Statistically Speaking 1: Diagnosing Assumption Violations in General(ized) Linear Models with DHARMa in R

Zachary Beneduci

2025-04-13

Purpose

Welcome to the first installment of the Statistically Speaking... blog? Tutorials? Not sure what this will morph into in the long run. What I am sure of is that many folks struggle with statistics - so much so that they may take a non-trivial amount of university courses and still feel like an ostrich with its head in the sand. Having taken a few of these courses myself, one area that is woefully neglected is that of checking model assumptions. Often, students are shown how to check assumptions for the general linear model (often abbreviated LM). Yet, this is not so straightforward for general**ized** linear (mixed) models (GL(M)Ms), and requires a departure from the standard residual plots. I hope that by the end of this document you'll feel a bit more confident that the GL(M)Ms you fit agree with your data.

An overview of linear models

Before I get into the main topic of the document, I think a refresher on linear models would serve the reader well. I'll go over:

1. the general linear model
2. fixed vs. random effects (as these are commonly used together in ecology)
3. and generalized linear models

The General Linear Model

Recall from your early algebra classes the slope-intercept form of the linear equation, which follows:

Eqn. 1a: $y = mx + b$,

where y is the value of the dependent variable, x is value of the independent variable, b is the y-intercept (the value at which the line crossed the y-axis), and m is the slope (the change in y for every unit of x).

For example, if $m = 2.6$ is the slope, $b = 3.4$ is the y-intercept, and I want to know the value of y when $x = 4$, we can solve the following equation for y:

Eqn. 1b: $y = 2.6 * 4 + 3.4$

which gives us 13.8.

Now, this is all well and good if you know the values for the variable x and coefficients (m and b). But let's say our information is reversed: we have a dataset, either collected in the field or provided by a colleague, where we have some number of paired values of x and y. And I want to know how x relates to y (i.e. the slope, m) and I similarly don't know b. Trying to figure this out with only Equation 1 and by hand would be insane. What, am I supposed to pick a pair of x and y values and arbitrarily select values of coefficients m and b until I find a solution? Doing this for more than one pair becomes infinitely daunting, especially since the coefficient values need to work for all possible variable pairs. Luckily, there are techniques to *estimate* what these coefficients should be.

This brings us to the linear model, which uses a technique called linear regression to relate (typically) one y to one or more xs. In linear modeling, the linear equation is represented a bit differently:

Eqn. 2a. $y = \beta_0 + \beta_1 X_1 + \epsilon$,

where y is still the independent variable, X_1 is the dependent variable, β_0 (read beta-0) which was b is now the y-intercept, and β_1 is which was m is now the slope. You'll also notice that there is this term ϵ (read epsilon) that stands for the residual error. You might be asking, what is the residual error?

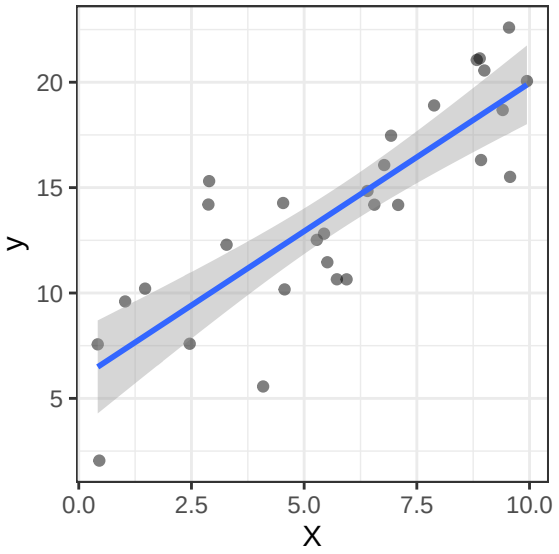
I'll illustrate by creating some data where we know the true coefficient values.

```
set.seed(123)
datum <- data.frame(X = runif(n = 30, min = 0, max = 10)) %>%
  mutate(error = rnorm(n = 30, mean = 0, sd = 2.84),
         y = 4.32 + 1.67*X + error)
```

