

# Splus Tools for Model Selection in Nonlinear Regression

Olaf Bunke<sup>1</sup>, Bernd Droge<sup>1</sup> and Jörg Polzehl<sup>2</sup>

<sup>1</sup> Institut für Mathematik, Humboldt-Universität zu Berlin  
Unter den Linden 6, D-10099 Berlin, Germany

<sup>2</sup> Konrad-Zuse-Zentrum für Informationstechnik  
Heilbronner Str. 10, D-10711 Berlin, Germany

## Abstract

The results of analyzing experimental data using a parametric model may heavily depend on the chosen model. With this paper we describe computational tools in Splus for the adequate selection of nonlinear regression models if the intended use of the model is among the following:

1. estimation of the unknown regression function, 2. prediction of future values of the response variable, 3. calibration or 4. estimation of some parameter with a certain meaning in the corresponding field of application. Moreover, we provide programs for variance modelling and for selecting an appropriate nonlinear transformation of the observations which may lead to an improved accuracy. We describe how the accuracy of the parameter estimators is assessed by a "moment oriented bootstrap procedure". This procedure is also used for the construction of confidence, prediction and calibration intervals. The use of our tools is illustrated by an example. Help files are given in an appendix.

Key words: Computing in nonlinear regression, model selection, bootstrap, cross-validation, variable transformation, variance modelling, prediction, calibration, confidence intervals.

AMS 1991 subject classifications: 62J99, 62J02, 62P10.

## 1. Introduction

Many papers and books are concerned with the analysis of experimental data estimating the parameters in a linear or nonlinear regression model, see e.g. Bunke and Bunke [2, 3], Seber and Wild [15] or Huet, Bouvier, Gruet and Jolivet [11]. However, in practice it is usually not known, whether a certain regression model describes the unknown true regression function sufficiently well. The results of the statistical analysis may depend heavily on the chosen model. Therefore there should be a careful selection of the model, based on the scientific and practical experience in the corresponding field of application and on statistical procedures. Moreover, a nonlinear transformation of the observations or an appropriate model for the variance structure can lead to an improved accuracy. In Bunke, Droge and Polzehl [5] we have provided a methodology for model selection, variance estimation and use of transformations in nonlinear regression. The aim of this paper is to describe the handling of computational tools written in Splus for analyzing data by the proposed strategies.

We assume to have possibly replicated observations of a response variable  $Y$  at fixed values  $x_1, \dots, x_k$  of explanatory variables (or nonrandom design points), which follow the model

$$Y_{ij} = f(x_i) + \varepsilon_{ij}, \quad i = 1, \dots, k, \quad j = 1, \dots, n_i, \quad \sum_{i=1}^k n_i = n. \quad (1.1)$$

In (1.1) the regression function  $f$  is unknown and real-valued and the  $\varepsilon_{ij}$  are uncorrelated random errors with zero mean and unknown variances  $\sigma_i^2 = \sigma^2(x_i)$ . The usual assumption of a homogeneous error variance  $\sigma_i^2 = \sigma^2$  is unrealistic in many applications and one is often confronted with heteroscedasticity problems.

The analysis of the data requires in general to estimate the regression function which describes the dependence of the response variable on the explanatory variables. This is usually done by assuming that this dependence may be described by a parametric model

$$f(x) = f(x, \vartheta) \quad (\vartheta \in \Theta), \quad (1.2)$$

where the function  $f(\cdot, \vartheta)$  is known up to a  $p$ -dimensional parameter  $\vartheta \in \Theta \subseteq \mathbf{R}^p$ , so that the problem reduces to estimate this parameter using the data. As an estimate of the parameter we will employ the ordinary least squares estimator (OLSE)  $\hat{\vartheta}$ , which is a minimizer of the sum of squares

$$S(\vartheta) = \sum_{i=1}^k \sum_{j=1}^{n_i} (Y_{ij} - f(x_i, \vartheta))^2 \quad (1.3)$$

with respect to  $\vartheta$ . In practice this is the most popular approach to estimate  $\vartheta$ . Weighted LSE will be discussed later in Section 2.. We preliminarily restrict ourselves to the case of real-valued design points, that is, we will assume to have only one explanatory variable, although later versions of the functions described here will cover the case of multivariate explanatory variables.

A starting point in an approach to model selection is the idea, that even if a certain regression model is believed to be convenient for given experimental data, either because of

theoretical reasonings or based on past experience with similar data, there is seldom sure evidence on the validity of a model of the form (1.2). Therefore a possible modification of the model could lead to a better fit or to more accurate estimates of the interesting parameters. Our model selection procedure starts with a set of competitive models  $f_m$  of similar qualitative behaviour corresponding to theoretical or practical experience in the field of application. From this set we select one (or several) appropriate candidate(s) by taking into account the intended usage of the model. We allow that the aim of the analysis is one of the following objectives:

- estimation of the unknown regression function,
- prediction of future values of the response variable,
- calibration or
- estimation of some parameter with a certain meaning in the corresponding field of application.

For a more detailed description the reader is referred to Bunke, Droge and Polzehl [5].

The tools provided with this paper allow to handle the following steps of our strategy for analyzing the dependence of the response variable on the explanatory variables:

- description of the data and of a class of reasonable regression models (possibly containing only a few or even a single model) in a standardized way
- generation of initial parameter estimates
- selection of a best regression model or of a class of acceptable regression models with respect to a cross-validation criterion corresponding to one of the objectives mentioned above. This includes the choice of a Box-Cox transformation as well as of a variance model used in (i) a weighted least squares estimate (WLSE) or (ii) estimating the variances  $\sigma_i^2$ .
- assessment of the accuracy of estimates in the chosen model by a moment oriented bootstrap methodology, construction of confidence, prediction and calibration intervals.
- standardized presentation of the results at each stage of the analysis.

The remaining sections of the paper explain this in more detail.

For a detailed description of the provided Splus functions and the structure of the resulting Splus objects we refer to the help files contained in the Appendix. The names of the regression models utilized in the examples of the helpfiles refer to the corresponding equation numbers in Bunke, Droge and Polzehl [5].

**Example** (*Esterase count data*). For the sake of illustration we discuss the statistical analysis of the esterase count data example in Carroll and Ruppert [7] (Table 2.3, pp. 46-47). In an assay for the concentration of an enzyme esterase, the observed concentration of esterase was recorded, and then in a binding experiment the number of bindings were counted.

In the original treatment of the example as a calibration problem, the objective is to take observed new counts and infer the corresponding concentration of esterase. Alternatively, another objective could be to estimate the regression function to obtain a good model for the dependence between esterase concentration and the number of bindings.

## 2. Model Selection Procedure

### 2.1 Initialization

Suppose one has to analyze the data mentioned in the introduction. The first step is to set up the data in a standard format under Splus. The functions provided here assume, that the data are given as a list with numerical components  $x$  (values of the explanatory variable) and  $y$  (values of the response variable), and character components  $xname$  and  $yname$  (will be used in summaries and plots as labels for  $x$  and  $y$ ).  $x$  and  $y$  have to be of the same length. For our example the data (source: Carroll and Ruppert [7]) may be provided by

```

esterase <- list{x=c(6.4, 6.7, 8.0, 8.1, 8.4, 8.6, 9.0, 9.5, 10.5, 10.8, 11.1, 11.6,
  12.1, 12.6, 12.8, 13.1, 13.3, 13.8, 13.9, 14.1, 14.6, 6.5, 7.8, 8.0, 8.2, 8.6,
  8.8, 9.2, 10.3, 10.6, 10.9, 11.6, 11.8, 12.3, 12.8, 13.1, 13.3, 13.7, 13.8,
  14.0, 14.4, 14.6, 14.6, 15.0, 15.2, 15.8, 16.0, 16.1, 16.9, 17.1, 17.7, 18.8,
  19.2, 20.5, 20.8, 20.9, 21.4, 21.8, 23.0, 23.5, 23.8, 24.4, 25.2, 25.5, 27.2,
  27.7, 29.0, 30.8, 35.2, 38.6, 40.5, 41.2, 44.5, 46.6, 52.1, 14.8, 15.2, 15.2,
  15.9, 16.0, 16.4, 17.0, 17.5, 18.1, 19.0, 20.5, 20.8, 20.8, 21.2, 21.8, 22.1,
  23.2, 23.7, 24.2, 24.6, 25.2, 26.9, 27.4, 29.0, 30.3, 33.6, 38.2, 39.1, 40.9,
  41.7, 45.0, 52.0, 52.4, 6.1, 3.1, 11.2, 5.6, 11.0), y=c(84, 86, 104, 96,
  124, 79, 79, 203, 191, 167, 116, 170, 233, 115, 201, 144, 139, 154, 97, 288,
  239, 85, 127, 107, 130, 105, 153, 100, 159, 93, 100, 97, 131, 256, 219, 215,
  268, 249, 226, 329, 255, 317, 216, 301, 389, 271, 148, 315, 126, 340, 276, 262,
  336, 393, 270, 343, 270, 296, 409, 381, 529, 566, 418, 435, 208, 412, 474, 438,
  416, 717, 695, 597, 718, 599, 921, 266, 193, 278, 250, 103, 256, 137, 136, 342,
  486, 354, 459, 260, 474, 416, 317, 376, 466, 412, 369, 531, 472, 646, 595, 527,
  635, 695, 1042, 239, 1006, 778, 789, 679, 52, 28, 373, 166, 423), xname =
  "Amount of esterase", yname = "Observed count")
attr(esterase, "doc") <- "Esterase Count Data"

```

In Carroll and Ruppert [7] the linear model

$$f(x, \vartheta) = \vartheta_1 + \vartheta_2 x \quad (2.1)$$

is used to analyze the data. Alternative models from the catalogue given in Bunke, Droge and Polzehl [5] could be

$$f(x, \vartheta) = \vartheta_1 \exp[\vartheta_2 / (x + \vartheta_3)] \quad (2.2)$$

$$f(x, \vartheta) = \vartheta_1 - \vartheta_2 \vartheta_3^x \quad (2.3)$$

$$f(x, \vartheta) = \vartheta_1 (1 - \exp(-\vartheta_2 x))^{\vartheta_3} \quad (2.4)$$

$$f(x, \vartheta) = \vartheta_1 + \vartheta_2 x^{\vartheta_3} \quad (2.5)$$

$$f(x, \vartheta) = \vartheta_1(1+x)^{\vartheta_2} \quad (2.6)$$

$$f(x, \vartheta) = \vartheta_1 x^{\vartheta_2 x - \vartheta_3} \quad (2.7)$$

$$f(x, \vartheta) = \frac{x}{\vartheta_1 + \vartheta_2 x + \vartheta_3 \sqrt{x}} \quad (2.8)$$

$$f(x, \vartheta) = \frac{1}{\vartheta_1 + \vartheta_2 x^{\vartheta_3}} \quad (2.9)$$

$$f(x, \vartheta) = \frac{\vartheta_2 + \vartheta_3 x}{1 + \vartheta_1 x} \quad (2.10)$$

$$f(x, \vartheta) = \vartheta_1 + \vartheta_2 \log(x + \vartheta_3) \quad (2.11)$$

$$f(x, \vartheta) = \frac{\vartheta_1 + \vartheta_3 x}{1 + \vartheta_2 x + \vartheta_4 x^2}. \quad (2.12)$$

Each regression model from this class will be specified by a list with components `$fkt` (containing an expression (third element of a formula object) employed to evaluate the regression model, with the explanatory variable denoted by `xr` and parameters denoted by `p1, p2, ...`), `$p` (number of parameters), `$inv` (containing an expression (third element of a formula object) used to evaluate the inverse of the regression model, with dependent variable denoted by `yr` and parameters denoted by `p1, p2, ...`) and `$name` (character string taken as a name for the model). The description of model (2.2) would be generated by

```
fkt2 <- yr ~ p1 * exp(p2/(xr + p3))
inv2 <- xr ~ p2/log(yr/p1) - p3
model2 <- list(fkt=fkt2[[3]], inv=inv2[[3]], p=3, name="Model (2.2)")
```

The class of models  $f_m$  under consideration is generated by

```
fktlesterase <- list(model11, model2, ..., model12)
```

The most crucial part in nonlinear regression is to find reasonable initial estimates of the parameters. This may even be impossible if the model under consideration is not well defined for the given design and shape of the data. To get rough starting values one should employ properties of the model and of the data like monotonicity and asymptotic behaviour. We provide the function `parinit` as an interactive tool to find initial parameter estimates successively for all models in the class. The function uses stochastic search minimizing the residual sum of squares (1.3) to find reasonable initial parameter estimates. Starting values can be supplied or generated randomly during the analysis. The original regression problem can be substituted by a regression problem obtained by clustering (see Hartley and Booker [10]) to get likely starting values. The parameter estimates found by stochastic search are improved by numerical optimization. Plots of the fitted models, residual plots and residual statistics are provided to assist the decision between accepting and rejecting the initial estimates. The calculated estimates can be accepted or alternatively the search can be continued with possibly new starting values or also the model can be excluded from the analyses. Information about initial estimates is added to the list describing the class of models under consideration (see Appendix). In our example initial parameter estimates can be obtained by

```
fktlesterase <- parinit(esterase, fktlesterase, graph=T)
```

The results may be summarized by

`print(fktlesterase)`

## 2.2 Variance Modelling and Estimation

The observation variances  $\sigma_i^2$  may be estimated by the intra sample estimates

$$s_i^2 = \frac{1}{n_i - 1} \sum_{j=1}^{n_i} |Y_{ij} - \bar{Y}_i|^2, \quad \bar{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij}, \quad (2.13)$$

if there are enough replications ( $n_i$  relatively large!). In such an exceptional situation these estimates may be utilized for calculating WLSE's.

An improved variance estimation may be possible using alternative variance models and taking into consideration, that it is possibly not sure that a certain variance model is adequate and moreover it is even not sure that a certain structure of the regression function is adequate. Our procedure (see Bunke, Droge and Polzehl [5, 6]) is based on a least squares fitting of alternative variance models to conveniently defined "observations"  $z_1, \dots, z_k$ . The "observations"  $z_i$  are defined in such a way, that they have (roughly) the variances  $\sigma_i^2$  as their expectation, as it is exactly the case for the estimates  $s_i^2$  given by (2.13). Assuming ordered univariate values  $x_1 < x_2 < \dots < x_k$  of the independent variable we make use of the "observations"

$$z_i := \begin{cases} s_i^2 & \text{if } n_i \geq 2 \\ \frac{1}{6} |2e_i - e_{i+1} - e_{i-1}|^2 & \text{if } n_i = 1 \quad (\text{no replications}). \end{cases} \quad (2.14)$$

Here we use the residuals

$$e_i = Y_{i1} - f_{m_*}(x_i, \hat{\vartheta}_{m_*}) \quad (2.15)$$

in employing the (best fitting) model  $f_{m_*}$ . This model is chosen among the admitted models  $f_m$  as that with smallest sum  $S(\hat{\vartheta}_m)$  of squared errors ( $\hat{\vartheta}_m$ : OLSE under  $f_m$ , see (1.3)). In the case of  $n_1 = 1$  and  $n_k = 1$  we employ  $z_1 = \frac{1}{2} |e_1 - e_2|^2$  and  $z_k = \frac{1}{2} |e_k - e_{k-1}|^2$ .

We fit alternative variance models of the form  $g(x, \sigma^2, \tau)$  to the above "observations"  $z_i$  by minimization of the sum of squares

$$S(\sigma^2, \tau) = \left\{ \sum_{i:n_i \geq 2} (n_i - 1) |s_i^2 - g(x_i, \sigma^2, \tau)|^2 + \sum_{i:n_i=1} |z_i - g(x_i, \sigma^2, \tau)|^2 \right\}. \quad (2.16)$$

We consider six alternative variance models, which are especially useful and have been proposed in the literature, see Carroll and Ruppert [7]:

(1) The first is the exponential model

$$g(x, \sigma^2, \tau) = \sigma^2 [f_{m_o}(x, \hat{\vartheta}_{m_o}) + a]^\tau \quad (2.17)$$

where we apply the model  $f_{m_o}$  ( $m_o = \hat{m}$  or  $\tilde{m}$ ) chosen by the procedure which will be described in Subsection 2.4 (using homogeneous variance). The constant  $a$  is chosen as  $a := .1 \cdot \max_i f_{m_o}(x_i, \hat{\vartheta}_{m_o}) - 1.1 \cdot \min_i f_{m_o}(x_i, \hat{\vartheta}_{m_o})$ .

(2) As an alternative model in place of (2.17) one may fit the model

$$g(x, \sigma^2, \tau) = \sigma^2 \exp[\tau f_{m_o}(x, \hat{\vartheta}_{m_o})]. \quad (2.18)$$

(3) Past experience (or the residuals after fitting by ordinary least squares) may suggest, that the variances  $\sigma_i^2$  do not vary monotonously with the mean  $f(x_i)$ , but behave approximately like a unimodal function of  $f(x_i)$  or shows the reverse behaviour. Then we could alternatively fit a quadratic variance model

$$g(x, \sigma^2, \tau) = \sigma^2 + \tau_1(f_{m_o}(x, \hat{\vartheta}_{m_o}) - \bar{f}_{m_o}) + \tau_2(f_{m_o}(x, \hat{\vartheta}_{m_o}) - \bar{f}_{m_o})^2, \quad (2.19)$$

where

$$\bar{f}_{m_o} := \frac{1}{k} \sum_{i=1}^k f_{m_o}(x_i, \hat{\vartheta}_{m_o}). \quad (2.20)$$

(4) A bell-shaped variance model

$$g(x, \sigma^2, \tau) = \sigma^2 + \sigma^2 \tau \left( \bar{f} - f_{m_o}(x, \hat{\vartheta}_{m_o}) \right) \left( f_{m_o}(x, \hat{\vartheta}_{m_o}) - \underline{f} \right), \quad (2.21)$$

$\underline{f} = \min_i f_{m_o}(x, \hat{\vartheta}_{m_o})$ ,  $\bar{f} = \max_i f_{m_o}(x, \hat{\vartheta}_{m_o})$ , has less parameters and may be useful e.g. for count data.

