

# Package ‘vegetarian’

February 20, 2015

**Type** Package

**Title** Jost Diversity Measures for Community Data

**Version** 1.2

**Date** 2009-07-20

**Author** Noah Charney, Sydne Record

**Maintainer** Noah Charney <noah@alumni.amherst.edu>

**Description** This package computes diversity for community data sets using the methods outlined by Jost (2006, 2007). While there are differing opinions on the ideal way to calculate diversity (e.g. Magurran 2004), this method offers the advantage of providing diversity numbers equivalents, independent alpha and beta diversities, and the ability to incorporate 'order' (q) as a continuous measure of the importance of rare species in the metrics. The functions provided in this package largely correspond with the equations offered by Jost in the cited papers. The package computes alpha diversities, beta diversities, gamma diversities, and similarity indices. Confidence intervals for diversity measures are calculated using a bootstrap method described by Chao et al. (2008). For datasets with many samples (sites, plots), `sim.table` creates tables of all pairwise comparisons possible, and for grouped samples `sim.groups` calculates pairwise combinations of within- and between-group comparisons.

**License** GPL-2

**LazyLoad** yes

**Depends** stats

**Repository** CRAN

**Date/Publication** 2012-10-29 08:59:58

**NeedsCompilation** no

R topics documented:

vegetarian-package . . . . .	2
bootstrap . . . . .	3
d . . . . .	5
H . . . . .	7
M.homog . . . . .	8
normalize.rows . . . . .	10
p.q.sum . . . . .	11
Rel.homog . . . . .	12
sim.groups . . . . .	13
sim.table . . . . .	15
simesants . . . . .	16
similarity . . . . .	18
turnover . . . . .	19
<b>Index</b>	<b>21</b>

---

vegetarian-package	<i>Jost Diversity Measures for Community Data</i>
--------------------	---

---

Description

This package computes diversity with uncertainty estimates for community data sets using the methods outlined by Jost (2006, 2007). While there are differing opinions on the ideal way to calculate diversity (e.g. Magurran 2004), this method offers the advantage of providing diversity 'numbers equivalents,' independent alpha and beta diversities, and the ability to incorporate 'order' (q) as a continuous measure of the importance of rare species in the metrics. The functions provided in this package largely correspond with the equations offered by Jost in the cited papers. The package computes alpha diversities, beta diversities, gamma diversities, and similarity indices. Uncertainty estimates for diversity measures are calculated using a bootstrap method described by Chao et al. (2008). Traditional diversity measures can also be output. For datasets with many samples (sites, plots), `sim.table` creates tables of all pairwise comparisons possible, and for grouped samples `sim.groups` calculates pairwise combinations of within- and between-group comparisons.

Details

Package:	vegetarian
Type:	Package
Version:	1.2
Date:	2009-08-23
License:	GPL-2
LazyLoad:	yes

The core of the vegetarian library is the `d` function, which calculates the basic alpha, beta, and gamma diversity 'numbers equivalents' from community data. `H` uses `d` to calculate the standard

diversity indices. The functions `similarity`, `M.homog`, `Rel.homog`, and `turnover` call `d` to compare diversity across communities. use `sim.table` and/or `sim.groups` to produce multiple pairwise similarity comparisons among many sample sites. All functions can output standard errors by calling `bootstrap` internally. For more detailed bootstrapping outputs, the user can call `bootstrap` separately. The function `normalize.rows` is called by `d` to convert count data into frequencies. The simple function, `p.q.sum` is called internally as core part of the diversity calculations, and is probably of little use to the average user, though it may be used to create more complex diversity measures. This update corrects an error in the `similarity` function and bootstrap standard error estimates from earlier versions.

### Author(s)

Noah Charney, Sydne Record

### References

- Chao, A, L. Jost, S. C. Chiang, Y.-H Jiang, R. L. Chazdon. 2008. A two-stage probabilistic approach to multiple-community similarity indices. *Biometrics* 64: 1178-1186.
- Jost, L. 2006. Entropy and diversity. *Oikos* 113(2): 363-375.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.
- Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology* 54: 427-432.

