Package 'statgenGxE'

November 12, 2025

Type Package

Title Genotype by Environment (GxE) Analysis

Version 1.0.11

Date 2025-11-12

Description Analysis of multi environment data of plant breeding experiments
following the analyses described in Malosetti, Ribaut,
and van Eeuwijk (2013), <doi:10.3389/fphys.2013.00044>.
One of a series of statistical genetic packages for streamlining the analysis of
typical plant breeding experiments developed by Biometris.
Some functions have been created to be used in conjunction with the R
package 'asreml' for the 'ASReml' software, which can be obtained upon
purchase from 'VSN' international (<https://vsni.co.uk/software/asreml-r/>).

License GPL

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Depends R (>= 3.3)

Imports emmeans, ggplot2 (>= 3.3.0), gridExtra, knitr, lme4, methods,
rlang, statgenSTA (>= 1.0.6), xtable

Suggests asreml(>= 3.0), rmarkdown, testthat

SystemRequirements pdflatex

VignetteBuilder knitr

URL <https://biometris.github.io/statgenGxE/index.html>,
<https://github.com/Biometris/statgenGxE/>

BugReports <https://github.com/Biometris/statgenGxE/issues>

NeedsCompilation no

Author Bart-Jan van Rossum [aut, cre] (ORCID:
<https://orcid.org/0000-0002-8673-2514>),
Fred van Eeuwijk [ctb] (ORCID: <https://orcid.org/0000-0003-3672-2921>),
Martin Boer [ctb] (ORCID: <https://orcid.org/0000-0002-1879-4588>),

Marcos Malosetti [ctb] (ORCID: <https://orcid.org/0000-0002-8150-1397>),
Daniela Bustos-Korts [ctb] (ORCID:
 <https://orcid.org/0000-0003-3827-6726>),
Emilie J Millet [ctb] (ORCID: <https://orcid.org/0000-0002-2913-4892>),
Joao Paulo [ctb] (ORCID: <https://orcid.org/0000-0002-4180-0763>),
Maikel Verouden [ctb] (ORCID: <https://orcid.org/0000-0002-4893-3323>),
Willem Kruijer [ctb] (ORCID: <https://orcid.org/0000-0001-7179-1733>),
Ron Wehrens [ctb] (ORCID: <https://orcid.org/0000-0002-8798-5599>),
Choazhi Zheng [ctb] (ORCID: <https://orcid.org/0000-0001-6030-3933>),
Turbet Delof Michel [ctb],
Rameau Jean-Francois [ctb]

Contents

Index   **43**

---

correlations   *Compute different types of correlations.*

Description

Compute three types of correlations for models fitted with a nesting factor.

- correlation between scenarios or environment types:

$$\sigma_G^2/(\sigma_G^2 + \sigma_{GS}^2)$$

- correlation between trials within scenarios or environment types:

$$(\sigma_G^2 + \sigma_{GS}^2)/(\sigma_G^2 + \sigma_{GS}^2 + \sigma_E^2)$$

- correlation between trials that belong to different scenarios/environment types:

$$\sigma_G^2/(\sigma_G^2 + \sigma_{GS}^2 + \sigma_E^2)$$

In these formulas the σ terms stand for the standard deviations of the respective model terms. So σ_S is the standard deviation for the scenario term in the model, σ_{GS} for the standard deviation of the genotype by scenario term and σ_E corresponds to the residual standard deviation.

Usage

```
correlations(varComp)
```

Arguments

varComp   An object of class varComp.

Value

A list with three correlations.

See Also

Other Mixed model analysis: `CRDR()`, `diagnostics()`, `gxeVarComp()`, `herit()`, `plot.varComp()`, `predict.varComp()`, `vc()`

CRDR	*Calculate the correlated response to selection*

Description

Calculate the correlated response to selection (CRDR) based on the fitted model. The CRDR is calculated as described by Atlin et al. E.g. for a model with trials nested within scenarios, which has a random part that looks like this: genotype + genotype:scenario + genotype:scenario:trial the CRDR is calculated as:

$$H1 = \sigma_G^2/(\sigma_G^2 + \sigma_S^2/s + \sigma_{ST}^2/st + \sigma_E^2/str)$$

$$H2 = (\sigma_G^2 + \sigma_S^2)/(\sigma_G^2 + \sigma_S^2 + \sigma_{ST}^2/st + \sigma_E^2/str)$$

$$CRDR = (\sigma_G^2/(\sigma_G^2 + \sigma_S^2)) * sqrt(H1/H2)$$

In these formulas the σ terms stand for the standard deviations of the respective model terms, and the lower case letters for the number of levels for the respective model terms. So σ_S is the standard deviation for the scenario term in the model and s is the number of scenarios. σ_E corresponds to the residual standard deviation and r to the number of replicates.

Usage

```
CRDR(varComp)
```

Arguments

varComp	An object of class varComp.

References

Atlin, G. N., Baker, R. J., McRae, K. B., & Lu, X. (2000). Selection response in subdivided target regions. Crop Science, 40(1), 7–13. doi:10.2135/cropsci2000.4017

See Also

Other Mixed model analysis: correlations(), diagnostics(), gxeVarComp(), herit(), plot.varComp(), predict.varComp(), vc()

---

diagnostics	*Get diagnostics for an object of class varComp*

Description

Get the diagnostics for the model fitted. This will print a table of combinations missing in the data. For each random factor in the model a table is printed.

Usage

```
diagnostics(varComp)
```

Arguments

varComp   An object of class varComp.

Value

A list of tables is invisibly returned.

See Also

Other Mixed model analysis: `CRDR()`, `correlations()`, `gxeVarComp()`, `herit()`, `plot.varComp()`, `predict.varComp()`, `vc()`

Examples

```
## Fit a mixed model.
geVarComp <- gxeVarComp(TD = TDMaize, trait = "yld")

## Display diagnostics.
diagnostics(geVarComp)
```

---

dropsPheno	*DROPS data set*

Description

This dataset comes from the European Union project DROPS (DROught-tolerant yielding PlantS). A panel of 256 maize hybrids was grown with two water regimes (irrigated or rainfed), in seven fields in 2012 and 2013, respectively, spread along a climatic transect from western to eastern Europe, plus one site in Chile in 2013. This resulted in 28 experiments defined as the combination of one year, one site and one water regime, with two and three repetitions for rainfed and irrigated treatments, respectively. A detailed environmental characterisation was carried out, with hourly

records of micrometeorological data and soil water status, and associated with precise measurement of phenology. Grain yield and its components were measured at the end of the experiment. 10 experiments have been selected from the full data set, two for each of the five main environmental scenarios that were identified in the data. The scenarios have been added to the data as well as a classification of the genotypes in four genetic groups.

Usage

    dropsPheno

Format

An object of class `data.frame` with 2460 rows and 20 columns.

Details

The data.frame contains the genotypic means (Best Linear Unbiased Estimators, BLUEs), with one value per experiment (Location × year × water regime) per genotype.
A data.frame with 2460 rows and 19 columns.

