Package 'audiolyzR'

February 19, 2015

Type Package

Title audiolyzR: Give your data a listen

Version 0.4-9

Date 2013-2-16

Author Eric Stone, Jesse Garrison

Maintainer Eric Stone <ericstone@temple.edu>

Depends hexbin, RJSONIO, plotrix

Description Creates audio representations of common plots in R

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2013-02-17 08:56:25

R topics documented:

`audiolyzR-package` *audiolyzR: Listen to your data*

Description

Intended as a tool for familiarization with a dataset, identification of outliers, and further analyses. This may also be helpful in describing data to the visually impaired.
NOTE: This package requires the installation of a separate standalone synthesizer application. The R functions will send plots to this program, which is where you will "play" your plots. See Details for details.
The standalone application uses QuickTime, without which visual details will not appear.

Details

Package:	audiolyzR
Type:	Package
Version:	0.4-9
Date:	2013-2-16
License:	GPL-2

audiolyzR translates scatterplots, scatterplot matrices, histograms, and (soon) other plots into corresponding audio graphics. You will see that the plots are played either by looping from left to right, or by directing an interactive cursor.
You will have live control over global volume and tempo, along with quality (major vs minor vs augmented, etc.), range of pitches (in case you have trouble hearing particularly high or low notes), and gap between loops.

Things to note and pay attention to while you listen:
1) The X or horizontal axis corresponds to time (not pitch).
2) Pitch corresponds to the Y or vertical axis.
3) Reverb is inversely proportional to correlation (more reverb for less correlation).
4) Synthesizer dryness is mildly related to number of points in a column.
5) Relative note volume is inversely proportional to the number of notes in a neighborhood of each (higher volume for fewer neighbors)

Instructions for the external **audiolyzR** application:
In order to run **audiolyzR**, you need to install the appropriate standalone application. The first time you run any **audiolyzR** command, it will automatically install the appropriate version for your system. If you prefer to download the files yourself:
Mac:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_mac_v5.zip

Windows:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_win_v5.zip

Also, make sure you are running a functioning version of Apple QuickTime, or you won't see the visual representation of the data in the synthesizer.

Author(s)

Eric Stone, Jesse Garrison,
Contributions from Nate Wheeler

Maintainer: <ericstone@temple.edu>

References

Baier T, Neuwirth E, Meo MD (2011). "Creating and Deploying an Application with RExcel and R." R Journal, 3/2.

Brown E (2012). playitbyr: Representing and exploring data through sound. R Package: Version 0.2-1, http://playitbyr.org/.

Carr D, Maechler M, Lewin-Koh N (2012). hexbin: Hexagonal Binning Routines. R Package: Version 1.26-0, http://cran.r-project.org/web/packages/hexbin/index.html.

CDC (2012). "National Health and Nutrition Examination Survey." www.cdc.gov/nchs/nhanes.htm.

Chambers JM (1983). Graphical methods for data analysis. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Chambers JM, Mathews MV, Moore FR (1974). "Auditory Data Inspection." Technical Memorandum, 74-1214-20.

Cycling74 (2012). Max/MSP. http://cycling74.com/docs/max6/dynamic/c74_docs.

Flowers J (2005). "THIRTEEN YEARS OF REFLECTION ON AUDITORY GRAPHING : PROMISES , PITFALLS , AND POTENTIAL NEW DIRECTIONS." Faculty Publications, Univ of Nebraska, pp. 406 to 409. http://digitalcommons.unl.edu/cgi/viewcontent.cgi?article=1429&context=psychfacpub.

Froehlich I (2011). "Tonematrix." http://tonematrix.audiotool.com.

Frysinger SP (2005). "A Brief History of Auditory Data Representation to the 1980s." First Symposium on Auditory Graphs, pp. 1980 to 1983.

Hermann T, Hunt A, Neuhoff JG, Walker BN, Nees MA (2011). The Sonification Handbook, Chapter 2: Theory of Sonification. ISBN 9783832528195.

Hunt A, Hermann T, Pauletto S (2004). "Interacting with sonification systems: closing the loop." Proceedings. Eighth International Conference on Information Visualisation, IV, 879 to 884. http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1320244.

Lang DT (2012). RJSONIO: Serialize R objects to JSON, JavaScript Object Notation. R Package: Version 1.0-1, http://cran.r-project.org/web/packages/RJSONIO/RJSONIO.pdf.

