Statistical Methods in Particle Physics

5. Parameter Estimation

Prof. Dr. Klaus Reygers (lectures)
Dr. Sebastian Neubert (tutorials)

Heidelberg University
WS 2017/18

Basics

Estimator

Suppose we have a measurement of n independent values

$$\vec{x} = (x_1, x_2, ..., x_n)$$

which follow the same underlying distribution f(x; θ),
e.g., f(x; θ) = 1/θ exp(–x/θ).

i.i.d. random variables = independent, identically distributed

An estimator is a function of the data which provides a numerical estimate of the parameter θ:

$$\hat{\theta}(\vec{x})$$

θ often is not only one parameter but a vector of parameters.

Properties of Estimators

Consistency

An estimator is consistent if it converges to the true value

$$\lim_{n\to\infty} \hat{\vec{\theta}} = \vec{\theta}$$

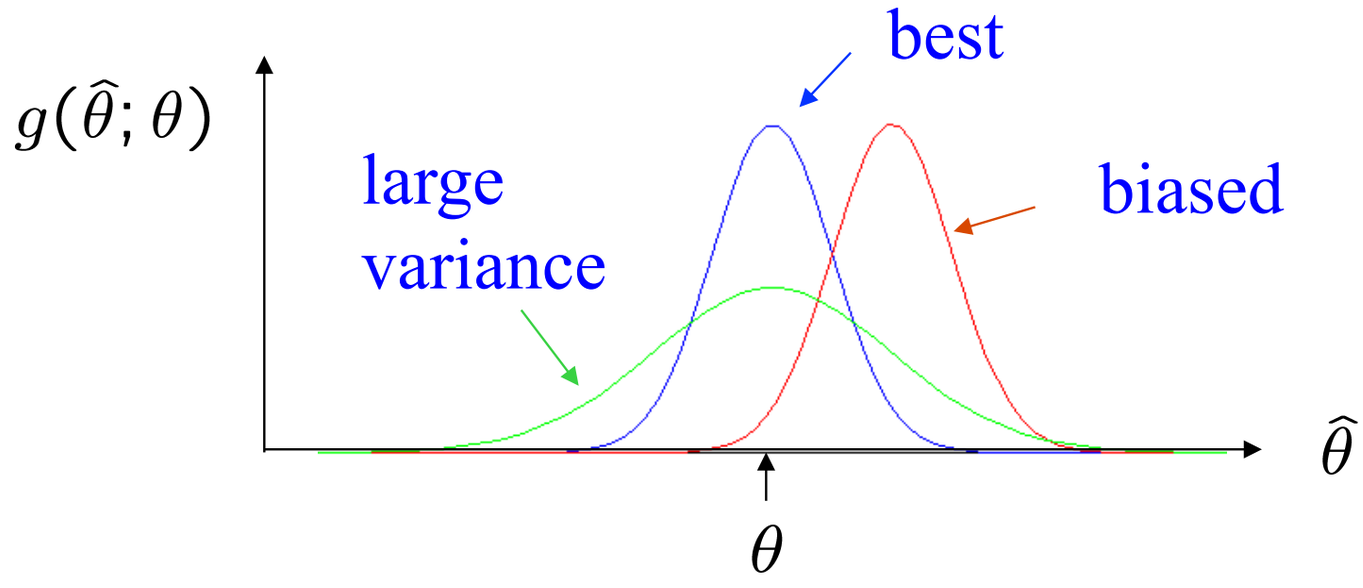
Bias

Difference btw. expectation value of estimator and true value

$$\vec{b} := E[\hat{\vec{\theta}}] - \vec{\theta}$$

Efficiency

An estimator is efficient if its variance V(θ) is small



http://www.pp.rhul.ac.uk/~cowan/stat_course.html

Example: Estimators for the lifetime of a particle

Estimator	Consistent?	Unbiased?	Efficient?
$\hat{\tau} = \frac{t_1+t_2+...+t_n}{n}$	yes	yes	yes
$\hat{\tau} = \frac{t_1+t_2+...+t_n}{n-1}$	yes	no	no
$\hat{\tau} = t_1$	no	yes	no

http://www.terascale.de/e149980/index_eng.html

Unbiased Estimator for Mean and Variance

Estimator for the mean: $$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

bias

$$b = E[\hat{\mu}] - \mu = 0, \qquad V[\hat{\mu}] = \frac{\sigma^2}{n}, \qquad \text{i.e.}, \sigma_{\hat{\mu}} = \frac{\sigma}{\sqrt{n}}$$

Estimator for the variance: $$s^2 := \hat{\sigma^2} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

$$b = E[s^2] - \sigma^2 = 0$$

$$\mathrm{V}[s^2] = \frac{\sigma^4}{n}\left((\kappa - 1) + \frac{2}{n-1}\right) = \frac{1}{n}\left(\mu_4 - \frac{n-3}{n-1}\sigma^4\right) \qquad \text{[without proof]}$$

κ: kurtosis

μ_4: fourth central moment

http://en.wikipedia.org/wiki/Variance#Distribution_of_the_sample_variance

Unbiased Estimator of the Variance: Derivation (I)

Consider n independent and identically distributed random variable x_i:

$$\mu := E[x_i], \quad \sigma^2 := V[x_i], \quad \bar{x} := \frac{1}{n}\sum_{i=1}^{n} x_i$$

We'll use:

$$\sigma^2 = E[x_i^2] - \mu^2 \quad \rightsquigarrow \quad E[x_i^2] = \mu^2 + \sigma^2$$

$$V[\bar{x}] = \frac{1}{n^2} V[\sum_{i=1}^{n} x_i] = \frac{1}{n} V[x_i] = \frac{\sigma^2}{n} \stackrel{!}{=} E[\bar{x}^2] - \mu^2 \quad \rightsquigarrow \quad E[\bar{x}^2] = \frac{\sigma^2}{n} + \mu^2$$

Now we calculate the expectation value of $\sum_{i=1}^{n}(x_i - \bar{x})^2$:

$$\sum_{i=1}^{n}(x_i - \bar{x})^2 = \sum_{i=1}^{n} x_i^2 - 2x_i\bar{x} + \bar{x}^2 = \left(\sum_{i=1}^{n} x_i^2\right) - n\bar{x}^2$$

$$E[\sum_{i=1}^{n}(x_i - \bar{x})^2] = E[\sum_{i=1}^{n} x_i^2] - E[n\bar{x}^2] = n(\mu^2 + \sigma^2) - \sigma^2 - n\mu^2 = (n-1)\sigma^2$$

Unbiased Estimator of the Variance: Derivation (II)

This means that

$$s^2 := \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

is an unbiased estimator of the variance, i.e., $E[s^2] = \sigma^2$.

Multiplying the sample variance by $n/(n-1)$ is known as Bessel's correction.

Note that s is not an unbiased estimator of the standard deviation:

https://en.wikipedia.org/wiki/Unbiased_estimation_of_standard_deviation

Maximum Likelihood Method

Likelihood Function

Suppose we have a measurement of *n* independent values

$$\vec{x} = (x_1, x_2, ..., x_n)$$

drawn from the distribution

$$f(x; \vec{\theta}), \quad \vec{\theta} = (\theta_1, \theta_2, ..., \theta_m)$$

The joint pdf for the observed values $\vec{x}$ is given by:

$$L(\vec{x}; \vec{\theta}) = \prod_{i=1}^{n} f(x_i; \vec{\theta})$$ "likelihood function"

We consider $\vec{x}$ as constant. The *maximum likelihood estimate* (MLE) of the parameters are the values $\hat{\vec{\theta}}$ for which $L(\vec{x}; \vec{\theta})$ has a global maximum.

In other words, we ask the question:

"For which parameters do the observed data have the highest probability?"

Maximum Likelihood Example 1: Exponential Decay

Consider exponential pdf: $$f(t;\tau) = \frac{1}{\tau} e^{-t/\tau}$$

Independent measurements drawn from this distribution: $t_1, t_2, \ldots, t_n$

Likelihood function: $$L(\tau) = \prod_{i=1}^{n} \frac{1}{\tau} e^{-t_i/\tau}$$

$L(\tau)$ is maximum when $\ln L(\tau)$ is maximum:

$$\ln L(\tau) = \sum_{i=1}^{n} \ln f(t_i;\tau) = \sum_{i=1}^{n} \left(\ln \frac{1}{\tau} - \frac{t_i}{\tau} \right)$$

Find maximum:

$$\frac{\partial \ln L(\tau)}{\partial \tau} = 0 \quad \rightsquigarrow \quad \sum_{i=1}^{n} \left(-\frac{1}{\tau} + \frac{t_i}{\tau^2} \right) = 0 \quad \rightsquigarrow \quad \hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} t_i$$

Maximum Likelihood Example 2: Gaussian (I)

Consider $x_1, x_2, \ldots, x_n$ drawn from Gaussian(μ, σ^2)

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Log-likelihood function:

$$\ln L(\mu,\sigma^2) = \sum_{i=1}^{n} \ln f(x_i;\mu,\sigma^2) = \sum_{i=1}^{n} \left(\ln \frac{1}{\sqrt{2\pi}} - \ln \sigma - \frac{(x_i-\mu)^2}{2\sigma^2} \right)$$

Derivatives w.r.t. μ and σ^2:

$$\frac{\partial \ln L(\mu,\sigma^2)}{\partial \mu} = \sum_{i=1}^{n} \frac{x_i - \mu}{\sigma^2}$$

$$\frac{\partial \ln L(\mu,\sigma^2)}{\partial \sigma^2} = \sum_{i=1}^{n} \left(\frac{(x_i-\mu)^2}{2\sigma^4} - \frac{1}{2\sigma^2} \right)$$

Maximum Likelihood Example 2: Gaussian (II)

Setting the derivatives w.r.t. μ and σ^2 to zero and solving the equations:

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \widehat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2$$

We find that the ML estimator for σ^2 is biased!

Properties of the ML Estimator

- The ML estimator is consistent, i.e., it approaches the true value in the limit of infinite measurements ($n \to \infty$)
- For finite n the ML estimator is in general biased
- The ML Estimator is invariant under parameter transformation

$$\psi = g(\theta) \quad \Rightarrow \quad \hat{\psi} = g(\hat{\theta})$$

Averaging Measurements with Gaussian Uncertainties (I)

pdf for measurement *i* (same mean, different σ):

$$f(x;\mu,\sigma_i^2) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu)^2}{2\sigma_i^2}}$$

Similar as before:

$$\ln L(\mu) = \sum_{i=1}^{n} \left(\ln \frac{1}{\sqrt{2\pi}} - \ln \sigma_i - \frac{(x_i-\mu)^2}{2\sigma_i^2} \right)$$

We obtain the formula for the weighted average that we already know from chapter 3:

$$\left.\frac{\partial \ln L(\mu)}{\partial \mu}\right|_{\mu=\hat{\mu}} = \sum_{i=1}^{n} \frac{x_i - \hat{\mu}}{\sigma_i^2} \stackrel{!}{=} 0 \qquad \Rightarrow \qquad \hat{\mu} = \frac{\sum_{i=1}^{n} \frac{x_i}{\sigma_i^2}}{\sum_{i=1}^{n} \frac{1}{\sigma_i^2}}$$

Uncertainty? Let's Tayler-expand, exact because ln L(μ) has a parabolic form:

$$\ln L(\mu) = \ln L(\hat{\mu}) + \underbrace{(\mu-\hat{\mu}) \left.\frac{\partial \ln L(\mu)}{\partial \mu}\right|_{\mu=\hat{\mu}}}_{=0} - \frac{h}{2}(\mu-\hat{\mu})^2, \qquad h = -\left.\frac{\partial^2 \ln L(\mu)}{\partial^2 \mu}\right|_{\mu=\hat{\mu}}$$

Averaging Measurements with Gaussian Uncertainties (II)

This means that the likelihood function is Gaussian:

$$L(\mu) \propto e^{-\frac{h}{2}(\mu-\hat{\mu})^2}$$

For the standard deviation we obtain:

$$\sigma_{\hat{\mu}} = 1/\sqrt{h} = \left(- \left. \frac{\partial^2 \ln L(\mu)}{\partial^2 \mu} \right|_{\mu=\hat{\mu}} \right)^{-1/2}$$

$$h = \sum_{i=1}^{n} \frac{1}{\sigma_i^2} \quad \Rightarrow \quad \sigma_{\hat{\mu}} = \left(\sum_{i=1}^{n} \frac{1}{\sigma_i^2} \right)^{-1/2}$$

Alternatively, one can obtain the uncertainty of the weighted average from the points where ln L drops by 1/2:

$$\ln L(\hat{\mu} \pm \sigma_{\hat{\mu}}) = \ln L(\hat{\mu}) - \frac{1}{2}$$

Likelihood Function and Minimum Variance Bound

Let's first consider likelihood function with only one parameter:

$$L(\vec{x};\theta) = L(x_1, x_2, ..., x_n;\theta) = \prod_{i=1}^{n} f(x_i;\theta)$$

Let $\hat{\theta}(\vec{x})$ be an unbiased estimator of the parameter θ

It can be shown that the variance (of any unbiased estimator) satisfies:

$$V[\hat{\theta}] \geq \frac{1}{E\left[-\frac{\partial^2 \ln L}{\partial^2 \theta}\right]}$$

For a biased estimator this becomes

$$V[\hat{\theta}] \geq \frac{\left(1+\frac{\partial b}{\partial \theta}\right)^2}{E\left[-\frac{\partial^2 \ln L}{\partial^2 \theta}\right]}$$

This bound is called Rao-Cramér-Frechet minimum variance bound (MVB)

MVB Example: Exponential Decay

Reminder:

$$\frac{\partial \ln L(\tau)}{\partial \tau} = 0 \quad \rightsquigarrow \quad \sum_{i=1}^{n} \left(-\frac{1}{\tau} + \frac{t_i}{\tau^2} \right) = 0 \quad \rightsquigarrow \quad \hat{\tau} = \frac{1}{n} \sum_{i=1}^{n} t_i$$

Variance of the estimated decay time:

$$\frac{\partial^2 \ln L(\tau)}{\partial^2 \tau} = \sum_{i=1}^{n} \left(\frac{1}{\tau^2} - 2\frac{t_i}{\tau^3} \right) = \frac{n}{\tau^2} - \frac{2}{\tau^3} \sum_{i=1}^{n} t_i = \frac{n}{\tau^2} \left(1 - \frac{2\hat{\tau}}{\tau} \right)$$

Minimum variance bound (MVB):

$$V[\hat{\tau}] \geq \frac{1}{E\left[-\frac{n}{\tau^2} \left(1 - \frac{2\hat{\tau}}{\tau} \right) \right]} = \frac{1}{-\frac{n}{\tau^2} \left(1 - \frac{2E[\hat{\tau}]}{\tau} \right)} = \frac{\tau^2}{n}$$

Uncertainty of the ML Estimator: Approach I (Minimum Variance Bound)

For any probability function $f(x;\theta)$ the likelihood function L approaches a Gaussian for large n, i.e., for a large number of events, and the variance of the ML estimator reaches the minimum variance bound.

