Tweet us:
@EischLab
@AJEisch
@yun_Sanghee
@itsmeraybye

Streamlined Analysis for Convoluted Rodent Touchscreen Data by Building a Customized Python Application

Raymon Shi[c], Harley A. Haas[c], Priya L. Kumar[c], Grace L. Bancroft[c], Ian F. Sanchez[c], Amelia J. Eisch[a,b], Sanghee Yun[a,b]

[a] Department of Anesthesiology and Critical Care Medicine, The Children's Hospital of Philadelphia (CHOP) Research Institute, Philadelphia, PA 19104
[b] Perelman School of Medicine, University of Pennsylvania, Philadelphia, PA USA 19104
[c] University of Pennsylvania, Philadelphia, PA USA 19104

QUESTIONS

- How can Python be used to merge massive amounts of data (6 months), omit duplicate data rows, automatically calculate average accuracy and latency duration in each reversal, and sort by test parameters?
- Can the Python app be versatile to be applicable across different touchscreen tests?
- Can the Python app be utilizable by experiementers with limited programming knowledge by providing error handling for user inputs?

APPROACH

- Built in coding for merging raw data files, omitting duplicate data, extracting and calculating parameters from raw data, and producing a CSV file with the organized data using Pandas.
- Built a graphical user interface with a main menu, sub-menus per touchscreen test paradigm, and multiple buttons with different functionalities using the Tkinter package.
- Provided error handling by anticipating user input errors and providing descriptive explanations for the cause of errors using Python error exception handling.

RESULTS

- Programmed a user-friendly app that can analyze different touchscreen tests (i.e. general touchscreen train, location discrimination train and test, extinction tests).
- Sorting output based on test parameters (i.e. difficulty, test day, animal ID, or block number) within the same test paradigm.
- The app has the ability to have interchangeable criteria values for different test paradigms (i.e. completed days, omission #, or reversal #).
- The app explicitly handles input errors via descriptive error messages.

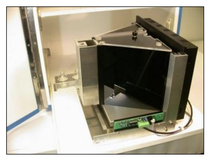
IMPLICATION

- Significantly reduced the amount of time and resources used for the touchscreen data analysis.
- Minimized human errors produced during manual calculations.
- Enabled easy data sharing with the organized output as a CSV file.

ABSTRACT

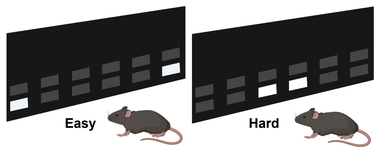
Touchscreen tests are one of the breakthrough models of measuring cognitive performance, such as pattern separation and cognitive flexibility, in both animal and human experiments. These tests are effective due to their translational nature and standardized data outputs. While touchscreen tests have standardized data outputs, there does not exist an automated process of easily cleaning raw touchscreen data. The data output file can easily contain thousands of rows and columns, which may take researchers weeks to months to completely analyze. Not only does manually analyzing raw touchscreen data take an unnecessary amount of time and resources, but also it will inevitably produce unwanted human errors. **This research explored the question of can Python be used to simplify the touchscreen data analysis process for ABET data**. We utilized Pandas, a software library built on top of Python used specifically for data analysis, as the foundation for the project. The DataFrame data structure and other useful functions were also used to extract important parameters from raw data and import them into a presentable comma separated values (CSV) file. The Tkinter package was used to construct a basic graphical user interface with a main menu, multiple sub-menu pages, easy-to-use buttons, and interchangeable criteria materials. As a result, the application significantly simplified and automated the touchscreen data analysis process for ABET touchscreen data. All the user has to do is click a button based on their desired functionality, locate the directory of the raw touchscreen data, and save the newly created CSV file in another directory. The newly created CSV file contains important parameters such as the percent correct, session duration, and number of trials for the first and second reversal. Additional parameters include the average latencies of reward collections, correct touches, and incorrect touches. The extracted CSV file can then be used to assess performance of tasks using graphing programmings such as Prism. The creation of a touchscreen data analysis application can transform the efficiency of data analysis within the touchscreen community. The data analysis application can be used across multiple ABET touchscreen tests such as General Touchscreen, Location Discrimination Reversal, Acquisition, and Extinction. Members of the touchscreen community who may not have programming experience can pilot the application and change the criterias to fit their own research. This data analysis application is available on my GitHub (raymon-shi) and may be downloaded and shared to members of the touchscreen community for their personal use.

