

# Package ‘tsvr’

February 27, 2019

**Version** 1.0.0

**Type** Package

**Title** Timescale-Specific Variance Ratio for Use in Community Ecology

**Description** Tools for timescale decomposition of the classic variance ratio of community ecology. Tools are as described in Zhao et al (in prep), extending commonly used methods introduced by Peterson et al (1975) <doi: 10.2307/1936306>.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Imports** graphics (>= 3.4.4), grDevices (>= 3.4.4), stats (>= 3.4.4)

**Suggests** knitr, rmarkdown, testthat, vdiff

**VignetteBuilder** knitr

**RoxygenNote** 6.0.1

**NeedsCompilation** no

**Author** Daniel C. Reuman [aut, cre],  
Lei Zhao [aut],  
Shaopeng Wang [aut]

**Maintainer** Daniel C. Reuman <reuman@ku.edu>

**Repository** CRAN

**Date/Publication** 2019-02-27 16:00:02 UTC

## R topics documented:

aggs	2
cospect	3
cv2	4
cv2f	5
errcheck_data	6
errcheck_tsvreq	6
errcheck_vreq	7
JRGdat	8

print.summary_tsvr . . . . .	8
set_ts . . . . .	9
tsvreq . . . . .	11
tsvreq_classic . . . . .	12
tsvreq_classic_methods . . . . .	13
tsvreq_methods . . . . .	15
vr . . . . .	16
vreq . . . . .	17
vreq_classic . . . . .	18
vreq_classic_ag_methods . . . . .	19
vreq_classic_methods . . . . .	21
vreq_LdM . . . . .	22
vreq_LdM_methods . . . . .	23
vreq_methods . . . . .	25
vrf . . . . .	26
wts . . . . .	27

## Index 28

---

aggts	<i>Aggregates a tsvreq_classic object across a set of timescales; also the constructor function for class vreq_classic_ag.</i>
-------	--

---

### Description

All the components of a tsvreq\_classic object can be aggregated across an arbitrary set of timescales, producing a new variance ratio equation - this function performs that aggregation. The function returns a vreq\_classic\_ag object, and is the constructor function of that class. The vreq\_classic\_ag class has slots com, comnull, vr, which are the same as a vreq object, but also has slot ts, which is the timescales over which aggregation was performed to get the object. The class inherits from vreq, which inherits from list.

### Usage

```
aggts(obj, ts)
```

### Arguments

obj	A tsvreq_classic object
ts	The timescales to aggregate over

### Details

Before aggregation is performed, the argument 'ts' is intersected with the canonical Fourier timescales greater than or equal to the Nyquist timescale, and the resulting timescales are then reflected about the Nyquist timescale. This is to account for the symmetry of Fourier transforms about the Nyquist frequency. The 'ts' slot of the output object shows the intersected, reflected timescales that were actually used for aggregation. See the examples.

**Value**

aggts returns an object of class vreq\_classic\_ag. Slots are:

com	the timescale-aggregated value of CVcom2
comnull	the timescale-aggregated value of CVcomip2
vr	the timescale-aggregated value of the classic variance ratio
ts	the timescales over which aggregation was performed

**Author(s)**

Shaopeng Wang, <shaopeng.wang@pku.edu.cn>; Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**References**

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

**See Also**

[tsvreq\\_classic](#), [vreq\\_classic\\_ag\\_methods](#), `browseVignettes("tsvr")`

**Examples**

```
X<-matrix(runif(10*100),10,100)
h<-tsvreq_classic(X)
res1<-aggts(h,h$ts[h$ts>=4])
res2<-aggts(h,h$ts[h$ts>=4 | h$ts<=4/3])
#res1 and res2 produce the same result
#because of Fourier symmetry around the
#Nyquist timescale - see Details
```

---

cospect

*Calculate the cospectrum between all pairs of time series*

---

**Description**

This function is used to calculate the cospectra between pairs of time series, including each time series with itself. These are based on simple ffts without smoothing.

**Usage**

```
cospect(X)
```

**Arguments**

X a matrix with counts or densities arranged in species by time step.

**Value**

cospect return a list with elements

frequency a vector from 0 to 1 of the frequencies used

cospectrum a 3D array, with cospectrum range in species by species by frequency

**Author(s)**

Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**Examples**

```
X<-matrix(runif(200,1,100), 10, 20)
ans<-cospect(X)
```

---

 cv2

---

*Calculates various measures of population and community variability*


---

**Description**

Calculates various measures of population and community variability

**Usage**

```
cv2(X, type)
```

**Arguments**

X A matrix with counts or densities arranged in species by time step

type If pop, calculate CVpop2. If com, calculate CVcom2. If comip, calculate CV-comip2. See the vignette for definitions of these quantities.

