Package 'modes'

March 7, 2016

Title Find the Modes and Assess the Modality of Complex and Mixture Distributions, Especially with Big Datasets

Version 0.7.0

Date 2016-03-06

Author Sathish Deevi [aut, cre],
4D Strategies [aut,own]

Maintainer Sathish Deevi <SathishCDeevi@gmail.com>

Description Designed with a dual purpose of accurately estimating the mode (or modes) as well as characterizing the modality of data. The specific application area includes complex or mixture distributions particularly in a big data environment. The heterogeneous nature of (big) data may require deep introspective statistical and machine learning techniques, but these statistical tools often fail when applied without first understanding the data. In small datasets, this often isn't a big issue, but when dealing with large scale data analysis or big data thoroughly inspecting each dimension typically yields an O(n^n-1) problem. As such, dealing with big data require an alternative toolkit. This package not only identifies the mode or modes for various data types, it also provides a programmatic way of understanding the modality (i.e. unimodal, bimodal, etc.) of a dataset (whether it's big data or not). See <http://www.sdeevi.com/modes_package> for examples and discussion.

Depends R (>= 3.2.2)

Collate 'Utility_functions.R' 'Nonparametric_functions.R' 'Parametric_functions.R'

URL http://www.sdeevi.com/modes_package
https://github.com/sathish-deevi/modes-Package/

NeedsCompilation no

Repository CRAN

Date/Publication 2016-03-07 07:59:58

R topics documented:

`modes-package` *modes: An R package to find the modes & assess the modality of complex or mixed distributions*

Description

The R package modes was designed with a dual purpose of accurately estimating the mode (or modes) as well as characterizing the modality of data. The specific application area includes complex or mixture distibutions particularly in a big data environment. The heterogenous nature of (big) data may require deep introspective statistical and machine learning techniques, but these statistical tools often fail when applied without first understanding the data. In small datasets, this often isn't a big issue, but when dealing with large scale data analysis or big data thoroughly inspecting each dimension typically yields an O(n^n-1) problem. As such, dealing with big data require an alternative toolkit. This package not only identifies the mode or modes for various data types, it also provides a programmatic way of understanding the modality (i.e. unimodal, bimodal, etc.) of a dataset (whether it's big data or not). See <http://www.sdeevi.com/modes_package> for examples and discussion.

Details

This package was designed to find the modes and aide in assessing the modality of a dataset. It was optimized programmatically to be as efficient on big data as possible. The enclosed techniques span various fields of statistics and machine learning from exploratory data analysis, to distribution theory, to univariate & multivariate statistics as well as data munging and multi-stage machine learning.

The key functions that are included in this package include:

Nonparametric

- Mode: This function calculates the mode for an integer valued data vector by default. It also calculates "nmore" more modes than the most frequently occurring value and can take in data that should be treated as integers, real numbers (which can optionally be rounded to the "digits" number of significant digits), or factors. This mode function finds the value(s)

that occur most frequently so, crucially, if there is a tie in the frequency count for the mode it will yield two modes instead of the lower valued mode. Yielding all modes instead of just the lowest mode is particularly important when more advanced statistics and machine learning techniques are employed.

- Bimodality Amplitude: This function calculates the Bimodality Ampltiude of a data vector. This is a measure of the proportion of bimodality and the existence of bimodality. The value lies between zero and one (that is: [0,1]) where the value of zero implies that the data is unimodal and the value of one implies the data is two point masses. The proportion of bimodality here is referring to the mixture proportions of two, say, Gaussian (normal) components that can have different frequencies.
- Bimodality Coefficient: This function calculates the Bimodality Coefficient of a data vector with the option for a finite sample (bias) correction. This bias correction is important to correct for the (well-documented) finite sample bias. The bimodality coefficient has a range of zero to one (that is: [0,1]) where a value greater than "5/9" suggests bimodality. The maximum value of one ("1") can only be reached when the distribution is composed of two point masses.
- Bimodality Ratio: This function calculates the Bimodality Ratio which is a measure of the proportion of bimodality. The proportion of bimodality here is referring to the mixture proportions of two, say, Gaussian (normal) components that can have different frequencies. For instance, a 50 separation will be different from a 25

