The first code snippet loads the ‘aomisc’ package and other necessary packages. Afterwards, it loads the ‘WinterWheat’ dataset and turns the ‘Year’ and ‘Block’ variables into factors.

```
library(plyr)
library(nlme)
library(aomisc)
## Loading required package: drc
## Loading required package: MASS
## Loading required package: drcData
##
## 'drc' has been loaded.
## Please cite R and 'drc' if used for a publication,
## for references type 'citation()' and 'citation('drc')'
##
## Attaching package: 'drc'
## The following objects are masked from 'package:stats':
##
##     gaussian, getInitial
data(WinterWheat)
WinterWheat$Year <- factor(WinterWheat$Year)
WinterWheat$Block <- factor(WinterWheat$Block)
head(WinterWheat, 10)
##    Plot Block Genotype Yield Year
## 1     2     1 COLOSSEO  6.73 1996
## 2     1     1    CRESO  6.02 1996
## 3    50     1   DUILIO  6.06 1996
## 4    49     1   GRAZIA  6.24 1996
## 5    63     1    IRIDE  6.23 1996
## 6    32     1 SANCARLO  5.45 1996
## 7    35     1   SIMETO  4.99 1996
## 8    33     1    SOLEX  6.08 1996
## 9   110     2 COLOSSEO  6.96 1996
## 10  137     2    CRESO  5.34 1996
```

Please, note that this is a multienvironment experiment as it is repeated in several years: each year is an ‘environment’ in itself. Furthermore, please note that the year effect (i.e. the environment effect) is of random nature: we select the years, but we cannot control the weather conditions.

Defining a linear mixed model

Dealing with the above dataset, a good candidate model for data analyses is the following linear model:

$$y_{ijk} = \mu + \gamma_{kj} + g_i + e_j + ge_{ij} + \varepsilon_{ijk}$$

where y is yield (or other trait) for the k-th block, i-th genotype and j-th environment, μ is the intercept, γ is the effect of the k-th block in the j-th environment, g is the effect of the i-th genotype, e is the effect of the j-th environment, ge is the interaction effect of the i-th genotype and j-th environment, while ε is the residual random term, which is assumed as normally distributed with variance equal to σ^2.

As I said before, the block effect, the environment effect and the 'genotype x environment' interaction are usually regarded as random. Therefore, they are assumed as normally distributed, with means equal to 0 and variances respectively equal to σ_{γ}^2, σ_e^2 and σ_{ge}^2. Indeed, the above model is a Linear Mixed Model (LMM).

Let's concentrate on σ_{ge}^2. It is clear that this value is a measure of instability: if it is high, genotypes may respond differently to different environments. In this way, each genotype can be favored in some specific environments and disfavored in some others. Shukla (1974) has suggested that we should allow σ_{ge}^2 assume a different value for each genotype and use these components as a measure of stability (stability variances). According to Shukla, a genotype is considered stable when its stability variance is lower than σ^2.

Piepho (1999) has shown that stability variances can be obtained within the mixed model framework, by appropriately coding the variance-covariance matrix for random effects. He gave SAS code to accomplish this task and, to me, it was not straightforward to transport his code into R. I finally succeeded and I though I should better share my code, just in case it helps someone save a few headaches.

Fitting a stability variance model

As we have to model the variance-covariance of random effects, we need to use the 'lme' function in the 'nlme' package (Pinheiro and Bates, 2000). The problem is that random effects are crossed and they are not easily coded with this package. After an extensive literature search, I could find the solution in the aforementioned book (at pag. 162-163) and in Galecki and Burzykowski (2013). The trick is made by appropriately using the ‘pdMat’ construct (‘pdBlocked’ and ‘pdIdent’). In the code below, I have built a block-diagonal variance-covariance matrix for random effects, where blocks and genotypes are nested within years:

```
model.mix <- lme(Yield ~ Genotype,
  random=list(Year = pdBlocked(list(pdIdent(~1),
                                    pdIdent(~Block -
                                    pdIdent(~Genotyp
  data=WinterWheat)
VarCorr(model.mix)
## Year = pdIdent(1), pdIdent(Block - 1), pdIdent(Ge
##                  Variance   StdDev
## (Intercept)      1.07314201 1.0359257
## Block1           0.01641744 0.1281306
## Block2           0.01641744 0.1281306
## Block3           0.01641744 0.1281306
## GenotypeCOLOSSEO 0.17034091 0.4127238
## GenotypeCRESO    0.17034091 0.4127238
## GenotypeDUILIO   0.17034091 0.4127238
## GenotypeGRAZIA   0.17034091 0.4127238
## GenotypeIRIDE    0.17034091 0.4127238
## GenotypeSANCARLO 0.17034091 0.4127238
## GenotypeSIMETO   0.17034091 0.4127238
## GenotypeSOLEX    0.17034091 0.4127238
## Residual         0.14880400 0.3857512
```

Wee see that the variance component for the ‘genotype x environment’ interaction is the same for all genotypes and equal to 0.170.

