

## Abgabe PHYSEC 3

### 1. Messungen

### 2. Implementierung Pearson Correlation

Die Aufgabenstellung ist, die in Abbildung 1 abgebildete Formel in einem vorgegeben Python-Code als Funktion *correlation(X, Y)* zu implementieren, sodass es von einem mitgegeben Framework fehlerlos getestet werden kann. Die vollständige *exercise3.py* kann hier heruntergeladen werden.

$$\rho(x, y) = \frac{\sum_{i=0}^{n-1} (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=0}^{n-1} (x_i - \bar{x})^2 \cdot \sum_{i=0}^{n-1} (y_i - \bar{y})^2}}$$

Abbildung 1: Auszug aus dem Assignment

```
import utils
import numpy

"""
Excercise 3:
Implement the Pearson correlation coefficient.
Do NOT use any given function for standard-deviation
or mean-value but implement them by yourself.

X, Y are given as lists.

Blockwise application is done outside so please use the
whole vectors at once.
"""

def correlation(X, Y):
    mean_x = numpy.mean(X) # Mittel Vektor X
    mean_y = numpy.mean(Y) # Mittel Vektor Y

    numerator = 0 # zaehler
    denominator_x = 0 # der X beinhaltene Faktor im Nenner
    denominator_y = 0 # der Y beinhaltene Faktor im Nenner
```

```

# Falls Vektoren unterschiedlich lang sind
if len(X) != len(Y):
    raise Exception("Length not equal!\n")

for i in range(len(X)):
    numerator += (X[i] - mean_x)*(Y[i] - mean_y)
    denominator_x += (X[i]-mean_x)*(X[i]-mean_x)
    denominator_y += (Y[i]-mean_y)*(Y[i]-mean_y)

denominator = numpy.sqrt(denominator_x * denominator_y)

# Falls Nenner=0, dann wird der jeweilige Eintrag ignoriert
# und wird somit auch nicht in die Skizzen eingetragen
if denominator == 0:
    return float('nan')

pearson = numerator/denominator

return pearson
# return utils.not_yet_implemented("Correlation")

```

### 3. Auswertung

### 4. Quantisierer Jana Multibit

### 5. Quantisierer Mathur Suhas

### 6. Bonus: Reading Assignment