ORIGIN := 0

Controlling Variance Structures with Generalized Least Squares Regression

Fitting a line or higher-order surface showing relationship between a single dependent variable (Y) and one or more independent variables (X1, X2… that may include either or both numerical values and factors) is relatively straightforward using R's lm() function. However, inferences drawn from this fit, including usual F and t-tests with associated probability values, require additional assumptions. Chief among these is Normal distribution of the Y's, or equivalently Normal distribution of the residuals (ε's) derived from standard Linear Regression models. Validation, including making graphs of residuals versus fitted values and independent variables, sometimes associated with formal tests, must be performed to see if assumptions are met. If not, then probability distributions of test statistics do not necessarily match those assumed and all results must be regarded with caution. When data violate assumptions, one common approach is to apply "variance stabilizing" transformations to one or all variables. Often this helps, but many times not so much. Recently, software has advanced to the point where much more sophisticated appraisal and correction of problems can be routinely performed. Shown here is use of "Variance Structures" implemented by gls(), lme(), and associated functions of the {nlme} package in R. These procedures treat the problem of linear regression as a "mixed model" problem by assuming different kinds of correlations between the variables. It is possible to try several variance structures "on for size" and pick one that seems to fit best. Testing the relative overall fit of each model with or without variance structure may be performed, or judged using AIC. Results of such tests, involving Full versus Reduced models, are generally deemed superior to more traditional approaches, such as Bartlett's test. This example is drawn from a highly recommended introduction in Zuur et al. 2009, *Mixed Effects Models and Extensions in Ecology with R.* Software and underlying math is extensively covered in Pinheiro & Bates 2004, *Mixed-Effects Models in S and S-Plus.*

Example in R:

```
library(nlme)
setwd("c:/DATA/Models")
#READING DATA AND DEFINING MONTH AS A FACTOR
SQ=read.table("Squid.txt",header=T)
SQ$fMONTH=factor(SQ$MONTH) #PUTS FACTOR INSIDE SQ
SQ
```

```
> SQ
    Specimen YEAR MONTH DML Testisweight fMONTH
1       1017 1991     2 136        0.006      2
2       1034 1990     9 144        0.008      9
3       1070 1990    12 108        0.008     12
4       1070 1990    11 130        0.011     11
5       1019 1990     8 121        0.012      8
6       1002 1990    10 117        0.012     10
7       1001 1991     5 133        0.013      5
8       1013 1990     7 105        0.015      7
9       1002 1990     7 109        0.017      7
10      1006 1990     7  97        0.017      7
11      1020 1990     9 144        0.022      9
....
759     1014 1991    10 495       21.939     10
760     1022 1991    10 541       22.133     10
761     1012 1991    10 488       22.415     10
762     1019 1990    11 399       22.468     11
763     1017 1991    10 488       22.874     10
764     1021 1991    10 483       23.162     10
765     1018 1991    10 538       24.195     10
766     1015 1991    10 420       24.292     10
767     1020 1991    10 543       24.746     10
768     1002 1991     9 508       37.811      9
```

Data is read from disk file here, but is also available in the {AED} package for R. The latter may be downloaded from: http://www.highstat.com After data is read in here, an additional column is constructed inside variable SQ for factor fMONTH. Total length each vector is 768.

```
M <- lm(Testisweight ~ DML * fMONTH,data=SQ)
anova(M)
summary(M)

M.LM=gls(Testisweight~DML*fMONTH,data=SQ)
anova(M.LM)
summary(M.LM)
```

