Datenanalyse
(PHY231)
Herbstsemester 2017

Olaf Steinkamp

36-J-05 olafs@physik.uzh.ch 044 63 55763

Vorlesungsprogramm

- **Einführung, Messunsicherheiten, Darstellung von Messdaten**
- **Grundbegriffe der Wahrscheinlichkeitsrechnung und Statistik**
  - Mittelwert, Standardabweichung, Kovarianz und Korrelation
- **Fehlerfortpflanzungsgesetz**
- **Wahrscheinlichkeitsverteilungen**
  - diskrete Verteilungen, kontinuierliche Verteilungen
  - zentraler Grenzwertsatz
- **Monte-Carlo Methode**
- **Wahrscheinlichkeitsverteilungen II**
  - Faltung zweier Verteilungen
  - Verteilungen zweier Variablen
- **Stichproben und Schätzfunktionen**
  - Maximum-Likelihood Methode
  - Methode der kleinsten Quadrate
- **Interpretation von Messergebnissen**
  - Konfidenzintervalle, Testen von Hypothesen

Beispielprogramme im Verzeichnis
`/disk/puma/da/vorl/ls`

Methode kleinster Quadrate

Zweite häufig benutzte Methode zur Herleitung von Schätzfunktionen

- **Stichprobe besteht aus *N* Wertepaaren (x_i,y_i)** (i = 1,…,N)
  - Messunsicherheiten auf den x_i seien vernachlässigbar klein
  - Messunsicherheiten σ_i auf den y_i
- **Funktion $f(x|a)$ sagt für jeden x-Wert den erwarteten y-Wert vorher**
  - $f(x|a)$ hängt von einem oder mehreren Parametern a ab
  - aus der Stichprobe sollen Schätzwerte $\hat{a}$ für diese Parameter abgeleitet werden
- **definiere "Chi-Quadrat" als Funktion der a**

$$\chi^2(a) = \sum_{i=1}^{N} \frac{\left(y_i - f(x_i|a)\right)^2}{\sigma_i^2}$$

  - Maß für Diskrepanz zwischen erwarteten und beobachteten Werten
- **wähle als Schätzwerte $\hat{a}$ diejenigen Werte der Parameter, die $\chi^2(a)$ minimieren**

Beispiel Proportionalität

<u>$f(x) = a \cdot x$, suche Schätzfunktion $\hat{a}$ für den Proportionalitätsfaktor a</u>

$$\chi^2(a) = \sum_{i=1}^{N} \left(\frac{y_i - a \cdot x_i}{\sigma_i} \right)^2$$

- bestimme Minimum von $\chi^2(a)$ analytisch

$$\left[\frac{d\chi^2}{da} \right]_{a=\hat{a}} = 2 \cdot \sum_{i=1}^{N} \left\{ \frac{y_i - \hat{a} \cdot x_i}{\sigma_i^2} \cdot (-x_i) \right\} = 0$$

- Vereinfachung: Messunsicherheiten auf allen y_i seien gleich gross ($\sigma_i = \sigma$)

$$-\frac{2}{\sigma^2} \cdot \sum_{i=1}^{N} \left\{ x_i \cdot (y_i - \hat{a} \cdot x_i) \right\} = 0 \quad \Rightarrow \quad \hat{a} = \frac{\sum_{i=1}^{N} \left\{ x_i y_i \right\}}{\sum_{i=1}^{N} x_i^2} = \frac{\overline{xy}}{\overline{x^2}}$$

- Unsicherheit auf $\hat{a}$ durch Fehlerfortpflanzung

$$\frac{\partial \hat{a}}{\partial y_i} = \frac{x_i}{\sum_{i=1}^{N} x_i^2} = \frac{x_i}{N \cdot \overline{x^2}} \quad \Rightarrow \quad V(\hat{a}) = \sum_{i=1}^{N} \left\{ \left(\frac{\partial \hat{a}}{\partial y_i} \right)^2 \sigma^2 \right\} = \frac{\sigma^2}{\left(N \cdot \overline{x^2} \right)^2} \cdot \sum_{i=1}^{N} x_i^2 = \frac{\sigma^2}{N \cdot \overline{x^2}}$$

Beispiel Anpassung einer Geraden

$f(x) = a_0 + a_1 \cdot x$, suche Schätzfunktionen für die Parameter a_0 und a_1

$$\chi^2(a_0, a_1) = \sum_{i=1}^{N} \left(\frac{y_i - a_0 - a_1 \cdot x_i}{\sigma_i} \right)^2$$

- **Ableiten nach a_0, a_1: zwei Gleichungen für zwei Schätzfunktionen $\hat{a}_0$ und $\hat{a}_1$**
  - Vereinfachung: nehme an, Unsicherheiten auf allen y_i gleich groß, $\sigma_i = \sigma$ für alle i

$$\left[\frac{\partial \chi^2}{\partial a_1} \right]_{\substack{a_0 = \hat{a}_0 \\ a_1 = \hat{a}_1}} = -\frac{2}{\sigma^2} \cdot \sum_{i=1}^{N} \left\{ \left(y_i - \hat{a}_0 - \hat{a}_1 \cdot x_i \right) \cdot x_i \right\} = 0 \quad \Leftrightarrow \quad \overline{xy} - \hat{a}_0 \overline{x} - \hat{a}_1 \cdot \overline{x^2} = 0$$

$$\left[\frac{\partial \chi^2}{\partial a_0} \right]_{\substack{a_0 = \hat{a}_0 \\ a_1 = \hat{a}_1}} = -\frac{2}{\sigma^2} \cdot \sum_{i=1}^{N} \left\{ y_i - \hat{a}_0 - \hat{a}_1 \cdot x_i \right\} = 0 \quad \Leftrightarrow \quad \overline{y} - \hat{a}_0 - \hat{a}_1 \cdot \overline{x} = 0$$

- **Auflösen nach $\hat{a}_0$ und $\hat{a}_1$:**

$$\hat{a}_0 = \frac{\overline{x^2}\,\overline{y} - \overline{x}\,\overline{xy}}{\overline{x^2} - \overline{x}^2} \qquad \hat{a}_1 = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\overline{x^2} - \overline{x}^2}$$

Beispiel Anpassung einer Geraden

Fehlermatrix für $\hat{a}_0$ und $\hat{a}_1$ durch Fehlerfortpflanzung

- für Varianz auf $\hat{a}_1$

$$V(\hat{a}_1) = \sum_{i=1}^{N} \left\{ \left(\frac{\partial \hat{a}_1}{\partial y_i} \right)^2 \cdot \sigma_i^2 \right\} \qquad \Bigg| \quad \hat{a}_1 = \frac{\overline{xy} - \overline{x}\,\overline{y}}{\overline{x^2} - \overline{x}^2} = \frac{1}{N} \cdot \sum_{i=1}^{N} \frac{(x_i - \overline{x}) \cdot y_i}{(\overline{x^2} - \overline{x}^2)}$$

$$= \frac{\sigma^2}{N^2 \cdot (\overline{x^2} - \overline{x}^2)^2} \cdot \sum_{i=1}^{N} \left[\frac{\partial}{\partial y_i} \left\{ (x_i - \overline{x}) \cdot y_i \right\} \right]^2 \qquad \Bigg| \quad \sum_{i=1}^{N} (x_i - \overline{x})^2 = N \cdot V(x) = N \cdot (\overline{x^2} - \overline{x}^2)$$

$$= \frac{\sigma^2}{N \cdot (\overline{x^2} - \overline{x}^2)}$$

- entsprechend für Varianz auf $\hat{a}_0$ und für Kovarianz von $\hat{a}_0$ und $\hat{a}_1$ …

$$\Rightarrow \quad \operatorname{cov}(\hat{a}_0, \hat{a}_1) = \frac{\sigma^2}{N \cdot (\overline{x^2} - \overline{x}^2)} \cdot \begin{pmatrix} \overline{x^2} & -\overline{x} \\ -\overline{x} & 1 \end{pmatrix}$$

! gleiche Ergebnisse wie mit Maximum Likelihood !

