

# Package ‘flatness’

June 29, 2021

**Type** Package

**Title** Indices and Tests for Assessing the Flatness of (Rank)  
Histograms

**Version** 0.1.4

**Maintainer** Michaël Zamo <michael.zamo@meteo.fr>

**Description** S3 classes, plotting functions, indices and tests to analyse the flatness of histograms. It is specifically (but not only) aimed for assessing whether “rank” histograms (much used in weather forecasting) are flat. Specifically functions are provided to use the Jolliffe-Primo flatness tests introduced in Jolliffe and Primo (2008, <doi:10.1175/2007MWR2219.1>). Flatness indices described in Wilks (2019, <doi:10.1175/MWR-D-18-0369.1>) can be computed. Finally a function to use the Benjamini-Hochberg procedure for multiple hypothesis testing is provided (Benjamini and Hochberg, 1995, <doi:10.1111/j.2517-6161.1995.tb02031.x>).

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** data.table, lattice, xtable

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Michaël Zamo [aut, cre]

**Depends** R (>= 3.5.0)

**Repository** CRAN

**Date/Publication** 2021-06-29 07:20:09 UTC

## R topics documented:

BH_procedure . . . . .	2
boxplot.rkhist . . . . .	3

deviate_ends . . . . .	4
deviate_linear . . . . .	4
deviate_U . . . . .	5
deviate_V . . . . .	6
deviate_wave . . . . .	6
ensembles . . . . .	7
flatness . . . . .	8
flatness_indices . . . . .	9
flatness_indices.default . . . . .	9
flatness_indices.matrix . . . . .	10
get_deviates . . . . .	11
is_JP_ready . . . . .	12
is_orthonormal . . . . .	13
JP_test . . . . .	13
levelplot.JPtest . . . . .	14
make_JP_ready . . . . .	15
orthonormalize . . . . .	16
plot.rkhist . . . . .	16
ppensembles . . . . .	17
print.JPtest . . . . .	18
print.rkhist . . . . .	18
rbind_rkhists . . . . .	19
rkhist . . . . .	19
rkhist.data.frame . . . . .	20
rkhist.default . . . . .	21
rkhist.list . . . . .	22
rkhist.matrix . . . . .	22

## Index 24

---

BH_procedure	<i>Multiple statistical hypothesis testing with the Benjamini-Hochberg procedure</i>
--------------	--

---

## Description

This function applies the procedure described in Benjamini & Hochberg (1995) for controlling the False Discovery Rate in multiple statistical hypothesis testing.

## Usage

```
BH_procedure(pvalues, alpha = 0.01, ...)
```

## Arguments

pvalues	a vector of p-values, with length N
alpha	a numeric between 0 and 1
...	same arguments as in function base::sort, except "index.return" which is set to TRUE.

**Details**

The procedure works as follows. Let  $N$  p-values  $p_i$  (with  $i = 1, \dots, N$ ) and a significance level  $\alpha$ . The decision threshold is  $p_k$  where  $k = \max(i; p_{(i)} \leq \alpha \frac{i}{N})$  where  $p_{(i)}$  are the sorted p-values  $p_i$ . For every test with  $p_i \leq p_k$ , the null hypothesis is rejected. By convention  $p_{(0)} = 0$ .

**Value**

A named list with entries:

- H0: a logical vector of length  $N$ . The  $n$ th entry has value TRUE if the null hypothesis associated with the  $n$ th p-value is not rejected and FALSE otherwise.
- pk: the decision threshold. A p-value under this threshold leads to rejection of the associated null hypothesis.
- alpha: the chosen significance level
- pvalues: the vector of p-values provided to apply the Benjamini-Hochberg procedure.

