Package ‘selection.index’

October 14, 2022

Type Package

Title Analysis of Selection Index in Plant Breeding

Version 1.1.4

Maintainer Zankrut Goyani <zankrut20@gmail.com>

Description The aim of most plant breeding programmes is simultaneous improvement of several characters. An objective method involving simultaneous selection for several attributes then becomes necessary. It has been recognised that most rapid improvements in the economic value is expected from selection applied simultaneously to all the characters which determine the economic value of a plant, and appropriate assigned weights to each character according to their economic importance, heritability and correlations between characters. So the selection for economic value is a complex matter. If the component characters are combined together into an index in such a way that when selection is applied to the index, as if index is the character to be improved, most rapid improvement of economic value is expected. Such an index was first proposed by Smith (1937 <doi:10.1111/j.1469-1809.1936.tb02143.x>) based on the Fisher's (1936 <doi:10.1111/j.1469-1809.1936.tb02137.x>) ``discriminant function'' Dabholkar (1999 <https://books.google.co.in/books?id=mlFtumAXQ0oC&lpg=PA4&ots=Xgxp1qLuxS&dq=elements%20of%20biometrical%20genetics&lr&pg=PP1#v=onepage&q&f=false>). In this package selection index is calculated based on the Smith (1937) selection index method.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

Depends R (>= 2.10)

Imports utils

URL https://github.com/zankrut20/selection.index

BugReports https://github.com/zankrut20/selection.index/issues

Suggests rmarkdown, markdown, knitr, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Zankrut Goyani [aut, cre, cph],
Alok Shrivasatva [aut]

Repository CRAN

Date/Publication 2022-06-13 22:00:02 UTC

R topics documented:

`comb.indices` *Construction of selection indices based on number of character grouping*

Description

Construction of selection indices based on number of character grouping

Usage

```
comb.indices(ncomb, pmat, gmat, wmat, wcol = 1, GAY)
```

Arguments

`ncomb`	Number of Characters/Traits group
`pmat`	Phenotypic Variance-Covariance Matrix
`gmat`	Genotypic Variance-Covariance Matrix
`wmat`	Weight Matrix
`wcol`	Weight column number incase more than one weights, by default its 1
`GAY`	Genetic Advance of comparative Character/Trait i.e. Yield (Optional argument)

Value

Data frame of all possible selection indices

Examples

```
gmat<- gen.varcov(seldata[,3:9], seldata[,2], seldata[,1])
pmat<- phen.varcov(seldata[,3:9], seldata[,2], seldata[,1])
wmat<- weight.mat(weight)
comb.indices(ncomb = 1, pmat = pmat, gmat = gmat, wmat = wmat, wcol = 1, GAY = 1.075)
```

`gen.advance` *Genetic Advance for PRE*

Description

Genetic Advance for PRE

Usage

```
gen.advance(phen_mat, gen_mat, weight_mat)
```

Arguments

`phen_mat`	phenotypic matrix value of desired characters
`gen_mat`	genotypic matrix value of desired characters
`weight_mat`	weight matrix value of desired characters

Value

Genetic advance of character or character combinations

Examples

```
gmat<- gen.varcov(seldata[,3:9], seldata[,2], seldata[,1])
pmat<- phen.varcov(seldata[,3:9], seldata[,2], seldata[,1])
gen.advance(phen_mat = pmat[1,1], gen_mat = gmat[1,1], weight_mat = weight[1,2])
```

`gen.varcov` *Genotypic Variance-Covariance Analysis*

Description

Genotypic Variance-Covariance Analysis

Usage

```
gen.varcov(data, genotypes, replication)
```

Arguments

`data`	traits to be analyzed
`genotypes`	vector containing genotypes/treatments
`replication`	vector containing replication

Value

A Genotypic Variance-Covariance Matrix

Examples

```
gen.varcov(data=seldata[,3:9], genotypes=seldata$treat,replication=seldata$rep)
```

`phen.varcov` *Phenotypic Variance-Covariance Analysis*

Description

Phenotypic Variance-Covariance Analysis

Usage

```
phen.varcov(data, genotypes, replication)
```

Arguments

`data`	traits to be analyzed
`genotypes`	vector containing genotypes/treatments
`replication`	vector containing replication

Value

A Phenotypic Variance-Covariance Matrix

Examples

```
phen.varcov(data=seldata[,3:9], genotypes=seldata$treat,replication=seldata$rep)
```

`rcomb.indices` *Remove trait or trait combination from possible trait combinations of possible Trait combinations*

Description

Remove trait or trait combination from possible trait combinations of possible Trait combinations

Usage

```
rcomb.indices(ncomb, i, pmat, gmat, wmat, wcol = 1, GAY)
```

Arguments

`ncomb`	Number of character combination
`i`	remove trait or trait combination
`pmat`	Phenotypic Variance Covariance Matrix
`gmat`	Genotypic Variance Covariance Matrix
`wmat`	Weight Matrix
`wcol`	Respective weight column number of Weight Matrix
`GAY`	Genetic Advance/Genetic Gain of base selection index

Value

Data frame of possible selection indices with per cent relative efficiency and ranking

Examples

```
gmat<- gen.varcov(seldata[,3:9], seldata[,2], seldata[,1])
pmat<- phen.varcov(seldata[,3:9], seldata[,2], seldata[,1])
rcomb.indices(ncomb = 2, i = 1, pmat = pmat, gmat = gmat, wmat = weight[,2:3], wcol = 1)
```

`seldata` *Selection Index DataSet*

Description

A dataset containing the data of three replications and 48 progenies with 7 different traits.

Usage

```
data(seldata)
```

Format

A data frame with 144 rows and 9 columns

Details

- rep. Replications
- treat. Treatments/Genotypes
- sypp. Seed Yield per Plant
- dtf. Days to 50
- rpp. Racemes per Plant
- ppr. Pods per Raceme
- ppp. Pods per Plant
- spp. Seeds per Pod
- pw. Pods Weight

`weight` *Weight dataset*

Description

A dataset containing the data of 2 different weights namely euqal weight and broad sense heritability

Usage

```
data(weight)
```

Format

A data frame with 7 rows and 3 columns

Details

- EW. Equal Weight
- h2. Broad Sense Heritability

`weight.mat` *Convert dataframe to matrix*

Description

Convert dataframe to matrix

Usage

```
weight.mat(data)
```

Arguments

`data`	dataframe of weight

Value

A matrix

Examples

```
weight.mat(data = weight)
```

Index