Package ‘benford.analysis’

October 12, 2022

Type  Package
Title  Benford Analysis for Data Validation and Forensic Analytics
Version  0.1.5
Author  Carlos Cinelli
Maintainer  Carlos Cinelli <carloscinelli@hotmail.com>
Description  Provides tools that make it easier to validate data using Benford's Law.
Depends  R (>= 3.0.0)
Imports  data.table
License  GPL-3
Suggests  testthat
URL  http://github.com/carloscinelli/benford.analysis
BugReports  http://github.com/carloscinelli/benford.analysis/issues
RoxygenNote  6.1.1
NeedsCompilation  no
Repository  CRAN
Date/Publication  2018-12-21 09:20:03 UTC

R topics documented:

  benford  ......................................................... 2
  benford.analysis  ............................................. 4
  census.2000_2010  ............................................. 5
  census.2009  ................................................... 6
  chisq  .......................................................... 6
  corporate.payment  ............................................ 7
  dfactor  ......................................................... 7
  duplicatesTable  .............................................. 8
  extract.digits  ............................................... 8
  getBfd  ......................................................... 9
  getData  ....................................................... 10
Benford Analysis of a dataset

This function validates a dataset using Benford’s Law. Its main purposes are to find out where the dataset deviates from Benford’s Law and to identify suspicious data that need further verification. For a more complete example, see the package help at benford.analysis.

Usage

benford(data, number.of.digits = 2, sign = "positive", discrete=TRUE, round=3)

Arguments

data a numeric vector.
number.of.digits how many first digits to analyse.

sign The default value for sign is "positive" and it analyzes only data greater than zero. There are also the options "negative" and "both" that will analyze only negative values or both positive and negative values of the data, respectively. For large datasets with both positive and negative numbers, it is usually recommended to perform a separate analysis for each group, for the incentives to manipulate the numbers are usually different.

discrete most real data - like population numbers or accounting data - are discrete, so the default is TRUE. This parameter sets rounding to the differences of the ordered data to avoid floating point number errors in the second order distribution, that usually occurs when data is discrete and the ordered numbers are very close to each other. If your data is continuous (like a simulated lognormal) you should run with discrete = FALSE.
round it defines the number of digits that the rounding will use if discrete = TRUE.

Value

An object of class Benford containing the results of the analysis. It is a list of eight objects, namely:

info general information, including
• data.name: the name of the data used.
• n: the number of observations used.
• n.second.order: the number of observations used for second order analysis.
• number.of.digits: the number of first digits analysed.

data a data frame with:
• lines.used: the original lines of the dataset.
• data.used: the data used.
• data.mantissa: the log data’s mantissa.
• data.digits: the first digits of the data.

s.o.data a data frame with:
• data.second.order: the differences of the ordered data.
• data.second.order.digits: the first digits of the second order analysis.

bfd a data frame with:
• digits: the groups of digits analysed.
• data.dist: the distribution of the first digits of the data.
• data.second.order.dist: the distribution of the first digits of the second order analysis.
• benford.dist: the theoretical benford distribution.
• data.second.order.dist.freq: the frequency distribution of the first digits of the second order analysis.
• data.dist.freq: the frequency distribution of the first digits of the data.
• benford.dist.freq: the theoretical benford frequency distribution.
• benford.so.dist.freq: the theoretical benford frequency distribution of the second order analysis.
• data.summation: the summation of the data values grouped by first digits.
• abs.excess.summation: the absolute excess summation of the data values grouped by first digits.
• difference: the difference between the data and benford frequencies.
• squared.diff: the chi-squared difference between data and benford frequencies.
• absolute.diff: the absolute difference between data and benford frequencies.

mantissa a data frame with:
• mean.mantissa: the mean of the mantissa.
• var.mantissa: the variance of the mantissa.
• ek.mantissa: the excess kurtosis of the mantissa.
• sk.mantissa: the skewness of the mantissa.
MAD: the mean absolute deviation.
distortion.factor: the distortion factor
stats: list of "htest" class statistics:
  • chisq: Pearson’s Chi-squared test.
  • mantissa.arc.test: Mantissa Arc Test.

Examples

data(corporate.payment) #loads data
bfd.cp <- benford(corporate.payment$Amount) #generates benford object
bfd.cp #prints
plot(bfd.cp) #plots

benford.analysis

Benford Analysis for data validation and forensic analytics

Description

The Benford Analysis package provides tools that make it easier to validate data using Benford’s Law. The main purpose of the package is to identify suspicious data that need further verification.

