

Otto-Friedrich-Universität Bamberg
Lehrstuhl für Statistik und Ökonometrie
Standort: Bamberg
Sommersemester 2022
Vorlesung: Blockseminar Survey-Methodik
Prüfer: Dr. Sara Bleniger

Statistische Geheimhaltung: Cell Key Methode

Joshua Simon
joshua-guenter.simon@stud.uni-bamberg.de
Master Survey Statistik, 4. Fachsemester
Matrikelnummer: 2032411
30. Juni 2022

Inhaltsverzeichnis

1. Einführung	4
2. Etablierte Geheimhaltungsverfahren	5
2.1. Methodische Grundlagen	5
2.2. Statistische Tabellen	5
3. Cell Key Methode	6
3.1. Verfahrensparamter und Überlagerungsmatrix	7
3.2. Methodik und Verfahrensdurchführung	9
3.2.1. Erzeugung der Originalwerte	10
3.2.2. Cell-Key-Bestimmung	10
3.2.3. Lookup-Modul	11
3.3. Besonderheiten der Cell Key Methode	12
3.4. Aufdeckungsrisiko	12
4. Ergebnisse und Auswertungen	12
5. Zusammenfassung und Fazit	13
Literatur	14
A. Python Implementierung	15

Abbildungsverzeichnis

1.	Beispiel für eine Überlagerungsmatrix mit Übergangswahrscheinlichkeiten für paarige Originalwerte und Zielhäufigkeiten	8
2.	Ablaufdiagramm der Cell Key Methode	9

Tabellenverzeichnis

1.	Beispiel für eine Häufigkeitstabelle	6
2.	Beispiel für eine Wertetabelle	6
3.	Mikrodaten mit Record-Keys	10
4.	Aggregierte Daten mit Record-Key-Summen und Cell-Key	11
5.	Überlagerungen und final geheimgehaltene Werte	11
6.	Beispiel zur Nicht-Additivität der CKM	12

Listings

1.	CKM Python Beispiel	15
----	-------------------------------	----

1. Einführung

Die amtliche Statistik sorgt mit einer Vielzahl an Veröffentlichungen für die Bereitstellung von aufbereiteten statistischen Informationen. Damit geht sie dem Ziel nach, Bürgern, Institutionen und anderen gesellschaftlichen Einrichtungen eine Datengrundlage für die Entscheidungsfindung zu bieten. Weiter dient die amtliche Statistik auch der Politik und der Wissenschaft als Datenquelle. Das Sammeln und Erheben dieser Daten stellt in vielen Fällen einen Eingriff auf das Recht der informationellen Selbstbestimmung für Personen und Entitäten dar. Dieses Recht ist das Fundament des modernen Datenschutzes und wird über Artikel 2 des Grundgesetzes abgedeckt. Es steht außer Frage, dass dieses Recht besonders schützenswert ist. Demnach steht auch die amtliche Statistik in der Pflicht dieser Verantwortung nachzukommen. Konkret manifestiert sich das Einhalten dieser Pflicht in dem sog. Statistikgeheimnis. Aus dem Bundesstatistikgesetz lässt sich hierzu der folgende Absatz aufgreifen (§16 Abs. 1 Satz 1 BStatG):

„Einzelangaben über persönliche und sachliche Verhältnisse, die für eine Bundesstatistik gemacht werden, sind von den Amtsträgern und für den öffentlichen Dienst besonders Verpflichteten, die mit der Durchführung von Bundesstatistiken betraut sind, geheim zu halten, soweit durch besondere Rechtsvorschrift nichts anderes bestimmt ist.“

Konkret möchte man mit der statistischen Geheimhaltung einen Schutz für einzelne Person und Entitäten vor der Offenlegung ihrer sensiblen Daten bieten. Dies dient im Weiteren auch der Aufrechterhaltung des Vertrauensverhältnisses zwischen den Befragten und den statistischen Ämtern und erhebenden Einrichtungen. Dies gewährleistet abermals die Zuverlässigkeit der Angaben und der Berichtswilligkeit der Befragten. Die vorausgegangenen Punkte werden in einer Begründung zum BStatG erwähnt [Nickl, 2019]. Ausnahmen von einer Geheimhaltung bestehen nur in Ausnahmefällen, z.B. wenn eine explizite Einwilligung zur Veröffentlichung durch den Befragten vorliegt oder wenn sich die Informationen aus allgemein zugänglichen Quellen von öffentlichen Stellen beziehen. Auch die inner-beördliche Übermittlung, Methodenentwicklung, Planungs- und Forschungszwecke werden über das BStatG geregelt.

