

Package ‘mmstat4’

September 29, 2023

Title Access to Teaching Materials from a ZIP File or GitHub

Version 0.1.6

Description Provides access to teaching materials for various statistics courses, including R programs, Shiny apps, data, and PDF/HTML documents. These materials are stored on the Internet as a ZIP file (e.g., in a GitHub repository) and can be downloaded and displayed or run locally. The content of the ZIP file is temporarily or permanently stored. By default, the package uses the GitHub repository 'sigbertklinke/mmstat4.data.' Additionally, the package includes 'association_measures.R' from the archived package 'ryouready' by Mark Heckman and some auxiliary functions.

License GPL-3

Encoding UTF-8

Imports digest, httr, tcltk, rappdirs, rio, rstudioapi, shiny, tools

Suggests knitr, rmarkdown

VignetteBuilder knitr

RoxygenNote 7.2.3

NeedsCompilation no

Author Sigbert Klinke [aut, cre] (<<https://orcid.org/0000-0003-3337-1863>>),
Jekaterina Zukovska [ctb] (<<https://orcid.org/0000-0002-7753-9210>>)

Maintainer Sigbert Klinke <sigbert@hu-berlin.de>

Repository CRAN

Date/Publication 2023-09-29 14:52:39 UTC

R topics documented:

association	2
cdf	4
defaultApp	5
getList	6
getMMstat	7
getText	7

gh	8
ghappAddin	9
ghdecompose	9
ghfile	10
ghget	10
ghlist	11
ghopenAddin	12
ghpath	13
ghquery	13
ghrepos	14
isLocal	15
normpathes	15
Rdups	16
Rlibs	16
Rsolo	17
toInt	18
toNum	18

Index	20
--------------	-----------

association	<i>Association measures</i>
-------------	-----------------------------

Description

Various association coefficients for nominal and ordinal data; the input formats follows `stats::chisq.test()`.

- concordant concordant pairs
- discordant discordant pairs
- ties.row pairs tied on rows
- ties.col pairs tied on columns
- nom.phi Phi Coefficient
- nom.cc Contingency Coefficient (Pearson's C) and Sakoda's Adjusted Pearson's C
- nom.TT Tshuprow's T (not meaningful for non-square tables)
- nom.CV Cramer's V (for 2 x 2 tables $V = \text{Phi}$)
- nom.lambda Goodman and Kruskal's Lambda with
 - lambda.cr The row variable is used as independent, the column variable as dependent variable.
 - lambda.rc The column variable is used as independent, the row variable as dependent variable.
 - lambda.symmetric Symmetric Lambda (the mean of both above).
- nom.uncertainty Uncertainty Coefficient (Theil's U) with
 - ucc.cr The row variable is used as independent, the column variable as dependent variable.

- `uc.rc` The column variable is used as independent, the row variable as dependent variable.
- `uc.symmetric` Symmetric uncertainty coefficient.
- `ord.gamma` Gamma coefficient
- `ord.tau` a vector with Kendall-Stuart Tau's
 - `tau.a` Tau-a (for quadratic tables only)
 - `tau.b` Tau-b
 - `tau.c` Tau-c
- `ord.somers.d` Somers' d
- `eta` Eta coefficient for nominal/interval data

Usage

```
concordant(x, y = NULL)
discordant(x, y = NULL)
ties.row(x, y = NULL)
ties.col(x, y = NULL)
nom.phi(x, y = NULL)
nom.cc(x, y = NULL)
nom.TT(x, y = NULL)
nom.CV(x, y = NULL)
nom.lambda(x, y = NULL)
nom.uncertainty(x, y = NULL)
ord.gamma(x, y = NULL)
ord.tau(x, y = NULL)
ord.somers.d(x, y = NULL)
eta(x, y, breaks = NULL)
```

Arguments

<code>x</code>	a numeric vector, table or matrix. <code>x</code> and <code>y</code> can also both be factors. For <code>eta</code> the independent nominal variable (factor or numeric).
<code>y</code>	a numeric vector; ignored if <code>x</code> is a table or matrix. If <code>x</code> is a factor, <code>y</code> should be a factor of the same length. For <code>eta</code> the dependent interval variable (numeric).

breaks either a numeric vector of two or more unique cut points or a single number (greater than or equal to 2) giving the number of intervals into which x is to be cut (only for eta).