**Value**

cv2 returns the value of population or community variability.

**Author(s)**

Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>; Shaopeng Wang, <shaopeng.wang@pku.edu.cn>

**References**

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

**See Also**

[vreq\\_classic](#), [vreq\\_LdM](#), [cv2f](#), [browseVignettes\("tsvr"\)](#)

**Examples**

```
X<-matrix(runif(200,1,100), 10, 20)
ans<-cv2(X, type="com")
```

---

 cv2f

---

*Compute a frequency-specific version of CVcom or CVcomip*


---

**Description**

Compute a frequency-specific version of CVcom or CVcomip.

**Usage**

```
cv2f(X, type)
```

**Arguments**

X	A matrix with counts or densities arranged in species by time step
type	If com, calculate a frequency-specific CVcom2. If comip, calculate a frequency-specific CVcomip2. See the vignette for definitions of these quantities.

**Value**

cv2f returns an object of type list consisting of

frequency	a vector from 0 to 1 (not including 0 and 1)
cv2	A vector of frequency-specific population or community variability

**Author(s)**

Shaopeng Wang, <shaopeng.wang@pku.edu.cn>; Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**References**

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

**See Also**

[tsvreq\\_classic](#), [cv2](#), [browseVignettes\("tsvr"\)](#)

**Examples**

```
X<-matrix(runif(200,1,100), 10, 20)
ans<-cv2f(X, type="com")
```

---

errcheck_data	<i>Error check a data matrix.</i>
---------------	-----------------------------------

---

**Description**

Error check a data matrix.

**Usage**

```
errcheck_data(X, calledby)
```

**Arguments**

X	a matrix with counts or densities arranged in species by years. No NAs or negative values allowed, constant species not allowed.
calledby	the function calling this one

**Value**

errcheck\_data returns nothing but throws an error if the inputs do not meet the requirements

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

---

errcheck_tsvreq	<i>Error check inputs for the creator function for the tsvreq class.</i>
-----------------	--

---

**Description**

Error check inputs for the creator function for the tsvreq class.

**Usage**

```
errcheck_tsvreq(ts, com, comnull, tsvr, wts)
```

**Arguments**

ts	Timescales, should be a numeric vector of nonnegative numbers
com	Should be a vector of nonnegative numbers, equal to comnull*tsvr
comnull	Should be a vector of nonnegative numbers
tsvr	Should be a vector of nonnegative numbers
wt	Should be a vector of nonnegative numbers

**Value**

errcheck\_tsvreq returns nothing but throws an error if the inputs do not meet the requirements of a tsvreq object

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

---

errcheck\_vreq      *Error check inputs for the creator function for the vreq class.*

---

**Description**

Error check inputs for the creator function for the vreq class.

**Usage**

```
errcheck_vreq(com, comnull, vr)
```

**Arguments**

com	Should be a positive number, equal to comnull*vr
comnull	Should be a positive number
vr	Should be a positive number

**Value**

errcheck\_vreq returns nothing but throws an error if the inputs do not meet the requirements of a vreq object

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

---

JRGdat	<i>Percent cover data from Jasper Ridge Biological Preserve serpentine grassland site</i>
--------	---

---

### Description

A data set of percent cover of plant species in a plot from Jasper Ridge Biological Preserve serpentine grassland site. See Hallett et al (2014), Ecology for details. Subplot 3 from block 3 from the control treatment was used. Plot size was 1 m<sup>2</sup>. Values are percents. Values occasionally sum across species to more than 100 because of overlapping plant canopies.

### Usage

```
JRGdat
```

### Format

A data frame with 28 rows and 27 columns. First column is the year of measurement, subsequent columns correspond to individual species, named in the column headers.

### References

Hallett et al (2014) Biotic mechanisms of community stability shift along a precipitation gradient. Ecology 95, 1693-1700.

---

print.summary_tsvr	<i>Print method for summary_tsvr class</i>
--------------------	--

---

### Description

Print method for summary\_tsvr class

### Usage

```
## S3 method for class 'summary_tsvr'
print(x, ...)
```

### Arguments

x	A summary_tsvr object
...	Not currently used. Included for argument consistency with existing generics.

### Value

print.summary\_tsvr is called for its effect of printing to the screen.



