Beispiel: Lineare Regression mit Mathematica

Daten aus einem csv-File einlesen

```
data = Import[
   "/Users/ihn/Documents/Teaching/VP-Leitung/DataAnalysis/2013NewProgram/Lectures
     given in 2013/6. Lecture/daten.csv"];
```

Daten in Tabellenform darstellen

In der einfachsten Form lassen sich die Daten mit dem Grid-Befehl darstellen.

```
Grid[data]
```

```
1  4.52318
2  6.07091
3  9.60907
4  14.1273
5  15.3875
6  18.9468
7  20.6456
8  23.6468
```

Dabei gibt es viele Optionen (siehe *Mathematica* Hilfe), welche die Darstellung übersichtlicher machen können:

```
Grid[Prepend[data, {Style["x", Bold], Style["y", Bold]}],
 Background → {None, {Gray, {LightGray, White}}},
 Dividers → {Black, {Black, Black}}, Frame → True]
```

x	y
1	4.52318
2	6.07091
3	9.60907
4	14.1273
5	15.3875
6	18.9468
7	20.6456
8	23.6468

Mit Datenlisten arbeiten

Länge der Datenliste (Zahl der Datenpunkte) ermitteln:

```
n = Length[data]
```

```
8
```

Auf einzelne Elemente (Datenpunkte) der Datenliste zugreifen:

```
data[[3]]
```

```
{3, 9.60907}
```

Auf Bereiche in der Datenliste zugreifen:

```
data[[Range[2, 4]]]
```

```
{{2, 6.07091}, {3, 9.60907}, {4, 14.1273}}
```

Nur die x-Werte der Datenpunkte 2 bis 4:

```
data[[Range[2, 4], 1]]
```

```
{2, 3, 4}
```

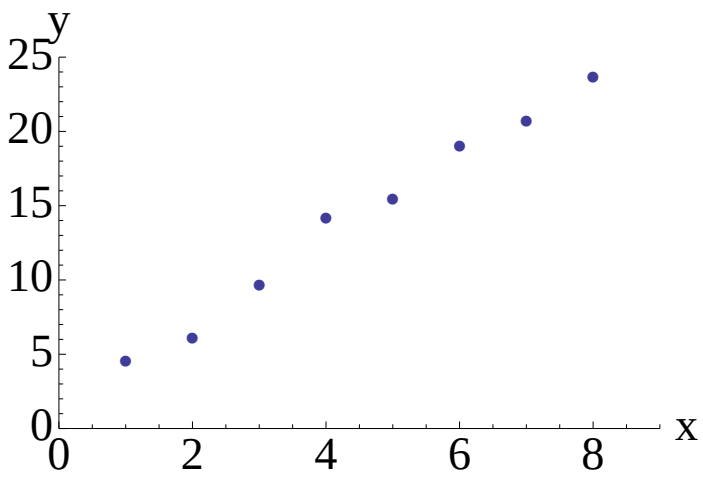
Nur die y-Werte der Datenpunkte 2 bis 4:

```
data[[Range[2, 4], 2]]
```

```
{6.07091, 9.60907, 14.1273}
```

Datenliste graphisch darstellen

```
g1 = ListPlot[data, PlotRange → {{0, 9}, {0, 25}},
  PlotMarkers → Automatic, LabelStyle → 24, AxesLabel → {"x", "y"}]
```



Statistik der Datenliste

Mittelwert der x-Werte:

```
Meanx = Mean[data[[Range[1, n], 1]]]
```

$$\frac{9}{2}$$

Mittelwert der y-Werte:

```
Meany = Mean[data[[Range[1, n], 2]]]
```

```
14.1196
```

Varianz der x-Werte:

```
Varx = CentralMoment[data[[Range[1, n], 1]], 2]
```

$$\frac{21}{4}$$

Varianz der y - Werte :

```
Vary = CentralMoment[data[[Range[1, n], 2]], 2]
```

```
41.9354
```

