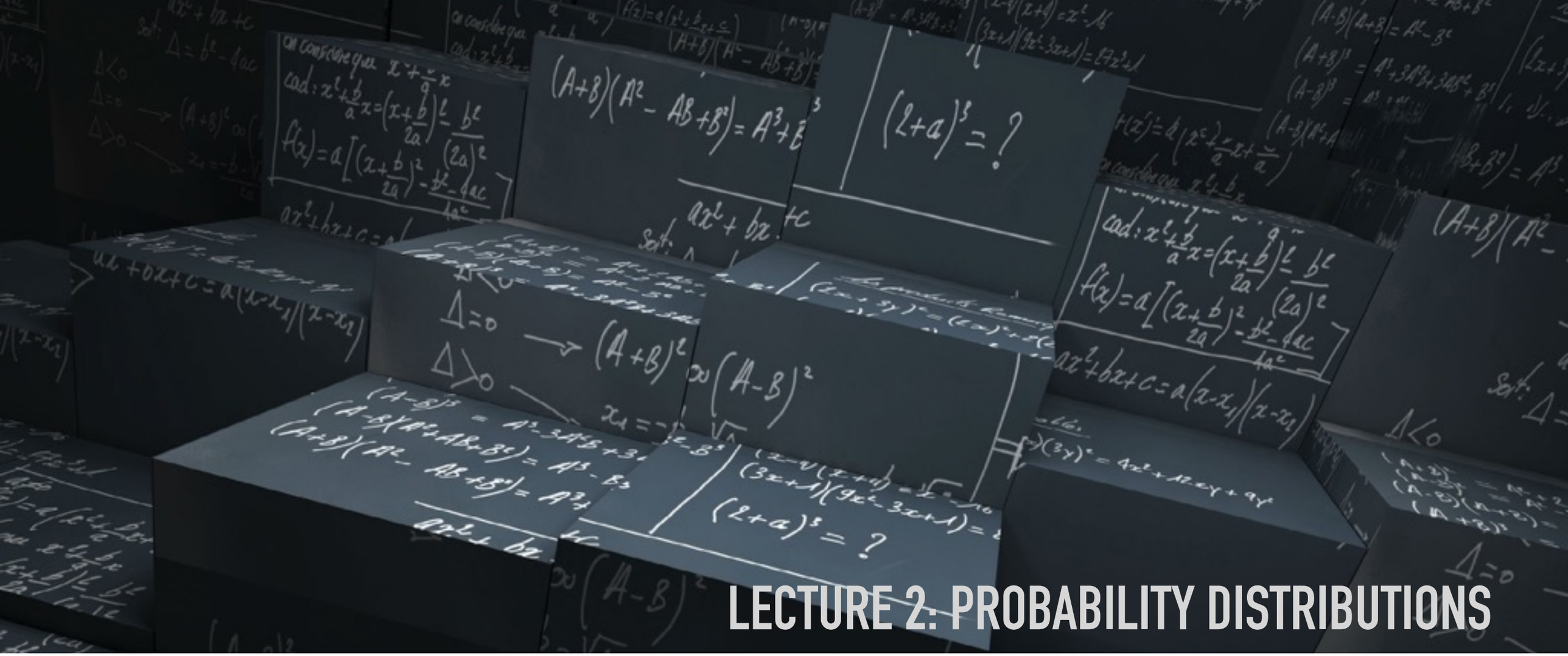


LECTURE 2: PROBABILITY DISTRIBUTIONS

STATISTICAL ANALYSIS IN EXPERIMENTAL PARTICLE PHYSICS

Kai-Feng Chen
National Taiwan University

PROPERTIES OF DISTRIBUTIONS

- Several useful quantities which characterize probability distributions. The PDF ***f(X)*** is used as a weighting function to obtain the corresponding quantities.
- The **expectation** ***E*** of a function ***g(X)*** is given by

$$E(g) = \langle g(X) \rangle = \int_{\Omega} g(X) f(X) dx$$

  where **Ω** is the entire space.
- The **mean** is simply the expected value of ***X***:

$$\mu = E(X) = \langle X \rangle = \int_{\Omega} X f(X) dx$$

- The expectation of the function $(X-\mu)^2$ is the **variance *V***:

$$V = \sigma^2 = E((X-\mu)^2) = \int_{\Omega} (X-\mu)^2 f(X) dx = \int_{\Omega} X^2 f(X) dx - \mu^2$$

COVARIANCE AND CORRELATION

- **Covariance** and **correlation** are two further useful numerical characteristics. Consider a joint density $f(X,Y)$ of two variables, the covariance is the expectation of $(X-\mu_X)(Y-\mu_Y)$:

$$\operatorname{cov}(X,Y) = E((X-\mu_X)(Y-\mu_Y)) = E(XY) - E(X)E(Y)$$

- Another one is the correlation coefficient, which is defined by

$$\operatorname{corr}(X,Y) = \rho(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- When there are more than 2 variables, the covariance (and correlation) can be still defined for each 2D joint distribution for X_i and X_j. The matrix with elements $\mathbf{cov}(X_i,X_j)$ is called the **covariance matrix** (or variance/error matrix). The diagonal elements are just the variances:

$$\operatorname{cov}(X_i,X_i) = E(X_i^2) - E(X_i)^2 = \sigma_{X_i}^2$$

UNCORRELATED? INDEPENDENT?

- A usual confusion is the two statements **"uncorrelated"** and **"independent"**. In fact the requirement for "uncorrelated" is much weaker than "independent".
- Consider a distribution $f(X)$ is symmetric along X (for example, a simple flat distribution within $[-1,+1]$), and consider a **very dependent** $Y = X^2$, this will give the following result:

$$E(X) = 0 \quad \text{and} \quad E(Y) = E(X^2) = \int X^2 f(X) dX = \sigma^2$$

$$\text{cov}(X, Y) = E(XY) - E(X)E(Y) = E(X^3) = 0$$

Since the covariance (and the correlation coefficient) is zero.
So X and Y are uncorrelated, although they are very dependent!

PRACTICE: MEAN, VARIANCE, COVARIANCE

- Let's practice these ideas with the following example code.
  - Generate a random distribution and use it to calculate the mean and variance.

example_01.cc

```
{
    TRandom3 rnd;
    TNtupleD *nt = new TNtupleD("nt","random data","x");
    for(int i=0; i<100000; i++) {
        double x = rnd.Uniform(-1.,1.);
        nt->Fill(&x);
    }

    double mean = 0., variance = 0.;
    for(int i=0; i<nt->GetEntries(); i++) {
        nt->GetEntry(i);
        double x = nt->GetArgs()[0];
        mean += x;
        variance += x*x;
    }
    mean /= nt->GetEntries();
    variance = variance/nt->GetEntries() - mean*mean;

    printf("Mean: %g\n",mean);
    printf("Variance: %g\n",variance);
}
```

Put in an uniform distribution here, you may try something else!

```
mean: -0.00107424
variance: 0.332499
```

for uniform distribution, the variance should be $(Max-Min)^2/12 \sim 0.333$

➤ How about the covariance?

example_02.cc

```
{
    TRandom3 rnd;
    TNtupleD *nt = new TNtupleD("nt","random data","x:y");
    for(int i=0; i<100000; i++) {
        double var[2];
        var[0] = rnd.Gaus(0.,1.);
        var[1] = rnd.Gaus(0.,1.)+var[0];
        nt->Fill(var);
    }

    double mean_x = 0., mean_y = 0.;
    double cov_xx = 0.,cov_xy = 0.,cov_yy = 0.;
    for(int i=0; i<nt->GetEntries(); i++) {
        nt->GetEntry(i);
        double x = nt->GetArgs()[0];
        double y = nt->GetArgs()[1];
        mean_x += x;  mean_y += y;
        cov_xx += x*x;
        cov_xy += x*y;
        cov_yy += y*y;
    }
    mean_x /= nt->GetEntries();
    mean_y /= nt->GetEntries();
    cov_xx = cov_xx/nt->GetEntries() - mean_x*mean_x;
    cov_xy = cov_xy/nt->GetEntries() - mean_x*mean_y;
    cov_yy = cov_yy/nt->GetEntries() - mean_y*mean_y;

    printf("Mean: (%g, %g)\n",mean_x, mean_y);
    printf("Covariance:\n%f, %f\n%f, %f\n",
         cov_xx, cov_xy, cov_xy, cov_yy);
}
```

Put in some correlated distributions!

You may try to remove the correlation and see the change in the off-diagonal term

```
Mean: (0.00267401, -0.000551784)
Covariance:
0.993943, 0.996197
0.996197, 1.996348
```

Time for all the lovely distributions!

BINOMIAL DISTRIBUTION (REVISIT)

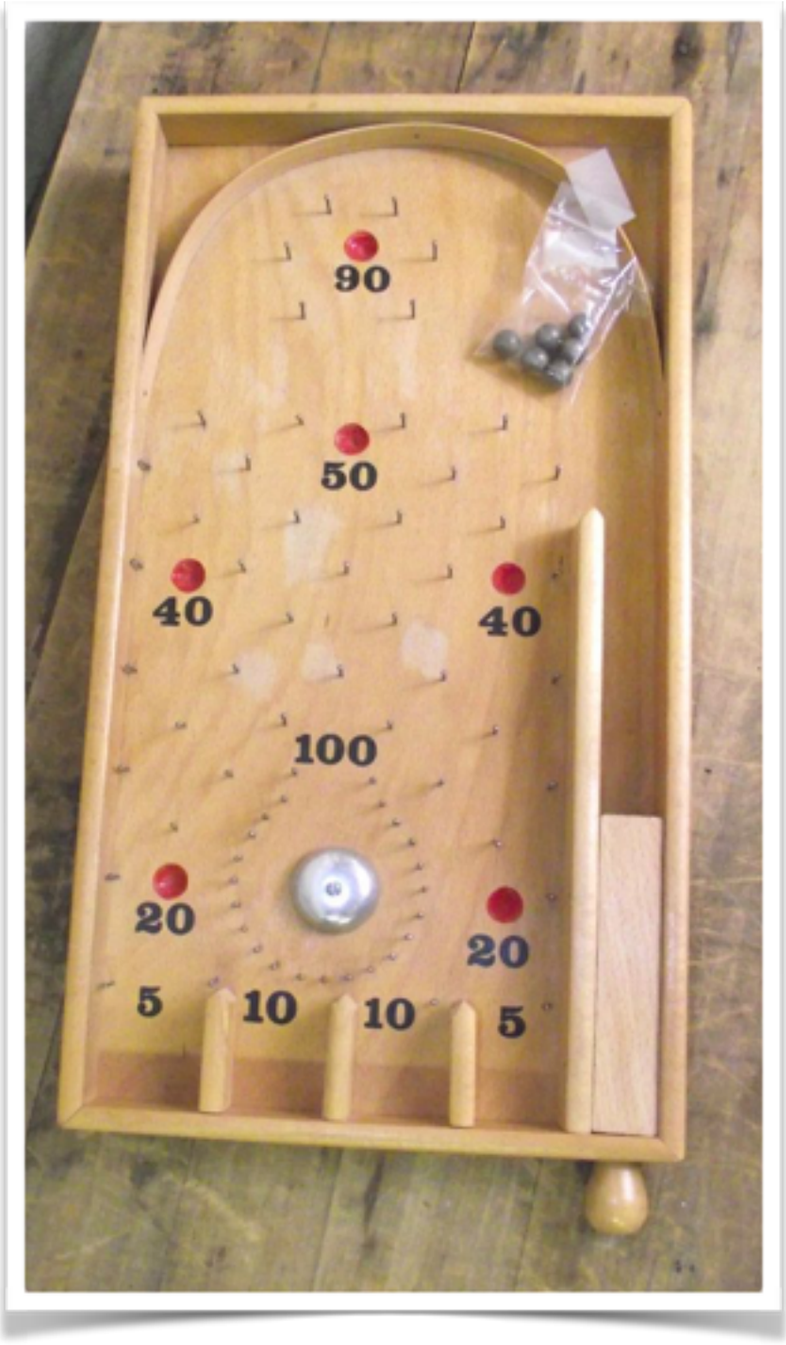
- We have already introduced the binomial distribution already. It gives the probability of finding exactly n successes in N trials, when the probability of success in each single trial is a constant p.
- The properties of the binomial distribution are
  - **variable:** a positive integer n $(0 \leq n \leq N)$
  - **parameters:** a positive integer N, a positive real number p $(0 \leq p \leq 1)$
  - **probability function:** $P(n) = \frac{N!}{n!(N-n)!} p^n (1-p)^{N-n}$
  - **expected value:** $E(n) = Np$
  - **variance:** $V(n) = Np(1-p)$

MULTINOMIAL DISTRIBUTION

- Generalization of binomial distribution to the case of multiple outcomes. It gives the probability of finding exactly n_i outcomes of type i (out of total k types, $1 \leq i \leq k$) in N independent trials, when the probability of outcome i in a single trial is p_i.
- Properties:
  - **variable:** positive integers n_i $(0 \leq n_i \leq N,\ i = 1, 2, \ldots, k)$
  - **parameters:** positive integers N, k, and positive real number p_i $(0 \leq p_i \leq 1,\ \Sigma p_i = 1)$
  - **probability function:** $P(n_1, n_2, \ldots, n_k) = \frac{N!}{n_1! n_2! \cdots n_k!} p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$
  - **expected value:** $E(n_i) = Np_i$
  - **variance:** $V(n_i) = Np_i(1-p_i)$

MULTINOMIAL DISTRIBUTION (CONT.)

- A classic pinball game is a typical example of multinomial distribution, if the total # of balls is fixed.
- As the setup given in the photo, assume the "slots" with 5 points have doubled probability comparing to the slots with 10 points, what are the expected counts and their variance for each slot, if $N = 10$?



Slot	1 (5pt)	2 (10pt)	3 (10pt)	4 (5pt)
p_i	0.333	0.167	0.333	0.167
E	3.33	1.67	1.67	3.33
V	2.22	1.39	1.39	2.22

- Surely if the # of balls is not fixed, it will follow Poisson distribution instead.

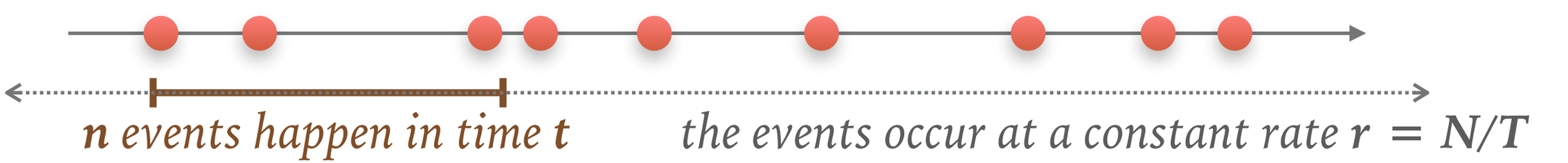
POISSON DISTRIBUTION

- The Poisson distribution gives the probability of finding exactly n events in a given **length of time** (and/or space), if the events occur independently at a constant rate.
- It is a special case of binomial distribution with $p \to 0$, $N \to \infty$, $\mu = Np$ as the finite constant; as $\mu \to \infty$, the Poisson distribution converges to the Normal distribution (Gaussian).

- Properties:
  - **variable:** positive integer n
  - **parameter:** positive real number μ
  - **probability function:** $P(n) = \frac{\mu^n e^{-\mu}}{n!}$
  - **expected value:** $E(n) = \mu$
  - **variance:** $V(n) = \mu$

Siméon Denis Poisson

POISSON DISTRIBUTION (CONT.)



- The probability to have n entries in time t, with expected value $\mu = rt$.
- It is a binomial distribution with a very large N and a very small $p = \mu/N$:

$$P(n) = \frac{N!}{n!(N-n)!}\left(\frac{\mu}{N}\right)^n\left(1-\frac{\mu}{N}\right)^{N-n}$$

$$= \frac{\mu^n}{n!}\underbrace{\frac{N(N-1)\cdots(N-n+1)}{N^n}}_{(\to 1)}\underbrace{\left(1-\frac{\mu}{N}\right)^N}_{(\to e^{-\mu})}\underbrace{\left(1-\frac{\mu}{N}\right)^{-n}}_{(\to 1)}$$

Limit N→∞

$$P(n) = \frac{\mu^n e^{-\mu}}{n!}$$

POISSON DISTRIBUTION (CONT.)

