Package STAR

Technical Documents

Biometrics and Breeding Informatics
Plant Breeding, Genetics and Biotechnology Division
INTERNATIONAL RICE RESEARCH INSTITUTE

Package “STAR”

December 2012

Type Package

Title Statistical Tools for Agricultural Research

Depends R (>= 2.15.1), cluster, lme4, multcomp

Description A package for data management and basic statistical analysis of experimental d

Version

Date

Author

Maintainer

License

R topics documented:

STAR package *Statistical Tool for Agricultural Research*

Description

This package is designed for data management and basic statistical analysis.

The package provides modules for generating randomization and layout of experimental designs commonly used in crop research, data management with a spreadsheet and basic statistical analysis including descriptive statistics, balanced analysis of variance, single site analysis, multi-environment analysis, correlation analysis, regression analysis and non-parametric data analysis.

Details

Package	STAR
Version	0.1
Date	December 2012
Depends	R (>= 2.11.1), tcltk
Suggests	
LazyLoad	
License	

Requirements

This package requires the tcltk package and tkrplot package.

Future Direction

Modules for general linear models, mixed models, combined analysis, multivariate analysis and G x E analysis will also be included in succeeding versions.

License

The release of this package is licensed under GPL version 2 or newer.

Warning

Classes and methods are considered either to be in its beta or alpha (pre-beta) stage.

Author(s)

Alaine A. Gulles <a.gulles@irri.org>, Violeta I. Bartolome, Leilani A. Nora, Christoffer Edd N. Relente and Guoyou Ye of the International Rice Research Institute

Reference(s)

ad.test *Anderson-Darling test for normality*

Description

Performs the Anderson-Darling test for the composite hypothesis of normality, see e.g. Thode (2002, Sec. 5.1.4).

Usage

```
ad.test(x)
```

Arguments

`x`	a numeric vector of data values, the number of which must be greater than 7. Missing values are allowed.

Details

The Anderson-Darling test is an EDF omnibus test for the composite hypothesis of normality. The test statistic is

$$A = -n - \frac{1}{n}\sum_{i=1}^{n}[2i-1][\ln(p_{(i)}) + \ln(1-p_{(n-i+1)})],$$

where $p_{(i)} = \Phi([x_{(i)} - \overline{x}]/s)$. Here, Φ is the cumulative distribution function of the standard normal distribution, and $\overline{x}$ and s are mean and standard deviation of the data values. The p-value is computed from the modified statistic $Z = A(1.0 + 0.75/n + 2.25/n^2)\backslash$ according to Table 4.9 in Stephens (1986).

Value

A list with class "htest" containing the following components:

`statistic`	the value of the Anderson-Darling statistic.
`p.value`	the p-value for the test.
`method`	the character string "Anderson-Darling normality test".
`data.name`	a character string giving the name(s) of the data.

Note

The Anderson-Darling test is the recommended EDF test by Stephens (1986). Compared to the Cramer-von Mises test (as second choice) it gives more weight to the tails of the distribution.

Author(s)

Juergen Gross from the package "nortest"

Reference(s)

Stephens, M.A. (1986): Tests based on EDF statistics. In: D’Agostino, R.B. and Stephens, M.A., eds.: Goodness-of-Fit Techniques. Marcel Dekker, New York.

Thode Jr., H.C. (2002): Testing for Normality. Marcel Dekker, New York.

See Also

shapiro.test for performing the Shapiro-Wilk test for normality. cvm.test, lillie.test, sf.test for performing further tests for normality. qqnorm for producing a normal quantile-quantile plot.

Examples

ad.test(rnorm(100, mean = 5, sd = 3))

ad.test(runif(100, min = 2, max = 4))

AggregateData *Compute Summary Statistics of Data Subsets*

Description

This function computes summary statistics for each subsets or group and creates a new aggregated file or creates a new variable in the existing data file that contains the aggregated data.

Usage

```
AggregateData(data, var, grp, stat = c("mean"), append = FALSE)
```

Arguments

`data` a non-null character string which indicate the name of the data frame

`var` a non-empty character vector which indicates the variables to aggregate

`grp` a non-empty character vector which indicates the grouping factor

`stat` a character string or vector which indicates the statistics that will be computed for each of the variables indicated in `var`

`append` logical: should the aggregated data be appended to the original data set?

Details

Observations are grouped together based on the values of the grouping factor (`grp`). Each unique combination of grouping factor values defines a group.

Variables to aggregate (`var`) are used with the statistics to create new aggregate variables. The available statistics for this function are: minimum, maximum, mean, median, sum, variance and standard deviation. The default statistics that will be computed is the mean.

If `append` is `FALSE` then a new data frame is created which contains only the aggregated data. If `append` is `TRUE` then new columns which contains the aggregated data will be added to the existing data.

Value

A data frame which contains the aggregated data only or the original data frame with the aggregated data.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

`aggregate`

Examples

## load the dataset "`Gerua`" from the STAR package

```
data(Gerua)
```

## create a new data frame which computes the default statistics ("mean") of variable "`GRNYLD`" in the data frame "`Gerua`", grouped according to "`TPL`"

```
AggregateData(data = 'Gerua', var = 'GRNYLD', grp = 'TPL')
```

## create a new data frame which contains the mean and the standard deviation of the different variables in "`Gerua`", grouped according to "`TPL`" and "`CULT`" and save the result in the data frame named "`aggrData1`"

```
aggrData1 <- AggregateData(data = 'Gerua', var = c('DFF', 'PLHT',
             'PNCLE', 'GRNYLD'), grp = c('TPL', 'CULT'), stat =
             c('mean', 'standard deviation'))
```

## computes the mean and standard deviation of the different variables in "`Gerua`", grouped according to "`TPL`" and "`CULT`" and the result is appended to the original data frame and save in the data frame named "`aggrData2`"

```
aggrData2 <- AggregateData(data = Gerua, var = c('DFF', 'PLHT',
             'PNCLE', 'GRNYLD'), grp = c('TPL', 'CULT'), stat =
             c('mean', 'standard deviation'), append = TRUE)
```

AOVTest *Performs Analysis of Variance*

Description

Performs analysis of variance with factors treated as fixed.

Usage

```
AOVTest(data, design, respvar, factor1, factor2 = NULL, factor3 =
        NULL, factor4 = NULL, rep1 = NULL, rep2 = NULL, set =
        NULL, descriptive = FALSE, normality = FALSE, homogeneity
        = FALSE, pwTest = NULL, pwVar = NULL, contrastOption =
        NULL, sig = 0.05, outputPath = NULL)
```

Arguments

`data` a non-null character string which indicates the name of the data frame containing the variables specified in `respvar`, `factor1`, `factor2`, `factor3`, `factor4`, `rep1` and `rep2`.

`design` a character string indicating the experimental design to be used and must be one of the following: “`CRD`” (Completely Randomized Design) , “`RCBD`” (Randomized Complete Block Design) , "`LSD`" (Latin Square Design), "`SplitCRD`" (Split Plot in CRD), "`SplitRCBD`" (Split Plot in RCBD), "`SplitLSD`" (Split Plot in LSD), "`Strip`" (Strip Plot Design), "`Split2CRD`" (Split-Split Plot in CRD), "`Split2RCBD`" (Split-Split Plot in RCBD), "`Split2LSD`" (Split-Split Plot in LSD), "`Strip-Split`" (Strip-Split Plot Design), "`Split3CRD`" (Split-Split-Split Plot in CRD) , "`Split3RCBD`" (Split-Split-Split Plot in RCBD), "`Split3LSD`" (Split-Split-Split Plot in CRD), or "`Strip-Split2`" (Strip-Split-Split Plot Design).

