计算物理 第二部分
第5讲

李强　北京大学物理学院中楼411
qliphy0@pku.edu.cn， 15210033542
http://www.phy.pku.edu.cn/~qiangli/CP2017.html

5. 随机数及简明概率论

真、伪随机数

产生方法：(1) 取中法, (2) 线性同余产生器, (3) 反馈位移寄存器

C/C++ Random generator

CERN Root三种Random generator

随机性的检验: 均匀性，独立性

简明概率论

平均值，方差　中心极限定理

常见分布函数: Gauss, Landau, Chi2分布

置信度

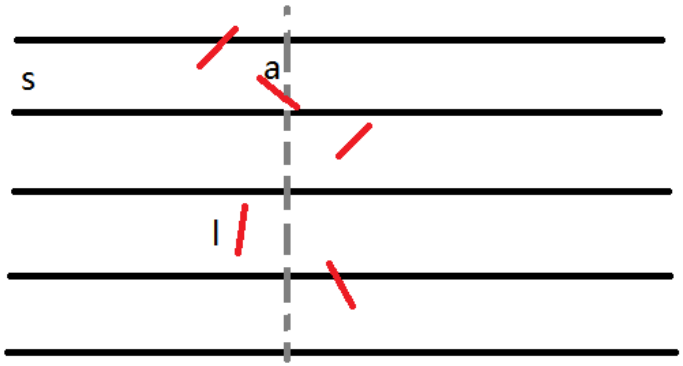
随机性的检验: correlation function

抽样

直接抽样, 变换抽样 ， 取舍取样

加、减、乘抽样

引言：布冯实验



试验者	时间	投掷次数	相交次数	圆周率估计值
Wolf	1850年	5000	2532	3.1596
Smith	1855年	3204	1218.5	3.1554
C.De Morgan	1860年	600	382.5	3.137
Fox	1884年	1030	489	3.1595
Lazzerini	1901年	3408	1808	3.1415929
Reina	1925年	2520	859	3.1795

$$\iint f(h,\alpha)\,dhd\alpha$$

- 该试验方案是：在平滑桌面上划一组相距为s的平行线，向此桌面随意投掷长度$l \le s$的细针，从针与平行线相交的概率就可以得到π的数值
- 设针与平行线的垂直方向的夹角为α，细针与平行线相交的概率为投影长度与平行线间距之比，即$\frac{l|\cos\alpha|}{s} = |\cos\alpha|$，又由于$|\cos\alpha|$的平均值为

$$\frac{1}{\pi}\int_0^{\pi} |\cos\alpha| d\alpha = \frac{2}{\pi}$$

- 假如在N次投针中，有M次和平行线相交。当N充分大时，相交的频数M/N就近似为细针与平行线相交的概率。既有

$$\pi \approx \frac{2N}{M}$$

随机数

- 随机数可分为两种类型：真随机数和伪随机数
- (一)真随机数
- 真随机数数列是在某次产生过程中是按照实验过程中表现的分布概率随机产生的，其结果是不可预测的，是不可见的。因而也不可能重复产生两个相同的真随机数数列
- 真随机数的产生方法：
  - (1)物理方法：放射性物质放出的粒子数，电子设备的热噪音，宇宙射线的触发时间，优点是产生的随机数的具有好的独立性和均匀性，缺点是技术含量要求高，需要配备专门的物质和仪器，费用昂贵
  - (2)人工方法：比如掷骰子，抛硬币，抽签，摇号等。优点是简单易行，缺点是如果需要大量的随机数的情况，执行效率非常低,而且无法重复实现，给验证结果带来很大困难
- 在密码学等关键性应用中，人们一般使用真随机数
- 计算机程序不能产生绝对的真随机数

伪随机数

- “伪”随机数的概念解释
- 计算机产生的都是”伪随机数“。真随机数是一种理想的随机数，即使计算机怎样发展，它也不会产生一串真随机数。计算机只能生成相对随机的随机数，即伪随机数
- 伪随机数并不是假随机数，这里的“伪”是有规律的意思，就是计算机产生的伪随机数既有真随机数的随机性质。又有伪随机数独有的规律性质。
- 这种规律性质是指计算机产生的伪随机数列是一个确定性数列，它是按照一定的算法通过数学递推公式模拟产生的，其结果是确定的，是可预见的。可以这样认为这个可预见的结果其出现的概率是100%。
- 但是如果确定性的计算方法经过精心设计，其依赖的信息足够大，足够复杂，使人力甚至是机器都难以猜出其中的规律(科学地讲是随机序列能够通过独立性，均匀性等一系列的统计检验)，我们就说输出的数是随机数序列与真正的随机数具有相近的性质。

伪随机数的产生

- 比如电脑可以记录鼠标点击次数、鼠标移动范围、键盘操作次数，系统使用时间（确精到毫秒以下）、当前时间（确精到毫秒以下）等不确定因素来做随机因子，这样出来的伪随机数也可作为真随机数来使用。
- 产生伪随机数的算法被称为伪随机数产生器。理论上，我们要求伪随机数产生器具备以下特征：
  - (1)良好的统计分布特性，这是最重要的一点；
  - (2)高效率的伪随机数产生；
  - (3)伪随机数产生的循环周期长，至少大于随机数使用的个数
  - (4)伪随机数可以重复产生；
- 然而实际使用的伪随机数产生程序还没有一个是十全十美的，因此我们要求产生的伪随机数应当能通过尽可能多的统计检验，以便人们放心使用

取中法

- 虽然$[0,1]$区间上的均匀分布十分简单，但是这个简单分布的随机数对使用随机模拟方法解决问题是十分重要的。许多其他形式的分布(如正态分布，指数分布，二项分布等)的随机数通过舍选变换等抽样方法，都可以从$[0,1]$ 区间的均匀分布的随机数，经过变换得到。
- 最早的伪随机数产生器可能是冯诺依曼平方取中法：该方法首先给出一个2r位的数，取它的中间的r位数码作为第一个伪随机数；然后将第一个伪随机数平方构成一个新的2r位数，再取中间的r位数作为第二个伪随机数…。如此循环便得到一个伪随机数序列。
- 比如一个十进制数列的例子。$x_0 = 6031$,递推之后$x_1 = 3729$，$,x_2 = 9054, x_3 = 9749$
- 相应的$[0,1]$上的均匀分布就为$\xi_i = \frac{x_i}{10000}$(注： 以后如不做特殊说明,一般$\xi, \xi_1, \xi_n \ldots$表示$[0,1]$区间的上均匀分布的抽样序列)
- 这种方法虽然简单，但是均匀性不好，且数列很快趋近于0，数列的长度也难以确定。

线性同余法

- 如今比较流行，并用得最多的是线性同余产生器，全称为Linear Congruence Generator,此方法利用数论中的同余运算来产生随机数，故称为同余发生器。有关同余运算的的相关性质请自行参考数学相关书籍。
- 通过如下的线性同余关系递推公式式来产生数列

$$x_{n+1} = (ax_n + c)(mod\, m)$$

- 那么$[0,1]$区间上的均匀分布的随机数为

$$\xi_n = x_n/m$$

- 显然$\xi_n \in [0,1]$,上式中参数$\mathrm{a, c, m, x_0}$的选取十分关键。其中$\mathrm{x_0}$称为种子，改变它的值就得到基本序列的不同区段。$\mathrm{a, c, m, x_0}$为大于零的整数，分别叫做乘子，增量，模和初值。选择这些参数时需要使得产生出的伪随机数的循环周期要尽可能长。

线性同余法

$$x_{n+1} = (6x_n + 7)(\text{mod } 5), \quad x_0 = 2$$

4, 1, 3, 0, 2, 4, 1, 3, ….

$$x_{n+1} = (27x_n + 11)(\text{mod } 54), \quad x_0 = 2$$

11, 38, 11, ….

线性同余法

$$x_{n+1} = (ax_n + c)(mod\, m)$$

- 当$c \neq 0$时可以实现最大周期m (注： 这里只给出结论，数学证明不需要作深入了解，相关证明可参考相关同余的性质),
- 同时要达到最大周期m,还需要同时满足以下条件：
  - (1)c与m互质(即它们的公因数只有1)
  - (2)对m的任一因子p,满足$a \equiv 1(\mathrm{mod}\, p)$,即$a-1$应被p 整除
  - (3)如果4是m的因子，则$a \equiv 1(\mathrm{mod}\, 4)$,即$a-1$应被4整除
- 实际应用中，通常选取:

$$m = 2^L,\ a = 4q+1\ c = 2p+1$$

- 其中p, q, L均为正整数，此时：

$$x_{n+1} = [(4q+1)x_n + (2p+1)](mod\, 2^L), \quad \xi_n = \frac{x_n}{2^L}$$

- 上述参数的一般是通过定性分析和计算机试验来选择，使得到的伪随机数列具有足够长的周期，而且独立性和均匀性都能通过一系列的检验

C++的线性同余法

- 如果想每次产生的随机数不同，通常我们需要 srand() 函数来初始化随机数发生器，即为随机数发生器设定种子：

$$\text{void srand(unsigned seed);}$$

- 此函数同样来自于stdlib.h。至于种子怎么选，一般常用的方法是利用time函数(声明在头文件time.h)来获得系统时间，然后将time_t型数据转化为unsigned型再传给srand函数，即

$$\text{srand((unsigned) time(NULL);}$$

- 如果想产生其他范围的随机数，例如[a, b]则需要利用求余运算

$$\text{rand()\%(b - a) + a;}$$

- 如果是产生[0, 1]范围内的浮点型随机数,则

$$\text{(double)rand()/RAND\_MAX;}$$

C/C++的线性同余法

- C语言中标准库就是用线性同余产生器实现随机数的生成，其中各个参数的设置如下：

$$m = 2^{31} \quad a = 1103515245 \quad c = 12345$$

- 这个算法的优势在于计算速度快，内存消耗少。并且可以看到它的随机数列周期为2^{31}。但是对于其它一些具体应用，随机数质量要求较高，比如像基于蒙特卡洛的算法粒子滤波器等就不适合用这种方法，因为对于这种算法而言，相邻的随机数并不独立，序列关联性较大。
- C语言下随机数生成函数,声明在头文件stdlib.h中

#include <cstdlib>

```
int rand();
```

- 此函数提供均匀随机数的生成，均匀分布的范围是[0, RAND_MAX],
- 因为rand()是按照指定顺序生成随机数的，所以如果每次随机种子相同，每次生成的随机数都是一致的。

C/C++的线性同余法

```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
 double mean=0.0;
 double variance=0.0;
 double chi2=0.0;
 double f[10];
 for(int i=0; i<10; i++) {
  f[i]=0.;
 }
 srand(unsigned(time(0)));
 for(int i=0; i<NN; i++) {
double fx=rand()/(double)RAND_MAX;
 if(fx<0.1) {
    f[0]+=1.0;
  }
  else if(fx<0.2) {
    f[1]+=1.0;
  }
  ……
  else{
    f[9]+=1.0;
  }
  mean+=fx;
  variance+=fx*fx;
}
  mean=mean/double(NN);
  variance=sqrt(variance/double(NN)-mean*mean);
for(int i=0; i<10; i++) {
     chi2+=(f[i]-double(NN)/10.0)*(f[i]-
        double(NN)/10.0)/(double(NN)/10.0);
}
 chi2=chi2/10.0;
cout<<"mean= "<<mean<<endl;
cout<<"variance= "<<variance<<endl;
cout<<"chi2= "<<chi2<<endl;
}
```

反馈移位寄存器

- 1965年，Tausworthe发明了反馈移位寄存器方法，可以用以消除同余法中的关联问题。但后来发现情况并非那么简单，但至少但至少该方法提供了随机数产生的另一种途径。
- 该方法是对整数进行位操作：首先用其它方法产生一个随机的整数序列，然后对两个整数进行XOR(异或)操作以产生一个新的随机整数()

$$I_n = I_{n-p} \oplus I_{n-q}$$

- 其中的[p, q]是一对整数，最佳的选择是满足条件:

$$p^2 + q^2 + 1 = \text{prime number}$$

- 如：$[31, 3]$, $[98, 27]$, $[250, 103]$, $[1279, (216, 418)]$等，其中的R250 $(\mathrm{p} = 250, \mathrm{q} = 103)$是最为常用的产生器。一般来说 $[p, q]$ 值越大，产生的随机数质量越好，而且起始的随机整数表的质量非常重要。

反馈移位寄存器

C 中为 m^n

- Fortran中对两个整数m和n进行“异或”位操作的函数是IEOR(m, n),例如：$I_1 = 6, I_{148} = 11$，
- 则，$I_{251} = I_1 \oplus I_{148} = 0110 \oplus 1011 = 1101 = 2^3 + 2^2 + 2^0 = 13$
- 程序中可写成$\mathrm{N(K)} = \mathrm{IEOR(N(K-250), N(K-103))}$,要求数组N中存储所有之前的250个随机整数。
- 当然要得到$[0, 1]$区间的随机数需再除以m
- 梅森旋转算法就是利用反馈移位寄存器产生随机数的，目前Python, Ruby, Matlab, C++11, 中都有这个算法的实现。梅森旋转算法周期很长$(2^{19937} - 1)$，远高于线性同余算法的2^{31}
- 不同的算法在不同的场合下各有利弊，统计检验也不能保证随机数的质量。一种产生器的好坏还是要在具体的应用中根据结果来判断。经验指出，一种新的随机数产生器要经过持久和广泛的应用累积才能知道它的各种隐含的缺陷。

CERN Root随机数产生器

1. From the original implementation in FORTRAN by Fred James as part of CLHEP. The initialisation is carried out using a **Multiplicative Congruential generator** using formula constants of L'Ecuyer as described in "F.James, Comp. Phys. Comm. 60
(1990) 329-344".
https://root.cern.ch/doc/master/TRandom1_8cxx_source.html

2. Random number generator class based on the **maximally quidistributed combined Tausworthe generator** by L'Ecuyer. The period of the generator is 2**88 (about 10**26) and it uses only 3 words for the state.
https://root.cern.ch/doc/master/classTRandom2.html

3. TRandom3, is based on the "**Mersenne Twister generator**", and is the recommended one, since it has good random proprieties (period of about 10**6000) and it is fast
http://root.cern.ch/root/html/TRandom3.html

gRandom = new TRandom1/2/3(0);

举例ran0

```
double ran0(long &idum)
{
  const int a = 16807, m = 2147483647,
q = 127773;
  const int r = 2836, MASK = 123459876;
  const double am = 1./m;
  long k;
  double ans;
  idum ^= MASK;
  k = idum/q;
  idum = a*(idum - k*q) - r*k;
  // add m if negative difference
  if(idum < 0) idum += m;
  ans=am*(idum);
  idum ^= MASK;
  return ans;
} // End: function ran0()
```

** The function ran0()
** is an "Minimal" random number generator of Park and Miller
** Set or reset the input value
** idum to any integer value (except the unlikely value MASK)
** to initialize the sequence; idum must not be altered between
** calls for successive deviates in a sequence.
** The function returns a uniform deviate between 0.0 and 1.0.

$$N_i = (aN_{i-1})\mathrm{MOD}(M)$$
$$M = aq + r,$$
$$q = [M/a], \quad r = M \ \mathrm{MOD}\ a.$$

$$(aN_{i-1})\mathrm{MOD}(M) = (aN_{i-1} - [N_{i-1}/q]M)\mathrm{MOD}(M),$$
$$(aN_{i-1})\mathrm{MOD}(M) = (aN_{i-1} - [N_{i-1}/q](aq + r))\mathrm{MOD}(M),$$

$$(aN_{i-1})\mathrm{MOD}(M) = (a(N_{i-1} - [N_{i-1}/q]q) - [N_{i-1}/q]r))\,\mathrm{MOD}(M),$$

⇨ $$(aN_{i-1})\mathrm{MOD}(M) = (a(N_{i-1}\mathrm{MOD}(q)) - [N_{i-1}/q]r))\,\mathrm{MOD}(M)$$

举例 ran2

```
double ran2(long &idum)
{
  const int IM1=2147483563;
  const int IM2=2147483399;
  const double AM = 1./IM1;
  const double IMM1 = IM1-1;
  const int IA1=40014;
  const int IA2=40692;
  const int IQ1=53668;
  const int IQ2=52774;
  const int IR1=12211;
  const int IR2=3791;
  const int NTAB=32;
  const int NDIV=1+IMM1/NTAB;
  const double EPS=1.2e-7;
  const double RNMX=1.0-EPS;
   int          j;
  long          k;
  static long    idum2 = 123456789;
  static long    iy=0;
  static long    iv[NTAB];
  double        temp;
```

```
if(idum <= 0) {
    if(-(idum) < 1) idum = 1;
    else            idum = -(idum);
    idum2 = (idum);
    for(j = NTAB + 7; j >= 0; j--) {
      k     = (idum)/IQ1;
                idum = IA1*(idum - k*IQ1) - k*IR1;
                if(idum < 0) idum +=  IM1;
                if(j < NTAB)  iv[j]  = idum;     }
    iy=iv[0];
  }
  k     = (idum)/IQ1;
  idum = IA1*(idum - k*IQ1) - k*IR1;
  if(idum < 0) idum += IM1;
  k     = idum2/IQ2;
  idum2 = IA2*(idum2 - k*IQ2) - k*IR2;
  if(idum2 < 0) idum2 += IM2;
  j     = iy/NDIV;
  iy    = iv[j] - idum2;
  iv[j] = idum;
  if(iy < 1) iy += IMM1;
  if((temp = AM*iy) > RNMX) return RNMX;
  else return temp;
} // End: function ran2()
```

The function ran2() is a long period (> 2 x 10^18) random number generator of L'Ecuyer and Bays-Durham shuffle and added safeguards. Call with idum a negative integer to initialize; thereafter, do not alter idum between successive deviates in a sequence. RNMX should approximate the largest floating point value that is less than 1. The function returns a uniform deviate between 0.0 and 1.0

随机性的统计检验

- 伪随机数的好坏通常是由各种统计检验来判定，这统计些检验包括：**均匀性检验、独立性检验、组合规律检验、无连贯性检验、参数检验**等等。
- 这是必要条件不是充分条件(不能通过检验的必定不是好的产生器，但通过有限个检验指标的也不能保证它是好的)
- 最基本的有两种：**均匀性检验**和**独立性检验**：
  - **均匀性检验**是指在$[0,1]$区间内等长度子区间中随机数的数量是一样的；
  - **独立性检验**是按先后顺序出现的若干个随机数中，每一个数的出现都和它前后的各个数无关。
- 一个好的伪随机数序列能通过的检验越多，那么该产生器就会越优良可靠。

- **均匀性检验：**
- 均匀性检验的方法很多，如有χ^2检验，K－S检验，序列检验。

随机性的统计检验

- 设有在区间$[0,1]$上的伪随机数序列为$\{\xi_1, \xi_2 \ldots \xi_n\}$
- 如果该伪随机数是均匀分布的，则将$[0,1]$区间分成k 个相等的子区间后，落在每个子区间的伪随机数个数应当近似为$\mathrm{m_k} = \frac{\mathrm{N}}{\mathrm{k}}$，此数为**理论频数**。
- 统计随机数落在第k个子区间的**实际频数**$\mathrm{n_k}$，它应当趋近于理论频数$\mathrm{m_k}$。
- 注意此处的$\mathrm{n_k}$是整数而$\mathrm{m_k}$可以是小数。
- 考察统计量χ^2，它定义为：

$$\chi^2 = \sum_{k=1}^{k} \frac{(n_k - m_k)^2}{m_k}$$

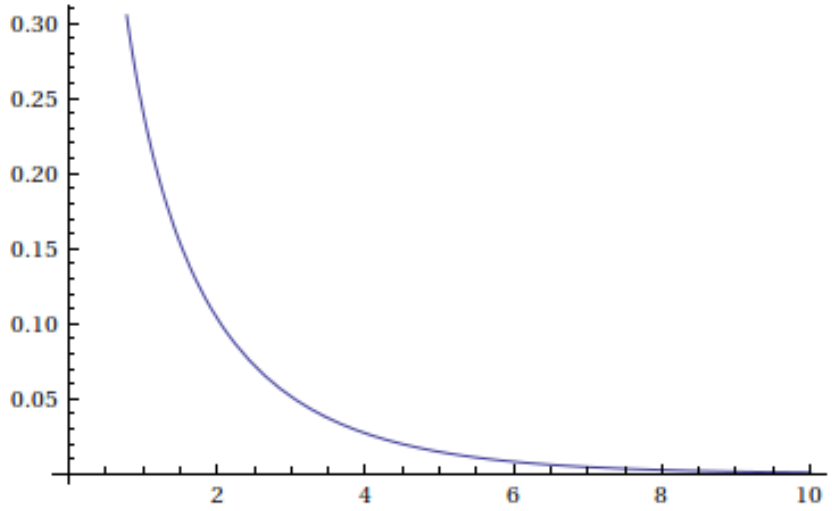
- 如果χ^2值很大，表示远远偏离理想值，因此要求χ^2值尽可能小。通常求和中的每一项的大小约为1，因此χ^2的值约为k
- 概率论中的Pearson定理说明，(15)式的极限概率分布是$\chi^2(k-1)$分布

随机性的统计检验

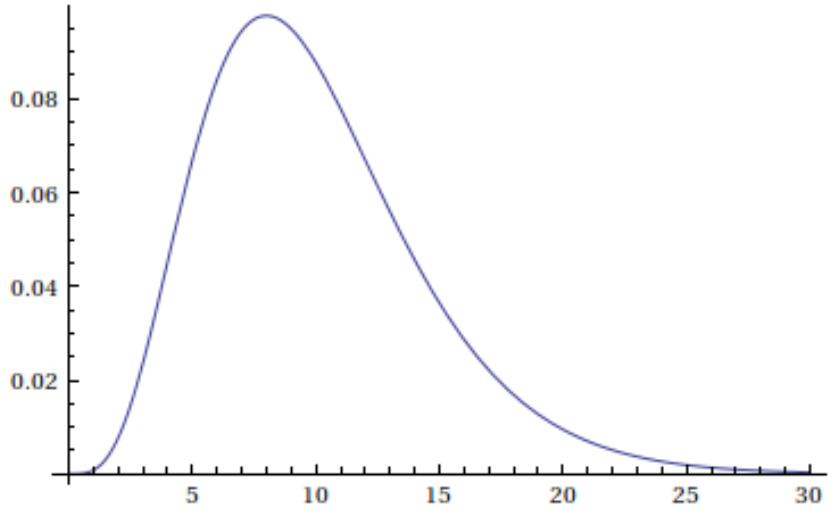
$$P(\chi^2 \le x \mid \upsilon) = \frac{1}{2^{\upsilon/2}\Gamma(\upsilon/2)} \int_0^x t^{(\upsilon-2)/2} e^{-t/2} dt$$

