

Automatic binning in visual predictive checks

Christian Sonehag, Niklas Olofsson, Rasmus Simander

Department of Scientific Computing, Uppsala University^{*†}

February 12, 2012

Abstract

We propose a novel automatic binning strategy for Visual Predictive Checks that mainly improves the automatic selection of the number of bins. Binning, given the number of bins, is performed starting from the construction of a data density function which is used in a optimization criteria to place bin boundaries where data is less dense in order to avoid splitting clusters of measured data into different bins. A simple but effective method for choosing the number of bins is also presented which is an important part of the algorithms performance. The proposed algorithm is demonstrated on various datasets, which were evaluated by senior modelers.

Keywords: Pharmacometrics, VPC, Visual Predictive Checks, Automatic binning

1 Background

Modeling and simulation of drug uptake, effects and elimination, so-called PKPD (Pharmacokinetic/Pharmacodynamic) modeling is becoming increasingly important in drug development. It integrates a pharmacokinetic and a pharmacodynamic model component into one set of mathematical expressions that enables the description of the time course of effect intensity in response to administration of a drug dose.

When a pharmaceutical company has used PKPD-modeling as part of the drug development process, diagnostics of the model must be included in the submission to the American Food and Drug Administration (FDA).

Visual Predictive Check (VPC) is one such method for model diagnostics. However, FDA is suspicious towards VPCs because the modelers must

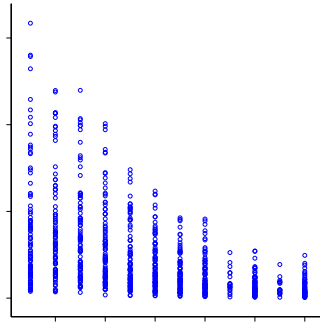
^{*}Address: Box 120, S-751 04 Uppsala, Sweden.

[†]In collaboration with: Uppsala University Pharmacometric group and AstraZeneca Södertälje

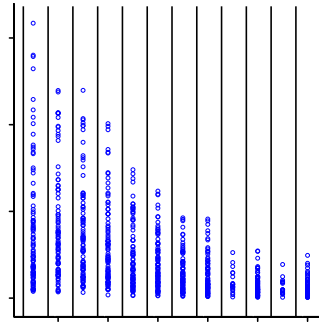
often manually bin the measured data and the choice of binning will affect the results of the diagnostic test. Because of this, FDA argues that by tweaking the binning also bad models could be made to look good. Binning the data is also very time consuming. It is therefore highly desirable for the pharmaceutical companies to have a completely automatic binning performed by an algorithm which is defined before the actual data is collected and the model is built.

2 What is VPC?

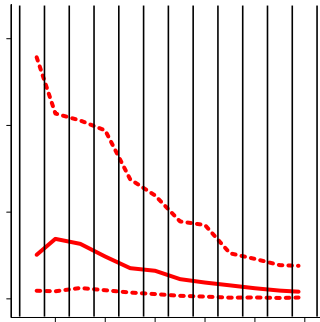
Visual Predictive Check (VPC) is a popular method for evaluating non-linear mixed effects models in pharmacometrics. It shows how well simulated data from a proposed model agree with measured data and hence if the model accurately describes the studied population and biological process.



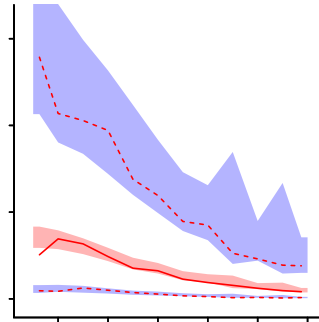
(a) Measured data



(b) Binned data



(c) Median and percentiles calculated in each bin



(d) The final VPC

Figure 1: Explanation of how VPCs are created

The VPC method starts out from a model that describes some property of a drug in the human body. This property could for instance be how the drug concentration varies over time in the blood. Several sets of data are simulated from this model. The measured data, Figure 1(a) is then grouped into time intervals, the data is *binned*. This is illustrated in 1(b). In this particular case, the procedure is trivial due to well-structured data. Normally it can very be difficult to see how an appropriate binning should be performed. From the binning the median and the percentiles are calculated for each time interval, Figure 1(c). In the next step several sets of data is simulated from the proposed model. Confidence intervals are then calculated for these medians and percentiles. Finally the median and the percentiles of the measured data are plotted together with the confidence intervals from the model (Figure 1(d)) which allows an evaluation of the correspondence between the measurements and the model. In this case it can be seen that the model correctly describes the measured data since the median and the percentiles are located inside the confidence intervals.

The reason for VPCs popularity is that the idea behind the diagnostic is simple and easily communicated to both modelers and other model stakeholders and by the retention of the original time-course profile and the y-axis units, the VPC graphs are illustrated on a relevant and easily appreciated time scale which can be powerful in guiding the modeler to the origin of a potential model misspecification [1].

3 Objectives

- In cooperation with modelers, define mathematical criteria to determine what is considered to be a good binning result.
- Implement a binning method that minimizes these criteria.
- Compare the performance of our method with the method recently proposed by M. Lavielle & K. Bleakley [2].

4 Desired binning characteristics

As far as we know, there has not yet been proposed any measure for what constitute a good binning and we had no data sets with ground truth at hand for our experiments. Therefore interviews with modelers that work with VPCs regularly were carried out. They revealed the following desired characteristics of how binning should be performed in order to get a good VPC:

- The median curve should look physiological. That is, the curve should be smooth.

- Distinct clusters of measurements should not be split into different bins.
- The binning should capture changes of the dependent variable (the y-variable).
- Fulfilling the above characteristics, the bins should have as similar amount of data points as possible.

Also to be able to calculate the percentiles, each bin needs a certain number of points (usually any bin with less than 10 points would create confidence intervals so large that they don't contribute to the VPC).

5 Existing methods for binning in VPC

In this section a description of some of the existing automatic methods for binning data is given. However, manual binning is still the most common method. This indicates that before the recent work of M. Lavielle & K. Bleakley [2] there was no existing automatic method that worked satisfyingly.

5.1 Equal width binning

The equal width binning method is the simplest method of binning where the time domain is divided into, as the name suggest, subintervals of equal width. The user specifies the number of bins K as input and the algorithm divides the interval into K subintervals of equal width.