(5) Sometimes also a simple linear model may be appropriate:

$$g(x, \sigma^2, \tau) = \sigma^2 + \tau(f_{m_o}(x, \hat{\vartheta}_{m_o}) - \bar{f}_{m_o}). \quad (2.22)$$

(6) In many cases a homogeneous variance estimate  $\hat{\sigma}^2$  would be more accurate than a heteroscedastic estimate  $\hat{\sigma}_i^2$  determined by a variance model, especially when the differences between the variances  $\sigma_i^2$  are moderate or small. Thus we would fit a constant model  $g(x, \sigma^2) \equiv \sigma^2$  to our observations  $(x_i, z_i)$  and obtain

$$\hat{\sigma}^2 = \frac{1}{n-q} \left\{ \sum_{i:n_i \geq 2} \sum_{j=1}^{n_i} |Y_{ij} - \bar{Y}_i|^2 + \sum_{i:n_i=1} z_i \right\}, \quad (2.23)$$

where  $q$  is the number of points  $x_i$  with  $n_i \geq 2$ .

Unfortunately, some of the six models have the disadvantage of possibly leading to negative estimates  $\hat{\sigma}_i^2 := g(x_i, \hat{\sigma}^2, \hat{\tau})$  for some design points  $x_i$ . We replace the negative (and also very small) estimates by some fixed small positive value, say by  $\hat{\sigma}_0^2 := 0.1\hat{\sigma}^2$ , where  $\hat{\sigma}^2$  is the homogeneous variance estimate (2.23). The variance estimate will be then

$$\bar{\sigma}_i^2 := \max\{\hat{\sigma}_i^2, \hat{\sigma}_0^2\} \quad \text{for } i = 1, \dots, k. \quad (2.24)$$

For selecting a convenient variance model to calculate WLSE's we refer to Subsection 2.4. For the purpose of variance estimation we propose to choose among the six variance estimators  $\bar{\sigma}_i^2$  one with smallest value of the "cross-validation" criterion

$$CV := \frac{1}{n-q} \sum_{i=1}^k w_i |z_i - \bar{\sigma}_{-i}^2|^2. \quad (2.25)$$

The weights  $w_i$  are either  $n_i - 1$  (for  $n_i \geq 2$ ) or 1 (for  $n_i = 1$ ), whereas the estimates  $\bar{\sigma}_{-i}^2$  are (essentially) estimators calculated by fitting the variance model leaving out the observations  $Y_{hl}$  entering in the calculation of  $z_i$  (for details we refer to Bunke, Droge and Polzehl [6]).

### 2.3 Variable Transformations

In several situations it may be advantageous to fit a regression model to transformed observations

$$Y_{ij}^T := T[Y_{ij}] \quad (i = 1, \dots, k; j = 1, \dots, n_i) \quad (2.26)$$

of the dependent variable, e.g. using a logarithmic transformation ( $T[y] = \log y$ ).

Such situations may be:

1. It is known, that a certain parametric model  $f(x, \vartheta)$  gives a good fit to the transformed observations  $Y_{ij}^T$  and small residuals  $Y_{ij}^T - f(x_i, \hat{\vartheta}_T)$ , where  $\hat{\vartheta}_T$  denotes the ordinary least squares estimate in such a fitting. In this case the inversely transformed model

$$f_{T^{-1}}(x, \vartheta) := T^{-1}[f(x, \vartheta)] \quad (2.27)$$

will be employed to estimate the regression function of the original observations by  $f_{T^{-1}}(x, \hat{\vartheta}_T)$

or

2. The variances of the variables  $Y_{ij}$  are markedly heteroscedastic and the transformation reduces or eliminates the heteroscedasticity: small differences  $\text{Var } Y_{ij}^T - \text{Var } Y_{h\ell}^T$ .

We propose to use the OLSE, even if the variances of the original or of the transformed observations are heteroscedastic. The basic idea is that the OLSE does not suffer from the possibly pernicious influence of errors in estimating the variances on the otherwise higher efficiency of the WLSE. *Therefore it may happen in some cases, that the simple calculation of an OLSE for conveniently transformed observations and a corresponding selected model leads to a more accurate estimation of parameters (or of the regression function) or to a better calibration than using a WLSE together with conveniently selected regression and variance models .*

With this transformation approach we have an alternative approach to parameter estimation under heteroscedastic variances, which may (or may not) be better than the WLSE approach discussed in Subsection 2.2. A choice between both should be done on the basis of the corresponding cross-validation value, as it will be explained in Subsection 2.4. Always it will be useful to compare the results of the analysis of the original observations with that of transformed observations and to make a decision about a possible transformation using a data dependent criterion.

The selection of a suitable transformation  $T$  is simplified, if  $T$  is restricted to a sufficiently small but flexible class of transformations. Such a class is the class of (modified) Box-Cox-transformations

$$T_\lambda(y) = \begin{cases} \{(s^{-1}[y + a_\lambda])^\lambda - 1\}/\lambda & \text{for } \lambda \neq 0 \\ \ln(s^{-1}[y + a_o]) & \text{for } \lambda = 0 \end{cases} \quad (2.28)$$

defined for real  $\lambda$  and convenient constants  $a_\lambda$  and where

$$s^2 := \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^{n_i} |Y_{ij} - \bar{Y}|^2, \quad \bar{Y} := \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^{n_i} Y_{ij}. \quad (2.29)$$



In many cases it will be sufficient to try some of the transformations given by a finite set of different  $\lambda$ -values (including  $\lambda = 0$  at least). A standard choice is:

$$\lambda = 0, 1/2, -1/2, 1, -1, 2, -2, \quad (2.30)$$

The values of the constants  $s_\lambda$  and  $a_\lambda$  are specified to allow a certain range and nonlinearity of the transformation (see Droge [8] and Bunke, Droge and Polzehl [5]).

We consider two main situations. If the objective is the estimation of the values of the true regression function  $f$  or the prediction of values of the original dependent variable  $y$  (and *not* a good fitting or prediction for any transformed dependent variable), these situations are:

**Case A:** The admitted models  $f_m$  (*perhaps only one!*) are considered as very good approximations to the regression function.

**Case B:** It is not known, which models are good approximations to the regression function.

In the case A the fit to the transformed observations  $Y_{ij}^T$  will be using the transformed regression functions

$$f_{T,m}(x, \vartheta) := T[f_m(x, \vartheta)] \quad (2.31)$$

and the corresponding LSE  $\hat{\vartheta}_{T,m}$  minimizing

$$S^{T,m}(\vartheta) := \sum_{i,j} |Y_{ij}^T - f_{T,m}(x_i, \vartheta)|^2. \quad (2.32)$$

In the case B we have to replace in (2.32) the models  $f_{T,m}$  by  $f_m$ .

## 2.4 Model Selection Criteria

As discussed in Bunke, Droge and Polzehl [5, 6], using the residual sum of squares  $S(\hat{\vartheta})$  or a visual impression from a graphical representation of the estimated regression curve together with the observations may lead to "overparametrized" models  $f(x, \vartheta)$ . Instead criteria incorporating the effects of variability of the parameter estimates in a given model should be employed. In the following we present appropriately defined cross-validation criteria for the objectives of estimation of the unknown regression function, prediction, calibration and estimation of a parameter of the form

$$\gamma = \gamma[f(x_1), \dots, f(x_k)], \quad (2.33)$$

see Bunke, Droge and Polzehl [5, 6] for a more detailed discussion. These cross-validation criteria are used to assess the performance of a model for the respective objective. This allows to compare different models in combination with a variance model or a Box-Cox-Transformation, using a specified value  $\lambda$ , and to select a best model or a set of acceptable candidates.

We give the criteria for selection of a regression model  $m$  in combination with a transformation  $T = T_\lambda$  (case **A**, transformation of both the response variable  $Y$  and the regression function  $f_m$ ). In case of using a variance model, no transformation is performed (transformation with  $\lambda = 1, a_\lambda = 0, s_\lambda = 1$ ) and WLSE's based on the variance model are used as

parameter estimates instead of OLSE's. For transformation case **B**  $f_m(x, \vartheta)$  has to be replaced by  $T^{-1}[f_m(x, \vartheta)]$ .

If the objective of the analysis is primarily the estimation of the regression function or curve, that is, of the values of the regression function itself over a region  $\mathcal{X}$  of interest (and only secondarily the analysis of its properties and the estimation of some of its parameters), then the weighted cross-validation criterion

$$C = \sum_{i=r}^s c_i \sum_{j=1}^{n_i} |Y_{ij} - f_m(x_i, \hat{\vartheta}_T^{ij})|^2 \quad (2.34)$$

is a convenient criterion characterizing the performance of the model  $f_m(x, \vartheta)$ . Here  $\hat{\vartheta}_T^{i,j}$  denotes the OLSE under transformation (2.32) calculated from the  $n-1$  observations left after deleting the observation  $(x_i, Y_{ij})$ . Its numerical calculation will be easy for well parameterized models using  $\hat{\vartheta}$  as a starting value. Further we assume in (2.34) that

$$c_i = (s - r + 1)^{-1} n_i^{-1}$$

and that the values  $x_r, x_{r+1}, \dots, x_s$  of the independent variable are in  $\mathcal{X}$ , while the other values are those (if any!) not contained in  $\mathcal{X}$ . If the independent variable is univariate and its values  $x_i$  are ordered according to their magnitude and if  $\mathcal{X} = [a, b]$  is an interval with

$$a \leq x_r < \dots < x_s \leq b, \quad (2.35)$$

then we apply weights  $c_i$  with

$$c_i = d_i [2n_i(b-a)]^{-1}, \quad (2.36)$$

$$d_i = x_{i+1} - x_{i-1} \quad (r < i < s), \quad (2.37)$$

$$d_r = 2(x_{r+1} - x_r), \quad d_s = 2(x_s - x_{s-1}). \quad (2.38)$$

If the user would not like to specify the interval  $[a, b]$ , then  $a = x_1$  and  $b = x_k$  should be the standard values. In the definition of the cross-validation criterion (2.34) we have introduced weights in order to take into account the distances between the different design points as well as the number of replications. A more detailed reasoning for choosing the weights just as in (2.36) is given in Bunke, Droge and Polzehl [5, 6], where  $C$  is characterized as an estimate of the mean squared error in estimating the values of the regression function. If the values  $x_i$  are equidistant and all contained in the interval  $[a, b]$  and if there are no replications ( $n_i = 1$ ), then the weights are identical:  $c_i = k^{-1}$  for  $i = 1, \dots, k$ .

The criterion (2.34) will also be convenient, if the estimated regression function will be used to predict by  $f_m(x, \hat{\vartheta}_T)$  the future values  $Y^*(x)$  of the dependent variable for given values  $x$  in  $[a, b]$  of the explanatory variable, assuming, that their "distribution" is represented to a certain extend by the design points  $x_r, \dots, x_s$ .

For some models  $f_m$  the estimates  $f_m(x_i, \hat{\vartheta}_T^{i,j})$  may not be defined for some  $i$ , e.g. in the exponential model (2.2) with  $\vartheta_1 > 0$  and  $\vartheta_2 < 0$ , the value  $f(x, \vartheta)$  tends to 0 for  $x \downarrow x_o = -\vartheta_3$  (convergence from the right) while it tends to  $\infty$  for  $x \uparrow x_o$  (convergence from the left). Such

cases are not disturbing, if in place of  $C$  we *always apply* the following modified cross-validation criterion (*full cross-validation*, see Bunke, Droge and Polzehl [6] and Droge [9]):

$$\tilde{C} = \sum_{i=r}^s c_i \sum_{j=1}^{n_i} |Y_{ij} - \tilde{y}_{ij}|^2, \quad (2.39)$$

where

$$\tilde{y}_{ij} = f_m(x_i, \tilde{\vartheta}_{T,m}^{i,j}) \quad (2.40)$$

and where  $\tilde{\vartheta}_{T,m}^{i,j}$  is the OLSE under transformation (2.32) calculated under the substitution of just the observation  $Y_{ij}$  by  $\hat{Y}_i = f_m(x_i, \hat{\vartheta}_{T,m})$ .

In a calibration problem a modified cross-validation criterion could be adequate assuming a calibration is demanded for observations  $Y$  of the dependent variable corresponding to a unknown fixed value  $x$  of the explanatory variable. We assume the value  $Y$  to be in the interval  $(\tilde{a}, \tilde{b})$  and estimate  $x$  by  $\hat{x}_m(Y, \hat{\vartheta}_{T,m})$ , where the calibration function  $\hat{x}_m(Y, \vartheta)$  is given by the inverse of  $f_m(x, \vartheta)$  as a function of  $x$ . The criterion is:

$$CC = \frac{1}{N} \sum_{i=1}^k \sum_{j \in J_i} |x_i - \hat{x}_m(Y_{ij}, \hat{\vartheta}_{T,m}^{i,j})|. \quad (2.41)$$

Here  $J_i$  contains all indices  $j = 1, \dots, n_i$  with  $\tilde{a} < Y_{ij} < \tilde{b}$  and  $N$  is the number of observations  $Y_{ij}$  in the interval  $(\tilde{a}, \tilde{b})$ . In case of a nonlinear model  $f_m$  it may occur that the value of  $\hat{x}_m(Y_{ij}, \hat{\vartheta}_{T,m}^{i,j})$  is not defined. In such cases or if  $\hat{x}_m(Y_{ij}, \hat{\vartheta}_{T,m}^{i,j})$  is outside of the interval  $[a, b]$  on which the regression function  $f_m(x, \vartheta)$  is considered, we use in the definition of the criterion the values of  $a$  or  $b$  instead of  $\hat{x}_m$ , depending on the monotonicity and the value of  $f_m(x, \hat{\vartheta}_{T,m})$ .

If the objective is to estimate a parameter of the form (2.33), then an adequate criterion would be

$$CG = \frac{1}{n_{rs}} \sum_{i=r}^s \sum_{j=1}^{n_i} |\gamma[\bar{Y}_1, \dots, \bar{Y}_k] - \gamma[f_m(x_1, \hat{\vartheta}_{T,m}^{i,j}), \dots, f_m(x_k, \hat{\vartheta}_{T,m}^{i,j})]|, \quad (2.42)$$

where

$$\bar{Y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} Y_{ij} \quad \text{and} \quad n_{rs} = \sum_{i=r}^s n_i. \quad (2.43)$$

(2.42) may be interpreted as a jackknife approximation to the mean absolute error for the estimate  $\hat{\gamma}$  (see Bunke, Droge and Polzehl [5, 6]).

The criterion (2.42) is only sensible if all replication sizes  $n_i$  are large or otherwise if the parameter (2.33) involves weighted sums of many values  $f(x_i)$ .

## 2.5 Tools for Model Selection

We provide the Splus functions *selectmv* (selection of regression and variance models) and *selectmt* (selection of regression models and transformations) to calculate the criteria given in the last subsection for a given class of regression models. In both functions it is assumed that the data and the class of regression functions are specified as described in Subsection 2.1. For detailed information about parameters and the returned value we refer to the appendix.

Figure 1: Esterase count data: Reference model and fitted variance models

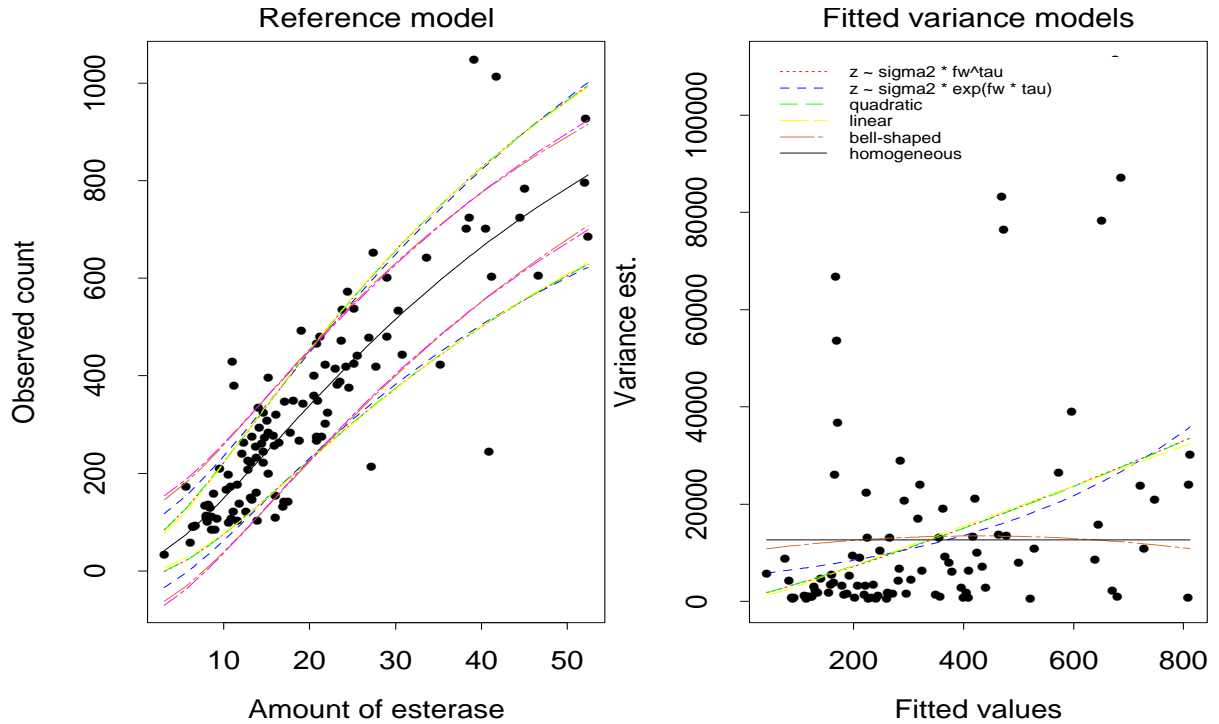


Table 1: Values of the cross-validation criterion  $CV$  and of the parameter estimates for all variance models.

