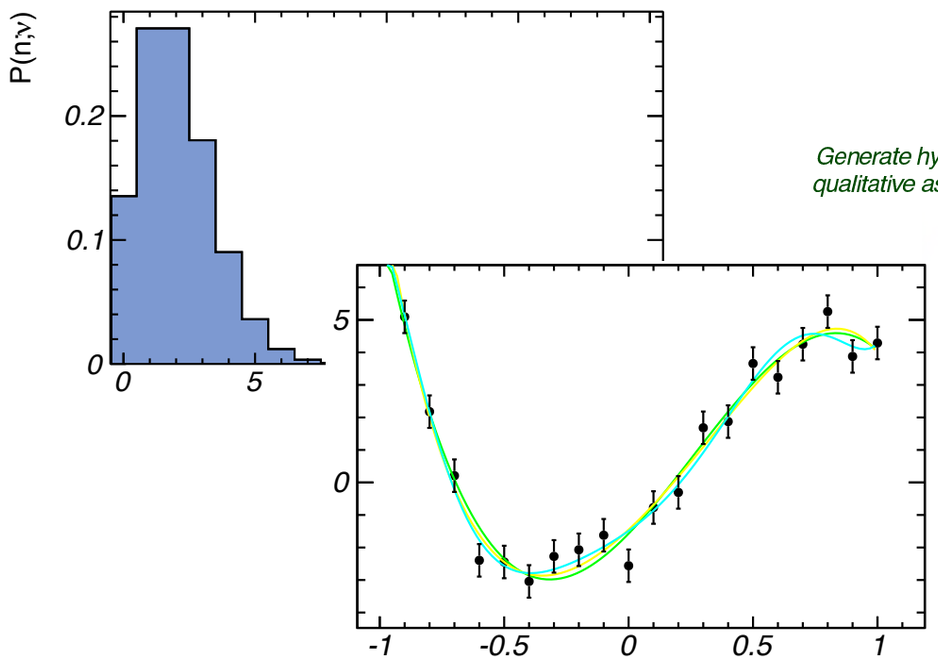
Karlsruher Institut für Technologie

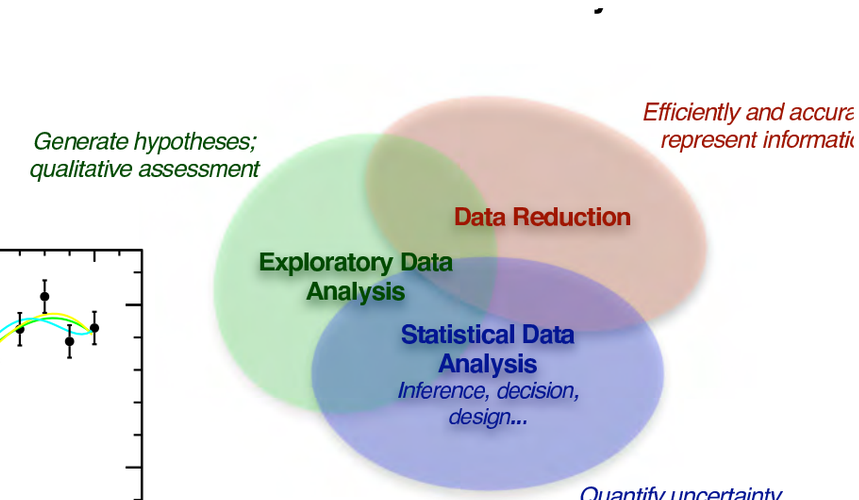
Rechnernutzung in der Physik

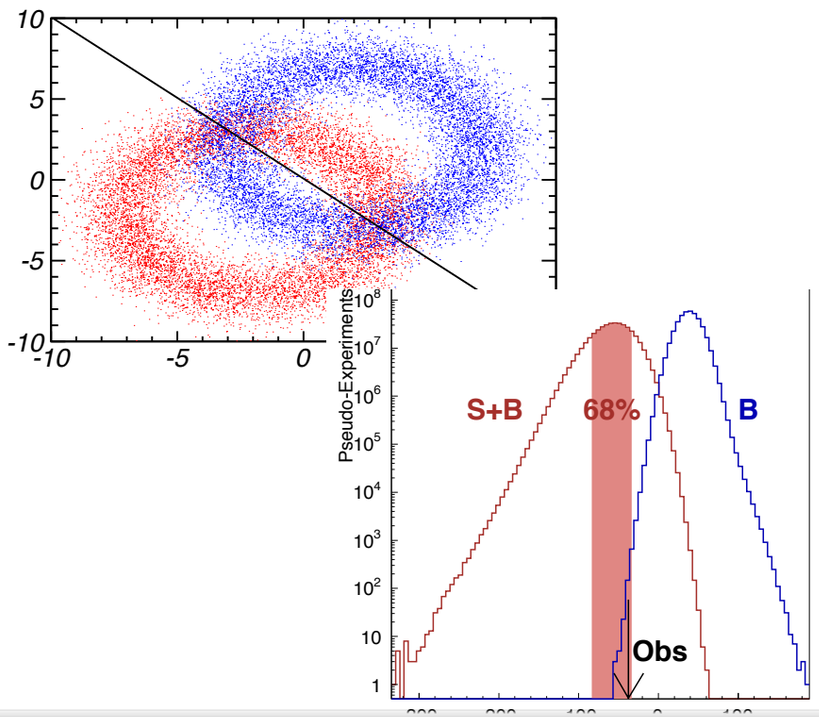
Teil 3 – Statistische Methoden der Datenanalyse

Karlsruher Institut für Technologie
Wintersemester 2012/2013

Ulrich Husemann
Institut für Experimentelle Kernphysik, Karlsruher Institut für Technologie







QISPOS-Anmeldung

- Rechnernutzung jetzt bei QISPOS freigeschaltet
  → bitte anmelden
  - Name: Rechnernutzung
  - Prüfungsnummer: 172
  - Anmeldebeginn: 15.10.12
  - Anmeldeende: 07.02.13
  - Rücktrittsende: 07.02.13
  - Prüfungsdatum: 08.02.13

Kurze Wiederholung

- Kontinuierliche Zufallsvariable:
  - Wahrscheinlichkeitsdichtefunktion (PDF): $f(x_0)\,dx$ = Wahrscheinlichkeit, x in infinitesimalem Intervall $[x_0, x_0+dx]$ zu finden
  - Kumulative Verteilungsfunktion (CDF)
  - In zwei Dimensionen: Randverteilung und bedingte Verteilung als Projektionen
- Algebraische Momente von Verteilungen
  - Erwartungswert $E[x]$
  - Varianz $V[x] = \sigma_x^2 = E[x^2]–E[x]^2$ (σ_x Standardabweichung)
  - Kovarianz: $\sigma_{xy} = E[(x–E[x])(y–E[y])]$
  - Kovarianzmatrix: symmetrische Matrix der Varianzen und Kovarianzen
- Funktionen von Zufallsvariablen: Gauß'sche Fehlerfortpflanzung

