Chapter 3
Disclosure Risk

Abstract One of the key tasks in SDC is to estimate the disclosure risk of individuals but also to estimate a global risk for the whole data set. A very basic idea is to calculate frequency counts of the categorical key variables. The concept of uniqueness and the concept of k-anonymity and l-diversity are important and outlined first. SUDA is extending the concept of k-anonymity it also searches for uniqueness in subsets of key variables. For surveys from complex designs, the estimation of frequency counts in the population and sample is of central interest. Mainly two approaches are used: the individual risk approach and the estimation of the global risk by log-linear models. For continuous key variables, other concepts are used to estimate the disclosure risk. They are rather based on distances than on counts. The risk estimation concepts presented here evaluate original data sets or data sets that are modified through traditional (perturbative) anonymization methods.

3.1 Introduction

Disclosure risk is defined based on assumptions of disclosure scenarios, that is, how the intruder might exploit the released data to reveal information about a respondent. For example, an intruder might achieve this by linking the released file with another data source that shares the same respondents and identifying variables. In another scenario, if an intruder knows that his/her acquaintance participated in the survey, he/she may be able to match his/her personal knowledge with the released data to learn new information about the acquaintance. In practice, most of the measures for assessing disclosure risks, as introduced below, are based on key variables, which are determined according to assumed disclosure scenarios. Risk assessment measures also differ for categorical and continuous variables.

A considerable number of research has been carried out in the area of statistical disclosure risk estimation. Some of the most influential works are Carlson (2002a, b), Hundepool et al. (2012), Willenborg and De Waal (2000), Skinner and Shlomo (2006).

Disclosure risk arises if a given data set is released. It is assumed that the risk r takes a non-negative real value ($r \geq 0$) and a risk of zero ($r = 0$) indicates no risk. Measuring the disclosure risk in a microdata set is a key task. Risk measures are

M. Templ, *Statistical Disclosure Control for Microdata*,
DOI 10.1007/978-3-319-50272-4_3

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 anonymization methods may reduce the disclosure risk (for anonymization methods, see Chap. 4).

In general, disclosure risk methods differ between categorical and continuous key variables. In the first subsections of this chapter only methods for categorical key variables are discussed. Thus a subset $\mathbf{Z}$ from the data set $\mathbf{U}$ is considered, with $\mathbf{d_j} \in \mathbf{Z}$ and $\mathbf{d_j}$ determines the j-th categorical variable.

Before estimating the disclosure risk, a closer look at frequency counts is necessary.

3.2 Frequency Counts

Computing frequency counts serves as a basis for many disclosure risk estimation methods. Therefore, this topic is intensively discussed on the next pages.

Consider a sample $\mathbf{S}$ with n observations or/and a finite population $\mathbf{U}$ with N observations. In general, if the frequency counts are estimated/calculated from a finite population $\mathbf{U}$ of size N, we speak about *population frequency counts*, if they are calculated from the sample $\mathbf{S}$, the term *sample frequency counts* is used.

The frequency counts can be computed for a combination of q variables that gives the distribution of frequency counts. The 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 variable.

Cross tabulation is a statistical process that summarizes categorical data to create a *contingency table*. A contingency table itself is a type of table that displays the frequency distribution of the categorical variables. The elements of a contingency table are denoted as cells. Every cell shows the frequency of one key whereas a key is one combination of categorical key variables.

All combinations of categories in the key variables can be calculated by cross tabulation of these variables. Each combination of values (=keys) 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 $\mathbf{U}$ or the sample $\mathbf{S}$ into C subpopulations/subsamples $\mathbf{U}_j \subseteq \mathbf{U}$ or $\mathbf{S}_j \subseteq \mathbf{S}$ respectively, with $j \in \{1, ..., C\}$.

To give an another example for keys, we consider 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").

The subpopulation $\mathbf{U}_j \subseteq \mathbf{U}$ or subsample $\mathbf{S}_j \subseteq \mathbf{S}$ contains all observations belonging to the j-th key, with $j \in \{1, 2, ..., C\}$. To give an example: if there are exactly five observations 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 $|\mathbf{U}_{woman,student,blue}| = 5$.

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

In R the function `table()` can be used to compute contingency tables on sample level. When having a population, this function can also be applied on population level. Otherwise, to estimate population frequencies, the function `tableWt()` is useful. It takes sample weights into account for Horwitz-Thompson weighted estimates (Horvitz and Thompson 1952) of population frequencies. Exemplary, the sample frequency calculation is shown based on two key variables for the Austrian EU-SILC data.

```
## sample frequency counts
table(eusilc[,c("db040", "pb220a")])

   ##                pb220a
   ## db040             AT   EU Other
   ##   Burgenland      453   16     7
   ##   Carinthia       842   19    26
   ##   Lower Austria 2207   26   107
   ##   Salzburg        655   25    83
   ##   Styria         1783   24    73
   ##   Tyrol           928   43    50
   ##   Upper Austria 2056   45   143
   ##   Vienna         1641   76   221
   ##   Vorarlberg      508    9    41
```

3.2.1 The Number of Cells of Equal Size

The number of cells of equal size (cell sizes) are obtained as follows. T_j is the number of cells of size j, i.e.

$$T_j = \sum_{i=1}^{C} \mathbb{1}(F_i = j), \;\; j = 0, 1, ..., N \quad , \tag{3.1}$$

The sample counterpart t_j is given by

$$t_j = \sum_{i=1}^{C} \mathbb{1}(f_i = j),\ \ j = 0, 1, ..., n \quad , \tag{3.2}$$

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

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

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. The relation between T_j and F_i

$$\sum_{j=1}^{N} jT_j = N = \sum_{i=1}^{C} F_i$$
$$\sum_{j=1}^{n} jt_j = n = \sum_{i=1}^{C} f_i \quad ,$$

we can proof as follows.

Proof

$$\bigcup_{i=1}^{C} \mathbf{U}_i = \mathbf{U}\ and\ \mathbf{U}_i \cap \mathbf{U}_j = \emptyset,\ \forall i \neq j$$
$$\Longleftrightarrow |\bigcup_{i=1}^{C} \mathbf{U}_i| = |\mathbf{U}|$$
$$\Longleftrightarrow \bigcup_{i=1}^{C} |\mathbf{U}_i| = |\mathbf{U}|$$
$$\Longleftrightarrow \sum_{i=1}^{C} F_i = N$$

□

As mentioned above, 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 of aboves equation, see also Sect. 3.2.1.

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

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

3.2.2 *Frequency Counts with Missing Values*

The following R code shows the frequency counts calculation with three categorical key variables (federal state (`db040`), household size (`hsize`) and citizenship (`pb220a`)) from the data set `eusilc`. The package **data.table** is used for fast calculations. For each key the frequencies are calculated with the following code.

```
dt <- data.table(eusilc)
dt[, .N ,by = list(db040, hsize, pb220a)]

   ##             db040 hsize pb220a   N
   ##   1:        Tyrol     3     AT 179
   ##   2:        Tyrol     3  Other   4
   ##   3:        Tyrol     3     NA  43
   ##   4:        Tyrol     4     AT 247
   ##   5:        Tyrol     4     NA 140
   ##  ---
   ## 221: Vorarlberg     2  Other   1
   ## 222:        Tyrol     8     AT   1
   ## 223:        Tyrol     7     AT   4
   ## 224:        Tyrol     7     NA   3
   ## 225: Burgenland     1     EU   1
```

There exist 225 combinations/keys. However, we see that missing values (`NA`) are considered as own category. However, given a missing value, an intruder must not know the true category. In the worst case scenario, he can predict the category correctly. In the best case scenario, the probability of guessing the correct category is one divided by the number of categories. In SDC, usually this second assumption is taken into account (see, e.g., Franconi and Polettini 2004).

However, more scenarios are thinkable. In the following five scenarios are reported how sample frequency counts can be calculated.

1. (current method) Missing values increase frequencies in other categories.
2. (conservative method) Misssing values do not increase frequencies in other categories but in those observation where a missing occurs.
3. (category size) Missing values do increase frequencies in other categories by a factor c. This method can be used as a general method to account for missing values in frequency calculations.
4. (conservative method 2) Misssing values do not increase frequencies in other categories.
5. (own category) Same as method 4, but missings are treated like an own category.

A (very) simple table should illustrate these methods. The default method in **sdcMicro** is based on the assumption that a missing value can stand for any category and so a missing value in a variable can increase the frequencies of several keys. For example, the frequency counts of the left table would lead to frequencies represented in the middle table (default approach) or in the right table (conservative approach).

key1	key2	key3
1	1	3
1	1	NA
2	1	3
NA	1	NA

$\longrightarrow$

key1	key2	key3	f_k
1	1	3	3
1	1	NA	3
2	1	3	2
NA	1	NA	4

till

key1	key2	key3	f_k
1	1	3	1
1	1	NA	3
2	1	3	1
NA	1	NA	4

The discussion on missing values continues in the next section, but also it is a topic in Sect. 4.2.2, because it makes sense to discuss it together with the topics related to local suppression.

3.2.3 Sample Frequencies in sdcMicro

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

Frequency counts are automatically estimated when creating a *sdcMicroObj* object. The general extractor function `get.sdcMicroObj` can be used (`slot` is equivalent) to extract sample and population frequencies from the current object:

```
sdc <- createSdcObj(eusilc,
          keyVars = c("db040", "hsize", "pb220a"),
          weightVar = "rb050", hhId = "db030")
head(get.sdcMicroObj(sdc, type="risk")$individual)

   ##              risk  fk        Fk    hier_risk
   ## [1,] 8.967734e-06 222 112014.46 6.044731e-05
   ## [2,] 4.308265e-05  47  23714.77 6.044731e-05
   ## [3,] 8.397756e-06 237 119583.00 6.044731e-05
   ## [4,] 5.250816e-06 387 190938.97 2.046126e-05
   ## [5,] 5.250816e-06 387 190938.97 2.046126e-05
   ## [6,] 4.979891e-06 408 201300.00 2.046126e-05
```

The column denoted by `fk` includes the sample frequency counts assigned to each observation, the other columns are discussed later.

But also without creating an object of class `sdcMicroObj`, one can use the function `freqCalc()` for frequency estimation directly on data frames. It includes basically three parameters (for benchmarking issues a fourth parameter is provided) determining the data set, the key variables and the vector of sampling weights.

```
args(freqCalc)

   ## function (x, keyVars, w = NULL, alpha = 1)
   ## NULL
```

(for details, see `?freqCalc`).

In the following, these frequency counts are calculated with this function and the counts are added to the data set `eusilc`.

```
## information on frequencies are assigned to each observation
counts <- freqCalc(eusilc,
        keyVars = c("db040", "hsize", "pb220a"))$fk
## add counts to data
eusilc <- cbind(eusilc, counts)
## first 6 rows of the sample
head(eusilc[,c("db040","hsize", "pb220a", "counts")])

   ##   db040 hsize pb220a counts
   ## 1 Tyrol     3     AT    222
   ## 2 Tyrol     3  Other     47
   ## 3 Tyrol     3   <NA>    237
   ## 4 Tyrol     4     AT    387
   ## 5 Tyrol     4     AT    387
   ## 6 Tyrol     4   <NA>    408
```

The frequencies are assigned to each individual. But we simple can aggregate this information, i.e. to receive sample frequencies (aggregated information on cells) presented for each key; we can do this with the following code.

```
X <- aggregate(counts ~ db040 + hsize + pb220a, eusilc, mean)
## number of keys
nrow(X)

   ## [1] 164

## unique keys (frequency = 1)
sum(X$counts == 1)

   ## [1] 2

## first 6 counts for keys (out of 164)
head(X[, c("db040", "hsize", "pb220a", "counts")])

   ##             db040 hsize pb220a counts
   ## 1      Burgenland     1     AT     57
   ## 2       Carinthia     1     AT    114
   ## 3 Lower Austria     1     AT    319
   ## 4        Salzburg     1     AT     93
   ## 5          Styria     1     AT    254
   ## 6           Tyrol     1     AT    100
```

There exist 164 possible keys (considering the specified three categorical key variables) and two unique combinations. Function `head()` shows the first 10 keys of the `eusilc` data set with its corresponding frequency counts.

In Fig. 3.1 the cell size indices are visualized from the `eusilc` data set related to three categorical key variables. There are many cells with small frequency counts and only few that include more than 200 observations (see Fig. 3.1). The figure can be produced with the following code.

```
library(ggplot2)
hist <- qplot(X$counts, xlab="Cell size", ylab="Count") +
  geom_bar(fill="grey50")
print(hist)
```

So far, we done all sample frequency calculations using the first method (*current method*). However, in **sdcMicro** a general approach—the method that we called *category size*—is available for use when using **sdcMicro** version >4.6.1 onwards. The function `freqCalc` has a function parameter `alpha`, which is a numeric value between 0 and 1 specifying how much keys that contain missing values (NAs) should contribute to the calculation of sample and population frequency counts. For the default value of 1 (*current method*), nothing changes with respect to the implementation in prior versions of **sdcMicro**. Each wildcard-match would be counted, while for `alpha=0` keys with missing values would be basically ignored.

