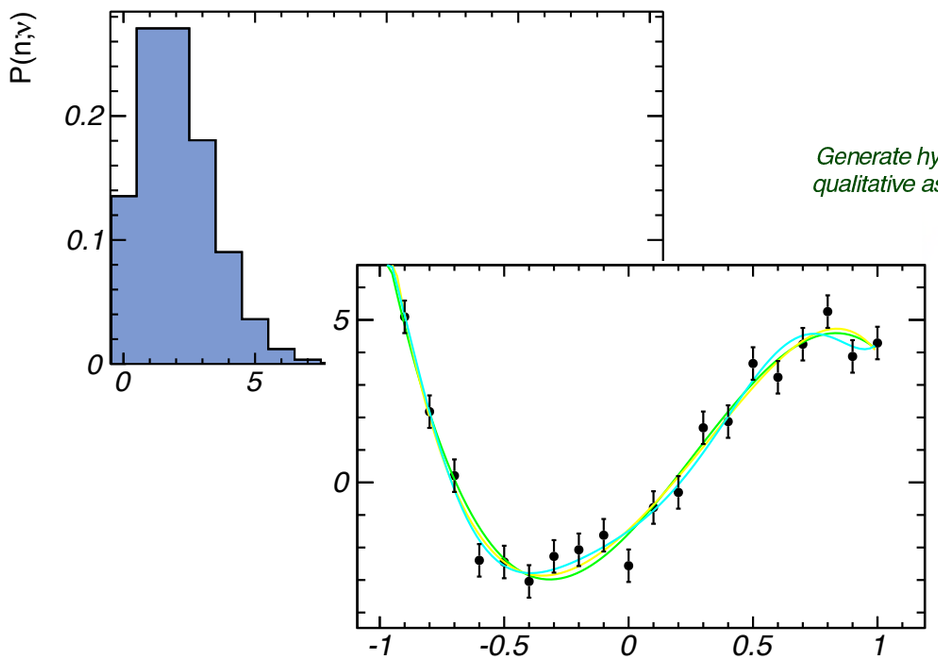
Karlsruher Institut für Technologie

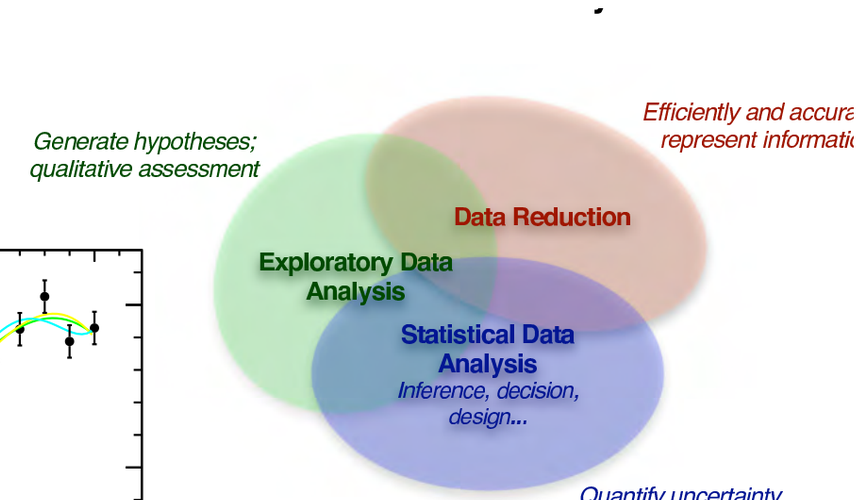
Rechnernutzung in der Physik

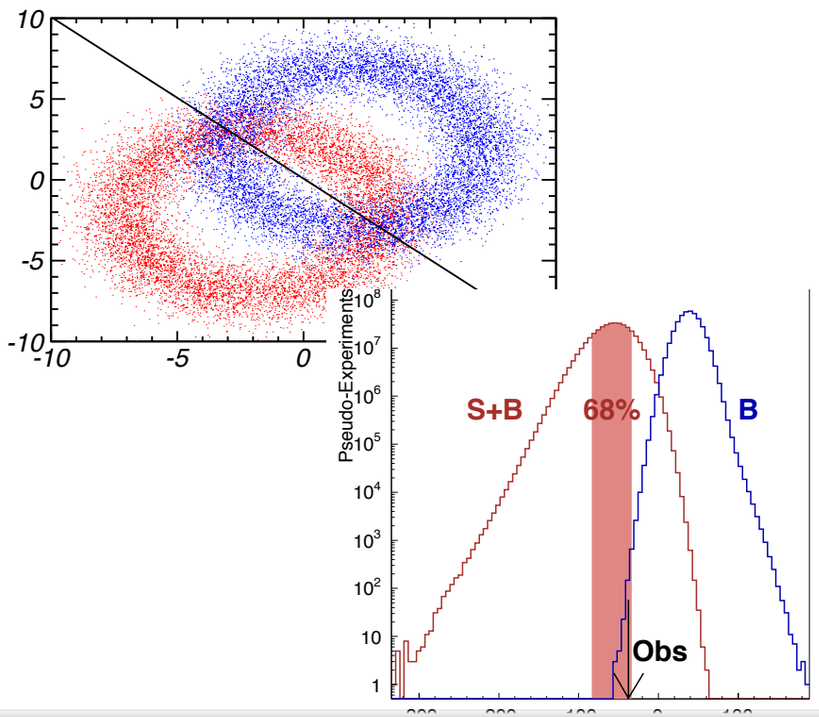
Teil 3 – Statistische Methoden der Datenanalyse

