Analysen mit der DRG-Statistik – Herausforderungen und Lösungsansätze

Tim Hochgürtel
Statistisches Bundesamt
Gustav-Stresemann-Ring 11
65189 Wiesbaden
tim.hochguertel@destatis.de

Tobias Lösch
Statistisches Bundesamt
Gustav-Stresemann-Ring 11
65189 Wiesbaden
tobias.loesch@destatis.de

Zusammenfassung

Über die kontrollierte Datenfernverarbeitung, die von den Forschungsdatenzentren der statistischen Ämter des Bundes und der Länder (FDZ) angeboten werden, können wissenschaftliche Datennutzer unter anderem mit der DRG-Statistik arbeiten.
Aufgrund von Umfang und Struktur der DRG-Statistik muss hierbei von langen Laufzeiten der Analyseprogramme ausgegangen werden. Der vorliegende Beitrag untersucht, wie sich die Laufzeiten zur Identifikation von Subpopulationen minimieren lassen, die sich über bestimmte ICD-Nebendiagnosen oder OPS-Kodes abgrenzen lassen. Hierzu werden die Laufzeiten verschiedener Methoden verglichen. Daneben wird das SAS-Makro *newvar* vorgestellt, welches vom FDZ entwickelt wurde. Dieses Makro unterstützt die zügige Erstellung neuer Dummy- und Summen-Variablen, indem es in Abhängigkeit von nutzerspezifizierten Parametern unter verschiedenen Methoden die jeweils effizienteste auswählt.

Schlüsselwörter: DRG-Statistik, Forschungsdatenzentren, Laufzeitanalyse, effiziente Programmierung, Makro %NEWVAR

1 Analysen mit der DRG-Statistik

Die *Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder* (FDZ) ermöglichen wissenschaftlichen Datennutzern 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.
Im Rahmen der kontrollierten Datenfernverarbeitung haben Datennutzer die Möglichkeit Auswertungsprogramme an das FDZ zu übermitteln. Im FDZ werden die Mikrodaten der DRG-Statistik mit diesen Auswertungsprogrammen analysiert. Der Datennutzer erhält seine Ergebnisse nach einer Geheimhaltungsprüfung. Zu beachten ist hierbei, dass eine Analyse des vollständigen Datenmaterials der DRG-Statistik ausschließlich mit der Statistiksoftware SAS möglich ist [1] [4].

Die DRG-Statistik bildet die Behandlungsfälle aller Krankenhäuser ab, die nach § 21 Krankenhausentgeltgesetz (KHEntgG) zur Meldung verpflichtet sind. Dies beinhaltet alle Krankenhäuser, die nach dem DRG-Vergütungssystem abrechnen und dem Anwen-

dungsbereich des § 1 KHEntgG unterliegen. Unter anderem werden auch Behandlungsfälle von Krankenhäusern der Bundeswehr erfasst, sofern es sich um die Behandlung von Zivilpersonen handelt. Krankenhäuser der Berufsgenossenschaften tragen ebenfalls 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 [3]. Behandlungsfälle von psychiatrischen und psychosomatischen Einrichtungen nach § 17b Abs. 1 Satz 1 zweiter Halbsatz des Krankenhausfinanzierungsgesetzes (KHG) können für Analysen ebenfalls nicht genutzt werden.

Mit der DRG-Statistik stehen dem Datennutzer somit alle vollstationären Krankenhausbehandlungen im DRG-Entgeltbereich in Deutschland für Analysen zur Verfügung. Dabei beinhaltet die Statistik mehrere hundert Variablen für bis zu 17,7 Millionen Behandlungsfälle pro Berichtsjahr. Ein Überblick über die Berichtsjahre, die zur Datennutzung zur Verfügung stehen, sowie eine Beschreibung der Variablen der DRG-Statstik können über die Web-Seite der FDZ abgerufen werden[1].

Die Datennutzer der kontrollierten Datenfernverarbeitung identifizieren im Rahmen von Analysen häufig besondere Teilpopulationen unter den Behandlungsfällen, welche über Diagnoseschlüssel (ICD-Kodes) oder Prozedurenschlüssel (OPS-Kodes) abgegrenzt werden. So werden beispielsweise Patientengruppen identifiziert, bei denen bestimmte Erkrankungen diagnostiziert wurden. Der Datennutzer benennt hierzu die entsprechenden ICD-Nebendiagnosen, nach denen die Abgrenzung vorgenommen wird. Ein Behandlungsfall, der eine vom Datennutzer angegebene ICD-Nebendiagnose aufweist, gehört zur Teilpopulation. Die Menge der ICD-Nebendiagnosen oder OPS-Kodes, welche der Datennutzer zur Identifikation der Teilpopulation verwendet, wird im Folgenden als Abgleichsliste bezeichnet.

