This course has already ended.

Calculations

Before you do calculations with your data, it is good to check if there are libraries that do exactly what you are intending to do. On the one side, libraries are usually performance-optimized. This will reduce the time and space your calculations need. Depending on the size of your data this can make the difference if you are able to compute the results in reasonable time or not. On the other side, using libraries makes your code less error-prone and easier to read.

Lets look at an example We previously talked about the calculation of the mean and the standard deviation of a list of numbers.

In order to measure the time spent, we use a timestamp before and after a caclution and can derive from that the time used by the calculation. To do so, we import the time package.

```
import numpy as np
import math
import time
```

Now we implement our own functions to calculate the average and the mean of a list of numbers:

```
def calc_average(grades):
    sum_grade = 0
    for grade in grades:
        sum_grade += grade
    avg = sum_grade / len(grades)
    return avg

def calc_std(grades):
    avg = calc_average(grades)
    squareSum = 0
    for grade in grades:
        gradeDiff = grade - avg
        squareSum += gradeDiff*gradeDiff
    temp = squareSum / len(grades)
    std = math.sqrt(temp)
    return(std)
```

We now want to compare the time spent to calculate the average and the standard deviation with our own functions and with the functions provided by numpy. First we fill a list with random integers. As our example is about grades, we choose the grades between 1 and 5.

```
import random
listLength = 50000000
randomlist = []
for i in range(listLength):
    randomlist.append(random.randint(1, 5))
```

We run our own functions and the corresponding numpy functions on the same input. We measure the time and print it.

```
startOwn = time.time()
avg = calc_average(randomlist)
std = calc_std(randomlist)
endOwn = time.time()
print("Average: ", avg)
print("Standard deviation: ", std)
print("My own calculations took: ")
print(endOwn-startOwn)

startNumpy = time.time()
avg = np.average(randomlist)
std = np.std(randomlist)
endNumpy = time.time()
print("Average: ", avg)
print("Standard deviation: ", std)
print("Numpy calculations took: ")
print(endNumpy-startNumpy)
```

The output looks like that:

```
Average:  2.99990066
Standard deviation:  1.4141579862175422
Calculations took:
8.333605766296387
Average:  2.99990066
Standard deviation:  1.4141579862701203
Calculations took:
5.98210072517395
```