`respvar` a non-empty character vector which indicates the response variable(s) to be used in the analysis

`factor1` a non-empty character vector which indicates the factor(s) to be used for CRD, RCBD and LSD or the main plot factor(s) for Split Family of Design, or the vertical factor(s) for the Strip Family of Design

`factor2` `NULL` for CRD, RCBD or LSD; or a character vector which indicate the sub plot factor(s) for Split Family of Design, or the horizontal factor(s) for the Strip Family of Design

`factor3` `NULL` for CRD, RCBD, LSD, Split Plot or Strip Plot; or a character vector which indicate the sub-sub plot factor(s) for Split-Split Design and Split-Split-Split Design; or the sub plot factor(s) for the Strip-Split Family of Design

`factor4` `NULL` for CRD, RCBD, LSD, Split Plot, Strip Plot, Split-Split Plot or Strip-Split Plot; or a character vector which indicate the sub-

sub-sub plot factor(s) for Split-Split-Split Design; or the sub-sub plot factor(s) for the Strip-Split-Split Design

`rep1` `NULL` or a character vector which indicate the blocking factor or the row blocking factor

`rep2` `NULL` or a character vector which indicate the column blocking factor

`set` `NULL` or a character vector which indicate the by variable

`descriptive` logical indicating whether the summary statistics will be computed

`normality` logical indicating whether normality test of the residuals using Shapiro-Wilk test will be performed

`homogeneity` logical indicating whether homogeneity test of the residuals using Bartlett's test will be performed

`pw.test` a character vector indicating the pairwise mean comparison procedure that will be performed or `NULL`

`pw.var` a character vector indicating the response variable where the pairwise mean comparison procedure specified in the argument `pw.test` will be performed or `NULL`

`contrastOption` `NULL` or a list containing the following components:

- `contrast.type` a numeric vector which indicates the code of the contrast analysis to be performed
- `contrast.f` a character string which indicates the factors where the contrast analysis will be performed
- `contrast.l` a character string

`sig` a single numeric value indicating the level of significance to be used to determine whether the effect is significant when the analysis of variance is performed and the level of significance to be used when one perform the pairwise mean comparison

`outputPath` `NULL` or path where the dataset containing the residuals and predicted values will be saved

Details

If at least one of the `pw.test` and `pw.test` is `NULL`, the default pairwise mean comparison will be performed for all significant effect in the analysis. If the level of the factor is less than or equal to 5, the Least Significant Difference test will be performed; otherwise, the Tukey's Honest Significant Difference test will be performed.

If both `pw.test` and `pw.test` are not `NULL`, `pw.test` is a character string which indicate the pairwise mean comparison method to be performed and can be any one of the following: '`LSD`' (Least Significant Difference test), '`duncan`' (Duncan Multiple Range Test), '`HSD`' (Tukey's Honest Significant Difference test), '`SNK`' (Student Newman-Keul's

Test), and/or `'scheffe'` (Scheffe's test). The `pw.var` is a character string which indicates the response variables where the speficied pairwise mean comparison method will be performed.

If `contrastOption` is not `NULL`, `contrast.type` is a numeric vector which indicates the code of the contrast analysis to be performed; the codes are: `2` for compare with control, `3` for orthogonal polynomial and `4` for user-specified contrast. The option orthogonal polynomial can only be performed if the levels of the factor are numeric. The `contrast.f` is a character string which indicates the factors where the contrast analysis will be performed. Only factors with more than two levels should be indicated. If compare with control will be performed, the `contrast.l` should contain the level of the factor which pertains to the control. If orthogonal polynomial will be performed, the `contrast.l` should contain the degree of orthogonal polynomial that will be performed. If user-specified contrast will be performed, the `contrast.l` should contain the number of contrast to be performed. The maximum number of contrast and the highest degree of orthogonal polynomial that can be performed is equal to the degrees of freedom of the factor being considered or the number of levels minus 1 of the factor being considered.

Value

A list containing the following components:

`data` a data frame containing the original data set with the predicted values and residual

`aovObject` a list of R object of class c`('aov', 'lm')` which contains the ANOVA table of each response variable

`rvWithSigEffect` a character vector indicating the response variable with at least one significant effect

`aovTable` a list of R object of class c`('anova', 'data.frame')`

`pwOption` a named list which contain the response variable, pariwise mean comparison procedure requested by user, a character vector contain the significant effect

`model` formula used when one perform analysis of variance

`model2`

`alpha` the level of significance used for the analysis

Author(s)

Reference(s)

See Also

Examples

## Example 1: Completely Randomized Design

## load the dataset "`Insecticides`" from the STAR package

```
data(Gerua)
```

## perform the analysis of variance in CRD using the default option with `Insecticide` as factor

```
AOVTest(data = 'Insecticides', design = 'CRD', respvar =
   'GrainYield', factor1 = 'Insecticide')
```

## Example 2: Randomized Complete Design

## load the dataset "`Gerua`" from the STAR package

```
data(Gerua)
```

## perform the analysis of variance in RCBD using the default function with `TPL` and `CULT` as factors and `REP` as the blocking factor

```
AOVTest(data = 'Gerua', design = 'RCBD', respvar = c('GRNYLD',
   'PNCLE'), factor1 = c('TPL', 'CULT')), rep1 = 'REP')
```

## perform the analysis of variance in RCBD that display the descriptive statistics of the response variable, test for normality of the residual and test for equality of variances

```
AOVTest(data = 'Gerua', design = 'RCBD', respvar = c('GRNYLD',
   'PNCLE'), factor1 = c('TPL', 'CULT')), rep1 = 'REP',
   descriptive = TRUE, normality = TRUE, homogeneity = TRUE)
```

## perform the analysis of variance in RCBD with user's choice of pairwise mean comparison method

```
AOVTest(data = 'Gerua', design = 'RCBD', respvar = c('GRNYLD',
   'PNCLE'), factor1 = c('TPL', 'CULT')), rep1 = 'REP', pw.test =
   c('LSD', 'duncan', 'HSD', 'SNK', 'scheffe'), pw.var =
   c('GRNYLD'), sig = 0.05)
```

## perform the analysis of variance in RCBD with contrast analysis (compare with control and orthogonal polynomial)

```
aovTest(design = 'RCBD', data = 'factor_4', respvar = c('wexr'),
   factor1 = c('var', 'wregime', 'das', 'depth'), rep1 = 'rep',
   contrastOption = list(contrast.type = c(2,3), contrast.f =
   c("das", "depth"), contrast.l = c("14", "1")))
```

AppendData *Append Two Data Frames*

Description

Append two data frame by common columns.

Usage

```
AppendData(MasterData, TransactionData, byMaster, byTransact,
           MasterVarKeep = NULL, TransactVarKeep = NULL)
```

Arguments

`MasterData, TransactionData`	a non-null character string which indicate the name of the data frame to be coerced to one
`byMaster, byTransact`	a character vector which indicate the name of the column in `MasterData` and `TransactionData`, respectively, which will be treated as common columns. See 'Details'.
`MasterVarKeep, TransactVarKeep`	`NULL` or list of variables in `MasterData` and `TransactionData`, respectively, that will be included aside from the common columns

Details

The `byMaster` and `byTransact` are both character vectors which specify the column names that will be combined as one. If `byMaster` and `byTransact` have different column names, the column names of the `byMaster` will be followed.

Value

A data frame. The columns are the columns that were combined followed by the remaining columns in the `MasterData` and then those from the `TransactionData`. If matching column names are included in the `MasterVarKeep` and `TransactVarKeep` these column names will have suffixes (".1" and ".2") appended to make the names of the resulting data frame unique. Additional column will be added (`Source`) which indicate from what data the observation came from. The column will have a value `1` which indicates that it came from the `MasterData` data frame and 2 which indicate that it came from the `TransactionData` data frame.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

bcPower

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

BIBDTest *Analysis of Variance for Balanced Incomplete Block Design*

Description

Perform analysis of variance for Balanced Incomplete Block Design and pairwise mean comparison on the adjusted means.

Usage

```
BIBDTest(data, respvar, trmt, block, method = NULL, descriptive =
         FALSE, alpha = 0.05)
```

Arguments

`data`	a non-null character string which indicates the name of the data frame
`respvar`	a non-empty character vector which indicates the name of the response variable
`trmt`	a non-null character string which indicates treatment
`block`	a non-null character string which indicates the block
`method`	`NULL` or a character vector which indicates method of pairwise mean comparison that will be performed. Valid value are: “lsd”, “tukey”, “snk” and “duncan”.
`descriptive`	logical: should descriptive statistics of the response variable be displayed?
`alpha`	a numeric value which indicate the level of significance. See ‘Details’.

Details

The `alpha` indicates the level of significance that will be used to determine whether the treatment effect is significant or not. This will also be the level of significant that will be used when performing pairwise mean comparison.

By default, the function will display a pairwise mean comparison. If the levels of the treatment is less than or equal to 5, the Least Significant Difference Test will be computed. Otherwise, the Tukey’s Honest Significant Difference Test will be performed.