In many cases it is impractical to calculate the MVB analytically. Instead, one uses the following approximation which is good for large n:

$$E\left[-\frac{\partial^2 \ln L}{\partial^2 \theta}\right] \approx -\frac{\partial^2 \ln L}{\partial^2 \theta}\bigg|_{\theta=\hat{\theta}}$$

The variance of the ML estimator is given by:

$$V[\hat{\theta}] = -\frac{1}{\frac{\partial^2 \ln L}{\partial^2 \theta}\Big|_{\theta=\hat{\theta}}}$$

Uncertainty of the ML Estimator: Approach II ("Graphical Method")

Taylor expansion of ln L around the maximum:

$$\ln L(\theta) = \ln L(\hat{\theta}) + \underbrace{\left[\frac{\partial \ln L}{\partial \theta}\right]_{\theta=\hat{\theta}} (\theta - \hat{\theta})}_{=0} + \frac{1}{2!}\left[\frac{\partial^2 \ln L}{\partial^2 \theta}\right]_{\theta=\hat{\theta}} (\theta - \hat{\theta})^2 + \ldots$$

$-\frac{1}{\sigma^2}$ for a Gaussian

If $L(\theta)$ is approximately Gaussian (ln $L(\theta)$ then is a approximately a parabola):

$$\ln L(\theta) \approx \ln L_{\max} - \frac{(\theta - \hat{\theta})^2}{2\widehat{\sigma^2_{\hat{\theta}}}}$$

good approximation in the large sample limit

One can then estimate the uncertainties from the points where ln L has dropped by 1/2 from its maximum:

$$\ln L(\hat{\theta} \pm \hat{\sigma}_{\hat{\theta}}) \approx \ln L_{\max} - \frac{1}{2}$$

- Can be used even if $L(\theta)$ is not Gaussian
- $L(\theta)$ Gaussian → results of approach I and II identical

Example:
Uncertainty of the Decay Time for an Exponential Decay

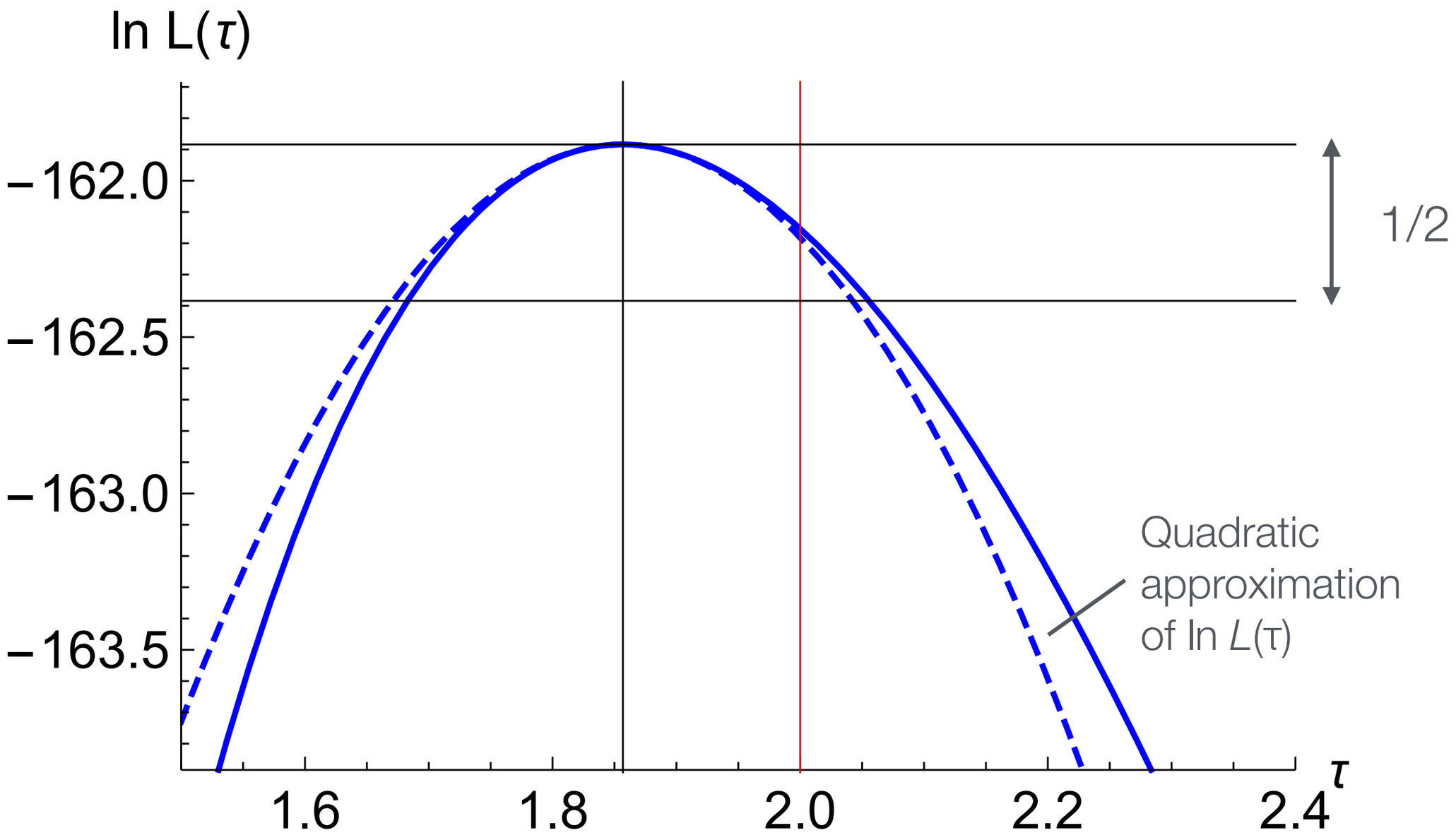
Variance of the estimated decay time:

$$\frac{\partial^2 \ln L(\tau)}{\partial^2 \tau} = \sum_{i=1}^{n} \left(\frac{1}{\tau^2} - 2\frac{t_i}{\tau^3} \right) = \frac{n}{\tau^2} - \frac{2}{\tau^3} \sum_{i=1}^{n} t_i = \frac{n}{\tau^2} \left(1 - \frac{2\hat{\tau}}{\tau} \right)$$

$$V[\hat{\tau}] = - \left(\frac{\partial^2 \ln L}{\partial^2 \theta} \right)^{-1}_{\tau=\hat{\tau}} = \frac{\hat{\tau}^2}{n} \qquad \rightsquigarrow \qquad \hat{\sigma} = \frac{\hat{\tau}}{\sqrt{n}}$$

Exponential Decay: Illustration

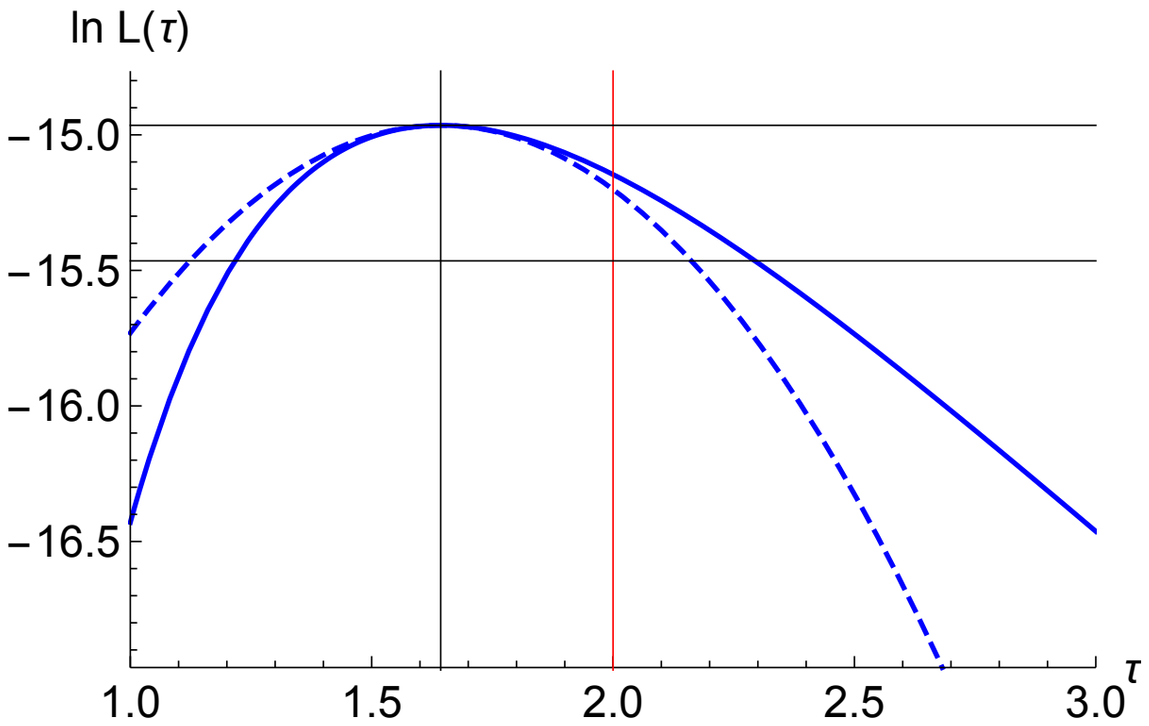
100 data points sampled from $f(t;\tau) = \frac{1}{\tau} e^{-t/\tau}$ with $\tau = 2$



ML estimate: $\hat{\tau} = 1.86 \pm 0.18$

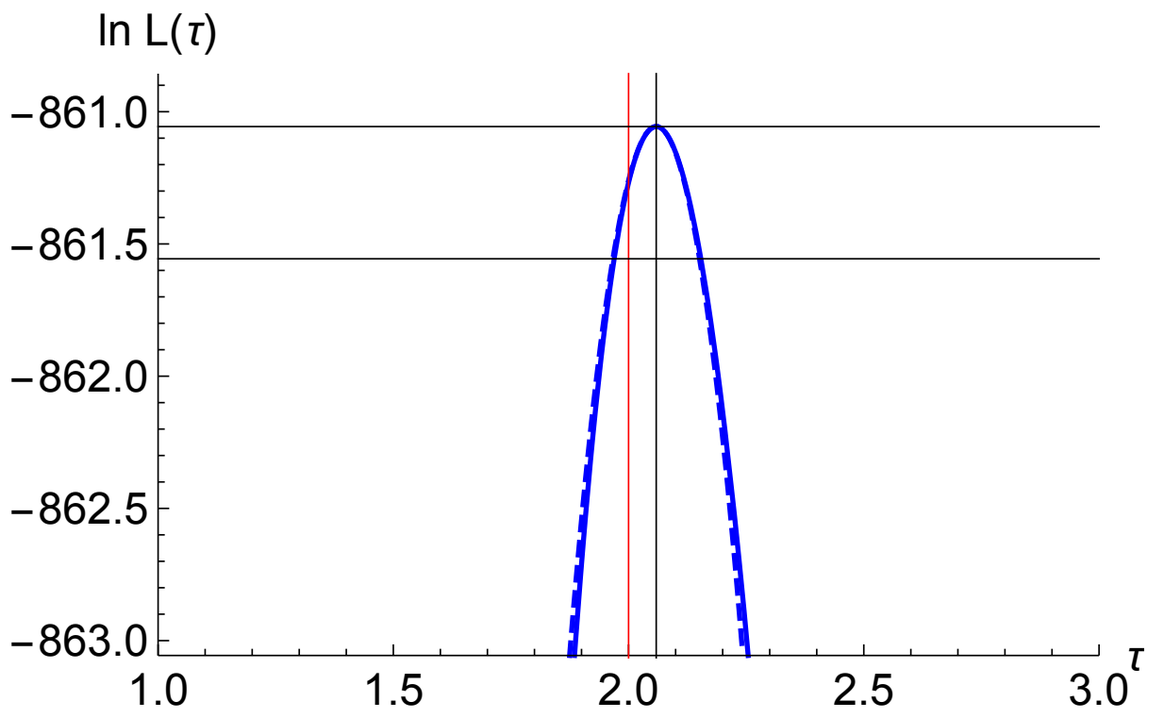
Exponential Decay: Log-Likelihood Function for Different Sample Sizes

10 data points



quadratic approximation of ln $L(\tau)$ is not very good

500 data points



quadratic approximation of ln $L(\tau)$ is excellent

Minimum Variance Bound for *m* Parameters

$$f(x;\vec{\theta}), \quad \vec{\theta} = (\theta_1, \theta_2, ..., \theta_m)$$

Minimum variance bound related to Fisher information matrix ($m \times m$ matrix):

$$V[\hat{\theta}_j] \geq (I(\vec{\theta})^{-1})_{jj}$$

$$I_{jk}[\vec{\theta}] = -E\left[\sum_{i=1}^{n} \frac{\partial^2 \ln f(x_i;\vec{\theta})}{\partial\theta_j \theta_k}\right] = -E\left[\frac{\partial^2 \ln L(\vec{\theta})}{\partial\theta_j \partial\theta_k}\right]$$

n = number of data events

Components I_{jk} of the Fisher information matrix can also be expressed as

$$I_{jk}[\vec{\theta}] = -n \int \frac{\partial^2 \ln f(x;\vec{\theta})}{\partial\theta_j \partial\theta_k} f(x;\vec{\theta})\, \mathrm{d}x = n \int \frac{\partial \ln f(x;\vec{\theta})}{\partial\theta_j} \frac{\partial \ln f(x;\vec{\theta})}{\partial\theta_k} f(x;\vec{\theta})\, \mathrm{d}x$$

$$= n \int \frac{1}{f(x;\vec{\theta})} \frac{\partial f(x;\vec{\theta})}{\partial\theta_j} \frac{\partial f(x;\vec{\theta})}{\partial\theta_k}\, \mathrm{d}x$$

Variance of the ML Estimator for *m* Parameters

For any probability function $f(x;\vec{\theta})$ the likelihood function L approaches a multi-variate Gaussian for large n

$$L(\vec{\theta}) \propto e^{-\frac{1}{2}(\vec{\theta}-\widehat{\vec{\theta}})^{\mathsf{T}}\, V^{-1}[\widehat{\vec{\theta}}]\,(\vec{\theta}-\widehat{\vec{\theta}})}$$

The variance of the ML estimator then reaches the MVB:

$$V[\widehat{\vec{\theta}}] \to I(\vec{\theta})^{-1}$$

Covariance matrix of the estimated parameters:

$$V[\widehat{\vec{\theta}}] \approx \left[-\left. \frac{\partial^2 \ln L(\vec{x};\vec{\theta})}{\partial^2 \vec{\theta}} \right|_{\vec{\theta}=\widehat{\vec{\theta}}} \right]^{-1}$$

or equivalently:

$$(V^{-1}[\widehat{\vec{\theta}}])_{ij} = -\left. \frac{\partial^2 \ln L(\vec{x};\vec{\theta})}{\partial\theta_i \partial\theta_j} \right|_{\vec{\theta}=\widehat{\vec{\theta}}}$$

Standard deviation of a single parameters:

$$\hat{\sigma}_{\hat{\theta}_j} = \sqrt{(V[\widehat{\vec{\theta}}])_{jj}}$$

$z \cdot \sigma$ contour (hyper surface) defined by:

$$\ln L(\vec{\theta}) = \ln L_{\max} - z^2/2$$

Example: 2 Parameter ML Fit (from G. Cowan's Book)

Scattering angle distribution, $x = \cos\theta$: $\quad f(x; a, b) = \dfrac{1 + ax + bx^2}{2 + 2b/3}$

Normalization: $\quad \displaystyle\int_{x_{\min}}^{x_{\max}} f(x; a, b)\,\mathrm{d}x = 1$

Example: $\quad a = 0.5,\ b = 0.5;\ x_{\min} = -0.95,\ x_{\max} = 0.95,\ 1000\,\mathrm{MC\ events}$

Numerical minimization with MINUIT:

$$\hat{a} = 0.53 \pm 0.07$$

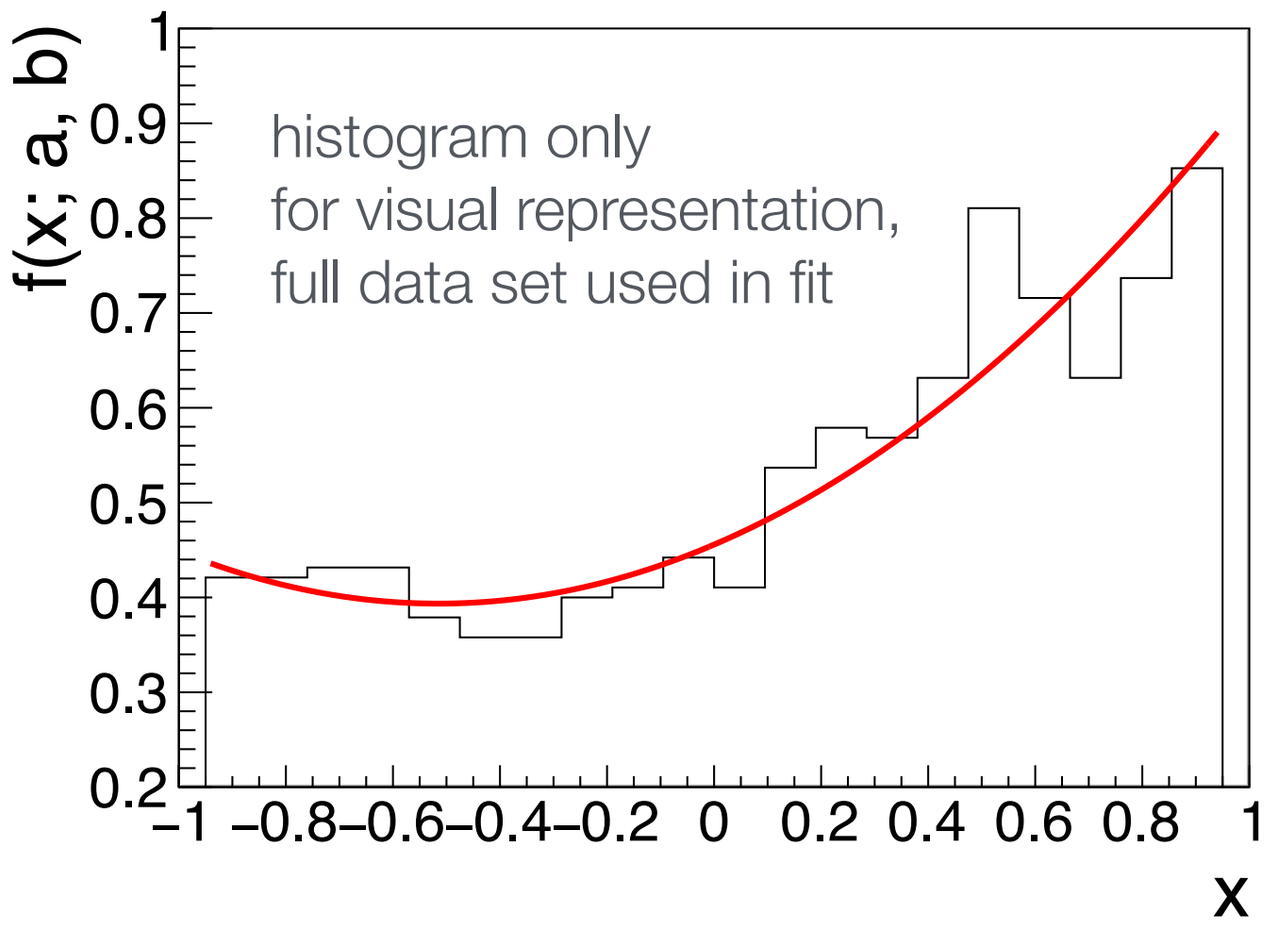
$$\hat{b} = 0.51 \pm 0.16$$

$$\mathrm{cov}[\hat{a}, \hat{b}] = 0.006$$

$$\rho = 0.476$$

Uncertainties and covariance from inverse of Hessian matrix:

$$(\widehat{V}^{-1})_{ij} = -\left.\frac{\partial^2 \ln L}{\partial\theta_i \partial\theta_j}\right|_{\vec{\theta}=\hat{\vec{\theta}}}$$



