

# THE BANFF SYSTEM FOR AUTOMATED EDITING AND IMPUTATION

Robert Kozak<sup>1</sup>

## ABSTRACT

Banff is a generalized system recently developed at Statistics Canada for the automated editing and imputation of quantitative survey data. It evolved from the Generalized Edit and Imputation System (GEIS), which has been used at Statistics Canada since the late 1980s. The system is a collection of nine independent, specialized SAS procedures that provide functionality similar to GEIS. However, Banff is much more flexible with respect to the operating environment and ease-of-use. This paper briefly reviews the initial development of Banff, and describes each of the system functions in detail. An overview of the methodology behind the functions is presented. Finally, there is a discussion about the current and future development of the system which is targeted at making Banff even more versatile, including the development of new methodologies and a graphical interface.

**KEY WORDS:** Editing, Error localization, Imputation, Generalized system, Outlier detection, Pro-rating, Survey data, SAS procedures.

## RÉSUMÉ

Banff est un système généralisé développé récemment à Statistique Canada pour la vérification et l'imputation automatique des données d'enquête quantitatives. Le système représente une évolution par rapport au Système généralisé de vérification et d'imputation (SGVI) qui est utilisé à Statistique Canada depuis la fin des années 80. Le système est une collection de neuf procédures SAS spécifiques et indépendantes qui offrent des fonctionnalités similaires au SGVI. Cependant, Banff est beaucoup plus flexible par rapport à l'environnement d'opération et aussi plus convivial. Cet article passe brièvement en revue le développement initial de Banff et décrit en détail les fonctionnalités du système. Un survol de la méthodologie associée à chacune des fonctions est présenté. Finalement, il y a une discussion sur les développements présents et futurs du système qui doivent rendre Banff encore plus versatile, incluant le développement de nouvelles méthodologies et d'une interface graphique.

**MOTS CLÉS :** Détection des valeurs aberrantes; données d'enquête; imputation; localisation des erreurs; procédures SAS; prorata; système généralisé; vérification.

## 1. INTRODUCTION

The Banff system for automated editing and imputation is a generalized system which can be applied to multiple survey applications, and consists of nine SAS procedures developed at Statistics Canada. Each of these procedures can be used independently or in combination to satisfy the requirements for the editing and imputation of a survey collecting primarily numerical data.

The first production release of the Banff system was in 2002. Banff is in fact a descendant of the Generalized Edit and Imputation System, or GEIS (Kovar, MacMillan and Whitridge, 1988), which has been used at Statistics Canada since the early 1990s, but which is now being phased out. For that initial release of Banff, the goal was to reproduce the methodology used in GEIS, but in a more user-friendly and flexible system. Thus, the current methodology of Banff is nearly identical to that of GEIS, with a few exceptions due to new initiatives undertaken since the first release of the system. However, there are several major structural differences between the two systems. The first is that while GEIS uses Oracle as its underlying database structure, Banff is based on the SAS architecture. Also, the modules in GEIS are linked to one another, while the individual SAS procedures in Banff are independent of one another. Finally, while both Banff and GEIS are available on the mainframe and UNIX computing platforms, only Banff is available in the Windows computing environment on the personal computer. Due to these differences, Banff is intrinsically much simpler to use and more flexible than GEIS.

---

<sup>1</sup> Robert Kozak, Business Survey Methods Division, Statistics Canada, Ottawa, Ontario, Canada K1A 0T6 (Robert.Kozak@statcan.ca)

## 2. TECHNICAL ASPECTS

The nine SAS procedures in Banff are intended to cover the survey development and data processing phases of edit development, outlier detection, error localization, imputation and pro-rating adjustment of data. As already stated, and it must be emphasized, the Banff procedures are independent of one another. The user can apply any or all of them in their data processing, and also in any order. However, it must also be emphasized that the outputs of one procedure can be used as the inputs to another to build an edit and imputation system for a survey centred around the Banff procedures. Any data imputed by Banff are output in “transaction” datasets; that is, the datasets contain only records which have had data changed. While this provides the user with great flexibility, it also entails that the user have more responsibility for providing “clean” and appropriate inputs to the system. It is often the case that an intermediate program is run between executions of Banff procedures to reconcile the inputs and outputs. For example, it is usually necessary to perform an update of the input data file and also of the input field status dataset, if applicable, through a SAS datastep once a Banff procedure has been run and new outputs have been produced, and before the next procedure is run.

