Irving Fisher Committee
on Central Bank Statistics

IFC Working Papers

No 11

Optimizing Checking of Statistical Reports

by Peter Askjær Drejer

July 2013

BANK FOR INTERNATIONAL SETTLEMENTS

IFC Working Papers are written by the staff of member institutions of the Irving Fisher Committee on Central Bank Statistics, and from time to time by, or in cooperation with, economists and statisticians from other institutions. The views expressed in them are those of their authors and not necessarily the views of the IFC, its member institutions or the Bank for International Settlements.

This publication is available on the BIS website (www.bis.org).

ISBN 92-9197-948-1 (online)

Optimizing Checking of Statistical Reports

Conceptualized Experiences from a Central Bank

Peter Askjær Drejer[1]

Abstract

The aim of this paper is to convey experiences and progress of Danmarks Nationalbank during the recent years in optimizing the process of checking statistical reports. While the empirical background is derived from work within the interest rate statistics, the paper generalizes and categorizes our experiences in order to make them applicable to other statistical areas and other collection systems. The paper considers five different areas for optimization: "*Checking aggregates*", "*Outlier identification*", "*Workflow*", "*IT tools*" and "*Encouraging checks by the reporting entity*".

[1] Money and Banking Statistics, National Bank of Denmark, Havnegade 5, 1093 Copenhagen, Denmark, (e-mail: pad@nationalbanken.dk)

Contents

1. Introduction[2]

Recent years have seen significant developments in both technologies and statistical requirements. While innovations within IT facilitate compilers' task of checking incoming data, an increased demand for granular data and requirements for more timely data poses new challenges. In this light, it is worthwhile to reassess existing procedures. This paper aims at conveying some experiences and progress of Danmarks Nationalbank during the recent years in optimizing the checking process. While the empirical background is derived from work within the interest rate statistics, we have tried to generalize and categorize our experiences in order to make them generally applicable to other areas of statistics and other systems.

The purpose of the data checking process is to ensure that data reported are without substantial errors and that the compiler learns about the story behind important changes in reported figures. In general, the data cleansing process can be divided into two steps: First a data validation process followed by the process of plausibility testing.[3] Data validation rules are built into the reporting system and the data validation process will check whether the data pass or fail the validation rules. This process does not require analytical involvement of compilers. For this reason, the main focus of optimizations in this paper will be within the plausibility testing process.

The core of the plausibility testing process consists of identifying possible errors. Thus, a natural way of optimizing is to address the ability to identify outliers. At the same time, however, it is a fact that plausibility testing takes up significant resources at the compiling institution and potentially also for the statistical reporter. Hence, the quality of the checking process should be measured not just on how well errors are corrected, but also on the size of the burden it puts on the compilers and reporters.

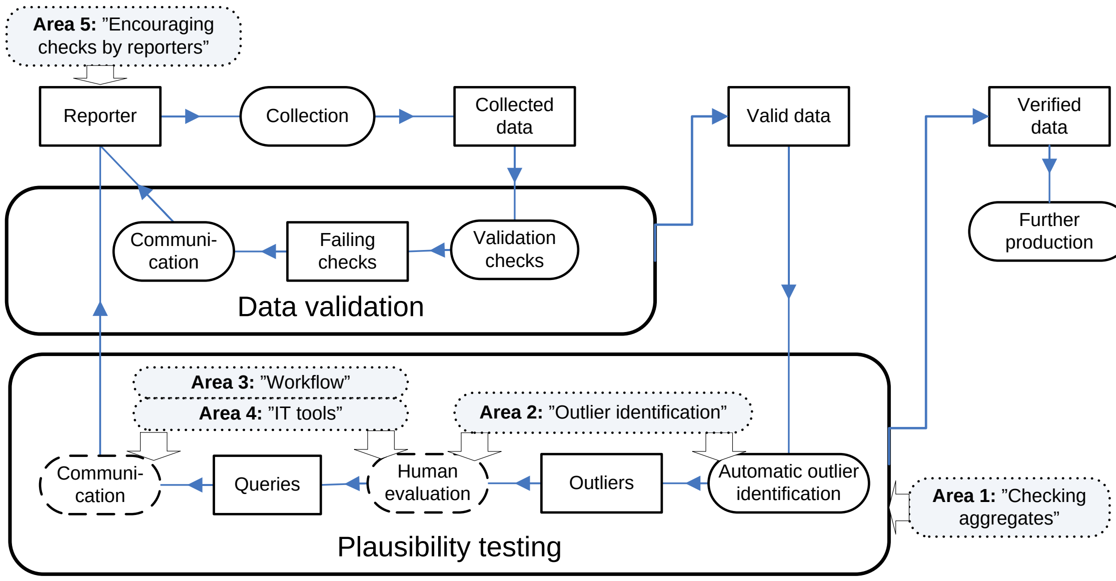
---

[2] The opinions expressed in this paper are those of the author and do not necessarily represent those of Danmarks Nationalbank. I thank Steen Ejerskov, Alexander Khalileev, Tue Mollerup Mathiasen, Mikkel Kragelund, Andreas Kuchler, Klaus Theill Jensen and Britta Gaarde for discussions and valuable comments and suggestions. I also thank the anonymous reviewers for valuable comments.

[3] The division into a data validation and a plausibility testing follows the terminology of IMF (2008), p. 31-33. Notice that this paper does not apply the subsequent division of the plausibility testing into three phases.

Chart 1

Optimization areas in relation to the checking process



Square boxes represent data. Boxes with round edges represent processes and boxes with dotted line and round edges represent processes with human involvement.

This paper describes five areas for optimization: *Checking aggregates*, *Outlier identification*, *Workflow*, *IT tools* and *Encouraging checks by the reporting entity.* The areas and their relation to a stylised checking process are depicted in Chart 1. Each area will be described in a separate section. This paper starts out with a section describing a standardized framework for defining a check. While such a framework at first may seem superfluous, it will be useful for understanding the subsequent sections.

In autumn 2011 a workshop was held, focussing on these five areas, with the participation of central banks and other compilers of banking statistics from around the world. As a follow-up, a survey was conducted on current practices and optimization efforts (see Drejer (2012)). All 30 institutions participating in the workshop answered the survey questionnaire. Results from this survey will be referred to in this paper.

2 A Framework for Defining a Check

The data checking process can be seen as made up of a number of checks that need to be performed. This section outlines a framework for describing any check performed on statistical reports. When check definitions become complex, their implementation and ongoing management will be greatly facilitated by such a framework. In general, it is our experience that the framework leads to a clearer discussion of optimization issues and facilitates the dialogue between data analysts and technical staff.

The framework will describe 5 steps that are needed in order to define a single check. While these steps will include notation for checks on aggregates (section 3), they will still be applicable to simpler checks without any aggregation. The framework definition will be illustrated using an example of checking loan data reported by a bank (see Chart 2). The same example will be continued under section 3.

