

# The wheat example from section 7.6 of the asreml manual (Butler et al. 2018)

*Chris Brien*

*19 May, 2019*

```
library(asreml, quietly=TRUE)

## Registered S3 methods overwritten by 'ggplot2':
##   method      from
##   [.quosures   rlang
##   c.quosures   rlang
##   print.quosures rlang

library(asremlPlus)
suppressMessages(library(qqplotr, quietly=TRUE))
options(width = 100)
```

## Get data available in asremlPlus

```
data(Wheat.dat)
```

## Fit the initial model

In the following a model is fitted that has the terms that would be included for a balanced lattice. In addition, a term WithinColPairs has been included to allow for extraneous variation arising between pairs of adjacent lanes. Also, separable ar1 residual autocorrelation has been included. This model represents the maximal anticipated model,

```
current.asr <- asreml(yield ~ WithinColPairs + Variety,
                      random = ~ Rep/(Row + Column) + units,
                      residual = ~ ar1(Row):ar1(Column),
                      data=Wheat.dat)
```

```
## Model fitted using the gamma parameterization.
```

```
## ASReml 4.1.0 Sun May 19 16:29:06 2019
```

##		LogLik	Sigma2	DF	wall	cpu
##	1	-724.121	23034.14	124	16:29:06	0.0
##	2	-717.415	9206.93	124	16:29:06	0.0 (2 restrained)
##	3	-694.875	26492.99	124	16:29:06	0.0 (2 restrained)
##	4	-694.160	33101.80	124	16:29:06	0.0 (1 restrained)
##	5	-692.002	36912.26	124	16:29:06	0.0 (1 restrained)
##	6	-691.789	46701.51	124	16:29:06	0.0 (2 restrained)
##	7	-691.834	46208.51	124	16:29:06	0.0 (1 restrained)
##	8	-691.775	47698.26	124	16:29:06	0.0
##	9	-691.771	47041.85	124	16:29:06	0.0

```
## Warning in asreml(yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Some components changed
## by more than 1% on the last iteration.
```

The warning from asreml is probably dies to a bound term.

Initialize a testing sequence by loading the current fit into an asrtests object

```
current.asrt <- as.asrtests(current.asr, NULL, NULL)
```

```
## Calculating denominator DF
```

Check for and remove any boundary terms

```
current.asrt <- rmboundary(current.asrt)
summary(current.asrt$asreml.obj)$varcomp
```

```
##               component      std.error      z.ratio bound %ch
## Rep:Row          4.293282e+03 3.199458e+03  1.3418779    P 0.0
## Rep:Column       1.575689e+02 1.480357e+03  0.1064398    P 0.7
## units           5.742689e+03 1.652457e+03  3.4752438    P 0.0
## Row:Column!R     4.706787e+04 2.515832e+04  1.8708669    P 0.0
## Row:Column!Row!cor 7.920301e-01 1.014691e-01  7.8056280    U 0.0
## Row:Column!Column!cor 8.799559e-01 7.370402e-02 11.9390486    U 0.0
```

```
print(current.asrt, which = "testsummary")
```

```
##
##
## Sequence of model terms whose status in the model has been investigated
##
## terms DF denDF p action
## 1 Rep 1 NA NA Boundary
```

Rep has been removed because it has been constrained to zero. Following the recommendation of Littell et al. (2006, p. 150), the bound on all variance components is set to unconstrained (U) so as to avoid bias in the estimate of the residual variance.