Banff is compliant with both SAS version 8 and version 9. The Banff procedures have the same “look and feel” as any of the regular SAS procedures, such as Proc Means, Proc Summary, etc., though the underlying core of their program code is in the “C” programming language. As well, the Banff procedures can make effective use of the “BY” statement in SAS to more efficiently process the survey data. The inputs and outputs of Banff are in the form of SAS datasets, although the inputs in their original form can be in any format that SAS can accept (e.g. Oracle databases, text files, etc.). For reference, a simple example of the SAS program code for one of the Banff procedures appears below:

```
proc outlier  
  data=outlierdata  
  outstatus=outlierstatus  
  method=current  
  mii=1.5  
  mei=1.3  
  mdm=0.05  
  boundstat;  
  id ident;  
  var X1 X2;  
  by province;  
run;
```

Data to be edited by Banff are assumed to be numeric, continuous and non-negative, and the edits used in Banff must be expressed in linear form. These limitations are generally the same for the outlier detection and imputation procedures. However, there are currently several exceptions to the non-negativity constraint; these are the Estimator Imputation procedure, the Pro-Rating procedure, and the “current method” in the Outlier Detection procedure, where negative data can be processed. The next major release of Banff will be able to process negative data in all procedures.

When running Banff, the user can select from one of three methods. The first is through a regular SAS session, by writing and submitting Banff programs in the SAS Program Editor window. The second is through using the “wizards” of SAS Enterprise Guide. The wizards are a tool which aids a programmer in writing their SAS programs through a graphical interface. The wizard method of running Banff is recommended mainly for small applications, for learning about Banff, or for demonstrations of the system. The third method is through the use of the “Banff processor”, which is essentially a series of metadata tables through which the user supplies all of the information required to run the Banff procedures, such as which procedure to run, the parameters, inputs and outputs, etc. The processor then generates the SAS program code from the metadata. This not only allows faster initial development, but also lets the user make changes to the procedure parameters and other inputs much more easily than would otherwise be the case, such as instead having to locate and re-write blocks of SAS program code. The Banff processor is currently being used for a single specific application at Statistics Canada, but is being generalized to be able to be applied to others.

### 3. THE BANFF PROCEDURES

Each of the Banff procedures is described below. More-detailed descriptions of the methodology behind the procedures is given in Banff Support Team (2003). As noted, the user can apply any or all of the Banff procedures to their data processing, and also in any order. For the purposes of this document, however, it is useful to present the procedures in the order that they might be logically used if one applied all nine to a survey's data.

#### 3.1 Proc Verifyedits – Edit Specification and Analysis

A useful initial step when using the editing functions of Banff is to analyze what relationships between the data fields might describe an “acceptable” record. These relationships are referred to as edits, and can be determined from analysis of the questionnaire, analysis of existing data and also subject matter knowledge. Proc Verifyedits can help the user to understand these conditions and to ensure that the edits accurately represent the constraints imposed on the data. In Proc Verifyedits, there is no actual use made of the survey data, rather it is only the edits themselves which are analyzed.

As stated earlier, data to be edited by Banff are numeric, non-negative and continuous. The user specifies edit rules which must be satisfied by the responses in the fields of each individual record. These edits must be linear equalities or inequalities such as:

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n \leq b \quad (1)$$

where  $x_1$  to  $x_n$  are the  $n$  responses supplied to the survey by a sampled unit, and  $a_1$  to  $a_n$  and  $b$  are constants specified by the user. Edits of the form  $x_i \geq 0$  ( $i=1, \dots, n$ ) are automatically added to the group of edits specified by the user for each field that is being edited, due to the positivity constraint. Therefore, data which are negative or missing always fail at least one of these positivity edits. The user must also indicate whether an edit describes a pass or a fail condition. The linearity restriction holds for any of the other Banff procedures which make use of edits, and is due to the linear programming techniques which are utilized in Banff. However, there are certain ways of respecifying some types of non-linear edits, such as using logarithms or close but slightly-more restrictive approximations of the original edits, to satisfy the linearity restriction.

