))
plt.subplot(1,2,1)
plt.hist(Q_9b.SP)
plt.title('SP')
plt.subplot(1,2,2)
plt.hist(Q_9b.WT)
plt.title('WT')
plt.show()
```



Q_11. 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?

Avg. weight of adult in mexico with 94% ci

In [91]:
```
stats.norm.interval(0.94,200,30/(2000**0.5))
```

Out[91]:
```
(198.738325292158, 201.261674707842)
```

Avg. weight of adult in mexico with 98% ci

In [92]:
```
stats.norm.interval(0.98,200,30/(2000**0.5))
```

Out[92]:
```
(198.43943840429978, 201.56056159570022)
```

Avg. weight of adult in mexico with 96% ci

In [93]:
```
stats.norm.interval(0.96,200,30/(2000**0.5))
```

Out[93]:
```
(198.62230334813333, 201.37769665186667)
```

==========

Q_12. 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 [94]:
```
ab = pd.Series([34,36,36,38,38,39,39,40,40,41,41,41,41,42,42,45,49,56])
```

mean

In [95]:
```
ab.mean()
```

Out[95]: 41.0

Median

In [96]:
```
ab.median()
```

Out[96]: 40.5

Variance

In [97]:
```
ab.var()
```

Out[97]: 25.529411764705884

Standard Deviation

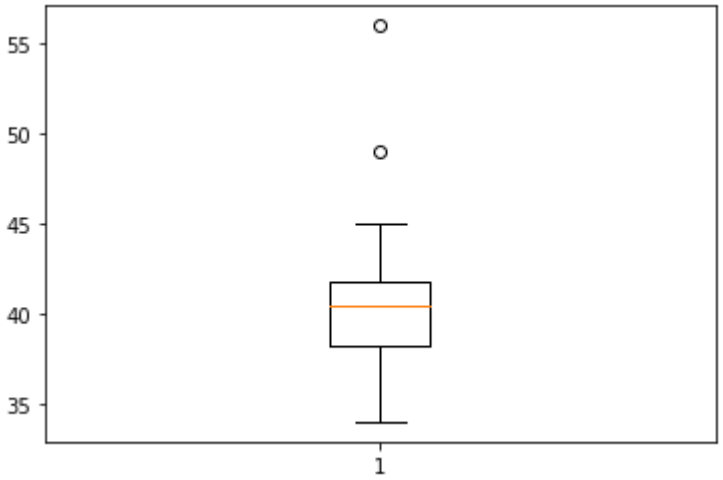
In [98]:
```
ab.std()
```

Out[98]:
```
5.05266382858645
```

Inferences

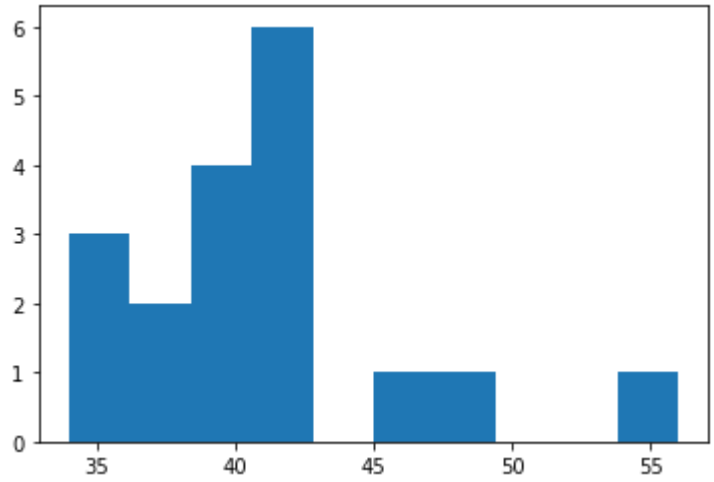
Boxplot

In [99]:
```
plt.boxplot(ab)
plt.show()
```



Histogram

In [100]:
```
plt.hist(ab,label='ab')
plt.show()
```



==========

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)

b. P(MPG<40)

c. P (20<MPG<50)

import Dataset

In [105]:
```
Cars = pd.read_csv('Cars (1).csv')
Cars
```

Out[105]:

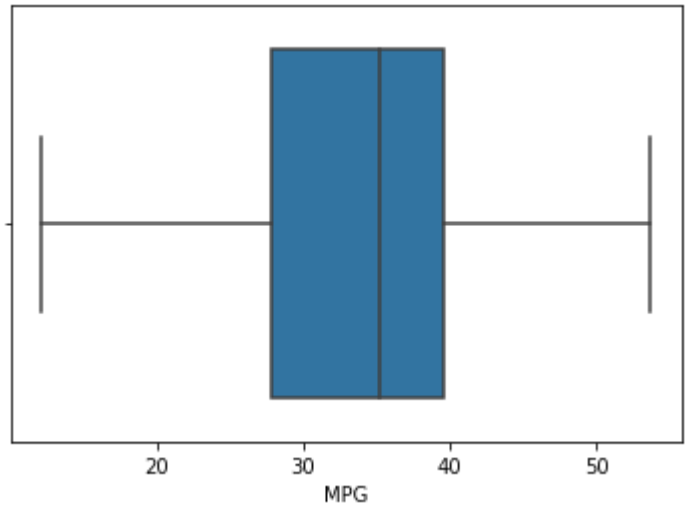
	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149
...	...	...	...	...	...