- Poisson distributions apply to various phenomena of discrete properties (*those that may happen 0, 1, 2, 3, … times during a given period of time or in a given area*) whenever the probability of the phenomenon happening is constant in time or space.
- For example:
  - number of soldiers killed by horse-kicks each year in each corps in the Prussian cavalry (*quote: L. J. Bortkiewicz*).
  - number of yeast cells used when brewing Guinness beer (*quote: W. S. Gosset*).

- And surely this works for HEP cases, like **particle decay and production**. The time interval between two successive events is actually exponentially distributed, and this is true for any Poissonian process!

SUMMING POISSONIAN VARIABLES

- Probability distribution of the sum of **two Poissonian variables** with expected values μ_1 and μ_2:

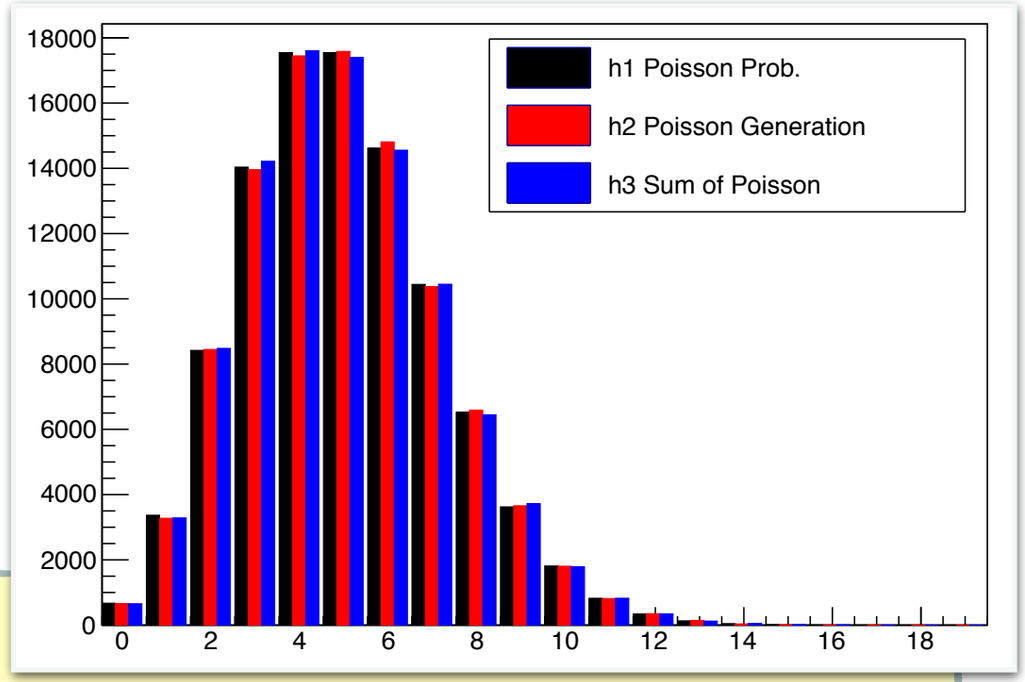
$$P'(n) = \sum_{m=0}^{n} P(m;\mu_1)P(n-m;\mu_2) = \frac{(\mu_1+\mu_2)^n}{n!}e^{-(\mu_1+\mu_2)} = P(n;\mu_1+\mu_2)$$

  The resulting distribution is still a **Poisson** with expected value $\mu_1+\mu_2$.

- This is rather useful when combining Poissonian signal and Poissonian background.
- The same conclusion holds for "*convolution*" of binomial and Poisson distributions – take a fraction of Poisson yield with a binomial "*efficiency*".
  - This is not a surprising result given the Poisson can be deduced from binomial.

PRACTICE: POISSON+POISSON

- Let's add multiple Poisson distributions together and see if the resulting distribution is also a Poisson?



example_03.cc

```
{
    TRandom3 rnd;
    const int NFILL = 100000;

    TH1D *h1 = new TH1D("h1","Poisson data",20,-0.5,19.5);
    h1->SetBarWidth(0.3);
    TH1D *h2 = (TH1D *)h1->Clone("h2");
    TH1D *h3 = (TH1D *)h1->Clone("h3");

    for(int i=0;i<20;i++) {
        double mu = 5.0;
        h1->SetBinContent(i+1,pow(mu,i)*exp(-mu)/TMath::Factorial(i)*NFILL);
    }
    for(int i=0; i<NFILL; i++) {
        int n1 = rnd.Poisson(5.0);
        h2->Fill(n1);

        int n2 = rnd.Poisson(2.5)+rnd.Poisson(1.5)+rnd.Poisson(1.0);
        h3->Fill(n2);
    }
}
```

POISSON ⊗ BINOMIAL

- Consider a Poisson distribution of expected value μ, take a total yield s_0 out of this distribution, together with a binomial efficiency ε. The probability of finding exactly s outcome events:

$$P(s_0;\mu) = \frac{e^{-\mu}\mu^{S_0}}{s_0!} \quad \otimes \quad B(s;s_0,\epsilon) = \frac{s_0!}{s!(s_0-s)!}\epsilon^s(1-\epsilon)^{s_0-s}$$

$$P'(s;\mu,\epsilon) = \sum_{s_0=s}^{\infty} P(s_0;\mu)B(s;s_0,\epsilon) = \sum_{s_0=s}^{\infty} \frac{e^{-\mu}\mu^{S_0}}{s_0!} \cdot \frac{s_0!}{s!(s_0-s)!}\epsilon^s(1-\epsilon)^{s_0-s}$$

$$= \frac{e^{-\mu}(\epsilon\mu)^s}{s!} \sum_{s_0=s}^{\infty} \frac{\mu^{s_0-s}(1-\epsilon)^{s_0-s}}{(s_0-s)!} = \frac{e^{-\mu}(\epsilon\mu)^s}{s!} \sum_{s_0=0}^{\infty} \frac{\mu^{s_0}(1-\epsilon)^{s_0}}{(s_0)!}$$

$$= \frac{e^{-\mu}(\epsilon\mu)^s}{s!} \cdot e^{\mu}e^{-\epsilon\mu} = \frac{e^{-\epsilon\mu}(\epsilon\mu)^s}{s!} = P(s;\epsilon\mu)$$

Just a Poisson with expected value of $\varepsilon\mu$.

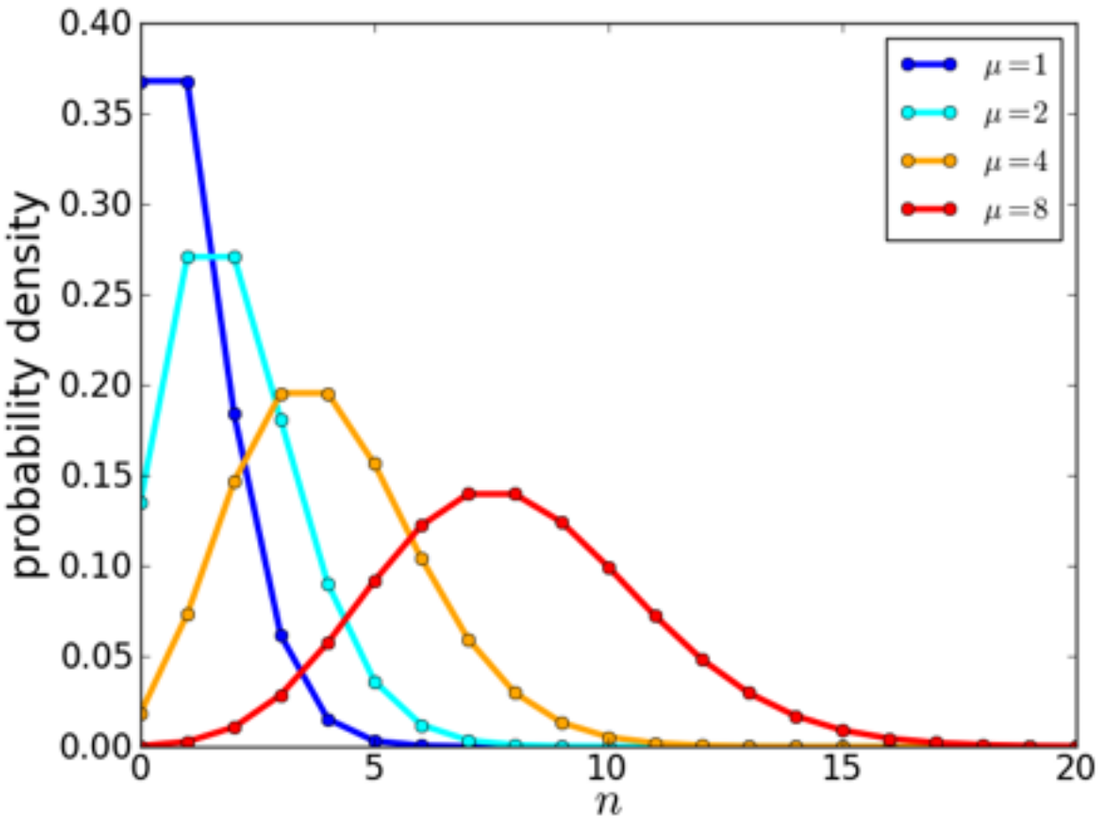
COMPOUND POISSON DISTRIBUTION

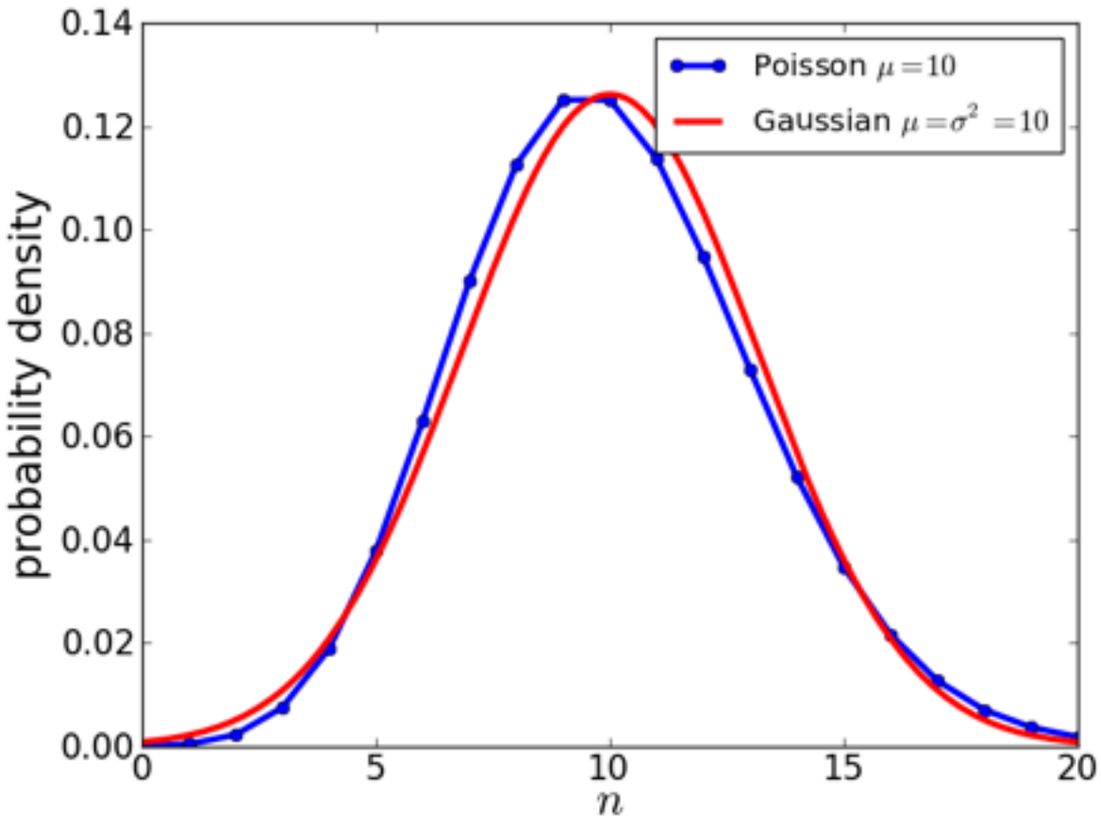
- **Compound Poisson** distribution (distribution of a branching process) is the sum of N Poisson variables n_i, all of mean μ, where N is also a Poisson variable of mean λ.

- Properties:
  - **variable:** positive integer n
  - **parameter:** positive real numbers λ, μ
  - **probability function:** $P(n) = \sum_{N=0}^{N} \left[\frac{(N\mu)^n e^{-N\mu}}{n!} \cdot \frac{\lambda^N e^{-\lambda}}{N!} \right]$
  - **expected value:** $E(n) = \lambda\mu$
  - **variance:** $V(n) = \lambda\mu(1+\mu)$

Sum of N Poisson distributions:
N fixed: Poisson
N is also Poisson: Compound Poisson

FROM POISSON TO GAUSSIAN





- As introduced earlier, when the expected value μ of the Poisson distribution increases, it converges to the **Normal distribution (Gaussian)**.
- Even the value of μ is only 10, the distribution is already rather close to a Gaussian with the same variance ($V=\sigma^2=\mu$).

NORMAL DISTRIBUTION / GAUSSIAN

- Gaussian is probably the most important / well-known / useful probability distribution.
- Properties:
  - **variable:** real number x
  - **parameter:** real numbers μ, σ
  - **probability function:**

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right]$$

  - **expected value:** $E(x) = \mu$
  - **variance:** $V(x) = \sigma^2$

Carl Friedrich Gauss

- A Gaussian distribution with $\mu=0$ and $\sigma=1$ is the **standard Normal density** function.
- A Gaussian with different σ**'s** for the left and right half of the distribution is usually called the **bifurcated Gaussian**.

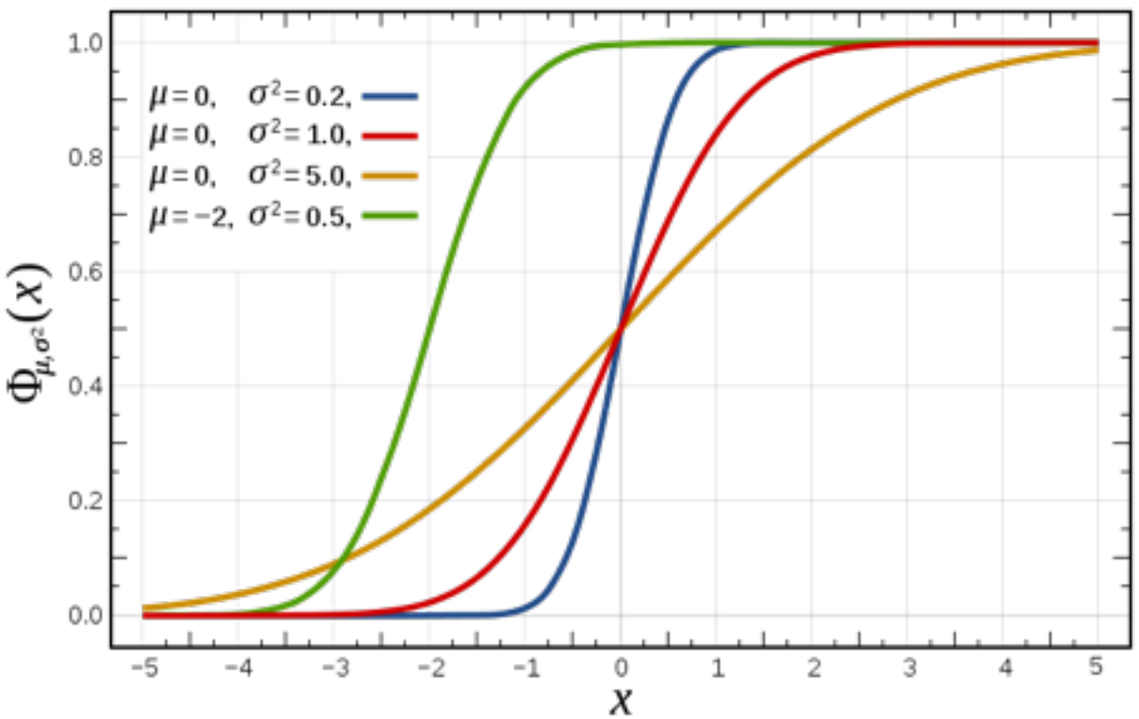
NORMAL DISTRIBUTION / GAUSSIAN (II)

- The cumulative distribution of the standard normal distribution can be related to the **error function**, **erf(x)**

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

$$\Phi(x) = \int_{-\infty}^{x} G(x'; \mu = 0, \sigma = 1) dx' = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

- The error function is what you can easy call within your program, if you want to calculate the integration of a Gaussian!

