TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

Master Thesis

Disclosure Risk Estimation for Survey Microdata

Ausgeführt am Institut für
Statistik und Wahrscheinlichkeitstheorie
der Technischen Universität Wien

unter der Anleitung von
Privatdoz. Dipl.-Ing. Dr.techn. Matthias Templ
Wiedner Hauptstrasse 8-10/107
1040 Wien

durch
Marius Totter
An der Unteren Alten Donau 91/171
1220 Wien

Wien, 8. Dezember 2014

Erklärung

Hiermit erkäre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die aus anderen Quellen entnommenen Stellen als solche gekennzeichnet habe.

Wien, den 24.11.2014

Marius Totter

Danksagung

An erster Stelle möchte ich mich bei meinem Betreuer Privatdozent Dipl.-Ing. Dr.techn. Matthias Templ bedanken. Ich wurde bestmöglich unterstützt und betreut. Zusätzlich konnte ich ein für mich neues und interessantes Forschungsgebiet kennenlernen, sowie mein Wissen in der Programmiersprache R erweitern.

Ein sehr großes Dankeschön möchte ich auch meiner Familie aussprechen, insbesondere Riki und Hannes Totter, die immer an mich geglaubt und mir das Studium erst ermöglicht haben.

ABSTRACT. The estimation of the re-identification risk of individuals in survey microdata is in main focus of this master thesis. For released confidential data it is mandatory that individuals have very low risk of identification, otherwise laws on data privacy are violated. Many different anonymisation methods exist and their aim is both, to reduce the disclosure risk and to minimize information loss at the same time. The disclosure risk itself is described mathematically and the corresponding methods are implemented in software. One approach for estimating disclosure risk measures of categorical variables is based on log-linear models, which are used for modeling frequency counts. Knowing the truth by using synthetic population data and sampling from it, four log-linear models are tested on four different sampling designs and three different categorical variable scenarios in order to evaluate the performance of the methods. Within a simulation study the influence of different sampling designs on the disclosure risk methods is under consideration.

Contents

1 Introduction

A microdata file is defined as a data set consisting of observations on individual units. A disclosure occurs when a person or organisation can learn something that they did not know already about an organisation or person via released data [Hundepool et al., 2010]. In today's world, information is available from a lot of sources and it is in doubt that there is non-existent or very low disclosure risk for data sets to release.

Outline:

In Sections 1.1 and 1.2 a general overview of protecting microdata and categorical variables is given. The focus is on the anonymisation of categorical key variables using methods like recoding, data supression or post-randomisation. Section 1.3 gives a brief insight about sampling techniques, which are applied in Section 4. Simple random sampling, proportional stratified sampling, equal stratified sampling and oversampling are considered. The problem of missing values is discussed and an imputation method based on a variation of the Gower distance is introduced. Section 2 includes the estimation of population frequency counts, whereby Section 2.1 gives general remarks on frequency counts. Section 2.2 describes the concept of k-anonymity, which is also a protecting method of categorical data. As discussed by Willenborg and de Waal [2001] the simplest approach to estimate population frequencies is pointed out in Section 2.3. The standard log-linear model is introduced as dicussed by Agresti [2002] and also two adapted models (Clogg-Eliason [Clogg and Eliason, 1987] and pseudo maximum likelihood method) are discussed, as discussed by Skinner and Vallet [2010]. Additional the weighted log-linear model is introduced. The focus is on the performance of this four models in different scenarios. Section 3 deals with risk estimation methods as considered by Templ et al. [2014a] and Shlomo and Skinner [2008]. For the numerical study two disclosure risk measures are for interest:

1. number of sample uniques that are population unique.
2. number of correct matches for sample uniques.

These two measures are described in Section 3.1.

The programming language R [R Core Team, 2014] is used for all examples and mainly for Section 4. Many R packages are used like `sdcMicro`, `simFrame`, `simPopulation`, `MASS`, `VIM`, `ggplot2` and `reshape2`. The most important package for this work is `sdcMicro` [Templ et al., 2014a], whereby this work complements the package with a function that estimates the above described risk measures using log-linear models. `sdcMicro` includes all methods of the popular software μ-Argus plus several new methods and improvements on data handling, computational speed and user-friendlyness. Section 4 describes the empirical results of a simulation study. An European Union Statistics on Income and Living Conditions sample data set is used to simulate a whole population. Four different sampling designs are used to draw samples from the population. Knowing the population the real disclosure risk can be calculated. Figure 6 shows the structure of Section 4. In Section 4.2 and 4.2.4 the results of the empirical study are reported. Some concluding remarks and directions for future research are given in Section 5.

1.1 General approach for the anonymisation of microdata

Sensitive data are collected in a lot of different fields. The disclosure problem relates to the possibility of identifiying records in released data sets. The aim of anonymisation methods is both, re-identification should be roughly impossible and the data utility of the released data set should be still high. Therefore this is an optimization problem, which depends on many factors, e.g. national laws or importance of deception. The R package `sdcMicro` [Templ et al., 2014b] offers a variaty of anonymisation methods.

Considering a data set $\mathbf{U}$ with a variety of variables $\mathbf{d_j}$ with $j \in \{1, 2, ..., l\}$, it is possible to classify every variable into one of at least three disjunct groups (see Figure 1) [Templ et al., 2014a].

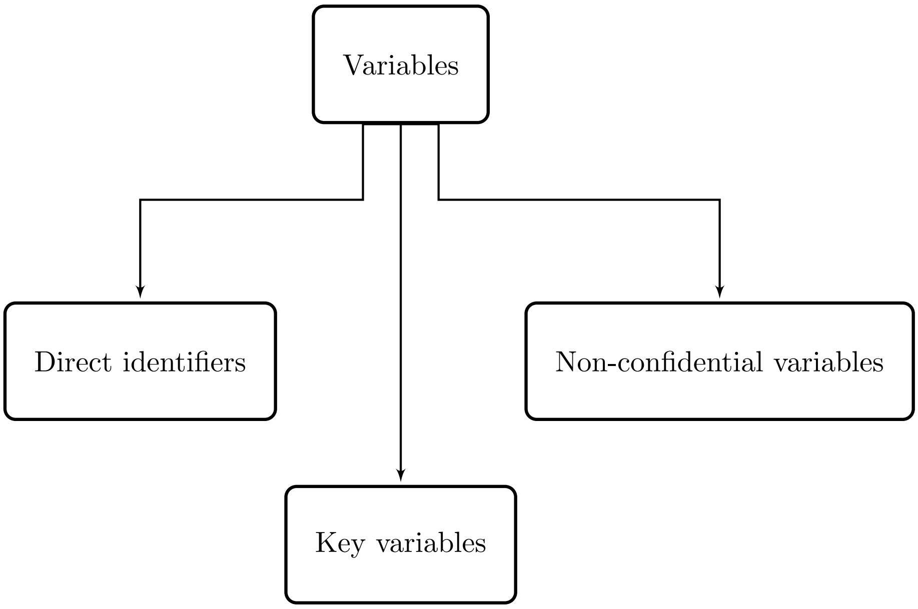


Figure 1: Three disjunct groups of variables.

Definition 1.1 (Direct identifiers)
Direct identifiers are variables that surely identify statistical units.

Definition 1.2 (Key variables)
Key variables are a set of variables that - if considered together - can be used to identify some individual units.

Key variables are often termed as implicit identifiers or quasi identifiers. In this study only methods for categorical key variables are discussed. Which means that a subset $\mathbf{Z}$ from the data set $\mathbf{U}$ is considered, with $\mathbf{d_j} \in \mathbf{Z}$ and $\mathbf{d_j}$ is the j-th categorical variable.

Definition 1.3 (Non-confidential variables)
Non-confidential variables are finally all variables that are not classified in Definitions 1.1 and 1.2.

Example 1.1 (Direct identifiers)
Direct identifiers are, for example, persons, addresses, social insurances, DNA, finger prints, account numbers, names of companies and value added tax identification numbers.

Example 1.2 (Key variables)
It might be possible to identify some individuals by using following combinations of variables:

1. *Gender, citizenship and occupation.*
2. *Establishment, revenue class and number of employees.*

These are two examples for key variables see Definition 1.2.

Remark (Sensitive variables)
Another group of variables is defined for specific protection methods, called sensitive variables, e.g. the income of a person or the health status of a person.

1.2 Protection of categorical variables

In this section some methods for protecting categorical variables are discussed. These methods are generally applied when the estimated re-identification risk (see Section 3) is too high.

Definition 1.4 (Categorical variable)
A categorical variable is a variable which can take only a finite number of values or characteristics.

Definition 1.5 (Categorical key variables)
Categorical key variables are both, categorical and key variables (see Definitions 1.2 and 1.4).

Definition 1.6 (Keys)
A key is a given combination of categories of categorical key variables. All possible combinations are defined as keys.

Example 1.3 (Keys)
Gender and occupation are the categorical key variables with the characteristics male and female for gender as well as Aut, EU and Other for occupation. Then a key is hereby assigned with e.g. male and Aut. There exist 6 possible keys.

1. *female, Aut*
2. *female, EU*
3. *female, Other*
4. *male, Aut*
5. *male, EU*
6. *male, Other*

Three protecting methods are mentioned below, which gives a short overview about masking methods. The application of protection methods yields a decrease of data utility. One goal is to release a safe microdata set and the other goal is to release a data set with high data utility. This leads to an optimization problem where data anonymization specialists have to make some decisions. This considerations should be mentioned, but they are not part of this study.

1.2.1 Recoding

The categories of selected key variables are assigned to broader categories. Global recoding leads to less keys and population uniques. The `sdcMicro` package [Templ et al., 2014b] contains the function `globalRecode()` to apply global recoding.

Example 1.4
The variable age with one year breaks is recoded into 10 *intervals/age groups.*

```
R > age <- sample(1:99, size = 25, replace = TRUE)
R > summary(factor(age))

14 15 23 28 32 33 39 42 52 58 59 63 70 71 80 82 84 86 94 96 97 99
 1  1  1  1  1  1  2  1  1  1  2  1  1  1  1  1  2  1  1  1  1  1
```

```
R > agerec1 <- cut(age, breaks = 10) #or
R > agerec2 <- globalRecode(age, breaks = 10,
+                             labels = paste("Int",1:10,sep=""))
R > summary(agerec2)

 Int1  Int2  Int3  Int4  Int5  Int6  Int7  Int8  Int9 Int10
    2     2     4     1     1     4     2     2     3     4
```

Remark

Recoding can also be applied to continous key variables, which means to discretize the continous variable. This concept is used in Section 4, where the continous variable netIncome (personal net income) is recoded into specific intervals.

1.2.2 Data suppression

The idea is to suppress certain values in one or more categorical variables by replacing them by missing values. In practice data suppression is often used in combination with global recoding. For more details see Willenborg and de Waal [2001]. To apply data suppression the function `localSupp()` (univariate) or `localSuppression()` (multivariate), available in `sdcMicro`, can be used.

Example 1.5

In this example the simulated data set `eusilcS` *(see Section 4 for detailed data describtion) is used to create an* `sdcMicro` *object. Values of the categorical key variable* `pb220a` *are suppressed in the second assingment.* `pb220a` *describes the persons's citizenship with levels AT, EU and Other.*

```
R >   sdcObj <- createSdcObj(eusilcS,
+                 keyVars=c("db040","hsize","rb090", "pb220a"), w= "rb050")
R >   sdcObj <- localSupp(sdcObj, keyVar="pb220a")
```

More advanced features based on the concept of k-anonymity (see Section 2.2) to supress a minimum amount of values is available in function `localSupression()`. See `?localSuppression` in R for more details.

1.2.3 Post randomisation

The Post RAndomisation Method (PRAM) is a probabilistic, perturbative method for disclosure control of categorical variables. As described in de Wolf et al. [1998] this method changes the scores on some categorical variables for certain records according to a prescribed probability mechanism. In the `sdcMicro` [Templ et al., 2014b] package the function `pram()` is implemented to apply post randomisation. This function randomly changes the values of variables on selected records according to an invariant probability transition matrix [Gross et al., 2004].

Example 1.6

Again the `eusilcS` *data set is used to create a* `sdcMicro` *object. Values of the categorical key variables* `rb090` *(person's gender) and* `pb220a` *(citizenship) are randomly changed.*

```
R >   res_pram <- pram(eusilcS, variables = c("rb090","pb220a"))
R >   print(res_pram)
```

```
Number of changed observations:
- - - - - - - - - - -
rb090 != rb090_pram : 520 (4.43%)
pb220a != pb220a_pram : 732 (6.24%)
```

Further functionality is available, see `?pram` in R.

1.3 Sampling techniques

In this Section all considered sampling methods of Section 4 are briefly described. For further information of this techniques, see Cochran [1977] and Lemeshow and Levy [2008].

1.3.1 Simple random sampling

Simple Random Sampling (SRS) without replacement is a method of selecting n units out of a population with N units such that every distinct sample has an equal chance of being drawn $\frac{N!}{n!(N-n)!}$. For sampling without replacement, a particular element can appear only once in a given sample. The probability of any unit being selected is equal to $\frac{n}{N} = \pi_i$, which concludes that the inclusion probabilities are equal for every unit.

Remark (SRS)

5 units are drawn from a data set of 100 records. So there are $\frac{100!}{5!(100-5)!} = 75287520$ *possible samples with equal selection probability. Using R, the possible combinations can be computed with the function* `choose(100,5)` *and the units can be drawn with the function* `sample(x=dataset, size=5,replace=FALSE)`*.*

Example 1.7 (SRS in R)

The function `srs()` *from the package* `simFrame` *[Alfons et al., 2010] is used to draw a sample with size 10 of the* `eusilcS` *data set, which is shown in Table 1.*

```
R > set.seed(23)
R > srs_R <- srs(length(eusilcS[,1]), 10, replace = FALSE)
R > print(xtable(eusilcS[srs_R, c("db040","hsize","rb090", "pb220a")],
+ caption = "A simple random sample of the data set eusilcS.", label="tab:srs11"))
```

	db040	hsize	rb090	pb220a
10510	Tyrol	3	female	Other
8792	Upper Austria	5	male	AT
2615	Vienna	1	male	AT
8246	Upper Austria	4	male	AT
4715	Carinthia	3	male	AT
6671	Styria	5	female	AT
3499	Vienna	3	female	Other
2728	Vienna	1	female	Other
9638	Salzburg	4	male	Other
601	Lower Austria	1	male	AT

Table 1: A simple random sample of the data set eusilcS.

1.3.2 Stratified random sampling

In stratified sampling the population of N units is divided into L disjoint subpopulations of $N_1, N_2, ..., N_L$ units [Cochran, 1977], with $N_1 + N_2 + ... + N_L = N$. The subpopulations are called strata. The sample drawings of each stratum are made independently and if a simple random sample is taken in each stratum the procedure is called stratified random sampling. This sampling method has many advantages over SRS described in Cochran [1977] and Lemeshow and Levy [2008]. Table 2 shows the notations.

Variable	Description
N_j	total number of units in stratum $j \in 1, ..., L$
n_j	number of units in sample stratum $j \in 1, ..., L$
$w_j = \frac{N}{N_j}$	stratum weight
$\pi_j = \frac{N_j}{N}$	inclusion probability
$f_j = \frac{n_j}{N_j}$	sampling fraction in the stratum

Table 2: Stratified random sampling notations

Definition 1.7 (Proportional stratified sampling)
In proportional stratified sampling the amount of drawn records is proportional to the strata size, i.e. the inclusion probability of stratum j is given by $\pi_j = \frac{N_j}{N}$, where N_j is the number of units in stratum j and N is the population size.

Example 1.8 (Proportional stratified sampling)

This example shows the proportional stratified sampling method from Section 4. Variable `tabr` *shows how many households (grouping variable "db030") should be drawn from each federal state (design variable - specifying variables to be used for stratified sampling "db040"). In this demonstration 1000 households are randomly drawn. For further information see* `?SampleControl` *in R.*

```
R >   (tabr <- round(tab <- (table(eusilcS$db040)/length(eusilcS$db040))*1000))

   Burgenland      Carinthia Lower Austria       Salzburg         Styria
           37             79           177             66            159
        Tyrol  Upper Austria        Vienna     Vorarlberg
           95            181           153             53

R >   #db030: household ID; db040: federal state (Austria)
R >   sc <- SampleControl(design = "db040", grouping = "db030",
+                           size = c(tabr), k = 1)
```

Definition 1.8 (Stratified sampling with equal size of each strata)

In equal stratified sampling the amount of drawn records from each stratum are equal, i.e. the inclusion probability of stratum j is given by $\pi_j = \frac{n}{N}$, where n is the number of drawn units in each stratum and N is the population size.

Example 1.9 (Stratified sampling with equal size of each strata)

The following code shows the equal stratified sampling method from Section 4. In this example 110 households are randomly drawn from each federal state.

```
R >   draw <- rep(110, times = 9)
R >   (names(draw) <- levels(eusilcS$db040))

[1] "Burgenland"     "Carinthia"      "Lower Austria" "Salzburg"
[5] "Styria"         "Tyrol"          "Upper Austria" "Vienna"
[9] "Vorarlberg"

R >   sc <- SampleControl(design = "db040", grouping = "db030",
+                           size = draw, k = 1)
```

Definition 1.9 (Unequal probability sampling)

The inclusion probability of individuals in the sampling frame depends on covariates, e.g. the household size.

Example 1.10 (Unequal probability sampling)

In this example the R function `midzuno()` *is used to draw a sample. Households with four or more persons are preferred. For further information see Alfons et al. [2010] and Midzuno [1952]. The midzuno method is a sampling technique for unequal probability sampling without replacement and fixed sample size. Especially, households of size 3 and more are oversampled in this example while small households are under-represented.*

```
R >    (n <- nrow(eusilcS))

[1] 11725

R >    prob <- inclusionProb(eusilcP$hsize, n)
R >    summary(factor(prob))

0.0636432719969603  0.127286543993921  0.190929815990881  0.254573087987841
              8602              14128              12429              13180
 0.318216359984802  0.381859631981762  0.445502903978722  0.509146175975683
              6745               2094                840                528
 0.572789447972643
               108

R >    mdraw <- midzuno(prob)
R >    sample_o <- eusilcP[mdraw,]
R >    summary(factor(sample_o$hsize))

   1    2    3    4    5    6    7    8    9
 524 1760 2413 3386 2137  812  388  250   55

R >    #in comparison
R >    summary(factor(eusilcS$hsize))

   1    2    3    4    5    6    7    8    9
1313 2770 2391 2752 1560  588  245   88   18
```

1.4 Missing values

Virtually all sample surveys include missing values. These can cause a significant effect on the conclusions that can be drawn from the data. To avoid measurement errors in Section 4, a population is considered which is simulated from imputed sample survey data. This implies that all samples drawn from this population don't have missing values. The assumption that there are no missing values makes it easier for an intruder, because there are no additional uncertainties. There are a lot of techniques to deal with missing data. In Section 4 the function `kNN()` of the R package `VIM` is used for imputation of missing values of the survey data, which is used to simulate a population (see Section 4.1.2).

1.4.1 Gower distance

The function `kNN()` is based on a variation of the Gower distance for numerical, categorical, ordered and semi-continous variables. The Gower distance is a very general distance measure that allows to measure the distance between objects of different types (categorical and continous). In order to handle different types of variables, the Gower's dissimilarity coefficient is used [Gower, 1971]. The extension of the Gower's dissimilarity coefficient by Kaufman and Rousseeuw [2005] is described in the following formula. Let $\mathbf{U}$ be a data set then the following distances are calculated:

$$d(i,j) = \sum_k (\delta_{ijk} \cdot d_{ijk}) / \sum_k \delta_{ijk} \quad .$$

Where d_{ijk} represents the distance between the i-th and j-th unit of the k-th variable, which depends on the nature of the variable. If the variable is logical or nominal the columns are considered as binary variables, for such cases $d_{ijk} = 0$ if $u_{ij} = u_{jk}$, otherwise $d_{ijk} = 1$. if the variables are continous the columns are considered as interval-scaled variables and $d_{ijk} = \frac{|u_{ik} - u_{jk}|}{r_k}$, whereby r_k is the range of the k-th variable. The weight δ_{ijk} is determined as follows:

1. $\delta_{ijk} = 0$, if $u_{ij} = \text{NA}$ or if $u_{jk} = \text{NA}$;

2. $\delta_{ijk} = 1$, in all other cases.

