

# Package: OpeNoise (via r-universe)

November 21, 2024

**Type** Package

**Title** Environmental Noise Pollution Data Analysis

**Version** 0.2-18

**Maintainer** Pasquale Scordino <scordino.pasquale@gmail.com>

**Imports** tidy, lubridate, ggplot2, pracma

**Description** Provides analyse, interpret and understand noise pollution data. Data are typically regular time series measured with sound meter. The package is partially described in Fogola, Grasso, Masera and Scordino (2023, <DOI:10.61782/fa.2023.0063>).

**Depends** R (>= 3.5.0)

**License** GPL (>= 3)

**URL** <https://arpapiemonte.github.io/openoise-analysis/>,  
<https://github.com/Arpapiemonte/openoise-analysis/>

**BugReports** <https://github.com/Arpapiemonte/openoise-analysis/issues/>

**Encoding** UTF-8

**LazyData** TRUE

**RoxygenNote** 7.3.2

**Suggests** rmarkdown, htmltools, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Pasquale Scordino [aut, cre], Simone Sperotto [ctb], Stefano Masera [ctb], Daniele Grasso [ctb], Jacopo Fogola [ctb]

**Repository** CRAN

**Date/Publication** 2024-11-20 10:40:10 UTC

**Config/pak/sysreqs** libicu-dev

## Contents

AcousticQuantilePlot . . . . .	2
AcousticWeightingTable . . . . .	3
AcuDNPercentile . . . . .	4
AcuPercentile . . . . .	5
avr.day.night . . . . .	5
dataset_impulsive1 . . . . .	6
dataset_impulsive2 . . . . .	7
dbsum . . . . .	7
deco.time . . . . .	8
dfBW . . . . .	8
dfImpulsiveTrasform . . . . .	9
energetic.mean . . . . .	9
energetic.min . . . . .	10
energetic_w.mean . . . . .	11
exampleHourlyData . . . . .	11
ExtractIndexMark . . . . .	12
HolidaysDate . . . . .	12
HourlyEmean . . . . .	13
IntrusiveIndex . . . . .	14
iso . . . . .	15
LdenCalculator . . . . .	15
markers . . . . .	16
Maskapply . . . . .	16
PIFA . . . . .	17
PIFC . . . . .	17
PlotNoiseTHcompare . . . . .	18
PlotNoiseTimeHistory . . . . .	19
PlotSpectrogram . . . . .	20
PTFA . . . . .	20
PTFC . . . . .	21
RoundTo . . . . .	21
runningLeq . . . . .	22
search.tone . . . . .	22
searchImpulse . . . . .	23
SELcalc . . . . .	24
<b>Index</b>	<b>25</b>

---

AcousticQuantilePlot *Plot acoustic quantile*

---

### Description

Returns a plot of acoustic quantile of 1/3 band frequency

**Usage**

```
AcousticQuantilePlot(df, Cols, Quantile, TimeZone = "UTC")
```

**Arguments**

df	is a dataframe
Cols	vector of index cols (1/3 band frequency)
Quantile	quantile, for example 0.95
TimeZone	Time zone dataset (default is UTC)

**Value**

an OpeNoise object of class ggplot

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Plot reverse quantile of 1/3 band frequency

library(lubridate)

datasetI <- dataset_impulsive1
datasetH <- dfImpulsiveTrasform(datasetI)

AcousticQuantilePlot(df = datasetH, Cols =c(3:38), Quantile = 0.95,
                     TimeZone = "UTC")
```

---

AcousticWeightingTable

*Weighting acoustic table*

---

**Description**

Weighting acoustic table

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

AcuDNPercentile	<i>Calculate reverse Percentile for period</i>
-----------------	--

---

**Description**

Returns a vector of acoustic percetile

**Usage**

```
AcuDNPercentile(df, parameter, from, to, period)
```

**Arguments**

df	is a dataframe with Leq data
parameter	is a parameter, example "LAeq"
from	is start hour
to	is end hour
period	is a period night or day

**Value**

a list of acoustic percentil values by night and daily period

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate reverse quantile of a dataframe by period  
  
data("exampleHourlyData")  
  
AcuDNPercentile(df = exampleHourlyData,  
                parameter = "leq",  
                from = "5",  
                to = "22",  
                period = "night")[1:5]
```

---

AcuPercentile	<i>Calculate reverse Percentile</i>
---------------	-------------------------------------

---

**Description**

Returns a vector of acoustic percentile

**Usage**

```
AcuPercentile(x)
```

**Arguments**

x is a vector with Leq data

**Value**

vector of acoustic percentil values

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate reverse quantile of a vector of values  
  
#data(PTFA)  
  
AcuPercentile(PTFA$LAeq)
```

---

avr.day.night	<i>Calculate daily and nightly energetic mean period</i>
---------------	--

---

**Description**

Returns a dataframe with energetic mean

**Usage**

```
avr.day.night(x, variable, period = "day", stat = "n_mean", ...)
```

**Arguments**

x	is a data frame
variable	is variable to apply function
period	is "day" or "night"
stat	is "n_mean" or "e_mean" like mean and energetic mean
...	another arguments

**Value**

dataframe of energetic mean values by night or daily period

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate energetic mean in nightly period (22-06)

#data(exampleHourlyData)

avr.day.night(exampleHourlyData, "leq", period = "night",
               stat = "e_mean")[1:5, ]

# Calculate energetic mean in daily period (06-22)
avr.day.night(exampleHourlyData, "leq", period = "day",
               stat = "e_mean")[1:5, ]

# Calculate mean in daily period (06-22)
avr.day.night(exampleHourlyData, "leq", period = "day",
               stat = "n_mean")[1:5, ]
```

---

dataset\_impulsive1      *Noise dataset of impulsive event (100 ms acquisition time)*

---

**Description**

Noise dataset of impulsive event (100 ms acquisition time)

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

dataset\_impulsive2      *Noise dataset of impulsive event (100 ms acquisition time)*

---

**Description**

Noise dataset of impulsive event (100 ms acquisition time)

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

dbsum      *Calculate energetic sum or difference of values*

---

**Description**

Returns energetic sum or difference of values

**Usage**

dbsum(x, y, operator)

**Arguments**

x	is first value or vector
y	is second value or vector
operator	is 1 for sum and -1 for difference

**Value**

vector of values contain energetic sum or difference

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate energetic sum or difference of values

dbsum(x = 55, y = 33, operator = 1)
dbsum(x = c(55 , 66), y = c(45, 50), operator = 1)

dbsum(x = c(70 , 68), y = c(55, 66), operator = -1)
```

deco.time

*Time decomposition*

---

**Description**

Trasform time from hours, minutes and seconds to seconds

**Usage**

```
deco.time(x, y, z, verbose = TRUE)
```

**Arguments**

x	are hours
y	are minutes
z	are seconds
verbose	logic argument that on or off message (default is TRUE)

**Value**

time decomposition in seconds

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Convert H:M:S to seconds
x = 5 # Hours
y = 25 # minutes
z = 50 # seconds

deco.time(x = x, y = y, z = z)
```

---

dfBW*Table's 1/3 octave bandwidth*

---

**Description**

Table's 1/3 octave bandwidth

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>



---

dfImpulsiveTrasform	<i>Trasform impulsive dataframe (100 ms samples) in dataframe (1s samples)</i>
---------------------	--

---

**Description**

Returns a dataframe (1s samples)

**Usage**

```
dfImpulsiveTrasform(dfImpulsive, statistic = energetic.mean)
```

**Arguments**

dfImpulsive is a dataframe for impulse (data acquired at 100 ms)  
statistic is energetic mean (default)

**Value**

dataframe

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data("dataset_impulsive1")  
#data("dataset_impulsive2")  
  
dfImpulsiveTrasform(dfImpulsive = dataset_impulsive1,  
                    statistic = energetic.mean)[1:5, ]
```

---

energetic.mean	<i>Logarithmic mean</i>
----------------	-------------------------

---

**Description**

Calculate logarithmic mean

**Usage**

```
energetic.mean(x)
```

**Arguments**

x is a vector of value in decibel (dB)

**Value**

logarithmic mean

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate energetic mean of vector's values of Leq
energetic.mean(c(55, 88, 66, 51, 70))
```

---

energetic.min

*Function that calculate min value*

---

**Description**

calculate min value

**Usage**

```
energetic.min(y)
```

**Arguments**

y is a numeric vector

**Value**

energetic min vector value

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

energetic_w.mean	<i>Weigth logarithmic mean</i>
------------------	--------------------------------

---

**Description**

Calculate weigth logarithmic mean respect to time

**Usage**

```
energetic_w.mean(x, t)
```

**Arguments**

x	is a vector of value in decibel (dB)
t	is a vector of time string "HH:MM:SS"

**Value**

weigth logarithmic mean respect to time

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate weight energetic mean  
energetic_w.mean(x = c(55.0, 70.0) , t = c("03:55:22", "01:33:12"))
```

---

exampleHourlyData	<i>Noise hourly data of misure in environmental open space</i>
-------------------	--

---

**Description**

Noise hourly data of misure in environmental open space

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

ExtractIndexMark      *Extract index and name of markers*

---

**Description**

Returns a list of index and name

**Usage**

```
ExtractIndexMark(filemarks, dataset, mp)
```

**Arguments**

filemarks      is a dataframe with date and markers  
dataset        is dataframe in analysis  
mp             is a name of misure point

**Value**

list of index and names

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data(PTFA)  
#data("markers")  
  
ExtractIndexMark(filemarks = markers , dataset = PTFA, mp = "PTFA")
```

---

HolidaysDate      *Calculate Holidays date (Gregorian calendar)*

---

**Description**

Returns a vector of holiday dates (Gregorian calendar)

**Usage**

```
HolidaysDate(year_holiday)
```

**Arguments**

year\_holiday is year example "2022" like character

**Value**

string vector of date

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
HolidaysDate(2022)
```

---

HourlyEmean	<i>Calculate hourly energetic mean</i>
-------------	--

---

**Description**

Returns a dataframe with hourly energetic mean

**Usage**

```
HourlyEmean(df, variable, timeZone = "Europe/Rome")
```

**Arguments**

df is a dataframe with date (Y-m-d H:M:S) and variables

variable is a variable name

timeZone is time zone default is Europe/Rome

**Value**

dataframe of hourly energetic mean values

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data(PTFA)
```

```
HourlyEmean(df = PTFA, variable = "LAeq")
```

---

IntrusiveIndex                      *Calculate Intrusive Index (UNI/TS 11844 march 2022)*

---

### Description

Returns a number

### Usage

```
IntrusiveIndex(dfa, dfr, BW)
```

### Arguments

dfa                      is a dataframe Lfa (enviromental sound levels) 1/3 octave spectra data  
dfr                      is a dataframe Lfr (residual sound levels) 1/3 octave spectra data  
BW                      a vector of 1/3 octave bandwidth data

### Value

string of intrusive index

### Author(s)

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

### Examples

```
# Calculation of the intrusiveness index

library(OpeNoise)
library(lubridate)

data("dataset_impulsive1")
data("dfBW")

# dataset handling
df_Imp_sec <- dfImpulsiveTrasform(dataset_impulsive1,
                                statistic = energetic.mean)
df_Imp_sec$date <- ymd_hms(df_Imp_sec$date, tz = "Europe/Rome")

# extraction of frequency bands from the dataset
freqDF <- df_Imp_sec[, grep("LZe\\.", names(df_Imp_sec))]

#####

#                      INTRUSIVENESS INDEX CALCULATION FUNCTION
```

```
#####
dfa <- freqDF # Environmental dataset simulation
dfr <- freqDF

# Residual dataset simulation by subtracting 4 from dfa
dfr[c(5,8,12,15), ] <- dfr[c(5,8,12,15), ] - 4

BW <- dfBW$BW # bandwidth

# application of the function
IntrusiveIndex(dfa, dfr, BW)
```

---

iso	<i>Parameters table of equal loudness curve A (ISO 226:1987 “Acoustics – Normal equal-loudness-level contours”)</i>
-----	---

---

### Description

Parameters table of equal loudness curve A (ISO 226:1987 “Acoustics – Normal equal-loudness-level contours”)

### Author(s)

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
 Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

LdenCalculator	<i>Calculate daily or total Lden (Day-evening-night level)</i>
----------------	--

---

### Description

Returns a dataframe with Lden

### Usage

```
LdenCalculator(dataframe, variable, type = "daily", ...)
```

### Arguments

dataframe	is a dataframe
variable	is name of variable
type	is "daily" or "total"
...	is another arguments

**Value**

dataframe with Lden values

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data("exampleHourlyData")
```

```
LdenCalculator(dataframe = exampleHourlyData, variable = "leq",
               type = "daily")
```

```
LdenCalculator(dataframe = exampleHourlyData, variable = "leq",
               type = "total")
```

---

markers

*Dataset with markers*

---

**Description**

Dataset with markers

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

Maskapply

*Add index and name of markers in misure dataframe*

---

**Description**

Returns a dataframe

**Usage**

```
Maskapply(filemarks, dataset, mp)
```

**Arguments**

filemarks is a dataframe with date and markers

dataset is dataframe in analysis

mp is a name of misure point



**Value**

dataframe with add marker column

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data(PTFA)
#data(markers)
```

```
Maskapply(filemarks = markers, dataset = PTFA, mp = "PTFA")[1:10, c(1, 2, 45)]
```

---

P1FA

*Noise data of misure in house open window condition*

---

**Description**

Noise data of misure in house open window condition

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

P1FC

*Noise data of misure in house close window condition*

---

**Description**

Noise data of misure in house close window condition

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

PlotNoiseTHcompare     *Plot time history and compare frequency components*

---

### Description

Returns a plot

### Usage

```
PlotNoiseTHcompare(  
  df,  
  variable,  
  listvar = NULL,  
  mp,  
  runleq = TRUE,  
  y_lim = c(20, 80)  
)
```

### Arguments

df	is a dataframe
variable	is Leq or another variable to plot first
listvar	are names of frequency component you want compare
mp	is a misure point
runleq	is logical value that plot running leq line
y_lim	y axe range

### Value

ggplot object

### Author(s)

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

### Examples

```
#data(P1FA)  
  
PlotNoiseTHcompare(df = P1FA ,  
  variable = "LAeq",  
  listvar = c("LZFmin.100",  
             "LZFmin.250"), mp = "P1FA",  
  runleq = TRUE,  
  y_lim = c(30, 70))
```

---

PlotNoiseTimeHistory *Plot time history of noise misure with marker and running Leq*

---

### Description

Returns a time history plot

### Usage

```
PlotNoiseTimeHistory(  
  df = NULL,  
  variable = NULL,  
  filemarks = NULL,  
  escl_marks = NULL,  
  mp,  
  y_lim = c(20, 80)  
)
```

### Arguments

df	is a dataframe with date, leq and markers
variable	is a string name of column you want plot
filemarks	is a dataframe with date and markers
escl_marks	is mark that you want esclude in plot
mp	is a name of misure point
y_lim	y axes range

### Value

ggplot object

### Author(s)

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

### Examples

```
#data(P1FA)  
#data(markers)  
  
PlotNoiseTimeHistory(df = P1FA, variable = "LAeq", mp = "P1FA", y_lim = c(40, 65))  
  
PlotNoiseTimeHistory(df = P1FA, variable = "LAeq", mp = "P1FA",  
  filemarks = markers, y_lim = c(40, 65))
```

```
PlotNoiseTimeHistory(df = P1FA, variable = "LAeq", mp = "P1FA", escl_marks = "escludi",  
y_lim = c(40, 65))
```

---

PlotSpectrogram	<i>Plot spectrogram</i>
-----------------	-------------------------

---

### Description

Returns a spectrogram

### Usage

```
PlotSpectrogram(df, cols, plot_title = NULL)
```

### Arguments

df	is a dataframe
cols	is cols index to plot
plot_title	is title of plot

### Value

ggplot object

### Author(s)

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

### Examples

```
#data(P1FA)  
  
PlotSpectrogram(df = P1FA, cols = c(3:38) , plot_title = "Spettrogram LZFmin")
```

---

PTFA	<i>Noise data of misure in house open window condition</i>
------	--

---

### Description

Noise data of misure in house open window condition

### Author(s)

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

PTFC

*Noise data of misure in house close window condition*

---

**Description**

Noise data of misure in house close window condition

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

---

RoundTo

*Round to Multiple*

---

**Description**

Returns a number rounded to the nearest specified multiple.

