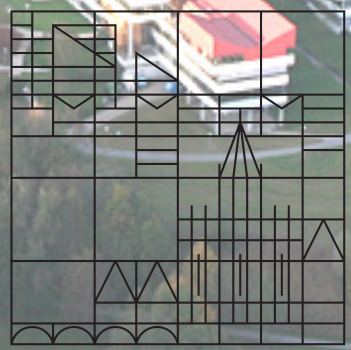
Computerphysik II

Teil 1 - Statistik und Datenanalyse

S. Gerlach

WiSe 2020/21

Universität
Konstanz



Statistik und Datenanalyse

INHALT

1. Kombinatorik
2. Diskrete Wahrscheinlichkeitsverteilungen
3. Numerische Integration
4. Kontinuierliche Wahrscheinlichkeitsverteilungen
5. Schätzmethoden und Anpassung von Funktionen

Kombinatorik - Permutation

Permutation: Anordnung von n Elementen ($n = 3$: 123, 132, 321, 213, 231, 312, 321)
Anzahl:

$$P_n = n!$$

Anzahl mit Wdh.:

$$P_n = \frac{n!}{n_1! n_2! \cdots n_k!}$$

Berechnung:
$n! = 1 \cdot 2 \cdots n$ (iterativ)
$n! = n(n-1)!$ (rekursiv)
In Python: `math.factorial(int)`

Kombinatorik - Permutation

Für große n (Striling-Formel):

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n$$

bzw. $\lg(n!) \approx n(\ln n - 1)$.

Anwendungen Fakultät:

- Binomialkoeffizient (siehe Kombination)
- Binomischer Lehrsatz
- Pascalsches Dreieck
- Exponentialfunktion ($\sum_{n=0}^{\infty} \frac{x^n}{n!}$)
- Taylorreihe
- Leibniz-Produktregel
- ...

Kombinatorik - Variation

Variation: Auswahl von k aus n Elementen (2 aus 3: 12, 13, 21, 23, 31, 32)

Reihenfolge wichtig

Anzahl:

$$V_{n,k} = \frac{n!}{(n-k)!}$$

Anzahl mit Wdh.:

$$V_{n,k} = \frac{n!}{(n-k)!}$$

Kombinatorik - Kombination

Kombination: Auswahl von k aus n Elementen (2 aus 3: 12, 13, 23)

Reihenfolge egal

Anzahl:

$$C_{n,k} = \frac{V_{n,k}}{P_k} = \frac{n!}{k!(n-k)!} = \binom{n}{k}.$$

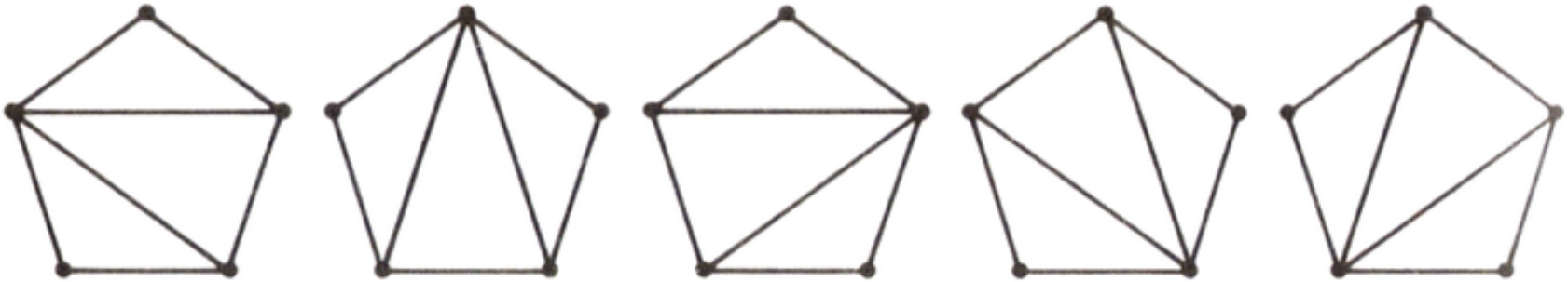
```
1 from scipy.misc import comb
2 print ("%d over %d =" % (n, k), comb(n, k, exact=True))
```

In Python: `itertools` (s. Übung)

Tipp: `list(...)` erzeugt eine iterierbare Liste

Kombinatorik - Beispiel

Catalan-Zahlen



`https://de.wikipedia.org/wiki/Catalan-Zahl`

$$C_n = \frac{1}{n+1}\binom{2n}{n}$$

$$C_{n+1} = \frac{2(2n+1)}{n+2} C_n$$

Diskrete Wahrscheinlichkeitsverteilungen

Zuerst: Zufallsexperimente mit diskreter Anzahl an Ergebnissen.

1. Beispiel: **Binomialverteilung**
Mit welcher Wahrscheinlichkeit gibt es k positive Ergebnisse bei n Versuchen eines Bernoulli-Experiments ("Münzwurf")?