76	322	36.900000	50	169.598513	16.132947
77	238	19.197888	115	150.576579	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

Boxplot

In [107]:
```
sns.boxplot(Cars.MPG)
```

Out[107]:
```
<AxesSubplot:xlabel='MPG'>
```



A.P(MPG>38)

In [108]:
```
1-stats.norm.cdf(38,Cars.MPG.mean(),Cars.MPG.std())
```

Out[108]:
```
0.3475939251582705
```

B. P(MPG<40)

In [109]:
```
1-stats.norm.cdf(40,Cars.MPG.mean(),Cars.MPG.std())
```

Out[109]:
```
0.27065012378483844
```

C. P (20<MPG<50)

In [110]:
```
stats.norm.cdf(0.50,Cars.MPG.mean(),Cars.MPG.std())-stats.norm.cdf(0.20,Cars.MPG.mean(),Cars.MPG.std())
```

Out[110]:
```
1.2430968797327613e-05
```

==========

Q_21.Check whether the data follows normal distribution¶

a) Check whether the MPG of Cars follows Normal Distribution

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

Q 21(a)

Import Important Libraries and Warnings

import Dataset

In [111]:
```
Cars = pd.read_csv('Cars (1).csv')
Cars
```

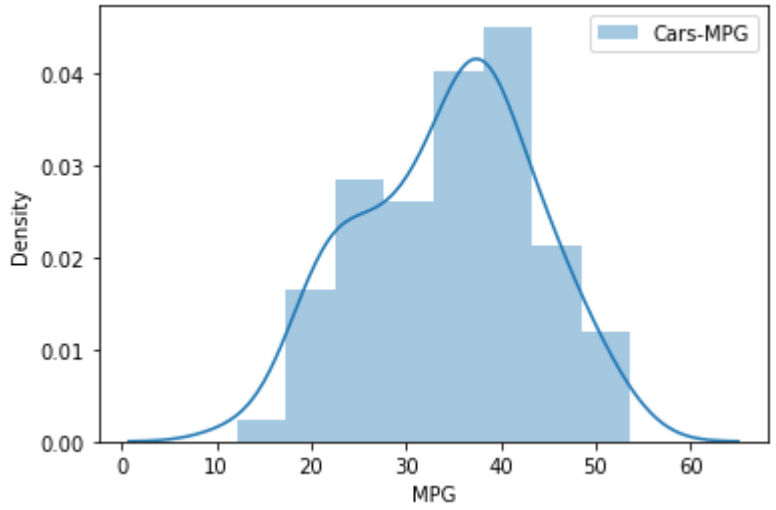
Out[111]:

	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

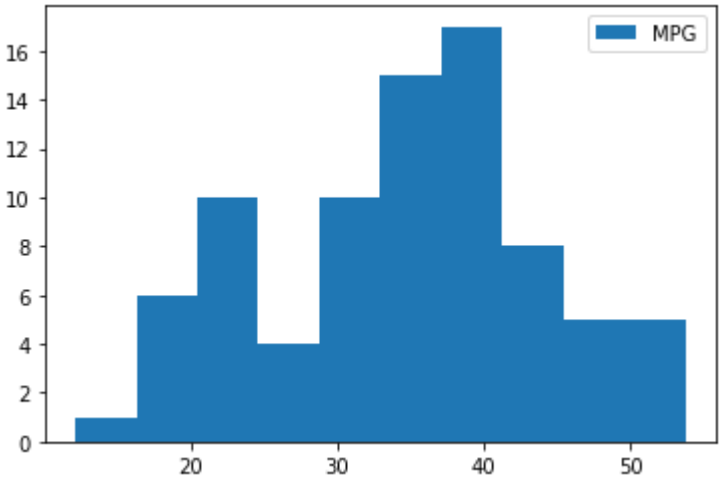
plotting Distribution for car-MPG

In [112]:
```
sns.distplot(Cars.MPG, label='Cars-MPG')
plt.xlabel('MPG')
plt.ylabel('Density')
plt.legend();
```



Histogram

In [113]:
```
plt.hist(Cars.MPG, label='MPG')
plt.legend();
```



mean

In [115]:
```
Cars.MPG.mean()
```

Out[115]:
```
34.422075728024666
```

Median

