



Konfidenzintervall und p-Wert im Bootstrap-Verfahren

**Perzentil-Verfahren
Perzentil-t -Verfahren
BCa-Verfahren
Standardfehler-basiertes Verfahren**

Kurt Holm

Almo Statistik-System
www.almo-statistik.de
kurt.holm@jku.at

Weitere Almo-Dokumente

Die folgenden Dokumente können alle von der Handbuchseite in <http://www.almo-statistik.de> kostenlos heruntergeladen werden

0. `Arbeiten mit Almo.PDF` (1 MB)
- 1a. `Eindimensionale Tabellierung.PDF` (1,8 MB)
- 1b. `Zwei- und drei-dimensionale Tabellierung.PDF` (1.1 MB)
2. `Beliebig-dimensionale Tabellierung.PDF` (1.7 MB)
3. `Nicht-parametrische Verfahren.PDF` (0.9 MB)
4. `Kanonische Analysen.PDF` (1.8 MB)
`Diskriminanzanalyse.PDF` (1.8 MB)
enthält: Kanonische Korrelation, Diskriminanzanalyse, bivariate Korrespondenzanalyse, optimale Skalierung
5. `Korrelation.PDF` (1.4 MB)
6. `Allgemeine multiple Korrespondenzanalyse.PDF` (1.5 MB)
7. `Allgemeines ordinales Rasch-Modell.PDF` (0.6 MB)
- 7a. `Wie man mit Almo ein Rasch-Modell rechnet.PDF` (0.2 MB)
8. `Tests auf Mittelwertsdifferenz, t-Test.PDF` (1,6 MB)
9. `Logitanalyse.pdf` (1,2MB) enthält Logit- und Probitanalyse
- 9b. `Bootstrap bei Logit- und Probitanalyse.pdf`
10. `Koeffizienten der Logitanalyse.PDF` (0,06 MB)
11. `Daten-Fusion.PDF` (1,1 MB)
12. `Daten-Imputation.PDF` (1,3 MB)
13. `ALM Allgemeines Lineares Modell.PDF` (2.3 MB)
- 13a. `ALM Allgemeines Lineares Modell II.PDF` (2.7 MB)
- 13b. `Bootstrap bei Allgemeinem Linearem Modell III.PDF`
14. `Ereignisanalyse: Sterbetafel-Methode, Kaplan-Meier-Schätzer, Cox-Regression.PDF` (1,5 MB)
15. `Faktorenanalyse.PDF` (1,6 MB)
- 15a. `Bootstrap bei Faktorenanalyse.PDF`
16. `Konfirmatorische Faktorenanalyse.PDF` (0,3 MB)
17. `Clusteranalyse.PDF` (3 MB)
18. `Pisa 2012 Almo-Daten und Analyse-Programme.PDF` (17 KB)
19. `Guttman- und Mokken-Skalierung.PFD` (0.8 MB)
20. `Latent Structure Analysis.PDF` (1 MB)
21. `Statistische Algorithmen in C` (80 KB)
22. `Conjoint-Analyse` (PDF 0,8 MB)
23. `Ausreisser entdecken` (PDF 170 KB)
24. `Statistische Datenanalyse Teil I, Data Mining I`
25. `Statistische Datenanalyse Teil II, Data Mining II`
26. `Statistische Datenanalyse Teil III, Arbeiten mit Almo-Datenanalyse-System`
27. `Mehrfachantworten. Tabellierung von Fragen mit Mehrfachantworten`
28. `Metrische multidimensionale Skalierung (MDS)` (0,4 MB)
29. `Metrisches multidimensionales Unfolding (MDU)` (0,6 MB)
30. `Nicht-metrische multidimensionale Skalierung (MDS)` (0,4 MB)
31. `Pfadanalyse.PDF` (0,7 MB)
32. `Datei-Operationen mit Almo` (1,1 MB)
33. `Wählerstromanalyse und Wahlhochrechnung` (1,6 MB)
34. `Soziometrie. Auswertung soziometrischer Daten` (0,5 MB)
35. `Konfidenzintervall und p-Wert beim Bootstrap-Verfahren` (200 KB)

Inhaltsverzeichnis

KONFIDENZINTERVALL UND P-WERT	4
1 Unser Beispiel	4
1.1. Notation	5
2 Zum Begriff "Konfidenzintervall"	6
3 Die Bootstrap-Eingabebox	8
3.1 Die Konfidenzintervall-Verfahren	8
3.2 Vergleich der Konfidenzintervall-Verfahren	9
3.3 Die aufsteigend sortierten Koeffizienten aus den Bootstrapstichproben (ASR).....	11
4 Das einfache Perzentil-Verfahren	12
4.1 Der Signifikanz p-Wert aus dem einfachen Perzentil-Verfahren.....	12
5 Das Perzentil-t -Verfahren	14
5.1 Der p-Wert aus dem Perzentil-t -Verfahren	16
5.2 Das symmetrische Perzentil-t-Verfahren	16
6 Das BC- und BCa-Verfahren	17
6.0 Notation und Übersicht	17
6.0.1 Übersicht	18
6.1 Der Kalkül	21
6.1.1 Grafische Darstellung des Konfidenzintervalls.....	25
6.1.2 Der Akzelerationskoeffizient	28
6.2 Der Signifikanz p-Wert beim BC- und BCa-Verfahren	29
6.2.1 Kein Null vorhanden	33
6.3 BC- bzw. BCa-Verfahren mit <i>mehrfachen</i> Originalwerten.....	33
6.3.1 Erweiterte Notation	33
6.3.2 Ein Beispiel	35
6.3.3 Die Bootstrap-Optionsbox für <i>mehrfache Originalwerte</i>	37
6.3.4 Die Modi für das BC- bzw. BCa-Verfahren bei mehrfachen Originalwerten.....	38
6.3.5 Änderungen im Kalkül	42
6.3.6 Der p-Wert bei mehrfachem Originalwert	43
6.3.7 Ergebnis-Ausgabe und "zusätzliche Informationen"	43
6.3.8 Das effektive Vertrauensniveau	45
7 "Standardfehler-basiertes" Verfahren.	46
7.1 Der Signifikanz p-Wert	47
A Exkurs: Die kumulative Standard-Normalverteilung.	48
B Exkurs: C-Programm für "z_von_p" und "p_von_z"	54
Literatur	60

Konfidenzintervall und p-Wert

1 Unser Beispiel

Wir rechnen mit der Bootstrap-Programm-Maske **Prog20BS.Alm** für die Datei "Adat.fre" eine Regressionsanalyse. Die Programm-Maske findet man nach Klick auf den Knopf "alle Progs" am Oberrand des Almo-Femsters. Die Datei ist im Almo-Ordner "Testdat" enthalten.

die abhängige Variable: Leistung (in einem Test)
die unabhängigen quantitativen Variablen: Alter (in Jahren)
Freizeit (in Stunden)
Urlaub (subjektive Erholung)

Zahl der Bootstrap-Stichproben: 1000
das Konfidenzniveau setzen wir auf: 95%
das Konfidenzintervall wird mit dem "einfachen Perzentil"-Verfahren ermittelt.

Die Eingabeboxen für die abhängige Variable "Leistung" und die 3 unabhängigen Variablen in der Programm-Maske Prog20BS sind folgende

quantitative abhängige Zielvariable

1
0=quant. Variable als diskrete Variable behandeln
1=quant. Variable als kontinuierliche Variable behandeln

Analyze-Variable: Unabhängige Variable

nominale unabhängige Variable

Interaktionen x. Ordnung bilden
Ausgewählte Interaktionen bilden
0 =keine Interaktionen bilden

paarweise Vergleiche für die nominalen unabhängigen Variablen rechnen

quantitative unabhängige Variable

ordinale unabhängige Variable

Wir erhalten aus Prog20BS folgendes (gekürztes) Ergebnis für die abhängige Variable "Leistung"

Tabelle 1 einfaches Perzentil-Verfahren
Abhängige Variable: Leistung

	Original- stichprobe	Bootstrap-Ergebnisse					
		Mitt.wert aus Bootstrap BM	Verzerr.	Standard fehler	Signifik. p	Konfidenzintervall einfaches Perzentil Konf.niv=0.950	
	Borg					unten	oben
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0010	0.6719	0.7588
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4300	-0.0433	0.0844
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0320	0.0281	0.4859
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9698	-21.9012
.....							
multiple Korr. R	0.9013	0.9018	0.0005	0.0087	-	0.8835	0.9178

1.1. Notation

In späteren Abschnitten werden wir die Notation erweitern. Siehe z.B. Abschnitt 6.0 und 6.3.1.

Wir verwenden für die Notation folgende Einzelbuchstaben und Buchstabenfolgen

B = Zahl der Bootstrapstichproben

ASR = das ist ein Kurzbezeichnung für die "aufsteigend sortierte Reihe der β -Werte aus den Bootstrapstichproben. (Siehe als Beispiel Abschnitt 3.3)

β = bezeichnet den Wert des Koeffizienten, der dem Bootstrap-Verfahren unterworfen wurde. Beim ALM z.B. ist dies der Regressionskoeffizient, bei der Faktorenanalyse ist es die Faktorladung, beim Korrelieren der Korrelationskoeffizient etc. Gemeint ist der Wert des Koeffizienten

k = ist die Stelle, die der Koeffizient in der ASR, der aufsteigend sortierte Reihe der β -Werte aus den Bootstrapstichproben einnimmt

org = bezeichnet den Koeffizienten, der aus der Originalstichprobe hervorgegangen ist

Borg = das ist die Kombination aus " β " und "org", also der Wert des β -Koeffizienten aus der Originalstichprobe.
 Im Text bezeichnen wir Borg auch oft kurz als "Originalwert"

korg = das ist die Kombination aus "k" und "org", also die Stelle des Borg-Koeffizienten in ASR.
 In der ASR können Werte aus Bootstrapstichproben enthalten sein, die exakt gleich dem Borg-Wert aus der Originalstichprobe sind. Das ist aber die Ausnahme. In diesem Fall bezeichnen wir die Stelle des nächst höheren Werts in der ASR ebenfalls mit korg

BM = mittlerer β -Wert aus allen Bootstrapstichproben

kM = Stelle k von BM in ASR

kMitte = Mitte von ASR (=B/2). Im Beispiel: 500

β_u = unterer Grenzwert des Konfidenzintervalls

β_o = oberer Grenzwert

ku = Stelle der unteren Konfidenzgrenze β_u in ASR

ko = Stelle der oberen Konfidenzgrenze β_o in ASR

konfniv = Vertrauensniveau (Beispiel: 95)

α = $1 - \text{konfniv}/100$ (Beispiel: = 0.05)
 $\alpha/2$ = $(1 - \text{konfniv}/100)/2$ (Beispiel: = 0.25)
 $1 - \alpha/2$ = $0.5 + \text{konfniv}/200$ (Beispiel: = 0.975)

α wird in der Literatur gelegentlich als "Vertrauens-Kriterium" bezeichnet

kNull = Stelle in der aufsteigend sortierten Reihe ASR, an welcher der β -Wert gleich 0 ist oder 0 zum 1. Mal überschritten wird

Um Verwechslungen zwischen "Konfidenzintervall" und "Konfidenzniveau" zu vermeiden werden wir letzteren Begriff durch "Vertrauensniveau" ersetzen - nicht jedoch die Kurzbezeichnungen

konfniv, ekonfniv

die überwiegend in Tabellen und originalen Almo-Ausgaben verwendet werden.

2 Zum Begriff "Konfidenzintervall"

Wir betrachten die Variable "Freizeit". Ihr Regressionskoeffizient $\beta_{org} = 0.0235$ ist ein *Punktschätzer* des "wahren" Populationswerts. Ob dieser Schätzwert brauchbar ist, kann man beurteilen, wenn man den Standardfehler und zusätzlich noch den p-Wert betrachtet. Für "Freizeit" ist der Standardfehler mit 0.0314 sogar größer als der β -Wert selbst, der p-Wert dann entsprechend mit 0.4300 auch nicht signifikant

Das Konfidenzintervall von $\beta_u = -0.0431$ bis $\beta_o = 0.0844$ ist ein *Intervallschätzer* für den "wahren" Populationswert. Er sagt uns: Wenn der empirische β_{org} -Wert = 0.0235 ist, dann weist das darauf hin, dass der "wahre" Populationswert im angegebenen Intervall von -0.0431 bis 0.0844 liegt - und dies mit einer von uns angeforderten Wahrscheinlichkeit von 95%.

Sätze zu Konfidenzintervall:

- Wir können nicht real 1000 Stichproben aus der Grundgesamtheit (Population) ziehen. Mit dem Bootstrap-Verfahren simulieren wir diesen Vorgang. Bei einem empirisch ermittelten β_{org} -Wert von 0.0235 und einem vorgegebenen Vertrauensniveau von $\text{konfniv} = 95$ würden wir mit 95%-iger Wahrscheinlichkeit erhalten:
 - 2,5%, d.h. 25 Stichproben, deren β -Werte unter β_u (-0.0433) liegen würde und ebenfalls 25 Stichproben, deren β -Werte über β_o (0.0844) liegen würden, zusammen also 5% bzw. 50 Stichproben
 - und 95%, d.h. 950 Stichproben, deren β -Werte zwischen den Werten $\beta_u \geq -0.0433$ und $\beta_o \leq 0.0844$ liegen.

Dies gilt für das "einfache" Perzentil-Verfahren. Bei den anderen Verfahren muss die Aufteilung der 5% nicht notwendigerweise symmetrisch sein. In Tabelle 2 wird gezeigt, wie sich das Konfidenzintervall verändert, wenn das Vertrauensniveau verändert wird

Tabelle 2

	Originalwert β_{org}	Vertrauensniveau konfniv	Konfidenzintervall		
			unten β_u	oben β_o	Breite
Freizeit	0.0235	90%	-0.0293	0.0746	0.1039
Freizeit	0.0235	95%	-0.0431	0.0844	0.1275
Freizeit	0.0235	99%	-0.0590	0.1059	0.1749

- Je größer das Vertrauensniveau umso breiter wird das Konfidenzintervall und umso weniger "eingrenzbare" wird der "wahre" Populationswert. Ein Konfidenzintervall mit beispielsweise einem Vertrauensniveau von 90% bedeutet: Der "wahre" Populationswert von β für Freizeit liegt mit 90% Wahrscheinlichkeit im Intervall von -0.0293 bis 0.0746. Mit der höheren Wahrscheinlichkeit von $\text{konfniv} = 95\%$ verbreitert sich das Intervall auf -0.0431 bis 0.0844 und bei $\text{konfniv} = 99\%$

nochmals auf -0.0590 bis 0.1059. Der genaue Wert des Koeffizienten in der Population wird also immer weniger "eingrenzbare" je höher das Vertrauensniveau vom Forscher angesetzt wird. Bei konfniv=99,99999.. kann schließlich jeder in der Stichprobe auftretende Wert der "wahre" Populationswert sein - und dies mit 99,999..%-iger Sicherheit. Der Informationsgewinn ist dann gleich Null.

3. Ein guter Kompromiss ist es, sich auf ein Vertrauensniveau von 95% als Standard zu einigen, wie dies in der Gemeinschaft der statistisch arbeitenden Forscher geschehen ist.
4. Konfidenzintervalle verschiedener Variablen vergleichen
Wir ermitteln mit der Bootstrap-Programm-Maske Prog05m8 den Mittelwert aus 4 Variablen der Datei "Adat.fre". Die Programm-Maske findet man nach Klick auf den Knopf "Verfahren" dann "Bootstrap" am Oberrand des Almo-Fensters. Die Datei ist im Almo-Ordner "Testdat" enthalten.

Bootstrap-Verfahren:
einfaches Perzentil-Verfahren

Mittelwert fuer quantitative Variable	arithmetischer Mittelwert					
	*a	*b	Standard fehler	Konfidenzintervall *d Vertrauensniveau: 95%		
	Original	Bootstrap		unten	oben	Breite
1 V1 Bildungsnive	9.4048	9.4066	0.1595	9.0971	9.7070	0.6099
2 V2 Berufsqualif	9.0932	9.0967	0.1465	8.7935	9.3856	0.5921
3 V3 Freizeit	8.8050	8.8054	0.1325	8.5382	9.0647	0.5266
4 V11 Alter	41.6769	41.6784	0.2633	41.1869	42.1867	0.9998

-
- *a Wert aus Originalstichprobe
 - *b Mittelwert aus 1000 Bootstrapstichproben
 - *d Vertrauensniveau für Konfidenzintervall: 95%

Die Frage lautet: Können die 4 Variablen in ihren Intervallgrenzen und in der Intervallbreite und dem Standardfehler verglichen werden und kann dadurch Die Güte bzw. die Brauchbarkeit der Variablen beurteilt werden ?

Die ertsten 3 Variablen V1,2,3 sind mit dem gleichen "Maßstab" gemessen worden. Die 4. Variable (das Alter der Probanden) in Jahren, mit einem anderen "Maßstab" Es wird sofort ersichtlich, dass ein Vergleich zwischen den Variablen nur möglich ist, wenn

1. das Vertrauensniveau gleich ist (das ist in unserem Beispiel der Fall)
2. und wenn die Variablen mit demselben "Maßstab" gemessen wurden. Das ist nicht der Fall

Die ersten 3 Variablen sind jedoch nach diesen beiden Kriterien vergleichbar. Dabei können wir eine Wertung vornehmen. Die "beste" Variable ist V3. Sie besitzt die engste Breite des Konfidenzintervalls und den kleinsten Standardfehler. Dann folgt in der Bewertung V2. Die "schlechteste" Variable ist V1 mit der größten Intervallbreite und dem größten Standardfehler. Wir erkennen, dass Breite und Standardfehler einander entsprechen und sogar austauschbar sind, wenn es darum geht, die Güte/Brauchbarkeit von vergleichbaren Variablen zu bewerten.

Wenn die oben formulierte 2. Bedingung nicht erfüllt ist, dann darf das Konfidenzintervall nicht verwendet werden, um Variable auf ihre Güte und Brauchbarkeit zu vergleichen, auch wenn sie dem gleichen Vertrauensniveau (von z.B.95%) unterworfen waren

5. Welche bedeutsame Information liefert das Konfidenzintervall.
 - a. Liegt der originale empirische Borg-Wert außerhalb des Konfidenzintervalls dann liegt er nicht in dem Bereich, in dem sich der "wahre" Populationswert mit einer Wahrscheinlichkeit von konfniv (z.B.95%) befindet. Sein p-Wert ist schlechter (also größer) als $1 - \text{konfniv}/100$ ($p > 0.05$).
Es muss jedoch eingeschränkt werden: Dass Borg außerhalb des Konfidenzintervalls liegt ist beim noch darzustellenden BC- und BCa-Verfahren nicht möglich. Beim einfachen Perzentil-Verfahren ist das im Prinzip möglich, aber real äußerst unwahrscheinlich.
 - b. Liegt der Wert 0 innerhalb des Intervalls (inklusive der Grenzwerte), dann

ist der Populationswert mit einer Irrtumswahrscheinlichkeit von $1 - \text{konfniv}/100$ nicht von 0 verschieden, bei $\text{konfniv}=95\%$ also mit 0.05. Liegt der empirische β -Wert auch innerhalb des Konfidenzintervalls, dann gilt das auch für ihn. Wir können kurz formulieren: Liegt 0 innerhalb des Konfidenzintervalls, dann ist β nicht signifikant. Liegt 0 außerhalb, dann ist β mit $p = 1 - \text{konfniv}/100$ signifikant.

3 Die Bootstrap-Eingabebox

Wir zeigen hier die Bootstrap-Box für das *Allgemeine Lineare Modell*. Die entsprechenden Eingabeboxen für die anderen Bootstrap-Programme in Almo sind weitgehend identisch mit der hier abgebildeten Eingabebox.

<input type="checkbox"/>	Loesche wieder diese Box (dann Voreinstellungen wieder gueltig)		
Option: Bootstrap			
<input type="checkbox"/>	<input type="checkbox"/>	1 = Bootstrap ausführen 0 = nicht	
<input type="checkbox"/>	<input type="checkbox"/>	wieviele Stichproben sollen gerechnet werden	<input type="button" value="Hilfe"/>
Konfidenzintervall			
<input type="checkbox"/>	<input type="checkbox"/>	Konfidenzniveau in %	<input type="button" value="Hilfe"/>
<input type="checkbox"/>	<input type="checkbox"/>	Konfidenzintervall berechnen mit	<input type="button" value="Hilfe"/>
		0 = einfachem Perzentil-Verfahren 1 = Perzentil-t-Verfahren 2 = symmetrisches Perzentil-t-Verfahren 3 = BC-Verfahren (bias corrected) 4 = BCa-Verfahren (bias corrected accelerated) 5 = Standard-basiertes Verfahren	
<input type="checkbox"/>	<input type="checkbox"/>	Startzahl Zufallsgenerator	<input type="button" value="Hilfe"/>
<input type="checkbox"/>	Behandlung von Multikollinearität in Bootstrap		<input type="button" value="Hilfe"/>
<input type="checkbox"/>	Bootstrap mit mehreren abhängigen Variablen		<input type="button" value="Hilfe"/>
<input type="checkbox"/>	Spezielle Ergebnisse ausgeben		

3.1 Die Konfidenzintervall-Verfahren

Almo berechnet die Unter- und Obergrenze des Konfidenzintervalls beim Bootstrapping durch folgende Verfahren

0. das einfache Perzentil-Verfahren. Im Eingabefeld muss eingegeben werden: 0
1. das Perzentil-t -Verfahren. Eingabe: 1
2. das symmetrische Perzentil-t -Verfahren. Eingabe: 2
3. das BC-Verfahren (BC = bias corrected). Eingabe: 3
4. das BCa-Verfahren (BCa= bias corrected and accelerated). Eingabe: 4
5. das "Standardfehler-basierte" Verfahren Eingabe: 5

Bei einigen statistischen Auswertungsverfahren ist das Perzentil-t-Verfahren in 1 und 2 nicht verfügbar

3.2 Vergleich der Konfidenzintervall-Verfahren

Wir haben das Beispiel aus Abschnitt 1 mit allen 5 Verfahren nochmals gerechnet. Das sind die Ergebnisse:

Tabelle 3.1

Abhängige Variable: Leistung

	Original- stichprobe	Bootstrap-Ergebnisse					
		Mitt.wert aus Bootstrap	Verzerr.	Standard fehler	Signifik. p	Konfidenzintervall	
						RM	Konf.niv=0.950 unten
ßorg	ßM						
einfaches Perzentil							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0010	0.6714	0.7593
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4300	-0.0433	0.0844
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0320	0.0255	0.4860
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9728	-21.8977
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8833	0.9178
Perzentil-t							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0010	0.6690	0.7560
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4340	-0.0373	0.0903
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0220	0.0368	0.4955
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9728	-21.8977
multiple Korrr. R *j)	0.9013	0.9018	0.0005	0.0087	-	0.8833	0.9178
symmetrisches Perzentil-t							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0010	0.6708	0.7573
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4340	-0.0400	0.0870
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0220	0.0305	0.4903
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9728	-21.8977
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8833	0.9178
BC-Verfahren (bias corrected)							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0010	0.6697	0.7566
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4300	-0.0458	0.0826
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0320	0.0281	0.4873
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9677	-21.8765
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8813	0.9166
BCa-Verfahren (bias corrected and accelerated)							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0010	0.6697	0.7566
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4300	-0.0456	0.0827
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0320	0.0281	0.4875
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9674	-21.8736
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8814	0.9166
Standardfehler-basiertes Verfahren							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0000	0.6700	0.7581
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4542	-0.0381	0.0851
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0264	0.0306	0.4903
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.8717	-21.8451
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8843	0.9184

Sätze zum Vergleich

1. Die Unterschiede zwischen den Verfahren sind gering. Besonders gering ist der Unterschied zwischen BC- und BCa-Verfahren. Offensichtlich ist die Schiefe der Verteilung (in unserem Beispiel) so minimal, dass sich eine Korrektur durch den "Akzelerationskoeffizienten" nicht rentiert. Andererseits ist der Unterschied in der Rechenzeit zwischen BC- und BCa-Verfahren vernachlässigbar gering, so dass nichts dagegen spricht, das kompliziertere BCa-Verfahren vorzuziehen.

