Package ‘fechner’

June 6, 2016

Version 1.0-3
Date 2016-06-05
Title Fechnerian Scaling of Discrete Object Sets
Description Functions and example datasets for Fechnerian scaling of discrete object sets. User can compute Fechnerian distances among objects representing subjective dissimilarities, and other related information. See package?fechner for an overview.
Author Ali Uenlue [aut, cre], Thomas Kiefer [aut, trl]
(Based on original MATLAB source by Ehtibar N. Dzhafarov.)
Maintainer Ali Uenlue <ali.uenlue@tum.de>
Depends R (>= 3.3.0)
Imports graphics, stats
LazyLoad yes
LazyData yes
License GPL (>= 2)
URL http://www.meb.edu.tum.de
NeedsCompilation yes
Repository CRAN
Date/Publication 2016-06-06 13:35:26

R topics documented:

fechner-package .................................................. 2
check.data ...................................................... 3
check.regular .................................................... 5
fechner ............................................................. 7
morse .............................................................. 13
noRegMin ......................................................... 14
plot.fechner ..................................................... 15
print.fechner ................................................... 17
print.summary.fechner ......................................... 18
Fechnerian scaling is a procedure for constructing a metric on a set of objects (e.g., symbols, X-ray films). The constructed Fechnerian metric represents subjective dissimilarities among the objects as perceived by a system (e.g., person, technical device). The package `fechner` provides functions and example datasets for performing and illustrating Fechnerian scaling of discrete object sets in R.

Fechnerian scaling of discrete object (or stimulus) sets provides a theoretical framework for deriving, so-called Fechnerian, distances among objects representing subjective dissimilarities. A Fechnerian metric on a set of stimuli is constructed from the probabilities with which the objects are discriminated from each other by a perceiving system. In addition to the oriented and overall Fechnerian distances, the package `fechner` also computes such related information as the points of subjective equality, the psychometric increments, the geodesic chains and loops with their corresponding lengths, and the generalized Shepardian dissimilarities (or S-index). Moreover, the package `fechner` provides functions for checking the required data format and the fundamental regular minimality/maximality condition. These concepts are explained in detail in the paper about the `fechner` package by Uenlue, Kiefer, and Dzhafarov (2009), and in the theoretical papers by Dzhafarov and Colonius (2006, 2007) (see ‘References’).

The package `fechner` is implemented based on the S3 system. It comes with a namespace, and consists of three external functions (functions the package exports): `check.data`, `check.regular`, and the main function of this package, `fechner`. It also contains six internal functions (functions not exported by the package), which are `plot`, `print`, and summary methods for objects of the class `fechner`, a `print` method for objects of the class `summary.fechner`, and two functions for computing intermediate graph-theoretic information: `plot.fechner`, `print.fechner`, `summary.fechner`, `print.summary.fechner`, and `fechner-internal`. The features of the package `fechner` are illustrated with accompanying two real datasets, `morse` and `wish`, and two artificial datasets, `regMin` and `noRegMin`.

**Details**

- **Package:** `fechner`
- **Type:** Package
- **Version:** 1.0-3
- **Date:** 2016-06-05
- **License:** GPL (>= 2)
Author(s)
Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.
Maintainer: Ali Uenlue <ali.uenlue@tum.de>

References
Dzhafarov, E. N. and Colonius, H. (2006) Reconstructing distances among objects from their dis-


Journal of Mathematical Psychology, 51, 290–304.

check.data

Check for Required Data Format

Description
check.data is used to check whether the data are of required format.

Usage
check.data(X, format = c("probability.different", "percent.same",
"general"))

Arguments
X
a required square matrix or data frame of numeric data. No NA, NaN, Inf, or
−Inf values are allowed.

format
an optional character string giving the data format to be checked. This must
be one of "probability.different", "percent.same", or "general", with
default "probability.different", and may be abbreviated to a unique prefix.

Details
The data must be a matrix or a data frame, have the same number of rows and columns, and be
numeric consisting of real numbers. In particular, no infinite, undefined, or missing values are
allowed. This is the general data format. The probability-different and percent-same formats, in
addition, require that the data lie in the intervals \([0, 1]\) and \([0, 100]\), respectively. If all of the re-
quirements for a data format are satisfied, the data are returned as a matrix with rows and columns
labeled; otherwise the function produces respective messages.

The labeling is as follows.

1. If the data are entered without any labeling of the rows and columns: The function does
the labeling automatically, as \(a_1, b_1, \ldots, z_1, a_2, b_2, \ldots, z_2, \ldots\), etc., up to \(a_9, b_9, \ldots, z_9\) if
the data are as large as \(234 \times 234\), or if the data are larger than \(234 \times 234\), the labeling is
\(v_1, v_2, \ldots, v_N\), where \(N \times N\) is the dimension of the data (and \(N > 234\)).
2. If the data are entered with either row or column labeling: In that case, the row or column labels are assigned to the columns or rows, respectively.

