Karlsruher Institut für Technologie

Vorlesung 04b: Rechnernutzung in der Physik

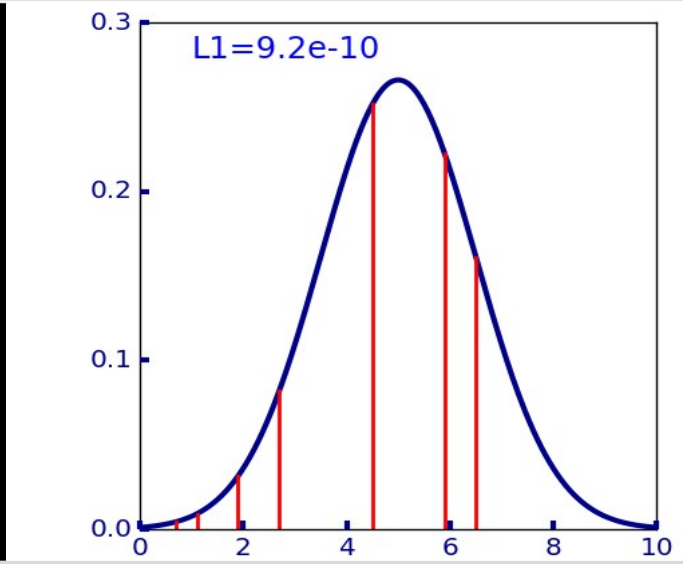
Parameteranpassung mit der Likelihood-Methode

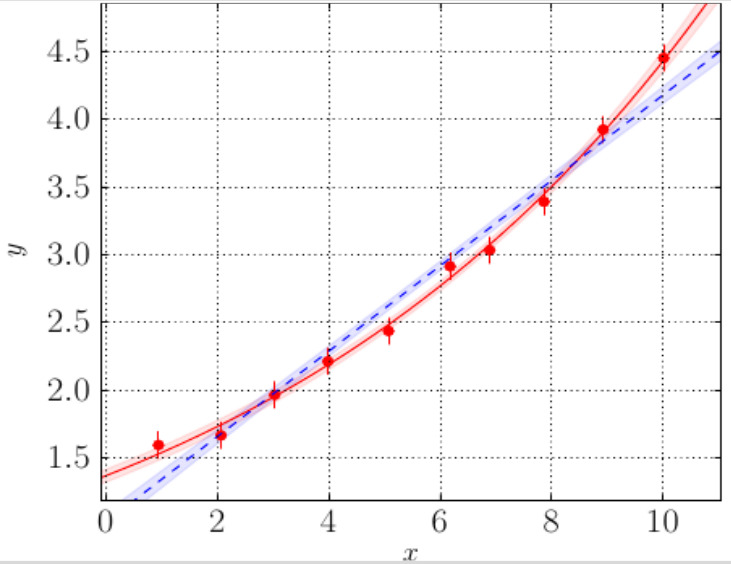
Fragen, Antworten und Beispiele

Günter Quast

Fakultät für Physik
Institut für Experimentelle Kernphysik

WS 2023/24





Beispiele

Aufgabe: Likelihood-Schätzungen

Mittelwert von Gaußverteilten Messwerten

- $f(x_i|\mu) = 1 / ((2\pi)^{1/2}\,\sigma)\;\exp[\,-(x_i - \mu)^2 / 2\sigma^2\,]$

$$f(x|a_0) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-(x_i - a_0)^2/2\sigma^2}$$

$$-\ln \mathcal{L} = -\sum_{i=1}^{N} \frac{(x_i - a_0)^2}{2\sigma^2} - N \ln(\sigma\sqrt{2\pi})$$

zweimal nach a_0 ableiten $\Rightarrow$

$$\frac{d^2 \ln \mathcal{L}}{d a_0^2} = -\frac{N}{\sigma^2}$$

$$\Rightarrow \boxed{\sigma(\hat{a}) = \frac{1}{\sqrt{N}} \cdot \sigma}$$

Aufgabe: Likelihood-Schätzungen

Mittelwert von Gaußverteilten Messwerten mit unterschiedlicher Auflösung: $f(x_i|\mu) = 1 / ((2\pi)^{1/2}\,\sigma_i)\;\exp[\,-(x_i-\mu)^2 / 2\sigma_i^2\,]$

$$F(a) = \text{const} + \frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - a)^2}{\sigma_i^2}$$