In mathematical terms this can be written as: given K bins, the width of each bin h is calculated as

$$h = \frac{t_{max} - t_{min}}{K} \quad (1)$$

It is obvious that this method will perform poorly when the data is not evenly distributed and can in the most extreme case have several empty bins because it does not take the actual distribution into account.

5.2 Equal size binning

Equal size binning puts the same number of points in each bin. In this case also the number of bins has to be specified by the user. Given the number of bins K and the total number of points n , ideally each bin should contain

$$\frac{n}{K} \quad (2)$$

number of points. In general however, n is not a multiple of K , so some of the bins must contain either $\lfloor n/K \rfloor$ or $\lfloor n/K \rfloor + 1$ points.

If we have many measurements at the same time points but the number of measurements is different between the time points this method breaks down. Instead of looking for an *equal* size binning we can look for a *similar* size binning. Let $t_1 < t_2 < \dots < t_P$ be the P different time points and m_1, m_2, \dots, m_P the number of measurements taken at each of these time points. For a given number of bins K , we are looking for the set of bins $I = (I_1, I_2, \dots, I_K)$ that minimizes

$$\sum_{k=1}^K \left| \sum_{j \in I_k} \left(m_j - \frac{n}{K} \right) \right| \quad (3)$$

where

$$n = \sum_{j=1}^P m_j.$$

Using *similar* instead of *equal* bin size results in a method that perform relatively well. However if the data is "clustered" around various time points, this approach does not provide a plausible binning since no consideration is taken to the clustered structure of the data. This results in that clusters of measurements are most likely to be split into different bins.

5.3 Modified K-means

In a recent article by M. Lavielle & K. Bleakley [2], a new method for automatic binning was proposed. It aims at resolving the issues with the two above stated methods by incorporating "cluster-awareness".

5.3.1 Placing the bins

M. Lavielle & K. Bleakleys method takes it's starting point from the well-known K-means algorithm [4] which, given a set of observations (x_1, x_2, \dots, x_n) aims to partition the n observations into K sets, $S = (S_1, S_2, \dots, S_K)$ so as to minimize the within-cluster sum of squares defined as follows:

$$\arg \min_K \sum_{k=1}^K \sum_{t_j \in S_k} (t_j - \bar{t}_k)^2. \quad (4)$$

This is a clustering method for arbitrary dimensions but for the purpose of binning, only the 1D case is considered. How eq. 4 is minimized is explained in the appendix, section A.

The K-means algorithm works well when dealing with a model having the same variance, that is, the spread of the data inside of each cluster is similar. M. Lavielle & K. Bleakley argued that it doesn't work as well when the spread of the data inside of each cluster varies, and gave an example for

when their method performs poorly (Figure 2(a)). In order to handle this they generalize the minimization criterion (eq. 4) and propose the following criterion:

$$J_{opt,\beta}(I) = \sum_{k=1}^K n_k (\sigma_k^2)^\beta \quad (5)$$

where $\beta \in (0, 1]$ and

$$\sigma_k^2 = \frac{1}{n_k} \sum_{j \in I_k} (t_j - \bar{t}_k)^2. \quad (6)$$

We note that in the case when $\beta = 1$ we have the minimizing criterion of the K-means algorithm. By letting β approach 0, more emphasis is made on selecting bins with different variability.

5.3.2 Choosing the number of bins

M. Lavielle & K. Blekley propose a model selection approach with the following penalized criteria:

$$U(I, \lambda) = \log(J_{opt,\beta}(I)) + \lambda \beta K(I). \quad (7)$$

where $K(I)$ is the number of bins in binning I . They choose I so that $U(I, \lambda)$ is minimized for a fixed λ .

A larger λ means fewer bins. After extensive numerical trials they suggest that $\lambda = 0.3$.

6 Binning Method

We also base our method on minimizing (4). As we described earlier this method is good in cases where the data consists of clusters of measurements with similar variances. In general, this is not the case for our data and the method could split a cluster of measurements into different bins where the variability is high while merging two clusters of measurements into the same bin where the variability is low as shown in Figure 2(a).

The problem is solved by adding a penalty term to the K-means minimization criteria, which penalizes adding a bin where the data is dense. Let W be the within-group variability:

$$W_k = \sum_{i \in I_k} (x_i - \bar{x}_k)^2. \quad (8)$$

where x_i is the coordinate of the independent variable for data point i and \bar{x}_k is the mean in bin k :

$$\bar{x}_k = \frac{1}{n_k} \sum_{i \in I_k} (x_i). \quad (9)$$

Then, given the bin edges $e = (e_2, e_3, \dots, e_K)$ we want to minimize¹

$$\sum_{k=1}^K W_k + \alpha \sum_{i=2}^K \phi(e_i) \quad (10)$$

where α is a scaling parameter and ϕ is a data density function.

6.1 Data density function ϕ

The data density function ϕ is obtained by *kernel density estimation* using a Gaussian kernel [3]. That is, a Gaussian density function is placed at each data point, and the sum of the density functions is computed over the range of the data, i.e

$$\phi(x) = \frac{1}{nh} \sum_{i=1}^n \text{Kernel} \left(\frac{x - x_i}{h} \right) \quad (11)$$

where

$$\text{Kernel}(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad (12)$$

is the Kernel.

If the data consists of clusters with an equal amount of measurements n_k from Gaussian distributions with the same standard deviation σ , the optimal bandwidth for the Gaussian kernel is [3]:

$$h = \sigma \left(\frac{4}{3n_k} \right)^{1/5} \approx 1.06 \sigma n_k^{-1/5} \approx \sigma n_k^{-1/5}.$$

where the final approximation results in a slightly smaller bandwidth which increases the resolution slightly, when the distribution is not perfectly Gaussian. For our purposes this bandwidth works well for measurements distributed in any bell-shaped manner.

If the data consists of K sets of Gaussian distributed measurements (X_1, X_2, \dots, X_K) with different unknown standard deviations $(\sigma_1, \sigma_2, \dots, \sigma_K)$ a single bandwidth will be either too narrow or too wide. If the bandwidth is too narrow it will split clusters of measurements where the variance is big and if it is too wide it will result in measurements with small variance being merged with neighboring clusters of measurements. To address this problem we make use of a per bin adaptive bandwidth.