### Examples

```
data(simesants)
d(simesants[,-1], boot=TRUE)
#remove column with site names
#calculates alpha diversity of entire data-set with standard error
```

---

bootstrap

*Estimates Uncertainties with Bootstrapping*


---

### Description

Diversity measure provided in the `vegetarian` library call `bootstrap` to estimate standard error for the parameter outputs. Bootstrap generates simulated populations from a multinormal distribution centered around the community matrix provided and applies the function of interest to the simulated data. Requires `stats` package.

### Usage

```
bootstrap(abundances, s.sizes = NULL, num.iter = 100, func, func.arg = "blank", sim.pop = FALSE, sim.p
```

**Arguments**

<code>abundances</code>	Community data as a matrix where columns are individual species and rows are sites. Matrix elements are abundance data (e.g. counts, percent cover estimates).
<code>s.sizes</code>	A vector of the total abundances at each site. If no vector is provided, then the scores in the abundance matrix will be treated as counts, and sample size of each site will be computed as the row sum.
<code>num.iter</code>	An integer giving the number of bootstrapping iterations.
<code>func</code>	The function on which bootstrapping will be performed. The function output must be a scalar.
<code>func.arg</code>	A list containing any additional arguments that must be fed to the function specified by <code>func</code> .
<code>sim.pop</code>	A logical specifying whether or not to output the simulated populations for each iteration.
<code>sim.par</code>	A logical specifying whether or not to output the individual parameters calculated on each simulated population.

**Details**

The procedure is outlined in Chao et al. (2008). For each site (row) of the original data set, the total sample size and the relative proportions of each species are used to construct a multinomial distribution. Scores are drawn from this multinomial distribution to generate a simulated population at that site. This is done for each row until an entire simulated community matrix is generated. The supplied function (`func`) is then applied to this simulated matrix to calculate the focal parameter for that iteration. This process of sampling from the multinomial distributions and applying the function is repeated for as many times as is specified by `num.iter`. The standard deviation of the bootstrap parameter values serves as the standard error for the parameter estimate based on the real data.

While this function is called internally by other functions in the *vegetarian* library, only the standard error is reported by those functions. The user may wish to call `bootstrap` directly to access the simulated populations and measure uncertainty in other ways.

**Value**

<code>StdErr</code>	Bootstrap Standard Error for parameter estimate
<code>Simulated.Populations</code>	(optional) Array containing simulated populations
<code>Simulated.Parameters</code>	(optional) Vector containing parameter for each simulated population

**Author(s)**

Noah Charney, Sydne Record

**References**

Chao, A, L. Jost, S. C. Chiang, Y.-H Jiang, R. L. Chazdon. 2008. A two-stage probabilistic approach to multiple-community similarity indices. *Biometrics* 64: 1178-1186.

See Also

[d](#)

Examples

```
data(simesants)
bootstrap(simesants[, -1], func=d)
boot_output<-bootstrap(simesants[, -1], func=d, num.iter=10, sim.pop=TRUE, sim.par=TRUE)
boot_output$Simulated.Parameters
boot_output$Simulated.Populations
simesants_freq<-normalize.rows(simesants[, -1])#not a necessary step, but just to show what you would do if your d
bootstrap(simesants_freq, func=d, num.iter=1000, s.sizes=c(198,186,102,108,187))
```

---

d	<i>'Numbers Equivalents' for Alpha, Beta and Gamma Diversity Indices</i>
---	--

---

Description

The d function calculates the numbers equivalent or effective numbers of species for alpha, beta or gamma diversity as described by Jost (2006, 2007). These are sometimes referred to as 'Hill numbers' (Hill 1973). Jost's interpretation of these number equivalents are more general than Hill's because as the order (q) changes, the sensitivity of the diversity to rare and common species changes. According to Jost (2007), for alpha diversity d calculates Equation 11a for the general case for q and Equation 11b for the Shannon case where q = 1. See Jost (2006) for more on the history and theory behind diversity indices and numbers equivalents.

