Linear Mixed models
using ASReml in ecological studies

沈阳, 14-16 August

chenlei@ibcas.ac.cn

Outline

Facts

- Most ecological datasets require complex analysis of variance.
- This is now commonly done with mixed models.
- R is bad at this, but fortunately there is ASReml that can be used with R.

Difference among fitting method in R

- lme4
- nlme
- ASReml

ASReml简介

- 收费软件，由VSN公司开发，该公司同时还开发了著名的GenStat软件；
- ASReml软件最初应用在作物和畜牧遗传育种学科，其最大的优势在于其可以很容易地和有效地处理大型数据集和复杂模型；
- 价格：50-250 英镑
- 平台： windows，mac，linux 以及R（windows，mac，linux ）
- 使用权限： 目前在苏黎世大学bernhard教授的资助下，植物所和北京大学购买了为期一年的无限license版本。

Software

- SAS: proc mixed
- ASREML
- aML
- SPSS: Linear Mixed Models
- STATA: xtreg, gllamm, xtmixed
- R: the lme，glmer，MCMCglmm function
- S+: linear mixed models
- Gauss
- Matlab
- Genstat: REML

为什么使用混合效应模型和asreml？

- 非平衡以及NA数据
- 复杂模型和大数据
- 方差分析（anova）
- 自相关

非平衡以及NA数据

- library(asreml);data(oats)
- oats[1,]

```
Blocks   Nitrogen  Subplots   Variety      Wplots   yield
  1      0.6_cwt      1      Marvellous      1       156
```

- xtabs(~Blocks+Nitrogen,data=oats)

```
                    Nitrogen
Blocks   0_cwt   0.2_cwt  0.4_cwt   0.6_cwt
   1       3        3        3         3
   2       3        3        3         3
   3       3        3        3         3
   4       3        3        3         3
   5       3        3        3         3
   6       3        3        3         3
```

- data=oats[sample(1:nrow(oats),40),]
- xtabs(~Blocks+Nitrogen,data=data)

	Nitrogen			
Blocks	0_cwt	0.2_cwt	0.4_cwt	0.6_cwt
1	1	3	1	2
2	1	1	0	2
3	3	1	1	2
4	2	1	2	3
5	3	2	3	0
6	2	2	1	1

- oats.asr1 <- asreml(yield ~ Variety*Nitrogen, random = ~ Blocks/Wplots, data=data)

```
oat.asr1
---- Wald tests:
               Df denDF   F.inc       Pr
(Intercept)       1   4.7 255.300 3.037e-05 ***
Variety          2   7.6   0.396    0.6862
Nitrogen          3  15.8  35.040 3.169e-07 ***
Variety:Nitrogen  6  16.5   2.074    0.1125
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


---- Variance components:
                              gamma component std.error  z.ratio constraint
Blocks!Blocks.var           1.568704  183.0749 175.47107 1.043334   Positive
Blocks:Wplots!Blocks.var 1.343620  156.8065 113.66641 1.379533   Positive
R!variance                   1.000000  116.7045  44.71473 2.609980   Positive
```

Balanced data：

oats.asr <- asreml(yield ~ Variety*Nitrogen,
 random = ~ Blocks/Wplots,
 data=oats)

```
oats.asr
---- Wald tests:
               Df denDF   F.inc       Pr
(Intercept)       1     5 245.100 1.932e-05 ***
Variety          2    10   1.485    0.2724
Nitrogen          3    45  37.690 2.458e-12 ***
Variety:Nitrogen  6    45   0.303    0.9322
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


---- Variance components:
                              gamma component std.error  z.ratio constraint
Blocks!Blocks.var          1.2111647  214.4771 168.83404 1.270343   Positive
Blocks:Wplots!Blocks.var 0.5989373  106.0618  67.87553 1.562593   Positive
R!variance                  1.0000000  177.0833  37.33244 4.743416   Positive
```