Im weiteren Verlauf dieser Arbeit sollen Geheimhaltungsverfahren und Geheimhaltungsregeln präsentiert werden, die im Einzelnen die statistische Geheimhaltung gewährleisten und damit dem Statistikgeheimnis der amtlichen Statistik nachkommen. Besondere Beachtung wird dabei der Cell Key Methode (CKM) geschenkt. Dieses Verfahren bietet ein Ansatz, welcher gegenüber anderen Verfahren gut zu Implementieren und zu Automatisieren ist. Gerade dieser Punkt ist in einem immer

weiter werdenden technologischen Umfeld nicht außer Acht zu lassen.

2. Etablierte Geheimhaltungsverfahren

In diesem Abschnitt sollen zunächst die grundlegenden Begrifflichkeiten für Geheimhaltungsverfahren und Geheimhaltungsregeln beschrieben werden.

2.1. Methodische Grundlagen

Grundsätzlich gibt es zwei sich unterscheidende methodische Ansätze, die bei der Geheimhaltung zum Tragen kommen können. Zum einen existieren *informationsreduzierende Methoden*. In der Gattung dieser Verfahren werden durch Aggregation oder Sperrung kritische Kategorien oder Werte die Aufdeckungsrisiken verhindert. Eine Aggregation meint in diesem Fall das Zusammenfassen zu übergeordneten Positionen, z.B. durch Summation kleinerer Positionen. Bei einer Sperrung ist auch oft von einer Löschung die Rede. Hier werden gezielt einzelne Werte identifiziert und aus der Tabelle entfernt. Als Kontrast stehen *datenverändernde Methoden* gegenüber. Hier werden durch gezielte Veränderungen der Daten - beispielsweise durch Runden oder Zufallsüberlagerungen - kritische Werte verfälscht, was auch für eine erfolgreiche statistische Geheimhaltung sorgen kann.

Weiter differenziert man Geheimhaltungsverfahren auch nach dem Zeitpunkt ihrer Durchführung. Die Daten können bereits vor der Tabellierung mit einer Geheimhaltung versehen werden. Man spricht hier von *pre-tabulare Verfahren*. Diese Verfahren werden als Anonymisierung bezeichnet, da die Daten im Vorfeld so verändert werden, dass keine kritischen Ergebnisse resultieren. Oftmals ist diese Art von Geheimhaltung aber nicht ausreichend, weshalb weitere Verfahren im Anschluss angewandt werden müssen. Man spricht nun von *post-tabulare Verfahren*. Ihre Mechanismen werden auf die fertig tabellierten Daten angewandt.

2.2. Statistische Tabellen

Gegenstand der Geheimhaltung stellen in dieser Arbeit statistische Tabellen dar. Ein Großteil der Veröffentlichungen der amtlichen Statistik sind selbst - oder beinhalten - statistische Tabellen, welche aus den amtlichen Daten abgeleitet werden. Maßgebend für die Anwendung eines Geheimhaltungsverfahrens ist die Art der zu veröffentlichenden Tabelle, die vorliegt. Man unterscheidet im allgemeinen zwischen *Häufigkeitstabellen* und *Wertetabellen*. Erstere stellen Häufigkeiten oder Fallzahlen dar, z.B. die Anzahl von Frauen und Männern innerhalb einer Universität. Werte-

tabellen hingegen stellen Wertesummen wie Umsätze dar. Diese unterschiedlichen Kontexte, in denen die Zahlen dieser Tabellen interpretiert werden können, fordern eine natürliche Unterscheidung innerhalb der Geheimhaltung. Es folgen zwei einfache Beispiel für diese Tabellentypen mit rein fiktiven Ausprägungen.

Studienfach	männlich	weiblich	insgesamt
Bauingenieurwesen	4	3	7
Informatik	9	12	21
Medizin	4	1	5
Survey Statistik	10	10	20
Gesamt	27	26	53

Tabelle 1: Beispiel für eine Häufigkeitstabelle

Brauerei	Mährs Bräu	Schinkerla	Käsmann	Gesamt
Umsatz	600 000	50 000	250 000	900 000

Tabelle 2: Beispiel für eine Wertetabelle

3. Cell Key Methode

Die im vorherigen Kapitel beschriebenen Geheimhaltungsverfahren müssen in der Regel - zumindest bis zu einem gewissen Grad - manuell durchgeführt werden und eine Automatisierung ist eher unflexibel. Mit der *Cell Key Methode (CKM)* wird ein Geheimhaltungsverfahren präsentiert, welches gut zu automatisieren und vergleichsweise einfach zu implementieren ist. Die Cell Key Methode ist auch als *ABS-Verfahren* bekannt. Der Name stammt von der schöpfenden Institution des Verfahrens, dem Australian Bureau of Statistics, ab. Durch die Verwendung von zufallsbasierten Additionen, den sog. Überlagerungen, werden Datenwerte verschleiert. Der Ermittlung einer solchen zufallsbasierten Addition liegt eine einmalig festzulegende Wahrscheinlichkeitsverteilung mit den möglichen Überlagerungswerten zugrunde [Enderle, 2019]. Für die Bestimmung dieser Überlagerungen wird ein deterministischer Mechanismus eingesetzt. Dieser nutzt den original Zellwert und den sog. Cell-Key um aus der Verteilung der Überlagerungswerte eine eindeutige Überlagerung zu ziehen [Enderle, 2019]. Eine mit dem CKM Verfahren geheimehaltene Tabelle veröffentlicht also die Summe aus Originalwerten und Überlagerungen. Die CKM zählt damit zu den datenverändernden Verfahren.

