German Record Linkage Center
WORKING PAPER SERIES
NO. WP-GRLC-2022-04 | JULY 28, 2022

Vergleich von Record-Linkage Methoden anhand der Mikrosimulation eines bundesweiten Schülerregisters

Severin Vincent Weiand

Inhaltsverzeichnis

Zusammenfassung

Die Verknüpfung von Datensätzen bietet für viele wissenschaftliche Fachgebiete zahlreiche Analysemöglichkeiten. In den meisten Fällen können die Datensätze jedoch nicht anhand von eindeutigen Identifikatoren verknüpft werden, sondern müssen mithilfe von Quasi-Identifikatoren (QID) wie z. B. Name oder Geburtstag verknüpft werden. Aus Datenschutzgründen und aufgrund des Prinzips der Datensparsamkeit muss die Zahl der QIDs für die Nutzung im administrativen Kontext auf ein notwendiges Minimum beschränkt werden.

Die Notwendigkeit von QIDs für Record-Linkage kann anhand der zu erwartenden Verbesserung der Linkage-Qualität durch Hinzunahme der QID bestimmt werden. Dies ist das Ziel dieser Arbeit. Dazu werden insgesamt fünf Record-Linkage Methoden verwendet und verglichen. Von besonderer Relevanz ist dabei die Notwendigkeit des Geburtsortes als QID. Die Datengrundlage stellt eine Mikro-Simulation eines bundesweiten Schülerregisters dar.

Es wird gezeigt, dass der Geburtsort ein relevantes Merkmal für das Linkage darstellt. Fehlt der Geburtsort, so entsteht insbesondere für Migrantinnen, die während ihrer Bildungslaufbahn heiraten, ein Linkage-Bias. Die besten Ergebnisse liefern Multiple Matchkeys und probabilistisches Record-Linkage mit einem ECM-Algorithmus.

Abstract

Linking datasets offers numerous analysis possibilities for many scientific disciplines. In most cases, datasets cannot be linked using unique identifiers, but must be linked using quasi-identifiers (QID) such as name or birthday. For data protection reasons and because of the principle of data minimisation, the number of QIDs for use in an administrative context must be minimized.

The necessity of QIDs for record linkage can be determined by the expected improvement in linkage quality by inclusion of the QID. This is the aim of this paper. For this purpose, five record linkage methods are used and compared. Of particular relevance is the necessity of the QID place of birth. The data basis is a microsimulation of an educational register.

It is shown that the place of birth is a relevant characteristic for the linkage. If place of birth is missing, a linkage bias arises, especially for migrant women who marry during their educational career. Multiple matchkeys and probabilistic record linkage with an ECM algorithm provide the best linkage results.

Schlagwörter: Record-Linkage, Bildungsregister, Schülerregister, Mikro-Simulation, Datenqualität, record linkage, data quality, educational register, microsimulation

Danksagungen: Der Autor bedankt sich bei Rainer Schnell, Manfred Antoni, Philip Höcker, Mara Köster und Ruth Lange für die hilfreichen Anmerkungen und Korrekturvorschläge.

Einleitung

Die Verknüpfung von Datensätzen ist eine zunehmend über alle wissenschaftlichen Fachgebiete reichende, an Relevanz gewinnende Methode. Von der Verknüpfung von Finanzdaten in der Ökonomie über Patienten-Daten in der Medizin bis hin zu Survey-Daten in den Sozialwissenschaften bietet Record-Linkage zahlreiche Möglichkeiten, um an zuvor nicht verfügbare Informationen zu gelangen (Christen, Ranbaduge et al. 2020: 10 ff.; Schnell 2019: 320).

Unter Record-Linkage wird die Verknüpfung von einer oder mehreren Entitäten über mehrere Datensätze hinweg verstanden. In vielen Fällen handelt es sich dabei um Personen. Für die Verknüpfung von Personen besteht oft keine eindeutige Personenkennung, wie z. B. eine Sozialversicherungsnummer, mit der das Linkage durchgeführt werden kann – das Fehlen einer Kennung ist der in Deutschland übliche Fall. Stattdessen müssen Personen anhand einer Summe von Merkmalen identifiziert werden, die meist eine eindeutige Verknüpfung ermöglichen. Diese Merkmale werden üblicherweise als Quasi-Identifikatoren (QIDs) bezeichnet und umfassen personenbezogene Merkmale wie Name oder Geburtsdatum (Christen, Ranbaduge et al. 2020: 414).

Datensätze, die anhand ihrer QIDs verknüpft werden sollen, variieren stark in Größe und Qualität. Die Qualität der Daten bemisst sich an der Zahl der Fehler, die in den Datensätzen vorhanden sind – daher fehlende, veraltete oder falsche Angaben, die getroffen wurden. Somit besteht die Möglichkeit, dass QIDs nicht übereinstimmen, obwohl sie zur selben Person gehören. Um eine eindeutige Identifizierung von Personen dennoch zu ermöglichen, wurden zahlreiche Record-Linkage Methoden entwickelt. Der Vergleich von einigen dieser Methoden ist ein Hauptziel dieser Bachelorarbeit.

Entscheidendes Merkmal für jede Methode ist die Fähigkeit, möglichst viele Records als Match zu klassifizieren, die zur gleichen Person gehören (*true Match*) und möglichst wenige, die zu unterschiedlichen Personen gehören (*true non-Match*). Die Güte der Klassifizierung wird allgemein als *Linkage-Qualität* bezeichnet (Christen 2012: 34).

Es gilt festzustellen, ob die verschiedenen QIDs unterschiedlichen Einfluss auf die Linkage-Qualität besitzen. Dies wird

1. durch die Stabilität eines QID – daher die Häufigkeit der Änderungen – sowie

2. durch die Anzahl und
3. Verteilung ihrer Ausprägungen

bestimmt.

Trivialerweise müssen die Merkmale auch in beiden Datensätzen vorhanden sein. Insgesamt steht somit meist nur eine begrenzte Zahl an QIDs für ein Record-Linkage Projekt zur Verfügung. Noch weniger davon haben einen maßgeblichen Einfluss auf die Linkage-Qualität.

Der Zweck des Record-Linkage besteht meist darin, Aussagen über eine Population zu treffen, die durch das Linkage als zusammengehörig identifiziert wurde (Doidge und Harron 2019). Wird eine Record durch eine Record-Linkage Methode falsch oder gar nicht mit einem anderen Record verknüpft, so hat dies Auswirkung auf die Aussagen, die getroffen werden. Falls dies systematisch geschieht, kommt es zwangsläufig zu einem *Linkage-Bias*. Dieser besagt, dass Aussagen über eine Subpopulation, die durch eine Record-Linkage Methode schlecht gelinkt wurde, verzerrt sind (Doidge und Harron 2019). Es gilt daher immer den Linkage-Bias zu minimieren. Dies geschieht maßgeblich über drei Einflussfaktoren:

1. Die Datenqualität,
2. die Auswahl der verwendeten QIDs und
3. die Record-Linkage Methode.

Ziel dieser Arbeit ist es, diese drei Einflussfaktoren auf den Linkage-Bias zu untersuchen und dabei besonders die Record-Linkage Methoden miteinander zu vergleichen. Grundlage für die Analyse sind Daten einer Mikro-Simulation eines bundesweiten Schülerregisters. Diese wurde im Rahmen einer Expertise erstellt, die vom Bundesministerium für Bildung und Forschung (BMBF) in Auftrag gegebenen wurde (Schnell 2022). Diese Mikro-Simulation bietet die Möglichkeit, den ersten Einflussfaktor zu untersuchen, da in der Simulation eine Steuerung der Datenqualität implementiert ist. Darüber hinaus bildet die Simulation die Grundlage für die möglichen QIDs, die für das Record-Linkage verwendet werden können.

Die Arbeit gliedert sich somit in drei Teile: In Kapitel 1 wird die Datenerzeugung durch die Mikro-Simulation beschrieben. In Kapitel 2 werden die unterschiedlichen Linkage-Methoden vorgestellt und ihre Implementation und Konfiguration diskutiert. Dies betrifft insbesondere die verwendeten QIDs und somit Einflussfaktoren zwei und drei. In Kapitel 3 werden schließlich die Ergebnisse vorgestellt und diskutiert. Die Idee und das generelle Design dieser Arbeit stammen vom Erstgutachter Prof. Dr. Rainer Schnell.

Kapitel 1

Datengrundlage

1.1 Goldstandard-Daten

Ein entscheidendes Problem von Record-Linkage besteht darin, den Erfolg eines Linkages zu bestimmen. Im Vorfeld ist nicht bekannt, ob zwei Records zur selben Entität gehören (wahrer Matchstatus). Daher kann keine Aussage über die Validität eines Matches gemacht werden. Die Zahl der zu erwartenden Matches und welche Konfigurationen der einzelnen Methoden ein gutes Ergebnis liefern, hängt somit von Erfahrungswissen und vorher getesteten Konfigurationen ab.

Zur Evaluation von Record-Linkage Methoden müssen daher Daten verwendet werden, bei denen der wahre Matchstatus im Vorfeld bekannt ist. Diese Daten werden als *Goldstandard-Daten* bezeichnet (*ground-truth data*). Goldstandard-Daten können durch mehrere Methoden erzeugt werden (Christen 2012: 164 f.):

1. Über vorher schon einmal verknüpfte und validierte reale Daten,
2. über die Verknüpfung von realen Daten anhand einer eindeutigen Kennung (z. B. E-Mail-Adresse oder Steuernummer) oder
3. über eine Simulation von Daten.

Der zentrale Vorteil bei Verwendung von realen Daten (Methoden 1 und 2) besteht darin, dass die Charakteristik der Daten bereits vorliegt. Die Datenqualität stimmt somit mit der in der Realität zu erwartenden überein. Dagegen existieren eine Reihe von Nachteilen bei der Verwendung von realen Daten. Insbesondere die Validität der Klassifizierung ist fragwürdig, aber auch das allgemeine Problem der Existenz eines geeigneten Datensatzes.

Für die erste Methode besteht das Problem, dass nicht alle true Matches gefunden und false Matches aussortiert wurden. Die Datensätze somit zu einem Teil nicht korrekt klassifiziert wurden. Für die

zweite Methode bedeutet dies, dass nicht alle Records über das gewünschte Merkmal verfügen, es selbst von Fehlern behaftet ist oder Personen mehrere Ausprägungen des Merkmals besitzen (z. B. mehrere E-Mail-Adressen). Dies hat zur Folge, dass nur eine Teilmenge der realen Daten klassifiziert werden kann. Darüber hinaus muss ein geeignetes Merkmal in den Daten existieren.

Eine Alternative zur Verwendung von realen Daten ist die Simulation von Daten. In diesem Fall werden Records künstlich erzeugt, sodass die Validität der Klassifikation immer gegeben ist. Die Charakteristik der Daten sowie die Datenqualität muss jedoch einem realen Datenbestand ähneln. Dieses Ziel kann häufig nur approximativ erreicht werden, da der Aufwand einer Simulation groß ist und nur begrenzte Ressourcen zur Verfügung stehen und nicht genügend Informationen über die Charakteristik der realen Daten vorliegen.

1.2 Mikro-Simulation eines Schülerregisters

Wie bereits beschrieben, wurde die Mikro-Simulation eines bundesweiten Schülerregisters im Rahmen einer vom BMBF in Auftrag gegebenen Expertise erstellt (Schnell 2022). Zunächst gilt es daher, die Problemstellung dieser Expertise zu erläutern. Die daraus resultierenden weiteren Probleme und die getroffenen Entscheidungen werden dann in den weiteren Abschnitten behandelt.

1.2.1 Problemstellung

Grund für die Expertise war das Ziel von Bund und Ländern, ein umfassendes Bildungsverlaufsregister zu realisieren. In diesem Register werden Bildungsverläufe vom Schuleintritt über die berufliche und akademische Bildungskarriere abgebildet. Schülerregister existieren schon seit Längerem in Deutschland, jedoch werden diese meist von den Kommunen verwaltet. Bildungsverläufe könnten daher nur selten vollständig rekonstruiert werden, da Personen in einer Kommune durchgehend wohnhaft bleiben müssten. Besonders beim Besuch einer Hochschule ist dies jedoch unwahrscheinlich.

Das Register entsteht daher durch die Zusammensetzung der kommunalen Register in einem regelmäßigen Turnus. Um Bildungsverläufe zu rekonstruieren, müssen dann die Records aus dem neuen Register mit den Records aus dem bestehenden Register mittels Record-Linkage verknüpft werden. Wie bereits geklärt, ist der Erfolg eines solchen Record-Linkages immer schwer zu bestimmen. Das Ziel der Expertise war daher, die zu erwartende Linkage-Qualität von Bildungsdaten auf Basis von Personenmerkmalen (QIDs) zu ermitteln.

Das größte für eine wissenschaftliche Analyse zugängliche Schülerregister befindet sich in Hamburg. Es lagen jedoch keine analysierbaren Längsschnittdaten für dieses Register vor. Aus Gründen des Datenschutzes ist es darüber hinaus sehr schwierig, mit dem Register zu arbeiten. Da somit keine

realen – und gleichzeitig klassifizierten – Daten für eine solche Problemstellung vorliegen, muss das Register simuliert werden. Die Größe des Registers bedingt darüber hinaus, dass eine Skalierung auf die Bundesrepublik nicht möglich ist. Dennoch bietet dieses Register einige Analysemöglichkeiten über die Beschaffenheit eines Schülerregisters, die für die Simulation genutzt wurden.

Da das Register laufend aktualisiert wird, müssen Bildungsverläufe von Personen über die Zeit simuliert werden. Dies bedeutet, dass sich die Merkmale einer Person durch verschiedene Lebensereignisse ändern können. Es handelt sich daher nicht um die einfache Generierung eines Datensatzes wie vielfach für Record-Linkage verwendet (Christen, Ranbaduge et al. 2020: 385 ff.), sondern um eine Simulation über einen Zeitverlauf mit mehreren Simulationseinheiten. Eine solche Simulation wird als *Mikro-Simulation* bezeichnet (Hannappel und Kopp 2020). Bisher gibt es keine publizierten Anwendungen von Mikro-Simulationen zur Beurteilung von Record-Linkage Methoden.

1.2.2 Simulationsumfang

Für die Erstellung einer umfassenden Mikro-Simulation benötigt es für gewöhnlich Jahre. Aufgrund der Komplexität der Wirklichkeit sowie des Mangels an geeigneten Informationen sind Verallgemeinerungen und Ungenauigkeiten immer vorhanden (Hannappel und Kopp 2020: 2 f.). Aufgrund des geringen zeitlichen und personellen Umfangs der Expertise, musste der Simulationsumfang an einigen Stellen begrenzt werden.

Da die Verknüpfung zweier Records nur dann problematisch ist, falls diese sich in den beiden Datensätzen unterscheiden, muss die Mikro-Simulation im Wesentlichen Änderungen am Register abbilden. Folgende Gründe für eine Änderung bestehen:

- Die Person ist umgezogen,
- sie hat ihren Namen geändert (insbesondere durch Heirat) oder
- sie hat die Schule/Bildungseinrichtung gewechselt.

In den genannten Fällen wird die Person im Register neu eingetragen. Dabei besteht immer die Möglichkeit, dass Fehler bei der Eingabe passieren. Diese Fehler sowie tatsächliche Änderungen von Wohnort oder Nachname müssen simuliert werden, um die Linkage-Qualität zu beurteilen.

Die Linkage-Qualität kann darüber hinaus auch gemindert werden, falls Personen sich in vielen Merkmalen gleichen. Hierzu zählen insbesondere Mehrlinge, die in einem Haushalt leben. Bei der Verwendung von üblichen QIDs können diese nur über den Vornamen und ggf. das Geschlecht unterschieden werden. Auch nicht verwandte Personen können sich in ihren Merkmalsausprägungen ähneln. Diese Wahrscheinlichkeit ist insbesondere durch die Verteilung von Namen und Geburtsdaten sowie die Population des Wohnorts bedingt. Auch diese Faktoren werden in der Simulation berücksichtigt.

Alle bisher genannten Faktoren, die die Linkage-Qualität mindern, erfolgen nicht gleichermaßen über eine Bildungslaufbahn hinweg. Vielmehr handeln Personen abhängig ihres Alters und der Art der besuchten Bildungseinrichtung anders. So ziehen bspw. Schüler seltener um als Studierende. Um die verschiedenen Handlungsmuster zu simulieren, werden die Personen im Register in vier Kategorien eingeordnet. Die Handlungsmöglichkeiten der Personen in den Kategorien werden im Folgenden kurz skizziert. Die Entscheidungen für bestimmte Verfahrensweisen sowie die Datengrundlage und konkrete Implementation werden in Abschnitt 1.2.4 beschrieben.

Da der Simulationsumfang begrenzt und darüber hinaus einige Daten nicht verfügbar sind, müssen auch innerhalb der genannten Faktoren Einschnitte gemacht werden. In diesen Fällen wird immer eine Entscheidung zugunsten einer Verschlechterung der Linkage-Qualität gewählt. Die Verschlechterung der Linkage-Qualität wird durch eine höhere Ähnlichkeit von Records erreicht. Es handelt sich daher immer um konservative Schätzungen.[1] Die genauen Fälle und Vorgehensweise werden in Abschnitt 1.2.4 beschrieben.

Schüler

Die meisten Personen gelangen als Schüler in das Bildungsregister. Da in Deutschland eine allgemeine Schulpflicht gilt, misst sich die Zahl der jährlich neuen Schüler an der Größe der vollständigen Geburtskohorte (ca. 700,000). Darüber hinaus wird erwartet, dass weitere 49,700 Schüler von außen durch bspw. Migration in das Bildungssystem gelangen.

Ein Schüler geht solange zur Schule, bis er einen erwarteten Abschluss erreicht. Während der Schulzeit können Schüler umziehen, auswandern und sterben. Darüber hinaus wechseln alle Schüler nach der vierten Klasse von der Primarstufe in die Sekundarstufe. Dies hat somit einen Schulwechsel zur Folge.

Nach Erlangung eines Bildungsabschlusses haben Schüler drei Möglichkeiten, ihre Bildungslaufbahn fortzusetzen: Ein Studium, eine Ausbildung oder das Ende der Bildungslaufbahn.[2] Zur Vereinfachung wird angenommen, dass Personen nur einen Bildungsweg einschlagen und diesen danach nicht mehr wechseln. Bildungskarrieren sind damit zusammenhängend. Beendet somit eine Person die Bildungslaufbahn, so wird sie nicht zu einem späteren Zeitpunkt wieder reaktiviert. Es gilt dabei anzumerken, dass keine Records gelöscht werden. Das Register wächst somit mit der Zeit an.[3]

[1] Besteht bspw. die Auswahl mehrerer plausibler Wertebereiche, so wird immer der kleinstmögliche gewählt. Dies führt dazu, dass sich Records in ihren Werten mehr ähneln und die Linkage-Qualität durch mehr falsch negative sinkt.

[2] Es werden keine Pausen im Bildungsverlauf wie z. B. ein Bundesfreiwilligendienst simuliert.

[3] Der Umfang der Simulation war zu gering um komplexere Bildungsverläufe zu erfassen. Eine Verweildauer in Übergangssystemen oder Praktika sowie Pausen vor dem Beginn einer Ausbildung bzw. eines Studiums wird nicht simuliert.

Studierende

Entscheidet sich ein Schüler für das Studium, so gelangt dieser in die Kategorie der Studierenden. Hinzu kommen ausländische Studierende, die auch in ein Bildungsregister aufgenommen werden. Beide Gruppen unterscheiden sich nach der Aufnahme in das Register in ihren Handlungsmöglichkeiten nicht.

Bei Studierenden sind wie alle demografischen Eigenschaften an die erwartete Bildungsdauer geknüpft. Dies bedeutet, dass auch der Tod über die erwartete Bildungsdauer simuliert wird. Studierende können daher nur umziehen – vorrangig in die Nähe einer Hochschule – und heiraten. Als Partner kommen alle Personen in Betracht, die über 18 Jahre alt und keine Schüler sind. Vorrangig befinden sich beide Partner im Register (weiterführend hierzu Abschnitt 1.3.3).

Auszubildende

Das duale Ausbildungssystem in Deutschland ist im internationalen Vergleich eine Spezialität der Berufsausbildung. Dabei wird praktisches und theoretisches Wissen gleichzeitig vermittelt, indem eine Ausbildung im Betrieb mit der in einer Berufsschule verbunden wird (Minssen 2006: 139). Dies hat zur Konsequenz, dass sich neben außerbetrieblichen Auszubildenden auch betriebliche Auszubildende in einem Schülerregister befinden.

Wie bei den Studierenden richtet sich auch bei den Auszubildenden alles nach der erwarteten Ausbildungsdauer. Während der Ausbildung können sie nur umziehen und heiraten. Das Heiratsverhalten ist dabei gleich zu den Studierenden. Es werden keine Ausländer simuliert, die für eine Berufsausbildung nach Deutschland ziehen, da diese Zahl als vernachlässigbar gering angenommen wird.

Erwachsene

Eine weitere Bildungseinrichtung in Deutschland ist die Volkshochschule. Die Besucher werden in der Simulation als Erwachsene kategorisiert. Aufgrund der geringen Bedeutung dieser Schulform für den Bildungsverlauf werden Erwachsene in der Simulation kaum berücksichtigt. Darüber hinaus bestand wenig Klarheit seitens des BMBF über die Art und Weise, wie Erwachsene in das Schülerregister gelangen können. Für Erwachsene wird daher nur die Heirat simuliert. Die Kategorie erweist sich, neben der antizipierten Form, im Kontext der Simulation zusätzlich als eine Restkategorie für alle Personen, die ihre Bildung abgeschlossen haben. Die Erwachsenen existieren insbesondere, um das Linkage durch die Größe des Datensatzes zu erschweren.

1.2.3 Linkage-Merkmale

Für die Auswahl der Linkage-Merkmale muss mit der Feststellung begonnen werden, dass QIDs personenbezogene Daten sind. Damit Record-Linkage mit amtlichen Daten anhand von QIDs erfolgen kann, muss somit zwangsläufig der Datenschutz berücksichtigt werden. Dabei gilt:

> „Die Verarbeitung personenbezogener Daten durch eine öffentliche Stelle ist zulässig, wenn sie zur Erfüllung der in der Zuständigkeit des Verantwortlichen liegenden Aufgabe oder in Ausübung öffentlicher Gewalt, die dem Verantwortlichen übertragen wurde, erforderlich ist“(§3 BDSG).

Die Verwendung von QIDs ist daher möglich, jedoch ist unklar, welche Daten in diesem Kontext erforderlich sind. Hierbei wird häufig das Prinzip der *Datensparsamkeit* (auch bekannt als „Datenminimierung”) angeführt. Dieses besagt, dass so wenige personenbezogene Daten wie möglich zur Verarbeitung preisgegeben werden sollen. Wie jedoch festgestellt wurde, führen (prinzipiell) mehr geeignete QIDs zu einer besseren Linkage-Qualität. Daher muss immer eine Balance zwischen Datensparsamkeit und Linkage-Qualität gefunden werden.

Hierzu gilt es festzustellen, dass sich der Einfluss der verschiedenen Merkmale auf die Linkage-Qualität oft stark unterscheidet. Dieser hängt von der Stabilität, Vollständigkeit sowie Anzahl und Verteilung der Ausprägungen des Merkmals ab. Wird daher das Prinzip der Datensparsamkeit auf das Schülerregister übertragen, so sollten die wenigen verwendeten Merkmale im Zeitverlauf möglichst stabil und distinktiv sowie wenig fehleranfällig sein. Hierzu zählen folgende Merkmale:

1. Vorname
2. Nachname
3. Geschlecht
4. Geburtsdatum (Tag, Monat, Jahr)
5. Geburtsort
6. Postleitzahl

Die genannten Merkmale werden alle für das Bildungsregister simuliert. Nach einer vertieften Auseinandersetzung mit dem Datenschutz und den auftraggebenden Behörden folgte im späteren Verlauf des Projekts der Entschluss, dass die Nutzung der Postleitzahl für das reale Schülerregister nicht zu erwarten ist. Die Postleitzahl wird daher *nicht* als Linkage-Merkmal verwendet, obwohl sie simuliert wurde.

Bereits die Verwendung des Geburtsortes wird als fragwürdig und somit nicht notwendig gesehen. Die Notwendigkeit dieses Merkmals wird dabei im Rahmen dieser Arbeit getestet. Insbesondere gilt es festzustellen, ob ein Linkage-Bias durch das Fehlen des Merkmals entsteht. Der größte Bias kann

dabei zwischen Migranten und Einheimischen entstehen, da sich diese in vielen Merkmalsausprägungen unterscheiden.

1.2.4 Simulationsablauf

Ausgangspunkt jeder Mikro-Simulation ist ein *Startdatensatz* (base population), in dem die zugrunde liegende Population möglichst genau abgebildet wird (Münnich et al. 2021: 244). In einem regelmäßigen Turnus wird dieser Datenbestand verändert. Da viele Ereignisse im deutschen Bildungssystem jährlich passieren (Einschulung, Einschreibung für die meisten Studiengänge, Ausbildungsbeginn), arbeitet die Registersimulation mit einem jährlichen Turnus. Die dadurch entstandenen Datensätze werden im Folgenden als *Folgejahr* bezeichnet.

Eine Besonderheit dieser Simulation besteht in der Möglichkeit, Datensätze mit unterschiedlichen Datenqualitäten zu simulieren. Wie in Abschnitt 1.2.2 beschrieben, besteht bei jeder Veränderung die Möglichkeit eines Datenfehlers. Die Wahrscheinlichkeit eines Datenfehlers wird als *Fehlerquote* bezeichnet. Die Simulation wurde mit insgesamt vier Fehlerquoten durchgeführt: 0.1%, 0.3%, 0.7% und 1% (siehe Abschnitt 1.3.2).[4]

In Abbildung 1 wird der grundlegende Simulationsablauf dargestellt. Der Prozess beginnt mit der Generierung des Startdatensatzes für ein bestimmtes Startjahr (siehe Abschnitt 1.3.5). In diesem Startdatensatz befinden sich alle Personen, die sich im gewählten Startjahr im Bildungssystem befinden. Der Datensatz versucht möglichst genau die reale Verteilung in Deutschland abzubilden. Für die Simulation wurde das Startjahr 2000 gewählt. Der Datensatz enthält keine Fehler, sodass alle Fehlerquoten den gleichen Ausgangsdatenbestand haben. Demzufolge wird der Startdatensatz im nächsten Schritt mit der ersten Fehlerquote belegt.

Ausgehend von dem mit Fehlern behafteten Startdatensatz wird im nächsten Schritt das Folgejahr generiert (siehe Abschnitt 1.3.6). Sowohl der Datensatz des Folgejahres als auch der Datensatz des aktuellen Jahres unterlaufen im nächsten Schritt einem Preprocessing (siehe Abschnitt 2.2). Dabei werden beide Datensätze bereinigt und für das Record-Linkage vorbereitet. Im nächsten Schritt wird das Record-Linkage mit verschiedenen Methoden durchgeführt. Dieses wird ausführlich in Abschnitt 2 beschrieben.

Wurden alle Methoden ausgeführt, so wird der Zähler für das Jahr erhöht und das nächste Folgejahr berechnet. Dies geschieht so lange, bis ein vordefiniertes Endjahr erreicht ist. Für die Simulation war dies das Jahr 2009.[5] Wurde dieses Jahr erreicht, so wird die nächste Fehlerquote aus der Menge der

[4]Siehe Schnell (2022: 7 f.) zur Plausibilisierung der Fehlerquoten.

[5]Start- und Endjahr wurden anhand der vorliegenden Daten gewählt. Darüber hinaus sollte die Simulation einen möglichst aktuellen Zeithorizont abbilden. Es gilt zu beachten, dass der Zeitraum gewählt wurde, nachdem die Simulation fertig implementiert wurde. Aufgrund des explorativen Charakters der Simulation war es zu Beginn nicht klar, wie viele Jahre simuliert werden müssen, um Effekte zu finden, und wie viele Jahre aufgrund

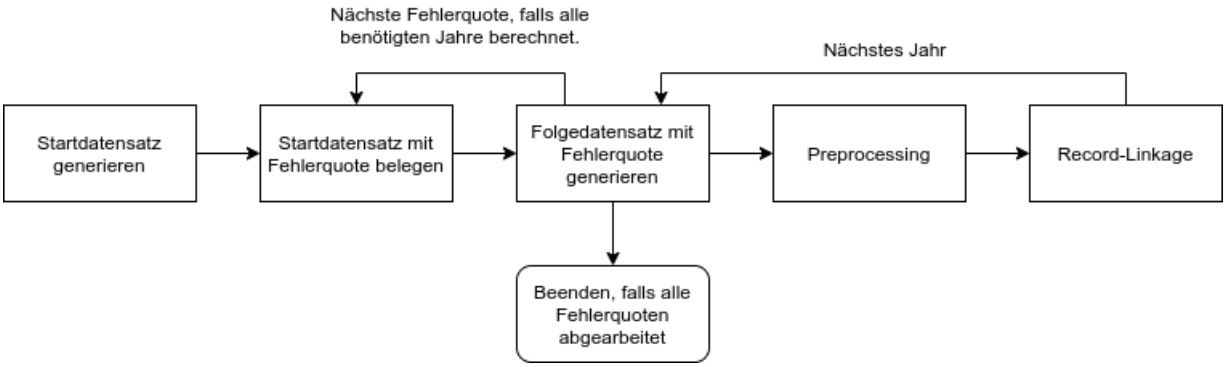

Abbildung 1: Simulationsablauf

Fehlerquoten gewählt. Mit dieser Fehlerquote wird dann der Startdatensatz belegt und die Simulation wieder bis zum Endjahr ausgeführt. Dies geschieht so lange, bis alle Fehlerquoten abgearbeitet wurden. Dann endet die Simulation.

1.3 Generierung der Daten

Wie sich aus dem Simulationsablauf entnehmen lässt, enthält die Simulation zwei Schritte zur Datengenerierung:

1. Die Generierung eines Startdatensatzes und
2. die Fortschreibung eines Datensatzes für alle Folgejahre.

Im Folgenden wird der genaue Ablauf dieser beiden Prozesse beschrieben. Dabei werden einige Prozeduren von beiden Datengeneratoren geteilt. Um Redundanz zu vermeiden, werden erst diese geteilten Prozeduren und danach die beiden Generatoren beschrieben.

Alle Prozesse der Mikro-Simulation verwenden Pseudozufallszahlen mit jahresabhängigen Startwerten (Seeds). Bedingt einer Fehlerquote und einem Jahr, können somit alle Datensätze vollständig repliziert werden.[6]

der zeitlichen Begrenztheit des des Projekts simuliert werden können. Am Anfang des Projekts wurde zunächst das Jahr 2018 als Endjahr gesetzt, weshalb meist die neusten Zahlen in der Simulation verwendet werden. Da es nur möglich ist Folgejahre zu berechnen (und nicht vorherige Jahre), ist die Entscheidung für das Startjahr ein wichtiger später unumkehrbarer Beschluss. Das Startjahr wurde daher auf 2000 gesetzt, sodass noch die Möglichkeit besteht genug Jahre bis zum spätest möglichen Jahr 2018 zu berechnen. Nach ersten Analysen hat sich jedoch gezeigt, dass es ausreichend und auch zeitlich nicht anders möglich war, das Bildungsregister bis zum Jahr 2009 zu simulieren.

[6]Technisch gesehen ist auch die Zahl der parallelen Prozesse relevant (120 Kerne). Dies liegt an der Parallelisierung der Datenverarbeitung, da jeder parallele Prozess einen eigenen Zufallsgenerator besitzt. Falls eine geringere Rechenleistung zur Verfügung steht, kann diese Eigenschaft jedoch manuell umgangen werden, indem die Zahl der Kerne auf 120 festgesetzt wird.

1.3.1 Generierung einer neuen Schülerkohorte

Zentral für die Simulation des Bildungsregisters ist die Generierung einer neuen Schülerkohorte. Ob ein Kind eingeschult wird, hängt in Deutschland für gewöhnlich von einem Stichtag ab. Alle Kinder, die vor dem entsprechenden Stichtag geboren wurden und noch nicht zur Schule gehen, werden in der Regel eingeschult. Da die Bildung in Deutschland Aufgabe der Bundesländer ist, existieren mehrere solcher Stichtage. Diese sind wiederum zeitlich nicht stabil. Zur Vereinfachung wird für die Simulation der 30.06. als fester Stichtag für alle Bundesländer angenommen. Zudem existieren keine Vor- oder Rückstufungen bei der Einschulung.[7]

Die neue Schülerkohorte besteht daher aus einem variablen Teil von Kindern, die vor dem aktuellen und nach dem letzten Stichtag geboren sind. Hinzu kommt ein fester Teil von 57,200 neu hinzukommenden (migriert, remigriert etc.) Kindern.[8] All diesen Kindern wird zu Beginn eine eindeutige und feste Personenidentifikationsnummer (PID) zugewiesen. Anhand dieser PID kann fortlaufend der wahre Matchstatus kontrolliert werden. Die Zusammensetzung aller anderen Eigenschaften eines Kindes wird im Folgenden erläutert.