^ A fully equivalent linear model can be fit using the standard lm() function, or by gls(). The latter is recommend for comparing results with models having different variance structures.

```
> anova(M)
Analysis of Variance Table

Response: Testisweight
             Df  Sum Sq Mean Sq  F value    Pr(>F)    
DML           1 11247.2 11247.2 1732.082 < 2.2e-16 ***
fMONTH       11  2099.1   190.8   29.388 < 2.2e-16 ***
DML:fMONTH   11  1678.0   152.5   23.492 < 2.2e-16 ***
Residuals   744  4831.1     6.5                       
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

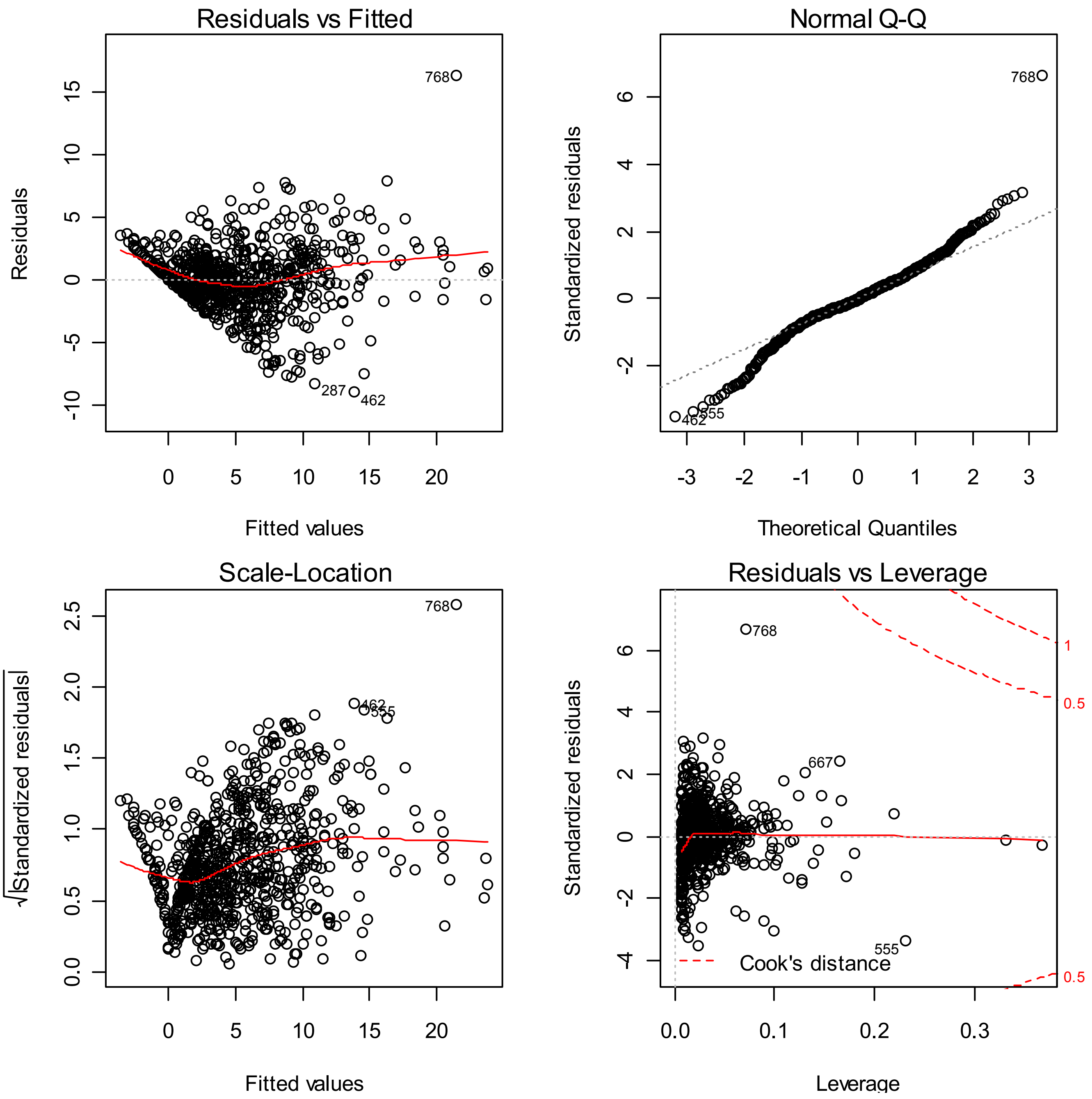
```
> anova(M.LM)
Denom. DF: 744 
            numDF  F-value p-value
(Intercept)     1 3299.174  <.0001
DML             1 1732.083  <.0001
fMONTH         11   29.388  <.0001
DML:fMONTH     11   23.492  <.0001
```

^ In R, anova() is a general function that calls hidden implementations anova.lm() and anova.gls(), for lm & gls objects respectively. Although the reports differ in form, the results are essentially the same. Slight differences in numerical results may occur as a result of rounding errors.

