Working Paper

UNITED NATIONS
ECONOMIC COMMISSION FOR EUROPE

CONFERENCE OF EUROPEAN STATISTICIANS

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

Software implementation of optimization-based selective editing techniques at Statistics Spain (INE)

Prepared by E. Esteban, S. Saldaña, and D. Salgado, Statistics Spain (INE), Spain

I. INTRODUCTION

1. In last years [1, 2, 3, 4, 5, 7] Statistics Spain (INE) has been proposing and using an optimization-based approach to selective editing to streamline the statistical data editing phase of the production process. The application of this approach to selectively edit quantitative variables has been undertaken in several short-term business statistics already in production (Retail Trade Index, Industrial Turnover Index, Industrial New Orders Received Index, Services Sector Activity Indicators, Short-Term Stock and Inventory) and some others already analysed with actual survey data and waiting for their new editing strategy to be incorporated in the data collection phase (Industrial Price Index and Export and Import Price Indices for Industrial Products).

2. The computer tools for the analysis during the development and testing phase of these ideas were constructed in R to be later developed under the programming standard for production applications and services at Statistics Spain. However, due to severe resource limitations, the original R tools were actually evolved and adapted for their use in a production environment. This is still an on-going process.

3. This document describes the main lines of the classes, methods, functions, and packages programmed in R to implement the optimization-based approach to selective editing. Some words of caution are in due order. This implementation is focused upon an attempt to build a standard architecture of R packages and functions valid for any statistics with a strong effort towards modularity and extensibility with current and future methodological refinements. The stress has not been placed over programming details (yet) but over producing a skeleton prototype architecture robust enough to be used in production conditions and still flexible enough to adapt to new situations.

4. As a highly important ingredient in this architecture we must mention that these tools are heavily based upon a standardised data model put in place internally at Statistics Spain as part of the modernisation proposals of the production process [8]. The bottom line of this model is the key-value pair structure: each single datum in the production process will be stored as a single value univocally identified by a key.

5. A first version of this data model was developed and used in production for all the statistical operations cited above. The corresponding version of the editing computer tools were also developed upon this original version. Over one thousand of files were generated feeding a centralised microdata repository for these surveys. The computer tools were used to run their editing strategies in production.

6. Recently we have substantially improved the model by using the internal system of metadata to compose the key of each variable. In the following, we propose to give a short summarised overview of the optimization approach thus drawing the methodological framework under which the computer editing tools have been designed and developed upon this second improved standardised data model.

II. OVERVIEW OF THE OPTIMIZATION APPROACH TO SELECTIVE EDITING

7. We give a short overview of this approach underlining those key elements implemented in the computer tools. The starting point is the formulation of the principles under which to build an editing strategy (at least the selection of units for interactive editing):

(i) Editing must **minimise** the amount of resources deployed to **interactive tasks**.
(ii) Data **quality** must be **ensured**.

8. Identifying the amount of resources with the amount of questionnaires to edit interactively and restricting ourselves to the accuracy dimension of data quality, these two principles can be translated into the following generic optimization problem [4]:

$$
\begin{aligned}
[P_0] \qquad & \max \ \mathbb{E}_m \left[\mathbf{1}^T \mathbf{R} | \mathbf{Z} \right] \\
\text{s.t.} \quad & \mathbb{E}_m \left[L \left(\hat{Y}^{(*,q)}(\mathbf{R}), Y^{(0,q)} \right) | \mathbf{Z} \right] \leq \eta_q, \quad q = 1, 2, ..., Q, \\
& \mathbf{R} \in \Omega_0,
\end{aligned}
$$

where

- m denotes the measurement error model linking the observed raw values $y_k^{(q)}$, true values $y_k^{(0,q)}$ and finally edited values $y_k^{(*,q)}$ for variable $y^{(q)}$;
- $\mathbf{R}$ is a binary random vector $\mathbf{R} \in \{0,1\}^{\times n}$ indicating whether each sample unit k must be edited interactively or not;
- $\mathbf{Z}$ is an arbitrary random variable containing all longitudinal, cross-sectional or multidimensional information available to undertake the selection of units;
- L denotes a loss function, basically either the absolute difference $L(a,b) = |a-b|$ or squared difference $L(a,b) = (a-b)^2$;
- $\hat{Y}^{(*,q)}$ and $Y^{(0,q)}$ denotes the (linear) estimator using edited values $y_k^{(*,q)}$ and the population total of variable $y^{(q)}$, respectively;
- η_q is an upper bound for expected loss of accuracy due to the presence of measurement errors;
- Ω_0 is a subset of $\{0,1\}^{\times n}$ thus allowing for some units to be selected of discarded beforehand.