Usage

```
d(abundances, lev = "alpha", wts = FALSE, q = 1, boot = FALSE, boot.arg = list(s.sizes = NULL, num.iter
```

Arguments

abundances	Community data as a matrix where columns are individual species and rows are sites. Matrix elements are abundance data (e.g. counts, percent cover estimates).
lev	Level of diversity to be calculated. Will accept: 'alpha,' 'beta,' or 'gamma.'
wts	A vector of community weights whose length equals the number of communities being compared. See details below for examples of when community weights might be treated as equal or unequal. Defaults to wts=FALSE where all communities are equally weighted.
q	Order of the diversity measure. Defaults to the Shannon case where q = 1.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty.
boot.arg	(optional) List of arguments to pass bootstrapping function: list(s.sizes=number you specify, num.iter=number you specify)

## Details

Orders higher than one are more sensitive to common species, and orders less than one are more sensitive to rare species. Species richness are diversity measures of  $q = 0$ . Shannon diversity measures are first order or  $q = 1$ . Simpson diversity measures are second order or  $q = 2$ .

Determining how to set community weights depends on the types of comparisons about communities that a researcher wants to make. For instance, Jost (2007) suggests that if species frequencies, but not the actual sizes of the communities are the topic of interest then communities should all be weighed equally. In contrast, community weights are usually unequal when the sizes of the communities has a large influence on the comparisons being made. For instance, when comparing the alpha and beta diversities of a landscape, using unequal community weights makes more sense.

## Value

D.Value	A scalar that gives the numbers equivalent of alpha, beta, or gamma diversity for the specified 'order' (q) and community weights.
StdErr	(optional) Standard error of value estimated through bootstrapping.

## Author(s)

Noah Charney, Sydne Record

## References

Jost, L. 2006. Entropy and diversity. *Oikos* 113(2): 363-375.

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology* 54: 427-432.

## See Also

[bootstrap,H](#)

## Examples

```
data(simesants)
d(simesants[,-1]) #remove column with site names; calculates alpha diversity of entire data-set
hemlock<-subset(simesants,Habitat=="Hemlock")
d(hemlock[-1])#alpha diversity of just hemlock site
d(simesants[,-1],lev="gamma", q=0)#species richness for entire data set
d(simesants[,-1],q=2,boot=TRUE)
d(simesants[,-1],q=2,boot=TRUE,boot.arg=list(num.iter=1000))
simesants_freq<-normalize.rows(simesants[,-1])#not a necessary step, but just to show what you would do if your d
d(simesants_freq,q=2,boot=TRUE,boot.arg=list(s.sizes=c(198,186,102,108,187)))
```

---

H *'Standard Diversity Indices' for Alpha, Beta, and Gamma Diversities*


---

**Description**

Jost describes the conversion of common indices to 'true' or 'standard' diversities (see Table 1 Jost (2006)). The H function calculates the alpha, beta, and gamma 'standard diversity indices.' Specification of different arguments allows for the calculation of species richness, Shannon entropy, Simpson concentration, Gini-Simpson Index, HCDT entropy, and Renyi entropy.

**Usage**

```
H(abundances, lev = "alpha", wts = FALSE, q = 1, HCDT = FALSE, gini = FALSE, boot = FALSE, boot.arg = 1)
```

**Arguments**

abundances	Community data as a matrix where columns are individual species and rows are sites. Matrix elements are abundance data (e.g. counts, percent cover estimates).
lev	Level of diversity to be calculated. Will accept: 'alpha,' 'beta,' or 'gamma.'
wts	A vector of community weights whose length equals the number of communities being compared. See details below for examples of when community weights might be treated as equal or unequal. Defaults to wts=FALSE where all communities are equally weighted.
q	Order of the diversity measure. Defaults to the Shannon case where $q = 1$ .
HCDT	Logical statement. If HCDT = TRUE, then the HCDT or 'Tsallis' entropy is calculated (Keylock 2005, Czachor and Knaudts 2002). Defaults to HCDT = FALSE.
gini	Logical statement. If gini = TRUE and $q = 2$ , then the function calculates the Gini- Simpson index. Defaults to gini = FALSE.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty.
boot.arg	(optional) List of arguments to pass bootstrapping function: list(s.sizes = number you specify, num.iter = number you specify)

**Details**

Orders equal to zero, one, or two calculate species richness, Shannon entropy, and Simpson concentration, respectively. Calculating HCDT entropy requires that  $q$  is not equal to zero, one, or two.