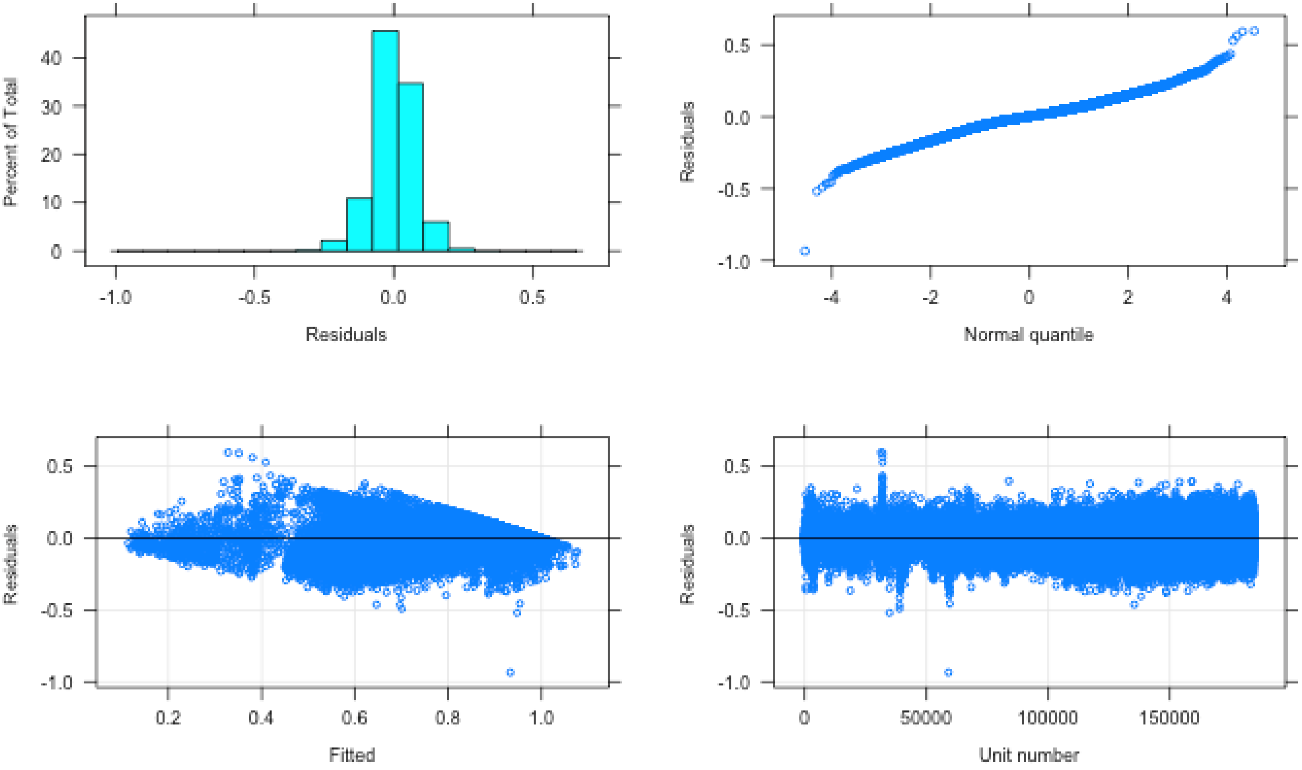
复杂模型和大数据

- BCI数据： 180，000＋记录，291个物种
- library(nlme)
- ptm <- proc.time()
- > lme1=lme(apd5~logdbh,random=~logdbh|sp+index20/index10,data=IApd_all)
- 错误于na.fail.default(list(logdbh = c(4.44500143383527, 0.182321556793955, :
-   missing values in object
- > proc.time() – ptm

