Serie 06: Methode der kleinsten Quadrate

Aufgabe 1

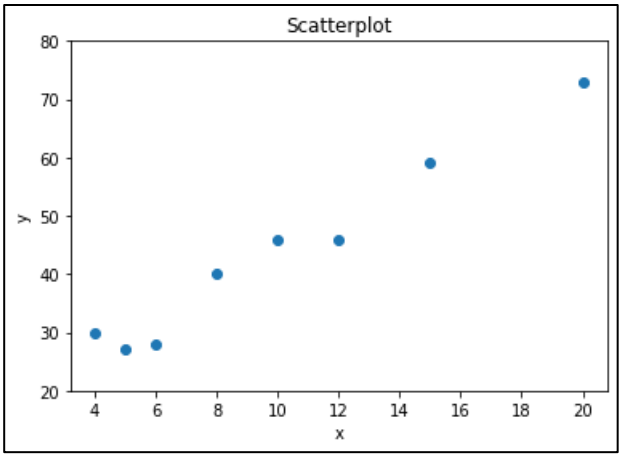
Berechnen Sie das Streudiagramm der Daten:

x	5	10	20	8	4	6	12	15
y	27	46	73	40	30	28	46	59

a) Zeichnen Sie das Streudiagramm der Daten.
b) Berechnen Sie die Gleichung der Regressionsgeraden mit Hilfe einer Tabelle.
c) Lösen Sie Aufgabe b) mit dem Taschenrechner bzw. mit Python.
d) Bestimmen Sie das Bestimmtheitsmass und die Korrelation.
e) Zeichnen Sie den Residuen Plot.

Lösung:

a)



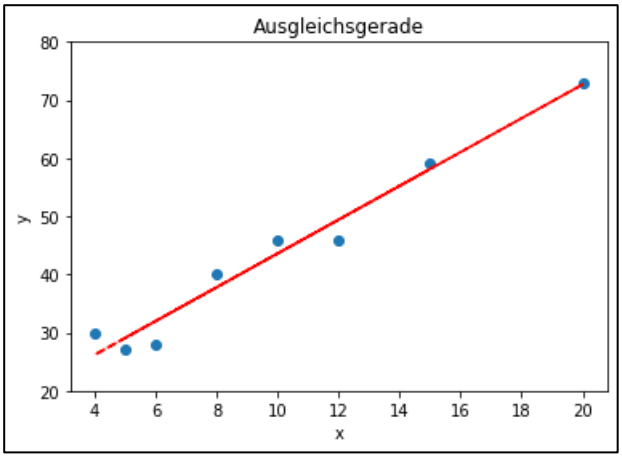

```
import numpy as np
import matplotlib.pyplot as plt

##Aufgabe 6.1
x=np.array([5, 10, 20, 8, 4, 6, 12, 15])
y=np.array([27, 46, 73, 40, 30, 28, 46, 59])
#Streudiagramm
plt.figure(1)
plt.scatter(x,y)
plt.xlabel('x')
plt.ylabel('y')
plt.ylim(20,80)
plt.title('Scatterplot')
```

b) und c)

$y = 2.90x + 14.58$

x	5	10	20	8	4	6	12	15	Mittelwert: $\bar{x}$=10
y	27	46	73	40	30	28	46	59	Mittelwert: $\bar{y}$=43.625
$x - \bar{x}$	-5	0	10	-2	-6	-4	2	5	
$y - \bar{y}$	-16.625	2.375	29.375	-3.625	-13.625	-15.625	2.375	15.375	
$(x - \bar{x})(y - \bar{y})$	83.125	0	293.75	7.25	81.75	62.5	4.75	76.875	Summe = 610
$(x - \bar{x})^2$	25	0	100	4	36	16	4	25	Summe = 210

$m = \frac{S_{xy}}{S_x^2} = \frac{610}{210} = 2.905$ $\quad$ $d = \bar{y} - m\bar{x} = 43.625 - 2.905 \cdot 10 = 14.575$




```
#Ausgleichsgerade
plt.figure(2)
m,d = np.polyfit(x, y, 1)
plt.plot(x, y, 'o', x, m*x+d, '--r')
plt.xlabel('x')
plt.ylabel('y')
plt.ylim(20,80)
plt.title('Ausgleichsgerade')
```

```
#Regressionsgerade 1. Variante
n=x.size
Mx = 1/n*sum(x)
My = 1/n*sum(y)
sx=np.sqrt(1/(n-1)*sum(np.square(x-Mx)))
sy=np.sqrt(1/(n-1)*sum(np.square(y-My)))
sxy=1/(n-1)*(sum((x-Mx)*(y-My)))
rxy1=sxy/(sx*sy)
m1=rxy1*sy/sx
d1=My-m*Mx
print('Variante 1')
print('Steigung m = ',m1, 'Achsabschnitt d = ',d1)
```

