Aufbereitung von Verlaufsdaten mit zeitveränderlichen Kovariaten mit SPSS

Brüderl, Josef; Ludwig-Mayerhofer, Wolfgang

Veröffentlichungsversion / Published Version
Zeitschriftenartikel / journal article

Aufbereitung von Verlaufsdaten mit zeitveränderlichen Kovariaten mit SPSS

von Josef Brüderl [1] und Wolfgang Ludwig-Mayerhofer [2]

Zusammenfassung

Die Analyse von Verlaufsdaten (Ereignisanalyse, Survivalanalyse) ist eines der am häufigsten angewendeten Verfahren zur Auswertung von Längsschnittdaten. Einer ihrer zentralen Vorzüge, die Berücksichtigung von Einflußgrößen, die sich im Zeitverlauf ändern, kann besonders effektiv in der Datenanalyse umgesetzt werden, wenn die Verlaufsdaten zum jeweiligen Zeitpunkt in Daten vor und nach der Änderung zerlegt werden. Der Beitrag erläutert, wie dieses »Episodensplitting« ohne große Mühe mit SPSS/PC+ (oder ähnlichen Programmen) und SPSS für Windows realisiert werden kann. Ein Beispiel zeigt den Ertrag einer Analyse mit zeitveränderlichen Kovariaten.

Abstract

Methods of event history (or survival, or failure time) analysis are widely used in longitudinal research. One of their most useful properties is the possibility to include time-varying covariates in the analysis. This can be achieved very efficiently by splitting episodes at the time a change in the relevant covariates occurs. This article shows how episode-splitting can be accomplished by means of SPSS/PC+ (or similar programs) and SPSS for Windows. An example demonstrates the benefit of taking time-varying covariates into consideration.

1. Zeitabhängige Kovariaten in der Verlaufsdatenanalyse

Die Analyse von Verlaufsdaten - auch als »Ereignisanalyse«, »Event History Analysis« oder »Survivalanalyse« bezeichnet - hat sich in den letzten zehn Jahren einen wichtigen Platz im sozialwissenschaftlichen Methodeninstrumentarium erobert (vgl. ***Andreß*** 1992; ***Blossfeld, Hamerle und Mayer*** 1986; ***Diekmann*** 1988; ***Diekmann und Mitter*** 1984; ***Ludwig-Mayerhofer*** 1994). Ein wichtiger Motor dieser Entwicklung war das zunehmende Interesse an Längsschnittuntersuchungen, Lebensverläufen und Veränderungen im Zeitver-

1 Dr. *Josef Brüderl* ist wissenschaftlicher Mitarbeiter am Institut für Soziologie der Ludwig-Maximilians-Universität München, Konradstr. 6, D-80801 München.

2 Dr. *Wolfgang Ludwig-Mayerhofer* ist derzeit Habilitationsstipendiat der Deutschen Forschungsgemeinschaft. Anschrift: c/o Münchner Projektgruppe für Sozialforschung (MPS) e.V., Dachauer Str. 189/111, D-80637 München.

lauf, also eine »Dynamisierung« des Blicks auf soziale Phänomene. Die Verfahren, auf die wir uns beziehen, sind dann angebracht, wenn *diskrete Zustandsänderungen* untersucht werden, etwa der Wechsel von Beschäftigung in Arbeitslosigkeit oder umgekehrt; die *Verweildauern* im Ausgangszustand sollten idealerweise auf einer *kontinuierlichen* Zeitskala gemessen sein, jedoch stehen auch Modelle für die Analyse *diskret* gemessener Dauern zur Verfügung ***(Hamerle und Tutz*** 1989; ein Anwendungsbeispiel bei ***Tutz und Georg*** 1991). Die Verfahren sind insbesondere auch dann geeignet, wenn aus Gründen des Studiendesigns (wegen Panelmortalität und dergleichen) nicht alle Verweildauern vollständig beobachtet werden konnten (sog. rechtszensierte Daten). Neben nicht-parametrischen Verfahren für die explorative Datenanalyse und die Auswertung einfacher Experimentaldaten sind für Sozialwissenschaftler insbesondere *semi-parametrische* und *parametrische* Verfahren von Bedeutung, die eine simultane multivariate Analyse einer Vielzahl von Einflüssen gestatten.

Einer der zentralen Vorzüge dieser Verfahren, wenn nicht der wichtigste überhaupt, ist die Möglichkeit, erklärende Variablen (Kovariaten) zu berücksichtigen, *die sich selbst im Zeitverlauf ändern.* Erst damit ist eine wirklich dynamische Analyse sozialer Prozesse möglich. Beispiele hierfür wären der Einfluß von Heirat und Geburt von Kindern auf den Erwerbsverlauf von Frauen ***(Tölke*** 1989) oder umgekehrt der Einfluß des Zeitpunkts, zu dem eine Ausbildung abgeschlossen wurde, auf den Zeitpunkt der Heirat und der Geburt von Kindern ***(Blossfeld und Huinink*** 1991; ***Brüderl und Klein*** 1993). Ebenso können veränderliche gesellschaftliche Meso- oder Makrobedingungen individuelle Verläufe beeinflussen; zu denken wäre etwa an den Einfluß betrieblicher Strukturmerkmale auf die Aufstiegschancen von Arbeitern ***(Brüderl, Preisendörfer und Ziegler*** 1993), oder an den Einfluß regionaler Arbeitsmarktchancen auf individuelle Beschäftigungs- oder Arbeitslosigkeitsverläufe ***(Hujer und Schneider*** 1992).

Zeitveränderliche Kovariaten können in der Verlaufsdatenanalyse in zweierlei Art und Weise verarbeitet werden. Ein spezielles Modell, das semi-parametrische Proportional-Hazards-Modell von ***Cox*** (1972), erlaubt es, solche Kovariaten direkt mit der »Prozeßzeit« zu verknüpfen. Weitaus universeller, nämlich auch im Rahmen parametrischer Modelle für stetige oder diskrete Verweildauern einsetzbar, ist die Möglichkeit des *»Episodensplittings«.* Dabei werden die einzelnen Episoden des untersuchten Prozesses immer dann in Teilepisoden zerlegt, wenn sich der Wert einer zeitveränderlichen Kovariaten ändert (zu einer genaueren Darstellung siehe unten). Diese Möglichkeit ist auch bei den semi-parametrischen Modellen nach Cox empfehlenswert - und zwar aus Gründen der Rechenzeit -, wenn nicht nur sehr wenige zeitveränderliche Kovariaten analysiert werden sollen.

Allerdings ist mit dem Episodensplitting in der Regel ein erhöhter Aufwand hinsichtlich des Datenmanagements verbunden, was möglicherweise manche Benutzer davon abhält, entsprechende Informationen vollständig auszunützen. Diese Haltung wird wahrscheinlich

noch verstärkt durch die Ausführungen in dem wohl bekanntesten deutschsprachigen Lehrbuch zur Ereignisanalyse (***Blossfeld, Hamerle und Mayer*** 1986, S. 196), wonach ein Episodensplitting mit Standard-Statistikpaketen wie SPSS nicht möglich sei. In diesem Buch werden die Leser auf selbstgeschriebene Programme in FORTRAN verwiesen. Diese sind auch leicht auf andere höhere Programmiersprachen übertragbar, aber man kann nicht davon ausgehen, daß die Fähigkeit zum sicheren Umgang mit solchen Programmiersprachen bei allen Sozialwissenschaftlern vorhanden oder ohne großen Aufwand herstellbar ist. Im übrigen ist das bei ***Blossfeld, Hamerle und Mayer*** (ebd.) dargestellte Beispiel mißverständlich, so daß in der deutschsprachigen Literatur keine adäquate Darstellung des Episodensplittings verfügbar ist.

Auch viele spezielle Softwarepakete zur Analyse von Verlaufsdaten enthalten keine ausreichenden Möglichkeiten zur Datenmanipulation, z.B. die Programme RATE ***(Tuma*** 1980), PARAT ***(Schneider*** 1991)[3] oder LIMDEP ***(Greene*** 1992). Das wohl flexibelste und umfangreichste Programm zur Verlaufsdatenanalyse, TDA ***(Rohwer*** 1993), verfügt zwar über eine Funktion zum Episodensplitting, jedoch dürften unerfahrene Benutzer mit dieser Option, die zudem nicht gut dokumentiert ist, eher zurückhaltenden Umgang pflegen.

Nach unseren Erfahrungen werden vielfach zur Datenhaltung und -aufbereitung Standardpakete, insbesondere SPSS, eingesetzt und spezielle Programme nur soweit erlernt, wie es für diejenigen Zwecke unbedingt erforderlich ist, die nur mit diesen Programmen erreicht werden können. Dieses Vorgehen ist auch durchaus sinnvoll, da gerade bei komplexeren Datentransformationen eine sichere Beherrschung des jeweiligen Programms notwendig ist. Daher wollen wir in diesem Beitrag zeigen, daß das Episodensplitting mit SPSS ohne großen Aufwand durchführbar ist. Unsere Beispiele beziehen sich sowohl auf SPSS/PC$^+$ als auch auf SPSS für Windows. Während im letzteren Fall eine spezielle Funktion verwendet wird, die das wiederholte Einlesen ein und derselben Datenzeile erlaubt ,[4] dürften die für SPSS/PC$^+$ dargestellten Verfahrensweisen mit geringen Modifikationen in praktisch allen üblichen Statistikpaketen umsetzbar sein und somit auch Nicht-SPSS-Benutzem Anregungen geben können. Im übrigen hoffen wir, daß unsere Beispiele auch unabhängig vom konkreten Problem lehrreich sind und die vielfach unterschätzten (wenn auch sicher nicht optimalen) Möglichkeiten von Standardstatistikpaketen für das Datenmanagement verdeutlichen können (siehe auch die Arbeit von ***Wolf*** 1993 zur Aufbereitung von Netzwerkdaten mit SPSS).

---

3 Da PARAT auf der Programmiersprache GAUSS basiert, kann diese natürlich zum Episodensplitting herangezogen werden, wozu Schneider (1991) auch einige Beispiele gibt. GAUSS ist jedoch - leider - unter Soziologen nicht sehr verbreitet.

4 Diese Funktion ist nicht in SPSS/PC+, aber neben SPSS für Windows auch in SPSS-X für Mainframes verfügbar.

2. Grundlagen des Episodensplittings

In diesem Abschnitt gehen wir kurz auf die grundlegende Datenstruktur der Verweildauer-analyse - vor allem bei Berücksichtigung zeitveränderlicher Kovariaten - ein. Für eine ausführliche inhaltliche Diskussion verschiedener Arten solcher Kovariaten und der dabei gegebenenfalls auftretenden Probleme sei auf die einschlägige Literatur verwiesen ***(Kalbfleisch/Prentice*** 1980, Kap. 5; ***Petersen*** 1986 sowie insbesondere die anwendungsbezogene Diskussion bei ***Yamaguchi*** 1991, S. 26 ff, 134 ff und 163 ff).

