Ausgleichsgeraden, Regressionskurven

Prof. Dr. Dörte Haftendorn, MuPAD 4, http://haftendorn.uni-lueneburg.de Aug.06

Automatische Übersetzung aus MuPAD 3.11, Nov. 05 Update 9.11.0

Es fehlen nocht textliche Änderungen, die MuPAD 4 direkt berücksichtigen, das ist in Arbeit.

Web: http://haftendorn.uni-lueneburg.de www.mathematik-verstehen.de

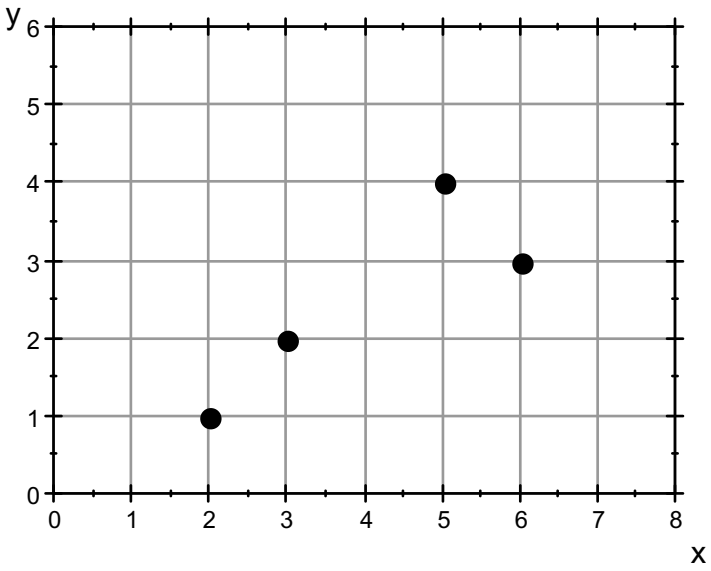
++

```
datenPunkte:=[2,1], [3,2],[5,4],[6,3]:
dp:=datenPunkte;
```

$$[2, 1], [3, 2], [5, 4], [6, 3]$$

Gesucht ist eine "beste" Gerade durch die Datenpunkte

```
graphDatenPunkte:=plot::Listplot([datenPunkte],
   LinesVisible=FALSE, PointSize=3,Scaling=Constrained,
    GridVisible=TRUE,ViewingBox=[0..8,0..6]):
plot(graphDatenPunkte)
```



Umwandlung der Information über die Daten
in x-Liste und y-Liste

```
anz:=nops(dp)
```

$$4$$

```
xd:=[dp[i][1]$i=1..anz];
yd:=[dp[i][2]$i=1..anz];
```

$$[2, 3, 5, 6]$$

$$[1, 2, 4, 3]$$

Für die lineare Regression gibt es einen eigenen Befehl.

Sie hat das Prinzip der kleinsten Fehlerquadratsumme als Grundlage.

```
reg:=stats::linReg(xd,yd)
```

$$\left[\left[\frac{1}{10}, \frac{3}{5}\right], \frac{7}{5}\right]$$

[[y-Abschnitt, Steigung], Fehlerquadratsumme]

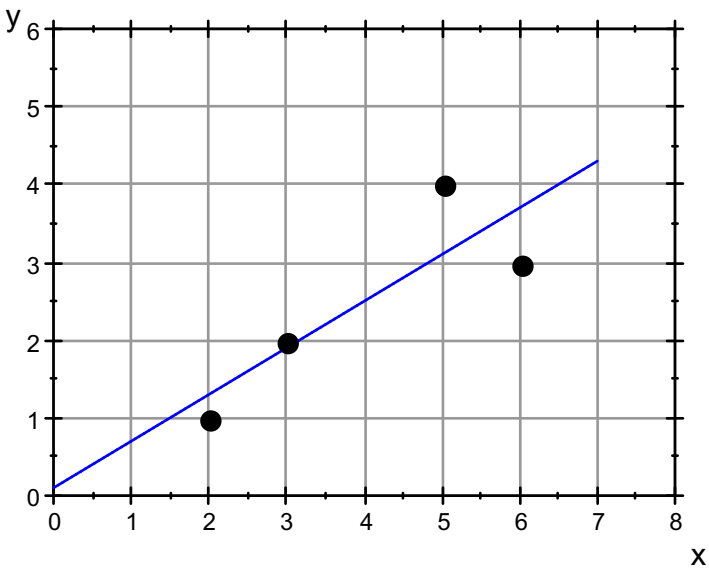
```
b:=reg[1][1]: m:=reg[1][2]:
g:=x->m*x+b; g(x); float(%)
```

$$x \to m \cdot x + b$$

$$\frac{3 \cdot x}{5} + \frac{1}{10}$$

$$0.6 \cdot x + 0.1$$

```
grreg:=plot::Function2d(g(x),x=0..7):
plot(grreg,graphDatenPunkte)
```



##

Beliebige Regressionskurven

Prinzip ist, dass hinter den Daten der gesuchte Funktionstyp mit Parametern angegeben wird. Danach wird die Variablenliste und die Paramtererliste genannt.

```
xd,yd
```

$$[2, 3, 5, 6], [1, 2, 4, 3]$$

Exponentialfunktion $f(x) = c \cdot e^{k\,x}$

```
regex:=stats::reg(xd,yd,c*exp(k*x), [x], [c, k])
```

