

Startkohorte 4: Klasse 9 (SC4)
Wellen 3 und 4 (SUF SC4 4.0.0)
Technical Report: Edition und Korrek-
tur der Lebensverlaufsdaten

Ralf Künster und Michael Ruland



Urheberrechtlich geschütztes Material
Leibniz-Institut für Bildungsverläufe e.V. (LifBi)
Wilhelmsplatz 3, 96047 Bamberg
Direktor: Prof. Dr. Hans-Günther Roßbach
Wissenschaftlich-kordinierende Geschäftsführerin: Dr. Jutta von Maurice
Kaufmännischer Geschäftsführer: Dr. Robert Polgar
Bamberg, 2014

Startkohorte 4: Klasse 9

Wellen 3 und 4

Edition und Korrektur der Lebensver-
laufdaten des NEPS SUF SC4: 4.0.0 (Re-
lease vom 30.06.2014)

Ralf Künstler und Michael Ruland

**Startkohorte 4 des Nationalen Bildungspanels
SUF SC4: 4.0.0 (Release vom 30.06.2014):
Edition und Korrektur der Lebensverlaufsdaten**

Technical Report

Ralf Künster und Michael Ruland

Nationales Bildungspanel, Etappe 6
Wissenschaftszentrum Berlin

V 1.0 19.11.2014

Einleitung

Mit Hilfe eines von den Autoren bereitgestellten do-Files werden die Lebensverlaufsdaten des aktuellen Releases der 3. und 4. Erhebungswelle der NEPS-Startkohorte 4 (SC4 4.0.0) nachträglich ediert. Die einzelnen Editions Schritte sind in diesem Dokument ausführlich dokumentiert. Es wird dringend empfohlen, diese Edition durchzuführen. Dabei werden keine weitreichenden inhaltlichen Editionen der Daten vorgenommen, sondern ausschließlich a) Korrekturen vorgenommen und b) Editions Schritte angewendet, die so auch in Startkohorte 6 zum Einsatz gekommen sind.

Betroffen sind folgende Bereiche:

- Anwendung von Editions routinen, die bei vergleichbaren Datenstrukturen der Startkohorte 6 zum Einsatz kamen (Beispiele: Edition von einmonatigen Episodenüberschneidungen sowie von ein- bis zweimonatigen Episodenlücken, Einfügung von Editions lückenepisoden bei Episodenlücken von mehr als zwei Monaten)
- Korrektur von Editions regeln (Beispiel: vollständige Löschung von Episoden, die in Erhebungswelle 3 erstmalig erfasst wurden und zum damaligen Interviewzeitpunkt andauerten, denen jedoch im Prüfmodul der Erhebungswelle 4 widersprochen wurde)
- Nachträgliche Korrektur von Problemen, die durch ein fehlerhaftes Erhebungsinstrument entstanden sind (Beispiel: fehlende Edition von in Erhebungswelle 4 erfassten Episoden, deren Startdatum in den Zeitbereich vor dem Interviewzeitpunkt von Welle 3 rückdatiert worden sind)

Nachfolgend werden die im do-File umgesetzten Editionsmaßnahmen einzeln und detailliert beschrieben. Die einzig erforderliche Anpassung, die Datennutzer zur Ausführung des do-Files zwingend vornehmen müssen, ist die Angabe von Laufwerk und Pfad zu den SUF-Daten hinter der Makro-Deklaration "global pfad..." in Zeile 14 des do-Files. Zur besseren Übersichtlichkeit ist das do-File in zehn nummerierte Funktionsblöcke aufgeteilt, in denen spezifische Teilaufgaben der Datenedition durchgeführt werden und auf die in den nachfolgenden Erläuterungen Bezug genommen wird. Die Einzelschritte sind im do-File zusätzlich kommentiert (siehe Anhang). Die Reihenfolge der Ausführung der Funktionsblöcke ist nicht beliebig und sollte daher nicht verändert werden.

Nach Ausführung des do-Files enthält das im Makro "pfad" angegebene Verzeichnis neben den ursprünglichen und unveränderten SUF-Files für die Längsschnittmodule und das Biografie-File auch die korrigierten Kopien dieser Dateien. Die Kopien haben den gleichen Namen wie die Original-Files, jeweils ergänzt um das Kürzel "V2".

Funktionsblock 1: Korrektur von in Welle 3 im Prüfmodul fälschlicherweise als andauernd markierten Lücken- und Elternzeitepisoden

In Welle 3 wurden in einigen wenigen Einzelfällen Lücken- und Elternzeitepisoden im Prüfmodul als "zum Zeitpunkt des Interviews andauernd" gekennzeichnet, obwohl diese Episoden in den Längsschnittmodulen als "nicht andauernd" markiert sind. Die Enddatumsangaben enden darüber hinaus deutlich vor dem Interviewzeitpunkt und es wurden im Prüfmodul keine Enddatumskorrekturen vorgenommen. Dieses Problem beruht auf einer - für spätere Wellen nunmehr behobenen - Fehlfunktion des Prüfmoduls. Eine zentrale Folge dieser Markierung der Episoden als "andauernd" war, dass diese Episoden als sogenannte "Aufsetzepisoden" mit Preload-Informationen in die nachfolgende Erhebungswelle 4 einbezogen wurden. Überraschenderweise widersprachen die Zielpersonen den Angaben dieser Episoden in der 4. Welle nicht, sondern sie verlängerten diese Episoden in mehr oder weniger umfangreichen Ausmaß. Da jedoch auch alle inhaltlichen Informationen zu diesen Episoden eindeutig darauf hinweisen, dass diese bereits in Welle 3 abgeschlossen waren, werden im Funktionsblock 1 des do-Files alle entsprechenden Subspells (Fortführungen der Episoden in der Welle 4) aus den Längsschnittfiles gelöscht, die Informationen zu den korrekten Enddatumsangaben übertragen und die Variable zur Andauer der Episoden auf "dauert nicht an" gesetzt. Im Biografie-File werden die Angaben zu Erhebungswelle, Enddatum und Andauer der Episoden analog korrigiert.

