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.)

```
squid <- read_delim("workshops/ZuurData/squid.txt")
```

```
## Rows: 768 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## dbl (5): Specimen, YEAR, MONTH, DML, Testisweight
```

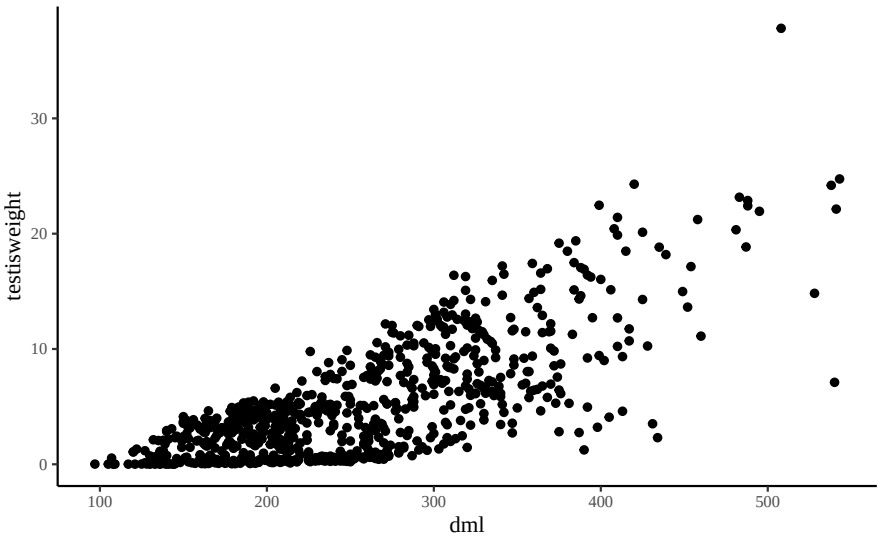
```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
squid <- squid %>%
  janitor::clean_names() %>%
  mutate(month = as_factor(month))
squid
```

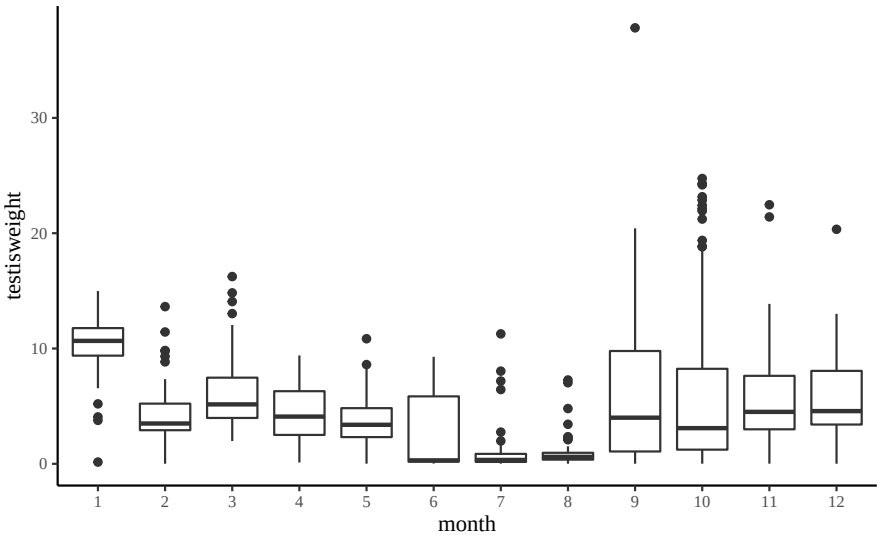
```
## # A tibble: 768 x 5
##    specimen  year month   dml testisweight
##       <dbl> <dbl> <fct> <dbl>        <dbl>
##  1     1017  1991 2       136        0.006
##  2     1034  1990 9       144        0.008
##  3     1070  1990 12      108        0.008
##  4     1070  1990 11      130        0.011
##  5     1019  1990 8       121        0.012
##  6     1002  1990 10      117        0.012
##  7     1001  1991 5       133        0.013
##  8     1013  1990 7       105        0.015
##  9     1002  1990 7       109        0.017
## 10     1006  1990 7        97        0.017
## # ... with 758 more rows
```