Für den folgenden kurzen Textblock ist es vorteilhaft, wenn der Leser Tabelle 3.2 im nächsten Abschnitt zuvor betrachtet.

2. Das einfache Perzentil-Verfahren berechnet das Konfidenzintervall bezüglich **kMitte** (im Beispiel: 500) und nicht bezüglich **korg**. Bei einem Vertrauensniveau von 95% liegen die untere und obere Konfidenzgrenze jeweils $95/2=47,5\%$ der aufsteigend sortierten β -Werte unterhalb und oberhalb der Mitte. Die Stelle **korg** des Originalwert (in ASR) oder Mittelwert aus allen β -Werten haben keinen Einfluß auf die Lage des Konfidenzintervalls. Wenn **korg** von **kMitte** um mehrere Stellen entfernt ist, dann schätzt es die Intervallgrenzen falsch ein. In unserem Beispiel beträgt dieser "bias" 11 Stellen.

3. Das BC- und BCa-Verfahren hingegen berücksichtigt, dass **korg** aus der Mitte heraus verschoben sein kann. Es ermittelt die Intervallgrenzen korrekt bezogen auf **korg**. Das Konfidenzintervall wird verschoben. Jedoch nicht linear und proportional zum "bias" sondern in z-Werten (siehe Abschnitt 6.1).

4. Das in Abschnitt 7 dargestellte "Standardfehler-basierte" Verfahren verwendet nicht die ASR. Es unterscheidet sich dadurch prinzipiell von den anderen Verfahren. Das Konfidenzintervall wird bezüglich β_{org} berechnet. Der Abstand der Intervallgrenzen β_u bzw. β_o von β_{org} wird wesentlich durch die Standardabweichung der β -Koeffizienten aus allen Bootstrapstichproben bestimmt. Diese wird jedoch in Bezug auf β_M , den mittleren β -Wert aus allen Bootstrapstichproben, berechnet. Es wird also die "Verzerrung" $\beta_{org}-\beta_M$ nicht ausgeglichen.

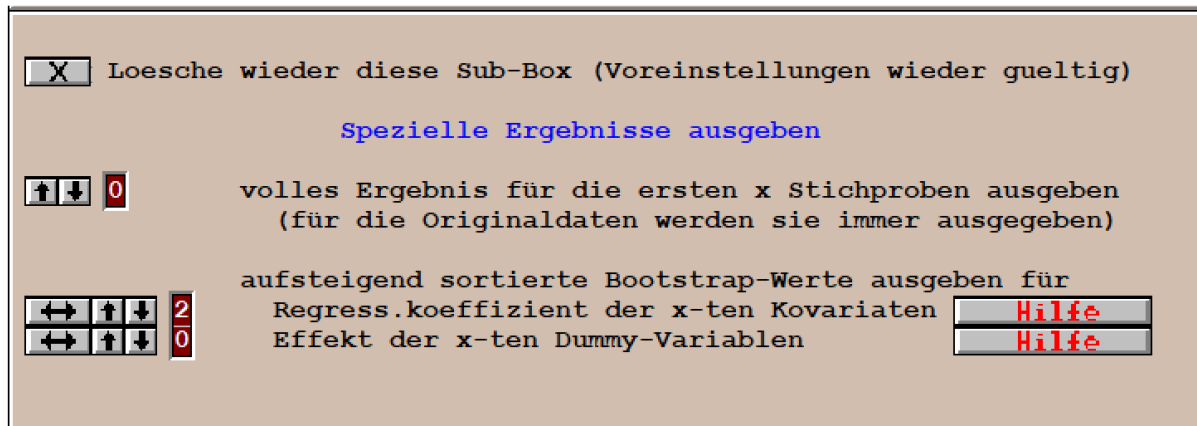
5. Liegt **korg** genau auf **kMitte**, dann sind die Intervallgrenzen **ku** und **ko** des BC- bzw. BCa-Verfahrens und des einfachen Perzentil-Verfahrens *annähernd* gleich. Wir werden in Abschnitt 6.1 und vor allem durch die Grafik in Abschnitt 6.1.1 zeigen, dass das BC- bzw. BCa-Verfahren die drei "strategischen" Stellen **korg**, **ku**, **ko** als Punkte auf der Kurve der "kumulativen Standard-Normalverteilung mit negativer Steigung" begreift. Eine exakte Gleichheit mit den drei entsprechenden k-Stellen aus dem einfachen Perzentil-Verfahren ist somit nicht möglich.

7. Liegt beim BC- und BCa-Verfahren **korg** genau auf **kMitte**, dann ist die Distanz von **kMitte** zu den beiden Intervallgrenzen **ku** und **ko** *gleich* groß. Das Konfidenzintervall **ku-kMitte-ko** ist symmetrisch.

8. Liegt beim BC- und BCa-Verfahren **korg** oberhalb von **kMitte** (z.B. ist **korg=600** von **B=1000**) dann ist die Distanz zur oberen Intervallgrenze **ko** kürzer als zur unteren. Liegt **korg** unterhalb **kMitte** dann ist umgekehrt die Distanz zu **ku** die kürzere. Diese Asymmetrie des Konfidenzintervalls ist umso stärker je weiter **korg** von **kMitte** entfernt ist.

3.3 Die aufsteigend sortierten Koeffizienten aus den Bootstrapstichproben (ASR)

Wird die letzte Sub-Box "Spezielle Ergebnisse ausgeben" geöffnet, dann sieht man folgendes



Diese Box bietet im vorletztem Eingabefeld dem Benutzer an, die aufsteigend sortierten β -Regressionskoeffizienten in der Ergebnisliste auszugeben. Wir werden für unser Beispiel die 1000 β -Regressionskoeffizienten der Variablen "Freizeit" anfordern. Wir erhalten:

Tabelle 3.2

Bootstrap-Stichprobe k	ASR aufsteigend sortierte Reihe der Bootstrap-Koeffizienten β	
1	-0.077992	
2	-0.077423	
.	.	
23	-0.044651	
24	-0.043870	
25	-0.043336	

26	-0.043098	<---- untere Konfidenzgrenze bei einfachem Perzentil $\beta_u = -0.043330$ an der Stelle $k_u = 25$
.	.	
.	.	
215	-0.000132	----- $\beta_{Null} = 0$ an der Stelle $k_{Null} = 215$
216	0.000040	
.	.	
.	.	
485	0.023105	
486	0.023185	----- Bootstrap-Mittelwert
487	0.023348	$BM = 0.023339$ an Stelle $k_M = 486$
488	0.023503	
489	0.023512	----- Originalwert in Originalstichprobe
490	0.023532	$\beta_{org} = 0.023525$ an Stelle $k_{org} = 489$
.	.	
.	.	
499	0.024101	
500	0.024138	----- Mitte an Stelle $k_{Mitte} = 500$
501	0.024251	
.	.	
.	.	
974	0.083679	
975	0.084352	----- obere Konfidenzgrenze <---- bei einfachem Perzentil $\beta_o = 0.084358$ an Stelle $k_o = 975$

976	0.084359	

.	.
998	0.107201
999	0.113112
1000	0.137050

Die Werte, die im ASR aufsteigend sortiert aufeinander folgen, sind für alle Konfidenzintervall-Verfahren gleich. Es sind die Koeffizienten, die aus den 1000 Bootstrapstichproben entstanden (Regressionskoeffizienten, Faktorladungen etc.) . Bei den beiden Perzentil-t-Verfahren allerdings sind es die Standardfehler der Koeffizienten. Wir werden das noch in Abschnitt 5 zeigen.

Auch die verschiedenen Stellen, wie **korg**, **kM**, **kNull** (mit ihren β -Werten) sind unabhängig vom Konfidenzintervall -Verfahren. Nicht jedoch die Stellen der Intervallgrenzen **ku** und **ko** (mit ihren β -Werten **β_u** und **β_o**). Sie werden entscheidend durch das gewählte Konfidenzintervall -Verfahren bestimmt.

4 Das einfache Perzentil-Verfahren

Betrachten wir die Regressionskoeffizienten β , die wir aus den 1000 Bootstrap-Stichproben für die Variable "Freizeit" aus unserem Beispiel in Abschnitt 1.1 erhalten haben. Die obere und untere Grenze des Konfidenzintervalls für das angeforderte Vertrauensniveau von 95% erhalten wir sehr einfach in folgender Weise: Vom maximalen β -Wert werden absteigend 2,5% von 1000 also 25 Werte heruntergezählt. Der dort in Position 975 stehende β -Wert ist die obere Intervallgrenze mit **$\beta_o=0.084358$** und **$ko=975$** . Entsprechend wird vom minimalen Wert ausgehend 25 Werte hinaufgezählt. Der dort in Position 25 stehende β -Wert ist der untere Grenzwert mit **$\beta_u= -0.043330$** und **$ku=25$** . Zwischen den beiden Grenzwerten befinden sich dann 95% aller Werte und außerhalb der Grenzwerte 5% aller Werte. Damit ist das 95%-Konfidenzintervall für den Koeffizienten β_{org} aus der Originalstichprobe gefunden.

4.1 Der p-Wert (Signifikanz) aus dem einfachen Perzentil-Verfahren

Der Mittelwert der 1000 Bootstrapwerte der Variablen Freizeit" **$\beta_M=0.023339$** liegt an der Stelle 486 der ASR und der Originalwert **$\beta_{org}=0.023525$** an der Stelle 489. Die Differenz wird in Almo unter der Bezeichnung *Verzerrung* ausgegeben. Beide liegen nicht auf der exakten Mitte an Stelle 500. Was wir beim BC. bzw. BCa-Verfahren in Abschnitt 6 als "bias" bezeichnen ist die Differenz von **korg** zu **kMitte**. Der Originalwert **β_{org}** liegt innerhalb des Konfidenzintervalls. Das wird fast immer so sein. Bedeutsam ist, dass **β_{Null}** an der 215. Stelle noch innerhalb des Konfidenzintervalls liegt.

Wir finden also folgende Situation vor:

(1) Der "wahre" Populationswert und (2) der Wert 0 und (3) der originale Regressionskoeffizient **β_{org}** liegen gemeinsam im Konfidenzintervall. Das bedeutet, dass **β_{org}** mit einer Wahrscheinlichkeit von 95% gleich 0 sein kann.

Bedeutsam ist es jedoch, wenn es gelingt, unabhängig vom vorgegebenen Vertrauensniveau, aus der Position von 0 in ASR den genauen p-Wert zu bestimmen.

Den Wert 0 für einen Koeffizienten als Testwert zu verwenden, ist nur dann sinnvoll

1. wenn der Koeffizient prinzipiell positive und negative Werte annehmen kann. Das ist der Fall beim Regressionskoeffizienten beim ALM und der Logit- und Probit-Analyse, dem Haupt-

und Interaktions-Effekt beim ALM, dem Korrelationskoeffizienten und der Faktorladung - aber z.B. nicht beim multiplen Korrelationskoeffizient R, der nur ≥ 0 sein kann.

2. wenn der Wert 0 des Koeffizienten in dem Sinn interpretierbar ist, dass *die Variable, auf die der Koeffizient hinweist, wirkungslos ist*. In unserem Beispiel ist das gegeben. Ein Regressionskoeffizient von 0 bei der Variablen "Freizeit" bedeutet, dass diese (unabhängige) Variable keinen Erklärungswert für die abhängige Variable "Leistung" besitzt.

Möglich wäre es auch als "Testwert" anstelle von 0 einen anderen Wert zu verwenden. Almo setzte beispielsweise beim Bootstrap der Eigenwerte bei der Faktorenanalyse als Testwert 1.0 ein (um zu überprüfen, ob ein Faktor noch das Kaiser-Kriterium erfüllt). Dieser Testwert wird in den aufsteigend sortierten Eigenwerten aus den Bootstrapstichproben gesucht.

Die *zweiseitige* Signifikanz (der p-Wert) ergibt sich beim einfachen Perzentil-Verfahren in folgender Weise:

Wir verwenden folgende Notation

B = Zahl der Bootstrapstichproben
ASR = aufsteigend sortierten Reihe der β -Werte aus den Bootstrapstichproben
kNull = Stelle in der aufsteigend sortierten Reihe ASR, an welcher der β -Wert=0 zum 1. Mal überschritten wird
kMitte= Mitte von ASR (=B/2=500)

Wir bestimmen die Stelle **kNull**, in der aufsteigend sortierten Reihe der β -Werte, an der der β -Wert 0 zum 1. Mal überschritten wird. Es gilt

Liegt **kNull** unterhalb der **kMitte** von ASR (das ist B/2 = 500) dann ist die Signifikanz

$$(1) \quad p = 2 * (kNull - 1) / B$$

In unserem Beispiel ist das der Fall. In obiger Tabelle 3.2 erkennen wir, dass **kNull** zwischen den Stellen 215 und 216 liegt; **kNull** ist also =216. Somit ist $p = 2 * (216 - 1) / 1000 = 0.4300$

Liegt **kNull** oberhalb der Mitte dann ist

$$(2) \quad p = (B - kNull + 1) / B$$

Beide Formeln können auch so zusammengefasst werden

$$(3) \quad p = 1 - 2 * \text{abs}(kMitte - kNull + 1) / B$$

abs = Absolutwert

Bemerkenswert ist, dass der p-Wert nur durch den Abstand von **kNull** zu **kMitte** bestimmt wird und nicht durch den Abstand von **kNull** zu **korg**, der k-Stelle, an der **borg** liegt.

Wie soll verfahren werden, wenn die aufsteigend sortierten Werte keinen Wert 0 aufweisen bzw. keinen Übergang von negativen Werten zu positiven (oder umgekehrt) enthalten - obwohl das im Prinzip möglich wäre. Das ist dann der Fall, wenn die jeweilige unabhängige Variable die abhängige Variable stark beeinflusst, d.h. wenn der Effekt bzw. Regressionskoeffizient einen großen positiven oder (bei gegenläufigem Einfluss) großen negativen Wert besitzt. Auch wenn sehr viele Bootstrapstichproben gerechnet werden, tritt

keine auf, die für die Variable den Wert 0 oder sogar einen Wert jenseits von 0 aufweist. Das bedeutet, dass die Variable *hoch signifikant* wirkt. In dieser Situation muss der ungünstigste Fall unterstellt werden, dass gerade unterhalb bzw. oberhalb der aufsteigend sortierten Werte der Wert 0 folgen würde - hätte man eine weitere Stichprobe gerechnet. Also berechnet somit das "optimale" Vertrauensniveau für ein Konfidenzintervall, dessen unterer Grenzwert der erste bzw. niedrigste Wert in der Sortierfolge ist und dessen oberer Grenzwert der letzte bzw. höchste Wert ist. Die zweiseitige Signifikanz ist dann sehr einfach $p=1/(Stichprobenzahl)$. Die Signifikanz der Variablen kann dann nur gleich diesem p-Wert oder besser (d.h. kleiner) sein. Sie ist nur durch die Stichprobenzahl bestimmt.

5 Das Perzentil-t -Verfahren

Wir verwenden folgende Notation:

β_{org} = Regressionskoeffizient für Variable i aus der originalen Stichprobe
 S_{org} = Standardfehler von β_{org}
 t_{org} = t-Wert für Variable i aus originaler Stichprobe

β = Regressionskoeffizient für Variable i aus den Bootstrap-Stichproben
 S = Standardfehler von β
 t = t -Wert der Variablen i aus den Bootstrap-Stichproben

$konfniv$ = Vertrauensniveau (Beispiel: 95)

α = $1-konfniv/100$ (Beispiel: = 0.05)
 $\alpha/2$ = $(1-konfniv/100)/2$ (Beispiel: = 0.25)
 $1-\alpha/2$ = $0.5+konfniv/200$ (Beispiel: = 0.975)

B = Zahl der Bootstrap-Sichproben

Für jede der 1000 Bootstrap-Stichproben muss der β -Wert und sein ihm zugehöriger Standardfehler S erhoben werden. Aus den beiden und dem β_{org} -Wert aus der originalen Stichprobe wird ein t -Wert gebildet

$$(1) \quad t = (\beta - \beta_{org}) / S$$

Beachte: $(\beta - \beta_{org})$ ist die "Verzerrung". Zu beachten ist auch, dass der t -Wert negativ werden kann

Nach Ablauf des Bootstraps verfügen wir also für die Variable i über 1000 t-Werte, die gemäß Gleichung (1) gebildet wurden. Diese Koeffizienten werden, wie beim einfachen Perzentil-Verfahren, in eine aufsteigend sortierte Reihe gebracht. Wir geben ein Vertrauensniveau von 95% vor. Entsprechend wird die Stelle 25+1 und die Stelle 975 herausgegriffen. Dort stehen die Grenzwerte des Konfidenzintervalls. An diesen Stellen stehen aber t-Werte und nicht β -Werte. Und diese sind es, die wir für die Konfidenz-Untergrenze und Obergrenze brauchen. Den zum jeweiligen t-Wert zugehörigen β -Wert erhält man gemäß folgender Gleichung

$$(2) \quad \beta' = \beta_{org} - S_{org} * t$$

β_{org} = das ist der Regressionseffekt der Variablen i aus der originalen Stichprobe

S_{org} = das ist dessen Standardfehler

β' = errechneter Regressionskoeffizient für Variable i aus den Bootstrap-stichproben

Es wäre auch naheliegend anstelle des Standardfehlers S_{org} aus der Originalstichprobe, den Standardfehler des Bootstrap-Mittelwerts in Gleichung (2) einzusetzen. Dieser entsteht aus den Standardabweichungen der einzelnen Standardfehler aller Bootstrapstichproben. Siehe dazu den Beitrag auf der Internetseite bei "R bloggers" mit Suchwort "understanding bootstrap confidence interval output from the r boot package"

Die untere Konfidenzgrenze ist (3a) $\beta_u = \beta_{org} - S_{org} * t_u$
 und die obere (3b) $\beta_o = \beta_{org} - S_{org} * t_o$

B = Zahl der Bootstichproben

t_o = t-Wert für die obere Konfidenzgrenze an der Stelle

$B * \alpha / 2 + 1$
 bei konfniv 95%: $1000 * 0.05 / 2 + 1 = 25 + 1 = 26$
 Siehe folgende 5

t_u = t-Wert für die untere Konfidenzgrenze an der Stelle

$B * (1 - \alpha / 2)$
 bei konfniv 95%: $1000 * (1 - 0.05 / 2) = 975$.
 Siehe folgende Tabelle 5

β_o = obere Konfidenzgrenze (errechnet gemäß Gleichung 2)
 β_u = untere Konfidenzgrenze (errechnet gemäß Gleichung 2)

In unserem Beispiel finden wir für die Variable "Freizeit" und ein vorgegebenes Vertrauensniveau von **konfniv=95%**

$\beta_{org} = 0.023525$ (Regress.koeff. aus Originalstichprobe)
 $S_{org} = 0.033842$ (Standardfehler aus Originalstichprobe)
 $t_u = 1.796678$ (t-Wert an Stelle 975. Siehe Tabelle 5)
 $t_o = -1.973695$ (t-Wert an Stelle 26. Siehe Tabelle 5)

Eingesetzt in Gleichung 3a und 3b erhalten wir

$\beta_u = 0.023525 - 0.033842 * 1.796678 = -0.037277$
 $\beta_o = 0.023525 - 0.033842 * -1.973695 = 0.090318$

Hier ist die gekürzte Reihenfolge der t- und β -Werte für die Variable "Freizeit"

Spalte Nr_t = Nummer der Bootstrapstichproben der aufsteigen sortierten t-Werte
 Spalte t-Wert = das sind die aufsteigend sortierten t-Werte gemäß Gleichung (1)
 Spalte β' = das sind die den t-Werten zugehörigen β -Werte. Sie wurden gemäß Gleichung (2) berechnet
 Spalte Nr_ β' = den zunehmenden t-Werten entsprechen abnehmende β -Werte. Entsprechend werden die β -Werte invers zu den t-Werten nummeriert

Tabelle 5

Nr_t	t-Wert	Nr_ β'	β'
1	-3.195921	1000	0.131680
2	-2.887321	999	0.121237
.	.	.	.
.	.	.	.
23	-2.082307	978	0.093994
24	-1.992466	977	0.090953
25	-1.979533	976	0.090516
26	-1.973695 <-- t_o	975	0.090318 <---- obere Konfidenzgrenze $\beta_o = 0.0903$
.	.	.	.
.	.	.	.
489	-0.000376	512	0.023538

490	0.000209	511	0.023518	<---- Originalwert $\beta_{org} = 0.023525$
491	0.001245	510	0.023483	
.	.	.	.	
495	0.003253	506	0.023415	
496	0.006739	505	0.023297	
497	0.007754	504	0.023263	<---- Bootstrap-Mittelwert $\beta_M = 0.23339$
498	0.014186	503	0.023045	
499	0.016731	502	0.022959	
500	0.019854	501	0.022853	<---- Median (Mittel)
501	0.020874	500	0.022819	
.	.	.	.	
783	0.691708	218	0.000117	
784	0.692519	217	0.000089	_____ $\beta_{Null} = 0$
785	0.700355	216	-0.000176	
.	.	.	.	
974	1.764826	27	-0.036199	
975	1.796678	26	-0.037277	_____ untere Konfidenzgrenze $\beta_u = -0.0373$
976	1.799305	25	-0.037366	
977	1.829012	24	-0.038371	
.	.	.	.	
999	2.597386	2	-0.064374	
1000	3.119338	1	-0.082038	

5.1 Der p-Wert aus dem Perzentil-t -Verfahren

Die Perzentil-t -Werte t aus den Bootstrapstichproben für die unabhängige Variablen i (in unserem Beispiel: die Variable "Freizeit") werden quadriert. Wir bezeichnen sie mit t^2 . Ebenso wird der t-Wert aus der originalen Stichprobe quadriert. Wir bezeichnen ihn mit t_{org}^2 .