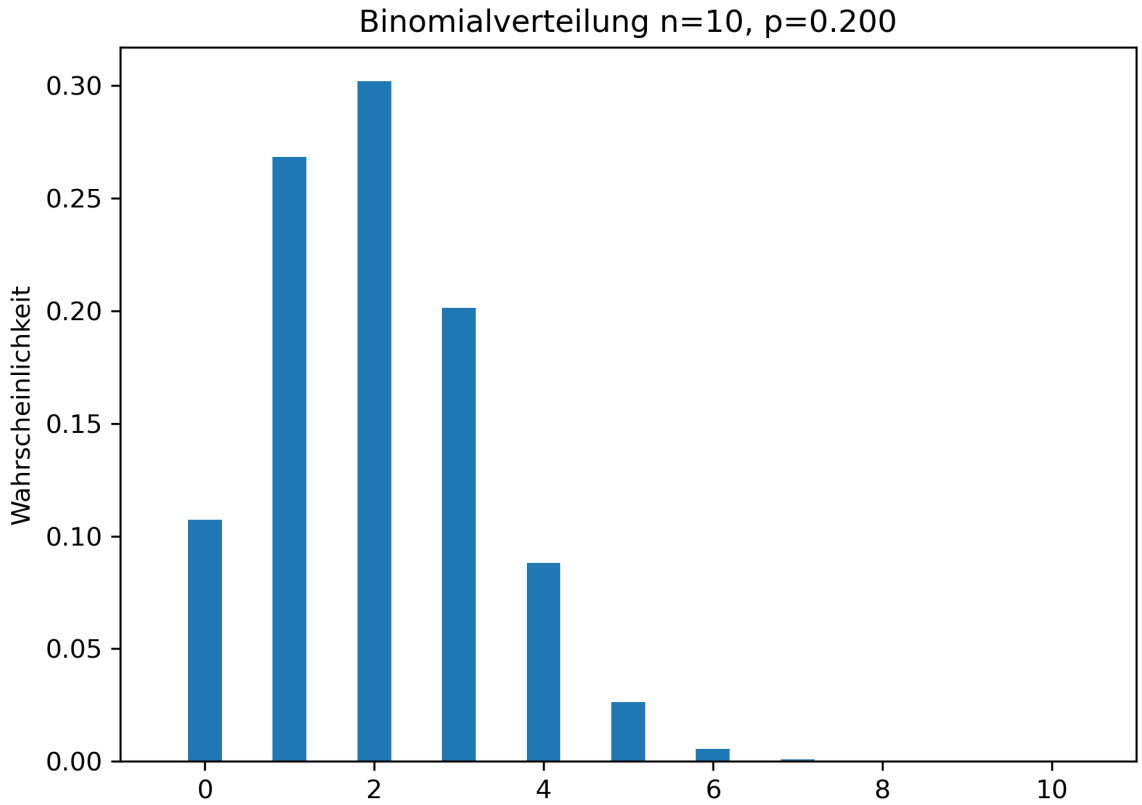
$$\boxed{P(k) = B_{n,p}(k) = \binom{n}{k} p^k (1-p)^{n-k}}$$

Mit $k = 0, \cdots, n$. $\binom{n}{k}$ ist die Anzahl der Kombinationen, da egal ist, wann die positiven Ergebnisse auftreten.

Ideale Münze ($p = 1/2$): $P(k) = \binom{n}{k} \frac{1}{2}^k \frac{1}{2}^{n-k} = \frac{1}{2^n} \binom{n}{k} = \frac{C_{n,k}}{V_{n,2}}$

Binomialverteilung

```
1 from scipy.stats import binom
2 import pylab as pl
3
4 x=pl.arange(-n,n+1,1)
5 pl.bar(x,binom.pmf(x,n,p),width=0.4,align="center")
```

