The Stata Journal (2007)
7, Number 3, pp. 376–387

Profile likelihood for estimation and confidence intervals

Patrick Royston
Cancer and Statistical Methodology Groups
MRC Clinical Trials Unit
London, UK
pr@ctu.mrc.ac.uk

Abstract. Normal-based confidence intervals for a parameter of interest are inaccurate when the sampling distribution of the estimate is nonnormal. The technique known as *profile likelihood* can produce confidence intervals with better coverage. It may be used when the model includes only the variable of interest or several other variables in addition. Profile-likelihood confidence intervals are particularly useful in nonlinear models. The command `pllf` computes and plots the maximum likelihood estimate and profile likelihood–based confidence interval for one parameter in a wide variety of regression models.

Keywords: st0132, pllf, profile likelihood, confidence interval, nonnormality, nonlinear model

1 Introduction

Venzon and Moolgavkar (1988) inspired this article, and the next two paragraphs briefly summarize their approach. The standard method of confidence interval (CI) construction is based on the asymptotic normality of the maximum likelihood estimate (MLE) $\widehat{\theta}$ of a parameter vector θ_0. However, properties of $\widehat{\theta}$ in small samples can be different from the asymptotic properties. When, for example, θ_0 is a scalar parameter, the usual $(1-\alpha)\,\%$ CI is given by $\widehat{\theta} \pm t_{1-\alpha/2}s$, where $t_{1-\alpha/2}$ is the $(1-\alpha/2)$th quantile of the normal or t distribution and s is the standard error of $\widehat{\theta}$. In *well-behaved* cases when asymptotics nearly enough apply, the coverage of θ_0 by the CI over repeated samples from the population will be approximately $1-\alpha$. A fraction $\alpha/2$ of the sample of $\widehat{\theta}$ values will exclude θ_0 symmetrically at each end of the interval. In *badly behaved* cases, the coverage may be far from $1-\alpha$ and may also be unequal at each extreme. Such asymmetric coverage occurs when the sampling distribution of $\widehat{\theta}$ is nonnormal, e.g., skewed.

A construction of confidence regions that is likely to be more robust in small samples may be derived from the asymptotic χ^2 distribution of the likelihood-ratio test statistic. Suppose that $\dim(\theta) = k \geq 1$ and that the log-likelihood $l(\theta)$ is defined for values of θ in a suitable k-dimensional parameter space. An approximate $(1-\alpha)\,\%$ confidence region for θ_0 is the set of values of θ satisfying

$$\left[\theta : 2\left\{l\left(\widehat{\theta}\right) - l(\theta)\right\} \leq c_{k;1-\alpha}\right] \tag{1}$$

where $c_{k;1-\alpha}$ is the $(1-\alpha)$th quantile of the χ^2 distribution on k degrees of freedom. CIs for individual components of θ_0 may be defined similarly, as in Cox (1970, 88). This profile-likelihood method reduces $l(\theta)$ to a function of one parameter β of interest by treating the other components of θ as nuisance parameters and maximizing the likelihood over them. See Venzon and Moolgavkar (1988) for more details.

Let β, an element of θ, be a scalar parameter of special interest. We wish to construct a profile-likelihood function of β. Let the population (true) value of β be β_0 and the profile log-likelihood (PLL) function of β be $l_p(\beta)$. The other components of θ are not explicitly mentioned in the PLL. In practice, all that is required to compute $l_p(\beta)$ and the corresponding likelihood-based CI for β_0 is to fix the value of β, find the MLE of the remaining components of θ, and evaluate $l_p(\beta)$. The process is repeated over a suitable grid of values of β until $(\beta_{\text{left}}, \beta_{\text{right}})$ are found nearly enough satisfying (1) for $\theta = \beta$, that is, satisfying

$$2\left\{l\left(\widehat{\beta}\right) - l_p\left(\beta_{\text{left}}\right)\right\} = 2\left\{l\left(\widehat{\beta}\right) - l_p\left(\beta_{\text{right}}\right)\right\} = c_{1;1-\alpha}$$

By definition, $l\left(\widehat{\beta}\right) = l_p\left(\widehat{\beta}\right)$. The command `pllf` aims to compute $(\beta_{\text{left}}, \beta_{\text{right}})$ within a multiparameter model. The special case that β is the only adjustable parameter is supported. Furthermore, `pllf` plots $l_p(\beta)$ over a suitable range of values, defined automatically or specified by the user.