Value

A list containing the following components:

`anovaTable`	a list which contains the ANOVA table of each response variable
`trmtStatistic`	a data frame which contains the treatment mean, adjusted treatment mean and adjust treatment effect for all response variable per levels of the treatment

`stdErr`	a data frame which contains the standard error of the difference, standard error of the adjusted treatment effect and standard error of the adjusted treatment mean for all response variable
`InterBlkEst`	a data frame which contains the interblock estimate for all response variable
`alpha`	a numeric value which indicate the level of significance used
`pairwise`	`NULL` or a list which contains the result of the pairwise mean comparison for all response variable. The list contains the method used, the critical value or critical range and the summary of the result of the pairwise mean comparison

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

## read the data set

```
dataTime <- read.csv(file = 'E:/BIB_TIME.csv')
```

## using the default option (descriptive statistics of the response variable is not displayed and perform the default pairwise mean comparison procedure)

```
BIBDTest(data = 'dataTime', respvar = 'time', trmt = 'catalyst',
         block = 'batch')
```

## perform Duncan Multiple Range Test and uses 10% as the level of significance

```
BIBDTest(data = 'dataTime', respvar = 'time', trmt = 'catalyst',
         block = 'batch', method = c('duncan'), descriptive =
         TRUE, alpha = 0.10)
```

BivariateCorrelationTest *Test of Association between paired samples*

Description

Perform test for association between paired samples using the Pearson’s product moment correlation coefficient, Kendall’s tau and Spearman’s rho.

Usage

```
BivariateCorrelationTest(data, var, method = “pearson”,
                         alternative = “two.sided”,
                         statistics = FALSE)
```

Arguments

`data`	a non-null character string which indicates the name of the data frame containing the variables specified in `var`
`var`	a non-empty character vector which indicates the name of the variables where test of association will be performed
`method`	a character vector indicating which correlation coefficient is to be used for the test and must be at least one of the following: “`pearson`” (default), “`kendall`” and/or “`spearman`”
`alternative`	a character string indicating the alternative hypothesis to be used and must be one of the following: “`two.sided`” (default), “`greater`” or “`less`”
`statistics`	logical indicating whether summary statistics will be computed for each of the variables in the object `var`

Details

Value

A list containing the following components:

`procedure`	a character string indicating the method used for measure the association between paired of variables
`estimate`	a matrix of the correlation coefficient between variables
`pvalue`	a matrix of the p-value of the test
`nobs`	a matrix of the number of paired observation used in measuring the association between pair of variables
`alternative`	a character string indicating the alternative hypothesis used

Author(s)

Alaine A. Gulles

Reference(s)

See Also

`cor`

```
cor.test
```

Examples

## load the data set `'Gerua'` from STAR package

```
data(Gerua)
```

## perform correlation analysis using the default options

```
BivariateCorrelationTest(data = 'Gerua', var = c('DFF', 'PLHT',
      'PNCLE', 'GRNYLD'))
```

## perform correlation analysis using the three available procedure

```
BivariateCorrelationTest(data = 'Gerua', var = c('DFF', 'PLHT',
      'PNCLE', 'GRNYLD'), method = c('pearson', 'kendall',
      'spearman'), alternative = 'two.sided', statistics = TRUE)
```

BoxsMTest

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

bptest

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

buildAOVModel

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

ClassInformation

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

ClusterAgglo

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

ClusterDivisive

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

ClusterKmeans

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

coef

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

---

combAOVTest

Description

Performs analysis of variance with factors treated as fixed.

Usage

```
combAOVTest(data, design, respvar, factor1, factor2 = NULL,
        factor3 = NULL, factor4 = NULL, rep1 = NULL, rep2 = NULL,
        set, descriptive = FALSE, normality = FALSE, homogeneity
        = FALSE, pwTest = NULL, pwVar = NULL, contrastOption =
        NULL, sig = 0.05, outputPath = NULL)
```

Arguments

`data`	a non-null character string which indicates the name of the data frame containing the variables specified in `respvar`, `factor1`, `factor2`, `factor3`, `factor4`, `rep1` and `rep2`.
`design`	a character string indicating the experimental design to be used and must be one of the following: “`CRD`” (Completely Randomized

Design) , “RCBD” (Randomized Complete Block Design) , "LSD" (Latin Square Design), "SplitCRD" (Split Plot in CRD), "SplitRCBD" (Split Plot in RCBD), "SplitLSD" (Split Plot in LSD), "Strip" (Strip Plot Design), "Split2CRD" (Split-Split Plot in CRD), "Split2RCBD" (Split-Split Plot in RCBD), "Split2LSD" (Split-Split Plot in LSD), "Strip-Split" (Strip-Split Plot Design), "Split3CRD" (Split-Split-Split Plot in CRD) , "Split3RCBD" (Split-Split-Split Plot in RCBD), "Split3LSD" (Split-Split-Split Plot in CRD), or "Strip-Split2" (Strip-Split-Split Plot Design).

respvar a non-empty character vector which indicates the response variable(s) to be used in the analysis

factor1 a non-empty character vector which indicates the factor(s) to be used for CRD, RCBD and LSD or the main plot factor(s) for Split Family of Design, or the vertical factor(s) for the Strip Family of Design

factor2 NULL for CRD, RCBD or LSD; or a character vector which indicate the sub plot factor(s) for Split Family of Design, or the horizontal factor(s) for the Strip Family of Design

factor3 NULL for CRD, RCBD, LSD, Split Plot or Strip Plot; or a character vector which indicate the sub-sub plot factor(s) for Split-Split Design and Split-Split-Split Design; or the sub plot factor(s) for the Strip-Split Family of Design

factor4 NULL for CRD, RCBD, LSD, Split Plot, Strip Plot, Split-Split Plot or Strip-Split Plot; or a character vector which indicate the sub-sub-sub plot factor(s) for Split-Split-Split Design; or the sub-sub plot factor(s) for the Strip-Split-Split Design

rep1 NULL or a character vector which indicate the blocking factor or the row blocking factor

rep2 NULL or a character vector which indicate the column blocking factor

set a non-empty character string which indicate the by variable

descriptive logical indicating whether the summary statistics will be computed

normality logical indicating whether normality test of the residuals using Shapiro-Wilk test will be performed

homogeneity logical indicating whether homogeneity test of the residuals using Bartlett’s test will be performed

pw.test a character vector indicating the pairwise mean comparison procedure that will be performed or NULL

pw.var a character vector indicating the response variable where the pairwise mean comparison procedure specified in the argument pw.test will be performed or NULL

contrastOption NULL or a list containing the following components:

contrast.type a numeric vector which indicates the code of the contrast analysis to be performed

contrast.f a character string which indicates the factors where the contrast analysis will be performed

contrast.l a character string

sig a single numeric value indicating the level of significance to be used to determine whether the effect is significant when the analysis of variance is performed and the level of significance to be used when one perform the pairwise mean comparison

outputPath NULL or path where the dataset containing the residuals and predicted values will be saved

Details

If at least one of the pw.test and pw.test is NULL, the default pairwise mean comparison will be performed for all significant effect in the analysis. If the level of the factor is less than or equal to 5, the Least Significant Difference test will be performed; otherwise, the Tukey's Honest Significant Difference test will be performed.

If both pw.test and pw.test are not NULL, pw.test is a character string which indicate the pairwise mean comparison method to be performed and can be any one of the following: 'LSD' (Least Significant Difference test), 'duncan' (Duncan Multiple Range Test), 'HSD' (Tukey's Honest Significant Difference test), 'SNK' (Student Newman-Keul's Test), and/or 'scheffe' (Scheffe's test). The pw.var is a character string which indicates the response variables where the speficied pairwise mean comparison method will be performed.

If contrastOption is not NULL, contrast.type is a numeric vector which indicates the code of the contrast analysis to be performed; the codes are: 2 for compare with control, 3 for orthogonal polynomial and 4 for user-specified contrast. The option orthogonal polynomial can only be performed if the levels of the factor are numeric. The contrast.f is a character string which indicates the factors where the contrast analysis will be performed. Only factors with more than two levels should be indicated. If compare with control will be performed, the contrast.l should contain the level of the factor which pertains to the control. If orthogonal polynomial will be performed, the contrast.l should contain the degree of orthogonal polynomial that will be performed. If user-specified contrast will be performed, the contrast.l should contain the number of contrast to be performed. The maximum number of contrast and the highest degree of orthogonal polynomial that can be performed is equal to the degrees of freedom of the factor being considered or the number of levels minus 1 of the factor being considered.

