wavpy

Release 0.2.7

Samuele Carcagno

May 08, 2023

CONTENTS:

CHAPTER

ONE

INTRODUCTION

Author
Samuele Carcagno

Module for reading and writing WAV files using MATLAB-style wavread and wavwrite functions. It is a simple but convenient wrapper to the scipy.io.wavfile module.

CHAPTER

TWO

WAVPY – MODULE TO READ AND WRITE WAVS

Module for reading and writing WAV files. It is a simple but convenient wrapper to the wave module and the scipy.io.wavfile module.

wavpy.**sound**(*snd*, *fs=48000*, *nbits=32*)

Play out a numpy array through the soundcard.

2.1 Parameters

snd
: [array of floats] The sound to be played.

fs
: [int] Sampling frequency of the sound.

nbits
: [int] Desired bit depth.

2.2 Examples

```
>>> sound(snd, fs=48000, nbits=32)
```

wavpy.**wavread**(*fName*, *scale=True*)

Read a WAV file into a numpy array.

2.3 Parameters

fName
: [string] Name of the WAV file to read

scale
: [boolean] Option valid only for the PCM wave format. If *True* the data will be returned as floaring point values ranging between -1 and 1. If *False* the data will be returned as the closest numpy integer type to the WAV bit depth, with values randing within the bit depth range.

2.4 Returns

snd : numpy array with the sound.

fs : sampling frequency.

nbits : bit depth.

2.5 Examples

```
>>> snd, fs, nbits = wavread("file.wav")
```

wavpy.**wavwrite**(*data*, *fs*, *nbits*, *fName*, *wave_format='PCM'*, *scale=True*)

Write a numpy array as a WAV file.

2.6 Parameters

data
: [array of floats] The data to be written to the WAV file.

fs
: [int] Sampling frequency of the sound.

nbits
: [int] Bit depth of the WAV file (currently only values of 16 and 32 are supported)

fName
: [string] Name of the WAV file.

scale
: [boolean] Option valid only for the PCM wave format. If the data are floating point values ranging from -1 to 1 and scale is set to *True* they will be converted to the range of the appropriate integer type (according to the chosen bit depth). If scale is set to *False* it is assumed that the values are already in the range of the appropriate integer type (e.g. between -2**15 and 2**15-1 for 16 bits). Note that if *wave_format* is set to *IEEE_FLOAT* the data are never scaled.

2.7 Examples

```
>>> wavwrite(data, 48000, 32, "file.wav")
```

CHAPTER

THREE

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX