

Virtuelle Tabellensammlung

Peter Naeve

`pnaeve@wiwi.uni-bielefeld.de`

September 2001
Diskussionspapier Nr. 474

Universität Bielefeld
Fakultät für Wirtschaftswissenschaften
Postfach 100131
33501 Bielefeld

1 Vorbemerkungen

1.1 Reminiszenzen

Eine meiner ersten Tätigkeiten am Lehrstuhl für Statistik von Professor Wetzel an der Universität Kiel war die technische Betreuung einer Statistischen Tabellensammlung für Studenten. Sie sollte preiswert sein, daher wurden alle Tabellen am Lehrstuhl auf der X1 (wer weiß, daß Dijkstra zu ihren Vätern gehört?) des Rechenzentrums errechnet und ausgedruckt. Die technische Qualität des Ausdrucks — das Drucktuch des Zeilendruckers war links immer stärker abgenutzt als rechts — ließ den Drucker für den angestrebten Photodruck Schlimmes befürchten, daher meine Tätigkeit als eine Art TQM-Beauftragter in der Druckerei. Das Ergebnis hat alle befriedigt. Ich habe viel über die Kunst des Druckens gelernt.

Dank eines Rufes an die FU Berlin bewegte sich das ganze Team in das neu gegründete Institut für Angewandte Statistik. Das Gefühl, etwas für den leichteren Zugang der statistischen Community zu statistischen Tabellen tun zu sollen, wurde in Berlin immer stärker.

Eine handliche Sammlung der wichtigsten statistischen Tabellen, die für die verschiedensten Anwendungsgebiete Verwendung finden könnte, fehlt bis jetzt auf dem deutschen Büchermarkt. Es scheint deshalb angebracht, diese Lücke durch eine wohlfeile Ausgabe zu schließen. Bei der Auswahl der Tabellen sind die Verfasser davon ausgegangen, möglichst vielseitig verwendbare Tafeln in die vorliegende Sammlung aufzunehmen. Die meisten Tabellen sind neu berechnet. Soweit der Rechenaufwand vertretbar war, wurde die Gelegenheit benutzt, bei vielen Tafeln auch Tabellenwerte für Parameterkonstellationen und Wahrscheinlichkeiten aufzunehmen, die in der Literatur nicht verfügbar sind.

Dieser Ausschnitt aus dem Vorwort von

Statistische Tabellen
Wolfgang Wetzel, Max-Detlev Jöhnk, Peter Naeye
Verlag Walter de Gruyter & Co.
Berlin, 1967

zeigt, daß wir dem Gefühl nachgegeben haben. Sicher hat dazu auch die legendäre IBM 1130 beigetragen, die in unserem Institut stand. Meine Rolle hatte sich erweitert. Die angesprochenen Neuberechnungen wurden von mir¹ durchgeführt.

1.2 Lokal virtuell

Die Zeit ist (leider) über die IBM 1130 hinweggegangen. Der Student vor seinem heimischen PC würde ob ihrer Rechenleistung nur müde den Kopf hebe. Auch auf dem Gebiet der Software hat sich vieles weiter entwickelt. Reichhaltige Angebote an statistischer Software machen ein anstrengendes Programmieren — ich hatte damals unzählige FORTRAN

¹Die Fehler – es gibt welche – gehen selbstverständlich auf meine Kappe

Anweisungen geschrieben — unnötig und eigentlich auch ein Hantieren mit Tabellensammlungen, denn die statistischen Systeme enthalten leicht (?) verwendbar ein umfangreiches Angebot an nützlichen Hilfsmitteln. Hat man beispielsweise die Verteilung xyz im Fokus, so bietet S-Plus mit

dxyz	Dichte (kontinuierlich), Wahrscheinlichkeitsfunktion (diskret)
pxyz	Verteilungsfunktion
qxyz	Inverse der Verteilungsfunktion
rxyz	Zufallszahlen gemäß Verteilung xyz

Wunderbar, damit ist doch ein für allemal auch das in dem zitierten Vorwort angesprochene Problem *“Soweit der Rechenaufwand vertretbar war, wurde die Gelegenheit benutzt, bei vielen Tafeln auch Tabellenwerte für Parameterkonstellationen und Wahrscheinlichkeiten aufzunehmen, die in der Literatur nicht verfügbar sind”* gelöst. Zumindest theoretisch.

Es zeigt sich nämlich, daß für viele Studenten der Sprung über die Tabellensammlung hinweg direkt an die Basis (in die statistische Software) doch eine zu hohe Anforderung ist. Mir scheint, ihnen könnte geholfen werden, wenn ihnen etwas an die Hand gegeben würde, mit dem sie umgehen können wie mit einer der traditionellen Tabellensammlungen, das aber keine Papiererscheinung hat, eben eine virtuelle Tabellensammlung.

Die hier vorgelegte virtuelle Tabellensammlung² ersetzt das Nachschlagen in papierernen Tabellensammlungen für die Hörer der Veranstaltung Statistik II. Es wird auf die in der statistischen Umgebung S-Plus implementierten Verteilungen zurückgegriffen. Die Vermittlungsarbeit wird von dem im Bereich *Statistik und Informatik* entwickelten “Browser” REVIVE geleistet³.

1.3 Das Angebot

Ehe der Leser seiner Neigung nachgibt, nach der Lektüre des Angebots zu nörgeln, da fehlt ja noch dies und das — natürlich fehlt noch dies und das —, möge er sich daran erinnern, daß er eine Tabellensammlung für eine Statistik II-Vorlesung an einer Fakultät für Wirtschaftswissenschaften in der Hand hat. (Geht das überhaupt mit einer virtuellen Tabellensammlung?) hat. Gemessen daran ist das Angebot doch wohl nicht schlecht.

²© Peter Naeve Statistik und Informatik, Universität Bielefeld, September 2001

³Neugierige seien auf die im Literaturverzeichnis angegebenen Arbeiten von Peter Wolf verwiesen.

Verteilungen

Normalverteilung	Seite 7
Exponentialverteilung	Seite 9
t-Verteilung	Seite 11
F-Verteilung	Seite 13
χ^2 -Verteilung	Seite 15
Binomialverteilung	Seite 17
Geometrische Verteilung	Seite 19
Poisson-Verteilung	Seite 21

Konfidenzintervalle

Normalverteilung: σ bekannt	Seite 24
Normalverteilung; Stichprobenumfang	Seite 24
Normalverteilung: σ unbekannt	Seite 25
Einfaches Konfidenzintervall: Bestimme $1 - \alpha$	Seite 26
Einfaches Konfidenzintervall: Bestimme c	Seite 26

1.4 Gebrauch

Das Kommando⁴

```
open.material("virtab")
```

in S-Plus gegeben, öffnet die virtuelle Tabellensammlung. Der Benutzer sieht sich folgenden Zeilen⁵ gegenüber.

```
+-----+
|
|           revive -- version 2.631
|
+-----+
:evaluate section 0
```

Virtuelle Tabellensammlung

```
revive:... 0 -:
```

Der Benutzer antwortet auf den Prompt -: mit der Eingabe s xx, wobei s xx die Angabe in der Zeile

Aktivierung: s xx

seiner ausgewählten Tabellenanfrage steht.

