

Einführung in die eigenständige Erstellung von Typisierungen am Beispiel des Mikrozensus Scientific Use Files 2014

Börlin, Simon

Veröffentlichungsversion / Published Version

Verzeichnis, Liste, Dokumentation / list

Zur Verfügung gestellt in Kooperation mit / provided in cooperation with:

GESIS - Leibniz-Institut für Sozialwissenschaften

Empfohlene Zitierung / Suggested Citation:

Börlin, S. (2020). *Einführung in die eigenständige Erstellung von Typisierungen am Beispiel des Mikrozensus Scientific Use Files 2014*. (GESIS Papers, 2020/01). Köln: GESIS - Leibniz-Institut für Sozialwissenschaften. <https://doi.org/10.21241/ssoar.66725>

Nutzungsbedingungen:

Dieser Text wird unter einer CC BY Lizenz (Namensnennung) zur Verfügung gestellt. Nähere Auskünfte zu den CC-Lizenzen finden Sie hier:
<https://creativecommons.org/licenses/by/4.0/deed.de>

Terms of use:

This document is made available under a CC BY Licence (Attribution). For more information see:
<https://creativecommons.org/licenses/by/4.0>

gesis

Leibniz-Institut
für Sozialwissenschaften

GESIS Papers

2020|01

**Einführung in die eigenständige
Erstellung von Typisierungen
am Beispiel des Mikrozensus
Scientific Use Files 2014**

Simon Börlin

GESIS Papers 2020|01

**Einführung in die eigenständige
Erstellung von Typisierungen am
Beispiel des Mikrozensus
Scientific Use Files 2014**

Simon Börlin

GESIS Papers

GESIS – Leibniz-Institut für Sozialwissenschaften
Dauerbeobachtung der Gesellschaft (DBG)
German Microdata Lab (GML)
Postfach 12 21 55
68072 Mannheim

E-Mail: gml@gesis.org

ISSN: 2364-3781 (Online)
Herausgeber,
Druck und Vertrieb: GESIS – Leibniz-Institut für Sozialwissenschaften
Unter Sachsenhausen 6-8, 50667 Köln

1 Überblick

Der Mikrozensus (MZ) ist die amtliche Haushaltsbefragung für Informationen zur Bevölkerung und zum Arbeitsmarkt. Er ist als einstufige Klumpenstichprobe mit einem Auswahlsatz von einem Prozent angelegt¹. Alle Haushalte und Personen, die in den ausgewählten Klumpen wohnen, werden als Beobachtungseinheiten erhoben. Darum liegen Informationen über alle sekundären Erhebungseinheiten vor. Dies sind neben den Personen eines Haushalts bzw. einer Gemeinschaftsunterkunft die Einheiten Haushalt, Lebensform und Familie. Zur Analyse dieser unterschiedlichen Ebenen werden einige Informationen, sogenannte Bandsatzergänzungen, nachträglich von dem Statistischen Bundesamt auf Basis der direkt erhobenen Fragen generiert. Zur Vereinfachung der Datenaufbereitung sind in dem Scientific Use File (SUF) des MZ ab 2015 solche Bandsatzergänzungen nicht mehr in den bereitgestellten Daten enthalten (Statistisches Bundesamt und GESIS 2019). Diese Typisierungen können für viele Fragestellungen relevant sein. Sie enthalten jedoch überwiegend Angaben, die aus anderen in den Daten enthaltenen Variablen eigenständig erstellt werden können. Darum wird in dieser Einführung anhand der Daten des MZ SUF 2014 (FDZ 2019) beschrieben, wie die betreffenden Typisierungen eigenständig auf Basis bestehender Variablen generiert werden können. In diesem Zuge wird die im Anhang und auf dem GESIS Mikrodateninformationssystem (MISSY) bereitgestellte Syntax² für die Statistikprogramme Stata, SPSS und SAS an einigen Beispielen näher erläutert.³

Tabelle 5 im Anhang gibt einen ersten Überblick über die 156 Typisierungen, welche in den bereitgestellten Syntaxen repliziert werden. Sie beinhalten unterschiedliche Informationen, die über verschiedenen Ebenen erzeugt werden können. Diese Ebenen sind:

- Haushalt (HH)
- Lebensform (LF)
- Familie (FAM)
- Haushaltsbezugsperson (HHBP)
- Lebensformbezugsperson (LFBP)
- Haupteinkommensbezieher (HEB)
- Lebenspartner der Bezugsperson der Lebensform (LP)

Zur Definition der einzelnen Begriffe sei an dieser Stelle auf die Seiten des GESIS Mikrodateninformationssystems (MISSY) verwiesen.⁴

In dieser Einführung gliedert sich das folgende Kapitel über das grundlegende Vorgehen bei satzübergreifenden Typisierungen in zwei Unterabschnitte. Dabei behandelt der erste Abschnitt 2.1 die Generierung auf Ebene von Haushalten, Lebensformen und Familien. Abschnitt 2.2 zeigt das Vorgehen für die übrigen Ebenen auf. In Abschnitt 3 wird eine alternative Erstellung von Typisierungen mithilfe zuvor erstellter Hilfsvariablen skizziert, die vor allem bei häufigen satzübergreifenden Arbeiten den Umgang erleichtert. Abschnitt 4 stellt abschließend etwaige Abweichungen der erstellten Typisierungen zu den Originalvariablen dar und benennt Gründe für diese Differenzen.

¹ Das hier verwendete Scientific Use File des MZ ist eine 70 prozentige Substichprobe.

² An dieser Stelle sei Sophie Kittelberger und Lukas Zettl für die äußerst hilfreiche Unterstützung bei der Erstellung der Syntaxen für Stata, SPSS und SAS gedankt.

³ <https://www.gesis.org/missy/materials/MZ/tools/typisierungen>

⁴ <https://www.gesis.org/missy/materials/MZ/definitionen>

2 Das grundlegende Vorgehen bei der eigenständigen Erstellung satzübergreifender Typisierungen

In diesem Abschnitt werden einige Variablen beispielhaft nachgebildet. Dabei befasst sich Abschnitt 2.1 mit der Generierung von Variablen auf Ebene von Haushalten, Lebensformen und Familien. Abschnitt 2.2 stellt das Vorgehen bei der Generierung von Typisierungen mit Haushalts- und Lebensformbezugspersonen, Haupteinkommensbeziehern und Lebenspartnern von Lebensformbezugspersonen dar.

2.1 Generierung von Typisierungen auf Haushalts- und Lebensform- und Familienebene

Ein Großteil der zu generierenden Variablen bezieht sich auf Informationen auf Haushalts- (HH), Lebensform- (LF) oder Familienebene (FAM). Die Grundidee bei der Erstellung solcher Typisierungen ist, dass eine bestimmte Information aus einer Variable auf alle Personen einer bestimmten Ebene (bspw. alle Personen eines Haushalts) übertragen wird. Da das Vorgehen für alle drei Ebenen sehr ähnlich ist, wird im Folgenden lediglich ein Beispiel auf Haushaltsebene näher erläutert.

In Tabelle 1 sind die Syntaxen zur Replikation der Variablen EF663 „Personen im Haushalt: Anzahl (Haushaltstypisierung)“ in Stata, SPSS und SAS dargestellt. Anhand der Befehle für Stata wird das grundlegende Vorgehen bei der Erstellung solcher Typisierungen nun näher erläutert.

Der Befehl `egen` erstellt eine neue Variable (`v663`) und füllt sie mit der Anzahl der Beobachtungen. Die Option `by()` begrenzt hierbei die Beobachtungen auf Ebene von Privathaushalten⁵. Die Verkettung der folgenden Variablen identifiziert die einzelnen Haushalte, sodass die Anzahl der Beobachtungen für jeden einzelnen Haushalt separat gezählt werden:

EF1 „Bundesland“

EF3 „Nr. des Auswahlbezirks (systemfrei)“

EF4 „Nr. des Haushalts im Auswahlbezirk (systemfrei)“

EF12 „Berichtsquartal“

EF5b „Jahresüberhang aus dem Vorjahr“

Anschließend wird die Variable gelabelt (`label variable`). Da die Originalvariable EF663 bei dem Wert 9 eine Vergrößerung aufweist (Value Label: „9 und mehr Personen“), wird diese Vergrößerung über den Befehl `recode` ebenfalls umgesetzt. Im Anschluss werden die User-Missings aus der Originalvariablen mithilfe der Variablen EF31 („Bevölkerung: Privathaushalte“) durch den Befehl `replace` angepasst. Die beiden Entfällt-Kategorien mit den Werten -8 („Leerstehende Wohnung, ausgefallene Privathaushalte und Gemeinschaftsunterkünfte“) und -7 („Auswahlbezirk ohne befragten Haushalt“) sowie die Bevölkerung in Gemeinschaftsunterkünften (Wert -1) besitzen dadurch wie in der Originalvariablen keine inhaltlich gültigen Werte mehr. Abschließend wird mit dem Befehl `tab` die Originalvariable (EF663) mit der selbst generierten Variable `v663` zur Kontrolle kreuztabelliert. Sind die beiden Variablen identisch, ist die Replikation gelungen. Bei Bedarf kann die Labelung der einzelnen Werte (Value Label) noch ergänzt werden. Aus Darstellungsgründen wird darauf verzichtet.

⁵ Personen in Gemeinschaftsunterkünften sind nicht Teil dieser Typisierung.

Tabelle 1: Syntax zur Erstellung der Typisierung EF663 „Personen im Haushalt: Anzahl (Haushaltstypisierung)“

Stata	SPSS	SAS
<pre>* EF663: Personen im Haus- halt: Anzahl egen v663 = total (1), by (EF1 EF3 EF4 EF12 EF5b) label variable v663 "Personen im Haushalt: Anzahl (Haus- haltstypisierung)" recode v663 10/99 = 9 replace v663 = -8 if EF31==8 replace v663 = -7 if EF31==7 replace v663 = -1 if EF31==2 tab EF663 v663, m</pre>	<pre>* EF663: Personen im Haus- halt: Anzahl. temporary. compute EF663k = 1. aggregate outfile = * mode = addvariables /break EF1 EF3 EF4 EF5b EF12 /v663 = SUM (EF663k). var lab v663 "Personen im Haushalt: Anzahl (Haus- haltstypisierung)". if(v663>= 10) v663 = 9. if(EF31= -8) v663 = -8. if(EF31= -7) v663 = -7. if(EF31= 2) v663 = -1. crosstabs EF663 by v663.</pre>	<pre>* EF663: Personen im Haushalt: Anzahl data tmp14 (compress = yes); set mz14.suf2014 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF663); run; proc sort data = tmp14; by EF1 EF3 EF4 EF5b EF12; run; proc means data = tmp14 (keep=EF1 EF3 EF4 EF5b EF12 EF31 where=(EF31=1)) nway noprint; by EF1 EF3 EF4 EF5b EF12; var EF31; output out = sum(drop= type freq) sum = v663; run; data tmp14 (compress = yes); merge tmp14 sum; by EF1 EF3 EF4 EF5b EF12; run; data tmp14; set tmp14; if (v663 >= 9 & v663 <= 99) then v663 = 9; if (v663 = . & EF31 = -8) then v663 = -8; if (v663 = . & EF31 = -7) then v663 = -7; if (v663 = . & EF31 = 2) then v663 = -1; run; proc freq data = tmp14 (where= (v663 ~= EF663)); table v663 * EF663; run;</pre>

Die Syntaxen für SPSS und SAS sind nach der gleichen Logik aufgebaut und werden an dieser Stelle nicht näher erläutert. Die vollständigen Syntaxen im Anhang und auf den Seiten von MIS-SY⁶ sind zum besseren Verständnis jedoch kommentiert.

Das Vorgehen bei der Generierung von Typisierungen auf Ebene der Lebensformen (LF) ist der Vorgehensweise auf Haushaltsebene sehr ähnlich. Die Abgrenzung der Einheiten über die Verkettung der Identifikationsvariablen wird um die Variable für die Lebensform ergänzt (bspw. in Stata mit `by(EF1 EF3 EF4 EF12 EF5b EF27)`):

EF27 „Nr. der Lebensform im Haushalt (Konzept der Lebensformen)“

Bei der Erstellung von Typisierungen auf Ebene von Familien (FAM) geschieht die Ergänzung über die Identifikationsvariable für Familien. Die Verkettung bspw. in Stata mit `by(EF1 EF3 EF4 EF12 EF5b EF25)` ist demnach um folgende Variable ergänzt:

EF25 „Nr. der Familie im Haushalt (trad. Familienkonzept)“

⁶ <https://www.gesis.org/missy/materials/MZ/tools/typisierungen>

2.2 Generierung von Typisierungen mit Haushalts- und Lebensformbezugspersonen, Haupteinkommensbeziehern, Lebenspartnern von Lebensformbezugspersonen

Bei Typisierungen zu Haushaltsbezugspersonen (HHBP), Lebensformbezugspersonen (LFBP), Haupteinkommensbeziehern (HEB) und Lebenspartnern von Lebensformbezugspersonen (LP) wird die grundlegende Syntax aus Abschnitt 2.1 ergänzt. Der Grundgedanke bei der Erstellung dieser Art von Typisierungen ist, dass die Information einer bestimmten Person auf andere Personen innerhalb einer Ebene übertragen wird. Neu ist demnach bei dieser Art der Generierung die zusätzliche Eingrenzung der entsprechenden Ausgangsperson (bspw. die LFBP), von der bestimmte Informationen auf die weitere Ebene (bspw. die Lebensform) übertragen werden soll. Dies wird am Beispiel der Typisierung EF819 „Geschlecht: Bezugsperson der Lebensform“ verdeutlicht.

Tabelle 2: Syntax zur Erstellung der Typisierung EF819 „Geschlecht: Bezugsperson der Lebensform“

Stata	SPSS	SAS
<pre>* EF819: Geschlecht: Bezugsperson der Lebensform egen v819 = sum (EF46*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27) label variable v819 "Geschlecht Bezugsperson Lebensform" replace v819 = -8 if EF31==8 replace v819 = -7 if EF31==7 replace v819 = -1 if EF31==2 tab EF819 v819, m</pre>	<pre>* EF819: Geschlecht: Bezugsperson der Lebensform . temporary. compute EF819k = EF46 * (EF35 = 1). aggregate outfile = * mode = addvariables /break EF1 EF3 EF4 EF5b EF12 EF27 /v819 = SUM (EF819k). var lab v819 "Geschlecht Bezugsperson Lebensform". if(EF31==8) v819 = -8. if(EF31==7) v819 = -7. if(EF31=2) v819 = -1. crosstabs EF819 by v819.</pre>	<pre>* EF819: Geschlecht: Bezugsperson der Lebensform data tmp14 (compress = yes); set mz14.suf2014 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF46 EF819); x819 = (EF46*(EF35=1 & EF44>=10)); run; proc sort data = tmp14; by EF1 EF3 EF4 EF5b EF12 EF27; run; proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF46 x819 where=(EF31=1)) nway noprint; by EF1 EF3 EF4 EF5b EF12 EF27; var x819; output out = sum(drop=_type_ _freq_) sum = v819; run; data tmp14 (compress = yes); merge tmp14 sum; by EF1 EF3 EF4 EF5b EF12 EF27; run; data tmp14; set tmp14; if (v819 = . & EF31 = -8) then v819 = -8; if (v819 = . & EF31 = -7) then v819 = -7; if (v819 = . & EF31 = 2) then v819 = -1; run; proc freq data = tmp14 (where= (v819 ^= EF819)); table v819 * EF819; run;</pre>

Die Syntax aus Tabelle 2 ist sehr ähnlich zur Syntax aus Tabelle 1 in Abschnitt 2.1. In Stata ist lediglich der Befehl `egen` etwas anders aufgebaut. Es wird die neue Variable `v819` gebildet und diese

mit der Angabe über das Geschlecht (EF46) der Lebensformbezugsperson (EF35=1) für jede Lebensform (`by (EF1 EF3 EF4 EF12 EF5b EF27)`) gefüllt. Das Vorgehen bei der Labelung, der Missingrekodierungen und der abschließenden Kontrolle ist aus Abschnitt 2.1 bekannt.

Für die Generierung von Typisierungen auf Ebene der HHBP wird stattdessen Variable EF37 „Stellung der Person im Haushalt“ zur Abgrenzung verwendet und bei der Verkettung auf Haushaltsebene auf Variable EF27 verzichtet (siehe auch Abschnitt 2.1). Die Abgrenzung der HEB geschieht über Variable EF662 „Haupteinkommensbezieher im Haushalt“ und EF31 „Bevölkerung: Privathaushalte“, währenddessen für LP EF35 „Stellung innerhalb der Lebensform“ benötigt wird. Das folgende Kapitel 3 liefert hierüber noch weitere Informationen.

3 Erstellung von Hilfsvariablen für die Ebene der Eingrenzung (in Stata und SPSS)

Bei der Erstellung aller Typisierungen können alternativ in einem ersten Schritt zunächst Hilfsvariablen erstellt werden, welche dann bei der Generierung der Typisierungen Anwendung finden. In den bereitgestellten Syntaxen für Stata und SPSS⁷ sind diese Befehle auskommentiert, da in der Regel nur einzelne Typisierungen gebraucht werden. Die Erstellung dieser Hilfsvariablen ist vor allem dann sinnvoll, wenn eine größere Anzahl an Typisierungen erstellt oder die Syntax lesefreundlicher gestaltet werden soll. Auch bei häufigen satzübergreifenden Arbeiten mit den Daten des MZ SUF können diese Hilfsvariablen hilfreich sein. Konkret sind in den bereitgestellten Syntaxen Hilfsvariablen für folgende fünf Ebenen aufgelistet und die hierfür nötigen Ausgangsvariablen:

Haushaltsbezugs person (HHBP): EF37 „Stellung der Person im Haushalt“

Lebensformbezugs person (LFBP): EF35 „Stellung innerh. der Lebensform (Konzept der Lebensformen)“

Haupteinkommensbezieher (HEB): EF662 „Haupteinkommensbezieher im Haushalt“ und EF31 „Bevölkerung: Privathaushalte“

Lebenspartner der Bezugsperson der Lebensform (LP): EF35 „Stellung innerh. der Lebensform (Konzept der Lebensformen)“

In Tabelle 3 sind die Syntaxen für Stata und SPSS aufgeführt, wie diese Hilfsvariablen erstellt werden können. Im Wesentlichen werden alle Hilfsvariablen in Stata mit den üblichen `generate`-Befehlen bzw. der Kombination mit `recode` erstellt. Hervorzuheben ist die Hilfsvariable HEB, welche neben der Abgrenzung mit Variable EF662 über die Identifikation der Haupteinkommensbezieher auch Variable EF31 hinzuzieht, da sie nur für Privathaushalte erstellt werden kann.

Tabelle 3: Syntax zur Erstellung der Hilfsvariablen HHBP, LFBP, HEB und LP

Stata	SPSS
<pre>* Haushaltsbezugs person (HHBP): recode EF37 (1=1) (*=0), gen(HHBP) label variable HHBP "Haushaltsbezugs person" * Lebensformbezugs person (LFBP): recode EF35 (1=1) (*=0), gen(LFBP) label variable LFBP "Lebensformbezugs person" * Haupteinkommensbezieher (HEB): gen HEB =1 if ((EF662>=1 & EF662<=4) & EF31==1) recode HEB (. =0) label variable HEB "Haupteinkommensbezieher" * Lebenspartner der Bezugsperson der Lebens- form (LP): recode EF35 (2=1) (*=0), gen(LP) label variable LP "Lebenspartner der Lebens- formbezugs person"</pre>	<pre>* Haushaltsbezugs person. compute HHBP = 0. if (EF37 = 1) HHBP = 1. var lab HHBP "Haushaltsbezugs person". * Bezugsperson in Lebensform. compute LFBP = 0. if (EF35 = 1) LFBP = 1. var lab LFBP "Lebensformbezugs person". * Haupteinkommensbezieher. compute HEB = 0. if ((EF662>=1 & EF662<=4) & EF31=1) HEB = 1. var lab HEB "Haupteinkommensbezieher". * Lebenspartner der Bezugsperson der Lebensform. compute LP = 0. if (EF35 = 2) LP = 1. var lab LP "Lebenspartner der Bezugsperson der Lebensform".</pre>

⁷ Für SAS werden keine Hilfsvariablen angeboten, da das Vorgehen zur Erstellung von Typisierungen von der Herangehensweise bei Stata und SPSS etwas abweicht.

Die auf diese Weise erstellten Hilfsvariablen ersetzen dann die manuellen Abgrenzungen in den Befehlen zur Generierung der Typisierungen. Am Beispiel aus Tabelle 2 zur Erstellung von EF819 „Geschlecht: Bezugsperson der Lebensform“ sieht der `egen`-Befehl in Stata wie folgt aus:

```
egen v819 = sum (EF46*LFBP), by (EF1 EF3 EF4 EF12 EF5b EF27)
```

Ergänzend hierzu können die Ebenen zu Haushalten, Lebensformen und Familien auch als Hilfsvariablen erstellt und verwendet werden. Das Vorgehen zur Erstellung dieser Identifikatoren (IDs) auf Ebene von Haushalten, Lebensformen und Familien basiert auf dem Vorgehen zur Erstellung des Personenidentifikators `idpersx`⁸ aus dem Bericht „Möglichkeiten der Verknüpfung von Mikrozensus-Querschnitterhebungen ab 2012 zu Panels“ (Herter-Eschweiler und Schimpl-Neimanns 2018) bzw. den hierfür bereitgestellten Syntaxen⁹. Die konkreten Syntaxen zur Erstellung der Identifikatoren sind in Tabelle 4 für Stata und SPSS dargestellt. Wird der Identifikator `idlfx` in das obige Beispiel zur Erstellung von EF819 eingesetzt, ergibt sich folgender kurzer `egen`-Befehl:

```
egen v819 = sum (EF46*LFBP), by (idlfx)
```

Anhand dieses Beispiels wird deutlich, dass sich das Arbeiten mit Hilfsvariablen und IDs vor allem dann lohnt, wenn sehr viele Typisierungen erstellt werden oder oft satzübergreifend gearbeitet wird.

⁸ Ab dem MZ SUF 2015 sind die Identifikatoren `idpers`, `idpersx`, `idhh` und `idhhx` bereits in den bereitgestellten Daten enthalten (Herter-Eschweiler und Schimpl-Neimanns 2019, 12f.).

⁹ <https://www.gesis.org/missy/materials/MZ/tools/panel2012>

Tabelle 4: Syntax zur Erstellung der IDs für Haushalte, Lebensformen und Familien

Stata	SPSS
<pre> * Haushalts-, Lebensform-, und Familienidentifikatoren (idhxx, idlfx, idfamx) * Vorbereitung der Identifikatoren: tostring EF1 , gen(s1) format(%02.0f) tostring EF3 , gen(s3) format(%06.0f) tostring EF4 , gen(s4) format(%03.0f) tostring EF5b , gen(s5b) format(%02.0f) tostring EF12 , gen(s12) format(%01.0f) tostring EF25 , gen(s27) format(%02.0f) tostring EF27 , gen(s27) format(%02.0f) * Verkettung der Ordnungsnummern zum alphanumerischen Indikator idhxx gen idhxx = s1 + s3 + s4 + s5b + s12 lab var idhxx "ID Haushalt (EF1 EF3 EF4 EF5b EF12) - Querschnittsorientiert" * Verkettung der Ordnungsnummern zum alphanumerischen Indikator idlfx gen idlfx = s1 + s3 + s4 + s5b + s12 + s27 lab var idlfx "ID Lebensform (EF1 EF3 EF4 EF5b EF12 EF27) - Querschnittsorientiert" * Verkettung der Ordnungsnummern zum alphanumerischen Indikator idfamx gen idlfx = s1 + s3 + s4 + s5b + s12 + s27 lab var idlfx "ID Familie (EF1 EF3 EF4 EF5b EF12 EF25) - Querschnittsorientiert" </pre>	<pre> * Haushalts-, Lebensform-, und Familienidentifikatoren (idhxx, idlfx, idfamx). * Vorbereitung der Identifikatoren: string s1 (A2) s3 (A6) s4 (A3) s5b(A2) s12 (A1) s25 (A2) s27 (A2). compute s1 = string(EF1,n2). compute s3 = string(EF3,n6). compute s12 = string(EF12,n1). * Führende Null umsetzen. do if (EF4>=0). compute s4 = string(EF4,n3). else if (EF4=-7). compute s4 = "0-7". end if. do if (EF5b>=0). compute s5b = string(EF5b,n2). else if (EF5b=-7). compute s5b = "-7". else if (EF5b = -8). compute s5b = "-8". end if. do if (EF25>=0). compute s25 = string(EF25,n2). else if (EF25=-7). compute s25 = "-7". else if (EF25=-8). compute s25 = "-8". end if. do if (EF27>=0). compute s27 = string(EF27,n2). else if (EF27=-7). compute s27 = "-7". else if (EF27=-8). compute s27 = "-8". end if. execute. * Verkettung der Ordnungsnummern zum alphanumerischen Indikator idhxx. string idhxx (A14). compute idhxx = con- cat(char.rpad(s1,2),char.rpad(s3,6),char.rpad(s4,3), char.rpad(s5b,2),char.rpad(s12,1)). var lab idhxx "ID Haushalt (EF1 EF3 EF4 EF5b EF12) - Querschnittsorientiert". * Verkettung der Ordnungsnummern zum alphanumerischen Indikator idlfx. string idlfx (A16). compute idlfx = con- cat(char.rpad(s1,2),char.rpad(s3,6),char.rpad(s4,3), char.rpad(s5b,2),char.rpad(s12,1),char.rpad(s27,2)). var lab idlfx "ID Lebensform (EF1 EF3 EF4 EF5b EF12 EF27) - Querschnittsorientiert". * Verkettung der Ordnungsnummern zum alphanumerischen Indikator idfamx. string idfamx (A16). compute idfamx = con- cat(char.rpad(s1,2),char.rpad(s3,6),char.rpad(s4,3), char.rpad(s5b,2),char.rpad(s12,1),char.rpad(s25,2)). var lab idfamx "ID Familie (EF1 EF3 EF4 EF5b EF12 EF25) - Querschnittsorientiert". </pre>

4 Grenzen der Replikation satzübergreifender Typisierungen

Im Zuge der eigenständigen Erstellung der 156 Typisierungen sind nicht alle erstellten Variablen vollständig mit den bereitgestellten Typisierungen aus dem MZ SUF 2014 identisch. Tabelle 5 im Anhang listet hinter dem jeweiligen Variablennamen mit einer hochgestellten Zahl in Klammern die Anzahl der abweichenden Fälle zur Originalvariablen aus dem MZ SUF 2014 auf. Die Fallzahlen für alle 22 betroffenen Variablen befinden sich im einstelligen Bereich. Diese sehr geringen Abweichungen sind darauf zurückzuführen, dass im Aufbereitungsprozess des Mikrozensus in den Statistischen Ämtern zwar umfangreiche Datenbereinigungen erfolgen, jedoch nicht alle Angaben geprüft und ggf. bereinigt werden (Statistisches Bundesamt und GESIS 2018). Insgesamt sind die Replikationen der Typisierungen am Beispiel des MZ SUF 2014 sehr gut umzusetzen.

5 Literatur

FDZ der Statistischen Ämter des Bundes und der Länder. 2019. Mikrozensus Scientific Use File 2014. DOI: 10.21242/12211.2014.00.00.3.1.0

Statistisches Bundesamt und GESIS (Hrsg.). 2018. Datenhandbuch zum Mikrozensus Scientific Use File 2014. Bonn und Mannheim.

https://www.gesis.org/missy/files/documents/MZ/DHB_2014.pdf.

Statistisches Bundesamt und GESIS (Hrsg.). 2019. Wichtige Informationen zur Nutzung des Mikrozensus Scientific Use Files 2015 - Dokumentation und Datenaufbereitung. Bonn und Mannheim.

https://www.gesis.org/missy/files/documents/MZ/readme/readme_suf2015.pdf.

Herter-Eschweiler, Robert, und Bernhard Schimpl-Neimanns. 2018. Möglichkeiten der Verknüpfung von Mikrozensus-Querschnitterhebungen ab 2012 zu Panels. Bonn und Mannheim.

https://www.gesis.org/missy/files/documents/MZ/panelbildung_suf2012.pdf.

6 Anhang

6.1 Übersicht der 156 zu generierenden Variablen

Tabelle 5: Übersicht der 156 zu generierenden Variablen

Variable	Variablenlabel	Ebene
EF663	Personen im Haushalt: Anzahl (Haushaltstypisierung)	HH
EF664	Erwerbstätige im Haushalt: Anzahl	HH
EF665	Erwerbslose (EU-Definition) im Haushalt: Anzahl	HH
EF666	Arbeitsuchende Nichterwerbspersonen im Haushalt: Anzahl	HH
EF667	Einkommensbezieher im Haushalt: Anzahl	HH
EF668	Ausländer im Haushalt: Anzahl	HH
EF669	Kinder: Anzahl im Haushalt	HH
EF670	Kinder (unter 1 Jahr): Anzahl im Haushalt	HH
EF671	Kinder (unter 2 Jahren): Anzahl im Haushalt	HH
EF672	Kinder (unter 3 Jahren): Anzahl im Haushalt	HH
EF673	Kinder (unter 4 Jahren): Anzahl im Haushalt	HH
EF674	Kinder (unter 5 Jahren): Anzahl im Haushalt	HH
EF675	Kinder (unter 6 Jahren): Anzahl im Haushalt	HH
EF676	Kinder (unter 7 Jahren): Anzahl im Haushalt	HH
EF677	Kinder (unter 8 Jahren): Anzahl im Haushalt	HH
EF678	Kinder (unter 9 Jahren): Anzahl im Haushalt	HH
EF679	Kinder (unter 10 Jahren): Anzahl im Haushalt	HH
EF680	Kinder (unter 11 Jahren): Anzahl im Haushalt	HH
EF681	Kinder (unter 12 Jahren): Anzahl im Haushalt	HH
EF682	Kinder (unter 13 Jahren): Anzahl im Haushalt	HH
EF683	Kinder (unter 14 Jahren): Anzahl im Haushalt	HH
EF684	Kinder (unter 15 Jahren): Anzahl im Haushalt	HH
EF685	Kinder (unter 16 Jahren): Anzahl im Haushalt	HH
EF686	Kinder (unter 17 Jahren): Anzahl im Haushalt	HH
EF687	Kinder (unter 18 Jahren): Anzahl im Haushalt	HH
EF688	Kinder (unter 27 Jahren): Anzahl im Haushalt	HH
EF689	Kinder (ab 18 Jahren): Anzahl im Haushalt	HH
EF690	Kinder (ab 27 Jahren): Anzahl im Haushalt	HH
EF691	Kinder (3-5 Jahre): Anzahl im Haushalt	HH
EF692	Kinder (6-9 Jahre): Anzahl im Haushalt	HH
EF693	Kinder (10-14 Jahre): Anzahl im Haushalt	HH
EF694	Kinder (15-17 Jahre): Anzahl im Haushalt	HH
EF695	Kinder (18-26 Jahre): Anzahl im Haushalt	HH

Variable	Variablenlabel	Ebene
EF696	Kinder (unter 6 Jahren): Anzahl im Haushalt, Schüler	HH
EF697	Kinder (6-9 Jahre): Anzahl im Haushalt, Schüler	HH
EF698 ⁽⁹⁾	Kinder (10-14 Jahre): Anzahl im Haushalt, Schüler	HH
EF699	Kinder (15-17 Jahre): Anzahl im Haushalt, Schüler	HH
EF700	Kinder (18-26 Jahre): Anzahl im Haushalt, Schüler	HH
EF701	Kinder (ab 27 Jahren): Anzahl im Haushalt, Schüler	HH
EF702	Personen (ab 65 Jahren): Anzahl im Haushalt	HH
EF703	Personen (ab 70 Jahren): Anzahl im Haushalt	HH
EF704	Personen (ab 75 Jahren): Anzahl im Haushalt	HH
EF705	Personen (ab 80 Jahren): Anzahl im Haushalt	HH
EF706	Personen (ab 85 Jahren): Anzahl im Haushalt	HH
EF709	Personen am Nebenwohnsitz: Anzahl im Haushalt	HH
EF711 ⁽²⁾	Familien: Anzahl im Haushalt (Konzept der Lebensformen)	HH
EF712	Ehepaare: Anzahl im Haushalt (Konzept der Lebensformen)	HH
EF713	Nichtehel. Lebensgemein.: Anzahl im Haushalt (Konzept der Lebensformen)	HH
EF714	Gleichgeschl. Lebensgemein.: Anzahl im Haushalt (Konzept der Lebensformen)	HH
EF715 ⁽²⁾	Alleinerziehende: Anzahl im Haushalt (Konzept der Lebensformen)	HH
EF716 ⁽³⁾	Alleinstehende: Anzahl im Haushalt (Konzept der Lebensformen)	HH
EF719	Eingetragene Lebenspartnerschaften: Anzahl im Haushalt	HH
EF727	1. ausl. Staatsangehörigkeit: Haupteinkommensbezieher	HEB
EF731	Geschlecht: Haupteinkommensbezieher	HEB
EF732	Alter: Haupteinkommensbezieher	HEB
EF734	Staatsangehörigkeit: Haupteinkommensbezieher	HEB
mig_heb	Migrationsstatus: Haupteinkommensbezieher	HEB
migs_heb	Staatsangehörigkeit zum Migrationsstatus: Haupteinkommensbezieher	HEB
EF735	Familienstand: Haupteinkommensbezieher, erweitert	HEB
EF736	Erwerbstyp: Haupteinkommensbezieher	HEB
EF737	Wirtschaftszweig Betrieb: Haupteinkommensbezieher (WZ08)	HEB
EF738	Beruf: Haupteinkommensbezieher (KldB 2010)	HEB
EF739	Beruf: Haupteinkommensbezieher (ISCO-08)	HEB
EF740	Stellung im Beruf: Haupteinkommensbezieher	HEB
EF741	Überw. Lebensunterhalt: Haupteinkommensbezieher	HEB
EF742	Nettoeinkommen (im letzten Monat): Haupteinkommensbezieher	HEB
EF743	Höchster allg. Schulabschluss: Haupteinkommensbezieher	HEB
EF744	Höchster berufl. Abschluss: differenz. Angabe, Haupteinkommensbezieher	HEB
EF745	Höchster schul./berufl. Abschluss: Haupteinkommensbezieher (ISCED 2011)	HEB
EF753	Geschlecht: Haushaltsbezugsp.	HHBP
EF754	Alter: Haushaltsbezugsp.	HHBP
EF756	Staatsangehörigk.: Haushaltsbezugsp.	HHBP

Variable	Variablenlabel	Ebene
EF757	Familienstand: Haushaltsbezugsp., erweitert	HHBP
EF758	Erwerbstyp: Haushaltsbezugsp.	HHBP
EF759	Wohnsitz: Haushaltsbezugsp.	HHBP
EF760	1. ausl. Staatsangehörigk.: Haushaltsbezugsp.	HHBP
EF770	Kinder: Anzahl in Lebensform	LF
EF771	Kinder (unter 1 Jahr): Anzahl in Lebensform	LF
EF772	Kinder (unter 2 Jahren): Anzahl in Lebensform	LF
EF773	Kinder (unter 3 Jahren): Anzahl in Lebensform	LF
EF774	Kinder (unter 4 Jahren): Anzahl in Lebensform	LF
EF775	Kinder (unter 5 Jahren): Anzahl in Lebensform	LF
EF776	Kinder (unter 6 Jahren): Anzahl in Lebensform	LF
EF777	Kinder (unter 7 Jahren): Anzahl in Lebensform	LF
EF778	Kinder (unter 8 Jahren): Anzahl in Lebensform	LF
EF779	Kinder (unter 9 Jahren): Anzahl in Lebensform	LF
EF780	Kinder (unter 10 Jahren): Anzahl in Lebensform	LF
EF781	Kinder (unter 11 Jahren): Anzahl in Lebensform	LF
EF782	Kinder (unter 12 Jahren): Anzahl in Lebensform	LF
EF783	Kinder (unter 13 Jahren): Anzahl in Lebensform	LF
EF784	Kinder (unter 14 Jahren): Anzahl in Lebensform	LF
EF785	Kinder (unter 15 Jahren): Anzahl in Lebensform	LF
EF786	Kinder (unter 16 Jahren): Anzahl in Lebensform	LF
EF787	Kinder (unter 17 Jahren): Anzahl in Lebensform	LF
EF788	Kinder (unter 18 Jahren): Anzahl in Lebensform	LF
EF789	Kinder (unter 27 Jahren): Anzahl in Lebensform	LF
EF790	Kinder (ab 18 Jahren): Anzahl in Lebensform	LF
EF791	Kinder (ab 27 Jahren): Anzahl in Lebensform	LF
EF792	Kinder (3-5 Jahre): Anzahl in Lebensform	LF
EF793	Kinder (6-9 Jahre): Anzahl in Lebensform	LF
EF794	Kinder (10-14 Jahre): Anzahl in Lebensform	LF
EF795	Kinder (15-17 Jahre): Anzahl in Lebensform	LF
EF796	Kinder (18-26 Jahre): Anzahl in Lebensform	LF
EF797	Kinder (unter 6 Jahren): Anzahl in Lebensform, Schüler	LF
EF798	Kinder (6-9 Jahre): Anzahl in Lebensform, Schüler	LF
EF799	Kinder (10-14 Jahre): Anzahl in Lebensform, Schüler	LF
EF800	Kinder (15-17 Jahre): Anzahl in Lebensform, Schüler	LF
EF801	Kinder (18-26 Jahre): Anzahl in Lebensform, Schüler	LF
EF802	Kinder (ab 27 Jahren): Anzahl in Lebensform, Schüler	LF
EF803	Alter: Jüngstes Kind in Lebensform	LF
EF804	Alter: Ältestes Kind in Lebensform	LF

Variable	Variablenlabel	Ebene
EF812	Personen am Nebenwohnsitz: Anzahl in Lebensform	LF
EF819	Geschlecht: Bezugsp. der Lebensform	LFBP
EF820	Alter: Bezugsp. der Lebensform	LFBP
EF822	Staatsangehörigk.: Bezugsp. der Lebensform	LFBP
mig_lf1	Migrationsstatus: Bezugsp. der Lebensform	LFBP
migs_lf1	Staatsangehörigk. zum Migrationsstatus: Bezugsp. der Lebensform	LFBP
EF823	Familienstand: Bezugsp. der Lebensform, erweitert	LFBP
EF824	Erwerbstyp: Bezugsp. der Lebensform	LFBP
EF825	Wirtschaftszweig Betrieb: Bezugsp. der Lebensform (WZ08)	LFBP
EF826	Beruf: Bezugsp. der Lebensform (KldB 2010)	LFBP
EF827	Beruf: Bezugsp. der Lebensform (ISCO-08)	LFBP
EF828	Stellung im Beruf: Bezugsp. der Lebensform	LFBP
EF829	Überw. Lebensunterhalt: Bezugsp. der Lebensform	LFBP
EF830	Nettoeinkommen (im letzten Monat): Bezugsp. der Lebensform	LFBP
EF831	Höchster allg. Schulabschluss: Bezugsp. der Lebensform	LFBP
EF832	Höchster berufl. Abschluss: differenz. Angabe, Bezugsp. der Lebensform	LFBP
EF833	Höchster schul./berufl. Abschluss: Bezugsp. der Lebensform (ISCED 2011)	LFBP
EF834	Arbeitszeittyp: Bezugsp. der Lebensform	LFBP
EF835	Normale Arbeitszeit (je Woche: Stunden): Bezugsp. der Lebensform	LFBP
EF836 ⁽²⁾	Grund geringerer Arbeitszeit (Berichtsw.): Bezugsp. der Lebensform	LFBP
EF837 ⁽²⁾	Erwerbsunterbrechung (Berichtsw.): Bezugsp. der Lebensform	LFBP
EF838	1. ausl. Staatsangehörigk.: Bezugsp. der Lebensform	LFBP
EF843	Geschlecht: Lebenspartner der Bezugsp. der Lebensform	LP
EF844 ⁽³⁾	Alter: Lebenspartner der Bezugsp. der Lebensform	LP
EF846 ⁽³⁾	Staatsangehörigk.: Lebenspartner in der Lebensform	LP
mig_lf2	Migrationsstatus: Lebenspartner in der Lebensform	LP
migs_lf2	Staatsangehörigk. zum Migrationsstatus: Lebenspartner in der Lebensform	LP
EF847 ⁽³⁾	Familienstand: Lebenspartner in der Lebensform	LP
EF848 ⁽³⁾	Erwerbstyp: Lebenspartner in der Lebensform	LP
EF849 ⁽³⁾	Wirtschaftszweig Betrieb: Lebenspartner in der Lebensform (WZ08)	LP
EF850	Beruf: Lebenspartner in der Lebensform (KldB 2010)	LP
EF851 ⁽¹⁾	Beruf: Lebenspartner in der Lebensform (ISCO-08)	LP
EF852 ⁽³⁾	Stellung im Beruf: Lebenspartner in der Lebensform	LP
EF853 ⁽³⁾	Überw. Lebensunterhalt: Lebenspartner in der Lebensform	LP
EF854 ⁽³⁾	Nettoeinkommen (im letzten Monat): Lebenspartner in der Lebensform	LP
EF855 ⁽³⁾	Höchster allg. Schulabschluss: Lebenspartner in der Lebensform	LP
EF856 ⁽³⁾	Höchster berufl. Abschluss: differenz. Angabe, Lebenspartner in der Lebensform	LP
EF857 ⁽³⁾	Höchster schul./berufl. Abschluss: Lebenspartner in der Lebensform (ISCED 2011)	LP
EF858 ⁽³⁾	Arbeitszeittyp: Lebenspartner in der Lebensform	LP

Variable	Variablenlabel	Ebene
EF859 ⁽³⁾	Normale Arbeitszeit (je Woche: Stunden): Lebenspartner in der Lebensform	LP
EF860 ⁽³⁾	Grund geringerer Arbeitszeit (Berichtsw.): Lebenspartner in der Lebensform	LP
EF861 ⁽³⁾	Erwerbsunterbrechung (Berichtsw.): Lebenspartner in der Lebensform	LP
EF862	1. ausl. Staatsangehörigk.: Lebenspartner in der Lebensform	LP
EF866	Kinder: Anzahl in Familie (trad. Familienkonzept)	FAM
EF867	Kinder (unter 18 Jahren): Anzahl in Familie (trad. Familienkonzept)	FAM

Anmerkung: Die hinter dem Variablennamen hochgestellte Zahl in Klammern (bspw. ⁽³⁾) gibt die Anzahl der abweichenden Fälle zur Originalvariablen an (siehe Kapitel 3).

