

Python for the practicing neuroscientist: an online educational resource

Emily Schlafly¹, Anthea Cheung², Samantha W Michalka³, Paul A. Lipton⁴, Caroline Moore-Kochlacs¹, Jason Bohland⁵, Uri T. Eden², and Mark Kramer^{2,6}

¹Graduate Program in Neuroscience, Boston University, Massachusetts 02215

²Department of Mathematics and Statistics, Boston University, Boston, Massachusetts 02215

³Olin College of Engineering, Needham, MA 02492

⁴Princeton School of Public and International Affairs, Princeton University, Princeton NJ 08544

⁵Department of Communication Science and Disorders, University of Pittsburgh, Pittsburgh, PA 15260

⁶Center for Systems Neuroscience, Boston University, Boston, Massachusetts 02215

July 1, 2020

Abstract

As neuronal data accumulates worldwide, accessible - yet rigorous - resources to develop hands-on experience with modern data analysis techniques are required. We present here an online educational resource for neural data analysis (<https://mark-kramer.github.io/Case-Studies-Python>). To reach the biologists, psychologists, and clinicians collecting neuronal data, we assume only a basic mathematics background, common to those trained in biological sciences. Through an interdisciplinary case-study approach, we use real-world data to motivate the study of modern quantitative analysis methods in Python. A modular format provides multiple coherent learning paths through the material, and thereby allows personalized learning for individuals with varying quantitative backgrounds and research interests, and flexible curation of material for redeployment in other curricula. Developed using Jupyter notebooks, the material supports fully interactive environments in most web browsers, and hosted on GitHub, the material is freely available for reuse, modification, and further development by the community.

Introduction

Sophisticated data collection requires sophisticated data analysis: every neuroscientist now requires skills in computer programming, statistics, and mathematics [1], [2]. However, most experimental and clinical neuroscientists lack sufficient training in these skills, and the idea of tackling a standard technical course or textbook on data analysis without sufficient tutorials, examples, and hands-on practice is daunting. Therefore, a fundamental gap has emerged in neuroscience labs between highly sophisticated and rich data collection procedures, and limited training to assess these data with state-of-the-art quantitative techniques. While important to training the next generation of neuroscientists, challenges exist in computational neuroscience education [1], compounded by recent requirements for remote learning and virtual classrooms [3].

To help address these challenges, we present here an online, freely available resource to enhance training in neural data analysis: <https://mark-kramer.github.io/Case-Studies-Python>. To reach the practicing neurosci-

entist, we assume only a basic knowledge of calculus and statistics, common to those trained in biological sciences [1]. We also invert the standard presentation of data analysis techniques. Typical statistics textbooks in data analysis begin with the development of theory and then describe applications. We instead begin each topic with an example case study of neural data (e.g., action potentials from rat hippocampus or scalp electroencephalogram data recorded from a human subject). These data then motivate the development and application of modern analysis techniques (e.g., visualization approaches, spectral analysis, bootstrapping, and generalized linear models). We emphasize a hands-on approach; example data sets are provided, and computer (Python) code interspersed within the material encourages direct interaction with the concepts. The data, analysis methods and code are not toy examples, but instead correspond directly with modern techniques in use today. Upon completing each case study, the learner will be able to immediately deploy these data analysis tools in his or her own research.

Results

All notebooks have consistent structure. This structure consists of: (1) A brief synopsis of the notebook data and analysis methods. (2) An on-ramp to immediately introduce - and evaluate in Python - a core concept of the notebook. (3) A description of the data, method of collection, and scientific question motivating analysis. (4) A detailed statement of the module goals and the data analysis tools to be studied. Then, (5) Data analysis begins with visualization approaches. (6) Data analysis continues with implementation of the techniques specific to the notebook, including development of appropriate theoretical material and practical implementations. (7) Data analysis concludes with rendering of results.

Notebooks are modular and can be completed in any order. Although we organize the notebooks sequentially, each notebook is independent of any other. In this way, learners and educators may develop individual learning paths through the material. For example, a learner with limited programming experience who is interested in the fundamentals of EEG analysis could begin with the extensive Python tutorial (Notebook #1), and then proceed to notebooks introducing the evoked response potential (Notebook #2) and spectral analysis (Notebooks #3 and #4). Or, an educator teaching a graduate course in computational neuroscience may introduce the topic of cross-frequency coupling (CFC, an active research area [4]) and provide students with a hands-on example of implementing and applying a measure of CFC (Notebook #7). A motivated individual could also complete the entire sequence of notebooks, in order.

GitHub, Python, and Jupyter notebooks for an interactive online learning environment. To facilitate an open, freely available distribution of the material, we use the programming language Python (<https://www.python.org/>), host all material on GitHub (<https://github.com/>), and organize the material into Jupyter notebooks [5], each focused on analysis of a specific data set. To interact with the material, a learner may download (or clone) the entire repository (including all data, Python code, and written material) for use on his or her own device. Alternatively, a learner may interact directly with the material through his or her web browser; no downloads or software installations required. Using Binder (<https://mybinder.org/>) or ThebeLab (<https://opendreamkit.org/>), the Jupyter notebooks become fully interactive environments; the data analysis (Python code) becomes executable and modifiable directly in most modern web browsers.