Let $h_k = \sigma_k n_k^{-1/5}$, $k = 1, 2, \dots, K$, then ideally we would like ϕ to be:

$$\phi_{ideal}(x) = \sum_{k=1}^K \frac{1}{n_k h_k} \sum_{x_i \in X_k} \text{Kernel} \left(\frac{x - x_i}{h_k} \right)$$

¹subject to the constraint that $n_k \geq 10 \forall k$

but since we don't know which x_i belongs to which X_k we can't obtain ϕ . However, we can get a usable ϕ by doing an initial binning $I_0 = (I_{0,1}, I_{0,2}, \dots, I_{0,K})$ that tries to group each X_k into a separate bin and define:

$$\phi(x) = \sum_{k=1}^K \frac{1}{n_{0,k} h_k} \sum_{x_i \in I_{0,k}} \text{Kernel} \left(\frac{x - x_i}{h_k} \right) \quad (13)$$

In all of our experiments we use the K-means algorithm to obtain the binning I_0 .

We also compensate for the inaccuracy of our first solution by doing an analysis in those bins that have data from more than one cluster of measurements. To do so we look at the kurtosis (the measure of "peakedness") defined as W_k^2/σ_k^4 . Gaussian distributed data has a kurtosis of 3. If a bin contains data from more than one cluster of measurements the kurtosis will resemble the one of the discrete uniform distribution which has a kurtosis in the interval $[1, 1.8)$ (the continuous uniform distribution having a kurtosis of 1.8). If the kurtosis indicated the data is more spread out in a bin than if there were a single Gaussian distribution we want a smaller bandwidth to resolve the fine structure of the data density. Thus we condition the bandwidth such that

$$h_k = \begin{cases} \sigma_{0,k} n_{0,k}^{-1/5} & W_{0,k}^2/\sigma_{0,k}^4 \geq C \\ \frac{1}{R} \sigma_{0,k} n_{0,k}^{-1/5} & W_{0,k}^2/\sigma_{0,k}^4 < C \end{cases}$$

where $R \geq 1$, and $C \in (1.8, 3)$. (We used $R = 4$ and $C = 2.5$ in our experiments).

6.2 Scale factor α

The method to minimize (10) can be written in short notation as

$$\text{Minimize } W + \alpha \Phi \quad (14)$$

Note that by the definition of Φ , having K clusters of data with a decent pre-binning with K bins, the local maximums of ϕ will be approximately of size 1. This can for example be seen in Figure 2(b). To relate W with Φ in the objective function $\alpha = C \max_k W_{0,k}$ was used as the scale factor. $W_{0,k}$ is here the disparity of bin k from the initial run and C a constant that was empirically determined to 7.8 for best results. The resulting binning of the final method can be seen in Figure 2(c).

6.3 Effect of the data density function

In Figure 2 the effect of the data density function is illustrated. Without the data density function only the K-means objective function is considered.

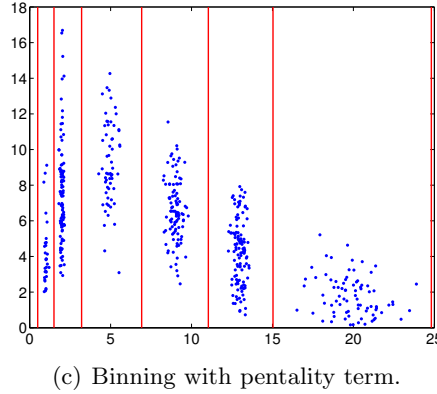
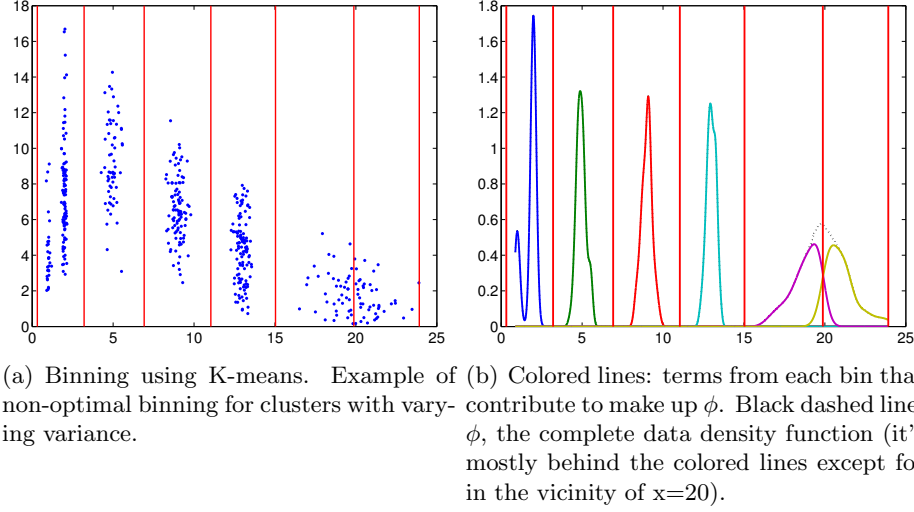


Figure 2: The effect of the potential function

In Figure 2(a) the last cluster has much spread and K-means undesirably splits it into two bins. Applying the data density function on this binning gives a contribution to the objective function which penalizes the algorithm in placing the bin on that location (Figure 2(b)). Instead the bin is placed so that the first two clusters are separated (Figure 2(c)). The binning now better corresponds to the desired binning characteristics of the modelers.

6.4 Taking the dependent variable into account

One of the desired binning characteristics from the modelers was to capture changes in the dependent variable y . Even though the data is dense somewhere we might want to place a bin there if there is a big change in y . Therefore we considered $\Delta_k y = |\bar{y}_k - \bar{y}_{k-1}|$ (where \bar{y}_k is the mean of the dependent variable in bin k) and incorporated it into our method as either

$$\text{Minimize } W + \alpha \sum_{i=2}^K \phi(e_i) - \sum_{i=2}^K f(\Delta_i y) \quad (15)$$

or

$$\text{Minimize } W + \alpha \sum_{i=2}^K \frac{\phi(e_i)}{f(\Delta_i y)} \quad (16)$$

to penalize less when there is a big change in y . We have tried numerous functions and what seemed to work best was the function

$$\text{Minimize } W + \alpha \sum_{i=2}^K \frac{\phi(e_i)}{1 + \ln(\Delta_i y + 1)} \quad (17)$$

as the value never goes below W and the logarithmic function attenuates large changes in $\Delta_k y$.

