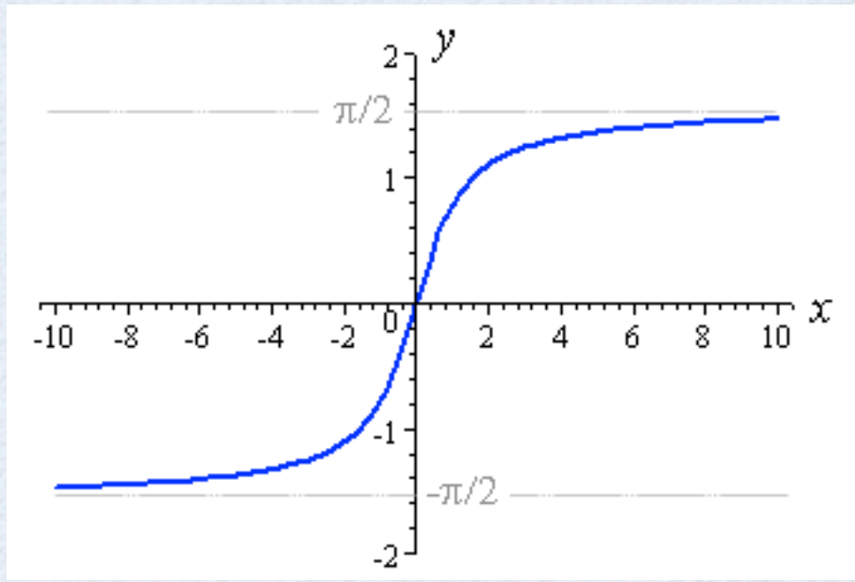
Overview of Probability and Statistics

ADMINISTRATIVE ISSUES

- Later today I will post 3 things:
  - Solutions to first homework
  - List of topics for paper presentation : please take a look and let me know what interests you (first come first served)
  - Example of how to compile code - turns out this depends on your compiler. I know how to do this on a linux, or mac system. For windows would have to download a compiler or ssh to another macine
- Typo in hw2 - variable name should be the one in the ntuple

ARCTAN



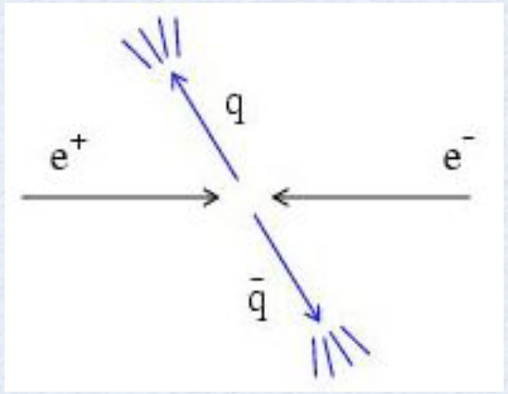
- The root library gives you two versions of the arc tan function
  - TMath::ATan(double x)
  - TMath::ATan2(double x,double y)
- Consider what happens if px = 0
  - the argument

$$\phi = \tan^{-1}(\frac{p_y}{p_x})$$

WHY DO WE CARE

- Measure event properties (number of muons, momentum of out particles

- Theories (for example Standard Model) predict their properties - mass of particles, strength of forces, angular distribution, etc



Observe events of a certain type

- We would like to :

  - from the data estimate that parameter

  - quantify how well we know it (what is the uncertainty)

  - Test the extent to which they agree with a theory

DEALING WITH UNCERTAINTY

- Theory is not deterministic:
  - Quantum Mechanics - only tells you about the probability for a particular process to occur

DEALING WITH UNCERTAINTY

- Theory is not deterministic:
  - Quantum Mechanics - only tells you about the probability for a particular process to occur
- Random Measurement Errors
  - Present even without quantum mechanics - (ask everyone to measure the length of a piece of paper with the same ruler - there will be a scatter around a central value

DEALING WITH UNCERTAINTY

- Theory is not deterministic:
  - Quantum Mechanics - only tells you about the probability for a particular process to occur

- Random Measurement Errors
  - Present even without quantum mechanics - (ask everyone to measure the length of a piece of paper with the same ruler - there will be a scatter around a central value

- Things we could know in principle but don't know exactly for various reasons
  - cost , time, etc..

DEALING WITH UNCERTAINTY

- Theory is not deterministic:
  - Quantum Mechanics - only tells you about the probability for a particular process to occur

- Random Measurement Errors
  - Present even without quantum mechanics - (ask everyone to measure the length of a piece of paper with the same ruler - there will be a scatter around a central value
- Things we could know in principle but don't know exactly for various reasons
  - cost , time, etc..
- We can quantify this with probability!

BASIC PROBABILITY

- Imagine a set S with two subsets A and B

For all $A \subset S, P(A) \geq 0$

$$P(S) = 1$$

If $A \cap B = \emptyset, P(A \cup B) = P(A) + P(B)$

Kolmogorov axioms (1933)

And...

$$P(\overline{A}) = 1 - P(A)$$

$$P(A \cup \overline{A}) = 1$$

$$P(\emptyset) = 0$$

if $A \subset B$, then $P(A) \leq P(B)$

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

CONDITIONAL PROBABILITY

The conditional probability of A given B :

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

Rolling dice :

$$P(n < 3 \mid n \text{ even}) = \frac{P((n<3) \cap n \text{ even})}{P(\text{even})} = \frac{1/6}{3/6} = \frac{1}{3}$$

A and B are independent if:

$$P(A \cap B) = P(A)P(B)$$

$$P(A|B) = \frac{P(A)P(B)}{P(B)} = P(A)$$

INTERPRETATION

- Relative Probability
  - Think of A and B as possible outcomes of repeated experiments (quantum mechanics, particle decay, etc...)

$$P(A) = \lim_{n\to\infty} \frac{\text{times outcome is } A}{n}$$

- Subjective Probability
  - A and B are hypothesis (either true or false) E.g. Horse number 6 will win the race

$$P(A) = \text{degree of belief that } A \text{ is true}$$

- we normally think of the relative probability in particle physics - but not always!

BAYES' THEOREM

From the definition of conditional probability :

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{and} \quad P(B|A) = \frac{P(B \cap A)}{P(A)}$$

but... $P(A \cap B) = P(B \cap A)$

so

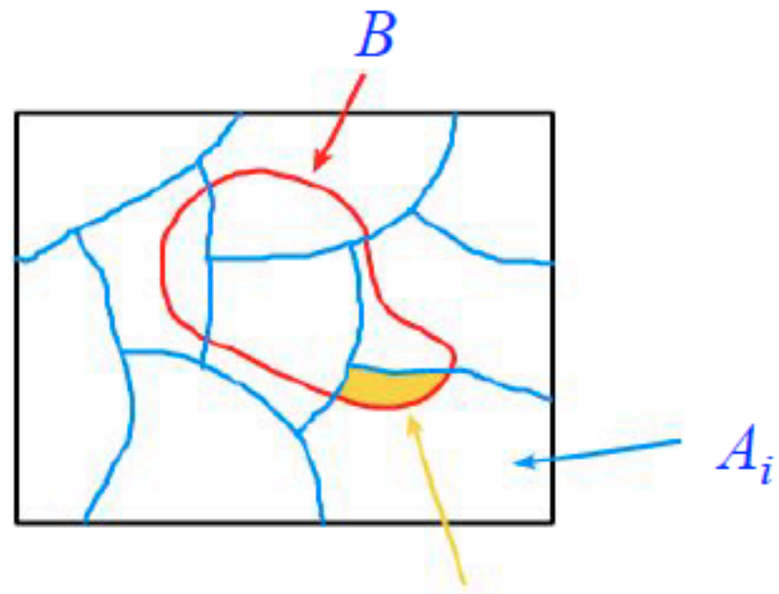
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

