Package 'infoDecompuTE'

July 22, 2025

Title Information Decomposition of Two-Phase Experiments

Version 0.6.2

Date 2020-03-28

Description The main purpose of this package is to generate the structure of the analysis of variance (ANOVA) table of the two-phase experiments. The user only need to input the design and the relationships of the random and fixed factors using the Wilkinson-Rogers' syntax, this package can then quickly generate the structure of the ANOVA table with the coefficients of the variance components for the expected mean squares. Thus, the balanced incomplete block design and provides the efficiency factors of the fixed effects can also be studied and compared much easily.

Depends R (>= 3.0.0)

Imports MASS

License GPL (>= 3)

Encoding UTF-8

LazyLoad yes

LazyData true

URL https://github.com/kcha193/infoDecompuTE

BugReports https://github.com/kcha193/infoDecompuTE/issues

RoxygenNote 7.1.0

NeedsCompilation no

Author Kevin Chang [aut, cre],
Katya Ruggiero [aut]

Maintainer Kevin Chang <k.chang@auckland.ac.nz>

Repository CRAN

Date/Publication 2020-03-28 09:10:02 UTC

Contents

`infoDecompuTE-package` *Information Decomposition of Two-phase Experiments*

Description

he main purpose of this package is to generate the structure of the analysis of variance (ANOVA) table of the two-phase experiments. The user only need to input the design and the relationships of the random and fixed factors using the Wilkinson-Rogers' syntax, this package can then quickly generate the structure of the ANOVA table with the coefficients of the variance components for the expected mean squares. Thus, the balanced incomplete block design and provides the efficiency factors of the fixed effects can also be studied and compared much easily.

Details

Package:	infoDecompuTE
Type:	Package
Version:	0.6.1
Date:	2018-05-28
License:	GPL (>= 3)
LazyLoad:	yes

Author(s)

Kevin Chang and Katya Ruggiero

Maintainer: Kevin Chang <k.chang@auckland.ac.nz>

`adjustEffectNames` *Adjust the Effects' Names*

Description

Adjust for appropriate syntax describing the effects matching the structural formula.

Usage

```
adjustEffectNames(effectsMatrix, effectNames)
```

Arguments

`effectsMatrix`	a matrix of variables by terms showing which variables appear in which terms generated by the `terms`.
`effectNames`	a vector of character containing the labels of the treatment or block terms in the model generated by the `terms`.

Value

A vector of character containing the labels of the terms in the model with appropriate syntax describing the effects.

Author(s)

Kevin Chang

Examples

```
str.for = "A*(B/E/C)*D"
effectsMatrix= attr(terms(as.formula(paste("~", str.for)), keep.order = TRUE) , "factors")
effectNames =  attr(terms(as.formula(paste("~", str.for)), keep.order = TRUE) , "term.labels")

adjustEffectNames(effectsMatrix, effectNames)
```

`adjustMissingLevels` *Adjust the Missing Levels*

Description

Adjust for appropriate syntax describing the effects matching the structural formula.

Usage

```
adjustMissingLevels(design.df, str.for)
```

Arguments

`design.df`	a data frame containing the experimental design. Requires every column be a `factor`.
`str.for`	a single string of characters containing the structural formula using the Wilkinson-Rogers' syntax.

Value

A list containing a data frame with the experimental design and a single string of characters containing the structural formula.

Author(s)

Kevin Chang

Examples

```
design.df = data.frame( Blk = factor(1:16),
                        Ani = factor(c( 1,1,2,2,
                                        1,1,2,2,
                                        1,1,2,2,
                                        1,1,2,2)),
                        Trt = factor(c( 1,2,3,4,
                                        1,2,3,4,
                                        1,2,3,4,
                                       1,2,3,4)), stringsAsFactors = TRUE )

adjustMissingLevels(design.df, str.for = "Ani/Trt")
```

chrisEx1 *Randomised Block design consisted of 6 blocks and 3 plots.*

Description

Randomised Block design consisted of 6 blocks and 3 plots.

Usage

```
chrisEx1
```

Format

A data frame with 18 rows and 5 variables:

Blocks Block factor containing 6 levels

Plots Plot factor containing 3 levels

A Treatment factor A containing 3 levels

B Treatment factor B containing 3 levels

C Treatment factor C containing 9 levels

Examples

```
data(chrisEx1)

summaryAovOnePhase(chrisEx1, "Blocks/Plots", "A*B*C")
```

chrisEx2 *Randomised Block design consisted of 8 blocks and 2 plots.*

Description

Randomised Block design consisted of 8 blocks and 2 plots.

Usage

```
chrisEx2
```

Format

A data frame with 18 rows and 5 variables:

Blocks Block factor containing 6 levels

Plots Plot factor containing 3 levels

A Treatment factor A containing 2 levels

B Treatment factor B containing 2 levels

C Treatment factor C containing 3 levels

Examples

```
data(chrisEx2)

summaryAovOnePhase(chrisEx2, "Blocks/Plots", "A*B*C")
```

`chrisEx3` *Randomised Block design consisted of 4 blocks and 2 plots.*

Description

Randomised Block design consisted of 4 blocks and 2 plots.