Step 1: Check Item

A check will be done to verify a reported figure. This figure may be directly reported by the reporting entity or it may be indirectly reported, taking the form of an aggregate within a single report or between multiple reports. This figure will be termed the *check item*:

$$\text{Check item} = c$$

If the check is on an aggregated figure, then a *subcomponent* structure should be defined (see section 3).

$$\text{Subcomponent} = s_i \in S \text{ , } \qquad \text{where} \quad c \equiv \sum_i s_i$$

where s_i denotes each subcomponent and S is the total amount of subcomponents. In the example depicted in Chart 2, the check item is an aggregate being the total loans reported by the bank. The subcomponents of the check item are loans by sector. Another example of a check item could be on the directly reported figure for loans to Households without defining a subcomponent structure.

Step 2: Benchmark Data

The next step is to define which data should be used to verify the check item. For each check we will define a vector of benchmark data:

$$\text{Benchmark data} = \mathbf{b}$$

The benchmark data can come into use in all the subsequent steps. It could be used for transforming the check item (see next step), it could be used in the outlier identification function (step 4) or it can serve a more indirect purpose as background information for the human evaluation (step 5). In our example, we use the simple benchmark of last period's observations of the check item and subcomponents.

Chart 2

Example of step 1-3 of a check definition

Benchmark data ***Subcomponents***

Sector	t-1	t	Item growth	Growth contribution
MFIs	1.881	1.323	-30%	-2%
Public sector	1.909	2.555	34%	2%
Households	12.277	12.359	1%	0%
Non-financial corp.	5.784	6.944	20%	4%
Investment funds	181	720	298%	2%
Other financial inst.	7.679	12.258	60%	15%
Other	1.569	1.495	-5%	0%
Total loans	31.280	37.654	20%	

Subcomponent contributions

Check Item ***Check Item Transformation***

Step 3: Check Item Transformation

The next step is to define the basis for the outlier identification. This will be termed the *check item transformation* and can be seen as the function that transforms the check item into a testable measure:

$$\text{Check item transformation} = t = f(c, \mathbf{b})$$

If the check item is an aggregate with subcomponents, we will define *subcomponent contributions* to describe how much each subcomponent contributes to the check item transformation:

$$\text{Subcomponent contribution of } i = sc(i) = SC(c, s_i, \mathbf{b})$$

where *SC*(*) is the function that defines subcomponent contributions subject to the constraint $f(c, \mathbf{b}) = \sum_i sc_i$.

Growth rate calculations and seasonal adjustments are both examples of check item transformations. In our example, we use last period's observations (our benchmark) to create a check item transformation in the form of a growth rate (see Chart 2). Accordingly the subcomponent contributions are calculated as the growth rate contributions of each subcomponent.

Step 4: Automatic Outlier Identification

The next step is to establish boundaries for determining which values of the check item transformation should be returned as automatically detected outliers. This step requires a definition of the methodology behind the outlier identification. We will denominate the outlier function by $O(*,\mathbf{b})$. The input to the outlier function can both be the check item transformation and subcomponent contributions.

In the example of Chart 2, we can search for outliers among check item transformations $O(f(t,\mathbf{b}),b)$. The function could return the value 1 (outlier) if the total loan growth, $f(c,\mathbf{b})$, exceeds +/-15%, and otherwise 0 (no outlier). Another possibility is to search for outliers among subcomponents $O(sc_i,\mathbf{b})$.

Step 5: Human Evaluation

The last step will be the human evaluation of automatically identified outliers. For each check, we need to define which data the analyst performing this task should be presented with. This includes the definition of error text messages and graphs and table definitions. Finally it should be defined how the outlier should be presented to the reporter in case the analyst decides to make a query about the item.

Continuing our example of checking total loans, the presentation depends on whether in step 4 we identified the outlier at item or subcomponent level. Assuming the latter we could define that the analyst should be presented with a graph of the history of the outlier subcomponent and its contribution to the total growth. In addition there would probably be text explaining the outlier's origin.

3. Checking Aggregates

The concept of checking data at aggregated levels is now a well-known phenomenon that is being increasingly employed in statistical production. Van den Dool et al. (2008) described a macro-micro approach to compiling statistics that has been adopted by De Nederlandsche Bank. Among compilers of banking statistics, 87% used some kind of aggregated checking in 2011 (Drejer (2012)).

The example in Chart 2 illustrated how a basic check on an aggregate works. If our outlier search was based on the month-to-month growth of each individual sector item (the directly reported data) we would get a relatively large number of suspicious items and we would need to assess the validity of each outlier. Using aggregated checking techniques help us focus on important developments that affect the aggregate. In this instance, there was a relatively large aggregate increase of 20% which can easily be traced back to the responsible subcomponent (loans to "Other financial institutions").

In general, checking of aggregates can be applied to almost any kind of statistics. However, compared to more traditional checking approaches, it implies some design challenges. This section is focused on how to define an aggregated check. Related to this, section 5 will consider the implications to the workflow.

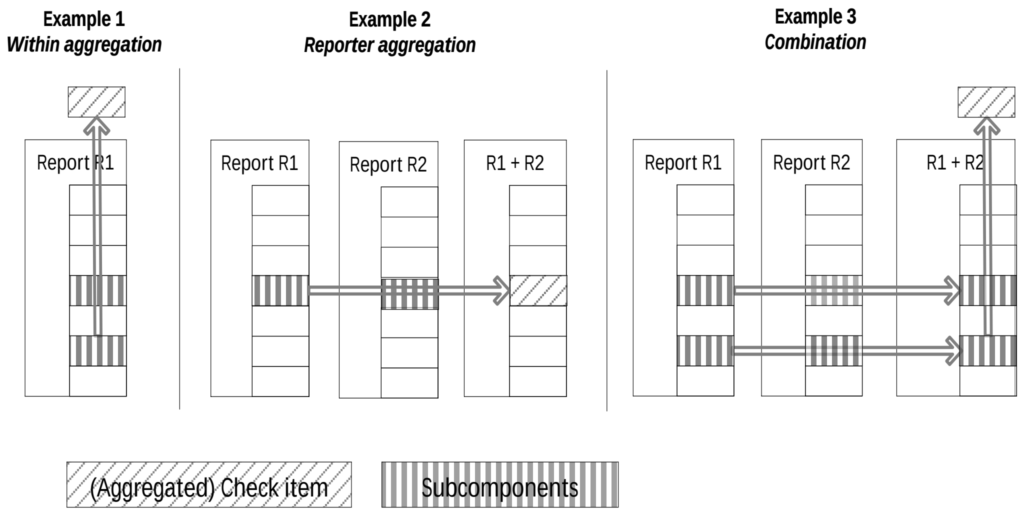
3.1 Type of Aggregation

Aggregated checking can be performed both within an individual report and by aggregating across individual reports. By checking aggregated figures within a report, our gain is that we do not have to check the most disaggregated figures reported. This type of check is illustrated in Example 1 of Chart 3 and will be referred to as a check *within aggregation*. On the other hand, by checking aggregates across individual reports, our gain is that we do not have to check each individual report, but only those contributing to the aggregate. This type of check is illustrated in Example 2 of Chart 3 and will be referred to as a check with *reporter aggregation.* It is possible to make checks that have both *within* and *reporter aggregation*. This is illustrated in Example 3 of Chart 3.

Notice that there can be intermediate steps of reporter aggregation, where not all reporters but only subgroups of reporters are aggregated. Also notice that a within aggregation can be both an explicitly reported aggregated figure and a calculated aggregation.

