In [ ]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import skew, kurtosis, spearmanr
```

1. Determine the central tendency of the below Population:

-993, -23,18,-2,-6,98,45,32,-45,843,1024,-256

In [ ]:

```
a = [-993, -23,18,-2,-6,98,45,32,-45,843,1024,-256]
print(np.mean(a))
```

```
61.25
```

2. Calculate the Standard Deviation and Variance of the below sample

In [ ]:

```
a = [-99, -2,18,-23,-61,1,982,45,32,-45]
print("standard deviation:", np.std(a,ddof=1))
print("variance:",np.var(a,ddof=1))
```

```
standard deviation: 318.30586268905296
variance: 101318.62222222223
```

3. You have 8 numbers. The mean is 6. You add 5 to each number in the group. What is the new mean?

In [ ]:

```
6 + 5
```

Out[ ]:

```
11
```

4. You have 15 numbers. The mean is 10, and the variance is 4. You multiply each number by 3. What is the new standard deviation?

In [ ]:

```
print("The new standard deviation is:", 3*2)
```

```
The new standard deviation is: 6
```

5. Temperature of 5 cities are given, from the given values, what would be the mean and standard deviation of temperature in Celsius?

(Hint: Celsius = 0.556F - 17.778)

City	Degrees Fahrenheit
Delhi	82
Bangalore	77

City	Degrees Fahrenheit
Coorg	41
Coimbatore	78
Chennai	84

In [ ]:

```
temps = {"City":["Delhi","Bangalore","Coorg","Coimbatore","Chennai"], "Temperature (F)":
[82,77,41,78,84]}
df = pd.DataFrame(temps,columns=["City", "Temperature (F)"])
def celsius(F):
    return 0.556*F - 17.778

df
```

Out[ ]:

	City	Temperature (F)
0	Delhi	82
1	Bangalore	77
2	Coorg	41
3	Coimbatore	78
4	Chennai	84

In [ ]:

```
df["Temperature (C)"] = df['Temperature (F)'].apply(celsius)
df
```

Out[ ]:

	City	Temperature (F)	Temperature (C)
0	Delhi	82	27.814
1	Bangalore	77	25.034
2	Coorg	41	5.018
3	Coimbatore	78	25.590
4	Chennai	84	28.926

In [ ]:

```
print("Mean of the temperature in celsius:", np.mean(df["Temperature (C)"]))
print("Standard deviation of the temperature in celsius:",np.std(df["Temperature (C)"]))
```

```
Mean of the temperature in celsius: 22.476400000000005
Standard deviation of the temperature in celsius: 8.84442046942591
```

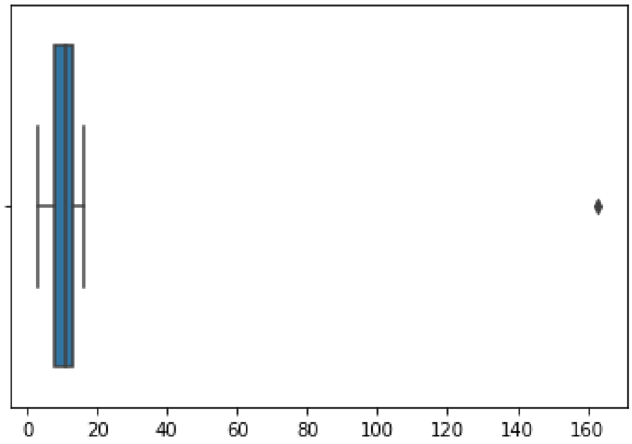
5. Construct a boxplot for the following data set.

3, 5, 8, 8, 9, 11, 12, 12, 13, 13, 163,5,8,8,9,11,12,12,13,13,16

In [ ]:

```
a = [3, 5, 8, 8, 9, 11, 12, 12, 13, 13, 163,5,8,8,9,11,12,12,13,13,16]
sns.boxplot(a)
plt.show()
```