2 Frequency counts

2.1 General remarks on frequency counts

Consider a finite population $\mathbf{U}$ of size N. Every record $\mathbf{x} \in \mathbf{U}$ consists of observed values (e.g. name, year of birth, address, gender, citizenship, occupation, income, weight,...). After deleting direct identifiers and defining q key variables, the population frequency counts can be computed for this combinations. The key variables $\mathbf{Z}_1, ..., \mathbf{Z}_q$ have to be categorical, with $C_1, ..., C_q$ characteristics respectively, i.e. $C_j = |\mathbf{Z}_j|$ is the amount of categories from one key variable.

Definition 2.1 (Contingency table)
A contingency table is a type of table that displays the frequency distribution of the categorical variables.

Remark
The elements of a contingency table are denoted as cells. Every cell shows the frequency of one key see Definition 1.6.

Remark
In R the functions `table()` and `tableWt()` are used to compute contingency tables. The second function also takes sample weights into account.

Definition 2.2 (Cross tabulation)
Cross tabulation is a statistical process that summarizes categorical data to create a contingency table.

All combinations of categories in the key variables can be calculated by cross tabulation of these variables. Each combination of values defines a cell in the table. The maximum number of all possible cells is given by $\prod_{i=1}^{q} C_i = C$.
Let $\mathbf{X}$ be the table of all combinations, which is for simplicity labeled as $1, 2, ..., C$. The different categories C of $\mathbf{X}$ divide the population into C subpopulations $\mathbf{U}_j \subseteq \mathbf{U}$ with $j \in \{1, ..., C\}$.

Remark (Key)
A key is one combination of categorical key variables.

Example 2.1
There are two categorical key variables $\mathbf{Z}_1$ (gender) and $\mathbf{Z}_2$ (eye-color) given, with $C_1 = |\mathbf{Z}_1| = 2$ ("man", "woman") and $C_2 = |\mathbf{Z}_2| = 3$ ("blue", "brown", "green") characteristics. Then there exist 6 keys, e.g. ("man", "blue") or ("woman", "green").

Remark

Subpopulation $\mathbf{U}_j \subseteq \mathbf{U}$ *contains all records belonging to the j-th key, with* $j \in \{1, 2, ..., C\}$*. E.g. there are exactly five records in a subpopulation* $\mathbf{U}_j$ *with the key: woman, student, blue. This key yields subpopulation* $\mathbf{U}_{woman,student,blue} \subseteq \mathbf{U}$ *with* $|\mathbf{U}_{woman,student,blue}| = 5$*.*

Definition 2.3 (Frequency counts)

The population frequency counts F_j *with* $j \in \{1, ..., C\}$ *are the numbers of records belonging to subpopulation* $\mathbf{U}_j$*, i.e.* $F_j = |\mathbf{U}_j|$*.*

Remark

The `sdcMicro` *package provides the function* `freqCalc()` *or* `measure_risk()` *which can be used to compute the (sample) frequency counts.*

Consider a random sample $\mathbf{S} \subseteq \mathbf{U}$ of size $n \leq N$ drawn from a finite population $\mathbf{U}$ of size N. Let π_j with $j \in \{1, 2, ..., N\}$ be the inclusion probabilities, which is the probability that a record $\mathbf{x}_j \in \mathbf{U}$ is chosen in the sample. The sample frequency counts are analogously defined as the population frequency counts F_j and denoted by f_j.

Definition 2.4 (Cell size indices)

T_j *is the number of cells of size j, i.e.*

$$T_j = \sum_{i=1}^{C} \mathbb{1}(F_i = j), \; j = 0, 1, ..., N \quad , \tag{1}$$

The sample counterpart t_j *is given by*

$$t_j = \sum_{i=1}^{C} \mathbb{1}(f_i = j), \; j = 0, 1, ..., n \quad , \tag{2}$$

where $\mathbb{1}_{\mathbf{A}}$ denotes the characteristic function of a subset $\mathbf{A}$ of a set $\mathbf{X}$, with $\mathbb{1}_{\mathbf{A}} : \mathbf{X} \to \{0, 1\}$ and

$$\mathbb{1}_{\mathbf{A}}(\mathbf{x}) := \begin{cases} 1 & \text{if } \mathbf{x} \in \mathbf{A} \\ 0 & \text{if } \mathbf{x} \notin \mathbf{A} \end{cases} .$$

The above definitions of T_j and t_j with $j \in 1, 2, ...C$ determines cell size indices of the population and sample. It is clear that there is a relation between T_j and F_i as well as for t_j and f_i.

Example 2.2 (Cell size indices)

The following R code shows the frequency counts calculation with three categorical key variables (federal state, household size and citizenship) of data set `eusilcS` *with the function* `freqCalc()`*.*

There are 160 possible keys with this three categorical key variables and three unique combinations. Function `head()` *shows the first 10 keys of the* `eusilcS` *data set with its corresponding frequency counts. In Figure 2 the cell size indices are visualised from the* `eusilcS` *data set related to three categorical key variables. There are many cells with less frequency counts and only a few with more than 200 observations (see Figure 2).*

```
R > counts <- freqCalc(eusilcS, c("db040","hsize", "pb220a"))$fk
R > x <- cbind(eusilcS, counts)
R > u_c <- aggregate(counts ~ db040 + hsize + pb220a, x, mean)
R > nrow(u_c)
```

```
[1] 160
```

```
R > sum(u_c$counts==1)
```

```
[1] 3
```

```
R > head(u_c[,c("db040","hsize", "pb220a", "counts")],10)
```

```
           db040 hsize pb220a counts
1     Burgenland     1     AT     41
2      Carinthia     1     AT     93
3  Lower Austria     1     AT    241
4       Salzburg     1     AT     80
5        Styria     1     AT    192
6          Tyrol     1     AT     84
7  Upper Austria     1     AT    188
8         Vienna     1     AT    281
9     Vorarlberg     1     AT     50
10    Burgenland     2     AT    117
```

Remark

Relation between T_j and F_i:

$$\sum_{j=1}^{N} jT_j = N = \sum_{i=1}^{C} F_i$$

$$\sum_{j=1}^{n} jt_j = n = \sum_{i=1}^{C} f_i \quad .$$

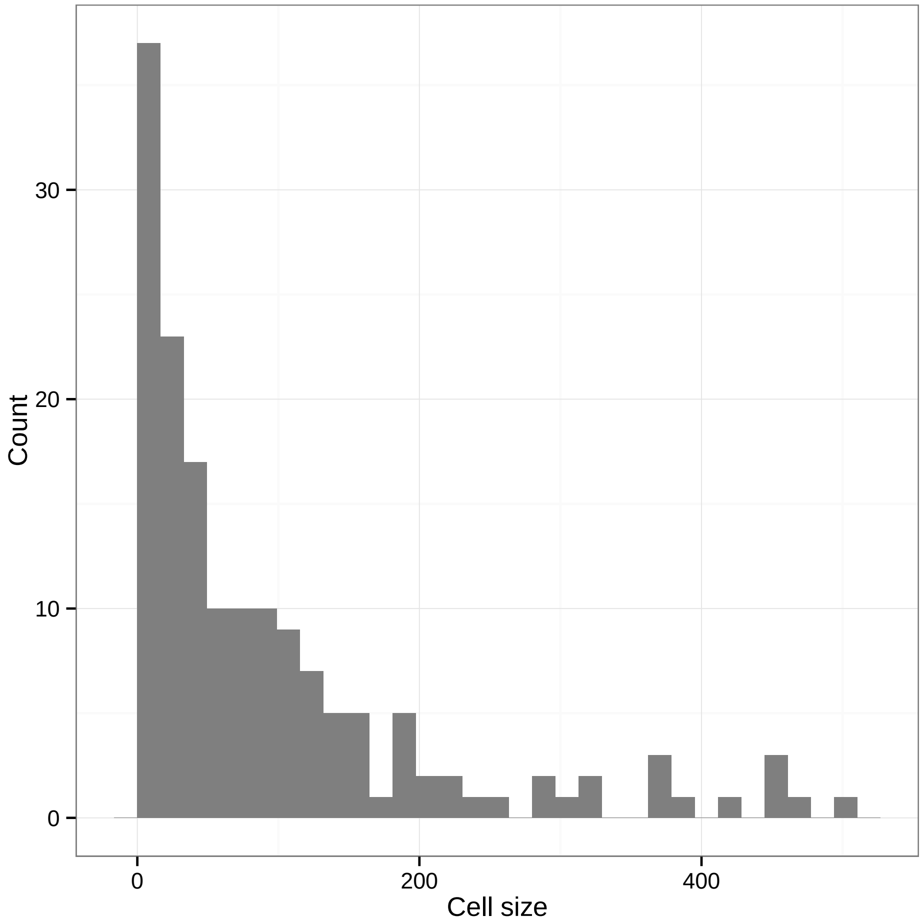


Figure 2: Cell size indices of Example 2.2.

Proof 2.1

of aboves equations, see Definition 2.3.

$$
\begin{aligned}
& \bigcup_{i=1}^{C} \mathbf{U}_i = \mathbf{U} \textit{ and } \mathbf{U}_i \cap \mathbf{U}_j = \emptyset, \ \forall i \neq j \\
\iff & |\bigcup_{i=1}^{C} \mathbf{U}_i| = |\mathbf{U}| \\
\iff & \bigcup_{i=1}^{C} |\mathbf{U}_i| = |\mathbf{U}| \\
\iff & \sum_{i=1}^{C} F_i = N
\end{aligned}
$$

□

Remark

There exists also a relation between the number of combinations and the cell size indices T_i and t_i:

$$
\sum_{j=0}^{N} T_j = \sum_{j=0}^{n} t_j = C \quad .
$$

Proof 2.2

of aboves equation, see also Definition 2.4.

$$
\begin{aligned}
\sum_{j=0}^{N} T_j &= \sum_{j=0}^{N} \sum_{i=1}^{C} \mathbb{1}(F_i = j) \\
&= \sum_{i=1}^{C} \sum_{j=0}^{N} \mathbb{1}(F_i = j) \overset{(1)}{=} \sum_{i=1}^{C} 1 = C \\
\sum_{j=0}^{n} t_j &= \sum_{j=0}^{n} \sum_{i=1}^{C} \mathbb{1}(f_i = j) \\
&= \sum_{i=1}^{C} \sum_{j=0}^{n} \mathbb{1}(f_i = j) \overset{(1)}{=} \sum_{i=1}^{C} 1 = C
\end{aligned}
$$

(1) *because* $0 \leq F_i \leq N$ *and* $0 \leq f_i \leq N, \ \forall i \in 1, ..., C.$ □

Example 2.3

A very simple data set of 14 records is used to explain this section. `Table 3` *shows the whole data. First the direct identifiers are deleted. In this demonstration only the variable name is a direct* *identifier.* `Gender` *and* `Occupation` *are defined as categorical key variables.* `Figure 3` *shows the factor level counts of the variable* `Gender` *on the left and* `Occupation` *on the right-hand side. Function* `table()` *is used to get a contingency table of the counts at each combination*

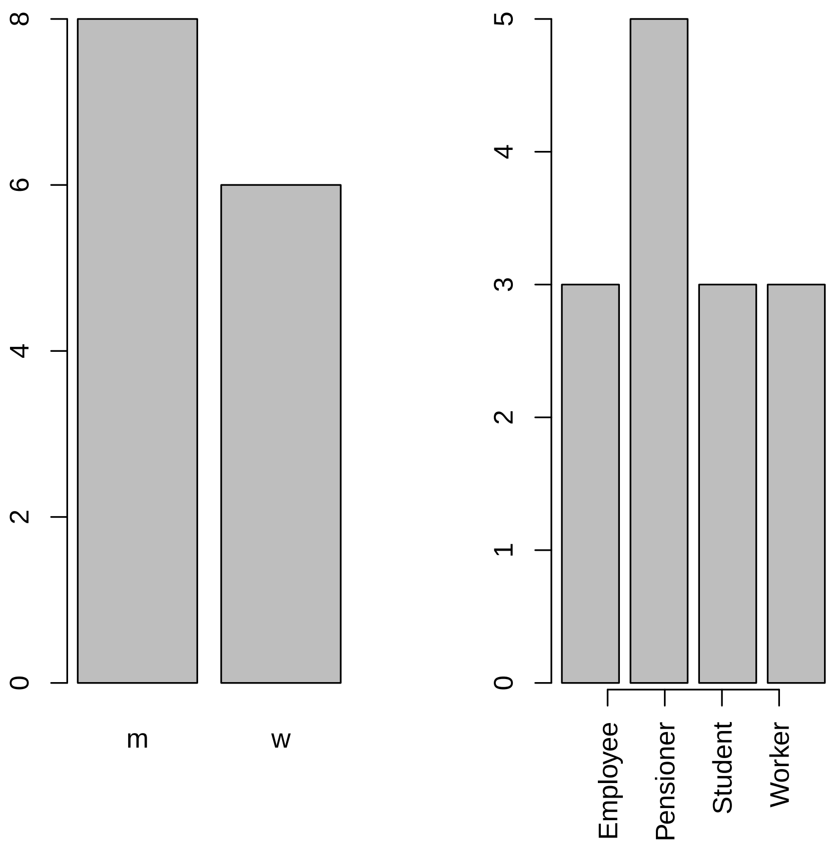


Figure 3: Barplots of variables Gender and Occupation of the example data set.

of factor levels from the variables `Gender` *and* `Occupation`. `tableWt()` *from the R package* `simPopulation` *[Alfons et al., 2011] computes the contingency table taking into account sample weights, which are given in column* `Weight` *of* `Table 3`.

```
R > table(daten[,c("Gender","Occupation")])

       Occupation
Gender Employee Pensioner Student Worker
     m        2         3       0      3
     w        1         2       3      0
```

```
R > tableWt(daten[,c("Gender","Occupation")], weights=daten[,"Weight"])

       Occupation
Gender Employee Pensioner Student Worker
     m      210       370       0    330
     w      140       230     330      0
```

The toy data set in Table 3 is used in the following sections to get a brief overview about the theoretic explanations.

	Name	Year of birth	Gender	Citizenship	Occupation	Income	Weight
1	Max Mustermann	1978	m	AUT	Worker	35000	110.00
2	Josef Meier	1945	m	AUT	Pensioner	23500	70.00
3	Sabine Schnuller	1991	w	AUT	Student	7000	80.00
4	John Doe	1966	m	US	Employee	41200	120.00
5	Susan Rose	1989	w	AUT	Student	0	130.00
6	Markus Roller	1972	m	AUT	Employee	31100	90.00
7	Christoph Valon	1944	m	AUT	Pensioner	21400	150.00
8	Ulrike Mayer	1932	w	D	Pensioner	17600	150.00
9	Stefan Fuchs	1992	m	AUT	Worker	27500	130.00
10	Rainer Thomas	1950	m	AUT	Pensioner	25700	150.00
11	Julia Gross	1976	w	AUT	Employee	37000	140.00
12	Nadine Glatz	1987	w	AUT	Student	0	120.00
13	Makro Dilic	1990	m	AUT	Worker	21050	90.00
14	Sandra Stadler	1941	w	AUT	Pensioner	28500	80.00

Table 3: Toy data set of Example 2.3.

2.2 Concept of k-anonymity

In Section 1.2 three methods of protecting categorical data are described. After explaining the concept of frequency counts the k-anonymity method can be introduced. Let $Z_1, ..., Z_q$ the categorical key variables of a data set with n records. Then k-anonymity is achieved if each possible combination of key variables contains at least k units in the microdata set, i.e. $f_j \geq k$ and $\forall j \in \{1, ..., n\}$.

One method for achieving k-anonymity is to recode (see Section 1.2.1) categorical key variables into broader classes. Another common method is data suppression (see Section 1.2.2). In the R package `sdcMicro` the function `localSuppression()` can be used to achieve k-anonymity. The algorithm of this function tries to find an optimal solution to suppress as few values as possible as described in Templ et al. [2014b]. Table 4 shows a data set of 12 individuals with three categorical variables. The forth column shows the calculated frequencies, which are computed with the function `freqCalc()`. Four observations violate 2-anonymity (see Table 4 and particular at rows 4, 6, 8 and 11) and six observations violate 3-anonymity.

```
R > library(sdcMicro)
R > set.seed(23)
R > data <- read.csv2(file="EasyExampleData.csv", header=TRUE)
R > (fk <- freqCalc(data[1:12,], keyVars=c("Gender","Citizenship","Occupation")))
```

```
 --------------------------
4 obs. violate 2-anonymity
6 obs. violate 3-anonymity
 --------------------------
```

	Gender	Citizenship	Occupation	fk
1	m	AUT	Worker	2
2	m	AUT	Pensioner	3
3	w	AUT	Student	3
4	m	US	Employee	1
5	w	AUT	Student	3
6	m	AUT	Employee	1
7	m	AUT	Pensioner	3
8	w	D	Pensioner	1
9	m	AUT	Worker	2
10	m	AUT	Pensioner	3
11	w	AUT	Employee	1
12	w	AUT	Student	3

Table 4: Example of sample frequency counts.

50 percent of the observations in the data set of Table 4 violate 3-anonymity. The aim of this example is to gain 3-anonymity, i.e. $f_j \geq 3$ with $j \in \{1, ..., 12\}$. The above mentioned function `localSuppression()` is used to achieve 3-anonymity. The same example data as in Table 4 is used to reach 3-anonymity. Table 5 shows the new frequency counts, which are calculated with `freqCalc()`. It is clear to see that there are six suppressed values, whereby four values are suppressed in the variable "Occupation" and two values in "Citizenship". Note that a missing value (denoted as NA, see Table 5) can stand for any possible value, therefore the frequency count for observation 4 is 7.

```
R > (kanonymity <- localSuppression(data[1:12,4:6], k=3,
+                                   keyVars=c("Gender","Citizenship","Occupation")))
```

```
 -----------------------
[1] "Total Suppressions in the key variables -6"
[1] "Number of suppressions in the key variables "


 0 2 4
 ------------
[1] "3-anonymity == TRUE"


 -----------------------
```

	Gender	Citizenship	Occupation	fk
1	m	AUT	Worker	4
2	m	AUT	Pensioner	5
3	w	AUT	Student	5
4	m	<NA>	<NA>	7
5	w	AUT	Student	5
6	m	AUT	<NA>	7
7	m	AUT	Pensioner	5
8	w	<NA>	<NA>	5
9	m	AUT	Worker	4
10	m	AUT	Pensioner	5
11	w	AUT	<NA>	5
12	w	AUT	Student	5

Table 5: Example of achieving 3-anonymity using `localSuppression()`.

2.3 Approach to estimate population frequency counts

As discussed by Willenborg and de Waal [2001] the simplest approach to estimate F_j under the assumption of simple random sampling without replacement is given by $\hat{F}_j = \frac{f_j}{f}$, where $f = \frac{n}{N}$ is the sampling fraction.

In practice this estimator will not provide workable solutions, see discussion Willenborg and de Waal [2001], e.g. n is small and N is much higher then $f_j = 0$ implies $\hat{F}_j = 0$ and $f_j = 1$ implies $\hat{F}_j = w$, where w is the weight of every drawn record. If the sampling scheme is not simple random sampling and the weights are known for every record in the sample data set, then the population frequencies $\hat{F}_j$ are the sum of the weights of each record which has the same key combination, i.e. $\hat{F}_j = \sum_{i \in |\mathbf{U}_j|} w_i$, where $|\mathbf{U}_j|$ is a subpopulation of $\mathbf{U}$ and w_i are the weights of record $\mathbf{i}$ in subpopulation $\mathbf{U}_j$.