Chart 3

Different types of aggregation



3.2 Defining Subcomponents and Subcomponent Contributions

In most instances of aggregated checking, we want to be able to break down the aggregate in order to identify which item or which reporter affected the aggregate. In order to do this efficiently, it is important to define which *subcomponents* make up the aggregated figure and define how the contribution from each of these subcomponents should be calculated. When we define subcomponents to a check item, in effect what we are doing is defining to what detail we subsequently want answers. As outlined in the framework, this measure of how much a subcomponent affected the aggregate check item transformation will be termed *the subcomponent contribution*.

Notice that these subcomponent structures and contributions should be viewed conceptually and need not necessarily be performed computationally for each check. Depending on the level of checking, one may decide to make the actual calculation of subcomponent contributions conditional to developments in the check item (see 3.3.2).

3.2.1 Subcomponents

As outlined in the framework, the subcomponents make up the check item. The set of subcomponents, S, could be defined across multiple dimensions including the reporter dimension. As an example, we first consider a check item with pure reporter aggregation, say total loans to the household sector granted by banks. Here, our check item and subcomponents can be defined as:

$$\text{Check item} = c = \sum_i s_i \qquad \text{[E1]}$$

where s_i is the total loans to households of bank $i \in I$, where I is the total population of banks. We may want to be able to dig further into each bank's total household loans. This could be done by defining subcomponents that include loan types and maturity dimensions:

$$\text{Check item} = c = \sum_i \sum_l \sum_m s_{i,l,m}$$

where $s_{i,l,m}$ is the household loans of loan type $l \in L$, where L is the complete set of loan types, in maturity band category $m \in M$, where *M* constitutes all maturity band categories, of bank i.

3.2.2 Subcomponent Contributions

Once the subcomponents have been defined, their contributions to the check item transformation should be calculated. As outlined in step 3 of the general framework, this calculation is done by defining a function that defines each subcomponent's share of the check item transformation.

Continuing the example above where we were checking the amount of total loans we might have a check item transformation that gives the monthly growth rate of the aggregate. If we have the same subcomponent structure as in E1, the subcomponent contributions would be defined as each reporter's growth rate contribution:

$$sc(i) = \frac{\Delta s_{t,i}}{\sum_{i=0}^{I} s_{t-1,i}} \qquad \text{[E2]}$$

3.2.3 Using Subcomponent Contributions as Analytical Tools

Besides its use in the checking process, the breakdown of aggregated level figures is often a useful analytical tool to explain what drives changes in the overall statistics. An example of this is the ECB's publishing of a breakdown of euro area interest level developments into weight and interest rate effects.

3.3 Approaches to Checking Aggregates

3.3.1 Four Ways of Checking Aggregates

When we perform the outlier identification for checks involving aggregates, we can search both at the aggregated check item level and at the subcomponent level, corresponding to using either the check item transformations or the subcomponent contributions as input for the outlier identification. By combining the different types of aggregation and the two possible levels of outlier identification, we can identify four categories of checking based on aggregates (see Chart 4). Notice that Chart 4 only depicts pure aggregation forms. As shown in Chart 3, there can also be combinations of reporter and within aggregation.

The term “macro checking” seems mainly to have been used for Method 1 for cases with reporter aggregation and outlier identification at the check item level. In many cases, it will be useful to consider the whole palette of possibilities.

Chart 4

Four methods of checking aggregates

	Reporter aggregation	*Within aggregation*
Outlier identification at *check item level*	Method 1	Method 2
Outlier identification at *subcomponent level*	Method 3	Method 4

3.3.2 Choice of Test Level

In the example used above of checking aggregated household loans across institutions, we had a check item transformation being the monthly change and subcomponent contributions defined as monthly growth rate contributions (as in E2). In the search for outliers, we either consider aggregated growth rate, $O_1(t)$, or the growth rate contributions, $O_2(sc_i)$. In the first case, the outlier is found at the aggregated level and the subcomponent contributions are used as supplementary information for the subsequent evaluation of the outlier. In the second case, the check will directly identify outliers on the subcomponent contributions. Both methods have advantages and disadvantages.

An advantage of focusing on the outlier identification on the aggregated level is that it helps to prioritize which suspicious subcomponent should be addressed first. This is in particular an advantage when considering reporter aggregates.

According to Drejer (2012), 77% of compilers using checks on aggregates perform the outlier search on the aggregated check item level. In many cases, however, it is worthwhile to consider doing the search at a subcomponent level. It is clear that the searching of outliers among subcomponent contributions potentially leads to a more direct and faster identification which is an advantage in the human evaluation process. In addition, this method has some important advantages when considering the overall workflow (see section 5 for a discussion of this issue).

An important difference between the two approaches is whether similar-sized movements in opposite direction within subcomponents will be identified as outliers. When focusing at the aggregate such movements will cancel out and no outlier be identified. On the other hand, when focusing on the subcomponents, such movements will be detected. Whether or not detecting these movements is preferable will depend on the nature of the reported data. In some instances, part of the rationale behind doing checks at the aggregated level is that offsetting movements should indeed not be considered. Consider for example a case where the check item is the profit of a firm and we define the subcomponents as the main items of the profit/loss account. Assume that since last report, this firm has expanded its operations and is reporting correctly. If we focus our check on subcomponent contributions we will probably get both turnover and cost as outliers. On the other hand, if we had focused on the check item (profit), we would have had a more normal development and probably no outlier, which in this case probably would be preferable.

It is worth noting that the two approaches can coexist. We may want to search for outliers at subcomponent level because it gives us a direct identification; however, in light of the above example, we may want to make this search conditional on the aggregated level check. This could be implemented using a logic relation like (if $O_1(t)$ is true then $O_2(sc_i)$).

3.3.3 Using Different Levels of Reporter Aggregation

There can be multiple uses of checking at reporter aggregation level. If the check item is based on the input of the entire reporting population, the purpose of the check is to address implications for the final statistics. Construction of check items at a lower level, however, can serve to ensure a higher quality at the reporter level. Often a large part of the reporting population will be small reporters. The contributions of other reporters to the total aggregate will probably be negligible and even drastic outliers at the individual level will not affect the total. This would be detected if we run checks on each reporter, but then we would have to check each one. By constructing subpopulations of reporters that normally would be assumed to have contributions of similar size, the micro level errors could to a much higher degree be detected in the macro checks. Hence, checks on aggregations of subpopulations can be a useful tool that can help more efficiently to ensure data quality at the reporter level. Another application of sublevel reporter aggregations is when the statistics are divided into different strata and we want to ensure quality at the strata level.

3.3.4 Data Availability

One major issue of performing checks on reporter aggregates is the limitation that all reports, that will contribute to the final aggregate, needs to be available before the checking can start. As this will not always necessarily be the case, it is preferable to have a procedure by which a preliminary "stand in" report can be generated.

3.4 Examples of Checks on Aggregates

3.4.1 Banks' Reporting of Loans and Choice of Subcomponent Structure

To illustrate various definitions of subcomponent structures, we expand the example from the section outlining the framework (see Chart 2) and consider a slightly more elaborate subcomponent structure (see Chart 5). In this example, our check item is still total loans and the new and more detailed subcomponent structure now includes a breakdown of counterparty residence country. As benchmark data, we still use last period's reported loans, and the check item transformation will be the change in the check item. The subcomponent contributions will again be the growth contributions.