**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**See Also**

[vreq\\_methods](#), [vreq\\_classic\\_methods](#), [vreq\\_LdM\\_methods](#), [vreq\\_classic\\_ag\\_methods](#), [tsvreq\\_methods](#), [tsvreq\\_classic\\_methods](#), [browseVignettes\("tsvr"\)](#)

**Examples**

```
res<-vreq(2,1,2)
summary(res)
```

---

set\_ts

*Set and get methods for classes in the tsvr package*

---

**Description**

Set and get methods for classes in the tsvr package. There are methods for each slot of each class, named set\_\* and get\_\* for \* the slot name. Below are listed function specs for the generics and the default methods.

**Usage**

```
set_ts(obj, newval)

## Default S3 method:
set_ts(obj, newval)

set_tsvr(obj, newval)

## Default S3 method:
set_tsvr(obj, newval)

set_wts(obj, newval)

## Default S3 method:
set_wts(obj, newval)

get_ts(obj)

## Default S3 method:
get_ts(obj)

get_tsvr(obj)
```

```
## Default S3 method:
get_tsvr(obj)

get_wts(obj)

## Default S3 method:
get_wts(obj)

set_com(obj, newval)

## Default S3 method:
set_com(obj, newval)

set_comnull(obj, newval)

## Default S3 method:
set_comnull(obj, newval)

set_vr(obj, newval)

## Default S3 method:
set_vr(obj, newval)

get_com(obj)

## Default S3 method:
get_com(obj)

get_comnull(obj)

## Default S3 method:
get_comnull(obj)

get_vr(obj)

## Default S3 method:
get_vr(obj)
```

### Arguments

obj	An object of one of the classes defined in the package
newval	A newvalue of the slot in question, for the set_* methods

### Details

There are methods for S3 classes defined in the package. See documentation for the generator functions for these classes (which in all cases have the same name as the class) for lists of slots for each class.

**Value**

set\_\* methods throw an error - setting of individual slots is not allowed, as it breaks consistency with the other slots. get\_\* just returns the value in question.

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**See Also**

[vreq](#)

---

 tsvreq

*Creator function for the tsvreq S3 class*

---

**Description**

The tsvreq (timescale-specific variance ratio equation) class is for storing functional equations based on a timescale-specific variance ratio. This is a general class from which other classes inherit (only tsvreq\_classic at this point). tsvreq inherits from the list class.

**Usage**

```
tsvreq(ts, com, comnull, tsvr, wts)
```

**Arguments**

ts	A vector of timescales
com	A numeric vector of the same length as ts containing nonnegative quantities
comnull	Another such
tsvr	Another such
wts	Another such

**Value**

tsvreq returns an object of class tsvreq. Slots are:

ts	the input
com	the input, equal to comnull*tsvr
comnull	the input
tsvr	the input
wts	the input

**Author(s)**

Shaopeng Wang, <shaopeng.wang@pku.edu.cn>; Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**See Also**

[tsvreq\\_methods](#), [tsvreq\\_classic](#), [vreq](#), [browseVignettes\("tsvr"\)](#)

---

 tsvreq\_classic

*Creator function for the tsvreq\_classic S3 class*


---

**Description**

The `tsvreq_classic` (timescale-specific variance ratio equation, classic variance ratio) class is for storing functional equations based on a timescale specific version of the classic variance ratio. Inherits from `tsvreq`, which inherits from `list`.

**Usage**

```
tsvreq_classic(X)
```

**Arguments**

`X` a matrix with counts or densities arranged in species by years

**Value**

`tsvreq_classic` returns a `tsvreq_classic` object. Slots are:

<code>ts</code>	a vector of timescales
<code>com</code>	a timescale-specific decomposition of <code>CVcom2</code>
<code>comnull</code>	a timescale-specific decomposition of <code>CVcomip2</code>
<code>tsvr</code>	a timescale-specific version of the classic variance ratio
<code>wts</code>	a vector of weights, same length as all the above

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**References**

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

**See Also**

[tsvreq\\_classic\\_methods](#), [tsvreq](#), [vreq\\_classic](#), [browseVignettes\("tsvr"\)](#)

**Examples**

```
X<-matrix(runif(10*100),10,100)
res<-tsvreq_classic(X)
```

---

tsvreq\_classic\_methods

*Basic methods for the tsvreq\_classic class*

---

**Description**

Set, get, summary, print and plot methods for the tsvreq\_classic class.