Geschlecht

Das Geschlecht der Kinder wird als gleichverteilt angenommen. Männlich wird als 1 codiert und weiblich als 2. Andere Formen des Geschlechts werden nicht berücksichtigt.

Mehrlinge

Es wird angenommen, dass die Zahl der Mehrlingsgeburten über den Simulationszeitraum konstant ist. Für eine Anzahl an Mehrlingen x (Zwillinge, Drillinge etc.) berechnet sich die entsprechende Wahrscheinlichkeit p_x als

$$p_x = \frac{\text{Zahl der Geburten mit x Mehrlingen}}{\text{Gesamtzahl der Geburten}}. \tag{1.1}$$

Da die Zahl der Vier- und Fünflinge immer sehr gering ist, wird diese Zahl den Drillingen hinzugerechnet. In der Simulation existieren daher nur Zwillinge und Drillinge. Im Jahr 2020 gab es 13,663

[7] Für ein Jahr j liegt die Geburt eines Kindes daher zwischen dem 01.07.(j-6) und dem 30.06.(j-5). Diese Vereinfachung ist möglich, da sie die Wahrscheinlichkeit für Uneindeutigkeiten erhöht und somit die Linkage-Qualität mindert.

[8] Die genaue Zusammensetzung dieser Zahl wird im Unterabschnitt Geburtsdatum beschrieben. Die Daten basieren auf einem für das Bundesministeriums für Familie, Senioren, Frauen und Jugend erstellten Gutachten (Bujard et al. 2019: 14). Dort wird das Alter von syrischen, afghanischen, irakischen und eritreischen Migranten untersucht, die zwischen 2015 und 2017 nach Deutschland eingewandert sind. Die Daten sind daher nicht direkt verallgemeinerbar, um die Zahl der Neuzugänge zu beurteilen. Da jedoch keine genaueren Daten vorliegen, wurde anhand dieser Daten die tatsächlich simulierte Zahl geschätzt.

Zwillings- und 204 Drillingsgeburten auf insgesamt 773,144 Geburten (Statistisches Bundesamt 2021b). Daraus folgt, dass $p_{x=2} \approx 1.8\%$ und $p_{x\geq 3} \approx 0.03\%$.

Nach den genannten Wahrscheinlichkeiten und der Rest-Wahrscheinlichkeit wird jedem Kind eine Geburtsart zugewiesen. Aus der Menge der Einzelgeburten erhalten alle Mehrlinge nun eine entsprechende Zahl an Geschwistern. Die Geschwister gehen dann untereinander eine Bindung ein. Dies bewirkt, dass die Geschwister denselben Nachnamen, Migrationsstatus und dasselbe Geburtsdatum bekommen. Darüber hinaus führt die Bindung dazu, dass in der gesamten Schulzeit die Geschwister immer gemeinsam umziehen. Nach der Schulzeit endet diese Bindung. Reguläre Geschwister werden aufgrund des zeitlichen Rahmens nicht direkt simuliert, sondern entstehen nur indirekt durch einen gleichen Nachnamen und Geburtsort eines anderen Records.

Schulabschluss

In der Simulation wird nicht nach Abschlussart, sondern nach Verweildauer im Bildungssystem unterschieden. Diese Vereinfachung ist möglich, da die Abschlussart kein Linkage-Merkmal ist und nur die Verweildauer relevant ist. Im Groben lassen sich zwei Verweildauern in Deutschland feststellen:

- Zehn Jahre für einen Haupt- oder Realschulabschluss (33% der jährlichen Abschlüsse) und
- 13 Jahre für ein (Fach-)Abitur (67% der jährlichen Abschlüsse).[9]

Nach den genannten Abschlusswahrscheinlichkeiten wird jedem Kind eine der beiden Verweildauern zugewiesen.

Adresse

Die Adressbildung beginnt bei der Festlegung des Bundeslandes. Dies erfolgt proportional zum Anteil der gemeldeten Schüler in einem Bundesland (Statistisches Bundesamt 2021a). Bundesländer mit einer durchschnittlich jüngeren Bevölkerung werden daher stärker gewichtet. Da die amtliche Statistik in Deutschland keine Postleitzahlen führt, wird anhand des Bundeslandes zunächst der amtliche Gemeindeschlüssel (AGS) bestimmt. Dies erfolgt proportional zur Bevölkerungszahl der einzelnen Gemeinden. Anhand der AGS wird im letzten Schritt eine Postleitzahl bestimmt. Auch diese erfolgt proportional zu einer geschätzten Bevölkerungszahl innerhalb des Postleitzahlraums.[10]

[9] Datengrundlage ist die Zahl der aktuell gemeldeten Schüler nach Schulform (Statistisches Bundesamt 2021a).

[10] Da die amtlichen Zahlen hierfür nicht existieren, wurde ein nicht amtlicher Datensatz verwendet, der die Bevölkerungszahl schätzt. Generiert wurde der Datensatz, indem die Rasterdaten des Zensus 2011 (100x100m Quadrate) über die Fläche der einzelnen Postleitzahlen gelegt wurde (https://blog.suche-postleitzahl.org/post/132153774751/einwohnerzahl-auf-plz-gebiete-abbilden (abgerufen am 13.04.2022)). Dieses Verfahren ist fehleranfällig, für eine approximative Schätzung anhand der AGS sind diese Daten aber ausreichend.

Migrationsstatus

Der Migrationsstaus wird zufällig über den Anteilswert der Migranten innerhalb einer AGS bestimmt.[11] Es wird angenommen, dass der Migrationsstatus unabhängig von der Aufenthaltsdauer und damit vom Geburtsdatum ist. Alle Kinder innerhalb einer AGS haben daher die gleiche Wahrscheinlichkeit, Migrant zu sein.

Für Migranten gilt, dass ihre Namen aus einer anderen Namensverteilung gezogen werden und der Geburtsort außerhalb von Deutschland liegt.[12] Da anzunehmen ist, dass ausländische – und daher weniger gebräuchliche – Namen häufiger von Datenfehlern betroffen sind, erhalten sie auch eine andere Fehlerquote (siehe Abschnitt 1.3.2).

Geburtsort

Alle Einheimischen erhalten den Gemeindenamen der zuvor generierten AGS als Geburtsort. Der Geburtsort aller Migranten wird aus einer Ortsliste der häufigsten Herkunftsländer von Migranten zufällig gezogen (gleichverteilt innerhalb eines Landes).[13]

Geburtsdatum

Da die Zahl der Geburten über das Jahr nicht gleichverteilt ist und das Geburtsdatum zudem ein zentrales Linkage-Merkmal ist, müssen Häufungen an Geburtsdaten simuliert werden. Die Festlegung des Geburtsdatums beginnt dabei mit der Bestimmung des genauen Geburtsjahres. Wie bereits beschrieben, ist die Zahl der Geburten vor dem aktuellen und nach dem letzten Stichtag bekannt. Aus der bisher generierten Kohorte wird daher genau diese Menge an Kindern zufällig ausgewählt. Dies geschieht – abgesehen von der Bindung bei Mehrlingen – ungeachtet sonstiger Eigenschaften.

Abhängig vom Geburtsjahr und den Grenzen der Stichtage kann nun ein Geburtsmonat festgelegt werden. Anhand einer Verteilung der Geburtenzahlen pro Monat und Jahr wird allen Kindern ein Geburtsmonat zugewiesen, proportional zu der Anzahl an Geburten innerhalb der genannten Grenzen.[14]

Für alle weiteren Neuzugänge wird angenommen, dass pro Jahr 3400 Kinder zwischen 10 und 17 und 5000 zwischen 7 und 9 Jahren zur Schule kommen. Hinzu kommen 7500 Kinder, die mit 6 Jahren und 2500 Kindern, die mit 5 Jahren zur Schule kommen.[15] Dabei handelt es sich um das Alter

[11] Die Angaben basieren auf einer kommerziellen Geomarketing-Datenbank.

[12] Migranten zweiter Generation mit Geburtsort in Deutschland werden als reguläre Deutsche behandelt.

[13] Die vorhanden Daten ließen es nicht zu, dass die Geburtsorte nach Einwohnerzahl gezogen werden konnten.

[14] Alle hier genannten Verteilungen wurden aus einer administrativen Datenbank berechnet.

[15] In Summe ergibt dies die oben bereits beschriebene Zahl an Neuzugängen: $3400 \cdot 8 + 5000 \cdot 3 + 7500 + 2500 = 57200$. Die Angaben weichen vom Ursprungsbericht ab, da ein nicht erkannter Fehler im Programmcode existiert. Aufgrund der Laufzeiten der Simulation konnte dieser Fehler jedoch behoben werden. Fälschlicherweise wurden

am Jahresanfang und nicht um das Alter bei der Einschulung. Es werden daher keine Fünfjährigen eingeschult. Alle Kinder, denen noch kein Geburtsjahr zugewiesen wurde, erhalten entsprechend den gerade beschriebenen Anzahlen ein Geburtsjahr. Abhängig vom Jahr wird auch diesen Kindern ein Geburtsmonat proportional zu den Geburten in einem Monat zugewiesen.

Zuletzt erhalten alle Kinder einen Geburtstag zugewiesen. Da sich Geburten auch über die Wochentage unterschiedlich verteilen, wird dafür zunächst der Wochentag der Geburt bestimmt. Hierzu wird eine Verteilung der täglichen Geburten innerhalb eines Jahres verwendet und daraus die Wahrscheinlichkeiten gebildet, in einem Monat an einem bestimmten Wochentag geboren zu sein. Abhängig vom Geburtsmonat erhält jedes Kind nach dieser Verteilung einen Wochentag zugewiesen. Mit Geburtsmonat und -jahr sowie dem Wochentag wird nun eine Menge an möglichen Geburtstagen für ein Kind berechnet. Innerhalb dieser Menge wird gleichverteilt ein Geburtstag gezogen und dem Kind zugewiesen. Wurde z. B. ein Kind an einem Donnerstag im März 2005 geboren, so wird der Geburtstag aus der Menge der möglichen Donnerstage ({3., 10., 17., 24., 31.}) gezogen.

Vor- und Nachname

Da sich im zeitlichen Verlauf die Namensgebung in den Geburtskohorten unterscheidet, ist zu erwarten, dass die Namensverteilung von Schülern sich von der restlichen Bevölkerung unterscheidet. Daher ist es entscheidend, die Namensgebung von Schülern in der Simulation an die echte Namensverteilung anzupassen. Dafür wurde für das Projekt eine verschlüsselte Joint-Distribution der Namen des Hamburger Schülerregisters verwendet.[16] [17]

In der Joint-Distribution sind die Kombinationen von Vornamen (bis zu sechs) und Nachnamen (bis zu vier) eines Kindes im Hamburger Schülerregister enthalten. Alle Namen wurden einzeln durch eine kryptografische Hash-Funktion verschlüsselt. Namen werden somit durch einen Hash-Wert repräsentiert und unkenntlich gemacht. Anhand der Hash-Werte können dennoch die benötigten Häufigkeiten berechnet werden.

In der Namensgenerierung wird zwischen Namen für Migranten und Einheimische sowie Jungen und Mädchen unterschieden. Alle Namen werden anhand der vollständigen Hamburger Joint-Distribution gebildet. Hierzu wurde ein Algorithmus konzipiert, der die Charakteristik der Hamburger Namensverteilung auf eine größere Population abbildet (siehe Abbildung 2). Die Funktionsweise des Algorithmus wird im Folgenden genauer beschrieben.

---

die 2500 Fünfjährigen doppelt berechnet. Anstatt ursprünglich 5000 angestrebten Sechsjährigen werden daher tatsächlich 7500 gebildet.

[16] Für die Bereitstellung der Namensdaten des Hamburger Schülerregisters danke ich Herrn Dr. habil. Tobias Brändle, Institut für Bildungsmonitoring und Qualitätsentwicklung (IfBQ).

[17] Auch die Hamburger Namensverteilung unterscheidet sich von der gesamtdeutschen Namensverteilung der Schüler. Mangels großer Schülerregister sind bessere Daten in Deutschland jedoch nicht verfügbar.

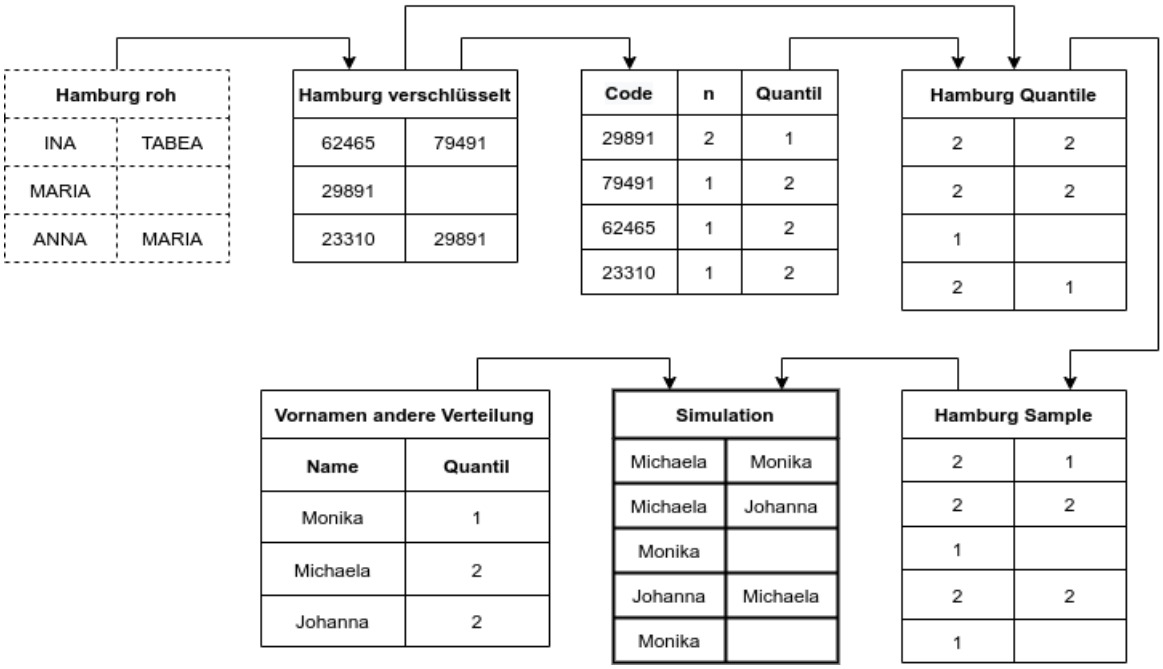

Abbildung 2: Beispielhafte Darstellung der Namensgenerierung für einen einheimischen weiblichen Vornamen. Der zugehörige Nachname wurde zur Vereinfachung weggelassen. Zudem wurden die Daten in nur zwei Quantile aufgeteilt und nur zwei Namensbestandteile verwendet.

Zu Beginn werden alle Hash-Werte in der Joint-Distribution ausgezählt und sortiert nach Häufigkeit einem Perzentil zugeordnet. Die Einordnung in ein Perzentil ist eindeutig, sodass diese nicht zwangsläufig gleich groß sind. Dies wird, unabhängig von der Position des Namens getrennt für Vor- und Nachname durchgeführt. Die resultierenden Tabellen werden nun genutzt, um die Hash-Werte in der Hamburger Verteilung durch das entsprechende Perzentil zu ersetzen. Der Wert des Perzentils gibt somit an, wie häufig der verwendete Name ist. Ein niedriger Wert gibt einen häufigen Namen an (z. B. Peter) und ein hoher Wert einen seltenen Namen (z. B. Jarno).

Auch aus einem großen administrativen Namensdatensatz werden nach dem oben beschriebenen Ablauf insgesamt sechs Perzentil-Zuordnungen durchgeführt – vier für Vornamen (männlich/ weiblich, ausländisch/einheimisch) und zwei für Nachnamen (ausländisch/einheimisch). Um Dopplungen eines Namens zu verhindern (z. B. Peter Peter), müssen die Perzentile bei dieser Zuordnung mindestens sechs (bei Vornamen) bzw. vier (bei Nachnamen) Namen beinhalten. Die Größen entsprechen der maximalen Anzahl an Namensbestandteilen der Joint-Distribution. So kann aus jedem Perzentil ein Name ohne Dopplungen generiert werden, indem bereits verwendete Namensbestandteile ausgeschlossen werden.

Um die Hamburger Namensverteilung (ca. 400,000 Namen) auf die Größe des Simulationsdatensatzes zu skalieren, wird nun eine Stichprobe mit Zurücklegen aus den mittels Perzentilen codierten Hamburger Namensdaten gezogen und jedem neuen Kind zugeordnet. Jedes Perzentil wird nun abhängig von den Eigenschaften eines Kindes (männlich/weiblich, ausländisch/einheimisch) durch einen Klarnamen ersetzt. Dieser Klarname wird aus der entsprechenden Tabelle bekannter Namen gezogen. Hierzu

wird ein Name aus dem gleichen Perzentil gezogen, wobei die Wahrscheinlichkeiten für einen Namen innerhalb der Perzentile gleichverteilt sind.

Die Charakteristik der Hamburger Namensverteilung wird beibehalten. Häufige Namen (wie z. B. Peter Müller) sind auch in den Registerdaten häufig. Falls die Namen mit abnehmender Häufigkeit in höhere Perzentile einsortiert werden, so bedeutet dies, dass sich in den höheren Perzentilen mehr Namen befinden. Somit werden trotz der geringen Größe des Hamburger Datensatzes seltene Namen erzeugt, da diese höheren Perzentile mehr Kombinationen zulassen. Alle genannten Eigenschaften werden für ein bundesweites Register erwartet.

Klasse

Zusätzlich zur angestrebten Verweildauer im Schulsystem wird die Angabe einer Klasse benötigt, mit der das Erreichen des bestimmten Abschlusses überprüft werden kann. Die Einstufung in einer Klasse geschieht abhängig vom Geburtsdatum und den Stichtagen ohne die Berücksichtigung von Zurückstufungen. Dies ist besonders für Kinder relevant, die von außen in das Schulsystem gelangt sind. Jedem Kind wird daher eine Klasse zugewiesen, in der sich die Kinder derselben Geburtskohorte befinden.

1.3.2 Erzeugung von Datenfehlern

Die Daten der Simulation sind so strukturiert, dass für jedes Linkage-Merkmal eine Spalte mit dem wahren und eine mit dem fehlerbehafteten Wert existiert. Alle Prozeduren (außer der Fehlerprozedur) verwenden und modifizieren immer die wahren Werte; alle Record-Linkage Methoden verwenden immer die fehlerbehafteten Werte. Die Fehlerprozedur überführt daher die wahren Werte in fehlerbehaftete.

Im Kontext der Simulation entstehen Datenfehler, wenn die Daten einer Person neu in das Register eingegeben werden. Daher werden Datenfehler immer auf eine Menge an N Records angewendet. Dies können z. B. alle Schüler sein, die umgezogen sind oder die Schule beendet haben. Aufgrund der Neueingabe werden im ersten Schritt alle fehlerbehafteten mit den wahren Werten ersetzt. Datenfehler im ursprünglichen Datensatz werden somit zunächst korrigiert.

Die korrigierten Daten werden im nächsten Schritt mit einem Fehler behaftet. Anhand der Fehlerquote F (0.1%, 0.3%, 0.7% und 1%, siehe Abschnitt 1.2.4) werden n Records ausgewählt, deren korrigierte Werte wieder mit einem Fehler belegt werden. Für jedes Linkage-Merkmal werden unabhängige Ziehungen von n verwendet.

Datenfehler können verschiedene Ursachen haben. Entscheidende Einflussfaktoren sind die Eingabemethode und Datenquelle. Ein Modell für die Fehlerarten, welche durch diese Einflussfaktoren

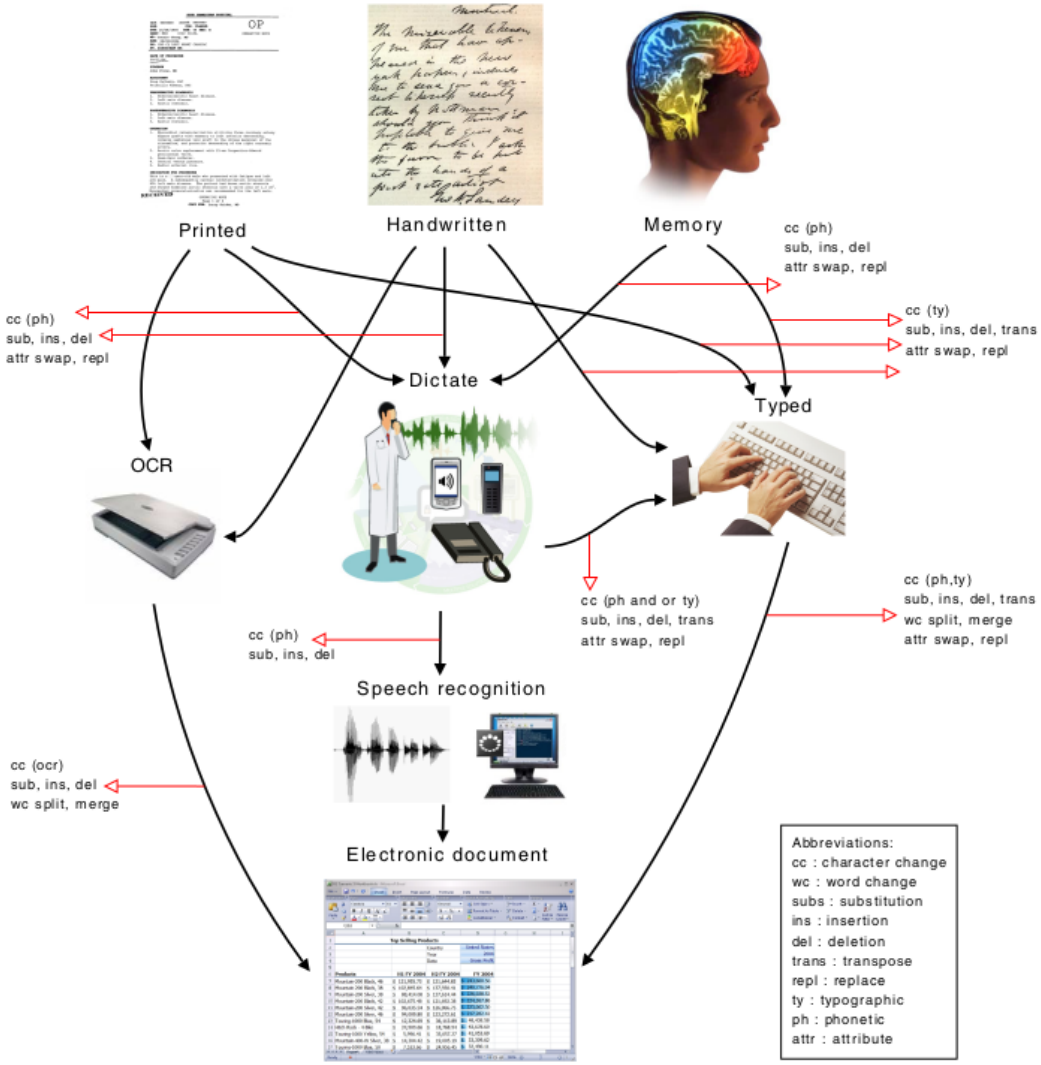

Abbildung 3: Mögliche Fehlerarten, die durch verschiedene Datenquellen und Eingabemethoden entstehen können (Christen und Pudjijono 2009: 511).

entstehen können, wird in Abbildung 3 dargestellt. Die Daten für ein reales Schülerregister folgen nach der Abbildung höchstwahrscheinlich zwei Pfaden: Handwritten-Typed-Document und Memory-Dictate-Typed-Document.

Ein Faktor, der in der Abbildung nicht berücksichtigt wird, ist die Geläufigkeit von Wörtern. So kann angenommen werden, dass weniger geläufige Wörter mehr Fehler enthalten. Diese Überlegung hat besonders einen Einfluss auf die Fehlerhäufigkeit bei Migranten. So kann angenommen werden, dass Migranten weniger geläufige Namen haben. Es kommt daher zum einen zu einer höheren Wahrscheinlichkeit von Fehlern und zum anderen werden Fehler seltener korrigiert. Insbesondere wenn keine lateinischen Schriftzeichen oder Sonderzeichen verwendet werden, können mehrere Schreibweisen für einen Namen existieren (McKenzie 2010), sodass unterschiedliche Einträge entstehen.

Auch für eine mündliche Angabe der Daten kann angenommen werden, dass mehr Fehler bei Migranten auftreten. Diese Annahme beruht nicht nur auf Verständigungsproblemen, sondern vor allem auf der Aussprache der Namen – und anderer Daten. Insbesondere die unterschiedliche Aussprache von Silben kann dazu führen, dass selbst klar ausgesprochene Namen falsch codiert werden, da die Schreibweise des Namens unterschiedlich aufgefasst wird.

Die genannten Gründe sprechen somit dafür, dass für Migranten ein *höheres* Fehlerpotenzial erwartet werden kann. Allgemein existieren wenige Publikationen – ungeachtet der Relevanz dieses Themas – über die genaue Zusammensetzung von Datenfehlern. Dies trifft auch für das genannte Fehlerpotenzial bei Migranten zu. Die Auswirkung der Faktoren wird daher mit einer *Verdopplung* der Fehlerquote geschätzt. Insgesamt kann somit die Menge n für ein Linkage-Merkmal m wie folgt berechnet werden:

$$n_m = \lceil N_{Einheimisch} \cdot F \rceil + \lceil N_{Migrant} \cdot F \cdot 2 \rceil. \tag{1.2}$$

Wurden alle Werte für n_m berechnet, so wird für jedes Linkage-Merkmal ein Fehler erzeugt. Da unterschiedliche Fehlerarten bei den Linkage-Merkmalen auftreten können, folgt jedes Merkmal einer eigenen *feldspezifischen Charakteristik*. Diese werden im Folgenden beschrieben.

Fehler bei Zeichenketten

Für Linkage-Merkmale mit Zeichenketten (Vorname, Nachname, Geburtsort) werden Tippfehler angenommen. Abbildung 3 zeigt eine breite Menge an möglichen Fehlern, die durch eine Eingabe entstehen können.

Der bereits beschriebene Mangel an Publikationen zu dem Thema führt jedoch dazu, dass für viele der benannten Fehler die anzunehmende Fehlerhäufigkeit nicht bekannt ist. Darüber hinaus können weitere Angaben über das Schülerregister getroffen werden. So kann davon ausgegangen werden, dass die Angabe der Daten verpflichtend ist. Fehlende Werte werden somit ausgeschlossen. Darüber hinaus kann eine Vielzahl von Fehlern durch ein Preprocessing bereinigt werden (siehe Abschnitt 2.2). Die Simulation solcher Fehler ist daher nicht notwendig. Die Fehlerprozedur wurde daher mit einer vereinfachten Form der Fehlermöglichkeiten bei Tippfehlern implementiert. Drei Faktoren für Tippfehler gilt es dabei zu klären: Anzahl, Art und Position.

Zunächst wird die Anzahl der Fehler festgelegt. Peterson (1986) konnte in den ihm verfügbaren Daten feststellen, dass 7.1% der Textfelder mehr als einen Tippfehler enthalten.[18] Diese Wahrschein-

[18]Die Daten entstammen aus 16 abgetippten Dokumenten mit insgesamt 369546 Wörtern, die von Peterson (1986) gesammelt wurden. Die Fehlerquote wurden berechnet, indem für jedes Wort im Datensatz alle möglichen Kombinationen von Falschschreibweisen gebildet wurden und der Datensatz auf jede Falschschreibweise durchsucht wurde.

Fehlerart	Anteil
Ersetzung eines Zeichens	43.1
Vertauschung zweier Zeichen	2.8
Löschung eines Zeichens	34.0
Einfügen eines Zeichens	20.1

Tabelle 1: Fehlerwahrscheinlichkeiten (in %)

lichkeit wird auch für die Fehlerprozedur angenommen, wobei nicht mehr als zwei Fehler in einem Feld auftreten können. Die Art und Position der beiden Fehler ist zudem unabhängig voneinander.

Im nächsten Schritt wird die Art des Fehlers bestimmt. Die verwendeten Arten und Wahrscheinlichkeiten für einen Fehler befinden sich in Tabelle 1 (Peterson 1986).

Im letzten Schritt wird die Position des Fehlers festgestellt. Hierbei ist bekannt, dass weniger Fehler am Anfang eines Wortes auftreten als am Ende. Christen und Pudjijono (2009) konnten feststellen, dass 8% der Fehler den ersten Buchstaben betreffen. Eine Fehlerwahrscheinlichkeit für jede Postionen, abhängig von der Wortlänge, ist jedoch nicht bekannt. So wird angenommen, dass 8% der Fehler den ersten Buchstaben betreffen und die restlichen 92% gleichverteilt die übrigen Buchstaben.

Mit Hilfe der Position und Art des Fehlers kann nun der Fehler in der Zeichenkette kreiert werden. Das Löschen gestaltet sich dabei als trivial. Das Ersetzen oder Einfügen eines Zeichens benötigt zusätzlich ein Ersetzungszeichen. Dieses Zeichen wird gleichverteilt aus den 26 Buchstaben des Alphabets gezogen. Darauffolgend wird das entsprechende Zeichen an der gewählten Position eingefügt.[19]

Unter der Vertauschung von Zeichen wird nur die Vertauschung zweier nebeneinanderliegender Zeichen verstanden. Daher muss in diesem Fall bestimmt werden, ob das Zeichen an der gewählten Position mit dem linken oder rechten angrenzenden Zeichen getauscht wird. In einigen Fällen ist eine Auswahl jedoch nicht möglich. So hat das erste und letzte Zeichen nur ein angrenzendes Zeichen. Darüber hinaus würde eine Vertauschung des zweiten Zeichens nach links einen Fehler des ersten Zeichens bewirken. Da diese Fehlerwahrscheinlichkeit festgelegt ist, kann das zweite Zeichen nur nach rechts vertauscht werden. Die Wahrscheinlichkeit für die beiden Möglichkeiten ist in allen anderen Fällen gleichverteilt.

[19]Wird ein Zeichen eingefügt, so wurde dies für die Positionsbestimmung mit in Betracht gezogen. Es ist daher möglich, dass Buchstaben am Anfang oder Ende eingefügt werden. Wird ein Buchstabe am Anfang eingefügt, so entspricht dies einem Fehler im ersten Buchstaben und ist daher mit der entsprechenden Wahrscheinlichkeit möglich.

Fehler bei Zahlenfeldern

Über die Art von Fehlern in Zahlenfeldern ist wenig bekannt. Es kann jedoch angenommen werden, dass mindestens eine einfache Eingabeprüfung besteht. Der Wertebereich wird somit auf eine geringe Anzahl an möglichen Eingaben reduziert. Dies mindert die Linkage-Qualität, da mehr Uneindeutigkeiten entstehen. Die Annahme einer Eingabeprüfung kann somit nach der Prämisse aus Abschnitt 1.2.2 getroffen werden.

Ein Fehler in einem Zahlenfeld gestaltet sich somit als gleichverteilte Auswahl eines neuen Wertes aus einer Menge an möglichen Werten. Von dieser Menge wird zudem der aktuelle Wert ausgeschlossen. Folgende Mengen wurden für die Merkmale angenommen:

- $Geschlecht = \{1, 2\}$
- $Geburtstag = \{1, 2, ..., 31\}$
- $Geburtsmonat = \{1, 2, ..., 12\}$
- $Geburtsjahr = \{1900, 1901, ..., 2018\}$
- $Postleitzahl = \{d \in D, D = \text{alle Postleitzahlen Deutschlands}\}$

1.3.3 Simulation von Ehen

Heiratsfähig sind in der Simulation alle Personen, die mindestens 18 Jahre alt sind und nicht mehr zur Schule gehen. Wird eine Ehe geschlossen, so ziehen Partner immer gemeinsam um. Darüber hinaus kann sich der Nachname eines Partners ändern. Die Art des Namenswechsels verteilt sich nach Tabelle 2.[20]

Um Paare zu finden, wird eine ausgewählte Heiratsziffer für Geschlecht und Alter (18-39) in den Jahren 1990-2018 verwendet (Bundesinstitut für Bevölkerungsforschung o. J.). Für ein gegebenes Jahr wird immer die entsprechende Heiratswahrscheinlichkeit verwendet. Liegt das zu simulierende Jahre außerhalb der verfügbaren Daten, so werden die letzten bekannten Zahlen verwendet. Die Heiratsziffer für z. B. das Jahr 1985 wird daher mit den Daten für das Jahr 1990 geschätzt. Ehen über 39 Jahre werden nicht simuliert, da zum einen die Zahlen nicht ermittelt werden konnten und zum anderen diese für das Schülerregister nicht mehr relevant sind.

