Practical Aspects of Method Comparison Studies

Version 2.8
7 August 2007

Bendix Carstensen — Steno Diabetes Center, Gentofte, Denmark
bxc@steno.dk
http://www.biostat.ku.dk/~bxc

Lyle Gurrin — School of Population Health, University of Melbourne
lgurrin@unimelb.edu.au
http://www.epi.unimelb.edu.au/about/staff/gurrin-lyle.html

Contents

1 Introduction

Method comparison studies atempt to determine the extent of the agreement between several methods measuring the same quantity. The ultimate goal is to establish prediction equations that convert from one method to another with confidence limits that reflect the accuracy of the conversion and account for all relevant sources of variation.

Comparing two methods of measurement is often conducted using the so-called "Bland-Altman" procedure which first appeared in 1983 [1]. This procedure, where one plots the difference against the mean for each pair of observations and compute limits of agreement as prediction limits for future differences between pairs of measurements, has become the *de facto* standard for analysis of method comparison studies without replicates [2]. In a more recent publication [3] Bland & Altman recommended performing replicate measurements and provided theory and examples for a couple of specific scenarios.

Carstensen [4] outlined a general model for comparing two or more methods of measurement, linking the mean measurements across each of several methods by linear functions and allowing arbitrary replication structure. The exchangeability structure of replicates is captured by two-way interactions between method, item (*i.e.* person, sample) and replicate. This approach generates formulae for translating measurements from one method to another.

In this paper we focus on prediction limits for these translation formulae, in the case where we have an observed measurement from one method and wish to predict the corresponding measurements by the other methods. A natural requirement for such translation formulae and prediction limits is that they behave symmetrically in the methods compared, such that prediction from method 1 to method 2 and from that prediction back to method 1 returns the initial observed value on method 1.

The algorithm proposed for fitting the models and obtaining maximum likelihood estimates of the parameters in [4] is absurdly complicated, especially since the model is but a nonlinear mixed effects model.

In this paper we use the `BUGS` software package[5] [1] for estimation. `BUGS` allows an implementaion of the model which is symmetric in the methods, even if it is overparametrized. Moreover `BUGS` has a facility for incorporating "blank" data arrays for generating predictions and prediction limits that simultaneouly incorporate parameter uncertainty and prediction variance, which we use to justify our approach to cmputation of predictin limits.

We give some illustrative examples and describe the **R**-package "MethComp" that generates and runs the `BUGS` code using either the `R2WinBUGS` interface between **R** and `WinBUGS` (or, eventually, the `BRugs` package in **R**).

2 Notation

We introduce the following naming conventions for the text and **R** computer program in the Appendix in order to make it compatible with the notation in [4]. We have data on I

[1] In this paper we take `BUGS` as a generic term for implementation of this language and associated MCMC methods like `Classic BUGS`, `WinBUGS`, `R2WinBUGS`, `BRugs` and `JAGS`

items (in clinical studies this will be the patient) measured using M different methods. There are R_{mi} replicates of the measurement on item i by method m. The measurements are denoted y_{mir}, where the subscripts in the set $\{m, i, r\}$ denoting method, item and replicate respectively are always used in the order. In the case of measurements made without replicates we drop the subscript r but the other two subscripts will always be present. We make no assumptions about the completeness or regularity of the observational scheme, but assume that the methods are exchangeable by specifying models with the same structure for each method. In the specification of the variance-component models we follow the terminology proposed by Littel *et al.* [6] where fixed effects are denoted by Greek letters and random effects by Latin.

3 Limits of agreement and prediction.

The standard setup for comparison of two measurement methods is one where a single measurement by each method is taken on each item. In that case the recommendation is to compute the *limits of agreement*, that is, a prediction interval for the difference between future measurements taken by each of the two methods on a single, new item.

Underlying this approach is a two-way analysis of variance model:

$$y_{mi} = \alpha_m + \mu_i + e_{mi}, \qquad e_{mi} \sim \mathcal{N}(0, \sigma_m^2), \qquad \text{with } \alpha_2 = 0 \text{ for convenience} \tag{1}$$

The differences $y_{1i} - y_{2i}$ have variance $\sigma_1^2 + \sigma_2^2$, and so the 95% prediction interval for a new difference is:

$$\alpha_1 - \alpha_2 \pm 1.96 \times \sqrt{\sigma_1^2 + \sigma_2^2} = \alpha_1 \pm 1.96 \times \sqrt{\sigma_1^2 + \sigma_2^2}$$

In practice the last term is computed as the empirical standard deviation of the differences. There are two assumptions underlying this model:

1. The mean difference between the methods is constant over the range of measurements.

2. The variation of the differences is constant over the range of measurements.

These assumptions are normally checked by making a so-called Bland-Altman plot [2], where differences are plotted against averages of methods for each pair of observations.

The question of predicting a future measurement by method 2, y_{20}, given an observation by method 1, y_{10}, can be treated in this framework by formulating model (1) as:

$$y_{10} \sim \mathcal{N}(\mu_0, \sigma_1^2), \qquad y_{20} \sim \mathcal{N}(\mu_0 + \alpha, \sigma_2^2) \tag{2}$$

with y_{10} and y_{20} independent. A 95% prediction interval for $y_{20} - y_{10}$ is

$$\alpha \pm 1.96\sqrt{\sigma_1^2 + \sigma_2^2}$$

which is estimated by the limits of agreement:

$$\bar{d} \pm 1.96 \times \text{s.d.}(d_i), \qquad \text{where } d_i = y_{2i} - y_{1i}.$$

The distribution of the unobserved measurement y_{20} is governed by two unknown parameters, μ_0 and α. Since we are dealing with a new item, most likely *not* drawn from the same population as the data, we only have *one* datapoint available for estimating μ_0, namely y_{10}. The maximum likelihood estimate under model (2) is then $\hat{\mu}_0 = y_{10}$. The calibration sample provides data for estimation of α, namely $\hat{\alpha} = \bar{d}$. We benefit from using all of the data in the estimation of α because we have assumed that the relationship between methods (a constant difference of α) is the same regardless of the value of μ. Therefore, the estimated mean for the unknown new observation y_{20} is $y_{10} + \bar{d}$. The variance of this estimator is $\sigma_1^2 + (\sigma_1^2 + \sigma_2^2)/N$, where N is the number of paired observations in the calibration sample. Since the variance of (the future observation) y_{20} is σ_2^2, the prediction interval for y_{20} will be:

$$y_{10} + \bar{d} \pm 1.96\sqrt{\sigma_1^2 + (\sigma_1^2 + \sigma_2^2)/N + \sigma_2^2} = y_{10} + \bar{d} \pm 1.96\sqrt{\frac{N+1}{N}(\sigma_1^2 + \sigma_2^2)}$$

This is another way of expressing the limits of agreement as a prediction interval for $y_{20} - y_{10}$: The estimate of the distribution of y_{20} given y_{10} is obtained by offsetting the distribution of $y_{20} - y_{10}$ by y_{10}. This is because the μ_is are not assumed to be drawn from a population distribution but specified instead by either circumstance or design and that the assumption of exchangeability with μ_0 is not reasonable. The prediction procedure therefore becomes symmetric in y_1 and y_2 in the sense that if we first predict y_{20} from y_{10} and then predict y_1 based on the predicted value of y_{20}, we will end up with the value of y_{10} with which we started.

4 Linear relationships between methods

In this section we generalize the framework for method comparison to models that accommodate more than two methods of measurement, replicates with different exchangeability structures and linear relationships between methods, instead of adopting the usual assumption of a constant difference between methods.

4.1 Introduction

The most general model for establishing linear prediction equations between methods which allows (random) interactions is:

$$y_{mir} = \alpha_m + \beta_m(\mu_i + a_{ir} + c_{mi}) + d_{mr} + e_{mir} \tag{3}$$

for $m = 1, \ldots, M$, $i = 1, \ldots, I$ and $r = 1, \ldots, R_{mi}$. In most setups of relevance, I will be much larger than M and R; the latter two will typically be smaller than 5. The parameter μ_i is the underlying (but unobserved) true measurement value for item i, and α_m and β_m are the parameters that defines the linear relationship between μ_i and the mean measurement of item i made by method m. The remaining parameters are assumed to be normally distributed, zero mean random effects:

- e_{mir} is the residual error, with method-specific standard deviation σ_m.

- d_{mr} is the method by replicate interaction with method-specific standard deviation ω_m.

- c_{mi} is the method by item interaction or "matrix effect" with method-specific standard deviation τ_m.

- a_{ir} is the item by replicate interaction, with standard deviation ν.