Usage

```
chrisEx3
```

Format

A data frame with 8 rows and 5 variables:

Blocks Block factor containing 4 levels

Plots Plot factor containing 2 levels

A Treatment factor A containing 2 levels

B Treatment factor B containing 2 levels

C Treatment factor C containing 4 levels

Examples

```
data(chrisEx2)

summaryAovOnePhase(chrisEx2, "Blocks/Plots", "A*B*C")
```

`getCoefVC.onePhase` *Get Variance Components' Coefficients and Mean Squares for Single-Phase or Two-Phase Experiments*

Description

Compute the variance components' coefficients and corresponding to random effects in the expected mean squares of ANOVA table in single-phase or two-phase experiments. These coefficients are then inserted to a matrix where the rows correspond to each source of variation and column correspond to DF and every variance component. The mean squares is calculated if the `response` argument is used.

Usage

```
getCoefVC.onePhase(
  Pb,
  design.df,
  v.mat,
  response,
  table.legend,
  decimal,
  digits
)
```

Arguments

Pb	a list of matrices generated by infoDecompMat function.
design.df	a data frame containing the experimental design. Requires every column be a factor.
v.mat	a list of matrix generated by getVMat.onePhase or getVMat.twoPhase.
response	a numeric vector contains the responses from the experiment.
table.legend	a logical allows users to generate a legend for the variance components of the ANOVA table for large designs. Default is FALSE, resulting in the use of original block factor names.
decimal	a logical allows users to display the coefficients as the decimals. Default is FALSE, resulting in the use of fractions.
digits	a integer indicating the number of decimal places. Default is 2, resulting in 2 decimal places.

Details

The main purpose of this function is to combine the matrices presenting every source of variation of the ANOVA table and the variance matrix to compute the coefficients of the variance components.

The complication arise in giving the row names of the matrix for the source of variation in the ANOVA table.

Value

A matrix containing the characters.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani <- as.factor(LETTERS[c(1,2,3,4,
                             5,6,7,8)])
  Trt <- as.factor(letters[c(1,1,1,1,
```

```
                              2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

blk.str <- "Ani"

rT <- terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)
blkTerm <- attr(rT,"term.labels")

Z <- makeBlkDesMat(design1, blkTerm)

trt.str <- "Trt"
fT <- terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm <- attr(fT, "term.labels")
effectsMatrix <- attr(fT, "factor")

T <- makeContrMat(design1, trtTerm, effectsMatrix, contr.vec = NA)

N =  makeOverDesMat(design1, trtTerm)

Replist = getTrtRep(design1, trtTerm)

Rep <- Replist$Rep
trt.Sca <- Replist$Sca

effFactors = lapply(makeOrthProjectors(Z), function(z)
      getEffFactor(z, T, N, Rep, trt.Sca))

#Now construct variance matrices
Pb <- effFactors[sort(1:length(effFactors), decreasing=TRUE)]

v.mat <- getVMat.onePhase(Z.Phase1 = Z, design.df = design.df, var.comp = NA)

getCoefVC.onePhase(Pb = Pb, design.df = design1, v.mat = v.mat, response = NA,
    table.legend = FALSE, decimal = FALSE, digit = 2)
```

`getCoefVC.twoPhase` *Get Variance Components' Coefficients and Mean Squares for Single-Phase or Two-Phase Experiments*

Description

Compute the variance components' coefficients and corresponding to random effects in the expected mean squares of ANOVA table in single-phase or two-phase experiments. These coefficients are then inserted to a matrix where the rows correspond to each source of variation and column correspond to DF and every variance component. The mean squares is calculated if the `response` argument is used.

Usage

```
getCoefVC.twoPhase(
  Pb,
  design.df,
  v.mat,
  response,
  table.legend,
  decimal,
  digits
)
```

Arguments

Pb	a list of matrices generated by infoDecompMat function.
design.df	a data frame containing the experimental design. Requires every column be a factor.
v.mat	a list of matrix generated by getVMat.onePhase or getVMat.twoPhase.
response	a numeric vector contains the responses from the experiment.
table.legend	a logical allows users to generate a legend for the variance components of the ANOVA table for large designs. Default is FALSE, resulting in the use of original block factor names.
decimal	a logical allows users to display the coefficients as the decimals. Default is FALSE, resulting in the use of fractions.
digits	a integer indicating the number of decimal places. Default is 2, resulting in 2 decimal places.

Details

The main purpose of this function is to combine the matrices presenting every source of variation of the ANOVA table and the variance matrix to compute the coefficients of the variance components.

The complication arise in giving the row names of the matrix for the source of variation in the ANOVA table.

Value

A matrix containing the characters.

Author(s)

Kevin Chang

`getEffFactor` *Construct the Matrix from Information Decomposition and Compute the Efficiency Factors of Treatment effects*

Description

Perform the information decomposition for either the block or treatment effects within a single stratum and Compute the Efficiency Factors for every treatment effect within a single stratum.

Usage

```
getEffFactor(z, T, N, Rep, trt.Sca)
```

Arguments

`z`	a matrix containing the orthogonal projector of a stratum generated by `makeOrthProjectors`.
`T`	a list of contrast matrices generated by `makeContrMat`.
`N`	a matrix containing the design matrix generated by `makeOverDesMat`.
`Rep`	a matrix containing the treatment replication number and is generated by `getTrtRep`.
`trt.Sca`	a numeric vector for computing a coefficients of the fixed effect parameter in EMS and is generated by `getTrtRep`.

Details

The main purpose of this function is to construct a list of resultant matrices associated with each source of variation after the information decomposition and to compute the canonical or average efficiency factors for each treatment effects in each stratum of ANOVA table.

The canonical efficiency factors are generated when the user input the treatment contrasts, otherwise the average efficiency factors, which is the harmonic mean of the canonical efficiency factors, are generated.

Value

A list of matrices and numeric vectors containing the efficiency factors of every treatment effect.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
```

```
})

blk.str = "Ani"

rT = terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)
blkTerm = attr(rT,"term.labels")

Z = makeBlkDesMat(design1, blkTerm)

trt.str = "Trt"
fT <- terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm <- attr(fT, "term.labels")
effectsMatrix <- attr(fT, "factor")

T <- makeContrMat(design1, trtTerm, effectsMatrix, contr.vec = NA)

N =  makeOverDesMat(design1, trtTerm)

Replist = getTrtRep(design1, trtTerm)

Rep <- Replist$Rep
trt.Sca <- Replist$Sca

effFactors = lapply(makeOrthProjectors(Z), function(z)
      getEffFactor(z, T, N, Rep, trt.Sca))
```

`getFixedEF.onePhase` *Get the Fixed Components' coefficients and Efficiency Factors of Single-Phase Experiments.*

Description

Calculate coefficients of fixed effects components of EMS and Treatment Efficiency Factors within each stratum in Single-Phase or two-phase experiment.

Usage

```
getFixedEF.onePhase(
  effFactors,
  trt.Sca,
  T,
  Rep,
  table.legend,
  decimal,
  digits,
  list.sep
)
```

Arguments

effFactors	a list of numeric vector generated by getEffFactor function.
trt.Sca	a numeric vector generated by getTrtRep function.
T	a list of matrices generated by makeContrMat function.
Rep	a numeric matrix generated by getTrtRep function.
table.legend	a logical allows users to generate a legend for the variance components of the ANOVA table for large designs. Default is FALSE, resulting in the use of original treatment factor names.
decimal	a logical allows users to display the coefficients as the decimals. Default is FALSE, resulting in the use of fractions.
digits	a integer indicating the number of decimal places. Default is 2, resulting in 2 decimal places.
list.sep	a logical allows users to present the efficiency factors and coefficients of the fixed effects a list of separate matrices.

Details

Constructs a matrix containing the coefficients of the coefficients of fixed effects components of EMS within each stratum. Also calculates and the average efficiency factors of each treatment effect across all strata

Construct a matrix contain the coefficients of the fixed Components and the average efficiency factors of single-phase experiments.

The function uses the efficiency factors generated by getEffFactor to calculated the coefficients of fixed Effects components of EMS and insert the treatment efficiency factor within each stratum.

The complication arise in giving the row names of the matrix for the source of variation in the ANOVA table.

Value

A matrix.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

blk.str <- "Ani"
```

```
rT <- terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)
blkTerm = attr(rT,"term.labels")

Z <- makeBlkDesMat(design1, blkTerm)

trt.str <- "Trt"
fT <- terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm <- attr(fT, "term.labels")
effectsMatrix <- attr(fT, "factor")

T <- makeContrMat(design1, trtTerm, effectsMatrix, contr.vec = NA)

N <- makeOverDesMat(design1, trtTerm)

Replist = getTrtRep(design1, trtTerm)

Rep <- Replist$Rep
trt.Sca <- Replist$Sca

effFactors = lapply(makeOrthProjectors(Z), function(z) getEffFactor(z, T, N, Rep, trt.Sca))


effFactors <- effFactors[sort(1:length(effFactors), decreasing=TRUE)]

getFixedEF.onePhase(effFactors = effFactors, trt.Sca = trt.Sca,  T = T, Rep = Rep,
table.legend = FALSE, decimal = FALSE, digits = 2, list.sep = TRUE)
```

`getFixedEF.twoPhase` *Get the Fixed Components' coefficients and Efficiency Factors of Two-Phase Experiments.*

Description

Calculate coefficients of fixed effects components of EMS and Treatment Efficiency Factors within each stratum in Single-Phase or two-phase experiment.

Usage

```
getFixedEF.twoPhase(
  effFactors,
  trt.Sca,
  T,
  Rep,
  table.legend,
  decimal,
  digits,
  list.sep
)
```

Arguments

effFactors	a list of numeric vector generated by getEffFactor function.
trt.Sca	a numeric vector generated by getTrtRep function.
T	a list of matrices generated by makeContrMat function.
Rep	a numeric matrix generated by getTrtRep function.
table.legend	a logical allows users to generate a legend for the variance components of the ANOVA table for large designs. Default is FALSE, resulting in the use of original treatment factor names.
decimal	a logical allows users to display the coefficients as the decimals. Default is FALSE, resulting in the use of fractions.
digits	a integer indicating the number of decimal places. Default is 2, resulting in 2 decimal places.
list.sep	a logical allows users to present the efficiency factors and coefficients of the fixed effects a list of separate matrices.

Details

Constructs a matrix containing the coefficients of the coefficients of fixed effects components of EMS within each stratum. Also calculates and the average efficiency factors of each treatment effect across all strata

Construct a matrix contain the coefficients of the fixed Components and the average efficiency factors of single-phase experiments.

The function uses the efficiency factors generated by getEffFactor to calculated the coefficients of fixed Effects components of EMS and insert the treatment efficiency factor within each stratum.

The complication arise in giving the row names of the matrix for the source of variation in the ANOVA table.

Value

A matrix.

Author(s)

Kevin Chang

---

getTrtCoef *Get the Treatment Coefficients*

Description

Compute the overall coefficients every treatment term including the interaction.

Usage

```
getTrtCoef(design.df, trtTerm)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor.
trtTerm	a vector of character containing the labels of the treatment terms in the model generated by terms.