Unbind Rep, Row and Column components and reload into an asrtests object

```
current.asr <- setvarianceterms(current.asr$call,
                                terms = c("Rep", "Rep:Row", "Rep:Column"),
                                bounds = "U")
```

```
## Model fitted using the gamma parameterization.
```

```
## ASReml 4.1.0 Sun May 19 16:29:07 2019
```

```
##      LogLik      Sigma2      DF      wall      cpu
## 1      -724.121      23034.14      124 16:29:07      0.0
## 2      -717.415       9206.93      124 16:29:07      0.0 (2 restrained)
## 3      -694.875      26492.99      124 16:29:07      0.0 (2 restrained)
## 4      -693.974      33129.65      124 16:29:07      0.0 (1 restrained)
## 5      -692.886      39662.12      124 16:29:07      0.0
## 6      -691.428      53103.83      124 16:29:07      0.0
## 7      -691.239      48092.17      124 16:29:07      0.0
## 8      -691.181      47278.94      124 16:29:07      0.0
## 9      -691.171      46850.98      124 16:29:07      0.0
## 10     -691.170      46690.46      124 16:29:07      0.0
```

```
## Warning in asreml(fixed = yield ~ WithinColPairs + Variety, random = ~Rep/(Row + : Some components
```

```
## changed by more than 1% on the last iteration.
current.asrt <- as.asrtests(current.asr, NULL, NULL)

## Calculating denominator DF
current.asrt <- rmboundary(current.asrt)
summary(current.asrt$asreml.obj)$varcomp

##              component      std.error   z.ratio bound %ch
## Rep          -2462.3785859 1.191435e+03 -2.066734    U 0.2
## Rep:Row       5012.4021416 3.396848e+03  1.475604    U 0.1
## Rep:Column    920.5936392 1.704008e+03  0.540252    U 1.1
## units        5964.9099379 1.608792e+03  3.707695    P 0.1
## Row:Column!R  46690.4620402 2.731906e+04  1.709080    P 0.0
## Row:Column!Row!cor 0.8152180 9.988929e-02  8.161216    U 0.1
## Row:Column!Column!cor 0.8857252 7.487875e-02 11.828793    U 0.0
print(current.asrt, which = "testsummary")

##
##
## Sequence of model terms whose status in the model has been investigated
##
## [1] terms DF      denDF p      action
## <0 rows> (or 0-length row.names)
print(current.asrt, which = "pseudoanova")

##
##
## ##### Pseudo-anova table for fixed terms
##
##
## Wald tests for fixed effects.
## Response: yield
##
##              Df denDF   F.inc    Pr
## (Intercept)    1   1.7 153.400 0.0115
## WithinColPairs 1  15.6   2.543 0.1308
## Variety        24  76.1  10.110 0.0000
```

Now the Rep component estimate is negative.

The `test.summary` output shows that no changes have been made to the model loaded using `as.asrtests`. The pseudo-anova table shows that Varieties are highly significant ( $p < 0.001$ )

## Check term for within Column pairs (a post hoc covariate)

```
current.asrt <- testranfix(current.asrt, term = "WithinColPairs",
                           drop.fix.ns=TRUE)

## Calculating denominator DF

## Warning in asreml(fixed = yield ~ Variety, random = ~Rep/(Row + Column) + : Some components changed
## by more than 1% on the last iteration.

## Calculating denominator DF
```

Generally, to determine what has been tested between two fits using `asreml` involves comparing two `asreml` calls and deciding what is different. For example what is the difference between the `asreml` call to fit the initial model and the following call?

```
current.asr <- asreml(yield ~ Variety,
  `random = ~ Rep/(Row + Column) + units,
  residual = ~ ar1(Row):ar1(Column),
  data=Wheat.dat)`
```

On the other hand, it is clear from the `testtranfix` call that the term `withinColPAirs` is being tested.

## Test nugget term

The nugget term represents non-spatial variance, such as measurement error. It is fitted using the `asreml` reserved word `units`.

```
current.asrt <- testtranfix(current.asrt, "units", positive=TRUE)
```

```
## Warning in asreml(fixed = yield ~ Variety, random = ~Rep + Rep:Row + Rep:Column, : Some components
## changed by more than 1% on the last iteration.
```

## Test Row autocorrelation

We begin testing the autocorrelation by dropping the Row autocorrelation.

```
current.asrt <- testresidual(current.asrt, "~ Row:ar1(Column)",
  label="Row autocorrelation",
  simpler=TRUE)
```

```
## Warning in asreml(fixed = yield ~ Variety, random = ~Rep/(Row + Column) + : Log-likelihood not
## converged
```

```
## Warning in asreml(fixed = yield ~ Variety, random = ~Rep/(Row + Column) + : Some components changed
## by more than 1% on the last iteration.
```

```
## Warning in newfit.asreml(asreml.obj, residual. = term.form, trace = trace, :
```

## Test Column autocorrelation (depends on whether Row autocorrelation retained)

The function `getTestPvalue` is used to get the p-value for the Row autocorrelation test. If it is significant then the Column autocorrelation is tested by dropping the Column autocorrelation, while retaining the Row autocorrelation. Otherwise the model with just Row autocorrelation, whose fit is returned via `current.asrt` after the test, is compared to one with no autocorrelation.

```
(p <- getTestPvalue(current.asrt, label = "Row autocorrelation"))
```

```
## [1] 4.654398e-06
```

```
{ if (p <= 0.05)
  current.asrt <- testresidual(current.asrt, "~ ar1(Row):Column",
    label="Col autocorrelation",
    simpler=TRUE, update=FALSE)
```

```

else
  current.asrt <- testresidual(current.asrt, "~ Row:Column",
                              label="Col autocorrelation",
                              simplifier=TRUE, update=FALSE)
}

