Soundscapy

A Python Package for Soundscape Assessment and Analysis

Andrew Mitchell

2024-08-26

University College London

Introduction

- Soundscape: A holistic approach to understanding and assessing acoustic environments

Soundscapy addresses these challenges through:

Figure 1: ISO 12913-3 recommendations for soundscape analysis

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)

Soundscapy addresses these challenges through:

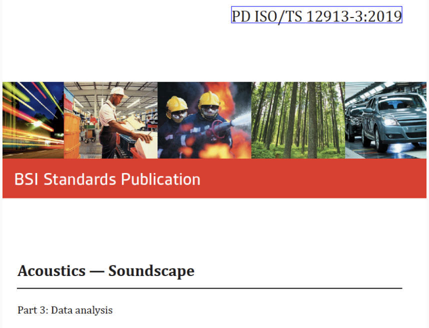


Figure 1: ISO 12913-3 recommendations for soundscape analysis

Introduction

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)
- Despite standardization, challenges persist in consistent application and interpretation

Soundscapy addresses these challenges through:

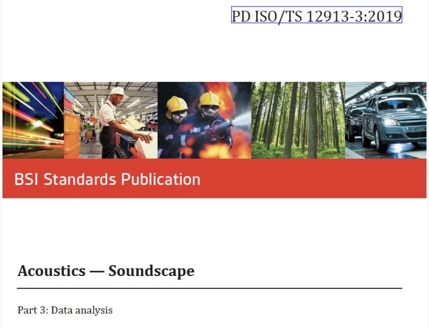


Figure 1: ISO 12913-3 recommendations for soundscape analysis

Introduction

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)
- Despite standardization, challenges persist in consistent application and interpretation

Soundscapy addresses these challenges through:

- An open-source Python package accessible to researchers and practitioners

Figure 1: ISO 12913-3 recommendations for soundscape analysis

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)
- Despite standardization, challenges persist in consistent application and interpretation

Soundscapy addresses these challenges through:

- An open-source Python package accessible to researchers and practitioners
- Alignment with ISO 12913-3 guidelines for consistency (including upcoming revisions)

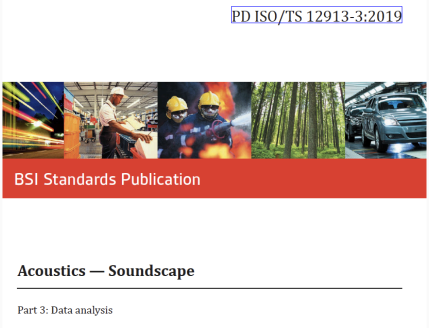


Figure 1: ISO 12913-3 recommendations for soundscape analysis

Introduction

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)
- Despite standardization, challenges persist in consistent application and interpretation

Soundscapy addresses these challenges through:

- An open-source Python package accessible to researchers and practitioners
- Alignment with ISO 12913-3 guidelines for consistency (including upcoming revisions)
- Facilitating reproducible and standardized analysis workflows

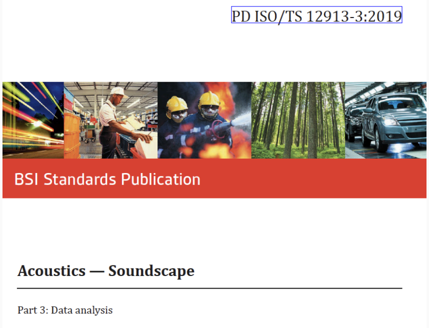


Figure 1: ISO 12913-3 recommendations for soundscape analysis

Introduction

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)
- Despite standardization, challenges persist in consistent application and interpretation

Soundscapy addresses these challenges through:

- An open-source Python package accessible to researchers and practitioners
- Alignment with ISO 12913-3 guidelines for consistency (including upcoming revisions)
- Facilitating reproducible and standardized analysis workflows
- Bridging the gap between theoretical soundscape concepts and practical applications

