# Diskrete Verteilungen

- **Binomialverteilung:** $X \sim Bin(n, \pi)$ $\quad E(X) = n \cdot \pi$ $\quad Var(X) = n \cdot \pi (1-\pi)$

$$F(x) = \sum_{i=0}^{x} \binom{n}{i} \pi^i (1-\pi)^{n-i} \qquad P(X=x) = \binom{n}{i} \pi^i (1-p)^{n-i}$$

→ Verteilung der Häufigkeit X eines Ereignisses bei n unabhängigen Wiederholungen eines Versuchs.

- **Poisson-Verteilung:** $X \sim Pois(\lambda)$ $\quad E(X) = \lambda$ $\quad Var(X) = \lambda$

$$F(x) = \sum_{i=0}^{x} e^{-\lambda} \cdot \frac{\lambda^x}{x!} \qquad P(X=x) = e^{-\lambda} \cdot \frac{\lambda^x}{x!}$$

→ Verteilung der Anzahl Ereignisse in gegebenem Zeitintervall (unabhängig und in konstanter Rate, z.B. radioaktiver Zerfall)

Approximation der Binomialverteilung: für $n \to \infty$: $p \to 0$, $n \cdot p \to \lambda$

Python:

st. binom

st. poisson

## Stetige Verteilungen ($F(x)$: kum. Verteilungsfunktion, $f(x)$: W'keitsdichte)

- **Uniforme Verteilung:** $X \sim Unif(a, b)$ $\quad E(X) = \frac{a+b}{2}$ $\quad Var(X) = \frac{(b-a)^2}{12}$

Breite: $b-a$ $\quad$ Höhe: $\frac{1}{b-a}$ $\quad$ Fläche: $1$ $\quad \sigma = \frac{b-a}{\sqrt{12}}$

$$F(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & x > b \end{cases} \qquad f(x) = \begin{cases} \frac{1}{b-a}, & a \le x \le b \\ 0, & \text{sonst} \end{cases}$$

Python: $X \sim Unif(3, 7)$: $a = 3$, $b = 7$ $\quad$ loc = a, scale = b − a

st.uniform

$P(4 \le X \le 6)$: uniform.cdf(x = 6, loc = 3, scale = 4) − uniform.cdf(x = 4, loc = 3, scale = 4)

- **Exponentialverteilung:** $X \sim Exp(\lambda)$ $\quad E(X) = \frac{1}{\lambda}$ $\quad Var(X) = \frac{1}{\lambda^2}$ $\quad \sigma = \frac{1}{\lambda}$

$$F(x) = \begin{cases} 1 - e^{-\lambda x}, & x \ge 0 \\ 0, & \text{sonst} \end{cases} \qquad f(x) = \begin{cases} \lambda \cdot e^{-\lambda x}, & x \ge 0 \\ 0, & \text{sonst} \end{cases}$$

Approximation der Poisson-Verteilung: $T \sim Exp(\lambda) \rightarrow N \sim Pois(\lambda \cdot t)$

$T$: Ausfallzeit, $N$: Anzahl Ausfälle, $t$: Intervall (Zeit zw. Ausfällen)

Python: $X \sim Exp(3)$: $\lambda = 3$ $\quad$ scale = $1/\lambda$

st.expon

$P(0 \le X \le 4)$: expon.cdf(x = 4, scale = 1/3)

· Normalverteilung: $X \sim \mathcal{N}(\mu, \sigma^2)$ $E(X) = \mu$ $Var(X) = \sigma^2$

$$F(x) = \int_{-\infty}^{x} f(y)\,dy \qquad f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Python: $IQ \sim \mathcal{N}(100, 15^2)$: $\mu = 100$, $\sigma = 15$ loc = 100, scale = 15

st. norm

$P(IQ > 130) = 1 - P(X \leq 130)$: 1 - norm.cdf(x = 130, loc = 100, scale = 15)

· Standardnormalverteilung: $X \sim \mathcal{N}(0, 1)$ $E(X) = \mu = 0$ $Var(X) = \sigma^2 = 1$

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

Standardisierung ($\mathcal{N}(\mu, \sigma^2) \to \mathcal{N}(0, 1)$): $z = \frac{x - \mu}{\sigma}$ $X \sim \mathcal{N}(\mu, \sigma^2)$ $Z \sim \mathcal{N}(0, 1)$

Python:

st. norm loc = 0, scale = 1 (Standardwerte → weglassen!)

