

Re: regression assumptions are violated, what next?

Source: <http://sci.tech-archive.net/Archive/sci.stat.math/2008-07/msg00242.html>

- *From:* Paul Rubin <rubin@xxxxxxx>
 - *Date:* Sat, 12 Jul 2008 12:31:07 -0400
-

Vlad Skvortsov wrote:

Hmm, in my case I'm going to use the model to simulate the system operation. So I'm **assuming** I need to know the residuals distribution, since the delay introduced by the component (in simulation) will be something like (for a linear model case):

$$d = \text{intercept} + \text{slope} * \text{illen} + \text{err}$$

where 'err' should come from residuals distribution. It doesn't have to be normal, of course (though it would simplify things).

It doesn't have to be linear, either. You just need $d = f(\text{illen}) + \text{err}$, or maybe $d = f(\text{illen}) * (1 + \text{err})$, with a known function f and a known distribution for err . (I would advocate the second form if there is evidence that residual variance increases with d , otherwise the first form.)

So my primary concern here is to get the regression coefficients right, and if I understood you correctly, it may be possible even if the residuals normality property is violated. Is that right?

Yes. More precisely, the OLS fit (the fit that minimizes squared error) is defined with no distributional assumptions, and ordinary regression finds it regardless of whether the errors are normal or not. If the errors are normal (and the observations are independent), the OLS fit also happens to be the maximum likelihood estimator (MLE). If the errors are not normal, the OLS and MLE fits typically are different, and to find the MLE fit you need to make an assumption about the error distribution (and use a nonlinear regression routine).

If yes,
how do I ensure that the coefficients are reasonable (apart from looking at the plot and checking that the confidence intervals are narrow enough for my situation)?

Re: regression assumptions are violated, what next?

If the model generates pattern-free residuals, I'd call it reasonable. A useful approach, particularly given your sample size, is to split the sample into disjoint fitting and validation samples. Estimate whatever model you settle on using just the fitting sample, then compute residuals for that model on the validation sample. Plot the validation residuals v. illen (hoping for no patterns), test the validation residuals for homoscedasticity (constant variance with respect to illen) and constant mean with respect to illen, and then just take your chances. When you split the sample, try to get roughly equal representations of every portion of the range of illen in both samples.

Regarding the residual plot, which is indeed a bit scary, I suggest you scatter plot your data separately for three ranges, say $illen \leq 600$, $600 < illen \leq 1000$, and $illen > 1000$. As best I can see, the overall

Just curious: when selecting subsets of data like this, am I expected to provide some kind of justification on choosing the splitting points? Or saying that "visually the system behaviour changes around this and that point" is sufficient?

Depends on the referee. :-) IIRC, in simulation there are various criteria for validity ("face validity" is the only term that comes to mind just now), and accuracy tracking historical results is one of them. If you can show that your random processing time generator produces values that match the historical data (for the same values of illen), you should be ok. You might try a version of a Turing test. After settling on a model, fitting it, picking a distribution for the error term and coding your generator, generate a new sample with the same values of illen as in the historical sample, but simulated values of processing time. Treat the new sample and the historical sample as two samples from bivariate distributions, and test whether the distributions are the same. (There are a number of tests you can apply. Pick one or two.) Alternatively, add an indicator variable that's coded 1 for simulated and 0 for historical, then see if you can predict the indicator using the values of illen and total (for instance, using a logistic regression model, or a discriminant function). If you can't tell the two samples apart, your generator should be good enough.

I suspect this heavy-tailedness might stem from the following. The machine that the system runs on occasionally performs some background processing which might introduce delays on its own. Thus the processing time, of course, can only get greater, and never smaller.

Maybe. It looked to me that the heavy tail showed up more at the low end of illen than at the high end, though. If the tail is from a source unrelated to the job at hand, it should manifest uniformly across the range of illen values.

Now, the next question is why do we see more outliers in the first range?

Re: regression assumptions are violated, what next?

Re: regression assumptions are violated, what next?

Exactly.

Looking at the data, there were 21470 requests shorter than 600 and about 12 outliers in that range; for request lengths 600–1000: 340 requests and 1 outlier; for 1000+: 120 requests and 0 outliers. So assuming that those extra load spikes on the processing machine are exponentially distributed, it might be that we observe more outliers for shorter request lengths just because there are so much more of them.

True, but if that (and your previous explanation) are the reasons for the outliers, then the outliers we do see in the higher illen range, while fewer in number, should be comparable in terms of deviation from the regression line. I don't think they are, although it's a bit hard to be certain given the decreasing frequency.

Does this sound reasonable and if yes, could I use these grounds to just reject the outliers data from the analysis?

Your explanation is plausible; I'm not sure it's entirely convincing (skepticism explained above). If the source of the outliers is indeed exogenous (such as the processor taking a coffee break), then for purposes of estimating/simulating the actual processing time of a job, it's appropriate to weed out the outliers before doing the estimation. On the other hand, if the source of those long total times is endogenous (occasionally a job is small but funky and needs a lot of extra time — for instance, it needs to access offline storage, or it writes to a tape drive rather than a disk drive), then no, weeding the "outliers" biases the estimation.

Let's say, for the sake of argument, that your explanation holds, and the outliers really are polluted by significant unrelated noise. For purposes of estimating the processing time of a simulated job, you can eliminate them from the sample. In your simulation, though, you may need to factor them back in, either as a second error term in the generator (so most observations have $d = f(\text{illen}) + \text{err}$ but occasionally a job has $d = f(\text{illen}) + \text{err} + \text{extra delay}$), or else estimate the frequency/duration of those exogenous delays and model them separately (as service interruptions/preemptions).

You could try fitting a quadratic or cubic model (cubic seems to work better than linear — I tried log–log and that didn't work well at all), or you might ask yourself whether there's justification to consider separate models for different ranges of illen.

I'm perfectly fine to use piecewise model. My ultimate goal is to create a model of this component to study how it would perform under different load patterns.

I will take a look at cubic models; may be you have a reference to a

Re: regression assumptions are violated, what next?

good resource at hand? Statistics wasn't my major, so I'm working my way through on my own. Book references will work too. :-)

Not really, but there's nothing magical about it. Instead of regressing total on illen, regress on illen, illen^2 , illen^3 (three predictor variables), plus of course the constant term. It's a linear regression in the sense that you use the same regression routine you used before. (The key is that it's linear in the coefficients, even though it's polynomial in illen.)

Ah, and last and not least — thanks a lot for looking into this! :-)

You're welcome.

/Paul

.