Merabet LB, Pascual-Leone A (2010). "Neural reorganization following sensory loss: the opportunity of change." Nature reviews. Neuroscience, 11(1), 44 to 52. ISSN 1471-0048. doi:10.1038/nrn2758. http://www.ncbi.nlm.nih.gov/pubmed/19935836.

Neuwirth E (2012). RExcelInstaller: Integration of R and Excel, (use R in Excel, read/write XLS files). R Package: Version 3.2.3-1, http://cran.r-project.org/web/packages/RExcelInstaller/index.html.

R Core Team (2012). R: A Language and Environment for Statistical Computing. R Foun- dation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, http://www.R-project.org.

Sarkar D (2008). Lattice: Multivariate Data Visualization with R. Springer Science+Business Media.

Sarkar D (2012). lattice: Lattice Graphics. R Package: Version 0.20-10, http://cran.r-project.org/web/packages/lattice/index.html.

Stein BE, Stanford TR (2008). "Multisensory integration: current issues from the perspective of the single neuron." Nature reviews. Neuroscience, 9(4), 255 to 66. ISSN 1471-0048. doi: 10.1038/nrn2331. http://www.ncbi.nlm.nih.gov/pubmed/18354398.

Stone E, Garisson J (2012). "Give Your Data a Listen." In UserR! Conference Proceedings. http://biostat.mc.vanderbilt.edu/wiki/pub/Main/UseR-2012/81-Stone.pdf.

Journal of Statistical Software 15 Tukey JW (1977). "Exploratory Data Analysis." Addison-Wesley.

Vercoe B (2012). The Canonical Csound Reference Manual. MIT Media Lab. Version 5.17.1.

Examples

```
##Basic audiolyzR examples
data(NHANES)

## Not run:
##Scatter Plot
audioScatter(BMI ~ Weight,data=NHANES)

##Scatterplot matrix
audioSplom(data=NHANES, bins=20)

##Histogram
audioHist(NHANES$Weight, name="Weight")

## End(Not run)
```

`audioHist` *Generate an audiolyzR version of a Histogram*

Description

Generates a histogram and translates it to **audiolyzR** format.

Usage

```
audioHist(x, name = "Variable", purge.plots = FALSE,
          bins = 30, breaks = "Scott", radius = floor(sqrt(bins))-1,
          key = "C", quality = "Major", tempo = 80, reverb = 1,
          directory = file.path (Sys.getenv("R_LIBS_USER"), "audiolyzR"),
        output = file.path (tempdir(), "json_matrix"), write.to.home = NULL, ...)
```

Arguments

x	A vector of values to be plotted as a histogram, of form `data.frame$variable`.
name	A character string specifying the name of x. This will be used to label the plots. The default is "Variable".
purge.plots	Specifies whether to erase existing **audiolyzR** plots that the synthesizer accesses. Default is FALSE, so plots will pile up unless you change it to TRUE.
bins	Optional specification of bins for hex binning step. Default is 30.
breaks	Optional specification of breaks for the histogram. Default is "Scott".
radius	Used to obtain the number of nearest neighbors for each bin/point. Default is the square root of number of bins.
key	Optional, not yet fully implemented. The desired key for the audio matrix. This can be adjusted manually in the **audiolyzR** synthesizer.
quality	Optional. Quality of the specified key. Default is "Major". This is not yet fully implemented, and can be changed manually in the synthesizer.
tempo	Optional. Tempo to set as default (You can change it manually in the synthesizer later). Default is 115 beats per minute.
reverb	Optional. A number between 0 and 1 (inclusive) that specifies the amount of reverb in the resulting audio plot. A value of 1 is no reverb while a value of 0 produces the most.
directory	The path of the parent directory containing the **audiolyzR** synthesizer. This defaults to the "audiolyzR" folder in your "R_LIBS_USER" directory.
output	This is simply a temporary directory by default. If you wish to change it, the only sensible option is the HOME directory. Included for transparency.
write.to.home	The standalone synthesizer application will look in your HOME directory for plot files when it opens. R will request permission to write there during the current session. If you refuse, it will still work, but you will need to drag the appropriate folder to the synthesizer (a message will explain how). Adjusting this in the function call will have no effect.
...	Additional arguments to be fed to `hexbin()` and `hist()`.

Value

Sends a JSON file to the **audiolyzR** synthesizer, and by default, plots the corresponding histogram in R.

