

Medical University of Vienna

Department of Medical Computer Sciences

Section of Clinical Biometrics

Section Head: Prof. M. Schemper

A-1090 VIENNA, Spitalgasse 23

Phone: (+43)(1) 40400/6688

Fax: (+43)(1) 40400/6687

e-mail: [biometrie@meduniwien.ac.at](mailto:biometrie@meduniwien.ac.at)

Technical Report 2/2004

**A SAS macro, S-PLUS library and  
R package to perform logistic regression  
without convergence problems**

Georg Heinze and Meinhard PLONER

e-mail: [georg.heinze@meduniwien.ac.at](mailto:georg.heinze@meduniwien.ac.at)

## Abstract

The phenomenon of separation is observed in the fitting process of a logistic model if the likelihood converges while at least one parameter estimate diverges to  $\pm$  infinity. Separation primarily occurs in small samples with several unbalanced and highly predictive covariates (cf. Heinze, 1999). A procedure by Firth (1993) originally developed to reduce the bias of maximum likelihood estimates provides an ideal solution to monotone likelihood (cf. Heinze & Schemper, 2002). It produces finite parameter estimates by means of penalized maximum likelihood estimation. Corresponding Wald tests and confidence intervals are available but it was shown that penalized likelihood ratio tests and profile penalized likelihood confidence intervals are often preferable.

This Technical Report presents a SAS macro, an S-PLUS library and an R package to apply Firth's procedure to logistic regression. The present report contains the complete User's Guide to these programs including syntax, computational methods and examples.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Estimation</b>	<b>6</b>
<b>3</b>	<b>The SPLUS/R library logistf</b>	<b>8</b>
3.0.1	Overview . . . . .	8
3.0.2	Installation of the library in S-PLUS . . . . .	9
3.0.3	Installation of the library in R . . . . .	10
3.0.4	Example: Epidemiological study . . . . .	10
3.1	The main function ‘logistf’ . . . . .	13
3.1.1	Syntax . . . . .	13
3.1.2	Required arguments . . . . .	13
3.1.3	Optional arguments . . . . .	13
3.1.4	Returned object . . . . .	14
3.1.5	Specific methods . . . . .	15
3.2	The function ‘logistftest’ . . . . .	15
3.2.1	Overview . . . . .	15
3.2.2	Syntax . . . . .	16
3.2.3	Required arguments . . . . .	16
3.2.4	Optional arguments . . . . .	17
3.2.5	Returned object . . . . .	17
3.2.6	Specific methods . . . . .	18
3.3	The function ‘logistfplot’ . . . . .	18
3.3.1	Overview . . . . .	18
3.3.2	Syntax . . . . .	18
3.3.3	Required arguments . . . . .	18
3.3.4	Optional arguments . . . . .	20
3.3.5	Returned object . . . . .	20
3.4	Troubleshooting . . . . .	20
<b>4</b>	<b>The SAS macro fl</b>	<b>22</b>
4.1	Introduction . . . . .	22
4.2	Working with the macro . . . . .	22
4.2.1	Basic options . . . . .	23

4.2.2	Model fitting options . . . . .	25
4.2.3	Output options . . . . .	25
4.2.4	Options useful for simulation . . . . .	28
4.2.5	Titles . . . . .	29
4.3	Printed output . . . . .	29
4.4	Examples . . . . .	29
4.5	Troubleshooting . . . . .	36
<b>5</b>	<b>Concluding remarks</b>	<b>37</b>
5.1	Citing . . . . .	37
5.2	Firth's bias reduction method for Cox regression . . . . .	37

# 1 Introduction

In logistic regression usually the parameters are estimated by maximizing the (log) likelihood function (maximum likelihood method). However, it is also known that there are certain situations where finite maximum likelihood parameter estimates do not exist, i. e. where the likelihood converges to a finite value while at least one parameter estimate diverges to  $\pm\infty$  (Albert & Anderson, 1984). This phenomenon is due to special conditions in a data set and is known as ‘separation’. The probability of occurrence of monotone likelihood is too high to be negligible (Heinze & Schemper, 2002; Heinze, 1999).

A new procedure that arrives at finite estimates for the parameters by a modification of the score function has been proposed for logistic regression by Heinze & Schemper (2002). This modification was originally developed by Firth (1993, 1992a, 1992b) to reduce the bias of maximum likelihood estimates in generalized linear models. These estimates are biased away from zero (cf. e. g. Schaefer, 1983; Cordeiro & McCullagh, 1991) and the occurrence of infinite parameter estimates in situations of separation can be interpreted as an extreme consequence of this property.

The behavior of the new procedure has been extensively studied by Heinze & Schemper (2002) and Heinze (1999).

In statistical software packages for logistic regression the convergence of the model fitting algorithm is usually based on the log likelihood (SAS Institute, 1999; Insightful, 2003). The resulting odds ratio estimates are based on that iteration where the log likelihood changes by less than a very small prespecified value (e. g.  $10^{-6}$ ). In case of monotone likelihood the algorithm converges but nonexistence of finite estimates is often overlooked. The resulting estimates are completely arbitrary and thus extremely inaccurate (Heinze & Schemper, 2002) and misleading.

This Technical Report contains the manual for the S-PLUS and R libraries and the SAS macro and thus extends previous Technical Reports (Ploner, 2001; Heinze, 1999), describing past versions of our programs, and our paper (Heinze & Ploner, 2003). In § 2 we describe the estimation procedure used in fitting the models. In § 3, we give an overview of the library and its installation on a PC. An *Epidemiological Study* example shows the simplicity of the library function call. In subsequent subsections, main function and two additional functions of the library are described by means of syntax and examples. In § 4 we describe the SAS macro’s functionality and options. Some concluding remarks about citing and a related approach to the problem of monotone likelihood in Cox regression are made in § 5.

## 2 Estimation

In general, maximum likelihood estimates are often prone to small sample bias. To reduce this bias, Firth (1993) suggested to maximize the penalized log likelihood  $\log L(\beta)^* = \log L(\beta) + 0.5 \log |I(\beta)|$ , where  $I(\beta)$  is the Fisher information matrix, i. e. minus the second derivative of the log likelihood. Applying this idea to logistic regression, the score function  $U(\beta)$  is replaced by the modified score function  $U(\beta)^* = U(\beta) + a$ , where  $a$  has  $r$ th entry  $a_r = 0.5 \text{tr} \{I(\beta)^{-1} [\partial I(\beta) / \partial \beta_r]\}$ ,  $r = 1, \dots, k$ . Heinze & Schemper (2002) give the explicit formulae for  $I(\beta)$  and  $\partial I(\beta) / \partial \beta_r$ .

In explaining the details of the estimation process we follow the description in Heinze & Ploner (2002). In our programs estimation of  $\beta$  is based on a Newton-Raphson algorithm. Parameter values are usually initialized with 0, but in general the user can specify arbitrary starting values.

With a starting value of  $\beta^{(0)}$ , the penalized maximum likelihood estimate  $\hat{\beta}$  is obtained iteratively:

$$\beta^{(s+1)} = \beta^{(s)} + I(\beta^{(s)})^{-1} U(\beta^{(s)})^* \quad (2.1)$$

If the penalized log likelihood evaluated at  $\beta^{(s+1)}$  is less than that evaluated at  $\beta^{(s)}$ , then  $\beta^{(s+1)}$  is recomputed by step-halving. For each entry  $r$  of  $\beta$  with  $r = 1, \dots, k$  the absolute step size  $|\beta_r^{(s+1)} - \beta_r^s|$  is restricted to a maximal allowed value  $\xi$ . These two means should avoid numerical problems during estimation. The iterative process is continued until the parameter estimates converge.