Details

More information can be found on its help documentation.
The main function is `benford`. It generates a Benford S3 object.
The package defines S3 methods for plotting and printing Benford type objects.
After running `benford` you can easily get the "suspicious" data by using the functions: `suspectsTable`, `getSuspects`, `duplicatesTable` and `getDuplicates`. See help documentation and examples for further details.
The package also includes 6 real datasets for illustration purposes.

References


Examples

data(corporate.payment) #gets data
cp <- benford(corporate.payment$Amount, 2, sign="both") #generates benford object
cp #prints
plot(cp) #plots

head(suspectsTable(cp),10) #prints the digits by decreasing order of discrepancies

#gets observations of the 2 most suspicious groups
suspects <- getSuspects(cp, corporate.payment, how.many=2)

duplicatesTable(cp) #prints the duplicates by decreasing order

#gets the observations of the 2 values with most duplicates
duplicates <- getDuplicates(cp, corporate.payment, how.many=2)

MAD(cp) #gets the Mean Absolute Deviation
chisq(cp) #gets the Chi-squared test

digits_50_and_99 <- getDigits(cp, corporate.payment, digits=c(50, 99))

---

census.2000_2010  


Description


Format

A data frame with 3143 rows and 5 variables

References


Description

A dataset containing the population of towns and cities of the United States, as of july of 2009.

Format

A data frame with 19509 rows and 3 variables

References


chisq

Gets the Chi-squared test of a Benford object

Description

It gets the Chi-squared test for a Benford object. See the section value of benford.

Usage

chisq(bfd)

Arguments

bfd  an object of class "Benford". See benford.

Value

A list with class "htest" containing the results of the Chi-squared test.

Examples

data(census.2009) #gets data
c2009 <- benford(census.2009$pop.2009) #generates benford object
chisq(c2009) # equivalent to c2009$stats$chisq
Description

A dataset of the 2010’s payments data of a division of a West Coast utility company.

Format

A data frame with 189470 rows and 4 variables

References


dfactor

Gets the Distortion Factor of a Benford object

Description

It gets the Distortion Factor of a Benford object. See the section value of benford.

Usage

dfactor(bfd)

Arguments

bfd       an object of class "Benford". See benford.

Value

The distortion factor.

Examples

data(corporate.payment) #gets data
cp <- benford(corporate.payment$Amount) #generates benford object
dfactor(cp) # equivalent to cp$distortion.factor
duplicatesTable  
*Shows the duplicates of the data*

**Description**

It creates a data frame with the duplicates in decreasing order.

**Usage**

```r
duplicatesTable(bfd)
```

**Arguments**

- `bfd`  
an object of class "Benford". See `benford`.

**Value**

A data frame with 2 variables: number and duplicates.

**Examples**

```r
data(census.2009) #gets data
c2009 <- benford(census.2009$pop.2009) #generates benford object
duplicatesTable(c2009)
```

---

extract.digits  
*Extracts the leading digits from the data*

**Description**

It extracts the leading digits from the data.

This function is used by the main function of the package `benford` to extract the leading digits of the data.

**Usage**

```r
extract.digits(data, number.of.digits = 2,
               sign="positive", second.order = FALSE, discrete=TRUE, round=3)
```
getBfd

Arguments

- **data**: a numeric vector.
- **number.of.digits**: how many first digits to analyse.
- **sign**: The default value for sign is "positive" and it analyzes only data greater than zero. There are also the options "negative" and "both" that will analyze only negative values or both positive and negative values of the data, respectively. For large datasets with both positive and negative numbers, it is usually recommended to perform a separate analysis for each group, for the incentives to manipulate the numbers are usually different.
- **second.order**: If TRUE, the function will extract the first digits of the second order distribution.
- **discrete**: Most real data - like population numbers or accounting data - are discrete, so the default is TRUE. This parameter sets rounding to the differences of the ordered data to avoid floating point number errors in the second order distribution, that usually occurs when data is discrete and the ordered numbers are very close to each other. If your data is continuous (like a simulated lognormal) you should run with discrete = FALSE.
- **round**: it defines the number of digits that the rounding will use if discrete = TRUE and second.order = TRUE.

Value

A data.frame with the data and the first digits.

---

getBfd  
*Gets the the statistics of the first Digits of a benford object*

Description

It gets the statistics of the first digits (Frequencies, Squared Differences, Absolute Differences etc). See the section value of benford.

Usage

getBfd(bfd)

Arguments

- **bfd**: an object of class "Benford". See benford.

Value

A data.frame with first digits and their statistics.
getDigits

Examples

```r
data(corporate.payment)
cp <- benford(corporate.payment$Amount) #generates benford object
getBfd(cp) # equivalent to cp$bfd
```

getData

gets the data used of a Benford object

Description

It gets the lines, values, mantissa and first digits of the data used of a Benford object. See the section value of `benford`.