Karlsruher Institut für Technologie
Wintersemester 2012/2013

Ulrich Husemann
Institut für Experimentelle Kernphysik, Karlsruher Institut für Technologie







QISPOS-Anmeldung

- Rechnernutzung jetzt bei QISPOS freigeschaltet
  → bitte anmelden
  - Name: Rechnernutzung
  - Prüfungsnummer: 172
  - Anmeldebeginn: 15.10.12
  - Anmeldeende: 07.02.13
  - Rücktrittsende: 07.02.13
  - Prüfungsdatum: 08.02.13

Kurze Wiederholung

- Monte-Carlo-Methoden
  - Methoden zur Erzeugung von Zufallszahlen mit beliebiger Verteilung: Transformationsmethode, Verwerfungsmethode, Majorantenmethode
  - Breite Anwendung: numerische Mathematik (z. B. Integration), angewandte Statistik, Simulation stochastischer Prozesse (z. B. Quantenmechanik)
  - Genauigkeit: Vorteile bei hochdimensionalen Problemen
- Parameterschätzung
  - Arbeit mit Stichproben einer Grundgesamtheit
  - Gesucht: Schätzfunktionen mit „günstigen” Eigenschaften (Erwartungstreue, Effizienz, …)
  - Systematische Methoden: Maximum Likelihood (ML) und Kleinste Quadrate (least squares, LS)

LS und ML: Erster Vergleich

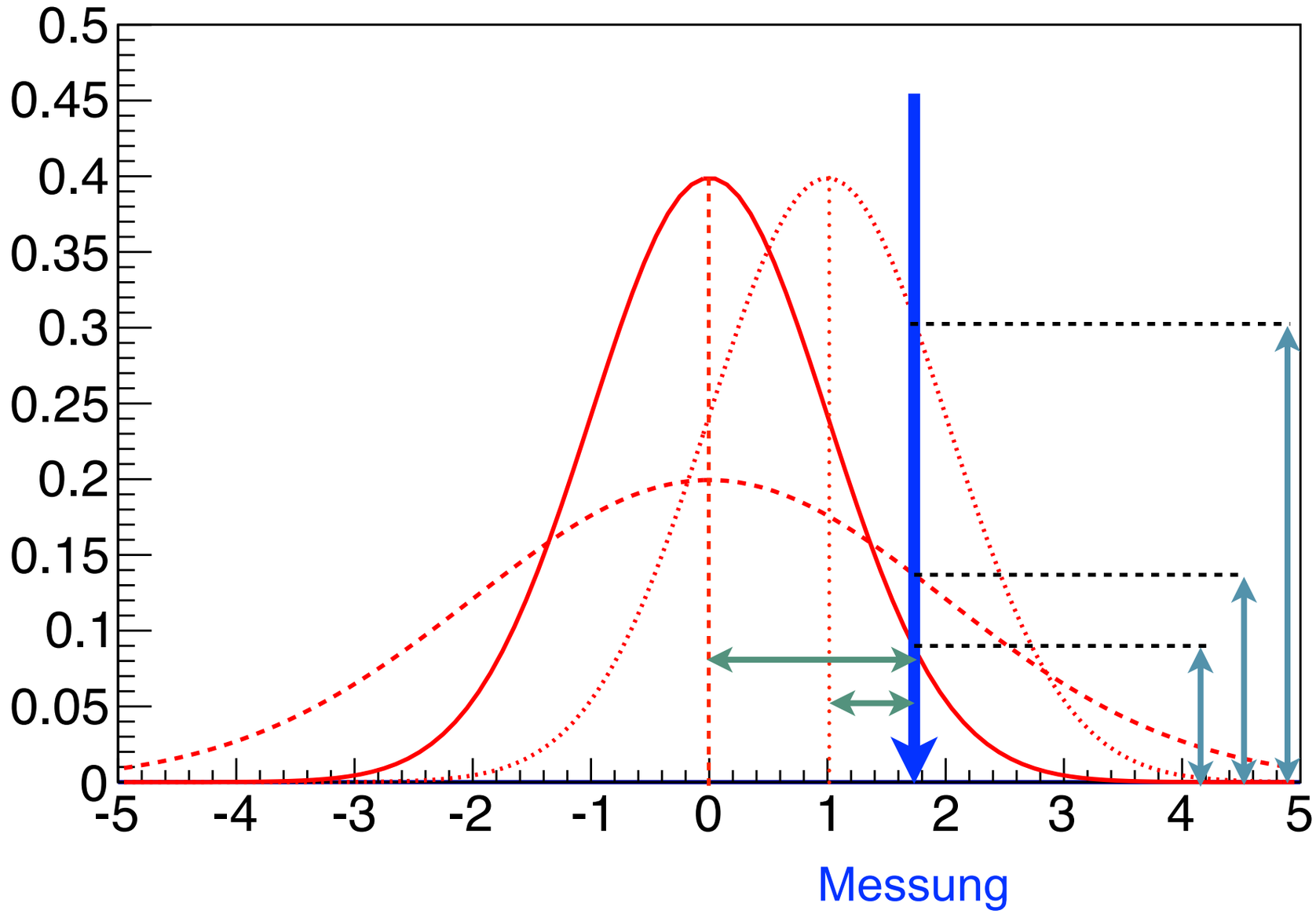
- Ausgangsfrage: welches Modell passt am besten zu den Daten?