Funktionsblock 2: Korrektur der Markierungsvariablen für andauernde Episoden bei in den Längsschnittmodulen widersprochenen Episoden sowie Korrektur von Episoden, die in Welle 4 neu erfasst, deren Datumsangaben jedoch auf einen Zeitpunkt vor dem Interviewdatum der Welle 3 rückdatiert wurden

In diesem Funktionsblock werden zwei Probleme parallel bearbeitet. Das erste Problem resultiert aus einer unvollständigen Anwendung von Editionsregeln in Bezug auf die Variable "Episode dauert an". Die zu Grunde liegende Editionsregel besagt zum einen, dass Episoden, die zum Interviewzeitpunkt der Vorwelle andauerten (sogenannte Aufsetzepisoden) und denen in der nächsten Erhebungswelle im Längsschnittmodul widersprochen wird, nur mit dem zeitlichen Anteil bis zum Interviewzeitpunkt der Vorwelle in die Daten aufgenommen werden. Hier wird dem Grundsatz gefolgt, dass die Befragten zum jeweiligen Interviewzeitpunkt am besten über ihre aktuelle Situation Bescheid wissen und sich der Widerspruch in der Folgewelle daher nur auf die Zeit seit dem letzten Interview bezieht, nicht jedoch auf die Zeit davor. Zum anderen beinhaltet die Editionsregel eine Korrektur der Markierung andauernder Episoden, da diese Episoden zwar zum Zeitpunkt der Vorwelle angedauert haben, sie in der aktuellen Welle aber natürlich nicht als andauernd gelten. Dieser zweite Teil der Edition wurde im aktuellen SUF der SC4 nicht umgesetzt. Im Rahmen der nachträglichen Edition werden sowohl in den Längsschnittfiles die jeweiligen Variablen zur Markierung andauernder Episoden (im Erwerbstätigkeitsmodul beispielsweise die Variable *ts1312c_g1*) auf "nicht andauernd" gesetzt als auch die entsprechende Variable *splast* im Biografie-File angepasst.

Das zweite Problem basiert auf einem Instrumentenfehler sowohl in den Längsschnittmodulen als auch im Prüfmodul. In der 4. Welle war es in den Längsschnittmodulen möglich, neue Episoden zu erfassen, deren Start- als auch Enddatum vor dem Interviewzeitpunkt der 3. Welle lag. Dies durfte eigentlich nicht möglich sein, da im Rahmen der Lebensverlaufserhebung im Panelinstrument nur Episoden berichtet werden sollten, die seit dem letzten Interview stattgefunden haben. Neben inhaltlichen Argumenten, die für eine Edition dieser Fälle sprechen, spielt hier die zeitliche Konsistenzprüfung im Rahmen der Lebensverlaufserhebung (Prüfmodul) eine entscheidende Rolle. Diese Prüfung bezieht sich nur auf den Zeitraum seit dem letzten Interview und eine Korrektur von Angaben vor dem letzten Interview würde die damalige Konsistenzprüfung erneut in Frage stellen. Da diese Vordatierung jedoch im Rahmen der 4. Erhebung möglich war, sowohl bei neu aufgenommenen Episoden als auch bei Korrekturen im Prüfmodul, ist eine Edition dieser Episoden notwendig. Durch das do-File werden alle rückdatierten Startdatumsangaben der Episoden in den editierten Datumsangaben der Längsschnittdateien und im Biografie-File einheitlich auf das Interviewdatum der letzten Erhebungswelle gesetzt, sofern das Enddatum der Episoden nicht ebenfalls vor das Interviewdatum von Welle 3 datiert wurde. Eine zusätzliche Variable in den Längsschnittdateien zeigt an, dass diese Episode editiert wurde. Wenn sowohl Start- als auch Enddatum einer in Welle 4 neu erfassten Episode vor das Interviewdatum von Welle 3 datiert wurden, dann erhalten die editierten Datumsangaben im Längsschnittmodul den Wert -55 "nicht ermittelbar" und die entsprechende Episode wird aus dem Biografie-File gelöscht.

Funktionsblock 3: Wiederherstellung der in der 3. Welle erstmalig erfassten andauernden Episoden (Aufsetzepisoden), denen in der 4. Welle im Prüfmodul widersprochen wurde

Episoden, die in der 3. Welle erstmalig erfasst wurden und dort als "andauernd" markiert waren (Aufsetzepisoden), wurden aus dem Biografie-File des SUF vollständig gelöscht, wenn die Zielpersonen ihnen im Prüfmodul der 4. Welle widersprochen haben. Diese vollständige und somit wellenübergreifende Löschung der Episoden wird mit Hilfe des do-Files an die bereits im Funktionsblock 2 erwähnte Editionsregel angepasst, wonach bei Widerspruch eine Schließung der Episoden zum vorigen Interviewzeitpunkt vorgesehen ist. Daher gilt für im Prüfmodul widersprochenen Aufsetzepisoden (wie auch bei Aufsetzepisoden, denen in den Längsschnittmodulen direkt widersprochen wird), dass das Enddatum dieser Episoden auf den Interviewzeitpunkt der letzten Erhebungswelle festgesetzt wird und die Episode daher auch im Biografie-File mit geändertem Enddatum erhalten bleibt. Der Variable *spms* des Biografie-Files wird für diese Episoden weiterhin der Code -20 "Episode im Prüfmodul annulliert" zugewiesen.