```

```

## Warning in DFdiff(bound.h1, bound.h0, DF = DF, bound.exclusions = bound.exclusions): There were a to
## The following bound terms occur in only one of the models compared and so were discounted:
## Row:Column!Row!cor

```

## Output the results

```
print(current.asrt, which = "test")
```

```

##
##
## Sequence of model terms whose status in the model has been investigated
##
##          terms DF denDF      p          action
## 1 WithinColPairs 1 15.6 0.1308          Dropped
## 2 units          1    NA 0.0006          Retained
## 3 Row autocorrelation 1    NA 0.0000 Unswapped - new unconverged
## 4 Col autocorrelation 2    NA 0.0000          Unswapped

```

```

info <- infoCriteria(current.asrt$asreml.obj)
summary(current.asrt$asreml.obj)$varcomp

```

```

##          component      std.error    z.ratio bound %ch
## Rep          -2392.1643977 1.194200e+03 -2.0031528    U 0.4
## Rep:Row          5035.7265976 3.405769e+03  1.4785872    U 0.3
## Rep:Column       762.1738644 1.612310e+03  0.4727218    U 1.3
## units          5933.5126842 1.610749e+03  3.6836971    P 0.1
## Row:Column!R     45973.2573646 2.635502e+04  1.7443833    P 0.0
## Row:Column!Row!cor    0.8101706 9.994728e-02  8.1059792    U 0.1
## Row:Column!Column!cor 0.8846993 7.502816e-02 11.7915640    U 0.0

```

The `test.summary` shows is that the model with Row and without Column autocorrelation failed to converge. It is also clear that no changes were made to the variance terms.

The `asreml.obj` in `current.asrt` contains the model selected by the testing process.

## Get current fitted asreml object and update to include standardized residuals

```

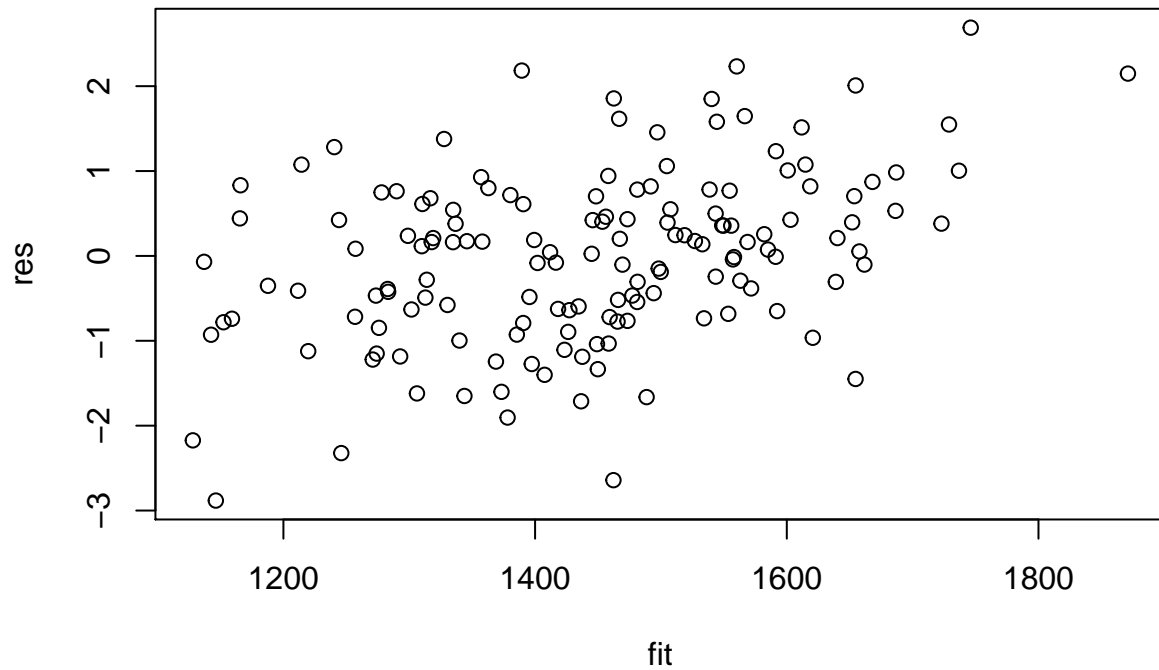
current.asr <- current.asrt$asreml.obj
current.asr <- update(current.asr, aom=TRUE)
Wheat.dat$res <- residuals(current.asr, type = "stdCond")
Wheat.dat$fit <- fitted(current.asr)

