

Appendix A

List of R Functions

List of `audio` 0.1.5 (Urbanek 2013), `monitor` 1.0.5 (Katz et al. 2016b), `phonTools` 0.2.2.1 (Barreda 2015), `seewave` 2.1.0 (Sueur et al. 2008a), `signal` 0.7.6 (Ligges et al. 2015), `soundecology` 1.3.2 (Villanueva-Rivera and Pijanowski 2016), `soundgen` 1.1.0 (Anikin 2017), `tuneR` 1.3.1 (Ligges et al. 2014), and `warbleR` 1.1.5 (Araya-Salas and Smith-Vidaurre 2016) functions used in this book and grouped by themes. Some functions can appear in several themes. The description comes from the description field of the documentation as provided by the package authors. A complete definition of each function is included in the related package documentation. The number of the main chapter where each function is introduced is given in the last column of the table.

Name	Package	Description	Chapter
<i>Input/output</i>			
<code>audio.drivers()</code>	<code>audio</code>	Lists all currently loaded and available audio drivers	4
<code>checkwavs()</code>	<code>warbleR</code>	Checks <code>.wav</code> files	4
<code>current.audio.driver()</code>	<code>audio</code>	Returns the name of the currently active audio driver	4
<code>getWavPlayer()</code>	<code>tuneR</code>	Gets the default player for <code>.wav</code> files	4
<code>loadsound()</code>	<code>phonTools</code>	Allows <code>.wav</code> files to be loaded into R	4
<code>load.wave()</code>	<code>audio</code>	Loads a <code>.wav</code> file	4
<code>load.audio.driver()</code>	<code>audio</code>	Attempts to load a modular audio driver	4
<code>listen()</code>	<code>seewave</code>	Plays a sound wave	4
<code>mp32wav()</code>	<code>warbleR</code>	Converts <code>.mp3</code> files to <code>.wav</code>	4
<code>play()</code>	<code>audio</code>	Plays <code>audioSample</code> objects	4

(continued)

Name	Package	Description	Chapter
<code>play()</code>	tuneR	Plays Wave objects	4
<code>playlist()</code>	seewave	Runs a playlist of sound files	4
<code>playsound()</code>	phonTools	Plays sounds in R using VLC player	4
<code>pause()</code>	audio	Pauses (stops) audio recording or playback	4
<code>querxc()</code>	warbler	Accesses Xeno-Canto recordings and metadata	4
<code>readMP3()</code>	tuneR	Reads an MPEG-2 layer 3 file into a Wave object	4
<code>readWave()</code>	tuneR	Reads and writes .wav files	4
<code>record()</code>	audio	Records audio	4
<code>resume()</code>	audio	Resumes previously paused audio recording or playback	4
<code>rewind()</code>	audio	Rewinds audio recording or playback	4
<code>savewav()</code>	seewave	Saves audio data as .wav file	4
<code>save.wave()</code>	audio	Saves audioSample objects as .wav file	4
<code>set.audio.driver()</code>	audio	Selects an audio driver as the current driver	4
<code>setWavPlayer()</code>	tuneR	Sets the default player for .wav files	4
<code>sox()</code>	seewave	Calls externally SoX	3
<code>wait()</code>	audio	Waits for an event during a recording session	4
<code>wav2flac()</code>	seewave	wav-flac file conversion	4
<code>writesound()</code>	phonTools	Creates a WAV file from a numeric vector or sound object	4
<code>writeWave()</code>	tuneR	Reads and writes Wave files	4
<code>xcmaps()</code>	warbler	Maps Xeno-Canto recordings by species	4
<i>Objects</i>			
<code>as.audioSample()</code>	audio	Converts an object into an audio sample object	4
<code>audioSample()</code>	audio	audiosample class and constructor	4
<code>duration()</code>	seewave	Duration of a time wave	4
<code>equalWave()</code>	tuneR	Checks for some kind of equality of objects of class Wave	4
<code>makesound()</code>	phonTools	Creates a sound object from a numeric vector.	4
<code>MCnames()</code>	tuneR	Defaults channel ordering for multichannel wave files (WaveMC objects)	6
<code>nchannel()</code>	tuneR	Number of channels	6
<code>updateWave()</code>	tuneR	Updates old Wave objects for use with new versions of tuneR	4
<code>Wave()</code>	tuneR	Constructors and coercion for class Wave objects	4
<code>WaveMC()</code>	tuneR	Constructors and coercion for class WaveMC objects	4

(continued)

Name	Package	Description	Chapter
<i>Edition</i>			
<code>addsilw()</code>	seewave	Adds or inserts a silence section	6
<code>bind()</code>	tuneR	Concatenates Wave objects	6
<code>channel()</code>	tuneR	Channel conversion for Wave objects	6
<code>crossFade()</code>	soundgen	Joins two waveforms by cross-fading	6
<code>cutw()</code>	seewave	Cuts a section of a time wave	6
<code>deletew()</code>	seewave	Deletes a section of a time wave	6
<code>downsample()</code>	tuneR	Downsamples a Wave or WaveMC object	6
<code>equalWave()</code>	tuneR	Checks Wave objects	4
<code>extractWave()</code>	tuneR	Extractor for Wave objects	6
<code>mono()</code>	tuneR	Converts stereo to mono and vice versa	6
<code>mutew()</code>	seewave	Replaces time wave data by 0 values	6
<code>normalize()</code>	tuneR	Rescales the range of values	6
<code>noSilence()</code>	tuneR	Cuts off silence from a Wave object	6
<code>panorama()</code>	tuneR	Narrows the panorama of a stereo sample	6
<code>pastew()</code>	seewave	Pastes a time wave to another one	6
<code>prepComb()</code>	tuneR	Prepares the combination/concatenation of Wave objects	6
<code>repw()</code>	seewave	Repeats a time wave	6
<code>resamp()</code>	seewave	Resamples a time wave	6
<code>rmoffset()</code>	seewave	Removes the offset of a time wave	6
<code>stereo()</code>	tuneR	Converts (extracts, joins) stereo to mono and vice versa	6
<code>zapsilw()</code>	seewave	Zaps silence periods of a time wave	6
<i>Time/amplitude</i>			
<code>acostat()</code>	seewave	Statistics on time and frequency STFT contours	11
<code>ama()</code>	seewave	Amplitude modulation analysis of a time wave	8
<code>corenv()</code>	seewave	Cross-correlation between two time wave envelopes	17
<code>crest()</code>	seewave	Crest factor	7
<code>discrets()</code>	seewave	Discretization of a numeric (time) series	10
<code>drawenv()</code>	seewave	Draws the amplitude envelope of a time wave	15
<code>dynoscillo()</code>	seewave	Dynamic oscillogram	5
<code>env()</code>	seewave	Amplitude envelope of a time wave	5
<code>oscillo()</code>	seewave	Shows a time wave as an oscillogram	5
<code>oscilloST()</code>	seewave	Shows a stereo time wave as oscillograms	6
<code>phaseplot()</code>	seewave	First, second, and third derivatives of a wave	10
<code>phaseplot2()</code>	seewave	Phase portrait of a wave	10
<code>powertrack()</code>	seewave	Creates a power track for a sound	5
<code>roughness()</code>	seewave	Roughness of a curve (a time wave or a spectrum)	10
<code>rugog()</code>	seewave	Rugosity of a time wave or time series	10
<code>segment()</code>	soundgen	Segments a sound	8
<code>setenv()</code>	seewave	Sets the amplitude envelope of a time wave to another one	15

(continued)