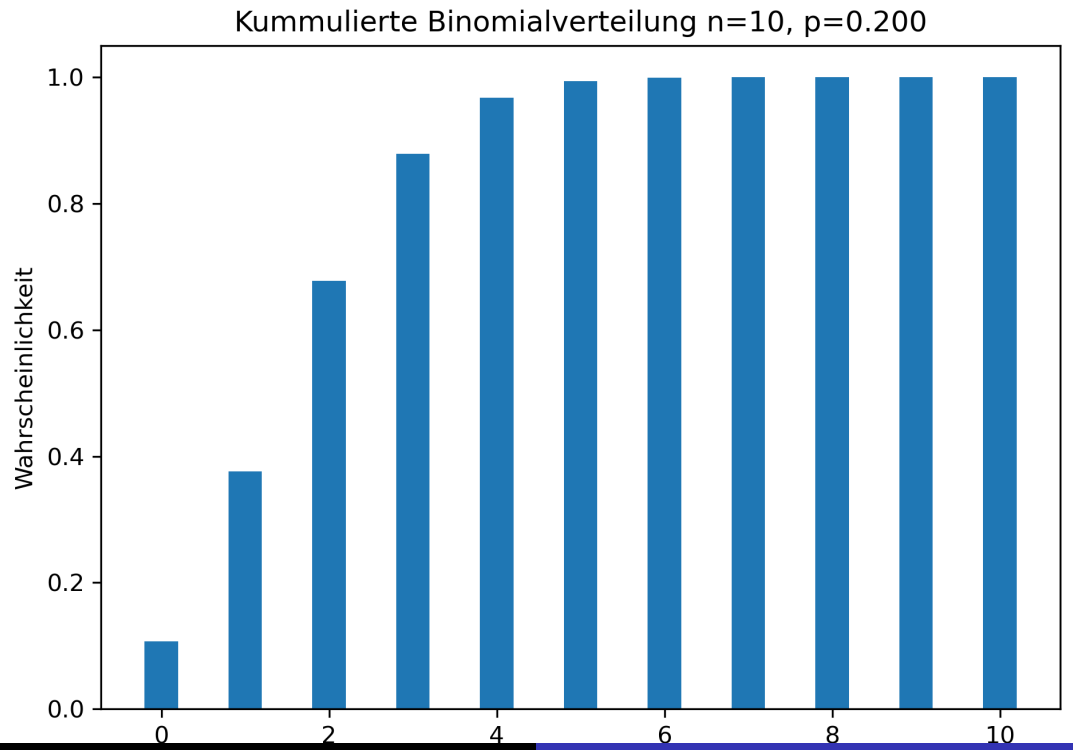


Kumulative Binomialverteilung

Wieviele Ereignisse sind kleiner als ein Wert k_0: Summierte Wahrscheinlichkeit

$$F_K(k_0) = P(K < k_0) = \sum_{k=0}^{\lfloor k_0 \rfloor} \binom{n}{k} p^k (1-p)^{n-k}$$

```
1 pl.bar(x,binom.cdf(x,n,p),width=0.4,align="center")
```



Binomialverteilung

Eigenschaften:

- Normierung: $\sum_{k=0}^{n} P(k) = \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} \overset{\text{binom. Lehrsatz}}{=} (p + (1-p))^n = 1$
- Erwartungswert: $\mu = \langle k \rangle = \sum_{k=0}^{n} kP(k) = \cdots = np$
  (= Summe der einzelnen Erwartungswerte)
- Varianz: $\sigma^2 = \sum_{k=0}^{n} (k-\mu)^2 P(k) = \langle (x - \mu^2) \rangle = \cdots = \langle x^2 \rangle - \langle x \rangle^2 = \cdots = np(1-p)$.

Damit ist die relative Breite:

$$\frac{\sigma}{\mu} = \sqrt{\frac{1-p}{n-p}} \sim \frac{1}{\sqrt{n}}$$

D.h. Erwartungswert lässt sich für $n \to \infty$ beliebig genau bestimmen.

Binomialverteilung - Beispiele

1. **Qualitätssicherung**:
Situation: Produkte am Fliessband sollen kontrolliert werden.
Gesucht ist die Fehlerquote p.
Idee: Messe Wahrscheinlichkeit für Null Fehler bei Auswahl von jeweils n Teilen:

$$W(p) = P(0) = B_{n,p}(k = 0) = (1 - p)^n$$

Damit

$$\boxed{p = 1 - \sqrt{n}P(0)}.$$

Beispiel: $n = 3$, gemessen $P(0) = 0,25 \rightarrow p = 0,37$.

2. **Ising-Spinkette**: s. Buch S. 248

Poissonverteilung

Grenzfall $n \to \infty$ bei konstantem Erwartungswert $\lambda = np$ (d. h. $p \to 0$):

$$\boxed{P_\lambda(k) = \frac{\lambda^k}{k!} \mathrm{e}^{-\lambda}} \qquad \textbf{Poissonverteilung}$$

Anwendung: viele, aber seltene Ereignisse

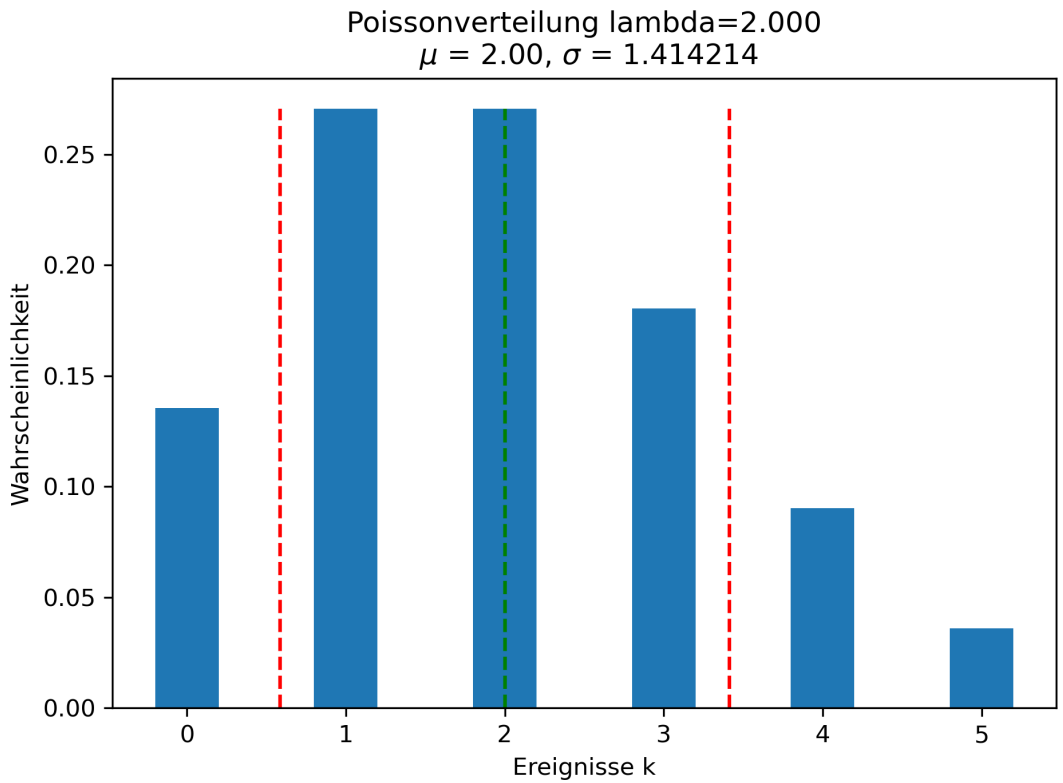
```
1 from scipy.stats import poisson
2 pl.bar(x,poisson.pmf(x,l),width=0.4,align="center")
```

Poissonverteilung

Eigenschaften:

- Normierung:
  $\sum_{k=0}^{n} P(k) = \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} e^{-\lambda} = e^{-\lambda} \sum_{k=0}^{\infty} \frac{\lambda^k}{k!} = e^{-\lambda} e^{\lambda} = 1.$
- Erwartungswert: $\mu = \sigma^2 = \cdots = \lambda$

Relative Breite: $\frac{\sigma}{\mu} = \sim \frac{1}{\sqrt{\lambda}}$.