Maximum Likelihood:
maximiere Höhe der PDF

(PDF muss bekannt sein)

Kleinste Quadrate:
minimiere Abstand vom Erwartungswert

(nur Erwartungswert bekannt)



Maximum-Likelihood-Prinzip

- Der Maximum-Likelihood-Schätzer für einen Parameter θ ist derjenige Wert von θ, für den die Likelihood-Funktion

$$L(\theta) = \prod_{i=1}^{n} f(x_i, \theta)$$

  den größten Wert annimmt: $L(\hat{\theta}) \geq L(\theta) \quad \forall\theta$

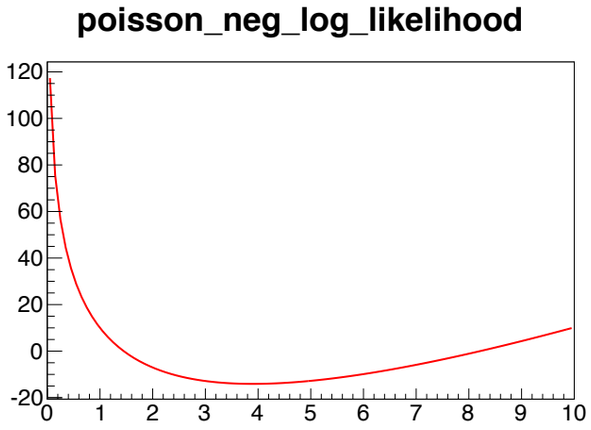
  - **x** = (x_1, …, x_n) Stichprobe der Größe n
  - $f(x_i;\theta)$: bekannte PDF der x_i, abhängig von Modellparameter θ

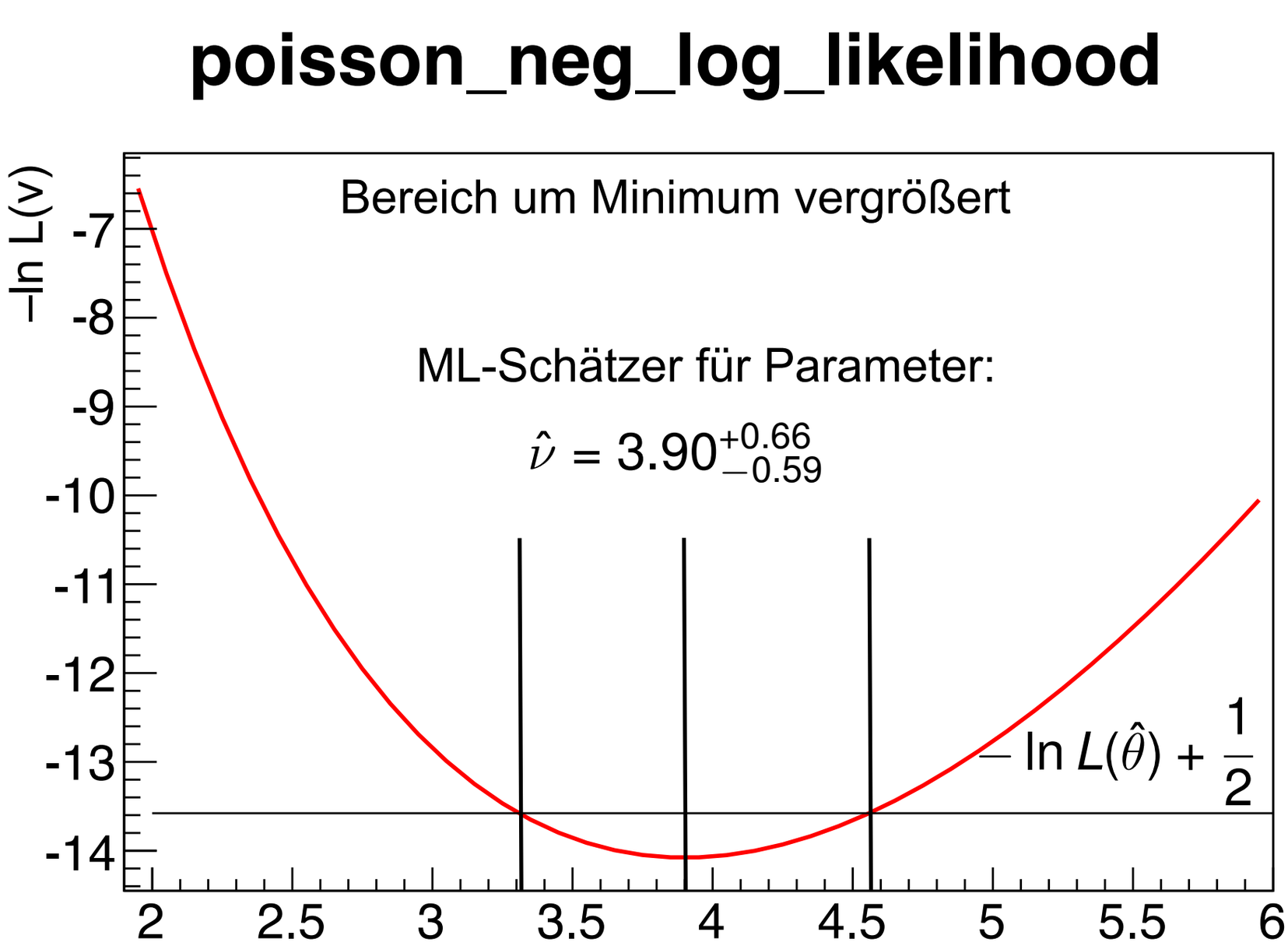
- Praxis: –ln L(θ) wird minimiert
  - Streng monotone Funktion mit denselben Extrema
  - Aus Produkten werden Summen
  - Numerische Algorithmen zur Minimierung vorhanden, z. B. Minuit