Normally a user would specify a system of equalities and inequalities which would define a “feasible region”, or “acceptance region”. That is, this system defines what an acceptable record would be. In general, this system with  $n$  variables and  $m$  user-specified equalities/inequalities could be written as (note that Banff has added the positivity edits for each of the  $n$  variables):

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &\leq b_1 \\ \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n &\leq b_m \\ x_1 &\geq 0 \\ \vdots & \\ x_n &\geq 0 \end{aligned} \quad (2)$$

Once the system of edits has been specified, one of the functions within the Verifyedits procedure checks that the edits are consistent with each other. If so, the procedure then identifies any redundant edits, deterministic variables or hidden equalities. From this information, the minimal set of edits may be determined. In other words, the smallest number of edits necessary to define the feasible region is identified in order to increase the efficiency of other procedures which will use the edits.

Another of the Proc Verifyedits functions will generate all of the extremal points, or vertices, of the feasible region. These points represent the most extreme data records which would be acceptable. This may give the user a better understanding of the shape of the feasible region which is being specified, especially in the case of multi-dimensional feasible regions.

The extremal points are determined through the use of Chernikova's Algorithm (Chernikova, 1964 and 1965). A complete description of its use in Banff may be found in Schiopu-Kratina and Kovar (1989).

Finally, the Verifiedits procedure generates additional edits which are implied by the user-specified group of edits. These implied edits can be inspected by the user to ensure that all relationships implied by the group of edits are acceptable. Once again, Chernikova's Algorithm is employed in this function.

### **3.2 Proc Editstats – Edit Summary Statistics Tables**

In the Verifiedits procedure, the linear edits themselves are examined with no reference to data. The Editstats procedure, on the other hand, applies a group of edits to respondent data records and determines if each record passes, misses or fails each edit. A record misses an edit when the sole reason for the record failing the edit is due to a missing value, and a record has an overall status of "miss" when all edit failures are due solely to missing data. "Fail" status for an edit is achieved when the edit fails due to one or more non-missing values, and fail status for a record occurs when one or more of the edit statuses for a record is "fail". The procedure produces five tables summarizing the results which may be used to fine-tune the group of edits, to evaluate the effects of imputation, or to calculate an estimate of the resources required for the error localization and imputation procedures. These five tables give:

- 1) the number of records which passed, missed and failed each edit,
- 2) the distribution of records which passed, missed and failed a given number of edits,
- 3) the number of records with pass, miss and fail overall record status,
- 4) the number of times each field was involved in an edit which passed, missed or failed, and
- 5) the number of times each field contributed to the overall record status.

For fine-tuning the edits, the user may observe the failure rates which occur when the edits are applied to a set of data records. A high failure rate for one particular edit might indicate that the user should modify that edit by relaxing the edit somewhat, or vice-versa. If this is done, it would then be advisable to resubmit the new group of edits to edit analysis. The Edit Summary Statistics Tables can also be used to estimate how many records will fail during Error Localization, and also subsequently to estimate how many fields will be subject to imputation. This information could be used to predict how much time will be required for the execution of the error localization and imputation stages, and also to estimate overall data processing budget requirements. The third major use of Proc Editstats is to assess the effectiveness of the imputation process. This could be done after all imputation has been completed, or even between applications of imputation methods. The Edit Summary Statistics Tables at any particular imputation stage could be compared to those produced before imputation began.

### **3.3 Proc Outlier – Outlier Detection**

The Outlier Detection procedure uses the univariate Hidiroglou-Berthelot method to identify outlying observations. Rather than comparing fields within each individual record, as is done with the linear edit rules, values of selected variables are compared across records. The user can select which variables outlier detection will be applied to. Outlying values are flagged by the procedure as requiring imputation. Values which are not extreme enough to require imputation, yet are sufficiently unusual that they should not be used later in the imputation procedures as contributors to calculated parameters for estimator imputation or as donor imputation data can also be identified by the procedure.

There are three methods of outlier detection available within the overall method: Ratio, Historical Trend, and Current. If a reliable auxiliary variable is available, then the user may wish to analyse the selected variable with the Ratio method. The Ratio method compares a function derived from the ratio of the analysis variable and the auxiliary variable in each record to bounds based on the same function from the other records. The Historical Trend method is a special case of the Ratio method where the historical value of the analysis variable is taken as the auxiliary value and which compares the analysis variable's trend over time in each record to bounds based on trends from the other records. Therefore, if historical data are available, the user may choose to use the Historical Trend method. If no reliable auxiliary variable or historical value is available, then the Current method must be used and each value of the analysis variable is compared to bounds which are based on the values in the other records. As noted before, the Current method of Proc Outlier can process negative data, whereas the other two methods cannot. In all three methods, the user may control the width and the shape of the "outlier intervals" through the use of various parameters. That is, the bounds which determine outliers or unusual values are

functions of these parameters and of the data. For a detailed explanation of these methods, see Hidioglou and Berthelot (1986).