9. The generic optimization problem P_0 reduces to a stochastic optimization problem or a combinatorial optimization problem depending on having only longitudinal information about each sample unit or having also cross-sectional information across the current sample (for example after data collection has been finalised).

A. The stochastic version

10. The stochastic version was originally proposed in [1]. Recently we have proposed a substantial change in the resolution of this problem trying to improve the selection of units [7]. The statistical operations cited in the introduction do not incorporate this version of the optimization problem and

further work needs to be undertaken to analyse this recent proposal. No further reference to this version will be made in the subsequent.

11. In substitution, the longitudinal phase of the editing strategy (when no current cross-sectional information in the sample is available to select the units), we have developed a computationally demanding proposal exploiting the historical data of each unit to build a validation interval *for each variable and each unit* using time series techniques.

12. The basic idea is to make a point prediction $\hat{y}_{kT}^{(q)}$ for time period T for each variable y_k and each unit k using a time series model upon the series $\{y_{kt}^{(*,q)}\}_{t=1,2,...}$ of edited values. This also includes the standard deviation of the prediction $\hat{\sigma}_{kT}^{(q)}$.

13. With these two values, we build a validation interval of the form $I_{kT}^{(q)} = [\hat{y}_{kT}^{(q)} - \xi_{kT}^{(q)} \cdot \hat{\sigma}_{kT}^{(q)}, \hat{y}_{kT}^{(q)} + \xi_{kT}^{(q)} \cdot \hat{\sigma}_{kT}^{(q)}]$, where $\xi_{kT}^{(q)}$ is an adjustment factor depending on the performance quality of the selection in preceding time periods (e.g. the hit rate). If the performance is good, we can increase the length of the intervals and thus gain in efficiency by relaxing the conditions under which the value is suspect of an influential error and vice versa.

14. The selection of units under the construction of these intervals is further controlled by defining a distance function $d(I_{kT}^{(q)}, y_k^{(q)})$ between the validation interval $I_{kT}^{(q)}$ and the observed raw value $y_k^{(q)}$. If this distance is above a given threshold $t_k^{(q)}$, then the variable vale $y_k^{(q)}$ is concluded to be suspect of an influential error.

15. The computation of thresholds is also undertaken by using times series tecniques. In particular, a historical set of intervals is maintained so that, together with the historical set of edited values, we can have the corresponding time series of distance $\{d_{kt}^{(q)}\}_{t=1,2,...}$ for each variable $y^{(q)}$ and each unit k. A synthetic threshold value is computed using the distance point prediction $\hat{d}_{kT}^{(q)}$ and an appropriate quantile $q_i(\hat{d}_{kT}^{(q)})$ of these values over each cell i of units (also appropriately chosen; e.g. if indices are computer by NACE class or group, these would be a natural choice):

$$t_k^{(q)} = (1 - \lambda) \cdot \hat{d}_k^{(q)} + \lambda \cdot q_i \text{ for each } k \in s_i.$$

B. **The combinatorial version**

16. When cross-sectional information of the sample units is available, the generic problem P_0 reduces to a combinatorial optimization problem:

$$\begin{aligned} [P_{co}(\mathbf{M}, \boldsymbol{\eta}, \Omega_0^*)] \qquad & \max \ \mathbf{1}^T \mathbf{r} \\ & \text{s.t.} \quad \mathbf{r}^T M^{(q)} \mathbf{r} \leq \eta_q, \quad q = 1, 2, ..., Q, \\ & \qquad \mathbf{r} \in \Omega_0^* \subset \{0, 1\}^{\times n}, \end{aligned}$$

where $M^{(q)}$ denote the so-called conditional measurement error moments for each variable $y^{(q)}$.