**Usage**

```
## S3 method for class 'tsvreq_classic'
summary(object, ...)

## S3 method for class 'tsvreq_classic'
print(x, ...)

## S3 method for class 'tsvreq_classic'
plot(x, filename = NA, ...)

## S3 method for class 'tsvreq_classic'
set_ts(obj, newval)

## S3 method for class 'tsvreq_classic'
set_com(obj, newval)

## S3 method for class 'tsvreq_classic'
set_comnull(obj, newval)

## S3 method for class 'tsvreq_classic'
set_tsvr(obj, newval)

## S3 method for class 'tsvreq_classic'
set_wts(obj, newval)

## S3 method for class 'tsvreq_classic'
get_ts(obj)

## S3 method for class 'tsvreq_classic'
get_com(obj)

## S3 method for class 'tsvreq_classic'
get_comnull(obj)
```

```
## S3 method for class 'tsvreq_classic'  
get_tsvr(obj)
```

```
## S3 method for class 'tsvreq_classic'  
get_wts(obj)
```

### Arguments

`object`, `x`, `obj` An object of class `tsvreq_classic`

... Passed to `plot`. Not currently used for other methods, included there only for argument consistency with existing generics.

`filename` A filename, no extension, could have a path. Used for saving a plot as a pdf. The default value `NA` causes the default plotting device to be used.

`newval` A new value, for the `set_*` methods

### Value

`summary.tsvreq_classic` produces a summary of a `tsvreq_classic` object. Methods `print.tsvreq_classic` and `plot.tsvreq_classic` are also available. For `tsvreq_classic` objects, `set_*` and `get_*` methods are available for all slots (see the documentation for `tsvreq_classic` for a list). The `set_*` methods just throw an error, to prevent breaking the consistency between the slots of a `tsvreq_classic` object.

### Author(s)

Daniel Reuman, <reuman@ku.edu>

### References

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

### See Also

[tsvreq\\_classic](#)

### Examples

```
X<-matrix(runif(10*100),10,100)  
res<-tsvreq_classic(X)  
get_ts(res)  
print(res)  
summary(res)
```

---

tsvreq_methods	<i>Basic methods for the tsvreq class</i>
----------------	---

---

**Description**

Set, get, summary, print and plot methods for the tsvreq class.

**Usage**

```
## S3 method for class 'tsvreq'  
summary(object, ...)
```

```
## S3 method for class 'tsvreq'  
print(x, ...)
```

```
## S3 method for class 'tsvreq'  
plot(x, filename = NA, ...)
```

```
## S3 method for class 'tsvreq'  
set_ts(obj, newval)
```

```
## S3 method for class 'tsvreq'  
set_com(obj, newval)
```

```
## S3 method for class 'tsvreq'  
set_comnull(obj, newval)
```

```
## S3 method for class 'tsvreq'  
set_tsvr(obj, newval)
```

```
## S3 method for class 'tsvreq'  
set_wts(obj, newval)
```

```
## S3 method for class 'tsvreq'  
get_ts(obj)
```

```
## S3 method for class 'tsvreq'  
get_com(obj)
```

```
## S3 method for class 'tsvreq'  
get_comnull(obj)
```

```
## S3 method for class 'tsvreq'  
get_tsvr(obj)
```

```
## S3 method for class 'tsvreq'  
get_wts(obj)
```

**Arguments**

object, x, obj	An object of class tsvreq
...	Passed to plot. Not currently used for other methods, included there only for argument consistency with existing generics.
filename	A filename, no extension, could have a path. Used for saving a plot as a pdf. The default value NA causes the default plotting device to be used.
newval	A new value, for the set_* methods

**Value**

summary.tsvreq produces a summary of a tsvreq object. Methods print.tsvreq and plot.tsvreq are also available. For tsvreq objects, set\_\* and get\_\* methods are available for all slots (see the documentation for tsvreq for a list). The set\_\* methods just throw an error, to prevent breaking the consistency between the slots of a tsvreq object.

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**See Also**

[tsvreq](#)

**Examples**

```
res<-tsvreq(ts=1:10,com=2*c(1:10),comnull=1:10,tsvr=rep(2,10),wts=rep(3,10))
get_ts(res)
print(res)
summary(res)
plot(res)
```

---

 vr

*Compute the classic or Loreau-de Mazancourt variance ratio*

---

**Description**

This function is used to compute the classical or Loreau-de Mazancourt variance ratio for a community in a single plot.

**Usage**

```
vr(X, method = "classic")
```



**Arguments**

`X` A matrix with counts or densities arranged in species by time steps

`method` If "classic" (default), use the classical method. If "LdM", use the Loreau-de Mazancourt method (see reference).

