Interpreting GLME Coefficients

How did the test result generate a *p-value*, and what should the experimenter consider in interpreting the results returned by the Matlab `fitglme` model? This document makes an argument for statistical reporting that makes use of a few different degrees of freedom in order to provide a broader heuristic for interpreting statistical significance. Any discussion in a scientific article should not only include a justification for why the number of degrees of freedom was selected in a given reported statistical test result, but also:

- Are there factors that could cause this to be an **over**-estimate of the number of degrees of freedom? How would those things change the interpretation?
- Are there factors that could cause this to be an **under**-estimate of the number of degrees of freedom? How would those things change the interpretation?

We are too-often trying to report a test result that is most convenient when in reality the results are ambiguous at best. If it is the case that the results are ambiguous, it should be made clear that this is the case, and the range of possible explanations that could best describe the outcome should be offered.

Post-Hoc t-tests for coefficient significance

The post-hoc comparisons for each coefficient result a test statistic (**`t`**), which corresponds to a *t-distribution* with a specified number of *degrees of freedom (***`df`***)*. The returned result (for a two-tailed test, which is the standard procedure since it "assumes less" about the prior distribution or comparison being made) is the probability (**`p`**) that the absolute value of **`t`** (if such an experiment were repeated an infinite number of times with all other elements the same) is greater than the value of **`t`** that was produced in this instance, given that the true mean of the coefficient being tested is actually zero.

Stated differently: how likely is it that we estimated this coefficient as non-zero purely by chance? To answer this, we must compare **`t`** to the prior (*t-distribution*) using **`df`**.

```
% For Coefficient estimates, process for estimating significance:
sig = @(t,df)(1 - tcdf(t,df) + tcdf(-t,df)); % p = sig(t,df) => p-value recovery procedure
result = @(t,df,p)fprintf(1,'\nP[|t| > %6.4f|df = %d) ~ %6.4f\n',t,df,p);
```

What is meant by `df`?

When dealing with time-series data or other types of data (in the Synaptophysin data, for example, it is spatial data from many pixels in a single image), we have many different samples and the model estimation process treats each sample as an individual degree of freedom in the post-hoc comparison. Strictly speaking, the total degrees of freedom computed in this model for each of the t-tests is the number of observations minus the number of fixed effects minus one.

Wait, what? Why?

Disclaimer: I'm not a "real" statistitician, but here is how I think about it:

You lose one degree of freedom each time you are computing differences.

Consider the simplest case, in which I have two samples, **A**, and **B**. I'd like to see "how different" are **A** and **B**. I have one measurement for each sample. There is really only one option: I now subtract one quantity from the other.

```
>> d = |A - B|
```

Note that we use the absolute deviation, since we only care about "how different" and not "which is greater" (a different question). It is now evident that `d` will only ever be one value, despite having two observations. Indeed, even if I compute the mean and variance of these samples, I will quickly see that the normalized deviation from the mean for each sample is equivalent. This generalizes to many samples. So, no matter what:

You lose one degree of freedom each time you are computing differences.

Now, consider a model with different coefficients. Each coefficient is a variable that is trying to "make sense" of the response data using some formulaic combination of explanatory data. Each coefficient term (including the coefficient for the model intercept term) therefore requires us to compute differences of things during the optimization procedure and we therefore lose a degree of freedom for each term. And, since we must also account for the process of taking the differences between the predictions generated by the linear combination of model terms from some estimated value, we lose an additional degree of freedom.

```
% Synaptophysin dataset: 220000 observations fit in GLME
df = 220000 - (12) - 1; % df = obs - (# terms) - 1
fprintf(1,'Degrees of freedom: %d\n', df);
```

```
Degrees of freedom: 219987
```

Why belabor this point about `df`?

The previous sub-section describes the *technically correct* way to handle the number of degrees of freedom. Wikipedia defines **Degrees of freedom (statistics)** as:

> In statistics, the number of **degrees of freedom** is the number of values in the final calculation of a statistic that are free to vary.[1]

However, when we take measurements from 10 individuals, do we assume that every single measurement is independent and that the number of observation values is equivalent to the number of "values" that are free to vary when interpreting the test statistic result?