Beste Schätzung:

$$\frac{dF(a)}{da} = -\sum_{i=1}^{n}\frac{x_i - a}{\sigma_i^2} = 0$$

Führe Gewichte $w_i = \frac{1}{\sigma_i^2}$ ein $\Rightarrow$

$$\hat{a} = \frac{\sum_i w_i x_i}{\sum_i w_i} \qquad V = \sigma^2(\hat{a}) = \frac{1}{\sum_i w_i}$$

bester Mittelwert = gewichtetes Mittel

$$\sigma(\hat{a}) = \left(\frac{d^2F}{da^2}\bigg|_{\hat{a}}\right)^{-1/2} = \frac{1}{\sqrt{\sum_i w_i}} = \frac{1}{\sqrt{\sum_i 1/\sigma_i^2}}$$

Beispiel für Likelihood-Schätzungen

$$P(\vec{x}\,|\{a\}) = N(\vec{x}\,|\mu,\sigma)$$

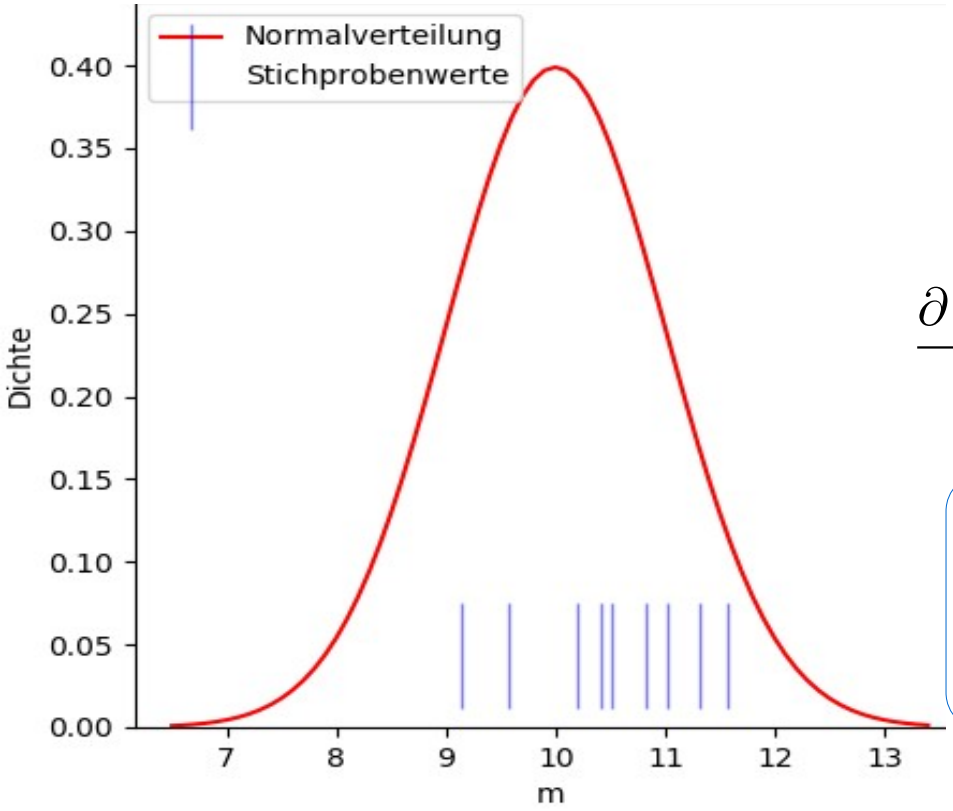
Erwartungswert und Varianz gaußverteilter Messungen

$$-\ln\mathcal{L}(\mu,\sigma\,|\vec{x}) = \frac{1}{2}\sum_i \left(\frac{x_i-\mu}{\sigma}\right)^2 \quad + \quad \sum_i \ln(\sqrt{2\pi}\sigma)$$