**Value**

`vr` returns the value of variance ratio

**Author(s)**

Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**References**

Loreau & Mazancourt, Species Synchrony and Its Drivers: Neutral and Nonneutral Community Dynamics in Fluctuating Environments. 2008, Am. Nat. 172(2)

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

Peterson, Stability of species and of community for the benthos of two lagoons. 1975, Ecology, 56, 958-965.

**See Also**

[vrf](#), [vreq](#), [vreq\\_classic](#), [vreq\\_LdM](#), [browseVignettes\("tsvr"\)](#)

**Examples**

```
X<-matrix(runif(200,1,100), 10, 20)
vr(X, method="LdM")
vr(X, method="classic")
```

---

vreq

*Creator function for the vreq S3 class*


---

**Description**

The `vreq` (variance ratio equation) class is for storing equations based on a variance ratio, as in Wang S. & Loreau M. (2016) Biodiversity and ecosystem stability across scales in metacommunities, Ecol Lett, 19, 510-518. This is a general class from which other classes inherit (`vreq_classic`, `vreq_LdM`). `vreq` inherits from the `list` class.

**Usage**

```
vreq(com, comnull, vr)
```

**Arguments**

com	A single positive number
comnull	Another single positive number
vr	Another single positive number

**Value**

vreq returns an object of class vreq. Slots are:

com	a single positive number equal to comnull*vr
comnull	a single positive number
vr	a single positive number

**Author(s)**

Shaopeng Wang, <shaopeng.wang@pku.edu.cn>; Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**References**

Wang S. & Loreau M. (2016) Biodiversity and ecosystem stability across scales in metacommunities. *Ecol Lett*, 19, 510-518.

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

Peterson, Stability of species and of community for the benthos of two lagoons. 1975, *Ecology*, 56, 958-965.

**See Also**

[vreq\\_methods](#), [vreq\\_classic](#), [vreq\\_LdM](#), [vreq\\_classic\\_ag\\_methods](#), [browseVignettes\("tsvr"\)](#)

---

vreq\_classic

*Creator function of vreq\_classic S3 class*

---

**Description**

The vreq\_classic (variance ratio equation, classic variance ratio) class is for storing equations based on the classic variance ratio. Inherits from the vreq class, which inherits from the list class.

**Usage**

```
vreq_classic(X)
```

**Arguments**

X A matrix with counts or densities arranged in species by years

**Value**

vreq\_classic returns a vreq\_classic object. Slots are:

com	the squared community CV, CVcom2
comnull	CVcomip2
vr	the classic variance ratio

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**References**

Peterson (1975) Stability of species and of community for the benthos of two lagoons. Ecology 56, 958-965.

**See Also**

[vreq\\_classic\\_methods](#), [vreq](#), [vreq\\_LdM](#), [vreq\\_classic\\_ag\\_methods](#), `browseVignettes("tsvr")`

**Examples**

```
X<-matrix(runif(10*100),10,100)
res<-vreq_classic(X)
```

---

vreq\_classic\_ag\_methods

*Basic methods for the vreq\_classic\_ag class*

---

**Description**

Set, get, summary, and print methods for the vreq\_classic\_ag class.

**Usage**

```
## S3 method for class 'vreq_classic_ag'
summary(object, ...)

## S3 method for class 'vreq_classic_ag'
print(x, ...)

## S3 method for class 'vreq_classic_ag'
set_com(obj, newval)

## S3 method for class 'vreq_classic_ag'
set_comnull(obj, newval)
```

```
## S3 method for class 'vreq_classic_ag'
set_vr(obj, newval)

## S3 method for class 'vreq_classic_ag'
set_ts(obj, newval)

## S3 method for class 'vreq_classic_ag'
get_com(obj)

## S3 method for class 'vreq_classic_ag'
get_comnull(obj)

## S3 method for class 'vreq_classic_ag'
get_vr(obj)

## S3 method for class 'vreq_classic_ag'
get_ts(obj)
```

### Arguments

object, x, obj An object of class `vreq_classic_ag`  
... Not currently used. Included for argument consistency with existing generics.  
newval A new value, for the `set_*` methods

### Value

`summary.vreq_classic_ag` produces a summary of a `vreq_classic_ag` object. A `print.vreq_classic_ag` method is also available. For `vreq_classic_ag` objects, `set_*` and `get_*` methods are available for all slots (see the documentation for `aggts` for a list). The `set_*` methods just throw an error, to prevent breaking the consistency between the slots of a `vreq_classic_ag` object.

### Author(s)

Daniel Reuman, <reuman@ku.edu>

### References

Zhao et al, (In prep) Decomposition of the variance ratio illuminates timescale-specific population and community variability.