Dann wird gezählt: Wie oft ist t^2 größer/gleich t_{org}^2 . Wir bezeichnen das Zählergebnis mit h .

Die Signifikanz p ist dann $(4) p = h/B$

In unsere Beispiel wird für die Variable "Freizeit" festgestellt, dass t^2 434 mal größer/gleich ist als t_{org}^2 . Daraus folgt ein Wert von $p = 0.434$

Im Vergleich dazu wurde mit dem einfachen Perzentil-Verfahren $p=0.430$ ermittelt.

Ist $h=0$ dann wird gerechnet (wie beim einfachen Perzentil-Verfahren) $p = 1/(B+1)$
 In diesem Fall muss interpretiert werden, dass der tatsächliche p-Wert mindestens $1/(B+1)$ ist oder kleiner, d.h. signifikanter. p ist dann nur durch die Stichprobenzahl B bestimmt.

5.2 Das symmetrische Perzentil-t-Verfahren

"Symmetrisch" bedeutet: Der Regressionskoeffizienten β_{org} aus der Originalstichprobe liegt genau in der Mitte zwischen der Untergrenze β_u und der Obergrenze β_o

Der oben definierte t -Wert wird absolut gesetzt

$$t^* = \text{abs}(t)$$

Das Vertrauensniveau sei 95%. Die t^* -Werte werden aufsteigend sortiert. Man beachte, dass die dabei entstehen Reihenfolge eine andere ist, als die in Tabelle 5 ausgegebene.

Der t*-Wert an der Stelle

$$B * (\text{konfniv}/100) = B * (1-\alpha) = 1000 * 0.95 = 950.$$

wird herausgegriffen. Wir bezeichnen ihn mit t_{org}^* . Für die Variable "Freizeit" finden wir an der Stelle 950 $t_{org}^*=1.87683$

Die untere Konfidenzgrenze ist dann $\beta_u = \beta_{org} - S * t_{org}^*$
 $= 0.023525 - 0.033842 * 1.87683 = -0.0400$

und die obere $\beta_o = \beta_{org} + S * t_{org}^*$
 $= 0.023525 + 0.033842 * 1.87683 = 0.087041$

Das Intervall liegt symmetrisch um β_{org} , den Regressionskoeffizienten aus der Originalstichprobe. D.h. $\beta_{org}=0.023525$ liegt genau in der Mitte zwischen der Untergrenze $\beta_u=-0.0400$ und der Obergrenze $\beta_o=0.087041$

6 Das BC- und BCa-Verfahren

BC steht für "bias corrected" und BCa für "bias corrected and accelerated".

Ein "bias" existiert, wenn k_{org} ungleich $B/2$ ist, d.h. wenn der Originalwert nicht auf der Mitte von ASR liegt. In diesem Fall wird das Konfidenzintervall verschoben oder anders formuliert: Das Konfidenzintervall wird auf die ASR-Stelle des Originalwerts k_{org} ausgerichtet. Die Verschiebung erfolgt nicht linear sondern in z-Werten. Wir werden das im Abschnitt 6.1 "Kalkül" detailliert ausführen.

6.0 Notation und Übersicht

Wir verwenden folgende Notation: (siehe auch Notation in Abschnitt 1.1 und 6.3.1)

B = Zahl der Bootstrapstichproben

ASR = aufsteigend sortierten Reihe der β -Koeffizienten aus den
 B Bootstrapstichproben

Mitte = Mitte von ASR $=B/2$

β = Koeffizient aus Bootstrapstichprobe i

β_M = Mittelwert der β -Werte aus allen Bootstrapstichproben

β_{org} = β -Koeffizient aus Originalstichprobe
(kurz: originaler β -Koeffizient)

k_{org} = das ist die Stelle k , an der β_{org} in ASR erstmalig überschritten wird minus 1. Also: $k_{org}=k-1$
= Zahl der β -Koeffizienten, die in ASR kleinergleich β_{org} sind

p_{org} = k_{org}/B . Das ist die Stelle des originalen β -Koeffizienten in ASR als Anteilswert bzw. p-Wert ausgedrückt

z_{org} = das ist p_{org} als z-Wert ausgedrückt

β_u = β -Koeffizient der unteren Konfidenzgrenze

β_o = β -Koeffizient der oberen Konfidenzgrenze

k_u = Stelle der unteren Konfidenzgrenze in ASR

k_o = Stelle der oberen Konfidenzgrenze in ASR

konfniv = Vertrauensniveau (Beispiel: 95)

$\alpha = 1 - \text{konfniv}/100$ (Beispiel: = 0.05)
 $\alpha/2 = (1 - \text{konfniv}/100)/2$ (Beispiel: = 0.025)
 $1 - \alpha/2 = 0.5 + \text{konfniv}/200$ (Beispiel: = 0.975)

$\text{ekonfniv} = \text{effektives Vertrauensniveau}$
 $\text{ealpha} = 1 - \text{ekonfniv}/100$
 $\text{ealpha}/2 = (1 - \text{ekonfniv}/100)/2$
 $1 - \text{ealpha}/2 = 0.5 + \text{ekonfniv}/200$

$\text{zorg} = \text{z-Wert des Originalwerts} - \text{genauer: } z = \text{z-Wert von } \text{korg}/B$
 $\text{zu} = \text{z-Wert von unterer Intervallgrenze}$
 $\text{zo} = \text{z-Wert von oberer Intervallgrenze}$
 $\text{z1} = \text{z-Wert von } \alpha/2$
 $\text{z2} = \text{z-Wert von } 1 - \alpha/2$

6.0.1 Übersicht

Wir werden im folgenden den Kalkül des BC-Verfahrens am Beispiel der Variablen "Freizeit" aus obiger Tabelle illustrieren und danach die Erweiterung durch das BCa-Verfahren darstellen. Die Erweiterung besteht lediglich darin dass (allerdings etwas aufwendig) ein Akzelerationskoeffizient errechnet wird, der dann in die Formel des BC-Verfahrens an einer einzigen Stelle eingesetzt wird.

Die Signifikanz p für β_{org} , den β -Wert aus der Originalstichprobe, wird dem BC- bzw. BCa-Verfahren angepasst. Siehe den folgenden Abschnitt 6.2. Sie wird nicht nach dem einfachen Perzentil-Verfahren, wie in Abschnitt 4.1, berechnet.

Das BCa-Verfahren erweitert das BC-Verfahren. Eine eventuell bestehende Schiefe der Verteilung wird korrigiert. Also liefert folgendes Ergebnis aus dem BC- und dem BCa-Verfahren (gekürzt). Man erkennt, dass die Unterschiede minimal sind. Offensichtlich ist in unserem Beispiel die Schiefe sehr gering.

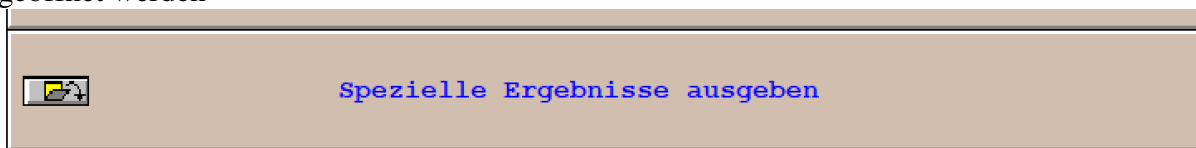
Tabelle 6.1

Abhängige Variable: Leistung

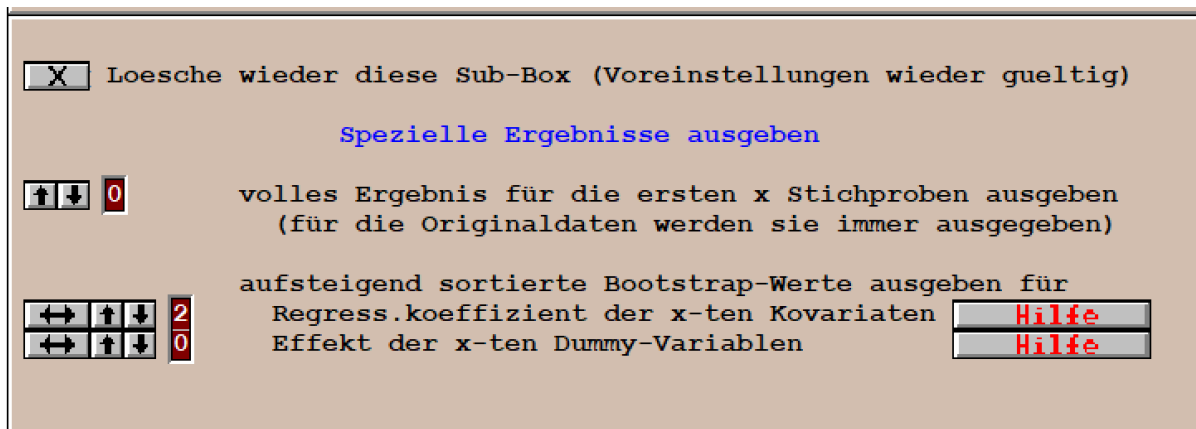
	Original- stichprobe	Bootstrap-Ergebnisse					
		Mitt.wert Bootstrap BM	Verzerr.	Standard fehler	Signifik. p	Konfidenzintervall Konf.niv=95%	
	β_{org}					unten	oben
BC-Verfahren							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0027	0.6697	0.7566
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4629	-0.0458	0.0826
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0304	0.0281	0.4873
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9677	-21.8765
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8813	0.9166
BCa-Verfahren							
1 Alter	0.7141	0.7151	0.0011	0.0225	0.0027	0.6697	0.7566
2 Freizeit	0.0235	0.0233	-0.0002	0.0314	0.4628	-0.0456	0.0827
3 Urlaub	0.2604	0.2565	-0.0040	0.1173	0.0303	0.0281	0.4875
Konstante	-23.3584	-23.3885	-0.0301	0.7719	0.0010	-24.9674	-21.8736
multiple Korrr. R	0.9013	0.9018	0.0005	0.0087	-	0.8814	0.9166

Der Regressionskoeffizient aus der Originalstichprobe ($\beta_{org}=0.023525$) und der Mittelwert der β -Koeffizienten aus den 1000 Bootstrapstichproben ($\beta_M=0.023339$) differieren. Bedeutsamer ist, dass β_{org} nicht auf der Stelle **Mitte** liegt, d.h. nicht in der "Mitte" (an der Stelle 500) der ASR plaziert ist (was allerdings aus obiger Ausgabe nicht sichtbar wird). Das ist der "bias". Wird das Konfidenzintervall errechnet, dann wird dieser "bias" von Almo-BCa-Programm automatisch korrigiert. Das Konfidenzintervall wird verschoben. Jedoch nicht linear und proportional zum "bias" sondern in z-Werten (siehe Abschnitt 6.1).

Betrachten wir zuerst einen Ausschnitt aus ASR, der aufsteigend sortierten Reihe der β -Koeffizienten für die Variable "Freizeit". Um diese zu erhalten muss in unserem Beispiel-Programm Prog20BS in der Bootstrap-Optionsbox die Sub-Box "spezielle Ergebnisse" geöffnet werden



Man sieht dann



Almo gibt dann für die 2. Variable (Freizeit) die ASR aus.

Wir haben die Ergebnisse aus dem einfachen Perzentil-Verfahren, dem BC-Verfahren und dem BCa-Verfahren in die ASR eingetragen. Alle drei wurden mit derselben Zufalls-Startzahl gerechnet, so dass die Reihenfolge der β -Werte jedes Mal dieselbe war.

Tabelle 6.2

Bootstrap-Stichprobe k	ASR einfaches Perzentil BC-Verfahren BCa-Verfahren β	
1	-0.077992	
2	-0.077423	
.	.	
.	.	
20	-0.048762	
21	-0.048043	_____ untere Konfidenzgrenze β_u
22	-0.045371	von BC = -0.045768
23	-0.044651	von BCa= -0.045561
22	-0.045371	
23	-0.044651	
24	-0.043870	
25	untere Konfidenzgrenze _____ -0.043336	
26	bei einfachem Perzentil= -0.043098	
.	.	
.	.	
215	-0.000132	_____ $\beta_{Null} = 0$
216	0.000040	

.	.		
.	.		
485	0.023105		
486	0.023185	_____	Bootstrap-Mittelwert
487	0.023348		BM =0.023339
488	0.023503		
489	0.023512	_____	Originalwert
490 <----	0.023532		Borg =0.023525
.	.		
.	.		
499	0.024101		
500	0.024138	-----	Mitte
501	0.024251		
.	.		
.	.		
972	0.081971	_____	obere Konfidenzgrenze β_0
973	0.083442		von BC =0.082584
974	0.083679		von BCa=0.082721
975	0.084352	_____	obere Konfidenzgrenze
976	0.084359		bei einfachem
.	.		Perzentil=0.084358
.	.		
998	0.107201		
999	0.113112		
1000	0.137050		

Borg liegt in unserem Beispiel näher bei dem Bootstrap β -Koeffizient **0.023532** an der Stelle 490 als bei **0.023512** an der Stelle 489, so dass auch 490 als die (gerundete) Stelle von **Borg** in ASR verstanden werden kann. An Stelle 490 steht somit der Bootstrap β -Koeffizient der **Borg** überschreitet und zugleich ist dies die Stelle von **Borg**.

Bei dem nicht in die Tabelle eingetragenen Perzentil-t-Verfahren ergab sich als

Konfidenz-Untergrenze -0.037277 an der Stelle in ASR: 34
und als Obergrenze 0.090318 985

und beim symmetrischen Perzentil-t-Verfahren

Konfidenz-Untergrenze -0.039990 an der Stelle in ASR: 29
und als Obergrenze 0.087040 978

Beim Perzentil-t-Verfahren wird nicht der Regressionskoeffizient β aus den Bootstrapstichproben in eine aufsteigend sortierte Reihe ASR gebracht, sondern der oben in Abschnitt 5 beschriebene Koeffizient t . Das Perzentil-t-Verfahren liefert ein nur ungefähr gleiches Ergebnis wie die anderen Verfahren

In Tabelle 6.2 erkennt man folgendes:

Die Mitte von ASR ist an Stelle	Mitte =500
korg liegt auf	korg =489
der "bias" beträgt	= 11 Stellen
die Konfidenz-Untergrenze liegt auf k_u	= 22
die Konfidenz-Obergrenze liegt auf k_o	=972
das Konfidenzintervall umfasst	=950 Stellen
das entspricht dem vorgegebenen Vertrauensniveau von	= 95
das Konfidenzintervall ist um	= 3 Stellen nach unten verschoben

der untere Teil des Konfidenzintervalls von k_{org} bis k_u umfasst =469 Stellen

der obere Teil des Konfidenzintervalls von k_{org} bis k_o umfasst =483 Stellen
 das Konfidenzintervall ist nicht symmetrisch

Pauschal zusammengefasst:

1. Die Konfidenzgrenzen beim einfachen Perzentil-Verfahren schließen je 2.5% der β -Werte unten und oben aus und 95% ein. Durch die Wahl des Vertrauensniveaus mit 95% wird das erzwingen. Beim BC- und BCa-Verfahren werden ebenfalls 95% der β -Werte eingeschlossen, aber unten und oben ungleich viele β -Werte, jedoch zusammen 5%, ausgeschlossen.
2. Im Vergleich zum einfachen Perzentil-Verfahren befindet sich bei BC und BCa die untere Konfidenzgrenze an Stelle 21/22 anstatt 26 und die obere Konfidenzgrenze an Stelle 972/973 anstatt 975. Das Konfidenzintervall ist um 3 bzw. 4 Stellen nach unten verschoben.
3. Der β_{org} -Koeffizient aus der Originalstichprobe befindet sich bei allen 3 Verfahren innerhalb des Konfidenzintervalls. Daraus folgt: Der β_{org} -Wert aus der Originalstichprobe befindet sich in demselben Intervall, in dem sich der "wahre" Populationswert mit einer Irrtumswahrscheinlichkeit von 5% befindet
4. Aber der β_{Null} -Koeffizient befindet sich ebenfalls im Intervall. Das bedeutet, dass der Populationswert und der empirische β_{org} -Wert mit einer Irrtumswahrscheinlichkeit von 5% ebenfalls 0 sein können.
5. Den "bias" erkennt man daran, dass der originale Regressionskoeffizient β_{org} an der Stelle 489 aus der Mitte von ASR an Stelle 500 um 11 Stellen verschoben ist. Die kombinierte Wirkung von "bias" und Schiefe im BCa-Verfahren wird auch noch daraus ersichtlich, dass der Mittelwert β_M aus den Bootstrapstichproben-Koeffizienten weder auf k_{org} noch auf **Mitte** liegt

6.1 Der Kalkül

In folgenden Schritten wird nun das BC- und das BCa-Verfahren gerechnet

1. In der aufsteigend sortierten Reihe ASR der β -Koeffizienten aus den 1000 Bootstrapstichproben wird die Stelle k gesucht, an welcher der β_{org} -Wert aus der Originalstichprobe zum ersten Mal überschritten wird. Dann kann die Stelle k_{org} von β_{org} auf folgende Weise festgelegt werden:
 - a. k_{org} ist die Stelle k an der β_{org} erstmalig überschritten wird minus 1. Also: $k_{org}=k-1$
 - b. In der Regel ist β_{org} nicht exakt gleich dem β -Wert an der Stelle $k-1$ sondern liegt zwischen den β -Werten an der Stelle $k-1$ und k . Dann wird k_{org} auf $k-1$ oder k gesetzt - je nachdem welcher β -Wert näher bei β_{org} liegt. Wir konnten oben in Tabelle 6.2 diesen Fall für unser Beispiel feststellen.

Wir entscheiden uns im folgenden für die einfachere Version aus Punkt a.

In unserem Beispiel ist dann $k_{org}=490-1$. Es wird folgender *Anteilswert* p_{org} gebildet, der in einen z-Wert umgerechnet wird

$$\begin{aligned} (1) \quad p_{org} &= k_{org} / (B+1) &&= (490-1) / 1001 = 0.488511 \\ (2) \quad z_{org} &= z_{von_p}(p_{org}) &&= 0.028810 \end{aligned}$$

Anmerkung: Dass im Nenner von (1) B um $+1$ vergrößert wird, hat primär den Zweck zu verhindern, dass p_{org} exakt 1.0 werden kann und

zorg gleich unendlich und eventuell das Programm abstürzen kann. Die Folge ist allerdings, dass dadurch störende Ungenauigkeiten entstehen können. Almo rechnet jetzt nicht mehr mit B+1 im Nenner sondern nur B.

korg kann interpretiert werden als die Zahl der der Bootstrap- β -Werte, die kleiner als der Originalwert β_{org} aus der Originalstichprobe sind. porg ist dann der Anteilswert an der Gesamtzahl B aller Bootstrapstichproben.

Die Funktion "z_von_p" in Formel 2 ist eine Funktion, die aus der "inversen kumulativen Standard-Normalverteilung (mit negativer Steigung)" den z-Wert von p (in unserem Fall: von porg) ausgibt. Der Begriff "invers" meint "umgekehrt". Es wird die Umkehrfunktion der kumulativen Standard-Normalverteilung (mit negativer Steigung) gebildet.

In Exkurs B wird "z_von_p" und die noch darzustellende inverse Funktion "p_von_z" als C-Programm dargestellt. Im Almo-Ordner "Algorithmen_in_C/Algorith_C/a_up_algorithm2.c" sind die beiden Funktionen ebenfalls in der Programmiersprache C enthalten.

Der Benutzer kann unsere Berechnungen leicht nachrechnen. Er muss nur in Almo auf den Knopf „stat.Tafelwert“ (am Oberrand des Almo-Hauptfensters) klicken und dann Pro00t3.Msk laden. Das Programm verwendet die Funktion "z_von_p". Dem Benutzer wird folgende Programm-Maske präsentiert, die wir hier mit porg=0.488511 ausgefüllt haben

Prog00t7.Msk

Gesucht: z
 Eingabe: p=0.488511 einseitig
 Ergebnis: z-Wert = 0.028804

Im Exkurs A am Ende des Handbuchs werden wir "z_von_p" und die "inverse kumulative Standard-Normalverteilung ausführlicher darstellen.

2. Auch das Vertrauensniveau konfniv=95% bzw. alpha wird in zwei z-Wert umgerechnet

$$(3) z_1 = z_{\text{von_p}}(\alpha/2) = z_{\text{von_p}}(0.025) = 1.962339$$

$$(4) z_2 = z_{\text{von_p}}(1-\alpha/2) = z_{\text{von_p}}(0.975) = -1.962339$$

Zum Parameter konfniv und alpha siehe Notation in Abschnitt 6.0. Wir wiederholen:

$$\begin{aligned} \text{konfniv} &= \text{Vertrauensniveau} && (\text{Beispiel: } 95) \\ \alpha &= 1-\text{konfniv}/100 && (\text{Beispiel: } = 0.05) \\ \alpha/2 &= (1-\text{konfniv}/100)/2 && (\text{Beispiel: } = 0.025) \\ 1-\alpha/2 &= 0.5+\text{konfniv}/200 && (\text{Beispiel: } = 0.975) \end{aligned}$$

Einschub 1:

Bei den Bootstrap-Programmen in Abschnitt 6.3 zu den Basis-Statistiken (Mittelwert, Median, Quartile, Häufigkeiten) in den Programmen Prog05m7, Prog05m8, Prog05m9 müssen z1 und z2 aus Gleichung 3 und 4 erweitert werden.

Damit die folgenden Ausführungen verständlich werden, sollte man zuerst weiter unten Einschub 2 lesen.

$$\begin{aligned} z_a &= z_{\text{von_p}}(\alpha/2) - z_{\text{von_p}}(\text{ealpha}/2) \\ z_b &= z_{\text{von_p}}(1-\alpha/2) - z_{\text{von_p}}(1-\text{ealpha}/2) \\ z_1 &= z_1 + z_a \\ z_2 &= z_2 + z_b \end{aligned}$$

z_a ist die Differenz der z-Werte des vorgegebenen Vertrauensniveaus "konfniv" minus dem effektiven Vertrauensniveau "ekonfniv" das unten in Gleichung 13 entsteht. z_a und z_b müssen zu z₁ und z₂ hinzugefügt werden.

