Irving Fisher Committee on
Central Bank Statistics

BIS

IFC Satellite Seminar on "Granular data: new horizons and challenges for central banks"

Statistical Disclosure Control (SDC) for results derived from combined confidential microdata[1]

Jannick Blaschke, Christian Hirsch and Robin Kollmann,
Deutsche Bundesbank

[1] This contribution was prepared for the IFC Satellite Seminar held at the ISI 64th World Statistics Congress, co-organised with the Bank of Canada in Ottawa, Canada, on 15 July 2023. The views expressed are those of the authors and do not necessarily reflect the views of the Bank of Canada, the BIS, the IFC or the other central banks and institutions represented at the event.

Statistical Disclosure Control (SDC) for results derived from combined confidential microdata

Jannick Blaschke, Christian Hirsch, Robin Kollmann[1]

Abstract

Most researchers combine two or more datasets in their projects. In addition to technical feasibility and the meaningfulness of content, it is also necessary for researchers working with confidential Bundesbank microdata to consider possible consequences for Statistical Disclosure Control (SDC). This paper highlights three possible SDC challenges that may arise from appending and merging datasets: (1) loss of rows, (2) newly generated missing values in SDC variables, and (3) newly generated duplicates. It then presents possible solutions and provides simplified examples and rules of thumb.

Keywords: statistical disclosure control, sdc, output control, research data centre, rdc, appending, merging, joining

[1] The authors gratefully acknowledge the contributions from our colleagues at the RDSC and Matthias Gomolka. All views expressed in this report are the authors' personal views and do not necessarily reflect the views of the Deutsche Bundesbank or the Eurosystem. The paper was completed while Kollmann was an intern at the RDSC of Deutsche Bundesbank.

Contents

Statistical Disclosure Control (SDC) for results derived from combined confidential microdata 1

1. Introduction 3

2. Brief theoretical foundation 3

2.1 Appending 4

2.2 Merging 4

3. What hampers SDC? 6

3.1 Challenge 1: Loss of rows 7

3.2. Challenge 2: Missing values in SDC variables 8

3.3 Challenge 3: Duplicates 10

4 When does appending or merging hamper SDC? 11

4.1 Appending 11

4.2 Merging 12

5. A rule of thumb to resolve the SDC challenges 13

5.1 Appending 14

5.2 Merging 14

Glossary 16

References 17

1. Introduction

Most empirical research papers contain results based on multiple datasets. We distinguish two kinds of data operations in this paper. "Appending" datasets, i.e. extending a dataset with additional rows from another dataset, and "merging" datasets, i.e. combining columns from two datasets.

Whenever merging two datasets, researchers must consider not only technical feasibility (i.e. that there is at least one overlapping column in both datasets that can be used to identify the rows that should be combined), but also the meaningfulness of the resulting dataset. If, for example, we are merging two datasets with company information, we should first ask ourselves whether the companies in both datasets are defined in the same way. E.g. if one dataset contains information on the entire group and the second only on the German part, the company names may be the same, but caution is advised when interpreting the variables of the merged dataset.

In addition, when working at the Research Data and Service Centre (RDSC) of the Deutsche Bundesbank, all possible effects on Statistical Disclosure Control (SDC) must be kept in mind at all times[2]. Results that do not pass SDC cannot be taken out of the RDSC's secure environment or used in publications. Confidential microdata differ in this regard from other data sources where researchers can focus exclusively on obtaining results.

The purpose of this paper is to raise awareness among researchers that working with confidential microdata means focusing on two equally important objectives: (i) achieving interesting research findings while also (ii) checking that these results comply with SDC. Furthermore, these two objectives arise concurrently as researchers who only focus on results usually risk time-consuming retroactive adjustments to their code to ensure compliance.

This paper is structured as follows. In Section 2, we briefly introduce the concepts of appending and merging. Section 3 explains the three SDC challenges loss of rows, missing values in SDC variables and newly generated duplicates, and presents a solution for each challenge. Bringing together the two previous sections, Section 4 shows under which conditions, these challenges occur when appending or merging datasets. Section 5 concludes with some rules of thumb.[3]

2. Brief theoretical foundation

Before we take a detailed look at the challenges of combining different datasets, we distinguish the most important terminologies from each other. We are aware that the shape of the resulting dataset can be influenced by additional options in the append

[2] More information on the RDSC's SDC rules can be found in the the "*Rules for visiting researchers at the RDSC*" (Research Data and Service Centre (2021)), which are available at https://www.bundesbank.de/resource/blob/826176/ffc6337a19ea27359b06f2a8abe0ca7d/mL/2021-02-gastforschung-data.pdf

[3] The technical report available from the RDSC website also contains two appendices with further examples and a formal proof of the statements of Table 8.

or merge command[4]. However, for the sake of clarity, we assume that the respective commands are used in their simplest form.

2.1 Appending

When a dataset B is added from below to a dataset A, we speak of "appending" dataset A with dataset B (see Figure 1). For a successful append, the columns of interest should be included in both A and B. The shape of the resulting dataset C will be as follows:

- The number of rows will be the sum of the rows in A and B.
- The number of columns will be the union of the columns in A and B.

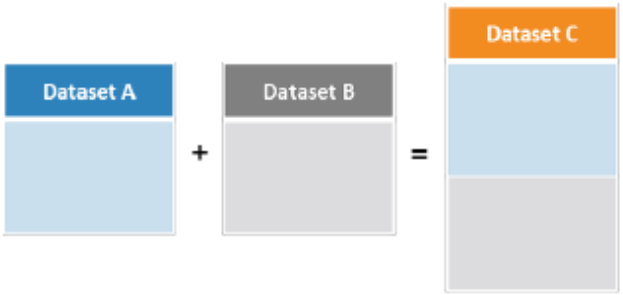


Figure 1: Appending two datasets A and B

2.2 Merging

The "merging" command can be used to combine the columns from two datasets to a single dataset (see Figure 2)[5]. As the datasets are combined sidewards, we often speak of a left and a right dataset After merging two datasets A and B, the shape of the resulting dataset C will be as follows:

- The number of rows will be smaller, equal or larger than the number of rows in A and B.
- The number of columns will be equal or larger than the number of columns in A and B.

[4] For example, Stata allows the "keep(varlist)" option in their append command, which allows the user to specify that the variables from the "using" dataset should be kept (StataCorp. (2021a)).