- 它给出了(15)式中的其中$\chi^2 \le x$的概率。整数υ 是系统的自由度，表示独立测量的次数。
- 但是有一个约束条件存在，$\sum_{k=1}^{k} m_k = N$,故自由度$\upsilon = k - 1$。
- 据此可以假定一个显著性水平值来进行检验。当给定显著水平α后(或置信度$1-\alpha$)，由方程$P(\chi_\alpha^2 \mid \upsilon) = 1 - \alpha$解出$\chi_\alpha$ 值，或者直接从χ^2表查得k $-$ 1个自由度的显著水平为α时的χ_α值。
- 如果由(16)式计算出来的χ小于χ_α，则认为在此置信度下，原伪随机数在$[0, 1]$区间是均匀分布的假定是正确的。
- 如果由(16)式计算出来的χ大于χ_α，则认为在α 的显著水平下，伪随机数不满足均匀性的要求。

```
Plot[(x^(n/2-1)*Exp[-x/2]/2^(n/2)/Gamma[n/2]) /. {n → 1}, {x, 0, 10}]
```




```
Plot[(x^(n/2-1)*Exp[-x/2]/2^(n/2)/Gamma[n/2]) /. {n → 10}, {x, 0, 30}]
```



随机性的统计检验

```
root [0] TMath::Prob(10.82,1)
(Double_t)1.00409489093039703e-03
root [1] TMath::Prob(10.83,1)
(Double_t)9.98686379180259171e-04
root [2] TMath::Prob(3.84,1)
(Double_t)5.00435212487051889e-02
```

- 通常置信度水平选取为0.99或者0.95。为了反映均匀性分布的特性，k的取值不宜太小，但也不能太大。一般选取的k值，要能使每个子区间有若干个伪随机数时就比较合适。

Degrees of freedom (df)	χ^2 value[19]										
1	0.004	0.02	0.06	0.15	0.46	1.07	1.64	2.71	3.84	6.64	10.83
2	0.10	0.21	0.45	0.71	1.39	2.41	3.22	4.60	5.99	9.21	13.82
3	0.35	0.58	1.01	1.42	2.37	3.66	4.64	6.25	7.82	11.34	16.27
4	0.71	1.06	1.65	2.20	3.36	4.88	5.99	7.78	9.49	13.28	18.47
5	1.14	1.61	2.34	3.00	4.35	6.06	7.29	9.24	11.07	15.09	20.52
6	1.63	2.20	3.07	3.83	5.35	7.23	8.56	10.64	12.59	16.81	22.46
7	2.17	2.83	3.82	4.67	6.35	8.38	9.80	12.02	14.07	18.48	24.32
8	2.73	3.49	4.59	5.53	7.34	9.52	11.03	13.36	15.51	20.09	26.12
9	3.32	4.17	5.38	6.39	8.34	10.66	12.24	14.68	16.92	21.67	27.88
10	3.94	4.87	6.18	7.27	9.34	11.78	13.44	15.99	18.31	23.21	29.59
P value (Probability)	0.95	0.90	0.80	0.70	0.50	0.30	0.20	0.10	0.05	0.01	0.001

the probability of observing a test statistic *at least* as extreme in a chi-squared distribution

随机数产生算法的表现 示例：

平均值；
方差；
chi2

$$\int_0^1 (x^2 - \frac{1}{2^2})dx = \frac{1}{3} - \frac{1}{4} = \frac{1}{12} \approx (0.2886)^2$$

随机性的统计检验

- **独立性检验：**
- 独立性检验的方法也有若干，如有列联表检验，相关系数检验，游程检验。这里介绍列联表检验。
- 如果把$[0,1]$上的伪随机序列$\{\xi_1,\xi_2,\cdots\cdots\xi_{2N}\}$分成两组：

$$\xi_1,\xi_3\cdots\cdots\xi_{2N-1} \qquad \xi_2,\xi_4\cdots\cdots\xi_{2N}$$

- 第一组作为随机变量x的取值，第二组作为随机变量y 的取值
- 在x − y平面内的单位正方形域$[0\le x\le 1, 0\le y\le 1]$上，分别以平行于坐标轴的平行线，将正方形域分成k × k个相同面积的小正方形网格。落在每个网格内的随机数的实际频数n_{ij}应当近似等于理论频数$m_{ij}=\frac{N}{k^2}$

由此可以算出χ^2为：

$$\chi^2=\sum_{i,j=1}^{k}\frac{(n_{ij}-m_{ij})^2}{m_{ij}}$$

同样，χ^2应满足$\chi^2((k-1)^2)$的分布。据此可以采用与均匀性检验的χ^2方法，假定出显著性水平来进行检验。

https://indico.ihep.ac.cn/event/4902/contribution/15/material/slides/0.pdf

Statistical Methods for Particle Physics

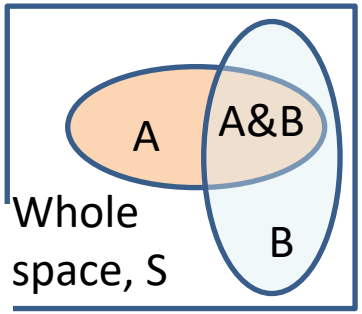
Lecture 1: intro, parameter estimation, tests

http://indico.ihep.ac.cn/event/4902/

iSTEP 2015
Shandong University, Jinan
August 11-19, 2015

Glen Cowan (谷林·科恩)
Physics Department
Royal Holloway, University of London
g.cowan@rhul.ac.uk
www.pp.rhul.ac.uk/~cowan

概率Probability



A definition of probability

Consider a set S with subsets A, 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)

Also define conditional probability of A given B:

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

Subsets A, B independent if: $P(A \cap B) = P(A)P(B)$

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

Interpretation of probability

I. Relative frequency

A, B, ... are outcomes of a repeatable experiment

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

cf. quantum mechanics, particle scattering, radioactive decay...

II. Subjective probability

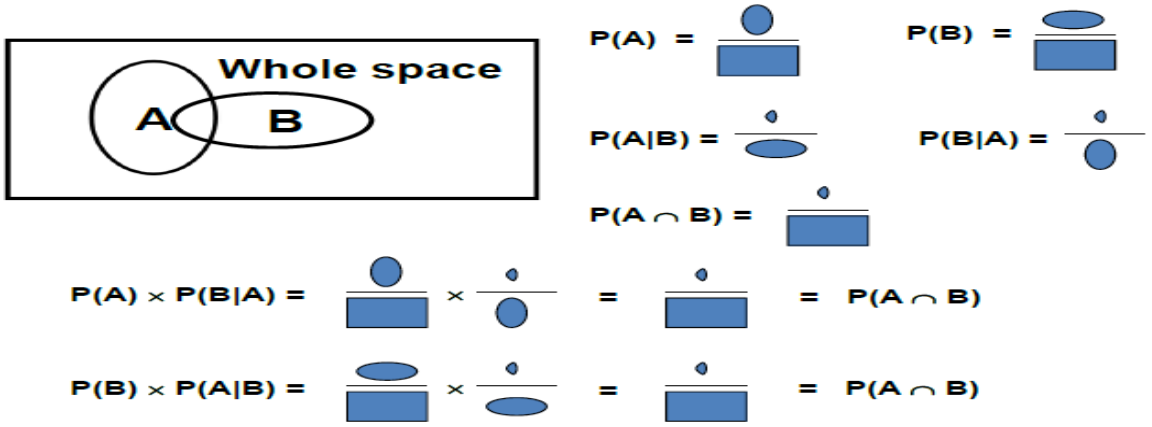
A, B, ... are hypotheses (statements that are true or false)

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

- Both interpretations consistent with Kolmogorov axioms.
- In particle physics frequency interpretation often most useful, but subjective probability can provide more natural treatment of non-repeatable phenomena:

systematic uncertainties, probability that Higgs boson exists,...

Bayes Theorem



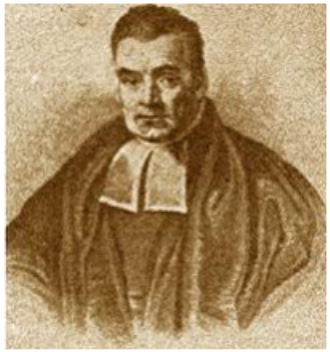
From the definition of conditional probability we have,

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

but $P(A \cap B) = P(B \cap A)$, so

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

Bayes' theorem



First published (posthumously) by the Reverend Thomas Bayes (1702−1761)

An essay towards solving a problem in the doctrine of chances, Philos. Trans. R. Soc. **53** (1763) 370; reprinted in Biometrika, **45** (1958) 293.

An example using Bayes' theorem

Suppose the probability (for anyone) to have a disease D is:

$$P(\mathrm{D}) = 0.001 \qquad P(\text{no D}) = 0.999$$

← prior probabilities, i.e., before any test carried out

Consider a test for the disease: result is + or −

$$P(+|\mathrm{D}) = 0.98 \qquad P(-|\mathrm{D}) = 0.02$$

← probabilities to (in)correctly identify a person with the disease

$$P(+|\text{no D}) = 0.03 \qquad P(-|\text{no D}) = 0.97$$

← probabilities to (in)correctly identify a healthy person

Suppose your result is +. How worried should you be?

The probability to have the disease given a + result is

$$p(\mathrm{D}|+) = \frac{P(+|\mathrm{D})P(\mathrm{D})}{P(+|\mathrm{D})P(\mathrm{D}) + P(+|\text{no D})P(\text{no D})}$$

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

$$= 0.032$$

← posterior probability

i.e. you're probably OK!

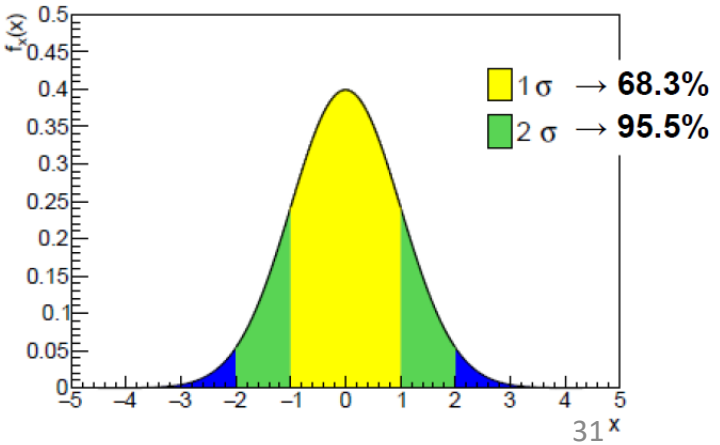
Your viewpoint: my degree of belief that I have the disease is 3.2%.

Your doctor's viewpoint: 3.2% of people like this have the disease.

Probability density functions

- Often a measurement of some observable quantity results in a continuous random variable. We can consider intervals in the possible values as subsets of the total sample space, and derive the probability that the measurement is included in them
  - $P(x_{meas}$ *in* $[x, x+dx[) = f(x)\,dx$
  - Of course this has no meaning unless $P(x \text{ in } S)=1$

- If *x* is single-dimensional, we realize that a set of probability values as the one defined above can be effectively drawn as a histogram, if we define the width of the bins to be *dx*.
- By letting *dx* go to zero, we obtain the continuous function *f(x)*. This is called "probability density function" of *x* (PDF).

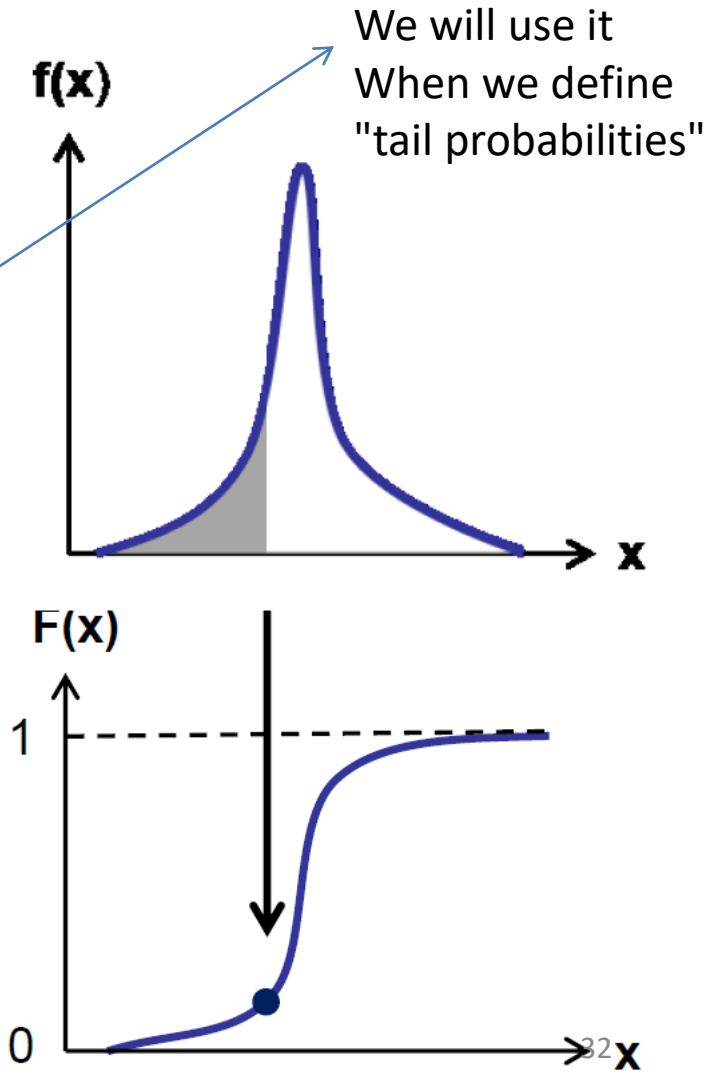


The cumulative distribution

- For any single-dimensional PDF f(x) it is straightforward to define the cumulative distribution F(x) as

$$F(x) = \int_{-\infty}^{x} f(x')dx'$$

- The meaning of F(x) is that of the probability to observe x' with a value smaller or equal to x
- This leads to the concept of quantile x_α, defined as value of x such that $F(x_\alpha)=\alpha$.
  - The special quantile with a name is the **median**, $x_{0.5}$.
  - Other important quantiles: $x_{0.05}$, $x_{0.95}$, $x_{0.16}$, $x_{0.84}$.
- The median, like the **mean** (see later) but unlike the **mode** [the most probable value of the distribution f(x)], can be a value not belonging to S



平均値

E[.]: the Mean

- The *probability density function* (pdf) **f(x)** of a random variable x is a normalized function which describes the probability to find x in a given range: as already written before,

$$P(x,x+dx) = f(x)dx$$

  - This is defined for continuous variables. For discrete ones, e.g. $P(n|\mu)=e^{-\mu}\mu^n/n!$, P is a probability tout-court.
- The *expectation value* of the random variable x is then defined as

$$E[x] = \int_{-\infty}^{+\infty} xf(x)dx = \mu$$

a function of the parameters of the model

- E[x], also called population mean , or simply **mean**, of x, thus depends on the distribution f(x). Note that E[x] is not a function of x, but it is rather a fixed quantity dependent on the form of the PDF f(x).
- The formulation of the expectation value is useful to define other properties of the PDF, as shown in the following.

方差

The variance

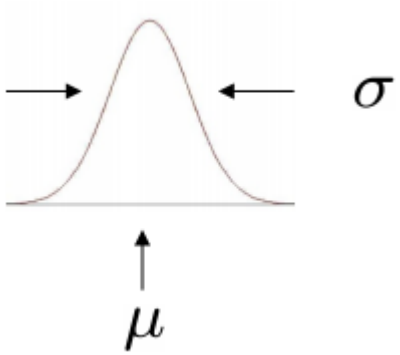
- Of crucial importance to determine the property of a distribution is the “second central moment” of x,

$$E[(x-E[x])^2] = \int_{-\infty}^{+\infty} (x-\mu)^2 f(x)dx = V[x]$$

also called *variance*. The variance describes the "spread" of the PDF around its expectation value. It enjoys the property that

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

as it is trivial to show.



- Also well-known is the *standard deviation* σ = sqrt(V[x]).

常见PDF: Binomial

Consider N independent experiments (Bernoulli trials):

outcome of each is 'success' or 'failure',

probability of success on any given trial is p.

Define discrete r.v. n = number of successes ($0 \leq n \leq N$).

Probability of a specific outcome (in order), e.g. 'ssfsf' is

$$pp(1-p)p(1-p) = p^n(1-p)^{N-n}$$

But order not important; there are $\dfrac{N!}{n!(N-n)!}$

ways (permutations) to get n successes in N trials, total probability for n is sum of probabilities for each permutation.

常见PDF: Binomial

The binomial distribution is therefore

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

random variable

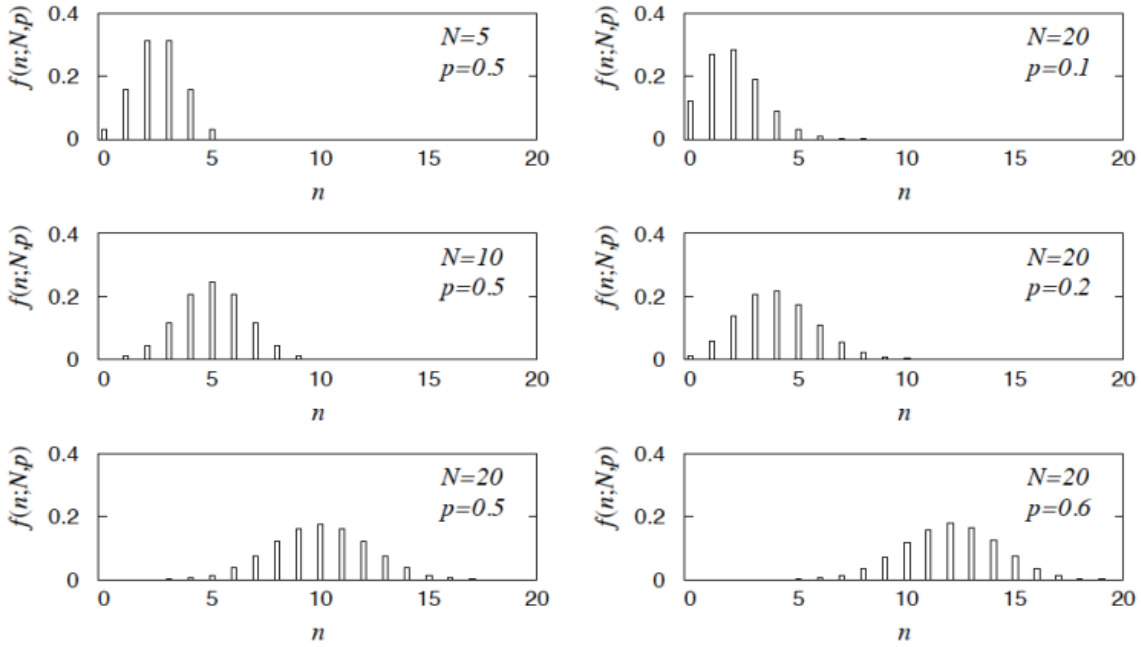
parameters

For the expectation value and variance we find:

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

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

Binomial distribution for several values of the parameters:



Example: observe N decays of $W^{\pm}$, the number n of which are $W \to \mu\nu$ is a binomial r.v., p = branching ratio.

常见PDF: Poisson

Consider binomial n in the limit

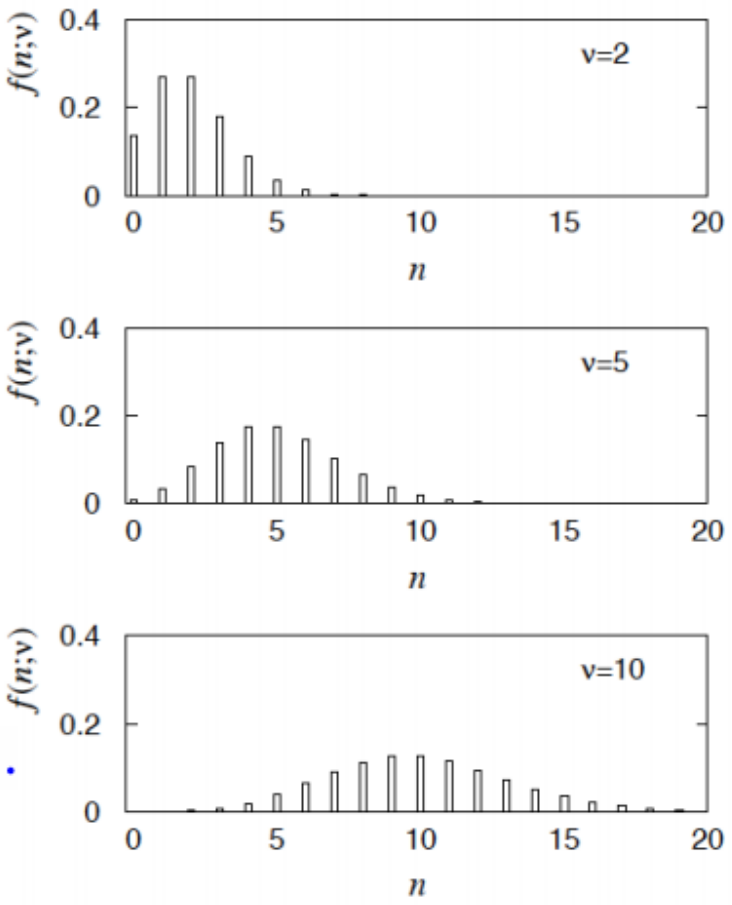
$$N \to \infty, \qquad p \to 0, \qquad E[n] = Np \to \nu \, .$$

→ n follows the Poisson distribution:

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

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

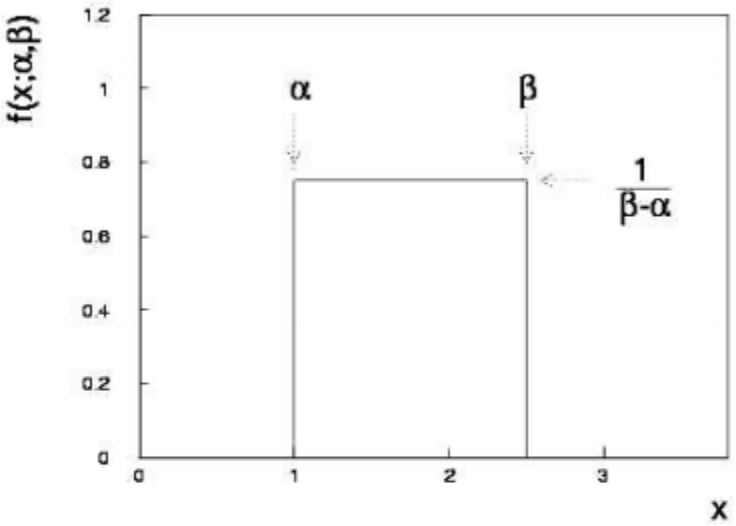
Example: number of scattering events n with cross section σ found for a fixed integrated luminosity, with $\nu = \sigma \int L \, dt$.