Using a variant of the syntax, `pllf` can alternatively compute the MLE and a PLL-based CI for one nonlinear parameter in a general regression model. Such a model is typically conditionally linear. A specific example is the well-known exponential model, e.g., $E(y) = \beta_0 + \beta_1 \exp(\gamma x)$, where β_0, β_1, and γ are parameters to be estimated. Writing the model as $\beta_0 + \beta_1 x^*$ with $x^* = \exp(\gamma x)$ gives a model linear in x^* but nonlinear in x. `pllf` can provide the MLE and PLL-based CI for γ not only in the normal-errors regression setting just illustrated but also for a wide variety of the many regression models implemented in Stata.

2 Syntax

`pllf` *regression_cmd regression_cmd_stuff* [*if*] [*in*] [*weight*] ,
`profile(`*xvarname* | [`[`*eqname*`]`] *paramname*`)` | `formula(`*formula*`)`
[`deviance gen(`*beta_var pll_var*`) difference level(`*cilevel*`) range(`*#1 #2*`)`
`maxcost(`*#*`) n(`*#*`) noci nodots nograph gropt(`*cline_options twoway_options*`)`
`levline(`*cline_options*`) cilines(`*cline_options*`) placeholder(`*string*`)`
regression_cmd_options]

where, in essence, any *regression_cmd* for which the parameters are estimated by maximum likelihood may be used. This includes `clogit`, `cnreg`, `glm`, `heckman`, `logistic`, `logit`, `mlogit`, `nbreg`, `gnbreg`, `ologit`, `oprobit`, `poisson`, `probit`, `regress`, `reg3`, `stcox`, `streg`, and `stpm` (Royston 2001, Royston and Parmar 2002), among others.

`pllf` has two basic syntaxes, depending on which option, `profile()` or `formula()`, is used. Let us call these syntaxes 1 and 2. See section 3 for more information on what these two syntaxes of `pllf` actually do.

With syntax 1, `profile()` must be specified. *regression_cmd_stuff* typically takes the simple form [*depvar*] *varlist*, although more complex syntax is supported according to the needs of *regression_cmd*. For example, for *regression_cmd* `ivregress`, *regression_cmd_stuff* takes the form *depvar* [$varlist_1$] (*varlist*$_2$ `=` $varlist_{\text{iv}}$).

With syntax 2, `formula()` and `range()` must be specified. *regression_cmd_stuff* is similar to that for syntax 1, except that *regression_cmd_stuff* must include the placeholder, which by default is `X`. This is substituted by a variable calculated according to the formula defined by `formula()`, which must also include the placeholder at least once.

All weight types supported by *regression_cmd* are allowed.

3 Description

`pllf` with the `profile()` option (syntax 1) computes the PLL function for the regression coefficient of a covariate *xvarname* defined by `profile(`*xvarname*`)` or of a parameter or a variable defined by `profile(`[`[`*eqname*`]`] *paramname*`)` within a model specified by *regression_cmd*, *regression_cmd_stuff*, and *regression_cmd_options*. Where possible, `pllf` reports PLL-based confidence limits, computed by a simple grid search. For the simple syntax *regression_cmd* [*depvar*] *varlist, xvarname* need not be a member of *varlist*, although including it is harmless. The results are saved to new variables assigned by the `gen()` option. The dataset length is increased if `n()` exceeds the current number of observations (`_N`).

`pllf` with the `formula()` option (syntax 2) computes the PLL function of a nonlinear parameter denoted by `X`. `X` is symbolically included where necessary in *regression_cmd_stuff*. In effect, `X` is replaced on the fly by the variable created by substituting the current value of `X` in *formula*. `pllf` reports the MLE and PLL-based confidence limits, computed by a simple grid search. Normal-based confidence limits are not computed. Other features are similar to those with syntax 1.