Die beiden erweiterten Gleichungen für z₁ und z₂ können auch so ausgedrückt werden

$$\begin{aligned} (3.1) \quad z_1 &= 2 * z_{\text{von_p}}(\alpha/2) - z_{\text{von_p}}(\text{ealpha}/2) \\ (4.1) \quad z_2 &= 2 * z_{\text{von_p}}(1-\alpha/2) - z_{\text{von_p}}(1-\text{ealpha}/2) \end{aligned}$$

3. Beim **BC-Verfahren** werden die z-Werte aus Originalwert und Vertrauensniveau in folgender Weise addiert:

$$\begin{aligned} (5) \quad z_u &= 2 * z_{\text{org}} + z_1 = 2 * 0.028810 + 1.962339 = 2.019956 \\ (6) \quad z_o &= 2 * z_{\text{org}} + z_2 = 2 * 0.028810 - 1.962339 = -1.904722 \end{aligned}$$

z_u = z-Wert von unterer Intervallgrenze

z_o = z-Wert von oberer Intervallgrenze

(siehe dazu die grafische Darstellung in Abschnitt 6.1.1)

Beim **BCa-Verfahren** wird zusätzlich zur "bias"-Korrektur eine eventuelle Schiefe der Verteilung korrigiert. Dazu müssen die Gleichung (5) und (6) durch den "Akzelerations"-Koeffizient **a** erweitert werden, der die Schiefe korrigiert.

$$\begin{aligned} (5a) \quad z_u &= z_{\text{org}} + (z_{\text{org}} + z_1) / (1 - a * (z_{\text{org}} + z_1)) \\ (6a) \quad z_o &= z_{\text{org}} + (z_{\text{org}} + z_2) / (1 - a * (z_{\text{org}} + z_2)) \end{aligned}$$

Bei **a=0** werden (5a) und (6a) zu obigen Gleichungen (5) und (6). Das ist der einzige Unterschied zwischen BC- und BCa-Verfahren. Dann läuft der Kalkül der beiden Verfahren gemeinsam weiter, so wie in den folgenden Gleichungen 7 bis 13. Wie der Akzelerationskoeffizient **a** gewonnen wird, zeigen wir im nächsten Abschnitt 6.1.2. Dort errechnen wir für

$$a = -0.00038236$$

Mit diesem a-Wert wollen wir unser Datenbeispiel mit dem BCa-Verfahren weiter rechnen. Für z_u und z_o erhalten wir

$$\begin{aligned} (5a) \quad z_u &= z_{\text{org}} + (z_{\text{org}} + z_1) / (1 - a * (z_{\text{org}} + z_1)) \\ &= 0.028810 + (0.028810 + 1.962339) / (1 - -0.00038236 * (0.028810 + 1.962339)) \\ &= 2.018442 \end{aligned}$$

$$\begin{aligned} (6a) \quad z_o &= z_{\text{org}} + (z_{\text{org}} + z_2) / (1 - a * (z_{\text{org}} + z_2)) \\ &= -1.906152 \end{aligned}$$

4. z_u und z_o werden zu p-Werten mit der Funktion 'p_von_z' zurückgerechnet

$$(7) \quad p_u = p_{\text{von_z}}(z_u) = p_{\text{von_z}}(2.018442) = 0.021906$$

$$(8) p_o = p_von_z(z_o) = p_von_z(-1.906152) = 0.971542$$

Der **pu**-Wert sagt aus, dass ein Anteil von **0.021906** also **2.1906%** der 1000 β -Werte in ASR kleiner sind als der untere Intervallgrenzwert und **po**, dass **97.1542%** kleiner als der obere Intervallgrenzwert sind. Wir müssen jetzt nur noch nachschauen welcher β -Wert sich an der Stelle 21,906 (gerundet: 22) und an der Stelle 971,542 (gerundet: 972) befindet. Das sind dann die Grenzwerte des Konfidenzintervall.

"**p_von_z**" ist eine Funktion, die aus der **kumulativen Standard-Normalverteilung** mit negativer Steigung nach Eingabe eines z-Wertes den p-Wert ausgibt. Siehe Exkurs am Ende des Handbuchs.

In Almo ist im Ordner "**Algorithmen_in_C/Algorith_C/a_up_algorith2.c**" diese Funktion in der Programmiersprache C enthalten. Im folgenden Abschnitt 6.1.1 werden wir die Kurve dieser Funktion abbilden. Verwendbar ist auch die in Almo enthaltene Funktion "**p_von_t**".

Der Benutzer kann unsere Berechnungen leicht nachrechnen. Er muss nur in Almo auf den Knopf „stat.Tafelwert“ (am Oberrand des Almo-Hauptfensters) klicken und dann **Prog00t3.Msk** (oder auch Prog00t2.Msk) laden. Dieses Programm verwendet die Funktion "**p_von_z**".

Prog00t3.Msk

Signifikanz von z
gegeben z-Wert; gesucht p

↔

2.01842

zu testender z-Wert

Eingabe: z = 2.01842

Ergebnis: p = 0.02179 (einseitig)

Wichtig ist festzuhalten: Der *Input* in die Funktion **p_von_z** ist **z**. Der *Output* ist **p**.

Im Exkurs am Ende dieses Handbuchs werden wir "**p_von_z**" und die "kumulative Standard-Normalverteilung" ausführlicher darstellen.

5. Die Stelle **ku** und **ko** der unteren und oberen Konfidenzgrenze in der ASR erhalten wir indem wir **pu** und **po** mit B (im Beispiel: 1000) multiplizieren

	exakt	gerundet
	-----	-----
(9) $ku' = B \cdot pu =$	21.905928	$ku = 22$
(10) $ko' = B \cdot po =$	971.541548	$ko = 972$

Die Stellen sollten ganzzahlige Werte sein. Am einfachsten ist es, ku' und ko' auf- bzw. abzurunden. Der Wert β_u für die untere Konfidenzgrenze könnte dann sehr einfach gleich dem an der 22. Stelle in ASR stehendem Wert $\beta(22)$ gesetzt werden und entsprechend β_o gleich dem Wert $\beta(972)$ an der 972 Stelle von ASR.

Eine andere Möglichkeit besteht darin, zwischen $\beta(21)$ und $\beta(22)$ bzw. zwischen $\beta(971)$ und $\beta(972)$ zu interpolieren. Also folgt SPSS und interpoliert "normalverteilt" (siehe SPSS Statistics 21 Algorithms, S.73).

Der untere und obere Grenzwert werden in Also "normalverteilt" interpoliert. Dabei entstehen schließlich die gesuchten Intervallgrenzen

$$\begin{aligned}(11) \text{ untere Konfidenzgrenze} &= \beta_u = -0.045561 \\(12) \text{ obere Konfidenzgrenze} &= \beta_o = 0.082721\end{aligned}$$

Mit k_u und k_o kann das *effektive Vertrauensniveau* $e_{konfniv}$ ermittelt werden. Es ist

$$\begin{aligned}& \text{(} k_u \text{ und } k_o \text{ nicht gerundet)} \\(13.1) \text{ } e_{konfniv} &= (971.541548 - 21.905928) / 1000 = 94.96\end{aligned}$$

$$\begin{aligned}& \text{(} k_u \text{ und } k_o \text{ gerundet)} \\(13.2) \text{ } e_{konfniv} &= (k_o - k_u + 1) / B = (972 - 22 + 1) / 1000 = 95.10\end{aligned}$$

Das effektive und das ursprünglich vorgegebene Vertrauensniveau von 95% stimmen überein - wenn auch auf Grund von Rechenungenauigkeiten nicht exakt

Einschub2: (Zu Bootstrap-Programmen in Abschnitt 6.3)

Beim Bootstrapping kann es geschehen, dass in der ASR mehrere hintereinander stehende β -Werte identisch mit dem β_{org} -Wert aus der Originalstichprobe sind. Dieser Fall tritt regelmäßig bei Bootstrap-Programmen zu den Basis-Statistiken (Mittelwert, Median, Quartile, Häufigkeiten) auf und wird dort auch in besonderer Weise behandelt. Das wird in Abschnitt 6.3 ausführlich dargestellt. In diesen Bootstrap-Programmen ist fast immer effektives und vorgegebenes Vertrauensniveau verschieden. Das effektive kann an das vorgegebene angepasst werden. Das geschieht dadurch, dass im Also-C-Programm mit $e_{konfniv}$ zurückgesprungen wird hinter Gleichung (4), dort eine Ergänzung vorgenommen wird und dann ein 2. Mal heruntergerechnet wird bis zu (13). Das dann entstehende zweite $e_{konfniv}$ ist dann mit dem ursprünglich vorgegebenen identisch.

6.1.1 Grafische Darstellung des Konfidenzintervalls

Um die untere und obere Intervallgrenze zu berechnen wird, wie wir oben ausgeführt haben, die Funktion "p_von_z" gebraucht. Die beiden Intervallgrenzen werden durch diese Funktion in einen p - und einen z -Wert gewandelt, die sich als 2 Punkte auf der Kurve der

kumulativen Standard-Normalverteilung
für p und z mit negativer Steigung

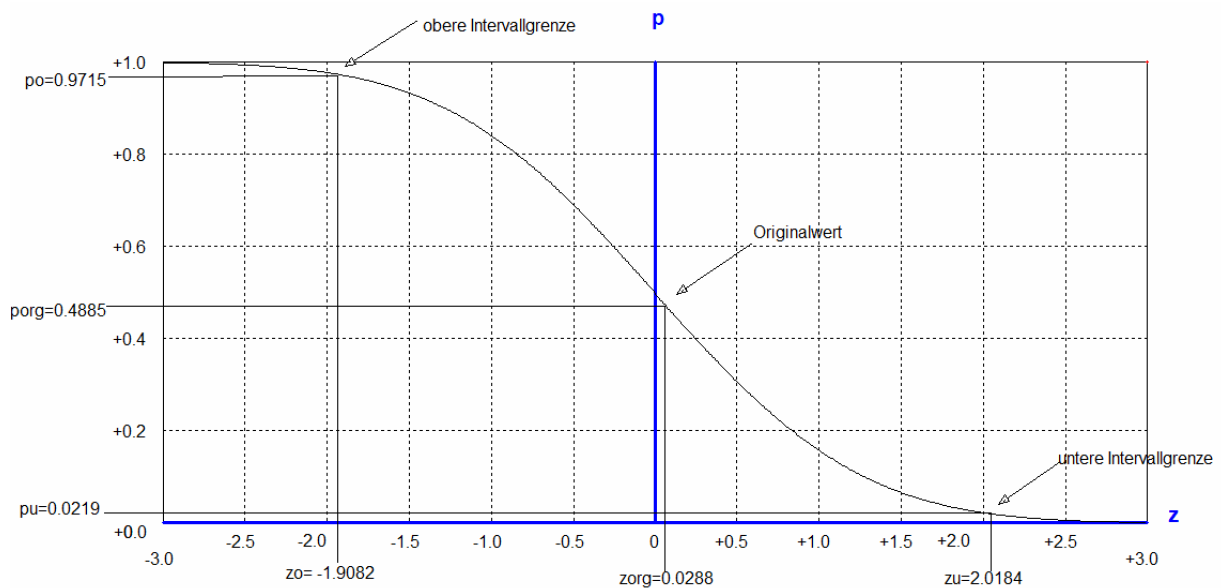
grafisch darstellen lassen. Für den Originalwert haben wir den p -Wert unmittelbar aus der ASR entnommen und den entsprechenden z -Wert durch die Funktion "z_von_p" gewonnen. Diese Funktion wurde durch die Kurve der umgedrehten (inversen) kumulativen Standard-Normalverteilung grafisch abgebildet.

Im Exkurs am Ende des Handbuchs werden (1) die allgemeine kumulative Normalverteilung, (2) die kumulative *Standard*-Normalverteilung mit positiver und negativer Steigung und (3)

die inverse kumulative *Standard*-Normalverteilung mit samt ihrer grafischen Abbild als Kurven ausführlich dargestellt.

Wir können nun die für das Konfidenzintervall charakteristischen drei Punkt, die beiden Intervallgrenzen und den Originalwert in die Kurve der kumulativen Standard-Normalverteilung für p und z mit negativer Steigung einzeichnen.

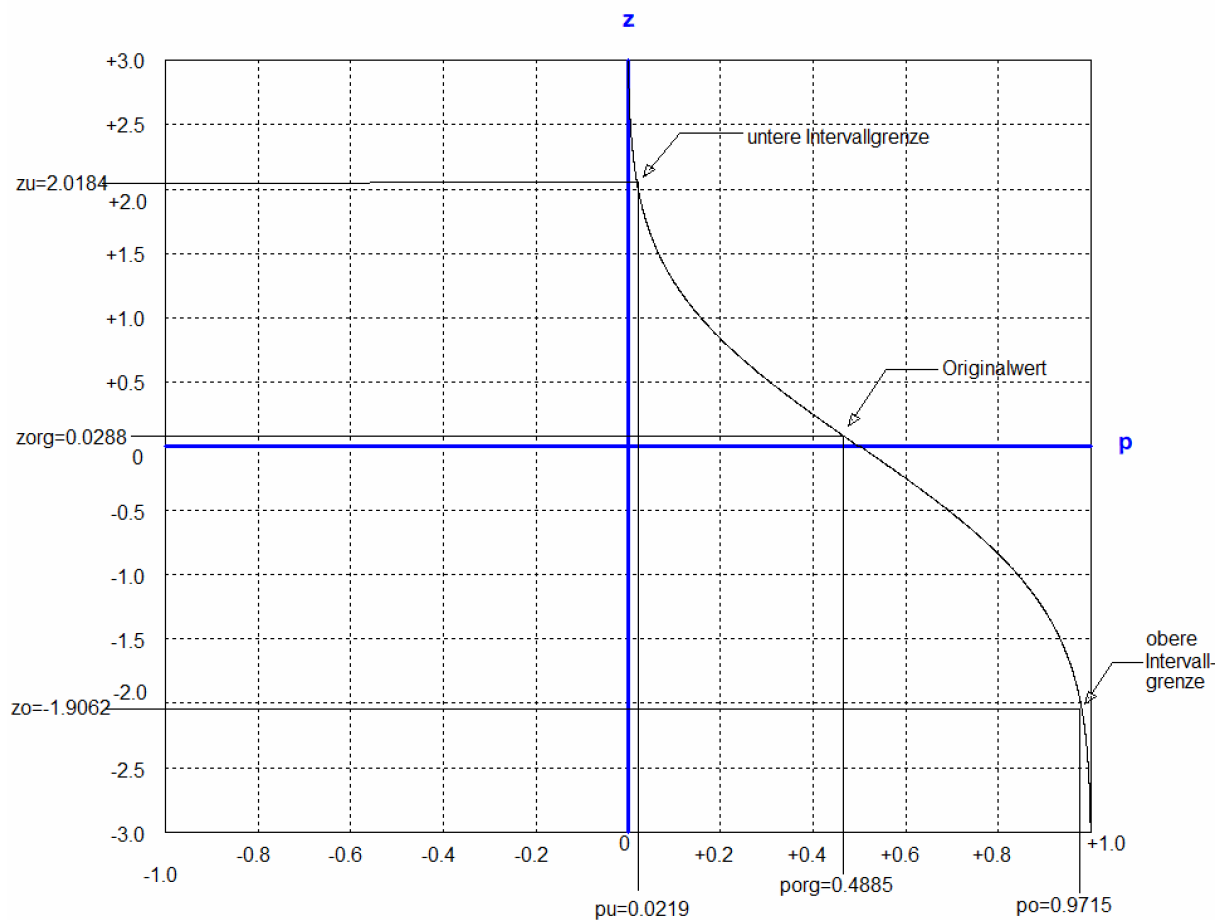
Kurve der kumulativen Standard-Normalverteilung mit negativer Steigung und den 3 Punkten für untere Intervallgrenze, Originalwert und obere Intervallgrenze



Diese Kurve mit den 3 Punkten wird im Grafikprogramm "KumNormVert.grf" durch Klick auf "alle Progs" am Oberrand des Almo-Fensters gefunden.

Wir können aber auch die "inverse kumulative Standard-Normalverteilung" verwenden um diese 3 "charakteristischen" Punkte des BC- bzw. BCa-Verfahrens grafisch zu verdeutlichen. Diese ist die Umkehrfunktion zur oben abgebildeten "kumulative Standard-Normalverteilung mit negativer Steigung"

Kurve der inversen kumulativen Standard-Normalverteilung (mit negativer Steigung) und den 3 Punkten für untere Intervallgrenze, Originalwert und obere Intervallgrenze



Mit "invers" ist "umgekehrt" (die Umkehrfunktion) gemeint. Die Programm-Maske für diese Grafik kann der Benutzer nach Klick auf "alle Progs" am Oberrand des Almo-Fensters unter dem Namen "invKum3Punkt.Gr" direkt laden.

Wir haben in die beiden Grafiken eingesetzt:

1. Den Originalwert aus der Originalstichprobe als Anteilswert **porg** und z-Wert **zorg**.
Dabei kann **porg** interpretiert werden als der Anteil von 1000 Werten in ASR, der kleinergleich dem **ßorg**-Wert aus der Originalstichprobe ist. Siehe oben Gleichung 1 und 2. In unserem Beispiel wird **ßorg** an der Stelle **korg=489** von ASR überschritten.
2. Die untere Intervallgrenze als Anteilswert **pu** und z-Wert **zu**.
Dabei kann **pu** interpretiert werden als der Anteil von 1000 Werten in ASR, der kleinergleich dem unteren Grenzwert des Konfidenzintervalls ist. In Prozent ausgedrückt: **2.1906%**. Bei **1000** Bootstrapstichproben ist das die Stelle (gerundet) **ku=22**
3. Die obere Intervallgrenze als Anteilswert **po** und z-Wert **zo**.
Dabei kann **po** interpretiert werden als der Anteil von 1000 Werten in ASR, der kleinergleich dem oberen Grenzwert des Konfidenzintervalls ist. In Prozent ausgedrückt: **97.1542%**. Bei **1000** Bootstrapstichproben ist das die Stelle (gerundet) **ko=972**

Es befinden sich also

$$\mathbf{konfniv} = 97.1542 - 2.1906 = 94.9636 \quad (\text{gerundet}) \quad 95\%$$

der Werte innerhalb des Konfidenzintervalls - so wie das durch das Vertrauensniveau vorgegeben war.

Die Grundannahme für das BC- bzw. BCa-Verfahren ist folgende:

Alle (z.B. 1000) Bootstrap-Werte in ASR lassen sich entsprechend ihrer Stelle k als p-Werte (nach der einfachen Formel $p=k/B$) ausdrücken. Mit der Funktion "**z_von_p**" wird der zugehörige z-Wert gefunden. So lassen sich alle Bootstrap-Werte als **(p, z)**-Punkte auf der Kurve der kumulativen Standard-Normalverteilung mit negativer Steigung oder deren Umkehrfunktion abtragen.

6.1.2 Der Akzelerationskoeffizient

Es muss also noch geklärt werden, wie der Akzelerationskoeffizient a gebildet wird. Das geschieht dadurch, dass ein "**Jackknife**"-Verfahren gerechnet wird.

Bei diesen Verfahren wird aus ASR, der aufsteigend sortierten Reihe der β -Koeffizienten aus den 1000 Bootstrapstichproben, der erste Wert (-0.077992, siehe Tabelle 6.2) herausgenommen. Es entsteht die Reihe 1. Aus deren 999 Werten wird der Mittelwert M1 errechnet. Dann wird eine 2. Reihe, bei der der 2. Wert (-0.077423) herausgenommen wird und von den verbleibenden 999 Werten der Mittelwert M2 gebildet wird usw. bis zur 1000. Reihe, bei der der 1000. Wert herausgenommen wurde. Auf diese Weise entstehen 1000 Mittelwerte M1 bis M1000. Wir geben zur Illustration die ersten 3 Mittelwerte aus

$$\mathbf{M1} = 0.023441$$

$$\mathbf{M2} = 0.023440$$

$$\mathbf{M3} = 0.023434$$

. .

$$\mathbf{GM} = 0.023339 \quad (\text{Gesamt_Mittel aus M1 bis M1000})$$

Mit folgender komplizierten Gleichung wird dann der "Akzelerations"-Koeffizienten a ermittelt. (Siehe Zhang u.a., 2010, Seite 118, Gleichung 19)

$$(13) \quad a = \frac{\sum_{i=1}^B (GM - M_i)^3}{6 * [\sum_{i=1}^B (GM - M_i)^2]^{\frac{3}{2}}}$$

$$a = -0.00038236$$

M_i = Mittelwert aus Jackknife-Stichprobe i
GM = Gesamtmittelwert aus allen einzelnen Jackknife-Mittelwerten
B = Zahl der Bootstrapstichproben
 \sum = die Summierung erfolgt für i=1 bis i=B=1000, d.h. über alle Bootstrapstichproben
 \wedge = das Symbol \wedge bedeutet "potenzieren"

Im Zähler steht die Summe der zur 3. Potenz erhobenen Abweichungen der einzelnen Mittelwerte M_i vom Gesamtmittelwert GM: $\text{Sum1} = \sum (GM - M_i)^3$

$$\text{Sum1 beträgt} = 9.89082e-007$$

Im Nenner steht die Summe der quadrierten Abweichungen $\text{Sum2} = \sum (GM - M_i)^2$, wobei diese quadrierte Summe noch mit 3/2 potenziert und dann noch mit 6 multipliziert wird

$$\text{Sum2 beträgt} = 2.25669e-012$$

6.2 Der p-Wert beim BC- und BCa-Verfahren

Wir verwenden folgende Notation

ASR = aufsteigend sortierte Reihe der β -Werte aus den Bootstrapstichproben
B = Zahl der Bootstrapstichproben
kNull = Stelle in ASR, an der der β -Koeffizient den Wert 0 besitzt oder überschreitet.
pNull = Anteilswert von kNull = kNull/B

korg = Stelle in ASR, an der der β -Koeffizient den Originalwert Borg (aus der Originalstichprobe) besitzt oder überschreitet
zorg = z-Wert des Originalwerts - genauer: $z = z_{\text{von}_p}(\text{korg}/B)$.
 Siehe Gleichung 1 und 2 in Abschnitt 6.1

konfniv = Vertrauensniveau (Beispiel: 95)
alpha = 1-konfniv/100 (Beispiel: = 0.05)
alpha/2 = (1-konfniv/100)/2 (Beispiel: = 0.025)
1-alpha/2 = 0.5+konfniv/200 (Beispiel: = 0.975)

Wir werden den Rechengang für die Signifikanz p beim BC- und BCa-Verfahren an unserem Beispiel an der Variablen "Freizeit" zeigen.