Several other $f(\Delta_k y)$ were considered and are presented in Section C together with some comments on why they didn't work. However, the general problem is illustrated in Figure 3.

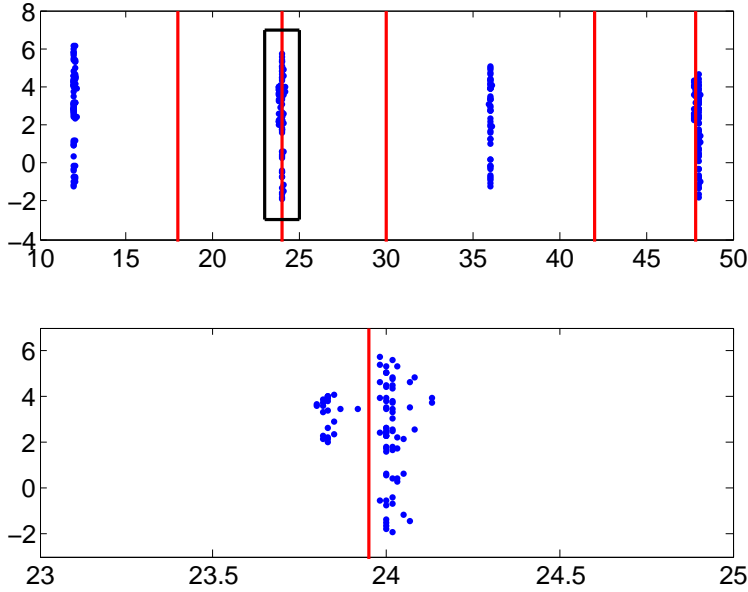


Figure 3: Shows the problem with taking the independent variable into account. Local variations in y causes our "y-aware" algorithms to split clusters in an unwanted way.

The problem has to do with the scale. On a large scale it is obvious that the splitting of the second cluster is unwanted, however on a small scale this is not obvious anymore. The mean of the dependent variable (y -direction) of the two clusters is different and thus the clusters should be separated.

6.5 Summary

To summarize this section, the methods that will be evaluated in Section 8 are the following:

Method1 : Minimize $W + \alpha\Phi$

Method2 : Minimize $W + \alpha \sum_{i=2}^K \frac{\phi(e_i)}{1 + \ln(\Delta_i y + 1)}$

7 Minimization method & estimation of the number of bins

The algorithm seeks to place out K bins in a way such that the objective function is minimized. To avoid having initial bins violating the minimal number of data points in each bin constraint, the equal size method with the same constraint is used as the initial guess. It can be computed very fast and gives a good starting point. We have not noticed any difference in result using different initial guesses, however an initial value that resembles the final result makes the algorithm converge faster.

To make the minimization algorithm more efficient, instead of minimizing the within-cluster variability W , the total variability T minus the between cluster variability B was minimized. This simplification is possible since the total variability equals $T = W + B$ [6]. The different variabilities are defined as

$$T = \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (18)$$

$$W = \sum_{k=1}^K \sum_{l=1}^{n_k} (x_{kl} - \bar{x}_k)(x_{kl} - \bar{x}_k)^T \quad (19)$$

$$B = \sum_{k=1}^K n_k (\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^T, \quad (20)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (21)$$

and x_{kl} is the x value in the l : th point in bin k and n_k is the number of points in bin k .

Algorithm 1 Minimization Algorithm

Continue optimization until the following stop criteria is fulfilled:

1. No bin boundary can favorably be moved between its two neighbors.
2. No bin boundary can favorably be moved in-between any two other bin boundaries.

Part I of the optimization

Try moving the bin edges one by one within its two neighbors to decrease the objective function.

Step 1:

Choose the bin edge that gives the biggest decrease (or smallest increase) when being moved one step to the left or right. Then calculate the objective function for every possible position between the two neighbour bin edges and move it where it is lowest.

Step 2:

Mark the moved bin edge as updated so that it can't be moved again unless any edge is moved first.

Step 3:

If a bin edge has been moved, update the neighboring bin edges so that they can be moved again.

Part II of the optimization

Try taking out the bin edges one by one and placing them in between two other bin edges to decrease the objective function

Step 1:

Calculate the increase in the objective function for removing any of the bin edges. Also calculate the decrease in the objective for adding an extra between bin edge between any two consecutive bin edges.

Step 2:

If there is any move of a boundary that results in a decreased objective function, perform the movement that results in the largest decrease and go back to PART I, else stop.

7.1 Choosing the number of bins

Determining the number of bins in the data is an important problem. It affects the resolution of the VPC and hence, also point 1 in the desired binning characteristics. In some cases the number of bins is obvious, which is the case in Figure 1. In general this is not the case and the number of bins must be estimated by the user of the VPC based on some prior knowledge of the data or estimated somehow. Since we want a fully automatic binning we need a method that does this for us.

A simple and direct strategy would be to use our objective functions (eq. 14 and eq. 17) not only to place our bins but also to estimate the number of bins. For our two methods presented in the previous section, we then get the following:

$$\arg \min_K W(K) + \Phi(K) \quad (22)$$

$$\arg \min_K W(K) + \alpha \sum_{i=2}^K \frac{\phi(e_i)}{1 + \ln(\Delta_i y + 1)} \quad (23)$$

After trials we could conclude that this way of estimating the number of bins works well when incorporating y while it has a tendency to underestimate the number of bins when only considering the independent variable.

There exists a variety of methods to estimate the number of bins. In an article written by G. Milligan & M. Cooper [5] different methods of estimating the number of clusters have been evaluated. The method that in general outperformed the others was the Calinski and Harabasz method,

$$\arg \max_K \frac{B/(K-1)}{W/(n-K)}. \quad (24)$$

where B is defined in eq. 20.

Tests of this method gave good results on most but not all data sets when using Method 1. It had a tendency to overestimate the number of bins. It performed worse when incorporating y , that is, Method 2.