Name	Package	Description	Chapter
<code>timer()</code>	seewave	Time measurements of a time wave	8
<code>th()</code>	seewave	Temporal entropy	16
<i>Frequency</i>			
<code>bark2hz()</code>	tuneR	Frequency scale conversion	9
<code>bwfilter()</code>	seewave	Butterworth frequency filter	14
<code>ceps()</code>	seewave	Cepstrum or real cepstrum	10
<code>coh()</code>	seewave	Coherence between two time waves	17
<code>comb_filter()</code>	seewave	Combfilter	14
<code>corspec()</code>	seewave	Cross-correlation between two frequency spectra	17
<code>cutspec()</code>	seewave	Cuts a frequency spectrum	10
<code>diffcumspec()</code>	seewave	Difference between two cumulative frequency spectra	16
<code>diffspec()</code>	seewave	Difference between two frequency spectra	16
<code>drawfilter()</code>	seewave	Draws the frequency response of a filter	14
<code>fbands()</code>	seewave	Frequency band plot (equalizer plot)	10
<code>FF()</code>	tuneR	Estimation of fundamental frequencies from a Wspec object	13
<code>FFpure()</code>	tuneR	Estimation of fundamental frequencies from a Wspec object	13
<code>ffilter()</code>	seewave	Frequency filter	14
<code>fma()</code>	seewave	Frequency modulation analysis	13
<code>fpeaks()</code>	seewave	Frequency peak detection	10
<code>ftwindow()</code>	seewave	Fourier transform windows	9
<code>fund()</code>	seewave	Fundamental frequency	10
<code>H()</code>	seewave	Total entropy	16
<code>hz2bark()</code>	tuneR	Frequency scale conversion	9
<code>hz2mel()</code>	tuneR	Frequency scale conversion	9
<code>HzToSemitones()</code>	soundgen	Convert Hz to semitones	9
<code>ifreq()</code>	seewave	Instantaneous frequency	13
<code>itakura.dist()</code>	seewave	Itakura-Saito distance	16
<code>kl.dist()</code>	seewave	Kullback-Leibler distance	16
<code>ks.dist()</code>	seewave	Kolmogorov-Smirnov distance	16
<code>lifter()</code>	tuneR	Liftering of cepstra	12
<code>localpeaks()</code>	seewave	Local maximum frequency peak detection	10
<code>logspec.dist()</code>	seewave	Log-spectral distance	16
<code>meanspec()</code>	seewave	Mean frequency spectrum of a time wave	11
<code>mel()</code>	seewave	Hertz/Mel conversion	9
<code>mel2hz()</code>	tuneR	Frequency scale conversion	9
<code>melfcc()</code>	tuneR	MFCC calculation	12
<code>melfilterbank()</code>	seewave	Mel-frequency filter bank	12
<code>notesDict()</code>	soundgen	Conversion table from Hz to semitones above C0 to musical notation	9
<code>notefreq()</code>	seewave	Frequency of a musical note	9

(continued)

Name	Package	Description	Chapter
<code>noteFromFF()</code>	<code>tuneR</code>	Derives notes from frequencies	10
<code>periodogram()</code>	<code>tuneR</code>	Periodogram (spectral density) estimation on <code>Wave</code> objects	10
<code>preemphasis()</code>	<code>seewave</code>	Preemphasis frequency filter	14
<code>Q()</code>	<code>seewave</code>	Resonance quality factor of a frequency spectrum	10
<code>roughness()</code>	<code>seewave</code>	Roughness of a curve (a time wave or a spectrum)	10
<code>SAX()</code>	<code>seewave</code>	Symbolic aggregate approximation	10
<code>sfm()</code>	<code>seewave</code>	Spectral flatness measure	10
<code>sh()</code>	<code>seewave</code>	Spectral entropy	10
<code>semitonesToHz()</code>	<code>soundgen</code>	Convert semitones to Hz	9
<code>simspec()</code>	<code>seewave</code>	Similarity between two frequency spectra	16
<code>soundscapec()</code>	<code>seewave</code>	Soundscape frequency spectrum of a time wave	11
<code>spec2cep()</code>	<code>tuneR</code>	Spectra to cepstra conversion	12
<code>spec()</code>	<code>seewave</code>	Frequency spectrum of a time wave	10
<code>specan()</code>	<code>warbleR</code>	Measures acoustic parameters in batches of sound files	10
<code>specprop()</code>	<code>seewave</code>	Spectral properties	10
<code>squarefilter()</code>	<code>seewave</code>	Frequency response of a square filter	14
<code>syмба()</code>	<code>seewave</code>	Symbol analysis	10
<code>zcr()</code>	<code>seewave</code>	Zero-crossing rate	13
<i>Time/frequency</i>			
<code>acoustat()</code>	<code>seewave</code>	Statistics on time and frequency STFT contours	11
<code>analyze()</code>	<code>soundgen</code>	Analyzes sound	13
<code>audspec()</code>	<code>tuneR</code>	Frequency band conversion	12
<code>autoc()</code>	<code>seewave</code>	Short-time autocorrelation of a time wave	13
<code>batchBinMatch</code>	<code>monitoR</code>	Batch template detection	17
<code>batchCorMatch</code>	<code>monitoR</code>	Batch template detection	17
<code>binMatch</code>	<code>monitoR</code>	Calculates spectrogram template matching scores	17
<code>ccoh()</code>	<code>seewave</code>	Continuous coherence function between two time waves	17
<code>combineBinTemplates</code>	<code>monitoR</code>	Combines acoustic template lists	17
<code>combineCorTemplates</code>	<code>monitoR</code>	Combines acoustic template lists	17
<code>corMatch</code>	<code>monitoR</code>	Calculates spectrogram template matching scores	17
<code>covspectro()</code>	<code>seewave</code>	Covariance between two spectrograms	17
<code>deltas()</code>	<code>tuneR</code>	Calculates delta MFCC features	12
<code>dfDTW()</code>	<code>warbleR</code>	Dynamic time warping on dominant frequency contours	17
<code>dfreq()</code>	<code>seewave</code>	Dominant frequency of a time wave	13

(continued)

Name	Package	Description	Chapter
<code>dynspec()</code>	seewave	Dynamic sliding spectrum	11
<code>dynspectro()</code>	seewave	Dynamic sliding spectrogram	11
<code>eventEval</code>	monitoR	Evaluates detected events with known event sources and times	17
<code>ffDTW()</code>	warbleR	Dynamic time warping on fundamental frequency contours	17
<code>ffilter()</code>	seewave	Frequency filter	14
<code>findformants()</code>	phonTools	Finds formants given a sound or set of LPC coefficients	12
<code>findPeaks()</code>	monitoR	Finds score peaks and detections in a <code>templateScores</code> object	17
<code>formanttrack()</code>	phonTools	Creates a formant track for a sound	13
<code>ftwindow()</code>	seewave	Fourier transform windows	9
<code>fund()</code>	seewave	Fundamental frequency	10
<code>getPeaks()</code>	monitoR	Extracts detections or peaks from a <code>detectionList</code> object	17
<code>hilbert()</code>	seewave	Hilbert transform and analytic signal	13
<code>ifreq()</code>	seewave	Instantaneous frequency	13
<code>istft()</code>	seewave	Inverse of the short-time Fourier transform	11
<code>lpc()</code>	phonTools	Predicts autoregressive filter coefficients	12
<code>lspec()</code>	warbleR	Creates long spectrograms of whole sound files	11
<code>makeBinTemplate()</code>	monitoR	Makes an acoustic template	17
<code>makeCorTemplate()</code>	monitoR	Makes an acoustic template	17
<code>manualoc()</code>	warbleR	Interactive view of spectrograms	11
<code>manualoc.df()</code>	warbleR	Data frame of <code>manualoc()</code> selections	11
<code>meanspec()</code>	seewave	Mean frequency spectrum of a time wave	11
<code>melodyplot()</code>	tuneR	Plots a melody	13
<code>powspec()</code>	tuneR	Power spectrum	11
<code>periodogram()</code>	tuneR	Periodogram (spectral density) estimation on <code>Wave</code> objects	11
<code>pitchtrack()</code>	phonTools	Creates a pitch (fundamental frequency) track for a sound	13
<code>readBinTemplates()</code>	monitoR	Reads acoustic templates from a local disk	17
<code>readCorTemplates()</code>	monitoR	Reads acoustic templates from a local disk	17
<code>showPeaks()</code>	monitoR	Views or verifies detections or peaks	17
<code>spectro()</code>	seewave	2D spectrogram of a time wave	11
<code>spectrogram()</code>	phonTools	Creates and displays spectrograms	11
<code>spectrogram()</code>	soundgen	Spectrogram	11
<code>spectro3D()</code>	seewave	3D-spectrogram of a time wave	11
<code>stft.ext()</code>	seewave	Short-time Fourier transform using external C libraries	11