**Usage**

```
RoundTo(x, multiple = 1, FUN = round)
```

**Arguments**

x	is a vector of value in decibel (dB)
multiple	numeric. The multiple to which the number is to be rounded. Default is 1.
FUN	the rounding function as character or as expression. Can be one out of trunc, ceiling, round (default) or floor.

**Value**

value or vector of values rounded

**Author(s)**

Andri Signorell <andri@signorell.net>

**Examples**

```
#data("P1FA")
```

```
RoundTo(x = P1FA$LReq, multiple = 0.5)[1:10]
```

---

runningLeq	<i>Calculate running Leq</i>
------------	------------------------------

---

**Description**

Returns a vector of energetic mean of Leq.....

**Usage**

```
runningLeq(x)
```

**Arguments**

x is a vector of values in decibel (dB)

**Value**

vector of mobile energetic average values

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data("P1FA")
runningLeq(x = P1FA$LAeq)[1:10]
```

---

search.tone	<i>Function research pure tone</i>
-------------	------------------------------------

---

**Description**

research pure tone

**Usage**

```
search.tone(x, statistic = energetic.mean, plot.tone = FALSE)
```

**Arguments**

x is a dataframe with lfmin...  
 statistic is statistic used default is energetic.mean  
 plot.tone is logic argument default is false don't plot result

**Value**

plot of 1/3 octave frequency and isofonic curve A (ISO 226:1987)

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Search pure tone from PTFA dataset
# dataframe have to contain date and LLfmin (minor value of Linear level frequency)
search.tone(PTFA[, c(3:38)], plot.tone = FALSE)

# Plot result
search.tone(PTFA[, c(3:38)], plot.tone = TRUE)
```

---

searchImpulse	<i>Search impulsive event</i>
---------------	-------------------------------

---

**Description**

Returns a list with dataframe of peaks impulsive and a plot

**Usage**

```
searchImpulse(df, cri1 = 6, cri2 = -10, Threshold = 30)
```

**Arguments**

df	is a impulse dataframe, samples of 100 ms
cri1	is first criteria 6dB (LAI <sub>max</sub> - LAS <sub>max</sub> > 6dB)
cri2	is second criteria -10dB ((LAF <sub>max</sub> - 10dB) < 1s)
Threshold	is minimum level for detect peaks

**Value**

list contain a dataframe of peaks values and a plot of it

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>

Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
#data("dataset_impulsive1")
#data("dataset_impulsive2")

searchImpulse(df = dataset_impulsive1)
searchImpulse(df = dataset_impulsive2)
```

---

**SELcalc***Calculate SEL (Single Event Level)*

---

**Description**

Returns SEL

**Usage**

```
SELcalc(x, t)
```

**Arguments**

x	is value in dB
t	is period in second

**Value**

value of cumulative energy in 1 second

**Author(s)**

Pasquale Scordino <p.scordino@arpa.piemonte.it>  
Simone Sperotto <s.sperotto@arpa.piemonte.it>

**Examples**

```
# Calculate SEL (Single Event Level)

SELcalc(x = 66.8, t = 938)
```



# Index

## \* data

AcousticWeightingTable, 3  
dataset\_impulsive1, 6  
dataset\_impulsive2, 7  
dfBW, 8  
exampleHourlyData, 11  
iso, 15  
markers, 16  
P1FA, 17  
P1FC, 17  
PTFA, 20  
PTFC, 21

AcousticQuantilePlot, 2  
AcousticWeightingTable, 3  
AcuDNPercentile, 4  
AcuPercentile, 5  
avr.day.night, 5

dataset\_impulsive1, 6  
dataset\_impulsive2, 7  
dbsum, 7  
deco.time, 8  
dfBW, 8  
dfImpulsiveTrasform, 9

energetic.mean, 9  
energetic.min, 10  
energetic\_w.mean, 11  
exampleHourlyData, 11  
ExtractIndexMark, 12

HolidaysDate, 12  
HourlyEmean, 13

IntrusiveIndex, 14  
iso, 15

LdenCalculator, 15

markers, 16

Maskapply, 16

P1FA, 17  
P1FC, 17  
PlotNoiseTHcompare, 18  
PlotNoiseTimeHistory, 19  
PlotSpectrogram, 20  
PTFA, 20  
PTFC, 21

RoundTo, 21  
runningLeq, 22

search.tone, 22  
searchImpulse, 23  
SELcalc, 24