

# FDZ-Arbeitspapier Nr. 44



Das SAS-Makro newvar.  
Entwicklung und Anwendung eines Hilfsinstruments  
zur effizienten Erstellung neuer Variablen in der DRG-Statistik

Tim Hochgürtel, Tobias Lösch

2012

## Impressum

Herausgeber: Statistische Ämter des Bundes und der Länder  
Herstellung: Statistisches Bundesamt - Forschungsdatenzentrum -  
Gustav-Stresemann-Ring 11, 65189 Wiesbaden  
Telefon 0611 75-4220 • Telefax 0611 75-3915  
Internet: <http://www.destatis.de>  
E-Mail: [forschungsdatenzentrum@destatis.de](mailto:forschungsdatenzentrum@destatis.de)

### Fachliche Informationen zu dieser Veröffentlichung:

Statistisches Bundesamt  
Forschungsdatenzentrum

Tel.: 0611 75-4220  
Fax.: 0611 75-3915  
[forschungsdatenzentrum@destatis.de](mailto:forschungsdatenzentrum@destatis.de)

### Informationen zum Datenangebot:

Statistisches Bundesamt  
Forschungsdatenzentrum

Tel.: 0611 75-4220  
Fax: 0611 72-3915  
[forschungsdatenzentrum@destatis.de](mailto:forschungsdatenzentrum@destatis.de)

Forschungsdatenzentrum der  
Statistischen Ämter der Länder  
– Geschäftsstelle –  
Tel.: 0211 9449-2873  
Fax: 0211 9449-8087  
[forschungsdatenzentrum@it.nrw.de](mailto:forschungsdatenzentrum@it.nrw.de)

Erscheinungsfolge: unregelmäßig  
Erschienen im März 2012

Diese Publikation wird kostenlos als PDF-Datei zum Download unter [www.forschungsdatenzentrum.de](http://www.forschungsdatenzentrum.de) angeboten.

© Information und Technik Nordrhein-Westfalen, Düsseldorf, 2012  
(im Auftrag der Herausbergemeinschaft)

Fotorechte Umschlag: © artSILENCEcom – Fotolia.com

Vervielfältigung und Verbreitung, nur auszugsweise, mit Quellenangabe gestattet. Alle übrigen Rechte bleiben vorbehalten.

**Bei den enthaltenen statistischen Angaben handelt es sich um eigene Arbeitsergebnisse der genannten Autoren im Zusammenhang mit der Nutzung der bereitgestellten Daten der Forschungsdatenzentren. Es handelt sich hierbei ausdrücklich nicht um Ergebnisse der Statistischen Ämter des Bundes und der Länder.**

# FDZ-Arbeitspapier Nr. 44

Das SAS-Makro newvar.  
Entwicklung und Anwendung eines Hilfsinstruments  
zur effizienten Erstellung neuer Variablen in der DRG-Statistik

Tim Hochgürtel, Tobias Lösch

2012



## **Das SAS-Makro *newvar*. Entwicklung und Anwendung eines Hilfsinstruments zur effizienten Erstellung neuer Variablen in der DRG-Statistik**

*Tim Hochgürtel*<sup>1</sup>, *Tobias Lösch*<sup>2</sup>

### **Zusammenfassung:**

Über die *Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder* ist es für externe Nutzer möglich, mit den Mikrodaten der DRG-Statistik (Fallpauschalenbezogenen Krankenhausstatistik) zu arbeiten. Im Rahmen der *kontrollierten Datenfernverarbeitung* müssen externe Nutzer jedoch mit langen Wartezeiten rechnen, da die Codes zur Analyse der DRG-Statistik sehr lange Rechenzeiten produzieren. Dies ist darauf zurückzuführen, dass es sich bei der DRG-Statistik um eine Vollerhebung mit einer Vielzahl alphanumerischer Variablen handelt. Daneben stellt die Analyse der DRG-Statistik hohe Anforderungen an die Programmierkenntnisse der Datennutzer.

Um eine effiziente Durchführung der Nutzeranalysen zu unterstützen wurde vom *Forschungsdatenzentrum des Statistischen Bundesamtes* das SAS-Makro *newvar* entwickelt. Mit Hilfe dieses Makros können neue Dummy- und Summen-Variablen auf Grundlage von Nebendiagnosen sowie Operationen- und Prozedurenschlüssel flexibel erstellt werden. Das Makro wählt hierbei in Abhängigkeit von nutzerspezifischen Parametern eine effiziente Methode zur Erstellung neuer Variablen. Ziel ist es, die Rechenzeiten der Analysen zu verkürzen.

Schlüsselworte: DRG-Statistik, Forschungsdatenzentrum, kontrollierte Datenfernverarbeitung, Makro

---

<sup>1</sup> Statistisches Bundesamt, [tim.hochguertel@destatis.de](mailto:tim.hochguertel@destatis.de)  
<sup>2</sup> Statistisches Bundesamt, [tobias.loesch@destatis.de](mailto:tobias.loesch@destatis.de)

## 1. Analyse der DRG-Statistik über das Forschungsdatenzentrum

Die *Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder* (FDZ) ermöglichen externen Nutzern die Analyse von Mikrodaten ausgewählter amtlicher Statistiken. Das Datenangebot umfasst hierbei auch die DRG-Statistik (Fallpauschalenbezogene Krankenhausstatistik). Eine Analyse der DRG-Statistik kann hierbei über die On-Site Zugangswege der *kontrollierten Datenfernverarbeitung* und des *Gastwissenschaftsarbeitsplatzes* realisiert werden.

Externe Nutzer haben die Möglichkeit in den Räumlichkeiten des *Forschungsdatenzentrums der Statistischen Ämter des Bundes und der Länder* den Zugang zu faktisch anonymen Mikrodaten der DRG-Statistik ab dem Erhebungsjahr 2008 zu erhalten. Gemäß § 16.6 Bundesstatistikgesetz (BStatG) ist die Nutzung des Gastwissenschaftlerarbeitsplatzes auf Datennutzer von *Hochschulen und sonstigen Einrichtungen mit Aufgabe der unabhängigen wissenschaftlichen Forschung* beschränkt (vgl. Zühlke u.a. 2003, Hochgürtel 2011).

Im Rahmen der kontrollierten Datenfernverarbeitung haben Nutzer die Möglichkeit Auswertungsprogramme an das FDZ zu übermitteln. Im FDZ werden die Mikrodaten der DRG-Statistik mit dem Auswertungsprogramm analysiert. Der Nutzer erhält seine Ergebnisse nach einer Geheimhaltungsprüfung. Zu beachten ist hierbei, dass eine Analyse der DRG-Statistik in der Regel nur mit der Statistiksoftware SAS möglich ist. Dies begründet sich durch die Größe der Mikrodatenfiles der DRG-Statistik. Daneben muss von den Nutzern die Wartezeit zwischen Einsenden des Auswertungsprogramms und dem Erhalt der Ergebnisse berücksichtigt werden. Die reine Laufzeit von Analyseprogrammen der Nutzer kann bei mehreren Wochen liegen. Auch dies ist unter anderem auf die Größe der Mikrodatenfiles zurückzuführen. Da immer mehrere Nutzerprojekte mit Daten der DRG-Statistik parallel durchgeführt werden, kann ein eingehendes Analyseprogramm häufig nicht direkt von den FDZ-Mitarbeitern gestartet werden. Da für die Durchführung von Projekten zur Analyse der DRG-Statistik von Seiten des FDZ nicht unbegrenzte Ressourcen eingesetzt werden können, hängt die Wartezeit unter anderem auch von der aktuellen Auslastung des FDZ ab.

Mit den Daten der DRG-Statistik wurden bereits eine Reihe von Projekten im FDZ erfolgreich umgesetzt (vgl. bspw. Stang / Stausberg 2009, Stang / Merrill / Kuss 2011). Aus diesen Projekten liegen im FDZ umfangreiche Erfahrungen bezüglich der Durchführung von Analysen externer Nutzer vor. Eine Schwierigkeit, der häufig in diesen Projekten begegnet wurde, liegt in den langen Rechenzeiten der Analyseprogramme. Diese langen Rechenzeiten machen den Projektverlauf sowohl für die FDZ-Mitarbeiter als auch für die externen Nutzer nur schwer kalkulierbar.

Die DRG-Statistik umfasst die Behandlungsfälle aller Krankenhäuser, welche nach § 21 Krankenhausentgeltgesetz (KHEntG) zur Meldung verpflichtet sind. Dies beinhaltet alle Krankenhäuser, welche nach dem DRG-Vergütungssystem abrechnen und dem Anwendungsbereich des § 1 KHEntG unterliegen. Unter anderem werden auch Behandlungsfälle von Krankenhäusern der Bundeswehr erfasst, sofern es sich um die Behandlung von Zivilpersonen handelt. Auch die Krankenhäuser der Berufsgenossenschaften tragen zur DRG-Statistik bei, wenn die Behandlungskosten durch die Krankenversicherung und nicht durch die Unfallversicherung getragen werden. In der DRG-Statistik sind jedoch die Behandlungsfälle von Krankenhäusern des Straf- und Maßregelvollzugs sowie von Polizeikrankenhäusern nicht enthalten (vgl. Spindler 2011, S. 349-351). Leistungen von psychiatrischen und psychosomatischen Einrichtungen nach § 17 Krankenhausgesetz (KHG) sind nicht zugänglich.

Mit der DRG-Statistik liegt somit eine Erhebung vor, die nahezu alle vollstationären Krankenhausbehandlungen in Deutschland als Vollerhebung abbildet. Erfasst werden hierbei verschiedene soziodemografische Variablen sowie umfangreiche Merkmale zu Diagnosen und zum Behandlungsverlauf von Krankenhauspatienten. So werden beispielsweise durchgeführte Operationen und medizinische Prozeduren genauso erfasst wie Haupt- und Nebendiagnosen. Der Anlage III kann eine Liste der Merkmale entnommen werden, welche im Kontext einer Analyse der DRG-Statistik zur Verfügung stehen.

Für jedes Erhebungsjahr bestehen separate Mikrodatenfiles. Diese jährlichen Files enthalten rund 17 Millionen Datensätze sowie mehrere hundert Variablen. Bei den Variablen handelt es sich zumeist um alphanumerische Variablen. Diese Mikrodatenfiles bilden das Ausgangsmaterial der Analysen in den Forschungsdatenzentren.

Die Mikrodatenfiles der DRG-Statistik enthalten pro Behandlungsfall einen Datensatz. Merkmale, welche für jeden Behandlungsfall mehrfach vorhanden sein können, werden daher in mehreren Variablen erfasst. Diese Variablen werden mit einem Variablennamen in die DRG-Statistik aufgenommen, welche eine laufende Nummer als Teil des Variablennamen enthält. Beispielsweise kann ein Patient über mehrere Nebendiagnosen verfügen. Diese sind auch in der DRG-Statistik enthalten.