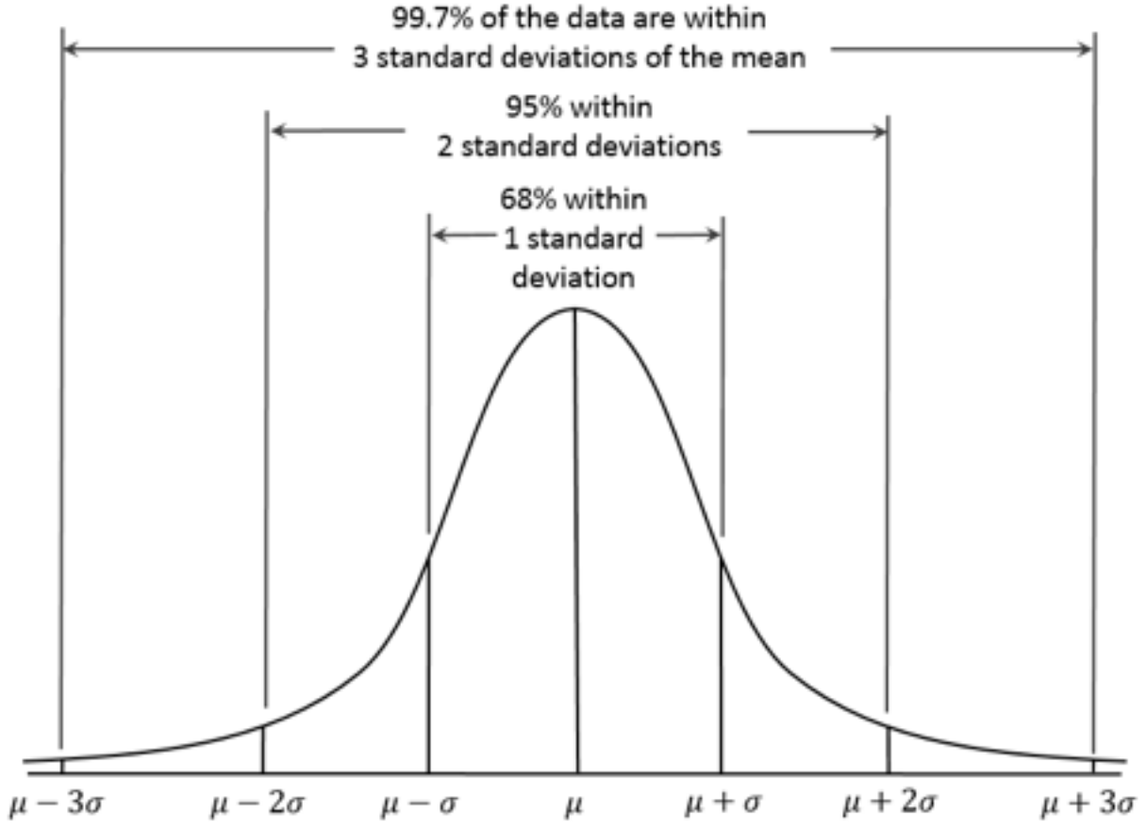


NORMAL DISTRIBUTION / GAUSSIAN (III)

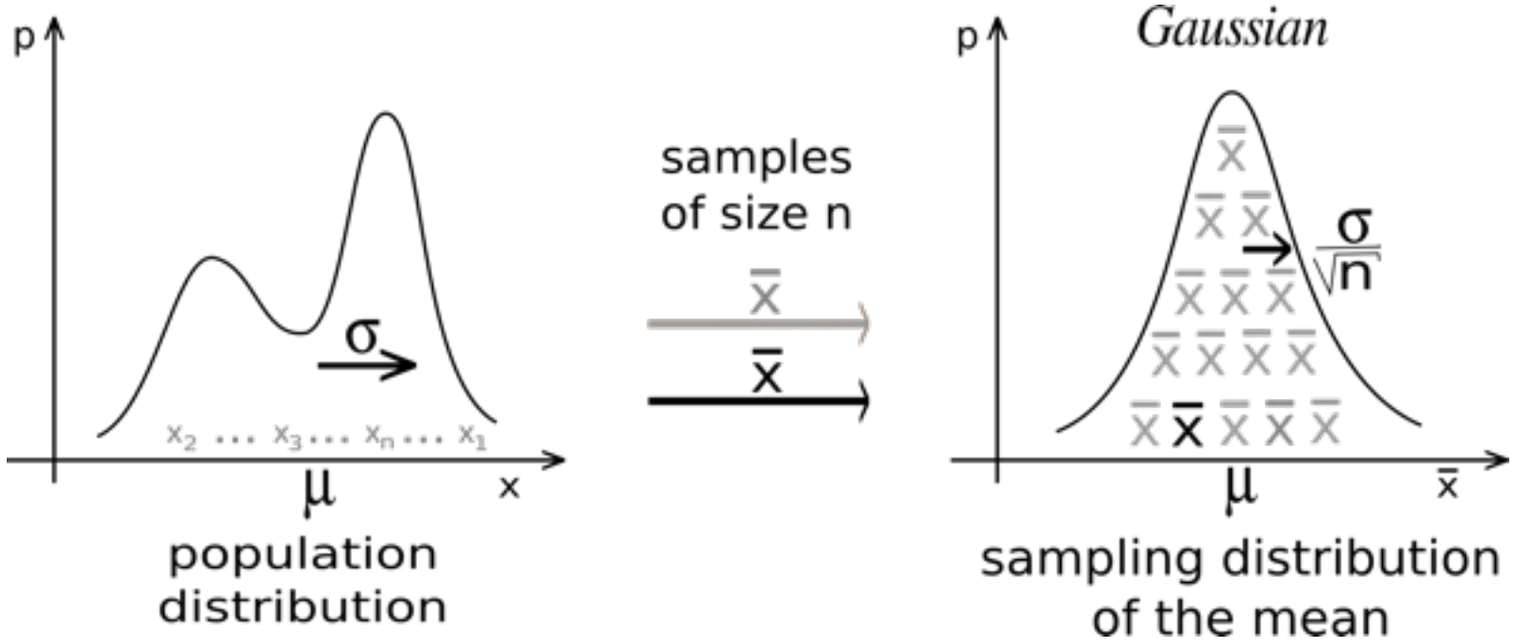
- On the other hand, the error function can be easily used to derive the **coverage probability** for a given standard deviation, e.g. 68.3% of a normal distribution is just within $\pm 1\sigma$ region, etc.

$$p(n) = \Phi(n) - \Phi(-n) = \mathrm{erf}\left(\frac{n}{\sqrt{2}}\right)$$

n	p(n)	1-p(n)
1σ	0.682 689	0.317 310
2σ	0.954 499	0.045 500
3σ	0.997 300	0.002 699
4σ	0.999 936	0.000 063



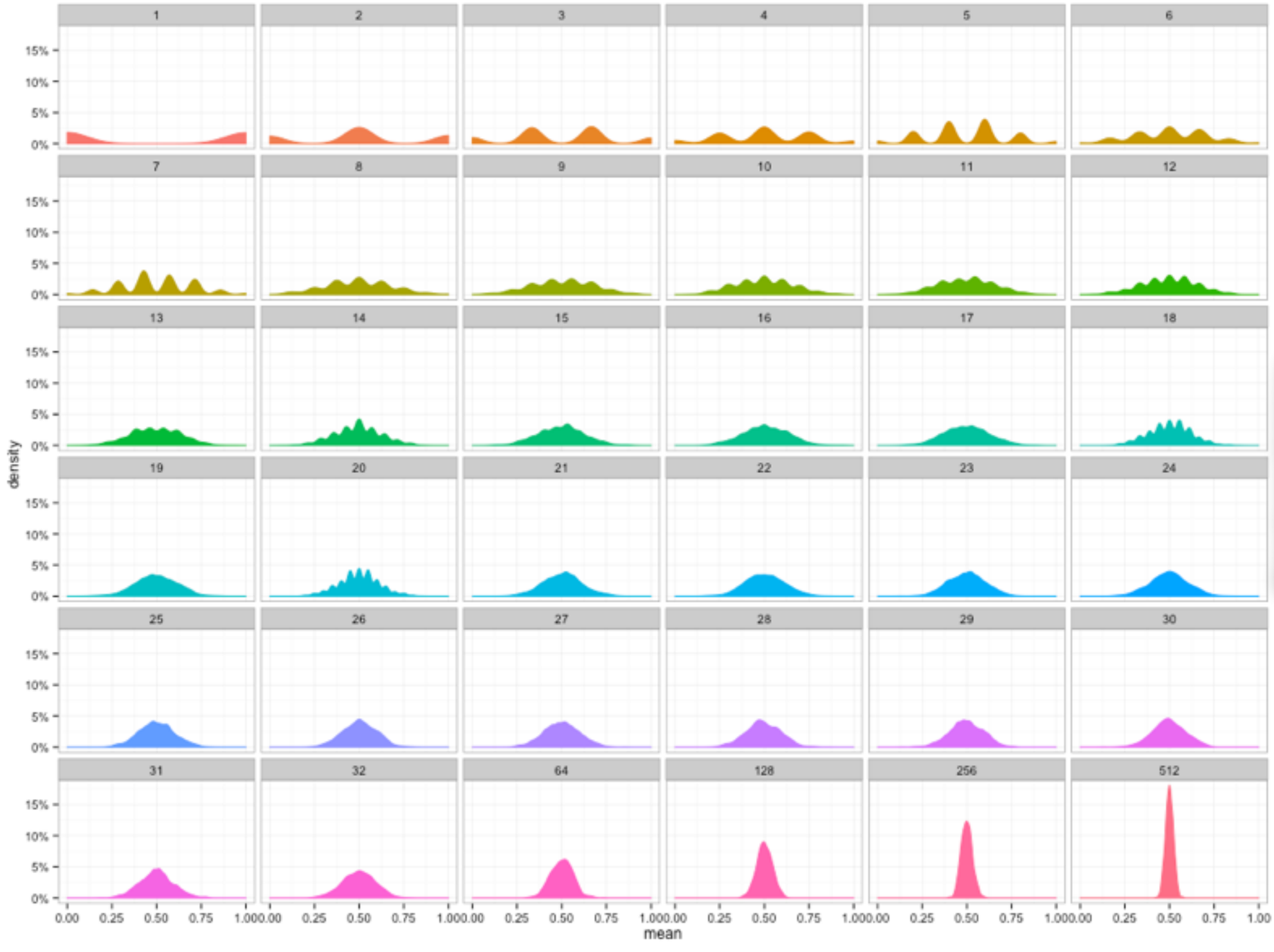
CENTRAL LIMIT THEOREM



- If we have a sequence of independent variable X_i, each from a distribution with mean μ_i and variance σ_i^2.
- The sum $S = \Sigma X_i$ will have a mean $\Sigma\mu_i$ and a variance $\Sigma\sigma_i^2$.
- This holds for **ANY distributions**, and the individual means and variances exist. The Central Limit theorem states, in the limit of large $N \to \infty$,

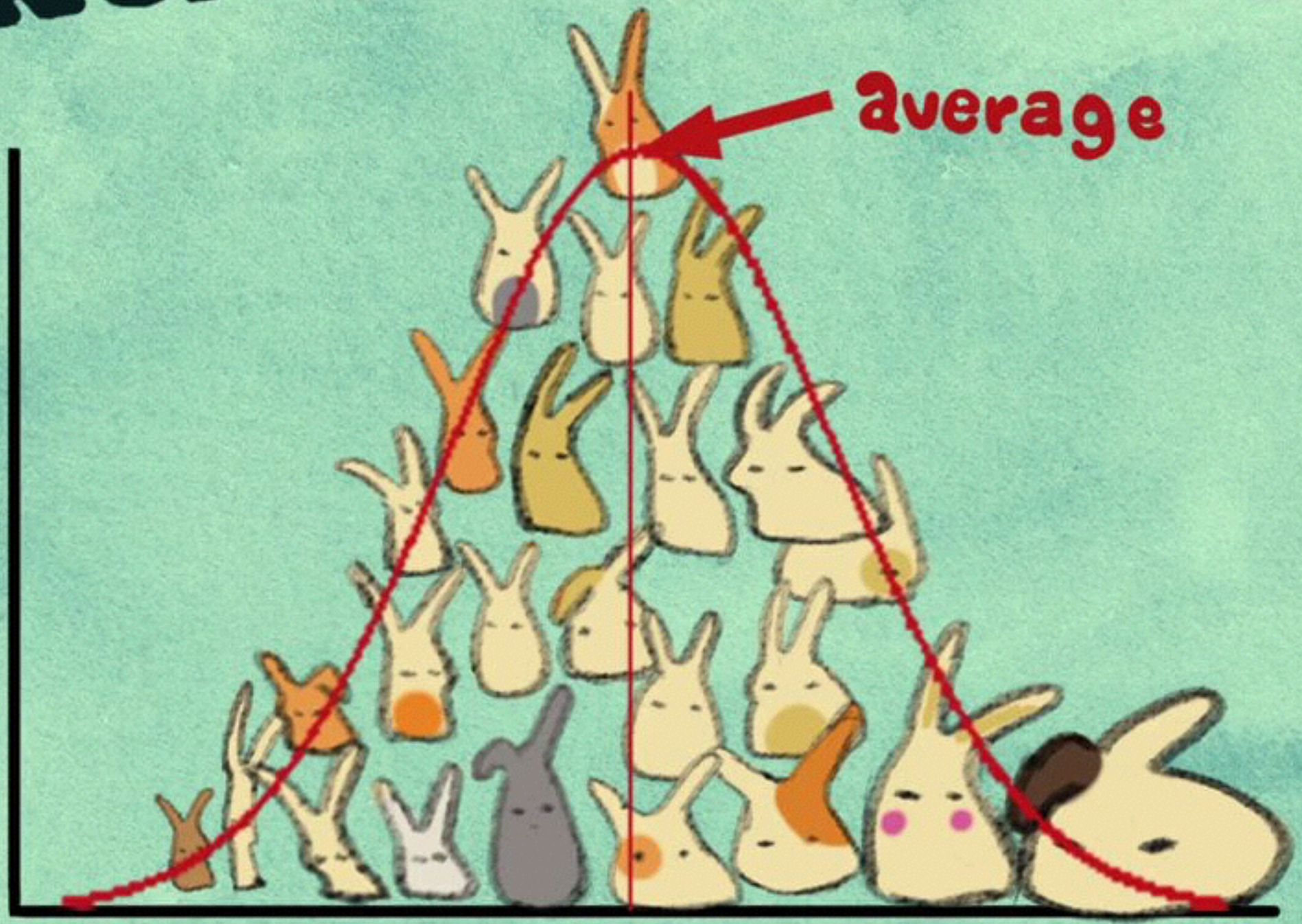
$$\frac{S - \sum_i^N \mu_i}{\sqrt{\sum_i^N \sigma_i^2}} \rightarrow \text{Gaussian}(x; \mu = 0, \sigma = 1)$$

CENTRAL LIMIT THEOREM (CONT.)



A simulation with binomial distributions up to N=512

Normal Distribution



COMBINATION OF 2 INDEPENDENT GAUSSIAN VARIABLES?

