DataCraft

6 Verteilungen

Grundbegriffe

DataCraft

Zufallsexperiment = wiederholbarer Vorgang mit bekannter Menge an Ergebnissen

- Beispiel 1: einmaliges Würfeln mit einem fairen Würfel
- Beispiel 2: Befragung eines Grundschülers zu seinem gestrigen Medienkonsum

Zufallsvariable = Beschreibung des Ergebnisses eines Zufallsexperiments mit einer reellen Zahl; jeder Wert der Zufallsvariable tritt mit einer bestimmten Wahrscheinlichkeit ein

- Beispiel 1: X = Augenzahl des Würfels, $X \in \{1, 2, 3, 4, 5, 6\}$
- Beispiel 2: Y = gestern verbrachte Zeit an Handy, PC und Fernseher in Stunden, $Y \in [0, 24)$

Diskrete Zufallsvariable = Zufallsvariable, die eine abzählbare Menge von Werten annehmen kann

- Die Zufallsvariable X im obigen Beispiel 1 ist diskret

Stetige Zufallsvariable = Zufallsvariable, die jeden beliebigen Wert in einem Intervall oder in einer Menge von Intervallen annehmen kann

- Die Zufallsvariable Y im obigen Beispiel 2 ist stetig (wenn mit beliebiger Genauigkeit
- gemessen wird)

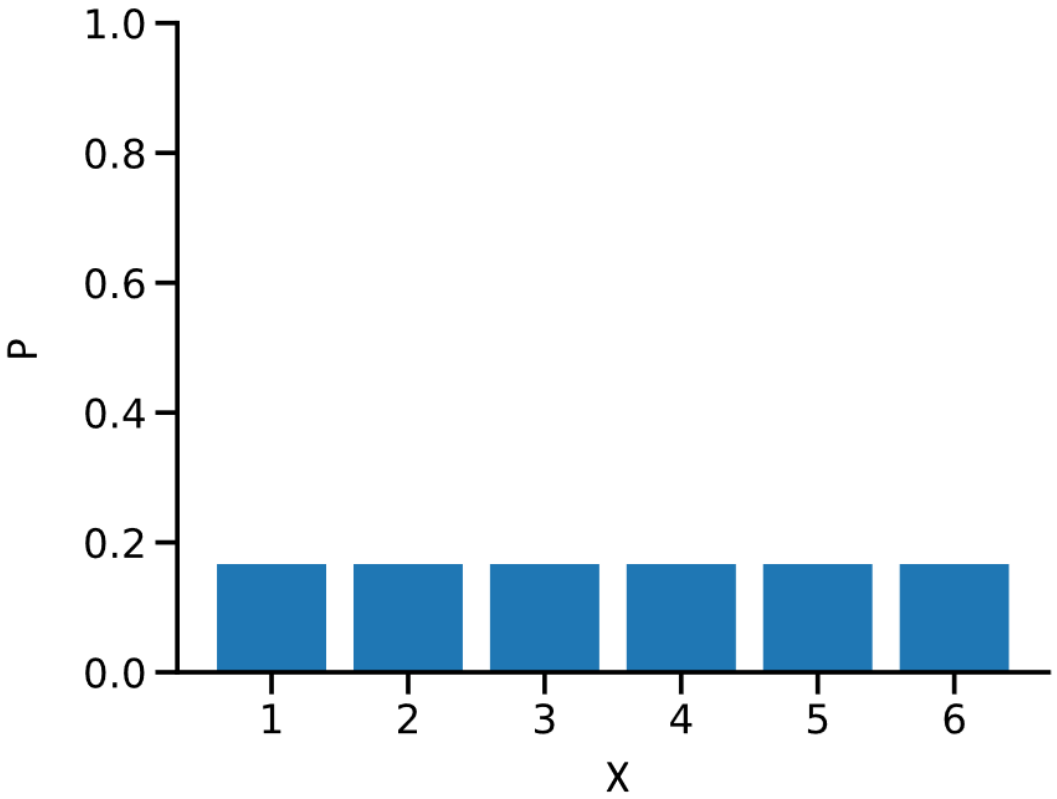
- Jede **diskrete Zufallsvariable** hat eine **Massefunktion** $f(x)$ (=Wahrscheinlichkeitsfunktion). Diese gibt an, mit welcher Wahrscheinlichkeit die Zufallsvariable welchen Wert annimmt.