Value

A list containing the following components:

`data` a data frame containing the original data set with the predicted values and residual

`aovObject` a list of R object of class `c('aov', 'lm')` which contains the ANOVA table of each response variable

`rvWithSigEffect` a character vector indicating the response variable with at least one significant effect

`aovTable` a list of R object of class `c('anova', 'data.frame')`

`pwOption` a named list which contain the response variable, pariwise mean comparison procedure requested by user, a character vector contain the significant effect

`model` formula used when one perform analysis of variance

`model2`

`alpha` the level of significance used for the analysis

Author(s)

Reference(s)

See Also

Examples

```
## Example 1: Completely Randomized Design
## load the dataset “Insecticides” from the STAR package
data(Gerua)
## perform the analysis of variance in CRD using the default option with Insecticide
   as factor
AOVTest(data = 'Insecticides', design = 'CRD', respvar =
   'GrainYield', factor1 = 'Insecticide')
```

CombineFactorLevels *Combine Levels of Several Factors*

Description

This function concatenates the levels of two or more variables of a data frame.

Usage

```
CombineFactorLevels(data, concatVar, targetName = NULL)
```

Arguments

`data`	a data frame
`concatVar`	names of two or more columns in the data frame that identify the variables whose levels will be concatenated
`targetName`	`NULL` or a character string which indicate the name of the new variable that will be created

Details

The values of `concatVar` will be combined by "_". If `targetName` is `NULL` the name of the new column that contains the combined levels is `NewVar`.

Value

A data frame with the new factor.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

## load the dataset `'Gerua'` from the STAR package

```
data(Gerua)
```

## create a new variable which combine the levels of TPL and CUL

```
newData <- CombineFactorLevels(data = Gerua, concatVar = c('TPL',
           'CULT'))
```

## create a new variable named `'TPLnCULT'` which combine the levels of TPL and CULT

```
newData <- CombineFactorLevels(data = Gerua, concatVar = c('TPL',
           'CULT'), targetName = 'TPLnCULT')
```

CompareMeans *Student's t-Test*

Description

Performs one and two sample t-tests for paired and independent samples.

Usage

```
CompareMeans(data, varX, varY = NULL, grp = NULL, testVal = 0,
             procedure = c("one", "paired", "independent"),
             alternative = c("two.sided", "greater", "less"),
             statistics = FALSE, CI = FALSE, confLevel = 0.95,
             normality = NULL, alpha = 0.05)
```

Arguments

`data` a non-null character string which indicates the name of the data frame

`varX` a non-empty character vector which indicates the response variables. See 'Details'.

`vary` `NULL` or a non-empty character vector which indicates the response variables. See 'Details'.

`grp` `NULL` a non-empty character vector which indicates the grouping variable. See 'Details'.

`testVal` a single number indicating the hypothesized value of the mean (or difference in means if one is performing a two sample t-test)

`procedure` a character string which indicates type of t-test to be performed. Must be one of the following: `"one"`(one sample t-test; default), `"paired"` (related sample t-test). or `"independent"` (two independent sample t-test).

`alternative` a character string indicating the alternative hypothesis to be used. Must be one of the following: "`two.sided`" (default), "`greater`" or "`less`"

`statistics` logical: should summary statistics will be computed for each of the variables in the object `varX` and/or `vary`?

`CI` logical: should confidence interval will be computed?

`confLevel` a single number indicating confidence level of the interval

`normality` `NULL` (default) or a character string indicating which method to be used to test for normality; must be one of the following: "`swilk`", "`sfrancia`", "`cramer`", "`anderson`" and/or "`ks`".

`alpha` a single number indicating the level of significance when performing test on equality of variance for two independent sample

Details

If `procedure` is `paired` then both `varX` and `varY` must be specified and they must be of the same length.

If `procedure` is `independent` then `grp` must be specified.

If `CI` is `TRUE` then confidence interval of the mean or mean difference will be computed. By default, a 95% confidence interval is computed. To construct other confidence interval, another value of `confLevel` must be specified.

If `statistics` is `TRUE` then numerical descriptive statistics (number of observation, mean or mean difference, standard deviation and standard error) will be computed for all variables specified in the argument `varX`.

Value

A list containing the following components:

`statistics` a data frame of the result of the t-test which include the following columns:

`equalVariance` a data frame of the result of the equality of variance when two independent sample test was performed

`alpha` the level of significance used in testing for the equality of variance when two independent sample test was performed

`estimate` a data frame of the descriptive statistics of the variable

`confLevel` the confidence level used when the confidence interval of the mean or mean difference was performed

`nullValue` the specified hypothesized value of the mean or mean difference depending a one-sample test or two-sample test

`alternative` a character string describing the alternative hypothesis used

`method` a character string indicating the type of t-test performed

Author(s)

Reference(s)

See Also

`t.test`

`var.test`

`FoldedFTest`

Examples

```
## load the dataset 'Gerua' from the STAR package
data(Gerua)
## perform one sample t-test for the variables
CompareMeans(data = 'Gerua', varX = c('DFF', 'PLHT', 'PNCLE',
             'GRNYLD', testVal = 0, procedure = 'one',
             alternative = 'two.sided')
## perform one sample t-test and display the summary statistics of the variables,
   construct a 90% confidence interval and request for test of normality using the
   Shapiro-Wilk
CompareMeans(data = 'Gerua', varX = c('DFF', 'PLHT', 'PNCLE',
             'GRNYLD', testVal = 0, procedure = 'one',
             alternative = 'two.sided', statistics = TRUE, CI =
             TRUE, confLevel = 0.90, normality = 'swilk')
## perform two independent sample t-test
CompareMeans(data = 'Gerua', varX = c('DFF', 'PLHT', 'PNCLE',
             'GRNYLD', grp = 'TPL', procedure = 'independent',
             alternative = 'two.sided', alpha = 0.05)
```

CompareMedian

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

CompareProportion

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

ConstructAOVTable

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

ContingencyCoef *Contingency Coefficient*

Description

Usage

```
ContingencyCoef(x, y = NULL)
```

Arguments

`x`	a vector or a matrix
`y`	a vector; ignored if `x` is a matrix

Details

If `x` is a matrix it should have at least two rows and columns. Otherwise, `x` and `y` must be vectors or factors of the same length.

Value

Author(s)

Reference(s)

See Also

`CramersV`, `LikelihoodRatioTest`, `PhiCoef`

Examples

ContrastCompute

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

CramersV *Cramer's V Statistics*

Description

Computes the Cramer's V statistics.

Usage

```
CramersV(x, y = NULL)
```

Arguments

`x`	a vector or a matrix
`y`	a vector; ignored if `x` is a matrix

Details

If `x` is a matrix it should have at least two rows and columns. Otherwise, `x` and `y` must be vectors or factors of the same length.

Value

A single numeric value which indicates the Cramer's V statistics.

Author(s)

Reference(s)

See Also

`ContingencyCoef`, `LikelihoodRatioTest`, `PhiCoef`

Examples

Crosstab *Cross tabulation*

Description

Performs cross tabulation and test of association for two-way table.

Usage

```
Crosstab(data, rowVar, colVar, obsFreq = TRUE, expFreq = FALSE,
         totPercent = FALSE, rowPercent = FALSE,
         colPercent = FALSE, chisq = FALSE, phi = FALSE,
         cramersv = FALSE, contingency = FALSE)
```

Arguments

`data` a non-null character string which indicates the name of the data frame containing the variables specified in `rowVar` and `colVar`

`rowVar` a non-empty character vector which indicates the row variable(s)

`colVar` a non-empty character vector which indicates the column variable(s)

`obsFreq` a logical value. If `TRUE`, the observe frequency will be displayed

`expFreq` a logical value. If `TRUE`, the expected frequency will be displayed

`totPercent` a logical value. If `TRUE`, the percentage total will be displayed

`rowPercent` a logical value. If `TRUE`, the row percentage will be displayed

`colPercent` a logical value. If `TRUE`, the column percentage will be displayed

`chisq` a logical value. If `TRUE`, the result of the chi-square test of independence will displayed

`phi` a logical value. If `TRUE`, the result of the phi coefficient will be displayed

`cramersv` a logical value. If `TRUE`, the result of the cramer’s V statistics displayed

`contingency` a logical value. If `TRUE`, the result of the contingency coefficient will be displayed

Details

Value

Author(s)

Reference(s)

See Also

Examples

```
Tenure_Status_and_Adoption <- read.csv(file =
     'E:/TenureStatusandAdoption.csv', header = TRUE, sep = ',',
     quote = '"', na.strings = c('NA', '.', ' ', ''))

Crosstab(rowVar = c('TenureStatus'), colVar = c('FarmerClassif'),
     data = 'Tenure_Status_and_Adoption', obsFreq = TRUE,
     expFreq = TRUE, totPercent = TRUE, rowPercent = TRUE,
     colPercent = TRUE, chisq = TRUE, phi = TRUE,
     cramersv = TRUE, contingency = TRUE)
```

css *Corrected Sum of Squares*

Description

Compute the corrected sum of squares.

