

```
1 import numpy as np
```

## DATA SET

```
1 a = np.array([7,8,888,1,2,3,4,5,6])
2 a
```

```
array([ 7,  8, 888,  1,  2,  3,  4,  5,  6])
```

```
1 asort=np.sort(a)
2 asort
```

```
array([ 1,  2,  3,  4,  5,  6,  7,  8, 888])
```

## MEASURE OF CENTRALITY

```
1 aMean = np.mean(asort)
2 aMean
```

```
102.66666666666667
```

```
1 aMedian = np.median(asort)
2 aMedian
```

```
5.0
```

```
1 from scipy import stats as st
2 aMode = st.mode(asort)
3 aMode
```

```
ModeResult(mode=array([1]), count=array([1]))
```

here we don't have any repeated value  
so showed  
it element = 1

## MEASURE OF SPREAD

```
1 aMax = np.max(asort)
2 aMax
```

```
888
```

```
1 aMin = np.min(asort)
2 aMin
```

```
1
```

```
1 aRange = aMax - aMin
2 aRange
```

887

## QUARTILES

```
1 aIQR0 = np.percentile(asort,0)
2 aIQR0
```

1.0

```
1 aIQR1 = np.percentile(asort,25)
2 aIQR1
```

3.0

```
1 aIQR2 = np.percentile(asort,50)
2 aIQR2
```

5.0

```
1 aIQR3 = np.percentile(asort,75)
2 aIQR3
```

7.0

```
1 aIQR4 = np.percentile(asort,100)
2 aIQR4
```

888.0

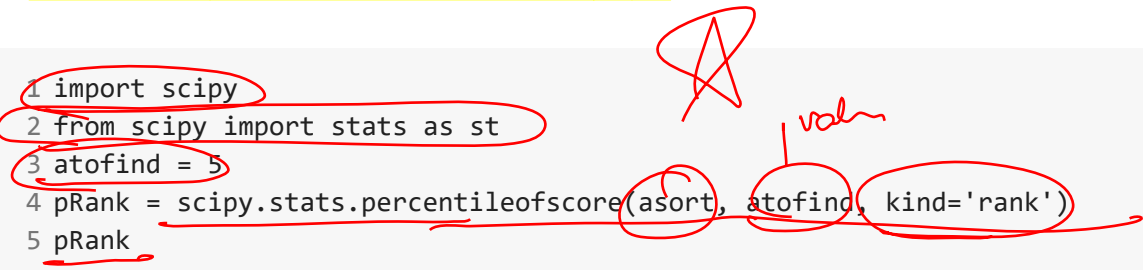
## INTER QUARTILE RANGE

```
1 IQR = aIQR3 - aIQR1
2 IQR
```

4.0

## FIND PERCENTILE RANK FOR A GIVEN VALUE

```
1 import scipy
2 from scipy import stats as st
3 atofind = 5
4 pRank = scipy.stats.percentileofscore(asort, atofind, kind='rank')
5 pRank
```



55.55555555555556

**OUTLIER**

```
1 aOutlier = awoOutlier = np.array([e for e in asort if (e < -1.5*aIQR1 or e > 1.5*aIQR3)])
2 aOutlier
```

array([888])

**ARRAY WITHOUT OUTLIERS**

```
1 awoOutlier = np.array([e for e in asort if (-1.5*aIQR1 < e < 1.5*aIQR3)])
2 awoOutlier
```

array([1, 2, 3, 4, 5, 6, 7, 8])

**Trimmed Mean**

```
1 aTrimMean = np.mean(awoOutlier)
2 aTrimMean
```

4.5

**Xi - Xbar**

array - number.

```
1 xi_xbar = asort - aMean
2 xi_xbar
```

array([-101.66666667, -100.66666667, -99.66666667, -98.66666667,  
-97.66666667, -96.66666667, -95.66666667, -94.66666667,  
785.33333333])

**VARIANCE**

```
1 avar=np.var(asort)
2 avar
```

77098.22222222222

**STANDARD DEVIATION**

```
1 astd = np.std(asort)
2 astd
```

277.66566626470444

**Z SCORE**

$$= \frac{x_i - \bar{x}}{\text{std.}}$$

```
1 zscore = (a-np.mean(a))/np.std(a)
2 zscore
```

```
array([-0.34453906, -0.3409376 ,  2.82834152, -0.36614778, -0.36254632,
       -0.35894487, -0.35534342, -0.35174196, -0.34814051])
```

## SCALE AND TRANSFORMATION

```
1 x=3
2 y=5
3 ast = (x*a)+y
4 ast
```

```
array([ 26,  29, 2669,   8,  11,  14,  17,  20,  23])
```

✓ 0s completed at 8:39 PM

