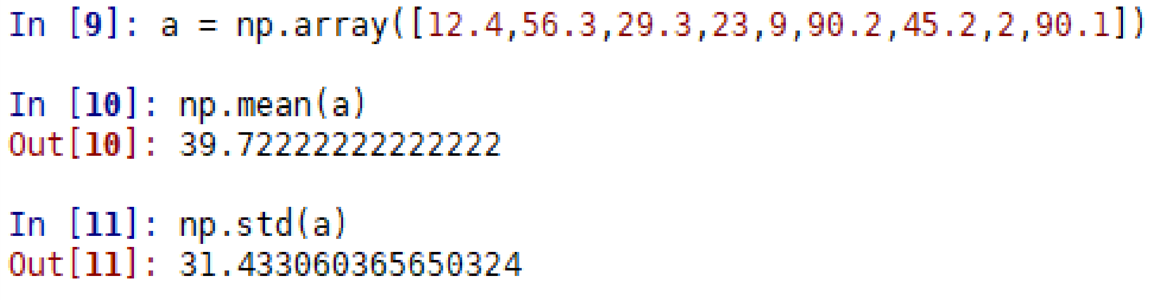
Sane's STATS Academy of Statistics

Functions

Functions

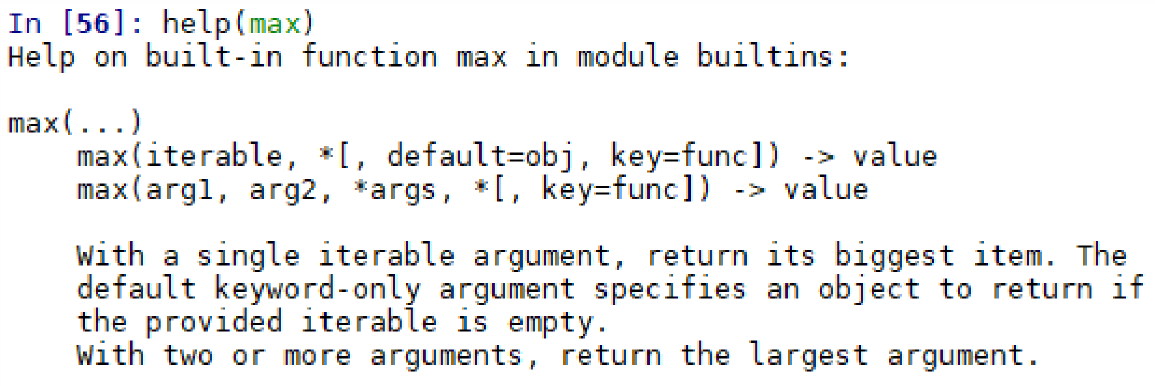
- Functions are units enabling specific tasks
- There are many built-in functions
- Help documentation of any function can be accessed with help() function



```
In [9]: a = np.array([12.4,56.3,29.3,23,9,90.2,45.2,2,90.1])

In [10]: np.mean(a)
Out[10]: 39.7222222222222

In [11]: np.std(a)
Out[11]: 31.433060365650324
```



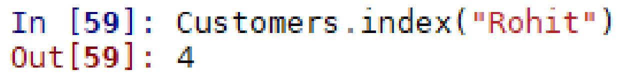
```
In [56]: help(max)
Help on built-in function max in module builtins:

max(...)
    max(iterable, *[, default=obj, key=func]) -> value
    max(arg1, arg2, *args, *[, key=func]) -> value

    With a single iterable argument, return its biggest item. The
    default keyword-only argument specifies an object to return if
    the provided iterable is empty.
    With two or more arguments, return the largest argument.
```

Methods

- Functions that belong to the object are called Methods
- Hence there are list methods, float methods, string methods etc.
- When any method is to be called then it is to be invoked on the object with "." specifier



```
In [59]: Customers.index("Rohit")
Out[59]: 4
```

Creating your own function

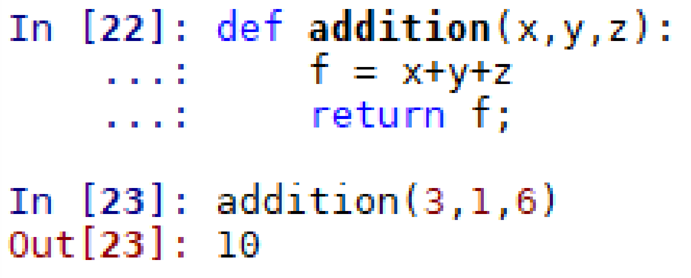
- We need to use the keyword def here

Syntax :

```
def userDefFunction (arg1, arg2, arg3 ...):
        statement 1
        statement 2
        … calculation of value
        return value ;
```

- While defining function, the indentation must be given as it is the part of the syntax