**References**

Benjamini, Y., & Hochberg, Y. (1995). "Controlling the false discovery rate: a practical and powerful approach to multiple testing." *Journal of the Royal statistical society: series B (Methodological)*, 57(1), 289-300. doi:<https://doi.org/10.1111/j.2517-6161.1995.tb02031.x>

---

 boxplot.rkhist

*Methods for function boxplot of an S3 object of class rkhist*


---

**Description**

For each rank in an rkhist object, draws a boxplot of the counts for all ensembles. A line at the count value for a perfectly flat rank histogram is also drawn.

**Usage**

```
## S3 method for class 'rkhist'
boxplot(x, mini = min(0, min(x)), what = "counts", ...)
```

**Arguments**

x	an rkhist S3 object containing the rank histograms whose boxplot must be drawn. See function flatness::rkhist.
mini	a numeric. The minimum value in the ylim argument used to plot. Relevant only when plotting counts, ignored otherwise.
what	a character string taking its value in c("counts", "percents", "proportions"). What must be printed.
...	further arguments passed to function graphics::boxplot.matrix

**Value**

a list as in graphics::boxplot.

---

deviate_ends	<i>Return a deviate vector with deviates at the extreme rank</i>
--------------	--

---

**Description**

Based on the formula given in Jolliffe and Primo (2008) this function returns a vector with the right properties to test the presence of a deviate to flatness at both extreme ranks in a rank histogram.

**Usage**

```
deviate_ends(k)
```

**Arguments**

k                    an integer. The number of components.

**Details**

Although the deviate vector is not a rank histogram, this function returns an `rkhist` object for the sake of simplicity. This allows for instance to plot the vector.

**Value**

An `rkhist` object.

**References**

Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR2219.1>

---

deviate_linear	<i>Return a deviate vector with a linear trend</i>
----------------	--

---

**Description**

Based on the formula given in Jolliffe and Primo (2008) this function returns a vector with the right properties to test the presence of a linear deviate to flatness in a rank histogram.

**Usage**

```
deviate_linear(k)
```

**Arguments**

k                    an integer. The number of components.

**Details**

Although the deviate vector is not a rank histogram, this function returns an `rkhist` object for the sake of simplicity. This allows for instance to plot the vector.

**Value**

An `rkhist` object.

**References**

Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR2219.1>

---

`deviate_U`*Return a deviate vector with a U-shape trend*

---

**Description**

Based on the formula given in Jolliffe and Primo (2008) this function returns a vector with the right properties to test the presence of a U-shape deviate to flatness in a rank histogram.

**Usage**

```
deviate_U(k)
```

**Arguments**

`k` an integer. The number of components.

**Details**

Although the deviate vector is not a rank histogram, this function returns an `rkhist` object for the sake of simplicity. This allows for instance to plot the vector.

**Value**

An `rkhist` object.

**References**

Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR2219.1>

---

deviate_V	<i>Return a deviate vector with a V-shape trend</i>
-----------	---

---

### Description

Based on the formula given in Jolliffe and Primo (2008) this function returns a vector with the right properties to test the presence of a V-shape deviate to flatness in a rank histogram.

### Usage

```
deviate_V(k)
```

### Arguments

k                    an integer. The number of components.

### Details

Although the deviate vector is not a rank histogram, this function returns an `rkhist` object for the sake of simplicity. This allows for instance to plot the vector.

### Value

An `rkhist` object.

### References

Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR2219.1>

---

deviate_wave	<i>Return a deviate vector with a wave-shape trend</i>
--------------	--

---

### Description

This function returns a deviate vector with the right properties to test the presence of a deviate to flatness with a sine trend in a rank histogram, as introduced in (Zamo et al. 2021) or (Zamo, 2016).

### Usage

```
deviate_wave(k)
```

### Arguments

k                    an integer. The number of components.

## Details

Although the deviate vector is not a rank histogram, this function returns an `rkhist` object for the sake of simplicity. This allows for instance to plot the vector.

## Value

An `rkhist` object.