$$f(1) = P(X = 1) = \frac{1}{6}$$

$$f(2) = \frac{1}{6}$$

$$f(3) = \frac{1}{6} \qquad f(5) = \frac{1}{6}$$

$$f(4) = \frac{1}{6} \qquad f(6) = \frac{1}{6}$$



Bernoulli-Verteilung: Eine Variable mit zwei Ausprägungen. Die Wahrscheinlichkeit für das eine Ereignis ist p, für das andere dementsprechend 1-p

- Münze: Wappen oder Zahl, p = 0.5
- eine Eins würfeln: p = 1/6
- …

Python:

```
import random
random.choices(population=[0,1], weights=[0.2, 0.8])
```

Exkurs: Zufallszahlen mit Numpy

DataCraft

Zuerst einen Zufallsgenerator erstellen

```
import numpy as np
z_gen = np.random.default_rng()
```

Optional einen seed angeben. Dieser startet den Zufallsgenerator an einem speziellen Punkt. Damit lassen sich die gleichen Zahlen bei einem weiteren Durchlauf generieren.

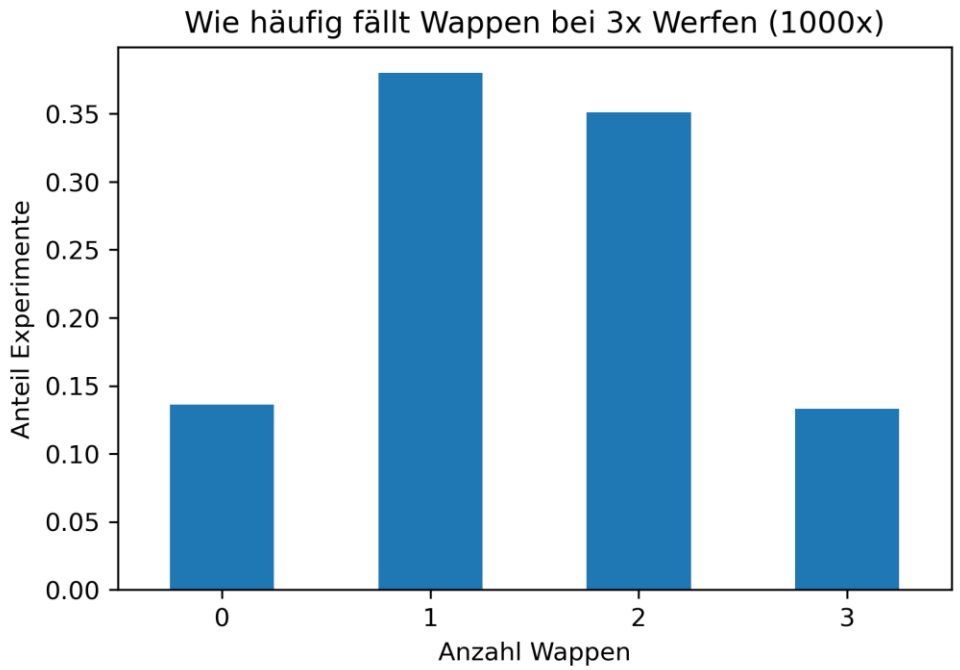
```
z_gen = np.random.default_rng(seed=42)
```