常见PDF: Uniform

Consider a continuous r.v. x with $-\infty < x < \infty$. Uniform pdf is:

$$f(x;\alpha,\beta) = \begin{cases} \frac{1}{\beta-\alpha} & \alpha \le x \le \beta \\ 0 & \text{otherwise} \end{cases}$$



$$E[x] = \frac{1}{2}(\alpha+\beta)$$

$$V[x] = \frac{1}{12}(\beta-\alpha)^2$$

N.B. For any r.v. x with cumulative distribution $F(x)$, $y = F(x)$ is uniform in [0,1].

Example: for $\pi^0 \to \gamma\gamma$, E_γ is uniform in $[E_{\min}, E_{\max}]$, with

$$E_{\min} = \frac{1}{2}E_\pi(1-\beta), \quad E_{\max} = \frac{1}{2}E_\pi(1+\beta)$$

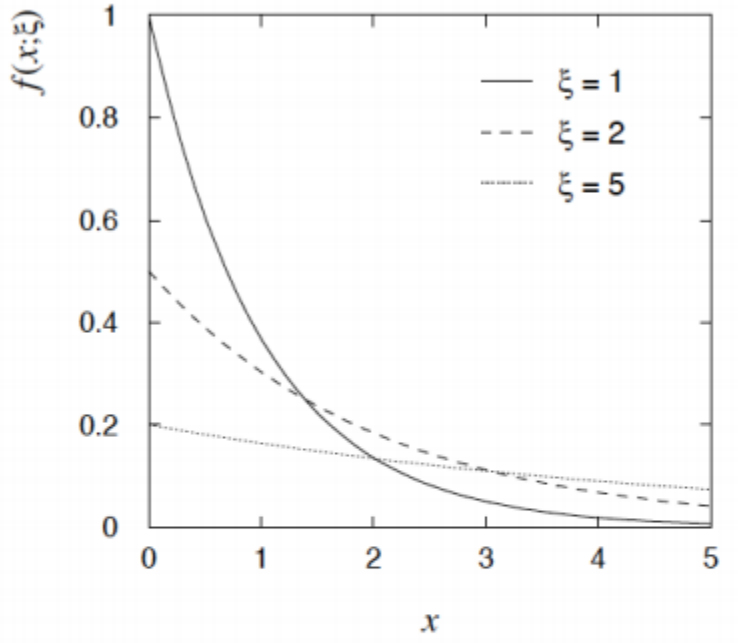
常见PDF: Exponential

The exponential pdf for the continuous r.v. x is defined by:

$$f(x;\xi) = \begin{cases} \frac{1}{\xi} e^{-x/\xi} & x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$E[x] = \xi$$

$$V[x] = \xi^2$$



Example: proper decay time t of an unstable particle

$$f(t;\tau) = \frac{1}{\tau} e^{-t/\tau} \qquad (\tau = \text{mean lifetime})$$

Lack of memory (unique to exponential): $f(t - t_0 | t \geq t_0) = f(t)$

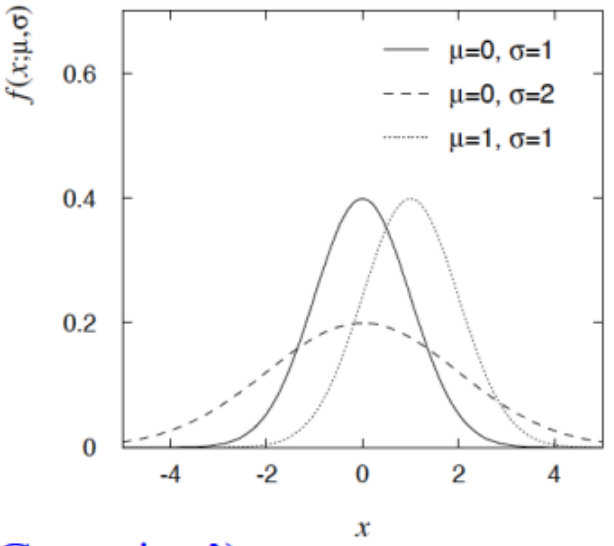
常见PDF: Gaussian

The Gaussian (normal) pdf for a continuous r.v. x is defined by:

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

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

$V[x] = \sigma^2$



Special case: $\mu = 0$, $\sigma^2 = 1$ ('standard Gaussian'):

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \qquad \Phi(x) = \int_{-\infty}^{x} \varphi(x')\,dx'$$

If $y \sim$ Gaussian with μ, σ^2, then $x = (y - \mu)/\sigma$ follows $\varphi(x)$.

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

常见PDF: Chi2

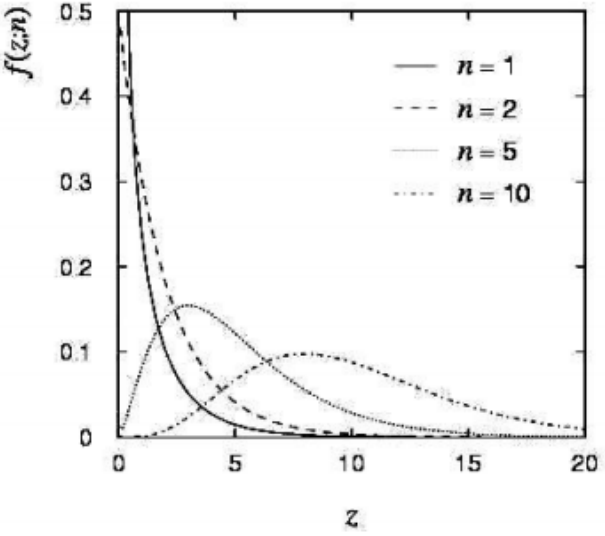
Chi-square (χ^2) distribution

The chi-square pdf for the continuous r.v. z ($z \geq 0$) is defined by

$$f(z;n) = \frac{1}{2^{n/2}\Gamma(n/2)} z^{n/2-1} e^{-z/2}$$

$n = 1, 2, \ldots =$ number of 'degrees of freedom' (dof)

$$E[z] = n, \quad V[z] = 2n .$$



For independent Gaussian x_i, $i = 1, \ldots, n$, means μ_i, variances σ_i^2,

$$z = \sum_{i=1}^{n} \frac{(x_i - \mu_i)^2}{\sigma_i^2}$$ follows χ^2 pdf with n dof.

Example: goodness-of-fit test variable especially in conjunction with method of least squares.

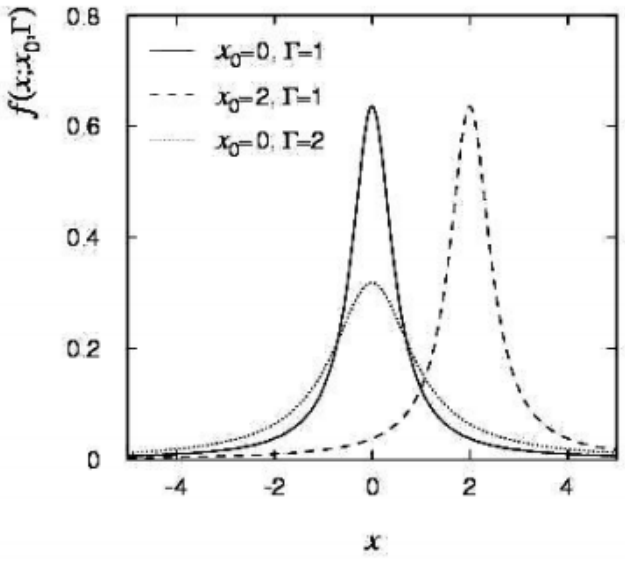
常见PDF: BW

Cauchy (Breit-Wigner) distribution

The Breit-Wigner pdf for the continuous r.v. x is defined by

$$f(x;\Gamma,x_0)=\frac{1}{\pi}\frac{\Gamma/2}{\Gamma^2/4+(x-x_0)^2}$$

($\Gamma=2$, $x_0=0$ is the Cauchy pdf.)



$E[x]$ not well defined, $V[x]\rightarrow\infty$.

x_0 = mode (most probable value)

Γ = full width at half maximum

Example: mass of resonance particle, e.g. ρ, K^*, ϕ^0, ...

Γ = decay rate (inverse of mean lifetime)

常见PDF: Landau

Landau distribution

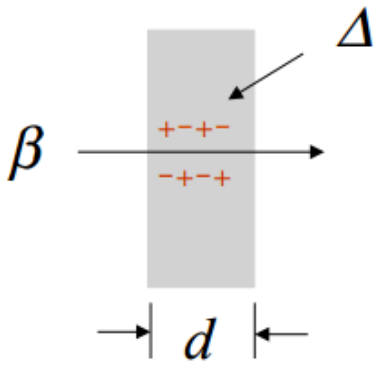
For a charged particle with $\beta = v/c$ traversing a layer of matter of thickness d, the energy loss Δ follows the Landau pdf:

$$f(\Delta;\beta) = \frac{1}{\xi}\phi(\lambda)\ ,$$

$$\phi(\lambda) = \frac{1}{\pi}\int_0^\infty \exp(-u\ln u - \lambda u)\sin\pi u\,du\ ,$$

$$\lambda = \frac{1}{\xi}\left[\Delta - \xi\left(\ln\frac{\xi}{\epsilon'} + 1 - \gamma_E\right)\right]\ ,$$

$$\xi = \frac{2\pi N_A e^4 z^2 \rho \sum Z}{m_e c^2 \sum A}\frac{d}{\beta^2}\ , \qquad \epsilon' = \frac{I^2 \exp\beta^2}{2m_e c^2 \beta^2\gamma^2}\ .$$

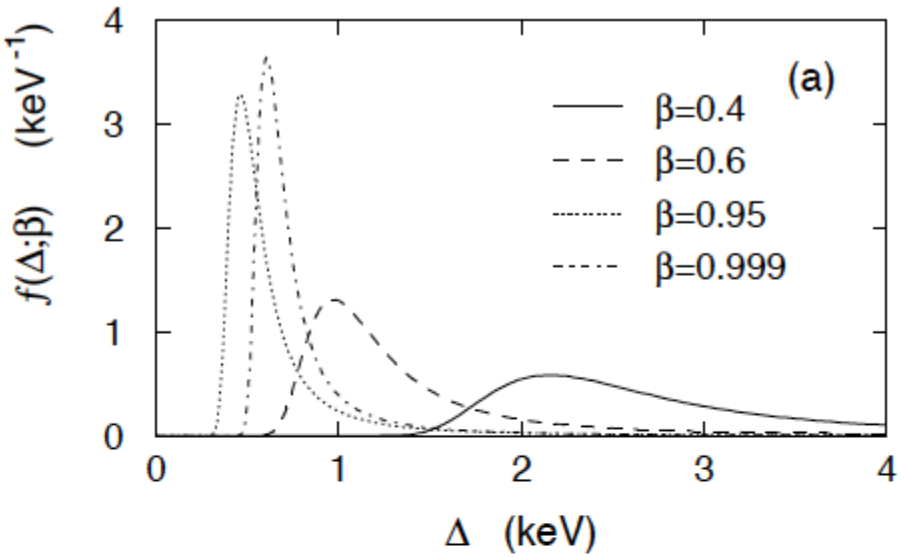


L. Landau, J. Phys. USSR **8** (1944) 201; see also
W. Allison and J. Cobb, Ann. Rev. Nucl. Part. Sci. **30** (1980) 253.

常见PDF: Landau

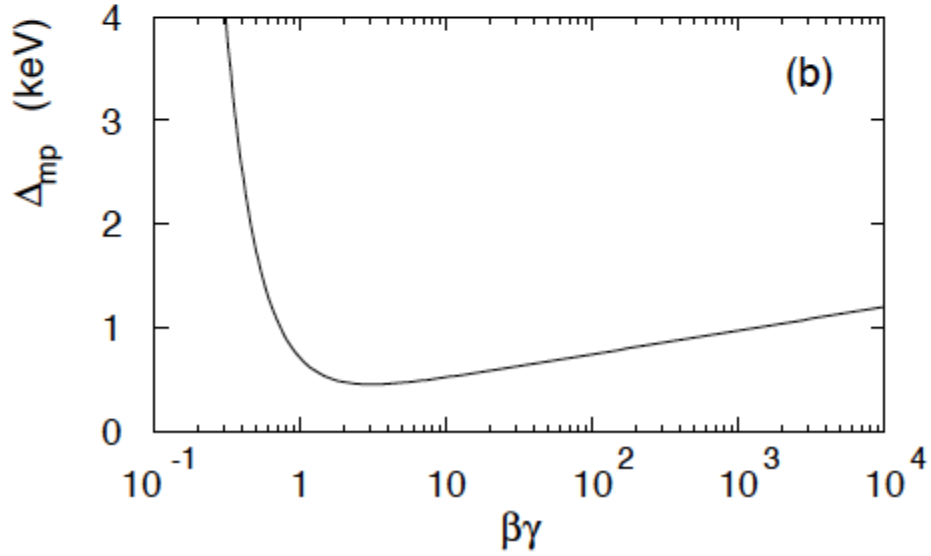
Long 'Landau tail'

→ all moments ∞



Mode (most probable value) sensitive to β,

→ particle i.d.



常见PDF: Beta

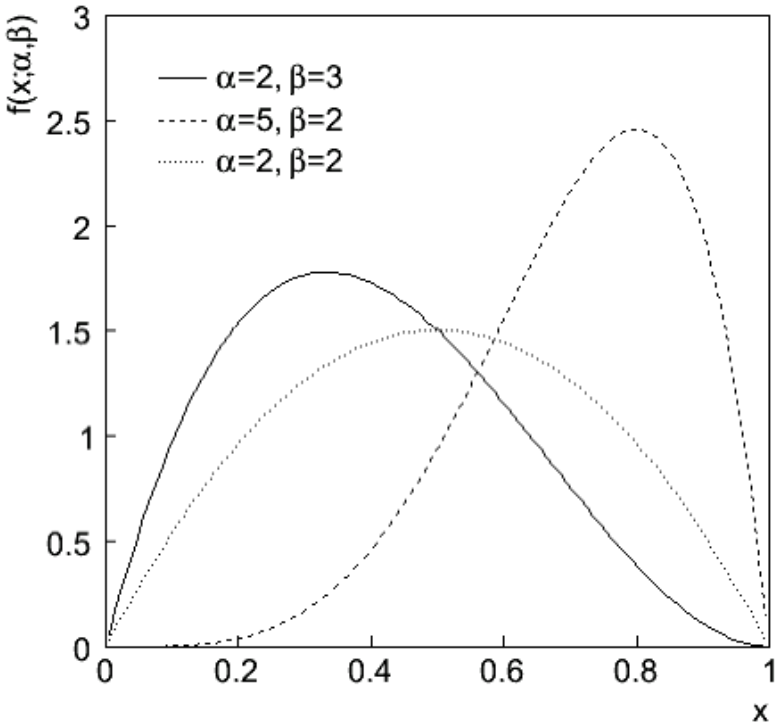
Beta distribution

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

$$E[x] = \frac{\alpha}{\alpha+\beta}$$

$$V[x] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$$

Often used to represent pdf of continuous r.v. nonzero only between finite limits.



常见PDF: Gamma

Gamma distribution

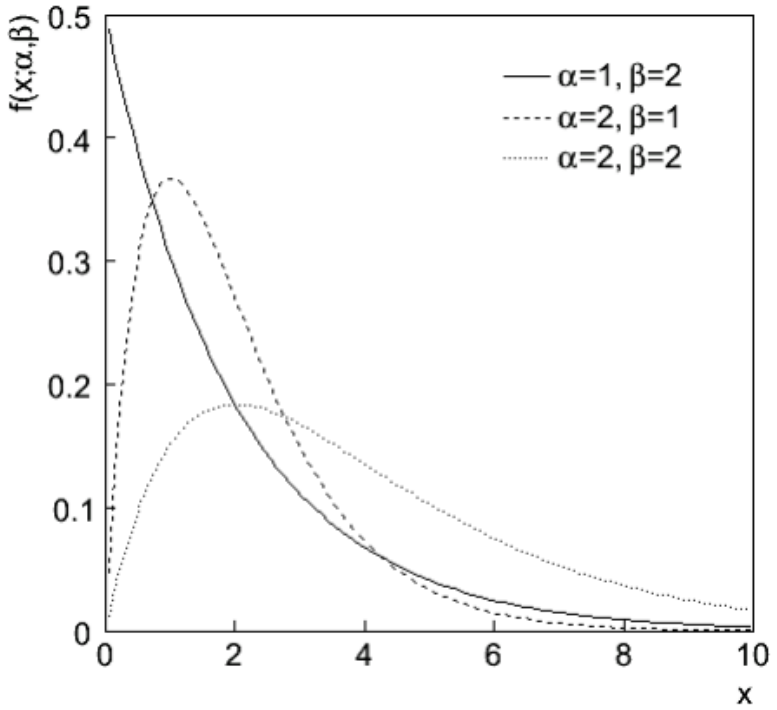
$$f(x;\alpha,\beta) = \frac{1}{\Gamma(\alpha)\beta^{\alpha}}\, x^{\alpha-1}\, e^{-x/\beta}$$

$$E[x] = \alpha\beta$$

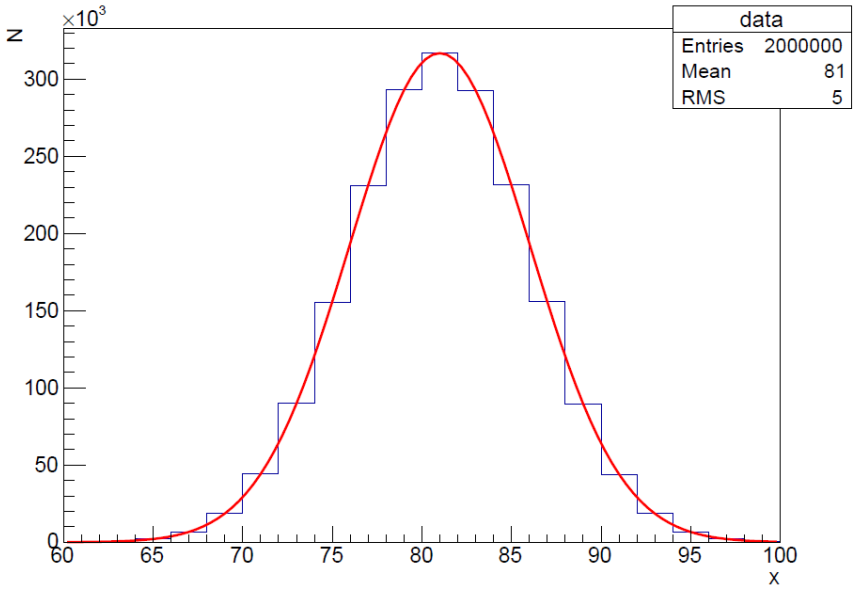
$$V[x] = \alpha\beta^2$$

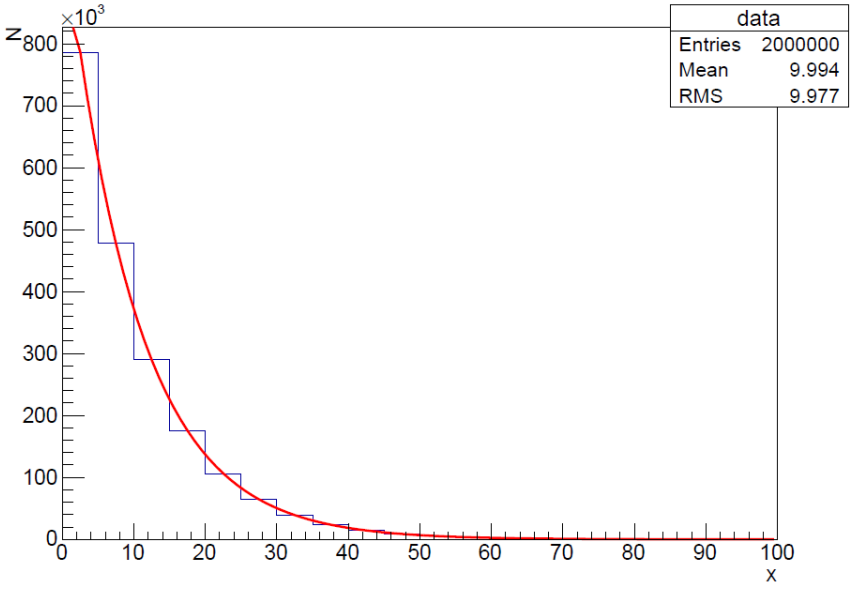
Often used to represent pdf of continuous r.v. nonzero only in $[0,\infty]$.

