```
# -*- coding: utf-8 -*-

# This Python-program was developed using "Enthought Canopy v. 1.4.1
# analysis environment, on a MacBook Pro running OS X 10.9.5
#
# written by T. Ihn, D-PHYS ETH Zurich, 5 Oct 2014

#
# Benötigte Befehle aus Programmbibliotheken
#
from csv import reader

from numpy import array
from numpy import mean
from numpy import var
from numpy import corrcoef
from numpy import polyfit
from numpy import arange

from matplotlib import rcParams

from matplotlib.pyplot import figure
from matplotlib.pyplot import axes
from matplotlib.pyplot import plot
from matplotlib.pyplot import axis
from matplotlib.pyplot import xticks
from matplotlib.pyplot import yticks
from matplotlib.pyplot import tick_params
from matplotlib.pyplot import xlabel
from matplotlib.pyplot import ylabel
from matplotlib.pyplot import title
from matplotlib.pyplot import text
from matplotlib.pyplot import show
from matplotlib.pyplot import savefig

from math import sqrt

from scipy.stats import linregress

#
# Daten aus einem csv-File einlesen
#
readerout = reader(open("daten.csv","rb"),delimiter=',');
```

```
x = list(readerout);
data = array(x).astype('float')
print "Datenarray:"
print data

#
# Mit Datenlisten arbeiten
#

# Länge der Datenliste (Zahl der Datenpunkte) ermitteln:
n = len(data)
print "n = " + str(n)

# Auf einzelne Elemente (Datenpunkte) der Datenliste zugreifen
print "Dritter Datenpunkt:"
print data[2]


# Auf Bereiche in der Datenliste zugreifen (Datenpunkte 2-4)
print "Datenpunkte 2-4:"
print data[1:4]

# Nur die x-Werte der Datenpunkte 2 bis 4
print "x-Werte der Datenpunkte 2-4:"
print data[1:4,0]
# alle x-Werte
x = data[:,0]

# Nur die y-Werte der Datenpunkte 2 bis 4
print "y-Werte der Datenpunkte 2-4:"
print data[1:4,1]
# alle y-Werte
y = data[:,1]

#
# Datenliste graphisch darstellen
#
figure(
    num=1,
    figsize=(7,5.08),
    dpi=80,
    facecolor='white')
rcParams['axes.linewidth']=2
```

```
axes([0.15,0.18,0.82,0.70],
    axisbg='lightgray')
plot(x,y,'bo')
axis([0,9,0,25])
xticks(arange(0,9,1),
    fontsize=24,
    fontname='Times New Roman')
yticks(arange(0,26,5),
    fontsize=24,
    fontname='Times New Roman')
tick_params(width=2,length=10)
xlabel(r'$x$',
    fontsize=24)
ylabel(r'$y$',
    fontsize=24)
show()

#
# Statistik der Datenliste
#

# Mittelwert der x-Werte
Meanx = mean(data[:,0])
print "x-Mittelwert = " + str(Meanx)

# Mittelwert der y-Werte
Meany = mean(data[:,1])
print "y-Mittelwert = " + str(Meany)

# Varianz der x-Werte
Varx = var(data[:,0])
print "x-Varianz = " + str(Varx)

# Varianz der y-Werte
Vary = var(data[:,1])
print "y-Varianz" + str(Vary)

# Empirischer Korrelationskoeffizient
rho = corrcoef(x.T,y.T)[0,1]
print "empirischer Korrelationskoeffizient = " + str(rho)

#
# Lineare Regression I: Schätzwerte und ihre Unsicherheiten (gemäss
```

```
#
print " "
print "Lineare Regression I: nach Vorlesung"
A0 = sqrt(Vary/Varx)*rho
B0 = Meany
C0 = Vary*(1-rho**2)
sigma2 = n*C0/(n-2)

sigmaA = sqrt(C0/Varx/(n-2))
print('A0 = %1.2f +/- %1.2f'% (A0,sigmaA))

