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
26. Juni 2022

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

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 mordernen Datenschutzes und wird über Artikel 2 des Grundgesetztes 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 Plficht in dem sog. Statistikgeheimnis. Aus dem Bundestatistikgesetzt 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 sensitven 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 Begrüdung zum BStatG erwähnt [Nickl, 2019]. Ausnahmen von einer Geheimhaltung bestehen nur in Ausnahmefällen, z.B. wenn eine explizite Einwillung 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 technolgischen 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 unterscheidenede methodische Ansätze, die bei der Geheimhaltung zum Tragen kommen können. Zum einen existieren *informationsreduzierende Methoden*. In der Gattung dieser Verfahren werden durch Aggregation oder Sperrrung kritische Kategorien oder Werte die Aufdeckungsrisiken verhindert. Eine Aggregation meint in diesem Fall das Zusammenfassen zu übergeordneteten 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 ferig 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 *Häufigkeitstabllen* und *Wertetabellen*. Erstere stellen Häufigkeiten oder Fallzahlen dar, z.B. die Anzahl von Frauen und Männern innerhalb einer Universität. Wertetabellen 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 beschrieben Geheimhaltungsverfahren müssen in der Regel - zumindest bis zu einem gewissen Grad - manuell durchgeführt werden und eine Automatisierung ist eher unfelxibel. 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 Institutuion 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 Mechaniusmus 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 geheimgehaltene Tabelle veröffentlicht also die Summe aus Originalwerten und Überlagerungen. Die CKM zählt damit zu den datenveränderndeden 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 Eigenschfaten 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 $i = \{0, 1, 2, \ldots, n\}$ 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 nichtlineares Gleichungssystem, in welchem die zuvor genannten stochastischen Eigenschaften als Nebenbedinungen eingehen. Diese lassen sich nun als

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

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

$$\sum_{i=-d}^{d} p_i = 1 \tag{3}$$

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 [] verwiesen.

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 1 visualisiert den schehmatischen Ablauf des zuvor beschrieben 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.

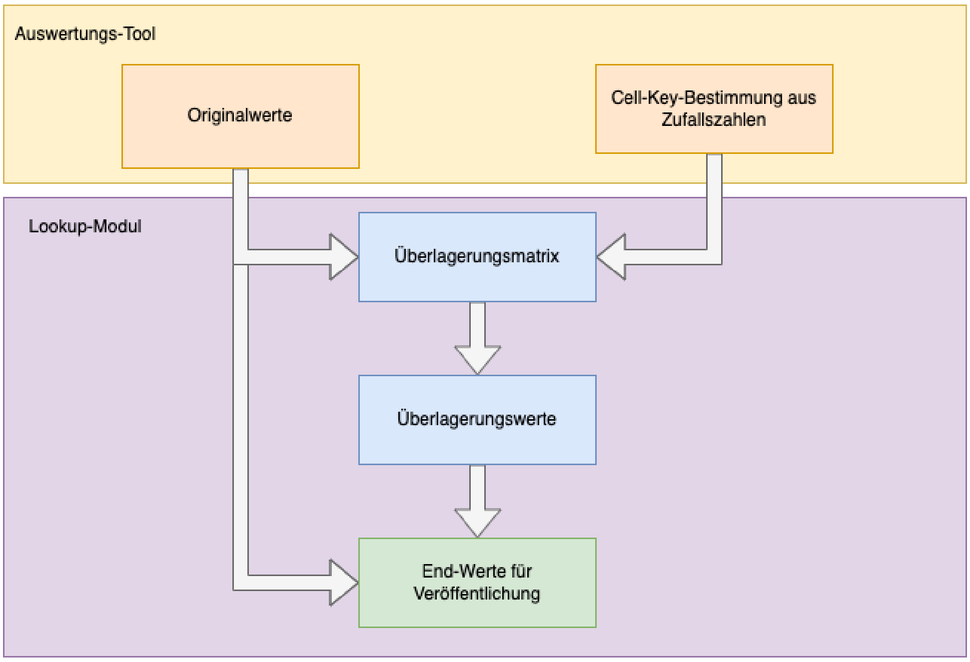

Abbildung 1: Ablaufdiagramm der Cell Key Methode

Die Einzelschritte des Verfahrens sollen nun im Detail beleuchtet werden.