→

Bedingung für Extremum:

$$\frac{\partial\left(-\ln\mathcal{L}(\mu,\sigma\,|\vec{x})\right)}{\partial\mu} = \sum_i \frac{\mu - x_i}{\sigma^2} \stackrel{!}{=} 0$$



$$\Rightarrow \quad \hat{\mu} = \frac{1}{N}\sum_i x_i$$ *Formel für Mittelwert*

$$\frac{\partial\left(-\ln\mathcal{L}(\mu,\sigma\,|\vec{x})\right)}{\partial\sigma} = \sum_i \frac{(x_i-\mu)^2}{\sigma^3} + \sum_i \frac{1}{\sigma} \stackrel{!}{=} 0$$

$$\Rightarrow \quad \hat{\sigma}^2 = \frac{1}{N}\sum_i (x_i-\mu)^2$$ *Bekannte Formel für Standardabweichung*

(aber ohne Korrektur auf Verzerrung !)

weiteres Beispiel zur Likelihood-Schätzung

$$P(\vec{x}\,|\{a\}) = N(\vec{x}\,|\mu,\vec{\sigma})$$

Erwartungswert gaußverteilter Messungen mit **verschiedenen Unsicherheiten**:

hängt hier nicht von μ ab

$$\rightarrow \quad -\ln\mathcal{L}(\mu,\vec{\sigma}\,|\vec{x}) = \frac{1}{2}\sum_i \left(\frac{x_i-\mu}{\sigma_i}\right)^2 + \cancel{\sum_i \ln(\sqrt{2\pi}\sigma_i)}$$

Bedingung für Extremum:

$$\frac{\partial\left(-\ln\mathcal{L}(\mu,\sigma\,|\vec{x})\right)}{\partial\mu} = \sum_i \frac{\mu - x_i}{{\sigma_i}^2} \stackrel{!}{=} 0$$

$$\Rightarrow \quad \hat{\mu} = \left(\sum_i \frac{1}{{\sigma_i}^2}\right)^{-1} \cdot \sum_i \frac{x_i}{{\sigma_i}^2}$$

Mit $1/{\sigma_i}^2$ gewichtete Summe der Messwerte (auch altbekannt)

Achtung:

Wenn die Unsicherheiten von den Parametern abhängen,
(z.B. bei relativen, auf den wahren Wert bezogenen Unsicherheiten),
sollte der Term mit den σ_i nicht weggelassen werden!

Verzerrung der Schätzung der Varianz

$$P(\vec{x}\,|\{a\}) = N(\vec{x}\,|\mu,\sigma)$$

Berechnung:

$$b_{\hat{v^2}} = E[\hat{v^2}] - \sigma^2 = E\left[\frac{1}{n_d}\sum_i^{n_d} {x_i}^2 - \hat{\mu}^2\right] - \sigma^2 = \frac{1}{n_d}\sum_i^{n_d} E[{x_i}^2] - \hat{\mu}^2 - \sigma^2\,.$$

durch intelligentes Einsetzen einer Null, $\mu^2 - \mu^2$ erhält man

$$b_{\hat{v^2}} = \frac{1}{n_d}(\sum_i^{n_d} E[{x_i}^2] - \mu^2) - (\hat{\mu}^2 - \mu^2) - \sigma^2 = \sigma^2 - {\sigma_{\hat{\mu}}}^2 - \sigma^2 = {\sigma_{\hat{\mu}}}^2 = \frac{\sigma^2}{n_d} \neq 0\,.$$

Likelihood-Schätzung der Varianz ist verzerrt !

Verzerrung geht für $n_d \to \infty$ gegen 0, d. h.

Likelihood-Schätzung ist „asymptotisch unverzerrt“ (gilt allgemein)

Unsicherheit des Mittelwerts aus ML-Schätzung

Beispiel von eben: Unsicherheit auf den Erwartungswert gaußverteilter Messungen:

$$-\ln\mathcal{L}(\mu,\sigma\,|\vec{x}) = \frac{1}{2}\sum_i \left(\frac{x_i-\mu}{\sigma}\right)^2 \quad + \quad \sum_i \ln(\sqrt{2\pi}\sigma)$$

zweite Ableitung:

$$\left.\frac{\partial^2\left(-\ln\mathcal{L}(\mu,\sigma\,|\vec{x})\right)}{\partial\mu^2}\right|_{\mu=\hat{\mu}} = \sum_i \frac{1}{\sigma^2} = \frac{N}{\sigma^2}$$

$$\Rightarrow \quad V(\hat{\mu}) = \frac{\sigma^2}{N}$$

d.h. Unsicherheit des Mittelwerts: $\sigma/\sqrt{N}$ *(Sie erinnern sich hoffentlich !?)*

Unsicherheit erreicht die CRF-Grenze, die Schätzung ist also effizient !