[5] Similar to merging, "joining"allows to combine the columns of two datasets. Depending on the programming language used, there can be slight differences between merge and join commands. In Python, for example, the method join() is used to combine two DataFrames based on their indexes whereas merge() requires specifying columns as a (merge) key (McKinney et al. (2010)). However, for the purpose of this paper, we will use merge as a synonym for join.

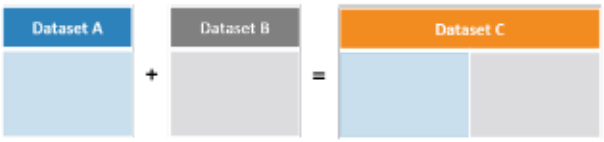


Figure 2: Merging two datasets A and B

Merges are usually done on indexes or key variables. When merging over an index, the first observation in A will be matched to the first observation in B, the second in A with the second in B, and so on. When merging on one (or multiple) key variables(s), this variable has to be included in both datasets so that the program can match similar values (e.g., if the key variable is a company ID, then information from A and B is merged per instance, i.e. per company, of this ID). If merging on key variables, the indexes will be ignored. For the remainder of this paper, we assume a merge on key variables.

Depending on the dataset structure, we distinguish between the following **relation types of the key variables**, which describe how the key variables of two datasets are related to each other when they are merged:

- **one-to-one (1:1)**: Each expression of the key variables used to merge occurs once in each of the two datasets.
- **Many-to-one (m:1)**: Each expression of the key variables used for the merge occurs 1-m times in the left dataset and - exactly once in the right dataset.
- **Many-to-many (m:m)**: Each expression of the key variables used for the merge occurs 1-m times in both datasets.
- **One-to-many (1:m)**: Each expression of the key variables used for the merge occurs exactly once in the left dataset and 1-m times in the right dataset.

In addition, we distinguish the following four merge types. They determine which rows will be kept and which ones will be dropped in the merged dataset:

- **Left outer merge**: Returns all rows from the left dataset and only the merged ones from the right dataset.
- **Right outer merge**: Returns all rows from the right dataset and only the merged ones from the left dataset.
- **Inner merge**: Returns only merged rows from both datasets.
- **Full (outer) merge**: Returns all rows from both datasets regardless if they were merged or not.

Figure 3 visualizes the four merge types using dataset A (left) and B (right) as an example:

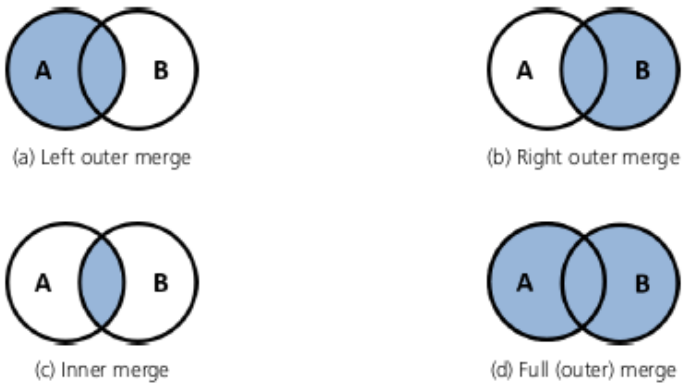


Figure 3: The four merge types

3. What hampers SDC?[6]

As part of their projects, researchers generate results based on confidential Bundesbank microdata. These could be descriptive statistics (e.g. a mean or a frequency table) as well as regressions. When performing SDC on such results, the dataset that was used to compute them (analysis dataset) ideally has the following properties:

1. Usually, results are not calculated directly on the unmodified Bundesbank data (original dataset), but may also be based on previously self-generated variables. Thus, for the SDC of results based on previously self-generated variables, it is imperative that all **rows** that were used for the generation of this variable are also still contained in the analysis dataset.
2. SDC variables do not contain **missing values**.
3. The dataset does not contain **duplicated observations**.

In reality, however, dataset transformations often lead to one or more of these assumptions being violated[7]. In this section, we show why a violation becomes a challenge for SDC and how researchers can address it. In Section 4, we explain whether and when this is the case when appending and merging two datasets.

[6] At this point, we would like to highlight that we try to describe SDC challenges in a short and concise way in this report. We explicitly point out that there will of course always be special cases for which the rules mentioned do not fully apply. We therefore strongly recommend thinking through the particular structure of the data provided by the RDSC and any changes made to it. All resulting SDC requirements should be kept in mind at all times.

[7] Even some of the datasets provided by the RDSC contain SDC variables with missing values or duplicated observations. This is not a sign of data quality issues but can have good reasons (e.g. missing values in the ID of a party that is only involved in certain transactions and therefore missing otherwise).

3.1 Challenge 1: Loss of rows

We describe this challenge and possible solutions in detail in a technical report (Blaschke et al. (2022)). Therefore, we will only give a brief overview in this paper.

Assume we have a dataset that contains company information with two SDC variables "Company_ID" and "Group_ID" (see Table 1a). For our analysis, we are only interested in the companies' groups rather than the individual companies. Therefore, we aggregate the data on the group level (see Table 1b). However, if we want to publish any results computed on "Group_Value", we cannot perform SDC proedures based on Table 1b as we no longer know the number of individual companies that belong to each group (see Principle O.1.3 "Adherence to minimum sample size" in Research Data and Service Centre (2021)) and we further cannot check for dominance (see Principle O.1.4, "Adherence to p% or dominance rule" in Research Data and Service Centre (2021)).

As described in Blaschke et al. (2022), we have to ensure that no SDC variables get lost when using a summary function on a dataset. Summary functions calculate results based on multiple values from multiple rows (e.g. an aggregation as in Table 1). Therefore, we have the following three options to ensure SDC compliance:

1. Perform a SDC before using the summary function and ensure that each row in the aggregated dataset will satisfy the SDC criteria.
2. Use summary functions only when taking into account the full set of SDC variables.
3. Do not drop the duplicated rows after using the summary function. Compute the results by using a dummy variable while performing SDC on the full dataset.

Company_ID	Group_ID	Value
111	AAA	1000
222	AAA	2000
333	AAA	1400
111	BBB	2100
111	CCC	100
111	DDD	100

(a) Company dataset before aggregation

Group_ID	Group_Value
AAA	4400
BBB	2100
CCC	100
DDD	100

(b) Company dataset after aggregation

Table 1: Loss of rows - Challenge

In our example, we choose the third option and define a dummy that is 1 each time "Group_ID" takes on a new value for the first time. Table 2 shows the result.