Die DRG-Statistik umfasst - neben einer Variablen für die Hauptdiagnose - 89 Variablen mit ICD-Nebendiagnosen sowie 100 Variablen zu den OPS-Kodes. Für die Identifikation einer Teilpopulation wird von einem Datennutzer in der Regel eine Dummy-Variable erstellt, welche den Wert 1 aufweist, sofern in den ICD-Nebendiagnosen oder OPS-Kodes eines Behandlungsfall die interessierenden Ausprägungen zu finden sind. Sonst nimmt die Dummy-Variable den Wert 0 an.
Für jeden Behandlungsfall ist in der DRG-Statistik ein Datensatz angelegt. In der Regel weist ein Behandlungsfall nur wenige ICD-Nebendiagnosen und OPS-Kodes auf. Daher sind nur wenige der 89 Variablen zu den ICD-Nebendiagnosen und der 100 Variablen zu den OPS-Kodes mit entsprechenden Werten belegt. Die meisten Variablen mit Bezug zu den ICD-Nebendiagnosen und OPS-Kodes sind mit Missings besetzt.

[1] http://www.forschungsdatenzentrum.de/bestand/drg/index.asp

Abbildung 1 zeigt die Datenstruktur der ICD-Nebendiagnosen an einem fiktiven Ausschnitt. Wenn einem Behandlungsfall *n* ICD-Nebendiagnosen zugeordnet sind, so weisen die Variablen icd_nd1 bis icd_nd*n* diese Nebendiagnosen aus. Die Variablen icd_nd*n+1* bis icd_nd89 sind dann für den entsprechenden Behandlungsfall mit Missings besetzt. In der Regel weisen die Behandlungsfälle einstellige Anzahlen von Nebendiagnosen aus.

	icd_nd1	icd_nd2	icd_nd3	icd_nd4	icd_nd5	icd_nd6	icd_nd7	icd_nd8	icd_
1	A1234								
2	A1234	A1234	A1234	A1234					
3									
4									
5	A1234								
6	A1234	A1234							
7	A1234								
8									
9									
10									
11									
12	A1234	A1234	A1234						
13	A1234	A1234							
14	A1234								
15	A1234	A1234	A1234						
16	A1234								
17	A1234	A1234							
18	A1234								
19									
20	A1234								
21	A1234	A1234	A1234	A1234	A1234	A1234	A1234		
22	A1234								
23	A1234								
24	A1234	A1234	A1234						
25	A1234	A1234							
26									
27	A1234								
28	A1234								
29	A1234	A1234	A1234	A1234					
30	A1234								
31									
32	A1234								
33	A1234	A1234							
34									
35	A1234	A1234	A1234						
36	A1234	A1234							
37	A1234								
38									

Abbildung 1: Fiktiver Ausschnitt aus dem Mikrodatenfile der DRG-Statistik

Die OPS-Kodes weisen eine identische Datenstruktur auf. Auch hier sind die Variablen ops_ko1 bis ops_ko*n* mit den *n* OPS-Kodes eines Behandlungsfalls besetzt. Die Variablen ops_ko*n+1* bis ops_ko100 weisen für den entsprechenden Behandlungsfall ebenfalls Missings aus. Es trifft auch für die OPS-Kodes zu, dass deren Anzahl für einen Behandlungsfall in der Regel einstellig ist.
Da sich die Analyse der DRG-Statistik durch lange Laufzeiten von bis zu mehreren Wochen auszeichnet, ist eine effiziente Programmierung bei dieser Statistik besonders bedeutsam und stellt hohe Anforderungen an den Datennutzer. Die Laufzeit, die benötigt wird, um eine Dummy-Variable zur Identifikation von Teilpopulationen zu erstellen, variiert je nach gewählter Methode der Dummy-Erstellung, der Anzahl der Behandlungsfälle im DRG-Mikrodatenfile und der Anzahl der ICD-Nebendiagnosen bzw.

OPS-Kodes, nach denen die betreffende Teilpopulation abgegrenzt wird. Diese Anzahl der ICD-Nebendiagnosen bzw. OPS-Kodes entspricht der Anzahl der Elemente in der Abgleichsliste.