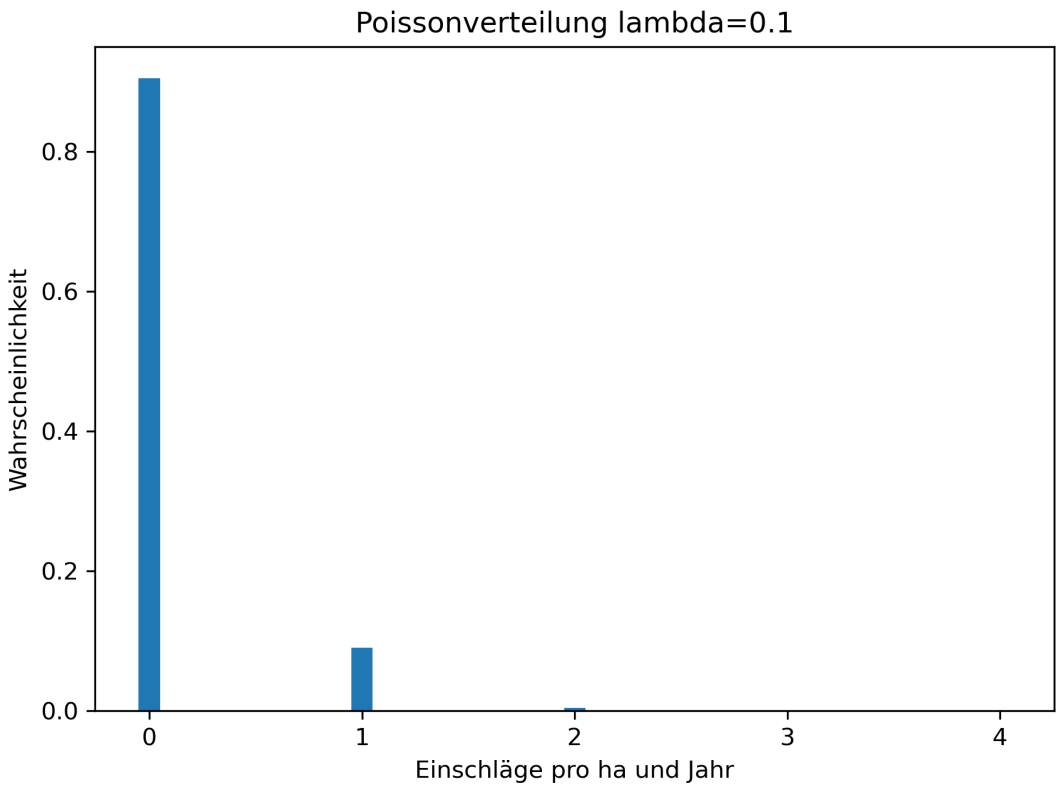


Poissonverteilung - Beispiel 1 - Blitzeinschläge

Im Mittel wurden 0,1 Blitzeinschläge pro Hektar in Deutschland registriert. Mit welcher Wahrscheinlichkeit schlägt kein, ein, zwei, ... Blitze auf einen Hektar ein?