Example: 2 Parameter ML Fit (root Code Snippets)

```
const Int_t n_sample = 1000;
Double_t data[n_sample];

const Double_t xmin = -0.95;
const Double_t xmax = 0.95;

// probability density function for x = cos(theta) (theta = scattering angle),
// normalized to unity
Double_t f(Double_t *x, Double_t *par) {

    Double_t a = par[0];
    Double_t b = par[1];

    return (6 * (1 + a * x[0] + b * x[0] * x[0])) /
           ((xmax - xmin) * (3 * a * (xmax + xmin) +
                             2 * (3 + b * (xmax * xmax + xmax * xmin + xmin * xmin))));
}

// negative log-likelihood function
void negative_log_likelihood(Int_t &npar, Double_t *gin, Double_t &nll, Double_t *par,
Int_t iflag) {

    Double_t sum = 0;

    for (Int_t i = 0; i < n_sample; i++) {
        Double_t fi = f(&data[i], par);

        sum += TMath::Log(fi);
    }
    nll = -sum;
}
```

data defined globally (ugly, but that's how it works in MINUIT)

parameter list as required by MINUIT

Example: 2 Parameter ML Fit (root Code Snippets)

```
// prepare minuit
Int_t nPar = 2; // number of fit parameters
TMinuit m(nPar);
m.SetFCN(negative_log_likelihood);
m.SetPrintLevel(0); // -1 quiet, 0 normal, 1 verbose

// 1 for chi2 fit, 0.5 for negative log-likelihood fir
// see section 1.4.1 in MINUIT manual, e.g., http://hep.fi.infn.it/minuit.pdf
m.SetErrorDef(0.5);

// parameters:
// parameter no., name, start value, step size, range min., range max.
// range min = range max = 0 -> no limits
m.DefineParameter(0, "a", 0.45, 0.01, 0, 0);
m.DefineParameter(1, "b", 0.45, 0.01, 0, 0);

// now ready for minimization step
m.Migrad();
m.Command("SHOW COV"); // show covariance matrix

// draw fit
Double_t a, a_err, b, b_err;
m.GetParameter(0, a, a_err);
m.GetParameter(1, b, b_err);
tf->SetParameters(a, b);
tf->SetLineColor(kRed);
tf->Draw("same");
```

Example: 2 Parameter ML Fit (MINUIT Output)

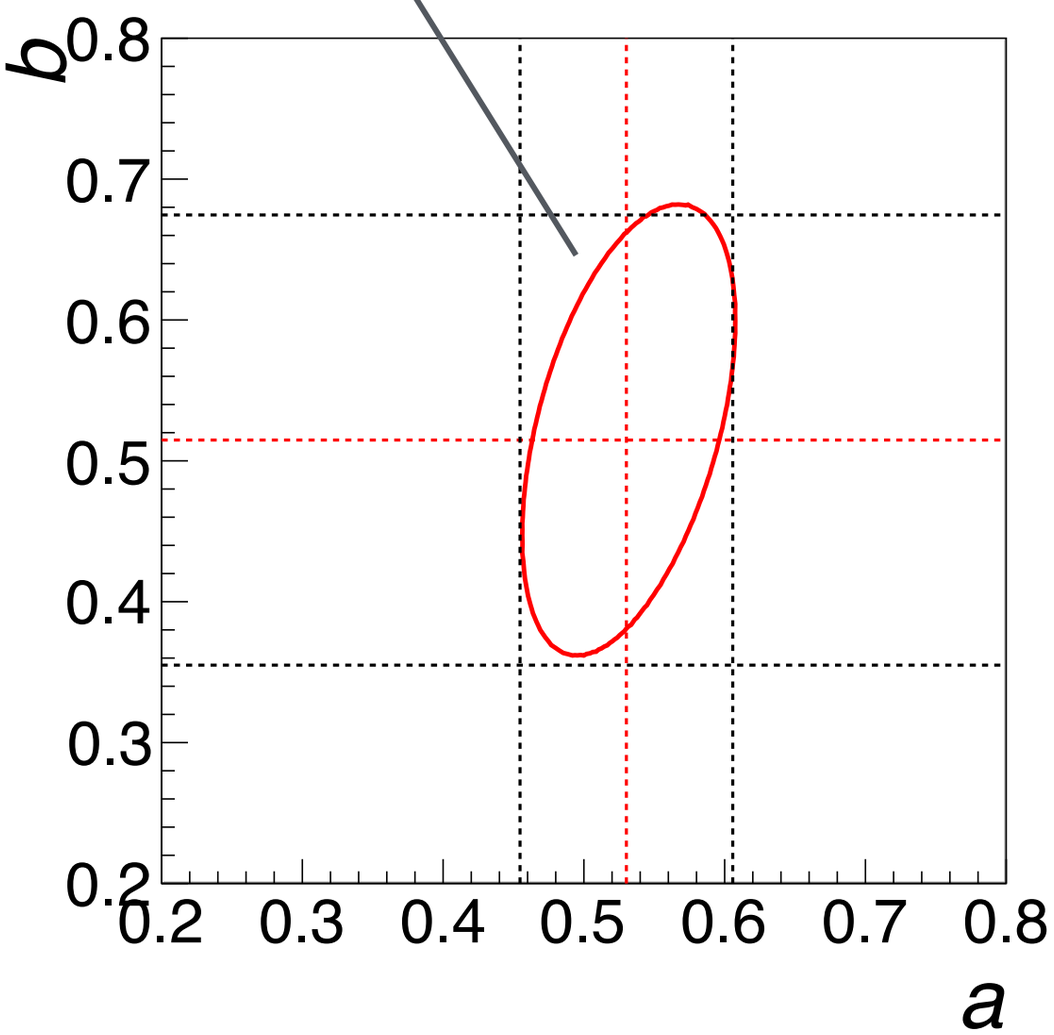
```
**********
**    3 **MIGRAD
**********
MIGRAD MINIMIZATION HAS CONVERGED.
MIGRAD WILL VERIFY CONVERGENCE AND ERROR MATRIX.
FCN=606.524 FROM MIGRAD    STATUS=CONVERGED      37 CALLS          38 TOTAL
                     EDM=2.20925e-08    STRATEGY= 1      ERROR MATRIX ACCURATE
  EXT PARAMETER                                   STEP         FIRST
  NO.   NAME      VALUE            ERROR          SIZE      DERIVATIVE
   1  a           5.30296e-01   7.55623e-02   1.13055e-03   1.84309e-03
   2  b           5.14883e-01   1.59791e-01   2.39145e-03  -9.42268e-04
                               ERR DEF= 0.5
**********
**    4 **SHOW COV
**********
EXTERNAL ERROR MATRIX.    NDIM=  25    NPAR=  2    ERR DEF=0.5
 5.710e-03  5.750e-03
 5.750e-03  2.553e-02
PARAMETER  CORRELATION COEFFICIENTS
      NO.  GLOBAL      1      2
       1  0.47626   1.000  0.476
       2  0.47626   0.476  1.000
```

FCN: value of function (– ln L in our case) at minimum

covariance and correlation matrix of the two parameters

Example: 2 Parameter ML Fit (Error Ellipse)

$$\ln L(a, b) = \ln L(a, b)_{\max} - \frac{1}{2}$$




```
// draw error ellipse: 200 points, parameters 0 and 1
TGraph *gr = (TGraph *)m.Contour(200, 0, 1);
TH2F *fr2 = new TH2F("fr2", "fr2", 1, 0.2, 0.8, 1, 0.2, 0.8);
fr2->SetXTitle("#it{a}");
fr2->SetYTitle("#it{b}");
fr2->Draw();
gr->SetLineColor(kRed);
gr->SetLineWidth(3);
gr->Draw("l");

TLine l;
l.SetLineWidth(2);
l.SetLineStyle(2); // dashed
l.SetLineColor(kRed);
l.DrawLine(a, 0.2, a, 0.8);
l.DrawLine(0.2, b, 0.8, b);
l.SetLineColor(kBlack);
l.DrawLine(a - a_err, 0.2, a - a_err, 0.8);
l.DrawLine(a + a_err, 0.2, a + a_err, 0.8);
l.DrawLine(0.2, b - b_err, 0.8, b - b_err);
l.DrawLine(0.2, b + b_err, 0.8, b + b_err);
```

Extended Maximum Likelihood Method (I)

In the standard ML method the information about the unknown parameters is encoded in the shape of the distribution of the data x_i.

Sometimes the number of observed events also contains information about the parameters, e.g., when we measure a rate.

Normal ML method:

$$\int f(x, \vec{\theta})\, \mathrm{d}x = 1$$

Extended ML method:

$$\int q(x, \vec{\theta})\, \mathrm{d}x = \nu(\vec{\theta}) = \text{ predicted number of events}$$

Extended Maximum Likelihood Method (II)

Normalized pdf:

$$\int f(x, \vec{\theta})\, \mathrm{d}x = 1$$

Likelihood function:

$$L(\vec{\theta}) = \frac{\nu^n e^{-\nu}}{n!} \prod_{i=1}^{n} f(x_i; \vec{\theta}) \qquad \text{where} \;\; \nu \equiv \nu(\vec{\theta})$$

Log-Likelihood function:

$$\ln L(\vec{\theta}) = -\ln(n!) - \nu(\vec{\theta}) + \sum_{i=1}^{n} \ln[f(x_i; \vec{\theta})\nu(\vec{\theta})]$$

ln(n!) does not depend on the parameters. So we need to minimize:

$$-\ln \tilde{L}(\vec{\theta}) = \nu(\vec{\theta}) - \sum_{i=1}^{n} \ln[f(x_i; \vec{\theta})\nu(\vec{\theta})]$$

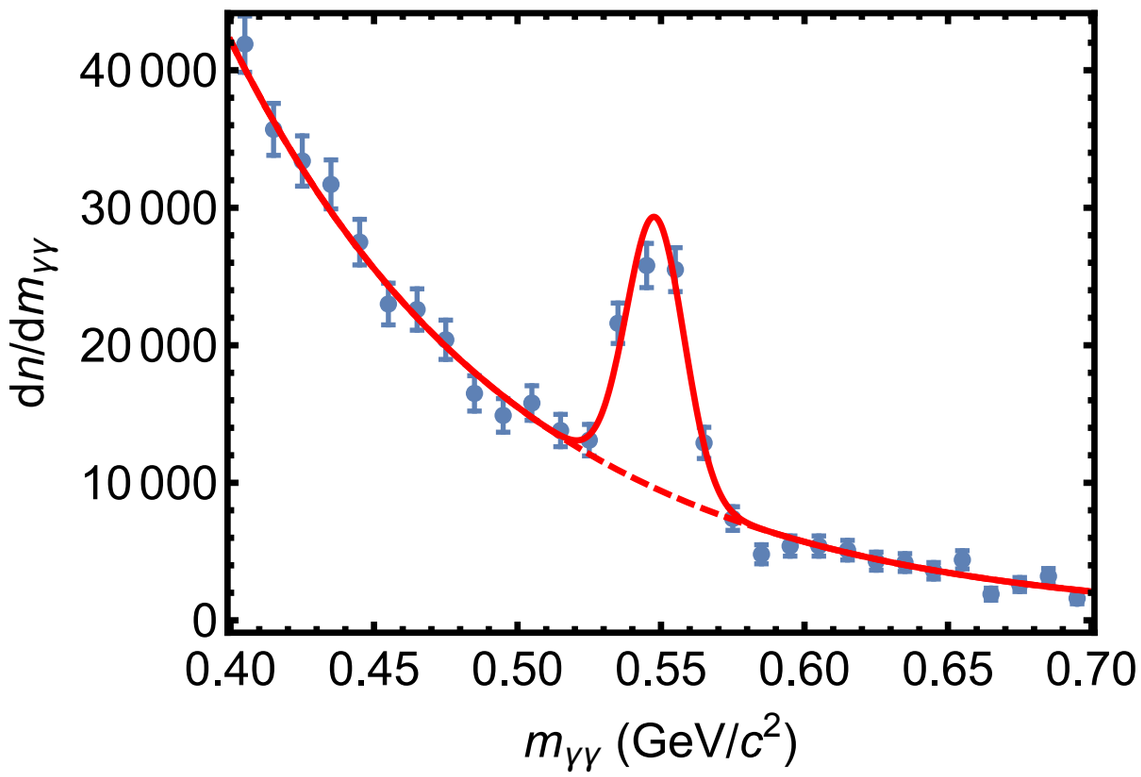
prediction for total number of events

Application of the Extended ML Method: Linear Combination of Signal and Background PDF (I)

Normalized pdf:

$$f(x; r_s, \vec{\theta}) = r_s f_s(x, \vec{\theta}) + (1 - r_s) f_b(x, \vec{\theta}), \qquad r_s = \frac{s}{s+b}, \quad 1 - r_s = \frac{b}{s+b}$$

$$-\ln L(s, b, \vec{\theta}) = +\ln(n!) + s + b - \sum_{i=1}^{n} \ln[s\, f_s(x_i, \vec{\theta}) + b\, f_b(x_i, \vec{\theta})]$$



Example

- Two-component fit (signal + background)
- Histogram only for visual representation
- We obtain a meaningful estimate of the uncertainties of *s* and *b*

Application of the Extended ML Method: Linear Combination of Signal and Background PDF (II)

Discussion:

We could have just fitted the normalized pdf:

$$f(x; r_s, \vec{\theta}) = r_s f_s(x, \vec{\theta}) + (1 - r_s) f_b(x, \vec{\theta}), \qquad r_s = \frac{s}{s+b}, \;\; 1 - r_s = \frac{b}{s+b}$$

Good estimate of the number of signal events: $r_s\, n$

However, $\sigma_{r_s}\, n$ is not a good estimate of the variation of the number of signal events (ignores fluctuations of n)