3.1. Verfahrensparamter und Überlagerungsmatrix

An statistische Geheimhaltungsverfahren, insbesondere den datenverändernden Verfahren, werde bestimmte Anforderungen gestellt. Für die Cell Key Methode werden in der amtlichen Statistik gewisse stochastische Eigenschaften gefordert, um die Qualität und Nachvollziehbarkeit der Ergebnisse zu sichern. Zu diesen Eigenschaften zählt einerseits die Unverzerrtheit der Überlagerungen [Enderle, 2019]. Damit meint man, dass der Überlagerungswert, welcher zu den Originalwerten addiert wird, im Mittel gleich Null ist. Es soll also die Erwartungstreue $E(z) = 0$ gelten. Darüber hinaus fodert man ebenfalls eine konstante Streuung der Verteilung der Überlagerungen [Enderle, 2019]. Es soll die Varianz erhalten bleiben, also $Var(z) = s^2$ gelten. Um diese beiden Eigenschaften zu kokretisieren, werden Verfahrensparamter eingeführt. Diese dienen weiter auch dazu das Geheimhaltungsverfahren - und damit die Überlagerungen - an verschiedene Kontexte anzupassen. Zu diesen Verfahrensparamter zählen nach [Höhne, 2019]:

- Eine boolsche Variable, die angibt, ob Originalwerte 1 und 2 geheimgehalten werden sollen.
- Der Anteil p_0 der nicht zu überlagernden Originalwerte.
- Die Maximalüberlagerung d .
- Die Standardabweichung der Überlagerungsbeiträge s .

Gestand des Interesses sind demnach Zufallsfunktionen z , die eine bedingte Wahrscheinlichkeitsverteilung auf die Originalwerte i mit Zielhäufigkeit j darstellen. Man sucht also für jeden Originalwert $i = \{0, 1, 2, \dots, n\}$ eine Wahrscheinlichkeitsverteilung der Form $z = p_i$ mit den Wahrscheinlichkeiten für die Übergänge v_i hin zu den Zielhäufigkeiten j [Enderle, 2019]. Diese Wahrscheinlichkeiten bilden ein nicht-lineares Gleichungssystem, in welchem die zuvor genannten stochastischen Eigenschaften als Nebenbedinungen eingehen. Diese lassen sich nun als

$$E(z) = \sum_{i=-d}^d p_i v_i = 0 \quad (1)$$

$$Var(z) = \sum_{i=-d}^d p_i v_i^2 = s^2 \quad (2)$$

$$\sum_{i=-d}^d p_i = 1 \quad (3)$$

schreiben [Höhne, 2019]. Dabei wird in der letzten Zeile (3) noch gefordert, dass die Summe der Übergangswahrscheinlichkeiten für einen Originalwert gleich 1 ist. Die Lösung dieses Problems lässt sich in Matrixform notieren. Man spricht hier von der sog. Überlagerungsmatrix mit Zeilen i und Spalten j , welche zur Bestimmung der Überlagerungsbeiträge j zu den Originalwerten i verwendet werden kann. Für weitere mathematische Details und Lösungsansätze sei an dieser Stelle auf [Giessing, 2016] verwiesen. In [Höhne, 2019] wird die Lösung eines solchen Gleichungssystems für die Verfahrensparameter $p_0 = 0,5 = 50\%$, $d = 4$ und $s = 2,25$ gezeigt. Diese Überlagerungsmatrix ist in Abbildung 1 in Form einer Heatmap zu sehen. Der *Python* Code zum Erstellen dieser Grafik ist in Anhang A zu finden.

