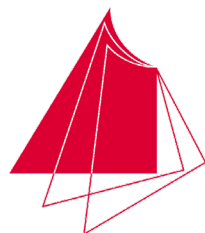


Design For Six Sigma

- MATLAB und Python Befehle -

Manfred Strohrmann
Stefan Günter



Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

Näher dran.

1 Einleitung

Dieses Semester wird ei Vorlesung von MATLAB auf Python umgestellt. Als Hilfestellung finden Sie in diesem Dokument Tabelle mit Befehlen zu den jeweiligen Kapiteln.

Zusätzlich empfehle ich zwei Quellen, die sich für mch als hilfreich erwiesen haben:

Buchempfehlungen



Jake Van der Plas: Data Science mit Python
1. Auflage, mitp Verlags GmbH, Frechen, 2017

Mit diesem Buch habe ich mir selbst Python beigebracht. Es behandelt die wesentlichen Operatoren und führt gut in Numpy, Pandas und Scikit-Learn ein.

Links im Internet

<https://hyperpolyglot.org/numerical-analysis2>

Hier werden die wesentlichen Befehle zur Numerik und Statistik für verschiedene Programmiersprachen aufgeführt. Damit wird der Umstieg von einer Programmiersprache auf eine andere stark vereinfacht.

2 Grundlagen der Wahrscheinlichkeitstheorie

Befehle zur Berechnung in MATLAB

Zur Berechnung der Anzahl von Permutationen, Kombinationen und Variationen werden im wesentlichen drei MATLAB-Befehle verwendet:

Tabelle 2.1: MATLAB-Befehle zur Berechnung von Permutationen, Kombinationen und Variationen

Fakultät	$M = N!$	<code>factorial(N)</code>
Potenz	$M = N^K$	<code>N^K</code>
Binomialkoeffizient	$M = \binom{N}{K} = \frac{N!}{K!(N-K)!}$	<code>nchoosek(N,K)</code>

Befehle zur Berechnung in Python

Zur Berechnung der Anzahl von Permutationen, Kombinationen und Variationen werden im wesentlichen drei Python-Befehle verwendet:

Tabelle 2.2: Python-Befehle zur Berechnung von Permutationen, Kombinationen und Variationen

Fakultät	$M = N!$	<code>scipy.special.factorial</code>
Potenz	$M = N^K$	<code>scipy.special.factorial</code>
Binomialkoeffizient	$M = \binom{N}{K} = \frac{N!}{K!(N-K)!}$	<code>scipy.special.comb</code>

3 Beschreibende Statistik univariater Daten

3.1 Häufigkeitsverteilungen

Befehle zur Beschreibung von Häufigkeiten in MATLAB

Zur Berechnung und Darstellung von Häufigkeitsverteilungen stehen in MATLAB diverse Funktionen zur Verfügung. Die wichtigsten sind in Tabelle 3.5 zusammengefasst.

Tabelle 3.1: Berechnung und Darstellung von Häufigkeitsverteilungen in MATLAB

MATLAB Befehl	Funktionsbeschreibung
sort(x)	Sortiert die Werte des Vektors x nach der Größe
cumsum(x)	Berechnet die kumulative Summe des Vektors x
hist(x)	Erzeugt ein Histogramm mit der absoluten Häufigkeit von x
tabulate(x)	Erstellt eine Häufigkeitstabelle aus x
bar(x)	Erzeugt ein Balkendiagramm aus x mit vertikaler Ausrichtung
barh(x)	Erzeugt ein Balkendiagramm aus X mit horizontaler Ausrichtung
pie(x)	Erzeugt ein Kreisdiagramm aus x

Befehle zur Beschreibung von Häufigkeiten in Python

Zur Berechnung und Darstellung von Häufigkeitsverteilungen stehen in Python diverse Funktionen zur Verfügung. Die wichtigsten sind in Tabelle 3.6 zusammengefasst.

Tabelle 3.2: Berechnung und Darstellung von Häufigkeitsverteilungen in Python

Python Befehl	Funktionsbeschreibung
sorted	Sortiert die Werte des Vektors x nach der Größe
numpy.cumsum	Berechnet die kumulative Summe des Vektors x
numpy.histogram	Erzeugt ein Histogramm mit der absoluten Häufigkeit von x
count	Erstellt eine Häufigkeitstabelle aus x
matplotlib.pyplot.bar	Erzeugt ein Balkendiagramm aus x mit vertikaler Ausrichtung
matplotlib.pyplot.barh	Erzeugt ein Balkendiagramm aus X mit horizontaler Ausrichtung
matplotlib.pyplot.pie	Erzeugt ein Kreisdiagramm aus x

3.2 Kennwerte einer Stichprobe

Zusammenfassung der Lagekennwerte von Stichproben

Zur besseren Übersicht fasst Tabelle 3.9 die Lagekennwerte einer Stichprobe zusammen.

