# Das Ziehen von Stichproben mit Hilfe des Programmpakets OSIRIS

Kirschner, Hans-Peter

Veröffentlichungsversion / Published Version
Zeitschriftenartikel / journal article

## DAS ZIEHEN VON STICHPROBEN MIT HILFE DES PROGRAMMPAKETS OSIRIS

Sowohl in der Forschungspraxis als auch in der Lehre sind häufig Stichproben unterschiedlichster Art aus maschinenlesbaren Datensätzen zu ziehen. Leider wird dies von den für die Sozialwissenschaften konzipierten Programmpaketen nicht ausreichend unterstützt. Im folgenden wird daher anhand von exemplarischen Beispielen demonstriert, wie man dennoch mit Hilfe der Recode-Syntax von OSIRIS III, Release 2, die gebräuchlichsten Stichprobenpläne in eine maschinelle Ziehungsprozedur umsetzen kann.

### 1. Grundsätzliches

Bei der Durchführung von Forschungsprojekten in den Sozialwissenschaften muß häufig aus einem maschinenlesbaren Datensatz eine mehr oder weniger komplexe Zufallsstichprobe (maschinell) gezogen werden. Anders als bei der Datenanalyse werden jedoch solche Aktivitäten von den gebräuchlichsten Programmpaketen BMDP, OSIRIS, SAS und SPSS nicht ausreichend unterstützt. Das führt häufig zu unbefriedigenden Problemlösungen. Denn - auf sich allein gestellt - wird der Forscher in den Sozialwissenschaften immer in der Gefahr sein, bei einer wenig adäquaten und mehr behelfsmäßigen Lösung stehenzubleiben, da von ihm selbstverständlich nicht das notwendige mathematisch-statistische und computertechnische Wissen erwartet werden kann, das zu einer wirklich angemessenen Lösung erforderlich ist. Aber selbst wenn eine Möglichkeit der Beratung und auch Hilfe von Spezialistenseite besteht, kommt es oft zu großen Reibungsverlusten bei der Transformation der notwendigerweise mehr inhaltlich betonten Ideen des Sozialwissenschaftlers in die ebenso notwendigerweise formale Sprache der Statistik und insbesondere der Programmierung. In diesen Fällen wäre dann die Problemlösung vielleicht technisch anspruchsvoller, würde aber u.U. den ursprünglichen Intentionen ebensowenig entsprechen wie die eigene behelfsmäßige Lösung des Forschers.

Wie auf den Gebieten des Datenmanagements bzw. der Datenanalyse sollte also auch für das Ziehen von Stichproben (zumindest) ein Programmpaket zur Verfügung stehen, dessen Handbuch in der Sprache der Sozialwissenschaften abgefaßt ist und das den einzelnen Forscher sowohl von den Überlegungen zur EDV-technischen Umsetzung der Ziehungsalgorithmen als auch von der Dateiverwaltung entlastet.

Ein solches Programmpaket existiert nicht. Es besteht daher bei ZUMA die Absicht, ein mit den existierenden Analysepaketen kompatibles System zu implementieren, das auf einfache Weise die Ziehung aller wesentlichen Arten von Zufallsstichproben gestattet.

Selbstverständlich waren im Rahmen der Dienstleistungsarbeiten und der eigenen Grundlagenforschung bei ZUMA bisher häufig - ohne die Hilfestellung eines Stichproben-Programmsystems - Zufallsstichproben aus maschinenlesbaren Dateien zu ziehen (vgl. ZUMANACHRICHTEN 3, S. 37f.). Dies geschah unter Ausnutzung verschiedener Optionen des Programmpakets OSIRIS. Die folgenden Abschnitte sollen in exemplarischen Beispielen die dabei gewonnenen Erfahrungen darstellen und damit für den Forscher in den Sozialwissenschaften die Möglichkeit schaffen, zunächst ohne ein speziell zugeschnittenes System, aber dennoch innerhalb eines anerkannten Analysepakets, selbständig Problemlösungen auf dem Gebiet der Zufallsstichproben anzugeben. Es ist zu hoffen, daß so das eingangs geschilderte Dilemma ein wenig gemildert werden kann und sich die "apparative Ausstattung" auf diesem Gebiet der recht komfortablen Situation auf den Gebieten des Datenmanagements und der Datenanalyse etwas annähert.

## 2. Ziehungsalgorithmen

Das Ziehen einer Zufallsstichprobe wird häufig modellhaft so erläutert, daß irgendwie numerierte Kugeln "blind" aus einem Hut gezogen werden. Ganz wesentlich dabei ist, ob eine Kugel, nachdem sie gezogen und nachdem ihre Nummer notiert wurde, wieder in den Hut zurückgelegt wird oder nicht; d.h. ob man "mit Zurücklegen" oder "ohne Zurücklegen" zieht. Es ist nun offensichtlich, daß man als äquivalente modellhafte Vorstellung für diese Ziehungsprozeduren sich alle Kugeln auf einer Schnur aufgereiht denken kann (in beliebiger Reihenfolge) und von links nach rechts durchlaufend für jede Kugel eine vom Zufall abhängige Entscheidung trifft, ob sie in der Stichprobe vertreten sein soll oder nicht. Man würde mit Zurücklegen ziehen, wenn man diesen Durchlauf so oft wiederholt, wie es die Stichprobengröße angibt, und pro Durchlauf genau eine positive Entscheidung trifft. Das Ziehen ohne Zurücklegen - also unter Ausschluß von Duplikaten in der Stichprobe - würde bedeuten, daß in einem Durchlauf die Anzahl der positiven Entscheidungen gleich der Stichprobengröße ist.