Wir ermitteln die Stelle in der aufsteigend sortierten Reihe der Bootstrap-Koeffizienten (ASR) an der der β -Koeffizient den Wert 0 besitzt oder überschreitet. Wir bezeichnen ihn mit **kNull**. Für "Freizeit" ist die Stelle **kNull=215**. In Abschnitt 6.1, Gleichung 9 und 10 haben wir die Stellen der Konfidenzgrenzen in ASR mit **ku** und **ko** bezeichnet. Für die Variable "Freizeit" haben wir **ku=22** und **ko=972** errechnet. Wenn nun **knull** (zufällig) gleich **ku**

oder k_0 wäre, dann wäre selbstverständlich die gesuchte zweiseitige Signifikanz $p = (100 - 95) / 100 = 0.05$, wobei 95 das vorgegebene Vertrauensniveau ist. Die folgenden Rechenschritte wären also überflüssig.

Das Glück werden wir selten haben. Wir müssen in folgender Weise vorgehen: Wir versuchen ein zusätzliches, hypothetisches Konfidenzintervall zu konstruieren, dessen untere oder obere Intervallgrenze den β -Wert=0 an der ASR-Stelle $k_{Null}=215$ besitzt oder gerade überschreitet. Wir bezeichnen es als "Null-Konfidenzintervall".

Dazu benötigen wir noch die Stelle k_{org} des originalen β -Werts in ASR. k_{org} ist 489, Das bedeutet: k_{Null} ist in unserem Beispiel kleiner als k_{org} .

Allgemein gilt:

Wenn k_{Null} kleiner k_{org} ist, dann ist die Nullstelle in ASR die untere Intervallgrenze des Null-Konfidenzintervalls. Wenn k_{Null} größer k_{org} ist dann ist die Nullstelle die obere Intervallgrenze. Was wir hier "Nullstelle" genannt haben wird in folgender Gleichung (14) als p_{Null} und z_{Null} eindeutig definiert.

Das Vertrauensniveau des Null-Konfidenzintervalls gilt es nun zu bestimmen. Konkret wird das Vertrauensniveau definiert durch folgende Parameter:

$$\begin{aligned} \text{konfniv} &= \text{Vertrauensniveau (z.B. 95)} \\ \alpha &= 1 - \text{konfniv} / 100 \quad (\text{z.B. } 0.05) \end{aligned}$$

Den alpha-Wert dieses Vertrauensniveaus gilt es zu finden. Er ist gleich dem gesuchten zweiseitigen Signifikanzwert p.

Der Rechengang verläuft in folgender Weise:

Wir ermitteln für die Stelle k_{Null} deren Anteilswert p_{Null} und den z-Wert von p_{Null} . Den speziellen Fall, dass die ASR keinen Wert 0 enthält, behandeln wir im nachfolgenden Abschnitt 6.2.1.

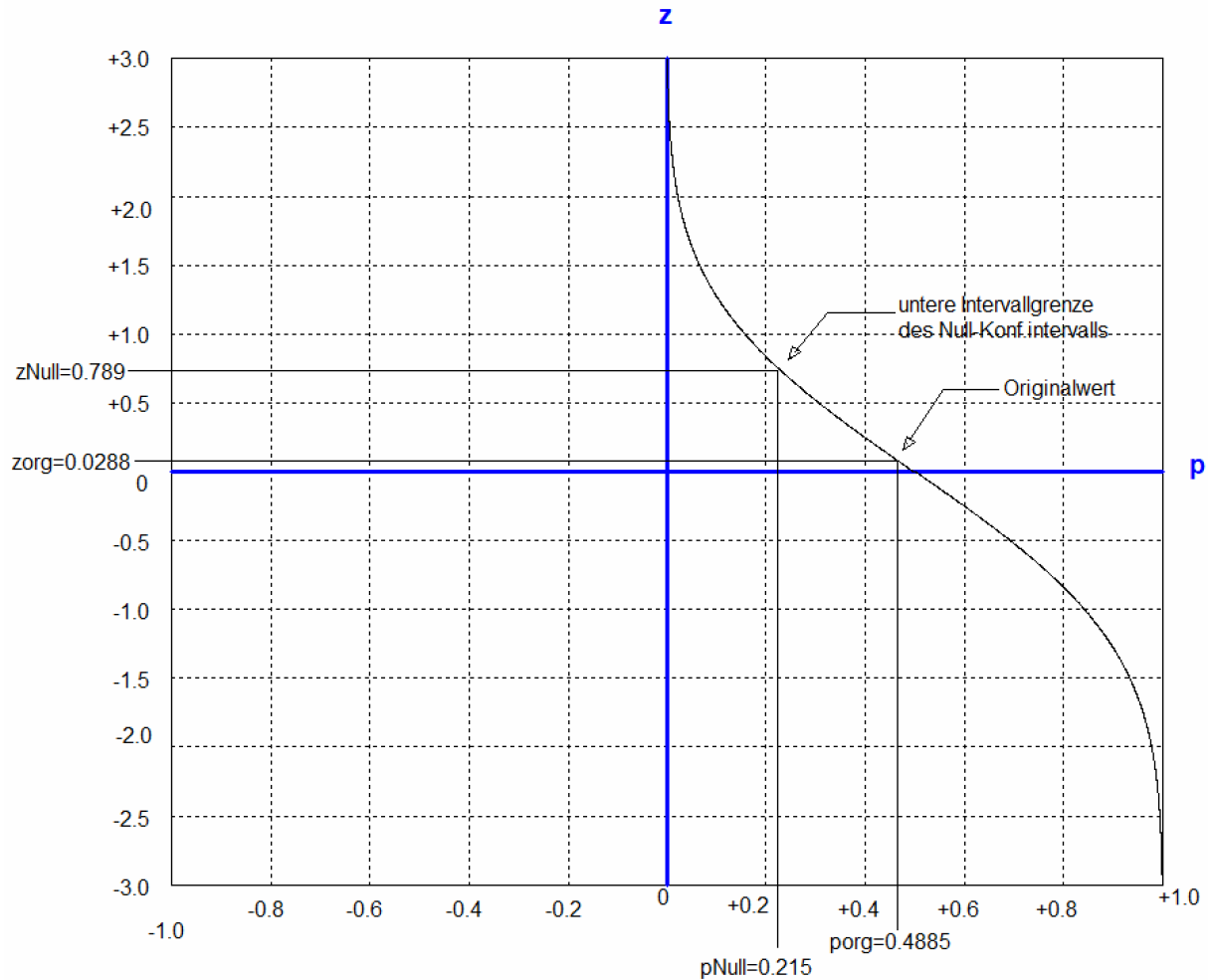
$$\begin{aligned} (14) \quad p_{Null} &= k_{Null} / B = 215 / B = 0.215 \\ z_{Null} &= z_{\text{von}_p}(p_{Null}) = z_{\text{von}_p}(0.215) = 0.789245 \end{aligned}$$

z_{Null} ist der z-Wert der unteren oder oberen Grenze des Null-Konfidenzintervalls - in unserem Beispiel, bei dem k_{Null} kleiner als k_{org} ist, der untere.

Wir kennen nun den Anteilswert und den z-Wert der unteren Intervallgrenze und aus Gleichung 1 und 2 aus Abschnitt 6.1 die entsprechenden Werte des Originalwerts

	<u>Anteilswert</u>	<u>z-Wert</u>
Originalwert an Stelle $k_{org}=489$	0.488511	0.028810
untere Intervallgrenze an Stelle $k_{Null}=215$	0.215	0.789245

Wir können nun, in der Art und Weise, mit der wir das in Abschnitt 6.1.1 getan haben, das Null-Konfidenzintervall als Kurve der *inversen kumulativen Standard-Normalverteilung* zeichnen. Wir haben in die folgende Grafik nur die untere Intervallgrenze eingetragen. Die obere brauchen wir nicht. Unser Anliegen ist es ja, die Signifikanz p zu finden



Im Abschnitt 6.1 wurde in Gleichung 5 und 6 die Bestimmungsgleichung für die untere **zu** und obere **zo** Intervallgrenze in z-Werten für das BC-Verfahren angegeben. Wir wiederholen sie.

Gleichung (5) aus Abschnitt 6.1: $z_u = 2 \cdot z_{org} + z_1$

Gleichung (6) aus Abschnitt 6.1: $z_o = 2 \cdot z_{org} + z_2$

Für das BCa-Verfahren muss noch der Akzelerations-Koeffizient **a** in die beiden Gleichungen eingefügt werden. Die Gleichungen für BCa sind in Abschnitt 6.1 angegeben. Wir wiederholen sie

Gleichung (5a) aus Abschnitt 6.1: $z_u = z_{org} + (z_{org} + z_1) / (1 - a \cdot (z_{org} + z_1))$

Gleichung (6a) aus Abschnitt 6.1: $z_o = z_{org} + (z_{org} + z_2) / (1 - a \cdot (z_{org} + z_2))$

Wir wiederholen auch die Zeichenerklärung aus Abschnitt 6.0

zorg = z-Wert des Originalwerts - genauer: $z = z_{von_p}(k_{org}/B)$.

Siehe Gleichung 1 und 2 in Abschnitt 6.1

zu= z-Wert von unterer Intervallgrenze

zo= z-Wert von oberer Intervallgrenze

z1= z-Wert von $\alpha/2$

z2= z-Wert von 1-alpha/2

Auf unser Null-Konfidenzintervall angewandt entspricht **zNull** (aus obiger Gleichung 14) den Parametern **zu** oder **zo** aus den Gleichungen 5 und 6 bzw. für BCa 5a und 6a aus Abschnitt 6.1.

Wir lösen die obigen Gleichungen 5 und 6 Abschnitt 6.1 nach **z1** und **z2** auf - dabei ist jetzt

z1= z-Wert von alpha/2 für das Null-Konfidenzintervall
z2= z-Wert von 1-alpha/2 für das Null-Konfidenzintervall

- (15) Wenn **kNull** kleiner **korg** dann ist (für das BC-Verfahren)
z1 = zNull-2*zorg
z2 = KeinWert
- (16) Wenn **kNull** größer **korg** dann ist
z2 = zNull-2*zorg
z1 = KeinWert

Man beachte: Die rechte Seite der Gleichungen 15 und 16 (und auch der nachfolgenden 15a und 16a) ist gleich. Die abhängige Variable auf der linken Seite ist jedoch einmal **z1** und einmal **z2**.

Für das **BCa**-Verfahren gilt nach entsprechender Umformung von (5a) und (6a) aus Abschnitt 6.1:

- (15a) Wenn **kNull** kleiner **korg** dann ist
z1 = (zNull-2*zorg-a*zNull*zorg)/(1+a*zNull) = 0.734320
z2 = KeinWert
- (16a) Wenn **kNull** größer **korg** dann ist
z2 = (zNull-2*zorg-a*zNull*zorg)/(1+a*zNull)
z1 = KeinWert

Ist **a=0** dann reduzieren sich die Gleichungen auf die entsprechenden für das BC-Verfahren.

Den gesuchten **alpha**-Wert, der gleich der gesuchten zweiseitigen Signifikanz **p** ist erhalten wir dann durch

- (17) Wenn **kNull** kleiner **korg** dann ist
alpha = 2* p_von_z(z1) = 0.462783
- (18) Wenn **kNull** größer **korg** dann ist
alpha = -2*(p_von_z(z2)+1)
- (19) **zweiseitige Signifikanz p = alpha**

In (17) und (18) wurde mit 2 multipliziert, da **z1** der z-Wert von **alpha/2** ist, genau so **z2**.

Für unser Beispiel finden wir also eine zweiseitige Signifikanz von **p=0.462783**.
Im Vergleich dazu haben wir für das einfache Perzentil-Verfahren ein **p=0.430000** erhalten.

Ein Beiprodukt, für das wir eigentlich keine Verwendung habe, ist das Vertrauensniveau des Null-Konfidenzintervalls. Wir bezeichnen es als **Null_konfniv**. Es ist

$$(20) \text{ Null_konfniv} = (1-\alpha)*100 = 53.7217$$

6.2.1 Keine Null vorhanden

Wie soll aber verfahren werden, wenn die z.B. 1000 aufsteigend sortierten Werte des Bootstrap-Koeffizient keinen Wert 0 aufweisen? In dieser Situation muss der ungünstigste Fall unterstellt werden, dass gerade unterhalb des ersten Werts in ASR bzw. oberhalb des letzten Werts in ASR der Wert 0 folgen würde - hätte man eine weitere Stichprobe gerechnet. Also berechnet somit ein "Null-Konfidenzintervall" (wie wir es oben nannten), dessen unterer Intervall-Grenzwert der erste Wert in ASR ist und dessen oberer Grenzwert der letzte Wert in ASR ist.

Wenn wir in ASR keinen β -Koeffizienten finden, der gleich 0 ist oder den Wert 0 überschreitet, dann müssen wir 2 Fälle unterscheiden:

Fall 1: Schon der 1. β -Koeffizient in ASR ist größer 0. Dann wird obige Gleichung (14) so verändert

$$(14.1) \begin{aligned} p_{\text{Null}} &= 1/B && = 0.001 \\ z_{\text{Null}} &= z_{\text{von_p}}(p_{\text{Null}}) && = 3.092 \end{aligned}$$

Fall 2: Alle β -Koeffizient in ASR sind negativ. Der Wert 0 wird bis zum letzten Wert in ASR nicht überschritten. Dann wird Gleichung (14) zu

$$(14.1) \begin{aligned} p_{\text{Null}} &= 1-1/B && = 0.999 \\ z_{\text{Null}} &= z_{\text{von_p}}(p_{\text{Null}}) && = -3.092 \end{aligned}$$

Danach wird wie oben beschrieben mit diesem z_{Null} -Wert weiter gerechnet. Der p-Wert ist nur von der Zahl der Stichproben abhängig.

6.3 BC- bzw. BCa-Verfahren mit *mehrfachen* Originalwerten

6.3.1 Erweiterte Notation

Wir verwenden in den folgenden Ausführungen diese Notation. Siehe auch Abschnitt 1.1 und 6.0

ASR = aufsteigend sortierte Reihe der Bootstrapkoeffizienten β aus den x Bootstrapstichproben

β = Wert des Bootstrap-Koeffizienten aus einer der Bootstrapstichproben
 k = Stelle von β in ASR

β_{org} = β -Wert aus der Originalstichprobe
 β_{org1} = erster (unterer) Originalwert in ASR
 β_{org2} = letzter (oberer) Originalwert in ASR

k_{org1} = Stelle des ersten Originalwert β_{org1} in ASR
 k_{org2} = Stelle des letzten Originalwert β_{org2} in ASR

k_M = Mitte zwischen k_{org1} und k_{org2}
 β_M = β -Wert an der Stelle k_M

β_u = β -Koeffizient der unteren Konfidenzgrenze
 β_o = β -Koeffizient der oberen Konfidenzgrenze

k_u = Stelle der unteren Konfidenzgrenze in ASR

ko = Stelle der oberen Konfidenzgrenze in ASR

konfniv = vorgegebenes Vertrauensniveau

ekonfniv = effektives Vertrauensniveau

βu1 = β-Koeffizient der unteren Konfidenzgrenze
berechnet bezüglich ersten Originalwert **βorg1** in ASR

βo1 = β-Koeffizient der oberen Konfidenzgrenze
berechnet bezüglich erstem Originalwert **βorg1** in ASR

βu2 = β-Koeffizient der unteren Konfidenzgrenze
berechnet bezüglich letztem Originalwert **βorg2** in ASR

βo2 = β-Koeffizient der oberen Konfidenzgrenze
berechnet bezüglich letztem Originalwert **βorg2** in ASR

Den beiden Symbolen **βu** und **βo** für den unteren und oberen Intervallgrenzwert haben wir den Zahlenindex 1 bzw. 2 angehängt. Damit soll ausgedrückt werden, dass der jeweilige Intervallgrenzwert determiniert wurde durch **βorg1** bzw. **βorg2**

In Almo sind diese 3 Programm-Masken für den Bootstrap der **Basis-Statistiken** vorhanden

1. **Prog05m7** Bootstrap der Anteilswerte von Häufigkeiten

2. **Prog05m8** Bootstrap des arithmetischen Mittelwerts

3. **Prog05m9** Bootstrap von Median und Quartilen

In diesen Programmen kann das Konfidenzintervall ebenfalls nach dem BC- bzw. BCa-Verfahren berechnet werden. Dabei tritt jedoch ein besonderes Problem auf, für das in den 3 Programm-Masken auch versucht wird, eine Lösung anzubieten.

Beim Bootstrap z.B. der Häufigkeiten bzw. deren Anteilswerten (in Prog05m7) sind in der Reihe der aufsteigend sortierten Bootstrapkoeffizienten fast immer mehrere β-Koeffizienten enthalten, deren Wert gleich dem **βorg** aus der Originalstichprobe ist. Der **βorg**-Wert ist *mehrfach* in ASR enthalten. Siehe nächste Tabelle 7.1. So besitzen im nachfolgenden Beispiel 180 Bootstrapstichproben einen β-Koeffizienten, dessen Wert gleich **βorg** aus der Originalstichprobe ist. Die mehrfach in ASR enthaltenen Originalwerte werden gelegentlich auch *Bindungen* (engl. "ties") genannt. Siehe "MedCalc" in Literaturliste.

Prinzipiell ist das Problem der *mehrfachen Originalwerte in ASR* nicht auf die Basis-Statistiken beschränkt. Es kann in jedem der in Almo vorhandenen Bootstrap-Programmen auftreten. Fast immer jedoch ist die Zahl der **βorg**-Wiederholungen so klein, manchmal nur eine oder zwei Wiederholungen, dass ein Problem gar nicht sichtbar wird. Sichtbar wird das Problem nur bei den drei oben genannten Programmen zu den Basis-Statistiken. Bei ihnen teilt Almo mit, von welcher Stelle **korg1** bis **korg2** die Originalwerte in der ASR enthalten sind. Auch das *effektive Vertrauensniveau* ist bei diesen Programmen kleiner oder größer als das vorgegebene, es sei denn, dass der Benutzer eine Anpassung in einer besonderen Option angefordert hat (siehe 6.3.7).

Das Problem der „mehrfachen Originalwerte“ besteht darin, dass unklar wird, wie das Konfidenzintervall berechnet werden soll. Es muss für das BC- bzw. BCa-Verfahren ein Modus gefunden werden, der den Sinn dieses Verfahrens, Verzerrung ("bias") und Schiefe der Verteilung zu korrigieren, erhält und möglichst das vorgegebene Vertrauensniveau beibehält. Almo schlägt dafür Modus 3 oder 4 vor.

6.3.2 Ein Beispiel

Im Bootstrap-Programm **Prog05m7.Msk** für Anteilswerte wird u.a. aus der Datei "Testdat.fre" die Variable "Leistung" ausgezählt. Die Datei umfasst 61 Datensätze. Die Variable "Leistung" nimmt die ganzzahligen Werte 1 bis 9 an. Für jede einzelne Ausprägung wird der Standardfehler und das Konfidenzintervall berechnet. Wir zeigen das Ergebnis gekürzt:

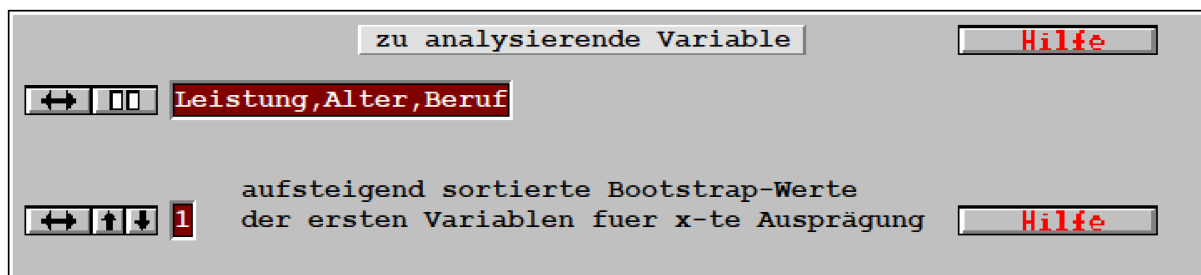
	Original-Stichprobe		Bootstrap der Anteilswerte				
	*a Haeufigkeit	*b Anteil	*c Anteil	*d Verzerr.	Stand. fehler *e	Konfidenzintervall *d BCa-Verfahren Konfniv=95.000 unten oben	
V5 Leistung							
1	5	0.0820	0.0829	-0.0010	0.0353	0.0328	0.1311
2	10	0.1639	0.1661	-0.0022	0.0488	0.0984	0.2459
3	13	0.2131	0.2132	-0.0000	0.0526	0.1311	0.3115
.

Gerechnet wurde mit BCa-Verfahren, Modus 4, keine Anpassung.

Wir wollen die Ausprägung "1" von Leistung betrachten.

In der Originalstichprobe haben 5 Probanden von 61 diese Ausprägung gewählt. Das ist ein Anteil von 8%, genau von einem Anteil von $\beta = 0.081967$. Wir dürfen vermuten, dass auch in mehreren der 1000 Bootstrapstichproben nur 5 Probanden (also ein Anteil von $\beta = 0.081967$) die "1" gewählt haben. Tatsächlich waren es 181 Bootstrapstichproben. In der Programm-Maske haben wir für die Ausprägung "1" angefordert, die ASR, d.h. die "aufsteigend sortierten Anteilswerte aus allen 1000 Booststrastichproben" auszugeben.

In der Eingabebox muss im 2. Eingabefeld eine "1" eingesetzt werden. Dann werden die Anteilswerte der 1. Ausprägung ausgegeben,



Wir erhalten folgende (gekürzte) Ausgabe.

Tabelle 7.1

Bootstrap-Stichprobe k	ASR Anteilswert β
1	0
.	.
5	0
6	0.0163934
.	.
30	0.0327869
.	.
418	0.0655738
419	0.0655738

420	0.0819672 <--- erster Originalwert β_{org1} an Stelle $k_{org1}=420$

```

421      0.0819672
.
500      0.0819672
.
599      0.0819672
600      0.0819672 <---letzter Originalwert Borg2 an Stelle korg2=600
-----
601      0.0983607
602      0.0983607
.
863      0.131148
.
1000     0.229508

```

Die Besonderheit ist, dass in mehreren Bootstrapstichproben ein Anteilswert β entstanden ist, der exakt gleich dem **Borg**-Wert aus der Originalstichprobe ist. In unserem Beispiel befindet sich von Stelle $k=420$ bis Stelle $k=600$ ein β -Wert, der exakt identisch ist mit dem **Borg**-Wert aus der Originalstichprobe. Diese Situation tritt vor allem auf, wenn die Variable, die dem Bootstrap-Verfahren unterworfen wird, einen kleinen Wertebereich mit nur wenige ganzzahligen Werten besitzt - wie in unserem Beispiel die 9 Werte 1 2 3 4 ... 9 und aus diesen ein einfacher Koeffizient, wie der Anteilswert oder der Median als Bootstrapkoeffizient errechnet wird.