Because our first approach to use the objective function had a tendency to underestimate the number of bins and the Calinski & Harabasz method had the opposite tendency the quotient was tested:

$$\arg \max_K \frac{B/(K-1)}{W/(n-K)} / (W + \alpha \Phi). \quad (25)$$

In many of the cases where the data was well separated both the objective function and the Calinski & Harabasz method gave the same result which coincides with the results of eq. (25). In the cases where they gave different results eq. (25) weighted the two methods in a way that the estimation of the number of bins gave a good result on the data sets used for method development.

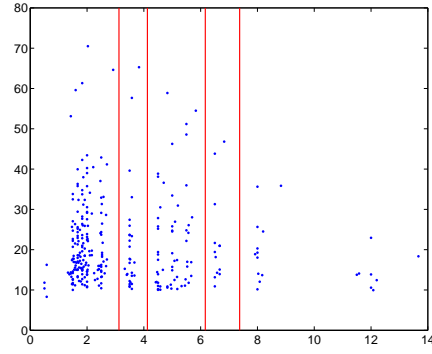
8 Result

We have implemented the proposed methods in Matlab. Ten data sets were used for validation. The parameters in our methods were chosen based on experiments on only the training data and were kept fixed during the validation. For M. Lavielles & K. Bleakleys method, Monolix was used with its standard parameters. The performance of our methods and M. Lavielles & K. Bleakleys method was visually evaluated by some of the modelers we interviewed for the desired binning characteristics. The evaluation was made on Method 1 with eq. (25) for estimating the number of bins and Method 2 with eq. (23).

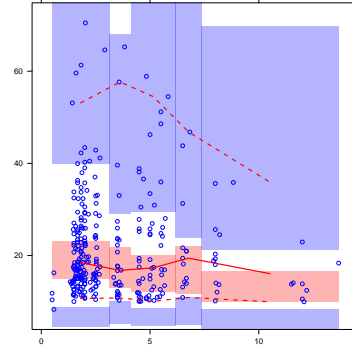
Firstly the data sets were binned by the different methods. Then the binnings were used to make the VPCs. Finally the VPCs were sent to the modelers for evaluation. In their opinion method 1 gave rise to a binning that definitely was useful. The performance of Method 1 was similar to that of M. Lavielles & K. Bleakleys while Method 2 captured too much local variation.

In Figure 4 the first of the validation data is shown. Looking at the data one can see that there is no obvious way to do the binning. This is the reason we needed the help from the modelers in order to evaluate the methods. The major reason for the methods to perform so differently were that the number of bins were chosen differently. M. Lavielle & K. Bleakleys method chooses much more bins which results in an irregular median of the VPC, Figure 4(f). The modelers argue that this is not motivated by the data since the overall changes are also captured by Method 1 and 2 with much fewer bins. In this particular case the modelers regarded Method 1 as the best followed by Method 2. M. Lavielles & K. Bleakleys method did not perform well on this data.

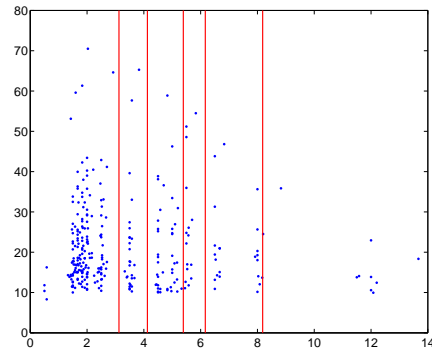
In Figure 5 we show a second data set. In this particular case M. Lavielles & K. Bleakleys method chooses fewer bins than our two methods so the results of the methods on this data set are quite opposite of the previous. In this case the modelers regarded M. Lavielles & K. Bleakleys method as better.



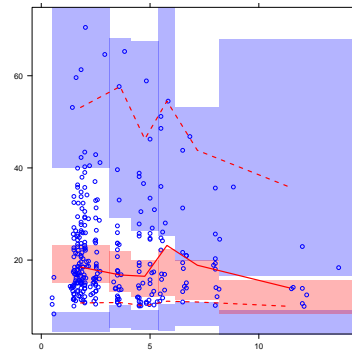
(a) Method 1 binning



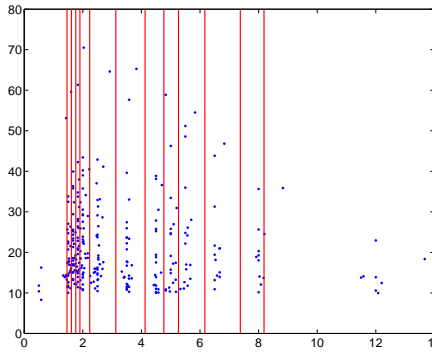
(b) VPC of method 1



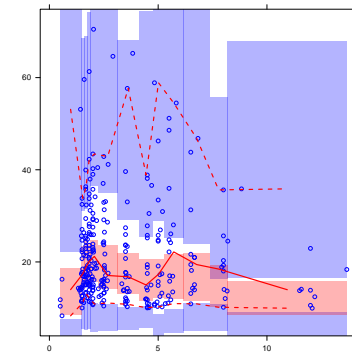
(c) Method 2 binning



(d) VPC of method 2



(e) M. Lavielle & K. Bleakley binning



(f) VPC of M. Lavielle & K. Bleakley

Figure 4: VPCs and binning created of the different methods. The data is from the article "Clinical pharmacokinetics of irinotecan and its metabolites: A population analysis"