Durch Aktivieren des jeweiligen Code-Chunks wird die betreffende statistische Handlung vollzogen. Der Benutzer wird nach den benötigten Angaben (x, w, \dots) gefragt. Die Eingabe

⁴Stand SS 00 für die Vorlesung Statistik II

⁵Dies ist der augenblickliche – September 2001 – Stand, der Browser wird weiterentwickelt, es kann sich also das Erscheinungsbild ändern.

wird überprüft, um sicher zu sein, daß der Benutzer keine unzulässigen Werte (z.B. $\sigma < 0$) eingibt.

Beispiel:

Hat der Benutzer z.B. die Eingabe `s 2` gemacht, d.h., er will die Parameter der Normalverteilung setzen, so könnte der Verlauf des Dialoges wie folgt aussehen.

```
Normalverteilung: Parametersetzung
```

```
Setze mu fest:
```

```
[1] 0
```

```
Setze sigma fest:
```

```
[1] -2
```

```
sigma muss > 0 sein, neuer Versuch!
```

```
[1] 1
```

```
Normalverteilung: mu = 0, sigma = 1
```

1.5 Los geht's

Genau genommen, es ist schon losgegangen. Damit nach dem oben angegebenen Kommando überhaupt etwas Sinnvolles vom Benutzer gemacht werden kann, ist die Tabellensammlung "aufzuschlagen". Dies wird mit Hilfe des hier angegebenen Code-Chunks automatisch erledigt.

```
1 <start 1>≡
  cat("\nVirtuelle Tabellensammlung\n\n")
```

This definition is continued in chunks 86 and 94–102.

Root chunk (not used in this document).

1.6 Global virtuell

Die in den vorigen Abschnitten vorgestellte virtuelle Tabellensammlung hat immer noch den Nachteil, daß es sich nur um eine lokale Form handelt. Wer nicht einmal Hörer der Veranstaltung Statistik II war, hat keinen Zugang zu dieser Tabellensammlung. Damit fallen wir aber weit hinter die alten papierenen Tabellensammlungen zurück, denen ja durch das Einstellen in eine Bibliothek eine allgemeine Zugänglichkeit eröffnet wurde.

Angeblich ist das Internet die Bibliothek der Zukunft. Einer virtuellen Tabellensammlung, die im Internet existieren würde, käme ein ähnlich globaler Status zu wie ihren papierenen Vorläufern in der Bibliothek. Wie eine gerade abgelieferte Diplomarbeit von Thomas Nolte zeigt, ist es technisch möglich, eine solche globale virtuelle Tabellensammlung zu implementieren. Packen wir es an. (Nicht hier, später aber sicher.)

2 Verteilungen

2.1 Aufbau

Für jede Verteilung werden die nachstehenden Möglichkeiten in der angegebenen Reihenfolge angeboten

1. Veränderung der Parametersetzungen, alle Verteilungen haben eine Defaultsetzung, die jeweils im entsprechenden Abschnitt angegeben ist. Die letzte Parametersetzung wird bis zur erneuten Veränderung beibehalten. Die erfolgte Parametersetzung wird durch Wiederholung der eingelesenen Werte bestätigt.
2. Auswertung der Verteilungsfunktion an der Stelle x — x wird vom Benutzer erfragt. Ein Formelsteckbrief der Verteilungsfunktion wird gegeben.
3. Bestimmung des Prozentpunktes x_w — w wird vom Benutzer erfragt.
4. Auswertung der Dichte bzw. der Wahrscheinlichkeitsfunktion an der Stelle x — x wird vom Benutzer erfragt. Ein Formelsteckbrief der Dichte bzw. Wahrscheinlichkeitsfunktion wird gegeben.

Die errechneten Werte für Verteilungsfunktion, Prozentpunkt und Dichte werden jeweils zusammen mit der gültigen Parametersetzung ausgegeben.

2.2 Normalverteilung

2.2.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
μ	0	mu
σ	1	sigma

Die Defaultsetzung erfolgt in Code-Chunk 95.

2.2.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (μ, σ) für die Normalverteilung verändert.

Aktivierung: s 2

```
2 <* 2>≡  
  <Normalverteilung: setze Parameter 49>
```

This definition is continued in chunks 3–39.

Root chunk (not used in this document).

2.2.3 Verteilungsfunktion

Steckbrief:

$$F(x; \mu, \sigma) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(u-\mu)^2}{2\sigma^2}} du \quad -\infty < x < +\infty; \quad -\infty < \mu < +\infty, \quad \sigma > 0$$

Aktivierung: s 3

Beispiel:

Normalverteilung: Verteilungsfunktion

gebe x an:

```
[1] 0
```

Normalverteilung: F(0;0,1) = 0.5

```
3 <* 2>+≡  
  <Normalverteilung: berechne Verteilungsfunktion 50>
```

2.2.4 Prozentpunkt

Aktivierung: s 4

Beispiel:

Normalverteilung: Prozentpunkt

gebe w an:

[1] 0.5

Normalverteilung: x.w = 0 mit $F(x.w;0,1) = 0.5$

4 <* 2>+≡

<Normalverteilung: bestimme Prozentpunkt 51>

2.2.5 Dichte

Steckbrief:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad -\infty < x < +\infty; \quad -\infty < \mu < +\infty, \quad \sigma > 0$$

Aktivierung: s 5

Beispiel:

Normalverteilung: Dichte

gebe x an:

[1] 0

Normalverteilung: $f(0;0,1) = 0.4$

5 <* 2>+≡

<Normalverteilung: berechne Dichte 52>

2.3 Exponentialverteilung

2.3.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
λ	1	lambda

Die Defaultsetzung erfolgt in Code-Chunk 96.

2.3.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (λ) für die Exponentialverteilung verändert.

Aktivierung: s 6

```
6 < * 2 > + ≡  
  < Exponentialverteilung: setze Parameter 53 >
```

2.3.3 Verteilungsfunktion

Steckbrief:

$$F(x; \lambda) = 1 - e^{-\lambda x} \quad x > 0; \quad \lambda > 0$$

Aktivierung: s 7

Beispiel:

Exponentialverteilung: Verteilungsfunktion

gebe x an:

[1] 1

Exponentialverteilung: F(1;1) = 0.6321

```
7 < * 2 > + ≡  
  < Exponentialverteilung: berechne Verteilungsfunktion 54 >
```

2.3.4 Prozentpunkt

Aktivierung: s 8

Beispiel:

Exponentialverteilung: Prozentpunkt

gebe w an:

[1] 0.5

Exponentialverteilung: $x.w = 0.693$ mit $F(x.w;1) = 0.5$

8 $\langle * 2 \rangle + \equiv$
 $\langle \text{Exponentialverteilung: bestimme Prozentpunkt 55} \rangle$

2.3.5 Dichte

Steckbrief:

$$f(x; \lambda) = \lambda e^{-\lambda x} \quad x > 0; \quad \lambda > 0$$

Aktivierung: s 9

Beispiel:

Exponentialverteilung: Dichte

gebe x an:

[1] 1

Exponentialverteilung: $f(1;1) = 0.37$

9 $\langle * 2 \rangle + \equiv$
 $\langle \text{Exponentialverteilung: berechne Dichte 56} \rangle$

2.4 t-Verteilung

2.4.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
ν	1	nu

Die Defaultsetzung erfolgt in Code-Chunk 97.

2.4.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (ν) für die t-Verteilung verändert.