Also e.g. sum of *n* exponential r.v.s or time until *n*th event in Poisson process ~ Gamma



```
void random2()
{
    gRandom = new TRandom3();
    // create a histogram
      TH1D * hist = new TH1D("data", ";x;N", 20, 0.0,
100.0);
    // fill in the histogram
      for (int i = 0; i < 2000000; ++i)
            hist->Fill(gRandom->Exp(10));
      TCanvas * c1= new TCanvas("c1",
"random",5,5,800,600);
      hist->Fit("expo");
      TF1 *fit = hist->GetFunction("expo");
      Double_t chi2 = fit->GetChisquare();
      cout<<chi2<<endl;
      hist->Draw();
      c1->SaveAs("random2.pdf");}
```

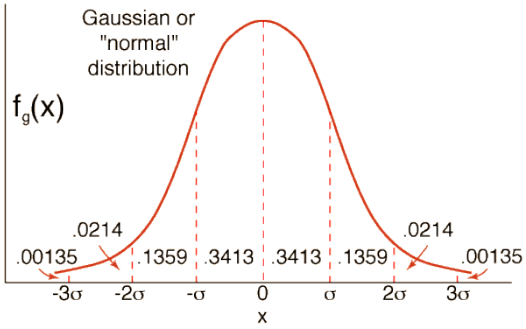


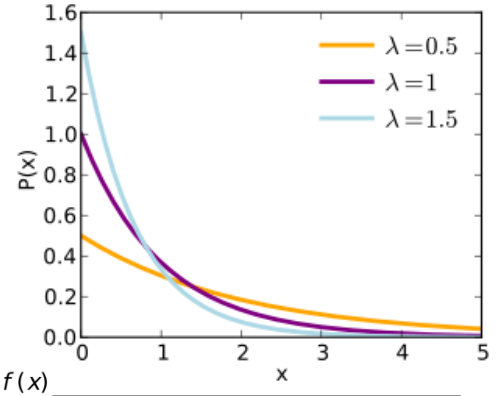


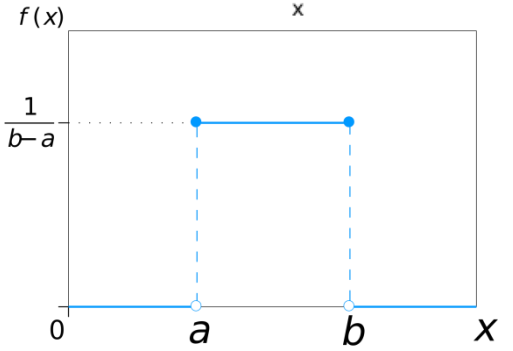
CERN Root 产生 PDF，并拟和

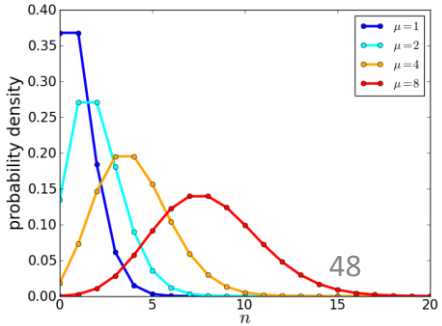
常见PDF

Name	Expression	Mean	Variance	Notable facts
Gaussian f(x;μ,σ)=	$e^{-[(x-\mu)^2/2\sigma^2]}/(2\pi\sigma^2)^{1/2}$	μ	σ^2	Limit of sum of random vars is Gaussian distr.
Negative Exponential f(x;τ)=	$e^{-x/\tau}/\tau$	τ	τ^2	Useful for particle decays; it is also a common prior in Bayesian calc's
Uniform f(x;α,β)=	(β-α)/2 for α<=x<=β 0 otherwise	(α+β)/2	$(\beta-\alpha)^2/12$	Any continuous r.v. can be easily transformed into uniform (→ MC method !)
Poisson f(x;μ)=	$e^{-\mu}\mu^N/N!$	μ	M	Turns into Gaussian for large μ



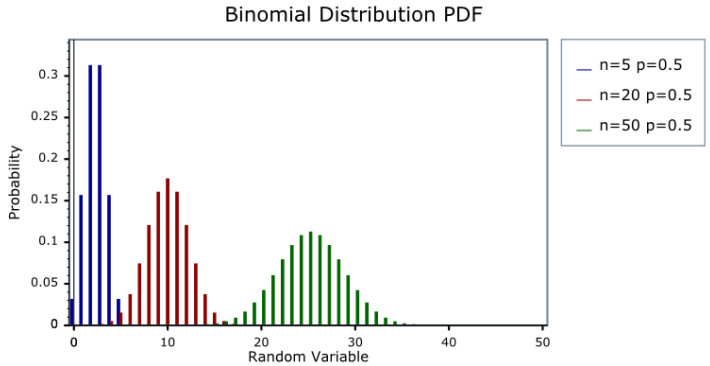


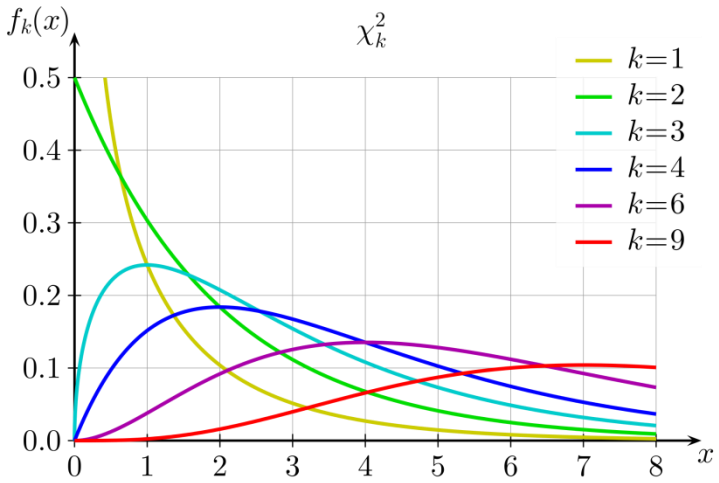


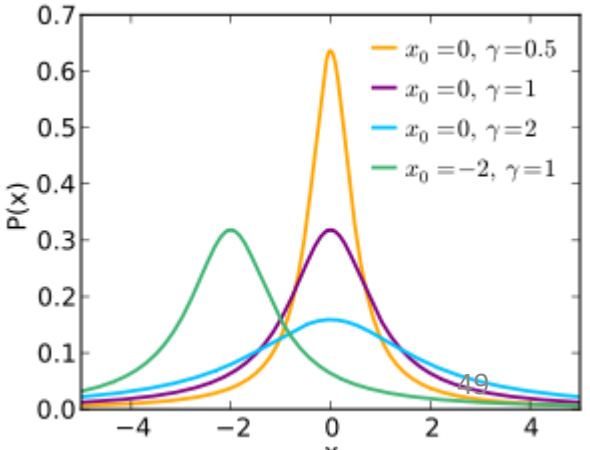


常见PDF

Name	Expression:	Mean	Variance	Fun facts
Binomial f(r;N,p)=	$N!\ p^r(1-p)^{N-r}/[r!(N-r)!]$	Np	Npq	Special case of Multinomial distribution
Chisquare f(x;N)=	$e^{-x/2}\ (x/2)^{N/2-1}/[2\Gamma(N/2)]$	n	2n	Turns into Gaussian for large n (>30 or so)
Cauchy f(x)=	$[\pi\gamma(1+(\frac{x-x_0}{\gamma})^2]^{-1}$	Undefined!	Infinite	AKA Breit-Wigner. Can be manageable if truncated







The Poisson counting experiment

Suppose we do a counting experiment and observe n events.

Events could be from *signal* process or from *background* – we only count the total number.

Poisson model:

$$P(n|s,b) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$

s = mean (i.e., expected) # of signal events

b = mean # of background events

CERN Root
p-value->significance

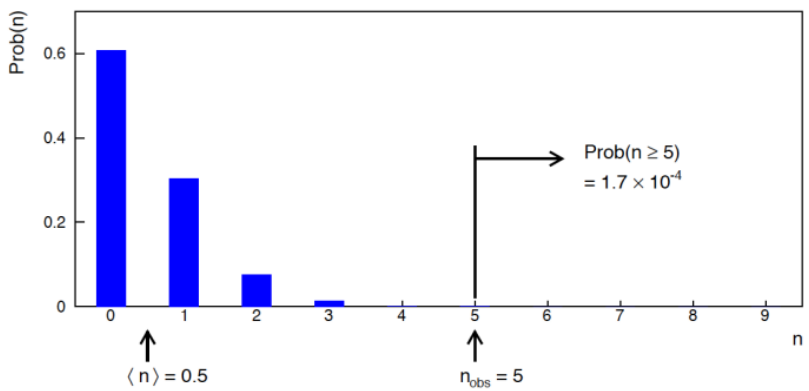
Poisson counting experiment: discovery p-value

Suppose $b = 0.5$ (known), and we observe $n_{obs} = 5$.

Should we claim evidence for a new discovery?

Give p-value for hypothesis $s = 0$:

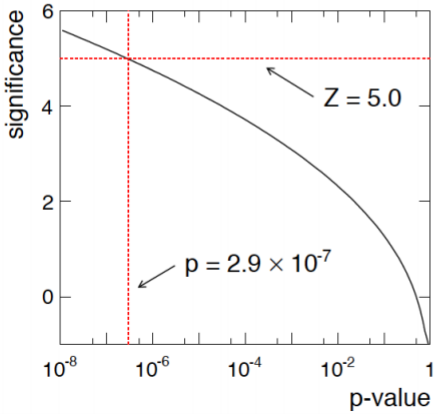
$$p\text{-value} = P(n \geq 5; b = 0.5, s = 0)$$
$$= 1.7 \times 10^{-4} \neq P(s = 0)!$$



Poisson counting experiment: discovery significance

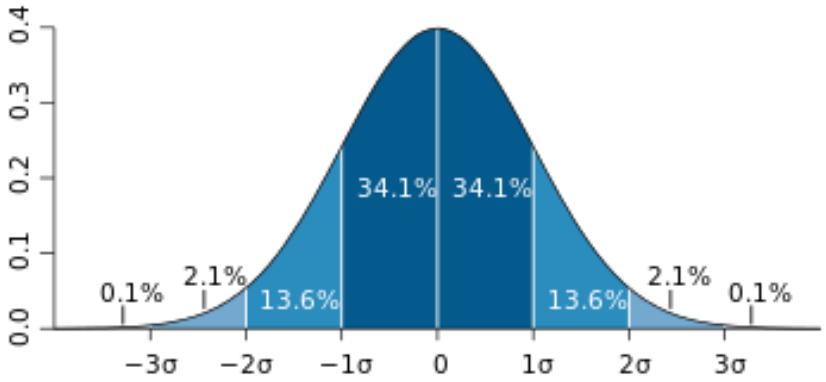
Equivalent significance for $p = 1.7 \times 10^{-4}$: $Z = \Phi^{-1}(1-p) = 3.6$

Often claim discovery if $Z > 5$ ($p < 2.9 \times 10^{-7}$, i.e., a "5-sigma effect")



In fact this tradition should be revisited: p-value intended to quantify probability of a signal-like fluctuation assuming background only; not intended to cover, e.g., hidden systematics, plausibility signal model, compatibility of data with signal, "look-elsewhere effect" (~multiple testing), etc.

Confidence interval	Proportion within	Proportion without	
	Percentage	Percentage	Fraction
0.674 490σ	50%	50%	1 / 2
0.994 458σ	68%	32%	1 / 3.125
1σ	68.268 9492%	31.731 0508%	1 / 3.151 4872
1.281 552σ	80%	20%	1 / 5
1.644 854σ	90%	10%	1 / 10
1.959 964σ	95%	5%	1 / 20
2σ	95.449 9736%	4.550 0264%	1 / 21.977 895
2.575 829σ	99%	1%	1 / 100
3σ	99.730 0204%	0.269 9796%	1 / 370.398
3.290 527σ	99.9%	0.1%	1 / 1000
3.890 592σ	99.99%	0.01%	1 / 10 000
4σ	99.993 666%	0.006 334%	1 / 15 787
4.417 173σ	99.999%	0.001%	1 / 100 000
4.5σ	*99.999 320 465 3751%*	*0.000 679 534 6249%*	*3.4 / 1 000 000 (on each side of mean)*
4.891 638σ	99.9999%	0.0001%	1 / 1 000 000
5σ	99.999 942 6697%	0.000 057 3303%	1 / 1 744 278
5.326 724σ	99.999 99%	0.000 01%	1 / 10 000 000
5.730 729σ	99.999 999%	0.000 001%	1 / 100 000 000
6σ	99.999 999 8027%	0.000 000 1973%	1 / 506 797 346
6.109 410σ	99.999 9999%	0.000 0001%	1 / 1 000 000 000
6.466 951σ	99.999 999 99%	0.000 000 01%	1 / 10 000 000 000
6.806 502σ	99.999 999 999%	0.000 000 001%	1 / 100 000 000 000
7σ	99.999 999 999 7440%	0.000 000 000 256%	1 / 390 682 215 445




```
root [0] RooStats::PValueToSignificance(1.7e-4/2)

RooFit v3.60 -- Developed by Wouter Verkerke and David Kirkby
                Copyright (C) 2000-2013 NIKHEF, University of California & Stanf
ord University
                All rights reserved, please read http://roofit.sourceforge.net/l
icense.txt

(Double_t)3.75987246477831949e+00
root [1] RooStats::PValueToSignificance(1.7e-4)
(Double_t)3.58274690211504376e+00
```

注意one-sided和two-sided

The Nobel Prize in Physics 2013

Photo: A. Mahmoud
François Englert
Prize share: 1/2

Photo: A. Mahmoud
Peter W. Higgs
Prize share: 1/2

The Nobel Prize in Physics 2013 was awarded jointly to François Englert and Peter W. Higgs *"for the theoretical discovery of a mechanism that contributes to our understanding of the origin of mass of subatomic particles, and which recently was confirmed through the discovery of the predicted fundamental particle, by the ATLAS and CMS experiments at CERN's Large Hadron Collider"*

Poisson Significance

$n \sim \text{Poisson}(s+b)$, where

s = expected number of events from signal,

b = expected number of background events.

To test for discovery of signal compute p-value of $s = 0$ hypothesis,

$$p = P(n \geq n_{\text{obs}}|b) = \sum_{n=n_{\text{obs}}}^{\infty} \frac{b^n}{n!} e^{-b} = 1 - F_{\chi^2}(2b; 2n_{\text{obs}})$$

Chi2分布左侧面积

Suppose $b = 0.5$ (known), and we observe $n_{\text{obs}} = 5$.

```
root [0] TMath::Prob(1,10)
(Double_t)9.9982788437004410e-01
root [1] 1-TMath::Prob(1,10)
(const double)1.7211562955893468e-04
root [2] TMath::NormQuantile(1-1.7211562955893468e-04/2)
(Double_t)3.7567771606901061e+00
root [3] RooStats::PValueToSignificance(1.7211562955893468e-04/2)
(Double_t)3.7567771606901061e+00
root [4] TMath::Prob(3.84, 1)
(Double_t)5.0043521248705189e-02
```

Approximate Poisson Significance

$s/\sqrt{b}$ for expected discovery significance

For large $s + b$, $n \to x \sim \text{Gaussian}(\mu,\sigma)$, $\mu = s + b$, $\sigma = \sqrt{(s + b)}$.

For observed value x_{obs}, p-value of $s = 0$ is Prob$(x > x_{obs} \mid s = 0)$,:

$$p_0 = 1 - \Phi\left(\frac{x_{\text{obs}} - b}{\sqrt{b}}\right)$$

Significance for rejecting $s = 0$ is therefore

$$Z_0 = \Phi^{-1}(1 - p_0) = \frac{x_{\text{obs}} - b}{\sqrt{b}}$$

Expected (median) significance assuming signal rate s is

$$\text{median}[Z_0|s + b] = \frac{s}{\sqrt{b}}$$

Likelihood-ratio test and Wilk's Theorem

Suppose we model data $\mathbf{y}$ with a likelihood $L(\boldsymbol{\mu})$ that depends on a set of N parameters $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)$. Define the statistic

$$t_{\boldsymbol{\mu}} = -2 \ln \frac{L(\boldsymbol{\mu})}{L(\hat{\boldsymbol{\mu}})} ,$$

where $\hat{\boldsymbol{\mu}}$ are the ML estimators for $\boldsymbol{\mu}$. The value of $t_{\boldsymbol{\mu}}$ is a measure of how well the hypothesized set of parameters $\boldsymbol{\mu}$ stand in agreement with the data. If the agreement is poor, then $\hat{\boldsymbol{\mu}}$ will be far from $\boldsymbol{\mu}$, the ratio of likelihoods will be low and $t_{\boldsymbol{\mu}}$ will be large. Larger values of $t_{\boldsymbol{\mu}}$ thus indicate increasing incompatibility between the data and the hypothesized $\boldsymbol{\mu}$.

According to Wilks' theorem, if the parameter values $\boldsymbol{\mu}$ are true, then in the asymptotic limit of a large data sample, the pdf of $t_{\boldsymbol{\mu}}$ is a chi-square distribution for N degrees of freedom. We will write this as

$$f(t_{\boldsymbol{\mu}}|\boldsymbol{\mu}) \sim \chi^2_N .$$

Poisson likelihood for parameter s is

$$L(s) = \frac{(s+b)^n}{n!} e^{-(s+b)}$$

Better Approximate Poisson Significance

$$q_0 = -2 \ln \frac{L(0)}{L(\hat{s})} = 2 \left(n \ln \frac{n}{b} + b - n \right) \quad \text{for } n > b, \ 0 \text{ otherwise}$$

For sufficiently large $s + b$, (use Wilks' theorem),

$$Z = \sqrt{2 \left(n \ln \frac{n}{b} + b - n \right)} \quad \text{for } n > b \text{ and } Z = 0 \text{ otherwise.}$$

To find median$[Z|s]$, let $n \rightarrow s + b$ (i.e., the Asimov data set):

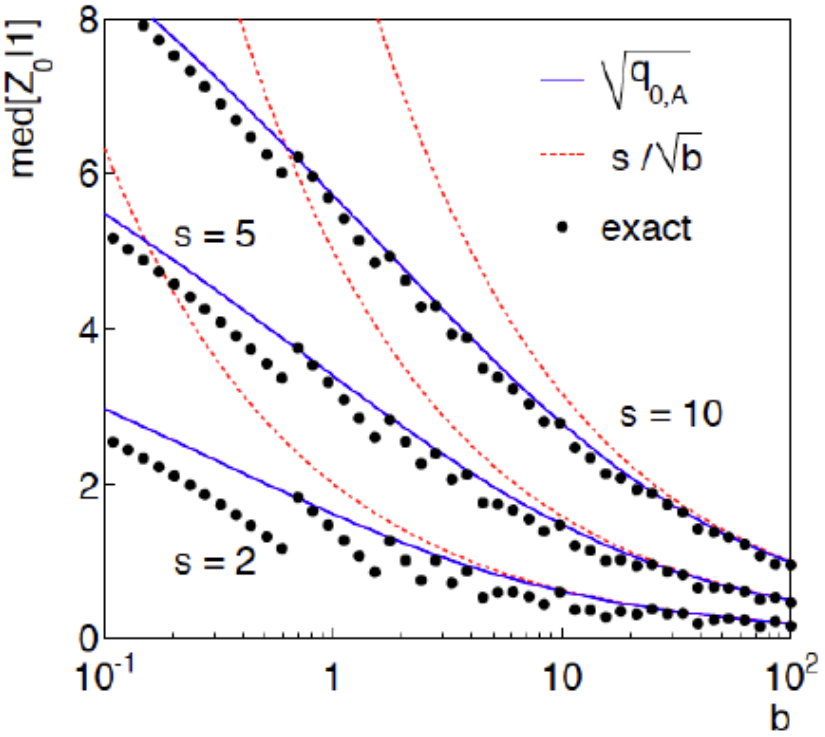
$$Z_A = \sqrt{2 \left((s+b) \ln \left(1 + \frac{s}{b} \right) - s \right)}$$

This reduces to $s/\sqrt{b}$ for $s << b$.

Approximate Poisson Significance

$n \sim \text{Poisson}(s+b)$, median significance, assuming s, of the hypothesis $s = 0$

CCGV, EPJC 71 (2011) 1554, arXiv:1007.1727



"Exact" values from MC, jumps due to discrete data.

Asimov $\sqrt{q_{0,A}}$ good approx. for broad range of s, b.

$s/\sqrt{b}$ only good for $s \ll b$.

$$\frac{5-0.5}{\sqrt{0.5}} \approx 6.364$$

$$\sqrt{2\left(5*\ln(1+4.5/0.5)-4.5\right)} \approx 3.745$$

Upper Limit

Frequentist upper limit on Poisson parameter

Consider again the case of observing $n \sim \text{Poisson}(s + b)$.

Suppose $b = 4.5$, $n_{\text{obs}} = 5$. Find upper limit on s at 95% CL.

Relevant alternative is $s = 0$ (critical region at low n)

p-value of hypothesized s is $P(n \leq n_{\text{obs}}; s, b)$

Upper limit s_{up} at CL $= 1 - \alpha$ found from

$$\alpha = P(n \leq n_{\text{obs}}; s_{\text{up}}, b) = \sum_{n=0}^{n_{\text{obs}}} \frac{(s_{\text{up}} + b)^n}{n!} e^{-(s_{\text{up}}+b)}$$

$$s_{\text{up}} = \frac{1}{2} F_{\chi^2}^{-1}(1 - \alpha; 2(n_{\text{obs}} + 1)) - b$$

$$= \frac{1}{2} F_{\chi^2}^{-1}(0.95; 2(5 + 1)) - 4.5 = 6.0$$

Upper Limit

```
qliphy@qliphy-XPS-8910:~$ root -l
root [0] TMath::ChisquareQuantile(0.95,12)
(Double_t)2.10260698174862313e+01
root [1] TMath::ChisquareQuantile(0.95,12)/2-4.5
(double)6.01303490874311564e+00
root [2] TMath::ChisquareQuantile(0.95,2)/2-0
(double)2.99573227355506377e+00
root [3] TMath::ChisquareQuantile(0.68,2)/2-0
(double)1.13943428318832352e+00
root [4] TMath::ChisquareQuantile(0.99,2)/2-0
(double)4.60517018598789285e+00
```

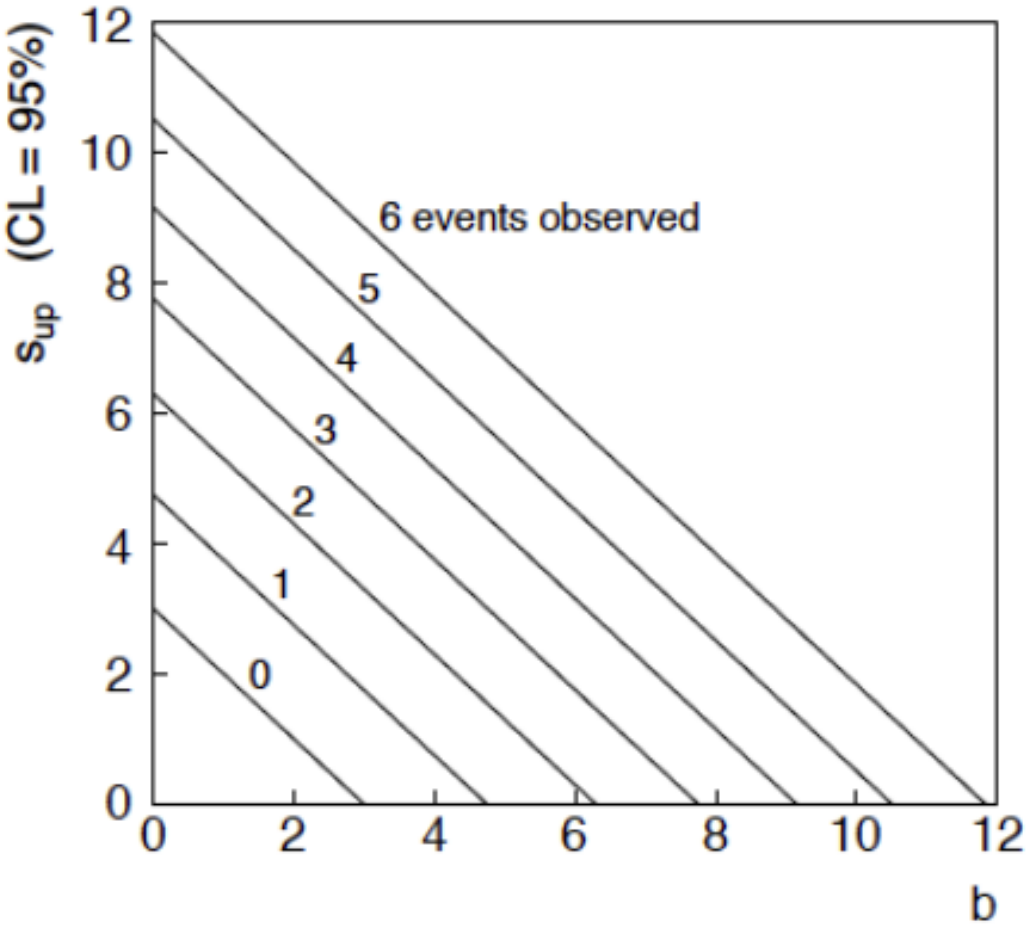
$$L(s) = (s+4.5)^5 / 5! \times \exp(-s-4.5)$$

$$\hat{s} = 0.5 \quad (\text{which maximize } L(s))$$

$$-2\ln\left[\frac{L(s)}{L(\hat{s})}\right] = 3.84 \Rightarrow \sup = 6.24$$

Likelihood ratio,
chi2-distribution
95% confidence level 3.84

For low fluctuation of n formula can give negative result for s_{up}; i.e. confidence interval is empty.