$$[[1.021440328,\ 0.2120924253],\ 1.841020551]$$

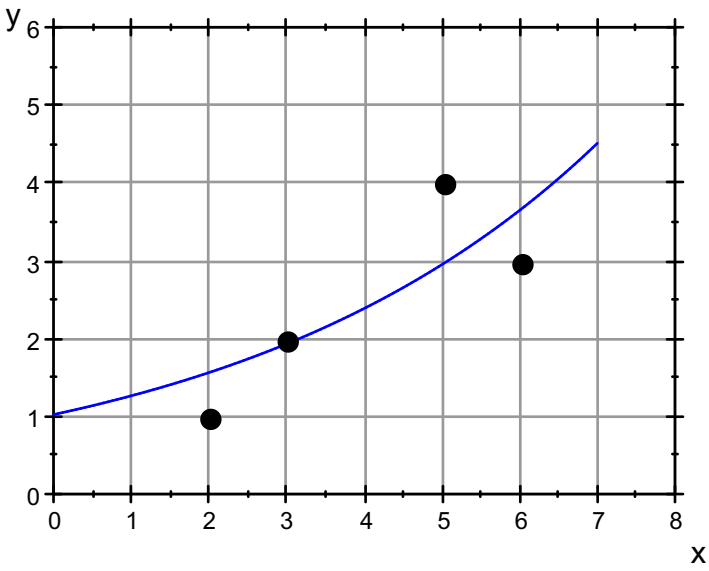
```
c:=regex[1][1]: k:=regex[1][2]:
ex:=x->c*exp(k*x); ex(x); float(%)
```

$$x \to c \cdot e^{k \cdot x}$$

$$1.021440328 \cdot e^{0.2120924253 \cdot x}$$

$$1.021440328 \cdot e^{0.2120924253 \cdot x}$$

```
grregex:=plot::Function2d(ex(x),x=0..7):
plot(grregex,graphDatenPunkte)
```



Potenzfunktion $f(x) = c \cdot x^k$

```
regpot:=stats::reg(xd,yd,c*x^k, [x], [c, k])
```

$$[[0.7383927614,\ 0.8912019613],\ 1.366549982]$$

```
cp:=regpot[1][1]: kp:=regpot[1][2]:
pot:=x->cp*x^kp; pot(x); float(%);
```

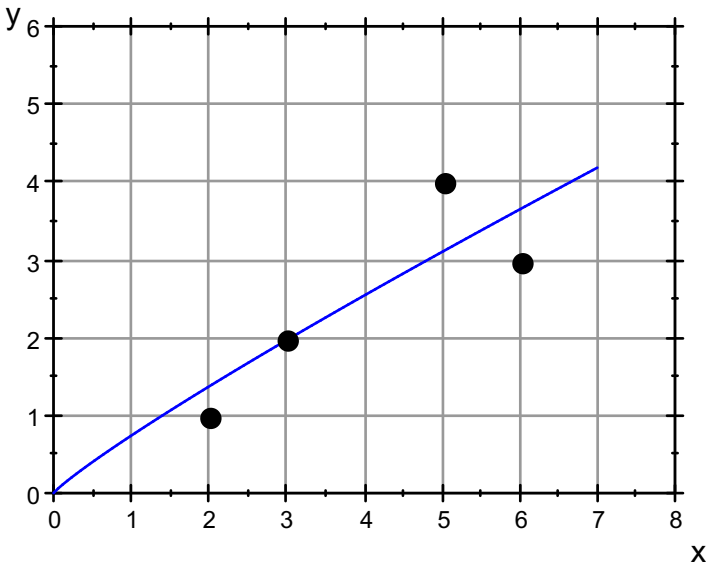
$$x \to cp \cdot x^{kp}$$

$$0.7383927614 \cdot x^{0.8912019613}$$

$$0.7383927614 \cdot x^{0.8912019613}$$

```
grregpot:=plot::Function2d(pot(x),x=0..7):
```

```
plot(grregpot,graphDatenPunkte)
```



Parabel $f(x)=a\cdot x^2+b\cdot x+c$

```
regpar:=stats::reg(xd,yd,A*x^2+B*x+C, [x], [A, B,C])
```

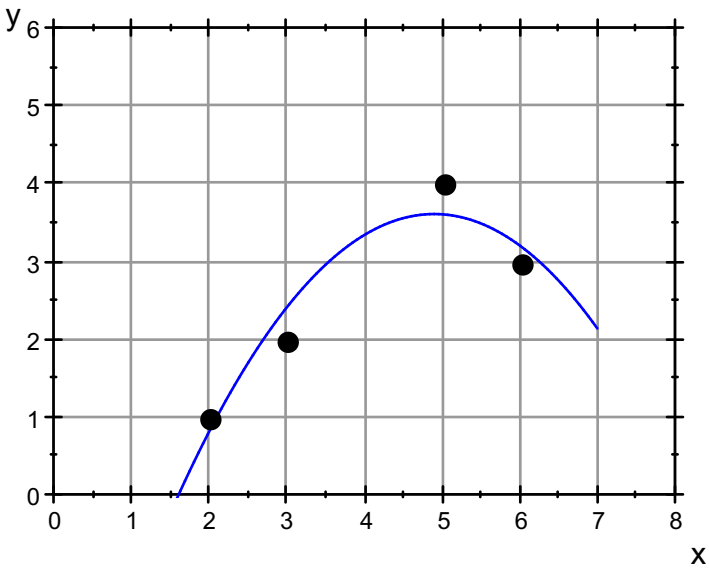
$$[[-0.3333333333,\ 3.266666667,\ -4.4],\ 0.4]$$

```
A:=regpar[1][1]: B:=regpar[1][2]: C:=regpar[1][3]:
par:=x->A*x^2+B*x+C; par(x)
```

$$x \to A\cdot x^2 + B\cdot x + C$$

$$-0.3333333333\cdot x^2 + 3.266666667\cdot x - 4.4$$

```
grregpar:=plot::Function2d(par(x),x=0..7):
plot(grregpar,graphDatenPunkte)
```


y
6
5
4
3
2
1
0
0
1
2
3
4
5
6
7
8
x