Value

The numeric vector.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

trt.str = "Trt"

fT = terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm = attr(fT,"term.labels")
effectsMatrix = attr(fT,"factor")

trt.Coef = getTrtCoef(design1, trtTerm)
```

---

getTrtRep *Calculate the Treatment Replication number*

Description

Calculate the replication number of every treatment term including the interaction. This is used to compute the treatment efficiency factors.

Usage

```
getTrtRep(design.df, trtTerm)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor.
trtTerm	a vector of character containing the labels of the treatment terms in the model generated by the terms.

Value

A list containing two objects. The first object is a matrix called Rep which contains the replication numbers, where the rows correspond to each treatment combination and the columns correspond to the treatment factors, i.e. the replication number with respect to each treatment factor based on the treatment combination. The second object called Sca which is a numeric vector for computing a coefficients of the fixed effect parameter in EMS.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

trt.str = "Trt"

fT = terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm = attr(fT,"term.labels")
effectsMatrix = attr(fT,"factor")

getTrtRep(design1, trtTerm)
```

getVMat.onePhase *Get the Variance Matrices for Single-Phase experiment*

Description

Construct the matrix for each variance components for the single-phase or two-phase experiment.

Usage

```
getVMat.onePhase(Z.Phase1, design.df, var.comp = NA)
```

Arguments

Z.Phase1	a list of block design matrix from makeBlkDesMat function from Phase 1 block structure.
design.df	a data frame containing the experimental design. Requires every column be a factor.
var.comp	a vector of characters containing the variance components of interest this allows the user to specify the variance components to be shown on the ANOVA table. This also allows the user to specify artificial stratum to facilitate decomposition. Default is NA, which uses every random factor as the variance components with the first phase's variance components in appear before the second phase's variance components.

Value

A list of matrices.

Author(s)

Kevin Chang

Examples

```
 design1 <- local({
    Ani = as.factor(LETTERS[c(1,2,3,4,
                              5,6,7,8)])
    Trt = as.factor(letters[c(1,1,1,1,
                              2,2,2,2)])
    data.frame(Ani, Trt, stringsAsFactors = TRUE )
  })

    blk.str = "Ani"

rT = terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)

    blkTerm = attr(rT,"term.labels")
Z = makeBlkDesMat(design1, rev(attr(rT,"term.labels")))
```

```
V = getVMat.onePhase(Z, design1)
```

---

getVMat.twoPhase *Get the Variance Matrices for Two-Phase experiment*

Description

Construct the matrix for each variance components for the single-phase or two-phase experiment.

Usage

```
getVMat.twoPhase(Z.Phase1, Z.Phase2, design.df, var.comp = NA)
```

Arguments

Z.Phase1	a list of block design matrix from makeBlkDesMat function from Phase 1 block structure.
Z.Phase2	a list of block design matrix from makeBlkDesMat function from Phase 2 block structure.
design.df	a data frame containing the experimental design. Requires every column be a factor.
var.comp	a vector of characters containing the variance components of interest this allows the user to specify the variance components to be shown on the ANOVA table. This also allows the user to specify artificial stratum to facilitate decomposition. Default is NA, which uses every random factor as the variance components with the first phase's variance components in appear before the second phase's variance components.

Value

A list of matrices.

Author(s)

Kevin Chang

Examples

```
   design2 <- local({
Run = as.factor(rep(1:4, each = 4))
Ani = as.factor(LETTERS[c(1,2,3,4,
                          5,6,7,8,
                          3,4,1,2,
                          7,8,5,6)])
Tag = as.factor(c(114,115,116,117)[rep(1:4, 4)])
```

```
  Trt = as.factor(letters[c(1,2,1,2,
                            2,1,2,1,
                            1,2,1,2,
                            2,1,2,1)])
  data.frame(Run, Ani, Tag, Trt, stringsAsFactors = TRUE )
})

    blk.str1 = "Ani"
    blk.str2 = "Run"

rT1 = terms(as.formula(paste("~", blk.str1, sep = "")), keep.order = TRUE)
#random terms phase 1
rT2 = terms(as.formula(paste("~", blk.str2, sep = "")), keep.order = TRUE)
#random terms phase 2

blkTerm1 = attr(rT1,"term.labels")
blkTerm2 = attr(rT2,"term.labels")

Z1 = makeBlkDesMat(design2, rev(blkTerm1))
Z2 = makeBlkDesMat(design2, rev(blkTerm2))

V = getVMat.twoPhase(Z1, Z2, design2, var.comp = NA)
```

`identityMat` *Identity Matrix*

Description

Construct an identity matrix.

Usage