Karlsruher Institut für Technologie

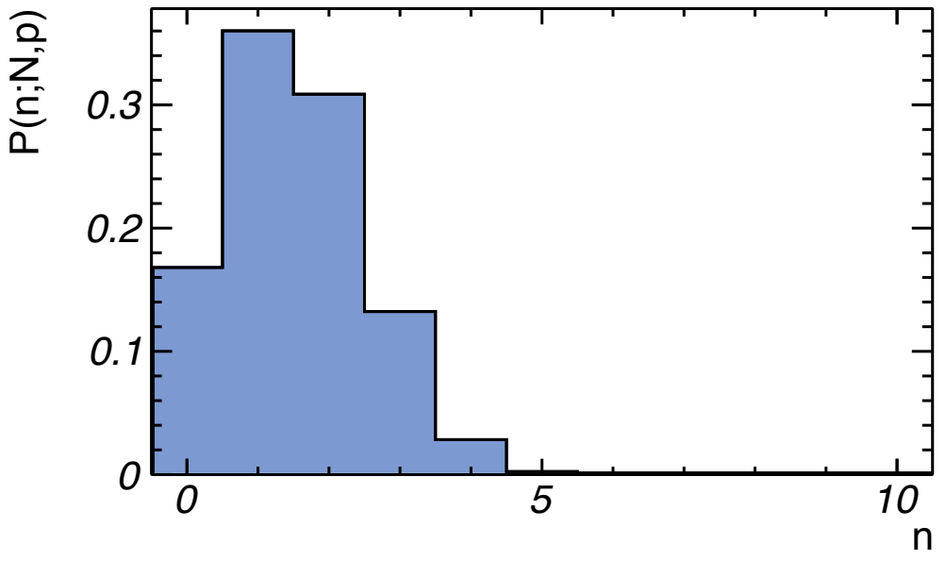
Kapitel 3.3

Beispiele für Wahrscheinlichkeits-verteilungen

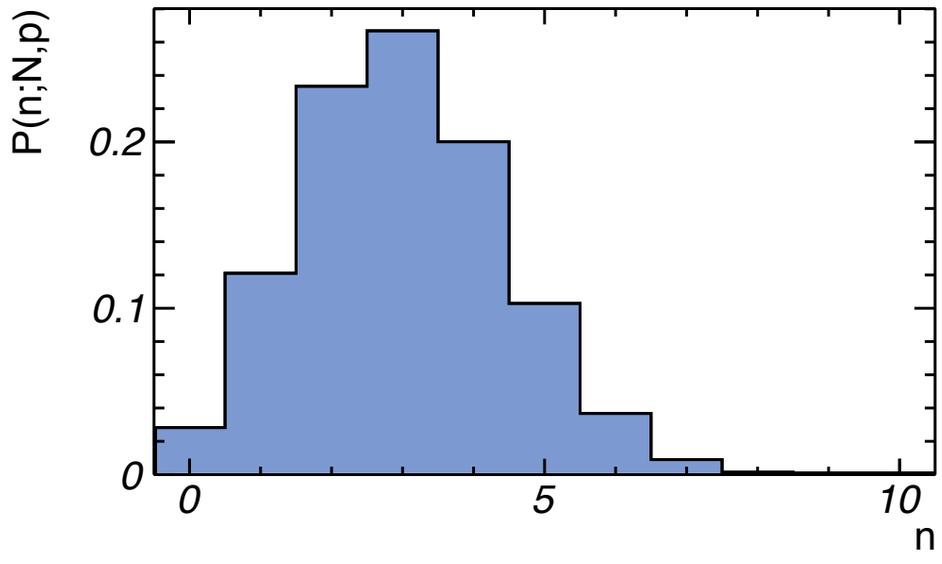
Binomialverteilung

$$P(n;N,p) = \binom{N}{n} p^n (1-p)^{N-n}$$

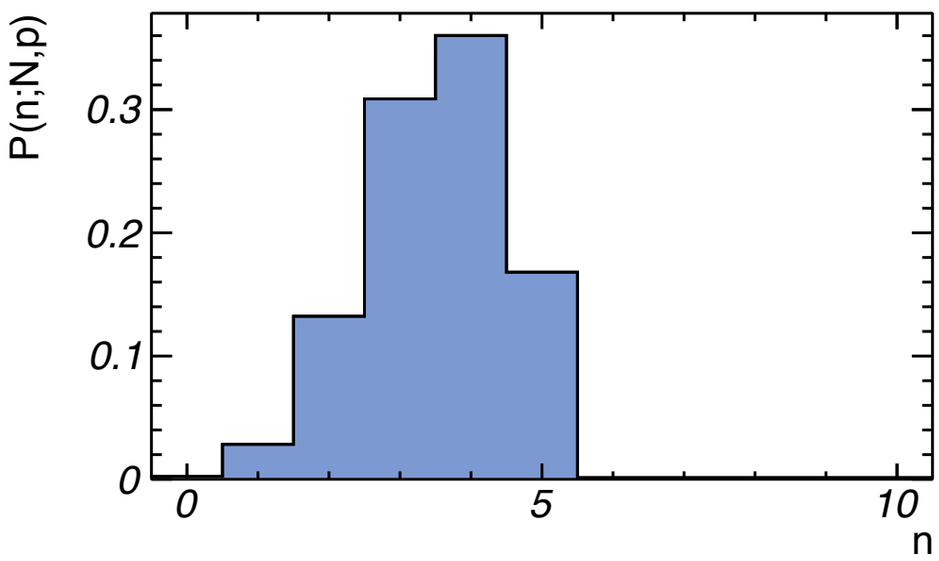
Binomial Probability: N = 5, p = 0.3