Funktionsblock 4: Beseitigung von einmonatigen Überschneidungen zwischen Hauptereignisepisodes

Unter einmonatiger Überschneidung zweier aufeinanderfolgender Episoden wird verstanden, dass eine vorangehende Episode innerhalb eines Lebenslaufs im gleichen Monat endet,

in dem eine nachfolgende Episode beginnt. Da die Episodendaten monatsgenau erfasst werden, ergibt sich rein rechnerisch eine Überschneidung von einem Monat, obwohl in den meisten Fällen die Episoden ‚in Wahrheit‘ unmittelbar aneinander anschließen, also nacheinander durchlebt wurden. Die Überschneidung ist daher eher der Ungenauigkeit der Erfassung der Episodenzeiten und der Toleranz des Erhebungsinstrumentes zuzuschreiben. Für die spätere Analyse der Längsschnittdaten sind diese einmonatigen Überschneidungen jedoch problematisch, da sich dadurch Parallelitäten in einem Monat ergeben, die in den meisten Fällen nicht wirklich vorhanden sind. Daher wurde festgelegt, diese einmonatigen Überschneidungen zu editieren.

Die Edition einmonatiger Überschneidungen erfolgt nur für Episoden, die von den Befragten entweder als Hauptereignis deklariert wurden (*spms* = 1 oder *spms* = -20), oder für Episoden, die als Nebenereignis gekennzeichnet sind, jedoch bereits während des Interviews im Prüfmodul von den befragten Personen als Ereignisse definiert wurden, die Zeiträume überbrücken, in denen kein Hauptereignis genannt wurde. Da diese Lückenzeiträume überbrückenden Nebenereignisse im SUF nicht mehr identifizierbar sind, werden im hier beschriebenen do-File nur Hauptereignisse in die Edition der einmonatigen Überschneidungen einbezogen.

Die Editionsregeln für einmonatige Überschneidungen besagen, dass die vorangehende Episode im Enddatum um einen Monat gekürzt und dadurch die Überschneidung beseitigt wird. Ist jedoch die vorangehende Episode insgesamt nur von einem Monat Dauer, dann wird die folgende Episode im Startdatum um einen Monat gekürzt. Sind beide Episoden nur einen Monat lang, bleiben beide Episoden erhalten. Gibt es eine Episode, die wegen einer entsprechenden Konstellation der führenden und folgenden Episoden sowohl im Start- als auch im Enddatum gekürzt werden muss, dann sollte diese Episode mindestens drei Monate andauern. Ist die Episode jedoch nur zwei Monate lang, so wird sie nur im Enddatum um einen Monat gekürzt.

Funktionsblock 5: Schließen von ein- bis zweimonatigen Lücken im Lebenslauf

Kurze Lücken von ein bis zwei Monaten im berichteten Lebenslauf sind in der Regel dem mangelnden Erinnerungsvermögen der Befragten für exakte Datumsangaben geschuldet. Da die Episoden im Erhebungsinstrument thematisch modular, also nicht entlang einer Zeitachse, erfasst werden, bietet das Instrument den Befragten nur wenig Unterstützung bei der exakten Datierung von aufeinanderfolgenden Episoden. Daher werden mit dem do-File ein- bis zweimonatigen Lücken im SUF der SC4 nachträglich editiert.

Die Editionsregeln besagen hier, dass bei einer Lücke von einem Monat das Enddatum der vorangehenden Episode verlängert wird. Bei einer zweimonatigen Lücke wird zusätzlich das Startdatum der nachfolgenden Episode um einen Monat vorverlegt.

Funktionsblock 6: Schließen von Lücken mit mehr als zwei Monaten

Das Vorkommen von längeren Lücken in den per Prüfmodul getesteten Lebensläufen ist äußerst selten und in der Regel darauf zurückzuführen, dass das Prüfmodul vom Interviewer vor Abschluss der Durchführung aller Prüfungen verlassen wurde. Zusätzlich kommt in Welle 4 der SC4 eine weitere Ursache für längere Lücken im Lebenslauf hinzu. In dieser Welle wurde für diejenigen Befragten, die zwischen Welle 2 und Welle 3 die Schule verlassen haben, die Schulgeschichte retrospektiv erfasst. Aus Gründen der zeitlichen Effizienz wurde bei der retrospektiven Erfassung der Schulgeschichte auf den Einsatz eines Prüfmoduls verzichtet. Dies hat zur Folge, dass in der Schulgeschichte vermehrt nicht berichtete Zeiträume vorzufinden sind.

Längere Lücken werden mit Hilfe des do-Files im SUF der SC4 derart editiert, dass für diese Zeiträume im Biografie-File neue Episoden vom Typ "Dateneditionslücke" eingefügt werden. Diese schließen die vorhandenen Lücken vollständig. Die Variable *wave* erhält den Wert -55 "nicht ermittelbar".

Funktionsblock 7: Schließen von Lücken zwischen dem 8. Geburtstag der Zielperson und dem Beginn der frühesten erfassten Episode

Wie bereits in den Erläuterungen zu Funktionsblock 6 erwähnt, wurde die Schulgeschichte der Zielpersonen nicht in Welle 3 mit Beginn der Erfassung der von den Befragten berichteten Lebensläufe erhoben, sondern erst in Welle 4 als separate retrospektive Biografie ohne Konsistenzprüfung mittels Prüfmodul. Dies hat einerseits den Nachteil, dass nur rund zwei Drittel der Befragten aus Welle 3 auch an Welle 4 teilgenommen haben und nur für diese Gruppe eine retrospektive Schulgeschichte vorliegt. Für ein Drittel der Befragten sind keine Daten zur Schulgeschichte vorhanden, sondern nur Angaben zum Status beim Verlassen der allgemeinbildenden Schule. Andererseits hat das Fehlen des Prüfmoduls vermehrt zu nicht berichteten Zeiträumen vor der Angabe zur ersten Schulepisode geführt. Da jedoch der Lebensverlauf der Befragten zumindest ab ihrem 8. Geburtstag dargestellt sein soll, wird durch den do-File im Falle einer Lücke für den Zeitraum zwischen dem 8. Geburtstag und dem Beginn der frühesten erfassten Episode eine neue Episode vom Typ "Dateneditionslücke" in den Biografie-File eingefügt. Für diese spezielle Editions-lückenvariante, die pro Fall nur ein einziges Mal vorhanden sein kann, wird der Wert der Variablen *splink* auf 990000 gesetzt. Die Variable *wave* erhält den Wert -55 "nicht ermittelbar".