Kleinste Quadrate und Maximum Likelihood

Annahme: Abweichungen der y_i von den wahren Werten gaußverteilt

- **Wahrscheinlichkeit für ein Wertepaar (x_i,y_i)**

$$p(y_i \mid a) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \cdot e^{-\frac{(y_i - f(x_i \mid a))^2}{2\sigma_i^2}}$$

- **log-likelihood für eine Stichprobe aus N Wertepaaren (x_i,y_i)**

$$\ln L(a) = \sum_{i=1}^{N} \ln p(y_i \mid a) = -\frac{1}{2} \cdot \sum_{i=1}^{N} \left(\frac{y_i - f(x_i \mid a)}{\sigma_i} \right)^2 \underbrace{- N \cdot \ln \sqrt{2\pi} - \sum_{i=1}^{N} \ln \sigma_i}_{= \text{konst}}$$

- **Maximieren von ln $L(a)$ = Minimieren von** $\displaystyle\sum_{i=1}^{N} \left(\frac{y_i - f(x_i \mid a)}{\sigma_i} \right)^2 = \chi^2(a)$

⇒ **die zwei Methoden sind identisch, wenn die Messbweichungen auf den y_i gaußverteilt sind**

⇒ **Methode kleinster Quadrate nimmt implizit an, daß die Messabweichungen auf den y_i gaußverteilt sind**

Beispiel Anpassung einer Geraden

<u>Allgemeiner Fall: Messunsicherheiten σ_i auf den y_i unterschiedlich groß</u>

- **ersetze überall die einfachen Mittelwerte durch gewichtete Mittelwerte**

$$\bar{x} = \frac{\sum_{i=1}^{N} x_i/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2} \qquad \bar{y} = \frac{\sum_{i=1}^{N} y_i/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2}$$

$$\overline{x^2} = \frac{\sum_{i=1}^{N} x_i^2/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2} \qquad \overline{xy} = \frac{\sum_{i=1}^{N} x_i y_i/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2}$$

- **ersetze in der Kovarianzmatrix σ^2 durch den gewichteten Mittelwert der σ_i^2**

$$\overline{\sigma^2} = \frac{\sum_{i=1}^{N} \sigma_i^2/\sigma_i^2}{\sum_{i=1}^{N} 1/\sigma_i^2} = \frac{N}{\sum_{i=1}^{N} 1/\sigma_i^2}$$

(Herleitungen wie für den Spezialfall gleich großer Messunsicherheiten, nur mehr Schreibarbeit)

Extrapolation der Geradengleichung

$Y = \hat{a}_0 + \hat{a}_1 \cdot X$: Unsicherheit auf Y durch Fehlerfortpflanzung

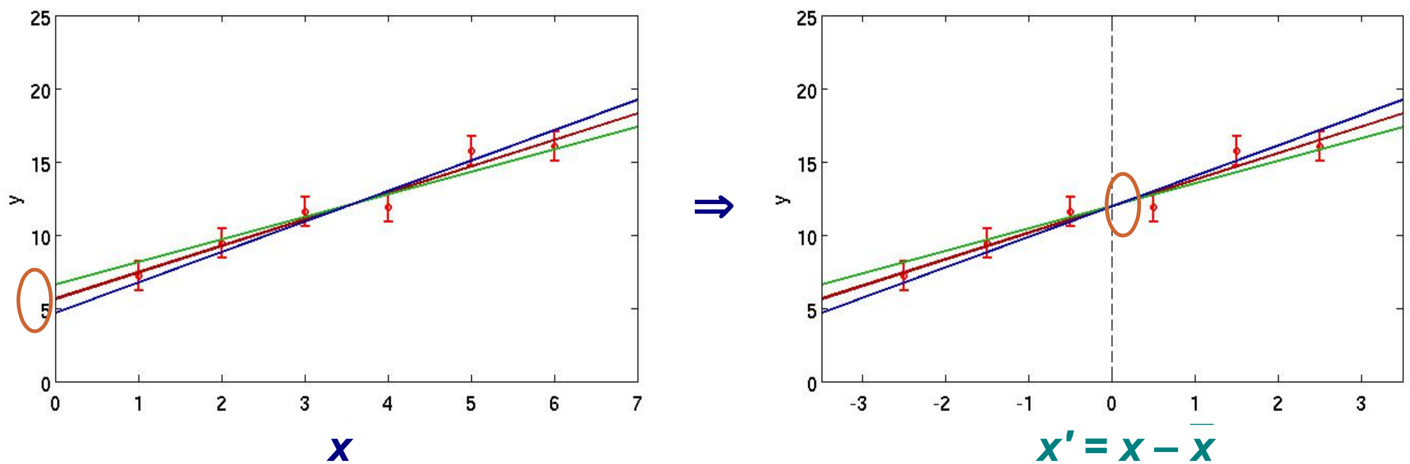
$$V(Y) = V(\hat{a}_0) + X^2 \cdot V(\hat{a}_1) + \boxed{2\,X \cdot cov(\hat{a}_0, \hat{a}_1)}$$

$$= \frac{\sigma^2}{N\left(\overline{x^2} - \overline{x}^2\right)} \cdot \left(\overline{x^2} + X^2 - \boxed{2\,\overline{x}\,X} \right)$$

- **Koordinatentransformation** $x_i \rightarrow x_i' = x_i - \overline{x}$

$$\Rightarrow \overline{x'} = 0$$

$\Rightarrow$ **Kovarianzterm verschwindet**

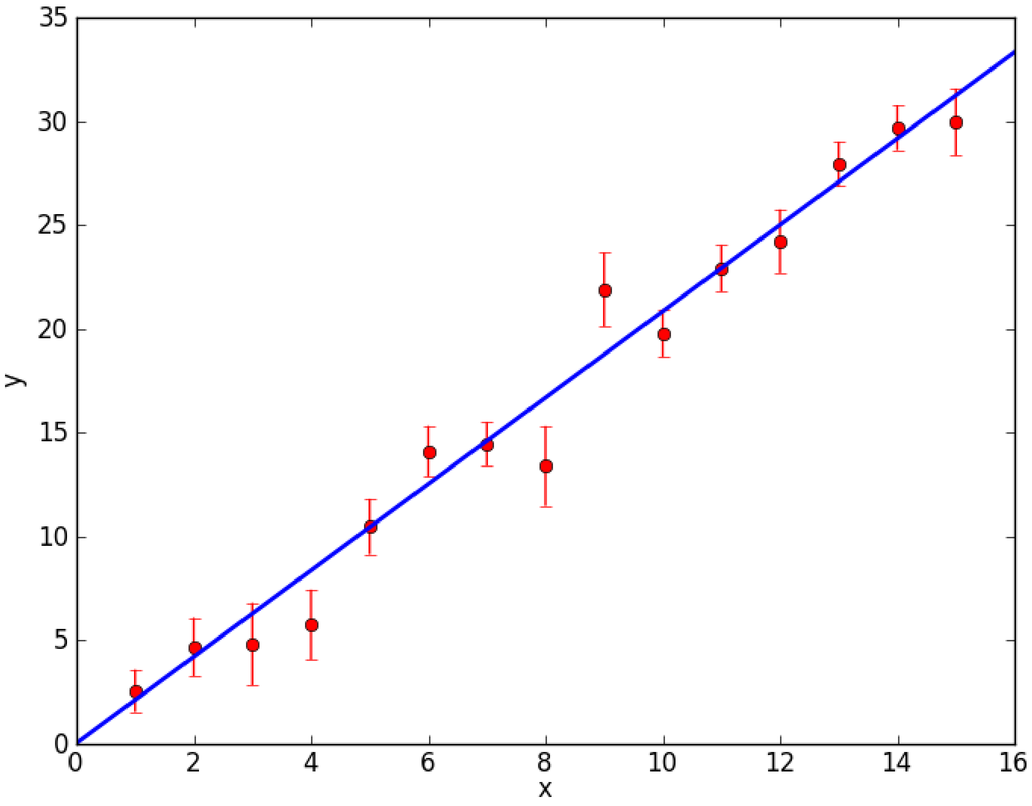


Bedeutung von χ^2

Geradenanpassung (unterschiedlich große Messunsicherheiten auf den y_i)

