Free and open-source software for soundscape visualization and binaural analysis

soundscapy

Soundscapy: A python package for soundscape assessment and analysis

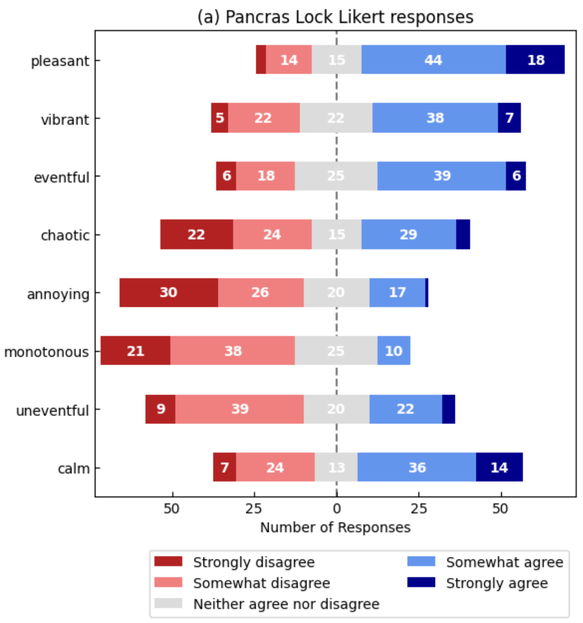
MITCHELL, Andrew *Institute for Environmental Design & Engineering, University College London, UK*

INTRO

- Soundscapy is a Python package designed to simplify the analysis of soundscape questionnaire and recording data.
- Implements the circumplex plotting methods proposed in ISO 12913-3 and Mitchell, Aletta, & Kang (2022).
- Automatically performs validation checks on datasets and calculates the ISO 12913 coordinate projections to get ISOPleasant and ISOEventful values.

Raw Response Plotting – Stacked Bar and Radar

Soundscapy includes standard plots of the questionnaire responses, using either a stacked bar (Likert-style) or a radar plot.



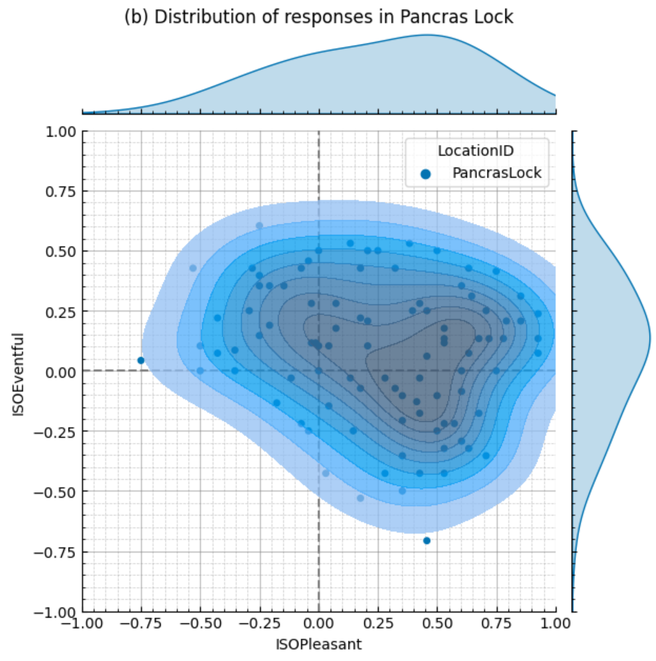
However, these attributes can be difficult to analyse individually. Instead we might prefer to work with two-dimensional continuous data.

The Circumplex Coordinates Approach

- By using the coordinate projection from ISO12913, we can instead **treat each response as a coordinate point**, and groups of responses as a distribution.
- **In contrast to ISO12913** the approach proposed in Mitchell, Aletta, & Kang (2022) transforms **each response separately.**

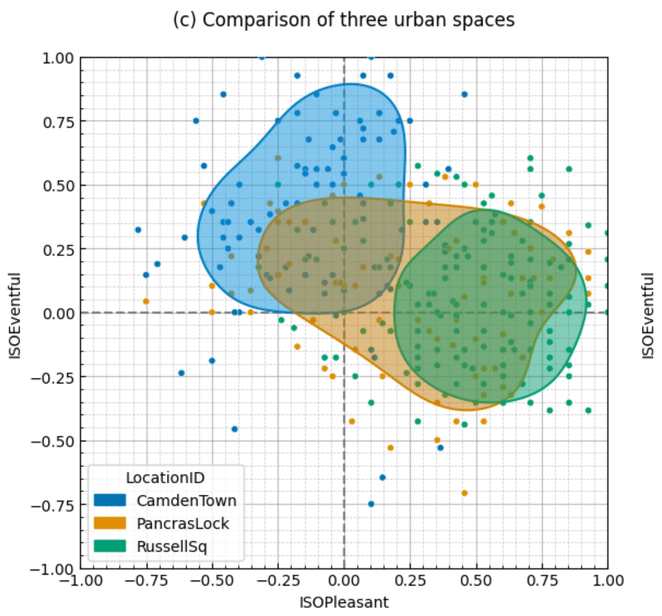
PROBABLILISTIC CIRCUMPLEX PLOTTING

- Soundscapy makes it easy to calculate, visualize, and further analyse these distributions.
- Figure (b) demonstrates the **distribution of soundscape assessments** in an urban area, with decile contour curves and marginal distributions.