Kapitel 3.5

Parameterschätzung

Varianz von ML-Schätzern





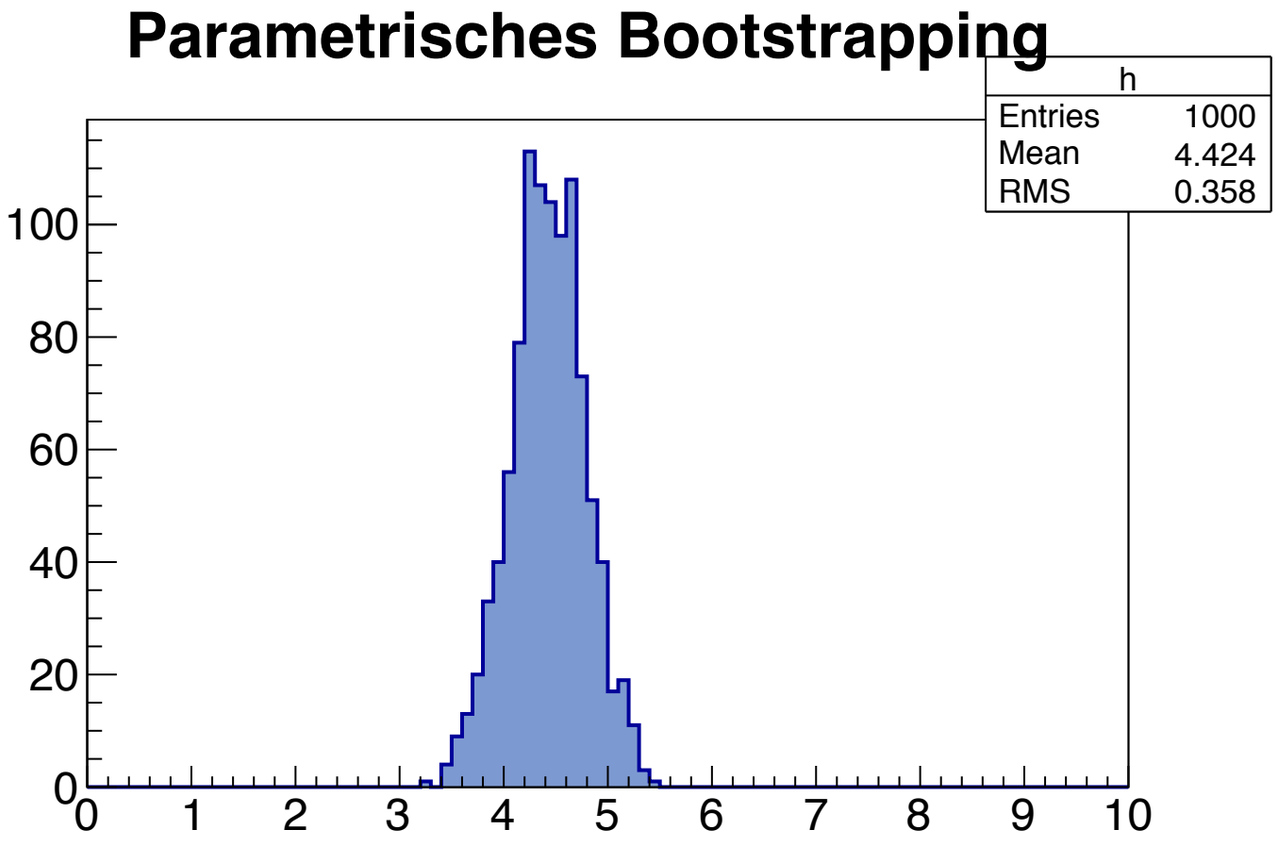
Parametrisches Bootstrapping

- Bootstrap-Verfahren (B. Efron, 1979)
  - Sammelbegriff für statistische Methoden, bei denen Schätzfunktionen durch wiederholtes „Ziehen mit Zurücklegen” aus einer Stichprobe gewonnen werden
  - Deutschsprachige Analogie: „An den eigenen Haaren aus dem Sumpf ziehen” (Baron von Münchhausen)

- Parametrisches Bootstrapping
  - Wahrer Parameter θ unbekannt, aber PDF f(x;θ) bekannt
  - Bestimmung eines Schätzwerts für θ aus Stichprobe der Größe n
  - MC-Methode (häufig: „Toy-MC”, „Ensembletest”, „Pseudoexperimente”): simuliere Stichprobe m mal, benutze dazu in f(x,θ) Schätzwert für θ (anstatt unbekanntem θ selbst)
    → Streuung der Resultate: Maß für Varianz des Schätzwerts

Parametrisches Bootstrapping

- Beispiel: Gaußverteilung mit „wahren” Werte $\mu_{wahr} = 4.5$ und $\sigma_{wahr} = 1.5$
  - Stichprobe aus n = 20
    Messungen: $\mu = 4.44$, $\sigma = 1.62$
  - Gesucht: Standardabweichung des Mittelwerts
  - Analytisches Resultat:
    $\sigma_{Mittelwert} = \sigma_{wahr}/\sqrt{n} = 0.34$
  - Bootstrapping: 1000 Pseudoexperimente mit μ und σ aus obiger Stichprobe