lsline.py

```
#!/usr/bin/env python
from pylab import *
import scipy.stats as stats
#
# erzeuge Wertepaare gemaess Geradengleichung
# mit gaussverteilten Abweichungen auf den y
#
a = 2.0 ; b = 0.5
x      = frange(1.0,15.0,1.0)
n      = len(x)
sig    = 1.0+rand(n)
y      = a*x+b+sig*stats.norm(0.0,1.0).rvs(n)
#
# berechne Schaetzwerte, Kovarianzmatrix
#
w      = sig**-2
sumw   = sum(w)
xmean  = dot(x,w)/sumw
ymean  = dot(y,w)/sumw
xymean = dot(x*y,w)/sumw
xxmean = dot(x*x,w)/sumw
varx   = xxmean-xmean**2
ahat    = (xymean-xmean*ymean)/varx
bhat    = ymean - ahat*xmean
vara    = 1.0/(sumw*varx)
varb    = vara*xxmean
covab   = -xmean*vara
#
# berechne chi2, Freiheitsgrade
#
chi2    = sum(((y-ahat*x-bhat)/sig)**2)
ndf     = n - 2
```



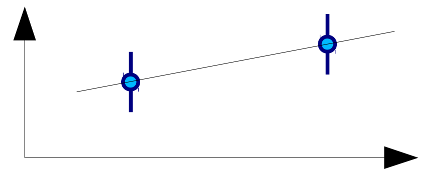

```
estimate for a =   1.942 +/- 0.074
estimate for b =   1.049 +/- 0.708
corr coeff     = -0.880
covariance matrix =   0.005 -0.046
                     -0.046  0.502

chi2 = 13.95 for ndf =  13
```

Bedeutung von χ^2

Quantitatives Maß für die Güte der Anpassung an die Daten

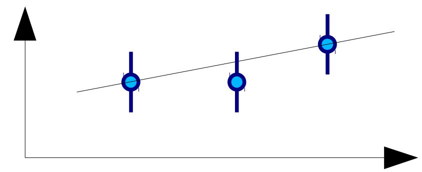
- für $f(x\,|\,a)$ mit den wahren Werte der Parameter a erwarte im Mittel

$$y_i - f(x_i\,|\,a) \approx \sigma_i \quad \Rightarrow \quad \chi^2 \approx N$$



- aber: bestimme Schätzwerte $\hat{a}$ aus den Wertepaaren

$$y_i - f(x_i\,|\,\hat{a}) < y_i - f(x_i\,|\,a) \quad \Rightarrow \quad \boxed{\chi^2 \approx n_f}$$



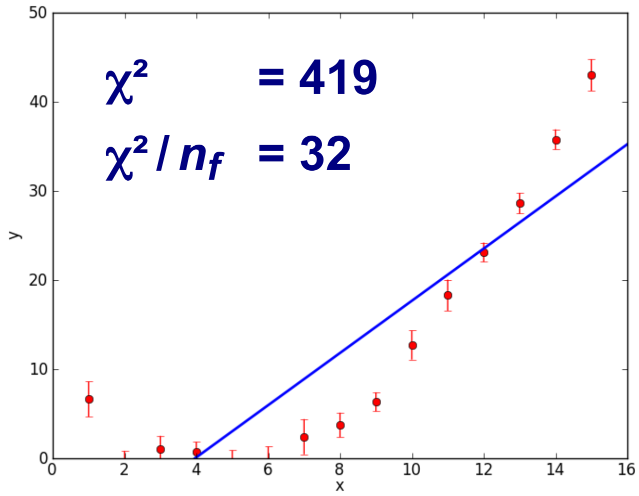
n_f = Anzahl der Messpaare – Anzahl der angepassten Parameter

- $\chi^2 \gg n_f$ ($\chi^2 / n_f \gg 1$) :
  - Messunsicherheiten unterschätzt ?
  - angenommene Form der Verteilung $f(x\,|\,a)$ falsch ?
- $\chi^2/n_f \ll 1$: Messunsicherheiten überschätzt ?

Beispiel: Bedeutung von χ^2

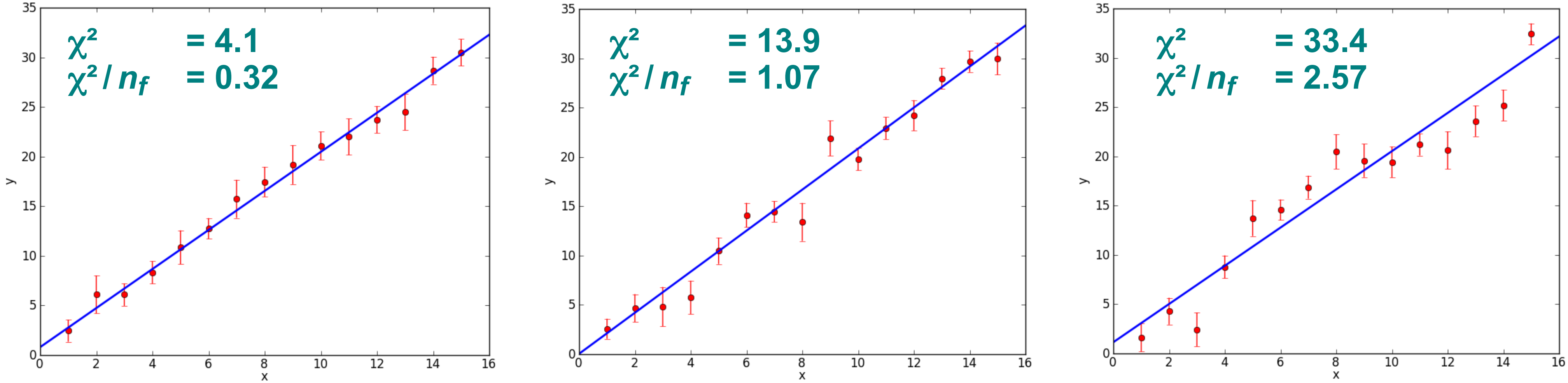
Beispiel: falsch angenommenes $f(x)$

- **erzeuge 15 (x_i,y_i)-Paare gemäß Polynom 2. Ordnung**
  - Abweichungen der y_i von $f(x_i)$ gemäß Gaußverteilung
- **passe Gerade $f(x) = a_0 + a_1 \cdot x$ an die Wertepaare an**
- **Anzahl Freiheitsgrade: $n_f = 15 - 2 = 13$**



Aber Vorsicht vor Überinterpretation: χ^2 ist Zufallsvariable

- **extreme Beispiele aus 50 Iterationen von lsline.py ($n_f = 13$):**



χ^2-Verteilung

Für gaußverteilte Messabweichungen auf den y_i:

Beweise z.B. in Barlow, “Statistics”

$$p\left(\chi^2 \mid n_f\right) = \frac{2^{-n_f/2}}{\Gamma\left(n_f/2\right)} \cdot \chi^{n_f-2} \cdot e^{-\chi^2/2}$$