3. If the data are entered with row and column labeling: Since the labeling of both the rows and columns is now provided by the user manually, the same labeling must be used for both. If this is the case, the labeling is adopted. Otherwise the function produces a respective message.

Value

If the data are of required format, check.data returns a matrix of the data with rows and columns labeled.

Author(s)

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

References


See Also

check.regular for checking regular minimality/maximality; fechner, the main function for Fechnerian scaling. See also fechner-package for general information about this package.

Examples

```r
## dataset \link{wish} is of probability-different format
check.data(wish)

## dataset \link{morse} is of percent-same format
check.data(morse, format = "percent.same")

## a matrix without any labeling of rows and columns, of general format
## check.data does the labeling automatically
(X <- ((-1) * matrix(1:16, nrow = 4)))
check.data(X, format = "general")

## examples of data that are not of any of the three formats

## message: data must be matrix or data frame
check.data(as.character(matrix(1:16, nrow = 4)))

## message: data must have same number of rows and columns
check.data(matrix(1:12, nrow = 4))
```
Description

check.regular is used to check whether the data satisfy regular minimality/maximality.

Usage

check.regular(x, type = c("probability.different", "percent.same", "reg.minimal", "reg.maximal"))

Arguments

x a required square matrix or data frame of numeric data. No NA, NaN, Inf, or -Inf values are allowed.

type an optional character string giving the type of check to be performed. This must be one of "probability.different", "percent.same", "reg.minimal", or "reg.maximal", with default "probability.different", and may be abbreviated to a unique prefix.

Details

The type argument specifies whether regular minimality or regular maximality is to be checked. "probability.different" and "percent.same" are for datasets in the probability-different and percent-same formats, and imply regular minimality and regular maximality checks, respectively. "reg.minimal" and "reg.maximal" can be specified to force checking for regular minimality and regular maximality, respectively, independent of the used dataset. In particular, "reg.minimal" and "reg.maximal" are to be used for datasets that are properly in the general format.

check.regular calls check.data. In particular, the rows and columns of the canonical representation matrix (see 'Value') are canonically relabeled based on the labeling provided by check.data. That is, using the check.data labeling, the pairs of points of subjective equality (PSEs) are assigned identical labels, leaving intact the labeling of the rows and relabeling the columns with their corresponding PSEs. If the data X do not satisfy regular minimality/maximality, check.regular produces respective messages. The latter give information about parts of X violating that condition.

Regular minimality/maximality is a fundamental property of discrimination and means that

1. every row contains a single minimal/maximal entry;
2. every column contains a single minimal/maximal entry;
3. an entry \( p_{ij} \) of \( X \) which is minimal/maximal in the \( i \)th row is also minimal/maximal in the \( j \)th column, and vice versa.
If \( p_{ij} \) is the entry which is minimal/maximal in the \( i \)th row and in the \( j \)th column, the \( i \)th row object (in one, the first, observation area) and the \( j \)th column object (in the other, the second, observation area) are called each other’s PSEs. In psychophysical applications, for instance, observation area refers to the two fixed and perceptually distinct areas in which the stimuli are presented pairwise; for example, spatial arrangement (left versus right) or temporal order (first versus second).

**Value**

If the data do satisfy regular minimality/maximality, `check.regular` returns a named list consisting of the following four components:

- **canonical.representation**
  a matrix giving the representation of \( X \) in which regular minimality/maximality is satisfied in the canonical form. That is, the single minimal/maximal entries of the rows and columns lie on the main diagonal (of the canonical representation). In addition, the rows and columns are canonically relabeled.

- **canonical.transformation**
  a data frame giving the permutation of the columns of \( X \) used to produce the canonical representation of \( X \). The first and second variables of this data frame, `observation.area.1` and `observation.area.2`, respectively, represent the pairs of PSEs. The third variable, `common.label`, lists the identical labels assigned to the pairs of PSEs.

- **check**
  a character string giving the check that was performed. This is either "regular minimality" or "regular maximality".

- **in.canonical.form**
  logical. If TRUE, the permutation of the columns used to obtain the canonical representation of \( X \) is the identity; that is, the original data \( X \) are already in the canonical form.

