UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

Work Session on Statistical Data Editing
(The Hague, Netherlands, 24-26 April 2017)

Methods Library as part of the modernization of the statistical production in Norway

Prepared by Aslaug Hurlen Foss, Øyvind Langsrud and Ane Seierstad, Statistics Norway

I. Introduction

1. Statistics Norway has started a process to modernize the statistical production. The first step is to build a common metadata and data storage on a modern platform. VTL ("Valdiation and Transformation Language") will be one of the main tools for working with the data. A Methods Library will be built to offer common methods throughout the production process. The methods in the library will be made CSPA-compatible (Common Statistical Production Architechure) and are so far written in the R language and gathered in R packages, which make the exchange with others easy. An administrative module to handle the administration of the methods and their metadata is also a part of this project. This is to ensure that the information needed by the users and the IT system is in place and taken care of. A Wiki page for user documentation is written, to inform the users about how and when to use the different methods offered by the Methods Library. A separate graphics module will handle all the graphics needed; hence all the variables used for graphics will have to be an output from the methods. The first methods included in the Methods Library are methods for macro-editing, imputation and confidentiality.

2. Modernisation of the statistical production is starting with a modernisation of KOSTRA – municipally state reporting. This is a collection of 20 statistical areas about services provided by the municipalities and the accounting for these services. These statistics are based on several sources: municipality survey, files from local administrative system and registers. One of the main goals for these statistics is the productions of indicators, which can be used for analysis and comparing the municipalities. The scope of the modernization of KOSTRA is from after micro editing and includes upgrade of Statbank and publishing solutions.

II. Modernisation of statistical production

A. Modernisation Standards in Statistics Norway

3. We are currently using four UNECE HLG-MOS (High Level Group for the Modernisation of Official Statistics) Modernisation Standards i.e. Generic Activity Model for Statistical Organisations (GAMSO), the Generic Business Process Model (GSBPM), Generic Statistical Information Model (GSIM) and the CSPA. See the Virtual Help Desk (http://www1.unece.org/stat/platform/display/VSH/Virtual+Standards+Helpdesk) for more information on these four standards.

4. We have assessed our current maturity levels for all four Modernisation Standards, using the Modernisation Maturity Model, and will be trying out the recently developed Modernisation Roadmap to reach our target maturity levels in our modernisation program. Our experiences will be fed back to the international statistical community.

5. We use the GSBPM v5 as a reference model and have actively participated in review rounds for all GSBPM versions. Our own internal business process model currently has two more levels of detail than the GSBPM.

6. GSBPM in combination with GSIM will be used to create an environment prepared for reuse and sharing of methods, components and processes in our current modernisation program. We are looking at the flow of information objects through our business process model. We have begun with the 114 GSIM information objects and will expand these as needed at the conceptual level. We will also be making our own Logical Information Model (LIM), based on the CSPA-LIM, to provide the level of detail needed for business services.

7. In moving away from the development and maintenance of silo applications and towards capability development, as in the GAMSO, we expect to increase our efficiency and enable the more rapid uptake of new data sources and production of new statistical products.

B. Methodology architecture

8. Throughout the production system, several statistical methods are needed. The previous production system embedded the methods inside the application, so that only statistics within the system could use the methods. The methods were tailor-made to suit the system and the statistics. Many methods have been duplicated in several production systems. When building a new system, a decision was made to build generic methods and put them in a library. Each method is independent and can be put together as building blocks. This is according to the recommendations from CSPA. The intention is that the methods are collected and maintained in one place. The Methods Library is governed and maintained by the Methods division at Statistics Norway. In building the Methods Library the recommendations from the paper "Methodology architecture" (Claude Poirier, 2015) are considered. In the web-based documentation, the methods will be mapped according to the GSBPM. The input and output of the methods will be documented and will follow the standard by GSIM. The decision on which methods are included in the library in the next years will depend on the program for modernisation of statistical production at Statistics Norway.

C. The use of Validation and Transformation Language (VTL)

9. VTL is implemented as the standard tool in the production system for working with data. Important functions are wrapped in user friendly interfaces, for example: importing data, merging two datasets and validation of lists of codes or classification. Some functions will probably be without user friendly wrapping, for example deriving new variables based on other variables. A VTL-editor will also be built, including a corresponding preliminary display of data. Then the users will be able to write their own code in VTL. For more information on what is implemented in VTL at Statistics Norway see: https://github.com/statisticsnorway/java-vtl

