Package ‘BenfordTests’

May 14, 2013

Type Package

Title Statistical Tests for Evaluating Conformity to Benford’s Law

Version 1.0.0

Date 2013-05-14

Depends R (>= 3.0.0)

Author Dieter William Joenssen, with contributions from Thomas Muellerleile

Maintainer Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

Description This package contains seven specialized statistical tests and support functions for determining if numerical data could conform to Benford’s law.

License GPL-3

Archs i386, x64

R topics documented:

`BenfordTests-package` *Statistical Tests for Benford's Law.*

Description

This package contains seven specialized statistical tests and support functions for determining if numerical data could conform to Benford's law.

Details

Package:	BenfordTests
Type:	Package
Version:	1.0.0
Date:	2013-05-14
License:	GPL-3

`BenfordTests` is the implementation of the seven most commonly used goodness-of-fit (GOF) tests to assess if data conforms to Benford's law.
Tests include:
Pearson *chi-square* statistic (Pearson (1900))
Kolmogorov-Smirnov *D* statistic (Kolmogorov (1933))
Freedman's modification of Watson's *U-square* statistic (Freedman (1981), Watson (1961))
Chebyshev distance *m* statistic (Leemis (2000))
Euclidean distance *d* statistic (Cho and Gaines (2007))
Judge-Schechter mean deviation *a-star* statistic (Judge and Schechter (2009))
Joenssen's *JP-square* statistic, a Shapiro-Francia type correlation test (Shapiro and Francia (1972))

All tests may be performed using more than one leading digit. All tests simulate the specific p-values required for statistical inference, while p-values for the *chi-square* and *D* statistics may also be determined using their asymptotic distributions.

Author(s)

Dieter William Joenssen

Maintainer: Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Cho WKT, Gaines BJ. Breaking the (Benford) law: Statistical fraud detection in campaign finance. The American Statistician. 2007;61(4):218-223.

Freedman LS. Watson's Un2 statistic for a discrete distribution. Biometrika. 1981;68(3):708-711.

Judge G, Schechter L. Detecting problems in survey data using Benford's law. Journal of Human Resources. 2009;44:1-24.

Kolmogorov AN. Sulla determinazione empirica di una legge di distibuzione. Giornale dell'Istituto Italiano degli Attuari. 1933;4:83-91.

Leemis LM, Schmeiser BW, Evans DL. Survival distributions satisfying Benford's law. The American Statistician. 2000;54(4):236-241.

Newcomb S. Note on the frequency of use of the different digits in natural numbers. American Journal of Mathematics. 1881;4(1):39-40.

Pearson K. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philosophical Magazine Series 5. 1900;50(302):157-175.

Shapiro SS, Francia RS. An approximate analysis of variance test for normality. Journal of the American Statistical Association. 1972;67:215-216.

Watson GS. Goodness-of-fit tests on a circle. Biometrika. 1961;48:109-114.

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Look at sample
X
#Look at the first digits of the sample
signifd(X)

#Perform a Chi-squared Test on the sample's first digits using defaults
chisq.benftest(X)
#p-value = 0.648
```

`chisq.benftest` *Pearsons's Chi-squared Goodness-of-Fit Test for Benford's Law*

Description

`chisq.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs Pearson's chi-square goodness-of-fit test to assert if the data conforms to Benford's law.

Usage

```
chisq.benftest(x = NULL, digits = 1, pvalmethod = "asymptotic", pvalsims = 10000)
```

Arguments

`x`	A numeric vector.
`digits`	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
`pvalmethod`	Method used for calculating the p-value. Either `"asymptotic"` or `"simulate"`.
`pvalsims`	An integer specifying the number of replicates to use if `pvalmethod = "simulate"`.

Details

A chi-square goodness-of fit test is performed on `signifd(x,digits)` versus `pbenf(digits)`. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class "`htest`" containing the following components:

`statistic`	the value of the chi-squared test statistic
`p.value`	the p-value for the test
`method`	a character string indicating the type of test performed
`data.name`	a character string giving the name of the data

Author(s)

Dieter William Joenssen `<Dieter.Joenssen@TU-Ilmenau.de>`

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Pearson K. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Philosophical Magazine Series 5. 1900;50(302):157-175.

See Also

`pbenf`

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform a Chi-squared Test on the sample's
#first digits using defaults but determine
#the p-value by simulation
chisq.benftest(X,pvalmethod ="simulate")
#p-value = 0.6401
```

`edist.benftest` *Euclidean Distance Test for Benford's Law*

Description

`edist.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs a goodness-of-fit test based on the Euclidean distance between the first digits' distribution and Benford's distribution to assert if the data conforms to Benford's law.