```

Do diagnostic checking

Do residuals-versus-fitted values plot

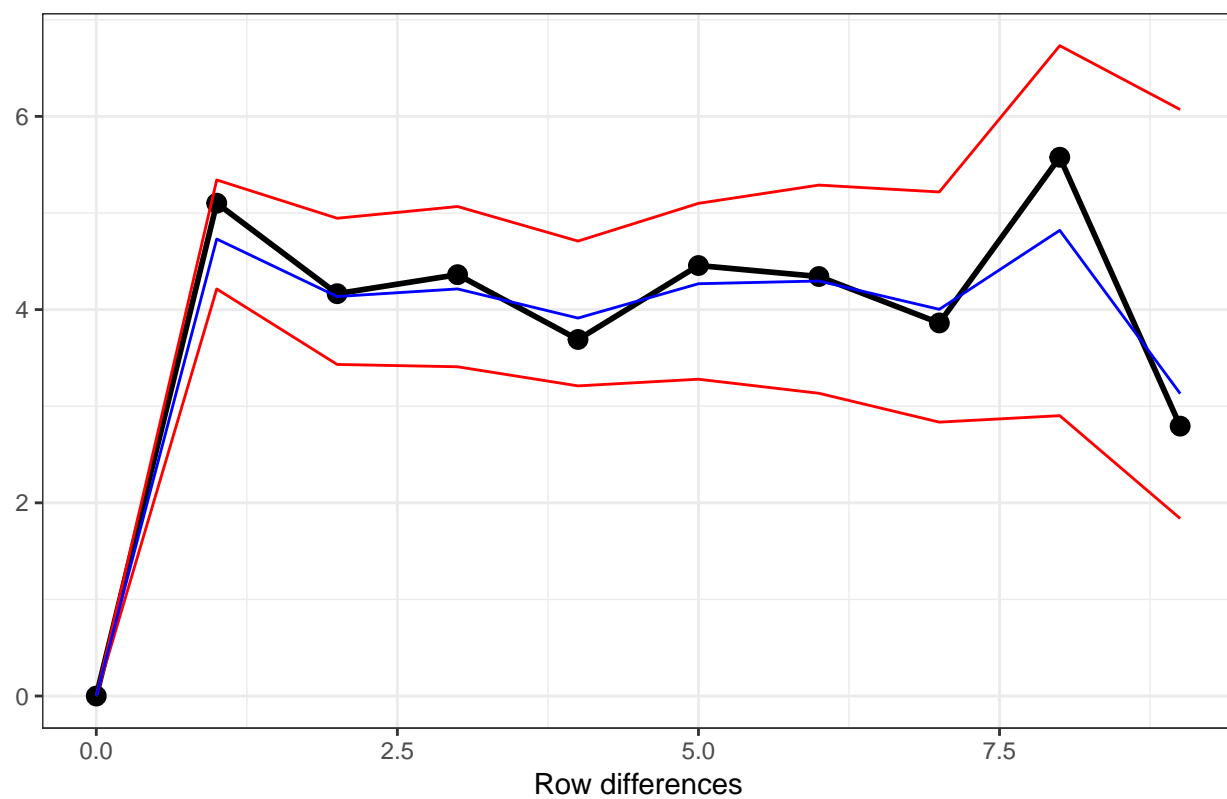
```
with(Wheat.dat, plot(fit, res))
```



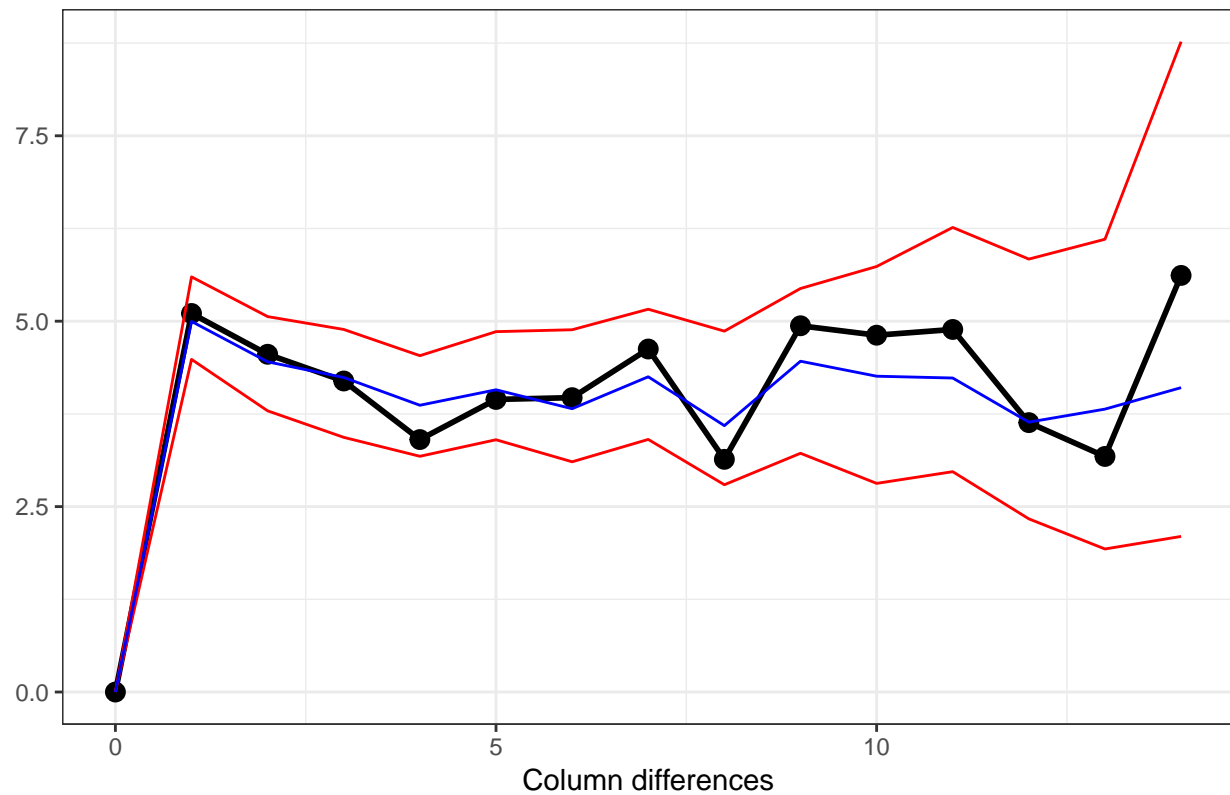
Plot varifaces

```
variofaces(current.asr, V=NULL, units="addtores",  
            maxiter=50, update = FALSE)