- **Erwartungswert:**

$$\langle\chi^2\rangle = n_f$$

- **Varianz:**

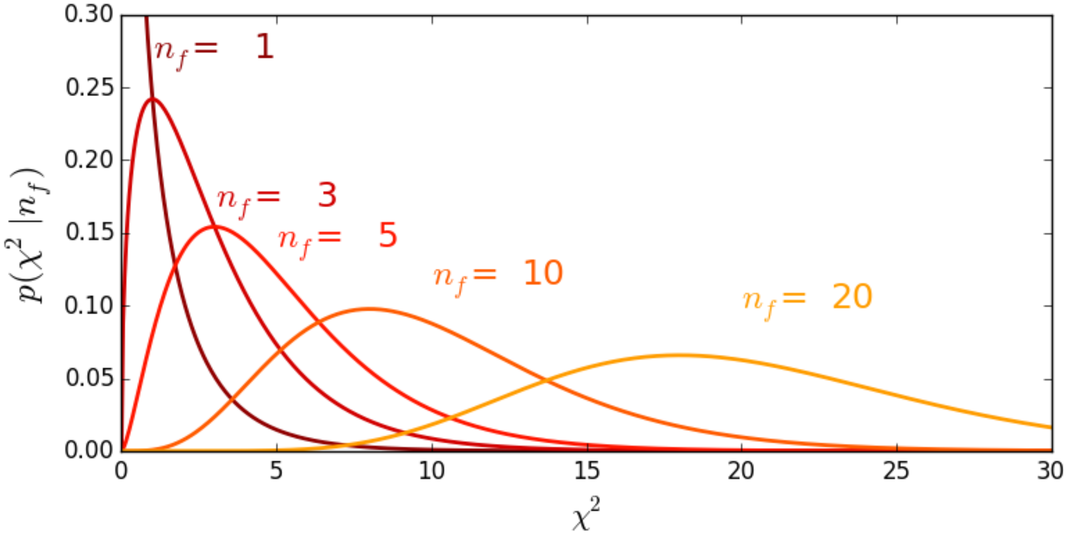
$$V(\chi^2) = 2 \cdot n_f$$

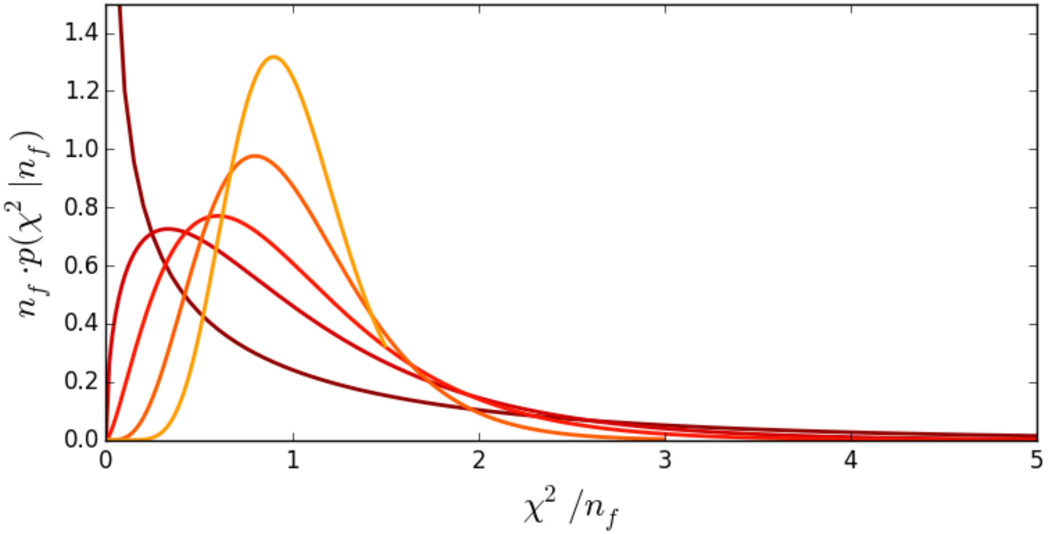
- **pylab:**

```
stats.chi2(ndf).pdf(x)
```

- **Γ–Funktion = Erweiterung der Fakultät n! auf reelle Zahlen**

$$\Gamma\left(x+1\right) \equiv x \cdot \Gamma\left(x\right) \quad \text{mit} \quad \Gamma\left(1\right) \equiv 1$$

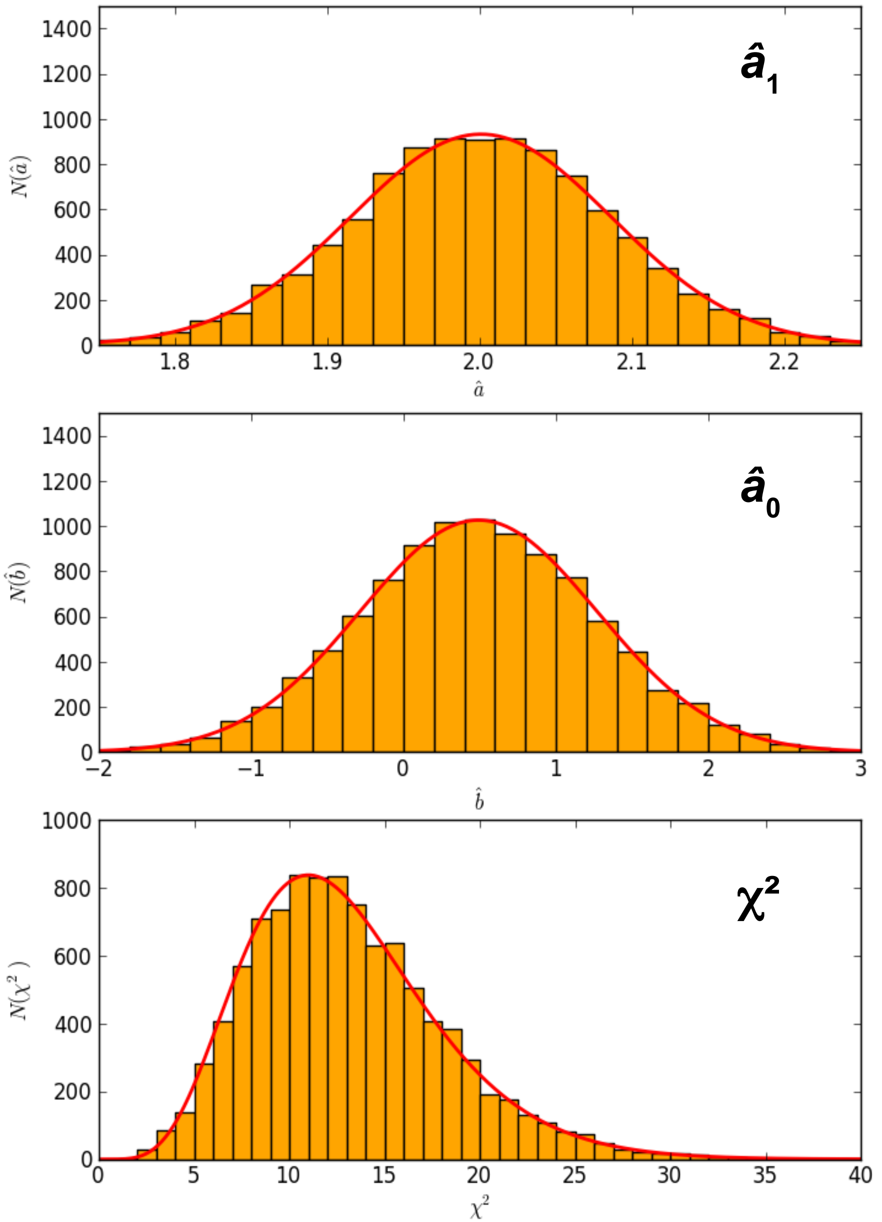




Beispiel Anpassung einer Geraden

10000 Iterationen von `lsline.py`

- **15 x_i – Werte, σ_i, a_0 und a_1 immer gleich**
- **15 y_i – Werte, für jede Iteration neu**
  - Abweichungen von $a_0 + a_1 \cdot x_i$ gemäß Gaußverteilung mit Standardabweichung σ_i
- **Histogramme: berechnete Schätzwerte und χ^2-Werte aus 10000 Iterationen**
- **Kurven: erwartete Verteilungen für Schätzwerte und χ^2**
  - **Gaußverteilung mit** $\mu_{\hat{a}_1} = a_1,\ \sigma_{\hat{a}_1} = \sqrt{V(\hat{a}_1)}$
  - **Gaußverteilung mit** $\mu_{\hat{a}_0} = a_0,\ \sigma_{\hat{a}_0} = \sqrt{V(\hat{a}_0)}$
  - **χ^2-Verteilung für $n_f = 13$**