Function Example

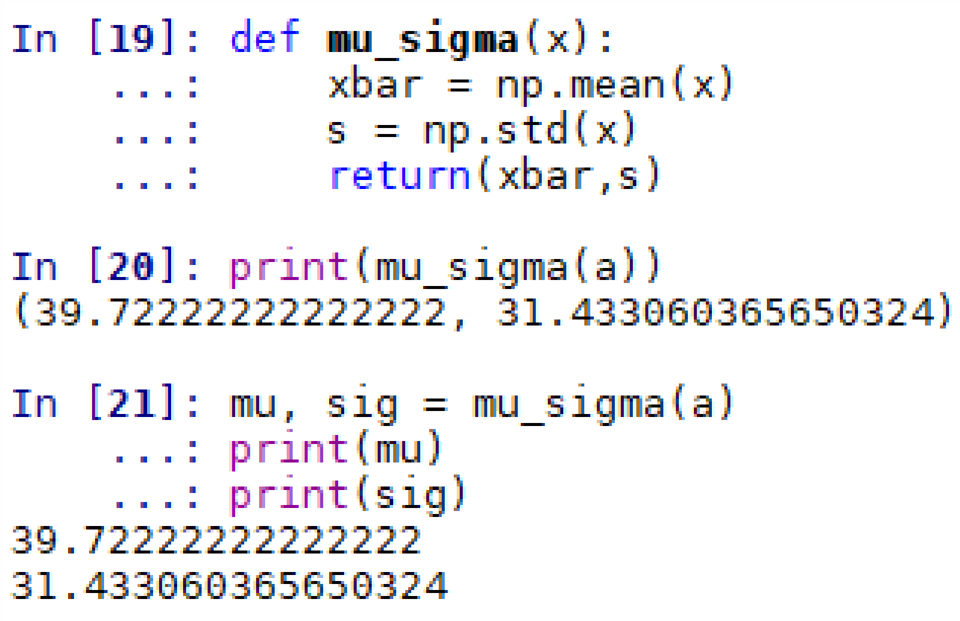


```
In [22]: def addition(x,y,z):
    ...:     f = x+y+z
    ...:     return f;

In [23]: addition(3,1,6)
Out[23]: 10
```

Function Returning Multiple Objects

- A function can also be defined that can return multiple objects



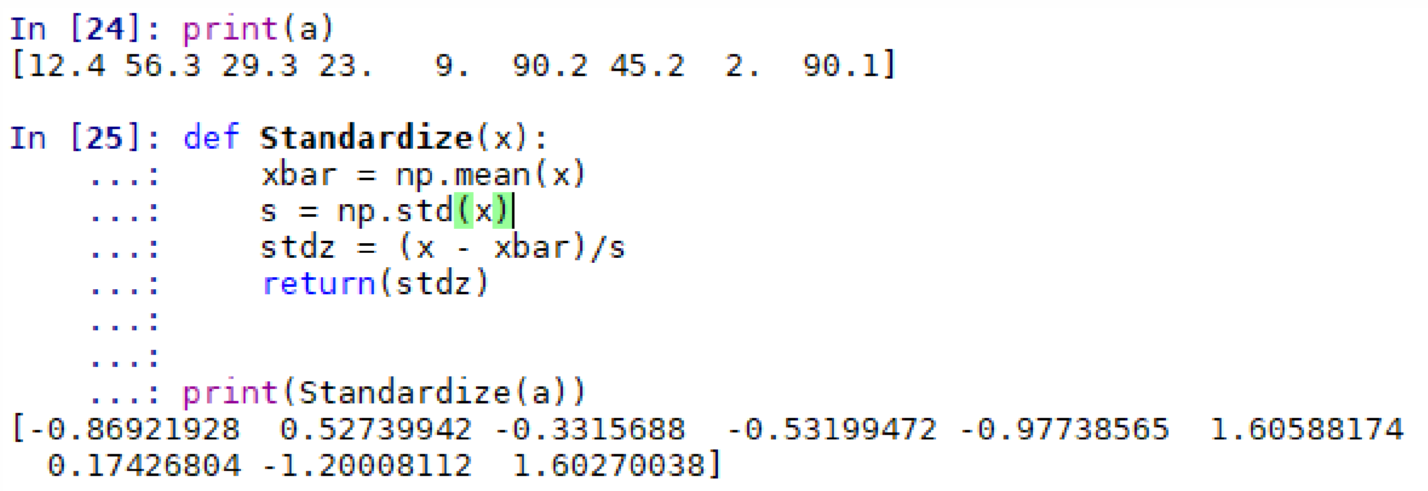
```
In [19]: def mu_sigma(x):
    ...:     xbar = np.mean(x)
    ...:     s = np.std(x)
    ...:     return(xbar,s)

In [20]: print(mu_sigma(a))
(39.72222222222222, 31.433060365650324)

In [21]: mu, sig = mu_sigma(a)
    ...: print(mu)
    ...: print(sig)
39.72222222222222
31.433060365650324
```

Functions for Transformations

- Functions can be used to transform data which is necessary for statistical analysis
- e.g. Standard Scaling: $\frac{X-\mu}{\sigma}$

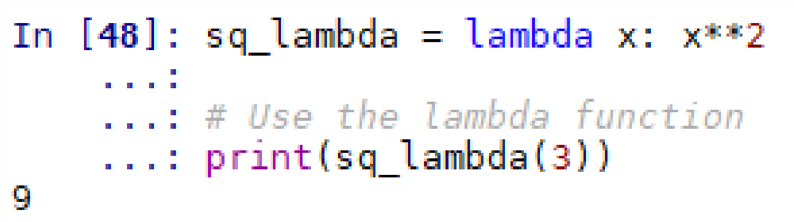


```
In [24]: print(a)
[12.4 56.3 29.3 23.   9.  90.2 45.2  2.  90.1]

In [25]: def Standardize(x):
    ...:     xbar = np.mean(x)
    ...:     s = np.std(x)
    ...:     stdz = (x - xbar)/s
    ...:     return(stdz)
    ...:
    ...:
    ...: print(Standardize(a))
[-0.86921928  0.52739942 -0.3315688  -0.53199472 -0.97738565  1.60588174
  0.17426804 -1.20008112  1.60270038]
```

Lambda Functions

- Anonymous functions in Python are known as lambda functions



```
In [48]: sq_lambda = lambda x: x**2
    ...:
    ...: # Use the lambda function
    ...: print(sq_lambda(3))
9
```

Questions?