Um den Datennutzern der DRG-Statistik ein Werkzeug zur Verfügung zu stellen, welches eine möglichst effiziente Erstellung von Variablen unterstützt, wurde das Makro *newvar* entwickelt. Als Effizienzkriterium wird hierbei die benötigte Laufzeit verwendet. Neben der möglichst hohen Effizienz steht ebenso die Anwenderfreundlichkeit im Vordergrund, die auf eine möglichst einfache und intuitive Anwendung des Makros *newvar* durch die Datennutzer abzielt [2]. Das Makro *newvar* wählt in Abhängigkeit von nutzerspezifizierten Parametern die nach den bisherigen Erkenntnissen effizienteste Methode zur Erstellung neuer Dummy-Variablen.

Daneben erlaubt das Makro auch die Erstellung von Variablen, in denen ausgezählt wird, wie häufig die Elemente der Abgleichsliste je Behandlungsfall gegeben sind. Diese Variablen werden als Summe der Übereinstimmungen zwischen den ICD-Nebendiagnosen bzw. OPS-Kodes eines Behandlungsfalls und den Elementen der Abgleichsliste („Treffer“) gebildet und im Folgenden als Summen-Variablen bezeichnet.

2 Berücksichtigte Methoden zur Entwicklung des Makros *newvar*

Für die Erstellung von Dummy- und Summen-Variablen können in SAS verschiedene Methoden genutzt werden. Für die Entwicklung des Makros *newvar* sind verschiedene Vorgehensweisen hinsichtlich ihrer Laufzeit getestet worden.

Bei der Erstellung von neuen Variablen wird versucht den Umstand auszunutzen, dass die Variablen zu den ICD-Nebendiagnosen sowie den OPS-Kodes zum größten Teil mit Missings besetzt sind. Eine Zelle, die keinen ICD-Kode oder OPS-Kode enthält, kann keinen „Treffer“ generieren, wenn diese Zelle mit der Abgleichsliste verglichen wird.

Zur Identifikation einer geeigneten Methode zur effizienten Erstellung neuer Dummy- und Summen-Variablen werden folgende Methoden berücksichtigt:

- Array-Methode: Nutzung einer ARRAY-Anweisung
- SQL-Methode: Nutzung der Prozedur TRANSPOSE in Kombination mit der Prozedur SQL
- Summary-Methode: Nutzung der Prozedur TRANSPOSE in Kombination mit der Prozedur SUMMARY
- IML-Methode: Nutzung der Prozedur IML

Es folgt eine kurze Erläuterung der Umsetzung der verschiedenen Methoden.

2.1 Array-Methode: Nutzung einer ARRAY-Anweisung

Mit einer ARRAY-Anweisung kann in SAS eine Reihe von Variablen indiziert werden. Die indizierten Variablen können anschließend in einer Schleife verarbeitet werden. Bei dem angestrebten Abgleich führt dies dazu, dass jede Zelle der indizierten Variablen mit jedem Element der Abgleichsliste abgeglichen wird. Im Falle der Erstellung einer Dummy-Variablen weist die neue Variable den Wert 1 auf, wenn mindestens eine Übereinstimmung zwischen den Ausprägungen der ICD-Nebendiagnosen oder OPS-Kodes eines Datensatzes und einem Element der Abgleichsliste vorliegt. Sonst ist die Dummy-Variable 0. Im Falle der Erstellung einer Summen-Variable weist die neue Variable die Anzahl der Übereinstimmungen zwischen den ICD-Kodes bzw. OPS- Kodes eines Datensatzes und der Abgleichsliste auf.

Ein Vorteil dieser Methode ist dadurch gegeben, das keine weiteren Mikrodatendateien oder Matrizen angelegt werden müssen. Demgegenüber steht der Nachteil, dass auch all jene Zellen mit der Abgleichsliste verglichen werden, die mit Missings besetzt sind. Da in der DRG-Statistik ein großer Anteil der Zellen zur Erfassung von ICD-Nebendiagnosen und OPS-Kodes Missings aufweisen, werden im Rahmen dieser Methode eine Vielzahl von Abgleichen durchgeführt, die keine Übereinstimmung liefern können.

