

# Package ‘Renaudpack2’

March 2, 2017

**Type** Package

**Title** Un fourre tout de mes fonctions...

**Version** 0.0.0.9003

**Date** 2016-04-22

**Author** Renaud Jaunatre

**Maintainer** Renaud <renaud.jaunatre@irstea.fr>

**Description** Un peu de tout j'ai dit...

**License** GPL-2

**LazyData** TRUE

**RoxygenNote** 6.0.1

**Imports** ade4, vegan, fmsb, lsmeans

## R topics documented:

anovLetters . . . . .	2
barres.plot . . . . .	2
barres.plot.beside . . . . .	3
BBtransf . . . . .	5
better_arrows . . . . .	5
Classes_def . . . . .	6
combin.tab . . . . .	7
combin.tabV0 . . . . .	7
ComStructIndices . . . . .	8
Couleur_continue . . . . .	9
DissRef . . . . .	10
Fonction_hist_double . . . . .	10
Head2head.plot . . . . .	11
IC . . . . .	12
label.cor . . . . .	12
label.corDCA . . . . .	13
label.corNMDS . . . . .	13
MultiDyn . . . . .	14
multivar.polyg . . . . .	15
occ . . . . .	16
radarchart2 . . . . .	16
rar.rm . . . . .	17

raup.calc . . . . .	17
Remplacer . . . . .	18
sem . . . . .	19
Sim_Clust_Ordre . . . . .	19
structure.plot . . . . .	20
structure.plotV2 . . . . .	21
Tableau_recap . . . . .	23
Time.factor.plot . . . . .	23

<b>Index</b>	<b>25</b>
--------------	-----------

---

anovLetters	<i>AnovLetters</i>
-------------	--------------------

---

## Description

Donne les lettres associees a un test post hocc de Tukey effectue apres une anova (implique donc de respecter les conditions d'utilisation des tests parametriques...)

## Usage

```
anovLetters(VAR, FAC, ALPHA = 0.05)
```

## Arguments

VAR	variable
FAC	facteur
ALPHA	(facultatif) donne le seuil de significativit?, 0.05 par default...

## Value

une liste de lettres qui correspondent aux modalites d'un facteur

## Examples

```
biomasse<-c(rnorm(35,20,5),rnorm(35,15,3),rnorm(35,5,3))
traitement<-factor(rep(c("T1","T2","T3"),each=35))
```

---

barres.plot	<i>Diagramme en barre</i>
-------------	---------------------------

---

## Description

Fait un diagramme en barre avec la moyenne et les barres d'erreurs, possibilite de changer le calcul des barres d'erreur, et d'ajouter les lettres des tests post hoc

## Usage

```
barres.plot(variable, Facteur, lettres = c(""), ecart = "sem",
  ylim = "NP", ...)
```

**Arguments**

variable	un vecteur avec les valeurs des variables
Facteur	un facteur donnant les differentes modalites
lettres	(facultatif) liste de type c("text","text") qui figurera au dessus des barres
ecart	(facultatif) type de calcul de la barre d'erreur (sem, IC, sd, var, etc.)
ylim	comme pour la fonction plot(), permet de preciser les limites de l'axe des ordonnees
...	possibilite de rajouter des arguments comme col,main,ylab,etc. associees a barplot

**See Also**

[barplot](#) pour tout les arguments dans ...

**Examples**

```
#Donnees
biomasse<-c(rnorm(35,20,5),rnorm(35,15,3),rnorm(35,5,3))
traitement<-factor(rep(c("T1","T2","T3"),each=35))
#Graphique de base
barres.plot(biomasse,traitement)
#En changeant les barres d'erreurs
barres.plot(biomasse,traitement,ecart=sd)
#En ajoutant les lettres de post-hoc
barres.plot(biomasse,traitement,lettres=c("a","ab","b"))
#En changeant des parametres classiques de barplot...
barres.plot(biomasse,traitement,,lettres=c("a","ab","b"),
col=c("cadetblue","gold","orange"),
xlab="Traitements herbicides",ylab="Biomasse",font.lab=3,
main="Efficacite des traitement")
```

---

barres.plot.beside	<i>barres.plot, mais en parallele</i>
--------------------	---------------------------------------

---

**Description**

Fait un diagramme en barre avec la moyenne et les barres d'erreurs, les barres sont regroupees par modalites d'un premier facteur comme barres.plot avec avec possibilite de changer le calcul des barres d'erreur, et d'ajouter les lettres des tests post hoc, ou des etoiles.

**Usage**

```
barres.plot.beside(VARI, FAC1, FAC2, lettres = c(""), etoiles = c("")),
  ecart = "sem", POSI = "none", ylim = "NP", cex.let = 1, srt.let = 0,
  ...)
```

## Arguments

VARI	une liste de valeurs de la variable (vecteur)
FAC1	une liste des modalités du premier facteur (factor) de même longueur que variable
FAC2	une liste des modalités du deuxième facteur (factor) de même longueur que variable
lettres	(facultatif) liste de type c("text", "text") qui figurera au dessus d'une barre
etoiles	(facultatif) liste de type c("text", "text") qui figurera au dessus d'un groupe de barres
ecart	(facultatif) type de calcul de la barre d'erreur "sem" (par défaut) calculera l'erreur standard (ecart type divisé par n) "IC" calculera l'intervalle de confiance à 5 pourcent (quantile 97.5 à n-1 degré de liberté multiplié par l'écart-type divisé par la racine de n-1) "var" ou "sd" ou toute autre fonction de calcul connue de R Attention, ne supporte pas les NA... (j'ai déjà essayé de modifier mais sans succès...)
POSI	position des labels pour le 2ème facteur, soit en haut "top", soit en bas, "bottom", soit absent "none"
ylim	comme pour la fonction <a href="#">plot</a> , permet de préciser les limites de l'axe des ordonnées
cex.let	facultatif, taille des lettres
srt.let	facultatif, angle des lettres
...	possibilité de rajouter des arguments comme col, main, ylab, etc. associées à <a href="#">barplot</a>