sigmaB = sigmaA*sqrt(Varx)
print('B0 = %1.2f +/- %1.2f'% (B0,sigmaB))

ssigma2 = n*C0/(n-2)*sqrt(2./(n-4))
print('sigma2 = %1.2f +/- %1.2f'% (sigma2,ssigma2))

#
# Fitkurve graphisch darstellen
#
xFit = arange(0,9,0.1)
yFit = A0*(xFit - Meanx) + B0
figure(
    num=2,
    figsize=(7,5.08),
    dpi=80,
    facecolor='white')
rcParams['axes.linewidth']=2
axes([0.15,0.18,0.82,0.70],
    axisbg='lightgray')
plot(xFit,yFit,'r-')
axis([0,9,0,25])
xticks(arange(0,9,1),
    fontsize=24,
    fontname='Times New Roman')
yticks(arange(0,26,5),
    fontsize=24,
    fontname='Times New Roman')
tick_params(width=2,length=10)
xlabel(r'$x$',
    fontsize=24)
ylabel(r'$y$',
    fontsize=24)
```

```
show()

#
# Daten mit Fitkurve graphisch darstellen
#
xFit = arange(0,9,0.1)
yFit = A0*(xFit - Meanx) + B0
figure(
    num=3,
    figsize=(7,5.08),
    dpi=80,
    facecolor='white')
rcParams['axes.linewidth']=2
axes([0.15,0.18,0.82,0.70],
    axisbg='lightgray')
plot(x,y,'bo',xFit,yFit,'r-')
axis([0,9,0,25])
xticks(arange(0,9,1),
    fontsize=24,
    fontname='Times New Roman')
yticks(arange(0,26,5),
    fontsize=24,
    fontname='Times New Roman')
tick_params(width=2,length=10)
text(0.3,22,r'$y = A_0\left(x-\overline{x_i}\right)+ B_0$',
    fontsize=24,
    fontname='Times New Roman')
text(0.3,19,r'$A_0 = 2.80 \pm 0.12$',
    fontsize=24,
    fontname='Times New Roman')
text(0.3,16,r"$B_0 = 14.11 \pm 0.29$",
    fontsize=24,
    fontname='Times New Roman')
title('Linear Regression',
    fontsize=24,
    fontname='Times New Roman')
xlabel(r'$x$',
    fontsize=24)
ylabel(r'$y$',
    fontsize=24)
show()
# Abbildung als pdf-file speichern
savefig('/Users/ihn/Documents/Teaching/VP-Leitung/DataAnalysis/2014/
```

```
#
# Variablen löschen
#
del n

#
# Lineare Regression II: mit polyfit
#
print " "
print "Lineare Regression II: mit polyfit"
p, residuals, rank, singular_values, rcond = polyfit(x-Meanx,y,1,ful
# p enthält den vektor mit den beiden Fitparametern (Steigung und y-
# residuals enthält die Summe der Quadrate der Abweichungen vom Fit
n = len(x);
C0 = residuals[0]/n
Varx = var(x)
sigma2 = residuals[0]/(n-2)
sigmaA = sqrt(C0/(n-2)/Varx)
sigmaB = sqrt(C0/(n-2))
ssigma2 = sigma2*sqrt(2./(n-4))
print('A0 = %1.2f +/- %1.2f'% (p[0],sigmaA))
print('B0 = %1.2f +/- %1.2f'% (p[1],sigmaB))
print('sigma2 = %1.2f +/- %1.2f'% (sigma2,ssigma2))

#
# Lineare Regression III: mit linregress
#
print " "
print "Lineare Regression III: mit linregress"

slope, intercept, r_value, p_value, std_err = linregress(x-Meanx,y)
# slope enthält die gesuchte Steigung A0
# intercept enhält den gesuchten y-Abschnitt B0
# r_value enthält den empirischen Regressionskoeffizienten
# std_err enthält die Standardabweichung von A0
Varx = var(x)
n = len(x)
C0 = Varx*(n-2)*std_err**2
sigma2 = n*C0/(n-2)
sigmaB = std_err*sqrt(Varx)
ssigma2 = sigma2*sqrt(2./(n-4))
print('A0 = %1.2f +/- %1.2f'% (slope,std_err))
```

```
print('B0 = %1.2f +/- %1.2f'% (intercept,sigmaB))
print('sigma2 = %1.2f +/- %1.2f'% (sigma2,ssigma2))
```