

DRG Grouper Design with Immediate Test of Resource Explanation

Örjan Leringe, DRG System Development AB, Stockholm

Tel mob +46 70 56 463 62

Tel office +46 8 746 09 25

orjan.leringe@drgsystem.com

www.drgsystem.com

Every DRG system in active use is in a state of continuous change. DRG's are altered, added or removed - usually on a yearly basis. The main reasons for changes are developments in medical treatment and management of health care. The two most important criteria for changes are the requirements of similar resource intensity within given DRG and similar type of patients in each DRG from a clinical perspective.

A new kind of grouper software is evolving with radically increased flexibility and ease of use. We will in this presentation explore the design criteria for this software. We will also describe an implementation in DRG Management System (DMS) which fulfills these criteria [1]. In particular we will show how this kind of software will give an immediate answer to the question on whether changes in the DRG algorithm will increase explanation of resource usage.

For the DRG software we formulate the design criteria

1. *The algorithm must be formulated in a formalized manner such that the software can be built from this and only this documentation*
2. *The formulation must be understandable by people from the economic and medical professions as well as computer professionals*
3. *There must be testing procedures and a complete test-batch with cost data for verification*
4. *Changes in the algorithm must be easy to do and implement*
5. *Consequences of the changes in the DRG algorithm can easily be analysed with statistical methods*

The Nordic countries are using DMS with the strictly defined NordDRG system [2]. A number of database tables define the algorithm unambiguously. This satisfies the first design criteria. Let us look at the other criteria.

How to understand the DRG algorithm

How do you formulate the DRG algorithm in a way understandable by people from the economic and medical professions as well as computer professionals? The obvious solution is to formulate the algorithm in a flowchart. The flowchart is in addition to being intuitive also a formal presentation. But how can we go from a flowchart to a computer program without a lot of programming with staffs of programmers? The solution is to translate the flowchart to a decision table.

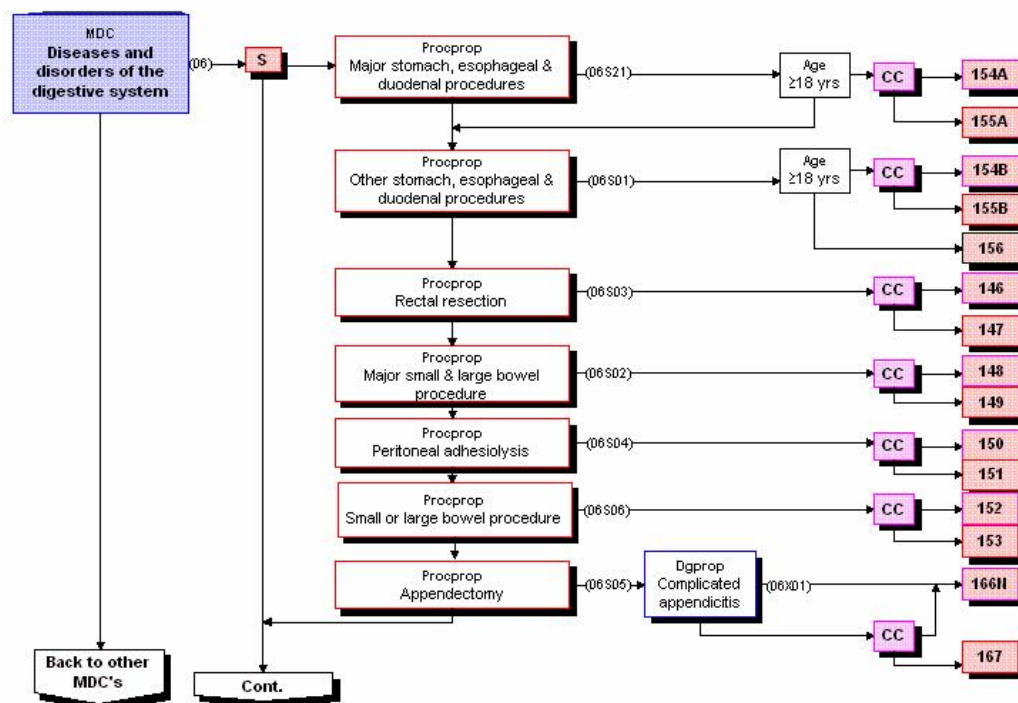
It can be shown that a flowchart always can be emulated by a decision table [3]. Finally, a decision table is computable; in particular it can automatically be translated to a computer program. This means that we can go from a decision table describing the DRG algorithm to an automatically generated computer program.