Aktivierung: s 10

```
10 <* 2>+≡  
    <t-Verteilung: setze Parameter 57>
```

2.4.3 Verteilungsfunktion

Steckbrief:

$$F(x; \nu) = \int_{-\infty}^x \frac{\Gamma((\nu + 1)/2)}{\sqrt{\nu\pi} \Gamma(\nu/2)} \left(1 + \frac{u^2}{\nu}\right)^{-(\nu+1)/2} du \quad -\infty < x < +\infty; \quad \nu = 1, 2, \dots$$

Aktivierung: s 11

Beispiel:

t-Verteilung: Verteilungsfunktion

gebe x an:
[1] 0

t-Verteilung: F(0;1) = 0.5

```
11 <* 2>+≡  
    <t-Verteilung: berechne Verteilungsfunktion 58>
```

2.4.4 Prozentpunkt

Aktivierung: s 12

Beispiel:

t-Verteilung: Prozentpunkt

gebe w an:

[1] 0.5

t-Verteilung: x.w = 0 mit $F(x.w;1) = 0.5$

12 $\langle * 2 \rangle + \equiv$

\langle t-Verteilung: bestimme Prozentpunkt 59 \rangle

2.4.5 Dichte

Steckbrief:

$$f(x; \nu) = \frac{\Gamma((\nu + 1)/2)}{\sqrt{\nu\pi} \Gamma(\nu/2)} \left(1 + \frac{x^2}{\nu}\right)^{-(\nu+1)/2} \quad -\infty < x < +\infty; \quad \nu = 1, 2, \dots$$

Aktivierung: s 13

Beispiel:

t-Verteilung: Dichte

gebe x an:

[1] 0

t-Verteilung: $f(0;1) = 0.32$

13 $\langle * 2 \rangle + \equiv$

\langle t-Verteilung: berechne Dichte 60 \rangle

2.5 F-Verteilung

2.5.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
n_1	2	n1
n_2	2	n2

Die Defaultsetzung erfolgt in Code-Chunk 98.

2.5.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (n_1, n_2) für die F-Verteilung verändert.

Aktivierung: s 14

```
14 <* 2>+≡  
    <F-Verteilung: setze Parameter 61>
```

2.5.3 Verteilungsfunktion

Steckbrief:

$$F(x; \nu) = \int_0^x \frac{(n_1/n_2)^{n_1/2}}{B(n_1/2, n_2/2)} u^{n_1/2-1} \left(1 + \frac{n_1}{n_2} u\right)^{-(n_1+n_2)/2} du \quad x > 0; \quad n_1, n_2 = 1, 2, \dots$$

Aktivierung: s 15

Beispiel:

F-Verteilung: Verteilungsfunktion

gebe x an:
[1] 2

F-Verteilung: F(2;2,2) = 0.6667

```
15 <* 2>+≡  
    <F-Verteilung: berechne Verteilungsfunktion 62>
```

2.5.4 Prozentpunkt

Aktivierung: s 16

Beispiel:

F-Verteilung: Prozentpunkt

gebe w an:

[1] 0.5

F-Verteilung: x.w = 1 mit $F(x.w;2,2) = 0.5$

16 $\langle * 2 \rangle + \equiv$
 $\langle F\text{-Verteilung: bestimme Prozentpunkt 63} \rangle$

2.5.5 Dichte

Steckbrief:

$$f(x; n_1, n_2) = \frac{(n_1/n_2)^{n_1/2}}{B(n_1/2, n_2/2)} x^{n_1/2-1} \left(1 + \frac{n_1}{n_2}x\right)^{-(n_1+n_2)/2} \quad x > 0; \quad n_1, n_2 = 1, 2, \dots$$

Aktivierung: s 17

Beispiel:

F-Verteilung: Dichte

gebe x an:

[1] 1

F-Verteilung: $f(1;2,2) = 0.25$

17 $\langle * 2 \rangle + \equiv$
 $\langle F\text{-Verteilung: berechne Dichte 64} \rangle$

2.6 χ^2 -Verteilung

2.6.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
ν	1	nu

Die Defaultsetzung erfolgt in Code-Chunk 99.

2.6.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (ν) für die χ^2 -Verteilung verändert.

Aktivierung: s 18

```
18 <* 2>+≡  
  <chisq-Verteilung: setze Parameter 65>
```

2.6.3 Verteilungsfunktion

Steckbrief:

$$F(x; \nu) = \int_0^x \frac{1}{2^{\nu/2} \Gamma(\nu/2)} u^{(\nu/2)-1} e^{-u/2} du \quad x > 0; \quad \nu = 1, 2, \dots$$

Aktivierung: s 19

Beispiel:

```
chisq-Verteilung: Verteilungsfunktion
```

```
gebe x an:
```

```
[1] 2
```

```
chisq-Verteilung: F(2;1) = 0.8427
```

```
19 <* 2>+≡  
  <chisq-Verteilung: berechne Verteilungsfunktion 66>
```

2.6.4 Prozentpunkt

Aktivierung: s 20

Beispiel:

chisq-Verteilung: Prozentpunkt

gebe w an:

[1] 0.5

chisq-Verteilung: x.w = 0.455 mit $F(x.w;1) = 0.5$

20 < * 2 > + ≡
< chisq-Verteilung: bestimme Prozentpunkt 67 >

2.6.5 Dichte

Steckbrief:

$$f(x; \nu) = \frac{1}{2^{\nu/2} \Gamma(\nu/2)} u^{(\nu/2)-1} e^{-u/2} \quad x > 0; \quad \nu = 1, 2, \dots$$

Aktivierung: s 21

Beispiel:

chisq-Verteilung: Dichte

gebe x an:

[1] 1

chisq-Verteilung: $f(1;1) = 0.24$

21 < * 2 > + ≡
< chisq-Verteilung: berechne Dichte 68 >

2.7 Binomialverteilung

2.7.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
n	10	n
p	0.5	p

Die Defaultsetzung erfolgt in Code-Chunk 100.

2.7.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (n, p) für die Binomialverteilung verändert.

Aktivierung: s 22

```
22 <* 2>+≡  
    <Binomialverteilung: setze Parameter 69>
```

2.7.3 Verteilungsfunktion

Steckbrief:

$$F(x; n, p) = \sum_{i \leq x} \binom{n}{i} p^i (1-p)^{n-i} \quad -\infty < x < +\infty; \quad i = 0, 1, \dots, n, \quad p \in [0, 1]$$

Aktivierung: s 23

Beispiel:

Binomialverteilung: Verteilungsfunktion

gebe x an:

[1] 3

Binomialverteilung: $F(3; 10, 0.5) = 0.172$

```
23 <* 2>+≡  
    <Binomialverteilung: berechne Verteilungsfunktion 70>
```

2.7.4 Prozentpunkt

Aktivierung: s 24

Beispiel:

Binomialverteilung: Prozentpunkt

gebe w an:

[1] 0.5

Binomialverteilung: x.w = 5 mit $F(x.w;10,0.5) \leq 0.5$

24 $\langle * 2 \rangle + \equiv$

\langle Binomialverteilung: bestimme Prozentpunkt 71 \rangle

2.7.5 Wahrscheinlichkeitsfunktion

Steckbrief:

$$f(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x} \quad x = 0, 1, \dots, n; \quad p \in [0, 1]$$

Aktivierung: s 25

Beispiel:

Binomialverteilung: Wahrscheinlichkeitsfunktion

gebe x an:

[1] 2

Binomialverteilung: $f(2;10,0.5) = 0.044$

25 $\langle * 2 \rangle + \equiv$

\langle Binomialverteilung: berechne Wahrscheinlichkeitsfunktion 72 \rangle

2.8 Geometrische Verteilung

2.8.1 Namen und Defaultwerte der Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
p	0.5	p

Die Defaultsetzung erfolgt in Code-Chunk 101.