```
> summary(M)

Call:
lm(formula = Testisweight ~ DML * fMONTH, data = SQ)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.9295 -1.2374 -0.0735  1.3379 16.3362 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   3.215e+00  1.874e+00   1.716 0.086645 .  
DML           2.116e-02  5.559e-03   3.806 0.000153 ***
fMONTH2      -6.720e+00  2.328e+00  -2.886 0.004010 ** 
fMONTH3      -3.927e+00  2.208e+00  -1.778 0.075756 .  
fMONTH4      -4.772e+00  2.288e+00  -2.086 0.037309 *  
fMONTH5      -2.771e+00  2.266e+00  -1.223 0.221627    
fMONTH6      -9.598e+00  2.416e+00  -3.972 7.81e-05 ***
fMONTH7      -7.495e+00  2.293e+00  -3.268 0.001132 ** 
fMONTH8      -7.479e+00  2.931e+00  -2.552 0.010917 *  
fMONTH9      -1.496e+01  2.035e+00  -7.352 5.17e-13 ***
fMONTH10     -1.232e+01  1.964e+00  -6.274 5.97e-10 ***
fMONTH11     -1.265e+01  2.192e+00  -5.771 1.16e-08 ***
fMONTH12     -9.236e+00  2.226e+00  -4.148 3.74e-05 ***
DML:fMONTH2   1.803e-02  8.310e-03   2.170 0.030331 *  
DML:fMONTH3   3.151e-03  6.866e-03   0.459 0.646376    
DML:fMONTH4   2.972e-03  7.432e-03   0.400 0.689383    
DML:fMONTH5  -8.677e-03  7.238e-03  -1.199 0.230987    
DML:fMONTH6   1.762e-02  8.418e-03   2.093 0.036671 *  
DML:fMONTH7   4.647e-03  7.993e-03   0.581 0.561125    
DML:fMONTH8   5.001e-04  1.056e-02   0.047 0.962229    
DML:fMONTH9   4.424e-02  6.213e-03   7.121 2.53e-12 ***
DML:fMONTH10  3.949e-02  5.967e-03   6.618 6.95e-11 ***
DML:fMONTH11  4.667e-02  7.464e-03   6.253 6.79e-10 ***
DML:fMONTH12  3.410e-02  7.702e-03   4.427 1.10e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1
‘ ’ 1

Residual standard error: 2.548 on 744 degrees of freedom
Multiple R-squared: 0.7567,	Adjusted R-squared:
0.7492 
F-statistic: 100.6 on 23 and 744 DF,  p-value: < 2.2e-16
```

```
#VALIDATION PLOTS
op=par(mfrow = c(2,2), mar=c(4,4,2,2))
plot(M)
par(op)
```

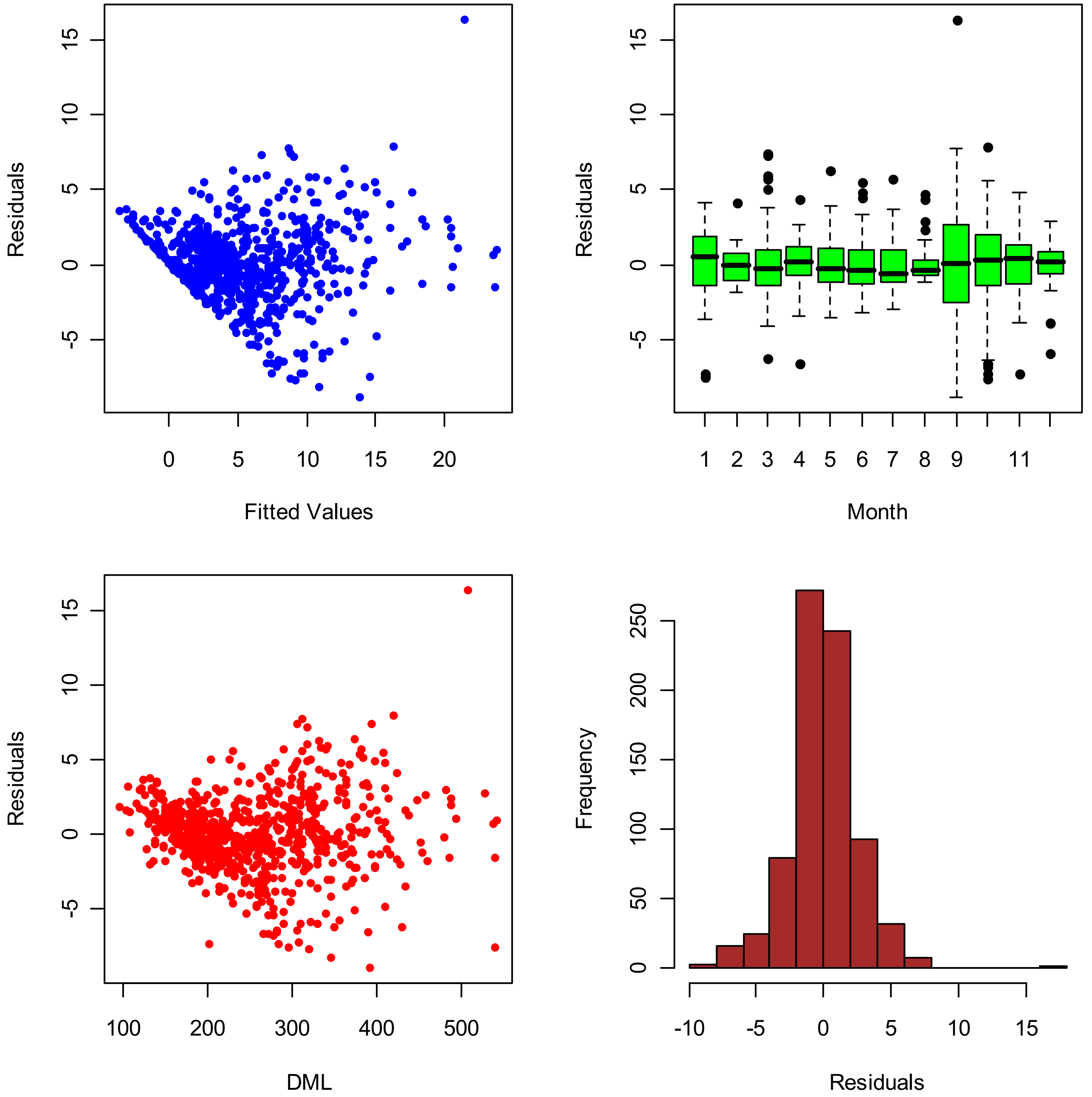
```
> summary(M.LM)