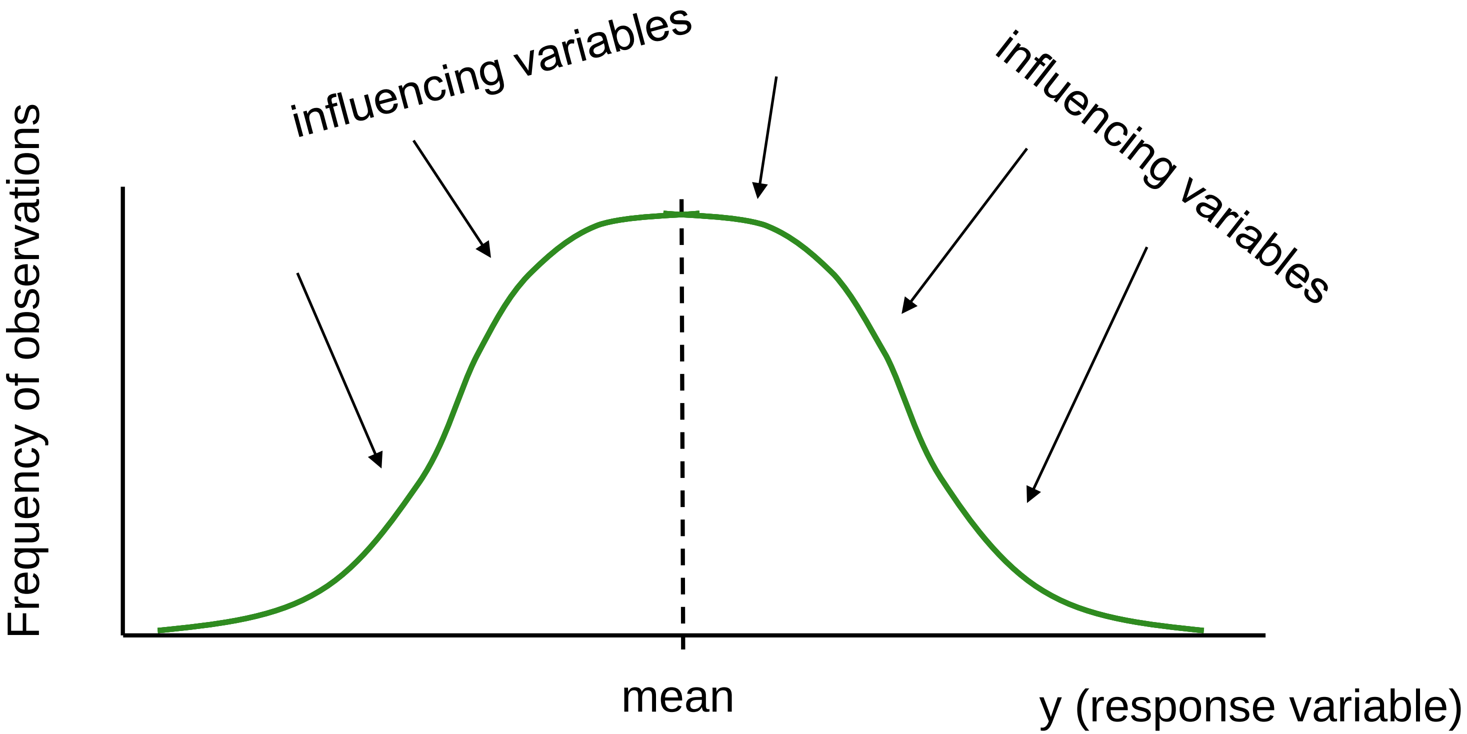
- ptm <- proc.time()
- > lm1=lmer(apd5~logdbh+(logdbh|sp)+(1|index20/index10),data=IApd_all)
- 警告信息：
- In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
-   Model failed to converge with max|grad| = 0.00991102 (tol = 0.002)
- > proc.time() - ptm
-    user system elapsed
- 84.445   1.554 86.012

复杂模型和大数据

- ptm <- proc.time()
- > lme1=asreml(apd5~logdbh,random=~sp+logdbh:sp+index20/index10,data=IApd_all,workspace=30e07)
- ASReml: Thu Aug 14 09:06:02 2014
- LogLik S2 DF wall cpu
- 379528.3291 0.0057184773 09:06:21 18.4
- 379754.3880 0.0057
- 379981.0217 0.0057
- 380146.5450 0.0057
- 380201.8266 0.0057
- 380213.5970 0.0057
- 380214.4784 0.0057
- 380214.4828 0.0057
- 380214.4831 0.0057
- 380214.4828 0.0057
- Finished on: Thu Aug 1
-
- LogLikelihood Converg
- > proc.time() - ptm
- user system elapsed
- 155.872 1.235 157.07