Parametric

- Ashman, Bird, and Zepf's D Statistic (Ashman's D): This function calculates a measure of how well differentiated two distributions (distribution components) are. For instance, if the two distributions are identical, this statistic is zero. A good rule of thumb is that if the statistic is above ~2, there is good separation. If you suspect that your data is bimodal this can be used by replicating the suspected mixture components and checking the statistic. Alternatively, if the components are known outright this is straightforward to implement.
- Bimodality Separation: the Bimodality Separation statistic measures how differentiated two distributions (distribution components) are. However, this statistic uses the added assumption that both are Gaussian (normal) distributions (or that the distribution is a mixture of two Gaussian (normal) components).

Author(s)

Sathish Deevi & 4D Strategies

References

Ashman, K., Bird, C., & Zepf, S. (1994). Detecting bimodality in astronomical datasets. The Astronomical Journal, 2348-2361.

Ellison, A. (1987). Effect of Seed Dimorphism on the Density-Dependent Dynamics of Experimental Populations of Atriplex triangularis (Chenopodiaceae). American Journal of Botany, 74(8), 1280-1288.

Zhang, C., Mapes, B., & Soden, B. (2003). Bimodality in tropical water vapour. Quarterly Journal of the Royal Meteorological Society, 129(594), 2847-2866.

See Also

http://www.sdeevi.com/modes_package

Examples

```
 #12 Examples of the most useful and common features of this package
#are included below.

##### Nonparametric Examples #####

### 1) Mode examples

 ##Example 1.1
 #data<-c(rep(6,9),rep(3,3))
 #mode(data,type=1,"NULL","NULL")

 ##Example 1.2
 #data<-c(rep(6,9),rep(3,9))
 #mode(data,type=1,"NULL","NULL")

 ##Example 1.3
 #data<-c(rep(6,9),rep(3,8),rep(7,7),rep(2,6))
 #mode(data,type=1,"NULL",2)

 ##Example 1.4
 #data<-c(rnorm(15,0,1),rnorm(21,5,1),rep(3,3))
 #mode(data)

 ##Example 1.5
 #data<-c(rep(6,3),rep(3,3),rnorm(15,0,1))
 #mode(data,3,NULL,4)
 #mode(data,type=2,digits=1,3)


### 2) Other General Parametric Examples

 ##Example 2.1
 #data<-c(rnorm(15,0,1),rnorm(21,5,1))
 #hist(data)
 #bimodality_amplitude(data,TRUE)
 #bimodality_coefficient(data,TRUE)
 #bimodality_ratio(data,FALSE)

 ##Example 2.2
 #data<-c(rnorm(21,0,1),rnorm(21,5,1))
 #hist(data)
 #bimodality_amplitude(data,TRUE)
 #bimodality_coefficient(data,TRUE)
 #bimodality_ratio(data,FALSE)

### 3) Mixture Proportions Examples
```

```
 ##Example 3.1
 #dist1<-rnorm(21,5,2)
 #dist2<-dist1+11
 #data<-c(dist1,dist2)
 #hist(data)
 #bimodality_amplitude(data,TRUE)
 #bimodality_ratio(data,FALSE)

 ##Example 3.2
 #dist1<-rnorm(21,-15,1)
 #dist2<-rep(dist1,3)+30
 #data<-c(dist1,dist2)
 #hist(data)
 #bimodality_amplitude(data,TRUE)
 #bimodality_ratio(data,FALSE)

 ##Example 3.4
 #dist1<-rep(7,70)
 #dist2<-rep(-7,70)
 #data<-c(dist1,dist2)
 #hist(data)
 #bimodality_ratio(data,FALSE)


##### Parametric Examples #####

### 4) Replicating a two component Gaussian (normal) mixture
### Example 4.1

 ##Draw data & plot the distribution
 #dist1<-rnorm(14,-5,1)
 #dist2<-rnorm(21,5,1)
 #plot(density(c(dist1,dist2)), main="Bimodal Gaussian mixture distribution")

 ##Calculate the means and standard deviations
 #mu1<-mean(dist1)
 #mu2<-mean(dist2)
 #sd1<-sd(dist1)
 #sd2<-sd(dist2)

 ##Apply measures
 #Ashmans_D(mu1,mu2,sd1,sd2)
 #bimodality_separation(mu1,mu2,sd1,sd2)


### 5) Applying to know mixture components
### Example 5.1

 ##Draw data & plot the distribution
 #data<-c(rnorm(15,0,1),rnorm(21,15,3))
 #plot(density(c(dist1,dist2)), main="Bimodal Gaussian mixture distribution")

 ##Apply measures
```

```
#Ashmans_D(mu1,mu2,sd1,sd2)
#bimodality_separation(mu1,mu2,sd1,sd2)
```