Usage

```
css(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Reference(s)

See Also

Examples

cv *Coefficient of Variation*

Description

Computes the coefficient of variation.

Usage

```
cv(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Reference(s)

See Also

Examples

DataAttribute *Data Attributes*

Description

This function determines the variables in the data frame and the type of the variable.

Usage

```
DataAttribute(data)
```

Arguments

`data` a data frame or a character string indicating the name of the data frame that will be used

Details

Value

A data frame containing the following columns:

`VAR NAME` all the variable names in the data frame

`TYPE` type of the variable e.g. (factor, integer, numeric, logical, etc)

`NLEVELS` if the variable is a factor, how many categories the variable has

`LEVELS` if the variable is a factor, contains the list of the levels of the factor

Author(s)

Reference(s)

See Also

Examples

```
GERUA <- read.csv(file = 'E:gerua_split.csv')
DataAttributes(GERUA)
```

DataTransformation *Data Transformation*

Description

This function performs data transformation using the logarithm, natural logarithm, square root, exponential, power transformation and standardize.

Usage

```
DataTransformation(data, var, transformation = c("log", "ln",
                   "sqrt", "exp", "power", "standardize"),
                   targetName = "newVar")
```

Arguments

`data`	a data frame.
`var`	a non-empty character string which indicates the variable that will be transformed
`transformation`	a character string which indicates the method of transformation that will be used. Possible values are: `log` (logarithm), `ln` (natural logarithm), `sqrt` (square root), `exp` (exponential), `power` (power) and `standardize` (standardize).

Details

This function requires that the variable to be transformed should be non-negative values.

The power transformation makes use of the Box-Cox transformation method.

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

```
Gerua <- read.csv(file = 'E:gerua_split.csv', header = TRUE)

DataTransform(data  =  'Gerua',  var  =  'DFF',  transformation  =
              'sqrt', targetName = 'DFF_sqrt')
```

DescriptiveStatistics *Numerical Descriptive Statistics*

Description

It displays univariate summary statistics for several variables per levels of a grouping variable in a single table.

Usage

```
DescriptiveStatistics(data, var, grp = NULL,
                           statistics = c("nnmiss", "mean", "sd"))
```

Arguments

`data`	a non-null character string which indicates the name of the data frame containing the variables specified in `var` and `grp`
`var`	a non-empty character vector which indicates the name of the variable(s) where summary statistics will be computed
`grp`	`NULL` (default) or a character vector which indicates the name of the variable(s) whose value define the subgroup for generating the descriptive statistic
`statistics`	a non-empty character vector which indicates the statistics to be computed. Must be at least one of the following: "`n`", "`nnmiss`", "`nmiss`", "`sum`", "`css`", "`ucss`", "`se.skew`", "`se.kurtosis`", "`range`", "`iqr`", "`var`", "`sd`", "`se.mean`", "`cv`", "`mean`", "`median`", "`mode`", "`min`", "`max`", "`q1`", "`q3`", "`skew`" and/or "`kurtosis`".

Details

The elements of `var` should correspond to numeric variable in the data frame.

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

## read the data set

```
Gerua <- read.csv(file = 'E:/gerua_split.csv')
```

## compute the default statistics for variables DFF, PLHT, PNCLE and GRNYLD

```
DescriptiveStatistics(data = “Gerua”, var = c(“DFF”, “PLHT”,
                      “PNCLE”, “GRNYLD”))
```

## compute the mean, standard deviation and skewness for variables DFF, PLHT, PNCLE and GRNYLD per level of TPL

```
DescriptiveStatistics(data = “Gerua”, var = c(“DFF”, “PLHT”,
                      “PNCLE”, “GRNYLD”), grp = “TPL”,
                      statistics = c(“mean”, “sd”, “skew”))
```

design.alpha

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

designAlphaLattice *Randomization for Alpha Lattice Design*

Description

Generate randomization for Alpha Lattice Design.

Usage

```
designAlphaLattice(generate, blksize, r = 2, trial = 1, file =
                   NULL)
```

Arguments

`generate`	a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors
`blksize`	number of plot per block
`r`	a single numeric value which indicates the number of replicates
`trial`	a single numeric value which indicates the number of trial
`file`	a character string indicating the labels for the factors or `NULL`

Details

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designLSD, designRCBD,
designSplit, designStrip
```

Examples

## generate 10 levels of a factor

```
designAlphaLattice(list(Treatment = 1:30), blksize = 3, r = 2,
                   trial = 1)
```

designAugmented *Randomization for Augmented Design*

Description

Generates randomization for augmented design in randomized complete block and Latin Square.

Usage

```
designAugmented(checkTrmt, newTrmt, r = NULL, trial = 1, design =
                c("rcbd", "lsd"), factorName = NULL, display =
                TRUE, file = NULL)
```

Arguments

`checkTrmt`	a non-null character strings which indicates the replicated treatment level or a single numeric value which indicates the number of levels of the replicated treatment
`newTrmt`	a non-null character strings which indicates the level of the unreplicated treatment or a single numeric value which indicates the number of levels of the unreplicated treatment
`r`	a single numeric value which indicates the number of replicate or `NULL`
`trial`	a single numeric value which indicates the number of trial
`design`	a character string which indicate the experimental design used. Must be any one of the following: `'rcbd'` (for randomized complete block design) or `'lsd'` (Latin Square design)
`display`	logical which indicate whether an output should be displayed in the R console
`file`	a character string indicating the labels for the factors or `NULL`

Details

If the design is `'rcbd'`, `r` should be a numeric value. Otherwise, `NULL`.

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designLSD, designRCBD,
designSplit, designStrip
```

Examples

## generate randomization for augmented RCBD wherein replicated treatment has 6 levels replicated 5 times and unreplicated treatment has 60 levels

```
designAugmented(checkTrmt = 6, newTrmt = 60, r = 5, trial = 2,
   design = "rcbd")
```

designAugmentedLSD

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

designAugmentedRCB

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

designBIBD *Randomization for Balanced Incomplete Block Design*

Description

Generate randomization for balanced incomplete block design.

Usage

```
designBIBD(generate, blkSize, trial = 1, display = TRUE,
           file = NULL)
```

Arguments

`generate`	a list that specify the factor. See details.
`blkSize`	a single numeric value which indicates the plots per block or the block size. See details.
`trial`	a single numeric value which indicates the number of trial
`display`	logical which indicate whether an output should be displayed in the R console
`file`	a character string indicating the labels for the factors or `NULL`

Details

The `generate` is a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors

The `blkSize` should be less than the number of treatment.

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designLSD, designRCBD,
designSplit, designStrip
```

Examples

## generate 10 levels of a factor

```
designCRD(list(variety = 10))
```

designCRD *Randomization for Completely Randomized Design*

Description

Generate randomization for completely randomized design for any number of factors.

Usage

```
designCRD(generate, r = 2, trial = 1, display = TRUE, file =
         NULL)
```

Arguments

`generate`	a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors
`r`	a single numeric value which indicates the number of replicates
`trial`	a single numeric value which indicates the number of trial
`display`	logical which indicate whether an output should be displayed in the R console
`file`	a character string indicating the labels for the factors or `NULL`

Details

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designLSD, designRCBD,
designSplit, designStrip
```

Examples

## generate 10 levels of a factor replicated twice for 1 trial

```
designCRD(list(variety = 10))
```

## generate a randomization for treatment combination of a 10 x 5 factorial experiment replicated 5 times for 3 trials

```
designCRD(generate = list(variety = 1:10, fertilizer = 5), r = 5,
   trial = 3)
```

designLattice *Randomization for Lattice Design*

Description

Generates randomization for partial and balanced lattice designs.

Usage

```
designLattice(generate, r = 1, trial = 1, file = NULL)
```

Arguments

`generate`	a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors
`r`	a single numeric value which indicates the number of replicates
`trial`	a single numeric value which indicates the number of trial
`file`	a character string indicating the labels for the factors or `NULL`

Details

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

`designAugmented, designLattice, designCRD, designRCBD, designSplit, designStrip`

Examples

## generate 10 levels of a factor

designLSD *Randomization for Latin Square Design*

Description

Generate randomization for Latin Square design for any number of factors.

