kollimator_ausgleichung.py

```python
# Ausgleichende Gerade
# ##############################################################################

# Authors:
# Joshua Wolf
# Silas Teske
# Lasse Zeh
# Christopher Mahn

# ##############################################################################

# Import of Libraries
# ------------------------------------------------------------------------------

import numpy as np
import math as ma
import matplotlib.pyplot as plt

# Functions
# ------------------------------------------------------------------------------

def emp_kov(x,y):
    n = x.shape[0]
    kov = 0
    m_x = np.average(x)
    m_y = np.average(y)

    for i in range(n):
        kov += ((x[i]-m_x)*(y[i]-m_y))/(n-1)
    return kov

def emp_varianz(p):
    n = p.shape[0]
    p_q = np.average(p)
    summe = 0

    for i in range(n):
        summe += ((p[i]-p_q)**2)
    v = summe/(n-1)
    return v

# Beginning of the Programm
# ------------------------------------------------------------------------------

daten = np.loadtxt("messwerte_v.txt")
x_v = daten[:,0]
y_v = daten[:,1]

fig = plt.plot(x_v,y_v,'o')
plt.grid()
plt.xlabel('x-Werte')
plt.ylabel('y-Werte')
plt.show(fig)

m_s = np.average(daten)
print("Mittelwert berechnet:",m_s)
m_s_x = np.average(x_v)
m_s_y = np.average(y_v)

e_k = emp_kov(x_v,y_v)
print("empirische Kovarianz berechnet:",emp_kov(x_v,y_v))

e_v_x = emp_varianz(x_v)
e_v_y = emp_varianz(y_v)
print("empirische Varianz x:",e_v_x)
print("empirische Varianz y:",e_v_y)

a = e_k / e_v_x
print("Steigung:",a)

b = (m_s_y) - (a*m_s_x)
print("y-Achsenabschnitt:",b)

y = (a*x_v)+b

v = y-y_v

fig = plt.plot(x_v,y_v,"o")
plt.plot(x_v,y)
plt.xlabel('x-Werte')
```

```
081 plt.ylabel('y-Werte')
082 plt.show(fig)
083
084 kk = (e_k)/((ma.sqrt(emp_varianz(x_v)))*(ma.sqrt((emp_varianz(y_v)))))
085 print("Korrelationskoeffizient:",kk)
```