Spotlighting the Participant: An Interactive Bokeh Visualization App for Soundscape Data Exploration

Project Report*

Fabian Rosenthal
Master Medientechnologie
Technische Hochschule Köln
fabian.rosenthal@smail.th-koeln.de

July 3, 2023

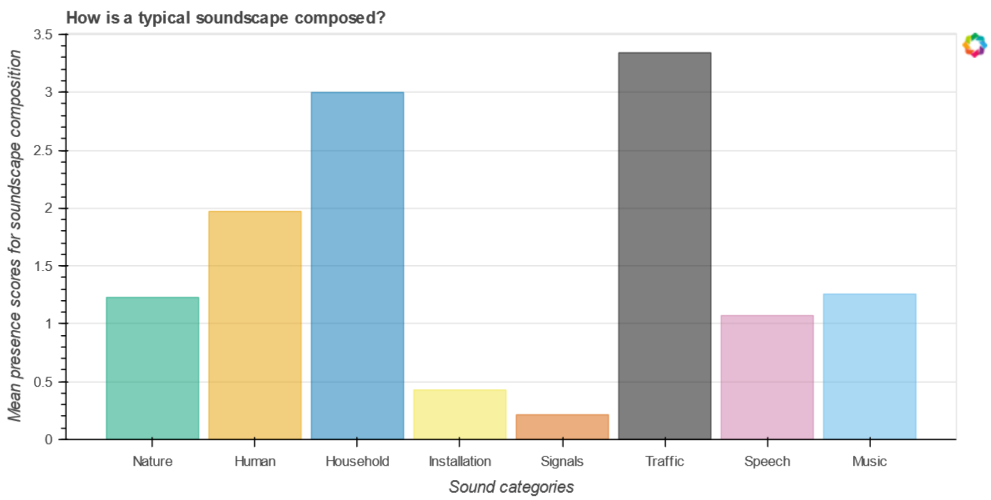

Figure 1: Example barplot representing the mean presence scores for categories in the soundscape composition of a *single* participant.

Abstract

In field study evaluation it is often difficult to gather a good intuition for the variation in participants and how they used the study instruments at hand. Single participants and their properties might contribute to overall statistics but might as well be hidden in the mass of all participants. Using a large crowdsourced dataset of Indoor Soundscapes, this report introduces an interactive application for Bokeh in Python. Hereby, it tries to reconnect with the data and explore it's properties by focussing on the participants. In order to achieve that, key aspects defining a single participant are viewed, timeseries as a series of the multiple observations during the study can be accessed and feature relationships can be explored. Furthermore, visually decisive data points can be selected and logged as marked by the user in a file. This adds another use of this app and makes it straight forward do take actions after visually identifying outliers in the data.

Keywords Data Visualization · Indoor Soundscape · Bokeh

*The code is available on `https://github.com/radlfabs/DS_Data_Visualization_2023_Fabian_Rosenthal` under MIT license.

1 Introduction

Soundscape research is done in many different fashions. Studies may be held completely online asking participants to report retrospectively Versümer et al. [2021], Torresin et al. [2022]. Other approaches ask participants on occasion and on scene to take a survey Mitchell et al. [2020]. Furthermore, studies may focus on creation of artificial scenarios merging real recordings with artificial masker sounds which are then rated by participants Lin [2023]. Gathering in-situ ratings of subjective soundscape perception usually comes with a lot more expense. However, it can be achieved by utilizing the Experience Sampling Method. This leads to complex datasets with a large number of features and attributes.

From an evaluation perspective, we, as researchers, try to ask the most comprehensive and meaningful research questions possible. This is, of course, critical to moving research forward. But in the process, nuances of what a data set is capable of saying can fall by the wayside.

To be specific: Fine nuances in data can not always be represented with statistical modelling or machine learning methods. Models might either show weak effects or low values in evaluation metrics, such as R-squared. This is foremost seen in gaps between lab experiments and data collected on-scene. On the other hand, large models in terms of capacity have the danger of overfitting to the data and are not desired either.

