Kirchhoff-Insitut für Physik
Physikalisches Insitut

Winter semester 2013-14
KIP CIP Pool (1.401)

# Exercises for Statistical Methods in Particle Physics

http://www.physi.uni-heidelberg.de/~nberger/teaching/ws13/statistics/statistics.php

**Dr. Niklaus Berger** (nberger@physi.uni-heidelberg.de)
**Dr. Oleg Brandt** (obrandt@kip.uni-heidelberg.de)

# Exercise 9: Fits in presence of systematic uncertainties and correlated parameter estimation

**8. December 2013**
**Hand-in solutions by 14:00, 15. December 2013**

Please send your solutions to `obrandt@kip.uni-heidelberg.de` by 15.12.2013, 14:00, punctually. Make sure that you use *SMIPP:Exercise09* as subject line. If plots are requested, please include print statements to produce pdf files in your code, and provide the plots separately. Please add comments to your source code explaining the steps. Test macros and programs before sending them off...

## 1 Properties of the $\chi^2$-test and the KS-test

In this Problem, we want to study the properties of the $\chi^2$-test and of the KS-test for the cases of limited and large statistics, and in presence of systematic uncertainties. We do this using a powerful tool pseudo-experiments which is widely used in particle physics in order to evaluate the performance of a given data analysis technique. Tests using ensembles of pseudo-experiments are based on the idea that one applies the data analysis technique not to data but to simulated pseudo-data, which is generated based on MC simulations, while treating it exactly like data otherwise.

### 1.1 Pseudo-experiments with statistical uncertainties only

To start, download from the website the file `ex_9_1_input.root`. It is based on the example of Problem 8.2 from the last Problem Sheet. First, we focus on the histograms `hsig` and `hbgr`. As we have verified in 8.2 that these two histograms describe the data adequately, we construct the data *template* for our pseudo-data distribution by adding those two histograms to give `htpl`. In the next step, we generate a pseudo-experiment by generating the distribution `hpse1` by independently Poisson-fluctuating the contents of the individual bins of `htpl` and setting the bin contents of `hpse1` accordingly. Note that, depending on the version of ROOT, you may need to call the `TH1::Sumw2()` method in order for the errors on the bins of `hpse1` to be recalculated to reflect the set bin contents. This way to construct our pseudo-experiment assumes that there are no systematic uncertainties which could make our simulations behave differently from data, and any possible differences are purely statistical. Now we can use `hpse1` as pseudo-data and do our tests from 8.2 to check the hypothesis that it is modelled adequately by the sum of `hsig` and `hbgr`. Generate 1000 pseudo-experiments following this prescription and please provide the plots of the first 3 of them (`hpse`$i$`.pdf`, $i = 1, 2, 3$) as a cross check. For each of the pseudo-experiments, determine $\chi^2$/D.o.F., $\chi^2$-probability and the KS-probability, and plot their distributions (ex_8_1_1_chisq.pdf,ex_8_1_1_chipr.pdf, and ex_8_1_1_kspr.pdf). What do you expect for the mean of the former? Do your findings match the expectation? What can you say about the latter two?

We now proceed to the case where we have the same underlying distribution, but limited statistics. To counter this, the signal and background histograms are adjusted to have wider bins: hsiglo and hbgrlo. Again, generate 1000 pseudo-experiments and plot the first 3 of them (hpselo$i$.pdf, $i = 1, 2, 3$). Test the hypothesis that hpselo$i$ are described by the sum of hsiglo and hbgrlo. What do you expect now for the $\chi^2$/D.o.F. in terms of mean *and* width, compared to the above case of high statistics? Check this by plotting $\chi^2$/D.o.F. for the 1000 pseudo-experiments. Similarly, plot the $\chi^2$-probability and the KS-probability (ex_8_1_1_lochisq.pdf,ex_8_1_1_lochipr.pdf, and ex_8_1_1_lokspr.pdf).

### 1.2 Pseudo-experiments with flat systematic uncertainties

Now we consider the case that we have a systematic uncertainty which affects the yield of the background. For simplicity, we assume that this effect is equidistributed in $m_{\gamma\gamma}$. This is reflected in hbgrsys. We assume that the signal distribuion hsig is not affected. Construct again 1000 pseudo-experiments which represent how data would look like if the background simulations were affected by this systematic uncertainty, i.e. by creating the signal+background template from the sum hsig and hbgrsys. As this is a *systematic* uncertainty, we are not allowed to change the hypothesis we are testing, i.e. that our (pseudo-)data is described by the sum of hsig and hbgr (note that this is the background distribution w/o systematic uncertainties!). Like above, determine the $\chi^2$/D.o.F., $\chi^2$-probability and the KS-probability, and plot their distributions (ex_8_1_2_chisq.pdf,ex_8_1_2_chipr.pdf, and ex_8_1_2_kspr.pdf). Interpret these three plots, in particularly focusing on the $\chi^2$-probability and the KS-probability. What is the advantage of KS-probability?