Grundsätzlich werden für die Verweildaueranalyse zwei Variablen benötigt, eine *Dauervariable* und eine *Zustandsvariable:* beide zusammen charakterisieren eine *Episode* (engl. *spell)* des untersuchten Prozesses. Die *Dauervariable* gibt an, wie lange sich eine Untersuchungseinheit in dem untersuchten Zustand befindet, es handelt sich also um die Dauer entweder bis zu einem Zustandswechsel oder bis zu dem Ausscheiden der Untersuchungseinheit aus der Studie (Rechts-Zensierung). Anstelle der Dauer ist es allerdings auch möglich - und beim Episodensplitting sogar unabdingbar -, die Zeitpunkte des Beginns und des Endes der jeweiligen Episode explizit anzugeben. Üblicherweise beginnt jede Episode zum Zeitpunkt Null (Analyse der Prozeßzeit). Andere zeitliche Bezugspunkte (etwa die Geburt, der Beginn des Berufslebens usw.) sind möglich, wir wollen sie im Rahmen dieser Arbeit aber ausklammern (vgl. dazu ***Blossfeld und Hamerle*** 1989). Eine Übertragung unserer Beispiele auf solche Konstellationen sollte aber ohne Schwierigkeiten möglich sein.

Die *Zustandsvariable* wird benötigt, um anzugeben, ob eine Untersuchungseinheit am Ende der Verweildauer in den Zielzustand wechselte oder nicht, im Falle mehrere Zielzustände auch, in welchen. Wie im Falle der Verweildauer können stattdessen (in Abhängigkeit vom verwendeten Programm!) auch zwei Variablen verwendet werden, von denen eine den Ausgangszustand und die andere den Zielzustand darstellt. Zensierte Fälle sind dann einfach jene, bei denen der Zielzustand mit dem Ausgangszustand identisch ist.

Wir wollen im folgenden sowohl die Dauer- als auch die Zustandsvariable explizit durch jeweils zwei Variablen charakterisieren, also durch Zeitpunkte von Beginn und Ende und die Zustände zu Beginn und Ende. Die Daten zweier Untersuchungseinheiten könnten dann - in dieser Reihenfolge - beispielsweise so aussehen:

Fall-Nr.	t_0	t_1	z_0	z_1
1	0	17	0	1
2	0	25	0	0

Die erste Untersuchungseinheit beginnt den untersuchten Prozeß zum Zeitpunkt 0 (t_0) und beendet ihn nach 17 Zeiteinheiten (t_1). Der Ausgangszustand wird durch den Wert 0 charak-

terisiert (z0), der Zielzustand durch den Wert 1 (z1); es hat also ein Zustandswechsel stattgefunden. Die zweite Untersuchungseinheit wurde 25 Zeiteinheiten beobachtet, der Zielzustand hat hier jedoch den Wert 0, d.h. die Beobachtung endete, ohne daß ein Zustandswechsel erfolgte.

Zu diesen Variablen, die den untersuchten Prozeß selbst charakterisieren, können nun metrische oder kategoriale Kovariaten kommen. Wie können nun hierbei Kovariaten berücksichtigt werden, die ihren Wert während des untersuchten Prozesses ändern? Wir wollen annehmen, daß wir beispielsweise den Einfluß des Familienstandes (Variable »X«) untersuchen wollen, der der Einfachheit halber als »nicht verheiratet« (Wert 0) und »verheiratet« (Wert 1) kodiert sein soll. Angenommen, die erste der beiden fiktiven Untersuchungspersonen heiratet nach 10 Monaten, während die zweite Untersuchungsperson von Anfang an verheiratet sei, nach 15 Monaten jedoch geschieden werden soll. Dies kann berücksichtigt werden, indem die beiden Episoden jeweils an der Stelle in zwei Teilepisoden aufgeteilt, also »gesplittet« werden, an der die Kovariate ihren Wert ändert. Selbstverständlich müssen die Informationen über den Wert der Kovariaten während dieser Teilepisoden hinzugefügt werden. Außerdem ist es erforderlich, die Variablen für den Zielzustand anzupassen, da zum Zeitpunkt der Veränderung der Kovariaten beide Untersuchungseinheiten sich immer noch im Ausgangszustand befinden. Nach dem entsprechenden Splitting sieht der Datensatz so aus:

Fall-Nr.	t_0	t_1	z_0	z_1	X
1	0	10	0	0	0
1	10	17	0	1	1
2	0	15	0	0	1
2	15	25	0	0	0

Man sieht, daß jede Episode zwei Teilepisoden erzeugt. Die erste Teilepisode endet zum Zeitpunkt 10 (tl). Ausgangs- und Zielzustand sind identisch (z0 und z1), was bedeutet, daß diese Teilepisode mit einer Zensierung endet. Die zeitabhängige Kovariate X weist den Wert Null auf, da die Person noch nicht verheiratet war. Die zweite Teilepisode beginnt zum Zeitpunkt 10. Hier ist die zeitabhängige Kovariate gleich eins, weil die Person im 10. Monat geheiratet hat. Analog wurde auch die zweite Episode aufgesplittet, nur daß hier die zeitabhängige Kovariate zuerst eins ist und dann auf null fällt (wegen der Scheidung). Es ist unbedingt zu beachten, daß die neu gebildeten Teilepisoden jeweils zu dem Zeitpunkt beginnen, an dem die vorherige Teilepisode der betreffenden Untersuchungseinheit endete. Es wird somit deutlich, daß es für die Analyse gesplitteter Episoden unabdingbar ist, nicht nur die Dauer jeder Episode, sondern explizit deren Beginn zu berücksichtigen. Deshalb muß man zur Analyse gesplitteter Episoden Programme verwenden, die die Startzeit

explizit berücksichtigen, also nicht - wie etwa SPSS für Windows oder BMDP - davon ausgehen, daß die Startzeit Null ist.

Es kann gezeigt werden, daß die Zerlegung eines Falles in mehrere Teilepisoden die Schätzung der Modellparameter und insbesondere die Standardfehler nicht beeinflußt (siehe ANHANG). Tatsächlich handelt es sich ja nur scheinbar um eine Erhöhung der Fallzahlen, denn Grundlage der Modellschätzung ist die Prozeßzeit, welche durch das Splitting nicht geändert wird.

3. Beispieldaten und Problemstellung

Im folgenden wollen wir, was soeben ganz elementar dargestellt wurde, anhand von Beispielen ausführlicher zeigen. Da in der Forschungspraxis recht unterschiedliche Datenkonstellationen auftreten können, die jeweils unterschiedliche (wenn auch natürlich verwandte) Problemlösungen erfordern, wollen wir zwei bekannte, aber in der Zeitstruktur etwas unterschiedliche Datensätze heranziehen.

Im ersten Fall liegen die Informationen über die Prozeßzeit relativ ungenau, nämlich auf der Basis von Monaten vor. Es handelt sich um Daten aus dem »Erwerbskalender« des Sozio-ökonomischen Panels (SOEP) (vgl. ***Projektgruppe »Das sozio-ökonomische Panel«*** 1990), in dem die Untersuchungspersonen Monat für Monat ihre Stellung hinsichtlich Erwerbsleben bzw. Ausbildung angeben. Im folgenden werden nur die Arbeitslosigkeitsepisoden untersucht. Der zweite Datensatz bezieht sich auf die »Bremer Längsschnitt-Stichprobe von Sozialhilfeakten« (LSA), und zwar auf die Daten der Antragskohorte 1983 (vgl. ***Voges und Zwick*** 1991). Dieser Datensatz enthält sehr genaue Angaben über Anfang und Ende des Sozialhilfebezugs.

Die Daten zur Prozeßzeit sind in den beiden Datensätzen in unterschiedlicher Art und Weise vercodet. Bevor wir in die Detailanalyse einsteigen, sei dies ganz kurz dargestellt. Wir verwenden die Daten des SOEP in Form der sog. Spell-Daten, die vom Projektträger DIW aus den Rohdaten gebildet werden. Wie gesagt, handelt es sich um monatsbezogene Daten. Diese werden auf einen standardisierten »Nullpunkt«, den Jahresbeginn 1983 bezogen. Beginnt eine Episode im Januar 1983, wird dies mit »1« verschlüsselt, beginnt sie im Februar dieses Jahres, mit »2«, usw.; die Episodenenden sind in gleicher Art und Weise kodiert.[5] Wir berücksichtigen im folgenden die Zustände »Arbeitslosigkeit« und »Vollzeitbeschäftigung«, welche mit »4« und mit »1« kodiert sind.[6] Gehen die Personen aus der

5 Die gleiche Datenstruktur, nur bezogen auf einen anderen Nullpunkt, findet sich in den Daten der Berliner Lebensverlaufsstudie, die bei *Blossfeld, Hamerle und Mayer* (1986) dargestellt werden.

6 Die »Zielzustände« sind in den Daten des DIW *nicht* enthalten; sie müssen von den Benutzern selbst definiert werden, was wiederum lehrreiche Beispiele für den Umgang mit SPSS abgeben würde.

Arbeitslosigkeit in andere Zielzustände ab, wird dies von uns als Zensierung behandelt, so daß in diesem Fall der Zielzustand ebenfalls mit »4« (re-)kodiert wurde.

Die Arbeitslosigkeitsepisoden *einiger fiktiver* Untersuchungspersonen[7] könnten beispielsweise so aussehen:

Fall-Nr.	t_0	t_1	z_0	z_1
1	2	16	4	1
2	3	8	4	1
3	17	23	4	4

Die erste Arbeitslosigkeitsepisode (z0 = 4) beginnt also im Februar 1983 (t0 = 2) und endet im April 1984 (t1 = 16) mit einem Übergang in die Vollzeitbeschäftigung (z1 = 1), usw.

Im Gegensatz dazu sind die Daten zum Sozialhilfebezug der LSA direkt in Kalenderzeit angegeben. Von drei beliebigen Fällen aus dem Datensatz werden im folgenden der Zeitpunkt des Beginns und des Endes sowie eine Variable dargestellt, die die Art der Beendigung des Sozialhilfebezugs angibt. (Da es sich bei allen Episoden um Sozialhilfeepisoden handelt, ist eine explizite Kodierung des Ausgangszustandes hier nicht erforderlich, sie kann bei Bedarf später durch Erzeugen einer geeigneten Konstanten erfolgen):

Fall-Nr.	T_0	M_0	J_0	T_1	M_1	J_1	z_1
1	3	9	83	31	12	84	3
2	21	3	83	31	12	83	2
3	6	6	83	16	6	83	3

Die erste Episode beginnt also am 3. September 1983 (To, Mo, Jo) und endet am 31. Dezember 1984 (T1, M1, Jl) mit dem Übergang in den Zustand »Sonstiges« (z1 = 3), usw.

