

voice

0.0.0.9047

Filipe J. Zabala

Graduate Program in Psychiatry  
Medical School · UFRGS

2021-09-23

- ① What is voice
- ② How to use voice
- ③ Audio variables
- ④ Current research

# 1. What is voice

- General tools for voice analysis

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in **R**

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in R
- Based on

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in R
- Based on
  - `tuneR`

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in R
- Based on
  - [tuneR](#)
  - [tidyverse](#) collection

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in **R**
- Based on
  - **tuneR**
  - **tidyverse** collection
  - **wrassp** - an **R** wrapper around **libassp**



# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in R
- Based on
  - [tuneR](#)
  - [tidyverse](#) collection
  - [wrassp](#) - an R wrapper around [libassp](#)
  - [Parselmouth](#) - a Python wrapper around [Praat](#)

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in **R**
- Based on
  - **tuneR**
  - **tidyverse** collection
  - **wrassp** - an **R** wrapper around **libassp**
  - **Parselmouth** - a **Python** wrapper around **Praat**
  - **pyannote-audio** - an open-source toolkit written in **Python** for speaker diarization based on **PyTorch** machine learning framework

# 1. What is voice

- General tools for voice analysis
- It is being developed to be an easy-to-use set of tools for audio analysis in **R**
- Based on
  - **tuneR**
  - **tidyverse** collection
  - **wrassp** - an **R** wrapper around **libassp**
  - **Parselmouth** - a **Python** wrapper around **Praat**
  - **pyannotate-audio** - an open-source toolkit written in **Python** for speaker diarization based on **PyTorch** machine learning framework
- Also uses **gm**, **music**, **Homebrew**, **wget**, **ffmpeg**, **MuseScore** and **Miniconda**

- You may use some user-friendly functions

---

<sup>1</sup> f0, formants, zcr, mhs, rms, gain, rfc, ac, mfcc, cep, dft, css, lps

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

- You may use some user-friendly functions
  - poetry to get the best words in their best order

---

<sup>1</sup> f0, formants, zcr, mhs, rms, gain, rfc, ac, mfcc, cep, dft, css, lps

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

## 2. How to use voice

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.



- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables
  - `notes` to assign frequencies to notes in different formats

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables
  - `notes` to assign frequencies to notes in different formats
  - `duration` to get the order and duration of notes

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

## 2. How to use voice

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables
  - `notes` to assign frequencies to notes in different formats
  - `duration` to get the order and duration of notes
- You may associate with other packages

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

## 2. How to use voice

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables
  - `notes` to assign frequencies to notes in different formats
  - `duration` to get the order and duration of notes
- You may associate with other packages
  - `music::noteDistance` to assess the tonal variability in speech

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

## 2. How to use voice

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables
  - `notes` to assign frequencies to notes in different formats
  - `duration` to get the order and duration of notes
- You may associate with other packages
  - `music::noteDistance` to assess the tonal variability in speech
  - `gm::Music` and `gm::show` to present the audio using playable music sheets

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

## 2. How to use voice

- You may use some user-friendly functions
  - `poetry` to get the best words in their best order
  - `splitw` to split the spoken parts into small blocks
  - `extract_features` to easily get a data frame with the main<sup>1</sup> and some extra<sup>2</sup> features
  - `smooth_df` to smooth numeric variables
  - `notes` to assign frequencies to notes in different formats
  - `duration` to get the order and duration of notes
- You may associate with other packages
  - `music::noteDistance` to assess the tonal variability in speech
  - `gm::Music` and `gm::show` to present the audio using playable music sheets
- A vignette is found at <http://filipezabala.com/voicegnette>

---

<sup>1</sup> `f0`, `formants`, `zcr`, `mhs`, `rms`, `gain`, `rfc`, `ac`, `mfcc`, `cep`, `dft`, `css`, `lps`

<sup>2</sup> 'formant dispersion' by Fitch (1997), 'formant position' by Puts, Apicella & Cárdena (2011), 'formant removal', 'formant nested removal' and 'formant position removal' purposed for the first time here.

- Proportion of speech on audio

## 3. Audio variables

- Proportion of speech on audio
- Summary statistics

---

<sup>3</sup>Gray colors indicate *work in progress*



- Proportion of speech on audio
- Summary statistics
  - Notes distance

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal
- Probabilities<sup>3</sup>

---

<sup>3</sup>Gray colors indicate *work in progress*

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal
- Probabilities<sup>3</sup>
  - Female/Male

---

<sup>3</sup>Gray colors indicate *work in progress*

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal
- Probabilities<sup>3</sup>
  - Female/Male
  - Age (mode of posterior distribution)

---

<sup>3</sup>Gray colors indicate *work in progress*

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal
- Probabilities<sup>3</sup>
  - Female/Male
  - Age (mode of posterior distribution)
  - Depression

---

<sup>3</sup>Gray colors indicate *work in progress*



- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal
- Probabilities<sup>3</sup>
  - Female/Male
  - Age (mode of posterior distribution)
  - Depression
  - Anxiety

---

<sup>3</sup>Gray colors indicate *work in progress*

- Proportion of speech on audio
- Summary statistics
  - Notes distance
  - Gain ('volume')
  - Fundamental frequency (F0)
  - Formant dispersion, position, removal, nested removal and position removal
- Probabilities<sup>3</sup>
  - Female/Male
  - Age (mode of posterior distribution)
  - Depression
  - Anxiety
  - ...

---

<sup>3</sup>Gray colors indicate *work in progress*

- Automatically identify speakers

- Automatically identify speakers
  - Proportion of speech on audio by speaker

- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker

- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker
  - Probabilities by speaker

- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker
  - Probabilities by speaker
- `plot_note` function to intuitively plot frequencies/notes

- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker
  - Probabilities by speaker
- `plot_note` function to intuitively plot frequencies/notes
- Chords progression (e.g. major/minor)



- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker
  - Probabilities by speaker
- `plot_note` function to intuitively plot frequencies/notes
- Chords progression (e.g. major/minor)
- Build a dashboard to allow easy interaction

- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker
  - Probabilities by speaker
- `plot_note` function to intuitively plot frequencies/notes
- Chords progression (e.g. major/minor)
- Build a dashboard to allow easy interaction
  - Allow user to send, link and record audios (`tabr` package)

- Automatically identify speakers
  - Proportion of speech on audio by speaker
  - Summary statistics by speaker
  - Probabilities by speaker
- `plot_note` function to intuitively plot frequencies/notes
- Chords progression (e.g. major/minor)
- Build a dashboard to allow easy interaction
  - Allow user to send, link and record audios (`tabr` package)
- `tag` function to enrich databases with audio variables