---

amps *A helper function to find the amplitudes by finding the peaks and antimodes of a data vector.*

Description

This function finds the peaks and antimodes by way of distribution density estimation. It may be convenient to think of this function as finding the global and local maxima and the local minima of a data vector. Note that the global minimum is asymptotically zero ("0")from based on distribution theory; for practical purposes, double checking the global minimum is good practice. Just run ("min(x))")

Usage

```
amps(x)
```

Arguments

x	Data vector.

Examples

```
data<-c(rnorm(15,0,1),rnorm(21,5,1))
amps(data)
```

Ashmans_D *A function to calculate Ashman, Bird, and Zepf's D Statistic*

Description

This function calculates Ashman's D, which is a measure of how well differentiated two distributions (distribution components) are. For instance, if the two distributions are identical, this statistic is zero. A good rule of thumb is that if the statistic is above ~2, there is good separation. If you suspect that your data is bimodal this can be used by replicating the suspected mixture components and checking the statistic. Alternatively, if the components are known outright this is straightforward to implement.

Usage

```
Ashmans_D(mu1, mu2, sd1, sd2, ...)
```

Arguments

mu1	The mean of mode 1
mu2	The mean of mode 2
sd1	The standard deviation of mode 1
sd2	The standard deviation of mode 2
...	Pass through arguments.

References

Ashman, K., Bird, C., & Zepf, S. (1994). Detecting bimodality in astronomical datasets. The Astronomical Journal, 2348-2361.

Examples

```
##Example 1
dist1<-rnorm(15,4,1)
dist2<-rnorm(21,5,1)
hist(c(dist1,dist2))

mu1<-mean(dist1)
mu2<-mean(dist2)
sd1<-sd(dist1)
sd2<-sd(dist2)
Ashmans_D(mu1,mu2,sd1,sd2)

##Example 2
data<-c(rnorm(15,0,1),rnorm(21,15,3))
hist(data)
Ashmans_D(0,15,1,3)
```

---

bimodality_amplitude *Bimodality Amplitude Function*

Description

This function calculates the Bimodality Ampltiude of a data vector. This is a measure of the proportion of bimodality and the existence of bimodality. The value lies between zero and one (that is: [0,1]) where the value of zero implies that the data is unimodal and the value of one implies the data is two point masses. The proportion of bimodality here is referring to the mixture proportions of two, say, Gaussian (normal) components that can have different frequencies. For instance, a 50 separation will be different from a 25 results of "Example 2" and "Example 3" to get a better understanding.

Usage

```
bimodality_amplitude(x, fig, ...)
```

Arguments

x	Data vector.
fig	Should a figure with the antimodes and peaks be plotted? Defaults to TRUE.
...	Pass through arguments.

References

Zhang, C., Mapes, B., & Soden, B. (2003). Bimodality in tropical water vapour. Quarterly Journal of the Royal Meteorological Society, 129(594), 2847-2866.

Examples

```
#Example 1
data<-c(rnorm(21,0,1),rnorm(21,5,1))
hist(data)
bimodality_amplitude(data,TRUE)

#Example 2
dist1<-rnorm(21,5,2)
dist2<-dist1+11
data<-c(dist1,dist2)
hist(data)
bimodality_amplitude(data,TRUE)

#Example 3
dist1<-rnorm(21,-15,1)
dist2<-rep(dist1,3)+30
data<-c(dist1,dist2)
hist(data)
bimodality_amplitude(data,TRUE)
```

---

bimodality_coefficient

Bimodality Coefficient

Description

This function calculates the Bimodality Coefficient of a data vector with the option for a finite sample (bias) correction. This bias correction is important to correct for the (well-documented) finite sample bias. The bimodality coefficient has a range of zero to one (that is: [0,1]) where a value greater than "5/9" suggests bimodality. The maximum value of one ("1") can only be reached when the distribution is composed of two point masses.

