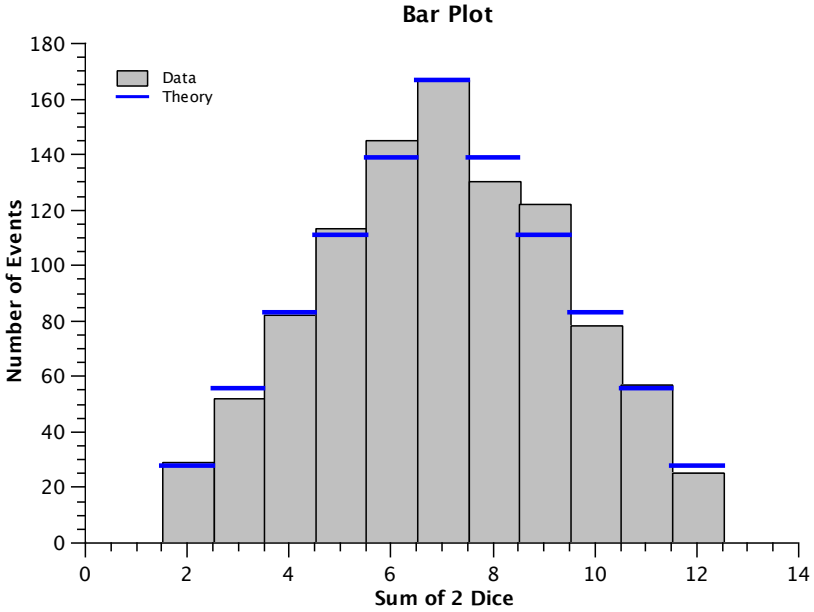
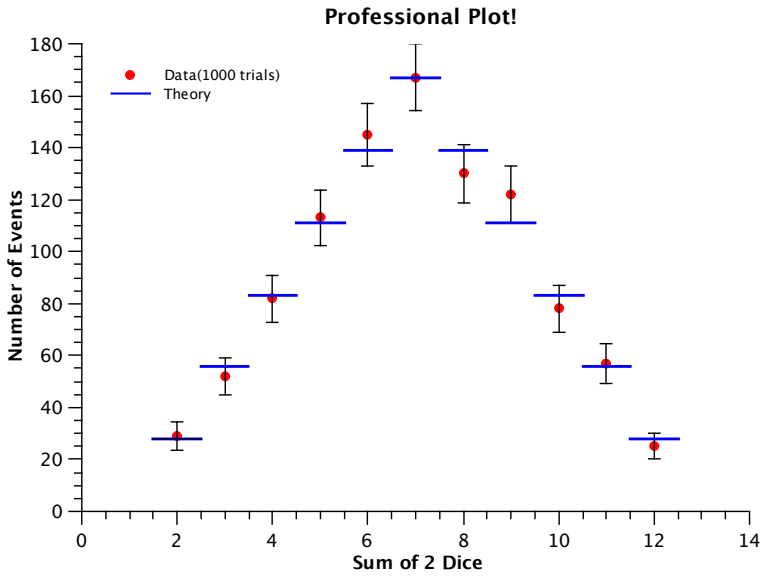
You must start using Error Bars in Lab 2

OK for Lab 1 **not** Lab 2



Labs 2-6



- **From now on data is represented by a symbol with an error bar.**
- **Theory is represented by lines.**

Histogram Bin Error

N – number of events
P_i – probability in ith bin $P_i = \int_{x_i}^{x_{i+1}} f(x)dx$
n_i – number of events in ith bin

N Known (M.C., Dice throws,…)

Binomial

$$\mu_{n_i} = NP_i = n_i$$

$$\sigma_{n_i} = \sqrt{NP_i(1-P_i)}$$

N Unknown (Everything else,…)

Binomial

$$\mu_{n_i} = NP_i = n_i$$

$$\sigma_{n_i} = \sqrt{NP_i(1-P_i)}$$

Poissonian (counting experiment)

$$\mu'_N = N \quad \text{so} \quad \mu'_{n_i} = P_i N$$

$$\sigma'_N = \sqrt{N} \qquad \sigma'_{n_i} = P_i\sqrt{N}$$

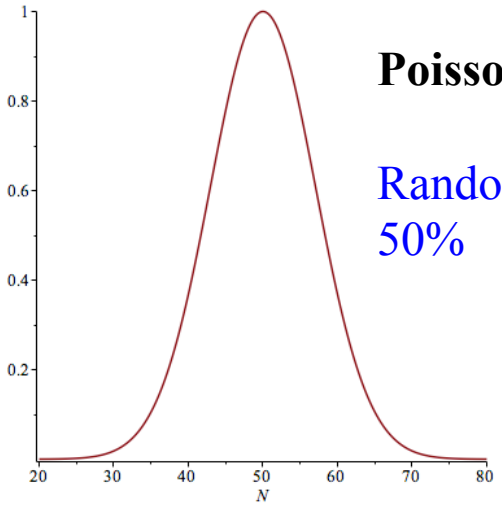
Combining Binomial and Poissonian

$$\sigma^2_{n_i Total} = \sigma^2_{n_i} + \sigma'^2_{n_i} = NP_i(1-P_i) + NP_i^2 = NP_i = n_i$$

A counting experiment in a single bin !