**Author(s)**

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

**References**


**See Also**

`check.data` for checking data format; `fechner`, the main function for Fechnerian scaling. See also `fechner-package` for general information about this package.
Examples

## dataset \link{wish} satisfies regular minimality in canonical form
check.regular(wish)

## dataset \link{regMin} satisfies regular minimality in non-canonical form and so is canonically transformed and relabeled
regMin
check.regular(regMin)

## dataset \link{noRegMin} does satisfy neither regular minimality nor regular maximality
check.regular(noRegMin, type = "probability.different")
check.regular(noRegMin, type = "reg.maximal")

## dataset \link{morse} satisfies regular maximality in canonical form
check.regular(morse, type = "percent.same")
## part of \link{morse} data satisfies regular maximality
check.regular(morse[c(2, 27:36), c(2, 27:36)], type = "reg.maximal")

---

**fechner**  

**Main Function For Fechnerian Scaling**

**Description**

fechner provides the Fechnerian scaling computations. It is the main function of this package.

**Usage**

fechner(\textbf{X}, \textbf{format} = \texttt{c("probability.different", "percent.same", "general")},
\textbf{compute.all} = \texttt{FALSE}, \textbf{check.computation} = \texttt{FALSE})

**Arguments**

\textbf{X} a required square matrix or data frame of numeric data. No NA, NaN, Inf, or -Inf values are allowed.

\textbf{format} an optional character string giving the data format that is used. This must be one of "probability.different", "percent.same", or "general", with default "probability.different", and may be abbreviated to a unique prefix.

\textbf{compute.all} an optional logical. The default value FALSE corresponds to short computation, which yields the main Fechnerian scaling computations. The value TRUE corresponds to long computation, which additionally yields intermediate results and also allows for a check of computations if check.computation is set TRUE.

\textbf{check.computation} an optional logical. If TRUE, the check for whether the overall Fechnerian distance of the first kind (in the first observation area) is equal to the overall Fechnerian distance of the second kind (in the second observation area) is performed. The check requires compute.all to be set TRUE.
Details

The format argument specifies the data format that is used. "probability.different" and "percent.same" are for datasets in the probability-different and percent-same formats, and in the latter case, the data are automatically transformed prior to the analysis using the transformation \((100 - X)/100\). "general" is to be used for datasets that are properly in the general data format. Note that for "percent.same", the data must satisfy regular maximality, for "probability.different" and "general", regular minimality (otherwise function fechner produces respective messages). In particular, data in the general format may possibly need to be transformed manually prior to calling the function fechner.

If compute.all = TRUE and check.computation = TRUE, the performed check computes the difference ‘overall Fechnerian distance of the first kind minus overall Fechnerian distance of the second kind’. By theory, this difference is zero. The function fechner calculates that difference and checks for equality of these Fechnerian distances up to machine precision (see ‘Value’). fechner calls check.regular, which in turn calls check.data. In particular, the specified data format and regular minimality/maximality are checked, and the rows and columns of the canonical representation matrix (see check.regular) are canonically relabeled based on the labeling provided by check.data.

The function fechner returns an object of the class fechner (see ‘Value’), for which plot, print, and summary methods are provided; plot.fechner, print.fechner, and summary.fechner, respectively. Moreover, objects of the class fechner are set the specific named attribute computation, which is assumed to have the value short or long indicating whether short computation (compute.all = FALSE) or long computation (compute.all = TRUE) was performed, respectively.

Value

If the arguments X, format, compute.all, and check.computation are of required types, fechner returns a named list, of the class fechner and with the attribute computation, which consists of 6 or 18 components, depending on whether short computation (computation is then set short) or long computation (computation is then set long) was performed, respectively.

The short computation list contains the following first 6 components, the long computation list the subsequent ones:

- **points.of.subjective.equality**
  - a data frame giving the permutation of the columns of X used to produce the canonical representation of X. The first and second variables of this data frame, observation.area.1 and observation.area.2, respectively, represent the pairs of points of subjective equality (PSEs). The third variable, common.label, lists the identical labels assigned to the pairs of PSEs. (first component of short computation list)

- **canonical.representation**
  - a matrix giving the representation of X in which regular minimality/maximality is satisfied in the canonical form. That is, the single minimal/maximal entries of the rows and columns lie on the main diagonal (of the canonical representation). In addition, the rows and columns are canonically relabeled.

- **overall.Fechnerian.distances**
  - a matrix of the overall Fechnerian distances (of the first kind); by theory, invariant from observation area.
geodesic.loops a data frame of the geodesic loops of the first kind; must be read from left to right for the first kind, and from right to left for the second kind.

graph.lengths.of.geodesic.loops
a matrix of the graph-theoretic (edge/link based) lengths of the geodesic loops (of the first kind).