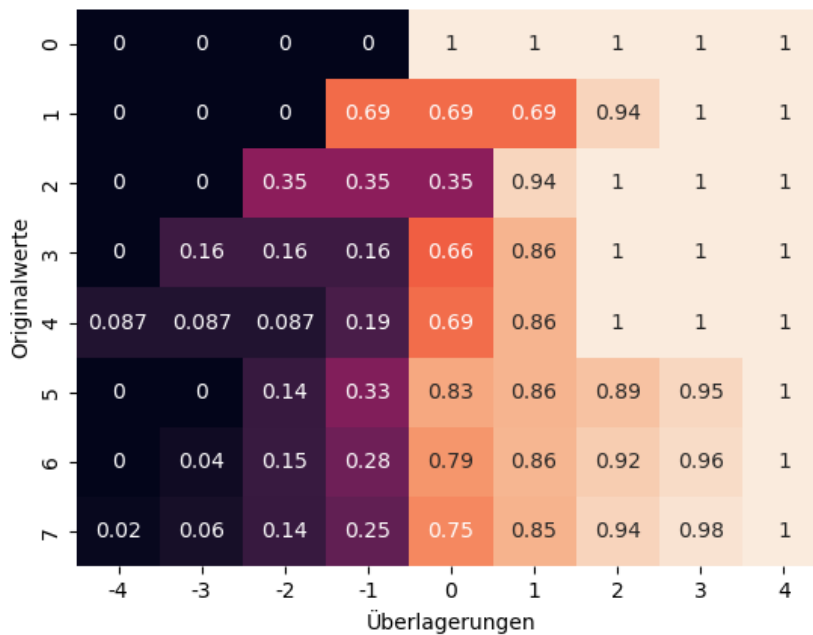


Abbildung 1: Beispiel für eine Überlagerungsmatrix mit Übergangswahrscheinlichkeiten für paarige Originalwerte und Zielhäufigkeiten

Es ist hierbei anzumerken, dass direkt eine symmetrische Verteilung für die Überlagerungsbeiträge gewählt wurde. Auf der x -Achse sind die anzuwendenden Überlagerungswerte von $-d$ bis d zu sehen. Auf der y -Achse sind die Originalwerte abgetragen. In der jeweiligen Zelle sind die einzelnen Übergangswahrscheinlichkeiten abzulesen. Für Originalwerte größer 7 ist die Zeile für Originalwerte gleich 7 maßgebend. Die Überlagerungsmatrix bildet den Kern der CKM. Ihre Anwendung wird im nächsten Abschnitt beschrieben.

3.2. Methodik und Verfahrensdurchführung

Die wichtigsten Bestandteile des Verfahrens werden in [Enderle, 2019] dargestellt. Ähnlich wie in [Wipke, 2018] beschrieben, lässt sich nun ein Algorithmus formulieren. Ausgangspunkt sind die in einer Statistik erhobenen Mikrodaten bzw. Mikrodatensätze. Das sind die plausibilisierten Einzeldatensätze.

1. Erzeugung der Originalwerte mit einem Auswertungstool
2. Cell-Key-Bestimmung aus Zufallszahlen innerhalb des Auswertungs-Tools
3. Lookup-Modul
 - a) Auslesen der Überlagerungswerte aus der Überlagerungsmatrix
 - b) Addieren der Überlagerungswerte und Originalwerte

Die folgende Abbildung 2 visualisiert den schematischen Ablauf des zuvor beschriebenen Algorithmus. Im Wesentlichen werden zwei technische System benötigt, um das Verfahren zu realisieren. Zum einen wird ein Auswertungstool benötigt, welches die gespeicherten Mikrodaten in Tabellenform bringt. Mit Hilfe des Lookup-Moduls werden dann die beiden Eingangsgrößen - Originalwerte und Cell-Keys - verwendet, um die Überlagerungswerte aus der Überlagerungsmatrix zu bestimmen. Diese Überlagerungswerte werden letztlich auf die Originalwerte addiert und stellen damit die finalen Werte für die Veröffentlichung dar.