Computation of profile penalized likelihood confidence intervals for parameters follows the algorithm of Venzon and Moolgavkar (1988). For testing the hypothesis of  $\gamma = \gamma_0$ , let the likelihood ratio statistic  $LR = 2 [\log L(\hat{\gamma}, \hat{\delta})^* - \log L(\gamma_0, \hat{\delta}_{\gamma_0})^*]$ , where  $(\hat{\gamma}, \hat{\delta})$  is the joint penalized maximum likelihood estimate of  $\beta = (\gamma, \delta)$ , and  $\hat{\delta}_{\gamma_0}$  is the penalized maximum likelihood estimate of  $\delta$  when  $\gamma = \gamma_0$ . The profile penalized likelihood confidence interval is the continuous set of values  $\gamma_0$  for which  $LR$  does not exceed the  $(1 - \alpha)100$ th percentile of the  $\chi_1^2$ -distribution. The confidence limits can therefore be found iteratively by approximating the penalized log likelihood function in a neighborhood of  $\beta$  by the quadratic function

$$\tilde{\ell}(\beta + \delta) = \ell(\beta) + \delta' U^* - \frac{1}{2} \delta' I \delta$$

where  $U^* = U(\beta)^*$  and  $-I = -I(\beta)$ . Suppose the confidence limits for parameter  $\beta_r$  are to be computed. Then the increment vector  $\delta$  for the next iteration is obtained by solving

the likelihood equations

$$\frac{\partial}{\partial \delta} \{ \tilde{\ell}(\beta + \delta) + \lambda(e'_r \delta - \theta) \} = 0$$

where  $\lambda$  is a Lagrange multiplier,  $e_r$  is the  $r$ th unit vector, and  $\theta$  is an unknown constant.

The solution is

$$\delta = I^{-1}(U^* + \lambda e_r) \quad (2.2)$$

Let  $\ell_0 = \ell_{\max} - \chi_{1,1-\alpha}^2/2$ , where  $\ell_{\max}$  is the maximized value of the penalized log likelihood, and  $\chi_{1,\alpha}^2$  is the  $\alpha$ -quantile of the  $\chi^2$ -distribution with one degree of freedom.

By substituting (2.2) into the equation  $\tilde{\ell}(\beta + \delta) = \ell_0$ ,  $\lambda$  can be estimated as

$$\lambda = \pm \left[ \frac{2 \left\{ \ell_0 - \ell(\beta) - \frac{1}{2} U^{*'} I^{-1} U^* \right\}}{e'_r I^{-1} e_r} \right]^{\frac{1}{2}}$$

The upper confidence limit for  $\beta_r$  is computed by starting at the penalized maximum likelihood estimate of  $\beta$  and iterating with positive values of  $\lambda$  until convergence is attained. The process is repeated for the lower confidence limit using negative values of  $\lambda$ . Convergence is declared on the current iteration if  $|\ell(\beta) - \ell_0| \leq \epsilon$  and  $(U^* + \lambda e_r)' I^{-1} (U^* + \lambda e_r) \leq \epsilon$ .

In some situations computation of profile penalized likelihood confidence intervals may be time consuming since the iterative procedure outlined above has to be repeated for the lower and for the upper confidence limits of each of the  $k$  parameters. In other problems one may not be interested in interval estimation, anyway. In such cases, the user can request computation of Wald confidence intervals and  $P$ -values, which are based on the normal approximation of the parameter estimates and do not need any iterative estimation process. Standard errors  $\hat{\sigma}_r$ ,  $r = 1, \dots, k$ , of the parameter estimates are computed as the roots of the diagonal elements of the variance matrix  $V(\hat{\beta}) = I(\hat{\beta})^{-1}$ . A  $(1 - \alpha) \times 100\%$  Wald confidence interval for parameter  $\beta_r$  is then defined as  $[\hat{\beta}_r + \Phi_{\alpha/2} \hat{\sigma}_r, \hat{\beta}_r + \Phi_{1-\alpha/2} \hat{\sigma}_r]$  where  $\Phi_\alpha$  denotes the  $\alpha$ -quantile of the standard normal distribution function. The adequacy of Wald confidence intervals for parameter estimates should be verified by plotting the profile penalized log likelihood (PPL) function. A symmetric shape of the PPL function allows use of Wald intervals, while an asymmetric shape demands profile penalized likelihood intervals (Heinze & Schemper, 2002).

## 3 The SPLUS/R library `logistf`

### 3.0.1 Overview

The S-PLUS library `logistf` provides a comprehensive tool to facilitate the application of Firth's modified score procedure in logistic regression analysis. It has been developed on a PC with S-PLUS 4.0 by Meinhard Ploner and Georg Heinze and later been translated to S-PLUS 6.0 by Harry Southworth ([harry.southworth@astrazeneca.com](mailto:harry.southworth@astrazeneca.com)). Daniela Dunkler's S-PLUS programming skills were very helpful when some bugs had to be fixed. The library should run on newer versions as well as with other operation systems like UNIX. The library is available at the web-site

<http://www.akh-wien.ac.at/imc/biometrie/programme/fl>

(or <http://www.meduniwien.ac.at/user/georg.heinze/programs/logistf>). The old S-PLUS 4 version can still be obtained by sending an e-mail to the author.

An R package `logistf` has been produced from the S-PLUS library by David Duffy ([davidd@qimr.edu.au](mailto:davidd@qimr.edu.au)). It can be downloaded and installed from CRAN ([cran.r-project.org](http://cran.r-project.org)). The structure of the R package is identical to that of the S-PLUS version. A similar R package `brlr` (bias-reduced logistic regression) was developed by David Firth and is also available at CRAN. However, `brlr` does not provide asymmetric confidence intervals based on profile penalized likelihood.

The call of the main function of the library follows the structure of the standard functions as `lm` or `glm`, requiring a `data.frame` and a `formula` for the model specification. The resulting object belongs to the new class `logistf`, which includes penalized maximum likelihood ('Firth-Logistic'- or 'FL'-type) logistic regression parameters, standard errors, confidence limits,  $p$ -values, the value of the maximized penalized log likelihood, the linear predictors, the number of iterations needed to arrive at the maximum and much more. Furthermore, specific methods for the resulting object are supplied. Additionally, a function to plot profiles of the penalized likelihood function and a function to perform penalized likelihood ratio tests have been included.