S.index a matrix of the generalized ‘Shepardian’ dissimilarity (or S-index) values. An S-index value is defined as the psychometric length of the loop between a row stimulus and a column stimulus containing only these two stimuli. (last component of short computation list)

points.of.subjective.equality
the same as in case of short computation; see above. (first component of long computation list)

canonical.representation
the same as in case of short computation; see above.

psychometric.increments.1
a matrix of the psychometric increments of the first kind.

psychometric.increments.2
a matrix of the psychometric increments of the second kind.

oriented.Fechnerian.differences.1
a matrix of the oriented Fechnerian distances of the first kind.

overall.Fechnerian.differences.1
a matrix of the overall Fechnerian distances of the first kind.

oriented.Fechnerian.differences.2
a matrix of the oriented Fechnerian distances of the second kind.

overall.Fechnerian.differences.2
a matrix of the overall Fechnerian distances of the second kind.

check if check.computation = TRUE, a list of two components: difference and are.nearly.equal. The component difference is a matrix of the differences of the overall Fechnerian distances of the first and second kind; ought to be a zero matrix. The component are.nearly.equal is a logical indicating whether this matrix of differences is equal to the zero matrix up to machine precision. If check.computation = FALSE, a character string saying “computation check was not requested”.

geodesic.chains.1
a data frame of the geodesic chains of the first kind.

geodesic.loops.1
a data frame of the geodesic loops of the first kind.

graph.lengths.of.geodesic.chains.1
a matrix of the graph-theoretic (edge/link based) lengths of the geodesic chains of the first kind.

graph.lengths.of.geodesic.loops.1
a matrix of the graph-theoretic (edge/link based) lengths of the geodesic loops of the first kind.

geodesic.chains.2
a data frame of the geodesic chains of the second kind.
geodesic.loops.2
   a data frame of the geodesic loops of the second kind.

graph.lengths.of.geodesic.chains.2
   a matrix of the graph-theoretic (edge/link based) lengths of the geodesic chains
   of the second kind.

graph.lengths.of.geodesic.loops.2
   a matrix of the graph-theoretic (edge/link based) lengths of the geodesic loops
   of the second kind.

S.index the same as in case of short computation; see above. (last component of long
   computation list)

Author(s)

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

References

Dzhafarov, E. N. and Colonius, H. (2006) Reconstructing distances among objects from their dis-

Journal of Mathematical Psychology, 51, 290–304.


See Also

check.data for checking data format; check.regular for checking regular minimality/maximality;
plot.fechner, the S3 method for plotting objects of the class fechner; print.fechner, the S3
method for printing objects of the class fechner; summary.fechner, the S3 method for summariz-
ing objects of the class fechner, which creates objects of the class summary.fechner; print.summary.fechner,
the S3 method for printing objects of the class summary.fechner. See also fechner-package for
general information about this package.

Examples

##
## (1) examples based on dataset \link{morse}
##
#
## dataset \link{morse} satisfies regular maximality in canonical form
morse
check.regular(morse, type = "percent.same")

# a self-contained 10-code subspace consisting of the codes for the
# letter B and the digits 0, 1, 2, 4, ..., 9
indices <- which(is.element(names(morse), c("B", c(0, 1, 2, 4:9))))
f.scal.morse <- fechner(morse, format = "percent.same")
f.scal.morse$geodesic.loops[indices, indices]
morse.subspace <- morse[indices, indices]
check.regular(morse.subspace, type = "percent.same")
## Fechnerian scaling analysis using short computation

```r
f.scal.subspace.mo <- fechner(morse.subspace, format = "percent.same")
identical(f.scal.morse$geodesic.loops[indices, indices],
         f.scal.subspace.mo$geodesic.loops)
identical(f.scal.morse$overall.Fechnerian.distances[indices, indices],
         f.scal.subspace.mo$overall.Fechnerian.distances)
```

## Fechnerian scaling analysis using long computation

```r
f.scal.subspace.long.mo <- fechner(morse.subspace, 
                                 format = "percent.same", 
                                 compute.all = TRUE, 
                                 check.computation = TRUE)
```

## Check whether the overall Fechnerian distance of the first kind is
## equal to the overall Fechnerian distance of the second kind
## the difference, by theory a zero matrix

```r
f.scal.subspace.long.mo$check[1]
```

## or, up to machine precision

```r
f.scal.subspace.long.mo$check[2]
```

## Plot of the S-index versus the overall Fechnerian distance
## for all (off-diagonal) pairs of stimuli

```r
plot(f.scal.subspace.long.mo)
```

## for all (off-diagonal) pairs of stimuli with geodesic loops
## containing at least 3 links

```r
plot(f.scal.subspace.long.mo, level = 3)
```

## Corresponding summaries, including Pearson correlation and C-index
## in particular, accessing detailed summary through assignment

```r
summary(f.scal.subspace.long.mo)
detailed.summary.mo <- summary(f.scal.subspace.long.mo, level = 3)
str(detailed.summary.mo)
```