The flowchart is based on information in a number of tables stored in a database. These include lists of properties of each diagnosis and procedure. These properties are used in the flowchart as actions variables – deciding which path to take in the flowchart.

The heart in the decision process is the logic table which essentially is a decision table. In order to fully understand the grouping algorithm you have to study the logic table. The flowchart gives the general path to the computed DRG but the logic table is needed for the details. But again, this does not imply need to have programming knowledge.

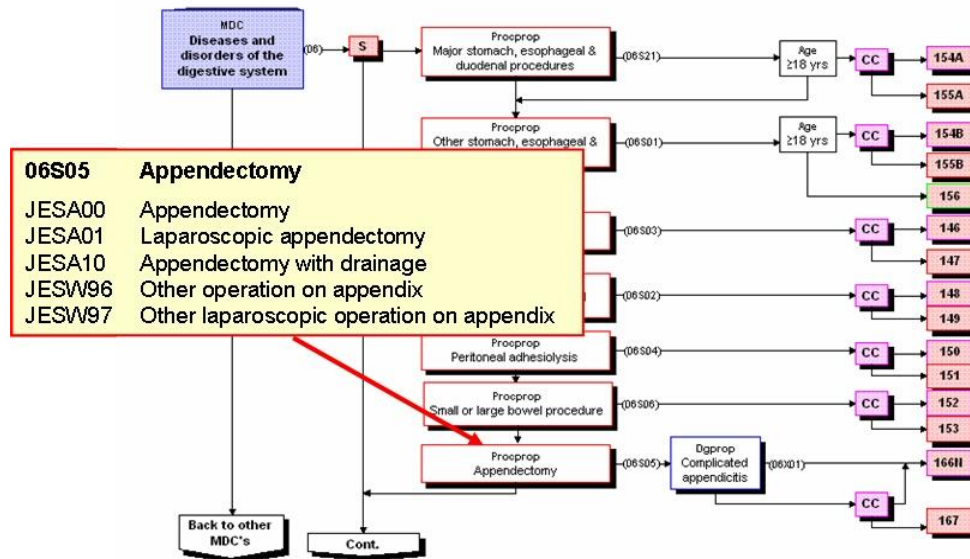
The DRG algorithm is formulated in a flowchart form. In this example we use a snapshot from the NordDRG Manual for 2007. (The full version of the manual is found at www.nordclass.uu.se).

Here we have arrived at MDC 06 and will study how we arrive finally at the DRG's 166N and 167 for appendectomies...



In the flowchart you will find decision variables of two kinds. One is based on information fetched directly from the input. An example of that is the condition "Age >= 18 years". Other decision variables come from properties attached to diagnoses or procedures. These properties are defined in special tables.