# Modelle für Messdaten

Wahrscheinlichkeitsdichte: $P(a < X \leq b) = F(b) - F(a) = \int_a^b f(x)\,dx$ ; $\int_{-\infty}^{\infty} f(x)\,dx = 1$

Erwartungswert: $E(X) = \mu_x = \int_{-\infty}^{\infty} x \cdot f(x)\,dx$ $\quad E(X^2) = \int_{-\infty}^{\infty} x^2 \cdot f(x)\,dx$

Varianz: $Var(X) = \sigma_x^2 = E[(X - E(X))^2] = \int_{-\infty}^{\infty} (x - E(X))^2 \cdot f(x)\,dx$

$Var(X) = E(X^2) - [E(X)]^2 \Leftrightarrow E(X^2) = Var(X) + [E(X)]^2$

Eigenschaften von Erwartungswert & Varianz:

$E(a + bX) = a + b \cdot E(X)$ $\quad Var(a + bX) = b^2 \cdot Var(X)$

$E(X + Y) = E(X) + E(Y)$ $\quad Var(X + Y) = Var(X) + Var(Y)$ $\quad (X \& Y$ i.)

$E(X \cdot Y) = E(X) \cdot E(Y)$ $\quad (X \& Y$ i.)

$E(g(X)) = \sum_i g(x_i) \cdot p(x_i)$ bzw. $\int_{-\infty}^{\infty} g(x) \cdot f(x)\,dx$ $\quad$ (diskret & stetig)

Quantile: Umkehrung der kumulativen Verteilungsfunktion $\quad$ cdf $\leftrightarrow$ ppf

$P(X \leq q(\alpha)) = \alpha$ $\quad F(q(\alpha)) = \alpha \Leftrightarrow q(\alpha) = F^{-1}(\alpha)$

$\alpha$-Quantile für $Y = a + bX$: $q_Y(\alpha) = a + b \cdot q_X(\alpha)$

$$f_Y(y) = \frac{1}{b} f_X\left(\frac{y - a}{b}\right)$$

Lineare Transformation: $X \sim \mathcal{N}(\mu, \sigma^2)$, $Y = a + bX \sim \mathcal{N}(a + b\mu, b^2\sigma^2)$

Beispiel: Temperatur Grad Celsius → Fahrenheit:

$$T_F = \frac{9}{5} \cdot T_C + 32 \; ; \; a = \frac{9}{5}, \; b = 32, \; \sigma_F = b \cdot \sigma_C = \frac{9}{5}\sigma_C$$

($X \& Y$ i.): $X$ und $Y$ unabhängig!

Gesetz der grossen Zahlen (GGZ): je grösser n, desto näher $\bar{X}_n$ am Erwartungswert; $n \to \infty \Rightarrow \bar{X}_n \to \mu$

Summe & rel. Häufigkeit:

$$\underline{E(S_n)} = E(X_1 + X_2 + \dots + X_n) = \sum_{i=1}^{n} E(X_i) = \underline{n \cdot \mu}$$

$$\underline{Var(S_n)} = Var(X_1 + X_2 + \dots + X_n) = \sum_{i=1}^{n} Var(X_i) = \underline{n \cdot \sigma_x^2}$$

$$\sigma(S_n) = \sqrt{n} \cdot \sigma_x$$

$$\underline{E(\bar{X}_n)} = E\left(\frac{X_1 + X_2 + \dots + X_n}{n}\right) = \frac{1}{n} \sum_{i=1}^{n} (E(X_i)) = \frac{1}{n} \cdot n\mu = \underline{\mu}$$

$$\underline{Var(\bar{X}_n)} = Var\left(\frac{X_1 + X_2 + \dots + X_n}{n}\right) = \frac{1}{n^2} \sum_{i=1}^{n} (Var(X_i)) = \frac{1}{n^2} \cdot n \cdot \sigma^2 x = \underline{\frac{\sigma_x^2}{n}}$$

Standardfehler: $\sigma(\bar{X}_n) = \frac{\sigma_x}{\sqrt{n}}$ relativer Fehler: $\bar{x}_n \pm \frac{\sigma(\bar{X}_n)}{\bar{x}_n} \cdot 100\%$

Zentraler Grenzwertsatz: $X_i \sim \mathcal{N}(\mu, \sigma^2)$, $X_1, X_2, \dots, X_n$ i.i.d. „je grösser n, desto besser die Approximation!"

$$S_n \approx \mathcal{N}(n \cdot \mu, n \cdot \sigma_x^2) \quad \bar{X}_n \approx \mathcal{N}\left(\mu, \frac{\sigma_x^2}{n}\right)$$