Tabelle 3.3: Lagekennwerte einer Stichprobe

Lagekennwert	Definition	Bemerkungen
Mittelwert	$\bar{x} = \frac{x_1 + \dots + x_N}{N} = \frac{1}{N} \cdot \sum_{n=1}^N x_n$	Empfindlich gegenüber Ausreißern
Mittelwert von Daten in Klassen	$\bar{x} = \sum_{n=1}^N (x_n \cdot h(c_n))$	
Median	$H(x_{\text{MED}}) = 0.5$	Weniger empfindlich gegenüber Ausreißern als der Mittelwert
Median von Daten in Klassen	$x_{\text{MED}} = c_{n-1} + \frac{d \cdot (0.5 - H(c_{n-1}))}{h(c_n)}$	
Geometrisches Mittel	$x_G = \sqrt[N]{x_1 \cdot \dots \cdot x_N}$	
Modus	häufigster Wert einer Stichprobe	

Da zur Berechnung der Lagekennwerte meist entsprechende Software verwendet wird, entfällt in vielen Fällen eine Einteilung der Daten in Klassen. MATLAB bietet einige Funktionen, mit denen die Lagekennwerte von Stichproben bestimmt werden können. Sie sind in Tabelle 3.10 aufgelistet.

Tabelle 3.4: Berechnung der Lagekennwerte von Stichproben in MATLAB

Lagekennwert	MATLAB-Befehl
Mittelwert	mean(X)
Median	median(X)
Geometrisches Mittel	geomean(X)
Modus	mode(X)

Vergleichbare Befehle existieren in Python, sie sind in Tabelle 3.11 zusammengestellt.

Tabelle 3.5: Berechnung der Lagekennwerte von Stichproben in Python

Lagekennwert	Python-Befehl
Mittelwert	numpy.median
Median	numpy.mean
Geometrisches Mittel	scipy.stats.mstats.gmean
Modus	scipy.stats.mode

Zusammenfassung der Streuungskennwerte von Stichproben

Zur besseren Übersicht fasst Tabelle 3.12 die Streuungskennwerte einer Stichprobe zusammen.

Tabelle 3.6: Streuungskennwerte einer Stichprobe

Streuungskennwert	Definition	Bemerkungen
Spannweite	$\Delta x = x_{\text{MAX}} - x_{\text{MIN}}$	Extrem empfindlich gegenüber Ausreißern
Varianz	$s^2 = \frac{1}{N-1} \cdot \sum_{n=1}^N (x_n - \bar{x})^2$	Weniger empfindlich gegenüber Ausreißern
Varianz in Klassen eingeteilter Daten	$s^2 = \frac{1}{N-1} \cdot \sum_{n=1}^N (h_A(c_n) \cdot (c_n - \bar{x})^2)$	
Standardabweichung	$s = \sqrt{\frac{1}{N-1} \cdot \sum_{n=1}^N (x_n - \bar{x})^2}$	Vergleichbar zur Varianz, aber in Einheiten der Zielgröße x
Standardabweichung in Klassen eingeteilter Daten	$s = \sqrt{\frac{1}{N-1} \cdot \sum_{n=1}^N (h_A(c_n) \cdot (c_n - \bar{x})^2)}$	
P-Quantil	$H(x_p) = P$	
P-Quantil in Klassen eingeteilter Daten	$x_p = c_{n-1} + \frac{d \cdot (P - H(c_{n-1}))}{h(c_n)}$	
Interquartilabstand	$IQR = x_{0.75} - x_{0.25}$	Robuster Streuungskennwert

Gerade bei größeren Stichprobenumfängen ist es aufwendig, die Streuungskennwerte manuell zu berechnen. Daher wird zur Auswertung meist entsprechende Software herangezogen. Die Befehle zur Berechnung der Streuungskenngrößen mit MATLAB sind in Tabelle 3.13 zusammengefasst.