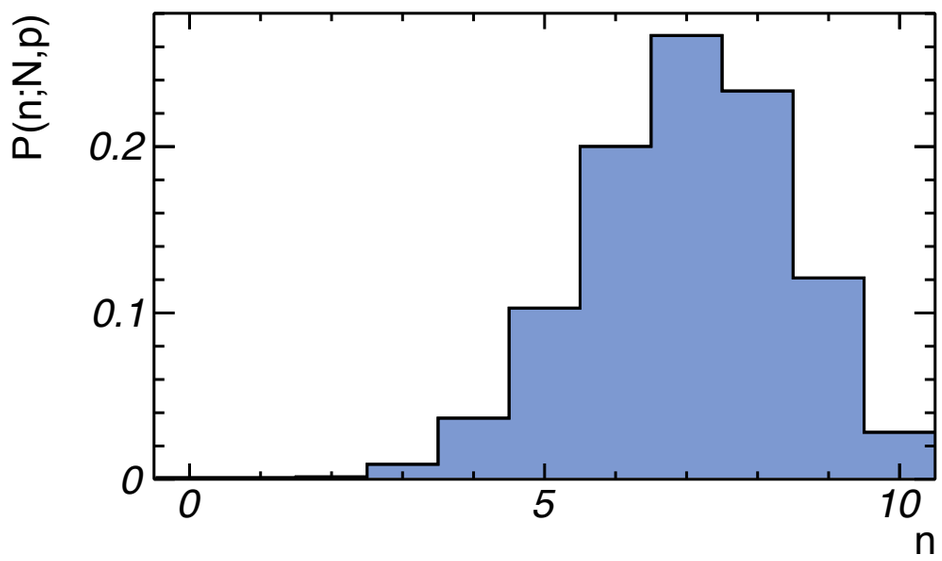
Binomial Probability: N = 10, p = 0.3



Binomial Probability: N = 5, p = 0.7

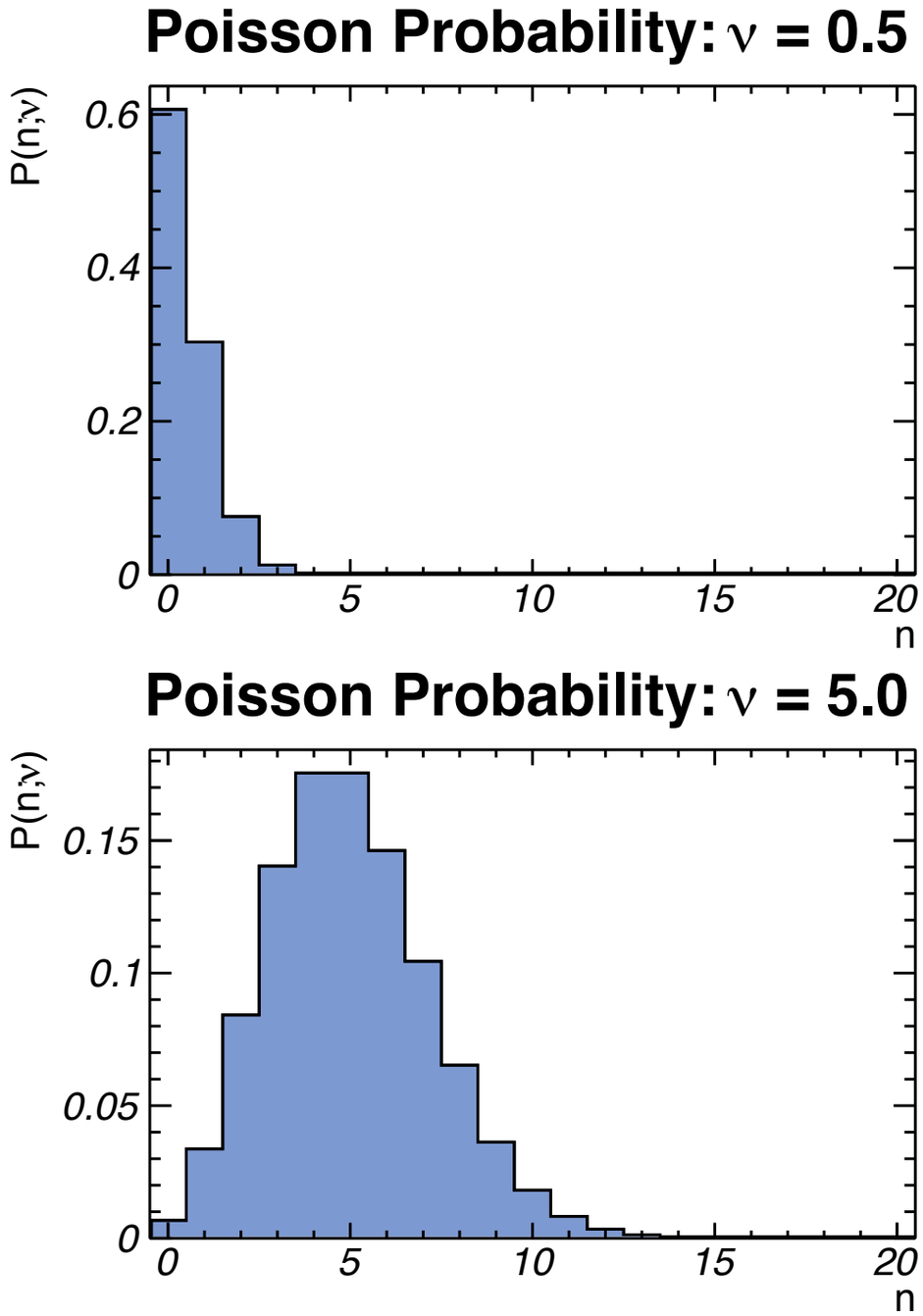


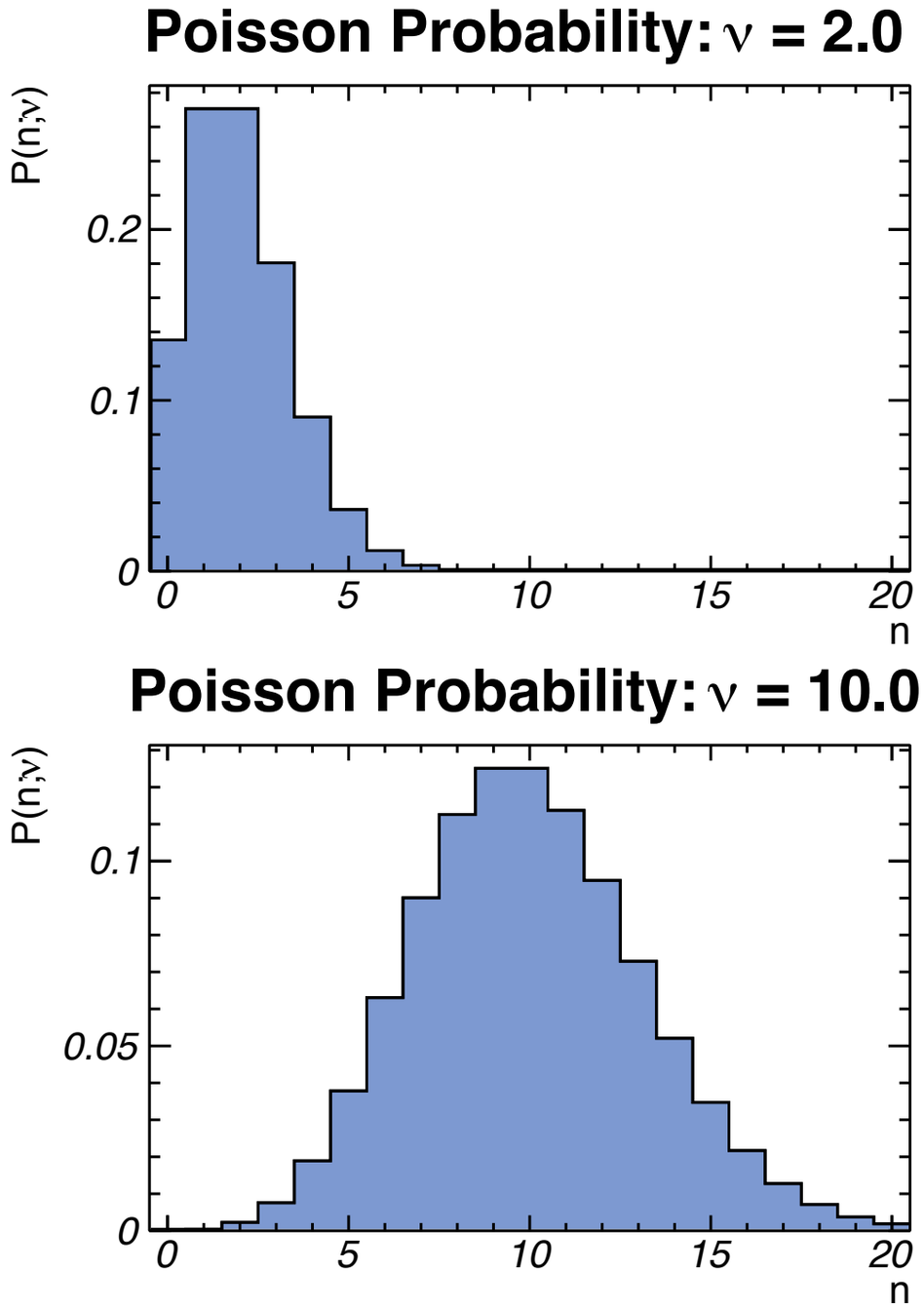
Binomial Probability: N = 10, p = 0.7



Poisson-Verteilung

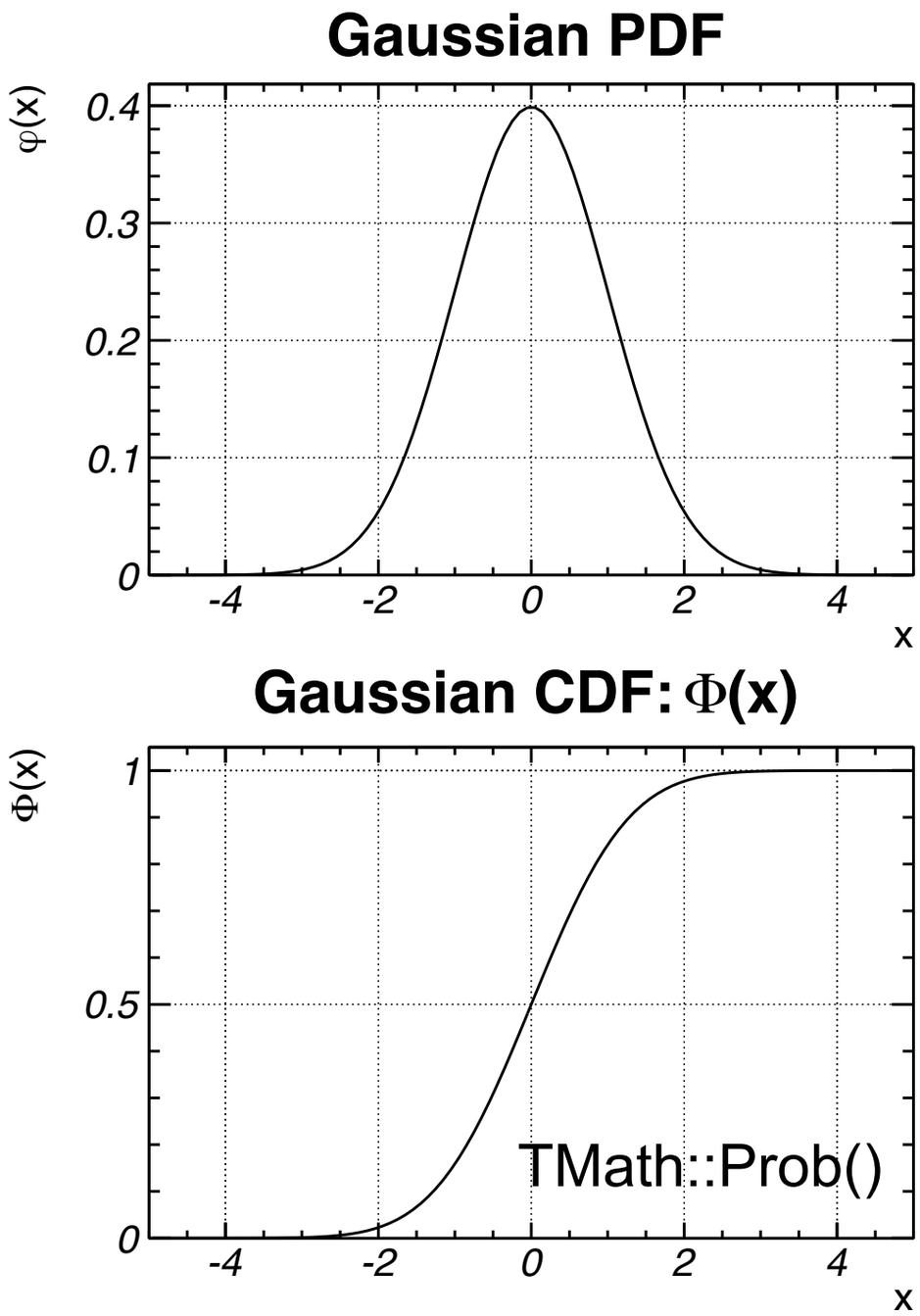
$$P(n;\nu) = \frac{\nu^n}{n!} e^{-\nu}$$