Variance model	$CV$ -value (2.25)	Values of the estimates for		
		$\sigma^2$	$\tau$ ( $\tau_1$ )	$\tau_2$
(2.17)	432615620	9.79	1.208	
(2.18)	458925519	5257.6	0.00237	
(2.19)	447790756	12288.	39.331	0.010239
(2.22)	<u>412023686</u>	12675	40.986	
(2.21)	462674900	10818	0.017683	
homogeneous	453078924	12610.		

To illustrate the model selection process performed by the two functions we reconsider our example. We assume that our main interest is in calibration, i.e. in estimating the amount of enzyme esterase corresponding to a newly observed value of the response variable.

To select a combination of regression and variance model suitable for calibration we apply *selectmv* as follows:

```
esterase.vcal <- selectmv(esterase, fktlesterase, graph=T)
```

In a first step candidates for the reference regression models required in the variance modelling

Table 2: Values of the cross-validation criteria  $C$  and  $CC$  for all combinations of a regression and a variance model.

Regression model	crit.	Variance model					
		(2.17)	(2.18)	(2.19)	(2.22)	(2.21)	homogeneous
(2.2)	C	15859	15919	15861	15853	15812	15860
	CC	3.87	3.87	3.868	3.869	3.836	3.835
(2.3)	C	16173	16200	16174	16176	16003	16058
	CC	3.911	3.865	3.912	3.929	3.822	3.815
(2.4)	C	15910	15951	15912	15911	15852	15892
	CC	3.853	3.832	3.852	3.861	3.785	<u>3.784</u>
(2.5)	C	16406	16439	16406	16405	16219	16281
	CC	3.939	3.894	3.94	3.956	3.868	3.856
(2.6)	C	17089	17032	17088	17098	16444	16515
	CC	4.15	4.162	4.151	4.149	4.318	4.3
(2.7)	C	15873	15961	15873	15861	15868	15917
	CC	no explicit inverse model					
(2.8)	C	15813	15862	15815	<u>15810</u>	15878	15903
	CC	4.458	4.641	4.457	4.514	6.908	6.679
(2.9)	C	15933	15975	15935	15933	15871	15913
	CC	3.854	3.833	3.853	3.861	3.788	3.786
(2.10)	C	16188	16219	16189	16190	16027	16083
	CC	3.914	3.868	3.914	3.931	3.828	3.821
(2.11)	C	16200	16234	16201	Inf	16049	16105
	CC	3.915	3.871	Inf	Inf	3.835	3.827
(2.12)	C	15997	16070	15998	15988	16274	16278
	CC	3.861	3.876	3.859	3.859	3.874	3.877
(2.1)	C	16469	16560	16471	16437	16380	16435
	CC	3.973	3.974	3.973	3.978	4.139	4.115

(see Subsection 2.2) are specified interactively using graphical menus provided by *selectmv*. The reference models  $m_*$  (employed in the definition of residuals (2.15)) and  $m_0$  (used as arguments of the variance models (2.17) - (2.22)) are chosen as the models minimizing the residual sum of squares (1.3) and the cross-validation criterion (2.34, without transformation) and found to be (2.12) and (2.2), respectively.

The variance models introduced in Subsection 2.2 are analyzed. The fits of the variance models are illustrated as, for our example, in Figure 1. The left hand panel exhibits the fit of the reference model  $m^*$  together with pointwise one-sigma regions based on the six different variance models. The right hand panel shows the fit of the variance models to the z-values. The values of the criterion for variance estimation (2.25) are given in Table 1.

Figure 2: Plot of the best combination of regression and variance model (Prediction)

model: (2.8)  $yr = xr/(p1 + p2 * xr + p3 * \sqrt{xr})$  RSS = 10440 C = 15810

$p1 = 0.109$   $p2 = 0.002$   $p3 = -0.019$

Variance model (WLSE):  $\sigma^2 + \tau * (rfw - \text{mean}(rfw))$

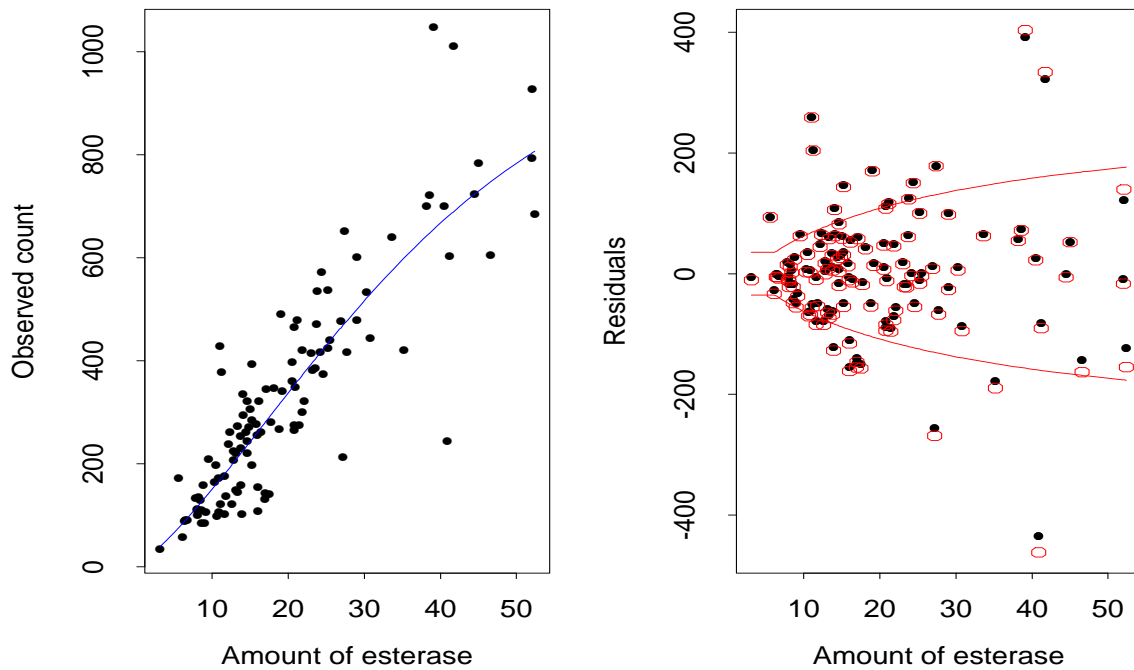


Figure 3: Plot of the best combination of regression and variance model (Calibration)

model: (2.4)  $yr = p1 * (1 - \exp(-p2 * xr))^{p3}$  RSS = 10460 C = 15890

calibration criterion (CC): 3.784

$p1 = 1177.97$   $p2 = 0.029$   $p3 = 1.491$

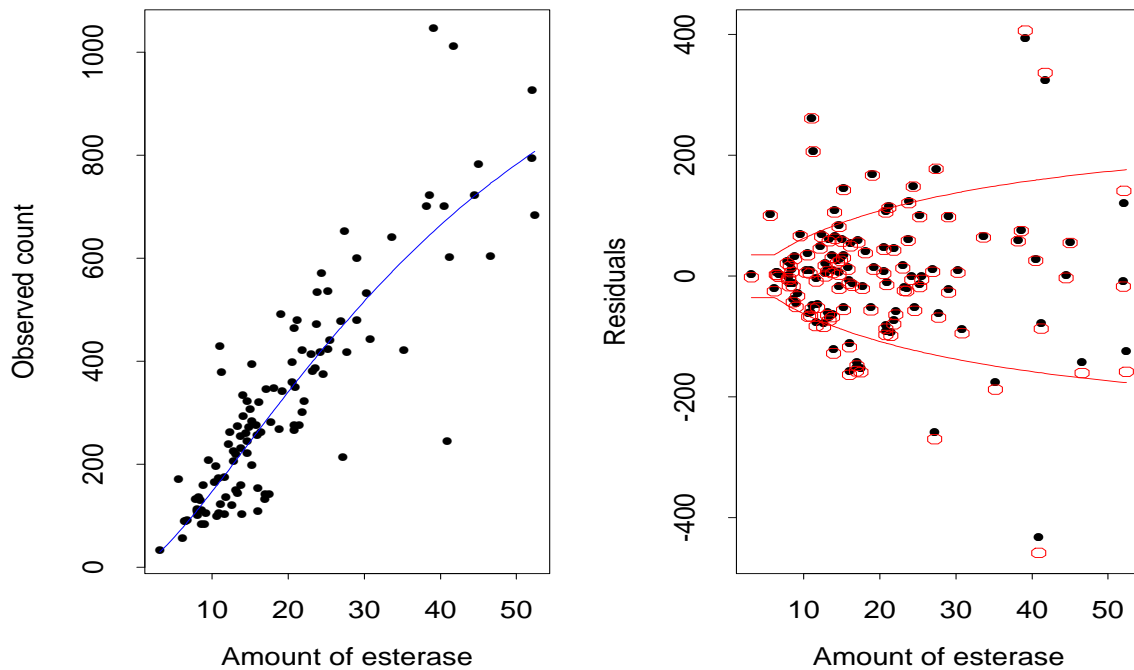


Table 3: Values of the cross-validation criteria  $C$  and  $CC$  for all combinations of a regression model and Box-Cox-transformation.

Regression model	crit.	$\lambda$ value in $T_\lambda$							
		2	1	.5	.25	0	-.5	-1	-2
(2.2)	C	16778	15860	15891	Inf	17020	17557	16901	16733
	CC	3.615	3.835	3.874	Inf	4.21	4.432	4.388	4.388
(2.3)	C	Inf	16058	16258	16240	Inf	Inf	Inf	Inf
	CC	Inf	3.815	3.983	4.092	Inf	Inf	Inf	Inf
(2.4)	C	Inf	15892	15839	Inf	16101	16502	16669	16547
	CC	Inf	3.784	3.875	Inf	4.057	4.262	4.308	4.324
(2.5)	C	22648	16281	16489	16151	16489	16499	16564	16567
	CC	3.678	3.856	4.006	4.067	4.162	4.293	4.292	4.267
(2.6)	C	Inf	16515	17228	21306	43448	20343	18374	17794
	CC	Inf	4.3	4.214	4.121	3.992	4.087	4.143	4.153
(2.7)	C	Inf	15917	15955	Inf	18383	19990	18874	18157
	CC	no explicit inverse model							
(2.8)	C	Inf	15903	15781	16255	17051	16444	16563	16700
	CC	Inf	6.679	7.329	17.6	18.26	5.371	4.53	4.554
(2.9)	C	16994	15913	15876	15891	16136	16496	16647	16671
	CC	<u>3.611</u>	3.786	3.877	3.941	4.056	4.257	4.305	4.322
(2.10)	C	Inf	16083	16267	16377	16629	16862	17015	16978
	CC	Inf	3.821	3.984	4.096	4.182	4.297	4.316	4.305
(2.11)	C	18867	16105	16235	Inf	Inf	Inf	Inf	Inf
	CC	3.643	3.827	3.986	Inf	Inf	Inf	Inf	Inf
(2.12)	C	17500	16278	15884	15809	16110	16523	16611	16968
	CC	4.375	3.877	3.887	3.956	4.086	4.279	4.325	4.425
(2.1)	C	16928	16435	16093	15750	<u>15734</u>	15961	16070	15980
	CC	4.92	4.115	3.967	4.036	4.153	4.345	4.398	4.371

Based on this information the set of regression and variance models and the criterion of interest have to be selected interactively using menus provided by *selectmv*. The function will analyze the specified combinations of regression and variance models providing values of the chosen criterion and graphical displays of the fitted regression function, observations, residuals (shown as points), cross-validated residuals (imaginable as the centres of the octagons) and one-sigma regions based on the truncated variance estimate  $\bar{\sigma}$  with smallest value of criterion (2.25) (see Figure 2 for the best model w.r.t. criterion  $C$  and Figure 3 for the best model w.r.t. criterion  $CC$ ). The returned object *esterase.vcal* is a list with class-attribute "nlsobjlist" ordered by the values of the criterion and containing the information for the single combinations of regression and variance model as components (with class-attribute "nlsobj"). The results

reported in Table 2 for both the cross validation criterion  $C$  for prediction and the calibration criterion  $CC$  clearly indicate, that variance modelling and use of WLSE leads to an improvement if the objective is prediction or estimation of the regression function while for calibration an OLSE should be preferred.

Alternatively we could look for a Box-Cox-transformation and regression model suitable for prediction or calibration. This is done calling:

```
esterase.tpred <- selectmt(esterase, fktlesterase,
                          lambda = c(2,1,.5,0,-.5,-1,-2), graph=T)
esterase.tcal <- selectmt(esterase, fktlesterase,
                          lambda = c(2,1,.5,0,-.5,-1,-2), graph=T)
```

and selecting the class of models and the appropriate criteria from the menus provided. Table 3 contains the corresponding results for both criteria using transformation case **A**. Pairs of  $(m, \lambda)$  for which the minimization algorithm (internal function *da.nls* from Splus) failed to produce reasonable estimates or failed in the cross-validation cycle are indicated by an Inf-value for the criteria in Table 3. Such a failure is notified at calculation time.

We see that for both purposes, calibration and prediction, transformation does a slightly better job than variance modelling, again leading to different models and transformations for both objectives. Notice that model (2.1), which was utilized in Carroll and Ruppert [7] for calibration, performs best for prediction, but with logarithmic transformation of both response and regression function. In the transformation case **B** (parameter *case* = "B" in the call of *selectmt*) and  $\lambda = 0$  with model

$$f(x, \vartheta) = \vartheta_2 \log(x - \vartheta_1) \tag{2.44}$$

leads to the same results. The best pair  $(m, \lambda)$  for calibration is model (2.9) with  $\lambda = 2$ .

### 3. Accuracy statistics, prediction and calibration intervals

Parameter estimates, predictions or calibration values are only of value, if they are complemented by an assessment of their accuracy. Therefore we offer functions to calculate estimates of accuracy characteristics like the standard deviation, the bias or the relative mean absolute errors of an estimator, of a predictor or of a calibration value. Also functions to construct confidence, prediction and calibration intervals are provided. These functions are based on the "moment oriented bootstrap" procedures introduced in Subsection 3.1 .

We assume, that a transformation  $T = T_\lambda$  of the observations and a model  $f_{T,m}$  have been chosen and a corresponding OLSE  $\hat{\vartheta} := \hat{\vartheta}_{T,m}$  is used as a parameter estimate. The accuracy assessments are realized for the estimates, predictions or calibrations based on model  $m$  and transformation  $T$  taken as fixed. The presentation is for the case A (see Subsection 2.3). It covers also the case B, replacing in this case  $f_{T,m}$  by  $f_m$  in all expressions.

The presented bootstrap procedures can also be applied when a model  $f_m$  for the regression function and a variance model are selected and if corresponding WLSE's are used for



the original observations (without any transformation). In this case we have to take  $\lambda = 1$  (identical transformation) and to replace the OLSE  $\hat{\vartheta}_{T,m}$  by the WLSE  $\hat{\vartheta}_m$ .

### 3.1 Bootstrap sampling

The bootstrap is a resampling method that tries to mimic the (unknown) distribution of the observations by generating artificial independent "bootstrap" samples from the fitted model. The estimates are repeatedly calculated for these bootstrap samples. The accuracy of the original estimate is inferred from the empirical distribution of these bootstrap estimates. To generate the bootstrap samples we exploit the following *moment oriented bootstrap procedure*.

We first search for a transformation  $T_*$  and a corresponding model  $f_{T_*,m_*}$  (see (2.31)) with the aim of having a distribution (of the transformed observations  $Y_{ij}^{T_*}$ ) being as near as possible to a normal distribution. This is accomplished by a maximum likelihood choice of  $T_*, m_*$  following Bunke [4]. We have to maximize a "normal likelihood" or equivalently to minimize

$$L(T, m) = n \ln \left[ \sum_{i,j} |Y_{ij}^T - f_{T,m}(x_i, \hat{\vartheta}_{T,m})|^2 \right] + 2n\lambda \ln s - 2(\lambda - 1) \sum_{i,j} \ln(Y_{ij} + \alpha_\lambda) : \quad (3.1)$$

$$L(T_*, m_*) = \min_{T,m} L(T, m) \quad (3.2)$$

We estimate the variances  $\text{Var} Y_{ij}^{T_*}$  using the approach of Subsection 2.2 for the transformed observations  $Y_{ij}^{T_*}$ . Let  $\hat{\sigma}_i^2$  be the estimate of the variance  $\text{Var} Y_{ij}^{T_*}$  provided by the variance model minimizing the criterion (2.25) (with truncation at 0). We then generate  $B$  bootstrap samples

$$Y_{ij}^{T_*,b} := f_{T_*,m_*}(x_i, \hat{\vartheta}_{T_*,m_*}) + \varepsilon_{ij}^b \quad (j = 1, \dots, n_i; \quad i = 1, \dots, k; \quad b = 1, \dots, B), \quad (3.3)$$

where  $\varepsilon_{ij}^b$  are independent realizations of a normal distribution with zero mean and variance  $\tilde{\sigma}_i^2 := \max\{0, \hat{\sigma}_i^2\}$ . It is clear that if  $f_{T_*,m_*}$  is only a rough approximation to the unknown true regression function (in comparison with the variability of the observations), then it makes sense to take the alternative procedure obtained by replacing  $f_{T_*,m_*}(x_i, \hat{\vartheta}_{T_*,m_*})$  by  $\bar{Y}_i^{T_*} = 1/n_i \sum_{j=1}^{n_i} Y_{ij}^{T_*}$ . This would avoid biases due to an incorrect model  $m_*$ .