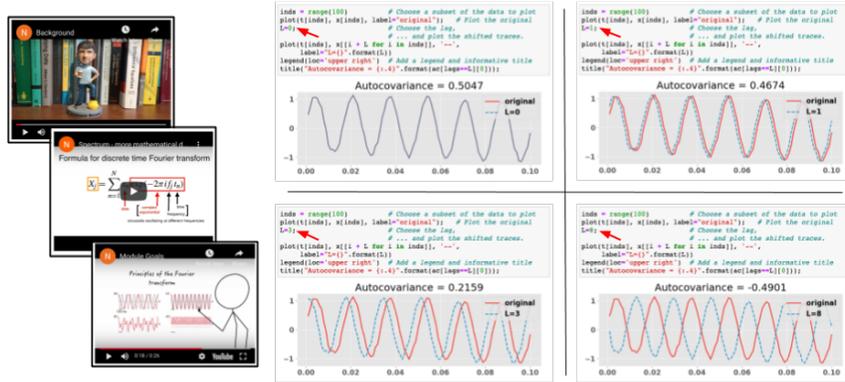


Figure 1. Notebooks introduce concepts with short videos and interactively explore mathematical ideas. In this example, the learner explores how the autocovariance is related to the lag between signals. Changing the parameter L (red arrows) within the executable notebook supports interactive learning, visualization, and exploration.

Use cases for learners and educators. The material was developed for investigators of any ability and at all career levels - spanning from the beginning undergraduate researcher to the established PI - to analyze and understand neural data. For those new to Python, Notebook #1 is dedicated to a thorough but rapid and practical introduction to the use of Python 3. We anticipate many scenarios for participation, for example:

- A clinical research lab requires new data analysis skills in scalp EEG. Lab members complete notebooks describing analysis of brain rhythms and rhythmic coupling of field data (Notebooks #3-5).
- A postdoctoral researcher requires modern spike-field coherence analysis techniques as part of her ongoing research. An expert in Python, she completes a single notebook (Notebook #11).
- An instructor plans to update his undergraduate laboratory course in basic neuroscience. In the lab, students record spike train data. The instructor augments this lab by requiring students complete a notebook introducing spike train analysis (Notebook #8) and apply basic visualizations to the spike train data recorded in lab.
- An instructor must suddenly teach her neuroscience course remotely. She selectively chooses content from a subset of notebooks to aggregate into a coherent curriculum. Students complete notebooks outside of class, and discuss material during (virtual) classroom time [3].

We expect interested researchers, students, and educators in the neuroscience community will develop many other uses beyond the examples listed here.

Notebooks are dynamic and available for further development by the community. The collaborative nature of GitHub allows for dynamic evolution of the material. A modification (e.g., an error correction) suggested by any learner can be flagged as an issue and incorporated into the material; versioning allows a complete record of attributable changes. Moreover, the repository is expandable. For example, a tutorial that implements and applies a new data analysis method to an example data set may be contributed as a standalone notebook. In this way, the community’s expertise allows the material to grow and include important topics not yet presented (e.g., analysis of imaging data).

Conclusion

We present here an online platform for training in neural data analysis (<https://mark-kramer.github.io/Case-Studies-Python>). The material is motivated by real-world examples in neural data analysis, modular,

executable and modifiable directly in most modern web browsers, and freely available for reuse and further development.

Acknowledgements

The authors acknowledge support from NIH NIGMS R25GM114827 and NSF DMS #1451384 (MAK). The authors thank MIT Press for publishing a complementary version of this material.

References

- [1] M. S. Goldman and M. S. Fee, “Computational training for the next generation of neuroscientists.,” *Curr Opin Neurobiol* , vol. 46, pp. 25–30, Jul. 2017, doi: 10.1016/j.conb.2017.06.007.
- [2] M. A. Kramer and U. T. Eden, *Case Studies in Neural Data Analysis* . Cambridge, MA: MIT Press, 2016.
- [3] S. Sandrone and L. D. Schneider, “Active and Distance Learning in Neuroscience Education,” *Neuron* , vol. 106, no. 6, pp. 895–898, Jun. 2020, doi: 10.1016/j.neuron.2020.06.001.
- [4] A. Hyafil, A.-L. Giraud, L. Fontolan, and B. Gutkin, “Neural Cross-Frequency Coupling: Connecting Architectures, Mechanisms, and Functions.,” vol. 38, no. 11, pp. 725–740, Nov. 2015, doi: 10.1016/j.tins.2015.09.001.
- [5] “Introducing Binder 2.0 — share your interactive research environment,” *eLife* , Nov. 30, 2017. <https://elifesciences.org/labs/8653a61d/introducing-binder-2-0-share-your-interactive-research-environment> (accessed Jun. 19, 2020).