Company_ID	Group_ID	Value	Group_Value	Dummy
111	**AAA**	1000	**4400**	**1**
222	AAA	2000	4400	0
333	AAA	1400	4400	0
111	**BBB**	2100	**2100**	**1**
111	**CCC**	100	**100**	**1**
111	**DDD**	100	**100**	**1**

Table 2: Loss of rows - Solution

3.2. Challenge 2: Missing values in SDC variables

Assume we are using a dataset on transactions where company A is selling a product to company B and for some transactions, the sale is through an intermediary. In addition, the dataset also includes the ID of company A's group (ownership information). All four parties (companies A, B, the intermediary as well as company A's group) have to be protected in the SDC (i.e. "Company_A_ID", "Group_A_ID", "Inter_ID" and "Company_B_ID" in Table 3 are SDC variables). Tables 3a and 3b show two examples of this dataset with missing values in one of the SDC variables.

Let us assume, we would like to compute the sum of column "Value" in Table 3a and perform an SDC afterwards. The result shows that the aggregate would be based on six different IDs in "Company_A_ID", "Group_A_ID", and "Company_B_ID" and that the dominance criteria would not be violated[8]. However, we only have a single observation with an intermediary and thus, the overall SDC check would result in a disclosure problem[9].

Company_A_ID	Group_A_ID	Inter_ID	Company_B_ID	Value
F05	C		L01	1100
G02	D	Y	V73	20
J08	E		R99	9800
P11	F		C22	3200
Z21	G		K09	4000
L99	H		Q33	6600

(a) Missing ID values are uncorrleated with other SDC variables

Company_A_ID	Group_A_ID	Inter_ID	Company_B_ID	Value
F05	C	X	L01	1100
G02		Y	V73	20
J08	C	Y	R99	9800
P11	D	Z	C22	3200
Z21	E	X	K09	4000
L99		Y	Q33	6600

(b) Missing ID values are correlated with other SDC variables

Table 3: Newly generated missing values - Challenge

[8] Computation of dominance: (9800 + 6600)/(1100 + 20 + 9800 + 3200 + 4000 + 6600) = 66.34% < 85%.

[9] Computation of dominance: 20 / 20 = 100% > 85%.

Does this mean that it is generally very unlikely to get results through the SDC if a dataset contains a rarely filled SDC variable? This question should be answered from the context. To do so, RDSC researchers must proceed as follows:

1. Test whether the missing values of the SDC variable can be filled by interpolation. Some SDC variables are correlated with other SDC variables and can thus be adequately explained by them. In the example in Table 3b, the "Group_A_ID" is missing for some transactions. The reason for this is that not all companies may belong to a group. In these cases, the missing group IDs may be imputed from the related ID variable (here "Company_A_ID"). Please refer to the respective data report to understand whether this procedure can be applied to an SDC variable in a specific dataset.

2. If interpolation is not possible and the SDC variable with the missing values is not sufficiently correlated with any other SDC variable, the missing values can be chronologically numbered. As previously mentioned, not all transactions in Table 3a are processed via an intermediary. As "Inter_ID" cannot be imputed from any other SDC variable, all missing values in "Inter_ID" can be assigned to random ID values that are unique per missing value. To avoid any confusion with the real IDs, the following naming convention must be observed: "sdc_" + consecutive number (e.g. "sdc_15").

Table 4 shows how an SDC following the two approaches would look for the cases presented in Table 3. The two additional variables indicate the sample size ("SDC_N") and the dominance ("SDC_Dom") for "Group_A_ID"[10] and "Inter_ID"[11]:

					for "Group_A_ID"	
Company_A_ID	Group_A_ID	Inter_ID	Company_B_ID	Value	SDC_N	SDC_Dom
F05	C	X	L01	1100		
G02	**G02**	Y	V73	20		
J08	C	Y	R99	9800	6	70.79%
P11	D	Z	C22	3200		
Z21	E	X	K09	4000		
L99	**L99**	Y	Q33	6600		

(a) Interpolated missing IDs

					for "Inter_ID"	
Company_A_ID	Group_A_ID	Inter_ID	Company_B_ID	Value	SDC_N	SDC_Dom
F05	C	**sdc_1**	L01	1100		
G02	D	Y	V73	20		
J08	E	**sdc_3**	R99	9800	6	66.34%
P11	F	**sdc_4**	C22	3200		
Z21	G	**sdc_5**	K09	4000		
L99	H	**sdc_6**	Q33	6600		

(b) Consecutively numbered missing IDs

Table 4: Newly generated missing values - Solutions

[10] Computation of dominance: (1100 + 9800 + 6600) / (1100 + 20 + 9800 + 3200 + 4000+ 6600) = 70.79% <85%.

[11] Computation of dominance: (9800 + 6600) / (1100 + 20 + 9800 + 3200 + 4000 + 6600) = 66.34% < 85%.

3.3 Challenge 3: Duplicates

Table 5a shows a dataset on company groups. Since we are also interested in the associated values for the individual companies, we enrich this dataset with data from Table 5b. As some groups own more than one company, we use a 1:m inner merge[12]. Table 5c shows the merged dataset. Although there are no missing values or deleted rows, there are duplicates in the rows previously contained in Table 5a.

Group_ID	Group_Value
AAA	1000
BBB	2100

(a) Company group dataset

Group_ID	Company_ID	Company_Value
AAA	F22	40
AAA	G54	50
AAA	O99	60
AAA	W71	70
AAA	A01	80
BBB	J77	10
BBB	C30	20

(b) Company dataset

Group_ID	Group_Value	Company_ID	Company_Value
AAA	1000	F22	40
AAA	1000	G54	50
AAA	1000	O99	60
AAA	1000	W71	70
AAA	1000	A01	80
BBB	2100	J77	10
BBB	2100	C30	20

(c) Merged dataset with company and group information

Table 5: Newly generated duplicates - Challenge

If we now calculate the mean of the variable "Group_Value", we cannot simply divide the sum of all values in "Group_Value" by the number of rows in "Group_ID". Instead, we must find a way to account for the duplicates in "Group_ID" and "Group_Value" that were generated when merging. For this, we best follow the approach presented in *Challenge 1: Loss of rows* where we used a dummy variable (see Table 2). Hence, the mean of "Group_Value" would have to be (1000+2100)/2 = 1550. Table 6 shows the resulting dataset including the SDC variables for sample size ("SDC_N") and dominance ("SDC_Dom")[13] for the SDC variable "Group_ID":

[12] Note, that since all Group_IDs are contained in both datasets, the result would have been the same for a left and a right outer merge or a full (outer) merge.