Funktionsblock 8: Schließen von Lücken zwischen letzter Episode und Interviewzeitpunkt

Lücken zwischen dem Ende der letzten Episode eines Lebenslaufs und dem Interviewzeitpunkt der aktuellen Welle treten extrem selten auf. Nur in einem Fall des aktuellen SUF der SC4 konnte ein echter nicht-berichteter Zeitraum entdeckt werden. Dieser wird durch den do-File in gleicher Weise wie lange nicht berichtete Zeiträume (siehe Funktionsblock 6) geschlossen. Hinzu kommen 11 Fälle, bei denen eine Lücke zwischen letzter Episode und Interviewzeitpunkt dadurch entsteht, dass das Interview nach Erfassung des Lebenslaufs für ein

bis zwei Monate unterbrochen und erst nach dieser Zeit durch eine erneute Kontaktaufnahme mit der befragten Person vervollständigt wurde. Das Interviewdatum wurde dadurch auf das Datum des letzten Kontakttermins gesetzt und ist deshalb nicht mehr identisch mit dem Enddatum der letzten Episode. Das do-File generiert auch für diese Zeiträume zusätzliche Editions-lückenepisoden, die jedoch in der Variable *sptype* den Code 98 "Editionslücke: Interviewunterbrechung nach LV-Erfassung" erhielten.

Funktionsblock 9: Einfügen einer Editions-lücke für Fälle ohne Episodendaten

Aufgrund einer sehr spezifischen Abfolge von Schulepisoden wurde in der 3. Welle für einen Fall (4014657) kein retrospektiver Lebenslauf erfasst. Dieser Fall hat später nicht an der Erhebungswelle 4 teilgenommen, sodass für den Fall weder in den Längsschnittmodulen noch im Biografie-File ein Eintrag vorhanden ist. Der do-File generiert für diesen Fall eine Editions-lückenepisode im Biografie-File, die vom 8. Geburtstag der Person bis zum Interviewdatum der 3. Welle reicht. Die Belegung der Variablen folgt den in Funktionsblock 7 beschriebenen Regeln.

Funktionsblock 10: Zusammenfügen der korrigierten Biografie-Informationen

In diesem Funktionsblock werden alle neu generierten Editions-lückenepisoden in den Biografie-File integriert und neu vergebene Codes mit Value Labels versehen. Temporär angelegte Zwischenergebnisdateien werden gelöscht.

Literatur:

Hess, Doris; Steinwede, Angelika; Schneider, Bernd (2012). Erhebung von retrospektiven Längsschnittdaten – Prüfmodul. *Dokumentation 04/2012*, Infas, Bonn.

Matthes, Britta; Drasch, Katrin; Erhardt, Klaudia; Künstler, Ralf; Valentin, Margot-Anna (2012). Arbeiten und Lernen im Wandel. Teil IV: Editionsbericht. *FDZ-Methodenreport 03/2012*, IAB, Nürnberg

Matthes, Britta; Reimer, Meike; Künstler, Ralf (2007). Techniken und Werkzeuge zur Unterstützung der Erinnerungsarbeit bei der computergestützten Erhebung retrospektiver Längsschnittdaten. *Methoden, Daten und Analysen. Zeitschrift für empirische Sozialforschung*, 1(1), 69-92.

