**My-Voice Analysis**

My-Voice Analysis is a Python library for the analysis of voice (simultaneous speech, high entropy) without the need of a transcription. It breaks utterances and detects syllable boundaries, fundamental frequency contours, and formants. Its built-in functions recognise and measures

1. gender recognition,
2. speech mood (semantic analysis),
3. pronunciation posterior score
4. articulation-rate,
5. speech rate,
6. filler words,
7. f0 statistics,

The library was developed based upon the idea introduced by Nivja DeJong and Ton Wempe [1], Paul Boersma and David Weenink [2], Carlo Gussenhoven [3], S.M Witt and S.J. Young [4] and Yannick Jadoul [5]. Peaks in intensity (dB) that are preceded and followed by dips in intensity are considered as potential syllable cores.

My-Voice Analysis is unique in its aim to provide a complete quantitative and analytical way to study **acoustic features of a speech**. Moreover, those features could be analysed further by employing Python’s functionality to provide more fascinating insights into speech patterns.

This library is for Linguists, scientists, developers, speech and language therapy clinics and researchers.

Please note that My-Voice Analysis is currently in initial state though in active development. While the amount of functionality that is currently present is not huge, more will be added over the next few months.

**Installation**

my-voice-analysis can be installed like any other Python library, using (a recent version of) the Python package manager pip, on Linux, macOS, and Windows:

pip install my-voice-analysis

or, to update your installed version to the latest release:

pip install -u my-voice-analysis

NOTE: After installing My-Voice-Analysis, copy the file myspsolution.praat from <https://github.com/Shahabks/my-voice-analysis> and save in the directory where you will save audio files for analysis.

Audio files must be in \*.wav format, recorded at 44 kHz sample frame and 16 bits of resolution.

**Example usage**

Gender recognition and mood of speech: Function *myspgend(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspgend(p,c)

[] a female, mood of speech: Reading, p-value/sample size= :0.00 5

Pronunciation posteriori probability score percentage: Function *mysppron(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.mysppron(p,c)

[]Pronunciation\_posteriori\_probability\_score\_percentage= :85.00

Detect and count number of syllables: Function *myspsyl(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspsyl(p,c)

[]number\_ of\_syllables= 154

Detect and count number of fillers and pauses: Function *mysppaus(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.mysppaus(p,c)

[]number\_of\_pauses= 22

Measure the rate of speech (speed): Function *myspsr(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspsr(p,c)

[]rate\_of\_speech= 3 # syllables/sec original duration

Measure the articulation (speed): Function *myspatc(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspatc(p,c)

[]articulation\_rate= 5 # syllables/sec speaking duration

Measure speaking time (excl. fillers and pause): Function *myspst(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspst(p,c)

[]speaking\_duration= 31.6 # sec only speaking duration without pauses

Measure total speaking duration (inc. fillers and pauses): Function *myspod(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspod(p,c)

[]original\_duration= 49.2 # sec total speaking duration with pauses

Measure ratio between speaking duration and total speaking duration: Function *myspbala(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspbala(p,c)

[]balance= 0.6 # ratio (speaking duration)/(original duration)

Measure fundamental frequency distribution mean: Function *myspf0mean(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0mean(p,c)

[]f0\_mean= 212.45 # Hz global mean of fundamental frequency distribution

Measure fundamental frequency distribution SD: Function *myspf0sd(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0sd(p,c)

[]f0\_SD= 57.85 # Hz global standard deviation of fundamental frequency distribution

Measure fundamental frequency distribution median: Function *myspf0med(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0med(p,c)

[]f0\_MD= 205.7 # Hz global median of fundamental frequency distribution

Measure fundamental frequency distribution minimum: Function *myspf0min(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0min(p,c)

[]f0\_min= 77 # Hz global minimum of fundamental frequency distribution

Measure fundamental frequency distribution maximum: Function *myspf0max(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0max(p,c)

[]f0\_max= 414 # Hz global maximum of fundamental frequency distribution

Measure 25th quantile fundamental frequency distribution: Function *myspf0q25(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0q25(p,c)

[]f0\_quan25= 171 # Hz global 25th quantile of fundamental frequency distribution

Measure 75th quantile fundamental frequency distribution: Function *myspf0q75(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.myspf0q75(p,c)

[]f0\_quan75= 244 # Hz global 75th quantile of fundamental frequency distribution

Overview: Function *mysptotal(p,c)*

import myspsolution as mysp

p="Walkers" # Audio File title

c=r"C:\Users\Shahab\Desktop\Mysp" # Path to the Audio\_File directory (Python 3.7)

mysp.mysptotal(p,c)

number\_ of\_syllables 154

number\_of\_pauses 22

rate\_of\_speech 3

articulation\_rate 5

speaking\_duration 31.6

original\_duration 49.2

balance 0.6

f0\_mean 212.45

f0\_std 57.85

f0\_median 205.7

f0\_min 77

f0\_max 414

f0\_quantile25 171

f0\_quan75 244

**Development**

My-Voice-Analysis was developed by MYOLUTIONS Lab in Japan. It is part of New Generation of Voice Recognition and Analysis Project in MYSOLUTIONS Lab. That is planned to rich the functionality of My-Voice Analysis by adding more advanced functions.

**References and Acknowledgements**

1. DeJong N.H, and Ton Wempe [2009]; “Praat script to detect syllable nuclei and measure speech rate automatically”; Behavior Research Methods, 41(2).385-390.
2. Paul Boersma and David Weenink;  <http://www.fon.hum.uva.nl/praat/>
3. Gussenhoven C. [2002]; “ Intonation and Interpretation: Phonetics and Phonology”; Centre for Language Studies, Univerity of Nijmegen, The Netherlands.
4. Witt S.M and Young S.J [2000]; “Phone-level pronunciation scoring and assessment or interactive language learning”; Speech Communication, 30 (2000) 95-108.
5. Jadoul Y. <https://parselmouth.readthedocs.io/en/latest/installation.html>

**MIT License**

* Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:
* The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
* THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.