Allowing for a different variance component per genotype is relatively easy, by replacing ‘pdIdent()’ with ‘pdDiag()’, as shown below:

```
model.mix2 <- lme(Yield ~ Genotype,
  random=list(Year = pdBlocked(list(pdIdent(~1),
                                    pdIdent(~Block -
                                    pdDiag(~Genotype
  data=WinterWheat)
VarCorr(model.mix2)
## Year = pdIdent(1), pdIdent(Block - 1), pdDiag(Ge
##                  Variance   StdDev
## (Intercept)      0.86592829 0.9305527
## Block1           0.01641744 0.1281305
## Block2            0.01641744 0.1281305
## Block3            0.01641744 0.1281305
## GenotypeCOLOSSEO 0.10427267 0.3229128
## GenotypeCRESO    0.09588553 0.3096539
## GenotypeDUILIO   0.47612340 0.6900170
## GenotypeGRAZIA   0.15286445 0.3909788
## GenotypeIRIDE    0.11860160 0.3443858
## GenotypeSANCARLO 0.02575029 0.1604690
## GenotypeSIMETO   0.42998504 0.6557324
## GenotypeSOLEX    0.06713590 0.2591060
## Residual         0.14880439 0.3857517
```

We see that we have now different variance components and we can classify some genotypes as stable (e.g. Sancarlo, Solex and Creso) and some others as unstable (e.g. Duilio and Simeto).

Working with the means

In his original paper, Piepho (1999) also gave SAS code to analyse the means of the ‘genotype x environment’ combinations. Indeed, agronomists and plant breeders often adopt a two-steps fitting procedure: in the first step, the means across blocks are calculated for all genotypes in all environments. In the second step, these means are used to fit a stability variance model. This two-step process is less demanding in terms of computer resources and it is correct whenever the experiments are equireplicated, with no missing ‘genotype x environment’ combinations. Furthermore, we need to be able to assume similar variances within all experiments.

I would also like to give an example of this two-step analysis method. In the first step, we can use the ‘ddply()’ function in the package ‘plyr’:

```
#First step
WinterWheatM <- ddply(WinterWheat, c("Genotype", "Ye
      function(df) c(Yield = mean(df$Yield)) )
```

Once we have retreived the means for genotypes in all years, we can fit a stability variance model, although we have to use a different approach, with respect to the one we used with the whole dataset. In this case, we need to model the variance of residuals, introducing within-group (within-year) heteroscedasticity. The ‘weights’ argument can be used, together with the ‘pdIdent()’ variance function, to allow for a different variance for each genotype. See the code below.

```
#Second step
model.mixM <- lme(Yield ~ Genotype, random = ~ 1|Yea
                 weights = varIdent(form=~1|Genotype
```

The code to retrieve the within-year variances is not obvious, unfortunately. However, you can use the folllowing snippet as a guidance.

```
vF <- model.mixM$modelStruct$varStruct
sdRatios <- c(1, coef(vF, unconstrained = F))
names(sdRatios)[1] <- "COLOSSEO"
scalePar <- model.mixM$sigma
sigma2i <- (scalePar * sdRatios)^2
sigma2i
##   COLOSSEO      CRESO     DUILIO     GRAZIA
## 0.15387857 0.14548985 0.52571780 0.20246664 0.168
##     SIMETO      SOLEX
## 0.47958756 0.11673900
```

The above code outputs ‘sigma2i’, which does not contain the stability variances. Indeed, we should remove the within-experiment error (σ^2), which can only be estimated from the whole dataset. Indeed, if we take the estimate of 0.1488 (see code above), divide by three (the number of blocks) and subtract from ‘sigma2i’, we get:

```
sigma2i - model.mix2$sigma^2/3
##   COLOSSEO      CRESO     DUILIO     GRAZIA
## 0.10427711 0.09588839 0.47611634 0.15286517 0.118
##     SIMETO      SOLEX
## 0.42998610 0.06713754
```

which are the stability variances, as obtained with the analyses of the whole dataset.

Thanks for reading!

References

- Annichiarico, P., 2002. Genotype x Environment Interactions - Challenges and Opportunities for Plant Breeding and Cultivar Recommendations. Plant Production and protection paper, Food & Agriculture Organization of the United Nations (FAO), Roma.
- Gałecki, A., Burzykowski, T., 2013. Linear mixed-effects models using R: a step-by-step approach. Springer, Berlin.
- Piepho, H.-P., 1999. Stability Analysis Using the SAS System. Agronomy Journal 91, 154–160.
- Pinheiro, J.C., Bates, D.M., 2000. Mixed-Effects Models in S and S-Plus, Springer-Verlag Inc. ed. Springer-Verlag Inc., New York.
- Shukla, G.K., 1972. Some statistical aspects of partitioning genotype-environmental components of variability. Heredity 29, 237–245.