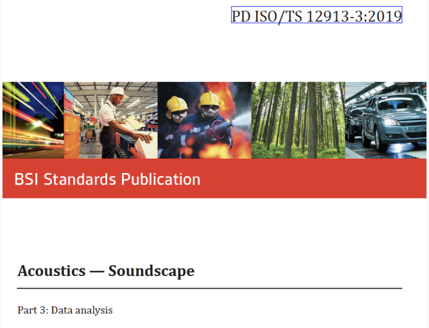


Figure 1: ISO 12913-3 recommendations for soundscape analysis

Introduction

- Soundscape: A holistic approach to understanding and assessing acoustic environments
- ISO 12913 series provides a standardized framework for soundscape data collection and analysis (introduced in 2018)
- Despite standardization, challenges persist in consistent application and interpretation

Soundscapy addresses these challenges through:

- An open-source Python package accessible to researchers and practitioners
- Alignment with ISO 12913-3 guidelines for consistency (including upcoming revisions)
- Facilitating reproducible and standardized analysis workflows
- Bridging the gap between theoretical soundscape concepts and practical applications
- Enabling large-scale soundscape studies with efficient data processing

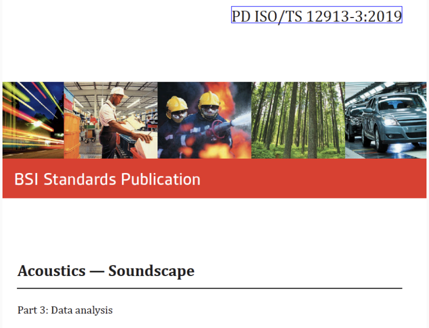


Figure 1: ISO 12913-3 recommendations for soundscape analysis

Soundscapy: Origins and Purpose

- Developed as part of the Soundscape Indices (SSID) project, an ERC Horizon 2020 Grant
- Built upon the SSID Protocol, a standardized method for *in situ* soundscape data collection
- Builds upon the extensive International Soundscape Database (ISD) and other ISO 12913-2 datasets

Key features and innovations:

- Advanced visualization using distributional methods for nuanced soundscape representation
- Comprehensive psychoacoustic and acoustic analysis capabilities for binaural recordings
- Optimized for efficient processing of large-scale soundscape datasets
- Ensures consistency with ISO 12913-3 guidelines throughout the analysis pipeline

Simple installation and intuitive usage

```
pip install soundscapy
```

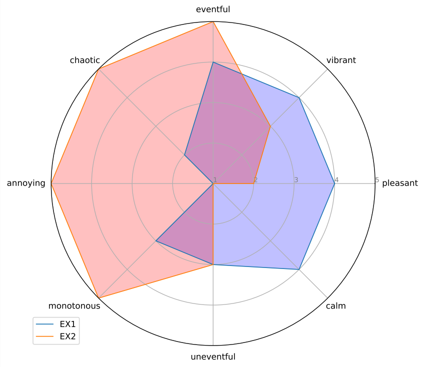
```python
import soundscapy as sspy
```

Soundscape Circumplex Model

- Proposed by Axelsson, Nilsson, and Berglund (2010)
- Comprises two orthogonal dimensions that capture key aspects of soundscape perception:
  - Pleasant-Annoying (x-axis)
  - Eventful-Uneventful (y-axis)
- Eight perceptual scales
- Officially adopted in ISO/TS 12913-2, quickly becoming the dominant soundscape perception assessment method[a]
- Serves as the cornerstone for Soundscapy's analytical approach and visualisations

[a] Aletta and Torresin (2023)

```python
from soundscapy.plotting import likert
likert.paq_radar_plot(sample_transform)
```



ISO Coordinates Calculation

- Transforms 8 PAQ responses into 2D coordinates
- Implements improved equations from ISO 12913-3 upcoming revisions
- Offers flexible input ranges (e.g., 1-5, 0-100) to accommodate various survey designs
- Supports different angle configurations for various survey translations[a]

$$P_{ISO} = \frac{1}{\lambda_{Pl}} \sum_{i=1}^{8} \cos\theta_i \cdot \sigma_i \tag{1}$$

$$E_{ISO} = \frac{1}{\lambda_{Pl}} \sum_{i=1}^{8} \sin\theta_i \cdot \sigma_i \tag{2}$$