Determining how to set community weights depends on the types of comparisons about communities that a researcher wants to make. For instance, Jost (2007) suggests that if species frequencies, but not the actual sizes of the communities are the topic of interest then communities should all be weighed equally. In contrast, community weights are usually unequal when the sizes of the communities has a large influence on the comparisons being made. For instance, when comparing the alpha and beta diversities of a landscape, using unequal community weights makes more sense.

Value

H.Value	A scalar that gives the 'standard diversity index' of alpha, beta, or gamma diversity for the specified combination of 'order' (q), entropy or index, and community weights.
StdErr	(optional) Standard error of value estimated through bootstrapping.
...	

Author(s)

Noah Charney, Sydne Record

References

Jost, L. 2006. Entropy and diversity. *Oikos* 113(2): 363-375.

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology* 54: 427-432.

See Also

[d bootstrap](#)

Examples

```
data(simesants)
H(simesants[,-1]) #remove column with site names; calculates alpha diversity of entire data-set
hemlock<-subset(simesants,Habitat=="Hemlock")
H(hemlock[-1])#alpha diversity of just hemlock site
H(simesants[,-1],lev="gamma", q=0)#species richness for entire data set
H(simesants[,-1],q=2,boot=TRUE)
H(simesants[,-1],q=2,boot=TRUE,boot.arg=list(num.iter=1000))
simesants_freq<-normalize.rows(simesants[,-1])#not a necessary step, but just to show what you would do if your d
H(simesants_freq,q=2,boot=TRUE,boot.arg=list(s.sizes=c(198,186,102,108,187)))
```

---

M.homog	<i>MacArthur's Homogeneity Measure</i>
---------	--

---

Description

Macarthur's homogeneity measure provides a gauge of the amount of total diversity contained in an average community or sample (MacArthur 1965). It can be derived from a transformation of the true beta diversity of order 1, the numbers equivalent of the beta Shannon entropy (Jost 2007 Equation 18). If the N communities being compared are equally weighted, then other values of q can be specified to calculate other familiar similarity indices (e.g. Jaccard index when q=0, Morisita-Horn index when q=2) (Jost 2006).



**Usage**

```
M.homog(abundances, abundances2 = NULL, q = 1, std = FALSE, boot = FALSE, boot.arg = list(s.sizes = NUL
```

**Arguments**

abundances	Community data as a matrix where columns are individual species and rows are sites or a vector of different species within a site. Matrix and vector elements are abundance data (e.g. counts, percent cover estimates).
abundances2	Community data, a vector of different species within a site. Vector elements are abundance data (e.g. counts, percent cover estimates). If abundances is given a matrix, then abundances2 defaults to NULL.
q	Order of the diversity measure. Defaults to the Shannon case where $q = 1$ .
std	Logical statement. If <code>std = TRUE</code> , then the data is standardized, so that the value returned is bounded between zero and one. The default is <code>std = FALSE</code> where there is no standardization of the data, and lower and upper limits of the value returned are $1/N$ and one, respectively.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty.
boot.arg	(optional) List of arguments to pass bootstrapping function: <code>list(s.sizes=number you specify, num.iter=number you specify)</code>

**Value**

M	A scalar, MacArthur's homogeneity measure, where the lower limit (either $1/N$ or zero depending on the specification of the argument <code>std</code> ) is the case when all communities are distinct and the upper limit (unity) occurs when all communities are exactly identical.
StdErr	(optional) Standard error of value estimated through bootstrapping.

**Author(s)**

Noah Charney, Sydne Record

**References**

- Jost, L. 2006. Entropy and diversity. *Oikos* 113(2): 363-375.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.
- Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology* 54: 427-432.

**See Also**

[Rel.homog bootstrap](#)

## Examples

```
data(simesants)
M.homog(simesants[1:2,-1])
hemlock<-subset(simesants,Habitat=="Hemlock")[, -1]
hardwood<-subset(simesants,Habitat=="Hardwood")[, -1]
M.homog(abundances=hemlock, abundances2=hardwood)
M.homog(simesants[1:2,-1], q=2, std=TRUE, boot=TRUE)
```

---

normalize.rows	<i>Converts absolute abundances to relative proportions of species using row normalization</i>
----------------	--

---

## Description

Generic function that takes a matrix of community data (where species are columns and rows are sites) and converts absolute abundances, in the form of counts or percent cover estimates, to relative proportions of species using row normalization.