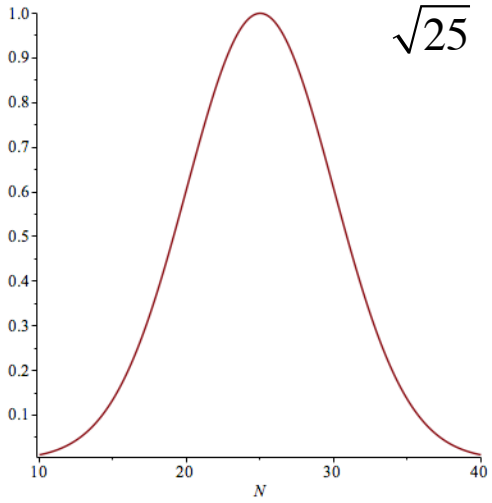
Counting Experiment (Nature)

Wait for fixed interval (100 s) for cosmics to arrive



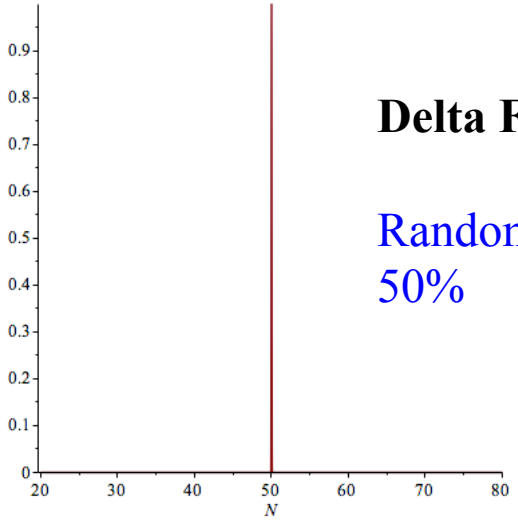
Poisson

Randomly accept 50%



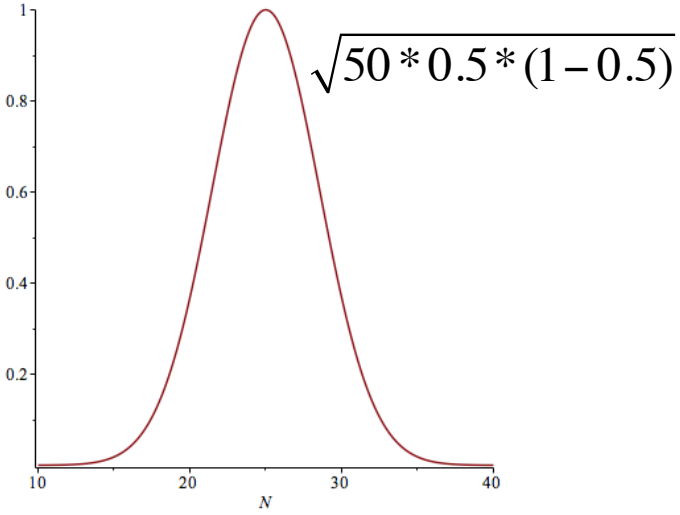
Dice Roll, Coin Toss, Monte Carlo

Flip a coin 50 times



Delta Function

Randomly accept 50%



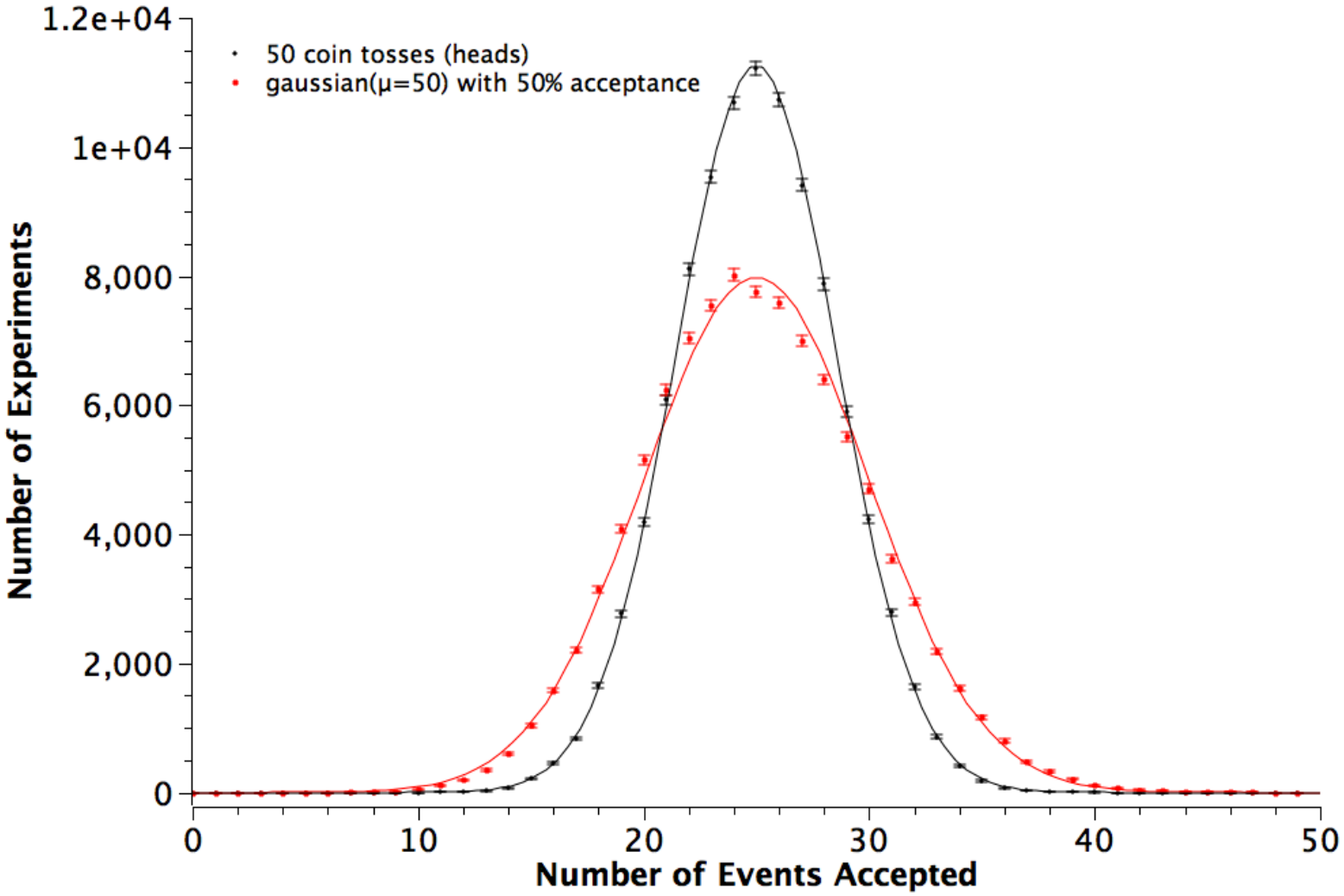
Test with 100,000 Monte Carlo Experiments

Throw Coin 50 times and randomly accept 50% (e.g. "heads")

expected: $\sigma = \sqrt{50 \bullet 0.5 \bullet (1-0.5)} = 3.5355$ result: 3.537

Generate N from Gaussian (~Poissonian) with mean of 50 and randomly accept 50%

expected: $\sigma = \sqrt{50 \bullet 0.5} = 5.0$ results: 4.997



C-CODE for Simulation

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

double gauss(double mean,double sigma);
int histpois[50]={0};
int histbinom[50]={0};

int main(){
 int N;
 int ntrial,datapnt;
 int npois,nbinom;
 double xpois,xbinom;
 double ng,xg,x2g;
 double np,xp,x2p;
 double nb,xb,x2b;
 double mean,sigma;
 srand(1234);
 ng=0.0;xg=0.0,x2g=0.0;
 np=0.0;nb=0.0;
 xp=0.0;xb=0.0;
 x2p=0.0;x2b=0.0;