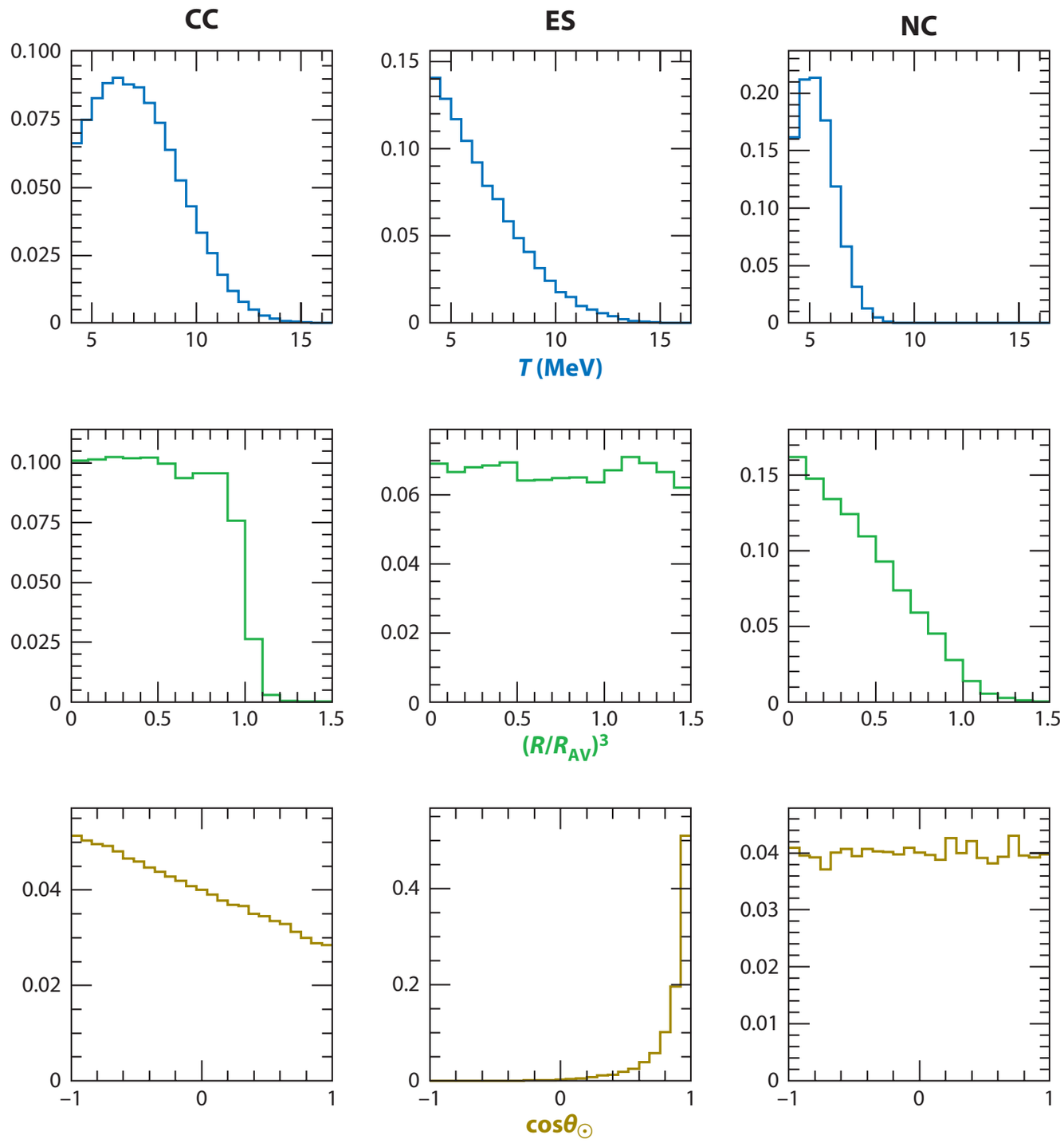
[C. Blocker, Maximum Likelihood Primer]

Real World Example of the Extended ML Method: Determination of Neutrino Fluxes in the SNO Exp.

energy

radius

direction



CC (only ν_e):

$$\nu_e + \mathrm{d} \rightarrow \mathrm{p} + \mathrm{p} + e^-$$

NC (all ν types):

$$\nu_i + \mathrm{d} \rightarrow \mathrm{p} + \mathrm{n} + \nu_i$$

ES (all ν types, mostly ν_e):

$$\nu_i + e^- \rightarrow \nu_i + e^-$$

The energy, radial, and directional distributions used to build probability density distributions to fit the SNO signal data.

$$N(E, r, \cos\theta) = N_{\mathrm{CC}}\, f_{\mathrm{CC}}(E, r, \cos\theta) + N_{\mathrm{ES}}\, f_{\mathrm{ES}}(E, r, \cos\theta) + N_{\mathrm{NC}}\, f_{\mathrm{NC}}(E, r, \cos\theta)$$

Annu. Rev. Nucl. Part. Sci. 2009.59:431

Maximum Likelihood Fits with Binned Data (I)

Common practice: data put into a histogram: $\vec{n} = (n_1, ..., n_k)$, $n_{\text{tot}} = \sum_{i=1}^{k} n_i$

Model prediction for the expected counts in bin i:

$$\nu_i(\vec{\theta}) = \nu_{\text{tot}} \int_{\text{bin } i} f(x; \vec{\theta})\, \mathrm{d}x \qquad \nu_i(\vec{\theta}) = (\nu_1, ..., \nu_k) \qquad \nu_{\text{tot}} = \sum_{i=1}^{N} \nu_i$$

If n_{tot} is fixed the probability to get a certain $\vec{n}$ is given by the multinomial distribution.

Multinomial distribution (generalization of binomial distribution):

→ k different possible outcomes, probability for outcome i is p_i, $\sum_{i=1}^{k} p_i = 1$

$$f(\vec{n}; n_{\text{tot}}, \vec{p}) = \frac{n_{\text{tot}}!}{n_1! \cdot ... \cdot n_k!}\, p_1^{n_1} \cdot ... \cdot p_k^{n_k} \qquad \vec{p} = (p_1, ..., p_k)$$

Maximum Likelihood Fits with Binned Data (II)

With $p_i = \nu_i / n_{\text{tot}}$ we write the likelihood of a certain $n_1, \ldots, n_k$ outcome as:

$$L(\vec{\theta}) = \frac{n_{\text{tot}}!}{n_1! \cdot \ldots \cdot n_k!} \left(\frac{\nu_1}{n_{\text{tot}}}\right)^{n_1} \cdot \ldots \cdot \left(\frac{\nu_k}{n_{\text{tot}}}\right)^{n_k} \qquad \nu_i(\vec{\theta}) = (\nu_1, \ldots, \nu_k)$$

Log-likelihood function:

$$\ln L(\vec{\theta}) = \sum_{i=1}^{k} n_i \ln \nu_i(\vec{\theta}) + C$$

Limit of zero bin width → usual unbinned maximum likelihood method

Treat the n_i as Poisson-distributed (n_{tot} fluctuates) → extended log-likelihood:

$$L(\vec{\theta}) = \prod_{i=1}^{k} \frac{\nu_i^{n_i}}{n_i!} e^{-\nu_i} \quad \rightarrow \quad \ln L(\vec{\theta}) = \sum_{i=1}^{k} n_i \ln \nu_i - \nu_i = -\nu_{\text{tot}} + \sum_{i=1}^{k} n_i \ln \nu_i$$

Relation to Bayesian Parameter Estimation

Bayesian posterior distribution:

$$p(\vec{\theta};\vec{x}) = \frac{L(\vec{x};\vec{\theta})\pi(\vec{\theta})}{\int L(\vec{x};\vec{\theta})\pi(\vec{\theta})\,\mathrm{d}\vec{\theta}}$$

Posterior distribution contains all information about the estimated parameters.

Often the mode (most probable value) of the posterior distribution is reported

→ Coincides with ML estimate for a flat prior distribution

Marginalization in case one is interested in only one parameter of the Bayesian posterior distribution:

$$p(\theta_j;\vec{x}) = \int p(\vec{\theta};\vec{x})\,\mathrm{d}\vec{\theta}_{k\neq j} = \frac{\int L(\vec{x};\vec{\theta})\pi(\vec{\theta})\,\mathrm{d}\vec{\theta}_{k\neq j}}{\int L(\vec{x};\vec{\theta})\pi(\vec{\theta})\,\mathrm{d}\vec{\theta}}$$

The Method of Least Squares

Least Squares from ML (I)

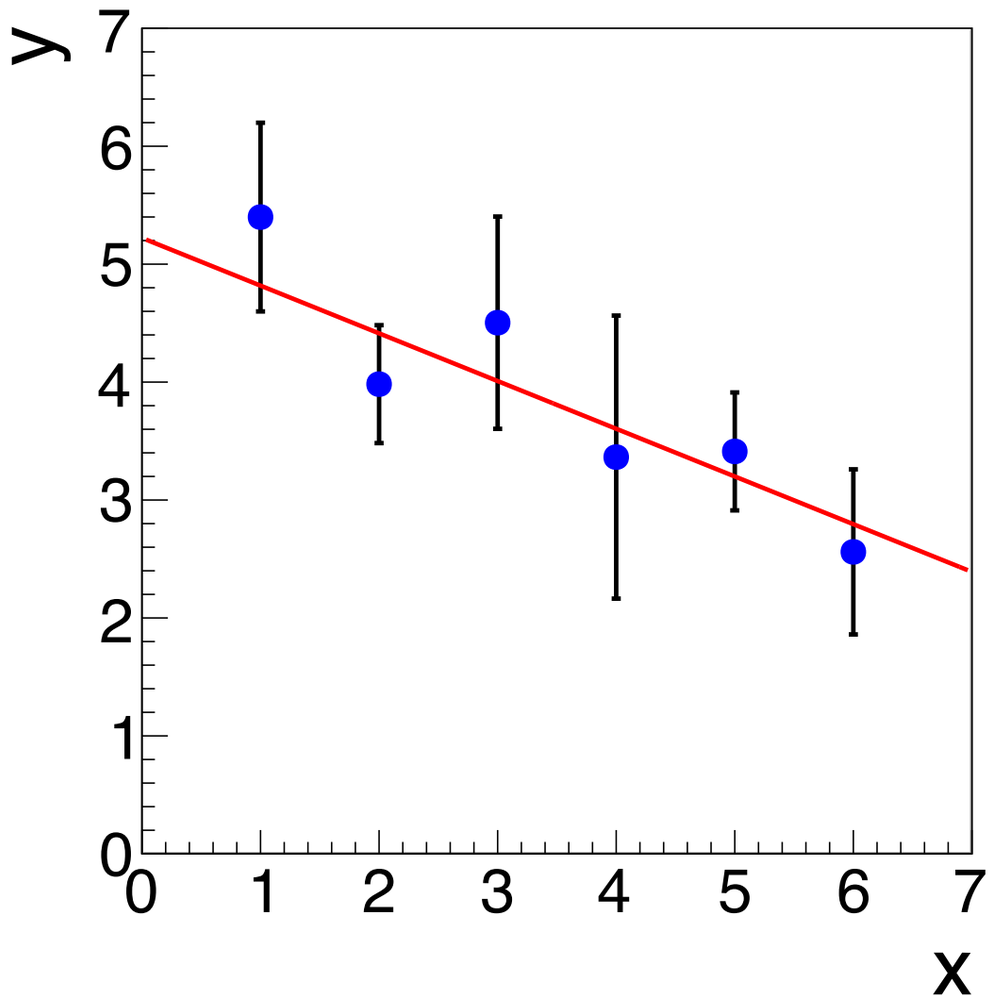
Consider n measured values $y_1(x_1)$, $y_2(x_2)$, … , $y_n(x_n)$ assumed to be independent Gaussian random variables with known variances:

$$V[y_i] = \sigma_i^2$$

Assume we have a function f with

$$E[y_i] = f(x_i; \vec{\theta})$$

We want to estimate $\vec{\theta}$



Likelihood function:

$$L(\vec{\theta}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left[-\frac{1}{2}\left(\frac{y_i - f(x_i; \vec{\theta})}{\sigma_i}\right)^2\right]$$

Least Squares from ML (II)

Log-likelihood function:

$$\ln L(\vec{\theta}) = -\frac{1}{2}\sum_{i=1}^{n}\left(\frac{y_i - f(x_i;\vec{\theta})}{\sigma_i}\right)^2 + \text{terms not depending on } \vec{\theta}$$

So maximizing the likelihood is equivalent to minimizing

$$\chi^2(\vec{\theta}) = \sum_{i=1}^{n}\left(\frac{y_i - f(x_i;\vec{\theta})}{\sigma_i}\right)^2$$

Minimizing χ^2 is called the method of least squares, goes back to Gauss and Legendre.

In other words, for Gaussian uncertainties the method of least squares coincides with the maximum likelihood method.

Minimization: $\dfrac{\partial\chi^2}{\partial\theta_j} = 0, \qquad j = 1, \ldots, m$ — Number of parameters

The χ^2 minimization is done numerically, e.g., using the MINUIT code

https://en.wikipedia.org/wiki/MINUIT

Generalized Least Squares for Correlated y_i

Suppose the y_i have a covariance matrix V and follow a multi-variate Gaussian:

$$g(\vec{y};\vec{\mu},V) = \frac{1}{(2\pi)^{n/2}|V|^{1/2}} \exp\left[-\frac{1}{2}(\vec{y}-\vec{\mu})^{\mathsf{T}} V^{-1}(\vec{y}-\vec{\mu})\right]$$

The generalized least-squares method then corresponds to minimizing:

$$\chi^2(\vec{\theta}) = (\vec{y}-\vec{f}(\vec{x};\vec{\theta}))^T V^{-1}(\vec{y}-\vec{f}(\vec{x};\vec{\theta}))$$

$$\vec{f}(\vec{x};\vec{\theta}) = (f(x_1;\vec{\theta}),\ldots,f(x_n;\vec{\theta}))$$

We can write this also as

$$\chi^2(\vec{\theta}) = \sum_{i,j}(y_i - f(x_i;\vec{\theta}))^T (V^{-1})_{ij}(y_j - f(x_j;\vec{\theta}))$$

Variance of the Least Squares Estimators

Using

$$\chi^2(\vec{\theta}) = -2 \ln L(\theta) + \text{const.}$$

we can use the result for the variance of the ML estimators and obtain

$$V[\hat{\vec{\theta}}] \approx 2 \left[\left. \frac{\partial^2 \chi^2(\vec{\theta})}{\partial^2 \vec{\theta}} \right|_{\vec{\theta}=\hat{\vec{\theta}}} \right]^{-1}$$

or equivalently:

$$(V^{-1}[\hat{\vec{\theta}}])_{ij} = \frac{1}{2} \left. \frac{\partial^2 \chi^2(\vec{x};\vec{\theta})}{\partial\theta_i \partial\theta_j} \right|_{\vec{\theta}=\hat{\vec{\theta}}}$$

Or determine 1σ uncertainties from the contour where

$$\chi^2(\vec{\theta}') = \chi^2_{\text{min}} + 1$$

For z·σ uncertainties the condition is

$$\chi^2(\vec{\theta}') = \chi^2_{\text{min}} + z^2$$

Linear Least Squares (I)

Consider a function linear in the parameters:

$$f(x;\vec{\theta}) = \sum_{j=1}^{m} a_j(x)\theta_j$$

n data points y_i

m parameters θ_j

χ^2 in matrix form:

$$\chi^2 = (\vec{y} - A\vec{\theta})^\mathsf{T} V^{-1} (\vec{y} - A\vec{\theta}),$$

$$= \vec{y} V^{-1} \vec{y} - 2\vec{y}^\mathsf{T} V^{-1} A\vec{\theta} + \vec{\theta}^\mathsf{T} A^\mathsf{T} V^{-1} A\vec{\theta}$$

A is a $n \times m$ matrix

$$A_{i,j} = a_j(x_i)$$

Set derivatives w.r.t. θ_i to zero:

$$\nabla\chi^2 = -2(A^\mathsf{T} V^{-1}\vec{y} - A^\mathsf{T} V^{-1} A\vec{\theta}) = 0$$

$$\vec{\nabla}(\vec{a}^\mathsf{T} M\vec{x}) = M^\mathsf{T}\vec{a} \qquad \vec{\nabla}(\vec{x}^\mathsf{T} M\vec{x}) = (M^\mathsf{T} + M)\vec{x} \overset{M \text{ symm.}}{=} 2M\vec{x}$$

Solution:

$$\widehat{\vec{\theta}} = (A^\mathsf{T} V^{-1} A)^{-1} A^\mathsf{T} V^{-1}\vec{y} \equiv L\vec{y}$$

Linear Least Squares (II)

Covariance matrix U from error propagation (exact, because estimated parameter vector is a linear function of the data points y_i)

$$L = \underbrace{(A^\mathsf{T} V^{-1} A)^{-1}}_{\text{symmetric } m \times m \text{ matrix}} A^\mathsf{T} V^{-1}$$

Covariance matrix U of the parameters:

$$\begin{aligned} U &= LVL^\mathsf{T} \\ &= (A^\mathsf{T} V^{-1} A)^{-1} A^\mathsf{T} V^{-1} V V^{-1} A (A^\mathsf{T} V^{-1} A)^{-1} \\ &= (A^\mathsf{T} V^{-1} A)^{-1} \end{aligned}$$