## Exemples

```
# Creation des donnees, avec 2 facteurs, l'age et le sexe, et une variable: taille :
Age=factor(rep(c(2,10,20,30),each=10))
Sexe=factor(rep(rep(c("Homme", "Femme"),5),4))
Taille=c(rnorm(5,60,7),rnorm(5,55,7),rnorm(5,145,15),rnorm(5,129,15),rnorm(5,175,15),
         rnorm(5,165,15),rnorm(5,175,15),rnorm(5,165,15))

# Un premier graphe le plus basique qui soit:
barres.plot.beside(Taille,Sexe,Age)
# et si on inverse les facteurs, avec des couleurs :
barres.plot.beside(Taille,Age,Sexe,col=c("burlywood1","darkseagreen3"))

# avec les annotations qui vont bien (attention, les lettres et etoiles
# sont à définir soit même, C-a-d après tests statistiques, ici pour l'exemple,
# c'est de l'aléatoire...), et avec des couleurs simples (mais moches...)
barres.plot.beside(Taille,Sexe,Age,POSI="bottom",
lettres=c("a","b","c","c","a","b","c","c"),
etoiles=c("*","***"),ylab="Taille (cm)",col=2:5)

# avec des couleurs choisies avec colorRampPalette
coul<-c(colorRampPalette(colors=c("olivedrab1", "grey22"))(4))
barres.plot.beside(Taille,Sexe,Age,POSI="bottom",
lettres=c("a","b","c","c","a","b","c","c"),
etoiles=c("*","***"),col=coul,ylab="Taille (cm)")

# Voici comment mettre une légende
legend("topleft",levels(Age),fill=coul,bty="n")
```

```
# Si on defini des couleurs pour chaque combinaison de modalite, il va falloir
# adapter la legende:
coul2<-c(colorRampPalette(colors=c("lightgoldenrod1", "gold4"))(4),
         colorRampPalette(colors=c("salmon", "firebrick4"))(4))
barres.plot.beside(Taille, Sexe, Age, POSI="bottom",
                  lettres=c("a", "b", "c", "c", "a", "b", "c", "c"),
                  etoiles=c("*", "***"), col=coul2, ylab="Taille (cm)")
legend("topleft", paste(rep(levels(Sexe), each=length(levels(Age))),
                       rep(levels(Age), 2), sep=" "), fill=coul2, bty="n")
```

BBtransf

*transformation de Braun-Blanquet*

### Description

Transforme les coef de Braun-Blanquet en pourcentage (moyenne des classes) Guinochet M. (1973). Phytosociologie. Masson, Paris

### Usage

```
BBtransf(comtab, plus = 0.1)
```

### Arguments

comtab	un tableau de releves de vegetation
plus	la valeurs par lequel le + a ete transcrit... souvent c'est 0.1, 0.2 ou 0.5 (par default ici c'est 0.1)

### Value

un tableau avec les valeurs transformees

better\_arrows

*Trace des fleches suite a une ACP*

### Description

Trace des fleches suite a une ACP

### Usage

```
better_arrows(ANALYSE, coef = 1, length = 0.1, angle = 15, col_text = 1,
...)
```

**Arguments**

ANALYSE	objets issues d'une ACP avec dudi.pca (ade4)
coef	un coef multiplicateur (pour exploser ou non la dispersion dans le plan)
length	longueur de l'encoche des fleches, 0.1 = 10pourcent par défaut
angle	angle des encoches des fleches, 15 = 15deg par défaut
col_text	couleur du texte
...	n'importe quel argument de la fonction arrows

**Exemples**

```
# Creation donnees :
sol<-c(rnorm(10,5,2),rnorm(10,15,3))
veg<-c(rnorm(20,21,5))
amphi<-c(rnorm(5,5,2),rnorm(5,5,3),rnorm(5,5,2),rnorm(5,5,3))
tablo<-data.frame(sol,veg,amphi)
# Faire l'ACP
library(ade4)
ACP<-dudi.pca(tab,scannf = F, nf = 2)
plot(ACP$li)
better_arrows(ACP,coef=2)
```

Classes\_def

*Definition de classes***Description**

Permet de transformer un vecteur d'une variable numerique en classes

**Usage**

```
Classes_def(variab, bornes, noms_classes)
```

**Arguments**

variab	un vecteur a transformer
bornes	les bornes definissant les classes, la borne inferieure de la classe la plus basse ainsi que la borne superieure de la classe la plus haute ne doivent pas etre renseignees
noms_classes	les noms des classes par lequel sera transforme le vecteur variab. Doit avoir une dimensions de length(bornes)+1

**Exemples**

```
variable<-c(0.5, #classe 1
1.2,1.3, #classe 2
2.1, #classe 3
7.2,8.4,9, #classe 5
10.5, #classe 6
22) #et classe 7
```

```
boundaries<-c(1,2,3,7,10,15)
classnames<-paste("classe",c(1:(length(bornes)+1)))
Classes<-Classes_def(variable,boundaries,classnames)
data.frame(Classes,variable)
```

combin.tab

*Combinaison de tableau***Description**

Permet de combiner 2 tableaux (de contingence ou pas), en fusionnant les colonnes ayant le meme nom

**Usage**

```
combin.tab(tableau1, tableau2)
```

**Arguments**

tableau1	le premier tableau, avec les variables en colonnes et les releves en ligne
tableau2	le deuxieme tableau, comme le tableau1

**Details**

Si on veut combiner plus de 2 tableaux, il vaut mieux faire un [write.table](#) et [read.table](#) pour chaque tableau intermediaire (j'ai deja eu quelques bugs...)