Generalized least squares fit by REML
  Model: Testisweight ~ DML * fMONTH 
  Data: SQ 
       AIC      BIC    logLik
  3752.084 3867.385 -1851.042

Coefficients:
                  Value Std.Error   t-value p-value
(Intercept)    3.215222 1.8740686  1.715637  0.0866
DML            0.021157 0.0055585  3.806244  0.0002
fMONTH2       -6.720020 2.3282175 -2.886337  0.0040
fMONTH3       -3.926923 2.2081953 -1.778340  0.0758
fMONTH4       -4.772045 2.2875351 -2.086108  0.0373
fMONTH5       -2.771457 2.2656715 -1.223239  0.2216
fMONTH6       -9.598061 2.4163109 -3.972196  0.0001
fMONTH7       -7.494959 2.2933137 -3.268178  0.0011
fMONTH8       -7.479426 2.9310875 -2.551758  0.0109
fMONTH9      -14.963009 2.0353133 -7.351698  0.0000
fMONTH10     -12.320827 1.9637718 -6.274063  0.0000
fMONTH11     -12.650490 2.1922272 -5.770611  0.0000
fMONTH12      -9.235813 2.2264505 -4.148223  0.0000
DML:fMONTH2    0.018032 0.0083101  2.169894  0.0303
DML:fMONTH3    0.003151 0.0068657  0.458989  0.6464
DML:fMONTH4    0.002972 0.0074316  0.399848  0.6894
DML:fMONTH5   -0.008677 0.0072378 -1.198801  0.2310
DML:fMONTH6    0.017620 0.0084180  2.093183  0.0367
DML:fMONTH7    0.004647 0.0079929  0.581432  0.5611
DML:fMONTH8    0.000500 0.0105569  0.047373  0.9622
DML:fMONTH9    0.044242 0.0062130  7.120835  0.0000
DML:fMONTH10   0.039495 0.0059673  6.618449  0.0000
DML:fMONTH11   0.046671 0.0074640  6.252829  0.0000
DML:fMONTH12   0.034099 0.0077022  4.427127  0.0000

 Correlation: 
... (REMOVED FROM THIS OUTPUT TO SAVE SPACE)
Standardized residuals:
        Min          Q1         Med          Q3 
  Max 
-3.50421681 -0.48560405 -0.02886169  0.52503032 
6.41081660 