Example 2.4

A given data set with 14 observations (see Table 3) and two categorical key variables (`Gender` and `Occupation`) is considered. The underlying R code shows the calculation of random weights and frequencies with the function `freqCalc()`. `fk_ex1` is an object of class freqCalc and `fk` is the frequency of equal observations in the two key variables (Gender and Occupation) (see R package description Templ et al. [2014b]). `Fk` is the estimated frequency in the population with the above described method. Table 6 shows the variable `data_ex1` with the calculated `fk`'s and estimated `Fk`'s. The table was created with the R package `xtable` und the function `xtable()` [Dahl, 2014].

```
R >   data_ex1 <- data[,c("Name","Gender","Occupation")]
R >   set.seed(23)
R >   weights <- round(sample(50:150, size=length(data_ex1[,1]), replace=T),
+                      digits=-1)
R >   data_ex1 <- data.frame(data_ex1,weights)
R >   fk_ex1 <- freqCalc(data_ex1, keyVars=c("Gender","Occupation"),w="weights")
R >   data_ex1 <- data.frame(data_ex1,fk_ex1$fk,fk_ex1$Fk)
R >   levels(data_ex1$Occupation)

[1] "Employee"  "Pensioner" "Student"   "Worker"

R >   (levels(data_ex1$Occupation) <- c("E", "P", "S", "W"))

[1] "E" "P" "S" "W"
```

Figure 4 is a mosaic visualisation of the two key variables, with the new factor levels E, P, S and W. This figur illustrates the relative amount of the sample frequency counts.

2.3.1 Standard log-linear model

Log-linear models are used for modeling cell counts in contingency tables, see Definitions 2.1 and 2.3. These models declare how the expected cell count depends on levels of the categorical (key) variables. Let $\boldsymbol{\mu} = (\mu_1, ..., \mu_C)'$ denote the expected counts for the number of C cells of a contingency table. As in Agresti [2002] multidimensional log-linear models for positive Poisson means have the following form:

$$log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\lambda} \quad , \tag{3}$$

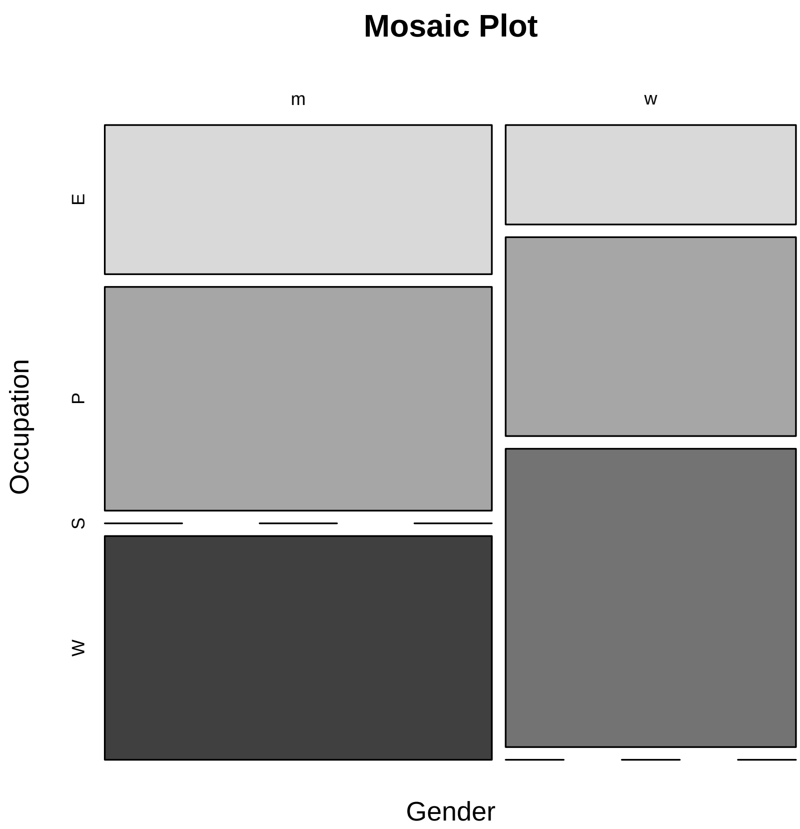


Figure 4: Mosaic plot of the sample frequency counts of data_ex1[,c("Gender","Occupation")].

	Name	Gender	Occupation	Weights	f_k	$\hat{F}_k$
1	Max Mustermann	m	Worker	110.00	3	330.00
2	Josef Meier	m	Pensioner	70.00	3	370.00
3	Sabine Schnuller	w	Student	80.00	3	330.00
4	John Doe	m	Employee	120.00	2	210.00
5	Susan Rose	w	Student	130.00	3	330.00
6	Markus Roller	m	Employee	90.00	2	210.00
7	Christoph Valon	m	Pensioner	150.00	3	370.00
8	Ulrike Mayer	w	Pensioner	150.00	2	230.00
9	Stefan Fuchs	m	Worker	130.00	3	330.00
10	Rainer Thomas	m	Pensioner	150.00	3	370.00
11	Julia Gross	w	Employee	140.00	1	140.00
12	Nadine Glatz	w	Student	120.00	3	330.00
13	Makro Dilic	m	Worker	90.00	3	330.00
14	Sandra Stadler	w	Pensioner	80.00	2	230.00

Table 6: Example of the simplest approach to estimate population frequencies.

where $log(\boldsymbol{\mu})$ is a $C \times 1$ vector containing the logarithms of the expected frequencies, $\mathbf{X}$ is a $C \times p$ model matrix and $\boldsymbol{\lambda}$ is a $p \times 1$ vector of model parameters.

Function `glm()` is used to fit log-linear models in R. The main arguments are formula, family and data, whereby the family is set to poisson.

Example 2.5 (Standard log-linear model)

In dependence on Section 4 the `eusilcS` data set is used as a sample of Austria's population. For every key the frequency counts and weights are calculated. The frequency counts are also calculated for the population (see variable `keysPop`). Variable `dataS` includes all possible population keys and the frequency counts of the sample `eusilcS`. Table 7 shows 15 keys of the table `dataS` and the corresponding frequency counts and weights. Table 8 shows the summary of the `glm()` output with the standard log-linear model. It is clear to see that the intercept is significantly non-zero. The p-value of the federal states is in most cases not significant, which means that the variable `db040` has not a significant contribution. The same holds for the variable `rb090`.The contribution of the variables `hsize`, `age` and `pb220a` is statistically significant at $\alpha = 0.05$.

```
R > keyVars <- c("db040","hsize","rb090", "age", "pb220a")
R > fk <- freqCalc(eusilcS, keyVars)$fk #sum(Fk==1)
R > eusilc <- cbind(eusilcS, fk)
```

	db040	hsize	rb090	age	pb220a	fk	weights
2430	Upper Austria	4	male	42.00	AT	9.00	30.70
10251	Lower Austria	6	female	20.00	Other	0.00	0.00
3455	Upper Austria	3	male	41.00	EU	1.00	6.86
11280	Vienna	3	female	40.00	Other	0.00	0.00
11901	Styria	4	male	52.00	Other	0.00	0.00
385	Carinthia	1	male	51.00	AT	4.00	77.57
4459	Carinthia	5	male	8.00	AT	0.00	0.00
11376	Salzburg	3	male	42.00	Other	0.00	0.00
4655	Salzburg	6	male	12.00	AT	0.00	0.00
3854	Upper Austria	4	female	45.00	Other	1.00	8.19
12062	Styria	5	female	55.00	Other	0.00	0.00
3825	Upper Austria	1	female	32.00	Other	1.00	7.16
9160	Vienna	1	male	44.00	EU	0.00	0.00
4831	Lower Austria	6	male	15.00	AT	0.00	0.00
869	Lower Austria	2	female	58.00	AT	7.00	8.33

Table 7: 15 random keys of table dataS.

```
R > form_keys <- as.formula(paste(" ~ ", "db040 + hsize + rb090 + age + pb220a"))
R > (form_standard <- as.formula(paste(c("fk", as.character(form_keys)), collapse = "")))
```

```
fk ~ db040 + hsize + rb090 + age + pb220a

R > mod_standard <- glm(form_standard, data = dataS, family = poisson())
R > mu_standard <- fitted(mod_standard)
R > summary(mu_standard)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.058   0.187   0.920   1.130   1.920   5.030
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.2405	0.0636	3.78	0.0002
db040Carinthia	0.0471	0.0624	0.75	0.4504
db040Lower Austria	0.3942	0.0559	7.05	0.0000
db040Salzburg	-0.0425	0.0642	-0.66	0.5081
db040Styria	0.2247	0.0567	3.96	0.0001
db040Tyrol	0.0318	0.0606	0.53	0.5995
db040Upper Austria	0.3486	0.0561	6.22	0.0000
db040Vienna	0.3698	0.0569	6.49	0.0000
db040Vorarlberg	-0.1010	0.0686	-1.47	0.1408
hsize	-0.0618	0.0068	-9.11	0.0000
rb090female	0.0067	0.0206	0.33	0.7431
age	0.0108	0.0005	22.01	0.0000
pb220aEU	-2.8937	0.0700	-41.35	0.0000
pb220aOther	-2.1246	0.0415	-51.16	0.0000

Table 8: Output of the standard log-linear model.

2.3.2 Clogg and Eliason method

As described in Clogg and Eliason [1987], Agresti [2002], Shlomo and Skinner [2008] the Clogg and Eliason approach additional considers the survey weights towards Equation (3). They extend the log-linear model from Equation 3 with an offset term $\mathbf{z} = (z_1, ..., z_C)'$ and $z_k = \frac{f_k}{\hat{F}_k}$ (see Definition 2.1), where $\hat{F}_k$ is the sum of survey weights across sample units in cell k. This consideration leads to the following adaption of the log-linear model:

$$log(\boldsymbol{\mu}) = log(\mathbf{z}) + \mathbf{X}\boldsymbol{\lambda} \quad . \tag{4}$$

Example 2.6 (Clogg and Eliason model)

To fit the Clogg and Eliason model the formula of the standard log-linear model is used. The `glm()` *argument* `offset` *is set to* $z_k = \frac{f_k}{\hat{F}_k}$. *To handle keys with zero the* $z_k = \frac{f_k}{\hat{F}_k}$ *are linear* *transformed, which only affects the intercept term. Table 9 yields the summary of the Clogg and Eliason example. The estimates of the Clogg and Eliason model yields the same significant variables, whereby the z-values differ in comparison to the standard method (see Table 8).*

```
R > z_k <- dataS$fk/dataS$weights
R > z_k[z_k=="NaN"] <- 0
R > z_k <- log(z_k + 0.1)
R > form_standard

fk ~ db040 + hsize + rb090 + age + pb220a

R > mod_EC <- glm(form_standard, data = dataS, family = poisson(), offset = z_k)
R > mu_EC <- fitted(mod_EC)
R > summary(mu_EC)

   Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
  0.075    0.173    0.529    1.130    1.460   22.100
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	1.7056	0.0664	25.67	0.0000
db040Carinthia	-0.0155	0.0623	-0.25	0.8040
db040Lower Austria	0.1280	0.0559	2.29	0.0220
db040Salzburg	0.0234	0.0642	0.36	0.7162
db040Styria	0.1396	0.0567	2.46	0.0138
db040Tyrol	0.0262	0.0606	0.43	0.6651
db040Upper Austria	0.2406	0.0560	4.30	0.0000
db040Vienna	0.1985	0.0571	3.48	0.0005
db040Vorarlberg	-0.0753	0.0685	-1.10	0.2716
hsize	-0.0178	0.0072	-2.49	0.0129
rb090female	-0.0071	0.0205	-0.34	0.7304
age	0.0049	0.0006	8.78	0.0000
pb220aEU	-1.9242	0.0700	-27.49	0.0000
pb220aOther	-1.3399	0.0416	-32.19	0.0000

Table 9: Output of the Clogg and Eliason model.

2.3.3 Pseudo maximum likelihood method

The fitted values for a linear model are solutions to the likelihood equations. We derive likelihood equations using Equation (3) for a log-linear model. For a vector of frequency counts $\mathbf{f}$ with

$\boldsymbol{\mu} = \mathbb{E}(f)$, the model is given by $log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\lambda}$, for which $log(\mu_i) = \sum_j x_{ij} \cdot \lambda_j$ for $\forall i \in \{1, ..., C\}$. The log likelihood for Poisson sampling is:

$$L(\boldsymbol{\mu}) = \sum_i f_i \cdot log(\mu_i) - \sum_i log(\mu_i) \quad . \tag{5}$$

Through Equations (3) and (5) the pseudo maximum likelihood approach yields the following equation:

$$log(\hat{\mathbf{F}}) = \mathbf{X}\boldsymbol{\lambda} \quad . \tag{6}$$

$\hat{F}_k$ is the sum of survey weights across sample units in cell k and $\hat{\mathbf{F}} = (\hat{F}_1, \hat{F}_2, ..., \hat{F}_C)'$.

Example 2.7 (Pseudo maximum likelihood model)

The $\hat{F}_k$ are scaled by a constant to avoid numerical problems. Scaling by variable `sf` do not affect the estimated counts with $sf = \frac{\sum_k f_k}{\sum_k \widehat{F_k}}$. Variable `form_pse` shows the formula for the `glm()` function. Table 10 shows the summary of the pseudo maximum likelihood example. The results differ to the above mentioned models in the variables Lower Austria, Vienna and hsize (see Tables 8 and 9).

```
R > sf <- sum(dataS$fk)/sum(dataS$weights)
R > eF_k <- round(dataS$weights*sf)
R > dataS_pse <- data.frame(dataS, eF_k)
R > (form_pse <- as.formula(paste(c("eF_k", as.character(form_keys)), collapse = "")))

eF_k ~ db040 + hsize + rb090 + age + pb220a

R > mod_pse <- glm(form_pse, data = dataS, family = poisson())
R > mu_pse <- fitted(mod_pse)
R > summary(mu_pse)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.076   0.205   1.050   1.140   1.950   4.270
```

2.3.4 Weighted log-linear model

The weighted log-linear model is an extension of the standard log-linear model, that also considers the weights of each cell, i.e. the linear predictor for $\boldsymbol{\mu}$ also contains the weights as an

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.1962	0.0604	3.25	0.0012
db040Carinthia	-0.0306	0.0585	-0.52	0.6010
db040Lower Austria	0.0151	0.0540	0.28	0.7802
db040Salzburg	0.0172	0.0588	0.29	0.7696
db040Styria	0.1291	0.0531	2.43	0.0150
db040Tyrol	0.0629	0.0559	1.13	0.2604
db040Upper Austria	0.2085	0.0527	3.96	0.0001
db040Vienna	0.0779	0.0548	1.42	0.1554
db040Vorarlberg	-0.2015	0.0650	-3.10	0.0019
hsize	-0.0112	0.0065	-1.71	0.0871
rb090female	-0.0185	0.0204	-0.91	0.3652
age	0.0120	0.0005	24.63	0.0000
pb220aEU	-2.8180	0.0681	-41.36	0.0000
pb220aOther	-2.0953	0.0412	-50.84	0.0000

Table 10: Output of the pseudo maximum likelihood model.

explanatory variable. The weighted log-linear model is given by:

$$log(\boldsymbol{\mu}) = \tilde{\mathbf{X}}\boldsymbol{\lambda} \quad , \tag{7}$$

where $log(\boldsymbol{\mu})$ is a $C \times 1$ vector containing the logarithms of the expected frequencies, $\tilde{\mathbf{X}}$ is a $C \times q$ model matrix and $\boldsymbol{\lambda}$ is a $q \times 1$ vector of model parameters.

Example 2.8 (Weighted log-linear model)