Für jedes Alter und Geschlecht wird eine Menge von Personen die heiraten zufällig ausgewählt. Die Größe der Stichprobe bestimmt sich nach der entsprechenden Heiratsziffer. Es wird zudem angenommen, dass sich das Alter der Partner maximal um 10 Jahre unterscheidet.

[20] Die Zahl der Namenswechsel wurden von der Gesellschaft für deutsche Sprache e. V. anhand einer Befragung von 174 Standesämtern mit durchschnittlich 20,000 Eheschließungen pro Jahrgang ermittelt (https://gfds.de/familiennamen-bei-der-heirat-und-vornamenprognose-2018 (abgerufen am 13.4.2022)).

Art des Wechsels	Anteil (in %)
Beide behalten ihren Namen	13.0
Die Frau wechselt den Namen	73.8
Der Mann wechselt den Namen	6.0
Die Frau erhält einen Doppelnamen	6.4
Der Mann erhält einen Doppelnamen	0.8

Tabelle 2: Namenswechsel nach der Hochzeit

Anhand der Menge an Personen, die heiraten werden, können nun Paare gebildet werden. Hierfür wird jeder Frau ein passender unverheirateter Mann zugewiesen.[21] Dies ist eine einfache Zufallsauswahl aus der Menge möglicher Kandidaten (gleichverteilt). Die Paarbildung erfolgt immer für alle Frauen eines gleichen Alters. Es wird mit 18 Jahren begonnen und danach iterativ bis 39 hochgezählt. Das Vorgehen lässt sich darin begründen, dass Frauen jünger heiraten als Männer. Da Partner mindestens 18 Jahre alt sein müssen, finden sich fortlaufend für Frauen immer ältere Partner, da die jüngeren durch die vorherigen Ehen bereits ausgeschlossen wurden.

Findet sich für eine Frau kein passender Mann, so wird der Nachname und die Adresse des Partners simuliert. Der Nachname wird aus einer großen kommerziellen Namensdatenbank gezogen. Die Adresse wird proportional zur Einwohnerzahl von Bundesland, AGS und PLZ gezogen (Vorgehen wie in Abschnitt 1.3.1).

Besteht für einen Partner bereits eine Ehe, so wird diese vorher aufgelöst. Dies führt zur Auflösung der Paarbindung der alten Ehe. Alle anderen Merkmale bleiben jedoch für den alten Ehepartner bestehen. Scheidungen können nur über diesen Weg auftreten.

1.3.4 Simulation der Ausbildungs- und Studiendauer

Die Dauer einer Ausbildung oder ein Studium ist nicht durch eine Schulpflicht noch eine vordefinierte Abschlusszeit determiniert. Daher muss simuliert werden, wie lange eine Person für eine Ausbildung bzw. ein Studium benötigt.

Für die Ausbildungsdauer wird die Verteilung der Tabelle 3 angenommen. Für die Studiendauer muss zunächst zwischen Bachelor- und Masterstudiengängen unterschieden werden. Dabei wird angenommen, dass 25% der Bachelor-Studierenden ihren Abschluss machen und 75% der Master-Studierenden. Abhängig von der Abschlussart wird die Studiendauer durch einen Algorithmus bestimmt (Listing 1.1).

[21] Zur Vereinfachung wurden nur heterosexuelle Ehen gebildet.

Ausbildungs-jahre	Anteil (in %)
1	1
2	4
3	30
4	64
5	1

Tabelle 3: Angenommene Verteilung der Dauer der Ausbildungsjahre

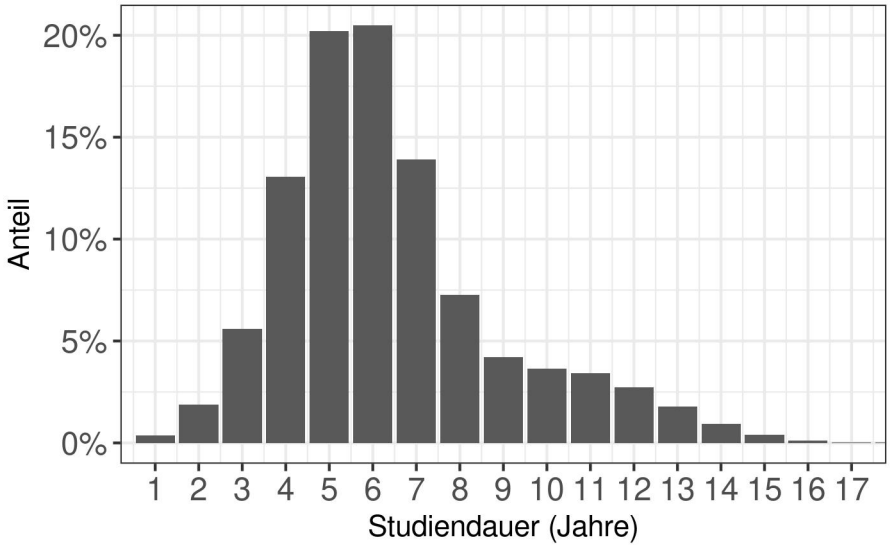

Abbildung 4: Approximative Verteilung der Studiendauer (berechnet für 1 Mio. Studierende).

```
n <- SampleSize
n1 <- (rnorm(n) * 3) + 10
n2 <- (rnorm(n) * 3) + 10
n1[n1 < 1] <- 1
n2[n2 < 1] <- 1
bachelor <- runif(n) > 0.75
# 25% der Bachelorstudierenden machen ihren Abschluss
master   <- runif(n) > 0.25
# 75% der Masterstudierenden machen ihren Abschluss
total <- c((n1 + n2)[n1 > 4 & bachelor & master],
(n1)[n1 <= 4 | !bachelor | !master])
studiendauer <- sample(ceiling(total / 2), n)
```

Listing 1.1: Erzeugung der Studiendauer in R

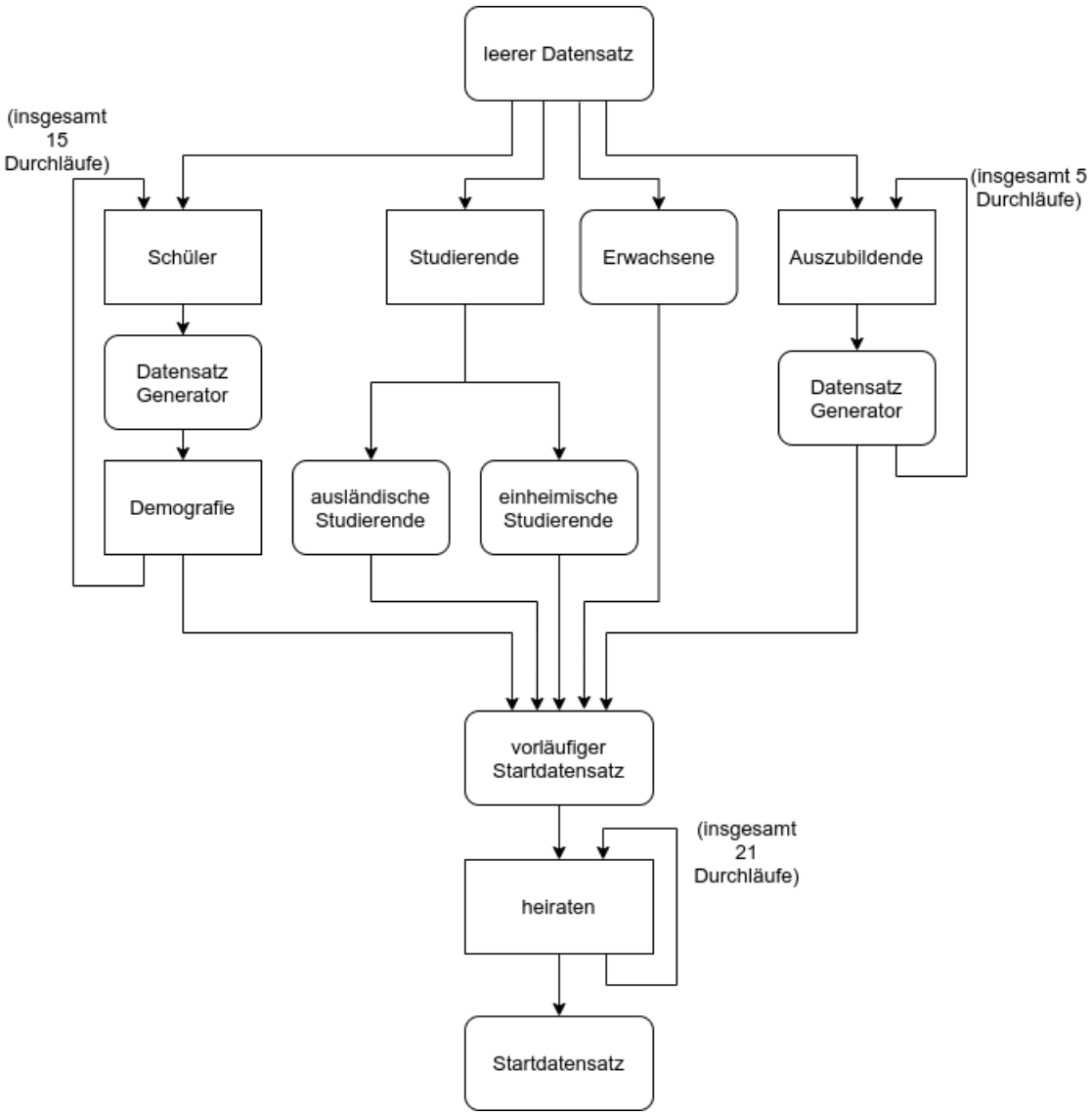

Abbildung 5: Erzeugung des Startdatensatzes

1.3.5 Generierung eines Startdatensatzes

Der Ablauf der Generierung wird in Abbildung 5 dargestellt. Für jede Kategorie wird ein eigener Startdatensatz erzeugt, welcher dann zu einem Gesamtdatensatz zusammengeführt wird. Die einzelnen Generatoren arbeiten daher unabhängig voneinander. Auf diese wird im Folgenden eingegangen.

Generierung der aktuellen Schüler

Zur Generierung der Schülerschaft wird der bereits beschriebene Datensatzgenerator verwendet (Abschnitt 1.3.1). Dieser erzeugt iterativ die Schülerkohorten der letzten 15 Jahre (vor dem Startdatum).

Der Zeitraum wurde gewählt, damit im Startdatensatz auch Schüler enthalten sind, die sich in der letzten (13.) Klasse befinden und mehrfach einen Jahrgang wiederholt haben.

Da für die Schüler keine erwartete, sondern eine benötigte Abschlussdauer simuliert wird, bedarf es zusätzlich eines Demografie-Moduls. Für alle Schüler wird daher die Wiederholung eines Jahrgangs sowie Auswanderung und Tod simuliert. So wird eine feste Quote von 1.4% aller Schüler zurückgestuft (Statistisches Bundesamt 2021c). Die Auswahl erfolgt gleichverteilt über alle Schüler ausgenommen der neuen Kohorte.

Auch die Auswanderung wird gleicherweise mit einer festen Quote von 0.2% simuliert. Mithilfe der Sterbewahrscheinlichkeit nach Alter wird der Tod simuliert. Schüler, die auswandern oder sterben, werden als abgeschlossen markiert und bleiben im Startdatensatz enthalten.

Nachdem alle Schülerkohorten erzeugt wurden, werden alle Personen aus dem entstandenen Datensatz gelöscht, die ihren Abschluss erreicht haben. Diese Personen sind keine Schüler mehr und werden bei der Generierung des Startdatensatzes an anderer Stelle simuliert.

Generierung der aktuellen einheimischen Studierenden

Für die Studierenden (ausländische und einheimische) wurde eine Tabelle zusammengestellt, welche die Zahl der aktuell real gemeldeten Studierenden an allen Hochschulen in Deutschland beinhaltet. Die Zahl der einheimischen Studierenden für den Startdatensatz richtet sich daher nach der Anzahl der aktuell in Deutschland gemeldeten Studierenden. Das Geschlecht wird dabei als gleichverteilt angenommen.

Für den Wohnort wird angenommen, dass Studierende in der Nähe der Hochschule wohnen. Daher wird ihnen die AGS der Hochschule zugewiesen, an der sie studieren.[22] Die Hochschule wird proportional zur aktuellen gemeldeten Studierendenzahl ausgewählt. Die Studierenden erhalten dann eine Postleitzahl innerhalb der AGS proportional zur Einwohnerzahl des Postleitzahlbezirks.

Für die Bestimmung des Geburtsortes wird angenommen, dass Studierende im gleichen Bundesland studieren, wo sie auch geboren sind. Sie sind daher für ihr Studium nicht weit weggezogen. Als Geburtsort wird wieder der Gemeindename verwendet. Dieser wird nach Bundesland und proportional zur Gemeindegröße zufällig gewählt.

Der Geburtstag wird nach dem Alter der Studierenden bestimmt. Das Geburtsjahr ergibt sich aus der Altersverteilung (Tabelle 4; Statistisches Bundesamt 2021c). Die Bestimmung von Tag und Monat erfolgt nicht Kohorten gebunden. Stattdessen werden sie gleichverteilt aus der Liste aller möglichen Tage des Geburtsjahres gezogen.

[22] Die Annahme wurde getroffen, da sie zu einer Verschlechterung der Linkage-Qualität führt.

Alter	Anteil
18–19	10
20–29	73
30–39	17

Tabelle 4: Altersverteilung der Studierenden (in %)

Der Name der Studierenden wird zufällig nach Geschlecht aus einer kommerziellen Datenbank gezogen. Die Studiendauer wird nach Abschnitt 1.3.4 bestimmt.

Als letztes wird das aktuelle Studienjahr festgelegt. Dabei wird angenommen, dass zu Beginn des Studiums alle Studierenden mindestens 18 Jahre alt waren. Innerhalb der so resultierenden möglichen Studienjahre wird für jeden Studierenden gleichverteilt ein aktuelles Studienjahr gewählt – daher eine Zahl im Bereich $[1; min(Geburtsjahr - 18, Studiendauer)]$.

Generierung der aktuellen ausländischen Studierenden

Die Generierung der ausländischen Studierenden erfolgt ähnlich zu den einheimischen Studierenden. Die Anzahl wird nach der aktuell gemeldeten Zahl ausländischer Studierender an den deutschen Hochschulen bestimmt. Auch das Geschlecht wird als gleichverteilt angenommen und der Wohnort wird gleich zu den einheimischen Studierenden erstellt (Wohnort in der Nähe der Hochschule).

Der Geburtstag wird auch gleich bestimmt, jedoch mit einer anderen Altersverteilung. So wird das Alter in diesem Fall gleichverteilt zwischen 18 und 33 Jahren gewählt.

Die kommerzielle Namensdatenbank erlaubt eine genauere Bestimmung von Namen nach Herkunftsland. Daher wird für jeden ausländischen Studierenden zuerst das Herkunftsland bestimmt. Dies erfolgt proportional zu einer Liste der häufigsten Herkunftsländer von ausländischen Studierenden in Deutschland (Statistisches Bundesamt 2021c). Anhand des Herkunftslandes wird dann der Name zufällig ausgewählt.

Auch die Bestimmung der Studiendauer und des aktuellen Studienjahres erfolgt gleich zu den einheimischen Studierenden (Studienjahr im Bereich $[1; min(Geburtsjahr - 18, Studiendauer)]$; Studiendauer nach Abschnitt 1.3.4).

Generierung der Erwachsenen

Es wird angenommen, dass 10% jeder Geburtskohorte im Verlauf ihres Lebens als Erwachsene in das Schülerregister gelangen.[23] Dabei werden nur die Geburtskohorten gewählt, die im Startjahr zwischen

[23]Leider gibt es hierzu keine bestätigten Zahlen. Es handelt sich daher um eine Schätzung.

30 und 70 Jahre alt sind. Nach der Geburtenzahl getrennt nach Geschlecht gelangen dann jeweils 10% der Männer und Frauen in das Register. Aus dieser Annahme ergibt sich bereits Geschlecht und Geburtsjahr. Geburtsmonat und -tag werden dann gleichverteilt aus der Liste der möglichen Tage des Geburtsjahres gezogen.

Der Wohnort ergibt sich über die Adressverteilung der Bundesrepublik. Die Bestimmung erfolgt gleich zu den Schülern. Daher wird erst das Bundesland, dann die AGS und schließlich die Postleitzahl bestimmt – alles proportional zur Einwohnerzahl.

Der Name wird zufällig aus einer kommerziellen Datenbank nach Geschlecht gezogen. Alle Erwachsenen sind Einheimische. Der Geburtsort ist daher in Deutschland. Dieser ergibt sich aus dem Namen des aktuellen Wohnortes, der bereits proportional verteilt ist. Der Zusammenhang zwischen Wohnort und Geburtsort hat dabei keinen Einfluss auf eine Record-Linkage Methode und ist somit unproblematisch.

Generierung der aktuellen Auszubildenden

Für die Auszubildenden wurde eine abgeänderte Form des Datensatzgenerators für Schüler verwendet (Abschnitt 1.3.1). Dieser unterscheidet sich in mehreren Punkten:

1. Die Zahl der Auszubildenden wird auch nach der Größe der Geburtskohorte bestimmt. Dabei wird jedoch zwischen Haupt-/Realschülern und Abiturienten unterschieden. Somit wird zwischen Personen die zum Jahresanfang des Ausbildungsbeginns 15 bzw. 16 Jahre und 18 bzw. 19 Jahre alt sind unterschieden. Die Zahl der Auszubildenden ergibt sich aus den Abschlusswahrscheinlichkeiten und der entsprechenden Wahrscheinlichkeit, eine Ausbildung zu beginnen. Diese beträgt 10% für Abiturienten und 26.5% für Haupt-/Realschüler.[24] Der entsprechende Anteil der Geburtskohorte ergibt dann die Zahl der Auszubildenden. Die Bestimmung des Geburtsmonats und -tages bleibt unverändert.
2. Die Verteilung des Geschlecht ist 57.7% Männer zu 42.3% Frauen. Diese ergibt sich aus der Wahrscheinlichkeit eine Ausbildung anzufangen nach Geschlecht.[25]
3. Mehrlinge werden generiert, die Verbindung wird jedoch getrennt.
4. Die Dauer der Schulzeit wird mit der Ausbildungsdauer ersetzt (siehe Abschnitt 1.3.4).

[24] Die genauen Zahlen werden in Abschnitt 1.3.1 und 1.3.6 behandelt. Aus den Zahlen folgt: 33% einer Kohorte machen kein (Fach-)Abitur und 6.5% beginnen danach keine weitere Qualifikation. 26.5% der Haupt-/Realschüler machen daher eine Ausbildung ($33\% - 6.5\% = 26.5\%$). 36.5% einer Kohorte machen insgesamt eine Berufsausbildung und somit 10% der Abiturienten ($36.5\% - 26.5\% = 10\%$).

[25] Aus der Studienanfängerquote ergibt sich, dass 53% der Männer und 61% der Frauen studieren (Statistisches Bundesamt 2022b). Darüber hinaus machen 5.1% der Männer und 8.3% der Frauen einen Abschluss ohne weitere Qualifikation. Die Geschlechterverteilung lässt sich über die Restmengen berechnen.

5. Es wird angenommen, dass Auszubildende einmal im Laufe ihrer Ausbildung umziehen. Daher wird auch die Dauer bis zu diesem Umzug festgelegt. Diese wird gleichverteilt aus dem Intervall $[1; Ausbildungsdauer]$ gewählt.[26]

Der abgewandelte Generator wird für die letzten fünf Jahre vor dem Startjahr ausgeführt (maximale Ausbildungsdauer). Die Dauer bis zum Umzug wird dabei für jedes Jahr herunter gezählt. Alle so erzeugten Auszubildenden gelangen in den Startdatensatz.

Erstellung der aktuellen Ehepaare

Zur Simulation der Ehepaare wird die Heiratsprozedur (Abschnitt 1.3.3) für die letzten 21 Jahre (vor dem Startdatum) ausgeführt. Das Alter der Schüler wird dabei zurückgerechnet. Die Prozedur wird daher so ausgeführt, dass die Heiratswahrscheinlichkeit verwendet wird, die ein Record vor j Jahren hatte ($j = [0; 20]$). Auch das Alter der Ehepartner wird gleichermaßen zurückgerechnet. Wurden alle Ehen erstellt, so ist der Startdatensatz fertig generiert.

1.3.6 Generierung eines Folgejahres

Ausgangspunkt für ein Folgejahr ist immer eine Kopie des Datensatzes des vorherigen Jahres. Ist dies das erste Jahr der Simulation, so wird der Startdatensatz verwendet, welcher vorher die Fehlerprozedur mit der aktuellen Fehlerquote durchlaufen hat. Darauf folgend wird das Verhalten der Personen über das Jahr simuliert sowie neue Personen generiert. Dies erfolgt über eine Reihe von Veränderungen an der Datensatzkopie.

Eine Übersicht über die Veränderungen, die durch ein Folgejahr entstehen, wird in Abbildung 6 dargestellt. Alle Schritte werden im Folgenden genauer beschrieben. Die Reihenfolge richtet sich dabei nach der implementierten Reihenfolge.

[26] Es war nicht möglich innerhalb des zeitlichen Rahmens geeignete Zahlen zu finden. Die Annahme eines Umzug erscheint jedoch handlungstheoretisch logisch. Mithilfe des ersten Gehalts werden viele Auszubildende aus dem Elternhaus ausziehen. Für weitere Umzüge ist die Zeit der Ausbildung jedoch meist zu kurz.

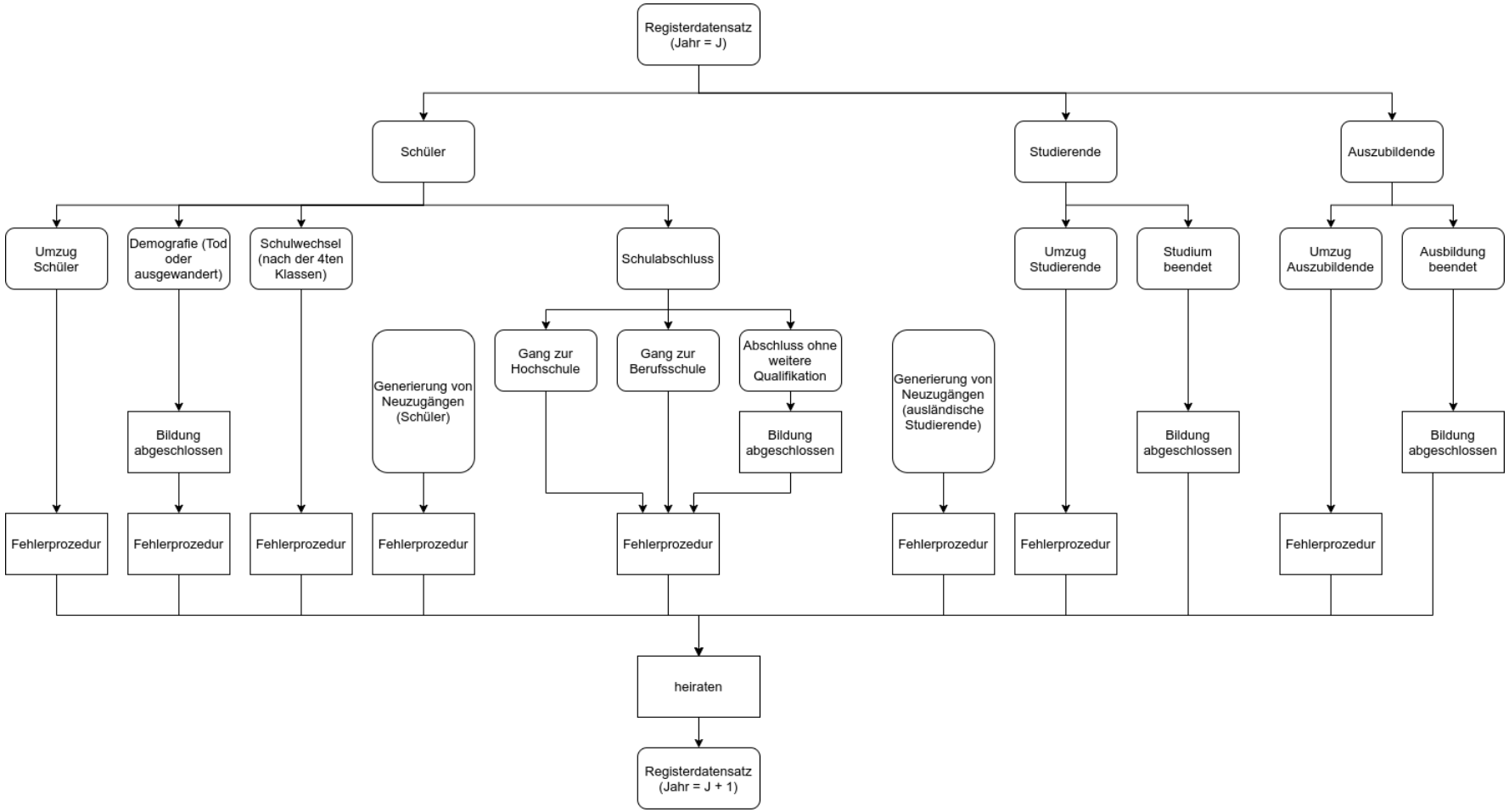

Abbildung 6: Vereinfachte Darstellung der Erzeugung eines Folgejahres. Einige Pfade fehlen in der Abbildung, z. B. ein direkter Pfad vom Ausgangsdatenbestand in das Folgejahr, falls keine Veränderungen stattfinden. Ebenso sind mehrere Ereignisse pro Jahr möglich.

Demografie

Von diesem Modul sind ausschließlich Schüler betroffen. Nur sie haben eine benötigte Abschlussdauer, sodass ein frühzeitiges Ausscheiden aus dem Schulsystem simuliert werden muss. Da in Deutschland die Schulpflicht gilt, sind die einzigen Ausfallmöglichkeiten Tod und Auswanderung. Die Auswanderung erfolgt gleichverteilt über die gesamte Schülerschaft mit einer festen Quote von 0.2%. Die Sterbewahrscheinlichkeit ist altersabhängig und berechnet sich nach einer Sterbetafel (Statistisches Bundesamt 2022a). Alle betroffenen Schüler werden als abgeschlossen (Erwachsene) markiert und sind somit nicht weiter Schüler.

Umzug Studierende

Die Zahl der Umzüge bei Studierenden basiert auf der Altersverteilung in der Umzugsstudie 2018 (Deutsche Post Adress GmbH 2018). Daraus folgt die Annahme, dass Studierende bis zu ihrem 24. Lebensjahr mit einer 16%-Wahrscheinlichkeit und danach mit einer 8%-Wahrscheinlichkeit umziehen. 3/6 der Umzüge geschehen innerhalb einer Postleitzahl, 2/6 innerhalb der AGS (wechsel der Postleitzahl) und 1/6 außerhalb der AGS. Verlässt ein Studierender seine aktuelle AGS, so wird angenommen, dass er in die Nähe einer Hochschule ziehen wird. Daher erhalten die Betroffenen die AGS einer Hochschule proportional zur Anzahl der dort gemeldeten Studierenden. Anschließend durchlaufen alle umgezogenen Studierenden die Fehlerprozedur.[27]

Umzug Auszubildende

Da Auszubildene im Laufe ihrer Ausbildung einmal umziehen und das Umzugsjahr bereits gesetzt ist, wird an dieser Stelle nur geprüft, ob dieses Jahr erreicht ist. Falls ja, so zieht der Auszubildende innerhalb seiner aktuellen AGS um. Alle Betroffenen durchlaufen dann die Fehlerprozedur.

Nächstes Schuljahr

Für alle Personen im Register wird nun das Schuljahr (bzw. Ausbildungsjahr oder Studienjahr) um eins erhöht. Da Schüler eine benötigte Abschlussdauer haben, muss für sie auch eine Rückstufung simuliert werden. Es wird angenommen, dass über alle Jahrgänge 1.4% der Schüler zurückgestuft werden. Für die Betroffenen wird dann der Zähler wieder um eins verringert.

[27] Die Datenbasis besteht aus Zahlen für Umzüge von Studierenden in Freiburg (Huinink und Kley 2011).

Schulabschluss

Haben Schüler ihre benötigte Abschlussdauer erreicht,[28] so wird an dieser Stelle über die weitere Bildungskarriere entschieden. Nach der Entscheidung und einem etwaigen Umzug durchlaufen alle Betroffenen die Fehlerprozedur. Darüber hinaus werden Mehrlingsbindungen nun aufgelöst, da die Betroffenen nun heiraten können.

Besuch einer Hochschule: Es wird angenommen, dass 53% der Männer und 61% der Frauen mit einem (Fach-)Abitur eine Hochschule besuchen (13 Jahre Schulzeit).[29] Die Studiendauer wird nach der in Abschnitt 1.3.4 vorgestellten Verfahrensweise berechnet.

Darüber hinaus wird angenommen, dass 3/6 zu Hause wohnen bleiben, 2/6 bundesweit eine Hochschule suchen und 1/6 zur nächst gelegenen Hochschule ziehen.[30] Für die letzten beiden Fälle wird angenommen, dass die Studierenden in die Nähe der Hochschule ziehen und damit die gleiche AGS erhalten. Die Zuweisung einer Hochschule für bundesweit suchende Studierende erfolgt proportional zur Zahl der Studierenden. Die Zuweisung einer in der Nähe gelegenen Hochschule erfolgt über eine iterative Streichung der letzten Stelle der AGS von Hochschule und Studierenden. Wurden auf diesem Weg mindestens fünf Hochschulen identifiziert, wird proportional zur Zahl der Studierenden einer dieser Hochschulen ausgewählt.

Abschluss ohne weitere Qualifikation: Es wird angenommen, dass 4.1% der Männer und 8.3% der Frauen keiner weiteren schulischen Qualifikation nachgehen (Bundesinstitut für Berufsbildung 2021: 279). Voraussetzung ist in diesem Fall ein Haupt-/Realschulabschluss (10 Jahre Schulzeit). Die Betroffenen werden dann als Erwachsene klassifiziert.

Besuch einer Berufsschule: Für die restlichen Personen wird angenommen, dass sie eine Berufsschule besuchen. Sie erhalten eine Ausbildungsdauer (siehe Abschnitt 1.3.4) und ein Umzugsdatum. Die Dauer bis zum Umzug wird gleichverteilt aus dem Intervall $[1; \textit{Ausbildungsdauer}]$ gewählt. Alle Auszubildenden bleiben somit zunächst zu Hause wohnen.

Heiraten

Eine Ehe wird für alle nicht-Schüler über 18 Jahre nach Abschnitt 1.3.3 simuliert. Alle betroffenen Personen durchlaufen die Fehlerprozedur.

[28] Schüler können auch die Schule abschließen, wenn sie mindestens 16 Jahre alt sind und mindestens viermal einen Jahrgang wiederholt haben. Dieser Weg wurde für extreme Ausreißer konzipiert und ist damit nicht die übliche Variante. Die betroffenen Schüler werden dann so behandelt, als hätten sie ihren zu erwartenden Abschluss erreicht.

[29] Es handelt sich hierbei um die gerundete Studienanfängerquote der letzten Jahre (Statistisches Bundesamt 2022b).

[30] Da keine geeigneten Zahlen zum initialen Wohnortswechsel vorlagen, wurden erneute die Zahlen der Umzügen in Freibug verwendet (Huinink und Kley 2011).

Umzug Schüler

Es wird angenommen, dass 3% der Schüler innerhalb einer AGS, 0.5% innerhalb eines Bundeslandes und weitere 0.5% außerhalb eines Bundeslandes umziehen. Die Auswahl der Schüler erfolgt als eine gleichverteilte feste Quote unabhängig von anderen Merkmalen. Da Mehrlinge immer gemeinsam umziehen, sind Geschwister von einem Umzug betroffen, auch wenn sie nicht für einen Umzug ausgewählt wurden. Alle betroffenen Schüler durchlaufen die Fehlerprozedur.

Schulwechsel

Da über die Zahl der Schulwechsel keine genauen Daten vorlagen, wird nur der sichere Schulwechsel von der Grundschule in die weiterführende Schule simuliert. Dieser betrifft alle Schüler, die neu in die fünfte Klasse gelangt sind. Alle betroffenen Schüler durchlaufen die Fehlerprozedur. Für die übrigen Schulwechsel besteht die Annahme, dass ein Umzug nicht zwangsläufig mit einem Schulwechsel einhergeht. Die nicht direkt simulierten Schulwechsel werden daher näherungsweise durch die Überschätzung der umzugsbedingten Schulwechsel abgedeckt.