## Usage

```
normalize.rows(abundance.matrix)
```

## Arguments

abundance.matrix

Community data as a matrix where columns are individual species and rows are sites. Matrix elements are abundance data (e.g. counts, percent cover estimates).

## Details

All of Jost's calculation for partitioning diversity are based on relative proportions of species in a community dataset (Jost 2006, Jost 2007). The `normalize.rows` function performs this necessary step of converting absolute abundances to relative proportions of species for Jost's subsequent calculations of diversity and similarity indices. The `normalize.rows` function is built in to the `D` function for Jost's calculations, but we provide it here for more general usage. For example, this function will also be useful when researchers are most interested in relative proportions of species within a community rather than absolute comparisons of species abundances across multiple sites.

## Value

A matrix of community data where the elements are relative proportions of species across sites (rows).

## Author(s)

Noah Charney, Sydne Record

## References

- Jost, L. 2006. Entropy and diversity. *Oikos* 113(2): 363-375.
- Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.
- Hill, M. 1973. Diversity and evenness: A unifying notation and its consequences. *Ecology* 54: 427-432.

## See Also

[d](#)

## Examples

```
data(simesants)
simesants_freq<-normalize.rows(simesants[,-1])
```

---

p.q.sum

*Sum of proportional abundance of species*

---

## Description

Nearly all diversity indices are based on a basic sum of the proportional abundances of species:  $\sum (p_i^q)$ , where  $p_i$  is the proportion of species  $i$  and  $q$  is the 'order' of the diversity measure. According to Jost (2007), order is a diversity measure's sensitivity to rare or common species. If referring to Jost 2007, p.q.sum calculates Equation 1. The p.q.sum function is built in to the D.gamma and D.alpha functions for Jost's calculations, but we provide it here for more general usage. For example, if a researcher wanted to build their own diversity index then they could use p.q.sum as a starting point.

## Usage

```
p.q.sum(p, q = 1)
```

## Arguments

- p** A vector or matrix of relative proportions of species.
- q** Order of the diversity measure. Defaults to the Shannon case where  $q = 1$ .

## Details

Orders higher than one are more sensitive to common species, and orders less than one are more sensitive to rare species. Species richness are diversity measures of  $q = 0$ . Shannon diversity measures are first order or  $q = 1$ . Simpson diversity measures are second order or  $q = 2$ .

Value

A scalar that is the sum of proportional abundances of species in the input community data (given as a vector or matrix).

Author(s)

Noah Charney, Sydne Record

References

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. Ecology 88(10): 2427-2439.

See Also

[d](#)

---

Rel.homog	<i>Relative Homogeneity</i>
-----------	-----------------------------

---

Description

Provides a comparison between communities with unequal weights based on the numbers equivalent of the Shannon beta diversity (Jost 2007). When community weights are not equal, the Shannon measures of diversity where  $q=1$ , are still valid and can be used to calculate a relative measure of homogeneity derived from MacArthur’s homogeneity measure (Equation 22 from Jost 2007).

Usage

Rel.homog(abundances, abundances2 = NULL, wts = FALSE, boot = FALSE, boot.arg = list(s.sizes = NULL, n

Arguments

abundances	Community data as a matrix where columns are individual species and rows are sites or a vector of different species within a site. Matrix and vector elements are abundance data (e.g. counts, percent cover estimates).
abundances2	Community data, a vector of different species within a site. Vector elements are abundance data (e.g. counts, percent cover estimates). If abundances is given a matrix, then abundances2 defaults to NULL.
wts	A vector of community weights whose length equals the number of communities being compared. See details below for examples of when community weights might be treated as equal or unequal. Defaults to wts=FALSE where all communities are equally weighted.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty.
boot.arg	(optional) List of arguments to pass bootstrapping function: list(s.sizes=number you specify, num.iter=number you specify)

Details

This measure assumes the Shannon case where the order  $q=1$ .