For each bootstrap sample (3.3) we calculate the corresponding OLSE  $\hat{\vartheta}_{T,m}^b := \hat{\vartheta}^b$  by minimization of

$$S^b(\vartheta) := \sum_{i=1}^k \sum_{j=1}^{n_i} \left( Y_{ij}^{T,b} - f_{T,m}(x_i, \vartheta) \right)^2. \quad (3.4)$$

The bootstrap sampling and the calculation of the OLSE for each bootstrap sample is done by the function *sensitive*. The function uses the information obtained for a single model in combination with a transformation  $(m, \lambda)$  (or  $m$  in combination with a variance model) in the analyses done with function *selectmt* (*selectmv*) and adds information about the bootstrap samples, the OLSE for the bootstrap samples, fitted values, and the variance model utilized to generate the  $\varepsilon_{ij}^b$ .

To provide confidence, prediction and calibration intervals we have to assess the variability of the bootstrap estimates. This is achieved by a double bootstrap procedure applying the technique described above to each bootstrap sample instead of the original sample. This is carried out by the function *dblbootstrap* which adds parameter estimates obtained from the second level (double) bootstrap samples and the parameter estimates for the variance model (obtained from the bootstrap samples and used to generate the second level (double) bootstrap samples) to the information about the model.

For our example this would be performed by the following calls:

```
esterase.tpredbest <- sensitive(esterase.tpred[[1]])
esterase.tpredbest <- dblbootstrap(esterase.tpredbest)
```

to get the bootstrap and double bootstrap statistics for the best model for prediction and

```
esterase.tcalbest <- sensitive(esterase.tcal[[1]])
esterase.tcalbest <- dblbootstrap(esterase.tcalbest)
```

for the best model for calibration purposes.

The model  $f_{T_*, m_*}$  is chosen from the set of models  $f_{T, m}$  specified in the preceding call of function *selectmt*. In our example this leads to  $\lambda_* = .25$  and  $m_*$  as model (2.12).

### 3.2 Accuracy Statistics and Confidence Intervals

The accuracy of an estimate  $\hat{\eta} = \eta(\hat{\vartheta}, f_m(x_1, \hat{\vartheta}), \dots, f_m(x_k, \hat{\vartheta}))$  of a one dimensional parameter  $\eta = \eta(\vartheta, f(x_1), \dots, f(x_k))$  can be assessed based on the estimates computed from the bootstrap samples by the functions *sensitive* and *dblbootstrap*. The variance of  $\hat{\eta}$  is estimated by

$$\widehat{\text{Var}} \hat{\eta} := \frac{1}{B-1} \sum_{b=1}^B (\hat{\eta}^b - \frac{1}{B} \sum_{d=1}^B \hat{\eta}^d)^2, \quad (3.5)$$

where  $\hat{\eta}^b = \eta(\hat{\vartheta}^b, f_m(x_1, \hat{\vartheta}^b), \dots, f_m(x_k, \hat{\vartheta}^b))$ , while the standard deviation of  $\hat{\eta}$  is assessed by

$$\hat{\sigma}_{\hat{\eta}} := \widehat{\text{Var}} \hat{\eta}^{1/2}. \quad (3.6)$$

The bias  $\text{Bias} \hat{\eta} = \mathbf{E}\hat{\eta} - \eta$  is estimated by

$$\widehat{\text{Bias}}(\hat{\eta}) = \frac{1}{B} \sum_{b=1}^B \hat{\eta}^b - \hat{\eta}, \quad (3.7)$$

and the mean absolute relative error  $\text{MARE} = |\eta|^{-1} \mathbf{E}|\hat{\eta} - \eta|$  by

$$\widehat{\text{MARE}}(\hat{\eta}) = \frac{1}{B} \sum_{b=1}^B \frac{|\hat{\eta}^b - \hat{\eta}|}{|\hat{\eta}|}. \quad (3.8)$$

Simple confidence intervals for  $\eta$  of approximate confidence  $(1 - \alpha)$  are given by

$$I_B := [b_{\alpha/2}, b_{1-\alpha/2}], \quad (3.9)$$

where  $b_\alpha$  denotes the  $\alpha$ -quantile of the empirical distribution of the bootstrap estimates  $\hat{\eta}^b$  ( $b = 1, \dots, B$ ). Such quantiles are easily determined, while the following improved confidence intervals require considerably more computation time, being based on a double bootstrap procedure. We estimate the standard error of  $\hat{\eta}$  for each of the bootstrap samples (3.3) using the double bootstrap estimates calculated by function *dblbootstrap* and applying the formula (3.6). We apply these estimates  $\hat{\sigma}_{\hat{\eta}}^b$  in a pivot statistics  $S$ . Let  $a_\alpha$  denote the  $\alpha$ -quantile of the empirical distribution of the pivot values

$$S^b = \frac{\hat{\eta}^b - \hat{\eta}}{\hat{\sigma}_{\hat{\eta}}^b}. \quad (3.10)$$

The improved  $(1 - \alpha)$ -confidence intervals are then given by

$$\tilde{I}_B := [\underline{\eta}, \bar{\eta}] = \left[ \hat{\eta} - a_{1-\alpha/2} \hat{\sigma}_{\hat{\eta}}, \hat{\eta} - a_{\alpha/2} \hat{\sigma}_{\hat{\eta}} \right]. \quad (3.11)$$

The function *confpar* allows to compute estimates of the standard deviation, bias, MARE and simple (if parameter *nb2* = *NULL*) or improved confidence intervals for the parameters involved in the model as well as for functions to compute  $\eta$  supplied either to *selectmv* (*selectmt*) or to *confpar* (using the parameter *fpar*).

Let us reconsider our example. If we are interested in the accuracy of parameter estimates and of the estimate of the linear slope

$$\gamma = \frac{\sum_{i=1}^k f(x_i)(x_i - \bar{x})}{\sum_{i=1}^k (x_i - \bar{x})^2} \quad \text{with} \quad \bar{x} = \frac{1}{k} \sum_{i=1}^k x_i \quad (3.12)$$

corresponding to the best model for prediction the call would be:

```
lslope <- function(x, y, ...)
{
  xmean <- mean(x)
  sum(y * (x - xmean))/sum((x - xmean)^2)
}
attr(lslope, "doc") <- "linear slope"
esterase.tpredbest_confpar(esterase.tpredbest, nb2=20, fpar=lslope)
```

leading to the following results:

	lbound <	value <	ubound	rel. error	bias	st.dev.
p1	-30.051	-23.861	-16.473	0.087843	0.26241	4.1042
p2	15.164	16.730	18.099	0.035471	-0.16574	0.7684

Statistics for *fpar*

	lbound <	value <	ubound	rel. error	bias	st.dev.
	15.164	16.730	18.099	0.035471	-0.16574	0.7684

### 3.3 Prediction and Calibration Intervals

For the construction of prediction and calibration intervals we consider the mean  $\bar{Y}(x)$  of  $l$  uncorrelated (future) values  $Y_j(x)$ ,  $j = 1, \dots, l$ , of the dependent variable for a fixed value  $x$  of the explanatory variable. We estimate the variance of the transformed variable  $T_*(Y_j(x))$  by

$$\tilde{\sigma}^2(x) := \max\{0, g(x, \hat{\sigma}^2, \hat{\tau})\} \quad (3.13)$$

where  $g(x, \sigma^2, \tau)$  is the variance model minimizing the criterion (2.25) as in Subsection 3.1, again based on the transformed observations  $Y_{ij}^{T_*}$ . For the bootstrap procedure we generate independent standard normally distributed values  $u_a^b$  ( $a = 1, \dots, r; b = 1, \dots, B$ ). A simple prediction interval for the (future) mean  $\bar{Y}(x)$ , with approximate coverage probability  $1 - \alpha$ , may be calculated with the  $\alpha$ -quantiles  $c_\alpha$  of the empirical distribution of the pivot values

$$S_a^b = f_{T_*,m}(x, \hat{\vartheta}^b) - \frac{\tilde{\sigma}(x)}{\sqrt{l}} u_a^b \quad (3.14)$$

These  $(1 - \alpha)$ -prediction intervals are:

$$I_P(x) := \left[ T_*^{-1}(c_{\alpha/2}), T_*^{-1}(c_{1-\alpha/2}) \right]. \quad (3.15)$$

Improved intervals are based on a double bootstrap procedure and on the  $\alpha$ -quantiles  $d_\alpha$  of the empirical distribution of the pivot values

$$S_a^b = \frac{f_{T_*,m}(x, \hat{\vartheta}^b) - f_{T_*,m}(x, \hat{\vartheta}) - \frac{\tilde{\sigma}(x)}{\sqrt{l}} u_a^b}{(\tilde{\sigma}^{2b}(x)/l + \hat{\sigma}_m^{2b}(x))^{1/2}}. \quad (3.16)$$

Here  $\tilde{\sigma}^{2b}(x)$  is the truncated estimate of the variance  $\text{Var} T_*(Y(x))$  obtained for the bootstrap sample  $\{Y_{ij}^{T_*,b}\}$  using function *dblbootstrap*. The term  $\hat{\sigma}_m^{2b}(x)$  in (3.16) is the estimate of the variance of the predictor  $\hat{\eta}^b := f_{T_*,m}(x, \hat{\vartheta}^b)$  calculated by double bootstrap.

The improved  $(1 - \alpha)$ -prediction interval is

$$\tilde{I}_P(x) := [\underline{I}_P, \bar{I}_P] = \left[ T_*^{-1}(\underline{f}(x)), T_*^{-1}(\bar{f}(x)) \right], \quad (3.17)$$

where

$$\underline{f}(x) = f_{T_*,m}(x, \hat{\vartheta}) + d_{\alpha/2} \left\{ \tilde{\sigma}^2(x)/l + \hat{\sigma}_m^2(x) \right\}^{1/2}, \quad (3.18)$$

$$\bar{f}(x) = f_{T_*,m}(x, \hat{\vartheta}) + d_{1-\alpha/2} \left\{ \tilde{\sigma}^2(x)/l + \hat{\sigma}_m^2(x) \right\}^{1/2}, \quad (3.19)$$

and where  $\hat{\sigma}_m^2(x)$  is the estimate of the variance of the predictor  $\hat{\eta} := f_{T_*,m}(x, \hat{\vartheta})$  obtained from the bootstrap estimates  $\hat{\vartheta}^b$ .

The calculation of prediction intervals can be performed using the function *predict.nlsobj*. For our example improved bootstrap prediction intervals for the mean of 4 replications of the response at points  $x = (10, 20, 30, 40, 50)$  are calculated by:

```
esterase.tpredbest <- predict(esterase.tpredbest, nb2=20,
                             additional = seq(10,50,10), replications = 4)
```

The results are presented as

Pointwise prediction and confidence intervals

x	lboud(y)	< lboud(f)	< f(x)	< ubound(f)	< ubound(y)
10	97.39	132.00	143.44	152.83	205.74
20	220.89	282.39	310.73	332.95	420.09
30	352.84	433.49	478.03	515.89	621.78
40	491.53	583.97	645.33	697.42	813.97
50	639.65	734.47	812.63	878.65	1003.70

and added to the object `esterase.tpredbest`.

For calibration intervals we use the calibration values defined by the inverse  $f_m^{-1}(\cdot, \vartheta)$  of the regression function  $f_m(\cdot, \vartheta)$ . That is the calibration value corresponding to the mean  $y := \bar{Y}(x)$  of  $l$  uncorrelated observations  $Y_j(x)$  of the dependent variable (for an unknown fixed value of the explanatory variable) would be:

$$\hat{x} = f_m^{-1}(y, \hat{\vartheta}). \quad (3.20)$$

Assuming that  $f_m(x, \vartheta)$  is monotonously increasing in  $x$ , a simple  $(1 - \alpha)$ -calibration interval is given by

$$I_C(y) := \left[ f_{T_*,m}^{-1}(g_{\alpha/2}, \hat{\vartheta}), f_{T_*,m}^{-1}(g_{1-\alpha/2}, \hat{\vartheta}) \right], \quad (3.21)$$

where  $g_\alpha$  denotes the  $\alpha$ -quantile of the empirical distribution of the pivot values

$$S_a^b = f_{T_*,m}(\hat{x}, \hat{\vartheta}^b) - \frac{\tilde{\sigma}(\hat{x})}{\sqrt{l}} u_a^b. \quad (3.22)$$

Improved  $(1 - \alpha)$ -calibration intervals are given by a double bootstrap procedure, namely by

$$\begin{aligned} \tilde{I}_C(y) := [\underline{I}_C, \bar{I}_C] &= \left\{ x : T_*(y) + \tilde{h}_{\alpha/2} \{ \tilde{\sigma}^2(\hat{x})/l + \hat{\sigma}_m^2(\hat{x}) \}^{1/2} < \right. \\ &< f_{T_*,m}(x, \hat{\vartheta}) < T_*(y) + \tilde{h}_{1-\alpha/2} \{ \tilde{\sigma}^2(\hat{x})/l + \hat{\sigma}_m^2(\hat{x}) \}^{1/2} \left. \right\}. \end{aligned} \quad (3.23)$$

Here  $h_\alpha$  denotes the  $\alpha$ -quantiles of the empirical distribution of the pivot values

$$S_a^b = \frac{f_{T_*,m}(\hat{x}, \hat{\vartheta}^b) - T_*(y) - \frac{\tilde{\sigma}(\hat{x})}{\sqrt{l}} u_a^b}{\{ \tilde{\sigma}^{2b}(\hat{x}_a^b)/l + \hat{\sigma}_m^{2b}(\hat{x}_a^b) \}^{1/2}}, \quad (3.24)$$

where  $\hat{x}_a^b = f_{T_*,m}^{-1}(T_*(y) + \frac{\tilde{\sigma}(\hat{x})}{\sqrt{l}} u_a^b, \hat{\vartheta}^b)$ .

Calibration intervals and calibration values can be obtained using the function `calibrate`. Confidence intervals for values of the inverse of the regression function are defined by (3.21 and 3.23) for  $l = \infty$ . These intervals are calculated by function `calibrate` if a value  $< 1$  is specified for the parameter `replications`. For our example improved bootstrap confidence intervals for the inverse of the regression function  $f^{-1}(y)$  at the eight values  $y = (50, 150, 250, 350, 450, 550, 650, 750)$  are calculated by:

```
esterase.tcalbest <- calibrate(esterase.tcalbest, nb2 = 20,
  additional = seq(50,750,100), replications = 0)
```

The results are presented in the form

Calibration intervals

y	lbound(y)	< finv(y)	< ubound(y)
50	3.1000	5.8231	10.177
150	7.0351	11.1010	13.348
250	12.2530	15.5170	17.602
350	17.4260	19.8940	22.537
450	21.6380	24.6010	27.046
550	26.0870	29.9890	31.556
650	33.5700	36.5580	36.924
750	42.7070	45.2140	45.710

and added to the object *esterase.tcalbest*.

#### 4. Presentation of Results

The results computed at any stage of the analysis are contained in objects of class *"fktlist"* (for the description of the class of regression models used in the analysis), *"nlsobj"* (for results corresponding to a single model) and *"nlsobjlist"* for lists consisting of components of class *"nlsobj"* (created by functions *selectmv* and *selectmt*). We provide generic functions to print and plot the objects obtained during the analyses. In both cases the information contained in the object is analyzed and printed or displayed in an appropriate way. We illustrate this using our example.

```
print(esterase.tcalbest, graph=T)
```

creates the plot shown in Figure 4. The calibration intervals are displayed as horizontal bars in the left hand panel together with the observations and the fitted curve. The information to print (contained in the object *esterase.tcalbest*), i.e. model description (including transformation), data, parameter estimates, values of the criteria computed and calibration intervals, can be selected interactively using menus and is provided in standardized form.

```
plot(esterase.tpredbest, transformed=T, cvres=T, additional=T)
```

will create the graphical display, shown in Figure 5, containing the main information obtained in the analysis of the best model for prediction. Model description, parameter estimates and computed criteria are given as headings of the plots. The left hand panel shows a plot (under transformation) of the data, fitted curve and prediction intervals (vertical bars) at the design points provided in the call of *predict*. The right hand panel shows a plot of both residuals and cross-validated residuals under transformation (visualized as in Figures 2 and 3). Prediction intervals are shown as vertical bars. If a best variance model with respect to criterion (2.25) was selected, pointwise one-sigma regions (like those in Figures 2 and 3) based on this variance model are displayed together with the residuals.