The predictor variable `form_keys` *is extended with the variable* `weights`*. Formula* `form_w` *specifies the response and predictors for the* `glm()` *function. Table 11 shows the summary of the weighted log-linear example. The intercept is not singificantly non-zero, whereby the other results conform with the standard and EC model.*

```
R > form_zw <- as.formula(paste(c(form_keys,"weights"),collapse="+"))
R > (form_w <- as.formula(paste(c("fk", as.character(form_zw)), collapse = "")))

fk ~ db040 + hsize + rb090 + age + pb220a + weights

R > mod_w <- glm(form_w, data = dataS, family = poisson())
R > mu_w <- fitted(mod_w)
R > summary(mu_w)

  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.062   0.189   0.832   1.130   1.890   9.210
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	0.1145	0.0639	1.79	0.0730
db040Carinthia	0.0422	0.0624	0.68	0.4988
db040Lower Austria	0.3780	0.0559	6.76	0.0000
db040Salzburg	-0.0495	0.0642	-0.77	0.4410
db040Styria	0.1824	0.0568	3.21	0.0013
db040Tyrol	0.0092	0.0606	0.15	0.8793
db040Upper Austria	0.2896	0.0561	5.16	0.0000
db040Vienna	0.3578	0.0570	6.28	0.0000
db040Vorarlberg	-0.0719	0.0686	-1.05	0.2943
hsize	-0.0581	0.0067	-8.62	0.0000
rb090female	0.0084	0.0206	0.41	0.6824
age	0.0091	0.0005	18.02	0.0000
pb220aEU	-2.6856	0.0708	-37.91	0.0000
pb220aOther	-1.9348	0.0427	-45.26	0.0000
weights	0.0123	0.0006	20.64	0.0000

Table 11: Output of the weighted log-linear model.

This study considers model based methods to estimate population frequency counts, as described by Carlson [2002a,b], Shlomo and Skinner [2008]. It is assumed that the cell frequencies are generated independently from Poisson distributions with individual rates λ_j, i.e. $F_j \sim Poisson(\lambda_j)$, $j \in \{1, ..., C\}$. This assumption holds if the sampling design is simple random sampling without replacement, then the distribution is hypergeometric with given N, C and π_j. If the number of cells is large enough each cell frequency may be approximated by a binomial distribution with parameters N and inclusion probability π_j. Since the population size is quite large and π_j small due to large C the Poisson distribution is used to approximate the binomial with $\lambda_j = N\pi_j$.

Surveys almost always will not drawn with simple random sampling. Complex sampling schemes are employed especially stratification methods, which are mentioned in Section 1.3 and also considered for the numerical study in Section 4. In Section 3 the effect of complex sampling schemes to the considered risk measures is discussed. As described by Shlomo and Skinner [2008] the assumption that $F_j \sim Poisson(\lambda_j)$, with $j \in \{1, ..., C\}$ and that the λ_j obey the log-linear models are unaffected by stratified sampling.

3 Disclosure risk

A considerable amount of research has been done in the area of statistical disclosure risk. This section is based on Carlson [2002a,b], Hundepool et al. [2010], Willenborg and de Waal [2001], Templ et al. [2014a] and Shlomo and Skinner [2008].

Definition 3.1 (Disclosure Risk)
Disclosure risk is the risk that disclosure will arise if a given data set is released.

It will be assumed that the risk r takes a non-negative real value and a risk of zero indicates no risk, i.e. $r \geq 0$ and $r = 0 \Rightarrow$ no risk. Measuring the disclosure risk in a microdata set is a key task and is applied in Section 4. Risk measures are essential to be able to decide, if the data set is protected enough to be released. If the data set is not protected enough certain protection methods have to be used, see Section 1.2.

3.1 Measuring the disclosure risk of categorical variables

In this study the main focus concerns on measuring the disclosure risk of categorical key variables (see Definition 1.5 in Section 1.2) from a random sample of a finite population. The aim is to define a local and global probability measure for given records that expresses the re-identification risk. A further assumption is that there is no measurement error, meaning that the recorded microdata and the prior information of the intruder are the same. A sample S is randomly drawn with a given sampling design from a finite population U. See discussion Shlomo and Skinner [2008] following assumptions have to hold:

- no measurement error
- random sampling design
- all records of subpopulation $\mathbf{U}_i$ have the same inclusion probability

Let F_j be the population frequency count (see Definition 2.3) in cell $j \in \{1, ..., C\}$ of the contingency table and C the amount of all cells. Under the assumptions that F_j and one record are known to the intruder, the probability that the record $\mathbf{x} \in \mathbf{U}$ may be identified is $\frac{1}{F_j}$, where j is the cell to which the record belongs, i.e. $x \in \mathbf{U}_j \subseteq \mathbf{U}$. The identification risk is maximum when the record is population unique, i.e. $F_j = 1$. In practice rare population combinations should be avoided (see Section 1.2, e.g. k-anonymity).

F_j is usually not known since in statistics in most cases information is on samples collected and only few information about the population is known. Therefore population parameters have to be estimated. The goal is to model and estimate the population frequency structure, i.e. F_j, T_j and especially T_1 [Carlson, 2002a], which is defined as the number of unique records in the population, based on sample information (see Definitions 2.3 and 2.4).

At this point we consider F_j as a stochastic variable without specific distribution assumptions. A measure of identification risk is given by

$$\mathbb{E}(1/F_j) = \sum_{i \in \mathbb{N}} \frac{1}{i} \mathbb{P}(F_j = i) \quad , \tag{8}$$

where $\mathbb{P}(F_j = i)$ denotes the probability that $F_j = i$, with $i = \{1, 2, ..., N\}$. If $i = 1$, we receive the probability of population uniqueness $\mathbb{P}(F_j = 1)$, which is the first term in the sum in (8).

As mentioned above we consider a random sample S of a finite population $\mathbf{U}$ of size N. The sample data is available to the intruder. Let f_j be the sample frequency counts (see Definition 2.3). This leads to two measures of interest:

$$m_1 = \mathbb{E}(1/F_j | f_j) \quad , \tag{9}$$

$$m_2 = \mathbb{P}(F_j = 1 | f_j) \quad . \tag{10}$$

Under random sampling the pairs (F_j, f_j) are independent and the first measure (9) is the conditional expactation of $1/F_j$ and second (10) the conditional probability that $F_j = 1$ given f_j. When $f_j = 1$, (9) is highest, which is the worst case. Additionally the following holds for (10) as described in Shlomo and Skinner [2008]:

$$\mathbb{P}(F_j = 1 | f_j = i) = \begin{cases} \in [0, 1] \, , & \text{if } i = 1 \\ 0, & \text{if } i \geq 2 \end{cases}$$

Consideration of the worst cases leads to the focus on the following measures:

$$m_{1j} = \mathbb{P}(F_j = 1 | f_j = 1) \quad , \tag{11}$$

$$m_{2j} = \mathbb{E}(1/F_j | f_j = 1) \quad . \tag{12}$$

The measures given in Equation (11) and (12) are per observation measures and their values can vary between observations.

Observation-level measures are discussed above. In the following, a measure for the global risk is described. This leads to consideration of aggregating observation-level measures given by

$$\hat{\tau}_1 = \sum_{\{j:f_j=1\}} m_{1j} = \sum_{\{j:f_j=1\}} \mathbb{P}(F_j = 1|f_j = 1) \quad , \tag{13}$$

$$\hat{\tau}_2 = \sum_{\{j:f_j=1\}} m_{2j} = \sum_{\{j:f_j=1\}} \mathbb{E}(1/F_j|f_j = 1) \quad . \tag{14}$$

The global risk measure $\hat{\tau}_1$ is the expected number of sample uniques that are population unique and $\hat{\tau}_2$ is the expected number of correct matches for sample uniques [Shlomo and Skinner, 2008]. If the count of combinations C is large, $\hat{\tau}_1$ will closely approximate τ_1,

$$\hat{\tau}_1 \overset{C\to\infty}{\longrightarrow} \tau_1 = \sum_{j\geq 1} \mathbb{1}(f_j = 1, F_j = 1) \quad , \tag{15}$$

The same holds for $\hat{\tau}_2$ with:

$$\hat{\tau}_2 \overset{C\to\infty}{\longrightarrow} \tau_2 = \sum_{j\geq 1} \frac{\mathbb{1}(f_j = 1)}{F_j} \quad . \tag{16}$$

The Population consists of N entities and the key divides the population into C cells. Each cell j is assigned a parameter $\lambda_j > 0$ satisfying $\sum_{j=1}^{C} \lambda_j = 1$ and a random independent variable F_j which is the population frequency in the cell j. With the assumption that $F_j \sim Poisson(\lambda_j)$, $j \in \{1, ..., C\}$, the following probability is given

$$\mathbb{P}(F_j = i) = \frac{\lambda_j^i e^{-\lambda_j}}{i!}, \; i \in \{0, 1, 2, 3, ...\} \quad . \tag{17}$$

The mean and variance of the random variables F_j is both equal to λ_j. It is also assumed that $f_j|F_j \sim Binomial(F_j, \pi_j)$, whereby π_j is the inclusion probability.

Remark

Note that a sample drawn using Bernoulli sampling on a Poisson distributed population will remain Poisson.

For the sample frequency counts holds $f_j = Poisson(\lambda_j \pi_j)$. To estimate the number of sample uniques that are population unique the following probability has to be calculated

$$\mathbb{P}(F_j = 1|f_j = 1) = e^{-\lambda_j(1-\pi_j)} \quad . \tag{18}$$

For the estimated risk measures $\hat{\tau}_1$ and $\hat{\tau}_2$ the following holds under the assumption of Poisson distribution

$$\hat{\tau}_1 = \sum_j \mathbb{1}(f_j = 1)\mathbb{P}(F_j = 1|f_j = 1) = \sum_{\{j:f_j=1\}} e^{-\lambda_j(1-\pi_j)} \quad , \tag{19}$$

$$\hat{\tau}_2 = \sum_j \mathbb{E}(\frac{1}{F_j}|f_j = 1) = \sum_{\{j:f_j=1\}} \frac{1 - e^{-\lambda_j(1-\pi_j)}}{\lambda_j(1-\pi_j)} \quad . \tag{20}$$

The assumptions at the end of Section 2 that $F_j \sim Poisson(\lambda_j)$ and that the λ_j fit the log-linear model are unaffected by a complex sampling scheme [Shlomo and Skinner, 2008]. If the sampling scheme is not SRS the risk measures $m_{1j} = e^{-\lambda_j(1-\pi_j)}$ and $m_{2j} = \frac{1-e^{-\lambda_j(1-\pi_j)}}{\lambda_j(1-\pi_j)}$ may be affected. But these expressions still hold if $\mathbb{P}(f_j = 1|F_j) = F_j\pi_j(1-\pi_j)^{F_j-1}$. In generall an useable approximation $\mathbb{P}(f_j = 1|F_j) \approx F_j\pi_j(1-\pi_j)^{F_j-1}$ sufficies good results. The next Section shows the affect of stratified sampling on the estimated risk measures $\hat{\tau}_1$ and $\hat{\tau}_2$.

4 Numerical study

4.1 Data

The European Union Statistics on Income and Living Conditions (EU-SILC) is a panel survey, where information about living conditions of private households is collected yearly. Since 2003, Austria is one of 31 countries, which are represented in this survey. In this study a EU-SILC population data is simulated using the R package `simPopulation` [Alfons et al., 2011]. The simulated data follows a close-to-reality approach and therefore a real-world situation can be assumed. For simulation details, see Alfons et al. [2011]. In this approach a data set with about 8 million records is created, which is nearly the total population amount of Austria.

4.1.1 Synthetic survey data: eusilcS

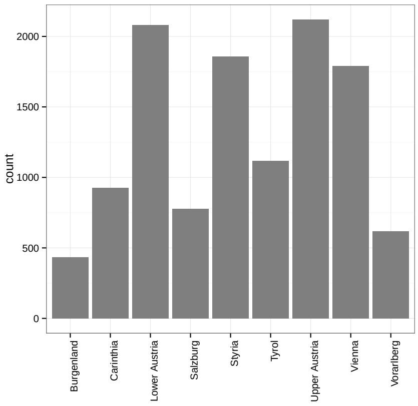
Variable	Description
db030	houshold ID
hsize	number of persons in the household
db040	federal state in which the household is located
age	person's age
rb090	person's gender
pl030	person's economic status
pb220a	person's citizenship
netIncome	personal net income
db090	household sample weights
rb050	personal sample weights

Table 12: Considered variables of the data frame eusilcS.

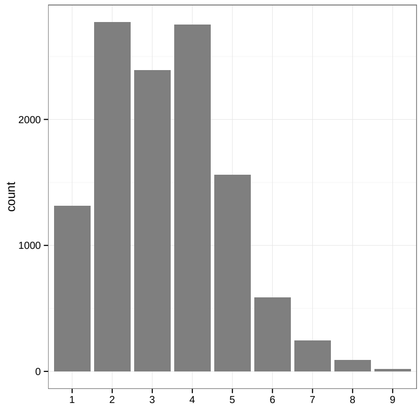
The R data set `eusilcS` is synthetically generated from real Austrian EU-SILC data [Alfons et al., 2011] from 2006. `eusilcS` is a data frame with 11725 observations, 18 variables and 4641 households and it is included in the R package `simPopulation`. Table 12 shows the ten considered variables of this study and Figure 5 shows six barplots of the variables `db040` (a), `hsize` (b), `age` (c), `pl030` (d), `pb220a` (e) and `netIncome` (f), whereby the unweighted values are shown. The particular factor levels of the variables are shown as well in Figure 5. Table 13 shows the first 12 observations of the `eusilcS` data set. The last three observations include missing values.

Remark

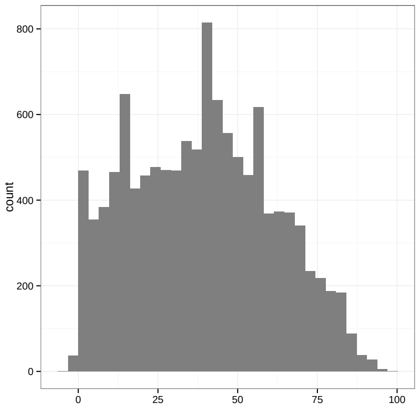
The sample weights `rb050` in the data set `eusilcS` are 100 times smaller than the real population size, just because of the reason for computational speed within the examples of the package.



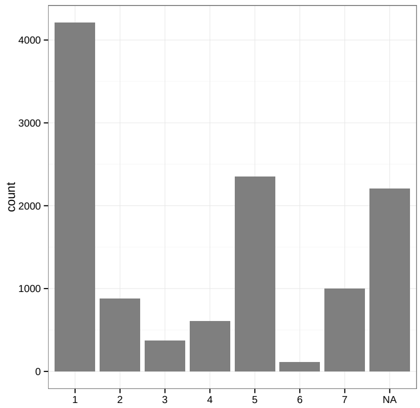
(a) Barplot of the federal states in which the households are located.



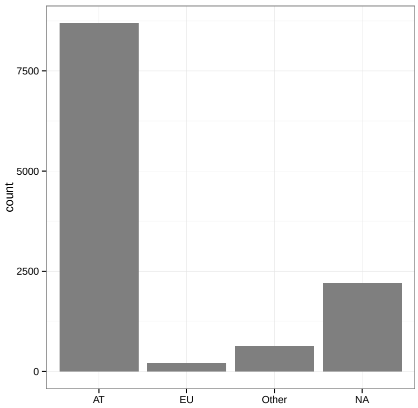
(b) Number of persons in the household.



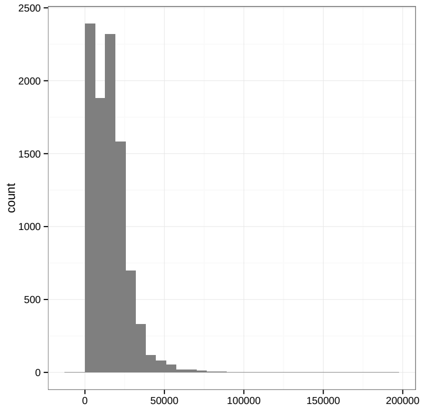
(c) Plot of the person's age.



(d) The person's economic status.



(e) Barplot of the person's citizenship.



(f) Histogram of the personal net income.

Figure 5: Visualisation of the distribution of certain variables of the data set eusilcS.

	hsize	db040	rb090	age	pl030	pb220a	netIncome	rb050
9292	2	Salzburg	male	72	5	AT	22675.48	7.82
9293	2	Salzburg	female	66	5	AT	16999.29	7.82
7227	1	Upper Austria	female	56	2	AT	19274.21	8.79
5275	1	Styria	female	67	5	AT	13319.13	8.11
7866	3	Upper Austria	female	70	5	AT	14365.57	7.51
7867	3	Upper Austria	male	46	3	AT	0.00	7.51
7868	3	Upper Austria	male	37	1	Other	21911.24	7.51
9860	5	Salzburg	male	41	1	AT	11682.22	6.75
9861	5	Salzburg	female	35	3	AT	5481.40	6.75
9862	5	Salzburg	female	9				6.75
9863	5	Salzburg	male	6				6.75
9864	5	Salzburg	female	3				6.75

Table 13: The first 12 persons of eusilcS.

4.1.2 Simulation of Austrian EU-SILC data

The above mentioned R package `simPopulation` contains the function `simEUSILC()` to simulate EU-SILC population data. The `simEUSILC()` function needs the `eusilcS` synthetic survey data set for simulation that is available in the package too.

It is assumed that there are no missing values in the population and also in the sample. If there are missing values in the sample the risk is almost always overestimated, because the measurement error biased the risk estimation. To avoid measurement errors, the missing values are replaced by estimated values. In this study the R function `kNN()` from the package `VIM` [Templ et al., 2013] is used, which is described in Section 1.4. After the imputation of the estimated values for all missing values in `eusilcS`, Austria's population is simulated, with the function `simEUSILC()` and the `eusilcS` data set. The whole population is simulated to compare the real disclosure risk with the estimated risks, which is the main idea of this numerical study, i.e. to see if the estimates of different simulation designs are useful. The simulated population has no missing values, which implies that there are no missing values in the samples, because there are not any protection methods applied.

4.2 Results

The empirical approach is described in Figure 6. First function `kNN()` is applied on the data set `eusilcS`. After the k-nearest neighbour imputation the population is simulated with the function `simEUSILC()`. Three different kinds of disclosure risk scenarios are used and closer described in Table 14. This table shows which categorical variable is assumed to be a categorical key variable. If the cell in column scenario is indexed with 1, the variable is considered as categorical

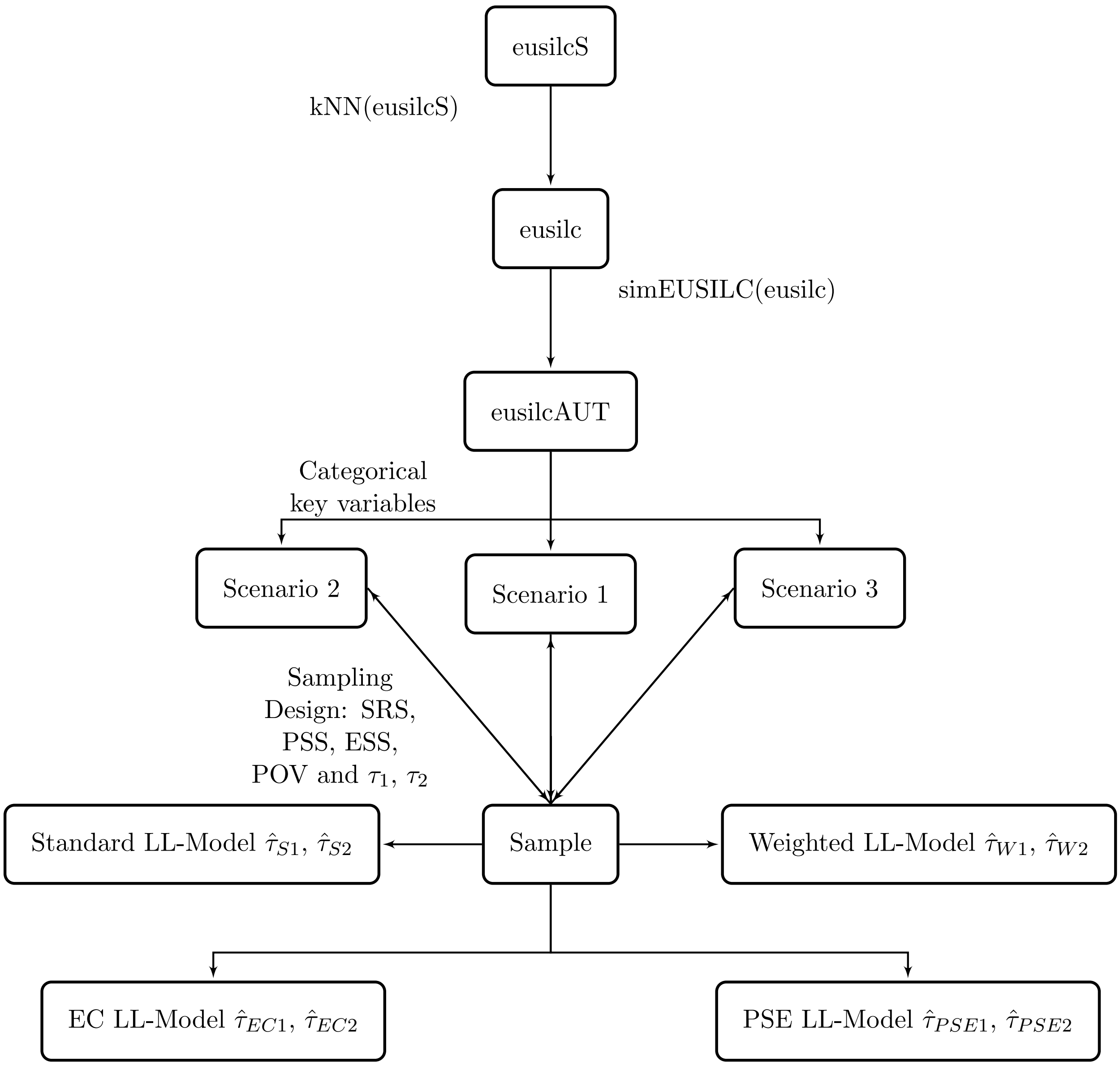


Figure 6: Diagram describing the workflow of the numerical study.

key variable. Each scenario has a different amount of keys. Four different sampling methods (SRS, proportional & equal stratified sampling, oversampling; see Section 1.3) are tested for each scenario. For every sampling method and disclosure risk scenario a few measures are estimated to compare the real disclosure risk with the estimated risk of the different log-linear models. 19 measures are described in Table 15. There are mainly two risk measures of interest:

- number of sample uniques that are population unique τ_1 (see Equation (13));
- number of correct matches for sample uniques τ_2 (see Equation (14)).

These two risk measures are estimated for each log-linear model. The difference between the estimated and real risk measures shows if the risk is well estimated or not. If $\hat{\tau}_{model1} - \tau_1 = 0$ or $\hat{\tau}_{model2} - \tau_2 = 0$, then the risk is perfectly estimated. If the difference is smaller then zero, the risk is underestimated and if it's higher, the risk is overestimated. Per simulation run 100 samples are drawn from the population with the R function `runSimulation()` [Alfons et al., 2010]. Each sample includes 4641 housholds.

Variable	Description	Scenario 1	Scenario 2	Scenario 3
hsize	number of persons in the household	1	1	1
db040	federal state in which the household is located	1	1	0
age	person's age	1	1	1
rb090	person's gender	1	1	1
pl030	person's economic status	0	1	1
pb220a	person's citizenship	1	1	0
netIncomeCat	personal net income divided into 15 intervals	0	1	1

Table 14: Categorical key variables of the three different disclosure risk scenarios.

	Name	Description
1	τ_1	real number of sample uniques that are population unique
2	τ_2	real number of correct matches for sample uniques
3	$\hat{\tau}_{S1}$	estimated number of sample uniques that are population unique using the standard log-linear method
4	$\hat{\tau}_{S2}$	estimated number of correct matches for sample uniques using the standard log-linear method
5	$\hat{\tau}_{EC1}$	estimated number of sample uniques that are population unique using the EC approach
6	$\hat{\tau}_{EC2}$	estimated number of correct matches for sample uniques using the EC approach
7	$\hat{\tau}_{PSE1}$	estimated number of sample uniques that are population unique using the PSE approach
8	$\hat{\tau}_{PSE2}$	estimated number of correct matches for sample uniques using the PSE approach
9	$\hat{\tau}_{W1}$	estimated number of sample uniques that are population unique using the weighted log-linear method
10	$\hat{\tau}_{W2}$	estimated number of correct matches for sample uniques with using weighted log-linear method
11	η_{S1}	difference between $\hat{\tau}_{S1} - \tau_1$
12	η_{S2}	difference between $\hat{\tau}_{S2} - \tau_2$
13	η_{EC1}	difference between $\hat{\tau}_{EC1} - \tau_1$
14	η_{EC2}	difference between $\hat{\tau}_{EC2} - \tau_2$
15	η_{PSE1}	difference between $\hat{\tau}_{PSE1} - \tau_1$
16	η_{PSE2}	difference between $\hat{\tau}_{PSE2} - \tau_2$
17	η_{W1}	difference between $\hat{\tau}_{W1} - \tau_1$
18	η_{W2}	difference between $\hat{\tau}_{W2} - \tau_2$
19	fk1	amount of sample frequency counts with characteristic 1, i.e. $sum(fk == 1)$

Table 15: List of all measures for each sampling method.

4.2.1 Disclosure risk scenario 1

`Disclosure risk scenario 1` calculates the risk for five categorical key variables, see Table 14. These key variables divide the population into 8448 keys and for every key the population frequency counts are calculated. The population frequency counts are shown in Figure 7. There are 33 unique persons in the population and 149 cells with less than 6 records. Figure 7 and 8 show the distribution of the population frequency counts with five categorical key variables. Figure 8 shows that there are less keys with with less counts. So the re-identification risk will be low. The distribution in Figure 8 must not disagree with the Poisson assumption, because there are C distribution parameters λ_j, $j \in \{1, 2, ..., C\}$.

To fit the log-linear models (standard, EC, PSE, weighted) the R function `glm()` of the standard package `stats` is used. The first and most important function argument of `glm()` is a formula specifing the response, predictors and possible interactions. In other words, from this formula, `glm()` builds a model (design) matrix and applies the (chosen family of) regression method on it. The following formulas are applied:

```
R > keyVars_S1 <- c("db040","hsize","rb090", "age", "pb220a")
R > f <- as.formula(paste(" ~ ", "db040 + hsize + rb090 + age + pb220a +
+                          age:rb090 + age:hsize + hsize:rb090"))
R > (f_standard_llm <- as.formula(paste(c("counts", as.character(f)),
+                          collapse = "")))

counts ~ db040 + hsize + rb090 + age + pb220a + age:rb090 + age:hsize +
    hsize:rb090

R > (f_pse_llm <- as.formula(paste(c("estimated_Fk", as.character(f)),
+                          collapse = "")))

estimated_Fk ~ db040 + hsize + rb090 + age + pb220a + age:rb090 +
    age:hsize + hsize:rb090

R > (f_weighted_llm <- as.formula(paste(c("counts",
+                        as.character(as.formula(paste(c(f,"weights"),collapse="+")))),
+                                collapse = "")))

counts ~ db040 + hsize + rb090 + age + pb220a + age:rb090 + age:hsize +
    hsize:rb090 + weights
```

Variable `keyVars_S1` includes the considered categorical key variables of `scenario 1`. `f_standard_llm, f_pse_llm and f_weighted_llm` describe the model to be fitted. The predictor has the form `response ~ predictors`. For example, a specification of the form `age:rb090` indicates the interaction for all categories of the predictors `age` and `rb090`. This 2-way intercation model performs best for `disclosure risk scenario 1`. For the EC approach the formula `f_standard_llm` is used and the offset term in function `glm()` is set to `offset` $= \frac{f_k}{\hat{F}_k}$. $\hat{F}_k$ are the estimated population frequency counts. They are calculated as the sum of weights across sample units in cell k. $\hat{F}_k$ is the response for the PSE model.

	1	2	3	4	5
Run	1.00	2.00	3.00	4.00	5.00
Sample	1.00	2.00	3.00	4.00	5.00
fk1	1564.00	1582.00	1544.00	1480.00	1582.00
τ_1	0.00	0.00	0.00	0.00	0.00
$\hat{\tau}_{S1}$	0.00	0.00	0.00	0.00	0.00
$\hat{\tau}_{EC1}$	0.00	0.00	0.00	0.00	0.00
$\hat{\tau}_{PSE1}$	0.00	0.00	0.00	0.00	0.00
$\hat{\tau}_{W1}$	0.00	0.00	0.00	0.00	0.00
τ_2	5.46	5.86	5.87	5.35	6.22
$\hat{\tau}_{S2}$	4.48	4.51	4.51	4.69	4.46
$\hat{\tau}_{EC2}$	4.44	4.47	4.46	4.64	4.42
$\hat{\tau}_{PSE2}$	4.48	4.51	4.51	4.69	4.46
$\hat{\tau}_{W2}$	3.95	4.01	3.98	4.08	4.02
η_{S1}	0.00	0.00	0.00	0.00	0.00
η_{EC1}	0.00	0.00	0.00	0.00	0.00
η_{PSE1}	0.00	0.00	0.00	0.00	0.00
η_{W1}	0.00	0.00	0.00	0.00	0.00
η_{S2}	-0.98	-1.35	-1.36	-0.66	-1.75
η_{EC2}	-1.03	-1.39	-1.41	-0.71	-1.80
η_{PSE2}	-0.98	-1.35	-1.36	-0.66	-1.75
η_{W2}	-1.51	-1.84	-1.88	-1.27	-2.20

Table 16: The first five simulation results of scenario 1.

For each method like SRS, proportional stratified sampling, equal stratified sampling and oversampling, a simulation function for `scenario 1` is used to calculate the risk measures, given in Table 15. Table 16 shows the first five simulation run results of SRS and disclosure risk scenario 1. There are about 10 per cent sample uniques in each sample, but the number of sample uniques that are population unique is zero within the first 5 runs, see τ_1 in Table 16. The number of correct matches for sample uniques is also very low, see τ_2. Four Figures 9, 10,

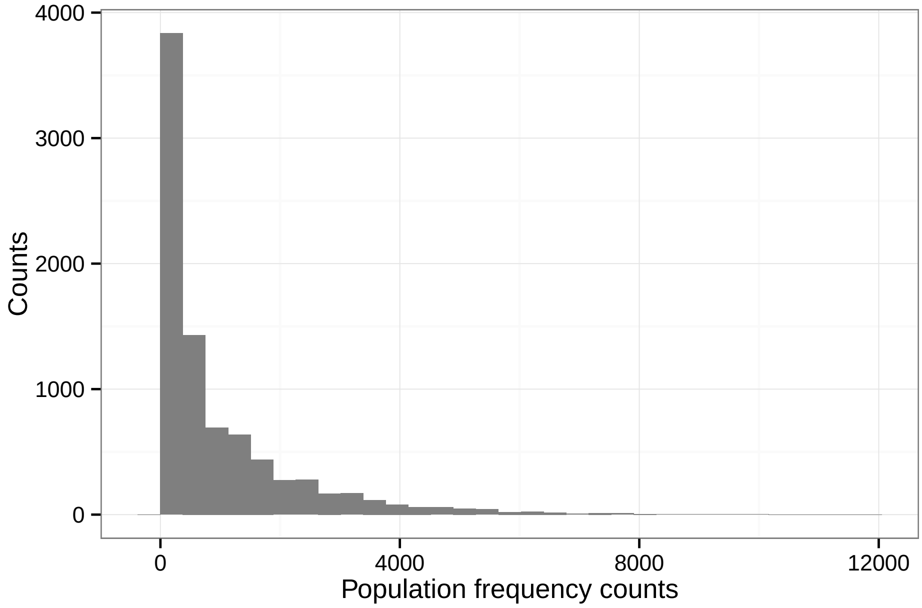


Figure 7: Histogram of population frequency counts of disclosure risk scenario 1.

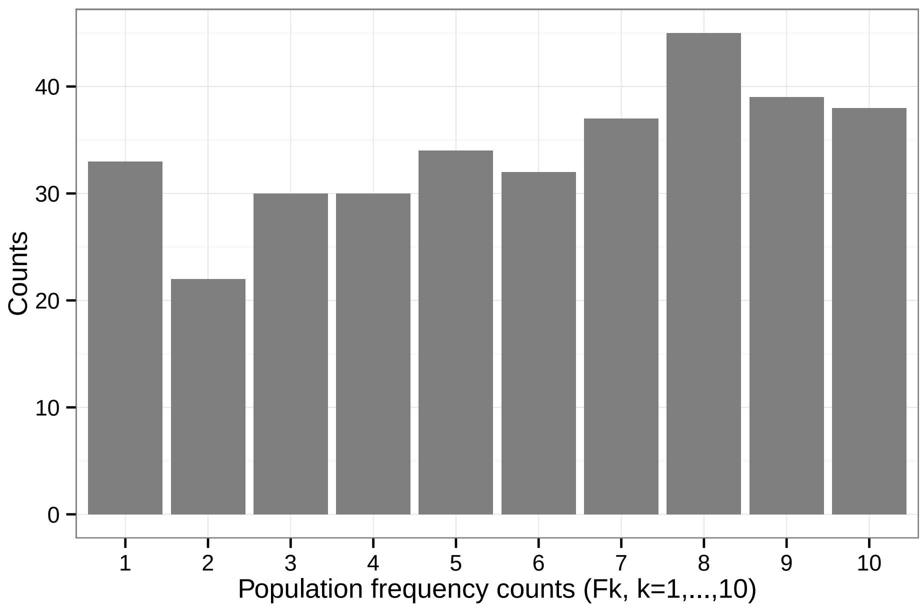


Figure 8: Histogram of population frequency counts of disclosure risk scenario 1, with Fk < 11.

11 and 12 describe the results of disclosure risk scenario 1.

Results for τ_1:

First the real (τ_1) and estimated number of sample uniques that are population unique ($\hat{\tau}_{S1}, \hat{\tau}_{EC1}, \hat{\tau}_{PSE1}, \hat{\tau}_{W1}$) are considered, see Figure 9 and 11. There are nearly the same results for SRS, ESS and PSS. The real risk τ_1 is almost in every drawn sample zero with a few exceptions, where $\tau_1 = 1$ respectively $\tau_1 = 2$. For these three sampling designs (SRS, ESS, PSS) the estimated risk measures of all considered models (standard, EC, PSE and weighted log-linear model) are close to 0. Figure 11 shows that the estimates with ESS, PSS and SRS underestimate the risk if $\tau_1 > 0$. If $\tau_1 = 0$ then $\hat{\tau}_{S1}, \hat{\tau}_{EC1}, \hat{\tau}_{PSE1}$ and $\hat{\tau}_{W1}$ are well estimated. Disclosure risk scenario 1 yields for SRS, ESS and PSS workable solutions, but the models are not resistant for samples with one or more sample uniques that are population unique. The POV design yields different results, see Figure 9. τ_1 is mostly overestimated, whereby the standard, EC and weighted log-linear model yields good results, except for outliers (i.e. $\tau_1 = 1$). The PSE model $\hat{\tau}_{PSE1}$ overestimates τ_1 between zero and seven. Thus the PSE model yields the worst results, because the overestimation is higher and a few risks are underestimated, too. The difference between the estimates and the real risk looks equal to the number of sample uniques that are population unique, but the underestimated outliers are clear to see ($\eta_{model} < 0$), as shown in Figure 11.

Results for τ_2:

The real τ_2 and estimated number of correct matches for sample uniques $\hat{\tau}_{S2}, \hat{\tau}_{EC2}, \hat{\tau}_{PSE2}$ and $\hat{\tau}_{W2}$ of disclosure risk scenario 1 yields other results than risk measure τ_1 and its assiociated estimates, see Figure 10 and 12. It's clear by definition that $\tau_2 > \tau_1$, whereby the number is not high as well. Because the disclosure risk scenario 1 has less keys and population uniques. PSS and SRS have nearly the same results for all estimates, see Figure 10. The estimated numbers of correct matches for sample uniques is underestimated with sampling designs PSS and SRS in all models, whereby the weighted log-linear model is sligthly worse, see Figure 12. ESS yields nearly the same results for the standard, EC and weighted log-linear model, but the PSE model gets other results. $\hat{\tau}_{PSE2}$ yields very good estimates for ESS, whereby the risk is a little bit overestimated. It is clear to see that $\hat{\tau}_{PSE2}$ is the best estimate with ESS, see Figure 12. The sampling design POV yields other results, too. Every model overestimates the risk, whereby the estimates of the standard, EC and weighted log-linear model are usable. The PSE model $\hat{\tau}_{PSE2}$

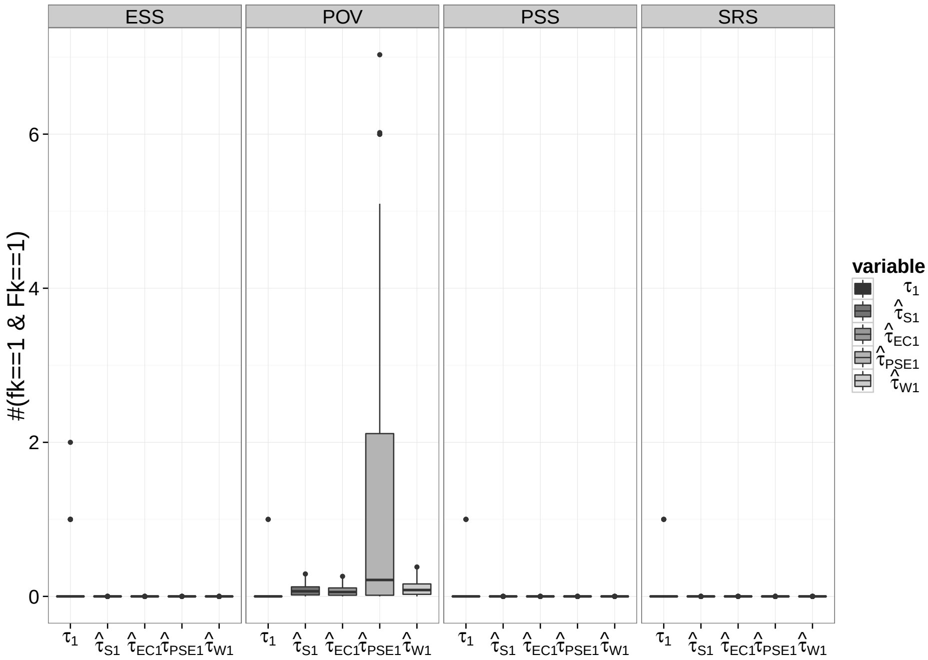


Figure 9: Real and estimated number of sample uniques that are population unique using disclosure risk scenario 1.

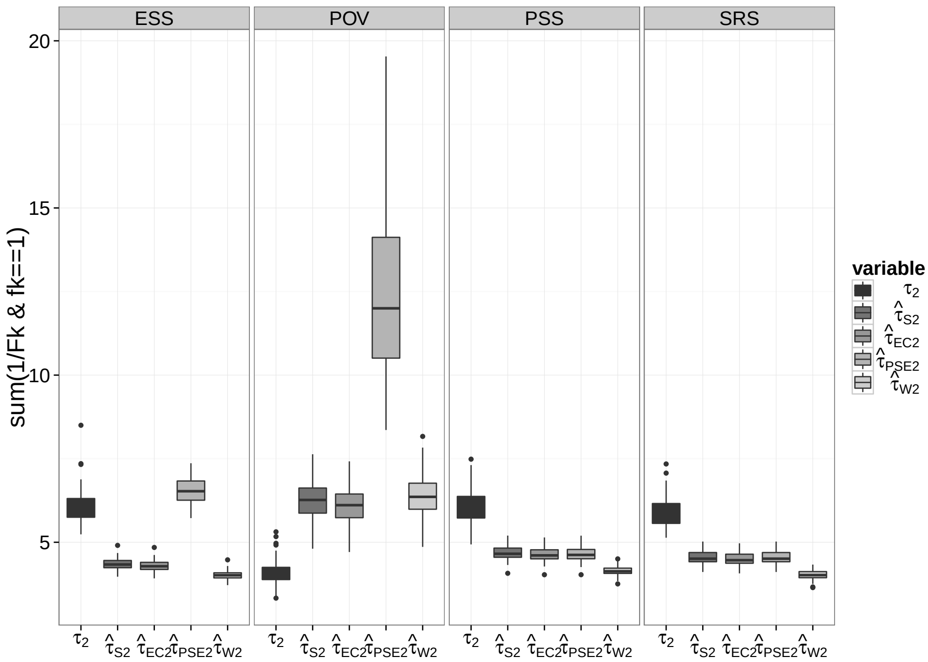


Figure 10: Real and estimated number of correct matches for sample uniques using disclosure risk scenario 1.

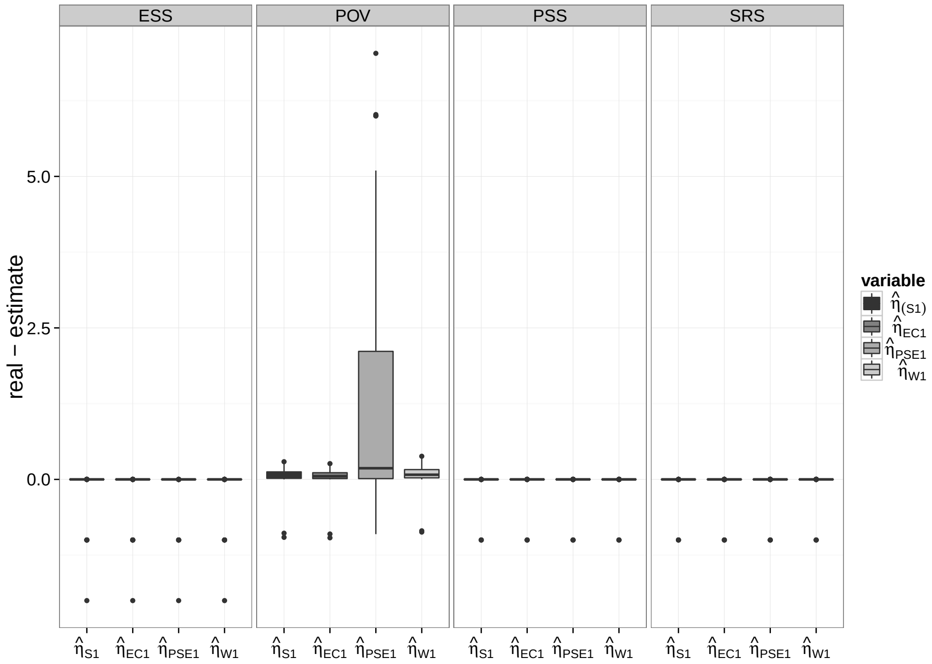


Figure 11: Difference between the real and estimated number of sample uniques that are population unique of disclosure risk scenario 1.

completely overestimates the risk (τ_2). PSE yields the best estimate with ESS and the worst with POV.

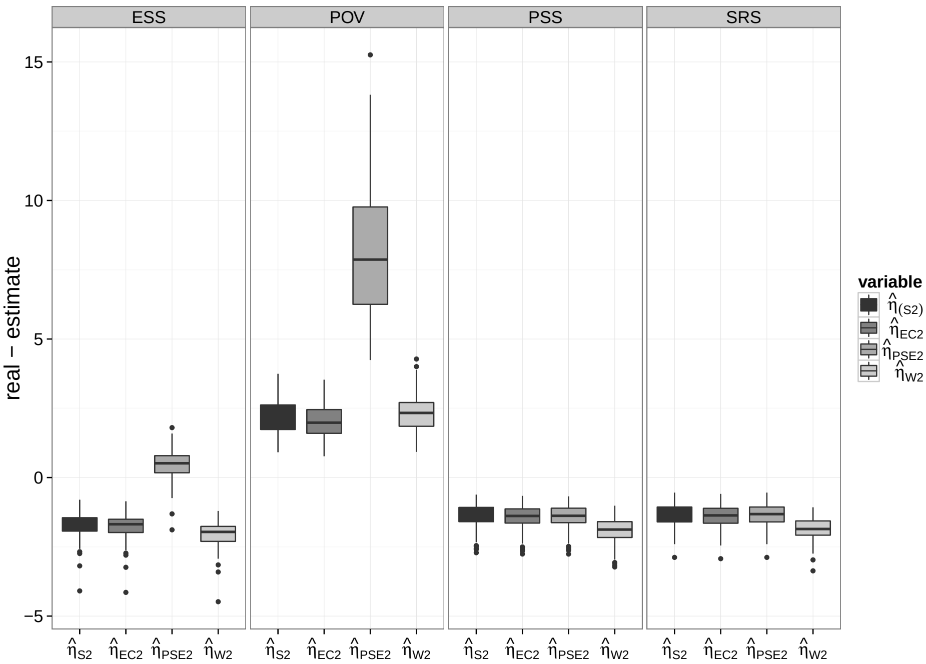


Figure 12: Difference between the real and estimated number of correct matches for sample uniques of disclosure risk scenario 1.

4.2.2 Disclosure risk scenario 2

The disclosure risk is estimated for seven categorical key variables, see Table 14. These seven variables divide the population into 138158 keys, which is a much higher amount than in `scenario 1` (8448 keys). Figure 13 shows a histogram of the population frequency counts. Most of the keys have few counts and only a few keys have high frequency counts. There are 24819 unique persons in the population and 60326 cells with less than 6 records. The population consists of 8182010 persons, which means that 0.30334 % of the persons are population unique. It is clear to see that there are much more keys with few counts than with `disclosure scenario 1`, compare Figure 14 and 8. To fit the log-linear models (standard, EC, PSE, weighted) the following formulas are applied:

```
R > keyVars_S2 <- c("db040","hsize","rb090", "age","pl030", "pb220a", "netIncomeCat")
R > f <- as.formula(paste(" ~ ", "netIncomeCat:rb090 + netIncomeCat:age + age:pl030 +
+                         db040 + hsize + rb090 + age + pl030 + pb220a + netIncomeCat"))
R > (f_standard_llm <- as.formula(paste(c("counts", as.character(f)),
+                         collapse = "")))

counts ~ netIncomeCat:rb090 + netIncomeCat:age + age:pl030 +
    db040 + hsize + rb090 + age + pl030 + pb220a + netIncomeCat

R > (f_pse_llm <- as.formula(paste(c("estimated_Fk", as.character(f)),
+                         collapse = "")))

estimated_Fk ~ netIncomeCat:rb090 + netIncomeCat:age + age:pl030 +
    db040 + hsize + rb090 + age + pl030 + pb220a + netIncomeCat

R > (f_weighted_llm <- as.formula(paste(c("counts",
+                       as.character(as.formula(paste(c(f,"weights"),collapse="+")))),
+                               collapse = "")))

counts ~ netIncomeCat:rb090 + netIncomeCat:age + age:pl030 +
    db040 + hsize + rb090 + age + pl030 + pb220a + netIncomeCat +
    weights
```

Variable `keyVars_S2` consists of the seven categorical key variables as decribed in Table 14. It might exist a better interaction model, because the `glm()` function cannot solve a more complex model. For more keys or complex models another function has to be implemented that handles such problems. But the predictors are quite good for this scenario. `f_standard_llm, f_pse_llm and f_weighted_llm` describe the model to be fitted. For the EC approach the formula `f_standard_llm` is used and the offset term in function `glm()` is set to offset $= \frac{f_k}{\hat{F}_k}$, which is also described in `scenario 1`. These four log-linear models are calculated for 100 sample runs. There are about 50 per cent unique records in each sample via srs, pss and ess as well as 30 per cent with proportinal oversampling. A high risk for τ_1 and τ_2 is expected, because the amount of sample uniques is very high. Figures 15, 16, 17 and 18 describe the results of `disclosure risk scenario 2`.

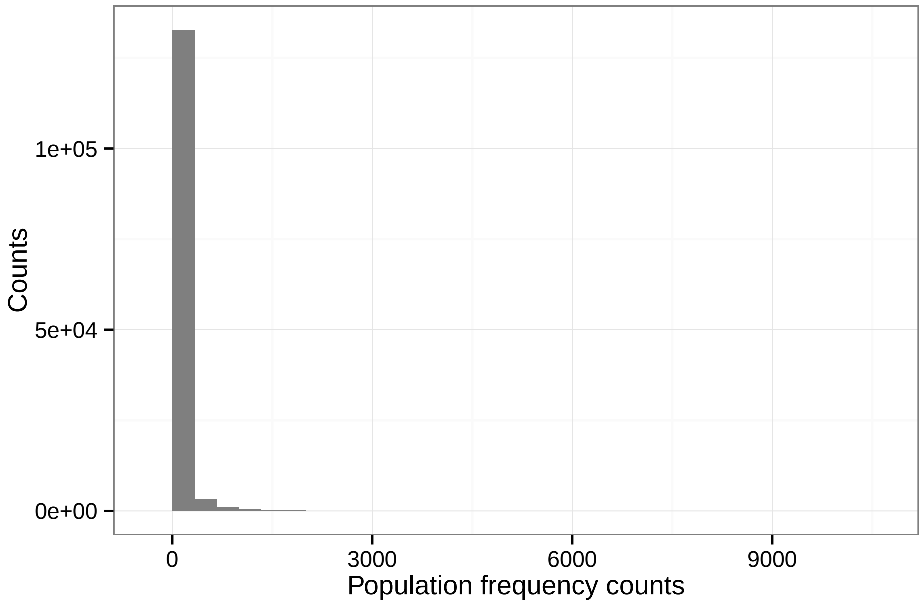


Figure 13: Histogram of population frequency counts of disclosure risk scenario 2.

Results for τ_1:

First the results of the real (τ_1) and estimated number of sample uniques that are population unique ($\hat{\tau}_{S1}, \hat{\tau}_{EC1}, \hat{\tau}_{PSE1}, \hat{\tau}_{W1}$) are considered, see Figure 15 and 17. There are nearly the same results for SRS and PSS. The arithmetic mean of τ_1 is 32.39 with SRS and 33.11 with PSS. The standard, EC and PSE log-linear models yield nearly the same results for SRS and PSS, whereby the risk is ligthly underestimated. The weighted log-linear approach yields a worse estimate in this case, because τ_1 is completly underestimated. Equal stratified sampling yields very good results with the PSE log-linear model. Whereby the standard, EC and weighted log-linear model underestimates the risk. It is clear to see that the weighted log-linear model

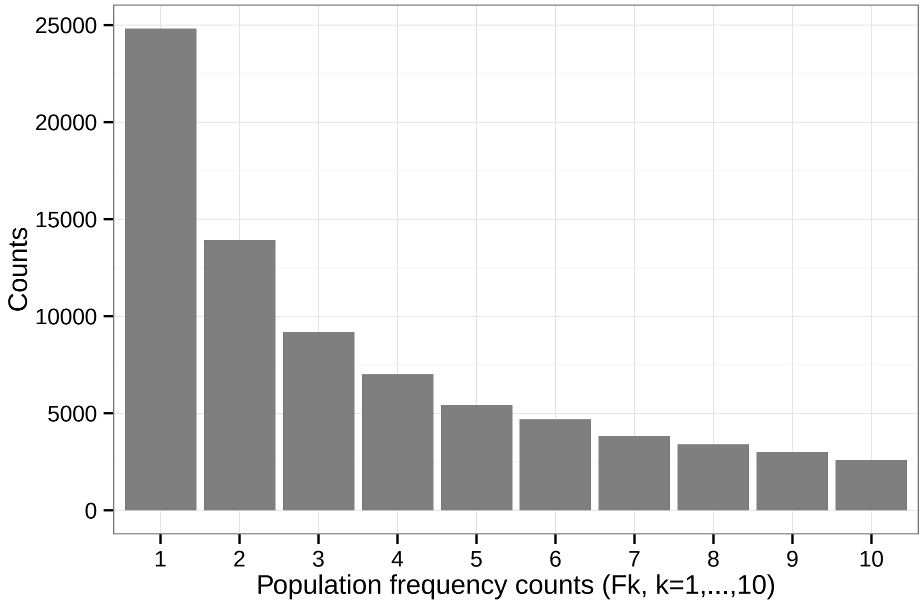


Figure 14: Histogram of population frequency counts of disclosure risk scenario 2, with Fk < 11.

performs worst with ESS. The POV design yields other results in comparison to SRS, ESS and PSS, see Figure 15. τ_1 is overestimated in each model. The standard, EC and weighted log-linear model yield good results, except for some samples in the weigthed log-linear model. The PSE approach yields massive overestimated results, because the estimated number of correct matches for sample uniques that are population unique is about six times higher than the real number. Figure 17 shows that the difference between the real and estimated risk is in eleven cases very close to zero and the worse models are clear to see, like the weighted log-linear model in three cases and the PSE model with POV.

Results for τ_2:

The real τ_2 and estimated number of correct matches for sample uniques $\hat{\tau}_{S2}, \hat{\tau}_{EC2}, \hat{\tau}_{PSE2}$ and $\hat{\tau}_{W2}$ of disclosure risk scenario 2 yields some other results than risk measure τ_1 and its assiociated estimates, see Figure 16 and 18. The amount of the real and expected number of correct matches for sample uniques is very high in all sampling designs, because the amount of sample uniques is very high in every sample. PSS and SRS yields nearly the same results for all estimates. The standard, EC and PSE model overestimates the risk τ_2, but the results are quite good. The weighted log-linear model underestimates the risk and the estimates are worse. The ESS design yields very good results for the standard and EC model. But not useful results

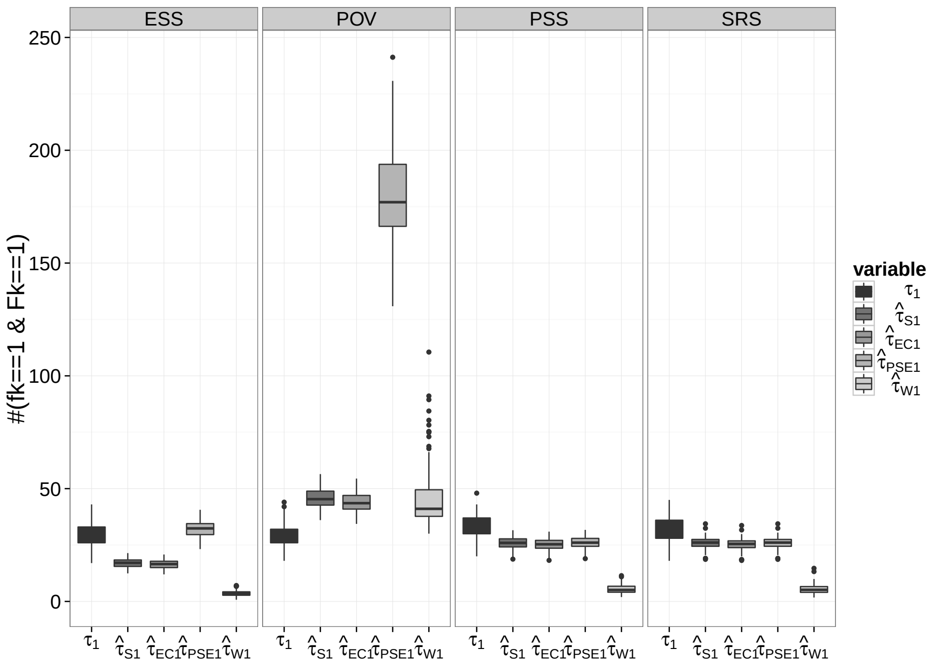


Figure 15: Boxplot of real and expected number of sample uniques that are population unique.

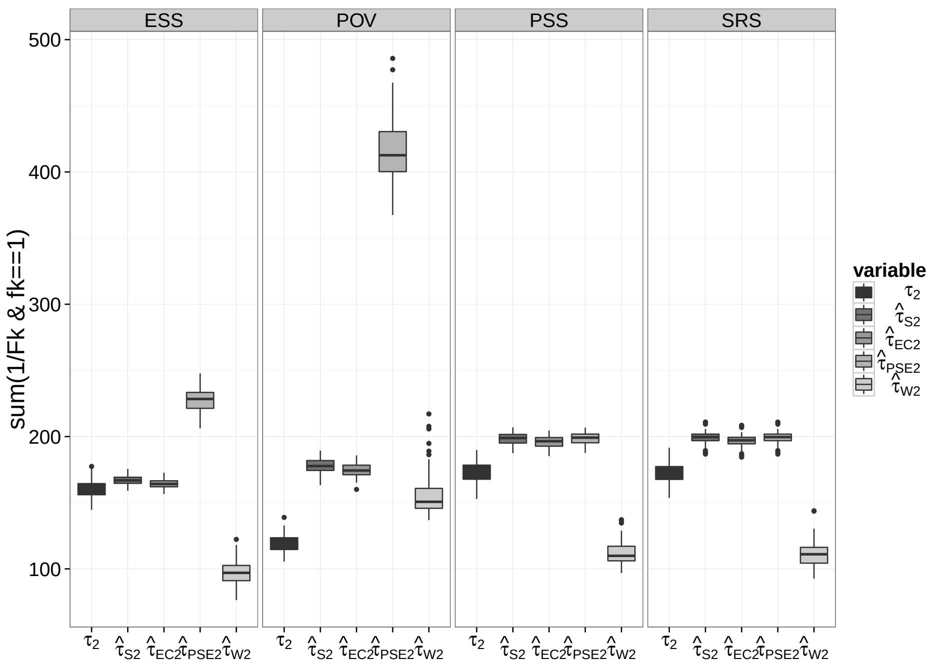


Figure 16: Boxplot of real and expected number of correct matches for sample uniques.

with the PSE and weighted log-linear approach, whereby PSE overestimates the risk and the weighted underestimates it. The sampling design POV yields other results again. Every model overestimates the risk, whereby the estimates of the standard, EC and weighted log-linear model are usable. The PSE model $\hat{\tau}_{PSE2}$ completely overestimates the risk (τ_2), which is clear to see in Figure 16. The standard and EC log-linear model yield workable results in every sampling design.

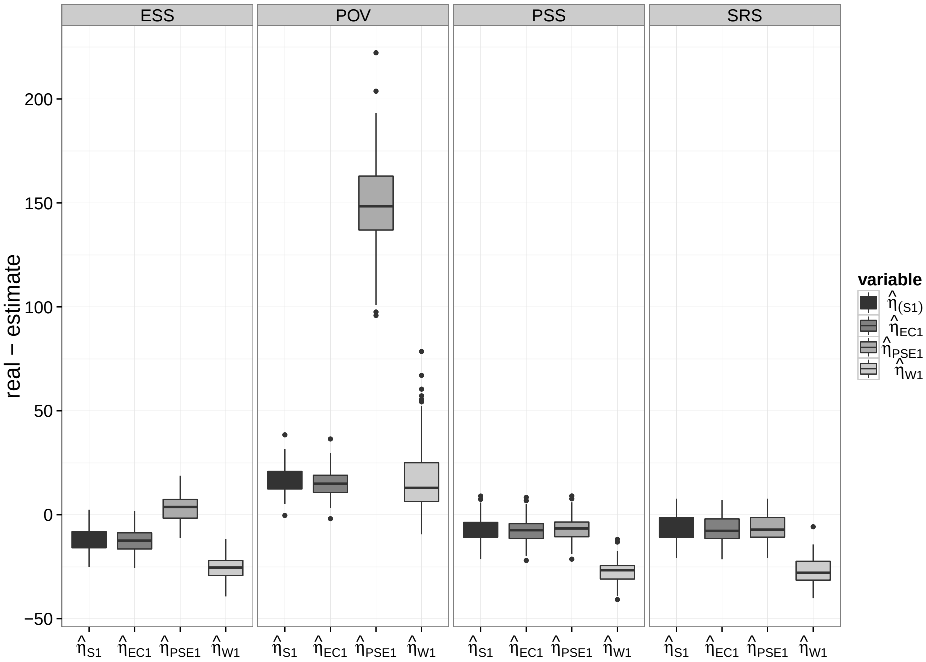


Figure 17: Boxplot of the difference between real and expected number of sample uniques that are population unique.

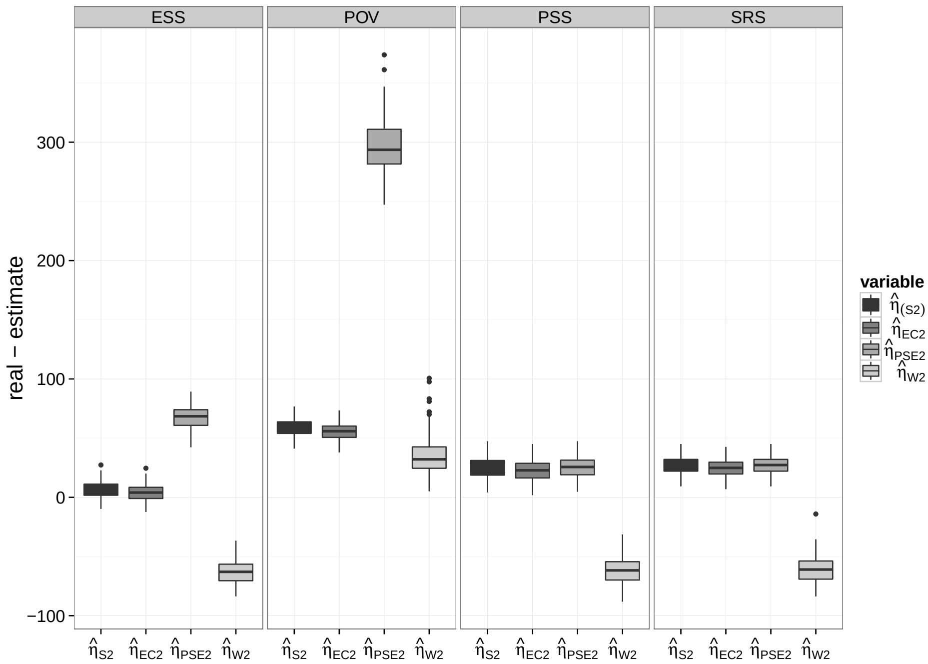


Figure 18: Boxplot of the difference between real and expected number of correct matches for sample uniques.

4.2.3 Disclosure risk scenario 3

`Disclosure risk scenario 3` estimates the risk for five categorical key variables, see Table 14. These seven variables divide the population into 32414 keys, which is more than in `scenario 1` (8448 keys). However, the amount of keys is lower as within `scenario 3` (138158 keys). Figure 19 shows a histogram of the population frequency counts, which looks similar to the population frequency counts of `scenario 2`, see Figure 13. Many keys with only few counts and only a few keys with large frequency counts exists. There are 2607 unique persons in the population and 7598 keys with less than 6 records. Since the population consists of 8182010 persons, 0.03186 % of persons are population unique. The population frequncy count structure of `Scenario 3` is somehow between `scenario 1` and `scenario 2`, compare Figures 20, 14 and 8. To fit the standard, EC, PSE and weighted log-linear models the following formulas are applied:

```
R > keyVars_S3 <- c("hsize","rb090", "age", "pl030", "netIncomeCat")
R > f <- as.formula(paste(" ~ ", "hsize + rb090 + age +
+                               pl030 + netIncomeCat + age:rb090 +
+                               age:hsize + netIncomeCat:age + rb090:age"))
R > (f_standard_llm <- as.formula(paste(c("counts", as.character(f)),
+                               collapse = "")))

counts ~ hsize + rb090 + age + pl030 + netIncomeCat + age:rb090 +
    age:hsize + netIncomeCat:age + rb090:age

R > (f_pse_llm <- as.formula(paste(c("estimated_Fk", as.character(f)),
+                               collapse = "")))

estimated_Fk ~ hsize + rb090 + age + pl030 + netIncomeCat + age:rb090 +
    age:hsize + netIncomeCat:age + rb090:age