Usage

```
designLSD(generate, trial = 1, display = TRUE, file = NULL)
```

Arguments

`generate`	a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors
`trial`	a single numeric value which indicates the number of trial
`display`	logical which indicate whether an output should be displayed in the R console
`file`	a character string indicating the labels for the factors or `NULL`

Details

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designCRD, designRCBD,
designSplit, designStrip
```

Examples

## generate 10 levels of a factor

```
designLSD(list(variety = 10))
```

## generate a randomization for treatment combination of a 10 x 5 factorial experiment replicated 5 times for 3 trials

```
designLSD (generate = list(variety = 1:10, fertilizer = 5), trial
   = 3)
```

designRCBD — *Randomization for Randomized Complete Block Design*

Description

Generate randomization for randomized complete block design for any number of factors.

Usage

```
designRCBD(generate, r = 2, trial = 1, display = TRUE, file =
         NULL)
```

Arguments

`generate` a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors

`r` a single numeric value which indicates the number of replicates

`trial` a single numeric value which indicates the number of trial

`display` logical which indicate whether an output should be displayed in the R console

`file` a character string indicating the labels for the factors or `NULL`

Details

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designCRD, designLSD,
designSplit, designStrip
```

Examples

## generate 10 levels of a factor

```
designRCBD(list(variety = 10))
```

## generate a randomization for treatment combination of a 10 x 5 factorial experiment replicated 5 times for 3 trials

```
designRCBD (generate = list(variety = 1:10, fertilizer = 5), r =
   5, trial = 3)
```

designSplit *Randomization for Family of Split Plot Design*

Description

Generates randomization for split plot design, split-split plot design and split-split-split plot design in complete block design (CRD), randomized complete block design (RCBD) or Latin Square design (LSD).

Usage

```
designSplit(main, sub, ssub = NULL, sssub = NULL, r = NULL,
            trial = 1, design = c('crd', 'rcbd', 'lsd'),
            display = TRUE, file = NULL)
```

Arguments

`main`	a list that specify the main plot factor. See details.
`sub`	a list that specify the sub plot factor. See details.
`ssub`	`NULL` or a list that specify the sub-sub plot factor. See details.
`sssub`	`NULL` or a list that specify the sub-sub-sub plot factor. See details.

`r` NULL or a single numeric value which indicates the number of replicates. See details.

`trial` a single numeric value which indicates the number of trial

`design` a character string indicating the experimental design to be used and must be one of “crd” (default), “rcbd”, or “lsd”.

`display` logical which indicate whether an output should be displayed in the R console

`file` a character string indicating the labels for the factors or NULL

Details

The `main`, `sub`, `ssub` and `sssub` are list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors

If the design is `crd` or `rcbd`, `r` should be non-null. Otherwise, `r` should be equal to `NULL`.

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designLSD, designRCBD,
designSplit, designStrip
```

Examples

## generate a randomization for split-plot design in CRD with variety (with 5 levels) as main plot factor and nitrogen (with 4 levels) as subplot factor, replicated 3 times conducted for 2 trials

```
designSplit(main = list(variety = 1:5), sub = = list(nitrogen =
            1:4), r = 3,  trial = 2, design = 'crd')
```

## generate a randomization for split-split-plot design in RCBD with A (with 5 levels) as main plot factor, B (with 4 levels) as subplot factor, and C (with 2 levels) as sub-sub plot factor with 3 blocks conducted for 2 trials

```
designSplit(main = list(A = 5), sub = = list(B = 4), ssub =
            list(C = 2), r = 3,  trial = 2, design = 'rcbd')
```

## generate a randomization for split-split-split-plot design in LSD with A (with 5 levels) as main plot factor, B (with 4 levels) as subplot factor, C (with 2 levels) as sub-sub plot factor and D (with 2 levels) as sub-sub-sub plot factor conducted for 2 trials

```
designSplit(main = list(A = 5), sub = = list(B = 4), ssub =
            list(C = 2), ), sssub = list(D = 2),  trial = 2,
            design = 'lsd')
```

designStrip *Randomization for the Family of Strip Plot Design*

Description

Generate randomization for strip plot design, strip-split plot design and strip-split-split plot design

Usage

```
designStrip(vertical, horizontal, sub = NULL, ssub = NULL, r = 2,
            trial = 1, display = TRUE, file = NULL)
```

Arguments

`vertical`	a list that specify the vertical factor. See details.
`horizontal`	a list that specify the horizontal factor. See details.
`sub`	`NULL` or a list that specify the sub plot factor. See details.
`ssub`	`NULL` or a list that specify the sub-sub plot factor. See details.
`r`	a single numeric value which indicates the number of blocks
`trial`	a single numeric value which indicates the number of trial
`display`	logical which indicate whether an output should be displayed in the R console
`file`	a character string indicating the labels for the factors or `NULL`

Details

The `vertical, horizontal`, `sub` and `ssub` are list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

Package agricolae

See Also

```
designAugmented, designLattice, designLSD, designRCBD,
designSplit, designStrip
```

Examples

## generate 10 levels of a factor

```
designCRD(list(variety = 10))
```

## generate a randomization for treatment combination of a 10 x 5 factorial experiment replicated 5 times for 3 trials

```
designCRD(generate = list(variety = 1:10, fertilizer = 5), r = 5,
   trial = 3)
```

FactorList *List of Factors and its levels*

Description

Generate a named list which contains the levels of a factor.

Usage

```
FactorList(generate)
```

Arguments

`generate` a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors

Details

If the input is a numeric value, the levels of the factor that will be generated is

Value

A named list which contain the levels of one of more factors.

Author(s)

Alaine A. Gulles

Reference(s)

Package dae

See Also

`GenerateFactor`

Examples

## if the input is a numeric value, output will be a list named `A` with levels `A1, A2, A3, A4, A5`

```
generate1 = list(A = 5)
FactorList(generate1)
```

## output will be a list with two elements named `variety` with levels `variety1, variety2, variety3, variety4, variety5` named `nitrogen` with levels `1, 2, 3, 4, 5`

```
FactorList(list(variety = 5, nitrogen = 1:5))
```

GenerateFactor *Generate levels a factor*

Description

Generate the levels of the treatment or levels of the treatment combination.

Usage

```
GenerateFactor(generate, times = 1)
```

Arguments

`generate`	a list of named object that specify the factors whose levels are to be generated. The component of should be either *i*) a single numeric value that is the number of levels or *ii*) a character vector that contains the labels of the levels of the factors
`times`	the number of times to repeat the whole generated patter of levels generated

Details

Value

A data frame of generated levels with columns corresponding to the name of the factors in the generate list.

Warning

Do not use the factor labels F and T as these might be confused with FALSE and TRUE.

Author(s)

Alaine A. Gulles

Reference(s)

Package dae

See Also

`FactorList`

Examples

## generate the factor Variety with levels `Variety1`, `Variety2`, `Variety3`, and `Variety4`

```
GenerateFactor(list(Variety = 4, Fertilizer = 1:5))
```

## generate the factor Fertilizer with levels `1`, `2`, `3`, `4` and `5`

```
GenerateFactor(list(Fertilizer = 1:5))
```

## generate the treatment combination of two factors Variety with levels `Variety1`, `Variety2`, `Variety3`, and `Variety4` and Fertilizer with levels `1`, `2`, `3`, `4` and `5`

```
GenerateFactor(list(Variety = 4, Fertilizer = 1:5))
```

HeteroskedasticityTest *Test for Homogeneity of Variances*

Description

Perform test for homogeneity of variance using the Bartlett's test and Levene's test.

Usage

```
HeteroskedasticityTest(data, var, grp, method = "bartlett")
```

Arguments

`data` a non-null character string which indicates the name of the data frame containing the variables specified in `var` and `grp`

`var` a non-empty character vector which indicates the name of the variable(s)

`grp` a non-null character string which indicates the grouping variable

`method` a character string indicating which method to be used for testing the equality of variances; must be one of the following: "`bartlett`" (default) and/or "`levene`"

Details

Value

A data frame

Author(s)

Reference(s)

See Also

`bartlett.test, levene.test`

Examples

## read the data set