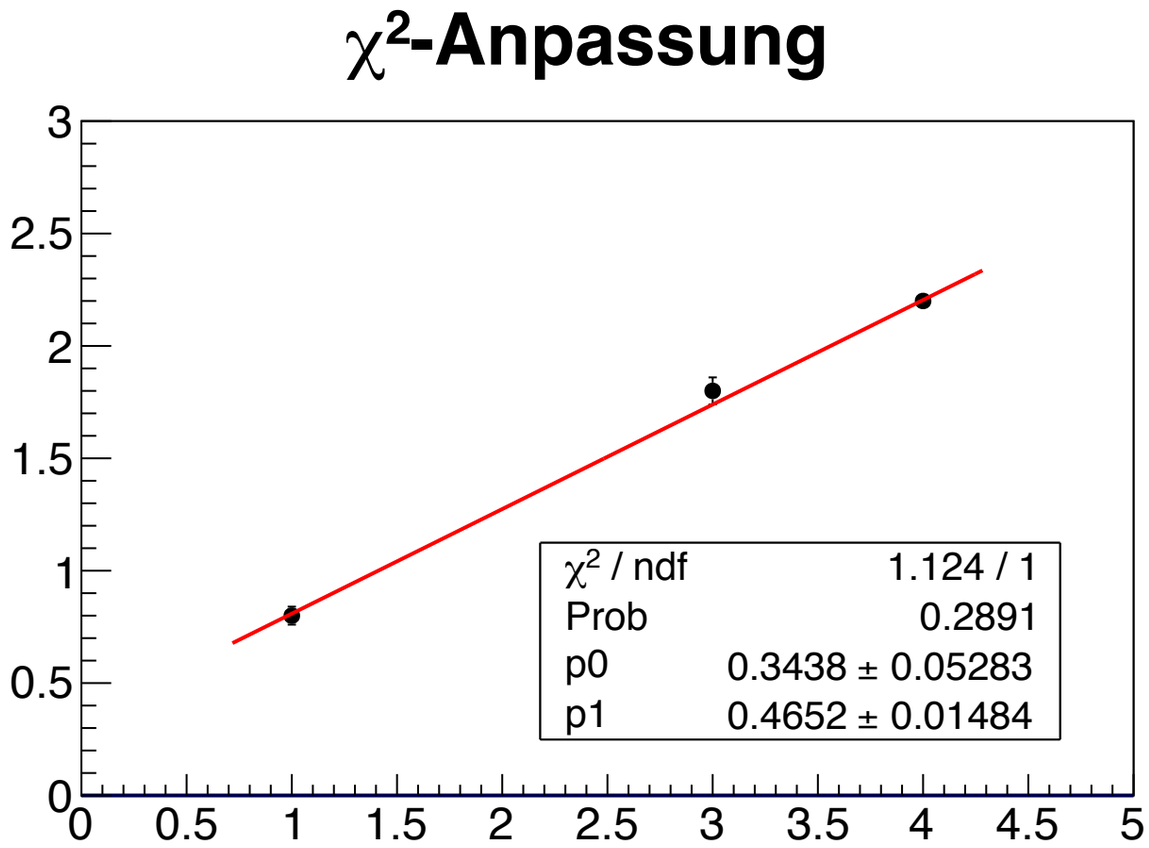
- Ergebnis: $\sigma_{Mittelwert} = 0.36$

χ^2-Anpassung

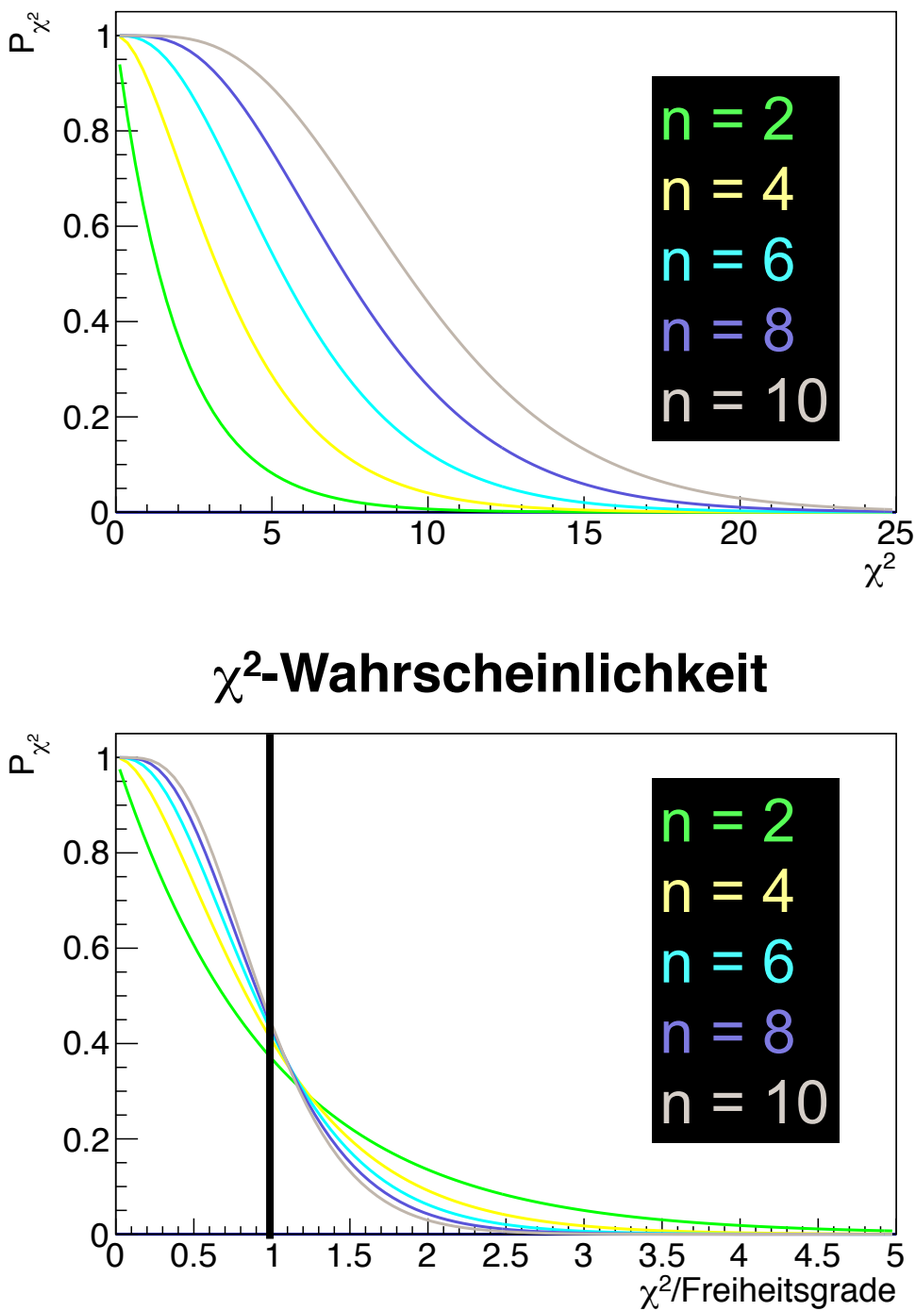
- χ^2-Anpassung in der Praxis
  - Viele Programmpakete erlauben χ^2-Anpassung (ROOT, Excel, Origin, QtiPlot, …) von Funktionen an Daten
  - Generell: Funktionalität verborgen, Benutzer müssen wissen, was sie tun!
- In ROOT:
  - Anpassung an Histogramme (TH1 usw.) und Graphen (TGraph usw.)
    → Methoden: TH1::Fit() und TGraph::Fit()
  - Einfache Anpassungsfunktionen vordefiniert (Polynome, Gauß, …)
  - Weitere Funktionen: selbst definieren (entweder als String oder Funktion mit definiertem Interface)
  - Allgemeinster Fall: χ^2-Funktion selbst vorgeben