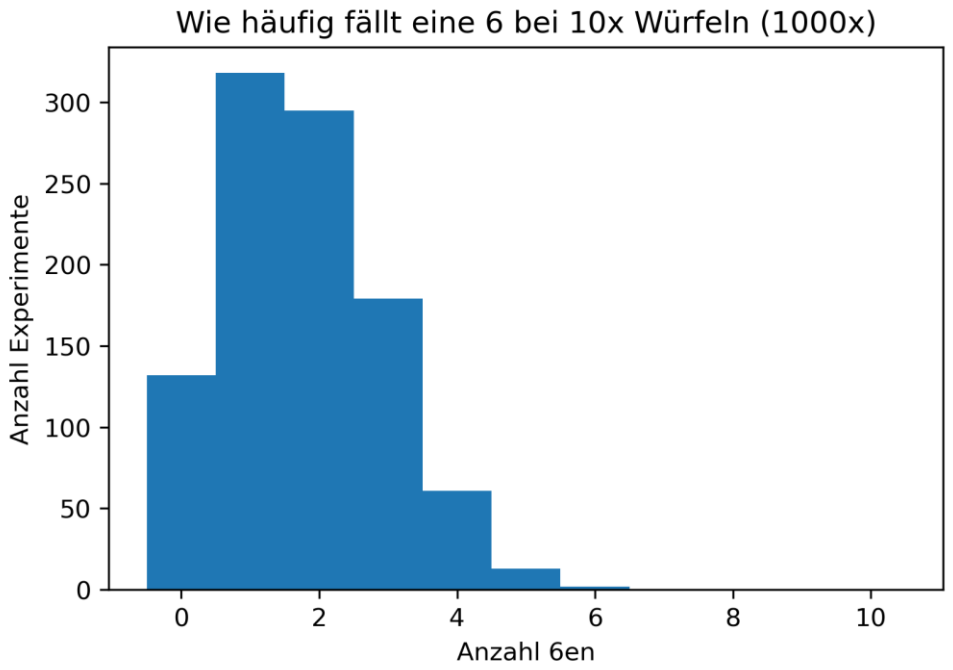
Dann eine oder mehrere Zufallszahlen ziehen. Für jede Verteilung gibt es eine Funktion.

```
df["x"] = z_gen.binomial(n=1, p=0.5, size=100)
```

Binomial-Verteilung: Mehrfaches (unabhängiges) Ausführen eines Bernoulli-Experiments

- Anzahl Wappen bei 3x Werfen einer Münze
- Anzahl Sechsen bei 10x Werfen eines Würfels





Die Wahrscheinlichkeit für eine Anzahl (X=k) kann direkt über diese Massefunktion berechnet werden:

$$p_k = \binom{n}{k} p^k (1-p)^{n-k}$$

- Was ist die Wahrscheinlichkeit für 2x Wappen bei 3x Werfen einer Münze?
- Was ist die Wahrscheinlichkeit für 4x Sechs bei 10x Werfen eines Würfels?

Die Wahrscheinlichkeit für bis zu einer Anzahl (x <= k) kann als Summe der einzelnen Wahrscheinlichkeiten berechnet werden

$$p_{\leq k} = p_0 + p_1 + \cdots + p_k$$

Fasst man diese Werte als Funktion, in die man die Parameter n, p und k einsetzt, auf, nennt man diese (kumulierte) Verteilungsfunktion.

$$F(n; p; k) = p_{\leq k} = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$$

Verteilungsfunktionen (CDF = cumulative distribution function) sind ein allgemeines Prinzip in der Stochastik und es gibt sie für jede Verteilung.

Die Verteilungsfunktion an Wert k entspricht "höchstens k Mal". Über die Gegenwahrscheinlichkeit lässt sich auch "mindestens k Mal" berechnen.

Achtung: Das Gegenteil von $\leq$ (*kleiner oder gleich*) ist $>$ (*echt größer*)

$$P(X \geq k) = P(X > k-1) = 1 - P(X \leq k-1) = 1 - p_{k-1}$$

- Was ist die Wahrscheinlichkeit für höchstens 2x Wappen bei 3x Werfen einer Münze?
- Was ist die Wahrscheinlichkeit für mindestens 4x Sechs bei 10x Werfen eines Würfels?