Generierung von Neuzugängen

Die neue Schülerkohorte wird nach Abschnitt 1.3.1 erstellt. Darüber hinaus gelangen jährlich 82,415 ausländische Studierende in das Schülerregister (Statistisches Bundesamt 2021c). Die Generierung richtet sich nach der Vorgehensweise aus Abschnitt 1.3.5 und gestaltet sich wie folgt:

Das Geschlecht der ausländischen Studierenden ist gleichverteilt. Darüber hinaus wird angenommen, dass sie in die Nähe einer Hochschule ziehen. Daher erhalten sie die AGS einer Hochschule proportional zur Anzahl der dort ausländischen Studierenden. Ihr Geburtsort wird aus der Liste der Geburtsorte für Migraten zufällig ausgewählt. Die Studiendauer wird nach Abschnitt 1.3.4 berechnet. Die Herkunftsländer verteilen sich nach der Liste der häufigsten Länder von ausländischen Studierenden. Der Name wird dann wieder anhand des Herkunftslandes und des Geschlechts bestimmt. Das Geburtsjahr wird nach einer Verteilung für Alter und Geschlecht von Ausländern bei Studienbeginn bestimmt (Statistisches Bundesamt 2021c). Das vollständige Geburtsdatum wird dann durch eine gleichverteilte Auswahl aller möglichen Tage des Geburtsjahres festgelget.

Nachdem alle Neuzugänge generiert wurden, durchlaufen sie die Fehlerprozedur. Danach werden sie dem Datensatz hinzugefügt. Nach diesem Schritt ist der Datensatz eines Folgejahres simuliert.

Kapitel 2

Record-Linkage Methoden zur Verknüpfung der simulierten Daten

2.1 Einführung in Record-Linkage

2.1.1 Maßzahlen zur Evaluierung

Um Record-Linkage Verfahren miteinander vergleichen zu können, bedarf es einiger Gütemaße für das erfolgte Linkage. Da es sich bei den simulierten Daten um Goldstandard-Daten handelt, können die klassifizierten Matches mit dem wahren Matchstatus verglichen werden. Daraus resultiert eine Kreuztabelle mit allen möglichen Klassifikationen für das Linkage (Abbildung 7). Vier verschiedene Kategorien lassen sich somit unterscheiden:

- **True Positves (TP)**: Dies sind alle zusammengehörigen Records, die auch als Matches klassifiziert wurden.
- **False Positives (FP)**: Dies sind nicht zusammengehörige Records, die dennoch als Match klassifiziert wurden. Diese Zahl sollte daher möglichst klein sein.
- **True Negatives (TN)**: Dies sind alle nicht zusammengehörigen Records, die auch nicht als Match klassifiziert wurden.
- **False Negatives (FN)**: Dies sind alle zusammengehörigen Records, die nicht als Match klassifiziert wurden. Auch diese Zahl sollte daher möglichst klein sein.

Anhand dieser vier Kategorien können nun einige Maßzahlen gebildet werden. Dabei ist beim Record-Linkage zentral, dass die Größe der Kategorien nicht gleich verteilt ist. Stattdessen ist die

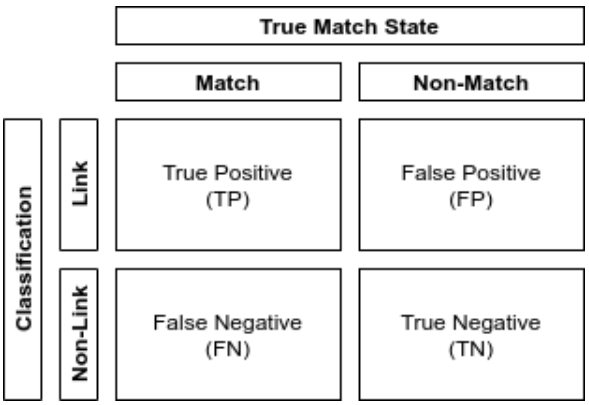

Abbildung 7: Klassifikationstabelle möglicher Zuordnung von Fällen (nach Christen 2012: 166)

Zahl der true Negative Matches meist deutlich größer als die übrigen Kategorien. Dies lässt sich leicht demonstrieren. So ist die Summe der vier Kategorien gleich der Anzahl aller Vergleiche (Talburt 2011: 13). Daher

$$TP + FP + TN + FN = |A| \times |B|, \tag{2.1}$$

wobei $|A|$ und $|B|$ jeweils die Zahl der Records in Datensatz A bzw. B ist. Da jedem Record aus A nur ein Record aus B zugeordnet werden kann, ist die Zahl wahren der Matches gleich der Länge des kleineren Datensatzes, sodass

$$TP + FN \leq \min(|A|, |B|) \tag{2.2}$$

gilt. Dies gilt auch für die Summe der möglichen Matches:

$$TP + FP \leq \min(|A|, |B|). \tag{2.3}$$

Nun wird der schlechtest mögliche Fall angenommen: Die Zahl der Klassifikationen ist gleich der Anzahl Records im kleineren Datensatz und die Zahl der true Positives ist gleich null. Für diesen Fall ergibt sich

$$TN_{max} = |A| \times |B| - 2 \times \min(|A|, |B|).$$

Setzt man nun für A und B die ersten beiden Datensätze der Simulation ein, so lautet die Zahl der maximalen true negatives:

$$TN_{max} = 18,914,154 \cdot 19,824,463 - 2 \cdot 18,914,154 = 3.75 \cdot 10^{14}. \tag{2.4}$$

Daraus ergibt sich, dass die Zahl der true Negatives in keiner Maßzahl verwendet werden sollte, da das Gewicht dieser Zahl so groß ist, dass eine Maßzahl nur noch schwer zu interpretieren ist. Für Record-Linkage werden daher vor allem zwei Maßzahlen verwendet (Christen 2012: 167):

$$Precision = \frac{TP}{TP + FP} \tag{2.5}$$

und

$$Recall = \frac{TP}{TP + FN}. \tag{2.6}$$

Sowohl Precision als auch Recall lassen sich leicht durch die Konfiguration einer Linkage-Methode beeinflussen (Christen, Ranbaduge et al. 2020: 62 f.). Wird bspw. ein Schwellenwert für die Klassifikation eines Matches sehr hoch gesetzt, so werden nur sehr sichere Matches klassifiziert, die wiederum meist true Positives sind. Dementsprechend würde ein sehr hoher Wert in der Precision erreicht werden. Wird der Schwellenwert hingegen gesenkt, so werden potenziell mehr Matches klassifiziert, was die Zahl der false Negatives senkt. Dadurch würde der Recall steigen.

Aus dem vorangegangenen Beispiel lässt sich entnehmen, dass die Werte nicht einzeln, sondern zusammen interpretiert werden müssen. Würde der Schwellenwert erhöht und eine hohe Precision erreicht, so ergibt sich gleichzeitig ein niedriger Recall. Dies gilt auch für den umgekehrten Fall, sodass bei der Senkung des Schwellenwertes auch die Precision sinkt. Mithilfe des F-Scores können Precision und Recall in einen Wert zusammengefasst werden. Der F-Score ist ursprünglich über das harmonische Mittel als

$$F = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} = \frac{2 \cdot TP}{FP + FN + 2TP} \tag{2.7}$$

definiert (Christen, Ranbaduge et al. 2020: 63).

Diese ursprüngliche Form des F-Scores ist jedoch angreifbar über die Zahl der klassifizierten Matches (Hand und Christen 2018). Beim Vergleich zweier Linkage-Methoden derselben beiden Datensätze ist daher der F-Score nicht vergleichbar, wenn die erste Methode bspw. 50,000 und die zweite 80,000 Matches klassifiziert. Hand und Christen (2018: 546) schlagen daher vor, das arithmetische Mittel zur Bildung des F-Scores zu verwenden. Dieser wird meist als $\overline{F}$ oder auch F_2 bezeichnet und wird definiert als

$$\overline{F} = \frac{Precision + Recall}{2}. \tag{2.8}$$

Von soziologischer Relevanz sind besonders die Auswirkungen von false Positves und false Negatives auf die spätere Auswertung der gelinkten Daten. Für ein Schülerregister lassen sich diese Auswirkungen mit einer Betrachtung der Bildungslaufbahn beispielhaft verdeutlichen:

Die beim Linkage verwendeten QIDs sind nahezu ausschließlich demografische Angaben einer Person. Kommt es zu einem false Positive Match, so werden zwei Personen gelinkt, die einen ähnlichen demografischen Hintergrund haben – z. B. in derselben Gemeinde am gleichen Tag mit demselben Nachnamen geboren wurden. Auch wenn der demografische Hintergrund durchaus Auswirkungen auf die Bildungslaufbahn haben wird, so ist dennoch zu erwarten, dass dadurch seltene Bildungslaufbahnen fälschlicherweise entstehen. Damit werden diese Bildungslaufbahnen bei einer hohen Zahl von false Positives insgesamt überschätzt.

False Negative Matches hingegen bewirken meist das Gegenteil. So sind seltene Bildungslaufbahnen häufiger von Umbrüchen geprägt. Dies bewirkt, dass die zugehörigen Records auch häufiger im Register verändert werden. Da jede Veränderung die Wahrscheinlichkeit für mindestens eine Falscheingabe erhöht, steigt auch die Wahrscheinlichkeit für einen false Negative Match. Daher würden seltene Bildungslaufbahnen bei einer hohen Zahl von false Negatives unterschätzt.

Ein unterschiedlicher Fokus auf false Positves und false Negatives bewirkt einen unterschiedlichen Fokus entweder auf Precision (Reduktion von false Positives) oder Recall (Reduktion von false Negatives). Da für den Normalbetrieb einer Administration false Positives meist die höheren Kosten erzeugen, liegt der Fokus meist auf einer hohen Precision (Schnell 2022: 21). Dementsprechend werden auch die hier verwendeten Record-Linkage Methoden – sofern es möglich ist – so konfiguriert, dass sie zu einer möglichst hohen Precision führen.

Das vorangegangene Beispiel lässt erkennen, warum eine gemeinsame Analyse von Precision und Recall sinnvoll ist. Nur durch gemeinsame Analyse kann gezeigt werden, dass das Ziel einer möglichst hohen Precision auch erreicht wurde. Dies kann nicht durch den F-Score erlangt werden, da er Precision und Recall zusammenführt.

2.1.2 Blocking

Wie Gleichung (2.4) eindrücklich zeigt, ist die Zahl aller möglichen Vergleiche sehr groß, wobei die Zahl der true Negatives fast genau so groß ist. Die allermeisten Vergleiche, die bei einem direkten Linkage zweier Datensätze entstehen, sind somit true Negatives. Da jeder zusätzliche Vergleich sowohl Rechenzeit als auch Speicherplatz bedarf, empfiehlt es sich bei größeren Datensätzen nicht alle Records miteinander zu vergleichen. Dieses Vorgehen wird als *Blocking* oder auch *Indexing* bezeichnet (Christen 2012: 69). Welche Records miteinander verglichen werden, wird über eine sogenannte *Blockingregel* bestimmt.

Die einfachste und wohl auch häufigste Form des Blockings ist das exakte Blocken. Dabei werden nur Records miteinander verglichen, die auf einer Menge an QIDs exakt übereinstimmen (*Blockingvariable*). Innerhalb der Blöcke werden wieder aller Records miteinander verglichen (Christen, Ranbaduge et al. 2020: 103). Unter der Annahme einer Gleichverteilung der Anzahl der Elemente in den Blöcken lässt sich die Zahl der Vergleiche c für zwei Datensätze mit m und n Records und b Blöcken als

$$c = b \cdot \left(\frac{n}{b} \cdot \frac{m}{b}\right) = \frac{m \cdot n}{b} \tag{2.9}$$

berechnen (Christen 2012: 81).

Aus der Gleichung folgt trivialerweise, dass die Zahl der Vergleiche sinkt, je mehr Blöcke vorhanden sind. Die endgültige Zahl der Vergleiche wird durch die Anzahl der Ausprägungen, der Varianz und die Entropie[31] der verwendeten Variablen bestimmt.

Das Problem beim Blocken besteht darin, dass true Matches auch in den gleichen Block eingeordnet werden müssen. Ist in einer der Blockingvariablen ein Datenfehler, so werden true Matches gar nicht erst miteinander verglichen und können so nicht richtig klassifiziert werden. Daher wurden einige andere Blockingverfahren entwickelt, die auch mit Datenfehlern in den Blockingvariablen umgehen können.

Ein Beispiel hierfür ist das Sliding Window Verfahren. Für numerische Werte werden nicht nur Records mit dem gleichen numerischen Wert verglichen, sondern auch mit $\pm w$ Werten über bzw. unter dem Ursprungswert. Für z. B. $w = 2$ und dem Geburtsjahr als numerischen Wert, werden für ein Record im Datensatz A mit dem Geburtsjahr 2001, alle Records im Datensatz B mit einem Geburtsjahr zwischen 1999 und 2003 verglichen.

Ein sliding Window Verfahren, welches auch für Zeichenketten funktioniert, ist *Sorted Neighbourhood* (auch bekannt als k-Nearest-Neighbours) (Christen 2012: 81 ff., Hernandez und Stolfo 1998). Hierbei werden alle Records anhand der Blocking-Variablen sortiert. Für den Vergleich eines Records in Datensatz A an der Stelle i dieser Sortierung, werden dann alle Records $i \pm k$ der Sortierung in Datensatz B ausgewählt und mit dem Record aus Datensatz A verglichen. Aus beiden Beispielen lässt sich erkennen, dass es einer theoretischen Begründung bedarf, warum Werte in diesem Wertebereich eine höhere Wahrscheinlichkeit für das Auffinden eines true Matches haben. Da solche Begründungen für die im sozialwissenschaftlichen Kontext verwendeten Blockingvariablen selten existieren, empfehlen sich diese Verfahren selten.[32]

Da selten Datenfehler und Veränderungen in allen QIDs eines Records vorkommen, können auch mehrere exakte Blockingregeln verwendet werden. Je nach Linkage-Methode kann dann die Klassifikation unterschiedlich erfolgen:

1. Nach jeder Blockingregel erfolgt eine Klassifikation und klassifizierte Links werden für die weiteren Blockingregeln ausgeschlossen.
2. Alle Links, die sich aus den einzelnen Blockingregeln ergeben, werden zusammengefasst und der jeweils beste Link für ein Record wird als Match klassifiziert.

[31] Die Entropie H misst den durchschnittlichen Informationsgehalt einer Zufallsvariablen X (Christen, Ranbaduge et al. 2020: 103). Sie wird definiert als

$$H(X) = -\sum^{x \in X} p(x) \cdot \log_2 p(x).$$

[32] Dies gilt vor allem für Klartextdaten. Für Daten, die mit Bloom-Filtern verschlüsselte wurden (siehe Abschnitt 2.3.3), ist *Sorted Neighbourhood* eine valide und gut funktionierende Blockingstrategie (Schnell 2016).

2.1.3 Deterministisches Record-Linkage

Die einfachste Art, zwei Merkmale miteinander zu vergleichen, ist die exakte Übereinstimmung. Daraus leitet sich die Vorgehensweise beim *deterministischen Record-Linkage* oder regelbasierten Record-Linkage ab. Zwei Records werden als Link klassifiziert, falls sie auf einer Menge von Merkmalen exakt übereinstimmen. Die Menge an Merkmalen wird allgemein als *Matchkey* bezeichnet (Herzog et al. 2007: 82).

Da ein Link beim deterministischen Record-Linkage über eine exakte (und eindeutige) Übereinstimmung erfolgt, kommt es nur sehr selten zu falsch positiven Matches. Da jeder Fehler in einem der Records eine exakte Übereinstimmung verhindert, kommt es dem gegenüber häufig zu falsch negativen Matches.

Grundsätzlich können nicht klassifizierte Matches von anderen Record-Linkage Methoden nochmals klassifiziert werden. Ein solches Vorgehen wird als *mehrstufige Record-Linkage-Strate-gie* bezeichnet. Deterministische Verfahren eigenen sich sehr gut als erster Schritt einer solchen Strategie, da sie eine hohe Precision und sehr geringe Laufzeiten haben – insbesondere wenn die vollständige Menge an Merkmalen als Matchkey verwendet wird.[33] Deterministisches Record-Linkage fungiert dann als Filter. Indem alle Matches, die auf der vollständigen Menge an Merkmalen übereinstimmen, aussortiert werden, kann die Zahl der Vergleiche für nachfolgende Methoden verringert werden. Dies reduziert nicht nur die Rechenzeit für diese nachfolgenden Methoden, sondern verbessert auch ihre Ausgangslage. So können bspw. weniger strenge Blocking-Regeln für Methoden mit einem höheren Rechenaufwand verwendet werden (siehe Abschnitt 2.1.2).

2.1.4 Probabilistisches Record-Linkage

Als *probabilistisches Record-Linkage* wird eine weitere Art von Record-Linkage Methoden verstanden, die besonders auf der Arbeit von Fellegi und Sunter (1969) aufbaut. Mit dem Ergebnis solcher Methoden kann immer eine *Match-Wahrscheinlichkeit* p berechnet werden, mit der zwei Records zur selben Entität gehören. Ein Link wird dann als Match klassifiziert, falls p über einem Schwellenwert T liegt (formal: $p \geq T$).[34] Daher können auch Ähnlichkeitsmaße (Abschnitt 2.1.5) verwendet werden, die in die Berechnung von p einfließen.

Eine vertiefte Beschreibung des probabilistischen Record-Linkages findet sich in Abschnitt 2.3.2.

[33] Deterministisches Record-Linkage lässt sich in der Praxis über eine „Join“ beider Datensätze erzielen. „Gejoint“ wird auf der Menge an Feldern, die den Matchkey bilden. In vielen Implementationen eines „Data-Frames“ (z. B. data.table (R) oder pandas (Python)) erfolgt diese Operation sehr schnell – wenn die Zahl der Matches für ein Matchkey nicht groß ist.

[34] Bei realen Linkage-Projekten ist ein zweiter Schwellenwert T_p für die Klassifizierung als partielle Matches möglich. Alle Records mit $T_p \leq p < T$ werden dann nochmal manuell klassifiziert (*Clerical Review*) (Herzog et al. 2007: 85 f.).

2.1.5 Ähnlichkeitsmaße

Die einfachste Form des Vergleichs ist die exakte Übereinstimmung – wie sie bspw. beim exakten Blocking verwendet wird. Aufgrund von Datenfehlern empfiehlt sich jedoch häufig, die Ähnlichkeit zweier Werte anhand eines *Ähnlichkeitsmaßes* zu berechnen. Das Ergebnis ist dann eine Ähnlichkeit sim der beiden Werte, wobei $0 \leq sim \leq 1$. Ist $sim = 1$, so stimmen die Werte exakt miteinander überein.

Auch für die Verwendung von Ähnlichkeitsmaßen muss theoretisch begründet werden können, warum ähnliche Werte eine höhere Wahrscheinlichkeit für ein Match haben. Eine solche Begründung ist für Daten mit Tippfehlern gegeben. Daher werden im Folgenden einige Vergleichsfunktionen für Zeichenketten vorgestellt.

Hamming-Distanz

Zeichenketten können sehr leicht miteinander verglichen werden, indem alle Zeichen gezählt werden, die an der gleichen Position nicht miteinander übereinstimmen. Dieses Verfahren wurde erstmalig von Hamming (1950) vorgeschlagen. Für zwei Zeichenketten x und y der Länge n definiert sich die *Hamming-Distanz* als

$$h(x, y) = |\{i \in 1, ..., n | x_i \neq y_i\}|, \tag{2.10}$$

wobei x_i und y_i jeweils die Zeichen an der Stelle i der beiden Zeichenketten sind. Ursprünglich wurde die Hamming-Distanz für Bit-Vektoren mit einheitlicher Länge definiert. Indem fehlende Zeichen als nicht übereinstimmend gezählt werden, kann die Hamming-Distanz aber auch für unterschiedlich lange Zeichenketten berechnet werden.

Da die Hamming-Distanz die Zahl der Unterschiede wiedergibt, beträgt der Wertebereich $0 \leq h(x, y) \leq max(|x|, |y|)$, wobei $|\cdot|$ die Länge einer Zeichenkette angibt. Für die Zeichenkette x und y der Länge $|x|$ und $|y|$, kann die normalisierte Hamming-Distanz als

$$sim_{Hamming}(x, y) = \frac{h(x, y)}{max(|x|, |y|)} \tag{2.11}$$

definiert werden (Pilar Angeles und Espino-Gamez 2015: 64).

Levenshtein-Distanz

Die Hamming-Distanz ist sehr anfällig für Verschiebungen. So kann ein zusätzliches Zeichen am Anfang einer Zeichenkette dazu führen, dass zwei ansonsten gleiche Zeichenketten als wenig ähnlich klassifiziert werden. Dieses Problem wird behoben, indem die benötigten Veränderungen gezählt werden, um eine

		0	1	2	3	4	5
			P	**E**	**T**	**E**	**R**
0		0	1	2	3	4	5
1	**P**	1	0	1	2	3	4
2	**E**	2	1	0	1	2	3
3	**D**	3	2	1	1	2	3
4	**R**	4	3	2	2	2	2
5	**O**	5	4	3	3	3	**3**

Abbildung 8: Editiermatrix zur Berechnung der Levenshtein-Distanz von „PETER“ und „PEDRO“ (Christen 2012: 103). Die Zahl in der rechten unteren Ecke (3) entspricht der Levenshtein-Distanz.

Zeichenkette in die jeweils andere zu transformieren. Dieses Vorgehen ist allgemein als *Editierdistanz* bekannt (Christen 2012: 103).

Die grundlegende Editierdistanz wird als *Levenshtein-Distanz* bezeichnet (Levenshtein 1966). Sie umfasst drei verschiedene Editierfunktionen: ersetzen, einfügen und löschen. Gebildet wird die Levenshtein-Distanz anhand einer Editiermatrix D (Abbildung 8). Es seien zwei Zeichenketten x und y mit den Längen $|x|$ und $|y|$ gegeben. Jede Zelle $d_{i;j}$ der Matrix D ($0 \leq i \leq |x|$ und $0 \leq j \leq |y|$) enthält den Wert für die Anzahl an Editierungen, die es benötigt, um die Zeichenkette $x[0, i]$ in die Zeichenkette $y[0, j]$ zu ändern. Dabei gibt $[\cdot, \cdot]$ den Ausschnitt aus der Zeichenkette x bzw. y an. Das Zeichen an der Stelle 0 ist immer ein leerer Wert – und somit eine Zeichenkette der Länge null. Die Levenshtein-Distanz befindet sich in der letzten Zelle der Matrix $d_{|x|;|y|}$.

Um die Editiermatrix für x und y zu erzeugen, wird zunächst eine leere Matrix initialisiert und die Zeichenketten x und y mit führenden leeren Wert eingetragen. Da die Kosten zur Transformation einer Zeichenkette s in eine leere Zeichenkette immer $|s|$ beträgt, wird die erste Zeile und Spalte mit aufsteigenden Werten initialisiert. Die anderen Zellen lassen sich dann rekursiv bestimmen. Jede Zelle wird wie folgt gefüllt (Christen 2012: 104):

- Falls $x[i] = y[j]$:

$$d_{i,j} = d_{i-1;j-1}. \tag{2.12}$$

- Falls $x[i] \neq y[j]$:

$$d_{i,j} = min \begin{cases} d_{i-1;j} + 1 & \text{wird ein Zeichen gelöscht,} \\ d_{i;j-1} + 1 & \text{ein Zeichen eingefügt oder} \\ d_{i-1;j-1} + 1 & \text{ein Zeichen ersetzt.} \end{cases} \tag{2.13}$$

Der Wertebereich der Levenshtein-Distanz ist genau wie bei der Hamming-Distanz von der Länge der Strings abhängig. Daher wird die Levenshtein-Distanz für gewöhnlich normalisiert. Für zwei Zeichenketten x und y mit der Länge $|x|$ und $|y|$ sowie der Levenshtein-Distanz $lev(x, y)$ beträgt die normalisierte Levenshtein-Distanz:

$$sim_{Levenshtein} = \frac{lev(x, y)}{max(|x|, |y|)}. \tag{2.14}$$

Q-Gramme

Ein anderer Ansatz zur Bestimmung der Ähnlichkeit zweier Zeichenketten sind *q-Gramme* (Christen 2012: 106). Dabei wird für eine Zeichenkette eine Menge an kurzen Zeichenketten der Länge q erzeugt. Für den Vergleich zweier Zeichenketten wird die Anzahl c der q-Gramme berechnet, die in beiden Zeichenketten vorkommen. Für zwei Zeichenketten x und y können so mehrere verschiedene Ähnlichkeitsmaße bestimmt werden:

$$sim_{Overlap}(x, y) = \frac{c}{min(|x|, |y|)} \tag{2.15}$$

$$sim_{Jaccard}(x, y) = \frac{c}{|x| + |y| - c}, \tag{2.16}$$

$$sim_{Dice}(x, y) = \frac{2 \cdot c}{|x| + |y|}, \tag{2.17}$$

wobei $|x|$ und $|y|$ die Anzahl an verschiedenen q-Grammen in den Zeichenketten x und y angibt.

Zur Erstellung eines q-Gramms existieren zahlreiche Möglichkeiten. Die wohl bekannteste und weit verbreitetste ist das Bigramm ($q = 2$). Die Menge an Bigrammen besteht aus allen zwei aufeinanderfolgenden Zeichen einer Zeichenkette. Für den Namen JOHN ist dies z. B. {JO, OH, HN}.

Würde der Name JOHN mit JOHNNY ({JO,OH,HN,NN,NY}) verglichen werden, so lässt sich bspw. der Dice-Koeffizient wie folgt berechnen:

$$sim_{Dice}(\{JO, OH, HN\}, \{JO, OH, HN, NN, NY\}) = \frac{2 \cdot 3}{3 + 5} = 0.75.$$

Jaro- und Jaro-Winkler-Distanz

Speziell für den Vergleich von Namen wurden mehrere Ähnlichkeitsmaße von Matthew A. Jaro und William E. Winkler im US Census Bureau entwickelt. Diese Ähnlichkeitsmaße kombinieren den Ansatz der Editier-Distanz und q-Gramme (Christen 2012: 109). Die grundlegende Funktion ist die *Jaro-*

Distanz (Jaro 1989). Für zwei Zeichenketten x und y der Länge $|x|$ und $|y|$ beträgt die Jaro-Distanz

$$sim_{Jaro}(x,y) = \frac{1}{3}\left(\frac{c}{|x|} + \frac{c}{|y|} + \frac{c-t}{c}\right), \tag{2.18}$$

wobei c die Anzahl der übereinstimmenden Zeichen ist und t die Anzahl der Vertauschungen (Yancey 2005). Ein Zeichen an der Stelle x_i ist ein übereinstimmendes Zeichen, wenn ein gleiches Zeichen y_j existiert, wobei

$$|i-j| < d, \quad 1 \leq i \leq |x|, 1 \leq j \leq |y| \tag{2.19}$$

und

$$d = \left\lfloor \frac{max(|x|,|y|)}{2} \right\rfloor. \tag{2.20}$$

Jedes Zeichen x_i kann nur mit einem anderen Zeichen y_j übereinstimmen. Alle übereinstimmenden Zeichen x_i und y_j bilden die geordneten Vektoren v_x und v_y. Über die *Hamming-Distanz* h lässt sich die Zahl der Vertauschungen als

$$t = \left\lfloor \frac{h(v_x, v_y)}{2} \right\rfloor \tag{2.21}$$

berechnen.

Eine Modifikation der *Jaro-Distanz* ist die *Jaro-Winkler-Distanz* (Winkler 1990). Da am Anfang eines Namens seltener Fehler auftauchen als am Ende, wird bei der *Jaro-Winkler-Distanz* der Anfang der Zeichenkette stärker gewichtet. Dies geschieht über eine Gewichtung der ersten $0 \leq p \leq 4$ übereinstimmenden Zeichen und einen Skalierungsfaktor k (üblicherweise 0.1). Daraus folgt:

$$sim_{\text{Jaro-Winkler}}(x,y) = sim_{Jaro}(x,y) + (1 - sim_{Jaro}(x,y)) \cdot 0.1 \cdot p. \tag{2.22}$$

Es existieren weitere Modifikationen dieser Distanz (Christen 2012 : 110). Diese sollen an dieser Stelle aber nicht weiter behandelt werden. Beim Vergleich von Namen beim Record-Linkage ist die *Jaro-Winkler-Distanz* meist die beste Wahl (Herzog et al. 2007: 131).

2.2 Preprocessing

Bevor zwei Datensätze gelinkt werden können, müssen diese in ein identisches Format übertragen werden. Dies beinhaltet eine Datenaufbereitung und Datenbereinigung (Wickham 2014). Darüber hinaus können vor dem Linkage einige Merkmale in zusätzliche Formate – wie z. B. phonetische Codierungen – überführt werden, die für eine Methode benötigt werden. Die Summe dieser Schritte wird im Allgemeinen als *Preprocessing* bezeichnet.

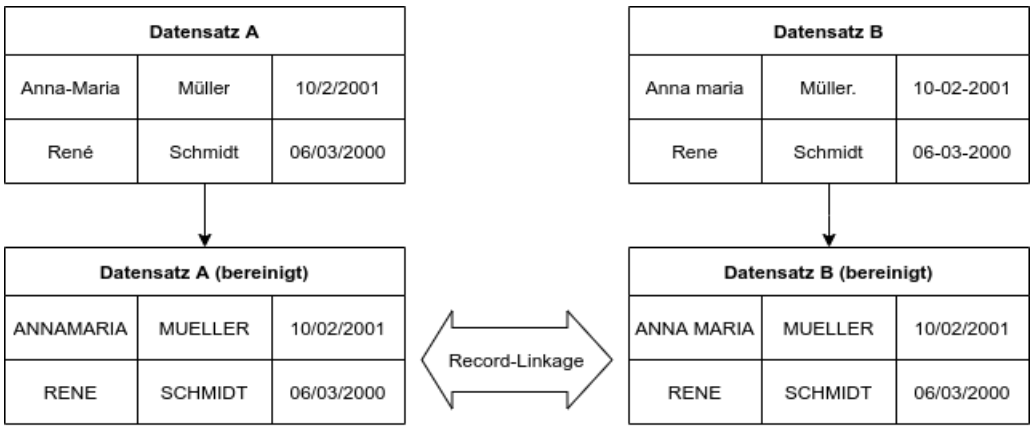

Abbildung 9: Datenbereinigung für zwei fiktive Datensätze mit anschließendem Record-Linkage

Im Folgenden werden alle Schritte des Preprocessings beschrieben, die für das Linkage der simulierten Registerdaten angewendet wurden. Es entfällt dabei der Schritt der Datenaufbereitung, da die Daten aus einer Mikro-Simulation stammen und sie somit den gleichen datengenerierenden Prozess haben. Daher haben alle Datensätze schon zu Beginn das gleiche Format.

2.2.1 Datenbereinigung

Wie in Abschnitt 1.3.2 bereits beschrieben, unterliegen Daten immer einem datengenerierenden Prozess. Dabei haben für gewöhnlich die beiden zu verknüpfenden Datensätze unterschiedliche datengenerierende Prozesse, da sie meist von unterschiedlichen Quellen stammen. Die Aufbereitung der Daten ist daher für jedes Record-Linkage-Projekt von höchster Relevanz. Ziel ist es, alle Merkmale in den beiden zu verknüpfenden Datensätze in ein gleiches Format zu überführen. In Abbildung 9 ist eine solche Bereinigung beispielhaft dargestellt. Eine Datenbereinigung kann jedoch auch zu weiteren Fehlern oder einer Verschlechterung der Linkagequalität führen. Die verwendeten Verfahren müssen daher auf die vorhandenen Daten abgestimmt werden.

Ziel bei der Bereinigung von Textfeldern ist die Reduktion von häufigen Datenfehlern, wobei der Qualitätsgewinn höher sein sollte als der Informationsverlust. Zwar bedarf jedes Record-Linkage-Projekt immer auch einige Vorüberlegungen für eine optimale Datenbereinigung, einige Schritte sind jedoch gängige Praxis (Wickham 2014).

In der deutschen Sprache fangen Namen z. B. immer mit einem Großbuchstaben an, auf welche dann Kleinbuchstaben folgen. Eine Abweichung von diesem Muster ist meist ein Datenfehler und das Muster selbst ist kein großer Informationsträger. Daher empfiehlt es sich, die Groß- und Kleinschreibung zu umgehen und alle Buchstaben entweder klein oder groß zu schreiben. Auch Sonderzeichen und Interpunktionszeichen (Punkt, Komma etc.) sind für Namen entweder unüblich oder sehr fehleranfällig.