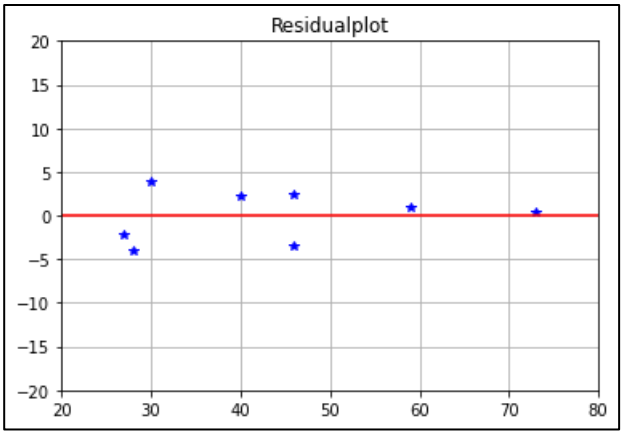
```
#Regressionsgerade 2. Variante
n=x.size
Mx = 1/n*sum(x)
My = 1/n*sum(y)
Sxy=sum((x-Mx)*(y-My))
Sxx=sum(np.square(x-Mx))
m2=Sxy/Sxx
d2=My-m*Mx
print('Variante 2')
print('Steigung m = ',m2, 'Achsabschnitt d = ',d2)
```

d) $r = 0.9840$; $B = r^2 = 0.9683$

```
#Bestimmtheitmass und Korrelation 1.
Variante
R21=np.square(rxy1)
print('Variante 1')
print('Korrelationskoeffizient rxy = ',rxy1,
'Bestimmtheitsmass R^2 = ',R21)
```

```
#Bestimmtheitsmass und Korrelation 2.
Variante
yb=m2*x+d2
Syy=sum(np.square(y-My))
Sybyb=sum(np.square(yb-My))
R22=Sybyb/Syy
rxy2=np.sqrt(R22)
print('Variante 2')
print('Korrelationskoeffizient rxy = ',rxy2,
'Bestimmtheitsmass R^2 = ',R22)
```

e)




```
#Residuen
Res=y-yb;
#Plot der Residuen
plt.figure(3)
plt.plot(y,Res,'b*',[20,80],[0,0],'r')
plt.xlim(20,80)
plt.ylim(-20,20)
plt.title('Residualplot')
plt.grid()
```

Aufgabe 2

Gegeben die Datenpaare

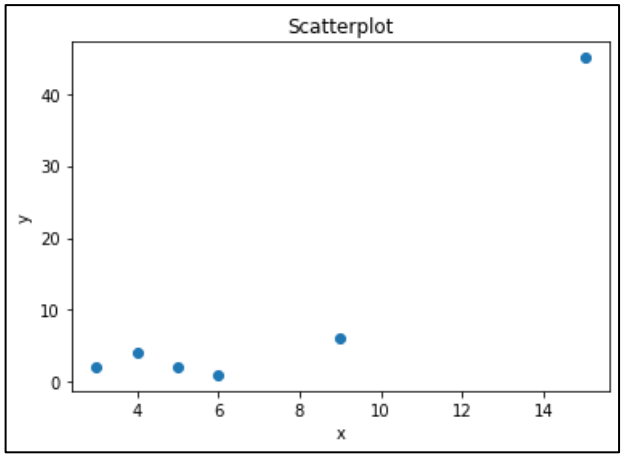
x	3	4	6	5	9	15
y	2	4	1	2	6	45

a) Zeichnen Sie das Streudiagramm

b) Bestimmen Sie die Regressionsgerade mit und ohne den letzten Punkt (Ausreisser), und tragen Sie die Regressionsgeraden ein. Welchen Einfluss hat der Ausreisser?