Simplified Comparisons

- Figure (c) shows how simplifying the distribution to the 50th percentile, it is easy to **make comparisons between locations, groups, or conditions.**



STRUCTURE & COMMANDS

Below is an example of all the code required to load the ISD, validate the data, and produce Figure (c) shown to the left.

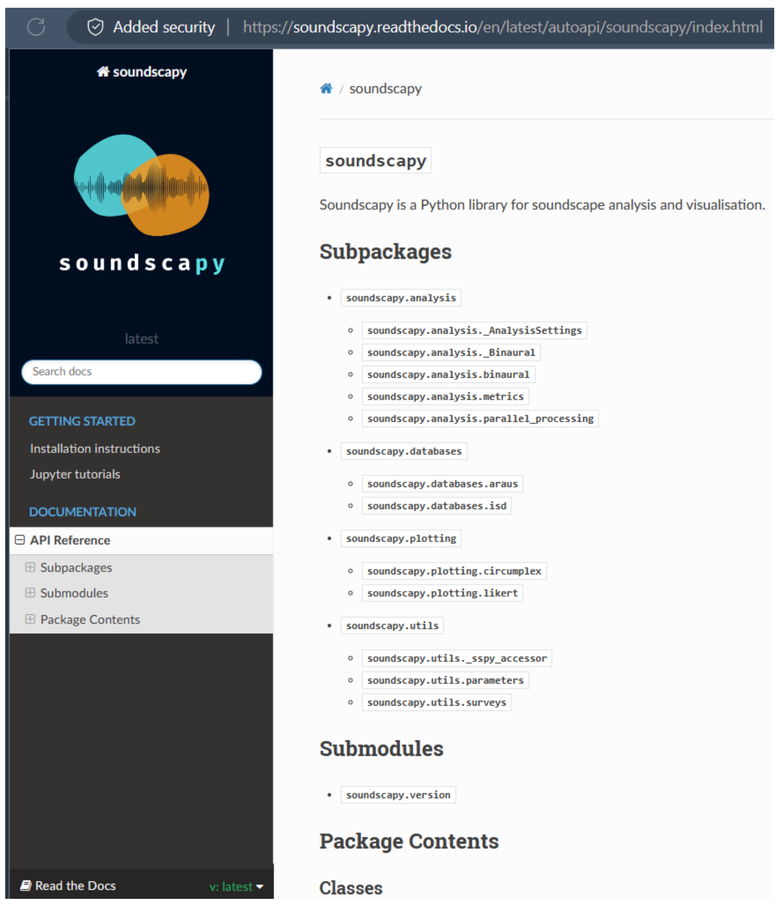
```
import soundscapy as sspy

data = sspy.isd.load_isd_dataset()
data, excl_data = data.sspy.validate_dataset(allow_na=False)

locations = ["CamdenTown", "PancrasLock", "RussellSq"]

data.query("LocationID in @locations").sspy.density(
    title = "Comparison of three locations",
    hue = "LocationID",
    density_type = "simple
)
```

Documentation and examples are available:



THE FUTURE

We aim to release a stable v1.0 in the next few months, with improved documentation and testing. Our plans for future enhancements are:

- Structural Summary Method (SSM)
- Integration of other datasets – ISD, ARAUS, SATP
- Soundscapy_viz GUI web app
- (Longterm) Incorporation of a predictive soundscape model to go from binaural recording to predicted soundscape plot.

PSYCHOACOUSTIC AND ECO-ACOUSTIC ANALYSIS

- In addition to making it simpler to work with soundscape survey data, Soundscapy makes it easy to perform **bulk analysis of binaural recordings**.
- By providing a streamlined and repeatable frontend to analysis libraries such as MoSQito, scikit-maad, and python-acoustics, we make it possible to calculate a wide range of metrics from **psychoacoustics, eco-acoustics, and environmental acoustics with consistent settings**.
- Calculation settings can be saved and shared in a JSON file.
- **Parallel calculations** drastically speed up bulk-processing.

```
from soundscapy import Binaural
b = Binaural.from_wav("example.wav")
b.mosqito_metric('loudness_zwtv', statistics=(5,50,'avg', 'max'), as_df=True, parallel=True
```

```
- Calculating MoSQITo metric: loudness_zwtv
```

[11]:

Recording	Channel	N_5	N_50	N_avg	N_max
CT101	Left	28.834482	23.164299	22.669519	36.160815
	Right	30.834215	23.939352	23.774119	37.762344

Scan to learn about and install Soundscapy

Scan to download the full paper

European Research Council
Established by the European Commission
UCL