Value

REL.HOMOG	A scalar between zero (all communities distinct) and one (all communities identical).
StdErr	(optional) Standard error of value estimated through bootstrapping.

Author(s)

Noah Charney, Sydne Record

References

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. Ecology 88(10): 2427-2439.

See Also

[M.homog](#), [similarity](#), [turnover](#).

Examples

```
data(simesants)
Rel.homog(simesants[1:2,-1])
hemlock<-subset(simesants,Habitat=="Hemlock")[, -1]
hardwood<-subset(simesants,Habitat=="Hardwood")[, -1]
Rel.homog(abundances=hemlock,abundances2=hardwood)
Rel.homog(simesants[1:2,-1], boot=TRUE)
```

---

sim.groups	<i>Within- and Between-Group Similarities</i>
------------	---

---

Description

Given two groups with multiple samples (e.g. sites, plots) the sim.groups function calculates all possible pairwise combinations of within- and between-group comparisons.

Usage

```
sim.groups(abundances1, abundances2, q = 1, labels = FALSE, boot = FALSE, boot.arg = list(s.sizes = NU
```

**Arguments**

abundances1	Community data as a matrix where columns are individual species and rows are sites or a vector of different species within a site. Matrix and vector elements are abundance data (e.g. counts, percent cover estimates).
abundances2	Community data, a vector of different species within a site. Vector elements are abundance data (e.g. counts, percent cover estimates). If abundances1 is given a matrix, then abundances2 defaults to a logical FALSE statement.
q	Order of the diversity measure. Defaults to the Shannon case where $q = 1$ .
labels	Logical statement. If labels=TRUE, then site names are given as the first column of the abundance matrix. The default is labels=FALSE where no site names are given in the abundance matrix.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty. If boot=TRUE, only standard errors will be output in table; to get both values and standard error of similarities, must call sim.groups twice, setting boot to both TRUE and FALSE
boot.arg	(optional) List of arguments to pass bootstrapping function: list(s.sizes=number you specify, num.iter=number you specify)

**Details**

Depending on the specification of the order ( $q$ ), other different similarity indices may be calculated (e.g. Sorenson index when  $q=0$ , Horn index when  $q=1$ , Morisita-Horn index when  $q=2$ ) (Jost 2007).

**Value**

within.1	Within-group similarities (or standard errors) for first group.
within.2	Within-group similarities (or standard errors) for second group.
between	Between group similarities (or standard errors).

**Author(s)**

Noah Charney, Sydne Record

**References**

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

**See Also**

[similarity](#), [sim.table](#)

## Examples

```
data(simesants)
sim.groups(simesants[1:3,],simesants[4:5,],labels=TRUE,boot=TRUE)
sim.groups(simesants[1:3,-1],simesants[4:5,-1])
sim.groups(simesants[1:3,-1],simesants[4:5,-1],labels=TRUE,boot=TRUE)#gives standard errors only
sim.groups(simesants[1:3,-1],simesants[4:5,-1],labels=TRUE,boot=TRUE,boot.arg=list(num.iter=200))#gives standard errors
```

---

sim.table	<i>Similarity Summary Table</i>
-----------	---------------------------------

---

## Description

Creates a summary table of community overlap for all possible pairwise combinations of samples (e.g. sites, plots) using the similarity function.

## Usage

```
sim.table(abundances, q = 1, labels = FALSE, half = TRUE, diag = TRUE, boot = FALSE, boot.arg = list(sizes=number you specify, num.iter=number you specify))
```

## Arguments

abundances	Community data as a matrix where columns are individual species and rows are sites. Matrix elements are abundance data (e.g. counts, percent cover estimates).
q	Order of the diversity measure. Defaults to the Shannon case where $q = 1$ .
labels	Logical statement. If labels=TRUE, then site names are given as the first column of the abundance matrix. The default is labels=FALSE where no site names are given in the abundance matrix.
half	Logical statement that changes the display of the resulting matrix. The default is half=TRUE, where the similarities are not repeated and are given as NaN values on one side of the diagonal in the resulting matrix. If half=FALSE, then the displayed similarities are repeated on either side of the diagonal of the resulting matrix.
diag	Logical statement that changes the display of the resulting matrix. The default is diag=TRUE, where the elements in the diagonal of the resulting similarity matrix are all equal to one because a sample should be completely identical to itself. If diag=FALSE, then the elements in the diagonal of the resulting similarity matrix are given NaN values.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty. If boot=TRUE, only standard errors will be output in table; to get both values and standard error of similarities, must call sim.groups twice, setting boot to both TRUE and FALSE
boot.arg	(optional) List of arguments to pass bootstrapping function: list(sizes=number you specify, num.iter=number you specify)