Das Package scipy bzw. dessen Untermodul scipy.stats bietet zu vielen Verteilungen die entsprechenden kumulierten Verteilungsfunktionen.
Installation mit `conda install scipy`

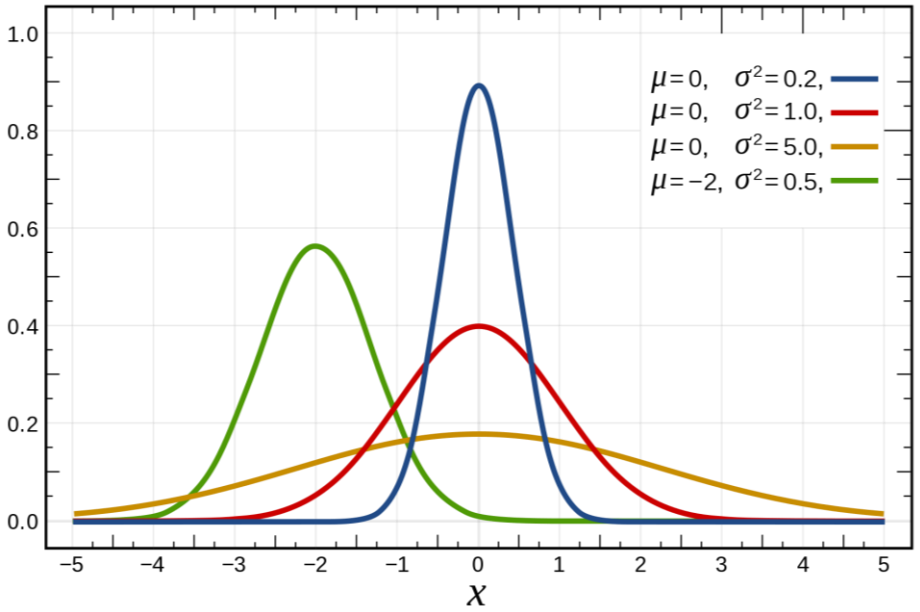
```
import scipy.stats

# Wahrscheinlichkeit für höchstens 2x Wappen bei 3x Werfen
scipy.stats.binom.cdf(k=2, n=3, p=0.5)

# Wahrscheinlichkeit für mindestens 4x Sechser bei 10x Würfeln
1 - scipy.stats.binom.cdf(k=3, n=10, p=1/6)

# Wahrscheinlichkeit für 2 bis 4 Sechser bei 10x Würfeln
scipy.stats.binom.cdf(k=4, n=10, p=1/6) -
      scipy.stats.binom.cdf(k=1, n=10, p=1/6)
```

Wird n immer größer, dann nähert sich die Binomialverteilung der **Normalverteilung** an (Faustregel: ab n=30)

Die Normalverteilung ist der wichtigste Verteilungstyp für stetige Zufallsvariablen (d.h. kontinuierliche Werte)