Im folgenden wollen wir anhand dieser beiden Datensätze drei Beispiele mit Episodensplitting zeigen: *Erstens* sollen häufig *individuelle* Veränderungen berücksichtigt werden, die zumeist bei den einzelnen Untersuchungspersonen zu je unterschiedlichen Zeitpunkten auftreten, etwa Heirat, Geburt von Kindern o.ä. Dieser Fall ist ohne großen Aufwand auch mit SPSS/PC+ zu meistern. Wir demonstrieren dies in Abschnitt 4.1 anhand der SOEP-Daten. Komplizierter ist der *zweite* Fall, bei dem ein Splitting zu *regelmäßigen und für alle Untersuchungseinheiten identischen* Zeitpunkten durchgeführt werden soll, um (beispielsweise) monatliche Arbeitslosenquoten berücksichtigen zu können. Im ersten Beispiel müssen die Angaben zu den betreffenden Veränderungen im Datensatz vorhanden sein. Im

7 Aus Gründen des Datenschutzes zeigen wir hier keine Originaldaten aus dem SOEP.

zweiten Beispiel hingegen sind die Splitting-Zeitpunkte gleichsam extern vorgegeben. Hier wäre ein Episodensplitting mit SPSS/PC+ zwar vom Grundprinzip her wiederum sehr einfach, aber mit enormem Aufwand verbunden, weshalb in diesem Fall SPSS für Windows eingesetzt wird. Dieses zweite Beispiel wird in Abschnitt 4.2 anhand der Bremer LSA-Daten vorgestellt. *Drittens* schließlich werden die ersten beiden Fälle kombiniert. In Abschnitt 4.3 wird vorgeführt, wie ein *Monatssplitting,* bei dem sich die zeitabhängigen Kovariaten *während des Monats ändern können,* durchzuführen ist.

4. Beispiele für das Episodensplitting mit SPSS

4.1 Episodensplitting zu individuenspezifischen Zeitpunkten

Wir beginnen mit dem Fall des Episodensplittings zu für die einzelnen Untersuchungspersonen spezifischen Zeitpunkten. Untersucht werden soll u.a. der Einfluß, den kleine Kinder im Haushalt auf den Verlauf der Arbeitslosigkeit haben. Wir können annehmen, daß diese bei Männern vergleichsweise bedeutungslos sind, während sie die Chancen von Frauen auf eine Beschäftigungsaufnahme deutlich verringern dürften (vgl. auch ***Ludwig-Mayerhofer*** 1990 anhand eines anderen Datensatzes). Als »kleine Kinder« wollen wir Kinder bis zum vollendeten 2. Lebensjahr definieren; dieser Wert hat sich in explorativen Analysen als am erklärungskräftigsten herausgestellt. Wie im obigen fiktiven Beispiel des Familienstands ist zu berücksichtigen, daß einerseits während der Arbeitslosigkeit Kinder geboren werden können, andererseits Kinder auch während der Arbeitslosigkeit das Alter von 2 Jahren überschreiten können, d.h. die zeitveränderliche Kovariate kann ihren Wert u.U. mehrfach wechseln. [8]

Es ist auch zu vermuten, daß kleine Kinder nicht erst bei der Geburt, sondern schon in einem gewissen Zeitabstand vorher die Wahrscheinlichkeit der Arbeitsaufnahme beeinflussen können. Dies ist aber kein grundsätzliches Problem, sondern nur eine Frage der Definition des Zeitpunktes; es ist sozusagen der Zeitpunkt der »Geburt« nach vorne zu verlegen. Wir werden im folgenden einen Zeitraum von 6 Monaten vor Geburt berücksichtigen, der Einfachheit halber aber weiterhin vom Zeitpunkt der »Geburt« sprechen. [9]

8 Das Beispiel ist damit - und auch, weil es mehrere Kinder geben kann, siehe unten - etwas komplexer als der von ***Blossfeld, Hamerle und Mayer*** (1986) herangezogene (und als mit SPSS nicht lösbar deklarierte) Fall des Heiratsdatums (Scheidung. Trennung oder Tod wurden dort nicht berücksichtigt). Eine zusätzliche Schwierigkeiten ist, daß Kinder auch den Haushalt der betreffenden Person verlassen können, etwa durch Trennung oder Scheidung vom Ehegatten oder auch durch Tod. Um eine zu große Unübersichtlichkeit der Beispiele zu vermeiden, haben wir letzteres in den folgenden Beispielen nicht berücksichtigt, wohl aber in den Datenauswertungen, die wir nachfolgend bringen.

9 In explorativen Auswertungen (zu den Ergebnissen siehe Abschnitt 5) hat sich dieser Abstand vor der Geburt als am erklärungskräftigsten herausgestellt. Dies entspricht auch einfachen »alltagstheoretischen« Überlegungen.

Wir wollen hier nicht auf die (nicht ganz einfachen) Details eingehen, wie aus den Daten des SOEP die relevanten Informationen hergestellt werden können. Es sei nur kurz darauf hingewiesen, daß für Kinder, die während der Laufzeit des SOEP - also ab 1983 - geboren wurden, die Geburtsdaten in Form von Monat und Jahr angegeben sind, während für früher geborene Kinder nur das Geburtsjahr bekannt ist. Die fehlenden Werte für den Geburtsmonat (die auch bei den eigentlich exakten Daten auftreten können) sollten sinnvollerweise durch einen geeigneten Wert (z.B. den Monat 7) substituiert werden. Außerdem ist zu berücksichtigen, daß Personen mehrere Kinder haben können. In der Analyse, deren Ergebnisse später kurz vorgestellt werden, wurden für maximal sechs Kinder jeder Person (und zwar die sechs jüngsten Kinder) die Geburtsdaten in geeigneter Form bereitgestellt. [10]

Wie ist es nun möglich, mit den Mitteln von SPSS/PC+ aus dieser personenbezogenen (genauer: episodenbezogenen) Datei eine Datei zu erzeugen, welche, wo erforderlich, mehrere Teilepisoden pro Fall enthält? Die Lösung ist einfach. Wir wollen hier eine sehr übersichtliche Befehlsfolge präsentieren, in der in einem ersten Schritt nur die »Geburt« eines Kindes berücksichtigt wird und in einem zweiten Schritt der Zeitpunkt, zu dem das Kind ein Alter von 24 Monaten überschreitet. Damit ist der erste Schritt direkt auf Variablen übertragbar (wie etwa den Zeitpunkt eines Ausbildungsabschlusses), die nicht »rückgängig« gemacht werden können. Mit geringfügigen Modifikationen ließen sich beide Schritte auch in einem Durchlauf erledigen.

Wir bilden zunächst für jede Arbeitslosigkeitsepisode zwei »Teilepisoden«: Die Teilepisode *bis* Geburt des ersten Kindes (im Datensatz) und die Teilepisode *ab* Geburt des ersten Kindes. Gibt es überhaupt keine Kinder (oder werden diese nach der Arbeitslosigkeitsepisode geboren), so braucht die Episode natürlich ebensowenig gesplittet zu werden wie dann, wenn ein Kind schon vor ihrem Beginn geboren wurde; nur der Wert der zeitveränderlichen Kovariaten »Zahl der Kinder« muß entsprechend kodiert werden. Anschließend werden die Teilepisoden nach Geburt des Kindes wieder in zwei Teilepisoden zerlegt: Die Teilepisode, bis das Kind ein Alter von 24 Monaten erreicht hat, und die Teilepisode danach. Auch hier kann die »Teilepisode danach« mit der gesamten Arbeitslosigkeitsepisode zusammenfallen, wenn nämlich das Kind schon vor ihrem Beginn älter als 24 Monate war. Diese Schritte können nun mit weiteren Kindern (oder auch beliebigen anderen Ereignissen) wiederholt werden. [11] Im folgenden Beispiel-Datensatz werden nur zwei Kinder berücksichtigt.

10 Im Datensatz des SOEP (genauer: in den Datensätzen der Wellen 1 bis 7) befinden sich Personen bzw. Haushalte mit maximal 10 Kindern. Da in der Auswertung auch Kinder bis zum Alter von 10 Jahren berücksichtigt werden sollten, schien es sinnvoll, die sechs jüngsten Kinder in jedem Haushalt einzubeziehen. Sind noch mehr Kinder vorhanden, kann davon ausgegangen werden, daß die ältesten Kinder während des Untersuchungszeitraumes schon älter als 10 Jahre sind, ganz abgesehen davon, daß es hinsichtlich der Arbeitsmarktteilnahme keinen Unterschied machen dürfte, ob eine Person sechs Kinder zu betreuen hat oder noch mehr.

11 An manchen Stellen könnten die nachfolgenden Programmbeispiele noch vereinfacht werden. Wir haben versucht, den Code einerseits so einfach wie möglich, andererseits möglichst deutlich zu halten.

Zuvor noch zwei Hinweise. Erstens wollen wir anmerken, daß - wie oben erwähnt - im Regelfall bei der *Datenauswertung* der Beginn der (ursprünglichen!) Episoden auf den Wert »0« gesetzt werden muß. Auch dies wäre grundsätzlich entweder schon vor dem Episodensplitting - wobei die Zeitpunkte der Geburt der Kinder entsprechend modifiziert werden müßten - oder aber unmittelbar bei der Durchführung des Splittings möglich. Wir wählen hier der Anschaulichkeit halber den Weg, den Beginn der »ursprünglichen« Episode als zusätzliche Variable im Datensatz zu belassen; die entsprechende Beziehung der einzelnen Teilepisoden auf diesen Beginn kann dann später realisiert werden.[12]

Ferner ein Hinweis zur speziellen Datenstruktur des SOEP. Dauert eine Episode einen Monat, so sind bei dieser Episode Anfangs- und Endmonat identisch. Dies bringt in der Datenauswertung Probleme, da ja die Dauer einer Episode aus der Differenz von Anfangs- und Endzeit bestimmt wird und Episoden von einem Monat Dauer scheinbar eine Dauer von 0 erhalten (was sowohl inhaltlich als auch datentechnisch nicht sinnvoll ist). Daher muß eine Konstante zu der Dauer addiert werden, was jedoch nach dem Episodensplitting nicht mehr geschehen darf, da dann diese Konstante zu jeder Teilepisode hinzugezählt würde. Daher gehen wir davon aus, daß bei den folgenden Daten schon vorher eine Konstante von 1 zum Episoden-Ende addiert wurde. Nach den Überlegungen von ***Hujer und Schneider*** (1986) sowie ***Petersen und Koput*** (1992) wäre es sinnvoller, einen Wert von 0,5 zu addieren (jedenfalls für die Analyse mit Modellen für kontinuierliche Verweildauern), da es sich bei den Daten des SOEP um »gruppierte« oder »aggregierte« Verweildauern handelt. Aus folgenden Gründen scheint uns eine Verlängerung um den Wert 1 gerechtfertigt: In der großen Mehrzahl der Fälle haben die Befragten im SOEP den Erwerbskalender so ausgefüllt, daß eine Episode jeweils in dem Monat *nach* dem Ende der vorherigen Episode beginnt (z.B. Ende einer Episode in Monat 8, Beginn der nächsten Episode im Monat 9). Ganz in Analogie zur oben dargestellten Logik des Episodensplittings ist es zur Erzeugung einer konsistenten Abfolge von Episoden am sinnvollsten, die Episoden in dem Monat enden zu lassen, in dem die nächste Episode beginnt (was im Grunde bedeutet, daß der Übergang von einem Zustand in den anderen um 0 Uhr des Monatsersten stattfindet). Würde man andere Werte als 1 hinzuaddieren, wäre die Summe aller Episoden entweder kürzer (Addition eines Betrages < 1) oder länger (Addition eines Betrages >1) als die Gesamtdauer des beobachteten Verlaufs. Auch hier gilt also, daß man (an sich sinnvolle) Regeln nicht schematisch übertragen, sondern die Datenlage im konkreten Einzelfall berücksichtigen sollte.