[13] Computation of dominance: (1000 + 2100) / (1000 + 2100) = 100% > 85%.

					for "Group_ID"	
Group_ID	Group_Value	Company_ID	Company_Value	Dummy	SDC_N	SDC_Dom
AAA	**1000**	F22	40	**1**		
AAA	1000	G54	50	0		
AAA	1000	O99	60	0		
AAA	1000	W71	70	0	2	100%
AAA	1000	A01	80	0		
BBB	**2100**	J77	10	**1**		
BBB	2100	C30	20	0		

Table 6: Newly generated missing values - Solution

4 When does appending or merging hamper SDC?

Readers should differentiate between existing challenges in datasets and newly created challenges that appear because of appending or merging datasets. This section explains origins of the latter. For SDC all challenges need to be adressed irrespective of their origin.

4.1 Appending

A dataset C that is generated by appending two datasets A and B will always contain all columns and rows from A and B. Hence, no columns or rows get lost. However, appending can generate **duplicates** and **missing values**. Duplicates occur if at least one row exists in both A and B. Missing values are generated, if there is at least one column that is only included in A or in B:

- Columns only included in A will lead to missing values in all rows in C that came from B.
- Columns only in B will lead to missing values in all rows in C that came from A.

Table 7 shows, how data for a new month (Dataset B) are appended to an existing dataset (dataset A). The length of dataset C is the sum of the rows in datasets A and B while the number of columns in C is the union of A and B. Variable "Group_ID" was only included in dataset B and therefore, all previous months in C are empty in "Group_ID". Finally, note, that no observations from SDC variables ("Company_ID", "Group_ID") were dropped. If "Group_ID" is an SDC variable, we have to address this challenge as described in the previous section.

Date	Company_ID	Company_Value
2021-01	F55	10
2021-01	L91	20
2021-02	F55	20
2021-02	L91	30

(a) Dataset A

Date	Company_ID	Group_ID	Company_Value
2021-03	F55	AAA	10
2021-03	L91	BBB	20

(b) Dataset B

Date	Company_ID	Group_ID	Company_Value
2021-01	F55		10
2021-01	L91		20
2021-02	F55		20
2021-02	L91		30
2021-03	F55	AAA	10
2021-03	L91	BBB	20

(c) Dataset C

Table 7: Example for appending datasets A and B

If we generated missing values in "Company_Value" we must not fill them with a value (e.g. zeros) but preserve their character as missing. Otherwise, we could not distinguish between a reported value (even if zero) and a missing value that must not be included in the SDC computation.

4.2 Merging

Whether SDC challenges occur after merging datasets depends in the

1. Relation of the key variables (e.g. 1:m vs. 1:1)
2. Merge type (e.g. left outer merge vs. inner merge)
3. Content of the datasets (e.g. if both datasets contain the same codes in the merge key variable(s))

As we have these three dimensions determining whether SDC challenges have to be considered. Table 8 gives a first indication about possible challenges. Note that the table only shows merge-induced SDC challenges but does not say anything about pre-merge characteristics of the individual datasets (e.g. dataset A could already have had SDC variables with missing values before the merge). [14]

[14] The technical report available from the RDSC website contains a formal proof of the statements of Table 8 in an appendix.

	Left dataset (dataset A)			Right dataset (dataset B)		
	Deleted rows	Duplicates	Missing values	Deleted rows	Duplicates	Missing values
1:1	None	None	None	Non-merged rows	None	Possible
m:m		Possible			Possible	
1:m					None	
m:1		None			Possible	

(a) Left outer merge

	Left dataset (dataset A)			Right dataset (dataset B)		
	Deleted rows	Duplicates	Missing values	Deleted rows	Duplicates	Missing values
1:1	Non-merged rows	None	Possible	None	None	None
m:m		Possible			Possible	
1:m					None	
m:1		None			Possible	

(b) Right outer merge

	Left dataset (dataset A)			Right dataset (dataset B)		
	Deleted rows	Duplicates	Missing values	Deleted rows	Duplicates	Missing values
1:1	Non-merged rows	None	None	Non-merged rows	None	None
m:m		Possible			Possible	
1:m					None	
m:1		None			Possible	

(c) Inner merge

	Left dataset (dataset A)			Right dataset (dataset B)		
	Deleted rows	Duplicates	Missing values	Deleted rows	Duplicates	Missing values
1:1	None	None	Possible	None	None	Possible
m:m		Possible			Possible	
1:m					None	
m:1		None			Possible	

(d) Full (outer) merge

Table 8: Possible SDC challenges by merge type

For example, in case of a left outer 1:m merge, Table 8 (a) states that when merging A and B, we keep all rows that are either only in A or both in A and B, while we drop those only occurring in B. For those rows that only exist in A, missing values are generated in SDC variables only existing in B. As we are using a 1:m merge we know that each expression of the key variables used for the merge occurs exactly once in A and 1-m times in B. Therefore, duplicates in A will be generated for each expression of key variables that exists multiple times in B.

5. A rule of thumb to resolve the SDC challenges

This section concludes the main findings on addressing SDC challenges from appending and merging datasets. Note again, that SDC challenges might already exist in the original data as provided by the RDSC. They have to be addressed irrespective of their origin.

5.1 Appending

Appending two datasets usually does not lead to big challenges when performing SDC. However, you should keep in mind that you might generate missing values in SDC variables. Furthermore, you should make sure not to drop any SDC variables even when they are rarely filled and not the focus of your research.

5.2 Merging

It is important that researchers understand that the three SDC challenges are not equally serious. While an analysis dataset with missing values or duplicates can usually be accounted for during SDC, this is no longer possible when rows are missing which are needed for SDC. In the latter case, the previous program code would have to be reworked. Therefore, it can be said that **missing rows are by far the biggest SDC challenge**.

As a general rule, we learn from Table 8 that:

- Deleted rows and missing values depend on the merge type.
- Duplicates depend on the relation of the key variables.