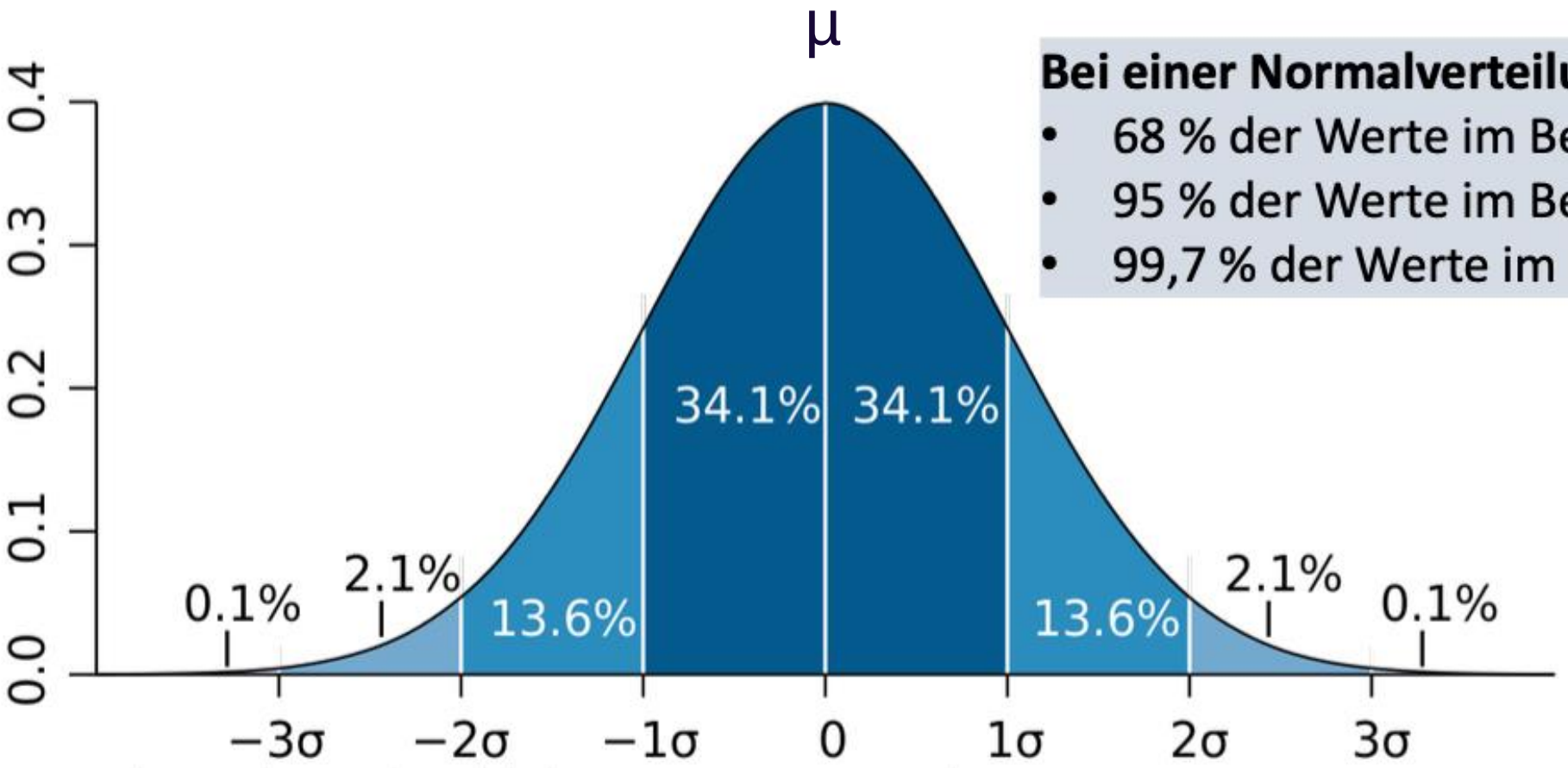
Eine Normalverteilung wird durch zwei Parameter beschrieben, dem Erwartungswert (~ Mittelwert) μ (griechisch my) und der Standardabweichung σ (griechisch sigma)

Die **Standard-Normalverteilung** hat μ=0 und σ=1, man schreibt auch N(0,1)



Alle Normalverteilungen haben folgende Charakteristika:

- glockenförmig
- Symmetrisch
- Modus, Median und Mittelwert fallen zusammen.
- Die Verteilung nähert sich asymptotisch der x-Achse (geht also von $-\infty$ bis $+\infty$).
- Extreme Ereignisse sind stets möglich, aber sehr unwahrscheinlich