```

Variogram face of Standardized conditional residuals for Row



Variogram face of Standardized conditional residuals for Column



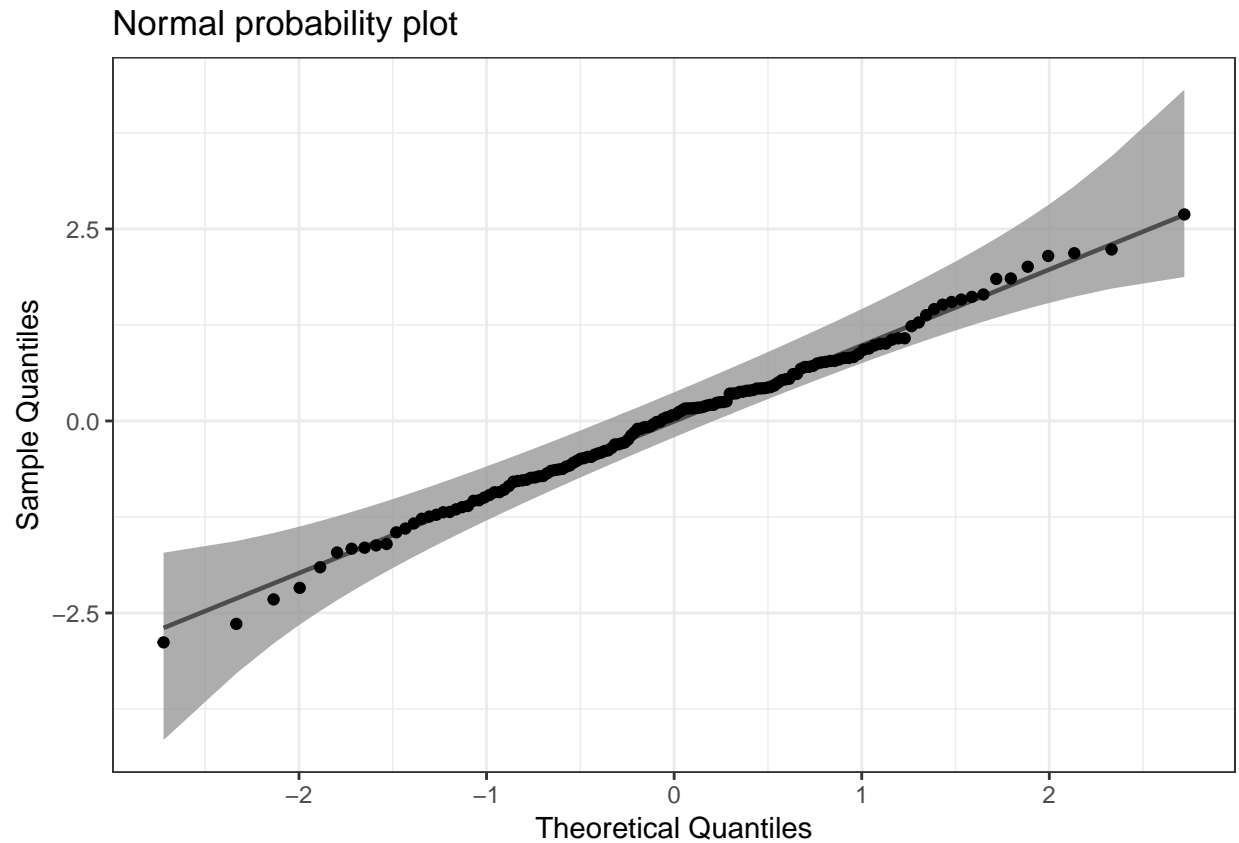
The variofaces are the lag 1 plots of the sample semivariogram with simulated confidence envelopes (Stefanova et al., 2009).