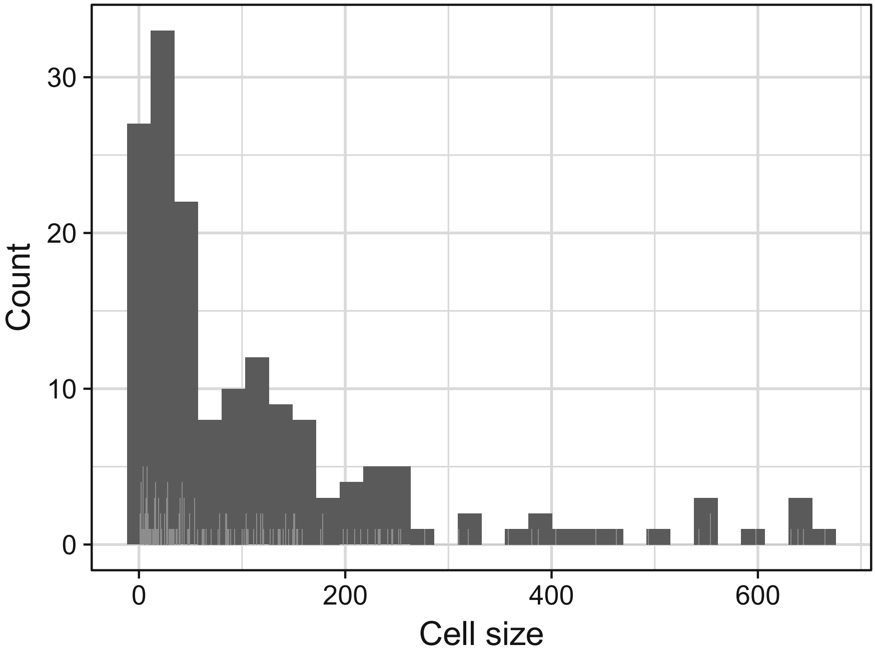


Fig. 3.1 Cell size indices of the Austrian EU-SILC data when selecting region, household size and citizenship as key variables

```
df <- data.frame("key1" = c(1,1,2,NA),
                 "key2" = c(1,1,1,1),
                 "key3" = c(3,NA,3,NA),
                 w = c(10,20,30,40))
f1 <- freqCalc(df, keyVars = 1:3, w = 4, alpha = 1)
f0 <- freqCalc(df, keyVars = 1:3, w = 4, alpha = 0)
f01 <- freqCalc(df, keyVars = 1:3, w= 4, alpha = 0.1)
d <- data.frame("f1"=f1$fk, "f0"=f0$fk, "f01"=f01$fk)
cbind(df, d, data.frame("F1"=f1$Fk, "F0"=f0$Fk, "F01"=f01$Fk))

  ##   key1 key2 key3  w f1 f0 f01  F1 F0 F01
  ## 1    1    1    3 10  3  1 1.2  70 10  16
  ## 2    1    1   NA 20  3  2 2.1  70 30  34
  ## 3    2    1    3 30  2  1 1.1  70 30  34
  ## 4   NA    1   NA 40  4  3 3.1 100 80  82
```

3.3 Principles of k-anonymity and l-diversity

Assuming that sample uniques are more likely to be re-identified, one way to protect confidentiality is to ensure that each distinct pattern of key variables is possessed by at least k records in the sample. This approach is called achieving k-anonymity (Samarati and Sweeney 1998; Samarati 2001; Sweeney 2002). More precisely, 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 $\forall j \in \{1, ..., n\}$.

A typical practice is to set $k = 3$, which ensures that the same pattern of key variables is possessed by at least three records in the sample. Using the previous notation, 3-anonymity means for all records.

Even if a group of observations fulfill k-anonymity, an intruder can be still discover sensitive information. For example, Table 3.1 satisfies 3-anonymity, given the two key variables gender and age. Suppose an intruder gets access to the sample inpatient records, however, and knows that his neighbor, a female in her twenties, recently went to the hospital. Since all records of females in their twenties have the same medical condition, the intruder discovers with certainty that his neighbor has cancer. In a different scenario, if the intruder has a male friend in his thirties who belongs to one of the first three records, the intruder knows that the incidence of his friend having heart disease is low and thus concludes that his friend has cancer.

The concept of l-diversity (Machanavajjhala et al. 2007) is used to address this limitation of k-anonymity. It was introduced as a stronger notion of privacy: a group of observations with the same pattern of key variables is l-diverse if it contains at least l "well-represented" values for the sensitive variable. Machanavajjhala et al. (2007)

Table 3.1 Example inpatient records illustrating k-anonymity and l-diversity

	Key variables		f_k	Sensitive variable	Distinct l-diversity
	Gender	Age group		Medical condition	
1	Male	30s	3	Cancer	2
2	Male	30s	3	Heart disease	2
3	Male	30s	3	Heart disease	2
4	Female	20s	3	Cancer	1
5	Female	20s	3	Cancer	1
6	Female	20s	3	Cancer	1

interpreted “well-represented” in a number of ways, and the simplest interpretation, distinct l-diversity, ensures that the sensitive variable has at least l distinct values for each group of observations with the same pattern of key variables. As shown in Table 3.1, the first three records are 2-diverse because they have two distinct values for the sensitive variable, medical condition.

Differences in values of the sensitive variable can be measured differently. We present here the distinct diversity that counts how many different values exist within a pattern/key. The l-diversity measure is automatically measured in **sdcMicro** for (and stored in) objects of class `sdcMicroObj` as soon as a sensitive variable is specified (using `createSdcObj`). Note that the measure can calculated at any time using `ldiversity(sdc)` with optional function parameters to select another sensitive variable, see `?ldiversity`. However, it can also be applied to data frames, where key variables (argument `keyVars`) and the sensitive variables (argument `ldiv_index`) must be specified as shown below:

```
res1 <- ldiversity(testdata,
                   keyVars=c("urbrur","water","sex","age"),
                   ldiv_index="income")
res1

   ## --------------------------
   ## L-Diversity Measures
   ## --------------------------
   ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   ##    1.00    4.00    8.00   10.85   17.00   35.00
```

Machanavajjhala et al. (2007) define three different l-diversity measures:

distinct l-diversity: as the simplest definition that ensures that at least l distinct values for the sensitive field in each key;

entropy l-diversity: as the most complex definition, which defines entropy of a key where the fraction of observations that have a sensitive value;

recursive l-diversity: as a compromise definition that ensures the most common sensitive value does not appear too often in a key while less common sensitive values are ensured not to appear too infrequently in the same key.

Additional, there is also a version implemented in **sdcMicro** when more than one sensitive variable is present in the data. This leads to multiple entropy and multi recursive l diversity measures. The result of all these measures are stored already in object `res1`, and the first six observations have the following l-diversity.

```
head(res1[,1:5])

   ##      income_Distinct_Ldiversity income_Entropy_Ldiversity
   ## [1,]                          7                   7.00000
   ## [2,]                          7                   7.00000
   ## [3,]                         19                  19.00000
   ## [4,]                         22                  21.65466
   ## [5,]                          5                   5.00000
   ## [6,]                          4                   4.00000
   ##      income_Recursive_Ldiversity MultiEntropy_Ldiversity
   ## [1,]                           7                       0
   ## [2,]                           7                       0
   ## [3,]                          19                       0
   ## [4,]                          21                       0
   ## [5,]                           5                       0
   ## [6,]                           4                       0
   ##      MultiRecursive_Ldiversity
   ## [1,]                         0
   ## [2,]                         0
   ## [3,]                         0
   ## [4,]                         0
   ## [5,]                         0
   ## [6,]                         0
```

For more information, see also the help files of **sdcMicro**.

Remark: k-anonymity depends on the chosen rules that determine how frequencies are estimated in case of missing values in the key variables. In Sect. 3.2.2 this has already been briefly touched, and five possible approaches how to estimate sample frequencies in case of missing values have been mentioned. This is further discussed in Sect. 4.2.2.

3.3.1 Simplified Estimation of Population Frequency Counts

Before more sophisticated methods are shown, the simplest approach to estimate population frequency counts is shown. For illustration, the toy data set given in Table 3.2 is used. In the following R code this data set is referred as `toyData`.

Of course, from Table 3.2 the direct identifiers have to be deleted before dissemination. In this demonstration only the variable `name` is a direct identifier. Let us fix `Gender` and `Occupation` as categorical key variables.

The population frequency counts are usually not known, since only few information is available about a population and typically not all key variables are present in the population.

Table 3.2 Toy data set (`toyData`) to illustrate the estimation of population frequency counts

	Name	Year of birth	Gender	Citizenship	Occupation	Income	Weight
1	Max Mustermann	1978	m	AUT	Worker	35,000	110
2	Josef Meier	1945	m	AUT	Pensioner	23,500	70
3	Sabine Schnuller	1991	w	AUT	Student	7000	80
4	John Doe	1966	m	US	Employee	41,200	120
5	Susan Rose	1989	w	AUT	Student	0	130
6	Markus Roller	1972	m	AUT	Employee	31,100	90
7	Christoph Valon	1944	m	AUT	Pensioner	21,400	150
8	Ulrike Mayer	1932	w	D	Pensioner	17,600	150
9	Stefan Fuchs	1992	m	AUT	Worker	27,500	130
10	Rainer Thomas	1950	m	AUT	Pensioner	25,700	150
11	Julia Gross	1976	w	AUT	Employee	37,000	140
12	Nadine Glatz	1987	w	AUT	Student	0	120
13	Makro Dilic	1990	m	AUT	Worker	21,050	90
14	Sandra Stadler	1941	w	AUT	Pensioner	28,500	80

It is often useful to assign the frequency counts to each observation. This information is automatically stored whenever an object of class *sdcMicroObj* is created. Note that also function `freqCalc` can be used to assign sample frequency counts to every observation of the data set.

Let us create first an object of class *sdcMicroObj*. Here we specify the key variables and provide the variable name for the vector of sampling weights.

```
sdcToy <- createSdcObj(toyData,
            keyVars = c("Gender", "Citizenship",
                        "Occupation"),
            numVars = "Income",
            weightVar = "Weight",
            )
```

As discussed by Willenborg and De Waal (2000) 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 general for this approach, the sample frequency counts are multiplied by their sampling weights and summed up. The sample and population frequency counts are already estimated and included in `sdcToy`. Let us extract this information to further investigate how the population frequency counts are estimated for this data set.

```
toy2 <- cbind(toyData[,c(3:5,7)],
            get.sdcMicroObj(sdcToy,
                      "risk")$individual[,2:3])
toy2

   ##    Gender Citizenship Occupation Weight fk  Fk
   ## 1       m         AUT     Worker    110  3 330
   ## 2       m         AUT  Pensioner     70  3 370
   ## 3       w         AUT    Student     80  3 330
   ## 4       m          US   Employee    120  1 120
   ## 5       w         AUT    Student    130  3 330
   ## 6       m         AUT   Employee     90  1  90
   ## 7       m         AUT  Pensioner    150  3 370
   ## 8       w           D  Pensioner    150  1 150
   ## 9       m         AUT     Worker    130  3 330
   ## 10      m         AUT  Pensioner    150  3 370
   ## 11      w         AUT   Employee    140  1 140
   ## 12      w         AUT    Student    120  3 330
   ## 13      m         AUT     Worker     90  3 330
   ## 14      w         AUT  Pensioner     80  1  80
```

We see that observation 6 is unique in the sample and the estimated population frequency $\hat{F}_k = 90$, which equals the sampling weight of observation 6. In addition we can observe that for the key $Gender = m \times Citizenship = AUT \times Occupation = Pensioner$ the sample frequency is 3, but the population frequency is 370. The estimated population frequencies are obtained by summing up the sample weights for observations corresponding to the same key. Population frequencies for the previous mentioned key can therefore be estimated by summation over the corresponding sampling weights, w_2, w_7 and w_{10}. In summary, three observations with the key $Genderm \times CitizenshipAUT \times OccupationPensioner$ exist in the sample and 370 observations with this pattern (key) can be expected to exist in the population.

In practice this simplified estimator of population frequency counts will not provide workable solutions to be used for estimating disclosure risk for complex survey data, see discussion Willenborg and De Waal (2000). For example, when 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 observation. If the weights are known for every observation in the sample data set, then the population frequencies $\hat{F}_j$ are the sum of the weights of each record that has the same key combination. Thus $\hat{F}_j = \sum_{i \in |\mathbf{S}_j|} w_i$, where $|\mathbf{S}_j|$ is a subsample of $\mathbf{S}$ regarding key j and w_i are the weights of record $\mathbf{i}$ in subsample $\mathbf{S}_j$.