Unsicherheit der Schätzung der Varianz einer Gaußverteilung

$$\hat{\sigma}^2 = \frac{1}{N}\sum_i (x_i - \mu)^2$$

Berechnung der Varianz aus CRF-Grenze : $V_{\hat{\sigma^2}} \geq \frac{\sigma^4}{n_d/2}$.

Man kann die Varianz auch direkt aus der Formel für den Mittelwert berechnen:

$$V_{\hat{\sigma^2}} = \frac{n_d - 1}{n_d}\frac{\sigma^4}{n_d/2}$$

Das ist kleiner als die Grenze !???

Wir haben den **Bias vergessen**, müssen CRF-Grenze mit Bias verwenden:

$$V_{\hat{\sigma^2}} \geq \left[-\left.\frac{\partial^2 \ln \mathcal{L}}{\partial p^2}\right|_{\hat{p}}\right]^{-1} \cdot \left[1 + \left.\frac{\partial b}{\partial p}\right|_{\hat{p}}\right]^2 = \left(\frac{n_d - 1}{n_d}\right)^2 \frac{\sigma^4}{n_d/2}$$

Korrekte Varianz ist größer als CRF-Grenze mit Bias √

→ ML-Schätzung der Varianz einer Gaußverteilung ist nicht optimal !

aber: Likelihood-Schätzung ist „asymptotisch effizient“ (gilt allgemein)

Korrelation von μ und σ

$$\left(V^{-1}\right)_{1,2} = -\left.\frac{\partial^2 \ln \mathcal{L}(\mathbf{a})}{\partial\mu\partial\sigma}\right|_{\hat{\mu},\hat{\sigma}} = -\frac{\partial^2}{\partial\mu\partial\sigma}\ \frac{1}{2}\sum_i \left(\frac{x_i-\mu}{\sigma}\right)^2 + \sum_i \ln(\sqrt{2\pi}\sigma)\Big|_{\hat{\mu},\hat{\sigma}} =$$

$$2\sum_i \frac{x_i-\hat{\mu}}{\hat{\sigma}^3} = \frac{2}{\hat{\sigma}^3}\sum_i x_i - \hat{\mu} = \frac{2}{\hat{\sigma}^3} N(\hat{\mu}-\hat{\mu}) = 0$$

d.h. die Schätzung von

Erwartungswert und Varianz einer Gaußverteilung sind unkorreliert !

Numerische Anpassung: PhyPraKit.phyFit und iminuit

package phyFit.py

Physics Fitting with *iminiut* [https://iminuit.readthedocs.ios/en/stable/]

Author: Guenter Quast, initial version Jan. 2021, updated Jun. 2021

Requirements:

- Python >= 3.6
- iminuit vers. > 2.0
- scipy > 1.5.0
- matplotlib > 3

The class *mnFit.py* uses the optimization and uncertainty-estimation package *iminuit* for fitting a parameter-dependent model f(x, *par) to data points (x, y) or a probability density function to binned histogram data or to unbinned data. Parameter estimation is based on pre-implemented Maximum-Likelihood methods, or on a user-defined cost function in the latter case, which provides maximum flexibility. Classical least-square methods are optionally available for comparison with other packages.

A unique feature of the package is the support of different kinds of uncertainties for x-y data, namely independent and/or correlated absolute and/or relative uncertainties in the x and/or y directions. Parameter estimation for density distributions is based on the shifted Poisson distribution, Poisson(x - loc, lambda), of the number of entries in each bin of a histogram.