Thus, we conclude the following rules of thumb:

- **Understand your data structure**. Key variables identify rows in datasets. The origin of many SDC challenges lies in the relation between these key variables in the original datasets. Once you are aware if you have e.g. a 1:m or a 1:1 relationship in your set of merge key variables, you can use Table 8 to understand which SDC challenges you are likely to encounter.
- **Understand the impact of your selection of merge key variables**. Merging datasets will never create new missing values in SDC variables that are in the set of merge key variables. However, sometimes (e.g. when the two datasets have different SDC variables) we cannot merge over all SDC variables in a dataset. This often happens when one of the datasets contains multiple parties (e.g. buyer and seller, or borrower and lender) or reflects ownership relations (e.g. parent and subsidiary). In these cases, we will most likely create new missing values in an SDC variable.
- **If you cannot reverse it try to avoid it**. As mentioned above deleted rows may not be reversed. When researchers decide to first compute a summary function based on multiple rows and then drop some of those rows, they should carefully plan in advance how to address the challenge of deleted rows. We strongly recommend to first merge the original datasets and then follow the approach from Blaschke et al. (2022).
- **Use full (outer) merges**. Except for full (outer) merges all merge type involve deleting rows from one or both datasets if key variables do not overlap perfectly.

- **Use 1:1 merges**. For all merge types, 1:1 merges perform best. However, we are well aware that 1:1 merges are rarely possible in reality. On the other end, m:m merges always perform worst and even in the Stata reference manual for the **merge** command, it says "*m:m specifies a many-to-many merge and is a bad idea. (...) If you think that you need an m:m merge, then you probably need to work with your data so that you can use a 1:m or m:1 merge.*" (StataCorp. (2021b)). By definition, 1:m (m:1) merges can duplicate a single row in the left (right) dataset up to m times.
- **Duplicated rows make untidy data**. Duplicated rows and tidy data make for a formidable trade-off (Wickham (2014)). A dataset is considered tidy if the same set of key variables identifies all variables in the dataset. By its very definition tidy datasets do not have duplicated rows. However, merging tidy datasets can lead to untidy datasets. For these cases, we recommend sticking with untidy dataset rather then trying to make them tidy again. Tidying a dataset almost always requires deleting rows, which may severely impair SDC.
- **Consider using dummy variables to navigate between different key variables.** Navigating untidy data for SDC requires understanding which key variables identify an observation (e.g. as done in Table 2). For these cases we recommend using a dummy variable to switch different key variables as described in Blaschke et al. (2022).

Glossary

- **Analysis dataset**. The "analysis dataset" is the dataset that is used to compute results. Normally, the analysis dataset differs from the original dataset (see below) due to data transformation steps (e.g. dropping of variables and rows, merging with other datasets, or generation of additional variables).
- **Key variables**. One or more variables (combined) enable the distinct identification of each cell of the other columns. As these variables are the key to identifying the other cells, we refer to them as "key variables". We refer to the key variables in their entirety as the "set of key variables". Database terminology may also call the set the "compound key" or "unit of observation". Merge key variables are key variables that are used to identify rows in a merge.
- **Original dataset**. The "original dataset" is the unmodified dataset as provided by the RDSC.
- **SDC variables**. Variables containing the IDs of sensitive entities are referred to as "SDC variables". As the classification of SDC variables is purely context-related and depends on whether the entities contained therein need to be protected or not, the RDSC specifies SDC variables for each original dataset in its data reports. These reports identify SDC variables as such for each individual dataset, which is why a variable may be an SDC variable for one original dataset but not for another.
- **Summary functions**. Some functions may calculate results based on multiple values from multiple rows, which leads to an aggregation of information. Examples include sums or means across several rows. We refer to these functions as "summary functions".

References

Blaschke, J., Gomolka, M., and Hirsch, C. (2022). *Statistical Disclosure Control (SDC) for results derived from aggregated confidential microdata, Technical Report 2022-01 – Version 1.0*. Deutsche Bundesbank, Research Data; Service Centre. Retrieved from https: //www.bundesbank.de/resource/blob/745168/3501912ca41dc894f605a9d24413dee1/mL/ 2022-01-sdc-data.pdf

McKinney, Wothers. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, *445*, 51–56. Austin, TX.

Research Data and Service Centre. (2021). *Rules for visiting researchers at the RDSC, Technical Report 2021-02 - Version 1-0*. Deutsche Bundesbank, Research Data; Service Centre. Retrieved from https://www.bundesbank.de/resource/blob/826176/ffc6337a19ea27359b06f2a8abe0ca7d/mL/2021-02-gastforschung-data.pdf

StataCorp. (2021a). *Stata 17 Base Reference Manual- append*. College Station, TX: Stata Press. Retrieved from https://www.stata.com/manuals/dappend.pdf

StataCorp. (2021b). *Stata 17 Base Reference Manual - merge*. College Station, TX: Stata Press. Retrieved from https://www.stata.com/manuals/dmerge.pdf

Wickham, H. (2014). Tidy data. *Journal of Statistical Software*, *59*(10), 1–23. https://doi.org/10. 18637/jss.v059.i10

DEUTSCHE BUNDESBANK EUROSYSTEM

Statistical Disclosure Control (SDC) for results derived from combined confidential microdata

Christian Hirsch and Jannick Blaschke, Deutsche Bundesbank, Research Data and Service Centre