To give a toy example to explain this issue, we assume that **U** in the following code is our population.

```
set.seed(12)
U <- data.frame("var1" = sample(1:2, 16, replace=TRUE),
                "var2" = sample(1:3, 16, replace=TRUE))
```

From **U** we draw a sample with simple random sampling. A seed for the random number generator is set to stay reproducible. Since we used simple random sampling and we have drawn 4 out of 16 observations ($\pi_i = \frac{1}{4}\ \ i = 1, \ldots, 4$), we know the sampling weights ($w_i = \frac{1}{\pi_i}$) and assign it to the sample **S**.

```
set.seed(124)
select <- sample(1:nrow(U), 4)
S <- U[select, ]
S$weights = rep(4, nrow(S))
S

   ##   var1 var2 weights
   ## 2    2    2       4
   ## 7    1    1       4
   ## 8    2    3       4
   ## 6    1    3       4
```

Now the frequencies can be calculated for the sample (f_i) and estimated for the population (F_i).

```
f <- freqCalc(S, c("var1", "var2"), w=3)
cbind(S, fk=f$fk, Fk=f$Fk)

   ##   var1 var2 weights fk Fk
   ## 2    2    2       4  1  4
   ## 7    1    1       4  1  4
   ## 8    2    3       4  1  4
   ## 6    1    3       4  1  4
```

We see from this output that the estimated population frequency counts for the key ($var1 = 2 \ \wedge \ var2 = 3$) is 4 as for any other estimated population frequencies. Next we see the true frequency counts in the population (`Fktrue`). For the third line, the true frequency in the population is 1 and we highly overestimate this frequency.

```
Fktrue <- freqCalc(U, c("var1", "var2"))
cbind(U[select, ], Fk=f$Fk, Fktrue=Fktrue$Fk[select])

   ##   var1 var2 Fk Fktrue
   ## 2    2    2  4      2
   ## 7    1    1  4      6
   ## 8    2    3  4      1
   ## 6    1    3  4      3
```

In other words, for complex samples, especially for socio-economic samples with comparable values of weights, we would always overestimate small population frequencies and if this estimation of population frequency will be the basis for risk estimation, the risk would highly be underestimated.

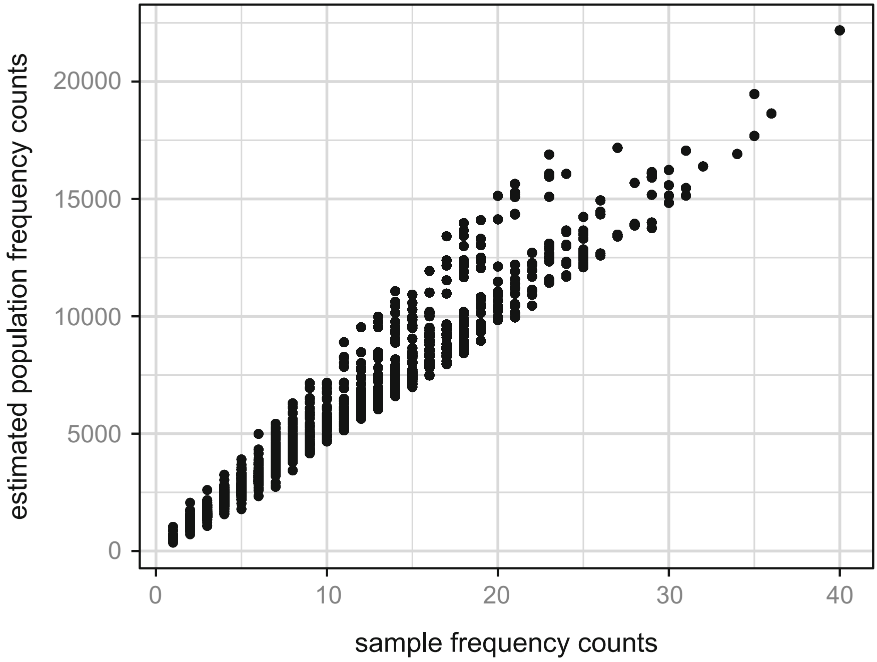
Of course, there is a relation between the sample frequency counts and the estimated population frequency counts, since the higher the sample frequencies, the more individuals contribute to the population frequencies. However, also the sampling weights play a role.

This relationship could also be shown using the following application. From the `eusilc` data set from package **laeken** the variables `age`, `pb220a` (citizenship) and `rb090` (gender) are selected as categorical key variables. First, we create an object of class *sdcMicroObj*.

```
library("laeken")
data(eusilc)
sdc <- createSdcObj(eusilc,
                    keyVars=c("age", "pb220a", "rb090", "db040"),
                    weightVar = "rb050")
```

We access the frequency counts and plot them.

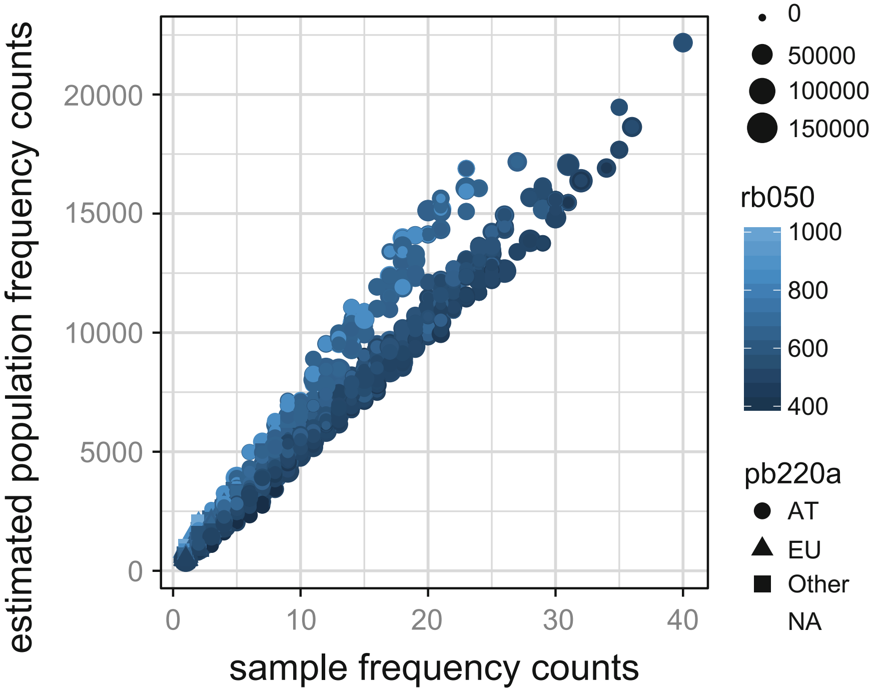
```
risk <- slot(sdc, "risk")$individual
freq <- data.frame(risk[, c("fk", "Fk")])

library("ggplot2")
gg <- ggplot(freq, aes(x=fk, y=Fk)) + geom_point() +
  xlab("sample frequency counts") +
  ylab("estimated population frequency counts")
print(gg)
```



We see that there is a linear trend, the higher the sample frequencies, the higher the population frequencies.

We can further investigate if some groups have lower frequencies, and we can observe the relation with income and the sampling weights.

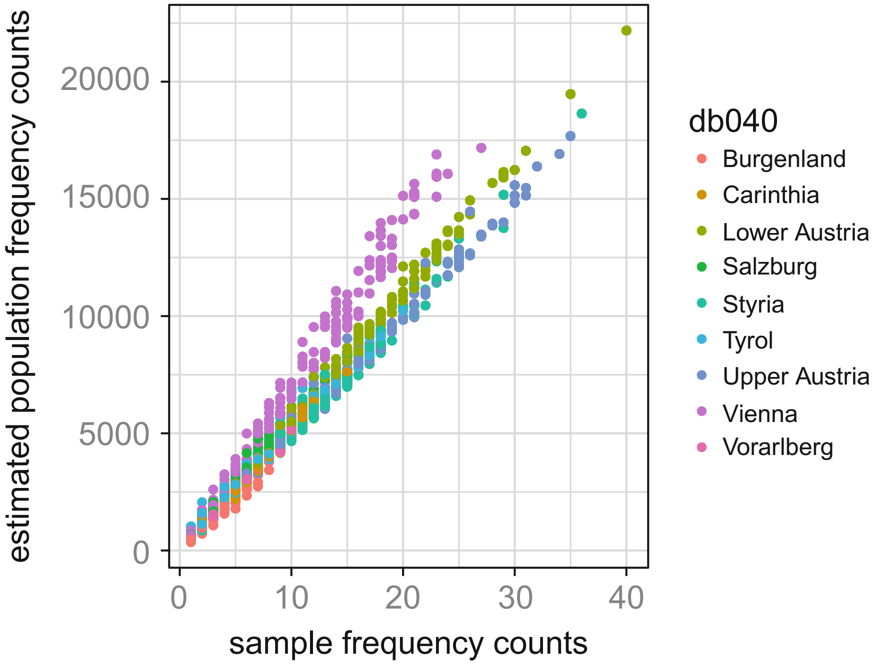
```
eusilc$fk <- freq$fk
eusilc$Fk <- freq$Fk
library(ggplot2)
gg <- ggplot(eusilc,
             aes(x=fk, y=Fk,
                 shape=pb220a, colour=rb050,
                 size=eqIncome)) +
             geom_point() +
             xlab("sample frequency counts") +
             ylab("estimated population frequency counts")
print(gg)
```



We see that individuals with higher incomes have generally lower frequencies according to the selected key variables. Moreover, the lower frequencies are more related to categories *EU* and *Others*. This is because *AT* is much more frequent than

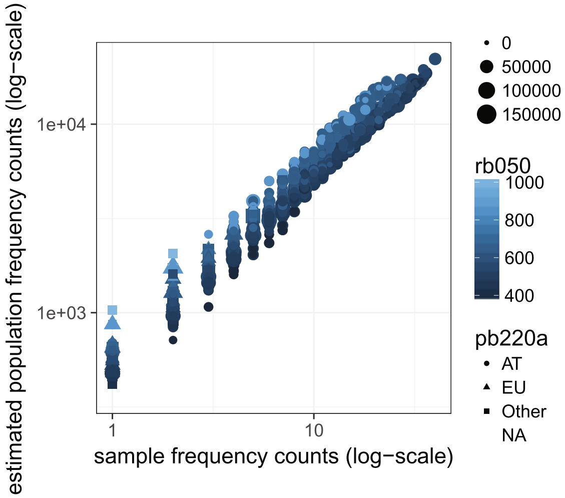
EU or *Others*. In addition, the sample design is indirectly reflected in this graphic. We can see this more clearly if we color the points depending on the region (db040) in the following graphic.

```
gg <- ggplot(eusilc, aes(x=fk, y=Fk, colour=db040)) +
            geom_point() + xlab("sample frequency counts") +
            ylab("estimated population frequency counts")
print(gg)
```



Lastly, we can look at log-transformed frequency counts. It is clearly visible that the estimated population frequencies depending on the high of the sampling weights.

```
gg <- ggplot(eusilc,
             aes(x=fk, y=Fk, shape=pb220a,
                 colour=rb050, size=eqIncome)) +
             geom_point() + scale_x_log10() + scale_y_log10() +
             xlab("sample frequency counts (log-scale)") +
             ylab("estimated population frequency counts (log-scale)")
print(gg)
```



Exercises:

Question 3.1 **Sample frequency counts versus estimated population frequency counts**
We now know how to calculate the sample frequency counts and already learned one possibility to estimate population frequency counts. Use the `eusilc` data set from package **laeken**. Assume the following disclosure scenario that defines `age`, `pb220a` (citizenship), `pl030` (education level), `rb090` (gender) and `hsize` (household size) as categorical key variables. Use the package **sdcMicro** to create an object of class *sdcMicroObj*. Access the sample and population frequency counts and plot them against each other. What can you observe?

3.4 Special Uniques Detection Algorithm (SUDA)

An alternative approach for defining disclosure risks is based on the concept of special uniqueness. For example, the eighth record in Table 3.3 is a sample unique with respect to the key variable set (i.e., {*age group*, *gender*, *income*, *education*}). Furthermore, a subset of the key variable set, for example, the combination of variables {*gender*, *education*} with the categories *male* and *university*}, is also unique in the sample. An observation is defined as a special unique with respect to a variable set Q, if it is sample unique both on Q and on a subset of Q (Elliott et al. 1998). Research has shown that special uniques are more likely to be population uniques than random uniques (Elliott et al. 2002).