17. The conditional moments, being conditional expectation values as they are [4], can be analytically computed once a measurement error model m has been chosen for $y^{(q)}$ and $y^{(0,q)}$. This so-called observation-prediction model is determined by $y_k = y_k^0 + \epsilon_k^{obs}$ and $y_k^0 = \hat{y}_k + \epsilon_k^{pred}$, where $\hat{y}_k$ denotes the predicted value according to an auxiliary model m^* (e.g. a time series model). The model is fully specified by (for each q):

(1) $\epsilon_k^{obs} = \delta_k^{obs} e_k$.
(2) $e_k \simeq Be(p_k)$, where $p_k \in (0,1)$.
(3) $\left(\epsilon_k^{pred}, \delta_k^{obs}\right) \simeq N\left(\mathbf{0}, \begin{pmatrix} \nu_k^2 & 0 \\ 0 & \sigma_k^2 \end{pmatrix}\right)$.
(4) ϵ_k^{pred}, δ_k^{obs} and e_k are jointly independent of $\mathbf{Z}_k^{cross}$.
(5) e_k is independent of ϵ_k^{pred} and δ_k^{obs}.

18. Using the absolute difference loss function, it can be shown [4] that the conditional error moment matrix is diagonal with entries given analytically by:

$$M_{kk} = \sqrt{\frac{2}{\pi}} \cdot \omega_k \cdot \nu_k \cdot {}_1F_1\left(-\frac{1}{2}; \frac{1}{2}; -\frac{(y_k - \hat{y}_k)^2}{2\nu_k^2}\right) \cdot \zeta_k\left(\frac{y_k - \hat{y}_k}{\nu_k}\right),$$

where $\zeta_k(x) = \frac{1}{1+\frac{1-p_k}{p_k}\left(\frac{\nu_k^2}{\sigma_k^2+\nu_k^2}\right)^{-1/2} \exp\left(-\frac{1}{2}\frac{\sigma_k^2}{\sigma_k^2+\nu_k^2}x^2\right)}$ and ω_k denote the sampling (design) weight of unit k.

19. Notice that this model is only valid for continuous variables. Indeed, as $|\frac{y_k-\hat{y}_k}{\nu_k}| \to \infty$, $M_{kk} \to \omega_k|y_k - \hat{y}_k|$, as in the traditional heuristic approach to selective editing [9]. As a matter of fact, conditional error moments can be interpreted as item (local) score functions computed according to an underlying measurement error model.

20. The model parameters $(p_k, \sigma_k, \nu_k, \hat{y}_k)$ will be estimated upon the historical data sets of the survey.

21. Once the moments are computed, the combinatorial optimization problem can be solved for each set of bounds η_q. However, as discussed in [4], a prioritization instead of a selection of units is more appropriate for production. This prioritization can be obtain by solving the problem with decreasing bounds η_q and fixing each new selected unit in each iteration [4]. This can be shown to be equivalent to sorting the units by their value S_k upon applying a function $S_k = S(M_{kk}^{(1)}, \ldots, M_{kk}^{(Q)})$ on their corresponding error moments [7].

22. This function S can be understood as a unit (global) score function and can be interpreted as the choice of progressive reduction of the (absolute pseudo-)bias due to the presence of measurement errors as we edit more and more questionnaires.

23. Once units are prioritized, the production conditions shall fix the total number N_{max} of questionnaires possibly edited interactively given restrictions such as time, human resources, ...

III. THE STANDARDISED DATA MODEL AND ITS TOOLS

A. The model

24. The data model consists of a compound key and a value per each single datum in the production process. The elements composing the key are:

- The identifier of the statistical variable (we call IDDD).
- A set of qualifiers, making use of internal metadata as much as possible.

25. As an illustrative example let us consider the final validated value of the turnover of a given statistical unit (business unit with internal ID 293493034MM) in a particular economic sector (retail sale in non-specialised stores with food, beverages or tobacco predominating – NACE Rev.2 code 47.11) and in a particular geographical region in Spain (NUTS2 – internal code 01) in the monthly Retail Trade Survey (internal code E30103) for a given reference time period (October, 2016) giving rise to the first dissemination release of this statistical output (not later revised figures). The semantic content of the qualifiers are contained in a data dictionary (DD) per statistical operation. Basically the qualifiers correspond to headings in standard internal or international classifications (as of this writing more than 45 classifications are maintained by the statistical metadata unit). The key will be composed by the following elements:

- IDDD: TURNOVER
- Qualifiers:
  - in file name: E30103, FF_V1 (V1 indicating the version of the data dictionary), MM102016, D_1;
  - inside file: 293493034MM47.1101.

