4-й семестр, lesson5
Probability distribution functions and minimisation

- Detailed descriptions can be found in
- http://pdg.lbl.gov/2015/reviews/rpp2015-rev-probability.pdf
- http://pdg.lbl.gov/2015/reviews/rpp2015-rev-statistics.pdf
- Let *x* be a possible outcome from an observation.
- We define $f(x;\vartheta)$ as the probability that the measurement's outcome lies between *x* and *x+dx*. The function $f(x;\vartheta)$ is called the *probability density function* (p.d.f.) , which may depend on one or more parameters ϑ. If *x* can take only discrete values (non-negatve integers) , then we use $f(x;\vartheta)$ to denote the probability to find the value *x*. The term p.d.f. Is taken to cover both the continuous and discrete cases, also technically the term density should only be used in the continuous case.
- The p.d.f. Is always normalized to unity. Both *x* and ϑ may have multiple components and are then often written as vectors.
- If ϑ is unknown, we may estimate its value from a given set of measurements of x, this is a central topic of *statistics*.

Moments of p.d.f.

- The cumulative distribution function *F(a)* is the probability that $x \leq a$:
- $F(a) = \int_{-\infty}^{a} f(x)dx;$
- If *x is discrete-valued, the integral is replaced by a sum.*
- The n^{th} moment of the r andom variable x is
- $a_n = \int_{-\infty}^{\infty} x^n f(x)dx;$
- And the n^{th} central moment of the x (or moment about the mean, a_1 is
- $m_n = \int_{-\infty}^{\infty} (x - a_1)^n f(x)dx;$
- The most commonly used moments are the mean μ and variance σ^2 : $\mu \equiv a_1;$
- $\sigma^2 \equiv m_2 = a_2 - \mu^2$.
- It is often convenient to use the standatd deviation of x, σ, devined as the square root of the variance.

Binomal distribution

- A random process with exactly two possible outcomes which occur with fixed probabilities is called a *Bernoully process*. If the probability of obtaining a certain outcome (a «success») in an individual trial is *p*, then the probability of obtaining exactly *r* successes (*r=0,1,2,…,N*) in *N* independent trials, without regards to the order of the successes and failures, is given by binomial distribution f(r:N.p)
- $f(r{:}\,N.p) = \frac{N!}{r!\,(N-r)!}\,p^r q^{N-r}$
- $r = 0.1.2\ldots, N;\ \ 0 \le p \le 1;\ \ q = 1\ - p;$
- $mean = Np;\ \ variance = \ \ Npq.$

Poisson distribution

- The Poisson distribution *f(n;ν)* gives the probability of finding exactly *n* events in a given interval of *x* (space or time) when the events occur independently of one another and of *x* at an average rate of *ν* per the given interval:
- $$f(n;\nu) = \frac{\nu^n e^{-\nu}}{n!};\ \ n = 0,1,2,\ldots;\ \nu > 0;$$
- $$Mean = \ \nu;\ variance = \ \nu;$$
- It is the limiting case $p \to 0, N \to \infty, Np = \ \nu$ of the binomial distribution. The Poisson distribution approaches the Gaussian distribution at large ν.

Normal or Gaussian distribution

- The Gaussian probability density function
- $f(x;\mu,\sigma^2) = \frac{1}{\sigma\sqrt{2\pi}}\exp(-(x-\mu)^2/2\sigma^2)$
- $-\infty < x < \infty,\ -\infty < \mu < \infty;\ \sigma > 0.$
- The Gaussian derives its importance in large part from *central limit theorem*:
- If independent random variables $x_1 \ldots x_n$ are distributrf according to any p.d.f. with finit mean and variance, then the sum $y = \sum_{i=0}^{n} x_i$ will have a p.d.f. that approaches a Gaussian at large n.
- The sum of a large number of fluctuations x_i will be distributed as a Gaussian, even if the x_i themselfs are not.

X^2 distribution

- If $x_1 \ldots x_n$ are independent Gaussian random variables, the sum $z = \sum_{i=1}^{n} (x_i - \mu)^2 / \sigma_i^2$ follows the χ^2 p.d.f. *with n degrees of freedom.*
- For large n, the χ^2 p.d.f. approaches a Gaussian with a mean μ=n and a variance σ^2 = 2n.
- The χ^2 p.d.f. is often used in evaluating the level of compatibility between observed data and a hypothesis for the p.d.f. that the data might follow.
- This method works well for 1-dimensional distributions if it is possible to subdivide the overall X-interval into many bins with many events in every bin and if the non-linearity of the p.d.f in all bins is small. The information of position of individual events in the bin is lost.
- If the requirements formulated above are not satisfied, then another method is used (so called Likelihood fit).

Introduction to MINUIT.

- Detailed description in MINUIT User's Guide is available at
- https://www.researchgate.net/publication/251575389_MINUIT_User's_Guide
- https://en.wikipedia.org/wiki/MINUIT
- MINUIT, now MINUIT2, is a numerical minimization computer program originally written in the FORTRAN programming language by CERN staff physicist Fred James in the 1970s. The program searches for minima in a user-defined function with respect to one or more parameters using several different methods as specified by the user. The original FORTRAN code was later ported to C++ by the ROOT project; both the FORTRAN and C++ versions are in use today. The program is very widely used in particle physics, and hundreds of published papers cite use of MINUIT.[2] In the early 2000s Fred James started a project to implement MINUIT in C++ using object-oriented programming. The new MINUIT is an optional package (minuit2) in the ROOT release. As of October 2014 the latest version is 5.34.14, released on 24 January 2014.[

Introduction to MINUIT

- MINUIT searches a minimum of functional which depends on several variable parameters (maximum 50). User provides a code in so called FCN function, which calculates values in several measured points according to the tested parametrization function and the current values of variable parameters. There are two methods, the χ^2 fit and the Minimal Likelihood fit. In the χ^2 fit, the measured values and errors of values are taken in several bins, the theoretical values are calculated with carrent values of variable parameters, and the χ^2 value is calculated. The FCN funciion is called many times, with various values of parameters.