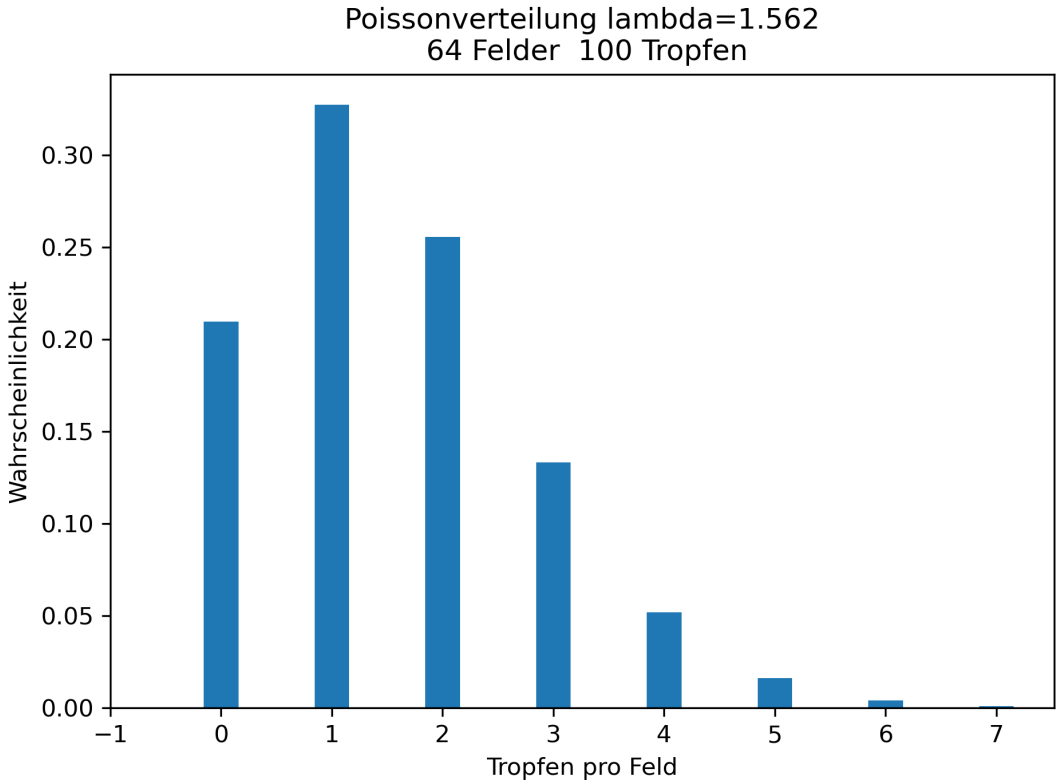
$\mu = 0,1 = \lambda$

$\rightarrow$ 90% kein Einschlag, 10% ein Einschlag.



Poissonverteilung - Beispiel 2 - Schachbrett

Es regnet auf ein Schachbrett. Wieviele Tropfen sind in einem Feld?
$N = 64$ Felder, $n = 100$ Tropfen: $\lambda = \mu = n/N \approx .1,56$:
Immerhin mit 5% Wahrscheinlichkeit 4 Regentropfen im Feld.

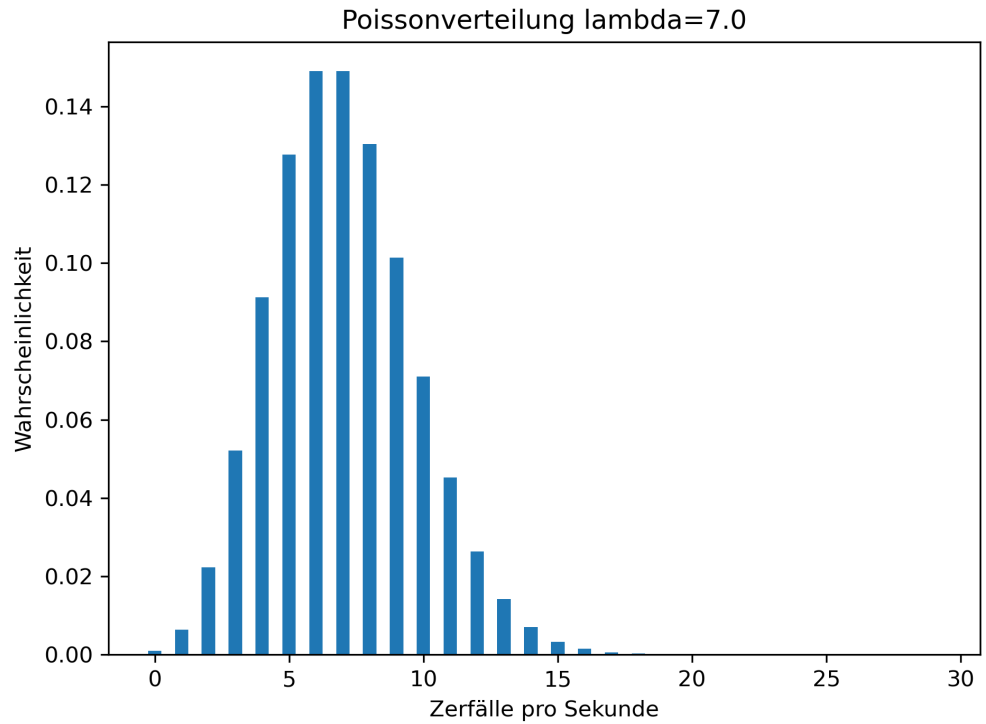


Poissonverteilung - Beispiel 3 - Radiaktiver Zerfall

Die Zerfälle pro Zeiteinheit Δt folgen für $\Delta t \ll t_{\mathrm{H}}$ der Poissonverteilung mit $\lambda = pN$. $p \approx \alpha \Delta t$ ist die Wahrscheinlichkeit für einen Zerfall, N die Anzahl der Teilchen.
Beispiel: $N = 10^{18}$ Atome, $t_{\mathrm{H}} = 4,5$ Mrd. Jahre. Damit $\alpha = \ln 2 / t_{\mathrm{H}}$ und für $\Delta t = 1$ s:

$$\lambda = \alpha N \Delta t \approx 7.$$

Es gibt also im Mittel 7 Zerfälle pro Sekunde mit der Verteilung:



Übergang

Mehr Beispiele: Übungsblatt

Diskrete Verteilungen oft unhandlich. Betrachte Grenzfälle:

$$\begin{array}{ccc} \text{Binomialverteilung} & \xrightarrow[\mu=np=\text{const}]{n\to\infty} & \text{Poissonverteilung} \\ \Big\downarrow{\scriptstyle n\to\infty} & & \Big\downarrow{\scriptstyle \lambda\to\infty} \\ \text{Normalverteilung} & = & \text{Normalverteilung} \end{array}$$

$\rightarrow$ Kontinuierliche Wahrscheinlichkeitsverteilungen

Numerische Integration

Um mit kontinuierlichen Wahrscheinlichkeitsverteilungen zu arbeiten, müssen wir nun nicht mehr Summieren, sondern integrieren. Numerisch brauchen wir dafür die **Numerische Integration**.