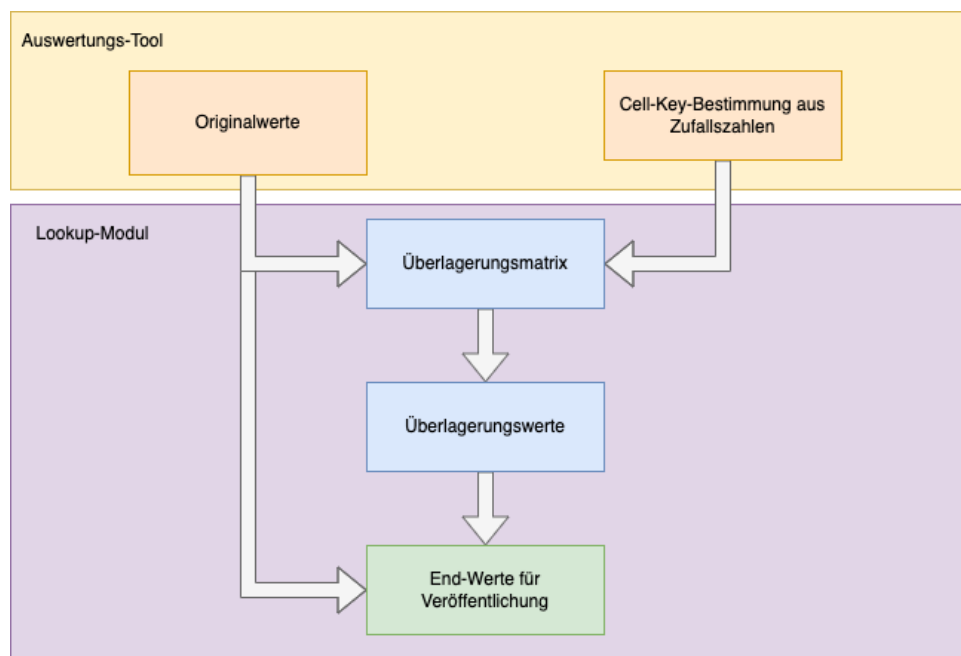


Abbildung 2: Ablaufdiagramm der Cell Key Methode

Die Einzelschritte des Verfahrens sollen nun im Detail beleuchtet werden.

3.2.1. Erzeugung der Originalwerte

Dieser Schritt ist spezifisch für die jeweilige Zieltabelle, die veröffentlicht werden soll. Im Allgemeinen werden Filterungen anhand bestimmter Merkmale vorgenommen und dann Summen der Fallzahlen gebildet. Für diese Operation - also die Tabellierung - führt man die Bezeichnung f ein.

3.2.2. Cell-Key-Bestimmung

Für die Bestimmung des Cell-Keys wird jedem Mikrodatsatz zunächst eine gleichverteilte Zufallszahl r , der sog. *Record-Key*, mit $r \sim \mathcal{U}(0,1)$ zugeordnet. Mit diesen Record-Keys wird dieselbe Auswertungstabelle wie mit den Originalwerten gebildet. Man führt also dieselbe Operation f auf den Daten durch, wie in Abschnitt 3.1.1. Es ergeben sich also Summen von Record-Keys. Von diesen Record-Key Summen werden nur die Nachkommastellen betrachtet. Dieser Wert definiert den *Cell-Key* [Enderle, 2019]. Um dieses Vorgehen zu verdeutlichen soll das folgende Beispiel aus der Hochschulstatistik betrachtet werden. Tabelle 3 zeigt die Mikrodaten mit Record-Keys, wie sie in einer Datenbank gespeichert sein könnten.

ID	Universität	Geschlecht	Record-Key
1	Würzburg	m	0,611853
2	Eichstätt	w	0,139494
3	München	w	0,292145
4	München	m	0,366362
5	Würzburg	m	0,456070
6	Würzburg	m	0,785176
7	Bamberg	m	0,199674
8	München	w	0,514234
9	München	m	0,592415
10	München	m	0,046450

Tabelle 3: Mikrodaten mit Record-Keys

Tabelle 4 stellt die Daten nach Durchführung der Tabellierung f dar. Die Daten wurden hier nach der Universität und nach dem Geschlecht zusammengefasst. Die entsprechenden Fallzahlen, Record-Key-Summen und Cell-Keys werden abgebildet. Die Werte in Zeile 5 von Tabelle 4 ergeben sich durch Zählen der Zeilen in Tabelle 3, in denen $Universität = Würzburg \wedge Geschlecht = m$. Die Record-Key-Summe in dieser Zeile ergibt sich nach $0,611853 + 0,456070 + 0,785176 = 1,853099$. Der daraus abgeleitete Cell-Key beträgt damit 0,853099.

ID	Universität	Geschlecht	Fallzahl	Record-Key-Summe	Cell-Key
1	Bamberg	m	1	0,199674	0,199674
2	Einchstätt	w	1	0,139494	0,139494
3	München	m	3	1,005227	0,005227
4	München	w	2	0,806379	0,806379
5	Würzburg	m	3	1,853099	0,853099

Tabelle 4: Aggregierte Daten mit Record-Key-Summen und Cell-Key

3.2.3. Lookup-Modul

Im Anschluss an die Bestimmung der Originalwerte und der dazugehörigen Cell-Keys gilt es nun die Überlagerungen zu berechnen. Für die Bestimmung eines Überlagerungswertes dient das Paar (*Originalwert*, *CellKey*) als Input. Damit meint man den Wert einer einzelnen Tabellenzeile und den über dieselbe Operation f berechneten Cell-Key. Das Lookup Modul stellt die Funktionalität bereit, anhand dieses Wertepaares den zugehörigen Überlagerungswert aus der Überlagerungsmatrix abzulesen. Um dies weiter zu illustrieren wird die Überlagerungsmatrix aus Abbildung 1 herangezogen. Betrachtet man Zeile 5 aus Tabelle 4, so liegt das Paar (*Originalwert* = 3, *CellKey* = 0,853099) vor. Um nun auch den dazu passenden Überlagerungswert zu bestimmen, wählt man die Zeile i der Überlagerungsmatrix, die dem *Originalwert* - hier also 3 - entspricht. Anschließend bestimmt man mit Hilfe des *CellKey* die Spalte j , für welche der Wert in der Matrix erstmalig kleiner als *CellKey* ist. In diesem Fall führt dies zu $j = 1$, da $0,853099 < 0,86$. Der Wert $j = 1$ ist damit der zu addierende Überlagerungswert. Daraus ergibt sich ein finaler Wert von $3 + 1 = 4$. Wendet man dieses Vorgehen vollständig auf das Beispiel aus Tabelle 4 an, so ergeben sich die finalen und damit geheimgehaltenen Werte aus Tabelle 5.