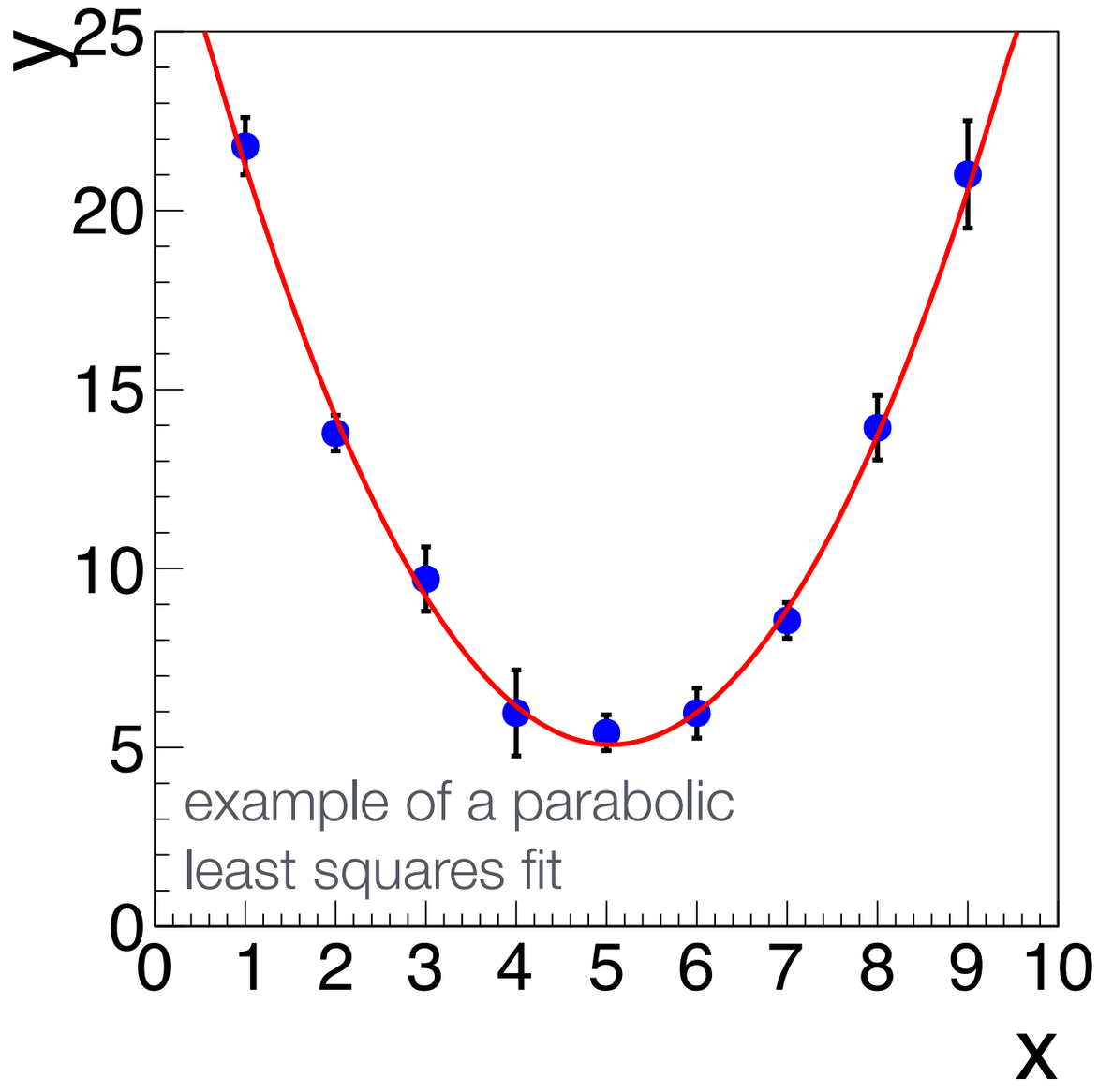
Here we use
$(XY)^\mathsf{T} = Y^\mathsf{T} X^\mathsf{T}$,
$[(A^\mathsf{T} V^{-1} A)^{-1}]^\mathsf{T} = (A^\mathsf{T} V^{-1} A)^{-1}$

Equivalently, calculate:

$$(U^{-1})_{ij} = \frac{1}{2} \left[\frac{\partial \chi^2}{\partial \theta_i \partial \theta_j} \right]_{\vec{\theta} = \hat{\vec{\theta}}}$$

Examples of Linear Least Squares Fits

- Constant function $y = \theta$
- Straight-line fit $y = \theta_0 + \theta_1 x$
- Parabolic fit $y = \theta_0 + \theta_1 x + \theta_2 x^2$
- Any polynomial fit
- Functions like $y = \theta \sin x$
  or $y = \theta \exp x$



Linear least square fit ≠ straight line fit

Example: Straight Line Fit: $y = \theta_0 + \theta_1 \cdot x$ (I)

The conditions $d\chi^2/d\theta_0$ and $d\chi^2/d\theta_1$ give two linear equations with two variables which is easy to solve.

Here we use the general solutions from the previous slide:

$$L = (A^T V^{-1} A)^{-1} A^T V^{-1} \qquad \hat{\vec{\theta}} = L\vec{y}$$

$$A^T = \begin{pmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \end{pmatrix} \qquad \vec{\theta} = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} \qquad V^{-1} = \begin{pmatrix} 1/\sigma_1^2 & & & \\ & 1/\sigma_2^2 & & \\ & & \ddots & \\ & & & 1/\sigma_n^2 \end{pmatrix}$$

$$A^T V^{-1} = \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \dots & 1/\sigma_n^2 \\ x_1/\sigma_1^2 & x_2/\sigma_2^2 & \dots & x_n/\sigma_n^2 \end{pmatrix}$$

$$A^T V^{-1} A = \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \dots & 1/\sigma_n^2 \\ x_1/\sigma_1^2 & x_2/\sigma_2^2 & \dots & x_n/\sigma_n^2 \end{pmatrix} \cdot \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} = \begin{pmatrix} \sum_i \frac{1}{\sigma_i^2} & \sum_i \frac{x_i}{\sigma_i^2} \\ \sum_i \frac{x_i}{\sigma_i^2} & \sum_i \frac{x_i^2}{\sigma_i^2} \end{pmatrix}$$

Example: Straight Line Fit: $y = \theta_0 + \theta_1 \cdot x$ (II)

The 2×2 matrix is easy to invert:

$$(A^T V^{-1} A)^{-1} = \frac{1}{[1][x^2] - [x][x]} \begin{pmatrix} [x^2] & -[x] \\ -[x] & [1] \end{pmatrix}$$

shorthand notation for the sum

$$\text{where} \quad [z] := \sum_i \frac{z}{\sigma_i^2}$$

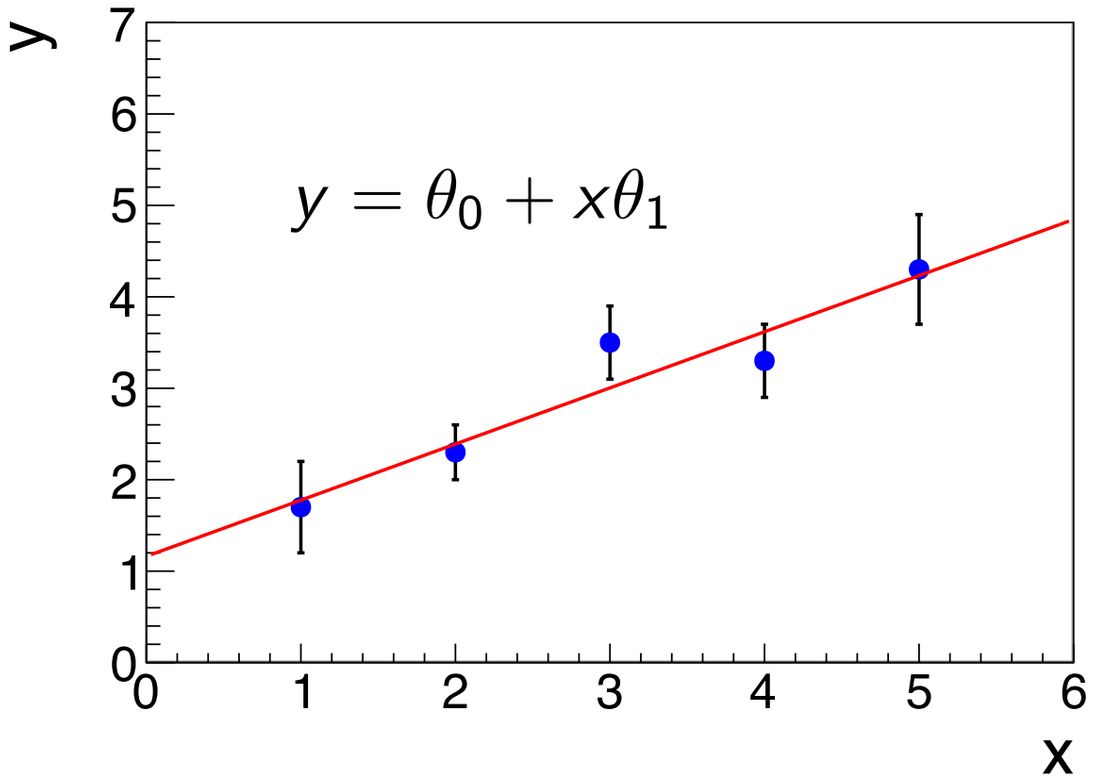
This gives:

$$\begin{aligned} L &= (A^T V^{-1} A)^{-1} A^T V^{-1} \\ &= \frac{1}{[1][x^2] - [x][x]} \begin{pmatrix} [x^2] & -[x] \\ -[x] & [1] \end{pmatrix} \cdot \begin{pmatrix} 1/\sigma_1^2 & 1/\sigma_2^2 & \dots & 1/\sigma_n^2 \\ x_1/\sigma_1^2 & x_2/\sigma_2^2 & \dots & x_n/\sigma_n^2 \end{pmatrix} \\ &= \frac{1}{[1][x^2] - [x][x]} \begin{pmatrix} [x^2]\frac{1}{\sigma_1^2} - [x]\frac{x_1}{\sigma_1^2} & \dots & [x^2]\frac{1}{\sigma_n^2} - [x]\frac{x_n}{\sigma_n^2} \\ -[x]\frac{1}{\sigma_1^2} + [1]\frac{x_1}{\sigma_1^2} & \dots & -[x]\frac{1}{\sigma_n^2} + [1]\frac{x_n}{\sigma_n^2} \end{pmatrix} \end{aligned}$$

We finally obtain:

$$\hat{\theta}_0 = \frac{[x^2][y] - [x][xy]}{[1][x^2] - [x][x]} \qquad \hat{\theta}_1 = \frac{-[x][y] + [1][xy]}{[1][x^2] - [x][x]}$$

Example: Straight Line Fit: $y = \theta_0 + \theta_1 \cdot x$ (III)



x	y	σ_y
1	1.7	0.5
2	2.3	0.3
3	3.5	0.4
4	3.3	0.4
5	4.3	0.6

Fit result (analytic):

$$[z] := \sum_i \frac{z}{\sigma_i^2}$$

$$\hat{\theta}_0 = \frac{[x^2][y] - [x][xy]}{[1][x^2] - [x][x]} = 1.16207$$

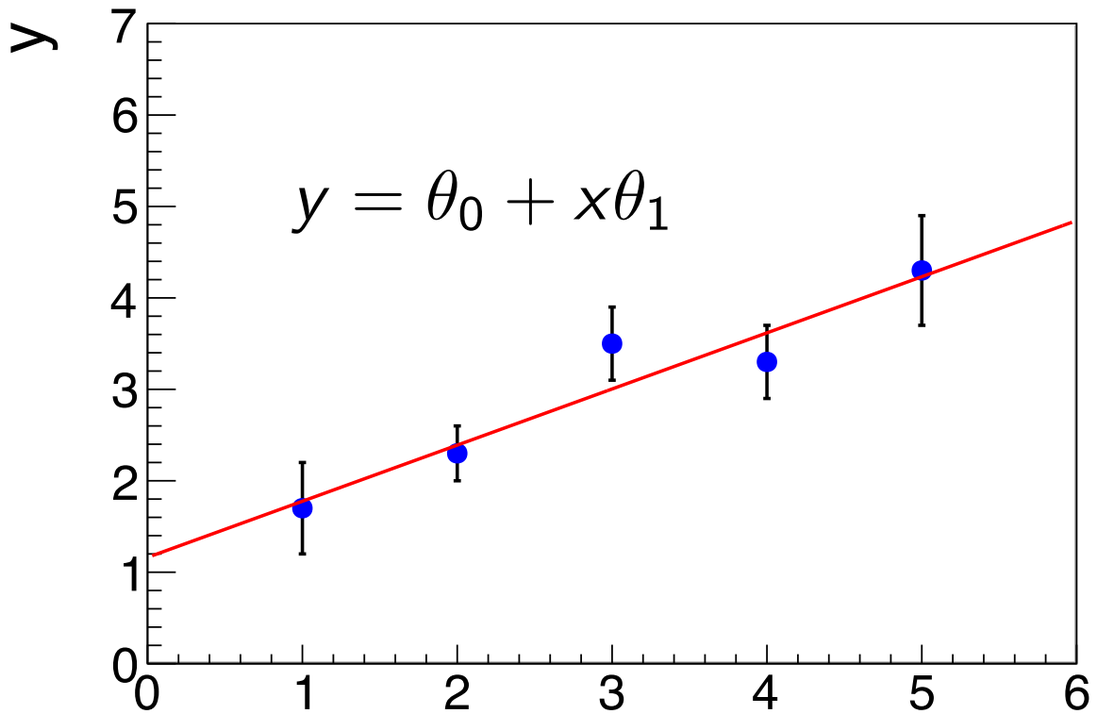
$$\hat{\theta}_1 = \frac{-[x][y] + [1][xy]}{[1][x^2] - [x][x]} = 0.613945$$

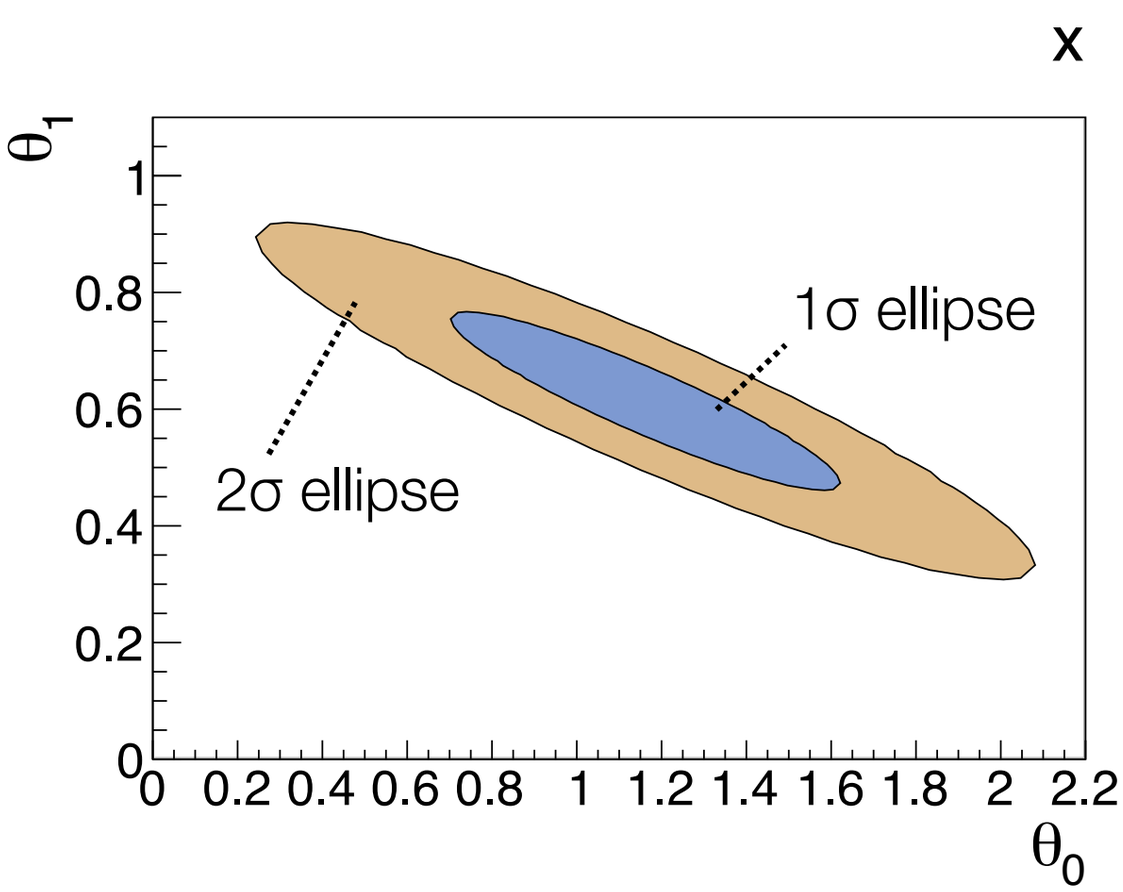
Covariance matrix of (θ_0, θ_1):

$$U = (A^T V^{-1} A)^{-1}$$

$$= \begin{pmatrix} 0.211186 & -0.0646035 \\ -0.0646035 & 0.0234105 \end{pmatrix}$$