Mann kann also das Ziehen einer Stichprobe sehr plausibel sequentiell darstellen. Von besonderem praktischen Interesse sind dabei Prozeduren, die dem Ziehen ohne Zurücklegen entsprechen, d.h. die nach möglichst nur einem "Abgehen" der Kugeln die fertige Stichprobe liefern, in der dann keine Duplikate enthalten sind.

Eine solche sequentielle Arbeitsweise der Prozedur ist nämlich besonders günstig auf den üblichen digitalen Großrechnern zu installieren. "In computer applications the sequential procedures are the most commonly employed due to the nature of storage and access of large files of data" (KENNEDY & GENTH, 1980). Ersetzt man "Kugel" durch "Fall", ist der Zusammenhang offensichtlich. Es sei im übrigen angemerkt, daß alle Analyseprogramme in den großen Programmpaketen die Dateien sequentiell abarbeiten und wo eben möglich auf einen einmaligen Durchgang durch die Fälle (= Kugeln) ausgelegt sind.

In den folgenden Abschnitten werden bis auf zwei Ausnahmen Techniken des Stichprobenziehens vorgestellt, die streng sequentiell angelegt und in der Sprache von OSIRIS III, Release 2, geschrieben sind. Natürlich ist es gerade wegen des sequentiellen Charakters dieser Techniken möglich, auch andere Programmsysteme zu verwenden. Die Entscheidung zugunsten von OSIRIS fiel im wesentlichen wegen der folgenden zwei Gründe:

- Von den zur Zeit verfügbaren Programmpaketen hat OSIRIS III, Release 2, eine für das Ziehen von Stichproben besonders geeignete und recht bequeme Syntax der Recode-Sprache und

- von denjenigen Paketen mit einer geeigneten Recode Sprache (wie z.B. SAS) scheint OSIRIS III, Release 2, am wenigsten rechnerabhängig zu sein.

Die Recode-Möglichkeiten der verwendeten OSIRIS-Version spielen insofern eine zentrale Rolle, als der Entscheidungsmechanismus für die Annahme oder Ablehnung einer "Kugel", also eines Falls, "programmiert" werden muß. Besonders einfach ist dies z.B. für das sog. "binomial sampling" (vgl. ZUMANACHRICHTEN 3, S. 34). Dort wird Fall für Fall eine Zufallszahl zwischen Null und Eins erzeugt und geprüft, ob diese kleiner oder gleich einer vorgegebenen Konstanten ist. Liefert die Entscheidungsregel die Antwort "Ja", erhält

der Fall in geeigneter Form eine Kennung, die ihn als zur Stichprobe gehörig ausweist. Ein Befehl vom Typ der "IF-Anweisung" reicht also hier völlig aus.

Die Entscheidungsregel für das "binomial sampling" zeichnet sich durch ihre "Gedächtnislosigkeit" aus: Das Resultat der Regel für den k-ten Fall ist offensichtlich unabhängig davon, wie in den Fällen 1,...,k-1 entschieden wurde.

Höhere Anforderungen an die Recode-Syntax stellen demgegenüber Algorithmen, die "mit Gedächtnis" arbeiten. Völlig analog zum "binomial sampling" wird zwar dabei Fall für Fall eine Entscheidungsregel abgefragt, jedoch ist der Ausgang der Entscheidung abhängig von den Entscheidungen, die für die zuvor abgearbeiteten Fälle getroffen wurden. Algorithmen dieser Art werden in den Abschnitten 3.1-3.4 verwendet (vgl. ZUMANACHRICHTEN 3, S. 40).+

Alle diese Entscheidungsregeln arbeiten mit Hilfe eines Zufallszahlengenerators, d.h. es wird ein Programm verwendet, das eine beliebig lange Folge von Zahlen produzieren kann, für die gilt, daß - grob gesagt - statistische Tests keine "Nicht-Zufälligkeiten" entdecken können (vgl. ZUMANACHRICHTEN 3, S. 31). Leider gibt es zwischen den verschiedenen zur Zeit programmierten Zufallszahlengeneratoren Qualitätsunterschiede; hinzu kommt, daß viele dieser Generatoren maschinenabhängig sind, also nicht ohne weiteres implementiert werden können. Diese Situation ist natürlich unbefriedigend, da gerade die Qualität der zugrundeliegenden Folge von Zufallszahlen beim Ziehen einer Zufallsstichprobe von entscheidender Bedeutung ist. Es wurde daher eine weitestgehend maschinenunabhängige Version des von KNUTH (1969: 30f.) empfohlenen "Algorithm M" bei ZUMA (in der Programmsprache FORTRAN) geschrieben (vgl. ZUMANACHRICHTEN 3, S. 31). Dieses Programm wurde inzwischen in sehr vielen Anwendungen erprobt, und einige der erzeugten Folgen von Zufallszahlen wurden statistischen Tests unterworfen. Es hat sich gezeigt, daß dieser Generator den anderen bei ZUMA verfügbaren Generatoren zumindest ebenbürtig ist. (Interessenten können eine Bandkopie oder eine Liste des Programms bei ZUMA anfordern.)

+ Für diese Algorithmen benötigt die Recode-Syntax Anweisungen vom Typ "GO TO", und sie sollte Möglichkeiten zur indirekten Adressierung von Variablen enthalten.