- If X and Y are two random variables, following two independent Gaussian distributions, then
  - Their sum $X+Y$ and difference $X–Y$ are also Gaussians; in fact, any linear combination of X and Y, e.g. $aX+bY$ are also Gaussian distributed.
    (Note: this is not a sum of two Gaussian PDF, but two random variables!)
  - Their product $X \times Y$ follows the **"product-normal"** distribution.
  - Their ratio X/Y follows the **Cauchy distribution** (or your familiar Breit-Wigner distribution).

MULTIVARIATE GAUSSIAN

- Multivariate Gaussian is a generalization of the 1D normal distribution to higher dimensions. It naturally takes a density function with a quadratic form in its exponent:

$$P(X) \propto \exp\left[-\frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{k} a_{ij}\left(\frac{X_i-\mu_i}{\sigma_i}\right)\left(\frac{X_j-\mu_j}{\sigma_j}\right)\right]$$

$$\propto \exp\left[-\frac{1}{2}(X-\mu)^T \cdot V^{-1} \cdot (X-\mu)\right]$$

- The quantity $(X-\mu)^T V^{-1}(X-\mu)$ is the **covariance form** of X, and it follows a x^2 distribution with k degrees of freedom. The matrix V is the covariant matrix of vector X introduced earlier:

$$V = \mathrm{cov}(X) = \begin{bmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \cdots & \rho_{1k}\sigma_1\sigma_k \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 \cdots & \vdots \\ \vdots & \vdots & \vdots \\ \rho_{1k}\sigma_1\sigma_k & \cdots & \sigma_k^2 \end{bmatrix}$$

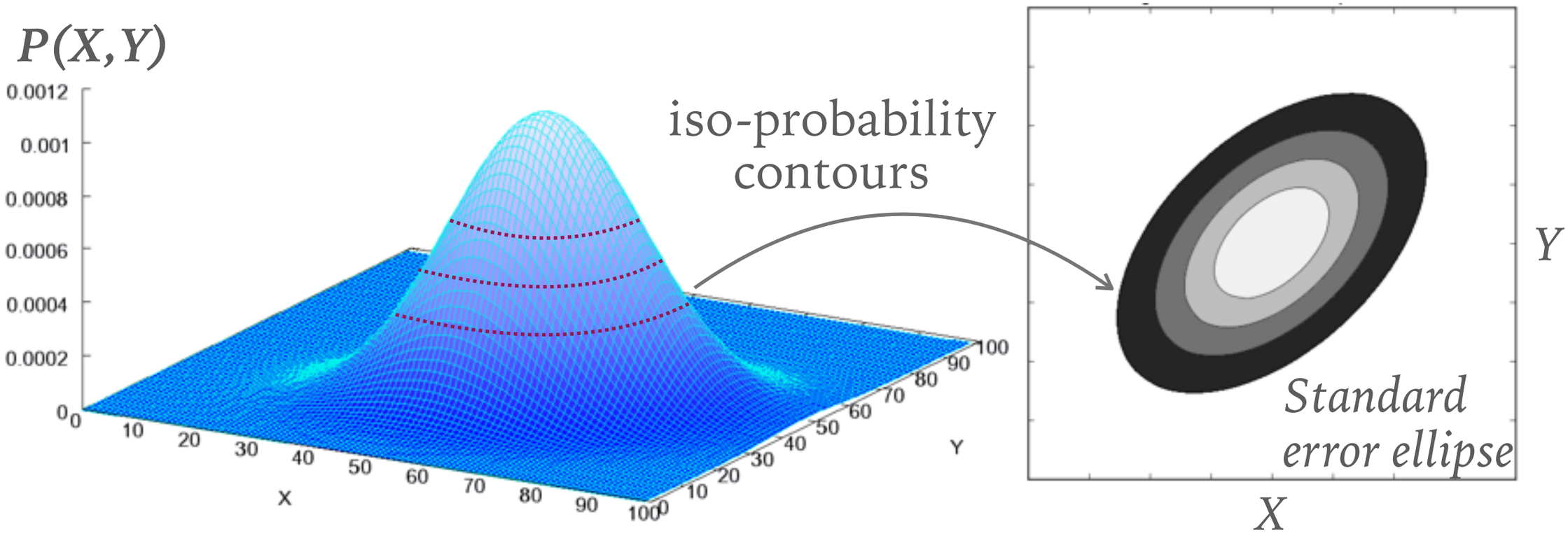
ρ_{ij} is the correlation coefficient between X_i and X_j.

JUST 2D GAUSSIAN

- Consider a simplified case of only 2D, ***X*** and ***Y***:

$$P(X,Y) = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} \times \exp\left\{-\frac{1}{2(1-\rho^2)}\left[\frac{(X-\mu_X)^2}{\sigma_X^2} + \frac{(Y-\mu_Y)^2}{\sigma_Y^2} - \frac{2\rho(X-\mu_X)(Y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right\}$$

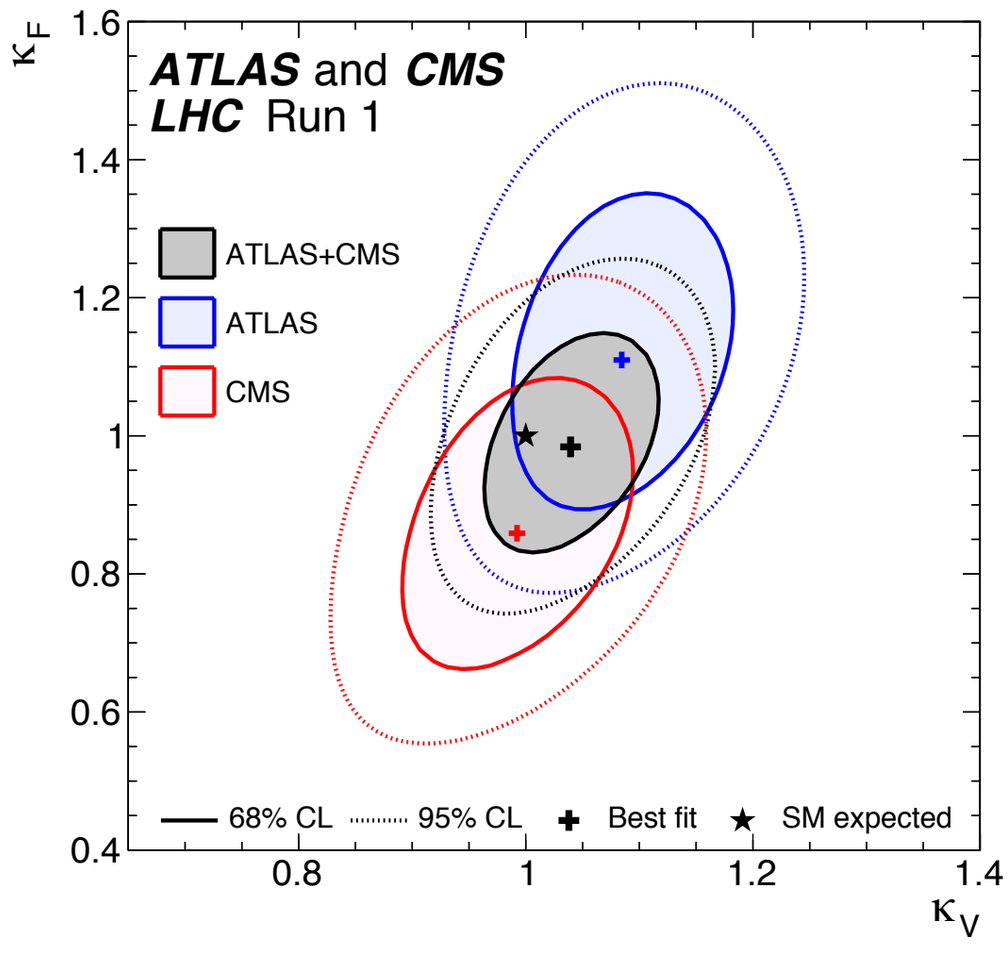
again ρ is the correlation coefficient between ***X*** and ***Y***.



ISO-PROBABILITY 2D CONTOUR

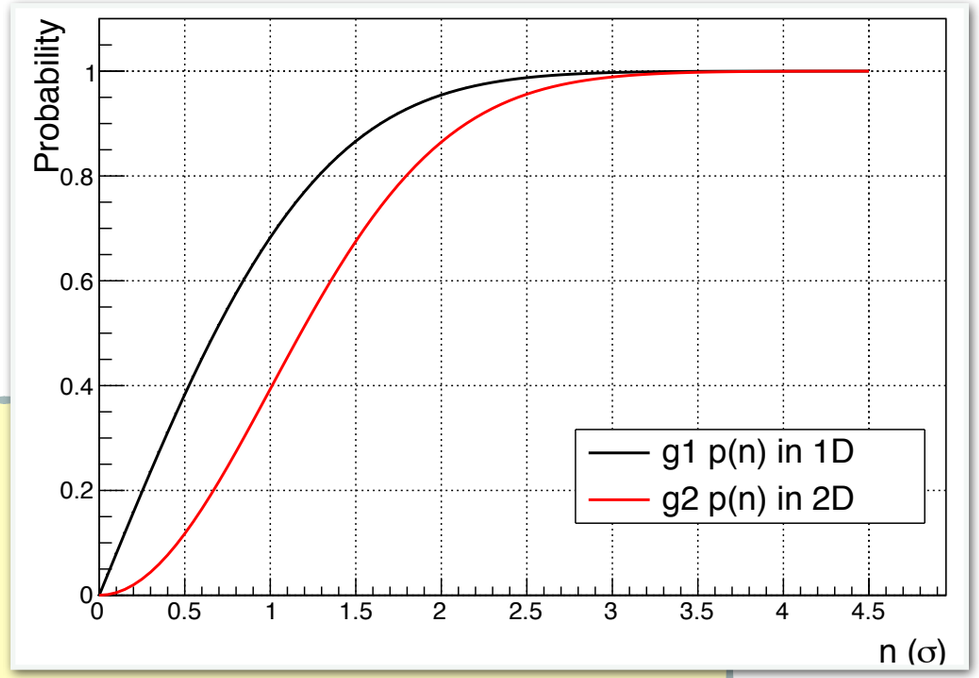
- **Your 1σ is not my 1σ**: it is mandatary to remember the conversion between # of σ's and the converging probability depends on the number of dimensions.
- You might notice that some of the 2D contour plots put 68%/95%, instead of 1σ/2σ, since 1σ in 2D does not cover the conventional 68% coverage probability.

n	p(n) in 1D	p(n) in 2D
1σ	0.6827	0.3934
2σ	0.9545	0.8647
3σ	0.9973	0.9889
1.515σ		0.6827
2.486σ		0.9545
3.439σ		0.9973



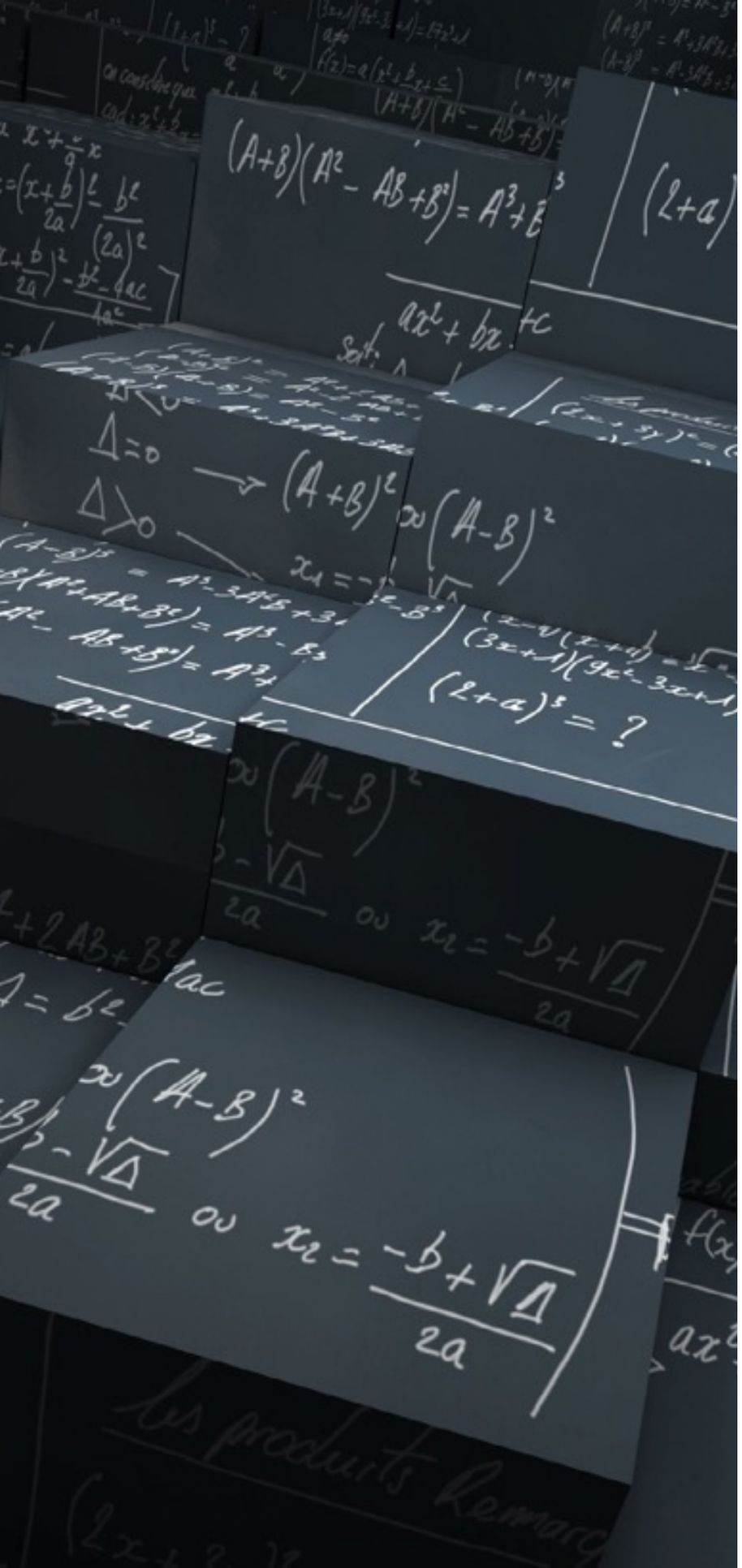
PRACTICE: PROBABILITY VERSUS N-SIGMA

- In fact it is easy to calculate this conversion table by yourself. The **TMath::Prob()** function can do it quickly.
- This is a simple example practice to draw the probability as a function of # of σ in 1D and 2D:



example_04.cc

```
{
    vector<double> vec, prob1, prob2;
    double sigma;
    while(sigma<4.5) {
        vec.push_back(sigma);
        prob1.push_back(1.-TMath::Prob(sigma*sigma,1));
        prob2.push_back(1.-TMath::Prob(sigma*sigma,2));
        sigma += 0.05;
    }

    TCanvas *c1 = new TCanvas();
    TGraph *g1 = new TGraph(vec.size(),vec.data(),prob1.data());
    g1->Draw();
    TGraph *g2 = new TGraph(vec.size(),vec.data(),prob2.data());
    g2->Draw("same");
}
```



OTHER COMMONLY USED FUNCTIONS

- Exponential
- Power law
- Chi-square distribution
- Cauchy/Briet-Wigner distribution
- Log-Normal distribution
- Landau distribution
- Crystal Ball function
- ARGUS function
- Threshold function
- Polynomials: Laurent/Legendre/ Chebyshev/Bernstein/…

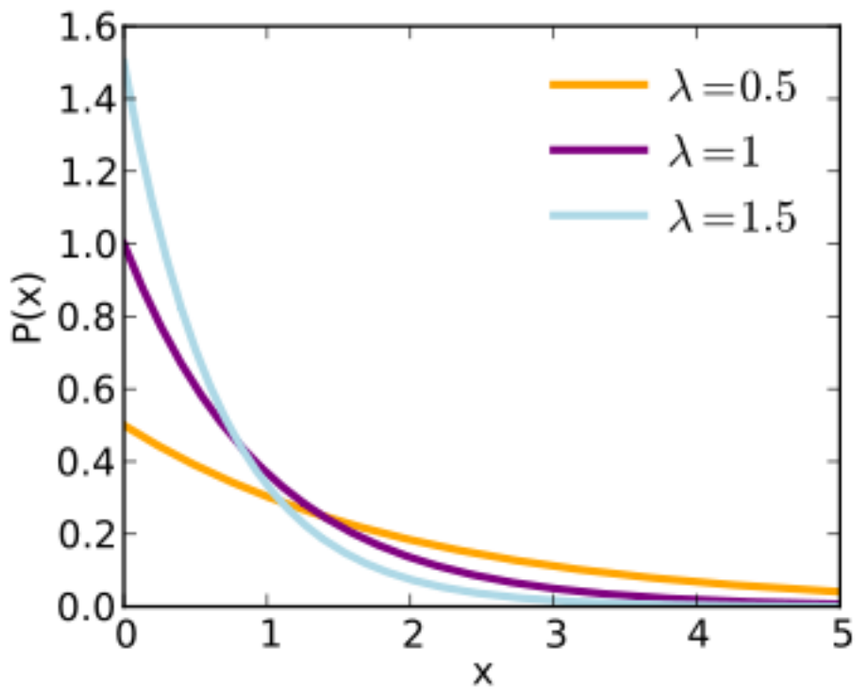
EXPONENTIAL DISTRIBUTION

- Consider events occurring randomly in time, with an average of λ events per unit time.
- The Poisson distribution describe the probability of N events occurring in a time interval t.

$$P(N|t) = \frac{1}{N!}(\lambda t)^N e^{-\lambda t}$$

- Then the **probability of no events in time t** follows the exponential distribution **$\exp(-\lambda t)$**.