Straight Line Fit: Comparison to MINIUT






```
// fit data points with linear function
TF1 *f = new TF1("f", "pol1", 0., 6.);
TFitResultPtr r = g->Fit("f", "F0qS", "", 0., 6.);
r->Print("V");
```

```
Minimizer is Minuit
Chi2                      =      2.29557
NDf                       =            3
Edm                       =  3.23988e-23
NCalls                    =           32
p0                        =      1.16207   +/-   0.45955
p1                        =     0.613945   +/-   0.153005

Covariance Matrix:
                               p0          p1
p0                        0.21119   -0.064603
p1                       -0.064603     0.02341

Correlation Matrix:
                               p0          p1
p0                             1    -0.91879
p1                       -0.91879           1
```

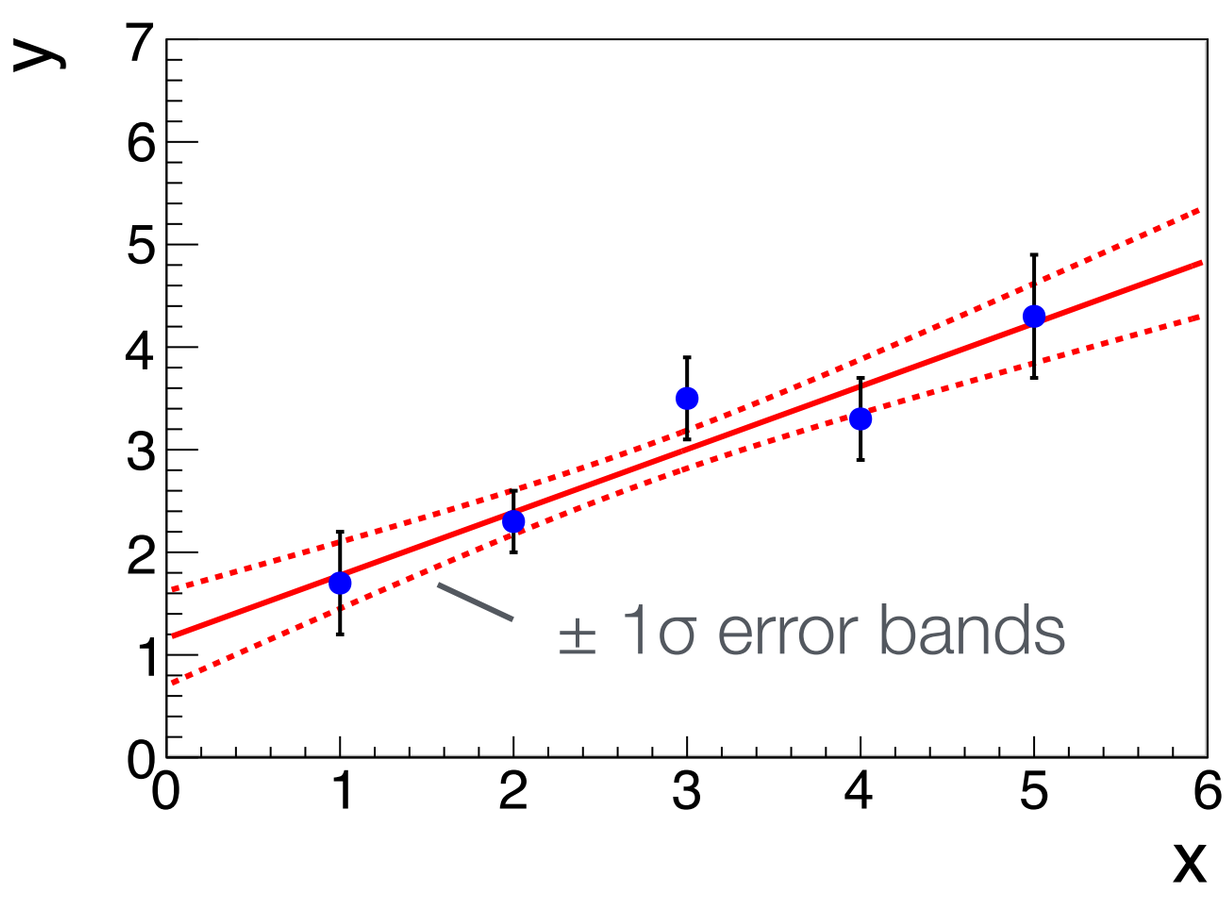
Propagation of Fit Parameter Uncertainties

$$y = \hat{\theta}_0 + \hat{\theta}_1 x \qquad A = \begin{pmatrix} \frac{\partial y}{\partial \hat{\theta}_0} \\ \frac{\partial y}{\partial \hat{\theta}_1} \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix}$$

$$\sigma_y^2 = A^\mathsf{T} V A = \begin{pmatrix} 1 & x \end{pmatrix} \begin{pmatrix} \sigma_0^2 & \mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] \\ \mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] & \sigma_1^2 \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}$$

$$= \begin{pmatrix} 1 & x \end{pmatrix} \begin{pmatrix} \sigma_0^2 + x\,\mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] \\ \mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] + x\sigma_1^2 \end{pmatrix}$$

$$= \sigma_1^2 x^2 + 2\mathrm{cov}[\hat{\theta}_0, \hat{\theta}_1] x + \sigma_0^2$$



Goodness-of-Fit

Least Squares Method: Goodness-of-Fit (I)

The minimum value of is a measure of the level of agreement between the model and the data;

$$\chi^2_{\text{min}} = \sum_{i=1}^{n} \left(\frac{y_i - f(x_i; \hat{\vec{\theta}})}{\sigma_i} \right)^2$$

Large χ^2_{min}: the model can can be rejected.

If the model is correct, then χ^2_{min} for repeated experiments follows a distribution

$$f(t; n_{\text{df}}) = \frac{1}{2^{n_{\text{df}}/2}\Gamma\left(\frac{n_{\text{df}}}{2}\right)} t^{n_{\text{df}}/2-1} e^{-t/2}, \qquad t = \chi^2_{\text{min}}$$

with $n_{\text{df}} = n - m = \text{number of data points} - \text{number of fit parameters}$

n_{df} = "number of degrees of freedom"

Least Squares Method: Goodness-of-Fit (II)

Expectation value of the χ^2 distribution is n_{df}
→ $\chi^2 \approx n_{\mathrm{df}}$ indicates a good fit

Consistency of a model with the data is quantified with the *p*-value:

$$p\text{-value} = \int_{\chi^2_{\mathrm{min}}}^{\infty} f(t; n_{\mathrm{df}})\, \mathrm{d}t$$

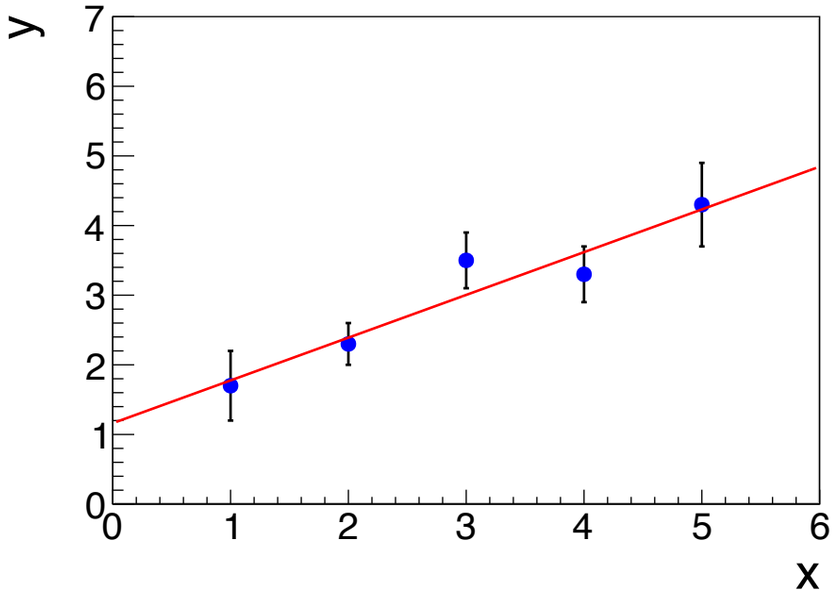
The *p*-value is the probability to get a χ^2_{min} as high as the observed one, or higher, if the model is correct.

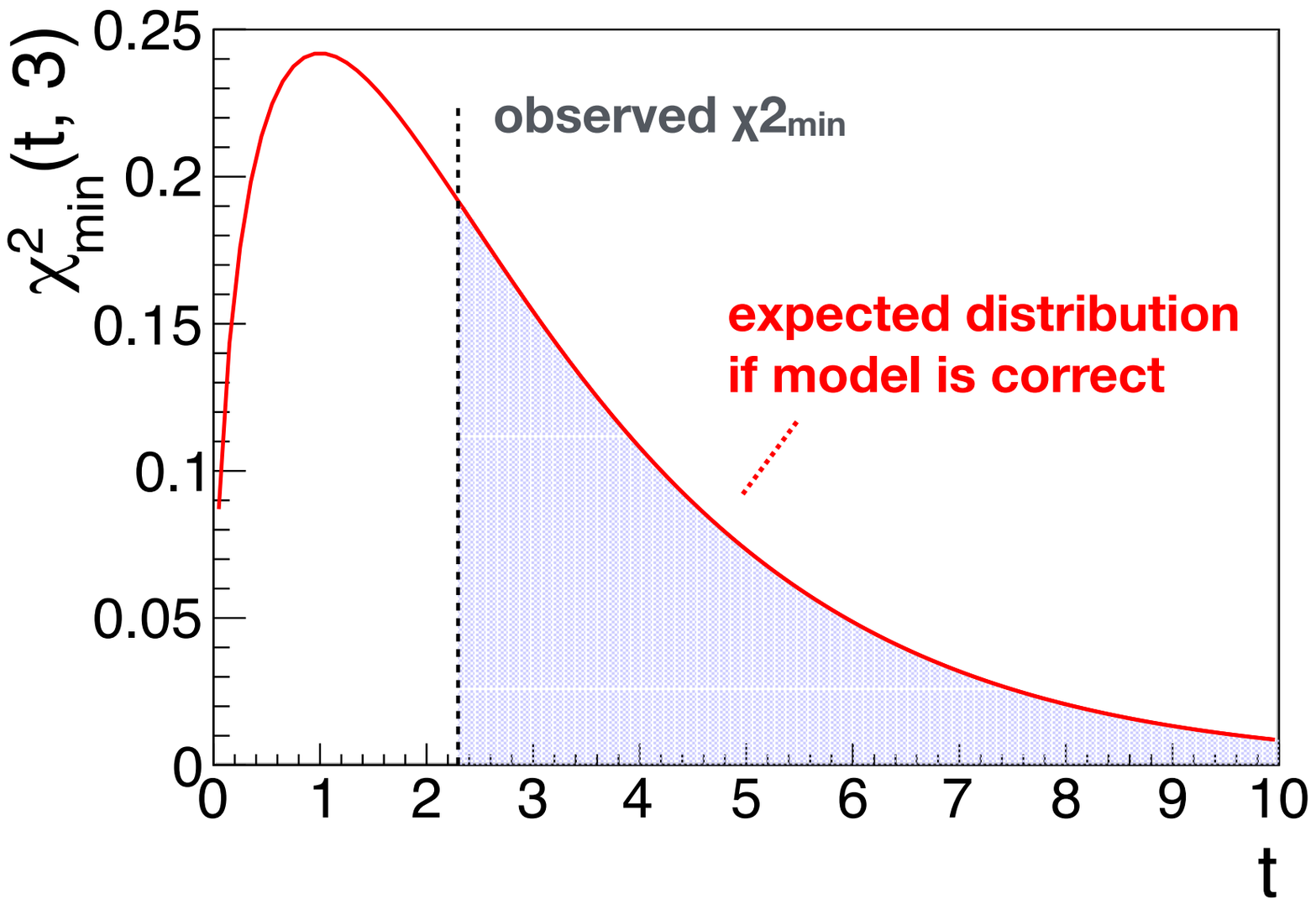
The *p*-value is **not** the probability that the model is correct.

p-value for the Straight Line Fit Example

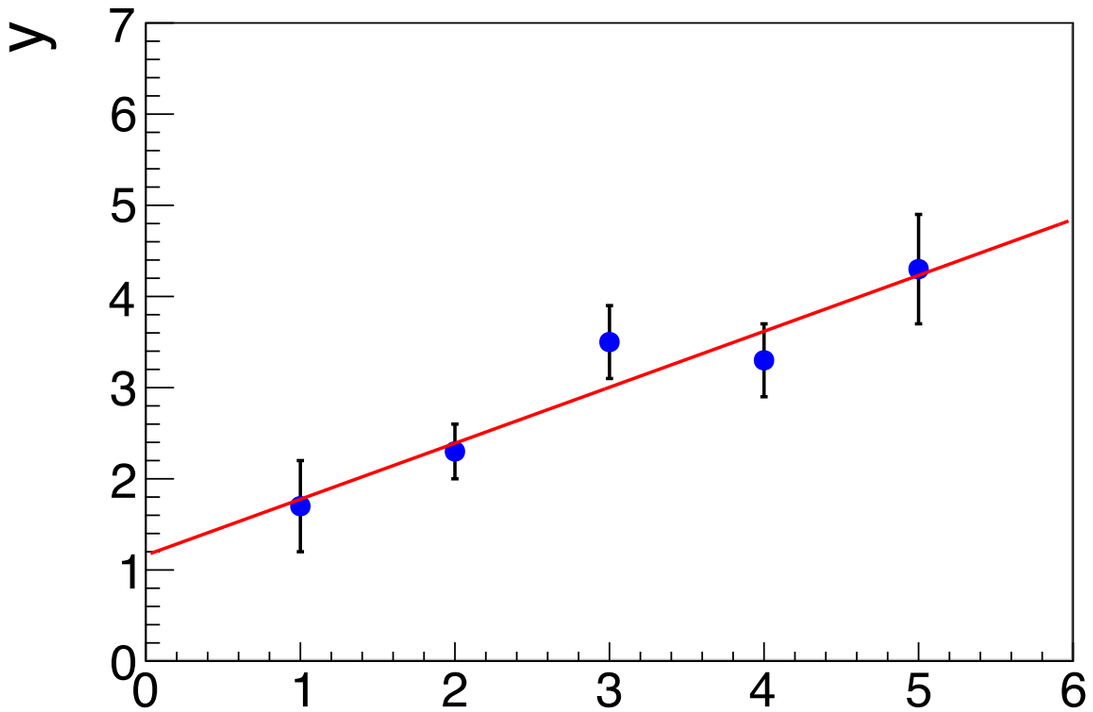
$\chi^2_{min} = 2.29557$, $n_{df} = 3$:

p-value = 0.51337





Constant Model ($y = \theta_0$) Rejected by Small p-value

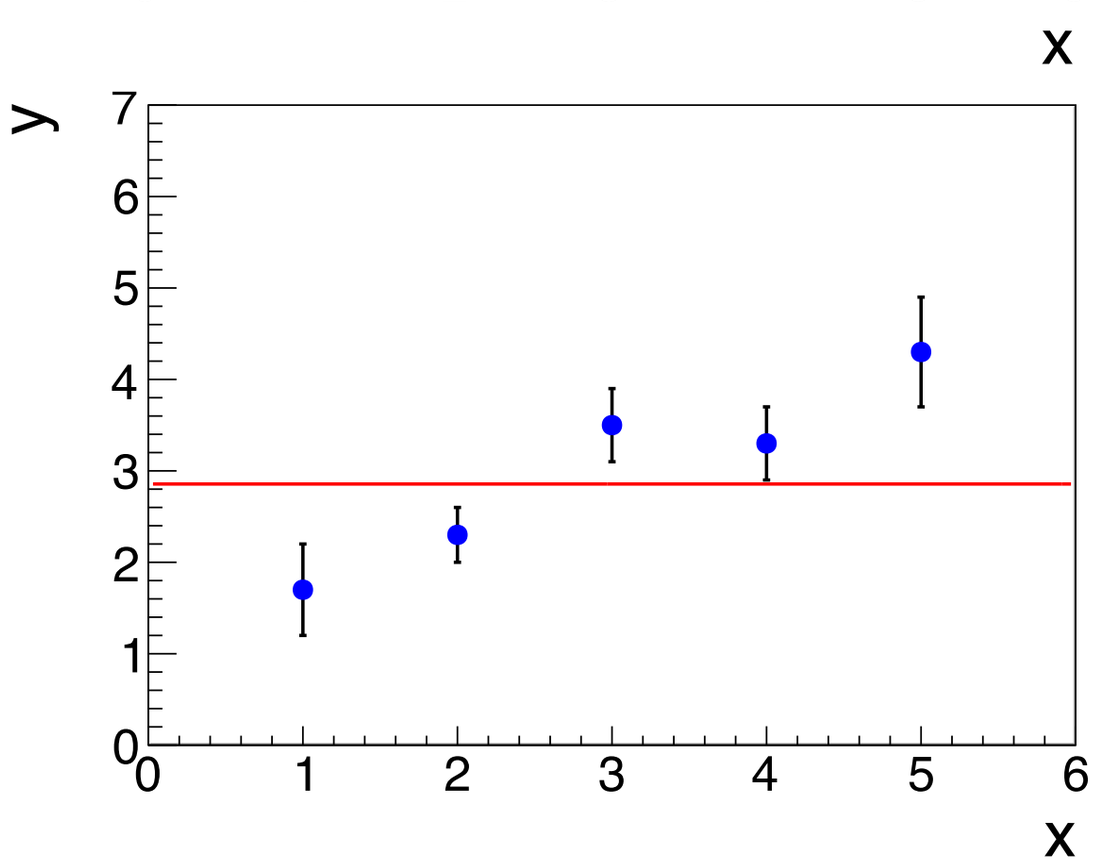


$\chi^2_{min} = 2.29557$, $n_{df} = 3$:

p-value = 0.51337

root [1] TMath::Prob(2.29557, 3)