FOR MANY SUBSETS

The law of total probability

Consider a subset B of the sample space S,

divided into disjoint subsets A_i such that $\cup_i A_i = S$,



$$\rightarrow B = B \cap S = B \cap (\cup_i A_i) = \cup_i (B \cap A_i),$$

$$\rightarrow P(B) = P(\cup_i (B \cap A_i)) = \sum_i P(B \cap A_i)$$

$$\rightarrow P(B) = \sum_i P(B|A_i)P(A_i)$$ law of total probability

Bayes' theorem becomes

$$P(A|B) = \frac{P(B|A)P(A)}{\sum_i P(B|A_i)P(A_i)}$$

EXAMPLE

- Prior Probabilities
  - P(disease) = 0.001
  - P(no disease) = 0.999
- Consider a test for that disease
  - P(+ | disease) = 0.98
  - P(- | disease) = 0.02
  - P(+ | no disease) = 0.03
  - P(- | no disease) = 0.97

Prior Probability before any test

Probability to (mis) identify an infected person

Probability to (mis) identify a healthy person

If you test + what is the probability that you are infected

APPLY THE THEOREM

$$P(disease|+) = \frac{P(+|disease)}{P(+|disease)P(disease) + P(+|nodisease)P(disease)}$$

$$= \frac{0.98 \times 0.001}{0.98 \times 0.001 + 0.03 \times 0.999}$$

$$= 0.032$$

- After accounting for prior probabilities the probability of having the disease is actually fairly small
- Bayesian point of view: There is a 3.2% chance that I have the disease
- Frequentest point of view: 3.2% of people taking this test who test positive will have be infected

FREQUENTIST APPROACH

- In the frequentist approach we think of probability as only being associated with data (i.e. the outcome of repeated experiments)
  - Probability is the limiting frequency of an infinite number of experiments
- Statements such as:
  - The Higgs boson exists
  - The value of the strong coupling constant is between 0.117 and 0.121
- have probabilities of either 1 or 0. We just don't know which!
- The tools of the frequency approach tell us about what to expect under certain models or probabilities and what the probability of particular data are with a particular model

BAYESIAN APPROACH

probability of the data assuming hypothesis H (the likelihood)

prior probability, i.e., before seeing the data

$$P(H|\vec{x}) = \frac{P(\vec{x}|H)\pi(H)}{\int P(\vec{x}|H)\pi(H)\,dH}$$

posterior probability, i.e., after seeing the data

normalization involves sum over all possible hypotheses

- Has an if/then flavor. If you observe this data how does that change your estimation of the probability of hypothesis H
- Note : There is no prescription for prior probabilities! This is a tricky point. You and I could have different estimates of the prior probability which can have - in general - a much different result with the same data!

BAYESIAN/FREQUENTIST

- in HEP you will see results interpreted according to both views
- Some people have VERY strong opinions on the matter
  - there are many papers, conferences, and very loud discussions on the matter
- Personal view:
  - Each have their place in data analysis
  - MOST of the time it doesn't make a big difference - but in some important cases it can
  - Most important point is to be clear about how the data was treated so that someone can understand what was done and reanalyze it if needed

RANDOM VARIABLES AND PDFS

Suppose the outcome of some experiment is some continuous variable x

$$P(x \text{ found in } [x, x+dx]) = f(x)\,dx$$

f(x) = probability density function, this just describes the probability of observing x in some range

$$\int_{-\infty}^{\infty} f(x)\,dx = 1 \qquad x \text{ must be somewhere}$$

Can also work for discrete cases

$$P(x_i) = p_i$$

$$\sum_i P(x_i) = 1$$

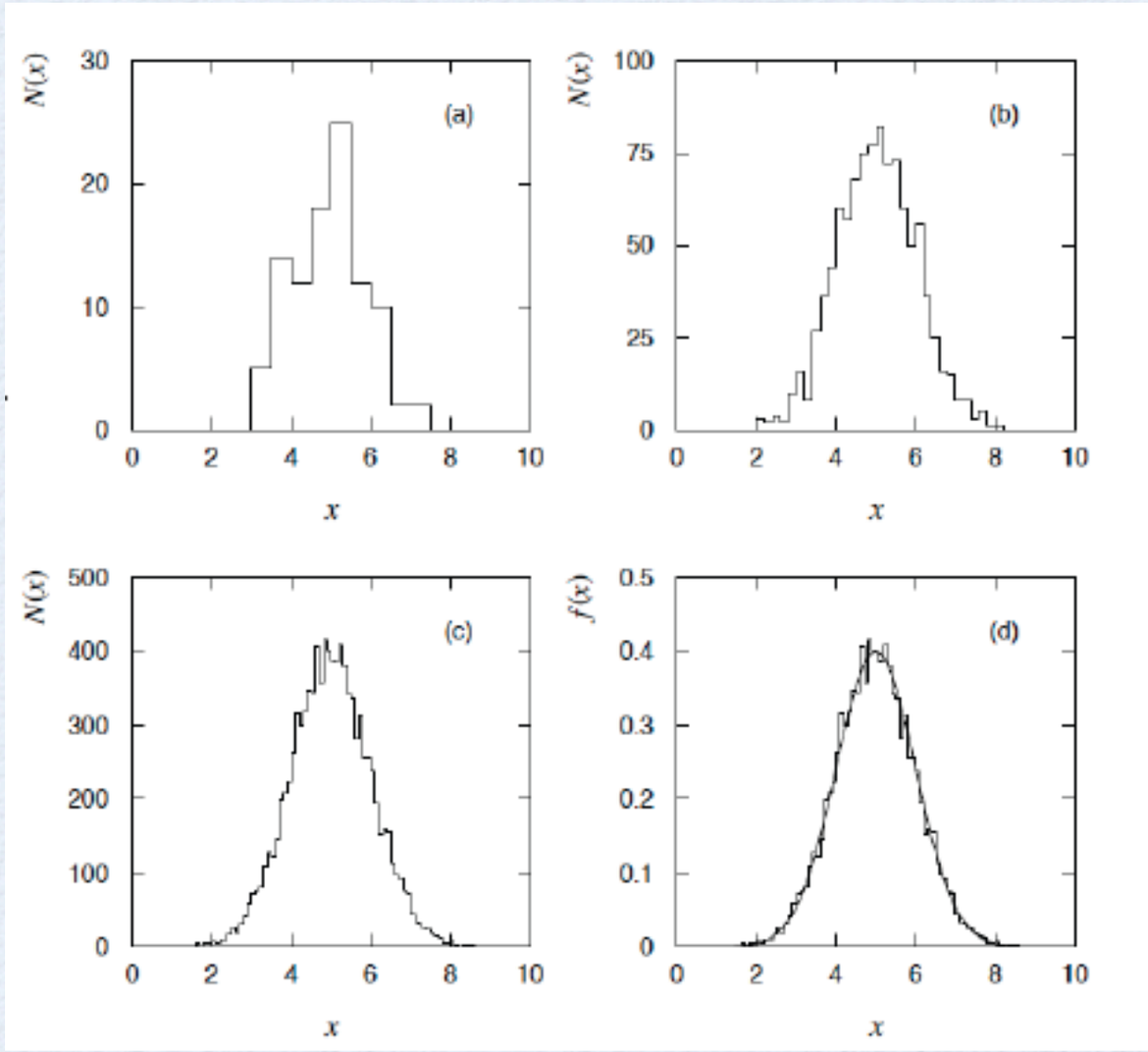
HISTOGRAMS AND PDF

- One can think of a PDF as a histogram with:
  - infinite statistics
  - zero bin width
  - normalized to unit area

$$f(x) = \frac{N(x)}{n\Delta x}$$

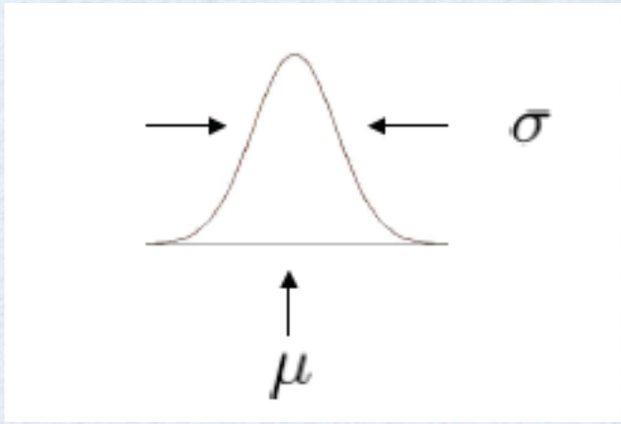
n = number of entries

Δx = bin width



Expectation values and all that

- Expectation value - what is the expected value of the variable x
- We would also like a measure of the spread of the variable
- We use the variance which is the expectation value of the distance between a value x and the mean squared
- Why do we use squared?