The example shows that the expanded subcomponent structure allows us to pin down the abnormal development at a more detailed level. While the more detailed subcomponent structure may generally seem preferable to the simpler one used in Chart 2, the preferable level of subcomponents need not always be the lowest. Specifically, when reports are very granulated, too much detail in the subcomponent structure may potentially lead to a huge amount of calculations and extend the work of the analyst towards too much detail. Also, if the evaluation of outliers is done in a standardized IT system, it may be preferable that checks have the same number or a maximum number of subcomponents.

Chart 5

Examples of expanded subcomponent structures for within aggregation

Subcomponents

Sector	Country	t-1	t	Item growth	Growth contribution
MFIs	Domestic	560	600	7%	0%
	Foreign	1.321	723	-45%	-2%
Public sector	Domestic	700	1.615	131%	3%
	Foreign	1.209	940	-22%	-1%
Households	Domestic	12.277	12.359	1%	0%
Non-financial corp.	Domestic	4.825	4.956	3%	0%
	Foreign	959	1.988	107%	3%
Investment funds	Domestic	111	709	539%	2%
	Foreign	70	11	-84%	0%
Other financial inst.	Domestic	6.224	10.797	73%	15%
	Foreign	1.455	1.461	0%	0%
Other	Domestic	1.569	1.495	-5%	0%
Total loans		31.280	37.654	20%	

Subcomponent contributions

Check Item

Check Item Transformation

The example may also illustrate aspects of the choice of approach towards search for outliers in aggregated checking. When the outlier identification considers the overall check item transformation (Method 2 in Chart 4), the considered measure is 20%. If this is found to be an outlier, we proceed to evaluate each subcomponent contribution. Alternatively, the outlier identification could occur directly at the subcomponent contributions (Method 4 in Chart 4), where the contribution of 15% would probably have been singled out as an outlier. It is apparent that the latter method in this case leads to a more direct identification of the outlier.

3.4.2 Interest Rate Statistics Implementation

Since 2004, we have based our checks of the interest rate statistics on reporter aggregates and have found that it significantly increases both the efficiency of the production and the quality of the statistics.

For the interest rate statistics, the check item will generally be an aggregated weighted interest rate and in the checking process we want to evaluate how it changed, thus the check item transformation is the change in the aggregate interest rate. In defining subcomponents and their contributions, we will first consider the case for pure reporter aggregation.

Since the check item is a weighted sum, finding subcomponent contributions is less straightforward than when the check item transformation is a simple sum, as in the above example. A way of decomposing a change in a weighted sum is by using the extended Marshall-Edgeworth decomposition (see Huerga and Steklacova (2008)):

$$\Delta I_t = I_t - I_{t-1} = \sum_{k=0}^{K} w_{k,t} \cdot i_{k,t} - \sum_{k=0}^{K} w_{k,t-1} \cdot i_{k,t-1}$$

$$= \sum_{k=0}^{K} \Delta i \cdot \frac{(w_{k,t} + w_{k,t-1})}{2} + \sum_{k=0}^{K} \Delta w_k \cdot \left[\frac{(i_{k,t} - I_t) + (i_{k,t-1} - I_{t-1})}{2} \right]$$

where I_t denotes the aggregated interest rate in period t, w_t is the weight for subcomponent k in period t and i_t is the interest rate of subcomponent k in period t. There is a total of *K* subcomponents. This formula decomposes the change in the aggregate interest rate into a sum of individual interest rate effects (first term on the right hand side) and a sum of individual weight effects (second term). The intuition behind the decomposition is as follows: the interest rate change of an individual reporter affects the aggregate change proportional to the average weighting over the two periods. A change in the weight affects the aggregate interest in a slightly more subtle way; here the effect depends on how far away the interest rate of the individual reporter was from the aggregated interest on average.

An important feature of this decomposition is that it isolates the contribution of each individual reporter to the aggregate whereby we can define our subcomponent contribution:

$$sc(k) = \Delta i \cdot \frac{(w_{k,t} + w_{k,t-1})}{2} + \Delta w_k \cdot \left[\frac{(i_{k,t} - I_t) + (i_{k,t-1} - I_{t-1})}{2} \right]$$

where *k* denotes the individual reporter. The subcomponent contributions can be evaluated either as a total or by the decomposed interest rate effect and weight effect.

The above formula is based on reporter aggregation of K reporters. It is straightforward to substitute individual institutions with *within aggregation* over J items. In this case, we could for example explain the development in the overall interest rate due to interest and weight changes in subcomponents (substituting J for K and w_j for w_k and i_j for i_k).

4. Outlier Identification

The process of identifying outliers will generally consist of two phases: 1) a computer searches for likely errors (outliers) given pre-specified algorithms, and 2) the automatically identified outliers are investigated by an analyst who decides whether the outliers were actually likely errors that should lead to further questions to the reporter.

In section 4.1 we consider the different sources of information that can be used as foundation for evaluating outliers. This benchmark information can both be used in the transformation of the check item, as part of the outlier determination or as supplementary information in the human evaluation. Hereafter, section 4.2 considers the issue of automatic outlier determination.

The human evaluation of outliers is covered by both section 5 and 6. At this point, it is worth noting that automatically identified outliers will in general have to undergo human evaluation and that this work will be roughly proportional to the number of outliers. For this reason, while we of course want to minimize the amount of undiscovered errors (false negatives), we also should pay attention to minimizing the amount of items identified as outliers that are not errors (false positives).

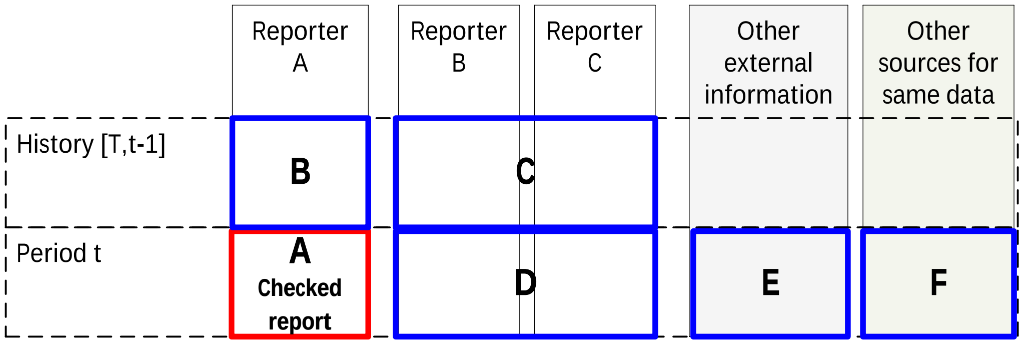
It should also be noted that sophisticated methods, both within benchmark data retrieval and outlier detection, can be complex to implement and may require substantial resources to maintain. In addition, if the process leading to an outlier classification is complex, it may not easily be understood by the analyst. Hence, one should make a methodological cost/benefit analysis before developing systems.