- If only two methods are involved, it is not possible to estimate separate variances for each method for the method×item or the method×replicate random effects.

  This is the reason that the variance component c_{mi} is multiplied with β_m. For $M = 2$ where the variance cannot not depend on m, the model would not be invariant under under rescaling of one of the methods. For $M \geq 3$ the two formulations lead to the same model.

4.2 Scaling

The model (3) is qualitatively different from the two-way analysis of variance model (1), because the methods are *scaled* relative to each other. In (1) we require that all methods measure on the same scale, otherwise differences between methods would be meaningless.

The model (3) will be invariant under scaling of measurements from one method; if e.g. the y_1s all were mutiplide by e.g. 2, then we would just get β_1, ω_1 and σ_1 twice as big, and corresponding change in α_1, but the relationship between methods would otherwise be the same.

4.3 Interpretation of variance components

All the two-way interactions, method×item, method×replicate and item×replicate are left unspecified in model (3). However, it should be considered how much of the interactions should be systematic (i.e. depending on other fixed, measured quantities like time/day):

method×replicate: In contrast to the other two-way interactions this one will have a limited number of levels, and as such be more likely to be modelled by a systematic effect instead. For example, if replicates are made on different days, a linear effect of day of measurement could be included in the model accounting for differential decay of measurements with time between methods.

method×item: In clinical chemistry this is often called a "matrix" effect, referring to the method-specific solution (matrix) used in the laboratory, which may interact differently with the items. The effect measures the random interactions with items that may influence the outcome. There is usually not much point in putting any further structure on this interaction, as the systematic part of the method × item interaction form the core of the model — the linear relationship between methods across the measurement scale represented by the items.

item×replicate: This is a random effect between replicates within item, but common for all methods, so it is not possible to have a method-specific variance for this. Furthermore, since the general linear structure of the relationship between methods formally allows methods to measure on different scales, this random effect must necessarily be on the "dimensionless" scale of the μs. Unless there is a special structure to the items, it is difficult to see what possible systematic effects could be brought to bear on this term.

This effect must be included in the model if replicates are linked across methods, e.g. if replicates are taken by all methods in parallel. An example of this would be measurement of a clinical feature by different methods, where replicates on the same persons are done on different days with a few days apart, but with all methods used each day. The item by replicate measurement would then model the random day-to-day variation of individuals, and induce a correlation between measurement by different methods taken on the same day.

Since the μs, the αs and the βs are only determined up to a linear/scale transformation (see section 5.1), the size of variance component ν is not meaningful in itself. It is only interpretable when scaled to a particular method as $\beta_m\nu$. Given the assumption of of linear (i.e. *scaled*) relationsgip between it is clear that the between replicates variation is specific to the measurement scale, and hence must be reported on a specific (or all) scale.

5 Estimation

5.1 The parametrization

The linear relationship linking methods in model (3) gives the following translation formula from method 1 to method 2 is (for the mean):

$$y_2 = \alpha_2 + \beta_2\mu = \alpha_2 + \beta_2(y_1 - \alpha_1)/\beta_1 = \big(\alpha_2 - \alpha_1\beta_2/\beta_1\big) + (\beta_2/\beta_1)y_1$$

i.e. the intercept and slope used for conversion from metod 1 to 2 are:

$$\begin{aligned} y_2 &= \alpha_{2\cdot1} + \beta_{2\cdot1}y_1 \\ \text{i.e.:} \qquad \alpha_{2\cdot1} &= \alpha_2 - \alpha_1\beta_2/\beta_1 \\ \beta_{2\cdot1} &= \beta_2/\beta_1 \end{aligned}$$

Model (3) is overparametrized — a linear transformation of the μs will just result in a linear transformtion of the αs and βs; a transformation $\mu_i \mapsto \xi_i = \gamma + \delta\mu_i$ will give the model for the means:

$$y_{mi} = \alpha_m + \beta_m\mu_i = \alpha_m + \beta_m(\xi_i - \gamma)/\delta = (\alpha_m - \beta_m\gamma/\delta) + (\beta_m/\delta)\xi_i$$

However $\alpha_{2\cdot1}$ and $\beta_{2\cdot1}$ are invariant under replacement of α_m by $\alpha_m - \beta_m\gamma/\delta$ and β_m by β_m/δ, $m = 1, 2$, this is left to the reader as a small algrebraic exercise.

5.2 Managing the overparametrization

In our implementation in BUGS we maintain the symmetry between methods by retaining the over-parametrised model (3), but use a prior (and hence posterior) for the μ_is with finite support, in practice a uniform on a suitable interval (details are in the Appendix). In practise this will keep the μ_is from wandering off to plus or minus infinity and keep chains in a bounded area of the parameter space. However, there is no guarantee that the μs, αs and βs will converge. What we can hope (and in practise see) is that the relevant linear functions of the αs and βs converge nicely, i.e. the parameters $\alpha_{k\cdot j}$ and $\beta_{k\cdot j}$ that represent the intercept and slope in the linear translation formulae from method j to method k. Thus we set up the model as specified in (3), but report the posterior of the parameters $\alpha_{k\cdot j} = \alpha_k - \alpha_j\beta_k/\beta_j$ and $\beta_{k\cdot j} = \beta_k/\beta_j$ and the variance components (i.e. the estimated standard deviations).

The posterior distributions of the μs are normally not of any interest.

5.3 Prediction limits - analytical solution

Prediction limits for the translation formulae based on the BUGS-implementation of model (3) can be constructed in two different ways.

Firstly, we pursue an analytically based approach. The translation formulae will be based on estimates of $\alpha_{2\cdot1}$ and $\beta_{2\cdot1}$ (e.g. posterior medians), and prediction limits on estimates of the variance components. Paralleling the computations in [4] we have for a potential observation on a new person, 0, by method 2, y_{20} (omitting replicate number as we are interested in prediction of a single measurement from a single measurement):

$$y_{20} = \alpha_2 + \beta_2(\mu_0 + a_0) + c_{20} + d_2 + e_{20}$$

If we want to predict this from a measurement by method 1, y_{10}, we use:

$$y_{10} = \alpha_1 + \beta_1(\mu_0 + a_0) + c_{10} + d_1 + e_{10} \quad \Rightarrow \quad \mu_0 + a_0 = \frac{y_{10} - \alpha_1 - c_{10} - d_1 - e_{10}}{\beta_1}$$

and hence the prediction:

$$\begin{aligned} y_{20} &= \alpha_2 + \beta_2(\mu_0 + a_0) + c_{10} + d_1 + e_{20} \\ &= \alpha_2 + \beta_2\frac{y_{10} - \alpha_1 - c_{10} - d_1 - e_{10}}{\beta_1} + c_{20} + d_2 + e_{20} \\ &= \alpha_{2.1} + \beta_{2.1}y_{01} - \beta_1/\beta_2(c_{10} + d_1 + e_{10}) + c_{20} + d_2 + e_{20} \end{aligned}$$

Note that the random individual×replicate term $a_{ir} = a_0$ vanishes from these calculations. This is because the new (single) observation is assumed to come from the same item and same replicate and thus the (item,replicate) specific terms cancel out by subtraction. Finally, by taking the variance of the right hand side of the equation and ignoring the estimation error in the αs and βs we have

$$\text{s.d.}(y_{2\cdot1}) = \sqrt{(\beta_2/\beta_1)^2(\tau_1^2 + \omega_1^2 + \sigma_1^2) + (\tau_2^2 + \omega_2^2 + \sigma_2^2)}$$

A conversion equation can be constructed by taking the posterior median of $\alpha_{2\cdot1} + \beta_{2\cdot1}y_1$ for a suitable grid of y_1s. Similarly, prediction limits can be constructed by taking the posterior median of $\alpha_{2\cdot1} + \beta_{2\cdot1}y_1 \pm 1.96 \times \text{s.d.}(y_{2\cdot1})$ for the chosen grid of y_ss. Note that a graph with the line of conversion and these limits will be applicable both ways.

In practical situations the effect of method×replicate will rarely be present, in which case the standard deviation of the prediction reduces to:

$$\text{s.d.}(y_{2\cdot1}) = \sqrt{(\beta_2/\beta_1)^2(\tau_1^2 + \sigma_1^2) + (\tau_2^2 + \sigma_2^2)}$$

5.3.1 Posterior medians for the intercept and slope

When reporting the results we would like to have *one* conversion method, i.e. we would like the following to hold:

$$\alpha_{1\cdot2} = -\alpha_{2\cdot1}/\beta_{2\cdot1} \quad \text{and} \quad \beta_{1\cdot2} = 1/\beta_{2\cdot1}$$

The latter is automatically fulfilled for the posterior medians of the slopes because:

$$\frac{1}{\text{median}(\beta_{2\cdot1})} = \text{median}\left(\frac{1}{\beta_{2\cdot1}}\right) = \text{median}(\beta_{1\cdot2})$$

— a simple consequence of the fact that the inverse is a monotone function. For other quantiles of the posterior we have similar results (allowing for the fact the the inverse is a decreasing function.