- Properties:
  - **variable:** real number x
  - **parameter:** real numbers λ
  - **probability function:** $P(x) = \lambda e^{-\lambda x}$
  - **expected value:** $E(x) = 1/\lambda$
  - **variance:** $V(x) = 1/\lambda^2$



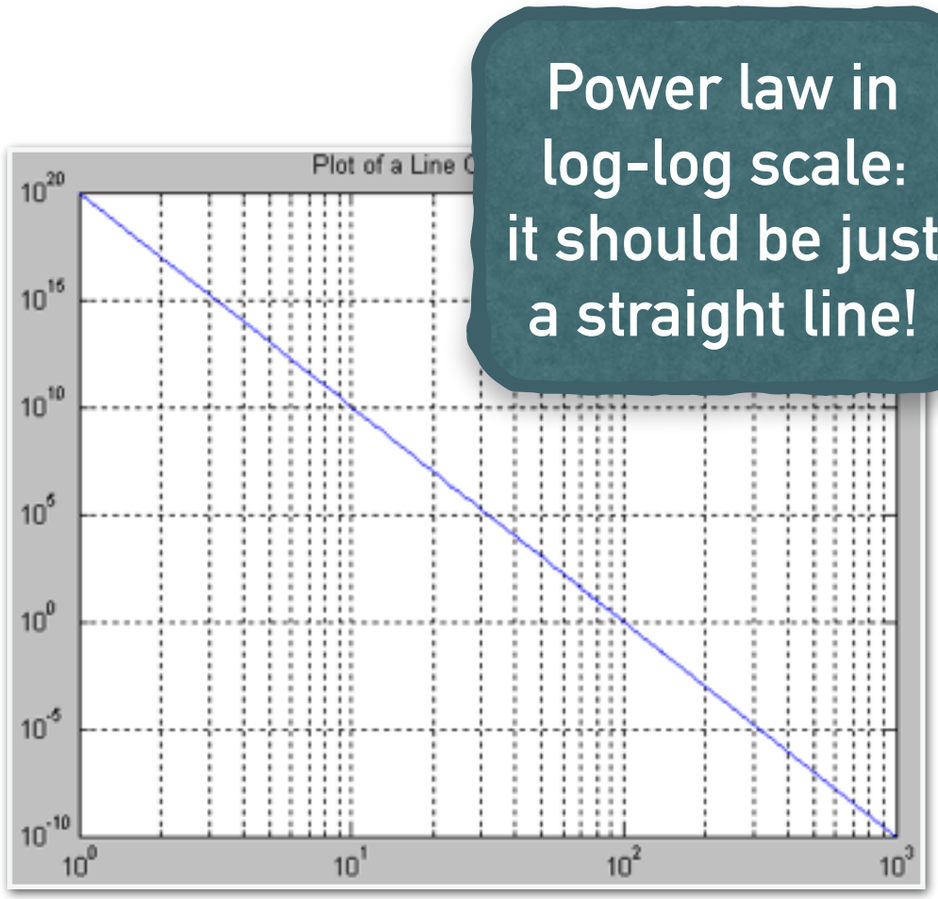
POWER LAW FUNCTION

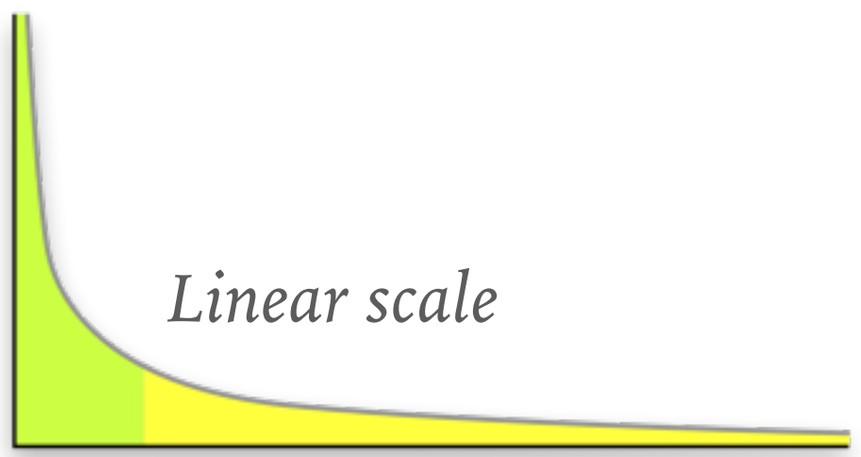
- **Power law function** is also kind of fast increasing/decreasing function commonly used in many places:

$$P(x) \propto x^{-k}$$

where ***k*** is a constant parameter of the distribution known as the **exponent** or **scaling parameter**.

- A comparison between exponential and power law (*and you can see they are actually very different!*)
  - Exponential: $P(x) = \text{const}^{x}$
  - Power law: $P(x) = x^{\text{const}}$





CHI-SQUARE DISTRIBUTION

- Suppose that X_1, X_2, … X_N are independent, **standard Normal variables**, The sum of the their squares

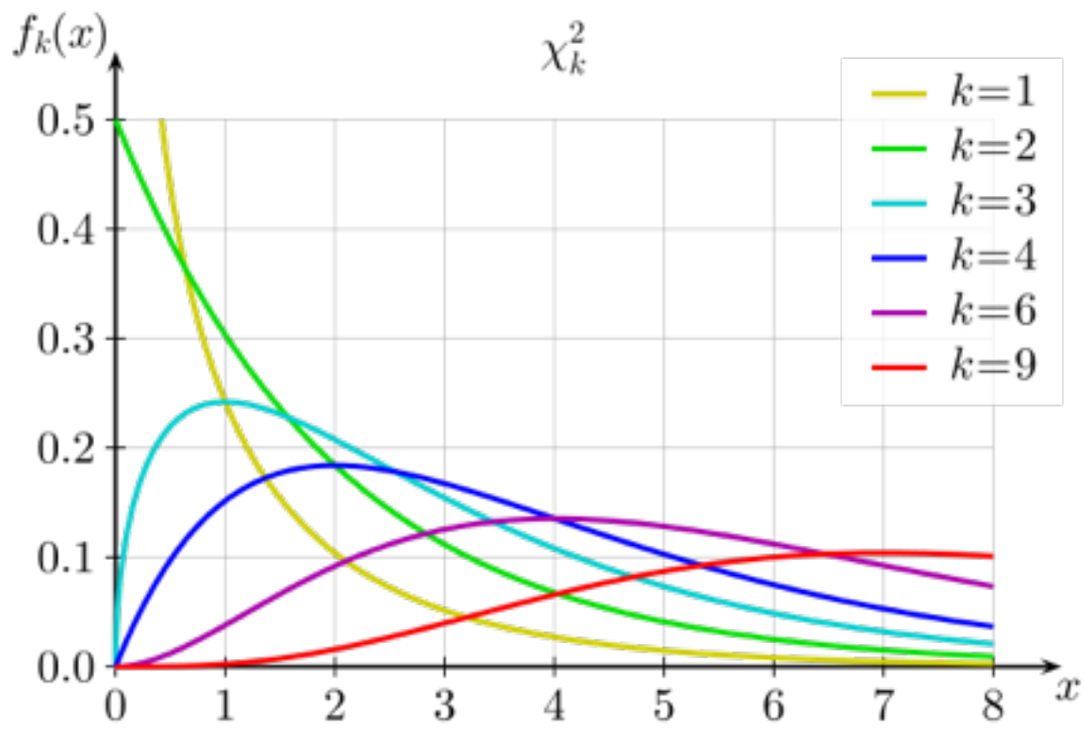
$$Q = \sum_{i=1}^{N} X_i^2 \;\Rightarrow\; \chi^2(N)$$

  follows a chi-square distribution $\chi^2(N)$, with N-degrees of freedom.

- Properties:
  - **variable:** real number x
  - **parameter:** positive integer N (*as "degrees of freedom"*)
  - **probability function:**

$$P(x) = \frac{\frac{1}{2}\left(\frac{x}{2}\right)^{(N/2)-1} e^{-x/2}}{\Gamma(N/2)}$$

  - **expected value:** $E(x) = N$
  - **variance:** $V(x) = 2N$

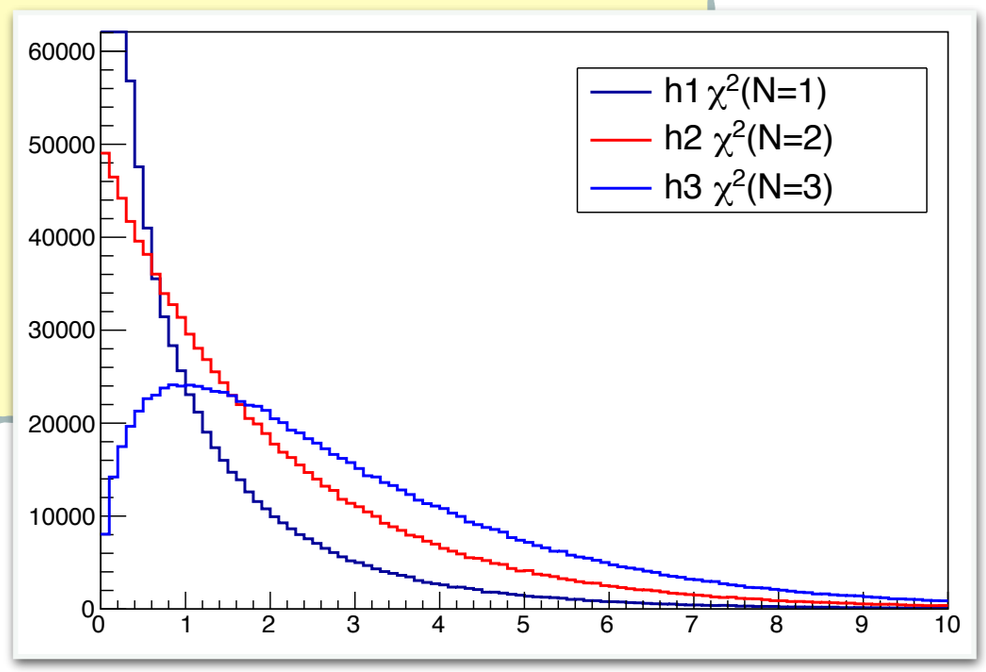


PRACTICE: THE PRINCIPLE OF CHI-SQUARE DISTRIBUTION

- Let's just add multiple Gaussian random variables and see if the resulting distribution follows the chi-square distributions.

example_05.cc

```
{
    TRandom3 rnd;
    TH1D *h1 = new TH1D("h1","chisquare data",100.,0.,10.);
    TH1D *h2 = (TH1D*)h1->Clone("h2");
    TH1D *h3 = (TH1D*)h1->Clone("h3");

    for(int i=0;i<1000000;i++) {
        h3->Fill(pow(rnd.Gaus(),2)+pow(rnd.Gaus(),2)+
                 pow(rnd.Gaus(),2));
        h2->Fill(pow(rnd.Gaus(),2)+pow(rnd.Gaus(),2));
        h1->Fill(pow(rnd.Gaus(),2));
    }
    h1->Draw();
    h2->SetLineColor(kRed);
    h2->Draw("same");
    h3->SetLineColor(kBlue);
    h3->Draw("same");
}
```



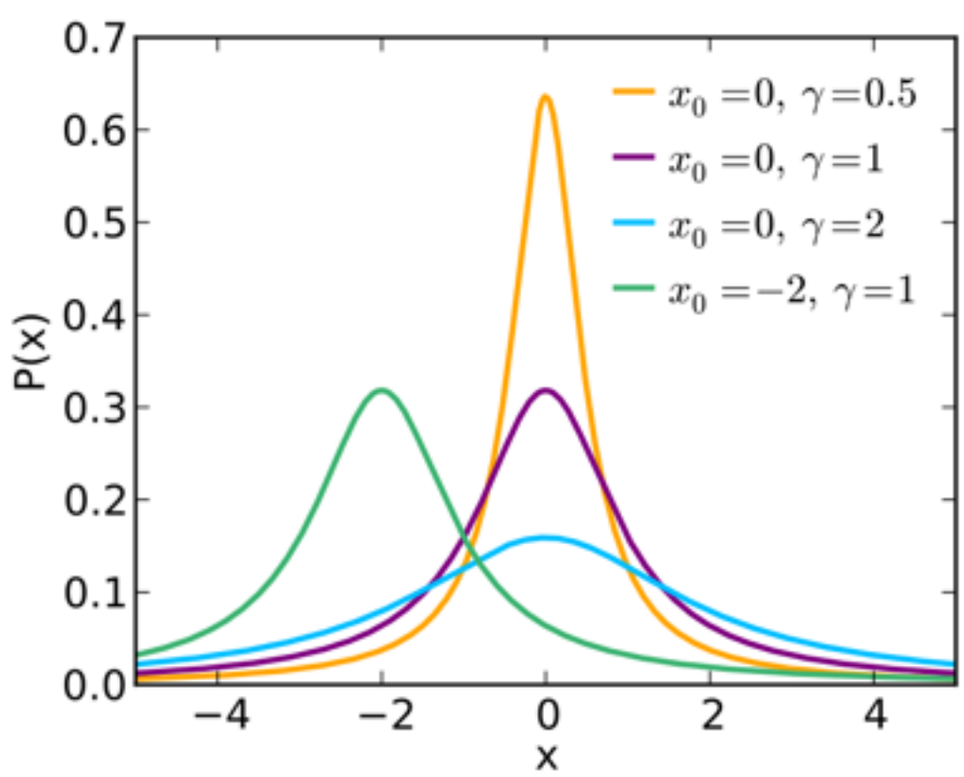
See if the output distributions agree with the curves given in the previous slide!

CAUCHY/BRIET-WIGNER DISTRIBUTION

- The Cauchy distribution is often used in statistics as the canonical example of a "*pathological*" distribution since both its expected value and its variance are undefined.
- It is identical to the physically important Breit-Wigner distribution.
- Properties:
  - **variable:** real number ***x***
  - **parameter:** BW-function has a location parameter and a scale parameter are included. (Note: the expected value and variance are still undefined!)
  - **probability function:**

$$P(x) = \frac{1}{\pi}\left[\frac{\Gamma}{\Gamma^2 + (x - x_0)^2}\right]$$

  - **expected value, variance:** ***undefined***



UNDEFINED MEAN? WHY?