(continued)

Name	Package	Description	Chapter
<code>templateCutoff()</code>	monitoR	Queries or sets template cutoffs	17
<code>timeAlign()</code>	monitoR	Condenses detections or peaks from multiple templates	17
<code>viewSpec()</code>	monitoR	Interactively views and annotates spectrograms	11
<code>wf()</code>	seewave	Waterfall display	11
<code>TKEO()</code>	seewave	Teager-Kaiser energy operator	13
<code>zc()</code>	seewave	Instantaneous frequency of a time wave by zero crossing	13
<i>Indices</i>			
<code>ACI()</code>	seewave	Acoustic complexity index	16
<code>acoustic_complexity()</code>	soundecology	Acoustic complexity index	16
<code>acoustic_diversity()</code>	soundecology	Acoustic diversity index	16
<code>acoustic_evenness()</code>	soundecology	Acoustic evenness index	16
<code>AR()</code>	seewave	Acoustic richness index	16
<code>bioacoustic_index()</code>	soundecology	Bioacoustic index	16
<code>diffcumspec()</code>	seewave	Difference between two cumulative frequency spectra	16
<code>diffspec()</code>	seewave	Difference between two frequency spectra	16
<code>fpeaks()</code>	seewave	Frequency peaks detection	10
<code>H()</code>	seewave	Total entropy	16
<code>itakura.dist()</code>	seewave	Itakura-Saito distance	16
<code>kl.dist()</code>	seewave	Kullback-Leibler distance	16
<code>ks.dist()</code>	seewave	Kolmogorov-Smirnov distance	16
<code>logspec.dist()</code>	seewave	Log-spectral distance	16
<code>M()</code>	seewave	Amplitude index	16
<code>multiple_sounds()</code>	soundecology	Multiple sound files	16
<code>ndsi()</code>	soundecology	Normalized difference soundscape index	16
<code>NDSI()</code>	seewave	Normalized difference soundscape index	16
<code>roughness()</code>	seewave	Roughness of a curve (a time wave or a spectrum)	10
<code>SAX()</code>	seewave	Symbolic aggregate approximation	10
<code>sfm()</code>	seewave	Spectral flatness measure	10
<code>sh()</code>	seewave	Spectral entropy	16
<code>simspec()</code>	seewave	Similarity between two frequency spectra	16
<code>th()</code>	seewave	Temporal entropy	16
<i>Filters and modifications</i>			
<code>afilter()</code>	seewave	Amplitude filter	15
<code>bwfilter()</code>	seewave	Butterworth frequency filter	14
<code>echo()</code>	seewave	Echo generator	15

(continued)

Name	Package	Description	Chapter
<code>fadew()</code>	<code>seewave</code>	Fade in and fade out of a time wave	6
<code>fir()</code>	<code>seewave</code>	Finite impulse response filter	14
<code>lfs()</code>	<code>seewave</code>	Linear frequency shift	15
<code>preemphasis()</code>	<code>seewave</code>	Preemphasis frequency filter	14
<code>revw()</code>	<code>seewave</code>	Time reverse of a time wave	6
<code>rmam()</code>	<code>seewave</code>	Removes the amplitude modulations of a time wave	15
<code>rmnoise()</code>	<code>seewave</code>	Removes Gaussian noise	14
<code>rmoffset()</code>	<code>seewave</code>	Removes the offset of a time wave	6
<i>Synthesis</i>			
<code>chirp()</code>	<code>signal</code>	Creates a chirp signal	18
<code>getRolloff()</code>	<code>soundgen</code>	Controls roll-off of harmonics	18
<code>noise()</code>	<code>tuneR</code>	Creates <code>Wave</code> objects of special waveforms	18
<code>noisew()</code>	<code>seewave</code>	Generates noise	18
<code>pulsw()</code>	<code>seewave</code>	Generates rectangle pulse	18
<code>sawtooth()</code>	<code>tuneR</code>	Creates <code>Wave</code> objects of special waveforms	18
<code>silence()</code>	<code>tuneR</code>	Creates <code>Wave</code> objects of special waveforms	18
<code>sine()</code>	<code>tuneR</code>	Creates <code>Wave</code> objects of special waveforms	18
<code>square()</code>	<code>tuneR</code>	Creates <code>Wave</code> objects of special waveforms	18
<code>soundgen()</code>	<code>soundgen</code>	Generates a sound	18
<code>soundgen_app()</code>	<code>soundgen</code>	<code>soundgen</code> shiny app	18
<code>synth()</code>	<code>seewave</code>	Synthesis of time wave (additive model)	18
<code>synth2()</code>	<code>seewave</code>	Synthesis of time wave (tonal model)	18
<code>vowelsynth()</code>	<code>phonTools</code>	Creates synthetic vowels using a cascade formant synthesizer	18
<i>Extra</i>			
<code>attenuation()</code>	<code>seewave</code>	Generates sound intensity attenuation data	7
<code>convSPL()</code>	<code>seewave</code>	Converts sound pressure level in other units	7
<code>dBscale()</code>	<code>seewave</code>	dB color scale for a spectrogram display	11
<code>dBweight()</code>	<code>seewave</code>	dB weightings	7
<code>meandB()</code>	<code>seewave</code>	Mean of dB values	7
<code>micsens()</code>	<code>seewave</code>	Microphone sensitivity	7
<code>moredB()</code>	<code>seewave</code>	Addition of dB values	7
<code>notenames()</code>	<code>tuneR</code>	Generates note names from numbers	9
<code>octaves()</code>	<code>seewave</code>	Octaves series	9
<code>quantMerge()</code>	<code>tuneR</code>	Quantization of notes to produce sheet music	13
<code>quantize()</code>	<code>tuneR</code>	Quantization of notes to produce sheet music	13
<code>quantplot()</code>	<code>tuneR</code>	Plots the quantization of a melody	13
<code>rms()</code>	<code>seewave</code>	Root mean square	7
<code>songmeter()</code>	<code>seewave</code>	Reads and interprets <code>songmeter</code> file name	4

Appendix B

Sound Samples

Here are information of the sounds used as samples, or examples. Several sounds were very kindly provided by colleagues.

Name: bat

Data source: file 'Pipistrellus_kuhlui.wav'

Description: one call of the European bat *Pipistrellus kuhlii*

Author: Jean-François Julien

Location: Cournonterral, France

Properties:

```
Wave Object
Number of Samples:      3841
Duration (seconds):     0.02
Samplingrate (Hertz):  192000
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):   16
```

Name: cockroach

Data source: file 'Elliptorhina_chopardi.wav'

Description: one male courtship call of the hissing cockroach from Madagascar *Elliptorhina chopardi*

Author: Jérôme Sueur

Location: laboratory, Orsay, France

Properties:

```
Wave Object
Number of Samples:      19137
Duration (seconds):     0.43
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Mono
PCM (integer format):   TRUE
Bit (8/16/24/32/64):  16
```

Name: elephant

Data source: file 'Loxodonta_africana.wav'

Description: eight sounds used in Gilbert et al. (2014, figures 2–3). Sounds 1–4 were produced produced with a 3 m hose pipe. Sounds 1 and 2 are low-amplitude, “non-brassy” sounds produced with a low-intensity source. Sounds 3 and 4 are high-amplitude, “brassy” sounds produced with a high (increasing)-intensity source. Sounds 5–8 are trumpet calls recorded from a 20-year-old female African elephant. Sounds 5 and 6 are low-amplitude, “non-brassy” trumpet calls. Sounds 7 and 8 are high-amplitude trumpet calls.