But this nice property does not hold for the intercept parameters, because both αs and βs are involved; in general:

$$\text{median}(\alpha_{1\cdot2}) \neq \text{median}(-\alpha_{2\cdot1}/\beta_{2\cdot1})$$

However, we can get a sensible compromise by using:

$$\tilde{\alpha}_{1\cdot2} = \big(\text{median}(\alpha_{1\cdot2}) + \text{median}(-\alpha_{2\cdot1})/\text{median}(\beta_{2\cdot1})\big)/2$$

Multiplying this by $\text{median}(\beta_{2\cdot1}) = 1/\text{median}(\beta_{1\cdot2})$, we get:

$$\big(\text{median}(\alpha_{1\cdot2})/\text{median}(\beta_{1\cdot2}) + \text{median}(-\alpha_{2\cdot1})\big)/2 = \tilde{\alpha}_{2\cdot1}$$

that is the quantity computed this way gives the same intercepts regardless of whether we compute it as $\tilde{\alpha}_{1\cdot2}$ and convert to $\tilde{\alpha}_{2\cdot1}$ or vice versa.

5.4 Prediction limits - simulation solution

Alternatively, we may incorporate prediction in `BUGS` estimation routine by setting up a series of dummy item numbers and a grid of values on one particular method, with the corresponding measurements on all other methods declared to be missing. This structure is replicated for each method of measurement using a different set of dummy item numbers and grid values.

During the sampling process, `BUGS` generates values for the missing nodes representing the posterior distribution of predicted measurements on the remaining methods given the

hypothetical observed value on the index method (i.e. the method for which the grid values were supplied).

In order to prevent these hypothetical data from influence the estimation of model parameters, we use the `cut` function provided in `BUGS`. This allows use of the current sampled values of parameters to generate realisations from predictive nodes, but does not allow data contributing to these nodes (such as the grid of values from each method that are being used to generate predicted measurements on other methods) to contribute information back to the simulation and thus estimation process.

6 Example

6.1 Oximetry measurements in infants

Patients who are critically ill are unable to send enough oxygen into the bloodstream, and so the level of oxygen saturation is monitored as an indicator of the severity of the patient's condition. The traditional method of measurement uses a sample of blood on which a chemical analysis is performed to determine the level of various gases in the blood ("co-oximetry"). A newer much more convenient method uses a device called a pulse oximeter which relies on a small sensor placed on a finger or toe to determine oxygen saturation by measuring the reflectance of light through the blood vessels ("pulse oximetry").

The data in this example come from a study performed at the Royal Children's Hospital in Melbourne to examine the agreement between pulse oximetry and co-oximetry in infants born preterm, many of whom were especially sick and therefore had oxygen saturation levels lower than those usually available to test the accuracy of pulse oximetry.

Sixty one infants contributed each 3 samples to the study taken pairwise by the two methods, so the measurements are linked across methods. In some cases the values for both the `co.ox` and `pulse.ox` measurements were not available on the second and third samples. Four infants had complete data on only two occasions and one baby has only complete data on one, giving a total of 354 measurments. The data are presented as a graph in figure 1.

6.2 Results

We fitted four different models for this dataset implied by equation (3) where d_{mr} is always zero and either c_{mi} or a_{ir} or both are zero, giving four possible models:

$$\begin{aligned}
y_{mir} &= \alpha_m + \beta_m \ \mu_i + e_{mir} \\
y_{mir} &= \alpha_m + \beta_m \ (\mu_i + a_{ir}) + e_{mir} \\
y_{mir} &= \alpha_m + \beta_m \ \mu_i + c_{mi} + e_{mir} \\
y_{mir} &= \alpha_m + \beta_m \ (\mu_i + a_{ir}) + c_{mi} + e_{mir}
\end{aligned}$$

The linking of the replicates across methods requires that the item by replicate interaction is included in the model, but we chose to fit the two models without this for illustrative purposes.

The presence of replicate measurements allows estimation of method-specific residual variances σ^2_m but since there are only two methods we are forced to assume a common

variance τ^2 for the matrix effects c_{1i} and c_{2i}. Parameter definitions and descriptions are listed in table 1 and posterior summary statistics for these same quantities of interest are shown in table 2 and the distribution for each of these quantities are shown in figure **??**.

Table 1: *Parameter definitions and descriptions form model (3). Note that all variance components are reported on all scales even if there may be only one.*

$\alpha_{2\cdot1}$	=	$\alpha_2 - \alpha_1\beta_2/\beta_1$	Intercept for predicting y_2 from y_1
$\beta_{2\cdot1}$	=	β_2/β_1	Slope for predicting y_2 from y_1
$\iota_{2\cdot1}$	=	$\alpha_{2\cdot1}/(1-\beta_{2\cdot1})$	
	=	$(\alpha_2\beta_1 - \alpha_1\beta_2)/(\beta_1 - \beta_2)$	Prediction intersect with the identity line
$\sigma_{2\cdot1}$	=	$\sqrt{(\beta_2/\beta_1)^2(\tau^2+\sigma_1^2)+(\tau^2+\sigma_2^2)}$	Standard deviation of predicting y_2 from y_1
$\sigma_{m\text{tot}}$	=	$\sqrt{\beta_m^2\nu^2+\beta_m^2\tau_m^2+\sigma_m^2)}$	Total standard deviation of y_m
$\beta_m\nu$			Replicate by item variation on the y_m scale
$\beta_m\tau$			Method by item variation on the y_m scale
σ_m			Method-specific residual variation

Table 2: *Summary statistics, from four different models. We ran 3 chains using 2,500 burnin iterations, and then sampled every third value for the next 2,500 iterations, giving a posterior sample of 2,500. Numbers are posterior medians and 2.5 and 97.5 percentiles.*

	Model 1	Model 2(i,r)	Model 3(m,i)	Model 4$(i,r)(m,i)$
$\alpha_{2\cdot1}$	6.44 (−0.66;13.11)	9.18 (3.10;14.88)	5.36 (−3.87;13.84)	3.94 (−9.46; 19.34)
$\beta_{2\cdot1}$	0.88 (0.79; 0.98)	0.85 (0.77; 0.93)	0.90 (0.78; 1.02)	0.91 (0.72; 1.08)
$\sigma_{2\cdot1}$	6.81 (6.23; 7.52)	5.86 (5.26; 6.52)	7.18 (6.43; 8.42)	7.61 (5.73; 17.01)
$\iota_{2\cdot1}$	54.9 (−14.5; 66.3)	59.6 (40.2; 66.7)	52.9 (−67.7; 160)	50.9 (−181; 261)
$\sigma_{1\text{tot}}$	4.21 (3.69; 4.87)	4.19 (3.67; 4.80)	4.79 (4.04; 5.83)	6.53 (4.95; 14.02)
$\sigma_{2\text{tot}}$	5.69 (5.07; 6.47)	6.41 (5.62; 7.18)	5.76 (5.15; 6.60)	7.25 (5.85; 12.81)
$\beta_2\nu$		3.10 (2.27; 3.82)		3.54 (2.75; 4.35)
$\beta_2\tau$			2.13 (0.79; 3.68)	4.31 (2.50; 11.59)
σ_1	4.21 (3.69; 4.87)	1.87 (0.62; 3.31)	4.12 (3.63; 4.70)	1.63 (0.21; 2.83)
σ_2	5.69 (5.07; 6.47)	5.60 (4.90; 6.32)	5.30 (4.70; 6.05)	4.26 (3.54; 4.97)

In this example there seems to be edge effects in the predictions from the model (fig. 2). In the central part of data there is a very good agreement with the prediction and the limits based on the posterior median of the analytically derived quantities in section 5.3.

The model chosen for the variance structure does have some influence on the conversion formulae or the prediction intervals as seen from figure 4. In particular, the two models where the item by replicate effect is included exhibits narrower prediction limits as would be expected since this variance component does not enter in the prediction formulae. This is because some of the variation in the data is allocated to ν^2 hence the sum of the variance components determining the width of the prediction interval is smaller.