Einige - wiederum fiktive - Datensätze sehen folgendermaßen aus:

data list free/

fallnr	beginn	ende	zustl	zust2	kindlm	kindlj	kind2m	kind2j.
1	2	16	4	1	3	84	99	99
2	2	8	4	1	3	84	99	99
3	2	8	4	1	1	83	99	99
4	2	16	4	4	11	80	*99*	*99*
5	2	16	4	4	11	81	99	99
6	2	50	4	4	7	79	10	83
7	2	60	4	4	10	83	7	85
8	2	50	4	1	10	83	7	85

12 Dies ist in dem Beispiel bei ***Blossfeld, Hamerle und Mayer*** (1986, S. 196) problematisch, da der Beginn der (Teil-)Episoden nicht entsprechend modifiziert wird, aber auch nicht, wie hier, der Beginn der ursprünglichen Episode im Datensatz behalten wird. Der dort vorgestellte Datensatz kann nur mit Modellen analysiert werden, die keine Zeitabhängigkeit des untersuchten Prozesses berücksichtigen, wie einem einfachen Exponentialmodell oder einem Cox-Modell.

Die eiste Variable ist die Fallnummer, es folgen die Zeitpunkte von Beginn und Ende des Prozesses. Die nächsten beiden Variablen beziehen sich auf Ausgangs- (ZUST1) und Endzustand (ZUST2), anschließend folgen die Angaben zur Geburt des ersten (KIND1M, KIND1J) und des zweiten Kindes (KIND2M, KIND2J). Fehlende Werte (wenn also kein Kind vorhanden ist) sind mit 99 kodiert.

Selbstverständlich könnten danach noch weitere Kovariaten folgen, wobei es je nach Umfang der Variablenmenge ratsam sein kann, diese erst später hinzuzufügen, da die Rechenzeit von SPSS erheblich von der Zahl der Variablen im Datensatz (und nicht nur der aktuell angesprochenen Variablen) beeinflußt wird.[13] Die folgenden Anweisungen erzeugen nun den gesplitteten Datensatz.

```
1   compute begmon = beginn.
2   compute kind = 0.
3   compute kbeg = kind1m + (kind1j - 83) * 12 - 6.

*   Teilepisoden bis Geburt eines Kindes.
4   compute tbeg = -1.
*   Wenn Geburt nach Ende der Episode ...
5   if (kbeg ge ende) tbeg = beginn.
6   if (kbeg ge ende) tend = ende.
7   if (kbeg ge ende) tzust2 = zust2.
*   Wenn Geburt während der Episode ...
8   if (kbeg lt ende and kbeg gt beginn) tbeg = beginn.
9   if (kbeg lt ende and kbeg gt beginn) tend = kbeg.
10  if (kbeg lt ende and kbeg gt beginn) tzust2 = zust1.
11  compute kind = kind + 0.
12  process if (tbeg ge 0).
13  save out = 'file1.sys'.

*   Teilepisoden nach Geburt
14  compute tbeg = -1.
*   Wenn Geburt vor Beginn der Episode ...
15  if (kbeg le beginn) tbeg = beginn.
16  if (kbeg le beginn) tend = ende.
17  if (kbeg le beginn) tzust2 = zust2.
*   Wenn Geburt während der Episode ...
18  if (kbeg lt ende and kbeg gt beginn) tbeg = kbeg.
19  if (kbeg lt ende and kbeg gt beginn) tend = ende.
20  if (kbeg lt ende and kbeg gt beginn) tzust2 = zust2.
21  compute kind = kind + 1.

*   Auswahl der Teil-Episoden ab Geburt.
22  select if (tbeg ge 0).
23  compute beginn = tbeg.
24  compute ende = tend.
25  compute zust2 = tzust2.
26  compute kend = kind1m + (kind1j - 83) * 12 + 24.

*   Teilepisoden: Lebensalter des Kindes bis 24 Monate.
27  compute tbeg = -1.
*   Wenn Kind nach der Episode 24 Monate wird ...
```

13 Hierfür ist die Prozedur »join match« in Verbindung mit der »table«-Option hilfreich.

```
28  if (kend ge ende) tbeg = beginn.
29  if (kend ge ende) tend = ende.
30  if (kend ge ende) tzust2 = zust2.
*   Wenn Kind während der Episode 24 Monate wird ...
31  if (kend lt ende and kend ge beginn) tbeg = beginn.
32  if (kend lt ende and kend ge beginn) tzust2 = zust1.
34  compute kind = kind.
35  process if (tbeg ge 0).
36  save out = 'file2.sys'.

*   Teilepisoden: Lebensalter des Kindes mehr als 24 Monate.
37  compute tbeg = -1.
*   Wenn Kind vor der Episode älter als 24 Monate ...
38  if (kend lt beginn) tbeg = beginn.
39  if (kend lt beginn) tend = ende.
40  if (kend lt beginn) tzust2 = zust2.
*   Wenn Kind während der Episode älter als 24 Monate ...
41  if (kend lt ende and kend ge beginn) tbeg = kend.
42  if (kend lt ende and kend ge beginn) tend = ende.
43  if (kend lt ende and kend ge beginn) tzust2 = zust2.
44  compute kind = kind - 1.
45  select if (tbeg ge 0).

*   Zusammenfügen der Teilepisoden.
46  add file =  *
        /file = 'file1.sys'
        /file = 'file2.sys'.
*   Wiederholung der Prozedur für das zweite Kind.
47  compute beginn = tbeg.
48  compute ende = tend.
49  compute zust2 = tzust2.
50  compute kbeg = kind2m + (kind2j - 83) * 12 - 6.
*   Usw. wie Zeilen 4 bis 46.
```

Zunächst wird der Monat des Beginns der Episode einer neuen Variablen BEGMON zugewiesen (Zeile 1) und die Variable KIND (Zahl der Kinder) wird mit dem Wert 0 initialisiert (Zeile 2). Zeile 3 definiert den Zeitpunkt 6 Monate vor der Geburt des Kindes in Relation zur zugrundeliegenden Zeitskala (Variable KBEG). In den Zeilen 4 bis 13 werden die Teilepisoden bis zur so definierten »Geburt« gebildet, deren Beginn notwendigerweise mit dem Beginn der Gesamtepisode identisch ist (Variable TBEG, siehe Zeile 5 bzw. 8). Wird das Kind erst nach der Arbeitslosigkeit »geboren«, sind Ende (Variable TEND) und Zielzustand (TZUST2) der Teilepisode identisch mit der gesamten Episode (Zeile 6 und 7); bei Geburt während der Arbeitslosigkeitsepisode endet die Teilepisode jedoch mit der Geburt des Kindes und ist zensiert (Zeile 9 und 10). Da kein Kind geboren wurde, wird die Variable KIND *um* den Wert 0 »erhöht«, und die so definierten Teilepisoden werden in einen Datensatz geschrieben (Zeile 11 bis 13). Ganz analog dazu werden die Teilepisoden ab »Geburt« definiert, wobei wiederum unterschieden werden muß zwischen dem Fall, daß das Kind bereits vor der Arbeitslosigkeit »geboren« wurde (Zeile 15 bis 17) und dem Fall der »Geburt« während der Arbeitslosigkeit (Zeile 18 bis 20). Die Variable KIND muß hier um den Wert 1 erhöht werden (Zeile 21).

Wenn nur die Tatsache der Geburt von Kindern berücksichtigt werden müßte, nicht aber das Überschreiten eines Alters von 24 Monaten, wäre die Datenaufbereitung mit der Selektion der jetzt definierten Teilepisoden (Zeile 22) und deren Zusammenfügen mit der vorher gesicherten Datei (im Beispiel genannt "file1.sys") beendet. So aber muß mit den jetzt definierten Teilepisoden weitergearbeitet werden. Zunächst werden aus Gründen der Vereinfachung Beginn, Ende und Zielzustand der Teilepisoden den alten Variablennamen zugewiesen (Zeile 23 bis 25). Dann wird in Analogie zur Geburt des Kindes der Zeitpunkt bestimmt, an dem das Kind 24 Monate alt wird (Variable KEND, Zeile 26). Jetzt können genau wie oben die so definierten Episoden in eine Teilepisode bis zum Alter von 24 Monaten (Zeile 27 bis 36) und eine Teilepisode nach dem Erreichen dieses Alters (Zeile 37 bis 44) gesplittet werden, wobei im letzteren Fall natürlich die Variable KIND um den Wert 1 verringert werden muß. Nachdem die im letzten Schritt definierten Teilepisoden ausgewählt wurden (Zeile 45), können sie mit den vorher »zwischengesicherten« Teilepisoden zu einer Datei zusammengefügt werden (Zeile 46).

Diese Daten können jetzt für das Splitting beim zweiten Kind verwendet werden. Hierzu werden wieder die hier neu definierten Daten für Beginn, Ende und Zielzustand den alten Variablen(namen) zugewiesen (Zeile 47 bis 49), und nunmehr kann in völliger Analogie zu oben, unter Berücksichtigung, daß nunmehr der Geburtszeitpunkt des zweiten Kindes definiert werden muß (Zeile 50; die Anweisung aus Zeile 26 muß natürlich ebenfalls modifiziert werden), das Splitting von neuem beginnen. Offensichtlich kann dieser Prozeß beliebig oft wiederholt werden.

Kurz hingewiesen sei noch darauf, wie berücksichtigt wird, wenn keine Kinder vorhanden sind. In diesem Fall wird durch die gewählte Kodierung für Monat und Jahr der Geburt ein Geburtszeitpunkt erzeugt, der weit nach Ende der Beobachtungsdauer liegt, so daß die entsprechenden Episoden immer als Episoden »vor der Geburt« definiert werden.