Plotting these data:

```
plot1 <- ggplot(datum, aes(X, y)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = T) +
  theme_bw()

plot1
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



And here we have our simulated data.

Notice, I've also fitted a line through them. The coefficients for this line were estimated with an approach called ordinary least squares (OLS). OLS is a common approach to estimate the true values of the coefficients for the line by finding the values that **minimize the sum of squared deviations** from that line. The sum of squared deviations (SS, also called sum of squares) is expressed as:

Eqn. 3: $SS = \sum_{i=1}^{n} (x_i - \bar{x})^2$

where SS is the sum of the residual $(x - \bar{x})$ squared. In our example, there are 30 residuals, each corresponding to the distance on the y-axis between the fitted line and each observation. Essentially, the model finds the values for coefficients β_0 and β_1 that result in the smallest SS. Notice in the equation we take the sum of the **squared** residuals. This forces all residuals below the line to be positive. Without doing this, the residuals

would too often cancel each other out and result in $SS = 0$. Kept this way, there are many possible lines that would fit through the observed data. Heck, even a β_1 of 0 would work. Squaring the residuals ensures that the line fits through the center of the data.

We can find the exact values from the OLS estimation:

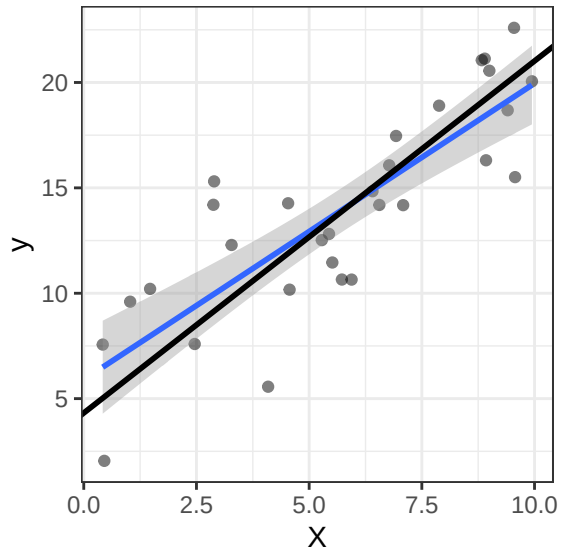
```
m1 <- lm(y~X, data = datum)

m1.sum <- summary(m1)

m1.sum$coefficients %>%
  as.data.frame() %>%
  knitr::kable()
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	5.900854	1.1441511	5.157408	1.8e-05
X	1.405901	0.1787509	7.865142	0.0e+00

These estimates aren't that far off. The intercept is a little to be desired, but the slope is about the same as what we set the true value at. Comparing the two lines:

```
plot1 +
  geom_abline(intercept = 4.32, slope = 1.67, linewidth = 1)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



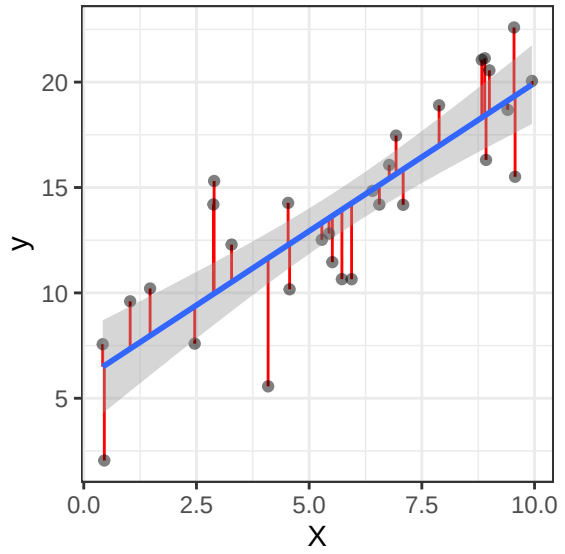
Even though the line is a bit off, notice that the true line is within the 95% confidence interval.

You might also be interested to see the residuals on the same plot:

```
m1.res <- residuals(m1)

datum <- datum %>%
  mutate(res = m1.res,
         y.ols = m1.sum$coefficients[1,1] + (m1.sum$coefficients[2,1]*X))

ggplot(datum, aes(X, y)) +
```

```
  geom_segment(aes(x = X, y = y.ols + res, xend = X, yend = y.ols), color = "red") +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = T) +
  theme_bw()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



As you can see, extracting the residuals from the model and adding these values to the fitted line gives the location of each observation.