The summary of the contents of the library can be obtained by calling (in S-PLUS):



```
> library(logistf, help=T)
Library LOGISTF
```

Logistic regression using Firth's bias reduction

Meinhard Ploner, Daniela Dunkler, Harry Southworth and Georg Heinze  
Section of Clinical Biometrics  
Medical University of Vienna  
georg.heinze@meduniwien.ac.at

Functions and data sets in the library:

logistf	fits a logistic regression with Firth's bias reduction
logistfplot	plots the profile penalized likelihood function
logistftest	penalized likelihood ratio test
print.logistf	prints an object of the class <logistf>
print.logistftest	prints an object of the class <logistftest>
summary.logistf	requests the summary of an object of the class <logistf>
sex	the College Woman data set in collapsed form
sex2	the College Woman data set in casewise form

Copyright Medical University of Vienna  
2001-2004, all rights reserved

The online documentation of the R library can be obtained in R by submitting:

```
> help(logistf)
```

### 3.0.2 Installation of the library in S-PLUS

Usually the library is distributed as a packed ZIP-archive. After unpacking the files to the directory `...\splus62netclient\library\` the S-PLUS-files are extracted to `...\splus62netclient\library\logistf\`. After starting an S-PLUS session the library can be loaded directly by submitting

```
> library(logistf)
```

### 3.0.3 Installation of the library in R

If your computer has a network connection, you can download the R package directly from CRAN by choosing `Packages>Install package(s) from CRAN...` from the menu and then choosing `logistf` from the list. Alternatively, if you have the package already downloaded as ZIP file, choose from the menu the item

`Packages>Install package(s) from local ZIP file...` and point the browser to the ZIP file. Then, load the library by `Packages>Load Package...` and choosing `logistf`.

### 3.0.4 Example: Epidemiological study

Use of the library `logistf` is exemplified using the *Epidemiological Study* data set of Foxman et al. (1997), which is available as an ASCII file on the world wide web address <http://www.cytel.com/examples/sex.dat>.

The data set consists of data from 130 college women with urinary tract infections and 109 uninfected controls. It includes the binary covariates age (`age`), oral contraceptive use (`oc`), condom use (`vic`), lubricated condom use (`vicl`), spermicide use (`vis`) and diaphragm used (`dia`). The main question arising from this data set is how urinary tract infection is related to age and contraceptive use.

Suppose the data has been stored in a S-PLUS `data.frame` named `sex`. Because the observations are stored in the collapsed form, i. e. for every covariate-combination the number of successes and total size is given, we have to modify to obtain one row per observation with an indicator for the events. Instead of 24 rows we obtain a new `data.frame` with 239 rows, the number of the observations:

```
> sex2 <- rbind(cbind(case=1, sex[rep(1:24, sex$uti), 3:8]),
               cbind(case=0, sex[rep(1:24, sex$GrpSize-sex$uti), 3:8]))
```

To obtain an object of the class `logistf`, one submits the following commands:

```
> library(logistf)  ## to load the library
> fit <- logistf(case ~ ., data=sex2)
```

A short overview of the results, consisting of a table containing variable names, parameter estimates, standard errors, confidence limits and  $p$ -values, can be obtained by using the `print.logistf` method, which is done automatically by putting the name of an object belonging to the specified class:

```

> fit
logistf(formula = case ~ ., data = sex2) Model fitted by Penalized ML
Confidence intervals and p-values by Profile Likelihood

              coef  se(coef) lower 0.95 upper 0.95      Chisq      p
(Intercept)  0.1202541  0.4855415 -0.8185777  1.0731445  0.06286298  8.020268e-001
age -1.1059815  0.4236601 -1.9737949 -0.3074251  7.50773092  6.143472e-003
oc -0.0688167  0.4437934 -0.9414289  0.7891995  0.02467044  8.751911e-001
vic  2.2688747  0.5484159  1.2730212  3.4354329  22.93139022  1.678877e-006
vicl -2.1114083  0.5430823 -3.2608638 -1.1177349  19.10407252  1.237805e-005
vis -0.7883170  0.4173676 -1.6080866  0.0151937  3.69740975  5.449701e-002
dia  3.0960078  1.6750220  0.7745682  8.0302936  7.89693139  4.951873e-003

```

Likelihood ratio test=49.090642792028 on 6 df, p=7.1508903243256e-009, n=239

The summary method `summary.logistf` produces more output, including the covariance-matrix:

```

> summary(fit)
logistf(formula = case ~ ., data = sex2)
...

```

Wald test = 30.6406242996162 on 6 df, p = 2.96843381479839e-005

Covariance-Matrix:

	(Intercept)	age	oc		dia
(Intercept)	0.23575053	-0.0274268412	-0.195208786	...	-0.026979464
age	-0.02742684	0.1794879029	0.001817975	...	-0.065571649
oc	-0.19520879	0.0018179746	0.196952606	...	0.025480432
vic	-0.12109972	-0.0231659050	0.087887752	...	0.025211511
vicl	-0.02893131	0.0345957183	0.025843014	...	-0.008037505
vis	-0.04648040	0.0007177051	0.042053095	...	-0.045265574
dia	-0.02697946	-0.0655716489	0.025480432	...	2.805698651

The output object consists of many attributes which can be extracted in the usual manner



```
> coef(fit)
(Intercept)      age          oc          vic          vicl          vis          dia
  0.1202541 -1.105982 -0.0688167  2.268875 -2.111408 -0.788317  3.096008
```

## 3.1 The main function ‘logistf’

### 3.1.1 Syntax

The definition of the main function is:

```
logistf <- function(formula=attr(data, "formula"), data=sys.parent(),
  pl = T, alpha = 0.05, maxit = 25, maxhs=5, epsilon = .0001,
  maxstep = 10, firth=T, beta0)
```

### 3.1.2 Required arguments

- **formula**: a formula object, with the response on the left of the  $\sim$  operator, and the model terms on the right. The response must be a vector with 0 and 1 or  $F$  and  $T$  for the model outcome, where the higher value (1 or  $T$ ) is modeled. It’s possible to include contrasts, interactions, nested effects, cubic or polynomial splines and all the features of the S-language, as well, e.g.  $Y \sim X1*X2 + ns(X3, df=4)$ .
- **data**: a `data.frame` where the variables named in the `formula` can be found, i. e. the variables containing the binary response and the covariates.

### 3.1.3 Optional arguments

- **pl**: specifies if confidence intervals and tests should be based on the profile penalized log likelihood (`pl=T`, the default) or on the Wald method (`pl=F`).
- **alpha**: the significance level (1– the confidence level, 0.05 as default)
- **maxit**: maximum number of iterations (default value is 25)
- **maxhs**: maximum number of step-halvings per iterations (default value is 5)
- **epsilon**: specifies the maximum allowed change in penalized log likelihood to declare convergence. Default value is 0.0001.

- **maxstep**: specifies the maximum change of (standardized) parameter values allowed in one iteration. Default value is 0.5.
- **firth**: use of Firth's penalized maximum likelihood (**firth=T**, default) or the standard maximum likelihood method (**firth=F**) for the logistic regression. Note that by specifying **pl=T** and **firth=F** (and probably a lower number of iterations) one obtains profile likelihood confidence intervals for maximum likelihood logistic regression parameters.
- **beta0**: specifies the initial values of the coefficients for the fitting algorithm.

### 3.1.4 Returned object

The object returned is of the class `logistf` and has the following attributes:

- **coefficients**: the coefficients of the parameter in the fitted model.
- **alpha**: the significance level (1– the confidence level) as specified in the input.
- **var**: the variance-covariance-matrix of the parameters.
- **df**: the number of degrees of freedom in the model.
- **loglik**: a vector of the (penalized) log-likelihood of the full and the restricted models.
- **iter**: the number of iterations needed in the fitting process.
- **n**: the number of observations.
- **terms**: an object of mode `expression` and class `term` summarizing the formula as described in the help of S-PLUS.
- **y**: the response-vector, i. e. 1 for successes (events) and 0 for failures.
- **formula**: the `formula` object, see S-PLUS help.
- **call**: the `call` object, see S-PLUS help.
- **linear.predictors**: a vector with the linear predictor of each observation.
- **predict**: a vector with the predicted probability of each observation.