3.2.1. Erzeugung der Originalwerte

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

3.2.2. Cell-Key-Bestimmung

Für die Bestimmung des Cell-Keys wird jedem Mikrodatensatz zunächst eine gleichverteile 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 diesseble 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.
Tabelle 4 stellt die Daten nach Durchführung der Tabellierung f dar. Die Daten wurden hier nach der Universität und nach dem Geschlecht zusammengefasst. Die

ID	Universität	Geschlecht	Record-Key
1	Würzburg	m	0,611853
2	Einchstätt	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

entsprechenden Fallzahlen, Record-Key-Summen und Cell-Keys werden abgebildet.

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

Die Werte in Zeile 5 von Tablle 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.

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 Tabellenzelle und den über dieselbe Operation f berechneteten Cell-Key. Das Lookup Modul stellt die Funktionalität bereit, anhand dieses Wertepaares den zugehörigen Überlagerungswert aus der Überlagerungsmatrix abzulesen.

3.3. Besonderheiten der Cell Key Methode

Test

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 vorgeschrieben 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 zurückgriffen. 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.

Höhne, Jörg und Julia Höninger: Die Cell-Key-Methode ein Geheimhaltungsverfahren. *Statistische Monatshefte Niedersachsen* 1, 2019.

Nickl, Andreas: Datenschutz, Geheimhaltung, Anonymisierung. *Einfuührungsfortbildung* 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.

```python
# ckm.py
# This python scripts implements the cell key method used
# for a toy example.
# ---------------------------------------------------------
# Joshua Simon, 11.05.2022


import math
import numpy as np
import pandas as pd


# Values for the overlay matrix and vector are taken from
# "Die Cell-Key-Methode  ein Geheimhaltungsverfahren"
# by Jörg Höhne und Julia Höninger.
OVERLAY_MATRIX = np.matrix([
    [0, 0, 0, 0, 1, 1, 1, 1, 1],
    [0, 0, 0, 0.6875, 0.6875, 0.6875, 0.9375, 1, 1],
    [0, 0, 0.3533, 0.3533, 0.3533, 0.9440, 0.9970, 0.9990, 1],
    [0, 0.1620, 0.1620, 0.1620, 0.6620, 0.8560, 0.9970, 0.9990, 1],
    [0.0870, 0.0870, 0.0870, 0.1920, 0.6920, 0.8590, 0.9970, 0.9990, 1],
    [0, 0, 0.1450, 0.3270, 0.8270, 0.8590, 0.8930, 0.9490, 1],
    [0, 0.0400, 0.1500, 0.2850, 0.7850, 0.8600, 0.9200, 0.9600, 1],
    [0.0200, 0.0600, 0.1450, 0.2500, 0.7500, 0.8550, 0.9400, 0.9800, 1]
])

CHANGE_VECTOR = [-4, -3, -2, -1, 0, 1, 2, 3, 4]


def generate_data(n, seed):
    """
    Generates some random data from sample attributes.
    Each row gets a random uniformly distributed record key
    between 0 and 1.
    """
    np.random.seed(seed)
    universities = ["Bamberg", "Wuerzburg", "Muenchen", "Eichstaett"]
    sex = ["m", "w"]
```

```python

    uni_data = np.random.choice(universities, size=n, replace=True, p=[0.15, 0.3, 0.5, 0.05])
    sex_data = np.random.choice(sex, size=n, replace=True, p=[0.5, 0.5])
    record_key_data = np.random.uniform(low=0.0, high=1.0, size=n)

    return pd.DataFrame(
        list(zip(uni_data, sex_data, record_key_data)),
        columns =['university', 'sex', 'record_key']
    )


def tabulate_data(data, rollout=False):
    """
    Generates the grouped frequency table with summed record keys.
    If the rollout option is true, all of the grouped sums are
    calculated as well.
    """
    grouped_data = data.groupby(["university", "sex"]).agg(["count", "sum"])
    grouped_data.columns = ["count", "record_key_sum"]
    grouped_data.reset_index(inplace=True)

    if rollout:
        rollout_data = data.loc[:, data.columns != "sex"].groupby(["university"]).agg(["count", "sum"])
        rollout_data.columns = ["count", "record_key_sum"]
        rollout_data.reset_index(inplace=True)
        rollout_data["sex"] = "i"
        rollout_data = rollout_data.iloc[:, [0,3,1,2]]

        sum_col = pd.DataFrame({
            "university": ["sum"],
            "sex": ["i"],
            "count": [grouped_data["count"].sum()],
            "record_key_sum": [grouped_data["record_key_sum"].sum()]
        })

        grouped_data = grouped_data.append([rollout_data, sum_col], ignore_index=True)
        grouped_data = grouped_data.sort_values(by=["university", "sex"])