```
Gerua <- read.csv(file = 'E:/gerua_split.csv')
```

## perform heteroskedasticity test for variables DFF and PLHT per level of TPL

```
HeteroskedasticityTest(data = 'Gerua', var = c('DFF', 'PLHT'),
     grp = c('TPL'), method = c('bartlett', 'levene'))
```

is.valid.name

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

is.wholenumber *Wholenumber*

Description

Determine if the input is a wholenumber or not.

Usage

```
is.wholenumber(x, tol = .Machine$double.eps^0.5)
```

Arguments

`x`	an object to be tested
`tol`	a numerical tolerance

Details

Value

Author(s)

Reference(s)

See Also

Examples

kurtosis *Kurtosis*

Description

Computes the kurtosis coefficient of a variable.

Usage

```
kurtosis(x, na.rm = TRUE)
coefKurtosis(x, na.rm = TRUE)
```

Arguments

`x` a numeric vector or data frame

`na.rm` a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Reference(s)

See Also

Examples

LikelihoodRatioTest *Likelihood Ratio Test*

Description

Usage

```
LikelihoodRatioTest(x, y = NULL)
```

Arguments

`x`	a vector or a matrix
`y`	a vector; ignored if `x` is a matrix

Details

If `x` is a matrix it should have at least two rows and columns. Otherwise, `x` and `y` must be vectors or factors of the same length.

Value

Author(s)

Reference(s)

See Also

`ContingencyCoef, CramersV, PhiCoef`

Examples

LinearRegressionAnalysis

Description

Perform linear regression analysis.

Usage

```
LinearRegressionAnalysis(data, depVar, indepVar, constant = TRUE,
                         statistics = TRUE, confInt = FALSE, confLevel
                         = 0.95, covMatrix = FALSE, normality = NULL,
                         heteroskedasticity = NULL, autoCorr = FALSE,
                         VIF = FALSE, COOKS = FALSE, leverage = FALSE)
```

Arguments

`data` a non-null character string which indicates the name of the data frame

`depVar` a non-empty character vector which indicates the dependent variable

`indepVar` a non-empty character vector which indicates the independent variable

`statistics` logical: should descriptive statistics of the dependent and independent variables be displayed? Default is `FALSE`. See ‘Details’.

`confInt` logical: should confidence interval of the regression constant be computed? Default is `FALSE`. See ‘Details’.

`confLevel` a single numeric value which indicate the level of confidence to be used when computing the confidence interval of the regression coefficient

`covMatrix` logical: should the covariance matrix of the dependent and independent variables be displayed? Default is `FALSE`.

`normality` `NULL` (default) or a non-empty character vector which indicates test for normality to be performed

`heteroskedasticity` `NULL` (default) or non-empty character vector which indicates test for heteroskedasticity to be performed

`autoCorr` logical: should durbin Watson statistics be computed? Default is `FALSE`. See ‘Details’.

`VIF` logical: should variance inflation factor be computed? Default is `FALSE`. See ‘Details’.

`COOKS` logical: should Cook’s distance be save? Default is `FALSE`.

`leverage` logical: should leverage value be save? Default is `FALSE`.

Details

Value

A list with the following components:

`dataframe` data frame of the original data set with the predicted and residual values.

`modelFit` R object

Author(s)

Reference(s)

See Also

Examples

```
LinearRegressionAnalysis(data "dataRead", depVar= c("GRNYLD"),
                         indepVar = c("PNCLE"), constant = TRUE,
                         statistics = TRUE, confInt = TRUE, confLevel =
                         0.95, covMatrix = TRUE, normality = NULL,
                         heteroskedasticity = FALSE, autoCorr = TRUE,
                         VIF = TRUE, COOKS = FALSE, leverage = FALSE)
```

MergeData *Merge Dataframe*

Description

Merge two data frame.

Usage

```
MergeData (MasterData, TransactionData, byMaster, byTransact,
           MasterVarInclude = names(MasterData),
           TransactVarInclude = names(TransactionData),allMaster
           = FALSE, allTransact = FALSE)
```

Arguments

`MasterData`

`TransactionData`

`byMaster, byTransact`

`MasterVarInclude = names(MasterData)`

`TransactVarInclude = names(TransactionData)`

`allMaster = FALSE`

`allTransact = FALSE`

Details

Value

Author(s)

Reference(s)

See Also

Examples

modalValue *Mode*

Description

Determine the modal value of a variable if it exist.

Usage

```
modalValue(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Reference(s)

See Also

Examples

NormalityTest *Test for Normality*

Description

Performs test for normality using the Wilk-Shapiro, Shapiro-Francia, Cramer-von Mises, Anderson-Darling and Kolmogorov-Smirnov test.

Usage

```
NormalityTest(data, var, grp = NULL, method = “swilk”)
```

Arguments

`data` a non-null character string which indicates the name of the data frame containing the variables specified in `var` and `grp`

`var` a non-empty character vector which indicates the name of the variable(s)

`grp` `NULL` or a character string which indicates the grouping variable

`method` a character string indicating which method to be used to test for normality; must be one of the following: “`swilk`” (default), “`sfrancia`”, “`cramer`”, “`anderson`” and/or “`ks`”.

Details

Value

A data frame.

Author(s)

Reference(s)

Package ‘nortest’

See Also

Examples

## read the data set

```
Gerua <- read.csv(file = 'E:/gerua_split.csv')
```

## perform normality test for variables DFF and PLHT using the default method (Shapiro-Wilk test)

```
NormalityTest(data = 'Gerua', var = c('DFF', 'PLHT'))
```

## perform normality test for variables PNCLE and GRNYLD per level of TPL

```
NormalityTest(data = 'Gerua', var = c('PNCLE', 'GRNYLD'), grp =
      c('TPL'), method = c('swilk', 'sfrancia', 'ks', 'cramer',
      'anderson'))
```

PhiCoef *Phi Coefficient*

Description

Usage

```
PhiCoef(x, y = NULL)
```

Arguments

`x`	a vector or a matrix
`y`	a vector; ignored if `x` is a matrix

Details

If `x` is a matrix it should have at least two rows and columns. Otherwise, `x` and `y` must be vectors or factors of the same length.

Value

Author(s)

Reference(s)

See Also

`ContingencyCoef, CramersV, LikelihoodRatioTest`

Examples

printCorrMatrix *Print the Correlation Matrix*

Description

Display the correlation matrix with its p-value and number of observation like SAS.

Usage

```
printCorrMatrix(coef, pval = NULL, n)
```

Arguments

`coef`	a matrix of the correlation coefficient
`pval`	`NULL` or a matrix p-value
`n`	a matrix of the number of observation

Details

Value

Author(s)

Reference(s)

See Also

`cor`

`cor.test`

Examples

printDataFrame *Printing Data Frames*

Description

Print a data frame.

Usage

```
printDataFrame(data)
```

Arguments

`data`	object of class `data.frame`

Details

Value

Author(s)

Reference(s)

See Also

`print`

`data.frame`

Examples

randomizeLattice *Randomization for Lattice Design*

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

skewness *Skewness Coefficient*

Description

Computes for the skewness coefficient.

Usage

```
skewness(x, na.rm = TRUE)
skew(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Reference(s)

See Also

Examples

SortCases *Sort the observation of a dataset*

Description

This function rearranges the observation of the data frame based on the value of one or more sorting variables.

Usage

```
SortCases(data, var, grp, sortBy = "ascending")
```

Arguments

`data`	a data frame or name of the data frame
`var`	a non-empty character vector which indicates the variable(s) which will be the basis for sorting the data frame
`sortBy`	a character string or vector which indicates the order by which the variable will be sorted. Possible value are “`ascending`” and/or “`descending`”

Details

If `var` contains more than one item, observations are sorted for each variable within categories of the preceding variable of the argument `var`. Further, each variable can be rearrange in ascending or descending order, with ascending order as the default option.

The default value for `sortBy` is `ascending`. The arguments `var` and `sortBy` must be of the same length. Incase `var` contains more than one item and `sortBy` has only one item, the variables will be rearrange according to `sortBy`. If `sortBy` contains more items than `var`, the programs creates a new vector of `sortBy` with the same length as `var`. If `var` contains more items than `sortBy`, the programs creates a new vector of `sortBy` with the same length as `var`

Value

A data frame.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

---

stdkurtosis *Standard Error of Kurtosis*

Description

Computes for the standard error of kurtosis.

Usage

```
stdkurtosis(x, na.rm = TRUE)
se.kurtosis(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Reference(s)

See Also

Examples

stdmean *Standard Error of the Mean*

Description

Computes for the standard error of the mean.

