Package 'staTools'

March 17, 2015

Title Statistical Tools for Social Network Analysis

Version 0.1.0

Author Alessandro Bessi

Maintainer Alessandro Bessi <alessandro.bessi@iusspavia.it>

Description A collection of statistical tools for social network analysis, with strong emphasis on the analysis of discrete powerlaw distributions and statistical hypothesis tests.

Depends R (>= 3.1.1)

License GPL (>= 2)

LazyData true

LinkingTo Rcpp

Imports Rcpp, VGAM, magicaxis

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-03-17 21:00:02

R topics documented:

`cdf` *Cumulative Distribution Function*

Description

Empirical Cumulative Distribution Function.

Usage

```
cdf(x)
```

Arguments

`x`	A vector of observations.

Examples

```
x = rdispl(n = 1e4, xmin = 10, alpha = 2.5, xmax = 1e5)
obs = cdf(x)$x
ecdf = cdf(x)$y
```

`data_cdf` *Data Cumulative Distribution Function*

Description

Empirical Cumulative Distribution Function of Data.

Usage

```
data_cdf(x)
```

Arguments

`x`	A vector of observations.

Examples

```
x = rdispl(n = 1e4, xmin = 10, alpha = 2.5, xmax = 1e5)
ecdf = data_cdf(x)
```

`ddispl` *Discrete Powerlaw Probability Mass Function*

Description

Probability mass function for the discrete power law distribution with parameters xmin and alpha.

Usage

```
ddispl(x, xmin, alpha, log = FALSE)
```

Arguments

`x`	Vector of quantiles.
`xmin`	The lower bound of the powerlaw distribution.
`alpha`	The scaling parameter.
`log`	Logical, whether return log values. By default is set to FALSE.

Examples

```
x = rdispl(n = 1e4, xmin = 10, alpha = 2.5)
ddispl(x, xmin = 10, alpha = 2.5, log = FALSE)
```

`displo` *Discrete Powerlaw Object*

Description

This function allows to create a discrete powerlaw object to analyze.

Usage

```
displo(x, summary = TRUE)
```

Arguments

`x`	A vector containing the observations.
`summary`	Logical, whether print a summary with some information concerning data. By default is set to TRUE.

Examples

```
data(moby)
x = moby
o = displo(x)
```

`fastsum` *Fast Sum implemented in Cpp.*

Description

Cpp function which speed up the computation of the Hurwitz zeta function.

Usage

```
fastsum(i, xmin, alpha)
```

Arguments

`i`	An integer.
`xmin`	An integer.
`alpha`	A real number greater than 1.

`getXmin` *Lower bound estimator for discrete powerlaw distributions*

Description

Lower bound estimator for discrete powerlaw distributions.

Usage

```
getXmin(o, g = 1, c = 90, k = 5, xmax = 1e+05)
```

Arguments

`o`	Discrete powerlaw object.
`g`	A guess on the true value of the lower bound.
`c`	Confidence on the guess. A value between 1 and 100. By default is set to 90.
`k`	Number of computations after a local minimum in the KS statistics is reached.
`xmax`	Max value considered in the estimation of the lower bound.

References

A. Bessi, Speeding up lower bound estimation in powerlaw distributions, arXiv

Examples

```
x = moby
o = displo(x)
est = getXmin(o)
```

`getXmin2` *Lower bound estimator for discrete powerlaw distributions*

Description

Lower bound estimator for discrete powerlaw distributions based on the distances between probability mass functions.

Usage

```
getXmin2(o, g = 1, c = 90, k = 5, xmax = 1e+05)
```

Arguments

`o`	Discrete powerlaw object.
`g`	A guess on the true value of the lower bound.
`c`	Confidence on the guess. A value between 1 and 100. By default is set to 90.
`k`	Number of computations after a local minimum in the KS statistics is reached.
`xmax`	Max value considered in the estimation of the lower bound.

References

A. Bessi, Speeding up lower bound estimation in powerlaw distributions, arXiv

Examples

```
x = moby
o = displo(x)
est = getXmin2(o)
```

inspect *Inspect Discrete Powerlaw Distributions*

Description

A graphical tool to inspect discrete powerlaw distributions.

Usage

```
inspect(o, plot = TRUE, guess = 1, showQ = FALSE, plothill = TRUE,
  summary = TRUE, xmax = 1e+05)
```

Arguments

o	A discrete powerlaw object.
plot	Logical, whether to show the plot. By default is set to TRUE.
guess	A guess on the true value of the lower bound. By default is set to 1.
showQ	Logical, whether to show the quantiles of the distribution. By default is set to FALSE.
plothill	Logical, whether to show Hill plot. By default is set to TRUE.
summary	Logical, whether to print some information about the powerlaw distribution. By default is set to TRUE.
xmax	The maximum value to consider as candidate for the lower bound.