Par default, les variables seront classees par ordre alphanbetique, si vous voulez garder les premieres colonnes (quadrat, date, traitement, hauteur de veg) au debut, il vaut mieux utiliser l'ancienne version de la fonction [combin.tabV0](#), qui demande un autre argument *Liste* une liste des noms des colonnes, pour la faire, sous excel, c'est simple, si vous copiez les noms d'especes a la suite (sur une meme colonnes), vous faites supprimer les doublons apres l'avoir mis par ordre alphanbetique, il suffit de rajouter les variables d'information au debut de la liste, et voila, vous avez votre liste en txt!

**Value**

un tableau combine

combin.tabV0

*Version 0 de combin.tab***Description**

Permet de combiner 2 tableaux (de contingence ou pas), en fusionnant les colonnes ayant le meme nom

**Usage**

```
combin.tabV0(Tableau1, Tableau2, Liste)
```

**Arguments**

Liste	une liste des noms des colonnes du tableau final souhaite
tableau1	le premier tableau, avec les variables en colonnes et les releves en ligne
tableau2	le deuxieme tableau, comme le tableau1

**Details**

Si on veut combiner plus de 2 tableaux, il vaut mieux faire un `write.table` et `read.table` pour chaque tableau intermediaire (j'ai deja eu quelques bugs...)

Par default, les variables seront classees par ordre alphabetique, si vous voulez garder les premieres colonnes (quadrat, date, traitement, hauteur de veg) au debut, il vaut mieux utiliser l'ancienne version de la fonction `combin.tabV0`, qui demande un autre argument Liste une liste des noms des colonnes, pour la faire, sous excel, c'est simple, si vous copiez les noms d'especes a la suite (sur une meme colonnes), vous faites supprimer les doublons apres l'avoir mis par ordre alaphabetique, il suffit de rajouter les variables d'information au debut de la liste, et voila, vous avez votre liste en txt!

**Value**

un tableau combine

---

ComStructIndices

*Community Structure Integrity Indices*


---

**Description**

Calculates indices of community integrity compared to a reference community according to Jaunatre et al. (2013).

**Usage**

```
ComStructIndices(REF, ASSESS, rar = 1)
```

**Arguments**

REF	is the reference community data matrix
ASSESS	is the assessed community data matrix
rar	(facultative) Minimum number of samples in which species have to be present to be taken into account in the calculation of indices. Default value is 1. It should not be used in the indices calculation, but it can be useful to reduce the number of species with the <code>structure.plot()</code> function.

**Value**

Comb	A combined community data matrix of reference and assessed communities
Nam_Tot	A list of species names corresponding to the Comb matrix
Nam_Tar	A list of the target species names
REF_Tab	Reference community data matrix (with zero values for species which were absent in the reference community)



ASSESS_Tab	Assessed community data matrix (with zero values for species which were absent in the assessed community)
SumMeanAbREF	Sum of mean abundances
SumAbASSESS	A list of sum of species abundance for each sample
Diff	Matrix of differences, for each species, between mean abundance in the reference samples and abundance in each assessed community samples
SumNeg	Sum of negative differences issued from Diff, i.e. sum of 'higher abundances' in each of assessed community sample
SumPos	Sum of positive differences issued from Diff, i.e. sum of 'missing abundances' in each of assessed community sample
CSII	A list of Community Integrity Index in each assessed community sample
HAI	A list of Higher Abundance Index in each of assessed community sample
CSIIinorm	A list of Normalized Community Integrity Index in each assessed community sample
AbMeanREFOnly	A list of mean abundances of target species in reference samples
ASSESSTarOnly_Tab	An assessed community data matrix with target species only
HigherOnly_Tab	An assessed community data matrix with non-target species only

**See Also**

[structure.plot](#) for graphical output

---

Couleur_continue	<i>Couleur continue</i>
------------------	-------------------------

---

**Description**

Transforme une variable en palette de couleur

**Usage**

```
Couleur_continue(VAR, COLORS = c("Red", "Blue"), minVAR = "def",
  maxVAR = "def", nombre = 100)
```

**Arguments**

VAR	un vecteur
COLORS	une liste de couleur cf l'argument colors de la fonction <a href="#">colorRampPalette</a>
minVAR	valeur minimale theorique (le minimum de VAR par default)
maxVAR	valeur maximale theorique (le maximum de VAR par default)
nombre	precision de la couleur, c-a-d le nombre d'elements composant la palette de couleur (100 par default)

**Value**

un vecteur de couleur de la meme longueur que VAR

**Examples**

```
xx<-c(1:11)
yy<-c(1,3,3.5,4,8,9,3,2.5,7,7.2,NA)
plot(xx,xx,cex=5,col=Couleur_continue(yy),pch=15)
plot(xx,xx,cex=5,col=Couleur_continue(yy,COLORS=c("Orange","Green")),pch=15)
plot(xx,xx,cex=5,col=Couleur_continue(yy,COLORS=c("Orange","Green"),maxVAR=100),pch=15)
```

DissRef

*Dissimilarite a une reference***Description**

Calcul de moyennes d'indices de similarite/dissimilarite entre des releves et plusieurs releves de references