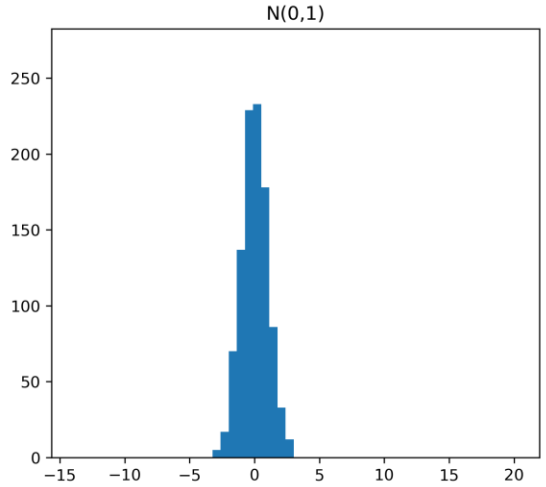


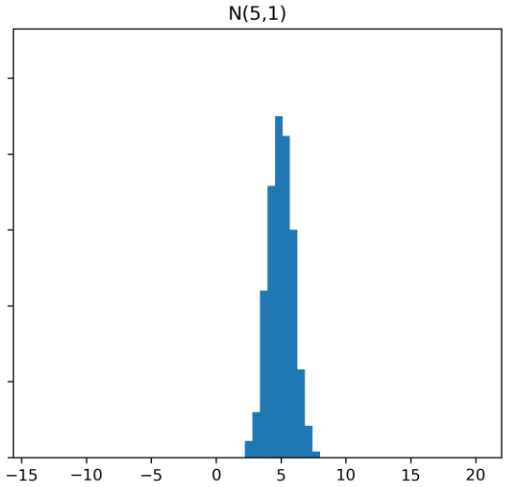
Bei einer Normalverteilung liegen

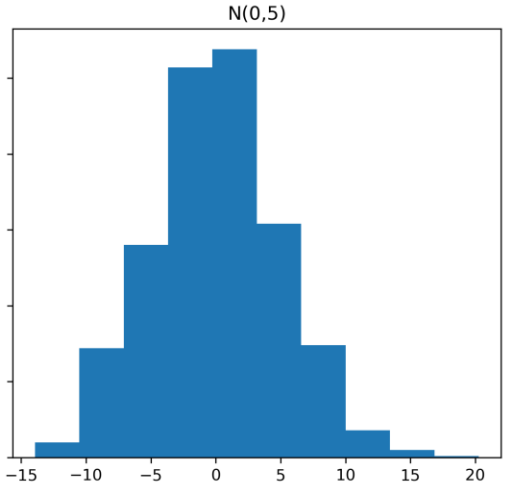
- 68 % der Werte im Bereich $\mu \pm 1 \cdot \sigma$
- 95 % der Werte im Bereich $\mu \pm 2 \cdot \sigma$
- 99,7 % der Werte im Bereich $\mu \pm 3 \cdot \sigma$.

Erzeugen normalverteilter Stichproben

```
n = 1000
normal_0_1 = z_gen.normal(loc=0, scale=1, size=n)
normal_5_1 = z_gen.normal(loc=5, scale=1, size=n)
normal_0_5 = z_gen.normal(loc=0, scale=5, size=n)

df = pd.DataFrame({"N(0,1)":normal_0_1, "N(5,1)":normal_5_1,
                   "N(0,5)":normal_0_5})
df.describe()
```



