Model frames and the Survival package

Terry M Therneau

June 4, 2026

The modeling functions in the survival package (aareg, coxph, pyears, survexp, survfit, and survreg) differ from almost every other package in R by having `model=FALSE` as the default. This has two major consequences for a user.

- When a follow up computation arises that needs something else from the data frame, something that was not saved, the routine will need to rebuild the model frame. Examples are a survival curve after a coxph model, and certain predicted values and residuals.
- In some cases, the data cannot be reconstructed.
  - The most common is when the data can't be found, itself usually a complex function of how R searches for things. This often occurs, for instance, if a call to coxph is within another function, and the coxph formula uses both local variables and variables via the `data=` option.
  - The data might be gone (or worse, changed).

Personally I only infrequently get caught by this. The solution is quite simple, which is to refit the model adding the `model=TRUE` option. Some more modern flavors of R may have this arise more often, I do not know.

Why have I chosen this route? First, let me point out that I didn't actually "do" anything. For the first half of the survival package's life lm() and glm() did exactly the same thing. And I will admit that inertia (and a bit of stubborness) is certainly one of the reasons for standing pat.

However, by far the largest motivation for not changing is confidentiality. I work in a major medical center (Mayo Clinic) on medical research, using real patient data. We take the issue of data confidentiality very seriously, and are quite careful with respect to where the data is stored and who has access to it. (One certain way to be fired at Mayo, and by that I mean "walked to the exit by security on that same day", is to inappropriately access or share patient data.) Yet each saved copy of an lm() (or random forest or whatever) model in R carries a silent, complete, and unencrypted copy of the data used to fit it; something of which most users remain blissfully unaware. If a model has per subject random effects or a robust variance, then `id clinic` was likely part of the call, i.e., each patient's personal identifier in our history system is also in the model data.

This makes me very nervous. Observing the increasingly sophisticated attacks on our institution's IT structure only adds to it. Perhaps my stance on the survival package is only one finger in a very leaky dam, but I'm not ready to join the problem. I am also fatalistic in suspecting

that whomever takes on this package when I step away will likely make this one of their first changes. (I'm 73, so not too far away.)

The second argument against change is size. I first caught on to this over a decade ago when I got a message that I was consuming far too much of our deparment disk space, which puzzled me. It turned out that I had been fitting several exploratory models to a very very large data set and saving the results (not survival models) and my .RData had exploded. Given the constant increase in available data storage this argument is less cogent than it once was, but I started my career scratching for every byte, and limited consumption is a hard habit to break once pounded in so deeply.