Histogrammierte Messdaten

Betrachte Zufallsvariable *x* mit Wahrscheinlichkeitsverteilung $p(x|a)$

- **Stichprobe besteht aus n Werten x_i ($i = 1,\dots, n$)**
  - Methode kleinster Quadrate funktioniert aber nur für Wertepaare !
- **Lösung: histogrammiere die x–Werte**
  - N Intervalle mit Zentren x_i und n_i Einträgen ⇒ N Wertepaare (x_i, n_i)
- **erwartete Anzahl Einträge im i-ten Intervall des Histogramms**

$$f(x_i|a) = n \cdot \int_{x_i-\Delta x_i/2}^{x_i+\Delta x_i/2} p(x|a)\,dx \approx n \cdot \Delta x_i \cdot p(x_i|a)$$

(Δx_i : Intervallbreiten)

- **Histogramm = Zählexperiment: Anzahl Einträge poissonverteilt**

$$\Rightarrow \quad \sigma_i = \sqrt{n_i}$$

- **zu minimierendes χ^2 :**

$$\chi^2(a) = \sum_{i=1}^{N} \frac{(n_i - f(x_i\,|\,a))^2}{\sigma_i^2} = \sum_{i=1}^{N} \frac{(n_i - f(x_i\,|\,a))^2}{n_i}$$

Methode kleinster Quadrate nimmt gaußverteilte Unsicherheiten an: gute Näherung, wenn n_i nicht zu klein

Korrelierte Messwerte

Kovarianzmatrix $V(y_i,y_j)$ für die beobachteten y-Werte

$$\chi^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\left\{\left(y_i - f(x_i\,|\,a)\right)\cdot\left[V^{-1}(y_i,y_j)\right]\cdot\left(y_j - f(x_j\,|\,a)\right)\right\}$$

- in Matrixschreibweise

$$\chi^2 = (\vec{y}-\vec{f})^T \cdot V_y^{-1} \cdot (\vec{y}-\vec{f})$$

- Kontrolle: für Spezialfall unkorrelierter y_i

$$V_y = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_N^2 \end{pmatrix} \quad \Rightarrow \quad V_y^{-1} = \begin{pmatrix} 1/\sigma_1^2 & 0 & \dots & 0 \\ 0 & 1/\sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma_N^2 \end{pmatrix}$$

$$\Rightarrow \quad \chi^2 = \sum_{i=1}^{N}\left\{\left(y_i - f(x_i\,|\,a)\right)\cdot\frac{1}{\sigma_i^2}\cdot\left(y_i - f(x_i\,|\,a)\right)\right\} = \sum_{i=1}^{N}\frac{\left(y_i - f(x_i\,|\,a)\right)^2}{\sigma_i^2} \quad ✓$$

Minimieren von $\chi^2(a)$

n Parameter: Minimum von χ^2 durch Lösen von $\partial\chi^2 / \partial a_k = 0$ für $k = 1,\ldots,n$

- **n Gleichungen für die n Parameter**
- **$f(x|a)$ linear in den $a_k \rightarrow$ lineares Gleichungssystem**
- **Ansatz: schreibe $f(x_i|a)$ als Summe linearer Terme in den Parametern**

$$f(x_i|a) = \sum_{k=1}^{n} c_k(x_i) \cdot a_k \quad \text{mit} \quad c_k(x_i) = \frac{\partial f(x_i|a)}{\partial a_k}$$

$f(x|a)$ linear in den a_k: c_k hängen nicht von den a_k ab

- **in Matrixschreibweise:**

$$\vec{f} = C \cdot \vec{a} \quad \text{mit} \quad C = \begin{pmatrix} \frac{\partial f(x_1|a)}{\partial a_1} & \cdots & \frac{\partial f(x_1|a)}{\partial a_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f(x_N|a)}{\partial a_1} & \cdots & \frac{\partial f(x_N|a)}{\partial a_n} \end{pmatrix}$$

$$\Rightarrow \quad \chi^2 = \left(\vec{y} - \vec{f}\right)^T \cdot V_y^{-1} \cdot \left(\vec{y} - \vec{f}\right) = \boxed{\left(\vec{y} - C\vec{a}\right)^T \cdot V_y^{-1} \cdot \left(\vec{y} - C\vec{a}\right)}$$

Minimieren von $\chi^2(a)$

$$\chi^2 = \left(\vec{y} - C\vec{a}\right)^T \cdot V_y^{-1} \cdot \left(\vec{y} - C\vec{a}\right)$$

- **nach den *a* ableiten und = 0 setzen:**

$$-2\left(\vec{y} - C\hat{\vec{a}}\right)^T \cdot V_y^{-1} \cdot C = 0$$

- **nach den *â* auflösen:**

$$\hat{\vec{a}} = \left(C^T \cdot V_y^{-1} \cdot C\right)^{-1} \cdot C^T \cdot V_y^{-1} \cdot \vec{y}$$

Funktion *f*(*x*|*a*) *von n* Parametern
Stichprobe aus *N* Wertepaaren
=>
$\vec{y}$: Spaltenvektor mit *N* Elementen
***V*: Matrix mit *N* Reihen und Spalten**
$\vec{a}$: Spaltenvektor mit *n* Elementen
***C*: Matrix mit *N* Reihen und *n* Spalten**

Fehlermatrix für die Schätzwerte durch Fehlerfortpflanzung

- **Fehlerfortpflanzung allgemein (**siehe Folien vom 11. Oktober**)**

$$\vec{f}\left(\vec{y}\right) = G \cdot \vec{y} \qquad \Rightarrow \qquad V_f = G \cdot V_y \cdot G^T$$

- **hier:**

$$\vec{f} = \hat{\vec{a}} \quad ; \quad G = \left(C^T \cdot V_y^{-1} \cdot C\right)^{-1} \cdot C^T \cdot V_y^{-1}$$

- **einsetzen, kürzen …**

$$V_{\hat{a}} = \left(C^T\, V_y^{-1}\, C\right)^{-1}$$

Anpassung einer Geraden

Einfachster Fall: y_i unkorreliert und alle σ_i gleich gross

$$V_y = \begin{pmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2 \end{pmatrix} \qquad V_y^{-1} = \begin{pmatrix} 1/\sigma^2 & 0 & \dots & 0 \\ 0 & 1/\sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma^2 \end{pmatrix}$$

$$c_{ik} = \frac{\partial f(x_i \mid a)}{\partial a_k} = \frac{\partial (a_0 + a_1 \cdot x_i)}{\partial a_k}$$

$$C = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \qquad C^T = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix}$$

- **Kovarianzmatrix für die Schätzwerte:**

$$V_{\hat{a}} = \left[C^T V_y^{-1} C \right]^{-1} = \left[\begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \cdot \frac{1}{\sigma^2} \cdot \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_N \end{pmatrix} \right]^{-1} = \sigma^2 \cdot \begin{pmatrix} N & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \end{pmatrix}^{-1}$$

$$= \frac{\sigma^2}{N \cdot \sum_{i=1}^N x_i^2 - \left(\sum_{i=1}^N x_i \right)^2} \cdot \begin{pmatrix} \sum_{i=1}^N x_i^2 & -\sum_{i=1}^N x_i \\ -\sum_{i=1}^N x_i & N \end{pmatrix} = \boxed{\frac{\sigma^2}{N \cdot \left(\overline{x^2} - \overline{x}^2 \right)} \cdot \begin{pmatrix} \overline{x^2} & -\overline{x} \\ -\overline{x} & 1 \end{pmatrix}}$$