Figure 4: Plot generated by `print(esterase.tcalbest, graph=T)`.

model: (2.9)  $yr = 1/(p1 + p2 * xr^p3)$  case= A lambda= 2 RSS = 10840 C = 16990  
 calibration criterion (CC): 3.611  
 p1 = 0.001 p2 = 0.504 p3 = -1.858

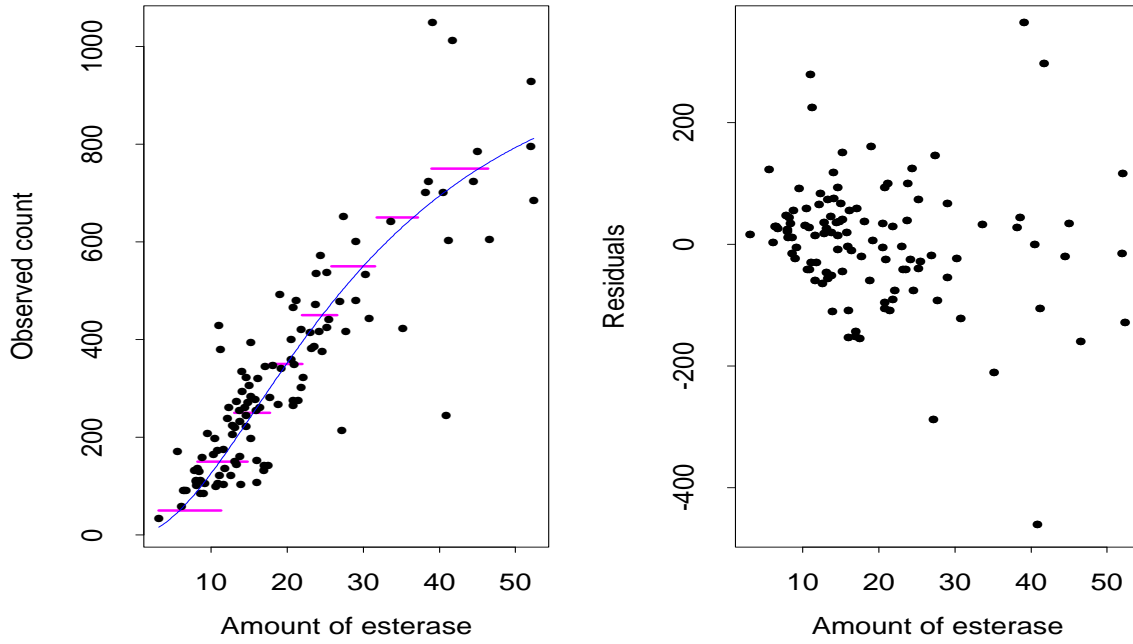
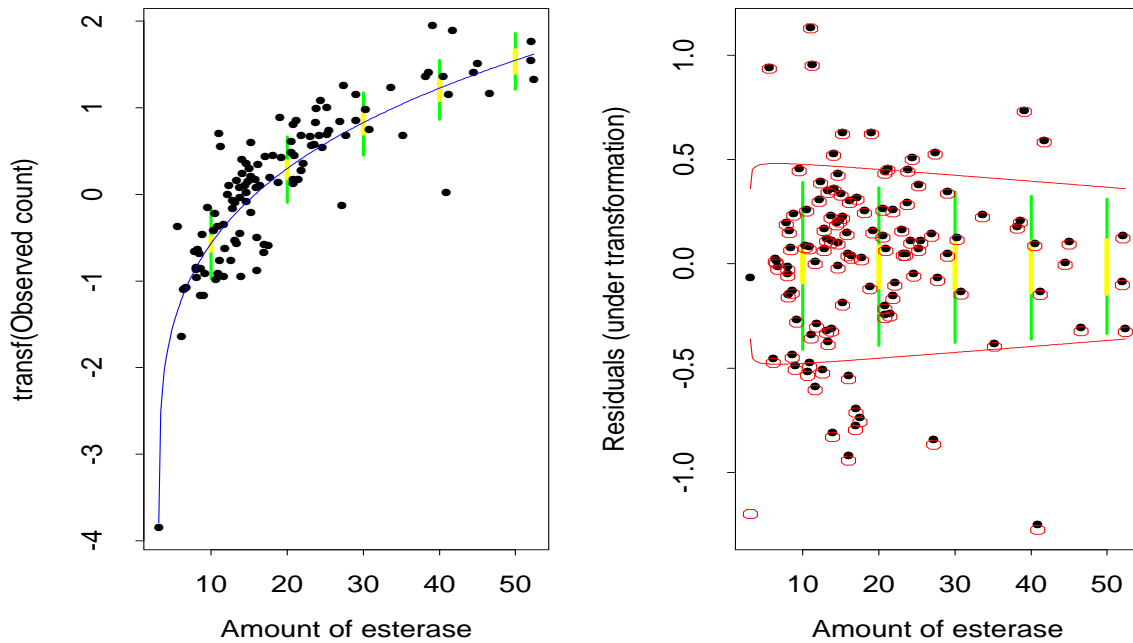


Figure 5: Plot generated by `plot(esterase.tpredbest, transformed=T, cvres=T, additional=T)`

model: linear (2.1)  $yr = p1 + p2 * xr$  case= A lambda= 0 RSS = 11050 C = 15730  
 calibration criterion (CC): 4.153  
 p1 = -23.861 p2 = 16.73



```
print(esterase.vpred)
```

creates an overview about the information contained in the list *esterase.vpred*, giving the values of the criteria computed for the models analyzed in the corresponding call of function *selectmv*, while

```
plot(esterase.vpred)
```

successively displays plots for the analyzed models.



## A Appendix

calibrate
-----------

### DESCRIPTION

Computes bootstrap calibration intervals in a nonlinear regression model with transformation or variance modelling

### USAGE

```
calibrate(model, nb=200, nb2=NULL, nb3=-1, design=F, additional=NULL,
          coverage = 0.9, replications=1, graph=F, display=NULL, useymean=F)
```

### REQUIRED ARGUMENTS

*model* object of class "nlsobj" (component of the list returned by function *selectmv* or *selectmt*) containing a description of a regression model (with variance model or transformation) and corresponding estimates.

### OPTIONAL ARGUMENTS

*nb* number of bootstrap samples

*nb2* number of samples in second level bootstrap. If *NULL* double bootstrap is not performed and only the simple bootstrap intervals are computed (leading to asymptotically less efficient but less time consuming statistics).

*nb3* number of independent standard normally distributed values  $u_a^b$  (for each bootstrap sample *b*) used in the definition of the bootstrap calibration intervals. If *nb3* < 1 the sample size is used instead of *nb3*.

*design* : logical flag: if *TRUE*, calibration intervals are calculated corresponding to the values of the response  $f(x_i, \theta)$  ( $x_i$  design points in the region of interest  $model\$A \leq f(x_i, \theta) \leq model\$B$  of the calibration criteria CC).

*additional* additional vector of responses for which calibration values and intervals are calculated.

*coverage* coverage probability of the calibration intervals

*replications* number of replications; responses given in parameter *additional* or used if *design* = *T* are considered as a mean of "*replications*" observations.

If *replications* < 1; responses given in parameter *additional* or used if *design* = *T* are considered as obtained without error (as a mean of infinitely many observations).

*graph* logical flag: if *TRUE*, the results are plotted.

*display* display to be used ("display hostname:0.0" , be sure to allow access before).

*useymean* logical flag: if *FALSE*, fitted values from the reference model are used to generate bootstrap samples; if *TRUE*, bootstrap samples are generated based on the vector of means of the observed responses at the design points (instead of fitted values from the reference model).

## VALUE

The parameter model is returned with the following components added (or updated):

*coveragec* coverage probability of the calculated calibration intervals.

*replicationsc* value of parameter replications

*ycal* fitted values at design points (larger than *model\$A* and smaller than *model\$B*, used if *design = T*).

*xhat* calibration values (values of inverse regression function) for *ycal*.

*lowcalconfp* lower bounds of calibration intervals for *ycal*.

*upcalconfp* upper bounds of calibration intervals for *ycal*.

*additionalcal* vector provided by parameter *additional*.

*xcaladd* calibration values (values of inverse regression function) for *additionalcal*.

*lowcaladdp* lower bounds of calibration intervals for *additionalcal*.

*upcaladdp* upper bounds of calibration intervals for *additionalcal*.

If the function *sensitive* is called by the function *calibrate*, the following components are added:

*tvartyp* number of variance model used to generate bootstrap samples (selected by cross-validation; in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*tvarpar* parameters of variance model used to generate bootstrap samples (calculated in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*trefvar* homogeneous variance estimate (under transformation if  $\lambda \neq 1$ ).

*tvcv* vector of values obtained for cross-validation criterion for variance estimation (for all variance models; calculated in the transformed space if  $\lambda \neq 1$ ).

*bootstrap* matrix of parameter estimates obtained from the bootstrap samples.

*yboot* bootstrap samples

*fwboot* fitted values obtained from the bootstrap samples

*useymean* logical flag given by parameter *useymean*

If the function *dblbootstrap* is called by the function *calibrate*, the following components are added:

*double* array of parameter estimates obtained from second level bootstrap samples (double bootstrap).

*bsvarpar* matrix of parameters of the variance model used to generate second level bootstrap samples; obtained from the bootstrap samples.

*bsrefvar* vector of homogeneous variance estimates obtained from the bootstrap samples.

## SIDE EFFECTS

If the object given by parameter model does not contain appropriate information about the bootstrap samples and estimates (bootstrap estimates not computed in advance or number of bootstrap samples less than *nb* or *model\$useymean*  $\neq$  *useymean*) the function *sensitive* is called to (re)calculate the bootstrap estimates. If *nb2*  $\neq$  *NULL* and *model* does not contain appropriate information about double bootstrap samples and estimates (double bootstrap estimates not computed in advance or number of double bootstrap samples less than *nb2* or the function *sensitive* has been called by *calibrate*) the function *dblbootstrap* is called to (re)do the second level bootstrap.

## DETAILS

The function computes calibrations and simple (*nb2* == *NULL*) or improved bootstrap calibration intervals based on the results obtained in the model selection process using the functions *selectmv* or *selectmt*.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

*selectmv*, *selectmt*, *nlsobj*, *predict.nlsobj*, *confpar*, *nlsobjlist*, *plot.nlsobj*, *print.nlsobj*, *plot.nlsobjlist*, *print.nlsobjlist*

## EXAMPLES

```
# generate the data of example 2
x <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0.02, 0.02, 0.02, 0.02,
      0.04, 0.04, 0.04, 0.04, 0.06, 0.06, 0.06, 0.06,
      0.08, 0.08, 0.08, 0.08, 0.1, 0.1, 0.1, 0.1, 0.2,
      0.2, 0.2, 0.2, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6, 0.6,
      0.6, 0.8, 0.8, 0.8, 0.8, 1, 1, 1, 1, 1.5, 1.5, 1.5,
      1.5, 2, 2, 2, 2, 4, 4, 4, 4, Inf, Inf, Inf, Inf)
logx <- log(x,10)
logx[logx == -Inf] <- -3
logx[logx == Inf] <- 2
# use -3 and 2 as codes for -Inf and Inf
```

```

y <- c(2868, 2785, 2849, 2805, 2779, 2588, 2701, 2752,
      2615, 2651, 2506, 2498, 2474, 2573, 2378, 2494,
      2152, 2307, 2101, 2216, 2114, 2052, 2016, 2030,
      1862, 1935, 1800, 1871, 1364, 1412, 1377 1304,
      910, 919, 855, 875, 702, 701, 689, 696, 586,
      596, 561, 562, 501, 495, 478, 493, 392, 358,
      399, 394, 330, 351, 343, 333, 250, 261, 244,
      242, 131, 135, 134, 133)
example2 <- list( x = logx, y = y, xname = "log-dose",
                 yname = "response")

# regression models, from catalog used in example 2
# indicator functions are used to identify -3 as -Inf
# and 2 as Inf
model1 <- yr ~ p1 * exp( - exp(p2 - p3 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model2 <- yr ~ p1 + p2/(1 + exp(p3 - p4 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p2
model3 <- yr ~ p1 + p2/(1 + exp(p3 - p4 * xr))^p5 *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p2
model4 <- yr ~ p1/(1 + exp(p2 - p3 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model5 <- yr ~ p1/(1 + exp(p2 - p3 * xr))^(1/p4) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model6 <- yr ~ p1 + p2 * exp( - exp(p3 - p4 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(2 <= xr) * p2

# inverse models (to be used for calibration)
invmod1 <- xr ~ (p2 - log(log(p1/yr)))/p3
invmod2 <- xr ~ (p3 - log(p2/(yr - p1) - 1))/p4
invmod3 <- xr ~ (p3 - log((p2/(yr - p1))^(1/p5) - 1))/p4
invmod4 <- xr ~ (p2 - log(p1/yr - 1))/p3
invmod5 <- xr ~ (p2 - log((p1/yr)^p4 - 1))/p3
invmod6 <- xr ~ (p3 - log(log(p2/(yr - p1))))/p4

# now create the list of models
model1 <- list(fkt = model1[[3]], inv = invmod1[[3]],
              p = 3, name = "Gompertz")
model2 <- list(fkt = model2[[3]], inv = invmod2[[3]],
              p = 4, name = "Logistic")
model3 <- list(fkt = model3[[3]], inv = invmod3[[3]],
              p = 5, name = "Richards")
model4 <- list(fkt = model4[[3]], inv = invmod4[[3]],
              p = 3, name = "(1.36)")

```

```

model5 <- list(fkt = model5[[3]], inv = invmod5[[3]],
              p = 4, name = "(1.42)")
model6 <- list(fkt = model6[[3]], inv = invmod6[[3]],
              p = 4, name = "(1.43)")
fktlistex2 <- list(model1, model2, model3, model4,
                  model5, model6)

# all other models from the catalog given in the
# references are to restrictive
fktlistex2 <- parinit( example2, fktlistex2, graph=T)
# now select a combination of regression and
# Box-Cox-Transformation
example2.tresults <- selectmt(example2, fktlistex2as,
                             lambda=c(-2,-1,-.5,0,.5,1,2), a=-2.99,b=1.99,A=200,B=2500,graph=T)
# select all models and the calibration criteria CC
# from the menu displayed
#
# now compute calibration intervals for the values of
# the response given in the argument additional
example2.besttcal <- calibrate(example2.tresults[[1]],nb2=20,
                              design=F, additional=seq(200,2700,500))
# display the results for the best model
plot(example2.besttcal,additional=T)
# plot results in the transformed space
plot(example2.besttcal, additional=T, transformed=T)

```

confpar
---------

## DESCRIPTION

Compute accuracy statistics and confidence intervals for the parameters in a nonlinear regression model.

## USAGE

```
confpar(model, nb=200, nb2=NULL, coverage=.9, fpar=NULL, useymean=F)
```

## REQUIRED ARGUMENTS

*model* object of class "*nlsobj*" (component of the list returned by function *selectmv* or *selectmt*) containing a description of a regression model (with variance model or transformation) and corresponding estimates.

## OPTIONAL ARGUMENTS

*nb* number of bootstrap samples

*nb2* number of samples in second level bootstrap. If *NULL* double bootstrap is not performed and only the simple bootstrap intervals are computed (leading to asymptotically less efficient but less time consuming statistics).

*coverage* coverage probability of the confidence intervals.

*fpar* function with arguments *x* (design), *fw* (values of the regression function at (*design*, *par*)), *par* (parameters); used to calculate a parameter of interest.

*useymean* logical flag: if *FALSE*, fitted values from the reference model are used to generate bootstrap samples; if *TRUE*, bootstrap samples are generated based on the vector of means of the observed responses at the design points (instead of fitted values from the reference model).

## VALUE

The parameter model is returned with the following components added (or updated):

*coverage* coverage probability of confidence intervals calculated by *confpar*.

*parstats* confidence intervals, estimates of relative absolute error, bias and standard deviation of the parameter estimates.

*fpar* function provided in parameter *fpar*.

*fparvalue* estimate of  $fpar(x, fw, par)$

*fparstats* confidence intervals, estimates of mean absolute relative error, bias and standard deviation of the estimated function  $fpar(x, fw, par)$ .

If the function *sensitive* is called by the function *confpar*, the following components are added:

*tvartyp* number of variance model used to generate bootstrap samples (selected by cross-validation; in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*tvarpar* parameters of variance model used to generate bootstrap samples (calculated in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*trefvar* homogeneous variance estimate (under transformation if  $\lambda \neq 1$ ).

*tvcv* vector of values obtained for cross-validation criterion for variance estimation (for all variance models; calculated in the transformed space if  $\lambda \neq 1$ ).

*bootstrap* matrix of parameter estimates obtained from the bootstrap samples.

*yboot* bootstrap samples

*fwboot* fitted values obtained from the bootstrap samples

*useymean* logical flag given by parameter *useymean*

If the function *dblbootstrap* is called by the function *confpar*, the following components are added:

*double* array of parameter estimates obtained from second level bootstrap samples (double bootstrap).

*bsvarpar* matrix of parameters of the variance model used to generate second level bootstrap samples; obtained from the bootstrap samples.

*bsrefvar* vector of homogeneous variance estimates obtained from the bootstrap samples.

## SIDE EFFECTS

If the object given by parameter model does not contain appropriate information about the bootstrap samples and estimates (bootstrap estimates not computed in advance or number of bootstrap samples less than *nb* or *model\$useymean*  $\neq$  *useymean*) the function *sensitive* is called to (re)calculate the bootstrap estimates. If *nb2*  $\neq$  *NULL* and model does not contain appropriate information about double bootstrap samples and estimates (double bootstrap estimates not computed in advance or number of double bootstrap samples less than *nb2* or the function *sensitive* has been called by *confpar*) the function *dblbootstrap* is called to (re)do the second level bootstrap.