Notice how qualifiers inside each file containing the data, being a string compound of substrings of fixed length, cannot be interpreted without the data dictionary. This entry will be stored in a file named E30103.FF_V1.MM102016.D_1 and the corresponding row in the file will be TURNOVER@@293493034MM47.1101@@32456, being the last number the value.

B. The packages

26. To implement this data model and to manage the repository of files we have developed four R packages with specific functionalities:

- Package RepoTime [10], to define classes and methods to manage the notation of the reference time periods (see example above).
- Package StQ [11], to define classes and methods to implement the key-value pair data model.
- Package RepoReadWrite [12], to read and write files from and to the physical repository.
- Package RepoUtils [13], to securely connect to the repository and some other related functions (this is highly specific of the internal architecture at Statistics Spain).

27. The central object in this implementation is that given by the class StQ (from Standardised Questionnaire), defined by two attributes, namely (i) a data.table containing the data in key-value form (one column per IDDD, per qualifier and for the value) and (ii) the data dictionary (DD). A raw version of this class has the same structure but with the data in a three-column data.table, one for the IDDD, one for the string of qualifiers, and one for the value. Notice the heavy dependence of the whole architecture on the package data.table [14] to define subclasses of data.table. Objects rawStQ and StQ of different time periods can be combined to form objects of class rawStQList and StQList, respectively. See figure 1. More details about this data model implementation will be given elsewhere. We focus on the editing tools.

IV. THE EDITING TOOLS

A. The principles

28. The implementation of the preceding editing methodology has been undertaken pursuing modularity and extensibility as much as possible not only with our own packages but trying a straightforward incorporation of other packages (SeleMix, validate, ...). One of the central decision regarding