Table 3.3 Example data set illustrating SUDA scores

	Age group	Gender	Income (k)	Education	f_k	SUDA score	Risk DIS-SUDA method
1	20s	Male	$\geq$50	High school	2	0	0
2	20s	Male	$\geq$50	High school	2	0	0
3	20s	Male	$\leq$50	High school	2	0	0
4	20s	Male	$\leq$50	High school	2	0	0
5	20s	Female	$\leq$50	University	1	1	0.0105
6	20s	Female	$\leq$50	High school	1	0.5	0.0046
7	20s	Female	$\leq$50	Middle school	1	1.75	0.0203
8	60s	Male	$\leq$50	University	1	2.25	0.0272

3.4.1 Minimal Sample Uniqueness

A set of computer algorithms, called SUDA, was designed to comprehensively detect and grade special uniques (Elliott et al. 2002). SUDA takes a two-step approach. In the first step, all unique attribute sets up to a user-specified size are located for each observation. SUDA considers only *Minimal Sample Uniques* (MSUs), which are unique variable sets without any unique subsets within a sample. In the example presented in Table 3.3, *male—university* is a MSU of observation 8 because none of its subsets, *male* or *university*, is unique in the sample. Also *60s* is a MSU in observation 8. An example for sample uniqueness but not being an MSU is the combination of *60s*, *male*, $\leq 50\,k$, *university*. This is a unique variable set, but not a MSU because its subsets (*60s*, *male*, *university*) and (*male*, *university*) are both unique subsets in the sample.

3.4.2 SUDA Scores

Once all MSUs have been found, a SUDA score is assigned to each observation indicating the risk using the size and distribution of MSUs within each observation (Elliott et al. 2002). The potential risk of the records is determined based on two issues:

1. the smaller the number of variables spanning the MSU within an observation, the higher the risk of the observation, and
2. the larger the number of MSUs in an observation, the higher the risk of the observation.

For each MSU of size k contained in a given observation, a score is computed by

$$s_i = \begin{cases} \frac{1}{q!}\prod_{i=k}^{M}(q-i), & \text{if } \; i \leq M. \\ 0, & \text{otherwise} \quad , \end{cases} \tag{3.3}$$

where M ($M > 0$) is the user-specified maximum size of MSUs, and q is the total number of categorical key variables in the data set. Note that if i is larger than M, the SUDA score should be set to zero. By definition, the smaller the size k of the MSU, the larger the score for the MSU (look at (q − i) in Eq. 3.3). The final SUDA score for the observation is computed by multiplying the scores for each MSU (have a look at the product in Eq. 3.3). In this way, observations with more MSUs are assigned a higher SUDA score.

The final SUDA score is calculated by normalizing these SUDA scores by dividing them by $q!$, with q being the number of key variables.

To illustrate how SUDA scores are calculated, we take a look at observation 8 in Table 3.3. This observation has two MSUs: *60s* of size 1, and (*male*, *university*) of size 2. Suppose the maximum size of MSUs is set at 3, the non-normalized score assigned to *60s* is computed by $\prod_{i=1}^{3}(4-i) = 6$, and the non-normalized score assigned to (*male*, *university*) is $\prod_{i=2}^{3}(4-i) = 2$. The normalized SUDA score is then obtained by normalizing the scores and by summation over these two normalized scores.

3.4.3 SUDA DIS Scores

In order to estimate observation-level disclosure risks, SUDA scores can be used in combination with the *Data Intrusion Simulation* (DIS) metric (Elliot and Manning 2003), which is a method for assessing a global disclosure risk for the entire data set. To receive the DIS score, loosely speaking, an iterative algorithm based on sampling of the data and matching of subsets of the sampled data with the original data is applied. This algorithm calculates the probabilities of correct matches given unique matches. It is, however, out of scope to precisely describe this algorithm here; we refer to Elliott et al. (2002), Elliot and Manning (2003) for details.

3.4.4 SUDA in sdcMicro

Both SUDA and DIS-SUDA scores can be computed using sdcMicro (Templ et al. 2015). Given that the implementation of SUDA can be computational demanding, sdcMicro uses an improved SUDA2 algorithm, which more effectively locates the boundaries of the search space for MSUs in the first step (Manning et al. 2008). Table 3.3 presents the observation-level risks estimated using the DIS-SUDA approach for the sample data set. Instead of replacing the risk measures introduced in Sect. 3.5.1, the SUDA scores and DIS-SUDA approach can be best used as an enhancement of the *k*-anonymity approach (see Sect. 3.3). For example, compared to the risk measures presented in Table 3.3, for records with the same sample frequency count, the DIS-SUDA score (see also Table 3.3) does not fully account for the sampling weights, while the risk measures based on negative binomial model are typically lower for records with larger sampling weights.

SUDA2 is implemented in **sdcMicro** as function `suda2()` based on `C++` code from the IHSN. Additional output, such as the contribution percentages of each variable to the score, are also available as an output of this function. The contribution to the SUDA score is calculated by assessing how often a category of a key variable contributes to the score. After an object of class `sdcMicroObj` has been built, no information about SUDA (dis) scores are stored. However, after applying SUDA on such an object, they are available, see the following code where also the print method is used. First we apply SUDA2 on a toy data set from Table 3.3.

```
tab <- data.frame("age" = c(rep("20s", 7), "60s"),
                  "gender" = c(rep("male", 4), rep("female", 3), "male"),
                  "income" = c("50k+", "50k+", rep("50k-", 6)),
                  "education" = c(rep("highschool", 4), "university",
                          "highschool", "middleschool", "university"))
su <- suda2(tab)
## print dis suda scores summary
su
```

```
##
## Dis suda scores table:
## - - - - - - - - - - -
##     Interval Number of records
## 1       == 0                 4
## 2 (0.0, 0.1]                 4
## 3 (0.1, 0.2]                 0
## 4 (0.2, 0.3]                 0
## 5 (0.3, 0.4]                 0
## 6 (0.4, 0.5]                 0
## 7 (0.5, 0.6]                 0
## 8 (0.6, 0.7]                 0
## 9      > 0.7                 0
## - - - - - - - - - - -
## Attribute contribution:
## - - - - - - - - - - -
##    variable contribution
## 1       age     40.90909
## 2    gender     27.27273
## 3    income      0.00000
## 4 education     68.18182
## - - - - - - - - - - -
```

We see that four observations has a considerable high risk. The individual SUDA scores and DIS scores are stored on the following list elements.

```
names(su)

   ## [1] "contributionPercent"
   ## [2] "score"
   ## [3] "disScore"
   ## [4] "attribute_contributions"
   ## [5] "attribute_level_contributions"
```

We have a look on the scores and dis SUDA scores.

```
su$score

   ## [1] 0.00 0.00 0.00 0.00 1.00 0.50 1.75 2.25

su$disScore

   ## [1] 0.000000000 0.000000000 0.000000000 0.000000000
   ## [5] 0.010460685 0.004580335 0.020271161 0.027212235
```

And we can evaluate which variables contributed most to these scores. For observation eight, this is, of course, variable age since it has a MSU of size 1 with respect to age that counts more than the MSU of size 2 regarding the combination of gender and university.

```
su$contributionPercent

   ##      age_contribution gender_contribution income_contribution
   ## [1,]        0.0000000           0.0000000                   0
   ## [2,]        0.0000000           0.0000000                   0
   ## [3,]        0.0000000           0.0000000                   0
   ## [4,]        0.0000000           0.0000000                   0
   ## [5,]        0.5000000           0.5000000                   0
   ## [6,]        0.0000000           1.0000000                   0
   ## [7,]        0.0000000           0.0000000                   0
   ## [8,]        0.7777778           0.2222222                   0
   ##      education_contribution
   ## [1,]              0.0000000
   ## [2,]              0.0000000
   ## [3,]              0.0000000
   ## [4,]              0.0000000
   ## [5,]              1.0000000
   ## [6,]              1.0000000
   ## [7,]              1.0000000
   ## [8,]              0.2222222
```

Let us resume by using a larger data set used before. Remember, we already created an *sdcMicroObj* called `sdc` from the data set `eusilc`. We apply SUDA directly on this object

```
sdc <- suda2(sdc)
```

The dis SUDA scores summary as well as all list elements can be extracted using `get.sdcMicroObj` or simply `slot`. We just look the results of the print method and obtain that 345 observations have higher dis SUDA score than 0.1.

```
su_silc <- slot(sdc, "risk")$suda
names(su_silc)

   ## [1] "contributionPercent"
   ## [2] "score"
   ## [3] "disScore"
   ## [4] "attribute_contributions"
   ## [5] "attribute_level_contributions"

su_silc

   ##
   ## Dis suda scores table:
   ## - - - - - - - - - - -
   ##     Interval Number of records
   ## 1         == 0             14482
   ## 2 (0.0, 0.1]               345
   ## 3 (0.1, 0.2]                 0
   ## 4 (0.2, 0.3]                 0
   ## 5 (0.3, 0.4]                 0
   ## 6 (0.4, 0.5]                 0
   ## 7 (0.5, 0.6]                 0
   ## 8 (0.6, 0.7]                 0
   ## 9       > 0.7                0
   ## - - - - - - - - - - -
   ## Attribute contribution:
   ## - - - - - - - - - - -
   ##   variable contribution
   ## 1      age    100.00000
   ## 2   pb220a     64.37055
   ## 3    rb090     26.60333
   ## 4    db040     80.99762
   ## - - - - - - - - - - -
```

3.5 The Individual Risk Approach

To estimate the frequencies of the population F_k, it is assumed that the population is drawn from a superpopulation. In fact, this means that the frequencies in the population either will be generated synthetically by drawing from a specific distribution of the frequency counts or quantiles of the assumed distribution of F_k are used. Using quantiles of the prior assumed distribution of F_k makes it possible to estimate the risk of each statistical unit. However, this estimation is just as good as the frequency counts of the population are modeled and how well the model assumption are fulfilled. Many suggestions exist in literature: the use of a Poisson-Gamma superpopulation model (Bethlehem et al. 1990), the Dirichlet-multinomial model (Hoshino and Takemura 1998), the negative binomial model (Benedetti and Franconi 1998; Franconi and Polettini 2004), a log-linear model (Skinner and Holmes 1998; Skinner and Shlomo 2006), a multinominal model (Forster and Webb 2007), the Poisson-inverse Gaussian model (Carlson 2002a) and references therein.

The estimation procedure of sample counts given the population counts is in the following modeled by assuming a negative binomial distribution (see Rinott and Shlomo 2006). This is also implemented in **sdcMicro** (see Templ et al. 2015) and called by the **sdcMicroGUI** (Kowarik et al. 2013).

3.5.1 The Benedetti-Franconi Model for Risk Estimation

For the popular *Benedetti-Franconi Model* (Benedetti and Franconi 1998; Franconi and Polettini 2004) or sometimes colloquially referred as the *Italian approach*, $F_k | f_k$ has to be estimated, i.e. the frequency counts in the population given the frequency counts in the sample. A common assumption is $F_k \sim Poisson(N\pi_k)$ (independently) (see, e.g., Franconi and Polettini 2004), where N is assumed to be known and with π_k the inclusion probabilities. (Binomial) Sampling from F_k means that $f_k|F_k \sim Bin(F_k, \pi_k)$. By standard calculations (see, e.g., Bethlehem et al. 1990) one gets

$$f_k \sim Poisson(N\pi_k) \text{ and } F_k|f_k \sim f_k + Poisson(N(1-\pi_k)) \ .$$

Concerning the risk estimation, the uncertainty on the frequency counts of the population is accounted in a Bayesian fashion by assuming that the population frequency given the sample frequency, $F_k|f_k$, is drawn from a negative binomial distribution with success probabilities p_k and the number of successes f_k (Polletini and Seri 2004; Rinott and Shlomo 2006). By using this assumption Benedetti and Franconi (1998) estimated the risk τ_2 by the well known and so called "model from Benedetti and Franconi". Using this background, Capobianchi et al. (2001) estimated the individual risk $\hat{r}_k$ as follows

$$\hat{r}_k = \left(\frac{\hat{p}_k}{1-\hat{p}_k}\right)^{f_k} \left\{ A_0 \left(1 + \sum_{j=0}^{f_k-3} (-1)^{j+1} \prod_{l=0}^{j} B_l \right) + (-1)^{f_k} \log(\hat{p}_k) \right\} \ , \tag{3.4}$$

whereas

$$\hat{p}_k = \frac{f_k}{\hat{F}_k} = \frac{f_k}{\sum\limits_{i \in \{j | x_{j.} = x_{k.}\}} \pi_i} \ ,$$

while

$$B_l = \frac{(f_k - 1 - l)^2}{(l+1)(f_k - 2 - l)} \frac{\hat{p}_k^{l+2-f_k} - 1}{\hat{p}_k^{l+1-f_k} - 1} \text{ and } A_0 = \frac{\hat{p}_k^{1-f_k} - 1}{f_k - 1} \ .$$

If $f_k = 1$ (Capobianchi et al. 2001) use

$$\hat{r}_k = \frac{\hat{p}_k}{1-\hat{p}_k} \log\left(\frac{1}{\hat{p}_k}\right) \ ,$$

while if $f_k = 2$ they use

$$\hat{r}_k = \frac{\hat{p}_k}{1-\hat{p}_k} - \left(\frac{\hat{p}_k}{1-\hat{p}_k}\right)^2 \log\left(\frac{1}{\hat{p}_k}\right) \quad .$$

If the sample is large the computation in Formula 3.4 becomes infeasible, but the following approximation gives workable approximations (Capobianchi et al. 2001):

$$\hat{r}_k = \frac{\hat{p}_k}{f_k - (1-\hat{p}_k)}$$

Using the toy data set from Table 3.3, we can compare the individual risk to the risk from SUDA.

```r
r <- measure_risk(tab,
  keyVars=c("age","gender","income","education"))$Res
```

The risk is high for all observations, but especially for observation five to eight. The risk is the same for these observation. Is it an indication that the individual risk approach is not as good compared to SUDA, since SUDA showed more variety in risk for each observation. Naturally observation eight should have higher risk which is not the case for the individual risk approach. Not at all. Or in other words, only for this particular toy data set that not includes sampling weights (nor have a hierarchical structure). We can learn also by looking follow-up examples that SUDA is a well-defined concept for measuring risk for census data or data without complex sampling designs.