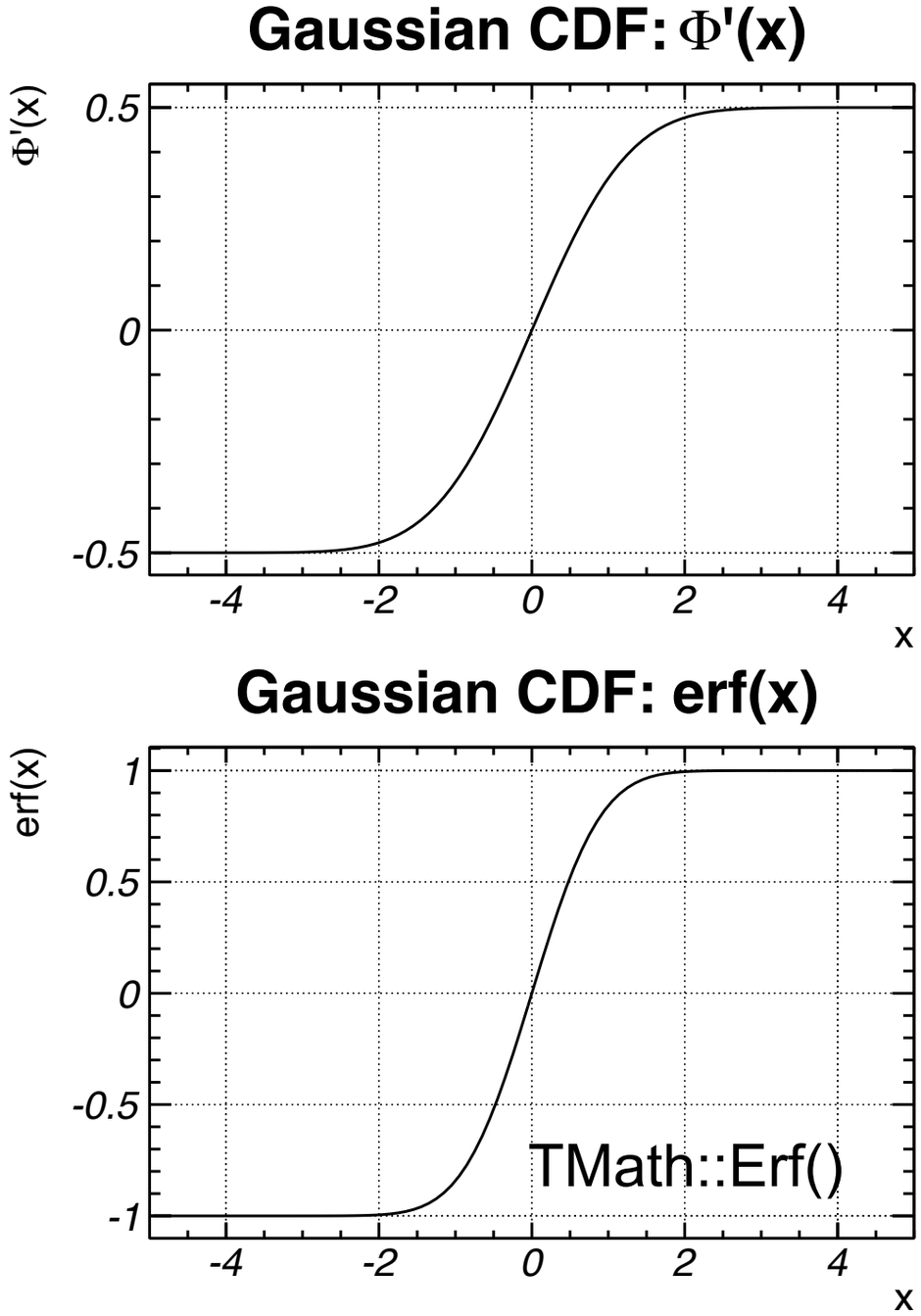




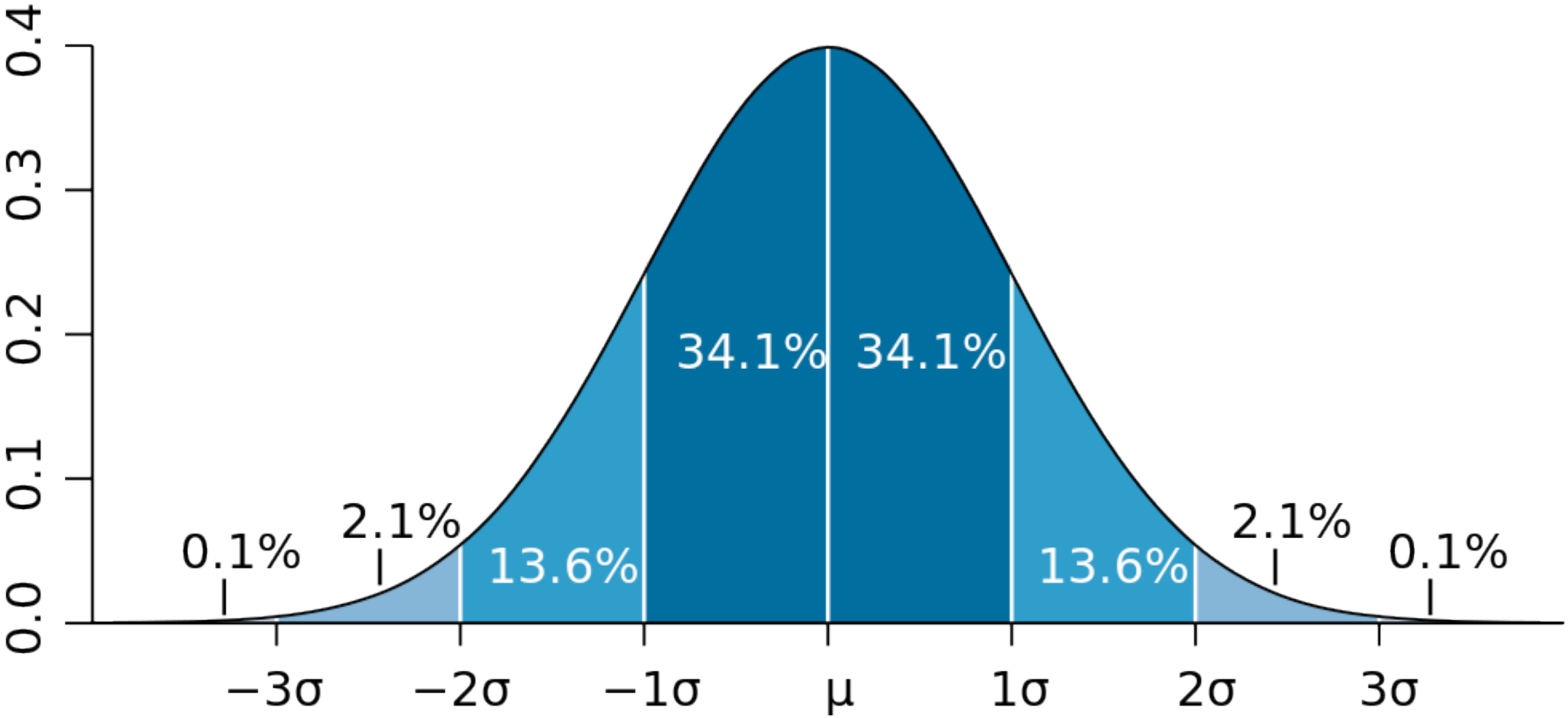
Normalverteilung

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





Gauß’sche W.keitsintervalle



[de.wikipedia.org]

→ typische Vertrauensintervalle z. B. für „Fehlerbalken” (später)

Gauß-Verteilung als Grenzfall

Binomialverteilung: P(n;N,p)

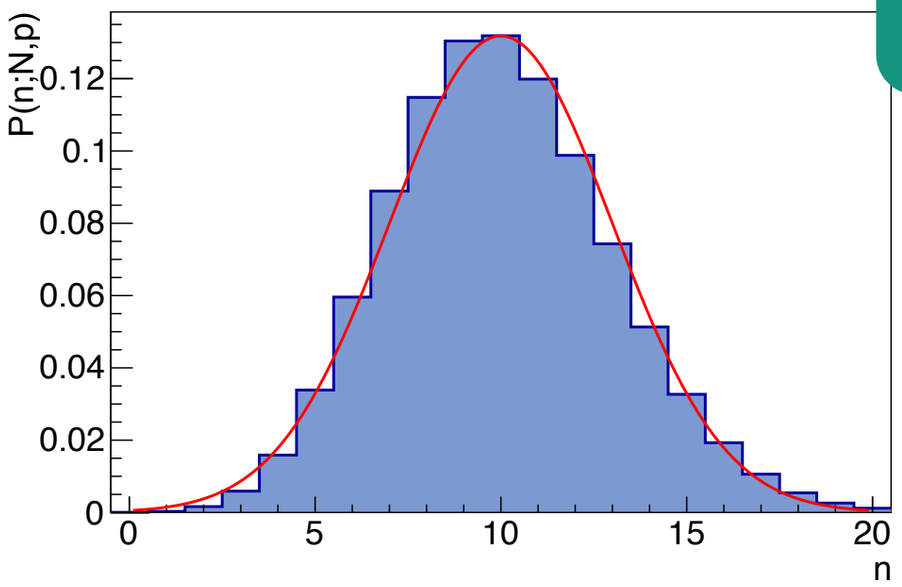
$N \to \infty,\ p \to 0$, $Np = \nu$

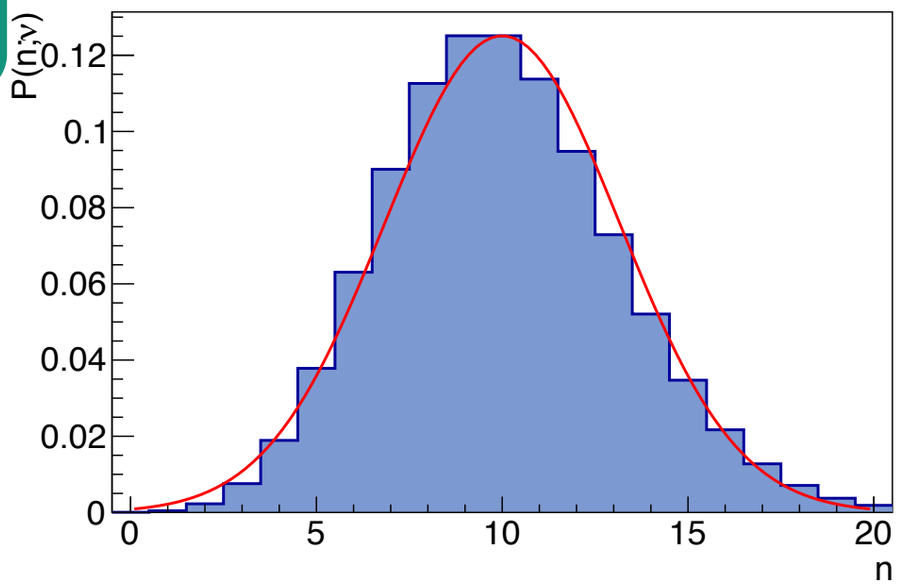
Poisson-Verteilung: P(n;ν)

$N \to \infty,\ p \to 0$
$\mu = Np,\ \sigma^2 = Np(1-p)$

ν „groß”
$\mu = \nu,\ \sigma^2 = \nu$

Gaußverteilung: G(x;μ.σ)

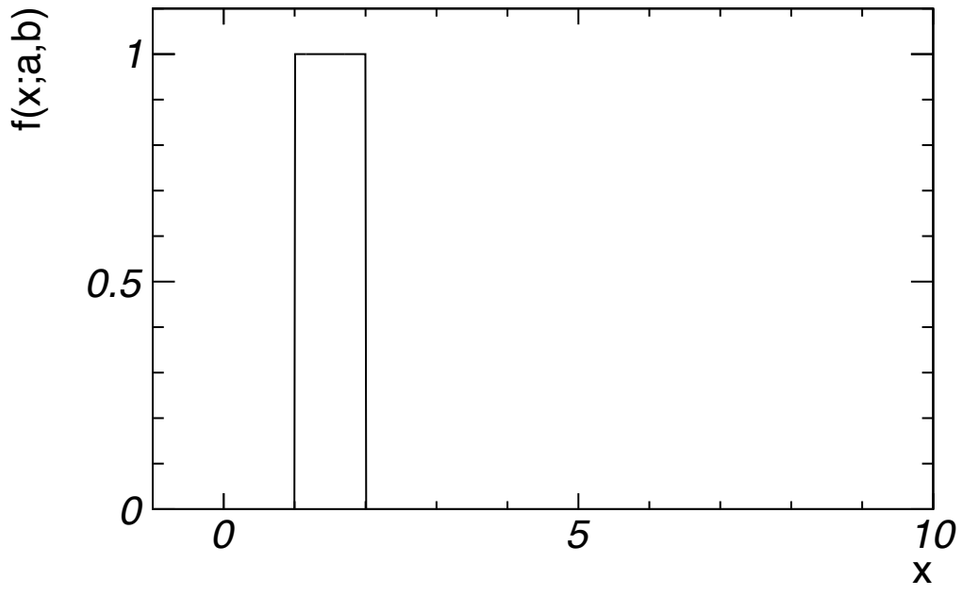




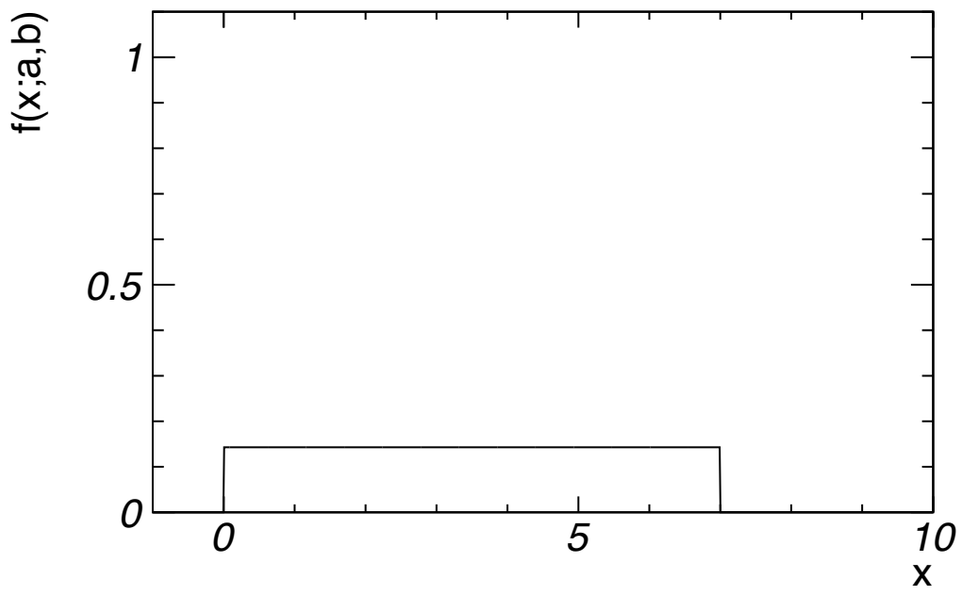
Gleich- und Exponentialverteilung

$$f(x; a, b) = \frac{1}{b-a} \text{ für } x \in [a, b]$$

Uniform PDF: x ∈[1,2]