## References

- Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR221>
- Zamo, Michaël, Liliane Bel, and Olivier Mestre. "Sequential aggregation of probabilistic forecasts—application to wind speed ensemble forecasts." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 70.1 (2021): 202-225. doi:<https://doi.org/10.1111/rssc.12455>
- Zamo, Michaël. *Statistical Post-processing of Deterministic and Ensemble Wind Speed Forecasts on a Grid*. Diss. Université Paris-Saclay (ComUE), 2016.

---

ensembles

*Ensemble forecasts of temperature and associated observation*

---

## Description

This is a data set containing the forecasts of five ensemble weather prediction models for two-meter temperature from March, 2019 to March, 2021.

## Usage

ensembles

## Format

A named list with five entries, each containing a data table with 731 rows and a varying number of columns (depending on the number of members).

**date\_run** initial condition date (with format YYYY-MM-DD)

**latitude** latitude of the forecast location (in degrees)

**longitude** longitude of the forecast location (in degrees)

**1 ... M** member forecast (M members) in Kelvins

**T** the measured two-meter air temperature in Kelvins

## Details

The five ensemble models are named CWAO (20 perturbed members, from ECCC), DEMS (11 perturbed members, from NCMRWF), ECMF (50 members, from ECMWF), EGRR (17 perturbed members, from UKMO) and RKSL (24 perturbed members, from KMA).

The forecasts are the ones at the nearest grid point to Toulouse-Blagnac station (France) in the TIGGE dataset. The observation is the two-meter height temperature measured at this same station, at 06UTC. The forecast initial time is 00UTC, with a 30 hour lead-time.

## Source

<https://apps.ecmwf.int/datasets/data/tigge/levtype=sfc/type=pf/>

[https://donneespubliques.meteofrance.fr/?fond=produit&id\\_produit=91&id\\_rubrique=32](https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=91&id_rubrique=32)

---

flatness

*flatness: a package to assess the flatness of (rank) histograms*

---

## Description

The flatness package offers tools (scores, tests, ...) to compute histograms and assess whether they are flat.

## Details

The S3 generic function `rkhist` allows to compute one or several rank histograms from ensemble forecasts and corresponding scalar observations. (In Meteorology an ensemble forecast is a set of forecasts for the same variable, aimed at assessing the forecasting uncertainty). It creates an object with class `rkhist` that can then be plotted and printed.

Flatness of (rank) histograms may then be tested with function `JP_test` that implements the Jolliffe-Primo flatness tests. This test requires a set of deviance vectors, some of which can be provided with functions named `deviate_XXX`. The user can easily implement its own deviate-returning function (please see details in `get_deviates` on how to do this). Functions `is_JP_ready` and `make_JP_ready` are provided to ensure that a set of deviate vectors meet the requirements to be used in the Jolliffe-Primo tests. The result of the test is stored in an object with class `JPtest` that can be printed or drawn with the function `lattice::levelplot`.

Flatness indices can be computed with the S3 generic function `flatness_indices`.

See the vignette for further details and an illustration with the datasets `ensembles` and `ppensembles` provided with this package.



---

flatness_indices	<i>Compute flatness indices for (rank) histograms</i>
------------------	---

---

**Description**

S3 generic function that computes and returns indices of flatness of one or several (rank) histograms, presented in Wilks (2019).

**Usage**

```
flatness_indices(rkhists, ...)
```

**Arguments**

rkhists	an object containing (rank) histograms
...	other arguments

**Details**

Currently the implemented flatness indices are the chi-square statistics, the reliability index and the entropy.

**Value**

the expected returned object is a matrix, with one flatness index in each column for each rank histogram (row-wise). The columns should be named, with "chisq" for the chi-square statistics, "RI" for the reliability index and "entropy" for the entropy.

**References**

Wilks, D. S. "Indices of rank histogram flatness and their sampling properties." *Monthly Weather Review* 147.2 (2019): 763-769. doi:<https://doi.org/10.1175/MWR-D-18-0369.1>

---

flatness_indices.default	<i>Default method for S3 generic function flatness_indices</i>
--------------------------	--

---

**Description**

Just generate an error.