## (2) examples based on dataset \link{wish}

## dataset \link{wish} satisfies regular minimality in canonical form

```r
check.regular(wish, type = "probability.different")
```
```r
## a self-contained 10-code subspace consisting of S, U, W, X,
## 0, 1, ..., 5
d <- which(is.element(names(wish), c("S", "U", "W", "X", 0:5)))
f.scal.wish <- fechner(wish, format = "probability.different")
f.scal.wish$geodesic.loops[indices, indices]
wish.subspace <- wish[indices, indices]
check.regular(wish.subspace, type = "probability.different")

## since the subspace is self-contained, results must be the same
f.scal.subspace.wi <- fechner(wish.subspace,
   format = "probability.different")
identical(f.scal.wish$geodesic.loops[indices, indices],
   f.scal.subspace.wi$geodesic.loops)
identical(f.scal.wish$overall.Fechnerian.distances[indices, indices],
   f.scal.subspace.wi$overall.Fechnerian.distances)

## dataset \url{wish} transformed to percent-same format
check.data(100 - (wish * 100), format = "percent.same")

## Fechnerian scaling analysis using short computation
f.scal.subspace.wi
str(f.scal.subspace.wi)
attributes(f.scal.subspace.wi)

## for instance, the graph-theoretic lengths of geodesic loops
f.scal.subspace.wi$graph.lengths.of.geodesic.loops

## Fechnerian scaling analysis using long computation
f.scal.subspace.long.wi <- fechner(wish.subspace,
   format = "probability.different",
   compute.all = TRUE,
   check.computation = TRUE)

f.scal.subspace.long.wi
str(f.scal.subspace.long.wi)
attributes(f.scal.subspace.long.wi)

## for instance, the oriented Fechnerian distances of the first kind
f.scal.subspace.long.wi$oriented.Fechnerian.distances.1

## or, graph-theoretic lengths of chains and loops
identical(f.scal.subspace.long.wi$graph.lengths.of.geodesic.chains.1 +
   t(f.scal.subspace.long.wi$graph.lengths.of.geodesic.chains.1),
   f.scal.subspace.long.wi$graph.lengths.of.geodesic.loops.1)

## overall Fechnerian distances are not monotonically related to
## discrimination probabilities; however, there is a strong positive
## correlation

cor(as.vector(f.scal.wish$overall.Fechnerian.distances),
   as.vector(as.matrix(wish)))

## check whether the overall Fechnerian distance of the first kind is
## equal to the overall Fechnerian distance of the second kind
## the difference, by theory a zero matrix
f.scal.subspace.long.wi$check[1]
## or, up to machine precision
```
Rothkopf’s (1957) Morse Code Data

Description

Rothkopf’s (1957) Morse code data of discrimination probabilities among 36 auditory Morse code signals for the letters A, B, . . . , Z and the digits 0, 1, . . . , 9.

Usage

morse

Format

The morse data frame consists of 36 rows and 36 columns, representing the Morse code signals for the letters and digits A, . . . , Z, 0, . . . , 9 presented first and second, respectively. Each number, an integer, in the data frame gives the percentage of subjects who responded ‘same’ to the row signal followed by the column signal.

Details

Each signal consists of a sequence of dots and dashes. A chart of the Morse code letters and digits can be found at http://en.wikipedia.org/wiki/Morse_code.

Rothkopf’s (1957) 36 × 36 Morse code data gives the same-different judgements of 598 subjects in response to the 36 × 36 auditorily presented pairs of Morse codes. Subjects who were not familiar with Morse code listened to a pair of signals constructed mechanically and separated by a pause of approximately 1.4 seconds. Each subject was required to state whether the two signals presented were the same or different. Each number in the morse data frame is the percentage of roughly 150 subjects.
Note

The original Rothkopf’s (1957) 36 × 36 dataset does not satisfy regular maximality. There are two maximal entries in row #2, of value 84, which are $p_{BB}$ and $p_{BX}$. Following the argument in Dzhafarov and Colonius (2006), a statistically compatible dataset is obtained by replacing the value of $p_{BX}$ with 83 and leaving the rest of the data unchanged. The latter is the dataset accompanying the package fechner.

For typographic reasons, it may be useful to consider only a small subset of the stimulus set, best, chosen to form a ‘self-contained’ subspace: a geodesic loop for any two of the subset’s elements (computed using the complete dataset) is contained within the subset. For instance, a particular self-contained 10-code subspace of the 36 Morse codes consists of the codes for the letter $B$ and the digits 0, 1, 2, 4, . . . , 9 (see fechner).