4 Options

`profile(`*xvarname* | [`[`*eqname*`]`] *paramname*`)` (syntax 1) is required. In the first format, the PLL function for the regression coefficient for *xvarname* is calculated. *xvarname* is a covariate in the main response model. In the second format, the PLL function for the parameter defined by [`[`*eqname*`]`] *paramname* is calculated. Typically, *paramname* will be an auxiliary parameter of some kind, such as a scale or shape parameter, with its own equation. For example, for the Weibull model, `profile([ln_p]_cons)` would give the PLL function for the log of the shape parameter, `p`.

`formula(`*formula*`)` (syntax 2) is required. *formula* defines a transformation involving at least one variable in the dataset and the parameter `X`. *formula* may be any valid Stata expression, i.e., `formula(exp(-X*x5))`.

`deviance` specifies the profile-deviance function, i.e., -2 times the PLL function. If `difference` is also specified, `deviance` produces the profile-deviance *difference*, i.e., -2 times the PLL difference.

`gen(`*beta_var pll_var*`)` creates two new variables: *beta_var* to contain the values of the regression coefficient over which the PLL is evaluated and *pll_var* to contain the PLL values. If `gen()` is not specified, the variables are created with default names of `_beta` and `_pll`, respectively.

`difference` computes the PLL function minus the maximized log likelihood for the model. See also the `deviance` option. Except in pathological cases, the resulting values are negative or zero. Pathological cases denote likelihood functions with multiple maximums or no maximum.

`level(#)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **20.7 Specifying the width of confidence intervals** (Stata 10) or [U] **20.6 Specifying the width of confidence intervals** (Stata 9).

`range(`*#1 #2*`)` with syntax 1 evaluates the PLL function over $\#1 \leq \beta \leq \#2$, where β is the regression coefficient for *xvarname*. The default is for *#1* and *#2* to be the confidence limits for β defined by the option `level()` and the usual assumption of a normal distribution for the MLE of β.

`range(`*#1 #2*`)` with syntax 2 is required, and it evaluates the PLL function over $\#1 \leq \texttt{X} \leq \#2$, where `X` is the nonlinear parameter of interest. `pllf` also seeks the MLE of `X`, but if the values of *#1* and *#2* are ill-chosen or the PLL function behaves *badly*, it may fail to find the MLE or give an inaccurate estimate. The most satisfactory situation is when the MLE lies between *#1* and *#2*, and this may be judged from the plot of the PLL function. Particularly with large sample sizes, the PLL function is often approximately quadratic with one maximum.

`maxcost(#)` sets an upper limit of $2 \times \#$ on the number of additional evaluations of the PLL when searching for the PLL-based confidence limits. You should rarely, if ever, need this option. `maxcost()` prevents the program from cycling forever when trying to find confidence limits in pathological cases (see the `difference` option). The default $\#$ is `n()`/2.

`n(#)` evaluates the PLL function at $\#$ equally spaced points. The default is `n(100)`.

`noci` suppresses calculation of the PLL-based confidence limits.

`nodots` suppresses dots. By default, the program displays a dot at each evaluation of the PLL.

`nograph` suppresses the line plot of the results.

`gropt(`*cline_options twoway_options*`)` supplies graph options to enhance the plot of the PLL (or a transformation of it) against β or `X`. The default graph is a line plot showing the PLL-based CI for β as vertical lines parallel to the y axis and the corresponding PLL value (or a transformation of it) as a horizontal line parallel to the x axis. Appropriate linear transformation of the PLL is applied when the `deviance` or `difference` option is specified.

`levline(`*cline_options*`)` specifies the rendition of the horizontal line showing the profile likelihood at the confidence level for the PLL-based CI.

`cilines(`*cline_options*`)` specifies the rendition of the vertical lines representing the bounds of the PLL-based CI.

`placeholder(`*string*`)` defines the placeholder character(s) used in syntax 2. Spaces or other punctuation characters are not allowed. The default *string* is `X` (capital).

regression_cmd_options may be any of the options appropriate to *regression_cmd*.

5 Example 1

We will use the breast cancer dataset, which Sauerbrei and Royston (1999) analyzed. The data are provided in `brcancer.dta` and relate to a set of 686 patients with lymph node–positive breast cancer. The outcome of interest is the recurrence-free survival time, that is, the duration in years from entry into the study (typically, the time of diagnosis of primary breast cancer) until either death or disease recurrence, whichever occurred first. There were 299 events for this outcome and the median follow-up time was about 5 years.