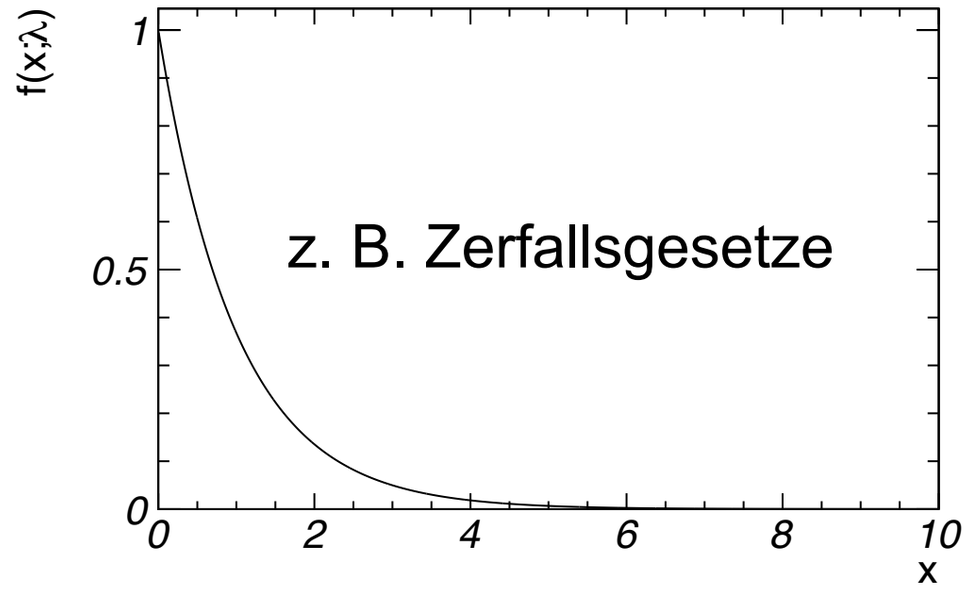


Uniform PDF: x ∈[0,7]

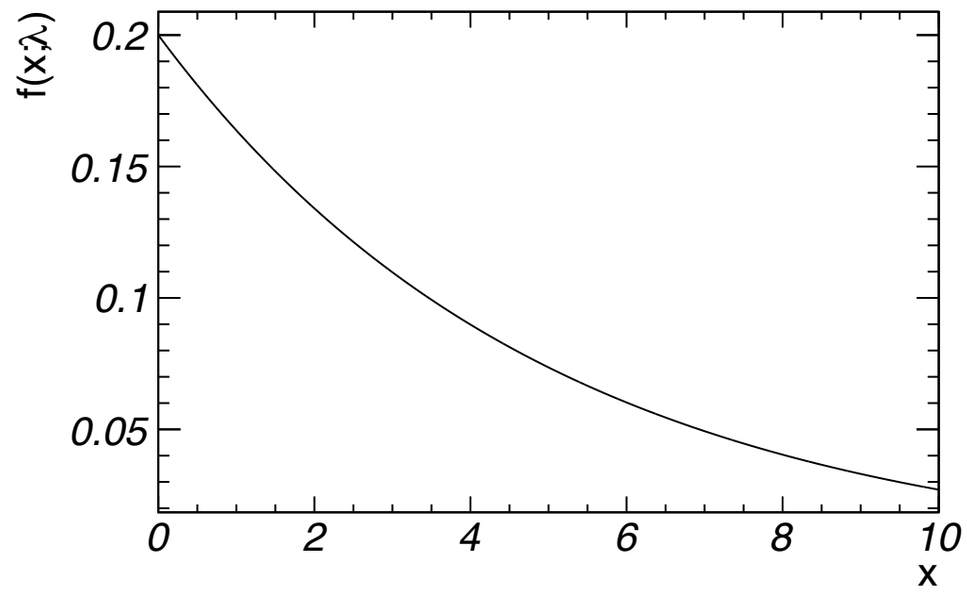


$$f(x; \lambda) = \lambda \exp[-\lambda x]$$

Exponential PDF: λ = 1.0



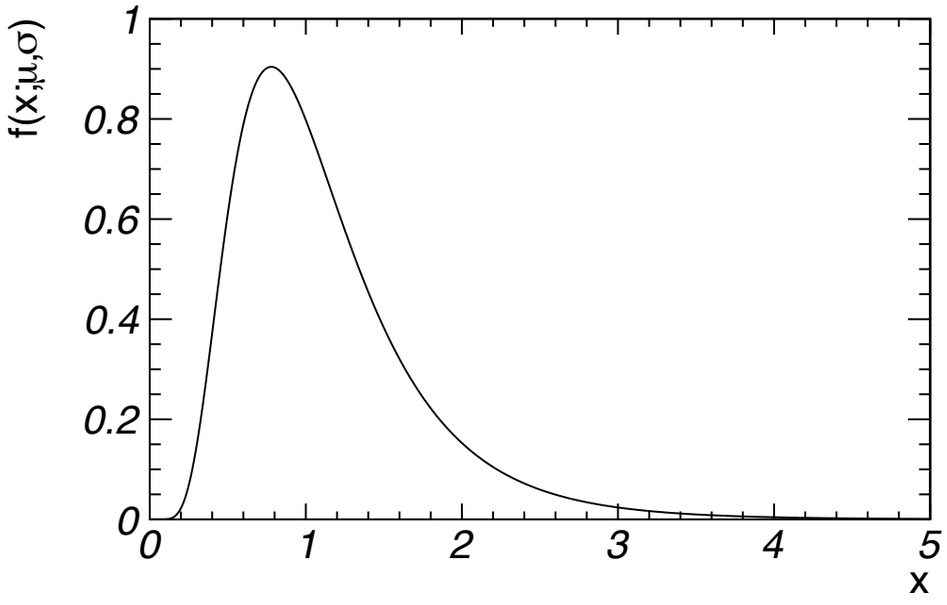
Exponential PDF: λ = 0.2



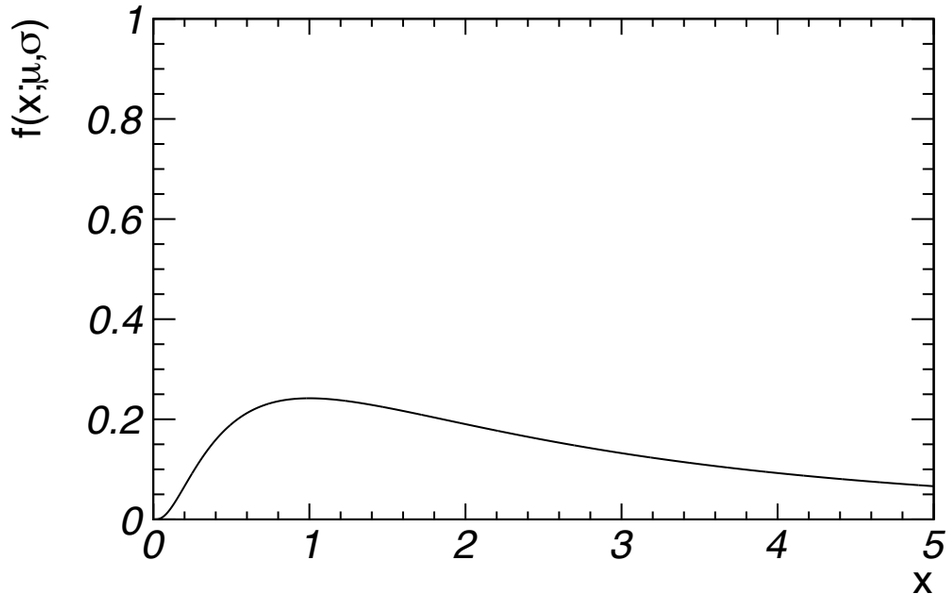
Log-Normalverteilung

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

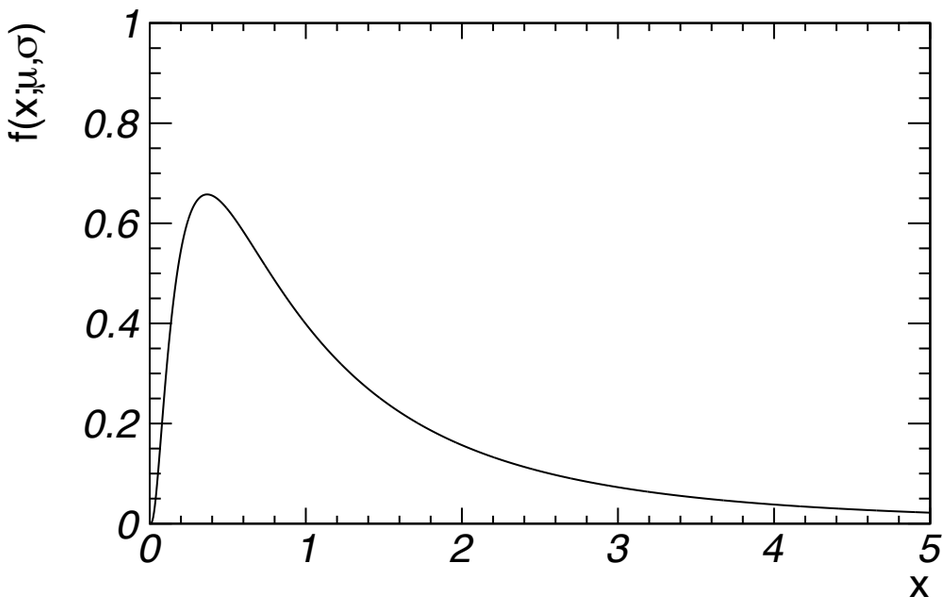
Log-Normal Probability: μ = 0.0, σ = 0.5