Tabelle 3.7: Berechnung der Streuungskennwerte von Stichproben in MATLAB

Streuungskennwert	MATLAB-Befehl
Spannweite	range(x) oder max(x)-min(x)
Varianz	var(x)
Standardabweichung	std(x)
p-Quantil	quantile(x,p)
Interquartilabstand	iqr(x)

Vergleichbare Befehle existieren in Python, sie sind in Tabelle 3.14 zusammengestellt.

Tabelle 3.8: Berechnung der Streuungskennwerte von Stichproben in Python

Streuungskennwert	Python-Befehl
Spannweite	max - min
Varianz	numpy.var
Standardabweichung	numpy.std
p-Quantil	numpy.quantile
Interquartilabstand	numpy.quantile

Schiefe oder Symmetrie einer Stichprobe

Der Befehl zur Berechnung der Schiefe in MATLAB ist in Tabelle 3.16 aufgeführt.

Tabelle 3.9: Berechnung der Symmetriekennwerte von Stichproben in MATLAB

Symmetriekennwert	MATLAB-Befehl
Momentenkoeffizient der Schiefe	skewness(x)

Tabelle 3.17 zeigt den Befehl zur Berechnung der Schiefe in Python.

Tabelle 3.10: Berechnung der Symmetriekennwerte von Stichproben in MATLAB

Symmetriekennwert	MATLAB-Befehl
Momentenkoeffizient der Schiefe	scipy.stats.skew

Boxplot

MATLAB und Python bieten zur Erstellung eines Box-Plots eine separate Funktion an.

Tabelle 3.11: Darstellung des Box-Plots einer Stichprobe in MATLAB

Darstellung	MATLAB-Befehl
Box-Plot	boxplot(x)

Tabelle 3.12: Darstellung des Box-Plots einer Stichprobe in Python

Darstellung	MATLAB-Befehl
Box-Plot	matplotlib.pyplot.boxplot

4 Univariate Wahrscheinlichkeitstheorie

4.1 Spezielle diskrete Verteilungen

Tabelle 4.1: Übersicht über diskrete Wahrscheinlichkeitsverteilungen

Name und Anwendung	Wahrscheinlichkeitsverteilung	Kenngößen μ und σ^2
Gleichverteilung: Gleiche Wahrscheinlichkeit für alle Ereignisse	$f(x) = p = \frac{1}{N}$	$\mu = \frac{1}{N} \cdot \sum_{n=1}^N x_n$ $\sigma^2 = \frac{1}{N} \cdot \sum_{n=1}^N (x_n - \mu)^2$
Bernoulli-Verteilung: Günstiges / ungünstiges Ereignis bei einfacher Ausführung des Experimentes	$f(x) = \begin{cases} p & \text{für } x = 1 \\ q = 1 - p & \text{für } x = 0 \end{cases}$	$\mu = p$ $\sigma^2 = p \cdot q$
Binomial-Verteilung: Anzahl günstiger Ereignisse bei N-facher Ausführung, konstante Erfolgswahrscheinlichkeit	$f(x) = \binom{N}{x} \cdot p^x \cdot q^{N-x}$	$\mu = N \cdot p$ $\sigma^2 = N \cdot p \cdot q$
Hypergeometrische Verteilung: Anzahl günstiger Ereignisse bei N-facher Ausführung, variable Erfolgswahrscheinlichkeit	$f(x) = \frac{\binom{G}{x} \cdot \binom{M-G}{G-x}}{\binom{M}{N}}$	$\mu = N \cdot \frac{G}{M}$ $\sigma^2 = \frac{N \cdot G \cdot (M-G) \cdot (M-N)}{M^2 \cdot (M-1)}$
Poisson-Verteilung: Approximation der Binomialverteilung	$f(x) = \frac{\mu^x}{x!} \cdot e^{-\mu}$	$\mu = N \cdot p$ $\sigma^2 = N \cdot p = \mu$
Geometrische Verteilung: Anzahl von Wiederholungen des Experimentes, bis das günstige Ereignis eintritt	$f(x) = p \cdot q^{x-1}$	$\mu = \frac{1}{p}$ $\sigma^2 = \frac{1-p}{p^2}$

Sowohl die Wahrscheinlichkeitsverteilung als auch die Verteilungsfunktion der in Tabelle 4.11 aufgelisteten Verteilungen sind bei MATLAB in der Statistic Toolbox implementiert. Dabei werden die folgenden Endungen für die MATLAB-Funktionen eingesetzt:

- pdf Wahrscheinlichkeitsverteilung $f(x)$ (probability density function)
- cdf Verteilungsfunktion $F(x)$ (cumulative distribution function)
- inv Inverse Verteilungsfunktion $F^{-1}(x)$ (inverse cumulative distribution function)
- rnd Zufallszahlen-Generator einer Verteilung

Tabelle 4.12 gibt eine Übersicht über eine Auswahl von MATLAB-Funktionen zu diskreten Verteilungen.