**Usage**

```
DissRef(RELEVES, ETENDUEREL, ETENDUERE, method, binary = T)
```

**Arguments**

RELEVES	tableau de releves especes en colonnes, releves en lignes
ETENDUEREL	etendues des releves sur lesquels le calcul sera fait (exemple 1:9 OU 3:11)
ETENDUERE	etendues des releves de reference (exemple 1:3)
method	methode a utiliser cf l'aide de <a href="#">vegdist</a>
binary	T ou F selon que l'on travaille en presence absence ou non (T par default)

**Value**

Renvoie une liste de valeurs, une par releves chaque valeur correspond a la moyenne des indices qui ont ete calcules entre ce releves et tous les releves de references

**See Also**

[vegdist raup.calc](#)

Fonction\_hist\_double

*Double distribution***Description**

Fonction specifique dans le cadre du stage de Nadege et Anaelle pour superposer deux distributions sur un meme graphe et avec calcul du khi2 automatique

**Usage**

```
Fonction_hist_double(Distribution1, Distribution2, BREAKS, coul1 = 1,
  coul2 = 2, leg1 = "", leg2 = "", ...)
```

**Arguments**

Distribution1	Premiere distribution
Distribution2	Deuxieme distribution
BREAKS	correspond a l'argument breaks de la fonction <code>hist</code> , quelque chose du type <code>seq(50,350,10)</code> fonctionne bien
cou1	la couleur de la distribution 1, afin d'avoir des transparences, c'est pratique d'utiliser la fonction <code>hist</code> de la maniere suivante : <code>rgb(139/255,62/255,47/255,0.5)</code>
cou2	la couleur de la distribution 2
leg1	nom de la distribution 1 (pour la legende)
leg2	nom de la distribution 2 (pour la legende)
...	d'autres argument de la fonction <code>hist</code>

Head2head.plot

*Plot en tete a tete***Description**

Trace des histogrammes en vis en vis avec barres d'erreurs et calculs de Wilcoxon en fonction d'un facteur a 2 modalites

**Usage**

```
Head2head.plot(TABL, FACTEUR, col1 = 2, col2 = 3, cex_noms = 1,
  erreur = "sem", adj_meth = "BH")
```

**Arguments**

TABL	un tableau de donnees avec les variables colonnes
FACTEUR	un vecteur facteur, devant etre factorise, contenant 2 modalites exactement de la meme taille que le nombre de lignes de TABL
col1	la couleur des barres de la modalite 1
col2	la couleur des barres de la modalite 2
cex_noms	taille de la police des noms des variables
erreur	type d'erreur calcule pour la barre d'erreur (erreur standard : "sem", intervalle de confiance : "IC", variance : var)
adj_meth	methode d'ajustement du p calcule : "bond", "BH", "hoch", etc. ou "none"

**Details**

Attention, le premier graphe sera la premiere modalite de ton facteur, par default dans l'ordre alphabetique...

Attention, il peut y avoir des messages d'erreur, c'est lie a des ex aequo dans les valeurs, pour le calcul de wilcoxon, mais bon, ca ne change rien...

---

IC	<i>Intervalle de confiance</i>
----	--------------------------------

---

**Description**

Calcul d'un intervalle de confiance a 5 pour cent

**Usage**

IC(x)

**Arguments**

x                      un vecteur

**Value**

une valeur d'intervalle de confiance

**Examples**

```
A<-rnorm(45,3,2)
sem(A)
```

---

label.cor	<i>Affichage d'especes correlees</i>
-----------	--------------------------------------

---

**Description**

Calcul les correlations des especes avec les axes puis n'affiche que les plus correlees

**Usage**

```
label.cor(ANALYSE, RELEVES, p1 = 0.05, p2 = 0.05, coef = 1, add = F)
```

**Arguments**

ANALYSE	Objet resultant d'une analyse multivariee <a href="#">dudi.pca</a> , <a href="#">dudi.coa</a> , <a href="#">metaMDS</a> ou <a href="#">decorana</a>
RELEVES	les releves ayant servi a faire l'analyse multivariee
p1	a p value minimale de correlation avec l'axe 1 pour etre affichee
p2	la p value minimale de correlation avec l'axe 2 pour etre affichee
coef	un coef multiplicateur (pour exploser ou non la dispersion dans le plan)
add	par default, creer un graphe seul, si add=T, les noms d'especes sont ajoute au graphe deja existant

**Details**

Attention, des erreurs souvent lorsqu'il y a des especes avec 0 occurences!!!! (hein Julie!)

**See Also**

[label.cor](#) apres [dudi.pca](#) ou [dudi.coa](#) ; [label.corNMDS](#) apres une NMDS avec [metaMDS](#) ;  
[label.corDCA](#) apres une DCA avec [decorana](#)

---

label.corDCA	<i>Affichage d'especes correlees</i>
--------------	--------------------------------------

---

**Description**

Calcul les correlations des especes avec les axes puis n'affiche que les plus correlees

**Usage**

```
label.corDCA(ANALYSE, RELEVES, p1 = 0.05, p2 = 0.05, coef = 1, add = F)
```

**Arguments**

ANALYSE	Objet resultant d'une analyse multivariee <a href="#">dudi.pca</a> , <a href="#">dudi.coa</a> , <a href="#">metaMDS</a> ou <a href="#">decorana</a>
RELEVES	les releves ayant servi a faire l'analyse multivariee
p1	a p value minimale de correlation avec l'axe 1 pour etre affichee
p2	la p value minimale de correlation avec l'axe 2 pour etre affichee
coef	un coef multiplicateur (pour exploser ou non la dispersion dans le plan)
add	par default, creer un graphe seul, si add=T, les noms d'especes sont ajoute au graphe deja existant

**Details**

Attention, des erreurs souvent lorsqu'il y a des especes avec 0 occurences!!!! (hein Julie!)