Es existieren hierzu die Variablen *icd\_nd1*, *icd\_nd2*, ..., *icd\_nd89*. Ähnlich verhält es sich beim OPS-Prozedurenschlüssel. Da Patienten über mehrere dieser OPS-Prozedurenschlüssel verfügen können, werden diese in mehreren Variablen erfasst. Diese Variablen tragen ebenfalls eine laufende Nummer in der Variablenbezeichnung. So existieren zur Erfassung der OPS-Prozedurenschlüssel die Variablen *ops\_ko1*, *ops\_ko2*, ..., *ops\_ko100*.

Die Nebendiagnosen werden mit dem ICD10-Schlüssel und Operationen und Prozeduren mit dem OPS-Schlüssel erfasst. Bei beiden Klassifikationen handelt es sich um sogenannten hierarchische

Schlüssel. Ein ICD-Schlüssel enthält beispielsweise bis zu fünf Stellen. Mit jeder Stelle nimmt die Differenziertheit der Diagnose zu. Eine Diagnose kann etwa mit E6602 codiert sein:

- ▲ Die erste Stelle bezeichnet ein Krankheitskapitel aus der ICD10-Klassifikation. Mit E werden *Endokrine, Ernährungs- und Stoffwechselkrankheiten* bezeichnet.
- ▲ Die ersten drei Stellen erfassen eine differenzierte Diagnose. So ist *Adipositas* mit E66 kodiert.
- ▲ Eine weitere Differenzierung der Diagnose ist mit der vierten Stelle gegeben. E660 bezeichnet *Adipositas durch übermäßige Kalorienzufuhr*.
- ▲ Die fünfte Stelle enthält die differenzierteste Verfeinerung. E6602 kodiert die Diagnose *Adipositas durch übermäßige Kalorienzufuhr bei einem Body-Mass-Index [BMI] von 40 und mehr*.

Der OPS-Schlüssel weist einen vergleichbaren Aufbau auf. Die vollständigen Verzeichnisse des ICD10 sowie des OPS-Schlüssels können über das *Deutsche Institut für Medizinische Dokumentation und Information* (DIMDI) eingesehen werden. Ein Zugriff auf die Schlüssel ist über das Web-Angebot möglich (<http://www.dimdi.de>).

Um die Ausführung der Auswertungsprogramme externer Nutzer im Rahmen der kontrollierten Datenfernverarbeitung möglichst effizient zu realisieren, wurde vom FDZ das Makro *newvar* entwickelt. Dieses Makro wurde mit der Intention erstellt, für künftige Analysen ein Werkzeug bereitzustellen, welches eine effiziente Erstellung neuer Variablen für eine Analyse der DRG-Statistik ermöglicht. Das Makro *newvar* leistet somit einen Beitrag zur Reduzierung der Rechenzeiten von Analyseprogrammen der DRG-Statistik.

Die Nutzer benötigen häufig die Information, ob ein Merkmalsträger Teil einer bestimmten Subpopulation ist oder ob der Merkmalsträger nicht zu dieser Subpopulation gehört. Hierzu werden alle Variablen *icd\_nd1*, *icd\_nd2*, ..., *icd\_nd89* oder *ops\_ko1*, *ops\_ko2*, ..., *ops\_ko100* darauf hin geprüft, ob bestimmte Ausprägungen in den *icd\_nd*-Variablen bzw. *ops\_ko*-Variablen vorhanden sind oder nicht. Es wird eine Dummy-Variable gebildet, welche die Ausprägung 1 annimmt, wenn der Merkmalsträger zur Subpopulation gehört. Ansonsten nimmt die Dummy-Variable den Wert 0 an.

In ähnlicher Weise wird von Nutzern die Häufigkeit benötigt, mit der bestimmte Ausprägungen in den *icd\_nd*-Variablen oder *ops\_ko*-Variablen vorhanden sind. Hierbei wird eine neue Variable erstellt,

welche die Anzahl der Übereinstimmungen der *icd\_nd*-Variablen oder *ops\_ko*-Variablen mit einer vom Nutzer definierten Liste angibt.

Die Erstellung einer neuen Dummy-Variable oder Summen-Variable ist sehr zeitaufwendig, da hier eine Vielzahl von Abgleichen durchgeführt wird. Je nach Art der Programmierung wird unter Umständen jede Zelle der *icd\_nd*-Variablen oder *ops\_ko*-Variablen mit jeder Ausprägung der Nutzerliste durchgeführt. Eine solche Programmierung ist in vielen Fällen nicht effizient und führt dazu, dass die Laufzeiten der Analyseprogramme unnötig verlängert werden.

Um hierfür eine Abhilfe zu schaffen, wurde vom FDZ das Makro *newvar* entwickelt. Dieses Makro erlaubt die Erstellung von Dummy-Variablen und Summen-Variablen. Der Nutzer muss lediglich einige Parameter benennen. Das Makro wählt selbständig in Abhängigkeit der benannten Parameter eine effiziente Methode zur Erstellung einer neuen Dummy-Variable oder Summen-Variable. Die Verwendung des Makros gewährleistet somit, dass die Erstellung neuer Dummy-Variablen oder Summen-Variablen nicht durch die Wahl einer ineffizienten Methode unnötig in die Länge gezogen wird.

## 2. Das Makro *newvar*

Bei Makros handelt es sich im Allgemeinen um die Automatisierung standardisierter Abläufe. Ein Makro erlaubt die wiederholte Abarbeitung einer definierten Programmsequenz, die durch Parameter flexibel gesteuert werden kann (vgl. Schendera 2005, S. 147ff.).

Zu unterscheiden sind Makrodefinition und die Ausführung eines Makros. Die Makrodefinition wurde im FDZ realisiert. Somit steht ein leistungsfähiges Werkzeug zur Verfügung, dass von Nutzern bedarfsgerecht genutzt werden kann. Der Nutzer kann das Makro im Rahmen seiner Analysen direkt ausführen, ohne das Makro selbst definieren zu müssen. Der Makro-Code ist in Anlage II einsehbar.

Grundsätzlich dient das Makro der Erstellung neuer Variablen. Es kann eine Dummy-Variable erstellt werden, welche den Wert 1 annimmt, wenn ein Behandlungsfall in den Nebendiagnosen oder den OPS-Variablen mindestens einen bestimmten Wert aufweist. Sonst ist der Wert der neuen Variable 0. Das Makro ermöglicht ebenfalls die Erstellung einer Summen-Variable, welche die Häufigkeit des Auftretens bestimmter Werte beinhaltet.

In der *kontrollierten Datenfernverarbeitung* steht das Makro *newvar* direkt zur Verfügung. Um einen flexiblen Einsatz des Makros zu ermöglichen müssen fünf Parameter durch den Nutzer bestimmt werden. Parameter sind die flexiblen Elemente des Makros. Entsprechend können Parameter als Stellschrauben verwendet werden, um das Makro für die eigenen Bedürfnisse anzupassen. Folgende Parameter sind zu bestimmen:

- (1) Benennung des Mikrodatenfiles in welchem die neue Variable angelegt wird.
- (2) Definition einer Liste mit Werten, welche mit den Nebendiagnosen bzw. OPS-Variablen der Behandlungsfälle abgeglichen wird.
- (3) Angabe, ob die neue Variable als Abgleich mit den Nebendiagnosen oder OPS-Schlüssel erstellt wird.
- (4) Benennung, ob eine Dummy- oder Summen-Variable erstellt wird.
- (5) Name der neuen Variablen.

Abb. 1 zeigt die grundlegende Struktur des Makros *newvar*. Um das Makro *newvar* zu benutzen müssen durch den Nutzer fünf `%let` -Makrovariablen definiert werden. Diese dienen bei der Ausführung des Makros *newvar* als Parameter.

Abb. 1: Grundlegende Struktur des Makros *newvar*

```
/* Das Mikrodatenfile muss spezifiziert werden, wenn eine Anwendung
außerhalb eines data-Steps durchgeführt wird, z. B.: */
%let user_file = drgfile;

/* Liste mit den Values muss spezifiziert, z. B.: */
%let user_liste = 'H351', 'J662', 'M8615', 'G83', 'A';

/* Variablenblöcke, auf das Makro angewandt wird, müssen benannt
werden. Gültige Werte: "icd_nd" und "ops_ko" */
%let user_block = icd_nd;

/* Funktion, mit der neue Variablen erstellt werden sollen, muss be-
nannt werden. Gültige Werte: "Dummy" oder "Summe" */
%let user_funktion = summe;

/* Name der neue Variable */
%let user_name = neuevariable;

/* Ausführen des Makros (Diese Sequenz muss unverändert in das Nut-
zer-Programm übernommen werden) */
%newvar(file= &user_file, liste=%quote(&user_liste), block=&user_block,
funktion=&user_funktion, name=&user_name);
```

Ein Nutzer kann sich das Makro für die eigenen Analysen anpassen. Hierbei müssen die `%let` -Makrovariablen nach dem Gleichheitszeichen entsprechend modifiziert werden. Hierzu kann die Code-Sequenz aus Abb. 1 in den eigenen Analysecode integriert werden. Die einzelnen `%let` -Makrovariablen müssen nach dem Gleichheitszeichen für die eigene Analyse angepasst werden.

**(a) Die Makrovariable `user_file`:**

In einem ersten Schritt muss die Makrovariable `user_file` definiert werden. Hierbei ist zu unterscheiden, ob das Makro innerhalb oder außerhalb eines data-Steps verwendet wird. Wenn das Makro `newvar` innerhalb eines data-Steps verwendet wird, bleibt die Makrovariable `user_file` leer (Abb. 2). Die neuen Variablen werden dann im aktuellen File erstellt.

Abb. 2: Definition der Makrovariable `user_file` bei Verwendung des Makros `newvar` innerhalb eines data-Steps

```
/* Die Makrovariable muss als leer spezifiziert werden, wenn eine
Anwendung innerhalb eines data-Steps durchgeführt wird, z. B.: */
%let user_file = ;
```

Wird das Makro `newvar` außerhalb eines data-Steps verwendet, muss die Makrovariable `user_file` ein existierendes Mikrodatenfile bezeichnen. In diesem File wird die neue Variable erstellt (vgl. Abb. 3).

Abb. 3: Definition der Makrovariable `user_file` bei Verwendung des Makros `newvar` außerhalb eines data-Steps

```
/* Das Mikrodatenfile muss spezifiziert werden, wenn eine Anwendung
außerhalb eines data-Steps durchgeführt wird, z. B.: */
%let user_file = drgfile;
```

Es ist zwingend notwendig, dass die Variablen `icd_nd1, ..., icd_nd89` bzw. `ops_ko1, ..., ops_ko100` in jenem File enthalten sind, in dem die neuen Variablen erstellt werden. Falls das Makro innerhalb eines data-Steps verwendet wird, müssen die Variablen `icd_nd1, ..., icd_nd89` bzw. `ops_ko1, ..., ops_ko100` zuvor in das File eingelesen werden. Wird das Makro `newvar` außerhalb eines data-Steps verwendet, muss mit der Makrovariable `user_file` ein File benannt werden, welches die Variablen `icd_nd1, ..., icd_nd89` bzw. `ops_ko1, ..., ops_ko100` bereits enthält.