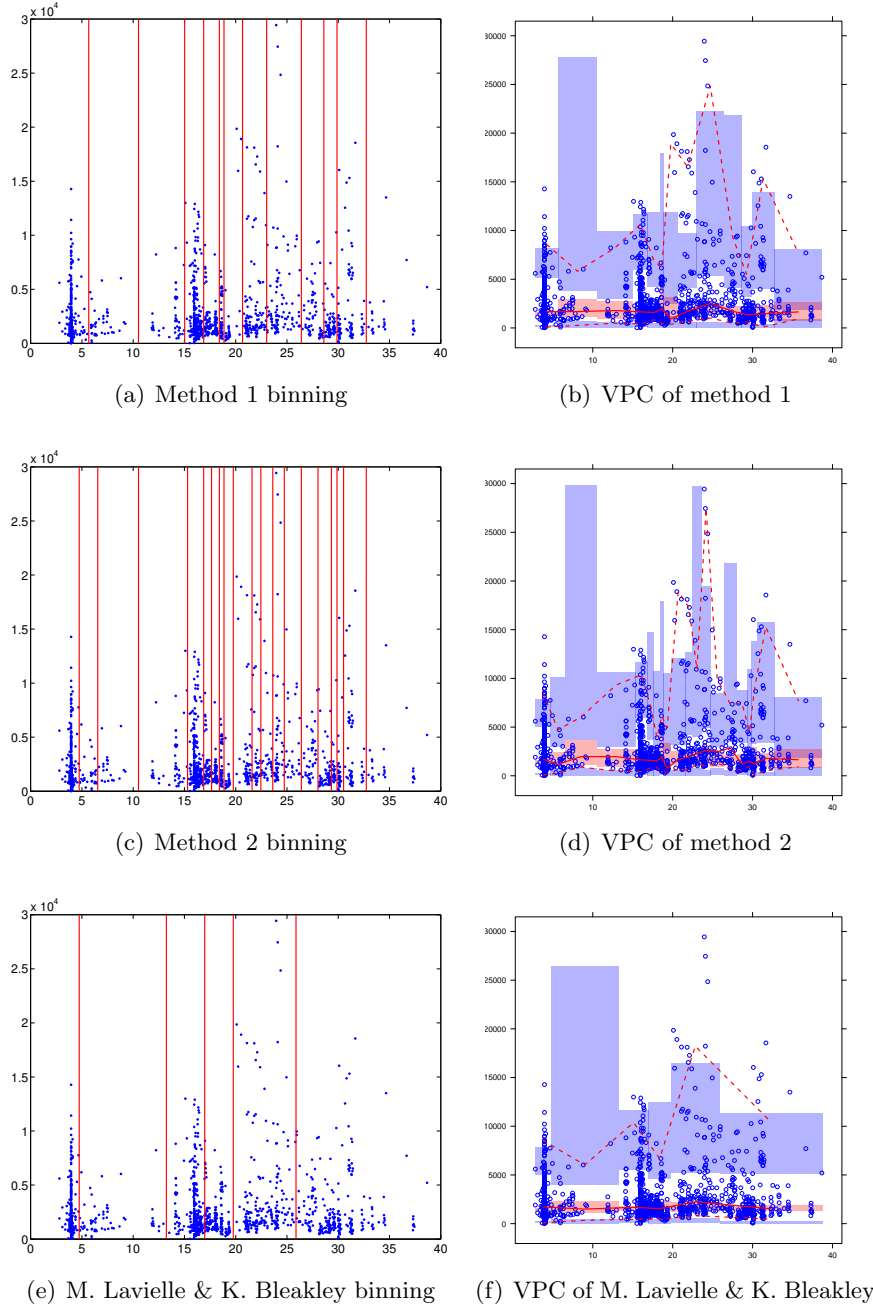


Figure 5: VPCs and binning created of the different methods. The model is an example of a 2-compartment PK model

9 Discussion

From the validation sets we were given and the evaluation given by the modelers it is hard to draw rigid conclusions. Partly because the feedback

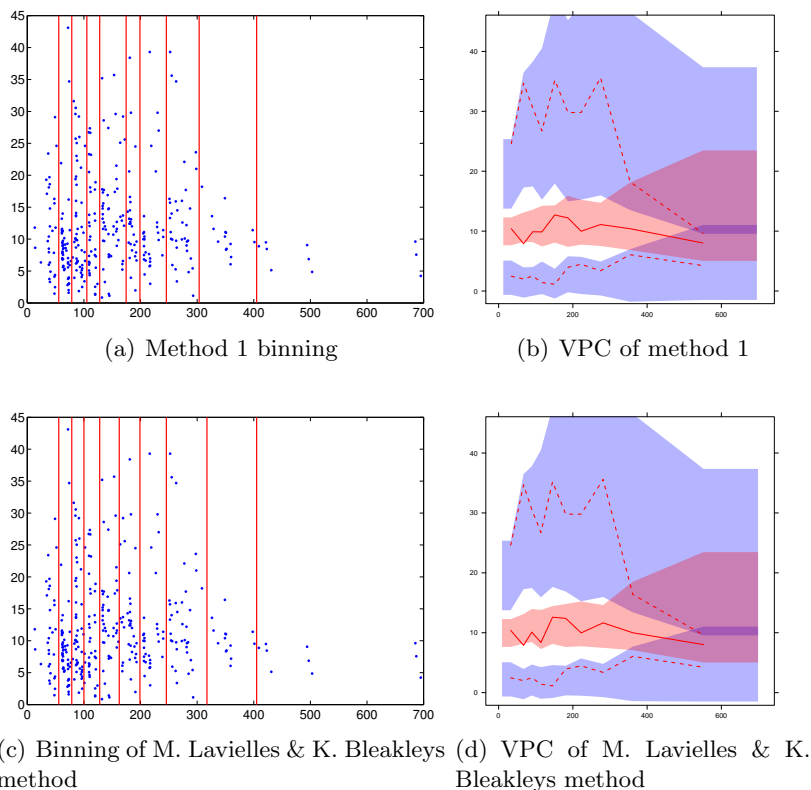


Figure 6: VPCs and binning comparison between Method 1 & M. Lavielles & K. Bleakleys method for a fixed number of bins.

consist of subjective opinions and partly because we cannot be sure that the data sets used in our validation represents the majority of data sets generated in drug development.

It seems that our Method 1 performs similar to the method of M. Lavielle & K. Bleakley. The major difference between them is how they estimate the number of bins. Method 1 performed best at some of the data sets and M. Lavielles & K. Bleakleys on the others. Method 2, that incorporated the dependent variable were not as good as the other two methods.

Also during the testing before the final validation we realized that Method 1 and M. Lavielles & K. Bleakley with a fixed number of bins gave very similar results for most of the data sets that we had at hand. In Figure 6 we show the resulting VPC of one of these data sets. As one can see, the results are very similar. The two methods do not place the bins at the exact same position but the differences are so small that it is difficult to see any differences between the two VPCs.

After the evaluation by the modelers and from our own experience it seems our method for choosing the number of bins for method 1 is more

robust than M. Lavielles & K. Bleakleys method. Only in very few cases (if any) the estimated number of bins of Method 1 is far away from what a manual estimation would produce while M. Lavielles & K. Bleakleys method for some data sets clearly overestimated the number of bins.