Usage

```
bimodality_coefficient(x, finite = TRUE, ...)
```

Arguments

x	Data vector.
finite	Should the finite sample size correction be applied to the skewness and kurtosis measures? Defaults to TRUE.
...	Pass through arguments.

References

Ellison, A. (1987). Effect of Seed Dimorphism on the Density-Dependent Dynamics of Experimental Populations of Atriplex triangularis (Chenopodiaceae). American Journal of Botany, 74(8), 1280-1288.

Examples

```
data<-c(rnorm(15,0,1),rnorm(21,5,1))
hist(data)
bimodality_coefficient(data,TRUE)
```

---

bimodality_ratio *Bimodality Ratio Function*

Description

This function calculates the Bimodality Ratio which is a measure of the proportion of bimodality. The proportion of bimodality here is referring to the mixture proportions of two, say, Gaussian (normal) components that can have different frequencies. For instance, a 50 separation will be different from a 25 results of "Example 2", "Example 3", and "Example 4" to get a better understanding.

Usage

```
bimodality_ratio(x, list = FALSE, ...)
```

Arguments

x	Data vector.
list	Calculate the Bimodality Ratio for a list of data vectors. This technique is faster than parallelizing for typical big datasets (i.e. when the length of a data vector <=1E9), though benchmarks weren't assessed beyond dimensions of 1E10 x 1E10. When selected, this outputs a list of Bimodality Ratios. Defaults to FALSE.
...	Pass through arguments.

References

Zhang, C., Mapes, B., & Soden, B. (2003). Bimodality in tropical water vapour. Quarterly Journal of the Royal Meteorological Society, 129(594), 2847-2866.

Examples

```
#Example 1
data<-c(rnorm(15,0,1),rnorm(21,5,1))
bimodality_ratio(data,FALSE)

values<-as.list(rep(list(rnorm(15,-4,2),rnorm(21,7,2),data),2))
bimodality_ratio(values,TRUE)

#Example 2
dist1<-rnorm(21,5,2)
dist2<-dist1+11
data<-c(dist1,dist2)
hist(data)
bimodality_ratio(data,FALSE)

#Example 3
dist1<-rnorm(21,-15,1)
dist2<-rep(dist1,3)+30
data<-c(dist1,dist2)
hist(data)
bimodality_ratio(data,FALSE)

#Example 4
dist1<-rep(7,70)
dist2<-rep(-7,70)
data<-c(dist1,dist2)
hist(data)
bimodality_ratio(data,FALSE)
```

---

bimodality_separation *Bimodality Separation Function*

Description

This function calculates the Bimodality Separation of a data vector. Similar to Ashman, Bird, and Zepf's D statistic ("Ashman's D"), the Bimodality Separation statistic measures how differentiated two distributions (distribution components) are. However, this statistic uses the added assumption that both are Gaussian (normal) distributions (or that the distribution is a mixture of two Gaussian (normal) components). For instance, if the two distributions are identical, this statistic is zero.

Usage

```
bimodality_separation(mu1, mu2, sd1, sd2, ...)
```

Arguments

mu1	The mean of mode 1
mu2	The mean of mode 2

sd1	The standard deviation of mode 1
sd2	The standard deviation of mode 2
...	Pass through arguments.

References

Zhang, C., Mapes, B., & Soden, B. (2003). Bimodality in tropical water vapour. Quarterly Journal of the Royal Meteorological Society, 129(594), 2847-2866.