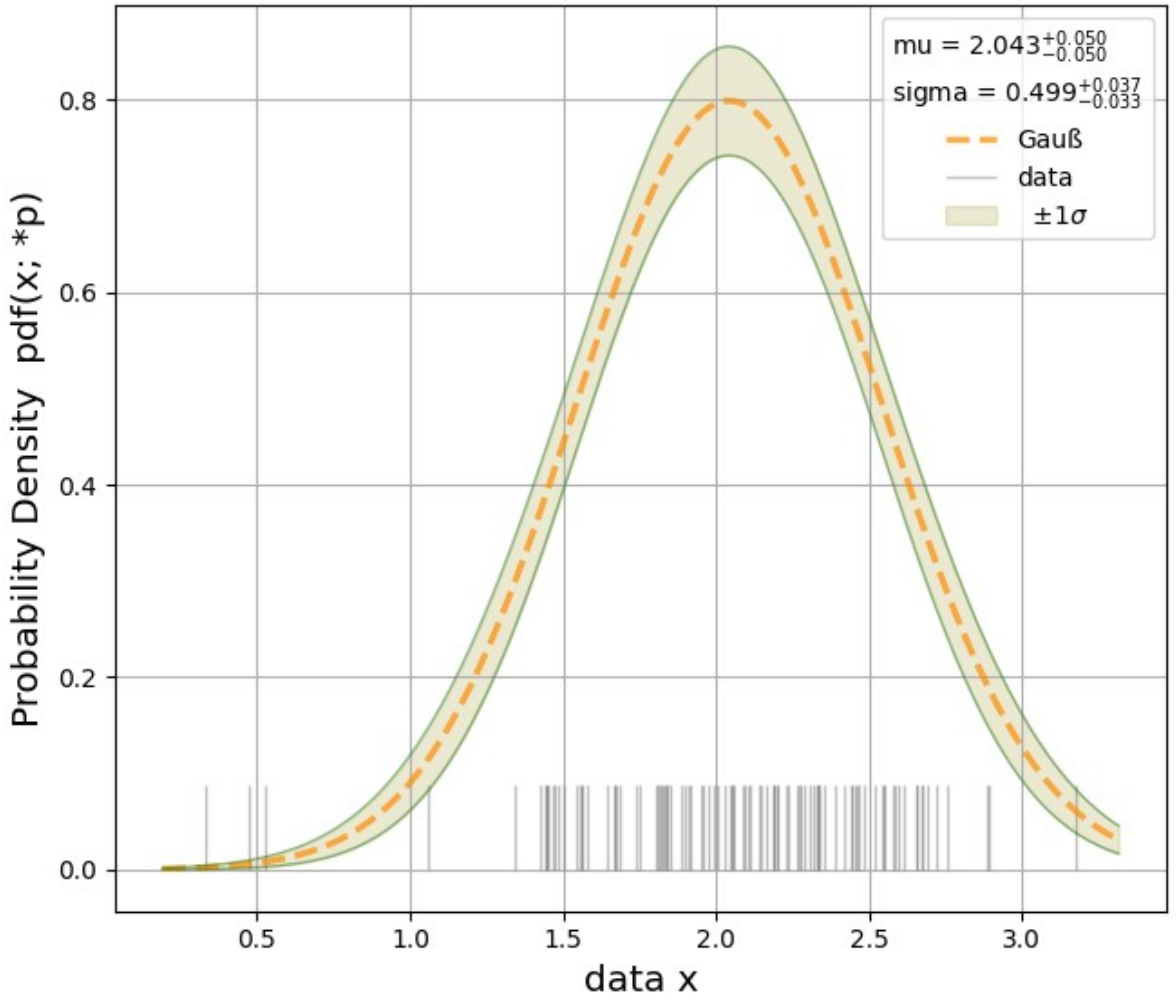
Parameter constraints, i.e. external knowledge of parameters within Gaussian uncertainties, limits on parameters in order to avoid problematic regions in parameter space during the minimization process, and fixing of parameters, e.g. to include the validity range of a model in the parameters without affecting the fit, are also supported by *mnFit*.

ML Gauß-Anpassung mit PhyPraKit.mFit()

Üblicherweise führt man Anpassungen mit Hilfe numerischer Verfahren durch:
Beispiel Skript `ML-Gauss.py` zur ML-Anpassung
einer Gauß-Verteilung an 100 Datenpunkte.