方差分析（anova）

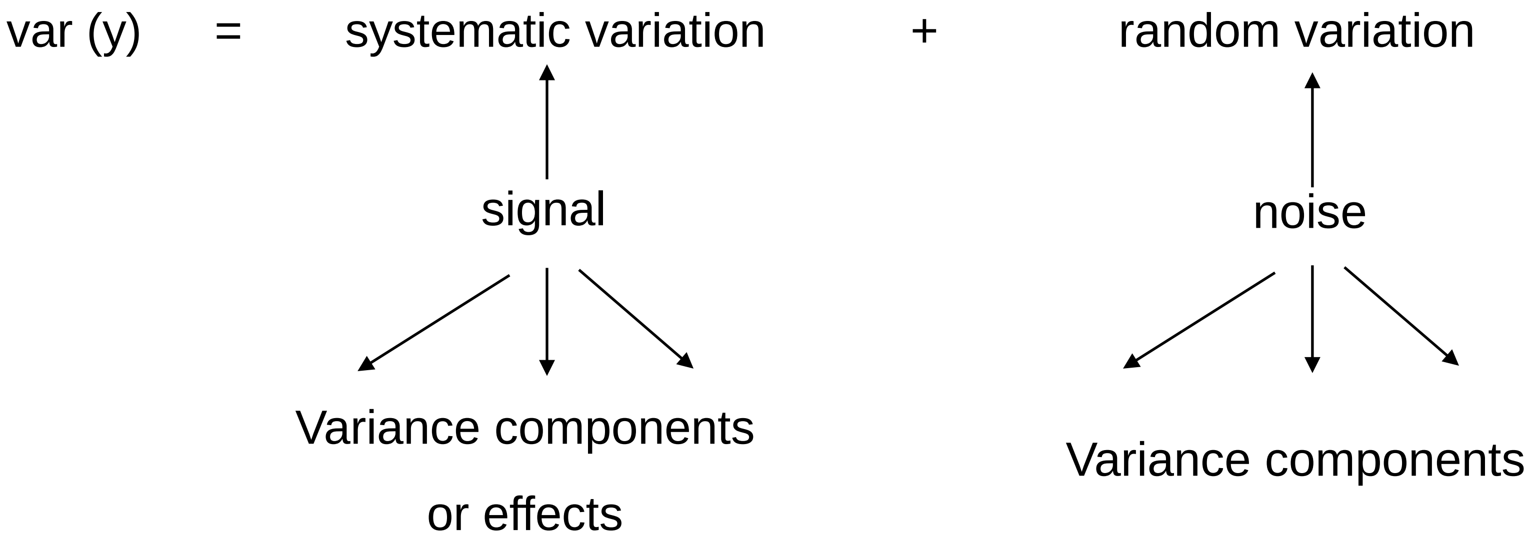


y = f (influencing variables)

$= b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots$

Goal of statistical analysis: finding pattern in data

Method of finding pattern in data



var (y) = systematic variation + random variation

signal

noise

Variance components or effects

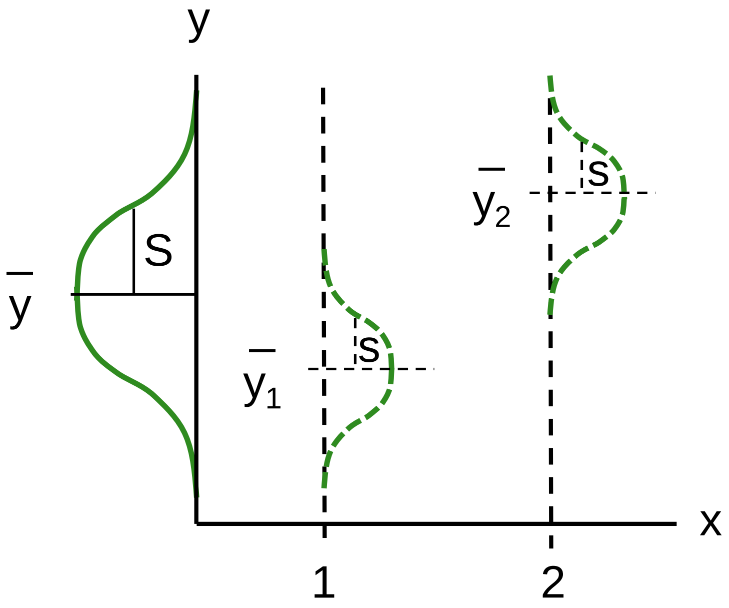
Variance components

Fitting statistical model → estimate effects b_0, b_1, etc.

Decomposing variation → estimate variance components and construct tests.

Constructing tests

We want to find out if our data belong to a single or several statistical populations. Let’s assume that we have an x-variable that allows us to make groups that are potentially representing different statistical populations:



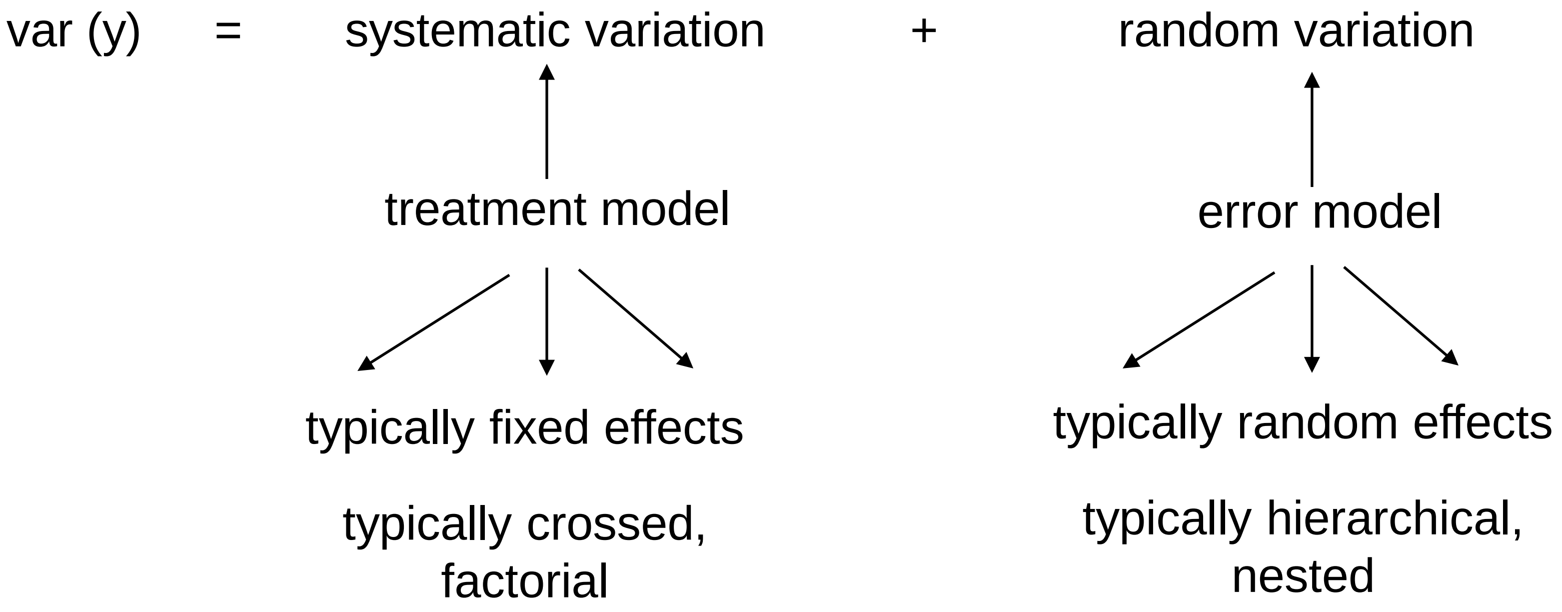
→ t =

with > 2 groups

→ F =

F < 1 strange: pair from different groups on average less difference than pair from same group!

The structure of general(ized) linear (mixed) models



var (y) = systematic variation + random variation

treatment model

typically fixed effects

typically crossed,
factorial

error model

typically random effects

typically hierarchical,
nested

Fixed effect: "all out of few", interest in each group
→ means and effects

Random effect: "few out of many", interest in variation among groups
→ extrapolation to "many"

Attention: "few" in fixed effects is less than "few" in random effects!
(see Green & Tukey 1960)

General points

Statistical model: $y = m + a + b + \dots + e$

Analysis of variance: $SS_{y\text{-}m} = SS_a + SS_b + \dots + SS_e$

Dividing by degrees of freedom:

$$MS_{total} \rightarrow MS_a, MS_b, \dots, MS_e$$

Using size of MS to test importance of term in model:

$$F = MS_a / MS_e = (s^2_e + k\, S^2_a) / s^2_e$$

Think about it this way: difference between 2 units from A = 1 and A = 2 is at least as large or larger than between 2 units from the same level of A!

Simulation of planned experiment

```
# Simulating data for precipitation experiment
#-------------------------------------------------------
# Prepare variables to be calculated later:
yvariable=blockresiduals=plotresiduals=unitresiduals=NULL

# Generate subplot variable:
subplots=seq(1:30)

# Generate factors:
plots=factor(gl(15,2)) # not yet randomly assigned, but doesn't matter for simulation

blocks=factor(gl(5,6))
treats=factor(gl(3,2,length=30),labels=c("50","100","150"))
veg=factor(gl(2,1,length=30),labels=c("grass","shrub"))
```

Simulation of planned experiment

```
# Set the mean:
mu=100
# Set starting value of random number generator
# so we always get same result:
set.seed(123)

# Make residuals with variance s2:
VCunits=25   # increase to see effect of residual variation
unitresiduals[1:30]=rnorm(30)*sqrt(VCunits)
unitresiduals=unitresiduals-mean(unitresiduals)

# Make plot and block residuals with corresponding variance components:
VCplots=9   # increase to see effect of plot variation
plotresiduals[1:15]=rnorm(15)*sqrt(VCplots)
plotresiduals=plotresiduals-mean(plotresiduals)
VCblocks=0   # this time set to zero
blockresiduals[1:5]=rnorm(5)*sqrt(VCblocks)
blockresiduals=blockresiduals-mean(blockresiduals)
```

Simulation of planned experiment (continued)