4.1 Available Benchmark Information

Various forms of information can be exploited for benchmarking data. Compilers traditionally have a tendency to rely entirely on the history of the check item itself. However, there are several other possibilities that will often be able to provide valuable benchmarks. Below is a description of areas that could provide reasonable benchmarks (see Chart 6). The extent to which different types of benchmark data is used by compilers are given in Chart 7.

Chart 6

Potential sources for benchmark data



4.1.1 *Other data in the same report (A)*

When checking one part of a report, other parts of it may often be useful as benchmarks. An example: If a report contains a firm's profit-loss accounts, we can assume a positive correlation between reported turnover and reported expenses. When using this kind of benchmark data, aggregated check methodologies will often be preferable.

4.1.2 *Report History (B)*

The history of a report can be used to establish some statistical features of the underlying data generating process. The most frequently used benchmark is probably last period's observation of the check item that is being used to construct the check item transformation of an absolute movement or percentage movement. In addition, the history of these changes could be used as a basis for establishing boundaries for what would be an outlier.

4.1.3 *General Reporter History (C)*

If there are similarities between the ways data are generated across reporters, then other reporters' historical reports for a specific item could be used to establish general features of an item applicable to all or subgroups. In addition, this kind of data will often have a panel structure that may be exploited for statistical inference procedures.

4.1.4 *Contemporaneously Incoming Reports (D)*

If the reporter's business can be assumed to be correlated, then contemporaneously incoming reports could contain valuable information about the normal development for a given item for the current reporting period. As such figures are preliminary and still potentially contain errors, it is important that any benchmarking deducted from these will not be sensitive to outliers (e.g. by using medians, trimmed means, etc.).

4.1.5 *External Sources (E)*

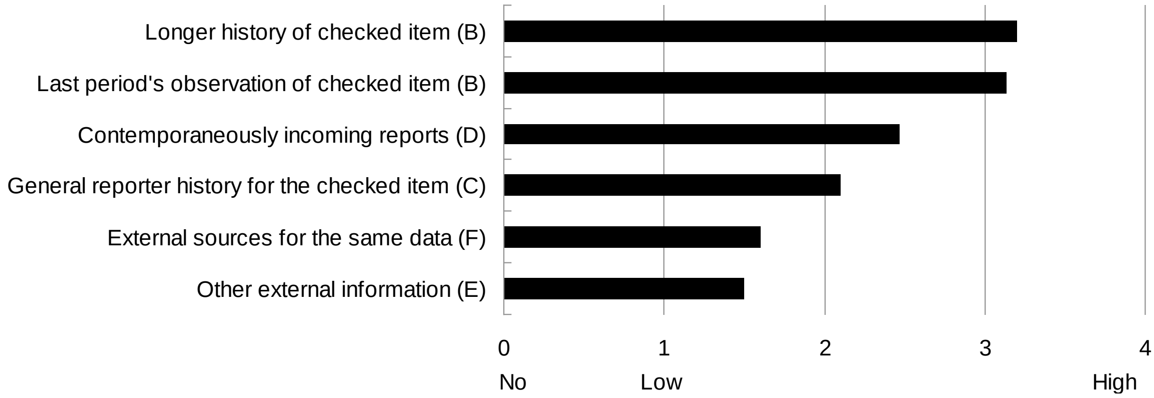
Other sources of information may be correlated with the information reported. For example, collected data on banks may be related to financial market data.

4.1.6 Sources for the Same Data (F)

If some kind of parallel reporting exists for the figures reported, this can be used as a checkup. An example of this is financial reports as benchmarks for collected balance sheet data.

Chart 7

Extent to which sources for benchmark data is used



Source: Drejer (2012)

4.2 Automatic Outlier Determination

The task of identifying outliers can be broken down into two steps: 1) we need a benchmark value, and 2) we need to set some confidence intervals around this in order to determine whether deviations from the benchmark are unusual. Using the notation from step 4 of the framework description in section 2, this amounts to defining the function $O(*,\mathbf{b}_t)$. This can be done either by trying to model the underlying statistical distribution or by applying more simple rules. Among compilers of banking statistics, simple rules were the predominant way of identifying outliers, while other statistical modelling was generally not used or only used to a very low extent (Drejer (2012)).

4.2.1 Heuristic Rules

Benchmarks and outlier bands can be set without the use of statistical measures. If, for example, the benchmark data is last period's observation and the check item transformation is the percentage movement, we can define a fixed percentage threshold that defines an outlier. While this approach risks having many false positives, it may be preferable for its simplicity and provide analysts confidence in knowing that everything that changed by more than x% compared to last period has been checked.

4.2.2 Statistical Model-based Determination

When using statistical methods, we try to model the underlying probability function behind the check item transformation in order to determine the expected mean and establish likelihood bands for this. Based on this statistical modelling, we can determine outliers in terms of likelihood of occurrence. There are numerous ways of statistically modelling distributions assuming various models for the data generating process. Some main assumptions are the definition of univariate or multivariate relationship and whether there is a time series model aspect. For an application with time series methodology to outlier identification (see Caporello and Maravall (2003)).

4.2.3 Machine Learning

The human evaluation basically consists of presenting the analyst with an outlier together with a fixed set of additional information and the analyst making a yes/no decision. If the outcome of this process is stored, the compiling institution will gradually build up a set of data containing the decisions of analysts. This kind of data set can be used to train algorithms to perform the yes/no decision, see for example Kotsiantis et al (2006). Such algorithms may aid or even replace the human evaluation.

4.3 Examples

4.3.1 Benchmark Data

For the interest rate statistics we have carried out empirical tests of numerous alternatives in order to find the optimal benchmark (see Bartmann and Drejer (2011)). On this basis, we found that the optimal source for retrieving benchmark data generally is contemporaneously incoming reports.

4.3.2 Defining Outliers by Empirical Distributions

In the interest rate statistics, we perform a check on each interest rate, where the check item is the change compared to last period, normalised by a benchmark change. The benchmark change is the median of contemporaneous reports for the same items.

In order to find outliers, we want to describe the statistical features of the check item transformation. Since the check item transformation is benchmarked, we do not expect deviations to vary over time and, in addition, we expect the distribution of data to follow the same distribution across reporters. On this background, we pool historical check item transformations over time and reporters in order to derive empirical distribution.[4] In this process, we take the intermediate step of pooling together items with similar distributional features in order to derive at smoother distributions. Chart 8 shows the lower and upper tails of three of these distributions described by selected percentiles for the check item transformation.