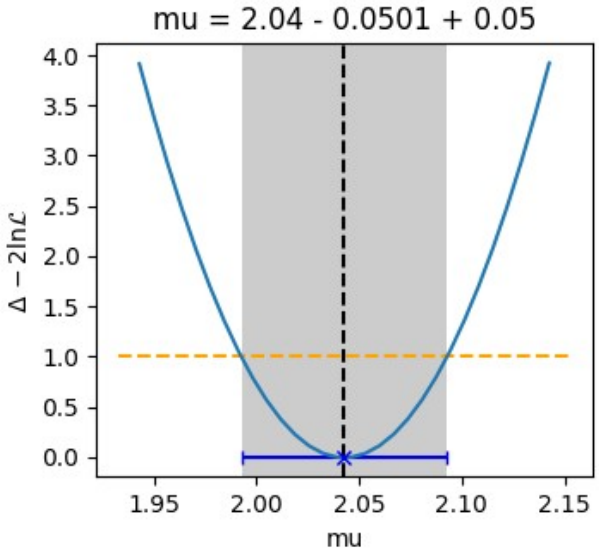
Ausgabe des Programms:

```
*==* unbinned ML Fit Result:
parameter values
 [2.04275072 0.49922757]
confidence intervals
 [[-0.05009282  0.04999974]
 [-0.03332887  0.03748651]]
correlation matrix
 [[1.00000000e+00 5.48652848e-04]
 [5.48652848e-04 1.00000000e+00]]
goodness-of-fit
 None
parameter names
 ('mu', 'sigma')
```



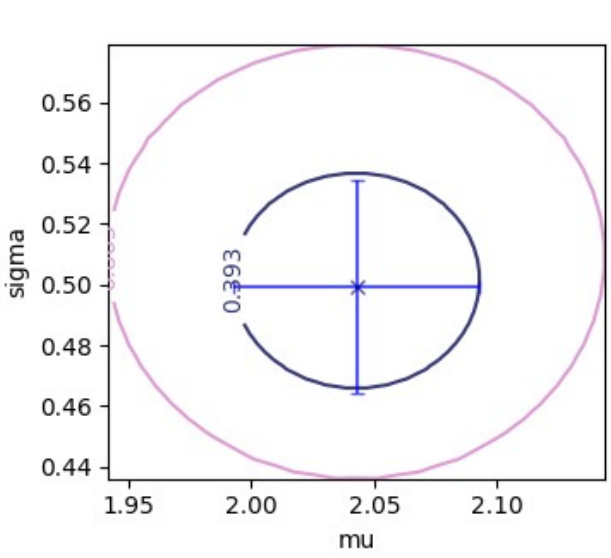
ML Gauß-Anpassung: Likelihood und Konturen

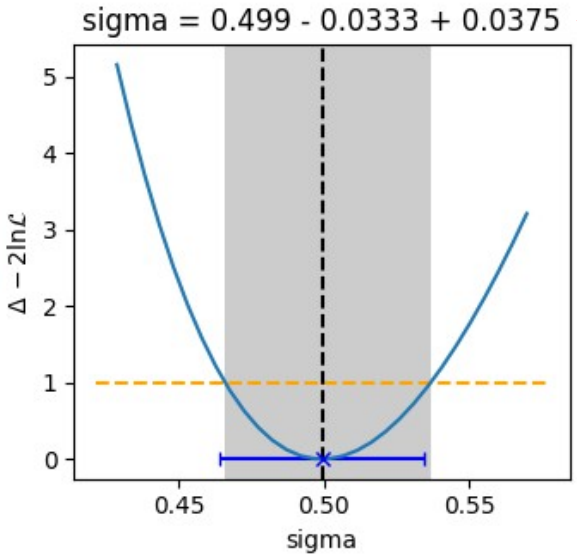
Likelihood-Kurven der Parameter und Kovarianzellipse:



Achtung:

Skript verwendet -2ln$\mathcal{L}$





Asymmetrische Unsicherheit auf Schätzwert von σ !