Diverse Werte

In unserem Beispiel habe in der Originalstichprobe von 61 Befragten 5 die Ausprägung 1 gewählt und 56 eine der anderen 8 Ausprägungen. Also ermittelt nun auch die Zahl der diversen Werte, die in ASR auftreten. In Abschnitt 6.3.7, Tabelle 7.3 wird ausgegeben, dass in den 1000 Bootstrapstichproben-Werte nur 14 Werte verschieden waren. Die ASR ist folgende:

diverse Werte	Anteilswert p	mehrfach x Mal	
-----	-----	-----	
1	0	5	
2	0.016393	24	
3	0.032787	86	<---- untere Intervallgrenze
4	0.049180	139	
5	0.065574	165	
6	0.081967	181	<---- Originalwert
7	0.098361	153	
8	0.114754	109	
9	0.131148	82	<---- Obere Intervallgrenze
10	0.147541	32	
11	0.163934	15	
12	0.180328	5	
13	0.196721	3	
14	0.229508	1	

		1000	

Wir haben mit allen 1000 Bootstrapwerten gerechnet und BCa-Modus 4 angewendet. Das effektive Vertrauensniveau **ekonfniv** ist 86.72%. Es wurde nicht an die vorgegebenen 95% angepasst. Wird die Zahl der Bootstrapstichproben auf 2000 erhöht, dann entstehen wieder dieselben diversen Werte. Nur ein neuer mit $p=0.213115$ kommt dazu. Es ist offenkundig, dass ca. 14 diverse Werte das Minimum für eine Bootstrap-Analyse sein sollten.

6.3.3 Die Bootstrap-Optionsbox für *mehrfache Originalwerte*

Die Bootstrap-Optionsbox für die 3 Programm-Masken der Basis-Statistiken Prog05m7, Prog05m8 und Prog05m9 ist etwas anders gestaltet als die für alle anderen Bootstrap-Programme

Loesche wieder diese Box (dann Voreinstellungen wieder gueltig)

Option: Bootstrap

1 1= Bootstrap ausführen
 0= nicht (voreingestellt)

wieviele Stichproben sollen gerechnet werden

0 **Ergebnisse für die ersten x Stichproben ausgeben**
 (für die Originaldaten werden sie immer ausgegeben)

Konfidenzintervall

 Konfidenzniveau in % Hilfe

3 **Konfidenzintervall berechnen mit ...** Hilfe
 0 = einfachem Perzentil-Verfahren
 1 = nicht möglich
 2 = nicht möglich
 3 = Perzentil-Verf. BC (bias corrected)
 4 = Perzentil-Verf. BCa (bias corrected accelerated)

Modus

 wenn BC- oder BCa-Verfahren dann Konf.intervall
 bezogen auf ... Hilfe

3 1=oberen Originalwert
 2=unteren Originalwert
 3=oberen u. unteren Originalwert
 4=oberen u. unteren Orig.wert (invers)
 5=oberen u. unteren Originalwert (Mitte)

1 **Konfidenzniveau anpassen** Hilfe
 0=nicht
 1=anpassen

Startzahl Zufallsgenerator

Dadurch dass **borg** mehrfach in ASR enthalten ist, stellt sich folgendes Problem: Bezüglich welchen **borg**-Werts an welcher Stelle **korg** soll das Konfidenzintervall berechnet werden? Die einfachste Art und Weise, das Problem zu lösen besteht allerdings darin, das einfache Perzentil-Verfahren zu verwenden. Der "bias" und die Schiefe der Verteilung werden dabei jedoch nicht korrigiert.

Soll das Konfidenzintervall jedoch nach dem BC- bzw. BCa-Kalkül berechnet werden, dann bietet Almo hier 5 Modi an, die in obiger Eingabebox vom Benutzer frei gewählt werden können. Sie werden in nachfolgendem Abschnitt 6.3.4 im Detail erläutert. . Bei Modus 3 z.B. wird die untere Konfidenzgrenze vom **ersten** an der Stelle **korg1** im ASR aufgetretenen **borg**-Wert und die obere Konfidenzgrenze vom **letzten** an der Stelle **korg2** im ASR aufgetretenen **borg**-Wert bestimmt.

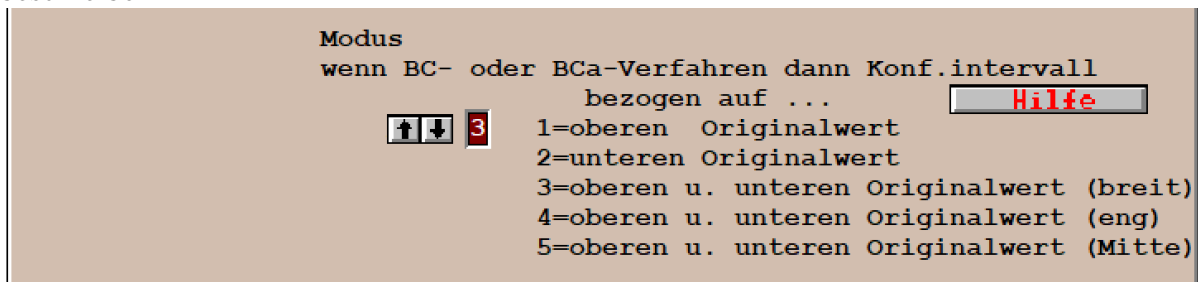
Außerdem wird dem Benutzer angeboten, das "effektive" Vertrauensniveau, an das von ihm vorgegebene anzupassen. In Abschnitt 6.3.8 wird ausführlich darauf eingegangen.

6.3.4 Die Modi für das BC- bzw. BCa-Verfahren bei mehrfachen Originalwerten

Das Problem, wenn mehrfache Originalwerte β_{org} in ASR vorhanden sind, ist:

Für welche Stelle von β_{org} in ASR soll nach dem BC- bzw. BCa-Verfahren die untere und obere Konfidenzgrenze ermittelt werden ?

Almo bietet folgende 5 Modi an, das Problem zu behandeln. Dabei können die "zusätzlichen Informationen" (siehe 6.3.7), die Almo ausgibt, helfen, sich zu entscheiden. Wir zeigen zuerst den betreffenden Ausschnitt aus der Bootstrap-Optionsbox und werden danach die 5 Modi beschreiben



Die beiden Konfidenzgrenzwerte β_u und β_o werden berechnet ...

Modus 1: ...hinsichtlich der Stelle k_{org2} des letzten (oberen) Originalwerts in ASR. Dem entspricht die Standard-Vorgehensweise beim BC- bzw. BCa-Verfahren. So entstehen die Konfidenzgrenzwerte β_{u2} und β_{o2}

Modus 2: ...hinsichtlich der Stelle k_{org1} des ersten (unteren) Originalwerts
So entstehen β_{u1} und β_{o1} .

Modus 3: ...hinsichtlich der Stelle k_{org1} des ersten (unteren) Originalwerts.
Dadurch wird der untere Konfidenzgrenzwert β_{u1} gewonnen

und

...hinsichtlich der Stelle k_{org2} des letzten (oberen) Originalwerts.
Dadurch wird der obere Konfidenzgrenzwert β_{o2} gewonnen.

Das Konfidenzintervall wird bei dieser Vorgehensweise relativ breit, kann jedoch durch das "angepasste effektive Vertrauensniveau" auf das vorgegebene Vertrauensniveau (von üblicherweise 95%) eingengt werden. Die mehrfach aufgetretenen β_{org} -Werte sind in das Konfidenzintervall eingeschlossen

Modus 4: ...hinsichtlich der Stelle k_{org1} des (ersten) unteren Originalwerts.
Dadurch wird der obere Konfidenzgrenzwert β_{o1} gewonnen

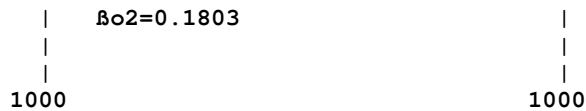
und

...hinsichtlich der Stelle k_{org2} des letzten (oberen) Originalwerts.
Dadurch wird der untere Konfidenzgrenzwert β_{u2} gewonnen.

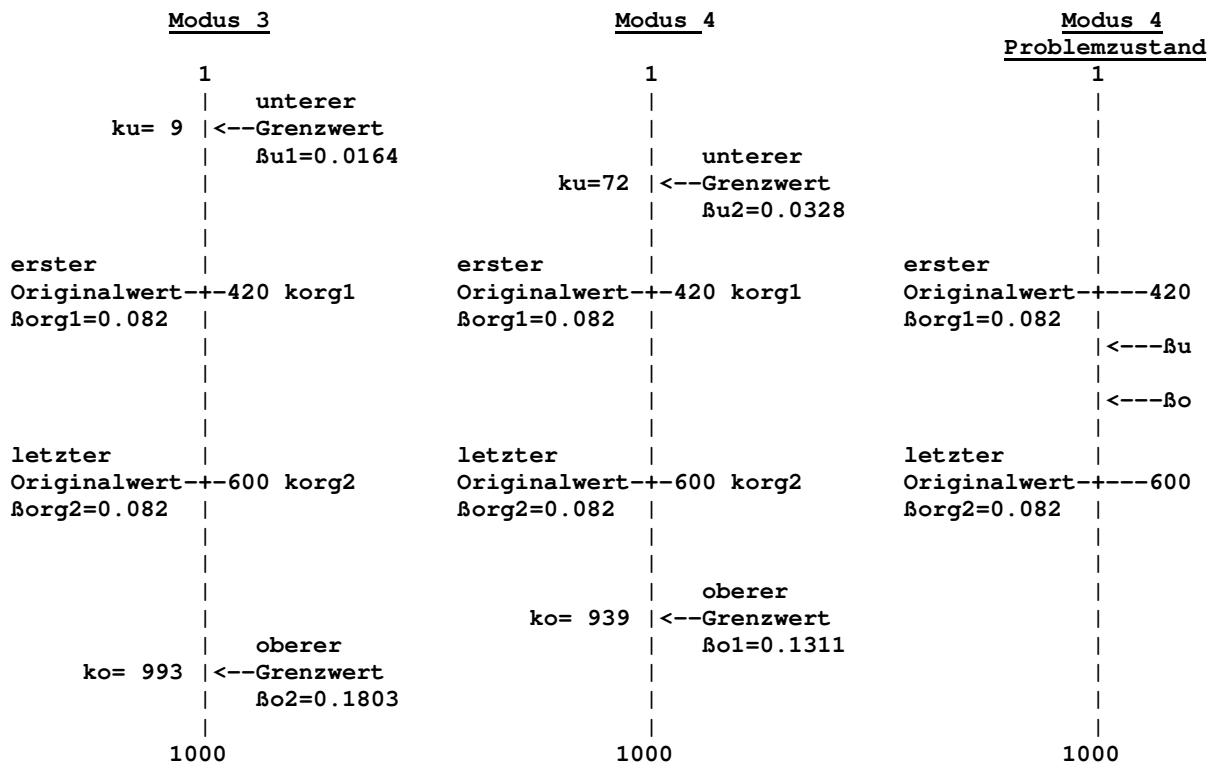
Bei Modus 4 werden im Vergleich zu Modus 3 die Konfidenzgrenzen "invers" durch β_{org1} und β_{org2} bestimmt.

Das Konfidenzintervall wird bei dieser Vorgehensweise relativ eng, kann jedoch durch das "angepasste effektive Vertrauensniveau" auf das vorgegebene Vertrauensniveau (von üblicherweise 95%) geweitet werden.

Modus 5: hinsichtlich der Mitte k_M zwischen den Stellen k_{org1} und k_{org2} des unteren und oberen Originalwerts.
So entstehen die Konfidenzgrenzwerte β_u und β_o



Modus 1 und **2** brücksichtigt nur einen der beiden Originalwerte, β_{org1} oder β_{org2} . Das spezifische Problem der "mehrfachen Originalwerte im ASR" wird ausgeblendet. Werden die beiden nacheinander gerechnet, dann können aus den Ergebnissen Modus 3 und 4 gebildet werden



Modus 3 ist wahrscheinlich die beste Wahl. Es erzeugt ein Konfidenzintervall in drei Teilbereichen: Dem unteren Intervallbereich von k_{org1} bis ku und den oberen Intervallbereich von k_{org2} bis ko . Und es bezieht die *mehrfachen Originalwerte* zwischen k_{org1} und k_{org2} in das Konfidenzintervall als dritten Teilbereich mit ein. Das Konfidenzintervall ist um diesen dritten Teilbereich breiter. Damit wird auch das Vertrauensniveau größer. Es vergrößert sich in unserem Beispiel von den *vorgegebenen* 95% auf ein *effektives Vertrauensniveau* von 98,41%.

Dies ist aber auch die Schwäche von Modus 3. Ist der Bereich der *mehrfachen Originalwerte* sehr groß, dann kann es den ganzen ASR-Bereich von 1 bis 1000 überstreichen. Sogar auch dann, wenn das Konfidenzintervall an das vorgegebene Vertrauensniveau *angepasst* ist. Wir werden im Folgenden noch darauf eingehen. Das heißt: Jeder der 1000 Bootstrap-Stichprobenwerte könnte in einem solchen Fall der wahre Wert aus der Population mit dem angegebenen Vertrauensniveau sein. Das bedeutet dann null Information.

Modus 4

Beim Modus 4 wird die *untere* Intervallgrenze ku durch den *oberen* (letzten) Originalwert k_{org2} bestimmt. Wir haben dies mit ku symbolisiert. Der obere Grenzwert ko wird durch den unteren Originalwert k_{org1} bestimmt. Der Teilbereich der *mehrfachen Originalwerte* zwischen k_{org1} und k_{org2} ist kein dritter, zusätzlicher Bereich. Er wird überdeckt durch das

k_M ist in unserem Beispiel $=510$. Wäre es genau gleich der Mitte von ASR, also $=B/2=500$, dann würden wir dasselbe Ergebnis für die Intervallgrenzstellen und -werte wie beim *einfachen Perzentilverfahren* bekommen.

6.3.5 Änderungen im Kalkül

In Abschnitt 6.1 haben wir in mehreren Gleichungen gezeigt, wie für den Originalwert β_{org} an der Stelle k_{org} in ASR die untere und obere Konfidenzgrenze gewonnen wird.

Es sei daran erinnert, dass k_{org} zuerst in einen Anteilswert und dann in einen z-Wert überführt wird. Siehe Abschnitt 6.1 Kalkül, Gleichung 1 und 2. Dem wird der z-Wert von $\alpha/2$ hinzugefügt. Bei $konfniv=95$ ist dies $z_{von_p}(0.025)$. Dadurch entsteht dann nach weiteren Umformungen die *untere Konfidenzgrenze*. Wird der z-Wert von $1-\alpha/2$ hinzugefügt - bei $konfniv=95$ ist dies $z_{von_p}(0.975)$ - dann entsteht die *obere Konfidenzgrenze*. Das ist das Standardmodell des BC- bzw. BCa-Verfahrens

Wir haben als beste Lösung des Problems der *mehrfachen β_{org} -Werte in ASR* Modus 3 vorgeschlagen und als zweitbesten den Modus 4. Bei Modus3 wird die untere Konfidenzgrenze vom **ersten** im ASR aufgetretenen β_{org} -Wert und die obere Konfidenzgrenze vom **letzten** im ASR aufgetretenen β_{org} -Wert bestimmt. Modus 4 ist invers zu Modus 3. Die untere Konfidenzgrenze wird vom oberen (letzten) β_{org} und die obere Konfidenzgrenze vom unteren (ersten) β_{org} bestimmt. Für die anderen Modi 1, 2 und 5 gelten unverändert die Gleichungen aus Abschnitt 6.1.

An Modus 3 und 4 angepasster Kalkül

Wir müssen den Kalkül, wie er in Abschnitt 6.1 für den einzigen Originalwert an der Stelle k_{org} vorgetragen wurde, anpassen an die Situation, dass jetzt 2 Originalwerte vorhanden sind, ein erster (unterer) an der Stellen k_{org1} und ein letzter (oberer) an der Stelle k_{org2} . Dazu müssen nur die Gleichungen 1, 2, 5, 6, 5a, 6a angepasst werden. Die anderen Gleichungen bleiben unverändert.

Notation:

k_{org1} = Stelle des ersten Originalwert β_{org1} in ASR
 k_{org2} = Stelle des letzten Originalwert β_{org2} in ASR

p_{org1} = Anteilswert des ersten Originalwerts an den B Werten in ASR (im Beispiel: 1000)
 p_{org2} = Anteilswert des letzten Originalwerts an den B Werten in ASR

z_{org1} = z-Wert von p_{org1}
 z_{org2} = z-Wert von p_{org2}

z_1 = z-Wert von $\alpha/2$
 z_2 = z-Wert von $1-\alpha/2$

z_u = z-Wert von unterer Intervallgrenze
 z_o = z-Wert von oberer Intervallgrenze

Die angepassten Gleichungen sind

$$(1.1) \quad p_{org1} = k_{org1}/(B+1)$$

$$(1.2) \quad p_{org2} = k_{org2}/(B+1)$$

$$\begin{aligned}
(2.1) \quad & \text{zorg1} = \text{z_von_p}(\text{porg1}) \\
(2.1) \quad & \text{zorg2} = \text{z_von_p}(\text{porg2}) \\
(5') \quad & \text{zu} = 2 * \text{zorg1} + \text{z1} \\
(6') \quad & \text{zo} = 2 * \text{zorg2} + \text{z2} \\
(5a') \quad & \text{zu} = \text{zorg1} + (\text{zorg1} + \text{z1}) / (1 - a * (\text{zorg1} + \text{z1})) \\
(6a') \quad & \text{zo} = \text{zorg2} + (\text{zorg2} + \text{z2}) / (1 - a * (\text{zorg2} + \text{z2}))
\end{aligned}$$

Wie Almo rechnet

"Mehrfache Originalwerte" in der ASR ereignen sich nicht nur bei den Programmen zu den Basis-Statistiken. Sie können auch z.B. bei einer Regressionsanalyse auftreten - ohne dass das vom Benutzer bemerkt wird. Wird dann standardmäßig im BC- oder BCa-Modus 1 gerechnet, dann wird das Konfidenzintervall falsch berechnet. Almo rechnet deswegen immer mit Modus 3. Ist dann im ASR der Originalwert nur 1 Mal vorhanden, dann ist automatisch **korg1=korg2** mit der Folge, dass in den oben angegebenen angepassten Gleichungen, die zwei **porg1** und **porg2** zusammenfallen und genauso die beiden **zorg1** und **zorg1**. Dies gilt auch für die p-Wert-Berechnung, die im nächsten Abschnitt behandelt wird.

6.3.6 Der Signifikanz p-Wert bei mehrfachem Originalwert

Im Fall der "mehrfachen Originalwerte" wird unterschieden zwischen **korg1**, der ersten (unteren) Stelle von β_{org} in ASR und **korg2**, der letzten (oberen) Stelle von β_{org} in ASR. In Abschnitt 6.2 zum BC- und BCa- Signifikanzwert p müssen in den Gleichungen 15, 15a, 16, 16a, 17 und 18 lediglich die beiden Wenn...-Formulierungen geändert werden in:

Wenn kNull kleiner korg1 ...
Wenn kNull größer korg2 ...

und **zorg** muss entsprechen in **zorg1** und **zorg2** getrennt werden. Dies gilt für Modus 3 und Modus 4. In den anderen Modi 1, 2 und 5 wird nicht zwischen **korg1** und **korg2** unterschieden. Der Rechengang bleibt so, wie in Abschnitt 6.2 beschrieben.

6.3.7 Ergebnis-Ausgabe und "zusätzliche Informationen"

Wir rechnen unser Beispiel aus Abschnitt 6.3.2 mit BCa-Modus 3. Als Bootstrap-Ergebnis erhalten wir eine sehr breite Tabelle, die wir hier aufspalten müssen. Die "zusätzlichen Informationen" müssen als selbständige Tabelle nach unten gestellt werden.

Tabelle 7.2: Häufigkeiten/Anteilswerte der Variablen "Leistung" BCa-Verfahren gerechnet mit Modus 3

Originalstichprobe		Bootstrap der Anteilswerte					
*a Haeufig	*b Anteil	*c Anteil	*d Verzerr.	Stand. *e fehler	Konfidenzintervall*f		
					unten	oben	
Leistung							
1	5	0.0820	0.0830	-0.0010	0.0352	0.0164	0.1803
2	10	0.1639	0.1663	-0.0023	0.0485	0.0656	0.2951
3	13	0.2131	0.2134	-0.0002	0.0521	0.1094	0.3443
4	15	0.2459	0.2453	0.0006	0.0544	0.1311	0.3770
5	8	0.1311	0.1314	-0.0002	0.0427	0.0328	0.2459
6	2	0.0328	0.0327	0.0001	0.0217	0.0000	0.0986
7	4	0.0656	0.0641	0.0015	0.0308	0.0000	0.1639

8	3	0.0492	0.0509	-0.0017	0.0285	0.0000	0.1311
9	1	0.0164	0.0140	0.0023	0.0150	0.0000	0.0984

Tabelle 7.3

zusätzliche Informationen								
	div	korg1	korg2	ku	ko	ekonfniv	a	Modus
	*1	*2	*3	*4	*5	*6	*7	*8
Leistung								
1	14	420	600	9	993	98.41	0.00182180	3:0
2	21	443	563	12	988	97.62	0.00177237	3:0
3	22	452	562	14	988	97.46	0.00102550	3:0
4	20	466	567	16	989	97.27	0.00079529	3:0
5	18	433	583	11	991	98.05	0.00098109	3:0
6	8	401	682	7	998	99.14	0.00295499	3:0
7	11	437	642	11	996	98.51	0.00211264	3:0
8	10	405	624	7	995	98.79	0.00255601	3:0
9	6	410	805	7	1000	99.26	0.00629335	3:0

Die zusätzlichen Informationen

Wir betrachten die 1.Ausprägung der Variablen "Leistung"

Tabelle 7.4

Original	Bootstrap		zusätzliche Informationen							
	Konfidenzintervall		div	korg1	korg2	ku	ko	ekonfniv	a	Modus
Borg	unten	oben	*1	*2	*3	*4	*5	*6	*7	*8
Leistung 1										
0.0820	0.0164	0.1803	14	420	600	9	993	98.41	0.00182180	3:0

Die zusätzlichen Informationen sind folgende

- *1 = div = Zahl der unterschiedlichen Werte in der aufsteigend sortierten Reihe der Bootstrap-Koeffizienten (ASR).
Für Median und Quartile sollten mindestens 6-9 unterschiedliche Werte in ASR vorhanden sein.
- *2 = korg1 = das ist die untere Stelle, an welcher der Originalwert Borg1 in der aufsteigend sortierten Reihe der B Bootstrapwerte β zum 1. Mal erscheint
- *3 = korg2 = das ist die obere Stelle, an welcher der Originalwert Borg2 in der aufsteigend sortierten Reihe zum letzten Mal erscheint
- *4 = ku = das ist die (gerundete) Stelle der unteren Konfidenzgrenze ku in ASR
- *5 = ko = das ist die (gerundete) Stelle der oberen Konfidenzgrenze ko in ASR
- *6 = ekonfniv = das ist der Prozentsatz der Bootstrap-Stichprobenwerte zwischen den Konfidenzgrenzen (inklusive dieser).
Das ist das "effektive" Vertrauensniveau ekonfniv, das bei allen Modi einen anderen Wert besitzen kann als das vom Benutzer (üblicherweise mit 95%) vorgegebene
- *7 = a = das ist der "Akzelerationskoeffizient" a (aus Gleichung 13).
Beim BC-Verfahren ist a=0
- *8 = Modus = das ist der Modus der Behandlung des BC- bzw. BCA-Verfahrens
Beispiel: 3:1
Vor Doppelpunkt: Modus
Nach Doppelpunkt: l=effektives Vertrauensniveau an vorgegebenes angepasst

0=effektives Vertrauensniveau - nicht angepasst!