### See Also

[aggts](#), [tsvreq\\_classic](#), [vreq\\_classic](#), [vreq\\_LdM](#), [vreq](#), [browseVignettes\("tsvr"\)](#)

### Examples

```
X<-matrix(runif(10*100),10,100)
h<-tsvreq_classic(X)
```

```
inp<-aggtts(h,h$ts[h$ts>4])
print(inp)
summary(inp)
```

---

vreq\_classic\_methods *Basic methods for the vreq\_classic class*

---

### Description

Set, get, summary, and print methods for the vreq\_classic class.

### Usage

```
## S3 method for class 'vreq_classic'
summary(object, ...)

## S3 method for class 'vreq_classic'
print(x, ...)

## S3 method for class 'vreq_classic'
set_com(obj, newval)

## S3 method for class 'vreq_classic'
set_comnull(obj, newval)

## S3 method for class 'vreq_classic'
set_vr(obj, newval)

## S3 method for class 'vreq_classic'
get_com(obj)

## S3 method for class 'vreq_classic'
get_comnull(obj)

## S3 method for class 'vreq_classic'
get_vr(obj)
```

### Arguments

object, x, obj An object of class vreq\_classic  
 ... Not currently used. Included for argument consistency with existing generics.  
 newval A new value, for the set\_\* methods

**Value**

`summary.vreq_classic` produces a summary of a `vreq_classic` object. A `print.vreq_classic` method is also available. For `vreq_classic` objects, `set_*` and `get_*` methods are available for all slots (see the documentation for `vreq_classic` for a list). The `set_*` methods just throw an error, to prevent breaking the consistency between the slots of a `vreq_classic` object.

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**References**

Peterson (1975) Stability of species and of community for the benthos of two lagoons. *Ecology* 56, 958-965.

**See Also**

[vreq\\_classic](#)

**Examples**

```
X<-matrix(runif(10*100),10,100)
res<-vreq_classic(X)
print(res)
summary(res)
```

---

vreq\_LdM

*Creator function of vreq\_LdM S3 class*

---

**Description**

The `vreq_LdM` (variance ratio equation, Loreau-de Mazancourt variance ratio) class is for storing equations based on the L-dM variance ratio. Inherits from the `vreq` class, which inherits from the `list` class.

**Usage**

```
vreq_LdM(X)
```

**Arguments**

X                    A matrix with counts or densities arranged in species by years

**Value**

vreq\_LdM returns a vreq\_LdM object. Slots are:

com	the squared community CV, CVcom2
comnull	CVpop2
vr	the L-dM variance ratio

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**References**

Loreau & Mazancourt, Species Synchrony and Its Drivers: Neutral and Nonneutral Community Dynamics in Fluctuating Environments. 2008, Am. Nat. 172(2)

**See Also**

[vreq\\_LdM\\_methods](#), [vreq\\_classic](#), [vreq](#), [browseVignettes\("tsvr"\)](#)

**Examples**

```
X<-matrix(runif(10*100),10,100)
res<-vreq_LdM(X)
```

---

vreq_LdM_methods	<i>Basic methods for the vreq_LdM class</i>
------------------	---

---

**Description**

Set, get, summary, and print methods for the vreq\_LdM class.

**Usage**

```
## S3 method for class 'vreq_LdM'
summary(object, ...)

## S3 method for class 'vreq_LdM'
print(x, ...)

## S3 method for class 'vreq_LdM'
set_com(obj, newval)

## S3 method for class 'vreq_LdM'
set_comnull(obj, newval)
```

```
## S3 method for class 'vreq_LdM'  
set_vr(obj, newval)  
  
## S3 method for class 'vreq_LdM'  
get_com(obj)  
  
## S3 method for class 'vreq_LdM'  
get_comnull(obj)  
  
## S3 method for class 'vreq_LdM'  
get_vr(obj)
```

### Arguments

object, x, obj An object of class vreq\_LdM  
... Not currently used. Included for argument consistency with existing generics.  
newval A new value, for the set\_\* methods

### Value

summary.vreq\_LdM produces a summary of a vreq\_LdM object. A print.vreq\_LdM method is also available. For vreq\_LdM objects, set\_\* and get\_\* methods are available for all slots (see the documentation for vreq\_LdM for a list). The set\_\* methods just throw an error, to prevent breaking the consistency between the slots of a vreq\_LdM object.