## 3. Exemplarische Beispiele

Es werden in den Abschnitten 3.1-3.4 Algorithmen vorgestellt, die Fall für Fall entscheiden, ob dieser zur Stichprobe gehört oder nicht. Selbstverständlich ist diese einfache Logik des Vorgehens in den angegebenen "Setups" allein schon deshalb nicht ohne weiteres zu erkennen, da - insbesondere in Abschnitt 3.2 - durch eine gewollte Realitätsnähe der Code recht umfänglich und komplex ist. Es sei aber betont, daß alle Recode-Anweisungen, die um den Kern der Ziehungsprozedur "herumgeschrieben" sind, im einzelnen außerordentlich naheliegen und dem erfahrenen OSIRIS-Benutzer keine Schwierigkeiten bereiten.

Im Abschnitt 3.5 werden zwei Verfahrensweisen dargestellt, mit deren Hilfe Stichproben aus Klumpen (= clusters) gezogen werden können.

### 3.1. Ziehen und Nachziehen aus großen Dateien

Gegeben ist eine Datei mit 100.000 Fällen und 18 Variablen. Es existiert ferner zu diesem Datensatz ein OSIRIS-Dictionary. Die Aufgabe besteht zunächst darin, eine Zufallsstichprobe der Größe 6500 mit Hilfe einer modifizierten Form des in den ZUMANACHRICHTEN 3, Anhang C, S. 40, beschriebenen Algorithmus zu ziehen. Die Modifizierung bedeutet, daß nicht pro Fall in der großen Datei eine Variable auf 1 (Fall gehört zur Stichprobe) oder 0 (Fall gehört nicht zur Stichprobe) gesetzt wird, sondern daß eine Output-Datei im OSIRIS-Format generiert wird, die nur noch die zur Stichprobe gehörigen Fälle enthält; der große Datensatz wird bei dieser Strategie also nur gelesen. Der zugehörige Recode hat die folgende Gestalt:

Recode I

```
$RECODE
$COMMENT    ES WIRD DAS 'ZIEHUNGSGEDAECHTNIS' BEREITGESTELLT.
$COMMENT    R1 ZAEHLT DIE ABGEARBEITETEN FAELLE.
$COMMENT    R2 ZAEHLT DIE IN DIE STICHPROBE GELANGENDEN FAELLE.
     CARRY(R1,R2)
```

```
$COMMENT    ES FOLGT DER ENTSCHEIDUNGSALGORITHMUS.
      R3=RAND(1,77777777)
$COMMENT    DIE ERZEUGTE ZUFALLSZAHL WIRD IN DAS INTERVALL
$COMMENT    (0,1) TRANSFORMIERT.
      R4=R3/77777777
$COMMENT    ES FOLGEN DIE ZWISCHENRECHNUNGEN FUER DIE ENT-
$COMMENT    SCHEIDUNGSABFRAGE.
      R5=(100000-R1)*R4
      R6=6500-R2
$COMMENT    ERHOEHUNG DES FALLZAEHLERS UM EINS
      R1=R1+1
$COMMENT    ALLEIN DIE IN DIE STICHPROBE GELANGENDEN FAELLE
$COMMENT    ERSCHEINEN IN DER OUTPUT-DATEI.
      IF R5 GE R6 THEN REJECT
     *ELSE R2=R2+1
```

Die nächste Aufgabe besteht darin, aus den restlichen 93.500 Fällen mit Hilfe derselben Prozedur, also wiederum in uneingeschränkter Zufallsauswahl, eine Stichprobe der Größe 3.200 nachzuziehen. Insgesamt wird also in uneingeschränkter Zufallsauswahl eine Stichprobe der Gesamtgröße 9.700 angestrebt.

Die Lösung des Problems macht sich zunutze, daß die Folge der Zufallszahlen, die der ersten Stichprobe zugrunde lag, vermöge der bekannten Startzahl 1 vollständig rekonstruierbar ist: Man läßt dementsprechend beim Nachziehen den Algorithmus für die erste Ziehung mitlaufen, trifft dabei jedoch nur für solche Fälle eine Entscheidung in Bezug auf die zweite Stichprobe, die für die erste Stichprobe nicht ausgewählt wurden. Es wird wiederum die Ausgangsdatei (100.000 Fälle!) lediglich gelesen und eine Output-Datei generiert, die genau 3.200 Fälle enthält. Der zugehörige Recode wird im Rahmen eines TRANS-Laufes eingesetzt. An die Ausgangsdatei wird vermöge TRANS eine Variable V19 hinzugefügt, die durch ihre konstante Ausprägung 2 anzeigen soll, daß es sich bei der Output-Datei um die zweite, nachgezogenen Stichprobe handelt.