R > (f_weighted_llm <- as.formula(paste(c("counts",
+                     as.character(as.formula(paste(c(f,"weights"),
+                               collapse="+")))), collapse = "")))

counts ~ hsize + rb090 + age + pl030 + netIncomeCat + age:rb090 +
    age:hsize + netIncomeCat:age + rb090:age + weights
```

Variable `keyVars_S3` consists of the five categorical key variables as decribed in Table 14. This 2-way intercation model performs very good for `disclosure risk scenario 3`. `f_standard_llm, f_pse_llm and f_weighted_llm` describe the model to be fitted. For the EC approach the formula `f_standard_llm` is used and the offset term in function `glm()` equals $\frac{f_k}{\hat{F}_k}$, which is also described in `scenario 1`. For every of the 100 drawn samples for each sampling design the standard, EC, PSE and weighted log-linear model is estimated. There are about 25 per cent unique records in each sample with SRS, PSS and ESS as well as 12 per cent with proportinal oversampling. The risk measures (τ_1 and τ_2) should be between the measures of `scenario 1` and `scenario 2`, because the amount of sample uniques is between these two scenarios. Figures 21, 22, 23 and 24 describe the results of `disclosure risk scenario 3`.

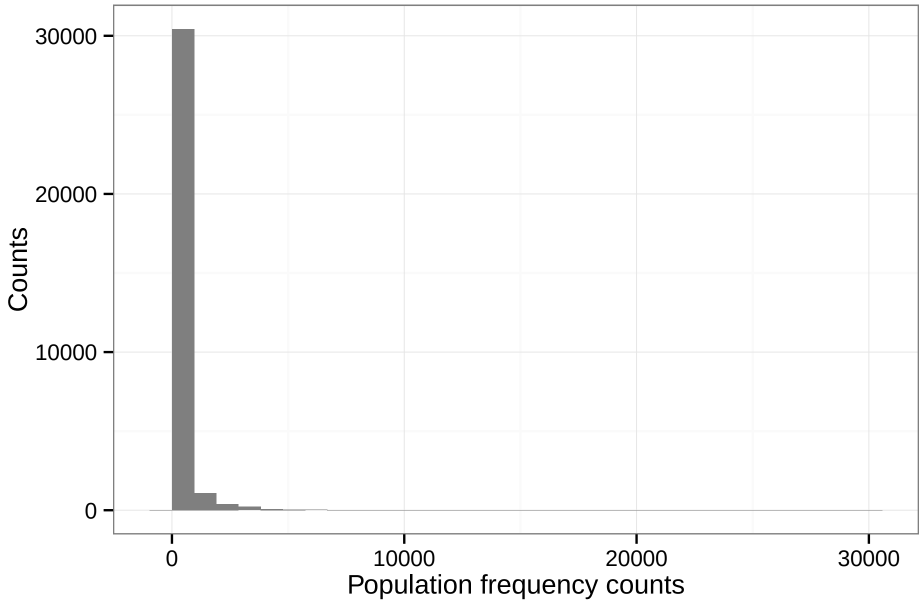


Figure 19: Histogram of population frequency counts of disclosure risk scenario 3.

Results for τ_1:

The estimated number of sample uniques that are population unique ($\hat{\tau}_{S1}, \hat{\tau}_{EC1}, \hat{\tau}_{PSE1}, \hat{\tau}_{W1}$) is considered, see Figure 21. Figure 23 shows the difference between the real and estimated number of sample uniques. If the difference between $\hat{\tau}_{model1} - \tau_1 < 0$ then the risk is underestimated. If the difference is close to zero the estimates are good. Figure 21 shows that there are nearly the same results for ESS, PSS and SRS. Depending on the sample drawn from the population, the risk (= amount of sample uniques that are population unique) is between $[0, 8]$, but the estimates of all four log-linear models are about 1, with small variances. Every log-linear model underestimates the risk with ESS, PSS and SRS. All models yield nearly the same results, see Figure 23. The proportional oversampling (POV) design yields different results, which is also

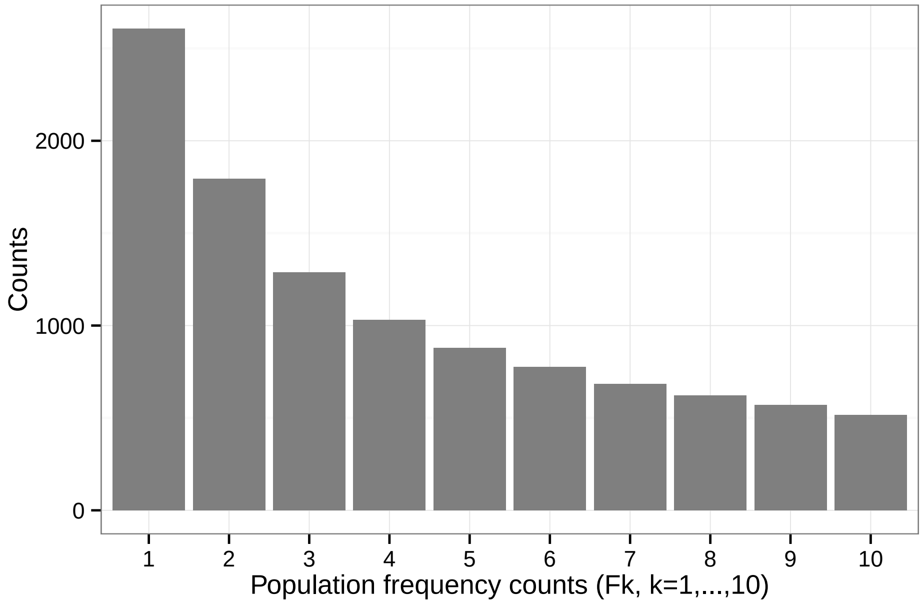


Figure 20: Histogram of population frequency counts of disclosure risk scenario 3, with Fk $<$ 11.

the case in `scenario` 1 and 2, see Figures 9, 15 and 21. τ_1 is overestimated, whereby the standard, EC and weighted log-linear model yield workable results. The weighted log-linear model performs better than the standard and EC approach. The PSE model $\hat{\tau}_{PSE1}$ completely overestimates τ_1. Thus the PSE model yields the worst results with POV.

Results for τ_2:

The real τ_2 and estimated number of correct matches for sample uniques $\hat{\tau}_{S2}, \hat{\tau}_{EC2}, \hat{\tau}_{PSE2}$ and $\hat{\tau}_{W2}$ of `disclosure risk scenario 3` shows other results than the above described estimates. Whereby PSS and SRS yield almost the same results for all models and the risk measure τ_2 is underestimated. The weighted log-linear model is worse than the standard, EC and PSE model, see Figure 22 and 24. The estimates with equal stratified sampling are very good and much better than with PSS and SRS. The standard, EC and PSE log-linear model slightly overestimate the risk and the weighted log-linear model yields a light underestimation. There are complete other results with POV. Every model overestimates the risk and none of the models yields useful results. The PSE model $\hat{\tau}_{PSE2}$ performs worst.

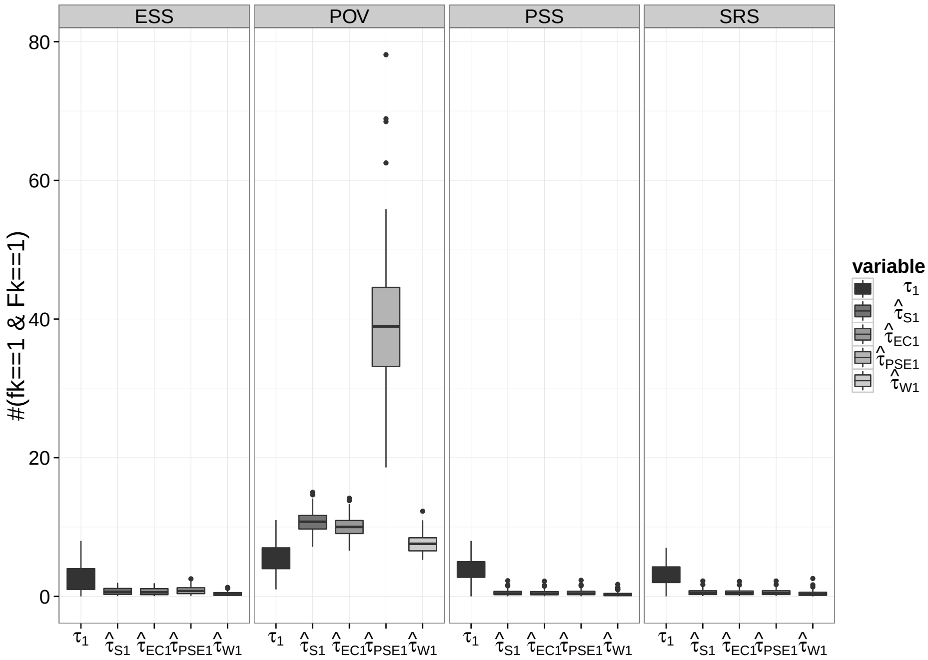


Figure 21: Boxplot of real and expected number of sample uniques that are population unique using scenario 3.

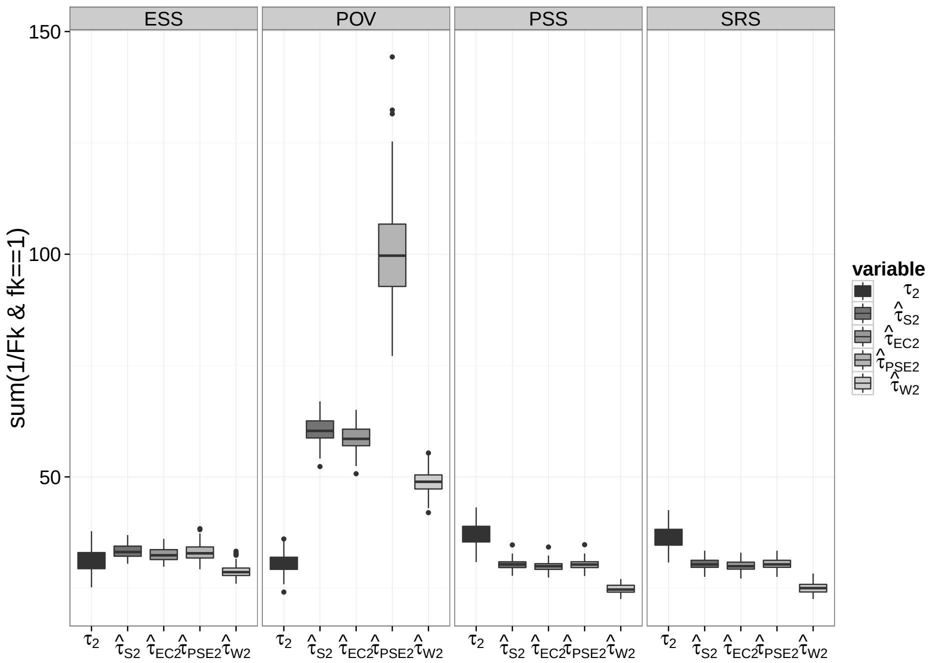


Figure 22: Boxplot of real and expected number of correct matches for sample uniques using scenario 3.

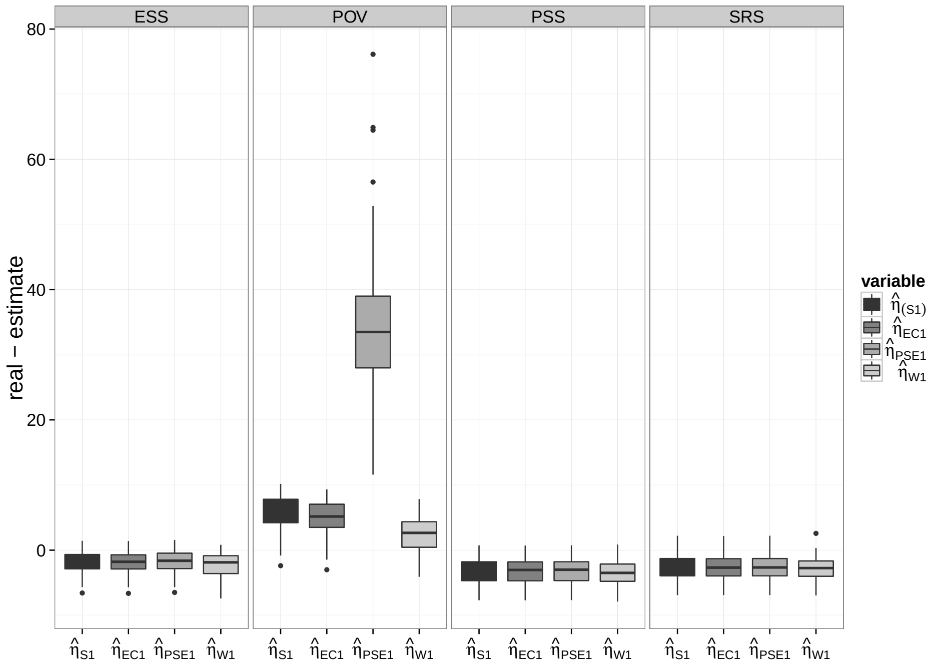


Figure 23: Boxplot of the difference between real and expected number of sample uniques that are population unique of scenario 3.

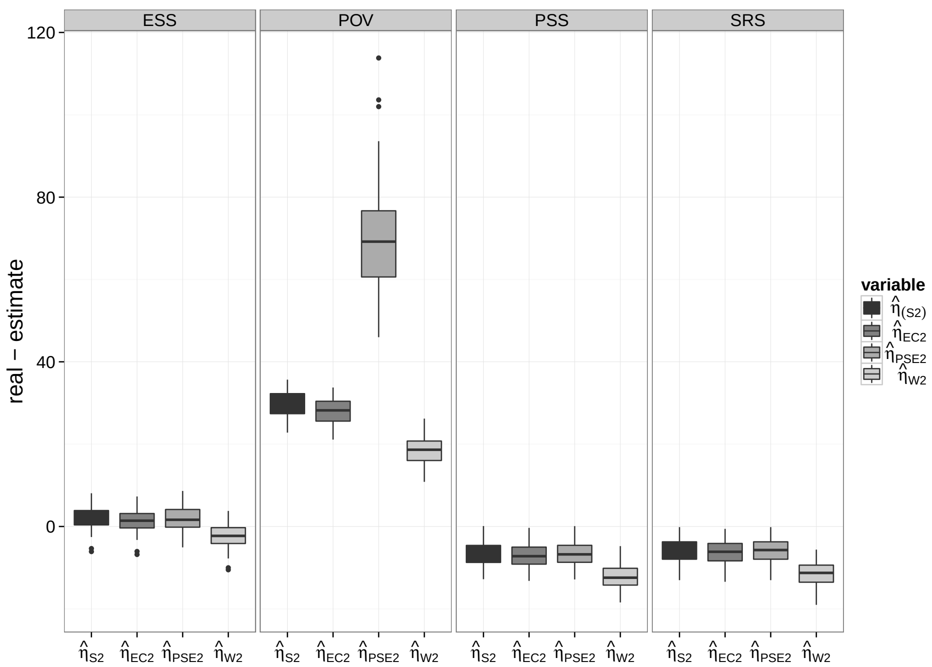


Figure 24: Boxplot of the difference between real and expected number of correct matches for sample uniques of scenario 3.

4.2.4 Scenario comparison

The following figures compare the three scenarios for each sampling design (equal stratified sampling, proportional oversampling, proportional stratified sampling, simple random sampling) and risk measures τ_1 and τ_2 (see Section 3.1).

Simple random Sampling (SRS):

Figure 25 shows the real and estimated number of sample uniques that are population unique. It is clear to see that the risk is higher if the amount of keys is higher. τ_1 is underestimated with each kind of log-linear models, whereby the weighted log-linear model performs worst (see Figure 25 and `scenario 2`). Figure 26 shows the estimation of the number of correct matches for sample uniques τ_2. It is clear to see that the weighted log-linear model performs worst in every scenario. The other models yield nearly the same results, whereby the risk is underestimated in `scenario` 1 and 3 and overestimated in `scenario 2`, which has the most keys and population uniques.

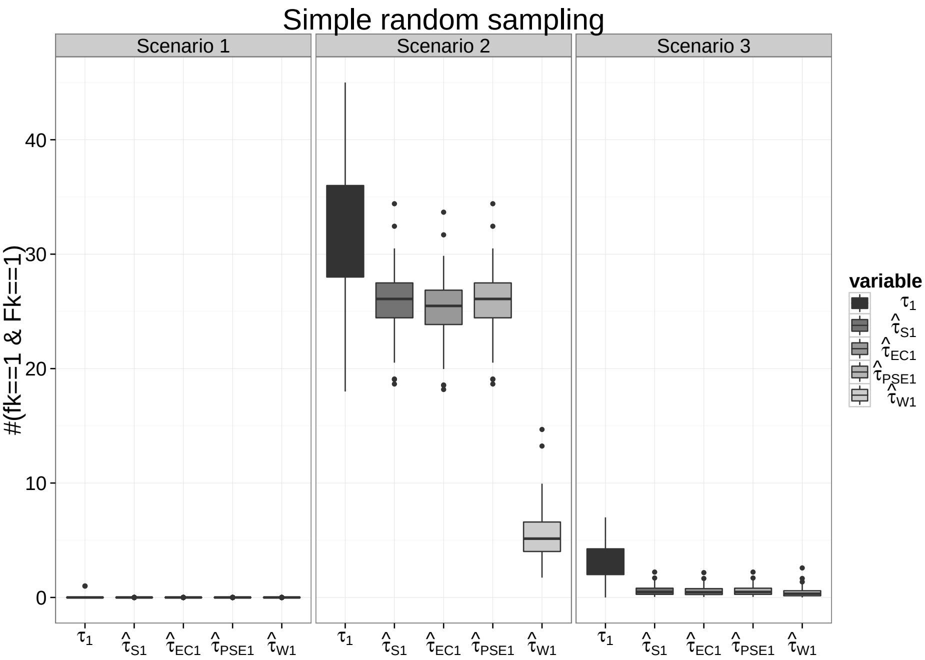


Figure 25: Scenario comparison of τ_1 with SRS.

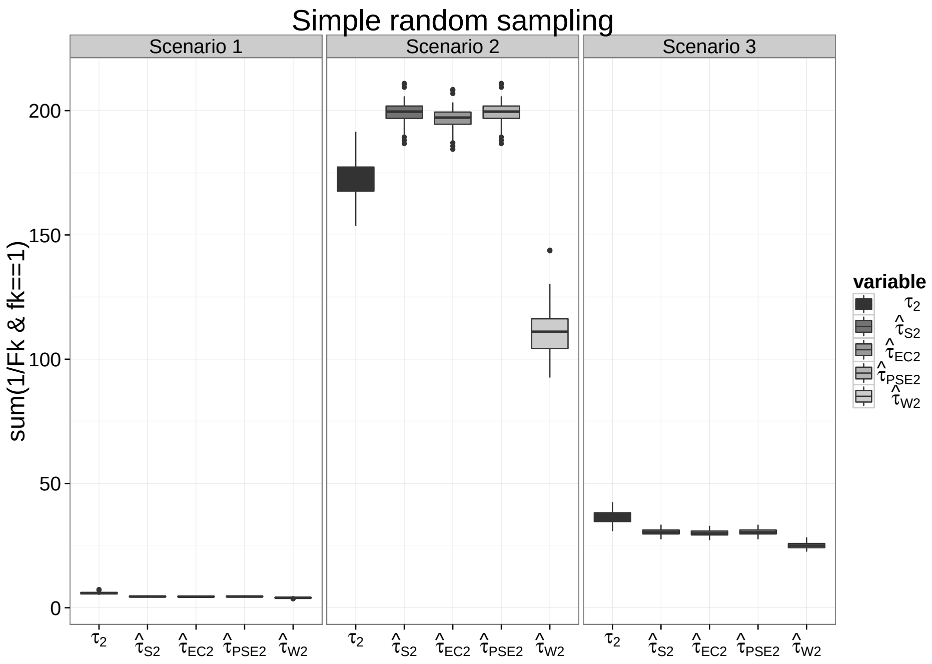


Figure 26: Scenario comparison of τ_2 with SRS.

Equal stratified sampling (ESS):

Figure 27 describes the estimation of τ_1. The risk is underestimated except the PSE estimator in `scenario 2`, whereby the weighted log-linear model performs worst. The boxplots in Figure 28 show nearly perfect estimates for `scenario` 1 and 3. `Scenario` 2 yields very good estimates with the standard and EC model. The weighted log-linear model underestimates the risk measure and the PSE model overestimates it.

Proportional stratified sampling (PSS):

The number of sample uniques that are population unique (τ_1) is underestimated in all cases, whereby the estimates in `scenario` 1 are only underestimated if $\tau_1 = 1$, see Figure 29. `Scenario` 2 shows that the weighted log-linear model performs worst, which is also the case with ESS and SRS. All the other models yield nearly the same estimates with proportional stratified sampling. Figure 30 shows the real and estimated numbers of correct matches for sample uniques. The estimates in `scenario` 1 and 3 are a bit underestimated. τ_2 is overestimated via the standard, EC and PSE model in `scenario` 2. The weighted log-linear model yields the worst results, as

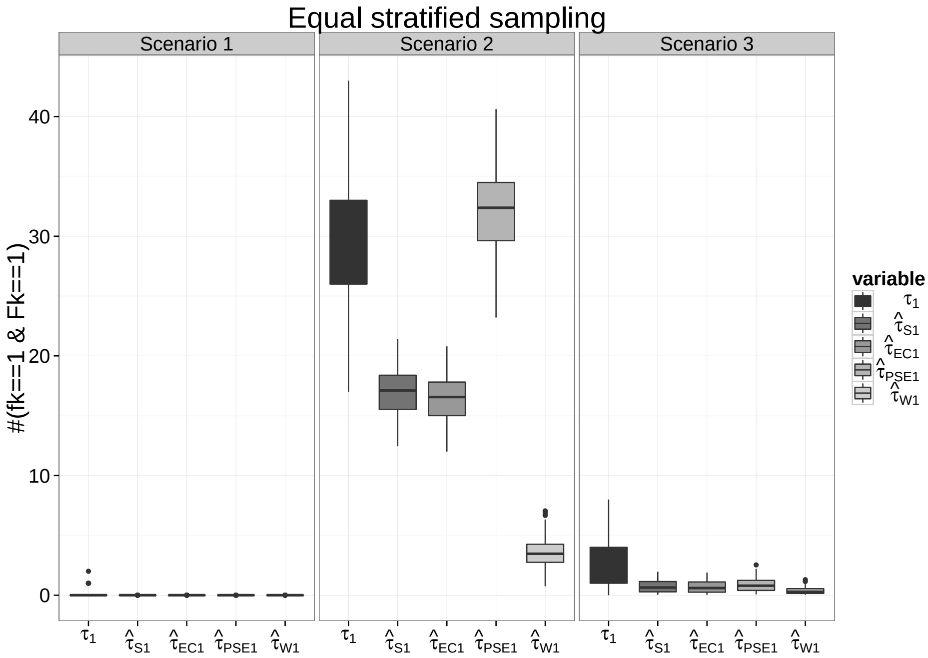


Figure 27: Scenario comparison of τ_1 with ESS.

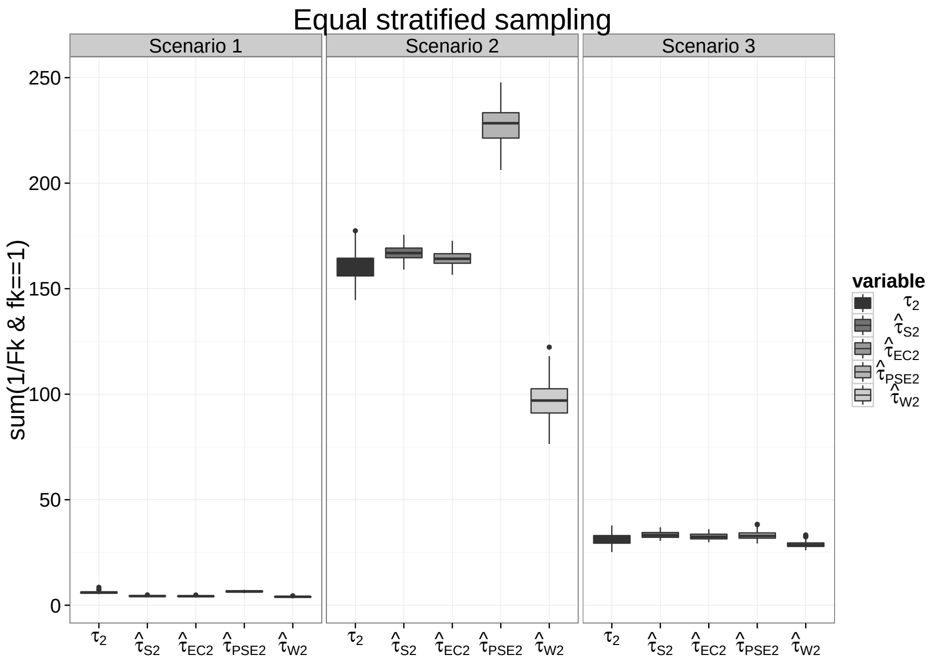


Figure 28: Scenario comparison of τ_2 with ESS.

also can be seen with SRS and ESS in Figures 26 and 28.

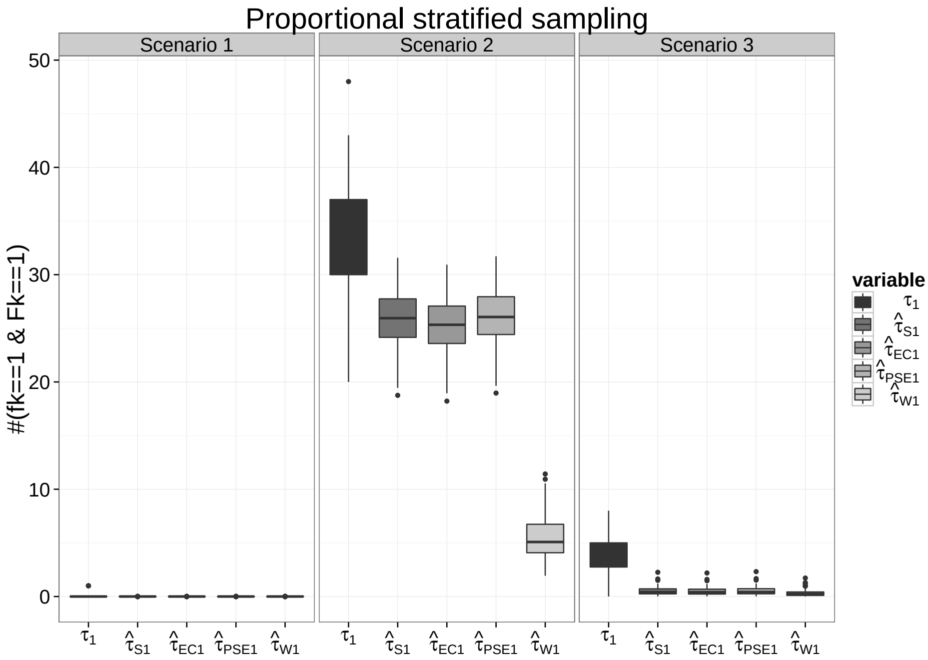


Figure 29: Scenario comparison of τ_1 with PSS.

Proportional oversampling (POV):

Proportional oversampling yields quite other results. Figure 31 shows that τ_1 is overestimated in nearly all cases, whereby the pseudo maximum likelihood (PSE) method performs worst. The other models yield usable results in each scenario. The estimated numbers of correct matches for sample uniques is overestimated, whereby the PSE model completly overestimates the risk.

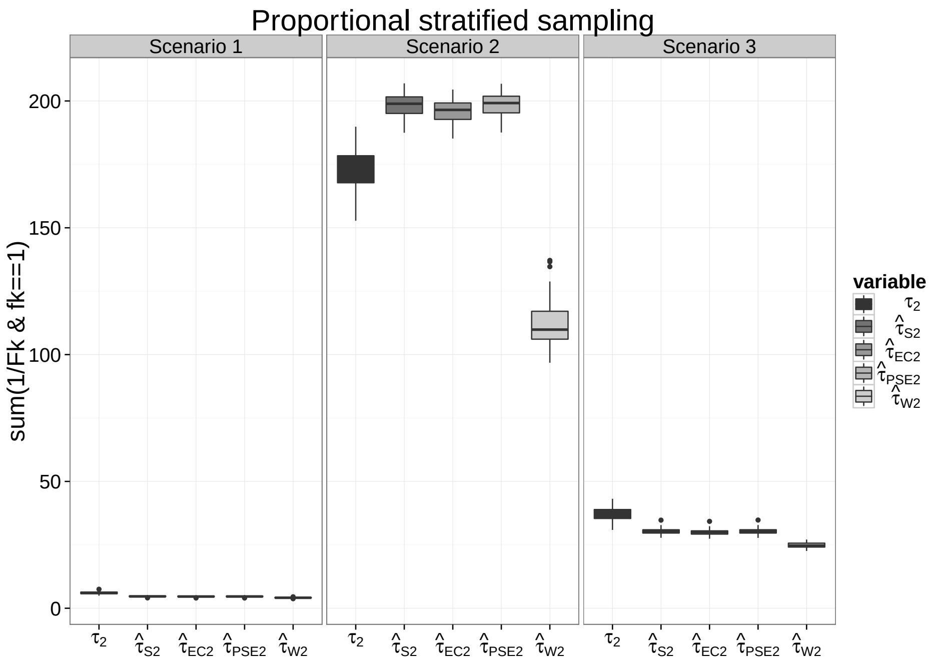


Figure 30: Scenario comparison of τ_2 with PSS.

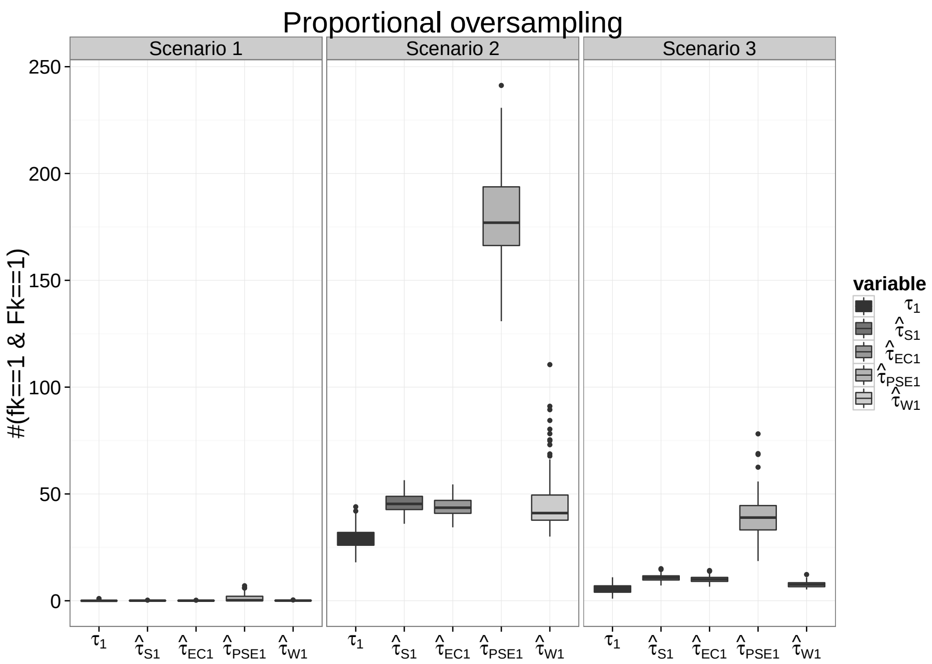


Figure 31: Scenario comparison of τ_1 with POV.

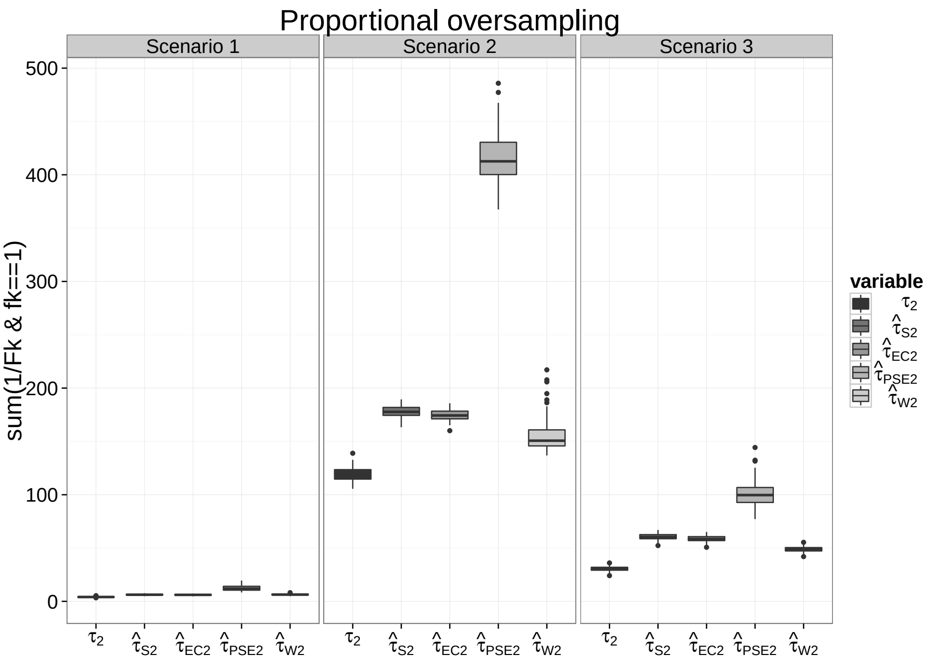


Figure 32: Scenario comparison of τ_2 with POV.

5 Conclusion

The use of Poisson log-linear models to estimate the number of sample uniques that are population unique (τ_1) and the number of correct matches for sample uniques (τ_2) of microdata are considered based on synthetic population data from Austria. The models are tested with four different sampling designs (equal stratified sampling, proportional oversampling, proportional stratified sampling, simple random sampling) and three disclosure risk scenarios with different amounts of keys (see Table 14). Two global risk measures of interest are considered (τ_1 and τ_2, see Section 3.1). Skinner and Vallet [2010] and Carlson [2002b] investigated lower population sizes ($N = 40000$ till $N = 268000$) and higher sample fractions (includes low values of the sampling weights), which results in simplified investigations of the disclosure risk. This master thesis goes beyond the empirical comparisons of the mentioned articles. The risk measures are estimated with realistic sizes of a population ($N = 8182010$) and a sample ($n \approx 12000$) by using different designs to draw the samples.