### Author(s)

Daniel Reuman, <reuman@ku.edu>

### References

Loreau & Mazancourt, Species Synchrony and Its Drivers: Neutral and Nonneutral Community Dynamics in Fluctuating Environments. 2008, Am. Nat. 172(2)

### See Also

[vreq\\_LdM](#)

### Examples

```
X<-matrix(runif(10*100),10,100)  
res<-vreq_LdM(X)  
print(res)  
summary(res)
```



---

vreq\_methods

*Basic methods for the vreq class*


---

### Description

Set, get, summary, and print methods for the vreq class.

### Usage

```
## S3 method for class 'vreq'
summary(object, ...)
```

```
## S3 method for class 'vreq'
print(x, ...)
```

```
## S3 method for class 'vreq'
set_com(obj, newval)
```

```
## S3 method for class 'vreq'
set_comnull(obj, newval)
```

```
## S3 method for class 'vreq'
set_vr(obj, newval)
```

```
## S3 method for class 'vreq'
get_com(obj)
```

```
## S3 method for class 'vreq'
get_comnull(obj)
```

```
## S3 method for class 'vreq'
get_vr(obj)
```

### Arguments

object, x, obj	An object of class vreq
...	Not currently used. Included for argument consistency with existing generics.
newval	A new value, for the set_* methods

### Value

summary.vreq produces a summary of a vreq object. A print.vreq method is also available. For vreq objects, set\_\* and get\_\* methods are available for all slots (see the documentation for vreq for a list). The set\_\* methods just throw an error, to prevent breaking the consistency between the slots of a vreq object.

**Author(s)**

Daniel Reuman, <reuman@ku.edu>

**See Also**

[vreq](#)

**Examples**

```
res<-vreq(com=2,comnull=1,vr=2)
print(res)
```

---

vrf

*Compute the frequency-specific variance ratio*

---

**Description**

This function computes the frequency-specific variance ratio for a community in a single plot.

**Usage**

```
vrf(X)
```

**Arguments**

X                    a matrix with counts or densities arranged in species by time step

**Value**

vrf returns a list consisting of

frequency           a vector from 0 to 1 (not including 0 and 1)

vr                   a vector of frequency-specific or frequency-decomposition of VR

**Author(s)**

Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**References**

<Lei's paper>

**See Also**

[vr](#)

**Examples**

```
X<-matrix(runif(200,1,100), 10, 20)
ans<-vrf(X)
```

---

wts	<i>Compute the weights (wts)</i>
-----	----------------------------------

---

**Description**

This function is used to compute weights (wts)

**Usage**

```
wts(X)
```

**Arguments**

X                    a matrix with counts or densities arranged in species by time step

**Value**

wts returns an object of class list consisting of

frequency           a vector from 0 to 1 (not include 0 and 1)

wts                  a vector of wts

**Author(s)**

Lei Zhao, <lei.zhao@cau.edu.cn>; Daniel Reuman, <reuman@ku.edu>

**References**

<Lei paper>

**Examples**

```
X<-matrix(runif(200,1,100), 10, 20)
ans<-wts(X)
```

# Index

## \*Topic **datasets**

- JRGdat, 8
- aggrs, 2, 20
- cospect, 3
- cv2, 4, 5
- cv2f, 5, 5
  
- errcheck\_data, 6
- errcheck\_tsvreq, 6
- errcheck\_vreq, 7
  
- get\_com(set\_ts), 9
- get\_com.tsvreq(tsvreq\_methods), 15
- get\_com.tsvreq\_classic(tsvreq\_classic\_methods), 13
- get\_com.vreq(vreq\_methods), 25
- get\_com.vreq\_classic(vreq\_classic\_methods), 21
- get\_com.vreq\_classic\_ag(vreq\_classic\_ag\_methods), 19
- get\_com.vreq\_LdM(vreq\_LdM\_methods), 23
- get\_comnull(set\_ts), 9
- get\_comnull.tsvreq(tsvreq\_methods), 15
- get\_comnull.tsvreq\_classic(tsvreq\_classic\_methods), 13
- get\_comnull.vreq(vreq\_methods), 25
- get\_comnull.vreq\_classic(vreq\_classic\_methods), 21
- get\_comnull.vreq\_classic\_ag(vreq\_classic\_ag\_methods), 19
- get\_comnull.vreq\_LdM(vreq\_LdM\_methods), 23
- get\_ts(set\_ts), 9
- get\_ts.tsvreq(tsvreq\_methods), 15
- get\_ts.tsvreq\_classic(tsvreq\_classic\_methods), 13
- get\_ts.vreq\_classic\_ag(vreq\_classic\_ag\_methods), 19
- get\_tsvr(set\_ts), 9
- get\_tsvr.tsvreq(tsvreq\_methods), 15
- get\_tsvr.tsvreq\_classic(tsvreq\_classic\_methods), 13
- get\_vr(set\_ts), 9
- get\_vr.vreq(vreq\_methods), 25
- get\_vr.vreq\_classic(vreq\_classic\_methods), 21
- get\_vr.vreq\_classic\_ag(vreq\_classic\_ag\_methods), 19
- get\_vr.vreq\_LdM(vreq\_LdM\_methods), 23
- get\_wts(set\_ts), 9
- get\_wts.tsvreq(tsvreq\_methods), 15
- get\_wts.tsvreq\_classic(tsvreq\_classic\_methods), 13
  