Value

the association coefficient(s)

Source

From the [archived ryouready package](#) by Mark Heckmann. The code for the calculation of `nom.lambda`, `nom.uncertainty`, `ord.gamma`, `ord.tau`, `ord.somers.d` was supplied by Marc Schwartz (under GPL 2) and checked against SPSS results.

Examples

```
## Nominal data
# remove gender from the table
hec <- apply(HairEyeColor, 1:2, sum)
nom.phi(hec)
nom.cc(hec)
nom.TT(hec)
nom.CV(hec)
nom.lambda(hec)
nom.uncertainty(hec)
## Ordinal data
# create a fake data set
ordx <- sample(5, size=100, replace=TRUE)
ordy <- sample(5, size=100, replace=TRUE)
concordant(ordx, ordy)
discordant(ordx, ordy)
ties.row(ordx, ordy)
ties.col(ordx, ordy)
ord.gamma(ordx, ordy)
ord.tau(ordx, ordy)
ord.somers.d(ordx, ordy)
## Interval/nominal data
eta(iris$Species, iris$Sepal.Length)
```

cdf

Generates and plots a cumulative distribution function.

Description

Generates and plots a cumulative distribution function.

Usage

```

cdf(x, ...)

## Default S3 method:
cdf(x, y, discrete = TRUE, ...)

## S3 method for class 'cdf'
plot(x, y, ..., col.01line = "black", pch = 19)

```

Arguments

x	numeric: x-values
...	further parameters given to graphics::plot()
y	numeric: y-values
discrete	logical: if distribution is discrete
col.01line	color: color of horizontal lines at 0 and 1 (default: black)
pch	point type: See graphics::points() for possible values and their interpretation (default: 19)

Value

returns a cdf object

Examples

```

# Binomial distribution
x <- cdf(0:10, pbinom(0:10, 10, 0.5))
plot(x)
# Exponential distribution
x <- seq(0, 5, by=0.01)
x <- cdf(x, pexp(x), discrete=FALSE)
plot(x)

```

defaultApp

defaultApp

Description

Tries to open the given file with the default application of the operating system using [base::system2\(\)](#). Only Windows (windows), macOS (darwin), Linux (linux) and FreeBSD (freebsd) is supported.

Usage

```
defaultApp(file, wait = FALSE, ...)
```

Arguments

file	character: file name
wait	logical: indicates whether the R interpreter should wait for the command to finish, or run it asynchronously (default: FALSE)
...	further arguments passed to system2

Value

Result of `try(system2, ...)`, invisibly

See Also

[berryFunctions::openFile\(\)](#)

Examples

```
if (interactive()) {  
  ghget()  
  defaultApp(ghlist("dataanalysis.pdf", full.names = TRUE))  
}
```

getList

getList

Description

Creates a list with element names replaced by `link{getText}`.

Usage

```
getList(...)
```

Arguments

...	named elements of a list
-----	--------------------------

Value

renamed list

Examples

```
getList(BOSTON=1, MTCARS=2)
```

<code>getMMstat</code>	<code><i>getMMstat</i></code>
------------------------	-------------------------------

Description

Allows to access the package internal `mmsstat` environment.

Usage

```
getMMstat(...)
```

Arguments

... elements

Value

the choosen element

Examples

```
getMMstat('version')
```

<code>getText</code>	<code><i>getText</i></code>
----------------------	-----------------------------

Description

Translates a given message into another language.

Usage

```
getText(msg)
```

Arguments

msg character vector

Value

vector of translated messages

Examples

```
getText('Test')
```

Description

gh performs the operation described below on a file `x`. A match for `x` is searched for the currently opened ZIP file. If no unique match is found, then an error is thrown. Otherwise, the following actions are performed:

- `gh(x, 'open')` or `ghopen(x)`: Opens a file in the local browser if the file extension is `html` or `pdf`, otherwise in the RStudio editor.
- `gh(x, 'load')` or `ghload(x)`: Loads the contents of a file with `import`.
- `gh(x, 'source')` or `ghsource(x)`: Executes the contents of a file with `source`.
- `gh(x, 'app')` or `ghapp(x)`: Tries to open the file with the default application of the OS, see `defaultApp()`.