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

```
#simple EDA
squid %>% ggplot(aes(dml,testisweight)) + geom_point()
```




```
squid %>% ggplot(aes(month,testisweight)) + geom_boxplot()
```




```
# make a plot looking at both predictors at once

squid %>% ggplot(aes(dml,testisweight)) + geom_point() +
  facet_wrap(~month)
```

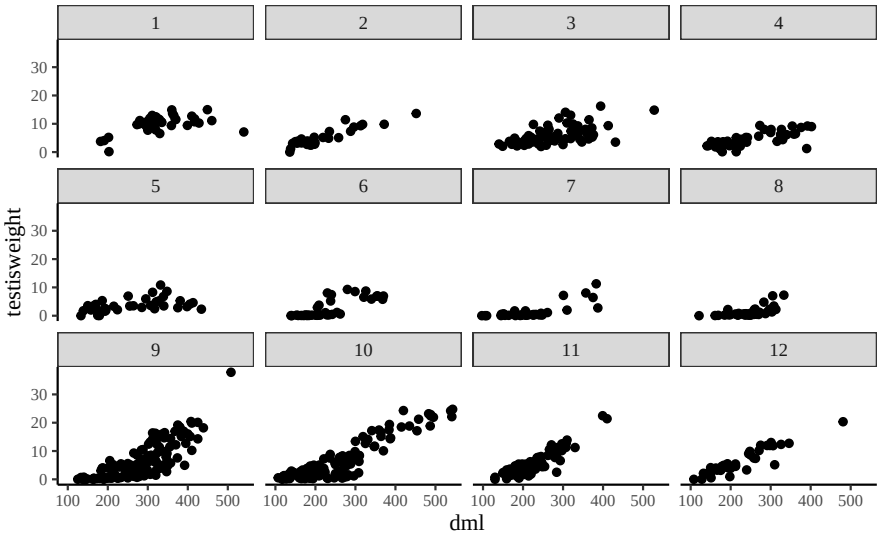


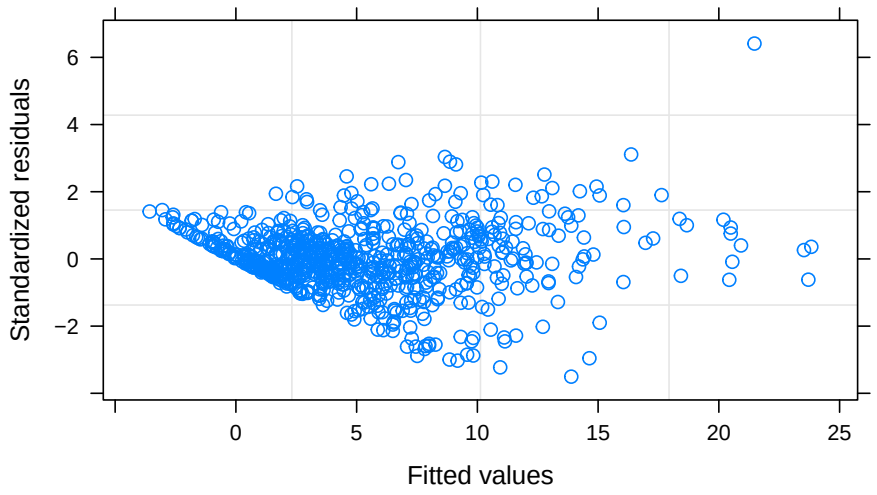
Figure 1: Plot weight vs length in each month.

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

```
pacman::p_load(nlme)
```

2. 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.

```
squid_lm <- gls(testisweight ~ dml * month, data = squid)
plot(squid_lm)
```