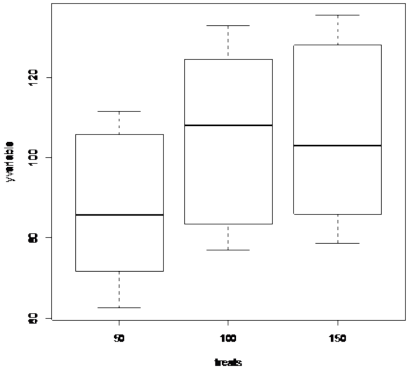
```
# Make treatment effects:
trteff=c(-10,5,5)
# Make vegetation effects:
vegeff=c(-20,20)
# Generate dependent variable:
yvariable[1:30]=mu+blockresiduals[blocks]+trteff[treats]+plotresiduals[plots]+vegeff[veg]+unitresiduals
# Show data frame:
data.frame(blocks,treats,plots,veg,yvariable)
   blocks treats plots   veg yvariable
1       1     50     1 grass  67.82666
2       1     50     1 shrub 109.47816
3       1    100      2 grass  91.25798
…
30      5    150    15 shrub 134.24261

# Consistency checks:
nlevels(treats)
[1] 3
cor(as.numeric(blocks),as.numeric(treats)) # orthogonality of factors
[1] 0

# Summary statistics:
mean(yvariable)
[1] 100
tapply(yvariable,list(treats,veg),length)
    grass shrub
50      5     5
100     5     5
150     5     5
```

```
tapply(yvariable,list(treats,veg),mean)
      grass    shrub
50  69.85477 105.9332
100 85.59894 126.8601
150 85.40344 126.3495
tapply(yvariable,blocks,length)
1 2 3 4 5
6 6 6 6 6
tapply(yvariable,blocks,mean)


 1         2         3         4         5
102.61193 101.44403  99.76383  95.72311 100.45709
# Plots:
plot(yvariable~treats)
```

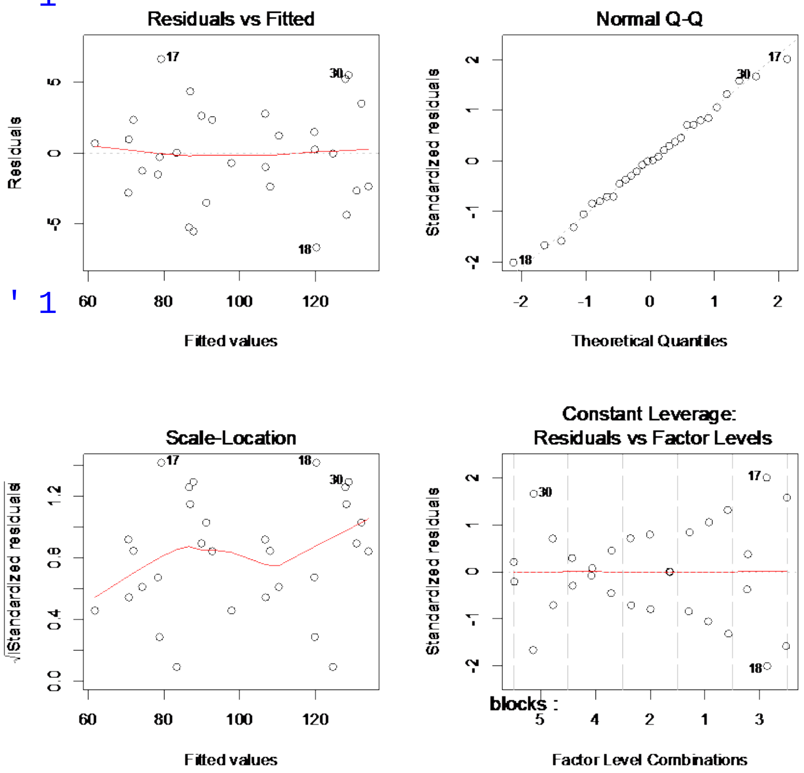


Simulation of planned experiment (continued 2)

```
# Preliminary analysis:
summary(aov(yvariable~blocks+treats+veg+treats:veg+Error(plots)))
Error: plots
          Df Sum Sq Mean Sq F value Pr(>F)
blocks     4  164.8    41.2   0.639 0.6492
treats     2 2199.0  1099.5  17.065 0.0013 **
Residuals  8  515.4    64.4
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: Within
           Df Sum Sq Mean Sq F value   Pr(>F)
veg         1  11660   11660 427.187 9.5e-11 ***
treats:veg  2     42      21   0.773   0.483
Residuals  12    328      27
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Check assumptions:
model1=lm(yvariable~blocks+treats+plots+veg+treats:veg)
par(mfrow=c(2,2))
plot(model1)
```