For the case of Indoor Soundscapes, this was the motivation to start a visualisation project with Bokeh in Python Bokeh Development Team [2018]. The goal is to be able to tell all kinds of little side stories, which would never pop up when the whole book was told by a statistical model.

This project makes use of a large crowd-sourced dataset of Indoor Soundscapes as published by Versümer et al. [2023a]. The dataset includes constant and time-variable personal and situational factors, multiple subjective ratings of soundscapes as well as objective measurements by the study equipment. With a number of 105 participants and 6594 total observations there is a lot to discover. While previous approaches dealt e.g. with modelling of subjective loudness evaluations as in Versümer et al. [2023b], we now focus on single participants and their properties. This gives us the opportunity to deal with the time series that arise when we look at all the observations given by a single participant in chronological order.

2 Material and Methods

In order to set the spotlight on the single participant, we have to find suiting visualization aesthetics for the constant as well as for the time varying data of the participants. Therefore, this project defines different sections in the app to respect the properties of the underlying data: We have to find meaningful visual encodings of the constant person factors as well as the subjective ratings stringed together to timeseries. Also we may want to show feature relationships as well. For the time series of observations, a new level of interpretation shall be achieved in the context of the person factors.

2.1 Controlling the App

The user of this app gets the possibility to observe different participants by selecting them using a slider widget (see fig. 2). The slider browses through the participants sorted by age (youngest to oldest). All visualizations and texts (if necessary) are updated accordingly. Some custom `CSS`-style is used to make the slider widget stick to the top right of the page and to add transparency. In addition, the user has the possibility to log participant IDs and selection keys corresponding to rows in the original data frame to find observations of interest after using the visualization app. Furthermore, all slider changes are tracked in the log file as well. This allows an analysis of what exactly has been viewed, afterwards.

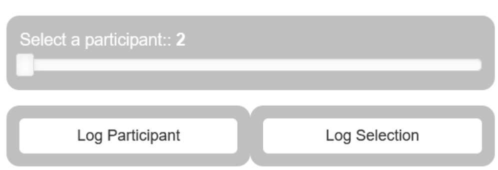

Figure 2: The slider widget to control the participant selection and its neighbors, the logging buttons. For selection purposes, participants are sorted by their age from youngest to oldest.

2.2 Personal Section & Radar Charts

At the top of the collections of visualizations, some selected key aspects of a single person are shown. We want to gather a quick overview on the circumstances the person is living in. This can be practical the number of people in the household but also some aggregated features that denote if a person feels well or unwell. Variables selected for this matter are mostly subjective ratings of psychological wellbeing, health, trait and noise sensitivity. These had been assessed with appropriate questionnaires before the study time. Hearing ability has been measured with audiograms. To make all this factors visually comprehensible they are separated into three sections where they can be visualized with radar charts. The axes are assigned with a single person factor. As a result we can create three radar charts adressing general physical and psychological health, noise sensitivty by work, sleep or habitation, as well as trait with its dimensions mood, wakefullness and rest. When all those features are scaled to the range of $[0, 1]$ and are similarly aligned in their connotation (e. g. from healthy to unhealthy, from good hearing ability to hearing impairment), it gets straight forward to understand the plots as a user: large, wide spread area of the polygon in the radar chart corresponds to a high expression in this person dimension.

2.3 Characterization of a Person's Study Time

In the next section we want to incorporate the timestamps of the observations. Specifically, we want to observe, in which hour of the day the single participant typically performed observations in the study. Therefore, we extract the hour of the Form_finish timestamp which marks the endpoint of a succesful observation. Then we count the number of observations that fall into the discrete hours and get a time histogram as an result. This gives us a clear and quick indication whether a participant did our study from *nine to five*, or if he or she had specific times of the day to focus on the study. In terms of aesthetics used for that plot this project uses a colored barplot. The barplot respects the discreteness of the hourly summed observations and avoids pseudo-continuity. A custom colormap was added. It uses a divergent and mirrored set of colors, which starts with deep purple for the hour zero, gets to bright yellow until twelve at noon and gets darker again until midnight. This just makes it nice to look at and adds a simple centering around the mid of the day to the barplot. This makes it easy to understand, even if you do not focus on the x-axis. The ability of the user to recognize the colors correctly is not important in this case. The centering works just as well if the color differences are perceived as shades instead of hue due to disabilities.