χ^2-Anpassung in ROOT

```
void fit()
{
  const UInt_t N = 3;
  double x[N] = { 1, 3, 4 };
  double y[N] = { 0.8, 1.8, 2.2 };
  double e[N] = { 0.04, 0.06, 0.02 };

  TGraphErrors* g =
    new TGraphErrors( N, x, y, 0, e );
  g->Draw( "ap" );

  TF1* f1 = new TF1( "f1", "pol1", 0, 5 );
  TF1* f2 = new TF1( "f2", "[0]+[1]*x", 0, 5 );
  TF1* f3 = new TF1( "f3", straightline, 0, 5, 2 );

  g->Fit( "pol1" );
  g->Fit( "f1" );
  g->Fit( "f2" );
  g->Fit( "f3" );
}

double straightline( double* x, double* par )
{
  return par[0] + par[1]*x[0];
}
```

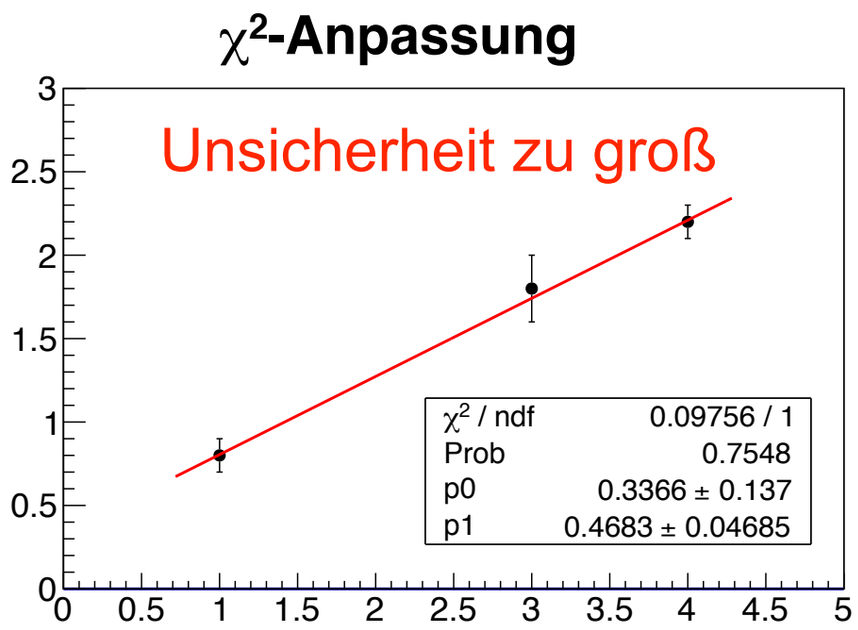