Anhang: Syntax do-File "kor_SUF_SC4_D_4-0-0.do"

```
* Ralf Künster und Michael Ruland, Wissenschaftszentrum Berlin,  
* Reichpietschufer 50, 10785 Berlin  
* Tel: 030/25491-175, Email: ralf.kuenster@wzb.eu  
* Tel: 030/25491-151, Email: michael.ruland@wzb.eu  
* do-File zur Edition und Korrektur des NEPS-SUF SC4: 4.0.0 (Release 30.06.2014)  
* Version vom 19.11.2014  
* Ziel des do-File ist es, Korrekturen und nachträgliche Editionsschritte in den  
* Längsschnittmodulen und im Biografie-File des oben erwähnten SUF vorzunehmen.  
* Mit dem nächsten Update des SUF der SC4 werden diese Maßnahmen vom Forschungs-  
* datenzentrum (FDZ-LIfBi) in den Daten implementiert.
```

```
version 13  
set more off
```

```
* Pfadangabe des Datenverzeichnisses (vom Nutzer anzupassen!):  
global pfad "M:\group\BAL\NEPS\Erhebungsdaten_intern\SUF SC4_D_4-0-  
0_20140630\Zeitkorrektur\"
```

```
* Funktionsblock 1:  
* Einzelfallkorrektur: Lückenepisoden wurden im Prüfmodul falsch als andauernd  
* markiert(Fall 4022151/300002, 4026436/300001, 4026867/300001) und eine  
* Erziehungszeitepisode wurde im Prüfmodul falsch als andauernd markiert  
*(Fall 4024162/291001).
```

```
use "${pfad}SC4_spGap_D_4-0-0.dta", clear  
drop if ID_t == 4022151 & splink == 300002 & subspell > 0  
drop if ID_t == 4026436 & splink == 300001 & subspell > 0  
drop if ID_t == 4026867 & splink == 300001 & subspell > 0  
replace wave = 3 if ID_t == 4022151 & splink == 300002  
replace spgen = 0 if ID_t == 4022151 & splink == 300002  
replace spext = 0 if ID_t == 4022151 & splink == 300002  
replace spstat = 10 if ID_t == 4022151 & splink == 300002  
replace ts3512m = 2 if ID_t == 4022151 & splink == 300002  
replace ts3512y = 2011 if ID_t == 4022151 & splink == 300002  
replace ts3512c = . if ID_t == 4022151 & splink == 300002  
replace ts3512m_g1 = 2 if ID_t == 4022151 & splink == 300002  
replace ts3512y_g1 = 2011 if ID_t == 4022151 & splink == 300002  
replace ts3512c_g1 = 2 if ID_t == 4022151 & splink == 300002  
replace wave = 3 if ID_t == 4026436 & splink == 300001  
replace spgen = 0 if ID_t == 4026436 & splink == 300001  
replace spext = 0 if ID_t == 4026436 & splink == 300001  
replace spstat = 10 if ID_t == 4026436 & splink == 300001  
replace ts3512m = 9 if ID_t == 4026436 & splink == 300001  
replace ts3512m_g1 = 9 if ID_t == 4026436 & splink == 300001  
replace wave = 3 if ID_t == 4026867 & splink == 300001  
replace spgen = 0 if ID_t == 4026867 & splink == 300001  
replace spext = 0 if ID_t == 4026867 & splink == 300001  
replace spstat = 10 if ID_t == 4026867 & splink == 300001  
replace ts3512m = 8 if ID_t == 4026867 & splink == 300001  
replace ts3512y = 2011 if ID_t == 4026867 & splink == 300001  
replace ts3512m_g1 = 8 if ID_t == 4026867 & splink == 300001  
save "${pfad}SC4_spGap_V2_D_4-0-0.dta", replace
```

```
use "${pfad}SC4_spParLeave_D_4-0-0.dta", clear  
replace ts2712c_g1 = 2 if ID_t == 4024162 & splink == 291001  
save "${pfad}SC4_spParLeave_V2_D_4-0-0.dta", replace
```

```
use "${pfad}SC4_Biography_D_4-0-0.dta", clear  
replace wave = 3 if ID_t == 4022151 & splink == 300002
```

```

replace splast = 2 if ID_t == 4022151 & splink == 300002
replace endm = 2 if ID_t == 4022151 & splink == 300002
replace endy = 2011 if ID_t == 4022151 & splink == 300002
replace wave = 3 if ID_t == 4026436 & splink == 300001
replace endm = 9 if ID_t == 4026436 & splink == 300001
replace wave = 3 if ID_t == 4026867 & splink == 300001
replace endm = 8 if ID_t == 4026867 & splink == 300001
replace splast = 2 if ID_t == 4024162 & splink == 291001
replace endm = 11 if ID_t == 4013892 & splink == 260002
replace endm = 11 if ID_t == 4019151 & splink == 230002
replace endm = 11 if ID_t == 4023095 & splink == 230002
save "${pfad}SC4_Biography_V2_D_4-0-0.dta", replace

* Funktionsblock 2:
* Korrektur der "dauert an"-Variable bei in den Längsschnittmodulen wider-
* sprprochenen Episoden von "dauert an" auf "dauert nicht an" und Korrektur von
* in der Welle 4 vor das Ende von Welle 3 rückdatierten Episoden durch
* Recodierung der editierten Zeitangaben in den Längsschnittmodulen.
* Einfügen einer Markierungsvariablen, die anzeigt, welche Episoden recodiert
* wurden und Recodierung der entsprechenden Datumsangaben im Biografie-File bzw.
* Löschen der Episode, falls auch das Ende der Episode vor dem Interviewzeit-
* punkt der letzten Erhebungswelle liegt.
use "${pfad}SC4_TargetMethods_D_4-0-0.dta", clear

* Ermittlung des Interviewdatums aus Welle 3.
keep if wave == 3
keep ID_t intm inty
save "${pfad}intdat_w3", replace

* Bearbeitung der Längsschnitt-Files.
local datstam "23 35 21 27 11 25 13 15"
local modul "Emp Gap Military ParLeave School Unemp VocPrep VocTrain"
local typ "etmod lumod wmod kieumod asmod almod bvmod abmod"
forvalues i = 1(1) 8 {
    local da : word `i' of `datstam'
    local mo : word `i' of `modul'
    local ty : word `i' of `typ'
    use "${pfad}SC4_sp`mo'_D_4-0-0.dta", clear
    if `i' != 5 {
        replace ts`da'12c_g1 = 2 if disagint == 1
    }
    if `i' == 5 {
        replace tf`da'12c_g1 = 2 if disagint == 1
    }
    if `i' == 2 {
        bysort ID_t splink: gen mod = _N
        gen `ty' = 4
        replace `ty' = 3 if mod > 1
        drop mod
    }
}
merge m:1 ID_t using "${pfad}intdat_w3"
drop if splink == .
gen start = ((ts`da'11y_g1 - 1960) * 12) + ts`da'11m_g1
gen end = ((ts`da'12y_g1 - 1960) * 12) + ts`da'12m_g1
gen intw3 = ((inty - 1960) * 12) + intm
gen ts`da'11x_g1 = 1 if start > 0 & start < intw3 & `ty' != 3 & wave == 4
replace ts`da'11x_g1 = 2 if end > 0 & end < intw3 & `ty' != 3 & wave == 4
replace ts`da'11m_g1 = intm if ts`da'11x_g1 == 1
replace ts`da'11y_g1 = inty if ts`da'11x_g1 == 1
replace ts`da'11m_g1 = -55 if ts`da'11x_g1 == 2

```

```

replace ts`da`lly_g1 = -55 if ts`da`llx_g1 == 2
replace ts`da`l2m_g1 = -55 if ts`da`llx_g1 == 2
replace ts`da`l2y_g1 = -55 if ts`da`llx_g1 == 2
label variable ts`da`llx_g1 "Episoden, die in der 4.Welle neu erfasst und vor
das Ende von Welle 3 datiert wurden."
label define vordat 1 "In Welle 4 im Startdatum vordatierte" 2 "In Welle 4 in
Start- und Enddatum vordatiert"
label values ts`da`llx_g1 vordat
drop intm inty intw3 start end _merge
save "${pfad}SC4_sp`mo`_V2_D_4-0-0.dta", replace
rename ts`da`llm_g1 sm
rename ts`da`lly_g1 sy
rename ts`da`llx_g1 sx
keep if disagint == 1 | sx == 1 | sx == 2
keep ID_t splink disagint sm sy sx
save "${pfad}and`mo`", replace
}