### (b) Die Makrovariable `user_liste`

Es muss eine Liste mit Merkmalsausprägungen spezifiziert werden. Wird eine Dummy-Variable erstellt, so nimmt diese den Wert 1 an, wenn die in `user_liste` genannten Werte mindestens einmal in den `icd_nd`-Variablen bzw. `ops_ko`-Variablen auftritt. Sonst ist die Dummy-Variable 0.

Wird eine Summen-Variable erstellt, enthält die neue Variable die Anzahl der Übereinstimmungen der `icd_nd`-Variablen bzw. `ops_ko`-Variablen mit der in der `user_liste` genannten Werten.

Bei der Erstellung der `user_liste` ist es möglich, sich die hierarchische Gliederung der Schlüssel zu Nutze zu machen. Soll beispielsweise eine Dummy-Variable erstellt werden, bei welcher der Wert 1 ist, wenn ein Behandlungsfall eine Nebendiagnose auf 3-Steller-Ebene aufweist, ist es ausreichend diesen 3-Steller anzugeben. Wenn die zu erstellende Dummy-Variable bei allen Behandlungsfällen den Wert 1 annehmen soll, falls in den Nebendiagnosen *Adipositas* diagnostiziert ist, so genügt es in `user_liste` den Code 'E66' zu benennen. Das Makro setzt alle Behandlungsfälle in der neuen Variable auf 1, wenn die ersten drei Stellen in mindestens einer Nebendiagnose mit 'E66' übereinstimmen, unabhängig davon wie die weiteren Stellen gefüllt sind.

Das Makro erlaubt in `user_liste` eine Liste mit Elementen beliebiger Länge. Das Beispiel in Abb. 4 enthält unter anderem 'A' als Element der Liste. Bei der Erstellung einer Dummy-Variable wird unter anderem diese 1, sobald eine Nebendiagnose vorhanden ist, welche mit 'A' beginnt. Wird eine Summen-Variable erstellt, wird hierbei unter anderem jeder Eintrag in den Nebendiagnosen `icd_nd1`, ..., `icd_nd89` gezählt, welcher mit 'A' beginnt. Die übrigen Elemente der Liste werden in äquivalenter Form bei der Erstellung der neuen Variable berücksichtigt.

Abb. 4: Definition der Makrovariable `user_liste`

```
/* Liste mit den Values muss spezifiziert werden, z. B.: */  
%let user_liste = 'H351', 'J662', 'M8615', 'G83', 'A';
```

Die Ergebnisse können sich in Abhängigkeit der Spezifikation der `user_liste` unterscheiden.

Wird eine Summen-Variable erstellt, welche die Häufigkeit des Auftretens eines bestimmten 4-Stellers auszählt, so können Ergebnisse differieren, wenn einerseits direkt nach dem 4-Steller und andererseits nach allen 5-Stellern gesucht wird, welche bezüglich der ersten vier Stellen mit dem 4-Steller

identisch sind. Ursache hierfür ist, dass nicht immer auch die letzten Stellen des ICD-Schlüssels in den Nebendiagnosen vorhanden sind. So können unvollständige Diagnosen auf einer 4-Steller-Ebene in die Summen-Variable eingehen, bei einer Suche nach 5-Stellern jedoch verloren gehen.

Bei der Spezifikation der `user_liste` wird empfohlen, auf Doppelbenennungen zu verzichten. Sonst werden bei der Erstellung von Summen-Variablen einzelne Nebendiagnosen mehrfach gezählt werden, wenn beispielsweise die Einträge 'A' und 'A1' angegeben werden. Da 'A1' eine Unterkategorie von 'A' darstellt, kann das Makro hier eine Diagnose unter Umständen mehrfach zählen. Valide Ergebnisse sind gewährleistet, wenn die Einträge in `user_liste` disjunkt sind.

Die Makrovariable `user_liste` kann eine beliebige Anzahl von Merkmalsausprägungen enthalten. Es ist zu beachten, dass jede Merkmalsausprägung mit einem einfachen Anführungszeichen (') beginnt und endet. Die Merkmalsausprägungen sind mit einem Komma getrennt. Das Makro kann nur dann fehlerfrei arbeiten, wenn die Formatanforderungen erfüllt sind.

### (c) Die Makrovariable `user_block`

Mit der Makrovariable `user_block` wird benannt, ob die neue Variable anhand der Nebendiagnose (`icd_nd`) oder der Ops-Kodes (`ops_ko`) gebildet wird.

Abb. 5: Definition der Makrovariable `user_block`

```
/* Variablenblöcke, auf welche das Makro angewandt wird, müssen  
benannt werden. Gültige Werte: "icd_nd" und "ops_ko" */  
%let user_block = icd_nd;
```

Gültige Einträge der Makrovariable `user_block` sind "icd\_nd" und "ops\_ko". Es muss darauf geachtet werden, dass die Variablen `icd_nd1, ..., icd_nd89` bzw. `ops_ko1, ..., ops_ko100` im Mikrodatenfile vorhanden sind, in welchem die neue Variable erstellt wird.

#### (d) Die Makrovariable `user_funktion`

Die Makrovariable `user_funktion` gibt an, ob die neue Variable als Dummy oder als Summe berechnet wird.

Abb. 6: Definition der Makrovariable `user_funktion`

```
/* Funktion, mit der neue Variablen erstellt werden sollen, muss be-
nannt werden. Gültige Werte: "Dummy" oder "Summe" */
%let user_funktion = summe;
```

Gültige Einträge der Makrovariable `user_funktion` sind "Dummy" und "Summe".

Die neue Variable nimmt im Falle einer Dummy-Variable den Wert 1 an, falls sich bezüglich der in `user_liste` spezifizierten Werten und den in `user_block` genannten Variablen Übereinstimmungen finden lassen. Sonst ist die Dummy-Variable 0.

Falls eine Summen-Variable erstellt wird, weist diese die Häufigkeit aus, mit der in den `user_block` Variablen Einträge zu finden sind, welche in `user_liste` angegeben sind. Falls keine Übereinstimmungen gegeben sind, ist die Summen-Variable 0.

#### (e) Die Makrovariable `user_name`

In der Makrovariable `user_name` muss benannt werden, welchen Namen die neue Variable tragen soll.

Abb. 7: Definition der Makrovariable `user_name`

```
/* Name der neue Variable */
%let user_name = neuevariable;
```

### (f) Ausführen des Makros *newvar*

Zur Ausführung des Makros muss die untere Zeile unverändert in den Programmcode integriert werden.

Abb. 8: Ausführen des Makros *newvar*

```
/* Ausführen des Makros (Diese Sequenz muss unverändert in das Nutzer-Programm übernommen werden) */  
%newvar(file= &user_file, liste=%quote(&user_liste), block=&user_block,  
funktion=&user_funktion, name=&user_name);
```

### 3. Beispielanwendungen des Makros *newvar*

Für eine Analyse werden zwei neue Variablen erstellt. Zum einen wird die Population der Patienten abgegrenzt, welche über die Nebendiagnose „*Adipositas durch übermäßige Kalorienzufuhr*“ verfügen. Hierzu soll eine Dummy-Variable „*adipositas*“ erstellt werden, die den Wert 1 annimmt, falls eine entsprechende Nebendiagnose vorliegt. Sonst soll die Dummy-Variable 0 sein. Die Diagnose „*Adipositas durch übermäßige Kalorienzufuhr*“ ist in der ICD-10 Klassifikation mit „E660“ kodiert.

Entsprechend können die %let-Makrovariablen als Parameter für das Makro *newvar* spezifiziert werden:

Die Abb. 11 zeigt, wie das Makro *newvar* innerhalb eines data-Steps verwendet wird. Hier bleibt die %let-Makrovariable *user\_file* leer. Abb. 12 zeigt, wie das Makro außerhalb eines data-Steps verwendet wird. Hier muss die %let-Makrovariable *user\_file* ein existierendes Mikrodatenfile bezeichnen, in welchem die neue Variable angelegt wird.

Da der Dummy den Wert 1 annehmen soll, falls bei einem Patienten in den Nebendiagnosen „*Adipositas durch übermäßige Kalorienzufuhr*“ diagnostiziert wurde, ist der entsprechende ICD-Kode zu benennen. Die %let-Makrovariable *user\_liste* wird entsprechend auf 'E660' gesetzt.

Das die neue Variable „*adipositas*“ auf Basis der Nebendiagnosen gebildet wird, ist als %let-Makrovariable *user\_block* der Wert *icd\_nd* zu benennen.

Da es sich bei der neu anzulegenden Variable „*adipositas*“ um eine Dummy-Variable handelt, muss das %let-Makrovariable *user\_funktion* mit dem Ausdruck *dummy* gefüllt werden.

Da die neu anzulegende Variable mit „*adipositas*“ bezeichnet wird, ist das `%let`-Makrovariable `user_name` entsprechend zu benennen.

Die `%let`-Makrovariable werden wie folgt definiert (die `%let`-Makrovariable `user_file` ist hierbei nicht mit aufgeführt):

Abb. 9: Beispiel für die Definition der `%let`-Variablen für eine Dummyerstellung

```
%let user_liste = 'E660';
%let user_block = icd_nd;
%let user_funktion = dummy;
%let user_name = adipositas;
```

Eine zweite Variable wird erstellt, welche die Anzahl der durchgeführten *diagnostische Laryngoskopien* und *diagnostische Pharyngoskopie* beinhaltet. Hierzu wird eine Summen-Variable „*diagnose*“ erstellt. Diese Variable gibt für jeden Behandlungsfall die Summe an, mit der in den `ops_ko`-Variablen die Ausprägungen '1610' (*diagnostische Laryngoskopien*) oder '1611' (*diagnostische Pharyngoskopie*) auftreten. Die `%let`-Makrovariable werden entsprechend wie folgt definiert:

Auch hier werden in `user_liste` die Ausprägungen definiert, welche zur Summenbildung berücksichtigt werden. Die `%let`-Makrovariable `user_block` wird mit „`ops_ko`“ gefüllt, da hier die Summenbildung der neuen Variable auf Basis der Operationen und Prozeduren realisiert wird.

Mit der `%let`-Makrovariable `user_funktion` wird die Funktion „`summe`“ benannt, die zur Bildung der neuen Variable verwendet wird. Mit `%let` -Makrovariable `user_name` wird der Name der neuen Variable mit „`diagnose`“ festgelegt.

