The Journal of Open Source Software

SweetBean: A declarative language for behavioral experiments with human and artificial participants

Younes Strittmatter [1,3¶] **and Sebastian Musslick** [2,3]

1 Princeton University, USA **2** Osnabrück University, Germany **3** Brown University, USA ¶ Corresponding author

DOI: 10.21105/joss.07703

Software

- Review
- Repository
- Archive

Editor: Stefan Appelhoff

Reviewers:

- @jodeleeuw
- @hauselin

Submitted: 09 December 2024
Published: 20 March 2025

Summary

sweetbean is an open-source, domain-specific declarative programming language built in Python, designed to simplify the synthesis of web-based behavioral experiments. It allows researchers to specify a behavioral experiment in declarative form as a sequence of events. Once specified, sweetbean can compile the experiment into a jsPsych experiment (Leeuw et al., 2023) for web-based behavioral study with human participants. In addition, sweetbean can generate prompts for conducting the same experiment with a large language model (LLM), enabling automated alignment of LLMs with human behavior.

The sweetbean package integrates with other tools that automate aspects of behavioral research, such as sweetpea (Musslick et al., 2020) for automating experimental design, or autora (Sebastian Musslick, 2024) for orchestrating closed-loop behavioral research studies. Together, these tools form an ecosystem for advancing behavioral research through automated scientific discovery.

Statement of need

The generation and execution of web-based experiments is a common task in the behavioral sciences. However, the process of designing and implementing such experiments can be difficult, especially if researchers don't have a background in web development. While platforms like jsPsych offer powerful solutions for running online experiments, their reliance on *JavaScript* can be a significant barrier for researchers who lack programming expertise in the language. Many researchers in the behavioral sciences, however, are already familiar with *Python* due to its widespread use in data analysis. Other platforms written in *Python*, like PsychoPy (Peirce et al., 2019), have made significant strides in addressing this issue by providing a graphical user interface for experiment design, along with functionality for exporting executable experiments into Python and JavaScript. However, PsychoPy is not specifically tailored for creating online JavaScript experiments, which can result in compatibility issues and unsupported PsychoPy components. sweetbean aims to simplify the process of generating online experiments by providing an intuitive declarative language for specifying experiments in Python. By abstracting the details of web development, sweetbean enables researchers to concentrate on the specification of the experiment itself, rather than on its implementation details.

Another challenge in behavioral research is the integration of LLMs into the experimental workflow. LLMs have the potential to simulate human behavior in a variety of tasks, making them valuable simulators for behavioral experiments (Binz et al., 2024; Dillion et al., 2023; Manning et al., 2024). Accordingly, researchers are increasingly interested in aligning the behavior of LLMs with human participants, calling for the development of tools that facilitate this alignment. sweetbean addresses this challenge by providing a way to generate prompts

for LLMs based on the same experiment specification used to generate web-based experiments for human participants. This allows researchers to easily generate synthetic behavioral data with LLMs, compare the behavior of LLMs to human participants, and to align LLMs with human behavior.

Beyond reducing technical barriers and facilitating alignment between machines and humans, `sweetbean` enhances reproducibility and collaboration by standardizing experiment specifications within a flexible and intuitive framework. This standardization not only saves time but also provides more robust and scalable experimental workflows for the behavioral research community.

When to Use SweetBean, jsPsych, or PsychoPy?

Several tools exist for designing and running behavioral experiments, each with different strengths depending on the research context. Here, we focus on `sweetbean`, `jsPsych`, and `PsychoPy` due to their prominence in web-based and Python-based experiment design.

The choice between `sweetbean`, `jsPsych`, and `PsychoPy` depends on the researcher's needs, technical expertise, and experimental requirements:

- Use `sweetbean` if you are comfortable with Python and want a declarative workflow that integrates with Python-based tools for experimental design, analysis, and automation. `Sweetbean` is particularly useful for comparing LLM-based experiments with human participants or embedding experiment generation within an automated research pipeline using `sweetpea` and `autora`. It is designed to be accessible and simple, but it is not as feature-rich or general-purpose as `jsPsych` or `PsychoPy`. While existing `jsPsych` plugins may not be directly supported, new plugins can be added based on community demand.
- Use `jsPsych` if you require full flexibility in JavaScript and access to its extensive ecosystem of extensions. If deep customization of experiment logic or fine-tuned control over the execution environment is necessary, `jsPsych` is the better choice. Maintained by a large community, it offers a broader range of plugins and features compared to `sweetbean`. Use `jsPsych` if you need a general-purpose tool for online experiments and are comfortable working with JavaScript.
- Use `PsychoPy` if you prefer a GUI-based approach or need to run experiments offline. While `PsychoPy` supports Python scripting, it is primarily designed for local (rather than web-based) experiments. It is a powerful tool for designing and running a wide range of studies, particularly those requiring precise timing and control over stimulus presentation. Use `PsychoPy` if you want a GUI-based experience, need to run experiments offline, or require high-precision timing control.

Acknowledgements

Y. S. and S.M. were supported by the Carney BRAINSTORM program at Brown University, as well as the National Science Foundation (2318549). S. M. also received support from Schmidt Science Fellows, in partnership with the Rhodes Trust.

References

Binz, M., Akata, E., Bethge, M., Brändle, F., Callaway, F., Coda-Forno, J., Dayan, P., Demircan, C., Eckstein, M. K., Éltető, N., & others. (2024). Centaur: A foundation model of human cognition. *arXiv Preprint arXiv:2410.20268*. https://doi.org/10.48550/arXiv.2410.20268

Dillion, D., Tandon, N., Gu, Y., & Gray, K. (2023). Can AI language models replace human participants? *Trends in Cognitive Sciences*, *27*(7), 597–600. https://doi.org/10.1016/j.tics.2023.04.008

Leeuw, J. R. de, Gilbert, R. A., & Luchterhandt, B. (2023). jsPsych: Enabling an open-source collaborative ecosystem of behavioral experiments. *Journal of Open Source Software*, *8*(85), 5351. https://doi.org/10.21105/joss.05351

Manning, B. S., Zhu, K., & Horton, J. J. (2024). *Automated social science: Language models as scientist and subjects*. National Bureau of Economic Research. https://doi.org/10.48550/arXiv.2404.11794

Musslick, S., Cherkaev, A., Draut, B., Butt, A. S., Darragh, P., Srikumar, V., Flatt, M., & Cohen, J. D. (2020). SweetPea: A standard language for factorial experimental design. *Behavior Research Methods*, 1–25. https://doi.org/10.3758/s13428-021-01598-2

Peirce, J., Gray, J., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*. https://doi.org/10.3758/s13428-018-01193-y

Sebastian Musslick, C. C. W., Benjamin Andrew. (2024). AutoRA: Automated research assistant for closed-loop empirical research. *Journal of Open Source Software*, *9*(104), 6839. https://doi.org/10.21105/joss.06839