Quelle: FDZ der Statistischen Ämter des Bundes und der Länder, Mikrozensus Scientific Use File, 2014, eigene Berechnungen.

6.2 Syntax für Stata (Version 1.0)

```

* Dateiformat: Dos\Windows
* Encoding: Unicode (UTF-8)

* Stata Syntax zum GESIS Paper:
* "Einführung in die eigenständige Erstellung von Typisierungen
* am Beispiel des Mikrozensus Scientific Use Files 2014"
* Diese Syntax ist abgelegt unter: https://www.gesis.org/missy/materials/MZ/tools/typisierungen

/* Autor: Simon Börlin
   GESIS - Leibniz-Institut für Sozialwissenschaften
   German Microdata Lab (GML)
   Postfach 12 21 55
   68072 Mannheim
   https://www.gesis.org/gml/gml-home/
   E-Mail: simon.boerlin@gesis.org
*/

* Version 1.0 (17.12.2019)

* Daten öffnen:
use <<MZ-SUF-2014.dta>>, clear

=====
/*
Hilfsvariablen (siehe Abschnitt 3):

* Haushaltsbezugsperson (HHBP):
recode EF37 (1=1) (*=0), gen(HHBP)
label variable HHBP "Haushaltsbezugsperson"

* Lebensformbezugsperson (LFBP):
recode EF35 (1=1) (*=0), gen(LFBP)
label variable LFBP "Lebensformbezugsperson"

* Haupteinkommensbezieher (HEB):
gen HEB =1 if ((EF662>=1 & EF662<=4) & EF31==1)
recode HEB (.=0)
label variable HEB "Haupteinkommensbezieher"

* Lebenspartner der Bezugsperson der Lebensform (LP):
recode EF35 (2=1) (*=0), gen(LP)
label variable LP "Lebenspartner der Lebensformbezugsperson"

* Haushalts-, Lebensform-, und Familienidentifikatoren (idhxx, idlfx, idfamx)
* Siehe https://www.gesis.org/missy/materials/MZ/tools/panel2012
* Vorgehen orientiert sich an Herter-Eschweiler & Schimpl-Neimanns 2018.
* Vorbereitung der Identifikatoren:
tostring EF1 , generate(s1) format(%02.0f)
tostring EF3 , generate(s3) format(%06.0f)
tostring EF4 , generate(s4) format(%03.0f)
tostring EF5b , generate(s5b) format(%02.0f)
tostring EF12 , generate(s12) format(%01.0f)
tostring EF25 , generate(s27) format(%02.0f)
tostring EF27 , generate(s27) format(%02.0f)

* Verkettung der Ordnungsnummern zum alphanumerischen Indikator idhxx
generate idhxx = s1 + s3 + s4 + s5b + s12
lab var idhxx "ID Haushalt (EF1 EF3 EF4 EF5b EF12) - Querschnittsorientiert"

* Verkettung der Ordnungsnummern zum alphanumerischen Indikator idlfx
generate idlfx = s1 + s3 + s4 + s5b + s12 + s27
lab var idlfx "ID Lebensform (EF1 EF3 EF4 EF5b EF12 EF27) - Querschnittsorientiert"

* Verkettung der Ordnungsnummern zum alphanumerischen Indikator idfamx
generate idlfm = s1 + s3 + s4 + s5b + s12 + s27
lab var idlfm "ID Familie (EF1 EF3 EF4 EF5b EF12 EF25) - Querschnittsorientiert"

*/
=====

* Beginn der Typisierungen:
=====

* EF663: Personen im Haushalt: Anzahl
egen v663 = total (1), by (EF1 EF3 EF4 EF12 EF5b)

```

```
label variable v663 "Personen im Haushalt: Anzahl (Haushaltstypisierung)"
tab v663
recode v663 10/99 = 9
replace v663 = -8 if EF31==8
replace v663 = -7 if EF31==7
replace v663 = -1 if EF31==2
tab EF663 v663, m

* EF664: Erwerbstätige im Haushalt: Anzahl
egen v664 = sum (EF29==1), by (EF1 EF3 EF4 EF12 EF5b)
label var v664 "Anzahl Erwerbstätige im Haushalt"
tab v664, m
recode v664 9/59 = .
replace v664 = -8 if EF31==8
replace v664 = -7 if EF31==7
replace v664 = -1 if EF31==2
tab EF664 v664, m

* EF665: Erwerbslose (EU-Definition) im Haushalt: Anzahl
egen v665 = sum (EF29==2), by (EF1 EF3 EF4 EF12 EF5b)
label variable v665 "Anzahl Erwerbslose im Haushalt"
tab v665, m
replace v665 = -8 if EF31==8
replace v665 = -7 if EF31==7
replace v665 = -1 if EF31==2
tab EF665 v665, m

* EF666: Arbeitsuchender Nichterwerbspersonen im Haushalt: Anzahl
egen v666 = sum (EF29==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v666 "Anzahl Arbeitsuchende Nichterwerbspersonen im Haushalt"
tab v666, m
replace v666 = -8 if EF31==8
replace v666 = -7 if EF31==7
replace v666 = -1 if EF31==2
tab EF666 v666, m

* EF667: Einkommensbezieher im Haushalt: Anzahl
egen v667 = sum (EF436!=90 & EF436!=99), by (EF1 EF3 EF4 EF12 EF5b)
label variable v667 "Anzahl Einkommensbezieher im Haushalt"
tab v667, m
recode v667 10/23 = 9
replace v667 = -8 if EF31==8
replace v667 = -7 if EF31==7
replace v667 = -1 if EF31==2
tab EF667 v667, m

* EF668: Ausländer im Haushalt: Anzahl
egen v668 = sum (EF368==8), by (EF1 EF3 EF4 EF12 EF5b)
label variable v668 "Ausländer im Haushalt"
tab v668, m
recode v668 10/11 = 9
replace v668 = -8 if EF31==8
replace v668 = -7 if EF31==7
replace v668 = -1 if EF31==2
tab EF668 v668, m

* EF669: Kinder: Anzahl im Haushalt
egen v669 = sum (EF35 == 3 & EF31 == 1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v669 "Anzahl Kinder im Haushalt"
tab v669, m
recode v669 11/14 = 9
replace v669 = -8 if EF31==8
replace v669 = -7 if EF31==7
replace v669 = -1 if EF31==2
tab EF669 v669, m

* EF670: Kinder (unter 1 Jahr): Anzahl im Haushalt
egen v670 = sum (EF44<1 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v670 "Anzahl Kinder unter 1"
tab v670, m
replace v670 = -8 if EF31==8
replace v670 = -7 if EF31==7
replace v670 = -1 if EF31==2
tab EF670 v670, m

* EF671: Kinder (unter 2 Jahren): Anzahl im Haushalt
egen v671 = sum (EF44<2 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v671 "Anzahl Kinder unter 2"
tab v671, m
replace v671 = -8 if EF31==8
replace v671 = -7 if EF31==7
replace v671 = -1 if EF31==2
```

```
tab EF671 v671, m

* EF672: Kinder (unter 3 Jahren): Anzahl im Haushalt
egen v672 = sum (EF44<3 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v672 "Anzahl Kinder unter 3"
tab v672, m
replace v672 = -8 if EF31==8
replace v672 = -7 if EF31==7
replace v672 = -1 if EF31==2
tab EF672 v672, m

* EF673: Kinder (unter 4 Jahren): Anzahl im Haushalt
egen v673 = sum (EF44<4 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v673 "Anzahl Kinder unter 4"
tab v673, m
replace v673 = -8 if EF31==8
replace v673 = -7 if EF31==7
replace v673 = -1 if EF31==2
tab EF673 v673, m

* EF674: Kinder (unter 5 Jahren): Anzahl im Haushalt
egen v674 = sum (EF44<5 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v674 "Anzahl Kinder unter 5"
tab v674, m
replace v674 = -8 if EF31==8
replace v674 = -7 if EF31==7
replace v674 = -1 if EF31==2
tab EF674 v674, m

* EF675: Kinder (unter 6 Jahren): Anzahl im Haushalt
egen v675 = sum (EF44<6 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v675 "Anzahl Kinder unter 6"
tab v675, m
replace v675 = -8 if EF31==8
replace v675 = -7 if EF31==7
replace v675 = -1 if EF31==2
tab EF675 v675, m

* EF676: Kinder (unter 7 Jahren): Anzahl im Haushalt
egen v676 = sum (EF44<7 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v676 "Anzahl Kinder unter 7"
tab v676, m
replace v676 = -8 if EF31==8
replace v676 = -7 if EF31==7
replace v676 = -1 if EF31==2
tab EF676 v676, m

* EF677: Kinder (unter 8 Jahren): Anzahl im Haushalt
egen v677 = sum (EF44<8 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v677 "Anzahl Kinder unter 8"
tab v677, m
replace v677 = -8 if EF31==8
replace v677 = -7 if EF31==7
replace v677 = -1 if EF31==2
tab EF677 v677, m

* EF678: Kinder (unter 9 Jahren): Anzahl im Haushalt
egen v678 = sum (EF44<9 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v678 "Anzahl Kinder unter 9"
tab v678, m
replace v678 = -8 if EF31==8
replace v678 = -7 if EF31==7
replace v678 = -1 if EF31==2
tab EF678 v678, m

* EF679: Kinder (unter 10 Jahren): Anzahl im Haushalt
egen v679 = sum (EF44<10 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v679 "Anzahl Kinder unter 10"
tab v679, m
replace v679 = -8 if EF31==8
replace v679 = -7 if EF31==7
replace v679 = -1 if EF31==2
tab EF679 v679, m

* EF680: Kinder (unter 11 Jahren): Anzahl im Haushalt
egen v680 = sum (EF44<11 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v680 "Anzahl Kinder unter 11"
tab v680, m
replace v680 = -8 if EF31==8
replace v680 = -7 if EF31==7
replace v680 = -1 if EF31==2
tab EF680 v680, m
```

```
* EF681: Kinder (unter 12 Jahren): Anzahl im Haushalt
egen v681 = sum (EF44<12 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v681 "Anzahl Kinder unter 12"
tab v681, m
replace v681 = -8 if EF31==8
replace v681 = -7 if EF31==7
replace v681 = -1 if EF31==2
tab EF681 v681, m

* EF682: Kinder (unter 13 Jahren): Anzahl im Haushalt
egen v682 = sum (EF44<13 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v682 "Anzahl Kinder unter 13"
tab v682, m
replace v682 = -8 if EF31==8
replace v682 = -7 if EF31==7
replace v682 = -1 if EF31==2
tab EF682 v682, m

* EF683: Kinder (unter 14 Jahren): Anzahl im Haushalt
egen v683 = sum (EF44<14 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v683 "Anzahl Kinder unter 14"
tab v683, m
replace v683 = -8 if EF31==8
replace v683 = -7 if EF31==7
replace v683 = -1 if EF31==2
tab EF683 v683, m

* EF684: Kinder (unter 15 Jahren): Anzahl im Haushalt
egen v684 = sum (EF44<15 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v684 "Anzahl Kinder unter 15"
tab v684, m
replace v684 = -8 if EF31==8
replace v684 = -7 if EF31==7
replace v684 = -1 if EF31==2
tab EF684 v684, m

* EF685: Kinder (unter 16 Jahren): Anzahl im Haushalt
egen v685 = sum (EF44<16 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v685 "Anzahl Kinder unter 16"
tab v685, m
replace v685 = -8 if EF31==8
replace v685 = -7 if EF31==7
replace v685 = -1 if EF31==2
tab EF685 v685, m

* EF686: Kinder (unter 17 Jahren): Anzahl im Haushalt
egen v686 = sum (EF44<17 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v686 "Anzahl Kinder unter 17"
tab v686, m
replace v686 = -8 if EF31==8
replace v686 = -7 if EF31==7
replace v686 = -1 if EF31==2
tab EF686 v686, m

* EF687: Kinder (unter 18 Jahren): Anzahl im Haushalt
egen v687 = sum (EF44<18 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v687 "Anzahl Kinder unter 18"
tab v687, m
recode v687 10 = .
replace v687 = -8 if EF31==8
replace v687 = -7 if EF31==7
replace v687 = -1 if EF31==2
tab EF687 v687, m

* EF688: Kinder (unter 27 Jahren): Anzahl im Haushalt
egen v688 = sum (EF44<27 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v688 "Anzahl Kinder unter 27"
tab v688, m
recode v688 11/14 = .
replace v688 = -8 if EF31==8
replace v688 = -7 if EF31==7
replace v688 = -1 if EF31==2
tab EF688 v688, m

* EF689: Kinder (ab 18 Jahren): Anzahl im Haushalt
egen v689 = sum (EF44>=18 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v689 "Anzahl Kinder ab 18"
tab v689, m
recode v689 7/59 = .
replace v689 = -8 if EF31==8
replace v689 = -7 if EF31==7
```



```

replace v689 = -1   if EF31==2
tab EF689 v689, m

* EF690: Kinder (ab 27 Jahren): Anzahl im Haushalt
egen v690 = sum (EF44>=27 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b)
label variable v690 "Anzahl Kinder ab 27"
tab v690, m
recode v690 7/59 = .
replace v690 = -8   if EF31==8
replace v690 = -7   if EF31==7
replace v690 = -1   if EF31==2
tab EF690 v690, m

* EF691: Kinder 3-5 Jahre: Anzahl im Haushalt
egen v691 = sum (3 <= EF44 & 5 >= EF44 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v691 "Anzahl Kinder 3-5"
tab v691, m
recode v691 5/23 = .
replace v691 = -8   if EF31==8
replace v691 = -7   if EF31==7
replace v691 = -1   if EF31==2
tab EF691 v691, m

* EF692: Kinder 6-9 Jahre: Anzahl im Haushalt
egen v692 = sum (6 <= EF44 & 9 >= EF44 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v692 "Anzahl Kinder 6-9"
tab v692, m
recode v692 5/23 = .
replace v692 = -8   if EF31==8
replace v692 = -7   if EF31==7
replace v692 = -1   if EF31==2
tab EF692 v692, m

* EF693: Kinder 10-14 Jahre: Anzahl im Haushalt
egen v693 = sum (10 <= EF44 & 14 >= EF44 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v693 "Anzahl Kinder 10-14"
tab v693, m
recode v693 5/23 = .
replace v693 = -8   if EF31==8
replace v693 = -7   if EF31==7
replace v693 = -1   if EF31==2
tab EF693 v693, m

* EF694: Kinder 15-17 Jahre: Anzahl im Haushalt
egen v694 = sum (15 <= EF44 & 17 >= EF44 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v694 "Anzahl Kinder 15-17"
tab v694, m
recode v694 5/23 = .
replace v694 = -8   if EF31==8
replace v694 = -7   if EF31==7
replace v694 = -1   if EF31==2
tab EF694 v694, m

* EF695: Kinder 18-26 Jahre: Anzahl im Haushalt
egen v695 = sum (18 <= EF44 & 26 >= EF44 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v695 "Anzahl Kinder 18-26"
tab v695, m
replace v695 = -8   if EF31==8
replace v695 = -7   if EF31==7
replace v695 = -1   if EF31==2
tab EF695 v695, m

* EF696: Kinder unter 6 Jahren: Anzahl im Haushalt, Schüler
egen v696 = sum (EF44 < 6 & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1 EF3 EF4 EF12 EF5b
EF8 EF9)
label variable v696 "Anzahl Kinder > 6, Schüler"
tab v696, m
replace v696 = -8   if EF31==8
replace v696 = -7   if EF31==7
replace v696 = -1   if EF31==2
tab EF696 v696, m

* EF697: Kinder 6-9 Jahre: Anzahl im Haushalt, Schüler
egen v697 = sum (EF44>=6 & EF44<=9 & EF31==1 & EF35==3 & EF287>=1 & EF287<=2), by (EF1 EF3 EF4
EF12 EF5b EF8 EF9)
label variable v697 "Anzahl Kinder 6-9, Schüler"
tab v697, m
replace v697 = -8   if EF31==8
replace v697 = -7   if EF31==7
replace v697 = -1   if EF31==2
tab EF697 v697, m

```

```

* EF698: Kinder 10-14 Jahre: Anzahl im Haushalt, Schüler
egen v698 = sum ((10 <= EF44 & 14 >= EF44) & EF543==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v698 "Anzahl Kinder 10-14, Schüler"
tab v698, m
replace v698 = -8 if EF31==8
replace v698 = -7 if EF31==7
replace v698 = -1 if EF31==2
tab EF698 v698, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF699: Kinder 15-17 Jahre: Anzahl im Haushalt, Schüler
egen v699 = sum ((15 <= EF44 & 17 >= EF44) & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1
EF3 EF4 EF12 EF5b EF8 EF9)
label variable v699 "Anzahl Kinder 15-17, Schüler"
tab v699, m
replace v699 = -8 if EF31==8
replace v699 = -7 if EF31==7
replace v699 = -1 if EF31==2
tab EF699 v699, m

* EF700: Kinder 18-26 Jahre: Anzahl im Haushalt, Schüler
egen v700 = sum ((18 <= EF44 & 26 >= EF44) & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1
EF3 EF4 EF12 EF5b EF8 EF9)
label variable v700 "Anzahl Kinder 18-26, Schüler"
tab v700, m
replace v700 = -8 if EF31==8
replace v700 = -7 if EF31==7
replace v700 = -1 if EF31==2
tab EF700 v700, m

* EF701: Kinder ab 27 Jahre: Anzahl im Haushalt, Schüler
egen v701 = sum (27 <= EF44 & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1 EF3 EF4 EF12
EF5b EF8 EF9)
label variable v701 "Anzahl Kinder ab 27, Schüler"
tab v701, m
replace v701 = -8 if EF31==8
replace v701 = -7 if EF31==7
replace v701 = -1 if EF31==2
tab EF701 v701, m

* EF702: Personen ab 65: Anzahl im Haushalt
egen v702 = sum (EF44>=65 & EF31==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v702 "Anzahl Personen ab 65"
tab v702, m
recode v702 10/21 = 9
replace v702 = -8 if EF31==8
replace v702 = -7 if EF31==7
replace v702 = -1 if EF31==2
tab EF702 v702, m

* EF703: Personen ab 70: Anzahl im Haushalt
egen v703 = sum (EF44>=70 & EF31==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v703 "Anzahl Personen ab 70"
tab v703, m
recode v703 10/21 = 9
replace v703 = -8 if EF31==8
replace v703 = -7 if EF31==7
replace v703 = -1 if EF31==2
tab EF703 v703, m

* EF704: Personen ab 75: Anzahl im Haushalt
egen v704 = sum (EF44>=75 & EF31==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v704 "Anzahl Personen ab 75"
tab v704, m
recode v704 10/20 = 9
replace v704 = -8 if EF31==8
replace v704 = -7 if EF31==7
replace v704 = -1 if EF31==2
tab EF704 v704, m

* EF705: Personen ab 80: Anzahl im Haushalt
egen v705 = sum (EF44>=80 & EF31==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v705 "Anzahl Personen ab 80"
tab v705, m
recode v705 15 = 9
replace v705 = -8 if EF31==8
replace v705 = -7 if EF31==7
replace v705 = -1 if EF31==2
tab EF705 v705, m

* EF706: Personen ab 85: Anzahl im Haushalt
egen v706 = sum (EF44>=85 & EF31==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)

```

```

label variable v706 "Anzahl Personen ab 85"
tab v706, m
replace v706 = -8 if EF31==8
replace v706 = -7 if EF31==7
replace v706 = -1 if EF31==2
tab EF706 v706, m

* EF709: Personen am Nebenwohnsitz: Anzahl im Haushalt
egen v709 = sum (EF30==3 & EF31==1), by (EF1 EF3 EF4 EF12 EF5b EF8 EF9)
label variable v709 "Anzahl Personen am Nebenwohnsitz"
tab v709, m
replace v709 = -8 if EF31==8
replace v709 = -7 if EF31==7
replace v709 = -1 if EF31==2
tab EF709 v709, m

* EF711: Familien: Anzahl im Haushalt (Konzept der Lebensformen)
egen v711 = sum ((EF809>=1 & EF809<=4) & EF31 == 1 & EF35 == 1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v711 "Anzahl Familien im Haushalt"
replace v711 = -8 if EF31==8
replace v711 = -7 if EF31==7
replace v711 = -1 if EF31==2
tab EF711 v711, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF712: Ehepaare: Anzahl im Haushalt (Konzept der Lebensformen)
egen v712 = sum ((EF809==1 | EF809==5) & EF31==1 & EF35==1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v712 "Anzahl Ehepaare im Haushalt"
replace v712 = -8 if EF31==8
replace v712 = -7 if EF31==7
replace v712 = -1 if EF31==2
tab EF712 v712

* EF713: Nichteheliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebensformen)
egen v713 = sum ((EF809==2 | EF809==6) & EF31==1 & EF35==1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v713 "Anzahl Nichtehelicher Lebensgemeinschaften im Haushalt"
replace v713 = -8 if EF31==8
replace v713 = -7 if EF31==7
replace v713 = -1 if EF31==2
tab EF713 v713

* EF714: Gleichgeschlechtliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebensformen)
egen v714 = sum ((EF809==3 | EF809==7) & EF31==1 & EF35==1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v714 "Anzahl Gleichgeschlechtlicher Lebensgemeinschaften im Haushalt"
replace v714 = -8 if EF31==8
replace v714 = -7 if EF31==7
replace v714 = -1 if EF31==2
tab EF714 v714

* EF715: Alleinerziehende: Anzahl im Haushalt (Konzept der Lebensformen)
egen v715 = sum (EF809==4 & EF31==1 & EF35==1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v715 "Anzahl Alleinerziehende im Haushalt"
replace v715 = -8 if EF31==8
replace v715 = -7 if EF31==7
replace v715 = -1 if EF31==2
tab EF715 v715, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF716: Alleinstehende: Anzahl im Haushalt (Konzept der Lebensformen)
egen v716 = sum (EF809==8 & EF31==1 & EF35==1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v716 "Anzahl Alleinstehende im Haushalt"
recode v716 10/23 = 9
replace v716 = -8 if EF31==8
replace v716 = -7 if EF31==7
replace v716 = -1 if EF31==2
tab EF716 v716, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF719: Eingetragene Lebenspartnerschaft: Anzahl im Haushalt (Konzept der Lebensformen)
egen v719 = sum (EF811==7 & EF31==1 & EF35==1), by (EF1 EF3 EF4 EF12 EF5b)
label variable v719 "Eingetragene Lebenspartnerschaften im Haushalt"
replace v719 = -8 if EF31==8
replace v719 = -7 if EF31==7
replace v719 = -1 if EF31==2
tab EF719 v719

* EF727: 1. ausländische Staatsangehörigkeit: Haupteinkommensbezieher
egen v727 = sum (EF369*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v727 "1. ausländische Staatsangehörigkeit Haupteinkommensbezieher"
replace v727 = -8 if EF31==8
replace v727 = -7 if EF31==7

```

```

replace v727 = -1 if EF31==2
tab v727 if EF727!=v727

* EF731: Geschlecht: Haupteinkommensbezieher
egen v731= sum (EF46*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v731 "Geschlecht Haupteinkommensbezieher"
replace v731 = -8 if EF31==8
replace v731 = -7 if EF31==7
replace v731 = -1 if EF31==2
tab EF731 v731

* EF732: Alter: Haupteinkommensbezieher
egen v732= sum (EF44*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v732 "Alter Haupteinkommensbezieher"
replace v732 = -8 if EF31==8
replace v732 = -7 if EF31==7
replace v732 = -1 if EF31==2
tab v732 if EF732!=v732

* EF734: Staatsangehörigkeit: Haupteinkommensbezieher
egen v734= sum (EF371*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v734 "Staatsangehörigkeit Haupteinkommensbezieher"
replace v734 = -8 if EF31==8
replace v734 = -7 if EF31==7
replace v734 = -1 if EF31==2
tab EF734 v734

* mig heb: Migrationsstatus: Haupteinkommensbezieher
egen vmig_heb= sum (EF2001*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable vmig_heb "Migrationsstatus Haupteinkommensbezieher"
replace vmig_heb = -8 if EF31==8
replace vmig_heb = -7 if EF31==7
replace vmig_heb = -1 if EF31==2
tab vmig heb if mig heb!=vmig heb

* migs heb: Staatsangehörigkeit: Haupteinkommensbezieher
egen vmigs_heb= sum (EF2007*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable vmigs_heb "Staatsangehörigkeit Haupteinkommensbezieher"
replace vmigs_heb = -8 if EF31==8
replace vmigs_heb = -7 if EF31==7
replace vmigs_heb = -1 if EF31==2
tab vmigs_heb if migs heb!=vmigs_heb

* EF735: Familienstand erweitert: Haupteinkommensbezieher
egen v735= sum (EF765*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v735 "Familienstand Haupteinkommensbezieher"
replace v735 = -8 if EF31==8
replace v735 = -7 if EF31==7
replace v735 = -1 if EF31==2
tab EF735 v735

* EF736: Erwerbstyp: Haupteinkommensbezieher
egen v736= sum (EF29*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v736 "Erwerbstyp Haupteinkommensbezieher"
replace v736 = -8 if EF31==8
replace v736 = -7 if EF31==7
replace v736 = -1 if EF31==2
tab EF736 v736

* EF737: Wirtschaftszweig Betrieb: Haupteinkommensbezieher
egen v737= sum (EF137*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v737 "Wirtschaftszweig Betrieb Haupteinkommensbezieher"
replace v737 = -8 if EF31==8
replace v737 = -7 if EF31==7
replace v737 = -1 if EF31==2
tab v737 if EF737!=v737

* EF738: Beruf (Kldb 2010): Haupteinkommensbezieher
egen v738= sum (EF114*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v738 "Beruf (Kldb) Haupteinkommensbezieher"
replace v738 = -8 if EF31==8
replace v738 = -7 if EF31==7
replace v738 = -1 if EF31==2
replace v738 = -2 if EF738 == -2
tab v738 if EF738!=v738

* EF739: Beruf (ISCO-08): Haupteinkommensbezieher
egen v739= sum (EF541*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v739 "Beruf (ISCO) Haupteinkommensbezieher"
replace v739 = -8 if EF31==8
replace v739 = -7 if EF31==7
replace v739 = -1 if EF31==2

```

```

replace v739 = -2   if EF739 == -2
tab v739 if EF739!=v739

* EF740: Stellung im Beruf: Haupteinkommensbezieher
egen v740= sum (EF117*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v740 "Stellung im Beruf Haupteinkommensbezieher"
replace v740 = -8   if EF31== -8
replace v740 = -7   if EF31== -7
replace v740 = -1   if EF31== 2
tab v740 if EF740!=v740

* EF741: Überwiegender Lebensunterhalt: Haupteinkommensbezieher
egen v741= sum (EF401*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v741 "Überwiegender Lebensunterhalt Haupteinkommensbezieher"
replace v741 = -8   if EF31== -8
replace v741 = -7   if EF31== -7
replace v741 = -1   if EF31== 2
tab v741 if EF741!=v741

* EF742: Nettoeinkommen letzter Monat: Haupteinkommensbezieher
egen v742= sum (EF436*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v742 "Nettoeinkommen Haupteinkommensbezieher"
replace v742 = -8   if EF31== -8
replace v742 = -7   if EF31== -7
replace v742 = -1   if EF31== 2
tab v742 if EF742!=v742

* EF743: Höchster allg. Schulabschluss: Haupteinkommensbezieher
gen EF310n = EF310
recode EF310n (6 = 8) (9 = 7) (7 = 9)
recode EF310n (-5 = 6) if EF309==9
recode EF310n (-5 = 8)
tab EF310 EF310n , m

egen v743= sum (EF310n*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v743 "Höchster allg. Schulabschluss Haupteinkommensbezieher"
replace v743 = -8   if EF31== -8
replace v743 = -7   if EF31== -7
replace v743 = -1   if EF31== 2
tab EF743 v743

* EF744: Höchster berufl. Abschluss: Haupteinkommensbezieher
egen v744= sum (EF320*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v744 "Höchster berufl. Schulabschluss Haupteinkommensbezieher"
replace v744 = -8   if EF31== -8
replace v744 = -7   if EF31== -7
replace v744 = -1   if EF31== 2
tab v744 if EF744!=v744

* EF745: Höchster berufl./schul. Abschluss: Haupteinkommensbezieher
egen v745= sum (EF540*((EF662>=1 & EF662<=4) & EF31==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v745 "Höchster berufl./schul. Abschluss Haupteinkommensbezieher (ISCED)"
replace v745 = -8   if EF31== -8
replace v745 = -7   if EF31== -7
replace v745 = -1   if EF31== 2
tab EF745 v745

* EF753: Geschlecht: Haushaltsbezugsperson
egen v753 = sum (EF46*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v753 "Geschlecht Haushaltsbezugsperson"
replace v753 = -8   if EF31== -8
replace v753 = -7   if EF31== -7
replace v753 = -1   if EF31== 2
tab v753 EF753

* EF754: Alter: Haushaltsbezugsperson
egen v754 = sum (EF44*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v754 "Alter Haushaltsbezugsperson"
replace v754 = -8   if EF31== -8
replace v754 = -7   if EF31== -7
replace v754 = -1   if EF31== 2
tab v754 if EF754!=v754

* EF756: Staatsangehörigkeit: Haushaltsbezugsperson
egen v756 = sum (EF371*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v756 "Staatsangehörigkeit Haushaltsbezugsperson"
replace v756 = -8   if EF31== -8
replace v756 = -7   if EF31== -7
replace v756 = -1   if EF31== 2
tab v756 if EF756!=v756

* EF757: Familienstand: Haushaltsbezugsperson

```

```
egen v757 = sum (EF765*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v757 "Familienstand Haushaltsbezugsperson"
replace v757 = -8 if EF31==8
replace v757 = -7 if EF31==7
replace v757 = -1 if EF31==2
tab v757 EF757

* EF758: Erwerbstyp: Haushaltsbezugsperson
egen v758 = sum (EF29*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v758 "Erwerbstyp Haushaltsbezugsperson"
replace v758 = -8 if EF31==8
replace v758 = -7 if EF31==7
replace v758 = -1 if EF31==2
tab v758 EF758

* EF759: Wohnsitz: Haushaltsbezugsperson
egen v759 = sum (EF30*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v759 "Wohnsitz Haushaltsbezugsperson"
replace v759 = -8 if EF31==8
replace v759 = -7 if EF31==7
replace v759 = -1 if EF31==2
tab v759 EF759

* EF760: 1. ausl. Staatsangehörigkeit: Haushaltsbezugsperson
egen v760 = sum (EF369*(EF37==1)), by (EF1 EF3 EF4 EF12 EF5b)
label variable v760 "1. ausl. staatsangehörigkeit Haushaltsbezugsperson"
replace v760 = -8 if EF31==8
replace v760 = -7 if EF31==7
replace v760 = -1 if EF31==2
tab v760 if EF760!=v760

* EF770: Kinder: Anzahl in Lebensform
egen v770 = sum ((0 <= EF44 & 99 >= EF44) & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v770 "Anzahl Kinder in Lebensform"
recode v770 11/14 = 9
tab v770, m
replace v770 = -8 if EF31==8
replace v770 = -7 if EF31==7
replace v770 = -1 if EF31==2
tab EF770 v770, m

* EF771: Kinder unter 1: Anzahl in Lebensform
egen v771 = sum (EF44<1 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v771 "Anzahl Kinder unter 1 in Lebensform"
tab v771, m
replace v771 = -8 if EF31==8
replace v771 = -7 if EF31==7
replace v771 = -1 if EF31==2
tab EF771 v771, m

* EF772: Kinder unter 2: Anzahl in Lebensform
egen v772 = sum (EF44<2 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v772 "Anzahl Kinder unter 2 in Lebensform"
tab v772, m
replace v772 = -8 if EF31==8
replace v772 = -7 if EF31==7
replace v772 = -1 if EF31==2
tab EF772 v772, m

* EF773: Kinder unter 3: Anzahl in Lebensform
egen v773 = sum (EF44<3 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v773 "Anzahl Kinder unter 3 in Lebensform"
tab v773, m
replace v773 = -8 if EF31==8
replace v773 = -7 if EF31==7
replace v773 = -1 if EF31==2
tab EF773 v773, m

* EF774: Kinder unter 4: Anzahl in Lebensform
egen v774 = sum (EF44<4 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v774 "Anzahl Kinder unter 4 in Lebensform"
tab v774, m
replace v774 = -8 if EF31==8
replace v774 = -7 if EF31==7
replace v774 = -1 if EF31==2
tab EF774 v774, m

* EF775: Kinder unter 5: Anzahl in Lebensform
egen v775 = sum (EF44<5 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v775 "Anzahl Kinder unter 5 in Lebensform"
tab v775, m
replace v775 = -8 if EF31==8
```

```

replace v775 = -7   if EF31==7
replace v775 = -1   if EF31==2
tab EF775 v775, m

* EF776: Kinder unter 6: Anzahl in Lebensform
egen v776 = sum (EF44<6 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v776 "Anzahl Kinder unter 6 in Lebensform"
tab v776, m
replace v776 = -8   if EF31==8
replace v776 = -7   if EF31==7
replace v776 = -1   if EF31==2
tab EF776 v776, m

* EF777: Kinder unter 7: Anzahl in Lebensform
egen v777 = sum (EF44<7 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v777 "Anzahl Kinder unter 7 in Lebensform"
tab v777, m
replace v777 = -8   if EF31==8
replace v777 = -7   if EF31==7
replace v777 = -1   if EF31==2
tab EF777 v777, m

* EF778: Kinder unter 8: Anzahl in Lebensform
egen v778 = sum (EF44<8 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v778 "Anzahl Kinder unter 8 in Lebensform"
tab v778, m
replace v778 = -8   if EF31==8
replace v778 = -7   if EF31==7
replace v778 = -1   if EF31==2
tab EF778 v778, m

* EF779: Kinder unter 9: Anzahl in Lebensform
egen v779 = sum (EF44<9 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v779 "Anzahl Kinder unter 9 in Lebensform"
tab v779, m
replace v779 = -8   if EF31==8
replace v779 = -7   if EF31==7
replace v779 = -1   if EF31==2
tab EF779 v779, m

* EF780: Kinder unter 10: Anzahl in Lebensform
egen v780 = sum (EF44<10 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v780 "Anzahl Kinder unter 10 in Lebensform"
tab v780, m
replace v780 = -8   if EF31==8
replace v780 = -7   if EF31==7
replace v780 = -1   if EF31==2
tab EF780 v780, m

* EF781: Kinder unter 11: Anzahl in Lebensform
egen v781 = sum (EF44<11 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v781 "Anzahl Kinder unter 11 in Lebensform"
tab v781, m
replace v781 = -8   if EF31==8
replace v781 = -7   if EF31==7
replace v781 = -1   if EF31==2
tab EF781 v781, m

* EF782: Kinder unter 12: Anzahl in Lebensform
egen v782 = sum (EF44<12 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v782 "Anzahl Kinder unter 12 in Lebensform"
tab v782, m
replace v782 = -8   if EF31==8
replace v782 = -7   if EF31==7
replace v782 = -1   if EF31==2
tab EF782 v782, m

* EF783: Kinder unter 13: Anzahl in Lebensform
egen v783 = sum (EF44<13 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v783 "Anzahl Kinder unter 13 in Lebensform"
tab v783, m
replace v783 = -8   if EF31==8
replace v783 = -7   if EF31==7
replace v783 = -1   if EF31==2
tab EF783 v783, m

* EF784: Kinder unter 14: Anzahl in Lebensform
egen v784 = sum (EF44<14 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v784 "Anzahl Kinder unter 14 in Lebensform"
tab v784, m
replace v784 = -8   if EF31==8
replace v784 = -7   if EF31==7

```

```
replace v784 = -1 if EF31==2
tab EF784 v784, m

* EF785: Kinder unter 15: Anzahl in Lebensform
egen v785 = sum (EF44<15 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v785 "Anzahl Kinder unter 15 in Lebensform"
tab v785, m
replace v785 = -8 if EF31==8
replace v785 = -7 if EF31==7
replace v785 = -1 if EF31==2
tab EF785 v785, m

* EF786: Kinder unter 16: Anzahl in Lebensform
egen v786 = sum (EF44<16 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v786 "Anzahl Kinder unter 16 in Lebensform"
tab v786, m
replace v786 = -8 if EF31==8
replace v786 = -7 if EF31==7
replace v786 = -1 if EF31==2
tab EF786 v786, m

* EF787: Kinder unter 17: Anzahl in Lebensform
egen v787 = sum (EF44<17 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v787 "Anzahl Kinder unter 17 in Lebensform"
tab v787, m
replace v787 = -8 if EF31==8
replace v787 = -7 if EF31==7
replace v787 = -1 if EF31==2
tab EF787 v787, m

* EF788: Kinder unter 18: Anzahl in Lebensform
egen v788 = sum (EF44<18 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v788 "Anzahl Kinder unter 18 in Lebensform"
tab v788, m
recode v788 10 = 9
replace v788 = -8 if EF31==8
replace v788 = -7 if EF31==7
replace v788 = -1 if EF31==2
tab EF788 v788, m

* EF789: Kinder unter 27: Anzahl in Lebensform
egen v789 = sum (EF44<27 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v789 "Anzahl Kinder unter 27 in Lebensform"
tab v789, m
recode v789 11/14 = 9
replace v789 = -8 if EF31==8
replace v789 = -7 if EF31==7
replace v789 = -1 if EF31==2
tab EF789 v789, m

* EF790: Kinder ab 18: Anzahl in Lebensform
egen v790 = sum (EF44>=18 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v790 "Anzahl Kinder ab 18 in Lebensform"
tab v790, m
replace v790 = -8 if EF31==8
replace v790 = -7 if EF31==7
replace v790 = -1 if EF31==2
tab EF790 v790, m

* EF791: Kinder ab 27: Anzahl in Lebensform
egen v791 = sum (EF44>=27 & EF31==1 & EF35==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v791 "Anzahl Kinder ab 27 in Lebensform"
tab v791, m
replace v791 = -8 if EF31==8
replace v791 = -7 if EF31==7
replace v791 = -1 if EF31==2
tab EF791 v791, m

* EF792: Kinder 3-5 Jahre: Anzahl in Lebensform
egen v792 = sum ((3 <= EF44 & 5 >= EF44) & (EF31==1 & EF35==3)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v792 "Anzahl Kinder 2-5 in Lebensform"
tab v792, m
replace v792 = -8 if EF31==8
replace v792 = -7 if EF31==7
replace v792 = -1 if EF31==2
tab EF792 v792, m

* EF793: Kinder 6-9 Jahre: Anzahl in Lebensform
egen v793 = sum ((6 <= EF44 & 9 >= EF44) & (EF31==1 & EF35==3)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v793 "Anzahl Kinder 6-9 in Lebensform"
tab v793, m
replace v793 = -8 if EF31==8
```



```

replace v793 = -7   if EF31==7
replace v793 = -1   if EF31==2
tab EF793 v793, m

* EF794: Kinder 10-14 Jahre: Anzahl in Lebensform
egen v794 = sum ((10 <= EF44 & 14 >= EF44) & (EF31==1 & EF35==3)) , by (EF1 EF3 EF4 EF12 EF5b
EF27)
label variable v794 "Anzahl Kinder 10-14 in Lebensform"
tab v794, m
replace v794 = -8   if EF31==8
replace v794 = -7   if EF31==7
replace v794 = -1   if EF31==2
tab EF794 v794, m

* EF795: Kinder 15-17 Jahre: Anzahl in Lebensform
egen v795 = sum ((15 <= EF44 & 17 >= EF44) & (EF31==1 & EF35==3)) , by (EF1 EF3 EF4 EF12 EF5b
EF27)
label variable v795 "Anzahl Kinder 15-17 in Lebensform"
tab v795, m
replace v795 = -8   if EF31==8
replace v795 = -7   if EF31==7
replace v795 = -1   if EF31==2
tab EF795 v795, m

* EF796: Kinder 18-26 Jahre: Anzahl in Lebensform
egen v796 = sum ((18 <= EF44 & 26 >= EF44) & (EF31==1 & EF35==3)) , by (EF1 EF3 EF4 EF12 EF5b
EF27)
label variable v796 "Anzahl Kinder 18-26 in Lebensform"
tab v796, m
replace v796 = -8   if EF31==8
replace v796 = -7   if EF31==7
replace v796 = -1   if EF31==2
tab EF796 v796, m

* EF797: Kinder unter 6 Jahren: Anzahl in Lebensform, Schüler
egen v797 = sum (EF44 < 6 & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1 EF3 EF4 EF12 EF5b
EF27)
label variable v797 "Anzahl Kinder unter 6 in Lebensform, Schüler"
tab v797, m
replace v797 = -8   if EF31==8
replace v797 = -7   if EF31==7
replace v797 = -1   if EF31==2
tab EF797 v797, m

* EF798: Kinder 6-9 Jahre: Anzahl in Lebensform, Schüler
egen v798 = sum ((6 <= EF44 & 9 >= EF44) & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1
EF3 EF4 EF12 EF5b EF27)
label variable v798 "Anzahl Kinder 6-9 in Lebensform, Schüler"
tab v798, m
replace v798 = -8   if EF31==8
replace v798 = -7   if EF31==7
replace v798 = -1   if EF31==2
tab EF798 v798, m

* EF799: Anzahl in Lebensform: Kinder 10-14 Jahre, Schüler
egen v799 = sum ((10 <= EF44 & 14 >= EF44) & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1
EF3 EF4 EF12 EF5b EF27)
label variable v799 "Anzahl Kinder 10-14 in Lebensform, Schüler"
tab v799, m
replace v799 = -8   if EF31==8
replace v799 = -7   if EF31==7
replace v799 = -1   if EF31==2
tab EF799 v799, m

* EF800: Anzahl in Lebensform: Kinder 15-17 Jahre, Schüler
egen v800 = sum ((15 <= EF44 & 17 >= EF44) & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1
EF3 EF4 EF12 EF5b EF27)
label variable v800 "Anzahl Kinder 15-17 in Lebensform, Schüler"
replace v800 = -8   if EF31==8
replace v800 = -7   if EF31==7
replace v800 = -1   if EF31==2
tab EF800 v800, m

* EF801: Anzahl in Lebensform: Kinder 18-26 Jahre, Schüler
egen v801 = sum ((18 <= EF44 & 26 >= EF44) & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1
EF3 EF4 EF12 EF5b EF27)
label variable v801 "Anzahl Kinder 18-26 in Lebensform, Schüler"
tab v801, m
replace v801 = -8   if EF31==8
replace v801 = -7   if EF31==7
replace v801 = -1   if EF31==2
tab EF801 v801, m