Abb. 10: Beispiel für die Definition der `%let`-Variablen für eine Summenbildung

```
%let user_liste = '1610', '1611';
%let user_block = ops_ko;
%let user_funktion = summe;
%let user_name = diagnose;
```

Zu beachten ist, dass bei der Erstellung beider Variablen mit der `%let`-Makrovariable `user_liste` die hierarchische Struktur der ICD- und OPS-Schlüssel genutzt wird. So wird beispielsweise bei der Erstellung der Dummy-Variable „*adipositas*“ der Dummy den Wert 1 annehmen, wenn mindestens eine Ausprägung der Nebendiagnosen bezüglich der ersten vier Stellen den Wert 'E660' annimmt. Die Ausprägung der fünften Stelle in den Nebendiagnosen wird dabei nicht beachtet. Enthält ein Be-

handlungsfall eine Nebendiagnose mit dem Wert 'E6602' nimmt der Dummy den Wert 1 an, da die ersten vier Stellen dem Wert 'E660' entsprechen. Bei der Summenbildung zur Variable „diagnose“ wird äquivalent vorgegangen. Hier werden bei der `%let`-Makrovariable `user_liste` zwei OPS-Viersteller angegeben. Zur Erstellung der Summen-Variablen „diagnose“ trägt jede Übereinstimmung in den ersten vier Stellen eines OPS-Kodes bei. Auch hier werden die weiteren Stellen nicht beachtet.

Die unteren Beispiele illustrieren, wie die beiden Variablen mit Hilfe des Makros `newvar` berechnet werden können. Hierbei ist der Einsatz des Makros innerhalb eines `data`-Steps (a) von der Anwendung des Makros außerhalb eines `data`-Steps (b) zu unterscheiden.

### (a) Erstellung von Variablen innerhalb eines `data`-Steps

Abb. 11: Beispiel zur Anwendung des Makros `newvar` innerhalb eines `data`-Steps

```
data myfile;
set lib.fall_ex_2009 (keep=icd_nd1 - icd_nd89 opd_ko1 - opd_ko100);

/* Erstellung der Variable „adipositas“ */

%let user_file = ;
%let user_liste = 'E660';
%let user_block = icd_nd;
%let user_funktion = dummy;
%let user_name = adipositas;

%newvar(file= &user_file, liste=%quote(&user_liste), block=&user_block,
funktion=&user_funktion, name=&user_name);

/* Erstellung der Variable „diagnose“ */

%let user_file = ;
%let user_liste = '1610', '1611';
%let user_block = ops_ko;
%let user_funktion = summe;
%let user_name = diagnose;

%newvar(file= &user_file, liste=%quote(&user_liste), block=&user_block,
funktion=&user_funktion, name=&user_name);

run;
```

## (b) Erstellung einer Summen-Variable außerhalb eines data-Steps

Abb. 12: Beispiel zur Anwendung des Makros *newvar* außerhalb eines data-Steps

```
data myfile;
set lib.fall_ex_2009 (keep=icd_nd1 - icd_nd89 opd_ko1 - opd_ko100);
run;
/* Erstellung der ersten Variable*/

%let user_file = myfile;
%let user_liste = 'E660';
%let user_block = icd_nd;
%let user_funktion = dummy;
%let user_name = adipositas;

%newvar(file= user_file, liste=%quote(&user_liste), block=&user_block,
funktion=&user_funktion, name=&user_name);

/* Erstellung der zweiten Variable*/

%let user_file = myfile;
%let user_liste = 'df321', '123x';
%let user_block = ops_ko;
%let user_funktion = summe;
%let user_name = summel;

%newvar(file= user_file, liste=%quote(&user_liste), block=&user_block,
funktion=&user_funktion, name=&user_name);
```

## 4. Fazit

Eine Nutzung der DRG-Statistik ist für externe Nutzer über die Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder möglich. Hierbei stellt die DRG-Statistik außergewöhnlich hohe Anforderungen an die Programmierkenntnisse der externen Nutzer. Durch den Umfang der Mikrodaten der DRG-Statistik weisen die Auswertungsprogramme der externen Nutzer unter Umständen eine lange Rechenzeit aus. Dies macht die Kalkulation des zeitlichen Aufwandes zur Erstellung der Ergebnisse äußerst schwierig.

Das Makro *newvar* liefert einen Beitrag zur Effizienzsteigerung der DRG-Analysen externer Nutzer in den Forschungsdatenzentren. Mit dem Makro *newvar* steht ein Werkzeug bereit, welches die Erstellung neuer Variablen auf einfache Weise möglich macht. Gleichzeitig garantiert das Makro *newvar* die Verwendung einer Methodik zur Erstellung neuer Variablen, welche die Laufzeiten der Programme reduziert.

## 5. Literatur

- Herff, Wolfgang u.a. 2003: Tipps & Tricks -für einen leichteren Umgang mit der SAS Software, <http://www.uni-potsdam.de/u/ksfe2003/Anlagen/Tipps-2003-Version1.pdf>.
- Hochgürtel, Tim 2011: Improvement of data access. On the way to Remote Data Access in Germany, <http://isi2011.congressplanner.eu/pdfs/950948.pdf>.
- Krämer, Walter / Schoffer, Olaf / Tschiersch, Lars 2008: Datenanalyse mit SAS. Statistische Verfahren und ihre grafische Anwendung, 2. überarb. und erw. Auflage, Springer; Heidelberg.
- Satchi, Thiru o.J.: Using the Magical Keyword "INTO:" in PROC SQL, Paper 71-27, <http://www2.sas.com/proceedings/sugi27/p071-27.pdf>.
- Schnitzer, Hanspeter 2011: SAS Makro %CheckPars – Makroparametercheck Deluxe, [http://saswiki.org/images/c/cf/15\\_KSFE\\_2011\\_-\\_Schnitzer\\_-\\_SAS\\_Makro\\_CheckPars.pdf](http://saswiki.org/images/c/cf/15_KSFE_2011_-_Schnitzer_-_SAS_Makro_CheckPars.pdf).
- Schendera, Christian 2005: Datenmanagement mit SPSS: Kontrollierter und beschleunigter Umgang mit Datensätzen, Texten und Werten, Springer; Heidelberg.
- Spindler, Jutta 2011: Fallpauschalenbezogene Krankenhausstatistik Diagnosen und Prozeduren der Krankenhauspatienten auf Basis der Daten nach § 21 Krankenhausentgeltgesetz, in: Klauber, Jürgen / Friedrich, Jörg / Wasem, Jürgen 2011: Krankenhausreport 2011. Schwerpunkt: Qualität durch Wettbewerb, Schattauer; Stuttgart, S. 349-378.
- Stang, Andreas / Stausberg, Jürgen 2009: Inpatient management of patients with skin cancer in Germany: an analysis of the nationwide DRG-statistic 2005–2006, in: British Journal of Dermatology 161 (s3), S 99 – 106, Special Issue: Advances in the Management of UVR induced Skin Cancers.
- Stang, Andreas / Merrill, Ray M. / Kuss, Oliver 2011: Hysterektomien in Deutschland: Eine DRG-basierte nationenweite Analyse der Jahre 2005–2006, in: Dtsch Arztebl Int 108(30), S. 508-514.
- Wameling, Almuth / Andreas D. Cullmann 2006: Skript zum Kurs SAS-Programmierung für Fortgeschrittene, <http://www.uni-forst.gwdg.de/~acullma/pub/sas/sas3/scriptsas3.pdf>.
- Zühlke, Silvia / Zwick, Markus / Scharnhorst, Sebastian / Wende, Thomas 2003: Die Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder, in: Wirtschaft und Statistik 10/2003, S. 906-911.

**Anlagen:**

Anlage 1: Laufzeitanalyse verschiedener Methoden zur Erstellung neuer Variablen

Anlage 2: Makro-Code

Anlage 3: Schlüsselverzeichnis der DRG-Statistik

## **A1. Laufzeitanalyse verschiedener Methoden zur Erstellung neuer Variablen**

### **A1.1. Ziel**

Das Makro *newvar* verfolgt die Idee die Laufzeiten der Programme zur Auswertung der DRG-Statistik zu optimieren. Hierzu wählt das Makro in Abhängigkeit von nutzerdefinierten Parametern automatisch die effizienteste Methode, um eine neue Variable zu erstellen.

Mit Hilfe von Simulationen wurden die Laufzeiten von drei verschiedenen Methoden zur Erstellung neuer Variablen ermittelt und miteinander verglichen. Bei diesen Methoden handelt es sich um (1) die Nutzung von Indexvariablen mittels einer Array-Funktion und einer Schleife, (2) einer Kombination der Prozeduren Transpose und SQL und (3) eine Kombination der Prozeduren Transpose und Summary.

Alle drei Methoden realisieren einen Abgleich zwischen einer nutzerspezifizierten Liste mit möglichen Ausprägungen und den Ausprägungen eines Merkmalsträger für bestimmte Variablen und erstellen ausgehend von diesem Abgleich eine neue Variable. Diese ist entweder eine Dummy-Variable, die angibt ob eine der Eintragungen der Liste bei einem Merkmalsträger vorhanden ist (1) oder nicht (0). Alternativ wird eine Summen-Variable erstellt, welche die Häufigkeit der Übereinstimmungen zwischen der Liste und den entsprechenden Variablen für jeden Merkmalsträger angibt.

Die Simulation wird durchgeführt, indem neben den verschiedenen Berechnungsmethoden (Array-Schleife, Transpose-SQL, Transpose-Summe) und Formen der Variable (Dummy oder Summe) auch die Größe des Datenfiles (Anzahl der Merkmalsträger) und die Länge der Liste (Anzahl der Einträge in der Liste) kontrolliert werden. Es werden verschiedene Durchläufe ausgeführt, wobei jeweils eine Größe variiert und die anderen Größen konstant gehalten werden. Für jeden realisierten Durchlauf wird die reine Rechenzeit des Programms (CPU-Time) mit Hilfe der SAS-Option „Fullstimer“ (detaillierte Informationen über die Rechenzeiten und Speicherverbrauch) in die Log-Datei geschrieben, anschließend ausgelesen und gemeinsam mit den jeweiligen, für den Durchlauf spezifischen Parametern, in ein dafür angelegtes Datenfile geschrieben.

## **A1.2. Beschreibung der Methoden**

### **A1.2.1 Array & Schleife**

Mit einem Array kann in SAS eine Reihe von Variablen indiziert werden. Die indizierten Variablen können anschließend in einer Schleife verarbeitet werden (vgl. Krämer / Schoffer / Tschiersch 2008, S. 33ff.). Im Folgenden wird daher von einer Array-Schleife gesprochen. Bei den durchzuführenden Abgleichen führt dies dazu, dass jede Zelle der indizierten Variablen mit jedem Eintrag der nutzerspezifischen Liste durchgeführt wird.

Der Nachteil dieser Methode ist, dass auch solche Zellen abgeglichen werden, die mit Missings besetzt sind. Da in der DRG-Statistik der größte Anteil der Zellen zur Erfassung von Nebendiagnosen und OPS-Schlüssel Missings aufweisen, wird im Rahmen dieser Methode eine Vielzahl von Abgleichen durchgeführt, welche keine Treffer generieren können. Die Anzahl der durchzuführenden Abgleiche ergibt sich aus dem Produkt der Anzahl der Merkmalsträger, der Anzahl der indizierten Variablen und der Anzahl der Ausprägungen, welche in der nutzerspezifischen Liste genannt sind.