```
identityMat(n)
```

Arguments

`n`	a numeric describes the dimension of the identity matrix.

Value

This function returns a matrix with the diagonal elements equal to one and the off-diagonal elements equal to zero.

Author(s)

Kevin

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

See Also

diag

Examples

```
identityMat(10)
```

`infoDecompMat` *Construct the Matrix from Information Decomposition*

Description

Perform the information decomposition for either the block or treatment effects within a single stratum.

Usage

```
infoDecompMat(z, T, N)
```

Arguments

z	a matrix containing the orthogonal projector for a single stratum generated by makeOrthProjectors.
T	a list of contrast matrices generated by makeContrMat.
N	a matrix containing the design matrix generated by makeOverDesMat.

Details

The main purpose of this function is to construct a list of resultant matrices associated with each source of variation after the information decomposition.

This list of matrices are then used to compute the coefficient of the variance components in the expected mean squares.

Value

A list of matrices.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

blk.str = "Ani"

rT = terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)
blkTerm = attr(rT,"term.labels")

Z = makeBlkDesMat(design1, blkTerm)
Pb = makeOrthProjectors(Z)

trt.str = "Trt"
fT <- terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm <- attr(fT, "term.labels")
effectsMatrix <- attr(fT, "factors")

T <- makeContrMat(design1, trtTerm, effectsMatrix, contr.vec = NA)

N =  makeOverDesMat(design1, trtTerm)


infoDecompMat(Pb[[1]], T, N)
```

`invInfMat` *Invert the Information Matrix*

Description

Using the eigenvalue decomposition method to invert the information matrix.

Usage

```
invInfMat(C, N, T)
```

Arguments

`C`	a matrix of block projector for a single stratum.
`N`	a matrix representation the smallest unit of block or treatment effects generated by `makeOverDesMat`.
`T`	a list of contrast matrices from `makeContrMat`.

Value

This function returns a matrix.

Author(s)

Kevin Chang

References

Nelder JA (1965b). "The Analysis of Randomized Experiments with Orthogonal Block Structure. II. Treatment Structure and the General Analysis of Variance." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 283(1393), 163-178.

Examples

```
m <- matrix(rnorm(10), 10, 10)

invInfMat(m, identityMat(10), identityMat(10))
```

J *Identity Matrix Minus Averaging Matrix*

Description

Construct a square matrix which the identity matrix minus the averging matrix.

Usage

```
J(n)
```

Arguments

n	a numeric describes the dimension of the square matrix.

Value

This function return a square matrix which the identity matrix minus the averaging matrix.

Author(s)

Kevin Chang

Examples

```
J(10)
```

K *Averaging Matrix*

Description

Construct a n-by-n averaging matrix.

Usage

```
K(n)
```

Arguments

n	a numeric describes the dimension of the averaging matrix.

Value

This function returns a $n \times n$ square matrix with all elements equal 1/n.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Examples

```
K(10)
```

makeBlkDesMat *Construct Block Design Matrix*

Description

Construct a binary matrix representing the block design. The rows are corresponding to the observations and the columns are corresponding to the blocks.

Usage

```
makeBlkDesMat(design.df, blkTerm)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor.
blkTerm	a vector of character containing the labels of the block terms in the model generated by the terms.

Value

A list of the binary matrices.

Author(s)

Kevin Chang

See Also

terms

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

blk.str = "Ani*Trt"

rT = terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)

blkTerm = attr(rT,"term.labels")
Z = makeBlkDesMat(design1, blkTerm)
```

makeContrMat *Make Contrast Matrix*

Description

Construct a list of contrast matrices for block for treatment effects.

Usage

```
makeContrMat(design.df, effectNames, effectsMatrix, contr.vec)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor.
effectNames	a vector of character containing the labels of the treatment or block terms in the model generated by the terms.
effectsMatrix	a matrix of variables by terms showing which variables appear in which terms generated by the terms.
contr.vec	a list of contrast vectors, this allows the user to specify the contrasts for each treatment or block factor. Note that if this argument is used, it is necessary to specify the contrasts for every treatment or block factor with the same order as effectNames. Default is NA, and the function output the C matrices described by John and Williams (1987).

Details

The main purpose of this function is to compute a list of C matrices described by John and Williams (1987). These C matrices are used for the information decomposition for every treatment effect in every stratum of the experiment.

If the user input their own defined contrasts for each treatment effects. This function will then transform the input contrasts to the C matrices for the treatment effects.

For the two-phase experiments, the same method of information decomposition is used for the block effects of Phase 1 experiment in the stratum defined from the block structure of the Phase 2 experiment. Hence, the C matrices for the block effects of Phase 1 experiment can also be constructed using this function.

Value

A list of contrast matrices.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

trt.str = "Trt"
fT <- terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)  #fixed terms

trtTerm <- attr(fT, "term.labels")
effectsMatrix <- attr(fT, "factor")

T <- makeContrMat(design1, trtTerm, effectsMatrix, contr.vec = NA)

#Fit each treatment contrasts as a vector seperately
Trt1 <- rep(c(1,-1), each = 4)
Trt2 <-  rep(c(1,-1), time = 4)
Trt3 <- Trt1*Trt2

T <- makeContrMat(design1, trtTerm, effectsMatrix,
      contr.vec =list(Trt = list(Trt1 = Trt1, Trt2 = Trt2, Trt3 = Trt3)))
```

`makeOrthProjectors` *Construct Orthogonal Projector Matrices*

Description

Construct a list of orthogonal projector matrices corresponding to all strata of the experiment.