```

* Übertragung der Korrekturen in den Biografie-File.

```

forvalues i = 1(1) 7 {
    local mo : word `i' of `modul'
    append using "${pfad}and`mo`"
}
sort ID_t splink
merge 1:1 ID_t splink using "${pfad}SC4_Biography_V2_D_4-0-0.dta"
replace splast = 2 if disagint == 1
replace startm = sm if sx == 1
replace starty = sy if sx == 1
drop if sx == 2
drop disagint sm sy sx _merge
sort ID_t splink
save "${pfad}bio_org", replace
forvalues i = 1(1) 8 {
    local mo : word `i' of `modul'
    erase "${pfad}and`mo`.dta"
}

```

* Funktionsblock 3:

* Wiederherstellung der Aufsetzepisoden, denen in der Welle 4 im Prüfmodul wider-
* sprochen wurde und die deswegen vollständig (also auch mit ihrem Welle3-
* Anteil) aus dem Biografie-File gelöscht wurden. Die Episoden bleiben in der
* Variable spms als "im Prüfmodul widersprochen" markiert.

* Selektion der im Prüfmodul widersprochenen Episoden.

```

local datstam "23 35 21 27 11 25 13 15"
local modul "Emp Gap Military ParLeave School Unemp VocPrep VocTrain"
local typ "26 30 25 29 22 27 23 24"
forvalues i = 1(1) 8 {
    local da : word `i' of `datstam'
    local mo : word `i' of `modul'
    local ty : word `i' of `typ'
    use "${pfad}SC4_sp`mo`_V2_D_4-0-0.dta", clear
    keep if spms == -20 & subspell == 2
    merge m:1 ID_t using "${pfad}intdat_w3"
    keep if _merge == 3
}

```

* Umwandlung dieser Episoden in die Struktur des Biografie-Files und

* Speicherung dieser neustrukturierten Records in einen separaten File.

```

gen sptype = `ty'
gen startm = ts`da`llm

```

```

    replace startm = startm - 20 if startm > 12 & startm < 40
    gen starty = ts`da'11y
    gen endm = intm
    gen endy = inty
    gen splast = 2
    replace wave = wave - 1
    keep ID_t wave sptype splink spms startm starty endm endy splast
    save "${pfad}bio9_`mo'", replace
}

* Integration der neu generierten Episoden in den Biografie-File.
use "${pfad}bio_org", clear
forvalues i = 1(1) 8 {
    local mo : word `i' of `modul'
    append using "${pfad}bio9_`mo'"
}
sort ID_t splink
save "${pfad}bio_org", replace
forvalues i = 1(1) 8 {
    local mo : word `i' of `modul'
    erase "${pfad}bio9_`mo'.dta"
}

* Funktionsblock 4:
* Separates Sichern von Episoden ohne berechenbare Datumsangabe. Separates
* Sichern von Episoden des Ereignistyps "Nebenereignis". Diese Episoden werden
* dem Biografie-File später wieder hinzugefügt.
use "${pfad}bio_org", clear
replace starty = 2002 if ID_t == 4015542 & splink == 220002
drop if starty > 0
save "${pfad}bio0", replace
use "${pfad}bio_org", clear
replace starty = 2002 if ID_t == 4015542 & splink == 220002
drop if spms != 2
save "${pfad}bio0b", replace

* Beseitigung von einmonatigen Überschneidungen zwischen Hauptereignisepisoden.
* Die jeweils zeitlich vorangehende Episode wird um einen Monat gekürzt, falls
* sie nicht nur einen Monat andauert. Falls die vorangehend Episode nur einen
* Monat andauert, wird die nachfolgende Episode um einen Monat gekürzt.
use "${pfad}bio_org", clear
replace starty = 2002 if ID_t == 4015542 & splink == 220002
drop if starty == -55
drop if spms == 2
gen start = ((starty - 1960) * 12) + startm
gen end = ((endy - 1960) * 12) + endm
gen dur = end - start + 1

* Erzeugung eines Files mit monatsbasierten Records.
expand dur
bysort ID_t splink: gen x = _n -1
gen monat = start + x
drop x
sort ID_t monat

* Markierung von Episodenmonaten, die Startmonat einer Episode sind.
gen markst = 1 if monat == start

* Markierung von Episodenmonaten, die Endmonat einer Episode sind.
gen marken = 1 if monat == end

```

```

save "${pfad}bio1", replace

* Markierung von Monaten, die sowohl Start- als auch Enddatum von Episoden sind.
gen markstmax = markst
gen markenmax = marken

* Markierung von Monaten, zu denen es eine einmonatige Episode gibt.
gen durmin = dur
collapse (max) markstmax markenmax (min) durmin, by (ID_t monat)

* Zuspielen der vorher generierten Monatsinformationen zu den Episodenmonats-
* daten.
merge 1:m ID_t monat using "${pfad}bio1"
sort ID_t monat
gen mark = .

* Markierung von Episodenmonaten, die am Ende gekürzt werden sollen.
replace mark = 1 if markstmax == 1 & markenmax == 1 & markst != 1 & marken == 1

* Markierung von Episodenmonaten, die am Beginn gekürzt werden sollen.
replace mark = 2 if markstmax == 1 & markenmax == 1 & markst == 1 & marken != 1 &
durmin == 1

* Rekonstruktion des Biografie-Files.
collapse (max) wave sptype spms splast startm starty endm endy dur (sum) mark,
by(ID_t splink)

* Festlegung, dass zweimonatige Episoden, die sowohl für eine Kürzung am End-
* datum, als auch am Beginndatum markiert sind, nur am Enddatum um einen Monat
* gekürzt werden.
replace mark = 1 if mark == 3 & dur == 2

* Änderung der Datumsangaben in Abhängigkeit von der Markierung der Episode.
replace endm = endm - 1 if (mark == 1 | mark == 3) & endm > 1
replace endy = endy - 1 if (mark == 1 | mark == 3) & endm == 1
replace endm = 12 if (mark == 1 | mark == 3) & endm == 1
replace startm = startm + 1 if (mark == 2 | mark == 3) & startm < 12
replace starty = starty + 1 if (mark == 2 | mark == 3) & startm == 12
replace startm = 1 if (mark == 2 | mark == 3) & startm == 12
drop dur mark
append using "${pfad}bio0b"
sort ID_t splink
save "${pfad}bio2", replace

* Funktionsblock 5:
* Schließung von ein- bis zweimonatigen Lücken. Bei einmonatigen Lücken wird das
* Ende der vorangehende Episode um einen Monat verlängert, bei zweimonatigen
* Lücken wird zusätzlich der Beginn der folgenden Episode um einen Monat vor-
* verlegt.
use "${pfad}bio2", clear
gen start = ((starty - 1960) * 12) + startm
gen end = ((endy - 1960) * 12) + endm
gen dur = end - start + 1
expand dur
bysort ID_t splink: gen x = _n -1
gen monat = start + x
drop x
sort ID_t monat splink
save "${pfad}bio3", replace