### **A1.2.2 Prozeduren Transpose & SQL / Summary**

Mittels der Verwendung der Methoden Transpose & SQL sowie Transpose & Summary werden die Anzahl der Abgleiche im Vergleich zur Array-Schleifen-Methode reduziert. Zunächst wird mit der Prozedur Transpose ein neuer Datenfile erzeugt, in welchem für jede Zelle der Abgleich-Variablen (Nebendiagnosen oder OPS-Schlüssel) eine Zeile angelegt wird. Daneben enthält das File eine Fallnummer, welche eine eindeutige Zuordnung einer Zeile zu einem Datensatz des Ausgangsfiles ermöglicht. In einem zweiten Schritt werden alle Zeilen, die bezüglich der Abgleichsvariablen ein Missing aufweisen, gelöscht. Da der Anteil der Missings bezüglich der Abgleichsvariablen sehr hoch ist, reduzieren sich die Anzahl der Datensätze im transponierten File. In einem nächsten Schritt werden die Abgleiche der Abgleichs-Variablen mit der nutzerspezifischen Liste durchgeführt. Die Anzahl der Abgleiche ergibt sich hierbei als Produkt der Anzahl der Datensätze des transponierten Files nach der Bereinigung durch die Löschung der Missings und der Anzahl der Einträge der nutzerspezifischen Liste. Im Vergleich zur Array-Schleifen-Methode reduzieren sich so die Anzahl der Abgleiche.

Erstellt wird eine dritte Variable im transponierten File. Diese nimmt den Wert 1 an, falls die Ausprägung der Abgleichsvariable in der nutzerspezifischen Liste vorhanden ist, sonst ist die Variable 0.

Da im transponierten Datenfile für jeden Datensatz des Ausgangsfiles (DRG-Statistik) mehrere Zeilen vorliegen können, die je nach dem Ergebnis des Abgleichs eine 1 oder eine 0 enthalten, muss dieser transponierte Datenfile noch bearbeitet werden. Hierfür stehen die Prozeduren SQL und Summary zur Auswahl.

Die Prozedur SQL ist hierbei so gestaltet, dass eine neue Variable im Ausgangsfile erzeugt wird, die bei der Erstellung einer Dummy-Variable den Wert 1 annimmt, wenn mindestens eine Ausprägung der dritten Variable im transponierten File den Wert 1 aufweist. Soll eine Summenvariable erzeugt werden, so wird ebenfalls über die Prozedur SQL eine Summe über alle Zeilen eines Merkmalsträgers im transponierten File gebildet.

Zum gleichen Ergebnis führt die Prozedur Summary. Hierbei wird für jeden Merkmalsträger des Ausgangsfiles im transponierten File hinsichtlich der dritten Variable das Maximum (im Falle der Erstellung einer neuen Dummy-Variable) oder die Summe (im Falle der Erstellung einer neuen Summenvariable) ermittelt.

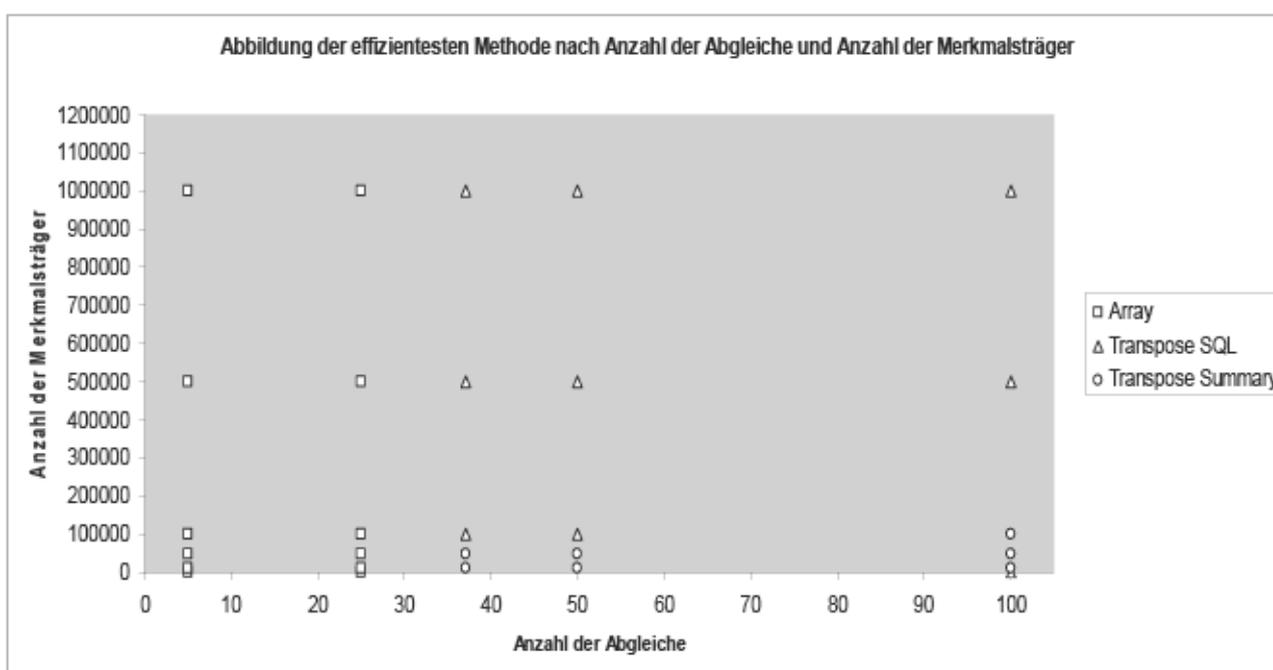
In beiden Fällen muss der transponierte und mit den Prozeduren SQL oder Summary bearbeitete Datenfile wieder an das Ausgangsmaterial angespielt werden. Hierbei wird ein Matching anhand der spezifischen Fallnummer durchgeführt. Zuletzt werden die Zeilen/Zellen, die ein Missing enthalten und nicht zum Abgleich herangezogen - aus dem transponierten Datenfile gelöscht - wurden, mit einer 0 besetzt.

### **A1.3. Vergleich der CPU-Time**

In SAS lassen sich verschiedene Laufzeiten für ein Programm ausgeben. Die „Realtime“, welche der Zeit zwischen Starten und Beenden eines Programms entspricht, ist die Summe der „System CPU Time“ und der „User CPU Time“. Die „System CPU Time“ gibt die Zeit an, die SAS für systemabhängige Rechenschritte oder –pausen benötigt. Sie ist vom Code des Nutzers nicht zu beeinflussen. Die „User CPU Time“ hingegen beschreibt ausschließlich die Zeit, die SAS für die Ausführung des vom Nutzer programmierten Codes benötigt. Beide Zeiten schließen sich gegenseitig aus und addieren sich zur Realtime. Wenn die Laufzeiten der verschiedenen Methoden verglichen werden, um die effizienteste Methode festzulegen, so sollte lediglich die CPU-Time für den Vergleich herangezogen werden. Andernfalls könnte die „System CPU Time“ das Ergebnis des Vergleichs verzerren, da diese nicht vom Code des Nutzers bedingt wird.

## A1.4. Ergebnisse

Anhand der erzielten Ergebnisse für die CPU-Time wird deutlich, dass die Array-Schleife unabhängig von der Anzahl der Merkmalsträger schneller ist als die beiden Alternativen, solange eine bestimmte Anzahl von Abgleichen nicht überschritten wird. Die folgende Grafik zeigt einen Ausschnitt der Ergebnisse der Laufzeitanalyse und bildet für verschiedene Durchläufe die jeweils effizienteste Berechnungsmethode ab.



Ausgehend von den Ergebnissen der oberen Abbildung lässt sich die Anzahl der Abgleiche, ab der die Berechnungsmethoden mit der Prozedur Transpose schneller sind, auf zwischen 25 und 37 Abgleiche eingrenzen. Folgende Tabelle zeigt den relevanten Ausschnitt der Ergebnistabelle für Abgleiche bei einer Listenlänge von 28 bis 30 für unterschiedliche Anzahl von Merkmalsträgern und zu erstellenden Variablen.

**Abb 2: CPU-Time der verschiedenen Berechnungsmethoden nach Form der Variable, Anzahl der Merkmalsträger und Anzahl der Abgleiche**

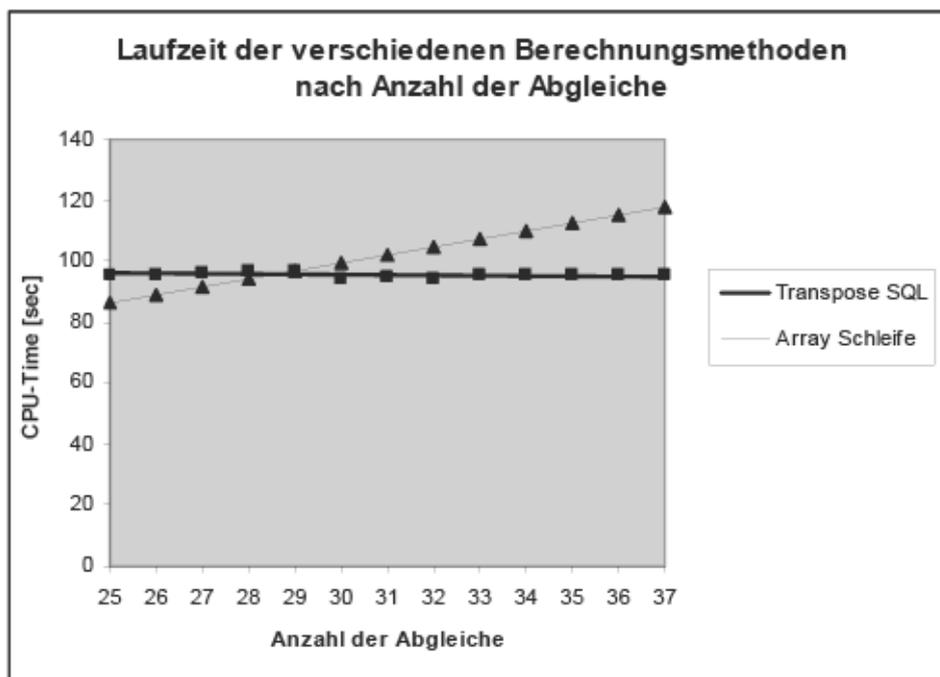
neue Variable	Anzahl der Merkmalsträger	Anzahl der Abgleiche	Array-Schleife	Transpose Summary	Transpose SQL
dummy	1000000	27	91,51	94,02	96,49
dummy	1000000	28	94,06	94,27	96,8
dummy	1000000	29	96,68	94,07	96,85
dummy	1000000	30	99,31	94,47	94,33
summe	1000000	27	91,9	96,99	94,31
summe	1000000	28	94,02	96,71	94,13
summe	1000000	29	96,93	95,33	94,09
summe	1000000	30	99,62	95,11	94,35
dummy	2000000	27	182,94	187,26	191,23
dummy	2000000	28	188,12	187,95	193,07
dummy	2000000	29	193,36	188,39	192,38
dummy	2000000	30	198,52	187,67	192,39
summe	2000000	27	182,52	188,21	187,72
summe	2000000	28	188,16	188,66	187,37
summe	2000000	29	193,08	188,68	187,47
summe	2000000	30	198,18	189,31	187,44
dummy	3000000	27	274,37	289,3	289,1
dummy	3000000	28	282,14	289,2	285,25
dummy	3000000	29	290,11	289,67	286,46
dummy	3000000	30	297,88	289,52	284,62
summe	3000000	27	273,79	281,38	281,59
summe	3000000	28	281,6	280,96	282,63
summe	3000000	29	289,54	282,27	281,6
summe	3000000	30	297,83	282,32	281,86
dummy	4000000	27	365,97	386	376,69
dummy	4000000	28	376,34	385,75	379,78
dummy	4000000	29	386,84	386,95	372,67
dummy	4000000	30	397,18	387,18	382,04
summe	4000000	27	365,11	374,79	375,88
summe	4000000	28	375,5	375,45	376,5
summe	4000000	29	385,92	376,47	377,46
summe	4000000	30	396,35	376,54	376,26
dummy	5000000	27	457,36	482,07	467,31
dummy	5000000	28	470,23	481,96	479,17
dummy	5000000	29	483,47	482,24	479,82
dummy	5000000	30	496,39	483,41	470,08
summe	5000000	27	456,21	471,27	473,83
summe	5000000	28	469,32	469,15	472,17
summe	5000000	29	482,32	469,25	477,54
summe	5000000	30	495,4	471,62	476,93