Usage

```
stdmean(x, na.rm = TRUE)
se.mean(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

stdskewness *Standard Error of Skewness*

Description

Computes for the standard error of skewness.

Usage

```
stdskewness(x, na.rm = TRUE)
se.skewness(x, na.rm = TRUE)
```

Arguments

`x`	a numeric vector or data frame
`na.rm`	a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

ToLong *Restructure Data Frame to Long Format*

Description

This function restructures a data frame from wide format to long format.

Usage

```
ToLong(data, varying, timevar, vnames, idvar, label = NULL)
```

Arguments

`data`	a data frame or name of the data frame
`varying`	names of one or more variables in the long format that identify multiple records form the same group/individual
`timevar`	the variable in long format that differentiates the multiple records from the same group or individual
`vnames`	names of the variables in the long format that corresponds to multiple variables in the wide format
`idvar`	names of one or more variables in the long format that identify multiple records form the same group/individual
`label`	`NULL` or a character vector which contains the levels of the vnames

Details

Value

The restructured data frame.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

`reshape, ToWide`

Examples

```
ToLong(data = Gerua, vnames = c('DFF', 'PLHT', 'PNCLE',
       'GRNYLD'), timevar = c('REP'), idvar = c('TPL', 'CULT'))
```

ToWide *Restructure Data Frame to Wide Format*

Description

This function restructures a data frame from long format to wide format. This functions can be used when there are groups of related rows in the data and the user want to appear in groups of columns in the new data file.

Usage

```
ToWide(data, vnames, timevar, idvar)
```

Arguments

`data`	a data frame or name of the data frame
`vnames`	name of one or more variables in the long format that which is to be divided into two or more columns in the wide format
`timevar`	the variable in long format that differentiates the multiple records from the same group or individual; name of one or more variables in the long format which will determine the groupings of the values of the variable(s) to be reshape
`idvar`	names of one or more variables in the long format that will be retained to describe the individual cases or observations in the wide format

Details

Value

The restructured data frame.

Author(s)

Alaine A. Gulles

Reference(s)

See Also

`reshape, ToLong`

Examples

```
ToWide(data = Gerua, vnames = c('DFF', 'PLHT', 'PNCLE',
       'GRNYLD'), timevar = c('REP'), idvar = c('TPL', 'CULT'))
```

ucss *Uncorrected Sum of Squares*

Description

Compute the uncorrected sum of squares.

Usage

```
ucss(x, na.rm = TRUE)
```

Arguments

`x` a numeric vector or data frame

`na.rm` a logical value indicating whether `NA` values should be remove before computation proceeds

Details

Value

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

Description

Usage

Arguments

Details

Value

Author(s)

Alaine A. Gulles

Reference(s)

See Also

Examples

DATA SET

Gerua *Gerua*

Description

The `Gerua` data frame has 156 rows and 7 variables.

Formats

This data frame contains the following columns:

REP	a factor with 3 levels (1 to 3) denoting the replicates
TPL	a factor with 2 levels (D and N)
CULT	a factor with 26 levels
DFF	a numeric vector of number of days after flowering
PLHT	a numeric vector of plant height
PNCLE	a numeric vector of number of panicle
GRNYLD	a numeric vector of grain yield

Details

Source

Examples

Insecticides

Description

The `Insecticides` data frame has 28 rows and 3 variables.

Formats

This data frame contains the following columns:

Treatment a factor with 7 levels (Azodin, DDT + γ-BHC, Dimercon-Boom, Dimercon-Knap, Dol-Mix (1 kg), Dol-Mix (2 kg) and Control) denoting the Different Foliar and Granular Insecticides

Rep a factor with 4 levels (1 to 4) denoting the replicates

GrainYield a numeric vector of grain yield in kg/ha

Details

Grain yield of rice resulting from use of different foliar and granular insecticides for the control of brown planthoppers and stem borers from a Completely Randomized Design Experiment with Four Replicates and 7 treatments.

Source

Gomez, Kwanchai A. and Arturo A. Gomez. (1984) *Statistical Procedures for Agricultural Research 2nd Edition*.

Examples

Maize *Maize*

Description

The `Maize` data frame has 16 rows and 4 variables.

Formats

This data frame contains the following columns:

Maize a factor with 4 levels denoting the Maize Hybrids (A, B, and D) and a Check Variety (C)

Row a factor with 4 levels (1 to 4) denoting the row blocking factor

Column a factor with 4 levels (1 to 4) denoting the column blocking factor

GrainYield a numeric vector of grain yield in kg/ha

Details

Grain yield of three promising maize hybrids and a check variety from an experiment with Latin Square design.

Source

Gomez, Kwanchai A. and Arturo A. Gomez. (1984) *Statistical Procedures for Agricultural Research 2nd Edition*.

Examples

SeedingRate *Seeding Rate*

Description

The `SeedingRate` data frame has 24 rows and 3 variables.

Formats

This data frame contains the following columns:

SeedingRate	a factor with 6 levels (25, 50, 75, 100, 125 and 150) denoting the Seeding Rate (treatment) in kg seed/ha
Rep	a factor with 4 levels (1 to 4) denoting the replicates
GrainYield	a numeric vector of grain yield in kg/ha

Details

Grain Yield of Rice Variety IR8 with Six Different Rates of Seeding, from a Randomized Complete Block Design Experiment with Four Replicates.

Source

Gomez, Kwanchai A. and Arturo A. Gomez. (1984) *Statistical Procedures for Agricultural Research 2nd Edition*.

Examples

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples

DATA SET

Gerua *Gerua*

Description

The `Gerua` data frame has 156 rows and 7 variables.

Formats

This data frame contains the following columns:

REP	a factor with 3 levels (1 to 3) denoting the replicates
TPL	a factor with 2 levels (D and N)
CULT	a factor with 26 levels
DFF	a numeric vector of number of days after flowering
PLHT	a numeric vector of plant height
PNCLE	a numeric vector of number of panicle
GRNYLD	a numeric vector of grain yield

Details

Source

Examples

Insecticides

Description

The `Insecticides` data frame has 28 rows and 3 variables.

Formats

This data frame contains the following columns:

Treatment	a factor with 7 levels (Azodin, DDT + γ-BHC, Dimercon-Boom, Dimercon-Knap, Dol-Mix (1 kg), Dol-Mix (2 kg) and Control) denoting the Different Foliar and Granular Insecticides

Rep	a factor with 4 levels (1 to 4) denoting the replicates
GrainYield	a numeric vector of grain yield in kg/ha

Details

Grain yield of rice resulting from use of different foliar and granular insecticides for the control of brown planthoppers and stem borers from a Completely Randomized Design Experiment with Four Replicates and 7 treatments.

Source

Gomez, Kwanchai A. and Arturo A. Gomez. (1984) *Statistical Procedures for Agricultural Research 2nd Edition*.

Examples

Maize *Maize*

Description

The `Maize` data frame has 16 rows and 4 variables.

Formats

This data frame contains the following columns:

Maize	a factor with 4 levels denoting the Maize Hybrids (A, B, and D) and a Check Variety (C)
Row	a factor with 4 levels (1 to 4) denoting the row blocking factor
Column	a factor with 4 levels (1 to 4) denoting the column blocking factor
GrainYield	a numeric vector of grain yield in kg/ha

Details

Grain yield of three promising maize hybrids and a check variety from an experiment with Latin Square design.

Source

Gomez, Kwanchai A. and Arturo A. Gomez. (1984) *Statistical Procedures for Agricultural Research 2nd Edition*.

Examples

SeedingRate *Seeding Rate*

Description

The `SeedingRate` data frame has 24 rows and 3 variables.

Formats

This data frame contains the following columns:

SeedingRate	a factor with 6 levels (25, 50, 75, 100, 125 and 150) denoting the Seeding Rate (treatment) in kg seed/ha
Rep	a factor with 4 levels (1 to 4) denoting the replicates
GrainYield	a numeric vector of grain yield in kg/ha

Details

Grain Yield of Rice Variety IR8 with Six Different Rates of Seeding, from a Randomized Complete Block Design Experiment with Four Replicates.

Source

Gomez, Kwanchai A. and Arturo A. Gomez. (1984) *Statistical Procedures for Agricultural Research 2nd Edition*.

Examples

Description

Usage

Arguments

Details

Value

Author(s)

Reference(s)

See Also

Examples