**Usage**

```
## Default S3 method:  
flatness_indices(rkhists, ...)
```

**Arguments**

rkhist            an object containing (rank) histograms  
...               other arguments

**Value**

No return value, called for side effects (generate an error message).

---

flatness\_indices.matrix

*Method for S3 generic function flatness\_indices and objects with class matrix*

---

**Description**

This function is the method called when using the S3 generic function flatness\_indices with an object of class matrix.

**Usage**

```
## S3 method for class 'matrix'  
flatness_indices(rkhist, col_wise = FALSE, ...)
```

**Arguments**

rkhist            an object containing (rank) histograms  
col\_wise          a logical. Are the rank histograms stored column wise, that is one rank histogram in each column?  
...               other arguments

**Details**

The input matrices is supposed to contain one rank histograms in each row, unless col\_wise == TRUE.

**Value**

See *Value* in function flatness\_indices

---

get_deviates	<i>Return a set of vectors with chosen shapes</i>
--------------	---

---

### Description

This function returns an `rkhist` object containing vectors with chosen shapes or trends. This is intended to be used to apply the Jolliffe-Primo flatness tests of rank histograms (see Jolliffe and Primo, 2008).

### Usage

```
get_deviates(k, shapes = c("linear", "U", "wave"), constrain = FALSE)
```

### Arguments

<code>k</code>	an integer. The number of possible ranks.
<code>shapes</code>	a vector of character strings. The required shapes of the vectors.
<code>constrain</code>	a logical. If TRUE the returned set of vectors is constrained to be orthonormal, with each vector having components summing to 0. This is required to use the vectors in the Jolliffe-Primo flatness test.

### Details

The convention is that each row of the `rkhist` object contains a vector. It is not required that the set be a basis.

For each shape in `shapes` this function calls a function named `'deviate_shape'` with one argument `k`. Some pre-coded functions already exist but the user can easily add its own by following this naming convention. The added function must have only one argument `k` and return an `rkhist` object. It is advised that the returned deviate vector's components should sum to 0 and have a unit module. But this can be imposed by setting the argument `constrain` to TRUE.

If `constrain == TRUE` the vector set is modified to have the right properties to be used in the Jolliffe-Primo test, through the Gram-Schmidt method. It is strongly advised to plot the resulting set with function `flatness::plot`, since this transformation may greatly change the shape of the original vectors.

### Value

An `rkhist` object with each row representing a vector of deviation from flatness.

### References

- Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR221>
- Zamo, Michaël, Liliane Bel, and Olivier Mestre. "Sequential aggregation of probabilistic forecasts—application to wind speed ensemble forecasts." *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 70.1 (2021): 202-225. doi:<https://doi.org/10.1111/rssc.12455>

- Zamo, Michaël. Statistical Post-processing of Deterministic and Ensemble Wind Speed Forecasts on a Grid. Diss. Université Paris-Saclay (ComUE), 2016.

### Examples

```

deviates <- get_deviates(k = 36, shapes = c("linear", "U", "V", "ends", "wave"))
plot(deviates)
isJPOK <- is_JP_ready(deviates)
JPdeviates <- make_JP_ready(deviates)
plot(JPdeviates)
JPcheck <- is_JP_ready(JPdeviates)

```

---

is\_JP\_ready

*Check whether a set of vectors obeys the constraints necessary to be used in the Jolliffe-Primo flatness test*

---

### Description

This function checks whether a set of vectors has the following two properties:

- the set is orthonormal
- each vector has components summing to zero (that is, it is a deviation)

### Usage

```
is_JP_ready(x, verbose = TRUE, tol = 1e-04)
```

### Arguments

**x** an rkhist or matrix object containing the vectors to check (by row).  
**verbose** a logical. Should the result of the check be displayed?  
**tol** a positive numeric. The accepted tolerance for the constraints.

