Mean, Median, and Mode

What can we learn from looking at a group of numbers?

In Machine Learning (and in mathematics) there are often three values that interests us:

- **Mean** - The average value
- **Median** - The mid point value
- **Mode** - The most common value

Example: We have registered the speed of 13 cars:

```
speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]
```

What is the average, the middle, or the most common speed value?

Mean

The mean value is the average value.

To calculate the mean, find the sum of all values, and divide the sum by the number of values:

```
(99+86+87+88+111+86+103+87+94+78+77+85+86) / 13 = 89.77
```

The NumPy module has a method for this. Learn about the NumPy module in our NumPy Tutorial.

Example

Use the NumPy `mean()` method to find the average speed:

```
import numpy

speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]

x = numpy.mean(speed)

print(x)
```

Median

The median value is the value in the middle, after you have sorted all the values:

```
77, 78, 85, 86, 86, 86, 87, 87, 88, 94, 99, 103, 111
```

It is important that the numbers are sorted before you can find the median.

The NumPy module has a method for this:

Example

Use the NumPy `median()` method to find the middle value:

```
import numpy

speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]

x = numpy.median(speed)

print(x)
```

If there are two numbers in the middle, divide the sum of those numbers by two.

```
77, 78, 85, 86, 86, 86, 87, 87, 94, 98, 99, 103

(86 + 87) / 2 = 86.5
```

Example

Using the NumPy module:

```
import numpy

speed = [99,86,87,88,86,103,87,94,78,77,85,86]

x = numpy.median(speed)

print(x)
```

Mode

The Mode value is the value that appears the most number of times:

```
99, 86, 87, 88, 111, 86, 103, 87, 94, 78, 77, 85, 86 = 86
```

The SciPy module has a method for this. Learn about the SciPy module in our SciPy Tutorial.

Example

Use the SciPy `mode()` method to find the number that appears the most:

```
from scipy import stats

speed = [99,86,87,88,111,86,103,87,94,78,77,85,86]

x = stats.mode(speed)

print(x)
```

What is Standard Deviation?

Standard deviation is a number that describes how spread out the values are.

A low standard deviation means that most of the numbers are close to the mean (average) value.

A high standard deviation means that the values are spread out over a wider range.

Example: This time we have registered the speed of 7 cars:

```
speed = [86,87,88,86,87,85,86]
```

The standard deviation is:

```
0.9
```

Meaning that most of the values are within the range of 0.9 from the mean value, which is 86.4.

Let us do the same with a selection of numbers with a wider range:

```
speed = [32,111,138,28,59,77,97]
```

The standard deviation is:

```
37.85
```

Meaning that most of the values are within the range of 37.85 from the mean value, which is 77.4.

As you can see, a higher standard deviation indicates that the values are spread out over a wider range.

The NumPy module has a method to calculate the standard deviation:

Example

Use the NumPy `std()` method to find the standard deviation:

```
import numpy

speed = [86,87,88,86,87,85,86]

x = numpy.std(speed)

print(x)
```

Example

```
import numpy

speed = [32,111,138,28,59,77,97]

x = numpy.std(speed)

print(x)
```

Variance

Variance is another number that indicates how spread out the values are.

In fact, if you take the square root of the variance, you get the standard deviation!

Or the other way around, if you multiply the standard deviation by itself, you get the variance!

To calculate the variance you have to do as follows:

1. Find the mean:

```
(32+111+138+28+59+77+97) / 7 = 77.4
```

2. For each value: find the difference from the mean:

```
 32 - 77.4 = -45.4
111 - 77.4 =  33.6
138 - 77.4 =  60.6
 28 - 77.4 = -49.4
 59 - 77.4 = -18.4
 77 - 77.4 = - 0.4
 97 - 77.4 =  19.6
```

3. For each difference: find the square value:

```
(-45.4)² = 2061.16
 (33.6)² = 1128.96
 (60.6)² = 3672.36
(-49.4)² = 2440.36
(-18.4)² =  338.56
(- 0.4)² =    0.16
 (19.6)² =  384.16
```

4. The variance is the average number of these squared differences:

```
(2061.16+1128.96+3672.36+2440.36+338.56+0.16+384.16) / 7 = 1432.2
```

Luckily, NumPy has a method to calculate the variance:

Example

Use the NumPy `var()` method to find the variance:

```
import numpy

speed = [32,111,138,28,59,77,97]

x = numpy.var(speed)

print(x)
```

Standard Deviation

As we have learned, the formula to find the standard deviation is the square root of the variance:

```
√1432.25 = 37.85
```

Or, as in the example from before, use the NumPy to calculate the standard deviation:

Example

Use the NumPy `std()` method to find the standard deviation:

```
import numpy

speed = [32,111,138,28,59,77,97]

x = numpy.std(speed)

print(x)
```

Symbols

Standard Deviation is often represented by the symbol Sigma: σ

Variance is often represented by the symbol Sigma Square: σ^2

What are Percentiles?

Percentiles are used in statistics to give you a number that describes the value that a given percent of the values are lower than.

Example: Let's say we have an array of the ages of all the people that lives in a street.

```
ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]
```

What is the 75. percentile? The answer is 43, meaning that 75% of the people are 43 or younger.

The NumPy module has a method for finding the specified percentile:

Example

Use the NumPy `percentile()` method to find the percentiles:

```
import numpy

ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]

x = numpy.percentile(ages, 75)

print(x)
```

Example

What is the age that 90% of the people are younger than?

```
import numpy

ages = [5,31,43,48,50,41,7,11,15,39,80,82,32,2,8,6,25,36,27,61,31]

x = numpy.percentile(ages, 90)

print(x)
```

Exercise to be done in lab:

1) Explore the numpy functions relevant statistical metrics described above. Try experimenting by using/setting the function parameters and note your observations in the observation note book as given below for each numpy function:

1) Title of statistical metric

2) Definition

3) numpy function name

4) syntax

5) brief description of function parameters

6) input and output as used while execution in python IDE.

7) errors and observations by changing the function parameters