Textfelder werden daher meist in reinem ASCII ohne Interpunktionszeichen codiert. Im Groben bedeutet dies, dass nach der Bereinigung nur Buchstaben von A bis Z sowie Zahlen und Leerzeichen in den Textfeldern stehen sollten. Alle Sonderzeichen werden daher entweder gelöscht oder auf ein ASCII-Zeichen reduziert (z. B. wird Ä zu AE und Á zu A). Die Bereinigung der Textfelder sollte zudem auf beiden Datensätzen exakt gleich ausgeführt werden. Dies bewirkt, dass gleiche Namen zwar umcodiert, aber immer noch gleich geschrieben werden.

Auch die simulierten Datensätze unterliegen einer solchen Bereinigung. Zwar besitzen die zu verknüpfenden Datensätze die gleiche Datenquelle, sodass eine Bereinigung theoretisch immer zu einem Informationsverlust führt, jedoch ist dieser Fall für ein Linkage-Projekt mit realen Daten nahezu ausgeschlossen. Um also auch den Informationsverlust einer Datenbereinigung zu simulieren, werden alle Textfelder (Vorname, Nachname, Geburtsort) in Großbuchstaben im ASCII-Format überführt. Alle Zahlenfelder (Geburtstag, -monat, -jahr, Geschlecht, Postleitzahl) haben bereits ein gleiches bereinigtes Format, sodass sie nicht weiter bearbeitet werden.

2.2.2 Phonetische Codierung

Eine weitere Fehlerquelle bei der Eingabe von Daten besteht darin, dass phonetisch gleich klingende Namen bei einer manuellen Dateneingabe verwechselt werden. So könnte bspw. der Name Marc fälschlicherweise als Mark geschrieben werden. Zur Lösung dieses Problems werden seit Längerem phonetische Codierungen verwendet. Bereits zahlreiche solcher phonetischen Codierungen existieren (Christen 2012: 74). Dadurch, dass sich in unterschiedlichen Sprachen die Aussprache von Wörtern unterscheidet, setzen auch die Phonetiken meist einen Schwerpunkt auf eine Sprache. Häufig ist dies Englisch. Für den deutschen Sprachraum wird meist die Kölner-Phonetik verwendet (Postel 1969). Einer der wohl umfangreichsten – und daher für den internationalen Gebrauch besten – Phonetiken wurde vom New York State Identification and Intelligence System entwickelt und wird gleichnamig als NYSIIS bezeichnet (Christen 2012: 76).

Die Aufgabe der Codierungen ist immer gleich: Ein Wort wird in einen *phonetischen Code* überführt, sodass ähnlich klingende Worte – im besten Fall – den gleichen Code erhalten. Somit können Worte auch in der Ähnlichkeit ihres Codes verglichen werden.

Die wohl bekannteste und auch mit die älteste phonetische Codierung ist Soundex (Christen 2012: 74). Obwohl Soundex eher für den englischen Sprachraum ausgelegt ist, wird er durch seine Bekanntheit und gute Performanz bis heute sehr häufig verwendet.[35] Auch für das Linkage der Simulationsdaten wird ausschließlich Soundex verwendet, da einerseits einige Methoden explizit Soundex verwenden und

[35]Die Häufigkeit der Anwendung entspricht nicht der Qualität der gebildeten Codes. So erzielt bspw. die Phonetik von Reth und Schek (1977) durchaus bessere Ergebnisse in der deutschen Sprache als die Kölner Phonetik (Bachteler und Schnell 2006). Im Kontext der Simulation ist es zudem vorstellbar, dass NYSIIS bessere Linkage-Ergebnisse als Soundex erzielen würde. Aus den Gründen der Pupolarität von Soundex wird jedoch davon abgesehen.

a, e, h, i, o, u, w, y	→	0
b, f, p, v	→	1
c, g, j, k, q, s, x, z	→	2
d, t	→	3
l	→	4
m, n	→	5
r	→	6

Abbildung 10: Transformationstabelle zur Bildung eines Soundex-Codes (Christen 2012: 77).

anderseits Soundex durch seine Bekanntheit und häufige Anwendung einen guten Vergleichsmaßstab bildet. Im Folgenden wird daher die Funktionsweise von Soundex genauer beschrieben.

Jeder Soundex-Code besteht aus einem Buchstaben, gefolgt von drei Zahlen. Ein Wort wird linear von Anfang bis Ende durchlaufen. Der erste Buchstabe des Wortes ist immer gleich dem Buchstaben im Soundex-Code. Alle weiteren Buchstaben werden gemäß einer Transformationstabelle in Zahlen codiert (Abbildung 10). Danach werden alle Nullen gelöscht und direkt aufeinanderfolgende gleiche Zahlen auf eine Zahl beschränkt. Hat der Code danach weniger als drei Zahlen, so werden an das Ende des Codes so viele Nullen angehängt, bis der Code drei Zahlen hat. Der Soundex-Code für den Namen MATTHIAS ist also zunächst M0330002. Nach der ersten Bereinigung ist der Code nur noch M32 und der endgültige Code ist mit dem Auffüllen der Nullen M320.

Aus der Transformationstabelle lässt sich auch erkennen, dass Soundex weder Sonderzeichen noch Interpunktionszeichen verarbeiten kann. Auch dies ist ein Grund dafür, alle Namen in ASCII zu codieren. Hinzu kommt, dass Soundex auch nicht mit Leerzeichen umgehen kann. Es bedarf daher eines einheitlichen Umgangs mit Mehrfachnamen, entweder durch die Reduktion auf den ersten Namen oder durch Zusammenfassen aller Namen. Letzteres wird für das Linkage der simulierten Daten gemacht. Alle Leerzeichen werden bei einem Namen gelöscht und es wird der Soundex des zusammenhängenden Namens gebildet. Der Name ANNA MARIA erhält daher den Soundex-Code A556 von ANNAMARIA. Insgesamt wird der Soundex für alle Textfelder (Vorname, Nachname, Geburtsort) gebildet.

2.2.3 Tokenbildung

Auch in anderen Kontexten ist der Umgang mit Mehrfachnamen relevant. Eine Lösung dieses Problems besteht in der Trennung eines Namens an den Leerzeichen in einzelne Namensbestandteile. Diese Namensbestandteile werden allgemein als *Tokens* bezeichnet. Es sind je nach Kontext zahlreiche Anwendungsfälle für diese Tokens denkbar, für das Record-Linkage der Simulationsdaten ist es jedoch wichtig, einen Namen auf ein Token zu reduzieren. Dieses Token sollte gleichzeitig über einen hohen Informationsgehalt verfügen. Für Vor- und Nachname ist dies der erste Name, da dieser Name häufig auch der Rufname ist und daher am häufigsten genannt wird. Gemeindenamen (Geburtsort)

enthalten hingegen manchmal ein Präfix (Bad, St. etc.) und/oder Suffix (am Main etc.). Die Präfix- und Suffix-Tokens sind *nicht-diskriminierende Token*. Die Verwendung dieser Tokens würde daher zu einem starken Informationsverlust führen, weshalb diese Tokens ausgeschlossen werden müssen.

Um den Geburtsort der Simulationsdaten auf einen Token zu reduzieren, wird daher das erste Token gewählt, das sich nicht in einer Menge an häufigen Präfixen befindet (GROSS, KLEIN, BAD, ALT, NEU, UNTER, OBER, ST, AM, SAN, LES, LOS). Somit müssen Suffixe nicht beachtet werden, da sie nach dem gewählten ersten Token vorkommen. Da Leerzeichen immer Tokens voneinander trennen, bezieht sich die Menge der Präfixe auf eigenständige Tokens. OBERHAUSEN wird daher weiterhin als OBERHAUSEN codiert, BAD LAUTERBERG IM HARZ jedoch als LAUTERBERG. Hinzu kommt, dass auch noch der Soundex des Tokens des Geburtsorts gebildet wird, da besonders Gemeindenamen stark in ihrer Länge variieren.

2.3 Record-Linkage Methoden

Insgesamt wurden 21 Record-Linkage Verfahren (Kombinationen von Linkage-Methoden und Konfigurationen) auf die simulierten Registerdaten angewendet. Drei Matchkey-Verfahren wurden jeweils auf den gesamten Datensatz angewendet (*Gesamtmenge*). Stimmen Records in beiden zu vergleichenden Datensätzen in allen Merkmalen überein[36], so werden diese für das weitere Linkage aussortiert (gefiltert). Die übrig gebliebenen Records bilden somit eine *Restmenge* an (noch) nicht gelinkten Records. Alle anderen 18 Verfahren arbeiten nur auf dieser Restmenge. Die Gründe für ein solches Vorgehen wurden in Abschnitt 2.1.3 erläutert.

Im Folgenden werden nun die einzelnen verwendeten Methoden erklärt und ihre genaue Konfiguration und Implementation beschrieben. Wie in Abschnitt 1.2.1 erläutert, ist für den Datenschutz das Argument der Datensparsamkeit zentral. Daher operieren alle Methoden sowohl auf der Gesamtmenge auch auf einer Teilmenge der Linkage-Merkmale. Für alle Methoden ist dabei die Fragestellung relevant, ob der Geburtsort ein relevantes Merkmal für die Linkage-Qualität ist (siehe Abschnitt 1.2.3).

Keinesfalls können alle bekannten Record-Linkage Methoden in dieser Arbeit angewendet werden. Daher wurde eine Auswahl von Methoden getroffen, entweder weil sie populär und allgemein verbreitet sind (Matchkeys, EM/ECM, Bloom-Filter) oder weil sie neu und sehr vielversprechend sind (Multiple Matchkeys, Signaturen).

Allen Methoden ist eine allgemeine Vorgehensweise gemein: Da in simulierten Daten jede Person immer durch exakt ein Record repräsentiert wird, existieren nur one-to-one Matches. Wird durch ein Verfahren daher einem Record im Datensatz A mit der gleichen Sicherheit mehrere Records im Datensatz B zugewiesen, so spricht dies gegen diese Annahme. In keiner der Methoden werden solche Links

[36]Dies entspricht dem Matchkey: Vor-, Nachname, Geburtstag, -monat, -jahr, Geschlecht und Geburtsort.

Datenbestand	Verwendete Merkmale
Gesamtmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht, Geburtsort
Gesamtmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht
Gesamtmenge	Vorname, Tag, Monat, Jahr, Geschlecht, Geburtsort
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht
Restmenge	Vorname, Tag, Monat, Jahr, Geschlecht, Geburtsort
Restmenge	S(Vorname), S(Name), Tag, Monat, Jahr, Geschlecht, Geburtsort
Restmenge	S(Vorname), S(Name), Tag, Monat, Jahr, Geschlecht, S(Geburtsort)
Restmenge	S(Vorname), S(Name), Tag, Monat, Jahr, Geschlecht
Restmenge	Vorname{2,3}, Name{2,3,5}, Tag, Monat, Jahr, Geschlecht

Tabelle 5: Verwendete Merkmale für die Matchkey-Verfahren. Einige Schreibweisen wurden für die Darstellung verkürzt. S(x) bedeutet, dass der Soundex von x verwendet wurde. Zur Darstellung des SLK-581 bedeutet x{i...}, dass nur die Zeichen von x an den genannten Indizes i verwendet werden.

(Mehrfachzuweisungen) als Match klassifiziert. Dies führt dazu, dass alle Verfahren besser miteinander vergleichbar und die Ergebnisse eindeutig sind.

2.3.1 Matchkeys

Die Funktionsweise von Matchkeys wurde bereits in Abschnitt 2.1.3 behandelt. Für das Linkage der Simulationsdaten bieten sich zahlreiche Merkmalskombinationen solcher Matchkeys an. Es wird dabei geklärt, welche Auswirkungen das Fehlen des Geburtsortes oder Nachnamens auf die Linkage-Qualität hat. Aufbauend auf dem Argument der Datensparsamkeit wird auch getestet, welche Linkage-Qualität zu erwarten ist, falls nur der Soundex einer Zeichenkette verwendet wird.

Tabelle 5 beinhaltet alle Merkmalskombinationen der Matchkeys. Zwei bekannte Formen des Matchkeys wurden dabei ebenfalls untersucht: Der Swiss ALC und der SLK-581.

Der *Swiss Anonymous Linkage Code* (ALC) ist ein Matchkey, der aus der zusammengesetzten Zeichenkette von Soundex(Vorname), Soundex(Nachname), Geburtstag, -monat, -jahr und Geschlecht gebildet wird (Borst et al. 2001). Diese Zeichenkette wird anschließend über eine kryptografische Hash-Funktion verschlüsselt. Diese wandelt die Zeichenkette in einen eindeutigen Zahlenwert um, ohne dass aus dem Zahlenwert die Zeichenkette wieder zurückzuverfolgen ist. Da gute Hash-Funktionen sehr selten gleiche Werte für unterschiedliche Eingaben produzieren (Kollision), ist der Einfluss einer Hash-Funktion auf das Linkage-Ergebnis vernachlässigbar. Dementsprechend wurde auf eine Verschlüsselung verzichtet.

Auch der australische *Statistical Linkage Key* (SLK-581) ist ein Matchkey der über eine konkatenierte Zeichenkette entsteht. In der Regel wird auch dieser Wert verschlüsselt. Aus den genannten

Gründen wird aber auch hier darauf verzichtet. Zur Bildung des SLK-581 wird der zweite, dritte und fünfte Buchstabe des Nachnamens sowie der zweite und dritte Buchstabe des Vornamens extrahiert und mit dem vollständigen Geburtsdatum und Geschlecht zu einer Zeichenkette verbunden (Karmel 2005). Für den Record BARBARA BUTLER, 02.01.1923, Weiblich (Code = 2), ist der SLK-581 bspw. UTEAR020119232 (Karmel 2005: 18).

2.3.2 Probabilistisches Record-Linkage mit ECM

Wie in Abschnitt 2.1.4 angemerkt, bauen probabilistische Record-Linkage Methoden meist auf dem Fellegi-Sunter-Modell (Fellegi und Sunter 1969) auf. Die Vorgehensweise bei diesem Modell lässt sich wie folgt erklären. Für zwei Datensätze A und B beinhaltet die Menge an Tupels (a, b)

$$A \times B = \{(a, b); a \in A, b \in B\}, \tag{2.23}$$

alle vergleichbaren Record-Kombinationen. Innerhalb dieser Kombinationen existiert die Menge M an true Matches und U an non-Matches (Herzog et al. 2007: 33):

$$M = \{(a, b); a \in A, b \in B, (\text{a,b}) \text{ ist ein true Match}\} \tag{2.24}$$

und

$$U = \{(a, b); a \in A, b \in B, (\text{a,b}) \text{ ist } \textbf{kein} \text{ true Match}\}.^{37} \tag{2.25}$$

Wird zusätzlich ein Ähnlichkeitsmaß verwendet, so wird der Grad der Übereinstimmung in unterschiedliche Level codiert.[38] Werden alle K Variablen zweier Records miteinander verglichen und die Übereinstimmung codiert, so entsteht der Vektor γ. Alle möglichen Kombinationen von γ ergeben die Menge Γ. Aus diesen Angaben kann ein Match-Score R gebildet werden. Dies ist das Verhältnis der Wahrscheinlichkeiten P, dass zwei Records a und b mit einem Übereinstimmungsmuster γ zu der Menge der Matches bzw. non-Matches gehören. Formal:

$$R = \frac{P(\gamma \in \Gamma | (a, b) \in M)}{P(\gamma \in \Gamma | (a, b) \in U)}. \tag{2.26}$$

Da M und U unbekannt sind, muss R anderes berechnet werden. So müssen für alle Variablen $1 \leq i \leq K$ – unter der Annahme, dass die Ausprägungen der Variablen unabhängig voneinander sind (*conditional independence*) – die m- und u-Wahrscheinlichkeiten berechnet werden. Diese geben an, mit welcher Wahrscheinlichkeit zwei Records a und b zur Menge der true Matches bzw. non-Matches

[37] Wie bereits gezeigt, gilt meistens $|U| \gg |M|$.

[38] Für gewöhnlich sind dies drei Level: Identisch, fast identisch und nicht übereinstimmend.

gehören, falls sie in der Variable i das gleiche Übereinstimmungsmuster haben. Daher:

$$m_i = \{a_i = b_i | (a, b) \in M; a \in A, b \in B\} \tag{2.27}$$

und

$$u_i = \{a_i = b_i | (a, b) \in U; a \in A, b \in B\}. \tag{2.28}$$

Mithilfe der m- und u-Parameter kann dann das Match-Gewicht w_i für jede Variable berechnet werden:

$$w_i = \begin{cases} \log_2\left(\frac{m_i}{u_i}\right), & \text{falls die Muster übereinstimmen} \\ \log_2\left(\frac{1-m_i}{1-u_i}\right), & \text{falls die Muster nicht übereinstimmen.} \end{cases} \tag{2.29}$$

Es lässt sich zeigen, dass die Summe der einzelnen Match-Gewichte den logarithmierten Match-Score ergeben:

$$\log_2(R) = \sum_{i=1}^{n} w_i. \tag{2.30}$$

Mithilfe des Bayes-Theorems kann zudem eine Match-Wahrscheinlichkeit p berechnet werden. Die Berechnung von p würde an dieser Stelle jedoch zu weit gehen und keinen Mehrwert über die Funktionsweise der Methode schaffen. Es sei daher auf die entsprechende Literatur verwiesen (Enamorado et al. 2019: 356).[39]

Auch die m- und u-Parameter sind zunächst unbekannt. Sie lassen sich aber anhand der vorliegenden Datensätze A und B mithilfe des *Expectation Maximization Algorithmus* (EM) berechnen. Der EM ist ein Algorithmus zur Schätzung der *Maximum Likelihood* (Herzog et al. 2007: 92). Die m- und u-Parameter werden daher mit Startwerten initialisiert[40] und iterativ anhand der bisherigen Werte neu geschätzt. Dies geschieht so lange, bis die Parameter sich kaum noch verändern (konvergieren).

Die Schätzungen des EM-Algorithmus erfolgen über einen E- (Expectation) und einen M-Schritt (Maximization). Da der Rechenaufwand des M-Schritts immer sehr groß ist, schlagen Meng und Rubin (1993) vor, den M-Schritt durch mehrere *Conditional Maximization* (CM) Schritte zu ersetzen. Dies verringert den Rechenaufwand und beschleunigt das Verfahren. Den daraus resultierenden Algorithmus nennen sie Expectation/Conditional Maximization (ECM).

Drei Implementationen des EM bzw. ECM wurden getestet:

1. *Splink* (Office for National Statistics 2020)
2. *FastLink* (Enamorado et al. 2019)

[39] Üblicherweise werden auch Häufigkeitsverteilungen von Merkmalen in die Wahrscheinlichkeitsberechnung mit einbezogen. So ist die Wahrscheinlichkeit das zwei Records zur selben Person gehören bspw. größer, wenn die Records einen seltenen Namen haben. Die Match-Wahrscheinlichkeit wird dann entsprechend der Häufigkeit angepasst. (Herzog et al. 2007; Enamorado et al. 2019).

[40] Es werden die von Jaro (1989) vorgeschlagenen Startwerte verwendet.

3. *recordlinkage* (de Bruin 2015).

Eine zunächst vielversprechende Implementation ist das Python-Paket *Splink*, welches das Linkage vollständig in *Spark* durchführt. Die Laufzeiten des Pakets sind sehr gut, jedoch skaliert das Paket für große Fallzahlen schlecht, sodass es den Arbeitsspeicher überlastet und deshalb sehr instabil ist. Daher kann das Paket nur mit stark restriktiven Blocking-Regeln verwendet werden.

FastLink ist ein R-Paket mit einigen Funktionen, die in C implementiert sind. Das Blocken bei den großen Fallzahlen der Simulationsdaten erwies sich bei Tests als Schwachstelle des Paktes. So gelang es nicht, mit geblockten Daten plausible Ergebnisse zu erzielen. Dieses Verhalten hat sich auch schon bei anderen Simulationen gezeigt (Schnell, Borgs et al. 2020).

Eine weitere Implementation ist das Python-Paket *recordlinkage*, das auf dem Paket *FEBRL* aufbaut (Christen 2008).[41] Das Paket *recordlinkage* verwendet zur Schätzung der Parameter einen ECM-Algorithmus. Auch mit lockeren Bocking-Regeln erwies sich das Paket als sehr stabil und produzierte dabei sehr gute Ergebnisse mit gleichzeitig sehr guten Laufzeiten. Das probabilistische Record-Linkage der Simulationsdaten wurde daher mit diesem Paket durchgeführt.

Zur Verknüpfung der Simulationsdaten wird ein mehrstufiges exaktes Blocking-Verfahren angewendet. Dabei werden für jeden Schritt alle klassifizierten Matches über dem Schwellenwert T aussortiert. Dadurch können immer lockerere Blocking-Regelen verwendet werden. Die Blocking-Regeln wurden für jede Merkmalskombination manuell im Vorhinein bestimmt. Hierfür waren zwei Faktoren relevant:

1. Die Anzahl der Vergleiche, die durch die Blocking-Regel entstehen und
2. die Wahrscheinlichkeit (weitere) Matches durch die (neuen) Blocking-Regeln zu finden.

Die Kombination an verwendeten Merkmalen und Blocking-Regeln befindet sich in Tabelle 6. Für alle Schritte wird der Schwellenwert $T = 0.8$ verwendet. Da die m- und u-Parameter von der verwendeten Blocking-Regel abhängig sind (Fellegi und Sunter 1969: 64), wird für jede Regel eine eigene Parameterschätzung durchgeführt.[42] Die Schätzung erfolgt immer anhand einer Stichprobe von jeweils 200,000 Records auf dem gesamten Datensatz (vor der Aussortierung der exakten Matches). Da sich Parameterschätzungen im Zeitverlauf selten ändern, werden für jede Fehlerquote die Parameter einmalig mit dem ersten Datensatzpaar geschätzt. Alle weiteren Datensatzpaare der entsprechenden Fehlerquote werden dann mit diesen geschätzten Parametern verknüpft.

Zum Vergleich von Zeichenketten (Vorname, Nachname, Geburtsort) wird die Jaro-Winkler-Distanz verwendet. Der Schwellenwert für eine wahrscheinliche Übereinstimmung liegt bei dem von Winkler (1990) empfohlenen Wert von 0.88. Alle anderen Variablen werden exakt miteinander verglichen.

[41] Das Paket *recordlinkage* sollte nicht mit dem R-Paket *RecordLinkage* (Sariyar und Borg 2010) verwechselt werden, welches im Vorfeld ausgeschlossen wurde, da es sich nicht für sehr große Datensätze eignet.

[42] Dies lässt sich leicht verdeutlichen, wenn über den Soundex des Vornamens geblockt wird. In diesem Fall ist die Wahrscheinlichkeit für gleiche Vornamen viel höher. Die Gewichtung des Vornamens muss daher geringer ausfallen.

Datenbestand	Verwendete Merkmale
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht, Geburtsort *Blocking-Regeln:* 1. Tag & Monat & Jahr 2. Soundex(Vorname) & Soundex(Name) 3. Soundex(Vorname) 4. Geburtsort
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht *Blocking-Regeln:* 1. Tag & Monat & Jahr 2. Soundex(Vorname) & Soundex(Name) 3. Soundex(Vorname) & Jahr

Tabelle 6: Verwendete Merkmale und Blocking-Regeln für den ECM-Algorithmus. Einige Schreibweisen wurden für die Darstellung verkürzt. Die Blocking-Regeln sind nach der Reihenfolge der Anwendung sortiert. Das & impliziert ein logisches Und. Für einen Block müssen daher alle mit einem & verbundenen Merkmale übereinstimmen.

2.3.3 Multibit Trees mit CLKs

Cryptographic Longterm Keys (CLKs) gehören zur Gruppe der Bloom-Filter Methoden. Es benötigt daher zunächst einiges Grundwissen über Bloom-Filter. Diese werden besonders im *Privacy Preserving Record Linkage* (PPRL) verwendet (Christen, Ranbaduge et al. 2020). Das Verfahren geht auf Schnell, Bachteler et al. (2009) zurück und wurde seitdem zahlreich modifiziert. Die Grundidee besteht in der Verschlüsselung einer Zeichenkette in einen Bit-Vektor. Zu Beginn werden alle Bits mit Nullen initialisiert. Darauf folgend wird die Zeichenkette so codiert, dass sie als Einsen auf dem Bit-Vektor repräsentiert werden kann. Die Codierung einer Eins erfolgt für denselben Eingabewert immer gleich. Somit können so erzeugte Bit-Vektoren miteinander verglichen werden (Ähnlichkeitsmaß), da die gleichen Eingabewerte immer zu den gleichen Bit-Vektoren führen (wenn alle Verschlüsselungskonfigurationen bekannt sind).

Abbildung 11 zeigt die schematische Darstellung eines Vergleichs mithilfe von Bloom-Filtern. Dafür wird die Zeichenkette zu Beginn in q-Gramme aufgeteilt (siehe Abschnitt 2.1.5). Zur Codierung der Zahlenwerte in Bits auf dem Vektor werden dann Hash-Funktionen benötigt. Diese codieren Eingabewerte in eine Zahl aus einem festen Zahlenbereich (Hash-Wert). Gleiche Eingabewerte führen immer zu gleichen Ergebnissen. Jedes entstandene q-Gramm wird mit k verschiedenen Hash-Funktionen verschlüsselt. Für den ermittelten Hash-Wert wird die Position auf dem Bit-Vektor berechnet, indem der Hash-Wert modulo der Länge des Bit-Vektors gerechnet wird. An dieser Position wird das Bit im Vektor auf eins gesetzt. Zeigen mehrere Hash-Werte auf die gleiche Position (Kollision), so bleibt das Bit auf eins gesetzt.

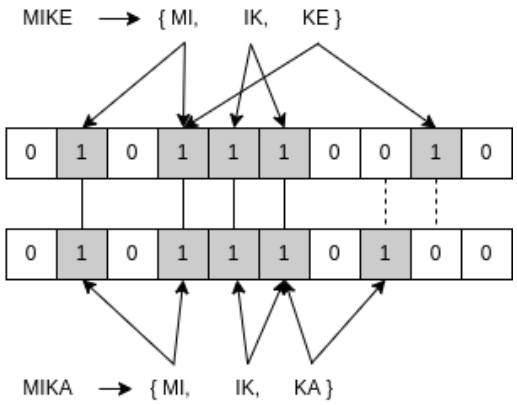

Abbildung 11: Schematische Darstellung zu Verschlüsselung und Abgleich der Namen MIKE und MIKA in Bloom-Filtern (l = 10, k = 2, q = 2).

Der so entstandene Bit-Vektor ist nun ein Bloom-Filter. Da q-Gramme in den Vektor verschlüsselt wurden, können Ähnlichkeitsmaße für q-Gramme verwendet werden, um die Vektoren zu vergleichen. Die Zahl der Einsen in einem Vektor repräsentiert die Zahl der q-Gramme und die Anzahl der übereinstimmenden Einsen die Zahl der gemeinsamen q-Gramme. Für das Beispiel aus Abbildung 11 kann somit bspw. der Dice-Koeffizient berechnet werden:

$$sim_{Dice}(x, y) = \frac{2 \cdot c}{|x| + |y|} = \frac{2 \cdot 4}{5 + 5} = 0.8.$$

Die ursprüngliche Veröffentlichung (Schnell, Bachteler et al. 2009) sieht vor, dass alle Linkage-Merkmale als einzelne Bloom-Filter verschlüsselt werden. Mithilfe von probabilistischen Record-Linkage Verfahren und der Ähnlichkeiten der einzelnen Merkmale können so Datensätze verknüpft werden.

Die Verschlüsselung der Merkmale in einzelne Bloom-Filter ist jedoch angreifbar durch Häufigkeitsangriffe und somit ein weniger sicheres Verfahren für PPRL. Mehrere Möglichkeiten wurden daher vorgeschlagen, um die Sicherheit von Bloom-Filtern zu erhöhen (Christen, Ranbaduge et al. 2020). Eines dieser Verfahren ist der Cryptographic Longterm Key (Schnell, Bachteler et al. 2011). Ein Beispiel für die Erstellung eines CLKs befindet sich in Abbildung 12. Die Grundidee für einen CLK beruht auf anonymen Linkage Codes (ALC), welche alle Linkage-Merkmale in eine Zeichenkette zusammenführen. Indem alle Merkmale in einzelne Bloom-Filter verschlüsselt und dann zu einem einzigen Bloom-Filter zusammengesetzt werden, ist ein solcher ALC auch als Bloom-Filter repräsentierbar. Dieser zusammengesetzte Bloom-Filter ist der CLK eines Records.

Aufgrund der vielen q-Gramme, die in einen CLK einfließen, werden für CLKs meist sehr lange Bit-Vektoren verwendet. Schnell, Bachteler et al. (2011) nutzen z. B. Vektoren mit 1000 Bits und $k = 10$ Hash-Funktionen. Darüber hinaus haben sehr lange Merkmale – wie z. B. Namen mit Adelstiteln – ein höheres Gewicht im CLK, als kürzere – wie z. B. der Geburtstag. Daher ist es häufig nötig, dass

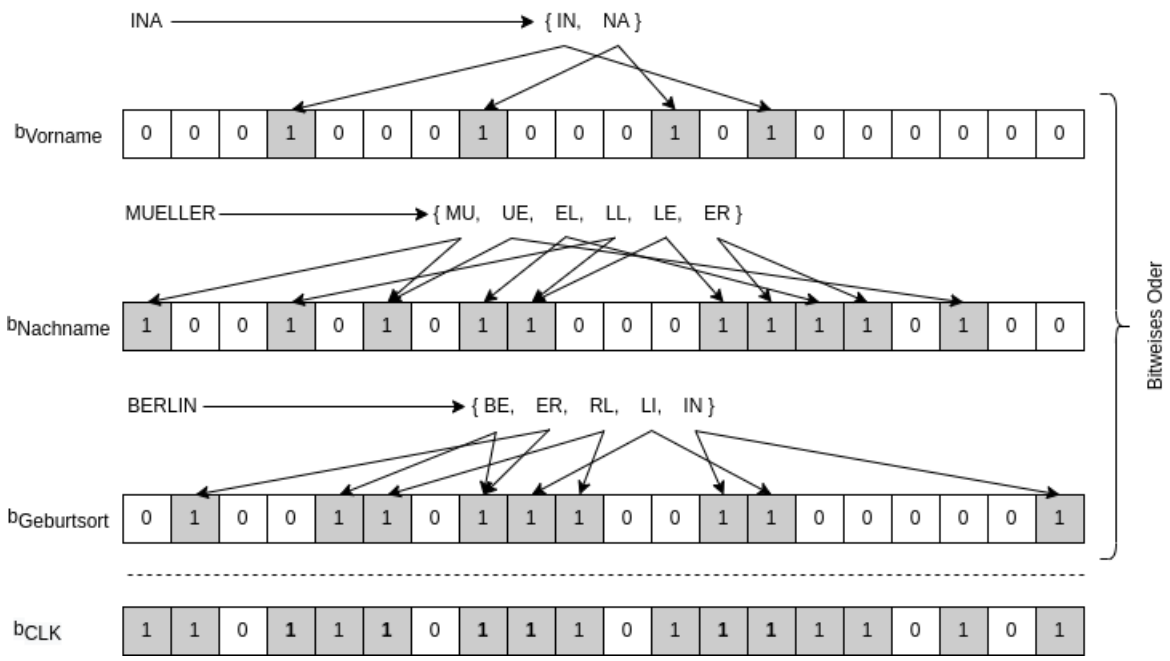

Abbildung 12: Schematische Darstellung der Verschlüsselung eines Records in einen CLK der Länge $l = 20$. Alle Merkmale wurden in Bigramme ($q = 2$) aufgeteilt und mit $k = 2$ Hash-Funktionen verschlüsselt. Die Abbildung ist angelehnt an Christen et al. (2020: 202).

Datenbestand	Verwendete Merkmale
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht, T(Geburtsort)
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht
Restmenge	T(Vorname), T(Name), Tag, Monat, Jahr, Geschlecht, T(Geburtsort)

Tabelle 7: Verwendete Merkmale zur Bildung der CLKs. Einige Schreibweisen wurden für die Darstellung verkürzt. T(x) bedeutet, dass das nur ein Token von x verwendet wird.

für die Erstellung eines CLKs nur ein Token eines potenziell langen Merkmals verwendet wird (siehe Abschnitt 2.2.3).

In Tabelle 7 werden alle Merkmalskombinationen aufgelistet, mit denen CLKs zur Verknüpfung der simulierten Daten gebildet werden. Es werden dabei Bit-Vektoren mit $l = 1000$ und $k = 10$ verwendet. Darüber hinaus werden alle Zeichenketten (Vorname, Nachname, Geburtsort) mit Bigrammen codiert ($q = 2$) und alle anderen Merkmale (Geburtstag, -monat, -jahr, Geschlecht) mit Unigrammen ($q = 1$).