Usage

```
gh(x, what = c("open", "load", "source", "app"), ...)
```

```
ghopen(x, ...)
```

```
ghload(x, ...)
```

```
ghsource(x, ...)
```

```
ghapp(x, ...)
```

Arguments

<code>x</code>	character(1): name of the file, app or data set
<code>what</code>	character or function: a name of a predefined function or another function. The function must have a formal parameter <code>file</code> .
<code>...</code>	further parameters used in <code>utils::browseURL()</code> , <code>rstudioapi::navigateToFile()</code> , <code>rio::import()</code> , or <code>base::source()</code> .

Value

invisibly the result of `utils::browseURL`, `rstudioapi::navigateToFile()`, `rio::import()`, or `base::source()`.

Examples

```
if (interactive()) {  
  x <- ghopen("bank2.SAV")  
  x <- ghload("bank2.SAV")  
}
```



```

    str(x)
    x <- ghsources("univariate/example_ecdf.R")
  }

```

ghappAddin

ghappAddin

Description

Runs a Shiny app from the downloaded zip file.

Usage

```
ghappAddin()
```

Value

nothing

Examples

```
if (interactive()) ghappAddin()
```

ghdecompose

ghdecompose

Description

Decomposes a path of a set of files (or dirs) in several parts:

Usage

```
ghdecompose(files, dirs = FALSE)
```

Arguments

files	character vector: path of files
dirs	logical: directory or files names (default: FALSE)

Details

- `outpath` the path part which is common to all files (basically the place where the ZIP file was extracted)
- `inpath` the path part which is not necessary for a unique address in the ZIP file
- `minpath` the minimal path part such that all files are addressable in a unique manner,
- `filename` the basename of the file, and
- `source` the input to `shortpath`.

Value

a data frame with five variables

Examples

```
ghget("dummy")
pdf <- ghdecompose(ghlist(full.names=TRUE))
pdf
```

ghfile

ghfile

Description

Finds either a unique match in the list of files or throws an error with possible candidate files.

Usage

```
ghfile(x)
```

Arguments

x character: file names

Value

the full matching file

Examples

```
ghfile("data/BANK2.sav")
if (interactive()) ghfile("data/BANK2.SAV") # throws an error
```

ghget

ghget

Description

Makes a repository the active repository and downloads it if necessary. The parameter `.tempdir` is TRUE (default) then the repository is stored in the in the temporary directory `tempdir()` else in the application directory `rappdirs::user_data_dir()` for mmstat4. The parameter `.tempdir` is not logical then the value will be used as installation path.

Usage

```
ghget(..., .force = FALSE, .tempdir = TRUE, .quiet = !interactive())
```

Arguments

...	parameters to set and activate a repository
.force	logical: download and unzip in any case? (default: FALSE)
.tempdir	logical or character: store download temporary or permanently (default: <code>getOption("mmstat4.tempdir")</code>)
.quiet	logical: show repository read attempts (default: <code>!interactive()</code>) <ul style="list-style-type: none"> if <code>.tempdir==TRUE</code> then the downloaded zip file will be stored temporarily in <code>tempdir()</code> if <code>.tempdir==FALSE</code> then the downloaded zip file will be stored temporarily in <code>rappdirs::user_data_dir()</code> otherwise it is assumed that you give the name of an existing directory to store the downloaded zip file

Details

Note, the list of repository names, directories and urls is stored in the installation directory, too.

Value

the name of the current key or nothing if unsuccessful

Examples

```
if (interactive()) {
  # get one of the default ZIP file from internet
  ghget("hu.data")
  # get a locally stored zip file
  ghget(dummy2=system.file("zip", "mmstat4.dummy.zip", package="mmstat4"))
  # get from an URL
  ghget(dummy.url="https://github.com/sigbertklinke/mmstat4.dummy/archive/refs/heads/main.zip")
}
```

ghlist

ghlist

Description

Returns unique (short) names for accessing each file in the repository according to a regular expression. For details about regular expressions, see [base::regex](#).