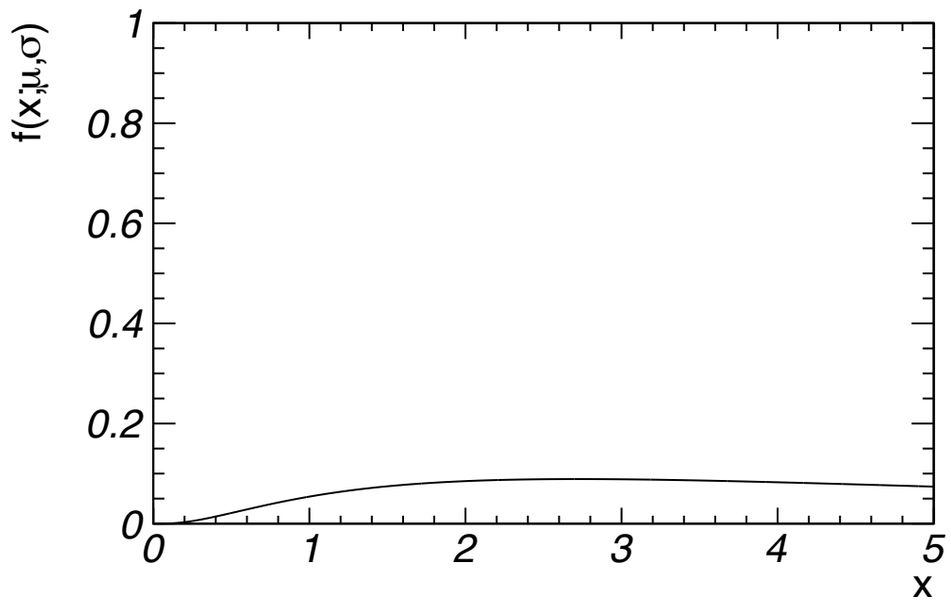
Log-Normal Probability: μ = 1.0, σ = 1.0



Log-Normal Probability: μ = 0.0, σ = 1.0



Log-Normal Probability: μ = 2.0, σ = 1.0



Log-Normalverteilung

Table 2. Comparing log-normal distributions across the sciences in terms of the original data. $\tilde{x}*$ is an estimator of the median of the distribution, usually the geometric mean of the observed data, and s estimates the multiplicative standard deviation, the shape parameter of the distribution; 68% of the data are within the range of $\tilde{x}* \times/ s^*$, and 95% within $\tilde{x}* \times/ (s^*)^2$. In general, values of s* and some of $\tilde{x}*$ were obtained by transformation from the parameters given in the literature (cf. Table 3). The goodness of fit was tested either by the original authors or by us.*

Discipline and type of measurement	Example	*n*	$\tilde{x}*$	s*	Reference
Geology and mining					
Concentration of elements	Ga in diabase	56	17 mg · kg^{-1}	1.17	Ahrens 1954
	Co in diabase	57	35 mg · kg^{-1}	1.48	Ahrens 1954
	Cu	688	0.37%	2.67	Razumovsky 1940
	Cr in diabase	53	93 mg · kg^{-1}	5.60	Ahrens 1954
	^{226}Ra	52	25.4 Bq · kg^{-1}	1.70	Malanca et al. 1996
	Au: small sections	100	(20 inch-dwt.)[a]	1.39	Krige 1966
	large sections	75,000	n.a.	2.42	Krige 1966
	U: small sections	100	(2.5 inch-lb.)[a]	1.35	Krige 1966
	large sections	75,000	n.a.	2.35	Krige 1966
Human medicine					
Latency periods of diseases	Chicken pox	127	14 days	1.14	Sartwell 1950
	Serum hepatitis	1005	100 days	1.24	Sartwell 1950
	Bacterial food poisoning	144	2.3 hours	1.48	Sartwell 1950
	Salmonellosis	227	2.4 days	1.47	Sartwell 1950
	Poliomyelitis, 8 studies	258	12.6 days	1.50	Sartwell 1952
	Amoebic dysentery	215	21.4 days	2.11	Sartwell 1950
Survival times after cancer diagnosis	Mouth and throat cancer	338	9.6 months	2.50	Boag 1949
	Leukemia myelocytic (female)	128	15.9 months	2.80	Feinleib and McMahon 1960
	Leukemia lymphocytic (female)	125	17.2 months	3.21	Feinleib and McMahon 1960
	Cervix uteri	939	14.5 months	3.02	Boag 1949
Age of onset of a disease	Alzheimer	90	60 years	1.16	Horner 1987
Environment					
Rainfall	Seeded	26	211,600 m^3	4.90	Biondini 1976
	Unseeded	25	78,470 m^3	4.29	Biondini 1976
HMF in honey	Content of hydroxymethylfurfurol	1573	5.56 g kg^{-1}	2.77	Renner 1970
Air pollution (PSI)	Los Angeles, CA	364	109.9 PSI	1.50	Ott 1978
	Houston, TX	363	49.1 PSI	1.85	Ott 1978
	Seattle, WA	357	39.6 PSI	1.58	Ott 1978
Aerobiology					
Airborne contamination by bacteria and fungi	Bacteria in Marseilles	n.a.	630 cfu m^{-3}	1.96	Di Giorgio et al. 1996
	Fungi in Marseilles	n.a.	65 cfu m^{-3}	2.30	Di Giorgio et al. 1996
	Bacteria on Porquerolles Island	n.a.	22 cfu m^{-3}	3.17	Di Giorgio et al. 1996
	Fungi on Porquerolles Island	n.a.	30 cfu m^{-3}	2.57	Di Giorgio et al. 1996
Phytomedicine					
Fungicide sensitivity, EC_{50}	Untreated area	100	0.0078 µg · ml^{-1} a.i.	1.85	Romero and Sutton 1997
Banana leaf spot	Treated area	100	0.063 µg · ml^{-1} a.i.	2.42	Romero and Sutton 1997
	After additional treatment	94	0.27 µg · ml^{-1} a.i.	>3.58	Romero and Sutton 1997
Powdery mildew on barley	Spain (untreated area)	20	0.0153 µg · ml^{-1} a.i.	1.29	Limpert and Koller 1990
	England (treated area)	21	6.85 µg · ml^{-1} a.i.	1.68	Limpert and Koller 1990
Plant physiology					
Permeability and solute mobility (rate of constant desorption)	*Citrus aurantium*/H_2O/Leaf	73	1.58 10^{-10} m s^{-1}	1.18	Baur 1997
	Capsicum annuum/H_2O/CM	149	26.9 10^{-10} m s^{-1}	1.30	Baur 1997
	Citrus aurantium/2,4–D/CM	750	7.41 10^{-7} 1 s^{-1}	1.40	Baur 1997
	Citrus aurantium/WL110547/CM	46	2.6310^{-7} 1 s^{-1}	1.64	Baur 1997
	Citrus aurantium/2,4–D/CM	16	n.a.	1.38	Baur 1997
	Citrus aurantium/2,4–D/CM + acc.1	16	n.a.	1.17	Baur 1997
	Citrus aurantium/2,4–D/CM	19	n.a.	1.56	Baur 1997

[E. Limpert, W.A. Stahel, M. Abbt, BioScience 51 (2001) 341]

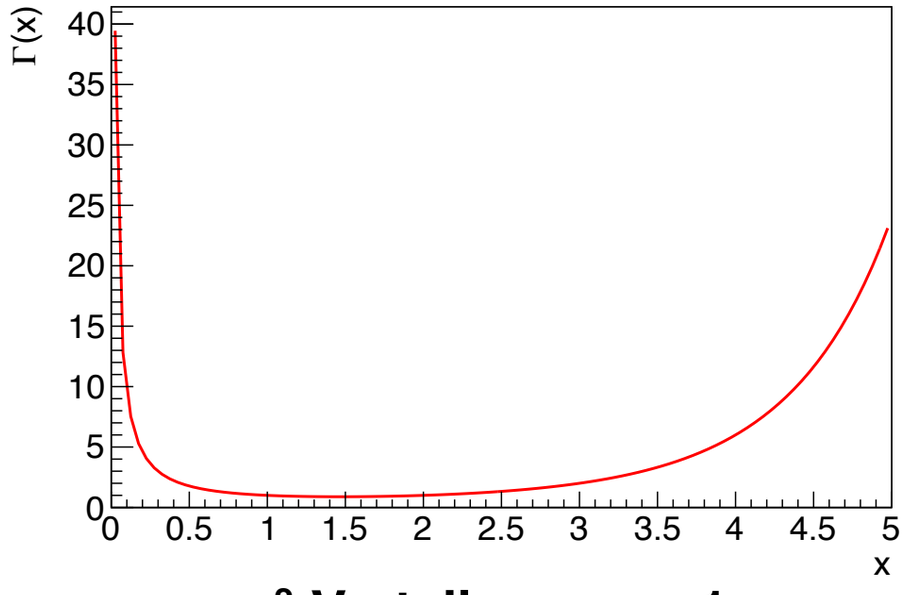
- Weitere Beispiele:
  - Verteilung von Einkommen in Bevölkerung
  - Größe von Städten
  - Black-Scholes-Modell: ln(Aktienkurs) folgt Random Walk
- Relevanz der Log-Normalverteilung durch zentralen Grenzwertsatz: für Produkte von Zufallszahlen folgt ln(x) Gaußverteilung

