Fitting counting data

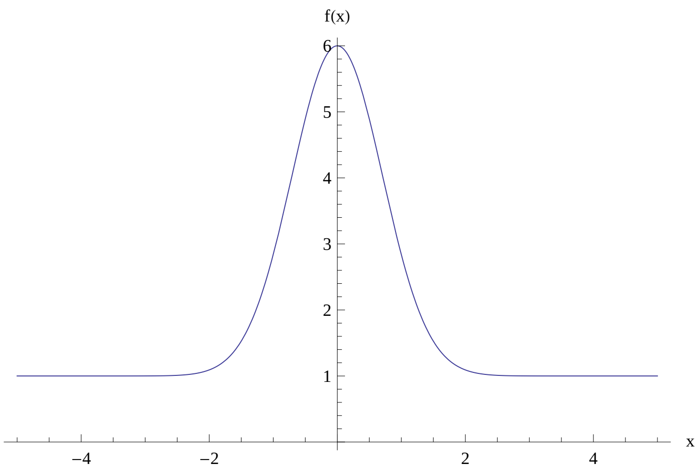
The model

Assume our counting data follows the function

In[1]:=
```
f[x_, A_, B_] := A Exp[-x^2] + B
```

In[2]:=
```
Plot[f[x, 5, 1], {x, -5, 5}, AxesLabel → {"x", "f(x)"}, AxesOrigin → {0, 0}]
```

Out[2]=



Our measurement outcome bins the data into bins of width 1/2. Bin centers are at half integer numbers.

The data

In a counting experiment, the counts in the bins are measured. The result is

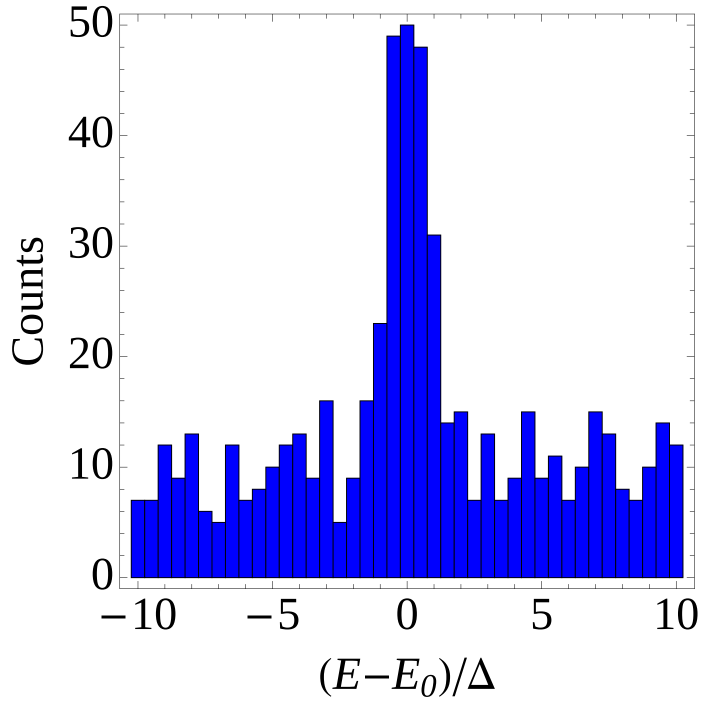
In[3]:=
```
data = Import[
  "/Users/ihn/Documents/Teaching/VP-Leitung/DataAnalysis/2014/6. Vorlesung/
    histdata.csv", "CSV"];
```

In[4]:=
```
xi = data[[All, 1]];
ci = data[[All, 2]];
```

Here we show the distribution:

In[6]:=
```
g1 = Show[Graphics[{Blue, EdgeForm[Black], Table[Rectangle[
      {xi[[n]] - 0.25, 0}, {xi[[n]] + 0.25, N[ci[[n]]]}], {n, 1, Length[xi]}]}],
  Axes → False, Frame → True, AspectRatio → 1, FrameTicksStyle → 24,
  FrameLabel → {Style["(E-E0)/Δ", 24], Style["Counts", 24]}]
```

Out[6]=



The maximum likelihood estimate

We assume a constant prior for A and B.

The -log of the posterior distribution is then, up to a constant

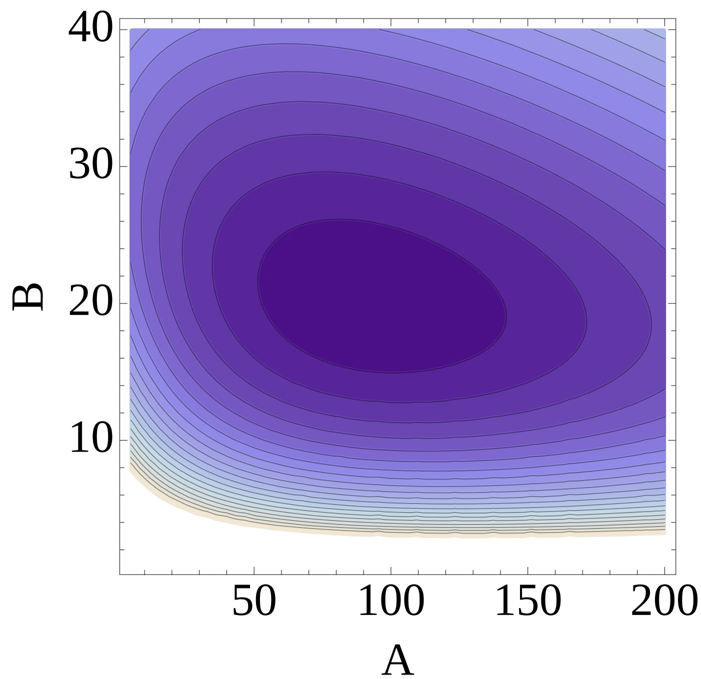
In[7]:=
```
L[A_, B_] :=
  Evaluate[- Sum[Log[PDF[PoissonDistribution[f[xi[[n]], A, B] / 2], ci[[n]]]], {n, 1, Length[xi]}]];
```

$$L[A\_, B\_] := \text{Evaluate}\left[-\sum_{n=1}^{\text{Length[xi]}} \text{Log[PDF[PoissonDistribution[f[xi[[n]], A, B] / 2], ci[[n]]]]}\right];$$

The Contour plot reveals a minimum around A=100, B=20:

In[8]:=
```
fig1 = ContourPlot[L[A, B], {A, 5, 200}, {B, 1, 40}, Contours → 20,
  FrameLabel → {Style["A", 24], Style["B", 24]}, FrameTicksStyle → 24]
```

Out[8]=



We find the minimum numerically:

In[9]:=
```
res = FindMinimum[{L[A, B], A > 0, B > 0}, {A, 100}, {B, 20}]
```

Out[9]= {105.346, {A → 91.6057, B → 20.0309}}

In[10]:=
```
Lmin = res[[1]]
```

Out[10]= 105.346

In[11]:=
```
A0 = A /. res[[2]]
```

Out[11]= 91.6057

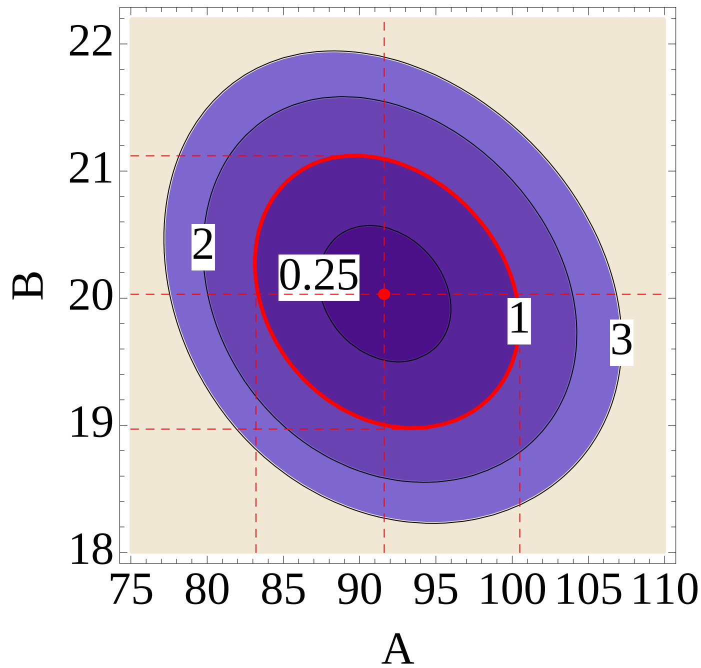
In[12]:=
```
B0 = B /. res[[2]]
```

Out[12]= 20.0309

Here is a contour plot of the posterior probability. The contour line L(A,B)=1 gives the standard errors for A and B, if projected on the coordinate axes. We determine the errors graphically.

In[13]:=
```
Show[ContourPlot[2 (L[A, B] - Lmin),
   {A, 75, 110}, {B, 18, 22.2}, Contours → {0.25, 1, 2, 3},
   FrameLabel → {Style["A", 24], Style["B", 24]}, FrameTicksStyle → 24,
   ContourLabels → (Text[Style[#3, 24], {#1, #2}, Background → White] &),
   ContourStyle → {Black, {Red, AbsoluteThickness[2]}, Black, Black}],
  Graphics[{Red, Disk[{A0, B0}, {0.4, .045}], Dashed,
    Line[{{A0, 18}, {A0, 22.2}}], Line[{{75, B0}, {110, B0}}],
    Line[{{83.2, 18}, {83.2, 20.4}}], Line[{{100.5, 18}, {100.5, 19.6}}],
    Line[{{75, 18.97}, {94, 18.97}}], Line[{{75, 21.12}, {90, 21.12}}]}]
]
```

Out[13]=



We determine the uncertainties from the graph:

In[14]:=
```
A0 - 83.2
100.5 - A0
```

Out[14]= 8.40568

Out[15]= 8.89432

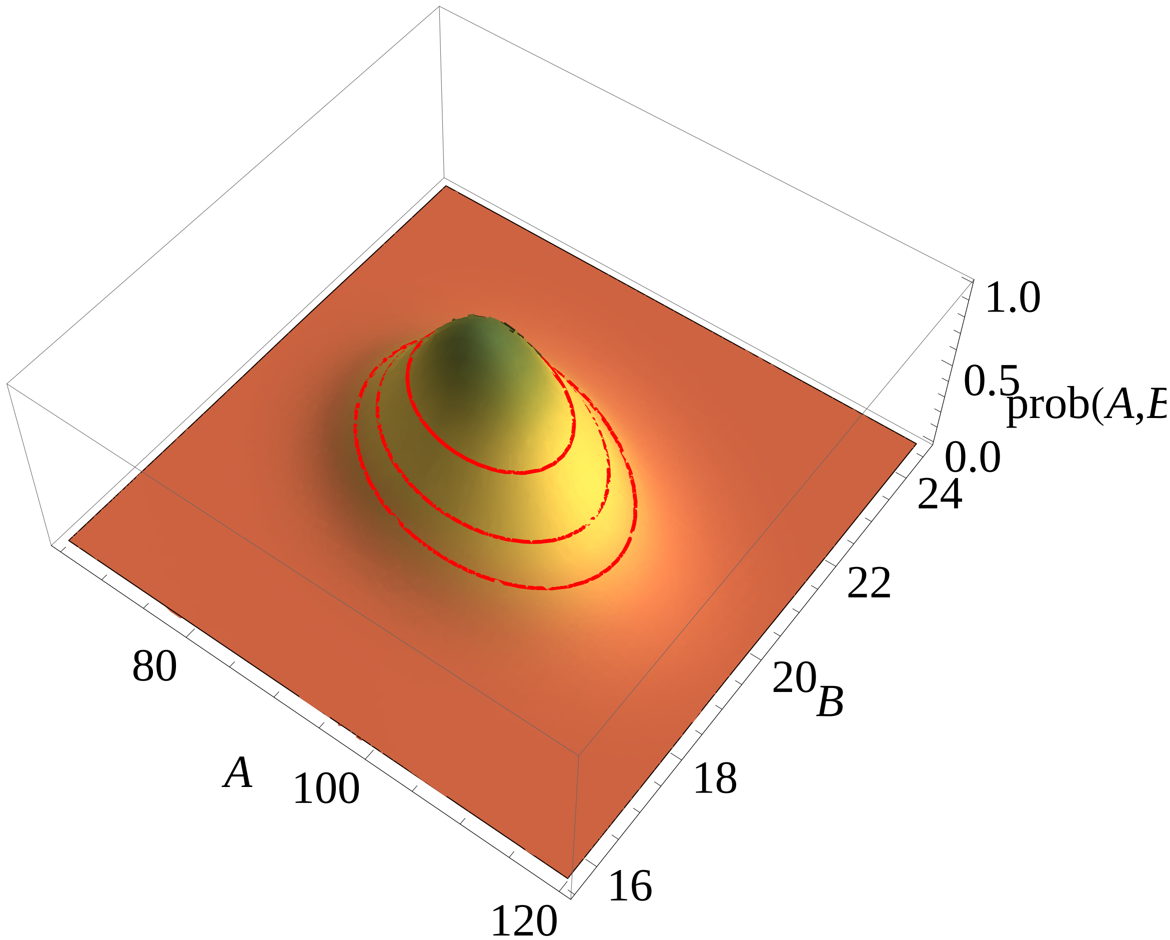
In[16]:=
```
B0 - 18.97
21.12 - B0
```

Out[16]= 1.06089

Here is a 3D-Plot of the posterior probability:

In[18]:=
```
Plot3D[Exp[-(L[A, B] - Lmin)], {A, 65, 120}, {B, 15.6, 24.6},
 AxesLabel → {Style["A", 24], Style["B", 24], Style["prob(A,B)", 24]},
 TicksStyle → 24, ColorFunction → "SandyTerrain",
 MeshFunctions -> {#3 &}, Mesh → {{Exp[-1 / 2], Exp[-1], Exp[-3 / 2]}},
 MeshStyle → {Red, AbsoluteThickness[2]}, PlotPoints → 50, PlotRange → All]
```

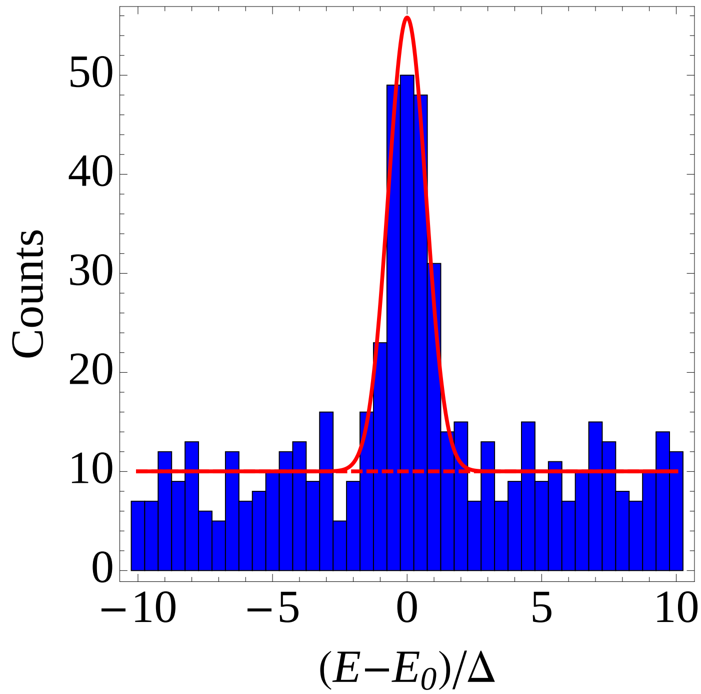
Out[18]=



Now we wish to generate a plot that shows the result of the fit with the data. First a plot of the result of the fit:

In[19]:=
```
g2 = Plot[{f[x, A0, B0] / 2, B0 / 2}, {x, -10, 10}, PlotRange → All,
   PlotStyle → {{Red, AbsoluteThickness[2]}, {Red, Dashed, AbsoluteThickness[2]}}];
Show[g1, g2, AxesLabel → {"x", "counts"}, AxesOrigin → {-11, 0}]
```

Out[20]=



Here we calculate the Hesse-Matrix:

In[21]:=
```
H = {{Evaluate[∂A ∂A (L[A, B])] /. res[[2]], Evaluate[∂A ∂B (L[A, B])] /. res[[2]]},
  {Evaluate[∂A ∂B (L[A, B])] /. res[[2]], Evaluate[∂B ∂B (L[A, B])] /. res[[2]]}}
```

Out[21]= {{0.0139854, 0.0245277}, {0.0245277, 0.911249}}

In[22]:= `H // MatrixForm`

Out[22]//MatrixForm=

$$\begin{pmatrix} 0.0139854 & 0.0245277 \\ 0.0245277 & 0.911249 \end{pmatrix}$$

Its inverse is the covariance matrix:

In[23]:= `iH = Inverse[H]`

Out[23]= {{75.0458, -2.01998}, {-2.01998, 1.15177}}

In[24]:= `iH // MatrixForm`

Out[24]//MatrixForm=

$$\begin{pmatrix} 75.0458 & -2.01998 \\ -2.01998 & 1.15177 \end{pmatrix}$$

From the elements of the covariance matrix we find the uncertainties in the parameters and the correlation coefficient:

In[25]:= $\sigma A = \sqrt{iH[[1, 1]]}$

Out[25]= 8.6629

In[26]:= $\sigma B = \sqrt{iH[[2, 2]]}$

Out[26]= 1.0732

In[27]:= ρ = iH[[1, 2]] / σA / σB

Out[27]= −0.217271