Usage

```
ghlist(
  pattern = ".",
  ignore.case = FALSE,
  perl = FALSE,
  fixed = FALSE,
```

```

    useBytes = FALSE,
    full.names = FALSE
  )

```

Arguments

pattern	character string containing a regular expression (or character string for fixed = TRUE) to be matched in the given character vector. Coerced by as.character to a character string if possible. If a character vector of length 2 or more is supplied, the first element is used with a warning. Missing values are allowed except for regexpr, gregexpr and regexec.
ignore.case	if FALSE, the pattern matching is <i>case sensitive</i> and if TRUE, case is ignored during matching.
perl	logical. Should Perl-compatible regexps be used?
fixed	logical. If TRUE, pattern is a string to be matched as is. Overrides all conflicting arguments.
useBytes	logical. If TRUE the matching is done byte-by-byte rather than character-by-character. See ‘Details’.
full.names	logical: should full names returned instead of short names (default: FALSE)

Value

character vector of short names

Examples

```
if (interactive()) ghlist()
```

ghopenAddin

ghopenAddin

Description

A RStudio addin to open a file from the downloaded zip file.

Usage

```
ghopenAddin()
```

Value

nothing

Examples

```
if (interactive()) ghopenAddin()
```

ghpath	<i>ghpath</i>
--------	---------------

Description

Returns a path for files based on ghdecompose.

Usage

```
ghpath(df, from = c("outpath", "inpath", "minpath", "filename"))
```

Arguments

df	data frame: returned from ghdecompose
from	character: either inpath (default), outpath, minpath, or filename

Value

a character vector with file paths

Examples

```
ghget("dummy")
pdf <- ghdecompose(ghlist(full.names=TRUE))
ghpath(pdf)
ghpath(pdf, 'o') # equals the input to ghdecompose
ghpath(pdf, 'i')
ghpath(pdf, 'm')
ghpath(pdf, 'f')
```

ghquery	<i>ghquery</i>
---------	----------------

Description

Queries the unique (short) names for each file in the repository. Several query methods are available, see Details.

Usage

```
ghquery(
  query,
  n = 6,
  full.names = FALSE,
  method = c("fpdist", "overlap", "tfidf"),
  costs = NULL,
```

```

    counts = FALSE,
    useBytes = FALSE
  )

```

Arguments

query	character: query string
n	integer: maximal number of matches to return
full.names	logical: should full names used instead of short names (default: FALSE)
method	character: method to be used (default: <code>fpdist</code>)
costs	a numeric vector or list with names partially matching ‘insertions’, ‘deletions’ and ‘substitutions’ giving the respective costs for computing the Levenshtein distance, or NULL (default) indicating using unit cost for all three possible transformations.
counts	a logical indicating whether to optionally return the transformation counts (numbers of insertions, deletions and substitutions) as the “counts” attribute of the return value.
useBytes	a logical. If TRUE distance computations are done byte-by-byte rather than character-by-character.

Details

The following query methods are available:

- `fpdist` uses a partial backward matching distance based on `utils::adist()`
- `overlap` uses the **overlap distance** for query and file names

Value

character vector of short names fitting best to the query

Examples

```
if (interactive()) ghquery("bank")
```

ghrepos

ghrepos

Description

Returns the known repositories and where there are stored. If `dir==''` then `tempdir()` is used for storage.

Usage

```
ghrepos()
```

Value

a data frame with the data about the repositories

Examples

```
ghrepos()
```

<code>isLocal</code>	<i>isLocal</i>
----------------------	----------------

Description

Checks if a Shiny app runs locally or on a server

Usage

```
isLocal()
```

Value

logical

Examples

```
isLocal()
```

<code>normpathes</code>	<i>normpathes</i>
-------------------------	-------------------

Description

Returns a list with normalized pathes.

Usage

```
normpathes(x)
```

Arguments

`x` file pathes

Value

A list of the same length as `x`, the *i*-th element of which contains the vector of splits of `x[i]`.

Examples

```
normpathes("CRAN/./mmstat4/python/./ghdist.R")
```

Rdups

Rdups

Description

Computes checksums to find duplicate files.

Usage

```
Rdups(files, ...)
```

Arguments

files character: file name(s)
... further parameters given to `digest::digest()`

Value

a list of file names with the same checksum or NULL

Examples

```
if (interactive()) {  
  files <- list.files(pattern="*.R$", full.names=TRUE, recursive=TRUE)  
  Rdups(files)  
}
```

Rlibs

Rlibs

Description

Counts the number of library and require commands in the files.