```

```

* EF802: Anzahl in Lebensform: Kinder ab 27 Jahren, Schüler
egen v802 = sum (EF44>=27 & EF31==1 & EF35==3 & (EF287==1 | EF287==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v802 "Anzahl Kinder ab 27 in Lebensform, Schüler"
tab v802, m
replace v802 = -8 if EF31==8
replace v802 = -7 if EF31==7
replace v802 = -1 if EF31==2
tab EF802 v802, m

* EF803: Alter jüngstes Kind in Lebensform
egen v803 = max (EF44*(EF35==3 & EF805==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v803 "Alter jüngstes Kind"
replace v803 = -8 if EF31==8
replace v803 = -7 if EF31==7
replace v803 = -1 if EF31==2
recode v803 (0 = -5) if EF770==0
tab EF803 if EF803!=v803

* EF804: Alter: Ältestes Kind in Lebensform
generate v804 = .
lab var v804 "Alter: Ältestes Kind in Lebensform"
egen v804_1 = max (EF44 * (EF35==3) if EF35>=1 & EF35<=3, by(EF1 EF3 EF4 EF5b EF12 EF27)
* ältestes Kind ist unter 1 Jahr alt
egen v804_0 = total(EF44==0 & EF35==3) if EF35>=1 & EF35<=3, by(EF1 EF3 EF4 EF5b EF12 EF27)

replace v804 = 0 if v804_0>=1 & v804_0<=3
replace v804 = v804_1 if v804_1>=1 & v804_1<=95
replace v804 = -5 if v804_0==0 & v804_1==0
tab EF804 v804

* EF812: Personen am Nebenwohnsitz: Anzahl in Lebensform
egen v812 = sum (EF30==3), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v812 "Anzahl Personen Nebenwohnsitz"
tab v812, m
replace v812 = -8 if EF31==8
replace v812 = -7 if EF31==7
replace v812 = -1 if EF31==2
tab EF812 v812, m

* EF819: Geschlecht: Bezugsperson der Lebensform
egen v819 = sum (EF46*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v819 "Geschlecht Bezugsperson Lebensform"
tab v819, m
replace v819 = -8 if EF31==8
replace v819 = -7 if EF31==7
replace v819 = -1 if EF31==2
tab EF819 v819, m

* EF820: Alter: Bezugsperson der Lebensform
egen v820 = sum (EF44*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v820 "Alter Bezugsperson Lebensform"
tab v820, m
replace v820 = -8 if EF31==8
replace v820 = -7 if EF31==7
replace v820 = -1 if EF31==2
tab v820 if EF820!=v820

* EF822: Staatsangehörigkeit: Bezugsperson der Lebensform
egen v822 = sum (EF371*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v822 "Staatsangehörigkeit Bezugsperson Lebensform"
tab v822, m
replace v822 = -8 if EF31==8
replace v822 = -7 if EF31==7
replace v822 = -1 if EF31==2
tab v822 if EF822!=v822

* mig_lf1: Migrationsstatus: Bezugsperson der Lebensform
egen vmig_lf1= sum (EF2001*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable vmig_lf "Migrationsstatus Bezugsperson Lebensform"
replace vmig_lf = -8 if EF31==8
replace vmig_lf = -7 if EF31==7
replace vmig_lf = -1 if EF31==2
tab vmig_lf1 if mig_lf1!=vmig_lf1

* migs_lf: Staatsangehörigkeit: Bezugsperson der Lebensform
egen vmigs_lf= sum (EF2007*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable vmigs_lf "Staatsangehörigkeit Bezugsperson Lebensform"
replace vmigs_lf = -8 if EF31==8
replace vmigs_lf = -7 if EF31==7
replace vmigs_lf = -1 if EF31==2

```

```

tab vmigs lf if migs lf1!=vmigs lf

* EF823: Familienstand: Bezugsperson der Lebensform
egen v823 = sum (EF765*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v823 "Familienstand Bezugsperson Lebensform"
tab v823, m
replace v823 = -8 if EF31==8
replace v823 = -7 if EF31==7
replace v823 = -1 if EF31==2
tab v823 EF823

* EF824: Erwerbstyp: Bezugsperson der Lebensform
egen v824 = sum (EF29*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v824 "Erwerbstyp Bezugsperson Lebensform"
tab v824, m
replace v824 = -8 if EF31==8
replace v824 = -7 if EF31==7
replace v824 = -1 if EF31==2
tab v824 EF824

* EF825: Wirtschaftszweig: Bezugsperson der Lebensform
egen v825 = sum (EF137*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v825 "Wirtschaftszweig Bezugsperson Lebensform"
tab v825, m
replace v825 = -8 if EF31==8
replace v825 = -7 if EF31==7
replace v825 = -1 if EF31==2
tab v825 if EF825!=v825

* EF826: Beruf (Kldb 2010): Bezugsperson der Lebensform
egen v826 = sum (EF114*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v826 "Beruf (Kldb 2010) Bezugsperson Lebensform"
replace v826 = -8 if EF31==8
replace v826 = -7 if EF31==7
replace v826 = -1 if EF31==2
replace v826 = -2 if EF826 == -2
tab v826 if EF826!=v826

* EF827: Beruf (ISCO-08): Bezugsperson der Lebensform
egen v827 = sum (EF541*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v827 "Beruf (ISCO-08) Bezugsperson Lebensform"
replace v827 = -8 if EF31==8
replace v827 = -7 if EF31==7
replace v827 = -1 if EF31==2
replace v827 = -2 if EF827 == -2
tab v827 if EF827!=v827

* EF828: Stellung im Beruf: Bezugsperson der Lebensform
egen v828 = sum (EF117*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v828 "Stellung im Beruf Bezugsperson Lebensform"
tab v828, m
replace v828 = -8 if EF31==8
replace v828 = -7 if EF31==7
replace v828 = -1 if EF31==2
tab v828 if EF828!=v828

* EF829: Überwiegender Lebensunterhalt: Bezugsperson der Lebensform
egen v829 = sum (EF401*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v829 "Überwiegender Lebensunterhalt Bezugsperson Lebensform"
tab v829, m
replace v829 = -8 if EF31==8
replace v829 = -7 if EF31==7
replace v829 = -1 if EF31==2
tab v829 if EF829!=v829

* EF830: Nettoeinkommen: Bezugsperson der Lebensform
egen v830 = sum (EF436*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v830 "Nettoeinkommen Bezugsperson Lebensform"
tab v830, m
replace v830 = -8 if EF31==8
replace v830 = -7 if EF31==7
replace v830 = -1 if EF31==2
tab v830 if EF830!=v830

* EF831: Höchster schul. Abschluss: Bezugsperson der Lebensform
gen EF310n1 = EF310
recode EF310n1 (6 = 8) (9 = 7) (7 = 9)
recode EF310n1 (-5 = 6) if EF309==9
recode EF310n1 (-5 = 8)
recode EF310n1 (-3 = 8)

egen v831 = sum (EF310n1*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)

```

```

label variable v831 "Höchster schul. Abschluss Bezugsperson Lebensform"
replace v831 = -8 if EF31==8
replace v831 = -7 if EF31==7
replace v831 = -1 if EF31==2
tab EF831 v831

* EF832: Höchster berufl. Abschluss: Bezugsperson der Lebensform
egen v832= sum (EF320*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v832 "Höchster berufl. Schulabschluss Bezugsperson Lebensform"
replace v832 = -8 if EF31==8
replace v832 = -7 if EF31==7
replace v832 = -1 if EF31==2
recode v832 (-3 = 88)
tab v832 if EF832!=v832

* EF833: Höchster berufl./ schul. Abschluss: Bezugsperson der Lebensform (ISCED 2011)
egen v833= sum (EF340*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v833 "Höchster berufl./schul. Abschluss Bezugsperson der Lebensform (ISCED)"
replace v833 = -8 if EF31==8
replace v833 = -7 if EF31==7
replace v833 = -1 if EF31==2
tab EF833 v833

* EF834: Arbeitszeittyp: Bezugsperson der Lebensform
gen EF130n = EF130
recode EF130n (-5 = 8)
recode EF130n (1/9 = 8) if EF129==1

egen v834 = sum (EF130n*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v834 "Arbeitszeittyp Bezugsperson der Lebensform"
tab v834, m
replace v834 = -8 if EF31==8
replace v834 = -7 if EF31==7
replace v834 = -1 if EF31==2
tab EF834 v834

* EF835: Normale Arbeitszeit (je Woche: Stunden): Bezugsperson der Lebensform
egen v835 = sum (EF131*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v835 "Normale Arbeitszeit Bezugsperson Lebensform"
replace v835 = -8 if EF31==8
replace v835 = -7 if EF31==7
replace v835 = -1 if EF31==2
recode v835 (-1 = -2)
tab EF835 if EF835!=v835

* EF836: Grund geringer Arbeitszeit: Bezugsperson der Lebensform
gen EF147n = EF147
recode EF147n (3 = 2) if EF29==1 & EF146==2
recode EF147n (4 = 3) if EF29==1 & EF146==2
recode EF147n (17 = 5) if EF29==1 & EF146==2
recode EF147n (8 = 6) if EF29==1 & EF146==2
recode EF147n (16 = 7) if EF29==1 & EF146==2
recode EF147n (1 2 6 9 10 11 12 13 14 15 18 19 99 = 8) if EF29==1 & EF146==2 & EF147!=3 &
EF147!=4 & EF147!=8 & EF147!=16 & EF147!=17
recode EF147n (-5 = 1) if EF29==1 & EF146!=2

egen v836= sum (EF147n*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v836 "Grund geringer Arbeitszeit Bezugsperson der Lebensform"
replace v836 = -8 if EF31==8
replace v836 = -7 if EF31==7
replace v836 = -1 if EF31==2
tab EF836 v836, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF837: Erwerbsunterbrechung (Berichtsw.): Bezugsperson der Lebensform
gen EF85nl = EF85
recode EF85nl (1/11=1) if EF29==1
recode EF85nl (-5=2) if EF29==1

egen v837= sum (EF85nl*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v837 "1. ausländische Staatsangehörigkeit Bezugsperson Lebensform"
replace v837 = -8 if EF31==8
replace v837 = -7 if EF31==7
replace v837 = -1 if EF31==2
recode v837 (3/10=-2)
tab EF837 v837, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF838: 1. ausländische Staatsangehörigkeit: Bezugsperson der Lebensform
egen v838= sum (EF369*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v838 "1. ausländische Staatsangehörigkeit Bezugsperson Lebensform"
replace v838 = -8 if EF31==8

```

```

replace v838 = -7   if EF31==7
replace v838 = -1   if EF31==2
tab v838 if EF838!=v838

* EF843: Geschlecht: Bezugsperson der Lebensform
egen v843= sum (EF843*(EF35==1)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v843 "Geschlecht Bezugsperson Lebensform"
replace v843 = -8   if EF31==8
replace v843 = -7   if EF31==7
replace v843 = -1   if EF31==2
tab v843 if EF843!=v843

* EF844: Alter: Lebenspartner der Bezugsperson der Lebensform
egen v844= sum (EF44*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v844 "Alter Lebenspartner Bezugsperson Lebensform"
replace v844 = -8   if EF31==8
replace v844 = -7   if EF31==7
replace v844 = -1   if EF31==2
recode v844 (0 = -5)
tab v844 if EF844!=v844
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF846: Staatsangehörigkeit: Lebenspartner in Lebensform
egen v846= sum (EF371*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v846 "Staatsangehörigkeit Lebenspartner in Lebensform"
replace v846 = -8   if EF31==8
replace v846 = -7   if EF31==7
replace v846 = -1   if EF31==2
recode v846 (0 = -5)
tab v846 if EF846!=v846
* ACHTUNG: Abweichungen zur Originaltypisierung

* mig_lf2: Migrationsstatus: Lebenspartner in Lebensform
egen vmig_lf2= sum (EF2001*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable vmig_lf2 "Migrationsstatus Lebenspartner in Lebensform"
replace vmig_lf2 = EF762 if vmig_lf2==0 & EF762<0

egen LP_DE = sum (EF2001==0 & (EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
replace vmig_lf2 = -5 if vmig_lf2==0 & LP_DE==0
tab vmig_lf2 mig_lf2 if vmig_lf2!=mig_lf2

* migs_lf2: Staatsangehörigkeit: Lebenspartner Lebensform
egen vmigs_lf2= sum (EF2007*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable vmigs_lf2 "Staatsangehörigkeit Lebenspartner in Lebensform"
replace vmigs_lf2 = EF762 if vmigs_lf2==0 & EF762<0

egen LP_DE2 = sum (EF2007==0 & (EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
replace vmigs_lf2 = -5 if vmigs_lf2==0 & LP_DE2==0
tab vmigs_lf2 migs_lf2 if vmigs_lf2!=migs_lf2

* EF847: Familienstand: Lebenspartner in Lebensform
egen v847= sum (EF765*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v847 "Familienstand Lebenspartner in Lebensform"
replace v847 = -8   if EF31==8
replace v847 = -7   if EF31==7
replace v847 = -1   if EF31==2
recode v847 (0 = -5)
tab EF847 v847, m
tab v847 if EF847!=v847
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF848: Erwerbstyp: Lebenspartner in Lebensform
egen v848= sum (EF29*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v848 "Erwerbstyp Lebenspartner in Lebensform"
replace v848 = -8   if EF31==8
replace v848 = -7   if EF31==7
replace v848 = -1   if EF31==2
recode v848 (0 = -5)
tab EF848 v848
tab v848 if EF848!=v848
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF849: Wirtschaftszweig: Lebenspartner in Lebensform
egen v849= sum (EF137*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v849 "Wirtschaftszweig Lebenspartner in Lebensform"
replace v849 = -8   if EF31==8
replace v849 = -7   if EF31==7
replace v849 = -1   if EF31==2
recode v849 (0 = -5)
tab v849 if EF849!=v849
* ACHTUNG: Abweichungen zur Originaltypisierung

```

```

* EF850: Beruf (Kldb 2010): Lebenspartner in Lebensform
egen v850= sum (EF114*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v850 "Beruf (Kldb) Lebenspartner in Lebensform"
replace v850 = -8 if EF31==8
replace v850 = -7 if EF31==7
replace v850 = -1 if EF31==2
replace v850 = -2 if EF850 == -2
recode v850 (0 = -5)
tab v850 if EF850!=v850

* EF851: Beruf (ISCO-08): Lebenspartner in Lebensform
egen v851= sum (EF541*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v851 "Beruf (ISCO) Lebenspartner in Lebensform"
replace v851 = -8 if EF31==8
replace v851 = -7 if EF31==7
replace v851 = -1 if EF31==2
replace v851 = -2 if EF851 == -2
recode v851 (0 = -5)
recode v851 (7314 = 7319)
recode v851 (8151 = 8153)
recode v851 (-5 = 3313) if EF33!=-5
tab v851 if EF851!=v851
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF852: Stellung im Beruf: Lebenspartner in Lebensform
egen v852= sum (EF117*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v852 "Stellung im Beruf Lebenspartner in Lebensform"
replace v852 = -8 if EF31==8
replace v852 = -7 if EF31==7
replace v852 = -1 if EF31==2
recode v852 (0 = -5)
tab EF852 v852
tab v852 if EF852!=v852
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF853: Überwiegender Lebensunterhalt: Lebenspartner in Lebensform
egen v853= sum (EF401*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v853 "Überwiegender Lebensunterhalt Lebenspartner in Lebensform"
replace v853 = -8 if EF31==8
replace v853 = -7 if EF31==7
replace v853 = -1 if EF31==2
recode v853 (0 = -5)
tab v853 if EF853!=v853
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF854: Nettoeinkommen letzter Monat: Lebenspartner in Lebensform
egen v854= sum (EF436*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v854 "Nettoeinkommen Lebenspartner in Lebensform"
replace v854 = -8 if EF31==8
replace v854 = -7 if EF31==7
replace v854 = -1 if EF31==2
recode v854 (0 = -5)
tab v854 if EF854!=v854
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF855: Höchster allg. Schulabschluss: Lebenspartner in Lebensform
gen EF310lp = EF310
recode EF310lp (6 = 8) (9 = 7) (7 = 9)
recode EF310lp (-5 = 6) if EF309==9
recode EF310lp (-5 = 8)
tab EF310 EF310lp , m

egen v855= sum (EF310lp*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v855 "Höchster allg. Schulabschluss Lebenspartner in Lebensform"
replace v855 = -8 if EF31==8
replace v855 = -7 if EF31==7
replace v855 = -1 if EF31==2
recode v855 (0 = -5)
tab EF855 v855
tab v855 if EF855!=v855
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF856: Höchster berufl. Abschluss: Lebenspartner in Lebensform
egen v856= sum (EF320*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v856 "Höchster berufl. Schulabschluss Lebenspartner in Lebensform"
replace v856 = -8 if EF31==8
replace v856 = -7 if EF31==7
replace v856 = -1 if EF31==2
recode v856 (0 = -5)
tab EF856 v856
tab v856 if EF856!=v856
* ACHTUNG: Abweichungen zur Originaltypisierung

```

```

* EF857: Höchster berufl./ schul. Abschluss: Lebenspartner in Lebensform (ISCED 2011)
egen v857= sum (EF540*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v857 "Höchster berufl./schul. Abschluss Lebenspartner in Lebensform (ISCED)"
replace v857 = -8 if EF31==8
replace v857 = -7 if EF31==7
replace v857 = -1 if EF31==2
recode v857 (0 = -5)
tab EF857 if EF857!=v857
tab EF857 v857
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF858: Arbeitszeittyp: Lebenspartner in Lebensform
gen EF130nlp = EF130
recode EF130nlp (-5 = 8)
recode EF130nlp (1/9 = 8) if EF129==1

egen v858 = sum (EF130nlp*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v858 "Arbeitszeittyp Lebenspartner in Lebensform"
replace v858 = -8 if EF31==8
replace v858 = -7 if EF31==7
replace v858 = -1 if EF31==2
* kein Lebenspartner in Lebensform -> v858=-5
egen LPinLF = sum(EF35==2), by(EF1 EF3 EF4 EF12 EF5b EF27)
recode v858 (0=-5) if LPinLF==0
drop LPinLF
tab EF858 v858
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF859: Normale Arbeitszeit (je Woche: Stunden): Lebenspartner in Lebensform
egen v859 = sum (EF131*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v859 "Normale Arbeitszeit Lebenspartner in Lebensform"
replace v859 = -8 if EF31==8
replace v859 = -7 if EF31==7
replace v859 = -1 if EF31==2
recode v859 (0 = -5)
tab v859 if EF859!=v859
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF860: Grund geringer Arbeitszeit: Lebenspartner in der Lebensform
gen EF147np = EF147
recode EF147np (3 = 2) if EF29==1 & EF146==2
recode EF147np (4 = 3) if EF29==1 & EF146==2
recode EF147np (17 = 5) if EF29==1 & EF146==2
recode EF147np (8 = 6) if EF29==1 & EF146==2
recode EF147np (16 = 7) if EF29==1 & EF146==2
recode EF147np (1 2 6 9 10 11 12 13 14 15 18 19 99 = 8) if EF29==1 & EF146==2 & EF147!=3 &
EF147!=4 & EF147!=8 & EF147!=16 & EF147!=17
recode EF147np (-5 = 1) if EF29==1 & EF146!=2

egen v860= sum (EF147np*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v860 "Grund geringer Arbeitszeit Lebenspartner in Lebensform"
replace v860 = -8 if EF31==8
replace v860 = -7 if EF31==7
replace v860 = -1 if EF31==2
recode v860 (0 = -5)
tab EF860 v860
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF861: Erwerbsunterbrechung (Berichtsw.): Lebenspartner in Lebensform
generate v861 = .
lab var v861 "Erwerbsunterbrechung (Berichtsw.): Lebenspartner in der Lebensform"
egen v861_1 = total(EF31==1 & EF35==2 & EF29==1 & EF85>=1 & EF85<=11) if EF35>=1 & EF35<=3,
by(EF1 EF3 EF4 EF5b EF12 EF27)
egen v861_2 = total(EF31==1 & EF35==2 & EF29==1 & EF85==5) if EF35>=1 & EF35<=3, by(EF1 EF3 EF4
EF5b EF12 EF27)
egen v861 m2 = total(EF31==1 & EF35==2 & EF29!=1) if EF35>=1 & EF35<=3, by(EF1 EF3 EF4 EF5b EF12
EF27)

replace v861 = 1 if v861_1==1 & EF31==1
replace v861 = 2 if v861_2==1 & EF31==1
replace v861 = -2 if v861 m2==1 & EF31==1
* kein Lebenspartner in Lebensform -> v861=-5
egen v861_m5 = total(EF35==2) if EF35>=1 & EF35<=3, by(EF1 EF3 EF4 EF5b EF12 EF27)
replace v861 = -5 if v861_m5==0 & EF31==1
* Ungültige Werte: EF35<1
replace v861 = EF31 if EF31<1 & v861==. // Ausfallsätze
replace v861 = -1 if EF31==2 & v861==. // Bev. in Gemeinschaftsunterk.
tab EF861 v861, m
* ACHTUNG: Abweichungen zur Originaltypisierung

* EF862: 1. ausländische Staatsangehörigkeit: Lebenspartner in Lebensform

```

```
egen v862= sum (EF369*(EF35==2)), by (EF1 EF3 EF4 EF12 EF5b EF27)
label variable v862 "1. ausländische Staatsangehörigkeit Lebenspartner in Lebensform"
replace v862 = -8 if EF31==8
replace v862 = -7 if EF31==7
replace v862 = -1 if EF31==2
recode v862 (0 = -5)
tab v862 if EF862!=v862

* EF866: Kinder: Anzahl in Familie (trad. Familienkonzept)
egen v866= sum (EF31==1 & EF34==3), by (EF1 EF3 EF4 EF12 EF5b EF25)
label variable v866 "Anzahl Kinder in Familie (trad. Familienkonzept)"
replace v866 = -8 if EF31==8
replace v866 = -7 if EF31==7
replace v866 = -1 if EF31==2
recode v866 (9/14 = 9)
tab EF866 v866

* EF867: Kinder unter 18 Jahren: Anzahl in Familie (trad. Familienkonzept)
egen v867= sum (EF31==1 & EF34==3 & EF44<18), by (EF1 EF3 EF4 EF12 EF5b EF25)
label variable v867 "Anzahl Kinder unter 18 Jahren in Familie (trad. Familienkonzept)"
replace v867 = -8 if EF31==8
replace v867 = -7 if EF31==7
replace v867 = -1 if EF31==2
recode v867 (10=9)
tab EF867 v867

* Ende der Syntax
```


6.3 Syntax für SPSS (Version 1.0)

```

* Encoding: ANSI (windows-1252) .
* IBM SPSS Statistics 24.0.0.1 .

* SPSS Syntax zum GESIS Paper:
  "Einführung in die eigenständige Erstellung von Typisierungen
  am Beispiel des Mikrozensus Scientific Use Files 2014"
* Diese Syntax ist abgelegt unter: https://www.gesis.org/missy/materials/MZ/tools/typisierungen

* Autor: Simon Börlin
  GESIS - Leibniz-Institut für Sozialwissenschaften
  German Microdata Lab (GML)
  Postfach 12 21 55
  68072 Mannheim
  https://www.gesis.org/gml/gml-home/
  E-Mail: simon.boerlin@gesis.org .

* Version 1.0 (17.12.2019).

* Daten öffnen:
GET file = "<<<mz2014.sav>>>".

MISSING VALUES ALL().
* Hinweis: Die User-Missing Values werden am Ende dieser Datei definiert.

*=====
* Hilfsvariablen (siehe Abschnitt 3):.
* Haushaltsbezugsperson.
compute HHBP = 0.
if (EF37 = 1) HHBP = 1.
var lab HHBP "Haushaltsbezugsperson".

* Bezugsperson in Lebensform.
compute LFBP = 0.
if (EF35 = 1) LFBP = 1.
var lab LFBP "Lebensformbezugsperson".

* Haupteinkommensbezieher.
compute HEB = 0.
if ((EF662>=1 & EF662<=4) & EF31=1) HEB = 1.
var lab HEB "Haupteinkommensbezieher".

* Lebenspartner der Bezugsperson der Lebensform.
compute LP = 0.
if (EF35 = 2) LP = 1.
var lab LP "Lebenspartner der Bezugsperson der Lebensform".

* Haushalts-, Lebensform-, und Familienidentifikatoren (idhxx, idlfx, idfamx).
* Siehe https://www.gesis.org/missy/materials/MZ/tools/panel2012 .
* Vorgehen orientiert sich an Herter-Eschweiler & Schimpl-Neimanns 2018.
* Vorbereitung der Identifikatoren: .
string s1 (A2) s3 (A6) s4 (A3) s5b(A2) s12 (A1) s25 (A2) s27 (A2).
compute s1 = string(EF1,n2).
compute s3 = string(EF3,n6).
compute s12 = string(EF12,n1).
* Führende Null umsetzen.
do if (EF4>=0).
  compute s4 = string(EF4,n3).
else if (EF4=-7).
  compute s4 = "0-7".
end if.
do if (EF5b>=0).
  compute s5b = string(EF5b,n2).
else if (EF5b=-7).
  compute s5b = "-7".
else if (EF5b = -8).
  compute s5b = "-8".
end if.
do if (EF25>=0).
  compute s25 = string(EF25,n2).
else if (EF25=-7).
  compute s25 = "-7".
else if (EF25=-8).
  compute s25 = "-8".
end if.
do if (EF27>=0).
  compute s27 = string(EF27,n2).

```

```

else if (EF27=-7).
  compute s27 = "-7".
else if (EF27=-8).
  compute s27 = "-8".
end if.
execute.

* Verkettung der Ordnungsnummern zum alphanumerischen Indikator idhxx.
string idhxx (A14).
compute idhxx = concat(char.rpad(s1,2),char.rpad(s3,6),char.rpad(s4,3),
  char.rpad(s5b,2),char.rpad(s12,1)).
var lab idhxx "ID Haushalt (EF1 EF3 EF4 EF5b EF12) - Querschnittsorientiert".

* Verkettung der Ordnungsnummern zum alphanumerischen Indikator idlfx.
string idlfx (A16).
compute idlfx = concat(char.rpad(s1,2),char.rpad(s3,6),char.rpad(s4,3),
  char.rpad(s5b,2),char.rpad(s12,1),char.rpad(s27,2)).
var lab idlfx "ID Lebensform (EF1 EF3 EF4 EF5b EF12 EF27) - Querschnittsorientiert".

* Verkettung der Ordnungsnummern zum alphanumerischen Indikator idfamx.
string idfamx (A16).
compute idfamx = concat(char.rpad(s1,2),char.rpad(s3,6),char.rpad(s4,3),
  char.rpad(s5b,2),char.rpad(s12,1),char.rpad(s25,2)).
var lab idfamx "ID Familie (EF1 EF3 EF4 EF5b EF12 EF25) - Querschnittsorientiert".

*=====.

*=====.
* Beginn der Typisierungen: .
*=====.

* EF663: Personen im Haushalt: Anzahl.
temporary.
compute EF663k = 1.
aggregate outfile = * mode = addvariables
  /break EF1 EF3 EF4 EF5b EF12 /v663 = SUM (EF663k).
var lab v663 "Personen im Haushalt: Anzahl (Haushaltstypisierung)".
if(v663>=10) v663 = 9.
if(EF31=2) v663 = -1.
if(EF31=-7) v663 = -7.
if(EF31=-8) v663 = -8.
crosstabs EF663 by v663.

* EF664: Erwerbstätige im Haushalt: Anzahl.
temporary.
compute EF29k = 0.
if (EF29 = 1) EF29k = 1.

aggregate outfile = * mode = addvariables
  /break EF1 EF3 EF4 EF5b EF12 /v664 = SUM (EF29k).
var lab v664 "Anzahl Erwerbstätige im Haushalt".

if(EF31=-8) v664 = -8.
if(EF31=-7) v664 = -7.
if(EF31=2) v664 = -1.
fre v664.

* EF665: Erwerbslose (EU-Definition) im Haushalt: Anzahl.
temporary.
compute EF29k = 0.
if (EF29 = 2) EF29k = 1.

aggregate outfile = * mode = addvariables
  /break EF1 EF3 EF4 EF5b EF12 /v665 = SUM (EF29k).
var lab v665 "Anzahl Erwerbslose im Haushalt".

if(EF31=-8) v665 = -8.
if(EF31=-7) v665 = -7.
if(EF31=2) v665 = -1.
fre v665.

* EF666: Arbeitsuchender Nichterwerbspersonen im Haushalt: Anzahl.
temporary.
compute EF29k = 0.
if (EF29 = 3) EF29k = 1.

aggregate outfile = * mode = addvariables
  /break EF1 EF3 EF4 EF5b EF12 /v666 = SUM (EF29k).
var lab v666 "Anzahl Arbeitsuchende Nichterwerbspersonen im Haushalt".

if(EF31=-8) v666 = -8.

```

```

if(EF31=-7) v666 = -7.
if(EF31=2) v666 = -1.
fre v666.

* EF667: Einkommensbezieher im Haushalt: Anzahl.

temporary.
compute EF667k = 0.
if (EF436<90) EF667k = 1.

aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v667 = SUM (EF667k).
var lab v667 "Anzahl Einkommensbezieher im Haushalt".
if(EF31=-8) v667 = -8.
if(EF31=-7) v667 = -7.
if(EF31=2) v667 = -1.
if(v667>9) v667 =9.
fre v667.

* EF668: Ausländer im Haushalt: Anzahl.

temporary.
compute EF368k = 0.
if (EF368 = 8) EF368k = 1.

aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v668 = SUM (EF368k).
var lab v668 "Ausländer im Haushalt".
if(EF31=-8) v668 = -8.
if(EF31=-7) v668 = -7.
if(EF31=2) v668 = -1.
if(v668>9) v668 = 9.
fre v668.

* EF669: Kinder: Anzahl im Haushalt.

temporary.
compute EF669k = 0.
if (EF35=3 & EF31=1) EF669k = 1.

aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v669 = SUM (EF669k).
var lab v669 "Anzahl Kinder im Haushalt".
if(EF31=-8) v669 = -8.
if(EF31=-7) v669 = -7.
if(EF31=2) v669 = -1.
if(v669>9) v669 = 9.
fre v669.

* EF670: Kinder (unter 1 Jahr): Anzahl im Haushalt.

temporary.
compute EF670k = 0.
if (EF35=3 & EF31=1 & EF44<1) EF670k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v670 = SUM (EF670k).
var lab v670 "Anzahl Kinder unter 1".
if(EF31=-8) v670 = -8.
if(EF31=-7) v670 = -7.
if(EF31=2) v670 = -1.
fre v670.

* EF671: Kinder (unter 2 Jahren): Anzahl im Haushalt:.

temporary.
compute EF671k = 0.
if (EF35=3 & EF31=1 & EF44<2) EF671k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v671 = SUM (EF671k).
var lab v671 "Anzahl Kinder unter 2".
if(EF31=-8) v671 = -8.
if(EF31=-7) v671 = -7.
if(EF31=2) v671 = -1.
fre v671.

* EF672: Kinder (unter 3 Jahren): Anzahl im Haushalt.

temporary.
compute EF672k = 0.
if (EF35=3 & EF31=1 & EF44<3) EF672k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v672 = SUM (EF672k).
var lab v672 "Anzahl Kinder unter 3".
if(EF31=-8) v672 = -8.
if(EF31=-7) v672 = -7.

```

```
if(EF31=2) v672 = -1.
fre v672.

* EF673: Kinder (unter 4 Jahren): Anzahl im Haushalt.
temporary.
compute EF673k = 0.
if (EF35=3 & EF31=1 & EF44<4) EF673k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v673 = SUM (EF673k).
var lab v673 "Anzahl Kinder unter 4".
if(EF31=-8) v673 = -8.
if(EF31=-7) v673 = -7.
if(EF31=2) v673 = -1.
fre v673.

* EF674: Kinder (unter 5 Jahren): Anzahl im Haushalt.
temporary.
compute EF674k = 0.
if (EF35=3 & EF31=1 & EF44<5) EF674k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v674 = SUM (EF674k).
var lab v674 "Anzahl Kinder unter 5".
if(EF31=-8) v674 = -8.
if(EF31=-7) v674 = -7.
if(EF31=2) v674 = -1.
fre v674.

* EF675: Kinder (unter 6 Jahren): Anzahl im Haushalt.
temporary.
compute EF675k = 0.
if (EF35=3 & EF31=1 & EF44<6) EF675k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v675 = SUM (EF675k).
var lab v675 "Anzahl Kinder unter 6".
if(EF31=-8) v675 = -8.
if(EF31=-7) v675 = -7.
if(EF31=2) v675 = -1.
fre v675.

* EF676: Kinder (unter 7 Jahren): Anzahl im Haushalt.
temporary.
compute EF676k = 0.
if (EF35=3 & EF31=1 & EF44<7) EF676k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v676 = SUM (EF676k).
var lab v676 "Anzahl Kinder unter 7".
if(EF31=-8) v676 = -8.
if(EF31=-7) v676 = -7.
if(EF31=2) v676 = -1.
fre v676.

* EF677: Kinder (unter 8 Jahren): Anzahl im Haushalt.
temporary.
compute EF677k = 0.
if (EF35=3 & EF31=1 & EF44<8) EF677k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v677 = SUM (EF677k).
var lab v677 "Anzahl Kinder unter 8".
if(EF31=-8) v677 = -8.
if(EF31=-7) v677 = -7.
if(EF31=2) v677 = -1.
fre v677.

* EF678: Kinder (unter 9 Jahren): Anzahl im Haushalt.
temporary.
compute EF678k = 0.
if (EF35=3 & EF31=1 & EF44<9) EF678k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v678 = SUM (EF678k).
var lab v678 "Anzahl Kinder unter 9".
if(EF31=-8) v678 = -8.
if(EF31=-7) v678 = -7.
if(EF31=2) v678 = -1.
fre v678.

* EF679: Kinder (unter 10 Jahren): Anzahl im Haushalt.
temporary.
compute EF679k = 0.
if (EF35=3 & EF31=1 & EF44<10) EF679k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v679 = SUM (EF679k).
```

```
var lab v679 "Anzahl Kinder unter 10".
if(EF31=-8) v679 = -8.
if(EF31=-7) v679 = -7.
if(EF31=2) v679 = -1.
fre v679.

* EF680: Kinder (unter 11 Jahren): Anzahl im Haushalt.
temporary.
compute EF680k = 0.
if (EF35=3 & EF31=1 & EF44<11) EF680k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v680 = SUM (EF680k).
var lab v680 "Anzahl Kinder unter 11".
if(EF31=-8) v680 = -8.
if(EF31=-7) v680 = -7.
if(EF31=2) v680 = -1.
fre v680.

* EF681: Kinder (unter 12 Jahren): Anzahl im Haushalt.
temporary.
compute EF681k = 0.
if (EF35=3 & EF31=1 & EF44<12) EF681k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v681 = SUM (EF681k).
var lab v681 "Anzahl Kinder unter 12".
if(EF31=-8) v681 = -8.
if(EF31=-7) v681 = -7.
if(EF31=2) v681 = -1.
fre v681.

* EF682: Kinder (unter 13 Jahren): Anzahl im Haushalt.
temporary.
compute EF682k = 0.
if (EF35=3 & EF31=1 & EF44<13) EF682k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v682 = SUM (EF682k).
var lab v682 "Anzahl Kinder unter 13".
if(EF31=-8) v682 = -8.
if(EF31=-7) v682 = -7.
if(EF31=2) v682 = -1.
fre v682.

* EF683: Kinder (unter 14 Jahren): Anzahl im Haushalt .
temporary.
compute EF683k = 0.
if (EF35=3 & EF31=1 & EF44<14) EF683k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v683 = SUM (EF683k).
var lab v683 "Anzahl Kinder unter 14".
if(EF31=-8) v683 = -8.
if(EF31=-7) v683 = -7.
if(EF31=2) v683 = -1.
fre v683.

* EF684: Kinder (unter 15 Jahren): Anzahl im Haushalt .
temporary.
compute EF684k = 0.
if (EF35=3 & EF31=1 & EF44<15) EF684k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v684 = SUM (EF684k).
var lab v684 "Anzahl Kinder unter 15".
if(EF31=-8) v684 = -8.
if(EF31=-7) v684 = -7.
if(EF31=2) v684 = -1.
fre v684.

* EF685: Kinder (unter 16 Jahren): Anzahl im Haushalt .
temporary.
compute EF685k = 0.
if (EF35=3 & EF31=1 & EF44<16) EF685k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v685 = SUM (EF685k).
var lab v685 "Anzahl Kinder unter 16".
if(EF31=-8) v685 = -8.
if(EF31=-7) v685 = -7.
if(EF31=2) v685 = -1.
fre v685.

* EF686: Kinder (unter 17 Jahren): Anzahl im Haushalt.
temporary.
compute EF686k = 0.
if (EF35=3 & EF31=1 & EF44<17) EF686k = 1.
```

```
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v686 = SUM (EF686k).
var lab v686 "Anzahl Kinder unter 17".
if(EF31=-8) v686 = -8.
if(EF31=-7) v686 = -7.
if(EF31=2) v686 = -1.
fre v686.

* EF687: Kinder (unter 18 Jahren): Anzahl im Haushalt .
temporary.
compute EF687k = 0.
if (EF35=3 & EF31=1 & EF44<18) EF687k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v687 = SUM (EF687k).
var lab v687 "Anzahl Kinder unter 18".
if(v687 = 10) v687 = $SYSMIS.
if(EF31=-8) v687 = -8.
if(EF31=-7) v687 = -7.
if(EF31=2) v687 = -1.
fre v687.

* EF688: Kinder (unter 27 Jahren): Anzahl im Haushalt.
temporary.
compute EF688k = 0.
if (EF35=3 & EF31=1 & EF44<27) EF688k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v688 = SUM (EF688k).
var lab v688 "Anzahl Kinder unter 27".
if(v688 > 10) v688 = $SYSMIS.
if(EF31=-8) v688 = -8.
if(EF31=-7) v688 = -7.
if(EF31=2) v688 = -1.
fre v688.

* EF689: Kinder (ab 18 Jahren): Anzahl im Haushalt.
temporary.
compute EF689k = 0.
if (EF35=3 & EF31=1 & EF44>17) EF689k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v689 = SUM (EF689k).
var lab v689 "Anzahl Kinder ab 18".
if(v689 > 6) v689 = $SYSMIS.
if(EF31=-8) v689 = -8.
if(EF31=-7) v689 = -7.
if(EF31=2) v689 = -1.
fre v689.

* EF690: Kinder (ab 27 Jahren): Anzahl im Haushalt.
temporary.
compute EF690k = 0.
if (EF35=3 & EF31=1 & EF44>26) EF690k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v690 = SUM (EF690k).
var lab v690 "Anzahl Kinder ab 27".
if(v690 > 6) v690 = $SYSMIS.
if(EF31=-8) v690 = -8.
if(EF31=-7) v690 = -7.
if(EF31=2) v690 = -1.
fre v690.

* EF691: Kinder 3-5 Jahre: Anzahl im Haushalt.
temporary.
compute EF691k = 0.
if (EF44>2 & EF44<6 & EF31=1 & EF35=3) EF691k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v691 = SUM (EF691k).
var lab v691 "Anzahl Kinder 3-5".
if(v691 > 4) v691 = $SYSMIS.
if(EF31=-8) v691 = -8.
if(EF31=-7) v691 = -7.
if(EF31=2) v691 = -1.
fre v691.

* EF692: Kinder 6-9 Jahre: Anzahl im Haushalt.
temporary.
compute EF692k = 0.
if (EF44>5 & EF44<10 & EF31=1 & EF35=3) EF692k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v692 = SUM (EF692k).
var lab v692 "Anzahl Kinder 6-9".
if(v692 > 4) v692 = $SYSMIS.
if(EF31=-8) v692 = -8.
```

```

if(EF31=-7) v692 = -7.
if(EF31=2) v692 = -1.
fre v692.

* EF693: Kinder 10-14 Jahre: Anzahl im Haushalt.
temporary.
compute EF693k = 0.
if (EF44>9 & EF44<15 & EF31=1 & EF35=3) EF693k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v693 = SUM (EF693k).
var lab v693 "Anzahl Kinder 10-14".
if(v693 > 4) v693 = $SYSMIS.
if(EF31=-8) v693 = -8.
if(EF31=-7) v693 = -7.
if(EF31=2) v693 = -1.
fre v693.

* EF694: Kinder 15-17 Jahre: Anzahl im Haushalt.
temporary.
compute EF694k = 0.
if (EF44>14 & EF44<18 & EF31=1 & EF35=3) EF694k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v694 = SUM (EF694k).
var lab v694 "Anzahl Kinder 15-17".
if(v694 > 4) v694 = $SYSMIS.
if(EF31=-8) v694 = -8.
if(EF31=-7) v694 = -7.
if(EF31=2) v694 = -1.
fre v694.

* EF695: Kinder 18-26 Jahre: Anzahl im Haushalt.
temporary.
compute EF695k = 0.
if (EF44>17 & EF44<27 & EF31=1 & EF35=3) EF695k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v695 = SUM (EF695k).
var lab v695 "Anzahl Kinder 18-26".
if(EF31=-8) v695 = -8.
if(EF31=-7) v695 = -7.
if(EF31=2) v695 = -1.
fre v695.

* EF696: Kinder unter 6 Jahren: Anzahl im Haushalt, Schüler.
temporary.
compute EF696k = 0.
if (EF44<6 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF696k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v696 = SUM (EF696k).
var lab v696 "Anzahl Kinder > 6, Schüler".
if(EF31=-8) v696 = -8.
if(EF31=-7) v696 = -7.
if(EF31=2) v696 = -1.
fre v696.

* EF697: Kinder 6-9 Jahre: Anzahl im Haushalt, Schüler.
temporary.
compute EF697k = 0.
if (EF44>5 & EF44<10 & EF31=1 & EF35=3 & EF287>0 & EF287<3) EF697k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v697 = SUM (EF697k).
var lab v697 "Anzahl Kinder 6-9, Schüler".
if(EF31=-8) v697 = -8.
if(EF31=-7) v697 = -7.
if(EF31=2) v697 = -1.
fre v697.

* EF698: Kinder 10-14 Jahre: Anzahl im Haushalt, Schüler.
temporary.
compute EF698k = 0.
if (EF44>9 & EF44<15 & EF543=1) EF698k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v698 = SUM (EF698k).
var lab v698 "Anzahl Kinder 10-14, Schüler".
if(EF31=-8) v698 = -8.
if(EF31=-7) v698 = -7.
if(EF31=2) v698 = -1.
crosstabs EF698 by v698.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF699: Kinder 15-17 Jahre: Anzahl im Haushalt, Schüler.
temporary.
compute EF699k = 0.