- `hat.diag`: a vector with the diagonal elements of the Hat Matrix.
- `method`: depending on the fitting method "Penalized ML" or "Standard ML".
- `method.ci`: the method in calculating the confidence intervals, i.e. "profile likelihood" or "Wald", depending on the argument `pl`.
- `ci.lower`: the lower confidence limits of the parameter.
- `ci.upper`: the upper confidence limits of the parameter.
- `prob`: the  $p$ -values of the specific parameters.

### 3.1.5 Specific methods

The definition of the methods `print` and `summary` is as follows:

```
print.logistf(object)
summary.logistf(object)
```

where `object` represents an object of the class `logistf`, which is returned by the function `logistf`. The specific `print`-method prints the coefficients, the standard errors, confidence limits and  $p$ -values of the model while the `summary`-method additionally shows the covariance-matrix of the coefficients.

## 3.2 The function 'logistftest'

### 3.2.1 Overview

This function performs a penalized likelihood ratio test on some (or all) selected factors. The resulting object is of the class `logistftest` and includes the information printed by the proper `print` method.

If for the data set we would like to test the specific hypothesis

$\beta_{vic} = 2, \beta_{vict} = 0$ , we do as follows:

```
> logistftest(case ~ ., sex2, test = ~ vic + vicl - 1, values = c(2, 0))
logistftest(formula = case ~ ., data = sex2,
             test = ~ vic + vicl - 1, values = c(2, 0))
Model fitted by Penalized ML
```

Factors fixed as follows:

```
(Intercept) age oc vic vicl vis dia
           NA NA NA  2    0 NA NA
```

Likelihoods:

```
Restricted model Full model difference
           -144.497  -132.5394   11.95762
```

Likelihood ratio test=23.9152322431463 on 2 df, p=6.4102254145881e-00

Testing the overall null hypothesis of  $\beta_i = 0$  would be performed by the following call:

```
> logistftest(case ~ ., data=sex2)
logistftest(formula = case ~ ., data = sex2) Model fitted by Penalized ML
```

Factores fixed as follows:

```
(Intercept) age oc vic vicl vis dia
           NA  0  0  0    0  0  0
```

Likelihoods:

```
Restricted model Full model difference
           -157.0847  -132.5394   24.54532
```

Likelihood ratio test=49.090642792028 on 6 df, p=7.1508903243256e-009

### 3.2.2 Syntax

The definition of the function is

```
logistftest <- function(formula=attr(data, "formula"), data=sys.parent(),
  test, values, maxit = 25, maxhs=5, epsilon = .0001,
  maxstep = 10, firth=T, beta0)
```

### 3.2.3 Required arguments

- **formula**: a formula object, with the response on the left of the  $\sim$  operator, and the model terms on the right. The response must be a vector with 0 and 1 or  $F$  and  $T$  for the model outcome, where the higher value (1 or  $T$ ) is modeled.



- **data**: a `data.frame` where the variables named in the `formula` can be found, i. e. the variables containing the binary response and the covariates.

### 3.2.4 Optional arguments

`logistftest` takes the same arguments as the main function `logistf`, except for `alpha`. Additionally, we can specify the factors/covariates to be tested and the null hypothesis values:

- **test**: righthand formula of parameters to test (e.g. `~ B + D - 1`). As default all parameter apart from the intercept are tested. If `-1` is not included in the formula, the intercept would be tested, too!  
As alternative to the formula one can give the indexes of the ordered effects to test (a vector of integers). To test only the intercept specify `test = ~ - .` or `test = 1`.
- **values**: null hypothesis values, default values are 0. For testing the specific hypothesis  $\beta_1 = 1, \beta_4 = 2, \beta_5 = 0$  we specify `test= ~ B1 + B4 + B5 - 1` and `values=c(1, 2, 0)`.
- **beta0**: usually omitted. If specified, the full model is not fitted, while the restricted model is fitted by starting with this parameter estimates. The option is utilized by the function `coxphfplot`.

### 3.2.5 Returned object

The object returned is of the class `logistf` and has the following attributes:

- **testcov**: a vector of the fixed values of each covariate; `NA` stands for a parameter which is not tested.
- **loglik**: a vector of the (penalized) log-likelihood of the full and the restricted models. If the argument `beta0` not missing, the full model isn't evaluated.
- **df**: the number of degrees of freedom in the model.
- **prob**: the  $p$ -value of the test.
- **call**: the `call` object, see S-PLUS help.
- **method**: depending on the fitting method "Penalized ML" or "Standard ML".
- **beta**: the coefficients on the restricted solution.

### 3.2.6 Specific methods

The definition of the method `print` is:

```
print.logistf(object)
```

where `object` represents an object of the class `logistftest`, which is the output of the function `logistftest`.

## 3.3 The function ‘logistfplot’

### 3.3.1 Overview

This function plots the profile likelihood of a specific parameter. In our example we get the profile of the parameter  $\beta_{dia}$  as follows:

```
> logistfplot(case ~ ., data=sex2, which= ~ dia - 1)
```

Fig. 1 shows the graph of the profile penalized likelihood function for parameter  $\beta_{dia}$ .

### 3.3.2 Syntax

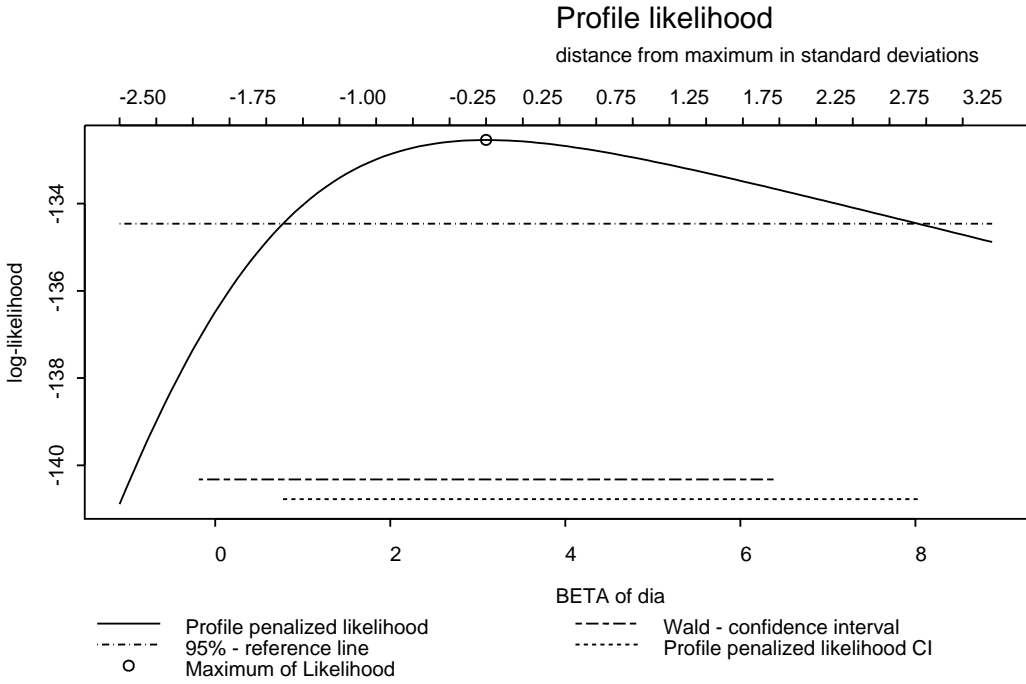
The definition of the function is

```
logistfplot <- function(formula = attr(data, "formula"),
  data = sys.parent(), which, pitch = 0.05, limits, alpha = 0.05,
  maxit = 25, maxhs = 5, epsilon = 0.0001, maxstep = 10, firth = T,
  legends = T)
```

### 3.3.3 Required arguments

- **formula**: a formula object, with the response on the left of the  $\sim$  operator, and the model terms on the right. The response must be a vector with 0 and 1 or  $F$  and  $T$  for the model outcome, where the higher value (1 or  $T$ ) is modeled.
- **data**: a `data.frame` where the variables named in the `formula` can be found, i. e. the variables containing the binary response and the covariates.
- **which**: a righthand formula specifying the plotted parameter, interaction or general term, e.g.  $\sim A - 1$  or  $\sim A : C - 1$ . The profile likelihood of the intercept would be obtained by the formula  $\sim - ..$

Figure 1: Profile penalized likelihood function for parameter  $\beta_{dia}$  of *Epidemiological Study* example.