- JRGdat, 8
  
- plot.tsvreq(tsvreq\_methods), 15
- plot.tsvreq\_classic(tsvreq\_classic\_methods), 13
- print.summary\_tsvr, 8
- print.tsvreq(tsvreq\_methods), 15
- print.tsvreq\_classic(tsvreq\_classic\_methods), 13
- print.vreq(vreq\_methods), 25
- print.vreq\_classic(vreq\_classic\_methods), 21
- print.vreq\_classic\_ag(vreq\_classic\_ag\_methods), 19
- print.vreq\_LdM(vreq\_LdM\_methods), 23
  
- set\_com(set\_ts), 9
- set\_com.tsvreq(tsvreq\_methods), 15
- set\_com.tsvreq\_classic(tsvreq\_classic\_methods), 13
- set\_com.vreq(vreq\_methods), 25
- set\_com.vreq\_classic(vreq\_classic\_methods), 21

- set\_com.vreq\_classic\_ag  
    (vreq\_classic\_ag\_methods), 19
- set\_com.vreq\_LdM (vreq\_LdM\_methods), 23
- set\_comnull (set\_ts), 9
- set\_comnull.tsvreq (tsvreq\_methods), 15
- set\_comnull.tsvreq\_classic  
    (tsvreq\_classic\_methods), 13
- set\_comnull.vreq (vreq\_methods), 25
- set\_comnull.vreq\_classic  
    (vreq\_classic\_methods), 21
- set\_comnull.vreq\_classic\_ag  
    (vreq\_classic\_ag\_methods), 19
- set\_comnull.vreq\_LdM  
    (vreq\_LdM\_methods), 23
- set\_ts, 9
- set\_ts.tsvreq (tsvreq\_methods), 15
- set\_ts.tsvreq\_classic  
    (tsvreq\_classic\_methods), 13
- set\_ts.vreq\_classic\_ag  
    (vreq\_classic\_ag\_methods), 19
- set\_tsvr (set\_ts), 9
- set\_tsvr.tsvreq (tsvreq\_methods), 15
- set\_tsvr.tsvreq\_classic  
    (tsvreq\_classic\_methods), 13
- set\_vr (set\_ts), 9
- set\_vr.vreq (vreq\_methods), 25
- set\_vr.vreq\_classic  
    (vreq\_classic\_methods), 21
- set\_vr.vreq\_classic\_ag  
    (vreq\_classic\_ag\_methods), 19
- set\_vr.vreq\_LdM (vreq\_LdM\_methods), 23
- set\_wts (set\_ts), 9
- set\_wts.tsvreq (tsvreq\_methods), 15
- set\_wts.tsvreq\_classic  
    (tsvreq\_classic\_methods), 13
- setget\_methods (set\_ts), 9
- summary.tsvreq (tsvreq\_methods), 15
- summary.tsvreq\_classic  
    (tsvreq\_classic\_methods), 13
- summary.vreq (vreq\_methods), 25
- summary.vreq\_classic  
    (vreq\_classic\_methods), 21
- summary.vreq\_classic\_ag  
    (vreq\_classic\_ag\_methods), 19
- summary.vreq\_LdM (vreq\_LdM\_methods), 23
  
- tsvreq, 11, 12, 16
- tsvreq\_classic, 3, 5, 12, 12, 14, 20
- tsvreq\_classic\_methods, 9, 12, 13
  
- tsvreq\_methods, 9, 12, 15
  
- vr, 16, 26
- vreq, 11, 12, 17, 17, 19, 20, 23, 26
- vreq\_classic, 5, 12, 17, 18, 18, 20, 22, 23
- vreq\_classic\_ag\_methods, 3, 9, 18, 19, 19
- vreq\_classic\_methods, 9, 19, 21
- vreq\_LdM, 5, 17–20, 22, 24
- vreq\_LdM\_methods, 9, 23, 23
- vreq\_methods, 9, 18, 25
- vrf, 17, 26
  
- wts, 27