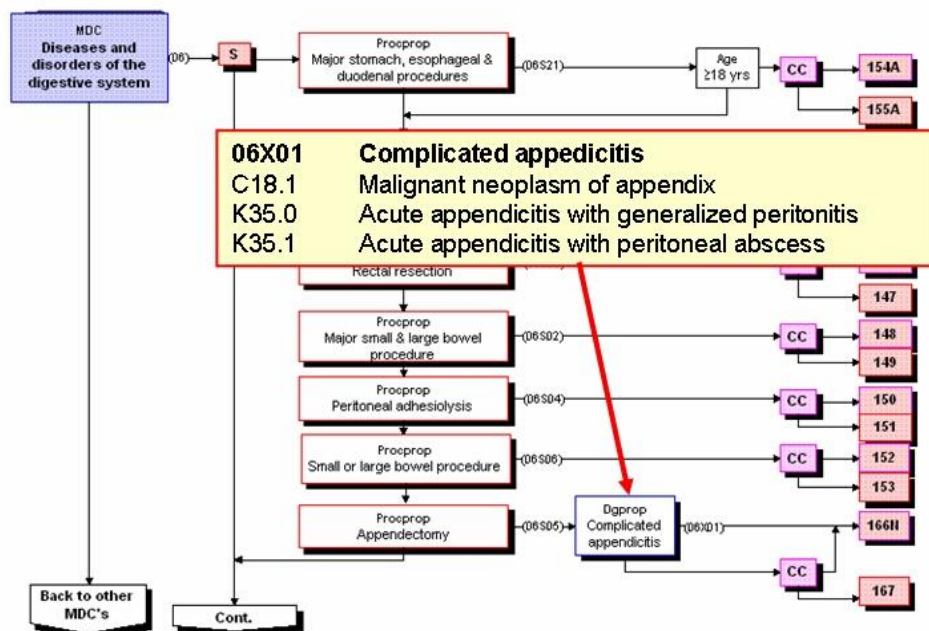
If you click on the box "Procprop Appendectomy" you will get information on the condition (property) for showing the right branch which is labeled "06S05". This decision variable defines the condition for appendectomy. The diagnoses which define the condition are displayed:



Next question is whether the DRG is complicated or not. By clicking on the box "Dgprop Complicated appendicitis" you will get information on which diagnoses will lead to a complicated DRG directly.

As you can see from the picture below there are three diagnoses which will lead to a complicated DRG: C18.1, K35.0 and K35.1. They are the diagnoses forming the property "06X01, Complicated appendicitis".

If none of these diagnoses are present there is still a possibility for this case to be a complicated DRG. This will happen if there is a secondary diagnosis which complicates the case. The mechanism behind this is shown in the ICD module and not here in this introductory example.



The steps to perform

The DMS tool allows you to define the properties for the diagnoses. The properties will, as you have seen from the example, decide which DRG we arrive at.

The procedures are assigned properties in a similar way.

The flowchart itself is represented as a decision table. A mapping from a flow-chart to a decision table is something you always can do in a more or less automatic way. We call the table which represent the flow-chart the *logic table*.

Assigning properties to diagnoses

The DMS tool is used to assign properties to diagnoses and procedures in order to make this work in the grouper software. Let us first look at how properties are assigned to diagnoses.

Highlighted in the screenshot below is the diagnosis C18.1. It has 5 different properties, shown to the right. One of them is the property called "DGPROP" with the value 06X01. This makes the diagnosis a member of the set of diagnoses which constitute the "complicated appendicitis".

06X01 Complicated appendicitis
C18.1 Malignant neoplasm of appendix
K35.0 Acute appendicitis with generalized peritonitis
K35.1 Acute appendicitis with peritoneal abscess

CODE	d_code	CHDATE	VAR TYPE	VARVAL	varvaltext
C18.1		2006-08-30 10:52	COMPL	06C02	Malignant neop
C18.1		2006-08-30 10:52	DGCAT	06M02	Malignancy of d
C18.1		2006-08-30 10:52	DGPROP	06X01	Complicated ap
C18.1		2006-08-30 10:52	DGPROP	15X39	Other significant
C18.1		2006-08-30 10:52	DGPROP	23X01	History of malign

As a primary diagnosis it excludes complications in these categories

CODE	D_CODE	COMPL	chdate	description
C18.1		06C02	2006-08-30 10:52	Malignant neoplasm of colon are

Testing

When we have assigned properties to diagnoses and procedures we want to test the new algorithm. By starting the testing module the new definitions are used to automatically construct a grouper.

The highlighted case has the diagnosis C18.1 and the procedure JESA00.

By pressing the button "Group Case" the highlighted patient record is sent to the grouper and the result is shown as highlighted below. The DRG is 166N and all the properties collected from the case is shown in the upper window.