In general, the individual risk methods considers sampling weights, i.e. data that are collected using a complex sampling design. Let us do the calculations on a larger data set. We use the `testdata` from package **sdcMicro**.

```r
sdc <- createSdcObj(testdata,
          keyVars=c(’urbrur’,’water’,’sex’,’age’),
          numVars=c(’expend’,’income’,’savings’),
          pramVars=c("walls"),
          w=’sampling_weight’,
          hhId=’ori_hid’)
```

The frequency counts can again be accessed by the following call, but also the estimated frequencies $\hat{F}_k$ explained in Sect. 3.3.1 and the individual risk is accessible. The risk as well as $\hat{F}_k$ is the same for each key k.

```r
risk <- get.sdcMicroObj(sdc, type="risk")$individual
head(risk)

   ##                risk  fk   Fk   hier_risk
   ## [1,] 1.663894e-03   7  700 0.004330996
   ## [2,] 1.663894e-03   7  700 0.004330996
   ## [3,] 5.552471e-04  19 1900 0.004330996
   ## [4,] 4.543389e-04  23 2300 0.004330996
   ## [5,] 2.493766e-03   5  500 0.009682082
   ## [6,] 3.322259e-03   4  400 0.009682082
```

However, if a data set also contains a hierarchical structure such as persons in households or employees in enterprises, we have to extend this approach, shown in the next section.

3.6 Disclosure Risks for Hierarchical Data

Many micro-data sets have hierarchical, or multilevel, structures; for example, individuals who are situated in households. Once an individual is re-identified, the data intruder may learn information about the other household members, too. It is important, therefore, to take into account the hierarchical structure of the data set when measuring disclosure risks. It is commonly assumed that the disclosure risk for a household is higher than or equal to the risk that at least one member of the household is re-identified. In any case, for hierarchical data, information collected at the higher hierarchical level (e.g., household level) would be equal for all individuals in the group belonging to that higher hierarchical level (e.g., household), e.g. household income or any household related information. This hierarchical structure creates a further level of disclosure risk because if one individual in the household is re-identified, the household structure allows for re-identification of the other household members in the same household. In addition, there is always information available that gives an indication which persons belong to the same household. This information can be included partly in the vector of sampling weights but often also this information is explicitly available in data sets (household ID).

A household-level disclosure risk can be estimated by subtracting the probability that no person from the household is re-identified from one. For example, if we consider a single household with three members, with individual disclosure risks of 0.1, 0.05 and 0.01, respectively, the disclosure risk for the entire household will be calculated as $1 - (1 - 0.1) \cdot (1 - 0.05) \cdot (1 - 0.01) = 0.15355$.

The individual and cluster/hierarchical risks are stored together with sample (f_k) and population counts (F_k) in slot `@risk$individual` and can be extracted by function `get.sdcMicroObj`. The household related individual risks can be seen in the following output. It is already available since in the previous code block, the function parameter `hhId` was set, and the household risk was automatically estimated when calling function `createSdcObj`.

```
head(cbind("household-ID"=testdata$ori_hid, risk))

   ##      household-ID         risk fk   Fk   hier_risk
   ## [1,]            1 1.663894e-03  7  700 0.004330996
   ## [2,]            1 1.663894e-03  7  700 0.004330996
   ## [3,]            1 5.552471e-04 19 1900 0.004330996
   ## [4,]            1 4.543389e-04 23 2300 0.004330996
   ## [5,]            2 2.493766e-03  5  500 0.009682082
   ## [6,]            2 3.322259e-03  4  400 0.009682082
```

Persons in the same households receive the same risk.

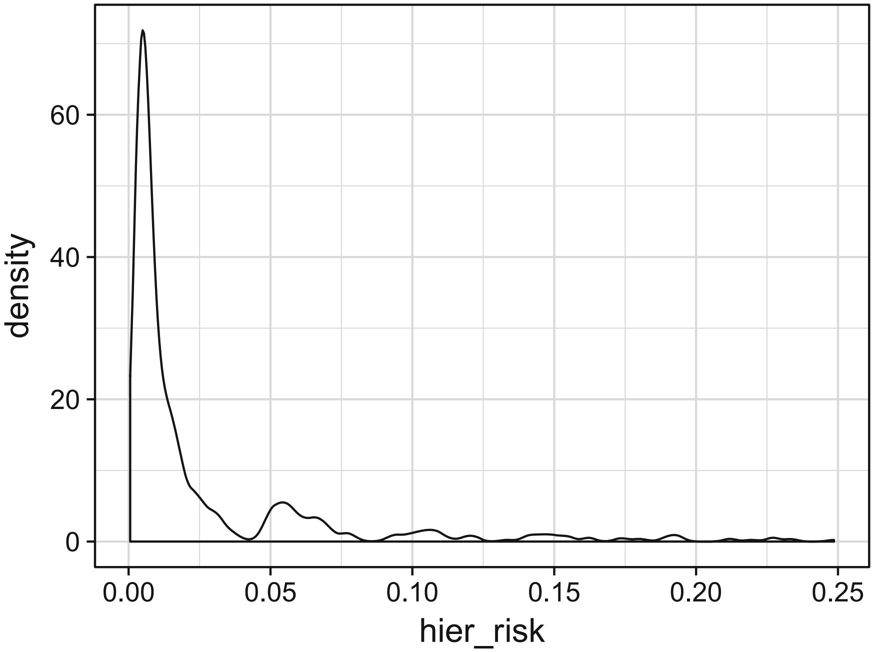


Fig. 3.2 Estimated density of the hierarchical risk

The distribution of the individual (hierarchical) risk is shown in Fig. 3.2.

It can be seen that quite some observations have high risk of disclosure, up to 0.25. Clearly, the risk should be reduced by applying SDC methods from Sect. 4.

Of course, there is also a relation between the risk and the estimated population frequencies, however this (negative) relation is not as strong (see Fig. 3.3), but in general this holds: the lower the sample and population frequency counts, the higher the individual risk.

Exercises:

Question 3.2 **Individual risk**
Take the `eusilc` data set from package **laeken**. Assume the following disclosure scenario that defines `age`, `pb220a` (citizenship), `pl030` (education level), `rb090` (gender) and `hsize` (household size) as categorical key variables. Use the package **sdcMicro** to create an object of class *sdcMicroObj* considering the sampling weights (function argument `weightVar` in `createSdcObj`) and the household ID (function argument `hhId`). Access the household risk and plot the distribution of the household risk. What can you detect? Are some estimated risks too high?

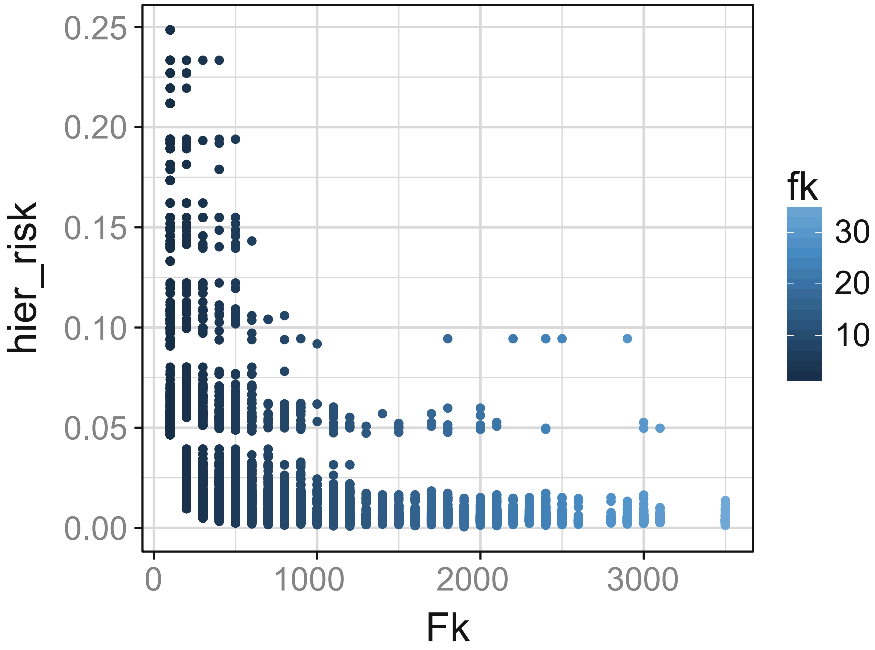


Fig. 3.3 Estimated population frequencies versus hierarchical risk

Question 3.3 **Individual risk versus household risk**
Take the same data and *sdcMicroObj* object created in the previous exercise. Access both the individual and the household risk and plot their distribution. What can you observe? Are the household risks higher than the individual risks?

3.7 Measuring Global Risks

In addition to record-level disclosure risk measures, a risk measure for the entire file-level or global risk micro-data set might be of interest. In this section, we present three common measures of global risks:

Expected number of re-identifications. The easiest measure of global risk is to sum up the record-level individual disclosure risks (defined in Sect. 3.5.1), which gives the expected number of re-identifications. Using the example from Sect. 3.6, the expected number of re-identifications is 117.2, the sum of the last column.

Global risk measure based on log-linear models. This measure, defined as the number of sample uniques that are also population uniques, is estimated using standard log-linear models (Skinner and Holmes 1998; Ichim 2008). The population frequency counts, or the number of units in the population that possess a specific pattern of key variables observed in the sample, are assumed to follow a Poisson distribution. The global risk can then be estimated by a standard log-linear model, using the main effects and interactions of key variables. A more precise definition is available in Skinner and Holmes (1998). The estimation of global risk using log-linear models is implemented in sdcMicro (Templ et al. 2015).

Benchmark approach. This measure counts the number of observations with record-level risks higher than a certain threshold and higher than the main part of the data. While the previous two measures indicate an overall re-identification risk for a microdata file, the benchmark approach is a relative measure that examines whether the distribution of record-level risks contains extreme values. For example, we can identify the number of records with individual risk satisfying the following conditions

$$r_i \geq 0.1 \wedge r_i \geq 2 \cdot (\tilde{\mathbf{r}} + 2 \cdot MAD(\mathbf{r})) \quad , \tag{3.5}$$

where $\mathbf{r}$ represents all record-level risks, and $MAD(\mathbf{r})$ is the median absolute deviation of all record-level risks.

Beneath is the print output of the corresponding function from **sdcMicro** showing both measures:

```
print(sdc, "risk")

   ## Risk measures:
   ##
   ## Number of observations with higher risk than the main part of the
   data: 0
   ## Expected number of re-identifications: 24.78 (0.54
   ##
   ## Information on hierarchical risk:
   ## Expected number of re-identifications: 117.20 (2.56
   ##
   ----------------------------------------------------------------------
```

If a cluster (e.g., households) has been defined, a global risk measurement taking into account this hierarchical structure is also reported.

3.7.1 *Measuring the Global Risk Using Log-Linear Models*:

In this section model based methods to estimate population frequency counts are considered as described in Carlson (2002a, b), Skinner and Shlomo (2006). 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 size of the population N, number of categories C and inclusion probabilities 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 $\lambda_j = N\pi_j$.

3.7.2 *Standard Log-Linear Model*

Log-linear models are used for modelling cell counts in contingency tables. 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:

$$log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\lambda} \quad , \tag{3.6}$$

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.