```

```
if (EF44>14 & EF44<18 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF699k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v699 = SUM (EF699k).
var lab v699 "Anzahl Kinder 15-17, Schüler".
if(EF31=-8) v699 = -8.
if(EF31=-7) v699 = -7.
if(EF31=2) v699 = -1.
fre v699.

* EF700: Kinder 18-26 Jahre: Anzahl im Haushalt, Schüler.
temporary.
compute EF700k = 0.
if (EF44>17 & EF44<27 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF700k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v700 = SUM (EF700k).
var lab v700 "Anzahl Kinder 18-26, Schüler".
if(EF31=-8) v700 = -8.
if(EF31=-7) v700 = -7.
if(EF31=2) v700 = -1.
fre v700.

* EF701: Kinder ab 27 Jahre: Anzahl im Haushalt, Schüler.
temporary.
compute EF701k = 0.
if (EF44>26 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF701k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v701 = SUM (EF701k).
var lab v701 "Anzahl Kinder ab 27, Schüler".
if(EF31=-8) v701 = -8.
if(EF31=-7) v701 = -7.
if(EF31=2) v701 = -1.
fre v701.

* EF702: Personen ab 65: Anzahl im Haushalt.
temporary.
compute EF702k = 0.
if (EF44>64 & EF31=1) EF702k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v702 = SUM (EF702k).
var lab v702 "Anzahl Personen ab 65".
if(v702>9) v702 = 9.
if(EF31=-8) v702 = -8.
if(EF31=-7) v702 = -7.
if(EF31=2) v702 = -1.
fre v702.

* EF703: Personen ab 70: Anzahl im Haushalt.
temporary.
compute EF703k = 0.
if (EF44>69 & EF31=1) EF703k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v703 = SUM (EF703k).
var lab v703 "Anzahl Personen ab 70".
if(v703>9) v703 = 9.
if(EF31=-8) v703 = -8.
if(EF31=-7) v703 = -7.
if(EF31=2) v703 = -1.
fre v703.

* EF704: Personen ab 75: Anzahl im Haushalt.
temporary.
compute EF704k = 0.
if (EF44>74 & EF31=1) EF704k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v704 = SUM (EF704k).
var lab v704 "Anzahl Personen ab 75".
if(v704>9) v704 = 9.
if(EF31=-8) v704 = -8.
if(EF31=-7) v704 = -7.
if(EF31=2) v704 = -1.
fre v704.

* EF705: Personen ab 80: Anzahl im Haushalt.
temporary.
compute EF705k = 0.
if (EF44>79 & EF31=1) EF705k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v705 = SUM (EF705k).
var lab v705 "Anzahl Personen ab 80".
if(v705>9) v705 = 9.
if(EF31=-8) v705 = -8.
if(EF31=-7) v705 = -7.
```



```

if(EF31=2) v705 = -1.
fre v705.

* EF706: Personen ab 85: Anzahl im Haushalt.
temporary.
compute EF706k = 0.
if (EF44>84 & EF31=1) EF706k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v706 = SUM (EF706k).
var lab v706 "Anzahl Personen ab 85".
if(v706>9) v706 = 9.
if(EF31=-8) v706 = -8.
if(EF31=-7) v706 = -7.
if(EF31=2) v706 = -1.
fre v706.

* EF709: Personen am Nebenwohnsitz: Anzahl im Haushalt.
temporary.
compute EF709k = 0.
if (EF30=3 & EF31=1) EF709k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF8 EF9 EF12 /v709 = SUM (EF709k).
var lab v709 "Anzahl Personen am Nebenwohnsitz".
if(EF31=-8) v709 = -8.
if(EF31=-7) v709 = -7.
if(EF31=2) v709 = -1.
fre v709.

* EF711: Familien: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF711k = 0.
if (EF809>0 & EF809<5 & EF31=1 & EF35=1) EF711k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v711 = SUM (EF711k).
var lab v711 "Anzahl Familien im Haushalt".
if(EF31=-8) v711 = -8.
if(EF31=-7) v711 = -7.
if(EF31=2) v711 = -1.
crosstabs EF711 by v711.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF712: Ehepaare: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF712k = 0.
if ((EF809=1 OR EF809=5) & EF31=1 & EF35=1) EF712k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v712 = SUM (EF712k).
var lab v712 "Anzahl Ehepaare im Haushalt".
if(EF31=-8) v712 = -8.
if(EF31=-7) v712 = -7.
if(EF31=2) v712 = -1.
fre v712.

* EF713: Nichteheliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF713k = 0.
if ((EF809=2 OR EF809=6) & EF31=1 & EF35=1) EF713k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v713 = SUM (EF713k).
var lab v713 "Anzahl Nichtehelicher Lebensgemeinschaften im Haushalt".
if(EF31=-8) v713 = -8.
if(EF31=-7) v713 = -7.
if(EF31=2) v713 = -1.
fre v713.

* EF714: Gleichgeschlechtliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF714k = 0.
if ((EF809=3 OR EF809=7) & EF31=1 & EF35=1) EF714k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v714 = SUM (EF714k).
var lab v714 "Anzahl Gleichgeschlechtlicher Lebensgemeinschaften im Haushalt".
if(EF31=-8) v714 = -8.
if(EF31=-7) v714 = -7.
if(EF31=2) v714 = -1.
fre v714.

* EF715: Alleinerziehende: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF715k = 0.
if (EF809=4 & EF31=1 & EF35=1) EF715k = 1.
aggregate outfile = * mode = addvariables

```

```

/break EF1 EF3 EF4 EF5b EF12 /v715 = SUM (EF715k).
var lab v715 "Anzahl Gleichgeschlechtlicher Lebensgemeinschaften im Haushalt".
if(EF31=-8) v715 = -8.
if(EF31=-7) v715 = -7.
if(EF31=2) v715 = -1.
crosstabs EF715 by v715.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF716: Alleinstehende: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF716k = 0.
if (EF809=8 & EF31=1 & EF35=1) EF716k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v716 = SUM (EF716k).
var lab v716 "Anzahl Alleinstehende im Haushalt".
if(v716>9) v716=9.
if(EF31=-8) v716 = -8.
if(EF31=-7) v716 = -7.
if(EF31=2) v716 = -1.
crosstabs EF716 by v716.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF719: Eingetragene Lebenspartnerschaft: Anzahl im Haushalt (Konzept der Lebensformen).
temporary.
compute EF719k = 0.
if (EF811=7 & EF31=1 & EF35=1) EF719k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v719 = SUM (EF719k).
var lab v719 "Eingetragene Lebenspartnerschaften im Haushalt".
if(EF31=-8) v719 = -8.
if(EF31=-7) v719 = -7.
if(EF31=2) v719 = -1.
fre v719.

* EF727: 1. ausländische Staatsangehörigkeit: Haupteinkommensbezieher.
temporary.
compute EF727k = EF369 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v727 = SUM (EF727k).
var lab v727 "1. ausländische Staatsangehörigkeit Haupteinkommensbezieher".
if(EF31=-8) v727 = -8.
if(EF31=-7) v727 = -7.
if(EF31=2) v727 = -1.
fre v727.

* EF731: Geschlecht: Haupteinkommensbezieher.
temporary.
compute EF731k = EF46 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v731 = SUM (EF731k).
var lab v731 "Geschlecht Haupteinkommensbezieher".
if(EF31=-8) v731 = -8.
if(EF31=-7) v731 = -7.
if(EF31=2) v731 = -1.
fre v731.

* EF732: Alter: Haupteinkommensbezieher.
temporary.
compute EF732k = EF44 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v732 = SUM (EF732k).
var lab v732 "Alter Haupteinkommensbezieher".
if(EF31=-8) v732 = -8.
if(EF31=-7) v732 = -7.
if(EF31=2) v732 = -1.
fre v732.

* EF734: Staatsangehörigkeit: Haupteinkommensbezieher.
temporary.
compute EF734k = EF371 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v734 = SUM (EF734k).
var lab v734 "Staatsangehörigkeit Haupteinkommensbezieher".
if(EF31=-8) v734 = -8.
if(EF31=-7) v734 = -7.
if(EF31=2) v734 = -1.
fre v734.

* mig heb: Migrationsstatus: Haupteinkommensbezieher.
temporary.
compute mig_heb_k = EF2001 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables

```

```

/break EF1 EF3 EF4 EF5b EF12 /vmig heb = SUM (mig heb k).
var lab vmig heb "Migrationsstatus Haupteinkommensbezieher".
if(EF31=-8) vmig heb = -8.
if(EF31=-7) vmig heb = -7.
if(EF31=2) vmig heb = -1.
fre vmig heb.

* migs heb: Staatsangehörigkeit: Haupteinkommensbezieher.
temporary.
compute migs heb k = EF2007 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /vmigs heb = SUM (migs heb k).
var lab vmigs heb "Staatsangehörigkeit Haupteinkommensbezieher".
if(EF31=-8) vmigs heb = -8.
if(EF31=-7) vmigs heb = -7.
if(EF31=2) vmigs heb = -1.
fre vmigs heb.

* EF735: Familienstand erweitert: Haupteinkommensbezieher.
temporary.
compute EF735k = EF765 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v735 = SUM (EF735k).
var lab v735 "Familienstand Haupteinkommensbezieher".
if(EF31=-8) v735 = -8.
if(EF31=-7) v735 = -7.
if(EF31=2) v735 = -1.
fre v735.

* EF736: Erwerbstyp: Haupteinkommensbezieher.
temporary.
compute EF736k = EF29 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v736 = SUM (EF736k).
var lab v736 "Erwerbstyp Haupteinkommensbezieher".
if(EF31=-8) v736 = -8.
if(EF31=-7) v736 = -7.
if(EF31=2) v736 = -1.
fre v736.

* EF737: Wirtschaftszweig Betrieb: Haupteinkommensbezieher.
temporary.
compute EF737k = EF137 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v737 = SUM (EF737k).
var lab v737 "Wirtschaftszweig Betrieb Haupteinkommensbezieher".
if(EF31=-8) v737 = -8.
if(EF31=-7) v737 = -7.
if(EF31=2) v737 = -1.
fre v737.

* EF738: Beruf (Kldb 2010): Haupteinkommensbezieher.
temporary.
compute EF738k = EF114 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v738 = SUM (EF738k).
var lab v738 "Beruf (Kldb) Haupteinkommensbezieher".
if(EF31=-8) v738 = -8.
if(EF31=-7) v738 = -7.
if(EF31=2) v738 = -1.
fre v738.

* EF739: Beruf (ISCO-08): Haupteinkommensbezieher.
temporary.
compute EF739k = EF541 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v739 = SUM (EF739k).
var lab v739 "Beruf (ISCO) Haupteinkommensbezieher".
if(EF31=-8) v739 = -8.
if(EF31=-7) v739 = -7.
if(EF31=2) v739 = -1.
if(EF739 = -2) v739 = -2.
fre v739.

* EF740: Stellung im Beruf: Haupteinkommensbezieher.
temporary.
compute EF740k = EF117 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v740 = SUM (EF740k).
var lab v740 "Stellung im Beruf Haupteinkommensbezieher".
if(EF31=-8) v740 = -8.
if(EF31=-7) v740 = -7.

```

```

if(EF31=2) v740 = -1.
fre v740.

* EF741: Überwiegender Lebensunterhalt: Haupteinkommensbezieher.
temporary.
compute EF741k = EF401 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v741 = SUM (EF741k).
var lab v741 "Überwiegender Lebensunterhalt Haupteinkommensbezieher".
if(EF31=-8) v741 = -8.
if(EF31=-7) v741 = -7.
if(EF31=2) v741 = -1.
fre v741.

* EF742: Nettoeinkommen letzter Monat: Haupteinkommensbezieher.
temporary.
compute EF742k = EF436 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v742 = SUM (EF742k).
var lab v742 "Nettoeinkommen Haupteinkommensbezieher".
if(EF31=-8) v742 = -8.
if(EF31=-7) v742 = -7.
if(EF31=2) v742 = -1.
fre v742.

* EF743: Höchster allg. Schulabschluss: Haupteinkommensbezieher.
compute EF310n = EF310.
recode EF310n (6=8) (9=7) (7=9).
if(EF310n=-5 & EF309=9) EF310n=6.
if(EF310n=-5) EF310n = 8.

temporary.
compute EF743k = EF310n * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v743 = SUM (EF743k).
var lab v743 "Höchster allg. Schulabschluss Haupteinkommensbezieher".
if(EF31=-8) v743 = -8.
if(EF31=-7) v743 = -7.
if(EF31=2) v743 = -1.
fre v743.

* EF744: Höchster berufl. Abschluss: Haupteinkommensbezieher.
temporary.
compute EF744k = EF320 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v744 = SUM (EF744k).
var lab v744 "Höchster berufl. Schulabschluss Haupteinkommensbezieher".
if(EF31=-8) v744 = -8.
if(EF31=-7) v744 = -7.
if(EF31=2) v744 = -1.
fre v744.

* EF745: Höchster berufl./schul. Abschluss: Haupteinkommensbezieher.
temporary.
compute EF745k = EF540 * ((EF662>=1 & EF662<=4) & EF31=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v745 = SUM (EF745k).
var lab v745 "Höchster berufl./schul. Abschluss Haupteinkommensbezieher (ISCED)".
if(EF31=-8) v745 = -8.
if(EF31=-7) v745 = -7.
if(EF31=2) v745 = -1.
fre v745.

* EF753: Geschlecht: Haushaltsbezugsperson.
temporary.
compute EF753k = EF46 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v753 = SUM (EF753k).
var lab v753 "Geschlecht Haushaltsbezugsperson".
if(EF31=-8) v753 = -8.
if(EF31=-7) v753 = -7.
if(EF31=2) v753 = -1.
fre v753.

* EF754: Alter: Haushaltsbezugsperson.
temporary.
compute EF754k = EF44 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v754 = SUM (EF754k).
var lab v754 "Alter Haushaltsbezugsperson".
if(EF31=-8) v754 = -8.
if(EF31=-7) v754 = -7.

```

```

if(EF31=2) v754 = -1.
fre v754.

* EF756: Staatsangehörigkeit: Haushaltsbezugsperson.
temporary.
compute EF756k = EF371 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v756 = SUM (EF756k).
var lab v756 "Staatsangehörigkeit Haushaltsbezugsperson".
if(EF31=-8) v756 = -8.
if(EF31=-7) v756 = -7.
if(EF31=2) v756 = -1.
fre v756.

* EF757: Familienstand: Haushaltsbezugsperson.
temporary.
compute EF757k = EF765 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v757 = SUM (EF757k).
var lab v757 "Familienstand Haushaltsbezugsperson".
if(EF31=-8) v757 = -8.
if(EF31=-7) v757 = -7.
if(EF31=2) v757 = -1.
fre v757.

* EF758: Erwerbstyp: Haushaltsbezugsperson.
temporary.
compute EF758k = EF29 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v758 = SUM (EF758k).
var lab v758 "Erwerbstyp Haushaltsbezugsperson".
if(EF31=-8) v758 = -8.
if(EF31=-7) v758 = -7.
if(EF31=2) v758 = -1.
fre v758.

* EF759: Wohnsitz: Haushaltsbezugsperson.
temporary.
compute EF759k = EF30 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v759 = SUM (EF759k).
var lab v759 "Wohnsitz Haushaltsbezugsperson".
if(EF31=-8) v759 = -8.
if(EF31=-7) v759 = -7.
if(EF31=2) v759 = -1.
fre v759.

* EF760: 1. ausl. Staatsangehörigkeit: Haushaltsbezugsperson.
temporary.
compute EF760k = EF369 * (EF37 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 /v760 = SUM (EF760k).
var lab v760 "1. ausl. staatsangehörigkeit Haushaltsbezugsperson".
if(EF31=-8) v760 = -8.
if(EF31=-7) v760 = -7.
if(EF31=2) v760 = -1.
fre v760.

* EF770: Kinder: Anzahl in Lebensform.
temporary.
compute EF770k = 0.
if (EF44>-1 & EF44<100 & EF31=1 & EF35=3) EF770k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v770 = SUM (EF770k).
var lab v770 "Anzahl Kinder in Lebensform".
if(EF31=-8) v770 = -8.
if(EF31=-7) v770 = -7.
if(EF31=2) v770 = -1.
if(v770 > 9) v770 = 9.
fre v770.

* EF771: Kinder unter 1: Anzahl in Lebensform.
temporary.
compute EF771k = 0.
if (EF44<1 & EF31=1 & EF35=3) EF771k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v771 = SUM (EF771k).
var lab v771 "Anzahl Kinder unter 1 in Lebensform".
if(EF31=-8) v771 = -8.
if(EF31=-7) v771 = -7.
if(EF31=2) v771 = -1.
fre v771.

```

```
* EF772: Kinder unter 2: Anzahl in Lebensform.
temporary.
compute EF772k = 0.
if (EF44<2 & EF31=1 & EF35=3) EF772k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v772 = SUM (EF772k).
var lab v772 "Anzahl Kinder unter 2 in Lebensform".
if(EF31=-8) v772 = -8.
if(EF31=-7) v772 = -7.
if(EF31=2) v772 = -1.
fre v772.

* EF773: Kinder unter 3: Anzahl in Lebensform.
temporary.
compute EF773k = 0.
if (EF44<3 & EF31=1 & EF35=3) EF773k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v773 = SUM (EF773k).
var lab v773 "Anzahl Kinder unter 3 in Lebensform".
if(EF31=-8) v773 = -8.
if(EF31=-7) v773 = -7.
if(EF31=2) v773 = -1.
fre v773.

* EF774: Kinder unter 4: Anzahl in Lebensform.
temporary.
compute EF774k = 0.
if (EF44<4 & EF31=1 & EF35=3) EF774k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v774 = SUM (EF774k).
var lab v774 "Anzahl Kinder unter 4 in Lebensform".
if(EF31=-8) v774 = -8.
if(EF31=-7) v774 = -7.
if(EF31=2) v774 = -1.
fre v774.

* EF775: Kinder unter 5: Anzahl in Lebensform.
temporary.
compute EF775k = 0.
if (EF44<5 & EF31=1 & EF35=3) EF775k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v775 = SUM (EF775k).
var lab v775 "Anzahl Kinder unter 5 in Lebensform".
if(EF31=-8) v775 = -8.
if(EF31=-7) v775 = -7.
if(EF31=2) v775 = -1.
fre v775.

* EF776: Kinder unter 6: Anzahl in Lebensform.
temporary.
compute EF776k = 0.
if (EF44<6 & EF31=1 & EF35=3) EF776k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v776 = SUM (EF776k).
var lab v776 "Anzahl Kinder unter 6 in Lebensform".
if(EF31=-8) v776 = -8.
if(EF31=-7) v776 = -7.
if(EF31=2) v776 = -1.
fre v776.

* EF777: Kinder unter 7: Anzahl in Lebensform.
temporary.
compute EF777k = 0.
if (EF44<7 & EF31=1 & EF35=3) EF777k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v777 = SUM (EF777k).
var lab v777 "Anzahl Kinder unter 7 in Lebensform".
if(EF31=-8) v777 = -8.
if(EF31=-7) v777 = -7.
if(EF31=2) v777 = -1.
fre v777.

* EF778: Kinder unter 8: Anzahl in Lebensform.
temporary.
compute EF778k = 0.
if (EF44<8 & EF31=1 & EF35=3) EF778k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v778 = SUM (EF778k).
var lab v778 "Anzahl Kinder unter 8 in Lebensform".
if(EF31=-8) v778 = -8.
if(EF31=-7) v778 = -7.
```

```
if(EF31=2) v778 = -1.
fre v778.

* EF779: Kinder unter 9: Anzahl in Lebensform.
temporary.
compute EF779k = 0.
if (EF44<9 & EF31=1 & EF35=3) EF779k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v779 = SUM (EF779k).
var lab v779 "Anzahl Kinder unter 9 in Lebensform".
if(EF31=-8) v779 = -8.
if(EF31=-7) v779 = -7.
if(EF31=2) v779 = -1.
fre v779.

* EF780: Kinder unter 10: Anzahl in Lebensform.
temporary.
compute EF780k = 0.
if (EF44<10 & EF31=1 & EF35=3) EF780k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v780 = SUM (EF780k).
var lab v780 "Anzahl Kinder unter 10 in Lebensform".
if(EF31=-8) v780 = -8.
if(EF31=-7) v780 = -7.
if(EF31=2) v780 = -1.
fre v780.

* EF781: Kinder unter 11: Anzahl in Lebensform.
temporary.
compute EF781k = 0.
if (EF44<11 & EF31=1 & EF35=3) EF781k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v781 = SUM (EF781k).
var lab v781 "Anzahl Kinder unter 11 in Lebensform".
if(EF31=-8) v781 = -8.
if(EF31=-7) v781 = -7.
if(EF31=2) v781 = -1.
fre v781.

* EF782: Kinder unter 12: Anzahl in Lebensform.
temporary.
compute EF782k = 0.
if (EF44<12 & EF31=1 & EF35=3) EF782k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v782 = SUM (EF782k).
var lab v782 "Anzahl Kinder unter 12 in Lebensform".
if(EF31=-8) v782 = -8.
if(EF31=-7) v782 = -7.
if(EF31=2) v782 = -1.
fre v782.

* EF783: Kinder unter 13: Anzahl in Lebensform.
temporary.
compute EF783k = 0.
if (EF44<13 & EF31=1 & EF35=3) EF783k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v783 = SUM (EF783k).
var lab v783 "Anzahl Kinder unter 13 in Lebensform".
if(EF31=-8) v783 = -8.
if(EF31=-7) v783 = -7.
if(EF31=2) v783 = -1.
fre v783.

* EF784: Kinder unter 14: Anzahl in Lebensform.
temporary.
compute EF784k = 0.
if (EF44<14 & EF31=1 & EF35=3) EF784k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v784 = SUM (EF784k).
var lab v784 "Anzahl Kinder unter 14 in Lebensform".
if(EF31=-8) v784 = -8.
if(EF31=-7) v784 = -7.
if(EF31=2) v784 = -1.
fre v784.

* EF785: Kinder unter 15: Anzahl in Lebensform.
temporary.
compute EF785k = 0.
if (EF44<15 & EF31=1 & EF35=3) EF785k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v785 = SUM (EF785k).
var lab v785 "Anzahl Kinder unter 15 in Lebensform".
```

```
if(EF31=-8) v785 = -8.
if(EF31=-7) v785 = -7.
if(EF31=2) v785 = -1.
fre v785.

* EF786: Kinder unter 16: Anzahl in Lebensform.
temporary.
compute EF786k = 0.
if (EF44<16 & EF31=1 & EF35=3) EF786k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v786 = SUM (EF786k).
var lab v786 "Anzahl Kinder unter 16 in Lebensform".
if(EF31=-8) v786 = -8.
if(EF31=-7) v786 = -7.
if(EF31=2) v786 = -1.
fre v786.

* EF787: Kinder unter 17: Anzahl in Lebensform.
temporary.
compute EF787k = 0.
if (EF44<17 & EF31=1 & EF35=3) EF787k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v787 = SUM (EF787k).
var lab v787 "Anzahl Kinder unter 17 in Lebensform".
if(EF31=-8) v787 = -8.
if(EF31=-7) v787 = -7.
if(EF31=2) v787 = -1.
fre v787.

* EF788: Kinder unter 18: Anzahl in Lebensform.
temporary.
compute EF788k = 0.
if (EF44<18 & EF31=1 & EF35=3) EF788k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v788 = SUM (EF788k).
var lab v788 "Anzahl Kinder unter 18 in Lebensform".
if(v788 = 10) v788 = 9.
if(EF31=-8) v788 = -8.
if(EF31=-7) v788 = -7.
if(EF31=2) v788 = -1.
fre v788.

* EF789: Kinder unter 27: Anzahl in Lebensform.
temporary.
compute EF789k = 0.
if (EF44<27 & EF31=1 & EF35=3) EF789k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v789 = SUM (EF789k).
var lab v789 "Anzahl Kinder unter 27 in Lebensform".
if(v789>9) v789 = 9.
if(EF31=-8) v789 = -8.
if(EF31=-7) v789 = -7.
if(EF31=2) v789 = -1.
fre v789.

* EF790: Kinder ab 18: Anzahl in Lebensform.
temporary.
compute EF790k = 0.
if (EF44>17 & EF31=1 & EF35=3) EF790k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v790 = SUM (EF790k).
var lab v790 "Anzahl Kinder ab 18 in Lebensform".
if(EF31=-8) v790 = -8.
if(EF31=-7) v790 = -7.
if(EF31=2) v790 = -1.
fre v790.

* EF791: Kinder ab 27: Anzahl in Lebensform.
temporary.
compute EF791k = 0.
if (EF44>26 & EF31=1 & EF35=3) EF791k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v791 = SUM (EF791k).
var lab v791 "Anzahl Kinder ab 27 in Lebensform".
if(EF31=-8) v791 = -8.
if(EF31=-7) v791 = -7.
if(EF31=2) v791 = -1.
fre v791.

* EF792: Kinder 3-5 Jahre: Anzahl in Lebensform.
temporary.
compute EF792k = 0.
```



```

if (EF44>2 & EF44<6 & EF31=1 & EF35=3) EF792k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v792 = SUM (EF792k).
var lab v792 "Anzahl Kinder 3-5 in Lebensform".
if(EF31=-8) v792 = -8.
if(EF31=-7) v792 = -7.
if(EF31=2) v792 = -1.
fre v792.

* EF793: Kinder 6-9 Jahre: Anzahl in Lebensform.
temporary.
compute EF793k = 0.
if (EF44>5 & EF44<10 & EF31=1 & EF35=3) EF793k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v793 = SUM (EF793k).
var lab v793 "Anzahl Kinder 6-9 in Lebensform".
if(EF31=-8) v793 = -8.
if(EF31=-7) v793 = -7.
if(EF31=2) v793 = -1.
fre v793.

* EF794: Kinder 10-14 Jahre: Anzahl in Lebensform.
temporary.
compute EF794k = 0.
if (EF44>9 & EF44<15 & EF31=1 & EF35=3) EF794k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v794 = SUM (EF794k).
var lab v794 "Anzahl Kinder 10-14 in Lebensform".
if(EF31=-8) v794 = -8.
if(EF31=-7) v794 = -7.
if(EF31=2) v794 = -1.
fre v794.

* EF795: Kinder 15-17 Jahre: Anzahl in Lebensform.
temporary.
compute EF795k = 0.
if (EF44>14 & EF44<18 & EF31=1 & EF35=3) EF795k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v795 = SUM (EF795k).
var lab v795 "Anzahl Kinder 15-17 in Lebensform".
if(EF31=-8) v795 = -8.
if(EF31=-7) v795 = -7.
if(EF31=2) v795 = -1.
fre v795.

* EF796: Kinder 18-26 Jahre: Anzahl in Lebensform.
temporary.
compute EF796k = 0.
if (EF44>17 & EF44<27 & EF31=1 & EF35=3) EF796k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v796 = SUM (EF796k).
var lab v796 "Anzahl Kinder 18-26 in Lebensform".
if(EF31=-8) v796 = -8.
if(EF31=-7) v796 = -7.
if(EF31=2) v796 = -1.
fre v796.

* EF797: Kinder unter 6 Jahren: Anzahl in Lebensform, Schüler.
temporary.
compute EF797k = 0.
if (EF44 < 6 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF797k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v797 = SUM (EF797k).
var lab v797 "Anzahl Kinder unter 6 in Lebensform, Schüler".
if(EF31=-8) v797 = -8.
if(EF31=-7) v797 = -7.
if(EF31=2) v797 = -1.
fre v797.

* EF798: Kinder 6-9 Jahre: Anzahl in Lebensform, Schüler .
temporary.
compute EF798k = 0.
if (EF44>5 & EF44<10 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF798k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v798 = SUM (EF798k).
var lab v798 "Anzahl Kinder 6-9 in Lebensform, Schüler".
if(EF31=-8) v798 = -8.
if(EF31=-7) v798 = -7.
if(EF31=2) v798 = -1.
fre v798.

* EF799: Anzahl in Lebensform: Kinder 10-14 Jahre, Schüler .

```

```

temporary.
compute EF799k = 0.
if (EF44>9 & EF44<15 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF799k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v799 = SUM (EF799k).
var lab v799 "Anzahl Kinder 10-14 in Lebensform, Schüler".
if(EF31=-8) v799 = -8.
if(EF31=-7) v799 = -7.
if(EF31=2) v799 = -1.
fre v799.

* EF800: Anzahl in Lebensform: Kinder 15-17 Jahre, Schüler .
temporary.
compute EF800k = 0.
if (EF44>14 & EF44<18 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF800k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v800 = SUM (EF800k).
var lab v800 "Anzahl Kinder 15-17 in Lebensform, Schüler".
if(EF31=-8) v800 = -8.
if(EF31=-7) v800 = -7.
if(EF31=2) v800 = -1.
fre v800.

* EF801: Anzahl in Lebensform: Kinder 18-26 Jahre, Schüler .
temporary.
compute EF801k = 0.
if (EF44>17 & EF44<27 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF801k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v801 = SUM (EF801k).
var lab v801 "Anzahl Kinder 18-26 in Lebensform, Schüler".
if(EF31=-8) v801 = -8.
if(EF31=-7) v801 = -7.
if(EF31=2) v801 = -1.
fre v801.

* EF802: Anzahl in Lebensform: Kinder ab 27 Jahren, Schüler .
temporary.
compute EF802k = 0.
if (EF44>26 & EF31=1 & EF35=3 & (EF287=1 OR EF287=2)) EF802k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v802 = SUM (EF802k).
var lab v802 "Anzahl Kinder ab 27 in Lebensform, Schüler".
if(EF31=-8) v802 = -8.
if(EF31=-7) v802 = -7.
if(EF31=2) v802 = -1.
fre v802.

* EF803: Alter jüngstes Kind in Lebensform.
temporary.
compute EF803k = (EF44*(EF35=3 & EF805=1)).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v803 = MAX (EF803k).
var lab v803 "Alter: Jüngstes Kind in Lebensform".
if(EF31=-8) v803 = -8.
if(EF31=-7) v803 = -7.
if(EF31=2) v803 = -1.
if(v803 = 0 & EF770 = 0) v803 = -5.
fre v803.

* EF804: Alter: Ältestes Kind in Lebensform.
if (EF35>0 & EF35<4) v804_1k = 1.
if (v804_1k=1) v804_1k = (EF44*(EF35=3)).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v804_1 = MAX (v804_1k).
fre v804 1.
* ältestes Kind ist unter 1 Jahr alt.
if (EF35>0 & EF35<4) v804_0k = 1.
if (v804_0k=1) v804_0k = (EF44=0 & EF35=3).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v804_0 = SUM (v804_0k).
fre v804 0.

if (v804_0>0 & v804_0<4) v804 = 0.
if (v804_1>0 & v804_1<96) v804=v804_1.
if (v804_0 = 0 & v804_1=0) v804=-5.
var lab v804 "Alter: Ältestes Kind in Lebensform".
fre v804.

* EF812: Personen am Nebenwohnsitz: Anzahl in Lebensform.
temporary.
compute EF812k = 0.

```

```

if (EF30 = 3) EF812k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v812 = SUM (EF812k).
var lab v812 "Anzahl Personen Nebenwohnsitz".
if(EF31=-8) v812 = -8.
if(EF31=-7) v812 = -7.
if(EF31=2) v812 = -1.
fre v812.

* EF819: Geschlecht: Bezugsperson der Lebensform .
temporary.
compute EF819k = EF46 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v819 = SUM (EF819k).
var lab v819 "Geschlecht Bezugsperson Lebensform".
if(EF31=-8) v819 = -8.
if(EF31=-7) v819 = -7.
if(EF31=2) v819 = -1.
fre v819.

* EF820: Alter: Bezugsperson der Lebensform .
temporary.
compute EF820k = EF44 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v820 = SUM (EF820k).
var lab v820 "Alter Bezugsperson Lebensform".
if(EF31=-8) v820 = -8.
if(EF31=-7) v820 = -7.
if(EF31=2) v820 = -1.
fre v820.

* EF822: Staatsangehörigkeit: Bezugsperson der Lebensform .
temporary.
compute EF822k = EF371 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v822 = SUM (EF822k).
var lab v822 "Staatsangehörigkeit Bezugsperson Lebensform".
if(EF31=-8) v822 = -8.
if(EF31=-7) v822 = -7.
if(EF31=2) v822 = -1.
fre v822.

* mig_lf1: Migrationsstatus: Bezugsperson der Lebensform.
temporary.
compute vmig_lf1k = EF2001 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /vmig lf1 = SUM (vmig lf1k).
var lab vmig_lf1 "Migrationsstatus Bezugsperson Lebensform".
if(EF31=-8) vmig_lf1 = -8.
if(EF31=-7) vmig_lf1 = -7.
if(EF31=2) vmig_lf1 = -1.
fre vmig_lf1.

* migs lf: Staatsangehörigkeit: Bezugsperson der Lebensform.
temporary.
compute vmigs_lf = EF2007 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /vmigs_lf = SUM (vmigs_lf).
var lab vmigs_lf "Staatsangehörigkeit Bezugsperson Lebensform".
if(EF31=-8) vmigs_lf = -8.
if(EF31=-7) vmigs_lf = -7.
if(EF31=2) vmigs_lf = -1.
fre vmigs_lf.

* EF823: Familienstand: Bezugsperson der Lebensform .
temporary.
compute EF823k = EF765 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v823 = SUM (EF823k).
var lab v823 "Familienstand Bezugsperson Lebensform".
if(EF31=-8) v823 = -8.
if(EF31=-7) v823 = -7.
if(EF31=2) v823 = -1.
fre v823.

* EF824: Erwerbstyp: Bezugsperson der Lebensform .
temporary.
compute EF824k = EF29 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v824 = SUM (EF824k).
var lab v824 "Erwerbstyp Bezugsperson Lebensform".
if(EF31=-8) v824 = -8.