The above findings can be generalised to the case of non-flat systematic uncertainties, however, qualitatively it is the same, and so will not be treated in this Problem.

(ex_8_1.C/.py)

## 2 Minmisation of correlated parameters with MINUIT

Minimisation problems of $-\ln L$ or a $\chi^2$ expression to extract physics parameters given a specific model are very common in particle physics. The underlying tool which used by ROOT to do this is the MINUIT routine, which was written back in the 70'ies, but turned out to be good enough to be still alive (albeit wrapped to be accessible in C++ or, recently, re-coded in C). For example, when fitting some datasets to some TF1 MINUIT can be used. In the following, we will learn to use MINUIT stand-alone, as it is much faster than via the TF1/TF2 interface and versatile (i.e. not limited to one or two free parameters etc). As an example, we will use the combination of two measurements, where some uncertainties are fully correlated and some are not. In such a case, the correct treatment of such correlations is important, as failure to do so can lead to biased results.

### 2.1 W mass measurement at LEP

At the LEP accelerator at CERN the mass of the W boson was measured in two different channels:

$$\begin{aligned} e^+e^- &\rightarrow W^+W^- \rightarrow q_1q_2q_3q_4 \\ e^+e^- &\rightarrow W^+W^- \rightarrow \ell\nu q_1q_2 \end{aligned}$$

The experimental signature in the detector for the first channel with four quarks are four reconstructed jets. The second kind of reaction is identified by a lepton (electron or muon) and two jets. The neutrino is not detected. The measured $W$ masses are:

$$\begin{aligned} 4 \text{ jets channel}: \quad m_W &= (80457 \pm 30 \pm 11 \pm 47 \pm 17 \pm 17) \text{ MeV} \\ \text{lepton} + 2 \text{ jets channel}: \quad m_W &= (80448 \pm 33 \pm 12 \pm \ \ 0 \pm 19 \pm 17) \text{ MeV}. \end{aligned}$$

The first two uncertainties are the statistical and systematic experimental uncertainties. They are uncorrelated. The third uncertainty is an uncertainty from theory only present in the four jets channel. The fourth uncertainty is 100% correlated because it comes from a common theoretical model. Also the last uncertainty which originates from the LEP accelerator is fully, i.e. 100% correlated between both measurements.
Construct a covariance matrix taking into account all uncertainties and their correlations. Use this covariance matrix to define a $\chi^2$ expression containing the average $W$ mass $\bar{m}_W$ as a free parameter. Determine $\bar{m}_W$ and its uncertainty by minimizing the $\chi^2$ expression with the `TMinuit` class (`ex_9_2_1.C/.py`). This can be done by e.g.:

```
TMinuit* minuit = new TMinuit(numberOfParameters) ;
// Define parameters
// identifier, name, start value, step width, bounds
minuit->DefineParameter(0, "m", 80000, 50, 0, 0) ;
// tell Minuit which function to use
minuit->SetFCN(&FCN) ;
// run minimisation and error calculation
minuit->Migrad() ;
// get fitted parameters and error
minuit->GetParameter(0, par, sigma) ;
```

Note that the function to be minimised, `FCN`, has to follow the following prototype:

```
void fcn(Int_t& npar, Double_t* gin, Double_t& f, Double_t* par, Int_t iflag);
```

This is the meaning of the variables:

| | |
|---|---|
| `npar` | number of free parameters involved in minimisation |
| `gin` | computed gradient values (optional) |
| `f` | function to be minimised |
| `par` | vector of constant and variable parameters |
| `flag` | to switch between several actions of FCN |

For further reference about MINUIT, the `TMinuit` class description on the ROOT website is a good idea for first-pass information, details can be found in the CERNLIB documentation online:
`wwwinfo.cern.ch/asdoc/minuit/minmain.html`.

## 2.2 Analytical solution

Because the minimization of the $\chi^2$ expression in Problem 9.2.1 is a linear problem, it can be solved analytically. Determine $\bar{m}_W$ and its error analytically and compare them to the result of Problem 9.2.1.

## 2.3 Calculate the individual uncertainties for combination

Estimate the contributions from statistical, systematic, theoretical and accelerator-based uncertainties to the uncertainty of the combined $W$ mass measurement. Use the quadratic difference between the total uncertainty and the uncertainty calculated with a covariance matrix where one component is removed.
(`ex_9_2_3.C/.py`)