2.2 SQL-Methode: Nutzung der Prozedur TRANSPOSE in Kombination mit der Prozedur SQL

Bei Verwendung einer Kombination der Prozeduren TRANSPOSE und SQL kann der hohe Anteil vom Missings genutzt werden, um die Anzahl der Abgleiche im Vergleich zur Array-basierten Methode deutlich zu reduzieren. Hierfür wird zunächst mit der Prozedur TRANSPOSE ein neues Datenfile erzeugt, in welchem für jede Zelle der Analysevariablen (ICD- oder OPS-Kodes) des Ausgangsfiles (DRG-Mikrodatenfile) ein Datensatz angelegt wird. Daneben enthält das transponierte File eine Fall-Nummer, welche eine eindeutige Zuordnung jedes Datensatzes des neuen transponierten Files zu einem Datensatz des Ausgangsfiles ermöglicht. Im zweiten Schritt werden alle Datensätze des transponierten Files gelöscht, die hinsichtlich der Analysevariable ein Missing aufweisen. Da der Anteil der Missings bezüglich der Analysevariablen sehr hoch ist, reduziert sich die Anzahl der Datensätze im transponierten Datenfile an dieser Stelle erheblich. Anschließend wird für die verbleibenden Datensätze des transponierten Files die Analysevariable mit den Elementen der Abgleichsliste verglichen. Führt ein Abgleich zwischen Analysevariable und Abgleichsliste zu einer Übereinstimmung, so nimmt eine Hilfsvariable den Wert 1 an, sonst ist die Hilfsvariable 0.
Im transponierten Datenfile liegen für Datensätze des Ausgangsfiles (DRG-Mikrodatenfile) gegebenenfalls mehrere Datensätze vor, die je nach dem Ergebnis des Abgleichs eine 1 oder eine 0 in der Hilfsvariable aufweisen. Für die Erstellung der neuen Dummy- oder Summen-Variable muss eine Aggregation des transponierten Files durchgeführt werden.

Mittels der Prozedur SQL wird ein Aggregatsfile aus dem transponierten File erstellt. Hierbei wird eine neue Variable im Aggregatsfile erzeugt, die bei der Erstellung einer Dummy-Variable den Wert 1 annimmt, wenn die Datensätze eine Fall-Nummer im transponierten File mindestens eine Ausprägung 1 hinsichtlich der Hilfsvariable aufweisen. Soll eine Summen-Variable erzeugt werden, so erstellt die Prozedur SQL die Summe der Hilfsvariable über alle Datensätze einer Fall-Nummer im transponierten File.

Im Anschluss muss die neue Dummy- oder Summen-Variable des Aggregatsfile an das DRG-Mikrodatenfile angespielt werden. Hierbei wird ein Matching anhand der spezifischen Fall-Nummer durchgeführt. Nachdem die Dummy- oder Summen-Variable dem DRG-Mikrodatenfiles zugespielt ist, wird für alle Datensätze, die noch keine Ausprägung hinsichtlich dieser Variable aufweisen, der Wert 0 zugewiesen.

2.3 Summary-Methode: Nutzung der Prozedur TRANSPOSE in Kombination mit der Prozedur SUMMARY

Eine weitere Methode zur Erstellung neuer Variablen nutzt die Kombination der Prozeduren TRANSPOSE und SUMMARY. Diese Methode funktioniert ähnlich wie die SQL-Methode. Im Kontext der Summary-Methode wird in Analogie zur SQL-Methode ein transponiertes Datenfile der Analysevariablen erzeugt, die Datensätze mit Missings in der Analysevariable gelöscht sowie eine Dummy-Hilfsvariable erzeugt, die genau dann den Wert 1 annimmt, sofern eine Übereinstimmung mit den Einträgen der Abgleichsliste gegeben ist.

Die Aggregation wird mit der Prozedur SUMMARY durchgeführt. Es wird für jede Fall-Nummer des transponierten Files hinsichtlich der Hilfsvariable das Maximum ermittelt, um eine Dummy-Variable im Aggregatsfile zu erstellen. Ist eine Summen-Variable zu erstellen, so wird die Summe der Hilfsvariable für jede Fall-Nummer berechnet und in die neue Variable des Aggregatsfile geschrieben.

In Analogie zur SQL-Methode wird die neue Dummy- oder Summen-Variable des Aggreationsfile mit den DRG-Mikrodaten gematcht. Datensätze des DRG-Mikrodatenfiles, welche nach dem Matching keinen Eintrag hinsichtlich der neuen Variable aufweisen, wird eine 0 zugewiesen.

2.4 IML-Methode: Nutzung der Prozedur IML

Mit Hilfe der Prozedur IML kann der Vergleich der Einträge in der Abgleichsliste mit den Werten der Variablen der ICD-Kodes bzw. OPS-Kodes zeilenweise (d.h. je Datensatz) erfolgen.