### Plot normal quantile plot

The plot is obtained using the `ggplot` function with extensions available from the `qqplotr` package (Cheng, 2018).

```
ggplot(data = Wheat.dat, mapping = aes(sample = res)) +
  stat_qq_band(bandType = "ts") + stat_qq_line() + stat_qq_point() +
  labs(x = "Theoretical Quantiles", y = "Sample Quantiles",
       title = "Normal probability plot") +
  theme(plot.title = element_text(size = 12, face = "bold")) + theme_bw()
```





Get Variety predictions and all pairwise prediction differences and p-values

```
Var.diffs <- predictPlus(classify = "Variety",
  asreml.obj=current.asr,
  error.intervals="halfLeast",
  wald.tab=current.asrt$wald.tab,
  sortFactor = "Variety",
  tables = "predictions")

##
##
## ##### Predictions for yield from Variety
##
##
## Notes:
## - The predictions are obtained by averaging across the hypertable
##   calculated from model terms constructed solely from factors in
##   the averaging and classify sets.
## - Use 'average' to move ignored factors into the averaging set.
## - The ignored set: Rep,Row,Column,units
## - Variety is included in this prediction
## - (Intercept) is included in this prediction
## - units is ignored in this prediction
##
##
```

##	Variety	predicted.value	standard.error	upper.halfLeastSignificant.limit
## 10	10	1168.989	120.4773	1228.315
## 1	1	1242.750	119.8109	1302.076
## 9	9	1257.137	119.9713	1316.463
## 16	16	1285.718	119.9405	1345.045
## 14	14	1293.527	119.9232	1352.853
## 23	23	1313.653	120.2934	1372.979
## 11	11	1322.159	120.1969	1381.485
## 7	7	1374.447	120.2412	1433.773
## 3	3	1394.070	120.4037	1453.396
## 4	4	1410.980	120.1060	1470.306
## 12	12	1444.557	120.6039	1503.883
## 8	8	1453.396	120.5945	1512.723
## 15	15	1458.383	120.4351	1517.709
## 5	5	1473.782	120.4460	1533.108
## 17	17	1487.828	120.2901	1547.154
## 6	6	1498.294	120.1194	1557.620
## 21	21	1517.121	120.2267	1576.448
## 2	2	1520.466	119.6327	1579.792
## 24	24	1533.769	120.3000	1593.095
## 18	18	1541.148	120.3669	1600.474
## 25	25	1575.794	120.5147	1635.121
## 22	22	1610.482	120.3286	1669.808
## 13	13	1610.762	120.4580	1670.088
## 20	20	1627.971	120.2333	1687.297
## 19	19	1652.992	120.3440	1712.318
##		lower.halfLeastSignificant.limit	est.status	
## 10		1109.663	Estimable	
## 1		1183.424	Estimable	
## 9		1197.811	Estimable	
## 16		1226.392	Estimable	
## 14		1234.200	Estimable	
## 23		1254.327	Estimable	
## 11		1262.833	Estimable	
## 7		1315.120	Estimable	
## 3		1334.744	Estimable	
## 4		1351.654	Estimable	
## 12		1385.231	Estimable	
## 8		1394.070	Estimable	
## 15		1399.057	Estimable	
## 5		1414.456	Estimable	
## 17		1428.502	Estimable	
## 6		1438.968	Estimable	
## 21		1457.795	Estimable	
## 2		1461.139	Estimable	
## 24		1474.442	Estimable	
## 18		1481.821	Estimable	
## 25		1516.468	Estimable	
## 22		1551.156	Estimable	
## 13		1551.436	Estimable	
## 20		1568.645	Estimable	
## 19		1593.666	Estimable	
##				
##				

```
## LSD values
##
## minimum LSD = 114.0128
##
## mean LSD = 118.6523
##
## maximum LSD = 123.3577
##
## (sed range / mean sed = 0.0788 )
```