Upper Limit

The Bayesian approach to limits

In Bayesian statistics need to start with 'prior pdf' $\pi(\theta)$, this reflects degree of belief about θ before doing the experiment.

Bayes' theorem tells how our beliefs should be updated in light of the data x:

$$p(\theta|x) = \frac{L(x|\theta)\pi(\theta)}{\int L(x|\theta')\pi(\theta')\,d\theta'} \propto L(x|\theta)\pi(\theta)$$

Integrate posterior pdf $p(\theta|\,x)$ to give interval with any desired probability content.

For e.g. $n \sim$ Poisson(s+b), 95% CL upper limit on s from

$$0.95 = \int_{-\infty}^{s_{\rm up}} p(s|n)\,ds$$

Upper Limit

Include knowledge that $s \geq 0$ by setting prior $\pi(s) = 0$ for $s < 0$.

Could try to reflect 'prior ignorance' with e.g.

$$\pi(s) = \begin{cases} 1 & s \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Not normalized but this is OK as long as $L(s)$ dies off for large s.

Bayesian interval with flat prior for s

Solve to find limit s_{up}:

$$s_{up} = \frac{1}{2} F_{\chi^2}^{-1}[p, 2(n+1)] - b$$

where

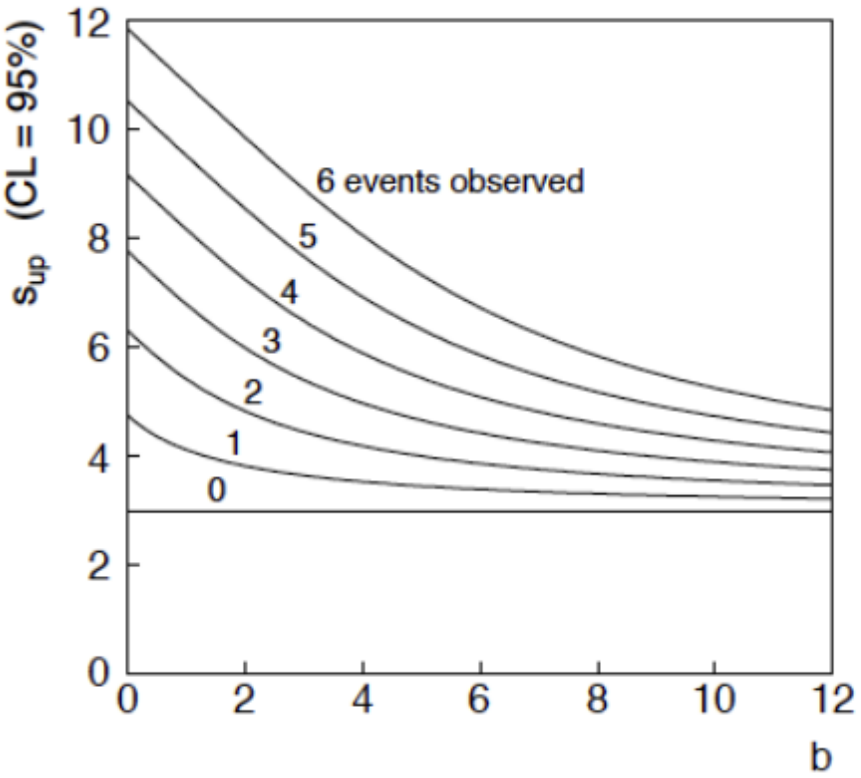
$$p = 1 - \alpha\left(1 - F_{\chi^2}[2b, 2(n+1)]\right)$$

For special case $b = 0$, Bayesian upper limit with flat prior numerically same as one-sided frequentist case ('coincidence').

Upper Limit

For $b > 0$ Bayesian limit is everywhere greater than the (one sided) frequentist upper limit.

Never goes negative. Doesn't depend on b if $n = 0$.



Confidence interval for Poisson Distribution

Confidence interval

The confidence interval for the mean of a Poisson distribution can be expressed using the relationship between the cumulative distribution functions of the Poisson and chi-squared distributions. The chi-squared distribution is itself closely related to the gamma distribution, and this leads to an alternative expression. Given an observation k from a Poisson distribution with mean μ, a confidence interval for μ with confidence level $1-\alpha$ is

$$\tfrac{1}{2}\chi^2(\alpha/2;2k) \leq \mu \leq \tfrac{1}{2}\chi^2(1-\alpha/2;2k+2),$$

or equivalently,

$$F^{-1}(\alpha/2;k,1) \leq \mu \leq F^{-1}(1-\alpha/2;k+1,1),$$

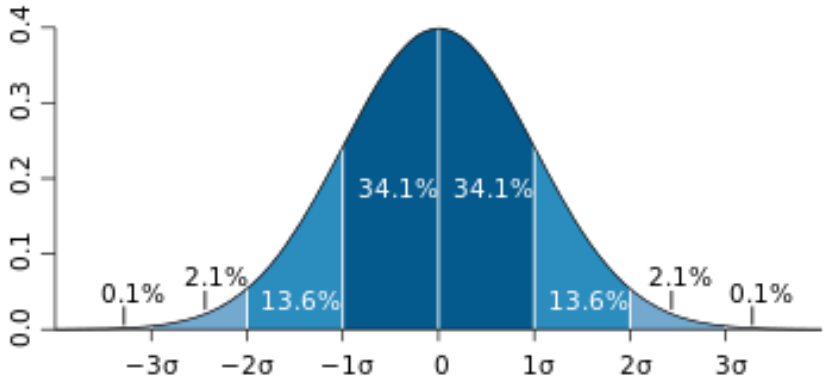
where $\chi^2(p;n)$ is the quantile function (corresponding to a lower tail area p) of the chi-squared distribution with n degrees of freedom and $F^{-1}(p;n,1)$ is the quantile function of a Gamma distribution with shape parameter n and scale parameter 1.[27][39] This interval is 'exact' in the sense that its coverage probability is never less than the nominal $1-\alpha$.

When quantiles of the Gamma distribution are not available, an accurate approximation to this exact interval has been proposed (based on the Wilson–Hilferty transformation):[40]

$$k\left(1-\frac{1}{9k}-\frac{z_{\alpha/2}}{3\sqrt{k}}\right)^3 \leq \mu \leq (k+1)\left(1-\frac{1}{9(k+1)}+\frac{z_{\alpha/2}}{3\sqrt{k+1}}\right)^3,$$

where $z_{\alpha/2}$ denotes the standard normal deviate with upper tail area $\alpha / 2$.

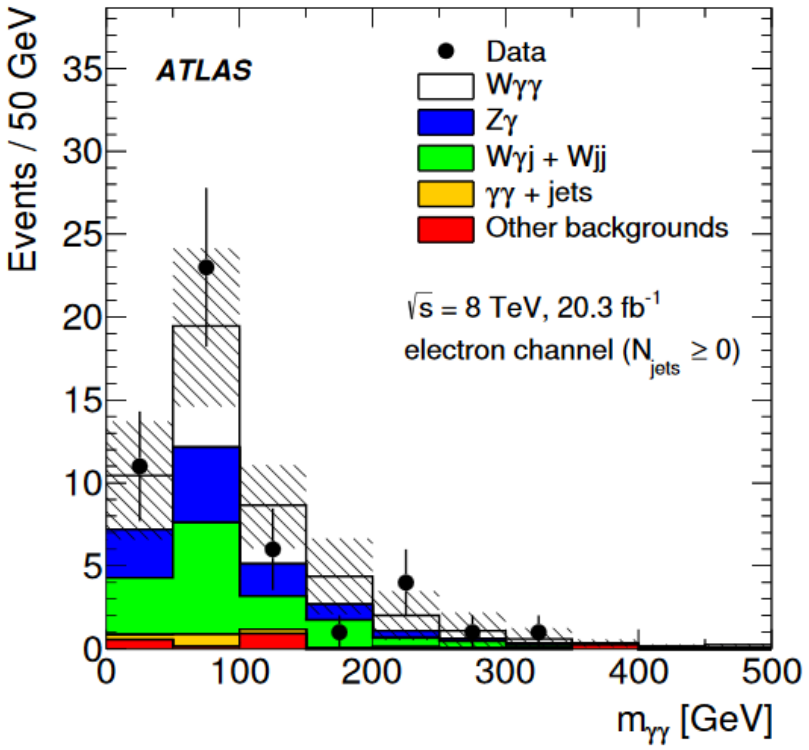
https://en.wikipedia.org/wiki/Poisson_distribution#Confidence_interval

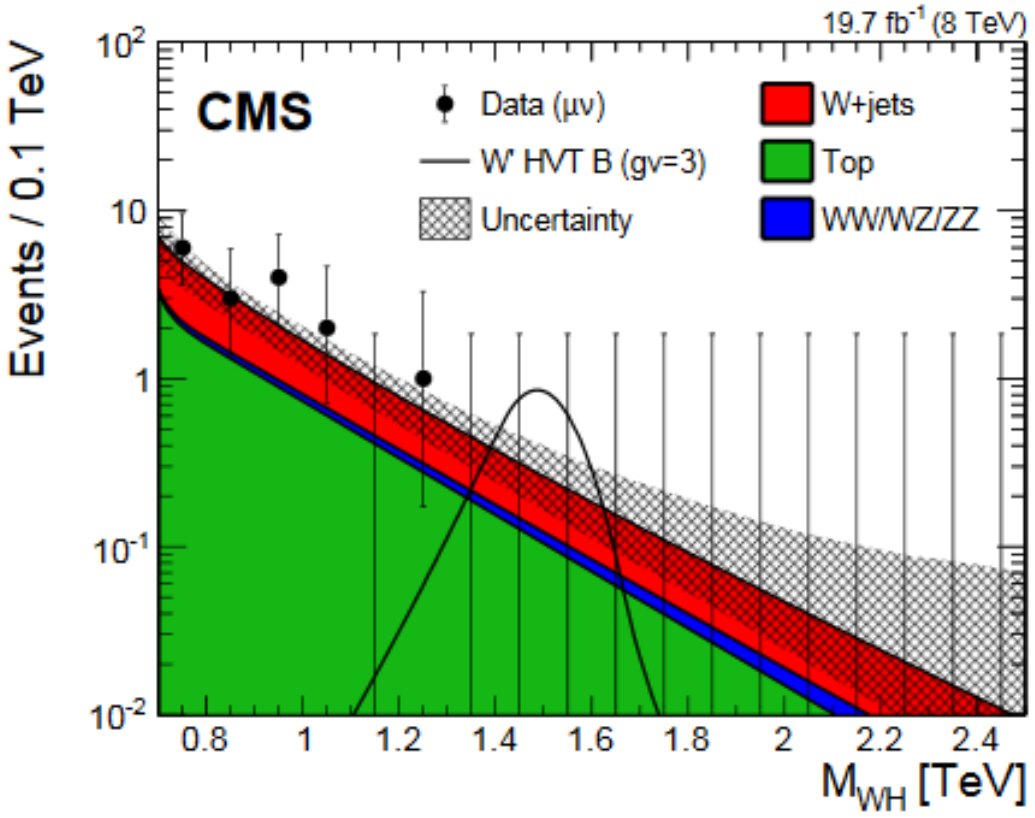



```
qliphy@qliphy-XPS-8910:~$ root -l
root [0] 0.5*TMath::ChisquareQuantile(1-(1-0.682689492)/2,2*0+2)
(const double)1.84102164457785333e+00
root [1] 0.5*TMath::ChisquareQuantile(1-(1-0.682689492),2*0+2)
(const double)1.14787446401725934e+00
root [2]
root [2] 0.5*TMath::ChisquareQuantile(1-(1-0.682689492)/2,2*1+2)
(const double)3.29952655855387977e+00
root [3] 0.5*TMath::ChisquareQuantile((1-0.682689492)/2,2*1)
(const double)1.72753779105486810e-01
root [4]
root [4] 0.5*TMath::ChisquareQuantile(1-(1-0.682689492)/2,2*2+2)
(const double)4.63785962279801200e+00
root [5] 0.5*TMath::ChisquareQuantile((1-0.682689492)/2,2*2)
(const double)7.08185440015169920e-01
```

https://arxiv.org/pdf/1503.03243.pdf

https://arxiv.org/pdf/1601.06431.pdf





左图没有采用**Poisson Errorbar**；右图采用了，结果可对照上一页数字

The central limit theorem

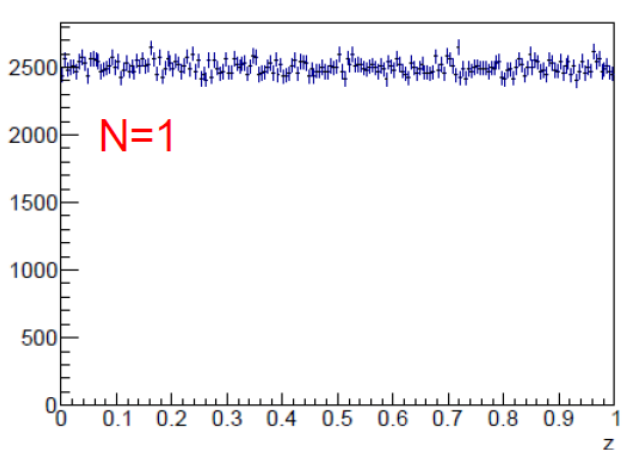
Test central limit theorem

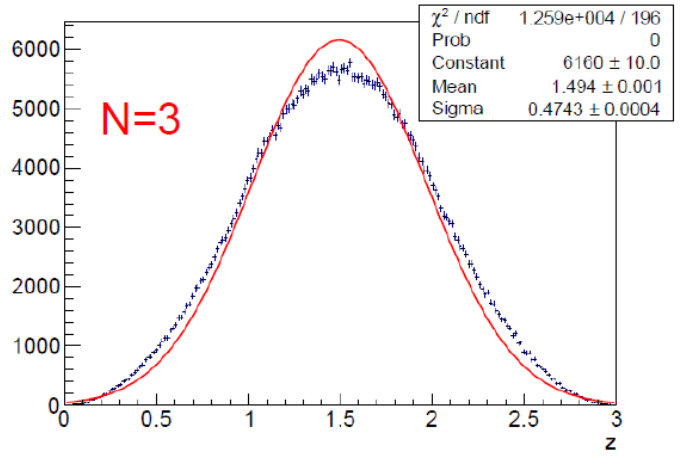
The sum of many random variables will tend to a Gaussian distribution!

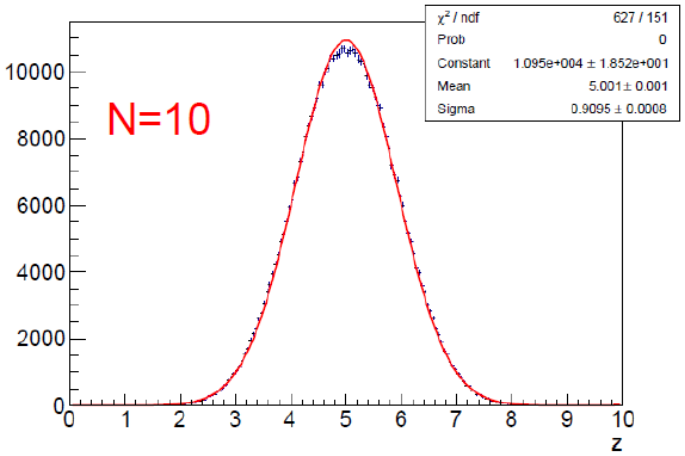
This is at the basis of MUCH of the statistics practice we do

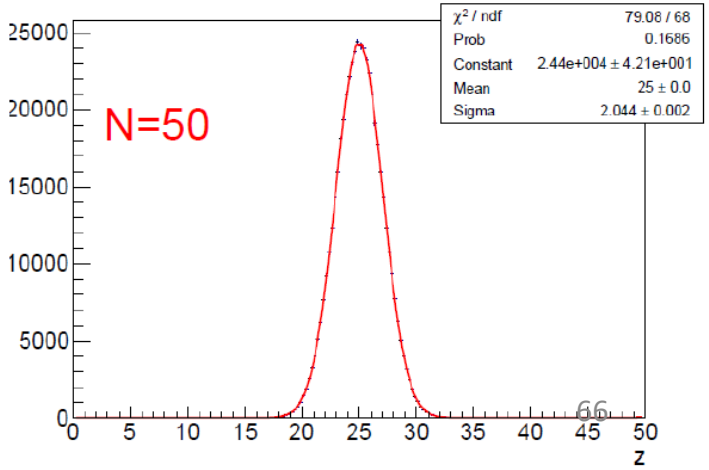
$$z = \sum_{i=1}^{N} x_i$$

Here we take x_i from a Uniform distribution and plot z when N becomes large





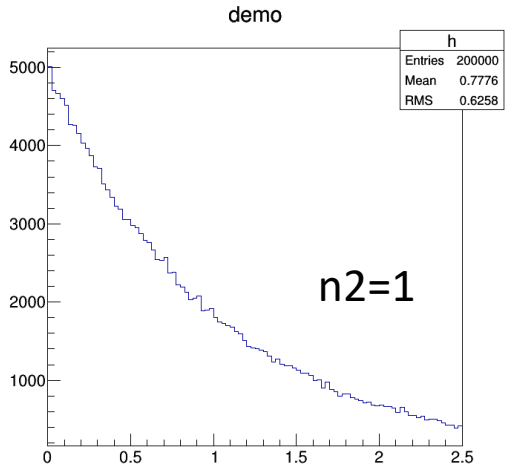


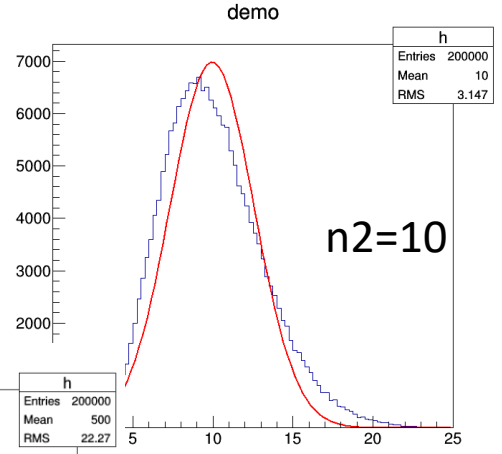


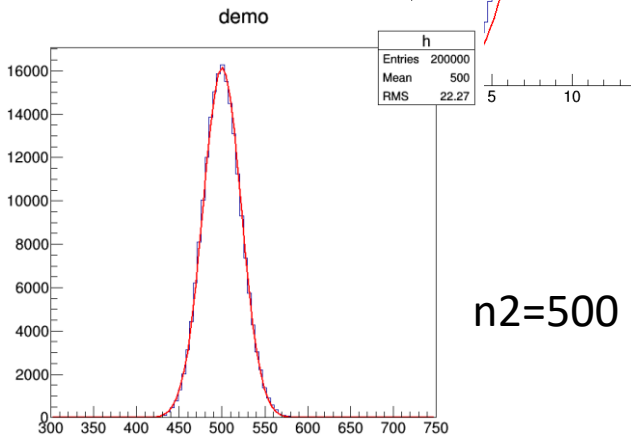
The central limit theorem

Test central limit theorem

```
void random0()
{   TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,700);
      gRandom = new TRandom3(0);
      int n=200000;
      int n2=500;
      double ftot;
      TF1 *f1 = new TF1("f1","gaus",0.6*n2,1.5*n2);
      TH1F *h = new
      TH1F("h","demo",100,0.6*n2,1.5*n2);
      for (int i = 0; i < n; ++i) {
        ftot=0.0;
          for (int j = 0; j < n2; ++j) {
          double x=gRandom->Exp(1);
          ftot+=x; }
        h->Fill(ftot);
      }
      h->Fit("gaus");
      TF1 *fit = h->GetFunction("gaus");
      Double_t chi2 = fit->GetChisquare();
      cout<<chi2<<endl;
      h->Draw();
      c1->SaveAs("Exp_500.png");
}
```







Covariance and correlation

- If you have two random variables x,y you can also define their *covariance*, defined as

$$V_{xy} = E[(x-\mu_x)(y-\mu_y)] = E[xy] - 2\mu_x\mu_y + \mu_x\mu_y =$$

$$= \int_{-\infty}^{+\infty} xyf(x,y)dxdy - \mu_x\mu_y$$

- This allows us to construct a covariance matrix **V**, symmetric, and with positive-defined diagonal elements, the individual variances σ_x^2,σ_y^2:

$$V = \begin{pmatrix} V_{xx} & V_{xy} \\ V_{yx} & V_{yy} \end{pmatrix} = \begin{pmatrix} \sigma_x^2 & r\sigma_x\sigma_y \\ r\sigma_y\sigma_x & \sigma_y^2 \end{pmatrix}$$

- A measure of how x and y are correlated is given by the correlation coefficient r:

$$r = \frac{V_{xy}}{\sigma_x\sigma_y}$$

- Note that if two variables are independent, $f(x,y) = f_x(x)\, f_y(y)$, then $r = V_{xy} = 0$ and $E[xy] = E[x]E[y] = \mu_x\mu_y$.