$$E[x] = \int x f(x) dx = \mu$$

$$V[x] = E[(x-\mu)^2] = \sigma^2$$

$$\sigma = \sqrt{\sigma^2}$$

COVARIANCE AND CORRELATION

$$\mathrm{cov}[x, y] = E[xy] - \mu_x \mu_y = E[(x - \mu_x)(y - \mu_y)]$$

$$\rho_{xy} = \frac{\mathrm{cov}[x, y]}{\sigma_x \sigma_y}$$

- Define the covariance between two variables and the related concept of the correlation coefficient

- If x and y are independent then

$$f(x, y) = f_x(x) f_y(y)$$

- so

$$E[xy] = \int \int xy\, f(x, y)\, dx dy = \mu_x \mu_y$$

Covariance and Correlation

- Define the covariance between two variables and the related concept of the correlation coefficient

$$\text{cov}[x, y] = E[xy] - \mu_x \mu_y = E[(x - \mu_x)(y - \mu_y)]$$

$$\rho_{xy} = \frac{\text{cov}[x, y]}{\sigma_x \sigma_y}$$

COVARIANCE AND CORRELATION

$$\mathrm{cov}[x, y] = E[xy] - \mu_x \mu_y = E[(x - \mu_x)(y - \mu_y)]$$

$$\rho_{xy} = \frac{\mathrm{cov}[x, y]}{\sigma_x \sigma_y}$$

- Define the covariance between two variables and the related concept of the correlation coefficient

- If x and y are independent then

$$f(x, y) = f_x(x) f_y(y)$$

COVARIANCE AND CORRELATION

$$\mathrm{cov}[x, y] = E[xy] - \mu_x \mu_y = E[(x - \mu_x)(y - \mu_y)]$$

- Define the covariance between two variables and the related concept of the correlation coefficient

$$\rho_{xy} = \frac{\mathrm{cov}[x, y]}{\sigma_x \sigma_y}$$

- If x and y are independent then

$$f(x, y) = f_x(x) f_y(y)$$

- so

$$E[xy] = \int \int xy\, f(x, y)\, dx dy = \mu_x \mu_y$$

COVARIANCE AND CORRELATION

$$\mathrm{cov}[x, y] = E[xy] - \mu_x \mu_y = E[(x - \mu_x)(y - \mu_y)]$$

- Define the covariance between two variables and the related concept of the correlation coefficient

$$\rho_{xy} = \frac{\mathrm{cov}[x, y]}{\sigma_x \sigma_y}$$

- If x and y are independent then

$$f(x, y) = f_x(x) f_y(y)$$

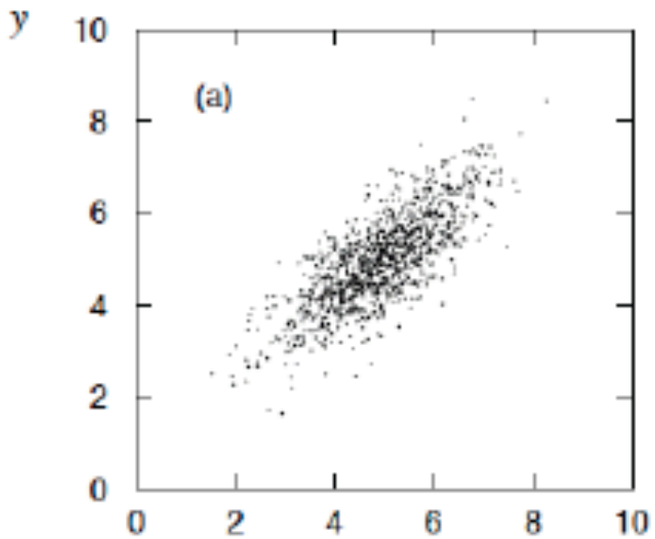
- so

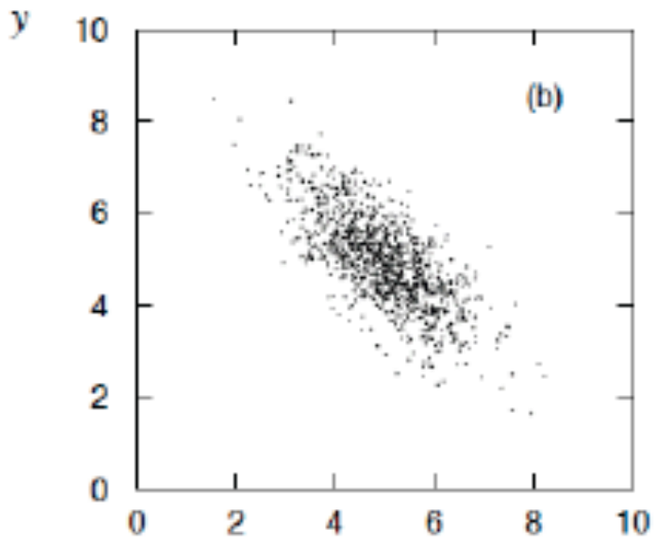
$$E[xy] = \int \int xy\, f(x, y)\, dx dy = \mu_x \mu_y$$

$$\mathrm{cov}[x, y] = 0$$

GRAPHICALLY

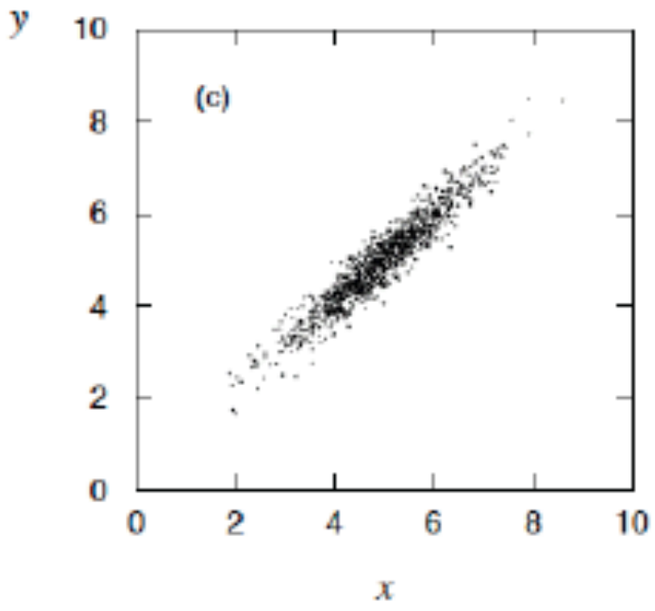
$\rho = 0.75$

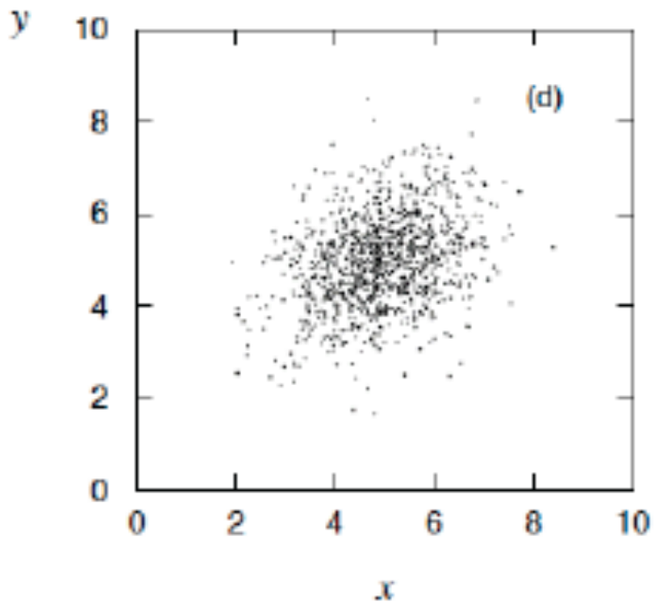




$\rho = -0.75$

$\rho = 0.95$





$\rho = 0.25$

In ROOT

- You can calculate this by yourself in a macro
- but if you have a 2D histogram root knows how to do it already
  - GetCovariance(Int_t axis1, Int_t axis2)
  - GetCorrelationFactor(Int_t axis1, Int_t axis2)