### **3.4 Proc Errorloc – Error Localization**

For each record which does not satisfy the system of edit rules, such as described for Proc Verifyedits, the Error Localization procedure identifies the fields which must be changed in each record so that the record can pass all of the edits. No data are changed in Proc Errorloc. The fields which require imputation are simply identified during the execution of the procedure, but no imputation is actually carried out.

By definition, any fields which contain values that are missing or negative will be considered as requiring imputation. Beyond those distinct conditions, to identify other fields requiring imputation and solve the error localization problem, Proc Errorloc makes use of Chernikova's Algorithm. The problem is defined as a cardinality-constrained linear program (Sande, 1979).

Banff makes use of a strategy to determine which fields need to change to satisfy the edits whereby it minimizes the number of fields to change, rather than minimizing the magnitude of the changes. This strategy is referred to as the Rule of Minimum Change (Fellegi and Holt, 1976, and Sande, 1979). This strategy attempts to preserve as much of the original data as possible. The user may have some control over this by specifying several parameters to the program. For instance, they may specify a maximum number of fields that can be changed (referred to as the maximum cardinality). If the only solution to the error localization problem exceeds this maximum, Banff will not offer a solution and will inform the user that the maximum has been exceeded. As well, weights may be assigned to individual fields to control the probability that certain fields will be selected as needing imputation where more than one possible solution exists to satisfy the edits. When weights are assigned, Banff selects the solution with the smallest weighted number of fields to change. In the case of there being more than one solution with the same minimum cardinality, Banff will randomly select one as the ultimate solution.

### **3.5 Proc Deterministic – Deterministic Imputation**

Proc Deterministic performs an analysis of each of the fields identified as requiring imputation to determine if there is only one possible value which would satisfy the edits specified by the user. If such a value can be determined, it is imputed during execution of this procedure.

It is not expected that Proc Deterministic would find acceptable imputed values for a large number of fields which require imputation. Still, it is useful to run deterministic imputation as a first step because it reduces the number of fields which will require imputation by other imputation methods. Also, the solutions that are found in this procedure may not be found by the other methods. In general, Proc Deterministic would be likely to find more solutions if there are equality edits.

### **3.6 Proc Donorimputation – Donor Imputation**

For each record which requires imputation (i.e., the recipient), Proc Donorimputation uses a nearest neighbour approach to find the record that is most similar to it and that will allow the imputed recipient record to pass the post-imputation edits specified by the user. Usually, the post-imputation edits are a slightly relaxed version of the original edits in order to increase the chance of successfully finding a suitable donor which is "close" to the recipient. This is especially true in the case of strict equality edits, where the equality could be replaced by two inequalities which form upper and lower bounds on one side of the original equality edit equation. The imputation is carried out if such a valid donor record is found. In donor imputation, all fields requiring imputation are taken from the same donor record and so the relationships between the imputed fields are maintained.

Proc Donorimputation analyzes each recipient to determine a set of fields to be used for calculating the distance from that recipient to the eligible donor records. These "matching fields" must be some or all of the valid values of the recipient. It is possible for a recipient to have no matching fields. The selection of matching fields depends on each recipient's pattern of fields to impute and the original edits. Thus, one recipient may have a set of matching fields different from another, and many different combinations of matching fields can be found among a group of recipients. The user can also specify "user-

specified” matching fields (or must-match fields) which must be included in the recipient's set of matching fields if their values are valid.

For determining which donor record is closest to a given recipient, Banff uses the  $\mathcal{L}^\infty$  norm to define the distance between two records. After ranking and then transforming the data values independently for each matching field into uniformly distributed (0,1) ranges to remove the effect of scale from the data, the distance between a record with transformed values of matching fields of  $(x_1, x_2, \dots, x_n)$  and a record with transformed values of matching fields of  $(y_1, y_2, \dots, y_n)$  is defined as:

$$\max(|x_1 - y_1|, |x_2 - y_2|, \dots, |x_n - y_n|) \tag{3}$$

This is often termed the “minimax” distance because the closest donor is the one with the smallest maximum absolute difference between the transformed values of its matching fields and those of the recipient.