Experiment experiments ID described by the three first letters of the city's name followed by the year of experiment and the water regime with W for watered and R for rain-fed.

parent1 identifier of donor dent line

Code_ID, Variety_ID, Accession_ID identifier of the genotype

geno.panel project in which the genetic material was generated

grain.yield genotypic mean for yield adjusted at 15\ in ton per hectare (t ha^-1)

grain.number genotypic mean for number of grain per square meter

grain.weight genotypic mean for individual grain weight in milligram (mg)

anthesis genotypic mean for male flowering (pollen shed), in thermal time cumulated since emergence (d_20°C)

silking genotypic mean for female flowering (silking emergence), in thermal time cumulated since emergence (d_20°C)

plant.height genotypic mean for plant height, from ground level to the base of the flag leaf (highest) leaf in centimeter (cm)

tassel.height genotypic mean for plant height including tassel, from ground level to the highest point of the tassel in centimeter (cm)

ear.height genotypic mean for ear insertion height, from ground level to ligule of the highest ear leaf in centimeter (cm)

year year in which the experiment was performed

loc location where the experiment was performed, a three letter abbreviation

scenarioWater water scenario for the experiment, well watered (WW) or water deficit (WD)

scenarioTemp temperature scenario for the experiment, Cool, Hot or Hot(Day)

scenarioFull the full scenario for the experiment, a combination of scenarioWater and scenarioTemp

geneticGroup the genetic group to which the genotype belongs

Source

doi:10.15454/IASSTN

References

Millet, E. J., Pommier, C., et al. (2019). A multi-site experiment in a network of European fields for assessing the maize yield response to environmental scenarios (Data set). doi:10.15454/IASSTN

`fitted.AMMI`	*Extract fitted values.*

Description

Extract the fitted values for an object of class AMMI.

Usage

```
## S3 method for class 'AMMI'
fitted(object, ...)
```

Arguments

`object`	An object of class AMMI
`...`	Not used.

Value

A data.frame with fitted values.

See Also

Other AMMI: `gxeAmmi()`, `plot.AMMI()`, `report.AMMI()`, `residuals.AMMI()`

Examples

```
## Run AMMI analysis on TDMaize.
geAmmi <- gxeAmmi(TD = TDMaize, trait = "yld")

## Extract fitted values.
fitAmmi <- fitted(geAmmi)
head(fitAmmi)
```

---

fitted.FW   *Extract fitted values.*

Description

Extract the fitted values for a fitted Finlay-Wilkinson model.

Usage

```
## S3 method for class 'FW'
fitted(object, ...)
```

Arguments

object   An object of class FW
...   Not used.

Value

A data.frame with fitted values.

See Also

Other Finlay-Wilkinson: [gxeFw](), [plot.FW](), [report.FW](), [residuals.FW]()

Examples

```
## Run Finlay-Wilkinson analysis.
geFW <- gxeFw(TD = TDMaize, trait = "yld")

## Extract fitted values.
fitFW <- fitted(geFW)
head(fitFW)
```

---

fitted.varCov   *Extract fitted values.*

Description

Extract the fitted values for an object of class varCov.

Usage

```
## S3 method for class 'varCov'
fitted(object, ...)
```

Arguments

object   An object of class varCov

...   Not used.

Value

A data.frame with fitted values.

See Also

Other varCov: gxeVarCov(), plot.varCov(), report.varCov(), residuals.varCov()

Examples

```
if (requireNamespace("asreml", quietly = TRUE)) {
  ## Select the best variance-covariance model using asreml for modeling.
  geVarCov <- gxeVarCov(TD = TDMaize, trait = "yld", engine = "asreml")
  ## Extract fitted values from the model.

  fitVarCov <- fitted(geVarCov)
  head(fitVarCov)
  }
```

---

gxeAmmi   *AMMI analysis*

Description

The Additive Main Effects and Multiplicative Interaction (AMMI) model fits a model which in-
volves the Additive Main effects (i.e. genotype and trial) along with Multiplicative Interaction
effects. The additive effects are the classical ANOVA main effects for genotype and environment,
the multiplicative effects follow from a principal component analysis on the interaction residuals (=
genotype by environment means after adjustment for additive genotype and environment effects).
This results in an interaction characterized by Interaction Principal Components (IPCA) enabling
simultaneous plotting of genotypes and trials.

The parameter nPC is used to indicate the number of principal components that is used in the princi-
pal component analysis (PCA). By setting this parameter to NULL the algorithm determines the best
number of principal components (see Details).

By specifying the parameter byYear = TRUE, a separate analysis will be done for every year in
the data. Combining the option with nPC = NULL may result in different numbers of principal com-
ponents per year. The AMMI estimates will still be returned as a single data.frame, but the other
results will be either lists or arrays.

Usage

```
gxeAmmi(
  TD,
  trials = names(TD),
  trait,
  nPC = 2,
  byYear = FALSE,
  center = TRUE,
  excludeGeno = NULL,
  useWt = FALSE
)
```

Arguments

TD	An object of class TD.
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
nPC	An integer specifying the number of principal components used as multiplicative term of genotype-by-trial interaction. If NULL, the number of principal components is determined by the algorithm using forward selection. See details.
byYear	Should the analysis be done by year? If TRUE the data is split by the variable year, analysis is performed and the results are merged together and returned.
center	Should the variables be shifted to be zero centered?
excludeGeno	An optional character vector with names of genotypes to be excluded from the analysis. If NULL, all genotypes are used.
useWt	Should weighting be used when modeling? Requires a column wt in TD.

Details

First a linear model $trait = genotype + trial + \epsilon$ is fitted with both genotype and trial fixed components in the model.

The residuals from the fitted model are then used in a PCA. If nPC is not NULL a single PCA is done using prcomp with maximum rank nPC.

In case nPC = NULL, the PCA is first done with one PC. Then using forward selection one by one the number of PCs is increased as long as the added component is significant in the analysis.

AMMI estimates are then computed using the results of the PCA.

Value

An object of class AMMI, a list containing:

envScores	A matrix with environmental scores.
genoScores	A matrix with genotypic scores.
importance	A data.frame containing the importance of the principal components.

anova	A data.frame containing anova scores of the AMMI analysis.
fitted	A data.frame containing fitted values from the AMMI model.
trait	A character string containing the analyzed trait.
envMean	A numerical vector containing the environmental means.
genoMean	A numerical vector containing the genotypic means.
overallMean	A numerical value containing the overall mean.

If byYear = TRUE, all returned items in the AMMI object except fitted will consist of a list of results by year.

References

Gauch H.G. (1992) Statistical Analysis of Regional Yield Trials: AMMI Analysis of Factorial Designs. Elsevier, Amsterdam.

Yan, W., Kang, M. (2002). GGE Biplot Analysis. Boca Raton: CRC Press.

See Also

Other AMMI: fitted.AMMI(), plot.AMMI(), report.AMMI(), residuals.AMMI()

Examples

```
## Run AMMI analysis on TDMaize.
geAmmi <- gxeAmmi(TD = TDMaize, trait = "yld")

## Summarize results.
summary(geAmmi)

## Create a biplot of genotypes and environment interaction with PC1 and PC2.
plot(geAmmi, plotType = "AMMI2")

## Create a pdf report summarizing the results.

report(geAmmi, outfile = tempfile(fileext = ".pdf"))
```

---

gxeFw   *Finlay-Wilkinson analysis*

Description

This function performs a Finlay-Wilkinson analysis of data classified by two factors.