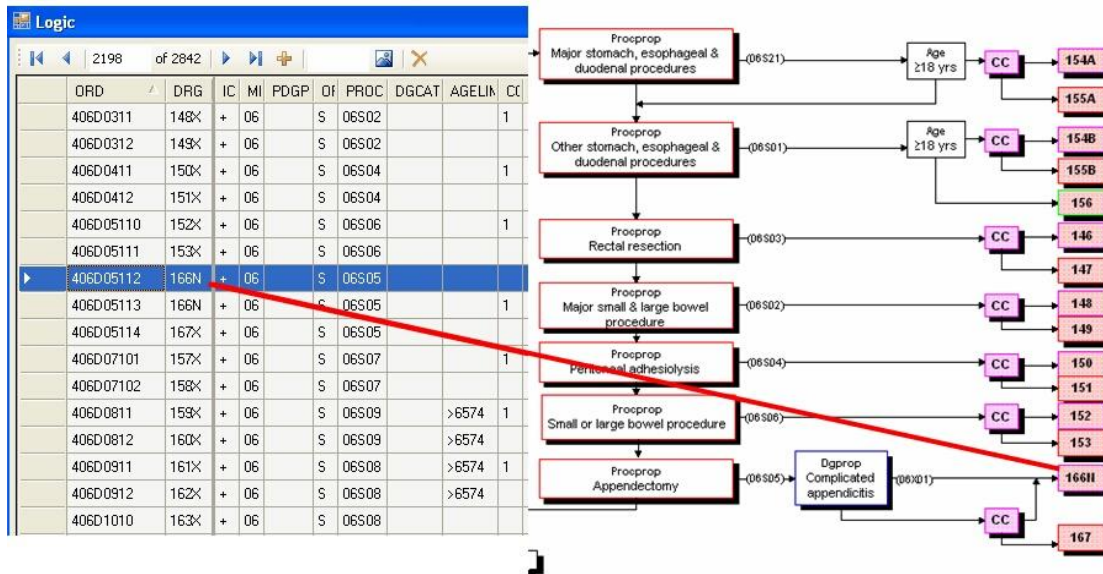
How the grouping is done

The grouping computation is performed by the grouper in two steps.

First all the properties of the diagnoses and procedures are collected.

Secondly the logic table is browsed from top to bottom for a matching row. This corresponds to reading the flowchart previously shown. The highlighted row is the first one which is matched by the properties found in the patient record. The computing stops here and the grouper returns the DRG 166N.

You can now see the correspondence between the logic table and the flowchart:



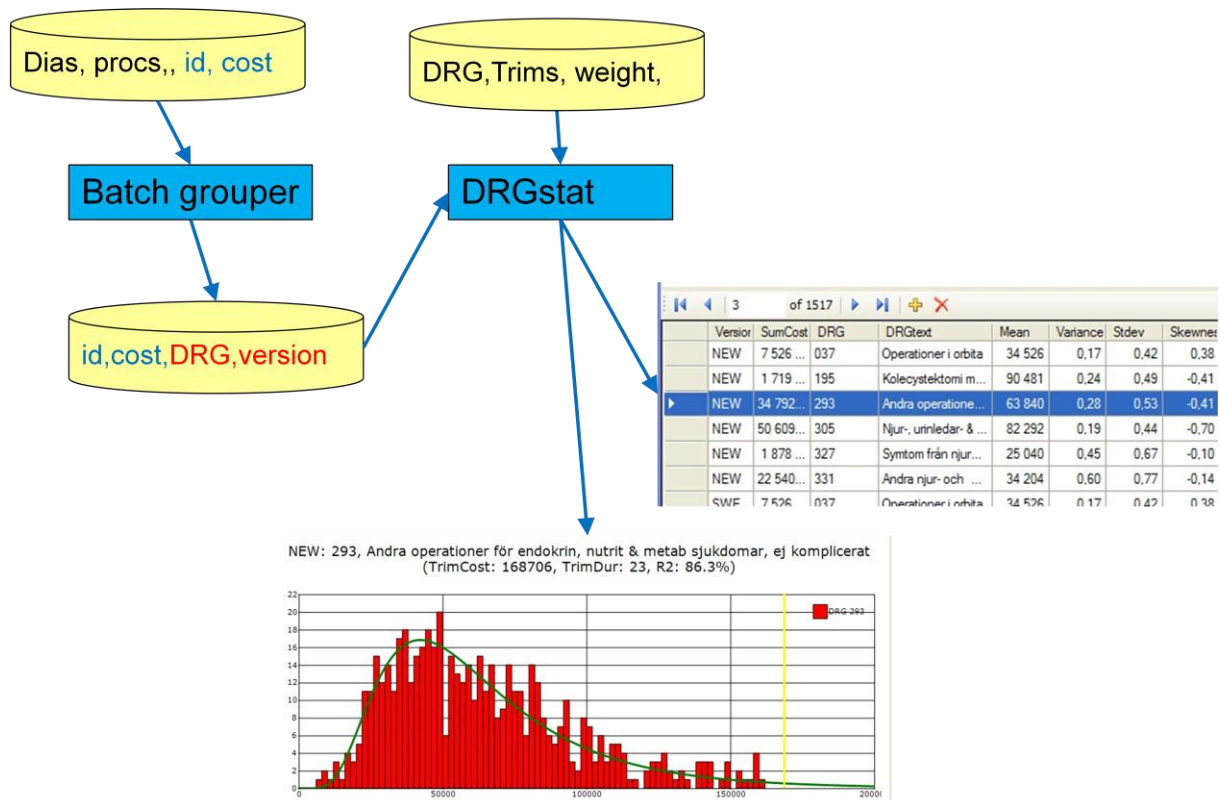
In this introductory example we have shown some main ideas but of course not all the details.

DRGstat, a program for DRG statistics

How do we know if the grouping algorithm is efficient in defining groups and putting patients with similar resource consumption in each group (DRG)? There are always lots of change proposals to the DRG algorithm every year. Thus it is critical for the success of the grouper design to early discard proposals which have negative effect on cost explanation and to verify proposals with a positive effect.

Tightly integrated with DMS is the program DRGstat which is a tool for examining cost explanation. At a given moment you have a set of groupers in various stages of development. These are called “versions”. Typical versions are previous official groupers and alfa, beta etc versions of current groupers under development.

Here is how you work. Start with choosing version and input file with test cases which have actual costs attached and start the batch grouper. In less than 5 minutes you will have the result outlined in this picture:



Calculation of total explanation values, R2 and RAR

Tools to analyze efficiency in cost explanation of the grouper are variance and regression analysis. In the DMS Statistics we calculate two values for the overall evaluation: R2 and RAR.

The R2 values are calculated with one-factor ANOVA. The DRG variable is used as factor and the patient cost as the dependant variable with the following formula:

$$R2 = 1 - \frac{\sum_{i=1}^n (x_i - m_{DRG})^2}{\sum_{i=1}^n (x_i - m_{TOT})^2}$$

Here m_{DRG} is the mean cost value of the DRG belonging to the current cost value x_i and m_{TOT} is the mean cost calculated for all the cases. Here $x_i - m_{DRG}$ expresses the residual, the individual deviation from the expected value. n is the total number of cases with cost data.

The value of R2 should near the value of 1. The value should be 0.65 or more. With R2 value of 0.7 we say that 70% of the variation is explained and 30% is unknown.

Instead of taking the square of the deviations from the means as in the first formula we can take the absolute value. The latter is called Reduction of Absolute Residuals, RAR, and is also a measure of how well the DRG algorithm can group patients with similar resource consumption. The RAR value is more robust and can be used for smaller populations like a hospital.

$$RAR = 1 - \frac{\sum_{i=1}^n |x_i - m_{DRG}|}{\sum_{i=1}^n |x_i - m_{TOT}|}$$

Depending on circumstances the RAR value should be above at least 0.45.