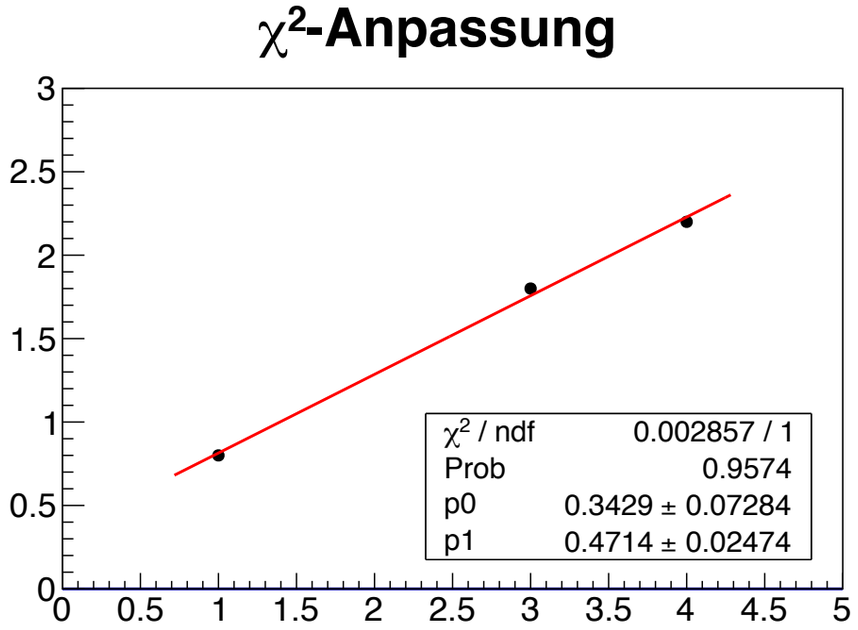
χ^2-Wahrscheinlichkeit

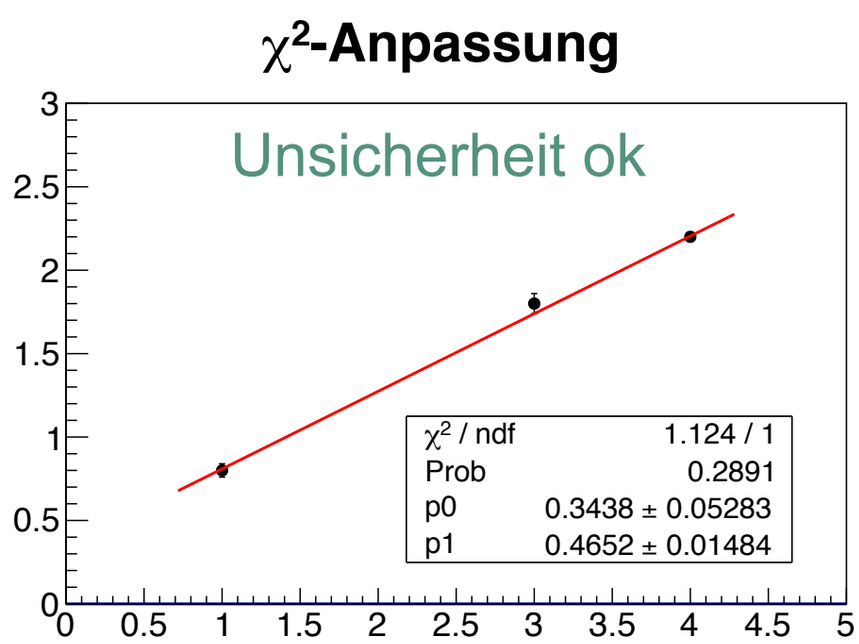


- Offene Frage: Güte der Anpassung (engl.: goodness-of-fit test)?
  - Wie gut passen Daten zu Modell?
  - Schlechte Übereinstimmung: Parameter nicht aussagekräftig
- χ^2-Wahrscheinlichkeit
  - Wahrscheinlichkeit für $\chi^2/n > 1$: ca. 40%, ungefähr unabhängig von n
  - χ^2/n Maß für Güte der Anpassung
- ROOT-Funktionen
  - ROOT::Math::chisquared_cdf_c()
  - TMath::Prob()

χ^2-Anpassung

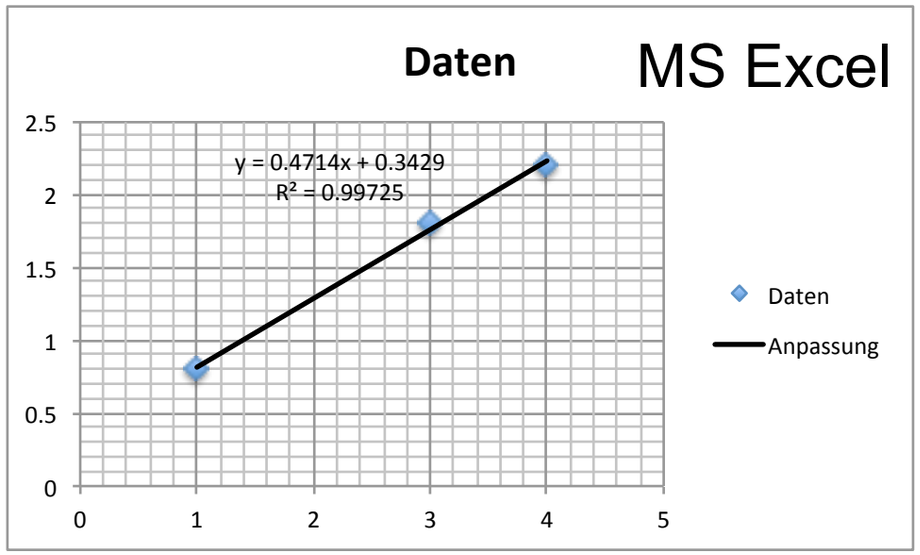






Unsicherheit 0:

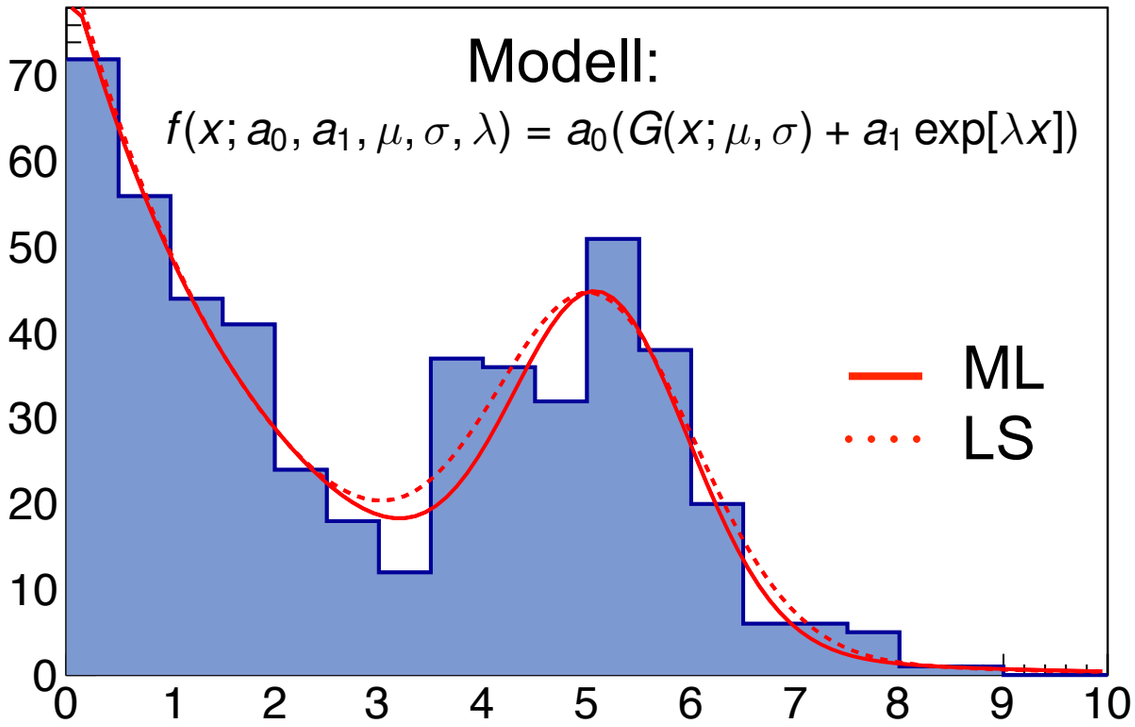
Vorsicht, Fitter macht implizit Annahmen über Unsicherheit!



LS und ML mit Histogrammen

- Anpassungsrechnung bei großen Datensätzen
  - Bisher: exakte Anpassung an jeden einzelnen Datenpunkt
  - Häufig: Datenreduktion notwendig → Histogramme
- ROOT:
  - Sowohl LS- als auch ML-Anpassung möglich
  - ML meist „besser” (besonders: Bins mit wenigen Einträgen, Integral der Verteilung)
  - Voreinstellung: LS!

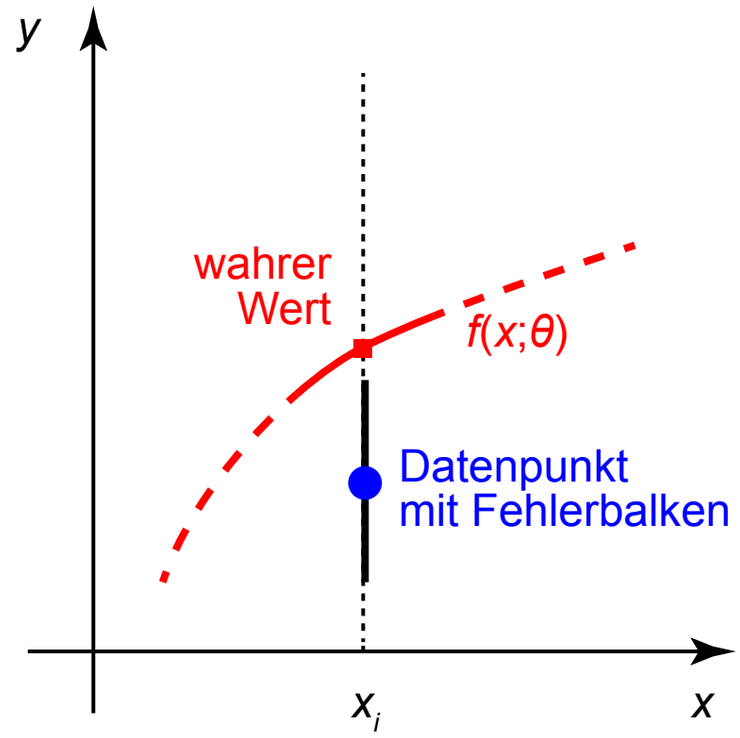
Histogrammanpassung




```
// Option E: verbesserte Fehlerschätzung
h->Fit( "ffit", "E" );

// Option L: Maximum Likelihood anstatt LS
h->Fit( "ffit", "EL+", "same" );
```

Was ist ein „Fehlerbalken”?

- Sichtweisen auf „Fehlerbalken”
  1. Fehlerbalken = Unsicherheit der Messung
  2. Datenpunkt = Teil einer Stichprobe = Zufallsvariable aus PDF $g(\hat{\theta};\theta)$ um wahren Wert (häufig: ±1σ einer Gaußverteilung)



- Größe des Fehlerbalkens aus Unsicherheit der Parameterschätzung
  - Bisher: Standardabweichung (bzw. Kovarianz) von ML- oder LS-Schätzern, ggf. nach Fehlerfortpflanzung
  - Problem: Standardabweichung ungünstig für asymmetrische Verteilungen (z. B. Exponentialverteilung: 86% Wahrscheinlichkeit in ±1σ)