2.8.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (p) für die Geometrische Verteilung verändert.

Aktivierung: s 26

```
26 <* 2>+≡  
    <Geometrische Verteilung: setze Parameter 73>
```

2.8.3 Verteilungsfunktion

Steckbrief:

$$F(x; p) = \sum_{i \leq x} (1 - p)^i p \quad -\infty < x < +\infty; \quad i = 0, 1, \dots, \quad p \in [0, 1]$$

Aktivierung: s 27

Beispiel:

Geometrische Verteilung: Verteilungsfunktion

```
gebe x an:  
[1] 3
```

Geometrische Verteilung: $F(3; 0.5) = 0.938$

```
27 <* 2>+≡  
    <Geometrische Verteilung: berechne Verteilungsfunktion 74>
```

2.8.4 Prozentpunkt

Aktivierung: s 28

Beispiel:

Geometrische Verteilung: Prozentpunkt

gebe w an:

[1] 0.5

Geometrische Verteilung: $x.w = 0$ mit $F(x.w;0.5) \leq 0.5$

28 $\langle * 2 \rangle + \equiv$

\langle Geometrische Verteilung: bestimme Prozentpunkt 75 \rangle

2.8.5 Wahrscheinlichkeitsfunktion

Steckbrief:

$$f(x;p) = (1-p)^x p \quad x = 0, 1, \dots; \quad p \in [0, 1]$$

Aktivierung: s 29

Beispiel:

Geometrische Verteilung: Wahrscheinlichkeitsfunktion

gebe x an:

[1] 2

Geometrische Verteilung: $f(2;0.5) = 0.12$

29 $\langle * 2 \rangle + \equiv$

\langle Geometrische Verteilung: berechne Wahrscheinlichkeitsfunktion 76 \rangle

2.9 Poisson-Verteilung

2.9.1 Namen und Defaultwerteder Parameter

Die Input-Namensgebung und die Defaultsetzung der Parameter ist

Parameter	Wert	Input-Name
λ	1	lambda

Die Defaultsetzung erfolgt in Code-Chunk 102.

2.9.2 Setzung der Parameter

Es wird nachhaltig bis zur nächsten Aktivierung die Parametereinstellung (λ) für die Poisson-Verteilung verändert.

Aktivierung: s 30

```
30 <* 2>+≡  
    <Poisson-Verteilung: setze Parameter 77>
```

2.9.3 Verteilungsfunktion

Steckbrief:

$$F(x; \lambda) = \sum_{i \leq x} \frac{\lambda^i}{i!} e^{-\lambda} \quad -\infty < x < +\infty; \quad i = 0, 1, \dots, \quad \lambda > 0$$

Aktivierung: s 31

Beispiel:

Poisson-Verteilung: Verteilungsfunktion

gebe x an:
[1] 3

Poisson-Verteilung: F(3;1) = 0.981

```
31 <* 2>+≡  
    <Poisson-Verteilung: berechne Verteilungsfunktion 78>
```

2.9.4 Prozentpunkt

Aktivierung: s 32

Beispiel:

Poisson-Verteilung: Prozentpunkt

gebe w an:

[1] 0.5

Poisson-Verteilung: $x.w = 1$ mit $F(x.w;1) \leq 0.5$

32 $\langle * 2 \rangle + \equiv$

\langle Poisson-Verteilung: bestimme Prozentpunkt 79 \rangle

2.9.5 Wahrscheinlichkeitsfunktion

Steckbrief:

$$f(x; \lambda) = \frac{\lambda^x}{x!} e^{-\lambda} \quad x = 0, 1, \dots; \quad \lambda > 0$$

Aktivierung: s 33

Beispiel:

Poisson-Verteilung: Wahrscheinlichkeitsfunktion

gebe x an:

[1] 2

Poisson-Verteilung: $f(2;1) = 0.18$

33 $\langle * 2 \rangle + \equiv$

\langle Poisson-Verteilung: berechne Wahrscheinlichkeitsfunktion 80 \rangle

3 Konfidenzintervall

Das Nachschlagen eines Wertes einer Verteilungsfunktion und das Aufsuchen eines Prozentpunktes sind sicher die häufigsten Handlungen, die mit einer Tabellensammlung vorgenommen werden. Aber auch schon dem Anfänger, wie es der Student einer Statistik II Vorlesung sicher ist, stellen sich komplexere Aufgaben, wie zum Beispiel das Umgehen mit Konfidenzintervallen.

Dabei sind Fragestellungen zu lösen, wie zum Beispiel

1. Bestimme bei bekannter Normalverteilung ein Konfidenzintervall vorgegebener Konfidenzwahrscheinlichkeit.
2. Wie lautet die Lösung, wenn leider σ nicht bekannt ist?
3. Gegeben sei ein Intervall bei bekannter Normalverteilung, wie groß ist die Konfidenzwahrscheinlichkeit?
4. Wie läßt sich ein Konfidenzintervall angeben, wenn man nichts über die Verteilung weiß?

Die nachstehenden Lösungsangebote sind ein erster Schritt in Richtung einer Tabellensammlung, mit der komplexe Fragestellungen einfach beantwortet werden können.

In jedem der folgenden Abschnitte ist die Fragestellung jeweils auch in "Formelform" angegeben.

3.1 Normalverteilung: σ bekannt

$$P\left(\bar{X} - z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + z_{1-\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha$$

3.1.1 Liefere $z_{1-\alpha/2}$

Aktivierung: s 34

Beispiel:

Konfidenzintervall: sigma bekannt

gebe alpha an:

[1] 0.05

Konfidenzintervall: z-Wert fuer 1-0.05/2 ist 1.96

34 $\langle * 2 \rangle + \equiv$

$\langle \text{Konfidenzintervall: z-Wert 81} \rangle$

3.1.2 Bestimme Stichprobenumfang

Aktivierung: s 35

Beispiel:

Konfidenzintervall: Stichprobenumfang bei sigma bekannt

gebe sigma an:

[1] 2

gebe alpha an:

[1] 0.05

gebe Laenge an:

[1] 0.5

Der Mindeststichprobenumfang ergibt sich zu $n = 246$

35 $\langle * 2 \rangle + \equiv$

$\langle \text{Konfidenzintervall: Umfang 82} \rangle$

3.2 Normalverteilung: σ unbekannt

$$P\left(\bar{X} - t_{\nu, 1-\alpha/2} \frac{\sigma}{\sqrt{n}} \leq \mu \leq \bar{X} + t_{\nu, 1-\alpha/2} \frac{\sigma}{\sqrt{n}}\right) = 1 - \alpha$$

3.2.1 Liefere $t_{1-\alpha/2}$

Aktivierung: s 36

Beispiel:

Konfidenzintervall: sigma unbekannt

gebe alpha an:

[1] 0.05

gebe nu an:

[1] 20

Konfidenzintervall: t-Wert fuer $1-0.05/2$ und $\nu = 20$ ist 2.086

36 $\langle * 2 \rangle + \equiv$

$\langle \text{Konfidenzintervall: t-Wert 83} \rangle$

3.3 verteilungsfrei

$$P(X_{(c)} < \tilde{\mu} < X_{(n-c+1)}) = 1 - \alpha$$

3.3.1 Bestimme $1 - \alpha$

Aktivierung: s 37

Beispiel: Einfaches Konfidenzintervall: Konfidenzniveau

gebe n an

[1] 20

gebe c an

[1] 4

Die Konfidenzwahrscheinlichkeit ist 0.997

37 $\langle * 2 \rangle + \equiv$

$\langle \text{Konfidenzintervall: eins minus alpha 84} \rangle$

3.3.2 Bestimme c

Aktivierung: s 38

Beispiel: Einfaches Konfidenzintervall: c

gebe n an

[1] 20

gebe alpha an

[1] 0.01

c = 4

38 $\langle * 2 \rangle + \equiv$

$\langle \text{Konfidenzintervall: c 85} \rangle$

```
39 <* 2>+≡
    cat("\n!! Keine weiteren Verteilungen, etc. im Angebot !!\n\n")
```

4 Verteilungen: Verfeinerungen

Dieser Teil ist nur für den an Implementation und Umsetzung Interessierten von Bedeutung. Alle anderen können an dieser Stelle das Lesen einstellen.

