STATS 3001 / STATS 4104 / STATS 7054
Statistical Modelling III
Workshop 5 - GLS

John Maclean

Load packages

```
pacman::p_load(tidyverse, gglm, broom)
```

Preface

Consider your standard linear model

$$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \ ,$$

with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$.

On the right-hand-side of the first equation, call the first term the *fixed part* and the second term the *random part*. So far we in these workshops we have altered the fixed part (using GAMs, additive models) to model nonlinear response variables.

We now begin to consider alterations to the random part. As we shall see, altering the random part of a model can let you incorporate:

- *Heterogeneity (this workshop)*
- Nested data (random effects)
- Temporal or Spatial correlations
- Multiple types of random noise

GLS

You have an excellent theory lecture on GLS. Recall that GLS is about fitting the standard linear model, but with the random part

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{V}) \ .$$

The theory goes through how to obtain the best estimator if $\mathbf{V}$ is known. What if $\mathbf{V}$ is not known?

Let's learn how to estimate it.

1. Load the `squid.txt` dataset and have a look. We will use `Month` as a factor (with 12 levels) along with `DML` (a measurement of squid length) to attempt to predict `Testisweight`. (Aside: the ecological goal was to understand how quickly male squid reach sexual maturation.)

2. Do EDA - make a few quick plots showing how the response variable varies with the predictors.

3. Load the `nlme` package (we'll be using the `nlme::gls` function).

4. Fit a linear model to both predictors, including an interaction term, using the `gls` function. (The notation is no different to using `lm`.) Look at the model diagnostics. Save the linear model as `squid_lm` - you will use it later as a comparison tool.

5. Here's your key fact o' the workshop: the gls function includes various ways to describe *or estimate* the GLS covariance matrix $\sigma^2\mathbf{V}$. Once you have saved one (as you will below), you include it in the optional weights argument to gls, like

```
my_model <- gls (..., weights = my_variance_structure)
```

In the following I give you a list of theoretical variance structures and tell you how to implement them in gls. For each one:

- compare to the EDA and state which feature of the data is being represented/modeled.
- consider whether your new new model is *nested* in squid_lm. That is, can you get back to squid_lm by setting some parameter(s) to 0? If the models are nested, you can write down a null hypothesis for gls. If possible, state it.
- fit the model using the commands provided, look at model diagnostics, and use anova to compare to squid_lm. Interpret the output of the anova.

The "fixed variance" structure.

You will have noticed that the spread in data increases with DML. The "fixed variance" idea is to assume that the variance of the random term scales with one (or more) numeric predictor(s). In this model, since variance increases with DML, we assume the i-th residual is $\epsilon_i \sim \mathcal{N}(0, \sigma^2\, DML_i)$. Advantage: we keep a single parameter σ^2 for the variance but model one type of heterogeneity. Use the command v1_fixed <- varFixed(~DML) to describe the "fixed variance" structure, varying with DML, described above. Then include in gls and follow the dot points above.

The "VarIdent" structure.

(Forget the fixed variance for the moment - we'll come back to it.) You will also have noticed that the variance of the data changes based on the Month. The "VarIdent" structure lets you estimate a different variance for each level of a factor. Change notation slightly and let ϵ_{ij} be the residual for the i-th data point in the j-th month. Then the model is

$$\epsilon_{ij} \sim \mathcal{N}(0, \sigma_j^2).$$

Use the command v2_ident <- varIdent(form = ~ 1 | MONTH) to describe the variance structure, then include in a gls call and go through the dot points.

The "VarPower" structure

Here the structure is $\epsilon_i \sim \mathcal{N}(0, \sigma^2\, |DML_i|^{2\delta})$, where δ is a new parameter that will be estimated. Include by calling v3_power <- varPower(form = ~DML).

In addition to going through the dot points above, make sure to identify δ in the output from summary(). I think of VarPower as a generalisation of VarFixed with some nice properties - model nesting, for one.

The "varConstPower" structure

Keep thinking about "varPower": what if the value of your predictor is 0, or nearly 0, somewhere? Then you have 0 variance in the data. We probably don't want that, and in that case a better model would be

$$\epsilon_i \sim \mathcal{N}\left[0,\ \sigma^2\left(\delta_1 + |DML_i|^{2\delta_2}\right)\right].$$

Test it here using the varConstPower function. In addition to the dot points above, check you can identify the two parameters from the model summary.

Combining variance 1

In one of your preceding variance structures, try altering DML to DML | MONTH or 1 | MONTH to DML | MONTH. Run the model, and work out and write down the structure for the variance.

The combination structure

Of course with this dataset we want to combine multiple variance structures: your EDA reveals that the squid data variance changes with both predictors. The `varComb` function takes the above functions *as inputs* and returns a variance structure that is the *product* of all the inputs. Use “varComb” to make a variance structure that includes two of the above structures and accounts for heteroscedasticity resulting from both, squid length and month. Write down the null hypothesis (if it exists), run your model, and use `anova` commands to decide which of your models is the best. Check the assumptions for your final model.