Hierzu werden in der IML-Umgebung alle Variablen der ICD-Nebendiagnosen bzw. des OPS-Kodes des DRG-Mikrodatenfiles in eine Matrix geschrieben. Auf Basis dieser

Matrix ist ein zeilenweiser Abgleich von Einträgen der Abgleichsliste mit den Zellen der Matrix möglich.

Zur Erstellung einer Dummy-Variable werden alle Zellen einer Zeile mit der Abgleichsliste verglichen, bis der erste "Treffer" vorliegt, oder das erste Missing gefunden wird (nach einem Missing folgen ausschließlich weitere Missings je Zeile). Sofern ein "Treffer" vorliegt, nimmt die Dummy-Variable für diese Zeile den Wert 1 an. Wird hingegen bis zum Auffinden des ersten Missings kein "Treffer" generiert, so ist der Dummy 0.

Zur Erstellung einer Summen-Variable werden alle Zellen einer Zeile bis zum Auftreten des ersten Missings mit der Abgleichsliste verglichen. Die Summen-Variable enthält danach die Anzahl der Übereinstimmungen.

Die Spalte der Matrix, welche die neue Dummy- bzw. Summen-Variable beinhaltet, wird danach den DRG-Mikrodaten zugespielt.

3 Messung der Laufzeiten der verschiedenen Methoden

Die Laufzeiten der verschiedenen Methoden zur Erstellung neuer Variablen wurden empirisch ermittelt. Für jede Methode wurden Laufzeiten für die folgenden Kombinationen aus Anzahl der Datensätze im Mikrodatenfile und Anzahl der Elemente in der Abgleichsliste gemessen:

(a) a*100.000 Datensätze im Mikrodatenfile kombiniert mit b Elementen in der Abgleichsliste, wobei $1 \le a \le 30, 1 \le b \le 47 \; und \; a, b \in \mathrm{N}$.
sowie
(b) c*1.000.000 Datensätze im Mikrodatenfile kombiniert mit d*100 Elementen in der Abgleichsliste, wobei $4 \le c \le 15, 1 \le d \le 10 \; und \; c, d \in \mathrm{N}$.

Im Rahmen der Messung der Laufzeiten werden Dummy-Variablen erstellt. Da die IML-Methode bereits im ersten Simulationsschritt (a) zu vergleichsweise schlechten Ergebnissen führte, wurde auf eine weitere Simulation der Abgleiche (b) verzichtet. Ein Grund für das schlechte Abschneiden der IML-Methode könnte der große Umfang der DRG-Mikrodatenfiles sein. Die Daten können dadurch nicht optimal im Arbeitsspeicher gehalten werden, was jedoch eine Voraussetzung für schnelle Berechnungen in der IML Umgebung ist.
Die SAS-Systemoption FULLSTIMER wird dafür genutzt, verschiedene Informationen zur Programmdurchführung im Log-File zu dokumentieren. Hierbei werden auch drei verschiedene Laufzeiten ausgewiesen. Die angegebene „Realtime“, welche der Zeit zwischen Starten und Beenden eines Programms entspricht, ist z.B. stark von Lese- und Schreibvorgängen des Festplattensystems beeinflusst und eignet sich daher wenig für eine Laufzeitmessung der verschiedenen Methoden. Die „System CPU Time“ gibt die Zeit an, die SAS für systemabhängige Rechenschritte oder –pausen benötigt und wird in

der Regel nicht maßgeblich von der Methode beeinflusst. Die „User CPU Time“ hingegen beschreibt die Zeit, die SAS für die reine Ausführung der Rechenschritte des Codes (hier: zur Ausführung der Methode) benötigt.

Der Vergleich der Laufzeiten der verschiedenen Methoden zur Ermittlung der effizientesten Methode basiert daher lediglich auf der „User CPU Time“.

Für jede Messung einer Laufzeit wird die „User CPU Time“ des Programms in eine separate Log-Datei geschrieben, anschließend mit Hilfe der SCAN-Funktion ausgelesen und zusammen mit weiteren Parametern (Methode, Anzahl der Datensätze, Anzahl der Elemente in der Abgleichsliste) in einen Datensatz eines dafür angelegten Datenfiles geschrieben. Dieses Datenfile bildet die Grundlage der Auswertung der Laufzeiten.