Session 3b: Data governance and quality

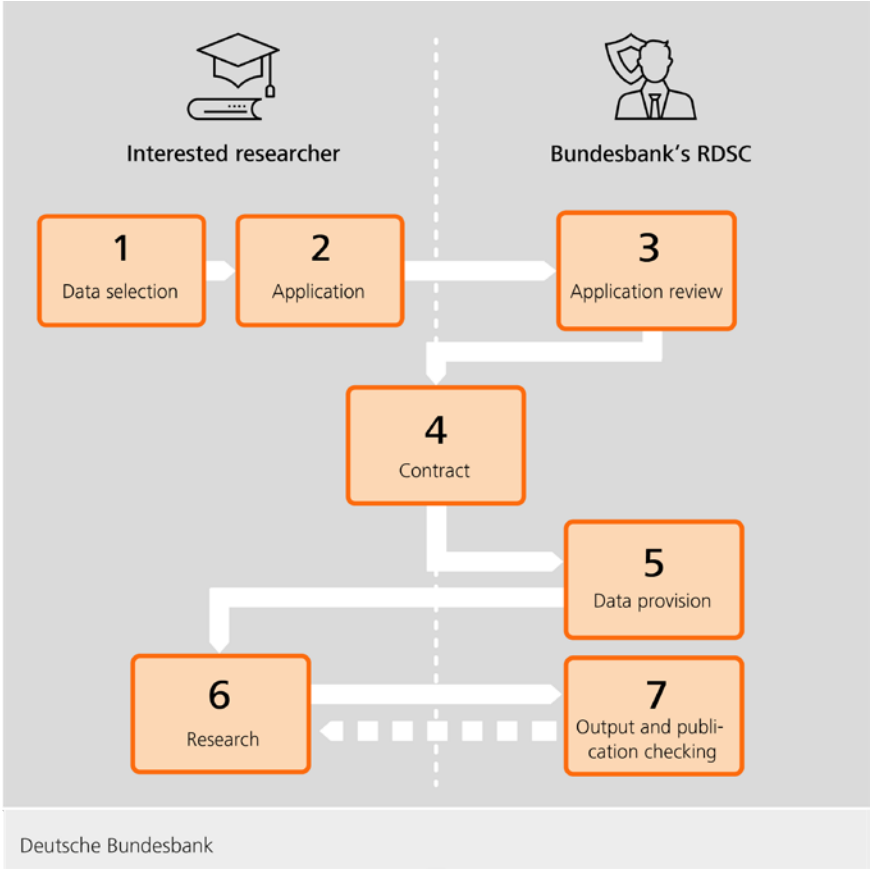
BoC-IFC Satellite Seminar on “Granular data: new horizons and challenges for central banks”

on 15 July 2023 in Ottawa, Canada

Research Data and Service Centre

1. Introducing the Research Data and Service Centre (RDSC)

The RDSC in a nutshell



The RDSC offers access for **non-commercial research** to (sensitive) **micro data** of the Bundesbank.

Currently, the RDSC counts around 300 active research projects

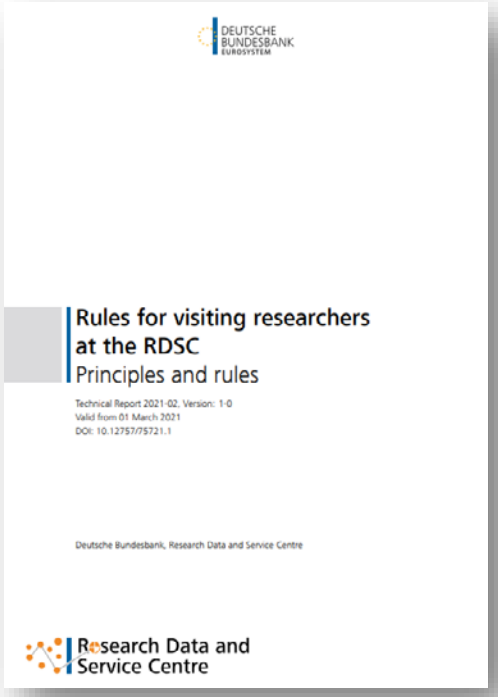
For more information please see the information on our **website**: https://www.bundesbank.de/en/bundesbank/research/rdsc/your-research-project-at-the-rdsc

2. Statistical Disclosure Control (SDC)
SDC principles and rules

All results that leave the RDSC must go through a **output control** that is conducted by the researcher and then **checked by two RDSC staff members**.

The two main rules are:

- **Minimum sample size**: Results intended for release must be based on at least **five** different observation units.
- **Dominance rule**: The combined shares of the two largest observation units should **not exceed 85%** of the total value under analysis.



All rules are outlined here:
Rules for visiting researchers: Principles and rules

2. Statistical Disclosure Control (SDC)

Example

	Company_ID	Company_Value
1	F22	40
2	G54	50
3	O99	60
4	W71	70
5	A01	80
6	J77	10
7	C30	20

The average of *Company_Value* is:

$$\frac{40 + 50 + 60 + 70 + 80 + 10 + 20}{7} = 47.1$$

Does this result comply with SDC?

Yes

- ✓ Based on 7 different companies
- ✓ Dominance criterion satisfied as $\frac{70+80}{40+50+60+70+80+10+20} \leq 0.85$

3. What hampers SDC?

Challenge 1: Loss of rows

Fixing a loss of rows might require **reworking previous code**

Company_ID	Group_ID	Value
111	AAA	1000
222	AAA	2000
333	AAA	1400
111	BBB	2100
111	CCC	100
111	DDD	100

(a) Company dataset before aggregation

Group_ID	Group_Value
AAA	4400
BBB	2100
CCC	100
DDD	100

(b) Company dataset after aggregation

- How many company IDs are behind each row?
- Is there a problem with dominance in *Company_ID*?

= SDC variable

3. What hampers SDC?

Challenge 2: Missing values in SDC variables

***Example**: Transaction between two companies with one intermediary party*

Company_A_ID	Group_A_ID	Inter_ID	Company_B_ID	Value
F05	C		L01	1100
G02	D	Y	V73	20
J08	E		R99	9800
P11	F		C22	3200
Z21	G		K09	4000
L99	H		Q33	6600

(a) Missing ID values are uncorrleated with other SDC variables

Company_A_ID	Group_A_ID	Inter_ID	Company_B_ID	Value
F05	C	X	L01	1100
G02		Y	V73	20
J08	C	Y	R99	9800
P11	D	Z	C22	3200
Z21	E	X	K09	4000
L99		Y	Q33	6600

(b) Missing ID values are correlated with other SDC variables

- How do we check the number of entities behind *Inter_ID* if some transactions simply do not have any intermediary party involved?
- How do we compute the dominance for *Inter_ID*?