D. User interface and integration of methods in the production system

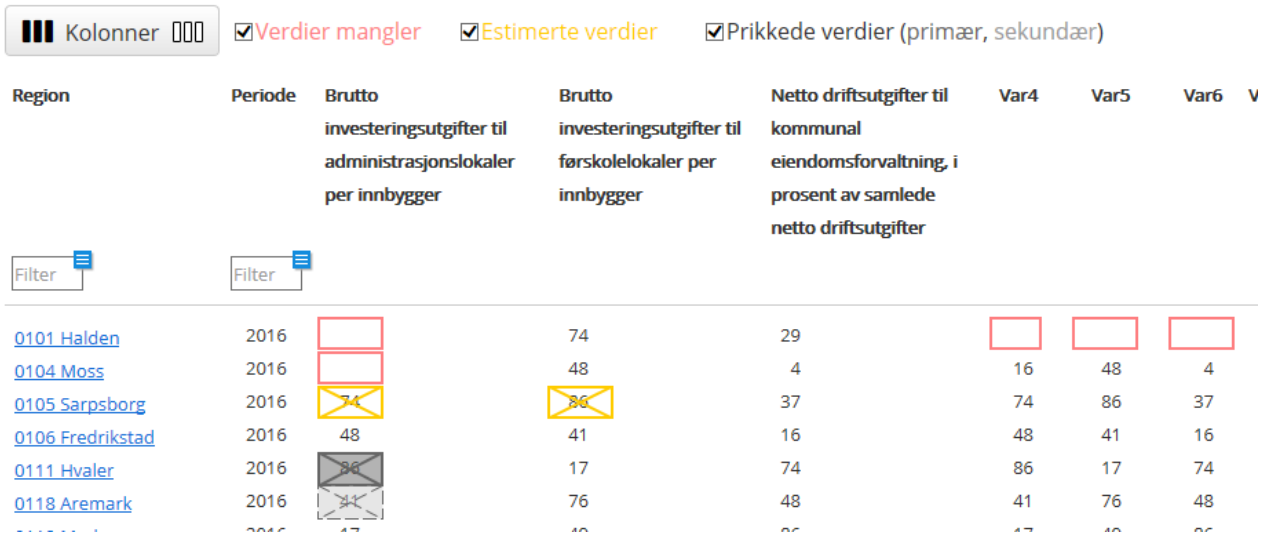
10. A web user interface is built and includes separate tabs for metadata, VTL, data editing (macro), imputation, confidentiality, preview of data and productions. Productions are a setup of different tables/dataset in a productions system until it is ready for publishing. Hopefully, there will also be a separate tab for "graphical - final inspection" and evaluation. The interface is built in Java and the framework of Vaadin.

Figure 1. Preliminary view of the user interface

11. Methods are set up so it is possible to run them both automatically and manually. In the modernisation of statistical production, it is important that as much as possible is run automatically and that process quality indicators are built-in. However, it is also important to have the possibility of a manual setup to enable to the user to select the best methods and based on investigative reruns looking at graphics and quality indicators. Different groups of methods have different challenges with the user interface. For imputation, the challenge is quality indicators at an aggregate level in addition to the viewing of micro data. For macro editing, the challenge is how to put different methods together. It is often convenient to have methods which provide an overview, control against the previous year and control against another variable, at the same time. For confidentiality the challenge in building the user interface is the number of variables and the connection between them.

12. For the final view of the data, it will be possible to see which value is missing, imputed or supressed by confidentiality, it will be shown in different colours, see figure 2. It will also be possible to turn on and off the cursor for this viewing. The same marking of outliers for macro editing methods could also be done.

Figure 2. Preliminary view of data with selection of values based on missing, imputed or supressed



E. Graphical display of methods and data

13. A module for using interactive figures in Highcharts is built as a part of the project. Then interactive figures can be used in the production process as well as in publishing article on the web. Interactive figures are figures with roll-over and selection features. The user can read values, mark parts of the figure and change which part of the figure will be shown. It will be possible to zoom in on parts of the figure. For an overview of the data, bar charts, line charts and tree maps will be available. For tree maps, it will be possible to show a second dimension, for example the change from last year's numbers in per cent. (M. Tennekes, E. De Jonge and P. Daas. 2012). For closer inspection of data, scatterplots, bubble and distribution graphs will be possible. In these plots it will be possible to show regression lines, limits of methods and outliers marked in different colours.

III. Building of the Methods Library

A. R programming and execution of R code

14. Execution of R code

Within the current IT system, Java interacts with R via RServe installed on a Linux platform. The architects are also looking at OpenCPU as an interface to R. Another possibility is Renjin which is a Java implementation of R. However, at present, many common R-packages for official statistics are not compatible with Renjin. Note also that the packages, data.table and dplyr, meant to save time and memory cannot be used within Renjin.

15. Standard for R programming

We have developed an internal R programming guide. The starting point was Google's R style guide. We collect the R code in packages and the functions are documented using roxygen2 which means that documentation and code are side-by-side on the source files.

We recommend writing the code as portable as possible. This is in accordance with the CRAN repository policy which says: *"Package authors should make all reasonable efforts to provide cross-platform portable code"*. In practice this means that we will, when possible, make the code compatible with Renjin. Especially, this means that we will only use packages such as data.table and dplyr if this in practise makes very important efficiency improvements.