Nach diesen Transformationen ist der gesplittete Datensatz beinahe für die Verlaufsdaten¬analyse fertig. Es muß nur noch der Beginn der ursprünglichen Episode (BEGMON) von den Variablen TBEG und TEND abgezogen werden, damit der Prozeß auch zum Zeitpunkt Null beginnt. Zur besseren Vergleichbarkeit mit den Ausgangsdaten zeigen wir im folgenden aber die Daten, wie sie *vor* diesem Schrift aussehen:

FALLNR	BEGMON	TBEG	TEND	ZUST1	TZUST2	KIND
1	2	2	9	4	4	0
1	2	9	16	4	1	1
2	2	2	8	4	1	0
3	2	2	8	4	1	1
4	2	2	16	4	4	0
5	2	2	11	4	4	1
5	2	11	16	4	4	0
6	2	2	4	4	4	0
6	2	4	34	4	4	1
6	2	34	50	4	4	0
7	2	2	4	4	4	0
7	2	4	25	4	4	1
7	2	25	34	4	4	2
7	2	34	55	4	4	1
7	2	55	60	4	4	0
8	2	2	4	4	4	0
8	2	4	25	4	4	1
8	2	25	34	4	4	2
8	2	34	50	4	1	1

Hier noch zwei Anmerkungen zu der Übertragung des Beispieles auf andere Programmpakete, Erstens seien Nicht-SPSS-Nutzer darauf hingewiesen, daß der Befehl »compute« nichts anderes als eine unbedingte Variablendefinition bewirkt, die in den meisten anderen Programmen - wie in den gängigen Programmiersprachen - einfach durch den Zuweisungsoperator erfolgt. Zweitens ist der Befehl »process if« zu erläutern. Er bedeutet eine temporäre Selektion von Fällen, die nur für den unmittelbar darauffolgenden Schritt gültig ist; anschließend steht wieder der gesamte Datensatz zur Verfügung. Sollte ein Programm über eine solche Möglichkeit nicht verfugen, muß der Datensatz nach der Fallselektion neu eingelesen werden.

Ein abschließender Hinweis zu diesem Teil: Im Falle sehr seltener Ereignisse wäre es denkbar, daß in einem Teilschritt überhaupt keine Episoden erzeugt werden, die den genannten Kriterien entsprechen. In diesem Fall würde SPSS/ PC^{+} nach der entsprechenden »process if« oder »select if« Anweisung mit einer Fehlermeldung, daß keine Fälle mehr zur Verfügung stehen, abbrechen. Tritt eine solche Meldung auf, müßte der Programmcode ad hoc so modifiziert werden, daß die auf die jeweilige Teil-Datei bezogenen Anweisungen sowohl beim Speichern bzw. der Auswahl der Teilepisoden als auch beim abschließenden Zusammenfügen derselben eliminiert werden.

4.2 Episodensplitting zu vorgegebenen Zeitpunkten

Im folgenden Beispiel soll der Bremer LSA-Datensatz in regelmäßigen monatlichen Abständen gesplittet werden. In unserem Beispiel ist dies erforderlich, um monatliche Arbeitslosenquoten berücksichtigen zu können. Der Datensatz der LSA unterscheidet sich vom SOEP vor allem dadurch, daß die Sozialhilfedauer explizit durch das genaue Kalenderdatum von Beginn und Ende definiert ist und viele Episoden auch mitten im Monat beginnen oder enden. Deshalb wird in den folgenden zwei Beispielen als Zeiteinheit ein Tag gewählt (bei obigem Beispiel mit dem SOEP war die Zeiteinheit ja ein Monat). Wir verwenden daher für das Episodensplitting eine Funktion, die Daten im gregorianischen Kalender in Tage seit einem fixen Bezugspunkt umrechnet. In SPSS heißt diese Funktion *yrmoda;* sie hat als Bezugspunkt den 14. Oktober 1582.

Die erste zeitabhängige Kovariate dieses zweiten Beispiels ist die monatliche Arbeitslosenquote in Bremen. Am günstigsten ist es, zuerst die Nummer des Monats der jeweiligen Teilepisode (bezogen auf den Januar 1983) zu konstruieren, und dann später die entsprechenden Arbeitslosenquoten aus einer eigenen Datei zu »matchen«. Wir demonstrieren im folgenden deshalb nur die Konstruktion der Monatsnummer. Die zweite zeitabhängige Variable sei wiederum die Kinderzahl. Auch in den LSA-Daten ist nur der Geburtsmonat eines Kindes bekannt. Deshalb liegt es nahe, die zeitabhängige Kovariate ab dem Geburtsmonat eines Kindes um eins zu erhöhen; wir setzen damit den Geburtszeitpunkt implizit auf den 1. eines Monats. Aus diesem Grund ist es bei diesem zweiten Beispiel nicht nötig, die Monate nochmals aufzusplitten. (Obwohl auch die LSA-Daten Angaben darüber enthalten, wann ein Kind den Haushalt verläßt, berücksichtigen wir dies hier nicht, um das Beispiel nicht zu kompliziert zu machen). Im dritten Beispiel werden wir annehmen, daß die Geburt am 15. eines Monats erfolgt, denn mancher Datensatz wird auch den Tag der Geburt enthalten. Dann wird es nötig, jeden Monat, in dem eine Geburt erfolgte, aufzusplitten.

Im Prinzip könnte man ein Monatssplitting auch mit SPSS/PC$^+$ analog zum ersten Beispiel durchführen. Da die Beobachtungszeit allerdings mehrere Jahre beträgt, muß man Dutzende von Datensätzen erzeugen. Wesentlich einfacher ist es deshalb, sich die Möglichkeit einen Datensatz mehrmals einzulesen, die SPSS für Windows bietet, zunutze zu machen.

Zuerst ist ein ASCII-Datensatz (in freiem Format am besten) zu erzeugen, der alle für das Episodensplitting nötigen Informationen enthält: Fallnummer (FALLNR), Beginn der Episode (BEGT, BEGM, BEGJ), Ende der Episode (ENDT, ENDM, ENDJ), Endzustand (ZUST2), Geburtsdatum des 1. Kindes (KIND1M, KIND1J), Geburtsdatum des 2. Kindes (KIND2M, KIND2J). Zusätzlich muß bei diesem Schritt bereits eine Variable konstruiert

werden, die die Zahl der Monatssplitts angibt (ZSPLIT). Die Programmanweisungen hierfür lauten:

```
compute jahre = endj-begj.
do if (jahre=0).
+ compute zsplit = endm-begm+1.
else if (jahre=1).
+ compute zsplit = 13-begm + endm.
else if (jahre ge 2).
+ compute zsplit = 13-begm + endm + (jahre-1)*12.
end if.

compute begmon = begm + (begj-83)*12.
```

Jeder Monat, während dessen die Episode andauerte, einschließlich des Beginn- und Endmonats, macht eine Teilepisode nötig. Der erste und der letzte Monatssplit kann dabei zwar eventuell nur wenige Tage lang sein (wenn der Beginn nahe beim Monatsende oder das Ende nahe beim Monatsanfang liegt), aber dennoch ist eine eigene Teilepisode erforderlich. Selbst wenn das Ende am Ersten eines Monats liegt, ist für diesen Monat eine Teilepisode von der Länge eines Tages nötig (es wird am Monatsersten 0.00 Uhr aufgesplittet). Zusätzlich wird noch die Hilfsvariable BEGMON erzeugt, die die Nummer des Beginnmonats bezogen auf den Januar 1983 enthält. Die Variable ZSPLIT sollte an erster Stelle des ASCII-Datensatzes abgespeichert werden. Unser Beispieldatensatz (BSP2.D1) sieht dann folgendermaßen aus:

ZSPLIT	FALLNR	BEG			END			ZUST2	KIND1		KIND2		BEGMON
16	1	3	9	83	31	12	84	3	9	83	10	84	9
10	2	21	3	83	31	12	83	2	10	84	9	99	3
1	3	6	6	83	16	6	83	3	6	83	9	99	6
11	4	1	7	83	10	5	84	3	9	99	9	99	7
11	5	1	7	83	10	5	84	3	1	82	9	99	7
11	6	1	7	83	15	5	84	3	5	84	9	99	7
14	7	31	12	83	1	1	85	2	12	83	11	84	12

Die erste Episode beginnt am 3.9.83 und endet am 31.12.84. Damit sind 16 Monatssplits erforderlich. Der Endzustand ist 3, das erste Kind wurde im September 1983 geboren, das zweite Kind im Oktober 1984. Der Beginnmonat (September 1983) ist der 9. Monat.

Dieser ASCII-Datensatz wird nun mit dem folgenden INPUT PROGRAM für SPSS für Windows eingelesen, aufgesplittet und als neuer ASCII-Datensatz abgespeichert:[14]

14 Die »data list«-Anweisung in Zeile 2 setzt voraus, daß jeder Record im Datensatz nur eine Zeile umfaßt; sonst müßte sie entsprechend modifiziert werden.

```
1   input program.
2   data list file="bsp2.d1" free
        /zsplit.

3   loop #i=1 to zsplit.
4   + reread.
5   + data list free
        /zsplit fallnr begt begm begj endt endm endj zust2
         kind1m kind1j kind2m kind2j begmon.

6   + compute begdat = yrmoda(begj,begm,begt).
7   + compute ind = trunc( (begm+#i-2) / 12 ).
8   + compute sj = begj + ind.
9   + compute sm = begm+#i-1 - ind*12.
10  + compute st = 1.
11  + if (#i=1) st = begt.

12  + compute monat = begmon+#i-1.
13  + compute kind = 0.
14  + do if (kind1j ne 99).
15  -   if (yrmoda(sj,sm,1) ge yrmoda(kind1j,kind1m,1)) kind = 1.
16  + end if.
17  + do if (kind2j ne 99).
18  -   if (yrmoda(sj,sm,1) ge yrmoda(kind2j,kind2m,1)) kind = 2.
19  + end if.

20  + do if (#i lt zsplit).
21  -   compute   tbeg = yrmoda(sj,sm,st)-begdat.
22  -   compute   tend = yrmoda(sj,sm+1,1)-begdat.
23  -   compute   tzust2 = 0.
24  + else if (#i=zsplit).
25  -   compute       tbeg = yrmoda(sj,sm,st)-begdat.
26  -   compute       tend = yrmoda(endj,endm,endt)-begdat+1.
27  -   compute       tzust2 = zust2.
28  + end if.

29  + end case.
30  end loop.
31  end input program.

32  compute diff=tend-tbeg.
33  freq var=diff.

34  write outfile="bsp2.d2"
        /fallnr monat tbeg tend tzust2 kind
        (f1.0,3x,f2.0,3x,f3.0,3x,f3.0,3x,f1.0,3x,f1.0).
35  execute.
```