```

```

if(EF31=-7) v824 = -7.
if(EF31=2) v824 = -1.
fre v824.

* EF825: Wirtschaftszweig: Bezugsperson der Lebensform .
temporary.
compute EF825k = EF137 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v825 = SUM (EF825k).
var lab v825 "Wirtschaftszweig Bezugsperson Lebensform".
if(EF31=-8) v825 = -8.
if(EF31=-7) v825 = -7.
if(EF31=2) v825 = -1.
fre v825.

* EF826: Beruf (Kldb 2010): Bezugsperson der Lebensform.
temporary.
compute EF826k = EF114 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v826 = SUM (EF826k).
var lab v826 "Beruf (Kldb 2010) Bezugsperson Lebensform" .
if(EF31=-8) v826 = -8.
if(EF31=-7) v826 = -7.
if(EF31=2) v826 = -1.
if(EF326=-2) v826 = -2.
fre v826.

* EF827: Beruf (ISCO-08): Bezugsperson der Lebensform.
temporary.
compute EF827k = EF541 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v827 = SUM (EF827k).
var lab v827 "Beruf (ISCO-08) Bezugsperson Lebensform".
if(EF31=-8) v827 = -8.
if(EF31=-7) v827 = -7.
if(EF31=2) v827 = -1.
if(EF326=-2) v827 = -2.
fre v827.

* EF828: Stellung im Beruf: Bezugsperson der Lebensform .
temporary.
compute EF828k = EF117 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v828 = SUM (EF828k).
var lab v828 "Beruf (ISCO-08) Bezugsperson Lebensform".
if(EF31=-8) v828 = -8.
if(EF31=-7) v828 = -7.
if(EF31=2) v828 = -1.
fre v828.

* EF829: Überwiegender Lebensunterhalt: Bezugsperson der Lebensform .
temporary.
compute EF829k = EF401 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v829 = SUM (EF829k).
var lab v829 "Überwiegender Lebensunterhalt Bezugsperson Lebensform".
if(EF31=-8) v829 = -8.
if(EF31=-7) v829 = -7.
if(EF31=2) v829 = -1.
fre v829.

* EF830: Nettoeinkommen: Bezugsperson der Lebensform .
temporary.
compute EF830k = EF436 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v830 = SUM (EF830k).
var lab v830 "Nettoeinkommen Bezugsperson Lebensform".
if(EF31=-8) v830 = -8.
if(EF31=-7) v830 = -7.
if(EF31=2) v830 = -1.
fre v830.

* EF831: Höchster schul. Abschluss: Bezugsperson der Lebensform .
compute EF310n1 = EF310.
recode EF310n1 (6=8) (9=7) (7=9).
if(EF310n1=-5 & EF309=9) EF310n1= 6.
if(EF310n1=-5) EF310n1= 8.
if(EF310n1=-3) EF310n1= 8.

temporary.
compute EF831k=EF310n1 * (EF35 = 1).
aggregate outfile = * mode = addvariables

```

```

/break EF1 EF3 EF4 EF5b EF12 EF27 /v831 = SUM (EF831k).
var lab v831 "Höchster schul. Abschluss Bezugsperson Lebensform".
if(EF31=-8) v831 = -8.
if(EF31=-7) v831 = -7.
if(EF31=2) v831= -1.
fre v831.

* EF832: Höchster berufl. Abschluss: Bezugsperson der Lebensform.
temporary.
compute EF832k = EF320 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v832 = SUM (EF832k).
var lab v832 "Höchster berufl. Schulabschluss Bezugsperson Lebensform".
if(EF31=-8) v832 = -8.
if(EF31=-7) v832 = -7.
if(EF31=2) v832= -1.
if(v832=-3) v832=88.
fre v832.

* EF833: Höchster berufl./ schul. Abschluss: Bezugsperson der Lebensform (ISCED 2011).
temporary.
compute EF833k = EF540 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v833 = SUM (EF833k).
var lab v833 "Höchster berufl./schul. Abschluss Bezugsperson der Lebensform (ISCED)".
if(EF31=-8) v833 = -8.
if(EF31=-7) v833 = -7.
if(EF31=2) v833= -1.
fre v833.

* EF834: Arbeitszeittyp: Bezugsperson der Lebensform .
compute EF130n = EF130.
if(EF130n=-5) EF130n = 8.
if(EF130n=1 & EF129 =1) EF130n = 8.
if(EF130n=2 & EF129 =1) EF130n = 8.
if(EF130n=3 & EF129 =1) EF130n = 8.
if(EF130n=4 & EF129 =1) EF130n = 8.
if(EF130n=5 & EF129 =1) EF130n = 8.
if(EF130n=6 & EF129 =1) EF130n = 8.
if(EF130n=7 & EF129 =1) EF130n = 8.
if(EF130n=8 & EF129 =1) EF130n = 8.
if(EF130n=9 & EF129 =1) EF130n = 8.
fre EF130n.

temporary.
compute EF834k = EF130n * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v834 = SUM (EF834k).
var lab v834 "Arbeitszeittyp Bezugsperson der Lebensform".
if(EF31=-8) v834 = -8.
if(EF31=-7) v834 = -7.
if(EF31=2) v834= -1.
fre v834.

* EF835: Normale Arbeitszeit (je Woche: Stunden): Bezugsperson der Lebensform .
temporary.
compute EF835k = EF131 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v835 = SUM (EF835k).
var lab v835 "Normale Arbeitszeit Bezugsperson Lebensform".
if(EF31=-8) v835 = -8.
if(EF31=-7) v835 = -7.
if(EF31=2) v835= -1.
if(v835=-1) v835=-2.
fre v835.

* EF836: Grund geringer Arbeitszeit: Bezugsperson der Lebensform .
compute EF147n = EF147.
if(EF147n = 3 & EF29=1 & EF146=2) EF147n=2.
if(EF147n = 4 & EF29=1 & EF146=2) EF147n=3.
if(EF147n = 17 & EF29=1 & EF146=2) EF147n=5.
if(EF147n = 8 & EF29=1 & EF146=2) EF147n=6.
if(EF147n = 16 & EF29=1 & EF146=2) EF147n=7.
if((EF147n= 1 OR EF147n=2 OR EF147n=6 OR EF147n=9 OR EF147n=10 OR EF147n=11 OR EF147n=12 OR
EF147n=13 OR EF147n=14 OR EF147n=15 OR EF147n=18 OR EF147n=19 OR EF147n=99)
& EF29=1 & EF146=2 & EF147~=3 & EF147~=4 & EF147~=8 & EF147~=16 & EF147~=17) EF147n = 8.
if(EF147n = -5 & EF29=1 & EF146~=2) EF147n=1.
fre EF147n.

temporary.
compute EF836k = EF147n * (EF35 = 1).
aggregate outfile = * mode = addvariables

```

```

/break EF1 EF3 EF4 EF5b EF12 EF27 /v836 = SUM (EF836k).
var lab v836 "Grund geringer Arbeitszeit Bezugsperson Lebensform".
if(EF31=-8) v836 = -8.
if(EF31=-7) v836 = -7.
if(EF31=2) v836=-1.
crosstabs EF836 by v836.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF837: Erwerbsunterbrechung (Berichtsw.): Bezugsperson der Lebensform.
compute EF85nl = EF85.
if(EF85nl>0 & EF85nl<12 & EF29=1) EF85nl = 1.
if(EF85nl=-5 & EF29=1) EF85nl=2.
fre EF85nl.

temporary.
compute EF837k = EF85nl * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v837 = SUM (EF837k).
var lab v837 "1. ausländische Staatsangehörigkeit Bezugsperson Lebensform".
if(EF31=-8) v837 = -8.
if(EF31=-7) v837 = -7.
if(EF31=2) v837=-1.
if(v837>2 & v837<11) v837=-2.
crosstabs EF837 by v837.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF838: 1. ausländische Staatsangehörigkeit: Bezugsperson der Lebensform.
temporary.
compute EF838k = EF369 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v838 = SUM (EF838k).
var lab v838 "1. ausländische Staatsangehörigkeit Bezugsperson Lebensform" .
if(EF31=-8) v838 = -8.
if(EF31=-7) v838 = -7.
if(EF31=2) v838=-1.
fre v838.

* EF843: Geschlecht: Bezugsperson der Lebensform.
temporary.
compute EF843k = EF843 * (EF35 = 1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v843 = SUM (EF843k).
var lab v843 "Geschlecht Bezugsperson Lebensform".
if(EF31=-8) v843 = -8.
if(EF31=-7) v843 = -7.
if(EF31=2) v843=-1.
fre v843.

* EF844: Alter: Lebenspartner der Bezugsperson der Lebensform.
temporary.
compute EF844k = EF44 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v844 = SUM (EF844k).
var lab v844 "Alter Lebenspartner Bezugsperson Lebensform".
if(EF31=-8) v844 = -8.
if(EF31=-7) v844 = -7.
if(EF31=2) v844=-1.
if(v844=0) v844=-5.
crosstabs EF844 by v844.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF846: Staatsangehörigkeit: Lebenspartner in Lebensform.
temporary.
compute EF846k = EF371 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v846 = SUM (EF846k).
var lab v846 "Staatsangehörigkeit Lebenspartner in Lebensform".
if(EF31=-8) v846 = -8.
if(EF31=-7) v846 = -7.
if(EF31=2) v846=-1.
if(v846=0) v846=-5.
crosstabs EF846 by v846.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* mig_lf2: Migrationsstatus: Lebenspartner in Lebensform .
temporary.
compute mig_lf2k = EF2001 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /vmig lf2 = SUM (mig lf2k).
var lab vmig lf2 "Migrationsstatus Lebenspartner in Lebensform".
if(vmig_lf2=0 & EF762<0) vmig_lf2 = EF762.
fre vmig_lf2.

```

```

temporary.
compute LP_DEk = (EF2001=0 & (EF35 = 2)).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /LP_DE = SUM (LP_DEk).
if(vmig_lf2 = 0 & LP_DE=0) vmig_lf2=-5.
fre vmig_lf2.

* migs lf2: Staatsangehörigkeit: Lebenspartner Lebensform.
temporary.
compute vmigs_lf2k = (EF2007*(EF35 = 2)).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /vmigs_lf2 = SUM (vmigs_lf2k).
var lab vmigs lf2 "Staatsangehörigkeit Lebenspartner in Lebensform".
if(vmigs lf2=0 & EF762<0) vmigs lf2=EF762.
fre vmigs_lf2.

temporary.
compute LP_DE2k = (EF2007=0 & (EF35 = 2)).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /LP DE2 = SUM (LP DE2k).
if(vmigs lf2 = 0 & LP DE2=0) vmigs lf2=-5.
fre vmigs_lf2.

* EF847: Familienstand: Lebenspartner in Lebensform.
temporary.
compute EF847k = EF765 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v847 = SUM (EF847k).
var lab v847 "Familienstand Lebenspartner in Lebensform".
if(EF31=-8) v847 = -8.
if(EF31=-7) v847 = -7.
if(EF31=2) v847= -1.
if(v847=0) v847= -5.
crosstabs EF847 by v847.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF848: Erwerbstyp: Lebenspartner in Lebensform.
temporary.
compute EF848k = EF29 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v848 = SUM (EF848k).
var lab v848 "Erwerbstyp Lebenspartner in Lebensform".
if(EF31=-8) v848 = -8.
if(EF31=-7) v848 = -7.
if(EF31=2) v848= -1.
if(v848=0) v848= -5.
crosstabs EF848 by v848.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF849: Wirtschaftszweig: Lebenspartner in Lebensform.
temporary.
compute EF849k = EF137 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v849 = SUM (EF849k).
var lab v849 "Wirtschaftszweig Lebenspartner in Lebensform".
if(EF31=-8) v849 = -8.
if(EF31=-7) v849 = -7.
if(EF31=2) v849= -1.
if(v849=0) v849= -5.
crosstabs EF849 by v849.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF850: Beruf (Kldb 2010): Lebenspartner in Lebensform.
temporary.
compute EF850k = EF114 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v850 = SUM (EF850k).
var lab v850 "Beruf (Kldb) Lebenspartner in Lebensform".
if(EF31=-8) v850 = -8.
if(EF31=-7) v850 = -7.
if(EF31=2) v850= -1.
if(EF850=-2) v850=-2.
if(v850=0) v850= -5.
fre v850.

* EF851: Beruf (ISCO-08): Lebenspartner in Lebensform .
temporary.
compute EF851k = EF541 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v851 = SUM (EF851k).
var lab v851 "Beruf (ISCO) Lebenspartner in Lebensform".

```

```

if(EF31=-8) v851 = -8.
if(EF31=-7) v851 = -7.
if(EF31=2) v851 = -1.
if(EF850=-2) v851 = -2.
if(v851=0) v851= -5.
if(v851=7314) v851 = 7319.
if(v851=8151) v851 = 8153.
if(v851=-5 & EF33=-5) v851=3313.
crosstabs EF851 by v851.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF852: Stellung im Beruf: Lebenspartner in Lebensform .
temporary.
compute EF852k = EF117 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v852 = SUM (EF852k).
var lab v852 "Stellung im Beruf Lebenspartner in Lebensform".
if(EF31=-8) v852 = -8.
if(EF31=-7) v852 = -7.
if(EF31=2) v852= -1.
if(v852=0) v852= -5.
crosstabs EF852 by v852.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF853: Überwiegender Lebensunterhalt: Lebenspartner in Lebensform .
temporary.
compute EF853k = EF401 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v853 = SUM (EF853k).
var lab v853 "Überwiegender Lebensunterhalt Lebenspartner in Lebensform".
if(EF31=-8) v853 = -8.
if(EF31=-7) v853 = -7.
if(EF31=2) v853= -1.
if(v853=0) v853= -5.
crosstabs EF853 by v853.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF854: Nettoeinkommen letzter Monat: Lebenspartner in Lebensform .
temporary.
compute EF854k = EF436 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v854 = SUM (EF854k).
var lab v854 "Nettoeinkommen Lebenspartner in Lebensform".
if(EF31=-8) v854 = -8.
if(EF31=-7) v854 = -7.
if(EF31=2) v854= -1.
if(v854=0) v854= -5.
crosstabs EF854 by v854.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF855: Höchster allg. Schulabschluss: Lebenspartner in Lebensform.
compute EF3101p = EF310.
recode EF3101p (6=8) (9=7) (7=9).
if(EF3101p=-5 & EF309=9) EF3101p=6.
if(EF3101p=-5) EF3101p=8.

temporary.
compute EF855k = EF3101p * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v855 = SUM (EF855k).
var lab v855 "Höchster allg. Schulabschluss Lebenspartner in Lebensform".
if(EF31=-8) v855 = -8.
if(EF31=-7) v855 = -7.
if(EF31=2) v855= -1.
if(v855=0) v855= -5.
crosstabs EF855 by v855.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF856: Höchster berufl. Abschluss: Lebenspartner in Lebensform .
temporary.
compute EF856k = EF320 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v856 = SUM (EF856k).
var lab v856 "Höchster berufl. Schulabschluss Lebenspartner in Lebensform".
if(EF31=-8) v856 = -8.
if(EF31=-7) v856 = -7.
if(EF31=2) v856= -1.
if(v856=0) v856= -5.
crosstabs EF856 by v856.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF857: Höchster berufl./ schul. Abschluss: Lebenspartner in Lebensform (ISCED 2011) .

```



```

temporary.
compute EF857k = EF540 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v857 = SUM (EF857k).
var lab v857 "Höchster berufl./schul. Abschluss Lebenspartner in Lebensform (ISCED)".
if(EF31=-8) v857 = -8.
if(EF31=-7) v857 = -7.
if(EF31=2) v857= -1.
if(v857=0) v857= -5.
crosstabs EF857 by v857.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF858: Arbeitszeittyp: Lebenspartner in Lebensform.
compute EF130nlp = EF130.
if (EF130nlp=-5) EF130nlp = 8.
if (EF130nlp>0 & EF130nlp<10 & EF129=1) EF130nlp = 8.

temporary.
compute EF858k = EF130nlp * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v858 = SUM (EF858k).
var lab v858 "Arbeitszeittyp Lebenspartner in Lebensform".
if(EF31=-8) v858 = -8.
if(EF31=-7) v858 = -7.
if(EF31=2) v858= -1.
temporary.
compute LPinLFk = (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /LPinLF = SUM (LPinLFk).
if(LPinLF=0 & v858=0) v858= -5.
crosstabs EF858 by v858.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF859: Normale Arbeitszeit (je Woche: Stunden): Lebenspartner in Lebensform.
temporary.
compute EF859k = EF131 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v859 = SUM (EF859k).
var lab v859 "Normale Arbeitszeit Lebenspartner in Lebensform".
if(EF31=-8) v859 = -8.
if(EF31=-7) v859 = -7.
if(EF31=2) v859= -1.
if(v859=0) v859= -5.
crosstabs EF859 by v859.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF860: Grund geringer Arbeitszeit: Lebenspartner in der Lebensform.
compute EF147np = EF147.
if(EF147np=3 & EF29=1 & EF146=2) EF147np=2.
if(EF147np=4 & EF29=1 & EF146=2) EF147np=3.
if(EF147np=17 & EF29=1 & EF146=2) EF147np=5.
if(EF147np=8 & EF29=1 & EF146=2) EF147np=6.
if(EF147np=16 & EF29=1 & EF146=2) EF147np=7.
if (EF147np= 1 OR EF147np=2 OR EF147np=6 OR EF147np=9 OR EF147np=10 OR EF147np=11 OR EF147np=12
OR EF147np=13 OR EF147np=14 OR EF147np=15 OR EF147np=18 OR EF147np=19 OR EF147np=99)
& EF29=1 & EF146=2 & EF147~3 & EF147~4 & EF147~8 & EF147~16 & EF147~17) EF147np = 8.
if(EF147np=-5 & EF29=1 & EF146~2) EF147np=1.
fre EF147np.

temporary.
compute EF860k = EF147np * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v860 = SUM (EF860k).
var lab v860 "Grund geringer Arbeitszeit Lebenspartner in Lebensform".
if(EF31=-8) v860 = -8.
if(EF31=-7) v860 = -7.
if(EF31=2) v860= -1.
if(v860=0) v860= -5.
crosstabs EF860 by v860.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF861: Erwerbsunterbrechung (Berichtsw.): Lebenspartner in Lebensform.
if (EF35>0 & EF35<4) v861_1k = 1.
if (v861_1k) v861_1k = (EF31=1 & EF35=2 & EF29 =1 & EF85>0 & EF85<12).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v861_1 = SUM (v861_1k).
fre v861_1

if (EF35>0 & EF35<4) v861_2k = 1.
if (v861_2k) v861_2k = (EF31=1 & EF35=2 & EF29 =1 & EF85=-5).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v861_2 = SUM (v861_2k).

```

```

fre v861 2

if (EF35>0 & EF35<4) v861_m2k = 1.
if (v861_m2k) v861_m2k = (EF31=1 & EF35=2 & EF29 ~|=1).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v861_m2 = SUM (v861_m2k).
fre v861_m2

* kein Lebenspartner in Lebensform -> v861=-5.
compute v861_m5k = 0.
if (EF35=2 & EF35>0 & EF35<4) v861_m5k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v861_m5 = SUM (v861_m5k).

if (v861_1 = 1 & EF31=1) v861 = 1.
if (v861_2 = 1 & EF31=1) v861 = 2.
if (v861_m2 = 1 & EF31=1) v861 = -2.
if (v861_m5 = 0 & EF31=1) v861 = -5.
var lab v861 "Erwerbsunterbrechung (Berichtsw.): Lebenspartner in der Lebensform".

* Ungültige Werte: EF35<1.
RECODE v861 (SYSMIS = 99).
if (v861= 99 & EF31<1) v861 = EF31.
if (v861= 99 & EF31=2) v861 = -1.
crosstabs EF861 by v861.
* ACHTUNG: Abweichungen zur Originaltypisierung.

* EF862: 1. ausländische Staatsangehörigkeit: Lebenspartner in Lebensform.
temporary.
compute EF862k = EF369 * (EF35 = 2).
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF27 /v862 = SUM (EF862k).
var lab v862 "1. ausländische Staatsangehörigkeit Lebenspartner in Lebensform".
if(EF31=-8) v862 = -8.
if(EF31=-7) v862 = -7.
if(EF31=2) v862=-1.
if(v862=0) v862=-5.
fre v862.

* EF866: Kinder: Anzahl in Familie (trad. Familienkonzept).
temporary.
compute EF866k = 0.
if (EF31=1 & EF34=3) EF866k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF25 /v866 = SUM (EF866k).
var lab v866 "Anzahl Kinder in Familie (trad. Familienkonzept)".
if(EF31=-8) v866 = -8.
if(EF31=-7) v866 = -7.
if(EF31=2) v866=-1.
if(v866>8) v866= 9.
fre v866.

* EF867: Kinder unter 18 Jahren: Anzahl in Familie (trad. Familienkonzept).
temporary.
compute EF867k = 0.
if (EF31=1 & EF34=3 & EF44<18) EF867k = 1.
aggregate outfile = * mode = addvariables
/break EF1 EF3 EF4 EF5b EF12 EF25 /v867 = SUM (EF867k).
var lab v867 "Anzahl Kinder unter 18 Jahren in Familie (trad. Familienkonzept)".
if(EF31=-8) v867 = -8.
if(EF31=-7) v867 = -7.
if(EF31=2) v867=-1.
if(v867>8) v867=9.
fre v867.

* Definition der User-Missing Values .
missing values
EF4 ( -7 ) /
EF5a ( -7, -8 ) /
EF5b ( -7, -8 ) /
EF6 ( -7, -8 ) /
EF7 ( -7, -8 ) /
EF17 ( -3, -7, -8 ) /
EF20 ( -7, -8 ) /
EF25 ( -1, -7, -8 ) /
EF27 ( -1, -7, -8 ) /
EF29 ( -7, -8 ) /
EF30 ( -7, -8 ) /
EF31 ( -7, -8 ) /
EF32 ( -7, -8 ) /
EF33 ( -8 THRU -1 ) /

```

EF33 (-1, -7, -8) /
EF34 (-1, -7, -8) /
EF35 (-1, -7, -8) /
EF36 (-1, -7, -8) /
EF37 (-1, -7, -8) /
EF38 (-8 THRU -3) /
EF39 (-3, -7, -8) /
EF44 (-7, -8) /
EF45 (-5, -7, -8) /
EF46 (-7, -8) /
EF47 (-7, -8) /
EF49 (-7, -8) /
EF50 (-8 THRU -1) /
EF51 (-8 THRU -1) /
EF52 (-8 THRU -1) /
EF53 (-8 THRU -1) /
EF54 (-8 THRU -1) /
EF55 (-8 THRU -1) /
EF56 (-8 THRU -1) /
EF57 (-8 THRU -1) /
EF58 (-8 THRU -1) /
EF63 (-7, -8) /
EF67 (-8 THRU -3) /
EF75 (-7, -8) /
EF76 (-5, -7, -8) /
EF77 (-8 THRU -3) /
EF79 (-8 THRU -3) /
EF83 (-8 THRU -3) /
EF84 (-3, -7, -8) /
EF85 (-8 THRU -2) /
EF86 (-8 THRU -2) /
EF87 (-8 THRU -2) /
EF88 (-8 THRU -3) /
EF90 (-8 THRU -3) /
EF91 (-8 THRU -3) /
EF92 (-8 THRU -3) /
EF93 (-8 THRU -3) /
EF94 (-8 THRU -3) /
EF94_f1 (-7, -8) /
EF94_f2 (-7, -8) /
EF94_f3 (-7, -8) /
EF94_f4 (-7, -8) /
EF96 (-8 THRU -3) /
EF105 (-8 THRU -3) /
EF106 (-8 THRU -3) /
EF114 (-8 THRU -3) /
EF114_f1 (-7, -8) /
EF114_f2 (-7, -8) /
EF114_f3 (-7, -8) /
EF114_f4 (-7, -8) /
EF116u1 (-8 THRU -2) /
EF116u2 (-8 THRU -2) /
EF116u3 (-8 THRU -2) /
EF116u4 (-8 THRU -2) /
EF116u5 (-8 THRU -2) /
EF116u6 (-8 THRU -2) /
EF117 (-8 THRU -2) /
EF120 (-8 THRU -2) /
EF121 (-8 THRU -2) /
EF122 (-8 THRU -2) /
EF123 (-8 THRU -2) /
EF124 (-8 THRU -2) /
EF125 (-8 THRU -2) /
EF126 (-8 THRU -2) /
EF127 (-8 THRU -2) /
EF128 (-8 THRU -2) /
EF129 (-8 THRU -2) /
EF130 (-8 THRU -2) /
EF131 (-8 THRU -2) /
EF132 (-8 THRU -2) /
EF133 (-8 THRU -2) /
EF134 (-8 THRU -2) /
EF137 (-8 THRU -2) /
EF138 (-8 THRU -2) /
EF139 (-8 THRU -2) /
EF140 (-8 THRU -2) /
EF146 (-8 THRU -2) /
EF147 (-8 THRU -2) /
EF148 (-8 THRU -2) /
EF149 (-8 THRU -2) /
EF150 (-8 THRU -2) /
EF151 (-8 THRU -2) /
EF152 (-8 THRU -2) /

```
EF153 ( -8 THRU -2 ) /
EF154 ( -8 THRU -2 ) /
EF160 ( -8 THRU -2 ) /
EF161 ( -8 THRU -2 ) /
EF162 ( -8 THRU -2 ) /
EF163 ( -8 THRU -2 ) /
EF164 ( -8 THRU -2 ) /
EF165 ( -8 THRU -2 ) /
EF166 ( -8 THRU -2 ) /
EF167 ( -8 THRU -2 ) /
EF171 ( -8 THRU -2 ) /
EF177 ( -8 THRU -2 ) /
EF178 ( -8 THRU -2 ) /
EF179u1 ( -8 THRU -2 ) /
EF179u2 ( -8 THRU -2 ) /
EF179u3 ( -8 THRU -2 ) /
EF179u4 ( -8 THRU -2 ) /
EF179u5 ( -8 THRU -2 ) /
EF187 ( -8 THRU -2 ) /
EF188 ( -8 THRU -2 ) /
EF190 ( -8 THRU -2 ) /
EF197 ( -8 THRU -2 ) /
EF200 ( -8 THRU -2 ) /
EF201 ( -8 THRU -2 ) /
EF202 ( -8 THRU -2 ) /
EF203 ( -8 THRU -2 ) /
EF206 ( -8 THRU -2 ) /
EF214 ( -8 THRU -2 ) /
EF215 ( -8 THRU -2 ) /
EF217 ( -8 THRU -2 ) /
EF223 ( -8 THRU -3 ) /
EF224 ( -8 THRU -3 ) /
EF225 ( -8 THRU -3 ) /
EF231 ( -8 THRU -3 ) /
EF232 ( -8 THRU -3 ) /
EF233 ( -8 THRU -3 ) /
EF234 ( -8 THRU -3 ) /
EF235 ( -8 THRU -3 ) /
EF243 ( -8 THRU -3 ) /
EF245 ( -8 THRU -3 ) /
EF246 ( -8 THRU -3 ) /
EF247 ( -8 THRU -3 ) /
EF254 ( -7, -8 ) /
EF255 ( -7, -8 ) /
EF256 ( -7, -8 ) /
EF257 ( -7, -8 ) /
EF258 ( -7, -8 ) /
EF259 ( -7, -8 ) /
EF260 ( -7, -8 ) /
EF261 ( -7, -8 ) /
EF262 ( -7, -8 ) /
EF263 ( -8 THRU -3 ) /
EF265 ( -8 THRU -3 ) /
EF266 ( -8 THRU -3 ) /
EF268 ( -7, -8 ) /
EF270 ( -7, -8 ) /
EF271 ( -7, -8 ) /
EF272 ( -7, -8 ) /
EF273 ( -7, -8 ) /
EF274 ( -7, -8 ) /
EF276 ( -8 THRU -3 ) /
EF277 ( -8 THRU -3 ) /
EF278 ( -8 THRU -3 ) /
EF279 ( -8 THRU -3 ) /
EF285 ( -8 THRU -3 ) /
EF286 ( -8 THRU -3 ) /
EF287 ( -5, -7, -8 ) /
EF288 ( -7, -8 ) /
EF289 ( -5, -7, -8 ) /
EF290 ( -5, -7, -8 ) /
EF291 ( -5, -7, -8 ) /
EF305 ( -8 THRU -3 ) /
EF306 ( -8 THRU -3 ) /
EF309 ( -8 THRU -3 ) /
EF310 ( -8 THRU -3 ) /
EF311 ( -3, -7, -8 ) /
EF312 ( -8 THRU -3 ) /
EF313 ( -8 THRU -3 ) /
EF314 ( -8 THRU -3 ) /
EF315 ( -8 THRU -3 ) /
EF316 ( -8 THRU -3 ) /
EF317 ( -8 THRU -3 ) /
```

EF319 (-8 THRU -3) /
EF320 (-3, -7, -8) /
EF323 (-3, -7, -8) /
EF325 (-8 THRU -3) /
EF326 (-8 THRU -3) /
EF327 (-8 THRU -3) /
EF328 (-8 THRU -3) /
EF329 (-8 THRU -3) /
EF330 (-8 THRU -3) /
EF366 (-7, -8) /
EF367 (-5, -7, -8) /
EF368 (-7, -8) /
EF369 (-5, -7, -8) /
EF370 (-7, -8) /
EF371 (-7, -8) /
EF372 (-5, -7, -8) /
EF373 (-5, -7, -8) /
EF374 (-5, -7, -8) /
EF375 (-5, -7, -8) /
EF376 (-5, -7, -8) /
EF377 (-5, -7, -8) /
EF378 (-5, -7, -8) /
EF379 (-5, -7, -8) /
EF380 (-5, -7, -8) /
EF381 (-5, -7, -8) /
EF383 (-7, -8) /
EF384 (-5, -7, -8) /
EF385 (-3, -7, -8) /
EF391 (-8 THRU -3) /
EF401 (-7, -8) /
EF402 (-7, -8) /
EF403 (-7, -8) /
EF406 (-7, -8) /
EF407 (-7, -8) /
EF408 (-7, -8) /
EF409 (-7, -8) /
EF410 (-7, -8) /
EF413 (-7, -8) /
EF413a (-7, -8) /
EF414 (-7, -8) /
EF414a (-7, -8) /
EF412 (-7, -8) /
EF415 (-7, -8) /
EF416 (-7, -8) /
EF417 (-7, -8) /
EF418 (-7, -8) /
EF419 (-7, -8) /
EF421 (-7, -8) /
EF422 (-7, -8) /
EF423 (-7, -8) /
EF424 (-7, -8) /
EF425 (-7, -8) /
EF426 (-7, -8) /
EF427 (-7, -8) /
EF428 (-5, -7, -8) /
EF429 (-7, -8) /
EF430 (-7, -8) /
EF431 (-7, -8) /
EF432 (-7, -8) /
EF433 (-7, -8) /
EF434 (-7, -8) /
EF435 (-7, -8) /
EF436 (-7, -8) /
EF438 (-5, -7, -8) /
EF439 (-7, -8) /
EF440 (-7, -8) /
EF441 (-7, -8) /
EF442 (-8 THRU -3) /
EF443 (-5, -7, -8) /
EF445 (-8 THRU -3) /
EF446 (-7, -8) /
EF451 (-5, -7, -8) /
EF452 (-5, -7, -8) /
EF455 (-5, -7, -8) /
EF489 (-8 THRU -1) /
EF490 (-8 THRU -1) /
EF491 (-8 THRU -1) /
EF492 (-8 THRU -1) /
EF493 (-8 THRU -1) /
EF494 (-8 THRU -1) /
EF495 (-8 THRU -1) /
EF497 (-8 THRU -1) /

EF499 (-8 THRU -1) /
EF500 (-8 THRU -1) /
EF501 (-8 THRU -1) /
EF503 (-8 THRU -1) /
EF504 (-8 THRU -1) /
EF521 (-8 THRU -3) /
EF540 (-3, -7, -8) /
EF541 (-8 THRU -3) /
EF542a (-8 THRU -2) /
EF542b (-8 THRU -2) /
EF542c (-8 THRU -3) /
EF543 (-7, -8) /
EF544 (-8 THRU -2) /
EF545 (-8 THRU -2) /
EF546 (-8 THRU -2) /
EF635 (-8 THRU -1) /
EF636 (-8 THRU -1) /
EF637 (-8 THRU -1) /
EF638 (-8 THRU -1) /
EF639 (-8 THRU -1) /
EF640 (-8 THRU -1) /
EF641 (-8 THRU -1) /
EF642 (-7, -8) /
EF643 (-8 THRU -1) /
EF644 (-8 THRU -1) /
EF645 (-6, -7, -8) /
EF646 (-8 THRU -1) /
EF647 (-8 THRU -1) /
EF648 (-8 THRU -1) /
EF649 (-8 THRU -1) /
EF650 (-8 THRU -1) /
EF651 (-8 THRU -1) /
EF652 (-8 THRU -1) /
EF660 (-1, -7, -8) /
EF661 (-1, -7, -8) /
EF662 (-1, -7, -8) /
EF663 (-1, -7, -8) /
EF664 (-1, -7, -8) /
EF665 (-1, -7, -8) /
EF666 (-1, -7, -8) /
EF667 (-1, -7, -8) /
EF668 (-1, -7, -8) /
EF669 (-1, -7, -8) /
EF670 (-1, -7, -8) /
EF671 (-1, -7, -8) /
EF672 (-1, -7, -8) /
EF673 (-1, -7, -8) /
EF674 (-1, -7, -8) /
EF675 (-1, -7, -8) /
EF676 (-1, -7, -8) /
EF677 (-1, -7, -8) /
EF678 (-1, -7, -8) /
EF679 (-1, -7, -8) /
EF680 (-1, -7, -8) /
EF681 (-1, -7, -8) /
EF682 (-1, -7, -8) /
EF683 (-1, -7, -8) /
EF684 (-1, -7, -8) /
EF685 (-1, -7, -8) /
EF686 (-1, -7, -8) /
EF687 (-1, -7, -8) /
EF688 (-1, -7, -8) /
EF689 (-1, -7, -8) /
EF690 (-1, -7, -8) /
EF691 (-1, -7, -8) /
EF692 (-1, -7, -8) /
EF693 (-1, -7, -8) /
EF694 (-1, -7, -8) /
EF695 (-1, -7, -8) /
EF696 (-1, -7, -8) /
EF697 (-1, -7, -8) /
EF698 (-1, -7, -8) /
EF699 (-1, -7, -8) /
EF700 (-1, -7, -8) /
EF701 (-1, -7, -8) /
EF702 (-1, -7, -8) /
EF703 (-1, -7, -8) /
EF704 (-1, -7, -8) /
EF705 (-1, -7, -8) /
EF706 (-1, -7, -8) /
EF707 (-1, -7, -8) /
EF708 (-1, -7, -8) /

EF709 (-1, -7, -8) /
EF710 (-1, -7, -8) /
EF711 (-1, -7, -8) /
EF712 (-1, -7, -8) /
EF713 (-1, -7, -8) /
EF714 (-1, -7, -8) /
EF715 (-1, -7, -8) /
EF716 (-1, -7, -8) /
EF717 (-8 THRU -1) /
EF718 (-8 THRU -1) /
EF719 (-1, -7, -8) /
EF721 (-1, -7, -8) /
EF722 (-1, -7, -8) /
EF723 (-1, -7, -8) /
EF724 (-1, -7, -8) /
EF725 (-1, -7, -8) /
EF727 (-8 THRU -1) /
EF731 (-1, -7, -8) /
EF732 (-1, -7, -8) /
EF734 (-1, -7, -8) /
mig heb (-1, -7, -8) /
migs heb (-1, -7, -8) /
EF735 (-1, -7, -8) /
EF736 (-1, -7, -8) /
EF737 (-8 THRU -1) /
EF738 (-8 THRU -1) /
EF739 (-8 THRU -1) /
EF740 (-8 THRU -1) /
EF741 (-1, -7, -8) /
EF742 (-1, -7, -8) /
EF743 (-1, -7, -8) /
EF744 (-1, -7, -8) /
EF745 (-1, -7, -8) /
EF746 (-1, -7, -8) /
EF747 (-1, -7, -8) /
EF753 (-1, -7, -8) /
EF754 (-1, -7, -8) /
EF756 (-1, -7, -8) /
EF757 (-1, -7, -8) /
EF758 (-1, -7, -8) /
EF759 (-1, -7, -8) /
EF760 (-8 THRU -1) /
EF761 (-1, -7, -8) /
EF762 (-1, -7, -8) /
EF763 (-1, -7, -8) /
EF764 (-8 THRU -1) /
EF765 (-7, -8) /
EF770 (-1, -7, -8) /
EF771 (-1, -7, -8) /
EF772 (-1, -7, -8) /
EF773 (-1, -7, -8) /
EF774 (-1, -7, -8) /
EF775 (-1, -7, -8) /
EF776 (-1, -7, -8) /
EF777 (-1, -7, -8) /
EF778 (-1, -7, -8) /
EF779 (-1, -7, -8) /
EF780 (-1, -7, -8) /
EF781 (-1, -7, -8) /
EF782 (-1, -7, -8) /
EF783 (-1, -7, -8) /
EF784 (-1, -7, -8) /
EF785 (-1, -7, -8) /
EF786 (-1, -7, -8) /
EF787 (-1, -7, -8) /
EF788 (-1, -7, -8) /
EF789 (-1, -7, -8) /
EF790 (-1, -7, -8) /
EF791 (-1, -7, -8) /
EF792 (-1, -7, -8) /
EF793 (-1, -7, -8) /
EF794 (-1, -7, -8) /
EF795 (-1, -7, -8) /
EF796 (-1, -7, -8) /
EF797 (-1, -7, -8) /
EF798 (-1, -7, -8) /
EF799 (-1, -7, -8) /
EF800 (-1, -7, -8) /
EF801 (-1, -7, -8) /
EF802 (-1, -7, -8) /
EF803 (-8 THRU -1) /
EF804 (-8 THRU -1) /

```
EF805 ( -8 THRU -1 ) /
EF806 ( -1, -7, -8 ) /
EF807 ( -5, -7, -8 ) /
EF808 ( -1, -7, -8 ) /
EF809 ( -1, -7, -8 ) /
EF810 ( -8 THRU -1 ) /
EF811 ( -8 THRU -1 ) /
EF812 ( -1, -7, -8 ) /
EF819 ( -1, -7, -8 ) /
EF820 ( -1, -7, -8 ) /
EF822 ( -1, -7, -8 ) /
mig_lf1 ( -1, -7, -8 ) /
migs_lf1 ( -1, -7, -8 ) /
EF823 ( -1, -7, -8 ) /
EF824 ( -1, -7, -8 ) /
EF825 ( -8 THRU -1 ) /
EF826 ( -8 THRU -1 ) /
EF827 ( -8 THRU -1 ) /
EF828 ( -8 THRU -1 ) /
EF829 ( -1, -7, -8 ) /
EF830 ( -1, -7, -8 ) /
EF831 ( -1, -7, -8 ) /
EF832 ( -1, -7, -8 ) /
EF833 ( -8 THRU -1 ) /
EF834 ( -8 THRU -1 ) /
EF835 ( -8 THRU -2 ) /
EF836 ( -8 THRU -1 ) /
EF837 ( -8 THRU -1 ) /
EF838 ( -8 THRU -1 ) /
EF843 ( -8 THRU -1 ) /
EF844 ( -8 THRU -1 ) /
EF846 ( -8 THRU -1 ) /
mig_lf2 ( -8 THRU -1 ) /
migs_lf2 ( -8 THRU -1 ) /
EF847 ( -8 THRU -1 ) /
EF848 ( -8 THRU -1 ) /
EF849 ( -8 THRU -1 ) /
EF850 ( -8 THRU -1 ) /
EF851 ( -8 THRU -1 ) /
EF852 ( -8 THRU -1 ) /
EF853 ( -8 THRU -1 ) /
EF854 ( -8 THRU -1 ) /
EF855 ( -8 THRU -1 ) /
EF856 ( -8 THRU -1 ) /
EF857 ( -8 THRU -1 ) /
EF858 ( -8 THRU -1 ) /
EF859 ( -8 THRU -1 ) /
EF860 ( -8 THRU -1 ) /
EF861 ( -8 THRU -1 ) /
EF862 ( -8 THRU -1 ) /
EF863 ( -1, -7, -8 ) /
EF864 ( -1, -7, -8 ) /
EF865 ( -1, -7, -8 ) /
EF866 ( -1, -7, -8 ) /
EF867 ( -1, -7, -8 ) /
EF986 ( -8 THRU -1 ) /
EF987 ( -8 THRU -1 ) /
EF988 ( -8 THRU -1 ) /
EF989 ( -8 THRU -1 ) /
EF990 ( -8 THRU -1 ) /
EF991 ( -8 THRU -1 ) /
EF992 ( -8 THRU -1 ) /
EF993 ( -8 THRU -1 ) /
EF994 ( -8 THRU -1 ) /
EF995 ( -8 THRU -1 ) /
EF1000u2 ( -8 THRU -1 ) /
EF1000u3 ( -8 THRU -1 ) /
EF1000u4 ( -8 THRU -1 ) /
EF1000u5 ( -8 THRU -1 ) /
EF1000u6 ( -8 THRU -1 ) /
EF1000u7 ( -8 THRU -1 ) /
EF1000u8 ( -8 THRU -1 ) /
EF1000u9 ( -8 THRU -1 ) /
EF1000u10 ( -8 THRU -1 ) /
EF1000u11 ( -8 THRU -1 ) /
EF1001u2 ( -8 THRU -1 ) /
EF1001u3 ( -8 THRU -1 ) /
EF1001u4 ( -8 THRU -1 ) /
EF1001u5 ( -8 THRU -1 ) /
EF1001u6 ( -8 THRU -1 ) /
EF1001u7 ( -8 THRU -1 ) /
EF1001u8 ( -8 THRU -1 ) /
```



```
EF1001u9 ( -8 THRU -1 ) /
EF1001u10 ( -8 THRU -1 ) /
EF1001u11 ( -8 THRU -1 ) /
EF1004 ( -8 THRU -1 ) /
EF1005 ( -8 THRU -1 ) /
EF1006 ( -8 THRU -1 ) /
EF1007 ( -8 THRU -1 ) /
EF1008 ( -7, -8 ) /
EF1009 ( -8 THRU -1 ) /
EF1010 ( -8 THRU -1 ) /
EF1011 ( -8 THRU -1 ) /
EF1012 ( -8 THRU -1 ) /
EF1013 ( -8 THRU -1 ) /
EF1014 ( -7, -8 ) /
EF1015 ( -7, -8 ) /
EF1016 ( -7, -8 ) /
EF1017 ( -7, -8 ) /
EF1080 ( -8 THRU -3 ) /
EF1081 ( -8 THRU -3 ) /
EF1082 ( -8 THRU -3 ) /
EF1083 ( -8 THRU -3 ) /
EF1084 ( -8 THRU -3 ) /
EF1085 ( -8 THRU -3 ) /
EF1086 ( -8 THRU -3 ) /
EF1087 ( -8 THRU -3 ) /
EF1088 ( -8 THRU -3 ) /
EF1089 ( -8 THRU -3 ) /
EF1090 ( -8 THRU -3 ) /
EF1091 ( -8 THRU -3 ) /
EF1092 ( -8 THRU -3 ) /
EF1093 ( -8 THRU -3 ) /
EF1094 ( -8 THRU -3 ) /
EF1095 ( -8 THRU -3 ) /
EF1096 ( -8 THRU -3 ) /
EF1097 ( -8 THRU -3 ) /
EF1098 ( -8 THRU -3 ) /
EF1100 ( -8 THRU -3 ) /
EF1101 ( -8 THRU -3 ) /
EF1102 ( -8 THRU -3 ) /
EF1103 ( -8 THRU -3 ) /
EF1104 ( -8 THRU -3 ) /
EF1105 ( -8 THRU -3 ) /
EF1106 ( -8 THRU -3 ) /
EF1107 ( -8 THRU -3 ) /
EF1108 ( -8 THRU -3 ) /
EF1109 ( -8 THRU -3 ) /
EF1110 ( -8 THRU -3 ) /
EF1111 ( -8 THRU -3 ) /
EF1112 ( -8 THRU -3 ) /
EF1113 ( -8 THRU -3 ) /
EF1114 ( -8 THRU -3 ) /
EF1115 ( -8 THRU -3 ) /
EF2001 ( -7, -8 ) /
EF2002 ( -8 THRU -1 ) /
EF2003 ( -8 THRU -1 ) /
EF2004 ( -8 THRU -1 ) /
EF2005 ( -8 THRU -1 ) /
EF2006 ( -7, -8 ) /
EF2007 ( -7, -8 ) /
EF2008 ( -7, -8 ) /
EF2009 ( -7, -8 ) /
EF2011 ( -1, -7, -8 ) /
EF2012 ( -1, -7, -8 ) /
v663 ( -8 THRU -1 ) /
v664 ( -8 THRU -1 ) /
v665 ( -8 THRU -1 ) /
v666 ( -8 THRU -1 ) /
v667 ( -8 THRU -1 ) /
v668 ( -8 THRU -1 ) /
v669 ( -8 THRU -1 ) /
v670 ( -8 THRU -1 ) /
v671 ( -8 THRU -1 ) /
v672 ( -8 THRU -1 ) /
v673 ( -8 THRU -1 ) /
v674 ( -8 THRU -1 ) /
v675 ( -8 THRU -1 ) /
v676 ( -8 THRU -1 ) /
v677 ( -8 THRU -1 ) /
v678 ( -8 THRU -1 ) /
v679 ( -8 THRU -1 ) /
v680 ( -8 THRU -1 ) /
v681 ( -8 THRU -1 ) /
```

```
v682 ( -8 THRU -1 ) /
v683 ( -8 THRU -1 ) /
v684 ( -8 THRU -1 ) /
v685 ( -8 THRU -1 ) /
v686 ( -8 THRU -1 ) /
v687 ( -8 THRU -1 ) /
v688 ( -8 THRU -1 ) /
v689 ( -8 THRU -1 ) /
v690 ( -8 THRU -1 ) /
v691 ( -8 THRU -1 ) /
v692 ( -8 THRU -1 ) /
v693 ( -8 THRU -1 ) /
v694 ( -8 THRU -1 ) /
v695 ( -8 THRU -1 ) /
v696 ( -8 THRU -1 ) /
v697 ( -8 THRU -1 ) /
v698 ( -8 THRU -1 ) /
v699 ( -8 THRU -1 ) /
v700 ( -8 THRU -1 ) /
v701 ( -8 THRU -1 ) /
v702 ( -8 THRU -1 ) /
v703 ( -8 THRU -1 ) /
v704 ( -8 THRU -1 ) /
v705 ( -8 THRU -1 ) /
v706 ( -8 THRU -1 ) /
v709 ( -8 THRU -1 ) /
v711 ( -8 THRU -1 ) /
v712 ( -8 THRU -1 ) /
v713 ( -8 THRU -1 ) /
v714 ( -8 THRU -1 ) /
v715 ( -8 THRU -1 ) /
v716 ( -8 THRU -1 ) /
v719 ( -8 THRU -1 ) /
v727 ( -8 THRU -1 ) /
v731 ( -8 THRU -1 ) /
v732 ( -8 THRU -1 ) /
v734 ( -8 THRU -1 ) /
vmig_heb ( -8 THRU -1 ) /
vmigs_heb ( -8 THRU -1 ) /
v735 ( -8 THRU -1 ) /
v736 ( -8 THRU -1 ) /
v737 ( -8 THRU -1 ) /
v738 ( -8 THRU -1 ) /
v739 ( -8 THRU -1 ) /
v740 ( -8 THRU -1 ) /
v741 ( -8 THRU -1 ) /
v742 ( -8 THRU -1 ) /
v743 ( -8 THRU -1 ) /
v744 ( -8 THRU -1 ) /
v745 ( -8 THRU -1 ) /
v753 ( -8 THRU -1 ) /
v754 ( -8 THRU -1 ) /
v756 ( -8 THRU -1 ) /
v757 ( -8 THRU -1 ) /
v758 ( -8 THRU -1 ) /
v759 ( -8 THRU -1 ) /
v760 ( -8 THRU -1 ) /
v770 ( -8 THRU -1 ) /
v771 ( -8 THRU -1 ) /
v772 ( -8 THRU -1 ) /
v773 ( -8 THRU -1 ) /
v774 ( -8 THRU -1 ) /
v775 ( -8 THRU -1 ) /
v776 ( -8 THRU -1 ) /
v777 ( -8 THRU -1 ) /
v778 ( -8 THRU -1 ) /
v779 ( -8 THRU -1 ) /
v780 ( -8 THRU -1 ) /
v781 ( -8 THRU -1 ) /
v782 ( -8 THRU -1 ) /
v783 ( -8 THRU -1 ) /
v784 ( -8 THRU -1 ) /
v785 ( -8 THRU -1 ) /
v786 ( -8 THRU -1 ) /
v787 ( -8 THRU -1 ) /
v788 ( -8 THRU -1 ) /
v789 ( -8 THRU -1 ) /
v790 ( -8 THRU -1 ) /
v791 ( -8 THRU -1 ) /
v792 ( -8 THRU -1 ) /
v793 ( -8 THRU -1 ) /
v794 ( -8 THRU -1 ) /
```

```
v795 ( -8 THRU -1 ) /
v796 ( -8 THRU -1 ) /
v797 ( -8 THRU -1 ) /
v798 ( -8 THRU -1 ) /
v799 ( -8 THRU -1 ) /
v800 ( -8 THRU -1 ) /
v801 ( -8 THRU -1 ) /
v802 ( -8 THRU -1 ) /
v803 ( -8 THRU -1 ) /
v804 (-5) /
v812 ( -8 THRU -1 ) /
v819 ( -8 THRU -1 ) /
v820 ( -8 THRU -1 ) /
v822 ( -8 THRU -1 ) /
vmig_lf1 ( -8 THRU -1 ) /
vmigs_lf ( -8 THRU -1 ) /
v823 ( -8 THRU -1 ) /
v824 ( -8 THRU -1 ) /
v825 ( -8 THRU -1 ) /
v826 ( -8 THRU -1 ) /
v827 ( -8 THRU -1 ) /
v828 ( -8 THRU -1 ) /
v829 ( -8 THRU -1 ) /
v831 ( -8 THRU -1 ) /
v832 ( -8 THRU -1 ) /
v833 ( -8 THRU -1 ) /
v834 ( -8 THRU -1 ) /
v835 ( -8 THRU -1 ) /
v836 ( -8 THRU -1 ) /
v837 ( -8 THRU -1 ) /
v838 ( -8 THRU -1 ) /
v843 ( -8 THRU -1 ) /
v844 ( -8 THRU -1 ) /
v846 ( -8 THRU -1 ) /
vmig_lf2 ( -8 THRU -1 ) /
vmigs_lf2 ( -8 THRU -1 ) /
v846 ( -8 THRU -1 ) /
v847 ( -8 THRU -1 ) /
v848 ( -8 THRU -1 ) /
v849 ( -8 THRU -1 ) /
v850 ( -8 THRU -1 ) /
v851 ( -8 THRU -1 ) /
v852 ( -8 THRU -1 ) /
v853 ( -8 THRU -1 ) /
v854 ( -8 THRU -1 ) /
v855 ( -8 THRU -1 ) /
v856 ( -8 THRU -1 ) /
v857 ( -8 THRU -1 ) /
v858 ( -8 THRU -1 ) /
v859 ( -8 THRU -1 ) /
v860 ( -8 THRU -1 ) /
v861 ( -8 THRU -1 ) /
v862 ( -8 THRU -1 ) /
v866 ( -8 THRU -1 ) /
v867 ( -8 THRU -1 )
```

.