Für die Übrigen hoffe ich, daß der hier praktizierte Literate-Programming-Stil das Verständnis trotz meiner Unzulänglichkeiten fördert.

4.1 Für alle

4.1.1 Kontrollierte Eingabe

Die Eingabe wird auf Einhaltung der jeweils angegebenen Bedingungen überprüft. Dazu gibt es die Funktion `kont.in`. (Siehe S. 40)

Argument für Verteilungsfunktion und Dichte Erlaubt sind alle Werte zwischen $-\infty$ und ∞ . Es liegt also Fall 1 der Funktion `kont.in` vor.

```
40 <read argument x 40>≡
    cat("\ngebe x an:\n")
    x <- kont.in("x",1)
```

This code is used in chunks 50, 52, 58, 60, 62, 66, 70, 74, and 78.

“Prozent” w für Prozentpunkt X_w Erlaubt sind Werte aus dem Intervall $[0,1]$. Es liegt also Fall 3 der Funktion `kont.in` vor.

```
41 <read argument w 41>≡
    cat("\ngebe w an:\n")
    w <- kont.in("w",3)
```

This code is used in chunks 51, 55, 59, 63, 67, 71, 75, and 79.

Alpha Erlaubt sind Werte aus dem Intervall $[0,1]$. Es liegt also Fall 3 der Funktion `kont.in` vor.

```
42 <read alpha 42>≡
    cat("\ngebe alpha an:\n")
    a <- kont.in("alpha",3)
```

This code is used in chunks 81–83 and 85.

Umfang n Erlaubt sind positive Integer. Es liegt also Fall 4 der Funktion `kont.in` vor.

```
43 <read n.k 43>≡  
  cat("\ngebe n an\n")  
  n.k <- kont.in("n",4)
```

This code is used in chunks 84 and 85.

4.1.2 Formatierung

Es werden diverse Formatierungsangebote gemacht. Dazu wird die S-Plus-Funktion `options` verwandt. In dieser Version einer virtuellen Tabellensammlung wurde davon abgesehen, dem Benutzer das Recht einzuräumen, die Stellenzahl selber zu kontrollieren.

Aus Gründen der leichteren “Lesbarkeit” wurde für die auszugebende Variable ein sprechender Name gewählt. Dies führt zu der vermeintlichen Redundanz in den Formatierungsangeboten.

```
44 <digit y 4 44>≡  
  set.digits(4)  
  y <- format(y)  
  set.digits(7)
```

This code is used in chunks 50, 54, 58, 62, and 66.

```
45 <digit y 3 45>≡  
  set.digits(3)  
  y <- format(y)  
  set.digits(7)
```

This code is used in chunks 70, 74, and 78.

```
46 <digit y 2 46>≡  
  set.digits(2)  
  y <- format(y)  
  set.digits(7)
```

This code is used in chunks 52, 56, 60, 64, 68, 72, 76, and 80.

```
47 <digit x 4 47>≡  
  set.digits(3)  
  x <- format(x)  
  set.digits(7)
```

This code is used in chunks 81 and 83.

```
48 <digit x 3 48>≡  
  set.digits(3)  
  x <- format(x)  
  set.digits(7)
```

This code is used in chunks 51, 55, 59, 63, 67, 71, 75, and 79.

4.2 Struktur

Alle Code-Chunks für die Verfeinerung der Berechnung von Verteilungsfunktion, Prozentpunkt und Dichte folgen dem gleichen Schema:

- 1 xyz-Verteilung: *was wird berechnet*
- 2 kontrollierte Benutzereingabe
- 3 S-Plus-Aufruf der benötigten Funktion (pxyz oder qxyz oder dxyz)
- 4 Formatierung des Ergebnisses
- 5 Ausgabe des Ergebnisses mit aktueller Parametersetzung

Bei der Parametersetzung sind die ersten beiden Schritte identisch. Nach erfolgter Setzung wird die aktuelle Parametersetzung angezeigt.

4.3 Normalverteilung

4.3.1 Setzung der Parameter

```
49 <Normalverteilung: setze Parameter 49>≡
   cat("\nNormalverteilung: Parametersetzung\n")
   cat("\nSetze mu fest:\n")
   mu <- kont.in("mu",1)
   cat("\nSetze sigma fest:\n")
   sig <- kont.in("sigma",2)
   cat(paste("\nNormalverteilung: mu = ",mu,", sigma = ",sig,"\n\n",sep=""))
```

This code is used in chunk 2.

4.3.2 Verteilungsfunktion

```
50 <Normalverteilung: berechne Verteilungsfunktion 50>≡
   cat("\nNormalverteilung: Verteilungsfunktion\n")
   <read argument x 40>
   y <- pnorm(x,mu,sig)
   <digit y 4 44>
   cat(paste("\nNormalverteilung: F(",x,";",mu,",",sig,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 3.

4.3.3 Prozentpunkt

```
51 <Normalverteilung: bestimme Prozentpunkt 51>≡
  cat("\nNormalverteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qnorm(w,mu,sig)
  <digit x 3 48>
  cat(paste("\nNormalverteilung: x.w = ",x," mit F(x.w;",mu,",",sig,") = ",w,
  "\n\n",sep=""))
```

This code is used in chunk 4.

4.3.4 Dichte

```
52 <Normalverteilung: berechne Dichte 52>≡
  cat("\nNormalverteilung: Dichte\n")
  <read argument x 40>
  y <- dnorm(x,mu,sig)
  <digit y 2 46>
  cat(paste("\nNormalverteilung: f(",x,";",mu,",",sig,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 5.

4.4 Exponentialverteilung

4.4.1 Setzung der Parameter

```
53 <Exponentialverteilung: setze Parameter 53>≡
  cat("\nExponentialverteilung: Parametersetzung\n")
  cat("\nSetze lambda fest:\n")
  l.e <- kont.in("lambda",2)
  cat(paste("\nExponentialverteilung: lambda = ",l.e,"\n\n",sep=""))
```

This code is used in chunk 6.

4.4.2 Verteilungsfunktion

```
54 <Exponentialverteilung: berechne Verteilungsfunktion 54>≡
  cat("\nExponentialverteilung: Verteilungsfunktion\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",2)
  y <- pexp(x,l.e)
  <digit y 4 44>
  cat(paste("\nExponentialverteilung: F(",x,";",l.e,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 7.

4.4.3 Prozentpunkt

```
55 <Exponentialverteilung: bestimme Prozentpunkt 55>≡
  cat("\nExponentialverteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qexp(w,1.e)
  <digit x 3 48>
  cat(paste("\nExponentialverteilung: x.w = ",x," mit F(x.w;",1.e,") = ",w,
  "\n\n",sep=""))
```

This code is used in chunk 8.

4.4.4 Dichte

```
56 <Exponentialverteilung: berechne Dichte 56>≡
  cat("\nExponentialverteilung: Dichte\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",2)
  y <- dexp(x,1.e)
  <digit y 2 46>
  cat(paste("\nExponentialverteilung: f(",x,";",1.e,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 9.