Usage

```r
data(nows.
```

Arguments

- `bfd` an object of class "Benford". See `benford`.

Value

A data.frame with the lines, values, mantissa and first digits of the data.

Examples

```r
data(corporate.payment)
cp <- benford(corporate.payment$Amount) #generates benford object
getData(cp) # equivalent to cp$data
```

getDigits

gets the data starting with some specific digits

Description

It subnets the original data according to the leading digits.

Usage

```r
data(nows.
```

Arguments

- `bfd` an object of class "Benford". See `benford`.
- `data` the original data of the analysis.
- `digits` the first digits to get.
getDuplicates

Value
The original data starting only with the leading digits.

Examples

data(census.2000_2010) # gets data

# generates benford object
c2010 <- benford(census.2000_2010$pop.2010)

# subsets data starting with digits 10 and 25
digits.10.25 <- getDigits(c2010, census.2000_2010, c(10,25))

getDuplicates

Gets the duplicates from data

Description
It gets the duplicates from the original data.

Usage
getDuplicates(bfd, data, how.many=2)

Arguments

  bfd an object of class "Benford". See benford.
  data the original data used for the benford analysis.
  how.many how many groups of duplicates to get.

Value
The duplicates from the original data.

Examples

data(census.2000_2010) # gets data
c2010 <- benford(census.2000_2010$pop.2010) # generates benford object
duplicates <- getDuplicates(c2010, census.2000_2010)
getSuspects

*Gets the 'suspicious' observations according to Benford's Law*

**Description**

It gets the original data from the 'suspicious' digits groups according to benford analysis.

**Usage**

```r
getSuspects(bfd, data, by="absolute.diff", how.many=2)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bfd</td>
<td>an object of class &quot;Benford&quot;. See <code>benford</code>.</td>
</tr>
<tr>
<td>data</td>
<td>the original data used for the benford analysis.</td>
</tr>
<tr>
<td>by</td>
<td>a character string selecting how to order the digits. It can be 'abs.excess.summation','difference','squared.diff' or 'absolute.diff'.</td>
</tr>
<tr>
<td>how.many</td>
<td>how many groups of digits to get.</td>
</tr>
</tbody>
</table>

**Value**

The 'suspicious' observations from the original data.

**Examples**

```r
data(lakes.perimeter)  # gets data
lk <- benford(lakes.perimeter[,1])  # generates benford object
suspects <- getSuspects(lk, lakes.perimeter)
```

---

**lakes.perimeter**

*Perimeter of lakes arround the world*

**Description**


**Format**

A data frame with 248607 rows and 1 variable.

**References**


**MAD**

*Gets the MAD of a Benford object*

**Description**

It gets the Mean Absolute Deviation (MAD) of a Benford object. See the section value of `benford`.

**Usage**

\[
\text{MAD}(\text{bfd})
\]

**Arguments**

- `bfd` an object of class "Benford". See `benford`.

**Value**

The MAD.

**Examples**