Recode II

```
$RUN TRANS
$RECODE
$COMMENT    R1 BZW R21 ZAEHLEN DIE ABGEARBEITETEN FAELLE.
$COMMENT    R2 BZW R22 ZAEHLEN DIE AUSZUSONDERNDEN BZW DIE
$COMMENT    IN DIE ZWEITE STICHPROBE GELANGENDEN FAELLE.
      CARRY(R1,R2,R21,R22)
$COMMENT    ES FOLGT DER RECODE FUER DIE ERSTE STICHPROBE,
$COMMENT    WOBEI JEDER FALL, DER ZUVOR IN DIE STICHPROBE
$COMMENT    GELANGTE, JETZT ABGELEHNT WIRD.
      R3=RAND(1,77777777)
      R4=R3/777777777
      R5=(100000-R1)*R4
      R6=6500-R2
      R1=R1+1
      IF R5 GE R6 THEN GO TO SAMP
      R2=R2+1
      REJECT
$COMMENT    ES FOLGT DER ZIEHUNGSALGORITHMUS IN BEZUG AUF
$COMMENT    DIE RESTLICHEN 93500 FAELLE. DIE KOMMENTARE
$COMMENT    FINDEN SICH IN RECODE I.
 SAMP R23=RAND(3,77777777)
      R24=R23/77777777
      R25=(93500-R21)*R24
      R26=3200-R22
      R21=R21+1
      IF R25 GE R26 THEN REJECT
      R22=R22+1
$COMMENT    DIE STICHPROBENVARIABLE R19 WIRD KONSTANT AUF
$COMMENT    DEN WERT 2 GESETZT.
      R19=2
      NAME R19 'STICHPROBENVAR'
$SETUP
STICHPROBE 2
TRAN,WIDTH=2,TYPE=1*
V1-V18*
R19*
```

Eine leichte Modifizierung von Recode II wurde bei ZUMA zur Ziehung der Zufallsstichprobe für das Projekt "Mannheimer Mietspiegel" angewendet. Trotz der großen Fallzahlen wurden akzeptable Laufzeiten von unter 30 Minuten erzielt (SIEMENS 4004, BS 2000). Die Technik von Recode I kann selbstverständlich auch zur reinen "repräsentativen" Datenreduktion verwendet werden.

### 3.2 Geschichtete Zufallsstichproben

Gegeben ist eine Datei im OSIRIS-Format mit einer bestimmten Anzahl von Fällen und Variablen. Insbesondere ist eine polytome Variable gegeben, deren 14 Ausprägungen die Gesamtheit der Fälle in 14 Schichten einteilen. Es wird vorausgesetzt, daß die Größen der Schichten bekannt sind.

Die Aufgabe besteht darin, aus 13 Schichten in uneingeschränkter Zufallsauswahl Stichproben vorgegebener Größe zu ziehen; dabei soll die Ziehung in den einzelnen Schichten unabhängig voneinander erfolgen. Des weiteren wird eine Output-Datei im OSIRIS-Format verlangt, die sowohl sämtliche Variablen der Input-Datei als auch eine Stichprobenvariable enthält. Diese soll die Werte 0 bzw. i haben, je nachdem ob der Fall zu keiner der Stichproben gehört bzw. zur Stichprobe aus der i-ten Schicht.

Zur Lösung des Problems wird wieder eine Spielart des Algorithmus aus Recode I herangezogen. Man macht sich im übrigen zunutze, daß die Entscheidungsregel an sich für alle Schichten gleich ist und lediglich berücksichtigt werden muß, daß jedem Fall, der zu einer der 13 Schichten gehört, eine schichtspezifische "Vorgeschichte" der Ziehung zuzuordnen ist.

Zur technischen Operationalisierung dieses Grundgedankens wird zunächst durch eine IF-Abfrage festgestellt, zu welcher Schicht der eingelesene Fall gehört. Danach wird der Entscheidungsalgorithmus, dem zuvor die schichtspezifischen Daten übergeben wurden, aufgerufen. Ist entschieden, ob der Fall zur Stichprobe gehört oder nicht, wird das zur Schicht gehörige "Ziehungsgedächtnis" entsprechend aktualisiert und danach der nächste Fall eingelesen. Der Recode III wird wie im vorigen Abschnitt 3.1 im Rahmen eines TRANS-Laufes dargestellt.