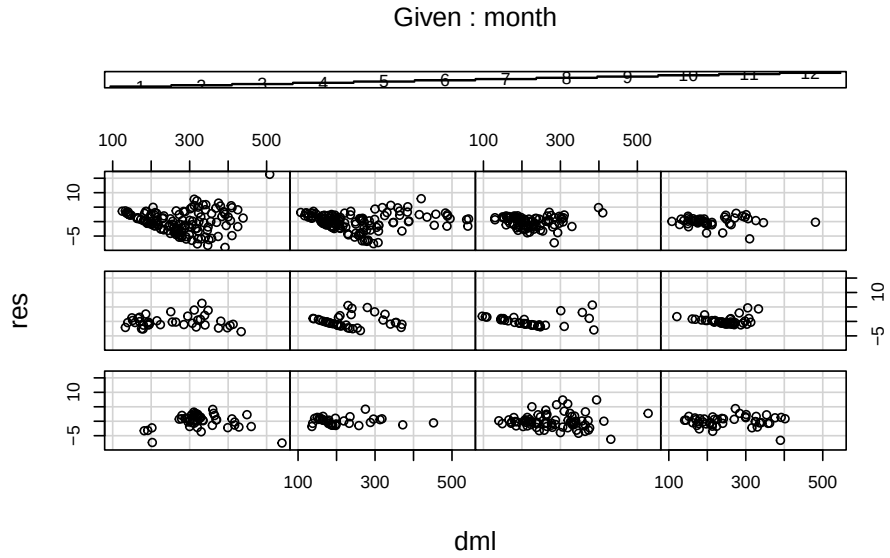



```
#summary(squid_lm)
#qqnorm(squid_lm)
```

Model diagnostics are sparse but can clearly see the heteroscedasticity. Comment: the assumption of normal residuals with constant variance is key to some derived quantities. Your F-statistic only has an F-distribution if that assumption is true; similarly your t-statistic only has a t-distribution if the assumption holds. The t-values and p-values cannot be trusted unless your model assumptions are met!

Let's use a tool to investigate the residuals a little more finely: the coplot.

```
res <- resid(squid_lm)
coplot(res ~ dml | month, data = squid)
```



The bottom left plot is month 1, the bottom right plot is month 4, and the middle left plot is month 5. Take a moment to work out the rule here.

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:

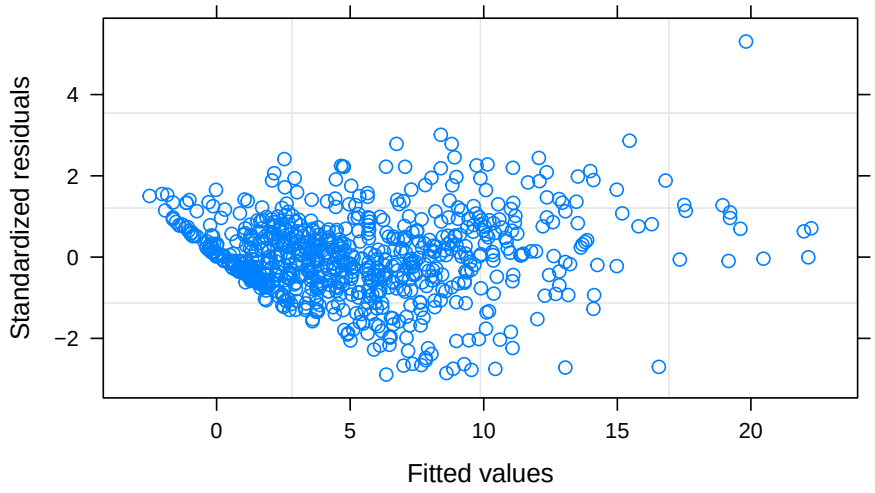
- compare to the EDA and state which feature of the data is being represented/modeled.
- consider whether your new new model is *nested* in `squid_lm`. That is, can you get back to `squid_lm` by setting some parameter(s) to 0? If the models are nested, you can write down a null hypothesis for `gls`. 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.

- compared to EDA: we are modelling the increase in variance with `dml` in the *first* scatterplot.
- no nesting of models - so `anova` can only compare information criterion and log-likelihoods.
- implement:

```
v1_fixed <- varFixed(~dml)
squid_fixed <- gls(testisweight ~ dml * month , weights = v1_fixed, data = squid)
plot(squid_fixed)
```




```
anova(squid_fixed, squid_lm)
```