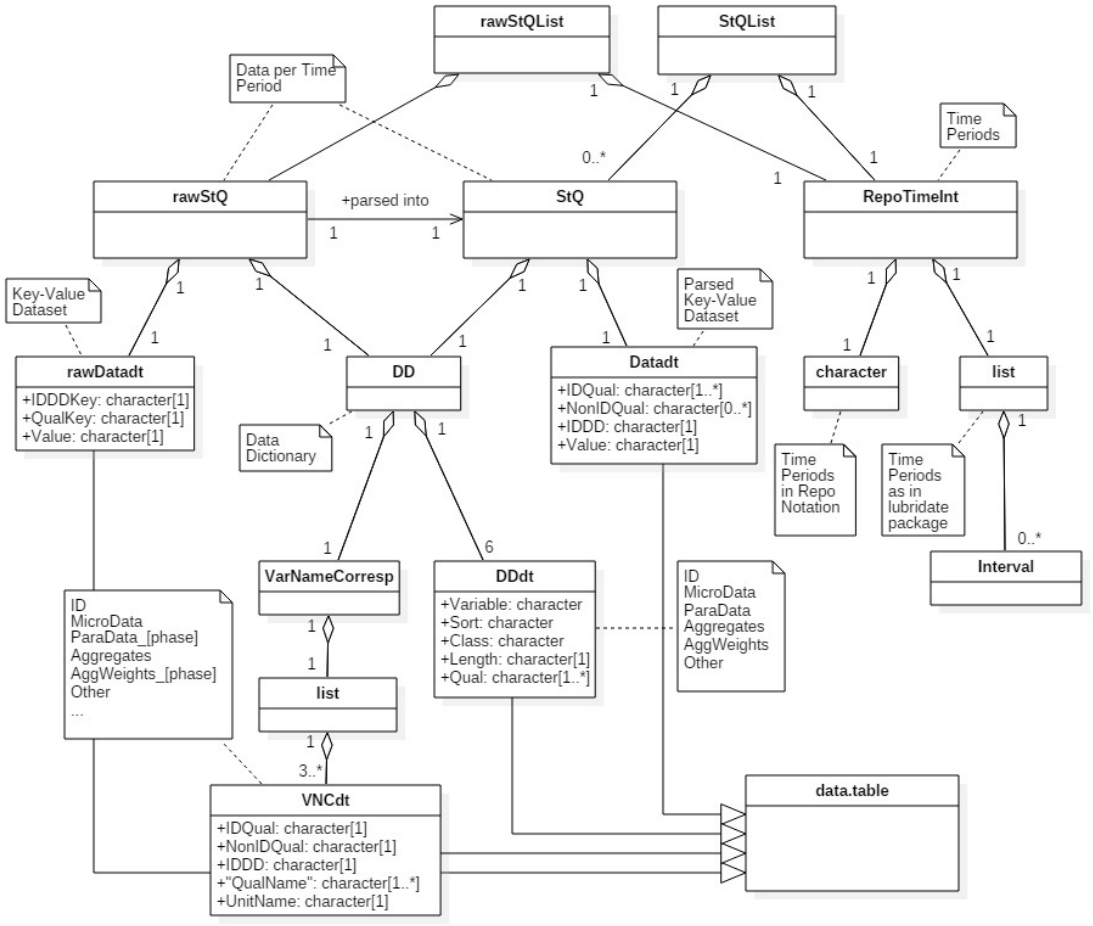

FIGURE 1. Class diagram to implement the key-value data pair.

the implementation is the *granularity* of the packages, functions, classes and methods. Our criterion has been to root the granularity in the statistical methodology itself.

29. With this criterion we easily recognize the following functionalities in the preceding editing methodology:

- For the longitudinal phase:
  - To make the point predictions $\hat{y}_{kT}^{(q)}$ and their standard deviations $\hat{\sigma}_{kT}^{(q)}$.
  - To compute the validation intervals $I_{kT}^{(q)}$. Notice that this requires basically to compute the adjustment factor $\xi_{kT}^{(q)}$.
  - To implement the distance between values and intervals.
  - To compute the thresholds $t_k^{(q)}$. Notice that this requires basically to compute the quantiles of the distance distributions.
- For the cross-sectional phase:
  - To estimate the observation-prediction model parameters p_k, σ_k, ν_k, $\hat{y}_k$ for each variable.
  - To compute the conditional error moments M_{kk} for each variable.
  - To prioritise units according to a unit score function S.
  - To make the selection of units according to the given restrictions (basically maximum number of units possibly edited in an interactive way).

30. To standardise the use in production we implement each single computation as follows: indicating the specific computation by the name of a function or a method (say, Computation) upon a particular data object (say, DataObject) with a concrete set of parameters in the form of an adequate object (say, ParameterObject), the computation is executed by calling Computation(DataObject, ParameterObject) returning the result of the computation (possibly incorporated in the updated data object).

31. Let us illustrate this approach with a concrete example. To compute the parameters of the observation-prediction model for continuous variables explained above. All parameters to be computed are contained in an object of class contObsPredModelParam: this is the data object in the preceding point[1]. Now to compute each parameter p_k, σ_k, ν_k, $\hat{y}_k$ we define a parameter object ObsErrSTDParam for the computation of the observation error standard deviations σ_k, a parameter object ErrorProbParam for the computation of the error probabilities p_k, a parameter object PredParam for the predicted values $\hat{y}_k$ and the prediction error standard deviations ν_k and a parameter object to fix the design weights ω_k. The corresponding computation is carried out by calling to the respective method ComputeErrorSTDParam, ComputeErrorProb, ComputePredParam, and setDesignW, respectively. For example, by calling ComputePredParam(contObsPredModelParamObject, PredParamObject) we obtained a new contObsPredModelParam object with the predicted values $\hat{y}_k$ and the standard deviations ν_k incorporated.

32. These parameter classes are indeed abstract classes and their concrete implementation depends on the particular statistical methods used to make the computations. In figure 2 we can see that the computation of both the observation error standard deviations σ_k and error probabilities p_k are carried out using maximum likelihood estimations over the historical data sets (hence their names). In the case of the predictions, two methods can be used, namely a prediction based either on linear regression models or on time series techniques (hence their names). In this way we reach the modularity and extensibility: we can define new subclasses of these abstract classes to introduce new methods (e.g. we could define a new class to predict based on the models in the package SeleMix [15]).