Lösung:

a)




```
import numpy as np
import matplotlib.pyplot as plt

#Aufgabe, 6.2
x=np.array([3, 4, 6, 5, 9, 15])
y=np.array([2, 4, 1, 2, 6, 45])
#Streudiagramm
plt.figure(1)
plt.plot(x,y,'o')
plt.title('Scatterplot')
plt.xlabel('x')
plt.ylabel('y')
```

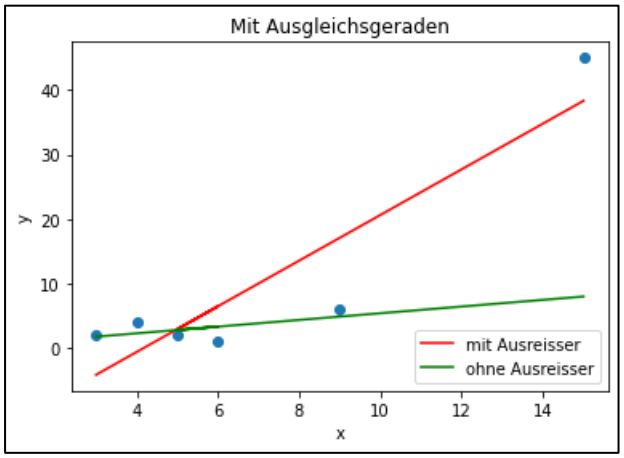
b)

```
#Regressionsgerade mit Ausreisser
n1=x.size
Mx1 = 1/n1*sum(x)
My1 = 1/n1*sum(y)
Sxy1=sum((x-Mx1)*(y-My1))
Sxx1=sum(np.square(x-Mx1))
m1=Sxy1/Sxx1
d1=My1-m1*Mx1
```

```
#Regressionsgerade ohne Ausreisser
xneu=x[0:-1]
yneu=y[0:-1]
n2=xneu.size;
Mx2 = 1/n2*sum(xneu)
My2 = 1/n2*sum(yneu)
Sxy2=sum((xneu-Mx2)*(yneu-My2))
Sxx2=sum(np.square(xneu-Mx2))
```

$y = 3.541 \cdot x - 15.577$

$y = 0.519 \cdot x + 0.198$




```
#Grafik mit beiden Regressionsgeraden
plt.figure(2)
plt.plot(x,y,'o')
plt.plot(x,m1*x+d1,'r',label='mit
Ausreisser')
plt.plot(x,m2*x+d2,'g',label='ohne
Ausreisser')
plt.legend(loc='lower right')
plt.title('Mit Ausgleichsgeraden')
plt.xlabel('x')
plt.ylabel('y')
```

Aufgabe 3

Für ein Unternehmen soll untersucht werden, welcher Zusammenhang zwischen Umsatz und Anzahl Beschäftigten gilt:

Anz. Beschäftigte	3	8	19	22	31	42	48	52	54
Umsatz in Mio.	2	31	49	65	84	96	117	129	146

a) Bestimmen Sie die Parameter der Regressionsgeraden.
b) Welchen Umsatz könnte das Unternehmen erwarten, wenn es 200 Beschäftigte hätte?
c) Berechnen Sie die korrigierte Gesamtvarianz, die korrigierte erklärte Varianz und die Summe der Residuen Quadrate.

Lösung: (mit exakten Werten berechnet)

a) $U = 2.46 \cdot B + 3.72$ b) $U = 495.15$ Mio.Fr.

c) Gesamtvarianz: $s_y{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(y_i - \bar{y})^2 = 2258.6$;

erklärte Varianz: $s_{\hat{y}}{}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 = 2202.2$;

Summe der Residuen Quadrate: $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = 451.02$.

Nichtlineares Verhalten

Aufgabe 4

Bestimmen Sie für die folgenden Daten das beste Modell vom Typ $y = C \cdot a^x$:

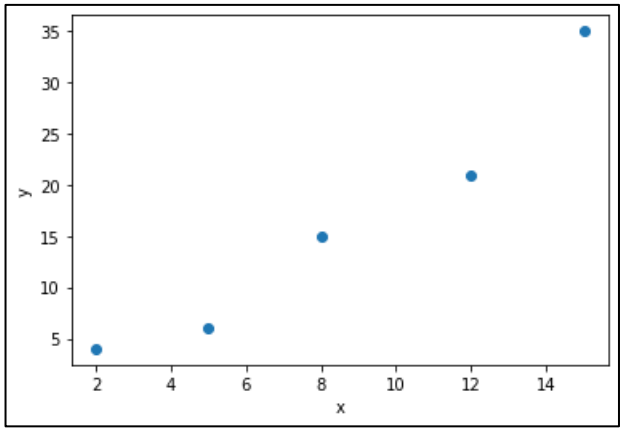
x	2	5	8	12	15
y	4	6	15	21	35

Bestimmen sie auch die durch das Modell errechneten y - Werte.