However, E[xy]=E[x]E[y] is not sufficient for x and y be independent! In everyday usage one speaks of "uncorrelated variables" meaning "independent". In statistical terms,uncorrelated is much weaker than independent!

随机性的统计检验

Random Number Correlation

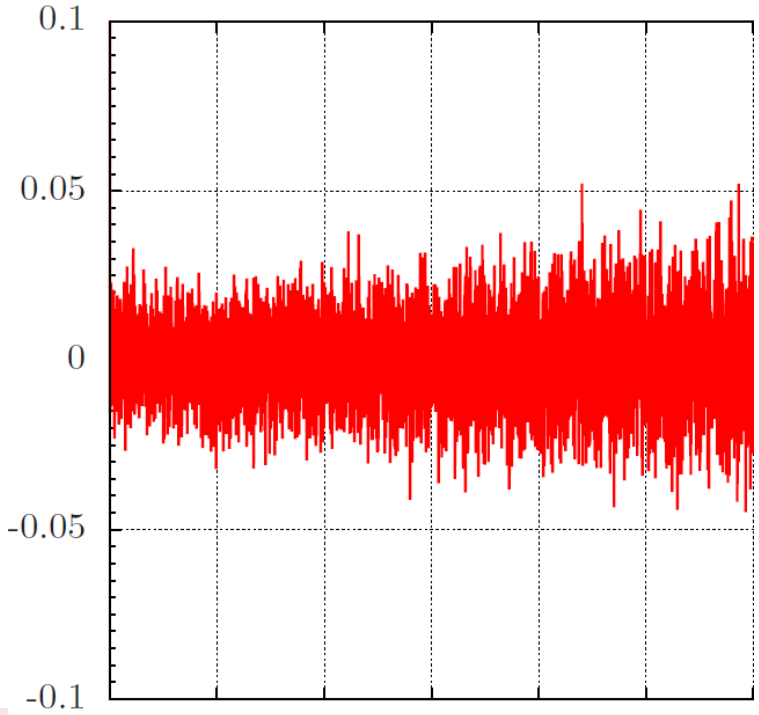
```
#include <iostream>
#include <cmath>
#include <cstdlib>
#include <ctime>
using namespace std;
int main()
{
 double mean=0.0, mean2=0.0;
 double variance=0.0, variance2=0.0;
 double var=0.0;
 int NN;
 cout << "Number =";
 cin >> NN;
 double f[NN], f2[NN];
 for(int i=0; i<NN; i++) {
  f[i]=0.; f2[i]=0.;
 }
 srand(unsigned(time(0)));
 for(int i=0; i<NN; i++) {
  double fx=rand()/(double)RAND_MAX;
  f[i]=fx;
  mean+=fx;
  variance+=fx*fx;
  double fx2=rand()/(double)RAND_MAX;
  f2[i]=fx2;
  mean2+=fx2;
  variance2+=fx2*fx2;
  var+=f[i]*f2[i];
 }
 mean=mean/double(NN);
 mean2=mean2/double(NN);
 variance=sqrt(variance/double(NN)-mean*mean);
 variance2=sqrt(variance2/double(NN)-mean2*mean2);
  var=(var/double(NN)-mean*mean2);

 cout<<"mean= "<<mean<<endl;
 cout<<"mean2= "<<mean2<<endl;
 cout<<"variance= "<<variance<<endl;
 cout<<"variance2= "<<variance2<<endl;
 cout<<"var/v1/v2= "<<var/variance/variance2<<endl;
}
```

随机性的统计检验 2

Random Number Correlation

```
for(int d=0; d<NN; d++) {
  for(int i=0; i<NN-d; i++) {
   dff[d]=dff[d]+(df[i]-mean)*(df[i+d]-mean);
  }
  dff[d]=dff[d]/double(NN-d)/variance/variance;
```



$C_0 = 1$. The non-vanishing of C_k for $k \neq 0$ means that the random numbers are not independent.

$$C_k = \frac{\langle x_{i+k} x_i \rangle - \langle x_i \rangle^2}{\langle x_i^2 \rangle - \langle x_i \rangle^2}, \qquad \langle x_{i+k} x_i \rangle = \frac{1}{N-k} \sum_{i=1}^{N-k} x_i x_{i+k}$$

直接抽样

- **直接抽样法**又称为反函数法,首先需得到连续性随机变量分布函数的解析表达式，然后要求它的分布函数的反函数存在，这种方法使用简单，应用范围最广
- 设连续型随机变量η的分布密度函数为f(x), 数学上它的分布函数为：

$$F(x) = \int_{-\infty}^{x} f(x)dx$$

同时ξ是[0,1]区间上的均匀分布的随机数,那么$\eta = F^{-1}(\xi)$就是满足分布密度函数f(x) 的一个抽样值

- 直接抽样法的缺点是当分布函数F(x)不能从分布密度函数f(x) 解析求出时，或者分布函数的反函数不能用初等函数表示（ 如正态分布或伽马分布），就不能采用这种方法

直接抽样例1：指数函数

- 指数分布的问题可用于描述粒子运动的自由程，粒子衰变寿命或射线与物质作用长度等许多物理问题。它的分布密度函数为：

$$f(x)=\begin{cases}\lambda e^{-\lambda x}, & x>0,\lambda>0\\ 0, & \text{其它.}\end{cases}$$

它的分布函数为

$$F(x)=\int_{-\infty}^{x}f(t)dt=\int_{0}^{x}\lambda e^{-\lambda t}dt=1-e^{-\lambda x}$$

设ξ是[0,1]区间上的均匀分布的随机数，令$\xi=\mathrm{F}(\eta)=1-e^{-\lambda\eta}$, 解此方程得到：

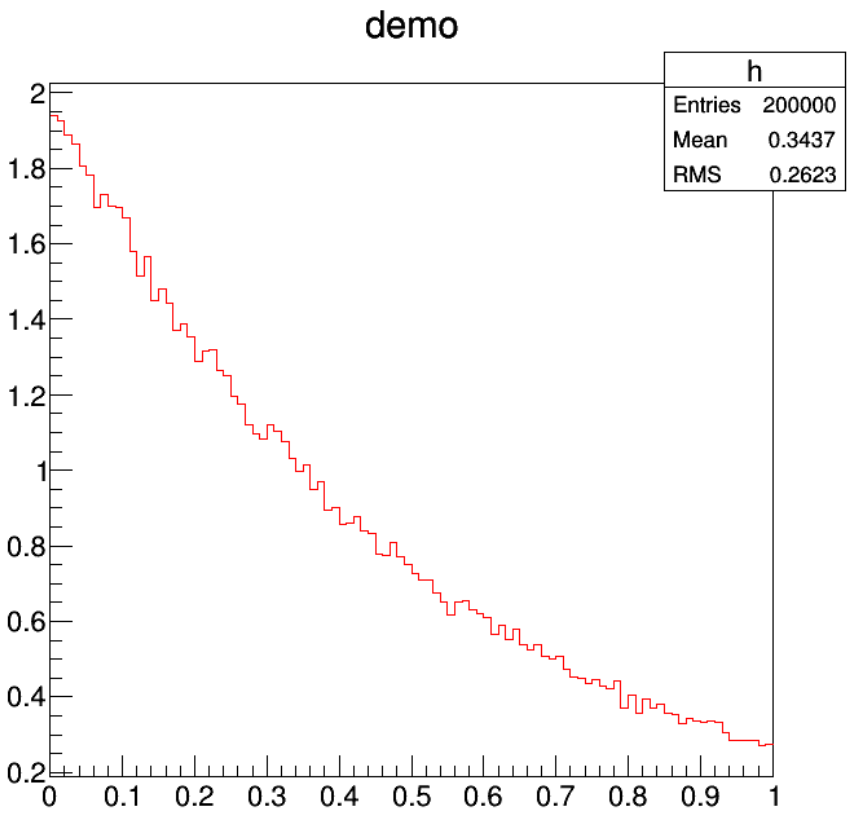
$$\eta=-\frac{1}{\lambda}ln(1-\xi)$$

由于1-ξ和ξ同样服从[0,1]区间的均匀分布，所以：

$$\eta=-\frac{1}{\lambda}ln(\xi)$$

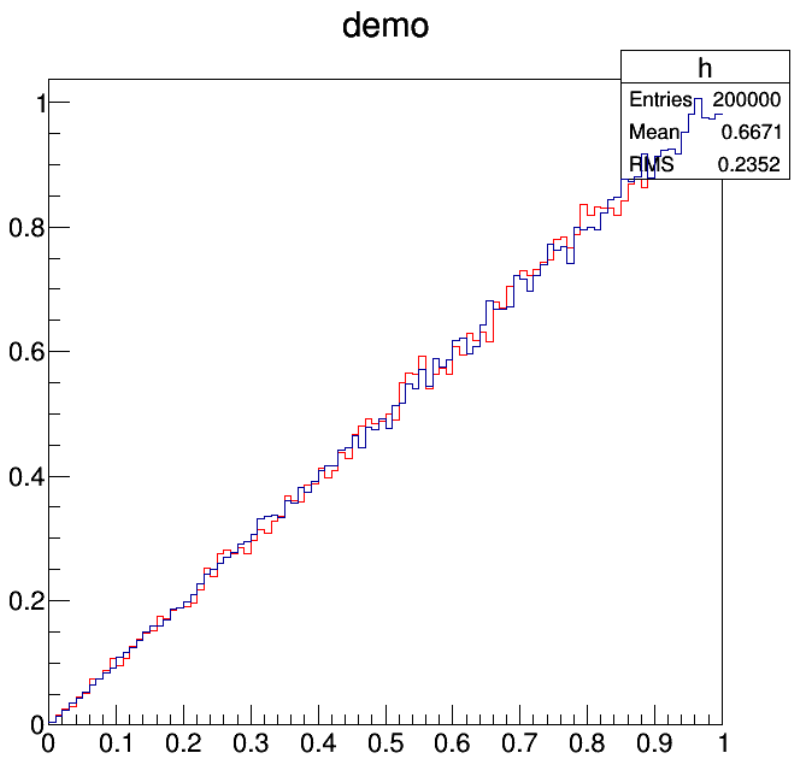
直接抽样例1：指数函数

```
#include "TF1.h"
#include "TMath.h"
void random0b()
{ // a*exp(-ax)dx=d(1-exp(-a*y))
  // a=2.0 here
      TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,700);
       gRandom = new TRandom3(0);
       int n=200000;
       TH1F *h = new TH1F("h","demo",100,0,1.);
       for (int i = 0; i < n; ++i) {
            double y=gRandom->Uniform(0,1);
            double x=-1./2.*TMath::Log(1.0-y);
         h->Fill(x);
       }
       h->SetLineColor(kRed);
         // dN/dX -> density
       h->Scale(1.0/0.01/double(n));
       h->Draw();
}
```



直接抽样例2：f(x)dx=xdx

```
void random0()
{  // xdx=1/2d(x^2)
   TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,700);
    gRandom = new TRandom3(0);
    int n=200000;
    TH1F *h = new TH1F("h","demo",100,0,1.);
    TH1F *h0 = new   TH1F("h0","demo",100,0,1.);
     for (int i = 0; i < n; ++i) {
        double y=gRandom->Uniform(0,1);
        double x=sqrt(2.*y);
         h->Fill(x);
         h0->Fill(y, y);  }
// dN/dX -> density
       h->Scale(1.0/0.01/double(n));
       h0->Scale(1.0/0.01/double(n));
       h->Draw();
       h0->Draw("same");
}
```



直接抽样例3

- 对如下的分布密度函数抽样

$$f(x) = (\frac{\gamma - 1}{x_0^{\gamma} - 1})x^{-\gamma}, \quad x_0 \le x, \gamma > 1$$

此式的分布密度函数对应的分布函数为：

$$F(x) = \frac{\int_{x_0}^{x} f(x)dx}{\int_{x_0}^{+\infty} f(x)dx} = 1 - (\frac{x_0}{x})^{\gamma - 1}$$

在[0,1]区间上随机抽取均匀分布的随机数ξ.令$\xi = \mathrm{F}(\eta) = 1 - (\frac{x_0}{x})^{\gamma-1}$，解此方程，考虑到$1 - \xi$ 和ξ都是[0,1]区间均匀分布的伪随机数，得到分布密度函数的抽样为：

$$\eta = x_0 \eta^{\frac{-1}{\gamma - 1}}$$

请自己检验

变换抽样

- 变换抽样法的基本思想是将一个比较复杂的分布抽样，变换为已经知道的，比较简单的分布抽样。例如要对满足分布函数f(x)的随机变量η 抽样，若要对它直接抽样是比较困难的。
- 这时如果存在另一个随机变量δ,它的分布密度函数为$\phi(y)$,其抽样方法已经掌握，并且也比较简单，那么我们可以设法寻找一个适当的变换关系$x = g(y)$。
- 如果g(y)的反函数存在，记为$g^{-1}(x) = h(x)$，根据概率论的知识，这时x满足的分布密度函数为$\phi(h(x))|h'(x)|$，

$$\phi(y)dy = \phi(h(x))dy = \phi(h(x))\,|\,h'(x)\,|\,dx$$

- 抽样步骤即首先对分布密度函数$\phi(y)$抽样得到δ，通过变换$\eta = g(\delta)$得到满足分布密度函数f(x)的抽样值。

$$f(x)dx = f(x)d[g(y)] = f(g(y))\,|\,g'(y)\,|\,dy$$

二维变换抽样

- 二维情况下的变换抽样法与一维情况完全类似的。假如我们要对满足联合分布密度函数f(x, y)的随机变量η, δ进行抽样。如果我们已经掌握了满足联合分布密度函数g(u, v)的随机变量η'，δ'的抽样方法，则可以寻找一个适当的变换

$$x = g_1(u, v),\ y = g_2(u, v),$$

g1，g2函数的反函数存在，记为

$$u = h_1(x, y),\ v = h_2(x, y)$$

该变换满足如下条件：

$$g(h_1(x, y), h_2(x, y))|J| = f(x, y)$$

$|J|$表示函数变换的Jacobi行列式，这样就可以通过变换式，由满足分布密度函数g(u, v) 的抽样值η'，δ'得到待求得满足分布密度函数f(x, y)的抽样值η，δ。

变换抽样例

- 下面以正态分布抽样为例，看一下变换抽样的具体应用。
- 设随机变量η满足正态分布，它的分布密度函数为：

$$f(x)=\frac{1}{\sqrt{2\pi}}\frac{1}{\sigma}exp(-\frac{(x-\mu)^2}{2\sigma^2}).$$

f(x)记为$N(\mu,\sigma^2)$，其中μ和σ^2 分别是随机变量η 的数学期望值和方差。

- 通常我们只需考虑标准正态分布的抽样方法即可。因为假如随机变量η满足正态分布，随机变量δ满足标准正态分布，则η和δ之间满足关系式 $\eta=\sigma\delta+\mu$
- 标准正态分布密度函数为：

$$f(x)=\frac{1}{\sqrt{2\pi}}exp(-x^2/2)$$

不能用一般函数解析积分求出分布函数F(x)，因而不能直接应用从均匀分布的抽样值变换到标准正态分布的抽样值

二维情形

$$f(x,y)=\frac{1}{2\pi}\exp[-(x^2+y^2)/2]$$

变换抽样例

二维情形

$$f(x,y)=\frac{1}{2\pi}\exp[-(x^2+y^2)/2]$$

- 但是可以采用一个巧妙的办法将两个独立的均匀分布的随机变量u，v变换为标准正态分布的随机变量x，y。这就是做变换：

$$x=\sqrt{-2lnu}cos(2\pi\nu)$$

$$y=\sqrt{-2lnu}sin(2\pi\nu)$$

反解上式得到：

$$u=exp(-\frac{1}{2}(x^2+y^2))=h_1(x,y)$$

$$\nu=\frac{1}{2\pi}tan^{-1}(y/x)=h_2(x,y)$$

$$\begin{vmatrix}\frac{\partial x}{\partial u} & \frac{\partial x}{\partial v}\\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v}\end{vmatrix}=\frac{2\pi}{u}$$

$$f(x,y)=\frac{u}{2\pi}$$

$$f(x,y)dxdy=dudv$$

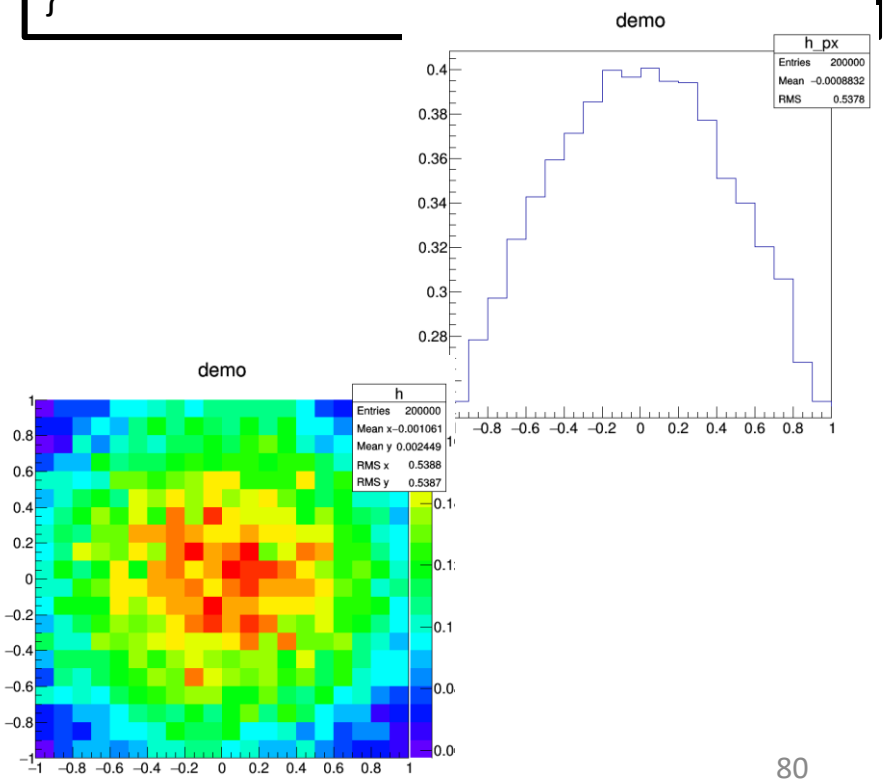
变换抽样例

```
#include "TF1.h"
#include "TMath.h"
void random2()
{ // f(x,y)=1/(2*pi)*exp[-(x^2+y^2)/2]
 // note \int_{-inf}^{inf} exp(-x^2/2)dx=sqrt(2*pi)
 TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,700);
    gRandom = new TRandom3(0);
    gRandom2 = new TRandom3(1);
    double pi  = 3.14159265358;
    int n=200000;
    double u,v, x, y;

   TH2F *h = new TH2F("h","demo",20,-1,1.,20,-1,1.);

  for (int i = 0; i < n; ++i) {
    u=gRandom->Uniform(0,1);
    v=gRandom2->Uniform(0,1);
    x=sqrt(-2.0*TMath::Log(u))*TMath::Cos(2.*pi*v);
    y=sqrt(-2.0*TMath::Log(u))*TMath::Sin(2.*pi*v);
    h->Fill(x, y);     }
```

```
//       h->SetLineColor(kRed);
//       h->Scale(1.0/(0.1*0.1)/double(n));
//       h->Draw("colz");
      TH1D *hx = h.ProjectionX();
       hx->Scale(1.0/(0.1)/double(n));
       hx->Draw();
//       TH1D *hy = h.ProjectionY();
//       hy->Scale(1.0/(0.1)/double(n));
//       hy->Draw();
}
```



变换抽样例：Marsaglia改进

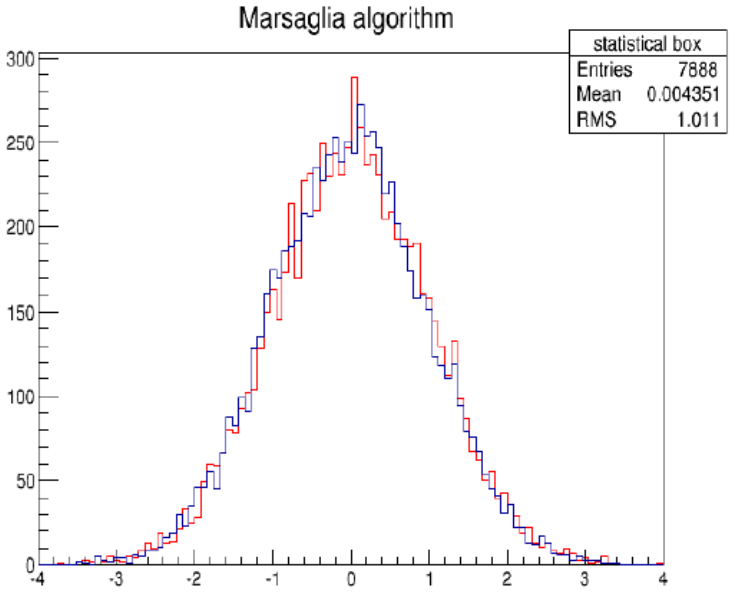
- 上面讲述的两种正态分布的变换抽样方法，前者虽然数学上很严密并且也容易编制程序，但是在用于产生随机数时却不够快，因为需要进行对数，开方，正弦，余弦运算，这些运算耗费机时；
- Marsaglia方法虽然多一些运算,并且在第(3)步时有大约21%的计算耗时被舍弃掉，但是不再做正弦和余弦运算，因而产生随机数的速度要快些

Marsaglia算法过程

上述正态分布的变换抽样法还可以做一些改进,这就是所谓的Marsaglia 方法。其抽样过程由下面四个步骤构成：

- (1) 产生[0,1]区间上的独立均匀分布随机数u和ν;
- (2)计算$w = (2u-1)^2 + (2\nu-1)^2$;
- (3)如果$w > 1$，回到(1)；否则执行(4);
- (4)计算$z = [-2ln(w)/w]^{\frac{1}{2}}$;
- (5)取$x = (2u-1) * z$，$y = (2\nu-1) * z$