Usage

```
makeOrthProjectors(BlkDesList)
```

Arguments

`BlkDesList`	a list of block design matrices generated by `makeBlkDesMat`.

Details

The strata decomposition is performed within this function. The first step is to convert the list of block design matrices generated by `makeBlkDesMat` to projection matrices using `projMat`. The second step is to use these projection matrices to project the raw data vector from one stratum to next stratum of the experiment; the resulting matrix corresponds to each stratum is the orthogonal projector matrix of the given stratum.

Value

A list containing matrices.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

blk.str = "Ani"

rT = terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)
blkTerm = attr(rT,"term.labels")

Z = makeBlkDesMat(design1, blkTerm)
Pb = makeOrthProjectors(Z)
```

`makeOverDesMat` *Construct the Overall Treatment or Block design Matrix*

Description

Construct the treatment or block matrix of the smallest unit based from the experimental design.

Usage

```
makeOverDesMat(design.df, effectNames)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor.
effectNames	a vector of character containing the labels of the treatment or block terms in the model generated by the terms.

Details

The main purpose this matrix is used in information decomposition. For the factorial experiment, this matrix is typically the treatment design matrix associated with the interaction effects, because the interaction effects are the smallest unit for the treatment effects.

For the two-phase experiments, the same method of information decomposition is used for the block effects of Phase 1 experiment in the stratum defined from the block structure of the Phase 2 experiment. Hence, the block design matrix of the smallest unit for the block effects of Phase 1 experiment can also be constructed using this function.

Value

A matrix where the rows correspond to the observation and columns correspond to the overall combination of the treatment factors or the block factors of the Phase 1 experiment.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

trt.str = "Trt"

fT = terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)

trtTerm = attr(fT,"term.labels")
effectsMatrix = attr(fT,"factor")

makeOverDesMat(design1, trtTerm)
```

projMat *Construct a Projection Matrix*

Description

Compute the projection matrix from a square matrix.

Usage

```
projMat(X)
```

Arguments

X	a square matrix.

Value

A square matrix.

Author(s)

Kevin Chang

Examples

```
m = matrix(1, nrow = 10, ncol = 3)
projMat(m)
```

summaryAovOnePhase *Summarize an Theoretical Analysis of Variance Model of Single-Phase Experiments*

Description

Computes the coefficients of the variance components for the expected mean squares for single-phase experiments. The function accepts a data frame of the experimental design with the structural formulae of the block and treatment factors. Two tables containing the variance components of the random effects and fixed effects are returned.

Usage

```
summaryAovOnePhase(
  design.df,
  blk.str,
  trt.str,
  var.comp = NA,
  trt.contr = NA,
  table.legend = FALSE,
  response = NA,
  latex = FALSE,
  fixed.names = NA,
  decimal = FALSE,
  digits = 2,
  list.sep = TRUE
)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor. Any punctuation or symbol such as dots or parentheses should be avoid for the column names..
blk.str	a single string of characters containing the structural formula for the block factors using the Wilkinson-Rogers' syntax.
trt.str	a single string of characters containing the structural formula for the treatment factors using the Wilkinson-Rogers' syntax.
var.comp	a vector of characters containing the variance components of interest this allows the user to specify the variance components to be shown on the ANOVA table. This also allows the user to specify artificial stratum to facilitate decomposition. Default is NA, which uses every random factor as the variance components from random.terms.
trt.contr	a list of treatment contrast vectors, this allows the user to specify the contrasts for each treatment factor. Note that if this argument is used, it is necessary to specify the contrasts for every treatment factor with the same order as fixed.terms. Default is NA, which uses the C matrix described by John and Williams (1987).
table.legend	a logical allows the users to use the legend for the variance components of the ANOVA table for a large design. Default is FALSE, which uses the original names.
response	a numeric vector contains the responses from the experiment.
latex	a logical allows the users to output the Latex script to Latex table. Once the Latex script is generated, it requires the user to install and load two Latex packages: booktabs and bm to compile the Latex script.
fixed.names	a vector of character allows the users to modify symbols for the fixed effects for the Latex outputs.
decimal	a logical allows users to display the coefficients as the decimals. Default is FALSE, resulting in the use of function fractions.

`digits` a integer indicating the number of decimal places. Default is 2, resulting in 2 decimal places.

`list.sep` a logical allows users to present the efficiency factors and coefficients of the fixed effects a list of separate matrices. Default is `TRUE`.

Value

The values returned depends on the value of the `table.legend` argument. If `table.legend = FALSE`, this function will return a list of two data frames. The first data frame contains the random effects and the second data frame contains the fixed effects. If the `table.legend` argument is `TRUE`, then it will return a list containing two lists. The first list consists of a data frame of random effects and a character string for the legend. The second list consists of a data frame of fixed effects and a character string for the legend. If `response` argument is used, the random effect table will have one extra column with of mean squares computed from the responses from the experiment.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Nelder JA (1965b). "The Analysis of Randomized Experiments with Orthogonal Block Structure. II. Treatment Structure and the General Analysis of Variance." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 283(1393), 163-178.