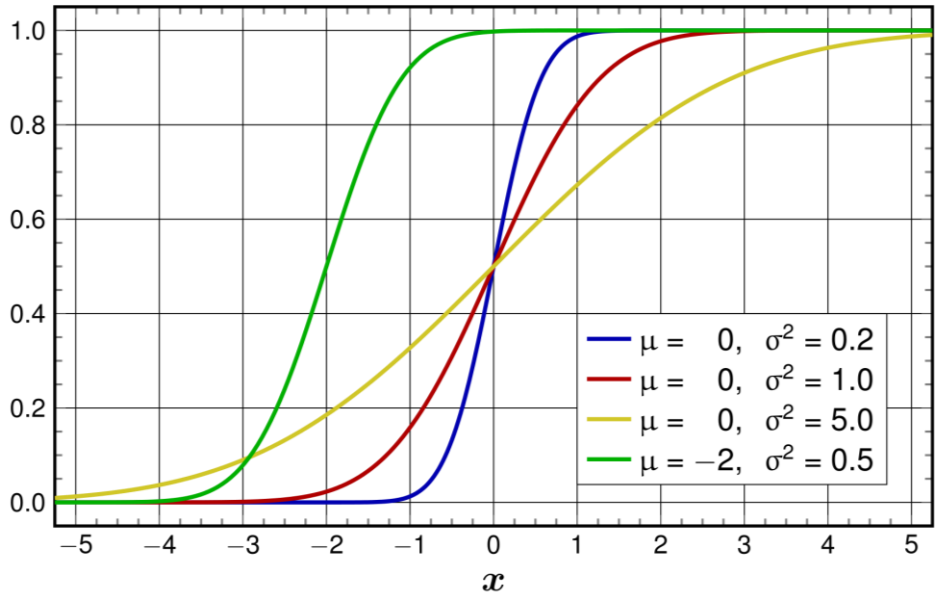




DataCraft

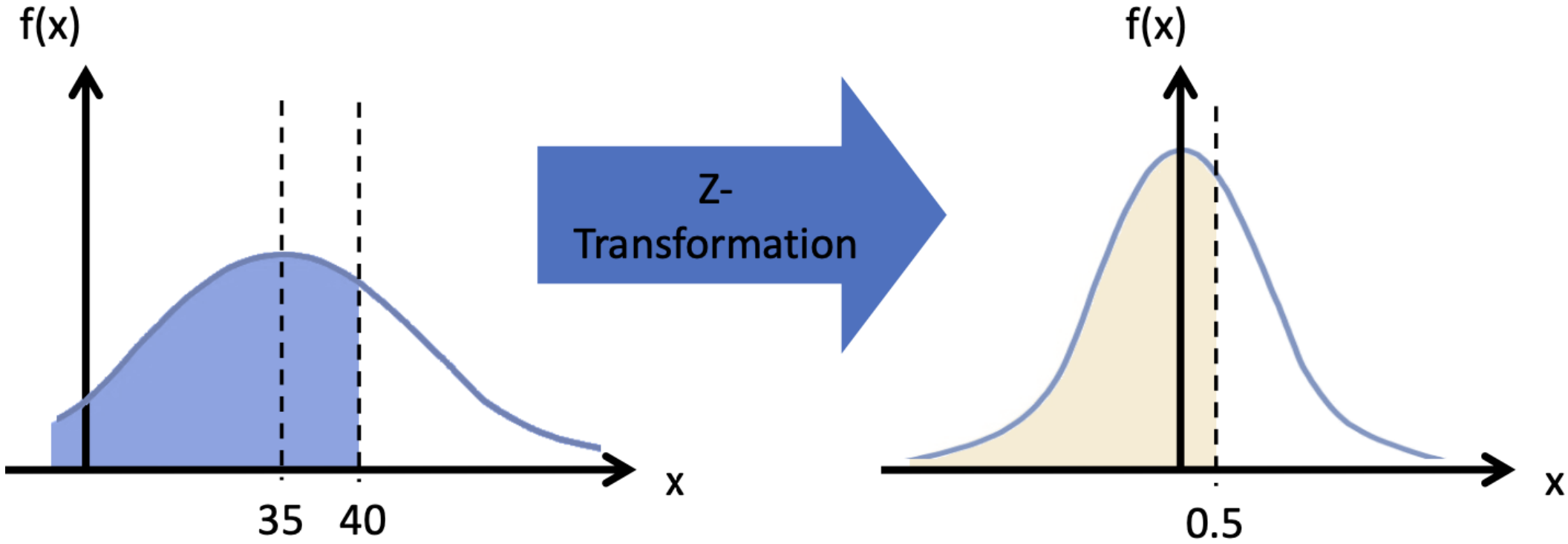
Auch für die Normalverteilung gibt es eine kumulierte Verteilungsfunktion