Author: Joël Gilbert

Location: Beauval Zoo, Saint-Aignan, France for the elephant trumpet calls

Properties:

```
Wave Object
Number of Samples:      366706
Duration (seconds):     8.32
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Mono
PCM (integer format):   TRUE
Bit (8/16/24/32/64):  16
```

Name: face

Data source: file 'synth-face.wav'

Description: a synthetic sound made of various items modulated in amplitude and frequency. The sound was created with the code detailed in Sect. 18.6.3.2.

Author: Jérôme Sueur

Location: *in silico*

Properties:

```
Wave Object
Number of Samples:      44100
Duration (seconds):     1
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Mono
PCM (integer format):   TRUE
Bit (8/16/24/32/64):  32
```

Name: femo
 Data source: file 'Allobates_femoralis.wav'
 Description: four vocalizations of the South-American dart poison frog *Allobates femoralis*
 Author: Jérôme Sueur
 Location: Kaw, Guiana, France
 Properties:

```
Wave Object
Number of Samples:      61740
Duration (seconds):    1.4
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: femo
 Data source: file 'Allobates_femoralis_2015-11-10_161500_GFT.wav'
 Description: 28 vocalizations of the South-American dart poison frog *Allobates femoralis*
 Author: Jérôme Sueur
 Location: Location: Kaw, Guiana, France
 Properties:

```
Wave Object
Number of Samples:      1323000
Duration (seconds):    30
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: forest
 Data source: file 'forest.wav'
 Description: one minute of soundscape recording in the tropical forest during the first part of the night
 Author: Jérôme Sueur
 Location: Kaw, Guiana, France
 Properties:

```
Wave Object
Number of Samples:      2646000
Duration (seconds):    60
```

(continued)

```

Samplingrate (Hertz): 44100
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16

```

Name: frog

Data source: file 'Eleutherodactylus_martinicensis.wav'

Description: 17 two-note vocalizations of the Martinique Robber frog *Eleutherodactylus martinicensis*

Author: Renaud Boistel

Location: Lesser Antilles, France

Properties:

```

Wave Object
Number of Samples: 316602
Duration (seconds): 19.79
Samplingrate (Hertz): 16000
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16

```

Name: guiana

Data source: files

```

'M-XV_20101125_000000.wav', 'M-XV_20101125_010000.wav',
'M-XV_20101125_020000.wav' 'M-XV_20101125_030000.wav',
'M-XV_20101125_040000.wav' 'M-XV_20101125_050000.wav',
'M-XV_20101125_060000.wav', 'M-XV_20101125_070000.wav',
'M-XV_20101125_080000.wav', 'M-XV_20101125_090000.wav',
'M-XV_20101125_100000.wav', 'M-XV_20101125_110000.wav',
'M-XV_20101125_120000.wav', 'M-XV_20101125_130000.wav',
'M-XV_20101125_140000.wav', 'M-XV_20101125_150000.wav',
'M-XV_20101125_160000.wav', 'M-XV_20101125_170000.wav',
'M-XV_20101125_180000.wav', 'M-XV_20101125_190000.wav',
'M-XV_20101125_200000.wav' 'M-XV_20101125_210000.wav',
'M-XV_20101125_220000.wav', 'M-XV_20101125_230000.wav'

```

Description: 24 stereo files recorded in the tropical forest in French Guiana every hour from 00:00 to 23:00 on the 25 November 2010

Author: Amandine Gasc and Jérôme Sueur

Location: Nouragues, Guiana, France

Properties: first file

```
Wave Object
Number of Samples:      2646016
Duration (seconds):    60
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Stereo
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: hello

Data source: file 'hello.wav'

Description: English word "hello" said by a 7-year old French native girl

Author: Jérôme Sueur

Location: Paris, France

Properties:

```
Wave Object
Number of Samples:      38400
Duration (seconds):    0.8
Samplingrate (Hertz):  48000
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: noise

Data source: file 'noise.wav'

Description: a white noise broadcast by a low-quality loudspeaker and recorded with a high-quality microphone in an anechoic chamber.

Author: Diego Llusia

Location: Paris, France

Properties:

```
Wave Object
Number of Samples:      44100
Duration (seconds):    1
Samplingrate (Hertz):  44100
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: peewit
 Data source: seewave accompanying data
 Description: song emitted by a peewit (lapwing) male *Vanellus vanellus*
 Author: Thierry Aubin
 Location: France
 Properties:

```
Wave Object
Number of Samples:      15561
Duration (seconds):    0.71
Samplingrate (Hertz):  22050
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: pellucens
 Data source: seewave accompanying data
 Description: two stridulations of the Italian tree cricket *Oecanthus pellucens*
 Author: Jérôme Sueur
 Location: Brain-sur-Allonnes, France
 Properties:

```
Wave Object
Number of Samples:      36476
Duration (seconds):    3.31
Samplingrate (Hertz):  11025
Channels (Mono/Stereo): Mono
PCM (integer format):  TRUE
Bit (8/16/24/32/64):  16
```

Name: sheep
 Data source: seewave accompanying data
 Description: a single bleat of the Préalpes-du-Sud *Ovis aries*
 Author: Frédéric Sèbe
 Location: Brouessy, France
 Properties:

```
Wave Object
Number of Samples:      19764
Duration (seconds):    2.47
Samplingrate (Hertz):  8000
```

(continued)

```
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16
```

Name: tico

Data source: seewave accompanying data

Description: four notes of the rufous-collared sparrow *Zonotrichia capensis*

Author: Thierry Aubin

Location: Brazil

Properties:

```
Wave Object
Number of Samples: 39578
Duration (seconds): 1.79
Samplingrate (Hertz): 22050
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16
```

Name: theremin

Data source: recording freely accessible at

<https://www.freesound.org/people/realthereimin/sounds/119007/>

Description: a frequency-modulated sound produced by a theremin, an electronic instrument

Author: –

Location: –

Properties:

```
Wave Object
Number of Samples: 626176
Duration (seconds): 14.2
Samplingrate (Hertz): 44100
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16
```

Name: toad

Data source: file 'Alytes_obstetricans.wav'

Description: four single-note vocalizations of the European midwife toad *Alytes*

obstetricans with important background noise due to wind and nocturnal insects (Orthoptera)

Author: Jérôme Sueur

Location: Badefols-sur-Dordogne, France

Properties:

```
Wave Object
Number of Samples: 264000
Duration (seconds): 5.5
Samplingrate (Hertz): 48000
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16
```

Name: tuningfork

Data source: file 'tuning-fork.wav'

Description: a synthetic 440Hz pure tone mimicking a A-tone tuning fork. The sound was created with `synth(f=44100, d=1, cf=440, output="Wave")` Author: Jérôme Sueur

Location: *in silico*

Properties:

```
Wave Object
Number of Samples: 44100
Duration (seconds): 1
Samplingrate (Hertz): 44100
Channels (Mono/Stereo): Mono
PCM (integer format): TRUE
Bit (8/16/24/32/64): 16
```