χ^2-Verteilung

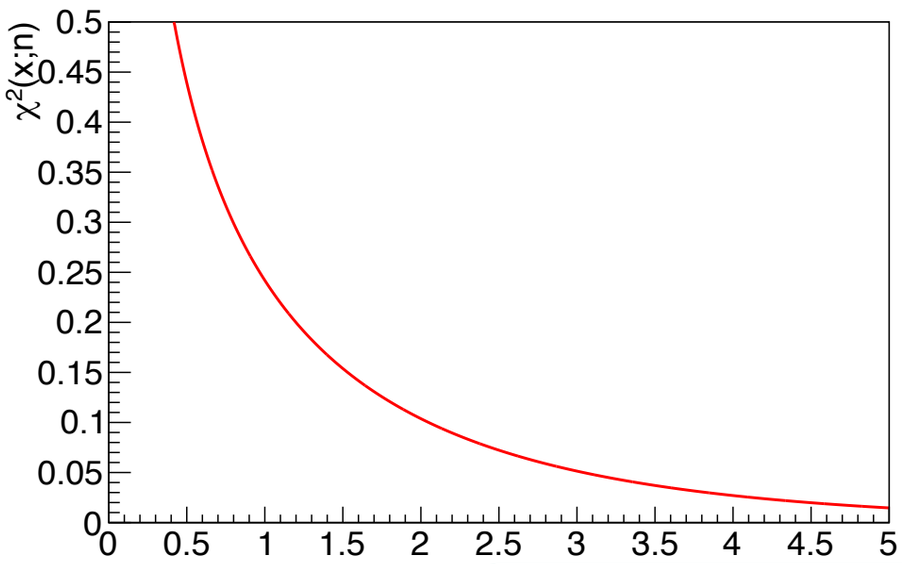
wichtig für statistische Tests

$$\chi^2(x;n) = \frac{x^{\frac{n}{2}-1}}{2^{\frac{n}{2}}\Gamma(\frac{n}{2})} \exp\left[-\frac{x}{2}\right]$$

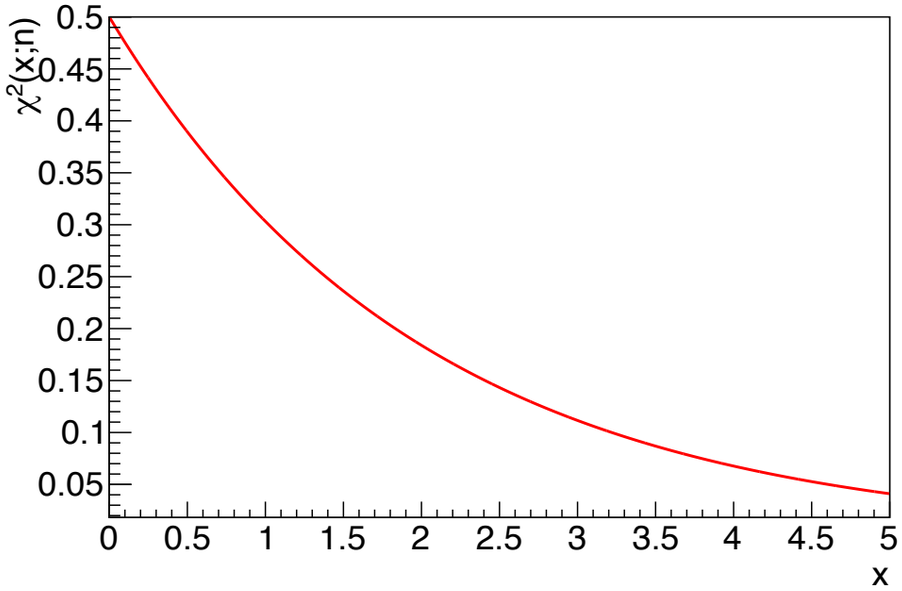
TMath::Gamma(x)



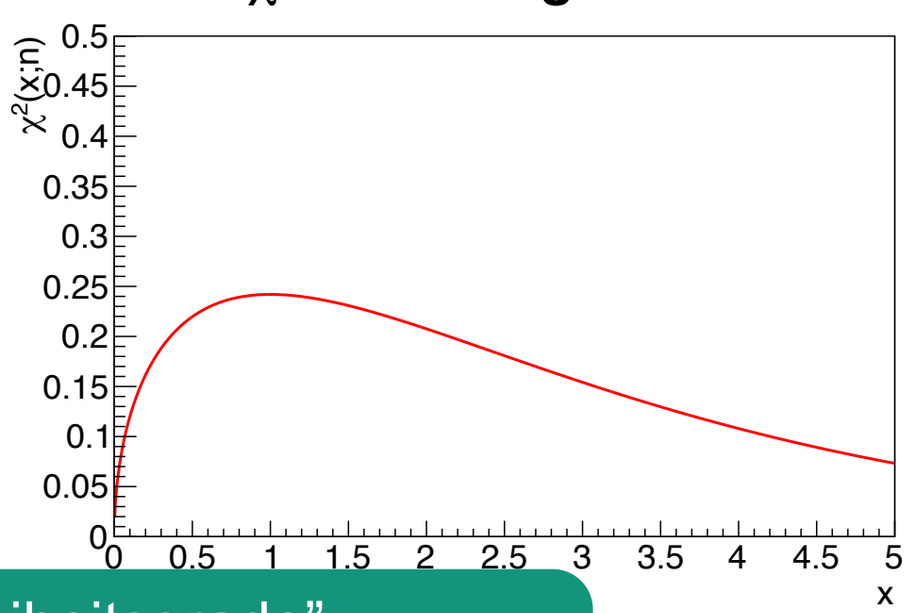
χ^2-Verteilung: n = 1



χ^2-Verteilung: n = 2



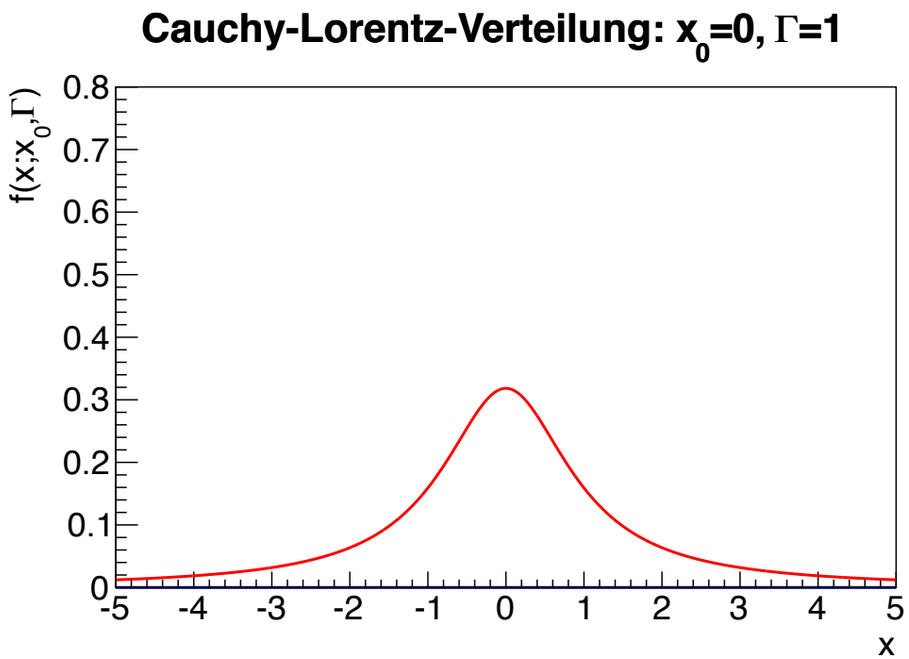
χ^2-Verteilung: n = 3

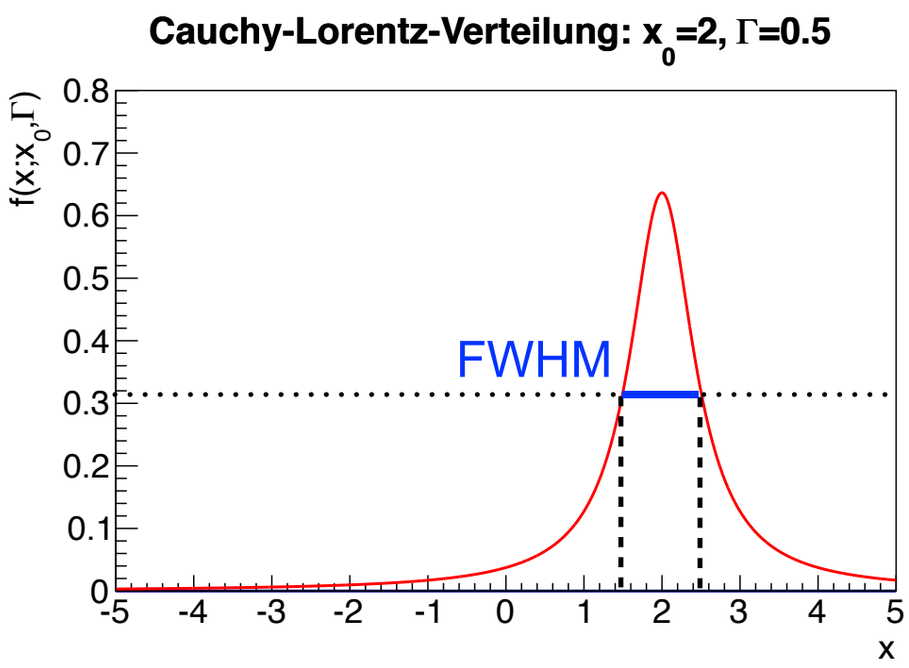


n: „Zahl der Freiheitsgrade”
Γ(x): Euler’sche Gammafunktion

Cauchy-Lorentz-Verteilung

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





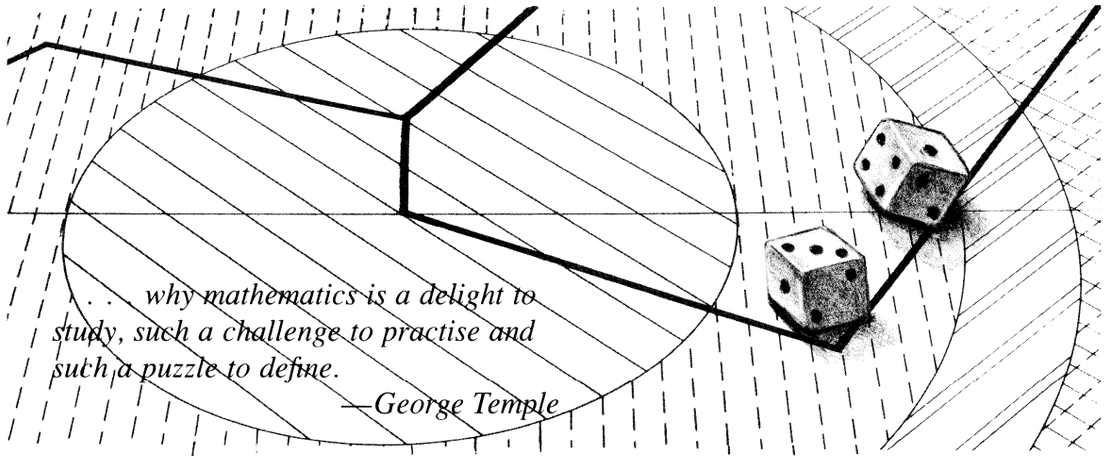
- z. B. Resonanzphänomene
- Eigenschaften
  - E[x] und alle höheren Momente divergent
  - Lagemaß: x_0
  - Maß für Breite: FWHM (full width at half maximum)
- Name in Teilchenphysik: Breit-Wigner-Verteilung