```
/usr/local/lib/python3.6/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the
following variable as a keyword arg: x. From version 0.12, the only valid positional argu
ment will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  FutureWarning
```



Contains an outlier 163 which can be clearly seen in the above plot.

7. Consider below dataset, calculate the skewness and then tell if it is left skewed or right skewed?

12, 13, 54, 56 , 25

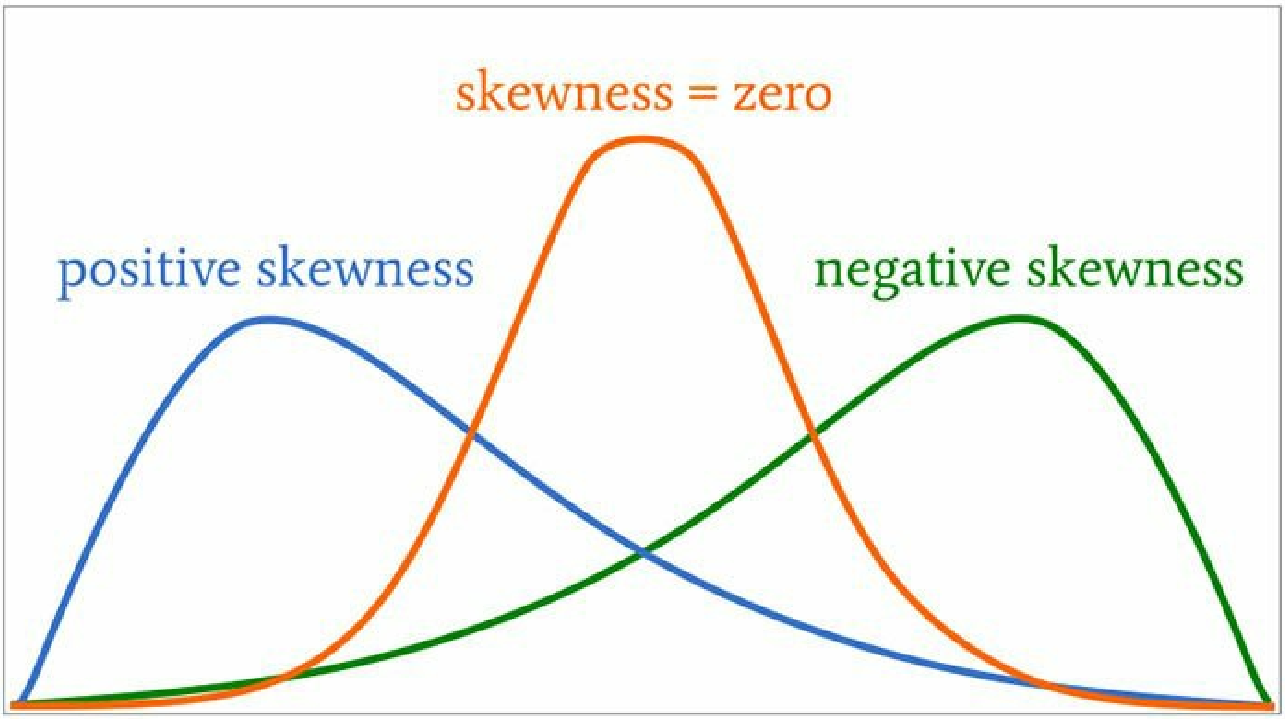
In [ ]:

```
skew([12, 13, 54, 56 , 25])
```

Out[ ]:

```
0.2563317051472635
```

The given data is positively skewed or the distribution leans towards the left similar to the blue plot in the image below:



8. Determine the excess-kurtosis of the below dataset and then categorize it according to the type of kurtosis it is and give its characteristics.

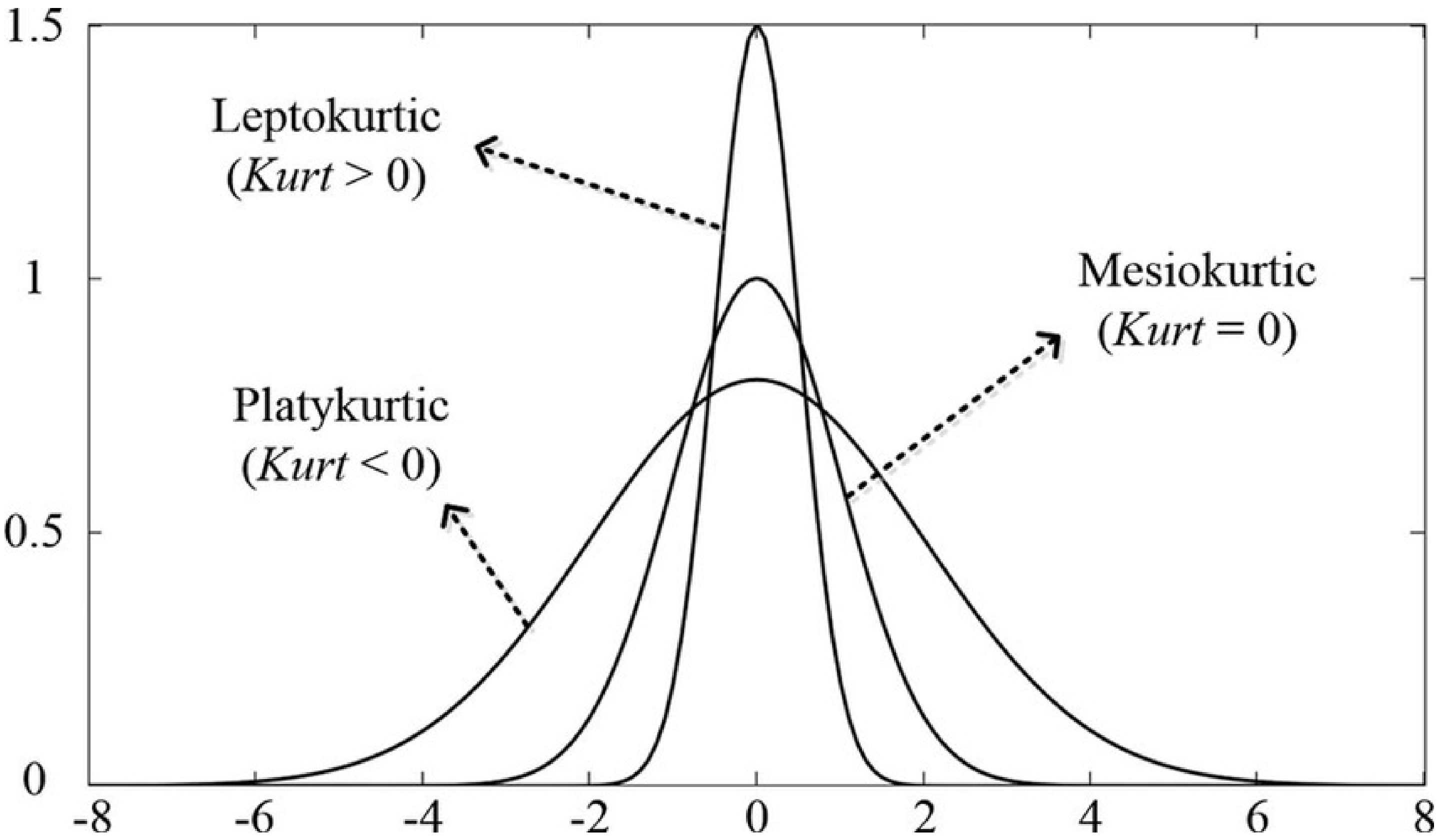
12, 13, 54, 56 , 25

In [ ]:

```
b = [12,   13,   54,   56 ,  25]
print(kurtosis(b))
```

```
-1.7721210214761647
```

A high positive value indicates a peaked, or leptokurtic, curve. A high negative value indicates a flattened, or Platykurtic, curve.



In this case the kurtosis value is less than zero indicating flattened curve. This indicates that there is *more probability mass in the tails than a normal distribution*. Or in other words a distribution with a negative kurtosis value indicates that the distribution has lighter tails than the normal distribution.

9. Determine the outliers in the below dataset using IQR formula.

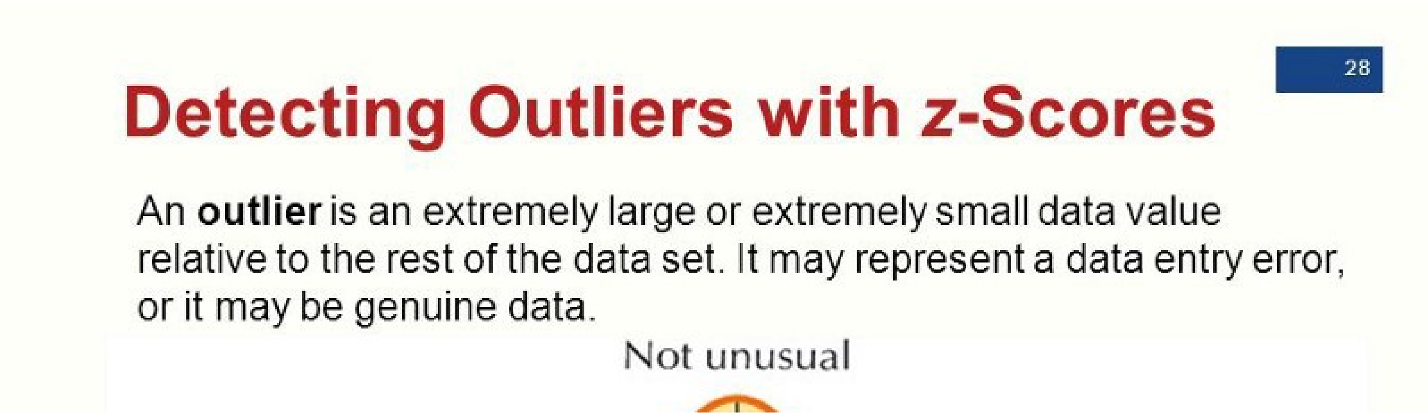
1, 99, 100, 101, 103, 109, 110, 201

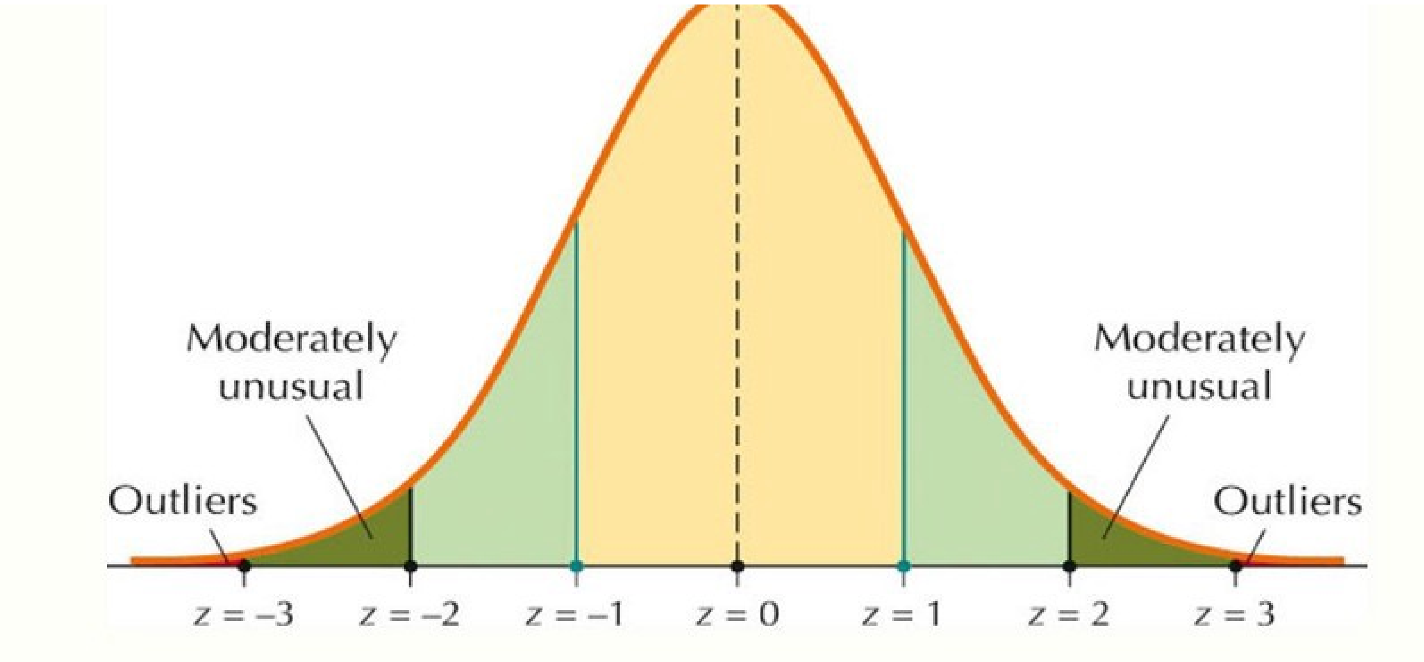
In [ ]:

```
b = [1, 99, 100, 101, 103, 109, 110, 201]
q1,q2 = np.percentile(b, [25,75])
iqr = q2-q1
print("interquartile range of the given data is:",iqr)
```

```
interquartile range of the given data is: 9.5
```

10. Determine the outlier in the below dataset using Z-score.





Considering z_score of 3 as outlier from the above figure

In [ ]:

```
z = np.array([1.5895, 1.6508, 1.7131, 1.7136,1.7212, 1.7296, 1.7343, 1.7663, 1.8018, 1.8
394, 1.8869, 1.9357, 1.9482, 2.1038, 10.8135, -0.0012])
print(z[abs(z)>=3])
```

```
[10.8135]
```

11. Below is the mark obtained by some students. Construct a bar chart for it:

In [ ]:

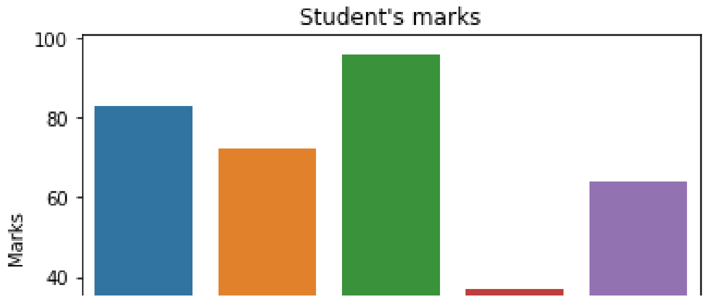
```
a = {"name":["Rohan","Bhavya","Sri","Riteish","Neha"], "Marks":[83,72,96,37,64]}
data = pd.DataFrame(a)
data
```

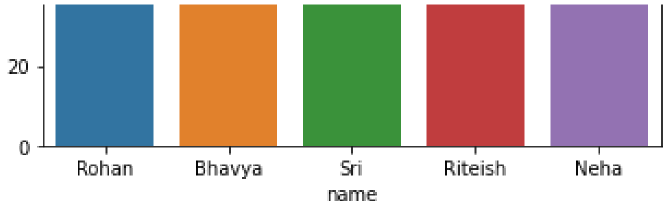
Out[ ]:

	name	Marks
0	Rohan	83
1	Bhavya	72
2	Sri	96
3	Riteish	37
4	Neha	64

In [ ]:

```
sns.barplot(x="name",y="Marks",data=data)
plt.title("Student's marks")
plt.show()
```





12. Covariance

Below are some observations obtained from a hospital and shows glucose level of some patients. Check the correlation of the variables and then tell if it has positive, negative or no relation between them. Calculate co-variance and also plot a scatter-plot to see the relation visually.

Covariance is a statistical tool that is used to determine the relationship between the movement of two variables

it is denoted by:

$$cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

when two variables are propotionate then they show a positive covariance value and vice versa.

In [ ]:

```
def covariance(x,y):
    '''
    Returns the covariance between two given variables
    '''
    if (len(x)==len(y)) and (len(x)>1 and len(y)>1):
        xm = np.mean(x)
        ym = np.mean(y)
        sum = 0
        for i in range(len(x)):
            sum += (x[i]-xm)*(y[i]-ym)
        return sum/(len(x)-1)
    else:
        return None
```

In [ ]:

```
a = {"patient":["A","B","C","D","E","F"], "Age (years)":[46,24,28,42,59,48], "Glucose Le
vel":[99,65,78,79,89,82]}
df = pd.DataFrame(a)
df
```

Out[ ]:

	patient	Age (years)	Glucose Level
0	A	46	99
1	B	24	65
2	C	28	78
3	D	42	79
4	E	59	89
5	F	48	82

In [ ]:

```
covariance(df['Age (years)'], df["Glucose Level"])
```

Out[ ]:

```
109.8
```

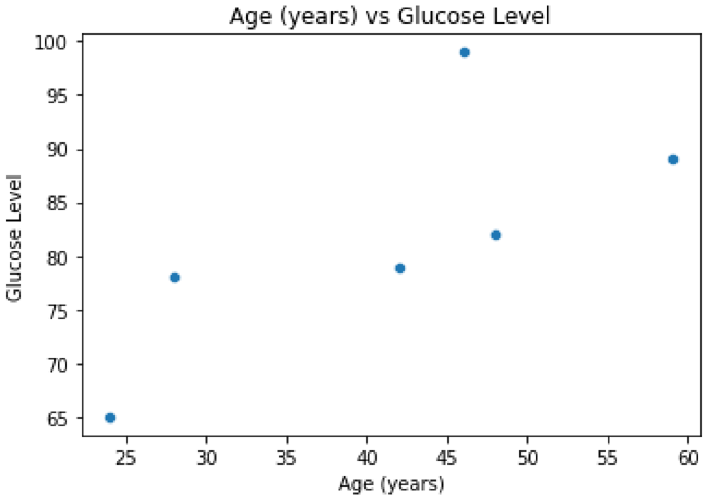
Since the covariance is positive value, this indicates that Age and Glucose level have a direct proportionality relationship

relationship.

We can visualize this relation in scatterplot between the two variables:

In [ ]:

```
sns.scatterplot(x="Age (years)", y="Glucose Level", data=df)
plt.title("Age (years) vs Glucose Level")
plt.show()
```



13. Correlation

Try to find out if there is any correlation between Physical Activity and Blood Pressure. Calculate Spearman Rank Correlation.

In [ ]:

```
a = {"Name":["Alan","Carl","David","Don","John","Matt","Mike","Neal","Rick","Rob"],
     "Physical activity (min)":[60,55,25,50,40,45,35,10,30,20],
     "Blood pressure (mm Hg)":[118,117,120,121,119,122,123,124,125,126]}
df = pd.DataFrame(a)
df
```

Out[ ]:

	Name	Physical activity (min)	Blood pressure (mm Hg)
0	Alan	60	118
1	Carl	55	117
2	David	25	120
3	Don	50	121
4	John	40	119
5	Matt	45	122
6	Mike	35	123
7	Neal	10	124
8	Rick	30	125
9	Rob	20	126

In [ ]:

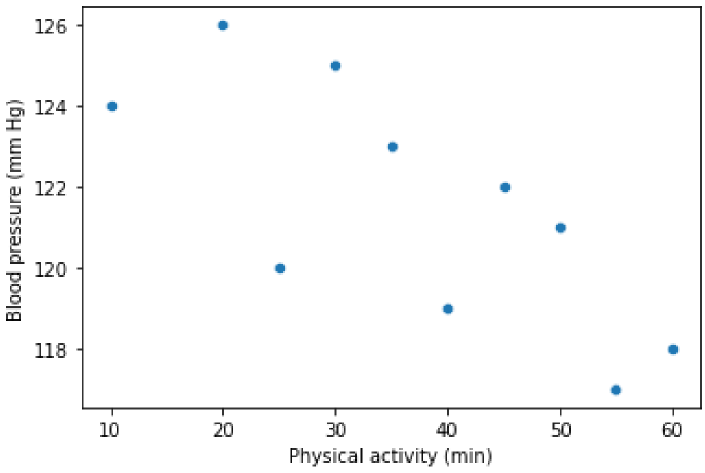
```
spearmanr(a=df['Physical activity (min)'], b=df['Blood pressure (mm Hg)'])
```

Out[ ]:

```
SpearmanrResult(correlation=-0.7575757575757575, pvalue=0.011143446799694208)
```

In [ ]:

```
sns.scatterplot(x='Physical activity (min)', y="Blood pressure (mm Hg)", data=df)
plt.title('Physical activity vs Blood pressure")
plt.show()
```



Shows a negative spearman correlation, indicating that there is a negative relation between the two.

Also the correlation value is 0.75 stating there is a strong negative correlation as per the image below: