

Boxdimension

Prof. Dr. Dörte Haftendorn: Mathematik mit MuPAD 4, Update Jan 08

<http://haftendorn.uni-lueneburg.de>

www.mathematik-verstehen.de

```
lg:=x->log(10,x):
```

Nikolaushaus

Methode 1,

k= Anzahl der in der Länge getroffenen Pixel, ausgezählt

m=Anzahl der von dem Fraktal getroffenen Pixel

```
m:=[214,122,94,61];  
k:=[40,26,19,13];  
lgm:=float(map(m,lg));  
lgk:=float(map(k,lg));
```

```
[214, 122, 94, 61]
```

```
[40, 26, 19, 13]
```

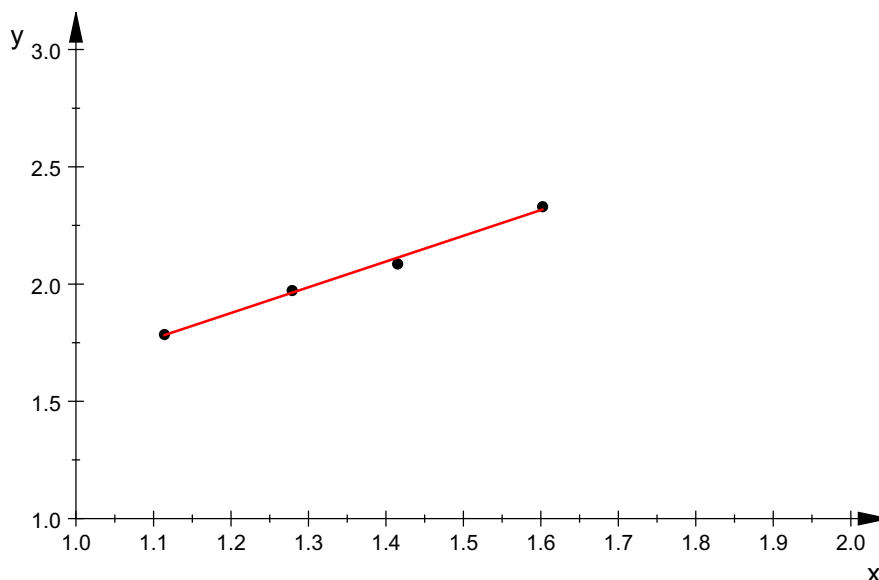
```
[2.330413773, 2.086359831, 1.973127854, 1.785329835]
```

```
[1.602059991, 1.414973348, 1.278753601, 1.113943352]
```

Direkt auftragen, gleich mit Regressionsgerade, (auch gut für Excel geeignet)

```
pktp:=plot::Scatterplot(lgk,lgm,ViewingBox=  
[1..2,1..3]);  
plot(pktp)
```

```
plot::Scatterplot(...)
```



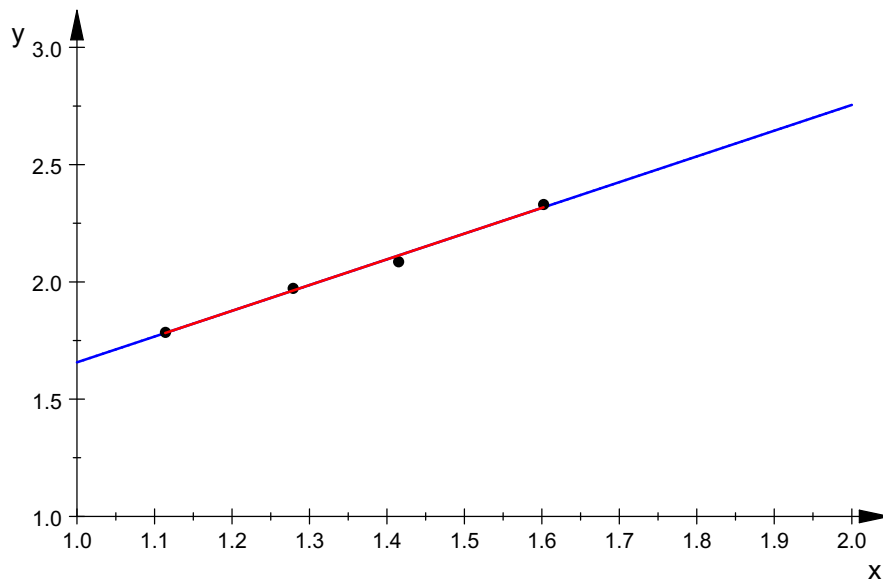
Berechnung der Regressionsgeraden

```
regger:=stats::linReg(lgk,lgm);  
reggerade:=plot::Function2d(regger[1][2]*x+regger[1]  
[1],x=0..2);  
plot(reggerade,pktp)
```

```
plot (reggerade ,pktp)
```

```
[[0.559564399, 1.09746205], 0.0009546116321]
```

```
plot::Function2d(1.09746205 · x + 0.559564399, x = 0 ..2)
```