```r
data(census.2000_2010) # gets data
c2010 <- benford(census.2000_2010$pop.2010) # generates benford object
MAD(c2010) # equivalent to c2010$MAD
```

---

**MAD.conformity**

*MAD conformity to Benford’s Law using the MAD*

**Description**

This function checks the MAD against the conformity criteria proposed by Nigrini (2012).

**Usage**

\[
\text{MAD.conformity}(\text{MAD} = \text{NULL}, \text{digits.used} = \text{c("First Digit", "Second Digit", "First-Two Digits", "First-Three Digits")})
\]

**Arguments**

- `MAD` The mean absolute deviation, as computed in the function `benford`
- `digits.used` How many digits used in the analysis.

**Value**

A list with the MAD, `digits.used` and the conformity level.
References

mantissa

*Gets the main stats of the Mantissa of a Benford object*

**Description**
It gets the Mean, Variance, Excess Kurtosis and Skewness of the Mantissa. See the section value of `benford`.

**Usage**
mantissa(bfd)

**Arguments**
bfd an object of class "Benford". See `benford`.

**Value**
A data.frame with the main stats of the Mantissa.

**Examples**
```r
data(corporate.payment) # gets data
cp <- benford(corporate.payment$Amount) # generates benford object
mantissa(cp) # equivalent to cp$mantissa
```

marc

*Gets the Mantissa Arc test of a Benford object*

**Description**
It gets the Mantissa Arc Test of a Benford object. See the section value of `benford`.

**Usage**
marc(bfd)

**Arguments**
bfd an object of class "Benford". See `benford`. 
Value
A list with class "htest" containing the results of the Matissa Arc test.

Examples
```r
data(corporate.payment) # gets data
cp <- benford(corporate.payment$Amount) # generates benford object
marc(cp) # equivalent to cp$stats$mantissa.arc.test
```

---

**p.these.digits**

*Probability of a digit sequence*

Description
It calculates the probability of a digit sequence "d".

Usage
```r
p.these.digits(d)
```

Arguments
- **d**
  - a digit sequence, like 1234 or 999999.

Value
The probability of the sequence d.

Examples
```r
p.these.digits(1) # 0.30103
p.these.digits(11) # 0.03778856
p.these.digits(999999) # 4.342947e-07
```

---

**p.this.digit.at.n**

*Probability of a digit at the nth position*

Description
It calculates the probability of digit "d" at the "n"th position.

Usage
```r
p.this.digit.at.n(d,n)
```
Arguments

d a digit from 0 to 9 (except at position n=1, where d cannot be 0, it will give you NA).
n the nth position.

Value

The probability of d at position n.

Examples

p.this.digit.at.n(1,1) # 0.30103
p.this.digit.at.n(1,2) # 0.1138901
p.this.digit.at.n(9,3) # 0.09826716

matrix <- as.data.frame(round(sapply(1:4, function(x) sapply(0:9,p.this.digit.at.n,n=x)),5))
names(matrix) <- paste0("n=" ,1:4)
rownames(matrix) <- paste0("d=" ,0:9)
matrix # a table with the probabilities of digits 0 to 9 in positions 1 to 4.

plot.Benford

Plot method for Benford Analysis

Description

The plot method for "Benford" objects.

Usage

## S3 method for class 'Benford'
plot(x,except=c("mantissa","abs diff"), multiple=TRUE, ...)

Arguments

x a "Benford" object
except it specifies which plots are not going to be plotted. Currently, you can choose from 7 plots: "digits", "second order", "summation", "mantissa", "chi square", "abs diff", "ex summation". If you want to plot all, just put except = "none". The default is not to plot the "mantissa" and "abs diff".
multiple if TRUE, all plots are grouped in the same window.
... arguments to be passed to generic plot functions,

Value

Plots the Benford object.
print.Benford

Description

The print method for "Benford" objects.

Usage

```r
## S3 method for class 'Benford'
print(x, how.many=5, ...)
```

Arguments

- `x` a "Benford" object.
- `how.many` a number that defines how many of the biggest absolute differences to show.
- `...` arguments to be passed to generic print functions.

Value

Prints the Benford object.

sino.forest

Financial Statemens of Sino Forest Corporation’s 2010 Report

Description

Financial Statemens numbers of Sino Forest Corporation’s 2010 Report.

Format

A data frame with 772 rows and 1 variable.

References

suspectsTable  
*Shows the first digits ordered by the mains discrepancies from Benford’s Law*

**Description**

It creates a data frame with the first digits and the differences from Benford’s Law in decreasing order.

**Usage**

```r
suspectsTable(bfd, by="absolute.diff")
```

**Arguments**

- `bfd`: an object of class "Benford". See `benford`
- `by`: a character string selecting how to order the digits. It can be 'abs.excess.summation','difference','squared.diff' or 'absolute.diff'.

**Value**

A data frame with 2 variables: digits and the group chosen in `by`.

**Examples**

```r
data(corporate.payment) #gets data
cp <- benford(corporate.payment$Amount) #generates benford object
suspectsTable(cp)
```

---

**taxable.incomes.1978**  
*Taxable Income 1978*

**Description**

Taxable Incomes of the 1978 Individual Tax Model File (ITMF).

**Format**

A data frame with 157518 rows and 1 variable.

**References**

Index

* dataset
  census.2000_2010, 5
  census.2009, 6
  corporate.payment, 7
  lakes.perimeter, 12
  sino.forest, 17
  taxable.incomes.1978, 18

  benford, 2, 4, 6–14, 18
  benford.analysis, 2, 4
  benford.analysis-package
    (benford.analysis), 4

  chisq, 6
  corporate.payment, 7

  dfactor, 7
  duplicatesTable, 4, 8

  extract.digits, 8

  getBfd, 9
  getData, 10
  getDigits, 10
  getDuplicates, 4, 11
  getSuspects, 4, 12

  lakes.perimeter, 12

  MAD, 13
  MAD.conformity, 13
  mantissa, 14
  marc, 14

  p.these.digits, 15
  p.this.digit.at.n, 15
  plot.Benford, 16
  print.Benford, 17

  sino.forest, 17

  suspectsTable, 4, 18

  taxable.incomes.1978, 18