User-defined parameters

- Apart from the code in FCN function, used must provide initial values of all parameters and specify the minimum and maximum allowed values for all parameters. User can fix one or several parameters. Remaining parameters are variable. At the beginning of work, MINUIT calls FCN twice with indicated parameter set, and verify that the FCN calculates the same value of functional.
- There are several methods of minimization: SEEK, SIMPLEX and MIGRAD. User indicates the sequence of fitting algorythms.
- The SEEK procedure makes a random variation of variable parameters within allowed intervals, calculates the FCN and selects the parameter set with the minimal FCN value. Number of calls is undicated by user. The SEEK is usually used at the 1st stage, when the coordinates of global minimum in the parameter space is not known.

SIMPLEX and MIGRAD

- SIMPLEX algorythm starts from initial point (possibly the SEEK result) and then searches minima by variation of individual parameters. Then the worst point is rejected, and SIMPLEX calculates position of new point wor the next trial. This method is used usualy for preparation of starting point for MIGRAD. SIMPLEX do not calculate the derivations of FCN by the parameters, and therefore it is less sensitive to the precision of calculation. SIMPLEX don't calculate the fitted errors of parameters.
- MIGRAD is the main algorythm, it calculates derivatives of FCN by variable parameters. User can calculate the derivatives in FCN analytically. If this is not done, then MIGRAD calculates the derivatives numerically.

MIGRAD

- Starting from the current point in the parameters space, the derivations permit to determine the direction to the minimum and a step is done in this direction. The procedure is repeated at the next iteration, and so on – untill the point with zero derivation is found. Then the 2-nd derivation in this point is calculated in order to be sure that there is a minimum indeed.
- The matrix of 2-nd derivatives should be positively definite.
- Then the matrix of fitted error is calculated (it corresponds to a contour where the FCN value increases by one in comparison with the minimal value.
- There is a possibility to search for another minimum, starting from a randomly chosen point.

MINOS procedure

- If the MIGRAD found a satisfactory minimum, it is possible to submit a supplementary procedure for calculation of fitted errors, MINOS. MINOS tries to change the fitted value for a certain parameter by a value close to the fitted error estimated in MIGRAD, then this parameter is temporarily fixed and the minimum from the fit of other parameters is searched. There is an iterative procedure, which searches the positive and negative variation at which the FCN value increases by one incomparison with initial minimum from MIGRAD. It is possible that the values of positive and negative errors are different.
- It happens quite often that a new minimum found during the work of MINOS, which is less than the initial minimum from MIGRAD. In this case the program goes back to minimisation stage.

Implimentation in ROOT

- Useful description of the ROOT system can be found in http://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuideA4.pdf
- Or http://root.cern.ch/root/htmldoc/guides/users-guide/ROOTUsersGuide.html
- In order to fit a data sets we need a model to describe our data, e.g. a probability density function describing our observed data. Let's start from a simple case: χ^2 fit of 1-dimensional histograms.
- Access to data and errors in individual bins from histogram named «hist»:
- double data = hist->GetBinContent(Ibin);
- double err = hist->GetBinError(Ibin);

Preparation for simple fit

- Notice that by default the error is taken as a Square Root from the bin content. This is not correct if we work with a histogram which is produce as a difference of two histograms. To get correct operetion with errors, one need to activate the option hist->Sumw2(); after histogram creation (before filling).
- If user has arrays with measured values <val> and errors <err>, it is possible to fill a histogram with those values with commands
- Hist->SetBinContent(J,val(J)); SetBinError(J,err(J)).
- Then user should define the fit function and the fitting range.
- For simple cases, it is possible to use predefiled finctions, like «pol1> for linear function (2 parameters), <pol2> for quadratic (3 parameters), «gaus» for Gaussian function (3 parameters, Normalization, mean and σ).

Resonances in ($\pi^+ \pi^- \pi^{0)}$ system

- Exercises:
- 1) set link for input Ntuple: ln -s /nfs/lfi.mipt.su/data/nikola/ves/run42/ntbeam_cher_r17_1_v15.root ntbeam_cher_r17_1_v15.root
- 2) copy file scp –p /nfs/lfi.mipt.su/data/nikola/ves/run42/example_5_edited.C . Look for this file and run it in root : .X example_5_edited.C ; histogram h_002 is created and wriyyen to output file ntuple.root
- Try to fit the distribution in h_002 by a Gauss function in mass range (0.65, 0.90) GeV with file from /nfs/lfi.mipt.su/data/nikola/ves/run42/example_6_edited.C
- needed edition: please set the starting values for 3 parameters in the Gauss

Exercises cont.

- Please compare the fitted curve with histogram.
- 3) In order to improve agreement between data and fit result, let's include a Background term (linear function). An example available in /nfs/lfi.mipt.su/data/nikola/ves/run42/example_7_edited.C
- Again, needed a starting point for 3+2 parameters (which might be very approximate)
- Run the fit and look for result. Sttill the χ^2 is bad, but the agreement between the data and the fit result is significantly better.