Estimating effect of smaller changes between versions

Often the changes in the grouping algorithm are minor. One or a few DRG:s are split, added or deleted. This means that in the new version some DRG's loose cases and some get new ones. If we compare the R2 and RAR values of the original version, call it A, and the new version, call it B, we will find that the values are changed very little. In this case we provide a new measure to better estimate the value of the changes.

We compute a RAR for all the cases in the DRG's which are part in the changes. This is done for both version A and version B. Based on this we define a new measure we call ΔRAR ("delta RAR").

Here is the exact definition:

Select all the cases in version A which does not belong to a DRG which has exactly the same cases in version A and version B. This means for instance that we select

- all the cases in a DRG in version A which has been deleted in B
- all the cases in a DRG in version A which has lost or added at least one case in B
- all the cases in a DRG in version B which belong to a new DRG

We have now formed a subset of the cases with the purpose of studying the DRG's affected by the changes and no other DRG's. For this subset we calculate the RAR value in both version A and B. The number of cases is now p and the residuals are compared with the mean value in version A and the new mean value in B. The m_{TOTAB} value is of course the same in version A and B since it is the mean of the costs for the same cases.

$$RAR_A = 1 - \frac{\sum_{i=1}^p |x_i - m_{DRG_A}|}{\sum_{i=1}^p |x_i - m_{TOTAB}|}$$

$$RAR_B = 1 - \frac{\sum_{i=1}^p |x_i - m_{DRG_B}|}{\sum_{i=1}^p |x_i - m_{TOTAB}|}$$

$$\Delta RAR = \left(\frac{RAR_B}{RAR_A} - 1 \right) * 100$$

The ΔRAR value is now a measure of *how well we have reduced the errors in cost explanation* expressed in percent. Note that the value will be negative if we have increased the errors in cost explanation.

Viewing individual DRG's with normal and lognormal distribution curves

In order to get an immediate intuitive view of how well the cost variations in individual groups follow the normal and/or the lognormal distribution we present this in the histograms.

The lognormal has this probability density function:

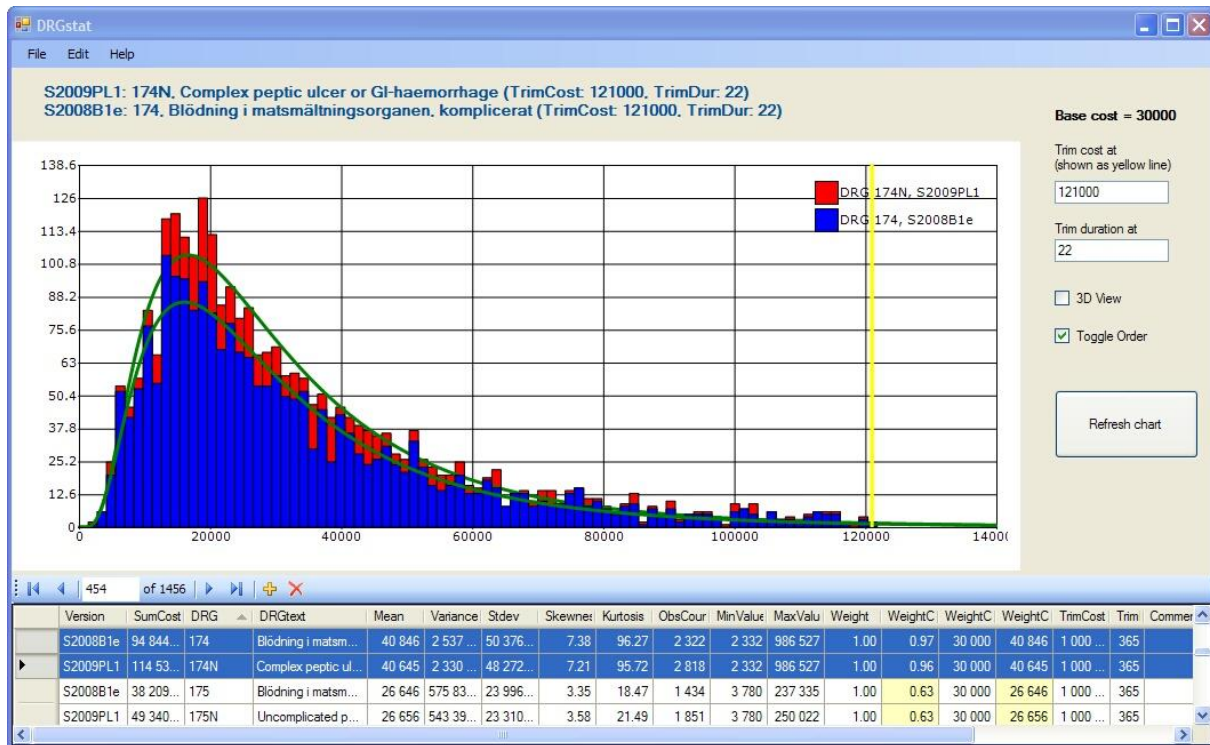
$$pdf(x) = \frac{1}{x \cdot \sigma \sqrt{2\pi}} \exp \left(-\frac{1}{2\sigma^2} (\ln(x) - \mu)^2 \right)$$

