Package ‘fitteR’

July 22, 2025

Type Package

Title Fit Hundreds of Theoretical Distributions to Empirical Data

Version 0.2.0

Date 2022-02-22

Author Markus Boenn

Maintainer Markus Boenn <markus.boenn.sf@gmail.com>

Description Systematic fit of hundreds of theoretical univariate distributions to empirical data via maximum likelihood estimation. Fits are reported and summarized by a data.frame, a csv file or a 'shiny' app (here with additional features like visual representation of fits). All output formats provide assessment of goodness-of-fit by the following methods: Kolmogorov-Smirnov test, Shapiro-Wilks test, Anderson-Darling test.

Depends R (>= 3.3.0), methods

Imports stats, utils, DT, shiny, dplyr, maxLik, R.utils, tools

Suggests actuar, ald, benchden, BiasedUrn, bridgedist, Davies, DiscreteInverseWeibull, DiscreteLaplace, DiscreteWeibull, emdbook, emg, EnvStats, evd, evir, ExtDist, extremefit, FAdist, FatTailsR, fBasics, fExtremes, flexsurv, gambin, gb, GenBinomApps, GeneralizedHyperbolic, gld, GLDEX, glogis, GSM, hermite, HyperbolicDist, KScorrect, loglognorm, marg, mc2d, minimax, msm, nCDunnett, NormalLaplace, normalp, ParetoPosStable, PearsonDS, poistweedie, polyaAeppli, qmap, QRM, ReIns, reliaR, Renext, revdbayes, RMKdiscrete, RMTstat, sadists, skellam, SkewHyperbolic, skewt, SMR, sn, stabledist, STAR, statmod, trapezoid, triangle, truncnorm, VarianceGamma

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-22 12:00:02 UTC

Contents

`ecdf2` *Calculate cumulative density*

Description

Calculates the cumulative density of a set of numeric values.

Usage

```
ecdf2(x, y = NULL)
```

Arguments

x	A numeric vector of which the ECDF should be calculated
y	A numeric vector. See details for explanation

Details

This function extends the functionality of of the standard implementation of ECDF. Sometimes it is desireable to get the ECDF from pre-tabulated values. For this, elements in x and y have to be linked to each other.

Value

A list

See Also

`ecdf` for the standard implementation of ECDF

Examples

```
x <- rnorm(1000)
e <- ecdf2(x)
str(e)
plot(e)
plot(e$x, e$cs)

x <- sample(1:100, 1000, replace=TRUE)
plot(ecdf2(x))
tab <- table(x)
x <- unique(x)
lines(ecdf2(x, y=tab), col="green")
```

`fitter` *Fit distributions to empirical data*

Description

Fits theoretical univariate distributions from the R universe to a given set of empirical observations

Usage

```
fitter(
  X,
  dom = "discrete",
  freq = NULL,
  R = 100,
  timeout = 5,
  posList = NULL,
  fast = TRUE
)
```

Arguments

`X`	A numeric vector
`dom`	A string specifying the domain of 'X'
`freq`	The frequency of values in 'X'. See details.
`R`	An integer specifying the number of bootstraps. See details.
`timeout`	An numeric value specifying the maximum time spend for a fit
`posList`	A list. See details.
`fast`	A logical. See details.

Details

This routine is the workhorse of the package. It takes empirical data and systematically tries to fit numerous distributions implemented in R packages to this data. Sometimes the empirical data is passed as a histogram. In this case 'X' takes the support and 'freq' takes the number of occurences of each value in 'X'. Although not limited to, this makes most sense for discrete data. If there is prior knowledge (or guessing) about candidate theoretical distributions, these can be specified by 'posList'. This parameter takes a list with names of items being the package name and items being a character vector containing names of the distribtions (with prefix 'd'). If all distributions of a package should be applied, this vector is set to `NA`. Fitting of some distributions can be very slow. They can be skipped if 'fast' is set to `TRUE`.