```

```

* Erzeugung eines Files mit monatsbasierten Records.
collapse (max) dur, by(ID_t monat)
drop dur

* Überprüfung des Abstands zwischen den Monatsangaben.
gen test = monat - monat[_n-1] if ID_t == ID_t[_n-1]

* Markierung von zweimonatigen Lücken zur Verlängerung der nachfolgenden Episode.
gen mark = 2 if test == 3

* Markierung von ein- bis zweimonatigen Lücken zur Verlängerung der voran-
* gehenden Episoden.
gsort +ID_t -monat
replace mark = 1 if (test[_n-1] == 2 | test[_n-1] == 3) & mark != 2
replace mark = 3 if (test[_n-1] == 2 | test[_n-1] == 3) & mark == 2
sort ID_t monat
collapse (max) mark, by(ID_t monat)

* Anspielen der Markierungsinformationen an den monatsbasierten Datenfile.
merge 1:m ID_t monat using "${pfad}bio3"

* Rekonstruktion des Biografie-Files.
collapse (max) wave sptype spms splast startm starty endm endy (sum) mark, by(ID_t splink)

* Änderung der Datumsangaben in Abhängigkeit von der Markierung der Episode.
replace endm = endm + 1 if (mark == 1 | mark == 3) & endm < 12
replace endy = endy + 1 if (mark == 1 | mark == 3) & endm == 12
replace endm = 1 if (mark == 1 | mark == 3) & endm == 12
replace startm = startm - 1 if (mark == 2 | mark == 3) & startm > 1
replace starty = starty - 1 if (mark == 2 | mark == 3) & startm == 1
replace startm = 12 if (mark == 2 | mark == 3) & startm == 1
drop mark
sort ID_t splink
save "${pfad}bio4", replace

* Funktionsblock 6:
* Schließung von Lücken mit mehr als zwei Monaten. Es wird eine Episode vom Typ
* "Dateneditionslücke" mit dem sptype 99 angelegt.
use "${pfad}bio4", clear
gen start = ((starty - 1960) * 12) + startm
gen end = ((endy - 1960) * 12) + endm

* Erzeugung eines Files mit monatsbasierten Records.
gen dur = end - start + 1
expand dur
bysort ID_t splink: gen x = _n - 1
gen monat = start + x
drop x
sort ID_t monat splink
collapse (max) dur, by(ID_t monat)
drop dur

* Markierung von Monaten, auf denen eine Lücke folgt und Berechnung der Lücken-
* dauer.
gen mark = .
replace mark = 1 if ID_t == ID_t[_n+1] & monat + 1 < monat[_n+1]
gen dur = .
replace dur = monat[_n+1] - monat - 1 if mark == 1

```



```

* Selektion der markierten Monate.
keep if mark == 1
gen start = monat + 1
gen end = start + dur -1
sort ID_t monat
bysort ID_t: gen x = _n

* Erzeugung einer Biografie-File-Datenstruktur und Füllung der Variablen.
gen wave = -55
gen sptype = 99
gen splink = 990000 + x
gen spms = 1
gen startm = mod(start,12)
gen starty = int(start/12) + 1960
replace starty = starty - 1 if startm == 0
replace startm = 12 if startm == 0
gen endm = mod(end,12)
gen endy = int(end/12) + 1960
replace endy = endy - 1 if endm == 0
replace endm = 12 if endm == 0
gen splast = 2
drop monat mark dur start end x
save "${pfad}bio5", replace

* Funktionsblock 7:
* Schließung von Lücken zwischen dem 8. Geburtstag der Befragten und dem Beginn
* der frühesten erfassten Episode. Dabei handelt es sich meist um Fälle, die
* nicht an der zweiten Erhebungswelle teilgenommen haben und denen deshalb die
* retrospektive Schulgeschichte fehlt. Diese Lücke vor dem erfassten Lebens-
* verlauf erhält den Typ "Dateneditions-lücke" mit dem sptype 99 und dem Wert
* 990000 in der Variable splink.
use "${pfad}bio4", clear

* Erzeugung eines Querschnittsdatenfiles, der die Fallnummern beinhaltet.
gen mark = 1
collapse (max) mark, by(ID_t)
save "${pfad}cases", replace

* Extraktion des Geburtsdatums aus den CohortProfile-Daten.
use "${pfad}SC4_CohortProfile_D_4-0-0.dta", clear
drop if wave != 1
merge 1:1 ID_t using "${pfad}cases"
keep if _merge == 3
keep ID_t tx8050m tx8050y
gen gebdat = ((tx8050y -1960) * 12) + tx8050m
replace gebdat = -97 if tx8050y == -97
drop tx8050m tx8050y
save "${pfad}gebdat", replace
use "${pfad}bio4", clear

* Anspielen des Geburtsdatums an den Biografie-File und Erzeugung einer Variable
* mit dem 8. Geburtstag der Befragten.
merge m:1 ID_t using "${pfad}gebdat"
drop if gebdat == -97
gen gebdat8 = gebdat + 96
gen start = ((starty - 1960) * 12) + startm

* Erzeugung eines Querschnittsfiles mit dem frühesten Startdatum des Lebenslaufs
* und dem Datum des 8. Geburtstags.
collapse (min) start gebdat8, by (ID_t)