Once the distances are calculated, the group of  $n$  closest donors is assembled and Banff begins searching the donor records, beginning with the closest donor, using a “k-d tree” (Friedman, Bentley and Finkel, 1977) to increase the efficiency of the search. The closest donor is examined to see whether the recipient would pass the post-imputation edits if the closest donor's values were transferred to the recipient's fields requiring imputation. If so, the imputation is performed and the system proceeds to the next recipient. If the imputed values of the closest donor do not allow the recipient to pass the post-imputation edits, then the next closest donor is tried. This process continues until a suitable donor is found or the number of donors tried has reached a limit predetermined by the user. If no suitable donor is found once the limit has been reached, the procedure stops the search and prints a message saying that the search was unsuccessful, and then goes on to the next recipient.

If a recipient has no valid matching fields, and so a distance cannot be calculated, the user can specify that Proc Donorimputation should randomly select a valid donor record from the pool of donors. Once again, the imputed values from the donor record must allow the imputed record to pass the post-imputation edits.

The user can exert some control over which records are included in the donor pool. For example, the user may choose to allow only data which is “original” data (i.e., data which has not been previously imputed) to be donated. Also, the user may pre-identify entire records as not being eligible to be used as donors. As well, values which have been identified as being “unusual” by Proc Outlier will be automatically excluded from eligibility as donor data if the user has included the appropriate flags for those values.

A minimum number and percentage of donors available in the donor pool may also be specified by the user. If these minimums are not satisfied, imputation will not proceed for the data group. This serves to ensure that the donor pool is not too diluted so that individual records will not serve as donors too frequently, which could possibly increase the variance of imputed estimators.

### 3.7 Proc Estimator – Estimator Imputation

Proc Estimator can impute multiple variables within a record during a run of the procedure using a variety of imputation estimators. However, imputation by estimator for a particular variable can be attempted only once during a run of Proc Estimator. If the variable is not imputed successfully, another estimator can be tried in a subsequent execution. This is one of the Banff procedures that can process negative data; the user may pre-designate negative data as valid or invalid data.

There are two types of estimators available in Banff: estimator functions and linear regression estimators. The estimator functions are essentially mathematical formulae, such as those for means, ratios, trends, etc. The linear regression estimators are expressions of linear regression models, and can contain any number of variables for the regressors:  $y = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n + \varepsilon$ , where  $y$  is the variable to be imputed or dependent variable,  $x_1$  to  $x_n$  are the regressors or independent variables,  $\beta_0$  to  $\beta_n$  are the regression parameters and  $\varepsilon$  is a random error term. The values of the beta parameters are calculated by applying the method of least squares. All of these estimators may use current and/or historical data. There is a selection of twenty pre-defined imputation estimator algorithms that are already coded into the system.

The user may specify their own custom-defined estimators if the pre-defined estimators do not include everything that is needed.

Unlike for Proc Donorimputation, there are no post-imputation edits to satisfy. Due to this, and because the estimators are applied independently to selected variables, the resulting imputed records may not pass the original edits. After the estimator imputation has been carried out, the imputed records may be processed through Proc Errorloc or the user's own error localization routine to determine if this has happened.

The user can designate a variable to be used as a weight in the calculation of the parameters of estimators. A typical example would be if records in the data group being imputed had different probabilities of selection in the original sampling plan. If a weight variable is chosen, then any means calculated by that particular estimator are weighted means. For the calculation of regression coefficients, this weight variable is also taken into account.

Another feature in the calculation of linear regression estimator values is the specification of a model variance variable. This variable is considered in the calculation of the regression coefficients. The variable being imputed is related to the model variance variable in that the conditional variance of the imputed variable is proportional to the model variance variable. The conditional variance for the variable being imputed is modelled as  $v_i \sigma^2$ , where  $v_i$  is the model variance variable and  $\sigma^2$  is the model variance, which is not required to be known but assumed to be the same for all records.