Anpassung einer Geraden

Einfachster Fall: y_i unkorreliert und alle σ_i gleich gross

$$C^T = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \quad ; \quad V_y^{-1} = \begin{pmatrix} 1/\sigma^2 & 0 & \dots & 0 \\ 0 & 1/\sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/\sigma^2 \end{pmatrix} \quad \Rightarrow \quad V_{\hat{a}} = \frac{\sigma^2}{N \cdot \left(\overline{x^2} - \overline{x}^2\right)} \cdot \begin{pmatrix} \overline{x^2} & -\overline{x} \\ -\overline{x} & 1 \end{pmatrix}$$

- **Schätzwerte**

$$\begin{pmatrix} \hat{a}_0 \\ \hat{a}_1 \end{pmatrix} = V_{\hat{a}} \cdot C^T \cdot V_y^{-1} \cdot \vec{y} = \frac{\sigma^2}{N \cdot \left(\overline{x^2} - \overline{x}^2\right)} \cdot \begin{pmatrix} \overline{x^2} & -\overline{x} \\ -\overline{x} & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \end{pmatrix} \cdot \frac{1}{\sigma^2} \cdot \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}$$

$$= \frac{1}{\overline{x^2} - \overline{x}^2} \cdot \begin{pmatrix} \overline{x^2} & -\overline{x} \\ -\overline{x} & 1 \end{pmatrix} \cdot \frac{1}{N} \cdot \begin{pmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N x_i y_i \end{pmatrix} = \boxed{\frac{1}{\overline{x^2} - \overline{x}^2} \cdot \begin{pmatrix} \overline{x^2}\,\overline{y} - \overline{x}\,\overline{xy} \\ \overline{xy} - \overline{x}\,\overline{y} \end{pmatrix}}$$

- **vgl. Rechnung "per Hand" (Folien 5/6)**

Taylorentwicklung von $\chi^2(a)$ um $a = \hat{a}$

$$\chi^2(a) = \chi^2(\hat{a}) + \underbrace{\sum_{i=1}^{n} \left\{ \left[\frac{\partial \chi^2}{\partial a_i}\right]_{a_i=\hat{a}_i} (a_i - \hat{a}_i) \right\}}_{=0,\ \text{weil } \hat{a} \text{ Minimum von } \chi^2(a)} + \frac{1}{2} \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} \left\{ \left[\frac{\partial^2 \chi^2}{\partial a_i \partial a_j}\right]_{\substack{a_i=\hat{a}_i \\ a_j=\hat{a}_j}} (a_i - \hat{a}_i)(a_j - \hat{a}_j) \right\}$$

- Terme höherer Ordnung verschwinden wenn $f(x\,|\,a)$ linear in den Parametern
- 2. Ableitungen:

$$\chi^2 = (\vec{y} - C\,\vec{a})^T \cdot V_y^{-1} \cdot (\vec{y} - C\,\vec{a}) \quad \Rightarrow \quad \frac{\partial^2 \chi^2}{\partial a_i \partial a_j} = 2\, C^T\, V_y^{-1}\, C = 2\, V_{\hat{a}}^{-1}$$

- Einsetzen in die Taylor-Entwicklung:

$$\boxed{\chi^2(a) = \chi^2(\vec{\hat{a}}) + \sum_{i=1}^{n} \sum_{j=1}^{n} (a_i - \hat{a}_i) \cdot \left[V^{-1}(\hat{a}_i, \hat{a}_j)\right] \cdot (a_j - \hat{a}_j)}$$

- für Funktion von nur einem Parameter a

$$\chi^2(a) = \chi^2(\hat{a}) + \frac{1}{\sigma_{\hat{a}}^2} \cdot (a - \hat{a})^2$$

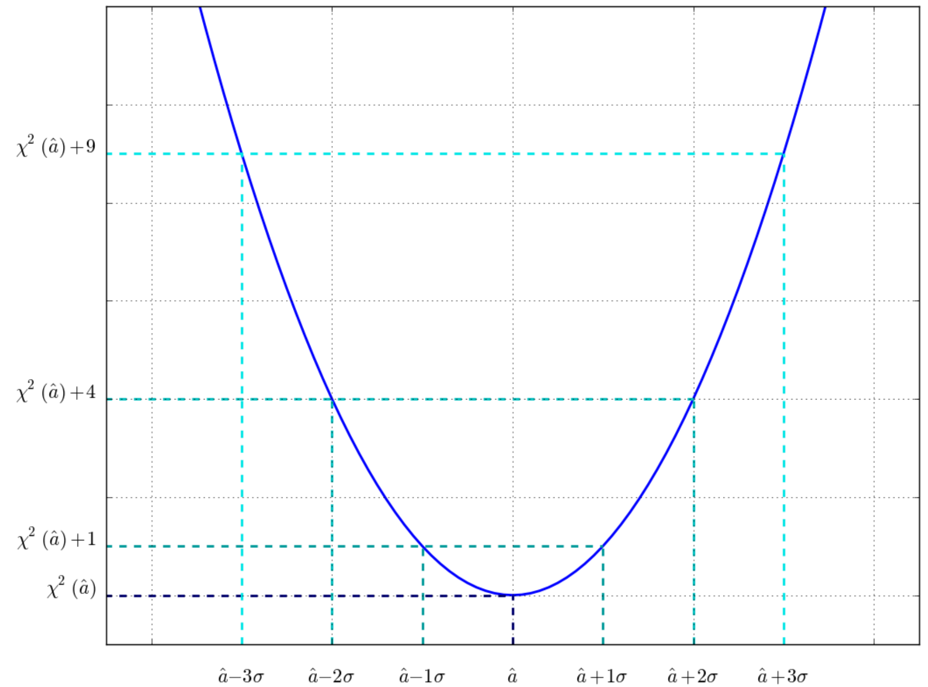
Funktion $f(x|a)$ von einem Parameter a

- $\chi^2(a)$ parabelförmig

$$\chi^2(a) = \chi^2(\hat{a}) + \frac{1}{\sigma_{\hat{a}}^2}\cdot(a-\hat{a})^2$$

- Unsicherheit $\sigma_{\hat{a}}$ aus Breite der Parabel

$$\chi^2(\hat{a}\pm\sigma_{\hat{a}}) = \chi^2(\hat{a}) + 1$$
$$\chi^2(\hat{a}\pm 2\sigma_{\hat{a}}) = \chi^2(\hat{a}) + 4$$
$$\chi^2(\hat{a}\pm 3\sigma_{\hat{a}}) = \chi^2(\hat{a}) + 9$$



Vergleich mit Maximum Likelihood

- gaußverteilte Messunsicherheiten auf den $y_i \Rightarrow$ beide Methoden identisch mit

$$\ln L(a) = -\chi^2(a)/2 + \text{konst}$$

- gaußverteilte Messunsicherheiten auf den $y_i \Rightarrow \ln L$ parabelförmig mit