Residual standard error: 2.548226 
Degrees of freedom: 768 total; 744 residual
```



^ Standard validation plots in R for linear model object made by lm().

Commands for "homemade" plots below:

```
#HOMEMADE PLOTS
op=par(mfrow = c(2,2), mar=c(4,4,2,2))
plot(fitted(M),resid(M),col='blue',xlab='Fitted Values',ylab='Residuals', pch=20)
plot(SQ$fMONTH, resid(M.LM),pch=19,col='green',xlab="Month",ylab="Residuals")
plot(SQ$DML, resid(M.LM),col='red',xlab="DML",ylab="Residuals",pch=20)
hist(resid(M.LM),col="brown",main='',xlab="Residuals")
par(op)
```



^ As described by Zuur et al, plots of residuals show distinct patterns with fitted values and with increasing values of continuous X-variable DML. Differences in variance are also noted among different levels of X-factor MONTH. Both problems require remedial action.

The general strategy is to fit several different variance structures to the linear model as specified by formulas identified by the "weights=" option in the gls() command (when weights are unspecified, this is the standard non-weighted linear model). Each specified weights= formula enters into a "mixed" linear model as a "random" component. Descriptions for each are given with the R command below.

```
M.LM=gls(Testisweight~DML*fMONTH,data=SQ)
```

$\varepsilon_i \sim N(0,\sigma^2)$

Error term in the linear model is homogeneous and only involves an estimate of σ^2.

Error term varies:

```
M.varFixed=gls(Testisweight~DML*fMONTH,
        weights=varFixed(~DML),
         data=SQ)
```

$\varepsilon_i \sim N(0,\sigma^2 \cdot DML_i)$

linearly with DML.

```
M.varIdent=gls(Testisweight ~ DML*fMONTH,
        weights=varIdent(form= ~ 1|fMONTH),
          data =SQ)
```

$\varepsilon_{ij} \sim N(0,\sigma_j^2)$

is different for each level j of factor MONTH indicated by σ_j^2.

```
M.varPowerA=gls(Testisweight ~ DML * fMONTH,
        weights = varPower(form =~ DML),
          data=SQ)
```

$\varepsilon_i \sim N(0,\sigma^2 \cdot |DML_i|^{2\delta})$

with DML raised to power 2δ.

```
M.varPowerB<-gls(Testisweight ~ DML * fMONTH,
        weights = varPower(form=~ DML | fMONTH),
          data = SQ)
```

$\varepsilon_{ij} \sim N(0,\sigma^2 \cdot |DML_{ij}|^{2\delta j})$

with DML raised to powers $2\delta_j$ for each level j of MONTH.

```
M.varExp=gls(Testisweight ~ DML * fMONTH,
        weights = varExp(form =~ DML),
          data = SQ)
```

$\varepsilon_i \sim N(0,\sigma^2 \cdot e^{2\delta(DMLi)})$

with e raised to the 2δ times DML power.

```
M.varConstPowerA=gls(Testisweight ~ DML * fMONTH,
       weights = varConstPower(form =~ DML),
         data = SQ)
```

$\varepsilon_i \sim N(0,\sigma^2 \cdot (\delta 1+|DML_i|^{2\delta 2})^2$

with σ^2 times $(\delta 1+|DML|^{2\delta 2})^2$

```
M.varConstPowerB=gls(Testisweight ~ DML * fMONTH,
        weights = varConstPower(form =~ DML | fMONTH),
         data = SQ)
```

$\varepsilon_{ij} \sim N(0,\sigma^2 \cdot (\delta 1_j+|DML_{ij}|^{2\delta 2j})^2$

with σ^2 times $(\delta 1_j+|DML|^{2\delta 2j})^2$ for each value j of MONTH

```
M.varComb=gls(Testisweight ~ DML * fMONTH,
        weights = varComb(varIdent(form= ~ 1 | fMONTH),
                  varExp(form =~ DML) ),
         data = SQ)
```

$\varepsilon_{ij} \sim N(0,\sigma^2 \cdot (\delta 1_j+e^{2\delta(DMLij)})$

with $(\delta_{1j}+e^{2\delta j(DMLij)})$ for each value of j of MONTH

In general, there may be little *a priori* reason to accept one of these variance structures over another. Note that only varIdent() applies to a factor variable by itself. Some, but not all, variance structures internest *only if* setting parameters σ, σ_j, δ or δ_j etc. in one model (FM) yields another (RM). The easiest way to judge between them is by using AIC.