33. Notice in figure 2 that more classes are defined at a lower level. In the computation of these parameters, many missing values arise due to different circumstances. These missing values are imputed to get a full final data set. The imputation method implementation follows the same principles. We define an abstract class to contain the imputation parameters: a mean imputation method in the figure, but more are currently produced as different subclasses.

34. We see three immediate benefits from this approach for the statistical production process. Firstly, we clearly set up two levels of work regarding these applications. On the one hand, we have a high-level use of setting parameters and applying the editing methodology in production by the production units themselves in, say, high-level scripts where the actual code implementation is transparent for them. This is especially adequate for domain experts and survey conductors without a strong computer science background. On the other hand, methodologists and computer scientists (hopefully data scientists in the near future) can work at a lower level extending and making the system evolve with new proposals or adjustments proposed by the methodologists, domain experts, or survey conductors. Secondly, the syntax is similar across the different statistical operations (indeed the statistical methodology is the same for all statistical operations) impinging thus on the normalization of the production process. Thirdly, by explicitly defining classes for the parameters of each computation, the process metadata are straightforwardly expressed improving the traceability of the process and allowing us more easily to feed the metadata system. Notice how the inputs (both data and parameters), the output and the throughput are indeed the key ingredients in this approach.

[1]The choice of name may appear inadequate at this point, but this object will indeed be later used as a parameter object, which is its main role in the methodology.

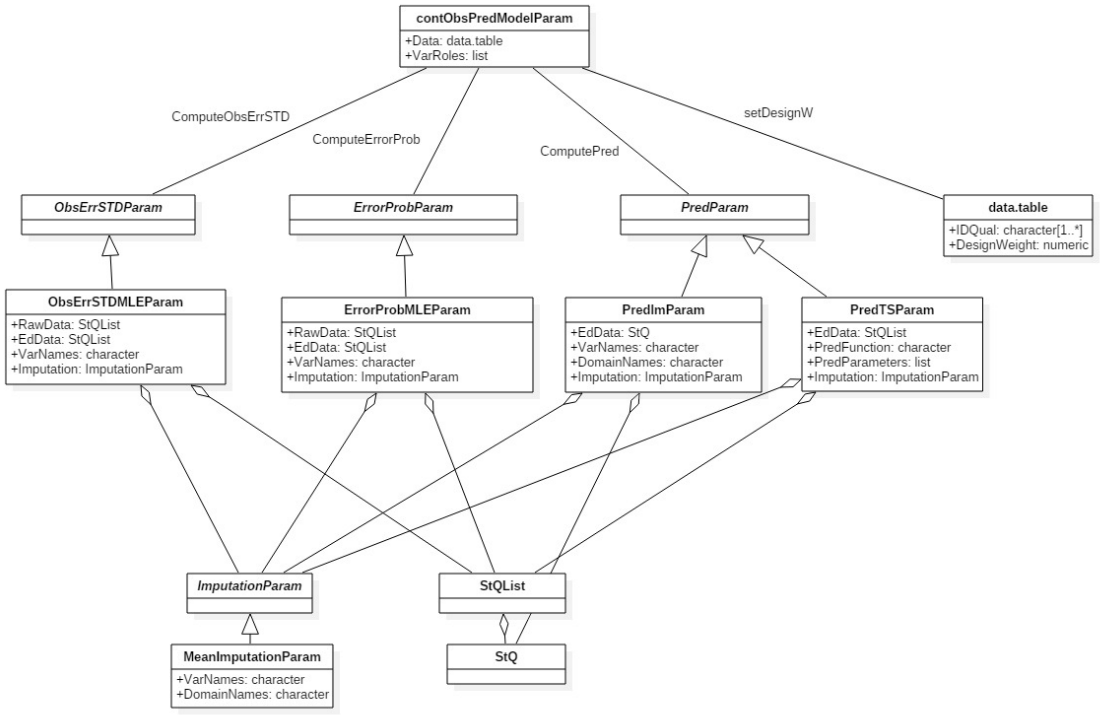

FIGURE 2. Class diagram to implement observation-predition model for continuous variables.

B. The packages

35. Following the preceding principles several packages have been developed at Statistics Spain and are under intense evolution. In the following we include a high-level description of each one.