References

- Adler D, Murdoch D (2016) rgl: 3D visualization device system (OpenGL). <http://CRAN.R-project.org/package=rgl>, R package
- Aldersley A, Champneys A, Homer M, Robert D (2016) Quantitative analysis of harmonic convergence in mosquito auditory interactions. *J R Soc Interface* 13:20151007
- Anikin A (2017) soundgen: Parametric voice synthesis. <https://CRAN.R-project.org/package=soundgen>, R package 1
- Araya-Salas M, Smith-Vidaurre G (2016) warbleR: an R package to streamline analysis of animal acoustic signals. *Meth Ecol Evol* 8:184–191
- Barreda S (2015) phonTools: functions for phonetics in R. <https://cran.r-project.org/web/packages/phonTools/index.html>, R package
- Beeman K (1998) Digital signal analysis, editing, and synthesis. Springer, Berlin/Heidelberg, pp 59–103
- Bennet-Clark HC (1999) Which Qs to choose: questions of quality in bioacoustics? *Bioacoustics* 9:351–359
- Boelman NT, Asner GP, Hart PJ, Martin RE (2007) Multi-trophic invasion resistance in Hawaii: bioacoustics, field surveys, and airborne remote sensing. *Ecol Appl* 17:2137–2144
- Boersma P (1993) Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. In: *Proceedings of the Institute of Phonetic Sciences, Amsterdam*, vol 17, pp 97–110
- Bogert BP, Healy MJR, Tukey JW (1963) The quefrency analysis of time series for echoes: cepstrum, pseudo-autocovariance, cross-cepstrum, and saphé cracking. In: *Time series analysis*. Wiley, New York, pp 209–243
- Borcard D, Gillet F, Legendre P (2011) Numerical ecology with R. Springer, New York
- Bradbury JW, Vehrencamp SL (1998) Principles of animal communication. Sinauer Associates, Sunderland
- Butterworth S (1930) On the theory of filter amplifiers. *Exp Wirel Wirel Eng* 7:536–541
- Butts CT (2008) network: a package for managing relational data in r. *J Stat Softw* 24(2):2–36
- Carson JR (1922) Notes on the theory of modulation. *Proc Institute Radio Eng* 10:57–64
- Cazelles B (2004) Symbolic dynamics for identifying similarity between rhythms of ecological time series. *Ecol Lett* 7:755–763
- Chang W, Cheng J, Allaire J, Xie Y, McPherson J (2016) shiny: web application framework for R. <https://CRAN.R-project.org/package=shiny>, R package
- Chowning JM (1973) The synthesis of complex audio spectra by means of frequency modulation. *J Audio Eng Soc* 21:526–531

- Cooley JW, Tukey JW (1965) An algorithm for the machine computation of complex Fourier series. *Math Comput* 19:297–301
- Cooper MA, Dultsev FN, Minson T, Ostanin VP, Abell C, Klenerman D (2001) Direct and sensitive detection of a human virus by rupture event scanning. *Nat Biotechnol* 19:833–837
- Cryer JD, Chan KS (2008) Time series analysis with applications in R. Springer, New York
- Das A (2012) Signal conditioning. An introduction to continuous wave communication and signal processing. Springer, Berlin
- Davis SB, Mermelstein P (1980) Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans Acoust Speech Signal Process* 28:357–366
- Deecke VB, Janik VM (2006) Automated categorization of bioacoustic signals: avoiding perceptual pitfalls. *J Acoust Soc Am* 119:645–653
- Depraetere M, Pavoine S, Jiguet F, Gasc A, Duvail S, Sueur J (2012) Monitoring animal diversity using acoustic indices: implementation in a temperate woodland. *Ecol Indic* 13:46–54
- Deutsch D (2010) The paradox of pitch circularity. *Acoust Today* 6:8–15
- Dziak RP, Bohnenstiehl DR, Baker ET, Matsumoto H, Caplan-Auerbach J, Embley RW, Merle SG, Walker SL, Lau TK, Chadwick WW (2015) Long-term explosive degassing and debris flow activity at west mata submarine volcano. *Geophys Res Lett* 42:1480–1487
- Fawcett T (2006) An introduction to ROC analysis. *Pattern Recogn Lett* 27:861–874
- Felisberto P, Jesus SM, Zabel F, Santos R, Silva J, Gobert S, Beer S, Björk M, Mazzuca S, Procaccini G, Runcie JW, Champenois W, Borges AV (2015) Acoustic monitoring of O₂ production of a seagrass meadow. *J Exp Mar Biol Ecol* 464:75–87
- Fisher RA (1936) The use of multiple measurements in taxonomic problems. *Ann Eugen* 7:179–188
- Fitch W, Neubauer J, Herzog H (2002) Calls out of chaos: the adaptive significance of nonlinear phenomena in mammalian vocal production. *Anim Behav* 63:407–418
- Fletcher NH (1992) Acoustic systems in biology. Oxford University Press, Oxford
- Fristrup KM, Watkins WA (1992) Characterizing acoustic features of marine animal sounds. Tech. rep., Woods Hole Oceanographic Institution Technical Report WHOI-92-04
- Fuller S, Axel AC, Tucker D, Gage SH (2015) Connecting soundscape to landscape: which acoustic index best describes landscape configuration? *Ecol Indic* 58:207–215
- Gagliano M, Mancuso S, Robert D (2012) Towards understanding plant bioacoustics. *Trends Plant Sci* 17:323–325
- Gasc A, Sueur J, Jiguet F, Devictor V, Grandcolas P, Burrow C, Depraetere M, Pavoine S (2013a) Assessing biodiversity with sound: do acoustic diversity indices reflect phylogenetic and functional diversities of bird communities? *Ecol Indic* 25:279–287
- Gasc A, Sueur J, Pavoine S, Pellens R, Grandcolas P (2013b) Biodiversity sampling using a global acoustic approach: contrasting sites with microendemism in new caledonia. *PLoS ONE* 8:e65311
- Gasc A, Pavoine S, Lellouch L, Grandcolas P, Sueur J (2015) Acoustic indices for biodiversity assessments: analyses of bias based on simulated bird assemblages and recommendations for field surveys. *Biol Conserv* 191:306–312
- Gençay R, Selçuk F, Whitcher B (2001) An introduction to wavelets and other filtering methods in finance and economics. Academic Press, San Diego
- Gilbert J, Dalmont JP, Potier R, Reby D (2014) Is nonlinear propagation responsible for the brassiness of elephant trumpet calls? *Acta Acustica United Acustica* 100:734–738
- Giorgino T (2009) Computing and visualizing dynamic time warping alignments in R: the dtw package. *J Stat Softw* 31:1–24
- Gustafsson MV, Aref T, Kockum AF, Ekström MK, Johansson G, Delsing P (2014) Propagating phonons coupled to an artificial atom. *Science* 346:207–211
- Hartmann WM (1998) Signals, sound, and sensation. Springer, New York
- Hermann T, Hunt A, Neuhoff JG (eds) (2011) The sonification handbook. Logos Verlag, Berlin
- Hopp SL, Owren MJ, Evans CS (1998) Animal acoustic communication. Springer, Berlin/Heidelberg