* Ende der Syntax.

6.4 Syntax für SAS (Version 1.0)

```

/* Dateiformat: Dos\Windows */
/* Encoding: ANSI (Windows-1252) */
/* SAS 9.4 for Windows */

/* SAS Syntax zum GESIS Paper:
   "Einführung in die eigenständige Erstellung von Typisierungen
   am Beispiel des Mikrozensus Scientific Use Files 2014"
   Diese Syntax ist abgelegt unter: https://www.gesis.org/missy/materials/MZ/tools/typisierungen

Autor: Simon Börlin
GESIS - Leibniz-Institut für Sozialwissenschaften
German Microdata Lab (GML)
Postfach 12 21 55
68072 Mannheim
https://www.gesis.org/gml/gml-home/
E-Mail: simon.boerlin@gesis.org
*/

/* Version 1.0 (17.12.2019) */

OPTIONS NOFMterr; /* keine Formate */

* EF663: Personen im Haushalt: Anzahl
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<MZ14.suf2014>>> (keep = EF1 /* Bundesland */
                        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
                        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
                        EF5b /* Jahresüberhang aus dem Vorjahr */
                        EF12 /* Berichtsquartal */
                        EF31 /* Bevölkerung: Privathaushalte */
                        EF663) /* Personen im Haushalt: Anzahl (Haushaltstypisierung) */ ;
run;

/* Sortieren nach IDs ~ Haushaltsnummer

Im MZ ab 2005 sind Haushalte mittels der Variablen bzw. Ordnungsnummern (IDs)
- Bundesland (EF1),
- Auswahlbezirksnummer (EF3),
- Nummer des Haushalts im Auswahlbezirk (EF4),
- Jahresüberhang aus dem Vorjahr (EF5b) und
- Berichtsquartal (EF12)
eindeutig identifiziert. */
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Privathaushalten, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var EF31;
  output out = sum(drop=_type_ _freq_ )
    sum = v663;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung (EF663>=9) replizieren und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v663 >= 9 & v663 <= 99) then v663 = 9; /* 9 und mehr Personen */
  if (v663 = . & EF31 = -8) then v663 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v663 = . & EF31 = -7) then v663 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v663 = . & EF31 = 2) then v663 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle: OK;
proc freq data = tmp14 (where= (v663 ~= EF663));
  table v663 * EF663;
run;

* EF664: Erwerbstätige im Haushalt: Anzahl
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<MZ14.suf2014>>> (keep = EF1 /* Bundesland */
                        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
                        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
                        EF5b /* Jahresüberhang aus dem Vorjahr */
                        EF12 /* Berichtsquartal */
                        EF31 /* Bevölkerung: Privathaushalte */
                        EF29 /* Erwerbsstatus */
                        EF664) /* Erwerbstätige im Haushalt: Anzahl (Haushaltstypisierung) */;
  x664 = (EF31=1 & EF29=1); /* Privathaushalt, Erwerbstätiger */
run;

* Sortieren nach IDs ;

```

```

proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der erwerbstätigen Personen in Privathaushalten, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF29 EF31 x664 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x664;
  output out = sum(drop=_type_ _freq_ )
  sum = v664;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v664 = . & EF31 = -8) then v664 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v664 = . & EF31 = -7) then v664 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v664 = . & EF31 = 2) then v664 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v664 ~= EF664));
  table v664 * EF664;
run;

* EF665: Erwerbslose (EU-Definition) im Haushalt: Anzahl
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF29 /* Erwerbsstatus */
  EF665) /* Erwerbslose (EU-Definition) im Haushalt: Anzahl (Haushaltstypisierung) */
  ;
  x665 = (EF31=1 & EF29=2) /* Privathaushalt | Erwerbslos */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der erwerbslosen Personen (EU-Definition) in Privathaushalten, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF29 EF31 x665 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x665;
  output out = sum(drop=_type_ _freq_ )
  sum = v665;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v665 = . & EF31 = -8) then v665 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v665 = . & EF31 = -7) then v665 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v665 = . & EF31 = 2) then v665 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v665 ~= EF665));
  table v665 * EF665;
run;

* EF666: Arbeitsuchende Nichterwerbspersonen im Haushalt: Anzahl
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF29 /* Erwerbsstatus */
  EF666) /* Arbeitsuchende Nichterwerbspersonen im Haushalt: Anzahl (Haushaltstypisie-
  rung) */ ;
  x666 = (EF31=1 & EF29=3) /* Privathaushalt | arbeitssuchende Nichterwerbsperson */;

```

```

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der arbeitsuchenden Nichterwerbspersonen in Privathaushalten, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF29 EF31 x666 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x666;
output out = sum(drop=_type_ _freq_ )
sum = v666;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v666 = . & EF31 = -8) then v666 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v666 = . & EF31 = -7) then v666 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v666 = . & EF31 = 2) then v666 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v666 ~= EF666));
  table v666 * EF666;
run;

* EF667: Einkommensbezieher im Haushalt: Anzahl
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF436 /* Nettoeinkommen im letzten Monat */
EF667) /* Einkommensbezieher im Haushalt: Anzahl (Haushaltstypisierung) */ ;
  x667 = (EF31=1 & EF436~=90 & EF436~=99) /* Privathaushalt | monatliches Einkommen
vorhanden */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Einkommensbezieher in Privathaushalten, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF436 x667 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x667;
output out = sum(drop=_type_ _freq_ )
sum = v667;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v667 >= 9 & v667 <=99) then v667 = 9;
  if (v667 = . & EF31 = -8) then v667 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v667 = . & EF31 = -7) then v667 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v667 = . & EF31 = 2) then v667 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v667 ~= EF667));
  table v667 * EF667;
run;

* EF668: Ausländer im Haushalt: Anzahl
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */

```

```

EF368 /* Deutsche Staatsangehörigkeit */
EF668 /* Ausländer im Haushalt: Anzahl (Haushaltstypisierung) */ ;
x668 = (EF31=1 & EF368=8) /* Privathaushalt | ausländische Staatsangehörigkeit */ ;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Ausländer in Privathaushalten, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF368 x668 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x668;
  output out = sum(drop=_type_ _freq_ )
  sum = v668;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung replizieren und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v668 >=9 & v668 <= 99) then v668 = 9; /* und mehr Personen */
  if (v668 = . & EF31 = -8) then v668 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v668 = . & EF31 = -7) then v668 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v668 = . & EF31 = 2) then v668 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v668 ~= EF668));
  table v668 * EF668;
run;

* EF669: Kinder: Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung innerhalb der Lebensform */
  EF669 /* Kinder: Anzahl im Haushalt (Haushaltstypisierung) */ ;
  x669 = (EF31=1 & EF35=3) /* Privathaushalt | Kind in Lebensform */ ;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 x669 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x669;
  output out = sum(drop=_type_ _freq_ )
  sum = v669;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung replizieren und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v669 >=9 & v669 <= 99) then v669 = 9; /* und mehr Personen */
  if (v669 = . & EF31 = -8) then v669 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v669 = . & EF31 = -7) then v669 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v669 = . & EF31 = 2) then v669 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v669 ~= EF669));
  table v669 * EF669;
run;

* EF670: Kinder (unter 1 Jahr): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */

```

```

EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF670 /* Kinder (unter 1 Jahr): Anzahl im Haushalt (Haushaltstypisierung) */;
x670 = (EF31=1 & EF35=3 & EF44<1) /* Privathaushalt | Kind in Lebensform | unter 1

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<1, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x670 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x670;
  output out = sum(drop=_type_ _freq_ )
  sum = v670;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung replizieren und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v670 = . & EF31 = -8) then v670 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v670 = . & EF31 = -7) then v670 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v670 = . & EF31 = 2) then v670 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v670 ~= EF670));
  table v670 * EF670;
run;

* EF671: Kinder (unter 2 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF671 /* Kinder (unter 2 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x671 = (EF31=1 & EF35=3 & EF44<2) /* Privathaushalt | Kind in Lebensform | unter 2

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<2, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x671 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x671;
  output out = sum(drop=_type_ _freq_ )
  sum = v671;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v671 = . & EF31 = -8) then v671 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v671 = . & EF31 = -7) then v671 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v671 = . & EF31 = 2) then v671 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v671 ~= EF671));
  table v671 * EF671;
run;

* EF672: Kinder (unter 3 Jahren): Anzahl im Haushalt

```



```

* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF672) /* Kinder (unter 3 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x672 = (EF31=1 & EF35=3 & EF44<3) /* Privathaushalt | Kind in Lebensform | unter 3
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<3, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x672 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x672;
  output out = sum(drop=_type_ _freq_ )
  sum = v672;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v672 = . & EF31 = -8) then v672 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v672 = . & EF31 = -7) then v672 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v672 = . & EF31 = 2) then v672 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v672 ~= EF672));
  table v672 * EF672;
run;

* EF673: Kinder (unter 4 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF673) /* Kinder (unter 4 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x673 = (EF31=1 & EF35=3 & EF44<4) /* Privathaushalt | Kind in Lebensform | unter 4
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<4, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x673 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x673;
  output out = sum(drop=_type_ _freq_ )
  sum = v673;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v673 = . & EF31 = -8) then v673 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v673 = . & EF31 = -7) then v673 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v673 = . & EF31 = 2) then v673 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;

```

```

proc freq data = tmp14 (where= (v673 ~= EF673));
  table v673 * EF673;
run;

* EF674: Kinder (unter 5 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF674) /* Kinder (unter 5 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
  x674 = (EF31=1 & EF35=3 & EF44<5) /* Privathaushalt | Kind in Lebensform | unter 5
  Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<5, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x674 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x674;
  output out = sum(drop=_type_ _freq_ )
  sum = v674;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v674 = . & EF31 = -8) then v674 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v674 = . & EF31 = -7) then v674 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v674 = . & EF31 = 2) then v674 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v674 ~= EF674));
  table v674 * EF674;
run;

* EF675: Kinder (unter 6 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF675) /* Kinder (unter 6 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
  x675 = (EF31=1 & EF35=3 & EF44<6) /* Privathaushalt | Kind in Lebensform | unter 6
  Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<6, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x675 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x675;
  output out = sum(drop=_type_ _freq_ )
  sum = v675;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;

```

```

    if (v675 = . & EF31 = -8) then v675 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v675 = . & EF31 = -7) then v675 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v675 = . & EF31 = 2) then v675 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v675 ~= EF675));
    table v675 * EF675;
run;

* EF676: Kinder (unter 7 Jahren): Anzahl im Haushalt;
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF676 /* Kinder (unter 7 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
    x676 = (EF31=1 & EF35=3 & EF44<7) /* Privathaushalt | Kind in Lebensform | unter 7

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* * Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<7, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x676 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x676;
    output out = sum(drop=_type_ _freq_ )
        sum = v676;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v676 = . & EF31 = -8) then v676 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v676 = . & EF31 = -7) then v676 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v676 = . & EF31 = 2) then v676 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v676 ~= EF676));
    table v676 * EF676;
run;

* EF677: Kinder (unter 8 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF677 /* Kinder (unter 8 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
    x677 = (EF31=1 & EF35=3 & EF44<8) /* Privathaushalt | Kind in Lebensform | unter 8

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<8, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x677 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x677;
    output out = sum(drop=_type_ _freq_ )
        sum = v677;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
    merge tmp14 sum;

```

```

    by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung (EF665>=9) replizieren und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v677 = . & EF31 = -8) then v677 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v677 = . & EF31 = -7) then v677 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v677 = . & EF31 = 2) then v677 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v677 ~= EF677));
table v677 * EF677;
run;

* EF678: Kinder (unter 9 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF678) /* Kinder (unter 9 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x678 = (EF31=1 & EF35=3 & EF44<9) /* Privathaushalt | Kind in Lebensform | unter 9
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<9, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x678 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x678;
output out = sum(drop=_type_ _freq_ )
sum = v678;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v678 = . & EF31 = -8) then v678 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v678 = . & EF31 = -7) then v678 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v678 = . & EF31 = 2) then v678 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v678 ~= EF678));
table v678 * EF678;
run;

* EF679: Kinder (unter 10 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF679) /* Kinder (unter 10 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x679 = (EF31=1 & EF35=3 & EF44<10) /* Privathaushalt | Kind in Lebensform | unter 10
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<10, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x679 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x679;
output out = sum(drop=_type_ _freq_ )

```

```

sum = v679;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung (EF665>=9) replizieren und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v679 = . & EF31 = -8) then v679 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v679 = . & EF31 = -7) then v679 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v679 = . & EF31 = 2) then v679 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v679 ~= EF679));
table v679 * EF679;
run;

* EF680: Kinder (unter 11 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF680 /* Kinder (unter 11 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x680 = (EF31=1 & EF35=3 & EF44<11) /* Privathaushalt | Kind in Lebensform | unter 11
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<11, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x680 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x680;
output out = sum(drop=_type_ _freq_)
sum = v680;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v680 = . & EF31 = -8) then v680 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v680 = . & EF31 = -7) then v680 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v680 = . & EF31 = 2) then v680 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v680 ~= EF680));
table v680 * EF680;
run;

* EF681: Kinder (unter 12 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF681 /* Kinder (unter 12 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x681 = (EF31=1 & EF35=3 & EF44<12) /* Privathaushalt | Kind in Lebensform | unter 12
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

```

```

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<12, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x681 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x681;
output out = sum(drop=_type_ _freq_ )
sum = v681;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v681 = . & EF31 = -8) then v681 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v681 = . & EF31 = -7) then v681 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v681 = . & EF31 = 2) then v681 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v681 ~= EF681));
table v681 * EF681;
run;

* EF682: Kinder (unter 13 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF682) /* Kinder (unter 13 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x682 = (EF31=1 & EF35=3 & EF44<13) /* Privathaushalt | Kind in Lebensform | unter 13

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<13, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x682 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x682;
output out = sum(drop=_type_ _freq_ )
sum = v682;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v682 = . & EF31 = -8) then v682 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v682 = . & EF31 = -7) then v682 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v682 = . & EF31 = 2) then v682 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v682 ~= EF682));
table v682 * EF682;
run;

* EF683: Kinder (unter 14 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF683) /* Kinder (unter 14 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x683 = (EF31=1 & EF35=3 & EF44<14) /* Privathaushalt | Kind in Lebensform | unter 14

Jahre alt */;
run;

```

```

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<14, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x683 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x683;
  output out = sum(drop=_type_ _freq_ )
  sum = v683;
run;

* Zusammenspielen ;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v683 = . & EF31 = -8) then v683 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v683 = . & EF31 = -7) then v683 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v683 = . & EF31 = 2) then v683 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v683 ~= EF683));
  table v683 * EF683;
run;

* EF684: Kinder (unter 15 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF684 /* Kinder (unter 15 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
  x684 = (EF31=1 & EF35=3 & EF44<15) /* Privathaushalt | Kind in Lebensform | unter 15
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<15, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x684 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x684;
  output out = sum(drop=_type_ _freq_ )
  sum = v684;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v684 = . & EF31 = -8) then v684 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v684 = . & EF31 = -7) then v684 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v684 = . & EF31 = 2) then v684 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v684 ~= EF684));
  table v684 * EF684;
run;

* EF685: Kinder (unter 16 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */

```

```

EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF685) /* Kinder (unter 16 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x685 = (EF31=1 & EF35=3 & EF44<16) /* Privathaushalt | Kind in Lebensform | unter 16

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<16, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x685 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x685;
  output out = sum(drop= type freq )
  sum = v685;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v685 = . & EF31 = -8) then v685 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v685 = . & EF31 = -7) then v685 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v685 = . & EF31 = 2) then v685 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v685 ~= EF685));
  table v685 * EF685;
run;

* EF686: Kinder (unter 17 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF686) /* Kinder (unter 17 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x686 = (EF31=1 & EF35=3 & EF44<17) /* Privathaushalt | Kind in Lebensform | unter 17

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<17, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x686 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x686;
  output out = sum(drop=_type_ _freq_ )
  sum = v686;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v686 = . & EF31 = -8) then v686 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v686 = . & EF31 = -7) then v686 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v686 = . & EF31 = 2) then v686 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v686 ~= EF686));
  table v686 * EF686;
run;

* EF687: Kinder (unter 18 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);

```



```

set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF687) /* Kinder (unter 18 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x687 = (EF31=1 & EF35=3 & EF44<18) /* Privathaushalt | Kind in Lebensform | unter 18

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<18, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x687 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x687;
output out = sum(drop=_type_ _freq_ )
sum = v687;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v687 >=9 & v687 <=99) then v687 = 9;
if (v687 = . & EF31 = -8) then v687 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v687 = . & EF31 = -7) then v687 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v687 = . & EF31 = 2) then v687 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v687 ~= EF687));
table v687 * EF687;
run;

* EF688: Kinder (unter 27 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF688) /* Kinder (unter 27 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x688 = (EF31=1 & EF35=3 & EF44<27) /* Privathaushalt | Kind in Lebensform | unter 27

Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<27, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x688 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x688;
output out = sum(drop=_type_ _freq_ )
sum = v688;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v688 >=9 & v688 <=99) then v688 = 9;
if (v688 = . & EF31 = -8) then v688 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v688 = . & EF31 = -7) then v688 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v688 = . & EF31 = 2) then v688 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;

```

```

proc freq data = tmp14 (where= (v688 ~= EF688));
  table v688 * EF688;
run;

* EF689: Kinder (ab 18 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF689) /* Kinder (ab 18 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
  x689 = (EF31=1 & EF35=3 & EF44>=18) /* Privathaushalt | Kind in Lebensform | ab 18
Jahre */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44>=18, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x689 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x689;
  output out = sum(drop=_type_ _freq_ )
  sum = v689;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v689 >=9 & v689 <=99) then v689 = 9;
  if (v689 = . & EF31 = -8) then v689 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v689 = . & EF31 = -7) then v689 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v689 = . & EF31 = 2) then v689 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v689 ~= EF689));
  table v689 * EF689;
run;

* EF690: Kinder (ab 27 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF690) /* Kinder (ab 27 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
  x690 = (EF31=1 & EF35=3 & EF44>=27) /* Privathaushalt | Kind in Lebensform | ab 27
Jahre */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44>=27, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x690 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x690;
  output out = sum(drop=_type_ _freq_ )
  sum = v690;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;

```

```

    if (v690 >=9 & v690 <=99) then v690 = 9;
    if (v690 = . & EF31 = -8) then v690 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v690 = . & EF31 = -7) then v690 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v690 = . & EF31 = 2) then v690 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v690 ~= EF690));
    table v690 * EF690;
run;

* EF691: Kinder (3-5 Jahre): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF691) /* Kinder (3-5 Jahre): Anzahl im Haushalt (Haushaltstypisierung) */;
    x691 = (EF31=1 & EF35=3 & 3<=EF44<=5) /* Privathaushalt | Kind in Lebensform | 3-5
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 3<=EF44<=5, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x691 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x691;
    output out = sum(drop=_type_ _freq_ )
    sum = v691;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v691 = . & EF31 = -8) then v691 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v691 = . & EF31 = -7) then v691 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v691 = . & EF31 = 2) then v691 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v691 ~= EF691));
    table v691 * EF691;
run;

* EF692: Kinder (6-9 Jahre): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF692) /* Kinder (6-9 Jahre): Anzahl im Haushalt (Haushaltstypisierung) */;
    x692 = (EF31=1 & EF35=3 & 6<=EF44<=9) /* Privathaushalt | Kind in Lebensform | 6-9
Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 6<=EF44<=9, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x692 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x692;
    output out = sum(drop=_type_ _freq_ )
    sum = v692;
run;

* Zusammenspielen;
data tmp14 (compress = yes);

```

```

merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v692 = . & EF31 = -8) then v692 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v692 = . & EF31 = -7) then v692 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v692 = . & EF31 = 2) then v692 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v692 ~= EF692));
table v692 * EF692;
run;

* EF693: Kinder (10-14 Jahre): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF693) /* Kinder (10-14 Jahre): Anzahl im Haushalt (Haushaltstypisierung) */;
x693 = (EF31=1 & EF35=3 & 10<=EF44<=14) /* Privathaushalt | Kind in Lebensform | 10-
14 Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 10<=EF44<=14, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x693 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x693;
output out = sum(drop=_type_ _freq_ )
sum = v693;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v693 = . & EF31 = -8) then v693 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v693 = . & EF31 = -7) then v693 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v693 = . & EF31 = 2) then v693 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v693 ~= EF693));
table v693 * EF693;
run;

* EF694: Kinder (15-17 Jahre): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF694) /* Kinder (15-17 Jahre): Anzahl im Haushalt (Haushaltstypisierung) */;
x694 = (EF31=1 & EF35=3 & 15<=EF44<=17) /* Privathaushalt | Kind in Lebensform | 15-
17 Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 15<=EF44<=17, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x694 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x694;

```

```

output out = sum(drop=_type_ _freq_ )
sum = v694;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v694 = . & EF31 = -8) then v694 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v694 = . & EF31 = -7) then v694 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v694 = . & EF31 = 2) then v694 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v694 ~= EF694));
table v694 * EF694;
run;

* EF695: Kinder (18-26 Jahre): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF695) /* Kinder (18-26 Jahre): Anzahl im Haushalt (Haushaltstypisierung) */;
x695 = (EF31=1 & EF35=3 & 18<=EF44<=26) /* Privathaushalt | Kind in Lebensform | 18-
26 Jahre alt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 18<=EF44<=26, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 x695 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x695;
output out = sum(drop=_type_ _freq_ )
sum = v695;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v695 = . & EF31 = -8) then v695 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v695 = . & EF31 = -7) then v695 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v695 = . & EF31 = 2) then v695 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v695 ~= EF695));
table v695 * EF695;
run;

* EF696: Kinder (unter 6 Jahren): Anzahl im Haushalt, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF287 /* Gegenwärtiger Schulbesuch */
EF696) /* Kinder (unter 6 Jahren): Anzahl im Haushalt, Schüler (Haushaltstypisie-
rung) */;
x696 = (EF31=1 & EF35=3 & EF44<6 & (EF287= 1 ! EF287=2)) /* Privathaushalt | Kind in
Lebensform | unter 6 Jahre alt, Schüler */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;

```

```

by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44<6, Schüler, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 EF287 x696 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x696;
output out = sum(drop=_type_ _freq_ )
sum = v696;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v696 = . & EF31 = -8) then v696 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v696 = . & EF31 = -7) then v696 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v696 = . & EF31 = 2) then v696 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v696 ~= EF696));
table v696 * EF696;
run;

* EF697: Kinder (6-9 Jahre): Anzahl im Haushalt, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF287 /* Gegenwärtiger Schulbesuch */
EF697) /* Kinder (6-9 Jahre): Anzahl im Haushalt, Schüler (Haushaltstypisierung) */;
x697 = (EF31=1 & EF35=3 & 6<=EF44<=9 & (EF287= 1 ! EF287=2)) /* Privathaushalt |
Kind in Lebensform | 6-9 Jahre alt | Schüler */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 6<=EF44<=9, Schüler, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 EF287 x697 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x697;
output out = sum(drop=_type_ _freq_ )
sum = v697;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v697 = . & EF31 = -8) then v697 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v697 = . & EF31 = -7) then v697 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v697 = . & EF31 = 2) then v697 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v697 ~= EF697));
table v697 * EF697;
run;

* EF698: Kinder (10-14 Jahre): Anzahl im Haushalt, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */

```

```

                EF287 /* Gegenwärtiger Schulbesuch */
                EF698 /* Kinder (10-14 Jahre): Anzahl im Haushalt, Schüler (Haushaltstypisierung)
*/;
                x698 = (EF31=1 & EF35=3 & 10<=EF44<=14 & (EF287= 1 ! EF287=2)) /* Privathaushalt |
Kind in Lebensform | 10-14 Jahre alt | Schüler */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 10<=EF44<=14, Schüler, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 EF287 x698 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x698;
    output out = sum(drop= type freq )
        sum = v698;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v698 = . & EF31 = -8) then v698 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v698 = . & EF31 = -7) then v698 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v698 = . & EF31 = 2) then v698 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v698 ~= EF698));
    table v698 * EF698;
run;

* EF699: Kinder (15-17 Jahre): Anzahl im Haushalt, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<=mz14.suf2014>> (keep = EF1 /* Bundesland */
                EF3 /* Nr. des Auswahlbezirks (systemfrei) */
                EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
                EF5b /* Jahresüberhang aus dem Vorjahr */
                EF12 /* Berichtsquartal */
                EF31 /* Bevölkerung: Privathaushalte */
                EF35 /* Stellung innerhalb der Lebensform */
                EF44 /* Alter */
                EF287 /* Gegenwärtiger Schulbesuch */
                EF699) /* Kinder (15-17 Jahre): Anzahl im Haushalt, Schüler (Haushaltstypisierung)
*/;
                x699 = (EF31=1 & EF35=3 & 15<=EF44<=17 & (EF287= 1 ! EF287=2)) /* Privathaushalt |
Kind in Lebensform | 15-17 Jahre alt | Schüler */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 15<=EF44<=17, Schüler, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 EF287 x699 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x699;
    output out = sum(drop=_type_ _freq_ )
        sum = v699;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v699 = . & EF31 = -8) then v699 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v699 = . & EF31 = -7) then v699 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v699 = . & EF31 = 2) then v699 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v699 ~= EF699));
    table v699 * EF699;
run;

* EF700: Kinder (18-26 Jahre): Anzahl im Haushalt, Schüler

```

```

* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF287 /* Gegenwärtiger Schulbesuch */
    EF700 /* Kinder (18-26 Jahre): Anzahl im Haushalt, Schüler (Haushaltstypisierung)
  );
  x700 = (EF31=1 & EF35=3 & 18<=EF44<=26 & (EF287= 1 ! EF287=2)) /* Privathaushalt |
  Kind in Lebensform | 18-26 Jahre alt | Schüler */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft 18<=EF44<=26, Schüler, die zur Bevölkerung in Privat-
  haushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 EF287 x700 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x700;
  output out = sum(drop=_type_ _freq_ )
  sum = v700;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v700 = . & EF31 = -8) then v700 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v700 = . & EF31 = -7) then v700 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v700 = . & EF31 = 2) then v700 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v700 ~= EF700));
  table v700 * EF700;
run;

* EF701: Kinder (ab 27 Jahre): Anzahl im Haushalt, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF287 /* Gegenwärtiger Schulbesuch */
    EF701 /* Kinder (ab 27 Jahre): Anzahl im Haushalt, Schüler (Haushaltstypisierung)
  );
  x701 = (EF31=1 & EF35=3 & EF44>=27 & (EF287= 1 ! EF287=2)) /* Privathaushalt | Kind
  in Lebensform | ab 27 Jahre alt | Schüler */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Kinder in Privathaushalten mit der Eigenschaft EF44>=27, Schüler, die zur Bevölkerung in Privathaus-
  halten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF44 EF287 x701 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x701;
  output out = sum(drop=_type_ _freq_ )
  sum = v701;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v701 = . & EF31 = -8) then v701 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v701 = . & EF31 = -7) then v701 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */

```



```

    if (v701 = . & EF31 = 2) then v701 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v701 ^= EF701));
    table v701 * EF701;
run;

* EF702: Personen (ab 65 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF44 /* Alter */
        EF702 /* Personen (ab 65 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
        x702 = (EF31=1 & EF44>=65) /* Privathaushalt | ab 65 Jahren */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Privathaushalten mit der Eigenschaft EF44>=65, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF44 x702 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x702;
    output out = sum(drop=_type_ _freq_ )
        sum = v702;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v702 >=9 & v702 <=99) then v702 = 9;
    if (v702 = . & EF31 = -8) then v702 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v702 = . & EF31 = -7) then v702 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v702 = . & EF31 = 2) then v702 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v702 ^= EF702));
    table v702 * EF702;
run;

* EF703: Personen (ab 70 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF44 /* Alter */
        EF703 /* Personen (ab 70 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
        x703 = (EF31=1 & EF44>=70) /* Privathaushalt | ab 70 Jahren */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Privathaushalten mit der Eigenschaft EF44>=70, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF44 x703 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x703;
    output out = sum(drop=_type_ _freq_ )
        sum = v703;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
    set tmp14;

```

```

if (v703 >=9 & v703 <=99) then v703 = 9;
if (v703 = . & EF31 = -8) then v703 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v703 = . & EF31 = -7) then v703 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v703 = . & EF31 = 2) then v703 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v703 ~= EF703));
table v703 * EF703;
run;

* EF704: Personen (ab 75 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF44 /* Alter */
EF704) /* Personen (ab 75 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x704 = (EF31=1 & EF44>=75) /* Privathaushalt | ab 75 Jahren */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Privathaushalten mit der Eigenschaft EF44>=75, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF44 x704 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x704;
output out = sum(drop=_type_ _freq_ )
sum = v704;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v704 >=9 & v704 <=99) then v704 = 9;
if (v704 = . & EF31 = -8) then v704 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v704 = . & EF31 = -7) then v704 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v704 = . & EF31 = 2) then v704 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v704 ~= EF704));
table v704 * EF704;
run;

* EF705: Personen (ab 80 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF44 /* Alter */
EF705) /* Personen (ab 80 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x705 = (EF31=1 & EF44>=80) /* Privathaushalt | ab 80 Jahren */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Privathaushalten mit der Eigenschaft EF44>=80, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF44 x705 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x705;
output out = sum(drop=_type_ _freq_ )
sum = v705;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

```

```

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v705 >=9 & v705 <=99) then v705 = 9;
if (v705 = . & EF31 = -8) then v705 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v705 = . & EF31 = -7) then v705 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v705 = . & EF31 = 2) then v705 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v705 ~= EF705));
table v705 * EF705;
run;

* EF706: Personen (ab 85 Jahren): Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF44 /* Alter */
EF706 /* Personen (ab 85 Jahren): Anzahl im Haushalt (Haushaltstypisierung) */;
x706 = (EF31=1 & EF44>=85) /* Privathaushalt | ab 85 Jahren */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Privathaushalten mit der Eigenschaft EF44>=85, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF44 x706 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x706;
output out = sum(drop=_type_ _freq_ )
sum = v706;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v706 >=9 & v706 <=99) then v706 = 9;
if (v706 = . & EF31 = -8) then v706 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v706 = . & EF31 = -7) then v706 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v706 = . & EF31 = 2) then v706 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v706 ~= EF706));
table v706 * EF706;
run;

* EF709: Personen am Nebenwohnsitz: Anzahl im Haushalt
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF30 /* Bevölkerung: Haupt- oder Nebenwohnsitz */
EF31 /* Bevölkerung: Privathaushalte */
EF709 /* Personen am Nebenwohnsitz: Anzahl im Haushalt (Haushaltstypisierung) */;
x709 = (EF30=3 & EF31=1) /* Nebenwohnsitz | Privathaushalt */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen am Nebenwohnsitz, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF30 EF31 x709 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x709;
output out = sum(drop=_type_ _freq_ )
sum = v709;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;

```

```

    by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp4;
set tmp4;
if (v709 = . & EF31 = -8) then v709 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v709 = . & EF31 = -7) then v709 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v709 = . & EF31 = 2) then v709 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp4 (where= (v709 ~= EF709));
table v709 * EF709;
run;

* EF711: Familien: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp4 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung in der Lebensform */
EF809 /* Typ der Lebensform (Konzept der Lebensformen) */
EF711) /* Familien: Anzahl im Haushalt (Konzept der Lebensformen) */;
x711 = (EF31=1 & EF35=1 & (EF809>=1 & EF809<=4)) /* Privathaushalt | Bezugsperson
der Lebensform | Lebensform mit Kindern */;
run;

* Sortieren nach IDs ;
proc sort data = tmp4;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Familien in Privathaushalten in einer Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehören;
proc means data = tmp4 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF809 x711 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x711;
output out = sum(drop=_type_ _freq_ )
sum = v711;
run;

* Zusammenspielen;
data tmp4 (compress = yes);
merge tmp4 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp4;
set tmp4;
if (v711 = . & EF31 = -8) then v711 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v711 = . & EF31 = -7) then v711 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v711 = . & EF31 = 2) then v711 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle; * 2 Abweichungen;
proc freq data = tmp4 (where= (v711 ~= EF711));
table v711 * EF711;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF712: Ehepaare: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp4 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung in der Lebensform */
EF809 /* Typ der Lebensform (Konzept der Lebensformen) */
EF712) /* Ehepaare: Anzahl im Haushalt (Konzept der Lebensformen) */;
x712 = ((EF31=1 & EF35=1) & (EF809=1 ! EF809=5)) /* Privathaushalt | Bezugsperson
der Lebensform | Lebensform mit Ehepaar */;
run;

* Sortieren nach IDs ;
proc sort data = tmp4;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Ehepaare in Privathaushalten in einer Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehören;
proc means data = tmp4 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF809 x712 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x712;

```

```

output out = sum(drop=_type_ _freq_ )
sum = v712;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v712 = . & EF31 = -8) then v712 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v712 = . & EF31 = -7) then v712 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v712 = . & EF31 = 2) then v712 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v712 ~= EF712));
table v712 * EF712;
run;

* EF713: Nichteheleliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung in der Lebensform */
EF809 /* Typ der Lebensform (Konzept der Lebensformen) */
EF713) /* Nichteheleliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebens-
formen) */;
x713 = ((EF31=1 & EF35=1) & (EF809=2 ! EF809=6)) /* Privathaushalt | Bezugsperson
der Lebensform | Nichteheleliche Lebensgemeinschaft */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Nichtehelelichen Lebensgemeinschaften in Privathaushalten in einer Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF809 x713 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x713;
output out = sum(drop=_type_ _freq_ )
sum = v713;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v713 = . & EF31 = -8) then v713 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v713 = . & EF31 = -7) then v713 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v713 = . & EF31 = 2) then v713 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v713 ~= EF713));
table v713 * EF713;
run;

* EF714: Gleichgeschlechtliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung in der Lebensform */
EF809 /* Typ der Lebensform (Konzept der Lebensformen) */
EF714) /* Gleichgeschlechtliche Lebensgemeinschaft: Anzahl im Haushalt (Konzept der
Lebensformen) */;
x714 = ((EF31=1 & EF35=1) & (EF809=3 ! EF809=7)) /* Privathaushalt | Bezugsperson
der Lebensform | Gleichgeschlechtliche Lebensgemeinschaft */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;

```

```

by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Personen in Gleichgeschlechtlichen Lebensgemeinschaften in einer Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF809 x714 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x714;
output out = sum(drop=_type_ _freq_ )
sum = v714;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v714 = . & EF31 = -8) then v714 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v714 = . & EF31 = -7) then v714 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v714 = . & EF31 = 2) then v714 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v714 ~= EF714));
table v714 * EF714;
run;

* EF715: Alleinerziehende: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung in der Lebensform */
EF809 /* Typ der Lebensform (Konzept der Lebensformen) */
EF715 /* Alleinerziehende: Anzahl im Haushalt (Konzept der Lebensformen) */;
x715 = (EF31=1 & EF35=1 & EF809=4) /* Privathaushalt | Bezugsperson der Lebensform |

Alleinerziehende */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der alleinerziehenden Personen in einer Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF809 x715 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x715;
output out = sum(drop=_type_ _freq_ )
sum = v715;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v715 = . & EF31 = -8) then v715 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v715 = . & EF31 = -7) then v715 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v715 = . & EF31 = 2) then v715 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle; * 2 Abweichungen;
proc freq data = tmp14 (where= (v715 ~= EF715));
table v715 * EF715;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF716: Alleinstehende: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung in der Lebensform */
EF809 /* Typ der Lebensform (Konzept der Lebensformen) */