### Value

A list with entries

**isOK** TRUE if the set obeys the constraints, FALSE otherwise

**tol** the tolerance allowed on the constraints

**crossprod** the cross product of the vectors

**sums** the sum of the components of each vector

**x** the checked vectors

---

is_orthonormal	<i>Check whether a vector set is orthonormal</i>
----------------	--

---

**Description**

Check whether a vector set is orthonormal

**Usage**

```
is_orthonormal(set, tol = 1e-04)
```

**Arguments**

set	a matrix. The convention used here is that each row of set contains a vector.
tol	a positive numeric. The accepted tolerance for the conditions of orthonormality.

**Value**

TRUE if the set of vectors is orthonormal, FALSE otherwise.

---

JP_test	<i>Implementation of the Jolliffe-Primo flatness tests for rank histograms</i>
---------	--

---

**Description**

Given a matrix of rank histograms and an orthonormal set of deviate vector(s), this function computes the projection components, test statistics and p-values of the Jolliffe-Primo flatness test for each inputted rank histogram. See Jolliffe and Primo (2008) for details of the method.

**Usage**

```
JP_test(rkhists, deviates, ...)
```

**Arguments**

rkhists	an rkhist, a matrix, or any other object that can be coerced to a matrix. It contains the rank histogram(s) whose flatness must be tested (one in each row).
deviates	the matrix containing the deviate vectors used for testing. Each row contains a deviate vector: the vector set must be orthonormal, and each deviate vector must have its components summing to zero.
...	further arguments (currently not used).

**Details**

Note that the test statistics and p-values of the projections over the residual vector (after removing all the projection on the deviates) are also computed and returned.

**Value**

A list (with additional first class JPtest) with the following entries:

**test** an array containing the result of the Jolliffe-Primo test(s). The first dimension is of length three (the projection over the deviate vectors, the test statistics and the p-values), the second and third dimensions correspond to the rank histogram and the test, respectively

**deviates** the set of deviate vectors used in the test

**rkhist** the tested rank histogram(s) (an rkhist object).

**References**

Jolliffe, Ian T., and Cristina Primo. "Evaluating rank histograms using decompositions of the chi-square test statistic." *Monthly Weather Review* 136.6 (2008): 2133-2139. doi:<https://doi.org/10.1175/2007MWR2219.1>

**Examples**

```
require(lattice)
require(xtable)
M <- 15
N <- 100
n <- 20
fcsts <- vector("list", n)
names(fcsts) <- letters[1:n]
obs <- rnorm(N)
for (i in 1:n) {
  fcsts[[i]] <- matrix(rnorm(M*N), ncol = M)
}
rkhsts <- rkhist(fcsts, obs)
deviates <- get_deviates(M + 1)
test <- JP_test(rkhsts, deviates)
print(test)
for (what in c("projections", "statistics", "pvalues")){
  levelplot(test, what = what, main = what, rotate = what == "pvalues")
}
xtable(test$test["pvalues", ,])
xtable(t(test$test["pvalues", ,]))
```

---

levelplot.JPtest      *Levelplot method for data in a JPtest object*

---

**Description**

Plot a chosen result matrix contained in the *test* entry of a JPtest object. The underlying function is the `lattice::levelplot.matrix` function.

**Usage**

```
## S3 method for class 'JPtest'
levelplot(JPobj, what = "pvalues", rotate = TRUE, plot = TRUE, ...)
```

**Arguments**

JPobj	the JPtest object to plot.
what	a character or an integer. Which component of the test entry in JPobj should be plotted? Can be any of <i>projections</i> , <i>statistics</i> , <i>pvalues</i> or, respectively, 1, 2 or 3.
rotate	a logical. Should the matrix containing the data be transposed before plotting? By default in <code>lattice::levelplot.matrix</code> , the rows of the plotted matrix correspond to the x-axis.
plot	a logical. Should the lattice object be plotted?
...	further arguments passed to the <code>lattice::levelplot</code> function.

**Value**

An object of class `trellis`, invisibly.