Examples

```
##Example 1
dist1<-rnorm(15,4,1)
dist2<-rnorm(21,5,1)
hist(c(dist1,dist2))

mu1<-mean(dist1)
mu2<-mean(dist2)
sd1<-sd(dist1)
sd2<-sd(dist2)
bimodality_separation(mu1,mu2,sd1,sd2)

#Example 2
data<-c(rnorm(15,0,1),rnorm(21,15,3))
hist(data)
bimodality_separation(0,15,1,3)
```

---

kurtosis *Kurtosis Function*

Description

This function calculates the excess kurtosis of a data vector with optional bias correction. Kurtosis is a meaure of the peakedness or how heavy the tails of a distribution are–this dual interpretation is a result of the obvious inverse relationship between fat tails and high peaks. Excess kurtosis is simply "kurtosis-3." This is a correction that is often done to allow for comparision to the normal distribution–which has a kurtosis of 3 and excess kurtosis of 0. A kurtosis greater than 0 means that the distribution is leptokurtic and so has a high peak with skinny tails. Conversely, a kurtosis less than 0 means that the distribution is platykurtic and so has a low peak and heavy tails. This interpretation is slightly more complicated once the distribution is not unimodal and/or non-zero skewness. Comparing to Gaussian (normal) moments is more acceptable in these cases.

Usage

```
kurtosis(x, finite)
```

Arguments

x	Data vector.
finite	Should the finite sample correction (bias correction) be used? Defaults to TRUE.

Examples

```
data<-c(rnorm(15,0,1),rnorm(21,5,1))
hist(data)
kurtosis(data,TRUE)
```

---

modes *Modes*

Description

This function calculates the mode for an integer valued data vector by default. It also calculates "nmore" more modes than the most frequently occurring value and can take in data that should be treated as integers, real numbers (which can optionally be rounded to the "digits" number of significant digits), or factors. This modes function finds the value(s) that occur most frequently so, crucially, if there is a tie in the frequency count for the mode it will yield two modes instead of the lower valued mode. Yielding all modes instead of just the lowest mode is particularly important when more advanced statistics and machine learning techniques are employed.

Usage

```
modes(data, type = 1, digits = "NULL", nmore = "NULL")
```

Arguments

data	Data vector.
type	The type of data the vector contains. Defaults to 1. • "1" converts the data to integers and then finds the mode(s) • "2" rounds the data to the number of significant digits found in 'digits' and then finds the mode(s) • "3" treats the data as factors and then finds the mode(s)
digits	How many significant digits should be retained? Defaults to NULL.
nmore	Specifices how many modes should be attempted. That is, it finds the "nmore" modes with decreasing frequency. For example, if the data is bimodal, specify nmore=2 and the two most frequent values with their frequency will be returned.
...	Pass through arguments.

Examples

```
#Example 1
data<-c(rep(6,9),rep(3,3))
modes(data,type=1,"NULL","NULL")

#Example 2
data<-c(rep(6,9),rep(3,9))
modes(data,type=1,"NULL","NULL")

#Example 3
data<-c(rep(6,9),rep(3,8),rep(7,7),rep(2,6))
modes(data,type=1,"NULL",2)

#Example 4
data<-c(rnorm(15,0,1),rnorm(21,5,1),rep(3,3))
modes(data)

#Example 5
data<-c(rep(6,3),rep(3,3),rnorm(15,0,1))
modes(data,3,NULL,4)
modes(data,type=2,digits=1,3)
```

---

nth_highest *N-th Highest Value Function*

Description

This function allows you to calculate the N-th highest number of a vector. In other words, it is the N-th max; it calculates the maximum after removing (n-1) higher numbers (maxes). Note that k=1 yields the traditional global maximum.

Usage

```
nth_highest(x, k = 1)
```

Arguments

x	Data vector.
k	The N-th highest value or N-th max. For example, k=2 yields the second highest value. Note that k=1 gives the highest value aka the global max. Defaults to 1.

Examples

```
data<-c(rnorm(15,0,1),rnorm(21,5,1))
nth_highest(data,2)
```

skewness *Skewness Function*

Description

This function calculates the skewness of a data with optional bias correction. The skewness is a measure of the symmetry of a distrbution. A negative skewness means the data is left skewed or has a fat left tail. The converse is true for a positive skew.

Usage

```
skewness(x, finite = TRUE)
```

Arguments

x	Data vector.
finite	Should the finite sample correction (bias correction) be used? Defaults to TRUE.

Examples

```
data<-c(rnorm(15,0,1),rnorm(21,5,1))
hist(data)
skewness(data,TRUE)
```

Index