4 Ergebnisse

Die Messung der Laufzeiten ermittelt für die vier verschiedenen Methoden, verschiedenen Kombinationen von Datensätzen im Mikrodatenfile (Beobachtungen) und unterschiedlicher Anzahl von Elementen in der Abgleichsliste (Abgleichslisteneinträge) die benötigte Laufzeit (Messung der "User CPU Time" in Sekunden für die IT-Infrastruktur des FDZ). Basierend auf diesen Ergebnissen wird die Laufzeit von allen vier Methoden mit folgendem Regressionsmodell geschätzt.

$$\widehat{\text{User-CPU-Time}} = \beta_1 * \frac{\text{Beobachtungen}}{10.000} + \beta_2 * \frac{\text{Abgleichslisteneinträge}}{100} + \beta_3 * \frac{\text{Beobachtungen}}{10.000} * \frac{\text{Abgleichslisteneinträge}}{100}$$

In der Regressionsgleichung stellt die User-CPU-Time die abhängige Variable dar, welche aus den unabhängigen Variablen geschätzt wird. Als unabhängige Variablen werden die normierte Anzahl der Datensätze im Mikrodatenfile und die normierte Anzahl der Elemente in der Abgleichsliste verwendet. Daneben wird ein Interaktionsterm zwischen den beiden unabhängigen Variablen verwendet.

Tabelle 1 weist die Beta-Koeffizienten sowie das korrigierte R^2 für die jeweiligen Regressionsmodelle aus.

Tabelle 1: Beta-Koeffizienten OLS-Regression

	Array-Methode	SQL-Methode	Summary-Methode	IML-Methode
Beobachtungen	1,24	5,21	5,47	-2,41
Abgleichslisteneinträge	-6,12	-0,37	0,63	4,09
Beobachtungen* Abgleichslisteneinträge	11,89	0,36	0,34	3618,09
korr. R^2	0,9950	0,9995	0,9999	1,0000

Alle vier Modelle liefern ein korrigiertes R^2 von über 99 %. Die gewählten Regressionsmodelle sind daher sehr gut geeignet, Prognosen über die benötigten Laufzeiten zu erstellen. Anhand der Regressionsgleichung können die Laufzeiten für eine bestimmte Methode in Abhängigkeit von der Anzahl der Datensätze im Mikrodatenfile und der Anzahl der Elemente in der Abgleichsliste somit näherungsweise vorausgesagt werden.

Es lässt sich daher in Abhängigkeit der beiden unabhängigen Variablen die jeweils effizienteste der vier Methoden ermitteln. Für eine Kombination aus beiden unabhängigen Variablen kann die benötige Laufzeit aller vier Methoden geschätzt werden. Die Methode, für welche die geringste Laufzeit prognostiziert wird, kann für die Erstellung der neuen Variable verwendet werden.

Das Makro *newvar* macht sich dies zu Nutze. In Abhängigkeit von der Anzahl der Datensätze im DRG-Mikrodatenfile und der Anzahl der Elemente in der Abgleichsliste kann die effektivste Methode zur Erstellung der neuen Variable ermittelt werden.

Abbildung 2 visualisiert die effizienteste der vier Methoden in Abhängigkeit der Anzahl der Datensätze im Mikrodatenfile und der Anzahl der Einträge in der Abgleichsliste. Als Datenbasis dienen hierbei nicht die empirische Messung der CPU-Time, sondern die geschätzten Laufzeiten der Regressionsgleichungen. Hierbei wird durch unterschiedliche Grautöne dargestellt, für welche Methode die geringste Laufzeit prognostiziert wird.

Die Array-Methode erweist sich als effizienteste Methode, solange die Anzahl von 34 Einträgen in der Abgleichsliste nicht überschritten wird. Die Anzahl der Beobachtungen spielt hierbei eine untergeordnete Rolle. Nur für den Fall, dass der Dummy für eine Population von weniger als 70.000 Beobachtungen berechnet wird, ist die Array-Methode

den anderen Methoden vorzuziehen, unabhängig von der Anzahl der Elemente in der Abgleichsliste.