```
AIC  ( M.LM,
       M.varFixed,
       M.varIdent,
       M.varPowerA,
       M.varPowerB,
       M.varExp,
       M.varConstPowerA,
       M.varConstPowerB,
       M.varComb  )
```

```
                 df      AIC
M.LM             25 3752.084
M.varFixed       25 3620.898
M.varIdent       36 3614.436
M.varPowerA      26 3473.019
M.varPowerB      37 3407.511
M.varExp         26 3478.152
M.varConstPowerA 27 3475.019
M.varConstPowerB 49 3431.511
M.varComb        37 3414.817
```

```
M.varPowerB=gls(Testisweight ~ DML * fMONTH,
        weights = varPower(form=~ DML | fMONTH),
          data = SQ)
```

< This model is preferred using AIC (smallest). Values confirmed in Zuur et al. p. 82.

Testing Improvement of Model with "optimal" Variance Structure:

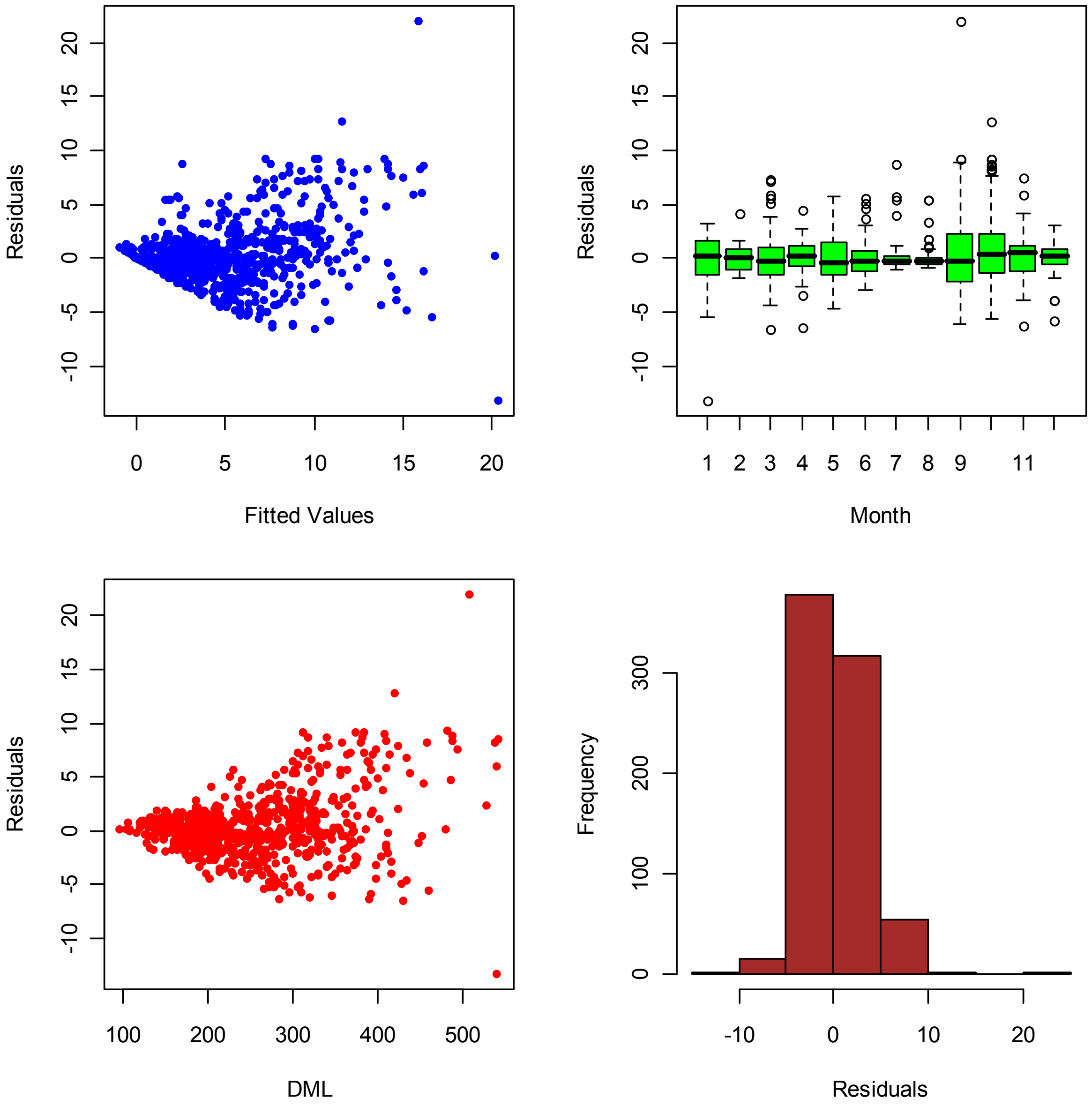
```
#TESTING IMPROVEMENT OF M.varPowerB MODEL
anova(M.LM,M.varPowerB)
```