COMMON PDFS

- There are many distributions that we use in HEP but there are a few that come up over and over again
- I will go through a few most commonly used ones and you can then look up other distributions and their properties
- Today I will discuss :
  - Binomial
  - Poisson
  - Gaussian

BINOMIAL DISTRIBUTION

- Consider N independent experiments (e.g. flipping a coin over and over again) which has one of two outcomes: success and failure
- Let the probability of 'success' be p. Then the probability of failure is (1-p)
- Then if we had a sequence of successes and failures then the probability of a particular sequence (since each experiment is independent) would look something like :
  - p*p*(1-p)*p*(1-p)*p
- If we have N experiments with successes we can write the probability for any particular sequence as $p^n(1-p)^{N-n}$
- Typically we don't care about the order of the sequence just number of successes

$$\frac{N!}{n!(N-n)!} \quad p^n(1-p)^{N-n}$$

BINOMIAL

$$f(n; N, p) = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n}$$

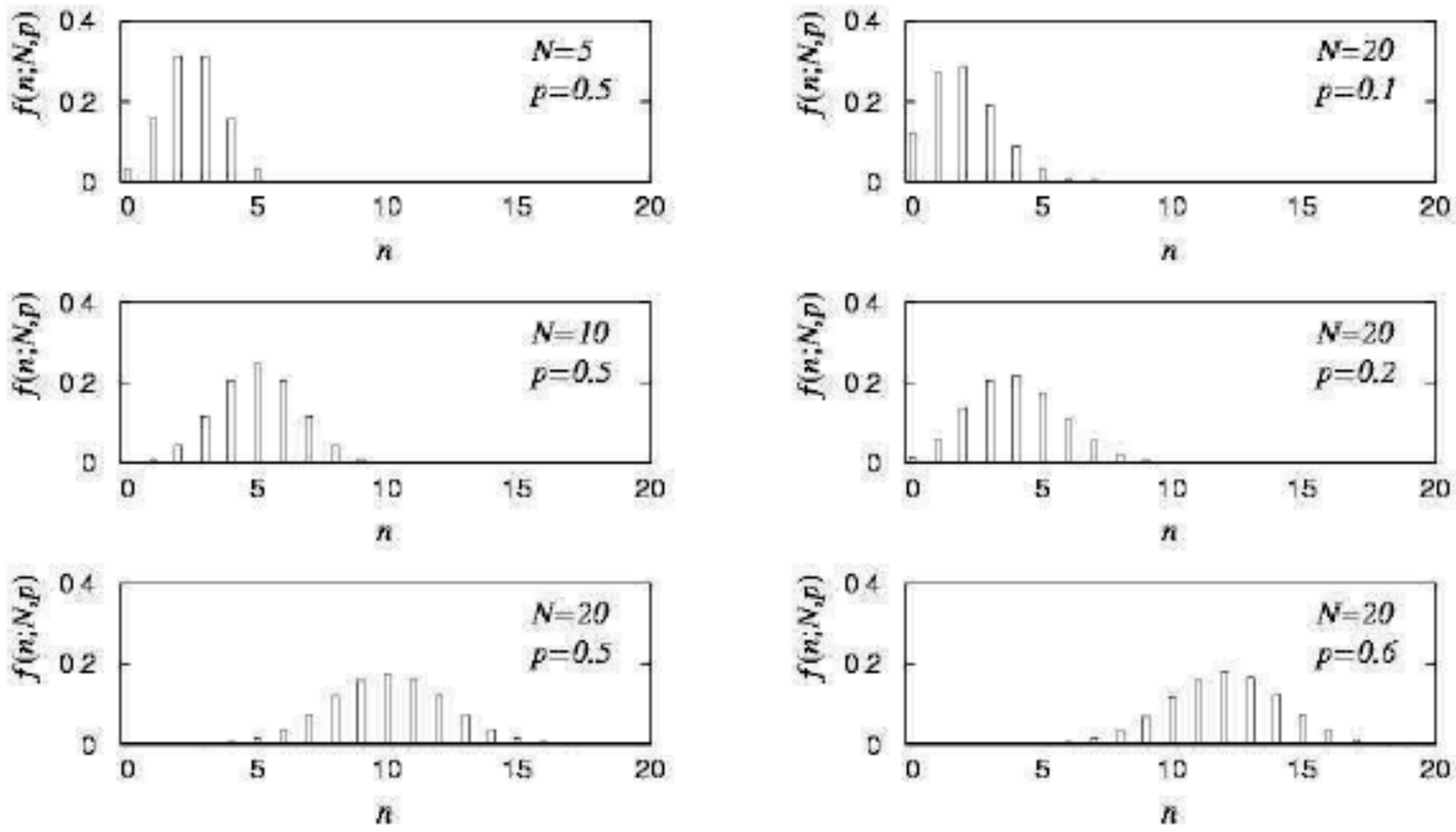
random variable

parameters

$$E[n] = \sum_{n=0}^{N} n f(n; N, p) = Np$$

$$V[n] = E[n^2] - (E[n])^2 = Np(1-p)$$

WHAT IT LOOKS LIKE



W decay into an electron and a neutrino with branching ratio p

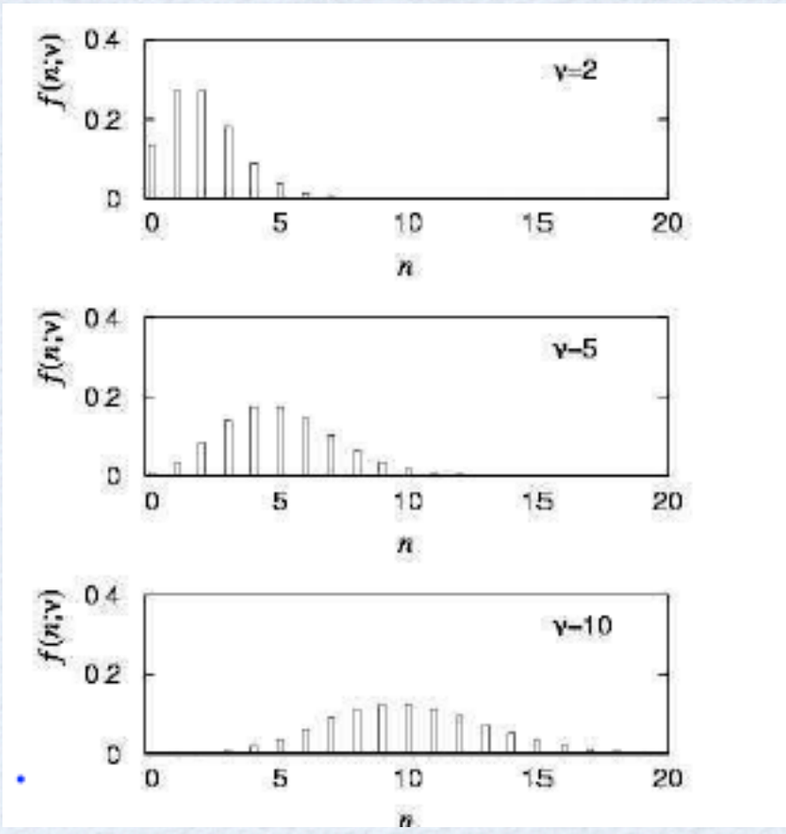
POISSON DISTRIBUTION

- Consider the binomial

$$N \to \infty, \qquad p \to 0, \qquad E[n] = Np \to \nu$$

$$f(n;\nu) = \frac{\nu^n}{n!} e^{-\nu} \quad (n \geq 0)$$

$$E[n] = \nu\,, \quad V[n] = \nu\,.$$



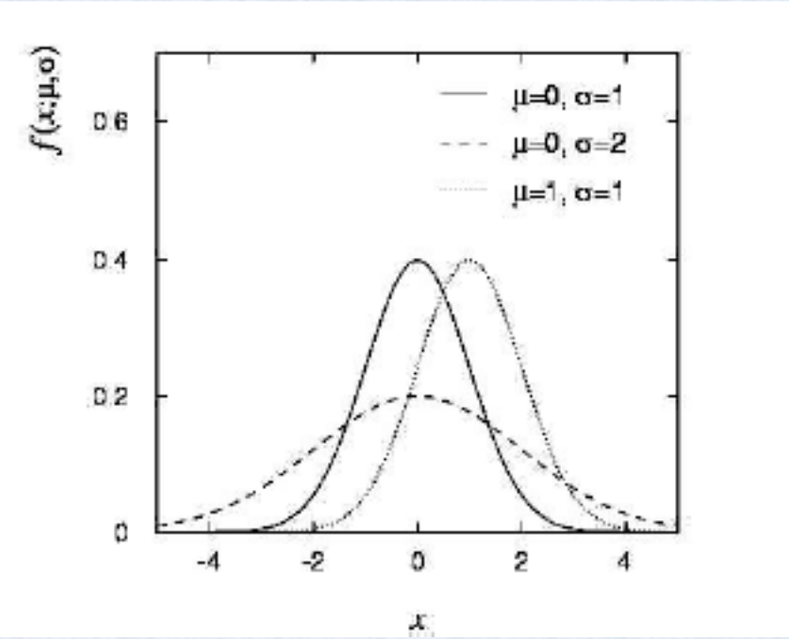
Example: number of scattering events n with cross section σ found for a fixed integrated luminosity, with $\nu = \sigma \int L\, dt$

GAUSSIAN DISTRIBUTION

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

$E[x] = \mu$ (N.B. often μ, σ^2 denote mean, variance of any r.v., not only Gaussian.)

$V[x] = \sigma^2$



This is probably the most important probability distribution!

CENTRAL LIMIT THEOREM

- The Gaussian pdf is so useful because almost any random variable that is a sum of a large number of small contributions follows it. This follows from the Central Limit Theorem:

For n independent r.v.s x_i with finite variances σ_i^2, otherwise arbitrary pdfs, consider the sum