2.4 Soundscape Composition

A next plot shows typical (averaged) soundscape composition. An example can be observed in fig. 1. The participant rated the presence of nine sound source categories for every observation with the use of a scale. So averaging over all observations per participant gives us a nice overview on what the participant was surrounded with on average. A barplot is a good aesthetics encoding the mean presence scores with the length of the bar because viewers will be able to understand differences quite easily. Also we use a colorblind friendly colormap to visually separate categories. Therefore, the mapping was adjusted to support the categories' characteristic (e.g. mapping the color green to the category nature).

In addition to the general person overview, e.g. containing the age, we now get a picture on how the participant behaves and what the typical habits are.

In the following sections we now really focus on the time series data corresponding to the series of observations every participant generated. So several from a soundscape theoretically stand point most important features have been selected to be plotted over time as well as against each other as scatter plots. This utilises the experience sampling method to the fullest and is one of the first attempts to visualize trends in the person's observations in this data set. And not only trends and habits can be observed with the time series plots. We also can gather a clear and nice overview on how questionnaires and scales of the study instruments where used and if that matches the intended use. We can also spot outliers quickly. Plotting feature over feature not only shows their general relation for a single participant but also gives insight into the person as well. This is especially true for the psychological dimension Valence and Arousal. These two orthogonal dimensions span a wide psychological space and clusters of observations falling into quadrants of this space can be intepreted accordingly. Therefore, this relationship plot seems to be pretty important to gathe information on the participant's psychological state. Also, we can distinguish between person types quickly, e.g. when observing a skewed distribution of data points towards maybe the quadrant of low arousal and low valence vs. as cluster in the high valence and high arousal quadrant. This gives deep individual insight, which cannot really be gathered in another way.

2.5 Implementation

This interactive visualization app was developed with Bokeh for Python. We are using a local `Bokeh` server which runs the python backend in addition to some cusom JavaScript code used in callback functions. During development, the elementary challenge has been the efficient switching of data on slider changes. The following section describes steps taken to reduce computational effort: In oder to create all the different visuals which are relying on some additional scaled features, grouped data, and time series efficiently, we precompute as much as possible at the first start up of the application. The next start-ups then make use of a pickled data source and will run slightly quicker. Generally speaking, we try to reduce computation on slider changes to changing data sources (e.g. by indexing operations or changing views on the data source by filtering). We try to avoid computation of new values completely and do not create any new figures on change of the slider. For any created figure the underlying data is just changed. To acomplish this, all angles, coordinates and features for the radar charts are precomputed completely, so that this data can be replaced quickly in the plots when the slider value is changed. Furthermore, all necesssary group-by operations and aggregations are precomputed, as well. In order to make the main app file as readible as possible, widgets, plotting functions, divs and constants have been moved to separate modules. Also, logging and preprocessing have been assigned to their own module. Besides `visualization_app.py`, which holds the main elements of the program and which constructs the layout of the app, the repository consists of the following modules:

- `widgets`: Definition of the slider, buttons, as well as their `CSS`-styles used in the main app.
- `plotting`: Definition of all functions responsible for plotting. They always will return a `Bokeh` figure object.
- `divs`: Declaration of Div objects and their texts and properties.
- `constants`: Declaration of relevant constants and variable selections including a dictionary with relevant tooltips for the hover/mouse-over functionality.
- `logging` provides all functionality related to the logging of participants and selected data points during app runtime.
- `preprocessing` holds functions used at first runtime of the app which help prepare the data for plotting.