Recode III

```
$RUN TRANS
$RECODE
$COMMENT    R6009 IST DIE STICHPROBENVARIABLE. DIESE
$COMMENT    HAT DEN WERT NULL, WENN DER FALL ZU KEINER
$COMMENT    DER 13 STICHPROBEN GEHOERT. SONST HAT R6009
$COMMENT    ALS WERT DIE NUMMER DER SCHICHT.
     R6009=0
$COMMENT    V400 IST DIE SCHICHTVARIABLE. IHRE AUSPRAEGUNG
$COMMENT    GIBT AN, ZU WELCHER SCHICHT EIN FALL GEHOERT.
$COMMENT    DIE SCHICHT MIT V400=0 WIRD BEIM ZIEHEN DER
$COMMENT    STICHPROBEN IGNORIERT.
     IF V400 EQ 0 THEN GO TO END
$COMMENT    ES WIRD DAS 'ZIEHUNGSGEDAECHTNIS' BEREITGESTELLT.
     CARRY(R5101-R5113,R5201-R5213)
$COMMENT    ES WIRD DIE SCHICHTNUMMER FESTGESTELLT.
$COMMENT    R6002 GIBT DIE SCHICHTGROESSE UND R6003 GIBT
$COMMENT    DIE STICHPROBENGROESSE PRO SCHICHT AN.
     IF V400 EQ 13 THEN R6002= 36 AND R6003=36 AND GO TO COMP
     IF V400 EQ 12 THEN R6002=113 AND R6003=36 AND GO TO COMP
     IF V400 EQ 11 THEN R6002=290 AND R6003=92 AND GO TO COMP
     IF V400 EQ 10 THEN R6002=313 AND R6003=99 AND GO TO COMP
     IF V400 EQ  9 THEN R6002= 73 AND R6003=23 AND GO TO COMP
     IF V400 EQ  8 THEN R6002= 41 AND R6003=13 AND GO TO COMP
     IF V400 EQ  7 THEN R6002= 57 AND R6003=18 AND GO TO COMP
     IF V400 EQ  6 THEN R6002=291 AND R6003=93 AND GO TO COMP
     IF V400 EQ  5 THEN R6002=167 AND R6003=53 AND GO TO COMP
     IF V400 EQ  4 THEN R6002= 89 AND R6003=28 AND GO TO COMP
     IF V400 EQ  3 THEN R6002= 57 AND R6003=18 AND GO TO COMP
     IF V400 EQ  2 THEN R6002= 97 AND R6003=31 AND GO TO COMP
     IF V400 EQ  1 THEN R6002=160 AND R6003=51
$COMMENT    ES FOLGT DER ENTSCHEIDUNGSALGORITHMUS.
 COMP R6000=V400
     R6001=RAND(1,77777777)
$COMMENT    DIE ANZAHLEN DER BISHER IN DER SCHICHT DES
$COMMENT    FALLES BEARBEITETEN FAELLE BZW. GEZOGENEN
```

```
$COMMENT    FAELLE WERDEN AN DEN ALGORITHMUS UEBERGEBEN.
      R6004=SELECT(FROM=R5101-R5113,BY=R6000)
      R6005=SELECT(FROM=R5201-R5213,BY=R6000)
$COMMENT    DIE ERZEUGTE ZUFALLSZAHL WIRD IN DAS INTERVALL
$COMMENT    (0,1) TRANSFORMIERT.
      R6006=R6001/77777777
$COMMENT    ES FOLGEN DIE ZWISCHENRECHNUNGEN FUER DIE ENT-
$COMMENT    SCHEIDUNGSABFRAGE.
      R6007=(R6002-R6004)*R6006
      R6008=R6003-R6005
$COMMENT    ERHOEHUNG DES FALLZAEHLERS IN DER SCHICHT UM EINS.
      R6004=R6004+1
$COMMENT    DIE STICHPROBENVARIABLE R6009 IST ENTWEDER GLEICH
$COMMENT    NULL (FALL NICHT IN DER STICHPROBE) ODER GLEICH DER
$COMMENT    NUMMER DER SCHICHT DES FALLES (FALL IN DER STICHPR.).
      IF R6007 GE R6008 THEN R6009=0
     *ELSE R6009=R6000 AND R6005=R6005+1
      NAME R6009 'STICHPROBENVAR'
$COMMENT    DIE ANZAHLEN DER BISHER IN DER SCHICHT DES FALLES
$COMMENT    BEARBEITETEN FAELLE BZW. GEZOGENEN FAELLE WERDEN
$COMMENT    AN DAS ZIEHUNGSGEDAECHTNIS UEBERGEBEN.
      SELECT(FROM=R5101-R5113,BY=R6000)=R6004
      SELECT(FROM=R5201-R5213,BY=R6000)=R6005
 END  CONTINUE
$SETUP
SCHICHTWEISES ZIEHEN
TRAN,WIDTH=2,VSTART=1*
V1-V400*
R6009*
```

Recode III ist insbesondere dann nützlich, wenn für eine Wiederholungsbefragung nur ein Teil der Adressen wiederverwendet werden soll und zudem eine Schichtung von Interesse ist. Bei ZUMA wurde dieses Verfahren mehrfach angewendet.

3.3 Replikationen

Gegeben ist eine Datei im OSIRIS-Format mit einer bestimmten Anzahl von Fällen und Variablen. Die Aufgabe besteht darin, aus dieser Datei in uneingeschränkter Zufallsauswahl 15 Stichproben mit derselben vorgegebenen Größe unabhängig voneinander zu ziehen. Verlangt wird eine Output-Datei im OSIRIS-Format, die sowohl sämtliche Variablen der Input-Datei als auch 15 Stichprobenvariablen enthält. Gehört ein Fall zur i-ten Stichprobe, soll die i-te Stichprobenvariable den Wert i annehmen, andernfalls soll der Wert gleich 0 sein.

Aufgabenstellungen dieser Art ergeben sich bei der Untersuchung der stichprobenbedingten Varianz von Parameterschätzungen. Die Grundidee ist dabei, durch das wiederholte Ziehen von Teilstichproben, durch Replikationen also, und durch anschließenden Vergleich der (Teilstichproben-)Parameterschätzungen schließlich Varianzschätzungen zu erhalten (vgl. KISH & FRAENKEL, 1970).

Zur Lösung des Problems wird eine Spielart des Algorithmus aus Recode I herangezogen. Damit wird pro Fall 15-mal entschieden, ob dieser zur Stichprobe gehören soll oder nicht. Die Entscheidungen sind natürlich unabhängig voneinander. Programmtechnisch wird stark von der Möglichkeit der indirekten Adressierung von Variablen Gebrauch gemacht.