Usage

```
gxeFw(
  TD,
  trials = names(TD),
  trait,
  maxIter = 15,
  tol = 0.001,
  sorted = c("descending", "ascending", "none"),
  genotypes = NULL,
  useWt = FALSE
)
```

Arguments

TD	An object of class TD.
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
maxIter	An integer specifying the maximum number of iterations in the algorithm.
tol	A positive numerical value specifying convergence tolerance of the algorithm.
sorted	A character string specifying the sorting order of the estimated values in the output.
genotypes	An optional character string containing the genotypes to which the analysis should be restricted. If NULL, all genotypes are used.
useWt	Should weighting be used when modeling? Requires a column wt in TD.

Value

An object of class FW, a list containing:

estimates	A data.frame containing the estimated values, with the following columns:
	• genotype: The name of the genotype.
	• sens: The estimate of the sensitivity.
	• se_sens: The standard error of the estimate of the sensitivity.
	• genMean: The estimate of the genotypic mean.
	• se_genMean: The standard error of the estimate of the genotypic mean.
	• MSdeviation: The mean square deviation about the line fitted to each genotype
	• rank: The rank of the genotype based on its sensitivity.
anova	A data.frame containing anova scores of the FW analysis.
envEffs	A data.frame containing the environmental effects, with the following columns:
	• trial: The name of the trial.
	• envEff: The estimate of the environment effect.
	• se_envEff: The standard error of the estimate of the environment effect.

- envMean: The estimate of the environment mean.
- rank: The rank of the trial based on its mean.

TD	The object of class TD on which the analysis was performed.
fittedGeno	A numerical vector containing the fitted values for the genotypes.
trait	A character string containing the analyzed trait.
nGeno	A numerical value containing the number of genotypes in the analysis.
nEnv	A numerical value containing the number of environments in the analysis.
tol	A numerical value containing the tolerance used during the analysis.
iter	A numerical value containing the number of iterations for the analysis to converge.

References

Finlay, K.W. & Wilkinson, G.N. (1963). The analysis of adaptation in a plant-breeding programme. Australian Journal of Agricultural Research, 14, 742-754.

See Also

Other Finlay-Wilkinson: `fitted.FW()`, `plot.FW()`, `report.FW()`, `residuals.FW()`

Examples

```
## Run Finlay-Wilkinson analysis on TDMaize.
geFW <- gxeFw(TDMaize, trait = "yld")

## Summarize results.
summary(geFW)

## Create a scatterplot of the results.
plot(geFW, plotType = "scatter")


## Create a report summarizing the results.
report(geFW, outfile = tempfile(fileext = ".pdf"))
```

---

gxeGGE   *GGE analysis*

Description

The Genotype plus Genotype by Environment interaction (GGE) model fits a model with trial as main fixed effect. Then a principal component analysis is done on the residuals. This results in an interaction characterized by Interaction Principal Components (IPCA) enabling simultaneous plotting of genotypes and trials.

The parameter nPC is used to indicate the number of principal components that is used in the principal component analysis (PCA). By setting this parameter to NULL the algorithm determines the best number of principal components (see Details).

By specifying the parameter byYear = TRUE, a separate analysis will be done for every year in the data. Combining the option with nPC = NULL may result in different numbers of principal components per year. The GGE estimates will still be returned as a single data.frame, but the other results will be either lists or arrays.

Usage

```
gxeGGE(
  TD,
  trials = names(TD),
  trait,
  nPC = 2,
  byYear = FALSE,
  center = TRUE,
  excludeGeno = NULL,
  useWt = FALSE
)
```

Arguments

TD	An object of class [TD]().
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
nPC	An integer specifying the number of principal components used as multiplicative term of genotype-by-trial interaction. If NULL, the number of principal components is determined by the algorithm using forward selection. See details.
byYear	Should the analysis be done by year? If TRUE the data is split by the variable year, analysis is performed and the results are merged together and returned.
center	Should the variables be shifted to be zero centered?
excludeGeno	An optional character vector with names of genotypes to be excluded from the analysis. If NULL, all genotypes are used.
useWt	Should weighting be used when modeling? Requires a column wt in TD.

Details

First a linear model $trait = trial + \epsilon$ is fitted with trial a fixed component in the model. The residuals from the fitted model are then used in a PCA. If nPC is not NULL a single PCA is done using `prcomp` with maximum rank nPC.

In case nPC = NULL, the PCA is first done with one PC. Then using forward selection one by one the number of PCs is increased as long as the added component is significant in the analysis.

GGE estimates are then computed using the results of the PCA.

---

gxeMegaEnv *Form mega environments based on fitted values from an AMMI model*

Description

This function fits an AMMI model and then using the fitted values produces a new factor clustering the trials. This factor is added as a column megaEnv to the input data. If a column megaEnv already exists this column is overwritten with a warning.

Mega environments are created by grouping environments based on their best performing genotype; i.e. environments that share the same best genotype belong to the same mega environment.

Usage

```
gxeMegaEnv(
  TD,
  trials = names(TD),
  trait,
  method = c("max", "min"),
  byYear = FALSE
)
```

Arguments

TD	An object of class TD.
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
method	A character string indicating the criterion to determine the best genotype per environment, either "max" or "min".
byYear	Should the analysis be done by year? If TRUE the data is split by the variable year, analysis is performed and the results are merged together and returned.

Value

An object of class megaEnv, a list consisting of

TD An object of class TD, the TD object used as input to the function with an extra column megaEnv.

summTab A data.frame, a summary table containing information on the trials in each mega environment.

trait The trait used for calculating the mega environments.

References

Atlin, G. N., R. J. Baker, K. B. McRae, and X. Lu. 2000. Selection Response in Subdivided Target Regions. Crop Sci. 40:7-13. doi:10.2135/cropsci2000.4017

See Also

Other mega environments: `plot.megaEnv()`, `predict.megaEnv()`

Examples

```
## Calculate mega environments for TDMaize.
gemegaEnv <- gxeMegaEnv(TD = TDMaize, trait = "yld")

## Calculate new mega environments based on the genotypes with the lowest
## value per environment.
gemegaEnv2 <- gxeMegaEnv(TD = TDMaize, trait = "yld", method = "min")
```

---

gxeStability   *Calculate stability coefficients for genotype-by-environment data*

Description

This function calculates different measures of stability, the cultivar-superiority measure of Lin & Binns (1988), Shukla's (1972) stability variance and Wricke's (1962) ecovalence.

Usage

```
gxeStability(
  TD,
  trials = names(TD),
  trait,
  method = c("superiority", "static", "wricke"),
  bestMethod = c("max", "min"),
  sorted = c("descending", "ascending", "none")
)
```

Arguments

TD	An object of class [TD](#).
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
method	A character vector specifying the measures of stability to be calculated. Options are "superiority" (cultivar-superiority measure), "static" (Shukla's stability variance) or "wricke" (wricke's ecovalence).
bestMethod	A character string specifying the criterion to define the best genotype. Either ″max″ or ″min″.
sorted	A character string specifying the sorting order of the results.