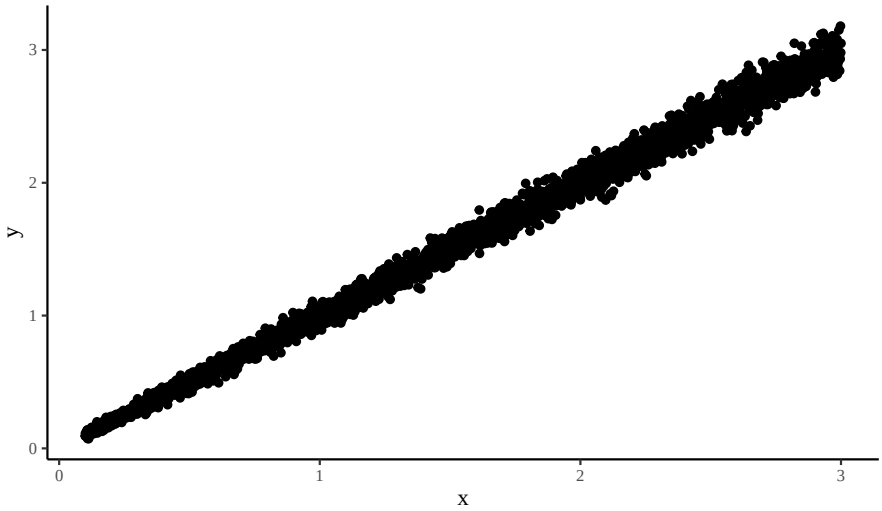
```
##              Model df      AIC      BIC    logLik
## squid_fixed      1 25 3620.898 3736.199 -1785.449
## squid_lm         2 25 3752.084 3867.385 -1851.042
```

Comments: heteroscedasticity still visible, but all components of `anova` favour `squid_fixed` over `squid_lm`. Information criteria are smaller, log likelihood is larger.

At this point you may have a question. What should the plot look like if assumptions are met?? That is, does `plot.gls` plot some type of scaled residual that, if the model's assumptions were met, would be now homoscedastic? That's my assumption but let's check. Make synthetic data:

```
x <- seq(0.1,3,by=0.001)
n<- length(x)
y<- x + 0.05*sqrt(x)*rnorm(n,0,1)
```

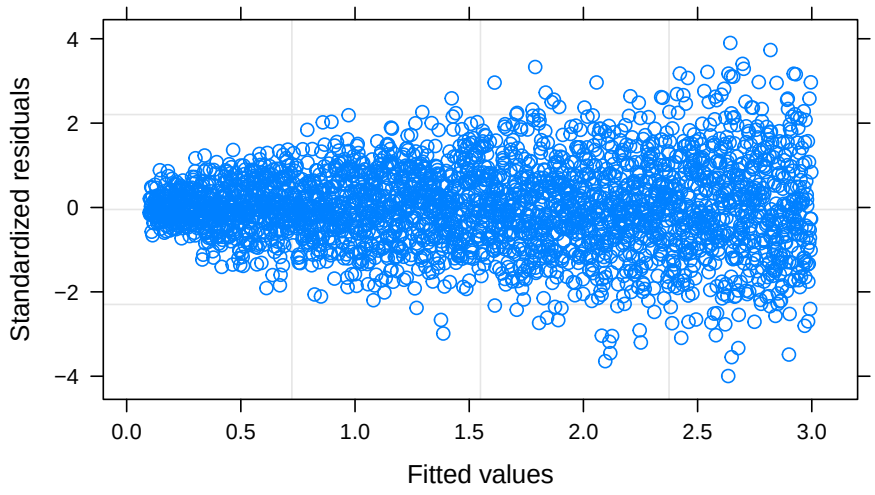
```
df <- tibble(x,y)
df %>% ggplot(aes(x,y)) + geom_point() + labs(caption = "Variance increases linearly with x")
```



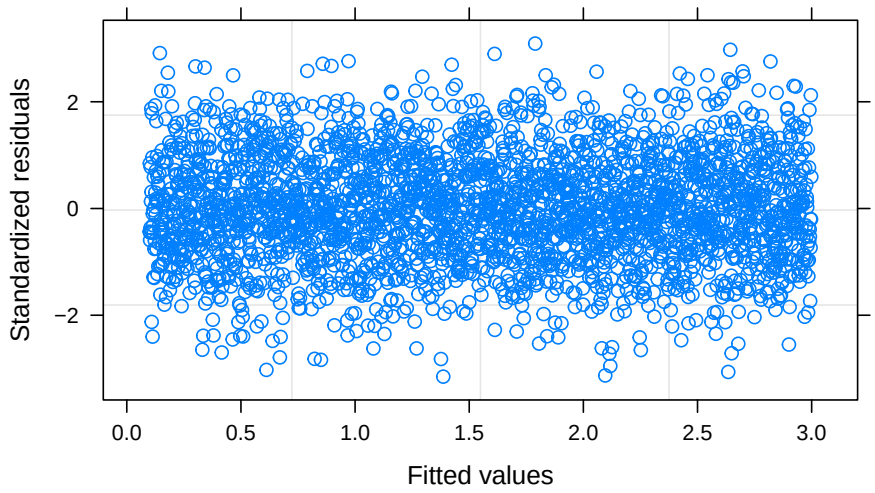
Variance increases linearly with x

```
df_lm <- gls(y~x, data = df)
var_df_fixed <- varFixed(~x)
df_fixed <- gls(y~x, weights = var_df_fixed, data = df)

plot(df_lm) #heteroscedastic
```




```
plot(df_fixed) #homoscedastic
```