Wilkinson GN, Rogers CE (1973). "Symbolic Description of Factorial Models for Analysis of Variance." *Applied Statistics*, 22(3), 392-399.

See Also

`terms` for more information on the structural formula.

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE)
})

summaryAovOnePhase(design1, blk.str = "Ani", trt.str = "Trt")

summaryAovOnePhase(design1, blk.str = "Ani", trt.str = "Trt",
latex = TRUE, fixed.names = c("\\tau"))
```

summaryAovTwoPhase *Summarize an Theoretical Analysis of Variance Model of Two-Phase Experiments*

Description

Computes the coefficients of the variance components for the expected mean squares for two-phase experiments. The function accepts a data frame of the experimental design with the structural formulae of the block and treatment factors. Two tables containing the variance components of the random effects and fixed effects are returned.

Usage

```
summaryAovTwoPhase(
  design.df,
  blk.str1,
  blk.str2,
  trt.str,
  var.comp = NA,
  blk.contr = NA,
  trt.contr = NA,
  table.legend = FALSE,
  response = NA,
  latex = FALSE,
  fixed.names = NA,
  decimal = FALSE,
  digits = 2,
  list.sep = TRUE
)
```

Arguments

design.df	a data frame containing the experimental design. Requires every column be a factor. Any punctuation or symbol such as dots or parentheses should be avoid for the column names..
blk.str1	a single string of characters containing the structural formula for the block factors of the first-phase experiment using the Wilkinson-Rogers' syntax.
blk.str2	a single string of characters containing the structural formula for the block factors of the second-phase experiment using the Wilkinson-Rogers' syntax.
trt.str	a single string of characters containing the structural formula for the treatment factors using the Wilkinson-Rogers' syntax.
var.comp	a vector of characters containing the variance components of interest this allows the user to specify the variance components to be shown on the ANOVA table. This also allows the user to specify artificial stratum to facilitate decomposition. Default is NA, which uses every random factor as the variance components with the first phase's variance components in random.terms1 appear before the second phase's variance components in random.terms2.

blk.contr	a list of first-phase block contrast vectors, this allows the user to specify the contrasts for each block factor in the first phase experiment. Note that if this argument is used, it is necessary to specify the contrasts for every treatment factor with the same order as fixed.terms. Default is NA, which uses the C matrix described by John and Williams (1987).
trt.contr	a list of treatment contrast vectors, this allows the user to specify the contrasts for each treatment factor. Note that if this argument is used, it is necessary to specify the contrasts for every treatment factor with the same order as fixed.terms. Default is NA, which uses the C matrix described by John and Williams (1987).
table.legend	a logical allows the users to use the legend for the variance components of the ANOVA table for a large design. Default is FALSE, which uses the original names.
response	a numeric vector contains the responses from the experiment.
latex	a logical allows the users to output the Latex script to Latex table. Once the Latex script is generated, it requires the user to install and load two Latex packages: booktabs and bm to compile the Latex script.
fixed.names	a vector of character allows the users to modify symbols for the fixed effects for the Latex outputs.
decimal	a logical allows users to display the coefficients as the decimals. Default is FALSE, resulting in the use of function fractions.
digits	a integer indicating the number of decimal places. Default is 2, resulting in 2 decimal places.
list.sep	a logical allows users to present the efficiency factors and coefficients of the fixed effects a list of separate matrices. Default is TRUE.

Value

The values returned depends on the value of the table.legend argument. If table.legend = FALSE, this function will return a list of two data frames. The first data frame contains the random effects and the second data frame contains the fixed effects. If the table.legend argument is TRUE, then it will return a list containing two lists. The first list consists of a data frame of random effects and a character string for the legend. The second list consists of a data frame of fixed effects and a character string for the legend. If response argument is used, the random effect table will have one extra column with of mean squares computed from the responses from the experiment.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Nelder JA (1965b). "The Analysis of Randomized Experiments with Orthogonal Block Structure. II. Treatment Structure and the General Analysis of Variance." *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 283(1393), 163-178.

Wilkinson GN, Rogers CE (1973). "Symbolic Description of Factorial Models for Analysis of Variance." *Applied Statistics*, 22(3), 392-399.

See Also

terms for more information on the structural formula.

Examples

```
#Phase 2 experiment
design2 <- local({
  Run = as.factor(rep(1:4, each = 4))
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8,
                            3,4,1,2,
                            7,8,5,6)])
  Sam = as.factor(as.numeric(duplicated(Ani)) + 1)
  Tag = as.factor(c(114,115,116,117)[rep(1:4, 4)])
  Trt = as.factor(c("healthy", "diseased")[c(1,2,1,2,
                            2,1,2,1,
                            1,2,1,2,
                            2,1,2,1)])
  data.frame(Run, Ani, Sam, Tag, Trt, stringsAsFactors = TRUE)
})
design2

summaryAovTwoPhase(design2, blk.str1 = "Ani", blk.str2 = "Run",
trt.str = "Tag + Trt")

#Add the sample into the Phase 1 block structure
summaryAovTwoPhase(design2, blk.str1 = "Ani/Sam", blk.str2 = "Run",
trt.str = "Tag + Trt")