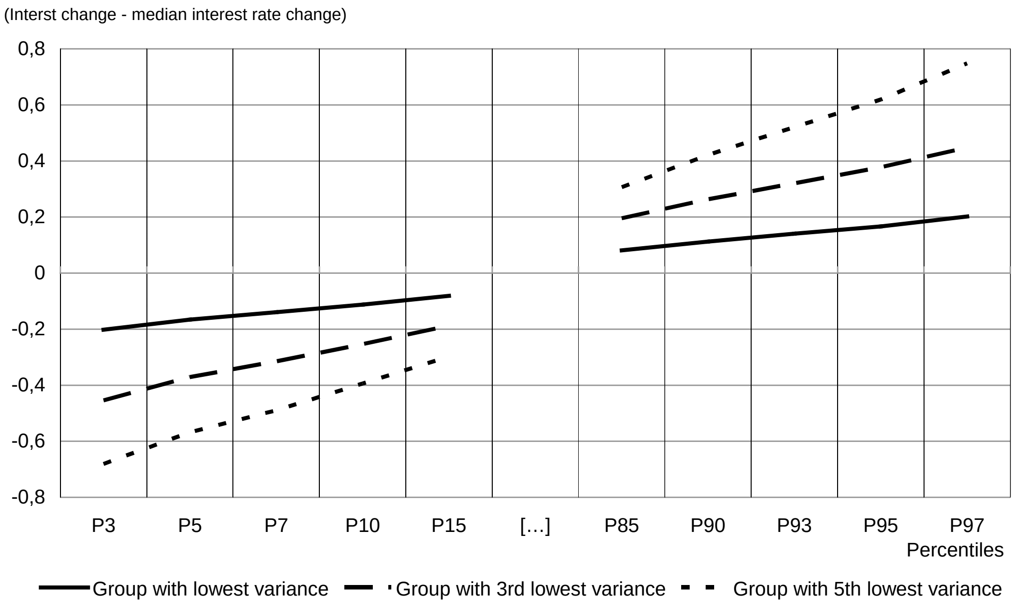
Since we have the complete distributional features of the check item transformation, we can set any level of confidence threshold. For each item, the analyst decides the importance and sets the confidence level accordingly. For example, interest rate changes on the overall interest rate for loans to households is of high priority, and accordingly, we want to examine approximately 15% of all observation, which corresponds to setting the lower and upper limits to the 7th and 93rd percentiles of the empirical distribution. Since this item belongs to the lowest variance group, its distribution is given by the blue line in Chart 8. This means that an outlier is identified if the deviation of the interest rate change in relation to the median interest rate change is below -0,14 pp. or above 0,14 pp. The selection is pre-programmed which means that the analyst will not have to worry about which level to check.

---

[4] The check item transformations are taken from the first received reports, still including errors.

Chart 8

Empirical distributions for the interest rate change (check item transformation)



5. Workflow

The optimal solution on how the workflow should be organized within a statistical production process, across different statistics and across collecting institutions, depends on factors that are specific to each statistics or compiling institution. Existing IT systems, quality standards, the division of receiving and checking data between departments, etc., will all imply limits for how the workflow can be arranged. However, some considerations will apply generally.

Mapping existing workflows in process diagrams is the foundation of finding ways to optimize the overall workflow. The processes mapped can be anything from the overall flow of data between departments and down to the specific tasks that an analyst performs when checking incoming reports. Often it makes sense to start at the very top level, as exemplified in Chart 1, and from there further expand individual processes.

5.1 Basic Workflows of Data Checking

Plausibility checking can involve different kinds of checks/tasks, which have various kinds of standardization and workflow. Since the production of statistics spans a limited number of days, there is a particular need for standardization and efficiency in the plausibility testing that takes place as part of the production. One way of dealing with this problem is to focus the production checks on *changes*, while the more elaborate checks, ensuring that the *levels* of the statistics are plausible, are done on an ad hoc basis between productions. Below are described 4 different kinds of workflows that can take place as part of the plausibility testing, each of which can be the subject of optimization.

5.1.1 Standardized Plausibility Checking of Reports (Production)

These are standardized checks performed during every production. Each check follows a definition as given in the framework section. In order to make the evaluation of outliers efficient the checking process should also be standardized.

5.1.2 Follow-up Evaluation (Production)

Following the initial checking of reports, reporters will often send revisions of their reports. These revisions have to be evaluated in terms of how well the problems have been addressed: Which items have changed since the last version? Did new errors occur? How were the questions addressed in the data? And how did the revision affect the aggregated picture? In order to answer these questions many different kinds of data are needed and if the workflow surrounding this process is not very well-organized and supported by specific IT tools, this part of the process can be the most time consuming.

5.1.3 Early Assessment of Compiled Aggregates (Production)

This is a check of reporter aggregations that is supposed to give an early indication of major quality issues in the final statistics. The assessment can be made on the basis of time series of aggregates, graphs or tables with drill-down features. Notice that this differs from the check on reporter aggregates in that this is not standardized, with defined outlier identification. Even though this is an assessment without any explicit outlier identification, the tools involved may adopt some of the methodology of aggregated checks and subcomponent and their contributions to help pinpoint the responsible reporter/item.

5.1.4 Ad Hoc Checks (Production and Non-Production)

Some types of check are not suitable to be implemented as standardized checks – either because data is not yet available, because the check is of a complex analytical nature, or that their relevance is confined to certain time periods. Examples of this are longer-term developments in item compositions and comparisons with other data sources for the same data (e.g. financial reports). These checks are essential for ensuring levels and composition of the statistics. The workflow surrounding these checks is generally unrelated to the production and can be performed at any time.

5.2 Coordinating Checking Between Different Compilations

The data contained in a report will often be used for multiple purposes. As an example, the report of a bank will both spill into the Loans and Deposits Statistics, but also into the Financial National Accounts and the Balance of Payments Statistics.

The analyst in the compiling institution who receives a specific report may be analytically involved in only parts of the final use of that report and his analytical checking may thus be focused on this part. In this case, other analysts will subsequently have to validate the report seen from their analytical perspective. This can lead to a situation where plausibility checks are done more than once, by different analysts and at different points in time, adding much complexity to the checking process for both compilers and reporters.

If all users of data define their data checking needs for each type of reporting and all checks are implemented by the institutional entity receiving data, such workflow complexities are to a large extent circumvented. This common definition is in turn facilitated by having a universal language for defining checks, as outlined in the Framework section.

5.3 Workflow Implications of Checks with Reporter Aggregation

As outlined in section 1, an important way to improve the efficiency of the statistical production is by using check items with reporter aggregation. This, however, potentially implies a new way of arranging the workflow.

5.3.1 Choosing Level of Checks

In a setup where all checks relate only to a single report, the workflow has the advantage of giving a simple workflow, where an analyst checks the report and gives feedback independently of other checking. This case is shown in panel A of Chart 9.

In section 3, two different ways of performing aggregated checks were described, with outlier identification at either the check item level or the subcomponent level. If we perform aggregated checks with outlier identification at the check item level, then our checking process will not be based on the individual reports, but instead around the check item (see panel B of Chart 9). Since there is a 1-to-many relation between a check and reports the problem of coordinating the feedback to reporters arises, and if the checks are performed by different persons, it creates interdependencies between the workflow of each person. If we perform aggregated checks on subcomponent contributions, the workflow can again be confined to a single report because the outliers identified have a 1-to-1 relation to reports (see panel C of Chart 9).

This illustrates some important workflow implications of doing checks based on reporter aggregation with outlier identification at the check level (Method 1 of Chart 4) vs. using outlier identification at the subcomponent level (Method 3 of Chart 4).