7 Discussion

Extending the comparison of methods of measurement to accommodate replicates, item-specific effects that vary across methods and non-constant bias comes at the expense of losing the computational ease of the "Bland-Altman" procedure. Beyond fitting the extended model is the need to generate prediction limits over the range of the observed data.

We have shown how to fit the general models outlined in [4] and generate prediction limits using `BUGS`. We found, however, that the approach using a grid of value for observed measurements on one method and using the posterior for the values of another method performed no better than using the medians of the posterior distributions for the relevant parameters and generating the prediction limits analytically. Hence we have chosen to implement only the analytical procedure in the `MethComp` package.

The prediction limits generated for the example in section 6 provided empirical support for the heuristic argument in section 3 that the limits of prediction should be symmetric in the models and will be the same regardless of whether one is predicting measurements on Method 2 from observed measurements on Method 1 or *vice versa*. We have ensured the

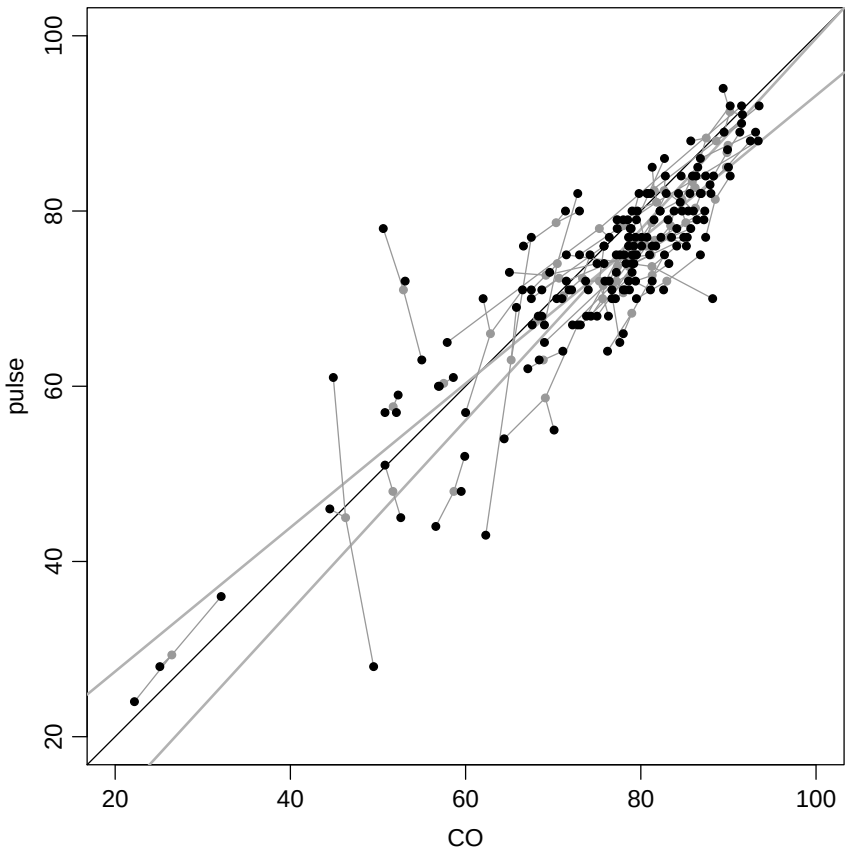

Figure 1: *Scatterplot of the oximetry data with observations from the same infant joined by line segments. The gray points are the means of the three replicates, the gray lines are the two regression lines using the replicate measurments.*

symmetry of the prediction limits in the `BUGS` model specification by eschewing the use of one method of measurement as a reference. We overcame the resulting non-identifiability of the model by using bounded support for the prior distribution of the parameters.

Care must be taken to specify an appropriate model for complex data structures involving replicates on two or more methods. The benefit of producing a correct model is a proper characterisation of the sources of variation and tighter prediction limits if the replicates are linked across methods.

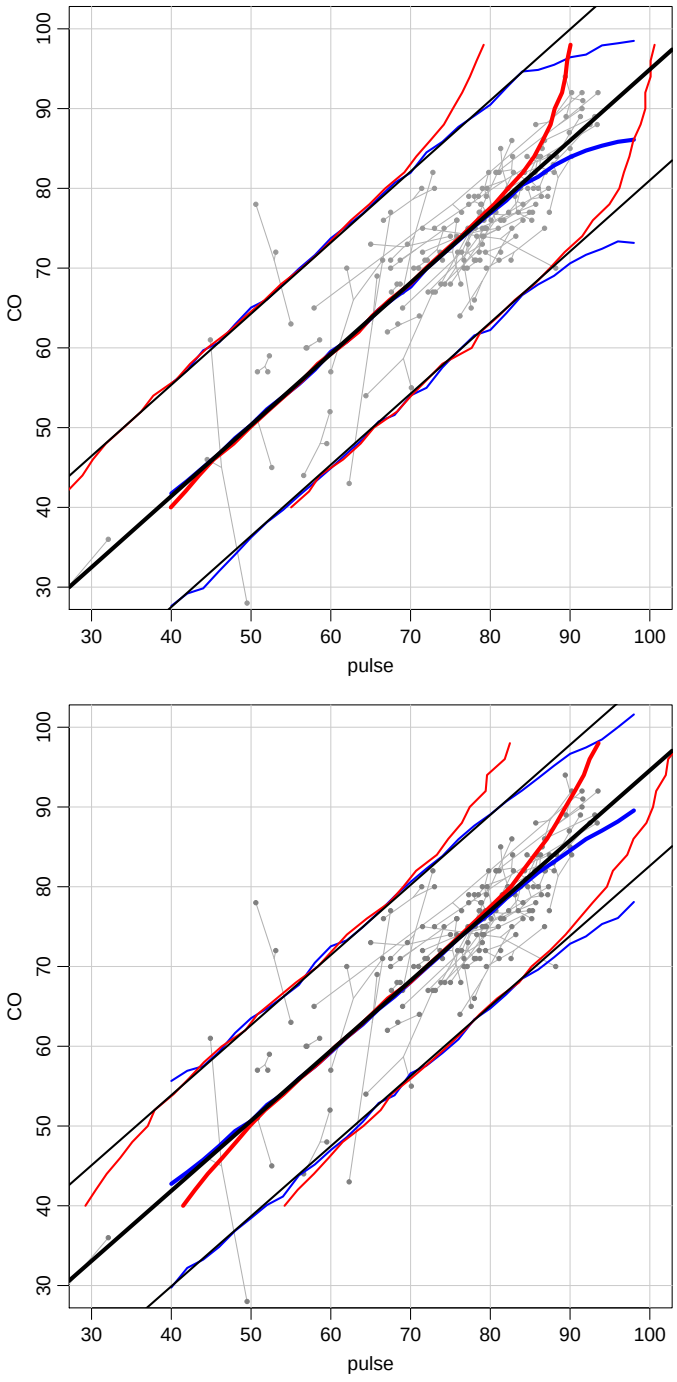

Figure 2: *Scatterplot of the oximetry data with observations from the same infant joined by line segments. The red and blue lines are the posterior medians and 2.5 and 97.5 percentiles in the predictive distribution under the model, using a grid of hypothetical observations 40,42,...,98.* **BUGS** *was run with 10,000 iterations and subsequently sampling every 10th of the next 10,000 iterations. The broken lines represents the posterior median of the analytically calculated prediction limits. The green line is the identity line.*
Top: Method by item and residual variances in model (model 3).
Bottom: Item by replicate, method by item and residual variances in model (model 4).