$$y = \sum_{i=1}^{n} x_i$$

In the limit $n \to \infty$, y is a Gaussian r.v. with

$$E[y] = \sum_{i=1}^{n} \mu_i \qquad V[y] = \sum_{i=1}^{n} \sigma_i^2$$

TMath

- http://root.cern.ch/root/html/TMath.html

- Contains many useful functions including Poisson, Guas, Poisson

- Many other

PARAMETER ESTIMATION

- Often it is the case that we have some data that we either know or think we know the probability distribution it comes from (shape of : lifetime of a particle, observed mass of a particle, etc)

  - but we DON'T know the parameters of this particular distribution (lifetime or mass of this PARTICULAR particle)

  - from our data we would like to get an estimate of these parameters

PARAMETER ESTIMATION

$$f(x;\theta) = \frac{1}{\theta} e^{-x/\theta}$$

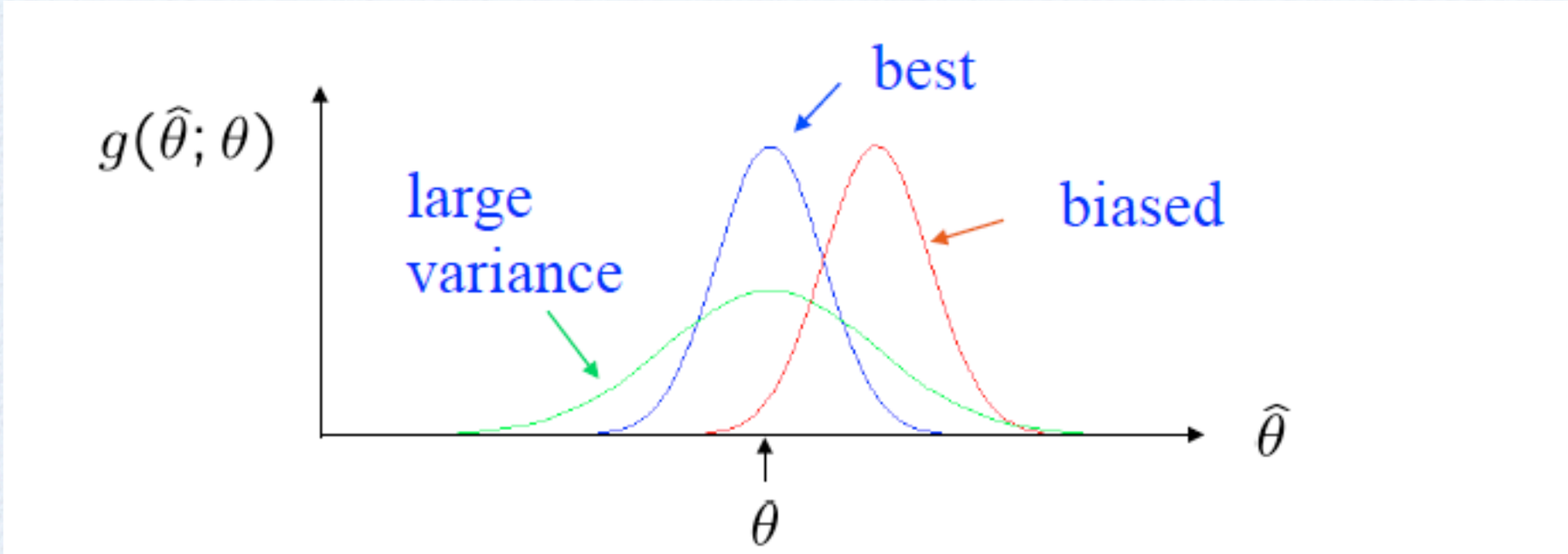
r.v. parameter

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

- Suppose we have some known distribution f and some measured data values
- We want to find some estimation of the parameter theta
- we call it the "estimator"

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

WHAT ARE WE LOOKING FOR?



- We would like an estimator that has little or zero bias (i.e. it estimates the true value of the parameter not something else) - we call this the “systematic error”
- We would like an estimator that has a small variance - we call this the “statistical error”

ESTIMATE OF THE MEAN

Parameter: $\mu = E[x]$

Estimator: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i \equiv \overline{x}$ ('sample mean')

We find: $b = E[\hat{\mu}] - \mu = 0$

$$V[\hat{\mu}] = \frac{\sigma^2}{n} \qquad \left(\sigma_{\hat{\mu}} = \frac{\sigma}{\sqrt{n}} \right)$$

ESTIMATOR OF VARIANCE

Parameter: $\sigma^2 = V[x]$

Estimator: $\widehat{\sigma^2} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x})^2 \equiv s^2$ ('sample variance')

- bias is again zero

LIKELIHOOD

- Suppose the entire result of an experiment (set of measurements) is a collection of numbers x, and suppose the joint pdf for the data x is a function that depends on a set of parameters θ:

$$f(\vec{x};\vec{\theta})$$

- Now evaluate this function with the data obtained and regard it as a function of the parameter(s). This is the likelihood function:

$$L(\vec{\theta}) = f(\vec{x};\vec{\theta})$$

LIKELIHOOD

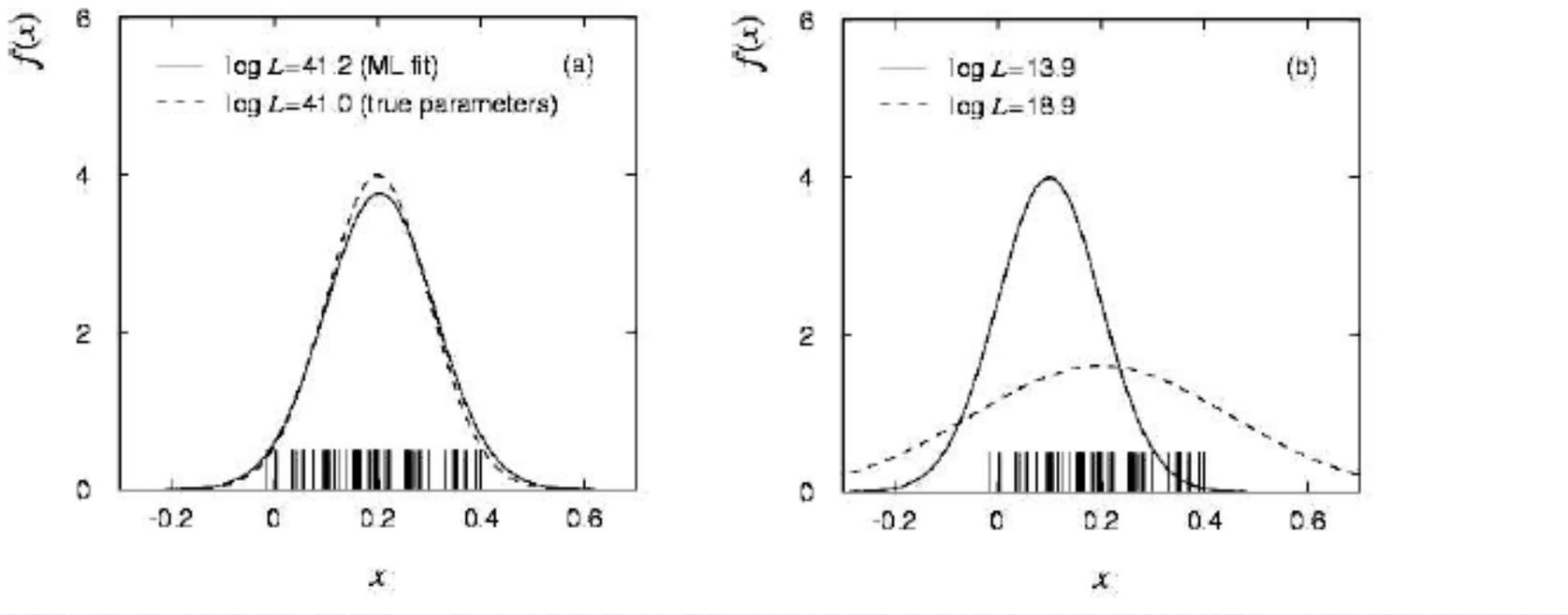
- Consider n independent observations of x: x1, ..., xn, where x follows f (x; θ). The joint pdf for the whole data sample is:

$$f(x_1, \ldots, x_n; \theta) = \prod_{i=1}^{n} f(x_i; \theta)$$

- The likelihood function is

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

LIKELIHOOD



- If the hypothesized θ is close to the true value, then we expect a high probability to get data like that which we actually found.

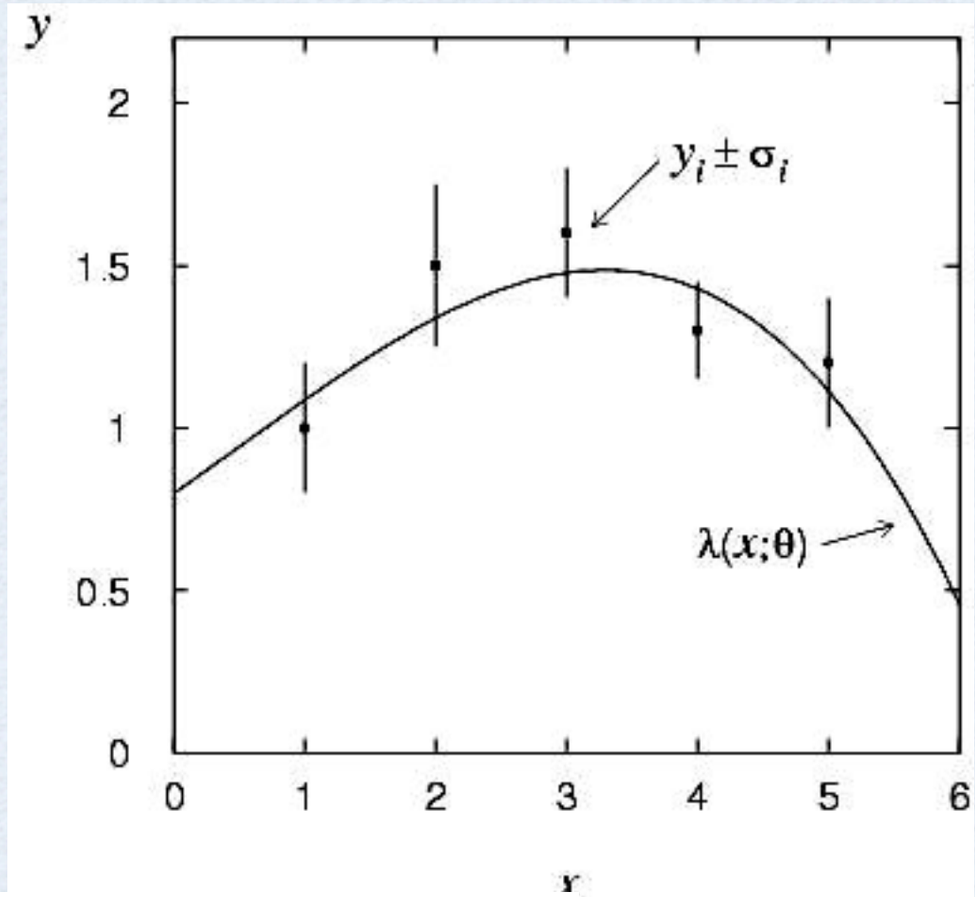
- So we define the maximum likelihood (ML) estimator(s) to bethe parameter value(s) for which the likelihood is maximum

LEAST SQUARES METHOD

- Suppose we measure N values, y1, ..., yN,assumed to be independent Gaussian random variables with $E[y_i] = \lambda(x_i; \theta)$
- Assume known values of the control variable x1, ..., xN and known variances

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

- We want to estimate θ, i.e., fit the curve to the data points.



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

LEAST SQUARES

Taking the logarithm we get

$$\ln L(\theta) = -\frac{1}{2}\sum_{i=1}^{N}\frac{(y_i - \lambda(x_i;\theta))^2}{\sigma_i^2} + \text{terms not depending on } \theta$$

Maximizing the likelihood is equivalent to minimizing

$$\chi^2(\theta) = \sum_{i=1}^{N}\frac{(y_i - \lambda(x_i;\theta))^2}{\sigma_i^2}$$

Often done numerically :
for simple one dimensional functions can be done analytically

FITTING WITH ROOT

- Today we will start with some very simple examples
  - From your example code make a histogram of one of the variables - say ebeam
  - There are a few simple commands we can do

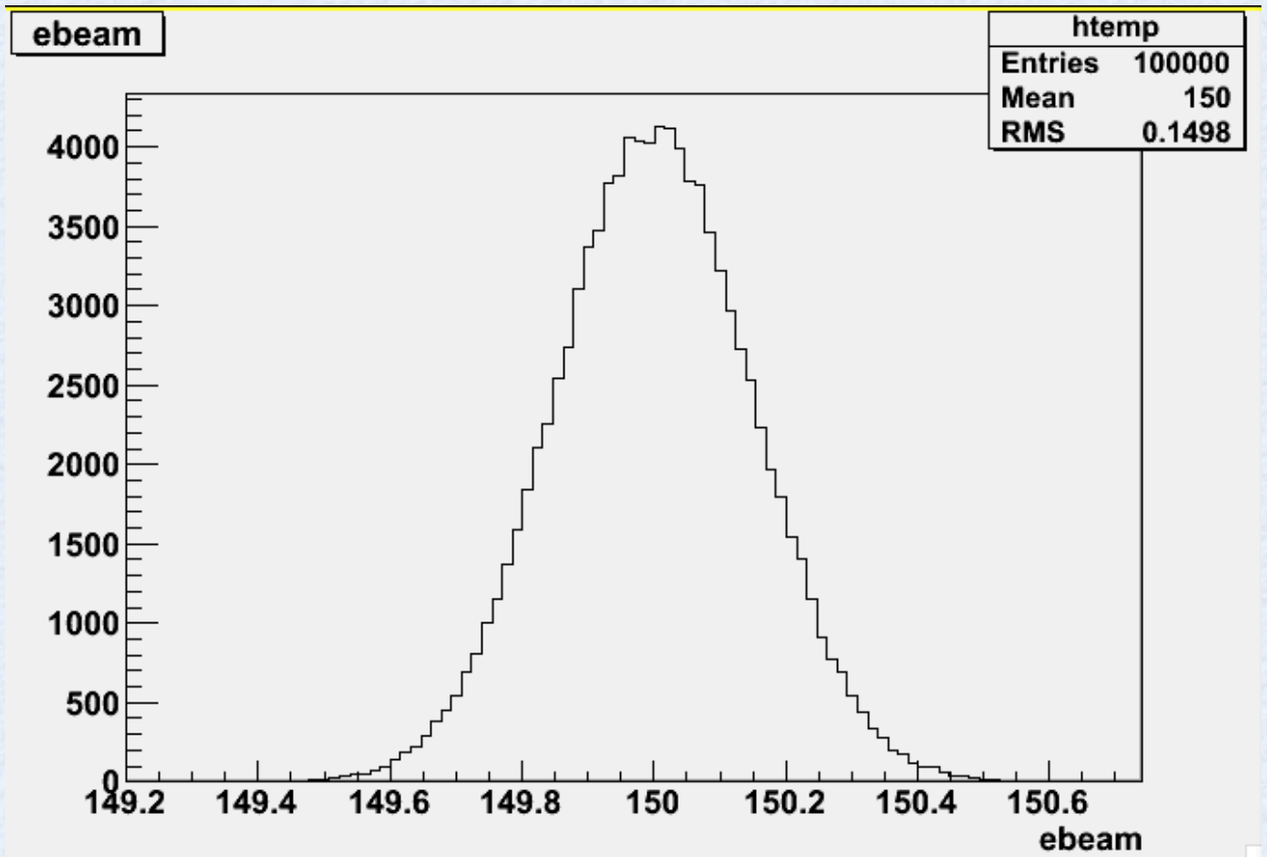
PREDEFINED FITTING

- Root has a few very common functions built in that it will fit for you very easily
  - gaussian
  - polynomial
  - exponential

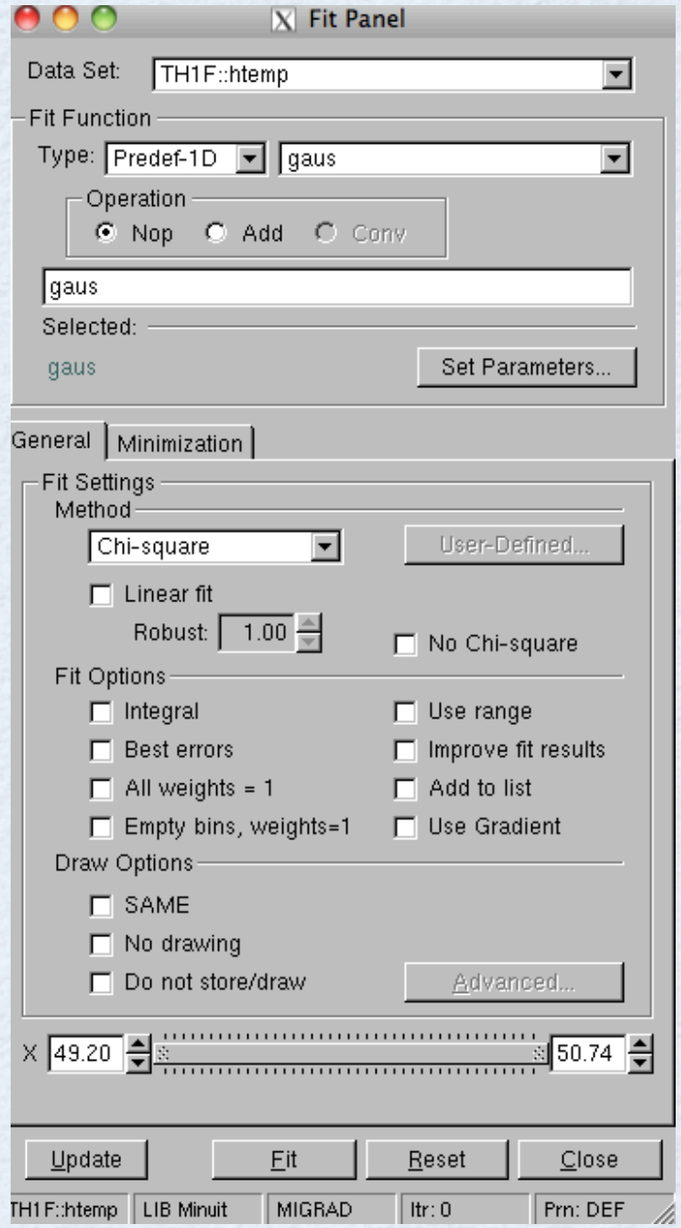
EXAMPLE

- if you have a histogram (like ebeam) that looks fairly gaussian you can fit it to a gaussian with
- h_ebeam.Fit('gaus')