Dafür sind wichtig:

- Newton-Cotes-Formeln und Zusammengesetze Formeln
- Uneigentliche Integrale
- Gauß-Quadratur

$\rightarrow$ siehe Buch Kapitel 9.5

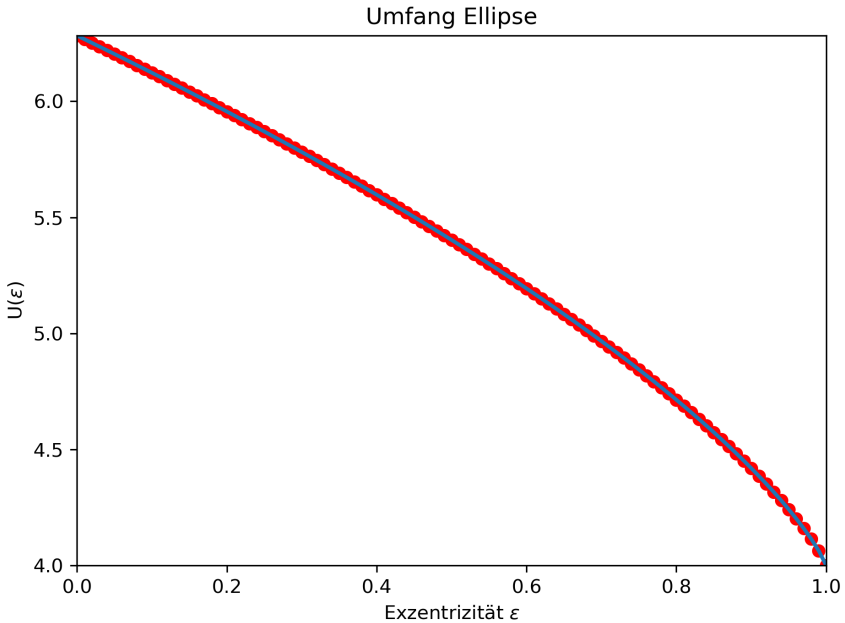
Num. Integration - Beispiel: Kurvenlänge einer Ellipse

Mit $\gamma(t) = \begin{pmatrix} a\cos t \\ b\sin t \end{pmatrix}$ $(t \in [0, 2\pi])$:

$$|\gamma(t)| = \sqrt{a^2 \sin^2 t + b^2 \cos^2 t} = a\sqrt{1 - \varepsilon^2 \cos^2 t}.$$

Damit ist die Kurvenlänge:

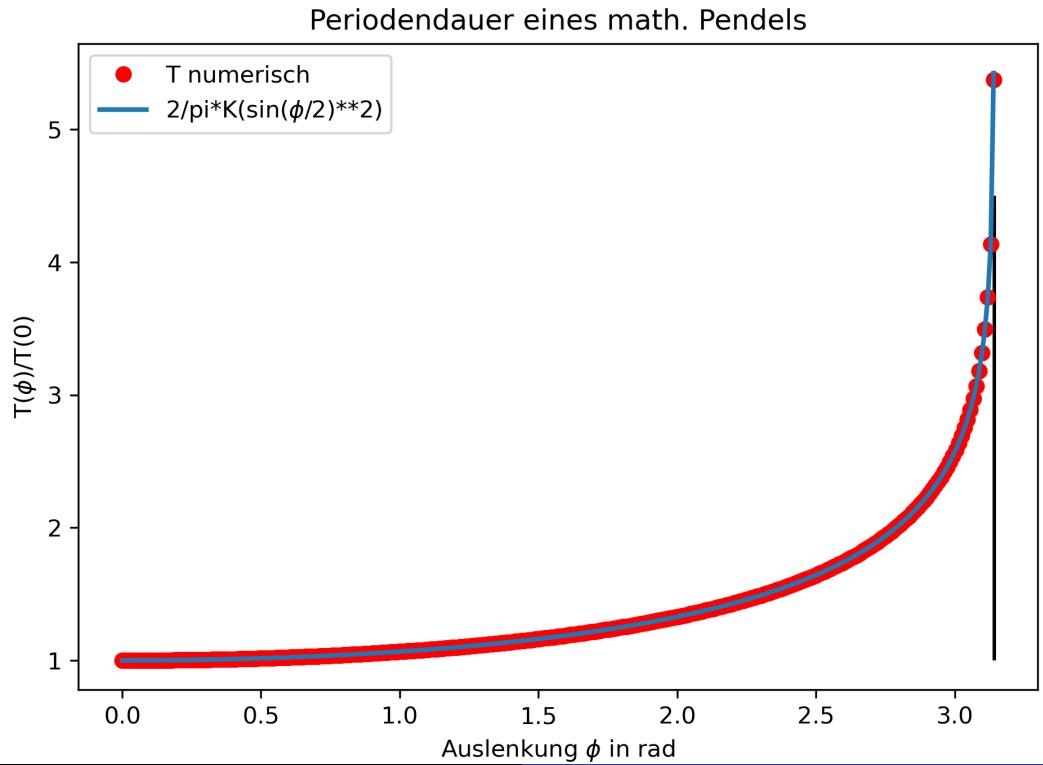
$$L = \int_0^{2\pi} |\gamma(t)| \, \mathrm{d}t = 4a \underbrace{\int_0^{\pi/2} \sqrt{1 - \varepsilon^2 \cos^2 t} \, \mathrm{d}t}_{E(\varepsilon^2), \text{Elliptisches Integral}}.$$



