

Package ‘soundmeterR’

December 15, 2021

Type Package

Title R package for sound meter alike analysis

Version 0.0.1

Author Cássio Rachid Simões, Shaka Nóbrega Marinho Furtado, Ingrid Maria Denóbile Torres, Marcelo Felgueiras Napoli and Carlos Barros de Araújo

Maintainer Cássio Rachid Simões <cassiorachid@gmail.com>

Description An R package designed for sound analysis through spectral analysis in octaves and one-thirds octaves, making possible calibration to obtain sound pressure level (SPL) and Sound Intensity Level (SIL).

URL <https://github.com/cassiorachid/soundmeterR>

BugReports <https://github.com/cassiorachid/soundmeterR/issues>

License GPL (>=3)

Imports tuneR, seewave

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1.9000

R topics documented:

dBtoLinear	2
LineartodB	3
rms.dB	4
timbre	5
timbreCal	6
Tweighting	7
Index	9

dBtoLinear

Convert deciBels scales to linear

Description

Function to convert dB scales. The conversion can be made either from dB to linear (dBtoLinear) or from linear to dB ([LineartodB](#)).

Usage

```
dBtoLinear(x, factor="IL", ref=1)
```

Arguments

x	Numerical. A numeric vector or a numeric matrix with dB values for dBtoLinear function or linear values for LineartodB .
factor	Character. Specify in what factor the function should use to convert your data. SPL (Sound Pressure Level) for amplitude like data (factor 20) or IL (Intensity Level) for power like (fator 10). (By default IL)
ref	Numerical. Reference value for conversion. For Sound in water the ref is 1 microPa and on air 20 microPa. (By default 1)

Details

For details about the factor choice, we recommend the reading of [this](#) web page.

Value

The same object of the input with the converted values.

Author(s)

Cássio Rachid Simões <cassiorachid@gmail.com>

See Also

[rms.dB](#), [LineartodB](#), [convSPL](#)

Examples

```
dBtoLinear(c(80,60,65,62))  
LineartodB(dBtoLinear(c(80,60,65,62)))
```

LineartodB	<i>Convert linear scales to decibels</i>
------------	--

Description

Function to convert dB scales. The conversion can be made either from dB to linear ([dBtoLinear](#)) or from linear to dB (LineartodB).

Usage

```
LineartodB(x, factor="IL", ref=1)
```

Arguments

x	Numerical. A numeric vector or a numeric matrix with dB values for dBtoLinear function or linear values for LineartodB.
factor	Character. Specify in what factor the function should use to convert your data. SPL (Sound Pressure Level) for amplitude like data (factor 20) or IL (Intensity Level) for power like (factor 10). (By default IL)
ref	Numerical. Reference value for conversion. For Sound in water the ref is 1 microPa and on air 20 microPa. (By default 1)

Details

For details about the factor choice, we recommend the reading of [this](#) web page.

Value

The same object of the input with the converted values.

Author(s)

Cássio Rachid Simões <cassiorachid@gmail.com>

See Also

[rms.dB](#), [dBtoLinear](#), [convSPL](#)

Examples

```
dBtoLinear(c(80,60,65,62))
LineartodB(dBtoLinear(c(80,60,65,62)))
```

`rms.dB`*Root Mean Square with dB values*

Description

Function to compute the root mean square (RMS) of values in decibels (dB).

Usage

```
rms.dB(x, level="SPL", ref=1, ...)
```

Arguments

<code>x</code>	Numerical. A numeric vector or a numeric matrix with dB values.
<code>level</code>	Character. Specify in what scale your data is. SPL for Sound Pressure Level or IL for Intensity Level. (By default SPL)
<code>ref</code>	Numerical. Reference value for conversion. For Sound in water the ref is 1 microPa and on air 20 microPa. (By default 1)
<code>...</code>	Further arguments passed to mean .

Details

This function converts your dB data to linear values (through [dBtoLinear](#) function), compute the Root Mean Square (rms), and converts the result back to dB (through [LineartodB](#) function).

This function was adapted from [meandB](#) and [rms](#) functions from [seewave](#) package. See their help for more details.

Value

A numeric value that represents the root mean square of x.

Author(s)

Cássio Rachid Simões <cassiorachid@gmail.com>

See Also

[meandB](#), [rms](#)

Examples

```
rms.dB(c(80, 60, 65, 62))
```

timbre	<i>Level per octaves or one-thirds octaves</i>
--------	--

Usage

```
timbre(files="wd", weighting="none", bands="thirds", ref=20, saveresults=F,
       outname=NULL, Leq.calib=NULL, Calib.value=NULL, time.mess=T, stat.mess=T)
```

Arguments

files	The audiofile to be analyzed. Can be "wd" to get all ".wav" files on the work directory, a file name (or a character containing a list of filenames) that exist in the work directory (only ".wav" files accepted), or an Wave object (or a list containing more than one Wave object). (By default: "wd")
weighting	Character. Argument passed to dBweight to indicate the weighting curve to use on the analysis. 'A', 'B', 'C', 'D', 'ITU', and 'none' are supported. See dBweight for details. (By default: "none")
bands	Character. Choose the type of frequency band of the output. "octaves" to octaves bands intervals or "thirds" to one-third octaves bands intervals. (by default: "thirds")
ref	Numerical. The reference value for dB conversion. For sound in water, the common is 1 microPa, and for sound on air 20 microPa. (By default 20)
saveresults	Logical. Set TRUE if you want to save a txt file with the results of the function execution. (By default: FALSE)
outname	Character. If saveresults is TRUE, you can specify a name to appear on the txt file name after the default name. (By default: NULL)
Leq.calib	Numerical. The sound pressure level (in dB SPL) that the signal in the audio file must have (by default: NULL). Can not be set if Calib.value is also set.
Calib.value	Numerical. The calibration value returned from the analysis of a reference signal using Leq.calib (by default: NULL). Can not be set if Leq.calib is also set.
time.mess	Logical. Activate or deactivate message of time to complete the function execution. (By default: TRUE)
stat.mess	Logical. Activate or deactivate status message of the function execution. (By default: TRUE)

Details

Caution: You need to use an audiofile with entire values of seconds of duration to avoid bugs. Example: 35s, 60s, 19s. By default, the function will trunc your audiofile to the next entire value of seconds.