Confirmed - as you would hope, the standard `gls` plot *standardises* the residuals so that they will be (if your model's assumptions are met) homoscedastic.

The "VarIdent" structure.

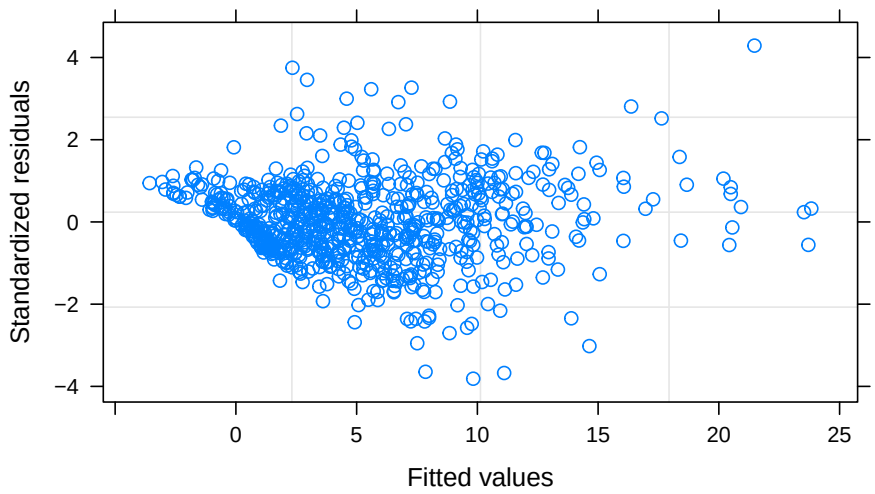
(Forget the fixed variance for the moment - we'll come back to it.) You will also have noticed that the variance of the data changes based on the `Month`. The "VarIdent" structure lets you estimate a different variance for each level of a factor. Change notation slightly and let ϵ_{ij} be the residual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

- here we model the different variance for each month shown in the second scatterplot in EDA.
- this *is* a nested model: you recover standard linear regression if all $\sigma_j = \sigma$. That means `anova` can compare the two models more directly.
- implement:

```
v2_ident <- varIdent(form = ~ 1 | month)
squid_ident<- gls(testisweight ~ dml * month, weights = v2_ident, data = squid)
plot(squid_ident)
```



```
anova(squid_ident, squid_lm)
```

```
##             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## squid_ident     1 36 3614.436 3780.469 -1771.218
## squid_lm        2 25 3752.084 3867.385 -1851.042 1 vs 2 159.6479  <.0001
```

Similarly to before - still heteroscedastic, but we have an additional tool. Note that since the models were nested the `anova` automatically includes a test and p-value for the added terms in the varIdent model.

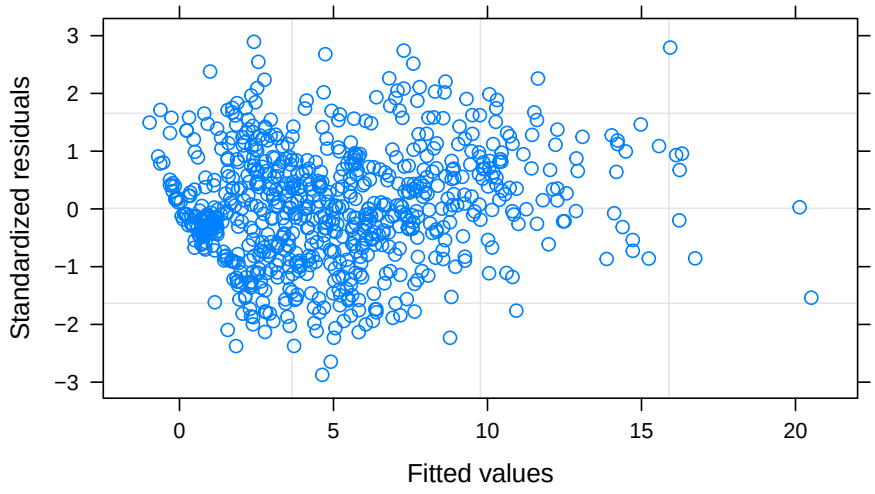
The "VarPower" structure

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. Comments:

- as with varFixed we are modelling the heteroscedasticity in data with `dml`.
- the linear model is nested: if $\delta = 0$ then you get back to standard linear regression. Nice added property: if you estimate $\delta \approx 0.5$ then you have recovered the varFixed model.
- implement:

```
v3_power <- varPower(form = ~dml)
squid_power <- gls(testisweight ~ dml * month , weights = v3_power, data = squid)
plot(squid_power)
```




```
anova(squid_lm, squid_power)
```

```
##             Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## squid_lm        1 25 3752.084 3867.385 -1851.042
## squid_power     2 26 3473.019 3592.932 -1710.509 1 vs 2 281.0648  <.0001
```

```
#summary(squid_power)
```

Have not totally fixed heteroscedasticity but this is the best model yet. Note from the `summary` command that δ (written as `power`) is estimated to be 1.76.

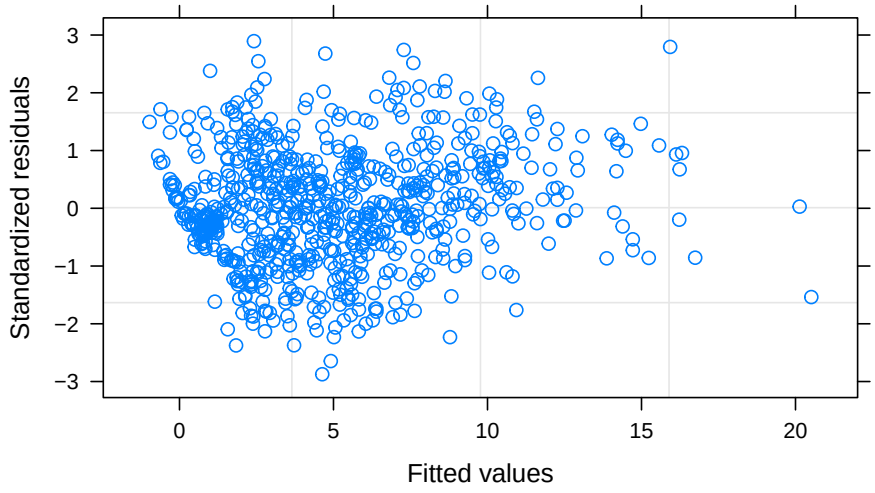
The "varConstPower" structure

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`.

```
v4_const_power <- varConstPower(form = ~dml)
squid_const_power <- gls(testisweight ~ dml * month , weights = v4_const_power, data = squid)
plot(squid_const_power)
```




```
anova(squid_lm, squid_const_power)
```

```
##                   Model df      AIC      BIC    logLik   Test L.Ratio p-value
## squid_lm              1 25 3752.084 3867.385 -1851.042
## squid_const_power     2 27 3475.019 3599.544 -1710.509 1 vs 2 281.065  <.0001
```

```
#summary(squid_const_power)
```

Here we conclude... probably little reason to use `varConstPower` for this dataset. The parameter δ_1 is estimated to be 0.099 - tiny compared to the variance in `dml`. Check:

```
anova(squid_power,squid_const_power)
```

```
##                   Model df      AIC      BIC    logLik   Test      L.Ratio
## squid_power           1 26 3473.019 3592.932 -1710.509
## squid_const_power     2 27 3475.019 3599.544 -1710.509 1 vs 2 0.0001170151
##                   p-value
## squid_power
## squid_const_power  0.9914
```