Marsaglia算法的高斯抽样图



```
void random2b()
{// f(x,y)=1/(2*pi)*exp[-(x^2+y^2)/2]
// note \int_{-inf}^{inf} exp(-x^2/2)dx=sqrt(2*pi)
//Marsaglia Method

TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,700);
    gRandom = new TRandom3(0);
    gRandom2 = new TRandom3(1);
    double pi  = 3.14159265358;
    int n=200000; int n1=0;
    double u,v,x,y,w,z;
   TH2F *h = new TH2F("h","demo",20,-1,1.,20,-1,1.);
    for (int i = 0; i < n; ++i) {
      u=gRandom->Uniform(0,1);
       v=gRandom2->Uniform(0,1);
       w=(2.*u-1.)*(2.*u-1.)+(2.*v-1.)*(2.*v-1.);
       if(w>1) continue;
       z=sqrt(-2.*TMath::Log(w)/w);
       x=(2.*u-1.)*z;
       y=(2.*v-1.)*z;
       h->Fill(x, y);
       n1++;
      }
```

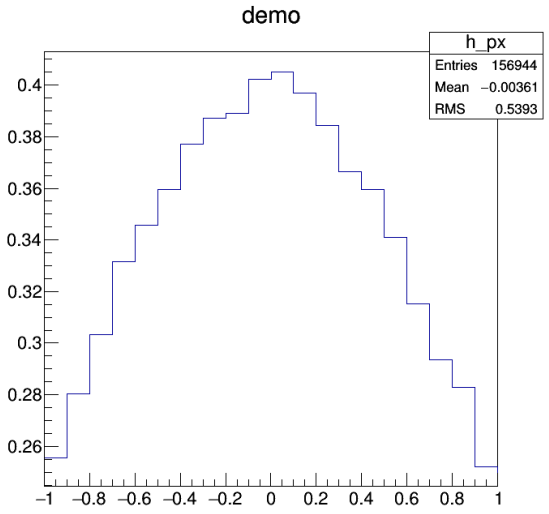
```
//       h->SetLineColor(kRed);
//       h->Scale(1.0/(0.1*0.1)/double(n));
//       h->Draw("colz");


      TH1D *hx = h.ProjectionX();
       hx->Scale(1.0/(0.1)/double(n1));
      hx->Draw();

//       TH1D *hy = h.ProjectionY();
//       hy->Scale(1.0/(0.1)/double(n));
//       hy->Draw();

      cout<<double(n1)/double(n)<<"
"<<1.0/sqrt(2.0*pi)<<endl;

      c1->SaveAs("sample2b_x.png");
}
```



舍选抽样

- 舍选法是冯. 诺曼(Von Neumann) 为克服直接抽样和变换抽样方法的困难最早提出来的。基本思想是按照给定的分布密度函数f(x),对均匀分布的随机数列ξ_n 进行舍选。舍选的原则是在f(x)大的地方，抽取较多的随机数ξ_i，在f(x)小的地方，抽取较少的随机数ξ_i，使得到的字样中ξ_i的分布满足分布密度函数f(x)的要求
- 这种方法对分布密度函数f(x)的上界容易得到的情况，总是可以采用的。优点是计算较为简单，可用于非常复杂的函数，使用广泛。
- 缺点是对f(x)在抽样范围内函数变化很大的时候，抽样效率很低，因为大量的均匀分布抽样点被舍弃了。

第一类舍选抽样

- 设随机变量η在[a,b]上的分布密度函数为f(x)，f(x)在区间[a,b]上的最大值存在，等于$\overset{123}{mac}$

$$L = \max_{x \in [a,b]} f(x)$$

- 采用舍选法的步骤为：
  - (1)选用均匀的[0,1]区间的随机数ξ_1，构造出[a,b]区间上的均匀分布的随机数$\delta = a + (b-a)\xi_1$
  - (2)再选取独立的均匀分布于[0,1]区间上的随机数ξ_2, 判断$L\xi_2 \leq f(\delta)$是否满足。如满足上面不等式，则执行(3)；如不满足，则返回到步骤(1)；
  - (3)选取$\eta = \delta$作为一个抽样值；
- 重复上面三个步骤，就可以产生出随机数序列ξ_n，它满足分布密度函数f(x)
- 舍选抽样的第二个步骤判断不等式$\xi_2 \leq \lambda f(\delta)$是为了保证随机点$(\delta, L\xi_2)$落在f(x)曲线的下面

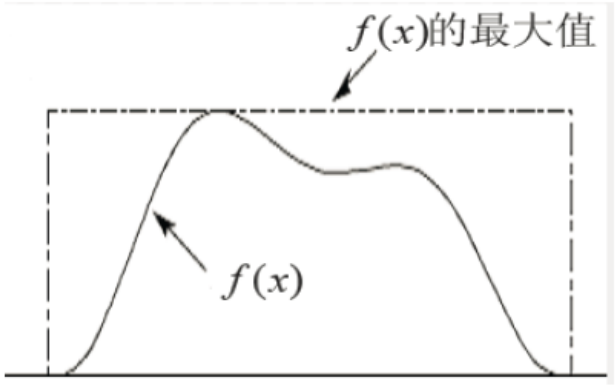
第一类舍选抽样

- 抽样效率的定义：

$$\epsilon = \frac{\text{满足判定条件的抽样点数}}{\text{产生的抽样点的总数}}$$

- 显然随机点$(\delta, L\xi_2)$落在曲线f(x)以下才被接受，并且所有产生的二维随机点都落在面积等于$L(b-a)$的矩形区域内，因此，可以算出采用该方法的抽样效率为：

$$E = \frac{\int_a^b f(x)dx}{L(b-a)} = \frac{1}{L(b-a)}$$



- 可以看到我们希望效率能够越高越好。如果L很大(即f(x))具有高峰，则此舍选抽样效率就不高(为此下面会介绍第二类舍选法)

第一类舍选抽样举例

- 对随机变量 η 抽样。它的分布密度函数为:

$$f(x)=\begin{cases}2x, & 0\le x\le 1\\ 0, & \text{其它.}\end{cases}$$

- 如果用直接抽样法，首先求出分布函数

$$F(x)=x^2$$

抽取在[0,1]区间上的均匀分布函数的随机数ξ。令

$$\xi=x^2$$

则有

$$x=\sqrt{\xi}$$

x为η的子样的一个个体。但是开方运算量较大，可改用舍选法来做。

$$L=\max_{x\in[0,1]} f(x)=\max_{x\in[0,1]} 2x=2$$

第一类舍选抽样举例

- 依照第一类舍选法的步骤：
  - 1. 依次产生独立的[0,1]区间上的均匀分布的随机数ξ_1, ξ_2
  - 2.判断$\xi_2 \leq \frac{1}{L} f(\xi_1) = \xi_1$ 是否成立
  - 3.若成立，则取$x = \xi_1$
  - 4.若上面不等式不成立，可以再产生一组ξ_1, ξ_2进行重复试验。但实际上，因为ξ_1, ξ_2本来就是任意的，如果$\xi_2 \leq \xi_1$不成立，必有$\xi_1 \leq \xi_2$。所以若$\xi_2 \leq \xi_1$不成立，只要将ξ_1和ξ_1 互换以下，这个不等式就必定成立。所以可以取 $x = \max(\xi_1, \xi_2)$
- 一般高次幂的情况。设 η 满足分布密度函数

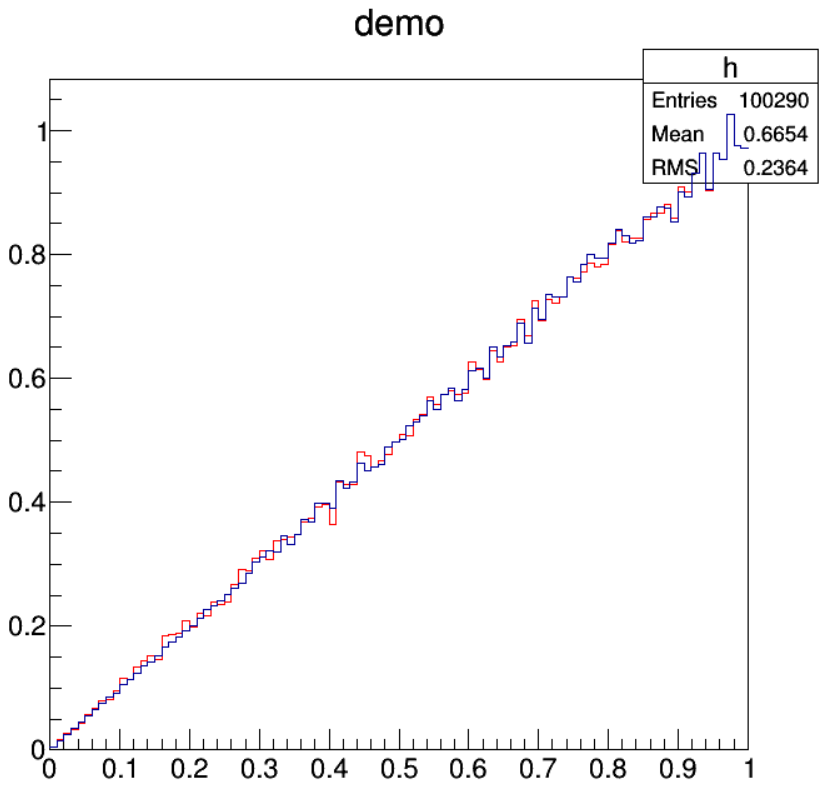
$$f(x) = \begin{cases} nx^{n-1}, & x \in [0,1], n = 1, 2... \\ 0, & \text{其它.} \end{cases}$$

用舍选法抽样，依次产生独立的[0,1]区间上的均匀分布的随机数$\xi_1, \xi_2 ... \xi_n$, 则取

$$x = \max(\xi_1, \xi_2 ... \xi_n)$$

第一类舍选抽样举例

```
#include "TF1.h"
#include "TMath.h"
void random3()
{ // xdx=1/2d(x^2)
   TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,700);
      gRandom = new TRandom3(0);
      gRandom2 = new TRandom3(1);
      int n=200000;
      TH1F *h = new TH1F("h","demo",100,0,1.);
      TH1F *h0 = new
TH1F("h0","demo",100,0,1.);
      for (int i = 0; i < n; ++i) {
        double x=gRandom->Uniform(0,1);
        double y=gRandom2->Uniform(0,1);
        if(y<=x) {h->Fill(x);}
        h0->Fill(x, x);     }
      h->SetLineColor(kRed);
      h->Scale(1.0/0.01/double(n));
      h0->Scale(1.0/0.01/double(n));
      h->Draw();
      h0->Draw("same");
      c1->SaveAs("sample_3_x.png"); }
```



第二类舍选抽样

- 首先，第二类舍选抽样方法实质上是后面讲到的第三类舍选法的特殊情况；
- 为了实现从已知分布密度函数f(x)抽样，选取与f(x) 取值范围相同的分布密度函数h(x)，并且f(x)可以写为

$$f(x) = L \cdot \frac{f(x)}{Lh(x)} h(x) = Lg(x)h(x)$$

其中L为常数，如果$L = \max \frac{f(x)}{h(x)}$。g(x) 可视为另一个随机变量的分布密度函数。

- 对满足分布函数f(x)的随机变量η的抽样，可以采用下面的步骤来实现
  - (1)由h(x)分布密度函数抽样得到η_h
  - (2)判别$\xi \leq g(\eta_h)$不等式是否成立。如果不成立，则返回到步骤(1)；
  - (3)选取$\eta = \eta_h$作为服从分布密度函数f(x)的一个抽样值。
- 第二类舍选法的抽样效率为$E = \frac{1}{L}$

第二类舍选抽样

- 第二类舍选法也即从h(x)中抽样η_h，并以$\xi \le g(\eta_h) = \frac{f(\eta_h)}{Lh(\eta_h)}$的条件概率接受它，下面证明$\eta_h$ 服从分布密度函数f(x)
- 证明：对于任意的x

$$P(x \le \eta \le x+dx) = P(x \le \eta_h \le x+dx \mid \xi \le \frac{f(\eta_h)}{Lh(\eta_h)})$$

由Bayes公式：

$$P(A_i \mid B) = \frac{P(A_i B)}{P(B)}$$

有：

$$= \frac{P(x \le \eta_h \le x+dx, \xi \le \frac{f(\eta_h)}{Lh(\eta_h)})}{P(\xi \le \frac{f(\eta_h)}{Lh(\eta_h)})} \quad = \frac{\int_x^{x+dx} \int_0^{\frac{f(\eta_h)}{Lh(\eta_h)}} h(\eta_h) d\eta_h d\xi}{\int_{-\infty}^{+\infty} \int_0^{\frac{f(\eta_h)}{Lh(\eta_h)}} h(\eta_h) d\eta_h d\xi}$$

$$= \frac{\int_x^{x+dx} \frac{f(\eta_h)}{Lh(\eta_h)} h(\eta_h) d\eta_h}{\int_{-\infty}^{+\infty} \frac{f(\eta_h)}{Lh(\eta_h)} h(\eta_h) d\eta_h} = \frac{\int_x^{x+dx} f(\eta_h) d\eta_h}{\int_{-\infty}^{+\infty} f(\eta_h) d\eta_h} = f(x)dx$$

第二类舍选抽样

- 使用第二类舍选法时，要注意：
  - (1)选取h(x)时要使得h(x)容易抽样；
  - (2)L值要尽量小，因为L小能提高抽样效率；
- 抽样效率的证明：

$$\epsilon = P(\xi \le \frac{f(\eta_h)}{L \cdot h(\eta_h)}) = \int_{-\infty}^{+\infty} \frac{f(\eta_h)}{L \cdot h(\eta_h)} h(\eta_h) d\eta_h = \frac{1}{L}$$

- 所以，L越小，抽样效率越高

- 例 标准正态分布密度函数可以写为

$$f(x) = \frac{1}{\sqrt{2\pi}} exp(-\frac{x^2}{2}) \quad (-\infty < x < +\infty)$$

- 由于相应的分布密度函数不存在反函数，故可以采用舍选法，令

$$h(x) \equiv e^{-x} \quad (-\infty < x < +\infty)$$

$$g(x) \equiv exp\{-(x-1)^2/2\} \quad (-\infty < x < +\infty)$$

- 由 $L = \max_{x \in [0,1]} \frac{f(x)}{h(x)}$ 算得：

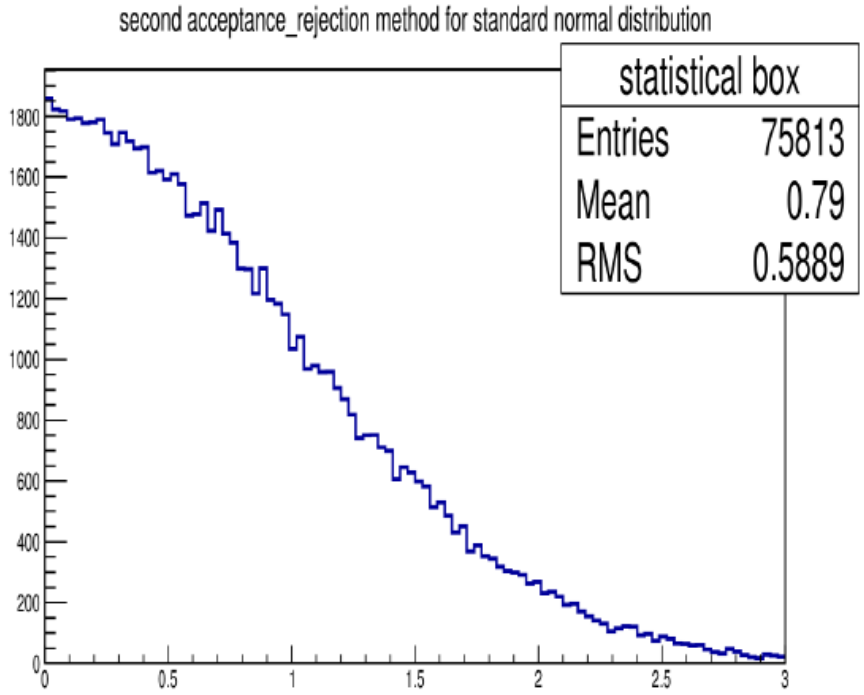
$$L = \sqrt{\frac{e}{2\pi}}$$

- 由于f(x)是x的偶函数，因而可以在$(0, +\infty)$ 区域上抽样后反射到$(-\infty, 0)$区间上的抽样值。这样我们可以只考虑$(0, +\infty)$区域的抽样

下面的例子中，由于只考虑了0，+inf，效率实际为1/(2L)

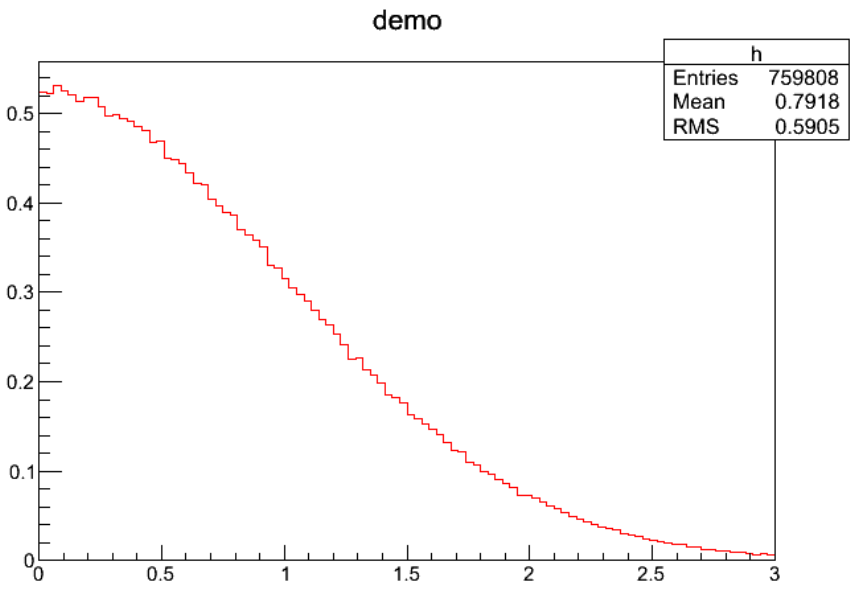
第二类舍选抽样

- (1)对h(x)的抽样可以用直接抽样法。由$\eta_h = -ln\xi_1$ 算出η_h的值；
- (2)然后产生随机数ξ_2，判别$\xi_2 \leq g(\eta_h)$是否成立，也即判断不等式$(\eta_h - 1)^2 \leq -2ln\xi_2$是否成立
- (3)如不成立，则舍弃，再重新由h(x)直接抽样；
- (4)如成立，则抽样值为η_h。该抽样的效率为$E = \sqrt{\frac{\pi}{2e}} \approx 76\%$。



示例

```
#include "TF1.h"
#include "TMath.h"
void random3b()
{// f(x)=1/sqrt(2*pi)*exp(-x^2/2), 0<x<inf
// h(x)=exp(-x)
TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,500);
      gRandom = new TRandom3(0);
      gRandom2 = new TRandom3(1);
      double pi  = 3.14159265358;
      int n=1000000; int n1=0;
      double u,v;
      double L=sqrt(exp(1.0)/2.0/pi);
      TH1F *h = new TH1F("h","demo",100,0,3
      for (int i = 0; i < n; ++i) {
        double k1=gRandom->Uniform(0,1);
        double k2=gRandom2->Uniform(0,1);
        x=-TMath::Log(k1);
        v=L*exp(-(x-1.)*(x-1.)/2.0)/L;
        if(k2<=v) {h->Fill(x,L); n1++;}     }
      h->SetLineColor(kRed);
      h->Scale(1.0/0.03/double(n));
      h->Draw();
      cout<<double(n1)/double(n)<<"
"<<1.0/L/2.0<<endl; }
```



加抽样

- 加抽样方法是对如下加分布给出的一种抽样方法：

$$f(x) = \sum_n P_n h_n(x)$$

- 其中$0 < P_n < 1$，$\sum_n P_n = 1$，且$h_n(x)$为与参数n 有关的分布密度函数
- 加分布抽样方法为：首先抽样确定n值，然后由$h_n(x)$中抽样x,具体：
  - (1)取[0,1]区间上均匀分布随机数ξ，解下面的不等式求得n。

$$\sum_{i=1}^{n-1} P_i < \xi \leq \sum_{i=1}^{n} P_i$$

  - (2)找到对应的$h_n(x)$，并对其抽样，得到最后的抽样值$\eta = \eta_{h_n}$
- 明显地概率P_n决定我们取哪个概率密度函数的分量进行抽样；
- 这样的抽样步骤实际上是蒙特卡洛迭加原则的应用

加抽样

- 例： 球壳均匀分布的抽样，设球壳内外半径分别为R_0和R_1，球壳内一点到球心距离为r，则r的分布函数为

$$f(r)=\frac{3r^2}{R_1^3-R_0^3},\quad R_0\le r\le R_1$$

- 解用直接抽样法，取$[0,1]$区间上的均匀分布随机数ξ，
  则$\eta=[(R_1^3-R_0^3)\xi+R_0^3]^{\frac{1}{3}}$ 的取值就是以f(r)分布的一个抽样值。
- 为了避免用运算量较大的开方运算，可以改用加分布抽样。令

$$r=(R_1-R_0)x+R_0,\quad \lambda=R_1^2+R_1R_0+R_0^2$$

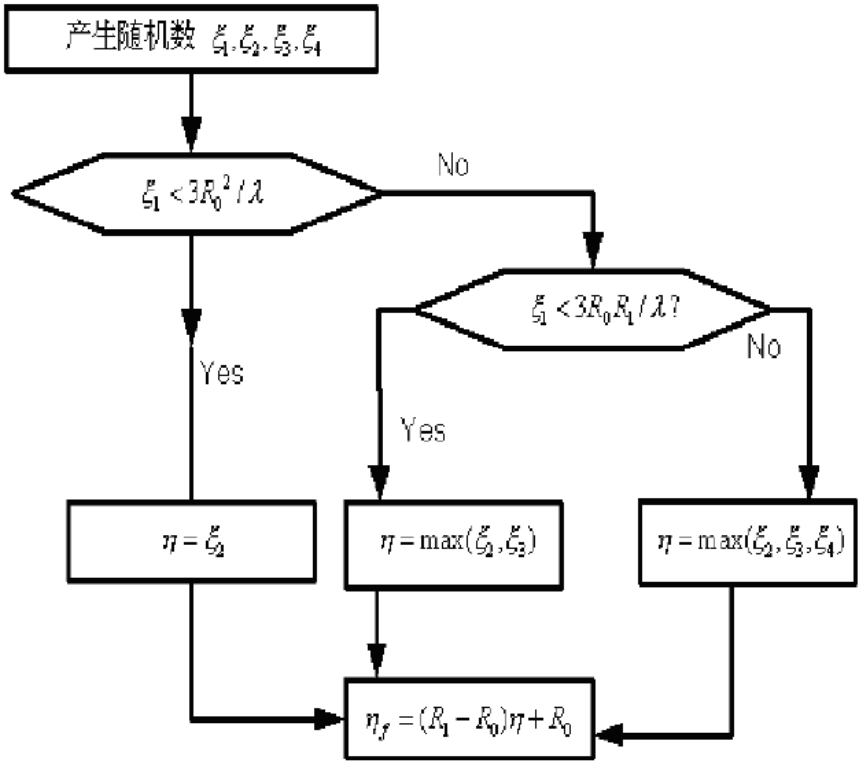
- 整理f(x)可以变成：

$$f(x)=\frac{(R_1-R_0)^2}{\lambda}3x^2+\frac{3R_0(R_1-R_0)}{\lambda}2x+\frac{3R_0^2}{\lambda}\cdot 1$$