- You might get surprised why the Cauchy distribution does not have a definite mean (and variance), even it has an obvious median point at the middle. This has to come back to the definition of mean:

$$\mu = E(X) = \langle X \rangle = \int_{\Omega} X f(X) dx$$

- Consider a Cauchy (or B-W function with $x_0=0$, $\Gamma=1$), by definition:

$$\mu = \int_{-\infty}^{+\infty} \frac{1}{\pi} \frac{x}{1+x^2} dx \quad \textit{This is a typical improper integral}$$

Thus

$$\mu = \lim_{L \to -\infty} \lim_{H \to +\infty} \int_{L}^{H} \frac{1}{\pi} \frac{x}{1+x^2} dx \quad \textit{or} \quad \mu = \lim_{H \to +\infty} \lim_{L \to -\infty} \int_{L}^{H} \frac{1}{\pi} \frac{x}{1+x^2} dx$$

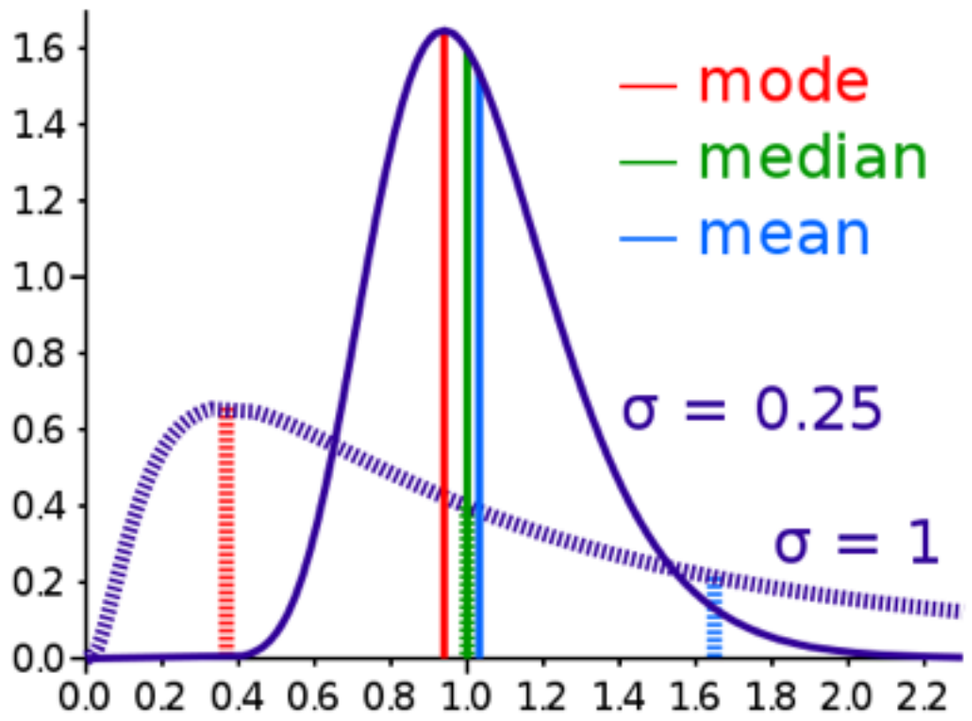
The two evaluations do not give the same finite result since the inner limit already diverges.

LOG-NORMAL DISTRIBUTION

- A log-normal distribution is a continuous probability distribution of a random variable whose logarithm is normally distributed, ie. if X is log-normally distributed, then $Y=ln(X)$ has a normal distribution.
- A log-normal process is the statistical realization of the multiplicative product of many independent positive random variables.
- Properties:
  - **variable:** real number x
  - **parameter:** real numbers μ, σ
  - **probability function:**

$$P(x) = \frac{1}{x\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right]$$

  - median: $\exp(\mu)$
  - mode(=maximum point): $\exp(\mu-\sigma^2)$
  - expected value: $E(x) = \exp(\mu+\sigma^2/2)$
  - variance: $V(x) = [\exp(\sigma^2)-1]\exp(2\mu+\sigma^2)$



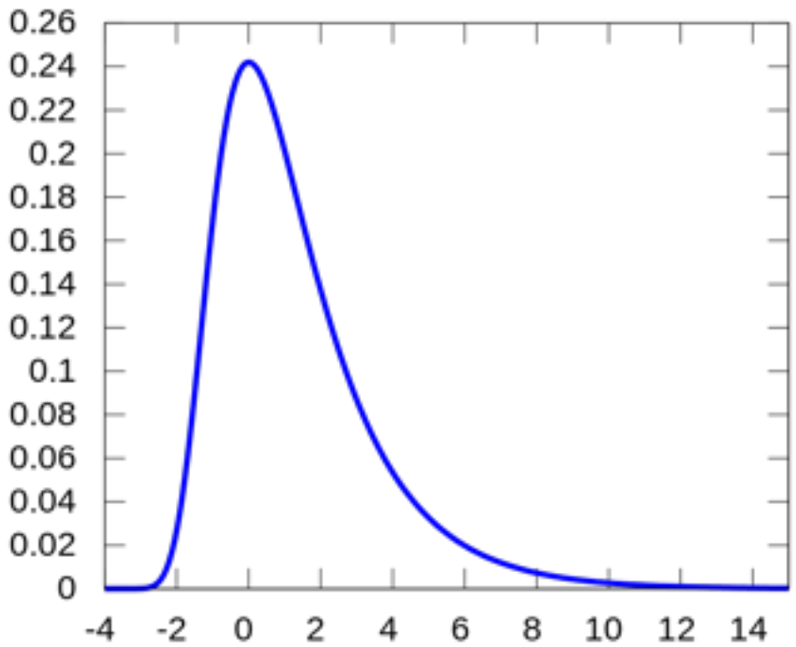
LANDAU DISTRIBUTION

- Widely used to model the fluctuations in the energy loss of particles passing though thin layers.
- Charged particles (*protons, pions, etc.*) which are in most cases close to MIPs, all produce approximately Landau-distributed spectra when traversing the matter.
- Because of the distribution's long tail, the moments of the distribution, like mean or variance, are undefined.
- Probability function:

$$P(x) = \frac{1}{\pi} \int_0^\infty \exp(-t \ln t - xt) \sin(\pi t) dt$$

Usual shift/scale applied:

$$P'(x; \mu, \sigma) = P\left(\frac{x - \mu}{\sigma}\right)$$

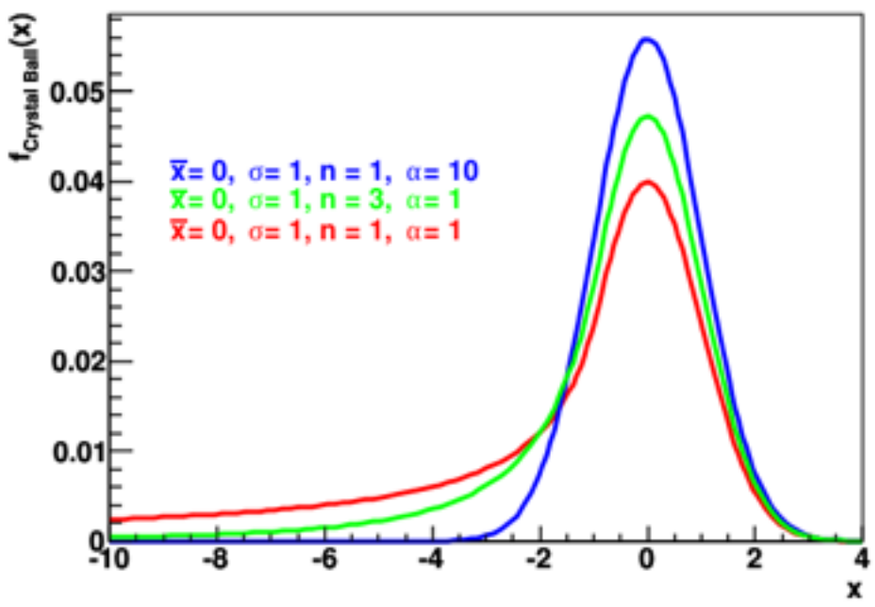


CRYSTAL BALL FUNCTION

- **Crystal Ball function** consists of a Gaussian core and a power-law low-end tail, below a certain threshold $\mu - a\sigma$.
- Named after the Crystal Ball collaboration at SLAC.
- Mostly used to describe the processes with strong energy lost and with a long tail to the left. For example, invariant mass of particles with photon in the final state.
- Probability function:

$$P(x) = N \cdot \begin{cases} A \cdot (B - \frac{x-\mu}{\sigma})^{-n}, & \text{if } x < \mu - \alpha\sigma \\ \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], & \text{if } x \geq \mu - \alpha\sigma \end{cases}$$

$$A = (n/\alpha)^n \exp(-\alpha^2/2), \quad B = n/\alpha - \alpha$$



ARGUS FUNCTION

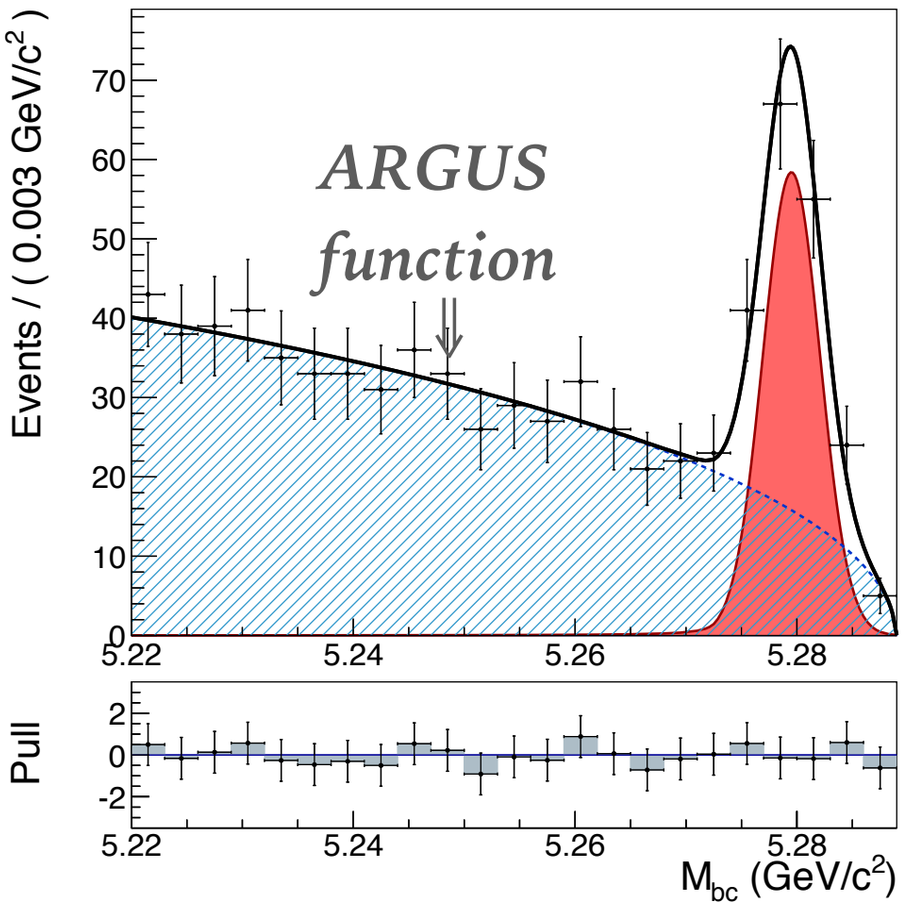
- The **ARGUS distribution** is the probability distribution to model the invariant mass distribution of a "continuum" background, in particular, near the kinematic threshold given by the beam energy.
- Named after the ARGUS experiment.
- Widely used by the B-factories *(or any similar collider experiment with fixed beam energy)*.
- Probability function:

$$P(x) = Nx\sqrt{1-\left(\frac{x}{\theta}\right)^2}\exp\left\{-\frac{\xi^2}{2}\left[1-\left(\frac{x}{\theta}\right)^2\right]\right\}$$

N: normalization factor;

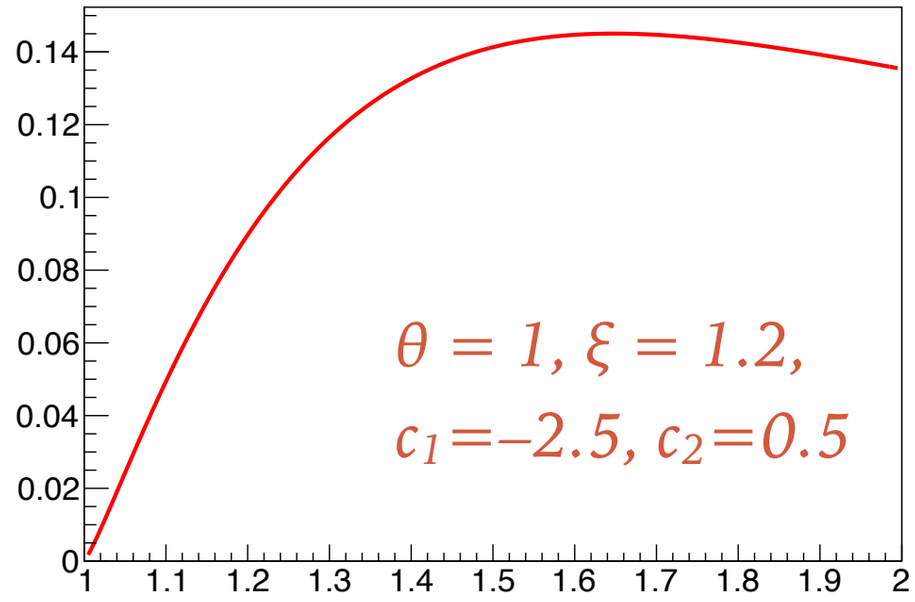
θ: kinematic upper bound or beam energy (fixed);

ξ: shape parameter



THRESHOLD FUNCTION (MN_FIT VER?)

- A “threshold function” usually refers to a step function with a “turn-on” threshold. Here we are going to discuss something very different.
- Interestingly this is not fully documented, but a very old fitting tool named mn_fit had introduced a convenient “threshold function” to model the distribution near a kinematic boundary.
- Probability function:



$$P(x) = N \cdot (x-\theta)^{\xi} \exp\left[c_1 * (x-\theta) + c_2 * (x-\theta)^2\right]$$

N: normalization factor;

θ: kinematic upper bound or beam energy (again, as a fixed parameter!);

ξ, c_1, c_2: shape parameters

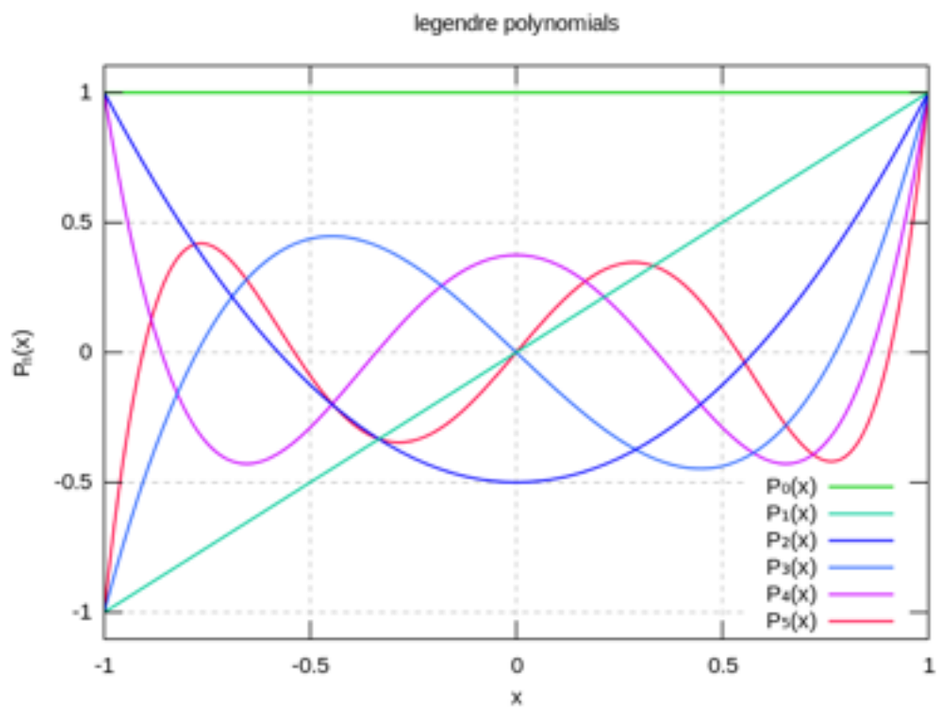
POLYNOMIALS ET AL

- Polynomials are probably the simplest way to model any unknown distributions. Although different definitions of polynomials are mathematically equivalent, but different polynomials indeed have different behavior.
- In particular, some of the polynomials (e.g. Legendre or Chebyshev) are orthogonal, they usually have a better behavior when expanding the order of polynomials.
- Simple polynomials:
  - **Power series:** $a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \cdots = \sum_{k=0}^{N} a_k x^k$
  - **Laurent polynomial:** same as above but $\boldsymbol{k}$ can be negative.