Num. Integration - Beispiel: Periodendauer des math. Pendels

$$T = \frac{4}{\omega} \int_0^{\pi/2} \frac{1}{\sqrt{1 - \sin^2 \frac{\varphi_0}{2} \sin^2 \varphi}} \, \mathrm{d}\varphi = \frac{4}{\omega} \, \mathrm{K}(\sin^2 \frac{\varphi_0}{2}).$$

`scipy.special.ellipk(k**2)`



Kontinuierliche Verteilungen

$\rightarrow$ siehe Buch Kapitel 18.1.3

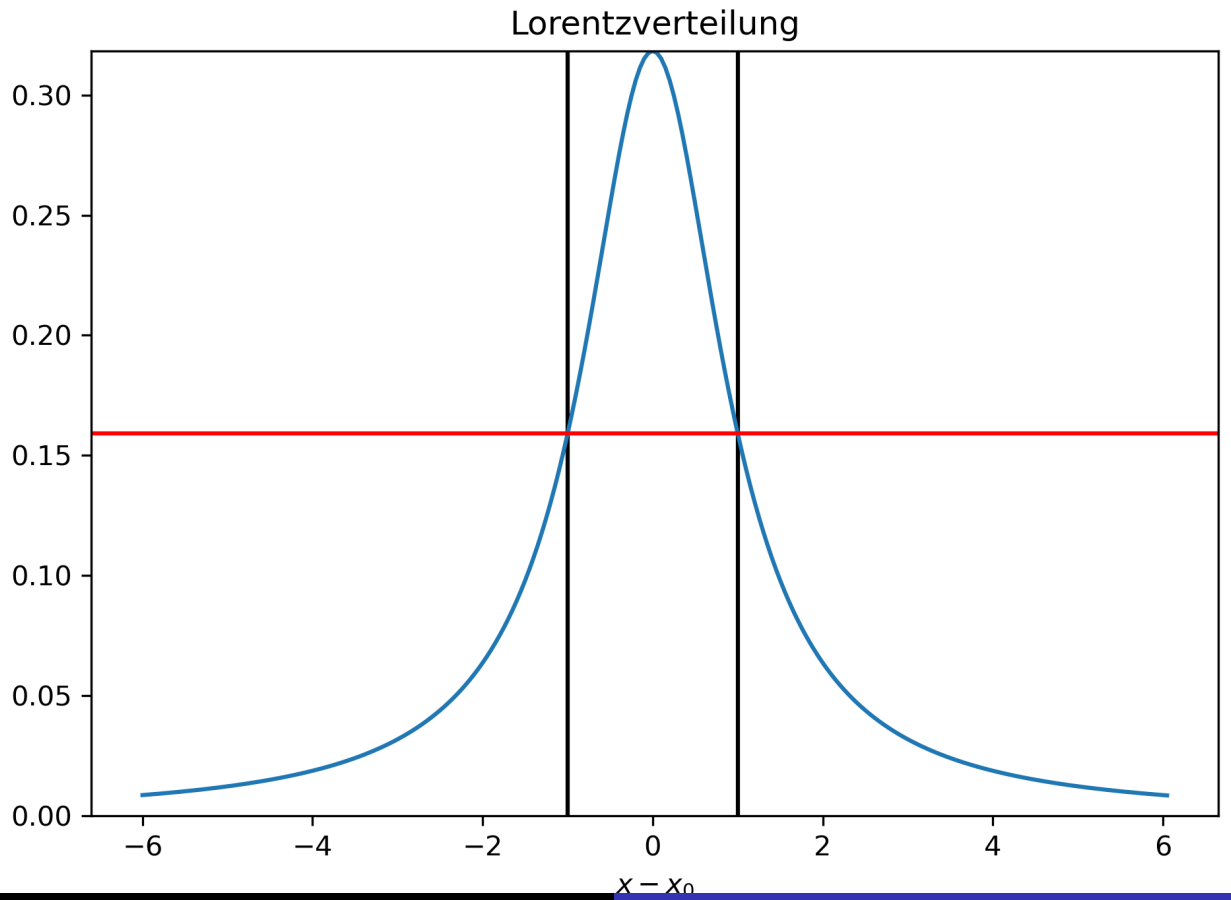
- Exponentialverteilung (Buch)
- Lorentz-Verteilung
- Normalverteilung (Buch)

Beispiele: Maxwell-Boltzmann (Buch)
Mehr Beispiele: Übung

Lorentz-Verteilung

(In der Mathematik: Cauchy-Lorentz-Verteilung)

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



Lorentz-Verteilung

Ursprung: Fourier-Trafo vom exp. Zerfall
Daher bekannt aus: Resonanzkurve, Spektrallinie,
"Breit-Wigner"-Resonanz

Eigenschaften:

- Erwartungswert: $\mu = \int_{-\infty}^{\infty} x f(x)\,\mathrm{d}x = \infty$!
- Varianz: $\sigma^2 = \int_{-\infty}^{\infty} x^2 f(x)\,\mathrm{d}x = \infty$!

Verwende daher Median x_0 und FWHM (*full width half maximum*)
$\Gamma = 2\gamma$ als Linienbreite.