During the development of Method 2 we realized that it is problematic to make the objective function dependent on variations in y . What we wanted to do was taking global variations in y into account to better adjust the binning to the specific behavior of the drug. But what we often ended up doing was bringing out local variations in y . This resulted in non-smooth median and percentile curves in the VPC. The same problem occurred for all of our methods incorporating awareness for variations in y . We would therefore like to rule out objective functions on the form of (15) and (16).

Generally, when the methods performed poorly they overestimated the number of bins which led to a too high resolution in the VPC which resulted in an unphysiological median curve. So if one wishes to improve any of these methods any further, it might be a good idea to focus on the way they determine the number of bins.

After the evaluation of the methods it has become more clear that fewer bins are often more desirable. There is a discrepancy for what could be considered a natural binning and what looks good in the VPC. Our method for estimating the number of bins was developed perhaps with too little consideration of the final VPC. This is mostly due to the lengthy process of creating VPCs when developing our methods.

9.1 Minimizing M. Lavielles & K. Bleakleys objective function using our minimization algorithm

In an earlier version of our minimization algorithm, we implemented the objective function by M. Lavielles & K. Bleakley. But when we further developed our minimization algorithm we did not update this implementation. The reason for this was that we had started using the software Monolix which has a complete implementation of M. Lavielles & K. Bleakley as described in their article [2].

However, M. Lavielles & K. Bleakleys objective function can still be implemented rather easily in our minimization algorithm. Starting out from Method 1, the data density function has to be removed. Furthermore the calculated between cluster variance has to be replaced by the within cluster variance but altered according to (6). Also equation (5) must be taken into consideration when calculating the final objective function value at the end of our minimization algorithm. Finally a new function must be written to go through the desired K values to find the appropriate number of bins.

9.2 Needed interactivity for our methods

Our two methods cannot be considered as fully automatic in their current state since they do not handle all data sets in an appropriate manner. It is therefore desirable to have some kind of interactivity of the methods when they are further implemented in PsN. We believe that the number of bins is the most important parameter to be able to tweak and play with in that case. The number of minimum points in each bin could also be a good variable to let the user change. Although it may not always affect the result that much, it can be important if one wants to avoid getting bins with too large confidence intervals for the median and the percentiles. We also have the parameter α in the objective function which can be changed to alter how much the data density function effects the final result. However we do not believe that this parameter adds that much to the user interactivity and it is neither apparent how this variable affects the binning.

Another element that could be implemented for better results is the possibility to choose to perform the binning on the altered data set where the independent variable values have been transformed so that their values is equal to the natural logarithm of their earlier value. The resulting bin edge can then be transformed back to the correct position by taking e to the power of the different bin edge values. This is a useful approach when the independent variable is the time and the sampling is done at increasingly longer intervals. In Figure 7 the effect of taking the logarithm of the independent variable values is presented in a case where the data set have an advantageous distribution for doing this. As can be seen, the new binning better captures the variation in the dependent variable for small time values.

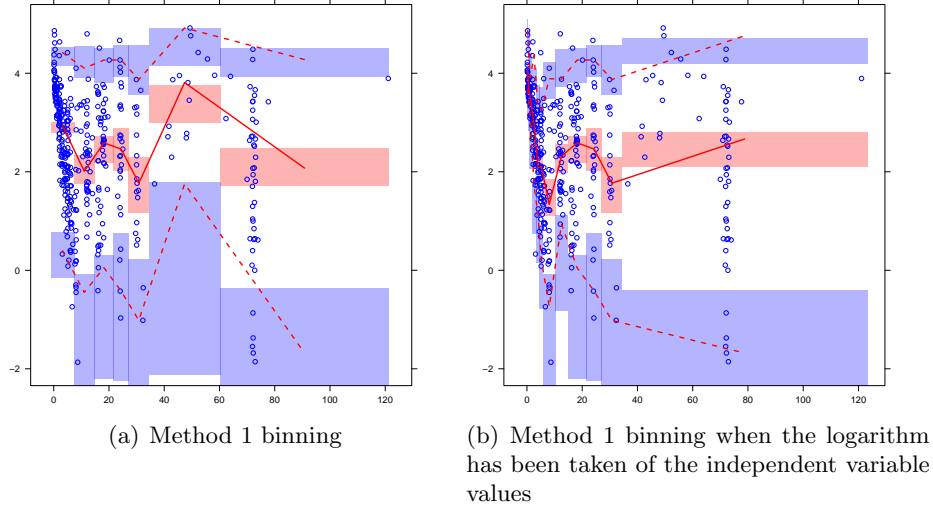


Figure 7: Effect of taking the logarithm of the independent variable values on a specific data set

9.3 Future work

Here we present some of our considerations and ideas on how to improve our methods that were not implemented due to time constraints or other difficulties.

9.3.1 Measure of the binning quality

Evaluation by the modelers can be very subjective and therefore an objective measure for what constitutes a good binning would be desirable. In the beginning of the project we developed an idea for how this could be done.

Usually when making a VPC, the VPC calls a program called NONMEM which, as input, takes the time points from the measured data and with the mathematical model simulates several data sets. Instead of feeding NONMEM with only the time points from which measures have been done we could also make our own input. By making the time steps small enough, a true distribution could be simulated (true of the mathematical model). From this distribution we then would calculate the "true" median.

We then proceed by making a simulation from the measured time points and apply the binning methods on this data. From the binning we calculate the median. This median is then compared to the "true" median of the distribution. This would then constitute a measure for how good the binning is.

We abandoned this approach early due to problems in making simulations of the true distribution. This was due to that the models often had covariates whose distribution was unknown and also due to the fact that it was very time consuming. However, we still think that this approach would constitute a good measure for the binning quality.

9.3.2 How to measure the smoothness of a curve

The second derivative tells something about the local curvature of a curve.² If we integrate the absolute value of the second derivative we get a measure for how curved the curve is.

$$\int_{x_{\min}}^{x_{\max}} |y''(x)| dx$$

A straight line will have a value of 0 while a zigzagged curve will have a very large value. A discrete approximation is:

$$\sum_{i=2}^{K-1} \left| \frac{y(x_{i+1}) - y(x_i)}{x_{i+1} - x_i} - \frac{y(x_i) - y(x_{i-1}))}{x_i - x_{i-1}} \right|$$

²Curvature is given by: $\kappa = |y''|/(1 + y'^2)^{3/2}$

with $y(x_i)$ being the median in bin i . It might be possible to analyze the value of this measure to determine a suitable amount of bins. However, if we use this directly in the optimization criteria for placing the bin boundaries it will make the median curve to look more linear and less curved than it necessarily is.