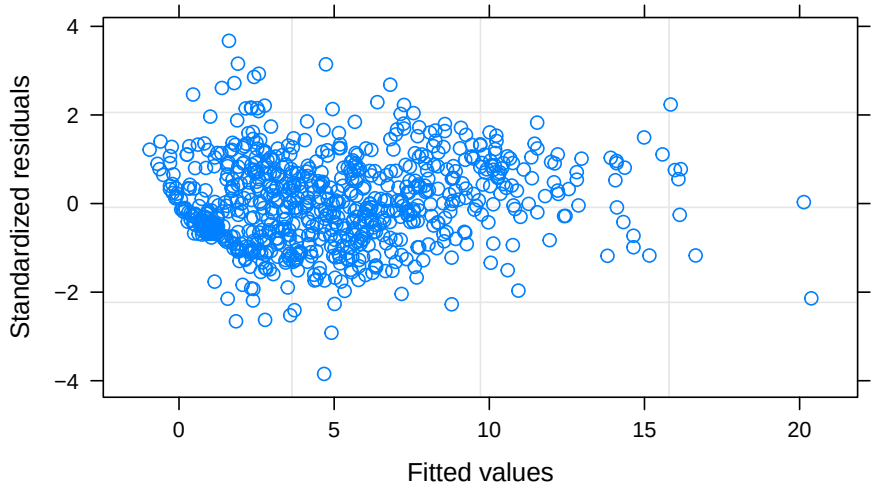
and confirm that there's no significant difference between the models (so we should use the simpler one, `squid_power`).

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.

Take my best model and alter:

```
v5_combo_power <- varPower(form = ~ dml | month)
squid_combo_power <- gls(testisweight ~ dml * month , weights = v5_combo_power, data = squid)
plot(squid_combo_power)
```




```
anova(squid_power, squid_combo_power)
```

```
##                   Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## squid_power           1 26 3473.019 3592.932 -1710.509
## squid_combo_power     2 37 3407.511 3578.156 -1666.755 1 vs 2 87.50799  <.0001
```

```
#summary(squid_power)
```

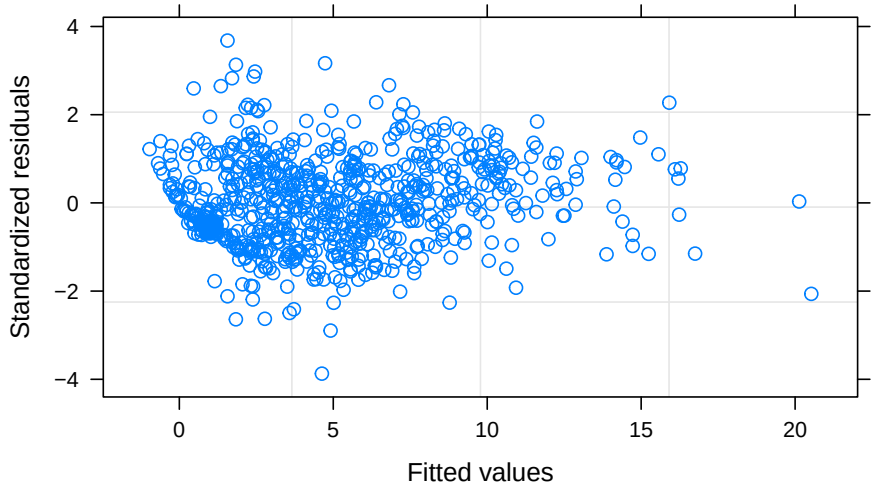
Comment: homoscedasticity much improved, though there is a cluster of data points near 0 on the fitted values (use a coplot to see why). The anova suggests further improvement.

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.

```
v6_final_combo <- varComb(
  varPower(form = ~ dml),
  varIdent(form = ~ 1 | month)
)
squid_final_combo <- gls(testisweight ~ dml * month , weights = v6_final_combo, data = squid)
```

```
plot(squid_final_combo)
```




```
anova(squid_power, squid_final_combo)
```

```
##                    Model df      AIC      BIC    logLik   Test  L.Ratio p-value
## squid_power            1 26 3473.019 3592.932 -1710.509
## squid_final_combo      2 37 3406.231 3576.877 -1666.116 1 vs 2 88.78751  <.0001
```

```
#summary(squid_power)
```

Comment: in two final models, we've managed to model much of the heterogeneity. The remaining 'homoscedasticity' (a cluster of data points with similar residuals near 0 on the fitted values) is, I think, a violation of independence on the data. Look for data points that appear to fall on a curve in my third EDA plot.