36. Package contObsPredModelParam [16]
This package implements the observation-prediction model for continuous variables explained above. In particular, it defines a class contObsPredModelParam for the model parameters together with the methods for their respective computation. For each parameter it defines an abstract class and methods to implement the computation using different methods. So far, maximum likelihood estimation is used for error probabilities and error observation standard deviations. Classes and methods for the predictions are defined in the package StQPrediction (see below).

37. Package StQPrediction [17]
This package defines a class and methods for the predictions inside the observation-prediction model. In particular, it defines an abstract class PredParam and a generic method Predict to produce the predictions. As of this writing, predictions based on linear regression models are currently included and predictions based on time series techniques are being refactored from the old version of the data model to the current improved version.

38. Package StQImputation [18]
This package implements different imputation methods both upon data.tables and upon StQ objects. In particular, it defines an abstract class ImputationParam and a generic method Impute to produce the imputation. As of this writing, mean imputation methods are included and an imputation method based on quantiles over benchmark variables is being refactored.

39. Package SelEditErrorMoment [19]
This package implements the computation of the conditional error moments using a contObsPredModelParam object. It defines an abstract class ErrorMomentParam and a generic method ComputeErrorMoment to produce the moments as an object of class ErrorMoment. As of this writing, computation of moments under the absolute difference loss function are currently implemented. The possibility to use a squared difference loss function was analysed in the research phase (see [4], but it has not been used in production. This package makes use of numerical functions to carry out the computation. These numerical functions are part of the package SelEditFunctions (see below).

40. Package SelEditFunctions [20]
This package contains diverse numerical functions used in other packages. In particular, it contains both the Kummer function ${}_1F_1(a;b;x)$ and the Minkowskian unit score function $S(\mathbf{x};\alpha;\mathbf{w}) = \left(\sum_{q=1}^{Q} w_q y_k^{\alpha}\right)^{\frac{1}{\alpha}}$.

41. Package SelEditUnitPriorit [21]
This package implements the prioritization of units using an object of class ErrorMoment and the unit score function as input. The latter is specified as an object of class UnitPrioritizationParam. The prioritization of units, implemented in the generic method PrioritizeUnits, can possibly be computed in independent population domains according to the specific survey needs. The prioritized units are contained in an object of class UnitPrioritization.

42. Package TSPred [22]
This package contains prediction functions and wrappers for prediction functions in other packages to be used upon the classes data.table and StQList. As of this writing, regular, seasonal, and both difference random walk models are implemented together with an automatic ARIMA modelling prediction based on the package forecast [23].

43. Package BestTSPred [24]
This package implements the class BestTSPredParam to compute the best prediction among those time series predictions specified as input and computed using the package TSPred.

V. CONCLUSIONS

44. We have presented a set of R packages implementing the optimization-based approach of selective editing developed at Statistics Spain last years. After a first implementation upon a first key-value pair data model, this data model has been improved by making an extensive use of statistical metadata and normalized classifications to compose the key of each single datum in the production process. Correspondingly, the editing tools used in production so far are being adapted.

45. The stress in the design of this implementation has been placed on the granularity and architecture of the diverse computer routines, functions, classes and methods combined in packages. Our criterion is to let the statistical methodology itself to determine this granularity. Thus, each homogeneous block in the methodology has been mapped to a package.

46. Furthermore, an object-oriented approach in combination with the functional paradigm allows us to implement the processes in a standard way as a production task (a computer function or procedure) over a data set (a data object) according to a set of parameters (a parameter object). The computer applications are thus closer to a normalized process metadata model.

47. From a general though strategic point of view for the modernization of the statistical production process, we find it fundamental the interplay between the statistical methodology and its computing implementation.

References

[1] I. Arbués, M. González, and P. Revilla. A class of stochastic optimization problems with application to selective data editing. *Optimization* **61**, 265–286 (2012).

[2] D. Salgado, I. Arbués, and E. Esteban. Two greedy algorithms for a binary quadratically constrained linear program in survey data editing. *Statistics Spain (INE) Working Paper 03/2012.*

[3] I. Arbués, P. Revilla, and D. Salgado. Optimization as a theoretical framework to selective editing. *UNECE Work Session on Statistical Data Editing*, WP1, 2012.