### 3.3.4 Optional arguments

`logistfplot` takes all arguments of the main function `logistf`, except for `pl`. In addition, we can specify the following arguments:

- **limits**: vector of the minimum and the maximum on the  $x$ -scale in standard deviations distant from the maximum likelihood. The default values are the extremes of both confidence intervals, Wald and PL, plus or minus half a standard deviation of the parameter, respectively.
- **pitch**: distances between the interpolated points in standard errors of the parameter estimate, the default value is 0.05.
- **legends**: if F, legends on the bottom of the plot would be omitted (default is T).

### 3.3.5 Returned object

The object returned is a simple `data.frame` containing three columns which allow reproducing the plot. Each row represents one point of the interpolation. The columns are as follows:

- **std**: distance from the maximum of the profile likelihood (in standard errors of the parameter estimate).
- **name**: the value of the parameter for the variable *name* specified in argument **which**.
- **loglik.pen**: the value of the penalized likelihood.

## 3.4 Troubleshooting

In some data sets, numerical problems in the estimation of profile likelihood confidence intervals can be avoided by restricting the `maxstep` parameter to values of 0.5, say. Also, standardization prior to computations can help much. If you standardize the independent variables  $x$  to a mean of 0 and a standard deviation of 1, you must rescale the obtained parameter estimates in order to have results related to the original data.

Let  $m_j$  and  $s_j$  denote the mean and standard deviation of variable  $x_j$ , respectively, and suppose that with standardized data  $x_j^* = (x_j - m_j)/s_j$ ,  $j = 1, \dots, k$ , you obtain parameter estimate  $\hat{\beta}_j^*$ ,  $j = 0, \dots, k$ ,  $\hat{\beta}_0^*$  denoting the estimate for the intercept parameter. Then the parameter estimates related to the original data can be computed by

$$\hat{\beta}_0 = \hat{\beta}_0^* - \sum_{j=1}^k \hat{\beta}_j^* m_j / s_j$$

and

$$\hat{\beta}_j = \hat{\beta}_j^* / s_j \quad j = 1, \dots, k$$

The covariance matrix  $\hat{V}$  for the original data is best obtained by computing

$$\hat{V} = (X'WX)^{-1}$$

where  $X$  is the original design matrix (including a column of 1's for the intercept), and  $W$  is the  $n \times n$  diagonal matrix

$$W = \begin{pmatrix} \hat{\pi}_1(1 - \hat{\pi}_1) & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & \hat{\pi}_n(1 - \hat{\pi}_n) \end{pmatrix}$$

Confidence limits (CLs) based on profile penalized likelihood for the parameters related to variables  $x_1, \dots, x_k$  can be obtained by dividing the CLs obtained from standardized data by the corresponding standard deviations of  $x_1, \dots, x_k$ . The estimate and profile penalized likelihood CLs for the intercept parameter are invariant to a rescaling of  $x_1, \dots, x_k$ , but not to a relocation of  $x_1, \dots, x_k$ , i.e., remain unchanged if the independent variables are only divided by their standard deviations but not centered.

## 4 The SAS macro fl

### 4.1 Introduction

The SAS macro %fl was written to facilitate application of Firth's modified score procedure (Firth, 1993; Heinze & Schemper, 2002) in logistic regression analysis. The macro was written in 1999 and has now been subject to a complete revision. The new version 2004.07 outperforms any old version (Heinze, 1999; Heinze & Ploner, 2003) in terms of speed (by a factor of at least 3) by a more efficient use of matrix programming in PROC IML, and in terms of computational control by some new options. The macro is available at the websites <http://www.akh-wien.ac.at/imc/biometrie/programme/fl> and <http://www.meduniwien.ac.at/user/georg.heinze/programs/logistf>.

Given the  $X$ -matrix is of full rank (which is normally the case), Firth's procedure (FL) removes the problem of quasicomplete or complete separation in logistic regression analysis. Supplied with a SAS data set as input, the macro output contains FL-type logistic regression parameters, standard errors, confidence limits,  $p$ -values, the value of the maximized penalized log likelihood, and the number of iterations needed to arrive at the maximum. All output produced is stored in a SAS data set. Furthermore, parameter estimates, penalized log likelihood and covariance matrix are saved in a SAS data set that has the same structure as the output data set that can be obtained by SAS/PROC LOGISTIC (SAS Institute, 1999) using the OUTEST option in the PROC LOGISTIC statement of that procedure. The profile of the penalized log likelihood function for any parameter can be plotted to judge the adequacy of Wald confidence intervals. Efficient processing of multiple input data sets is possible through a BY variable, similarly to PROC LOGISTIC. Finally, the option of supplying offset values to the program enables computation of penalized likelihood ratio tests.

### 4.2 Working with the macro

In the following list of options available in %fl, (*new*) indicates options new in version 2004.07:

```
%fl(<data=SAS data set,>  
    <y=variable,>  
    varlist=variables,  
    <noint=value,>
```

```

<odds=value,>
<test=variables,>
<pl=value,>
<plint=value,> (new)
<print=value,>
<notes=value,> (new)
<profile=variables,
<profsel=value,>
<profser=value,>
<profn=value,>
<outprof=SAS data set,>
<outest=SAS data set,>
<outtab=SAS data set,>
<global=SAS data set,> (new)
<out=SAS data set,>
<pred=variable,>
<lower=variable,>
<upper=variable,>
<h=variable,>
<maxit=value,>
<maxhs=value,>
<epsilon=value,>
<maxstep=value,>
<standard=value,> (new)
<by=variables,>
<offset=SAS data set>);

```

These options can be categorized into basic options, output options, model fitting options, and options useful for simulation.

#### 4.2.1 Basic options

- `data=SAS data set` names the input SAS data set. The default value is `_LAST_`.
- `y=variable` names the dependent variable containing values 1 and 0 only. The default value is `Y`.