For both estimator functions and linear regression estimators, a randomly selected residual may be included in the estimator. This residual is added to the value of the imputed variable. This is useful in attempting to create the same variability in the imputed values as in the non-imputed values. If the randomly selected residual is to be included, Banff will first calculate the value for the variable to be imputed according to the specified estimator. Then one of the records which contributed to the parameter calculations will be randomly selected, taking its weight into account in the selection. Finally, the residual will be calculated as the difference between the actual reported value and the value estimated by the estimator for the randomly selected record, and that value will be added to the imputed value for the imputed record. If a model variance variable is included in the definition of a linear regression estimator, it will also be included in the calculation of the random residual. If the residual for the randomly selected record  $J$  is denoted by  $R_J$ , the model variance variable by  $v$ , an exponent specified by the user for the model variance variable by  $p$ , and the time period (current or historical) for the model variance variable by  $T$ , then the random residual added to the imputed variable  $y_i$  is:

$$R_i^* = R_J \sqrt{\frac{v_{iT}^p}{v_{JT}^p}} \tag{4}$$

Similar to Proc Donorimputation, the user can control which records can contribute to the calculation of parameters for each of the individual estimators. The user may choose to allow any data or only data which is "original" data to be used in the parameter calculations. As well, the user may pre-identify entire records as not being eligible to be used. And again similar to Proc Donorimputation, a minimum number and percentage of records available to calculate the required parameters must be present in order to compute the imputed value. If these minimums are not satisfied, imputation will not proceed for the affected estimator, but will proceed for all other estimators for which the minimums are met. It should also be noted that values which were identified as being "unusual" by Proc Outlier will be automatically excluded from parameter calculation if the user has included the appropriate flags for those values.

### 3.8 Proc Prorate – Pro-Rating

The Pro-Rating procedure will pro-rate groups of variables to ensure that the sum of parts adds up to the desired total within a record. A collection of equality edits to be pro-rated is defined by the user. The pro-rating edit rules are allowed to be nested to an unlimited degree, so that individual components must sum to subtotals, and the subtotals must sum to the next-higher level total. Only the components identified in an equation can be changed; the total is assumed to be correct.

The procedure can distinguish between original and previously-imputed original data and so can be applied as a post-imputation procedure, as well as a stand-alone function. For each case where a summation does not match the total, the equation in question is submitted to a pro-rating algorithm in order to proportionally rake the components to match the

total. Zero values are not modified by the process. A rounding algorithm is then applied to ensure the variables involved are output with the number of decimals specified by the user. This is another Banff procedure where negative data can currently be processed.

Since the procedure can recognize previously-imputed and original data, the user can control which of these types are actually adjusted for each one of the components through the specification of a parameter. It is also possible to apply weights to each of the components to control the amount of change for one component relative to the others. If the user is not comfortable with large changes to the incoming data, they can specify a global bound on the relative change, which is applied to all components. If these bounds are exceeded, the pro-rating is not carried out and the user is notified of the problem with identification of which specific component(s) exceeded the bounds.

### **3.9 Proc Massimputation – Mass Imputation**

Proc Massimputation is especially useful in the case of two-phase surveys, where detailed information is only collected for the second-phase sample (or subsample) selected from a large first-phase sample. An alternative to the difficulty of computing classical estimates based on subsampling weights is to mass impute the missing information for the second-phase nonsampled units through the use of donor imputation methodology. In this way, a complete data file is created for all first phase sample units. In the case of mass imputation, the records which require imputation are known, and the fields to be imputed are known and identical for all records. The set of core information collected from the entire sample and the extra items collected from the sub-sample should have already been edited and imputed. Thus, no consistency edits (nor post-imputation edits) need to be applied for the records imputed in Proc Massimputation.

Since there are no edits involved, it follows that the determination of matching fields by the system cannot be performed. However, the user may specify matching fields to be used in finding donors. The matching fields are specified as parameters of the program. In the case of no valid matching fields being available, the user can have the system randomly select a donor for a recipient. Also, like for Proc Donorimputation, the user can specify the minimum percentage and number of donor records that must be available for imputation to proceed.

Although the mechanics of the selection of the donor are very similar to Proc Donorimputation, including the calculation of the “minimax” distance, there is one important difference. This is that post-imputation edits are not required. In fact, the procedure will simply impute data from the closest donor record, or from the first randomly selected donor record in the case of no valid matching fields being available.

## **4. FUTURE DEVELOPMENT**

Since its inception as essentially a much more flexible and user-friendly version of GEIS with nearly identical methodology, Banff has continued to evolve. Although there have been numerous enhancements made to the system, including some methodological changes such as a generalization of the Hidiriglou-Berthelot method in Proc Outlier, and others like the inclusion of negative data processing capabilities in several of the procedures, there have been several other areas that have been identified for improvement.