Source


References


See Also

check.data for checking data format; check.regular for checking regular minimality/maximality; fechner, the main function for Fechnerian scaling. See also wish for Wish’s Morse-code-like data, and fechner-package for general information about this package.

---

noRegMin

Artificial Data: Regular Minimality Violated

Description

Artificial data of fictitious ‘discrimination probabilities’ among 10 fictitious stimuli.

Usage

noRegMin
Format

The `noRegMin` data frame consists of 10 rows and 10 columns, representing the fictitious stimuli presented in the first and second observation area, respectively. Each number, a numeric, in the data frame is assumed to give the relative frequency of perceivers scoring ‘different’ to the row stimulus ‘followed’ by the column stimulus.

Note

This dataset is artificial and included for illustrating regular minimality being violated. It differs from the artificial data `regMin` only in the entry in row #9 and column #10.

References


See Also

`regMin` for the other artificial data satisfying regular minimality in non-canonical form; `check.data` for checking data format; `check.regular` for checking regular minimality/maximality; `fechner`, the main function for Fechnerian scaling. See also `morse` for Rothkopf’s Morse code data, `wish` for Wish’s Morse-code-like data, and `fechner-package` for general information about this package.

Examples

```r
## dataset noRegMin violates regular minimality
noRegMin
check.regular(noRegMin, type = "reg.minimal")
```

---

### S3 method to plot objects of the class `fechner`

#### Usage

```r
## S3 method for class 'fechner'
plot(x, level = 2, ...)```

Arguments

x a required object of class fechner, obtained from a call to the function fechner.

level an optional numeric, integer-valued and greater than or equal to 2, giving the level of comparison of the S-index and the overall Fechnerian distance G.

... further arguments to be passed to or from other methods. They are ignored in this function.

Details

The plot method graphs the results obtained from Fechnerian scaling analyses. It produces a scatterplot of the overall Fechnerian distance $G$ versus the $S$-index, with rugs added to the axes and jittered (amount = 0.01 of noise) to accommodate ties in the $S$-index and $G$ values. The diagonal line $y = x$ is for visual inspection of the deviations of the two types of values.

The level of comparison refers to the minimum number of links in geodesic loops. That is, choosing level $n$ means that comparison involves only those $S$-index and $G$ values that have geodesic loops containing not less than $n$ links.

If there are no (off-diagonal) pairs of stimuli with geodesic loops containing at least level links (in this case a plot is not possible), plot.fechner stops with an error message.

Value

If the arguments x and level are of required types, and if there are (off-diagonal) pairs of stimuli with geodesic loops containing at least level links, plot.fechner produces a plot, and invisibly returns NULL.

Author(s)

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

References


See Also

print.fechner, the S3 method for printing objects of the class fechner; summary.fechner, the S3 method for summarizing objects of the class fechner, which creates objects of the class summary.fechner; print.summary.fechner, the S3 method for printing objects of the class summary.fechner; fechner, the main function for Fechnerian scaling, which creates objects of the class fechner. See also fechner-package for general information about this package.
print.fechner

Examples

```r
## Fechnerian scaling of dataset \link{wish}
f.scal.wish <- fechner(wish)