External Application Instructions

The **audiolyzR** package requires a free standalone synthesizer application. the first time you run an `audio*Plot*` function, the program will download and install automatically. By default, the app installs into the `file.path (Sys.getenv("R_LIBS_USER"), "audiolyzR")` directory, which is where the package's functions will look for it. You are free to install it whereever you like, but you must specify its parent `directory` in your function calls.

If you prefer to install it yourself:

audiolyzR application for *Mac*:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_mac_v5.zip

audiolyzR application for *Windows*:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_win_v5.zip

Side Effects

Saves a file to the `output` directory in order to generate audio. Also, by default, a plot will be produced in the graphics window.

Author(s)

Eric Stone, Jesse Garrison

References

audiolyzR: http://biostat.mc.vanderbilt.edu/wiki/pub/Main/UseR-2012/81-Stone.pdf
Max/MSP: http://cycling74.com/whatismax/

See Also

`hexbin`

`hist`

Examples

```
##Basic audioScatter example
## Not run:
data(NHANES)
audioHist(NHANES$Diet.Iron, name="Dietary Iron")

## End(Not run)
```

---

audioScatter *Generate an audiolyzed version of a scatterplot*

Description

Transaltes a scatterplot into an audio graph where notes and chords correspond to a binned version of the plot.

Usage

```
audioScatter(x, y = NULL, z = NULL, data, purge.plots = FALSE, show.plots = TRUE,
             bins = 30, aspect = 1, radius = floor(sqrt(bins))-1,
             key = "C", quality = "Major", tempo = 115,
             directory = file.path (Sys.getenv("R_LIBS_USER"), "audiolyzR"),
          output = file.path (tempdir(), "json_matrix"), write.to.home = NULL, ...)

panel.audiolyzR(x, y, ...)
```

Arguments

x	A formula specifying either 2 or 3 variables from the specified data object. For now, if 3 variables are specified, the 3rd is considered a "conditional" variable and it atop the interaction between the first 2, in the same manner as a call like `xyplot (y ~ x + z, data=data, outer=FALSE)` -or- The x-axis variable name as a text string.
y	Specifies the y-axis variable if formula not used, should also be a text string.
z	An optional "conditional" variable that is plotted atop the interaction between the first 2
data	Required data frame object.
purge.plots	Specifies whether to erase the directory containing the plots that the external **audiolyzR** synthesizer accesses. Default is FALSE, so plots will pile up unless you change it to TRUE.
show.plots	Specifies whether to print the binned version of the plot in R's grpahics window. Default is set to TRUE.
bins	Optional specification of bins for hex binning step. Default is 30.
aspect	Optional aspect ratio adjustment. Default is 1:1.
radius	Used to obtain the number of nearest neighbors for each bin/point. Default is square root of number of bins.
key	Optional, not yet fully implemented. The desired key for the audio matrix. This can be adjusted manually in the **audiolyzR** synthesizer (starting note).
quality	Optional. Quality of the specified key. Default is "Major". This is not yet fully implemented, and can be changed manually in the **audiolyzR** synthesizer.
tempo	Optional. Tempo for chord progressions in the audio plot. Default is 115 bpm, and it can be adjusted once the plot is generated.
directory	The path of the parent directory containing the **audiolyzR** synthesizer. This defaults to the "audiolyzR" folder in your "R_LIBS_USER" directory.
output	This is simply a temporary directory by default. If you wish to change it, the only sensible option is the HOME directory. Included for transparency.
write.to.home	The standalone synthesizer application will look in your HOME directory for plot files when it opens. R will request permission to write there during the current session. If you refuse, it will still work, but you will need to drag the appropriate folder to the synthesizer (a message will explain how). Adjusting this in the function call will have no effect.

... Additional arguments to be fed to `hexbin()` and `hexbinplot()`.

Details

The panel function `panel.audiolyzR` adds an audio plot component to `xyplot()`. This functions at a basic level of implementation and only currently supports `outer = FALSE` plots. Support for more complex **lattice** graphics will be incorporated into future releases.

Value

Sends a JSON file to The audiolyzR, which is then played. This requires the installation of The audiolyzR. Also plots the resulting hexbin plot(s)

External Application Instructions

The **audiolyzR** package requires a free standalone synthesizer application. the first time you run an `audio*Plot*` function, the program will download and install automatically. By default, the app installs into the `file.path (Sys.getenv("R_LIBS_USER"), "audiolyzR")` directory, which is where the package's functions will look for it. You are free to install it whereever you like, but you must specify its parent `directory` in your function calls.