- **varlist=variables** names a list of independent variables, separated by blanks. There is no default value. This option is required.
- **noint=value** suppresses estimation of an intercept if set to 1. The default value is 0.
- **odds=value** requests estimates and confidence limits for odds ratios if set to 1. The default value is 0.
- **test=variables** requests a likelihood ratio test for the null hypothesis that all parameters corresponding to variables specified in this option are zero.
- **pl=value** specifies the method of computing confidence limits and tests for parameters. If **pl=1**, then profile penalized likelihood confidence limits for parameters and penalized likelihood ratio tests will be computed. Otherwise, confidence limits and tests will be based on estimated variance (Wald method). Default value is 1.
- **plint=value** specifies the method of computing confidence limits and tests for the intercept. This option is only active if **pl=1**. If **plint=0**, which is the default value, confidence limits for the intercept will be based on the estimated variance (Wald method) instead of profile penalized likelihood. The value 1 requests computation of profile penalized likelihood confidence interval for the intercept. This option has been added as we frequently noticed numerical problems when estimating the profile likelihood confidence limits for the intercept parameter.
- **profile=variables** requests a plot of the profile penalized log likelihood function for all variables specified in this options. Of course, these variables have to appear also in the **varlist** option. The  $x$ -axis ranges for this plot are automatically chosen by the macro, but can also be specified by the user in terms of standard errors to the left and right from the point estimate (options **profsel** and **profser**, respectively). Also the number of profile likelihood evaluations (**profn**, default value=100) can be chosen. If the **profile** options is used, then the data set specified in option **outprof** will consist of the variables
  - **\_name\_** containing the covariate's name
  - **\_b\_** containing the values on the  $x$ -axis (the values for  $\beta$ )
  - **\_profli\_** containing the values of the profile penalized log likelihood



- `_normal_` containing the values  $\ell_{\max} - 0.5(\beta - \hat{\beta})^2/\hat{\sigma}^2$  (where  $\ell_{\max}$  is the maximized penalized log likelihood) which represent the Wald (normal) approximation to the profile penalized log likelihood
- `_refer_` containing the reference line (the values of  $\beta$  where the profile penalized log likelihood function and normal approximation crosses the reference line are the profile penalized likelihood and Wald confidence limits, respectively).
- any BY-variables specified

#### 4.2.2 Model fitting options

- `maxit=value` specifies the maximum number of iterations. Default value is 50.
- `maxhs=value` specifies the maximum number of step-halvings allowed in one iteration. Default value is 5.
- `epsilon=value` specifies the maximum allowed sum of absolute changes in parameter values to declare convergence. Default value is 0.0001.
- `maxstep=value` specifies the maximum change of parameter values allowed in one iteration. Default value is 5.
- `standard=value` specifies if the data should be standardized prior to (profile) likelihood maximization. Standardization usually improves the numerical stability of likelihood maximization algorithms and may reduce the number of iterations. Default value is 1, requesting standardization. The only impact of standardization that can be seen by the user is that the value of the penalized likelihood is different from, yet proportional to, the value obtained by estimation from non-standardized data.

#### 4.2.3 Output options

- `print=value` suppresses printed output if set to 0. Default value is 1.
- `outest=SAS data set` names a SAS data set containing parameter estimates, penalized log likelihood and covariance matrix. There is no default value. The data set contains a variable for the intercept parameter and one variable for each explanatory variable in the `varlist` option. The `outest` data set contains one

observation for each BY-group containing the FL-type estimates of the regression coefficients. Additionally, there are observations containing the rows of the estimated covariance matrix of the parameter estimators for each by group. The `outest` data set contains the following variables:

- any BY variables specified
  - `_CODE_`, a variable containing the value -1 indicating a line with parameter estimates or the subsequent numbers of the covariates indicating lines containing corresponding rows of the estimated covariance matrix.
  - if `noint=0`, the variable `INTERCPT`
  - one variable for each explanatory variable in the `varlist` statement.
  - `_LINK_`, a character variable of length 8 with the value `LOGIT`
  - `_PENLIK_`, the maximized penalized log likelihood at the FL estimate (where `_CODE_=-1`) or, if `p1=1`, the maximized penalized log likelihood with the restriction that the corresponding parameter is 0 (where `_CODE_> 0`)
  - `_LNLIKE_`, the log likelihood at the FL estimate (where `_CODE_=-1`) or, if `p1=1`, the log likelihood at the FL estimate maximizing the penalized likelihood with the restriction that the corresponding parameter is 0 (where `_CODE_> 0`)
  - `_IT_`, the number of iterations needed to arrive at the maximum of the penalized likelihood
  - `_RESP_`, the number of responses
  - `_NORESP_`, the number of nonresponses
  - `_TYPE_`, a character variable of length 8 with two possible values: `PARMS` for parameter estimates or `COV` for covariance estimates
  - `_NAME_`, a character variable of length 8 containing the name `ESTIMATE` for parameter estimates or the name of each explanatory variable or `INTERCPT` for the covariance estimates
- `outtab=SAS data set` names a SAS data set containing parameter estimates, standard errors, Wald confidence limits and  $p$ -values. The default value is `_OUTTAB`. The data set contains one observation per explanatory variable or intercept parameter and BY-group. It contains the following variables:

- any BY variables specified
  - `_RBY_`, the ascending rank of the corresponding BY-group
  - `_VAR_`, the subsequent number for each explanatory variable in the `varlist` option or 0 for the intercept parameter
  - `_NAME_`, a character variable of length 8 containing the name of each explanatory variable or INTERCPT
  - `BETA`, the FL parameter estimates
  - `STDERR`, the estimated standard error of the corresponding parameter estimate
  - `CI_LO`, the lower confidence limit for the parameter estimate
  - `CI_UP`, the upper confidence limit for the parameter estimate
  - `P_VALUE`, the  $p$ -value for  $H_0 : \beta_r = 0$ .
  - `_ITER_`, the number of iterations that the model fitting algorithm needed to arrive at the maximum of the penalized log likelihood.
- `global=SAS data set` names a SAS data set containing global likelihood ratio and Wald tests of the hypothesis that  $beta = 0$ . The default value is `_globtest`. The data set contains two observation per BY-group. It contains the following variables:
    - any BY variables specified
    - `_RBY_`, the ascending rank of the corresponding BY-group
    - `testtype`, a numerical variable indicating likelihood ratio test (1) or Wald test (2)
    - `type`, a character variable of length 16 containing the name of the test (“Likelihood ratio” or “Wald”)
    - `ChiSq`, the value of the  $\chi^2$ -statistic
    - `df`, the number of degrees of freedom (the number of independent variables in the model)
    - `P_value`, the  $p$ -value

If the options `odds` is set to 1, then this data set will also contain the following variables:

- `ODDS`, the estimated odds ratio

- `OR_LO`, the lower confidence limit for the odds ratio
  - `OR_UP`, the upper confidence limit for the odds ratio
- `out=SAS data set` creates a new SAS data set that contains all the variables in the input data set and the predicted probability of an event response, the lower and upper confidence intervals for the predicted probability, and the diagonal element of the hat matrix.
    - `pred=variable` assigns a name for the variable in the `out` data set containing the predicted probabilities of an event response. Default value is `_PRED_`.
    - `lower=variable` assigns a name for the variable in the `out` data set containing the lower confidence limit of the predicted probability. Default value is `_LOWER_`.
    - `upper=variable` assigns a name for the variable in the `out` data set containing the upper confidence limit of the predicted probability. Default value is `_UPPER_`.
    - `h=variable` assigns a name for the variable in the `out` data set containing the diagonal element of the hat matrix. Default value is `_H_`.
  - `notes=value` If set to 1, requests notes in the SASLOG about the number of iterations needed for computation of profile likelihood confidence intervals and the profile of the penalized likelihood. Default value is 1.