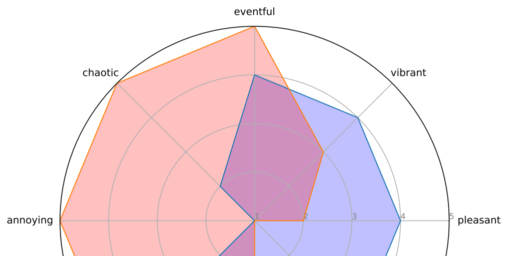
[a] Aletta et al. (2024)

```
# Load data from the ISD
data = sspy.isd.load()

# Apply built-in data quality checks
data, excl_data = sspy.isd.validate(data, allow_paq_na=False)

# Calculate the ISO Coordinates
data = sspy.surveys.add_iso_coords(data)
```

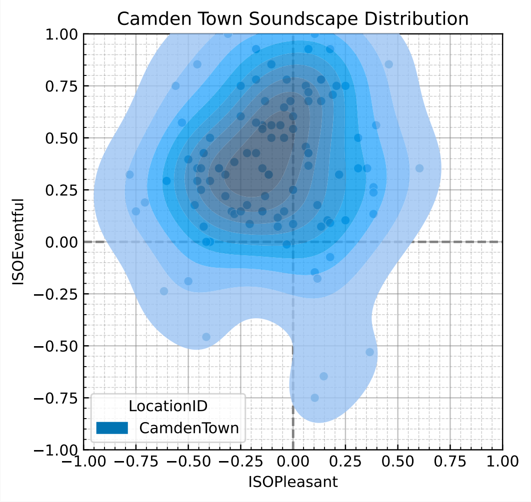
	PAQ1	PAQ2	PAQ3	PAQ4	PAQ5	PAQ6	P
0	2.0	4.0	2.0	1.0	2.0	2.0	4
1	2.0	4.0	4.0	4.0	4.0	4.0	1
2	5.0	3.0	3.0	1.0	2.0	1.0	3
3	5.0	3.0	3.0	1.0	2.0	2.0	3
4	5.0	3.0	3.0	2.0	2.0	3.0	3



Distribution-based Analysis

- Developed by Mitchell, Aletta, and Kang (2022)
- Represents the collective perception of a soundscape as a distribution, using kernel density estimation
- Key Benefits:
  - Visualizes central tendency, dispersion, and skewness of perceptions
  - Illustrates the shape and extent of soundscape perception in circumplex space
  - Allows for statistical comparisons between different soundscapes or groups
- Additional features:
  - Calculation of percentiles and other statistical measures
  - Multiple visualization options (both Seaborn and Plotly backends available)

```python
sspy.plotting.density_plot(
  sspy.isd.select_location_ids(data, "CamdenTown"),
  title="Camden Town Soundscape Distribution",
  hue="LocationID",
  incl_scatter=True
)
```




```python
sspy.plotting.density_plot(
  sspy.isd.select_location_ids(data, ("CamdenTown", "PancrasLock")),
```

Databases Integration

- International Soundscape Database (ISD)[a]
  - 2,706 high-quality, real-world binaural recordings
  - 3,589 *in situ* survey responses
  - Data from 6 cities across Europe and China
- Soundscape Attributes Translation Project (SATP)[b]
  - 19,089 responses
  - 708 participants
  - 19 languages
- ARAUS (Affective Responsees to Augmented Urban Soundscapes)[c]
  - Currently under development
  - Large-scale lab-based dataset of augmented soundscapes

[a] Mitchell et al. (2021)
[b] Oberman et al. (2024)
[c] Ooi et al. (2023)

```python
# Load ISD data
isd_data = sspy.isd.load()

# Load SATP data
import soundscapy.databases.satp as satp
satp_data = satp.load_zenodo()

print(f"ISD shape: {isd_data.shape}")
print(f"SATP shape: {satp_data.shape}")

sspy.isd.soundscapy_describe(isd_data).head(5)
```

```
ISD shape: (3589, 142)
SATP shape: (17441, 16)
```

	count	ISOPleasant	ISOEventful	pl
CarloV	126	0.518	-0.012	0.
SanMarco	99	0.221	0.373	0.
PlazaBibRambla	24	0.463	-0.023	0.
CamdenTown	105	-0.103	0.364	0.
EustonTap	100	-0.211	0.190	0.