4.5 t-Verteilung

4.5.1 Setzung der Parameter

```
57 <t-Verteilung: setze Parameter 57>≡
  cat("\nt-Verteilung: Parametersetzung\n")
  cat("\nSetze nu fest:\n")
  nu.t <- kont.in("nu",4)
  cat(paste("\nt-Verteilung: nu = ",nu.t,"\n\n",sep=""))
```

This code is used in chunk 10.

4.5.2 Verteilungsfunktion

```
58 <t-Verteilung: berechne Verteilungsfunktion 58>≡
  cat("\nt-Verteilung: Verteilungsfunktion\n")
  <read argument x 40>
  y <- pt(x,nu.t)
  <digit y 4 44>
  cat(paste("\nt-Verteilung: F(",x,";",nu.t,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 11.

4.5.3 Prozentpunkt

```
59 <t-Verteilung: bestimme Prozentpunkt 59>≡
  cat("\nt-Verteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qt(w,nu.t)
  <digit x 3 48>
  cat(paste("\nt-Verteilung: x.w = ",x," mit F(x.w;",nu.t,") = ",w,
  "\n\n",sep=""))
```

This code is used in chunk 12.

4.5.4 Dichte

```
60 <t-Verteilung: berechne Dichte 60>≡
  cat("\nt-Verteilung: Dichte\n")
  <read argument x 40>
  y <- dt(x,nu.t)
  <digit y 2 46>
  cat(paste("\nt-Verteilung: f(",x,";",nu.t,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 13.

4.6 F-Verteilung

4.6.1 Setzung der Parameter

```
61 <F-Verteilung: setze Parameter 61>≡
  cat("\nF-Verteilung: Parametersetzung\n")
  cat("\nSetze n1 fest:\n")
  n1 <- kont.in("n1",4)
  cat("\nSetze n2 fest:\n")
  n2 <- kont.in("n2",4)
  cat(paste("\nF-Verteilung: n1 = ",n1," , n2 = ", n2,"\n\n",sep=""))
```

This code is used in chunk 14.

4.6.2 Verteilungsfunktion

```
62 <F-Verteilung: berechne Verteilungsfunktion 62>≡
  cat("\nF-Verteilung: Verteilungsfunktion\n")
  <read argument x 40>
  y <- pf(x,n1,n2)
  <digit y 4 44>
  cat(paste("\nF-Verteilung: F(",x,";",n1,"",n2,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 15.

4.6.3 Prozentpunkt

```
63 <F-Verteilung: bestimme Prozentpunkt 63>≡
  cat("\nF-Verteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qf(w,n1,n2)
  <digit x 3 48>
  cat(paste("\nF-Verteilung: x.w = ",x," mit F(x.w;",n1,",",n2,") = ",w,
  "\n\n",sep=""))
```

This code is used in chunk 16.

4.6.4 Dichte

```
64 <F-Verteilung: berechne Dichte 64>≡
  cat("\nF-Verteilung: Dichte\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",2)
  y <- df(x,n1,n2)
  <digit y 2 46>
  cat(paste("\nF-Verteilung: f(",x,";",n1,",",n2,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 17.

4.7 χ^2 -Verteilung

4.7.1 Setzung der Parameter

```
65 <chisq-Verteilung: setze Parameter 65>≡
  cat("\nchisq-Verteilung: Parametersetzung\n")
  cat("\nSetze nu fest:\n")
  nu.c <- kont.in("nu",4)
  cat(paste("\nchisq-Verteilung: nu = ",nu.c,"\n\n",sep=""))
```

This code is used in chunk 18.

4.7.2 Verteilungsfunktion

```
66 <chisq-Verteilung: berechne Verteilungsfunktion 66>≡
  cat("\nchisq-Verteilung: Verteilungsfunktion\n")
  <read argument x 40>
  y <- pchisq(x,nu.c)
  <digit y 4 44>
  cat(paste("\nchisq-Verteilung: F(",x,";",nu.c,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 19.

4.7.3 Prozentpunkt

```
67 <chisq-Verteilung: bestimme Prozentpunkt 67>≡
  cat("\nchisq-Verteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qchisq(w,nu.c)
  <digit x 3 48>
  cat(paste("\nchisq-Verteilung: x.w = ",x," mit F(x.w;",nu.c,") = ",w,
  "\n\n",sep=""))
```

This code is used in chunk 20.

4.7.4 Dichte

```
68 <chisq-Verteilung: berechne Dichte 68>≡
  cat("\nchisq-Verteilung: Dichte\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",2)
  y <- dchisq(x,nu.c)
  <digit y 2 46>
  cat(paste("\nchisq-Verteilung: f(",x,";",nu.c,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 21.

4.8 Binomialverteilung

4.8.1 Setzung der Parameter

```
69 <Binomialverteilung: setze Parameter 69>≡
  cat("\nBinomialverteilung: Parametersetzung\n")
  cat("\nSetze n fest:\n")
  n.b <- kont.in("n",4)
  cat("\nSetze p fest:\n")
  p.b <- kont.in("p",3)
  cat(paste("\nBinomialverteilung: n = ",n.b," p = ",p.b,"\n\n",sep=""))
```

This code is used in chunk 22.

4.8.2 Verteilungsfunktion

```
70 <Binomialverteilung: berechne Verteilungsfunktion 70>≡
  cat("\nBinomialverteilung: Verteilungsfunktion\n")
  <read argument x 40>
  y <- pbinom(x,n.b,p.b)
  <digit y 3 45>
  cat(paste("\nBinomialverteilung: F(",x,";",n.b,"",p.b,") = ",y,"\n\n",sep=""))
```

This code is used in chunk 23.

4.8.3 Prozentpunkt

```
71 <Binomialverteilung: bestimme Prozentpunkt 71>≡
  cat("\nBinomialverteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qbinom(w,n.b,p.b)
  <digit x 3 48>
  cat(paste("\nBinomialverteilung: x.w = ",x," mit F(x.w;"",n.b,"",p.b,"") <= ",w,
  "\n\n",sep=""))
```

This code is used in chunk 24.

4.8.4 Wahrscheinlichkeitsfunktion

```
72 <Binomialverteilung: berechne Wahrscheinlichkeitsfunktion 72>≡
  cat("\nBinomialverteilung: Wahrscheinlichkeitsfunktion\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",5,n.b)
  y <- dbinom(x,n.b,p.b)
  <digit y 2 46>
  cat(paste("\nBinomialverteilung: f(",x,";"",n.b,"",p.b,"") = ",y,"\n\n",sep=""))
```

This code is used in chunk 25.

4.9 Geometrische Verteilung

4.9.1 Setzung der Parameter

```
73 <Geometrische Verteilung: setze Parameter 73>≡
  cat("\nGeometrische Verteilung: Parametersetzung\n")
  cat("\nSetze p fest:\n")
  p.g <- kont.in("p",7)
  cat(paste("\nGeometrische Verteilung: p = ",p.g,"\n\n",sep=""))
```

This code is used in chunk 26.