ID	Universität	Geschl.	Fallzahl	Überlagerung	Finaler Wert
1	Bamberg	m	1	-1	0
2	Einchstätt	w	1	-1	0
3	München	m	3	-3	0
4	München	w	2	1	3
5	Würzburg	m	3	1	4

Tabelle 5: Überlagerungen und final geheimgehaltene Werte

In Anhang A ist eine *Python* Realisierung dieses Verfahrens zu finden.

3.3. Besonderheiten der Cell Key Methode

Ein Besonderheit, die sich aus der Verwendung der Cell Key Methode ergibt, ist die Nicht-Additivität des Verfahrens. Dadurch, dass Rand- und Zwischensummen nicht erst nach der Geheimhaltung, sondern ebenfalls während der Tabellierung gebildet werden, unterliegen auch diese Werte dem Geheimhaltungsmechanismus. Um dies zu veranschaulichen soll abschließend ein weiteres Fallbeispiel aus der Hochschulstatistik betrachtet werden. Tabelle 6 zeigt für eine Universität jeweils eine Insgesamt-Position i sowie die Unterteilung nach dem Geschlecht m und w . Es sind sowohl die Originalwerte als auch die final überlagerten Werte abgebildet. Auch die Positionen für die Zwischensummen i wurden mit der Tabellierungs-Operation f auf Basis der Originalwerte und der Record-Keys gebildet. Damit entsteht also ein eigener Cell-Key und Überlagerungswert für diese Zahlen. Für das Beispiel der Universität Bamberg aus Tabelle 6 sieht man schnell, dass für die Originalwerte die gewohnte Additivität vorhanden ist, denn $166 = 75 + 91$. Führt man diese Betrachtung allerdings auf den überlagerten Werten durch, so erhält man eine falsche Aussage, denn $168 \neq 75 + 91$. Diese Phänomän bezeichnet man als *Nicht-Additivität* einer statistischen Tabelle.

ID	Universität	Geschl.	orig. Fallzahl	überl. Fallzahl
1	Bamberg	i	166	168
2	Bamberg	m	75	75
3	Bamberg	w	91	91
4	Einchstätt	i	46	48
5	Einchstätt	m	17	20
6	Einchstätt	w	32	29

Tabelle 6: Beispiel zur Nicht-Additivität der CKM

3.4. Aufdeckungsrisiko

4. Ergebnisse und Auswertungen

In diesem Kapitel sollen Ergebnisse über die Datenqualität zusammengetragen werden. Die Analyse solcher (Meta-)Daten ist zentral für das Data Engineering, da hiermit das implementierte System validiert und eine mögliche Verbesserung der Datenqualität durch verschiedene Aufbereitungsprozesse gemessen werden kann. Im Anschluss folgt ein kurzer Überblick über die erhobenen Daten selbst.

5. Zusammenfassung und Fazit

Damit das Bayerische Landesamt für Statistik seiner gesetzlich vorgeschriebenen Pflichten nachkommen kann, sind die unterschiedlichsten Technologien und Verfahren notwendig. Neben dem Erheben der Daten bei den Meldern, gehören auch die Plausibilisierung und die Aufbereitung der Daten zu seinen Aufgaben. Um eine so große Datenmenge effizient zu verarbeiten, wird auf moderne Lösungen aus der Informationstechnologie zurückgegriffen. Hierzu zählen verschiedene Skriptsprachen, Datenbanken und Datawarehouses. Um die Qualität der Daten zu wahren, sind neben rein technischer Kontrollen auch sehr fachliche Zusammenhänge zu prüfen und ggf. zu korrigieren. Dies erfordert ein tiefes inhaltliches Verständnis der Daten und ihrer Merkmale. Eine gewisse Interpretation und Analyse sind also auch im Data Engineering notwendig, um die Datenqualität und den Datenfluss zu erhalten, sowie die Daten zu publizieren und damit die Pflicht der amtlichen Statistik zu erfüllen.