Tabelle 4.2: Übersicht über diskrete Wahrscheinlichkeitsverteilungen in MATLAB

Verteilung	Wahrscheinlichkeitsverteilung $f(x)$	Verteilungsfunktion $F(x)$	inverse Verteilungsfunktion $F^{-1}(x)$	Zufallszahlen-generator
Gleichverteilung	unidpdf(x,N)	unidcdf(x,N)	unidinv(P,N)	unidrnd(N,m,n)
Binomial-Verteilung	binopdf(x,N,p)	binocdf(x,N,p)	binoinv (Y,N,p)	binornd(N,p,m,n)
Hypergeometrische Verteilung	hygepdf(x,M,G,N)	hygecdf(x,M,G,N)	hygeinv(P,M,G,N)	hygernd(M,G,N,m,n)
Poisson-Verteilung	poisspdf(x, μ)	poisscdf(x, μ)	poissinv(P, μ)	poissrnd(μ ,m,n)
Geometrische Verteilung	geopdf(x,p)	geocdf(x,p)	geoinv(P,p)	geornd(p,m,n)

Tabelle 4.13 gibt eine Übersicht über eine Auswahl von Python-Funktionen der Bibliothek scipy.stats zu diskreten Verteilungen.

Tabelle 4.3: Übersicht über diskrete Wahrscheinlichkeitsverteilungen der Python Bibliothek scipy.stats

Verteilung	Wahrscheinlichkeitsverteilung $f(x)$	Verteilungsfunktion $F(x)$	inverse Verteilungsfunktion $F^{-1}(x)$	Zufallszahlen-generator
Gleichverteilung	randint.pmf	randint.cdf	randint.ppf	randint.rvs
Binomial-Verteilung	binom.pmf	binom.cdf	binom.ppf	binom.rvs
Hypergeometrische Verteilung	hypergeom.pmf	hypergeom.cdf	hypergeom.ppf	hypergeom.rvs
Poisson-Verteilung	poisson.pmf	poisson.cdf	poisson.ppf	poisson.rvs
Geometrische Verteilung	geom.pmf	geom.cdf	geom.ppf	geom.rvs

4.2 Spezielle stetige Verteilungen

Tabelle 4.4: Übersicht über stetige Wahrscheinlichkeitsverteilungen und ihre Anwendungen (Teil1)

Name und Anwendung	Definition der Wahrscheinlichkeitsdichte	Kenngößen μ und σ
<p>Gleichverteilung:</p> <p>Beschreibung von Wartezeiten und Diskretisierungsvorgängen</p>	$f(x) = \frac{1}{b-a}$	$\mu = \frac{a+b}{2}$ $\sigma^2 = \frac{(b-a)^2}{12}$
<p>Symmetrische Dreiecksverteilung:</p> <p>Toleranzverteilung bei Fertigungsprozessen</p>	$f(x) = \begin{cases} \frac{4}{(b-a)^2} \cdot (x-a) & \text{für } a < x < \mu \\ \frac{4}{(b-a)^2} \cdot (b-x) & \text{für } \mu < x < b \end{cases}$	$\mu = \frac{a+b}{2}$ $\sigma^2 = \frac{(b-a)^2}{24}$
<p>Weibull-Verteilung:</p> <p>Lebensdauer von Produkten, Zeiträumen bis zum Schadensfall, Ausfallwahrscheinlichkeit ändert sich über der Beobachtungszeit</p>	$f(x) = \frac{\beta}{\eta} \cdot \left(\frac{x}{\eta}\right)^{\beta-1} \cdot e^{-\left(\frac{x}{\eta}\right)^\beta}$	<p>siehe Abschnitt 4.6.3</p>
<p>Exponential-Verteilung:</p> <p>Lebensdauer von Produkten, Zeiträumen bis zum Schadensfall, Ausfallwahrscheinlichkeit ändert sich nicht über der Beobachtungszeit</p>	$f(x) = \lambda \cdot e^{-\lambda \cdot x}$	$\mu = \frac{1}{\lambda}$ $\sigma^2 = \frac{1}{\lambda^2}$
<p>Rayleigh-Verteilung:</p> <p>Rechtsschiefe Verteilung zur Beschreibung des Betrages zweier normalverteilter Zufallsgrößen</p>	$f(x) = \frac{1}{b^2} \cdot x \cdot e^{-\frac{1}{2} \frac{x^2}{b^2}}$	$\mu = b \cdot \sqrt{\frac{\pi}{2}}$ $\sigma^2 = \frac{4-\pi}{2} \cdot b^2$
<p>Normalverteilung:</p> <p>Approximation von Zufallsprozessen, insbesondere bei der Messwert-verarbeitung und bei der Prozessregelung</p>	$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2}$	μ σ^2
<p>Logarithmische Normalverteilung:</p> <p>rechtsschiefe Verteilungen wie Lebensdauern, Wartezeiten oder Einkommen</p>	$f(y) = \frac{1}{\sigma_x \cdot \sqrt{2 \cdot \pi}} \cdot \frac{1}{y} \cdot e^{-\frac{1}{2} \left(\frac{\ln(y) - \mu_x}{\sigma_x}\right)^2}$	$\mu_Y = e^{\mu_x + \frac{\sigma_x^2}{2}}$ $\sigma_Y^2 = e^{2\mu_x + \sigma_x^2} \cdot (e^{\sigma_x^2} - 1)$
<p>Betragsverteilung 1. Art:</p> <p>Verteilungsfunktion des Betrages der Abweichungen um einen Sollwert</p>	$f(x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \left(e^{-\frac{1}{2} \left(\frac{x-\mu}{\sigma}\right)^2} + e^{-\frac{1}{2} \left(\frac{x+\mu}{\sigma}\right)^2} \right)$	