5.3.2 Combining Checks on Reporter Aggregates with Individual Checks

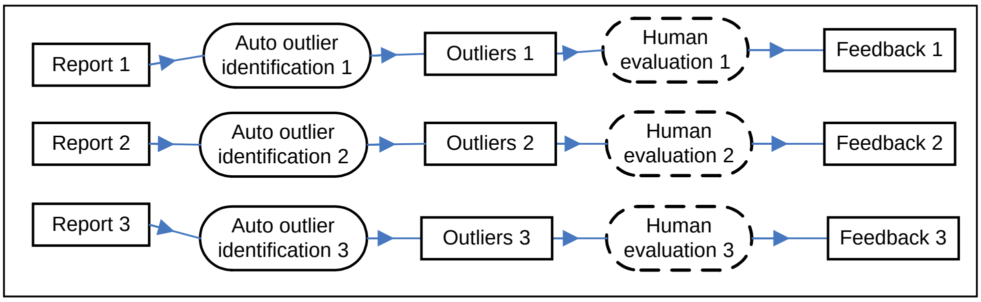
In cases where compilers also use data in disaggregated form, they cannot rely entirely on checks done on reporter aggregates, but also need to evaluate each individual report. For such compilers, it has been common practise to start by performing checks on the individual reports and, once these checks are passed, proceed to the checks on reporter aggregates. According to Drejer (2012), 81% of compilers that use checks on reporter aggregates utilize this workflow. This, however, potentially has two disadvantages.

Firstly, when performing two separate rounds of checking, there is a potential for two rounds of feedback and two different analysts being involved in checking a report which creates a need for coordination. A way of overcoming this problem is to perform checks on reporter aggregates at the subcomponent level (Method 3 of Chart 4). As described in 5.2.1, the checking on reporter aggregates in this case becomes attributable to the individual reports

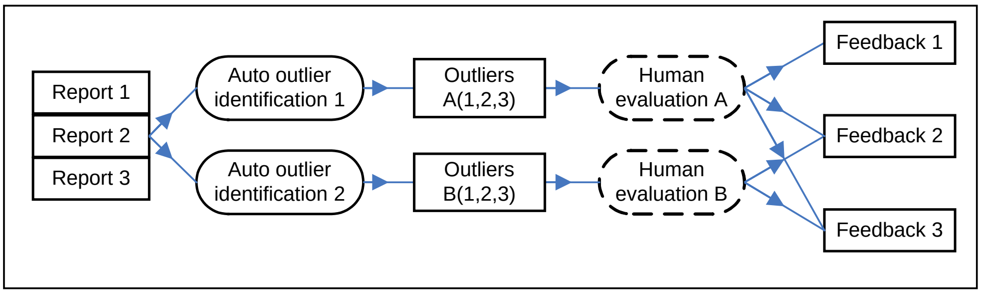
Chart 9

Different ways of organising the workflow

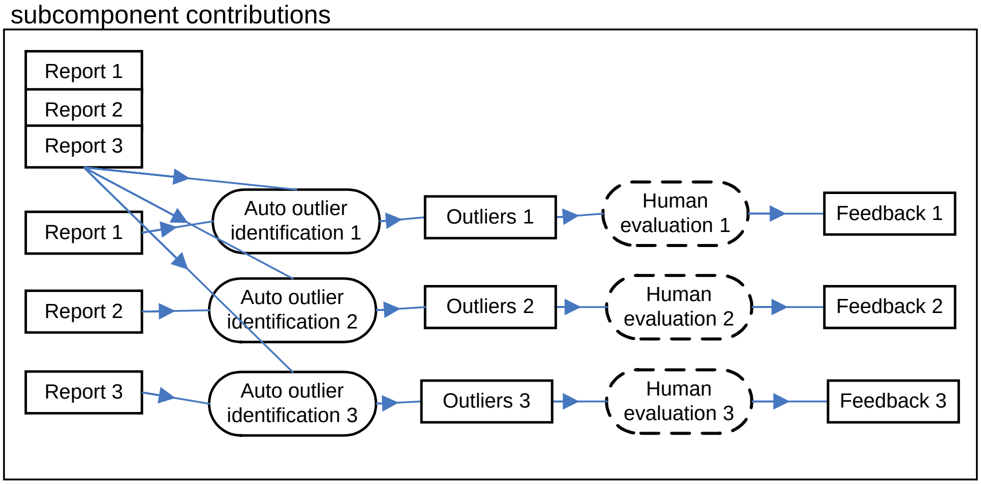
A. Checking without reporter aggregation



B. Checking with reporter aggregation, outlier identification based on check item



C. Checking with reporter aggregation, outlier identification based on subcomponent contributions



Square boxes represent data. Boxes with round edges represent processes and boxes with dotted line and round edges represent processes with human involvement.

and hence the checking can be merged with the individual checks. Thus, there is a workflow with one analyst doing one check and performing one feedback.

Secondly, in cases where checks on reporter aggregates are done focusing on the aggregated check item (Method 1 of Chart 4) one of the main advantages is that it helps to prioritize between which reports to investigate first. By starting with individual checks in a more arbitrary fashion, compilers cannot exploit this advantage.

At Danmarks Nationalbank we have the need to check data at individual reporter level for the interest rate statistics. As we want an integrated approach to checking data, we have chosen to merge individual checking with checking of reporter aggregates focusing on the subcomponent level. However, in order to exploit the prioritizing advantages of checking reporter aggregates at the aggregate level, we have supplemented the checking process with a tool that enables us to check at the aggregated level (Method 3 of Chart 4) and drill down to reporter level. This tool is used in the very beginning of the production process to get a first overview of overall developments and significant contributors to help prioritize the checking and throughout the production to keep track of the current state.

6. The IT Application for Human Evaluation

The human evaluation is aided by IT applications. Since they are tools used for repetitive evaluation outlier after outlier, report after report, production after production, they need to be designed for practical implementation. In general, if assessing an outlier involves many manual procedures, it significantly slows down the process, tires the analyst and increases the likelihood that procedures will be skipped.

An important general aspect in the design is to confine and specialize the tool to the specific task it needs to do. Too much flexibility in the tools will often slow down the process because of additional data processing time and the lack of guidance of what to do.

6.1 Experiences from the Interest Rate Statistics

The current IT application used for checking interest rate statistics at Danmarks Nationalbank has continuously been developed in close collaboration with data analysts.[5] Over the years different kinds of features were added, some of which have later been abandoned as they turned out not to be durable, however, other survived and turned out to add great efficiency gains. In the following, we will try to sum up some of the concepts that were really durable. The description covers two main tools and some remarks on technical aspects.

6.1.1 Tool for Evaluating Automatically Identified Outliers

For the process of checking the (potentially large) amount of automatically identified outliers, we have found a need for tool with the following features:

Providing Supplementary Information

Alongside the outlier the analyst should receive information that can support the decision making. The better information available, the easier and more qualified decisions can be made by the analyst.

Graphical Presentation

Data related to the outlier, should be presented graphically. An inspection of 2-3 graphs can give the same overall information as considering 7 data series, but in a much faster and intuitive way. Also importantly, when looking through numerous outliers, reading/considering

[5] The tool is technically based on a combination of SAS and Excel, which enables a large degree of flexibility for changing the setup and adding new features.

figures is tiresome. In total, a graphic presentation offers a substantial gain to efficiency and should be considered a must in any checking tool.

Integrating Historical Decisions and Explanations

A good amount of outliers will be due to factors that are more or less systematic in nature and will make them look erroneous even though they are not. Both information on previous decisions about the outlier and explanations to previous outliers may greatly help to explain the current outlier. Hence presenting decision history and previous dialogue for outliers along with the current outlier will often greatly help the checking process.