We have set `error.intervals` to `halfLeast` so that the limits for  $\pm 0.5LSD$  are calculated. When these are plotted overlapping error bars indicate predictions that are not significant, while those that do not overlap are significantly different (Snee, 1981).

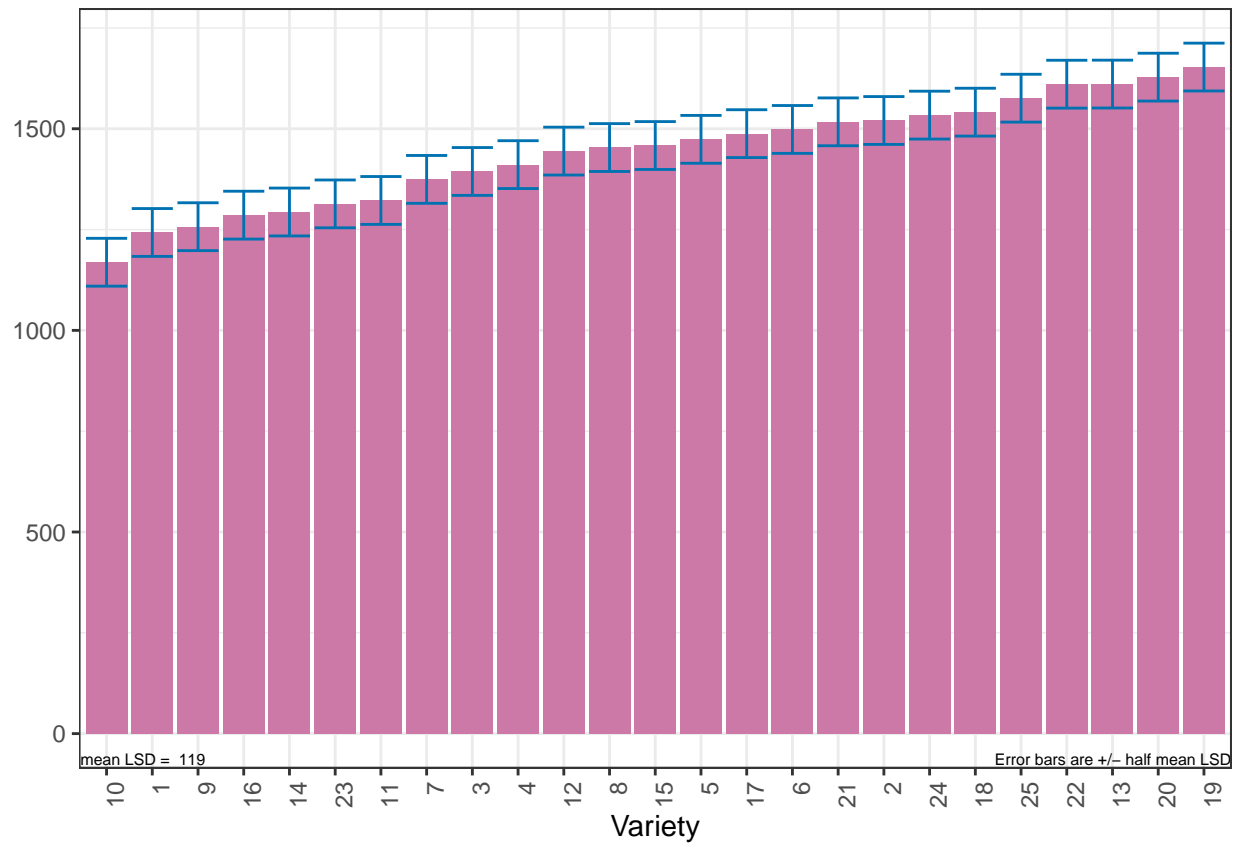
Also set was `sortFactor`, so that the results would be ordered for the values of the predictions for Variety.