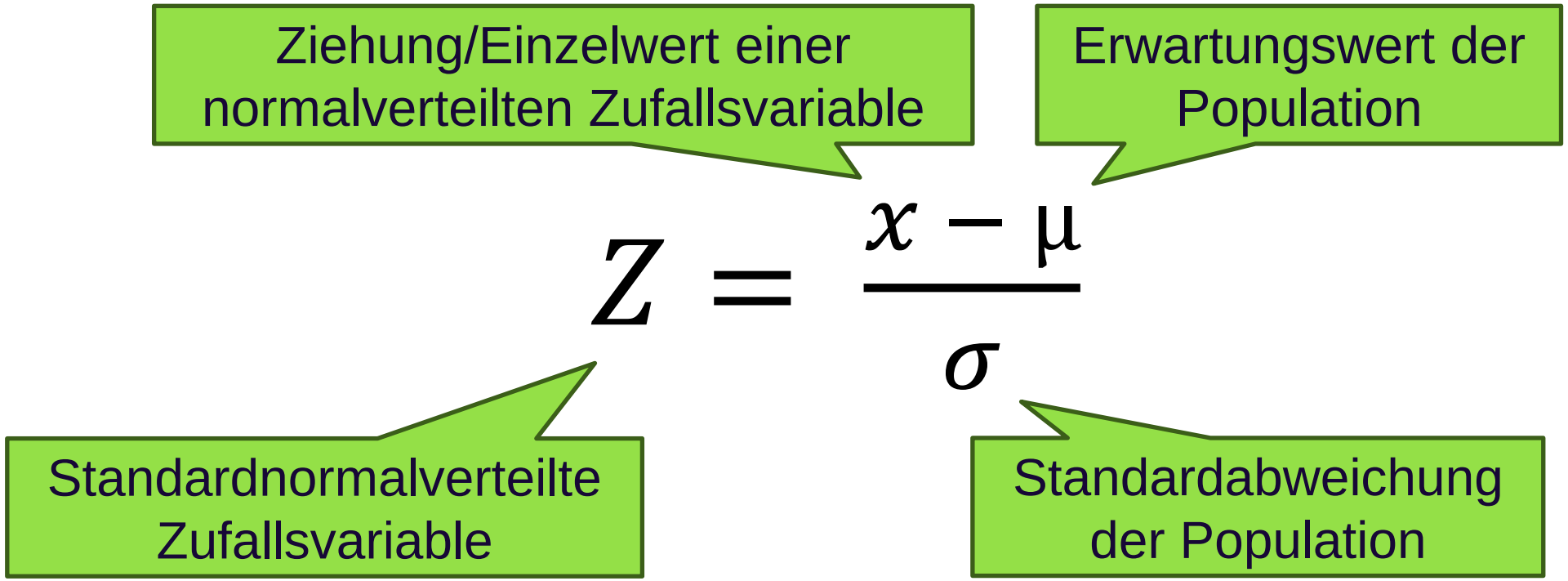
```
import scipy.stats

scipy.stats.norm.cdf(0.55,0,1)
# 0.70884
# ca. 70% der Werte sind kleiner als 0,55

# Umkehrfunktion der CDF = Quantilsfunktion
# Für welchen Wert liegen x% darunter
# PPF = percent point function
scipy.stats.norm.ppf(0.7, 0, 1)
```



Es gilt allgemein: Die kumulierte Verteilungsfunktion ist die Umkehrfunktion der Quantilsfunktion

Für die Standardnormalverteilung ($\mu = 0 \mid \sigma = 1$) gibt es Nachschlagetabellen, bei welchen Werten (sog. Z-Werte) welche kumulierten Wahrscheinlichkeiten vorliegen. Daher kann es nützlich sein, normalverteilte Variablen zu Standardisieren. Diese standardisierung nennt man Z-Transformation.




Ziehung/Einzelwert einer normalverteilten Zufallsvariable
Erwartungswert der Population
$Z = \frac{x - \mu}{\sigma}$
Standardnormalverteilte Zufallsvariable
Standardabweichung der Population