Die Betrachtung aller Ergebnisse lässt keinen eindeutigen Rückschluss zu, wann die Prozedur SQL oder die Prozedur Summary effizienter zu einer Berechnung der neuen Variable führt. Tendenziell lässt sich ein geringfügiger Effizienzgewinn mit der Prozedur SQL gegenüber der Prozedur Summary erzielen. Daher wird fortan nur noch zwischen der Array-Schleife und der Kombination der Prozeduren Transpose und SQL unterschieden.

Daneben ist festzustellen, dass die verschiedenen Formen der zu berechnenden Variable (Dummy oder Summe) keine Unterscheidung der Berechnungsmethoden veranlassen.

Die Auswertung der Ergebnisse im Bereich von 25 – 37 Abgleichen zeigte, dass unabhängig von der Anzahl der Merkmalsträger ab 28 Abgleichen die Berechnungsmethode über die Prozeduren Transpose und SQL der Array-Schleife vorzuziehen ist.

Die folgende Grafik zeigt einen Ausschnitt des Laufzeitverlaufs exemplarisch für einen Abgleich über eine Million Merkmalsträger.



Bei einer höheren Anzahl von Merkmalsträgern erfolgt eine Verschiebung der Grafen entlang der Ordinatenachsen. Die Laufzeiten werden also länger. Gleichzeitig erfolgt nur eine geringfügige Verschiebung auf der Abszissenachse. Der Schnittpunkt bleibt dabei konstant im Bereich von 28 Abgleichen. Ausgehend von dieser Feststellung kann man ab einer Liste von >28 Abgleichen die Prozeduren Transpose und SQL nutzen um eine effiziente Berechnung der neuen Variable zu erzielen. Bei einer Liste von  $\leq 28$  Abgleichen ist die Berechnung über die Array-Schleife zu bevorzugen.

#### **A1.5. Entscheidungsregel des Makros *newvar***

Das Makro *newvar* verwendet die Methode der Array-Schleife wenn ein Nutzer weniger als 29 Ausprägungen in einer Abgleichsliste spezifiziert hat. Falls die Anzahl der Ausprägungen in der Nutzerliste bei mehr als 28 liegt, wird auf die Methode Transpose und SQL zurückgegriffen.

## A2. Makro-Code

```

/*****
/* Mainz / Wiesbaden, im Dezember 2011
/* Version 20120114
/* Tim Hochgürtel, Tobias Lösch (FDZ)
/*****/

/***** Makro "newvar" zur Erstellung neuer Variablen der DRG-Statistik *****/
/*****/

/** Intention *****/
/* den Nutzer der kontrollierten Datenfernverarbeitung soll ein Makro bereitgestellt
/* werden, dass die effiziente Erstellung neuer Variablen ermöglicht. Diese neue
/* Variablen wird auf Basis der Variablen icd_nd oder ops_ko erstellt. Es können
/* Dummies oder Summen erstellt werden.
/*****/

/**** Struktur des Programms *****/
/* 1. Schritt: Beginn Makro
/* 2. Schritt: Prüfung: Sind Parameter richtig spezifiziert worden vom Nutzer?
/* 3. Schritt: Ermittlung des Dateinamen für Makrobenutzung innerhalb eines Data-Steps
/* 4. Schritt: Ermittlung der Anzahl der Values die in der Liste spezifiziert wurden
/* 5. Schritt: Ermittlung der Anzahl der Blockvariablen
/* 6. Schritt: Erstelle Makros mit Listeneinträgen und Länge der Listeneinträgen
/* 7. Schritt: Hilfsmakro für Funktionswahl
/* 8. Schritt: Auswahl der Methode
/* 9. Schritt: Methode I: array
/* 10. Schritt: Methode II:Transpose/SQL
/* 11. Schritt: SchlieÙe Makro
/*****/

/*****/
/* 1. Schritt: Beginn Makro
/*****/

%macro newvar(file, liste, block, funktion, name);

/*****/
/* 2. Schritt: Prüfung: Sind Parameter richtig spezifiziert worden vom Nutzer?
/*****/

/** Prüfe file-Makro: Ist angegebene Datei vorhanden ? *****/
%if &file ne %then %do;
  %local existenz;
  %str(data _null_);
  if 0 then set &file;
  stop;
  run;
);
%if &syserr=0 %then %let existenz=yes;
%else %let existenz=no;

%if (&existenz=yes) %then %put HINWEIS: Die angegebene Datei &file existiert.;
%else %do;
  %put FEHLER: Die als Parameter FILE benannte Datei &file existiert nicht.
  Ausführung wurde abgebrochen.;
  %goto ENDE;
%end;
%end;

```

```

/** Prüfe liste-Makro: Ist die Variablenliste gefüllt ? *****/
%if &liste ne %then %put HINWEIS: Das Makro Liste enthält Einträge.;
%else %do;
  %put Fehler: Das Makro Name LISTE hat keinen gültigen Eintrag.
  Ausführung wurde abgebrochen.;
  %goto ENDE;
%send;

/** Prüfe block-Makro: Hat Blockmakro gültigen Wert ? *****/
%if &block = ops_ko or &block = icd_nd %then
  %put HINWEIS: Das Blockmakro hat mit &block einen gültigen Wert.;
%else %do;
  %put FEHLER: Das Makro Block hat keinen gültigen Eintrag.
  Ausführung wurde abgebrochen.;
  %goto ENDE;
%send;

/** Prüfe funktion-Makro: Hat Funktionsmakro gültigen Wert ? *****/
%if &funktion = summe or &funktion = dummy %then
  %put HINWEIS: Das Funktionsmakro hat mit &funktion einen gültigen Wert.;
%else %do;
  %put FEHLER: Das Makro Funktion hat keinen gültigen Eintrag.
  Ausführung wurde abgebrochen.;
  %goto ENDE;
%send;

/** Prüfe name-Makro: Ist Name der neuen Variable angegeben ? *****/
%if &name ne %then
  %put HINWEIS: Ein Name für die neue Variable ist mit &name angegeben.;
%else %do;
  %put FEHLER: Das Makro Name hat keinen gültigen Eintrag.
  Ausführung wurde abgebrochen.;
  %goto ENDE;
%send;

/*****
/* 3. Schritt: Ermittlung des Dateinamen für Makrobenutzung innerhalb eines Data-Steps*/
*****/

%if &file = %then %do;
  %local varianteB;
  %let varianteB = 1;

  proc contents data=_last_ out=filename noprint;
  run;

  proc sql noprint;
    select memname into: name_last_
    from filename;
  quit;

  proc datasets library = work nolist;
    delete filename;
  run;

  %local file;
  %let file = &name_last_;
%send;

```

```

/*****
/* 4. Schritt: Ermittlung der Anzahl der Values die in der Liste spezifiziert wurden */
/*****
/* Hinweis: das Makro "anzahl_eintraege" enthält Anzahl der Einträge in der Liste */

%local anzahl_eintraege;

proc iml;
  call symput('anzahl_eintraege',left(char(nrow({&liste})))));
quit;

/***** Blanks löschen *****/
%let anzahl_eintraege= %eval(&anzahl_eintraege*1);

/*****
/* 5. Schritt: Erstelle Makros mit Listeneinträgen und Länge der Listeneinträgen */
/*****

proc iml;
  a = t(1:&anzahl_eintraege);
  b = {&liste};
  c = length(b);
  create listfile var{a b c};
  append;
  close listfile;
quit;

proc sql noprint;
  select distinct b
  into:bl - :b&anzahl_eintraege
  from listfile
  order by a;
run;

/* Hinweis: Die Makros bl,...,b&anzahl_eintraege beinhalten die einzelnen List-Einträge*/

proc sql noprint;
  select distinct c,c format 1.
  into:Ml separated by ""
  from listfile
  order by a;
run;

/* Hinweis: Das Makro Ml enthält die Länge der einzelne Listeeinträge als Folge *****/

proc datasets library=work;
  delete listfile;
run;

/*****
/* 6. Schritt: Ermittlung der Anzahl der Blockvariablen */
/*****

%local maxvalue;

%if &block = icd_nd %then %let maxvalue=89;
%else %let maxvalue=100;
/*****
/* 7. Schritt: Hilfsmakro für Funktionswahl */
/*****

%local func;

%if &funktion = dummy %then %let func=max;
%else %let func=sum;

```

```

/*****
/* 8. Schritt: Auswahl der Methode
/*****

%if &anzahl_eintraege > 28 %then %goto TRANSPOSE;

/*****
/* 9. Schritt: Methode I: array
/*****

%if varianteB ne 1 %then %do;

    data &file (compress=yes);
        set &file;

%end;

&name=0;
help=0;

array ma {&maxvalue} &block.1 - &block&maxvalue;

%do i = 1 %to &anzahl_eintraege;
    %local list;
    %let list = &b&i;
    %local laenge;
    %let laenge = %substr(&M1, &i,1);

    do j = 1 to &maxvalue;
        if (substr(ma(j), 1, &laenge) in("&list")) then help = help + 1;
    end;

%end;

%if &funktion=dummy %then %do;
if help > 0 then &name=1;
%end;
%else %do;
&name = help;
%end;

drop j help;

run;

%goto ENDE;

/*****
/* 10. Schritt: Methode II: Transpose/SQL
/*****

%TRANSPOSE:

%if varianteB ne 1 %then %do;
    data &file (compress=yes);
        set &file;
%end;

lfdnr= _N_;
run;

proc transpose data=&file out=data_transpose;
    var &block.1 - &block.&maxvalue.;
    by lfdnr;
run;

