Package 'rYWAASB'

September 30, 2024

Type Package

Title Simultaneous Selection by Trait and WAASB Index

Version 0.2

Date 2024-09-23.

Maintainer Ali Arminian <abeyran@gmail.com>

Description This tool proposes a new ranking algorithm that utilizes a ``Y*WAASB'' biplot generated by the 'metan'. The aim of the current package is to effectively distinguish the top-ranked genotypes in MET (Multi-Environmental Trials). For a detailed explanation of the process of obtaining ``WAASB'', ``WAASBY'' indices, and a ``Y*WAASB'' biplot, refer to the manual included in this package as well as the study by Olivoto & Lúcio (2020) <doi:10.1111/2041-210X.13384>. In this context, ``WAASB'' refers to the ``Weighted Average of Absolute Scores'' provided by Olivoto et al. (2019) <doi:10.2134/agronj2019.03.0220>, which quantifies the stability of genotypes across different environments using linear mixed-effect models. To run the package, you need to extract the ``WAASB'' and ``WAASBY'' coefficients using the 'metan' and apply them. This tool utilizes PCA (Principal Component Analysis) and differentiates the entries which may be genotypes, hybrids, varieties, etc using ``WAASB'', ``WAASBY'', and a combination of the specified trait and WAASB index.

License GPL-3

URL https://github.com/abeyran/rYWAASB

BugReports https://github.com/abeyran/rYWAASB/issues

Depends R (>= 3.5)

Imports ggplot2, graphics, factoextra, FactoMineR, lifecycle, mathjaxr

Suggests car, metan, devtools, usethis, testthat, knitr, rmarkdown, roxygen2, spelling

VignetteBuilder knitr

RdMacros mathjaxr

Encoding UTF-8

RoxygenNote 7.3.2

Language en-US

LazyData true

LazyLoad true

NeedsCompilation no

Author Ali Arminian [aut, cre, cph] (<https://orcid.org/0000-0003-4749-6085>)

Repository CRAN

Date/Publication 2024-09-30 07:10:08 UTC

Contents

`rYWAASB-package` *Simultaneous Selection by Trait and WAASB Index*

Description

rYWAASB performs a new ranking algorithm based on a "Y*WAASB" biplot generated by the 'metan' package which is used in MET(Multi-Environmental Trials). This package effectively distinguishes the top-ranked genotypes based on a given trait (e.g. grain yield or any other trait in agricultural experiments) and the "WAASB" index in the Genotype-by-environment interaction effect studies. Note: Fortunately, this package can impute missing observations and computes them, eliminating any concerns about their presence in the data set. A complete guide may be found at: https://github.com/abeyran/rYWAASB/issues

Author(s)

Ali Arminian abeyran@gmail.com

See Also

Useful links:

- https://github.com/abeyran/rYWAASB
- Report bugs at https://github.com/abeyran/rYWAASB/issues

bar_plot1 *The first barplot of the ranks of genotypes*

Description

[Stable]

- `bar_plot1()` creates a bar plot for the new index (rYWAASB for individuals) for simultaneous selection of genotypes by trait and WAASB index using ggplot2.

Usage

```
bar_plot1(datap)
```

Arguments

`datap`	The data set

Value

Returns an object of class `gg, ggmatrix`.

Author(s)

Ali Arminian abeyran@gmail.com

References

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Examples

```
data(maize)
bar_plot1(maize)
```

bar_plot2 *The second barplot of the ranks of genotypes*

Description

[Stable]

- `bar_plot2()` creates the 2nd barplot of the ranks of genotypes using the `graphics` package.

Usage

```
bar_plot2(datap, verbose = FALSE)
```

Arguments

datap	The data set
verbose	If verbose = TRUE then some results are printed

Value

Returns an object of class gg, graphics

Author(s)

Ali Arminian abeyran@gmail.com

Examples

```
data(maize)
bar_plot2(maize, verbose = FALSE)
```

---

data_ge *Dataset2: a tibble containing ENV, GEN, REP factors and GY(grain yield) and HM agronomic traits from the* metan *package.*

Description

Dataset2: a tibble containing ENV, GEN, REP factors and GY(grain yield) and HM agronomic traits from the metan package.

Usage

```
data(data_ge)
```

Format

A data.frame with 420 rows in 5 columns.

ENV a character vector

GEN a character vector

REP a character vector

GY a numeric vector

HM a numeric vector

References

Olivoto, T., & Lúcio, A.D.C.2020. metan: An R package for multi-environment trial analysis. Methods in Ecology and Evolution , 11(6), 783-789.