Value

A list serving as an unformatted report summarizing the fitting.

Note

To reduce the computational efforts, usage of the parameter 'posList' is recommended. If not specified, the function will try to perform fits to distributions from _ALL_ packages listed in supported.packages.

Author(s)

Markus Boenn

See Also

printReport for post-processing of all fits

Examples

```
# continous empirical data
x <- rnorm(1000, 50, 3)
if(requireNamespace("ExtDist")){
r <- fitter(x, dom="c", posList=list(stats=c("dexp"), ExtDist=c("dCauchy")))
}else{
r <- fitter(x, dom="c", posList=list(stats=c("dexp", "dt")))
}

# discrete empirical data
x <- rnbinom(100, 0.5, 0.2)
r <- fitter(x, dom="dis", posList=list(stats=NA))
```

---

printReport *Prepare report of fitting*

Description

Prepares a summary of the fitting as csv or shiny

Usage

```
printReport(x, file = NULL, type = "csv")
```

Arguments

x	The output of fitter
file	A character string giving the filename (including path) where the report should be printed
type	A character vector giving the desired type(s) of output

Details

The routine generates a simple csv file, which is the most useful output in terms of reusability. However, the shiny output is more powerful and provides an overview of the statistics and a figure for visual/manual exploration of the fits. Irrspective of output type being “csv” or “shiny”, the fit-table has the following format

package package name

distr name of the distribution

nargs number of parameters

args names of parameters, comma-seperated list

estimate estimated values of parameters, comma-seperated list

start start values of parameters, comma-seperated list

constraints were constraints used, logical

runtime the runtime in milliseconds

KS test statistic D of a two-sided, two-sample Kolmogorov-Smirnov test

pKS P-value of a two-sided, two-sample Kolmogorov-Smirnov test

SW test statistic of a Shapiro-Wilks test

pSW P-value of a Shapiro-Wilks test

Value

A list with items

`table`	A `data.frame` with the same formating as the resulting csv file.
`shiny`	if `"shiny" %in% type`: a shiny object

Author(s)

Markus Boenn

Examples

```
# discrete empirical data
x <- rnbinom(100, 0.5, 0.2)
r <- fitter(x, dom="dis", posList=list(stats=NA))
# create only 'shiny' app
out <- printReport(r, type="shiny")
names(out)
## Not run:  out$shiny
out <- printReport(r, type=c("csv")) # warning as 'file' is NULL,
str(out) # but table (data.frame) returned
```

pvalue2stars *Significance stars*

Description

Get stars indicating the magnitude of significance of a P-value.

Usage

```
pvalue2stars(x, ns = "")

pvalues2stars(x, ns = "")
```

Arguments

`x`	Numeric value or numeric vector, typically a P-value from a statistical test.
`ns`	A character string specifying how insignificant results should be marked. Empty string by default.

Details

While the function `pvalue2stars` accepts only a single value, the function `pvalues2stars` is a wrapper calling `pvalue2stars` for a vector. The range of x is not checked. However, a check is done, if x is numeric at all.

Value

String(s) of stars or points.

Author(s)

Markus Boenn

Examples

```
x <- runif(1, 0,1)
pvalue2stars(x)

x <- 0.5
pvalue2stars(x, ns="not signif")

x <- c(0.0023, 0.5, 0.04)
pvalues2stars(x, ns="not signif")
```

`supported.packages` *Supported packages*

Description

Get a list of currently supported packages

Usage

```
supported.packages()
```

Details

Numerous R-packages are supported, each providing a couple of theoretical statistical distributions for discrete or continuous data. Beside ordinary distributions like normal, t, exponential, ..., some packages implement more exotic distributions like truncrated alpha.

Value

A character vector

Note

Some of the distributions are redundant, i.e. they are implemented in more than one package.

Author(s)

Markus Boenn

Examples

```
sp <- supported.packages()
head(sp)
```

Index