```

```

EF716) /* Alleinstehende: Anzahl im Haushalt (Konzept der Lebensformen) */;
x716 = (EF31=1 & EF35=1 & EF809=8) /* Privathaushalt | Bezugsperson der Lebensform |
Alleinstehende */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der alleinstehenden Personen in einer Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF809 x716 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x716;
  output out = sum(drop=_type_ _freq_ )
  sum = v716;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v716 >=9 & v716 <=99) then v716 = 9;
  if (v716 = . & EF31 = -8) then v716 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v716 = . & EF31 = -7) then v716 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v716 = . & EF31 = 2) then v716 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle; * 3 Abweichungen;
proc freq data = tmp14 (where= (v716 ~= EF716));
  table v716 * EF716;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF719: Eingetragene Lebenspartnerschaften: Anzahl im Haushalt (Konzept der Lebensformen)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<cmz14.suf2014>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung in der Lebensform */
  EF811 /* Nichteheleiche Lebensgemeinschaft nach Familienstand der Partner (Konzept
der Lebensformen) */
  EF719) /* Eingetragene Lebenspartnerschaften: Anzahl im Haushalt (Konzept der Le-
bensformen) */;
  x719 = (EF31=1 & EF35=1 & EF811=7) /* Privathaushalt | Bezugsperson der Lebensform |
eingetragene Lebenspartnerschaft */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der eingetragenen Lebenspartnerschaften in einer Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF35 EF811 x719 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x719;
  output out = sum(drop=_type_ _freq_ )
  sum = v719;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v719 = . & EF31 = -8) then v719 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v719 = . & EF31 = -7) then v719 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v719 = . & EF31 = 2) then v719 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v719 ~= EF719));
  table v719 * EF719;
run;

* EF727: 1. ausl. Staatsangehörigkeit: Haupteinkommensbezieher

```

```

* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF369 /* 1. ausländische Staatsangehörigkeit */
    EF662 /* Haupteinkommensbezieher im Haushalt */
    EF727 /* 1. ausl. Staatsangehörigkeit: Haupteinkommensbezieher */;
    x727 = (EF369*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der 1. ausländischen Staatsangehörigkeit des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF369 EF662 x727 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x727;
  output out = sum(drop=_type_ _freq_ )
  sum = v727;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v727 = . & EF31 = -8) then v727 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v727 = . & EF31 = -7) then v727 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v727 = . & EF31 = 2) then v727 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v727 ~= EF727));
  table v727 * EF727;
run;

* EF731: Geschlecht: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF46 /* Geschlecht */
    EF662 /* Haupteinkommensbezieher im Haushalt */
    EF731 /* Geschlecht: Haupteinkommensbezieher */;
    x731 = (EF46*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Geschlechts des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF46 EF662 x731 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x731;
  output out = sum(drop=_type_ _freq_ )
  sum = v731;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v731 = . & EF31 = -8) then v731 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v731 = . & EF31 = -7) then v731 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v731 = . & EF31 = 2) then v731 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v731 ~= EF731));

```



```

        table v731 * EF731;
run;

* EF732: Alter: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF44 /* Alter */
        EF662 /* Haupteinkommensbezieher im Haushalt */
        EF732 /* Alter: Haupteinkommensbezieher */;
        x732 = (EF44*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Alters des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF44 EF662 x732 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x732;
    output out = sum(drop=_type_ _freq_ )
        sum = v732;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v732 = . & EF31 = -8) then v732 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v732 = . & EF31 = -7) then v732 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v732 = . & EF31 = 2) then v732 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v732 ~= EF732));
    table v732 * EF732;
run;

* EF734: Staatsangehörigkeit: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF44 /* Alter */
        EF662 /* Haupteinkommensbezieher im Haushalt */
        EF371 /* Staatsangehörigkeit typisiert */
        EF734 /* Staatsangehörigkeit: Haupteinkommensbezieher */;
        x734 = (EF371*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Staatsangehörigkeit des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF371 EF662 x734 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x734;
    output out = sum(drop=_type_ _freq_ )
        sum = v734;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v734 = . & EF31 = -8) then v734 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */

```

```

if (v734 = . & EF31 = -7) then v734 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v734 = . & EF31 = 2) then v734 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v734 ~= EF734));
table v734 * EF734;
run;

* mig_heb: Migrationsstatus: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF2001 /* Migrationsstatus (differenz. Angabe) */
mig_heb) /* Migrationsstatus: Haupteinkommensbezieher */;
xmig_heb = (EF2001*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Hauptein-
kommensbezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Migrationsstatus des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) ge-
hört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF2001 EF662 xmig_heb where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var xmig_heb;
output out = sum(drop=_type_ _freq_ )
sum = vmig_heb;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (vmig_heb = . & EF31 = -8) then vmig_heb = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte
und Gemeinschaftsunterkünfte) */
if (vmig_heb = . & EF31 = -7) then vmig_heb = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (vmig_heb = . & EF31 = 2) then vmig_heb = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (vmig_heb ~= mig_heb));
table vmig_heb * mig_heb;
run;

* migs_heb: Staatsangehörigkeit zum Migrationsstatus: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF2007 /* Staatsangehörigkeit: Migrationsstatus */
migs_heb) /* Saatsangehörigkeit zum Migrationsstatus: Haupteinkommensbezieher */;
xmigs_heb = (EF2007*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Hauptein-
kommensbezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Staatsangehörigkeit zum Migrationsstatus des Haupteinkommensbeziehers, der zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF2007 EF662 xmigs_heb where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var xmigs_heb;
output out = sum(drop=_type_ _freq_ )
sum = vmigs_heb;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

```

```

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (vmigs_heb = . & EF31 = -8) then vmigs_heb = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und Gemeinschaftsunterkünfte) */
if (vmigs_heb = . & EF31 = -7) then vmigs_heb = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (vmigs_heb = . & EF31 = 2) then vmigs_heb = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (vmigs_heb ~= migs_heb));
table vmigs_heb * migs_heb;
run;

* EF735: Familienstand: Haupteinkommensbezieher, erweitert
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF765 /* Familienstand, erweitert (Konzept der Lebensformen) */
EF735 /* Familienstand: Haupteinkommensbezieher, erweitert */;
x735 = (EF765*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Familienstands des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF765 EF662 x735 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x735;
output out = sum(drop=_type_ _freq_ )
sum = v735;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v735 = . & EF31 = -8) then v735 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v735 = . & EF31 = -7) then v735 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v735 = . & EF31 = 2) then v735 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v735 ~= EF735));
table v735 * EF735;
run;

* EF736: Erwerbstyp: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF29 /* Erwerbstyp */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF736 /* Erwerbstyp: Haupteinkommensbezieher */;
x736 = (EF29*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Erwerbstyps des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF29 EF662 x736 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x736;
output out = sum(drop=_type_ _freq_ )
sum = v736;
run;

* Zusammenspielen;

```

```

data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v736 = . & EF31 = -8) then v736 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v736 = . & EF31 = -7) then v736 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v736 = . & EF31 = 2) then v736 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v736 ~= EF736));
  table v736 * EF736;
run;

* EF737: Wirtschaftszweig: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF137 /* Wirtschaftszweig Betrieb (WZ08) */
  EF662 /* Haupteinkommensbezieher im Haushalt */
  EF737) /* Wirtschaftszweig: Haupteinkommensbezieher */;
  x737 = (EF137*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Wirtschaftszweigs des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF137 EF662 x737 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x737;
  output out = sum(drop=_type_ _freq_ )
  sum = v737;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v737 = . & EF31 = -8) then v737 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v737 = . & EF31 = -7) then v737 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v737 = . & EF31 = 2) then v737 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v737 ~= EF737));
  table v737 * EF737;
run;

* EF738: Beruf (KldB 2010): Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF114 /* Beruf (KldB 2010) */
  EF662 /* Haupteinkommensbezieher im Haushalt */
  EF738) /* Beruf (KldB 2010): Haupteinkommensbezieher */;
  x738 = (EF114*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Berufs (KldB 2010) des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF114 EF662 x738 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;

```

```

var x738;
output out = sum(drop=_type_ _freq_ )
sum = v738;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v738 = . & EF31 = -8) then v738 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v738 = . & EF31 = -7) then v738 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v738 = . & EF31 = 2) then v738 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v738 = -5) then v738 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v738 ~= EF738));
table v738 * EF738;
run;

* EF739: Beruf (ISCO-08): Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF541 /* Beruf (ISCO-08) */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF739 /* Beruf (KldB 2010): Haupteinkommensbezieher */;
x739 = (EF541*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Berufs (ISCO-08) des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) ge-
hört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF541 EF662 x739 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x739;
output out = sum(drop=_type_ _freq_ )
sum = v739;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v739 = . & EF31 = -8) then v739 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v739 = . & EF31 = -7) then v739 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v739 = . & EF31 = 2) then v739 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v739 = -5) then v739 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v739 ~= EF739));
table v739 * EF739;
run;

* EF740: Stellung im Beruf: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF117 /* Stellung im Beruf */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF740 /* Stellung im Beruf: Haupteinkommensbezieher */;
x740 = (EF117*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;

```

```

proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Stellung im Beruf des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF117 EF662 x740 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x740;
output out = sum(drop=_type_ _freq_ )
sum = v740;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v740 = . & EF31 = -8) then v740 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v740 = . & EF31 = -7) then v740 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v740 = . & EF31 = 2) then v740 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if v740 = -5 then v740 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v740 ~= EF740));
  table v740 * EF740;
run;

* EF741: Überwiegender Lebensunterhalt: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF401 /* Überwiegender Lebensunterhalt */
EF662 /* Haupteinkommensbezieher im Haushalt */
EF741) /* Überwiegender Lebensunterhalt: Haupteinkommensbezieher */;
  x741 = (EF401*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des überwiegenden Lebensunterhalts des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF401 EF662 x741 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x741;
output out = sum(drop=_type_ _freq_ )
sum = v741;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v741 = . & EF31 = -8) then v741 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v741 = . & EF31 = -7) then v741 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v741 = . & EF31 = 2) then v741 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v741 ~= EF741));
  table v741 * EF741;
run;

* EF742: Nettoeinkommen (im letzten Monat): Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF436 /* Nettoeinkommen (im letzten Monat) */

```

```

EF662 /* Haupteinkommensbezieher im Haushalt */
EF742 /* Nettoeinkommen (im letzten Monat): Haupteinkommensbezieher */;
x742 = (EF436*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Nettoeinkommens des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF436 EF662 x742 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x742;
  output out = sum(drop=_type_ _freq_ )
  sum = v742;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v742 = . & EF31 = -8) then v742 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v742 = . & EF31 = -7) then v742 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v742 = . & EF31 = 2) then v742 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v742 ~= EF742));
  table v742 * EF742;
run;

* EF743: Höchster allgemeiner Schulabschluss: Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF309 /* Allgemeiner Schulabschluss */
  EF310 /* Höchster allgemeiner Schulabschluss */
  EF662 /* Haupteinkommensbezieher im Haushalt */
  EF743) /* Höchster allgemeiner Schulabschluss: Haupteinkommensbezieher */;
run;

* Variable EF310 recodieren;
data tmp14;
  set <<mz14.suf2014>>;
  EF310n = EF310;
  if (EF310 = 6) then EF310n = 8;
  if (EF310 = 9) then EF310n = 7;
  if (EF310 = 7) then EF310n = 9;
  if (EF310 = -5 & EF309 = 9) then EF310n = 6;
  if (EF310 = -5) then EF310n = 8;
run;

* Erstellen der Hilfsvariable x743;
data tmp14 (compress = yes);
  set <<mz14.suf2014>>;
  x743 = (EF310n*(EF31=1 & (EF662>=1 & EF662<=4))) /*Privathaushalt, Haupteinkommensbezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des höchstens allgemeinen Schulabschlusses des Haupteinkommensbeziehers, der zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF310n EF662 x743 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x743;
  output out = sum(drop=_type_ _freq_ )
  sum = v743;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;

```

```

    if (v743 = . & EF31 = -8) then v743 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v743 = . & EF31 = -7) then v743 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v743 = . & EF31 = 2) then v743 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle: OK;
proc freq data = tmp14 (where= (v743 ~= EF743));
    table v743 * EF743;
run;

* EF744: Höchster beruflicher Abschluss: differenzierte Angabe, Haupteinkommensbezieher
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF320 /* Höchster beruflicher Abschluss (einschließlich Abschlussgrad) */
        EF662 /* Haupteinkommensbezieher im Haushalt */
        EF744) /* Höchster beruflicher Abschluss: differenzierte Angabe, Haupteinkommensbe-
zieher */;
    x744 = (EF320*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des höchsten beruflichen Abschlusses des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF320 EF662 x744 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x744;
    output out = sum(drop=_type_ _freq_ )
        sum = v744;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v744 = . & EF31 = -8) then v744 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v744 = . & EF31 = -7) then v744 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v744 = . & EF31 = 2) then v744 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v744 ~= EF744));
    table v744 * EF744;
run;

* EF745: Höchster beruflicher/schulischer Abschluss: Haupteinkommensbezieher (ISCED 2011)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF31 /* Bevölkerung: Privathaushalte */
        EF540 /* Höchster beruflicher oder allgemeiner Abschluss (ISCED 2011) */
        EF662 /* Haupteinkommensbezieher im Haushalt */
        EF745) /* Höchster beruflicher/schulischer Abschluss: Haupteinkommensbezieher (ISCED
2011) */;
    x745 = (EF540*(EF31=1 & (EF662>=1 & EF662<=4))) /* Privathaushalt | Haupteinkommens-
bezieher im Haushalt */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des höchsten beruflichen Abschlusses des Haupteinkommensbeziehers, der zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF540 EF662 x745 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
    var x745;
    output out = sum(drop=_type_ _freq_ )
        sum = v745;
run;

* Zusammenspielen;

```



```

data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v745 = . & EF31 = -8) then v745 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v745 = . & EF31 = -7) then v745 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v745 = . & EF31 = 2) then v745 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v745 ~= EF745));
  table v745 * EF745;
run;

* EF753: Geschlecht: Haushaltsbezugsperson
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF37 /* Stellung der Person im Haushalt */
  EF46 /* Geschlecht */
  EF753) /* Geschlecht: Haushaltsbezugsperson */;
  x753 = (EF46*(EF37=1)) /* Haushaltsbezugsperson */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Geschlechts der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF46 x753 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x753;
  output out = sum(drop=_type_ _freq_ )
  sum = v753;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v753 = . & EF31 = -8) then v753 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v753 = . & EF31 = -7) then v753 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v753 = . & EF31 = 2) then v753 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v753 ~= EF753));
  table v753 * EF753;
run;

* EF754: Alter: Haushaltsbezugsperson
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF31 /* Bevölkerung: Privathaushalte */
  EF37 /* Stellung der Person im Haushalt */
  EF44 /* Alter */
  EF754) /* Alter: Haushaltsbezugsperson */;
  x754 = (EF44*(EF37=1)) /* Haushaltsbezugsperson */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Alters der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF44 x754 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
  var x754;
  output out = sum(drop=_type_ _freq_ )
  sum = v754;
run;

```

```

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v754 = . & EF31 = -8) then v754 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v754 = . & EF31 = -7) then v754 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v754 = . & EF31 = 2) then v754 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v754 ~= EF754));
  table v754 * EF754;
run;

* EF756: Staatsangehörigkeit: Haushaltsbezugsperson
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF37 /* Stellung der Person im Haushalt */
    EF371 /* Staatsangehörigkeit typisiert */
    EF756) /* Staatsangehörigkeit: Haushaltsbezugsperson */;
  x756 = (EF371*(EF37=1)) /* Haushaltsbezugsperson */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der Staatsangehörigkeit der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushalten (EF31=1) ge-
  hört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF371 x756 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x756;
  output out = sum(drop=_type_ _freq_ )
  sum = v756;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v756 = . & EF31 = -8) then v756 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v756 = . & EF31 = -7) then v756 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v756 = . & EF31 = 2) then v756 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v756 ~= EF756));
  table v756 * EF756;
run;

* EF757: Familienstand: Haushaltsbezugsperson, erweitert
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF31 /* Bevölkerung: Privathaushalte */
    EF37 /* Stellung der Person im Haushalt */
    EF765 /* Familienstand, erweitert (Konzept der Lebensformen) */
    EF757) /* Familienstand: Haushaltsbezugsperson, erweitert */;
  x757 = (EF765*(EF37=1)) /* Haushaltsbezugsperson */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Familienstands der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF765 x757 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12;
  var x757;

```

```

output out = sum(drop=_type_ _freq_ )
sum = v757;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v757 = . & EF31 = -8) then v757 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v757 = . & EF31 = -7) then v757 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v757 = . & EF31 = 2) then v757 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v757 ~= EF757));
table v757 * EF757;
run;

* EF758: Erwerbstyp: Haushaltsbezugsperson
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF37 /* Stellung der Person im Haushalt */
EF29 /* Erwerbstyp */
EF758 /* Erwerbstyp: Haushaltsbezugsperson */;
x758 = (EF29*(EF37=1)) /* Haushaltsbezugsperson */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Erwerbstyps der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF29 x758 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x758;
output out = sum(drop=_type_ _freq_ )
sum = v758;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v758 = . & EF31 = -8) then v758 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v758 = . & EF31 = -7) then v758 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v758 = . & EF31 = 2) then v758 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v758 ~= EF758));
table v758 * EF758;
run;

* EF759: Wohnsitz: Haushaltsbezugsperson
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF37 /* Stellung der Person im Haushalt */
EF30 /* Bevölkerung: Haupt- und Nebenwohnsitz */
EF759 /* Wohnsitz: Haushaltsbezugsperson */;
x759 = (EF30*(EF37=1)) /* Haushaltsbezugsperson */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe des Wohnsitzes der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF30 x759 where=(EF31=1)) nway noprint;

```

```

by EF1 EF3 EF4 EF5b EF12;
var x759;
output out = sum(drop=_type_ _freq_ )
sum = v759;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v759 = . & EF31 = -8) then v759 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v759 = . & EF31 = -7) then v759 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v759 = . & EF31 = 2) then v759 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v759 ~= EF759));
table v759 * EF759;
run;

* EF760: 1. ausländische Staatsangehörigkeit: Haushaltsbezugsperson
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF31 /* Bevölkerung: Privathaushalte */
EF37 /* Stellung der Person im Haushalt */
EF369 /* 1. ausländische Staatsangehörigkeit */
EF760) /* 1. ausländische Staatsangehörigkeit: Haushaltsbezugsperson */;
x760 = (EF369*(EF37=1)) /* Haushaltsbezugsperson */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12;
run;

* Summe der 1. ausländischen Staatsangehörigkeit der Haushaltsbezugsperson, die zur Bevölkerung in Privathaushal-
ten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF31 EF37 EF369 x760 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12;
var x760;
output out = sum(drop=_type_ _freq_ )
sum = v760;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v760 = . & EF31 = -8) then v760 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v760 = . & EF31 = -7) then v760 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v760 = . & EF31 = 2) then v760 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v760 ~= EF760));
table v760 * EF760;
run;

* EF770: Kinder: Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF770) /* Kinder: Anzahl in Lebensform (Konzept der Lebensform) */;
x770 = ((EF31=1 & EF35=3) & (0 <= EF44 & 99 >=EF44)) /* Privathaushalt | Kind in
Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;

```

```

    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x770 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x770;
output out = sum(drop=_type_ _freq_ )
sum = v770;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Vergrößerung replizieren und Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v770 >=9 & v770 <= 99) then v770 = 9; /* und mehr Personen */
if (v770 = . & EF31 = -8) then v770 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v770 = . & EF31 = -7) then v770 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v770 = . & EF31 = 2) then v770 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v770 ~= EF770));
table v770 * EF770;
run;

* EF771: Kinder (unter 1 Jahr): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF771) /* Kinder (unter 1 Jahr): Anzahl in Lebensform (Konzept der Lebensform) */ ;
x771 = (EF31=1 & EF35=3 & EF44<1) /* Privathaushalt | Kind unter 1 Jahr */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<1, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x771 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x771;
output out = sum(drop=_type_ _freq_ )
sum = v771;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v771 = . & EF31 = -8) then v771 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v771 = . & EF31 = -7) then v771 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v771 = . & EF31 = 2) then v771 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v771 ~= EF771));
table v771 * EF771;
run;

* EF772: Kinder (unter 2 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */

```

```

EF772) /* Kinder (unter 2 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x772 = (EF31=1 & EF35=3 & EF44<2) /* Privathaushalt | Kind unter 2 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<2, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x772 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x772;
  output out = sum(drop=_type_ _freq_ )
  sum = v772;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v772 = . & EF31 = -8) then v772 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v772 = . & EF31 = -7) then v772 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v772 = . & EF31 = 2) then v772 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v772 ~= EF772));
  table v772 * EF772;
run;

* EF773: Kinder (unter 3 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF773) /* Kinder (unter 3 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x773 = (EF31=1 & EF35=3 & EF44<3) /* Privathaushalt | Kind unter 3 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<3, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x773 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x773;
  output out = sum(drop=_type_ _freq_ )
  sum = v773;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v773 = . & EF31 = -8) then v773 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v773 = . & EF31 = -7) then v773 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v773 = . & EF31 = 2) then v773 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v773 ~= EF773));
  table v773 * EF773;
run;

* EF774: Kinder (unter 4 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */

```

```

EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF774 /* Kinder (unter 4 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x774 = (EF31=1 & EF35=3 & EF44<4) /* Privathaushalt | Kind unter 4 Jahren */ ;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<4, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x774 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x774;
  output out = sum(drop=_type_ _freq_ )
  sum = v774;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v774 = . & EF31 = -8) then v774 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v774 = . & EF31 = -7) then v774 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v774 = . & EF31 = 2) then v774 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v774 ~= EF774));
  table v774 * EF774;
run;

* EF775: Kinder (unter 5 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF775) /* Kinder (unter 5 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x775 = (EF31=1 & EF35=3 & EF44<5) /* Privathaushalt | Kind unter 5 Jahren */ ;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<5, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x775 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x775;
  output out = sum(drop=_type_ _freq_ )
  sum = v775;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v775 = . & EF31 = -8) then v775 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v775 = . & EF31 = -7) then v775 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v775 = . & EF31 = 2) then v775 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v775 ~= EF775));

```

```

table v775 * EF775;
run;

* EF776: Kinder (unter 6 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF776) /* Kinder (unter 6 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x776 = (EF31=1 & EF35=3 & EF44<6) /* Privathaushalt | Kind unter 6 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

** Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<6, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x776 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x776;
output out = sum(drop=_type_ _freq_ )
sum = v776;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v776 = . & EF31 = -8) then v776 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v776 = . & EF31 = -7) then v776 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v776 = . & EF31 = 2) then v776 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v776 ~= EF776));
table v776 * EF776;
run;

* EF777: Kinder (unter 7 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF777) /* Kinder (unter 7 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x777 = (EF31=1 & EF35=3 & EF44<7) /* Privathaushalt | Kind unter 7 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<7, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x777 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x777;
output out = sum(drop=_type_ _freq_ )
sum = v777;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;

```



```

    if (v777 = . & EF31 = -8) then v777 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
    Gemeinschaftsunterkünfte) */
    if (v777 = . & EF31 = -7) then v777 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v777 = . & EF31 = 2) then v777 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v777 ~= EF777));
    table v777 * EF777;
run;

* EF778: Kinder (unter 8 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF778) /* Kinder (unter 8 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
    x778 = (EF31=1 & EF35=3 & EF44<8) /* Privathaushalt | Kind unter 8 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* * Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<8, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x778 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
    var x778;
    output out = sum(drop=_type_ _freq_ )
    sum = v778;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v778 = . & EF31 = -8) then v778 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
    Gemeinschaftsunterkünfte) */
    if (v778 = . & EF31 = -7) then v778 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v778 = . & EF31 = 2) then v778 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v778 ~= EF778));
    table v778 * EF778;
run;

* EF779: Kinder (unter 9 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF779) /* Kinder (unter 9 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
    x779 = (EF31=1 & EF35=3 & EF44<9) /* Privathaushalt | Kind unter 9 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<9, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x779 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
    var x779;
    output out = sum(drop=_type_ _freq_ )
    sum = v779;
run;

* Zusammenspielen;

```

```

data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v779 = . & EF31 = -8) then v779 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v779 = . & EF31 = -7) then v779 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v779 = . & EF31 = 2) then v779 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v779 ~= EF779));
  table v779 * EF779;
run;

* EF780: Kinder (unter 10 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung innerhalb der Lebensform */
  EF44 /* Alter */
  EF780) /* Kinder (unter 10 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
  x780 = (EF31=1 & EF35=3 & EF44<10) /* Privathaushalt | Kind unter 10 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<10, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x780 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x780;
  output out = sum(drop=_type_ _freq_ )
  sum = v780;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v780 = . & EF31 = -8) then v780 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v780 = . & EF31 = -7) then v780 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v780 = . & EF31 = 2) then v780 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v780 ~= EF780));
  table v780 * EF780;
run;

* EF781: Kinder (unter 11 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung innerhalb der Lebensform */
  EF44 /* Alter */
  EF781) /* Kinder (unter 11 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
  x781 = (EF31=1 & EF35=3 & EF44<11) /* Privathaushalt | Kind unter 11 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<11, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;

```

```

proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x781 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x781;
output out = sum(drop=_type_ _freq_ )
sum = v781;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v781 = . & EF31 = -8) then v781 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v781 = . & EF31 = -7) then v781 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v781 = . & EF31 = 2) then v781 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v781 ~= EF781));
table v781 * EF781;
run;

* EF782: Kinder (unter 12 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF782) /* Kinder (unter 12 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x782 = (EF31=1 & EF35=3 & EF44<12) /* Privathaushalt | Kind unter 12 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<12, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x782 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x782;
output out = sum(drop=_type_ _freq_ )
sum = v782;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v782 = . & EF31 = -8) then v782 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v782 = . & EF31 = -7) then v782 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v782 = . & EF31 = 2) then v782 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v782 ~= EF782));
table v782 * EF782;
run;

* EF783: Kinder (unter 13 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF783) /* Kinder (unter 13 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x783 = (EF31=1 & EF35=3 & EF44<13) /* Privathaushalt | Kind unter 13 Jahren */ ;
run;

```

```

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<13, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x783 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x783;
  output out = sum(drop=_type_ _freq_ )
  sum = v783;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v783 = . & EF31 = -8) then v783 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v783 = . & EF31 = -7) then v783 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v783 = . & EF31 = 2) then v783 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v783 ~= EF783));
  table v783 * EF783;
run;

* EF784: Kinder (unter 14 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<sz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF784) /* Kinder (unter 14 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
  ;
  x784 = (EF31=1 & EF35=3 & EF44<14) /* Privathaushalt | Kind unter 14 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<14, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x784 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x784;
  output out = sum(drop=_type_ _freq_ )
  sum = v784;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v784 = . & EF31 = -8) then v784 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v784 = . & EF31 = -7) then v784 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v784 = . & EF31 = 2) then v784 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v784 ~= EF784));
  table v784 * EF784;
run;

* EF785: Kinder (unter 15 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<sz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */

```

```

EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF785 /* Kinder (unter 15 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
x785 = (EF31=1 & EF35=3 & EF44<15) /* Privathaushalt | Kind unter 15 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<15, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x785 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x785;
output out = sum(drop=_type_ _freq_ )
sum = v785;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v785 = . & EF31 = -8) then v785 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v785 = . & EF31 = -7) then v785 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v785 = . & EF31 = 2) then v785 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v785 ~= EF785));
  table v785 * EF785;
run;

* EF786: Kinder (unter 16 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF786 /* Kinder (unter 16 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
);
x786 = (EF31=1 & EF35=3 & EF44<16) /* Privathaushalt | Kind unter 16 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<16, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x786 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x786;
output out = sum(drop=_type_ _freq_ )
sum = v786;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v786 = . & EF31 = -8) then v786 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v786 = . & EF31 = -7) then v786 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v786 = . & EF31 = 2) then v786 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v786 ~= EF786));
  table v786 * EF786;
run;

```

```

* EF787: Kinder (unter 17 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF787) /* Kinder (unter 17 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
  x787 = (EF31=1 & EF35=3 & EF44<17) /* Privathaushalt | Kind unter 17 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<17, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x787 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x787;
  output out = sum(drop=_type_ _freq_ )
  sum = v787;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v787 = . & EF31 = -8) then v787 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v787 = . & EF31 = -7) then v787 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v787 = . & EF31 = 2) then v787 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v787 ~= EF787));
  table v787 * EF787;
run;

* EF788: Kinder (unter 18 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF788) /* Kinder (unter 18 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
;
  x788 = (EF31=1 & EF35=3 & EF44<18) /* Privathaushalt | Kind unter 18 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<18, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x788 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x788;
  output out = sum(drop=_type_ _freq_ )
  sum = v788;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v788 >= 9 & v788 <= 99) then v788 = 9; /* 9 und mehr Personen */
  if (v788 = . & EF31 = -8) then v788 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v788 = . & EF31 = -7) then v788 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */

```

```

    if (v788 = . & EF31 = 2) then v788 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v788 ~= EF788));
    table v788 * EF788;
run;

* EF789: Kinder (unter 27 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF789) /* Kinder (unter 27 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
    ;
    x789 = (EF31=1 & EF35=3 & EF44<27) /* Privathaushalt | Kind unter 27 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<27, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x789 where=(EF31=1)) nway noprint;
    by EF1 EF3 EF4 EF5b EF12 EF27;
    var x789;
    output out = sum(drop=_type_ _freq_ )
        sum = v789;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v789 >= 9 & v789 <= 99) then v789 = 9; /* 9 und mehr Personen */
    if (v789 = . & EF31 = -8) then v789 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v789 = . & EF31 = -7) then v789 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v789 = . & EF31 = 2) then v789 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v789 ~= EF789));
    table v789 * EF789;
run;

* EF790: Kinder (ab 18 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF790) /* Kinder (ab 18 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */
    ;
    x790 = (EF31=1 & EF35=3 & EF44>=18) /* Privathaushalt | Kind ab 18 Jahren */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44>=18, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x790 where=(EF31=1)) nway noprint;
    by EF1 EF3 EF4 EF5b EF12 EF27;
    var x790;
    output out = sum(drop=_type_ _freq_ )
        sum = v790;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12 EF27;

```

```

run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v790 = . & EF31 = -8) then v790 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v790 = . & EF31 = -7) then v790 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v790 = . & EF31 = 2) then v790 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v790 ~= EF790));
table v790 * EF790;
run;

* EF791: Kinder (ab 27 Jahren): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF791) /* Kinder (ab 27 Jahren): Anzahl in Lebensform (Konzept der Lebensform) */ ;
x791 = (EF31=1 & EF35=3 & EF44>=27) /* Privathaushalt | Kind ab 27 Jahren */ ;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44>=27, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x791 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x791;
output out = sum(drop=_type_ _freq_ )
sum = v791;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v791 = . & EF31 = -8) then v791 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v791 = . & EF31 = -7) then v791 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v791 = . & EF31 = 2) then v791 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v791 ~= EF791));
table v791 * EF791;
run;

* EF792: Kinder (3-5 Jahre): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF792) /* Kinder (3-5 Jahre): Anzahl in Lebensform (Konzept der Lebensform) */ ;
x792 = (EF31=1 & EF35=3 & 3<=EF44<=5) /* Privathaushalt | Kind 3-5 Jahre */ ;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 3<=EF44<=5, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x792 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x792;
output out = sum(drop=_type_ _freq_ )
sum = v792;

```



```

run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v792 = . & EF31 = -8) then v792 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v792 = . & EF31 = -7) then v792 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v792 = . & EF31 = 2) then v792 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v792 ~= EF792));
  table v792 * EF792;
run;

* EF793: Kinder (6-9 Jahre): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
                        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
                        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
                        EF5b /* Jahresüberhang aus dem Vorjahr */
                        EF12 /* Berichtsquartal */
                        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
                        EF31 /* Bevölkerung: Privathaushalte */
                        EF35 /* Stellung innerhalb der Lebensform */
                        EF44 /* Alter */
                        EF793 /* Kinder (6-9 Jahre): Anzahl in Lebensform (Konzept der Lebensform) */ ;
  x793 = (EF31=1 & EF35=3 & 6<=EF44<=9) /* Privathaushalt | Kind 6-9 Jahre */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 6<=EF44<=9, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x793 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x793;
  output out = sum(drop=_type_ _freq_ )
  sum = v793;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v793 = . & EF31 = -8) then v793 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v793 = . & EF31 = -7) then v793 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v793 = . & EF31 = 2) then v793 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v793 ~= EF793));
  table v793 * EF793;
run;

* EF794: Kinder (10-14 Jahre): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<mz14.suf2014>> (keep = EF1 /* Bundesland */
                        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
                        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
                        EF5b /* Jahresüberhang aus dem Vorjahr */
                        EF12 /* Berichtsquartal */
                        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
                        EF31 /* Bevölkerung: Privathaushalte */
                        EF35 /* Stellung innerhalb der Lebensform */
                        EF44 /* Alter */
                        EF794 /* Kinder (10-14 Jahre): Anzahl in Lebensform (Konzept der Lebensform) */ ;
  x794 = (EF31=1 & EF35=3 & 10<=EF44<=14) /* Privathaushalt | Kind 10-14 Jahre */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

```

```

* Summe der Kinder in einer Lebensform mit der Eigenschaft 10<=EF44<=14, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x794 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x794;
output out = sum(drop=_type_ _freq_ )
sum = v794;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v794 = . & EF31 = -8) then v794 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v794 = . & EF31 = -7) then v794 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v794 = . & EF31 = 2) then v794 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v794 ~= EF794));
table v794 * EF794;
run;

* EF795: Kinder (15-17 Jahre): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF795) /* Kinder (15-17 Jahre): Anzahl in Lebensform (Konzept der Lebensform) */ ;
x795 = (EF31=1 & EF35=3 & 15<=EF44<=17) /* Privathaushalt | Kind 15-17 Jahre */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 15<=EF44<=17, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x795 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x795;
output out = sum(drop=_type_ _freq_ )
sum = v795;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v795 = . & EF31 = -8) then v795 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v795 = . & EF31 = -7) then v795 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v795 = . & EF31 = 2) then v795 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v795 ~= EF795));
table v795 * EF795;
run;

* EF796: Kinder (18-26 Jahre): Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF796) /* Kinder (18-26 Jahre): Anzahl in Lebensform (Konzept der Lebensform) */ ;
x796 = (EF31=1 & EF35=3 & 18<=EF44<=26) /* Privathaushalt | Kind 18-26 Jahre */ ;
run;

```

```

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 18<=EF44<=26, die zur Bevölkerung in Privathaushalten
(EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x796 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x796;
  output out = sum(drop=_type_ _freq_ )
  sum = v796;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v796 = . & EF31 = -8) then v796 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v796 = . & EF31 = -7) then v796 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v796 = . & EF31 = 2) then v796 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v796 ~= EF796));
  table v796 * EF796;
run;

* EF797: Kinder (unter 6 Jahren): Anzahl in Lebensform, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF287 /* Gegenwärtiger Schulbesuch */
    EF797 /* Kinder (unter 6 Jahren): Anzahl in Lebensform, Schüler (Konzept der Le-
bensform) */ ;
  x797 = (EF31=1 & EF35=3 & EF44<6 & (EF287=1 ! EF287=2)) /* Privathaushalt | Kind
unter 6 Jahren | Schüler */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44<6, Schüler, die zur Bevölkerung in Privathaushal-
ten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF287 x797 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x797;
  output out = sum(drop=_type_ _freq_ )
  sum = v797;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v797 = . & EF31 = -8) then v797 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v797 = . & EF31 = -7) then v797 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v797 = . & EF31 = 2) then v797 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v797 ~= EF797));
  table v797 * EF797;
run;

* EF798: Kinder (6-9 Jahre): Anzahl in Lebensform, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */

```

```

EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF287 /* Gegenwärtiger Schulbesuch */
EF798) /* Kinder (6-9 Jahre): Anzahl in Lebensform, Schüler (Konzept der Lebensform)
*/ ;
x798 = (EF31=1 & EF35=3 & 6<=EF44<=9 & (EF287=1 ! EF287=2)) /* Privathaushalt | Kind
6-9 Jahre | Schüler */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 6<=EF44<=9, Schüler, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF287 x798 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x798;
  output out = sum(drop=_type_ _freq_ )
  sum = v798;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v798 = . & EF31 = -8) then v798 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v798 = . & EF31 = -7) then v798 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v798 = . & EF31 = 2) then v798 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v798 ~= EF798));
  table v798 * EF798;
run;

* EF799: Kinder (10-14 Jahre): Anzahl in Lebensform, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung innerhalb der Lebensform */
  EF44 /* Alter */
  EF287 /* Gegenwärtiger Schulbesuch */
  EF799) /* Kinder (10-14 Jahre): Anzahl in Lebensform, Schüler (Konzept der Lebens-
form) */ ;
  x799 = (EF31=1 & EF35=3 & 10<=EF44<=14 & (EF287=1 ! EF287=2)) /* Privathaushalt |
Kind 10-14 Jahre | Schüler */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 10<=EF44<=14, Schüler, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF287 x799 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x799;
  output out = sum(drop=_type_ _freq_ )
  sum = v799;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v799 = . & EF31 = -8) then v799 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v799 = . & EF31 = -7) then v799 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v799 = . & EF31 = 2) then v799 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

```

```

* Kontrolle;
proc freq data = tmp14 (where= (v799 ~= EF799));
    table v799 * EF799;
run;

* EF800: Kinder (15-17 Jahre): Anzahl in Lebensform, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF287 /* Gegenwärtiger Schulbesuch */
        EF800) /* Kinder (15-17 Jahre): Anzahl in Lebensform, Schüler (Konzept der Lebens-
form) */ ;
    x800 = (EF31=1 & EF35=3 & 15<=EF44<=17 & (EF287=1 ! EF287=2)) /* Privathaushalt |
Kind 15-17 Jahre | Schüler */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 15<=EF44<=17, Schüler, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF287 x800 where=(EF31=1)) nway noprint;
    by EF1 EF3 EF4 EF5b EF12 EF27;
    var x800;
    output out = sum(drop=_type_ _freq_ )
    sum = v800;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v800 = . & EF31 = -8) then v800 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v800 = . & EF31 = -7) then v800 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v800 = . & EF31 = 2) then v800 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v800 ~= EF800));
    table v800 * EF800;
run;

* EF801: Kinder (18-26 Jahre): Anzahl in Lebensform, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF287 /* Gegenwärtiger Schulbesuch */
        EF801) /* Kinder (18-26 Jahre): Anzahl in Lebensform, Schüler (Konzept der Lebens-
form) */ ;
    x801 = (EF31=1 & EF35=3 & 18<=EF44<=26 & (EF287=1 ! EF287=2)) /* Privathaushalt |
Kind 18-26 Jahre | Schüler */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft 18<=EF44<=26, Schüler, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF287 x801 where=(EF31=1)) nway noprint;
    by EF1 EF3 EF4 EF5b EF12 EF27;
    var x801;
    output out = sum(drop=_type_ _freq_ )
    sum = v801;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;

```

```

    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v801 = . & EF31 = -8) then v801 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v801 = . & EF31 = -7) then v801 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v801 = . & EF31 = 2) then v801 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v801 ~= EF801));
table v801 * EF801;
run;

* EF802: Kinder (ab 27 Jahren): Anzahl in Lebensform, Schüler
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF287 /* Gegenwärtiger Schulbesuch */
EF802) /* Kinder (ab 27 Jahren): Anzahl in Lebensform, Schüler (Konzept der Lebens-
form) */ ;
x802 = (EF31=1 & EF35=3 & EF44>=27 & (EF287=1 ! EF287=2)) /* Privathaushalt | Kind
ab 27 Jahren | Schüler */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Kinder in einer Lebensform mit der Eigenschaft EF44>=27, Schüler, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF287 x802 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x802;
output out = sum(drop=_type_ _freq_ )
sum = v802;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v802 = . & EF31 = -8) then v802 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v802 = . & EF31 = -7) then v802 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v802 = . & EF31 = 2) then v802 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v802 ~= EF802));
table v802 * EF802;
run;

* EF803: Alter: Jüngstes Kind in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF805 /* Geschwisterfolge: Stellung der ledigen Kinder */
EF803) /* Alter: Jüngstes Kind der Lebensform (Konzept der Lebensform) */ ;
x803 = (EF44*(EF35=3 & EF805=1)) /* Kind in Lebensform | jüngstes Kind in Lebensform
*/ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

```

```

* Summe des Kindes in einer Lebensform mit der Eigenschaft "ältestes Kind in LF", das zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF805 x803 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x803;
output out = sum(drop=_type_ _freq_ )
sum = v803;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v803 = . & EF31 = -8) then v803 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v803 = . & EF31 = -7) then v803 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v803 = 0 & EF770=0) then v803 = -5; /* Entfällt (Kein Kind in Lebensform) */
if (v803 = . & EF31 = 2) then v803 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v803 ~= EF803));
table v803 * EF803;
run;

* EF804: Alter: Ältestes Kind in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF804 /* Alter: Jüngstes Kind der Lebensform (Konzept der Lebensform) */ ;
x804 = EF44* /* */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Kindes in einer Lebensform mit der Eigenschaft "Jüngstes Kind in LF", das zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF804 x804 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x804;
output out = sum(drop=_type_ _freq_ )
sum = v804;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v804 = . & EF31 = -8) then v804 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v804 = . & EF31 = -7) then v804 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v804 = 0 & EF770=0) then v804 = -5; /* Entfällt (Kein Kind in Lebensform) */
if (v804 = . & EF31 = 2) then v804 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v804 ~= EF804));
table v804 * EF804;
run;

* EF812: Personen am Nebenwohnsitz: Anzahl in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF30 /* Haupt- oder Nebenwohnsitz */
EF812 /* Personen am Nebenwohnsitz: Anzahl in Lebensform (Konzept der Lebensform)
*/ ;