The normal distribution has this density function:

$$pdf(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

We calculate the pdf functions and multiply by total number of inner cases to get the correct scaling for drawing in the histogram.

Here is an example of histogram with lognormal distribution curve:



You select two DRG's in the table to have them compared in a histogram. The DRG 174 is in version S2009PL1 replaced by 174N. In the table we observe how the variance has decreased slightly.

In addition to calculate *mean*, *standard deviation* and *variance* of the observation, the following values in the table are calculated:

Skewness, a measure of the skew in the distribution. Negative if skew is to the left and positive if skew is to the right.

Kurtosis, a measure of sharpness of peak in the distribution. Should be as close to zero as possible. We notice that the example shows a case where this also has decreased slightly.

The example also shows how a change in the grouper algorithm often influences many groups. For instance we see that the number of cases has changed. In order to evaluate the usefulness of the change you also have to look at the overall indicators R2 and RAR. When working with one or a few changes the ΔRAR value will be even more practical as described earlier.

Deciding cut-offs for trimming outliers

The yellow line in the graph above indicates the cut-off value.

The graphing capabilities allow the user to change cut-offs for individual DRG's and immediately see the effect.

In order to set the cut-offs for all DRG's in one calculation you can use the following algorithm. For every DRG we compute this value as the cost cut-off:

$$Cutoff_i = Md_i + k * (Q3_i - Q1_i)$$

Md_i , $Q3_i$ and $Q1_i$ stand for median, upper quartile and lower quartile of the cost of DRG number i . You can then let the program compute the value k in order to get a total number of outliers to be 5% of the cases (or any other percentage you choose).

Conclusions

We have shown how software like DMS can shorten the time span between the birth of an idea on change in DRG algorithm and the moment when the usefulness of the idea is verified or rejected.

In many cases this process can be shortened from several months to a few hours of testing.

Acknowledgements

I want to thank *Kenneth Carling, professor of Statistics in Sweden*, for stimulating discussions on DRG statistics. He for instance initiated the idea of defining the ΔRAR measure for immediate feedback on change proposals in the DRG algorithm.

Also thanks to the *Nordic Center, its partners and manager Dr Martti Virtanen* who started this work and without them this work would not have been possible.

References

- [1] *User Manual for DRG Management System – DMS*
DRG System Development AB, Stockholm, www.drgsystem.com
- [2] NordDRG Users' Manual, Nordic Centre for Classifications in Health Care,
www.nordclass.uu.se
- [3] *On the Emulation of Flowcharts by Decision Tables*
Art Lew, University of Hawaii at Manoa, *Communications of the ACM*, December 1982, Volume 25 Number 12