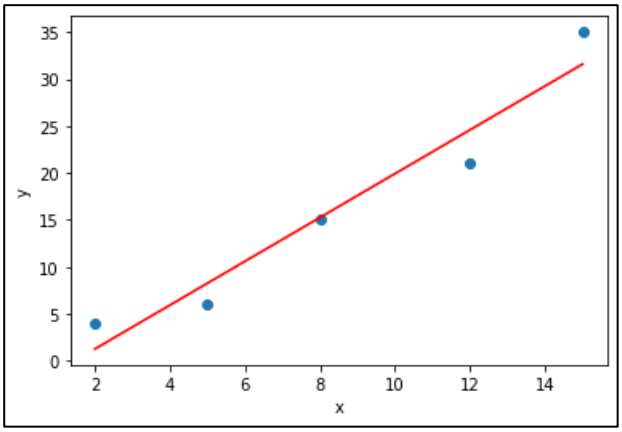
Lösung:

Ansatz $y = C \cdot a^x$ => $y = 2.95 \cdot 1.18^x$.

Streuplot von x und y zeigt nichtlineares Verhalten:



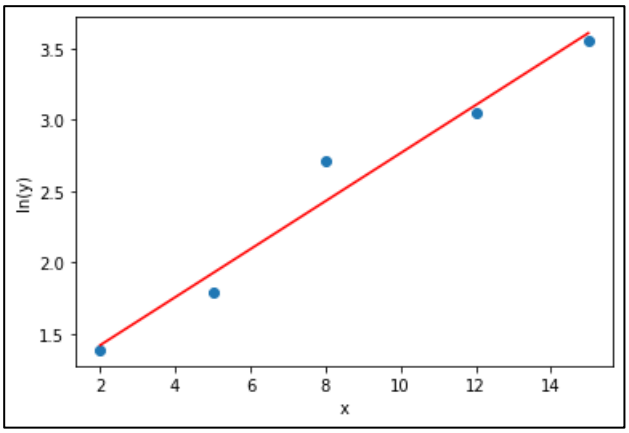

```
import numpy as np
import matplotlib.pyplot as plt

#Aufgabe 6.4
x=np.array([2, 5, 8, 12, 15])
y=np.array([4, 6, 15, 21, 35])
#Streudiagramm
plt.figure(1)
plt.scatter(x,y)
plt.xlabel('x')
plt.ylabel('y')
```




```
plt.figure(2)
m1,d1 = np.polyfit(x, y, 1)
plt.plot(x,y,'o',x,m1*x+d1,'r')
plt.xlabel('x')
plt.ylabel('y')
```

Gerade beschreibt Verhalten unzureichend
→ nichtlineares Modell

Transformation
von $y = C \cdot a^x$ nach $ln(y) = ln(a) \cdot x + ln\, C$ (Geradengleichung)




```
#Tranformation der y-Werte
ly=np.log(y)
#Neuer Plot
plt.figure(3)
m2,d2 = np.polyfit(x, ly, 1)
plt.plot(x,ly,'o',x,m2*x+d2,'r')
plt.xlabel('x')
plt.ylabel('ln(y)')
```

```
#Regression mit den logarithmierten Werten
n=x.size
Mx = 1/n*sum(x)
Mly = 1/n*sum(ly)
Sxly=sum((x-Mx)*(ly-Mly))
Sxx=sum(np.square(x-Mx))
lm=Sxly/Sxx
ld=Mly-lm*Mx
#Rücktransformation der Werte
m3=np.exp(lm)
d3=np.exp(ld)
```

Aufgabe 5

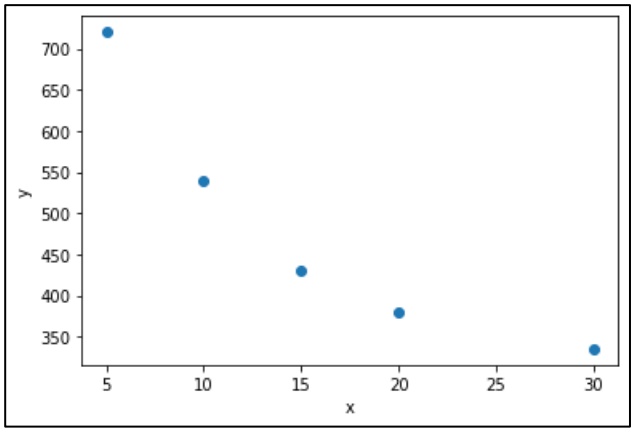
Ein gegossener Glaskörper kühlt bei einer Aussentemperatur von 300K langsam ab. Man beobachtet die folgenden Temperaturen:

Zeit (in h)	5	10	15	20	30
Temp.T(Kelvin)	720	540	430	380	335

a) Zeichnen Sie ein Streudiagramm.
b) Welches Modell passt?
c) Verwenden Sie das Modell $T - 300 = C \cdot a^t$. Zeichnen Sie dementsprechend ein Streudiagrammmit den Daten $x = t$ und $y = ln(T - 300)$. Bestimmen Sie aus dem linearen Trend die Grössen a und C.
d) Berechnen Sie mit den modellierten Grössen: Nach wie vielen Stunden ist der Glaskörper auf 305K abgekühlt?

Lösung:

a)



c) $T - 300 = C \cdot a^t$ => $ln(T - 300) = ln(C) + t \cdot ln(a)$ => $T = 639 \cdot 0.905^t + 300$.

	1	2	3	4	5		Mittelwerte
x	5	10	15	20	30		16
x^2	25	100	225	400	900		330
y	720	540	430	380	335		481
$\ln(y - 300)$	6.040	5.481	4.868	4.382	3.555		4.865
$x \cdot \ln(y - 300)$	30.201	54.806	73.013	87.641	106.660		70.464

$s_{xy} = \overline{x \cdot \ln(y - 300)} - \bar{x} \cdot \overline{\ln(y - 300)} = 70.464 - (16 \cdot 4.865) = -7.378$

$s_x^2 = \overline{x^2} - \bar{x}^2 = 330 - 16^2 = 74$

$m = \frac{s_{xy}}{s_x^2} = -\frac{7.378}{74} = -0.0997$

$d = \overline{\ln(y - 300)} - m\bar{x} -= 4.865 - (-0.0997) \cdot 16 = 6.460$

Rücktransformation: $C = e^d = 639.352$ und $a = e^m = 0.905$

d) $T = 305K$: $t = \frac{\ln(T-300) - \ln(639)}{\ln(0.905)}$ ≈Nach 48 Stunden und 36 Minuten

Aufgabe 6

Von einem bestimmten Autotyp wurden bei den Angeboten von Gebrauchtwagen die gefahrenen km mit den Preisen verglichen:

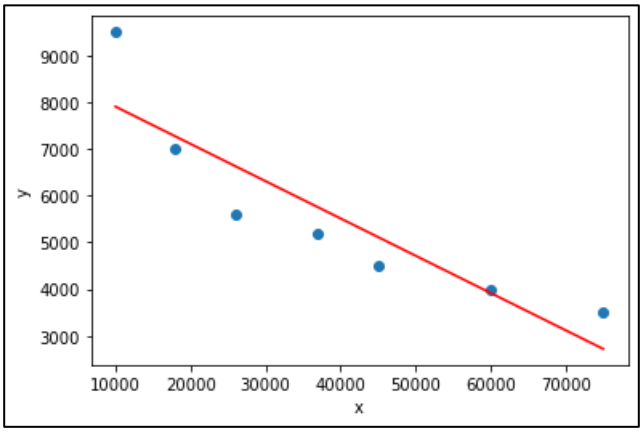
x (in T km)	10	18	26	37	45	60	75
y (CHF.)	9500	7000	5600	5200	4500	4000	3500

Bestimmen Sie mit Hilfe der linearen Regression einen funktionellen Zusammenhang zwischen den Variablen x und y. wie teuer würde ein Gebrauchtwagen mit 90'000 km geschätzt?