Integrating Communication with the Reporter

Administrative tasks such as generating a joint error report or handling contact information are unrelated to the content of the statistical reports and thus distract the analyst from his core work. Therefore administrative tasks relating to the communication with reporters should be minimized. Ideally the feedback system is integrated in the checking system so that a report will be automatically generated.

6.1.2 Tool for Assessing Revisions

When reporters send revised reports, it is important to be able to directly compare the failing checks of the current version with those for earlier versions for the same period. If the tool does not incorporate such comparisons, the analyst will have to do this manually which disturbs the workflow.

6.1.3 Technical Aspects of the IT Applications

Some general features can make IT applications better tools:

No Waiting Time

Apart from slowing down the checking process by its duration, waiting time related to IT tools can also seriously harm concentration and productivity for the analyst. Therefore, the aim should be no waiting time, which is generally best achieved by having generated all data used in the checking application before the checking starts.

Ergonomics

Looking through a list of outliers is a standardized task that requires submitting the same commands repetitively to the computer. It is our experience that this is done faster and more ergonomically friendly by the use of keyboard rather than mouse. While it might seem like a minor point, it can have major impact on the speed and comfort of the human evaluation process.

One Tool

Preferably, all steps of the human evaluation process in a production setting should be consolidated into one application, so that the analyst will not have to switch between applications. While such a goal is most easily achieved when the application is developed in-house, there is also scope for approaching this goal when purchasing software. Keeping the goal in mind throughout the design phase and when purchasing software will help achieve it.

7. Encouraging Checks by the Reporting Entity

In the ideal situation, all reports are submitted without errors. While this may seem like utopia, steps can be taken in order to promote a development towards reports being checked more carefully by the reporter before being submitted. One example of this is where compilers enable the reporter to do self-checking. Another example is where compilers try to increase reporters' interest in their reporting by providing a useful individualized output.

7.1 Sharing Check Definitions and Allowing for Pre-emptive Comments

Generally reporters would prefer to avoid a process of follow-up queries upon their submission of the report, and would rather be able to deal with potential problems at the time when they first create their report. By sharing checking definitions, reporters can proactively run plausibility checks on their reports and correct potential errors before submitting data.

In combination with the above, a feature that enables reporters to attach pre-emptive comments to their reports could avert an enquiry about an outlier if a good explanation already exists. If the reporter sees that his report will result in an error he would be able to attach a comment explaining the development.

Another way of achieving the above two objectives is by allowing reporters online access to (parts of) the same checking systems as compilers use. In this way, reporters could submit reports and see failing checks and directly comment on these.

7.2 Providing an Individualized Report to Reporters

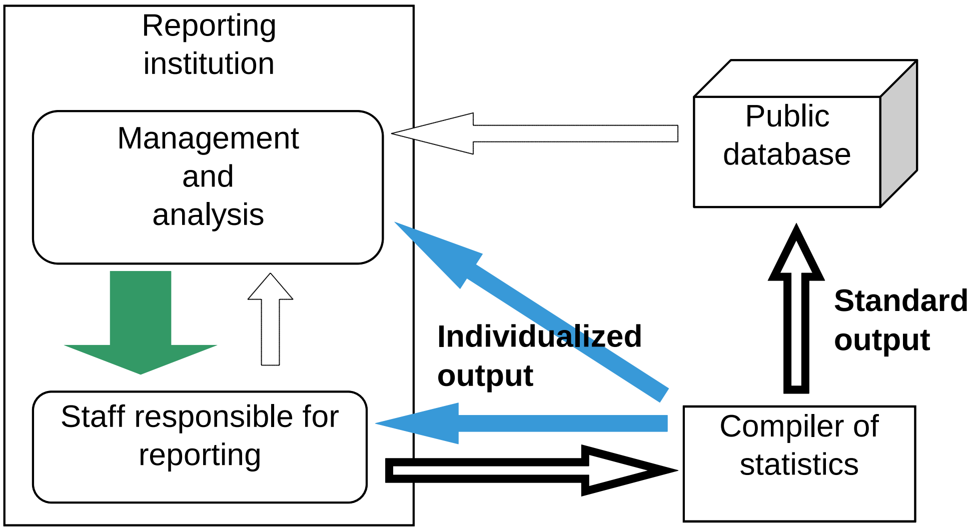
One fundamental problem in motivating reporters to provide a report of high quality is that they do not get any payoff from doing so. On the contrary, many reporters see the statistical reporting only as a burden. For this reason, reporters are generally sceptical about new statistical requirements.

The punitive measures that compilers of official statistics often can use will generally only apply to reporters with very low quality reports and not spur reporters to change from mediocre to high quality. However, if reporters could get a useful output from their reports, their motivation may increase.

One way of giving the reporter a useful output from their reporting is by providing a follow-up output in which the reporter's own figures are related to sector statistics. In its most basic form, this type of output is just facilitating the task of merging published statistics with the reporter's own reported figures – a task that could be done by the reporter but would require substantial work. This output promotes the organizational awareness of the report and legitimizes spending more resources on the reporting (see Chart 9).

Chart 9

Providing an output based on reporters' own reports



Arrows with black edges denote the standard data flow. Arrows with grey edges denote data flows that could, but likely do not, occur. Blue arrows denote the delivered individualized output. Green arrow symbolize organizational awareness.

7.2.1 Example of an Implementation

At Danmarks Nationalbank, we have implemented a solution, where we provide monetary and financial institutions with an individualized output to their reporting. The reporters were involved when we developed the output in order to maximize its usefulness. In addition to providing useful information for the banks, the output also offers an opportunity of representing the reported data in a way that shows how a reporter's reporting affects key statistics.

A survey carried out among our reporters showed that 75% of larger reporters use the data while only 29% of smaller reporters use the output. While it is difficult to measure the direct impact on reporting quality, it is our experience that this has significantly heightened the awareness of the statistics among reporters and that this in turn has led to reports of higher quality and a more positive attitude towards general collaboration and new data requirements.

References

Bartmann, M. and Drejer, P. A. (2011): Finding an optimal benchmark for checking interest rate statistics. *Work Note - available upon request.*

Caporello, G and Maravall, A. (2003): A tool for quality control of time series data Program "TERROR". *Documento Ocasional 0301, Servicio de Estudios, Banco de España.*

van den Dool, G., de Vor, M. and van der Wal, D. (2008): A macro-micro approach to compiling statistics. *BIS Irving Fisher Committee on Central Bank Statistics Working Papers*, No 2*.*

Drejer, P.A. (2012): Survey on Practices in Optimizing Data Checking. *Work Note - available upon request.*

Huerga, J. and Steklacova, L. (2008): An application of index numbers theory to interest rates. *ECB Working Paper Series*, No. 939*.*

IMF (2008): *Monetary and Financial Statistics - Compilation Guide*.

A. Kotsiantis, S.B., A Zaharakis, I.D., A Pintelas, P.E. (2006): Machine learning: a review of classification and combining techniques *Artificial Intelligence Review*, Volume 26, Issue 3, November, pp. 159-190.