```

```

data data_transpose (compress=yes);
  set data_transpose;
  if coll = "" then delete;
  drop _NAME_;
  helpvar=0;

%do i = 1 %to &anzahl_eintraege;
  %local list;
  %let list = &b&i;
  %local laenge;
  %let laenge = %substr(&M1, &i,1);
  if (substr(coll, 1, &laenge) in: ("&list")) then helpvar=1;

%end;
run;

proc sql;
  create table A
  as select lfdnr, &func(helpvar) as &name
  from data_transpose
  group by lfdnr;
quit;

data &file (compress=yes);
  merge &file (in=a) A (in=b);
  by lfdnr;
  if a then output;
run;

data &file (compress=yes);
  set &file;
  if &name=. then &name=0;
  drop lfdnr;
run;

proc datasets library = work nolist;
  delete A data_transpose;
run;

/*****
/* 11. Schritt: SchlieÙe Makro */
*****/

%ENDE:
%mend newvar;

/*****
/***** Ende *****/
*****/

```

### A3. Schlüsselverzeichnis der DRG-Statistik

Quelle	Variable	Format	Länge	Wdh.	Inhalt/Bemerkungen
Algor.	typ_fall	a	1	1	Filter "vollstationär" (Auswertungen erfolgen ausschließlich für vollstationäre Patienten)  1 = Ja 2 = Nein
Algor.	typ_bereich	a	1	1	Filter für reine DRG-Fälle (Auswertungen erfolgen ausschließlich für DRG-Patienten)  1 = DRG 2 = PSY/3 = PIA
Algor.	kh_land	a	2	1	Land des Instituts
Algor.	kh_rb	a	1	1	Regierungsbezirk des Instituts
Algor.	kh_kreis	a	2	1	Kreis des Instituts
Algor.	kh_gem	a	3	1	Gemeinde des Instituts
Algor.	kh_typ_gem	a	2	1	Siedlungsstrukturelle Gebietstypen der Institute (17er-Gliederung - Erläuterung unter <a href="http://www.bbsr-bund.de">www.bbsr-bund.de</a> )
Algor.	kh_plz	a	5	1	PLZ des Instituts
Algor.	pat_land	a	3	1	Patienten-Land, auch Ausland
Algor.	pat_rb	a	1	1	Patienten-Regierungsbezirk
Algor.	pat_kreis	a	2	1	Patienten-Kreis
Algor.	pat_gem	a	3	1	Patienten-Gemeinde
Algor.	pat_typ_gem	a	2	1	Siedlungsstrukturelle Gebietstypen der Patienten (17er-Gliederung - Erläuterung unter <a href="http://www.bbsr-bund.de">www.bbsr-bund.de</a> )
Algor.	pat_plz	a	5	1	PLZ des Patienten
Datei 4	sex	a	1	1	Geschlecht
Algor.	alter	n	3	1	Alter (999 = unbekannt)
Algor.	typ_alter	a	2	1	Alter gruppiert  01 = alter >= 0 und alter < 1 02 = alter < 5 03 = alter < 10 04 = alter < 15 05 = alter < 20 06 = alter < 25 07 = alter < 30 08 = alter < 35 09 = alter < 40 10 = alter < 45 11 = alter < 50 12 = alter < 55 13 = alter < 60 14 = alter < 65 15 = alter < 70 16 = alter < 75 17 = alter < 80 18 = alter < 85 19 = alter < 90 20 = alter < 95 21 = alter <= 110 22 = alter = 999 (unbekannt)

Neu ab BJ 2010

Algor.	typ_geb	a	1	1	Neugeborenes (im Geburtsmonat aufgenommen) 1 = Ja 2 = Nein
Datei 4	aufn_anl	a	1	1	Aufnahmearbeit E = Einweisung durch einen Arzt Z = Einweisung durch einen Zahnarzt N = Notfall R = Aufnahme nach vorausgehender Behandlung in einer Rehabilitationseinrichtung V = Verlegung mit Behandlungsdauer im verlegenden Krankenhaus länger als 24 Stunden A = Verlegung mit Behandlungsdauer im verlegenden Krankenhaus bis zu 24 Stunden (für Aufnahmen ab dem 1.1.2007) G = Geburt B = Begleitperson
Datei 4	aufn_grd	a	4	1	Aufnahmegrund 01xx Krankenhausbehandlung, vollstationär 02xx Krankenhausbehandlung, vollstationär mit vorausgegangener vorstationärer Behandlung 03xx Krankenhausbehandlung, teilstationär 04xx vorstationäre Behandlung ohne anschließende vollstationäre Behandlung 05xx stationäre Entbindung 06xx Geburt 07xx Wiederaufnahme wegen Komplikationen (Fallpauschale) nach KFPV 2003 08xx Stationäre Aufnahme zur Organentnahme  xx entspricht den nach Schlüssel 1 der Anlage 2 zur § 301-Vereinbarung möglichen Werte
Datei 4	aufn_gew	n	4	1	Aufnahmegewicht in Gramm  Nur bei Kindern bis zur Vollendung des 1. Lebensjahres, bei Neugeborenen zählt das Geburtsgewicht.
Datei 4	entl_grd	a	3	1	Entlassungsgrund 01x = Behandlung regulär beendet 02x = Behandlung regulär beendet, nachst. Beh.vorgesehen 03x = Behandlung aus sonstigen Gründen beendet 04x = Behandlung gegen ärztlichen Rat beendet 059 = Zuständigkeitswechsel des Kostenträgers (bei tagesbezogenen Entgelten) 069 = Verlegung in ein anderes Krankenhaus 079 = Tod 089 = Verlegung in ein anderes Krankenhaus im Rahmen einer Kooperation 099 = Entlassung in eine Rehabilitationseinrichtung 109 = Entlassung in eine Pflegeeinrichtung 119 = Entlassung in ein Hospiz 139 = externe Verlegung zur psychiatrischen Behandlung 14x = Behandlung aus sonstigen Gründen beendet, nachstationäre Behandlung vorgesehen 15x = Behandlung gegen ärztlichen Rat beendet, nachstationäre Behandlung vorgesehen 179 = interne Verlegung mit Wechsel zwischen den Entgeltbereichen der DRG-Fallpauschalen, nach der BpflV oder für besondere Einrichtungen nach § 17b Abs. 1 Satz 15 KHG 229 = Fallabschluss (interne Verlegung) bei Wechsel zwischen voll- und teilstationärer Behandlung  x entspricht den nach Schlüssel 5 der Anlage 2 zur § 301-Vereinbarung möglichen Werte

Quelle	Variable	Format	Länge	Wdh.	Inhalt/Bemerkungen
Datei 4	vorstat_tage	n	4	1	Behandlungstage vorstationär
Datei 4	nachstat_tage	n	4	1	Behandlungstage nachstationär
Datei 4	andere_tage	n	4	1	Belegungstage in einem anderen Entgeltbereich (ein Wechsel zwischen den Entgeltbereichen in Folge einer internen oder externen Verlegung mit anschließender Wiederaufnahme bzw. Rückverlegung in den ursprünglichen Entgeltbereich)
Datei 4	urlaub_psy	n	4	1	Urlaubstage im BpflV-Entgeltbereich
Datei 2	ohne_tage	n	4	1	Tage ohne Berechnung/Behandlung
Algor.	tage	n	5	1	Verweildauer (Stundenfälle werden als 1 Tag berechnet)
Algor.	typ_vwd	a	2	1	Typ Verweildauer 01 = std_fall = 1 02 = tage = 1 03 = tage = 2 04 = tage = 3 05 = tage = 4 06 = tage = 5 07 = tage = 6 08 = tage = 7 09 = tage <= 9 10 = tage <= 12 11 = tage <= 14 12 = tage <= 21 13 = tage <= 28 14 = tage <= 35 15 = tage <= 42 16 = tage <= 70 17 = tage <= 182 18 = tage <= 365 19 = tage <= 99999 99 = sonst
Algor.	std_fall	a	1	1	Stundenfall 1 = Ja 2 = Nein
Algor.	icd_hd3	a	3	1	ICD-Kode 3Steller Hauptdiagnose Diagnoseschlüssel in der gültigen ICD-10GM-Version analog zur § 301-Vereinbarung
Algor.	icd_hd4	a	4	1	ICD-Kode 4Steller Hauptdiagnose Diagnoseschlüssel in der gültigen ICD-10GM-Version analog zur § 301-Vereinbarung
Datei 5	icd_hd	a	5	1	ICD-Kode Hauptdiagnose 5Steller (bis zum kodierbaren Endpunkt) Diagnoseschlüssel in der gültigen ICD-10GM-Version analog zur § 301-Vereinbarung
Datei 5	icd_nd	a	5	89	ICD-Kode Nebendiagnose Diagnoseschlüssel in der gültigen ICD-10GM-Version analog zur § 301-Vereinbarung
Datei 8	ops_ko	a	6	100	OPS-Kode Prozedurenschlüssel in der gültigen OPS-Version analog zur § 301-Vereinbarung 2005-2007 Schlüssel für unbekannt = 999999 ab 2008 Schlüssel für unbekannt = AAAAAA
Algor.	typ_op	a	1	1	Operation lt. Kap. 5  1 = Ja / 2 = Nein