Empirischer Korrelationskoeffizient:

$$\rho = \frac{\text{CentralMoment[data, \{1, 1\}]}}{\sqrt{\text{Varx Vary}}}$$

```
0.994133
```

Lineare Regression, Variante I: Schätzwerte und ihre Unsicherheiten (gemäss Vorlesung)

$$A_0 = \sqrt{\frac{\text{Vary}}{\text{Varx}}}\ \rho$$

```
2.80967
```

$$B_0 = \text{Meany}$$

```
14.1196
```

$$C_0 = \text{Vary}\ \left(1 - \rho^2\right)$$

```
0.490655
```

$$\sigma_A = \sqrt{\frac{C_0}{\text{Varx}\ (n - 2)}}$$

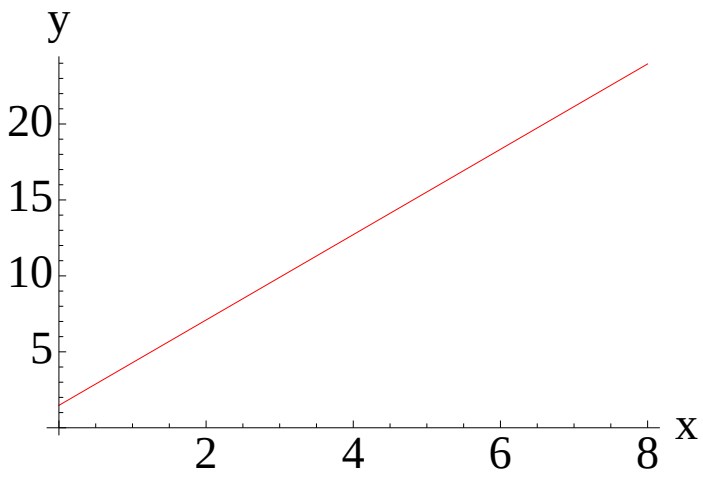
```
0.124805
```

$$\sigma_B = \sqrt{\frac{C_0}{(n - 2)}}$$

```
0.285965
```

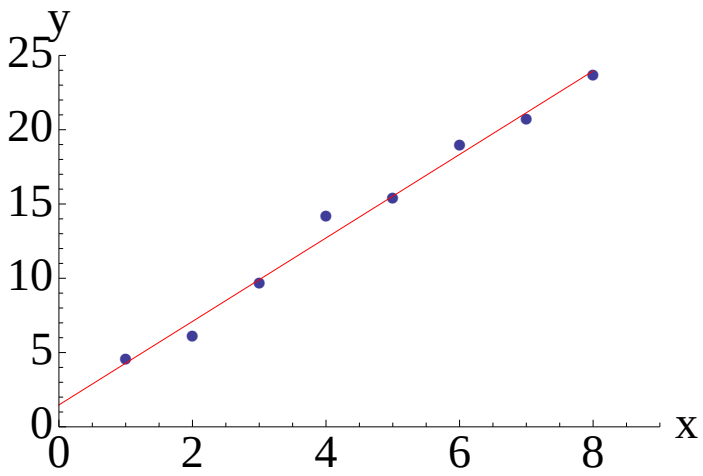
Fitkurve graphisch darstellen

```
g2 = Plot[A0 (x - Meanx) + B0, {x, 0, 8},
  PlotStyle → Red, AxesLabel → {"x", "y"}, LabelStyle → 24]
```



Daten mit Fitkurve graphisch darstellen

```
Show[g1, g2]
```



Variablen löschen

```
n =.;
```

Lineare Regression, Variante II: mit LinearModelFit-Funktion

Mit dem folgenden “high-level” Befehl kann man eine lineare Regressionsgerade an Daten fitten:

```
model = LinearModelFit[data, x, x]
```

```
FittedModel[ 1.47614 + 2.80967 x ]
```