3.7.3 *Clogg and Eliason Method*

As described in Clogg and Eliason (1987),Agresti (2002),Skinner and Shlomo (2006) the Clogg and Eliason approach additional considers the survey weights towards Eq. (3.6). They extend the log-linear model from Eq. 3.6 with an offset term $\mathbf{z} = (z_1, ..., z_C)'$ and $z_k = \frac{f_k}{\hat{F}_k}$, 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:

$$log(\boldsymbol{\mu}) = log(\mathbf{z}) + \mathbf{X}\boldsymbol{\lambda} \quad . \tag{3.7}$$

3.7.4 Pseudo Maximum Likelihood Method

The fitted values for a linear model are solutions to the likelihood equations. We derive likelihood equations using Eq. (3.6) for a log-linear model. For a vector of frequency counts $\mathbf{f}$ with $\boldsymbol{\mu} = \mathbb{E}(\mathbf{f})$, the model is given by $log(\boldsymbol{\mu}) = \mathbf{X}\boldsymbol{\lambda}$, for which $log(\mu_i) = \sum_j x_{ij} \cdot \lambda_j, \forall i \in \{1, ..., C\}$. The log likelihood for Poisson sampling is:

$$L(\boldsymbol{\mu}) = \sum_i f_i \cdot log(\mu_i) - \sum_i log(\mu_i) \quad . \tag{3.8}$$

Through Eqs. (3.6 and 3.8) the pseudo maximum likelihood approach yields the following equation:

$$log(\hat{\mathbf{F}}) = \mathbf{X}\boldsymbol{\lambda} \quad . \tag{3.9}$$

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

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

$$log(\boldsymbol{\mu}) = \tilde{\mathbf{X}}\boldsymbol{\lambda} \quad , \tag{3.10}$$

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.

3.8 Application of the Log-Linear Models

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 specifying 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:

```
data(eusilc)
keyVars <- c("db040", "hsize", "rb090", "age", "pb220a", "pl030")
sdc <- createSdcObj(eusilc, keyVars = keyVars,
                    weightVar = "rb050", hhId = "db030")
form <- as.formula(paste(" ~ ", "db040 + hsize + rb090 +
             age + pb220a + age:rb090 + age:hsize +
             hsize:rb090"))
standardLLM <- as.formula(paste(c("fk",
                      as.character(form)),
                      collapse = ""))
standardLLM

  ## fk ~ db040 + hsize + rb090 + age + pb220a + age:rb090 + age:hsize +
  ##      hsize:rb090
  ## <environment: 0x117db9a60>

pseLLM <- as.formula(paste(c("Nk",
                      as.character(form)),
                      collapse = ""))
pseLLM

  ## Nk ~ db040 + hsize + rb090 + age + pb220a + age:rb090 + age:hsize +
  ##      hsize:rb090
  ## <environment: 0x117db9a60>

weightedLLM <- as.formula(paste(c("fk",
                    as.character(as.formula(
                      paste(c(form,"Fk"),
                            collapse="+")))),
                            collapse = ""))
weightedLLM

  ## fk ~ db040 + hsize + rb090 + age + pb220a + age:rb090 + age:hsize +
  ##      hsize:rb090 + Fk
  ## <environment: 0x117db9a60>
```

The vector named `keyVars` includes the considered categorical key variables. `standardLLM`, `pseLLM` and `weightedLLM` 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 this disclosure risk scenario. For the EC approach the formula `standardLLM` 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.

Next we need the frequency counts for each key. Remember that any function of **sdcMicro** (`freqCalc`, `createSdcObj`, `measure_risk`) assigns the frequencies to each observations. Therefore, the first action is to aggregate them to have it for each key.

```
## get frequencies
fk <- freqCalc(eusilc, keyVars, w="rb050")
## assign it to the data set
eusilc$fk <- as.numeric(fk$fk)
eusilc$Fk <- as.numeric(fk$Fk)
## aggregate, to have it for each key only
mu <- aggregate(fk ~ hsize + rb090 + age + db040 + pb220a + pl030,
                eusilc, unique)
## aggregate the weights
Fk <- aggregate(Fk ~ hsize + rb090 + age + db040 + pb220a + pl030,
                eusilc, unique)
## save it in a new data.frame
counts <- data.frame(mu, Fk=Fk$Fk)
#counts <- counts[,c(keyVars,"fk","weights")]
counts$age <- as.numeric(counts$age)
head(counts)

##   hsize rb090 age      db040 pb220a pl030 fk       Fk
## 1     5  male  16 Burgenland     AT     1  2 715.7143
## 2     7  male  16 Burgenland     AT     1  1 554.5000
## 3     4  male  18 Burgenland     AT     1  2 997.1515
## 4     4  male  19 Burgenland     AT     1  1 498.5758
## 5     4  male  20 Burgenland     AT     1  2 997.1515
## 6     5  male  20 Burgenland     AT     1  1 357.8571
```

The standard model can now be estimated as seen in the following code.

```
mod_standard <- glm(standardLLM, data = counts, family = poisson())
lambda_standard <- fitted(mod_standard)
summary(lambda_standard)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5746  1.5340  1.8870  1.8920  2.2610  3.7250
```

The fitted parameters and test statistics of the coefficients are presented in Table 3.4. It is clear to see that the intercept is significantly non-zero. The p-value of the federal states is in some cases not significant, but in general the variable corresponding to federal state (db040) has a significant contribution. The variables rb090 (gender) and age are not significant. The contribution of the variables hsize (household size) and pb220a (citizenship) is statistically significant at $\alpha = 0.05$.

The Clogg and Eliason method uses an offset term, which is defined as log-ratios of sample and estimated population frequency counts.

```
EC <- counts$fk/counts$Fk
EC <- log(EC + 0.1)
mod_EC <- glm(standardLLM, data = counts, family = poisson(), offset = EC)
lambda_EC <- fitted(mod_EC)
summary(lambda_EC)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5734  1.5340  1.8870  1.8920  2.2620  3.7220
```

Table 3.4 Fitted regression coefficients, standard errors, value of the test statistics and p-value from the standard log-linear model

	Estimate	Std. Error	z value	Pr(>$\vert z\vert$)
(Intercept)	0.4180	0.0829	5.04	0.0000
db040Carinthia	0.1020	0.0568	1.79	0.0728
db040Lower Austria	0.3812	0.0503	7.57	0.0000
db040Salzburg	0.0122	0.0585	0.21	0.8346
db040Styria	0.2380	0.0514	4.63	0.0000
db040Tyrol	0.1381	0.0555	2.49	0.0129
db040Upper Austria	0.3494	0.0506	6.91	0.0000
db040Vienna	0.3009	0.0515	5.84	0.0000
db040Vorarlberg	0.0527	0.0625	0.84	0.3986
hsize	0.0244	0.0167	1.46	0.1435
rb090female	−0.2620	0.0736	−3.56	0.0004
age	0.0070	0.0012	5.78	0.0000
pb220aEU	−0.6653	0.0605	−11.00	0.0000
pb220aOther	−0.5766	0.0383	−15.06	0.0000
rb090female:age	0.0021	0.0011	2.03	0.0424
hsize:age	−0.0018	0.0003	−5.49	0.0000
hsize:rb090female	−0.0114	0.0127	−0.90	0.3706

The summary of the regression model of the Clogg and Eliason method is shown in Table 3.5. The results are comparable to the standard method.

The pseudo Likelihood method uses the scaled population frequencies.

```r
## Pseudo Likelihood: Nk ~ keyVars
f <- sum(counts$fk) / sum(counts$Fk)
N_k <- round(counts$Fk * f) #round
counts <- data.frame(counts, Nk = N_k)
mod_pse <- glm(pseLLM, data = counts, family = poisson())
lambda_pse <- fitted(mod_pse)
summary(lambda_pse)

   ##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   ##  0.5235  1.5090  1.8950  1.9420  2.3390  4.3440
```

It’s summary is shown in Table 3.6. The only difference to the standard and Clogg and Eliason method is in the significance in the coefficients belonging to the federal states (`db040`), which is now mostly highly significant.

Table 3.5 Fitted regression coefficients, standard errors, value of the test statistics and p-value from the Clogg and Eliason method

	Estimate	Std. error	z value	Pr(>$\|z\|$)
(Intercept)	2.7009	0.0829	32.58	0.0000
db040Carinthia	0.1042	0.0568	1.83	0.0669
db040Lower Austria	0.3843	0.0503	7.63	0.0000
db040Salzburg	0.0162	0.0585	0.28	0.7816
db040Styria	0.2397	0.0514	4.67	0.0000
db040Tyrol	0.1406	0.0555	2.53	0.0113
db040Upper Austria	0.3510	0.0506	6.94	0.0000
db040Vienna	0.3070	0.0515	5.96	0.0000
db040Vorarlberg	0.0547	0.0625	0.88	0.3813
hsize	0.0240	0.0167	1.44	0.1506
rb090female	−0.2621	0.0737	−3.56	0.0004
age	0.0071	0.0012	5.79	0.0000
pb220aEU	−0.6653	0.0605	−11.00	0.0000
pb220aOther	−0.5764	0.0383	−15.05	0.0000
rb090female:age	0.0021	0.0011	2.04	0.0418
hsize:age	−0.0018	0.0003	−5.51	0.0000
hsize:rb090female	−0.0114	0.0127	−0.90	0.3692

The weighted log-linear version (see the results in Table 3.7) do have the summed weights (population frequency counts) as one of the predictors. The results differs from the previous methods, and only the intercept, citizenship (`pb220a`) and the sum of weights (`Fk`) are significant.

```
mod_w <- glm(weightedLLM, data = counts, family = poisson())
lambda_w <- fitted(mod_w)
summary(lambda_w)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    0.868    1.344    1.456    1.892    1.831   44.880
```

The estimated $\hat{\lambda}_k$, $k = 1, ..., C$ are input to the global risk measures explained in the following section.

Table 3.6 Fitted regression coefficients, standard errors, value of the test statistics and p-value from the pseudo maximum likelihood method

	Estimate	Std. error	z value	Pr(>|z|)
(Intercept)	0.3952	0.0829	4.76	0.0000
db040Carinthia	0.1580	0.0579	2.73	0.0064
db040Lower Austria	0.4539	0.0515	8.82	0.0000
db040Salzburg	0.0867	0.0594	1.46	0.1443
db040Styria	0.2683	0.0527	5.09	0.0000
db040Tyrol	0.2115	0.0565	3.74	0.0002
db040Upper Austria	0.3807	0.0519	7.34	0.0000
db040Vienna	0.5517	0.0518	10.66	0.0000
db040Vorarlberg	0.1127	0.0635	1.78	0.0757
hsize	0.0127	0.0167	0.76	0.4469
rb090female	−0.2605	0.0728	−3.58	0.0003
age	0.0073	0.0012	6.05	0.0000
pb220aEU	−0.6686	0.0588	−11.36	0.0000
pb220aOther	−0.5886	0.0377	−15.60	0.0000
rb090female:age	0.0028	0.0010	2.69	0.0071
hsize:age	−0.0020	0.0003	−5.96	0.0000
hsize:rb090female	−0.0170	0.0127	−1.34	0.1815

3.9 Global Risk Measures

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

$$\mathbb{E}(1/F_j) = \sum_{i \in \mathbb{N}} \frac{1}{i} \mathbb{P}(F_j = i) \quad , \tag{3.11}$$

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 (3.11).