Value

An object of class [stability](#), a list containing:

superiority	A data.frame containing values for the cultivar-superiority measure of Lin and Binns.
static	A data.frame containing values for Shukla's stability variance.
wricke	A data.frame containing values for Wricke's ecovalence.
trait	A character string indicating the trait that has been analyzed.

References

Lin, C. S. and Binns, M. R. 1988. A superiority measure of cultivar performance for cultivar x location data. Can. J. Plant Sci. 68: 193-198

Shukla, G.K. 1972. Some statistical aspects of partitioning genotype-environmental components of variability. Heredity 29:237-245

Wricke, G. Uber eine method zur erfassung der okologischen streubreit in feldversuchen. Zeitschrift für Pflanzenzucht, v. 47, p. 92-96, 1962

See Also

Other stability: [plot.stability](#)(), [report.stability](#)()

Examples

```
## Compute three stability measures for TDMaize.
geStab <- gxeStability(TD = TDMaize, trait = ″yld″)

## Summarize results.
summary(geStab)

## Create plot of the computed stability measures against the means.
plot(geStab)
```

```
## Create a .pdf report summarizing the stability measures.
report(geStab, outfile = tempfile(fileext = ".pdf"))


## Compute Wricke's ecovalance for TDMaize with minimal values for yield as
## the best values. Sort results in ascending order.
geStab2 <- gxeStability(TD = TDMaize, trait = "yld", method = "wricke",
                        bestMethod = "min", sorted = "ascending")
summary(geStab2)
```

---

gxeVarComp   *Mixed model analysis of GxE table of means*

Description

This function fits a mixed model best fitting to the data in a TD object. The exact model fitted
is determined by both the structure of the genotype by environment table of observations and the
chosen parameters.

Six different types of models can be fitted depending on the structure of the environments in the
data. These models are described in the table below, together with the function parameters used in
gxeVarComp to fit the model.

Structure of environments	Model
Environments correspond to trials	**trait** = trial + **genotype + genotype:trial**
Trials form a factorial structure of locations x years	**trait** = year + location + year:location + **genotype + genotype:ye**
Trials are nested within year	**trait** = year + year:trial + **genotype + genotype:year + genotype**
Trials are nested within locations	**trait** = location + location:trial + **genotype + genotype:location**
Trials correspond to locations within regions across years	**trait** = region + region:location + year + region:year + region:loca
Trials are nested within scenarios	**trait** = scenario + scenario:trial + **genotype + genotype:scenario**

In the models above the random part of the model is printed bold.
For data in the form of GxE means, the last random term in all models above will become a residual
term. If the GxE means are provided together with weights, then a residual term will be added to
the models above.

The function first fits a model where all model terms are included as fixed terms. Based on the
ANOVA table of this model, terms in the fixed part of the model that are likely to give a problem
when fitting the mixed model are removed because of the reduced connectivity and number of avail-
able observations to estimate that model term. Also a warning is printed if the mean sum of squares
for a model term points to a possible zero variance component in the mixed model.

Then a model is fitted where all model terms are included as random terms. Based on the variance
components in this model the percentage of variance explained by each of the model components
is determined. The percentages of variance are printed in the model summary, together with the

variance components. The latter are presented on a standard deviation scale.

Finally a mixed model is fitted as specified in the overview above. Based on this mixed model variance components can be computed using [vc](), heritabilies can be computed using [herit]() and predictions can be made using [predict.varComp](). Predictions of genotypic performance can be made at the level of individual trials, or for groups of trials by using predictLevel.

Usage

```
gxeVarComp(
  TD,
  trials = names(TD),
  trait,
  engine = c("lme4", "asreml"),
  locationYear = FALSE,
  nestingFactor = NULL,
  regionLocationYear = FALSE,
  useWt = FALSE,
  diagnostics = FALSE
)
```

Arguments

TD	An object of class [TD]().
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
engine	A character string specifying the engine used for modeling. Either "lme4" or "asreml".
locationYear	Should a model be fitted assuming a factorial structure of locations x years?
nestingFactor	A character string specifying a column in TD specifying the nesting structure of the trials.
regionLocationYear	
	Should a model be fitted assuming locations within regions across years?
useWt	Should the model be fitted using weights? Doing so requires a column wt in the data. If useWt = FALSE, the default, and the data contains no replicates, the last model term will be dropped and used as homogeneous residual.
diagnostics	Should diagnostics on missing combinations of model variables be printed?

Value

An object of class varComp, a list containing:

fitMod	The fitted model.
modDat	A data.frame containing the data used when fitting the model.
nestingFactor	A name of the variable used as nesting variable in the model.

useLocYear	A boolean specifying if a model containing location x year interaction was fitted.
fullRandVC	A data.frame containing the variance components for the fully random model.
aovFullMixedMod	
	A data.frame containing the ANOVA table for the fully fixed model.
engine	The engine used for fitting the model.
diagTabs	A list of data.frame, one for each random model term, containing the missing combinations in the data for that term.

See Also

Other Mixed model analysis: CRDR(), correlations(), diagnostics(), herit(), plot.varComp(), predict.varComp(), vc()

Examples

```
## Fit a mixed model.
geVarComp <- gxeVarComp(TD = TDMaize, trait = "yld")

## Summarize results.
summary(geVarComp)

## Plot the standard deviations.
plot(geVarComp)

## Generate predictions
pred <- predict(geVarComp, predictLevel = "trial")
head(pred)
```

---

gxeVarCov   *Selects the best variance-covariance model for a set of trials*

Description

This function selects the best covariance structure for genetic correlations between trials. It fits a range of variance-covariance models (identity, compound symmetry (cs), diagonal, simple correlation with heterogeneous variance (outside), heterogeneous compound symmetry (hcs), first order factor analytic (fa), second order factor analytic (fa2) and unstructured), and selects the best one using a goodness-of-fit criterion. See details for the exact models fitted.

Usage

```
gxeVarCov(
  TD,
  trials = names(TD),
  trait,
 models = c("identity", "cs", "diagonal", "hcs", "outside", "fa", "fa2", "unstructured"),
```

```
    engine = c("lme4", "asreml"),
    criterion = c("BIC", "AIC"),
    ...
)
```

Arguments

TD	An object of class TD.
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
trait	A character string specifying the trait to be analyzed.
models	A character vector specifying the models to be fitted.
engine	A character string specifying the engine used for modeling. Either "lme4" or "asreml".
criterion	A string specifying a goodness-of-fit criterion. Either "AIC" or "BIC".
...	Further arguments to be passed to the modeling engine.

Details

The models fitted are of the form $y_{ij} = \mu_j + \epsilon_{ij}$, where y_{ij} is the phenotypic value of genotype i in environment j, μ_j is the environmental mean, and ϵ_{ij} represents mainly genetic variation, although some non-genetic variation may be included as well. The random term ϵ_{ij} is modeled in eight ways as described in the table below.