- Ihaka R (2010) R: lessons learned, directions for the future. In: Joint statistical meetings proceedings
- Ihaka R, Gentleman R (1996) R: A language for data analysis and graphics. *J Comput Graph Stat* 5:299–314
- Josse J, Pagès J, Husson F (2008) Testing the significance of the RV coefficient. *Comput Stat Data Anal* 53:82–91
- Kabacoff R (2013) R in action, data analysis and graphics with R. Manning Publications, New York
- Kaiser JF (1990) On a simple algorithm to calculate the “energy” of a signal. In: International conference on acoustics, speech, and signal processing, ICASSP-90, pp 381–384
- Kantz H, Schreiber T (2003) Non linear time series analysis. Cambridge University Press, Cambridge
- Kasten EP, Gage SH, Fox J, Joo W (2012) The remote environmental assessment laboratory’s acoustic library: an archive for studying soundscape ecology. *Eco Inform* 12:50–67
- Katz J, Hafner SD, Donovan T (2016a) Assessment of error rates in acoustic monitoring with the R package monitor. *Bioacoustics* 25:177–196
- Katz J, Hafner SD, Donovan T (2016b) Tools for automated acoustic monitoring within the R package monitor. *Bioacoustics* 25:191–210
- Kendrick P, Lopez L, Waddington D, Young R (2016) Assessing the robust of soundscape complexity indices. In: 23rd international congress on sound and vibration, Athens, pp 1–8
- Klatt DH (1980) Software for a cascade/parallel formant synthesizer. *J Acoust Soc Am* 67:971–995
- Kvedalen E (2003) Signal processing using the Teager energy operator and other nonlinear operators. Master’s thesis, Department of Informatics, University of Oslo
- Larsen NJ, Wahlberg M (2017) Sound and sound sources. In: Comparative bioacoustics: an overview. Bentham Science, Sharjah, pp 3–61
- Lauterborn W, Parlitz U (1988) Methods of chaos physics and their application to acoustics. *J Acoust Soc Am* 84:1975–1993
- Legendre P, Anderson MJ (1999) Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. *Ecol Monogr* 69:1–24
- Lellouch L, Pavoine S, Jiguet F, Glotin H, Sueur J (2014) Monitoring temporal change of bird communities with dissimilarity acoustic indices. *Methods Ecol Evol* 5:495–505
- Lemon RE (1971) Vocal communication by the frog *Eleutherodactylus martinicensis*. *Copeia* 49:211–217
- Ligges U, Krey S, Mersmann O, Schnackenberg S (2014) tuneR: Analysis of music. <http://r-forge.r-project.org/projects/tuner>, R package
- Ligges U, Short T, Kienzle P (2015) signal: signal processing. <http://CRAN.R-project.org/package=signal>, R package
- Lin J, Keogh E, Lonardi S, Chiu B (2003) A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on research issues in data mining and knowledge discovery, DMKD ’03. ACM, New York, pp 2–11
- Magurran AE, McGill BJ (2011) Biological diversity, frontiers in measurement and assessment. Oxford University Press, Oxford
- Mallat S (2009) A wavelet tour of signal processing: the sparse way. Elsevier, Amsterdam
- Mbu Nyamsi RG, Aubin T, Bremond JC (1994) On the extraction of some time dependent parameters of an acoustic signal by means of the analytic signal concept. Its application to animal sound study. *Bioacoustics* 5:187–203
- McGregor PK (ed) (2005) Animal communication networks. Cambridge University Press, Cambridge
- Mellinger DK, Clark CW (2003) Blue whale (*Balaenoptera musculus*) sounds from the North Atlantic. *J Acoust Soc Am* 114(2):1108–1119
- Merchant ND, Frstrup KM, Johnson MP, Tyack PL, Witt MJ, Blondel P, Parks SE (2015) Measuring acoustic habitats. *Methods Ecol Evol* 6:257–265

- Meyer D, Buchta C (2016) proxy: distance and similarity measures. <https://CRAN.R-project.org/package=proxy>, R package
- Mezquida DA, Martinez JL (2009) Platform for beehives monitoring based on sound analysis. a perpetual warehouse for swarm's daily activity. *Span J Agric Res* 7:824–828
- Nason GP (2008) Wavelet methods in statistics with R. Springer, New York
- Oppenheim AV, Schaffer RW (1975) Digital signal processing. Prentice-Hall, Upper Saddle River
- Oppenheim A, Schaffer R (2004) From frequency to quefrency: a history of the cepstrum. *IEEE Signal Process Mag* 21:95–106
- Percival DB, Walden AT (2000) Wavelet methods for time series analysis. Cambridge University Press, Cambridge
- Pieretti N, Farina A, Morri FD (2011) A new methodology to infer the singing activity of an avian community: the acoustic complexity index (ACI). *Ecol Indic* 11:868–873
- Quatieri TF (2002) Discrete-time speech signal processing: principles and practice. Pearson, Noida
- Ramsay JO, Silverman BW (2005) Functional data analysis. Springer, New York
- Rice AN, Land BR, Bass AH (2011) Nonlinear acoustic complexity in a fish “two-voice” system. *Proc R Soc B: Biol Sci* 278:3762–3768
- Richter I, Koenders C, Auster HU, Frühauff D, Götz C, Heinisch P, Perschke C, Motschmann U, Stoll B, Altwegg K, Burch J, Carr C, Cupido E, Eriksson A, Henri P, Goldstein R, Lebreton JP, Mokashi P, Nemeth Z, Nilsson H, Rubin M, Szegö K, Tsurutani BT, Vallat C, Volwerk M, Glassmeier KH (2015) *Ann Geophys* 33:1031–1036
- Risset JC (1978) Paradoxes de hauteur. Tech. rep., IRCAM - Centre Georges Pompidou
- Robert P, Escoufier Y (1976) A unifying tool for linear multivariate statistical methods: the RV-coefficient. *J R Stat Soc* 25:257–265
- Rodriguez A, Gasc A, Pavoine S, Grandcolas P, Gaucher P, Sueur J (2014) Temporal and spatial variability of animal sound within a neotropical forest. *Eco Inform* 21:133–143
- Rossing TD (ed) (2007) Handbook of acoustics. Springer, New York
- Royer JY, Chateau R, Dziak R, Bohnenstiehl D (2015) Seafloor seismicity, Antarctic ice-sounds, cetacean vocalizations and long-term ambient sound in the Indian ocean basin. *Geophys J Int* 202:748–762
- Rumsey F, McCormick T (2002) Sound and recording. An introduction. Focal Press, Boston
- Russell DA (2000) On the sound field radiated by a tuning fork. *Am J Phys* 68:1139–1145
- Russell DA, Junell J, Ludwigsen DO (2013) Vector acoustic intensity around a tuning fork. *Am J Phys* 81:99–103
- Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans Acoust Speech Signal Process* 26:43–49
- Schafer RM (1977) The soundscape: our sonic environment and the tuning of the world. Destiny Books, Rochester
- Schloerke B, Crowley J, Cook D, Briatte F, Marbach M, Thoen E, Elberg A, Larmarange J (2017) GGally: Extension to ‘ggplot2’. <https://CRAN.R-project.org/package=GGally>, R package
- Senin P (2015) jmotif: tools for time series analysis based on symbolic aggregate discretization. <http://CRAN.R-project.org/package=jmotif>, R package
- Shannon CE (1949) Communication in the presence of noise. *Proc Inst Radio Eng* 37:10–21
- Shannon CE, Weaver W (1949) The mathematical theory of communication. Illinois University Press, Urbana
- Sharan VR, Moir TJ (2016) An overview of applications and advancements in automatic sound recognition. *Neurocomputing* 200:22–34
- Shepard RN (1964) Circularity in judgments of relative pitch. *J Acoust Soc Am* 36:2346–2353
- Shumway RH, Stoffer DS (2006) Time series analysis and its applications with R examples. Springer, Cham
- Snell RC, Milinazzo F (1993) Formant location from LPC analysis data. *IEEE Trans Speech Audio Process* 1:129–134
- Soetaert K (2014) diagram: functions for visualising simple graphs (networks), plotting flow diagrams. <https://CRAN.R-project.org/package=diagram>, R package
- Speaks CE (1999) Introduction to sound. Singular Publishing Group, San Diego/London