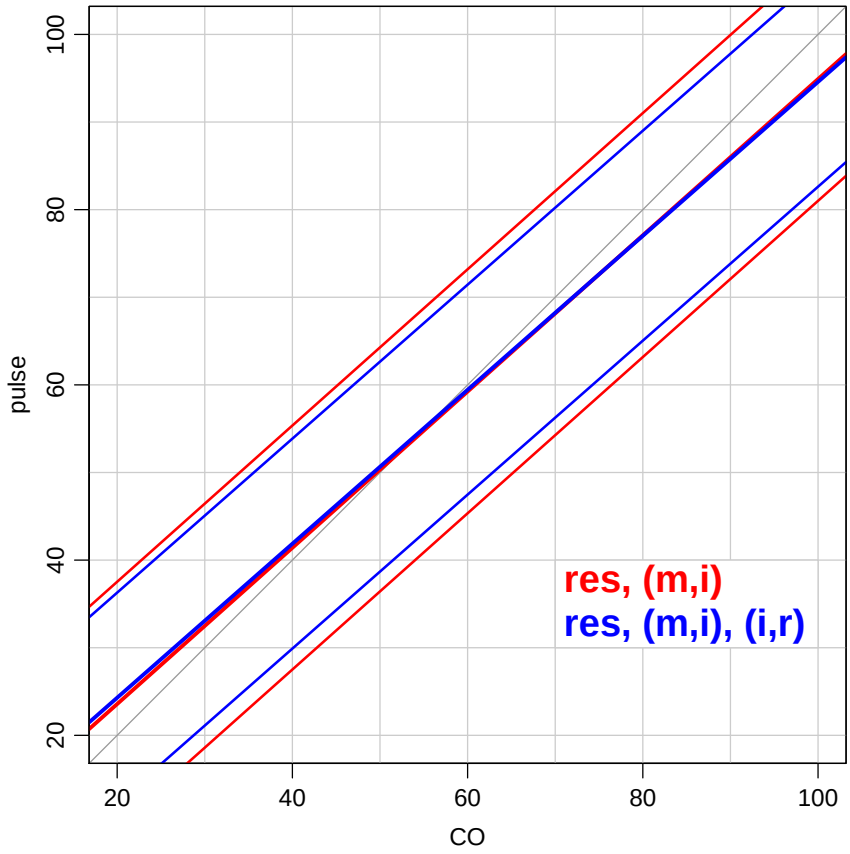

Figure 3: *Prediction limits based posterior medians of* $\alpha_{2.1}$, $\beta_{2.1}$ *and the prediction standard deviations, based om models (3) (red, wrong) and (4) (blue, correct).*

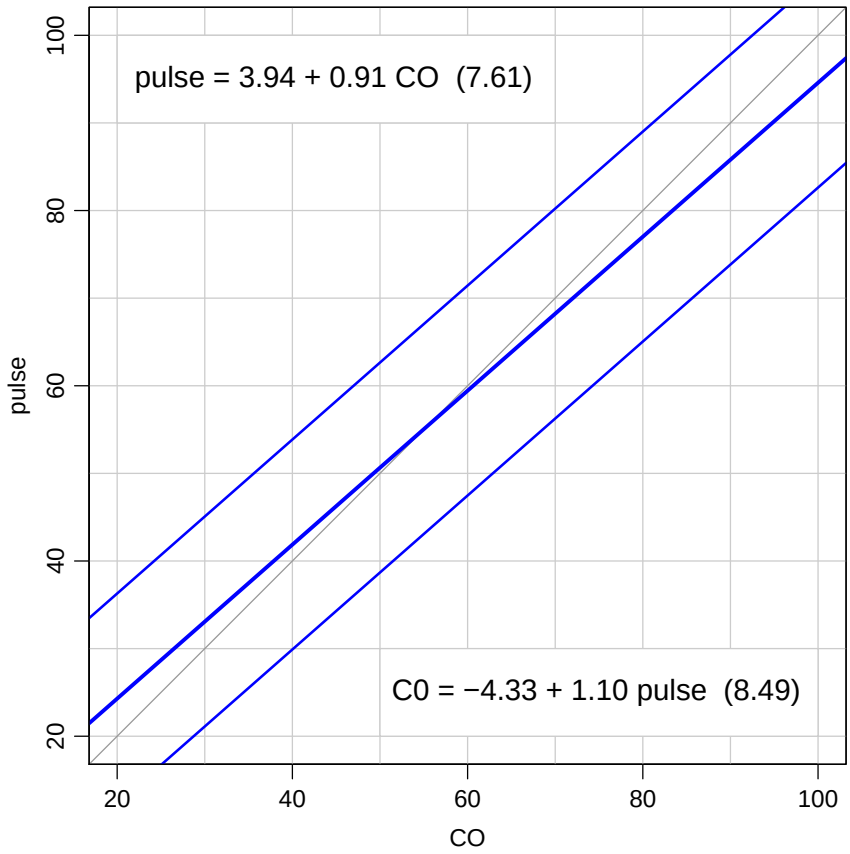

Figure 4: *Prediction limits based posterior medians of* $\alpha_{2\cdot1}$, $\beta_{2\cdot1}$ *and the prediction standard deviations based on model 4 from table 2.*

References

[1] DG Altman and JM Bland. Measurement in medicine: The analysis of method comparison studies. *The Statistician*, 32:307–317, 1983.

[2] JM Bland and DG Altman. Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, i:307–310, 1986.

[3] JM Bland and DG Altman. Measuring agreement in method comparison studies. *Statistical Methods in Medical Research*, 8:136–160, 1999.

[4] B Carstensen. Comparing and predicting between several methods of measurement. *Biostatistics*, 5(3):399–413, Jul 2004.

[5] N Best DJ Spiegelhalter, A Thomas and D Lunn. *WinBUGS User Manual 1.4.1 September 2004*. MRC Biostatistics Unit, Cambridge University & Imperial College, London, 2004.

[6] Stroup Littel, Milliken and Wolfinger. *SAS System for Mixed Models*. SAS Institute, Cary, NC., 1996.

A Computational aspects

A.1 The `MethComp` package for R

The function in the `MethComp` package that does the work is also called `MethComp`. It works by extracting information from a supplied dataframe and writing a file with `BUGS` code. The data are formatted as a data frame with four columns named `meth` (for the method of measurement), `item` (the item measured by each method), `repl` (the replicate indicating repeated measurement of the same item by the same method) and the outcome `y`, corresponding to the notation earlier in the paper. The prior for the μs is set to be uniform on an interval which corresponds to the range of all measurements expanded by 30% of its length at either end. Finally, the prior for the variances are taken to be uniform on intervals from 0 to 10 times the width of the range of data.

Initial values are then constructed from the dataframe and sent off to `WinBUGS` together with the code, using the `bugs` function from the `R2WinBUGS` package or running directly in **R** using the routines of the `BRugs` package. The returned posteriors of the αs and βs are then converted to posteriors of the parameters of the linear conversion formulae (`alphai.j`,`betai.j`) and the posterior of the point where the conversion line intersects the identity line (`id.cuti.j`). The model parameters and node names in the `BUGS` code appear in table 3.

Table 3: *Model parameters and the corresponding node in the* `BUGS` *computing code.*

Model parameter	`BUGS` parameter
μ_i	`mu[i]`
α_m	`alpha[m]`
β_m	`beta[m]`
c_{mi}	`e.mi`
a_{ir}	`e.ir`
d_{mr}	`e.mr`
e_{mir}	`e.mir`
τ_m	`sigma.mi[m]`
ν	`sigma.ir`
ω_m	`sigma.mr[m]`
σ_m	`sigma.mir[m]`

B Manual for the `MethComp` package, 0.1.1

The following is a printout of the help pages for the functions and datasets available in the `MethComp` package.

`abconv` *Derive linear conversion coefficients from a set of indeterminate coefficients*

Description

If a method comparison model is defined as $y_{mi} = \alpha_m + \beta_m \mu_i$, $m = 1, 2$ the coefficients of the linear conversion form method 1 to 2 are computed as well as the point where the linear conversion function intersects the identity line. The function is designed to work on numerical vectors of posterior samples from BUGS output.

Usage

```
abconv( a1, b1 = 1:4, a2 = NULL, b2 = NULL,
        col.names = c("alpha.2.1", "beta.2.1", "id.2.1") )
```

Arguments

`a1`	Numerical vector of intercepts for first method. Alternatively a dataframe where the vectors are selected from.
`b1`	Numerical vector of slopes for first method. If `a1` is a dataframe, this is assumed to be a numerical vector of length 4 pointing to the columns of `a1` with the intercepts and slopes.
`a2`	Numerical vector of intercepts for second method.
`b2`	Numerical vector of slopes for second method.
`col.names`	Names for the resulting three vectors.

Value

A dataframe with three columns: intercept and slope for the conversion from method 1 to method 2, and the value where the conversion is the identity.

Author(s)

Bendix Carstensen, Steno Diabetes Center, http://www.biostat.ku.dk/~bxc

References

B Carstensen: Comparing and predicting between several methods of measurement, Biostatistics, 5, pp 399-413, 2004

See Also

`BA.plot`, `MethComp`

Examples

```
abconv( 0.3, 0.9, 0.8, 0.8 )
```

`BA.plot` *Bland-Altman plot for dataframes*

Description

Computes limits of agreement and produces a Bland-Altman plot of differences versus averages for two methods of measurement. The function is just a wrapper allowing a dataframe with columns item, meth and y (and possibly repl) to be used as input to a Bland-Altman plot, using `BlandAltman`.