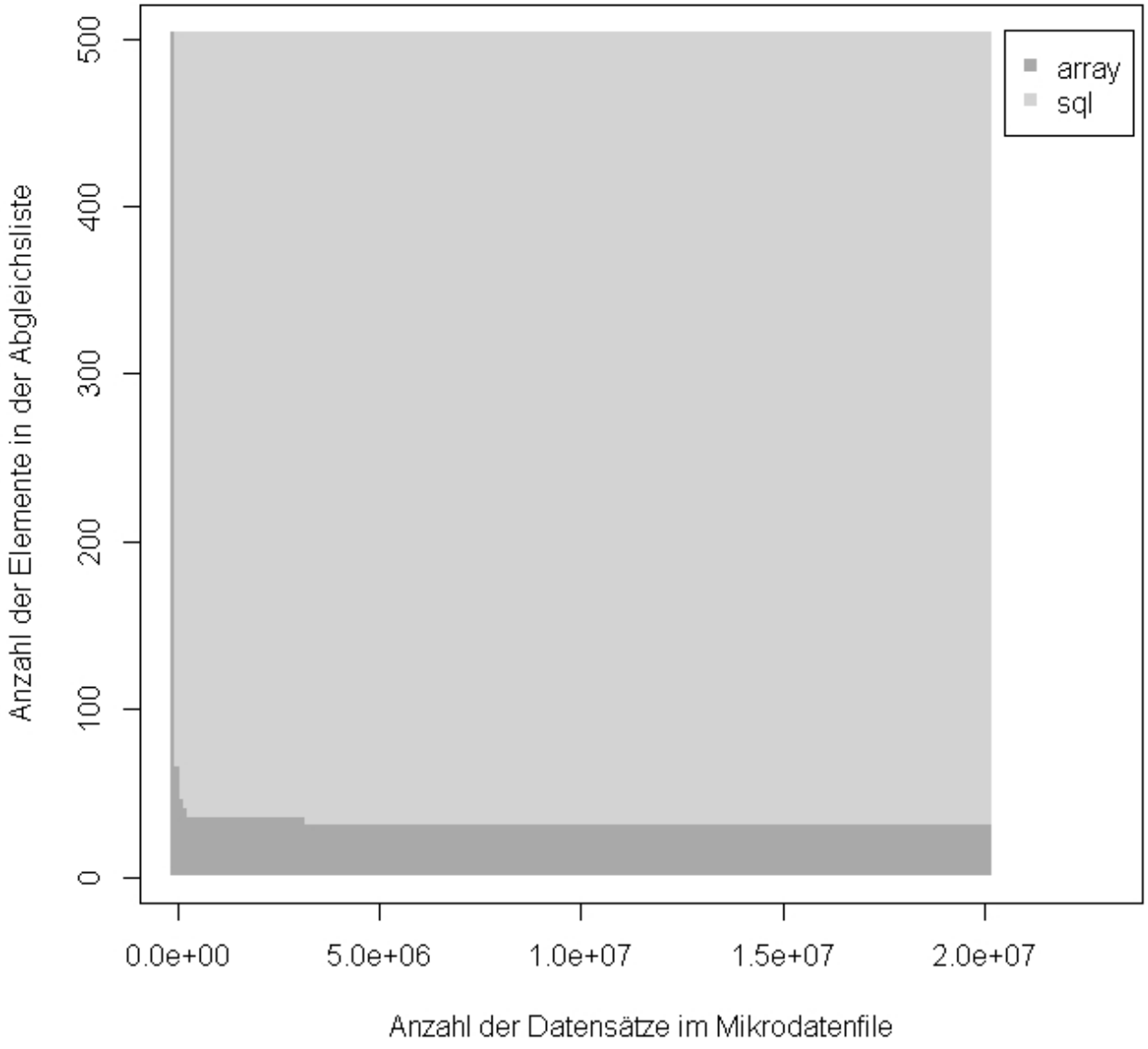

Abbildung 2: Effizienteste Methode in Abhängigkeit von Anzahl der Datensätze im Mikrodatenfile und Anzahl der Elemente in der Abgleichsliste

Wenn die Anzahl der Beobachtungen bei mehr als 70.000 liegt und die Anzahl der Elemente in der Abgleichsliste größer 34 ist, erweist sich die SQL-Methode als effizienter als die übrigen Methoden. Hierbei zeigt sich auch, dass die SQL-Methode nur geringfügig weniger Laufzeit benötigt als die Summary-Methode.

Die IML-Methode schneidet durchgehend am schlechtesten ab. Es lässt sich weder empirisch noch über die Regressionsschätzer eine Konstellation identifizieren, in welcher sich die IML-Methode als effizienteste erweist.