Solution: report `p` for a range of `df`

The correct reporting procedure, which is *never* followed, would be to report statistical significance for different values of **`df`**, identify clearly what is meant when using that value for a given **`df`** (i.e. what assumptions on the independence of each observed data sample), and then offer the interpretation of the statistical outcome. In the case that statistical results return values of **`p`** that change statistical significance for a given threshold (**`alpha`**, typically equal to 0.01) in a manner that is dependent upon **`df`**, then a clear explanation of what could confound interpretation of the results and any possible follow-ups that should be performed would be the most prudent thing to put into the discussion.

Original Result (from `fitglme`)

```
% For example, we have the following tStat value for the
```

```
% Synaptophysin GLME coefficient 'Group_ADS:Hemisphere_LH:Area_RFA'
t = 2.74462511308895;
p = sig(t,df);
result(t,df,p);
```

```
P[|t| > 2.7446|df = 219987) ~ 0.0061
```

However, do we really believe that there are 219,987 independent values generating the observed statistical test result?

- We could argue that there are **more** values (*in which case, p-value is smaller*): the many unknown variables relating to neurophysiology, environment, time, etc.
- We could argue that there are **less** values (*indeed, this is what we **always** attempt to do in science when answering some question*).

Let's look at some scenarios.

`df` relates to images

In this scenario, we consider our total number of observations as only relating to the total number of recording sessions we did. I think this is the most reasonable, because it reflects "how much data was collected?" This took considerable time and effort, and the processing on each of those images was done independently in the sense that it wasn't all just collected at once by an automaton.

In this case, let's roughly approximate that there are **10** animals and roughly **10** sections per animal (to use "round" numbers).

```
df = ((10 - 1) * (10 - 1)) - 1; % ( sections * animals ) = 80 df
p = sig(t,df);
result(t,df,p);
```

```
P[|t| > 2.7446|df = 80) ~ 0.0075
```

`df` relates to number of fixed-effect levels

This could also be a sensible way to think about it. In this case, we consider the degrees of freedom in the model, which directly relates to the number of categorical fixed-effects in the model. Those often reflect the most-salient experimental manipulations or covariates that are being tracked and related to interpret the results, at any rate. In this case, we look at the interaction **`Group:Hemisphere:Area`**, which has

- 3 levels for **`Group`**
- 2 levels for **`Hemisphere`**
- 2 levels for **`Area`**

We use **`effects`** for the **`DummyVarCoding`** parameter in **`fitglme`**. Therefore, the number of degrees of freedom for each fixed effect is the number of levels minus one (*again, we are taking **differences** when using the "effects" DummyVarCoding method*).

```
df = ((3 - 1) * (2 - 1) * (2 - 1)) - 1; % ( group * hemisphere * area )= 2 df
p = sig(t,df);
result(t,df,p);
```

```
P[|t| > 2.7446|df = 1) ~ 0.2224
```

This is the *most-extreme* case and keep in mind that we are not considering another important effect (**`Image/Section`**).

`df` relates to fixed-effects and top components of trends-by-section

We could do a principal components analysis of the per-day trends to formally produce an estimate given some percent of the data explained by the top principal components of trends by section, but that is probably overkill (and likely would not change the interpretation much at this point). It is not unreasonable to think that there could be 3 or 4 different "patterns" we could expect to observe in terms of section trends for individuals. Using the conservative value of 3, we alter **`df`** from the previous result.

```
df = ((3 - 1) * (2 - 1) * (2 - 1) * 3) - 1; % (group * hemisphere * area * <section estimate #
p = sig(t,df);
result(t,df,p);
```

```
P[|t| > 2.7446|df = 5) ~ 0.0406
```

Now, once again considering each rat as an independent dimension (which is not unfair, I think, when using this "few" number of rats), we have:

```
df = ((3 - 1) * (2 - 1) * (2 - 1) * 3 * 10) - 1; % (group * hemisphere * area * <section> * <ar
p = sig(t,df);
result(t,df,p);
```

```
P[|t| > 2.7446|df = 59) ~ 0.0080
```