Tabelle 4.5: Übersicht über stetige Wahrscheinlichkeitsverteilungen in MATLAB

Verteilung	Wahrscheinlichkeitsverteilung $f(x)$	Verteilungsfunktion $F(x)$	inverse Verteilungsfunktion $F^{-1}(x)$	Zufallszahlen-generator
Gleichverteilung	<code>unifpdf(x,a,b)</code>	<code>unidcdf(x,a,b)</code>	<code>unidinv(P,a,b)</code>	<code>unifrnd(a,b)</code>
Weibull-Verteilung	<code>wblpdf(x,η,β)</code>	<code>wblcdf(x,η,β)</code>	<code>wblinv(P,η,β)</code>	<code>wblrnd(η,β)</code>
Exponentialverteilung	<code>exppdf(x,μ)</code>	<code>expcdf(x,μ)</code>	<code>expinv(P,μ)</code>	<code>expmnd(μ)</code>
Rayleigh-Verteilung	<code>raylpdf(x,b)</code>	<code>raylcdf(x,b)</code>	<code>raylinv(P,b)</code>	<code>raylrnd(b)</code>
Normalverteilung	<code>normpdf(x,μ,σ)</code>	<code>normcdf(x,μ,σ)</code>	<code>norminv(P,μ,σ)</code>	<code>normrnd(μ,σ)</code>
Logarithmische Normalverteilung	<code>lognpdf(x,μ,σ)</code>	<code>logncdf(x,μ,σ)</code>	<code>logninv(P,μ,σ)</code>	<code>lognrnd(μ,σ)</code>

Vergleichbare Python-Befehle bietet `scipy.stats`, sie sind in Tabelle 4.17 zusammengestellt.

Tabelle 4.6: Übersicht über stetige Wahrscheinlichkeitsverteilungen der Python Bibliothek `scipy.stats`

Verteilung	Wahrscheinlichkeitsverteilung $f(x)$	Verteilungsfunktion $F(x)$	inverse Verteilungsfunktion $F^{-1}(x)$	Zufallszahlen-generator
Gleichverteilung	<code>uniform.pdf(x,a,b)</code>	<code>uniform.cdf(x,a,b)</code>	<code>uniform.ppf(P,a,b)</code>	<code>uniform.rvs(a,b)</code>
Weibull-Verteilung	<code>weibull_min.pdf(x,η,β)</code>	<code>weibul_min.cdf(x,η,β)</code>	<code>weibul_min.ppf(P,η,β)</code>	<code>weibul_min.rvs(η,β)</code>
Exponentialverteilung	<code>expon.pdf(x,μ)</code>	<code>expon.cdf(x,μ)</code>	<code>expon.ppf(P,μ)</code>	<code>expon.rvs(μ)</code>
Rayleigh-Verteilung	<code>raylpdf(x,b)</code>	<code>raylcdf(x,b)</code>	<code>raylinv(P,b)</code>	<code>raylrnd(b)</code>
Normalverteilung	<code>norm.pdf(x,μ,σ)</code>	<code>norm.cdf(x,μ,σ)</code>	<code>norm.ppf(P,μ,σ)</code>	<code>norm.rvs(μ,σ)</code>
Logarithmische Normalverteilung	<code>lognorm.pdf(x,μ,σ)</code>	<code>lognorm.cdf(x,μ,σ)</code>	<code>lognorm.ppf(P,μ,σ)</code>	<code>lognorm.rvs(μ,σ)</code>