<u>QQ-Plot</u> „Passen Messdaten zu bestimmter Verteilung?" (hier: Normalverteilung)

```
x = pd.Series([...])  # Messwerte
alpha_k = (np.arange(1, x.size+1) - 0.5) / x.size
quant_theor = st.norm.ppf(q=alpha_k, loc=x.mean(), scale=x.std())
quant_empir = np.sort(x)
plt.plot(quant_theor, quant_empir, 'o')
# alternativ
st.probplot(x, plot=plt)
```

## Parameterschätzung

- **Momentenmethode**: Verteilung bekannt, Parameter unbekannt

z.B. Exponentialverteilung: unbekannten Parameter $\lambda$ anhand $E(X)$ berechnen

$$E(X) = \frac{1}{\lambda} \quad \Leftrightarrow \quad \lambda = \frac{1}{E(X)}$$ (theoretischer Parameter → durch empirischen ersetzen)

$$\lambda \to \hat{\lambda}, \quad E(X) \to \bar{x} \quad \Rightarrow \hat{\lambda} = \frac{1}{\bar{x}}$$

- **Maximum-Likelihood-Methode**

$n$ Beobachtungen $x_1, x_2, \dots, x_n$ i.i.d., z.B. $X \sim Bin(n, \pi)$ mit $n=100$, $X=58$

gesucht: $\pi$, das möglichst gut zu Beobachtung passt

Idee: Ergebnis $X=58$ gilt als das wahrscheinlichste: $P(X=58) = \binom{100}{58} \pi^{58} (1-\pi)^{42}$

muss maximal werden → nach $\pi$ ableiten und $=0$ setzen (kompliziert!!)

Vereinfachung: beide Seiten Logarithmieren, da Extremum von $\log(f(x))$ auch ein Extremum von $f(x)$ ist!

- **Likelihood-Funktion**

$n$ Beobachtungen $x_1, x_2, \dots, x_n$ gegeben, Parameter $p$ (von $f(x)$) gesucht

$L(p) = f(x_1) \cdot f(x_2) \cdot f(x_3) \cdots f(x_n)$ → Logarithmieren (einfacher!)

- **Logarithmen-Gesetze**

$$y = \log_a(x) \Leftrightarrow a^y = x \qquad a^{\log_a(x)} = x \qquad \log_a(a) = 1 \qquad \log_a(1) = 0$$

$$\log(uv) = \log(u) + \log(v) \qquad \log\left(\frac{u}{v}\right) = \log(u) - \log(v)$$

$$\log(u^r) = r \cdot \log(u) \qquad \log_a(x) = \frac{\ln(x)}{\ln(a)} = \frac{\log_b(x)}{\log_b(a)}$$

## Hypothesentest

Passt Messreihe zu Grösse? Passt Mittelwert der Messreihe zu wahrem Mittelwert?

1. Modell: $X_1, X_2, \dots, X_n$ i.i.d. $\sim \mathcal{N}(\mu, \sigma^2)$
2. Nullhypothese: $H_0: \mu = \mu_0$
3. Alternativhypothese: $H_A: \mu \neq \mu_0$ (beidseitig), $\mu < \mu_0$, $\mu > \mu_0$ (einseitig)
4. Teststatistik unter $H_0$: $$X_n \sim \mathcal{N}\left(\mu, \frac{\sigma^2}{\sqrt{n}}\right)$$
5. Signifikanzniveau: $\alpha = 0.05 \hat{=} 5\%$
6. Verwerfungsbereich: $K = (-\infty, x_{min}] \cup [x_{max}, +\infty)$ (beidseitig) ppf(q=...)
7. Testentscheid: $\bar{X}_n \in K$: $H_0$ wird verworfen! $p \leq \alpha$ p = cdf(x=...)
   $\bar{X}_n \notin K$: $H_0$ wird beibehalten! $p > \alpha$

P-Wert: W'keit, dass unter $H_0$ ein mindestens so extremes Ereignis in Richtung der Alternative beobachtet wird wie das bereits beobachtete.

$\sigma_x$ (Standardabweichung der Population bekannt?

- ja: z-Test (norm.ppf / norm.cdf)
- nein: t-Test (t.ppf / t.cdf) mit $n-1$ Freiheitsgraden

$$T = \frac{\bar{X}_n - \mu_x}{\hat{\sigma}_x / \sqrt{n}} \quad \text{oder} \quad T = \frac{\bar{D}_n - \mu_0}{\hat{\sigma}_D / \sqrt{n}} \quad \text{für Differenzen (gepaarte Stichproben)}$$

Python:

z-Test: norm.cdf(x=[sample mean], loc=[pop. mean], scale=[pop std] / n ** 0.5)

~~(einseitig: zweiseitig: / 2 - Division durch 2)~~

t-Test: t.cdf(x=[sample mean], loc=[pop. mean], scale=[sample std]/$\sqrt{n}$, df=n-1)

oder mit obiger Teststatistik T: t.cdf(x=[sample mean], df=n-1)

oder: tv = ttest_1samp(a=[data], popmean=[$\mu_0$]).statistic

p = t.cdf(tv, df=n-1)

## Vertrauensintervall

Angabe, mit welcher W'keit sich das wahre $\mu$ in welchem Bereich befindet.