In [116]:
```
Cars.MPG.median()
```

Out[116]:
```
35.15272697
```

Q 21(b)

import Dataset

In [117]:
```
wcat = pd.read_csv('wc-at.csv')
wcat
```

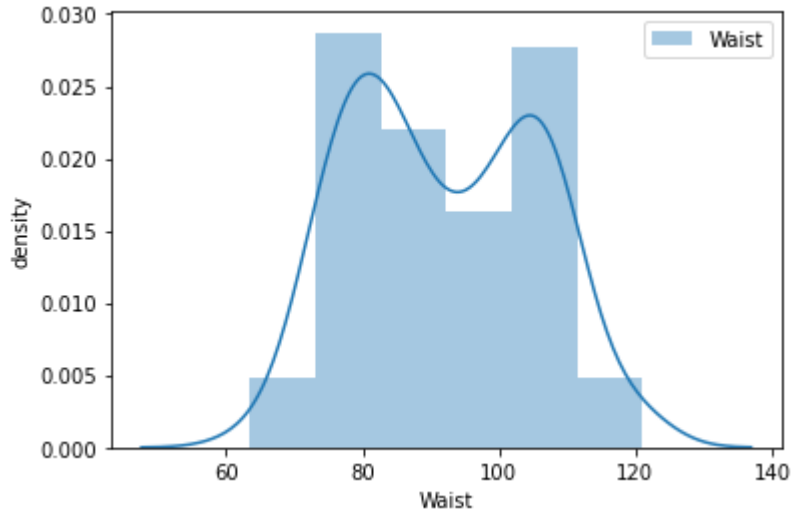
Out[117]:

	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

Plotting Distribution for Waist Circumference (Waist)

In [118]:
```
sns.distplot(wcat.Waist, label='Waist')
plt.ylabel('density')
plt.legend();
```



Mean

In [119]:
```
wcat.Waist.mean()
```

Out[119]: 91.90183486238533

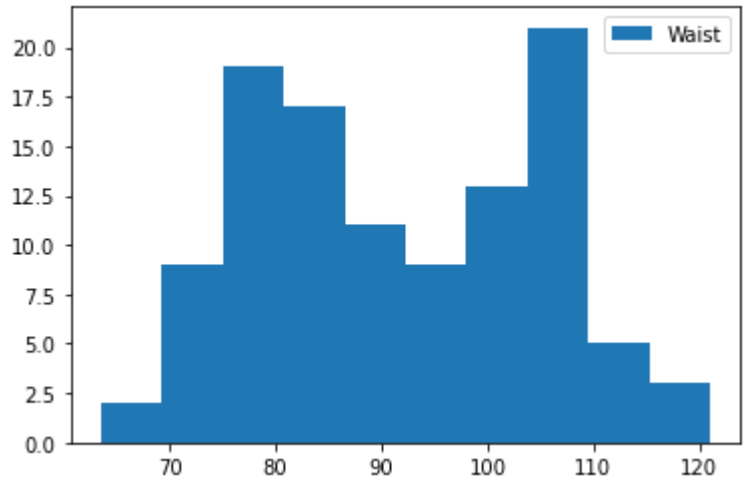
Median

In [120]:
```
wcat.Waist.median()
```

Out[120]: 90.8

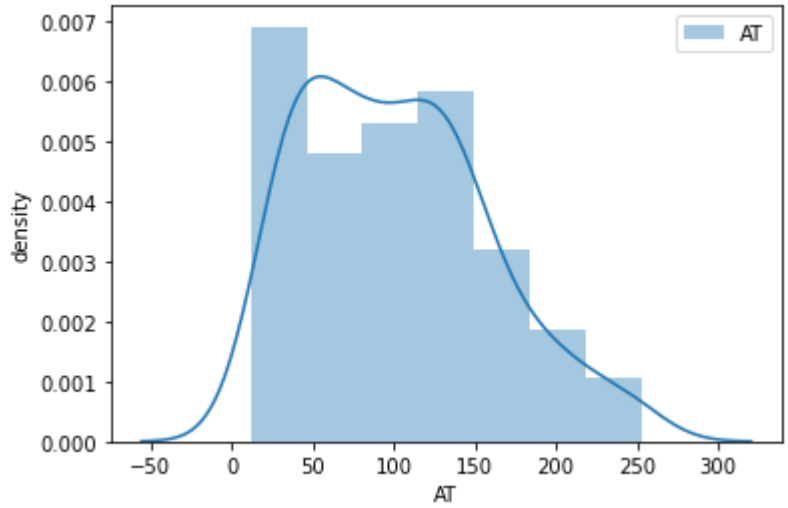
Histogram

In [121]:
```
plt.hist(wcat.Waist,label='Waist')
plt.legend();
```



Plotting Distribution for Adipose Tissue (AT)

In [122]:
```
sns.distplot(wcat.AT, label='AT')
plt.ylabel('density')
plt.legend();
```



Mean

In [123]:
```
wcat.AT.mean()
```

Out[123]:
```
101.89403669724771
```

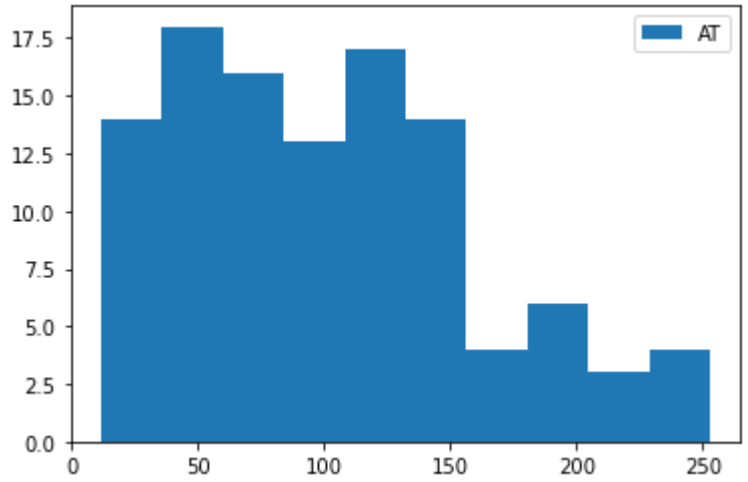
Median

In [125]:
```
wcat.AT.median()
```

Out[125]:
```
96.54
```

Histogram

In [126]:
```
plt.hist(wcat.AT,label='AT')
plt.legend();
```



==========

Q_22.Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval

In [127]:
```
# Area Left(AL) = (1+CI) / 2
```

In [128]:
```
a = (1+.90)/2
a
```

Out[128]:
```
0.95
```

In [129]:
```
b = (1+.94)/2
b
```

Out[129]: 0.97

In [130]:
```
c = (1+.6)/2
c
```

Out[130]: 0.8

Z-scores of 90% confidence interval