Lösung:

1. Möglichkeit: Ansatz $y = C \cdot a^x \Rightarrow y = 9171 \cdot 0.99^x$.

2. Möglichkeit: Ansatz $y = a \cdot x^b \Rightarrow y = 5329.5 \cdot x^{-0.48}$.

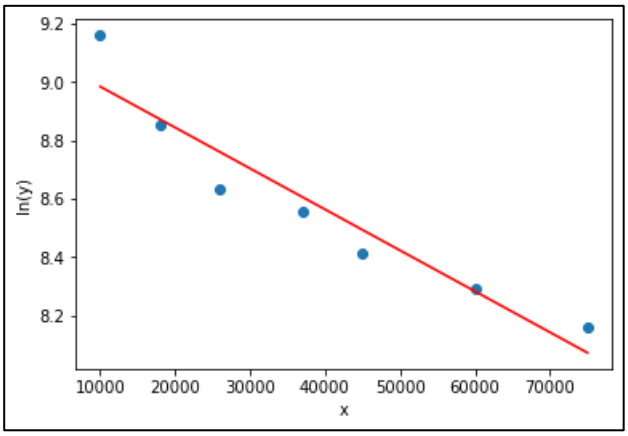



```
import numpy as np
import matplotlib.pyplot as plt

#Aufgabe 6.6
x=np.array([10000, 18000, 26000, 37000,
45000, 60000, 75000])
y=np.array([9500, 7000, 5600, 5200, 4500,
4000, 3500])
#Streudiagramm
plt.figure(1)
m1,d1 = np.polyfit(x, y, 1)
plt.plot(x,y,'o',x,m1*x+d1,'r')
plt.xlabel('x')
plt.ylabel('y')
```

Gerade beschreibt Verhalten unzureichend

→ nichtlineares Modell

1. Möglichkeit: $y = C \cdot a^x$



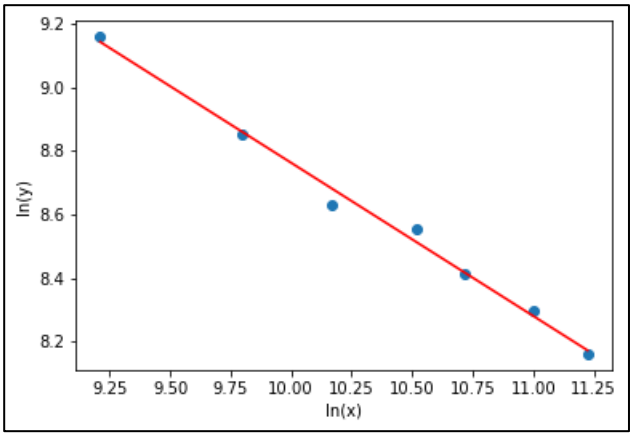

```
#Tranformation der y-Werte
ly=np.log(y)
#Neuer Plot
plt.figure(2)
m2,d2 = np.polyfit(x, ly, 1)
plt.plot(x,ly,'o',x,m2*x+d2,'r')
plt.xlabel('x')
plt.ylabel('ln(y)')
```

```
#Regression mit den logarithmierten y-Werten
n=x.size
Mx = 1/n*sum(x)
Mly = 1/n*sum(ly)
Sxly=sum((x-Mx)*(ly-Mly))
Sxx=np.sum(np.square(x-Mx))
lm3=Sxly/Sxx
ld3=Mly-lm3*Mx
#Rücktransformation der Werte
m3=np.exp(lm3)
d3=np.exp(ld3)
```

Berechnung ergibt: $y = C \cdot a^x \Rightarrow y = 9171 \cdot 0.99^x$

Bessere Näherung, noch nicht optimal

2. Möglichkeit: $y = a \cdot x^b$




```
#Tranformation der x- und y-Werte
lx=np.log(x)
#Neuer Plot
plt.figure(3)
m4,d4 = np.polyfit(lx, ly, 1)
plt.plot(lx,ly,'o',lx,m4*lx+d4,'r')
plt.xlabel('ln(x)')
plt.ylabel('ln(y)')
```

```
#Regression mit den logarithmierten Werten
n=x.size
Mlx = 1/n*sum(lx)
#Mly = 1/n*sum(ly)
Sxly=sum((lx-Mx)*(ly-Mly))
Slxlx=sum(np.square(lx-Mlx))
m5=Sxly/Slxlx
ld5=Mly-m5*Mlx
#Rücktransformation der Werte
m5
d5=np.exp(ld5)
```

Berechnung ergibt:

$y = a \cdot x^b \Rightarrow y = 793454.49 \cdot x^{-0.4822}$

Mehrere Variablen

Aufgabe 7

In einer Region soll der Gasverbrauch (gasv) aufgrund des Gaspreises (gpr) und des Fernwärmepreises (fpr) modelliert werden. Es wird das Regressionsmodell

gasv = a·gpr + b·fpr + c aufgestellt.