- Staszewski WJ, Robertson AN (2007) Time-frequency and time-scale analyses for structural health monitoring. *Philos Trans R Soc A: Math Phys Eng Sci* 365:449–477
- Stevens SS, Volkman J, Newman EB (1937) A scale for the measurement of the psychological magnitude pitch. *J Acoust Soc Am* 8:185–190
- Sueur J, Aubin T (2006) When males whistle at females: complex fm signals in cockroaches. *Naturwissenschaften* 93:500–505
- Sueur J, Farina A (2015) Ecoacoustics: the ecological investigation and interpretation of environmental sound. *Biosemiotics* 26:493–502
- Sueur J, Aubin T, Simonis C (2008a) seewave: a free modular tool for sound analysis and synthesis. *Bioacoustics* 18:213–226
- Sueur J, Pavoine S, Hamerlynck O, Duvail S (2008b) Rapid acoustic survey for biodiversity appraisal. *PLoS ONE* 3:e4065
- Sueur J, Mackie D, Windmill JFC (2011) So small, so loud: Extremely high sound pressure level from a pygmy aquatic insect (Corixidae, Micronectinae). *PLoS ONE* 6:e21089
- Sueur J, Farina A, Gasc A, Pieretti N, Pavoine S (2014) Acoustic indices for biodiversity assessment and landscape investigation. *Acta Acustica United Acustica* 100:772–781
- Sylvander M, Ponsolles C, Benahmed S, Fels JF (2007) Seismoacoustic recordings of small earthquakes in the Pyrenees: experimental results. *Bull Seismol Soci Am* 97:294–304
- Teetor P (2011) *R Cookbook*. O'Reilly, Sebastopol
- Tokuda IT (2017) Nonlinear dynamics and temporal analysis. In: *Comparative bioacoustics: an overview*. Bentham Science, Oak Park, pp 336–357
- Towsey M, Planitz B, Nantes A, Wimmer J, Roe P (2012) A toolbox for animal call recognition. *Bioacoustics* 21:107–125
- Towsey M, Wimmer J, Williamson I, Roe P (2014) The use of acoustic indices to determine avian species richness in audio-recordings of the environment. *Eco Inform* 21:110–119
- Urbanek S (2013) audio: audio interface for R. <https://CRAN.R-project.org/package=audio>, R package
- Venables WN, Ripley BD (2002) *Modern applied statistics with S*, 4th edn. Springer, New York
- Villanueva-Rivera LJ, Pijanowski BC (2016) soundecology: soundscape ecology. <https://CRAN.R-project.org/package=soundecology>, R package
- Villanueva-Rivera L, Pijanowski B, Doucette J, Pekin B (2011) A primer of acoustic analysis for landscape ecologists. *Landsc Ecol* 26:1233–1246
- Welch PD (1967) The use of fast Fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Trans Audio Electroacoust* 15:70–73
- Wickham H (2009) *ggplot2. Elegant graphics for data analysis*. Springer, Dordrecht
- Wickham H (2014) *Advanced R*. Chapman & Hall/CRC Press, Boca Raton
- Xie Y (2013) *Dynamic documents with R and knitr*. The R Series. Chapman & Hall/CRC, Boca Raton
- Zwicker E (1961) Subdivision of the audible frequency range into critical bands (frequenzgruppen). *J Acoust Soc Am* 33:248–248

Index

- .flac format, 94
- .mp3 format, 33, 92, 93, 95, 97
- .wav format, 33, 85, 87, 90, 92

- acceleration, 9, 10
- acoustat, 359
- acoustic illusion, 589, 594
- aliasing, 32, 140, 459
- alignment, 530
- AM. *see* amplitude
- amplitude
 - crest, 168
 - envelope, 125, 126, 128, 132, 133, 135, 138, 189, 191, 193, 205, 465, 474, 483, 503, 524, 531, 568, 599, 602
 - instantaneous, 11, 12, 26, 167, 598, 600
 - maximum, 12, 168
 - modulation (AM), 26, 125, 126, 205, 242, 280, 428, 478, 564, 574, 580, 599
 - peak-to-peak, 12, 168
 - root-mean-square (RMS), 12, 161, 168, 169, 173, 298, 352
 - unit, 10, 115
- analytic signal, 127, 128, 418
- annotation, 258, 348, 353, 357, 358, 540, 541, 547
- AR model. *see* autoregressive model
- area under the curve, 537, 549, 551, 552
- attenuation, 17, 19, 177–179
- AUC. *see* area under the curve
- audition, 381
- auditory spectrogram, 386, 387
- autocorrelation, 395, 405, 413
- automatic identification, 534
- autoregressive model, 394

- Bark scale, 229, 390
- Bessel function, 281, 283
- bin template matching, 538
- bioacoustics, 2
- bit, 30, 35

- calibration, 181, 182
- Carson's rule, 281
- celerity, 12
- centroid, 300
- cepstrum, 241, 242, 273, 274, 302, 381, 405, 408
- channel, 84, 89, 90, 92, 142–145
- chirp, 574, 579, 593
- clarinet, 587
- clicks, 150, 153
- clipping, 33
- clustering, 511
- communication, 35
- complex numbers, 319, 332
- contour lines, 338
- convolution, 455, 467, 521
- covariance, 527, 528
- cross-correlation, 521, 522, 526, 527, 538, 541, 542, 547
- cumulative probability mass function, 495, 497, 499
- cut, 146

- dB. *see* deciBel
- db-RDA. *see* redundancy analysis
- DC. *see* direct current
- DCT. *see* discrete cosine function

- decade, 438, 462
- deciBel, 14, 16, 17, 174, 175, 235, 238, 294, 332, 338, 340, 438
- decoration, 258, 348
- delete, 149
- delta coefficients, 385, 392
- dendrogram, 511
- DFT, 252
- digitization, 30, 167
- digitization depth. *see* quantization
- direct current, 217, 249, 278, 395, 564
- discrete cosine function, 384, 388
- discretization, 286–288, 309
- displacement, 9, 10
- distance matrix, 506
- dominant frequency, 533
- downsample, 317
- DTW. *see* dynamic time warping
- duration, 11, 185
- dynamic time warping, 530, 531, 534

- echo, 241, 242, 467
- ecoacoustics, 2, 479
- ellipse, 515
- energy, 13, 169
- entropy, 298–301, 500, 501
- evenness, 298, 299
- external software, 74–76

- fade-in and fade-out, 163, 594, 595
- far-field, 9
- FFT, *see* Fourier
- fftw, 80
- file list, 94
- filter, 600
 - amplitude, 468
 - Butterworth, 445, 447, 448
 - comb, 443, 444
 - definition, 436
 - finite impulse response (FIR), 455, 457, 459, 462, 467, 473, 474
 - frequency response, 394, 398
 - mel-frequency, 382
 - pre-emphasis, 440
 - preemphasis, 382, 385, 390, 395
 - roll-off rate, 438, 445, 462
 - smoothing, 449
- FIR. *see* filter
- flatness, 298, 299, 301
- FM. *see* frequency
- formant, 23, 395, 396, 398, 416, 417, 604

- Fourier
 - fast transform (FFT), 225, 229, 235
 - inverse transform (IFT), 240
 - Jean-Joseph, 213
 - series, 214–217, 219, 222, 564
 - short-time Fourier transform (STFT), 205, 309, 315, 329, 349, 382, 392, 451, 486, 489, 504, 527, 538
 - transform, 224, 311
 - transformation, 213, 214, 216
 - window, 237, 238, 254, 297, 316, 317, 329, 332, 375, 382
- frequency
 - bandwidth, 294
 - beating, 564
 - carrier, 23, 280, 281, 283
 - coherence, 528, 529
 - definition, 21
 - dominant, 23, 272, 274, 280, 400, 403, 405, 407, 469
 - fundamental, 22, 23, 217, 242, 273, 274, 276, 280, 303, 405, 407, 409–411, 461, 590, 600, 604
 - instantaneous, 23, 26, 418–420, 474, 598–600, 602
 - modulation (FM), 26, 199, 252, 281, 420, 428, 476, 478, 574, 575, 582, 593, 594, 597, 599, 602
 - resonant, 23, 294, 295, 394
 - shift, 473, 476
 - sidebands, 278, 280, 281, 283–285
- frequency spectrum. *see* spectrum
- function
 - apply, 54
 - argument, 47, 48
 - definition, 47
 - help, 48
 - new, 48, 49

- gain, 181–183
- glissando, 593
- gold number, 586
- graphic
 - definition, 65
 - ggplot2, 71, 262
 - high level plot functions, 66
 - layout, 69, 70, 362
 - low level functions, 67, 68
 - parametrization, 67
 - save, 71, 72, 364, 365
- graphical user interface, 40
- GUI. *see* graphical user interface

- Hanning window, 238
- harmonic, 23, 242, 276, 278, 280, 453, 461, 472, 568, 571, 573, 579, 585, 590, 601
- heatmap, 509
- Heisenberg
 - box, 312, 314, 315, 329
 - principle, 254, 312, 313, 315
 - uncertainty, 353
- Hierarchical cluster analysis, 511
- Hilbert transform, 127, 128, 189, 205, 418, 420, 474, 598
- histogram, 155