ML-Gauss.py

```
# set Model to use
modelPDF=fGauss

# genereta data (here: drwan from Gaussian distribution)
mu0=2.
sig0=0.5
np.random.seed(314159)  # initialize random generator
Data = mu0 + sig0 * np.random.randn(100)

rdict = mFit( modelPDF,
        data = Data,           # data
        p0=None,               # initial guess for parameter values
        constraints=None,  # Gaussian parameter constraints
        neg2logL = True,     # use  -2 * ln(L)
        plot=True,             # plot data and model
        plot_band=True,     # plot model confidence-band
        plot_cor=True,       # plot profiles likelihood and contours
        showplots=False,   # show / don't show plots
        quiet=False,           # suppress informative printout if True
        axis_labels=['data x', 'Probability Density  pdf(x; *p)'],
        data_legend = 'data',
        model_legend = 'Gauß' )

# Print results
print('\n*==* unbinned ML Fit Result:')
for key in rdict:
    print("{}\n".format(key), rdict[key])

plt.show()
```

Beispiel Likelihood-Schätzung der Lebensdauer

Lebensdauer aus Exponentialverteilung:

$$P(t\,|\tau) = \frac{1}{\tau}\exp\left(-t/\tau\right)$$

→ $$-\ln\mathcal{L}(\tau\,|\vec{t}) = \sum_i \ln(\tau) + \sum_i t_i/\tau$$

Bedingung für Extremum:

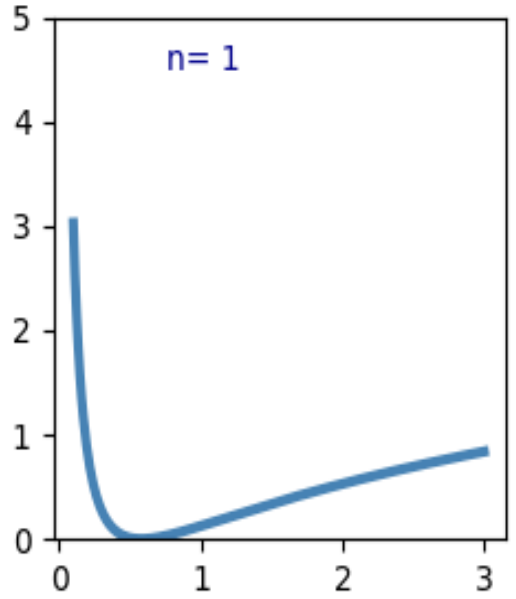
$$\frac{\partial\left(-\ln\mathcal{L}(\tau\,|\vec{t})\right)}{\partial\tau} = \sum_i \frac{1}{\tau} - \sum_i \frac{t_i}{\tau^2} \stackrel{!}{=} 0$$

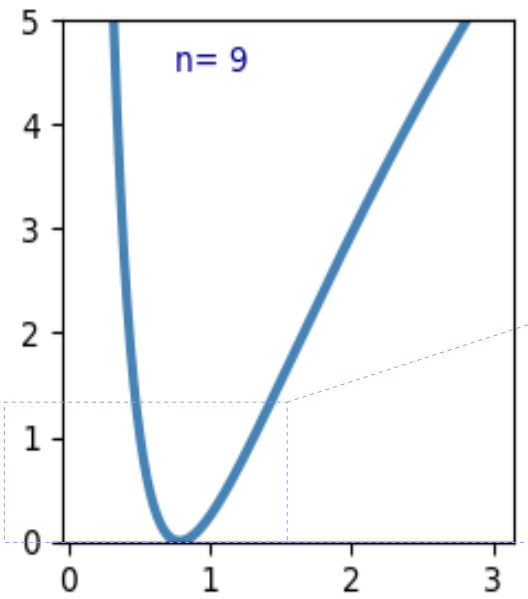
$$\Rightarrow \quad \hat{\tau} = \frac{1}{N}\sum_i t_i$$

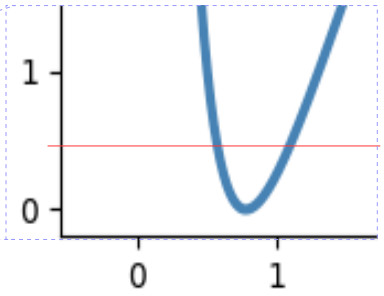
Mittelwert der gemessenen Lebensdauern

(deshalb heißt τ ja auch „mittlere Lebensdauer“)

neg. logL-Kurven sind meist keine Parabeln !