Usage

```
edist.benftest(x = NULL, digits = 1, pvalmethod = "simulate", pvalsims = 10000)
```

Arguments

`x`	A numeric vector.
`digits`	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
`pvalmethod`	Method used for calculating the p-value. Currently only `"simulate"` is available.
`pvalsims`	An integer specifying the number of replicates used if `pvalmethod = "simulate"`.

Details

A statistical test is performed utilizing the Euclidean distance between `signifd(x,digits)` and `pbenf(digits)`. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the Euclidean distance test statistic
`p.value`	the p-value for the test
`method`	a character string indicating the type of test performed
`data.name`	a character string giving the name of the data

Author(s)

Dieter William Joenssen `<Dieter.Joenssen@TU-Ilmenau.de>`

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Cho WKT, Gaines BJ. Breaking the (Benford) law: Statistical fraud detection in campaign finance. The American Statistician. 2007;61(4):218-223.

Morrow J. Benford's law, families of distributions and a test basis. 2010. http://www.johnmorrow.info/projects/benford/benfordMain.pdf.

See Also

pbenf

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform a Euclidean Distance Test on the
#sample's first digits using defaults
edist.benftest(X,pvalmethod ="simulate")
#p-value = 0.6085
```

---

jpsq.benftest — *Joenssen's* JP-square *Test for Benford's Law*

Description

`jpsq.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs a goodness-of-fit test based on the correlation between the first digits' distribution and Benford's distribution to assert if the data conforms to Benford's law.

Usage

```
jpsq.benftest(x = NULL, digits = 1, pvalmethod = "simulate", pvalsims = 10000)
```

Arguments

`x`	A numeric vector.
`digits`	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
`pvalmethod`	Method used for calculating the p-value. Currently only `"simulate"` is available.
`pvalsims`	An integer specifying the number of replicates used if `pvalmethod = "simulate"`.

Details

A statistical test is performed utilizing the sign-preserved squared correlation between `signifd(x,digits)` and `pbenf(digits)`. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class "`htest`" containing the following components:

`statistic`	the value of the *JP-square* test statistic
`p.value`	the p-value for the test
`method`	a character string indicating the type of test performed
`data.name`	a character string giving the name of the data

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Joenssen DW. A new test for Benford's distribution [abstract]. In: Abstract-proceedings of the 3rd joint Statistical Meeting DAGStat, March 18-22, 2013; Freiburg, Germany.

Shapiro SS, Francia RS. An approximate analysis of variance test for normality. Journal of the American Statistical Association. 1972;67:215-216.

See Also

pbenf

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform Joenssen's \emph{JP-square} Test
#on the sample's first digits using defaults
jpsq.benftest(X)
#p-value = 0.3241
```

---

ks.benftest — *Kolmogorov-Smirnov Test for Benford's Law*

Description

`ks.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs the Kolmogorov-Smirnov goodness-of-fit test to assert if the data conforms to Benford's law.

Usage

```
ks.benftest(x = NULL, digits = 1, pvalmethod = "simulate", pvalsims = 10000)
```

Arguments

`x`	A numeric vector.
`digits`	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
`pvalmethod`	Method used for calculating the p-value. Currently only `"simulate"` is available.
`pvalsims`	An integer specifying the number of replicates used if `pvalmethod = "simulate"`.

Details

A Kolmogorov-Smirnov test is performed between `signifd(x,digits)` and `pbenf(digits)`. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class "`htest`" containing the following components:

`statistic`	the value of the Kolmogorov-Smirnov D test statistic
`p.value`	the p-value for the test
`method`	a character string indicating the type of test performed
`data.name`	a character string giving the name of the data

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Kolmogorov AN. Sulla determinazione empirica di una legge di distibuzione. Giornale dell'Istituto Italiano degli Attuari. 1933;4:83-91.