Usage

```
BA.plot( y1, y2, meth.names = NULL,
         mean.repl = FALSE, comp.levels=1:2, ...)
```

Arguments

`y1`	Measurements by method 1. Alternatively a dataframe with columns `meth`, `item`, `y`, and possibly `repl`.
`y2`	Corresponding measurements by method 2. Ignored if `y1` is a dataframe.
`meth.names`	Names for the two methods. Used for annotation of the plot. If not supplied and `y1` is a dataframe this is derived from the factor level names of `meth`.
`mean.repl`	Logical. If there are replicate measurements by each method should the means by `item` and `meth` be formed before further ado. WARNING: This will give too narrow limits of agreement.
`comp.levels`	Levels of the `meth` factor to compare. May be used to switch the order of the methods compared by specifying `comp.meth=2:1`.
`...`	Further arguments passed on the the `BlandAltman` function.

Value

A list with 2 elements:

`lim.agree`	A matrix of limits of agreement as rows and estimate and c.i. as columns.
`p.value`	P-value for the hypothesis that the mean difference is 0. Usually a lame thing to use.

Side effect: A Bland-Altman plot is produced using the function `BlandAltman`.

Author(s)

Bendix Carstensen, Steno Diabetes Center, http://www.biostat.ku.dk/~bxc

References

JM Bland and DG Altman: Statistical methods for assessing agreement between two methods of clinical measurement, Lancet, i, 1986, pp. 307-10

See Also

`BlandAltman`, `MethComp`

Examples

```
data( ox )
str( ox )
# A wrong and a correct plot of the data.
par( mfrow=c(1,2), mar=c(4,4,1,4) )
BA.plot( ox, mean.repl=TRUE , ymax=30 )  # Too narrow limits
BA.plot( ox, mean.repl=FALSE, ymax=30 ) # (Almost) correct limits
# The same illustrating the use of method names
par( mfrow=c(1,2), mar=c(4,4,1,4) )
BA.plot( ox, mean.repl=TRUE , meth.names=c("one","two"), ymax=30 )
BA.plot( ox, mean.repl=FALSE, meth.names=c("one","two"), ymax=30 )
```

`BlandAltman` *Bland-Altman plot of differences versus averages.*

Description

For two vectors of equal length representing measurements of the same quantity by two different methods, the differences are plotted versus the average. The limits of agreement (prediction limits for the differences) are plotted, optionally with c.i.s.

Usage

```
BlandAltman(x, y,
          x.name = NULL, y.name = NULL, maintit = "",
          cex = 1, pch = 16, col.points = "black", col.lines = "blue",
          limx=NULL, ymax=NULL, xlab=NULL, ylab=NULL,
          print = TRUE, conf.int = TRUE,
          digits = 3, alpha = 0.05,
          ... )
```

Arguments

`x`	Numerical vector of measurements by 1st method.
`y`	Numerical vector of measurements by 2nd method. Must of same length as `x`.
`x.name`	Label for the 1st method (`x`).
`y.name`	Label for the 2nd method (`y`).
`maintit`	Main title for the plot
`cex`	Character expansion for the points.
`pch`	Plot symbol for points.
`col.points`	Color for the points.
`col.lines`	Color for the lines indicating limits of agreement.
`limx`	x-axis limits.
`ymax`	Scalar. The y-axis will extend from -ymax to +ymax.
`xlab`	x-axis label.
`ylab`	y-axis label.
`print`	Logical: Should the limits of agreement and the c.i.s of these be printed?

`conf.int`	Logical: Should confidence inetrvals for the mean difference and the limits of agreement be plotted too?
`digits`	How many decimal places should be used when printing limits of agreement? Used both for the printing of results and for annotation of the plot.
`alpha`	1 minus confidence level used when computing confidence intervals and limits of agreement.
`...`	Further arguments passed on the the plot() function making the plot.

Value

A list with 2 elements:

`lim.agree`	A matrix of limits of agreement as rows and estimate and c.i. as columns.
`p.value`	P-value for the hypothesis that the mean difference is 0. Usually a lame thing to use.

Author(s)

Jaro Lajovic,⟨jaro.lajovic@mf.uni-lj.si⟩, 2004; modified 2007 by Bendix Carstensen ⟨bxc@steno.dk⟩, http://www.biostat.ku.dk/~bxc.

References

JM Bland and DG Altman: Statistical methods for assessing agreement between two methods of clinical measurement, Lancet, i, 1986, pp. 307-310.

JM Bland and DG Altman. Measuring agreement in method comparison studies. Statistical Methods in Medical Research, 8:136-160, 1999.

See Also

BA.plot, MethComp.

Examples

```
data( ox )
# Wrong to use mean over replicates
mtab <- with( ox, tapply( y, list(item, meth), mean ) )
CO <- mtab[,"CO"]
pulse <- mtab[,"pulse"]
BlandAltman( CO, pulse )
# (almost) Right to use replicates singly
mtab <- with( ox, tapply( y, list(interaction(item,repl), meth), mean ) )
CO <- mtab[,"CO"]
pulse <- mtab[,"pulse"]
BlandAltman( CO, pulse )
```

`cardiac` *Measurement of cardiac output by two different methods.*

Description

For each subject cardiac output is measured repeatedly (three to six times) by impedance cardiography (IC) and radionuclide ventriculography (RV).

Usage

```
data(cardiac)
```

Format

A data frame with 120 observations on the following 4 variables.

`meth` a factor with levels `IC` `RV`

`item` a numeric vector giving the item number.

`repl` a numeric vector with replicate number.

`y` the measuremnts of cardiac output.

Details

It is not entirely clear from the source whether the replicates are exchangeable within (method,item) or whether they represent pairs of measurements. From the description it looks as if replicates are linked between methods, but in the paper they are treated as if they were not.

Source

The dataset is adapted from table 4 in: JM Bland and DG Altman: Measuring agreement in method comparison studies. Statistical Methods in Medical Research, 8:136-160, 1999. Originally supplied to Bland & Altman by Dr LS Bowling, see: Bowling LS, Sageman WS, O'Connor SM, Cole R, Amundson DE. Lack of agreement between measurement of ejection fraction by impedance cardiography versus radionuclide ventriculography. Critical Care Medicine 1993; 21: 1523-27.

Examples

```
library( MethComp )
library( R2WinBUGS )
data(cardiac)
options( bugs.directory="c:/Stat/Bugs/WinBUGS14/" )
card.mi.ir <- MethComp( cardiac, random=c("mi","ir"), n.iter=100 )
card.mi    <- MethComp( cardiac, random=c("mi"), n.iter=100 )
card.mi.ir
card.mi
```

`Deming` *Regression with errors in both variables (Deming regression)*

Description

The function makes a regression of y on x, assuming that both x and y are measured with error. This problem only has an analytical solution if the ratio of the variances is known, hence this is required as an input parameter.

Usage

```
Deming(x, y, vr = sdr^2, sdr = sqrt(vr), boot = FALSE, alpha = 0.05)
```

Arguments

`x`	numerical variable.
`y`	numerical variable.
`vr`	The assumed known ratio of the (residual) variance of the `y`s relative to that of the `x`s. Defaults to 1.
`sdr`	do. for standard deviations. Defaults to 1. `vr` takes precedence if both are given.
`boot`	Should bootstrap estimates of standard errors of parameters be done? If `boot==TRUE`, 1000 bootstrap samples are done, if `boot` is numeric, `boot` samples are made.
`alpha`	What significance level should be used when displaying confidence intervals?

Details

The formal model underlying the procedure is based on a so called functional relationship:

$$x_i = \xi_i + e_{1i}, \qquad y_i = \alpha + \beta\xi_i + e_{2i}$$

with $\mathrm{var}(e_{1i}) = \sigma$, $\mathrm{var}(e_{2i}) = \lambda\sigma$, where λ is the known variance ratio.

The estimates of the residual variance is based on a weighting of the sum of squared deviations in both directions, divided by $n - 2$. The ML estimate would use $2n$ instead, but in the model we actually estimate $n + 2$ parameters — α, β and the n ξs.

This is not in Peter Sprent's book (see references).

Value

If `boot==FALSE` a named vector with components `alpha`, `beta`, `sigma.x`, `sigma.y`.

If `boot==TRUE` a matrix with rows `Intercept`, `Slope`, `sigma.x`, `sigma.y`, and colums giving the estimates, the bootstrap standard error and the bootstrap estimate and c.i. as the 0.5, $\alpha/2$ and $1 - \alpha/2$ quantiles of the sample.

Author(s)

Bendix Carstensen, Steno Diabetes Center, http://www.biostat.ku.dk/~bxc.

References

Peter Sprent: Models in Regression, Methuen & Co., London 1969, ch.3.4.