## DETAILS

The function computes bootstrap confidence intervals, bootstrap estimates of mean absolute relative error, bias and standard deviation of estimates of the parameters in the model and of parameters calculated by the function *fpar* and the optional parameter *fg* in the call of *selectmv* or *selectmt*. The analysis is based on the results obtained in the model selection process using the functions *selectmv* or *selectmt*.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

*selectmv*, *selectmt*, *nlsoj*, *predict.nlsobj*, *calibrate*, *nlsojlist*, *plot.nlsobj*, *print.nlsobj*, *plot.nlsobjlist*, *print.nlsobjlist*

## EXAMPLES

```
# generate the data of example 2
x <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0.02, 0.02, 0.02, 0.02,
      0.04, 0.04, 0.04, 0.04, 0.06, 0.06, 0.06, 0.06,
      0.08, 0.08, 0.08, 0.08, 0.1, 0.1, 0.1, 0.1, 0.2,
      0.2, 0.2, 0.2, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6, 0.6,
```

```

0.6, 0.8, 0.8, 0.8, 0.8, 1, 1, 1, 1, 1.5, 1.5, 1.5,
1.5, 2, 2, 2, 2, 4, 4, 4, 4, Inf, Inf, Inf, Inf)
logx <- log(x,10)
logx[logx == -Inf] <- -3
logx[logx == Inf] <- 2
# use -3 and 2 as codes for -Inf and Inf
y <- c(2868, 2785, 2849, 2805, 2779, 2588, 2701, 2752,
2615, 2651, 2506, 2498, 2474, 2573, 2378, 2494,
2152, 2307, 2101, 2216, 2114, 2052, 2016, 2030,
1862, 1935, 1800, 1871, 1364, 1412, 1377 1304,
910, 919, 855, 875, 702, 701, 689, 696, 586,
596, 561, 562, 501, 495, 478, 493, 392, 358,
399, 394, 330, 351, 343, 333, 250, 261, 244,
242, 131, 135, 134, 133)
example2 <- list( x = logx, y = y, xname = "log-dose",
yname = "response")
# regression models, from catalog used in example 2
# indicator functions are used to identify -3 as -Inf
# and 2 as Inf
model1 <- yr ~ p1 * exp( - exp(p2 - p3 * xr)) *
I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model2 <- yr ~ p1 + p2/(1 + exp(p3 - p4 * xr)) *
I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p2
model3 <- yr ~ p1 + p2/(1 + exp(p3 - p4 * xr))^p5 *
I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p2
model4 <- yr ~ p1/(1 + exp(p2 - p3 * xr)) *
I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model5 <- yr ~ p1/(1 + exp(p2 - p3 * xr))^(1/p4) *
I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model6 <- yr ~ p1 + p2 * exp( - exp(p3 - p4 * xr)) *
I((-3 < xr) & (xr < 2)) + I(2 <= xr) * p2
# inverse models (to be used for calibration)
invmod1 <- xr ~ (p2 - log(log(p1/yr)))/p3
invmod2 <- xr ~ (p3 - log(p2/(yr - p1) - 1))/p4
invmod3 <- xr ~ (p3 - log((p2/(yr - p1))^(1/p5) - 1))/p4
invmod4 <- xr ~ (p2 - log(p1/yr - 1))/p3
invmod5 <- xr ~ (p2 - log((p1/yr)^p4 - 1))/p3
invmod6 <- xr ~ (p3 - log(log(p2/(yr - p1))))/p4
# now create the list of models
model1 <- list(fkt = model1[[3]], inv = invmod1[[3]],
p = 3, name = "Gompertz")

```



```

model2 <- list(fkt = model2[[3]], inv = invmod2[[3]],
              p = 4, name = "Logistic")
model3 <- list(fkt = model3[[3]], inv = invmod3[[3]],
              p = 5, name = "Richards")
model4 <- list(fkt = model4[[3]], inv = invmod4[[3]],
              p = 3, name = "(1.36)")
model5 <- list(fkt = model5[[3]], inv = invmod5[[3]],
              p = 4, name = "(1.42)")
model6 <- list(fkt = model6[[3]], inv = invmod6[[3]],
              p = 4, name = "(1.43)")
fktlistex2 <- list(model1, model2, model3, model4,
                  model5, model6)
# all other models from the catalog given in the
# references are to restrictive
fktlistex2 <- parinit( example2, fktlistex2, graph=T)
fgammaint <- function(x, y, ...)
# function to approximate the area under the curve
{
    k <- length(x)
    sum((y[-1] + y[ - k]) * diff(10^x))/2
}
# now select a combination of regression and
# variance model
example2.vresults <- selectmv(example2, fktlistex2as,
                             a=-2.99, b=1.99, fg=fgammaint, graph=T)
# select all models, all variance models and the parameter
# estimation criteria CG from the menu displayed
# now compute accuracy statistics for model parameters and
# fgammaint(...) for the best model
example2.bestvpar <- confpar(example2.vresults[[1]],nb2=20)
# display the results for the best model
print(example2.bestvpar,graph=T)

```

dblbootstrap
--------------

#### DESCRIPTION

Compute second level (double) bootstrap estimates for a nonlinear regression model

#### USAGE

dblbootstrap(model, nb2=20)

## REQUIRED ARGUMENTS

*model* object of class "*nlsobj*" (object returned by function *sensitive*) containing a description of a regression model (with variance model or transformation) and corresponding estimates (including first level bootstrap estimates).

## OPTIONAL ARGUMENTS

*nb2* number of samples in second level bootstrap.

## VALUE

The parameter model is returned with the following components added (or updated):

*double* array of parameter estimates obtained from second level bootstrap samples (double bootstrap).

*bsvarpar* matrix of estimated parameters of the variance model used to generate second level bootstrap samples ( obtained from bootstrap samples).

*bsrefvar* vector of homogeneous variance estimates obtained from the bootstrap samples.

## SIDE EFFECTS

The function assumes to find information added by function *sensitive*. If there was no preceding call of function *sensitive* this is notified and the parameter model is returned unchanged.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

*sensitive*, *selectmv*, *selectmt*, *nlsobj*, *predict.nlsobj*, *calibrate*, *confpar*, *nlsobjlist*, *plot.nlsobj*, *print.nlsobj*, *plot.nlsobjlist*, *print.nlsobjlist*

nlsobj
--------

## DESCRIPTION

This is an object of class "*nlsobj*". The model choice functions *selectmt* (selection of models with transformation) and *selectmv* (selection of models with variance estimation) create a list of class "*nlsobjlist*". The components of this list are object of class "*nlsobj*" containing the results calculated for a single model. The functions *sensitive* (first

level bootstrap), *dblbootstrap* (second level (double) bootstrap), *confpar* (accuracy statistics and confidence intervals of parameters), *predict.nlsobj* (prediction and confidence intervals) and *calibrate* (calibration intervals) use the information about a single model contained in this object and add their results as additional components. The object is a list consisting of the following components:

## STRUCTURE

Model description (returned from *selectmv* or *selectmt*)

*name* name of the regression model

*formula* the formula used to evaluate the model.

*invfkt* the formula used to evaluate the calibrations (if calibration is performed).

*fgamma* the formula used to evaluate the parameter function in criterion CG (if criterion CG is used).

*a* lower bound of  $x$  in the definition of the cross-validation criteria.

*b* upper bound of  $x$  in the definition of the cross-validation criteria.

*A* lower bound of  $y$  in the definition of the calibration criterion (if calibration was performed).

*B* upper bound of  $y$  in the definition of the calibration criterion (if calibration was performed).

*case* transformation case used in function *selectmt*; "A" if transformation of both response variable and regression function, or "B" if only transformation of the response variable (see references).

*lambda* exponent  $\lambda$  used in Box-Cox-transformation (if transformation is performed).

*nr* position of the model in the list *fktl* provided to *selectmt* or *selectmv*.

Description of data returned by *selectmv* or *selectmt*

*x* values of the explanatory variable (ordered with replications).

*y* values of the response variable (ordered according to  $x$ ).

*xname* label to be used for explanatory variable.

*yname* label to be used for response variable.

*xk* vector of ordered design points (without replications).

*ni* vector of number of replications at the design points  $xk$ .

*xm* vector of first positions of components of  $xk$  in vector  $x$  ( $xk == x[xm]$ ).

*k* number of different design points.

*xo* vector of positions of components of  $x$  in vector  $xk$  ( $x == xk[xo]$ ).

*yq* vector of means of observed responses at design points  $xk$ .

Results returned from *selectmv* or *selectmt*

*res* vector of residuals.

*par* parameter estimates.

*fw* vector of fitted values.

*alpha* shift parameter used in Box-Cox-transformation (if transformation is performed).

*scale* scale parameter used in Box-Cox-transformation (if transformation is performed).

*rss* residual sum of squares.

*rssl* residual sum of squares under transformation (*case* = "A" and *lambda*  $\neq$  1).

*rescv* vector of cross-validated residuals (for  $a \leq x \leq b$  if cross-validation was performed).

*parcv* matrix of parameter estimates from cross-validation steps performed.

*cv* value of the cross-validation Criteria C or C-tilde (if cross-validation was performed).

*xcal* vector of calibration values (values of inverse regression function for observations *y* of the response variable; if calibration was performed).

*xcalcv* vector of cross-validated calibration values (for  $a \leq x \leq b$ ,  $A \leq y \leq B$ ; if calibration was performed).

*cvcal* value of the calibration criterion CC (if calibration was performed).

*rescal* calibration residuals.

*gvalue* estimated value of the parameter function *fgamma*.

*cvpar* value of the criterion CG (if estimation of a parameter function was performed).

*refmodstar* formula used to evaluate the reference model  $m_*$  (best fitting model) in the calculation of the residuals  $e_i$  in the variance estimation procedure.

*refparstar* parameter estimates used within *refmodstar*.

*reflambdastar* lambda value of optimal transformation  $T_*$  (minimizing the expression (3.1)) (if transformation was performed) .

*ccode* return code from *do\_nls*.

*ccodecv* maximal return code from *do\_nls* in the cross-validation steps (if cross validation was performed).

Description of the variance estimation (returned from *selectmv*)

*vartyp* number of variance model used in WLSE.

*varpar* vector of estimated parameters of variance model used in WLSE.

*bestvartyp* number of best variance model according to the cross-validation criterion for variance estimation.

*bestvarpar* vector of estimated parameters of the best variance model according to the cross-validation criterion for variance estimation.

*refmod* formula for evaluating the reference model used to calculate the functional values employed as arguments in the variance models.

*refpar* parameter estimates used within *refmod*.

*refvar* homogeneous variance estimate.

*vcv* vector of values obtained for cross-validation criterion for variance estimation. *Inf* values correspond to variance models excluded from the analysis using the parameter *vmodels* of function *selectmv*.

Components added by function *sensitive*

*tvartyp* number of variance model used to generate bootstrap samples (selected by cross-validation; in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*tvarpar* parameters of variance model used to generate bootstrap samples (calculated in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*trefvar* homogeneous variance estimate (under transformation if  $\lambda \neq 1$ ).

*tcvcv* vector of values obtained for cross-validation criterion for variance estimation (for all variance models; calculated in the transformed space if  $\lambda \neq 1$ ). *Inf* values correspond to variance models excluded from the analysis using the parameter *vmodels* of function *sensitive*.

*bootstrap* matrix of parameter estimates obtained from the bootstrap samples.

*bsrefpar* matrix of parameter estimates for the reference model  $m_0$  obtained from the bootstrap samples. Computed in case of variance modelling. If transformation is used the reference model  $m_0$  is assumed to coincide with the model under consideration, i.e. *bsrefpar* would duplicate the information contained in *bootstrap*.

*yboot* matrix of bootstrap samples

*fwboot* matrix of fitted values obtained from the bootstrap samples.

*alphastar* shift parameter used in Box-Cox-transformation with  $\lambda_*$

*alphastarb* vector of shift parameters used for bootstrap samples in Box-Cox-transformations with  $\lambda_*$

*alphab* vector of shift parameters used for bootstrap samples in Box-Cox-transformations with  $\lambda$

*scaleb* vector of scale parameters used for bootstrap samples in Box-Cox-transformations (independent of lambda)

*useymean* logical flag: if *FALSE*, fitted values from the reference model were used to generate bootstrap samples; if *TRUE*, bootstrap samples were generated based on the vector of means of the observed responses at the design points (instead of fitted values from the reference model).

Components added by function *dblbootstrap*

*double* array of parameter estimates obtained from second level bootstrap samples (double bootstrap).

*bsvarpar* matrix of estimated parameters of the variance model used to generate second level bootstrap samples ( obtained from bootstrap samples).

*bsrefvar* vector of homogeneous variance estimates obtained from the bootstrap samples.

Components added by function *predict.nlsobj*

*coveragep* coverage probability of the prediction intervals and confidence intervals for values of the regression function.

*replicationsp* number of replications; prediction intervals were computed for the mean of "replicationsp" future values of the response variable at a given design point.

*additional* additional vector of design points for which prediction and confidence intervals are calculated.

*lowconf* vector of lower confidence bounds for values of the regression function at the design points *xk*.

*upconf* vector of upper confidence bounds for values of the regression function at the design points *xk*.

*lowconfp* vector of lower prediction bounds at the design points *xk*.

*upconfp* vector of upper prediction bounds at the design points *xk*.

*lowadd* vector of lower confidence bounds for values of the regression function at the additional design points.

*upadd* vector of upper confidence bounds for values of the regression function at the additional design points.

*fwadd* vector of fitted values at the additional design points.

*lowaddp* vector of lower prediction bounds at the additional design points.

*upaddp* vector of upper prediction bounds at the additional design points.

Components added by function *calibrate*

*coveragec* coverage probability of the calculated calibration intervals.

*replicationsc* number of replications; responses used for calibration are considered as a mean of "replications" observations.

*ycal* vector of fitted values at design points (larger than *A* and smaller than *B*).

*xhat* vector of calibration values (values of inverse regression function) for *ycal*.

*lowcalconfp* vector of lower bounds of calibration intervals for *ycal*.

*upcalconfp* vector of upper bounds of calibration intervals for *ycal*.

*additionalcal* vector of additional values of the response variable for which calibration values and intervals were calculated.

*xcaladd* vector of calibration values (values of inverse regression function) for *additionalcal*.

*lowcaladdp* vector of lower bounds of calibration intervals for *additionalcal*.

*upcaladdp* vector of upper bounds of calibration intervals for *additionalcal*.

Components added by function *confpar*

*coverage* coverage probability of confidence intervals calculated by *confpar*.

*parstats* confidence intervals, estimates of relative absolute error, bias and standard deviation of the parameter estimates.

*fpar* function provided to *confpar* with arguments *x* (design), *fw* (values of the regression function at (*design, par*)), *par* (parameters); used to calculate a parameter of interest.

*fparvalue* estimate of  $fpar(x, fw, par)$

*fparstats* confidence intervals, estimates of mean absolute relative error, bias and standard deviation of the estimated function  $fpar(x, fw, par)$ .

## DETAILS

Objects of class "*nlsobj*" are created as components of the returned list by the functions *selectmv* (model selection with variance estimation) and *selectmt* (model selection with transformation). They contain all results of global interest calculated for a single regression model (in combination with a variance model or using a Box-Cox transformation with fixed  $\lambda$ ). The information contained in the object is used by the functions *sensitive*, *dblbootstrap*, *predict.nlsobj*, *calibrate* and *confpar*, which themselves add their results to the object. There are generic functions *print.nlsobj* and *plot.nlsobj* to print or plot the results obtained so far in a standardized way.

see also

*selectmt*, *selectmv*, *sensitive*, *dblbootstrap*, *predict.nlsobj*, *calibrate*, *confpar*, *print.nlsobj*, *plot.nlsobj*

nlsobjlist
------------

## DESCRIPTION

This is an object of class "*nlsobjlist*". It is created by the model choice functions *selectmt* (selection of models with transformation) or *selectmv* (selection of models with variance estimation). The components of the list are objects of class "*nlsobj*" containing the results

belonging to single models (regression function and transformation or variance model) considered in the process of model choice. The components are ordered by the value of the selected criteria. There are generic functions *print.nlsobjlist* and *plot.nlsobjlist* to display the the results in a standardized way.

see also

*selectmt*, *selectmv*, *sensitive*, *dblbootstrap*, *predict.nlsobj*, *calibrate*, *confpar*, *print.nlsobj*, *plot.nlsobj*, *print.nlsobjlist*, *plot.nlsobjlist*

parinit
---------

## DESCRIPTION

Interactive tool to obtain initial parameter estimates for a list of nonlinear regression models

## USAGE

```
parinit(data, fctl=fctl, lambda=1, steps=10, simp=100, graph=F, display=NULL,
        case="A")
```

## REQUIRED ARGUMENTS

*data* regression data; list with numerical components  $\$x$  (values of the explanatory variable) and  $\$y$  (values of the response variable), character components  $\$xname$  and  $\$yname$  (will be used in summaries and plots as labels for  $\$x$  and  $\$y$ ).  $\$x$  and  $\$y$  have to be of the same length.

*fctl* list specifying the class of models under consideration. The components of the list contain the description of the regression models contained in the class. The single model is specified using a list with components  $\$fkt$  (containing an expression (third element of a formula object) used to compute the regression model, with the explanatory variable denoted by  $xr$  and parameters denoted by  $p1, p2, \dots$ ),  $\$p$  (number of parameters),  $\$inv$  (containing an expression (third element of a formula object) used to compute the inverse of the regression model, with dependent variable denoted by  $yr$  and parameters denoted by  $p1, p2, \dots$ ) and  $\$name$  (character string used as a name of the model). (see example) Standard: *fctl* (a list of sigmoidal and asymptotic models with class attribute "*fctl*")

## OPTIONAL ARGUMENTS

*lambda* exponent  $\lambda$  used in Box-Cox-transformation.

*steps* maximal number of attempts to solve the Hartley-Booker-Equations.

*simp* influences the number of iterations in stochastic search (positive numeric)



*graph* logical flag: if *TRUE*, the actual fit and residuals are plotted in a graphics window to assist in the decision process. Graphics menus are used.

*display* display to be used in case of *graph = T* ("-display hostname:0.0" , be sure to allow access before)

*case* transformation case to be used ; "A" if transformation of both response variable and regression function, or "B" if only transformation of the response variable (see references).

## VALUE

list of regression models. The list consists of descriptions of all regression models contained in the list provided by parameter *fktl* and accepted in the preselection process performed by *parinit*. The components of the list contain the description of the regression models given with parameter *fktl* with initial parameter estimates (component *\$par*), value of *lambda* (component *\$lambda*) and case (component *\$case*) used for transformation and the internal return code from *do\_nls* (component *\$ccode*) added. The class attribute of the list is set to "*fktlist*".