```
> anova(M.LM,M.varPowerB)
            Model df      AIC      BIC    logLik   Test  L.Ratio p-value
M.LM            1 25 3752.084 3867.385 -1851.042
M.varPowerB     2 37 3407.511 3578.156 -1666.755 1 vs 2 368.5728  <.0001
```

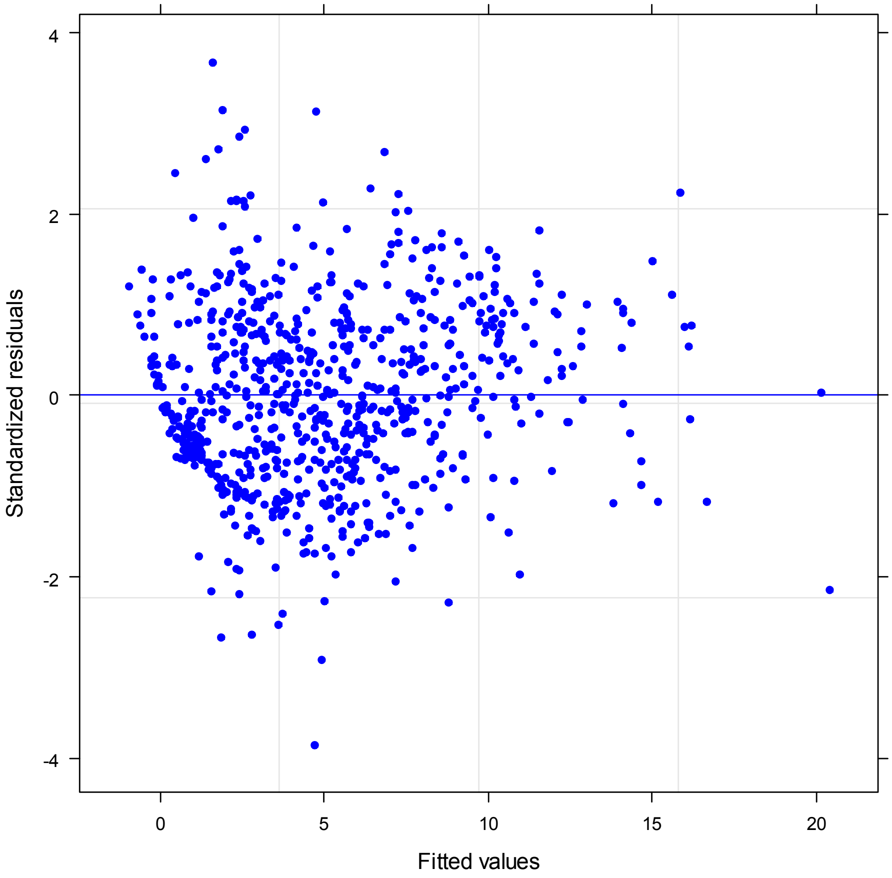
Model Validation:

Indicates strong preference for more complex model M.varPowerB. ^

```
#VALIDATION PLOTS
M2=M.varPowerB
op=par(mfrow = c(2,2), mar=c(4,4,2,2))
plot(fitted(M2),resid(M2),col='blue',xlab='Fitted Values',ylab='Residuals', pch=20)
plot(SQ$fMONTH, resid(M2),col='green',xlab="Month",ylab="Residuals")
plot(SQ$DML, resid(M2),col='red',xlab="DML",ylab="Residuals",pch=20)
hist(resid(M2),col="brown",main='',xlab="Residuals")
par(op)
```



```
#PLOTTING STANDARDIZED RESIDUALS
plot(M2,pch=20,col='blue')
```



According to Zuur et al., looking for patterns in standardized residuals of a model with variance structures is the preferable way to judge performance. Above, the pattern looks a little more random than for the linear model above. In general, one hopes to see no pattern at all.