These authors derived a Cox proportional hazards model for recurrence-free survival time that included five covariates: age (`x1`) with a fractional polynomial transformation with powers -2 and -0.5, tumor grade 2/3 (`x4a`), number of positive lymph nodes (`x5`) with the exponential transformation `x5e` $= \exp(-0.12 \times$ `x5` $)$, progesterone receptors (`x6`) with a fractional polynomial transformation with power 0.5, and hormonal therapy with tamoxifen (`hormon`). The commands to obtain a PLL-based CI for the covariate `x4a` within the above-mentioned model are as follows:

```
. use brcancer
. stset rectime censrec
. fracgen x1 -2 -0.5
. fracgen x6 0.5
. pllf stcox x1_1 x1_2 x4a x5e x6_1 hormon, profile(x4a) gropts(saving(fig1))
```

Figure 1 shows the resulting graph.

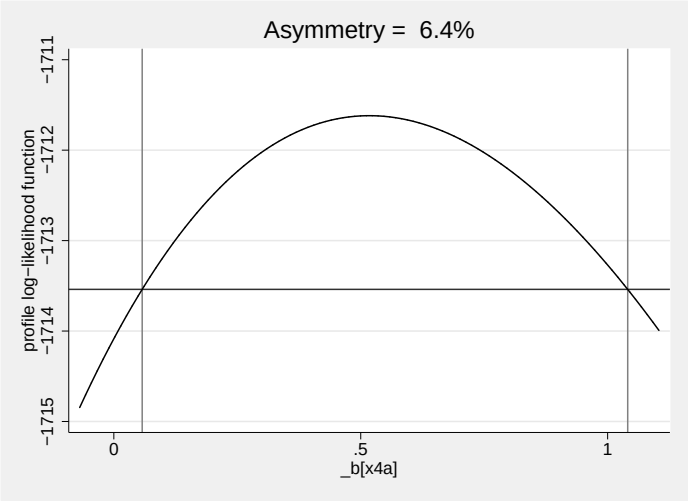

Figure 1: PLL function of the regression coefficient for `x4a` in a multivariable Cox model for the breast cancer data

The horizontal line represents the χ^2 quantile $c_{1;1-\alpha}$ for $1-\alpha = 0.95$, linearly transformed to the scale of the log likelihood for the model. (Since a Cox model is fitted, we use the partial log likelihood rather than the log likelihood here.) In this example, $\widehat{\beta} = 0.517$ and $l_\beta\left(\widehat{\beta}\right) = -1{,}711.62$. The horizontal line is drawn at $-1{,}711.62 - 3.84/2 = -1{,}713.54$, where $c_{1;1-\alpha} = 3.84$ is the 95th centile of χ^2 on 1 degree of freedom. The vertical lines show the 95% PLL-based confidence limits, which are $(0.057, 1.041)$. They should be compared with the normal-based 95% CI, which is $(0.029, 1.006)$. Although the difference between the two CIs is not huge, it is not nothing either, demonstrating that even in a reasonably large sample (686 patients, 299 events) PLL-based CIs may have something to offer.

The tails of the curve in figure 1 extend unequally on both sides of the CI because of how `pllf` selects its grid of β values. The terminals of the grid are taken as $\widehat{\beta} \pm 1.2t_{1-\alpha/2}s$, the idea being that a normal-based CI stretched by 20% should (and usually does) cover the PLL-based CI. The asymmetry of the PLL-based CI about $\widehat{\beta}$ in this example causes the unequal tail lengths of the profile-likelihood curve. One could achieve a more visually appealing plot by a better choice of terminals through the `range()` option.

5.1 Asymmetry of the PLL-based CI

A convenient measure of nonnormality of the sampling distribution of $\widehat{\beta}$ is the asymmetry, say, A, of the PLL-based CI. A reasonable definition of this measure is length of upper arm minus length of lower arm, divided by length of CI. Multiply the measure by

100 to express it as a percentage. By definition, a normal-based CI has no asymmetry, so $A = 0$. The asymmetry of the CI for `x4a` is

$$A = 100 \times \{(1.041 - 0.517) - (0.517 - 0.057)\} / (1.041 - 0.057) \simeq 6.5$$

i.e., the difference in arm length is about 6.5% of the CI length.