## SIDE EFFECTS

The function *parinit* should be used before calling to the functions *selectmv* and *selectmt*. Function *selectmt* assumes that initial estimates have been computed using digestible values of *case* and *lambda*.

## DETAILS

The function supports several strategies to find initial estimates for the regression problems defined by *data* and *fktl*. At first a subclass of models can be selected (models deleted here will not be in the returned list). The models in this class are successively analyzed. Starting parameters are generated automatically if not contained in the model description. Based on status information obtained for the actual parameter values, a cycle of stochastic searches is started as long as the user decides that further improvement is necessary. New starting values can be provided as a guess or generated. The parameter estimates can be accepted or in case of missing flexibility the model can be deleted. To obtain improvements of the initial estimates the Hartley-Booker-Equations (obtained by grouping the data to form *p* clusters) may be solved by a combination of stochastic search and numerical minimization. Then the original problem is analyzed using stochastic search and numerical minimization of the least squares criteria.

see also

`print.fktlist`, `selectmt`, `selectmv`

## EXAMPLES

```
# construct a list of models:
model1 <- list(fkt = (yr ~ p1 - p2 * exp( - p3 * xr^p4))[[3]],
              p=4, inv = (xr ~ (log(p2/(p1 - yr))/p3)^(1/p4))[[3]],
              name="Weibull (1.7)")
model15 <- list(fkt =
              (yr ~ p1 + p2 * (1 - exp( - p3 * xr))^p4)[[3]], p=4,
              inv = (xr ~ - log(1 - ((yr - p1)/p2)^(1/p4))/p3)[[3]],
              name="(1.44)")
fctl <- list(model1, ..., model15)
# generate the data of Example 1
x <- c( 9, 14, 21, 28, 42, 57, 63, 70, 79)
y <- c( 8.93, 10.80, 18.59, 22.33, 39.35, 56.11, 61.73,
       64.62, 67.08)
example1 <- list( x = x, y = y, xname =
                 "Time after pasture", yname = "Yield")
# complete fctl by initial parameter estimates, use
# graphical display
fctlneu <- parinit(example1, fctl = fctl, graph = T)
# print the result of the initialization
print(fctlneu)
```

plot.nlsobj
-------------

## DESCRIPTION

Generic function to plot the results for a single model (in combination with transformation or variance model) obtained by the functions *selectmv*, *selectmt*, *predict.nlsobj*, *calibrate* and *confpar*.

## USAGE

```
plot.nlsobj(nlsobj, display=NULL, additional=F, design=T, vertical=F, transformed=F,
           cvres=F)
```

## REQUIRED ARGUMENTS

*nlsobj* object of class "*nlsobj*" (see *nlsobj* for a detailed description).

## OPTIONAL ARGUMENTS

*display* display to be used ("*-display hostname:0.0*" , be sure to allow access before).

*additional* logical flag: if *TRUE*, prediction, confidence and calibration intervals (if computed) at the points provided to the argument *additional* in the preceding calls of the functions *predict.nlsobj* and *calibrate* are displayed.

*design* logical flag: if *TRUE*, prediction, confidence and calibration intervals (if computed) at the design points (for the fitted values in case of calibration) are displayed.

*vertical* logical flag: determines vertical or horizontal arrangement of plots.

*transformed* logical flag: if *TRUE*, results under transformation are plotted

*cvres* logical flag: if *TRUE*, cross-validated residuals are plotted additionally (symbol: octagon; color: 2)

see also

*nlsobj*, *selectmv*, *selectmt*, *nlsobjlist*, *plot.nlsobjlist*, *print.nlsobj*, *print.nlsobjlist*, *confpar*, *predict.nlsobj*, *calibrate*

## plot.nlsobjlist

### DESCRIPTION

Generic function to plot the results returned by function *selectmv* or *selectmt*. The function realizes successive calls of *plot.nlsobj* for the models analyzed by *selectmv* or *selectmt*.

### USAGE

```
plot.nlsobjlist(nlsobjlist, display=NULL, transformed=F, cvres=F)
```

### REQUIRED ARGUMENTS

*nlsobjlist* list of class "*nlsobjlist*" returned by function *selectmv* or *selectmt*. The components of the list are objects of class "*nlsobj*" containing the results of a single model.

### OPTIONAL ARGUMENTS

*display* display to be used ("*-display hostname:0.0*" , be sure to allow access before).

*transformed* logical flag: if *TRUE*, results under transformation are plotted

*cvres* logical flag: if *TRUE*, cross-validated residuals are plotted additionally (symbol: octagon; color: 2)

## DETAILS

The function creates menus containing information about names of the regression models and the value of lambda or the number of the variance model used for each of the components of the list `nlsojlist` created by function `selectmt` or `selectmv`. The menu entries are ordered by the values of criterion used in the analysis. The single models to plot can be selected successively.

see also

`nlsoj`, `selectmv`, `selectmt`, `print.nlsojlist`

predict.nlsoj
---------------

## DESCRIPTION

Compute bootstrap prediction and confidence intervals in a nonlinear regression model with transformation or variance modelling

## USAGE

```
predict.nlsoj(model, nb=200, nb2=NULL, nb3=-1, design=F, additional=NULL,
              coverage=.9, replications=1, graph=F, display=NULL, useymean=F)
```

## REQUIRED ARGUMENTS

*model* object of class "*nlsoj*" (component of the list returned by function *selectmv* or *selectmt*) containing a description of a regression model (with variance model or transformation) and corresponding estimates.

## OPTIONAL ARGUMENTS

*nb* number of bootstrap samples

*nb2* number of samples in second level bootstrap. If *NULL* double bootstrap is not performed and only the simple bootstrap intervals are computed (leading to asymptotically less efficient but less time consuming statistics).

*nb3* number of independent standard normally distributed values  $u_a^b$  (for each bootstrap sample *b*) used in the definition of the bootstrap prediction intervals. If *nb3* < 1 the sample size is used instead of *nb3*.

*design* logical flag: if *TRUE*, prediction and confidence intervals are calculated at design points  $x_i$  with if *model*\$*a* <=  $x_i$  <= *model*\$*b* (design points in the region of interest in the cross-validation criteria C and C-tilde).

*additional* additional vector of design points for which prediction and confidence intervals are calculated.

*coverage* coverage probability of the prediction and confidence intervals

*replications* number of replications; prediction intervals are computed for the mean of "replications" future values of the response variable at a given design point.

*graph* logical flag: if *TRUE*, the results are plotted.

*display* display to be used ("display hostname:0.0" , be sure to allow access before).

*useymean* logical flag: if *FALSE*, fitted values from the reference model are used to generate bootstrap samples; if *TRUE*, bootstrap samples are generated based on the vector of means of the observed responses at the design points (instead of fitted values from the reference model).

## VALUE

The parameter *model* is returned with the following components added (or updated):

*coveragep* coverage probability of prediction and confidence intervals calculated by function *predict.nlsobj*

*replicationsp* number of replications; prediction intervals were computed for the mean of *replicationsp* future values of the response variable at a given design point.

*additional* additional vector of design points for which prediction and confidence intervals are calculated.

*lowconf* vector of lower confidence bounds for values of the regression function at the design points *xk* (ordered design points without replications).

*upconf* vector of upper confidence bounds for values of the regression function at the design points *xk*.

*lowconfp* vector of lower prediction bounds at the design points *xk*.

*upconfp* vector of upper prediction bounds at the design points *xk*.

*lowadd* vector of lower confidence bounds for values of the regression function at the additional design points .

*upadd* vector of upper confidence bounds for values of the regression function at the additional design points .

*fwadd* vector of fitted values at the additional design points.

*lowaddp* vector of lower prediction bounds at the additional design points.

*upaddp* vector of upper prediction bounds at the additional design points.

If the function *sensitive* is called by the function *predict.nlsobj* the following components are added:

*tvartyp* number of variance model used to generate bootstrap samples (selected by cross-validation; in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*tvarpar* parameters of variance model used to generate bootstrap samples (calculated in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*trefvar* homogeneous variance estimate (under transformation if  $\lambda \neq 1$ ).

*twcv* vector of values obtained for cross-validation criterion for variance estimation (for all variance models; calculated in the transformed space if  $\lambda \neq 1$ ).

*bootstrap* matrix of parameter estimates obtained from the bootstrap samples.

*yboot* bootstrap samples

*fwboot* fitted values obtained from the bootstrap samples

*useymean* logical flag given by parameter *useymean*

If the function *dblbootstrap* is called by the function *predict.nlsobj*, the following components are added:

*double* array of parameter estimates obtained from second level bootstrap samples (double bootstrap).

*bsvarpar* matrix of parameters of the variance model used to generate second level bootstrap samples; obtained from the bootstrap samples.

*bsrefvar* vector of homogeneous variance estimates obtained from the bootstrap samples.

## SIDE EFFECTS

If the object given by parameter *model* does not contain appropriate information about the bootstrap samples and estimates (bootstrap estimates not computed in advance or number of bootstrap samples less than *nb* or *model\$useymean*  $\neq$  *useymean*) the function *sensitive* is called to (re)calculate the bootstrap estimates. If *nb2*  $\neq$  *NULL* and *model* does not contain appropriate information about double bootstrap samples and estimates (double bootstrap estimates not computed in advance or number of double bootstrap samples less than *nb2* or the function *sensitive* has been called by *calibrate*) the function *dblbootstrap* is called to (re)do the second level bootstrap.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

*selectmv*, *selectmt*, *nlsobj*, *calibrate*, *confpar*, *nlsobjlist*, *plot.nlsobj*, *print.nlsobj*, *plot.nlsobjlist*, *print.nlsobjlist*

## EXAMPLES

```
# generate the data of example 2
x <- c(0, 0, 0, 0, 0, 0, 0, 0, 0, 0.02, 0.02, 0.02, 0.02,
      0.04, 0.04, 0.04, 0.04, 0.06, 0.06, 0.06, 0.06,
      0.08, 0.08, 0.08, 0.08, 0.1, 0.1, 0.1, 0.1, 0.2,
      0.2, 0.2, 0.2, 0.4, 0.4, 0.4, 0.4, 0.6, 0.6, 0.6,
      0.6, 0.8, 0.8, 0.8, 0.8, 1, 1, 1, 1, 1.5, 1.5, 1.5,
      1.5, 2, 2, 2, 2, 4, 4, 4, 4, Inf, Inf, Inf, Inf)
logx <- log(x,10)
logx[logx == -Inf] <- -3
logx[logx == Inf] <- 2
# use -3 and 2 as codes for -Inf and Inf
y <- c(2868, 2785, 2849, 2805, 2779, 2588, 2701, 2752,
      2615, 2651, 2506, 2498, 2474, 2573, 2378, 2494,
      2152, 2307, 2101, 2216, 2114, 2052, 2016, 2030,
      1862, 1935, 1800, 1871, 1364, 1412, 1377, 1304,
      910, 919, 855, 875, 702, 701, 689, 696, 586,
      596, 561, 562, 501, 495, 478, 493, 392, 358,
      399, 394, 330, 351, 343, 333, 250, 261, 244,
      242, 131, 135, 134, 133)
example2 <- list( x = logx, y = y, xname = "log-dose",
                 yname = "response")
# regression models, from catalog used in example 2
# indicator functions are used to identify -3 as -Inf
# and 2 as Inf
model1 <- yr ~ p1 * exp( - exp(p2 - p3 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model2 <- yr ~ p1 + p2/(1 + exp(p3 - p4 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p2
model3 <- yr ~ p1 + p2/(1 + exp(p3 - p4 * xr))^p5 *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p2
model4 <- yr ~ p1/(1 + exp(p2 - p3 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model5 <- yr ~ p1/(1 + exp(p2 - p3 * xr))^(1/p4) *
          I((-3 < xr) & (xr < 2)) + I(xr <= -3) * p1
model6 <- yr ~ p1 + p2 * exp( - exp(p3 - p4 * xr)) *
          I((-3 < xr) & (xr < 2)) + I(2 <= xr) * p2
# inverse models (to be used for calibration)
invmod1 <- xr ~ (p2 - log(log(p1/yr)))/p3
invmod2 <- xr ~ (p3 - log(p2/(yr - p1) - 1))/p4
```

```

invmod3 <- xr ~ (p3 - log((p2/(yr - p1))^(1/p5) - 1))/p4
invmod4 <- xr ~ (p2 - log(p1/yr - 1))/p3
invmod5 <- xr ~ (p2 - log((p1/yr)^p4 - 1))/p3
invmod6 <- xr ~ (p3 - log(log(p2/(yr - p1))))/p4
# now create the list of models
model1 <- list(fkt = model1[[3]], inv = invmod1[[3]],
              p = 3, name = "Gompertz")
model2 <- list(fkt = model2[[3]], inv = invmod2[[3]],
              p = 4, name = "Logistic")
model3 <- list(fkt = model3[[3]], inv = invmod3[[3]],
              p = 5, name = "Richards")
model4 <- list(fkt = model4[[3]], inv = invmod4[[3]],
              p = 3, name = "(1.36)")
model5 <- list(fkt = model5[[3]], inv = invmod5[[3]],
              p = 4, name = "(1.42)")
model6 <- list(fkt = model6[[3]], inv = invmod6[[3]],
              p = 4, name = "(1.43)")
fktlistex2 <- list(model1, model2, model3, model4,
                  model5, model6)
# all other models from the catalog given in the
# references are to restrictive
fktlistex2 <- parinit(example2, fktlistex2, graph=T)
# now select a combination of regression and
# variance model
example2.vresults <- selectmv(example2, fktlistex2as,
                              a=-2.99, b=1.99, graph=T)
# select all models, variance models and the cross-
# validation criteria C or C-tilde from the menus displayed
#
# now compute predictions, prediction and confidence
# intervals for the values of the explanatory variable
# given in the argument additional
example2.bestvpred <- predict(example2.vresults[[1]], nb2=20,
                              design=F, additional=seq(-3,2,.5))
# display the results for the best model
plot(example2.bestvpred,additional=T)
# plot results in the transformed space
plot(example2.bestvpred, additional=T, transformed=T))

```



## DESCRIPTION

Formats and prints the information contained in the description of a class of nonlinear regression models. This method is automatically called by *print* when an object of class "*fktlist*" is given.

## USAGE

```
print.fktlist(fktlist)
```

## REQUIRED ARGUMENTS

*fktlist* list specifying the class of regression models under consideration. The components of the list contain the description of the regression models contained in the class. A single regression model is specified using a list with components *\$fkt* (containing an expression (third element of a formula object) used to compute the regression model, with the explanatory variable denoted by *xr* and parameters denoted by *p1, p2, ...*), *\$p* (number of parameters), *\$inv* (containing an expression (third element of a formula object) used to compute the inverse of the regression model, with dependent variable denoted by *yr* and parameters denoted by *p1, p2, ...*), *\$name* (character string used as a name of the model), *\$par* (initial parameter estimates obtained by function *parinit*), *\$case* and *\$lambda* (the characterization of the transformation used in function *parinit*) and *\$ccode* (the return code from the last call of *da.nls* performed for the model in function *parinit*).

see also

*parinit*, *selectmv*, *selectmt*

## EXAMPLES

```
# construct a list of models:
model1 <- list(fkt = (yr ~ p1 - p2 * exp( - p3 * xr^p4))[[3]],
              p=4, inv = (xr ~ (log(p2/(p1 - yr))/p3)^(1/p4))[[3]],
              name="Weibull (1.7)")
model15 <- list(fkt =
              (yr ~ p1 + p2 * (1 - exp( - p3 * xr))^p4)[[3]], p=4,
              inv = (xr ~ - log(1 - ((yr - p1)/p2)^(1/p4))/p3)[[3]],
              name="(1.44)")
fktl <- list(model1, ..., model15)
# print the current list
print(fktlneu)
# generate the data of Example 1
```

```

x <- c( 9, 14, 21, 28, 42, 57, 63, 70, 79)
y <- c( 8.93, 10.80, 18.59, 22.33, 39.35, 56.11, 61.73,
      64.62, 67.08)
example1 <- list( x = x, y = y, xname =
                  "Time after pasture", yname = "Yield")
# complete fctl by initial parameter estimates, use
# graphical display
fctlneu <- parinit(example1, fctl = fctl, graph = T)
# print the result of the initialization
print(fctlneu)

```

print.nlsobj
--------------

## DESCRIPTION

Generic function to print the results for a single model (in combination with transformation or variance modelling) obtained by the functions *selectmv*, *selectmt*, *confpar*, *predict.nlsobj* and *calibrate*.

## USAGE

```
print.nlsobj(nlsobj, graph=F, display=NULL)
```

## REQUIRED ARGUMENTS

*nlsobj* object of class "*nlsobj*" (see *nlsobj* for a detailed description).

## OPTIONAL ARGUMENTS

*graph* logical flag: if *TRUE*, the results will be displayed using the function *plot.nlsobj*.

Graphics menus are used in this case to select the range of information to be printed.

*display* display to be used ("display hostname:0.0" , be sure to allow access before).

see also

*nlsobj*, *selectmv*, *selectmt*, *nlsobjlist*, *plot.nlsobj*, *plot.nlsobjlist*, *print.nlsobjlist*, *predict.nlsobj*, *calibrate*, *confpar*

## print.nlsobjlist

### DESCRIPTION

Generic function to print the results returned by *selectmv* or *selectmt*. The results are given as a table containing a model identifier and the values of the criteria computed by *selectmv* or *selectmt*. The models are ordered by the criteria selected in *selectmv* or *selectmt*.