Integration with multiple libraries:

- Python Acoustics: For standard acoustic metrics
- scikit-maad: Advanced ecological soundscape analysis
- MoSQITo: Psychoacoustics metrics

Wide range of metrics:

- SPL (L_{Zeq}, L_{Aeq}, L_{Ceq})
- Loudness (N)
- Sharpness (S)
- Roughness (R)
- Bioacoustic Indices
- Many more...

Key advantages:

- Completely free, open license, and open source
- Adherence to ISO and other relevant standards
- Optimized for binaural recordings
- Consistent calculation methods across studies
- Extensible framework for adding new metrics

```python
from soundscapy.audio.analysis_settings import MetricSettings
from soundscapy.audio import Binaural

b = Binaural.from_wav("data/CT101.wav")

laeq_settings = MetricSettings(
    run = True,
    statistics = (5, 50, 'avg', 'max'),
    label="LAeq",
)

b.pyacoustics_metric('LAeq', metric_settings=laeq_settings).round(2)
```

Recording	Ch
CT101	Le Ri

```python
b.mosqito_metric('sharpness_din_perseg').round(2)
```

Recording	Ch
CT101	[8]Le

- **Unified Output Format**: Ensures consistency across datasets, simplifying analysis.
- **Parallel Processing**: Leverages multi-core CPUs for efficient computation.
- **Reproducbible Analysis**: Shareable plain-text analysis configuration and documentation.

Performance Example:

- 20 recordings (10 min 41 sec total)
- Serial processing: 29.4 minutes
- Parallel processing: 7.7 minutes
- Speed-up factor: 3.8x

```python
from soundscapy.audio import AudioAnalysis

wav_folder = Path("data")

# Initialize AudioAnalysis with default settings
analysis = AudioAnalysis()

# Analyse a folder of recordings
folder_results = analysis.analyze_folder(
    wav_folder,
    calibration_file="data/Levels.json"
)

# Print results
folder_results.head()
```

```
Analyzing files:   0%|          | 0/20 [00:00<?, ?it
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
[Warning] Signal resampled to 48 kHz to allow calcul
```

Customization and Flexibility

- **Reproducible Analysis**: YAML-based configs ensure consistent, replicable results.
- **Version Control Friendly**: Easily track, share, and collaborate on configurations.
- **Data Science Integration**:
  - Works with pandas for data manipulation
  - Compatible with matplotlib/seaborn and plotly for visualizations
  - Leverages popular Python libraries
- **Extensible Architecture**:
  - Allows addition of new metrics and methods
  - Adaptable to diverse research requirements
  - Potential for community-driven improvements

```python
# Customize analysis settings
new_config = {
  "MoSQITo": {
    "loudness_zwtv": {
      "statistics": ("avg", 5, 50),
      "func_args": {"field_type": "diffuse"}
    }
  }
}

analysis.update_config(new_config)
analysis.save_config("updated_config.yaml")
```

```yaml
# Settings file for batch acoustic analysis.
# This file configures which metrics are calculated and how they are processed.

version: "1.0"

# Python Acoustics Library Settings
# Supported metrics: LAeq, LZeq, LCeq, SEL   Mitchell, 23/08/2024, 19:15 • refactored binaural analysis functions
# Supported statistics: avg/mean, max, min, kurt, skew, any integer from 1-99
PythonAcoustics:
    # A-weighted equivalent continuous sound level
    LAeq:
        run: true  # Set to false to skip this metric
        main: 'avg'  # Main statistic to calculate
        statistics: [5, 10, 50, 90, 95, 'avg', 'min', 'max', 'kurt', 'skew']  # List of statistics to calculate
        channel: ["Left", "Right"]  # Channels to analyze
        label: 'LAeq'  # Label for the metric in output
        func_args:  # Additional arguments for the metric function
            time: 0.125  # Time interval for calculation (seconds)
            method: "average"  # Method for calculating Leq

    # Z-weighted (unweighted) equivalent continuous sound level
    LZeq:
        run: true
        main: 'avg'
        statistics: [5, 10, 50, 90, 95, 'min', 'max', 'kurt', 'skew']
        channel: ["Left", "Right"]
        label: 'LZeq'
        func_args:
            time: 0.125
            method: "average"

    # C-weighted equivalent continuous sound level
    LCeq:
        run: true
        main: 'avg'
        statistics: [5, 10, 50, 90, 95, 'min', 'max', 'kurt', 'skew']
```