Literatur

- Enderle, Tobias und Meike Vollmar: Geheimhaltung in der Hochschulstatistik. *WISTA* | 6, Statistisches Bundesamt (Destatis), Wiesbaden 2019.
- Giessing, Sarah: Computational Issues in the Design of Transition Probabilities and Disclosure Risk Estimation for Additive Noise. *LNCS*, vol. 9867, Springer International Publishing, 2016.
- Höhne, Jörg und Julia Höninger: Die Cell-Key-Methode – ein Geheimhaltungsverfahren. *Statistische Monatshefte Niedersachsen* 1, 2019.
- Nickl, Andreas: Datenschutz, Geheimhaltung, Anonymisierung. *Einführungsfortbildung* Bayerisches Landesamt für Statistik, Fürth, 2019.
- Rothe, Patrick: Statistische Geheimhaltung – Der Schutz vertraulicher Daten in der amtlichen Statistik - Teil 1: Rechtliche und methodische Grundlagen *Bayern in Zahlen* 5, Bayerisches Landesamt für Statistik, München, 2015.
- Rothe, Patrick: Statistische Geheimhaltung – Der Schutz vertraulicher Daten in der amtlichen Statistik - Teil 2: Herausforderungen und aktuelle Entwicklungen. *Bayern in Zahlen* 8, Bayerisches Landesamt für Statistik, München, 2015.
- Wipke, Mirko: Geheimhaltung im Data Warehouse - Prototypische Implementierung von automatisierter Geheimhaltung im Data Warehouse für die amtliche Hochschulstatistik in Bayern. *Bayern in Zahlen* 12, Bayerisches Landesamt für Statistik, Fürth, 2018.

A. Python Implementierung

Nachfolgend ist die vollständige *Python* Implementierung des CKM Verfahrens für ein Testbeispiel aus der Hochschulstatistik abgebildet.

```
1 # ckm.py
2 # This python scripts implements the cell key method used
3 # for a toy example.
4 # -----
5 # Joshua Simon, 11.05.2022
6
7
8 import math
9 import numpy as np
10 import pandas as pd
11 import seaborn as sns
12 import matplotlib.pyplot as plt
13
14
15 # Values for the overlay matrix and vector are taken from
16 # "Die Cell-Key-Methode ein Geheimhaltungsverfahren"
17 # by Jörg Höhne und Julia Höninger.
18 OVERLAY_MATRIX = np.matrix([
19     [0, 0, 0, 0, 1, 1, 1, 1, 1],
20     [0, 0, 0, 0.6875, 0.6875, 0.6875, 0.9375, 1, 1],
21     [0, 0, 0.3533, 0.3533, 0.3533, 0.9440, 0.9970, 0.9990, 1],
22     [0, 0.1620, 0.1620, 0.1620, 0.6620, 0.8560, 0.9970, 0.9990, 1],
23     [0.0870, 0.0870, 0.0870, 0.1920, 0.6920, 0.8590, 0.9970,
24      0.9990, 1],
25     [0, 0, 0.1450, 0.3270, 0.8270, 0.8590, 0.8930, 0.9490, 1],
26     [0, 0.0400, 0.1500, 0.2850, 0.7850, 0.8600, 0.9200, 0.9600, 1],
27     [0.0200, 0.0600, 0.1450, 0.2500, 0.7500, 0.8550, 0.9400,
28      0.9800, 1]
29 ])
30
31 CHANGE_VECTOR = [-4, -3, -2, -1, 0, 1, 2, 3, 4]
32
33 def matrix_plot(matrix, vector):
34     sns.heatmap(matrix, annot=True, xticklabels=vector, cbar=False)
35     plt.xlabel("Überlagerungen")
36     plt.ylabel("Originalwerte")
37     plt.show()
38
39 def generate_data(n, seed):
```

```

40     """
41     Generates some random data from sample attributes.
42     Each row gets a random uniformly distributed record key
43     between 0 and 1.
44     """
45     np.random.seed(seed)
46     universities = ["Bamberg", "Wuerzburg", "Muenchen", "Eichstaett
47 ]
48     sex = ["m", "w"]
49
50     uni_data = np.random.choice(universities, size=n, replace=True,
51                                p=[0.15, 0.3, 0.5, 0.05])
52     sex_data = np.random.choice(sex, size=n, replace=True, p=[0.5,
53 0.5])
54     record_key_data = np.random.uniform(low=0.0, high=1.0, size=n)
55
56     return pd.DataFrame(
57         list(zip(uni_data, sex_data, record_key_data)),
58         columns=['university', 'sex', 'record_key']
59     )
60
61 def tabulate_data(data, rollout=False):
62     """
63     Generates the grouped frequency table with summed record keys.
64     If the rollout option is true, all of the grouped sums are
65     calculated as well.
66     """
67     grouped_data = data.groupby(["university", "sex"]).agg(["count"
68 , "sum"])
69     grouped_data.columns = ["count", "record_key_sum"]
70     grouped_data.reset_index(inplace=True)
71
72     if rollout:
73         rollout_data = data.loc[:, data.columns != "sex"].groupby([
74 "university"]).agg(["count", "sum"])
75         rollout_data.columns = ["count", "record_key_sum"]
76         rollout_data.reset_index(inplace=True)
77         rollout_data["sex"] = "i"
78         rollout_data = rollout_data.iloc[:, [0,3,1,2]]
79
80         sum_col = pd.DataFrame({
81             "university": ["sum"],
82             "sex": ["i"],
83             "count": [grouped_data["count"].sum()],