Recode IV

```
$RUN TRANS
$RECODE
$COMMENT    DIE KONSTANTEN R400, R500 UND R600 STEHEN
$COMMENT    FUER DIE ANZAHL DER TEILSTICHPROBEN, FUER
$COMMENT    DIE GROESSE DER DATEI UND DIE GROESSE DER
$COMMENT    TEILSTICHPROBEN.
      R400=15
      R500=2000
      R600=1000
$COMMENT    R2 DIENT ALS 'SCHLEIFENZAEHLER' UND WIRD
$COMMENT    AM BEGINN DER ABARBEITUNG EINES FALLES AUF
```

```
$COMMENT    EINS GESETZT.
      R2=1
$COMMENT    ES WIRD DAS 'ZIEHUNGSGEDAECHTNIS' BEREITGESTELLT.
      CARRY(R100,R111-R125)
$COMMENT    LOOP IST DER BEGINN EINER SCHLEIFE, DIE 15-MAL
$COMMENT    DURCHLAUFEN WIRD. BEI JEDEM DURCHLAUF WIRD DER
$COMMENT    ZIEHUNGSALGORITHMUS ANGESPROCHEN.
 LOOP R3=RAND(5,77777777)
$COMMENT    DIE ERZEUGTE ZUFALLSZAHL WIRD IN DAS INTERVALL
$COMMENT    (0,1) TRANSFORMIERT.
      R4=R3/77777777
$COMMENT    DIE ANZAHL DER BISHER GEZOEGENEN FAELLE WIRD AN
$COMMENT    DEN ALGORITHMUS UEBERGEBEN.
      R200=SELECT(FROM=R111-R125,BY=R2)
$COMMENT    ES FOLGEN DIE ZWISCHENRECHNUNGEN FUER DIE ENT-
$COMMENT    SCHEIDUNGSABFRAGE.
      R5=(R500-R100)*R4
      R6=R600-R200
$COMMENT    DIE STICHPROBENVARIABLE IST ENTWEDER GLEICH NULL
$COMMENT    (FALL NICHT IN DER STICHPROBE) ODER GLEICH DER
$COMMENT    NUMMER DER STICHPROBE.
      IF R5 GE R6 THEN GO TO SL
      SELECT(FROM=R301-R315,BY=R2)=R2
      R200=R200+1
      GO TO SLL
  SL  SELECT(FROM=R301-R315,BY=R2)=0
$COMMENT    DIE ANZAHL DER BISHER GEZOGENEN FAELLE
$COMMENT    WIRD AN DAS 'ZIEHUNGSGEDAECHTNIS' UEBERGEBEN.
 SLL  SELECT(FROM=R111-R125,BY=R2)=R200
$COMMENT    ERHOEHUNG DES SCHLEIFENZAEHLERS UM 1 UND ABFRAGE
$COMMENT    AUF ENDE DER SCHLEIFE.
      R2=R2+1
      IF R2 LE R400 THEN GO TO LOOP
$COMMENT    FUER ALLE 15 STICHPROBEN WIRD DER FALLZAEHLER
$COMMENT    R100 UM EINS ERHOEHT.
      R100=R100+1
$SETUP
```

```
15 TEILSTICHPROBEN
TRANS,WIDTH=2*
V1=V300*
R301-R315*
```

Der Einsatz des Recode IV beschränkte sich bei ZUMA bisher auf den Bereich der Grundlagenforschung zu Gewichtungen und Gewichtungseffekten. Es wurde untersucht, welche Variation die Schätzungen von Randverteilungen von Teilstichprobe zu Teilstichprobe aufweisen, und zwar bei Datensätzen mit rund 2.000 Fällen.

Es sei darauf hingewiesen, daß der Recode IV darauf zugeschnitten ist, daß die zugrundeliegende (Haupt-)Stichprobe in uneingeschränkter Zufallsauswahl gezogen wurde. Trifft dies nicht zu, wurden also z.B. ungleiche Auswahlwahrscheinlichkeiten verwendet, wird die Umsetzung in einen OSIRIS-Recode eventuell beträchtlich komplizierter.

### 3.4 Systematisches Ziehen

Gegeben ist eine Datei im OSIRIS-Format mit einer bestimmten Anzahl von Fällen und Variablen. Insbesondere ist eine Variable gegeben, deren Ausprägungen jeweils als "Bedeutungsgewicht" des Falles interpretiert werden.

Die Aufgabe besteht darin, aus dieser Datei systematisch mit einem Zufallsstart eine Stichprobe so zu ziehen, daß die einzelnen Fälle mit einer Wahrscheinlichkeit in die Stichprobe gelangen, die ihrem Bedeutungsgewicht proportional ist. Es wird eine Output-Datei im OSIRIS-Format verlangt, die sowohl sämtliche Variablen der Input-Datei als auch eine Stichprobenvariable enthält. Diese soll angeben, in welcher Form ein Fall in die Stichprobe gelangt ist.+

Die Lösung des Problems weicht insofern von derjenigen in den Abschnitten 3.1-3.3 ab, als jetzt zwar immer noch Fall für Fall eine Entscheidung getroffen wird, dieses jedoch auf einem zufälligen Beginn beruht. Es wird

+ Fälle mit sehr großem Bedeutungsgewicht können bei kleinen Schrittweiten mehr als einmal in die Stichprobe gelangen (vgl. HANSEN, HURWITZ, MADOW, 1953, S. 341f.)

also nicht pro Fall eine neue Zufallszahl erzeugt, sondern lediglich zu Anfang ein zufälliger "Start" bestimmt.