As mentioned above we consider a random sample **S** of a finite population **U** of size N. The sample data is available to the intruder. Let f_j be the sample frequency counts. This leads to two measures of interest as described in Skinner and Shlomo (2006):

$$m_1 = \mathbb{E}(1/F_j | f_j) \quad , \tag{3.12}$$

Table 3.7 Fitted regression coefficients, standard errors, value of the test statistics and p-value from the weighted log-linear method

	Estimate	Std. error	z value	Pr(>$\|z\|$)
(Intercept)	0.1371	0.0846	1.62	0.1052
db040Carinthia	−0.0016	0.0569	−0.03	0.9782
db040Lower Austria	−0.0327	0.0511	−0.64	0.5229
db040Salzburg	−0.0950	0.0586	−1.62	0.1047
db040Styria	0.0440	0.0515	0.85	0.3929
db040Tyrol	−0.0147	0.0556	−0.26	0.7921
db040Upper Austria	0.0620	0.0508	1.22	0.2229
db040Vienna	−0.2285	0.0527	−4.33	0.0000
db040Vorarlberg	−0.0314	0.0625	−0.50	0.6157
hsize	0.0031	0.0171	0.18	0.8570
rb090female	0.0013	0.0749	0.02	0.9859
age	−0.0007	0.0013	−0.53	0.5948
pb220aEU	−0.2705	0.0609	−4.44	0.0000
pb220aOther	−0.2369	0.0389	−6.09	0.0000
Fk	0.0004	0.0000	72.29	0.0000
rb090female:age	−0.0002	0.0011	−0.16	0.8707
hsize:age	0.0002	0.0003	0.57	0.5662
hsize:rb090female	−0.0096	0.0128	−0.75	0.4533

$$m_2 = \mathbb{P}(F_j = 1 | f_j) \quad . \tag{3.13}$$

Under random sampling the pairs (F_j, f_j) are independent and the first measure (3.12) is the conditional expectation of $1/F_j$ and second (3.13) the conditional probability that $F_j = 1$ given f_j. When $f_j = 1$, (3.12) is highest, which is the worst case. Additionally the following holds for (3.13):

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

Considering the worst case, i.e. $f_j = 1$, leads to the focus on the following measures:

$$m_{1j} = \mathbb{P}(F_j = 1 | f_j = 1) \quad , \tag{3.14}$$

$$m_{2j} = \mathbb{E}(1/F_j | f_j = 1) \quad . \tag{3.15}$$

The measures given in Eqs. (3.14 and 3.15) 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

$$\hat{\tau}_1 = \sum_{\{j:f_j=1\}} m_{1j} = \sum_{\{j:f_j=1\}} \mathbb{P}(F_j = 1 | f_j = 1) \quad , \tag{3.16}$$

$$\hat{\tau}_2 = \sum_{\{j:f_j=1\}} m_{2j} = \sum_{\{j:f_j=1\}} \mathbb{E}(1/F_j | f_j = 1) \quad . \tag{3.17}$$

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 (Skinner and Shlomo 2006). If the count of combinations C is large, $\hat{\tau}_1$ will closely approximate τ_1,

$$\hat{\tau}_1 \xrightarrow{C\to\infty} \tau_1 = \sum_{j\geq 1} \mathbb{1}(f_j = 1, F_j = 1) \quad , \tag{3.18}$$

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

$$\hat{\tau}_2 \xrightarrow{C\to\infty} \tau_2 = \sum_{j\geq 1} \frac{\mathbb{1}(f_j = 1)}{F_j} \quad . \tag{3.19}$$

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

$$\mathbb{P}(F_j = i) = \frac{\lambda_j^i e^{-\lambda_j}}{i!}, \; i \in \{0, 1, 2, 3, ...\} \quad . \tag{3.20}$$

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

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

$$\mathbb{P}(F_j = 1 | f_j = 1) = e^{-\lambda_j(1-\pi_j)} \quad . \tag{3.21}$$

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

$$\hat{\tau}_1 = \sum_j \mathbb{1}(f_j = 1)\mathbb{P}(F_j = 1 | f_j = 1) = \sum_{\{j : f_j = 1\}} e^{-\lambda_j(1-\pi_j)} \quad , \tag{3.22}$$

$$\hat{\tau}_2 = \sum_j \mathbb{E}(\frac{1}{F_j} | f_j = 1) = \sum_{\{j : f_j = 1\}} \frac{1 - e^{-\lambda_j(1-\pi_j)}}{\lambda_j(1 - \pi_j)} \quad . \tag{3.23}$$

With the following code the risk measures can be estimated. The function `modRisk` includes also all steps shown in the previous code chunks.

```
## risk for unmodified data using six key variables
m1 <- modRisk(sdc, method = "default", weights = eusilc$rb050,
         formulaM = form, bound = 5)@risk$model[1:2]
m2 <- modRisk(sdc, method = "CE", weights = eusilc$rb050,
         formulaM = form, bound = 5)@risk$model[1:2]
m3 <- modRisk(sdc, method = "PML", weights = eusilc$rb050,
         formulaM = form, bound = 5)@risk$model[1:2]
m4 <- modRisk(sdc, method = "weightedLLM", weights = eusilc$rb050,
         formulaM = form, bound = 5)@risk$model[1:2]
modelrisk <- data.frame(
               "method" = c("standard", "CE", "PML", "weightedLLM"),
               "tau1" = c(m1$gr1, m2$gr1, m3$gr1, m4$gr1),
               "tau2" = c(m1$gr2, m2$gr2, m3$gr2, m4$gr2))
modelrisk

   ##         method      tau1      tau2
   ## 1    standard 0.2964092 0.3604305
   ## 2          CE 0.2952641 0.3596631
   ## 3         PML 0.2960892 0.3596322
   ## 4 weightedLLM 0.3241130 0.3829982
```

Since we have 4109 uniques (27,78% of the data), the risk should be high. This is true for both measures $\hat{\tau}_1$ and $\hat{\tau}_1$. The risk is higher but comparable through all methods for $\hat{\tau}_2$.

Let us compare these risk estimates with previous risk approaches, the k-anonymity approach, the individual risk approach and SUDA2.

The percentage of observations violating 2-anonymity is 27.713% (4109 observations) and 3-anonymity is 46.854% (6947); see `print(sdc)` for details. This is also reflected in the SUDA scores table:

```
sdc <- suda2(sdc)
slot(sdc, "risk")$suda2

   ##
   ## Dis suda scores table:
   ## - - - - - - - - - - - -
   ##     Interval Number of records
   ## 1        == 0             10745
   ## 2 (0.0, 0.1]               4058
   ## 3 (0.1, 0.2]                 22
   ## 4 (0.2, 0.3]                  2
   ## 5 (0.3, 0.4]                  0
   ## 6 (0.4, 0.5]                  0
   ## 7 (0.5, 0.6]                  0
   ## 8 (0.6, 0.7]                  0
   ## 9       > 0.7                 0
   ## - - - - - - - - - - - -
   ## Attribute contribution:
   ## - - - - - - - - - - - -
   ##   variable contribution
   ## 1    db040     66.98131
   ## 2    hsize     64.66156
   ## 3    rb090     29.92720
   ## 4      age     95.66937
   ## 5   pb220a     31.54173
   ## 6    pl030     57.12919
   ## - - - - - - - - - - - -
```

However, the individual risk is

```
print(sdc, "risk")

   ## Risk measures:
   ##
   ## Number of observations with higher risk than the main part of the
   data: 0
   ## Expected number of re-identifications: 57.49 (0.39
   ##
   ## Information on hierarchical risk:
   ## Expected number of re-identifications: 199.16 (1.34
   ##
   ----------------------------------------------------------------------
```

This is much smaller than the risk estimated by log-linear modelling. Because of the high amount of uniques, the individual risk may underestimate the true risk in this case.

3.10 Quality of the Risk Measures Under Different Sampling Designs

Surveys almost always are not drawn with simple random sampling.

The assumptions that $F_j \sim Poisson(\lambda_j)$ and that the λ_j fit the log-linear model are unaffected by a complex sampling scheme (Skinner and Shlomo 2006). However, 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 general an usable approximation $\mathbb{P}(f_j = 1|F_j) \approx F_j\pi_j(1-\pi_j)^{F_j-1}$ provides good results.

The models are tested by Totter (2015) 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. From a close-to-reality population samples are drawn with this design. Knowing the true risk, the risk estimates from the samples have been evaluated.

The models sometimes underestimates the true disclosure risk depending on the disclosure scenario. One important point for good model performances is to choose a well-defined good interaction model (see also Skinner and Shlomo 2006). If the quality of the model is weak and/or there are too few predictors the disclosure risk will be underestimated. Model selection, i.e. to find a good fitting model with high predictive power, is therefore a crucial part of any risk estimation using log-linear models. Another criterion 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, equal stratified sampling and proportional stratified sampling. The weighted log-linear model performs worst. Depending on the disclosure scenario, a few times the pseudo maximum method turned out to give worse predictions of disclosure risk.

Exercises:

Question 3.4 Estimate the global risk for data set `eusilc` as done above. Then, use a subset of `eusilc` and compare the results on the global disclosure risk. Do smaller data sets imply higher risks?

Question 3.5 Compare other risk measures such as the global risk estimated from the household risks. Does a smaller data set still imply a higher risk?

3.11 Disclosure Risk for Continuous Variables

The concept of uniqueness might not be applicable to continuous key variables, especially those with an infinite range, since almost every record in the data set will then be identified as unique. In this case, a more applicable method is to assess risk based on record linkages.

Assume a disclosure scenario where an intruder has access to a data set that has been perturbed before release, as well as an external data source that contains information on the same respondents included in the released data set. Because the original values of the released data set have been perturbed, the intruder attempts to match records in the released data set with those in the external data set using common variables. Assume that the external data source, to which the intruder has access, is the original data file of the released data set. Essentially, the record linkage approach assesses to what extent records in the perturbed data file can be correctly matched with those in the original data file. There are three general approaches to record linkage:

Distance-based record linkage. Pagliuca and Seri (1999) computes distances between records in the original data set and the protected data set. Suppose we have obtained a protected data set A' after applying some SDC methods to the original data set A. For each record in the protected data set A, we compute its distance to every record in the original data set, and consider the nearest and the second nearest records. Suppose we have identified x_1 and x_2 from the original data set as the nearest and second-nearest records, respectively, from record x_i in the protected data set. If x_1 is the original record used to generate x_i, or, in other words, record x_i in the protected data set A' and in the original data set A refer to the same respondent, then we mark record x_i "linked". Similarly, if record x_i was generated from r_2 (the second-nearest record in the original data set), we mark x_i "linked to the 2nd nearest". We proceed the same way for every record in the protected data set A'. Finally, disclosure risk is defined as the percentage of records in the protected data set A' marked as "linked" or "linked to the 2nd nearest". This record-linkage approach based on distance is compute-intensive and thus might not be applicable for large data sets.

Probabilistic record linkage. Alternatively, probabilistic record linkage (Jaro 1989) pairs records in the original and protected data sets, and uses an algorithm to assign a weight for each pair that indicates the likelihood that the two records refer to the same respondent. Pairs with weights higher than a specific threshold are labeled as "linked", and the percentage of records in the protected data marked as "linked" is the disclosure risk.

Interval disclosure. In addition, a third risk measure is called interval disclosure (Pagliuca and Seri 1999), which simplifies the distance-based record linkage and thus is more applicable for large data sets. In this approach, after applying SDC methods to the original values, we construct an interval around each masked value. The width of the interval is based on the rank of the value the variable takes on or its standard deviation. We then examine whether the original value of the variable falls within the interval. The measure of disclosure risk is the proportion of original values that fall into the interval, assuming a worst case scenario that any original value that falls within the interval presents a correct match (or refers to the same respondent as) the masked value.

The latter approach is also described in Mateo-Sanz et al. (2004).The length of the intervals based on the standard deviation of the variable under consideration is calculated (see Fig. 3.4, upper left graphic; the boxes express the intervals).

Distance-based risks for continuous key variables are automatically estimated for objects of class `sdcMicroObj` using function `dRisk()`.

Let us create again an object of class *sdcMicroObj*. Note that continuous key variables have to be specified.

```
sdc <- createSdcObj(testdata,
          keyVars=c('urbrur','water','sex','age'),
          numVars=c('expend','income','savings'),
          pramVars=c("walls"),
          w='sampling_weight',
          hhId='ori_hid')
```

Current risks can be printed with:

```
print(sdc, "numrisk")

   ## Numerical key variables: expend, income, savings
   ##
   ## Disclosure risk is currently between [0.00
   ##
   ## Current Information Loss:
   ##   - IL1: 0.00
   ##   - Difference of Eigenvalues: 0.000
   ## --------------------------------------------------------------------
```

This reports the percentage of observations falling within an interval centered on its masked value whereas the upper bound corresponds to a worst case scenario where an intruder is sure that each nearest neighbor is indeed the true link. For a more detailed discussion we refer to Mateo-Sanz et al. (2004). Since no anonymization has been applied on the continuous key variables, the disclosure risk can be high (up to 100%) (and the information loss, discussed later in Chap. 5), is 0.

After applying a perturbation method (details on methods can be found in the next chapter), the risk is smaller.

```
sdc <- microaggregation(sdc)
print(sdc, "numrisk")

   ## Numerical key variables: expend, income, savings
   ##
   ## Disclosure risk is currently between [0.00
   ##
   ## Current Information Loss:
   ##   - IL1: 0.08
   ##   - Difference of Eigenvalues: 0.020
   ## --------------------------------------------------------------------

## try another method
sdc <- undolast(sdc)
sdc <- addNoise(sdc, method="correlated2")
print(sdc, "numrisk")

   ## Numerical key variables: expend, income, savings
   ##
   ## Disclosure risk is currently between [0.00
   ##
   ## Current Information Loss:
   ##   - IL1: 0.11
   ##   - Difference of Eigenvalues: 0.260
   ## --------------------------------------------------------------------
```

Exercises:

Question 3.6 **Risk for continuous variables**
Take the `testdata` data set from package **sdcMicro** again. Assume the following disclosure scenario: Set variables `expend`, `income` and `savings` as continuous key variables and optionally define a few variables as categorical key variables. Use the package **sdcMicro** to create an object of class *sdcMicroObj* by also considering the sampling weights (function argument `weightVar` in function `createSdcObj`). Apply additive noise and correlated noise and determine which of the two methods includes higher risk. Both methods can be specified in function `addNoise`.