## results are plotted for comparison levels 2 and 5
plot(f.scal.wish)
plot(f.scal.wish, level = 5)
```

print.fechner  

Print Method for Objects of Class fechner

Description

S3 method to print objects of the class fechner.

Usage

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

Arguments

- `x` a required object of class fechner, obtained from a call to the function fechner.
- `...` further arguments to be passed to or from other methods. They are ignored in this function.

Details

The print method prints the main results obtained from Fechnerian scaling analyses, which are the overall Fechnerian distances and the geodesic loops.

Value

If the argument `x` is of required type, print.fechner prints the overall Fechnerian distances and the geodesic loops, and invisibly returns `x`.

Author(s)

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

References


See Also

plot.fechner, the S3 method for plotting objects of the class fechner; summary.fechner, the S3 method for summarizing objects of the class fechner, which creates objects of the class summary.fechner; print.summary.fechner, the S3 method for printing objects of the class summary.fechner; fechner, the main function for Fechnerian scaling, which creates objects of the class fechner. See also fechner-package for general information about this package.

Examples

## Fechnerian scaling of dataset \link{wish}
## overall Fechnerian distances and geodesic loops are printed
(f.scal.wish <- fechner(wish))

---

print.summary.fechner  Print Method for Objects of Class summary.fechner

Description

S3 method to print objects of the class summary.fechner.

Usage

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

Arguments

x  a required object of class summary.fechner, obtained from a call to the function summary.fechner (through generic function summary).

...  further arguments to be passed to or from other methods. They are ignored in this function.

Details

The print method prints the summary information about objects of the class fechner computed by summary.fechner, which are the number of stimuli pairs used for comparison, a summary of the corresponding \( S \)-index values, a summary of the corresponding Fechnerian distance \( G \) values, the Pearson correlation, the \( C \)-index, and the comparison level. Specific summary information details such as individual stimuli pairs and their corresponding \( S \)-index and \( G \) values can be accessed through assignment (see 'Examples').

Value

If the argument x is of required type, print.summary.fechner prints the afore mentioned summary information in 'Details', and invisibly returns x.
Author(s)

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

References


See Also

plot.fechner, the S3 method for plotting objects of the class fechner; print.fechner, the S3 method for printing objects of the class fechner; summary.fechner, the S3 method for summarizing objects of the class fechner, which creates objects of the class summary.fechner; fechner, the main function for Fechnerian scaling, which creates objects of the class fechner. See also fechner-package for general information about this package.

Examples

## Fechnerian scaling of dataset \link{morse}
## summary information about the Fechnerian scaling object are printed
## accessing detailed summary through assignment
(detailed.summary <- summary(fechner(morse, format = "percent.same")))
str(detailed.summary)
detailed.summary$pairs.used.for.comparison[3, ]

---

regMin

**Artificial Data: Regular Minimality In Non-canonical Form**

Description

Artificial data of fictitious ‘discrimination probabilities’ among 10 fictitious stimuli.

Usage

regMin

Format

The regMin data frame consists of 10 rows and 10 columns, representing the fictitious stimuli presented in the first and second observation area, respectively. Each number, a numeric, in the data frame is assumed to give the relative frequency of perceivers scoring ‘different’ to the row stimulus ‘followed’ by the column stimulus.
Note
This dataset is artificial and included for illustrating regular minimality in the non-canonical form. It differs from the artificial data noRegMin only in the entry in row \#9 and column \#10.

References


See Also
noRegMin for the other artificial data violating regular minimality; check.data for checking data format; check.regular for checking regular minimality/maximality; fechner, the main function for Fechnerian scaling. See also morse for Rothkopf's Morse code data, wish for Wish's Morse-code-like data, and fechner-package for general information about this package.

Examples
```r
## dataset regMin satisfies regular minimality in non-canonical form
regMin
check.regular(regMin, type = "reg.minimal")
```

---

**summary.fechner** Summary Method for Objects of Class fechner

Description
S3 method to summarize objects of the class fechner.

Usage
```r
## S3 method for class 'fechner'
summary(object, level = 2, ...)
```

Arguments
- `object` a required object of class fechner, obtained from a call to the function fechner.
- `level` an optional numeric, integer-valued and greater than or equal to 2, giving the level of comparison of the S-index and the overall Fechnerian distance \( G \).
- `...` further arguments to be passed to or from other methods. They are ignored in this function.
summary.fechner

Details

The summary method outlines the results obtained from Fechnerian scaling analyses. It computes the Pearson correlation coefficient and the C-index (see Uenlue, Kiefer, and Dzhafarov (2009))

\[ C = \frac{2 \sum (S - G)^2}{\sum S^2 + \sum G^2} \]

for specific (controlled by the argument level) stimuli pairs with their corresponding S-index and G values.

The level of comparison refers to the minimum number of links in geodesic loops. That is, choosing level \( n \) means that comparison involves only those S-index and G values that have geodesic loops containing not less than \( n \) links.

If there are no (off-diagonal) pairs of stimuli with geodesic loops containing at least level links (in this case a summary is not possible), summary.fechner stops with an error message.

The function summary.fechner returns an object of the class summary.fechner (see ‘Value’), for which a print method, print.summary.fechner, is provided. Specific summary information details such as individual stimuli pairs and their corresponding S-index and G values can be accessed through assignment (see ‘Examples’).

Value

If the arguments object and level are of required types, and if there are (off-diagonal) pairs of stimuli with geodesic loops containing at least level links, summary.fechner returns a named list, of the class summary.fechner, consisting of the following four components:

- pairs.used.for.comparison
  - a data frame giving the pairs of stimuli (first variable stimuli.pairs) and their corresponding S-index (second variable S.index) and G (third variable Fechnerian.distance.G) values used for comparison.

- Pearson.correlation
  - a numeric giving the value of the Pearson correlation coefficient if it exists, or a character string saying “Pearson’s correlation coefficient is not defined” if it does not exist.

- C.index
  - a numeric giving the value of the C-index.

- comparison.level
  - a numeric giving the level of comparison used.

Author(s)

Thomas Kiefer, Ali Uenlue. Based on original MATLAB source by Ehtibar N. Dzhafarov.

References


See Also

`plot.fechner`, the S3 method for plotting objects of the class `fechner`; `print.fechner`, the S3 method for printing objects of the class `fechner`; `print.summary.fechner`, the S3 method for printing objects of the class `summary.fechner`; `fechner`, the main function for Fechnerian scaling, which creates objects of the class `fechner`. See also `fechner-package` for general information about this package.

Examples