Examples

```
library(rYWAASB)
data(data_ge)
```

`maize` *Dataset1: a tibble containing GEN, Trait,* `WAASB` *and* `WAASBY` *indexes.*

Description

Dataset1: a tibble containing GEN, Trait, `WAASB` and `WAASBY` indexes.

Usage

```
data(maize)
```

Format

A `data.frame` with 20 observations (genotypes) within rows and columns including the trait (named as `Y`), `WAASB` and `WAASBY` indexes values.

`GEN` a character vector saved as factor

`Y` a numeric vector

`WAASB` a numeric vector

`WAASBY` a numeric vector

The input format of table of data(NA free), here *maize* data, should be as follows:

GEN	Y	WAASB	WAASBY
Dracma	262.22	0.81	81.6
DKC6630	284.04	2.20	88.5
NS770	243.48	0.33	71.4
...			

Examples

```
library(rYWAASB)
data(maize)
ranki(maize)
bar_plot1(maize)
bar_plot2(maize)
PCA_biplot(maize)
```

nbclust *Data read and estimate the cluster number*

Description

[Experimental]

`nbclust()` Reads and prepares the data, and determine the optimum number of clusters using Average Silhouette Method by `factoextra` package. The average silhouette approach assesses the quality of clustering by evaluating how well each object fits within its cluster. A high average silhouette width signifies effective clustering. This method calculates the average silhouette for different values of k, and the optimal number of clusters (k) is the one that maximizes the average silhouette across a range of potential k values.

Usage

```
nbclust(datap, verbose = FALSE)
```

Arguments

`datap`	The data set
`verbose`	If `verbose = TRUE` then some results are

Details

The `silhouette` coefficient (SC) refers to a criterion to decide number of clusters.It is defined as follows. Though there are numerous methods determining number of clusters such as `the gap statistic` etc.

$$SC = maxK\bar{S}K$$

.

In other words, for each observation i, the `silhouette width` $s(i)$ is defined as follows: Put $a(i)$ = average dissimilarity between i and all other points of the cluster to which i belongs (if i is the only observation in its cluster, $s(i) := 0$ without further calculations). For all other clusters C, put $d(i, C)$ = average dissimilarity of i to all observations of C. The smallest of these $d(i, C)$ is $b(i) = min(C)d(i, C)$, and can be seen as the dissimilarity between i and its "neighbor" cluster, i.e., the nearest one to which it does not belong. Finally,

$$s(i) =:= \frac{b(i) - a(i)}{max(a(i), b(i))}$$

- Note: The clustering methods can be: "average", "centroid", "complete", "mcquitty", "median", "single", "ward.D", "ward.D2" and, Distance methods can be as: "binary", "canberra", "euclidean", "manhattan", "minkowski", "maximum", "pearson", "spearman", "kendall" which may be used in `shipunov` or `factoextra` packages. In this package we just applied `average=UPGMA` and `ward` algorithms.

Value

Returns a data frame

Author(s)

Ali Arminian abeyran@gmail.com

References

Lletı, R., Ortiz, M.C., Sarabia, L.A., Sánchez, M.S. 2004. Selecting variables for k-means cluster analysis by using a genetic algorithm that optimizes the silhouettes, Analytica Chimica Acta, 515(1): 87-100.

Rousseeuw, P.J. (1987) Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math., 20, 53-65.

https://uc-r.github.io/

Examples

```
library(factoextra)

data(maize)
maize <- as.data.frame(maize)
row.names(maize) <- maize[, 1]
maize[, 1] = NULL
GEN <- row.names(maize)
maize <- scale(maize)
nbclust(maize, verbose = FALSE)

# Performing bootstrap or jackknife clustering
# by shipunov package. The examples should be run in the
# console manually due to problems occurs in the ORPHANED
# package `shipunov`.
#
# 1- Bootstrap clustering:
# data.jb <- Jclust(maize,
#   method.d = "euclidean",
#    method.c = "average", n.cl = 2,
#     bootstrap = TRUE)
#
# plot.Jclust(data.jb, top=TRUE, lab.pos=1,
#  lab.offset=1, lab.col=2, lab.font=2)
# Fence(data.jb$hclust, GEN)
#
# data.jb <- Jclust(maize,
#  method.d = "euclidean",
#    method.c = "ward.D", n.cl = 2,
#     bootstrap = TRUE)
#
# plot.Jclust(data.jb, top=TRUE, lab.pos=1,
#            lab.offset=1, lab.col=2, lab.font=2)
# Fence(data.jb$hclust, GEN)
```

```
#
# if(verbose = TRUE):
# cat("\nnumber of iterations:\n", data.jb$iter, "\n")
#
# for "bootstrap":
# data.jb$mat <- as.matrix((data.jb$mat))
# data.jb$mat
# cat("\nmatrix of results:\n", data.jb$mat, "\n")
# cat("clustering info, by eucledean distance measure:\n")
# print(data.jb$hclust)
# cat("groups:\n", data.jb$gr, "\n")
# cat("\nsupport values:\n", data.jb$supp, "\n")
# cat("\nnumber of clusters used:\n", data.jb$n.cl, "\n")

# 2- Jackknife clustering:
# data.jb <- Bclust(maize,
#   method.d = "euclidean", method.c = "average",
#    bootstrap = FALSE)
# plot(data.jb)
#
# data.jb <- Bclust(maize,
#   method.d = "euclidean", method.c = "ward.D",
#    bootstrap = FALSE)
# plot(data.jb)
#
# if(verbose = TRUE):
# For"jackknife":
# cat("Consensus:\n", data.jb$consensus, "\n")
# cat("Vlaues:\n", data.jb$values, "\n")
```

