

# Package: wav (via r-universe)

October 30, 2024

**Title** Read and Write WAV Files

**Version** 0.1.0.9000

**Description** Efficiently read and write Waveform (WAV) audio files  
<<https://en.wikipedia.org/wiki/WAV>>. Support for unsigned 8 bit  
Pulse-code modulation (PCM), signed 12, 16, 24 and 32 bit PCM  
and other encodings.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.2.1

**LinkingTo** Rcpp

**Imports** Rcpp

**Suggests** covr, testthat (>= 3.0.0), patrick

**Config/testthat/edition** 3

**SystemRequirements** C++11

**URL** <https://github.com/mlverse/wav>, <https://mlverse.github.io/wav/>

**BugReports** <https://github.com/mlverse/wav/issues>

**Repository** <https://mlverse.r-universe.dev>

**RemoteUrl** <https://github.com/mlverse/wav>

**RemoteRef** HEAD

**RemoteSha** 5826370dafa626aeb3399297543d33ac358c9d5f

## Contents

read_wav	2
<b>Index</b>	<b>4</b>

read\_wav

*Read/write wav files***Description**

Efficiently read and write **WAV files**.

**Usage**

```
read_wav(path)
```

```
write_wav(x, path, sample_rate = 44100, bit_depth = 32, ..., normalize = TRUE)
```

**Arguments**

path	Path to file that will be read or written to.
x	Numeric matrix with dimensions [n_channels, n_samples]. Values in the matrix should be <double> in the range [-1, 1] or integers in the range [-.Machine\$integer.max, .Machine\$integer.max], ie. 32 bits signed integers like R integers containing the amplitudes. Depending on the value of normalize and the bit_depth you can use different ranges.
sample_rate	Sample rate in Hz of the associated samples.
bit_depth	Bit depth of associated samples. This only affects the precision data is saved to the file.
...	Currently unused.
normalize	Boolean indicating whether integers should be normalized before writing. Only used when <code>write_wav()</code> is called with a integer matrix. For example when you write a sample with a amplitude value of 2147483647 and bit_depth = 8, you would need to normalize this integer so it actually refers to the maximum unsigned int available (i.e. 255). You can avoid normalizing when the amplitudes are already in the correct integer range for the bit_depth you are saving, in this case provide <code>normalize = FALSE</code> .

**Value**

- When reading: A numeric matrix with samples. It also contains the attributes `sample_rate` and `bit_depth`.
- When writing: A boolean which is TRUE if writing was successful and FALSE otherwise.

**Functions**

- `write_wav()`: Write a wav file.

**Examples**

```
x <- matrix(sin(440 * seq(0, 2*pi, length = 44100)), nrow=1)
tmp <- tempfile(fileext = ".wav")
write_wav(x, tmp)
y <- read_wav(tmp)
all.equal(as.numeric(x), as.numeric(y), tolerance = 1e-7)
```

# Index

`read_wav`, [2](#)

`write_wav (read_wav)`, [2](#)

`write_wav()`, [2](#)