POLYNOMIALS ET AL (II)

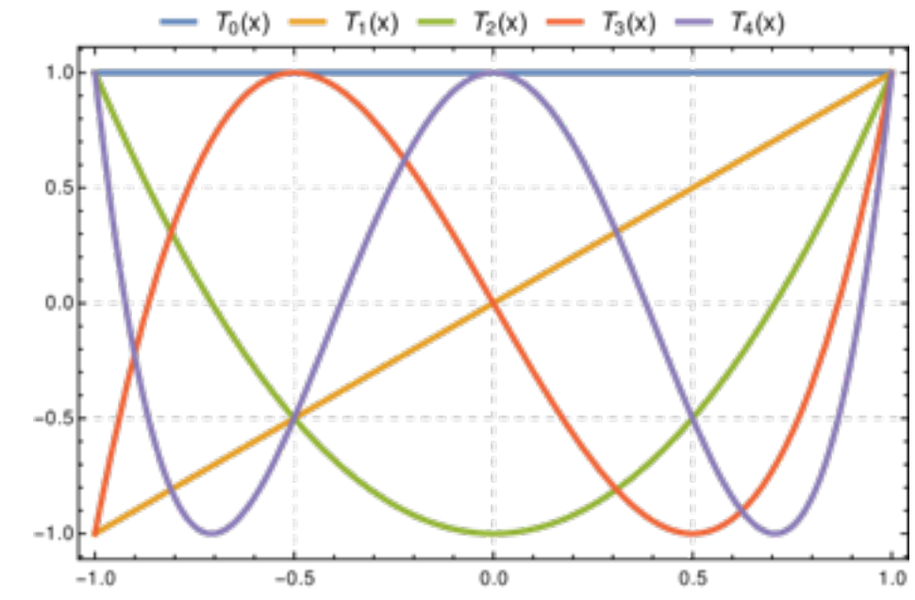
- **Legendre polynomials:** as general solutions to Legendre's Equation, and are azimuthally symmetric.

$$P_0(x) = 1, \quad P_1(x) = x$$
$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x)$$



- **Chebyshev polynomials:** as a sequence of orthogonal polynomials and can be defined recursively.

$$T_0(x) = 1, \quad T_1(x) = x$$
$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$



POLYNOMIALS ET AL (III)

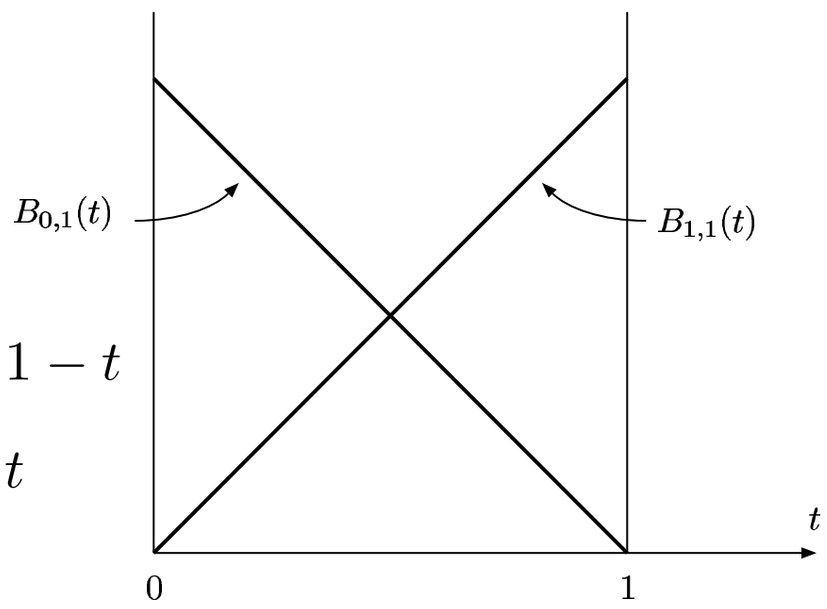
- Probabilities should be always "positive defined", but this is not the case for usual power-series based polynomials. The function can easily go to negative and break the evaluation of probability.
- **Bernstein polynomials** are constructed with sets of non-negative bases and are generally convenient for PDF modeling.
- Bernstein polynomials of degree n are defined by

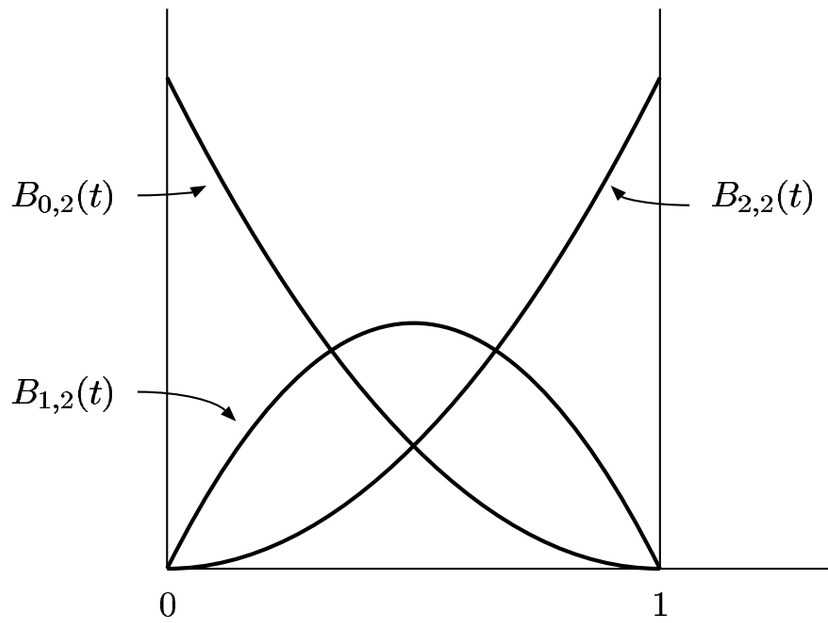
$$B_{i,n}(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{(n-i)} \quad (0 \le t \le 1)$$

- Examples:

$$B_{0,1}(t) = 1 - t$$

$$B_{1,1}(t) = t$$





$$B_{0,2}(t) = (1-t)^2$$

$$B_{1,2}(t) = 2t(1-t)$$

$$B_{2,2}(t) = t^2$$

If one function cannot describe your data, you can combine more!

JOINT MULTIPLE FUNCTIONS TOGETHER

- It is not surprising that a single function cannot fully describe your data in a more general case. One of the straightforward ways to improve the modeling is to joint multiple functions into a single PDF.
- No matter how complicated construction of the model, as a PDF, the overall normalization should be always held:

$$P(X) = \sum_i f_i P_i(X)$$

$P_i(x)$: *individual model*

f_i: *coefficient of each model*

$$\int_\Omega P(X)dX = 1$$

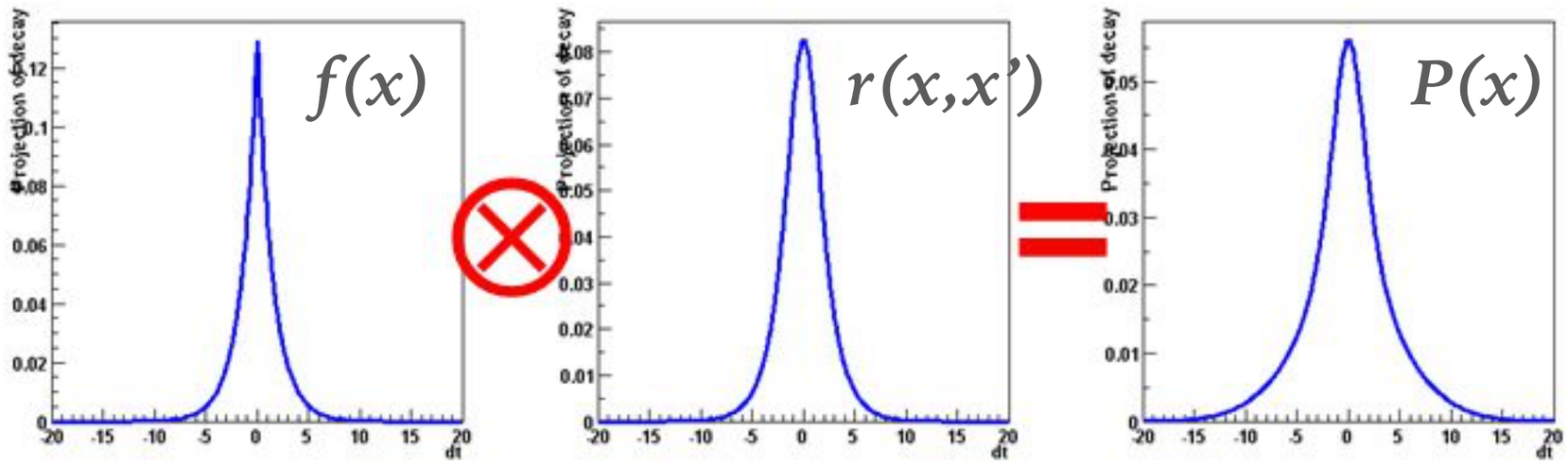
- If the normalization is not properly calculated, it might be resulting a biased parameter estimation (to be discussed in the next lecture).

CONVOLUTION

- Convolution is a typical way to add "smearing" to your given distribution. For example, adding detector resolution to a known PDF.
- Consider an intrinsic/truth PDF, $f(x)$, together with a resolution model $r(x,x')$ which gives the probability of measuring x' out of a true value of x.
- If the resolution function r is a Gaussian, the σ is the experimental resolution.
- Then the joint PDF which includes both intrinsic information and experimental resolution is to convolute f with r:

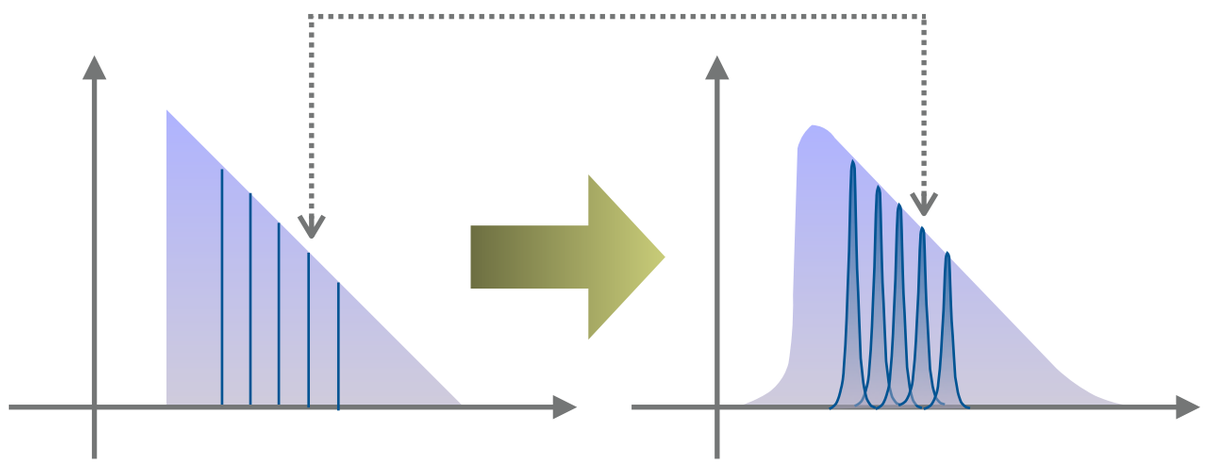
$$P(x) = f \otimes r = \int_{\Omega} f(x') \cdot r(x, x')dx'$$

For example, invariant mass distribution of a narrow particle

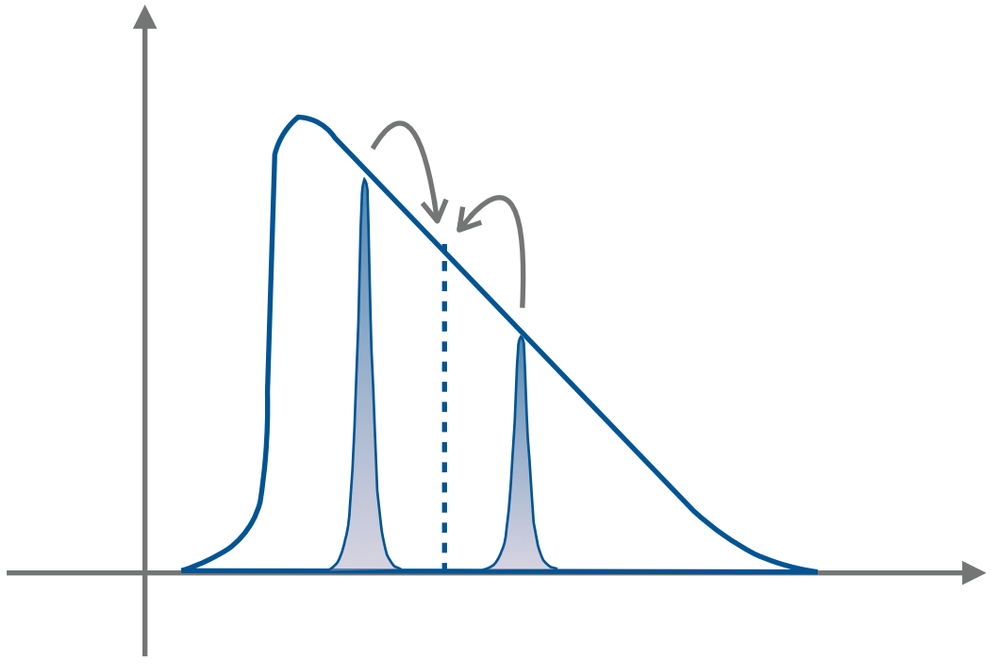


CONVOLUTION (CONT.)

- Consider a convolution with Gaussian model, the convoluted distribution can be interpreted by replacing each slice of the original distribution by a Gaussian.

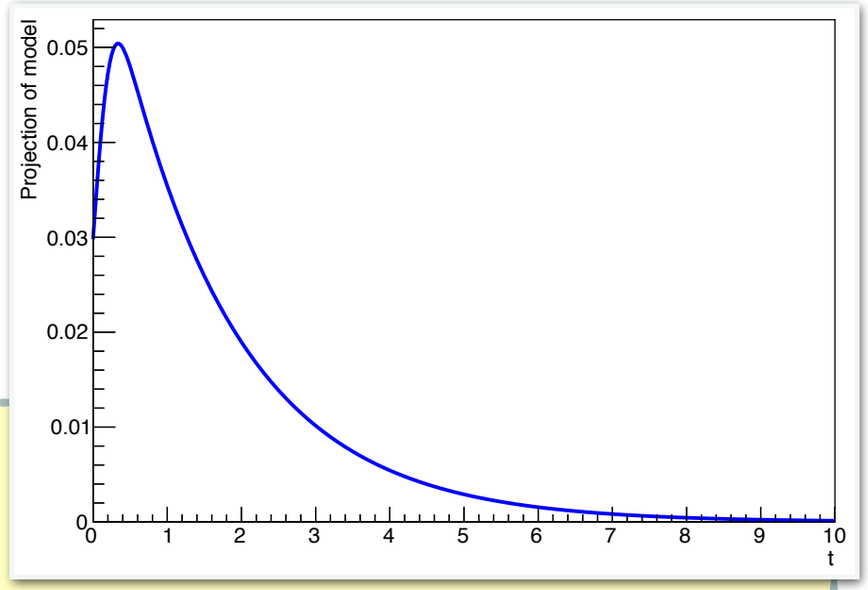


- On the other hand, each slice of the final distribution has the contribution from nearby slices from the original distribution, according to the probability given by a Gaussian.