5 Das SAS-Makro *newvar*

Die hier vorgestellte Untersuchung hat das Ziel, die Laufzeiten der Analysen der DRG-Statistik im Rahmen der kontrollierten Datenfernverarbeitung in den FDZ zu reduzieren. Hierbei sollen die Nutzer die Möglichkeit erhalten, in ihren Analysen die Erstellung von Dummy-Variablen zur Abgrenzung von Subpopulationen mit möglichst geringer Laufzeit umsetzen zu können. Daneben soll die Erstellung von Summen-Variablen unterstützt werden.

Die Möglichkeit einer effizienten Erstellung solcher Dummy- und Summen-Variablen soll dabei nutzerfreundlich umgesetzt werden. Daher soll die Wahl der effizientesten der vier untersuchten Methoden für den Nutzer mit möglichst wenig Entscheidungslast verbunden sein.

Für diesen Zweck ist vom FDZ das SAS-Makro *newvar* entwickelt worden. Das Makro nutzt in Abhängigkeit von nutzerspezifizierten Parametern die jeweils effizienteste Methode zur Erstellung neuer Variablen. Die Entwicklung des Makros basiert auf den Erkenntnissen von Laufzeitanalysen.

Das SAS-Makro *newvar* verfügt über fünf Parameter, welche dem Nutzer eine individuelle Anpassung an die eigenen Anforderungen ermöglichen. Im Folgenden werden diese Parameter kurz beschrieben. Weitere Hinweise zum Makro *newvar* finden sich bei [2].

- *user_file*: Falls ein Nutzer das Makro außerhalb eines Data-Steps anwendet, muss mit dem Parameter *user_file* die SAS-Datendatei benannt werden, in welcher die neue Variable erstellt wird. Falls eine Anwendung des Makros innerhalb eines Data-Steps erfolgt, bleibt der Parameter leer.
- *user_liste*: Mit diesem Parameter gibt der Nutzer die Elemente an, welche mit den ICD-Nebendiagnosen bzw. OPS-Kodes der Behandlungsfälle abgeglichen werden. Dieser Parameter entspricht der Abgleichsliste.
- *user_block*: Der Parameter wird genutzt, um anzugeben, ob der Abgleich über die Nebendiagnosen oder OPS-Kodes durchgeführt wird.
- *user_funktion*: Der Nutzer kann die neue Variable als Dummy oder Summe erstellen. Die Dummy-Variable nimmt den Wert 1 an, falls ein Behandlungsfall in den Nebendiagnosen bzw. OPS-Kodes über mindestens eine Ausprägung verfügt, welche im Parameter *user_liste* benannt ist. Sonst ist die Dummy-Variable 0. Wenn als *user_funktion* "Summe" gewählt wird, enthält die neue Variable die Anzahl der Übereinstimmungen von *user_liste* und Nebendiagnosen bzw. OPS-Kodes.
- *user_name*: Mit dem Parameter *user_name* bestimmt der Nutzer den Namen der neu anzulegenden Variable.

Die folgende Code-Sequenz zeigt beispielhafte Anwendungen des Makros *newvar*.

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

/* Erstellung einer 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 einer Variable „diagnose“ */
%let user_file = ;
%let user_liste = '1610', '1611', '1630y', '16500', '8860x';
%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;
```

Das Makro wählt hierbei selbständig die effizienteste der implementierten Methoden. Der Nutzer des Makros muss lediglich die individuellen Parameter für seine Analyse bestimmen.

Literatur

[1] T. Hochgürtel: Improvement of data access. On the way to Remote Data Access in Germany, http://isi2011.congressplaner.eu/pdfs/950948.pfd, 2011

[2] T. Hochgürtel, T. Lösch: Das SAS-Makro *newvar*. Entwicklung und Anwendung eines Hilfsinstruments zur effizienten Erstellung neuer Variablen in der DRG-Statistik, FDZ-Arbeitspapier Nr. 44, http://www.forschungsdatenzentrum.de/publikationen/veroeffentlichungen/fdz_arbeitspapier-44.pdf, 2012.

[3] J. Spindler: Fallpauschalenbezogene Krankenhausstatistik. Diagnosen und Prozeduren der Krankenhauspatienten auf Basis der Daten nach § 21 Krankenhausentgeltgesetz, in: J. Klauber u.a.: Krankenhausreport 2013. Schwerpunkt Mengendynamik: mehr Menge, mehr Nutzen?, Schattauer, Stutttgart, S. 385-415, 2013.

[4] S. Zühlke u.a.: Die Forschungsdatenzentren der Statistischen Ämter des Bundes und der Länder, in: Wirtschaft und Statistik 10/2003, S. 906-911, 2003.