Nikolaushaus

Methode 2,

k= Anzahl der in der Länge getroffenen Pixel,

Ausgerechnet aus Gesamterstreckung 600 Pixel und

Rasterweite w in Pixeln, wie sie auf meinen Blättern angeben ist.

m=Anzahl der von dem Fraktal getroffenen Pixel

```
m:=[214,122,94,61];  
w:=[15,24,32,48];  
k:=float([600/w[i] $ i=1..4]);  
lgm:=float(map(m,lg));  
lgk:=float(map(k,lg));  
pktp:=plot::Scatterplot(lgk,lgm);  
plot(pktp)
```

```
[214, 122, 94, 61]
```

```
[15, 24, 32, 48]
```

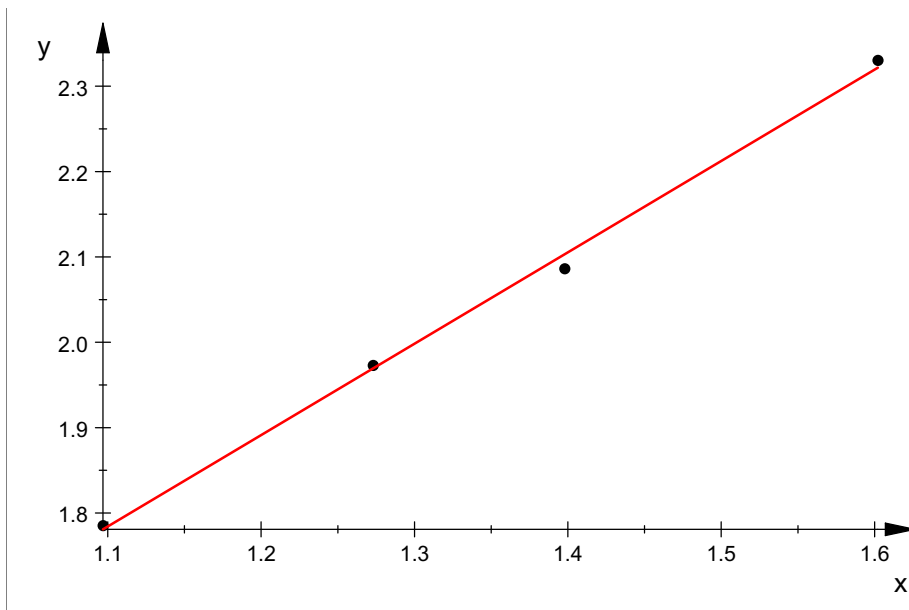
```
[40.0, 25.0, 18.75, 12.5]
```

```
[2.330413773, 2.086359831, 1.973127854, 1.785329835]
```

```
[1.602059991, 1.397940009, 1.273001272, 1.096910013]
```

```
plot::Scatterplot(...)
```





Berechnung der Regressionsgeraden

```
regger:=stats::linReg(lgk,lgm);
regger[1][1]
[[0.6067473036, 1.070453826], 0.0003920471857]
0.6067473036
```

Zusätzliche Zeichnung

```
reggerade:=plot::Function2d(regger[1][2]*x+regger[1]
[1],x=0..2);
plot(reggerade,pktp)
plot::Function2d(1.070453826 · x + 0.6067473036, x = 0 ..2)
```