See Also

`pbenf`

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform a Kolmogorov-Smirnov Test on the
#sample's first digits using defaults
ks.benftest(X)
#0.7483
```

`mdist.benftest` *Chebyshev Distance Test (maximum norm) for Benford's Law*

Description

`mdist.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs a goodness-of-fit test based on the Chebyshev distance between the first digits' distribution and Benford's distribution to assert if the data conforms to Benford's law.

Usage

```
mdist.benftest(x = NULL, digits = 1, pvalmethod = "simulate", pvalsims = 10000)
```

Arguments

`x`	A numeric vector.
`digits`	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
`pvalmethod`	Method used for calculating the p-value. Currently only `"simulate"` is available.
`pvalsims`	An integer specifying the number of replicates used if `pvalmethod = "simulate"`.

Details

A statistical test is performed utilizing the Chebyshev distance between `signifd(x,digits)` and `pbenf(digits)`. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class `"htest"` containing the following components:

`statistic`	the value of the Chebyshev distance (maximum norm) test statistic
`p.value`	the p-value for the test
`method`	a character string indicating the type of test performed
`data.name`	a character string giving the name of the data

Author(s)

Dieter William Joenssen `<Dieter.Joenssen@TU-Ilmenau.de>`

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Leemis LM, Schmeiser BW, Evans DL. Survival distributions satisfying Benford's law. The American Statistician. 2000;54(4):236-241.

Morrow J. Benford's law, families of distributions and a test basis. 2010.
http://www.johnmorrow.info/projects/benford/benfordMain.pdf.

See Also

pbenf

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform a Chebyshev Distance Test on the
#sample's first digits using defaults
mdist.benftest(X)
#p-value = 0.6421
```

---

meandigit.benftest *Judge-Schechter Mean Deviation Test for Benford's Law*

Description

`meandigit.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs a goodness-of-fit test based on the deviation in means of the first digits' distribution and Benford's distribution to assert if the data conforms to Benford's law.

Usage

```
meandigit.benftest(x = NULL, digits = 1, pvalmethod = "simulate", pvalsims = 10000)
```

Arguments

`x`	A numeric vector.
`digits`	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
`pvalmethod`	Method used for calculating the p-value. Currently only `"simulate"` is available.
`pvalsims`	An integer specifying the number of replicates used if `pvalmethod = "simulate"`.

Details

A statistical test is performed utilizing the deviation beteen the mean digit of `signifd(x,digits)` and pbenf`(digits)`. The resulting statistic is normalized to [0,1]. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class "`htest`" containing the following components:

`statistic`	the value of the *a-star* test statistic
`p.value`	the p-value for the test
`method`	a character string indicating the type of test performed
`data.name`	a character string giving the name of the data

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Judge G, Schechter L. Detecting problems in survey data using Benford's law. Journal of Human Resources. 2009;44:1-24.

See Also

pbenf

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform a Judge-Schechter Mean Deviation Test
#on the sample's first digits using defaults
meandigit.benftest(X)
#p-value = 0.1458
```

---

pbenf — *Distribution Function for Benford's Distribution*

Description

Returns the complete Benford distribution function for a given number of first digits.

Usage

```
pbenf(digits = 1)
```

Arguments

digits	An integer determining the number of first digits for which the pdf is returned, i.e. 1 for 1:9, 2 for 10:99 etc.

Value

Returns an object of class "table" containing the expected density of Benford's distribution for the given number of digits.

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

See Also

qbenf; rbenf

Examples

```
#show Benford's predictions for the frequencies of the first digit values
pbenf(1)
```

---

qbenf *Quantile Function for Benford's Distribution*

Description

Returns the complete quantile function for Benford's distribution with a given number of first digits.

Usage

```
qbenf(digits = 1)
```

Arguments

digits	An integer determining the number of first digits for which the qdf is returned, i.e. 1 for 1:9, 2 for 10:99 etc.

Value

Returns an object of class "table" containing the expected quantile function of Benford's distribution with a given number of digits.

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

See Also

pbenf; rbenf

Examples