3.12 Special Treatment of Outliers When Calculating Disclosure Risks

It is worth to show alternatives to the previous distance-based risk measure. Such alternatives took either distances between every observation into account or are based on covariance estimation (as shown here). Thus, they are computational more intensive, which is also the reason why they are not available in **sdcMicroGUI** but only in **sdcMicro** for experienced users.

Almost all data sets used in official statistics contain units whose values in at least one variable are quite different from the general observations. As a result, these variables are very asymmetrically distributed. Examples of such outliers might be enterprises with a very high value for turnover or persons with extremely high income. In addition, multivariate outliers exist (see Templ and Meindl 2008).

Unfortunately, intruders may want to disclose a large enterprise or an enterprise with specific characteristics. Since enterprises are often sampled with certainty or have a sampling weight close to 1, intruders can often be very confident that the enterprise they want to disclose has been sampled. In contrast, an intruder may not be as interested to disclose statistical units that exhibit the same behavior as most other observations. For these reasons, it is good practice to define measures of disclosure risk that take the outlyingness of an observation into account. For details, see Templ and Meindl (2008). Outliers should be much more perturbed than non-outliers because these units are easier to re-identify even when the distance from the masked observation to its original observation is relatively large.

This method for risk estimation (called RMDID2 in Fig. 3.4) is also included in the **sdcMicro** package. It works as described in Templ and Meindl (2008) and is listed as follows:

1. Robust Mahalanobis distances (*RMD*) (see, for example Maronna et al. 2006) are estimated between observations (continuous variables) to obtain a robust, multivariate distance for each unit.
2. Intervals are estimated for each observation around every data point of the original data points. The length of the intervals depends on squared distances calculated in step 1 and an additional scale parameter. The higher the *RMD* of an observation, the larger the corresponding intervals.
3. Check whether the corresponding masked values of a unit fall into the intervals around the original values. If the masked value lies within such an interval, the entire observation is considered unsafe. We obtain a vector indicating which observations are safe or which are not. For all unsafe units, at least m other observations from the masked data should be very close. Close is quantified by specifying a parameter for the length of the intervals around this observation using Euclidean distances. If more than m points lie within these small intervals, we can conclude that the observation is *safe*.

Figure 3.4 depicts the idea of weighting disclosure risk intervals. For simple methods (top left and right graphics), the rectangular regions around each value are the same size for each observation. Our proposed methods take the *RMD*s of each observation into account. The difference between the bottom right and left graphics is that, for method *RMDID2*, rectangular regions are calculated around each masked variable as well. If an observation of the masked variable falls into an interval around the original value, check whether this observation has close neighbors. If the values of at least m other masked observations can be found inside a second interval around this masked observation, these observations are considered *safe*.

These methods are also implemented and available in sdcMicro as `dRisk()` and `dRiskRMD()`. The former is automatically applied to objects of class *sdcMicroObj*, while the latter has to be specified explicitly and can currently not be applied using the graphical user interface.

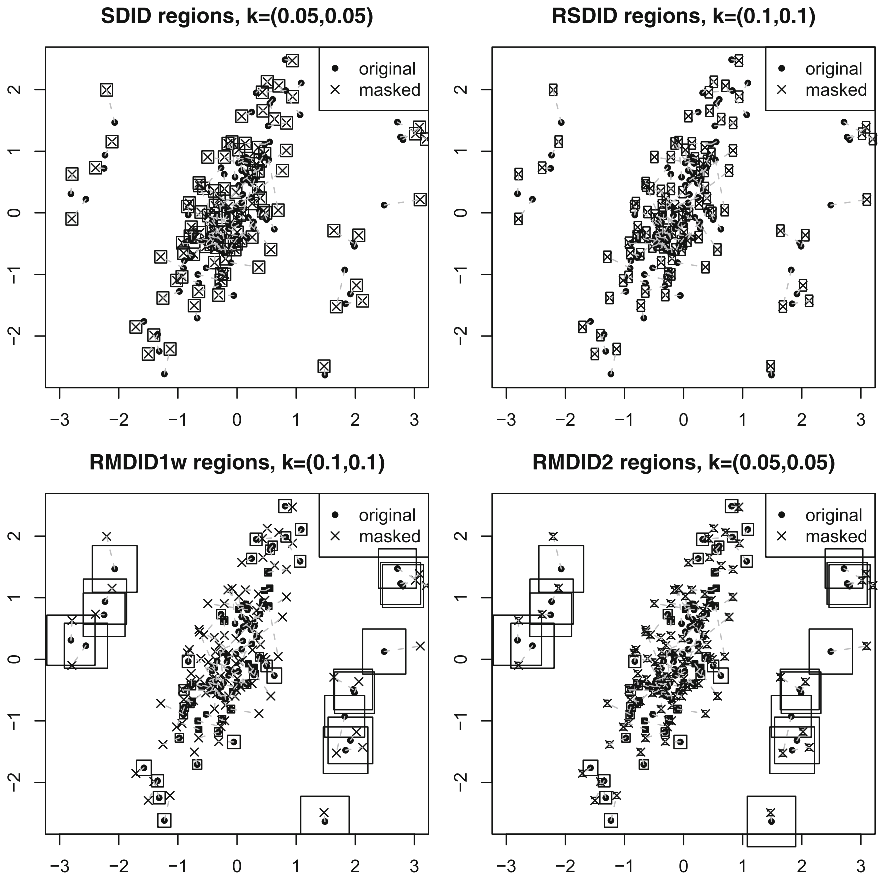


Fig. 3.4 Original and corresponding masked observations (perturbed by adding additive noise). In the *bottom right* graphic, small additional regions are plotted around the masked values for *RMDID2* procedures. The larger the intervals the more the observations is an outlier for the latter two methods. Originally published in Templ and Meindl (2008). Published with kind permission of ©Springer-Verlag Berlin Heidelberg 2008. All Rights Reserved

The methods just discussed are also available in **sdcMicro** as function `dRiskRMD()`. While `dRisk()` is automatically applied to objects of class `sdcMicroObj`, `dRiskRMD()` has to be called once to fill the corresponding slot:

```
sdc <- dRiskRMD(sdc)
```

The reason of not automatically estimating this risk is that robust estimations are computational expensive for large data sets since the estimation is based on a robust fit of a covariance matrix using the MCD-estimator. Whenever this risk measure is estimated, the results are saved in slot `risk$numericRMD` or can alternatively extracted using `get.sdcMicroObj(sdc,"risk")$numericRMD`.

References

Agresti, A. (2002). *Categorical data analysis* (2nd ed.). Wiley Series in Probability and Statistics Hoboken: Wiley-Interscience.

Benedetti, R., & Franconi, L. (1998). Statistical and technological solutions for controlled data dissemination. In *Pre-proceedings of New Techniques and Technologies for Statistics* (pp. 225–232).

Bethlehem, J. G., Keller, W. J., & Pannekoek, J. (1990). Disclosure control of microdata. *Journal of the American Statistical Association*, *85*(409), 38–45.

Capobianchi, A., Polettini, S., & Lucarelli M. (2001). Strategy for the implementation of individual risk methodology into μ-ARGUS. Technical report, Report for the CASC project. No: 1.2-D1.

Carlson, M. (2002a). Assessing microdata disclosure risk using the poisson-inverse gaussian distribution. *Statistics in Transition*, *5*, 901–925.

Carlson, M. (2002b). An empirical comparison of some methods for disclosure risk assessment. Technical Report, Stockholms Universitet.

Clogg, C. C., & Eliason S. R. (1987) Some common problems in log-linear analysis. *Sociological Methods & Research*, 8–44.

Elliot, M., & Manning, A. M. (2003). *Using dis to modify the classification of special uniques*. Luxembourg: In Joint UNECE/Eurostat work session on statistical data confidentiality.

Elliott, C., Skinner, C. J., & Dale, A. (1998). Special uniques, random uniques, and sticky populations: Some counterintuitive effects of geographical detail on disclosure risk. *Statistics: Research in Official*, 53–67.

Elliott, C., Manning, A. M., & Ford, R. W. (2002). A computational algorithm for handling the special uniques problem. *International Journal of Uncertainty, Fuzziness and Knowledge Based System*, *1*(1), 493–509.

Forster, J., & Webb, E. (2007). Bayesian disclosure risk assessment: Predicting small frequencies in contingency tables. Journal of the Royal Statistical Society. *Series C (Applied Statistics)*, *56*, 551–570.

Franconi, L., & Polettini, S. (2004). Individual risk estimation in μ-Argus: A review. In J. Domingo-Ferrer (Ed.), *Privacy in statistical databases*. Lecture Notes in Computer Science (pp. 262–272). Springer.

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, *47*(260), 663–685.

Hoshino, N., & Takemura, A. (1998). On the relation between logarithmic series model and other superpopulation models useful for microdata disclosure risk assessment.

Hundepool, A., Domingo-Ferrer, J., Franconi, L., Giessing, S., Nordholt, E. S., Spicer, K., et al. (2012). *Statistical disclosure control*. Wiley Series in Survey Methodology: Wiley. ISBN 9781118348222. https://books.google.at/books?id=BGa3zKkFm9oC.

Ichim, D. (2008). Extensions of the re-identification risk measures based on log-linear models. In J. Domingo-Ferrer & Y. Saygin (Eds.), *Privacy in statistical databases*. LNCS (vol. 5262, pp. 203–212). Heidelberg: Springer.

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida. *Journal of the American Statistical Association*, *84*, 414–420.

Kowarik, A., Templ, M., Meindl, B., & Fonteneau, F. (2013). *sdcMicroGUI: Graphical user interface for package sdcMicro*. R package version 1.1.1. http://CRAN.R-project.org/package=sdcMicroGUI.

Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, *1*(1). ISSN 1556-4681.10.1145/1217299.1217302.

Manning, A. M., Haglin, D. J., & Keane J. A. (2008). A recursive search algorithm for statistical disclosure assessment. *Data Mining and Knowledge Discovery*, *16*(2), 165–196. ISSN 1384-5810.10.1007/s10618-007-0078-6. http://dx.doi.org/10.1007/s10618-007-0078-6.

Maronna, R. A., Martin, R. D., & Yohai, V. J. (2006). *Robust statistics: Theory and methods*. New York: Wiley.

Mateo-Sanz, J. M., Sebe, F., & Domingo-Ferrer, J. (2004). Outlier protection in continuous microdata masking. *Privacy in statistical databases*. Lecture Notes in Computer Science (Vol. 3050, pp. 201–215). Springer.

Pagliuca, D., & Seri, G. (1999). Some results of individual ranking method on the system of enterprise accounts annual survey. Technical report, Esprit SDC Project, Boston. Deliverable MI-3/D2.

Polletini, S., & Seri, G. (2004). Guidelines for the protection of social micro-data using individual risk methodology. Deliverable no. 1.2-d3, CASC Project. http://neon.vb.cbs.nl/casc/.

Rinott, Y., & Shlomo, N. (2006). A generalized negative binomial smoothing model for sample disclosure risk estimation. In *Privacy in statistical databases*. Lecture Notes in Computer Science (pp. 82–93). Springer.

Samarati, P., & Sweeney, L. (1998). Protecting privacy when disclosing information: *k*-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, SRI International.

Samarati, P. (2001). Protecting respondents' identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering*, *13*(6), 1010–1027.

Skinner, C. J., & Holmes, D. J. (1998). Estimating the re-identification risk per record in microdata. *Journal of Official Statistics*, *14*, 361–372.

Skinner, C. J., & Shlomo, N. (2006). Assessing identification risk in survey microdata using log-linear models. S3ri methodology working papers, m06/14, University of Southampton, Southampton Statistical Sciences Research Institute.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(5), 557–570.

Templ, M., Meindl, B. (2008). Robust statistics meets SDC: New disclosure risk measures for continuous microdata masking. In *Privacy in Statistical Databases*. Lecture Notes in Computer Science (Vol. 5262, pp. 177–189). Springer.

Templ, M., Meindl, B., & Kowarik, A. (2015). Statistical disclosure control for micro-data using the R package sdcMicro. *Journal of Statistical Software*, *67*(1), 1–37.

Totter, M. (2015). Disclosure risk estimation for survey data. Master's thesis, Vienna University of Technology, Vienna, Austria. supervisor: M. Templ.

Willenborg, L., & De Waal, T. (2000). *Elements of statistical disclosure control*. ISBN: 0387951210.