The following preprocessing steps have been done to the original study data:

- Implement a custom MinMax- and Standard-Scaler to not depend on large packages such as `scikit-learn`.
- Sort the rows of the dataset by the age of the participants but preserve the order of observations within one participant.
- To access the person factors which are constant for each participant, we precompute a constant dataframe which has the length of number of participants. This can be indexed directly.
- The constant features are rescaled using our custom Scaler classes. Also we redirect two features: *Hearing_ability* is computed as $1 -$ *Hearing_impairment*. *Resilience* is computed as $1 -$ *Anxiety*.
- The data for the radar charts is stored in a *Python dictionary* of *pandasḊataFrames*.
- The values of the *common_location* columns are extracted with the help of the regex module to make it a bit prettier as plain text: `str.replace(r"(\w)([A-Z])", r"\1 \2", regex=True)`
- The hourly counts statistic is precomputed by performing a *group-by* operation on the hour of the time stamps followed by a call of the `count` function in Pandas.
- The radar charts are precomputed as *Angles* $=$ `numpy.linspace(0, 2*pi, N_{properties})` and the coordinates for the plots can then be obtained by taking the $\sin($*Angles*$)$ and $\cos($*Angles*$)$.

In the case of the person data div, we use a grouped data source with distinct rows per participant which can easily be indexed by the slider value. The precomputed radar charts data is stored in Numpy arrays and is directly used as the figures data source. All time series and feature relation plots share a single Bokeh ColumnDataSource. We invoke a change in the plots by changing CDSview objects on change of the slider. The view enforces a BooleanFilter on the ColumnDataSource in order to filter for only those observations which correspond to the selected participant. This turns out to be a straight forwards and efficient way to update multiple plots at the same time.

3 Results and Discussion

In order to analyze if and how the selected aesthetics work with the given data set of crowd-sourced Indoor Soundscapes, this section will show some simple case-examples as a demonstration.

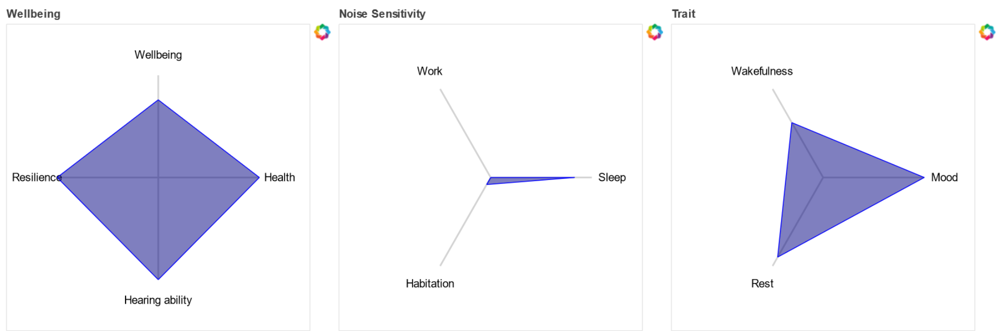

Figure 3: Radar charts of a 23 year-old male participant.

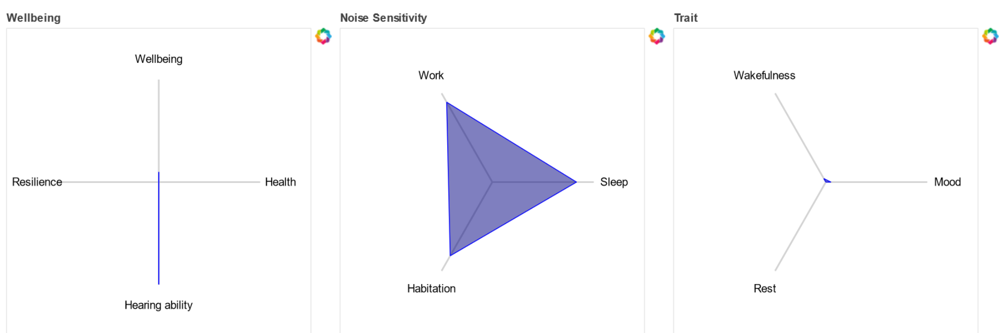

Figure 4: Radar charts of a 19 year-old female participant.