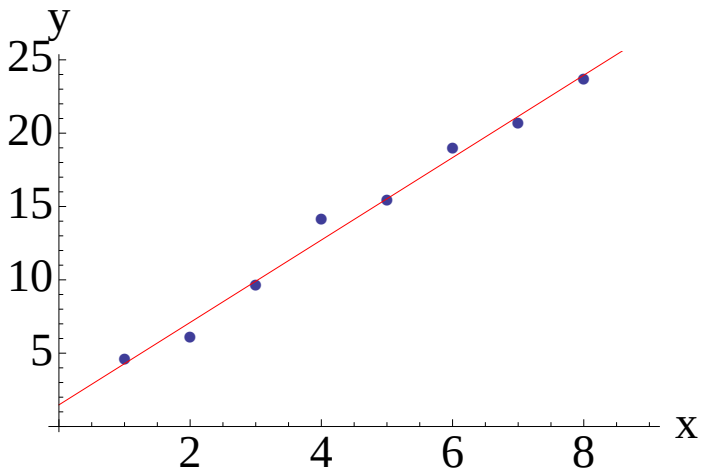
Fitkurve extrahieren:

```
model["BestFit"]
```

$$1.47614 + 2.80967\, x$$

Fitkurve darstellen:

```
Show[ListPlot[data, PlotMarkers → Automatic],
 Plot[model["BestFit"], {x, 0, 9}, PlotStyle → Red], PlotRange → {{0, 9}, {0, 25}},
 AxesOrigin → {0, 0}, AxesLabel → {"x", "y"}, LabelStyle → 24]
```



Die Fitparameter und ihre Fehler lassen sich so extrahieren :

```
model["ParameterTable"]
```

	Estimate	Standard Error	t-Statistic	P-Value
1	1.47614	0.630236	2.3422	0.0576706
x	2.80967	0.124805	22.5124	5.0275×10^{-7}

Aber Vorsicht: der Fit mit der Funktion f(x) = a x + b führt zu korrelierten Parametern a und b. Die Angabe des “Standard Error” reicht daher nicht aus, vielmehr muss eine Korrelationsmatrix angegeben werden. Dieses Problem entsteht nicht beim Fitten mit f(x) = A (x - $\overline{x}$) + B, da in diesem Fall die Parameter A und B unkorreliert sind!

Hier wäre ein Weg drumherum : wir verschieben die Daten “von Hand” um den Mittelwert von x.

```
datashifted = Table[{data[[j, 1]] - Meanx, data[[j, 2]]}, {j, 1, n}]
```

$$\left\{\left\{-\frac{7}{2}, 4.52318\right\}, \left\{-\frac{5}{2}, 6.07091\right\}, \left\{-\frac{3}{2}, 9.60907\right\}, \left\{-\frac{1}{2}, 14.1273\right\},\right.$$
$$\left.\left\{\frac{1}{2}, 15.3875\right\}, \left\{\frac{3}{2}, 18.9468\right\}, \left\{\frac{5}{2}, 20.6456\right\}, \left\{\frac{7}{2}, 23.6468\right\}\right\}$$

```
model = LinearModelFit[datashifted, x, x]
```

$$\text{FittedModel}\left[\boxed{14.1196 + 2.80967\, x}\right]$$

Jetzt stimmen die Fitparameter mit unseren Ergebnissen oben überein, und auch der “Standard Error” gibt unsere Ergebnisse von oben genau wieder.

```
model["ParameterTable"]
```

	Estimate	Standard Error	t-Statistic	P-Value
1	14.1196	0.285965	49.3755	4.62844×10^{-9}
x	2.80967	0.124805	22.5124	5.0275×10^{-7}

Fazit: benutze keine Fitroutinen irgendeiner Software, von der Du nicht *genau* weisst, was sie macht und die Du nicht an einfachsten Beispielen getestet hast! Die Zahl unkritischer Anwender von Statistiksoftware ist bereits gross genug, die Zahl der grob falschen Ergebnisse ist unermesslich.