CLKs bieten den Vorteil, dass die Ähnlichkeit zweier Records nun als eine Match-Wahrschein-lichkeit interpretiert werden kann. Aufgrund der Länge der Vektoren weist der Vergleich zweier CLKs einen hohen Rechenaufwand auf. Daher existieren mehrere Strukturen, die den Vergleich beschleunigen, indem unplausible Matches aussortiert werden. Eine dieser Strukturen ist der Sorted Neighbourhood Ansatz, bei dem die CLKs nach der Anzahl Einsen im Vektor (Hamming-Gewicht) sortiert und nur

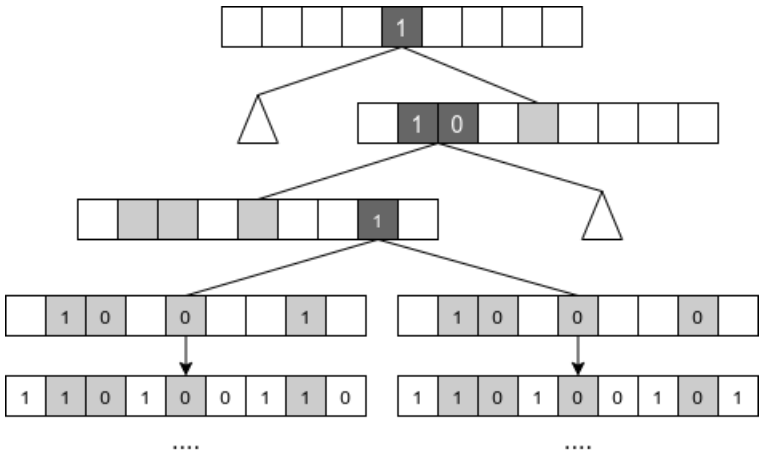

Abbildung 13: Beispiel für ein Multibit-Tree (angelehnt an Kristensen et al. 2010: 5). Dunkle Quadrate markiere Bits, an denen der Baum geteilt wird. Für linke Äste trifft die Bedingung zu, für rechte nicht. Zur Vereinfachung wurden weitere Unterbäume als Dreiecke abgekürzt. An den Blättern hängt wiederum die Liste der einsortierten Bit-Vektoren.

die k-nächsten Vektoren miteinander verglichen werden (Hernandez und Stolfo 1998). Dieser Ansatz ist zwar sehr schnell, für große Datensätze erweist er sich jedoch als fehleranfällig (Schnell 2016).

Eine Datenstruktur, die auch für große Datensätze schnelle und gute Linkage-Ergebnisse erzielt (Borgs 2019) sind Multibit Trees (Kristensen et al. 2010). Ein Multibit Tree ist ein Binärbaum – eine Baumstruktur mit maximal zwei Ästen an jedem Knoten (siehe Abbildung 13) – in dem Bit-Vektoren einsortiert sind. Äste werden anhand von Übereinstimmungsmustern in ein oder mehreren Bit-Positionen gebildet. Dabei wird versucht einen möglichst balancierten Baum zu erzeugen. Dies bedeutet, dass sich in jedem Ast ungefähr gleich viele Bit-Vektoren befinden. Sind in einem Ast weniger als n Bit-Vektoren vorhanden, so endet der Generierungsprozess für diesen Ast. Die n Bit-Vektoren bilden dann ein Blatt. Dies geschieht so lange, bis alle Bit-Vektoren sich in einem Blatt befinden.[43]

Um zwei Datensätze mit CLKs zu verknüpfen, wird zunächst aus einem Datensatz eine solche Baumstruktur erzeugt. Danach werden alle Records des anderen Datensatzes nacheinander in diesem Baum gesucht. Als Ähnlichkeitsmaß verwenden Multibit Trees den Tanimoto-Koeffizienten. Dieser berechnet sich für zwei Bit-Vektoren a und b mit der Länge l und den einzelnen Bits an der Stelle i als

$$sim_{Tanimoto}(a,b) = \frac{\sum_{i=1}^{l}(a_i \wedge b_i)}{\sum_{i=1}^{l}(a_i \vee b_i)}, \tag{2.31}$$

wobei $\wedge$ das logische Und und $\vee$ das logische Oder darstellen. Mithilfe eines Records und einem Ast, kann das Maximum des Tanimoto-Koeffizienten berechnet werden, der für alle Records innerhalb des Astes erreicht werden kann. Durch die Angabe eines Grenzwertes wird so die Zahl der Vergleiche reduziert, da nur Records über dem Grenzwert miteinander verglichen werden. Zur Verknüpfung der simulierten Daten wurden die Grenzwerte 0.8 und 0.85 getestet.

[43] Kristensen et al. (2010: 5) verwenden $n = 6$.

Multibit Trees sind somit eine Blocking-Strategie für die CLKs. Da für die Verknüpfung der Simulationsdaten die Zahl der Vergleiche immer noch zu groß ist, wird darüber hinaus exaktes Blocken auf die CLKs angewendet. Dabei werden alle Blockingvariablen zusätzlich zu den CLKs gespeichert und immer ein Multibit Tree für alle CLKs gebildet, die in der Blockingvariable exakt übereinstimmen.

In der Implementation des Verfahrens wird getrennt voneinander zweimal geblockt: Erst über das Geburtsjahr und danach über den Nachnamen. Da immer nur über ein Merkmal geblockt wird, ist zu erwarten, dass der überwiegende Teil der potenziellen Matches mindestens einmal miteinander verglichen wird. Da dies getrennt geschieht, werden am Ende alle Ergebnisse zusammengeführt und jeweils das beste Ergebnis für ein Record-Paar gewählt.

2.3.4 Multiple Matchkeys

Von Randall et al. (2019) stammt eine Methode, die auf den positiven Eigenschaften von deterministischen Verfahren aufbaut. Hierzu wird nicht einer, sondern mehrere Matchkeys (daher *Multiple Matchkeys*) verwendet, um Records zu klassifizieren. Für die Auswahl der Matchkeys schlägt Randall (2019: 4) vor, die Match-Gewichte aus dem probabilistischen Record-Linkage zu verwenden. Alle möglichen Matchkeys sind damit gleich der Menge Γ – wobei nur exakte Abgleiche verwendet werden.

Ein Matchkey $\gamma \in \Gamma$ wird in Betracht gezogen, falls

$$\sum_{i=1}^{K} w_i \geq S, \tag{2.32}$$

wobei S einen gewählten Schwellenwert repräsentiert und w_i das Gewicht für die Ausprägung von γ an der Stelle i (siehe hierzu Gleichung 2.29, S. 51). Aus der so entstandenen Menge an Matchkeys werden anschließend alle redundanten Matchkeys entfernt und auf einen gemeinsamen Matchkey reduziert. Existieren bspw. zwei Matchkeys, die sich nur in der Ausprägung des Vornamens unterscheiden, so lässt sich der Matchkey auf alle gemeinsamen Merkmale ohne den Vornamen reduzieren.[44]

Randall et al. (2019) beschreibt jedoch weder wie sich der Schwellenwert S am besten bestimmen lässt noch wie Matches klassifiziert werden. Für die Simulationsdaten wurde der Schwellenwert daher manuell bestimmt. Die Klassifizierung erfolgt über die Zahl der übereinstimmenden Matchkeys. Nicht eindeutige Zuweisungen werden auch hier als non-Match klassifiziert.

Die m- und u-Parameter wurden für jede Merkmalskombination mit dem Paket *recordlinkage* geschätzt (Abschnitt 2.3.2). Die Schätzungen erfolgen einmalig ohne Blocking-Regeln mit einer Stichprobe von 10,000 Records des ersten Datensatzpaares (2000, 2001). Alle Merkmale werden exakt miteinander verglichen. Es werden daher keine Ähnlichkeitsmaße angewendet.

[44] In der Implementation des Verfahrens wurden die Matchkeys mit dem Originalcode von Randall bestimmt (implementiert in Python). Aus Urheberrechtsgründen befindet sich dieses Codesegment daher nicht im Anhang.

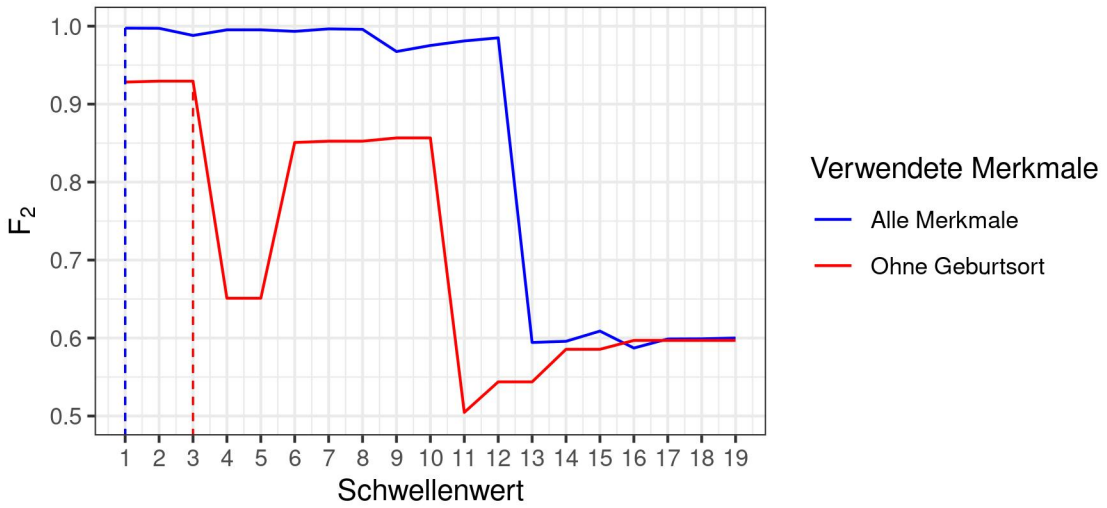

Abbildung 14: Ermittlung der besten Schwellenwerte anhand des F_2-Werts für eine Reihe an Schwellenwerten. Durchgeführt am ersten Datensatzpaar (2000, 2001) mit der Fehlerquote 0.1%. Die zwei gestichelten waagerechten Linien kennzeichnen die Maxima.

Das Verfahren wird mit zwei Merkmalskombinationen durchgeführt (siehe Tabelle 8). Die Bestimmung der Schwellenwerte für diese Merkmalskombinationen erwies sich als nicht trivial. Daher wurden die Schwellenwerte für die erste Fehlerquote (0.1%) und das erste Datensatzpaar (2000, 2001) bestimmt und auf die weiteren Fehlerquoten übertragen. Hierzu wurde das Datensatzpaar mit der Multiple-Matchkey-Methode verknüpft und die Linkage-Ergebnisse für eine Spanne von Schwellenwerten ausgewertet.

In den meisten Fällen gilt

$$\log_2\left(\frac{m_i}{u_i}\right) > 0,$$

sodass nur nach positiven Schwellenwerten gesucht wurde. Da keine Informationen über die zu erwartende Größe der Schwellenwerte vorlag, wurde im Ganzzahlbereich zwischen 1 und 19 nach passenden Werten gesucht. Die Ergebnisse dieser Verküpfungen befinden sich in Abbildung 14. Die Maximalwerte (gemessen am F_2-Wert) sind mit einer waagerechten Linie gekennzeichnet.[45] Diese Schwellenwerte werden für das Linkage aller Datensätze verwendet.

Auffällig an den Ergebnissen sind die niedrigen Schwellenwerte, die zu einer hohen Linkage-Qualität führen. Hier ist von entscheidender Bedeutung, dass der aktuelle Datensatz immer eine perfekte Teilmenge des Folgejahresdatensatzes ist. Eindeutige Links, die durch sehr schwache Matchkeys gefunden wurden (z. B. Vorname und Geburtstag) haben daher immer noch eine hohe Wahrscheinlichkeit für einen echten Match (true Positive). Dies gilt nicht für Datensätze mit einem geringeren Overlap. Es bedarf daher weiterer Forschung, wie gut die Methode beim Vorliegen von Overcoverage funktioniert.

[45] F_2 wurde zur vereinfachten Darstellung verwendet. Die Schlussfolgerung der Ergebnisse ist für eine getrennte Betrachtung von Precision und Recall gleich.

Datenbestand	Verwendete Merkmale	Schwellenwert
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht, Geburtsort	1
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht	3

Tabelle 8: Verwendete Merkmale für das Multiple-Matchkey-Verfahren. Einige Schreibweisen wurden für die Darstellung verkürzt.

2.3.5 Signaturen

Von Ranbaduge et al. (2021) wurde ein bisher nicht publiziertes Verfahren vorgeschlagen, welches Datensätze anhand einer Menge von Signaturen verknüpft. Die Idee einer Signatur geht auf Zhang et al. (2018) zurück. Dort wird eine Signatur wie folgt definiert:

> „If two records share a common subrecord that can be used to uniquely identify an entity, then these two records can be linked no matter what data quality issues they each have. We call such a subrecord a signature of its entity“ (Zhang et al. 2018: 2).

Zhang et al. (2018) konkretisieren Signaturen darüber hinaus als eine zusammengesetzte Zeichenkette. Diese setzt sich aus einer vordefinierten Reihenfolge an Merkmalen zusammen, sodass sie exakt oder über ein Ähnlichkeitsmaß verglichen werden können. Prinzipiell ist dabei jede Merkmalskombination möglich, jedoch haben einige Kombinationen eine deutlich höhere Wahrscheinlichkeit, eindeutige Records identifizieren zu können. Diese Kombination werden als *candidate signatures* bezeichnet. Die Bestimmung der candidate signatures erfolgt nach Zhang et al. (2018) manuell.

Zhang et al. (2018) schlagen zudem ein Verfahren vor, mit dem die Wahrscheinlichkeit berechnet werden kann, dass zwei Records die gleichen Signaturen teilen (Match-Wahrscheinlichkeit). Anhand dieser Wahrscheinlichkeit erfolgt dann die Klassifizierung.

Ranbaduge et al. (2021) erweitert dabei das Konzept um die Möglichkeit, Signaturen probabilistisch zu bestimmen. Eine Signatur ist in dieser konkreten Implementation eine Aneinanderreihung von vollständigen Merkmalen, wobei die Merkmale exakt miteinander verglichen werden. Eine Signatur ist somit mit einem Matchkey gleichzusetzen, sodass die Methode – in dieser konkreten Implementation – mit der Multiple-Matchkey-Methode vergleichbar ist. Die Methoden unterscheiden sich jedoch in der Selektion der Matchkeys/Signaturen sowie der Match-Wahrscheinlichkeit, die für die Multiple-Matchkey-Methode nicht vorliegt.[46]

Eine Signatur wird selektiert, falls die Completeness und der Gini-Impurity-Koeffizient[47] über einem Schwellenwert liegen. Da keine fehlenden Werte simuliert wurden, ist die Completeness – die Zahl

[46] Darüber hinaus verwendet Ranbaduge et al. (2021) Graphen, um Recordpaare in einem Kontext (z. B. Haushalt) zu vergleichen, falls mehrere eindeutige Signaturen gefunden wurden. Da ein solcher Kontext in den Simulationsdaten nicht existiert, wurden jedoch keine Graphen verwendet.

[47] Siehe hierzu Han und Kamber (2012: 341).

Datenbestand	Verwendete Merkmale
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht, Geburtsort
Restmenge	Vorname, Name, Tag, Monat, Jahr, Geschlecht

Tabelle 9: Verwendete Merkmale für die Signaturen. Einige Schreibweisen wurden für die Darstellung verkürzt.

Records mit fehlenden Werten – für alle Merkmale eins. Darüber hinaus sollen Signaturen möglichst eindeutig sein. Dies hat zur Folge, dass Signaturen nicht reduziert werden – wie es bei den Multiple-Matchkeys der Fall ist. Stattdessen werden alle Teilmengen einer selektierten Signatur aus der Menge der möglichen Signaturen ausgeschlossen. Am Ende werden die n besten Signaturen verwendet, wobei n vorher als Parameter definiert wird.

Für die Klassifizierung wird eine Ähnlichkeit zweier Signaturen nach der von Zhang et al. (2018) vorgestellten Methode bestimmt. Liegt der berechnete Wert über einem Schwellenwert, so werden zwei Records als Match klassifiziert.

Die simulierten Daten werden mit der Originalimplementation von Ranbaduge verknüpft.[48] Es werden die vorkonfigurierten Schwellenwerte verwendet ($\tau = 0.8$, $\rho = 0.8$, $weight = 0.6$). Die Implementation des Verfahrens verfügte nicht über die Möglichkeit, die candidate signatures mit einem getrennten Datensatz zu bestimmen. Daher werden diese immer mit den Restmengen und nicht mit den vollständigen Datensätzen bestimmt. Die Bestimmung der candidate signatures erfolgt bei jedem Datensatzpaar neu. Die verwendeten Merkmalskombinationen befinden sich in Tabelle 9.

[48] Aus Urheberrechtsgründen befindet sich der Programmcode (implementiert in Python) nicht im Anhang.

Kapitel 3

Ergebnisse

Alle Ergebnisse der angewendeten Record-Linkage Verfahren befinden sich in Abbildung 15 bis 19. Die Plots bilden jeweils Precision oder Recall der einzelnen Verfahren nach Fehlerquote und Jahr ab. Darüber hinaus werden getrennte Ergebnisse für Einheimische und Migranten dargestellt, anhand derer der Linkage-Bias zu erkennen ist.[49]

An dieser Stelle gilt anzumerken, dass ca. 7%-10% der Personen in der Simulation Migranten sind. Somit haben Migranten ein deutlich geringeres Gewicht auf das Gesamtergebnis. Daher sind die Gesamtergebnisse immer sehr nahe an den Ergebnissen für Einheimische.

3.1 Gesamtmenge

In Abbildung 15 und 16 werden Precision und Recall für die drei Verfahren dargestellt, die auf der Gesamtmenge operieren. Da alle deterministische Verfahren sind, ist die Precision – wie zu erwarten – sehr hoch. Unterschiede sind nur im Promillebereich feststellbar. Bei den höheren Fehlerquoten wird jedoch sichtbar, dass das Fehlen des Geburtsortes einen Linkage-Bias erzeugt. Durch die Größe des Registers sollte beachtet werden, dass eine solche geringe Abweichung immer noch mehrere Tausend falsch positive Fälle umfasst.[50] Auswirkungen auf Analyseergebnisse sind durch diesen geringen Prozentsatz jedoch nicht zu erwarten.

Anhand des Recall-Plots (Abbildung 16) können die unterschiedlichen Fehlerquoten sehr gut erkannt werden. So sinkt der Recall mit dem Anstieg der Fehler. Darüber hinaus ist die Verdopplung der Fehler-

[49] Ob ein Migrant oder Einheimischer richtig gelinkt wurde, entscheidet sich immer auf Basis des vorhandenen Records (Vorjahres-Record). Wurde z. B. ein Einheimischer aus dem Vorjahr mit einem Migranten aus dem aktuellen Datensatz gelinkt, so zählt dies als Falschklassifizierung eines Einheimischen.

[50] Für das Linkage zwischen 2008-2009 und einer Fehlerquote von 1% sind es 2409 falsch positive Migranten bei insgesamt 26,114,282 true Matches.

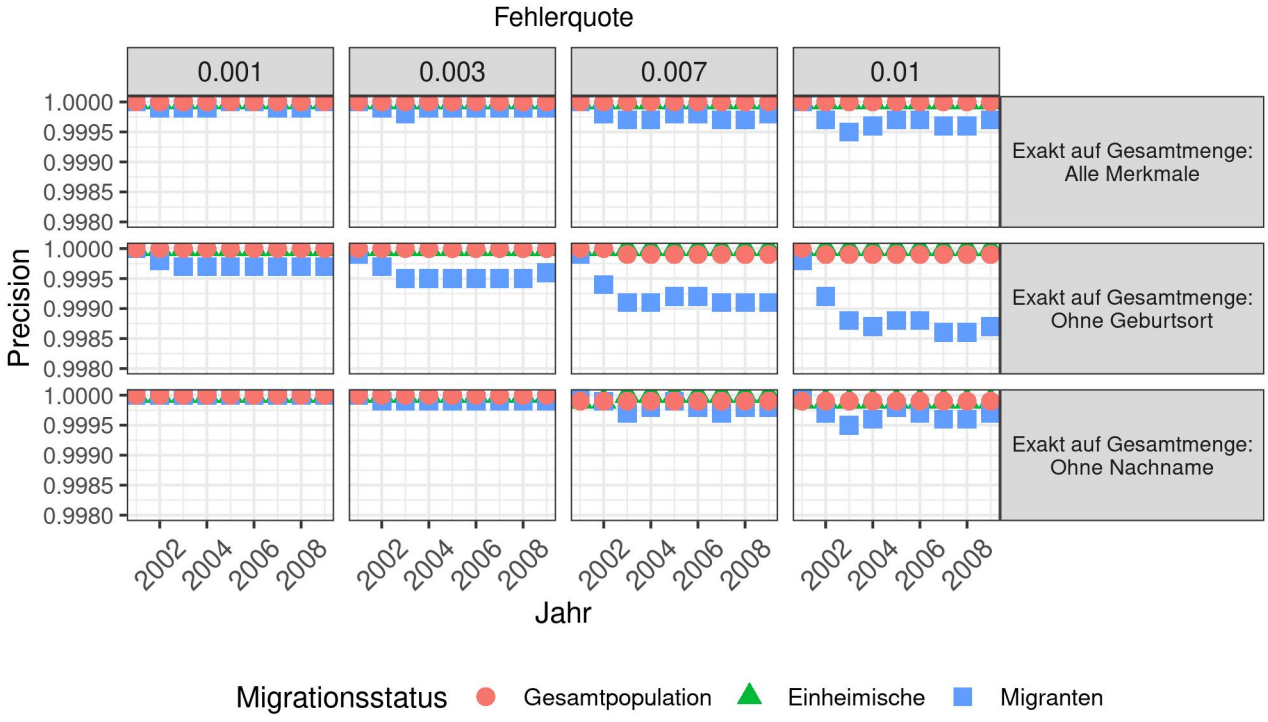

Abbildung 15: Precision aller Linkage-Verfahren, die auf dem vollständigen Datensatz operieren.

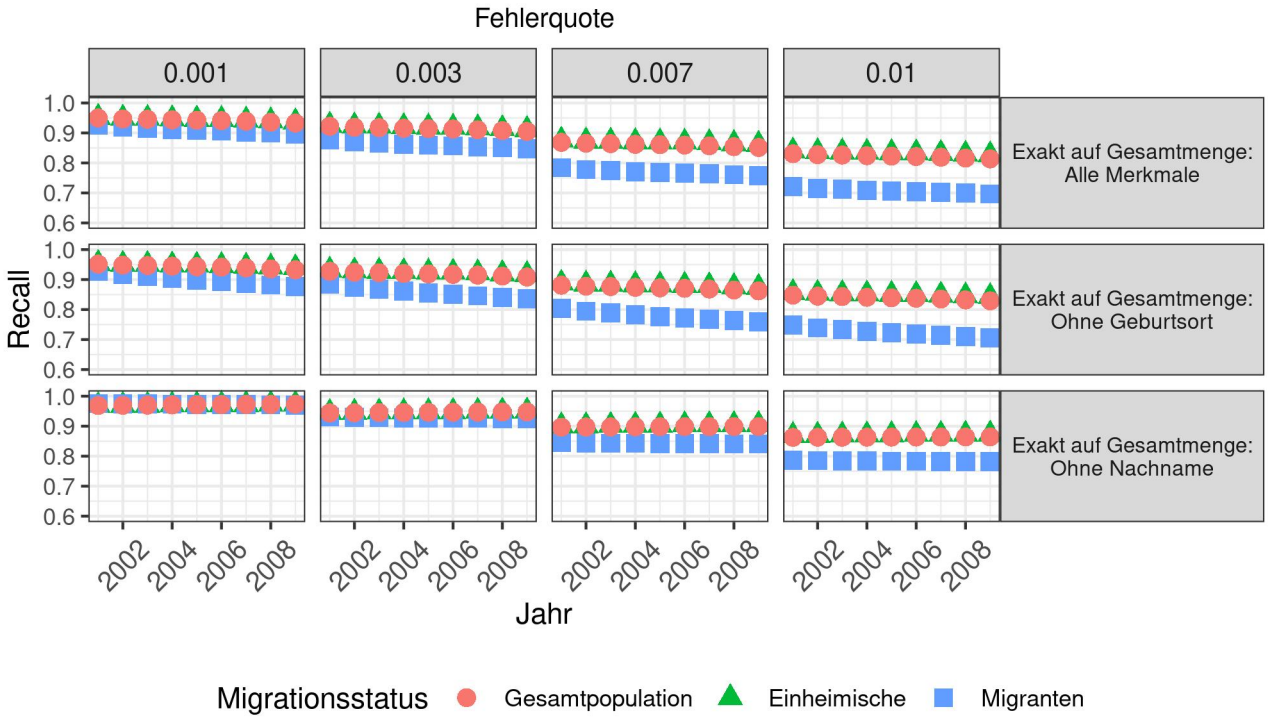

Abbildung 16: Recall aller Linkage-Verfahren, die auf dem vollständigen Datensatz operieren.

quote für Migranten durch die schlechteren Ergebnisse erkennbar. Dies führt jedoch nicht zwangsläufig zu einem Linkage-Bias, da die falsch Negativen immer noch durch ein mehrstufiges Linkage-Verfahren richtig klassifiziert werden können.

Die Recall-Ergebnisse zeigen auch die Auswirkung der Ehen auf das Linkage. So kann durchgehend ein höherer Recall erzeugt werden, wenn der Nachname kein Linkage-Merkmal ist.

Wird ein mehrstufiges Linkage-Verfahren angewendet, so hat die Precision des ersten Schritts einen höheren Stellenwert als der Recall. Da die Precision bei der Verwendung von allen Linkage-Merkmalen am höchsten ist, ist dies somit der beste erste Schritt. Ein Verzicht auf den Nachnamen als Linkage-Merkmal ist somit für den ersten Schritt *keine* bessere Entscheidung.

Insgesamt spiegeln die Ergebnisse der drei deterministischen Verfahren auf der Gesamtmenge größtenteils die Erwartungen wider. Lediglich ein minimaler Linkage-Bias (bemessen an der Precision) kann festgestellt werden, wenn der Geburtsort fehlt. Aufgrund des niedrigen Recalls, sind daher die weiteren Verfahren, welche auf der Restmenge operieren, von höherer Relevanz.

3.2 Restmenge

Da für die Verfahren, die auf der Restmenge operieren, eine mehrstufige Linkage-Strategie angenommen wurde, werden nur die Endergebnisse beider Schritte dargestellt. Somit wurden die Linkage-Ergebnisse des genannten Verfahrens mit den Ergebnissen (true Positives und false Positives) des vollständigen Matchkeys (alle Merkmale) auf der Gesamtmenge addiert (Abbildung 15 und 16 Zeile 1). In Abbildung 17 und 18 werden die Precision-Ergebnisse dargestellt; in Abbildung 19 und 20 die Recall-Ergebnisse.

3.2.1 Einfache Matchkeys

Abbildung 17 fasst die Precision-Ergebnisse für alle deterministischen Verfahren zusammen. Für die einfachen Matchkeys (Zeilen 1-6) ist die Precision wie zu erwarten sehr gut. Auch hier ist ein leichter Verlust der Precision zu erkennen, wenn der Geburtsort fehlt (Zeilen 1, 5 und 6).

Auch der Recall ähnelt den Gesamtmengen Ergebnissen (Abbildung 19). Die besten Ergebnisse für die einfachen Matchkey-Verfahren konnten erzielt werden, wenn der Nachname weggelassen wird. Darauf folgt der SLK-581 und der Swiss ALC. Die schlechtesten Recall-Werte entstehen beim Fehlen des Geburtsortes (Abbildung 19 Zeile 1). Alle einfachen Matchkey-Verfahren weisen einen starken Linkage-Bias auf. Dieser kann nicht durch die Verwendung von phonetischen Codierungen oder der Reduktion von Linkage-Merkmalen aufgelöst werden.

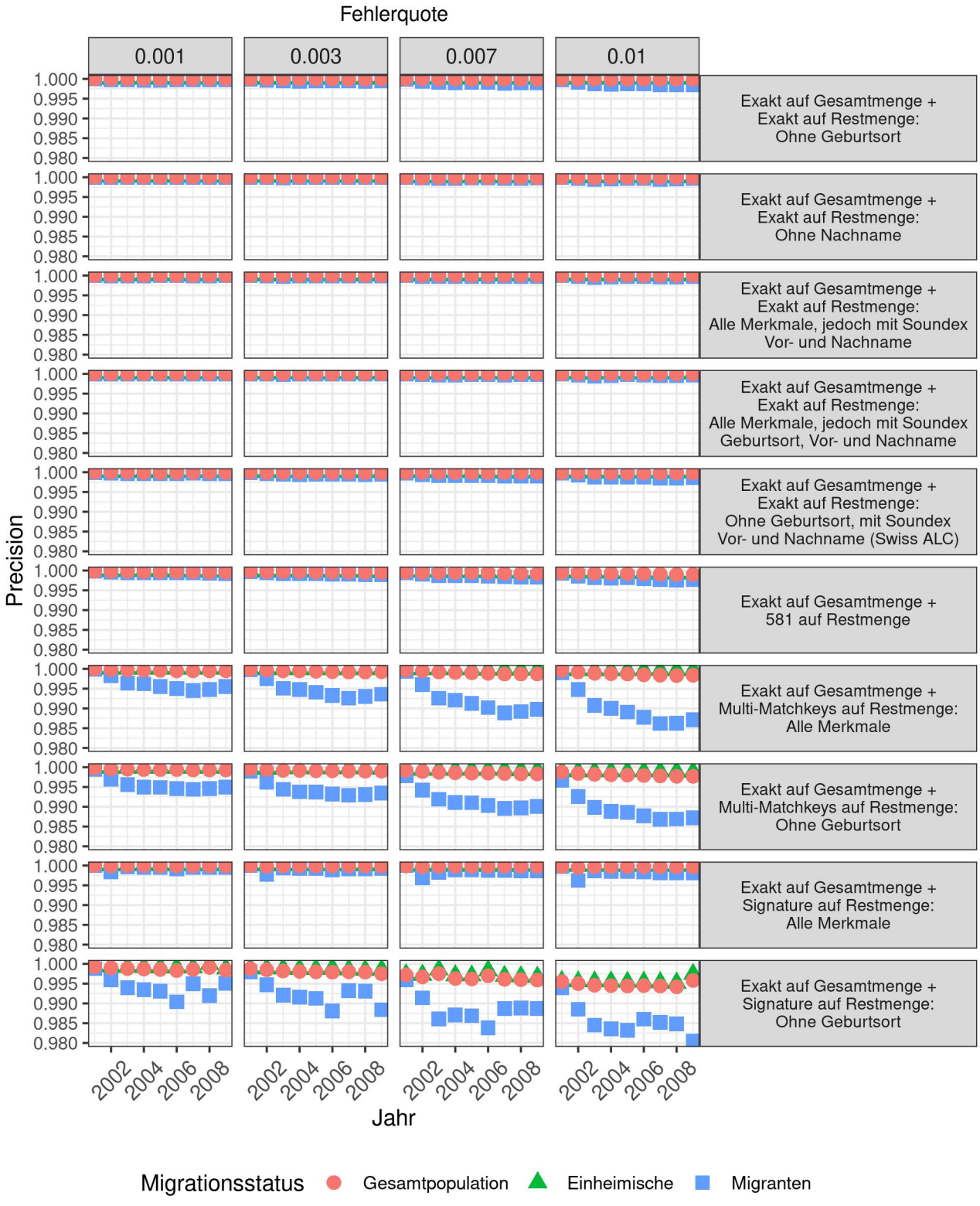

Abbildung 17: Precision aller mehrstufigen Linkage-Verfahren (Teil 1). Diese operieren immer auf der Restmenge an Records, die nicht exakt auf allen Merkmalen verknüpft werden konnten.