WE Deming: Statistical adjustment of data, New York: Wiley, 1943. [This is a reference taken from a reference list — I never saw the book myself].

See Also

MethComp

Examples

```
# Some data
x <- runif(100,0,5) + rnorm(100)
y <- 2 + 3 * x + rnorm(100,sd=2)
# Deming regression with equal variances, variance ratio 2.
Deming(x,y)
Deming(x,y,vr=2)
Deming(x,y,boot=TRUE)
# Plot data with the two classical regression lines
plot(x,y)
abline(lm(y~x))
ir <- coef(lm(x~y))
abline(-ir[1]/ir[2],1/ir[2])
abline(Deming(x,y,sdr=2)[1:2],col="red")
abline(Deming(x,y,sdr=10)[1:2],col="blue")
# Comparing classical regression and "Deming extreme"
summary(lm(y~x))
Deming(x,y,vr=1000000)
```

`fat` *Measurements of subcutaneous and visceral fat*

Description

43 persons had Subcutaneous and Visceral fat thickness measured at Steno Diabetes Center in 2006 by two observers; all measurements were done three times. The interest is to compare the measurements by the two observers. Persons are items, observers are methods, the three replicates are exchangeable within (person,observer)=(item,method)

Usage

```
data(fat)
```

Format

A data frame with 258 observations on the following 6 variables.

`Id` Person id.

`Obs` Observers, a factor with levels `KL` and `SL`.

`Rep` Replicate — exchangeable within person and observer.

`Sub` Subcutaneous fat measured in mm.

`Vic` Visceral fat measured in mm.

Examples

```
data(fat)
str(fat)
```

`hba1c` *Measurements of HbA1c from Steno Diabetes Center*

Description

Three analysers (machines) for determination of HbA1c (glycosylated haemoglobin) were tested on samples from 38 individuals. Each had drawn a venous and capillary blood sample. These were analysed on five different days.

Usage

```
data(hba1c)
```

Format

A data frame with 835 observations on the following 6 variables.

`dev` Type of machine used. A factor with levels `BR.V2`, `BR.VC` and `Tosoh`.

`type` Type of blood analysed (capillary or venous). A factor with levels `Cap` `Ven`

`item` Person-id. A numeric vector

`d.samp` Day of sampling.

`d.ana` Day of laboratory analysis.

`y` The measured value of HbA1c.

Details

In the terminology of method comparison studies, methods is the cross-classification of `dev` and `type`, and replicate is `d.ana`. It may be of interest to look at the effect of time between `d.ana` and `d.samp`, i.e. the time between sampling and analysis.

Source

Bendix Carstensen, Steno Diabetes Center.

References

These data were analysed as example in: Carstensen: Comparing and predicting between several methods of measurement, Biostatistics 5, pp. 399–413, 2004.

Examples

```
data(hba1c)
str(hba1c)
```

`MethComp` *Fit a model for method comparison studies using WinBUGS*

Description

A model linking each of a number of methods of measurement linearly to the "true" value is set up in BUGS and run via the function `bugs` from the `R2WinBUGS` package.

Usage

```
MethComp( data,
        random = c("mi", "ir"),
          beta = TRUE,
      n.chains = 3,
        n.iter = 2000,
      n.burnin = n.iter/2,
        n.thin = ceiling((n.iter - n.burnin)/1000),
bugs.directory = options("bugs.directory")[[1]],
         debug = FALSE,
       clearWD = TRUE,
bugs.code.file = "qwzx.bug",
     code.only = FALSE,
           ... )
## S3 method for class 'MethComp':
summary(object, ...)
## S3 method for class 'MethComp':
print(x, across, digits=3, ... )
```

Arguments

`data`	Data frame with variables `meth`, `item`, `repl` and `y`. `y` represents a measurement on an `item` (typically patient or sample) by method `meth`, in replicate `repl`.
`random`	Which random effects should be included in the model?. Enter `NULL` if none is desired.
`beta`	Logical. Should a slope other than 1 be allowed? If `FALSE` the bias between methods will be assumed constant.
`n.chains`	How many chains should be run by WinBUGS — passed on to `bugs`.
`n.iter`	How many total iterations — passed on to `bugs`.
`n.burnin`	How many of these should be burn-in — passed on to `bugs`.
`n.thin`	How many should samples — passed on to `bugs`.
`bugs.directory`	Where is WinBUGS (>=1.4) installed — passed on to `bugs`. The default is to use a parameter from options(). If you use this routinely, this is most conveniently set in your `.Rprofile`.
`debug`	Should WinBUGS remain open after running — passed on to `bugs`.
`clearWD`	Should the working directory be cleared for junk files after the running of WinBUGS — passed on to `bugs`.

`bugs.code.file`
: Where should the bugs code go?

`code.only`
: Should `MethComp` just create a bugs code file and a set of inits?

`...`
: Additional arguments passed on to `bugs`.

`object`
: A `MethComp` object

`x`
: A `MethComp` object

`across`
: Should the summary of conversion formulae be printed with α, β and prediction sd. across or down?

`digits`
: Number of digits after the decimal point when printing.

Details

The model set up for an observation y_{mir} is:

$$y_{mir} = \alpha_m + \beta_m(\mu_i + b_{ir} + c_{mi}) + e_{mir}$$

where b_{ir} is a random `item` by `repl` interaction (included if `"ir" %in% random`) and c_{mi} is a random `meth` by `item` interaction (included if `"mi" %in% random`). The μ_i's are parameters in the model but are not monitored — only the αs, βs and the variances of b_{ir}, c_{mi} and e_{mir} are monitored and returned. The estimated parameters are only determined up to a linear transformation of the μs, but the linear functions linking methods are invariant. The identifiable conversion parameters are:

$$\alpha_{m\cdot k} = \alpha_m - \alpha_k\beta_m/\beta_k, \quad \beta_{m\cdot k} = \beta_m/\beta_k$$

The posteriors of these are derived and included in the `posterior`, which also will contain the posterior of the variance components (the sd's, that is). Furthermore, the posterior of the point where the conversion lines intersects the identity as well as the prediction sd's between any pairs of methods are included.

The function `summary.MethComp` method gives estimates of the conversion parameters that are consistent. Clearly,

$$\text{median}(\beta_{1\cdot 2}) = 1/\text{median}(\beta_{2\cdot 1})$$

because the inverse is a monotone transformation, but there is no guarantee that

$$\text{median}(\alpha_{1\cdot 2}) = \text{median}(-\alpha_{2\cdot 1}/\beta_{2\cdot 1})$$

and hence no guarantee that the parameters derived as posterior medians produce conversion lines that are the same in both directions. Therefore, `summary.MethComp` computes the estimate for $\alpha_{2\cdot 1}$ alpha.2.1 as

$$(\text{median}(\alpha_{1\cdot 2}) - \text{median}(\alpha_{2\cdot 1})/\text{median}(\beta_{2\cdot 1}))/2$$

and the estimate of $\alpha_{1\cdot 2}$ correspondingly. The resulting parameter estimates defines the same lines.

Value

If `code.only==FALSE`, an object of class `MethComp` which is a list with three components:

`summary`
: Matrix with a summary of the posterior of the variance components and the parameters linking the methods.

`posterior`
: Dataframe with the posterior samples of the interesting parameters.

`org.summary`
: Summary of the original parameters as monitored by `WinBUGS`.

`random`
: A character sting indicationg which random effects are in the model.

`methods`
: A character string of the names of the methods.

`data`
: The original data frame used in the computations. This is intended for us in `plot.MethComp`.

Author(s)

Bendix Carstensen, Steno Diabetes Center, http://www.biostat.ku.dk/~bxc, Lyle Gurrin, University of Melbourne, http://www.epi.unimelb.edu.au/about/staff/gurrin-lyle.

References

B Carstensen: Comparing and predicting between several methods of measurement, Biostatistics, 5, pp 399-413, 2004

See Also

BA.plot,
code{BA.plot}

Examples

```
data( ox )
str( ox )
MethComp( ox, code.only=TRUE, bugs.code.file="ox-ex.bug", random=c("mi") )
shell( "type ox-ex.bug" ) # only works on windows

### These next lines only work if you properly name the path to WinBUGS
### What is written here is not necessarily correct on your machine.
library(R2WinBUGS)
# options( bugs.directory="c:/Program Files/WinBUGS14/" )
options( bugs.directory="c:/Stat/Bugs/WinBUGS14/")
ox.res <- MethComp( ox, random=c("mi"), n.iter=100 )
str( ox.res )
str( ox.res[[2]] )
print( ox.res )
```

`milk` *Measurement of fat content of human milk by two different methods.*

Description

Fat content of human milk determined by measurement of glycerol released by enzymic hydrolysis of triglycerides (Trig) and measurement by the Standard Gerber method (Gerber). Units are (g/100 ml).