PRACTICE: EXPONENTIAL CONVOLUTED WITH GAUSSIAN

- The convolution requires some integration works. In most of the cases it is difficult to do it analytically.
- There is no direct implementation within ROOT itself but RooFit does have the functionality to perform the convolution.
- Here are an example how to obtained a convoluted exponential function within RooFit:



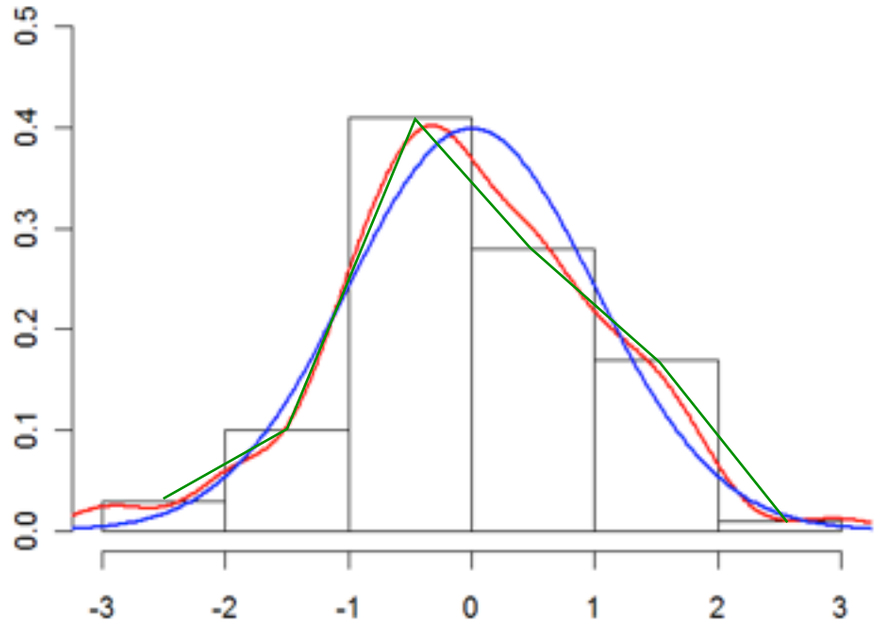
example_06.cc

```
{
    using namespace RooFit;

    RooRealVar t("t","t",0.,10.);
    RooRealVar lambda("lambda","Decay parameter",1.6);
    RooGaussModel res("res","Resolution model",t,RooConst(0.),RooConst(0.2));
    RooDecay model("model","model",t,lambda,res,RooDecay::SingleSided);

    RooPlot *frame = t.frame();
    model.plotOn(frame);

    frame->Draw();
}
```

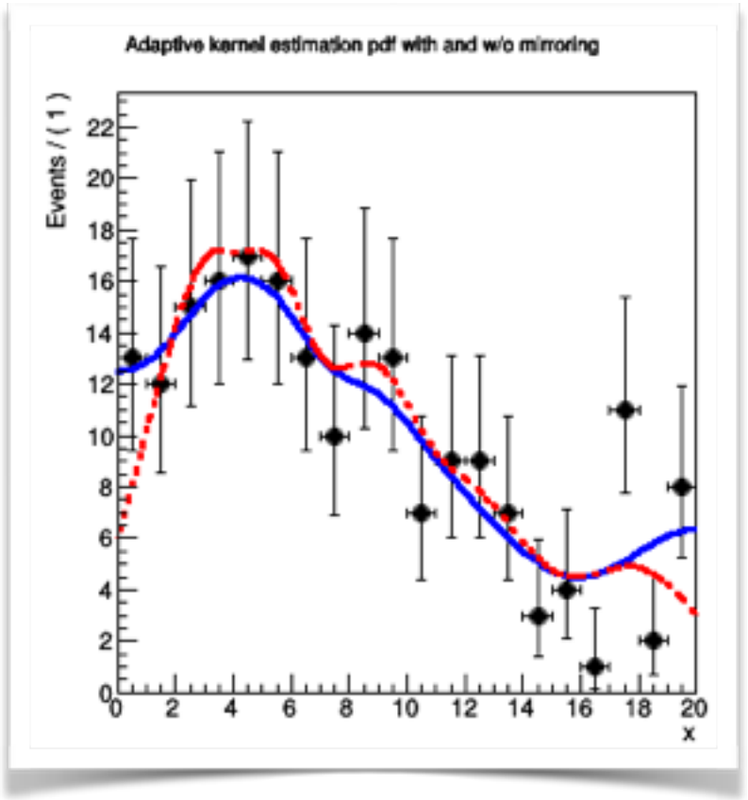
$$f(t;\lambda) \propto \exp(-\lambda t) \text{ and } r(t,t') \propto \exp\left[-\frac{(t'-t)^2}{2\sigma^2}\right]$$

NON-PARAMETRIC MODELS

- Sometimes the data might be difficult to be modelled with a specific function form. In this a situation, some non-parametric models could be a good option.
- Surely one can still build a model with high-order polynomials, but it might generate some not-so-natural models with many small detailed structures.
- In some of the cases a non-parametric model, for example, a histogram-based PDF (*with some smoothing interpolation if needed*) can be a cost-effective solution.



KERNEL ESTIMATION



- The kernel estimation is simple way to convert a set of data to a (*smoothed*) function form.
- The general kernel estimate of the parent distribution is given by

$$P(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h_i}K\left(\frac{x-t_i}{h_i}\right)$$

where t_i are the sampled events, h_i is the smoothing (bandwidth) parameter. An obvious choice of kernel K is a Gaussian with $\mu=0$ and $\sigma=1$:

$$K(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$$

So this is just a sum of many Gaussians!

- An adaptive choice of the bandwidth h_i is usually introduced in the implementation. A factor ρ is applied to scale the width for each event. See [hep-ex/0011057](hep-ex/0011057) for details.

KERNEL ESTIMATION

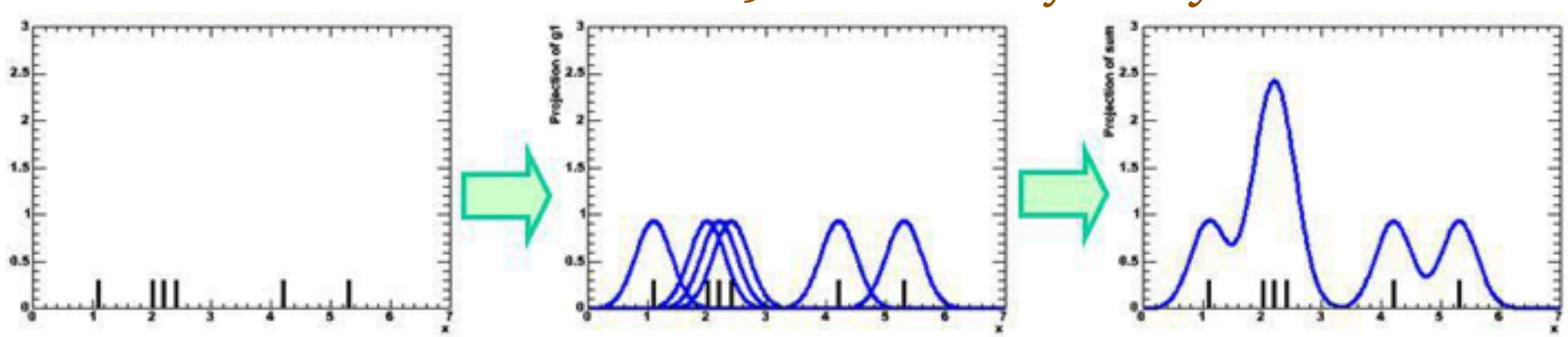
- The kernel estimation is simple way to convert a set of data to a (*smoothed*) function form. The general kernel estimate of the parent distribution is given by

$$P(x) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{h_i}K\left(\frac{x-t_i}{h_i}\right) \Leftarrow K(x) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{x^2}{2}\right)$$

where t_i are the sampled events, h_i is the smoothing (bandwidth) parameter. An obvious choice of kernel K is a Gaussian with $\mu=0$ and $\sigma=1$.

- An adaptive choice of the bandwidth h_i is usually introduced in the implementation. A factor ρ is applied to scale the width for each event. See hep-ex/0011057 for details.

Just a sum of many Gaussians!

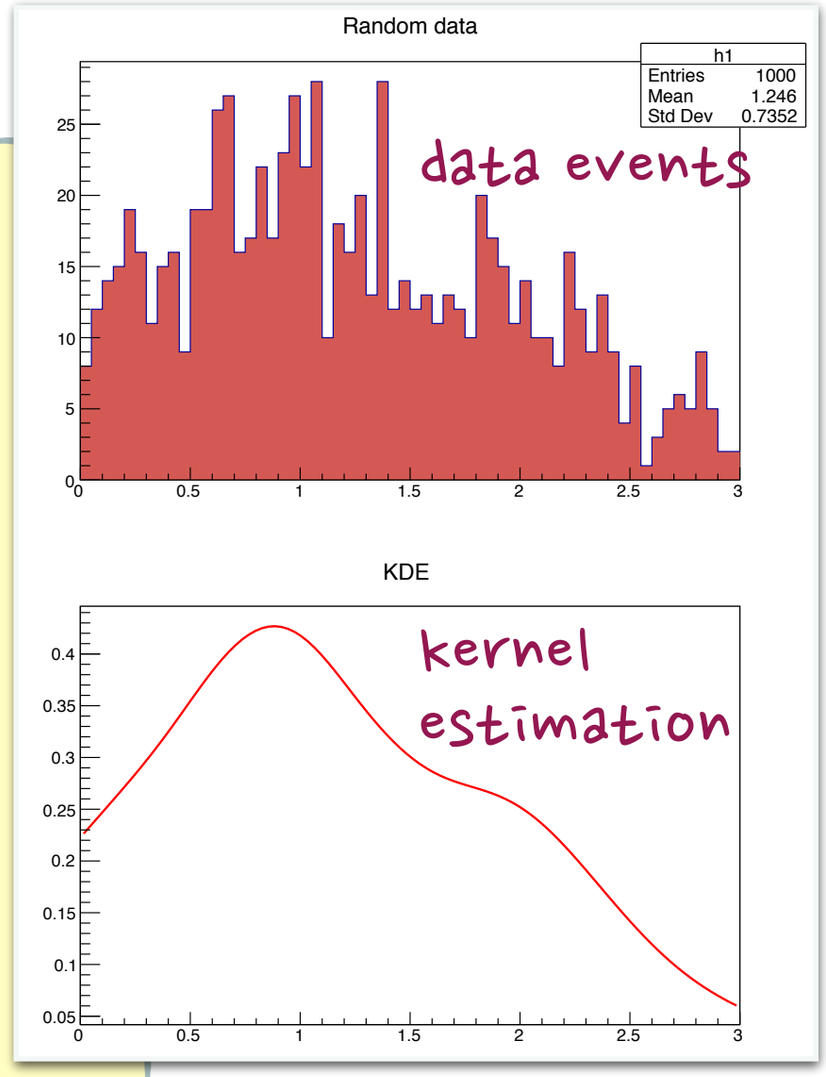


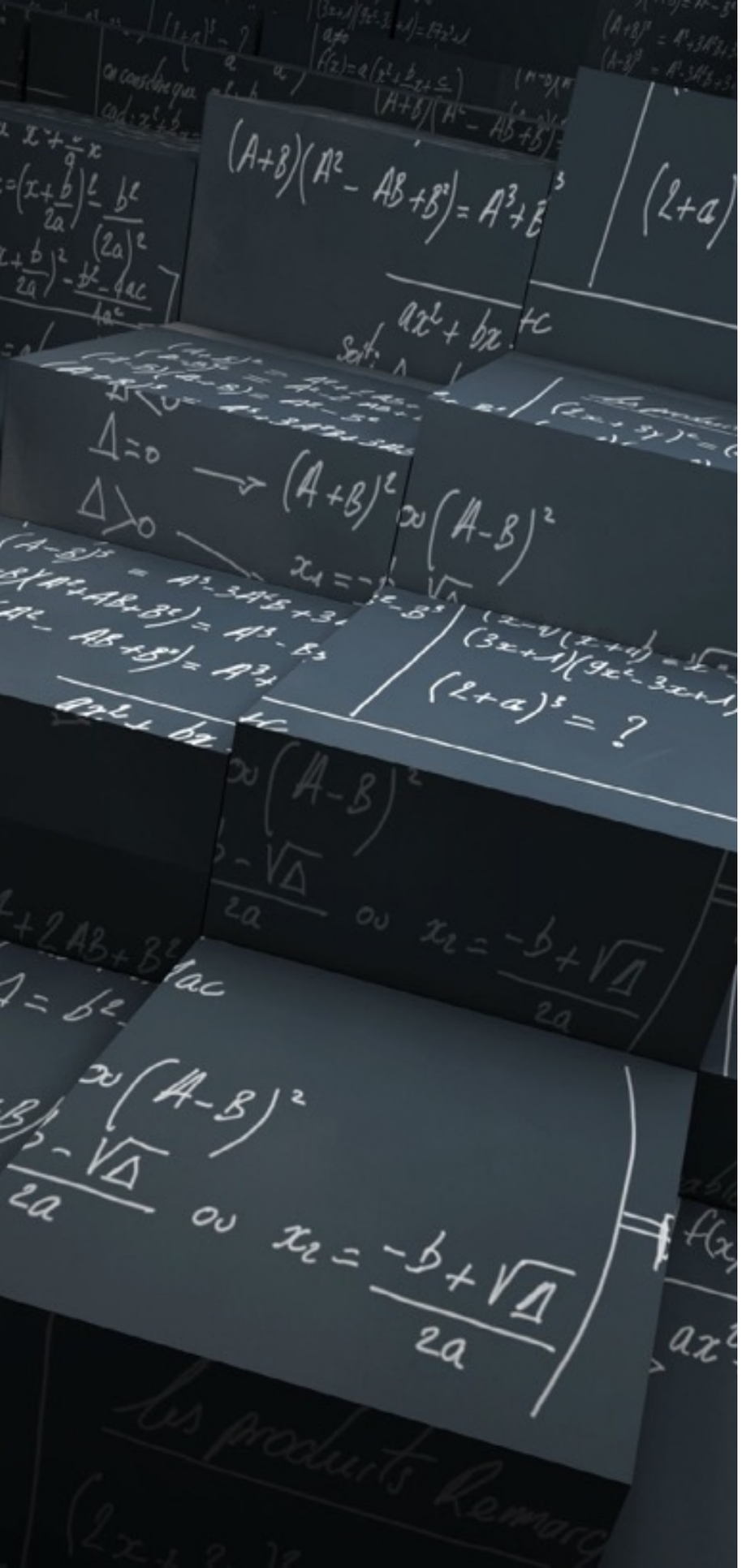
PRACTICE: KERNEL ESTIMATION

- In ROOT there is already an implementation: **TKDE**. Or in RooFit the **RooKeysPdf** is the most straightforward tool.

example_07.cc

```
{
    TRandom3 rnd;
    TH1D *h1 = new TH1D("h1","Random data",60,0.,3.);

    vector<double> vec;
    for(int i=0;i<1000;i++) {
        double x = rnd.Gaus(1.,1.);
        vec.push_back(x);
        h1->Fill(x);
    }

    TKDE *kde = new TKDE(vec.size(),vec.data(),0.,3.);
    TCanvas *c1 = new TCanvas("c1","",600,800);
    c1->Divide(1,2);
    c1->cd(1);
    h1->SetFillColor(50);
    h1->Draw();
    c1->cd(2);
    kde->Draw();
}
```





SUMMARY

- In this lecture we went though many commonly used probability distributions, including Poisson, Gaussian, etc.
- These distributions and models could be very useful to describe the distributions for your (*upcoming*) studies!
- For the next lecture, we are going to discuss how to extract unknown parameters out of your data and model.