a) Bestimmen Sie mit Python die Parameter a,b,c.

b) Berechnen Sie die geschätzten Werte für den Gasverbrauch sowie die Residuen, stellen Sie die Werte in einer Tabelle zusammen

c) Zeichnen Sie einen Residuenplot (Residuen gegen die Schätzwerte) und beurteilen Sie damit das Modell.

gasv	10	10.6	10.4	11.1	11.9	13.8	13.7	13.7	12.2	12.9	13.6	13.8	13.6	13.6	13.8
gpr	0.92	1.04	1.15	1.11	1.08	1.11	1.05	0.84	0.80	0.80	0.82	0.85	0.83	0.80	0.78
fpr	0.90	1.04	1.08	1.11	1.10	1.11	1.14	1.07	1.02	1.00	1.01	1.02	1.00	0.97	0.95

Lösung:

a)

```
import numpy as np
import matplotlib.pyplot as plt

#Aufgabe 6.7
#Gaspreis
gasv=np.array([10, 10.6, 10.4, 11.1, 11.9, 13.8, 13.7, 13.7, 12.2, 12.9, 13.6, 13.8,
13.6, 13.6, 13.8])
#Gaspreis
gpr=np.array([0.92, 1.04, 1.15, 1.11, 1.08, 1.11, 1.05, 0.84, 0.80, 0.80, 0.82, 0.85,
0.83, 0.80, 0.78])
#Fernwärmepreis
fpr=np.array([0.90, 1.04, 1.08, 1.11, 1.10, 1.11, 1.14, 1.07, 1.02, 1.00, 1.01, 1.02,
1.00, 0.97, 0.95 ])
#Darstellung der Daten
fig1=plt.figure(1)
ax = fig1.add_subplot(111, projection='3d')
for k in range(0, gasv.size,1):
    ax.scatter(gpr[k],fpr[k],gasv[k],c='b', marker='o')

#Lineare Regression 3d
#minimize Sum((gasv[i] - (a*gpr[i] + b*fpr[i] + c))^2)
#then this is a standard linear algebra problem.
A = np.column_stack([gpr, fpr, np.ones_like(gpr)])
abc, residuals, rank, s = np.linalg.lstsq(A, gasv, rcond=-1)
a=abc[0]
b=abc[1]
c=abc[2]c=R(3)
```

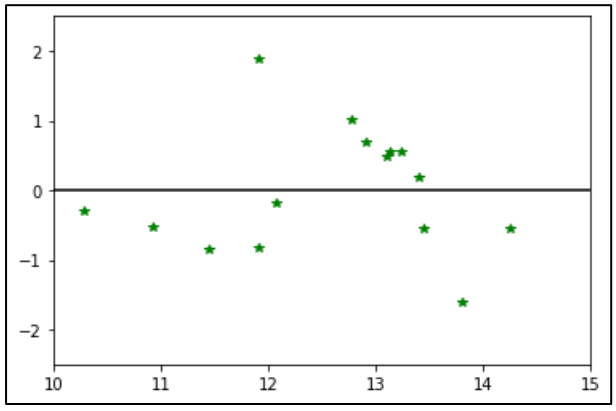
Ergibt: gasv = –11.34·gpr + 18.02·fpr + 4.50

b)

Gasverbrauch	BerechneterVerbrauch	Residuen
10.0	10.289	-0.289
10.6	11.451	-0.851
10.4	10.924	-0.524
11.1	11.919	-0.819
11.9	12.079	-0.179
13.8	11.919	1.881
13.7	13.14	0.56
13.7	14.26	-0.56
12.2	13.813	-1.613
12.9	13.452	-0.552
13.6	13.406	0.194
13.8	13.246	0.554
13.6	13.112	0.488
13.6	12.912	0.688
13.8	12.778	1.022

```
#Prognostizierte Preise
Bergasv=a*gpr+b*fpr+c
Res=gasv-Bergasv
columns = ("Gasverbrauch",
"BerechneterVerbrauch", "Residuen")
cellText=np.column_stack([gasv,
Bergasv.round(3), Res.round(3)])
plt.figure(2)
plt.axis('off')
plt.table(cellText=cellText,colLabel
s=columns,loc='center' )
```

c)




```
#Residuenplot
plt.figure(3)
plt.plot(Bergasv,Res,'g*')
plt.xlim([10,15])
plt.ylim([-2.5,2.5])
plt.plot([10,15],[0,0],'k')
```