4.9.2 Verteilungsfunktion

```
74 <Geometrische Verteilung: berechne Verteilungsfunktion 74>≡
  cat("\nGeometrische Verteilung: Verteilungsfunktion\n")
  <read argument x 40>
  y <- pgeom(x,p.g)
  <digit y 3 45>
  cat(paste("\nGeometrische Verteilung: F(",x,";"",p.g,"") = ",y,
  "\n\n",sep=""))
```

This code is used in chunk 27.

4.9.3 Prozentpunkt

```
75 <Geometrische Verteilung: bestimme Prozentpunkt 75>≡
  cat("\nGeometrische Verteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qgeom(w,p.g)
  <digit x 3 48>
  cat(paste("\nGeometrische Verteilung: x.w = ",x," mit F(x.w;",p.g,")",
    " <= ",w,"\n\n",sep=""))
```

This code is used in chunk 28.

4.9.4 Wahrscheinlichkeitsfunktion

```
76 <Geometrische Verteilung: berechne Wahrscheinlichkeitsfunktion 76>≡
  cat("\nGeometrische Verteilung: Wahrscheinlichkeitsfunktion\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",6)
  y <- dgeom(x,p.g)
  <digit y 2 46>
  cat(paste("\nGeometrische Verteilung: f(",x,;",",p.g,") = ",y,
    "\n\n",sep=""))
```

This code is used in chunk 29.

4.10 Poisson-Verteilung

4.10.1 Setzung der Parameter

```
77 <Poisson-Verteilung: setze Parameter 77>≡
  cat("\nPoisson-Verteilung: Parametersetzung\n")
  cat("\nSetze lambda fest:\n")
  l.p <- kont.in("lambda",2)
  cat(paste("\nPoisson-Verteilung: lambda = ",l.p,"\n\n",sep=""))
```

This code is used in chunk 30.

4.10.2 Verteilungsfunktion

```
78 <Poisson-Verteilung: berechne Verteilungsfunktion 78>≡
  cat("\nPoisson-Verteilung: Verteilungsfunktion\n")
  <read argument x 40>
  y <- ppois(x,l.p)
  <digit y 3 45>
  cat(paste("\nPoisson-Verteilung: F(",x,;",",l.p,") = ",y,
    "\n\n",sep=""))
```

This code is used in chunk 31.

4.10.3 Prozentpunkt

```
79 <Poisson-Verteilung: bestimme Prozentpunkt 79>≡
  cat("\nPoisson-Verteilung: Prozentpunkt\n")
  <read argument w 41>
  x <- qpois(w,l.p)
  <digit x 3 48>
  cat(paste("\nPoisson-Verteilung: x.w = ",x," mit F(x.w;","l.p,")",
    " <= ",w,"\n\n",sep=""))
```

This code is used in chunk 32.

4.10.4 Wahrscheinlichkeitsfunktion

```
80 <Poisson-Verteilung: berechne Wahrscheinlichkeitsfunktion 80>≡
  cat("\nPoisson-Verteilung: Wahrscheinlichkeitsfunktion\n")
  cat("\ngebe x an:\n")
  x <- kont.in("x",6)
  y <- dpois(x,l.p)
  <digit y 2 46>
  cat(paste("\nPoisson-Verteilung: f(",x;","l.p,") = ",y,
    "\n\n",sep=""))
```

This code is used in chunk 33.

5 Konfidenzintervall: Verfeinerung

5.1 Normalverteilung: σ bekannt

5.1.1 Liefere $z_{1-\alpha/2}$

```
81 <Konfidenzintervall: z-Wert 81>≡
  cat("\nKonfidenzintervall: sigma bekannt\n")
  <read alpha 42>
  x <- qnorm(1-a/2)
  <digit x 4 47>
  cat(paste("\nKonfidenzintervall: z-Wert fuer 1-",a,"/2 ist ",x,"\n\n",sep=""))
```

This code is used in chunk 34.

5.1.2 Bestimme Stichprobenumfang

```
82 <Konfidenzintervall: Umfang 82>≡
  cat("\nKonfidenzintervall: Stichprobenumfang bei sigma bekannt\n")
  cat("\ngebe sigma an:\n")
  sig.k <- kont.in("sigma",2)
  cat("\ngebe Laenge an:\n")
  laen <- kont.in("Laenge",2)
  <read alpha 42>
  z <- qnorm(1-a/2)
  umf <- ((2*sig.k*z)/laen)^2
  cat(paste("\nDer Mindeststichprobenumfang ergibt sich zu n =",
            ceiling(umf),"\n\n"))
```

This code is used in chunk 35.

5.2 Normalverteilung: σ unbekannt

5.2.1 Liefere $t_{1-\alpha/2}$

```
83 <Konfidenzintervall: t-Wert 83>≡
  cat("\nKonfidenzintervall: sigma unbekannt\n")
  <read alpha 42>
  cat("\ngebe nu an:\n")
  nu.k <- kont.in("nu",4)
  x <- qt(1-a/2,nu.k)
  <digit x 4 47>
  cat(paste("\nKonfidenzintervall: t-Wert fuer 1-",a,"/2 und nu = ",nu.k,
            " ist ",x,"\n\n",sep=""))
```

This code is used in chunk 36.

5.3 verteilungsfrei

5.3.1 Bestimme $1 - \alpha$

```
84 <Konfidenzintervall: eins minus alpha 84>≡
  cat("Einfaches Konfidenzintervall: Konfidenzniveau\n")
  <read n.k 43>
  n.k2 <- floor(n.k/2)
  cat("\ngebe c an\n")
  c.k <- kont.in("c",5,n.k2)
  al <- 2*pbinom(c.k-1,n.k,0.5)
  ko <- 1-al
  set.digits(4)
  ko <- format(ko)
  set.digits(7)
  cat(paste("\nDie Konfidenzwahrscheinlichkeit ist",ko,"\n\n"))
```

This code is used in chunk 37.

5.3.2 Bestimme c

```
85 <Konfidenzintervall: c 85>≡
  cat("Einfaches Konfidenzintervall: c\n")
  <read n.k 43>
  <read alpha 42>
  aa <- a/2
  c.k <- qbinom(aa,n.k,0.5)
  cat(paste("\nc = ",c.k,"\n\n"))
```

This code is used in chunk 38.

6 Funktionen und Defaultsetzungen

6.1 Kontrollierter Input

Die vom Benutzer abgeforderten Parametersetzungen und Argumentangaben müssen je nach Verteilung gewisse Nebenbedingungen erfüllen. Es ist daher eine Eingaberoutine erforderlich, die die Einhaltung dieser Bedingungen erzwingt. Es sollen die nachstehenden Fälle betreut werden.

Fall	Bedeutung
1	Eingabe ist numerisch, dank <code>scan</code>
2	Eingabe muß größer Null sein
3	Eingabe muß im Intervall $[0,1]$ liegen
4	Eingabe muß Integer größer Null sein
5	Eingabe muß Integer aus Menge $\{0, \dots, \text{grenz}\}$ sein
6	Eingabe muß Integer sein
7	Eingabe muß im Intervall $(0,1)$ liegen

6.1.1 Die Funktion `kont.in`

Die eingeforderte Aufgabe soll von der Funktion `kont.in` erledigt werden. Neben der Angabe des jeweils betrachteten Falles (Parameter `fall`) wird der Name (`who`) der einzulesenden Variablen benötigt, damit eine sinnvolle Kommunikation mit dem Benutzer im Fehlerfall stattfinden kann. Auf dem Parameter `grenz` wird die im 5. Fall benötigte Information übergeben. Die Defaultsetzung ist 1.

