```
import numpy as np
```

DATA SET

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

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

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

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

MEASURE OF CENTRALITY

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

```
102.66666666666667
```

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

```
5.0
```

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

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

here we' don't have any repeated vale

So showed

1st element = 1

MEASURE OF SPREAD

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

```
888
```

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

```
1
```

```
aRange = aMax - aMin
aRange
```

```
887
```

QUARTILES

```python
aIQR0 = np.percentile(asort,0)
aIQR0
```

```
1.0
```

```python
aIQR1 = np.percentile(asort,25)
aIQR1
```

```
3.0
```

```python
aIQR2 = np.percentile(asort,50)
aIQR2
```

```
5.0
```

```python
aIQR3 = np.percentile(asort,75)
aIQR3
```

```
7.0
```

```python
aIQR4 = np.percentile(asort,100)
aIQR4
```

```
888.0
```

INTER QUARTILE RANGE

```python
IQR = aIQR3 - aIQR1
IQR
```

```
4.0
```

FIND PERCENTILE RANK FOR A GIVEN VALUE

A

val

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

```
55.55555555555556
```

OUTLIER

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

```
array([888])
```

ARRAY WITHOUT OUTLIERS

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

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

Trimmed Mean

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

```
4.5
```

Xi - Xbar

array – number

```
xi_xbar = asort - aMean
xi_xbar
```

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

VARIANCE

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

```
77098.22222222222
```

STANDARD DEVIATION

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

```
277.66566626470444
```

Z SCORE

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

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

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

SCALE AND TRANSFORMATION

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

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

✓ 0s completed at 8:39 PM ● ×