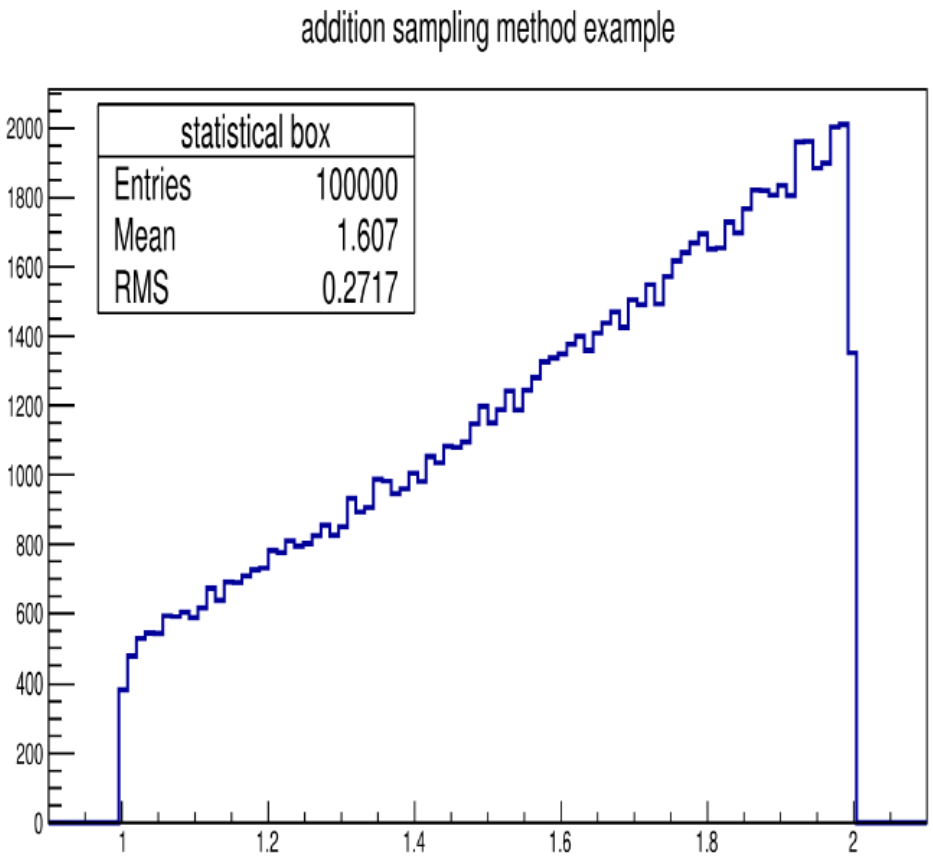
- 其中我们视$P_1=\frac{3R_0^2}{\lambda}$，$P_2=\frac{3R_0(R_1-R_0)}{\lambda}$，$P_3=\frac{(R_1-R_0)^2}{\lambda}$

加抽样

加抽样方法举例流程图

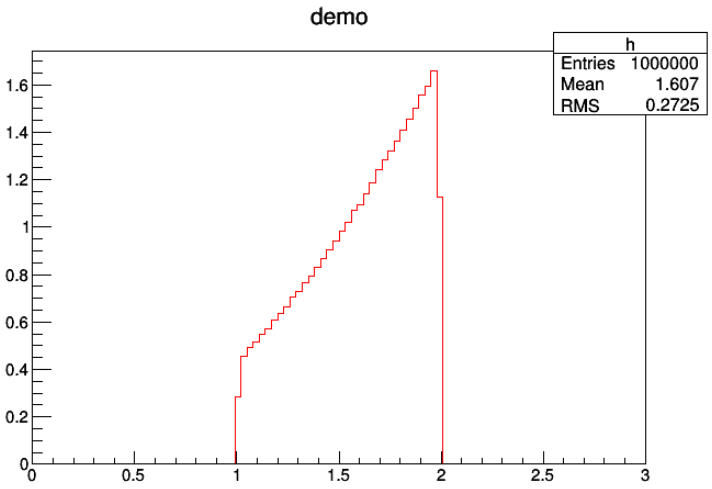


加抽样方法举例结果图



- 这里取球壳内外半径分别为$R_0 = 1$，$R_1 = 2$

加抽样




```
#include "TF1.h"
#include "TMath.h"
void random4()
{TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,500);
   gRandom = new TRandom3(0); ……
   gRandom4 = new TRandom3(3);
   double pi  = 3.14159265358;
   int n=10000;  int n1=0;
   double ld=7.; double x;
   double p1, p2, p3;
   p1=3./ld;  p2=3./ld;  p3=1./ld;
   TH1F *h = new TH1F("h","demo",100,0,3.);
   for (int i = 0; i < n; ++i) {
      double k1=gRandom->Uniform(0,1); ……
      double k4=gRandom4->Uniform(0,1);
        x=k2;
        if(k1>p1 && k1<(p1+p2)) {
           if(k3>x){x=k3;} }
        if(k1>(p1+p2)) {
           if(k3>x){x=k3;}
           if(k4>x){x=k4;} }
        double y=(2.-1.)*x+1.;
        h->Fill(y);   }
      h->SetLineColor(kRed);
      h->Scale(1.0/0.03/double(n));
      h->Draw(); }
```

减抽样

- 减抽样方法是分布密度函数满足如下形式的一种抽样方法：

$$f(x) = A_1 g_1(x) - A_2 g_2(x)$$

- 其中$A_1, A2$为非负实数，$g_1(x), g_2(x)$均为分布密度函数，令m 为$\frac{g_2(x)}{g_1(x)}$的下界，即

$$m = \min_{x \in [a,b]} \frac{g_2(x)}{g_1(x)}$$

- 则

$$f(x) = g_1(x)[A_1 - A_2 \frac{g_2(x)}{g_1(x)}] \leq g_1(x)(A_1 - A_2 m)$$

- 这里我们令：

$$h_1(x) = \frac{f(x)}{(A_1 - A_2 m) g_1(x)}$$

- 那么f(x)可以写为：

$$f(x) = (A_1 - A_2 m) h_1(x) g_1(x)$$

减抽样

- 上式$h_1(x)$可以写为：

$$h_1(x) = \frac{A_1}{A_1 - A_2 m} - \frac{A_2}{A_1 - A_2 m} \frac{g_2(x)}{g_1(x)}$$

- 要求$g_1(x)$易于抽样，同时$h_1(x)$作为抽样舍选条件的函数，这样就可以按照第二类舍选法抽样即可。
- 抽样效率为：$E_1 = \frac{1}{(A_1 - A_2 m)}$
- 类似上述方法，我们可以将f(x)表示为：

$$f(x) = g_2(x)[A_1 \frac{g_1(x)}{g_2(x)} - A_2]$$

那么f(x)可写为：

$$f(x) = \frac{A_1 - A_2 m}{m} h_2(x) g_2(x)$$

- 其中

$$h_2(x) = \frac{A_1 m}{A_1 - A_2 m} \frac{g_1(x)}{g_2(x)} - \frac{A_2 m}{A_1 - A_2 m}$$

减抽样

- 同样这之后按照第二类舍选法来处理，舍选效率为：

$$E_2 = \frac{m}{(A_1 - A_2 m)} = mE_1$$

- 对减抽样法改写为哪种形式，取决于$g_2(x)$是否抽样方便。若$g_2(x)$比$g_1(x)$更易于抽样，则使用后者，反之，则使用前者。
- 当对$g_1(x)$和$g_2(x)$抽样难度相差无几时，就根据$m > 1$ 或$m < 1$来判断哪一种方式抽样的效率高，最后采用效率高的抽样密度函数表示
- 举例：$\mathrm{f(x)} = 2(1 - \mathrm{x}),\ 0 \le x \le 1$这是$\beta$ 分布抽样的一个特例：
- 取$A_1 = 2,\ A_2 = 1,\ g_1(x) = 1,\ g_2(x) = 2x$，此时显然$m = 0$，则按照第一种形式的减抽样方法，有：

$$h_1(x) = \frac{A_1}{A_1 - A_2 m} - \frac{A_2}{A_1 - A_2 m}\frac{g_2(x)}{g_1(x)} = 1 - x$$

- 于是抽样步骤即为：

减抽样

$\mathrm{for(int\ i = 0; i < N_{events}; i++)}$

$\{\mathrm{if}(\xi_1 > 1 - \xi_2)\ \ \mathrm{continue};$

$\xi = \xi_2;\}$

- 由于$1 - \xi_2$可用ξ_2代替，该抽样方法可以简化为：

$\mathrm{for(int\ i = 0; i < N_{events}; i++)}$

$\{\mathrm{if}(\xi_1 > \xi_2)\ \ \mathrm{continue};$

$\xi = \xi_2;\}$

- 对于$\xi_2 > \xi_1$的情况，可取$\xi = \xi_1$，因此：

$$\xi = \min(\xi_1, \xi_2)$$

乘抽样

- 概率密度函数可以写成如下形式的乘分布：

$$f(x) = H(x)g(x)$$

- 其中H(x)是非负函数，$g(x)$为任意分布的密度函数。
- 令L为$H(x)$的上界，乘抽样方法程序如下：
  - (1)在[0,1]区间上抽取均匀分布随机数ξ，由$g(x)$分布密度函数抽样得到η_g
  - (2)判别$\xi \leq \frac{H(\eta_g)}{L}$是否成立。如果不成立，则返回到步骤(1)
  - 选取$\eta = \eta_g$作为服从分布密度函数f(x)的一个抽样值。
- 类似于第二类舍选抽样法的原理，该方法的抽样效率为：

$$\epsilon = \frac{1}{L}$$

乘抽样：麦克斯韦分布抽样

- 麦克斯韦分布密度函数的一般般形式为：

$$f(x) = \frac{2\beta^{\frac{3}{2}}}{\sqrt{\pi}}\sqrt{x}\cdot e^{-\beta\cdot x},\quad x \geq 0$$

- 根据乘抽样方法，这里我们令：

$$g(x) = \frac{2\beta}{3}e^{-\frac{2}{3}\beta\cdot x},\quad x \geq 0$$

- 同时：

$$H(x) = \frac{3\beta^{\frac{1}{2}}}{\sqrt{\pi}}\sqrt{x}\cdot e^{-\frac{1}{3}\beta\cdot x},\quad x \geq 0$$

- 对$H(x)$取极值可以得到：

$$L = \sqrt{\frac{27}{2\pi\cdot e}}$$

- 对于g(x)使用直接抽样法的知识可以得到抽样为：

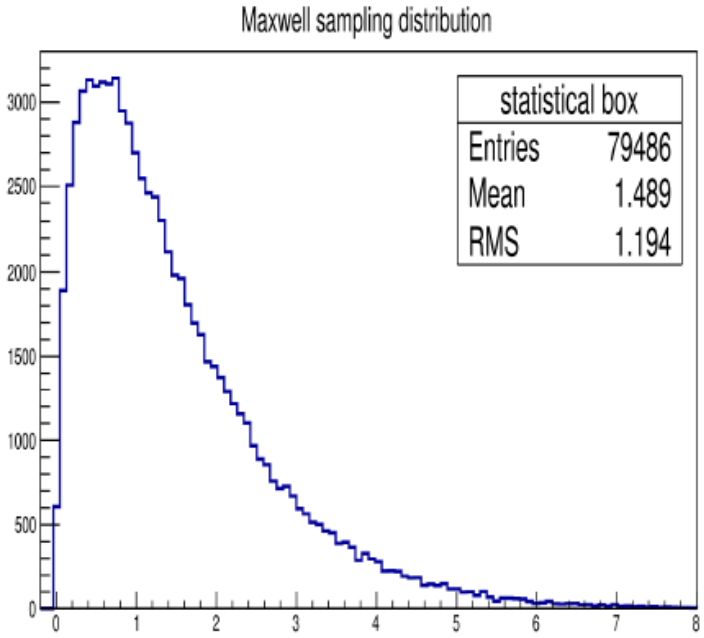
$$\eta_g = -\frac{3}{2\beta}\ln\xi_2$$

乘抽样：麦克斯韦分布抽样

- 则麦克斯韦分布的抽样方法为：

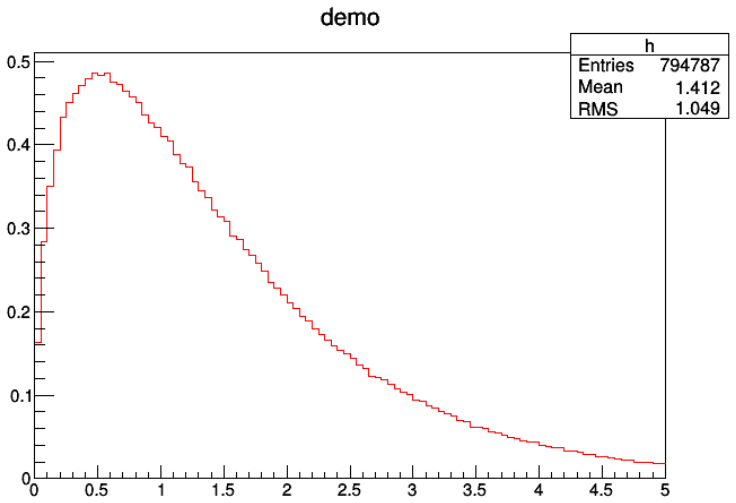
$$\text{for(int i} = 0; \text{i} < \text{N}_{\text{events}}; \text{i}++)$$
$$\{\text{if}(\xi_1^2 > -\text{e}\xi_2) \cdot \ln\xi_2; \text{p continue};$$
$$\xi = -\frac{3}{2\beta}\ln\xi_2;\}$$

- 该分布的抽样效率为：$\epsilon = \sqrt{\frac{2\pi e}{27}} \approx 0.795$



乘抽样：麦克斯韦分布抽样

```
//Maxwell Distri.
// f(x)=2*sqrt(x)/sqrt(pi)*exp(-x), 0<x<inf
// f(x)=H(x)*g(x)
//    g(x)=2/3*exp(-2x/3)
//    H(x)=3/sqrt(pi)*sqrt(x)*exp(-x/3)
//     L=max(H(x))=sqrt(27/2/pi/e)
```




```
#include "TF1.h"
#include "TMath.h"
void random5()
{
    TCanvas *c1= new TCanvas("c1", "demo", 10,10,700,500);
      gRandom = new TRandom3(0);
      gRandom2 = new TRandom3(1);
      double pi  = 3.14159265358;
      int n=1000000; int n1=0;
      double u,v;
      double L=sqrt(27./2./pi/exp(1.0));
      TH1F *h = new TH1F("h","demo",100,0,5.);
       for (int i = 0; i < n; ++i) {
         double k1=gRandom->Uniform(0,1);
         double k2=gRandom2->Uniform(0,1);
         double x=-3./2.*TMath::Log(k1);
         if(k2*k2<=(-exp(1.0)*k1*TMath::Log(k1))) {h-
>Fill(x); n1++;}
      }
      h->SetLineColor(kRed);
      h->Scale(1.0/0.05/double(n1));
      h->Draw();
     cout<<double(n1)/double(n)<<" "<<1.0/L<<"
    "<<2./sqrt(pi)/exp(1.0)<<" "<<2./sqrt(pi)*sqrt(2.)*exp(-2.0)<<endl;
 }
```

乘加抽样

- 在实际问题中，经常会遇到如下形式的分布：

$$f(x) = \sum_n H_n(x) g_n(x), \quad x \in [a, b]$$

- 其中$H_n(x)$为非负函数，$g_n(x)$为任意的分布密度函数，不失一般性，下面我们只考虑两项(n = 2)的情况，对更多项($n > 2$)情况的一般表示可以以此作推广。
- 设η的分布密度函数为：

$$f(x) = H_1(x) g_1(x) + H_2(x) g_2(x)$$

- 如果令：

$$P_1 = \int_a^b H_1(x) g_1(x), \quad P_2 = \int_a^b H_2(x) g_2(x)$$

- 则必有$P_1 + P_2 = 1$。这样我们可以改写f(x)为：

$$f(x) = P_1 \frac{H_1(x)}{P_1} g_1(x) + P_2 \frac{H_2(x)}{P_2} g_2(x) = P_1 g_1'(x) + P_2 g_2'(x)$$

乘加抽样

- 其中：

$$g_1'(x) = \frac{H_1(x)}{P_1} g_1(x) \quad g_2'(x) = \frac{H_2(x)}{P_2} g_2(x)$$

- 可以看到上式所表示的分布密度函数形式就可以采用加分布抽样法。
- 我们也可以采用另一种方式，将公式改写为：

$$f(x) = (M_1 + M_2)\{\frac{M_1}{M_1 + M_2} \frac{H_1(x)}{M_1} g_1(x) + \frac{M_2}{M_1 + M_2} \frac{H_2(x)}{M_2} g_2(x)\}$$

- 其中M_1和M_2分别是$H_1(x)$和H_2在区域$[a, b]$ 上的上界。令：

$$P_1 = \frac{M_1}{M_1 + M_2}, \quad P_2 = \frac{M_2}{M_1 + M_2}$$

$$L_1 = L_2 = M_1 + M_2, \quad H_1(x) = M_1 h_1(x), \quad H_2(x) = M_2 h_2(x)$$

- 则：

$$f(x) = P_1[L_1 h_1(x) g_1(x)] + P_2[L_2 h_2(x) g_2(x)]$$

乘加抽样

- 这样的分布密度函数形式就可以采用加分布抽样和第二类舍选法抽样。这种处理方法的效率不如前一种方法高，但省掉了公式中的积分计算。
- 例子： 光子散射后能量分布的抽样
- 令光子散射前后的能量分别为α和α'(以m_0c^2为单位，m_0 为电子静止质量，c为光速)，$x = \frac{\alpha}{\alpha'}$，则x的分布函数为：

$$f(\frac{x}{\alpha}) = \frac{1}{K(a)}[(\frac{\alpha+1-x}{a\cdot x})^2 + \frac{1}{x} - \frac{1}{x^2} + \frac{1}{x^3}],\ 1 \le x \le 1+2\alpha$$

- 该分布即为光子散射的能量分布，其中$K(\alpha)$为归一因子：
- 把光子的散射能量分布改写成如下的形式：

$$f(\frac{x}{\alpha}) = \frac{1}{K(a)}\{[(\frac{\alpha+1-x}{\alpha})^2 + 1]\frac{1}{x^2} + \frac{(x-1)^2}{x^3}\}$$

- 在$[1, 1+2\alpha]$上定义如下函数：

$$g_1(\frac{x}{\alpha}) = \frac{1+2\alpha}{2\alpha}\cdot\frac{1}{x^2} \qquad g_2(\frac{x}{\alpha}) = \frac{1}{2\alpha}$$

乘加抽样

$$H_1(\frac{x}{\alpha}) = \frac{2\alpha}{K(\alpha)(1+2\alpha)}[(\frac{\alpha+1-x}{\alpha})^2+1],$$

$$H_2(\frac{x}{\alpha}) = \frac{2\alpha}{K(\alpha)}\frac{(x-1)^2}{x^3}$$

- 则有：

$$f(\frac{x}{\alpha}) = H_1(\frac{x}{\alpha})\cdot g_1(\frac{x}{\alpha}) + H_2(\frac{x}{\alpha})\cdot g_2(\frac{x}{\alpha})$$

- 使用乘加抽样法，可以算得$H_1(\frac{x}{\alpha})$，$H_2(\frac{x}{\alpha})$的最大值分别为：

$$M_1 = \frac{4\alpha}{K(\alpha)(1+2\alpha)} \quad M_2 = \frac{8\alpha}{27\cdot K(\alpha)}$$

- 由直接抽样法，算得由$g_1(\frac{x}{\alpha})$，$g_2(\frac{x}{\alpha})$的抽样值分别为：

$$\eta_{g_1} = \frac{1+2\alpha}{1+2\alpha\xi_2} \quad \eta_{g_2} = 1+2\alpha\xi_2$$

乘加抽样

- 光子散射能量分布的抽样方法为：

(1)产生$[0,1]$区间上的独立的均匀分布的随机数ξ_1，ξ_2和ξ_3；

(2)判断$\xi_1 \leq \frac{27}{4\alpha+29}$是否满足。如满足上面不等式，则执行(3);如不满足，则执行(5)；

(3)由直接抽样法从$g_1(\frac{x}{\alpha})$中抽取随机数,即抽取$\eta_{g_1} = \frac{1+2\alpha}{1+2\alpha\xi_2}$,判断$\xi_3 \leq \frac{H_1(\frac{x}{\alpha})}{M_1}$是否成立，即判断$\xi_3 \leq \frac{1}{2}[(\frac{\alpha+1-\eta_{g_1}}{\alpha})^2+1]$，是否成立。若成立，则执行步骤(4),若不成立，则返回步骤(1)

(4)取$\eta = \eta_{g_1}$作为最终抽样值；

(5)由直接抽样法从$g_2(\frac{x}{\alpha})$中抽取随机数,即抽取$\eta_{g_2} = 1+2\alpha\xi_2$，判断$\xi_3 \leq \frac{H_2(\frac{x}{\alpha})}{M_2}$是否成立，即判断$\xi_3 \leq \frac{27}{4}\frac{(\eta_{g_2}-1)^2}{\eta_{g_2}^3}$是否成立。若成立，则执行步骤(6),若不成立，则返回步骤(1)

(6)取$\eta = \eta_{g_2}$作为最终抽p样值；

请自己检验

作业：

1. P20: 随机数的统计检验

产生一组[0-1]之间均匀分布的随机数， 统计[0-0.1-0.2…0.9-1.0]各个区间的撒点数。进行Chi2检验。