The touchscreen (TS) operant chamber used to perform tests that measure rodent cognitive performance, such as pattern separation and cognitive flexibility.



Lafayette Instrument: Model 80614

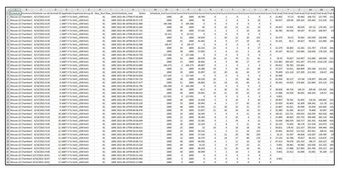
An example of TS test paradigms is the location discrimination reversal (LDR) test. Rodents are challenged to choose the correct lit window in order to receive a reward. After a set amount of trials, the position of the correct side is reversed.



TS tests are effective due to their translational nature and standardized data outputs.

Experimental rodent touchscreen results can be used to solve similar cognitive problems in humans.

Long experiments generate raw output data that contains thousands of rows and columns. Manually analyzing the data takes a tremendous amount of time (weeks/months) and resources.



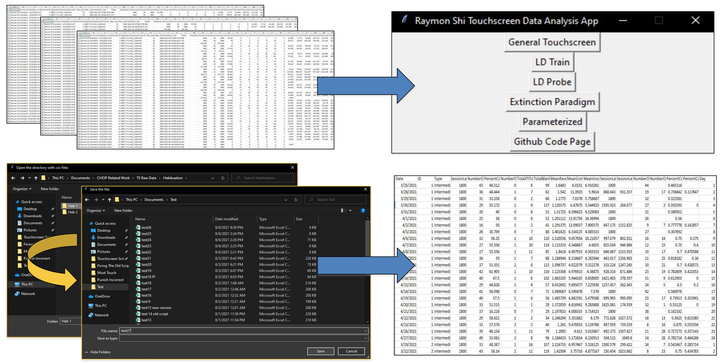
A screenshot of a small portion of raw touchscreen data.

Python is a high-level versatile scripting language used for web development, data analysis and visualization, machine learning, desktop graphic user interfaces (GUIs), and software development.

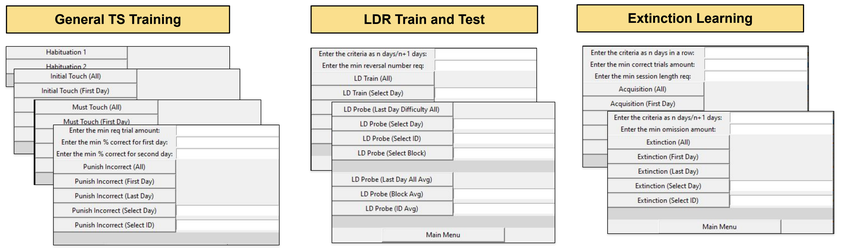
Pandas is an open source data analysis library available on the Python platform. It has many useful features such as the DataFrame data structure, the ability to read and write data from multiple sources, and the ability to easily manipulate data.

A Brand New Automated Workflow

Simply provide the application with raw touchscreen data and save the organized CSV file in a new directory!



Analyzes Several Different Touchscreen Test Paradigms



Specialized Data Sorting Based on Difficulty, Test Trial Day, and Animal ID

Date	ID	Type	Day	NumberO	PercentCc	NumberO
5/18/2021	1	easy	2	57	59.649	2
5/18/2021	2	hard	2	49	38.776	1
5/18/2021	3	hard	2	37	51.351	0
5/18/2021	4	easy	2	43	58.14	0
5/18/2021	5	easy	2	69	59.42	2
5/18/2021	6	hard	2	74	55.405	1
5/18/2021	7	hard	2	53	24.528	0
5/18/2021	8	easy	2	54	50	1
5/18/2021	9	easy	2	74	56.757	2
5/18/2021	10	hard	2	14	64.286	1
5/18/2021	11	hard	2	26	38.462	0
5/18/2021	12	easy	2	60	58.333	1
5/18/2021	13	easy	2	46	65.217	2
5/18/2021	14	hard	2	70	51.429	2
5/18/2021	15	hard	2	45	53.333	1
5/18/2021	16	easy	2	24	25	0
5/18/2021	17	easy	2	78	64.103	5
5/18/2021	18	hard	2	42	66.667	2
5/18/2021	19	hard	2	34	55.882	1
5/18/2021	20	easy	2	60	50	2
5/18/2021	21	easy	2	47	53.191	2
5/18/2021	22	hard	2	79	53.165	2
5/18/2021	23	hard	2	48	41.667	1
5/18/2021	24	easy	2	20	40	1
5/18/2021	25	easy	2	54	46.296	1
5/18/2021	26	hard	2	23	65.217	1
5/18/2021	27	hard	2	22	40.909	0
5/18/2021	28	easy	2	27	33.333	0
5/18/2021	29	easy	2	38	42.105	1
5/18/2021	30	hard	2	50	48	1

