Referee report of “The Current State of R Tools for Nonlinear Least Squares Modeling”.

I appreciate the effort that the authors have made in this revision. I believe the paper has been considerably improved.

I think it would useful to expand a bit on some of the comments in the section “An illustrative example”. In particular, the choice of starting estimates, at least for the third formulation of the model, doesn’t make sense to me.

Purpose of nonlinear least squares fits

In contrast to linear least squares or techniques like generalized additive models (GAMs), which provide empirical or data-driven models, I feel that the goal of a nonlinear regression analysis, at least in part, centers on the parameter estimates and the interpretation of the parameters. That is, the model is usually derived from a mechanistic model of how the data are generated, this model depends on certain parameters, and the estimates of those parameters are of interest in and of themselves.

If the parameters are of interest, the analyst should be able to “ball park” the values of the parameters from the data, probably aided by plotting the data.

I feel there are three stages in a nonlinear least squares fit:

1. Determine starting estimates of the parameters from the data and knowledge of the interpretation of the parameters.
2. Starting at these estimates use an iterative algorithm that seeks to reduce the sum of squared residuals at each iteration.
3. Determine if convergence has been achieved at the current iteration and, if not, go to step 2.

Admittedly, the vast majority of research and software development is focussed on step 2 but steps 1 and 3 are also important and an analyst performing a nonlinear regression analysis should be able to formulate starting estimates for a model.

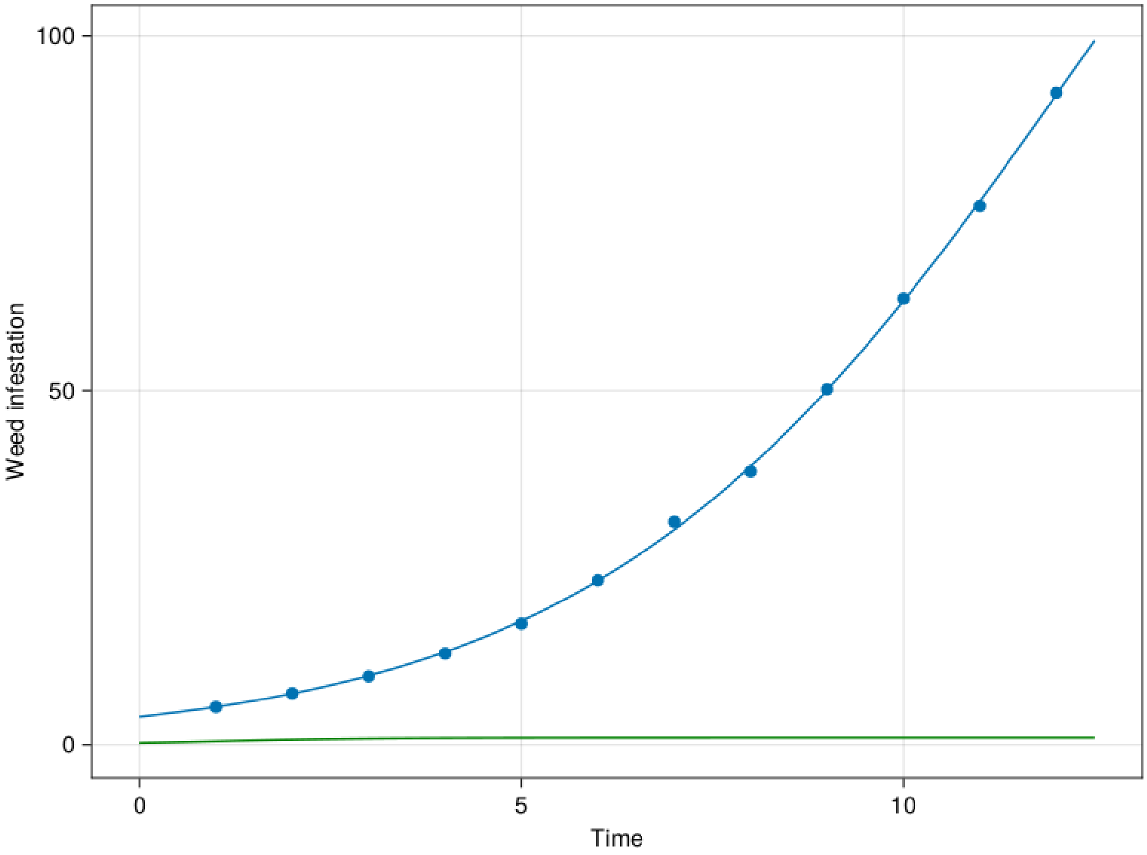

Figure 1: Hobbs weed infestation data with fitted 3-parameter logistic curve (blue) and curve from the starting estimates in third parameterization of the model (green).

Interpretation of parameters in the three-parameter logistic model

The logistic growth model is based on vertically scaling and possibly shifting the cumulative distribution function of the logistic distribution. If the lower limit of the predicted response is set at zero, there is only a vertical scaling constant which determines the asymptotic predicted response as $x \to \infty$. The three-parameter logistic model incorporates this vertical scaling parameter with location and scale parameters on the horizontal axis. (The four-parameter logistic model incorporates both shift and scale parameters on the vertical axis.)

The common way of writing location/scale parameters on the horizontal axis results in the location being the value on the x-axis of the inflection point (when the fitted value is midway between the lower and upper vertical limits) and the scale being the x-axis distance between the midpoint and the point where the fitted value is roughly one-quarter of the distance from the lower limit and the upper vertical limits.

It is straightforward to look at a data plot, such as Figure 1, and choose appropriate starting values for the parameters in this representation of the three-parameter logistic.

However, one does not look at this plot and decide that $(1, 1, 1)$, which would give the green line in the plot, is a reasonable set of parameter values for this model applied to these data.

As we can see the fitted curve is a very good fit to these data but the curve from the unit starting estimates is not even close to the observed data. It doesn't even overlap on the y-axis with the observed data.

So these are egregiously bad starting estimates for the parameters. The authors state that fitting this model is very tricky but in my previous report I showed that is was trivial to fit this model using the self-starting model function `SSlogis`, because it goes through steps of using the data to obtain good starting values.

A good portion of this paper is studying whether different nonlinear least squares software implementations can converge to appropriate parameters estimates from these ridiculously bad starting values and, quite frankly, I don't see the point.

It is the author's prerogative to use whatever criterion for comparison of nonlinear least squares implementations that they feel is appropriate but I think they should clarify to the reader that they are using intentionally bad and inappropriate starting estimates as a way of stressing algorithms.

On page 4 the authors state that if a selfStart model is not available `nls()` sets the starting values to 1. That may be the case now but the original intention was certainly that a user calling `nls()` would provide starting estimates suitable to the model and data.

Also on page 4 I don't understand the comment in the `hobbs.res` function that `This variant uses looping`. I don't see any explicit loops. There will naturally be looping in the C code that implements the arithmetic operations on vectors but that is inevitable.

Is it really intended that anyone using the function-based specification of the model will encode their data into the model function like `hobbs.res` and `hobbs.jac`? Generally we would expect to separate the specification of the data from the model - otherwise there is a great risk of, say, the residual and the Jacobian function accidentally getting out of sync.

Also, this is the first time that the authors have invoked their non-standard definition of "residual" as the negative of what most others mean by "residual", and they haven't mentioned this. I would discourage the authors from using an idiosyncratic definition of "residual" simply to save writing a negative sign in a formula for the Jacobian.