```

```

* Selektion der Fälle, deren frühestes Startdatum nach dem 8.Geburtstag liegt.
gen mark = 1 if start > gebdat8
drop if mark == .

* Erzeugung einer Biografie-File-Datenstruktur und Füllung der Variablen.
gen wave = -55
gen sptype = 99
gen splink = 990000
gen spms = 1
gen startm = mod(gebdat8,12)
gen starty = int(gebdat8/12) + 1960
replace starty = starty - 1 if startm == 0
replace startm = 12 if startm == 0
gen endm = mod(start - 1,12)
gen endy = int((start - 1)/12) + 1960
replace endy = endy - 1 if endm == 0
replace endm = 12 if endm == 0
gen splast = 2
drop start gebdat8 mark
save "${pfad}bio6", replace

* Funktionsblock 8:
* Schließung von Lücken zwischen letzter Episode und Interviewzeitpunkt. Es
* wird eine Episode vom Typ "Dateneditions-lücke" mit dem sptype 99 angelegt,
* wenn die Dauer zwischen Interviewdatum und Ende der letzten Episode größer
* als zwei Monate ist. Ist die Dauer zwischen Interviewdatum und Ende der
* letzten Episode geringer als drei Monate, dann handelt es sich um eine
* Unterbrechung und spätere Fortführung des Interviews nach der Lebensver-
* laufserfassung. In diesem Fall wird eine Episode vom Typ "Editionslücke:
* Interviewunterbrechung nach LV-Erfassung" mit dem neuen Code 98 eingefügt.
use "${pfad}SC4_TargetMethods_D_4-0-0.dta", clear

* Extraktion des Interviewdatums aus den TargetMethods-Daten.
merge m:1 ID_t using "${pfad}cases"
keep if _merge == 3
drop if inty == . | inty == -55
gen intdat = ((inty - 1960) * 12) + intm
collapse (max) intdat, by (ID_t)
save "${pfad}intdat", replace

* Anspielen des Geburtsdatums an den Biografie-File.
merge 1:m ID_t using "${pfad}bio4"
gen end = ((endy - 1960) * 12) + endm

* Erzeugung eines Querschnittsfiles mit dem spätestem Enddatum des Lebenslaufs
* und dem Interviewdatum.
collapse (max) end intdat splink, by (ID_t)

* Selektion der Fälle, deren spätestes Enddatum vor dem Interviewdatum liegt.
gen mark = 1 if end < intdat
replace mark = 2 if end > intdat
drop if mark != 1

* Erzeugung einer Biografie-File-Datenstruktur und Füllung der Variablen.
gen wave = -55
gen sptype = 99
replace sptype = 98 if intdat - end < 3
replace splink = splink + 1 if splink >= 990001 & sptype == 99
replace splink = 990001 if splink < 990001 & sptype == 99
replace splink = 980001 if sptype == 98

```

```

gen spms = 1
gen startm = mod(end,12) + 1
gen starty = int(end/12) + 1960
replace starty = starty + 1 if startm == 1
gen endm = mod(intdat,12)
gen endy = int(intdat/12) + 1960
replace endy = endy - 1 if endm == 0
replace endm = 12 if endm == 0
gen splast = 1
drop end intdat mark
save "${pfad}bio7", replace

* Funktionsblock 9:
* Einfügen einer Editionsücke für Fälle ohne Episodendaten (nur 1 Fall vor-
* handen: 4014657). Diese Editionsücke wird identisch konstruiert wie Lücken
* vor dem Lebenslauf. Das Enddatum wird aus dem Interviewdatum der letzten ,
* Erhebungswelle an der diese Person teilgenommen hat, generiert.
use "${pfad}SC4_TargetMethods_D_4-0-0.dta", clear

* Selektion aller Fälle mit gültigem Interviewdatum aus TargetMethods-Daten.
sort ID_t wave
keep if inty != . & inty != -55

* Löschung der Fälle, für die es zumindest eine Episode im Biografie-File gibt.
merge m:1 ID_t using "${pfad}gebdat"
drop if gebdat != .

* Erzeugung eines Querschnittsfiles mit der letzten Information zur Wellenteil-
* nahme und zum Interviewdatum.
collapse (last) wave intm inty, by (ID_t)
save "${pfad}case_nolv", replace
use "${pfad}SC4_CohortProfile_D_4-0-0.dta", clear
drop if wave != 1
drop wave

* Anspielen des Interviewdatums und der Wellenteilnahme an CohortProfile-Daten.
merge 1:1 ID_t using "${pfad}case_nolv"
keep if _merge == 3

* Erzeugung einer Biografie-File-Datenstruktur und Füllung der Variablen.
gen sptype = 99
gen splink = 990000
gen spms = 1
gen startm = tx8050m
gen starty = tx8050y + 8
gen endm = intm
gen endy = inty
gen splast = 1
keep ID_t wave sptype splink spms startm starty endm endy splast
save "${pfad}bio8", replace

* Funktionsblock 10:
* Zusammenfügung der korrigierten Biografie-Informationen.
use "${pfad}bio2", clear
drop if ID_t > 0
local fname "4 0 5 6 7 8"
forvalues i = 1(1) 6 {
    local fn : word `i' of `fname'
    append using "${pfad}bio`fn'"
}

```

```

sort ID_t splink
label define de1870 98 "Editionslücke: Interviewunterbrechung nach LV-Erfassung",
add
label define de2574 -55 "nicht ermittelbar", add
save "${pfad}SC4_Biography_V2_D_4-0-0.dta", replace

* Löschung der temporären Zwischendatensätze.
local fname "bio_org bio0 bio0b bio1 bio2 bio3 bio4 bio5 bio6 bio7 bio8 case_nolv
cases gebdat intdat intdat_w3"
forvalues i = 1(1) 16 {
    local fn : word `i' of `fname'
    erase "${pfad}\`fn'.dta"
}

```