```



```

80         "record_key_sum": [grouped_data["record_key_sum"].sum()
81     ]
82     })
83     grouped_data = grouped_data.append([rollout_data, sum_col],
84         ignore_index=True)
85     grouped_data = grouped_data.sort_values(by=["university", "
86     sex"])
87
88     return grouped_data
89
90 def get_cell_key(value: float) -> float:
91     """
92     Returns the decimal part of a floating point number.
93     """
94     return value - int(value)
95
96 def get_len_of_int(value: int) -> int:
97     """
98     Returns the length (= number of digits) of an positive integer.
99     """
100    return int(math.log10(value)) + 1
101
102
103 def get_overlay_matrix_value(matrix, vector, values,
104     record_key_sums, seed, p0=1) -> list:
105     """
106     Returns the overlay value given by the overlay matrix and
107     vector
108     for a value-record_key_sum-pair.
109     The overlay value is determined by the value itself and the
110     floating
111     point digits of the record_key_sum value. The value is used as
112     a
113     row-index to find the row in the overlay matrix. If the value
114     and
115     therefore the row-index is out of range, the last row of the
116     matrix
117     is used. In the selected row, the index of the column, where
118     the
119     record_key_sum is bigger than the column value is then used as
120     in index

```

```

113     for the overlay vector. The selected value of this vector is
114     the
115     overlay value which is to add to the original table value. The
116     probability
117     p0 determines the chance, that the overlay value is actually
118     used.
119     """
120     np.random.seed(seed)
121     overlay_col = []
122     num_rows, _ = matrix.shape
123
124     for value, record_key_sum in zip(values, record_key_sums):
125         if value == 0:
126             overlay_col.append(value)
127             continue
128         elif value < num_rows:
129             cell_keys = matrix[value, :]
130         else:
131             cell_keys = matrix[num_rows - 1, :]
132
133         for index, key in enumerate(cell_keys.tolist()[0]):
134             if key > get_cell_key(record_key_sum):
135                 overlay_value = vector[index]
136                 break
137         else:
138             overlay_value = vector[-1]
139
140         if p0 is not None:
141             overlay_value = np.random.choice([overlay_value, 0],
142 size=1, p=[1 - p0, p0])[0]
143             overlay_col.append(overlay_value)
144
145     return overlay_col
146
147 def apply_ckm(data, matrix, vector, value_col_names,
148 record_key_names, seed, p) -> pd.DataFrame:
149     """
150     Applies the Cell Key Method to the named columns of a data set.
151     Therefore the overlay value is calculated and added to the
152     named
153     columns.
154     Returns a DataFrame with the overlaid data.
155     """
156     output_data = data.copy()

```

```

152     for col_name, record_key_name, p0 in zip(value_col_names,
record_key_names, p):
153         output_data[col_name] = data[col_name] +
get_overlay_matrix_value(matrix, vector, data[col_name], data[
record_key_name], seed, p0)
154     return output_data
155
156
157 if __name__ == "__main__":
158     data = generate_data(1001, 42)
159     table_data = tabulate_data(data)
160     overlayed_data = apply_ckm(table_data, OVERLAY_MATRIX,
CHANGE_VECTOR, ["count"], ["record_key_sum"], seed=42, p=[0])
161
162     #print(data)
163     print(table_data)
164     print(overlayed_data)
165
166     matrix_plot(OVERLAY_MATRIX, CHANGE_VECTOR)
167
168     #print(get_cell_key(2.456))
169     #print(get_len_of_int(1000))
170     #print(get_overlay_matrix_value(OVERLAY_MATRIX, CHANGE_VECTOR,
251, 120.846))
171
172     print("Done.")

```

Listing 1: CKM Python Beispiel

Ich erkläre hiermit, dass ich die Seminararbeit mit dem Titel *Statistische Geheimhaltung: Cell Key Methode* im *Sommersemester 2022* selbständig angefertigt, keine anderen Hilfsmittel als die im Literaturverzeichnis genannten benutzt und alle aus den Quellen und der Literatur wörtlich oder sinngemäßübernommenen Stellen als solche gekennzeichnet habe.

Bamberg, den 30. Juni 2022

Unterschrift