5.2 Effect of sample size

Figure 2 shows how the PLL and normal CIs change as the sample size increases.

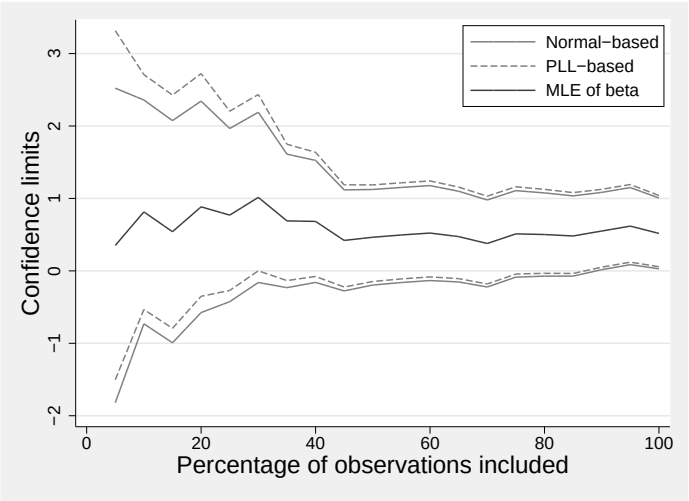

Figure 2: Comparison of normal- and PLL-based CIs for β for `x4a`. The sample size is varied between 5% and 100% of the original sample of 686 patients.

The data were sorted in random order and the first 5%, 10%, ..., 100% of the observations used in the PLL calculation. The plot shows that although the width of the two types of CI is similar, the asymmetry is different. In effect, the PLL-based CIs are shifted upward by an amount that increases with reduced sample size. Figure 3 shows the asymmetry statistic A.

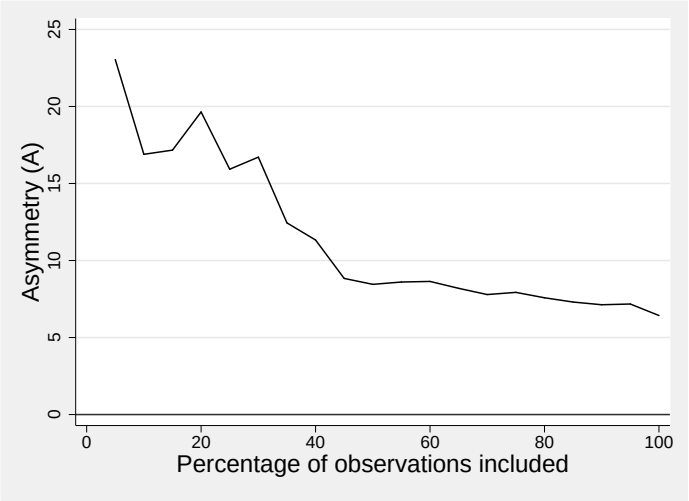

Figure 3: Asymmetry (A) of the PLL-based CI for `x4a` as a function of the proportion of observations included in the analysis

When only 5% of the observations are included, difference in arm lengths is nearly 25% of the total arm length, which is substantial. Figure 2 shows that in absolute terms, the two types of CI in this example differ markedly only when the small sample is less than about 40% of the original 686. The asymmetry is then greater than 10%.

6 Example 2

We will use the breast cancer dataset again to illustrate the use of syntax 2 of `pllf` to obtain a CI for a parameter in a nonlinear regression. Sauerbrei and Royston (1999) reported the value of γ in the negative exponential transformation $\texttt{x5e} = \exp(-\gamma \times \texttt{x5})$ to be 0.12 but gave no standard error or CI for γ. One can obtain a PLL-based CI for γ within a Cox regression model for the data by using `pllf`. Let us suppose that we have no idea of the MLE of γ or its CI. We choose a wide initial range, e.g., $(-1, 1)$, for γ and proceed as follows:

```
. use brcancer
(German breast cancer data)
. pllf stcox X, formula(exp(-X*x5)) range(-1 1)
.....................................................................
> ............
```

_t	Coef.	Std. Err.	[95% PLL Conf. Int.]	
X	.1183861	.038756	.0619803	.213901

Note: Std. Err. is pseudo standard error, derived from PLL CI

Figure 4 shows the resulting plot.