#### 4.2.4 Options useful for simulation

- `by=variables` requests separate analyses on observations in groups defined by the BY variable(s).
- `offset=SAS data set` names an input data set containing offset values of parameter estimates. This data set should contain the same variables as are specified in `varlist`, plus a variable named `INTERCEP` (unless `noint=1`) and, if the `by`-option is used, the variable(s) specified in this option. Therefore the `offset` data set should have as many observations as there are BY-groups in the input data set. If a particular parameter in a particular BY-group should be estimated, then its value should be missing in the `offset` data set, otherwise the parameter will be treated as fixed at the value found in the `offset` data set. If a variable contained in `varlist`

ist not defined in the `offset` data set, its parameter value will be estimated in any BY-group.

#### 4.2.5 Titles

Titles 1–3 are not used by the macro. These titles can be set by the user in a statement before the macro call. Titles 4 and 5 are used by the macro. These titles are deleted on exit.

### 4.3 Printed output

If `print=0`, `%f1` does not produce any printed output. Otherwise, printed output includes the following:

- an initial page describing the macro and the names of the data set, the dependent and independent variables and any output data sets.
- if the `by` option is not used, a page containing the number of iterations needed to arrive at the maximum of the penalized log likelihood, the value of the penalized log likelihood, and the numbers of events, non-events and total observations
- for each BY-group a table containing the global likelihood ratio and Wald tests
- for each BY-group a table containing, for each parameter, the FL estimate, its estimated standard error, lower and upper confidence limits, and  $p$ -value
- if `odds=1`, for each BY-group a table containing, for each parameter, the FL odds ratio estimate ( $\exp(\beta_r)$ ), lower and upper confidence limits, and  $p$ -value
- if the `test` option is used, an additional page containing the penalized log likelihood  $\chi^2$ , associated degrees of freedom and  $p$ -value for the test that all parameters corresponding to variables in the `test` option are zero

### 4.4 Examples

We exemplify use of the `%f1` SAS macro program by means of the analysis of an endometrial cancer study and thank Dr. E. Asseryanis from the Vienna General Hospital for providing the data set. Purpose of this study of 79 patients was to explain the state of the endometrium by the putative risk factors (=covariates) neovascularization ( $NV$ ),

pulsatility index of arteria uterina (*PI*), and endometrium height (*EH*). The state of the endometrium was histologically graded (*HG*) and classified as 0 (=0-II) and 1 (=III-IV) for 30 and 49 patients, respectively. *NV* is coded as 1 (present) for 13 patients and 0 (absent) for 66 patients. The two continuous covariates *PI* and *EH* range from 0 to 49 and from 0.27 to 3.61, with medians of 16 and 1.64, respectively. Suppose the data has been stored in a SAS data set `bsp.endo`. To obtain an FL analysis with a table containing variable names, parameter estimates, standard errors, profile penalized likelihood confidence limits and *p*-values based on penalized likelihood ratio tests, the macro call

```
title "Analysis of endometrian cancer study";
%fl(data=bsp.endo, y=hg, varlist=nv pi eh);
```

will produce the following four output pages:

Analysis of endometrian cancer study

```
FFFFF L          Logistic regression
F      L          with Firth's bias reduction:
FFF    L
F      L          A solution to the problem of separation
F      LLLLL     in logistic regression
```

```
Author:          Georg Heinze
Version:         2004.07
```

```
Methods published in: Heinze, G. & Schemper, M. (2002). A solution
                      to the problem of separation in logistic
                      regression. Statistics in Medicine 21(16)
                      2409-2419.
```

```
Data set:        BSP.ENDO
Dependent variable: HG
Independent variables: NV PI EH
```

```
Table with parameter estimates saved as _OUTTAB.
Estimates and covariance matrix saved as _OUTEST.
```

Page 2:

Analysis of endometrian cancer study

Model fitting information

NOTE: Penalization computed from standardized data.

Iterations	Penalized log likelihood	Number of responses	Number of nonresponses	Number of observations
8	-24.9225	30	49	79

Page 3:

Analysis of endometrian cancer study

Testing global null hypothesis beta=0

Type	ChiSq	df	P_value
Likelihood Ratio	43.6558	3	<.0001
Wald	17.4797	3	0.0006

Analysis of endometrian cancer study

FL estimates, profile penalized likelihood confidence limits  
and penalized likelihood ratio tests

NOTE: Confidence interval for Intercept based on Wald method.

Variable	Parameter estimate	Standard Error	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
INTERCEP	3.77456	1.48869	0.85672	6.69239	0.0042
NV	2.92927	1.55076	0.60977	7.85456	0.0091
PI	-0.03475	0.03958	-0.12446	0.04046	0.3875
EH	-2.60416	0.77602	-4.36518	-1.23272	<.0001

To obtain estimated odds ratios and corresponding confidence intervals, the macro is called using the `odds` option

```
%fl(data=bsp.endo, y=y, varlist=nv eh pi, odds=1);
```

leading to the following output:



Page 1-4 omitted

Analysis of endometrian cancer study

FL odds ratio estimates, profile penalized likelihood confidence limits  
and penalized likelihood ratio tests

Variable	Odds ratio	Lower 95% c.l.	Upper 95% c.l.	Pr > Chi-Square
NV	18.7140	1.84000	2577.46	0.0091
PI	0.9658	0.88298	1.04	0.3875
EH	0.0740	0.01271	0.29	<.0001

The estimated probabilities of a high histological grading (III-IV) can be obtained by using the out option:

```
%fl(data=bsp.endo, y=hg, varlist=nv eh pi, out=prob, pred=prob,  
    lower=p_lo, upper=p_up);  
proc print data=prob;  
var nv eh pi hg prob p_lo p_up;  
run;
```

The output data set 'PROB' contains predicted probabilities as well as confidence limits:

*Output omitted*

OBS	NV	EH	PI	HG	PROB	P_LO	P_UP
1	0	1.64	13	0	0.27928	0.15998	0.44085
2	0	1.50	28	0	0.24885	0.10130	0.49335
3	0	2.02	29	0	0.07630	0.01839	0.26702
4	0	2.26	16	0	0.06496	0.01860	0.20297
5	0	1.33	11	0	0.48220	0.28478	0.68534
6	0	2.29	15	0	0.06237	0.01727	0.20115
7	0	3.14	8	0	0.00919	0.00074	0.10362
8	0	2.37	19	0	0.04489	0.01041	0.17354
9	0	2.33	12	0	0.06238	0.01637	0.21010
10	0	2.68	34	0	0.01230	0.00101	0.13286