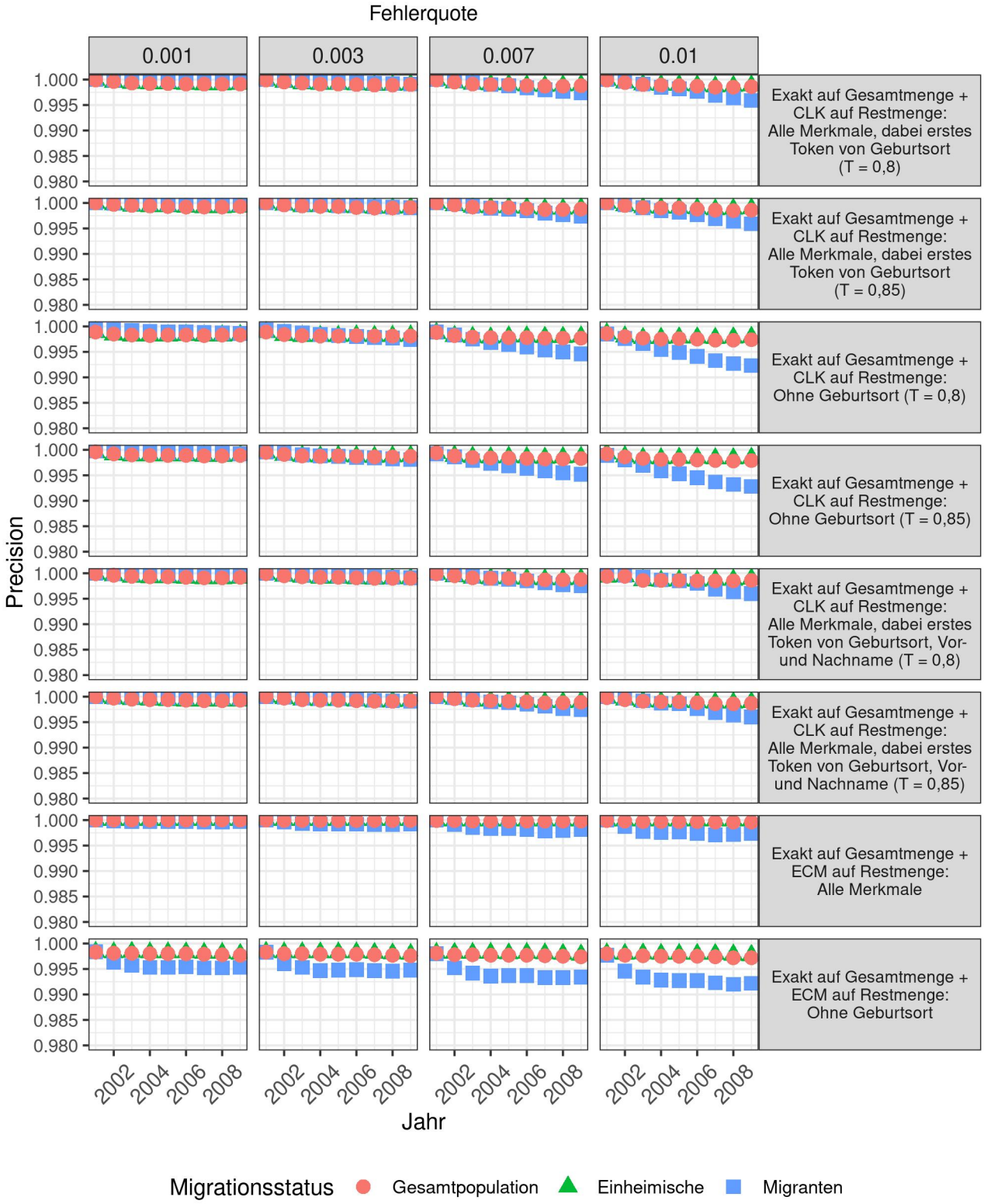

Abbildung 18: Precision aller mehrstufigen Linkage-Verfahren (Teil 2). Diese operieren immer auf der Restmenge an Records, die nicht exakt auf allen Merkmalen verknüpft werden konnten. Die Tanimoto-Distanz wurde mit T abgekürzt.

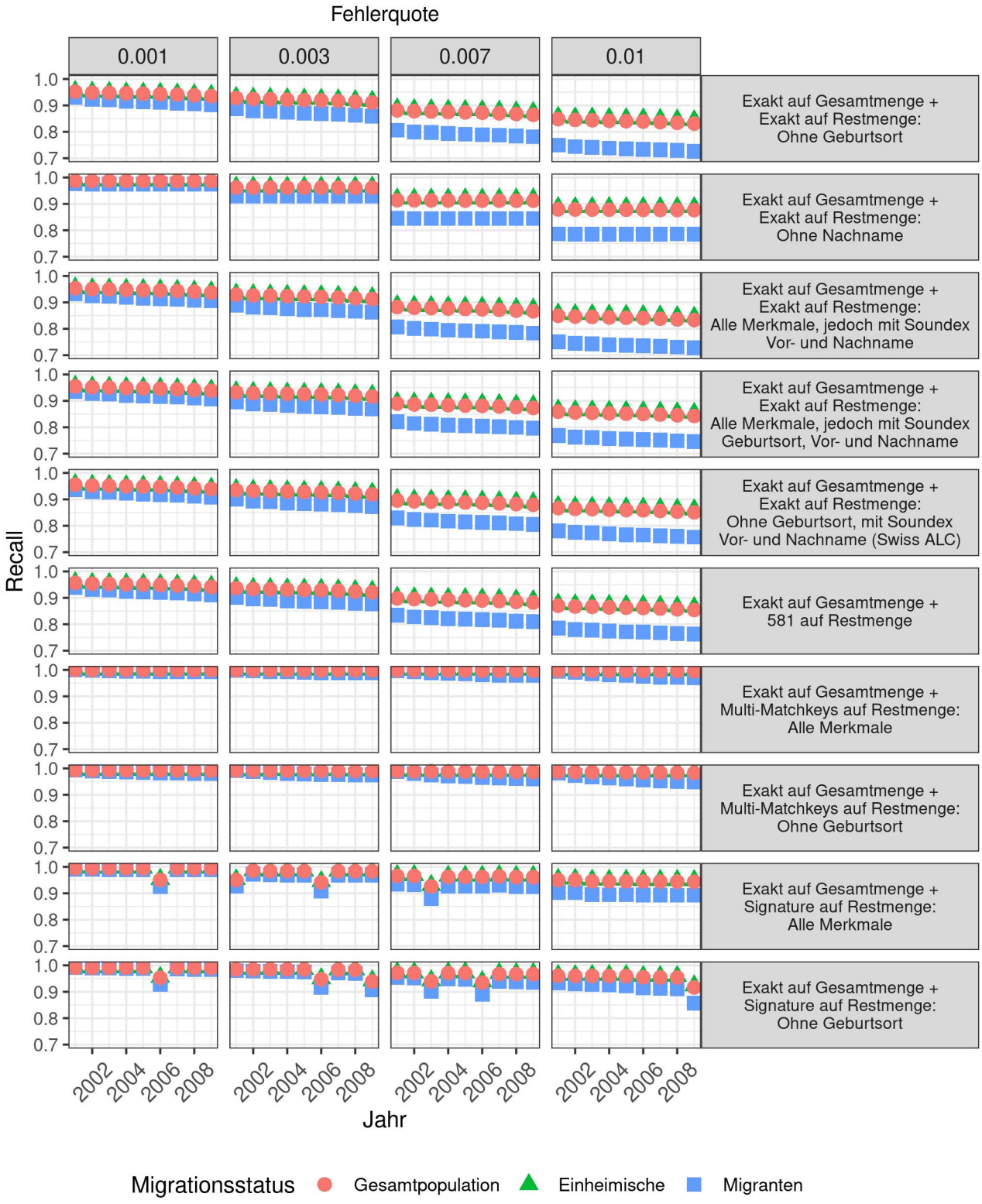

Abbildung 19: Recall aller mehrstufigen Linkage-Verfahren (Teil 1). Diese operieren immer auf der Restmenge an Records, die nicht exakt auf allen Merkmalen verknüpft werden konnten.

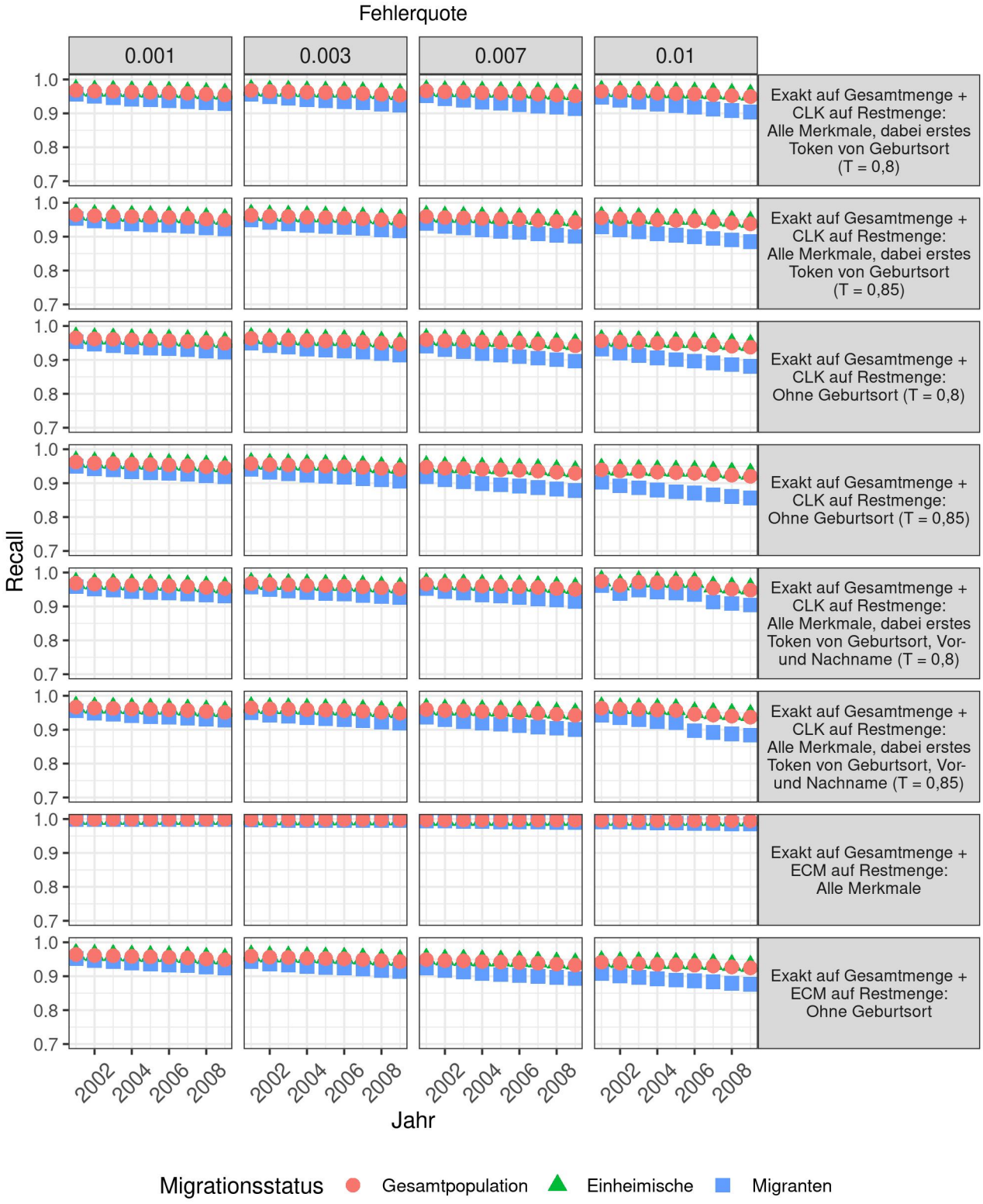

Abbildung 20: Recall aller mehrstufigen Linkage-Verfahren (Teil 2). Diese operieren immer auf der Restmenge an Records, die nicht exakt auf allen Merkmalen verknüpft werden konnten. Die Tanimoto-Distanz wurde mit T abgekürzt.

3.2.2 Multiple Matchkeys und Signaturen

Wie bei der Diskussion der Methoden festgestellt werden konnte, ähneln sich diese beiden Verfahren sehr. Daher werden sie zusammen analysiert.

Die Precision beider Verfahren zeigt einen Linkage-Bias, wenn der Geburtsort fehlt (Abbildung 17 Zeilen 8 und 10). Darüber hinaus ist das Gesamtergebnis für beide Methoden schlechter. Für das Multiple-Matchkey-Verfahren bleibt der Bias bei der Verwendung aller Merkmale bestehen (Zeile 7). Bei Signaturen ist dieser in diesem Fall nur in äußerst geringem Maße vorhanden (Zeile 9). Werden alle Linkage-Merkmale genutzt, so erreichen beide Methoden eine sehr hohe Precision, die mit den einfachen Matchkeys vergleichbar ist.

Beide Verfahren erzielen insgesamt einen hohen Recall. Die Multiple Matchkeys erzeugen dabei die besten Werte über alle Verfahren hinweg. Auch hier ist ersichtlich, dass das Fehlen des Geburtsortes dazu führt, dass sich die Ergebnisse insgesamt verschlechtern und ein Linkage-Bias entsteht. Der Bias bleibt bei beiden Methoden bestehen, unabhängig davon, ob der Geburtsort verwendet wird.

Mehrere Anmerkungen lassen sich zu den Ergebnissen machen: Bei den Signaturen sind immer wieder Ausreißer zu erkennen. Diese sind höchstwahrscheinlich Artefakte, die durch eine falsche Auswahl der candidate Signatures auf Basis der Restmenge oder durch falsche Parameterwahl entstanden sind.[51] Aufgrund der Neuheit des Verfahrens sollen diese Artefakte jedoch nicht inhaltlich interpretiert werden.[52] Weiter gilt es anzumerken, dass insbesondere die Multiple Matchkeys höchst effizient implementiert werden können. So betrug die Laufzeit für jedes Linkage nicht mehr als zwölf Minuten.[53]

Aufgrund der besseren Linkage-Ergebnisse sind beide Methoden den einfachen Matchkeys vorzuziehen.[54] Insbesondere die geringen Laufzeiten und sehr guten Linkage-Ergebnisse sprechen dafür. Es muss jedoch erneut darauf hingewiesen werden, dass ein Vorjahresdatensatz immer eine perfekte Teilmenge eines Folgejahresdatensatzes ist. Da dies der Fall ist, sind die Multiple Matchkeys die eindeutig beste Methode. Es ist jedoch zu erwarten, dass die Methode deutlich schlechtere Ergebnisse erzielt, falls die Datensätze eine geringere Schnittmenge haben. In diesem Fall liefern höchstwahrscheinlich Si-

[51] Aus den Ergebnissen ist erkennbar, dass bei den Ausreißern andere candidate Signatures gewählt wurden als bei den Fällen mit besseren Ergebnissen. Hier zeigt sich auch, dass meist die gleichen candidate Signatures gewählt werden. Eine einmalige Auswahl würde daher reichen.

[52] Zur Prüfung dieser Hypothese wurde versucht, die Signaturen auf der Gesamtmenge zu berechnen. Die Laufzeit des Programms betrug dabei über zwei Wochen und endete mit einem Speicherfehler. Die Hypothesen können somit nicht mit dem vorliegenden Programm bestätigt werden.

[53] Diese Zeit beinhaltet nicht die Parameterschätzung und Schwellenwert-Auswahl. Die Implementation wurde zudem nicht parallelisiert. Die Implementation der Signaturen wies hingegen Laufzeiten von 1-80 Stunden auf. Da die Signaturen auf einem anderen, langsameren Rechner berechnet wurden, mehrere Linkage-Verfahren gleichzeitig gestartet und bei jedem Verfahren Graphen gebildet und candidate Signatures ausgewählt wurden, sollten die Zahlen nicht zu genau verglichen werden. Darüber hinaus lag bei der Implementation der Signaturen nicht die Effizienz im Fokus. Eine deutliche Effizienzsteigerung ist höchstwahrscheinlich möglich.

[54] In der hier vorgestellten Form sind Multiple Matchkeys für PPRL ungeeignet, da die Methode ohne eine Modifikation angreifbar ist (Vidanage et al. 2020).

gnaturen die besseren Ergebnisse. Für die Überprüfung dieser Hypothese ist weiterführende Forschung notwendig.

Ein großer Vorteil der Multiple Matchkeys ist die Einfachheit des Verfahrens. Die Eigenschaft, dass eine Methode leicht erklärbar ist, kann bspw. bei Verhandlungen über den Datenschutz mit Juristen hilfreich sein. Darüber hinaus bietet auch diese Methode zahlreiche Potenziale. So könnten phonetische Codierungen oder eine Selektion von Zeichen – wie beim SLK-581 – die Ergebnisse noch weiter verbessern. Die größte Schwäche der Methode liegt bisher in der Auswahl eines geeigneten Schwellenwerts für die Selektion der Matchkeys. Auch die Signaturen haben noch zahlreiche nicht genutzte Potenziale, wie die Möglichkeiten der Graphen-Bildung und die Verwendung von Ähnlichkeitsmaßen.

Insgesamt zeigte sich, dass die Selektion der geeigneten Matchkeys bzw. candidate Signatures bei beiden Methoden nicht optimal funktionierte. Ähnliche Methoden wie die des Office for National Statistics (ONS; Shipsey und Plachta 2021) verwenden daher auch vordefinierte Matchkeys. Aus den genannten Ergebnissen schließend scheint dies eine legitime Verfahrensweise zu sein, falls genügend Informationen über die Daten vorliegen, um die Matchkeys zu bestimmen. Da das ONS-Verfahren nicht getestet wurde, kann diese Hypothese jedoch nicht verifiziert werden. Alle genannten Probleme und Potenziale stellen somit Aufgaben für zukünftige Forschung dar.

3.2.3 CLKs mit Multibit Trees

Abbildung 18 zeigt die Precision-Ergebnisse aller probabilistischen Verfahren. Wie zu erwarten ist die Precision der CLKs höher, wenn ein höherer Schwellenwert zur Klassifikation verwendet wird. Insgesamt ist die Precision auf einem sehr hohen Niveau. Das Verwenden des ersten Tokens von Vor- und Nachname hat dabei keinen Einfluss auf die Precision. Fehlt jedoch der Geburtsort, so ist ein Linkage-Bias bei den höheren Fehlerquoten zu erkennen.

Die Gesamtergebnisse des Recalls (Abbildung 20) sind für ein verschlüsseltes Verfahren sehr gut. Bei allen Verfahren zeigt sich ein Linkage-Bias, jedoch ist dieser wieder am höchsten, wenn der Geburtsort fehlt. Darüber hinaus lässt sich erkennen, dass die besten Ergebnisse erzielt werden, wenn die vollständigen Namen verwendet werden.

Auffällig ist die Verschlechterung aller CLK Ergebnisse über die Simulationsjahre. Die Ergebnisse der CLKs sind somit stärker von der Größe des Datensatzes abhängig als es bei den anderen Methoden der Fall ist. Insgesamt sind die Ergebnisse für eine PPRL Methode sehr gut. Nach den hier ermittelten Ergebnissen sind CLKs somit für PPRL empfehlenswert.

3.2.4 Probabilistisches Record-Linkage mit ECM

Werden alle Linkage-Merkmale verwendet, so liefert der ECM die zweitbesten Ergebnisse über alle Verfahren hinweg (nach den Multiple Matchkeys). Insbesondere der Recall ist unabhängig von der Fehlerquote immer sehr hoch. Die Precision der Migranten bricht hingegen bei hohen Fehlerquoten leicht ein und erzeugt einen geringen Bias.[55]

Besonders beim probabilistischen Linkage zeigen sich die Auswirkungen eines fehlenden Geburtsorts deutlich. In diesem Fall sind sowohl Precision als auch Recall mit den Ergebnissen der CLKs vergleichbar, nicht jedoch mit den Ergebnissen, wenn der Geburtsort vorhanden ist. Zudem zeigt sich ein deutlicher Bias zwischen Migranten und Einheimischen bei Precision und Recall, wenn der Geburtsort nicht vorhanden ist.

Darüber hinaus muss erwähnt werden, dass die Methode auch für Datensätze mit einer geringen Schnittmenge sehr gut funktioniert. In diesen Fällen ist sie den Multiple Matchkeys vorzuziehen.

3.3 Linkage-Bias für abhängige Variablen

Wie gezeigt werden konnte, verknüpfen die meisten Verfahren Migranten mit weniger Erfolg. Die Linkage-Biases sind jedoch durch die Unterscheidung von Precision und Recall sowie durch die Zeitachse schwer vergleichbar. Daher wird im Folgenden der Bias genauer analysiert.

Das Ergebnis eines nicht erfolgten Links ist ein fehlender Wert im gelinkten Datensatz. Bei fehlenden Werten wird allgemein zwischen drei Typen unterschieden (Collins et al. 2001; Graham 2009; Schnell 2019):

1. „*Missing completely at random*“ (MCAR): Der fehlende Wert lässt sich durch keine bekannte Variable erklären. Die Ausfälle sind vollkommen zufällig.
2. „*Missing at random*“ (MAR): Der fehlende Wert lässt sich durch eine bekannte Variable erklären. Der Ausfall für ein Record mit dieser Variablenausprägung ist jedoch zufällig.
3. „*Missing not at random*“ (MNAR): Der fehlende Wert lässt sich nur durch den fehlenden Wert selbst erklären.

Nicht erfolgte Links können in den allermeisten Fällen MAR zugeordnet werden. Im Kontext des Linkage-Bias ist die Wahrscheinlichkeit eines fehlenden Werts abhängig von der Zugehörigkeit zu

[55] Dieses Verhalten deutet auf einen zu niedrig gewählten Schwellenwert hin (0.8). So könnte wahrscheinlich die Precision verbessert werden, wenn dieser Wert höher gesetzt wird. Dies würde gleichzeitig den Recall verändern. Die tatsächlichen Effekte einer Änderung des Schwellenwerts kann im Rahmen dieser Arbeit nicht überprüft werden.

einer (Sub-)Population. Ausfälle bei Migranten und Einheimischen erfolgen somit zufällig innerhalb der beiden Gruppen.

Auf dieser Basis ist es möglich, den Effekt der unterschiedlichen Linkage-Qualitäten für Migranten und Einheimische auf abhängige Variablen zu berechnen. So kann gezeigt werden, wie stark bspw. der Notenschnitt eines Jahrgangs vom Populationswert abweicht, weil ein Linkage-Verfahren Migranten schlechter verknüpft. Der Unterschied dieser beiden Werte ist der Linkage-Bias.

Die Ursache für einen nicht erfolgten Link sind Veränderungen eines Records. Da diese simuliert wurden, ist es nun möglich zu ermitteln, welchen Einfluss diese Veränderungen auf die Linkage-Ergebnisse haben. Die wesentlichen Ereignisse, die eine Veränderung verursachen, sind: Schulwechsel, Umzüge und Ehen. Daraus können drei abhängige Variablen abgleitet werden:

1. Bildungsdauer,
2. Anzahl der Umzüge und
3. Anzahl der Ehen.

Die Variablen sind nicht unabhängig voneinander. Insbesondere die Bildungsdauer korreliert mit den anderen Variablen. Dennoch ist es möglich, den Einfluss der Verfahren auf diese abhängigen Variablen zu beurteilen.

Da das Bildungsregister verwendet wird, um Bildungsverläufe zu analysieren, wird der Linkage-Bias auf genau diese untersucht. Die Verläufe entstehen durch das wiederholte Verknüpfen der Datensätze. Das Ergebnis ist daher ein *Längsschnittdatensatz*. Zur Vergleichbarkeit besteht dieser Längsschnittdatensatz nur aus durchgehend verknüpften Records. Alle Records, die in *jedem* Jahr true oder false Positive klassifiziert wurden. Wurde somit ein Record in einem Jahr nicht klassifiziert, so wird er aus dem Längsschnittdatensatz entfernt – auch wenn es zu einem späteren Zeitpunkt wieder verknüpft wird. Darüber werden jedes Jahr alle Neuzugänge dem Längsschnittdatensatz hinzugefügt.

Zur Berechnung des Bias wird der Standardized Bias verwendet (Collins et al. 2001). Dieser wird definiert als

$$\text{Standardized Bias} = 100 \cdot \frac{\overline{x}_j - \mu_j}{s_j}, \tag{3.1}$$

wobei s_j und $\overline{x}_j$ Standardabweichung und Mittelwert der durchgehend gelinkten Population bis zum Jahr j sind und μ_j der Populationsmittelwert zum Zeitpunkt j ist. Der Standardized Bias gibt daher die prozentuale Abweichung vom Populationsmittelwert in Standardabweichungen an. Ein kritischer Effekt auf eine Schätzung ist nach Collins et al. (2001) ab einer Abweichung von $\pm 40\%$ zu erwarten.

Die Standardized Biases aller Verfahren, abhängigen Variablen und (Sub-)Populationen werden für alle gelinkten Jahre ($j = 2009$) berechnet. Der Bias der drei abhängigen Variablen wird jeweils für die Gesamtpopulation sowie für Migranten und Einheimische getrennt berechnet. Darüber hinaus wird

bei der Anzahl der Ehen zusätzlich das Geschlecht unterschieden, da die Namensänderung bei der Ehe (siehe Abschnitt 1.3.3) abhängig vom Geschlecht ist.

Die Ergebnisse werden in Tabelle 10 dargestellt (Werte über ±40% sind hervorgehoben). Eindeutig ist zu erkennen, dass bei den meisten Verfahren der Bias für die Anzahl der Ehen am größten ist, insbesondere für weibliche Migranten. In vielen Fällen beträgt dort die Abweichung über 100% – mehr als eine Standardabweichung. Dies ist besonders der Fall, wenn der Geburtsort als Linkage-Merkmal fehlt.[56] Inhaltlich bedeutet dies, dass Migrantinnen, die während ihrer Bildungslaufbahn heiraten, im Längsschnittdatensatz häufig fehlen. Dies ist ein hoch relevantes Ergebnis.

Die Zahl der Umzüge wird durch alle Verfahren wenig unterschätzt. Der Einfluss eines Umzugs auf die Linkage-Ergebnisse ist somit gering.

Auch die Bildungsdauer wird insgesamt wenig unterschätzt. Einzig die Bildungsdauer von Migranten wird bei den einfachen Matchkey-Verfahren deutlich unterschätzt. Dies lässt sich auf die doppelte Fehlerquote zurückführen.

Die insgesamt geringsten Bias-Ergebnisse haben die Multiple Matchkeys und der ECM – beide bei der Verwendung aller Merkmale. Gefolgt werden diese von den Signaturen jeweils mit allen Merkmalen sowie dem einfachen Matchkey unter der Verwendung aller Merkmale mit Soundex Vorname, Nachname und Geburtsort. Warum dieser Matchkey einen geringen Bias erzeugt, kann nicht erklärt werden.[57] Sowohl wenn der Soundex von Vor- und Nachname mit dem vollständigen Geburtsort als auch ohne Geburtsort verwendet wird, ist ein Bias vorhanden. Auch die Linkage-Ergebnisse für diesen Matchkey weichen nicht von denen der anderen Matchkeys ab. So decken sich alle weiteren Bias-Ergebnisse mit den Linkage-Ergebnissen. Es bleibt daher offen, warum der Soundex des Geburtsorts dazu geführt hat, dass der Bias sich so stark verringert. Die Ergebnisse für diesen Matchkey sollten daher nicht verallgemeinert werden, ohne dass eine Begründung dieses Verhaltens vorliegt.

[56] Die einzige Ausnahme bildet das Signatur-Verfahren. Wie bereits beschrieben waren in diesem Fall die Linkage-Ergebnissen besser, wenn dort der Geburtsort fehlt. Dementsprechend ist der Bias auch geringer, wenn der Geburtsort fehlt.

[57] Die Ergebnisse wurden mehrfach neu berechnet und auf Fehler überprüft.

Tabelle 10: Standardized Bias (in %, Gleichung 3.1) verschiedener abhängiger Variablen zwischen dem Gesamtdatensatz (gelinkte und nicht gelinkte Fälle) und den erfolgreich durchgehend gelinkten Fällen (M = Männlich, W = Weiblich). Die Ergebnisse sind für jedes Verfahren und jede Fehlerquote angegeben. Das Ergebnis ist als prozentuale Abweichung vom Populationsparameter in Standardabweichungen zu interpretieren. Abweichungen größer/kleiner ±40% sind hervorgehoben.

Methode	Fehler in %	Bildungsdauer	$\text{Bildungsdauer}_{\text{Mig.}}$	$\text{Bildungsdauer}_{\text{Ein.}}$	Umzüge	$\text{Umzüge}_{\text{Mig.}}$	$\text{Umzüge}_{\text{Ein.}}$	Ehen	$\text{Ehen}_{\text{Mig.}}$	$\text{Ehen}_{\text{Ein.}}$	Ehen_{M}	$\text{Ehen}_{\text{Mig., M}}$	$\text{Ehen}_{\text{Ein., M}}$	Ehen_{W}	$\text{Ehen}_{\text{Mig., W}}$	$\text{Ehen}_{\text{Ein., W}}$
Exakt auf Gesamtmenge	0.1	−12.379	−8.313	−12.889	−4.287	−3.628	−4.563	−26.612	**-47.661**	−25.406	−4.081	−6.529	−3.925	**-45.098**	**-117**	**-42.532**
+ Exakt auf Restmenge:	0.3	−14.466	−15.209	−14.625	−5.811	−8.511	−5.878	−27.755	**-51.329**	−26.582	−5.225	−9.397	−5.02	**-46.289**	**-122.416**	**-43.865**
Ohne Geburtsort	0.7	−18.876	−30.224	−18.396	−9.339	−18.828	−9.101	−30.325	**-60.38**	−29.173	−7.718	−16.033	−7.401	**-49.05**	**-137.963**	**-46.824**
	1	−22.419	**-42.608**	−21.504	−12.102	−27.645	−11.602	−32.489	**-67.949**	−31.335	−9.731	−21.563	−9.328	**-51.483**	**-151.052**	**-49.361**
Exakt auf Gesamtmenge	0.1	−12	−8.113	−12.492	−4.143	−3.567	−4.409	−25.98	**-46.207**	−24.81	−3.888	−6.212	−3.742	**-43.871**	**-109.51**	**-41.405**
+ Exakt auf Restmenge:	0.3	−14.038	−14.951	−14.183	−5.662	−8.443	−5.72	−27.187	**-49.997**	−26.044	−5.097	−9.217	−4.896	**-45.122**	**-114.958**	**-42.79**
Ohne Nachname	0.7	−18.374	−29.879	−17.884	−9.17	−18.725	−8.93	−29.884	**-59.457**	−28.75	−7.726	−16.273	−7.4	**-47.997**	**-129.985**	**-45.849**
	1	−21.86	−42.18	−20.94	−11.932	−27.393	−11.444	−32.182	**-67.264**	−31.043	−9.856	−22.086	−9.442	**-50.581**	**-142.533**	**-48.53**
Exakt auf Gesamtmenge	0.1	−11.904	−7.779	−12.413	−4.073	−3.348	−4.346	−25.941	**-46.053**	−24.766	−3.848	−6.109	−3.703	**-43.832**	**-109.241**	**-41.356**
+ Exakt auf Restmenge:	0.3	−13.743	−13.936	−13.934	−5.442	−7.727	−5.525	−27.064	**-49.437**	−25.915	−4.985	−8.821	−4.791	**-44.982**	**-114.019**	**-42.629**
Alle Merkmale, jedoch	0.7	−17.632	−27.15	−17.245	−8.597	−16.822	−8.408	−29.523	**-57.792**	−28.383	−7.405	−15.11	−7.097	**-47.579**	**-127.201**	**-45.397**
mit Soundex Vor- und	1	−20.738	−37.992	−19.949	−11.067	−24.465	−10.645	−31.588	**-64.466**	−30.451	−9.329	−20.18	−8.942	**-49.903**	**-137.602**	**-47.821**
Nachname																
Exakt auf Gesamtmenge	0.1	−0.884	−2.844	−0.753	−0.666	−1.96	−0.612	−0.497	−1.173	−0.472	−0.521	−1.156	−0.508	−0.511	−1.258	−0.478
+ Exakt auf	0.3	−2.739	−8.955	−2.334	−2.041	−6.065	−1.855	−1.59	−3.952	−1.488	−1.652	−3.96	−1.574	−1.648	−4.197	−1.525
Restmenge: Alle	0.7	−6.725	−22.472	−5.81	−5.089	−15.518	−4.652	−3.943	−10.388	−3.682	−4.073	−10.478	−3.854	−4.093	−10.982	−3.791
Merkmale, jedoch mit	1	−9.959	−33.668	−8.698	−7.561	−23.193	−6.939	−5.862	−16.072	−5.477	−6.003	−16.015	−5.675	−6.121	−17.076	−5.679
Soundex Geburtsort,																
Vor- und Nachname																
Exakt auf Gesamtmenge	0.1	−11.831	−7.584	−12.349	−4.013	−3.208	−4.291	−25.887	**-45.928**	−24.711	−3.807	−6.039	−3.661	**-43.76**	**-108.895**	**-41.282**
+ Exakt auf Restmenge:	0.3	−13.517	−13.276	−13.739	−5.262	−7.282	−5.359	−26.901	**-49.065**	−25.749	−4.844	−8.565	−4.651	**-44.791**	**-113.32**	**-42.429**
Ohne Geburtsort, mit	0.7	−17.073	−25.485	−16.749	−8.141	−15.623	−7.985	−29.105	**-56.631**	−27.968	−7.045	−14.293	−6.749	**-47.087**	**-125.239**	**-44.895**
Soundex Vor- und	1	−19.892	−35.482	−19.185	−10.367	−22.674	−9.988	−30.927	**-62.549**	−29.799	−8.776	−18.763	−8.413	**-49.101**	**-134.798**	**-47.01**
Nachname (Swiss ALC)																
Exakt auf Gesamtmenge	0.1	−11.501	−7.326	−12.011	−3.882	−3.101	−4.152	−25.401	**-44.828**	−24.249	−3.637	−5.774	−3.497	**-42.869**	**-103.937**	**-40.46**
+ 581 auf Restmenge	0.3	−13.129	−12.862	−13.349	−5.085	−7.061	−5.18	−26.356	**-47.737**	−25.232	−4.627	−8.183	−4.443	**-43.831**	**-107.727**	**-41.538**
	0.7	−16.528	−24.629	−16.216	−7.847	−15.109	−7.695	−28.448	**-54.769**	−27.34	−6.738	−13.649	−6.455	**-45.992**	**-117.978**	**-43.866**
	1	−19.189	−34.227	−18.505	−9.954	−21.867	−9.587	−30.182	**-60.425**	−29.083	−8.404	−17.955	−8.056	**-47.893**	**-127.077**	**-45.861**

Tabelle 10: Fortsetzung: Standardized Bias (in %) verschiedener abhängiger Variablen zwischen dem Gesamtdatensatz (gelinkte und nicht gelinkte Fälle) und den erfolgreich durchgehend gelinkten Fällen (M = Männlich, W = Weiblich).