Model	Description	var(g_{ij})	cov(g_{ij};g_{ik})	Number of parameters
identity	identity	σ_G^2	0	1
cs	compound symmetry	$\sigma_G^2 + \sigma_{GE}^2$	σ_{GE}^2	2
diagonal	diagonal matrix (heteroscedastic)	$\sigma_{GE_j}^2$	0	J
hcs	heterogeneous compound symmetry	$\sigma_G^2 + \sigma_{GE_j}^2$	σ_G^2	$J+1$
outside	heterogeneity outside	$\sigma_{G_j}^2$	θ	$J+1$
fa	first order factor analytic	$\lambda_{1j}^2 + \sigma_{GE_j}^2$	$\lambda_{1j}\lambda_{1k}$	$2J$
fa2	second order factor analytic	$\lambda_{1j}^2 + \lambda_{2j}^2 + \sigma_{GE_j}^2$	$\lambda_{1j}\lambda_{1k} + \lambda_{2j}\lambda_{2k}$	$3J-1$
unstructured	unstructured	$\sigma_{G_j}^2$	$\sigma_{G_{j,k}}^2$	$J(J+1)/2$

In this table J is the number of environments, σ_G^2 the variance component for the genotype main effects, σ_{GE}^2 the variance component for GxE interactions. $\sigma_{G_j}^2$ and $\sigma_{GE_j}^2$ are the environment specific variance components for the genotype main effects and GxE interaction in environment j. $\sigma_{G_{j,k}}^2$ is the genetic covariance between environments j and k. θ is the common correlation between environments and λ_{1j} and λ_{2j} are environment specific multiplicative parameters.

Value

An object of class varCov, a list object containing:

STA	An object of class STA containing the best fitted model.
choice	A character string indicating the best fitted model.

summary	A data.frame with a summary of the fitted models.
vcov	The covariance matrix of the best fitted model.
criterion	A character string indicating the goodness-of-fit criterion used for determining the best model, either "AIC" or "BIC".
engine	A character string containing the engine used for the analysis.
trait	A character string containing the trait analyzed.
dat	A data.frame with the full data set used for the analysis.

Note

If engine = "lme4", only the compound symmetry model can be fitted.

See Also

Other varCov: fitted.varCov(), plot.varCov(), report.varCov(), residuals.varCov()

Examples

```
## Select the best variance-covariance model using lme4 for modeling.
geVarCov <- gxeVarCov(TD = TDMaize, trait = "yld")

## Summarize results.
summary(geVarCov)


## Create a pdf report summarizing the results.
report(geVarCov, outfile = tempfile(fileext = ".pdf"))



if (requireNamespace("asreml", quietly = TRUE)) {
  ## Select the best variance-covariance model using asreml for modeling.
  ## Use BIC as a goodness-of-fit criterion.
  geVarCov2 <- gxeVarCov(TD = TDMaize, trait = "yld", engine = "asreml",
                         criterion = "BIC")

  summary(geVarCov2)

  ## Plot a heatmap of the correlation matrix for the best model.
  plot(geVarCov2)
  }
```

---

herit	*Calculate heritability*

Description

Calculate the heritability based on the fitted model. For balanced data, the heritability is calculated as described by Atlin et al. E.g. for a model with trials nested within locations, which has a random part that looks like this: genotype + genotype:location + genotype:location:trial the heritability is computed as

$$\sigma_G^2/(\sigma_G^2 + \sigma_L^2/l + \sigma_{LT}^2/lt + \sigma_E^2/ltr)$$

In this formula the σ terms stand for the standard deviations of the respective model terms, and the lower case letters for the number of levels for the respective model terms. So σ_L is the standard deviation for the location term in the model and l is the number of locations. σ_E corresponds to the residual standard deviation and r to the number of replicates.

When the data is unbalanced a more general form of this formula is used as described in Holland et al. Here the numerator l is replaced by the harmonic means of the number of locations across genotypes. The other numerators are replaced correspondingly. For balanced data this more general form gives identical results as the form described by Atlin et al.

Usage

```
herit(varComp)
```

Arguments

varComp   An object of class varComp.

References

Atlin, G. N., Baker, R. J., McRae, K. B., & Lu, X. (2000). Selection response in subdivided target regions. Crop Science, 40(1), 7–13. doi:10.2135/cropsci2000.4017

Holland, J.B., W.E. Nyquist, and C.T. Cervantes-Martínez. (2003). Estimating and interpreting heritability for plant breeding: An update. Plant Breed. Rev. 2003:9–112. doi:10.1002/9780470650202.ch2

See Also

Other Mixed model analysis: `CRDR()`, `correlations()`, `diagnostics()`, `gxeVarComp()`, `plot.varComp()`, `predict.varComp()`, `vc()`

Examples

```
## Fit a mixed model.
geVarComp <- gxeVarComp(TD = TDMaize, trait = "yld")

## Compute heritability.
herit(geVarComp)
```

---

multMissing   *Estimate missing values in multivariate data*

Description

This function estimates missing values for units in a multivariate dataset, using an iterative regression technique.

Usage

```
multMissing(Y, maxIter = 10, naStrings = NULL)
```

Arguments

Y	A matrix, data.frame or vector of multivariate data.
maxIter	An integer specifying the maximum number of iterations.
naStrings	A character vector of strings which are to be interpreted as NA values.

Details

Initial estimates of the missing values in each variate are formed from the variate means using the values for units that have no missing values for any variate. Estimates of the missing values for each variate are then recalculated as the fitted values from the multiple regression of that variate on all the other variates. When all the missing values have been estimated the variate means are recalculated. If any of the means differs from the previous mean by more than a tolerance (the initial standard error divided by 1000) the process is repeated, subject to a maximum number of repetitions defined by maxIter option. The default maximum number of iterations (10) is usually sufficient when there are few missing values, say two or three. If there are many more, 20 or so, it may be necessary to increase the maximum number of iterations to around 30. The method is similar to that of Orchard & Woodbury (1972), but does not adjust for bias in the variance-covariance matrix as suggested by Beale & Little (1975).

Value

An object of the same class as the input Y with the missing values replaced by their estimates.

References

Beale, E.M.L. & Little, R.J.A. (1975). Missing values in multivariate analysis. Journal of the Royal Statistical Society, Series B, 37, 129-145.

Orchard, T. & Woodbury, M.A. (1972). A missing information principle: theory and applications. In: Proceedings of the 6th Berkeley Symposium in Mathematical Statistics and Probability, Vol I, 697-715.

Examples

```
M <- matrix(c("1", "2", "3", NA, "b", "5", "6",
              "6", "5", "b", NA, "3", "2", "1"), nrow = 7, ncol = 2)

## Estimate missing values treating "b" as NA.
multMissing(M, naStrings = "b")
```

---

plot.AMMI   *Plot function for class AMMI*

Description

Two types of biplot can be made. A plot of genotype and environment means vs PC1 (AMMI1) or a biplot of genotypes and environment interaction with PC1 and PC2 (AMMI2).

If the AMMI analysis was done by year, a separate plot will be made for every year in the data. For some years the number of principal components may be lower than the number specified on the secondary axis. If this is the case this year is skipped when plotting. If this happens for all years the function returns an error.