 for(ntrial=0;ntrial<100000;ntrial++){
   //binomial fixed N
   nbinom=0;
  for(datapnt=0;datapnt<50;datapnt++){
    if((double)rand()/(double)RAND_MAX<0.5)nbinom++;
  }
   xbinom=nbinom;
   nb=nb+1.0;
   xb=xb+xbinom;
  x2b=x2b+xbinom*xbinom;
  if(nbinom>-1&&nbinom<50)histbinom[nbinom]++;
  //poisson gaussian N
   npois=0;
   N=gauss(50.5,sqrt(50.));  // the 0.5 is for roundoff
   ng=ng+1.0;
   xg=xg+N;
   x2g=x2g+N*N;
  for(datapnt=0;datapnt<N;datapnt++){
    if((double)rand()/(double)RAND_MAX<0.5)npois++;
  }
   xpois=npois;
   np=np+1.0;
   xp=xp+xpois;
  x2p=x2p+xpois*xpois;
  if(npois>-1&&npois<50)histpois[npois]++;
 }
 // gauss
 mean=xg/ng;
 sigma=sqrt(x2g/ng-mean*mean);
 printf(" gauss mean %f sigma %f \n",mean,sigma);
 // binom
 mean=xb/nb;
 sigma=sqrt(x2b/nb-mean*mean);
 printf(" binomial mean %f sigma %f \n",mean,sigma);
 // pois
 mean=xp/np;
 sigma=sqrt(x2p/np-mean*mean);
 printf(" poisson  mean %f sigma %f \n",mean,sigma);
 return 1;
}

double gauss(double mean, double sigma){
  int i;
  double x;
  x=0.0;
  for(i=0;i<12;i++)x=x+(double)rand()/(double)RAND_MAX-0.5;
  x=x*sigma;
  x=x+mean;
    return x;
}
```