Let's explore the constant person factors represented by radar charts first. By browsing through the data using the slider we can quickly observe different characteristics of participants. First, we will have a look at a 23 year-old male participant in fig. 3. We can observe large areas in radar charts one (Wellbeing) and three (Trait), whereas radar chart two shows little expression (Noise Sensitivity). The interpretation seems fairly simple: The participant scores rather balanced across all wellbeing-, health- and trait-related questions. The audiogram shows no or very low impairment. The observation of the middle placed radar chart shows a relatively high noise sensitivity in his sleep. Figure 4 on the other hand refers to a participant showing a completely different profile: Our radar charts show perceived problems in almost all ares of the questionnaires. The health and wellbeing scores are low combined with high noise sensitivity. The objective measures of hearing ability show no problems, though. We quickly get an impression on where self described problems may lie with the selected participant. On the other hand, our radar chart seems to be not sufficient visualizing single category values or differences, when all other dimension show opposing values. This could be improved in the future, by adding and resizing single value points on the axes, so that values (single dimensions) that stand out from the radar charts get more visibility.

Our visualization app can also be useful for the porpose of finding *black sheep*. It can help identify problems in the data, i. e. on the time aspects as well as the scales used: Fig. 5 shows how the visualization can help to quickly identify cases where obersvations are given in only a small timerange of the participation in the study. Such study participation is problematic because it could be cause of of bias in subjective scoring as well as reduce variance in sound sources. On the other hand, problems with the use of scales can also be identified. In fig. 6 we can observe how a participant has only used the main verbal loudness categories for the subjective loudness scores. The noticeable jitter may appear because the participants had to touch the fine-tuning slider in the study app, in order to be directed forwards. However, the scale has been used rather discretely than continuously (which was intended).

Such little and simple observations can be made all over this data set when exploring it using this visualization app. Also, the principles applied for the current data set in use can be applied to many other data sets from the Soundscape research field as well. The modular app design can be adapted to your personal needs and wishes. Furthermore, functionality can easily be added later on.

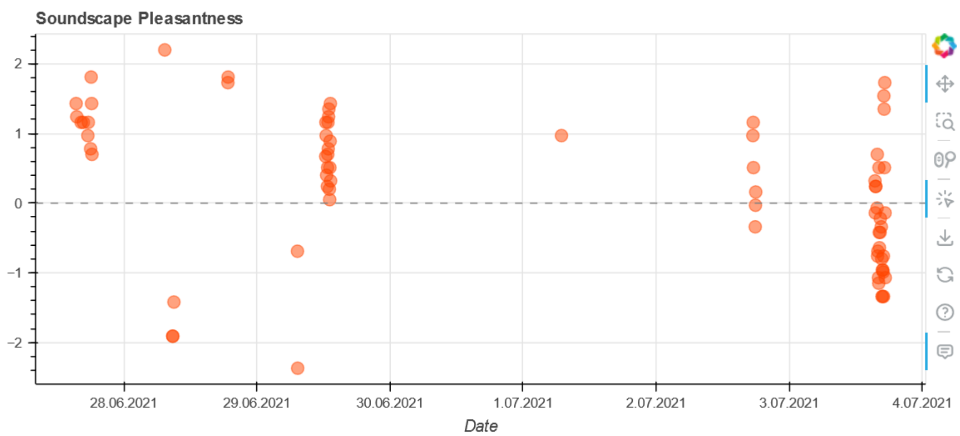

Figure 5: Problematic spread of observations over time: A large number of observations has been given in only a couple of hours.