15. **Functions for the library**

The R functions explicitly included in the library are the links between R and the other system. It is therefore important that such functions follow a standard which is currently:

- The first input parameter is a data set of type data.frame.
- The other parameter must be vectors (normally of length one) of type character, numeric, integer or logical.
- A special variant of the character type is *variable name*, referring to variables in the input data set.
- One such *variable name* parameter represents a unique identifier of observations.
- Another special variant of the character type is list which means that a list of allowed input elements is pre-specified.
- When input is numeric or integer, the minimum and/or maximum can be specified.
- Output is a single data set of type data.frame.

The last point seems too restrictive and will probably be changed. The specification of input variables works well for ordinary cases. But in general, we may want to specify more than a single variable name, for example whether the actual variable should be taken from the current year or the previous year. We have also made an effort to handle this issue.

The R code consists of several functions that are not exposed in the library and for such functions these rules do not apply. From the library point of view these functions are only internal functions. However, such functions can be visible in the R packages (exported and documented as usual). This means that R users have access to extra functionality beyond the methods in the library. An example is the general function for imputation described below.

16. **Writing and storing code**

- The code is stored in an internal version control repository using Git using the web-based hosting service Bitbucket.
- R programmers can set up projects in RStudio to synchronise with this system. This means that it is possible for several programmers to work on the same package concurrently.
- The process of building the packages for RServe is handled by the IT team based on the code stored in the repository.
- The packages may in addition be distributed to R users internally and externally. The package files to be distributed ("tar.gz", "zip" and "pdf") are created by the R programmers.
- So far two packages are published on CRAN, "SSBtools" and "easySdcTable". Another external possibility is GitHub.
- An overview of our R packages is given on our internal wiki.

B. Administrative module

17. An administrative module is developed as a part of the project. This is to secure a good connection between the R code, IT system and information needed by the users. A Wiki page for user documentation is written, to inform the users about how and when to use the different methods offered by the Methods library.

18. Every time a new method is included in the Methods library, information about this method has to be entered into the administrative module. Also, if an existing method is changed, the information about the method has to be updated in the administrative module. The administrative module contains a list of all the methods that are included in the Methods Library. Every method is given a name which should be easily understood by the user of the Methods Library. The relevant R function is also included, along with a link to the place in GIT where the function code is stored and the different versions are

handled. A short explanation of the method has to be included, as well as a link to the Wiki page with more exhaustive information about the method.

19. For every method, names of all the parameters have to be included; one technical name pointing to the corresponding parameter in the R code and a name meant for the users to understand. Every parameter and its purpose must be explained. This is again a short explanation; a deeper explanation can be found on the Wiki page about the method. Every parameter is then described further, by type (variable, integer, floating-pint number, text, list and variable), parameter limits, default values and whether or not the users can choose multiple values for the different parameters.

20. The output data from the different methods is also defined here, so that the system can handle the output variables correctly throughout the rest of the process. Here, the output variables are given a user-friendly name, along with the corresponding name in the R output dataset. Relevant graphs for each method are also set up here, to offer the users additional information about the output from the methods. The explanation of these variables and graphs can be found on the Wiki user documentation page.

C. Statistical methods in the library

21. **Macro editing methods**

The users are requesting quick, simple methods for macro editing. More advanced methods are often regarded as a "black box" and are not used. We have therefore chosen to start with simple methods in the library and later advance to more complicated functions. We have established two methods which provide an overview, two methods for comparing with previous years and two methods for comparing with correlated variables.

22. *Analysis at an aggregate level* is a simple comparison of two numerical variables at an aggregated level. When editing data, we often wish to compare the present period with an earlier period or with a similar variable from another source. Doing this at an aggregated level might reduce the amount of editing required at the micro level. This function can also be used to analyse the editing process; looking at the aggregate values of the original and edited data will give the statistician a quick impression of the effect of the changes. We can choose to compare only identical units or all units.

23. *Listing greatest values* is a function that compares two numerical variables on an individual level, by ranking them by size. We can choose to list out all the units with one or both variables greater than a threshold value, or to list out for example the 10 greatest values for both variables. This ranking can also be done within different strata, if this is asked for when calling the function.

24. *Listing greatest differences* is also a simple comparison of two numerical variables on an individual level. This time the values of the variables are ranked by the size of the difference between them. The influence on the total and strata is calculated. By listing out the units with the largest differences, the most influential errors on the aggregated level are picked-up. We can choose to list out a given number of units or all units with a difference from previous period greater than a chosen threshold.