Package Documentation

- **Comprehensive Documentation**:
  - Installation guide
  - Quickstart tutorials
  - API reference
  - Example notebooks
- **GitHub Repository**:
  - Open-source code
  - Issue tracking
  - Contribution guidelines
- **Regular Updates**:
  - Version changelog
  - Deprecation notices
  - New feature announcements

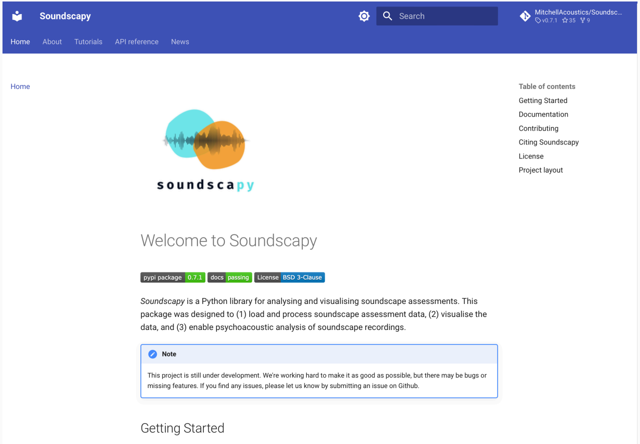
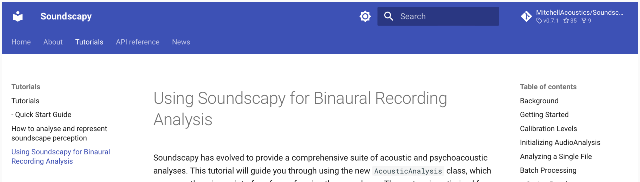


Figure 2: Soundscapy Documentation Homepage



Future Work: Predictive Models[1]

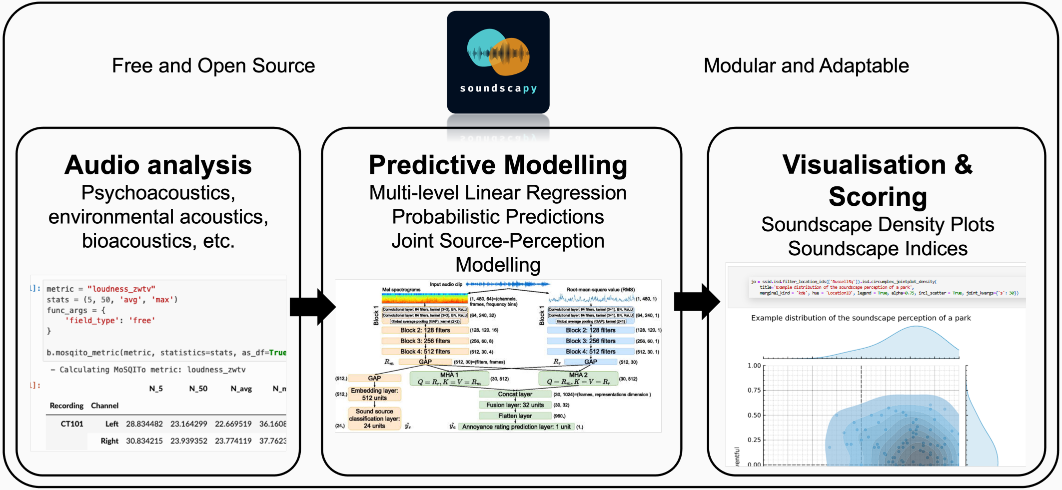


Figure 4: Modular Framework

- **Pretrained Models Integration**:
  - Predict soundscape perception from acoustic features
  - ~~Multiple model support for comparison~~
- **Design Optimization**:
  - Predict impact of soundscape interventions
  - Assist in urban planning decisions