Simulation of planned experiment (continued 3)

```
# Mixed model analysis with ASReml:
library(asreml)
library(pascal)
model2=asreml(fixed=yvariable~treats+veg+treats:veg,random=~blocks+plots)
test.asreml(model2)  # function from Pascal for nice output
---- Wald tests:
              Df denDF     F.inc       Pr
(Intercept)  1    12       5292.0     < 2.2e-16 ***
treats         2    12        19.4       0.0001739 ***
veg            1    12       427.2      9.498e-11 ***
treats:veg   2    12        0.8       0.4831817
…
---- Stratum variances:
         df Variance    plots R!variance
plots      12 56.68498 1.999998          1
R!variance 12 27.29388 0.000000          1
---- Variance components:
                          gamma       component    std.error         z.ratio      constraint
blocks!blocks.var 5.768099e-07 1.574338e-05 6.427208e-06 2.449490   Boundary
plots!plots.var   5.384195e-01 1.469556e+01 1.284223e+01 1.144315    Positive
R!variance        1.000000e+00 2.729388e+01 1.114268e+01 2.449490    Positive
```

Simulation of Haihua’s planned experiment (continued 4)

```
# Pascal’s correction to get negative variance components:
model3=asreml.nvc(fixed=yvariable~treats+veg+treats:veg,random=~blocks+plots)
test.asreml(model3)
---- Wald tests:
              Df denDF    F.inc      Pr
(Intercept)  1     4       7282.0    1.130e-07 ***
treats        2     8       17.1       0.0013 **
veg           1    12       427.2      9.498e-11 ***
treats:veg   2    12        0.8        0.4832
…
---- Stratum variances:
         df Variance blocks plots R!variance
blocks      4 41.19585      6     2          1
plots      8 64.42960      0     2          1
R!variance 12 27.29389      0     0          1
---- Variance components:
                          gamma      component       std.error      z.ratio          constraint
blocks!blocks.var -0.1418740   -3.872292       7.238673    -0.534945     Unconstrained
plots!plots.var       0.6802936    18.567856      17.043714   1.089425     Unconstrained
R!variance            1.0000000    27.293888      11.142683  2.449490     Positive
…
```

自相关

- 空间采样数据:
- M2c<asreml(fixed=PC1~F1,rcov=~aexp(x.coord,y.coord),data=data)
- m2b<-asreml(fixed=PC1~F1,rcov=~ar1(x.coord):ar1(y.coord),data=data)

- 其他软件包:
- lme
- glmmPQL
- Gls
- Dorman, F. C. et al. 2007.Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. Ecography.

不同方法结果的差别

- lm1 <- asreml(yield ~ Variety*Nitrogen, random = ~ Blocks/Wplots, data=oats)
- lm2 <- lme(yield ~ Variety*Nitrogen, random = ~ 1|Blocks/Wplots, data=oats)
- lm3 <- lmer(yield ~ Variety*Nitrogen+(1|Blocks/Wplots), data=oats)

