

Modules and packages in python

The max() function

The Python max() function is used to find the highest value in an array.

Example 1.

```
Average_pulse_max = max(80, 85, 90, 95, 100, 105, 110, 115, 120, 125)
```

```
print (Average_pulse_max)
```

The min() function

The Python min() function is used to find the lowest value in an array.

Example 0.2

```
Average_pulse_min = min(80, 85, 90, 95, 100, 105, 110, 115, 120, 125)  
print(Average_pulse_min)
```

- **What is Statistics Used for?**

- Statistics is used in all kinds of science and business applications.
- Statistics gives us more accurate knowledge which helps us make better decisions.
- Statistics can focus on making **predictions** about what will happen in the future. It can also focus on **explaining** how different things are connected.
- **Note:** Good statistical explanations are also useful for predictions.

The Center of the Data

The center of the data is where most of the values in the data are located. Averages are measures of the location of the center.

There are different types of averages. The most commonly used are:

- Mean
- Median
- Mode

Note: In statistics, averages are often referred to as 'measures of central tendency'.

For example, using the values:

40, 21, 55, 21, 48, 13, 72

Mean

The mean is usually referred to as 'the average'.

The mean is the sum of all the values in the data divided by the total number of values in the data:

$$(40 + 21 + 55 + 31 + 48 + 13 + 72)/7 = \underline{38.57}$$

Note: There are multiple types of mean values. The most common type of mean is the **arithmetic** mean.

'mean' refers to the arithmetic mean.

Median

The **median** is the 'middle value' of the data.

The median is found by **ordering all the values** in the data and picking the middle value:

13, 21, 21, 40, 48, 55, 72

The median is less influenced by **extreme** values in the data than the mean.

Changing the last value to 356 does not change the median:

13, 21, 21, 40, 48, 55, **356**

The median is still 40.

Changing the last value to 356 changes the **mean** a lot:

$$(13 + 21 + 21 + 40 + 48 + 55 + 72)/7 = \underline{38.57}$$

$$(13 + 21 + 21 + 40 + 48 + 55 + \mathbf{356})/7 = \underline{79.14}$$

Note: Extreme values are values in the data that are much smaller or larger than the average values in the data.

Mode

The **mode** is the value(s) that appears most often in the data:

40, 21, 55, 21, 48, 13, 72

Here, 21 appears two times, and the other values only once. The mode of this data is 21.

The mode is also used for **categorical** data, unlike the median and mean. Categorical data can't be described directly with numbers, like names:

Alice, John, Bob, Maria, John, Julia, Carol

Here, John appears two times, and the other values only once. The mode of this data is John.

Note: There can be more than one mode if multiple values appear the same number of times in the data.

Python Package

A package is a container that contains various functions to perform specific tasks. For example, the math package includes the `sqrt()` function to perform the square root of a number.

While working on big projects, we have to deal with a large amount of code, and writing everything together in the same file will make our code look messy. Instead, we can separate our code into multiple files by keeping the related code together in packages.

Now, we can use the package whenever we need it in our projects. This way we can also reuse our code.

- **Python Modules**

- As our program grows bigger, it may contain many lines of code. Instead of putting everything in a single file, we can use modules to separate codes in separate files as per their functionality. This makes our code organized and easier to maintain.
- Module is a file that contains code to perform a specific task. A module may contain variables, functions, classes etc. Let's see an example,

Python Module addition

```
def add(a, b):  
    result = a + b  
    return result
```

Here, we have defined a function add() inside a module named example. The function takes in two numbers and returns their sum.

Example

```
import datetime
```

```
x = datetime.datetime.now()  
print(x)
```

Example

```
import datetime
```

```
x = datetime.datetime.now()
```

```
print(x.year)
```

```
print(x.strftime("%A"))
```

Example

```
import math
```

```
x = math.sqrt(64)
```

```
print(x)
```

Example

```
import math
```

```
x = math.ceil(1.4)
```

```
y = math.floor(1.4)
```

```
print(x) # returns 2
```

```
print(y) # returns 1
```