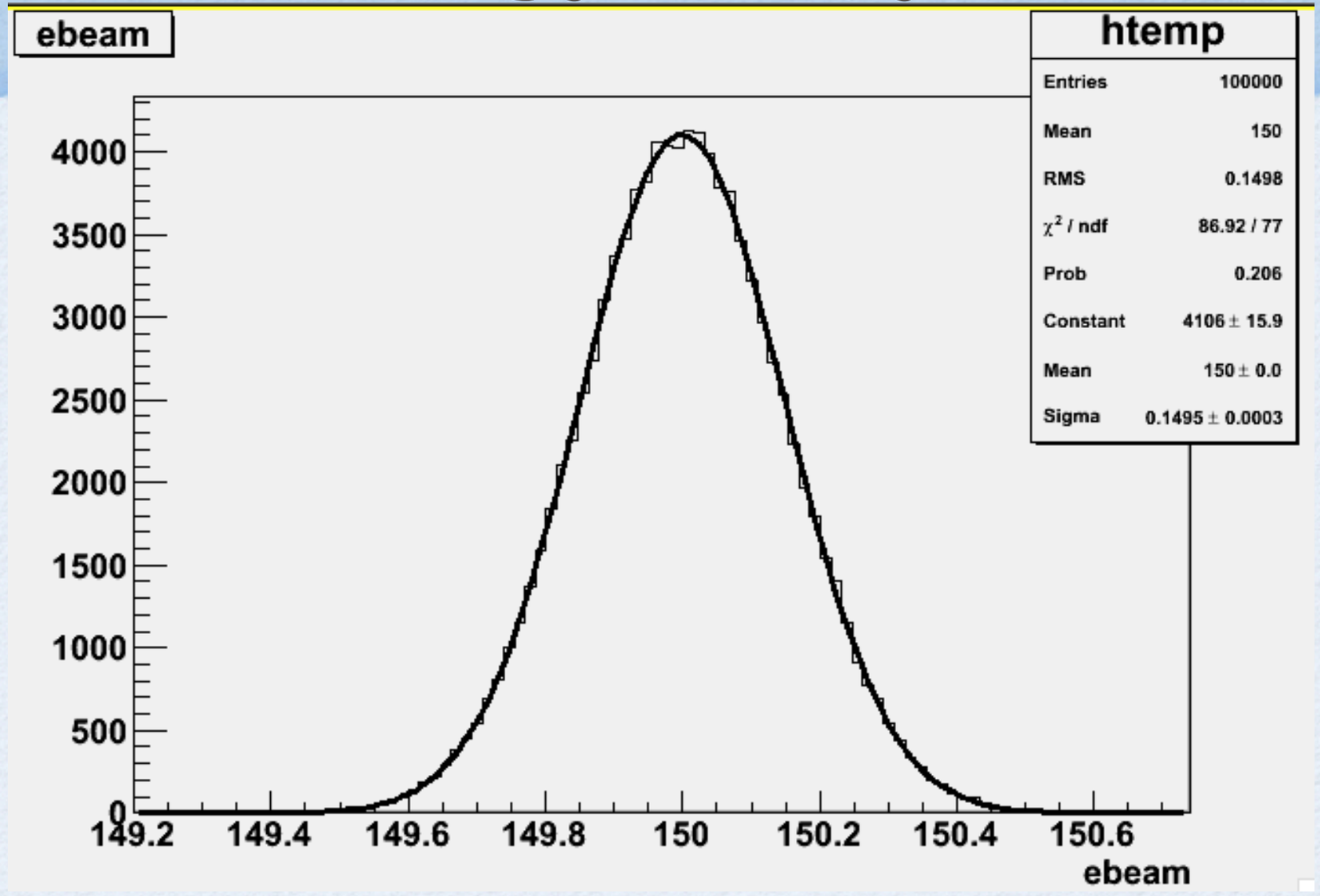
FITTING FROM GUI



right click on histogram
choose Fit Panel



RESULTS

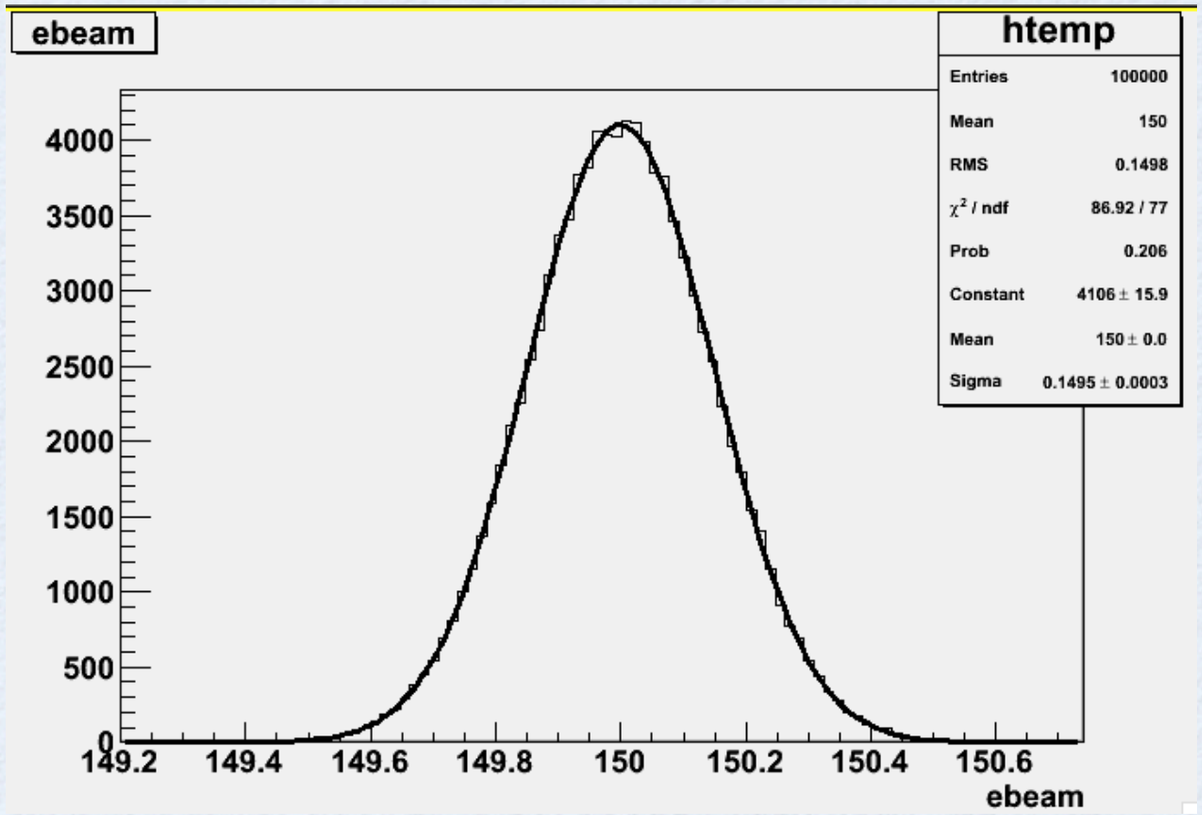


Right Click on statistics box and choose SetFitOpt

Change it to 1111111

RESULTS

- Reports result of fit and uncertainty on parameters
- chi2/ndf



What does this mean

- In general we are looking to minimize the chi2

- Number of degrees of freedom = number of data points - 1

- For a good fit we expect chi2/NDF ~ 1

  - If very large - this means function is very far from each point . Bad Fit

  - If much smaller than 1 this is also suspect - statistical fluctuations are not consistent with uncertainty on each point. Typically this means our assigned uncertainties are too large

$$\chi^2(\theta) = \sum_{i=1}^{N} \frac{(y_i - \lambda(x_i;\theta))^2}{\sigma_i^2}$$

FITTING WITH MACRO

- Lets look at an example: FittingDemo.C is on blackboard and lxplus at ~/kblack/public
- Download it now and we will walk through it

FITTING DEMO

```
// Quadratic background function
Double_t background(Double_t *x, Double_t *par) {
   return par[0] + par[1]*x[0] + par[2]*x[0]*x[0];
}


// Lorenzian Peak function
Double_t lorentzianPeak(Double_t *x, Double_t *par) {
  return (0.5*par[0]*par[1]/TMath::Pi()) /
    TMath::Max( 1.e-10,(x[0]-par[2])*(x[0]-par[2])
   + .25*par[1]*par[1]);
}

// Sum of background and peak function
Double_t fitFunction(Double_t *x, Double_t *par) {
  return background(x,par) + lorentzianPeak(x,&par[3]);
}
```

Define a function which we will fit our fake data

Background we take as a polynomial

Signal as a Lorenzian

THE MAIN FUNCTION

```
void FittingDemo() {
 //Bevington Exercise by Peter Malzacher, modified by Rene Brun

   const int nBins = 60;

   Double_t data[nBins] = { 6, 1,10,12, 6,13,23,22,15,21,
                            23,26,36,25,27,35,40,44,66,81,
                            75,57,48,45,46,41,35,36,53,32,
                            40,37,38,31,36,44,42,37,32,32,
                            43,44,35,33,33,39,29,41,32,44,
                            26,39,29,35,32,21,21,15,25,15};
   TCanvas *c1 = new TCanvas("c1","Fitting Demo",10,10,700,500);
   c1->SetFillColor(33);
   c1->SetFrameFillColor(41);
   c1->SetGrid();

   TH1F *histo = new TH1F("histo",
      "Lorentzian Peak on Quadratic Background",60,0,3);
   histo->SetMarkerStyle(21);
   histo->SetMarkerSize(0.8);
   histo->SetStats(0);

   for(int i=0; i < nBins;  i++) histo->SetBinContent(i+1,data[i]);
```

Generate some fake data and put it in a histogram

NOW FIT

```
// create a TF1 with the range from 0 to 3 and 6 parameters
TF1 *fitFcn = new TF1("fitFcn",fitFunction,0,3,6);
fitFcn->SetNpx(500);
fitFcn->SetLineWidth(4);
fitFcn->SetLineColor(kMagenta);

// first try without starting values for the parameters
// This defaults to 1 for each param.
// this results in an ok fit for the polynomial function
// however the non-linear part (lorenzian) does not
// respond well.
fitFcn->SetParameters(1,1,1,1,1,1);
histo->Fit("fitFcn","0");

// second try: set start values for some parameters
fitFcn->SetParameter(4,0.2); // width
fitFcn->SetParameter(5,1);   // peak

histo->Fit("fitFcn","V+","ep");
```

Note: in general you have to give fitting programs an initial guess at the parameters as for nonlinear functions the program iterates numerically

MAKE A SILLY PICTURE

```
// improve the picture:
TF1 *backFcn = new TF1("backFcn",background,0,3,3);
backFcn->SetLineColor(kRed);
TF1 *signalFcn = new TF1("signalFcn",lorentzianPeak,0,3,3);
signalFcn->SetLineColor(kBlue);
signalFcn->SetNpx(500);
Double_t par[6];

// writes the fit results into the par array
fitFcn->GetParameters(par);

backFcn->SetParameters(par);
backFcn->Draw("same");

signalFcn->SetParameters(&par[3]);
signalFcn->Draw("same");

// draw the legend
TLegend *legend=new TLegend(0.6,0.65,0.88,0.85);
legend->SetTextFont(72);
legend->SetTextSize(0.04);
legend->AddEntry(histo,"Data","lpe");
legend->AddEntry(backFcn,"Background fit","l");
legend->AddEntry(signalFcn,"Signal fit","l");
legend->AddEntry(fitFcn,"Global Fit","l");
legend->Draw();
```

Extract the parameters and draw the result on the same place as the data and label everything