- IEEE, 33
- IFT. *see* Fourier
- impedance, 16
- index
 - acoustic complexity, 489, 490
 - acoustic diversity, 486
 - acoustic entropy, 486
 - acoustic evenness, 488
 - acoustic richness, 484
 - amplitude, 482
 - Bhattacharyya distance, 494
 - bioacoustic, 482
 - cumulative spectral dissimilarity, 497
 - definition, 479
 - Gini coefficient, 488
 - Hellinger distance, 494
 - Itakuro-Saito, 500
 - Kolmogorov-Smirnov, 497
 - Kullback-Leibler, 500
 - log-spectral, 502
 - mutual information, 501
 - normalised difference soundscape, 491
 - number of frequency peaks, 491
 - Pearson correlation coefficient, 503
 - relative frequency similarity, 502
 - RV correlation coefficient, 504
 - spectral dissimilarity, 496
 - spectral entropy, 485
 - temporal dissimilarity, 503
 - temporal entropy, 483
 - wave dissimilarity, 504
- information theory, 7
- inharmonic, 23
- intensity, 13, 18
- interference, 564, 565
- inverse short-time Fourier transform, 351, 451, 453, 470, 473. *see* Fourier
- ISTFT. *see* Fourier

- kurtosis, 171, 172, 301

- lifter, 385, 389
- linear predictive coefficients, 394
- LPC. *see* linear predictive coefficients

- map, 98, 99
- mel(ody) scale, 230, 381, 382
- mel-frequency cepstral coefficients, 381, 385, 389
- MFCC, 385. *see* mel-frequency cepstral coefficients
- mix, 145
- mono, 144, 145
- movie, 368
- moving average, 131, 132, 136
- moving kernel, 133
- moving sum, 132
- multichannels, 89, 91, 144
- music, 589
- musical note, 231, 232, 234, 411, 579

- near field, 9
- noise, 25, 35, 36, 463, 604
 - control, 566
 - pink, 462, 557
 - red, 557
 - white, 462, 472, 557
- non linearity, 303
- normal distribution, 589, 593, 602
- Nyquist frequency, 32, 227, 459
- Nyquist-Shannon theorem, 32

- object
 - attribute, 43, 44
 - character string, 64
 - class, 41, 44
 - concatenation, 47, 59, 60
 - dimension, 44
 - import/export, 61
 - indexing, 54–58, 63
 - mode, 44
 - name, 43
 - operator, 46
 - structure, 45
 - vectorization, 62
- octave, 231, 232, 264, 271, 438, 462
- offset, 159, 160
- OLA. *see* overlap-add method
- oscillogram, 111, 113–117, 119, 121, 123, 138, 188, 335

- overlap, 130
- overlap-add method, 351
- overplotting, 124
- oversampling, 33
- overtone, 22, 273, 422

- PAA. *see* piecewise aggregate approximation
- Parseval's theorem, 226
- paste, 150, 152
- PCA. *see* principal component analysis
- PCM, 30, 33
- PCoA. *see* principal coordinate analysis
- peak detection, 265, 267, 268, 271, 273, 302, 491
- period, 11, 21
- phase, 9, 20, 303, 305, 419, 438, 564, 565
- piecewise aggregate approximation, 291
- playback, 103
- playlist, 107
- PMF. *see* probability mass function
- pole-zero diagram, 398
- power, 13, 169, 171, 177
- power spectral density, 248, 320, 321
- precision, 189, 190, 192, 198
- preemphasis filter, 395
- pressure, 8, 10, 13, 14
- principal component analysis, 512–514
- principal coordinate analysis, 512–514
- probability mass function, 248, 483, 485, 495, 500, 502, 503
- PSD, 256. *see* power spectral density
- pulse, 558, 559
- pure tone, 278, 281, 283, 472, 478, 564, 566, 569, 574, 584, 587, 599, 602

- Q. *see* quality factor
- quality factor, 294
- quantization, 30, 32, 162
- quefrency, 242

- RDA. *see* redundancy analysis
- recall, 536
- receiver operating characteristic curve, 537, 549–551
- record, 108
- rectangular window, 237
- redundancy analysis, 512–515
- repeat loop, 103, 104, 369
- repetition rate, 188, 205, 207, 208, 210
- resampling, 407
- resolution, 210, 252, 254, 263, 297, 300, 301, 313–315, 317
- resonator, 295
- reverberation, 467
- reverse, 155
- RMS. *see* amplitude
- ROC curve. *see* receiver operating characteristic curve
- root-mean-square. *see* amplitude
- roughness, 298
- rugosity, 298

- sampling, 32
 - change speed, 141
 - definition, 30
 - downsampling, 139, 140
 - oversampling, 140, 141
- sampling frequency, 30
 - change speed, 107
- sawtooth, 24, 561, 562
- SAX. *see* symbolic aggregate approximation
- scaling, 256
- script
 - condition, 50, 51
 - for loop, 51, 52, 123, 191, 299, 474, 493, 590, 593, 601
 - repeat, 52
 - sourcing, 74
 - structure, 48
 - things to do, 73
 - while, 52
- sensitivity, 182, 536
- Shannon, 34–36
- Shannon evenness, 485
- Shepard scale, 589, 593
- shiny, 609
- short-time Fourier transform. *see* Fourier
- sideband, 564, 580, 582
- signal-to-noise ratio, 173, 186, 200
- SIL. *see* sound intensity level
- silence, 155, 157, 555, 558, 565, 590, 595, 597, 602
- sine wave, 25. *see* pure tone
- skewness, 171, 172, 300
- sliding window, 130, 131, 314
- smoothing, 131–133, 135, 193, 524
- smoothing kernel. *see* moving kernel
- SNR. *see* signal-to-noise ratio
- Song Meter, 99, 480, 481
- sound intensity level, 16
- sound pressure level, 14, 174, 177
- sound velocity level, 16
- specificity, 536

- spectral leakage, 237, 254
- spectrogram, 335, 540, 577, 586, 592, 594
 - 3D, 372, 374
 - animation, 373
 - annotation, 353, 357
 - decoration, 348
 - definition, 309, 315
 - display, 319, 320, 322, 324, 326, 334
 - dynamic, 366
 - long, 356
 - measurements, 353
 - movie, 368
 - parametrization, 358
 - plate, 362
 - waterfall, 370
- spectrum
 - frequency, 221, 224, 226, 247–249, 251, 252, 257, 258, 260, 263, 265, 272, 274, 275, 278, 281, 285, 288, 293, 300, 337, 463, 485, 491, 492, 495–497, 501, 506, 526, 533, 557
 - mean frequency, 375, 377
 - phase, 221, 224
 - soundscape, 377
- speech, 381, 382, 394, 604
- SPL. *see* sound pressure level
- splines, 449
- square, 24, 560
- stereo, 84, 85, 88, 92, 125, 143–145, 567
- STFT. *see* Fourier
- stridulation, 295, 594
- SVL. *see* sound velocity level
- sweep, 574
- symbolic aggregate approximation, 291
- symbolic series, 286, 288, 291, 501
- synthesis
 - additive synthesis, 564, 604
 - articulatory synthesis, 604
 - formant synthesis, 604
 - modulation synthesis, 574
 - speech synthesis, 604
 - tonal synthesis, 598
- Teager-Kaiser energy operator, 427, 428
- template, 521, 535, 536, 538, 539, 541, 542, 545, 546
- time series, 29, 30, 83, 84
- TKEO. *see* Teager-Kaiser energy operator
- transfer function, 394, 395, 436, 438, 440, 443, 445, 447, 451, 459, 461–463
- triangle, 24, 561, 563
- trim, 155
- tuning fork, 8
- two-voice, 278
- uncertainty principle. *see* Heisenberg principle
- velocity, 9, 12
- vocal folds, 394
- vocal tract, 394
- voice, 413
- vowel, 604–606
- Ward distance, 511
- wavelength, 9, 11, 12
- wavelet transform, 214
- Western music, 231, 410, 579, 584, 587
- working directory, 40
- ZC. *see* zero-crossing
- ZCR. *see* zero-crossing rate
- zero-crossing, 421–423
- zero-crossing rate, 423
- zero-padding, 315, 316, 332
- zero-pading, 317
- zoom
 - in time, 121, 123, 202, 203