(double) 0.513370



$\chi^2_{min} = 18.3964$, $n_{df} = 4$:

p-value = 0.001032

TMath::Prob(18.3964, 4)

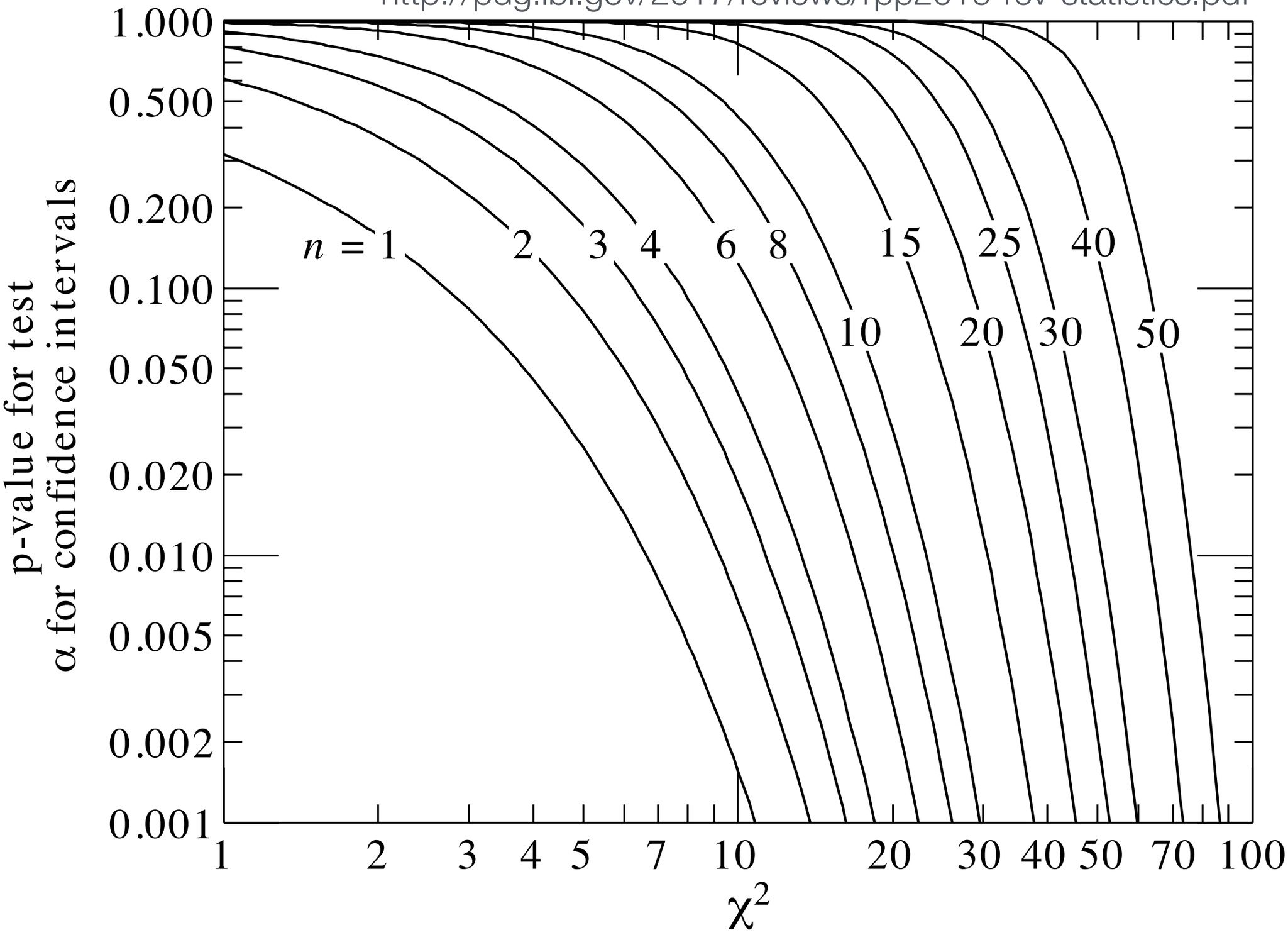
(double) 0.001032

$\theta_0 = 2.86 \pm 0.18$

stat. uncertainty of the fit parameter does not tell us whether model is correct

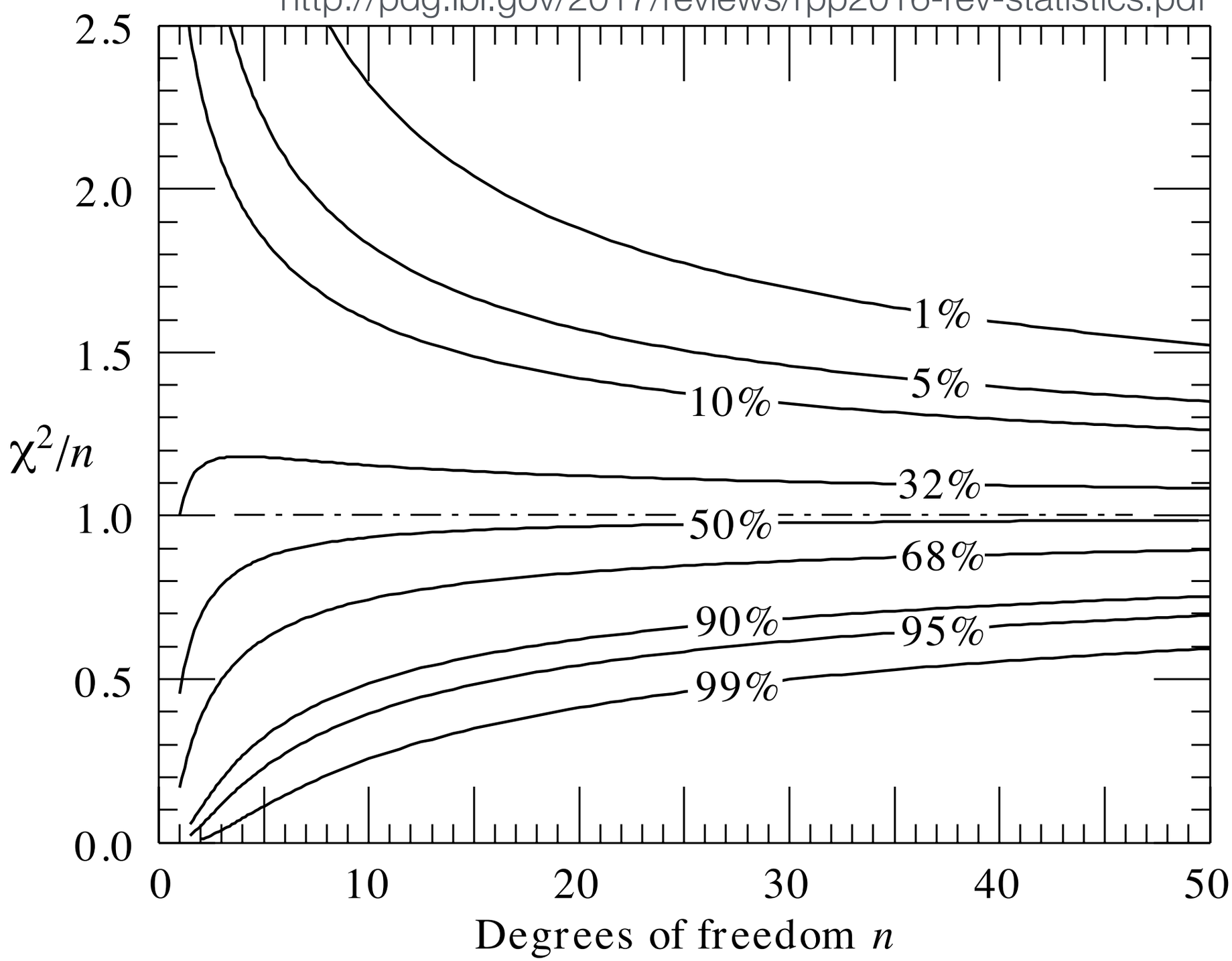
p-value for different χ^2_{min} and n_{df}

http://pdg.lbl.gov/2017/reviews/rpp2016-rev-statistics.pdf



Confidence Intervalls for χ^2_{min} / n_{df} as a fct. of n_{df}

http://pdg.lbl.gov/2017/reviews/rpp2016-rev-statistics.pdf



Least-Squares Fits to Histograms

Consider histogram with k bins and n_i counts in bin i. If n_i is not too small one can use the Gaussian approximation of the Poisson distribution and apply the least-squared method:

Pearson's χ^2:

$$\chi^2(\vec{\theta}) = \sum_{i=1}^{k} \frac{(n_i - \nu_i(\vec{\theta}))^2}{\nu_i(\vec{\theta})}$$

Neyman's χ^2:

$$\chi^2(\vec{\theta}) = \sum_{i=1}^{k} \frac{(n_i - \nu_i(\vec{\theta}))^2}{n_i}$$

Problems arise in bins with few entries (typically less than 5), in particular in Neyman's χ^2.

Bins with zero entries are problematic, typically omitted from the fit
→ leads to biased fit results

Goodness-of-Fit for Unbinned ML Fits (I)

In case of an unbinned ML fit one can put data and model prediction into a histogram and perform a χ^2 test.

Consider the ratio

L: likelihood

$$\lambda = \frac{L(\vec{n}|\vec{\nu})}{L(\vec{n}|\vec{n})}, \qquad \vec{\nu} = \vec{\nu}(\vec{\theta}), \quad \vec{\theta} = (\theta_1, ..., \theta_m)$$

For the multinomial ("M", n_{tot} fixed) and Poisson distributed data ("P") one obtains

k: number of bins of the histogram

$$\lambda_{\text{M}} = \prod_{i=1}^{k} \left(\frac{\nu_i}{n_i}\right)^{n_i}, \qquad \lambda_{\text{P}} = e^{n_{\text{tot}} - \nu_{\text{tot}}} \prod_{i=1}^{k} \left(\frac{\nu_i}{n_i}\right)^{n_i}$$

We then consider

$$\chi^2 := -2 \ln \lambda$$

Goodness-of-Fit for Unbinned ML Fits (II)

For multinomially distributed data in the large sample limit

$$\chi^2_{\mathrm{M}} := -2 \ln \lambda_{\mathrm{M}} = 2 \sum_{i=1}^{k} n_i \ln \frac{n_i}{\hat{\nu}_i}$$

follows a χ^2 distribution for $N - m - 1$ degrees of freedom.

In case of Poisson distributed data

$$\chi^2_{\mathrm{P}} := -2 \ln \lambda_{\mathrm{P}} = 2 \sum_{i=1}^{k} \left(n_i \ln \frac{n_i}{\hat{\nu}_i} + \hat{\nu}_i - n_i \right)$$

follows a χ^2 distribution for $N - m$ degrees of freedom in the large sample limit.

Goodness-of-Fit ML Test Using L_{max}

For ML fits the value of the likelihood function at the maximum $L_{max}(x|\theta_0) \equiv L_{max,obs}$ is sometimes used as a Goodness-of-Fit test

- Generate pseudo-data based on best-fit parameters
- Repeat fit with pseudo data → L_{max} distribution
- From the L_{max} distribution one can determine how likely it is to find a value $L_{max,obs}$ or smaller

However, this method is generally discouraged

- Biased and not invariant with respect to change of variables
- From J. Heinrich, PHYSTAT2003, arXiv:physics/0310167
  "The method is fatally flawed in the unbinned case. Don't use it. Complain when you see it used."

Weighted Average of Correlated Data Points

Consider n data points y_i with covariance matrix V: $\vec{y} = (y_1, y_2, \ldots, y_n)$

One can calculate a weighted average λ by minimizing

$$\chi^2(\lambda) = (\vec{y} - \vec{\lambda})^\mathsf{T} V^{-1} (\vec{y} - \vec{\lambda})$$

$\vec{\lambda} := (\lambda, \lambda, \ldots, \lambda)$

One obtains (here without calculation):

$$\hat{\lambda} = \sum_{i=1}^{N} w_i y_i \qquad w_i = \frac{\sum_{j=1}^{n} (V^{-1})_{i,j}}{\sum_{k,l=1}^{n} (V^{-1})_{k,l}}$$

Variance results from error propagation:

$$\sigma^2_{\hat{\lambda}} = \vec{w}^\mathsf{T} V \vec{w} = \sum_{i,j=1}^{n} w_i V_{ij} w_j$$

Minimizing the χ^2 gives the *best linear unbiased estimate* (BLUE) → linear unbiased estimator with the lowest variance

- BLUE combination may be biased if uncertainties not known or are estimated from measured values
- Improvement: iterative approach (rescaling uncertainties based on previous iteration)

Special Case:
Weighted Average of Two Correlated Measurements

Consider two measurements with covariance matrix V (ρ = correlation coeff.):

$$y_1,\ y_2 \qquad V = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}$$

Applying the formulas from the previous slide:

$$V^{-1} = \frac{1}{1-\rho^2} \begin{pmatrix} \frac{1}{\sigma_1^2} & \frac{-\rho}{\sigma_1\sigma_2} \\ \frac{-\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{pmatrix} \qquad \hat{\lambda} = wy_1 + (1-w)y_2$$

$$w = \frac{\sigma_2^2 - \rho\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

$$V[\hat{\lambda}] = \sigma^2 = \frac{(1-\rho^2)\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2}$$

equivalently:

$$\frac{1}{\sigma^2} = \frac{1}{1-\rho^2}\left[\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} - \frac{2\rho}{\sigma_1\sigma_2}\right]$$

Weighted Average of Correlated Measurements: An Interesting Example

Measure length of an object with two rulers, calibrated to be accurate at T_0. Temperature coefficients c_1 and c_2 of the rulers known. Estimates of the true length:

$$y_i = L_i + \underbrace{c_i(T - T_0)}_{\text{correction for temperature dependence of the rulers}}$$

Now we would like to take the weighted average of the two measurements y_i:

$$\sigma_i^2 = \sigma_L^2 + c_i\sigma_T^2, \quad \text{cov}[y_1, y_2] = c_1 c_2 \sigma_T^2$$

We use the following parameters:

$$c_1 = 0.1, \quad L_1 = 2.0 \pm 0.1, \quad y_1 = 1.80 \pm 0.22, \quad T_0 = 25°\text{C}$$
$$c_2 = 0.2, \quad L_2 = 2.3 \pm 0.1, \quad y_2 = 1.90 \pm 0.41, \quad T = (23 \pm 2)°\text{C}$$