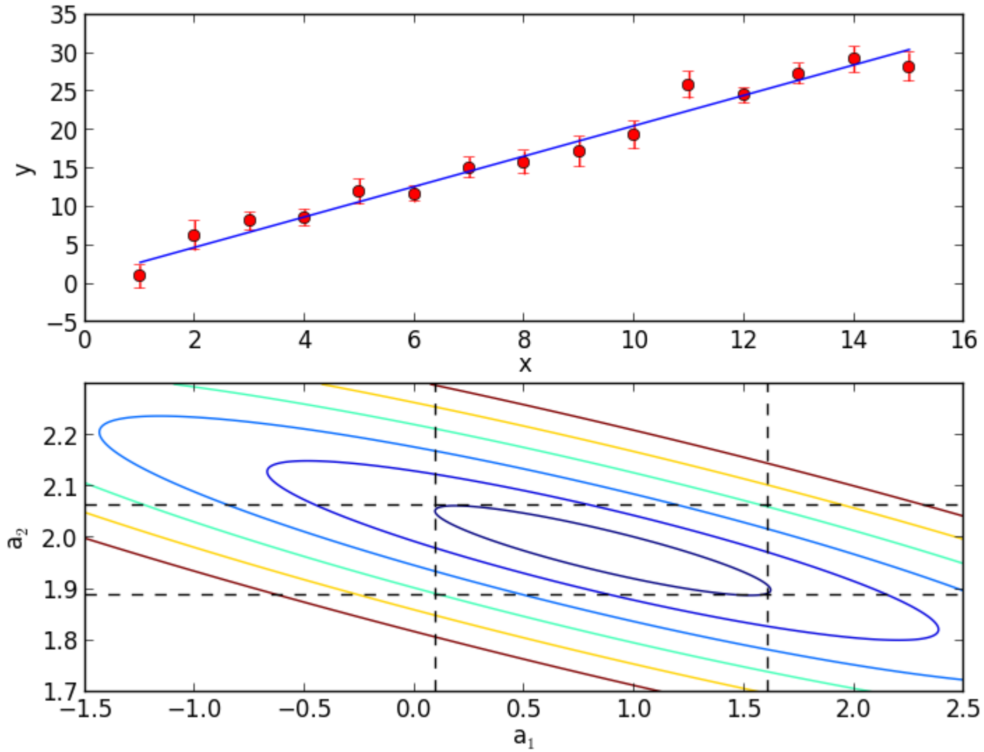
$$\ln L(\hat{a}\pm k\cdot\sigma_{\hat{a}}) = \ln L(\hat{a}) - k^2/2$$

Anpassung einer Geraden

lspoly1.py

```
#!/usr/bin/env python
from pylab import *
from scipy import stats
from linearls import *
#
# erzeuge Messpaare mit gaussverteilten Abweichungen von Gerade
#
a     = [0.5,2.0]
x     = frange(1.0,15.0,1.0)
nx    = len(x)
sigy = rand(nx)+1.0
y     = a[0] + a[1]*x+stats.norm().rvs(len(x))*sigy
#
# analytisch mit linearls
#
w       = sig**-2
sumw    = sum(w)
xmean   = dot(x,w)/sumw
ymean   = dot(y,w)/sumw
xymean = dot(x*y,w)/sumw
xxmean = dot(x*x,w)/sumw
varx    = xxmean-xmean**2
a1hat    = (xymean-xmean*ymean)/varx
a0hat    = ymean - ahat*xmean
vara1    = 1.0/(sumw*varx)
vara0    = vara1*xxmean
cova0a1 = -xmean*vara1
chi2     = sum(((y-a1hat*x-a0hat)/sig)**2)
ndf      = n - 2#
# abtasten des chi^2 Profils
#
a0t,a1t   = meshgrid(frange(-1.5,2.5,0.01),frange(1.5,2.5,0.01))
chi2      = zeros(shape(a1t))
for i in range(0,nx):
  chi2   += (y[i]-a0t-a1t*x[i])**2/err[i]**2
#
a0hat     = a0t.flatten()[argmin(chi2)]
a1hat     = a1t.flatten()[argmin(chi2)]
#
chi2 -= chi2.min()
a0lo   = a0t[chi2 <= 1].min() ; a0hi   = a0t[chi2 <= 1].max()
a1lo   = a1t[chi2 <= 1].min() ; a1hi   = a1t[chi2 <= 1].max()
```

Zwei Parameter: Fehlerellipsen




```
Parameters +/- uncertainties:
 0.820032+/-0.779911
 2.048122+/-0.085428
Covariance matrix:
[[ 0.60826108 -0.05871365]
 [-0.05871365  0.0072979 ]]
Chi^2/ndf =  16.89/  13
```

analytisch

```
a0 = 0.82+ 0.77/- 0.77
a1 = 2.05+ 0.08/- 0.08
```

χ^2-Profil

$f(x|a)$ nicht linear in den Parametern a

$\partial\chi^2 / \partial a_j$ ergibt kein lineares Gleichungssystem → keine exakte Lösung

- **Ansatz: linearisiere Problemstellung, suche iterativ näherungsweise Lösung**
  - "rate" Startwerte $\vec{a}^0 = (a_1^0, a_2^0, \ldots, a_n^0)$ für die n Parameter
  - berechne Korrekturen $\delta\vec{a}^0$, sodass $\vec{a}^1 = \vec{a}^0 + \delta\vec{a}^0$ näher beim Minimum liegt
- **Taylor-Entwicklung von $\partial\chi^2 / \partial a_j$ um die Startwerte a_j^0**

$$\left[\frac{\partial\chi^2}{\partial a_j}\right]_{\vec{a}=\vec{a}^0+\delta\vec{a}^0} \approx \left[\frac{\partial\chi^2}{\partial a_j}\right]_{\vec{a}=\vec{a}^0} + \sum_{k=1}^{n}\left[\frac{\partial^2\chi^2}{\partial a_k\,\partial a_j}\right]_{\vec{a}=\vec{a}^0}\cdot\delta a_k^0 + \ldots \qquad \text{für}\quad j=1,\ldots,n$$

  - nehme an, die $\delta\vec{a}^0$ sind nicht zu gross: breche nach linearem Term ab und setze

$$\left[\frac{\partial\chi^2}{\partial a_j}\right]_{\vec{a}=\vec{a}^0} + \sum_{k=1}^{n}\left[\frac{\partial^2\chi^2}{\partial a_k\,\partial a_j}\right]_{\vec{a}=\vec{a}^0}\cdot\delta a_k^0 = 0 \qquad \text{für}\quad j=1,\ldots,n$$

- **lineares Gleichungssystem für die $\delta\vec{a}^0$**

$$\boxed{\delta\vec{a}^0 = -G^{-1}\cdot\vec{g}} \qquad \text{mit} \quad G_{jk} = \left[\frac{\partial^2\chi^2}{\partial a_j\,\partial a_k}\right]_{\vec{a}=\vec{a}^0} \quad \text{und} \quad g_j = \left[\frac{\partial\chi^2}{\partial a_j}\right]_{\vec{a}=\vec{a}^0}$$

$f(x|a)$ nicht linear in den Parametern a

Iteriere mit $\vec{a}^1 = \vec{a}^0 + \delta\vec{a}^0$ als neuen "Startwerten"

- $\vec{a}^1$ sollten näher beim Minimum liegen als $\vec{a}^0$
- Korrekturen $\delta\vec{a}^1$ sollten kleiner sein als $\delta\vec{a}^0$

Iteriere solange bis eine Konvergenzbedingung erfüllt ist

- $\delta\vec{a}^i$ ist kleiner als ein vorgegebener Wert
- $\chi^2(a)$ ändert sich um weniger als einen vorgegebenen Wert
- eine vorgegebene maximale Anzahl Iterationen ist erreicht