---

make_JP_ready	<i>Modify a set of rkhist objects so that they can be used in the Jolliffe-Primo flatness test</i>
---------------	--

---

**Description**

The Jolliffe-Primo flatness test requires deviate vectors that are orthonormal and that each has components summing to zero. This function ensures this by using, if it is required, the Gram-Schmidt method to make the set orthonormal and also makes the sum of each vector's components equal to zero.

**Usage**

```
make_JP_ready(x, verbose = TRUE)
```

**Arguments**

x	an rkhist object.
verbose	a logical. Should the result of the check be displayed?

**Details**

Note this procedure may greatly change the shape of vectors.

**Value**

An rkhist object, containing the modified vectors set obeying the requirements for the Jolliffe-Primo flatness test.

---

orthonormalize	<i>Make a vector set orthonormal</i>
----------------	--------------------------------------

---

**Description**

This function uses the Gram-Schmidt method to make a set of vector orthonormal.

**Usage**

```
orthonormalize(set)
```

**Arguments**

set                    a matrix. The convention used here is that each row of set contains a vector.

**Value**

A matrix with the same dimension as set containing an orthonormal set of vector. The vector are stored in each row.

---

plot.rkhist	<i>Method for function plot an S3 object of class rkhist</i>
-------------	--

---

**Description**

Plot a rank histogram stored in an object of S3 class rkhist, with an horizontal dashed line indicating the expected count for a perfectly flat rank histogram.

**Usage**

```
## S3 method for class 'rkhist'
plot(x, mini = min(0, min(x)), what = "counts", ...)
```

**Arguments**

x                    an rkhist S3 object containing the rank histogram to plot. See function flatness::rkhist.  
 mini                a numeric. The minimum value in the ylim argument used to plot. Relevant only when plotting counts, ignored otherwise.  
 what                a character string taking its value in c("counts", "percents", "proportions"). What must be plotted.  
 ...                other arguments passed to function base::plot.default. Some arguments are already used in this function, and may not be changed.

**Value**

NULL, invisibly.



---

ppensembles	<i>Post-processed ensemble forecasts of temperature and associated observation</i>
-------------	--

---

### Description

This is a dataset containing the post-processed forecasts of five ensemble weather prediction models for two-meter temperature from March, 2019 to March, 2021.

### Usage

ppensembles

### Format

A named list with five entries, each containing a `data.table` with 731 rows.

**date\_run** initial condition date (with format YYYY-MM-DD)

**latitude** latitude of the forecast location (in degrees)

**longitude** longitude of the forecast location (in degrees)

**1 ... 30** forecast in Kelvins sampled from the gaussian distribution. The forecasts are sorted in increasing order.

**T** the measured two-meter air temperature in Kelvins

### Details

Each ensemble has been post-processed with the non homogeneous regression technique, described in Gneiting et al.(2005). In a nutshell the true distribution is supposed to be gaussian, with mean and standard deviation being a linear function of the ensemble mean and standard deviation (respectively). The intercept and slope of each regression is determined by minimizing the CRPS over a 60-day sliding window. The forecast in the data set is a sample of 30 values from this gaussian distribution.

The five ensemble models are named based on the raw ensemble: CWA0 (from ECCC), DEMS (from NCMRWF), ECMF (from ECMWF), EGRR (from UKMO) and RKSL (from KMA).

The raw forecasts are the ones at the nearest grid point to Toulouse-Blagnac station (France) in the TIGGE data set. The observation is the two-meter height temperature measured at this same station, at 06UTC. The forecast initial time is 00UTC, with a 30 hour lead-time.

### Source

<https://apps.ecmwf.int/datasets/data/tigge/levtype=sfc/type=pf/>

[https://donneespubliques.meteofrance.fr/?fond=produit&id\\_produit=91&id\\_rubrique=32](https://donneespubliques.meteofrance.fr/?fond=produit&id_produit=91&id_rubrique=32)

**References**

Gneiting, Tilmann, et al. "Calibrated probabilistic forecasting using ensemble model output statistics and minimum CRPS estimation." *Monthly Weather Review* 133.5 (2005): 1098-1118. doi:<https://doi.org/10.1175/MWR2>

---

print.JPtest	<i>Print method for a JPtest object</i>
--------------	---

---

**Description**

Print method for a JPtest object

**Usage**

```
## S3 method for class 'JPtest'
print(x, what = c("projections", "statistics", "pvalues"), ...)
```

**Arguments**

x	the JPtest object that must be printed.
what	a character vector. What must be printed? Can contain any or several among c("projections", "statistics", "pvalues").
...	further arguments. Passed to print method for matrix objects.

**Value**

No return value, called for side effects.

---

print.rkhist	<i>Print an S3 object of class rkhist</i>
--------------	---

---

**Description**

Print an S3 object of class rkhist, using the same layout as base::print.table.

**Usage**

```
## S3 method for class 'rkhist'
print(x, what = "counts", ...)
```

**Arguments**

x	an rkhist S3 object containing the rank histogram to print. See function flatness::rkhist.
what	a character string taking its value in c("counts", "percents", "proportions"). What must be printed.
...	further arguments, passed to print.

**Value**

No return value, called for side effects.

---

rbind_rkhists	<i>Stack objects of class rkhist</i>
---------------	--------------------------------------

---

**Description**

Use rbind to stack several rank histograms (stored in S3 objects of class rkhist, or matrix, or vectors).

**Usage**

```
rbind_rkhists(..., names = NULL)
```

**Arguments**

...	matrix, rkhist objects or vectors to bind by row.
names	a character vector of names to give to each rank histogram (i.e. each row of the resulting matrix). Although its length must be equal to the number of row in the resulting matrix, NAs are allowed.

**Value**

An object of S3 class rkhist

---

rkhist	<i>Return the rank histogram of an observation in an ensemble forecast</i>
--------	--

---

**Description**

This S3 generic function is intended to compute the rank of each observation when pooled with its associated ensemble forecast and return the count in each rank (i.e. the rank histogram).

**Usage**

```
rkhist(fcst, obs, ...)
```

**Arguments**

fcst	an object containing the ensemble forecasts.
obs	an object containing the observation associated to the forecast in fcst.
...	additional arguments.

**Details**

For new methods, the output should be an object of class `c("rkhist", "matrix")`, with one rank histogram in each row. Rows may be named.

**Value**

An S3 object of class `rkhist` (indeed a matrix containing the count for each rank, with class `"rkhist"`). Each row of the matrix contains the counts for one rank histogram.

**Examples**

```
set.seed(42)
N <- 1000
M <- 20
fcst <- matrix(rnorm(N*M), ncol = M)
fcst2 <- matrix(rnorm(N*M), ncol = M)
obs <- rnorm(N)
## Computation of one rank histogram
# Named
rkh <- rkhist(fcst, obs, names = "a")
print(rkh)
plot(rkh)
# Unnamed
rkh2 <- rkhist(fcst2, obs)
print(rkh2)
plot(rkh2)

## Computation of two rank histograms, from a list of forecasts, with the
## same observation vector
fcstsl <- list(fcst, fcst2)
rkhs1 <- rkhist(fcstsl, obs, names = c("a", NA))
print(rkhs1)
plot(rkhs1)

## Concatenation of two rank histograms, with different names
rkhs <- rbind_rkhists(rkh, rkh2, names = letters[3:4])
rownames(rkhs)
print(rkhs)
plot(rkhs)
```

---

rkhist.data.frame

*Method of S3 generic function rkhist for data.frame objects*


---

**Description**

This is the method called when the `fcst` argument in function `rkhist` is a data frame.

**Usage**

```
## S3 method for class 'data.frame'
rkhist(fcst, obs, names = NULL, ...)
```

**Arguments**

fcst	an object containing the ensemble forecasts. It must be a data frame, each row containing a forecast.
obs	an object containing the observation associated to the forecast in fcst. It can be a vector or a matrix, whose length is the same as the number of row in fcst
names	a character vector. The row names in the returned rkhist object.
...	additional arguments, passed to function base::rank.