These function works only with mono audiofiles.

The audio files need to have at least 44100Hz of sampling rate.

If you intend to work with decibels at full scale (dBFS), we recommend setting ref=1. With this, your results will be relative to 0 dBFS.

Author(s)

Cássio Rachid Simões <cassiorachid@gmail.com>

References

Power spectrum adapted from: Carcagno, S. 2013. Basic Sound Processing with R [Blog post]. Retrieved from <http://samcarcagno.altervista.org/blog/basic-sound-processing-r/>

Miyara, F. 2017. Software-Based Acoustical Measurements. Springer. 429 pp. DOI: 10.1007/978-3-319-55871-4

timbreCal

Timbre analysis for audiofiles with reference signal

Description

This function passes the parameters to `timbre` to automatize the calibration and return spectral analysis with dB SPL results.

Usage

```
timbreCal(files="wd", SignalDur=NULL, RefValue=NULL, ref=20, weighting="none",
          bands="thirds", saveresults=F, outname=NULL, time.mess=T, stat.mess=T)
```

Arguments

files	The audiofile to be analyzed. Can be "wd" to get all ".wav" files on the work directory, a file name (or a character containing a list of filenames) that exist in the work directory (only ".wav" files accepted), or an Wave object (or a list containing more than one Wave object). (By default: "wd")
SignalDur	Numerical. Specify the reference signal duration (in seconds) on the beginning of the audiofile. (By default: NULL)
RefValue	Numerical. Specify the reference signal sound pressure level (in deciBells SPL) on the beginning of the audiofile. (By default: NULL)
ref	Numerical. The reference value for dB conversion. For sound in water, the common is 1 microPa, and for sound on air 20 microPa. (By default 20)
weighting	Character. Indicate the weighting curve to use on the anlysis. A, B, C and none are supported. (By default: "none")
bands	Character. Choose the type of frequency band of the output. "octaves" to octaves bands intervals or "thirds" to one-third octaves bands intervals. (by deaefault: "thirds")
saveresults	Logical. Set TRUE if you want to save a txt file with the results of the function execution. (By default: FALSE)
outname	Character. If saveresults is TRUE, you can specify a name to appear on the txt file name after the default name. (By default: NULL)

<code>time.mess</code>	Logical. Activate or deactivate message of time to complete the function execution. (By default: TRUE)
<code>stat.mess</code>	Logical. Activate or deactivate status message of the function execution. (By default: TRUE)

Details

To use this function, the audio file must begin with 2 seconds of silence, followed by a reference signal with known SPL, followed by another 2 seconds of silence, and the following sound to analyze.

The duration of the reference signal must be specified (in seconds) on the `SignalDur` argument and his value (in dB SPL) on the `refValue` argument.

Author(s)

Cássio Rachid Simões <cassiorachid@gmail.com>

See Also

[timbre](#)

Tweighting	<i>Time Weighting of a Audiofile</i>
------------	--------------------------------------

Description

Escrever descrição

Usage

```
Tweighting(file, window = "fast", Leq.calib = NULL, ...)
```

Arguments

<code>file</code>	Wave object
<code>window</code>	Character. Wich time window should be used. 'fast' or 'slow' are accepted. (by default: "fast")
<code>Leq.calib</code>	Numeric. The sound pressure level (in dB SPL) that the signal in the audio file must have (by default: NULL). This parameter is passed to timbre function.
<code>...</code>	Further arguments passed to timbre .

Details

This function split your audiofile in smaller files defined as `fast` (0.125s) and `slow` (1s) and analyze each one with [timbre](#) function.

Value

A numeric vector

See Also

[timbre](#), [soundmeter](#)

Examples

```
#creating an example sound file
som=sine(1000, duration = 44500)

#default options without calibration (results in dBFS)
Tweighting(som)

#Simulation of a calib signal with a Leq of 94dB in the field ####
#fast
Tweighting(som, window = "fast", bands="octaves", Leq.calib=94)
#slow
Tweighting(som, window = "slow", bands="octaves", Leq.calib=94)

#Using the result of the simulation above to calibrate the sound and output
#fast
Tweighting(som, window = "fast", bands="octaves", Calib.value=309.67)
#slow
Tweighting(som, window = "slow", bands="octaves", Calib.value=309.67)
```


Index

convSPL, [2](#), [3](#)

dBtoLinear, [2](#), [3](#), [4](#)

dBweight, [5](#)

LineartodB, [2](#), [3](#), [4](#)

mean, [4](#)

meandB, [4](#)

rms, [4](#)

rms.dB, [2](#), [3](#), [4](#)

seewave, [4](#)

soundmeter, [8](#)

timbre, [5](#), [6–8](#)

timbreCal, [6](#)

Tweighting, [7](#)