**See Also**

[label.cor](#) apres [dudi.pca](#) ou [dudi.coa](#) ; [label.corNMDS](#) apres une NMDS avec [metaMDS](#) ;  
[label.corDCA](#) apres une DCA avec [decorana](#)

---

label.corNMDS	<i>Affichage d'especes correlees</i>
---------------	--------------------------------------

---

**Description**

Calcul les correlations des especes avec les axes puis n'affiche que les plus correlees

**Usage**

```
label.corNMDS(ANALYSE, RELEVES, p1 = 0.05, p2 = 0.05, coef = 1, add = F)
```

**Arguments**

ANALYSE	Objet resultant d'une analyse multivariee <a href="#">dudi.pca</a> , <a href="#">dudi.coa</a> , <a href="#">metaMDS</a> ou <a href="#">decorana</a>
RELEVES	les releves ayant servi a faire l'analyse multivariee
p1	a p value minimale de correlation avec l'axe 1 pour etre affichee
p2	la p value minimale de correlation avec l'axe 2 pour etre affichee
coef	un coef multiplicateur (pour exploser ou non la dispersion dans le plan)
add	par default, creer un graphe seul, si add=T, les noms d'especes sont ajoute au graphe deja existant

**Details**

Attention, des erreurs souvent lorsqu'il y a des especes avec 0 occurences!!!! (hein Julie!)

**See Also**

[label.cor](#) apres [dudi.pca](#) ou [dudi.coa](#) ; [label.corNMDS](#) apres une NMDS avec [metaMDS](#) ; [label.corDCA](#) apres une DCA avec [decorana](#)

MultiDyn

*Tracer la dynamique en multivariee***Description**

Trace les trajectoires de barycentres de modalites apres une analyse multivariee

**Usage**

```
MultiDyn(MultCOO, FACTime, FACT2, Axis = c(1, 2), LTY = 1, CEX = 1,
  COL = 1, ARR = F, LARR = 0.1, new = "yes", ...)
```

**Arguments**

MultCOO	Les coordonnees de points sortant d'une analyse multivariee, par exemple si apres AFC sous ade4 (nommee AFC_calc) : AFC_calc\$li par exemple si apres NMDS sous vegan (nommee NMDS_calc) : NMDS\$points
FACTime	le facteur temps, chaque modalite correspond un point de la ligne de trajectoire, il faut imperativement les avoir classe dans l'ordre
FACT2	le facteur qui permet de regrouper des points, pour chaque modalite sera representee une trajectoire, le nom de la modalite est affiche au debut de chaque trajectoire
Axis	(facultatif, default: c(1,2)) precise les numero des axes a tracer (correspondant au numero des colonnes de MultCOO)
LTY	(facultatif, default: 1) le format des lignes, cf lty
CEX	(facultatif, default: 1) la taille du texte, cf cex
COL	(facultatif, default: 1) la couleur des lignes et du texte, si une seule couleur est precisee, toutes les trajectoires sont de la meme couleur, si plusieurs couleurs sont precisees, une couleur par trajectoire

ARR	(facultatif, default: F) trace ou non une fleche a la fin de la trajectoire si ARR=T
LARR	(facultatif, default: 0.1) si ARR=T, determine la taille de la fleche
new	si new="yes", trace les fleches sur une nouvelle fenetre (default), sinon, sur le graphe existant
...	d'autres arguments applicables a la fonction <a href="#">plot</a>

---

multivar.polyg	<i>multivar.polyg</i>
----------------	-----------------------

---

## Description

Dessine des polygones autour des points en fonction des modalites d'un facteur

## Usage

```
multivar.polyg(ANAcOO, FAC, pch = 1, col_dot = 1, col_fill = NA,
  col_text = 1, col_bord = 1, cex_lab = 1, dot = "yes", lab = "yes",
  new = "yes", sep = "no", ...)
```

## Arguments

ANAcOO	coordonnees des points d'une analyse multivariees (__\$points pour une NMDS, __\$li pour une AFC, etc.)
FAC	facteur qui permettra de separer les polygones
pch	symbole des points (noir par default)
col_dot	couleur des points (noir par default)
col_fill	couleur du remplissage des polygone (transparent par default)
col_text	couleur des lignes du texte (noir par default)
col_bord	couleur des lignes du polygone (noir par default)
cex_lab	taille des noms de modalites (si lab="yes")
dot	si dot="yes", trace les points de chaque individu, "yes" par default
lab	si lab="yes", affiche le nom de la modalite au niveau du barycentre de ses points, "yes" par default
new	si new="yes", trace les polygones sur une nouvelle fenetre (default), sinon, sur le graphe existant
sep	si sep="yes", trace chaque polygone sur une fenetre graphique differente, "no" par default
...	possibilite de rajouter des arguments comme main,ylab,etc. associees a plot

## Examples

```
library(vegan)
data(varespec)
data(varechem)
facteur_Alu<-factor(ifelse(varechem$Al<50,"Al50",
                           ifelse(varechem$Al<110,"Al110",
                                   ifelse(varechem$Al<230,"Al230","Al++"))))

ord <- metaMDS(varespec)
multivar.polyg(ord$points,facteur_Alu)
```