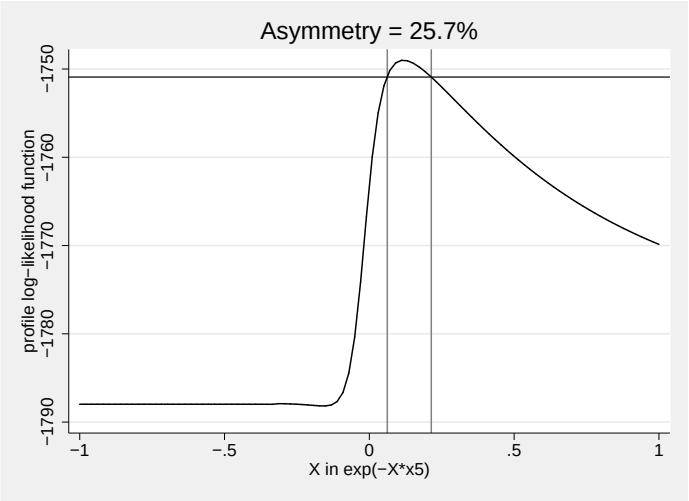

Figure 4: PLL function for the nonlinear parameter γ in a Cox regression on $\exp(-\gamma \times x5)$ for the breast cancer dataset, derived from `pllf` using the option `range(-1 1)`. Vertical lines show the PLL-based 95% CI. Since the curve is far from quadratic, the distribution of $\widehat{\gamma}$ is clearly far from normal.

The overall shape of the PLL function is far from quadratic, but on the basis of the wide range of PLL values on the vertical axis, the required range has been overshot by some considerable margin. The MLE of $\widehat{\gamma}$ is reported as 0.118 with a PLL-based CI of $(0.062, 0.214)$. With this in mind, we repeat the command with a narrower range of, say, 0.05–0.25:

```
. pllf stcox X, formula(exp(-X*x5)) range(.05 .25)
.................................................
```

_t	Coef.	Std. Err.	[95% PLL Conf.	Int.]
X	.1174223	.0392578	.0603041	.2141917

Note: Std. Err. is pseudo standard error, derived from PLL CI

The MLE and CI change a little, to 0.117 and $(0.060, 0.214)$, respectively. The asymmetry A is 25.8%, which is large; you can clearly see it in figure 5.

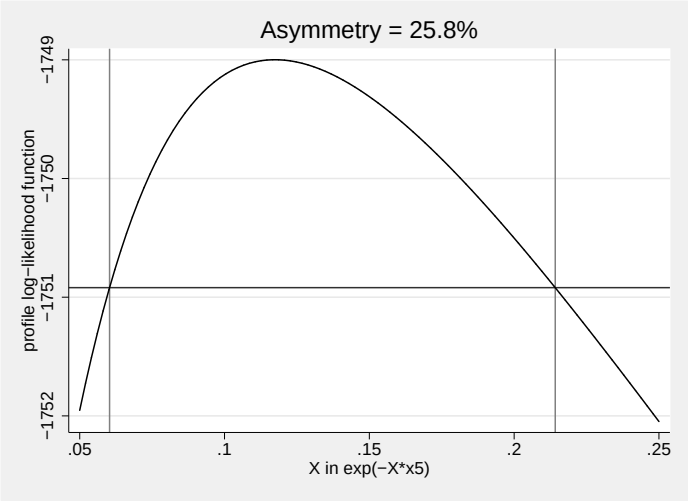

Figure 5: PLL function for the nonlinear parameter γ in a Cox regression on $\exp(-\gamma \times x5)$ for the breast cancer dataset, derived from `pllf` using the option `range(0.05 0.25)`. Vertical lines show the PLL-based 95% CI. Even within this restricted range and with the rather large sample size, the distribution of $\widehat{\gamma}$ is not normal.

Normal-based confidence limits would be inaccurate in this example.