Zuerst wird die Zahl der nötigen Teilepisoden eingelesen (Zeile 2). Dann wird eine Schleife definiert, die ZSPLIT-mal abgearbeitet wird (Zeile 3). Bei jedem Durchlauf wird die Datenzeile neu eingelesen (Zeilen 4 und 5). Die dann folgenden Anweisungen in den Zeilen 6-28 definieren dann jeweils eine Teilepisode, die mit der END CASE Anweisung (Zeile 29) in die Arbeitsdatei übernommen wird. Zuerst werden zwei Hilfsvariablen konstruiert. BEGDAT (Zeile 6) enthält den Anfangszeitpunkt der Episode, und IND (Zeile 7) ist ein Indikator, der die Zahl der übersprungenen Jahresgrenzen festhält. Dann werden mit SJ, SM und ST das Anfangsjahr, der Anfangsmonat und der Anfangstag der Teilepisode errechnet (Zeilen 8-11). Als nächstes werden die zeitabhängigen Kovariaten aktualisiert

(Zeilen 12-19). Zuerst wird bei jeder Teilepisode die Monatsnummer um eins erhöht (MONAT), dann wird die aktuelle Kinderzahl festgestellt (KIND). Schließlich wird der Beginnzeitpunkt (TBEG), der Endzeitpunkt (TEND) und der Endzustand (TZUST2) errechnet (Zeilen 20-28). Dabei sind zwei Fälle zu unterscheiden. Bei allen Teilepisoden, außer der letzten, ist der Endzeitpunkt der Erste des folgenden Monats und der Endzustand ist gleich dem Ausgangszustand (die Teilepisode ist mithin zensiert). Bei der letzten Teilepisode ist der Endzeitpunkt das tatsächliche Ende der Episode plus einen Tag (es wird angenommen, daß eine Episode am Endtag 24.00 Uhr endet). Der Endzustand ist gleich dem Endzustand der Episode. Damit ist eine Teilepisode fertig, und ein neuer Durchlauf beginnt (Zeile 30).

Als nächstes empfehlen sich einige Daten-Checks. In den Zeilen 32 und 33 wird hierzu die Differenz zwischen Endzeitpunkt und Beginnzeitpunkt der Teilepisoden berechnet. Bei korrektem Episodensplitting muß diese Differenz zwischen 1 und 31 betragen. Als weiteren Check sollte man sich einige Fälle der gesplitteten Daten ansehen, und mit den Ausgangsdaten vergleichen. Zum Schluß wird in Zeile 34 eine ASCII-Datei (BSP2.D2) herausgeschrieben, die alle nötigen Informationen enthält, um die gesplitteten Daten mit einem Programmpaket für Ereignisdatenanalyse weiter zu analysieren: den Beginnzeitpunkt (TBEG), den Endzeitpunkt (TEND), den Endzustand (TZUST2) und die beiden zeitabhängigen Kovariate (MONAT, KIND). Der Datensatz in unserem Beispiel sieht dann so aus:

FALLNR	MONAT	TBEG	TEND	TZUST2	KIND
1	9	0	28	0	1
1	10	28	59	0	1
1	11	59	89	0	1
1	12	89	120	0	1
1	13	120	151	0	1
1	14	151	180	0	1
1	15	180	211	0	1
1	16	211	241	0	1
1	17	241	272	0	1
1	18	272	302	0	1
1	19	302	333	0	1
1	20	333	364	0	1
1	21	**364**	394	0	1
1	22	394	425	0	2
1	23	425	455	0	2
1	24	455	486	3	2
7	12	0	1	0	1
7	13	1	32	0	1
7	14	32	61	0	1
7	15	61	92	0	1
7	16	92	122	0	1
7	17	122	153	0	1
7	18	153	183	0	1
7	19	183	214	0	1
7	20	214	245	0	1
7	21	245	275	0	1
7	22	275	306	0	1
7	23	306	336	0	2
7	24	336	367	0	2
7	25	367	368	2	2

Für die Episode 1 wurden 16 Teilepisoden generiert. Die erste Teilepisode beginnt im 9. Monat (September 1983). Der Beginnzeitpunkt ist 0 und der Endzeitpunkt ist 28 (vom 3.9.83 bis zum 1.10.83 sind es 28 Tage). Der Endzustand ist 0, denn es handelt sich nicht um die letzte Teilepisode. Das erste Kind dieser Person wurde (annahmegemäß) am 1.9.83 geboren, weshalb die Kinderzahl 1 ist. Man sieht, daß alle Teilepisoden jeweils mit dem Endzeitpunkt der vorhergehenden Teilepisode beginnen. Im Monat 22 (Oktober 1984) erhöht sich die Kinderzahl auf 2, weil in diesem Monat das zweite Kind zur Welt kam. Schließlich ist bei der letzten Teilepisode der Endzustand gleich 3, was dem Endzustand der Episode entspricht. Analog kann man auch die Episode 7 rekonstruieren. Man beachte, daß diese Episode so konstruiert wurde, daß die erste und die letzte Teilepisode jeweils nur einen Tag lang sind.

4.3 Episodensplitting zu vorgegebenen und individuenspezifischen Zeitpunkten

In diesem Abschnitt wenden wir uns der Kombination obiger Beispiele zu: Wir erzeugen wieder Monatssplits, nehmen aber nun an, daß der Tag der Geburt eines Kindes bekannt ist (hier soll es der 15. des Geburtsmonats sein). Damit wird es erforderlich, die Monats-Teilepisode, in die die Geburt eines Kindes fällt, in zwei Teilepisoden (vor und nach dem Tag der Geburt) aufzusplitten. Dies erfordert einige Erweiterungen der Programme aus Beispiel 2.

Zuerst wird wie oben ein Episodendatensatz erzeugt (BSP3.D1). Zusätzlich zu den Anweisungen bei Beispiel 2, mit denen ZSPLIT und BEGMON generiert werden, sind hier folgende Anweisungen nötig:

```
1   compute kind1t = 15.
2   compute kind2t = 15.
3   compute endmon = endm + (endj-83)*12.

4   do if (kind1j=99).
5   - compute       emon1 = 0.
6   else if (kind1j ne 99).
7   - compute       emon1 = kind1m + (kind1j-83)*12.
8   - if (kind1t=1) emon1 = 0.
9   -if(yrmoda(kind1j,kind1m,kind1t) <=yrmoda(begj,begm,begt))emon1=0.
10  -if(yrmoda(kind1j,kind1m,kind1t) > yrmoda(endj,endm,endt))emon1=0.
11  end if.

12  do if (kind2j=99).
13  - compute       emon2 = 0.
14  else if (kind2j ne 99).
15  - compute       emon2 = kind2m + (kind2j-83)*12.
16  - if (kind2t=1) emon2 = 0.
17  -if(yrmoda(kind2j,kind2m,kind2t) <=yrmoda(begj,begm,begt))emon2=0.
18  -if(yrmoda(kind2j,kind2m,kind2t) > yrmoda(endj,endm,endt))emon2=0.
19  end if.
```

In den Zeilen 1 und 2 wird der Tag der Geburt auf 15 gesetzt (was bei Anwendungen dieses Programms nicht erforderlich wäre, bei denen der Tag der Geburt bekannt ist). In der Zeile 3 wird die Nummer des Endmonats berechnet. In den Zeilen 4-11 wird die Nummer des ersten Ereignismonats errechnet (analog in den Zeilen 12-19 für die zweite Geburt). Im nachfolgenden Splitting-Programm wird diese Nummer dann dazu verwendet, den Monat zu identifizieren, der aufgesplittet werden muß. Deshalb wird der Ereignismonat auf 0 gesetzt, wenn (1) keine Geburt stattfand, (2) eine Geburt am 1. eines Monats stattfand (die Geburtszeit wird als Tag der Geburt 0.00 Uhr angenommen), (3) die Geburt vor dem Episodenbeginn stattfand, oder (4) die Geburt nach dem Episodenende stattfand. Nur wenn die Nummer des Ereignismonats größer null ist, wird im nachfolgenden Programm der entsprechende Monat in zwei Teilsplits zerlegt. Der Episodendatensatz BSP3.D1 sieht dann folgendermaßen aus:

ZSPLIT

FALLNR	BEG			END			ZUST2	KIND1		KIND2		BEGMON	EMON1	EMON2
16	1	3	9	83	31	12	84	3	9	83	10	84	9	9	22
10	2	21	3	83	31	12	83	2	10	84	9	99	3	0	0
1	3	6	6	83	16	6	83	3	6	83	9	99	6	6	0
11	4	1	7	83	10	5	84	3	9	99	9	99	7	0	0
11	5	1	7	83	10	5	84	3	1	82	9	99	7	0	0
11	6	1	7	83	15	5	84	3	5	84	9	99	7	17	0
14	7	31	12	83	1	1	85	2	12	83	11	84	12	0	23

Insgesamt müssen also 5 Monate aufgesplittet werden. Bei Episode 1 z.B. müssen der Monat 9 und der Monat 22 jeweils in zwei Teilepisoden zerlegt werden. Das INPUT PROGRAM für SPSS für Windows sieht nun folgendermaßen aus:

```
1   input program.
2   data list file="bsp3.d1" free
        /zsplit.

3   loop #i=1 to zsplit.
4   + reread.
5   + data list free
        /zsplit fallnr begt begm begj endt endm endj zust2
         kind1m kind1j kind2m kind2j begmon emon1 emon2.

6   + compute kind1t = 15.
7   + compute kind2t = 15.
8   + compute begdat = yrmoda(begj,begm,begt).
9   + compute ind = trunc( (begm+#i-2) / 12 ).
10  + compute sj = begj + ind.
11  + compute sm = begm+#i-1 - ind*12.
12  + compute st = 1.
13  + if (#i=1) st = begt.
14  + compute monat = begmon+#i-1.
15  + compute kind = 0.
16  + do if (kind1j ne 99).
17  -   if (yrmoda(sj,sm,st) ge yrmoda(kind1j,kind1m,kind1t)) kind=1.
18  + end if.
```

```
19  + do if (kind2j ne 99).
20  -   if (yrmoda(sj,sm,st) ge yrmoda(kind2j,kind2m,kind2t)) kind=2.
21  + end if.

22  + leave kind1t,kind2t,begdat,sj,sm,st,monat,kind.

23  + do if (monat ne emon1 and monat ne emon2).
      .
      . Zeilen 20-28 aus Beispiel 2

24  + else if (monat=emon1).

25  -   compute   tbeg = yrmoda(sj,sm,st)-begdat.
26  -   compute   tend = yrmoda(kind1j,kind1m,kind1t)-begdat.
27  -   compute   tzust2 = 0.
28  -   end case.

29  -   reread.
30  -   data list free
          /zsplit fallnr begt begm begj endt endm endj zust2
           kind1m kind1j kind2m kind2j begmon emon1 emon2.
31  -   compute   kind = kind+1.
32  -   compute   tbeg = yrmoda(kind1j,kind1m,kind1t)-begdat.
33  -   do if (#i lt zsplit).
34  -     compute   tend = yrmoda(sj,sm+1,1)-begdat.
35  -     compute   tzust2 = 0.
36  -   else if (#i=zsplit).
37  -     compute   tend = yrmoda(endj,endm,endt)-begdat+1.
38  -     compute   tzust2 = zust2.
39  -   end if.

40  + else if (monat=emon2).
      .
      . Analog für das zweite Kind
      .
41  + end if.

42  + end case.
43  end loop.
44  end input program.
```