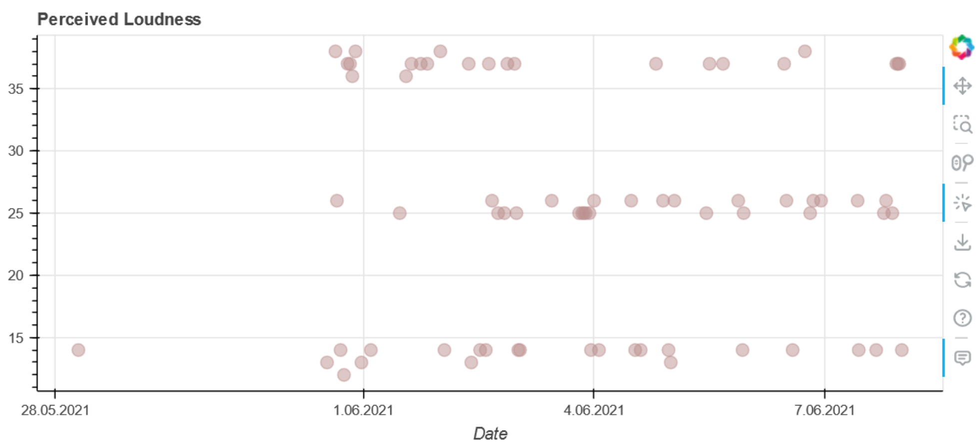

Figure 6: Problematic use of study instruments: Here, the scale for perceived loudness was not used as intended which results in a poor feature distribution.

4 Future Contributions

In the future, the radar charts can be improved by adding points to the axes for single dimensions. This will improve readibility and visibility in cases where single dimensions stand out in a radar chart (e.g. where one dimnesion is not zero, but all other dimensions are. At the moment, this is an edge case hard to visually understand). Regarding the tooltips for the mouse-over functionality, there seems to be a problem when multiple points are close together. The mouse-over tooltip then does not really follow the user's cursor but sticks weirdly to single data points which then are hard to switch with the mouse. More generally, it would be preferable to make this visualization run locally in the browser. Therefore, it would be necessary to replace the Bokeh server elements with custom JavaScript. It would also be interesting to measure performance comparing the Bokeh server solution as well as the solution where it runs in the browser.

5 Conclusion

This report introduces an interactive and modular data visualization app using Bokeh for Python. It is specifically developed to shed light on the participants of a large crowd-sourced dataset of Indoor Soundscapes published by Versümer et al. [2023a]. Our example cases show how easy it is to find interesting views and aspects in the data that had been hidden before. Our logging functionality makes it also straightforward to mark observations as problematic and find them later on by reading out the log files. This really adds another connection of visualization, understanding of the data as well as taking actions accordingly.

References

Siegbert Versümer, Jochen Steffens, and Patrick Blättermann. Utilizing the percentile lasso to identify variables that influence retrospective judgments of everyday sounds 6. community response to noise and noise annoyance. 06 2021.

Simone Torresin, Eleanor Ratcliffe, Francesco Aletta, Rossano Albatici, Francesco Babich, Tin Oberman, and Jian Kang. The actual and ideal indoor soundscape for work, relaxation, physical and sexual activity at home: A case study during the covid-19 lockdown in london. *Frontiers in Psychology*, 13, 12 2022. doi:10.3389/fpsyg.2022.1038303.

Andrew Mitchell, Tin Oberman, Francesco Aletta, Mercede Erfanian, Magdalena Kachlicka, Matteo Lionello, and Jian Kang. The soundscape indices (ssid) protocol: A method for urban soundscape surveys—questionnaires with acoustical and contextual information. *Applied Sciences*, 10(7), 2020. ISSN 2076-3417. doi:10.3390/app10072397. URL `https://www.mdpi.com/2076-3417/10/7/2397`.

Hao Weng Lin. Recording and evaluating soundscapes in singapore, 2023. URL `https://dr.ntu.edu.sg/handle/10356/166909`.

Bokeh Development Team. *Bokeh: Python library for interactive visualization*, 2018. URL `https://bokeh.pydata.org/en/latest/`.

Siegbert Versümer, Jochen Steffens, and Fabian Rosenthal. Extensive crowdsourced dataset of in-situ evaluated binaural soundscapes of private dwellings containing subjective sound-related and situational ratings along with person factors to study time-varying influences on sound perception — research data, April 2023a. URL `https://doi.org/10.5281/zenodo.7858848`.

Siegbert Versümer, Jochen Steffens, and Stefan Weinzierl. Day-to-day loudness assessments of indoor soundscapes: Exploring the impact of loudness indicators, person, and situation. *The Journal of the Acoustical Society of America*, 153:2956, 05 2023b. doi:10.1121/10.0019413.