In the next version of Banff, which is currently under development, all of the procedures will be able to process negative data. There will also be an improvement made to the Pro-Rating procedure in its handling of negative data. The SAS Enterprise Guide wizards that were created for Banff are presently a prototype. New wizards for each of the Banff procedures will also be released with the next version of Banff, utilizing the capabilities of the latest version of SAS Enterprise Guide. Finally, the Banff-related messages in the SAS log window will be bilingual; the user will have the choice of viewing the messages in French or English.

Beyond these near-future enhancements, other development activities are also being evaluated. As stated earlier, Banff is currently only able to process numerical data (and in certain cases, ordered qualitative data). While this is not a major issue for business surveys, much of the data collected by social surveys is qualitative. In order for Banff to be a truly universal editing and imputation tool, the addition of the ability for it to process, edit and impute this type of data must be considered.

Specifying edits in the linear format required by Banff can be a significant limitation for some users. Non-linear edits, such as those of the “if-then-else” type, for example, can be approximated in some cases by linear edits using some known techniques. A decision table software package called Logiplus (Systems Development Division, 2000) has been developed at Statistics Canada which allows virtually any style of edit to be specified through decision tables and converted into SAS program code. It is possible that this program could be interfaced with Banff, thus combining the usual Banff linear equations and the Logiplus interface for the definition of edits. This type of functionality will be important if development is undertaken to implement qualitative data processing in Banff.

As noted, the Banff processor is being used by one particular application at Statistics Canada. However, right now the processor is basically a customized tool for that application. There is great potential for the processor to be utilized by many other users if the processor can be generalized to some degree, since the basic concepts behind the processor are applicable to essentially any survey’s data processing stream. This would allow Banff to expand its user base more effectively as it would allow any complex application to implement and maintain an editing and imputation system centred around Banff much more efficiently.

## 5. CONCLUSION

Since its introduction in 2002, the Banff editing and imputation system has continued to evolve and improve, and will continue to do so in the foreseeable future. This evolution has included further enhancements in both the methodological and operational capabilities of the system. Along the way, more users, both internal and external to Statistics Canada, have become new clients of the system. The feedback provided by these users has been important to the system development and support teams in providing new ideas and functionality for Banff. Thus far, the clients of Banff have been generally restricted to the business survey realm, but it is expected that with the addition of qualitative data processing capabilities, Banff would become a universal editing and imputation tool without regard to the subject matter of the client survey.

## ACKNOWLEDGEMENT

The author would like to thank the referees for their insightful and helpful comments.

## REFERENCES

- Banff Support Team (2003). Functional Description of the Banff System for Edit and Imputation System. Statistics Canada, Quality Assurance and Generalized Systems Section technical report.
- Chernikova, N.V. (1964). Algorithm for finding a general formula for the nonnegative solutions of a system of linear equations. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 4, 151-158.
- Chernikova, N.V. (1965). Algorithm for finding a general formula for the nonnegative solution of a system of linear inequalities. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 5, 228-233.
- Fellegi, I.P., and Holt D. (1976). A systematic approach to automatic edit and imputation. *Journal of the American Statistical Association*, 71, 17-35.
- Friedman, J.H., Bentley, J.L. and Finkel, R.A. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transaction on Mathematical Software*, 3, 209-226
- Hidiroglou, M.A. and Berthelot, J.-M. (1986). Statistical editing and imputation for periodic business surveys. *Survey Methodology*, 12, 73-83.
- Kovar, J.G., MacMillan, J. and Whitridge, P. (1988). Overview and strategy for the Generalized Edit and Imputation System. (Updated February 1991). Statistics Canada, Methodology Branch Working Paper No. BSMD-88-007E/F.
- Sande, G. (1979). Numerical Edit and Imputation. Presented at the 42nd International Statistical Institute Meeting, Manila, Philippines.

Schiopu-Kratina, I. and Kovar, J.G. (1989). Use of Chernikova's algorithm in the Generalized Edit and Imputation System. Statistics Canada, Methodology Branch Working Paper No. BSMD-89-001E.

Systems Development Division. (2001). Logiplus User's Guide. Statistics Canada internal report.