25. *The Hidiriglou-Berthelot method* is a method used to control the value of a variable against the value of the same variable in a previous period. The method is based on the characteristics of the data that is being checked, which means that it takes into account both the changing level between two periods and the fact that small values tend to have greater variability. There are three parameters in the Hidiroglou-Berthelot method that can be adjusted, so that the method will pick out more or fewer points for further checking. The default values of the three parameters have to be set based on analyses and experience.

26. *The quartile method* is a way to control two numerical variables that are related. The method makes use of the robust parameters, median and quartile of the ratio between the two variables, to check for extreme values. Using the first and third quartiles, an interval of acceptable values is calculated. Units are picked out for further checking if the value of their ratio is outside the calculated interval.
The formula of the interval of acceptable values is given by:

($q_{0.25}$-k1($q_{0.5}$-q0$_{.25}$), $q_{0.75}$+k2($q_{0.75}$-$q_{0.5}$)), where $q_{0.5}$, $q_{0.25}$ and $q_{0.75}$ is the median, 1st and 3rd quartiles of the ratio. The parameters k1 and k2 control the width of the interval and should be set based on analyses and experience.

27. **Imputation based on linear models**

The starting point for imputation in the Methods Library was the established tools used today. There are two important methods currently in practise: ordinary linear regression and the ratio model (weighted regression without a constant term). Furthermore, observations are categorised based on externally studentized (out-of-sample normalised) residuals.

28. To implement all the requested methods, a general function, named LmImpute, was made that could take possible future extensions into account This function is the working horse within several library functions. In addition to the data set, important input parameters to the function are:

(a) *model:* A general formula. The formula can involve several dependent variables (y's) and several independent variables (x's). Transformations of y (e.g. log) are also possible.
(b) *weights:* A weight formula (weighted regression).
(c) *limitModel:* The final model used to impute extreme and missing values will be based on observations with studentized residuals below this limit (representative observations). The studentized residuals used are from the last iteration.
(d) *limitIterate*: In an iteration process, observations with studentized residuals above this limit are successively thrown out of the model.
(e) *limitImpute:* Observations with studentized residuals above this limit are considered as extreme/wrong and are therefore imputed. The studentized residuals used are from the last iteration.
(f) *estimationGroup*: It is possible to specify a grouping of observations so that total estimates are computed within each group.

Other input parameters are used to specify what kind of output will be produced. Among the output possibilities we have the imputed values, total estimate with standard errors, studentized residuals and outside-model residuals.

In the Methods Library we have the following functions that make use of LmImpute:

29. *ImputeRegression:* Imputations are performed within several strata and the possible models include ordinary linear regression, a ratio model, regression without a constant term, a simple mean model (only constant term) and a ratio model extended with a constant term.

30. *ImputeRegression2:* This function is similar to ImputeRegression except that imputation (of y) is performed in two rounds, first using a primary variable (typically y from last year) and thereafter a secondary variable is used for cases where the primary is missing. The final standard errors take into account variability from both imputation models.

31. *ImputeRegressionMulti:* This is simultaneous imputation of several dependent variables (y's) using the same dependent variable (x). The categorization into representative observations (for the model) and extreme observations (to be imputed) is the same for all dependent variables.

32. ImputeHistory: This is direct imputation of the most resent non-missing historical value. Standard error estimates are based on the naive model where the difference between the current and the historical value is assumed to be pure error. A trick is used to utilize the function LmImpute also in this case.

33. *OutlierRegression:* This function is similar to ImputeRegression and is used to find outliers using a limit for studentized residuals. Imputation is not performed.

34. **Statistical Disclosure Control**

Based on the statistical disclosure control package “sdcTable” a function for table suppression according to a frequency rule was made. Making the function general and at the same time satisfying the standard for the library was especially challenging. New methodology was developed and among other things automatic detection of hierarchical relationships between variables. The result is made available as a public package: https://CRAN.R-project.org/package=easySdcTable

IIII. Experience from statisticians in taking part in a modernisation program

34. The role of the statistical methodologists was not defined within either the IT-project or modernisation program. We have tried to define our role, but it has been a gradual process to become incorporated into the team. We are between the user and the IT-staff. Reaching a common understanding of a user story or problem has often taken time. The amount of administration of the project is huge, also for the statisticians taking part.

References:

Claude Poirier, Statistics Canada. Documented under the auspices of the HLG Modernisation Committee on Production and Methods. METHODOLOGY ARCHITECTURE V0.04. Released date: 15 July 2016 Version 0.04

VTL – version 1.1 8 (Validation & Transformation Language) Part 2 – Reference Manual https://sdmx.org/wp-content/uploads/VTL-1-1-review-Reference-Manual-20161017-final.pdf

R-server: https://www.rforge.net/Rserve/

M. Tennekes, E. De Jonge and P. Daas. 2012. UNECE. “Innovative visual tools for data editing” http://www.unece.org/fileadmin/DAM/stats/documents/ece/ces/ge.44/2012/30_Netherlands.pdf