Methode	Fehler in %	Bildungsdauer	$\text{Bildungsdauer}_{\text{Mig.}}$	$\text{Bildungsdauer}_{\text{Ein.}}$	Umzüge	$\text{Umzüge}_{\text{Mig.}}$	$\text{Umzüge}_{\text{Ein.}}$	Ehen	$\text{Ehen}_{\text{Mig.}}$	$\text{Ehen}_{\text{Ein.}}$	Ehen_{M}	$\text{Ehen}_{\text{Mig., M}}$	$\text{Ehen}_{\text{Ein., M}}$	Ehen_{W}	$\text{Ehen}_{\text{Mig., W}}$	$\text{Ehen}_{\text{Ein., W}}$
Exakt auf Gesamtmenge	0.1	0.198	1.881	−0.004	0.199	1.434	−0.002	−0.286	−1.506	−0.201	−0.047	−0.531	−0.014	−0.363	−1.518	−0.304
+ Multi-Matchkeys auf	0.3	0.196	2.415	−0.071	0.22	1.647	−0.023	−0.501	−1.988	−0.404	−0.088	−0.797	−0.049	−0.673	−2.179	−0.592
Restmenge: Alle	0.7	0.179	3.32	−0.199	0.288	2.448	−0.076	−0.966	−3	−0.845	−0.222	−1.417	−0.168	−1.324	−3.536	−1.203
Merkmale	1	0.168	3.898	−0.281	0.301	2.751	−0.123	−1.353	−3.885	−1.209	−0.372	−1.902	−0.31	−1.849	−4.747	−1.688
Exakt auf Gesamtmenge	0.1	−0.43	3.199	−0.831	0.249	2.426	−0.074	−5.67	−6.639	−5.628	−0.317	−1.308	−0.242	−8.672	−8.776	−8.742
+ Multi-Matchkeys auf	0.3	−0.577	3.665	−1.049	0.181	2.485	−0.179	−6.21	−7.769	−6.129	−0.515	−2.053	−0.411	−9.446	−10.284	−9.459
Restmenge: Ohne	0.7	−0.907	4.331	−1.496	0.133	3.179	−0.356	−7.295	−10.072	−7.141	−1.013	−3.715	−0.848	−10.952	−13.264	−10.861
Geburtsort	1	−1.19	4.592	−1.846	0.009	3.206	−0.528	−8.104	−11.869	−7.895	−1.426	−4.945	−1.221	−12.056	−15.624	−11.884
Exakt auf Gesamtmenge	0.1	−2.612	−2.038	−2.714	−0.828	−1.05	−0.874	−5.553	−8.817	−5.297	−0.647	−1.209	−0.616	−8.186	−13.098	−7.802
+ Signature auf	0.3	−3.699	−5.241	−3.648	−1.47	−3.352	−1.409	−11.392	−17.599	−10.968	−1.704	−3.256	−1.625	−17.584	−27.698	−16.901
Restmenge: Alle	0.7	−4.74	−11.861	−4.314	−2.696	−7.914	−2.473	−9.033	−16.189	−8.63	−2.567	−6.128	−2.413	−13.047	−22.871	−12.475
Merkmale	1	−5.129	−15.618	−4.528	−3.394	−10.277	−3.137	−5.271	−13.083	−4.872	−2.894	−7.599	−2.708	−6.666	−16.463	−6.148
Exakt auf Gesamtmenge	0.1	−2.575	−0.765	−2.788	−0.621	−0.157	−0.73	−9.111	−11.298	−8.95	−0.736	−1.238	−0.708	−14.072	−17.151	−13.85
+ Signature auf	0.3	−6.479	−3.713	−6.832	−2.132	−1.608	−2.329	−12.703	−18.387	−12.311	−1.565	−2.862	−1.496	−19.296	−28.229	−18.699
Restmenge: Ohne	0.7	−5.661	−7.325	−5.667	−2.293	−4.472	−2.309	−15.381	−23.479	−14.899	−2.83	−5.743	−2.693	−23.755	−36.619	−22.996
Geburtsort	1	−6.928	−9.343	−6.933	−3.097	−5.547	−3.177	−12.809	−19.327	−12.483	−3.082	−6.487	−2.957	−18.596	−27.275	−18.147
Exakt auf Gesamtmenge	0.1	−0.009	0.073	−0.023	0.002	0.08	−0.016	−0.347	−0.487	−0.338	−0.052	−0.168	−0.045	−0.508	−0.624	−0.502
+ ECM auf Restmenge:	0.3	−0.133	0.018	−0.165	−0.036	0.053	−0.072	−0.703	−1.324	−0.656	−0.15	−0.431	−0.134	−0.997	−1.755	−0.94
Alle Merkmale	0.7	−0.332	0.094	−0.409	−0.072	0.18	−0.156	−1.426	−3.12	−1.293	−0.34	−1.006	−0.298	−2	−4.213	−1.828
	1	−0.494	0.049	−0.598	−0.118	0.174	−0.23	−1.968	−4.455	−1.779	−0.516	−1.503	−0.454	−2.743	−6.038	−2.493
Exakt auf Gesamtmenge	0.1	−10.88	−5.262	−11.505	−3.383	−1.66	−3.714	−24.687	**-43.631**	−23.508	−3.39	−5.28	−3.242	**-41.519**	**-100.111**	−39.088
+ ECM auf Restmenge:	0.3	−11.261	−6.373	−11.834	−3.609	−2.685	−3.886	−24.935	**-44.501**	−23.764	−3.732	−6.253	−3.556	**-41.715**	**-100.622**	−39.334
Ohne Geburtsort	0.7	−12.055	−8.807	−12.515	−4.269	−4.187	−4.534	−25.471	**-46.581**	−24.311	−4.449	−8.466	−4.207	**-42.159**	**-101.941**	−39.876
	1	−12.691	−10.96	−13.051	−4.73	−5.775	−4.94	−25.895	**-48.179**	−24.751	−5.017	−10.195	−4.732	**-42.499**	**-103.072**	**-40.297**

Tabelle 10: Fortsetzung: Standardized Bias (in %) verschiedener abhängiger Variablen zwischen dem Gesamtdatensatz (gelinkte und nicht gelinkte Fälle) und den erfolgreich durchgehend gelinkten Fällen (M = Männlich, W = Weiblich).

Methode	Fehler in %	Bildungsdauer	$\text{Bildungsdauer}_{\text{Mig.}}$	$\text{Bildungsdauer}_{\text{Ein.}}$	Umzüge	$\text{Umzüge}_{\text{Mig.}}$	$\text{Umzüge}_{\text{Ein.}}$	Ehen	$\text{Ehen}_{\text{Mig.}}$	$\text{Ehen}_{\text{Ein.}}$	Ehen_{M}	$\text{Ehen}_{\text{Mig., M}}$	$\text{Ehen}_{\text{Ein., M}}$	Ehen_{W}	$\text{Ehen}_{\text{Mig., W}}$	$\text{Ehen}_{\text{Ein., W}}$
Exakt auf Gesamtmenge	0.1	−9.618	−3.641	−10.276	−2.826	−0.682	−3.205	−24.727	**-44.137**	−23.524	−3.264	−5.266	−3.104	**-41.913**	**-103.316**	−39.438
+ CLK auf Restmenge:	0.3	−9.607	−3.243	−10.31	−2.766	−0.753	−3.148	−24.769	**-44.363**	−23.581	−3.403	−5.765	−3.23	**-41.902**	**-102.943**	−39.465
Alle Merkmale, dabei	0.7	−9.604	−2.694	−10.375	−2.764	−0.124	−3.259	−24.897	**-45.059**	−23.74	−3.733	−6.977	−3.529	**-41.913**	**-102.305**	−39.562
erstes Token von	1	−9.633	−2.495	−10.438	−2.748	−0.165	−3.271	−25.031	**-45.827**	−23.896	−4.014	−8.07	−3.781	**-41.956**	**-102.327**	−39.674
Geburtsort (T = 0.8)																
Exakt auf Gesamtmenge	0.1	−10.942	−4.697	−11.626	−3.338	−1.2	−3.715	−25.836	**-45.81**	−24.61	−3.573	−5.612	−3.413	**-43.993**	**-111.298**	**-41.4**
+ CLK auf Restmenge:	0.3	−10.942	−4.668	−11.641	−3.295	−1.524	−3.661	−25.96	**-46.382**	−24.751	−3.809	−6.403	−3.628	**-44.043**	**-111.053**	**-41.506**
Alle Merkmale, dabei	0.7	−11.008	−5.161	−11.699	−3.432	−1.636	−3.873	−26.288	**-48.036**	−25.11	−4.374	−8.322	−4.146	**-44.191**	**-111.538**	**-41.774**
erstes Token von	1	−11.123	−6	−11.778	−3.535	−2.362	−3.97	−26.609	**-49.468**	−25.463	−4.859	−9.934	−4.6	**-44.392**	**-112.186**	**-42.077**
Geburtsort (T = 0.85)																
Exakt auf Gesamtmenge	0.1	−10.828	−4.32	−11.537	−3.288	−0.976	−3.686	−25.455	**-45.418**	−24.23	−3.569	−5.793	−3.395	**-43.08**	**-107.78**	**-40.533**
+ CLK auf Restmenge:	0.3	−10.765	−3.804	−11.53	−3.206	−0.991	−3.623	−25.605	**-46.349**	−24.38	−3.84	−7.017	−3.616	**-43.146**	**-107.314**	**-40.655**
Ohne Geburtsort (T =	0.7	−10.688	−3.3	−11.517	−3.264	−0.449	−3.81	−26.009	**-48.627**	−24.786	−4.5	−9.74	−4.179	**-43.352**	**-107.278**	**-40.971**
0.8)	1	−10.672	−3.288	−11.52	−3.292	−0.645	−3.867	−26.396	**-50.635**	−25.175	−5.064	−11.998	−4.674	**-43.599**	**-107.874**	**-41.299**
Exakt auf Gesamtmenge	0.1	−11.271	−5.071	−11.955	−3.515	−1.426	−3.894	−26.102	**-46.536**	−24.864	−3.8	−6.1	−3.625	**-44.276**	**-112.731**	**-41.672**
+ CLK auf Restmenge:	0.3	−11.324	−5.445	−11.998	−3.593	−2.024	−3.958	−26.548	**-48.096**	−25.312	−4.357	−7.725	−4.131	**-44.638**	**-113.372**	**-42.098**
Ohne Geburtsort (T =	0.7	−11.511	−6.758	−12.126	−3.981	−2.761	−4.405	−27.518	**-51.866**	−26.279	−5.546	−11.453	−5.211	**-45.43**	**-115.558**	**-43.019**
0.85)	1	−11.73	−8.201	−12.277	−4.276	−3.901	−4.685	−28.289	**-54.771**	−27.063	−6.455	−14.298	−6.057	**-46.099**	**-117.595**	**-43.791**
Exakt auf Gesamtmenge	0.1	−10.391	−4.294	−11.057	−3.116	−1.04	−3.477	−24.417	**-42.802**	−23.257	−3.1	−4.837	−2.961	**-41.31**	**-98.04**	−38.904
+ CLK auf Restmenge:	0.3	−10.38	−3.952	−11.086	−3.039	−1.126	−3.402	−24.427	**-42.93**	−23.287	−3.218	−5.279	−3.071	**-41.256**	**-97.477**	−38.896
Alle Merkmale, dabei	0.7	−10.411	−3.614	−11.171	−3.064	−0.658	−3.533	−24.479	**-43.429**	−23.382	−3.493	−6.371	−3.323	**-41.172**	**-96.537**	−38.916
erstes Token von	1	−9.646	−2.745	−10.426	−2.755	−0.299	−3.26	−22.016	−39.54	−21.02	−3.195	−6.476	−3.019	−36.083	**-79.593**	−34.161
Geburtsort, Vor- und Nachname (T = 0.8)																
Exakt auf Gesamtmenge	0.1	−10.724	−4.72	−11.383	−3.261	−1.306	−3.614	−24.913	**-43.526**	−23.751	−3.212	−4.961	−3.078	**-42.306**	**-101.325**	−39.861
+ CLK auf Restmenge:	0.3	−10.819	−4.967	−11.48	−3.264	−1.786	−3.6	−24.965	**-43.907**	−23.83	−3.378	−5.6	−3.232	**-42.289**	**-100.988**	−39.912
Alle Merkmale, dabei	0.7	−11.092	−5.933	−11.726	−3.475	−2.214	−3.872	−24.871	**-44.745**	−23.792	−3.727	−7.108	−3.552	**-41.739**	**-99.421**	−39.522
erstes Token von	1	−10.965	−5.887	−11.613	−3.383	−2.337	−3.805	−24.24	**-44.024**	−23.225	−3.772	−7.656	−3.598	**-40.319**	**-94.568**	−38.261
Geburtsort, Vor- und Nachname (T = 0.85)																

Zusammenfassung

In dieser Arbeit wurde die Mikro-Simulation eines bundesweiten Schülerregisters beschrieben und dessen Datensätze mittels Record-Linkage verknüpft. Hierzu wurde eine Zahl von Record-Linkage Methoden vorgestellt und angewendet. Eine wesentliche Fragestellung war, ob der Geburtsort ein relevantes Linkage-Merkmal ist und ob durch das Fehlen dieses Merkmals ein Linkage-Bias entsteht. Beide Fragen konnten beantwortet werden. Der Geburtsort ist ein relevantes Merkmal zur Verknüpfung des Schülerregisters.[58] So werden insbesondere Migrantinnen, die während des Bildungsverlaufs heiraten, nicht verlinkt, wenn der Geburtsort fehlt.

Die besten Ergebnisse erzielten Multiple Matchkeys und probabilistisches Record-Linkage mit Parametern, die durch einen ECM geschätzt wurden, bei der Verwendung aller möglichen Linkage-Merkmale. Diese klassifizieren die meisten Records richtig und die Ergebnisse führen zu keiner systematischen Verzerrung.

Ist der Geburtsort nicht als Merkmal verfügbar, so erzielen Multiple Matchkeys und Signaturen die besten Ergebnisse. Da der Effekt der Namensänderung bei einer Ehe starke Auswirkungen auf das Linkage hat, können darüber hinaus auch gute Ergebnisse erzielt werden, wenn der Nachname nicht als Merkmal verwendet wird. Dies spricht jedoch nicht dafür, das Merkmal nicht zu erfassen, da ein optimales Linkage ohne Nachnamen nicht mehr möglich ist.

Multiple Matchkeys erzeugen dabei geringfügig bessere Ergebnisse. Die simulierten Datensätze haben jedoch immer einen vollständigen Overlap. Dies muss für die Evaluierung der Ergebnisse in Betracht gezogen werden. Es ist zu erwarten, dass Multiple Matchkeys nur in diesem Fall bessere Ergebnisse als das probabilistische Linkage erzielen.

Neue Record-Linkage Methoden, die mehrere Matchkeys verwenden (Multiple Matchkeys und Signaturen) sind sehr vielversprechend – insbesondere aufgrund ihrer geringen Laufzeiten und guten Linkage-Qualität. Die Ergebnisse dieser Arbeit weisen jedoch zudem darauf hin, dass die Bestimmung der Matchkeys bzw. candiadate Signatures kein triviales Verfahren ist. Es bedarf somit weiterer Forschungsarbeit, um die Bestimmung zu verbessern.

[58] Die Ergebnisse lassen sich durchaus für große administrative Datensätze verallgemeinern.

CLKs, die in Multibit Trees einsortiert werden, weisen für ein verschlüsseltes Verfahren (PPRL) sehr gute Ergebnisse auf. Sie sind jedoch am stärksten von der Größe des Datensatzes abhängig.

Alle Formen der einfachen Matchkeys erwiesen sich als nicht geeignete Verfahren zur Verknüpfung der Daten. Sie haben zum einen die schlechtesten Linkage-Ergebnisse und erzeugen zum anderen den höchsten Bias.

Literatur

Bachteler, T. und R. Schnell (2006). Ein Performanz-Vergleich zwischen der Kölner und der von Reth-Schek Phonetik. In: *Zentrum für quantitative Methoden und Surveyforschung Universität Konstanz*, S. 1–4.

Borgs, C. (2019). Optimal Parameter Choice for Bloom Filter-Based Privacy-Preserving Record Linkage. Universität Duisburg-Essen.

Borst, F., F.-A. Allaert und C. Quantin (2001). The Swiss Solution for Anonymously Chaining Patient Files. In: *Proceedings of the 10th World Congress on Medical Informatics: Held in London, United Kingdom*. Hrsg. von V. L. Patel, R. Rogers und R. Haux. Studies in Health Technology and Informatics 84. Amsterdam: IOS Press.

Bujard, M., C. Diehl, M. Kreyenfeld, C. K. Spieß und Wissenschaftliche Beirat für Familienfragen (2019). *Familien mit Fluchthintergrund: Aktuelle Fakten zu Familienstruktur, Arbeitsmarktbeteiligung und Wohlbefinden*. Bundesministerium für Familie, Senioren, Frauen und Jugend. URL: https://www.bmfsfj.de/resource/blob/140756/d9b5173da1eca339f2507a4c60bcffdd/familien-mit-fluchthintergrund-aktuelle-fakten-data.pdf (zuletzt zugegriffen am 06. 05. 2022).

Bundesinstitut für Berufsbildung (2021). *Datenreport zum Berufsbildungsbericht 2021 Informationen und Analysen zur Entwicklung der beruflichen Bildung*. Bonn: Bundesinstitut für Berufsbildung.

Bundesinstitut für Bevölkerungsforschung (o. J.). *Heiratsziffer in Ausgewählten Altersgruppen Nach Geschlecht in Deutschland (1990-2018)*. URL: https://www.bib.bund.de/Permalink.html?id=10221920 (zuletzt zugegriffen am 13. 04. 2022).

Christen, P. (2008). Febrl – a Freely Available Record Linkage System with a Graphical User Interface. In: *HDKM '08 Proceedings of the Second Australasian Workshop on Health Data and Knowledge Management*. Hrsg. von J. R. Warren, P. Yu, J. Yearwood und J. D. Patrick. Wollongong: ACS, S. 17–25.

Christen, P. (2012). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Data-Centric Systems and Applications. Berlin, New York: Springer.

Christen, P. und A. Pudjijono (2009). Accurate Synthetic Generation of Realistic Personal Information. In: *Advances in Knowledge Discovery and Data Mining*. Hrsg. von T. Theeramunkong, B. Kijsirikul, N. Cercone und T.-B. Ho. Bd. 5476. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, S. 507–514.

Christen, P., T. Ranbaduge und R. Schnell (2020). *Linking Sensitive Data: Methods and Techniques for Practical Privacy-Preserving Information Sharing*. Cham: Springer.

Collins, L., J. L. Schafer und C.-M. Kam (2001). A Comparison of Inclusive and Restrictive Strategies in Modern Missing Data Procedures. In: *Psychological Methods* 6.4, S. 330–351.

De Bruin, J. (2015). Probabilistic Record Linkage with the Fellegi and Sunter Framework: Using Probabilistic Record Linkage to Link Privacy Preserved Police and Hospital Road Accident Records. Delft University of Technology.

Deutsche Post Adress GmbH (2018). *Umzugsstudie 2021 – so Zieht Deutschland um*. URL: http://www.postadress.de/umzugsstudie.pdf (zuletzt zugegriffen am 04. 05. 2022).

Doidge, J. C. und K. L. Harron (2019). Reflections on Modern Methods: Linkage Error Bias. In: *International Journal of Epidemiology* 48.6, S. 2050–2060.

Enamorado, T., B. Fifield und K. Imai (Mai 2019). Using a Probabilistic Model to Assist Merging of Large-Scale Administrative Records. In: *American Political Science Review* 113.2, S. 353–371.

Fellegi, I. P. und A. B. Sunter (1969). A Theory for Record Linkage. In: *Journal of the American Statistical Association* 64.328, S. 1183–1210.

Graham, J. W. (2009). Missing Data Analysis: Making It Work in the Real World. In: *Annual Review of Psychology* 60.1, S. 549–576.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. In: *The Bell System Technical Journal* 29.2, S. 147–160.

Han, J. und M. Kamber (2012). *Data Mining: Concepts and Techniques*. 3rd ed. Burlington, MA: Elsevier.

Hand, D. und P. Christen (2018). A Note on Using the F-Measure for Evaluating Record Linkage Algorithms. In: *Statistics and Computing* 28.3, S. 539–547.

Hannappel, M. und J. Kopp (2020). *Mikrosimulationen: Methodische Grundlagen und ausgewählte Anwendungsfehler*. Wiesbaden: Springer.

Hernandez, M. A. und S. J. Stolfo (1998). Real-World Data Is Dirty: Data Cleansing and the Merge/Purge Problem. In: *Data Mining and Knowledge Discovery* 2, S. 9–37.

Herzog, T. N., F. Scheuren und W. E. Winkler (2007). *Data Quality and Record Linkage Techniques*. New York ; London: Springer.

Huinink, J. und S. Kley (2011). *Migrationsentscheidungen im Lebensverlauf*. Unter Mitarb. von I. F. E. U. A. S. (Universität Bremen. Version 1.0.0. GESIS Data Archive. URL: https:

//search.gesis.org/research_data/ZA5228?doi=10.4232/1.11063 (zuletzt zugegriffen am 29. 04. 2022).

Jaro, M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. In: *Journal of the American Statistical Association* 84.406, S. 414–420.

Karmel, R. (2005). *Data Linkage Protocols Using a Statistical Linkage Key*. Australian Institute of Health and Welfare, Canberra.

Kristensen, T. G., J. Nielsen und C. N. Pedersen (2010). A Tree-Based Method for the Rapid Screening of Chemical Fingerprints. In: *Algorithms for Molecular Biology* 5.1, S. 9–20.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In: *Soviet Physics. Doklady* 10.8, S. 707–710.

McKenzie, P. (2010). *Falsehoods Programmers Believe About Names*. URL: https://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names/ (zuletzt zugegriffen am 13. 04. 2022).

Meng, X.-L. und D. B. Rubin (1993). Maximum Likelihood Estimation via the ECM Algorithm: A General Framework. In: *Biometrika* 80.2, S. 267–278.

Minssen, H. (2006). *Arbeits- und Industriesoziologie: eine Einführung*. Sozialwissenschaftliche Studienbibliothek Bd. 3. Frankfurt am Main: Campus-Verlag.

Münnich, R., R. Schnell, H. Brenzel, H. Diekmann, S. Dräger, J. Emmenegger, P. Höcker, J. Kopp, H. Merkle, K. Neufang, M. Obersneider, J. Reinhold, J. Schaller, S. Schmaus und P. Stein (27. Apr. 2021). A Population Based Regional Dynamic Microsimulation of Germany: The MikroSim Model. In: *methods* data, 23 Pages.

Office for National Statistics (2020). *Splink: MoJ's Open Source Library for Probabilistic Record Linkage at Scale*. URL: https://www.gov.uk/government/publications/joined-up-data-in-government-the-future-of-data-linking-methods/splink-mojs-open-source-library-for-probabilistic-record-linkage-at-scale (zuletzt zugegriffen am 04. 05. 2022).

Peterson, J. L. (1986). A Note on Undetected Typing Errors. In: *Communications of the ACM* 29.7, S. 633–637.

Pilar Angeles, M. del und A. Espino-Gamez (2015). Comparison of methods Hamming Distance, Jaro, and Monge-Elkan. In: *DBKDA 2015 the Seventh International Conference on Advances in Databases, Knowledge, and Data Applications*. Hrsg. von F. Laux, A. Schmidt, M. Del Pilar Angeles und N. Kiyoshi. Rome.

Postel, H. J. (1969). Die Kölner Phonetik: Ein Verfahren Zur Identifizierung von Personennamen Auf Der Grundlage Der Gestaltanalyse. In: *IBM-Nachrichten* 19, S. 925–931.

Ranbaduge, T., P. Christen und R. Schnell (19. Apr. 2021). Large Scale Record Linkage in the Presence of Missing Data. In: arXiv: 2104.09677. URL: http://arxiv.org/abs/2104.09677 (zuletzt zugegriffen am 09. 03. 2022).

Randall, S., A. P. Brown, A. M. Ferrante und J. H. Boyd (2019). Privacy Preserving Linkage Using Multiple Dynamic Match Keys. In: *International Journal of Population Data Science* 4.1.

Reth, H.-P. von und H.-J. Schek (1977). *Eine Zugriffsmethode für die phonetische Ähnlichkeitssuche (Technical Report No. 77.03.002)*. Heidelberg: IBM Scientific Center.

Sariyar, M. und A. Borg (2010). The RecordLinkage Package: Detecting Errors in Data. In: *The R Journal* 2.2, S. 61–67.

Schnell, R. (2016). Record Linkage. In: *The SAGE Handbook of Survey Methodology*. Hrsg. von C. Wolf, D. Joye, T. W. Smith und Y.-c. Fu. Thousand Oaks: SAGE Publications, S. 662–669.

Schnell, R. (2019). *Survey-Interviews: Methoden standardisierter Befragungen*. 2. Auflage. Studienskripten zur Soziologie. Wiesbaden: Springer VS.

Schnell, R. (2022). *Verknüpfung von Bildungsdaten in einem Bildungsregister mittels Record-Linkage auf Basis von Personenmerkmalen*. Bundesministerium für Bildung und Forschung.

Schnell, R., T. Bachteler und J. Reiher (2009). Privacy-Preserving Record Linkage Using Bloom Filters. In: *BMC Medical Informatics and Decision Making* 9.41, S. 1–11.

Schnell, R., T. Bachteler und J. Reiher (2011). A Novel Error-Tolerant Anonymous Linking Code. In: *SSRN Electronic Journal*.

Schnell, R., C. Borgs und V. Wedekind (2020). *Durchführung einer Simulationsstudie als Gutachten im Bereich Registerzensus*. Technical report. Destatis.

Shipsey, R. und J. Plachta (2021). *Linking with Anonymised Data – How Not to Make a Hash of It*. URL: https://www.gov.uk/government/publications/joined-up-data-in-government-the-future-of-data-linking-methods/linking-with-anonymised-data-how-not-to-make-a-hash-of-it (zuletzt zugegriffen am 25. 03. 2022).

Statistisches Bundesamt (2021a). Allgemeinbildende Schulen. In: *Fachserie 11* 1.

Statistisches Bundesamt (2021b). *Daten zu den Mehrlingsgeburten für die Jahre 2016 bis 2020*. URL: https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Bevoelkerung/Geburten/Tabellen/geburten-mehrlinge.html (zuletzt zugegriffen am 13. 04. 2022).

Statistisches Bundesamt (2021c). Studierende an Hochschulen. In: *Fachserie 11* 4.1.

Statistisches Bundesamt (2022a). *Sterbetafel (Periodensterbetafel): Deutschland, Jahre, Geschlecht, Vollendetes Alter*. URL: https://www-genesis.destatis.de/genesis/online?sequenz=tabelleErgebnis&selectionname=12621-0001&sachmerkmal=GES&sachschluessel=GESM#abreadcrumb (zuletzt zugegriffen am 13. 04. 2022).

Statistisches Bundesamt (2022b). *Studienanfängerquote: Deutschland, Jahre, Geschlecht*. URL: https://www-genesis.destatis.de/genesis/online?sequenz=tabelleErgebnis&selectionname=21381-0003&regionalschluessel=#abreadcrumb (zuletzt zugegriffen am 21. 04. 2022).

Talburt, J. R. (2011). *Entity resolution and information quality*. eng. 1st edition. Amsterdam ; Elsevier/Morgan Kaufmann.

Vidanage, A., T. Ranbaduge, P. Christen und S. Randall (2020). Privacy Attack on Multiple Dynamic Match-key Based Privacy-Preserving Record Linkage. In: *International Journal of Population Data Science* 5.1.

Wickham, H. (2014). Tidy Data. In: *Journal of Statistical Software* 59.10.

Winkler, W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. In: *Proceedings of the Section on Survey Research Methods*.

Yancey, W. E. (2005). Evaluating String Comparator Performance for Record Linkage. In: *Research Report Series (Statistics #2005-05)*.

Zhang, Y., K. S. Ng, M. Walker, P. Chou, T. Churchill und P. Christen (2018). Scalable Entity Resolution Using Probabilistic Signatures on Parallel Databases. In: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*, S. 2213–2221.

Anhang A

Programmcode

A.1 Laufzeiten

Methode	Fehlerquote							
	0.1%		0.3%		0.7%		1%	
	$\sum t$	$\bar{t}$	$\sum t$	$\bar{t}$	$\sum t$	$\bar{t}$	$\sum t$	$\bar{t}$
Exakt (Gesamtmenge)	33 m	1 m	32 m	1 m	32 m	1 m	36 m	1 m
Exakt (Restmenge)	1 m	2 s	2 m	3 s	3 m	4 s	4 m	5 s
SLK-581	2 m	17 s	4 m	23 s	5 m	33 s	6 m	43 s
CLK	22 h	25 m	34 h	38 m	78 h	87 m	192 h	214 m
ECM	6 h	18 m	8 h	26 m	12 h	41 m	17 h	57 m
Signatur	73 h	4 h	155 h	9 h	455 h	25 h	845 h	47 h
Multiple Matchkeys	1 h	4 m	2 h	6 m	3 h	9 m	3 h	10 m

Tabelle 11: Summierte und durchschnittliche Laufzeit der einzelnen Record-Linkage Methoden für das Linkage der Schülerregister (zehn simulierte Jahre entsprechen neun Linkage-Vorgängen). Es gilt zu beachten, dass manche Methoden mehr Aufrufe haben als andere (siehe Abschnitt 2.3). Darüber hinaus wurde die Berechnung der CLKs nicht einberechnet. Diese beträgt ca. 10 min für jedes Datensatzpaar. Die Signaturen wurden auf dem langsameren Server berechnet und wurden zudem mit mehreren Prozessaufrufen gleichzeitig ausgeführt (siehe Fußnote 53 S. 68). Die Ergebnisse sind daher nicht vollständig vergleichbar.

Die Simulation wurde im Wesentlichen auf einem Server von Delta-Computer Products mit einer AMD EPYC 7702P CPU (64 Kerne, 128 Threads, max. 2.1 GHz) und 1 Terabyte RAM unter Ubuntu 20.04.4 mit R 4.1.2 und Python 3.8.10 ausgeführt. Die Erstellung eines Folgejahres dauert ca. 30 min. Das Linkage der Signaturen und einiger CLKs wurde aus Zeitgründen auf einem weiteren Server mit vier Intel Xeon CPU E7-4850 (jeweils 10 Kerne, 20 Threads, max. 2 GHz) 128 GB RAM und 256 GB Swap unter Ubuntu 18.04.6 mit R 4.1.2 und Python 3.6.9 ausgeführt.

Darüber hinaus wurden mehrere Aufrufe der CLKs und besonders die Signaturen gleichzeitig ausgeführt. Dies war aus Zeitgründen nötig und möglich, da die Implementation nicht parallelisiert ist.

Durch die beschriebene Verfahrensweise sind die Laufzeiten schlecht vergleichbar. In Tabelle 11 wird die summierte und durchschnittliche Laufzeit für alle Methoden aufgelistet. Die genannten Zahlen sind somit grobe Richtwerte zur Abschätzung der Dauer eines Verfahrens.

IMPRINT

Publisher

German Record-Linkage Center
Regensburger Str. 100
D-90478 Nuremberg

Editor

Rainer Schnell

Template layout

Christine Weidmann

Download

www.record-linkage.de

The German Record Linkage Center is funded by the German Research Foundation (DFG).

DOI: 10.17185/duepublico/76361
URN: urn:nbn:de:hbz:465-20220729-161020-7