Quelle	Variable	Format	Länge	Wdh.	Inhalt/Bemerkungen
Algor.	z_bel_oper	n	2	1	Anzahl Belegoperateure
Algor.	z_bel_an	n	2	1	Anzahl Beleganästhesisten
Algor.	z_bel_heb	n	2	1	Anzahl Beleghebammen
Algor.	fab_max	a	4	1	Fachabteilung mit längster Verweildauer
Algor.	tage_max	n	5	1	Längste Verweildauer  (Bei mehreren Fachabteilungen: Tatsächlich verbrachte Tage in der fab_max)
Datei 3	fab	a	4	100	Fachabteilungen
Algor.	tage_fa	n	5	100	Verweildauer Fachabteilung
Algor.	typ_abt	a	1	1	Abteilungstyp  1 = nur Hauptabteilung 2 = nur Belegabteilung 3 = nur besondere Einrichtung 4 = mehrere verschiedene Belegungen
Datei 3	abt_art	a	2	100	Abteilungsart
Algor.	drgh	a	4	1	DRG-Kode (Leit-DRG-Kode vom InEK gruppiert)
Algor.	partition	a	1	1	DRG-Partition (M, O, A)
Algor.	drg	a	4	30	DRG-Kode
Datei 2	ik	a	9	1	Institutionskennzeichen
Datei 2	entl_ort	a	2	1	Entlassender Standort
Datei 2	fall_nr	a	8	1	Anonymisierte Fallnummer
Algor.	patient	n	3	1	Zählfeld (konstant 1)
Algor.	res	A	10	1	Reservefeld
Algor.	BwrlVol	n	8	1	DRG-Erlös (ab Berichtsjahr 2007-2009)  (Für eine näherungsweise Hochrechnung des Erlösvolumens ausschließlich in Hauptabteilungen; berechnet über die DRG-Bewertungsrelation lt. Fallpauschalenkatalog multipliziert mit Landesbasisfallwert. Zu- und Abschläge sowie Zusatzentgelte sind nicht berücksichtigt.)
Algor.	cm_vol	n	8	1	Case Mix Erlösvolumen in EURO (ab BJ 2010)  Das bewertete Erlösvolumen wird ermittelt aus dem Produkt der effektiven Bewertungsrelation und dem jeweiligen Landesbasisfallwert (mit Angleichungsbetrag) der behandelten Krankenhausfälle. Zusatzentgelte und nicht mit dem Fallpauschalenkatalog vergütete vollstationäre Leistungen sind in der Berechnung nicht eingeschlossen.
Algor.	cm_n	n	8	1	Gültige Fälle (Zähler) für den Case Mix (ab BJ 2010) Ohne Fälle der Integrierten Versorgung (3. und 4. Stelle jeweils mit Kodierung 41 – 47 vom Aufnahmegrund)
Algor.	cm	n	8	1	Case Mix (CM) (ab BJ 2010) Der Case Mix (CM), für „ökonomische Fallmischung“, entspricht dem gesamten bewerteten Ressourceneinsatz innerhalb des DRG-Entgeltsystems und beschreibt das Leistungsgeschehen z. B. einer Fachabteilung, eines Krankenhauses oder einer Region. Der Case Mix ergibt sich aus der Addition der effektiven Bewertungsrelationen aller behandelten Krankenhausfälle im jeweiligen Berichtsjahr. In die Berechnung ist die effektive Bewertungsrelation der DRG-Fallpauschale des Behandlungsfalls einbezogen. Berücksichtigt werden dabei tagesbezogene Abschläge bei Unterschreitung der unteren Grenzverweildauer und Zuschläge bei Überschreitung der oberen Grenzverweildauer sowie Verlegungen nach den Regelungen der Fallpauschalenverordnung. Zusatzent-

					gelte sowie nicht mit dem Fallpauschalenkatalog bewertete und vergütete vollstationäre Leistungen sind in der Berechnung nicht eingeschlossen.
Datei 4	beatm	n	3	1	Beatmungszeit in Stunden (ab 2005)
Algor.	split	a	4	1	Aufteilung einer Basis-DRG nach Schweregrad bzw. Ressourcenverbrauch, 4. Stelle der DRG-Notation (ab 2005)
Algor.	auf_monat	n	3	1	Aufnahmemonat (ab BJ 2009)

Rot markierte Merkmale sind für Auswertungen und zur Veröffentlichung nicht freigegeben  
 Grün markierte Merkmale sind nur als Identifikator verwendbar

## **Bisher sind in der Reihe folgende FDZ-Arbeitspapiere erschienen:**

*Arbeitspapier Nr. 43:* Average wage, qualification of the workforce and export performance in German enterprises: Evidence from KombiFiD data, J. Wagner, Februar 2012

*Arbeitspapier Nr. 42:* The Quality of the KombiFiD-Sample of Enterprises from Manufacturing Industries: Evidence from a Replication Study, J. Wagner, Dezember 2011

*Arbeitspapier Nr. 41:* How to define an enterprise and assign trade declarations to the right one: Exploration of German traders' micro transaction data, C. Stirböck, August 2011

*Arbeitspapier Nr. 40:* Definition von nutzerseitigen Kriterien für Datenstrukturfiler, J. Höninger/ M. Rosemann/ R. Voshage, Juli 2011

*Arbeitspapier Nr. 39:* Improvement of data access. The long way to remote data access in Germany, M. Brandt/ M. Zwick, Juni 2011

*Arbeitspapier Nr. 38:* Decentralised Access to Confidential Microdata in Europe, M. Brandt/ M. Zwick, Juni 2011

*Arbeitspapier Nr. 37:* Masking Micro Data with Stochastic Noise, J. Höhne/ J. Höninger, Mai 2011

*Arbeitspapier Nr. 36:* Enthüllungsrisiko beim Remote Access: Die Schwerpunkteigenschaft der Regressionsgerade, A. Vogel, April 2011

*Arbeitspapier Nr. 35:* Temporary agency work and firm performance, S. Nielen/ A. Schiersch, April 2011

*Arbeitspapier Nr. 34:* Harmonisation of statistical confidentiality in the Federal Republic of Germany, M. Brandt/ A. Crößmann/ C. Gürke, März 2011

*Arbeitspapier Nr. 33:* Remote Access. Eine Welt ohne Mikrodaten ??., G. Ronning/ P. Bleninger/ J. Drechsler/ C. Gürke, Januar 2011

*Arbeitspapier Nr. 32:* Compiling a Harmonized Database from Germany's 1978 to 2003 Sample Surveys of Income and Expenditure., T. Bönke/ C. Schröder/ C. Werdt, Mai 2010

*Arbeitspapier Nr. 31:* The Research Potential of New Types of Enterprise Data based on Surveys from Official Statistics in Germany., J. Wagner, Oktober 2009

*Arbeitspapier Nr. 30:* Geschlechterspezifische Einkommensunterschiede bei Selbstständigen im Vergleich zu abhängig Beschäftigten - Ein empirischer Vergleich auf der Grundlage steuerstatistischer Mikrodaten, P. Eilsberger/ M. Zwick, Januar 2008

*Arbeitspapier Nr. 29:* Reichtum in Niedersachsen und anderen Bundesländern - Ergebnisse der Steuergeschäftsstatistik 2003 für Selbstständige (Freie Berufe und Unternehmer) und abhängig Beschäftigte, P. Böhm/ J. Merz, November 2008

*Arbeitspapier Nr. 28:* Exports and Productivity in the German Business Services Sector. First Evidence from the Turnover Tax Statistics Panel, A. Vogel, Juli 2009

*Arbeitspapier Nr. 27:* Künstler in den Daten der amtlichen Statistik, C. Haak, August 2008

*Arbeitspapier Nr. 26:* Union Density and Varieties of Coverage: The Anatomy of Union Wage Effects in Germany, B. Fitzenberger/ K. Kohn/ A. C. Lembcke, August 2008

*Arbeitspapier Nr. 25:* German engineering firms during the 1990's. How efficient are export champions?, A. Schiersch, Juli 2008

*Arbeitspapier Nr. 24:* Zum Einkommensreichtum Älterer in Deutschland – Neue Reichtumskennzahlen und Ergebnisse aus der Lohn- und Einkommensteuerstatistik (FAST 2001), P. Böhm/ J. Merz, Februar 2008

*Arbeitspapier Nr. 23:* Neue Datenangebote in den Forschungsdatenzentren. Betriebs- und Unternehmensdaten im Längsschnitt, M. Brandt/ D. Oberschachtsiek/ R. Pohl, November 2007

*Arbeitspapier Nr. 22:* Stichprobendaten von Versicherten der gesetzlichen Krankenversicherung- Grundlage und Struktur des Datenmaterials, P. Lugert, Dezember 2007

*Arbeitspapier Nr. 21:* KombiFiD - Kombinierte Firmendaten für Deutschland, S. Bender/ J. Wagner/ M. Zwick, November 2007

*Arbeitspapier Nr. 20:* Neue Möglichkeiten zur Nutzung vertraulicher amtlicher Personen- und Firmendaten, U. Kaiser/ J. Wagner, Juni 2007

*Arbeitspapier Nr. 18:* Die Gehalts- und Lohnstrukturerhebung: Methodik, Datenzugang und Forschungspotential, H.-P. Hafner/ R. Lenz, Mai 2007

*Arbeitspapier Nr. 17:* Anonymisation of Linked Employer Employee Datasets. Theoretical Thoughts and an Application to the German Structure of Earnings Survey, H.-P. Hafner/ R. Lenz, Dezember 2006

*Arbeitspapier Nr. 16:* Die europäische Union - Integration von unten oder Eliteprojekt? Eine Sekundäranalyse von Mikrodaten der amtlichen Statistik, R. Nauenburg, November 2006

*Arbeitspapier Nr. 15:* Keeping in Touch - A Benefit of Public Holidays Using German Time Use diary Data, J. Merz/ L. Osberg, November 2006

*Arbeitspapier Nr. 14:* Zur Konzeption eines Taxpayer-Panels für Deutschland, D. Vorgrimler/C. Gräßl/ S. Kriete-Dodds, November 2006

*Arbeitspapier Nr. 13:* Anonymisierte Daten der amtlichen Steuerstatistik, D. Vorgrimler, September 2006

*Arbeitspapier Nr. 12:* Mikrosimulation in der Betriebswirtschaftlichen Steuerlehre, R. Maiterth, August 2006

*Arbeitspapier Nr. 11:* Der Anteil der freien Berufe und der Gewerbetreibenden an der Gemeindefinanzierung, M. Zwick, September 2006

*Arbeitspapier Nr. 10:* Konstruktion und Bewertung eines ökonomischen Einkommens aus der faktisch anonymisierten Lohn- und Einkommensteuerstatistik, T. Bönke/F. Neher/C. Schröder, August 2006

*Arbeitspapier Nr. 9:* Anonymising business micro data - results of a German project, R. Lenz/ M. Rosemann/D. Vorgrimler/R. Sturm, Juni 2006

*Arbeitspapier Nr. 8:* Scientific analyses using the Continuing Vocational Training Survey 2000, R. Lenz/ H.-P. Hafner/ D. Schmidt, Juni 2006

*Arbeitspapier Nr. 7:* A standard for the release of microdata, R. Lenz/ D. Vorgrimler/ M. Scheffler, Juni 2006

*Arbeitspapier Nr. 6:* Measuring the disclosure protection of micro aggregated business microdata, R. Lenz, Juni 2006

*Arbeitspapier Nr. 5:* De facto anonymised microdata file on income tax statistics 1998, J. Merz/ D. Vorgrimler/ M. Zwick, Oktober 2005

*Arbeitspapier Nr. 4:* Matching German turnover tax statistics, R. Lenz/ D. Vorgrimler, Juni 2005

*Arbeitspapier Nr. 3:* The research data centres of the Federal Statistical Office and the statistical offices of the Länder, S. Zühlke/ M. Zwick/ S. Scharnhorst/ T. Wende, März 2005

*Arbeitspapier Nr. 2:* Eine kommunale Einkommen- und Körperschaftsteuer als Alternative zur deutschen Gewerbesteuer: Eine empirische Analyse für ausgewählte Gemeinden, R. Maiterth/ M. Zwick, April 2005

*Arbeitspapier Nr. 1:* Ein Vergleich der Ergebnisse von Mikrosimulationen mit denen von Gruppensimulationen auf Basis der Einkommensteuerstatistik, H. Müller, März 2005

Stand: 14.03.2012