### USAGE

```
print.nlsobjlist(nlsobjlist)
```

### REQUIRED ARGUMENTS

*nlsobjlist* list of class "*nlsobjlist*" returned by function *selectmv* or *selectmt*. The components of the list are objects of class "*nlsobj*" containing the results of a single model.

see also

*selectmv*, *selectmt*, *nlsobj.object*

## selectmt

### DESCRIPTION

Menu program for model selection and variable transformation

### USAGE

```
selectmt(data, fktlist=fktlist, lambda=1, fg=NULL, a=min(data$x), b=max(data$x),  
         case="A", A=min(data$y), B=max(data$y), graph=F, display=NULL)
```

### REQUIRED ARGUMENTS

*data* regression data; list with numerical components  $x$  (values of the explanatory variable) and  $y$  (values of the response variable), character components  $xname$  and  $yname$  (will be used in summaries and plots as labels for  $x$  and  $y$ ).  $x$  and  $y$  have to be of the same length.

*fktlist* list specifying the class of regression models under consideration. The components of the list contain the description of the regression models contained in the class. A single regression model is specified using a list with components  $fkt$  (containing an expression (third element of a formula object) used to compute the regression model, with the explanatory variable denoted by  $xr$  and parameters denoted by  $p1, p2, \dots$ ),

$\$p$  (number of parameters),  $\$inv$  (containing an expression (third element of a formula object) used to compute the inverse of the regression model, with dependent variable denoted by  $yr$  and parameters denoted by  $p1, p2, \dots$ ),  $\$name$  (character string used as a name of the model),  $\$par$  (initial parameter estimates obtained by function *parinit*),  $\$case$  and  $\$lambda$  (the characterization of the transformation used in function *parinit*). *ftl* can be generated from a simpler standard form using the function *parinit* (see example).

## OPTIONAL ARGUMENTS

- lambda* vector of parameters *lambda* used in Box-Cox-transformations. Each pair of *lambda* and regression function defines a transformation model. Standard value *lambda* = 1 (with *case* = "A") stands for no transformation.
- fg* the formula to evaluate the parameter function in criterion CG, standard name *fgamma*.
- a* lower bound for  $x$  in the definition of the cross-validation criteria C and C-tilde, standard *min(data\$x)*.
- b* upper bound for  $x$  in the definition of the cross-validation criteria C and C-tilde, standard *max(data\$x)*.
- case* character (value "A" or "B"); transformation case; "A" specifies transformation of both response variable and regression function, "B" calls for transformation of the response variable only (see references).
- A* lower bound for  $y$  in the definition of the calibration criterion, standard *min(data\$y)*.
- B* upper bound for  $y$  in the definition of the calibration criterion, standard *max(data\$y)*.
- graph* logical flag: if *TRUE*, the results for the analyzed models are plotted. Graphics menus are used.
- display* display to be used ("display hostname:0.0" , be sure to allow access before).

## VALUE

the returned object is a list of class "*nlsobjlist*". The components of the list are objects of class "*nlsobj*" containing the results belonging to single models (regression function and transformation) considered in the process of model choice. The components are ordered by the value of the selected criteria. Results can be displayed using *print(returnedobject)* and *plot(returnedobject)* (for results of all analyzed models), or *print(returnedobject[[i]])* and *plot(returnedobject[[i]])* for a more detailed information about the  $i$  – th model (ordered by the value of the model selection criteria).

## SIDE EFFECTS

the function assumes that initial estimates have been computed by function *parinit* using digestible values of *case* and *lambda* (same specification of *case*, same value of *lambda* if

*case = "B"*). In case of indigestible transformations this will be notified and an empty object will be returned.

## DETAILS

The function allows model selection among a class of nonlinear regression models. The class of regression models can be defined using the parameter *fktlist*. Initial parameter estimates can be obtained and included into *fktlist* using the function *parinit*. The models to be analyzed can be selected from this list interactively. The function provides the following criteria for model selection:

- Residual Sum of Squares,
- cross-validation criterion C (Prediction),
- full cross-validation criterion C-tilde (Prediction),
- Calibration-criterion CC and
- Parameter estimation criterion CG.

The Box-Cox-transformations can be performed for both case "A" and case "B" by specifying the *lambda* and *case* parameters. If *case = "B"* is specified only a single  $\lambda$ -value can be used. If *graph = T* graphics menus are used and a graphical representation of the analyzed models is provided. Results will be ordered by the value of the selected criteria.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

*parinit*, *selectmv*, *nlsobj*, *nlsobjlist*, *plot.nlsobjlist*, *print.nlsobjlist*, *plot.nlsobj*, *print.nlsobj*, *confpar*, *predict.nlsobj*, *calibrate*

## EXAMPLES

```
# construct a list of models:
model1 <- list(fkt = (yr ~ p1 - p2 * exp( - p3 * xr^p4))[[3]],
              p=4, inv = (xr ~ (log(p2/(p1 - yr))/p3)^(1/p4))[[3]],
              name="Weibull (1.7)")
model15 <- list(fkt =
              (yr ~ p1 + p2 * (1 - exp( - p3 * xr))^p4)[[3]], p=4,
              inv = (xr ~ - log(1 - ((yr - p1)/p2)^(1/p4))/p3)[[3]],
              name="(1.44)")
fkt1 <- list(model1, ..., model15)
```

```

# generate the data of Example 1
x <- c( 9, 14, 21, 28, 42, 57, 63, 70, 79)
y <- c( 8.93, 10.80, 18.59, 22.33, 39.35, 56.11, 61.73, 64.62, 67.08)
example1 <- list( x = x, y = y, xname =
                  "Time after pasture", yname = "Yield")
# complete fktl by initial parameter estimates, use
# graphical display
fktlneu <- parinit(example1, fktl = fktl, graph = T)
# analyze the specified class of models using
# transformation case A and lambda values
# -2, -1, -.5, 0, .5, 1 and 2 . Allow graphics.
example1.tresults <- selectmt(example1, fktlneu,
                              lambda=c(-2,-1,-.5,0,.5,1,2), graph=T)
# display the results
plot(example1.tresults)
# print the values of the criteria for all models and
# transformations considered
print(example1.tresults)
# print detailed results for the best model with respect
# to the selected criteria
print(example1.tresults[[1]])

```

selectmv
----------

## DESCRIPTION

Menu program for model selection and variance estimation

## USAGE

```

selectmv(data, fktlist=fktlist, vmodels=NULL, fg=NULL, a=min(data$x),
         b=max(data$x), A=min(data$y), B=max(data$y), graph=F, display=NULL)

```

## REQUIRED ARGUMENTS

*data* regression data; list with numerical components  $x$  (values of the explanatory variable) and  $y$  (values of the response variable), character components  $xname$  and  $yname$  (will be used in summaries and plots as labels for  $x$  and  $y$ ).  $x$  and  $y$  have to be of the same length.

*fktlist* list specifying the class of regression models under consideration. The components of the list contain the description of the regression models contained in the class. A single regression model is specified using a list with components  $fkt$  (containing an

expression (third element of a formula object) used to compute the regression model, with the explanatory variable denoted by  $xr$  and parameters denoted by  $p1, p2, \dots$ ),  $\$p$  (number of parameters),  $\$inv$  (containing an expression (third element of a formula object) used to compute the inverse of the regression model, with dependent variable denoted by  $yr$  and parameters denoted by  $p1, p2, \dots$ ),  $\$name$  (character string used as a name of the model),  $\$par$  (initial parameter estimates obtained by function *parinit*),  $\$case$  and  $\$lambda$  (the characterization of the transformation used in function *parinit*;  $case = "A"$  or  $lambda = 1$  is required). *fklist* can be generated from a simpler standard form using the function *parinit* (see example).

## OPTIONAL ARGUMENTS

- vmodels* integer vector, restricts the analysis to the variance models specified by their number. If NULL; all six variance models are used.
- fg* the formula to evaluate the parameter function in criterion CG, standard name *fgamma*.
- a* lower bound for  $x$  in the definition of the cross-validation criteria C and C-tilde, standard *min(data\$x)*.
- b* upper bound for  $x$  in the definition of the cross-validation criteria C and C-tilde, standard *max(data\$x)*.
- A* lower bound for  $y$  in the definition of the calibration criterion, standard *min(data\$y)*.
- B* upper bound for  $y$  in the definition of the calibration criterion, standard *max(data\$y)*.
- graph* logical flag: if *TRUE*, the results for the analyzed models are plotted. Graphics menus are used.
- display* display to be used ("*-display hostname:0.0*" , be sure to allow access before).

## VALUE

the returned object is a list of class "*nlsobjlist*". The components of the list are objects of class "*nlsobj*" containing the results belonging to single models (regression function and variance model) considered in the process of model choice. The components are ordered by the value of the selected criteria. Results can be displayed using *print(returnedobject)* and *plot(returnedobject)* (for results of all analyzed models), or *print(returnedobject[[i]])* and *plot(returnedobject[[i]])* for a more detailed information about the  $i$  – *th* model (ordered by the value of the model selection criteria).

## SIDE EFFECTS

the function assumes that initial estimates have been computed by function *parinit* with specified transformation  $case = "A"$  or  $lambda = 1$  ( $case = "A"$  and  $lambda = 1$  (no transformation) should be preferred). In case of indigestible values of *case* and *lambda* (in *fklist*) this will be notified and an empty object will be returned.

## DETAILS

The function allows model selection among a class of nonlinear regression models. The class of regression models can be defined using the parameter *fktlist*. Initial parameter estimates can be obtained and included into *fktlist* using the function *parinit*. The models to be analyzed can be selected from this list interactively. The function provides the following criteria for model selection:

- Residual Sum of Squares,
- cross-validation criterion C (Prediction),
- full cross-validation criterion C-tilde (Prediction),
- Calibration-criterion CC and
- Parameter estimation criterion CG.

The function allows to chose between 6 variance models (see references) for both the purpose of variance estimation and selection of an appropriate WLSE. Results for the analyzed combinations of regression and variance models will be ordered by the value of the selected criteria.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

*parinit*, *selectmt*, *nlsobj*, *nlsobjlist*, *plot.nlsobjlist*, *print.nlsobjlist*, *plot.nlsobj*, *print.nlsobj*, *confpar*, *predict.nlsobj*, *calibrate*

## EXAMPLES

```
# construct a list of models:
model1 <- list(fkt = (yr ~ p1 - p2 * exp( - p3 * xr^p4))[[3]], p=4,
  inv = (xr ~ (log(p2/(p1 - yr))/p3)^(1/p4))[[3]],
  name="Weibull (1.7)")
model15 <- list(fkt = (yr ~ p1 + p2 * (1 - exp( - p3 * xr))^p4) [[3]],
  p=4, inv = (xr ~ - log(1 - ((yr - p1)/p2)^(1/p4))/p3) [[3]],
  name="(1.44)")
fktl <- list(model1, ..., model15)
# generate the data of Example 1
x <- c( 9, 14, 21, 28, 42, 57, 63, 70, 79)
y <- c( 8.93, 10.80, 18.59, 22.33, 39.35, 56.11, 61.73, 64.62, 67.08)
example1 <- list( x = x, y = y, xname =
```



```

        "Time after pasture", yname = "Yield")
# complete fctl by initial parameter estimates, use
# graphical display
fctlneu <- parinit(example1, fctl = fctl, graph = T)
# analyze the specified class of regression and variance models
example1.vresults <- selectmv(example1, fctlneu, graph=T)
# display the results
plot(example1.vresults)
# print the values of the criteria for all models and
# variance models considered
print(example1.vresults)
# print detailed results for the best model with respect
# to the selected criteria
print(example1.vresults[[1]])

```

sensitive
-----------

## DESCRIPTION

Compute bootstrap estimates for a nonlinear regression model

## USAGE

```
sensitive(model, nb=200, vmodels=NULL, useymean=F, graph=F, display=NULL)
```

## REQUIRED ARGUMENTS

*model* object of class "nlsobj" (component of the list returned by function *selectmv* or *selectmt*) containing a description of a regression model (with variance model or transformation) and corresponding estimates.

## OPTIONAL ARGUMENTS

*nb* number of bootstrap samples

*vmodels* integer vector, restricts the choice of a variance model used to generate the bootstrap errors  $\varepsilon_{ij}^b$  to the models specified in this parameter. If *NULL*; all six variance models are analyzed.

*useymean* logical flag: if *FALSE*, fitted values from the reference model are used to generate bootstrap samples; if *TRUE*, bootstrap samples are generated based on the vector of means of the observed responses at the design points (instead of fitted values from the reference model).

*graph* logical flag: if *TRUE*, the choice of variance models for the bootstrap resampling is illustrated.

*display* display to be used ("display hostname:0.0" , be sure to allow access before).

## VALUE

The object *model* is returned with the following components added (or updated):

*tvartyp* number of variance model used to generate bootstrap samples (selected by cross-validation; in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*tvarpar* parameters of variance model used to generate bootstrap samples (calculated in the transformed space in case of transformation with  $\lambda \neq 1$ ).

*trefvar* homogeneous variance estimate (under transformation if  $\lambda \neq 1$ ).

*tvcv* vector of values obtained for cross-validation criterion for variance estimation (for all variance models; calculated in the transformed space if  $\lambda \neq 1$ ). *Inf* values correspond to variance models excluded from the analysis using the parameter *vmodels*.

*bootstrap* matrix of parameter estimates obtained from the bootstrap samples.

*bsrefpar* matrix of parameter estimates for the reference model  $m_0$  obtained from the bootstrap samples. Computed in case of variance modelling. If transformation is used the reference model  $m_0$  is assumed to coincide with the model under consideration, i.e. *bsrefpar* would duplicate the information contained in *bootstrap*.

*yboot* matrix of bootstrap samples

*fwboot* matrix of fitted values obtained from the bootstrap samples.

*alphastar* shift parameter used in Box-Cox-transformation with  $\lambda_*$

*alphastarb* vector of shift parameters used for bootstrap samples in Box-Cox-transformations with  $\lambda_*$

*alphab* vector of shift parameters used for bootstrap samples in Box-Cox-transformations with  $\lambda$

*scaleb* vector of scale parameters used for bootstrap samples in Box-Cox-transformations (independent of lambda)

*useymean* logical flag: if *FALSE*, fitted values from the reference model were used to generate bootstrap samples; if *TRUE*, bootstrap samples were generated based on the vector of means of the observed responses at the design points (instead of fitted values from the reference model).

## DETAILS

The function performs the bootstrap sampling as described in Subsection 3.1. The set of competitive variance models (compared by criterion (2.25)) can be restricted by use of the parameter *vmodels* if the function is called directly rather than by functions *calibrate*, *confpar* or *predict.nlsobj*.

If for a bootstrap sample numerical instabilities are recognized in the calculation of the parameter estimates stochastic search will be performed in advance to the call of *da.nls*. Numerical problems are notified and should lead to a critical revision of the results.

## REFERENCES

O. Bunke, B. Droge and J. Polzehl (1995). Model selection, transformations and variance estimation in nonlinear regression. Discussion paper No. 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin.

see also

`selectmv`, `selectmt`, `nlsobj`, `dblbootstrap`, `nlsobjlist`, `plot.nlsobj`, `print.nlsobj`, `plot.nlsobjlist`, `print.nlsobjlist`, `predict.nlsobj`, `calibrate`, `confpar`

## REFERENCES

- [1] R.A. Becker, J.M. Chambers, and A.R. Wilks. *The New S Language*. Wadsworth and Brooks/Cole, Pacific Grove, CA, 1988.
- [2] H. Bunke and O. Bunke (Eds.). *Statistical Inference in Linear Models*. Wiley, Chichester, 1986.
- [3] H. Bunke and O. Bunke (Eds.). *Nonlinear Regression, Functional Relations and Robust Methods*. Wiley, Chichester, 1989.
- [4] O. Bunke. The Roles of Transformations and Response Functions chosen by Maximum Likelihood, *Math. Operationsforsch. Statist Ser. Statist.*, 13, 395-405, 1982.
- [5] O. Bunke, B. Droge, and J. Polzehl. Model selection, transformations and variance estimation in nonlinear regression. Discussion paper 95-52, Sonderforschungsbereich 373, Humboldt-Universität, Berlin, 1995.
- [6] O. Bunke, B. Droge, and J. Polzehl. Selection of regression and variance models in nonlinear regression. Discussion paper, Sonderforschungsbereich 373, Humboldt-Universität, Berlin, 1995.
- [7] R.J. Carroll and D. Ruppert. *Transformation and Weighting in Regression*. Chapman and Hall, London, 1988.
- [8] B. Droge. On a computer program for the selection of variables and models in regression analysis. In V. Fedorov, W.G. Müller, and I.N. Vuchkov, editors, *Model-Oriented Data Analysis*, pages 181–192, Heidelberg, 1992. Springer-Verlag.
- [9] B. Droge. Some comments on cross-validation. Discussion Paper 7, Sonderforschungsbereich 373, Humboldt-Universität, Berlin, 1994.
- [10] H.O. Hartley and A. Booker. Nonlinear least squares estimation. *Annals of Mathematical Statistics*, 36:638–650, 1965.
- [11] S. Huet, A. Bouvier, M.A. Gruet, and E. Jolivet. Practical use of statistical tools for parametric nonlinear regression models. Technical report, INRA, Jouy-en-Josas, France, 1995.
- [12] Statistical Sciences. *S-PLUS Programmers's Manual, Version 3.2*. StatSci, a division of MathSoft, Inc., Seattle, 1993.
- [13] Statistical Sciences. *S-PLUS Reference Manual, Version 3.2*. StatSci, a division of MathSoft, Inc., Seattle, 1993.
- [14] Statistical Sciences. *S-PLUS User's Manual, Version 3.2*. StatSci, a division of MathSoft, Inc., Seattle, 1993.
- [15] G.A.F. Seber and C.J. Wild. *Nonlinear Regression*. Wiley, New York, 1989.