Hineinzoomen in interessanten Bereich:
→ näherungsweise parabolisch

s. Skript
`nlLExp.py`

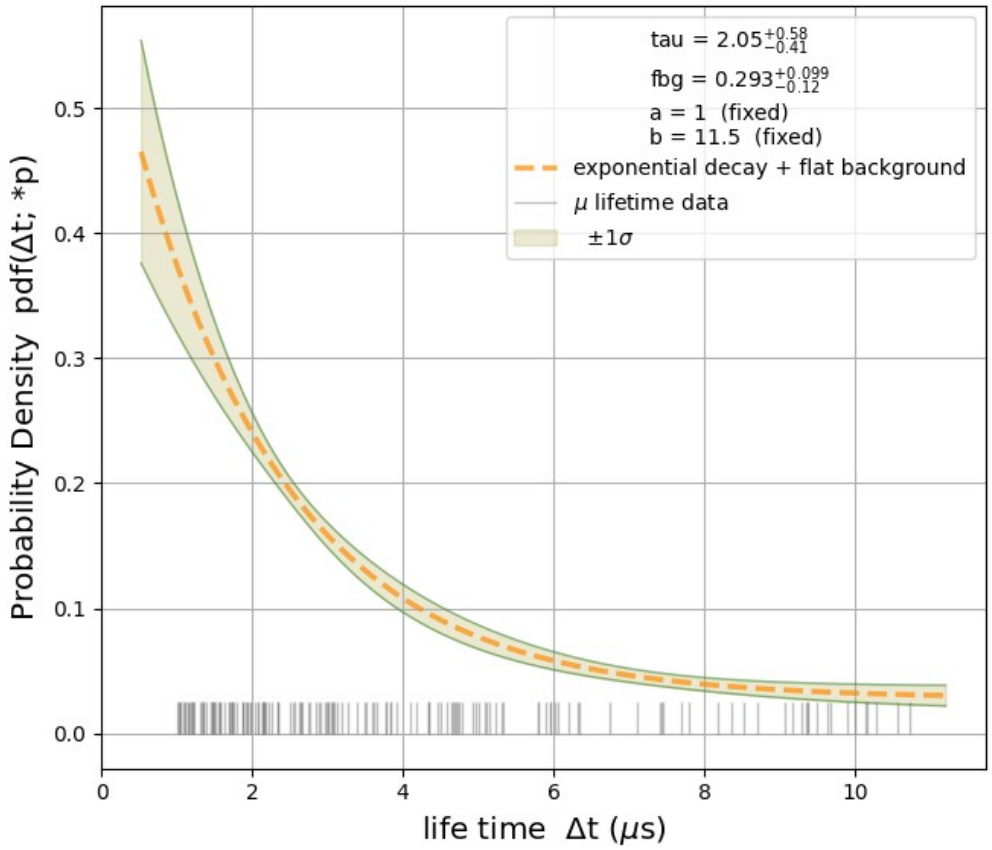
Lebensdauer mit phyFit

!!! PDF muss normiert sein!

```
def modelPDF(t, tau=2., fbg=0.2, a=1., b=11.5):
  """Probability density function for an exponential decay with flat background.
  The pdf is normed for the  interval [a=1µs,  b=11.5µs);
  these parameters a and b must be fixed in the fit!
  """
  pdf1 = np.exp(-t / tau) / tau / (np.exp(-a / tau) - np.exp(-b / tau))
  pdf2 = 1. / (b - a)
  return (1 - fbg) * pdf1 + fbg * pdf2
```

Beispiel Skript

ML-Exponential.py



Verwendung von phyFit

Die hier gezeigten Beispiele basieren mit Absicht auf dem schlanken Anpassungswerkzeug *phyFit*, um die Transparenz der Vorgehensweise sicher zu stellen und Ihnen durch Modifikation der Beispiele eigene Studien und Anpassungen an spezielle Problemstellungen zu er möglichen. Da *phyFit* nur eine schlanke Interface-Ebene darstellt und im Übrigen auf dem im wissenschaftlichen Umfeld seit langem etablierten und anerkannten Paket MINUIT zur Optimierung in hochdimensionalen Parameterräumen und zur Schätzung der Parameterunsicherheiten basiert, sind Ihre damit erzielten Ergebnisse fachlich sehr gut abgesichert und in eigenen Arbeiten "zitierfähig".

Auszug aus Jupyter-Tutorial „negLogLFits.ipynb“