Gute Startwerte $\vec{a}^0$ sind sehr wichtig

- Konvergenz ist umso schneller, je näher die $\vec{a}^0$ beim wahren Minimum liegen
- liegen die $\vec{a}^0$ zu weit vom Minimum, besteht die Gefahr, dass der Algorithmus nicht konvergiert oder in einem falschen (lokalen) Minimum konvergiert

pylab Funktion leastsq

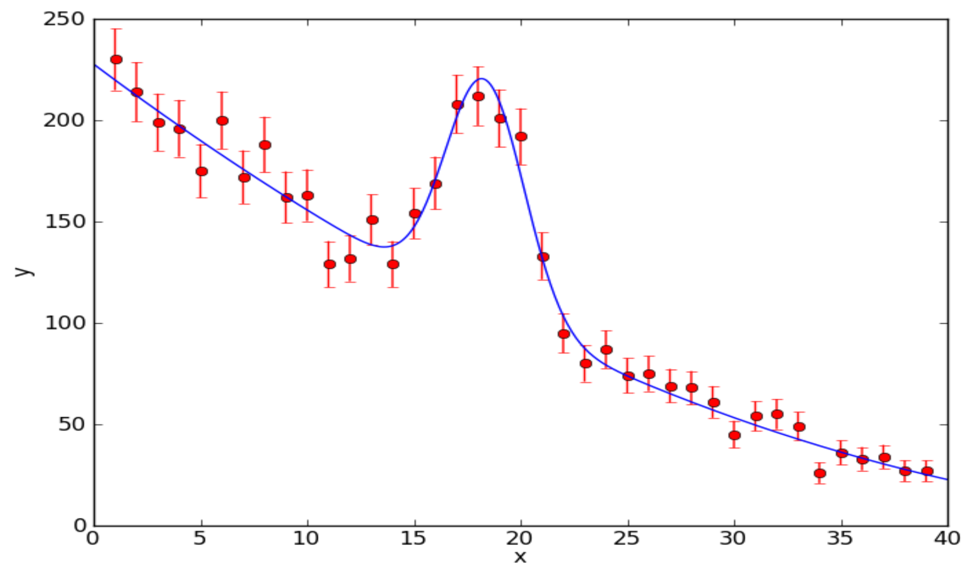
Beispiel: gaußverteiltes Signal über parabolischem Untergrund

lsgausspoly2.py

```
#!/usr/bin/env python
from pylab import *
from scipy import stats
from scipy.optimize import leastsq
#
# definiere anzupassende Funktion und
# Hilfsfunktion fuer leastsq
#
def func(a,x):
    return a[0]+a[1]*x+a[2]*x**2+\
    a[3]*stats.norm(a[4],a[5]).pdf(x)
def chi(a,x,y,yerr):
    return (func(a,x)-y)/yerr
#
# lese Messpaare von Datei
#
data = loadtxt('lsgausspoly2.dat')
x = data[:,0] ; y = data[:,1] ; yerr = y**0.5
#
# definiere Startwerte und rufe leastsq auf
#
astart = array([200,-10,0.05,500,15,1])
val    = leastsq(chi,astart,args=(x,y,yerr),
                 maxfev=10000,full_output=1)
#
# extrahiere Resultate der Anpassung
#
ahat = val[0]
cov  = val[1]
chi2 = sum(chi(ahat,x,y,yerr)**2)
ndf  = len(x)-len(astart)
#
# weitere Befehle hier nicht gezeigt
```

- input: $(f(x_i) - y_i) / \sigma_i$
- output: Schätzwerte und Kovarianzmatrix für die Fitparameter

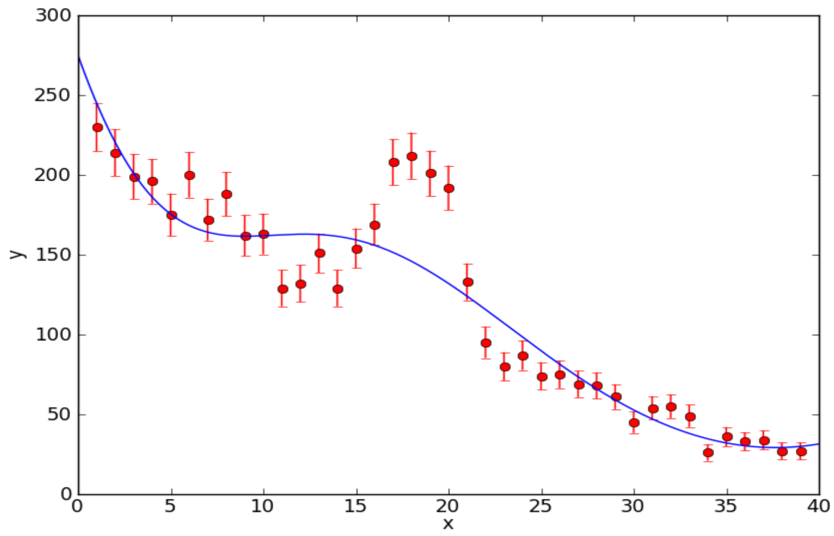
help(leastsq) in pylab

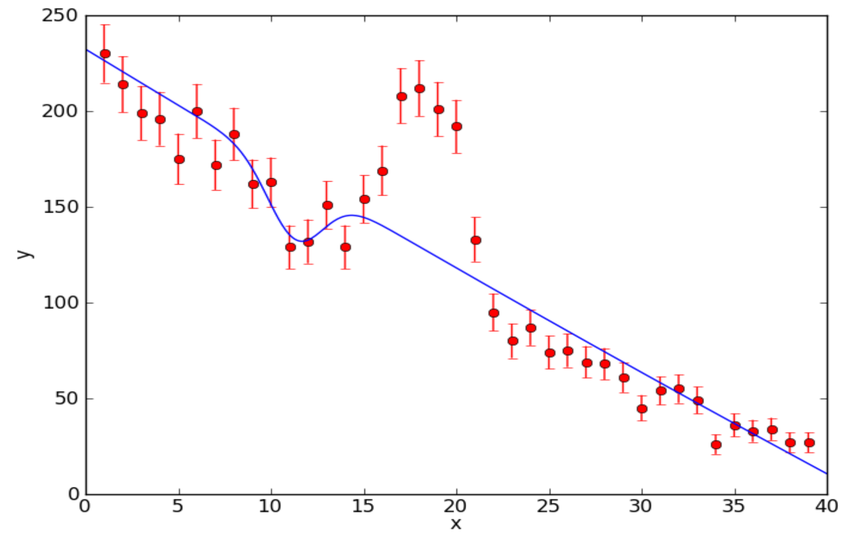



```
initial values     fitted values +/- uncertainties
    200.00              227.652 +/- 42.459
    -10.00               -7.881 +/- 0.457
      0.05                0.069 +/- 0.000
    500.00              526.533 +/- 2511.432
     15.00               18.333 +/- 0.027
      1.00                1.846 +/- 0.029
chi2/ndf 29.890/33
```

Wichtigkeit der Startwerte

Beispiele für schlechte Startwerte ⇒ Konvergenz in falschem Minimum






```
initial values     fitted values +/- uncertainties
      1.00               621.575 +/- 27837.719
      1.00               -31.436 +/- 97.038
      1.00                 0.417 +/- 0.021
      1.00             -7638.539 +/- 28822191.005
      1.00                 0.234 +/- 6.507
      1.00                 8.788 +/- 3.871
chi2/ndf 142.843/33
```

```
initial values     fitted values +/- uncertainties
    200.00               232.292 +/- 50.121
    -10.00                -5.873 +/- 0.405
      0.05                 0.008 +/- 0.000
    500.00              -126.264 +/- 1820.915
     10.00                11.303 +/- 0.225
      1.00                 1.486 +/- 0.252
chi2/ndf 181.038/33
```

Zusammenfassung

Methode kleinster Quadrate zur Herleitung von Schätzfunktionen

- **ergibt als Schätzwerte die Werte der Parameter, die die Diskrepanz zwischen vorhergesagten und beobachteten Messwerten minimieren**
- **ist für gaußverteilte Abweichungen der y_i äquivalent zu Maximum-Likelihood**

Vorteil gegenüber Maximum-Likelihood Methode

- **Wert von χ^2 im Minimum gibt quantitatives Mass für die Güte der Anpassung**
  - $\chi^2/\mathrm{ndf} \gg 1$: starker Hinweis darauf, dass etwas nicht stimmt (z.B. Messunsicherheiten unterschätzt, Funktion $f(x|a)$ beschreibt die Daten nicht)

Nachteile gegenüber Maximum-Likelihood Methode

- **kann nur auf Wertepaare (x_i,y_i) oder Histogramme (x_i,n_i) angewandt werden**
  - Erinnerung: histogrammieren → Informationsverlust
- **basiert auf der Annahme, dass die Abweichungen auf den y_i gaußverteilt sind**
  - Messabweichungen nicht gaußverteilt → benutze Maximum-Likelihood Methode