Usage

```
## S3 method for class 'AMMI'
plot(
  x,
  ...,
  plotType = c("AMMI1", "AMMI2", "GGE2"),
  scale = 1,
  plotGeno = TRUE,
  colorGenoBy = NULL,
  colGeno = NULL,
  sizeGeno = 0,
  plotConvHull = FALSE,
  plotEnv = TRUE,
  colorEnvBy = NULL,
  colEnv = NULL,
  sizeEnv = 3,
  envFactor = 1,
```

```
    primAxis = "PC1",
    secAxis = "PC2",
    rotatePC = NULL,
    title = NULL,
    output = TRUE
)
```

Arguments

x	An object of class AMMI
...	Not used.
plotType	A character string indicating which plot should be made. Either "AMMI1" for an AMMI1 plot (genotype and environment means vs PC1) or "AMMI2" for an AMMI2 biplot (genotypes and environment interaction with PC1 and PC2) respectively. For results of a GGE analysis only an GGE2 biplot can be made and plotType may be ignored.
scale	A numerical value. The variables are scaled by lambda ^ scale and the observations by lambda ^ (1 - scale) where lambda are the singular values computed by [princomp](#) in [gxeAmmi](#). Normally 0 <= scale <= 1, and a warning will be issued if the specified scale is outside this range.
plotGeno	Should genotypes be plotted?
colorGenoBy	A character string indicating a column in the TD used as input for the AMMI analysis by which the genotypes should be colored. If NULL all genotypes will be colored in black.
colGeno	A character vector with plot colors for the genotypes. A single color when colorGenoBy = NULL, a vector of colors otherwise.
sizeGeno	An numerical value indicating the text size for plotting the genotypes. Use sizeGeno = 0 for plotting genotypes as points instead of using their names.
plotConvHull	Should a convex hull be plotted around the genotypes. If TRUE a convex hull is plotted. For GGE2 biplots lines from the origin of the plot perpendicular to the edges of the hull are added. Only valid for AMMI2 and GGE2 biplots.
plotEnv	Should environments be plotted?
colorEnvBy	A character string indicating a column in the TD used as input for the AMMI analysis by which the environments should be colored. If NULL all genotypes will be colored in red.
colEnv	A character string with plot colors for the environments. A single color when colorEnvBy = NULL, a vector of colors otherwise.
sizeEnv	An integer indicating the text size for plotting the environments.
envFactor	A positive numerical value giving a factor by which to blow up the environmental scores. Providing a value between 0 and 1 will effectively blow up the genotypic scores.
primAxis	A character string indicating the principal component to be plotted on the primary axis of the AMMI2 plot. Has to be given as "PCn" where n is the number of the principal component.

secAxis	A character string indicating the principal component to be plotted on the secondary axis of the AMMI2 plot. Has to be given as "PCn" where n is the number of the principal component. n Has to differ from primAxis.
rotatePC	A character string indicating a genotype or environment that is to be aligned with the positive x-axis in the plot.
title	A character string used a title for the plot.
output	Should the plot be output to the current device? If FALSE only a list of ggplot objects is invisibly returned.

Value

A biplot depending on plotType. The ggplot object for the biplot is returned invisibly.

See Also

Other AMMI: fitted.AMMI(), gxeAmmi(), report.AMMI(), residuals.AMMI()

Examples

```
## Run AMMI analysis.
geAmmi <- gxeAmmi(TD = TDMaize, trait = "yld")

## Create an AMMI1 biplot.
plot(geAmmi)

## Create an AMMI2 biplot.
plot(geAmmi, plotType = "AMMI2", scale = 0.5)

## Create an AMMI2 biplot, with HN96b along the positive x-axis.
plot(geAmmi, plotType = "AMMI2", scale = 0.5, rotatePC = "HN96b")

## Run GGE analysis.
geGGE <- gxeGGE(TD = TDMaize, trait = "yld")

## Create an GGE2 biplot.
## Add a convex hull.
plot(geGGE, plotType = "GGE2", scale = 0.5, plotConvHull = TRUE)
```

---

plot.FW   *Plot function for class FW*

Description

Four types of plot can be made. A scatter plot for genotypic mean, square root of mean squared deviation and sensitivity, a line plot with fitted lines for each genotype, a trellis plot with individual slopes per genotype and a scatter plot of fitted values in the worst and best trial.
It is possible to select genotypes for the trellis plot using the genotypes parameter. If there are more than 64 genotypes, only the first 64 are plotted in the trellis plot.

Usage

```
## S3 method for class 'FW'
plot(
  x,
  ...,
  plotType = c("scatter", "line", "trellis", "scatterFit"),
  order = c("ascending", "descending"),
  response = c("predicted", "observed"),
  colorGenoBy = NULL,
  colGeno = NULL,
  genotypes = NULL,
  title = paste("Finlay & Wilkinson analysis for", x$trait),
  output = TRUE
)
```

Arguments

x	An object of class FW.
...	Not used.
plotType	A character string indicating which plot should be made. Either "scatter", "line" or "trellis" for creating a scatter plot of genotypic means, mse and sensitivities, a plot of fitted lines for each genotype or a trellis plot of the individual genotype slopes respectively.
order	A character string specifying whether the results in the line plot should be ordered in an increasing (or decreasing) order of sensitivities. Ignored if plotType is not "line".
response	A character string specifying whether in the line plot the "predicted" or the "observed" data should be plotted. Ignored if plotType is not "line".
colorGenoBy	A character string indicating a column in the TD used as input for the Finlay Wilkinson analysis by which the genotypes should be colored. If NULL all genotypes will be colored differently.
colGeno	A character vector with plot colors for the genotypes. A single color when colorGenoBy = NULL, a vector of colors otherwise.
genotypes	An optional character string containing the genotypes to be plotted in the trellis plot. If NULL all genotypes are plotted. If more than 64 genotypes are selected, only the first 64 are plotted.
title	A character string used a title for the plot.
output	Should the plot be output to the current device? If FALSE, only a list of ggplot objects is invisibly returned.

Value

A plot depending on plotType.

See Also

Other Finlay-Wilkinson: fitted.FW(), gxeFw(), report.FW(), residuals.FW()

Examples

```
## Run Finlay-Wilkinson analysis.
geFW <- gxeFw(TD = TDMaize, trait = "yld")

## Create a scatter plot.
plot(geFW)

## Create a line plot.
plot(geFW, plotType = "line")

## Create a line plot showing observed data value for genotypes and fitted lines.
## Display trials in descending order.
plot(geFW, plotType = "line", order = "descending", response = "observed")


## Create a trellis plot.
plot(geFW, plotType = "trellis")

## Create a scatter plot of fitted values for the worst and best trials.
plot(geFW, plotType = "scatterFit")
```

---

plot.megaEnv *Plot function for class megaEnv*

Description

Function for creating scatter plots of predicted values in computed mega environments.

Usage

```
## S3 method for class 'megaEnv'
plot(
  x,
  ...,
  engine = c("lme4", "asreml"),
  colorGenoBy = NULL,
  title = paste("Scatterplot of mega environments for", x$trait),
  output = TRUE
)
```

Arguments

x	An object of class megaEnv.
...	Further arguments to be passed on to underlying plot functions.
engine	A character string specifying the engine used for making the predictions on which the plots are based.

colorGenoBy	A character string indicating a column in `TD` by which the genotypes in the scatter plots are colored. If `NULL` all genotypes are displayed in black.
title	A character string used a title for the plot.
output	Should the plot be output to the current device? If `FALSE` only a ggtable object is invisibly returned.