First the estimates of the number of sample uniques that are population unique (τ_1) are considered. All models underestimate the real value of τ_1 in each scenario with respect to three sampling designs (simple random sampling, equal stratified sampling and proportional stratified sampling). The estimates are underestimated because the fitted values of the Poisson log-linear models are too small. One reason can be that the interaction models are not perfectly chosen and the other reason can be that the number of keys are too less. It should be mentioned that `scenario 1` indicates a very low disclosure risk and it is therefore difficult to make statements about the quality of the estimates. There is only one exception with equal stratified sampling and the pseudo maximum likelihood model in `scenario 2`, which yields good estimates and no significant underestimation. All in all, the pseudo maximum likelihood method seems to perform best with simple random, equal stratified and proportional stratified sampling, because the response variable $log(\hat{F}_k)$ yields better estimates, where $\hat{F}_k$ is the sum of survey weights across sample units in key $k \in \{1, 2, ...C\}$. τ_1 is underestimated because the λ_j, $\{j : f_j = 1\}$ are overestimated and τ_1 is estimated by $\hat{\tau}_{1model} = \sum_{\{j:f_j=1\}} e^{-\lambda_j(1-\pi_j)}$. The weighted log-linear model performs worst with simple random sampling (SRS), equal stratified sampling (ESS) and proportional stratified sampling (PSS), because the weights are not correlated with the response variable in every sampling design and so the interaction model is wrongly chosen. Proportional oversampling (POV) yields other results, because the terms $e^{-\lambda_j(1-\pi_j)}$ are overestimated, which depends on the inclusion probabilities. The pseudo maximum likelihood method completly overestimates the risk, because the model cannot handle the specific inclusion probabilities.

For the number of correct matches of sample uniques (τ_2), which is estimated by $\hat{\tau}_{2model} = \sum_{\{j:f_j=1\}} \frac{1-e^{-\lambda_j(1-\pi_j)}}{\lambda_j(1-\pi_j)}$, the estimates with simple random sampling (SRS) and proportional stratified sampling (PSS) are similar, whereby the risk is underestimated with `scenario` 1 and 3 and overestimated with `scenario 2`. One reason is that the interaction models are not perfectly chosen. The underestimation can result of a too low number of keys (C) in `scenario` 1 and 3, because the risk measures are consistent. The equal stratified sampling design (ESS) yields a good performance with the standard and Eliason-Clogg method. The pseudo maximum likelihood method (PSE) overestimates the risk in `scenario 2`, because the model underestimates the frequency counts λ_j, with $\{j : f_j = 1\}$. The opposite is true for the weighted log-linear model in `scenario 2`. With proportional oversampling (POV) the risk estimates of all models are overestimated, whereby the pseudo maximum likelihood method performs worst, the high inclusion probabilities intensify the underestimation of λ_j, with $\{j : f_j = 1\}$, which leads to an overestimation of τ_2.

One important point for good model performances is to choose a well-defined good interaction model (see also Shlomo and Skinner [2008]). If there are too less predictors the model is underestimated. Another criterium is the amount of keys, whereby a high amount of keys will generally give better results. All in all the standard method, the Eliason-Clogg and the pseudo maximum likelihood approach perform best and yield nearly the same results with simple random sampling (SRS), equal stratified sampling (ESS) and proportional stratified sampling (PSS). The weighted log-linear model performs worst.

For future tasks the consideration of missing values may lead to another choice of models. In this work only samples without missing values are considered. Another consideration could be the estimation of variances to investigate about the quality and uncertainty of point estimates (see discussion by Skinner and Vallet [2010]). It is also reasonable to test the models with other survey data and by using other sampling designs.

References

A. Agresti. *Categorical Data Analysis*. Wiley Series in Probability and Statistics. Wiley-Interscience, 2nd edition, 2002.

A. Alfons, M. Templ, and P. Filzmoser. *The R Package simFrame: An Object-Oriented Framework for Statistical Simulation*, 2010. URL `http://CRAN.R-project.org/package=simFrame`. R package version 0.5.3.

A. Alfons, S. Kraft, P. Filzmoser, and M. Templ. *Simulation of Close-to-Reality Population Data for Household Surveys with Application to EU-SILC*, 2011. URL `http://CRAN.R-project.org/package=simPopulation`. R package version 0.4.1.

M. Carlson. Assessing Microdata Disclosure Risk Using the Poisson-Inverse Gaussian Distribution. Technical report, 2002a.

M. Carlson. An Empirical Comparison of Some Methods for Disclosure Risk Assessment. Technical report, 2002b.

C. C. Clogg and S. R. Eliason. Some Common Problems in Log-Linear Analysis. *Sociological Methods & Research*, pages 8–44, 1987.

W. G. Cochran. *Sampling Techniques*. John Wiley & Sons, Inc., 1977. ISBN 047116240X.

D. B. Dahl. *xtable: Export tables to LaTeX or HTML*, 2014. URL `http://CRAN.R-project.org/package=xtable`. R package version 1.7-3.

P. de Wolf, J. M. Gouweleeuw, P. Kooiman, and L. Willenborg. Reflections on PRAM. Technical report, Statistics Netherlands, Department of Statistical Methods, 1998.

J. C. Gower. A General Coefficient of Similarity and Some of Its Properties. *Biometrics*, pages 623–637, 1971.

B. Gross, P. Guiblin, and K. Merrett. Implementing the Post Randomisation Method to the Individual Sample of Anonymised Records (SAR) from the 2001 Census. Technical report, Statistical Disclosure Control Centre, Methodology Group, Office for National Statistics., 2004.

A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, R. Lenz, J. Longhurst, E. Schulte Nordholt, G. Seri, and P. De Wolf. *Handbook on Statistical Disclosure Control*, 2010.

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley Series in Probability and Statistics. Wiley, 2005. ISBN 978-0-471-73578-6.

St. Lemeshow and Paul S. Levy. *Sampling of Populations: Methods and Applications*. A John Wiley & Sons, Inc., fourth edition edition, 2008. ISBN 978-0-470-04007-2.

H. Midzuno. On the Sampling System with Probability Proportional to Sum of Size. *Annals of the Institute of Statistical Mathematics*, pages 99–107, 1952.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org/`.

N. Shlomo and C. J. Skinner. Assessing Identification Risk in Survey Microdata Using Log-Linear Models. *Journal of the American Statistical Association*, pages 989–1001, 2008.

C. J. Skinner and L.-A. Vallet. Fitting Log-Linear Models to Contingency Tables from Surveys with Complex Sampling Designs: An Investigation of the Clogg-Eliason Approach. *Sociological methods and research*, pages 83–108, 2010.

M. Templ, A. Alfons, A. Kowarik, and B. Prantner. *VIM: Visualization and Imputation of Missing Values*, 2013. URL `http://CRAN.R-project.org/package=VIM`. R package version 4.0.0.

M. Templ, A. Kowarik, and B. Meindl. Introduction to Statistical Disclosure Control (SDC). Project: Relative to the testing of SDC algorithms and provision of practical SDC, data analysis OG, 2014a.

M. Templ, A. Kowarik, and B. Meindl. *sdcMicro: Statistical Disclosure Control Methods for Anonymization of Microdata and Risk Estimation*, 2014b. URL `http://CRAN.R-project.org/package=sdcMicro`. R package version 4.3.0.

L. Willenborg and T. de Waal. *Elements of Statistical Disclosure Control*. Springer-Verlag, 2001. ISBN 9780387951218.