---

occ	<i>Graphe d'occ</i>
-----	---------------------

---

### Description

Fonction specifique dans le cadre de l'analyse des resumes de REVER... permet de faire un graphe d'occurrence des mots en donnant le vecteur

### Usage

```
occ(VAR, CEX = 0.5, postext = 1, ec = 5, reor = T, x11 = F, ...)
```

### Arguments

VAR	le vecteur avec les mots
CEX	expansion factor des noms des occurrences
postext	la position du nombre de (cf axe des y)
ec	ecart avec la marge <a href="#">par</a> et son parametre oma
reor	si T, dans l'ordre croissant, sinon, dans l'ordre donne
x11	si T, une nouvelle fenetre est ouverte, sinon, non
...	d'autres arguments de la fonction <a href="#">barplot</a>

### Exemples

```
Texte_exemple=c("Banana","Fraise","Chocolat","Chocolat","Chocolat","Fraise","Banana","Banana","Banana","Banana","Banana")
occ(Texte_exemple,main="Exemple parfum de glaces...")
```

---

radarchart2	<i>Diagramme radar avec possibilite de ponderer..</i>
-------------	---

---

### Description

Diagramme radar avec possibilite de ponderer...

### Usage

```
radarchart2(df, axistype = 0, seg = 4, pty = 16, pcol = 1:8,
  plty = 1:6, plwd = 1, pdensity = NULL, pangle = 45, pfcot = NA,
  cglty = 3, cglwd = 1, cglcol = "navy", axislabcol = "blue",
  title = "", maxmin = TRUE, na.itp = TRUE, centerzero = FALSE,
  vlabels = NULL, vlce = NULL, caxislabels = NULL, calcex = NULL,
  paxislabels = NULL, palcex = NULL, coef = 1, ...)
```



**Details**

Fonction en cours de production... pas d'aide pour le moment... Modif de la fonction radarchart du package (fmsb) avec ajout de l'argument coef pour ponderer les variables...

Minato Nakazawa (2014). fmsb: Functions for medical statistics book with some demographic data. R package version 0.5.1.

<http://CRAN.R-project.org/package=fmsb>

neecessite de charger la library fmsb

---

rar.rm	<i>Suppression d'especes rares</i>
--------	------------------------------------

---

**Description**

Supprime les especes n'ayant pas un certain nombre d'occurences

**Usage**

```
rar.rm(tableau.AD, n)
```

**Arguments**

tableau.AD	un tableau de contingence
n	un nombre minimal d'occurence

**Value**

le meme taleau de contingence, mais sans les especes dont le nombre d'occurence est inferieur a n

**Examples**

```
#a construire...
```

---

raup.calc	<i>valeur de similarite/dissimilarite</i>
-----------	---

---

**Description**

calcule un indice de similarite/dissimilarite entre 2 releves

**Usage**

```
raup.calc(tableau, B, A, method, binary = T)
```

**Arguments**

tableau	tableau de releves especes en colonnes, releves en lignes
B	numero du releves dont le numero de ligne est le plus eleve
A	numero du releves dont le numero de ligne est le plus faible
method	ethode a utiliser cf l'aide de <a href="#">vegdist</a>
binary	T ou F selon que l'on travaille en presence absence ou non (T par default)

**Value**

une valeur de similarite/dissimilarite

**See Also**

[vegdist](#)

---

Remplacer	<i>Permet, pour chaque valeur possible d'un vecteur, de les remplacer par des valeurs predefinies</i>
-----------	---

---

**Description**

Permet, pour chaque valeur possible d'un vecteur, de les remplacer par des valeurs predefinies

**Usage**

```
Remplacer(LISTE, VALEURS, VALEURS_NEW)
```

**Arguments**

LISTE	un vecteur
VALEURS	les valeurs possibles de LISTE (possibilite d'utiliser unique(LISTE) ou levels(LISTE))
VALEURS_NEW	les valeurs de remplacements, a donner dans le meme ordre que VALEURS

**Value**

un vecteur ayant la meme longueur que LISTE

**Examples**

```
chiffres<-c(1,1,1,2,4,3,2,1,2,3,3,1,3,2)
valeurs_avant<-c(1,2,3,4)
valeurs_apres<-c("A","B","C","D")

Remplacer(chiffres,valeurs_avant,valeurs_apres)
```

---

sem	<i>sem</i>
-----	------------

---

### Description

Calcul de l'erreur standard

### Usage

```
sem(x)
```

### Arguments

x                      un vecteur

### Details

formule ecart-type/racine du nombre de valeur. Attention, il faudrait rajouter un argument pour utiliser la formule n-1 quand neccessaire...

### Value

une valeur d'erreur standard

### Examples

```
A<-rnorm(45,3,2)
sem(A)
```

---

Sim_Clust_Ordre	<i>Similarite releves-syntaxon (cf these Julie)</i>
-----------------	---

---

### Description

Calcul la similiarite des releves entre un tableau de releves (releves regroupes dans un cluster) et un groupement vegetal de reference (de type alliance vegetale)

### Usage

```
Sim_Clust_Ordre(TAB_FREQ_CLU, NOM_CLU, TAB_ORDRE, METH = "bray", BINA = F)
```

**Arguments**

TAB_FREQ_CLU	tableau de releves des especes en colonnes et des frequences des especes de chaque cluster en ligne (2 colonnes de variables et colonnes especes). Les valeurs correspondent a la frequence des especes pour chaque cluster
NOM_CLU	Nom du cluster choisit
TAB_ORDRE	tableau des releves de references qui definit les differentes alliances vegetales (les especes en colonne et les alliances en ligne) (4 colonnes de variables et des colonnes especes). Les valeurs correspondent a la frequence des especes pour chaque alliance.
METH	methode de similarite a utiliser (par default = bray)
BINA	binarite T ou F selon que l'on travaille en presence/absence ou non (par default = FALSE)