In [131]:
```
stats.norm.ppf(.97)
```

Out[131]: 1.8807936081512509

Z-scores of 60% confidence interval

In [132]:
```
stats.norm.ppf(0.8)
```

Out[132]: 0.8416212335729143

==========

Q_23. Calculate the Z scores of 90% confidence interval,94% confidence interval, 60% confidence interval

t scores of 95% confidence interval for sample size of 25

In [133]:
```
stats.t.ppf(.975,24)
```

Out[133]: 2.0638985616280205

t scores of 96% confidence interval for sample size of 2

In [134]:
```
stats.t.ppf(.98,24)
```

Out[134]: 2.1715446760080677

t scores of 99% confidence interval for sample size of 2

In [135]:
```
stats.t.ppf(.995,24)
```

Out[135]: 2.796939504772804

==========

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

Assume Null Hypothesis is: Ho = Avg life of Bulb >= 260 days

Alternate Hypothesis is: H1 = Avg life of Bulb < 260 days

find t-scores at x=260; t=(s_mean-P_mean)/(s_SD/sqrt(n))

In [136]:
```
t=(260-270)/(90/18**0.5)
t
```

Out[136]:
```
-0.4714045207910317
```

Find P(X>=260) for null hypothesis

P_Value = 1-stats.t.cdf(abs(t_scores), df = n-1)... Using cdf function

In [137]:
```
P_Value = 1-stats.t.cdf(abs(-.4714),df=17)
P_Value
```

Out[137]:
```
0.32167411684460556
```

OR P_Value = stats.t.sf(abs(t_score), df = n-1)... Using sf function

In [138]:
```
P_Value=stats.t.sf(abs(-0.4714),df=17)
P_Value
```

Out[138]:
```
0.32167411684460556
```

==========

In [ ]:
```

```