Karlsruher Institut für Technologie

Kapitel 3.4

Monte-Carlo-Methoden

Monte-Carlo-Methoden



[N. Metropolis, Los Alamos Science (1997)]

It's called Monte Carlo because you're playing on someone else's money! (B. Jacobson, Berkeley)

- Monte-Carlo(MC-)Methoden: numerische Techniken zur Berechnung von Wahrscheinlichkeiten mit Zufallszahlen

- Geschichte der MC-Methode
  - Manhattan Project (Los Alamos): Berechnung stochastischer Prozesse für Atombomben mithilfe der ersten Computer
  - J. v. Neumann: Diffusion von Neutronen in spaltbarem Material kann mit MC-Methoden behandelt werden
  - Erste Veröffentlichung: N. Metropolis, S. Unam, The Monte Carlo Method, Journal of the American Statistical Association 44 (1949) 335

Zufallszahlen

- Alle MC-Methoden beruhen auf (Pseudo-)Zufallszahlen
- „Wahre” Zufallszahlen nur aus „echten” stochastischen Prozessen
  - Zeitspanne zwischen zwei radioaktiven Zerfällen
  - PCs: Rauschen der Elektronik, Drift der Systemzeit, Gerätetreiber → Entropiepool, in Unix: /dev/random
  - Nachteil: für viele Anwendungen zu langsam
- Pseudo-Zufallszahlen:
  - Zufallszahlen aus deterministischer Zahlensequenz mit sehr langer Periode
  - Ziel: nur geringe Korrelationen zwischen Zufallszahlen
  - ROOT-Klassen: TRandom, TRandom1, TRandom2, TRandom3

Zufallszahlengeneratoren

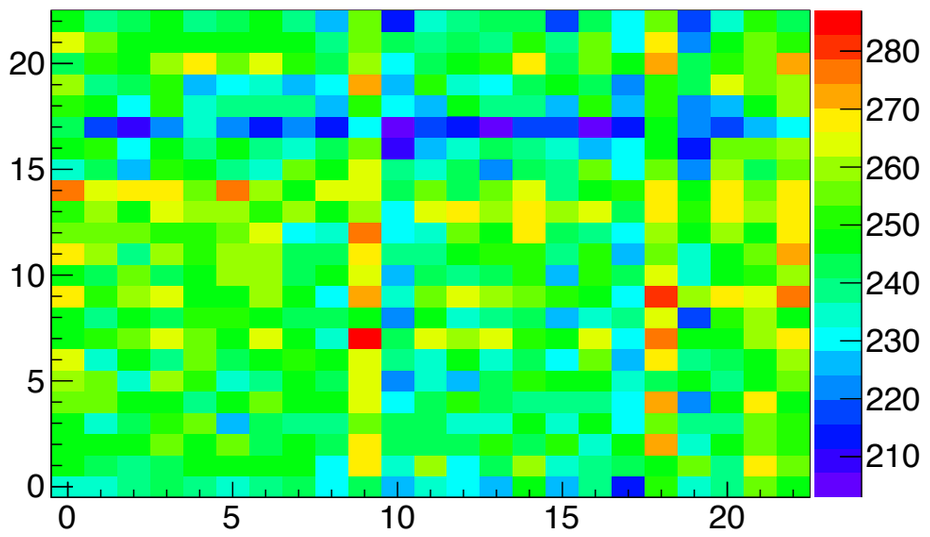
- Multiplikative linear kongruenter Generatoren (LCG):
  - Iterative Erzeugung von Zufallszahlen mit

    $x_{i+1} = (ax_i + b) \mod m$

  - Modulo-Operation: höchstwertiges Bit abgeschnitten → pseudo-zufällig
  - Beispiel: ROOT-Klasse TRandom
  - Probleme: Relativ kurze Perioden (2^{31}), untere Bits der Zufallszahlen sind korreliert → heute nicht mehr verwendet
  - Generell: Startwert („seed”) geschickt wählen!

```
489 Double_t TRandom::Rndm(Int_t)
490 {
491    //  Machine independent random number generator.
492    //  Based on the BSD Unix (Rand) Linear congrential generator.
493    //  Produces uniformly-distributed floating points between 0 and 1.
494    //  Identical sequence on all machines of >= 32 bits.
495    //  Periodicity = 2**31, generates a number in ]0,1].
496    //  Note that this is a generator which is known to have defects
497    //  (the lower random bits are correlated) and therefore should NOT be
498    //  used in any statistical study).
499
500 #ifdef OLD_TRANDOM_IMPL
501    const Double_t kCONS = 4.6566128730774E-10;
502    const Int_t kMASK24  = 2147483392;
503
504    fSeed *= 69069;
505    UInt_t jy = (fSeed&kMASK24); // Set lower 8 bits to zero to assure exact float
506    if (jy) return kCONS*jy;
507    return Rndm();
508 #endif
509
510    const Double_t kCONS = 4.6566128730774E-10; // (1/pow(2,31))
511    fSeed = (1103515245 * fSeed + 12345) & 0x7fffffffUL;
512
513    if (fSeed) return  kCONS*fSeed;
514    return Rndm();
```

Linear Kongruenter Generator: Korrelation

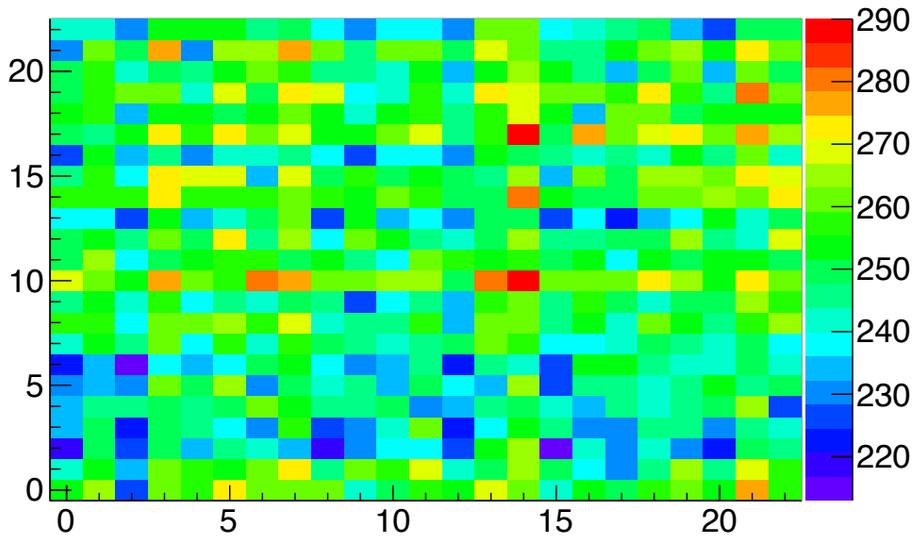


Korrelation innerhalb der Mantisse zweier aufeinander folgender Zufallszahlen

Zufallszahlengeneratoren

```
Double_t TRandom3::Rndm(Int_t)
{
//  Machine independent random number generator.
//  Produces uniformly-distributed floating points in ]0,1]
//  Method: Mersenne Twistor

   UInt_t y;

   const Int_t  kM = 397;
   const Int_t  kN = 624;
   const UInt_t kTemperingMaskB =  0x9d2c5680;
   const UInt_t kTemperingMaskC =  0xefc60000;
   const UInt_t kUpperMask =       0x80000000;
   const UInt_t kLowerMask =       0x7fffffff;
   const UInt_t kMatrixA =         0x9908b0df;

   if (fCount624 >= kN) {
      register Int_t i;

      for (i=0; i < kN-kM; i++) {
         y = (fMt[i] & kUpperMask) | (fMt[i+1] & kLowerMask);
         fMt[i] = fMt[i+kM] ^ (y >> 1) ^ ((y & 0x1) ? kMatrixA : 0x0);
      }

      for (   ; i < kN-1    ; i++) {
         y = (fMt[i] & kUpperMask) | (fMt[i+1] & kLowerMask);
         fMt[i] = fMt[i+kM-kN] ^ (y >> 1) ^ ((y & 0x1) ? kMatrixA : 0x0);
      }

      y = (fMt[kN-1] & kUpperMask) | (fMt[0] & kLowerMask);
      fMt[kN-1] = fMt[kM-1] ^ (y >> 1) ^ ((y & 0x1) ? kMatrixA : 0x0);
      fCount624 = 0;
   }

   y = fMt[fCount624++];
   y ^=  (y >> 11);
   y ^= ((y << 7 ) & kTemperingMaskB );
   y ^= ((y << 15) & kTemperingMaskC );
   y ^=  (y >> 18);

   if (y) return ( (Double_t) y * 2.3283064365386963e-10); // * Power(2,-32)
   return Rndm();
}
```

Mersenne Twister: Korrelation



- Standard-Generator heute: „Mersenne-Twister”
  - Basiert auf Mersenne-Primzahlen (Primzahlen in 2^n–1)
  - Extrem lange Periode: 2^{19937}–1
  - Hinreichend schnell, hochgradig gleichverteilt bis zu 623 Dimensionen
  - Seed: 624 Zufallszahlen aus LCG → 624 Zufallszahlen parallel
  - ROOT: TRandom3