$I_{95\%} = [\bar{x}_n - z_{1-\frac{\alpha}{2}} \cdot \sigma_x/\sqrt{n},\ \bar{x}_n + z_{1-\frac{\alpha}{2}} \cdot \sigma_x/\sqrt{n}]$ $\quad \alpha = 0.05$

norm.interval(alpha=0.95, loc=[sample mean], scale=[sample std]/$\sqrt{n}$)

t.interval(... df=n-1)

$\mu$ bekannt: $I = [\mu - z_{\frac{\alpha}{2}} \cdot \sigma,\ \mu + z_{1-\frac{\alpha}{2}} \cdot \sigma]$

## Weitere Testverfahren

| Stichproben | gepaart | ungepaart |
|---|---|---|
| normalverteilt | ttest_rel(x,y) | ttest_ind(x,y, equal_var=False) |
| nicht normalverteilt | Vorzeichentest [1] | mannwhitneyu(x,y) |
| | wilcoxon(d, correction=True) [2] | |

[1] Vorzeichentest: binom_test(x=[Anzahl >0], n=[Anzahl], p=0.5)

[2] Wilcoxon: symmetrische Verteilung vorausgesetzt, i.d.R. mächtigen als t-Test

## Varianzanalyse (ANOVA)

$H_0$: alle Gruppen folgen gleichem Modell, g Gruppen mit n Beobachtungen

$Y_{ij} = \mu + \tau_i + \epsilon_{ij}$, $Y_{ij}$: j-te Beobachtung i-ter Gruppe, $\mu$: gemeinsamen (globalen) Mittelwert,

$\tau_i$ = behandlungsspezifische Abweichung, $\epsilon_{ij}$: Fehlerterm pro Gruppe und Behandlung

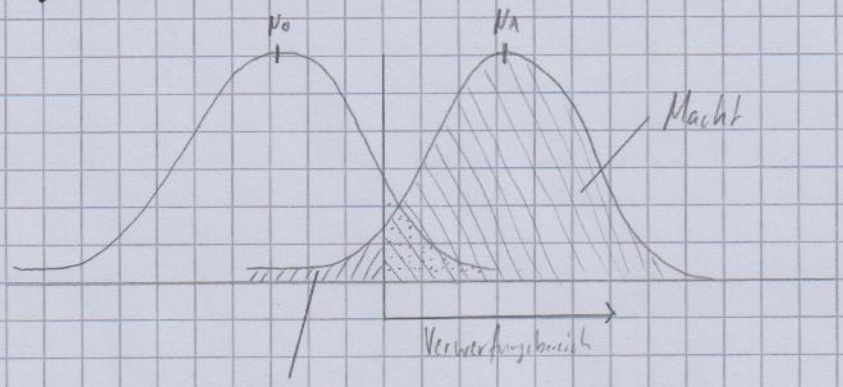
$\mu = \mu_1$ (1. Gruppe als Referenz) → $\tau_1 = 0$, $\tau_2 = \mu_2 - \mu_1$, ...

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm
fit = ols('Zielvariable ~ Faktor(en)', data=df).fit()   # fit.summary(), fit.params
fit_pred = fit.get_prediction()
fit_pred.conf_int()  # 95% - Vertrauensintervall
anova_lm(fit)
```

[1] Zv ~ F1+F2 (additiv) Zv ~ C(F1+F2)

Zv ~ F1*F2 (multiplikativ) Zv ~ C(F1*F2)

# Fehler, Signifikanzniveau & Macht




Fehler 1. Art: $H_0$ fälschlicherweise verworfen $\hat{=}$ Signifikanzniveau $\alpha$

Fehler 2. Art: $H_0$ fälschlicherweise beibehalten $\hat{=}$ 1 - Macht von $\mu_A$

Bootstrap: durch Resampling einer Testreihe Informationen darüber gewinnen

Verfahren: pro Messpunkt eine Kugel in die Urne legen, Ziehen mit Zurücklegen

```
s = np.random.choice(x, n*nboot, replace=True)  # n: Grösse der Probe, nboot: Parameter
s = np.reshape(s, (n, nboot))  # n Spalten, nboot Zeilen
s = s.mean(axis=0)  # spaltenweise
np.sort(s)
np.percentile(s, q=[2.5, 97.5])  # 95%-Vertrauensintervall
```

„echter" Bootstrap:

```
s = np.random.choice(x, n*nboot, replace=True)
table = np.reshape(s, (n, nboot))
col_mean = np.mean(table, axis=0)
delta_mean = col_mean - x.mean()
bounds = np.percentile(delta_mean, q=[2.5, 97.5])
interval = x.mean() - [bounds[1], bounds[0]]
```

## Zeitreihen

Muster: Trend (langfristig), Saisonalität (wiederholend), serielle Korrelation (benachbarte Werte)

- DataFrame: Datumsspalte als Zeitindex:
  `df['date_col'] = pd.DatetimeIndex(df['date_col'])`
  `df.set_index('date_col', inplace=True)`
- Einschränkung auf Zeitfenster Jan 2001 – Jun 2015:
  `df.loc['2001-01':'2015-06']`
- Spaltenweise plotten: `df.plot(subplots=True)`
- Box-Cox-Transformation zur Korrektur von Schiefe & Varianz

$$g(x) = \begin{cases} \frac{x^\lambda - 1}{\lambda} & \text{für } \lambda \neq 0 \\ \log(x) & \text{für } \lambda = 0 \end{cases}$$

```
def boxcox(x, l): #x: Numpy-Array, l: Lambda
    return np.log(x) if (l==0)
    else (x**l - 1)/l
```

↳ Parameter $\lambda$ (l) so wählen, dass Kurve möglichst linear / konstante Varianz!

- Zeitverschiebung (shifting) mit Lag von $k$: $g(x_i) = x_{i-k}$ ($k>0$: rückwärts) !
  `df['col'].shift(4)  # k=-4` ($k>0$: vorwärts)
  Spezialfall $k=1$: backshift $B(x_i) = x_{i-1}$ → Anwendung: für Differenzen
- Boxplot: `df.boxplot('col', by='index_col')`
- Lag-Plot: `from pandas.plotting import lag_plot`  `lag_plot(df['col'])`
- Zerlegung von Zeitreihen: $X_k = m_k + s_k + z_k$ bzw. $X_k = m_k \cdot s_k \cdot z_k$ (additiv / multipl.)
  $m_k$: Trendkomp., $s_k$: saisonaler Effekt, $z_k$: Fehlerkom
- Moving-Average-Filter: Trend unter saisonalem Effekt abschätzen (Fensterbreite wählen)
  `df['trend'] = df['x'].rolling(window=12).mean()`
  `df['seasonal'] = df['x'] - df['trend']`  # $\hat{s}_k = x_k - \hat{m}_k$
- STL: seasonal decomposition of time  `from stldecompose import decompose`
- Vergleich von Datensätzen und. Gr.ordnung: `decompose(np.log(df['x']), period=12).plot()`

```
x = np.log(a.astype('float')) - np.log(a.shift(1).astype('float'))
y =        b                           b
plot(x, y)
```

- Durchschnittliche Saisonalität (Beispiel: 12 Monate, 20 Jahre)

```
df2 = df['seasonal'].values.reshape((12, 20)) # 12 cols w, 20 rows y
avg = np.nanmean(df2, axis=0) # column-wise (along cols)
df['seasonal_avg'] = np.tile(A=avg, reps=12) # original length
residue = df['seasonal'] - df['seasonal_avg']
```

- Dekomposition (keine Ausreisser, saison. konstant)

```
from statsmodels.tsa.seasonal import seasonal_decompose
seasonal_decompose(df['x'], model='additive', freq=12).plot()
seasonal_decompose(np.log(df['x']), model='additive').resid.plot()
```

# Ableitungs- & Integrationsregeln

| | | |
|---|---|---|
| $f(x) = x^n$ | $f'(x) = n \cdot x^{n-1}$ | $F(x) = \frac{1}{n+1} \cdot x^{n+1} + C$ |
| $f(x) = e^x$ | $f'(x) = e^x$ | $F(x) = e^x + C$ |
| $f(x) = a^x$ | $f'(x) = a^x \cdot \ln(a)$ | $F(x) = \frac{1}{\ln(a)} \cdot a^x + C, \quad a > 0, a \neq 1$ |
| $f(x) = \ln(x)$ | $f'(x) = \frac{1}{x}$ | $F(x) = x \cdot \ln(x) - x + C$ |
| $f(x) = \frac{1}{x} = x^{-1}$ | $f'(x) = -\frac{1}{x^2} = -x^{-2}$ | $F(x) = \ln(\lvert x \rvert) + C, \quad x \neq 0$ |