```r
## Fechnerian scaling of dataset \link{wish}
f.scal.wish <- fechner(wish)

## results are summarized for comparison levels 2 and 5
summary(f.scal.wish)
summary(f.scal.wish, level = 5)

## accessing detailed summaries through assignment
str(detailed.summary.11 <- summary(f.scal.wish))
detailed.summary.15 <- summary(f.scal.wish, level = 5)
detailed.summary.15$pairs.used.for.comparison[1, ]

## to verify the obtained summaries
f.scal.wish$geodesic.loops
f.scal.wish$s.index
f.scal.wish$overall.Fechnerian.distances
```

---

### wish

**Wish’s Morse-code-like Data**

Description

Wish’s (1967) Morse-code-like data of discrimination probabilities among 32 auditory Morse-code-like signals.

Usage

`wish`

Format

The `wish` data frame consists of 32 rows and 32 columns, representing the Morse-code-like signals (see ‘Details’) presented first and second, respectively. Each number, a numeric, in the data frame gives the relative frequency of subjects who responded ‘different’ to the row signal followed by the column signal.
Details

The 32 Morse-code-like signals in Wish’s (1967) study were 5-element sequences $T_1P_1T_2P_2T_3$, where $T$ stands for a tone (short or long) and $P$ stands for a pause (1 or 3 units long). As in Dzhafarov and Colonius (2006), the stimuli are labeled $A$, $B$, $\ldots$, $Z$, 0, 1, $\ldots$, 5, in the order they are presented in Wish’s (1967) article.

Wish’s (1967) $32 \times 32$ Morse-code-like data gives the same-different judgements of subjects in response to the $32 \times 32$ auditorily presented pairs of codes.

Note

The original Wish’s (1967) $32 \times 32$ dataset does not satisfy regular minimality. There is the entry $p_{TV} = 0.03$, which is the same as $p_{VV}$ and smaller than $p_{TT} = 0.06$. Following the argument in Dzhafarov and Colonius (2006), a statistically compatible dataset is obtained by replacing the value of $p_{TV}$ with 0.07 and leaving the rest of the data unchanged. The latter is the dataset accompanying the package fechner.

For typographic reasons, it may be useful to consider only a small subset of the stimulus set, best, chosen to form a ‘self-contained’ subspace: a geodesic loop for any two of the subset’s elements (computed using the complete dataset) is contained within the subset. For instance, a particular self-contained 10-code subspace of the 32 Morse-code-like signals consists of $S$, $U$, $W$, $X$, 0, 1, $\ldots$, 5 (see fechner).

Source


References


See Also

check.data for checking data format; check.regular for checking regular minimality/maximality; fechner, the main function for Fechnerian scaling. See also morse for Rothkopf’s Morse code data, and fechner-package for general information about this package.
Index

*Topic **attribute**
  fechner, 7

*Topic **classes**
  fechner, 7

*Topic **datasets**
  morse, 13
  noRegMin, 14
  regMin, 19
  wish, 22

*Topic **hplot**
  plot.fechner, 15

*Topic **manip**
  check.data, 3
  check.regular, 5
  fechner, 7

*Topic **methods**
  plot.fechner, 15
  print.fechner, 17
  print.summary.fechner, 18
  summary.fechner, 20

*Topic **models**
  fechner, 7

*Topic **package**
  fechner-package, 2

*Topic **print**
  plot.fechner, 15
  print.fechner, 17
  print.summary.fechner, 18
  summary.fechner, 20

*Topic **univar**
  fechner, 7
  summary.fechner, 20

*Topic **utilities**
  check.data, 3
  check.regular, 5

check.data, 2, 3, 5, 6, 8, 10, 14, 15, 20, 23
check.regular, 2, 4, 5, 8, 10, 14, 15, 20, 23

fechner, 2, 4, 6, 7, 14–20, 22, 23
fechner.internal, 2
fechner-package, 2
morse, 2, 13, 15, 20, 23
noRegMin, 2, 14, 20
plot.fechner, 2, 8, 10, 15, 18, 19, 22
print.fechner, 2, 8, 10, 16, 17, 19, 22
print.summary.fechner, 2, 10, 16, 18, 18, 21, 22
regMin, 2, 15, 19
summary, 18
summary.fechner, 2, 8, 10, 16, 18, 19, 20
wish, 2, 14, 15, 20, 22

24