Zuerst wird wieder die Schleife gestartet, die Datenzeile eingelesen und der Tag der Geburt auf den 15. festgelegt. Ebenso wie in Beispiel 2 wird dann der Beginn der Teilepisode festgestellt (Zeilen 8-13), und die zeitabhängigen Kovariaten werden aktualisiert (Zeilen 14-21). Hier ist allerdings zu beachten, daß nun der Tag der Geburt bekannt ist und nicht mehr als der Monatserste angenommen wird. Nun muß zusätzlich eine LEAVE-Anweisung angeführt werden (Zeile 22), damit beim Aufsplitten eines Ereignismonats der Monatsbeginn, die Monatsnummer und die aktuelle Kinderzahl durch das erforderliche Neueinlesen der Datenzeile nicht verloren gehen. Als nächstes müssen wieder Beginnzeitpunkt, Endzeitpunkt und Endzustand einer Teilepisode berechnet werden. Hierbei sind nun drei Fälle zu unterscheiden. (1) Es handelt sich nicht um einen Ereignismonat (Zeile 23): Die Berechnung unterscheidet sich nicht von der im 2. Beispiel angeführten. (2) Es handelt sich um den ersten Ereignismonat (Zeile 24): Der Endzeitpunkt der ersten Monatsteilepisode ist

nun das Geburtsdatum des Kindes. Diese Monatsteilepisode wird dann in die Arbeitsdatei geschrieben (Zeile 28) und die Datenzeile wird erneut eingelesen (Zeilen 29 und 30). Die Kinderzahl wird um eins erhöht, der Beginnzeitpunkt der zweiten Monatsteilepisode ist das Geburtsdatum, und der Endzeitpunkt wird wie gehabt bestimmt. (3) Es handelt sich um den zweiten Ereignismonat (Zeile 40): Analog zum ersten Ereignismonat.

Zum Schluß sollte man wieder Daten-Checks durchführen und die Daten als ASCII-Datei herausschreiben. Die Datei sieht dann folgendermaßen aus:

FALLNR	MONAT	TBEG	TEND	TZUST2	KIND
1	9	0	12	0	0
1	9	12	28	0	1
1	10	28	59	0	1
1	11	59	89	0	1
1	12	89	120	0	1
1	13	120	151	0	1
1	14	151	180	0	1
1	15	180	211	0	1
1	16	211	241	0	1
1	17	241	272	0	1
1	18	272	302	0	1
1	19	302	333	0	1
1	20	333	364	0	1
1	21	364	394	0	1
1	22	394	408	0	1
1	22	408	425	0	2
1	23	425	455	0	2
1	24	455	486	3	2
.					
.					
.					
7	12	0	1	0	1
7	13	1	32	0	1
7	14	32	61	0	1
7	15	61	92	0	1
7	16	92	122	0	1
7	17	122	153	0	1
7	18	153	183	0	1
7	19	183	214	0	1
7	20	214	245	0	1
7	21	245	275	0	1
7	22	275	306	0	1
7	23	306	320	0	1
7	23	320	336	0	2
7	24	336	367	0	2
7	25.	367	368	2	2

Beim ersten Fall erkennt man, daß bereits der erste Monat in zwei Teilepisoden zerlegt wurde. Am 12. Tag wurde das Kind geboren (Beginn: 3.9., Geburt: 15.9.). Ebenso wurde der Monat 22 aufgeteilt, und bei Fall 7 der Monat 23.

Das hier vorgeführte Verfahren kann im Prinzip auf eine beliebige Zahl von Ereignissen angewandt werden. Man muß dann natürlich entsprechend viele Ereignismonatsvariablen definieren und beim INPUT PROGRAM entsprechend viele Fälle unterscheiden. Dann empfiehlt es sich, die Anweisungen zum Aufsplitten eines Monats (Zeilen 25-39) als Makro zu definieren, um den Programmcode übersichtlich zu halten. Noch komplexer wird das Episodensplitting, wenn in einem Monat mehr als ein Ereignis auftreten kann. Dann muß ein Monat eventuell in drei oder mehr Teilepisoden aufgesplittet werden, was die Zahl der Fallunterscheidungen drastisch erhöht. In diesem Fall kann es unter Umständen einfacher sein, gleich Tagessplits zu erzeugen und analog zum Beispiel 2 vorzugehen. Allerdings wird dann die Splittingdatei enorm groß werden.

5. Ein Anwendungsbeispiel

Wir wollen zum Schluß die Fruchtbarkeit der Einbeziehung zeitveränderlicher Kovariaten an einem kleinen Beispiel demonstrieren. Wir verwenden dazu die Daten aus dem SOEP (Welle 1 bis 7), die schon vielfach für die Analyse von Arbeitslosigkeitsverläufen herangezogen wurden. Häufig wurde dabei allerdings auf die Analyse zeitveränderlicher Kovariaten verzichtet (vgl. die früheren Analysen bei ***Klein*** **1**990 und ***Ludwig-Mayerhofer*** 1992). ***Hujer und Schneider*** (1992) haben insbesondere den Einfluß von Arbeitsmarktbedingungen untersucht, d.h. monatliche Parameter zur Charakterisierung der Arbeitsmarktsituation bzw. allgemein der wirtschaftlichen Lage, aber auch Variablen zur Erfassung von Saisonalitätseffekten (Dummy-Variablen für einzelnen Monate) herangezogen. Wir verwenden im folgenden monatliche Daten der Arbeitslosenquoten im jeweiligen Bundesland.[15] Wie unsere oben dargestellten Beispiele ahnen lassen, vermuten wir außerdem Einflüsse der familiären Situation, jedenfalls bei Frauen. Aus Platzgründen, und weil es uns nur um ein kleines Demonstrationsbeispiel, nicht um eine umfassende Analyse des Arbeitslosigkeitsverlaufs geht, ist hier eine ausführliche Diskussion der Datenbasis nicht möglich; viele Angaben dazu finden sich an anderer Stelle ***(Ludwig-Mayerhofer 1992).*** Im Gegensatz zu dieser Arbeit, wo Übergänge in Voll- und Teilzeitbeschäftigung zusammengefaßt wurden, beziehen wir uns hier nur auf die Vollzeitbeschäftigung. Verwendet wird nur die jeweils erste Arbeitslosigkeitsepisode einer jeden Person. Zur Datenauswertung ziehen wir ein log-logistisches Modell heran *(**Andreß*** 1992, S. 292 f; ***Blossfeld, Hamerle und Mayer*** 1986, S. 240 ff.), welches die Zeitabhängigkeit des untersuchten Verlaufs nach den bisherigen Auswertungen am besten widerspiegelt. Die Modellschätzung erfolgte mit dem Programm TDA von *Götz Rohwer.*

---

15 ***Hujer und Schneider*** hatten einen Datensatz zur Verfügung, der sich auf tiefer gegliederte Raumordnungsregionen bezog. Solche Daten werden inzwischen vom Projektträger nicht mehr weitergegeben.

Tabelle 1: Multivariate Analysen des Übergangs Arbeitslosigkeit - Vollzeitbeschäftigung (Log-logistisches-Modell, Daten aus dem SOEP, Welle 1 bis 7)
kursiv: zeitveränderliche Kovariaten; R: Referenzkategorie bei Dummy-Variablen

	Männer		Frauen	
	Koeff.	**S.E.**	**Koeff.**	**S.E.**
Arbeitslosenquote	-0.077	0.020 **	-0.067	0.032 *
Kalendermonat (R: Juli)				
Januar	-0.356	0.236	-0.047	0.430
Februar	0.030	0.214	-0.228	0.422
März	0.491	0.214 *	0.708	0.358 *
April	0.202	0.220	0.030	0.405
Mai	-0.322	0.249	0.285	0.380
Juni	-0.254	0.249	0.131	0.386
August	0.034	0.220	0.386	0.335
September	-0.233	0.234	0.919	0.331 *
Oktober	-0.626	0.247 *	-0.114	0.379
November	-0.614	0.249 *	-0.077	0.366
Dezember	0.090	0.218	0.774	0.357 *
Kleine Kinder im Haushalt				
Alter 6 bis 24 Monate	0.082	0.162	-2.601	0.577 ***
Alter über 2 bis 10 Jahre	-0.106	0.127	-0.732	0.293 *
Alter (R: Bis 30 Jahre)				
31 bis 50 Jahre	-0.205	0.121	-0.772	0.210 ***
Über 50 Jahre	-2.137	0.202 ***	-2.543	0.458 ***
Erwerbsstatus vorher (R: Andere)				
Vollzeitbeschäftigung	0.328	0.123 **	0.184	0.177
Teilzeitbeschäftigung	-0.167	0.415	-0.491	0.261
Im Haushalt	-8.128	123.277	-1.569	0.486 **
»Stellung im Haushalt« (R: Haushaltsvorstand)				
Ehepartner	-0.159	0.304	-0.564	0.212 **
Lebensgefährte	0.326	0.219	-0.515	0.276
Kind	-0.026	0.134	-0.293	0.233
Berufsausbildung (R: Keine)				
Lehre	0.216	0.120	0.330	0.179
Fach-/Meisterschule	0.805	0.233 ***	-0.009	0.340
Universität/FH	0.458	0.189 *	0.223	0.311
Sonstige	0.258	0.193	0.197	0.252
Chronische Krankheit	-0.353	0.115 **	-0.245	0.193
Konstante	-0.986	0.288 ***	-1.416	0.435 **
Shape-Parameter	0.528	0.041 ***	0.293	0.057 ***
-2 Log-Likelihood des Null-Modells	2845.72		1801.7	
-2 Log-Likelihood des vollen Modells	2442.24		1515.7	
-2 Log-Likelihood des Modells ohne zeitveränderliche Kovariaten	2500.00		1602.7	
Zahl der Episoden	630		593	
Zahl der gesplitteten Episoden	4626		5145	
Zahl der Ereignisse	418		216	

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Die *Modellschätzungen* (vgl. Tabelle 1) zeigen, wie ergiebig die Verwendung zeitveränderlicher Kovariaten ist. Bei Männern wie Frauen ist der Einfluß der regionalen Arbeitsmarktsituation ebenso bedeutsam wie saisonale Effekte: Höhere regionale Arbeitslosenquoten bewirken eine niedrigere Hazardrate (also längere Arbeitslosigkeitsdauern), darüber hinaus zeigen sich auch explizit die saisonalen Schwankungen des Arbeitsmarkts (bessere Beschäftigungschancen im Frühjahr, zumindest bei den Männern schlechtere im Herbst). [16] Bei den Frauen spielt außerdem die familiäre Situation eine ganz entscheidende Rolle. Neben der »Stellung im Haushalt« bei Beginn der Episode (niedrigere Übergangsrate der verheirateten Frauen) [17] ist es vor allem das Vorhandensein von Kindern bis zu 24 Monaten, aber auch noch bis zum Alter von 10 Jahren, das eine ganz entscheidende Verlängerung der Dauer bis zu einem Übergang in Vollzeitbeschäftigung bewirkt. Die Bedeutung der zeitveränderlichen Kovariaten wird deutlich, wenn man jeweils ein Modell ohne diese berechnet. Die Erklärungskraft des multivariaten Modells nimmt bei Männern wie Frauen in signifikanter Weise ab, bei den Frauen - wohl wegen des starken Effekts der Kinder - allerdings erheblich mehr.