`PCA_biplot` *The PCA biplot with loadings*

Description

[Stable]

- `PCA_biplot()` creates the PCA (Principal Component Analysis) biplot with loadings for the new index `rYWAASB` for simultaneous selection of genotypes by trait and WAASB index. It shows `rYWAASB`, `rWAASB` and `rWAASBY` indices (r: ranked) in a biplot, simultaneously for a better differentiation of genotypes. In PCA biplots controlling the color of variable using their contrib i.e. contributions and cos2 takes place.

Usage

```
PCA_biplot(datap)
```

Arguments

`datap`	The data set

Details

PCA is a machine learning method and dimension reduction technique. It is utilized to simplify large data sets by extracting a smaller set that preserves significant patterns and trends(1). According to Johnson and Wichern (2007), a PCA explains the var-covar structure of a set of variables

$X_1, X_2, ..., X_p$ with a less `linear` combinations of such variables. Moreover the common objective of PCA is 1) data reduction and 2) interpretation.

Biplot and PCA: The biplot is a method used to visually represent both the rows and columns of a data table. It involves approximating the table using a two-dimensional matrix product, with the aim of creating a plane that represents the rows and columns. The techniques used in a biplot typically involve an eigen decomposition, similar to the one used in PCA. It is common for the biplot to be conducted using mean-centered and scaled data(2).

Algebra of PCA: As Johnson and Wichern (2007) stated(3), if the random vector $\mathbf{X}' = X_1, X_2, ..., X_p$ have the covariance matrix $\sum$ with eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$.

Regarding the linear combinations:

$$Y_1 = a_1'X = a_{11}X_1 + a_{12}X_2 + ... + a_{1P}X_p$$

$$Y_2 = a_2'X = a_{21}X_1 + a_{22}X_2 + ... + a_{2p}X_p$$

$$...$$

$$Y_p = a_p'X = a_{p1}X_1 + a_{p2}X_2 + ... + a_{pp}X_p$$

where $Var(Y_i) = \mathbf{a_i'} \sum \mathbf{a_i}$, i = 1, 2, ..., p $Cov(Y_i, Y_k) = \mathbf{a_i'} \sum \mathbf{a_k}$, i, k = 1, 2, ..., p

The principal components refer to the uncorrelated linear combinations $Y_1, Y_2, ..., Y_p$ which aim to have the largest possible variances.

For the random vector $\mathbf{X}' = [X_1, X_2, ..., X_p]$, if $\sum$ be the associated covariance matrix, then $\sum$ have the eigenvalue-eigenvector pairs $(\lambda_1, e_1), (\lambda_2, e_2), ..., (\lambda_p, e_p)$, and as said $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_p \geq 0$.

Then the ith principal component is as follows:

$$Y_i = \mathbf{e_i'X} = e_{i1}X_1 + e_{i2}X_2 + ... + e_{ip}X_p, i = 1, 2, ..., p$$

, where $Var(Y_i) = (e_i' \sum e_i) = \lambda_i, i = 1, 2, ..., p$ $Cov(Y_i, Y_k) = \mathbf{e_i'} \sum \mathbf{e_i} = \mathbf{0}, \mathbf{i} \not\equiv \mathbf{k}$, and: $\sigma_{11} + \sigma_{22} + ... + \sigma_{pp} = \sum_{i=1}^{p} Var(X_i) = \lambda_1 + \lambda_2 + ... + \lambda_p = \sum_{i=1}^{p} Var(Y_i)$.