**Value**

renvoie un tableau avec une valeur de similarite (1-indice de Bray Curtis) et un rang (l'alliance la plus similaire) pour chaque alliance en comparaison avec le cluster choisit

**Examples**

```
# Exemple d'application
# chargement du package vegan
library(vegan)
# Creation de tableaux de variables / facteurs
tord<-data.frame(list(nom_ordre = c("Sisymbrietea", "Sisymbrietea", "Sisymbrietea"),
  Ordres = c("0.1.", "0.1.", "0.1."),
  alliances = c("A.1.2.", "A.1.3.", "A.1.4."),
  Nb_syntaxons = c(22,28,32),
  Sp1=c(1,0,0),
  Sp2=c(1,1,0),
  Sp3=c(0,1,1),
  Sp4=c(0,1,0)))

tord
tclu<-data.frame(list(Num_releve = c("3", "4", "6"),All_rel = c("Carriere", "Carriere", "Carriere"),
  Sp1=c(0,0,0.5),
  Sp2=c(1,0,1),
  Sp3=c(1,0,0.1),
  Sp4=c(0.5,0,0),
  Sp5=c(0,1,0),
  Sp6=c(0,1,0)))

tclu
# Utilisation de la fonction pour le cluster 6 (=Num_releve) par exemple
Sim_Clust_Ordre(tclu,NOM_CLU="6",tord,METH="bray",BINA=F)
```

---

structure.plot

---

*structure community plots*


---

**Description**

Performs a barplot of abundances of species in assessed community compared to a REference community

**Usage**

```
structure.plot(INDICE, FACTOR, MULTI = T, MTITLE = "", ABMAX = 5,
  col1 = "grey60", col2 = "white", col3 = "red", col4 = "orange",
  noms = "T", cex_noms = 1, ...)
```

**Arguments**

INDICE	An object issued from <a href="#">ComStructIndices</a> function
FACTOR	A factor list, a barplot of species mean abundances will be performed for each factor level. If no factor is specified, MULTI=F should be specified.
MULTI	If no factor is specified, MULTI=F should be specified
MTITLE	Main title of the plot
ABMAX	Numerical value of the maximum abundance
col1	Colour information for the Reference mean abundances barplot
col2	Colour information for the Reference mean abundances in assessed community barplot, i.e. "missing abundances"
col3	Colour information for the abundances of target species in the assessed community
col4	Colour information for the "higher abundances" in the assessed community
noms	If other than "T", species names are not given
cex_noms	expansion factor for species names
...	other parameters from the <a href="#">barplot</a> function

**See Also**

[ComStructIndices](#)

---

structure.plotV2      *structure plot version 2*

---

**Description**

idem que [structure.plot](#) mais avec quelques arguments en plus (notament barres d'erreur, calcul de difference a la reference, etc.)

**Usage**

```
structure.plotV2(INDICE, FACTOR, MULTI = T, MTITLE = "", ABMAX = 5,
  col1 = "grey60", col2 = "white", col3 = "red", col4 = "orange",
  noms = "T", cex_noms = 1, erreur = "sem", w_err = 1, sp_star = 1,
  adj_meth = "BH", stars = "T", BASE = T, ...)
```

## Arguments

INDICE	An object issued from <a href="#">ComStructIndices</a> function
FACTOR	A factor list, a barplot of species mean abundances will be performed for each factor level. If no factor is specified, MULTI=F should be specified.
MULTI	If no factor is specified, MULTI=F should be specified
MTITLE	Main title of the plot
ABMAX	Numerical value of the maximum abundance
col1	Colour information for the Reference mean abundances barplot
col2	Colour information for the Reference mean abundances in assessed community barplot, i.e. "missing abundances"
col3	Colour information for the abundances of target species in the assessed community
col4	Colour information for the "higher abundances" in the assessed community
noms	If other than "T", species names are not given
cex_noms	expansion factor for species names
erreur	(facultatif) type de calcul de la barre d'erreur "sem" (par défaut) calculera l'erreur standard (ecart type divise par n) "IC" calculera l'intervalle de confiance a 5 pourcent (quantile 97.5 a n-1 degre de liberte multiplie par l'ecart-type divise par la racine de n-1) "var" ou "sd" ou tout autre fonction de calcul connue de R Attention, ne supporte pas les NA... (j'ai deja essaye de modifier mais sans succes...)
w_err	largeur de la barre d'erreur, 1 par défaut Visiblement ca peut encore etre optimise, la taille par défaut est bonne dans certains cas moins dans d'autres...a ajuster donc
sp_star	ecartement des etoiles par rapport aux barres d'erreurs, 1 par défaut
adj_meth	methode d'ajustement du p : "bond", "BH", "hoch", etc. ou "none" (le p correspond a un test de wilcoxon entre la reference et la modalite evaluee)
stars	"T" par défaut, dessine les etoiles, toute autre mention ne les dessinera pas
BASE	T par défaut, dessine le graphe de refe pour chaque modalite, si F, pas de ligne de base
...	all other arguments used by <a href="#">barplot</a>

## Details

Attention, le p est ajuste pour une modalite, en toute rigueur il faudrait sans doute l'ajuster sur l'ensemble des calculs Attention, il peut y avoir des messages d'erreur, c'est lie a des ex aequo dans les valeurs, pour le calcul de wilcoxon, mais bon, ca ne change rien...'