Recode V

```
$RUN TRANS
$RECODE
$COMMENT    R100 IST DIE STICHPROBENVARIABLE.
      R100=0
$COMMENT    ES WIRD DAS 'ZIEHUNGSGEDAECHTNIS' BEREIT-
$COMMENT    GESTELLT.
$COMMENT    R20  DIENT ZUR KUMULIERUNG DER BEDEUTUNGS-
$COMMENT    GEWICHTE.
$COMMENT    R200 ZAEHLT DIE SCHRITTE.
$COMMENT    R202 IST DIE STARTZAHL.
$COMMENT    R300 ZAEHLT DIE ABGEARBEITETEN FAELLE.
      CARRY(R20,R200,R202,R300)
$COMMENT    DIE ERZEUGUNG DER STARTZAHL WIRD VOM ZWEITEN
$COMMENT    FALL AN UEBERSPRUNGEN.
      IF R300 NE 0 THEN GO TO FLAG
$COMMENT    ERZEUGUNG DER STARTZAHL / DIE SCHRITTWEITE IST ELF.
      R201=RAND(0,77777777)
      R202=R201*11/77777777
$COMMENT    KUMULATION DER BEDEUTUNGSGEWICHTE, DIE VON
$COMMENT    VARIABLE V1 BESCHRIEBEN WERDEN.
 FLAG R10=R20
      R20=R20+V1
$COMMENT    DER ENTSCHEIDUNGSALGORITHMUS WEIST R100 DEN
$COMMENT    WERT DER VIELFACHHEIT DES AUFTRETENS DES
$COMMENT    FALLES IN DER STICHPROBE ZU.
 STEP R203=R202+(R200*11)
      IF R10 GE R203 OR R20 LT R203 THEN GO TO END
      R100=R100+1
      R200=R200+1
      GO TO STEP
$COMMENT    ERHOEHUNG DES FALLZAEHLERS UM EINS.
  END R300=R300+1
```

```
$SETUP
BEDEUTUNGSGEWICHTE
TRAN,WIDTH=2*
V1-V99*
R100*
```

Für den Spezialfall, daß alle Bedeutungsgewichte identisch sind, stellt der Recode V offensichtlich das übliche systematische Ziehen dar.

Bei ZUMA wurde der Recode V bisher ausschließlich im Methodenprojekt "Gewichtung und Gewichtungseffekte" eingesetzt.

### 3.5 Stichproben aus Klumpen

Gegeben ist eine Datei im OSIRIS-Format mit einer bestimmten Anzahl von Fällen und Variablen. Diese Datei ist nach Gruppen=Klumpen angeordnet, d.h. die Fälle innerhalb eines Klumpens sind unmittelbar benachbart gespeichert. Im folgenden wird zunächst angenommen, daß die Klumpengröße nicht bekannt ist. Danach wird der Fall bekannter Gruppengröße diskutiert.

Die Aufgabe besteht darin, in uneingeschränkter Zufallsauswahl aus jedem der Klumpen eine Stichprobe der Größe 1 zu ziehen. Es wird eine Output-Datei im OSIRIS-Format verlangt, die sämtliche Variablen der Input-Datei enthält, und die reduziert ist auf die in die Stichprobe gelangten Fälle.

Zur Lösung des Problems bei unbekannter Klumpengröße wird das in den ZUMA-NACHRICHTEN 3, S.35, beschriebene "Mischungsverfahren" herangezogen, und zwar klumpenweise. Der Grundgedanke beim Recode VI ist dabei, die Fälle eines Klumpens in eine zufällige Reihenfolge zu bringen, um dann den an der ersten Stelle stehenden Fall als Stichprobe der Größe 1 zu erhalten. Dieses Verfahren ist allerdings im Gegensatz zu denjenigen in den Abschnitten 3.2-3.4 nicht mit nur einem Durchlauf durch die Input-Datei zu bewältigen. Es muß nämlich zunächst in einem ersten Arbeitsgang jeder Klumpen für sich nach der den einzelnen Fällen zugeordneten Zufallszahl sortiert werden; danach wird eine Output-Datei erstellt, die nur noch den ersten Fall eines jeden Klumpens enthält.

Das notwendige zweimalige Erstellen einer Datei kann als "Preis" dafür bezeichnet werden, daß die Klumpengrößen im einzelnen nicht im Datensatz enthalten sind.

Bei bekannten Klumpengrößen ist demgegenüber mit Hilfe des den Abschnitten 3.1-3.4 zugrundeliegenden Algorithmus die Ziehung der Stichprobe in einem Durchgang möglich. Dies wird in dem Recode VII demonstriert. Dabei besteht der Grundgedanke zur Lösung des Problems darin, beim Duchlaufen des Datensatzes durch eine IF-Abfrage jeweils den Anfang eines Klumpens festzustellen, um dann die Parameter des Ziehungsalgorithmus mit den klumpenspezifischen Werten, also insbesondere der Klumpengröße, zu besetzen.

Anders als in den Abschnitten 3.1-3.4 sind beim folgenden Recode VI für unbekannte Klumpengröße die Recode-Anteile ganz außerordentlich einfach: die eigentliche Lösung des Problems liegt hier mehr in der sinnvollen Hintereinanderschaltung verschiedener OSIRIS-Programme (TRANS-SORT-TRANS).