The function `predictPlus` returns an `alldiffs` object, a list consisting of the following components:

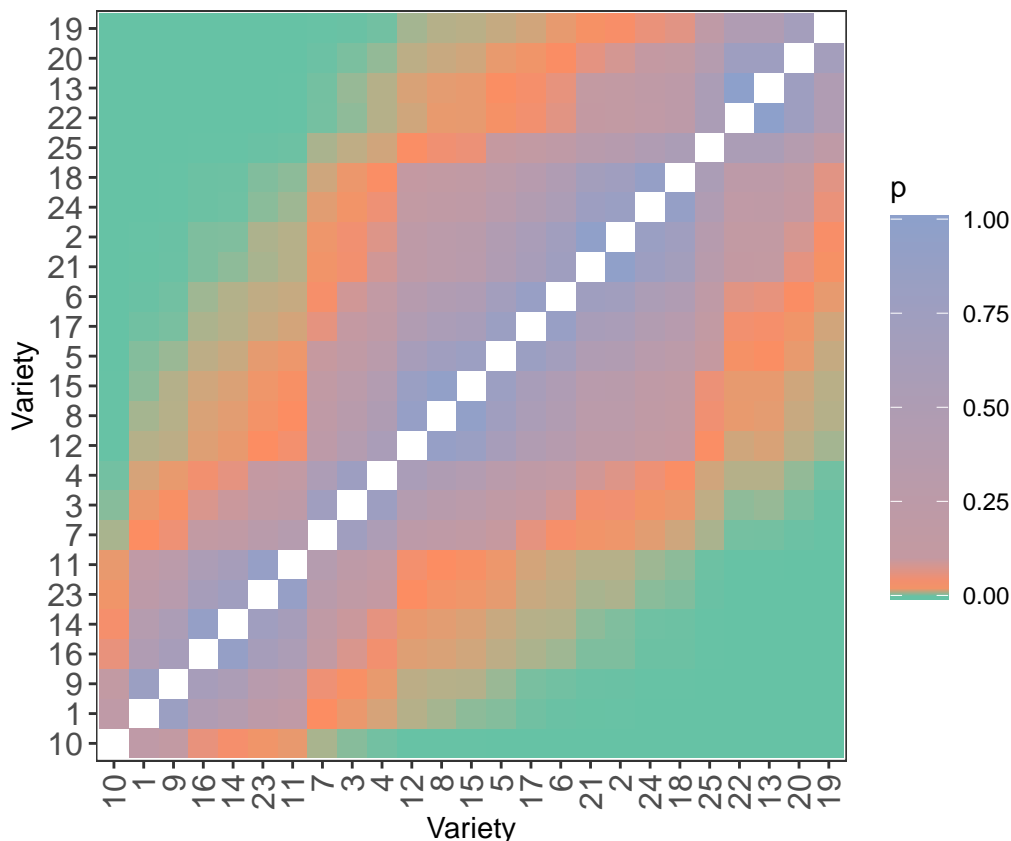
- `predictions`: the predictions, their standard errors and error intervals;
- `vcov`: the variance matrix of the predictions;
- `differences`: all pairwise differences between the predictions,
- `p.differences`: p-values for all pairwise differences between the predictions;
- `sed`: the standard errors of all pairwise differences between the predictions;
- `LSD`: the mean, minimum and maximum LSDs.

**Plot the Variety predictions, with halfLSD intervals, and the p-values**

```
plotPredictions(Var.diffs$predictions,
  classify = "Variety", y = "predicted.value",
  error.intervals = "half")
```



```
plotPvalues(Var.diffs)
```



## References

- Butler, D. G., Cullis, B. R., Gilmour, A. R., Gogel, B. J. and Thompson, R. (2018). *ASReml-R Reference Manual Version 4*. VSN International Ltd, <http://asreml.org>.
- Cheng, R. (2018) QTLRel: tools for mapping of quantitative traits of genetically related individuals and calculating identity coefficients from pedigrees. <https://CRAN.R-project.org/package=QTLRel>.
- Gilmour, A. R., Thompson, R., & Cullis, B. R. (1995). Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models. *Biometrics*, **51**, 1440–1450.
- Snee, R. D. (1981). Graphical Display and Assessment of Means. *Biometrics*, **37**, 835–836.
- Stefanova, K. T., Smith, A. B. & Cullis, B. R. (2009) Enhanced diagnostics for the spatial analysis of field trials. *Journal of Agricultural, Biological, and Environmental Statistics*, **14**, 392–410.