**Details**

This function calculates pairwise similarity for two or more samples, so the abundance data must be a matrix with two or more rows (samples). Depending on the specification of the order (q), other different similarity indices may be calculated (e.g. Sorenson index when  $q=0$ , Horn index when  $q=1$ , Morisita-Horn index when  $q=2$ ) (Jost 2007).

**Value**

A matrix of pairwise similarities showing community overlap calculated using the similarity function.

**Author(s)**

Noah Charney, Sydne Record

**References**

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

**See Also**

[similarity](#), [sim.groups](#)

**Examples**

```
data(simesants)
sim.table(simesants[, -1])
sim.table(simesants, labels=TRUE)
sim.table(simesants, labels=TRUE, diag=FALSE)
sim.table(simesants, labels=TRUE, half=FALSE)
sim.table(simesants, labels=TRUE, boot=TRUE) #standard errors only
sim.table(simesants, labels=TRUE, boot=TRUE, boot.arg=list(num.iter=200), q=2) #standard errors only
```

---

simesants

---

*Harvard Forest Simes Tract Ant Community Data*


---

**Description**

Within the Simes tract of Harvard Forest, we sampled ants by hand collection along 100-m transects at eight sites with the following microhabitats: two sites each of hemlock, white pine, and hardwood overstories; a transition between hardwood swamp and pine; and a rocky-slope with a hardwood overstory.

**Usage**

```
data(simesants)
```



**Format**

A data frame with 5 observations on the following 23 variables.

Habitat a factor with levels Hardwood Hardwood\_Pine\_Swamp Hardwood\_Rocky\_Slope Hemlock  
White\_Pine

Acasub a numeric vector

Aphrud a numeric vector

Camher a numeric vector

Campen a numeric vector

Creceer a numeric vector

Crelin a numeric vector

Dolpla a numeric vector

Dolpus a numeric vector

Forneo a numeric vector

Forsuba a numeric vector

Lasali a numeric vector

Lasnea a numeric vector

Lasspe a numeric vector

Lasumb a numeric vector

Myrpun a numeric vector

Myrscu a numeric vector

Myrsmi a numeric vector

Stebre a numeric vector

Steimp a numeric vector

Stesch a numeric vector

Tapses a numeric vector

Temlon a numeric vector

**Details**

Columns correspond to the ant species, designated by a six letter code combining the first three letters of the genus with the first three letters of the species.

**Source**

<http://harvardforest.fas.harvard.edu/data/archive.html>

**Examples**

```
data(simesants)
```

---

similarity	<i>Similarity Summary Table</i>
------------	---------------------------------

---

**Description**

Creates a summary table of community overlap for all possible pairwise combinations of samples (e.g. sites, plots) using the `similarity` function.

**Usage**

`similarity(abundances, abundances2 = NULL, q = 1, boot = FALSE, boot.arg = list(s.sizes = NULL, num.it`

**Arguments**

<code>abundances</code>	Community data as a matrix where columns are individual species and rows are sites. Matrix elements are abundance data (e.g. counts, percent cover estimates). May be given as a vector with <code>abundances2</code> as a vector for a second community.
<code>abundances2</code>	Community data, a vector of different species within a site. Vector elements are abundance data (e.g. counts, percent cover estimates). If <code>abundances</code> is given a matrix, then <code>abundances2</code> defaults to a logical <code>FALSE</code> statement.
<code>q</code>	Order of the diversity measure. Defaults to the Shannon case where $q = 1$ .
<code>boot</code>	Logical indicating whether to use bootstrapping to estimate uncertainty.
<code>boot.arg</code>	(optional) List of arguments to pass bootstrapping function: <code>list(s.sizes=number you specify, num.iter=number you specify)</code>

**Details**

This function calculates pairwise similarity for two or more samples, so the abundance data must be a matrix with two or more rows (samples). Depending on the specification of the order ( $q$ ), other different similarity indices may be calculated (e.g. Sorenson index when  $q=0$ , Horn index when  $q=1$ , Morisita-Horn index when  $q=2$ ) (Jost 2007).