See Also

Other mega environments: [gxeMegaEnv](#)(), [predict.megaEnv](#)()

Examples

```
## Compute mega environments for TDMaize.
geMegaEnv <- gxeMegaEnv(TD = TDMaize, trait = "yld")

## Create a scatter plot of predicted values.
plot(geMegaEnv)
```

---

plot.stability   *Plot function for class stability*

Description

Function for creating scatter plots of the square roots of the computed stability measures against the means.

Usage

```
## S3 method for class 'stability'
plot(
  x,
  ...,
  colorGenoBy = NULL,
  colGeno = NULL,
  title = paste("Stability coefficients for", x$trait),
  output = TRUE
)
```

Arguments

x	An object of class stability.
...	Not used.
colorGenoBy	A character string indicating a column in the `TD` used as input for the stability analysis by which the genotypes should be colored. If `NULL` all genotypes will be colored black.

colGeno	A character vector with plot colors for the genotypes. A single color when `colorGenoBy = NULL`, a vector of colors otherwise.
title	A character string used a title for the plot.
output	Should the plot be output to the current device? If `FALSE` only a list of ggplot objects is invisibly returned.

Value

A list of ggplot object is invisibly returned.

See Also

Other stability: [gxeStability](), [report.stability]()

Examples

```
## Compute three stability measures for TDMaize.
geStab <- gxeStability(TD = TDMaize, trait = "yld")

## Create scatter plots of the computed stability measures against the means.
plot(geStab)
```

---

plot.varComp	*Plot function for class varComp*

Description

A plot is created of either the standard deviations of each of the terms in the fitted model or the percentage of variance explained by each of the terms in the fitted model. Also the degrees of freedom for each of the terms is shown in the plot.

Usage

```
## S3 method for class 'varComp'
plot(x, ..., plotType = c("sd", "percVar"), title = NULL, output = TRUE)
```

Arguments

x	An object of class varComp
...	Not used.
plotType	A character string. Either "sd" to plot the standard deviation of the variance components, or "percVar" to plot the percentage of variance explained by each variance component.
title	A character string used a title for the plot.
output	Should the plot be output to the current device? If `FALSE` only a ggplot object is invisibly returned.

Value

A ggplot object is invisibly returned.

See Also

Other Mixed model analysis: CRDR(), correlations(), diagnostics(), gxeVarComp(), herit(),
predict.varComp(), vc()

Examples

```
## Fit a mixed model.
geVarComp <- gxeVarComp(TD = TDMaize, trait = "yld")

## Plot the standard deviations.
plot(geVarComp)
## Plot the percentage of variance explained.
plot(geVarComp, plotType = "percVar")
```

---

plot.varCov   *Plot function for class varCov*

Description

Function for plotting a heatmap of the correlation matrix for objects of class varCov.

Usage

```
## S3 method for class 'varCov'
plot(x, title = paste("Heatmap for model:", x$choice), ..., output = TRUE)
```

Arguments

x	An object of class varCov
title	A character string used a title for the plot.
...	Not used.
output	Should the plot be output to the current device? If FALSE only a ggplot object is invisibly returned.

See Also

Other varCov: fitted.varCov(), gxeVarCov(), report.varCov(), residuals.varCov()

Examples

```
if (requireNamespace("asreml", quietly = TRUE)) {
  ## Select the best variance-covariance model using asreml for modeling.
  geVarCov <- gxeVarCov(TD = TDMaize, trait = "yld", engine = "asreml")

  ## Create a heatmap of the correlation matrix for the best model.
  plot(geVarCov)
  }
```

---

predict.megaEnv   *Compute BLUPS based on a set of mega environments*

Description

This function calculates Best Linear Unbiased Predictors (BLUPS) and associated standard errors based on a set of mega environments.

Usage

```
## S3 method for class 'megaEnv'
predict(
  object,
  ...,
  trials = names(object$TD),
  useYear = FALSE,
  engine = c("lme4", "asreml")
)
```

Arguments

object	An object of class megaEnv.
...	Further parameters passed to either asreml or lmer.
trials	A character string specifying the trials to be analyzed. If not supplied, all trials are used in the analysis.
useYear	Should year be used for modeling (as years within trials). If TRUE, TD should contain a column "year".
engine	A character string specifying the engine used for modeling.

Value

A list consisting of two data.frames, predictedValue containing BLUPs per genotype per mega environment and standardError containing standard errors for those BLUPs.

See Also

Other mega environments: [gxeMegaEnv](), [plot.megaEnv]()

Examples

```
## Compute mega environments for TDMaize.
geMegaEnv <- gxeMegaEnv(TD = TDMaize, trait = "yld")

## Compute BLUPS and standard errors for those mega environments.
megaEnvPred <- predict(geMegaEnv)
head(megaEnvPred$predictedValue)
head(megaEnvPred$standardError)
```

---

predict.varComp   *Predictions based on a fitted varComp model.*

Description

Predictions are made based on the fitted model in the varComp object. These predictions can be at genotype level, at genotype x trial level or at the level of genotype x nestingFactor. If the model was fitted with trial as year x location then genotype x trial level becomes genotype x year x location.

Usage

```
## S3 method for class 'varComp'
predict(object, ..., predictLevel = "genotype")
```

Arguments

object   An object of class varComp.

...   Not used.

predictLevel A character string, the level at which prediction should be made. Either "geno-
              type" for prediction at genotype level, "trial" for predictions at genotype x trial
              level, the variable used as nesting factor for predictions at the level of genotype
              x nestingFactor level, or one or more of the extra terms used in the model. E.g.
              c("region", "year") for a model fitted with regionLocationYear = TRUE.

Value

A data.frame with predictions.

See Also

Other Mixed model analysis: [CRDR](), [correlations](), [diagnostics](), [gxeVarComp](), [herit](), [plot.varComp](), [vc]()

Examples

```
## Fit a mixed model.
geVarComp <- gxeVarComp(TD = TDMaize, trait = "yld")

## Predictions at genotype level.
predGeno <- predict(geVarComp)
head(predGeno)

## Predictions at genotype x trial level.
predGenoTrial <- predict(geVarComp, predictLevel = "trial")
head(predGenoTrial)
```

---

report.AMMI *Report method for class AMMI*

Description

A pdf report will be created containing a summary of an AMMI object. Simultaneously the same report will be created as a tex file.

Usage

```
## S3 method for class 'AMMI'
report(x, ..., outfile = NULL)
```

Arguments

x	An object of class AMMI.
...	Not used.
outfile	A character string, the name and location of the output .pdf and .tex file for the report. If NULL, a report with a default name will be created in the current working directory.

Value

A pdf and tex report.

See Also

Other AMMI: `fitted.AMMI()`, `gxeAmmi()`, `plot.AMMI()`, `residuals.AMMI()`

Examples

```
## Run AMMI analysis on TDMaize.
geAmmi <- gxeAmmi(TD = TDMaize, trait = "yld")

## Create a pdf report summarizing the results.
report(geAmmi, outfile = tempfile(fileext = ".pdf"))
```

---

report.FW   *Report method for class FW*

Description

A pdf report will be created containing a summary of an FW object. Simultaneously the same report will be created as a tex file.