```

```

x812 = (EF31=1 & EF30=3) /* Privathaushalt | wohnhaft am Nebenwohnsitz */ ;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Personen am Nebenwohnsitz, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF30 x812 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x812;
  output out = sum(drop=_type_ _freq_ )
  sum = v812;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v812 = . & EF31 = -8) then v812 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v812 = . & EF31 = -7) then v812 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v812 = . & EF31 = 2) then v812 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v812 ~= EF812));
  table v812 * EF812;
run;

* EF819: Geschlecht: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung innerhalb der Lebensform */
  EF44 /* Alter */
  EF46 /* Geschlecht */
  EF819) /* Geschlecht: Bezugsperson der Lebensform (Konzept der Lebensform) */ ;
  x819 = (EF46*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Geschlechts der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF46 x819 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x819;
  output out = sum(drop=_type_ _freq_ )
  sum = v819;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v819 = . & EF31 = -8) then v819 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v819 = . & EF31 = -7) then v819 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v819 = . & EF31 = 2) then v819 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v819 ~= EF819));
  table v819 * EF819;
run;

* EF820: Alter: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */

```



```

EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF820 /* Alter: Bezugsperson der Lebensform (Konzept der Lebensform) */ ;
x820 = (EF44*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Alters der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x820 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x820;
  output out = sum(drop= type freq )
  sum = v820;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v820 = . & EF31 = -8) then v820 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v820 = . & EF31 = -7) then v820 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v820 = . & EF31 = 2) then v820 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v820 ~= EF820));
  table v820 * EF820;
run;

* EF822: Staatsangehörigkeit: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<=mz14.suf2014>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
  EF31 /* Bevölkerung: Privathaushalte */
  EF35 /* Stellung innerhalb der Lebensform */
  EF44 /* Alter */
  EF371 /* Staatsangehörigkeit typisiert */
  EF822) /* Staatsangehörigkeit: Bezugsperson der Lebensform (Konzept der Lebensform)
  */ ;
  x822 = (EF371*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Staatsangehörigkeit der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
  gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF371 x822 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x822;
  output out = sum(drop=_type_ _freq_ )
  sum = v822;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v822 = . & EF31 = -8) then v822 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v822 = . & EF31 = -7) then v822 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v822 = . & EF31 = 2) then v822 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v822 ~= EF822));
  table v822 * EF822;
run;

* mig_lfl1: Migrationsstatus: Bezugsperson der Lebensform

```

```

* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF2001 /* Migrationsstatus (differenzierte Angabe) */
    mig_lf1 /* Migrationsstatus: Bezugsperson der Lebensform */ ;
    xmig_lf1 = (EF2001*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Migrationsstatus der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF2001 xmig_lf1 where=(EF31=1)) nway
  noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var xmig_lf1;
  output out = sum(drop=_type_ _freq_ )
  sum = vmig_lf1;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (vmig_lf1 = . & EF31 = -8) then vmig_lf1 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushal-
te und Gemeinschaftsunterkünfte) */
  if (vmig_lf1 = . & EF31 = -7) then vmig_lf1 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (vmig_lf1 = . & EF31 = 2) then vmig_lf1 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (vmig_lf1 ~= mig_lf1));
  table vmig_lf1 * mig_lf1;
run;

* migs_lf1: Staatsangehörigkeit zum Migrationsstatus: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF2007 /* Staatsangehörigkeit: Migrationsstatus */
    migs_lf1 /* Staatsangehörigkeit zum Migrationsstatus: Bezugsperson der Lebensform */;
  ;
  xmits_lf1 = (EF2007*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Staatsangehörigkeit zum Migrationsstatus der Bezugsperson der Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF2007 xmits_lf1 where=(EF31=1)) nway
  noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var xmits_lf1;
  output out = sum(drop=_type_ _freq_ )
  sum = vmigs_lf1;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (vmigs_lf1 = . & EF31 = -8) then vmigs_lf1 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushal-
te und Gemeinschaftsunterkünfte) */

```

```

    if (vmigs_lf1 = . & EF31 = -7) then vmigs_lf1 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (vmigs_lf1 = . & EF31 = 2) then vmigs_lf1 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (vmigs_lf1 ~= migs_lf1));
    table vmigs_lf1 * migs_lf1;
run;

* EF823: Familienstand: Bezugsperson der Lebensform, erweitert
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF765 /* Familienstand, erweitert (Konzept der Lebensformen) */
        EF823) /* Familienstand: Bezugsperson der Lebensform, erweitert */;
    x823 = (EF765*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Familienstands der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF765 x823 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
    var x823;
    output out = sum(drop=_type_ _freq_ )
    sum = v823;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
    set tmp14;
    if (v823 = . & EF31 = -8) then v823 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
    if (v823 = . & EF31 = -7) then v823 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
    if (v823 = . & EF31 = 2) then v823 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v823 ~= EF823));
    table v823 * EF823;
run;

* EF824: Erwerbstyp: Bezugsperson der Lebensform, erweitert
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF29 /* Erwerbstyp */
        EF824) /* Erwerbstyp: Bezugsperson der Lebensform, erweitert */;
    x824 = (EF29*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Erwerbstyps der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF29 x824 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
    var x824;
    output out = sum(drop=_type_ _freq_ )
    sum = v824;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
    merge tmp14 sum;
    by EF1 EF3 EF4 EF5b EF12 EF27;

```

```

run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v824 = . & EF31 = -8) then v824 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v824 = . & EF31 = -7) then v824 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v824 = . & EF31 = 2) then v824 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v824 ~= EF824));
table v824 * EF824;
run;

* EF825: Wirtschaftszweig Betrieb: Bezugsperson der Lebensform (WZ08)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF137 /* Wirtschaftszweig Betrieb (WZ08) */
EF825) /* Wirtschaftszweig Betrieb: Bezugsperson der Lebensform (WZ08) */ ;
x825 = (EF137*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Wirtschaftszweigs (WZ08) der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF137 x825 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x825;
output out = sum(drop=_type_ _freq_ )
sum = v825;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v825 = . & EF31 = -8) then v825 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v825 = . & EF31 = -7) then v825 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v825 = . & EF31 = 2) then v825 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v825 ~= EF825));
table v825 * EF825;
run;

* EF826: Beruf: Bezugsperson der Lebensform (KldB 2010)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF114 /* Beruf (KldB 2010) */
EF826) /* Beruf: Bezugsperson der Lebensform (KldB 2010) */ ;
x826 = (EF114*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Berufs der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF114 x826 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x826;
output out = sum(drop=_type_ _freq_ )

```

```

sum = v826;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v826 = . & EF31 = -8) then v826 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v826 = . & EF31 = -7) then v826 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v826 = . & EF31 = 2) then v826 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if v826 = -5 then v826 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v826 ~= EF826));
table v826 * EF826;
run;

* EF827: Beruf: Bezugsperson der Lebensform (ISCO-08)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF541 /* Beruf (KldB 2010) */
EF827) /* Beruf: Bezugsperson der Lebensform (ISCO-08) */ ;
x827 = (EF541*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Berufs (ISCO-08) der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF541 x827 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x827;
output out = sum(drop=_type_ _freq_ )
sum = v827;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v827 = . & EF31 = -8) then v827 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v827 = . & EF31 = -7) then v827 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v827 = . & EF31 = 2) then v827 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if v827 = -5 then v827 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v827 ~= EF827));
table v827 * EF827;
run;

* EF828: Stellung im Beruf: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF117 /* Stellung im Beruf */
EF828) /* Stellung im Beruf: Bezugsperson der Lebensform */ ;
x828 = (EF117*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;

```

```

proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Stellung im Beruf der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF117 x828 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x828;
output out = sum(drop=_type_ _freq_ )
sum = v828;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v828 = . & EF31 = -8) then v828 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v828 = . & EF31 = -7) then v828 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v828 = . & EF31 = 2) then v828 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v828 = -5) then v828 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v828 ~= EF828));
  table v828 * EF828;
run;

* EF829: Überwiegender Lebensunterhalt: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF401 /* Überwiegender Lebensunterhalt */
EF829) /* Überwiegender Lebensunterhalt: Bezugsperson der Lebensform */ ;
x829 = (EF401*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des überwiegenden Lebensunterhalts der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushal-
ten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF401 x829 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x829;
output out = sum(drop=_type_ _freq_ )
sum = v829;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v829 = . & EF31 = -8) then v829 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v829 = . & EF31 = -7) then v829 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v829 = . & EF31 = 2) then v829 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v829 ~= EF829));
  table v829 * EF829;
run;

* EF830: Nettoeinkommen (im letzten Monat): Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */

```

```

EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF436 /* Nettoeinkommen (im letzten Monat) */
EF830 /* Nettoeinkommen (im letzten Monat): Bezugsperson der Lebensform */ ;
x830 = (EF436*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Nettoeinkommens der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF436 x830 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x830;
  output out = sum(drop=_type_ _freq_ )
  sum = v830;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v830 = . & EF31 = -8) then v830 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v830 = . & EF31 = -7) then v830 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v830 = . & EF31 = 2) then v830 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v830 ~= EF830));
  table v830 * EF830;
run;

* EF831: Höchster allgemeiner Schulabschluss: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF310 /* Höchster allgemeiner Schulabschluss */
EF831) /* Höchster allgemeiner Schulabschluss: Bezugsperson der Lebensform */ ;

run;

* Variable EF310 recodieren;
data tmp14;
  set <<<mz14.suf2014>>>;
  EF310n = EF310;
  if EF310 = 6 then EF310n = 8;
  if EF310 = 9 then EF310n = 7;
  if EF310 = 7 then EF310n = 9;
  if EF310 = -5 & EF309=9 then EF310n = 6;
  if EF310 = -5 then EF310n = 8;
  if EF310 = -3 then EF310n = 8;
run;

* Erstellen der Hilfsvariable x831;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>>;
  x831= (EF310n*(EF35=1 & EF44>=10));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des höchsten allgemeinen Schulabschlusses der Bezugsperson der Lebensform, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF310 x831 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x831;
  output out = sum(drop=_type_ _freq_ )
  sum = v831;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

```

```

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v831 = . & EF31 = -8) then v831 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v831 = . & EF31 = -7) then v831 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v831 = . & EF31 = 2) then v831 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v831 ~= EF831));
table v831 * EF831;
run;

* EF832: Höchster beruflicher Abschluss: differenzierte Angabe, Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF320 /* Höchster beruflicher Abschluss (einschließlich Abschlussgrad) */
EF832) /* Höchster beruflicher Schulabschluss: Bezugsperson der Lebensform */ ;
x832 = (EF320*(EF35=1 & EF44>=10));

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des höchsten beruflichen Abschlusses der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaus-
halten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF320 x832 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x832;
output out = sum(drop=_type_ _freq_ )
sum = v832;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v832 = . & EF31 = -8) then v832 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v832 = . & EF31 = -7) then v832 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v832 = . & EF31 = 2) then v832 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v832 = -3) then v832 = 88; /* Kein (beruflicher) Abschluss (einschließlich Schüler an allgemeinbildenden
Schulen) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v832 ~= EF832));
table v832 * EF832;
run;

* EF833: Höchster schulischer/beruflicher Abschluss: Bezugsperson der Lebensform (ISCED 2011)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF540 /* Höchster beruflicher oder allgemeiner Abschluss (ISCED 2011) */
EF833) /* Höchster beruflicher oder allgemeiner Schulabschluss: Bezugsperson der
Lebensform */ ;
x833 = (EF540*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des höchsten schulischen/beruflichen Abschlusses der Bezugsperson der Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF540 x833 where=(EF31=1)) nway noprint;

```



```

by EF1 EF3 EF4 EF5b EF12 EF27;
var x833;
output out = sum(drop=_type_ _freq_ )
sum = v833;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v833 = . & EF31 = -8) then v833 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v833 = . & EF31 = -7) then v833 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v833 = . & EF31 = 2) then v833 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v833 ~= EF833));
table v833 * EF833;
run;

* EF834: Arbeitszeittyp: Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF130 /* Teilzeittätigkeit: Grund */
EF834) /* Arbeitszeittyp: Bezugsperson der Lebensform */;

run;

* Variable EF130 recodieren;
data tmp14;
set <<<mz14.suf2014>>>;
EF310n = EF310;
if EF130 = -5 then EF130n = 8;
if 1<=EF130>=9 & EF129=1 then EF130n= 8;
run;

* Erstellen der Hilfsvariable x834;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>>;
x834= (EF130*(EF35=1 & EF44=>10));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Arbeitszeittyps der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF130 x834 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x834;
output out = sum(drop=_type_ _freq_ )
sum = v834;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v834 = . & EF31 = -8) then v834 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v834 = . & EF31 = -7) then v834 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v834 = . & EF31 = 2) then v834 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle: OK;
proc freq data = tmp14 (where= (v834 ~= EF834));
table v834 * EF834;
run;

* EF835: Normale Arbeitszeit (je Woche: Stunden): Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */

```

```

EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF131 /* Normale Arbeitszeit (je Woche): Stunden */
EF835) /* : Bezugsperson der Lebensform */ ;
x835 = (EF131*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der normalen Arbeitszeit der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF131 x835 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x835;
  output out = sum(drop=_type_ _freq_ )
  sum = v835;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v835 = . & EF31 = -8) then v835 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v835 = . & EF31 = -7) then v835 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v835 = . & EF31 = 2) then v835 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v835 = -1) then v835 = -2; /* Entfällt: Nichterwerbstätiger */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v835 ~= EF835));
  table v835 * EF835;
run;

* EF836: Grund geringerer Arbeitszeit (Berichtswert): Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF147 /* Geringere Arbeitszeit (Berichtswert): Grund */
EF836) /* Grund geringerer Arbeitszeit (Berichtswert): Bezugsperson der Lebensform
*/ ;
run;

* Variable EF147 recodieren;
data tmp14;
  set <<<mz14.suf2014>>>;
  EF147n = EF147;
  if EF147 = 3 & EF29=1 & EF146=2 then EF147n=2;
  if EF147 = 4 & EF29=1 & EF146=2 then EF147n=3;
  if EF147 = 17 & EF29=1 & EF146=2 then EF147n=5;
  if EF147 = 8 & EF29=1 & EF146=2 then EF147n = 6;
  if EF147 = 16 & EF29=1 & EF146=2 then EF147n = 7;
  if (EF147 = 1 ! EF147=2 ! EF147=6 ! EF147=9 ! EF147=10 ! EF147=11 ! EF147=12 ! EF147=13 ! EF147=14 ! EF147=15 !
EF147=18 ! EF147=19 ! EF147=99) & EF29=1 & EF146=2 & EF147~=3 & EF147~=4 & EF147~=8 & EF147~=16 & EF147~=17 then
EF147n = 8;
  if EF147 = -5 & EF29=1 & EF146~=2 then EF147n=1;
run;

* Erstellen der Hilfsvariable x836;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>>;
  x836= (EF147n*(EF35=1 & EF44=>10));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe Grund geringer Arbeitszeit (Berichtswert) der Bezugsperson der Lebensform, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF147n x836 where=(EF31=1)) nway
noprint;

```

```

by EF1 EF3 EF4 EF5b EF12 EF27;
var x836;
output out = sum(drop=_type_ _freq_ )
sum = v836;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v836 = . & EF31 = -8) then v836 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v836 = . & EF31 = -7) then v836 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v836 = . & EF31 = 2) then v836 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v836 ~= EF836));
table v836 * EF836;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF837: Erwerbsunterbrechung (Berichtswert): Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF44 /* Alter */
EF85 /* Geringere Arbeitszeit (Berichtswert): Grund */
EF837) /* Erwerbsunterbrechung (Berichtswert): Bezugsperson der Lebensform */ ;
run;

* Variable EF85 recodieren;
data tmp14;
set <<<mz14.suf2014>>>;
EF85n = EF85;
if EF85=1 & EF85<=11 & EF29=1 then EF85n=1;
if EF85=-5 & EF29=1 then EF85n=2;
run;

* Erstellen der Hilfsvariable x837;
data tmp14 (compress = yes);
set <<<mz14.suf2014>>>;
x837= (EF85n*(EF35=1 & EF44=>10));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Erwerbsunterbrechung der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF85n x837 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x837;
output out = sum(drop=_type_ _freq_ )
sum = v837;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v837 = . & EF31 = -8) then v837 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v837 = . & EF31 = -7) then v837 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v837 = . & EF31 = 2) then v837 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v837 ~= EF837));
table v837 * EF837;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF838: 1. ausländische Staatsangehörigkeit: Bezugsperson der Lebensform
* Datenauswahl einlesen;

```

```

data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF44 /* Alter */
    EF369 /* 1. ausländische Staatsangehörigkeit */
    EF838) /* 1. ausländische Staatsangehörigkeit: Bezugsperson der Lebensform */ ;
  x838 = (EF369*(EF35=1 & EF44>=10)) /* Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der 1.ausländischen Staatsangehörigkeit der Bezugsperson der Lebensform, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 EF369 x838 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x838;
  output out = sum(drop=_type_ _freq_ )
  sum = v838;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v838 = . & EF31 = -8) then v838 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v838 = . & EF31 = -7) then v838 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v838 = . & EF31 = 2) then v838 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v838 ~= EF838));
  table v838 * EF838;
run;

* EF843: Geschlecht: Lebenspartner der Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF46 /* Geschlecht */
    EF843) /* Geschlecht: Lebenspartner der Bezugsperson der Lebensform */ ;
  x843 = (EF46*(EF35=2)) /* Lebenspartner der Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Geschlechts des Lebenspartners der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushal-
ten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF46 x843 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x843;
  output out = sum(drop=_type_ _freq_ )
  sum = v843;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v843 = . & EF31 = -8) then v843 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v843 = . & EF31 = -7) then v843 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v843 = . & EF31 = 2) then v843 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v843 = 0) then v843 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

```

```

* Kontrolle;
proc freq data = tmp14 (where= (v843 ~= EF843));
    table v843 * EF843;
run;

* EF844: Alter: Lebenspartner der Bezugsperson der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF44 /* Alter */
        EF844) /* Alter: Lebenspartner der Bezugsperson der Lebensform */ ;
    x844 = (EF44*(EF35=2)) /* Lebenspartner der Bezugsperson der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Alters des Lebenspartners der Bezugsperson der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF44 x844 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x844;
output out = sum(drop=_type_ _freq_ )
sum = v844;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v844 = . & EF31 = -8) then v844 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v844 = . & EF31 = -7) then v844 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v844 = . & EF31 = 2) then v844 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v844 = 0) then v844 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v844 ~= EF844));
    table v844 * EF844;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF846: Staatsangehörigk.: Lebenspartner in der Lebensform ;
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF371 /* Staatsangehörigkeit typisiert */
        EF761 /* Bevölkerung: Wohnsitz der Lebensform (Konzept der Lebensformen) */
        EF846) /* Staatsangehörigk.: Lebenspartner in der Lebensform */ ;
    if (EF35 = 2) then LP = 1; /* Ehefrau oder Lebenspartner der Bezugsperson in der Lebensform 1 = ja, 0 =
sonst */
    if (EF35 ^= 2) then LP = 0;
    x846 = (EF371*(EF35=2)); /* Staatsangehörigkeit | Lebenspartner in der Lebensform */

run;

proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Staatsangehörigkeit des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF371 x846 LP where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x846 LP;
output out = sum(drop=_type_ _freq_ )
sum = v846 /* v846 := Staatsangeh. | Lebensp. der BP in Leb.form */
n LP; /* n_LP := Anzahl Lebensp. der BP in Leb.form, 0 : kein Leb.partn. ... */
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;

```

```

    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v846 = . & EF31 = -8) then v846 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v846 = . & EF31 = -7) then v846 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v846 = . & EF31 = 2) then v846 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v846 = 0 & n_LP = 0) then v846 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v846 ~= EF846));
table v846 * EF846 /* nocol norow nopercent missing */;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* mig_lf2: Migrationsstatus: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF2001 /* Migrationsstatus (differenzierte Angabe) */
mig_lf2) /* Migrationsstatus: Lebenspartner in der Lebensform */ ;
xmig_lf2 = (EF2001*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Migrationsstatus des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF2001 xmig_lf2 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
var xmig_lf2;
output out = sum(drop=_type_ _freq_ )
sum = vmig_lf2;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (vmig_lf2 = . & EF31 = -8) then vmig_lf2 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushal-
te und Gemeinschaftsunterkünfte) */
if (vmig_lf2 = . & EF31 = -7) then vmig_lf2 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (vmig_lf2 = . & EF31 = 2) then vmig_lf2 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (vmig_lf2 = 0) then vmig_lf2 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (vmig_lf2 ~= mig_lf2));
table vmig_lf2 * mig_lf2;
run;

* migs_lf2: Staatsangehörigkeit zum Migrationsstatus: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF2007 /* Staatsangehörigkeit: Migrationsstatus */
migs_lf2) /* Staatsangehörigkeit Migrationsstatus: Lebenspartner in der Lebensform
*/ ;
xmigs_lf2 = (EF2007*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Staatsangehörigkeit zum Migrationsstatus des Lebenspartners in der Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehört;

```

```

proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF2007 xmigs_lf2 where=(EF31=1)) nway
noprnt;
by EF1 EF3 EF4 EF5b EF12 EF27;
var xmigs_lf2;
output out = sum(drop=_type_ _freq_)
sum = vmigs_lf2;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (vmigs_lf2 = . & EF31 = -8) then vmigs_lf2 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und Gemeinschaftsunterkünfte) */
if (vmigs_lf2 = . & EF31 = -7) then vmigs_lf2 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (vmigs_lf2 = . & EF31 = 2) then vmigs_lf2 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (vmigs_lf2 = 0) then vmigs_lf2 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (vmigs_lf2 ~= migs_lf2));
table vmigs_lf2 * migs_lf2;
run;

* EF847: Familienstand: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF30 /* Haupt- oder Nebenwohnsitz */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF765 /* Familienstand */
EF847 /* Familienstand: Lebenspartner in der Lebensform */;
x847 = (EF765*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Familienstands des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF30 EF31 EF35 EF765 x847 where=(EF31=1)) nway noprnt;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x847;
output out = sum(drop=_type_ _freq_)
sum = v847;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v847 = . & EF31 = -8) then v847 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und Gemeinschaftsunterkünfte) */
if (v847 = . & EF31 = -7) then v847 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v847 = . & EF31 = 2) then v847 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v847 = 0) then v847 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v847 ~= EF847));
table v847 * EF847;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF848: Erwerbstyp: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
set <<<mz14.suf2014>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF29 /* Erwerbstyp */;

```

```

EF848) /* Erwerbstyp: Lebenspartner in der Lebensform */ ;
x848 = (EF29*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Erwerbstyps des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF29 x848 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x848;
  output out = sum(drop=_type_ _freq_ )
  sum = v848;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v848 = . & EF31 = -8) then v848 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v848 = . & EF31 = -7) then v848 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v848 = . & EF31 = 2) then v848 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v848 = 0) then v848 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v848 ~= EF848));
  table v848 * EF848;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF849: Wirtschaftszweig Betrieb: Lebenspartner in der Lebensform ;
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF29 /* Erwerbstyp */
    EF137 /* Wirtschaftszweig Betrieb (WZ08) */
    EF849) /* Wirtschaftszweig Betrieb: Lebenspartner in der Lebensform */ ;
  if (EF35 = 2) then LP = 1; /* Ehefrau oder Lebenspartner der Bezugsperson in der Lebensform 1 = ja, 0 =
sonst */
  if (EF35 ^= 2) then LP = 0;
  x849 = (EF137*(EF35=2 & EF29=1)); /* Wirtschaftszweig Betrieb | Lebenspartner in der Lebensform | Erwerbstä-
tig */
run;

proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Wirtschaftszweigs des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF29 EF137 x849 LP where=(EF31=1)) nway
noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x849 LP;
  output out = sum(drop=_type_ _freq_ )
  sum = v849
  n_LP;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v849 = . & EF31 = -8) then v849 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v849 = . & EF31 = -7) then v849 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v849 = . & EF31 = 2) then v849 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v849 = 0 & n_LP = 0) then v849 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
  if (v849 = 0 & n_LP = 1) then v849 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v849 ~= EF849));

```



```

        table v849 * EF849 / nocol norow nopercnt missing;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF850: Beruf: Lebenspartner in der Lebensform (KldB 2010);
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF29 /* Erwerbstyp */
        EF114 /* Beruf (KldB 2010) */
        EF850) /* Beruf: Lebenspartner in der Lebensform (KldB 2010) */ ;
    if (EF35 = 2) then LP = 1; /* Ehefrau oder Lebenspartner der Bezugsperson in der Lebensform 1 = ja, 0 =
sonst */
    if (EF35 ^= 2) then LP = 0;
    x850 = (EF114*(EF35=2 & EF29=1)); /* Wirtschaftszweig Betrieb | Lebenspartner in der Lebensform | Erwerbstätig */
run;

proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Berufs des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF29 EF114 x850 LP where=(EF31=1)) nway
noprnt;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x850 LP;
output out = sum(drop=_type_ _freq_)
sum = v850
n_LP;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
merge tmp14 sum;
by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
set tmp14;
if (v850 = . & EF31 = -8) then v850 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
if (v850 = . & EF31 = -7) then v850 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
if (v850 = . & EF31 = 2) then v850 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
if (v850 = 0 & n_LP = 0) then v850 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
if (v850 = 0 & n_LP = 1) then v850 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v850 ^= EF850));
table v850 * EF850 / nocol norow nopercnt missing;
run;

* EF851: Beruf: Lebenspartner in der Lebensform (ISCO-08);
* Datenauswahl einlesen;
data tmp14 (compress = yes);
    set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
        EF3 /* Nr. des Auswahlbezirks (systemfrei) */
        EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
        EF5b /* Jahresüberhang aus dem Vorjahr */
        EF12 /* Berichtsquartal */
        EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
        EF31 /* Bevölkerung: Privathaushalte */
        EF35 /* Stellung innerhalb der Lebensform */
        EF29 /* Erwerbstyp */
        EF541 /* Beruf (ISCO-08) */
        EF851) /* Beruf: Lebenspartner in der Lebensform (ISCO-08) */ ;
    if (EF35 = 2) then LP = 1; /* Ehefrau oder Lebenspartner der Bezugsperson in der Lebensform 1 = ja, 0 =
sonst */
    if (EF35 ^= 2) then LP = 0;
    x851 = (EF541*(EF35=2 & EF29=1)); /* Beruf (ISCO-08) | Lebenspartner in der Lebensform | Erwerbstätig */
run;

proc sort data = tmp14;
    by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Berufs (ISCO-08) des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF29 EF541 x851 LP where=(EF31=1)) nway
noprnt;
by EF1 EF3 EF4 EF5b EF12 EF27;
var x851 LP;
output out = sum(drop=_type_ _freq_)
sum = v851
n_LP;

```

```

run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v851 = . & EF31 = -8) then v851 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v851 = . & EF31 = -7) then v851 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v851 = . & EF31 = 2) then v851 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v851 = 0 & n_LP = 0) then v851 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
  if (v851 = 0 & n_LP = 1) then v851 = -2; /* Entfällt (Nichterwerbstätige) */
  if (v851 = 7314) then v851 = 7319;
  if (v851 = 8151) then v851 = 8153;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v851 ~= EF851));
  table v851 * EF851 / nocol norow nopercnt missing;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF852: Stellung im Beruf: Lebenspartner in der Lebensform;
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF29 /* Erwerbstyp */
    EF117 /* Stellung im Beruf */
    EF852) /* Stellung im Beruf: Lebenspartner in der Lebensform */ ;
  if (EF35 = 2) then LP = 1; /* Ehefrau oder Lebenspartner der Bezugsperson in der Lebensform 1 = ja, 0 =
sonst */
  if (EF35 ^= 2) then LP = 0;
  x852 = (EF117*(EF35=2 & EF29=1)); /* Stellung im Beruf | Lebenspartner in der Lebensform | Erwerbstätig */
run;

proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Stellung im Beruf des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF29 EF117 x852 LP where=(EF31=1)) nway
  noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x852 LP;
  output out = sum(drop=_type_ _freq_ )
    sum = v852
    n_LP;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v852 = . & EF31 = -8) then v852 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v852 = . & EF31 = -7) then v852 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v852 = . & EF31 = 2) then v852 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v852 = 0 & n_LP = 0) then v852 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
  if (v852 = 0 & n_LP = 1) then v852 = -2; /* Entfällt (Nichterwerbstätige) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v852 ~= EF852));
  table v852 * EF852 / nocol norow nopercnt missing;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF853: Überwiegender Lebensunterhalt: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */

```

```

EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF401 /* Überwiegender Lebensunterhalt */
EF853 /* Überwiegender Lebensunterhalt: Lebenspartner in der Lebensform */ ;
x853 = (EF401*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des überwiegenden Lebensunterhalts des Lebenspartners in der Lebensform, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF401 x853 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x853;
  output out = sum(drop= type freq )
  sum = v853;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v853 = . & EF31 = -8) then v853 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v853 = . & EF31 = -7) then v853 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v853 = . & EF31 = 2) then v853 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v853 = 0) then v853 = -5; /* Entfällt (Kein Lebenspartner vorhanden) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v853 ~= EF853));
  table v853 * EF853;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF854: Nettoeinkommen (im letzten Monat): Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
EF3 /* Nr. des Auswahlbezirks (systemfrei) */
EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
EF5b /* Jahresüberhang aus dem Vorjahr */
EF12 /* Berichtsquartal */
EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
EF31 /* Bevölkerung: Privathaushalte */
EF35 /* Stellung innerhalb der Lebensform */
EF436 /* Nettoeinkommen (im letzten Monat) */
EF854 /* Nettoeinkommen (im letzten Monat): Lebenspartner in der Lebensform */ ;
  x854 = (EF436*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Nettoeinkommens des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF436 x854 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x854;
  output out = sum(drop=_type_ _freq_ )
  sum = v854;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v854 = . & EF31 = -8) then v854 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v854 = . & EF31 = -7) then v854 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v854 = . & EF31 = 2) then v854 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v854 = 0) then v854 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v854 ~= EF854));
  table v854 * EF854;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

```

```

* EF855: Höchster allgemeiner Schulabschluss: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF310 /* Höchster allgemeiner Schulabschluss */
    EF855) /* Höchster allgemeiner Schulabschluss: Lebenspartner in der Lebensform */ ;

run;

* Variable EF310 recodieren;
data tmp14;
set <<<cmz14.suf2014>>>;
EF310n = EF310;
if EF310 = 6 then EF310n = 8;
if EF310 = 9 then EF310n = 7;
if EF310 = 7 then EF310n = 9;
if EF310 = -5 & EF309=9 then EF310n = 6;
if EF310 = -5 then EF310n = 8;
run;

* Erstellen der Hilfsvariable x855;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>>;
  x855= (EF310n*(EF35=2));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des höchsten allgemeinen Schulabschlusses des Lebenspartners in der Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF310n x855 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x855;
  output out = sum(drop=_type_ _freq_ )
  sum = v855;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v855 = . & EF31 = -8) then v855 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v855 = . & EF31 = -7) then v855 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v855 = . & EF31 = 2) then v855 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
run;

* Kontrolle;
proc freq data = tmp14 (where= (v855 ~= EF855));
  table v855 * EF855;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF856: Höchster beruflicher Abschluss: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF320 /* Höchster beruflicher Abschluss */
    EF856) /* Höchster beruflicher Abschluss: Lebenspartner in der Lebensform */ ;
  x856 = (EF320*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des höchsten beruflichen Abschlusses des Lebenspartners in der Lebensform, die zur Bevölkerung in Privat-
haushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF320 x856 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x856;
  output out = sum(drop=_type_ _freq_ )
  sum = v856;

```

```

run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v856 = . & EF31 = -8) then v856 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v856 = . & EF31 = -7) then v856 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v856 = . & EF31 = 2) then v856 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v856 = 0) then v856 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v856 ~= EF856));
  table v856 * EF856;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF857: Höchster beruflicher/schulischer Abschluss: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF540 /* Höchster beruflicher/schulischer Abschluss */
    EF857) /* Höchster beruflicher/schulischer Abschluss: Lebenspartner in der Lebens-
  form */ ;
  x857 = (EF540*(EF35=2)) /* Lebenspartner in der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des höchsten beruflichen/schulischen Abschlusses des Lebenspartners in der Lebensform, die zur Bevölkerung
  in Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF540 x857 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x857;
  output out = sum(drop=_type_ _freq_ )
  sum = v857;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v857 = . & EF31 = -8) then v857 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v857 = . & EF31 = -7) then v857 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v857 = . & EF31 = 2) then v857 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v857 = 0) then v857 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v857 ~= EF857));
  table v857 * EF857;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF858: Arbeitszeittyp : Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF130 /* Teilzeittätigkeit: Grund */
    EF858) /* Arbeitszeittyp: Lebenspartner in der Lebensform */ ;
run;

* Variable EF130 recodieren;
data tmp14;

```

```

set <<<cmz14.suf2014>>;
EF130n = EF130;
if EF130 = -5 then EF130n = 8;
if ((EF130>=1 & EF130<=9) & EF129=1) then EF130n = 8;
run;

* Erstellen der Hilfsvariable x858;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>;
  x858= (EF130n*(EF35=2));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe des Erwerbstyps des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten (EF31=1)
gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF130n x858 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x858;
  output out = sum(drop=_type_ _freq_ )
  sum = v858;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v858 = . & EF31 = -8) then v858 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v858 = . & EF31 = -7) then v858 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v858 = . & EF31 = 2) then v858 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v858 = 0) then v858 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v858 ~= EF858));
  table v858 * EF858;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF859: Normale Arbeitszeit (je Woche: Stunden): Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF131 /* Normale Arbeitszeit (je Woche: Stunden) */
    EF859) /* Normale Arbeitszeit (je Woche: Stunden): Lebenspartner in der Lebensform
*/ ;
  x859 = (EF131*(EF35=2)) /* Lebenspartner in der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der normalen Arbeitszeit des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF131 x859 where=(EF31=1)) nway noprint;
by EF1 EF3 EF4 EF5b EF12 EF27;
  var x859;
  output out = sum(drop=_type_ _freq_ )
  sum = v859;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v859 = . & EF31 = -8) then v859 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v859 = . & EF31 = -7) then v859 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v859 = . & EF31 = 2) then v859 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v859 = 0) then v859 = -5;
run;

```

```

* Kontrolle;
proc freq data = tmp14 (where= (v859 ~= EF859));
  table v859 * EF859;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF860: Grund geringerer Arbeitszeit: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF147 /* Geringere Arbeitszeit (Berichtswert): Grund */
    EF860 /* Grund geringerer Arbeitszeit: Lebenspartner in der Lebensform */ ;

run;

* Variable EF147 recodieren;
data tmp14;
  set <<<mz14.suf2014>>;
  EF147n = EF147;
  if (EF147=3 & EF29=1 & EF146=2) then EF147n = 2;
  if (EF147=4 & EF29=1 & EF146=2) then EF147n = 3;
  if (EF147=17 & EF29=1 & EF146=2) then EF147n = 5;
  if (EF147=8 & EF29=1 & EF146=2) then EF147n = 6;
  if (EF147=16 & EF29=1 & EF146=2) then EF147n = 7;
  if ((EF147=1 | EF147=2 | EF147=6 | EF147=18 | EF147=19 | EF147=99) | (EF147>=9 & EF147<=15)) & EF29=1 & EF146=2)
  then EF147n = 8;
  if (EF147=-5 & EF29=1 & EF146^=2) then EF147n = 1;
run;

* Erstellen der Hilfsvariable x860;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>>;
  x860= (EF147n*(EF35=2));
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe Grund geringerer Arbeitszeit des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF147n x860 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x860;
  output out = sum(drop=_type_ _freq_ )
  sum = v860;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v860 = . & EF31 = -8) then v860 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v860 = . & EF31 = -7) then v860 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v860 = . & EF31 = 2) then v860 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v860 = 0) then v860 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v860 ~= EF860));
  table v860 * EF860;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF861: Erwerbsunterbrechung (Berichtsw.): Lebenspartner in Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<mz14.suf2014>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF29 /* Erwerbstyp */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF85 /* Erwerbsunterbrechung (Berichtsw.): Grund */
    EF861 /* Erwerbsunterbrechung (Berichtsw.): Lebenspartner in Lebensform */ ;
  x861 = (EF85*(EF35=2)) /* Lebenspartner in der Lebensform */;

run;

```

```

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der Erwerbsunterbrechung des Lebenspartners in der Lebensform, die zur Bevölkerung in Privathaushalten
(EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF29 EF31 EF35 EF85 x861 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x861;
  output out = sum(drop=_type_ _freq_ )
  sum = v861;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v861 = . & EF31 = -8) then v861 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v861 = . & EF31 = -7) then v861 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v861 = . & EF31 = 2) then v861 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v861 = 0 & EF31 = 1) then v861 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v861 ~= EF861));
  table v861 * EF861;
run;
/* ACHTUNG: Abweichungen zur Originaltypisierung */

* EF862: 1. ausländische Staatsangehörigkeit: Lebenspartner in der Lebensform
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */
    EF12 /* Berichtsquartal */
    EF27 /* Nr. der Lebensform im Haushalt (Konzept der Lebensform) */
    EF31 /* Bevölkerung: Privathaushalte */
    EF35 /* Stellung innerhalb der Lebensform */
    EF369 /* 1. ausländische staatsangehörigkeit */
    EF862) /* 1. ausländische Staatsangehörigkeit: Lebenspartner in der Lebensform */ ;
  x862 = (EF369*(EF35=2)) /* Lebenspartner in der Lebensform */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Summe der 1. ausländischen Staatsangehörigkeit des Lebenspartners in der Lebensform, die zur Bevölkerung in
Privathaushalten (EF31=1) gehört;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF27 EF31 EF35 EF369 x862 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF27;
  var x862;
  output out = sum(drop=_type_ _freq_ )
  sum = v862;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF27;
run;

* Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v862 = . & EF31 = -8) then v862 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
Gemeinschaftsunterkünfte) */
  if (v862 = . & EF31 = -7) then v862 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v862 = . & EF31 = 2) then v862 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v862 = 0) then v862 = -5;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v862 ~= EF862));
  table v862 * EF862;
run;

* EF866: Kinder: Anzahl in Familie (trad. Familienkonzept)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<cmz14.suf2014>>> (keep = EF1 /* Bundesland */
    EF3 /* Nr. des Auswahlbezirks (systemfrei) */
    EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
    EF5b /* Jahresüberhang aus dem Vorjahr */

```



```

EF12 /* Berichtsquartal */
EF25 /* Nr. der Familie im Haushalt (traditionelles Familienkonzept) */
EF31 /* Bevölkerung: Privathaushalte */
EF34 /* Stellung innerhalb der Familie (trad. Familienkonzept) */
EF866 /* Kinder: Anzahl in Familie (trad. Familienkonzept) */ ;
x866 = (EF31=1 & EF34=3) /* Privathaushalt | Lediges Kind in der Familie */;

run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF25;
run;

* Summe der Kinder in traditionellem Familienkonzept, die zur Bevölkerung in Privathaushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF25 EF31 EF34 x866 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF25;
  var x866;
  output out = sum(drop= type freq )
  sum = v866;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF25;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v866 = . & EF31 = -8) then v866 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v866 = . & EF31 = -7) then v866 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v866 = . & EF31 = 2) then v866 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v866 >=9) then v866 = 9;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v866 ~= EF866));
  table v866 * EF866;
run;

* EF867: Kinder unter 18 Jahren: Anzahl in Familie (trad. Familienkonzept)
* Datenauswahl einlesen;
data tmp14 (compress = yes);
  set <<<tmp14.suf2014>>> (keep = EF1 /* Bundesland */
  EF3 /* Nr. des Auswahlbezirks (systemfrei) */
  EF4 /* Nr. des Haushalts im Auswahlbezirk (systemfrei) */
  EF5b /* Jahresüberhang aus dem Vorjahr */
  EF12 /* Berichtsquartal */
  EF25 /* Nr. der Familie im Haushalt (traditionelles Familienkonzept) */
  EF31 /* Bevölkerung: Privathaushalte */
  EF34 /* Stellung innerhalb der Familie (trad. Familienkonzept) */
  EF44 /* Alter */
  EF867) /* Kinder unter 18 Jahren: Anzahl in Familie (trad. Familienkonzept) */ ;
  x867 = (EF31=1 & EF34=3 & EF44<18) /* Privathaushalt, Lediges Kind in der Familie,
  über 18 Jahre */;
run;

* Sortieren nach IDs ;
proc sort data = tmp14;
  by EF1 EF3 EF4 EF5b EF12 EF25;
run;

* Summe der Kinder in traditionellem Familienkonzept mit der Eigenschaft EF44<18, die zur Bevölkerung in Privat-
  haushalten (EF31=1) gehören;
proc means data = tmp14 (keep = EF1 EF3 EF4 EF5b EF12 EF25 EF31 EF34 EF44 x867 where=(EF31=1)) nway noprint;
  by EF1 EF3 EF4 EF5b EF12 EF25;
  var x867;
  output out = sum(drop=_type_ _freq_ )
  sum = v867;
run;

* Zusammenspielen;
data tmp14 (compress = yes);
  merge tmp14 sum;
  by EF1 EF3 EF4 EF5b EF12 EF25;
run;

* Vergrößerung und Missing Werte zuweisen;
data tmp14;
  set tmp14;
  if (v867 = . & EF31 = -8) then v867 = -8; /* Entfällt (Leerstehende Wohnung, ausgefallene Privathaushalte und
  Gemeinschaftsunterkünfte) */
  if (v867 = . & EF31 = -7) then v867 = -7; /* Entfällt (Auswahlbezirk ohne befragten Haushalt) */
  if (v867 = . & EF31 = 2) then v867 = -1; /* Entfällt (Gemeinschaftsunterkunft) */
  if (v867 >=9) then v867 = 9;
run;

* Kontrolle;
proc freq data = tmp14 (where= (v867 ~= EF867));
  table v867 * EF867;
run;

/* Ende der Syntax */

```