[1] Hou et al. (2024)

Future Work: Soundscape Perception Indices (SPI)[2]

- **Single Index Calculation**:
  - Simplify complex soundscape data
  - Enable quick comparisons between locations
- **Standardized Reporting**:
  - Facilitate communication with stakeholders
  - Support policy-making processes
- **Customizable Weighting**:
  - Adapt to different cultural contexts
  - Account for varying research priorities
- **Applications**:
  - Urban planning and design optimization
  - Environmental quality assessment
  - Cross-cultural soundscape studies

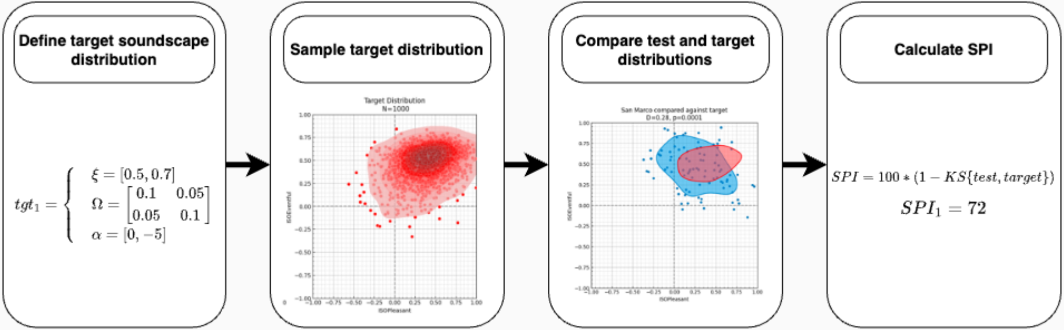


Figure 5: SPI Calculation Framework

[2]Mitchell et al. (2024)

Conclusion

- **Standardized Analysis**: Consistent with ISO 12913-3 guidelines
- **User-Friendly**: Easy-to-use Python package for researchers and practitioners
- **Powerful Tools**:
  - Distribution-based visualization
  - Comprehensive psychoacoustic analysis
  - Efficient batch processing
- **Future-Ready**:
  - Ongoing development for predictive models
  - SPI for simplified soundscape assessment
- **Get Involved**:
  - GitHub: github.com/MitchellAcoustics/Soundscapy
  - Docs: soundscapy.readthedocs.io
  - PyPI: `pip install soundscapy`

References

Aletta, Francesco, Andrew Mitchell, Tin Oberman, Jian Kang, Sara Khelil, Tallal Abdel Karim Bouzir, Djihed Berkouk, et al. 2024. "Soundscape Descriptors in Eighteen Languages: Translation and Validation Through Listening Experiments." *Applied Acoustics* 224 (September): 110109. https://doi.org/10.1016/j.apacoust.2024.110109.

Aletta, Francesco, and Simone Torresin. 2023. "Adoption of ISO/TS 12913-2:2018 Protocols for Data Collection from Individuals in Soundscape Studies: An Overview of the Literature." *Current Pollution Reports*, October. https://doi.org/10.1007/s40726-023-00283-6.

Axelsson, Östen, Mats E. Nilsson, and Birgitta Berglund. 2010. "A principal components model of soundscape perception." *The Journal of the Acoustical Society of America* 128 (5): 2836–46. https://doi.org/10.1121/1.3493436.

Hou, Yuanbo, Qiaoqiao Ren, Andrew Mitchell, Wenwu Wang, Jian Kang, Tony Belpaeme, and Dick Botteldooren. 2024. "Soundscape Captioning Using Sound Affective Quality Network and Large Language Model." https://doi.org/10.48550/ARXIV.2406.05914.

Mitchell, Andrew, Francesco Aletta, and Jian Kang. 2022. "How to Analyse and Represent Quantitative Soundscape Data." *JASA Express Letters* 2 (3): 037201. https://doi.org/10.1121/10.0009794

Also check out my other package for even more advanced circumplex analysis, currently under development!

This presentation is available online at https://mitchellacoustics.quarto.pub/interno
2024-soundscapy-pres