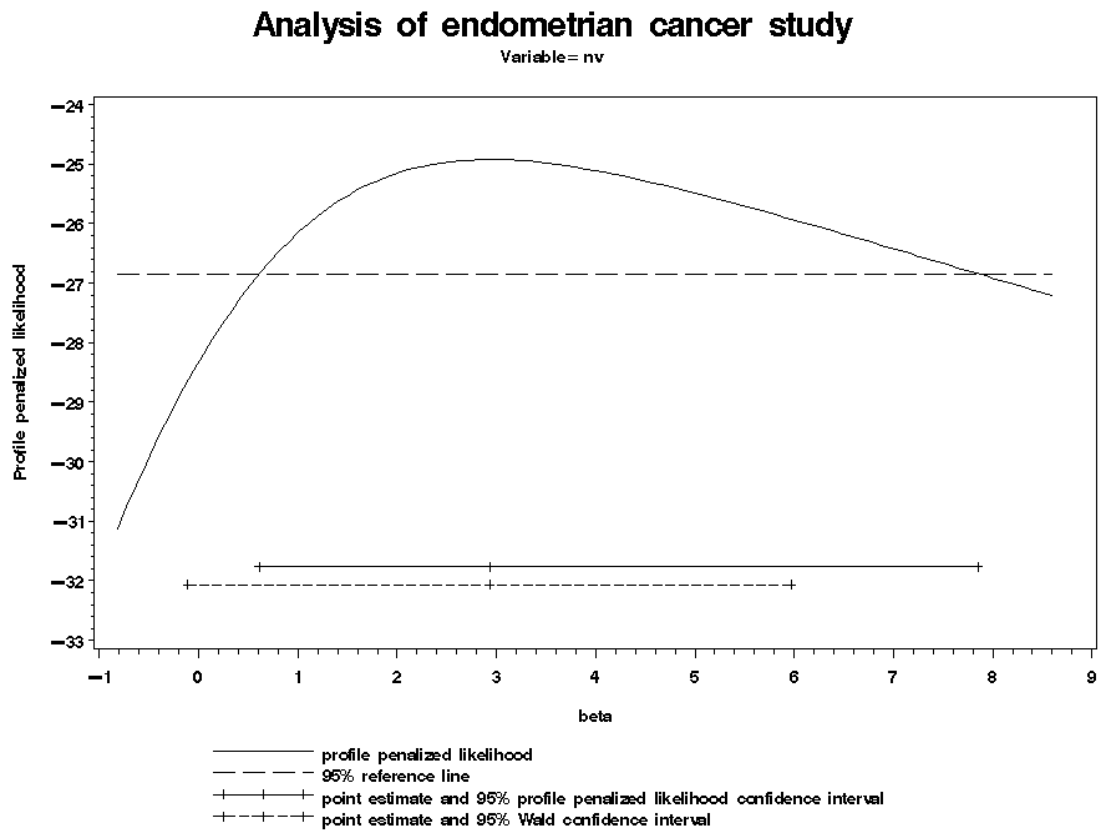
*Output omitted*

The `profile` option can be used to obtain a plot of the profile of the penalized log likelihood for a parameter.

```
%f1(data=bsp.endo, y=hg, varlist=nv eh pi, profile=nv);
```

The values of the penalized log likelihood are plotted against the parameter values of *NV* by the macro (see Fig. 2).

Figure 2: Profile penalized likelihood function for parameter  $\beta_{NV}$  of the endometrian cancer study.



The `test` option can be used to test the simultaneous effect of more than one independent variables on the outcome variable. In our example, to test the hypothesis  $H_0 : \beta_{EH} = \beta_{PI} = 0$  the macro call

```
%fl(data=bsp.endo, y=hg, varlist=nv pi eh,
test=eh pi);
```

is submitted. On the fifth (or sixth) page of output, the following table is displayed:

	Penalized		
Tested	log	Degrees	
variable	likelihood	of	Pr >
(s)	Chi-square	freedom	Chi-square
EH PI	17.8667	2	0.0001

## 4.5 Troubleshooting

Some numerical problems may arise in computation of

- profile penalized likelihood (PPL) confidence intervals, particularly for the intercept parameter. If errors occur, we recommend the following procedure:
  1. Try `p1=0` to see if the problem is in the computation of the confidence intervals. If not, there may be multicollinearity in the design matrix.
  2. Try a smaller value for `maxstep`, 1 or 0.5, say.
  3. Is standardization requested (`standard=1`)?
  4. Try `plint=0`, which prevents computation of a PPL confidence interval for the intercept.
- profile of the penalized likelihood (option `profile`). If errors occur, we suggest to
  1. check if standardization is requested (`standard=1`)
  2. try a smaller value for `maxstep`, 1 or 0.5, say.

## 5 Concluding remarks

### 5.1 Citing

If the method presented here is used, please cite our methodological paper:

- Heinze, G. and Schemper, M. (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine* **21**(16) 2409–2419.

Additionally, you may cite our technical paper:

- Heinze, G. and Ploner, M. (2003). Fixing the nonconvergence bug in logistic regression with SPLUS and SAS. *Computer Methods and Programs in Biomedicine* **71** 181–187.

### 5.2 Firth’s bias reduction method for Cox regression

A similar solution was found for the problem of monotone likelihood in Cox regression, which causes nonconvergence of maximum likelihood parameter estimates. Our approach was published in Heinze & Schemper (2001) and programs described in Heinze & Ploner (2002) are available at the websites

<http://www.akh-wien.ac.at/imc/biometrie/programme/fc> and

<http://www.meduniwien.ac.at/user/georg.heinze/programs/fc>.

## References

- ALBERT, A. & ANDERSON, J. A. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* **71**, 1–10.
- CORDEIRO, G. M. & MCCULLAGH, P. (1991). Bias correction in generalized linear models. *Journal of the Royal Statistical Society B* **53**, 629–643.
- FIRTH, D. (1992a). Bias reduction, the Jeffreys prior and GLIM. In Fahrmeir, L., Francis, B., Gilchrist, R., & Tutz, G., editors, *Advances in GLIM and Statistical Modelling*, pages 91–100. Springer-Verlag, New York.
- FIRTH, D. (1992b). Generalized linear models and Jeffreys priors: an iterative weighted least-squares approach. In Dodge, Y. & Whittaker, J., editors, *Computational Statistics*, volume 1, pages 553–557, Heidelberg. Physica-Verlag.
- FIRTH, D. (1993). Bias reduction of maximum likelihood estimates. *Biometrika* **80**, 27–38.
- FOXMAN, B., MARSH, J., GILLESPIE, B., RUBIN, N., KOOPMAN, J. S., & SPEAR, S. (1997). Condom use and first-time urinary tract infection. *Epidemiology* **8**, 637–641.
- HEINZE, G. (1999). *Technical Report 10: The application of Firth's procedure to Cox and logistic regression*. Department of Medical Computer Sciences, Section of Clinical Biometrics, Vienna University, Vienna.
- HEINZE, G. & PLONER, M. (2002). SAS and SPLUS programs to perform Cox regression without convergence problems. *Computer methods and programs in Biomedicine* **67**, 217–223.
- HEINZE, G. & PLONER, M. (2003). Fixing the nonconvergence bug in logistic regression with SPLUS and SAS. *Computer Methods and Programs in Biomedicine* **71**, 181–187.
- HEINZE, G. & SCHEMPER, M. (2001). A solution to the problem of monotone likelihood in Cox regression. *Biometrics* **57**, 114–119.
- HEINZE, G. & SCHEMPER, M. (2002). A solution to the problem of separation in logistic regression. *Statistics in Medicine* **21**, 2409–2419.
- INSIGHTFUL CORP. (2003). *S-PLUS 6.2*. Reinach, Switzerland.
- PLONER, M. (2001). *Technical Report 2: An S-PLUS library to perform logistic regression without convergence problems*. Department of Medical Computer Sciences, Section of Clinical Biometrics, Vienna University, Vienna.
- SAS INSTITUTE (1999). *SAS/STAT User's Guide, Version 8*. SAS Institute Inc., Cary, NC.
- SCHAEFER, R. L. (1983). Bias correction in maximum likelihood logistic regression.

*Statistics in Medicine* **2**, 71–78.

VENZON, D. J. & MOOLGAVKAR, S. H. (1988). A method for computing profile-likelihood based confidence intervals. *Applied Statistics* **37**, 87–94.