Recode VI

```
Zwei Dateikommandos für Eingabedateien   erster
Zwei Dateikommandos für Ausgabedateien   TRANS-Lauf

Zwei Dateikommandos für den Sortierlauf

Zwei Dateikommandos für Eingabedateien   zweiter
Zwei Dateikommandos für Ausgabedateien   TRANS-Lauf

$RUN TRANS
$RECODE
$COMMENT    JEDEM FALL IN DER EINGABEDATEI WIRD EINE
$COMMENT    ACHTSTELLIGE ZUFALLSZAHL ZUGEORDNET.
     R100=RAND(1,77777777)
$SETUP
ZUFALLSZAHLEN
TRAN,WIDTH=8*
V1-V99*
R100*
```

In dem anschliessenden Sortierlauf wird die Output-Datei des ersten TRANS-Laufes nach der Klumpen-ID und dann nach V100 klumpenweise aufsteigend sortiert. Da dies maschinenabhängig ist, wird hier auf eine detaillierte Darstellung verzichtet.

```
$RUN TABLES
$RECODE
$COMMENT    ES WIRD DER IN DEM 'DURCHGEMISCHTEN' KLUMPEN
$COMMENT    AN DER ERSTEN STELLE STEHENDE FALL IN DIE OUT-
$COMMENT    PUT-DATEI UEBERNOMMEN. V1 IST DIE KLUMPEN-ID.
     CARRY(R10)
     IF R10 EQ V1 THEN REJECT
     R10=V1
$SETUP
KLUMPENSTICHPROBE
TRAN,INFI=INX,VSTART=1*
V2-V100*
V1*
```

Recode VI kann offenbar leicht bei Klumpen, die sämtlich größer als 2 sind, etwa auf Klumpenstichproben der Größe 2 verallgemeinert werden. Bei von Klumpen zu Klumpen variierender Stichprobengröße sind jedoch umfangreichere Modifikationen erforderlich. Bei ZUMA wurde Recode VI bisher eingesetzt, um bei einer Repräsentativstudie (Nationaler Sozialer Survey) den sog. Schwedenschlüssel zu simulieren: Es wurde untersucht, inwieweit sich die prozentuale Verteilung bestimmter Alters- und Geschlechtskohorten ändert, wenn pro Haushalt der Stichprobe erneut zufällig genau eine Person (maschinell) ausgewählt wird - das erste Mal tat dies im Feld der Interviewer eben mit Hilfe des Schwedenschlüssels.

Der folgende Recode VII setzt voraus, daß eine Gruppengrößenvariable existiert, die pro Fall als Ausprägung die Gruppengröße des Falles besitzt, und daß die Datei nach den Gruppen sortiert ist - vgl. die Voraussetzungen am Anfang dieses Abschnitts.

Recode VII

```
$RECODE
$COMMENT    ES WIRD ZUERST ABGEPRUEFT, OB EIN NEUER KLUMPEN
$COMMENT    BEGINNT. IST DIES DER FALL, WIRD DAS 'ZIEHUNGSGE-
$COMMENT    DAECHTNIS' (R1,R2) AUF (0,0) GESETZT. V1 IST DIE
$COMMENT    KLUMPENIDENTIFIKATION.
     CARRY(R10)
     IF R10 NE V1 THEN R1=0 AND R2=0
     R10=V1
$COMMENT    ES WIRD DAS 'ZIEHUNGSGEDAECHTNIS' BEREITGESTELLT.
     CARRY(R1,R2)
     R3=RAND(1,77777777)
$COMMENT    DIE ERZEUGTE ZUFALLSZAHL WIRD IN DAS INTERVALL
$COMMENT    (0,1) TRANSFORMIERT.
     R4=R3/77777777
$COMMENT ES FOLGEN DIE ZWISCHENRECHNUNGEN FUER DIE ENT-
$COMMENT SCHEIDUNGSABFRAGE. V2 IST DIE KLUMPENGROESSE.
     R5=(V2-R1)*R4
     R6=1-R2
$COMMENT    ERHOEHUNG DES FALLZAEHLERS UM EINS.
     R1=R1+1
$COMMENT    ALLEIN DIE IN DIE STICHPROBE GELANGENDEN FAELLE
$COMMENT    ERSCHEINEN IN DER OUTPUT-DATEI.
     IF R5 GE R6 THEN REJECT
    *ELSE R2=R2+1
```

Wie beim Recode VI gilt auch beim Recode VII, daß bei hinreichend großer minimaler Klumpengröße auch größere Stichproben pro Klumpen gezogen werden können. Jedoch muß dazu die Stichprobengröße konstant sein. Werden klumpenabhängige Stichprobengrößen gefordert, so müssen diese für Recode VII in einer zusätzlichen Variablen VXYZ dem Datensatz hinzugefügt werden. Das Statement 'R6=1-R2' wird dann zu 'R6=VXYZ-R2'. Recode VII wurde bisher bei ZUMA nicht im größeren Maßstabe angewendet.

Für die Behandlung stichprobentheoretischer und stichprobenpraktischer Problemstellungen ist bei ZUMA Hans-Peter Kirschner verantwortlich, der auch diesen Artikel verfaßt hat.

Literatur:

HANSEN, M.H., HURWITZ, W.N. & MADOW, W.G. Sample survey methods and theory. Methods and Applications. New York: John Wiley, 1953.

KENNEDY, W.J.Jr. & GENTLE, J.E. Statistical computing. New York: Marcel Dekker, 1980.

KIRSCHNER, H.P. Einige Bemerkungen zur maschinellen Ziehung von Zufallsstichproben. Zumanachrichten 3, 1978, 28-41.

KISH, L. & FRANKEL, M.R. Balanced repeated replications for standard errors. J. American Statist. Ass., 1970, 65, 1071-1094.

KNUTH, D.E. The art of computer programming. Seminumerical algorithms. Reading: Addison-Wesley Publishing Company, 1969.