6.3.8 Das effektive Vertrauensniveau

Vorgegebenes und effektives Vertrauensniveau

Unter *6 bzw. **ekonfniv** wird das *effektive Vertrauensniveau* angegeben, das sich vom *vorgegebenen* Vertrauensniveau unterscheidet. Letzteres beträgt in unserem obigen Modus3-Beispiel 95%. Als effektives gibt Almo 98,41% aus. Wir können somit interpretieren: "Die Wahrscheinlichkeit, dass sich der (wahre) Populationswert im Intervall von **0.0164 bis 0.1803** befindet beträgt 98,41%. Das effektive Vertrauensniveau ist der prozentuale Anteil der Intervallbreite an der Zahl der Bootstrapstichproben.

$$\text{ekonfniv} = (\text{ko} - \text{ku} + 1) / B = (993 - 9 + 1) / 1000 = 98,50$$

Programmintern sind die unter *4 und *5 in der Tabelle 7.4 angegebenen Stellen **ku** und **ko** der Konfidenzgrenzen Dezimalzahlen. Almo rechnet mit diesen. In obiger Tabelle 7.4 werden **ku** und **ko** gerundet ausgegeben. Wird mit diesen gerechnet, so entsteht natürlich ein minimaler Unterschied.

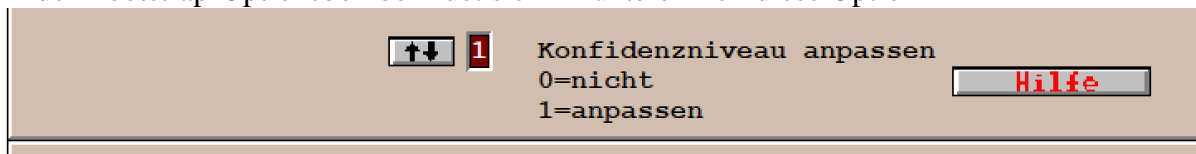
Prinzipiell gilt, dass wenn die Stelle **korg1** des unteren Originalwerts und die Stelle **korg2** des oberen Originalwerts nicht zusammen fallen, vorgegebenes und effektives Konfidenzniveau verschieden sind - und dies umso deutlicher je weiter **korg1** und **korg2** voneinander entfernt sind.

Effektives Vertrauensniveau an vorgegebenes anpassen

Vertrauensniveau und *Breite des Konfidenzintervalls* sind voneinander abhängig. Ein kleines Vertrauensniveau mit geringer Intervallbreite kann beim BC- bzw. BCa-Verfahren umgerechnet werden in ein höheres Vertrauensniveau mit größerer Intervallbreite - und umgekehrt. Wir können also das Vertrauensniveau von 98,41% in unserem Beispiel mit Modus 3 anpassen an das ursprünglich gewünschte Niveau von 95%. Wie das rechnerisch geschieht, ist oben in 6.1 Einschub 1 und 2 ausgeführt

Das Konfidenzintervall wird dadurch enger. Ein wesentlicher Vorteil des Anpassens ist, dass alle Variable (in unserem Beispiel: alle Anteilswerte der Variablen "Leistung") dasselbe Vertrauensniveau besitzen und dadurch vergleichbar werden.

In der Bootstrap-Optionsbox befindet sich im unteren Teil diese Option



Wird 1 eingesetzt, dann wird umgerechnet auf das ursprünglich gewünschte Vertrauensniveau (von beispielsweise 95%). Almo liefert dann folgendes Ergebnis für Modus 3 und 4. Wir zeigen das jeweils *ohne* und *mit* Anpassung nur für die 1. Ausprägung der Variablen "Leistung".

Tabelle 7.5

Original	Bootstrap		zusätzliche Informationen							
	Konfidenzintervall		div	korg1	korg2	ku	ko	ekonfniv	a	Modus
Borg	unten	oben	*1	*2	*3	*4	*5	*6	*7	*8

ohne Anpassung
Modus 3

0.0820	0.0164	0.1803	14	420	600	9	993	98.41	0.00182180	3:0
Modus 4										
0.0820	0.0328	0.1311	14	420	600	72	939	86.71	0.00182180	4:0

mit Anpassung

Modus 3										
0.0820	0.0164	0.1639	14	420	600	28	978	95,00	0.00182180	3:1
Modus 4										
0.0820	0.0164	0.1639	14	420	600	27	977	95.00	0.00182180	4:1

Ein *bedeutsames Ergebnis* ist, dass die Konfidenzgrenzen bei Modus 3 und Modus 4 in ihrem Wert β_u und β_o und ihrer Stelle k_u und k_o in ASR nach der Anpassung identisch sind. Das ist immer so. Sie sind aber nicht immer *exakt* ähnlich.

7 "Standardfehler-basiertes" Verfahren

Wir verwenden folgende Notation

B = Zahl der Bootstrapsstichproben
 β_{org} = origi...
 β_i = Koeffizient (z.B. Regressionskoeffizient) aus Bootstrapsstichprobe *i*
 β_M = Mittelwert aus den β_i -Werten aus den *B* (z.B. 1000) Bootstrapsstichproben
 β_u = unterer Grenzwert ...
 β_o = oberer Grenzwert
s = Standardabweichung der z.B. 1000 β_i -Werten aus den 1000 Bootstrapsstichproben
Dies ist beim Bootstrap gleich dem Standardfehler von β_{org}
konfniv = vom Benutzer vorgegebenes Vertrauensniveau (z.B. 95.0)
alpha = 1-konfniv/100

Aus '**konfniv**' erhalten wir sehr einfach '**alpha**' und den z-Wert von **alpha** gemäß folgender Gleichungen. (Wir geben zur Illustration die jeweiligen Ergebniszahlen für unser Beispiel aus Abschnitt 1 mit einem Vertrauensintervall von 95% an)

(0)	alpha	= 1-konfniv/100	(=0.05 auf alpha umgerechnetes Vertrauensniveau)
(1)	alpha/2	= (1-konfniv/100)/2	(=0.025)
(2)	zu	= z_von_p(alpha/2)	(= 1.96 z-Wert von alpha/2)
(3)	zo	= z_von_p(1-alpha/2)	(= -1.96 z-Wert von 1-alpha/2)

Beachte: **alpha/2** bzw. **1-alpha/2** wurde durch die **Almo-Funktion 'z_von_p'** von einem zwischen 0 und 1 liegenden p-Wert in einen z-Wert gewandelt, der zwischen minus-unendlich bis plus-unendlich liegt. Siehe dazu Abschnitt 6.1.

Die Standardabweichung *s* entsteht aus

$$(4) \quad S = \sqrt{\frac{\sum(\beta - \beta_M)^2}{B}}$$

$(\beta - \beta_M)^2$ wird über alle *B* Bootstrapsstichproben summiert

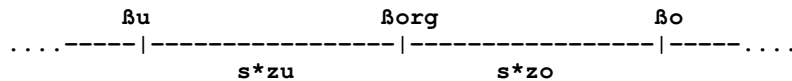
Das Vertrauensintervall besitzt dann diese Grenzwerte

(5)	unterer Grenzwert	$\beta_u = \beta_{org} - s \cdot zu$
(6)	oberer	$\beta_o = \beta_{org} - s \cdot zo$

'zo' ist immer negativ und als Absolutwert gleich 'zu'. Gleichung 6 kann also auch so geschrieben werden:

$$(6a) \beta_0 = \beta_{org} + s \cdot z_u$$

Als Schema dargestellt:



Wir erkennen, dass die geläufige Formel, mit der das Konfidenzintervall für einen Koeffizienten aus einer *einzelnen* Stichprobe (z.B. der Original-Stichprobe) geschätzt wird, auf das Bootstrap-Verfahren übertragen wird. Deswegen auch die Bezeichnung "**Standardfehler-basiertes Konfidenzintervall**". Die Information, die in der ASR, der aufsteigend sortierten Reihe der Bootstrap β -Werte wird in den Formeln 1 bis 6 nicht genutzt. Die Standardabweichung bzw. der Standardfehler s lässt sich auch aus den 1000 unsortierten Bootstrap β -Werte errechnen.

Gegen das "Standardfehler-basierte" Verfahren muss eingewendet werden, dass der Standardfehler (bzw. die Standardabweichung) bezüglich dem Mittelwert β_M aus alle Bootstrapstichproben berechnet wird und nicht bezüglich β_{org} . Die "Verzerrung" $\beta_{org} - \beta_M$ wird nicht korrigiert.

7.1 Der Signifikanz p-Wert

Unsere Vorgehensweise ist ähnlich der in Abschnitt 6.2, die wir dort für den p-Wert beim BCa-Verfahren vorgetragen haben. Wir ermitteln ein spezielles Konfidenzintervall für β_{org} , dessen unterer oder oberer Grenzwert den β -Wert =0 besitzt. In Abschnitt 6.2 haben wir dieses als "Null-Konfidenzintervall" bezeichnet. Der alpha-Wert dieses speziellen Intervalls ist dann der zweiseitige Signifikanzwert p für β_{org}

Die Werte von β_{org} und s sind aus der Berechnung des Konfidenzintervalls bereits bekannt

1. Für den Fall, dass β_{org} positiv, d.h. grössergleich 0 ist, setzen wir in Gleichung (5) β_u auf 0

$$\begin{array}{ll} \text{daraus folgt} & (7) \quad 0 = \beta_{org} - s \cdot z_u \\ & (8) \quad z_u = \beta_{org} / s \end{array}$$

'zu' wurde in (2) als z-Wert von $\alpha/2$ definiert. In Gleichung (8) kann also auf der linken Gleichungsseite auch gesetzt werden:

$$\begin{array}{ll} \text{daraus folgt} & (9) \quad z_{\text{von } p}(\alpha/2) = \beta_{org} / s \\ & (10) \quad p = \alpha = 2 * p_{\text{von } z}(\beta_{org} / s) \quad // \text{ p=Signifikanzwert} \end{array}$$

Beachte: β_{org} / s ist ein z-Wert. Daraus wird dann durch die Almo-Funktion ' $p_{\text{von } z}$ ' ein zwischen 0 bis 1 liegender p-Wert gewonnen. ' $p_{\text{von } z}$ ' ist die inverse Funktion zu ' $z_{\text{von } p}$ '. Siehe dazu Abschnitt 6.1 und Exkurs B

2. Für den Fall, dass β_{org} negativ, d.h. kleiner 0 ist, setzen wir in Gleichung (6) β_o auf 0

$$\begin{array}{ll} \text{daraus folgt} & (11) \quad 0 = \beta_{org} - s \cdot z_u \\ & (12) \quad z_o = \beta_{org} / s \end{array}$$

z_o wurde in (3) als z-Wert von $1-\alpha/2$ definiert. Es gilt also die Gleichung

(13) $z_{\text{von}_p}(1-\alpha/2) = \beta_{\text{org}}/s$
 daraus folgt (14) $p = 2 * (1-p_{\text{von}_z}(\beta_{\text{org}}/s))$

3. Da z_0 als Absolutwert gleich z_u ist, können die Gleichungen (10) und (14) in eine einzige zusammengefasst werden

(15) $p = \alpha = 2 * p_{\text{von}_z}(\text{abs}(\beta_{\text{org}}/s))$

abs=Absolutwert

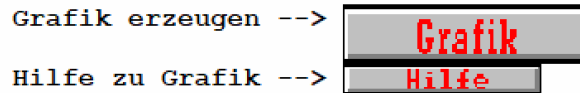
Die Signifikanz p kann also unabhängig vom Vorzeichen von β_{org} bestimmt werden.

A Exkurs: Die kumulative Standard-Normalverteilung.










(Ogive der Normalverteilung)

1. Die allgemeine kumulative Normalverteilung





Mit der Programm-Maske "Nonlin1.Gr" kann die Kurve der kumulativen Normalverteilung gezeichnet werden. Die Maske wird gefunden durch Klick auf den Knopf "alle Progs" am Oberrand des Almo-Fensters.





Kurventyp

0 = Gerade	$Y=a * X + \text{const}$	
1 = Parabel oder Hyperbel	$Y=a * X^{**}b + \text{const}$	
2 = Exponentialfunktion	$Y=a * e^{**}(b*x) + \text{const}$	
3 = allgem. Exponentialfunktion	$Y=a * b^{**}X + \text{const}$	
4 = Gompertz-Kurve	$Y=a * b^{**}(c^{**}X)$	
5 = Polynom 2. Grades	$Y=a*X + b*X*X + \text{const}$	
6 = Polynom 3. Grades	$Y=a*X + b*X*X + c*X*X*X + \text{const}$	
7 = logarithmische Funktion	$Y=a * \log_{10}(X) + \text{const}$	
8 = logistische Funktion I	$Y=1/(1/a+e^{**}-(b+c*X))$	
9 = Ogive / kumulative Normalverteilung		
10= logistische Funktion II	$Y=1/(1/a+e^{**}-(c*(b+X)))$	
15= inverse kumulative Standard-Normalverteilung		

Werte für die Parameter der Funktion

	1	a
	0	b
	-1	c
	0	const

Koordinatensystem

	3	x Länge der x-Achse
	1	y Länge der y-Achse

Nachdem die Maske geladen ist, muss auf Kurve Nr. 9 eingestellt werden und die Parameter festgelegt werden

Um die Kurve zeichnen zu können, verwendet Almo diese Formel:

$$Y = \text{Integral} (a \cdot c / 2.507 * e^{(-0.5 * (b + c * (X - \text{const}))^2)})$$

In übersichtlicher Form:

(E1)

$$y = \int \frac{a \cdot c}{2.507} \cdot e^{-0.5 \cdot (b + c \cdot (x - \text{const}))^2} dx$$

Die Integration erfolgt von minus unendlich bis X.

2.507 ist Wurzel aus $2 \cdot \pi$ (genauer auf 9 Stellen: 2.506 628 274)

Die Parameter der Gleichung

Jeder Parameter beeinflusst die Gestalt und die Lage der Kurve in einer für ihn typischen Art und Weise. Die nachfolgenden Angaben gelten unter der Annahme, dass die anderen Parameter konstant gehalten werden

a bestimmt die Obergrenze, der sich die Ogive annähert

b bestimmt die horizontale Lage der Kurve.

Je größer b umso weiter links liegt die Kurve - bei positivem c.
Ist c negativ dann umgekehrt weiter rechts.

c bestimmt die Steilheit der Ogive.

Je größer c absolut ist umso steiler verläuft die Kurve.

Ist das Vorzeichen positiv, dann wächst die Kurve von unten links nach rechts oben

Ist das Vorzeichen negativ, dann umgekehrt von links oben nach rechts unten

const verschiebt die Ogive nach rechts-

wenn a=1 und b=0 dann um den Betrag von const

X ist die maximale Länge der horizontalen x-Achse

bis zu der die Kurve gezeichnet werden soll.

Der Benutzer kann X beliebig festlegen

Y ist die Höhe der vertikalen y-Achse

bis zu der die Kurve gezeichnet werden soll.

Der Benutzer kann Y beliebig festlegen

Die maximale Höhe, die nicht überschritten werden kann ist jedoch durch den Parameter a festgelegt

2. Die kumulative *Standard*-Normalverteilung

erhält man mit

a = 1
b = 0
c = 1
const = 0
Xmax = 3 oder größer (x entspricht dem z-Wert)
Ymax = 1 (y entspricht dem p-Wert)

Werte für die Parameter der Funktion	
<input type="text" value="1"/>	a
<input type="text" value="0"/>	b
<input type="text" value="1"/>	c
<input type="text" value="0"/>	const

Koordinatensystem	
<input type="text" value="3"/>	x Länge der x-Achse
<input type="text" value="1"/>	y Länge der y-Achse

Die Gleichung (E1) vereinfacht sich zu

(E2)

$$p = \int \frac{c}{2.507} \cdot e^{-\frac{z^2}{2}} dz$$

das Integral reicht,

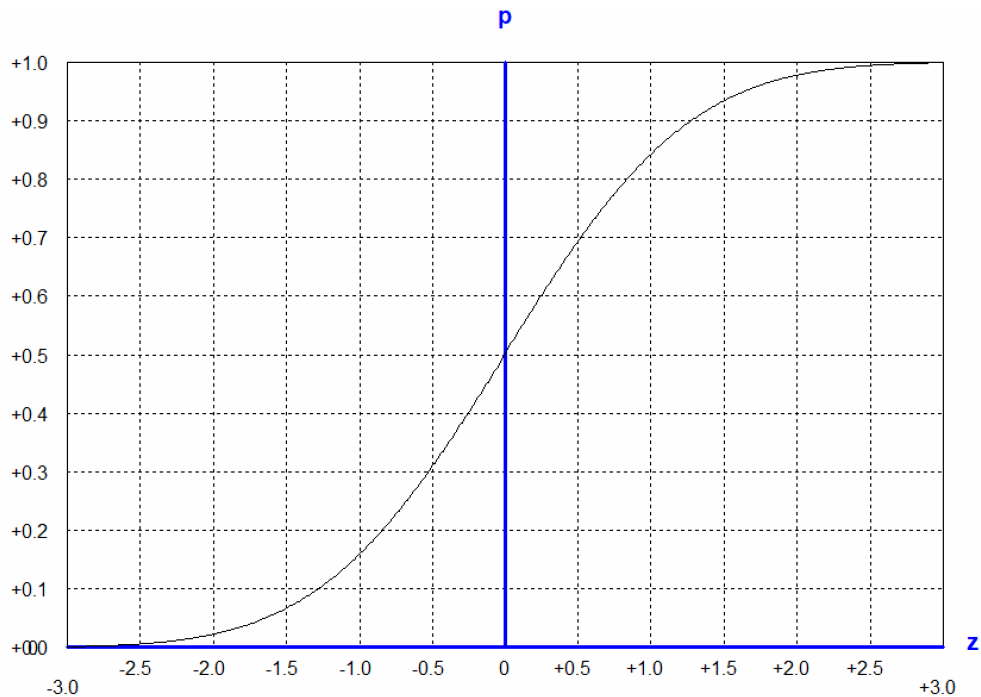
wenn $c=1$, von $-\infty$ nach z (untere bzw. linksseitige kumulative Standard Normalverteilung)

wenn $c=-1$, von z nach $+\infty$ (obere bzw. rechtsseitige kumulative Standard Normalverteilung).

Dieser Fall wird in folgendem Punkt 3 behandelt.

Die abhängige Variable y kann jetzt als p -Wert begriffen werden, deren Wert auf 0 bis 1 beschränkt ist und die unabhängige Variable als z , deren Wert zwischen $-\infty$ und $+\infty$ liegen kann.

Es entsteht folgende Kurve



Die Kurve besitzt ihren Wendepunkt bei $z=0$. p ist dann $=0.5$. Auf der linken Seite nähert sich die Kurve $z= - \text{unendlich}$ und auf der rechten Seite $z= + \text{unendlich}$ an. In der Grafik ist ab ca. $z=-3$ oder $+3$ kein Unterschied in z mehr wahrnehmbar.

3. Die Kurve der "kumulativen *Standard-Normalverteilung* mit negativer Steigung"

erhält man, wenn man die obigen Parameter beibehält, den Steigungsparameter jedoch auf $c = -1$ setzt. Dadurch verläuft die Kurve von links oben nach rechts unten

Die Almo-Funktion "**p_von_z**" ist eine Almo-Software-Funktion, die aus der *kumulativen Standard-Normalverteilung* mit negativer Steigung nach Eingabe eines z -Wertes den p -Wert ausgibt. Die Almo-Funktion wird im BC-bzw. BCa-Kalkül in Abschnitt 6.1 für die Gleichungen 7 und 8 gebraucht. In Exkurs B wird das C-Programm für "**p_von_z**" und "**z_von_p**" vorgestellt. In Almo sind im Ordner "**Algorithmen_in_C/Algorith_C/a_up_algorithm2.c**" diese Funktionen in der Programmiersprache C enthalten.

Der Benutzer kann für jeden beliebigen z -Wert den zugehörigen p -Wert ermitteln. Beide Werte bilden dann einen Punkt auf der nachfolgend abgebildeten Kurve. Der Benutzer muss in Almo auf den Knopf „stat.Tafelwert“ (am Oberrand des Almo-Hauptfensters) klicken und dann **Prog00t3.Msk** laden. Dieses Programm verwendet die Funktion "**p_von_z**".