If you prefer to install it yourself:

audiolyzR application for *Mac*:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_mac_v5.zip

audiolyzR application for *Windows*:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_win_v5.zip

Side Effects

Saves a file to the `output` directory in order to generate audio. Also, a plot will be produced in the graphics window.

Author(s)

Eric Stone, Jesse Garrison

References

audiolyzR: http://biostat.mc.vanderbilt.edu/wiki/pub/Main/UseR-2012/81-Stone.pdf
Max/MSP: http://cycling74.com/whatismax/

See Also

`hexbin`

Examples

```
##Basic audioScatter example

## Not run:
data(NHANES)
audioScatter("Weight","BMI",data=NHANES)
audioScatter(BMI ~ Weight + Transferin,data=NHANES)

## End(Not run)
```

`audioSplom` *audioSplom geneartes a scatterplot matrix-like series of tone matrices*

Description

Translates a scatterplot matrix into an audioplot, with help from `hexplom`. The **audiolyzR** plots are played according to the selected plot in the drop-down menu. The result can be considered similar to "movements" of a piece of music, with each scatterplot representing a separate movement within the larger piece.

Usage

```
audioSplom(x = NULL, data, purge.plots = FALSE,
           bins = 30, aspect = 1, radius = floor (sqrt(bins)) - 1,
           key = "C", quality = "Major", tempo = 115,
           directory = file.path (Sys.getenv("R_LIBS_USER"), "audiolyzR"),
         output = file.path (tempdir(), "json_matrix"), write.to.home = NULL, ...)
```

Arguments

`x`	An optional character vector of names c("name1","name2",etc.) -or- An optional formula specifying variables to be included in the splom. All that is required is a data.frame object.
`data`	Required data frame object
`purge.plots`	Specifies whether to erase the directory containing the plots that the **audiolyzR** synthesizer accesses. Default is FALSE, so plots will pile up unless you change it to TRUE.
`bins`	Optional specification of bins for hex binning step. Default is 30.
`aspect`	Optional aspect ratio adjustment. Default is 1
`radius`	Used to obtain the number of nearest neighbors for each bin/point. Default is square root of number of bins.
`key`	Optional, not yet fully implemented. The desired key for the audio matrix. This can be adjusted manually in the **audiolyzR** synthesizer.
`quality`	Optional. Quality of the specified key. Default is "Major". This is not yet fully implemented, and can be changed manually in the **audiolyzR** synthesizer.

`tempo`	Optional. Tempo to set as default (You can also change tempo in the **audiolyzR** synthesizer). Default is 115 bpm.
`directory`	The path of the parent directory containing the **audiolyzR** synthesizer. This defaults to the "audiolyzR" folder in your "R_LIBS_USER" directory.
`output`	This is simply a temporary directory by default. If you wish to change it, the only sensible option is the HOME directory. Included for transparency.
`write.to.home`	The standalone synthesizer application will look in your HOME directory for plot files when it opens. R will request permission to write there during the current session. If you refuse, it will still work, but you will need to drag the appropriate folder to the synthesizer (a message will explain how). Adjusting this in the function call will have no effect.
`...`	Additional arguments to be fed to `hexbin()` and `hexbinplot()`.

Value

Produces a hexplom of the specified or supported variables, and then plays the corresponding audio scatterplots, generated in the **audiolyzR** standalone.

External Application Instructions

The **audiolyzR** package requires a free standalone synthesizer application. the first time you run an `audio*Plot*` function, the program will download and install automatically. By default, the app installs into the `file.path (Sys.getenv("R_LIBS_USER"), "audiolyzR")` directory, which is where the package's functions will look for it. You are free to install it whereever you like, but you must specify its parent `directory` in your function calls.

If you prefer to install it yourself:

audiolyzR application for *Mac*:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_mac_v5.zip

audiolyzR application for *Windows*:
http://s3.amazonaws.com/audiolyzR/installers/the_audiolyzR_win_v5.zip

Side Effects

Saves a file to the `output` directory in order to generate audio. Also, a plot will be produced in the graphics window.

Author(s)

Eric Stone, Jesse Garrison

References

audiolyzR: http://biostat.mc.vanderbilt.edu/wiki/pub/Main/UseR-2012/81-Stone.pdf
Max/MSP: http://cycling74.com/whatismax/

See Also

hexbin

Examples

```
##Simple audioSplom example
## Not run:
data(NHANES)
audioSplom(data=NHANES)

## End(Not run)
```

Index