= SDC variable

3. What hampers SDC?

Challenge 3: Duplicates

Group_ID	Group_Value
AAA	1000
BBB	2100

(a) Company group dataset

Group_ID	Company_ID	Company_Value
AAA	F22	40
AAA	G54	50
AAA	O99	60
AAA	W71	70
AAA	A01	80
BBB	J77	10
BBB	C30	20

(b) Company dataset

Group_ID	Group_Value	Company_ID	Company_Value
AAA	1000	F22	40
AAA	1000	G54	50
AAA	1000	O99	60
AAA	1000	W71	70
AAA	1000	A01	80
BBB	2100	J77	10
BBB	2100	C30	20

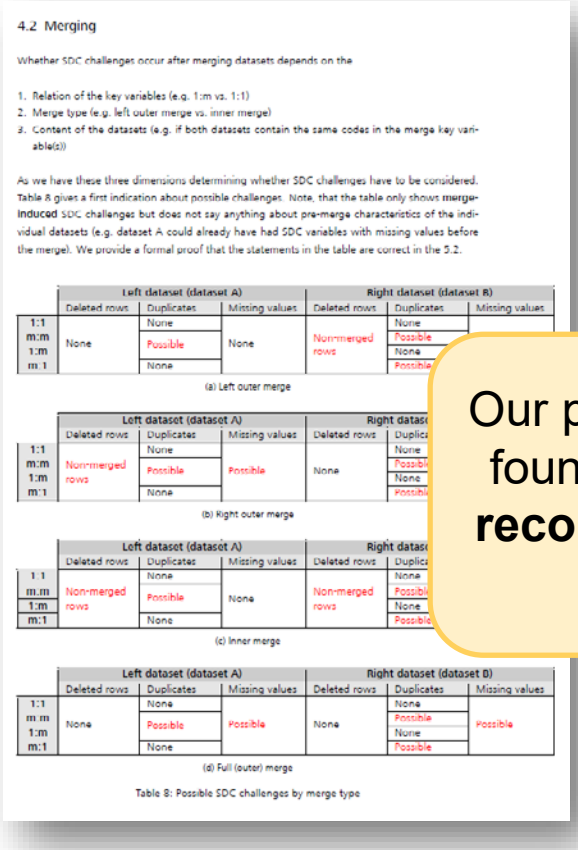
(c) Merged dataset with company and group information

When computing e.g. a mean on Group_Value, we should only consider rows 1 and 6. The SDC must do the same.

= SDC variable

4. Contribution of our paper

The challenges are deterministic but complicated…



4.2 Merging

Whether SDC challenges occur after merging datasets depends on the

1. Relation of the key variables (e.g. 1:m vs. 1:1)
2. Merge type (e.g. left outer merge vs. inner merge)
3. Content of the datasets (e.g. if both datasets contain the same codes in the merge key variable(s))

As we have these three dimensions determining whether SDC challenges have to be considered, Table 8 gives a first indication about possible challenges. Note, that the table only shows merge-induced SDC challenges but does not say anything about pre-merge characteristics of the individual datasets (e.g. dataset A could already have had SDC variables with missing values before the merge). We provide a formal proof that the statements in the table are correct in the 5.2.

	Left dataset (dataset A)			Right dataset (dataset B)		
	Deleted rows	Duplicates	Missing values	Deleted rows	Duplicates	Missing values
1:1	None	None	None	Non-merged rows	None	
m:m		Possible			Possible	
1:m		None			None	
m:1		None			Possible	

(a) Left outer merge

	Left dataset (dataset A)			Right dataset
	Deleted rows	Duplicates	Missing values	Deleted rows
1:1	Non-merged rows	None	Possible	None
m:m		Possible		
1:m		None		
m:1		None		

(b) Right outer merge

	Left dataset (dataset A)			Right dataset
	Deleted rows	Duplicates	Missing values	Deleted rows
1:1	Non-merged rows	None	None	Non-merged rows
m:m		Possible		
1:m		None		
m:1		None		

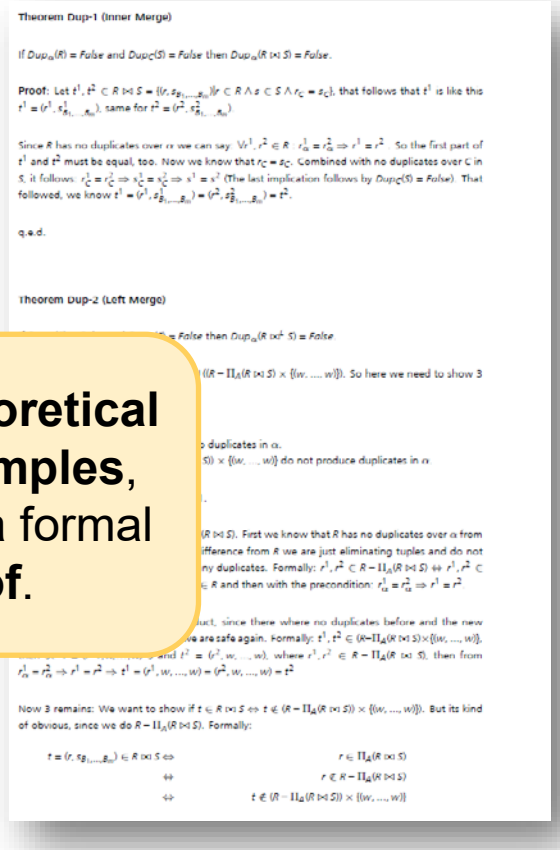
(c) Inner merge

	Left dataset (dataset A)			Right dataset (dataset B)		
	Deleted rows	Duplicates	Missing values	Deleted rows	Duplicates	Missing values
1:1	None	None	Possible	None	None	Possible
m:m		Possible			Possible	
1:m		None			None	
m:1		None			Possible	

(d) Full (outer) merge

Table 8: Possible SDC challenges by merge type

Our paper provides a **theoretical** foundation, detailed **examples**, **recommendations** and a formal **mathematical proof**.



Theorem Dup-1 (Inner Merge)

If $Dup_\alpha(R) = False$ and $Dup_C(S) = False$ then $Dup_\alpha(R \bowtie S) = False$.

Proof: Let $t^1, t^2 \subset R \bowtie S = \{(r, s_{\beta_1,\ldots,\beta_m}) | r \subset R \wedge s \subset S \wedge r_C = s_C\}$, that follows that t^1 is like this $t^1 = (r^1, s^1_{\beta_1,\ldots,\beta_m})$, same for $t^2 = (r^2, s^2_{\beta_1,\ldots,\beta_m})$.

Since R has no duplicates over α we can say: $\forall r^1, r^2 \in R: r^1_\alpha = r^2_\alpha \Rightarrow r^1 = r^2$. So the first part of t^1 and t^2 must be equal, too. Now we know that $r_C = s_C$. Combined with no duplicates over C in S, it follows: $r^1_C = r^2_C \Rightarrow s^1_C = s^2_C \Rightarrow s^1 = s^2$ (The last implication follows by $Dup_C(S) = False$). That followed, we know $t^1 = (r^1, s^1_{\beta_1,\ldots,\beta_m}) = (r^2, s^2_{\beta_1,\ldots,\beta_m}) = t^2$.

q.e.d.