Interestingly, Total population variance = $\sigma_{11} + \sigma_{22} + ... + \sigma_{pp} = \lambda_1 + \lambda_2 + ... + \lambda_p$.

Another issues that are significant in PCA analysis are:

1. The proportion of total variance due to (explained by) the kth principal component:

$$\frac{\lambda_k}{(\lambda_1 + \lambda_2 + ... + \lambda_p)}, k = 1, 2, ..., p$$

2. The correlation coefficients between the components Y_i and the variables X_k is as follows: $\rho_{Y_i, X_k} = \frac{e_{ik}\sqrt{\lambda_i}}{\sqrt{\sigma_{kk}}}$, i,k = 1, 2, ..., p

Please note that PCA can be performed on `Covariance` or `correlation matrices`. And before PCA the data should be centered, generally.

Value

Returns a a list of dataframes

Author(s)

Ali Arminian abeyran@gmail.com

References

(1) https://builtin.com

(2) https://pca4ds.github.io/biplot-and-pca.html.

(3) Johnson, R.A. and Wichern, D.W. 2007. Applied Multivariate Statistical Analysis. Pearson Prentice Hall. 773 p.

Examples

```
data(maize)
PCA_biplot(maize)
```

---

ranki *The values and ranks of genotypes*

Description

[Stable]

`ranki()` function ranks the genotypes (or entries) based on a new index utilizing the given trait and "WAASB" index to simultaneous select the top-ranked ones. This can be compared with WAASBY index of Olivoto (2019). We suggest users handle the missing data in inputs before considering analyses, due rank codes dose not implement a widespread algorithm to do this task. WAASB(Weighted Average of Absolute Scores), Computes the Weighted Average of Absolute Scores (Olivoto et al., 2019) for quantifying the stability of *g* genotypes conducted in *e* environments using linear mixed-effect models.

Usage

```
ranki(datap)
```

Arguments

`datap`	The data set

Details

According to Olivoto et al. (2019a), WAASB(The weighted average of absolute scores) is computed considering all Interaction Principal Component Axis (IPCA) from the Singular Value Decomposition (SVD) of the matrix of genotype-environment interaction (GEI) effects generated by a linear mixed-effect model, as follows:

$$WAASB_i = \sum_{k=1}^{p} |IPCA_{ik} \times EP_k| / \sum_{k=1}^{p} EP_k$$

where $WAASB_i$ is the weighted average of absolute scores of the *i*th genotype; $IPCA_{ik}$ is the score of the *i*th genotype in the *k*th Interaction Principal Component Axis (IPCA); and EP_k is the explained variance of the *k*th IPCA for *k = 1,2,..,p*, considering $p = min(g - 1; e - 1)$.

Further, $WAASBY_i$ is a superiority or simultaneous selection index allowing weighting between mean performance and stability

$$WAASBY_i = \frac{(rY_i \times \theta_Y) + (rW_i \times \theta_s)}{\theta_Y + \theta_s}$$

, where $WAASBY_i$ is the superiority index for genotype i that weights between mean performance and stability; θ_Y and θ_s are the weights for mean performance and stability, respectively; rY_i and rW_i are the rescaled values for mean performance $\bar{Y}_i$ and stability W_i, respectively of the genotype *i*. For the details of calculations, rescalling and mathematics notations see (Olivoto et al., 2019).

Finally, $rYWAASB_i$ index is the sum of the ranks of the trait (rY_i) and WAASB index ($rWAASB_i$) for each individual:

$$rYWAASB_i = rY_i + rWAASB_i$$

.

The input format of table of data(NA free), here *maize* data, should be as follows:

GEN	Y	WAASB	WAASBY
Dracma	262.22	0.81	81.6
DKC6630	284.04	2.20	88.5
NS770	243.48	0.33	71.4
...			

Value

Returns a data frame showing numerical rankings

Author(s)

Ali Arminian abeyran@gmail.com

References

Olivoto, T., Lúcio, A., DC, da Silva, J.A.G., Sari, B.G. and Diel, M. 2019. Mean performance and stability in multi-environment trials II: Selection based on multiple traits. Agronomy Journal, 111(6):2961-2969.

Olivoto, T., & Lúcio, A.D.C.2020. metan: An R package for multi-environment trial analysis. Methods in Ecology and Evolution, 11(6), 783-789.

Kang, M.S. 1988. "A Rank-Sum Method for Selecting High-Yielding, Stable Corn Genotypes." Cereal Research Communications 16: 113–15.

Examples

```
data(maize)
ranki(maize)
```

Index