Examples

```
x = moby
o = displo(x)
inspection = inspect(o, guess = 7)
```

len *Length of a Vector*

Description

Length function for lazy people.

Usage

```
len(x)
```

Arguments

x	A vector.

Examples

```
x = moby
n = len(x)
```

MAE *Mean Absolute Error*

Description

Mean Absolute Error.

Usage

```
MAE(x, y)
```

Arguments

x,y	Two vectors of the same length.

Examples

```
x = runif(10)
y = runif(10)
MAE(x,y)
```

MAPE *Mean Absolute Percentage Error*

Description

Mean Absolute Percentage Error.

Usage

```
MAPE(x, y)
```

Arguments

x,y	Two vectors of the same length.

Examples

```
x = runif(10)
y = runif(10)
MAPE(x,y)
```

moby *Moby Dick word count*

Description

The frequency of occurrence of unique words in the novel Moby Dick by Herman Melville.

Usage

```
moby
```

Format

A vector.

Source

M. E. J. Newman, Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46, 323 (2005)

MPE *Mean Percentage Error*

Description

Mean Percentage Error.

Usage

```
MPE(x, y)
```

Arguments

x,y	Two vectors of the same length.

Examples

```
x = runif(10)
y = runif(10)
MPE(x,y)
```

MSE *Mean Squared Error*

Description

Mean Squared Error.

Usage

```
MSE(x, y)
```

Arguments

x,y	Two vectors of the same length.

Examples

```
x = runif(10)
y = runif(10)
MSE(x,y)
```

pdispl *Discrete Powerlaw Distribution Function*

Description

Complementary cumulative distribution function for the discrete power law distribution with parameters xmin and alpha.

Usage

```
pdispl(q, xmin, alpha, lower.tail = TRUE)
```

Arguments

q	Vector of quantiles.
xmin	The lower bound of the powerlaw distribution.
alpha	The scaling parameter.
lower.tail	Logical, whether is returned the cumulative distribution function insted of the complementary cumulative distribution function. By default is set to TRUE.

Examples

```
x = rdispl(n = 1e4, xmin = 10, alpha = 2.5)
pdispl(x, xmin = 10, alpha = 2.5, lower.tail = TRUE)
```

plotfit *Plot Fit of Discrete Powerlaw Distributions*

Description

Plot fit of discrete powerlaw distributions.

Usage

```
plotfit(o, xmax = 1e+05)
```

Arguments

o	A discrete powerlaw object.
xmax	The maximum value to show.

Examples

```
x = moby
o = displo(x)
getXmin(o)
plotfit(o)
```

plothill *Hill plot*

Description

Hill plot for discrete power law distributions.

Usage

```
plothill(o, gxmin = 0, xmax = 1e+05)
```

Arguments

o	A discrete powerlaw object.
gxmin	Guess on the true value of the lower bound.
xmax	Maximum value considered as candidate for the lower bound. Default is set to 1e5.

Examples

```
x = moby
o = displo(x)
plothill(o)
```

plotmultifit *Plot Multiple Fit of Discrete Powerlaw Distributions*

Description

Plot multiple fit of discrete powerlaw distributions.

Usage

```
plotmultifit(o)
```

Arguments

o	A discrete powerlaw object.

Examples

```
x = moby
o = displo(x)
getXmin(o)
plotmultifit(o)
```

plotols *Plot OLS Fit of Discrete Powerlaw Distributions*

Description

Plot OLS fit of discrete powerlaw distributions.

Usage

```
plotols(o)
```

Arguments

o	A discrete powerlaw object.

Examples

```
x = moby
o = displo(x)
plotols(o)
```

pmf *Probability Mass Function*

Description

Empirical Probability Mass Function.

Usage

```
pmf(x)
```

Arguments

x	A vector of observations.

Examples

```
x = rdispl(n = 1e4, xmin = 10, alpha = 2.5, xmax = 1e5)
obs = pmf(x)$x
probs = pmf(x)$y
```

rdispl *Discrete Powerlaw Random Generator*

Description

Random generator of discrete power law distribution with parameters xmin and alpha.

Usage

```
rdispl(n, xmin, alpha, xmax = 1e+05)
```

Arguments

n	Number of observations.
xmin	The lower bound of the powerlaw distribution.
alpha	The scaling parameter.
xmax	The maximum value generated.

Examples

```
x = rdispl(n = 1e4, xmin = 10, alpha = 2.5, xmax = 1e5)
```

RMSE *Root Mean Squared Error*

Description

Root Mean Squared Error.

Usage

```
RMSE(x, y)
```

Arguments

x,y	Two vectors of the same length.

Examples

```
x = runif(10)
y = runif(10)
RMSE(x,y)
```

std *Unity-based Normalization*

Description

Unity-based normalization of a vector.

Usage

```
std(x)
```

Arguments

x	A vector to normalize.

Examples

```
x = moby
z = std(x)
```

Index