**Value**

<code>Similarity</code>	A matrix of pairwise similarities showing community overlap calculated using the <code>similarity</code> function.
<code>StdErr</code>	(optional) Standard error of value estimated through bootstrapping.

**Author(s)**

Noah Charney, Sydne Record

**References**

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

See Also

[sim.table](#), [sim.groups](#), [bootstrap](#)

Examples

```
data(simesants)
hemlock<-subset(simesants,Habitat=="Hemlock")[, -1]
hardwood<-subset(simesants,Habitat=="Hardwood")[, -1]
similarity(rbind(hemlock,hardwood))
similarity(hemlock,abundances2=hardwood)
similarity(hemlock,abundances2=hardwood,boot=TRUE)
similarity(hemlock,abundances2=hardwood,q=2,boot=TRUE,boot.arg=list(num.iter=200))
```

---

turnover	<i>Turnover Rate per Sample</i>
----------	---------------------------------

---

Description

For numerous communities of equal weights, the numbers equivalent of the Shannon beta diversity and the number of samples (N) can be used to calculate the turnover rate per sample (Equation 25 from Jost 2007, Harrison et al. 1992).

Usage

```
turnover(abundances, abundances2 = NULL, q = 1, boot = FALSE, boot.arg = list(s.sizes = NULL, num.iter
```

Arguments

abundances	Community data as a matrix where columns are individual species and rows are sites or a vector of different species within a site. Matrix and vector elements are abundance data (e.g. counts, percent cover estimates).
abundances2	Community data, a vector of different species within a site. Vector elements are abundance data (e.g. counts, percent cover estimates). If abundances1 is given a matrix, then abundances2 defaults to NULL.
q	Order of the diversity measure. Defaults to the Shannon case where q = 1.
boot	Logical indicating whether to use bootstrapping to estimate uncertainty.
boot.arg	(optional) List of arguments to pass bootstrapping function: list(s.sizes=number you specify, num.iter=number you specify)

Value

Turnover	A scalar ranging from zero (no turnover between samples) to one (completely different samples).
StdErr	(optional) Standard error of value estimated through bootstrapping.

**Author(s)**

Noah Charney, Sydne Record

**References**

Jost, L. 2007. Partitioning diversity into independent alpha and beta components. *Ecology* 88(10): 2427-2439.

Harrison, S., S. Ross, and J. Lawton. 1992. Beta diversity on geographic gradients in Britain. *Journal of Animal Ecology* 61: 151-158.

**See Also**

[M.homog](#), [Rel.homog](#), [similarity](#), [bootstrap](#)

**Examples**

```
data(simesants)
turnover(simesants[,-1])
turnover(simesants[,-1],q=2,boot=TRUE)
turnover(simesants[,-1],q=2,boot=TRUE,boot.arg=list(num.iter=500))
```

# Index

- \*Topic **datasets**
  - simesants, [16](#)
- \*Topic **nonparametric**
  - bootstrap, [3](#)
- \*Topic **package**
  - vegetarian-package, [2](#)
- \*Topic **robust**
  - bootstrap, [3](#)
- bootstrap, [3](#), [6](#), [8](#), [9](#), [19](#), [20](#)
- d, [5](#), [5](#), [8](#), [11](#), [12](#)
- H, [6](#), [7](#)
- M. homog, [8](#), [13](#), [20](#)
- normalize.rows, [10](#)
- p.q.sum, [11](#)
- Rel.homog, [9](#), [12](#), [20](#)
- sim.groups, [13](#), [16](#), [19](#)
- sim.table, [14](#), [15](#), [19](#)
- simesants, [16](#)
- similarity, [13](#), [14](#), [16](#), [18](#), [20](#)
- turnover, [13](#), [19](#)
- vegetarian (vegetarian-package), [2](#)
- vegetarian-package, [2](#)