Theorem Dup-2 (Left Merge)

$= False$ then $Dup_\alpha(R \bowtie S) = False$.

$((R - \Pi_A(R \bowtie S)) \times \{(w, \ldots, w)\})$. So here we need to show 3

duplicates in α.

$S)) \times \{(w, \ldots, w)\}$ do not produce duplicates in α.

$(R \bowtie S)$. First we know that R has no duplicates over α from ifference from R we are just eliminating tuples and do not ny duplicates. Formally: $r^1, r^2 \subset R - \Pi_A(R \bowtie S) \Leftrightarrow r^1, r^2 \subset$ $\in R$ and then with the precondition: $r^1_\alpha = r^2_\alpha \Rightarrow r^1 = r^2$.

uct, since there where no duplicates before and the new e are safe again. Formally: $t^1, t^2 \in (R - \Pi_A(R \bowtie S)) \times \{(w, \ldots, w)\}$, and $t^2 = (r^2, w, \ldots, w)$, where $r^1, r^2 \in R - \Pi_A(R \bowtie S)$, then from $r^1_\alpha = r^2_\alpha \Rightarrow r^1 = r^2 \Rightarrow t^1 = (r^1, w, \ldots, w) = (r^2, w, \ldots, w) = t^2$

Now 3 remains: We want to show if $t \in R \bowtie S \Leftrightarrow t \notin (R - \Pi_A(R \bowtie S)) \times \{(w, \ldots, w)\}$. But its kind of obvious, since we do $R - \Pi_A(R \bowtie S)$. Formally:

$$t = (r, s_{\beta_1,\ldots,\beta_m}) \in R \bowtie S \Leftrightarrow r \in \Pi_A(R \bowtie S)$$
$$\Leftrightarrow r \notin R - \Pi_A(R \bowtie S)$$
$$\Leftrightarrow t \notin (R - \Pi_A(R \bowtie S)) \times \{(w, \ldots, w)\}$$

4. Contribution of our paper
Merging

Loss of rows

Missing values in SDC variables

Duplicates

1. Understand your **data structure**
2. Understand the impact of your selection of merge **key variables**
3. If you cannot reverse it, try to **avoid** it
4. Use **(full) outer** merges
5. Use **1:1** merges
6. Understand that duplicated rows make **untidy data**
7. Consider using **dummy variables** to navigate between different key variables

4. Contribution of our paper

Merging - Example for solving a loss of rows

Company_ID	Group_ID	Value
111	AAA	1,000
222	AAA	2,000
333	AAA	1,400
111	BBB	2,100
111	CCC	100
555	DDD	200

m:1 inner merge via *Group_ID*

Group_ID	Group_Value
AAA	9,000
BBB	10,000
CCC	15,000

Company_ID	Group_ID	Value	Group_Value
111	**AAA**	1,000	**9,000**
111	**BBB**	2,100	**10,000**
111	**CCC**	100	**15,000**

Challenge 1: Loss of rows

- How many company *Company_IDs* are behind each row?
- Is there a problem with dominance in *Company_ID*?

= SDC variable

4. Contribution of our paper

Merging - Example for solving a loss of rows

Company_ID	Group_ID	Value
111	AAA	1,000
222	AAA	2,000
333	AAA	1,400
111	BBB	2,100
111	CCC	100
555	DDD	200

m:1 inner merge via *Group_ID*

Group_ID	Group_Value
AAA	9,000
BBB	10,000
CCC	15,000

Company_ID	Group_ID	Value	Group_Value	Dummy
111	**AAA**	1,000	**9,000**	**1**
222	AAA	2,000	9,000	0
333	AAA	1,400	9,000	0
111	**BBB**	2,100	**10,000**	**1**
111	**CCC**	100	**15,000**	**1**

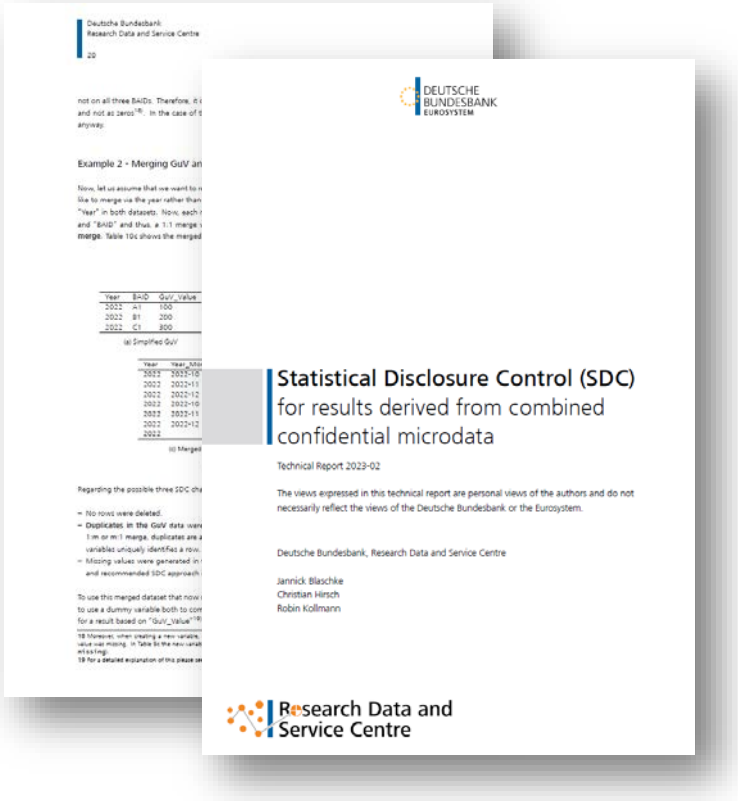
SDC for *Group_ID*:

X *based on only 3 different groups*

✓ Dominance criterion satisfied as $\frac{10000+15000}{3*9000+10000+15000} \leq 0.85$

= SDC variable

Conclusion and contact



The full report is available on our website:
https://preview.buba.de/blueprint/servlet/resource/blob/912040/ff1b34cc84c41541f65fc98aae431e7f/mL/2023-02-sdc-data.pdf

Thank you.

Research Data and Service Centre

Christian Hirsch
(christian.hirsch@bundesbank.de)
Jannick Blaschke
(jannick.blaschke@bundesbank.de)

Website: www.bundesbank.de/rdsc