#Assuming there is crossing between the animals and samples
summaryAovTwoPhase(design2, blk.str1 = "Ani*Sam", blk.str2 = "Run",
trt.str = "Tag + Trt")


#Set Artificial stratum
design2$AniSet = as.factor(c(2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1))
design2

summaryAovTwoPhase(design2, blk.str1 =  "Ani/Sam", blk.str2 = "AniSet/Run",
trt.str = "Tag + Trt", var.comp = c("Ani:Sam", "Ani", "Run"))

#Define traetment contrasts
TagA = rep(c(1,1,-1,-1),time = 4)
TagB = rep(c(1,-1,1,-1),time = 4)
TagC = TagA * TagB
Tag = list(TagA = TagA, TagB = TagB, TagC = TagC)
Tag
```

```
Trt = as.numeric(design2$Trt)-1.5
Trt


summaryAovTwoPhase(design2, blk.str1 =  "Ani/Sam", blk.str2 = "Run",
trt.str = "Tag + Trt",
trt.contr = list(Tag = list(TagA = TagA, TagB = TagB, TagC = TagC), Trt = Trt),
table.legend = TRUE)

#Compute MS
set.seed(527)
summaryAovTwoPhase(design2, blk.str1 = "Ani/Sam", blk.str2 = "Run",
trt.str = "Tag + Trt", response = rnorm(16))$ANOVA

#Generate Latex scripts
summaryAovTwoPhase(design2, blk.str1 = "Ani/Sam", blk.str2 = "Run",
trt.str = "Tag + Trt", latex = TRUE, fixed.names = c("\\gamma", "\\tau"))

#Generate Latex scripts with MS
set.seed(527)
summaryAovTwoPhase(design2, blk.str1 = "Ani/Sam", blk.str2 = "Run",
trt.str = "Tag + Trt", response = rnorm(16), latex = TRUE,
fixed.names = c("\\gamma", "\\tau") )
```

`toLatexTable` *Convert the R output to Latex Table*

Description

Print the Latex scripts on the screen for the user to output the table from the Latex output.

Usage

```
toLatexTable(ANOVA, EF, fixed.names)
```

Arguments

`ANOVA`	a matrix containing the coefficients of the variance components in EMS of ANOVA table generated by `getCoefVC.onePhase` or `getCoefVC.twoPhase`.
`EF`	a matrix containing the coefficient of the fixed effects components and the treatment average efficiency factors generated by `getFixedEF.onePhase` or `getFixedEF.onePhase` function.
`fixed.names`	a vector of character allows the users to modify symbols for the fixed effects.

Details

Once the Latex script is generated, it requires the user to install and load two Latex packages: booktabs and bm to compile the Latex script.

Author(s)

Kevin Chang

Examples

```
design1 <- local({
  Ani = as.factor(LETTERS[c(1,2,3,4,
                            5,6,7,8)])
  Trt = as.factor(letters[c(1,1,1,1,
                            2,2,2,2)])
  data.frame(Ani, Trt, stringsAsFactors = TRUE )
})

blk.str <- "Ani"

rT <- terms(as.formula(paste("~", blk.str, sep = "")), keep.order = TRUE)
blkTerm <- attr(rT,"term.labels")

Z <- makeBlkDesMat(design1, blkTerm)


trt.str = "Trt"
fT <- terms(as.formula(paste("~", trt.str, sep = "")), keep.order = TRUE)

trtTerm <- attr(fT, "term.labels")
effectsMatrix <- attr(fT, "factor")

T <- makeContrMat(design1, trtTerm, effectsMatrix, contr.vec = NA)

N <- makeOverDesMat(design1, trtTerm)

Replist = getTrtRep(design1, trtTerm)

Rep <- Replist$Rep
trt.Sca <- Replist$Sca

effFactors = lapply(makeOrthProjectors(Z), function(z)
      getEffFactor(z, T, N, Rep, trt.Sca))

effFactors <- effFactors[sort(1:length(effFactors), decreasing=TRUE)]

v.mat <- getVMat.onePhase(Z.Phase1 = Z, design.df = design.df, var.comp = NA)

ANOVA <- getCoefVC.onePhase(Pb = effFactors, design.df = design1, v.mat = v.mat,
    response = NA, table.legend = FALSE, decimal = FALSE, digits = 2)

EF <- getFixedEF.onePhase(effFactors = effFactors, trt.Sca = trt.Sca,  T = T,
```

```
  Rep = Rep,
table.legend = FALSE, decimal = FALSE, digits = 2, list.sep = FALSE)

toLatexTable(ANOVA = ANOVA, EF = EF, fixed.names = c("\\tau"))
```

---

tr *Trace of the Matrix*

Description

Compute the trace of the square matrix.

Usage

```
tr(X)
```

Arguments

X	a square matrix.

Value

A numeric value.

Author(s)

Kevin

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

See Also

diag

Examples

```
m = matrix(1, nrow = 10, ncol = 10)
tr(m)
```

`unity` *Construct a unity vector*

Description

Construct a vector with all elements unity.

Usage

```
unity(n)
```

Arguments

`n`	a numeric describe the length of vector.

Value

This function returns a n
$times1$ matrix will all elements unity.

Author(s)

Kevin Chang

References

John J, Williams E (1987). *Cyclic and computer generated Designs*. Second edition. Chapman & Hall.

Examples

```
unity(10)
```

Index