```
qbenf(1)

qbenf(1)==cumsum(pbenf(1))
```

rbenf — *Random Sample Satisfying Benford's Law*

Description

Returns a random sample with length n satisfying Benford's law.

Usage

```
rbenf(n)
```

Arguments

n	Number of observations.

Value

Returns a random sample with length n satisfying Benford's law.

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

See Also

qbenf; pbenf

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Look at sample
X
#should be
# [1] 6.159420 1.396476 5.193371 2.064033 7.001284 5.006184
#7.950332 4.822725 3.386809 1.619609 2.080063 2.242473 1.944697 5.460581
#[15] 6.443031 2.662821 2.079283 3.703353 1.364175 3.354136
```

signifd *Leading Digits*

Description

Returns the specified number of significant digits for each element of a given vector.

Usage

```
signifd(x = NULL, digits = 1)
```

Arguments

x	A numeric vector.
digits	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.

Details

x is a numeric vector of arbitrary length. Unlike other solutions, this function will work reliably with all real numbers.

Value

Returns a vector of integers the same length as the input vector x.

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

See Also

chisq.benftest; ks.benftest; usq.benftest; mdist.benftest; edist.benftest; meandigit.benftest; jpsq.benftest

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Look at the first digits of the sample
signifd(X)
#should be:
#[1] 6 1 5 2 7 5 7 4 3 1 2 2 1 5 6 2 2 3 1 3
```

`signifd.seq` *Sequence of Possible Leading Digits*

Description

Returns a vector containing all possible significant digits for a given number of places.

Usage

```
signifd.seq(digits = 1)
```

Arguments

`digits`	An integer determining the number of first digits to be returned, i.e. 1 for 1:9, 2 for 10:99 etc.

Value

Returns an integer vector.

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

Examples

```
signifd.seq(1)
seq(from=1,to=9)==signifd.seq(1)

signifd.seq(2)
seq(from=10,to=99)==signifd.seq(2)
```

`usq.benftest` *Freedman-Watson U-squared Test for Benford's Law*

Description

`usq.benftest` takes any numerical vector reduces the sample to the specified number of significant digits and performs the Freedman-Watson test for discreet distributions between the first digits' distribution and Benford's distribution to assert if the data conforms to Benford's law.

Usage

```
usq.benftest(x = NULL, digits = 1, pvalmethod = "simulate", pvalsims = 10000)
```

Arguments

x	A numeric vector.
digits	An integer determining the number of first digits to use for testing, i.e. 1 for only the first, 2 for the first two etc.
pvalmethod	Method used for calculating the p-value. Currently only "simulate" is available.
pvalsims	An integer specifying the number of replicates used if pvalmethod = "simulate".

Details

A Freedman-Watson test for discreet distributions is performed between `signifd(x,digits)` and `pbenf(digits)`. `x` is a numeric vector of arbitrary length. Values of `x` should be continuous, as dictated by theory, but may also be integers. `digits` should be chosen so that `signifd(x,digits)` is not influenced by previous rounding.

Value

A list with class "htest" containing the following components:

statistic	the value of the *U-square* test statistic
p.value	the p-value for the test
method	a character string indicating the type of test performed
data.name	a character string giving the name of the data

Author(s)

Dieter William Joenssen <Dieter.Joenssen@TU-Ilmenau.de>

References

Benford F. The law of anomalous numbers. Proceedings of the American Philosophical Society. 1938;78:551-572.

Freedman LS. Watson's Un2 statistic for a discrete distribution. Biometrika. 1981;68(3):708-711.

Watson GS. Goodness-of-fit tests on a circle. Biometrika. 1961;48:109-114.

See Also

pbenf

Examples

```
#Set the random seed to an arbitrary number
set.seed(421)
#Create a sample satisfying Benford's law
X<-rbenf(n=20)
#Perform Freedman-Watson U-squared Test on
#the sample's first digits using defaults
usq.benftest(X)
#p-value = 0.4847
```

Index