Literatur

Andreß, Hans-Jürgen, 1992:
Verlaufsdatenanalyse (Historical Social Research/Historische Sozialforschung, Supplement/Beiheft No. 5). Köln: Zentrum für Historische Sozialforschung.

Blossfeld, Hans-Peter; Hamerle, Alfred, 1989:
Using Cox Models to Study Multiepisode Processes.
In: Sociological Methods & Research 17, S. 432-448.

Blossfeld, Hans-Peter; Hamerle, Alfred; Mayer, Karl-Ulrich, 1986:
Ereignisanalyse. Frankfurt/New York: Campus.

Blossfeld, Hans-Peter; Huinink, Johannes, 1991:
Human Capital Investments or Norms of Role Transition? How Women's Schooling and Career Affects the Process of Family Formation.
In: American Journal of Sociology 97, S. 143-168.

Brüderl, Josef; Klan, Thomas, 1993:
Bildung und Familiengründungsprozeß deutscher Frauen: Humankapital- und Institutioneneffekt.
In: *Andreas Diekmann und Stefan Weick,* (Hrsg.), Der Familienzyklus als sozialer Prozeß. Bevölkerungssoziologische Untersuchungen mit den Methoden der Ereignisanalyse. Berlin: Duncker & Humblot, S. 194-215.

Brüderl, Josef; Preisendörfer, Peter; Ziegler, Rolf, 1993:
Upward Mobility in Organizations: The Effects of Hierarchy and Opportunity Structure.
In: European Sociological Review 9, S. 173-188.

16 Die im Vergleich zum Juli nicht niedrigere Abgangsrate im Dezember bei den Männern und die sogar signifikant erhöhte Rate in diesem Monat bei den Frauen stellt dagegen wahrscheinlich ein Erhebungsartefakt dar.

17 Personen, die den Haushalt - und damit in aller Regel auch die Stellung im Haushalt - wechselten, wurden mit einer eigenen Dummy-Variable gekennzeichnet, da bei diesen Personen auch die Kinder-Daten nicht exakt zuordenbar gewesen wären. Außerdem waren fehlende Werte in den Variablen Ausbildung und Krankheit durch Dummy-Variablen gekennzeichnet. Da alle diese Dummy-Variablen nicht signifikant von Null verschieden waren, und weil sie nicht von inhaltlichem Interesse sind, wurden sie in Tabelle 1 nicht ausgewiesen.

Cox, D. R., 1972:
Regression Models and Life-Tables (with Discussion).
In: J. Roy. Statist. Soc., Series B 34, S. 187-220.

Diekmann, Andreas, 1988:
Ereignisdatenanalyse - Beispiele, Probleme und Perspektiven.
In: ZUMA-Nachrichten 23, S. 7-25.

Diekmann, Andreas; Mitter, Peter, 1984:
Methoden zur Analyse von Zeitverläufen. Stuttgart: Teubner.

Greene, William E. 1992:
LIMDEP, Version 6.0. New York: Econometric Software.

Hamerle, Alfred; Tutz, Gerhard, 1989:
Diskrete Modelle zur Analyse von Verweildauern und Überlebenszeiten. Frankfurt/New York: Campus.

Hujer, Reinhard; Schneider, Hilmar, 1986:
Semi-parametrische und parametrische Ratenmodelle. Frankfurt/Mannheim: Sonderforschungsbereich 3, Arbeitspapier Nr. 200.

Hujer, Reinhard; Schneider, Hilmar, 1992:
Strukturelle und institutionelle Determinanten der Arbeitslosigkeit aus mikroanalytischer Sicht.
In: ***Reinhard Hujer, Hilmar Schneider und Wolfgang Zapf*** (Hrsg.), Herausforderungen an den Wohlfahrtsstaat im strukturellen Wandel. Frankfurt/New York: Campus, S. 315-341.

Kalbfleisch, J. D.; Prentice, R. L., 1980:
The Statistical Analysis of Failure Time Data. New York: Wiley.

Klein, Thomas, 1990:
Arbeitslosigkeit und Wiederbeschäftigung im Erwerbsverlauf.
In: Kölner Zeitschrift für Soziologie und Sozialpsychologie 42. S. 688-705.

Ludwig-Mayerhofer, Wolfgang, 1990:
Arbeitslosigkeit im Erwerbsverlauf.
In: Zeitschrift für Soziologie 19, S. 345-359.

Ludwig-Mayerhofer, Wolfgang, 1992:
Fakt und Artefakt in der Analyse von Arbeitslosigkeitsverläufen.
In: Kölner Zeitschrift für Soziologie und Sozialpsychologie 44, S. 124-133.

Ludwig-Mayerhofer, Wolfgang, 1994:
Statistische Modellierung von Verlaufsdaten in der Analyse sozialer Probleme.
In: Soziale Probleme 5 (im Erscheinen)

Petersen, Trond, 1986:
Estimating Fully Parametric Hazard Rate Models with Time-dependent Covariates.
In: Sociological Methods & Research 14, S. 219-246.

Petersen, Trond; Koput, Kennah W., 1992:
Time-Aggregation Bias in Hazard-Rate Models With Covariates.
In: Sociological Methods & Research 21, S. 25-51.

Projektgruppe "Das Sozio-ökonomische Panel", 1990:
Das Sozio-ökonomische Panel für die Bundesrepublik Deutschland nach fünf Wellen.
In: Vierteljahreshefte für Wirtschaftsforschung, S. 141-151.

Rohwer, Götz, 1993:
TDA Working Papers. Bremen: Ms.

Schneider, Hilmar, 1991:
Verweildaueranalyse mit GAUSS. Frankfurt/New York: Campus,

Tölke, Angelika, 1989:
Lebensverläufe von Frauen. Familiäre Ereignisse, Ausbildungs- und Erwerbsverhalten. München: DJI.

Tuma, Nancy B., 1980:
Invoking RATE. Menlo Park: SRI International.

Tutz, Gerhard; Georg, Werner, 1991:
Diskrete Hazardraten- Modelle in der Shell-Jugendstudie 1985: Eine Anwendung des Programms GLAMOUR.
In: ZA-lnformation 29. S. 81-93.

Voges, Wolfgang; Zwick, Michael, 1991:
Die Bremer Stichprobe von Sozialhilfeakten: Möglichkeiten für die empirische Sozialforschung.
In: Zeitschrift für Soziologie 20, S. 78-81.

Wolf, Christof, 1993:
Egozentrierte Netzwerke: Datenorganisation und Datenanalyse.
In: ZA-lnformation 32, S. 72-94.

Yamaguchi, Kazuo, 1991:
Event History Analysis. Newbury Park: Sage.

Anhang

An dieser Stelle sei kurz auf die statistischen Grundlagen des Episodensplittings eingegangen. Betrachten wir eine Episode mit Beginnzeitpunkt 0 und Endzeitpunkt T und einer zeitabhängigen Kovariate x(u), wobei u die Prozeßzeit mißt. Für eine Episode i, die mit Zensierung endet, lautet der Log-Likelihood-Beitrag [18]:

$$\ln(L_i) = -\int_0^T r_i\,[u \mid x(u)]du. \qquad (1)$$

r ist die Ratenfunktion, die von der Zeit u abhängt und von der zeitabhängigen Kovariate x(u) beeinflußt wird. Nun gibt es zwei Möglichkeiten. Wenn x(u) eine bekannte Funktion der Prozeßzeit ist, kann eventuell das Integral gelöst und die Likelihood programmiert werden. Wenn keine funktionale Beziehung für x(u), sondern nur die Zeitpunkte, zu denen sich x ändert, bekannt sind - und dies ist der häufigere Fall -, muß folgendermaßen vorgegangen werden. Nehmen wir an, daß x sich zu t_1 von x_0 zu x_1 verändert. Dann lautet der Likelihood-Beitrag von Episode i:

$$\ln(L_i) = -\left\{\int_0^{t_1} r_i[u \mid x_0]du + \int_{t_1}^T r_i[u \mid x_1]du\right\}. \qquad (2)$$

Man sieht, daß das Integral in zwei Teile aufgespalten ist, und zwar so, daß unter jedem Integral die Kovariate konstant ist. Um diese Summe von Integralen nicht extra programmieren zu müssen, wendet man nun den Trick des Episodensplittings an, denn die Likelihood-Beiträge aller Teilepisoden werden zur Errechnung der Gesamt-Likelihood aufsummiert. Damit ist die Gesamt-Likelihood aller Teilepisoden identisch zur Gesamt-Likelihood, die sich aus der Summierung der Likelihood-Beiträge nach Formel (2) ergäbe. Dies hat den Vorteil, daß die üblichen Formeln zur Berechnung der Likelihood mit konstanten Kovariaten verwendet werden können (mit dem einzigen Unterschied, daß berücksichtigt werden muß, daß die untere Integrationsgrenze nun nicht mehr notwendigerweise 0 ist). Man erkennt auch, daß das Episodensplitting zwar die Fallzahl vergrößert, aber die Gesamt-Likelihood sich nicht verändert. Deshalb führt Episodensplitting *nicht* zu einer Verkleinerung der Standardfehler der Likelihood-Schätzer. Dementsprechend können Episoden so oft gesplittet werden, wie dies erforderlich ist, so daß ohne weiteres auch mehrere Wechsel einer Kovariaten (z.B. Heirat, Scheidung, erneute Heirat) und mehrere zeitveränderliche Kovariaten berücksichtigt werden können.

18 Für eine Episode, die mit einem Ereignis endet, kommt noch ein weiterer Term hinzu. Die Argumentation ist aber völlig analog zu dem Fall mit Zensierung.