## See Also

[ComStructIndices](#) et [structure.plot](#)

## Examples

```
IndiC=ComStructIndices(relevés[Type=="COUS",],relevés,rar=5)
structure.plotV2(IndiC,Type,ABMAX=5,sp_star=2,w_err=0.1)
```

---

Tableau_recap	<i>Tableau recapitulatif de donnees</i>
---------------	---

---

### Description

Permet de creer automatiquement un tableau de moyenne, erreur

### Usage

```
Tableau_recap(VAR, FAC, ROUND, LETTRES = c(""), ERROR = sem)
```

### Arguments

VAR	le vecteur avec les mots
FAC	un facteur donnant les differentes modalites
ROUND	un nombre de decimal pour l'arrondi des moyennes et erreurs
LETTRES	(facultatif) liste de type c("text","text") qui figurera au dessus des barres
ERROR	(facultatif) type de calcul de la barre d'erreur (sem, IC, sd, var, etc.)

### Exemples

```
#Donnees
biomasse<-c(rnorm(35,20,5),rnorm(35,15,3),rnorm(35,5,3))
traitement<-factor(rep(c("T1","T2","T3"),each=35))
Tableau_recap(biomasse,traitement,2)
#Si on a fait un test post-hoc :
anov<-aov(biomasse~traitement)
shapiro.test(anov$residuals)
summary(anov)
TukeyHSD(anov)
#les lettres sont donc a, b et c :
Tableau_recap(biomasse,traitement,2,c("a","b","c"))
```

---

Time.factor.plot	<i>Time.factor.plot</i>
------------------	-------------------------

---

### Description

Trace une courbe par modalite d'un facteur d'une variable en fonction du temps (moyenne + barres d'erreurs)

### Usage

```
Time.factor.plot(TIME, FACTOR, variable, xlab = "TIME", ylab = "variable",
  pch = c(0), couleur = c(0), lty = c(0), lwd = c(0), xlim = "NP",
  ylim = "NP", Posit_leg = "topleft", linktyp = "b", etoiles = c(""),
  ...)
```

## Arguments

TIME	un vecteur comportant les donnees "temps" ou toute autre variable quantitative qui sera tracee en abscisse
FACTOR	un vecteur comportant les donnees du facteur (il sera transformee en facteur qualitatif)
variable	un vecteur comportant les donnees a tracer
xlab	etiquette des abscisses, par default: TIME
ylab	etiquette des ordonnees, par default: variable
pch	comme pour la fonction <code>plot</code> , permet d'assigner un symbole par modalite du facteur FACTOR
couleur	permet d'assigner une couleur a chaque modalite du facteur FACTOR
lty	permet d'assigner une forme de ligne par modalite du facteur FACTOR
lwd	permet d'assigner une épaisseur de ligne par modalite du facteur FACTOR
xlim	comme pour la fonction <code>plot()</code> , permet de preciser les limites de l'axe des abscisses
ylim	comme pour la fonction <code>plot()</code> , permet de preciser les limites de l'axe des ordonnees
Posit_leg	permet de preciser la position de la legende, soit <code>c(x,y)</code> , soit "topleft", "bottom-right", etc.
linktyp	permet de preciser quel type de lien entre les points... : "p" for points, "l" for lines, "o" for overplotted points and lines, "b", "c")
etoiles	permet de mettre du texte (ou des etoiles) au dessus de chacune des "TIME"
...	autres arguments associes a la fonction <code>plot</code>

## Details

Ne pas faire attention aux messages d'erreur du type suivant, je n'ai pas encore reussi a les retirer...  
 Messages d'avis : "1: In if (couleur == c(0)) rep(1, length(levels(factor(FACTOR)))) else couleur : la condition a une longueur > 1 et seul le premier element est utilise"

## Examples

```
# Exemple d'application :
# Creation des variables / facteurs
tim<-rep(c(1:4),each=6)
fac<-factor(rep(rep(c("A","B"),each=3),4))
vari<-c(rnorm(3,5,2),rnorm(3,3,2),rnorm(3,3,2),rnorm(3,8,2),
        rnorm(3,8,2),rnorm(3,10,2),rnorm(3,16,2),rnorm(3,12,2))

#Utilisation de la fonction :
Time.factor.plot(tim,fac,vari,etoiles=c("*","***","","***"))
```



# Index

anovLetters, 2

barplot, 3, 4, 16, 21, 22  
barres.plot, 2  
barres.plot.beside, 3  
BBtransf, 5  
better\_arrows, 5

Classes\_def, 6  
colorRampPalette, 9  
combin.tab, 7  
combin.tabV0, 7, 7, 8  
ComStructIndices, 8, 21, 22  
Couleur\_continue, 9

decorana, 12–14  
DissRef, 10  
dudi.coa, 12–14  
dudi.pca, 12–14

Fonction\_hist\_double, 10

Head2head.plot, 11  
hist, 11

IC, 12

label.cor, 12, 13, 14  
label.corDCA, 13, 13, 14  
label.corNMDS, 13, 13, 14

metaMDS, 12–14  
MultiDyn, 14  
multivar.polyg, 15

occ, 16

par, 16  
plot, 4, 15, 24

radarchart2, 16  
rar.rm, 17  
raup.calc, 10, 17  
read.table, 7, 8  
Remplacer, 18

sem, 19  
Sim\_Clust\_Ordre, 19  
structure.plot, 9, 20, 21, 22  
structure.plotV2, 21

Tableau\_recap, 23  
Time.factor.plot, 23

vegdist, 10, 18

write.table, 7, 8