**Value**

An S3 object of class rkhist (indeed just 1-row matrix containing the count for each rank, with class == "rkhist").

---

rkhist.default	<i>Default method for S3 generic function rkhist</i>
----------------	--

---

**Description**

Just generate an error.

**Usage**

```
## Default S3 method:
rkhist(fcst, obs, ...)
```

**Arguments**

fcst	an object containing the ensemble forecasts.
obs	an object containing the observation associated to the forecast in fcst.
...	additional arguments.

**Value**

No return value, called for side effects (generate an error message).

---

rkhist.list

*Method of S3 generic function rkhist for list objects*


---

### Description

This is the method called when the `fcst` argument in function `rkhist` is a list.

### Usage

```
## S3 method for class 'list'
rkhist(fcst, obs, names = NULL, ...)
```

### Arguments

<code>fcst</code>	an object containing the ensemble forecasts. It must be a list, each entry containing the forecast of one ensemble. The forecast of each ensemble is a matrix whose rows contain one forecast.
<code>obs</code>	an object containing the observation associated to the forecast in <code>fcst</code> . It can be a vector or a list, whose length is the same as the number of row in <code>fcst</code> . If a vector, it is used as the observation vector for all the ensemble in <code>fcst</code> . If it is a list, each entry is associated to the same entry in <code>fcst</code> .
<code>names</code>	a character vector. The row names in the returned <code>rkhist</code> object. If missing, the row names are the names of the list <code>fcst</code> (if any).
<code>...</code>	additional arguments, passed to function <code>base::rank</code> .

### Value

An S3 object of class `rkhist` (indeed an N-row matrix containing the count for each rank, with class "rkhist"). N is the length of `fcst`, i.e. the number of ensembles.

---

rkhist.matrix

*Method of S3 generic function rkhist for matrix objects*


---

### Description

This is the method called when the `fcst` argument in function `rkhist` is a matrix.

### Usage

```
## S3 method for class 'matrix'
rkhist(fcst, obs, names = NULL, ...)
```

**Arguments**

<code>fcst</code>	an object containing the ensemble forecasts. It must be a matrix, each row containing a forecast.
<code>obs</code>	an object containing the observation associated to the forecast in <code>fcst</code> . It can be a vector or a matrix, whose length is the same as the number of row in <code>fcst</code>
<code>names</code>	a character vector. The row names in the returned <code>rkhist</code> object.
<code>...</code>	additional arguments, passed to function <code>base::rank</code> .

**Value**

An S3 object of class `rkhist` (indeed just 1-row matrix containing the count for each rank, with `class == "rkhist"`).

# Index

- \* **datasets**
  - ensembles, [7](#)
  - ppensembles, [17](#)
- BH\_procedure, [2](#)
- boxplot.rkhist, [3](#)
- deviate\_ends, [4](#)
- deviate\_linear, [4](#)
- deviate\_U, [5](#)
- deviate\_V, [6](#)
- deviate\_wave, [6](#)
- ensembles, [7](#)
- flatness, [8](#)
- flatness\_indices, [9](#)
- flatness\_indices.default, [9](#)
- flatness\_indices.matrix, [10](#)
- get\_deviates, [11](#)
- is\_JP\_ready, [12](#)
- is\_orthonormal, [13](#)
- JP\_test, [13](#)
- levelplot.JPtest, [14](#)
- make\_JP\_ready, [15](#)
- orthonormalize, [16](#)
- plot.rkhist, [16](#)
- ppensembles, [17](#)
- print.JPtest, [18](#)
- print.rkhist, [18](#)
- rbind\_rkhists, [19](#)
- rkhist, [19](#)
- rkhist.data.frame, [20](#)
- rkhist.default, [21](#)
- rkhist.list, [22](#)
- rkhist.matrix, [22](#)