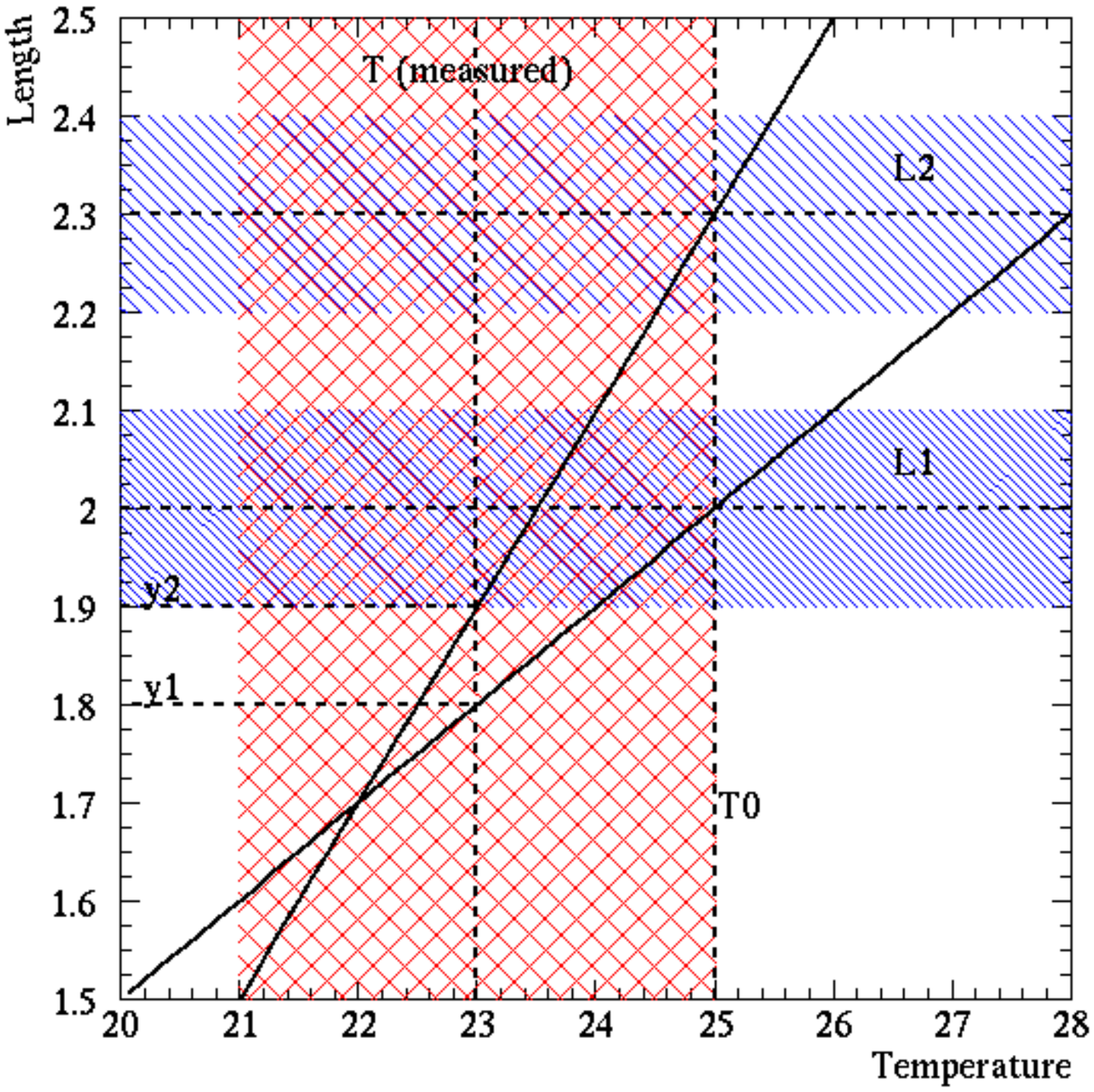
and obtain the following weighted average:

$$y = 1.75 \pm 0.19$$

Weird: the weighted average does not lie between y_1 and y_2. What is going on?

Taken from http://www.phas.ubc.ca/~oser/p509/Lec_10.pdf (an example adapted from Cowan's book)

Weighted Average of Correlated Measurements: An Interesting Example



y_1 and y_2 calculated assuming $T = 23°C$

Fit adjusts temperature and finds best agreement at $T = 22°C$

Temperature in these measurements is a nuisance parameter

We have an example in which data themselves provide information about a nuisance parameter

Taken from http://www.phas.ubc.ca/~oser/p509/Lec_10.pdf (an example adapted from Cowan's book)

PDG Averaging Procedure (I)

Treatment of correlated systematic uncertainties:

In fitting or averaging, we usually do not include correlations between different measurements, but we try to select data in such a way as to reduce correlations. Correlated errors are, however, treated explicitly when there are a number of results of the form $A_i \pm \sigma_i \pm \Delta$ that have identical systematic errors Δ. In this case, one can first average the $A_i \pm \sigma_i$ and then combine the resulting statistical error with Δ. One obtains, however, the same result by averaging $A_i \pm (\sigma_i^2 + \Delta_i^2)^{1/2}$, where $\Delta_i = \sigma_i \Delta [\sum (1/\sigma_j^2)]^{1/2}$. This procedure has the advantage that, with the modified systematic errors Δ_i, each measurement may be treated as independent and averaged in the usual way with other data. Therefore, when appropriate, we adopt this procedure. We tabulate Δ and invoke an automated procedure that computes Δ_i before averaging and we include a note saying that there are common systematic errors.

http://pdg.lbl.gov/2017/reviews/rpp2016-rev-rpp-intro.pdf

PDG Averaging Procedure (II)

http://pdg.lbl.gov/2017/reviews/rpp2016-rev-rpp-intro.pdf

5.2.2. *Unconstrained averaging:* To average data, we use a standard weighted least-squares procedure and in some cases, discussed below, increase the errors with a "scale factor." We begin by assuming that measurements of a given quantity are uncorrelated, and calculate a weighted average and error as

$$\overline{x} \pm \delta\overline{x} = \frac{\sum_i w_i\, x_i}{\sum_i w_i} \pm \left(\sum_i w_i\right)^{-1/2} \ , \tag{1}$$

where

$$w_i = 1/(\delta x_i)^2 \ .$$

Here x_i and δx_i are the value and error reported by the ith experiment, and the sums run over the N experiments. We then calculate $\chi^2 = \sum w_i(\overline{x} - x_i)^2$ and compare it with $N-1$, which is the expectation value of χ^2 if the measurements are from a Gaussian distribution.

If $\chi^2/(N-1)$ is less than or equal to 1, and there are no known problems with the data, we accept the results.

If $\chi^2/(N-1)$ is very large, we may choose not to use the average at all. Alternatively, we may quote the calculated average, but then make an educated guess of the error, a conservative estimate designed to take into account known problems with the data.

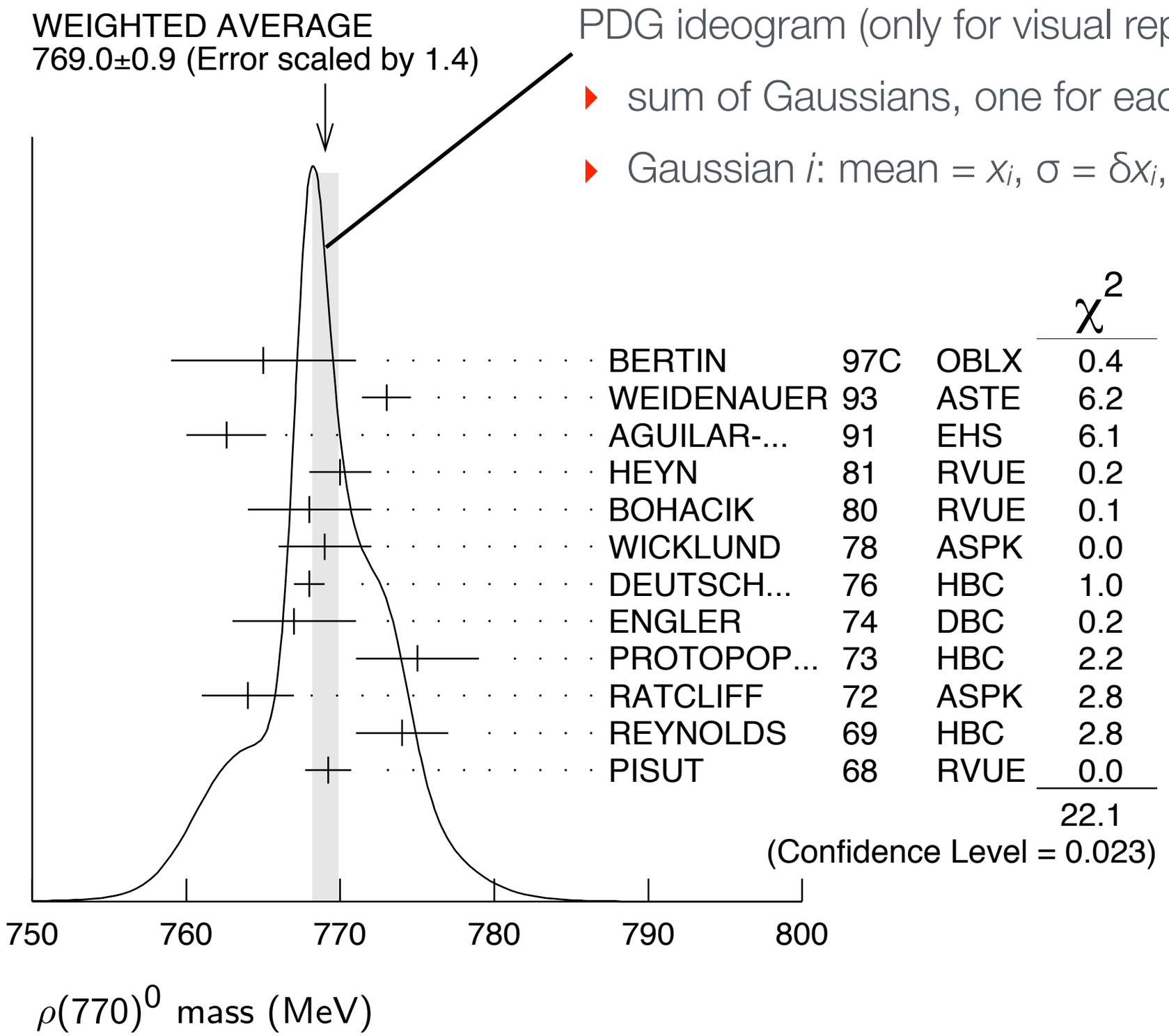
Finally, if $\chi^2/(N-1)$ is greater than 1, but not greatly so, we still average the data, but then also do the following:

(a) We increase our quoted error, $\delta\overline{x}$ in Eq. (1), by a scale factor S defined as

$$S = \left[\chi^2/(N-1)\right]^{1/2} \ . \tag{2}$$

Our reasoning is as follows. The large value of the χ^2 is likely to be due to underestimation of errors in at least one of the experiments. Not knowing which of the errors are underestimated, we assume they are all underestimated by the same factor S. If we scale up all the input errors by this factor, the χ^2 becomes $N-1$, and of course the output error $\delta\overline{x}$ scales up by the same factor. See Ref. 3.

PDG Averaging Procedure (III)



PDG ideogram (only for visual representation):

- sum of Gaussians, one for each measurement
- Gaussian i: mean = x_i, $\sigma = \delta x_i$, area $\propto 1/\delta x_i$

Another Approach To Least Squares Fits in Case of Correlated Systematic Uncertainties

Correlated systematic uncertainties can be taken into account with generalized χ^2:

$$\chi^2(\vec{\theta}) = (\vec{y} - \vec{f}(\vec{x};\vec{\theta}))^T V^{-1} (\vec{y} - \vec{f}(\vec{x};\vec{\theta})), \qquad V = \underbrace{V_{\text{stat}}}_{\text{diagonal}} + V_{\text{sys}}$$

Another approach (sometime called 'pull method'):

$$\chi^2 = \sum_{i=1}^{n} \frac{(y_i + \varepsilon\sigma_{i,\text{sys}} - f(x_i;\vec{\theta}))^2}{\sigma^2_{i,\text{stat}}} + \varepsilon^2$$

penalty term
("ε = systematic deviation in units of the standard deviation")

The pull method puts nuisance parameters on the same footing as other parameters. The penalty term is none other than a frequentist version of the Bayesian prior on the nuisance parameter.

Summary: Maximum Likelihood and χ^2 Method

Maximum likelihood method:

$$L(\vec{\theta}) = \prod_{i=1}^{n} f(x_i; \vec{\theta}) \qquad \frac{\partial \ln L}{\partial \theta_i} = 0, \quad i = 1, \ldots, m \quad \rightsquigarrow \quad \widehat{\vec{\theta}}$$

$$U[\widehat{\vec{\theta}}] = -H^{-1}, \; h_{ij} = \left. \frac{\partial^2 \ln L}{\partial \theta_i \partial \theta_j} \right|_{\widehat{\vec{\theta}}}, \qquad H = (h_{ij}), \quad U = (u_{ij}), \quad u_{ij} = \mathrm{cov}[\hat{\theta}_i, \hat{\theta}_j]$$

$U[\widehat{\vec{\theta}}]$: covariance matrix of the estimated parameters θ_i

Least-squares method:

$$\chi^2(\vec{\theta}) = -2 \ln L(\vec{\theta}) + \text{constant} \overset{\text{no correlations btw. the } y_i}{=} \sum_{i=1}^{n} \frac{(y_i - \mu(x_i; \vec{\theta}))^2}{\sigma_i^2}$$

in case of correlations

$$\chi^2(\vec{\theta}) = (\vec{y} - \vec{\mu}(\theta))^{\mathsf{T}} V^{-1} (\vec{y} - \vec{\mu}(\theta)), \qquad V = (v_{ij}), \quad v_{ij} = \mathrm{cov}[y_i, y_j]$$

$$\frac{\partial \chi^2}{\partial \theta_i} = 0, \quad i = 1, \ldots, m \quad \rightsquigarrow \quad \widehat{\vec{\theta}}$$

$U[\widehat{\vec{\theta}}]$: covariance matrix of the θ_i

$$U[\widehat{\vec{\theta}}] = 2H^{-1}, \; h_{ij} = \left. \frac{\partial^2 \chi^2}{\partial \theta_i \partial \theta_j} \right|_{\widehat{\vec{\theta}}}$$

Discussion of Fit Methods

- Unbinned maximum likelihood fit
  - \+ Don't need to bin data (no loss of information)
  - \+ Works with multi-dimensional data
  - \+ No Gaussian assumption
  - – No direct goodness of fit estimate
  - – Can be computationally expensive
  - – Can't plot directly with data
- Least-squares fit
  - \+ fast, robust, easy
  - \+ goodness of fit
  - \+ can plot with data
  - \+ works fine at high statistics
  - – data should be Gaussian

RooFit

- Toolkit for modeling distribution of events in a physics analysis
  - PDFs, composite data models
  - Unbinned maximum likelihood fits
  - Generation of "toy Monte Carlo" samples … and much more
- Originally developed for the BaBar collaboration (SLAC)
- Integrated with and built upon ROOT
- Links
  - http://roofit.sourceforge.net/
  - https://root.cern.ch/roofit-20-minutes
- Slides: http://roofit.sourceforge.net/docs/tutorial/intro/roofit_tutorial_intro.pdf
- Documentation
  - Manual: http://root.cern.ch/download/doc/RooFit_Users_Manual_2.91-33.pdf
  - Quick start: https://root.cern.ch/download/doc/roofit_quickstart_3.00.pdf
- Tutorial macros: `$ROOTSYS/tutorials/roofit`
  - also here: https://root.cern.ch/root/html/tutorials/roofit/index.html

RooFit – Core Design

Mathematical concept		RooFit class
variable	x	**RooRealVar**
function	$f(x)$	**RooAbsReal**
PDF	$f(x)$	**RooAbsPdf**
space point	$\vec{x}$	**RooArgSet**
integral	$\int_{x_{\min}}^{x_{\max}} f(x)dx$	**RooRealIntegral**
list of space points		**RooAbsData**

RooFit – Maximum Likelihood Fit Example (I)

```
void roofit_maximum_likelihood_example() {

    // --- Observable ---
    RooRealVar mes("mes", "m_{ES} (GeV)", 5.20, 5.30);

    // --- Build Gaussian signal PDF ---
    RooRealVar sigmean("sigmean", "B^{#pm} mass", 5.28, 5.20, 5.30);
    RooRealVar sigwidth("sigwidth", "B^{#pm} width", 0.0027, 0.001, 1.);
    RooGaussian gauss("gauss", "gaussian PDF", mes, sigmean, sigwidth);

    // --- Build Argus background PDF ---
    RooRealVar argpar("argpar", "argus shape parameter", -20.0, -100., -1.);
    RooArgusBG argus("argus", "Argus PDF", mes, RooConst(5.291), argpar);

    // --- Construct signal+background PDF ---
    RooRealVar nsig("nsig", "#signal events", 200, 0., 10000);
    RooRealVar nbkg("nbkg", "#background events", 800, 0., 10000);
    RooAddPdf sum("sum", "g+a", RooArgList(gauss, argus), RooArgList(nsig, nbkg));

    // --- Generate a toyMC sample from composite PDF ---
    RooDataSet *data = sum.generate(mes, 2000);

    // --- Perform extended ML fit of composite PDF to toy data ---
    sum.fitTo(*data, Extended());
```

RooFit – Maximum Likelihood Fit Example (II)

```
    // --- Plot toy data and composite PDF overlaid ---
    RooPlot *mesframe = mes.frame();
    data->plotOn(mesframe);
    sum.plotOn(mesframe);
    sum.plotOn(mesframe, Components(argus), LineStyle(kDashed));
    mesframe->Draw();

}
```