不同方法结果的差别

- **Asreml:**

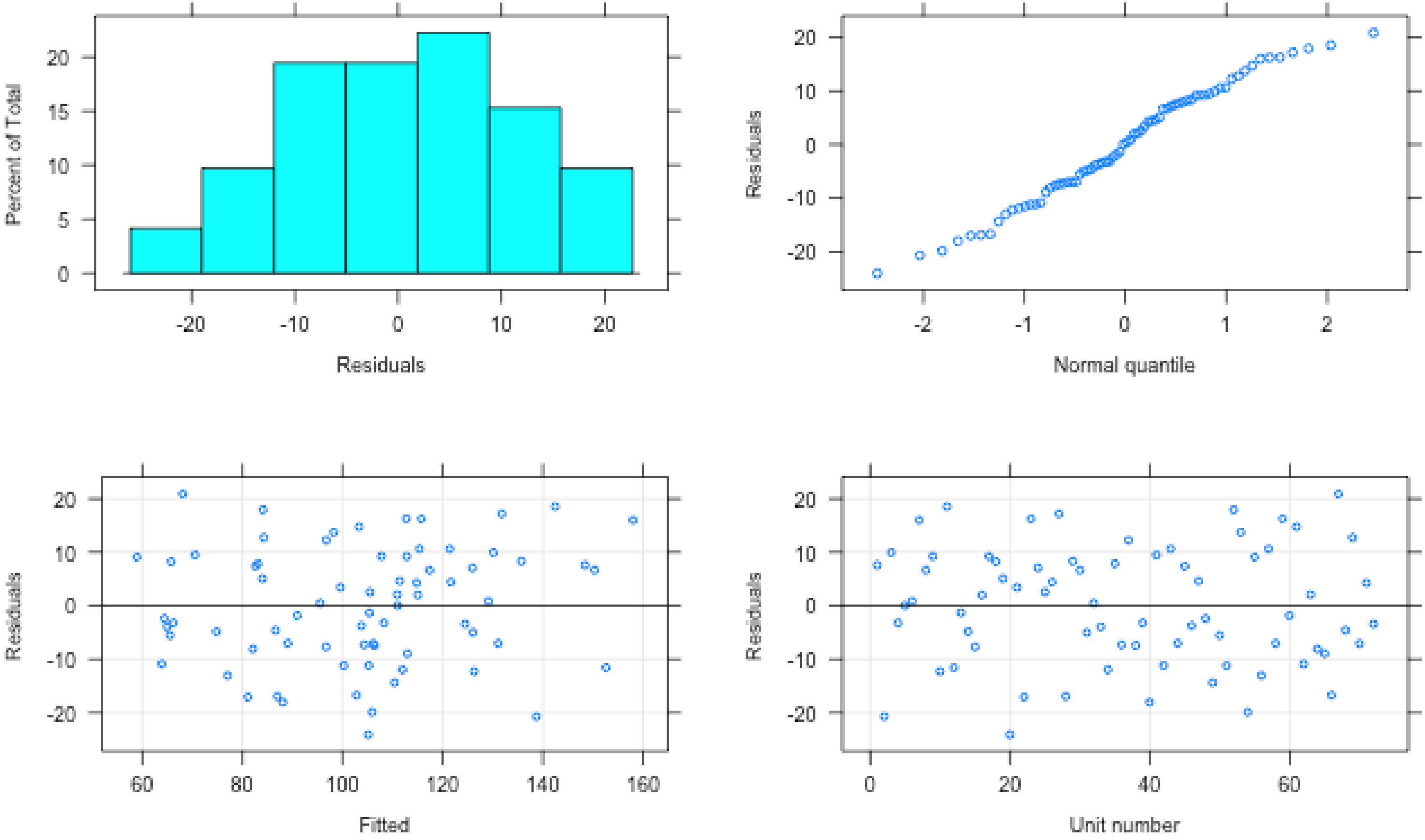
```
•  test.asreml(lm1)
•  ---- Wald tests:
•                      Df    denDF   F.inc        Pr
•  (Intercept)       1       5       245.100 1.932e-05 ***
•  Variety             2      10       1.485    0.2724
•  Nitrogen           3       45        37.690 2.458e-12 ***
•  Variety:Nitrogen  6   45   0.303    0.9322
•  Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

•  ---- Variance components:
•                                             gamma component std.error  z.ratio constraint
•  Blocks!Blocks.var                  1.2111647  214.4771 168.83404 1.270343   Positive
•  Blocks:Wplots!Blocks.var        0.5989373  106.0618  67.87553 1.562593   Positive
•  R!variance                            1.0000000  177.0833  37.33244 4.743416   Positive
```

不同方法结果的差别

```
summary(lm1,all=T)$coef.fixed
                                          solution      std error      z ratio
Variety_Marvellous:Nitrogen_0.2_cwt     3.3333333     10.865337    0.3067860
Variety_Marvellous:Nitrogen_0.4_cwt    -4.1666667     10.865337   -0.3834825
Variety_Marvellous:Nitrogen_0.6_cwt    -4.6666667     10.865337   -0.4295004
Variety_Victory:Nitrogen_0.2_cwt       -0.3333333     10.865337   -0.0306786
Variety_Victory:Nitrogen_0.4_cwt        4.6666667     10.865337    0.4295004
Variety_Victory:Nitrogen_0.6_cwt        2.1666667     10.865337    0.1994109
Nitrogen_0.2_cwt                       18.5000000      7.682954    2.4079281
Nitrogen_0.4_cwt                       34.6666667      7.682954    4.5121535
Nitrogen_0.6_cwt                       44.8333333      7.682954    5.8354293
Variety_Marvellous                      6.6666667      9.715025    0.6862223
Variety_Victory                        -8.5000000      9.715025   -0.8749334
(Intercept)                            80.0000000      9.106977    8.7844734
```

不同方法结果的差别



- VarietyVictory:Nitrogen0.6_cwt 2.16667 10.865342 45 0.199411 0.8428

总结

- Asreml能完成lmer以及lme能做大部分分析；
- Asreml对大数据以及复杂模型具有很好的拟合效率；
- Asreml能够处理negative variance component；
- Asreml对fixed effect 的检验基于wald test以及F .inc 值，比简单t检验更可靠；
- Asreml 能够直接对variance component显著度进行检验，并计算std.error，而不需要进行额外的LRT检验；
- Etc。。。

致谢

- 感谢Bernhard Schmid教授
- http://www.vsni.co.uk/