Prog00t3.Msk

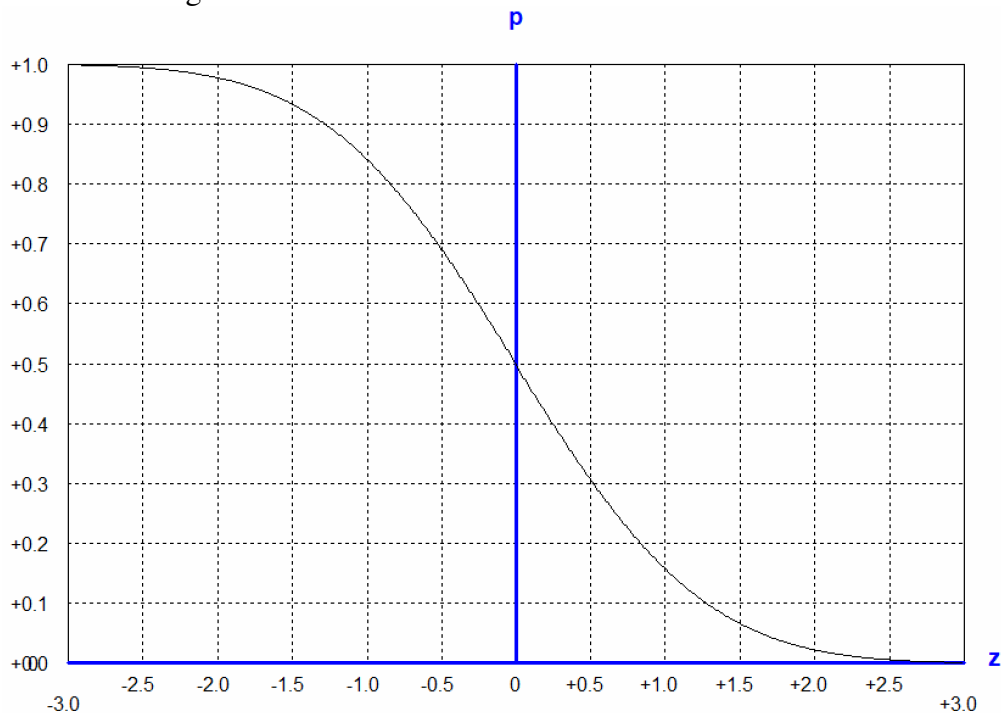
Signifikanz von z
gegeben z-Wert; gesucht p

zu testender z-Wert

Eingabe: $z = 2.01842$

Ergebnis: $p = 0.02179$ (einseitig)

Wichtig ist festzuhalten: Der *Input* in die Funktion `p_von_z` ist z . Der *Output* ist p .
Es entsteht folgende Kurve



4. Die Kurve der "inversen kumulativen Standard-Normalverteilung" (mit negativer Steigung)

Wenn der Benutzer diese Kurve erzeugen will, dann muss er in der Programm-Maske "Nonlin1.Gr1" die **Kurven-Nr. = 15** setzen und folgende Werte für die Parameter eingeben.

```

a      = 1
b      = 0
c      = -1
const  = 0
Xmax   = 1   (x entspricht dem p-Wert)
ymax   = 3   (y entspricht dem z-Wert)

```

Werte für die Parameter der Funktion

↔	↔	↔	↔	1	a
↔	↔	↔	↔	0	b
↔	↔	↔	↔	-1	c
↔	↔	↔	↔	0	const

Koordinatensystem

↔	↔	1	x Länge der x-Achse
↔	↔	3	y Länge der y-Achse

Diese Kurve ist im Kurvenverlauf identisch mit der in Punkt 3 abgebildeten "kumulativen Standard-Normalverteilung mit negativer Steigung". Das Koordinatensystem ist jedoch verdreht. Die p -Achse ist die horizontale Achse und die z -Achse ist die vertikale Achse.

Die Gleichung dieser Kurve ist die Umkehrfunktion zu obiger Gleichung (**E2**). Der Begriff "*invers*" sollte vielleicht besser durch "umgekehrt" ersetzt werden.

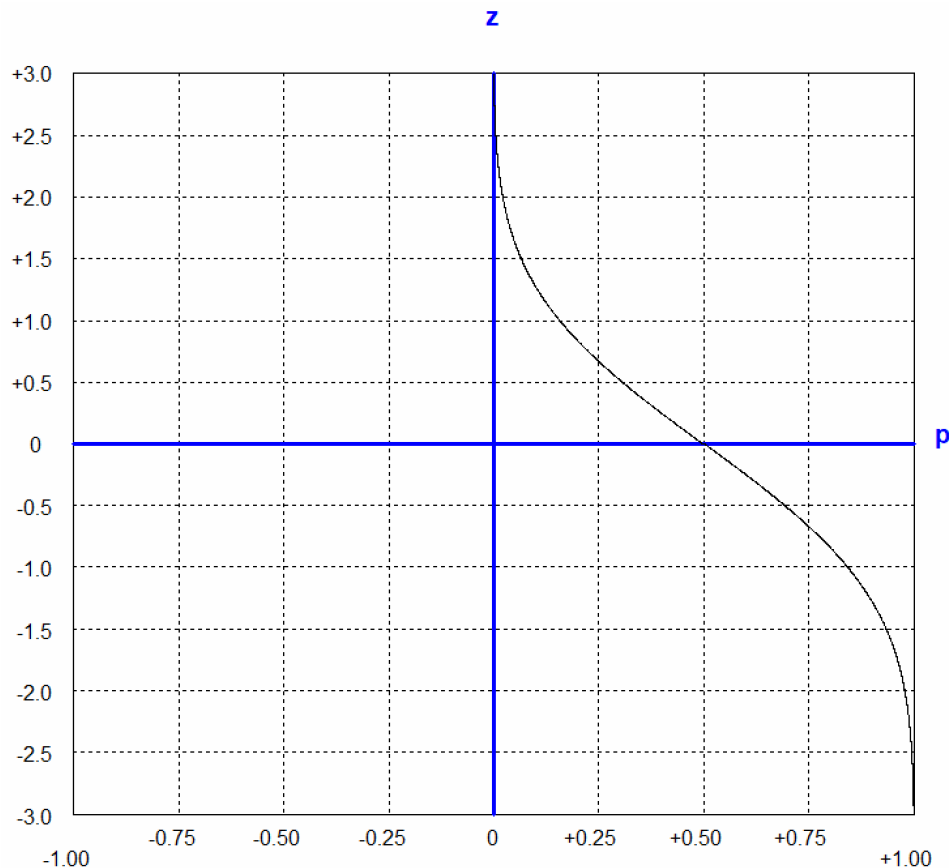
Die Kurve entsteht (grafisch) auf folgende Weise:

Das Koordinatensystem der "kumulativen Standard-Normalverteilung mit negativer Steigung" ($c = -1$) wird

(1) um 90 Grad nach links gedreht. Die neuen Koordinaten (x,y) für jeden Kurvenpunkt entstehen durch Vertauschen der seitherigen Koordinaten: $(-y,x)$

(2) danach wird um die (neue) vertikale Achse gespiegelt. Jeder neue x -Wert wird mit -1 multipliziert

Dies ist die Kurve



Die *ausgefüllte* Programm-Maske für oben abgebildete Grafik kann auch unmittelbar geladen und gestartet werden. Nach Klick auf "alle Progs" muss noch die Maske "invKumStNormal.grf" selektiert werden.

Die Funktion "**z_von_p**" ist eine Almo-Software-Funktion, die aus der "*inversen kumulativen Standard-Normalverteilung* mit negativer Steigung" den **z**-Wert von **p** ausgibt. Die Almo-Funktion wird im BC-bzw. BCa-Kalkül in Abschnitt 6.1 für die Gleichungen 2 gebraucht.

In Almo ist im Ordner "**Algorithmen_in_C/Algorith_C/a_up_algorith2.c**" die Funktion "**z_von_p**" unter dem Namen "**z_wert4**" in der Programmiersprache C enthalten. Verwendbar ist auch die in Almo enthaltene Funktion "**t_wert**", die bei Stichprobengrößen von 1000 und mehr ein identisches Ergebnis bis mindestens zur 4 Kommastelle erbringt. In den beiden Almo-Funktionen muss als 2. Parameter "einseitig" eingegeben werden.

Die Funktion "z_von_p" kann durch Klick auf den Knopf **stat. Tafelwerte** in der Knopfleiste am Oberrand von Almo aufgerufen werden. Dann Prog00t6 oder Prog00t7 wählen. Der Benutzer erhält dann folgende Programm-Maske

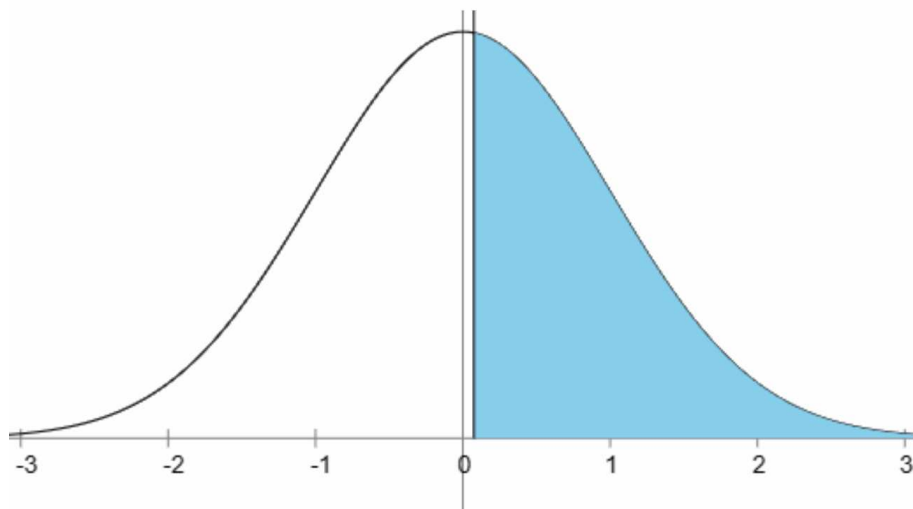
Prog00t7.Msk

Gegeben: Irrtumswahrscheinlichkeit p	
Gesucht: z	<input type="button" value="Hilfe"/>

↔	0.488511	p-Wert	
↑↓	1		1 =einseitig 2 =zweiseitig

Gesucht: z
 Eingabe: p=0.488511 einseitig
 Ergebnis: z-Wert = 0.028804

Im Internet sind mehrere *statistische online-Kalkulatoren* verfügbar, mit denen ebenfalls der z-Wert aus dem gegebenen p-Wert (und umgekehrt) errechnet werden kann - beispielsweise jener der "University of Illinois at Urbana-Champaign" (siehe Literaturliste). Dort wird auch die folgende Standard-Normalverteilungs-Kurve abgebildet, die wir hier wiedergeben.



Auf der horizontalen x-Achse sind die z-Werte abgetragen - hier nur von -3 bis +3. Die *rechte*, blau eingefärbte Fläche ("upper cumulative") unter der Standard-Normalverteilungs-Kurve entspricht einem Anteil von 0,488511 der Gesamtfläche. Die Fläche beginnt rechts bei **+unendlich**. Sie wird dann nach links begrenzt, wenn sie einen Anteil von $p=0,488511$ erreicht hat. Dies geschieht bei einem z-Wert von **0.02881**. Unserer Funktion „z_von_p“ ermittelt diesen z-Wert.

Die weiter oben abgebildete Kurve der "inversen kumulativen Standard-Normalverteilung mit negativer Steigung" gibt den z-Wert abhängig von p aus. Der Punkt **p=0.488511, z=0.02881** auf der Kurve entspricht der blauen Fläche in der hier abgebildeten Grafik.B

B Exkurs: C-Programm für "z_von_p" und "p_von_z"

```
#define begin      {
#define end        }
```

```

#define endif      }
#define endelse   }
#define endfor    }

#include <math.h>      //weitere include-files werden gebraucht

double ex_eps = 2.22044605e-14; //globaler Grenzwert

/*-----
die Funktionen "p_von_z" und "z_von_p" verwenden folgende Funktionen
gammafunc      Gamma-Funktion
prob           ermittelt Signifikanz p von F, t, z, Chi-Quadrat
               (geringere Genauigkeit)
exaktprob      ermittelt Signifikanz p von F, t, z, Chi-Quadrat
               (gute Genauigkeit)
probinvers     ermittelt p aus F, t, z, Chi-Quadrat. Inverse Funktion zu
               'exaktprob'
-----*/

/*-----*/
/*      gammafunc      */ //wird in 'exaktprob' gebraucht
/*Gamma-Funktion fuer      */
/*ny=1,2,3,4,...      */
/*ny=0.5, 1.5, 2.5, 3.5,...*/
/*-----*/
double gammafunc(double ny)
begin
int i,n;
double y,z;

if(ny > 170.6) return -1.0;

z= floor(ny);
if(ny-z < ex_eps)      /*ny=1 2 3 ...*/
y=ffak(ny-1.0);
else begin      /*ny=0.5 1.5 2.5 ...*/
n =(int)floor(ny+0.1);
y=1.772453851;      /*sqrt(pi)*/
z=0.5;
for(i=1; i<=n; i++) begin
y *= z;
z++;
endfor
endelse
return(y);
end /*gammafunc*/

/*-----*/
/*      prob      */
/* Signifikanz von F, t, z, Chi-Quadrat      */
/* mit geringer Genauigkeit. Wird nur bei      */
/* extremen Werten anstelle 'exaktprob'      */
/* verwendet. Aufruf erfolgt in 'exaktprob'      */
/*      */
/* nach D.J.Veldmann: Fortran programming for      */
/* the behavioral sciences,      */
/* Holt,Rinehardt and Winston,N.Y. 1967, S. 129*/
/* Von Fortran auf C uebertragen      */
/*      */
/*Eingabe fuer:      */
/* F:      prob(df,dfe,f)      */

```

```

/* Chi-Quadrat:   prob( df, 1000, chi/df)          */
/* t:             prob( 1, dfe, t*t)              */
/* z:             prob( 1, 1000, z*z)             */
/*                                                       */
/*Return: p                                             */
/*-----*/
double prob(int df,int dfe,double f)
begin
static int   umdreh,hdf,hdfe;
static double a,b,z,dfr,kor,p,zz,z2,z3;

umdreh=0;
hdf   =df;
a     =df;
hdfe  =dfe;
b     =dfe;
dfr   =b;

if(f <= .0) begin //negativ nicht zulaessig !
  p=1.0;
  goto raus;
endif
if( fabs(a*b*f) < ex_eps ) begin
  p=1.0;
  goto raus;
endif
a=0.2222222222 / a;
b=0.2222222222 / b;

z= ( (1.0-b) * pow(f,0.3333333333) -1.0+a)
  / sqrt(b* pow(f,0.6666666667)+a);
if( dfe < 4 ) begin
  kor = z * 0.08 * pow( fabs(z),4.0) / pow(dfr,3.0);
  z += kor;
endif
zz=z;
if(z < 0) begin
  umdreh=1;
  z= -z;
endif

p=0.5e0/ pow((1.0+z*(0.196854+z*(0.115194+
  z*(0.000344+z*0.019527))),4.0);

if(umdreh) p=1.0-p;

/*-- Korrektur --*/
if(hdf==1) begin                               /*nur wenn df=1, bei t und z*/
  if(hdfe <= 6) goto raus;                     /*erst ab dfe >= 7 */
  if(zz > 3.6) begin                           /*oberer Rand*/
    p=0.0;
    goto raus;
  endif
  if(zz > 2.5) begin
    p -= (-4.3355*zz + 0.7057*zz*zz + 9.2441) / 10000.0;
    goto raus;
  endif
  if(zz > 2.338) goto raus;
  //Bereich zwischen 2.3338 und 2.5 braucht keine Korrektur
  if(zz > 1.3112) begin
    p += (-3.9232*zz + 9.2441) / 1000.0;
    goto raus;
  endif
endif

```



```

if(zz > 0.5132) begin
  z2=zz*zz; z3=z2*zz;
  p += (32.6684*zz - 20.2055*z2 + 3.7958*z3
        - 12.6404) / 1000.0;
  goto raus;
endif
if(zz > -.53) begin
  z2=zz*zz; z3=z2*zz;
  p += (5.6326*zz + 24.1465*z2 - 17.692*z3 - 7.3155) / 1000.0;
  goto raus;
endif

p += (-55.8702*zz - 31.3897) / 1000.0;
if(p > 1.0) p=1.0;
endif
/*-- Ende Korrektur --*/

raus:
return(p);
end /* prob */

/*-----*/
/*          exaktprob          */
/* Signifikanz p des F-,t-,z-,Chi-Quadrat Werts*/
/* mit maximaler Genauigkeit  */
/*                               */
/* nach O.G.Ludwig, Communications of the ACM, */
/* 6, June 1963,Algorithm 179      */
/* https://dl.acm.org/doi/10.1145/366604.366633*/
/* Von Algol auf C uebertragen    */
/*                               */
/* Eingabe fuer:                  */
/* F:          exaktprob(df,dfe,f)  */
/* t:          exaktprob(1,dfe,t*t) */
/* z:          exaktprob(1,3000,z*z) */
/* Chi-Quadrat: exaktprob(df,3000,chi/df) */
/*                               */
/* Return: p                       */
/*                               */
/* Beachte: p fuer t oder z kann negativ werden*/
/* dann muss ausserhalb 'exaktprob' noch gesetzt*/
/* werden                           */
/*      bei zweiseitig   p=1.0-p     */
/*      bei einseitig    p=1.0-p/2   */
/*-----*/
double exaktprob(int k1,int k2,double f)

begin
static int  alter;
static double p,q,x,y,term,term1,temp,temp1,index,qrecur,finsum,infsum,h1,
             maxdouble=h_maxfloat;

if(f <= .0) begin temp=1.0; goto fertig; endif

h1=k1*f;
if(fabs(h1*k2) < ex_eps) begin temp=1.0; goto fertig; endif

x = k2/(k2+h1);
p = k2/2.0;
q = k1/2.0;

if(x <= 0.5)
  alter=0;

```

```

else begin
    alter =1;
    temp=p;
    p=q;
    q=temp;
    x=1.0-x;
endelse

finsum= 0.0;
term=1.0;
temp=1.0-x;
qrecur=index=q;

noch1:
index -= 1.0;
if(index > ex_eps) begin
    qrecur = index;
    term  *= (qrecur+1)/(temp*(p+qrecur));
    finsum += term;
    goto noch1;
endif

infsum= term= 1.0;
index= .0;
noch2:
if(term > ex_eps) begin
    index++;
    term *= x * (index-qrecur) * (p+index-1.0) / (index * (p+index));
    infsum += term;
    goto noch2;
endif

temp=temp1=gammafunc(qrecur);          /*overflow moeglich*/
term=term1=gammafunc(qrecur+p);       /*overflow moeglich*/

h1=term1/temp1;

index=qrecur;
noch3:
if(index <= q-0.5) begin
    y      = (index+p)/index;
    if(maxdouble/y < h1) goto rechne_mit_prob; /*h1 haette overflow*/
    h1     *= y;
    index += 1.0;
    goto noch3;
endif

if(finsum > ex_eps &&
    maxdouble/finsum < h1) goto rechne_mit_prob; /*h1 haette overflow*/
if(p > 170.6) goto rechne_mit_prob;          /*overflow moeglich*/

temp= pow(x,p) *
      ( infsum * term / (p*temp) + finsum * h1 * pow((1.0-x),q) / q )
      / gammafunc(p);

if(alter)
    temp=1.0-temp;

fertig: ;
return(temp);

rechne_mit_prob:          /*exaktprob nicht moeglich, rechne mit prob*/
temp=prob(k1, k2, f);

```

```

return(temp);

end /*exaktprob*/

/*-----*/
/*          probinvers          */
/*Eingabe: Irrtumswahrscheinlichkeit p0 (0 - 1) */
/*Return:  F, t, z, Chi-Quadrat          */
/*Ausgabeparameter: Zahl der Iterationen zz          */
/*          */
/*Aufruf fuer Return von          */
/* F:  probinvers(p0,df,dfe, 1.0e-12,exaktprob,&zz)*/
/* Chi:probinvers(p0,df,1000,1.0e-12,exaktprob,&zz)*/
/* t:  probinvers(p0,1, dfe, 1.0e-12,exaktprob,&zz)*/
/* z:  probinvers(p0,1, 1000,1.0e-12,exaktprob,&zz)*/
/*          */
/* t,z ist dann =sqrt(probinvers(..))          */
/* Chi          =df * probinvers(..)          */
/* F          =probinvers(..)          */
/*          */
/* zz ist Zahl der Iterationen des Kalkuels          */
/*          Kurt Holm          */
/*-----*/
double probinvers(double p0,int df,int dfe,double epsilon,double
(*prob)(),int *zz)

/*double (*prob)() ist Zeiger auf eine der 2 in ALMO enthaltenen prob-
Funktionen: prob oder exaktprob*/

begin
int    z;
double f,p,anf,abstand;

z=0;
anf=0.0;

f=16.0;          /*zuerst mit F=16 probieren*/
p=(*prob)(df,dfe,f);
if(p0 >= p)
    abstand=f;

else begin          /*F zu klein, mit 65536 beginnen*/
    f=65536.0;
    p=(*prob)(df,dfe,f);
    if(p0 < p) return(66000.0); /*Signal, dass f > 65536*/
    abstand=f;
endelse

noch:
    abstand /= 2.0;
    f=anf+abstand;
    p=(*prob)(df,dfe,f);
    if(p > p0) anf=f;
    z++;

if(z>80) goto raus;          /*mit bis jetzt errechnetem f-Wert abbrechen*/

if( fabs(p-p0) > epsilon ) goto noch;

raus: *zz=z;
return(f);
end /*probinvers*/

```

```

/*-----*/
/*          z_von_p          */
/*Eingabe:          */
/* p = einseitig oder zweiseitig */
/* Return:  z-Wert      */
/*          Kurt Holm    */
/*-----*/
double z_von_p(double p,
               int    seitig) // 1=einseitiger Vertrauensbereich
2=zweiseitiger
begin
int    it;
double z;
double vor;
vor=1.0;
if(seitig==1 && p>0.5)  p=1.0-p; vor= -1.0;

if(seitig==1) p *= 2.0;
if(p >= 1.0)  p=1.0; return .0;
z=probinvers(p, 1, 6000, 1.0e-12, exaktprob, &it);
z=sqrt(fabs(z));
z *= vor;
return(z);
end /*z_von_p*/

/*-----*/
/*          p_von_z          */
/*Eingabe:          */
/* z = einseitig oder zweiseitig */
/* Return:  p-Wert      */
/*          Kurt Holm    */
/*-----*/
double p_von_z( double z,          //einseitiger oder zweiseitiger z-Wert
                int    einseitig) //1=einseitig 2=zweiseitig
begin
double p1;
p1 = exaktprob(1, 6000, z*z);
if (einseitig==1) p1 /= 2.0;
if(z<.0) p1=1.0-p1;
return p1;
end

```

Literatur

Davison, A.C. & Hinkley, D.V.: Bootstrap methods and their application, 2006, 8th Edn. Cambridge University Press

Thomas J. DiCiccio and Bradley Efron: Bootstrap Confidence Intervals, Statistical Science 1996, Vol. 11, No. 3, 189-228. Download im Internet mit dem Namen der Autoren als Suchwort. Ausführliche und anspruchsvolle Darstellung des BCa-Verfahrens

Elias, Christopher J.: Percentile and Percentile-t Bootstrap Confidence Intervals: A Practical Comparison, Journal of Econometric Methods, 2013, Band 4, Heft 1, S 153-161

Statistische Online Kalkulatoren:

Keisan online calculator. <https://keisan.casio.com/exec/system/1180573165>

University Illinois <http://courses.atlas.illinois.edu/spring2016/STAT/STAT200/pnormal.html>

Zhang, Preacher, Luo: Bootstrap Confidence Intervals for Ordinary Least Squares Factor and Correlations in Exploratory Factor Analysis, *Multivariate Behavioral Research*, 2010,45:1, S.117 ff., <https://doi.org/10.1080/00273170903504836>

Siehe auch die Darstellung "Chapter 11 The Bootstrap" der Carnegie Mellon University im Internet: <https://www.stat.cmu.edu/~larry/=sml/Boot.pdf>