[4] I. Arbués, P. Revilla, and D. Salgado. An optimization approach to selective editing. *Journal of Official Statistics* **29**, 489–510 (2013).

[5] R. López-Ureña, M. Mancebo, S. Rama, and D. Salgado. Application of the optimization approach to selective editing in the Spanish Industrial Turnover Index and Industrial New Orders Received Index Survey. *Statistics Spain (INE) Working Paper 04/2014.*

[6] P. Revilla. Developing a theoretical frame work for selective editing based on modelling and optimization. *UNECE Work Session on Statistical Data Editing*, WP19, 2015.

[7] D. Salgado. A modern vision of official statistical production. *Statistics Spain (INE) Working Paper 03/2016.*

[8] E. Esteban, M. Novás, S. Saldaña, D. Salgado, and L. Sanguiao. A lightweight key-value pair data model implementation for a standard production process. To be presented at *New Techniques and Technologies for Statistics Conference, Brussels, 14-16 March 2017.*

[9] T. de Waal, J. Pannekoek, and S. Scholtus. *Handbook of statistical data editing and imputation.* Wiley, 2011.

[10] E. Esteban, S. Saldaña, and D. Salgado. *RepoTime: Implementation of a notation for time intervals.* R package version 0.2.0. https://github.com/david-salgado/RepoTime.

[11] E. Esteban, S. Saldaña, and D. Salgado. *StQ: Tools to manage metadata-incorporated key-value pair datasets.* R package version 0.3.15. https://github.com/david-salgado/StQ.

[12] E. Esteban, S. Saldaña, and D. Salgado. *RepoReadWrite: Read and write files from/to the microdata repository.* R package version 0.3.4. https://github.com/david-salgado/RepoReadWrite.

[13] E. Esteban, S. Saldaña, and D. Salgado. *RepoUtils: Implementation of tools to map and work with repositories.* R package version 0.1.2. https://github.com/david-salgado/RepoUtils.

[14] M. Dowle and A. Srinivasan. *data.table: Extension of 'data.frame'.* R package version 1.10.0. https://CRAN.R-project.org/package=data.table.

[15] U. Guarnera and T. Buglielli. *SeleMix: Selective Editing via Mixture Models.* R package version 1.0.1. https://CRAN.R-project.org/package=SeleMix.

[16] E. Esteban, S. Saldaña, and D. Salgado. *contObsPredModelParam: Class and methods for the parameters of a continuous observation- prediction model.* R package version 0.0.1. https://github.com/david-salgado/contObsPredModelParam.

[17] E. Esteban, S. Saldaña, and D. Salgado. *StQPrediction: Define S4 classes and methods to make predictions.* R package version 0.0.1. https://github.com/david-salgado/StQPrediction.

[18] E. Esteban, S. Saldaña, and D. Salgado. *StQImputation: Classes and methods to implement different imputation methods upon StQ objects.* R package version 0.0.1. https://github.com/david-salgado/StQImputation.

[19] E. Esteban, S. Saldaña, and D. Salgado. *SelEditErrorMoment: Compute the conditional measurement error moments under the optimization approach to selective editing.* R package version 0.0.1. https://github.com/david-salgado/SelEditErrorMoment.

[20] E. Esteban, S. Saldaña, and D. Salgado. *SelEditFunctions: Functions for selective editing.* R package version 0.0.1. https://github.com/david-salgado/SelEditFunctions.

[21] E. Esteban, S. Saldaña, and D. Salgado. *SelEditUnitPriorit: Classes and methods to implement unit prioritization.* R package version 0.0.1. https://github.com/david-salgado/SelEditUnitPriorit.

[22] E. Esteban, S. Saldaña, and D. Salgado. *TSPred: Point and std prediction of time series.* R package version 0.2.5. https://github.com/elisa-esteban/TSPred.

[23] R.J. Hyndman and Y. Khandakar. *Automatic time series forecasting: the forecast package for R.* Journal of Statistical Software, **26**, 1–22 (2008).

[24] E. Esteban, S. Saldaña, and D. Salgado. *BestTSPred: Construction of objects of class BestTSPredParam.* R package version 0.0.1. https://github.com/elisa-esteban/BestTSPred.