Die Funktion `kont.in` beendet den Dialog mit dem Benutzer erst dann, wenn eine fall-gerechte Eingabe von ihm gemacht wurde.

```
86 <start 1>+≡
   kont.in <- function(who,fall,grenz=1){
   korrekt <- FALSE
   while ( korrekt == FALSE ){
     res <- scan(n=1)
     switch(fall,
       { <Fall 1 87>
       },
       { <Fall 2 88>
       },
       { <Fall 3 89>
       },
       { <Fall 4 90>
       },
       { <Fall 5 91>
       },
       { <Fall 6 92>
```

```

    },
    { <Fall 7 93>
  }) }
return(res) }

```

6.1.2 Allgemeine Annahmen

Die Benutzereingabe steht auf der Variablen `res`.

Für die folgenden Verfeinerungsschritte wird naiv unterstellt, daß die interne Darstellung der Zahlen keine Rolle spielt, z.B daß `res == floor(res)` den Wert *true* hat, wenn `res` den Wert 1.0 hat.

6.1.3 Verfeinerung für Fall 1

Dank der Voreinstellung von `scan` erfolgt immer eine korrekte Eingabe.

```

87 <Fall 1 87>≡
    korrekt <- TRUE

```

This code is used in chunk 86.

6.1.4 Verfeinerung für Fall 2

Der Vergleich `res > 0` bringt's.

```

88 <Fall 2 88>≡
    if (res > 0) { korrekt <- TRUE
    }else{
        cat(paste("\n",who," muss > 0 sein, neuer Versuch!\n"))}

```

This code is used in chunk 86.

6.1.5 Verfeinerung für Fall 3

Die beiden Vergleiche `res <= 1` und `res >= 0` müssen den Wert *true* ergeben, damit die Eingabe korrekt ist.

```

89 <Fall 3 89>≡
    if ((res <= 1) & (res >= 0)) { korrekt <- TRUE
    }else{
        cat(paste("\n",who," muss aus [0,1] sein, neuer Versuch!\n"))}

```

This code is used in chunk 86.

6.1.6 Verfeinerung für Fall 4

Hat `res == floor(res)` den Wahrheitswert `true` und ist außerdem `res` größer Null, dann liegt eine korrekte Eingabe vor.

```
90 <Fall 4 90>≡
    if ((res > 0) & (res == floor(res)))
        { korrekt <- TRUE
          res <- as.integer(res)
        }else{
          cat(paste("\n",who," muss Integer > 0 sein, neuer Versuch!\n"))}
```

This code is used in chunk 86.

6.1.7 Verfeinerung für Fall 5

Zusätzlich zum Fall 5 muß noch die Bedingung `res <= grenz` erfüllt sein.

```
91 <Fall 5 91>≡
    if ((res >= 0) & (res <= grenz) & (res == floor(res)))
        { korrekt <- TRUE
          res <- as.integer(res)
        }else{
          cat(paste("\n",who," muss Integer aus [0,",grenz,"] sein,",
                    "neuer Versuch!\n"))}
```

This code is used in chunk 86.

6.1.8 Verfeinerung für Fall 6

Fast wie Fall 4, nur daß `res` größer gleich Null sein muß.

```
92 <Fall 6 92>≡
    if ((res >= 0) & (res == floor(res)))
        { korrekt <- TRUE
          res <- as.integer(res)
        }else{
          cat(paste("\n",who," muss Integer sein, neuer Versuch!\n"))}
```

This code is used in chunk 86.

6.1.9 Verfeinerung für Fall 7

Die beiden Vergleiche `res < 1` und `res > 0` müssen den Wert *true* ergeben, damit die Eingabe korrekt ist.

```
93 <Fall 7 93>≡
    if ((res < 1) & (res > 0)) { korrekt <- TRUE
      }else{
          cat(paste("\n",who," muss aus (0,1) sein, neuer Versuch!\n"))}
```

This code is used in chunk 86.

6.2 Digits

Diese Funktion setzt die Stellenzahl für die Ausgabe auf den gewünschten Wert `d`

```
94 <start 1>+≡
    set.digits <- function(d) options(digits=d)
```

6.3 Defaultsetzungen ...

6.3.1 ... für Normalverteilung

Siehe auch S. 7

```
95 <start 1>+≡
    mu <- 0
    sig <- 1
```

6.3.2 ... für Exponentialverteilung

Siehe auch S. 9

```
96 <start 1>+≡
    l.e <- 1
```

6.3.3 ... für t-Verteilung

Siehe auch S. 11

```
97 <start 1>+≡
    nu.t <- 1
```

6.3.4 ... für F-Verteilung

Siehe auch S. 13

```
98 <start 1>+≡  
    n1 <- 2  
    n2 <- 2
```

6.3.5 ... für χ^2 -Verteilung

Siehe auch S. 15

```
99 <start 1>+≡  
    nu.c <- 1
```

6.3.6 ... für Binomialverteilung

Siehe auch S. 17

```
100 <start 1>+≡  
    n.b <- 10  
    p.b <- 0.5
```

6.3.7 ... für Geometrische Verteilung

Siehe auch S. 19

```
101 <start 1>+≡  
    p.g <- 0.5
```

6.3.8 ... für Poisson-Verteilung

Siehe auch S. 21

```
102 <start 1>+≡  
    l.p <- 1
```

7 Literatur

Becker, R.A., Chambers, J., Wilks, A.R.: The New S Language.
A Programming Environment for Data Analysis and Graphics
Wadsworth and Brooks/Cole, Pacific Grove, 1988

Nolte, Th.: Eine WEB-basierte statistische Tabellensammlung
Konzeption und Implementation eines Prototyps
Fakultät für Wirtschaftswissenschaften, Universität Bielefeld
Diplomarbeit, September 2001

Wetzel, W., Jöhnk, M.-D., Naeve, P.: Statistische Tabellen
Verlag Walter de Gruyter & Co., Berlin, 1967

Wolf, H. P.: Eine Reportschmiede für den Datenanalytiker
Fakultät für Wirtschaftswissenschaften, Universität Bielefeld
Diskussionspapier 301, 1995

Wolf, H. P.: REVIVE-Funktionen zur Arbeit mit wiederbelebbaren Papieren unter R
Fakultät für Wirtschaftswissenschaften, Universität Bielefeld
Diskussionspapier 415, 1999

8 Index

Als eine Art Index wird hier noch einmal eine Übersicht der Code-Chunks gegeben.

Verteilungen

Verteilung	Parameter	Verteilungsfkt	Prozentpkt	Dichte
Normalverteilung	s 2	s 3	s 4	s 5
Exponentialverteilung	s 6	s 7	s 8	s 9
t-Verteilung	s 10	s 11	s 12	s 13
F-Verteilung	s 14	s 15	s 16	s 17
χ^2 -Verteilung	s 18	s 19	s 20	s 21
Binomialverteilung	s 22	s 23	s 24	s 25
Geometrische Verteilung	s 26	s 27	s 28	s 29
Poisson-Verteilung	s 30	s 31	s 32	s 33

Konfidenzintervalle

Konfidenzintervall	Sektion
z-Wert	s 34
Länge	s 35
t-Wert	s 36
Einfach: $1 - \alpha$	s 37
Einfach: c	s 38