10 Summary and conclusions

Both Method 1 and M. Lavielles & K. Bleakleys method are steps in the right direction of creating automatic binning in VPCs. The results from our Method 1 were considered as "definitely useful" by the modelers and will hopefully ease their work in terms of better diagnostics and less hours spent in creating VPCs.

Incorporating variations in the dependent variable was dismissed since it did not give any meaningful results.

Acknowledgements

We would like to thank our supervisor Kajsa Harling for her dedication to our project. Also we would like to thank the modelers for their help in establishing the desired binning characteristics and the evaluating our methods.

References

- [1] M. Bergstrand, A. Hooker, J. Wallin, M. Karlsson, *Prediction-Corrected Visual Predictive Checks for Diagnosing Nonlinear Mixed-Effects Models*, January 4, 2011
- [2] M. Laville, K. Bleakley, *Automatic data binning for improved visual diagnosis of pharmacometric models*, June 9, 2011
- [3] Silverman, B.W. *Density Estimation for Statistics and Data Analysis*, 1998.
- [4] J.B. MacQueen. Some methods for classification and analysis of multi-variate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281-297. University of California Press, 1967.
- [5] G. Milligan, M. Cooper, *An examination of procedures for determining the number of clusters in a data set*, Psychometrika-vol. 50, no 2, 159-179. June 1985

- [6] M. Yan, *Methods of Determining the Number of Clusters in a Data Set and a New Clustering Criterion*, November 2005

Appendix

A K-means algorithm

Algorithm 2 K-means Algorithm

Given an initial set of k means $m = (m_1^1, m_2^1, \dots, m_k^1)$ the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation to the cluster with the closest mean. $S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\| \leq \|x_p - m_j^{(t)}\| \forall 1 \leq j \leq k\}$

Update step: Calculate the new means to be the centroid of the observations in the cluster. $m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$

Repeat: Until no changes can be made in the assignment step.

B Matlab code

The following matlab code files have been supplied with this report.

- * readFromFile.m - function that reads in data generated from PsN into matlab
- * arrayToText.m - function that transforms an array of number into a string
- * firstMethod.m - implementation of our first method
- * secondMethod.m - implementation of our second method
- * chooseKFirst.m - function that chooses the number of bin for our first method
- * chooseKSecond.m - function that chooses the number of bin for our second method
- * gaussFilter.m - implementation of gauss filter used for smoothing
- * indexToIdv.m - index handling function used by firstMethod.m and secondMethod.m

- * plotBins.m - function that plots the edges of the bins together with the data
- * saveCommand.m - function that generates the new command line for PsN
- * HowToRun.m - example file of how the matlab files should be used

An example of a data set has also been supplied together with files. How this data can be binned with these functions is exemplified in HowToRun.m. For more information, see the comments in the code.

C Abandoned methods

Many methods where the dependent variable y was incorporated were tried out. However, most of them were abandoned since the results did not look promising. Some of the methods below are clustering techniques meant for 2D-clustering and they are not appropriate for 1D-binning. B, W, and T are defined in eq. 20, eq. 19 and 18 respectively for all the following methods.

Method 3

(Considered first.) It was motivated from the article of M. Lavielle & K. Bleakley. The objective function for this method is:

$$\text{Minimize } \frac{W}{B}.$$

Method 4

The covariance was also tried:

$$\begin{aligned} \text{Minimize } & - \sum_{k=1}^K B_{12,k} B_{11,k} = \\ & = \text{Minimize } - \sum_{k_1}^K N_k |T_{12} - (x_i - \bar{x}_k)(y_i - \bar{y}_k)| (T_{11} - (x_i - \bar{x}_k)^2) \end{aligned}$$

Method 5

$$\text{Minimize } tr(BW^{-1})$$

Method 6

$$\text{Minimize } tr(W) = W_{11} + W_{22}$$

which is calculated by minimizing

$$-B_{11} - B_{22} = \sum_{k=1}^K -N_k((x_i - \bar{x}_k)^2 + (y_i - \bar{y}_k)^2).$$

Method 7

As Method 5 but applies scaling

$$\sum_{k=1}^K -N_k \left(\frac{(x_i - \bar{x}_k)^2}{\omega_x} + \frac{(y_i - \bar{y}_k)^2}{\omega_y} \right)$$

Method 8

This method is a further development of the equal size method. The equal size method is equivalent to integrating the number of data points and putting out a bin edge at every place where the integral passes a value which is a multiple of the total number of data points divided by the number of desired bins.

The idea was then to multiply the number of data points at each unique independent value with some measure of how much the dependent value was changing in that point. We used different ways of defining this measure.

- * Take the M closest points alternatively the points that lie within distance D of every unique idv value. Then fit a line to those data point and use the slope of the line as a measure of the local change in the dv value. Finally, adjust the number of data points at the unique idv value with that measure.
- * Calculate the mean dv value for every unique idv value. Then we translated this value to a grid with higher resolution than the current one consisting only of the unique idv values. The mean dv values were now smoothed out with a gaussian window of appropriate size (the size was problematic to chose). After this has been done a derivative of each unique idv value could be calculated by taking taking the dv difference between the point directly to the left and the right of each point corresponding to a unique idv value in the new grid. This gave us a measure that was used to adjust the integrand (the number of data points at each idv value).

Method 9

This method started out by creating a bin for every unique idv value. The two closest bins in the idv direction were then merged together until only M bins were remaining, where $M > K$. $M = 2K$ were mainly used. The two closest were now merged according to a different distance measure until the remaining number of bins were K . The different distances between two neighbouring bins used were

- * Distance defined as the shortest distance from any point in the first bin to any point in the other bin. Motivation: To see if this gave any other result than the normal distance measure.
- * Distance in idv direction between the two bin centers multiplied by the current minimum number of data point in the two bins. Motivation: Try getting bins with similar number of data points.
- * Euclidean distance between the bin centers. Motivation: incorporate variation in the dependent variable.
- * Euclidean distance between the bin centers where the dv component was diminished. Motivation: diminish the effect of the variations in the dependent variable.