Finally, a reviewer suggested that it would be useful if `pllf` could also compute the score function, that is, the first derivative of the log likelihood evaluated at β. Since $l_p(\beta)$ is a smooth function of β, this calculation can be done with good accuracy by using the standard Stata program `dydx`, which calculates first derivatives. For example, to compute and plot the score function for the above example, one could code

```
. pllf stcox X, formula(exp(-X*x5)) range(.05 .25)
. dydx _pll _beta, gen(score)
. line score _beta, sort xline(.1174) yline(0) ytitle("Score function")
```

Figure 6 shows such a plot.

(*Continued on next page*)

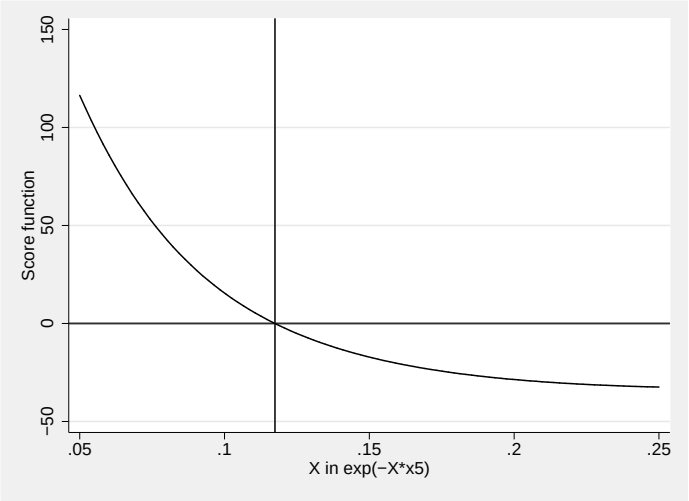

Figure 6: Score function for the nonlinear parameter γ in a Cox regression on $\exp(-\gamma \times x5)$ for the breast cancer dataset. The horizontal line shows $y = 0$, and the vertical line shows the MLE of γ.

If the parameter estimate were normally distributed, the plot would be linear. Such is clearly not the case here.

7 Comments and conclusions

Both examples indicate that using profile likelihood improves on normal-based CIs. The main reason is that the likelihood-ratio statistic tends to approach its asymptotic χ^2 distribution more rapidly than the equivalent Wald statistic. One may also view `pllf` as a convenient tool to check the validity of the normal assumption in critical cases. In principle, one could do so also by using the bootstrap. However, for CI calculations not assuming normality, the bootstrap can become computationally expensive. The bootstrap is usually assumed valid, even in small samples, but that may not be so; the bootstrap gives only asymptotically correct results. Also, in applications in nonlinear modeling for which syntax 2 of `pllf` is required, computing the MLE of a nonlinear parameter in each bootstrap sample may not be straightforward.

To prove the worth of the PLL approach empirically would require extensive simulation studies. For present purposes, relying on theoretical arguments on the superiority of PLL-based CIs suffices.

8 Acknowledgments

I thank Vincent L. Wiggins, StataCorp, and Ian White, MRC Biostatistics Unit, Cambridge, for helpful comments and encouragement in this project, and an anonymous reviewer for suggestions that helped me to improve the manuscript and the software.

9 References

Cox, D. R. 1970. *Analysis of Binary Data*. London: Methuen.

Royston, P. 2001. Flexible parametric alternatives to the Cox model, and more. *Stata Journal* 1: 1–28.

Royston, P., and M. K. B. Parmar. 2002. Flexible parametric proportional-hazards and proportional-odds models for censored survival data, with application to prognostic modelling and estimation of treatment effects. *Statistics in Medicine* 21: 2175–2197.

Sauerbrei, W., and P. Royston. 1999. Building multivariable prognostic and diagnostic models: Transformation of the predictors using fractional polynomials. *Journal of the Royal Statistical Society, Series A* 162: 71–94.

Venzon, D. J., and S. H. Moolgavkar. 1988. A method for computing profile-likelihood-based confidence intervals. *Applied Statistics* 37: 87–94.

About the author

Patrick Royston is a medical statistician with nearly 30 years' experience, with a strong interest in biostatistical methods and in statistical computing and algorithms. He now works in cancer clinical trials and related research issues. Currently, he is focusing on problems of model building and validation with survival data, including prognostic factor studies; on parametric modeling of survival data; on multiple imputation of missing values; and on new trial designs.