4.3 Prüf- oder Testverteilungen

Tabelle 4.7: Übersicht über die Testverteilungen und ihre Anwendungen

Name und Anwendung	Definition der Wahrscheinlichkeitsdichte	Kenngößen μ und σ
t-Verteilung mit v Freiheitsgraden Statistische Bewertung von Mittelwerten auf Basis von Stichproben bei unbekannter Varianz	$f(t) = \frac{\Gamma\left(\frac{v+1}{2}\right)}{\sqrt{v \cdot \pi} \cdot \Gamma\left(\frac{v}{2}\right) \cdot \left(1 + \frac{t^2}{v}\right)^{\frac{v+1}{2}}}$	$\mu = 0$ $\sigma^2 = \frac{v}{v-2}$
Chi ² -Verteilung mit v Freiheitsgraden Statistische Bewertung der Varianz einer Grundgesamtheit auf Basis von Stichproben	$f(\chi) = K_v \cdot \chi^{\frac{v-2}{2}} \cdot e^{-\frac{\chi}{2}}$	$\mu = v$ $\sigma^2 = 2 \cdot v$
f-Verteilung mit v_1, v_2 Freiheitsgraden Statistische Vergleich der Varianz zweier Grundgesamtheiten auf Basis von Stichproben	$f(f) = \frac{\Gamma\left(\frac{v_1+v_2}{2}\right)}{\Gamma\left(\frac{v_1}{2}\right) \cdot \Gamma\left(\frac{v_2}{2}\right)} \cdot \left(\frac{v_1}{v_2}\right)^{\frac{v_1}{2}} \cdot \frac{f^{\frac{v_1}{2}-1}}{\left(\frac{v_1}{v_2} \cdot f + 1\right)^{\frac{v_1+v_2}{2}}}$	$\mu = \frac{v_2}{v_2-2}$ $\sigma^2 = \frac{2 \cdot v_2^2 \cdot (v_2 + v_1 - 2)}{v_1 \cdot (v_2 - 4) \cdot (v_2 - 2)^2}$

In der Statistic-Toolbox von MATLAB sind die Prüf- und Testverteilungen implementiert. Tabelle 4.19 gibt einen Überblick über die entsprechenden Funktionen.

Tabelle 4.8: Übersicht über die Testverteilungen in MATLAB

Verteilung	Dichtefunktion $f(x)$	Wahrscheinlichkeitsfunktion $F(x)$	inverse Wahrscheinlichkeitsfunktion $F^{-1}(x)$
t-Verteilung	tpdf(x, v)	tcdf(x, v)	tinv(P, v)
Chi-Quadrat-Verteilung	chi2pdf(x, v)	chi2cdf(x, v)	chi2inv(P, v)
f-Verteilung	fpdf(x, v1, v2)	fcdf(x, v1, v2)	finv(P, v1, v2)

Auch in Python sind entsprechende Funktionen implementiert, Tabelle 4.20 gibt einen Überblick über diese Funktionen.

Tabelle 4.9: Übersicht über die Testverteilungen in der Python Bibliothek scipy.stats

Verteilung	Dichtefunktion $f(x)$	Wahrscheinlichkeits- funktion $F(x)$	inverse Wahrscheinlich- keitsfunktion $F^{-1}(x)$
t-Verteilung	<code>t.pdf(x, v)</code>	<code>t.cdf(x, v)</code>	<code>t.ppf(P, v)</code>
Chi-Quadrat-Ver- teilung	<code>chi2.pdf(x, v)</code>	<code>chi2.cdf(x, v)</code>	<code>chi2.ppf(P, v)</code>
f-Verteilung	<code>f.pdf(x, v1, v2)</code>	<code>f.cdf(x, v1, v2)</code>	<code>f.ppf(P, v1, v2)</code>