Usage

```
data(milk)
```

Format

A data frame with 90 observations on the following 2 variables.

`meth` a factor with levels `Gerber Trig`

`y` a numeric vector

Source

The dataset is adapted from table 3 in: JM Bland and DG Altman: Measuring agreement in method comparison studies. Statistical Methods in Medical Research, 8:136-160, 1999. See: Lucas A, Hudson GJ, Simpson P, Cole TJ, Baker BA. An automated enzymic micromethod for the measurement of fat in human milk. Journal of Dairy Research 1987; 54: 487-92.

Examples

```
data(milk)
str(milk)
plot(milk)
plot( y[meth=="Trig"]~y[meth=="Gerber"],data=milk,
      xlab="Fat (g/100 ml; Gerber)",
      ylab="Fat (g/100 ml; Trig.)")
abline(0,1)
```

`ox` *Measurement of oxygen saturation in blood*

Description

61 children had their blood oxygen content measured at the Children's Hospital in Melbourne, either with a chemical method analysing gases in the blood (`CO`) or by a pulse oximeter measuring transcutaneously (`pulse`). Replicates are linked between methods; i.e. replicate 1 for each of the two methods are done at the same time. However, replicate measurements were taken in quick succession so the pairs of measurements are exchangeable within person.

Usage

```
data(ox)
```

Format

A data frame with 354 observations on the following 4 variables.

`meth` Measurement methods, factor with levels `CO`, `pulse`

`item` Id for the child

`repl` Replicate of measurements. There were 3 measurements for most children, 4 had only 2 replicates with each method, one only 1

`y` Oxygen saturation in percent.

Examples

```
data(ox)
str(ox)
with( ox, table(table(item)) )
par( mfrow=c(1,2), mar=c(4,4,1,4) )
BA.plot( ox, ymax=20 )
BA.plot( ox, ymax=20, mean.repl=TRUE )
```

`perm.repl` *Randomly permute replicates within methods*

Description

Replicates are randomly permuted within (item,method) in a dataframe representing a method comparison study.

Usage

```
perm.repl( data )
```

Arguments

`data`	A data frame with columns `meth`, `item`, `repl` and `y`.

Value

A dataframe where the rows (i.e. replicates) are randomly permuted within (`meth`,`item`), and subsequently ordered by (`meth`,`item`,`repl`)

Author(s)

Bendix Carstensen, Steno Diabetes Center, http://www.biostat.ku.dk/~bxc

Examples

```
# Replicates are linked, so randomly permuting them inflates the
# limits of agreement.
data(ox)
par( mfrow=c(1,2), mar=c(4,4,1,4) )
BA.plot( ox, ymax=30, digits=1 )
BA.plot( perm.repl(ox), ymax=30, digits=1 )
```

`plot.MethComp` *Plot estmiated conversion lines and formulae.*

Description

Plots the pairwise conversion formulae between methods from a `MethComp` object.

Usage

```
plot.MethComp( x,
             axlim = range( attr(x,"data")$y ),
             which,
          lwd.line = c(3,1), col.line = rep("black",2), lty.line=rep(1,2),
               eqn = TRUE, digits = 2,
              grid = FALSE, col.grid=gray(0.8),
            pl.obs = FALSE,
           col.pts = "black", pch.pts = 16, cex.pts = 0.8,
               ... )
```

Arguments

`x`	A `MethComp` object
`axlim`	The limits for the axes in the panels
`which`	Numeric vector or vector of method names. Which of the methods should be included in the plot?
`lwd.line`	Numerical vector of length 2. The width of the conversion line and the prediction limits. If the second values is 0, no prediction limits are drawn.
`col.line`	Numerical vector of length 2. The color of the conversion line and the prediction limits.
`lty.line`	Numerical vector of length 2. The line types of the conversion line and the prediction limits.
`eqn`	Should the conversion equations be printed on the plot?. Defaults to `TRUE`.
`digits`	How many digits after the decimal point shoudl be used when printing the conversion equations.
`grid`	Should a grid be drawn? If a numerical vector is given, the grid is drawn at those values.
`col.grid`	What color should the grid have?
`pl.obs`	Logical or character. Should the points be plotted. If `TRUE` or `"repl"` paired values of single replicates are plotted. If `"perm"`, replicates are randomly permuted within (item, method) befor plotting. If `"mean"`, means across replicates within item, method are formed and plotted.
`col.pts`	What color should the observation have.
`pch.pts`	What plotting symbol should be used.
`cex.pts`	What scaling should be used for the plot symbols.
`...`	Parameters to pass on. Currently not used.

Value

Nothing.

See Also

`MethComp`, `print.MethComp`

Examples

```
data( hba1c )
str( hba1c )
hba1c <- transform( hba1c, meth = interaction(dev,type),
                           repl = d.ana )
# options( bugs.directory="c:/Program Files/WinBUGS14/" )
options( bugs.directory="c:/Stat/Bugs/WinBUGS14/")
hb.res <- MethComp( hba1c, n.iter=100 )
str( hb.res )
par( ask=TRUE )
plot( hb.res )
plot( hb.res, pl.obs=TRUE, which=1:4 )
```

`plvol` *Measurements of plasma volume measured by two different methods.*

Description

For each subject (`item`) the plasma volume is expressed as a percentage of the expected value for normal individuals. Two alternative sets of normal values are used, named Nadler and Hurley respectively.

Usage

```
data(plvol)
```

Format

A data frame with 198 observations on the following 3 variables.

`meth` a factor with levels `Hurley` `Nadler`

`item` a numeric vector

`y` a numeric vector

Source

The datset is adapted from table 2 in: JM Bland and DG Altman: Measuring agreement in method comparison studies. Statistical Methods in Medical Research, 8:136-160, 1999. Originally supplied to Bland & Altman by C Doré, see: Cotes PM, Doré CJ, Liu Yin JA, Lewis SM, Messinezy M, Pearson TC, Reid C. Determination of serum immunoreactive erythropoietin in the investigation of erythrocytosis. New England Journal of Medicine 1986; 315: 283-87.

Examples

```
data(plvol)
str(plvol)
plot( y[meth=="Nadler"]~y[meth=="Hurley"],data=plvol,
      xlab="Plasma volume (Hurley) (pct)",
      ylab="Plasma volume (Nadler) (pct)" )
abline(0,1)
par( mar=c(4,4,1,4) )
BA.plot(plvol)
```

`sbp` *Systolic blood pressure measured by three different methods.*

Description

For each subject (`item`) there are three replicate measurements by three methods (two observers, J and R and the automatic machine, S). The replicates are exchangeable within method, item.

Usage

```
data(sbp)
```

Format

A data frame with 765 observations on the following 4 variables.

`meth` Methods, a factor with levels `J`(observer 1), `R`(observer 2) and `S`(machine)

`item` Person id, numeric.

`repl` Replicat number, a numeric vector

`y` Systolic blood pressure masurement, a numeric vector

Source

The dataset is adapted from table 1 in: JM Bland and DG Altman: Measuring agreement in method comparison studies. Statistical Methods in Medical Research, 8:136-160, 1999. Originally supplied to Bland & Altman by E. O'Brien, see: Altman DG, Bland JM. The analysis of blood pressure data. In O'Brien E, O'Malley K eds. Blood pressure measurement. Amsterdam: Elsevier, 1991: 287-314.

Examples

```
data(sbp)
par( mfrow=c(2,2), mar=c(4,4,1,4) )
BA.plot( sbp, comp=1:2 )
BA.plot( sbp, comp=2:3 )
BA.plot( sbp, comp=c(1,3) )
library( R2WinBUGS )
options( bugs.directory="c:/Stat/Bugs/WinBUGS14/" )
# Grossly inadequate number of iterations
sbp.1 <- MethComp( sbp, random=c("mi"), n.iter=100 )
sbp.1
```

`tab.repl` *Table of replicates by each method*

Description

Creates a table classified by method and no. of replicate measurments which in each entry has the number of items with that number of replicates on that method

Usage

```
tab.repl(data)
```

Arguments

`data`	Data frame with variables `meth`, `item`, `repl` and `y`. `y` represents a measurement on an `item` (typically patient or sample) by method `meth`, in replicate `repl`.

Value

A table classified by method and no. of replicate measurments.

Author(s)

Bendix Carstensen, ⟨bxc@steno.dk⟩

See Also

`MethComp`

Examples

```
data(ox)
tab.repl(ox)
```