```

```
    return grouped_data


def get_cell_key(value: float) -> float:
    """
    Returns the decimal part of a floating point number.
    """
    return value - int(value)


def get_len_of_int(value: int) -> int:
    """
    Returns the length (= number of digits) of an positive integer.
    """
    return int(math.log10(value)) + 1


def get_overlay_matrix_value(matrix, vector, values, record_key_sums, seed, p0=1) -> list:
    """
    Returns the overlay value given by the overlay matrix and vector
    for a value-record_key_sum-pair.
    The overlay value is determined by the value ifself and the floating
    point digits of the record_key_sum value. The value is used as a
    row-index to find the row in the overlay matrix. If the value and
    therefore the row-index is out of range, the last row of the matrix
    is used. In the selected row, the index of the column, where the
    record_key_sum is bigger than the column value is then used as in index
    for the overlay vector. The selected value of this vector is the
    overlay value which is to add to the original table value. The probability
    p0 determines the chance, that the overlay value is actually used.
    """
    np.random.seed(seed)
    overlay_col = []
    num_rows, _ = matrix.shape
```

```python

    for value, record_key_sum in zip(values, record_key_sums):
        if value == 0:
            overlay_col.append(value)
            continue
        elif value < num_rows:
            cell_keys = matrix[value, :]
        else:
            cell_keys = matrix[num_rows - 1, :]

        for index, key in enumerate(cell_keys.tolist()[0]):
            if key > get_cell_key(record_key_sum):
                overlay_value = vector[index]
                break
        else:
            overlay_value = vector[-1]

        if p0 is not None:
            overlay_value = np.random.choice([overlay_value, 0], size=1, p=[1 - p0, p0])[0]
        overlay_col.append(overlay_value)

    return overlay_col


def apply_ckm(data, matrix, vector, value_col_names, record_key_names, seed, p) -> pd.DataFrame:
    """
    Applys the Cell Key Method to the named columns of a data set.
    Therefore the overlay value is calculated and added to the named
    columns.
    Returns a DataFrame with the overlayed data.
    """
    output_data = data.copy()
    for col_name, record_key_name, p0 in zip(value_col_names, record_key_names, p):
        output_data[col_name] = data[col_name] + get_overlay_matrix_value(matrix, vector, data[col_name], data[record_key_name], seed, p0)
    return output_data


if __name__ == "__main__":
    data = generate_data(1001, 42)
```

```
    table_data = tabulate_data(data)
    overlayed_data = apply_ckm(table_data, OVERLAY_MATRIX,
   CHANGE_VECTOR, ["count"], ["record_key_sum"], seed=42, p=[0])

    #print(data)
    print(table_data)
    print(overlayed_data)

    #print(get_cell_key(2.456))
    #print(get_len_of_int(1000))
    #print(get_overlay_matrix_value(OVERLAY_MATRIX, CHANGE_VECTOR,
   251, 120.846))

    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 26. Juni 2022
Unterschrift