Vertrauensintervall: $\int_{-\Gamma/2}^{\Gamma/2} = 50\%$.
CDF: $F(x) = \int_{-\infty}^{x} f(x')\,\mathrm{d}x' = \frac{1}{\pi}\,\mathrm{atan}(\frac{x-x_0}{\gamma}) + \frac{1}{2}$.

Schätzmethoden

Suche anhand einer Stichprobe die optimalen Parameter einer Verteilung bzw. Funktion
$\rightarrow$ Anpassung der Verteilung/Funtion an die Stichprobe/Messwerte durch Optimierung
2 wichtige Verfahren:

- **Maximum Likelyhood-Methode** [ML] $\rightarrow$ s. Kapitel 18.2.1
- **Kleinste-Quadrate-Methode** (*least square*, LS) s. Kapitel 9.6

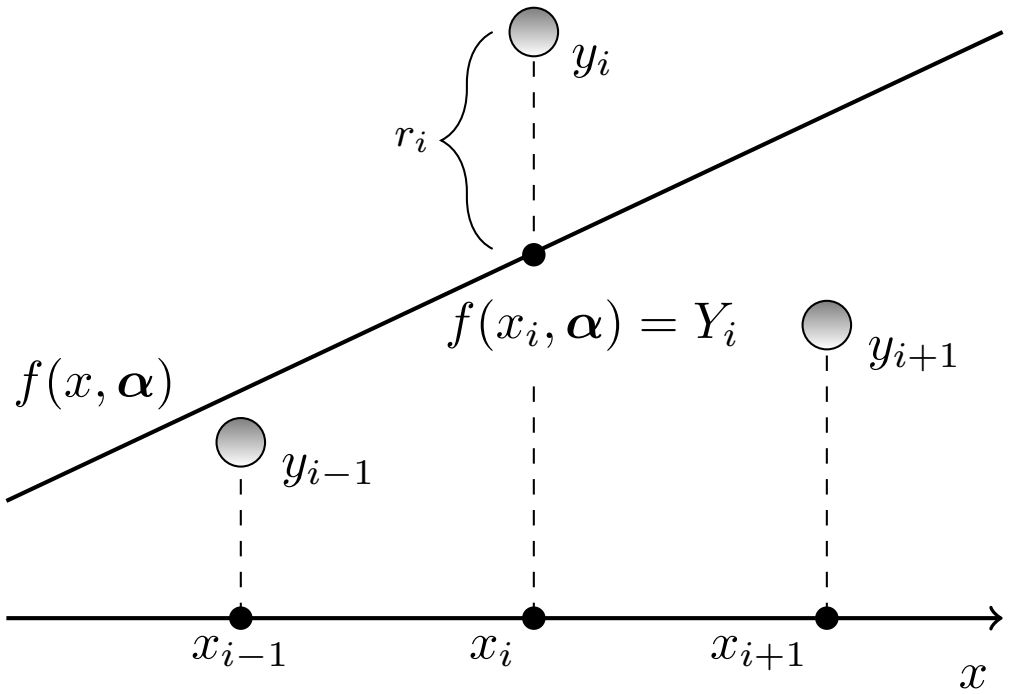
Kleinste-Quadrate-Methode

Anpassung (*fit*) einer Modellfunktion an (Mess-)Daten durch Optimierung der Modellparameter

n Messwerte $y_i(x_i)$ $(i = 1, \cdots, n)$

Modellfunktion: $f(x; \underbrace{\alpha_1, \cdots, \alpha_m}_{\boldsymbol{\alpha}, \text{Modellparameter}})$

Suche Parameter, sodass die Abweichungen (Residuen) möglichst klein sind.



Kleinste-Quadrate-Methode

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} (y_i - f(x_i, \boldsymbol{\alpha}))^2 = \min_{\boldsymbol{\alpha}} \sum_{i=1}^{N} r_i^2(\boldsymbol{\alpha}) = \min_{\boldsymbol{\alpha}} \chi^2(\boldsymbol{\alpha}). \quad (1)$$

Lineare Anpassung: Modellfunktion ist linear ($f(x; a, b) = a + bx$) $\rightarrow$. Optimale Parameter lassen sich direkt berechen. ($\rightarrow$ Kapitel 9.6.1)

Polynomregression: Modellfunktion ist ein Polynom ($\rightarrow$ Kapitel 9.6.2)

Polynomapproximation: Anpassung einer bel. Funktion mit einem Polynom ($\rightarrow$ Kapitel 18.2.2)

Allgemeiner Fall: **Nichtlineare Anpassung**

Nichtlineare Anpassung

Lösung des Optimierungsproblems für eine allg. Modellfunktion mithilfe von Minimierungsmethoden der Linearen Algebra
($\rightarrow$ Kapitel 18.2.3)
[Für Experten:]
Minimierung durch Anwendung des (mehrdimensionalen) Newton-Verfahrens auf den Gradienten der Modellfunktion. Bei Verwendung von $\chi^2(\boldsymbol{\alpha})$ ergibt sich das **Gauß-Newton-Verfahren**.
Um die Konvergenz zu garantieren, führt man einen Parameter λ ein und erhält endlich den **Levenberg-Marquardt-Algorithmus**.
($\rightarrow$ Kapitel 11.3)
In Python:
`scipy.optimize.curve_fit(f,x,y)` $\rightarrow$ `popt,pcor`
Andere Programme mit Fitfunktionen: Gnuplot, Origin, LabPlot, Igorpro, ...