Usage

```
## S3 method for class 'FW'
report(x, sortBy = c("sens", "genMean", "mse"), ..., outfile = NULL)
```

Arguments

x	An object of class FW.
sortBy	A character string indicating by which variable the estimates should be sorted. Either sens(itivity), genMean (genotypic Mean) or mse (mean squared error).
...	Not used.
outfile	A character string, the name and location of the output .pdf and .tex file for the report. If NULL, a report with a default name will be created in the current working directory.

Value

A pdf and tex report.

See Also

Other Finlay-Wilkinson: fitted.FW(), gxeFw(), plot.FW(), residuals.FW()

Examples

```
## Run Finlay-Wilkinson analysis on TDMaize.
geFW <- gxeFw(TDMaize, trait = "yld")


## Create a report summarizing the results.
report(geFW, outfile = tempfile(fileext = ".pdf"))
```

---

report.stability   *Report method for class stability*

Description

A pdf report will be created containing a summary of an object of class stability. Simultaneously the same report will be created as a tex file.

Usage

```
## S3 method for class 'stability'
report(x, ..., outfile = NULL)
```

Arguments

x	An object of class stability.
...	Not used.
outfile	A character string, the name and location of the output .pdf and .tex file for the report. If NULL, a report with a default name will be created in the current working directory.

Value

A pdf and tex report.

See Also

Other stability: gxeStability(), plot.stability()

Examples

```
## Compute three stability measures for TDMaize.
geStab <- gxeStability(TD = TDMaize, trait = "yld")

## Create a .pdf report summarizing the stability measures.
report(geStab, outfile = tempfile(fileext = ".pdf"))
```

---

report.varCov   *Report method for class varCov*

Description

A pdf report will be created containing a summary of an object of class varCov. Simultaneously the
same report will be created as a tex file.

Usage

```
## S3 method for class 'varCov'
report(x, ..., outfile = NULL)
```

Arguments

x   An object of class varCov.

...   Not used.

outfile   A character string, the name and location of the output .pdf and .tex file for
                the report. If NULL, a report with a default name will be created in the current
                working directory.

Value

A pdf and tex report.

See Also

Other varCov: `fitted.varCov()`, `gxeVarCov()`, `plot.varCov()`, `residuals.varCov()`

Examples

```
## Select the best variance-covariance model using asreml for modeling.
if (requireNamespace("asreml", quietly = TRUE)) {
  geVarCov <- gxeVarCov(TD = TDMaize, trait = "yld", engine = "asreml")

  ## Create a pdf report summarizing the results.
  report(geVarCov, outfile = tempfile(fileext = ".pdf"))
  }
```

residuals.AMMI *Extract residuals.*

Description

Extract the residuals for the fitted AMMI model.

Usage

```
## S3 method for class 'AMMI'
residuals(object, ...)
```

Arguments

object   An object of class AMMI

...   Not used.

Value

A data.frame with residuals.

See Also

Other AMMI: `fitted.AMMI()`, `gxeAmmi()`, `plot.AMMI()`, `report.AMMI()`

Examples

```
## Run AMMI analysis on TDMaize.
geAmmi <- gxeAmmi(TD = TDMaize, trait = ”yld”)

## Extract residuals.
residAmmi <- residuals(geAmmi)
head(residAmmi)
```

residuals.FW *Extract residuals.*

Description

Extract the residuals for a fitted Finlay-Wilkinson model.

Usage

```
## S3 method for class 'FW'
residuals(object, ...)
```

Arguments

object   An object of class FW

...   Not used.

Value

A data.frame with residuals.

See Also

Other Finlay-Wilkinson: `fitted.FW()`, `gxeFw()`, `plot.FW()`, `report.FW()`

Examples

```
## Run Finlay-Wilkinson analysis.
geFW <- gxeFw(TD = TDMaize, trait = "yld")

## Extract residuals.
residFW <- residuals(geFW)
head(residFW)
```

---

residuals.varCov   *Extract residuals.*

Description

Extract the residuals for the best model.

Usage

```
## S3 method for class 'varCov'
residuals(object, ...)
```

Arguments

object   An object of class varCov

...   Not used.

Value

A data.frame with residuals.

See Also

Other varCov: `fitted.varCov()`, `gxeVarCov()`, `plot.varCov()`, `report.varCov()`

Examples

```
## Select the best variance-covariance model using asreml for modeling.
if (requireNamespace("asreml", quietly = TRUE)) {
  geVarCov <- gxeVarCov(TD = TDMaize, trait = "yld", engine = "asreml")

  ## Extract residuals from the model.
  residVarCov <- residuals(geVarCov)
  head(residVarCov)
  }
```

TDMaize   *Field data for a maize experiment in Tlaltizapan, Mexico*

Description

A dataset converted into a TD object containing data corresponding to an F2 maize reference population from CIMMYT maize drought breeding program, which was derived from the cross of a drought-tolerant line (P1) with a drought susceptible line (P2) as described in detail by Ribaut et al. (1996, 1997).

DNA from 211 F2 plants was extracted to produce information for 132 co-dominant markers on 10 linkage groups. Phenotypic evaluations were performed on 211 F2:3 families, each one derived from an original F2 plant. The families were evaluated under different water and nitrogen regimes during 1992, 1994 and 1996. In the winter of 1992 three water regimes were imposed on the trials: well watered (NS), intermediate stress (IS) and severe stress (SS). In the winter of 1994, only the IS and SS trials were available. Nitrogen availability varied in the 1996 trials, with two low nitrogen treatments (LN, in winter and summer) and one high-nitrogen treatment (HN in summer). In each of the trials, five traits were evaluated but only grain yield is included in the data.

Usage

```
TDMaize
```

Format

A TD object, a list containing 8 data.frames, each with the following columns:

trial trial, a combination of water regime, year and nitrogen treatment

genotype genotype

yld grain yield in tons

Source

<https://link.springer.com/article/10.1007/BF00221905>

References

Ribaut JM, Hoisington DA, Deutsch JA, Jiang C, Gonzalez de Leon D (1996) Identification of quantitative trait loci under drought conditions in tropical maize.1. Flowering parameters and the anthesis-silking interval. Theor Appl Genet 92:905–914

Ribaut JM, Jiang C, Gonzalez de Leon D, Edmeades GO, Hoisington DA (1997) Identification of quantitative trait loci under drought conditions in tropical maize.2. Yield components and marker-assisted selection strategies. Theor Appl Genet 94:887–896

---

vc	*Extract variance components*

Description

Extract variance components from an object of class varComp.

Usage

```
vc(varComp)
```

Arguments

varComp   An object of class varComp.

Value

A data.frame with variance components and standard errors for the random components in the fitted model.

See Also

Other Mixed model analysis: CRDR(), correlations(), diagnostics(), gxeVarComp(), herit(), plot.varComp(), predict.varComp()

Examples

```
## Fit a mixed model.
geVarComp <- gxeVarComp(TD = TDMaize, trait = "yld")

## Extract variance components.
vc(geVarComp)
```

Index