Usage

```
Rlibs(files)
```

Arguments

files character: file name(s)

Value

a table how frequently the packages are called

Examples

```
if (interactive()) {  
  files <- list.files(pattern="*.R$", full.names=TRUE, recursive=TRUE)  
  Rlibs(files)  
}
```

Rsolo

Rsolo

Description

Checks whether all specified files are valid R files that can be executed independently of each other.
If an error occurs then:

Usage

```
Rsolo(files, start = 1, path = NULL, open = rstudioapi::navigateToFile, ...)
```

Arguments

files	character: file name(s)
start	integer: numeric index from which file to start (default: 1)
path	character: path to start from (default: getwd())
open	function: function or function name to call after an error occurs (default: rstudioapi::navigateToFile)
...	further parameters given to the function in open

Details

1. If open is a function name or a function with a file parameter, then Rsolo will try to open the faulty R source file, otherwise not.
2. The execution of Rsolo is stopped.

If you do not want the faulty R-file to be opened immediately, use open=0

Value

nothing

Examples

```
if (interactive()) {  
  files <- list.files(pattern="*.R$", full.names=TRUE, recursive=TRUE)  
  Rsolo(files)  
}
```

toInt	<i>toInt</i>
-------	--------------

Description

Converts *x* to an integer. If the conversion fails or the integer is outside *min* and *max* then `NA_integer_` is returned

Usage

```
toInt(x, min = -Inf, max = +Inf)
```

Arguments

<i>x</i>	input object
<i>min</i>	numeric: minimal value
<i>max</i>	numeric: maximal value

Value

a single integer value

Examples

```
toInt(3.0)
toInt("3.0")
toInt("test")
```

toNum	<i>toNum</i>
-------	--------------

Description

Converts *x* to a numeric. If the conversion fails or the value is outside *min* and *max* then `NA` is returned

Usage

```
toNum(x, min = -Inf, max = +Inf)
```

Arguments

<i>x</i>	input object
<i>min</i>	numeric: minimal value
<i>max</i>	numeric: maximal value

Value

a single integer value

Examples

```
toNum(3.0)  
toNum("3.0")  
toNum("test")
```

Index

as.character, [12](#)
association, [2](#)

base::regex, [11](#)
base::source(), [8](#)
base::system2(), [5](#)

cdf, [4](#)
concordant (association), [2](#)

defaultApp, [5](#)
defaultApp(), [8](#)
digest::digest(), [16](#)
discordant (association), [2](#)

eta (association), [2](#)

getList, [6](#)
getMMstat, [7](#)
getText, [7](#)
gh, [8](#)
ghapp (gh), [8](#)
ghappAddin, [9](#)
ghdecompose, [9](#)
ghfile, [10](#)
ghget, [10](#)
ghlist, [11](#)
ghload (gh), [8](#)
ghopen (gh), [8](#)
ghopenAddin, [12](#)
ghpath, [13](#)
ghquery, [13](#)
ghrepos, [14](#)
ghsource (gh), [8](#)
graphics::plot(), [5](#)
graphics::points(), [5](#)

isLocal, [15](#)

nom.cc (association), [2](#)
nom.CV (association), [2](#)
nom.lamba (association), [2](#)
nom.lambda (association), [2](#)
nom.phi (association), [2](#)
nom.TT (association), [2](#)
nom.uncertainty (association), [2](#)
normpaths, [15](#)

ord.gamma (association), [2](#)
ord.somers.d (association), [2](#)
ord.tau (association), [2](#)

plot.cdf (cdf), [4](#)

rappdirs::user_data_dir(), [10, 11](#)
Rdups, [16](#)
regular expression, [12](#)
rio::import(), [8](#)
Rlibs, [16](#)
Rsolo, [17](#)
rstudioapi::navigateToFile(), [8](#)

stats::chisq.test(), [2](#)

tempdir(), [10, 11](#)
ties.col (association), [2](#)
ties.row (association), [2](#)
toInt, [18](#)
toNum, [18](#)

utils::adist(), [14](#)
utils::browseURL, [8](#)
utils::browseURL(), [8](#)