Sorting by difficulty separates the LDR Probe difficulties (easy/hard) and the LDR Train difficulty (intermediate).

Date	ID	Day	SessionLe	NumberO	PercentCc	NumberO
6/15/2021	1	10	1800	40	45	0
6/15/2021	2	10	1800	45	42.222	2
6/15/2021	3	10	1800	21	14.286	0
6/15/2021	4	10	1800	40	42.5	0
6/15/2021	5	10	1800	53	50.943	2
6/15/2021	6	10	1800	69	40.58	1
6/15/2021	7	10	1800	55	49.091	1
6/15/2021	8	10	1800	44	54.545	2
6/15/2021	9	10	1800	68	52.941	1
6/15/2021	10	10	1800	21	52.381	1
6/15/2021	11	10	1800	28	57.143	1
6/15/2021	12	10	1800	49	46.939	1
6/15/2021	13	10	1800	39	41.026	2
6/15/2021	14	10	1800	54	42.593	1
6/15/2021	15	10	1800	36	50	1
6/15/2021	16	10	1800	25	48	1
6/15/2021	18	10	1800	45	37.778	1
6/15/2021	19	10	1800	34	52.941	2
6/15/2021	20	10	1800	57	52.632	3
6/15/2021	21	10	1800	27	29.63	1
6/15/2021	22	10	1800	79	56.962	2
6/15/2021	23	10	1800	53	52.83	2
6/15/2021	24	10	1800	25	56	0
6/15/2021	25	10	1800	45	57.778	3
6/15/2021	26	10	1800	10	10	0
6/15/2021	27	10	1800	43	62.791	1
6/15/2021	28	10	1800	24	37.5	0
6/15/2021	29	10	1800	25	32	0
6/15/2021	30	10	1800	56	42.857	1

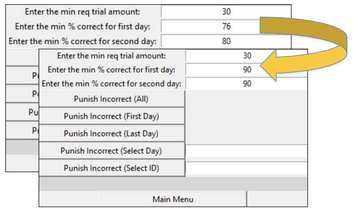
Sorting by day will show all the animal data rows for the selected day.

Date	ID	Type	Day	NumberO	PercentCc	NumberO
5/17/2021	21	easy	1	57	57.895	4
5/18/2021	21	easy	2	47	53.191	2
5/19/2021	21	hard	3	27	33.333	0
5/20/2021	21	hard	4	32	59.375	0
5/21/2021	21	easy	5	51	60.784	3
5/24/2021	21	easy	6	65	55.385	2
5/25/2021	21	hard	7	35	54.286	1
5/26/2021	21	hard	8	43	34.884	1
6/14/2021	21	easy	9	68	61.765	4
6/15/2021	21	easy	10	27	29.63	1
6/16/2021	21	hard	11	34	58.824	0
6/17/2021	21	hard	12	34	35.294	1
6/18/2021	21	easy	13	44	45.455	2
6/21/2021	21	easy	14	63	60.317	3
6/22/2021	21	hard	15	24	37.5	0
6/23/2021	21	hard	16	25	36	0
6/24/2021	21	easy	17	44	68.182	2
6/25/2021	21	easy	18	46	56.522	2
6/28/2021	21	hard	19	53	45.283	1
6/29/2021	21	hard	20	55	50.909	2
6/30/2021	21	easy	21	55	58.182	3
7/1/2021	21	easy	22	69	71.014	4
7/2/2021	21	easy	23	66	50	2
7/5/2021	21	hard	24	70	54.286	2
7/7/2021	21	easy	25	74	59.459	5
7/8/2021	21	easy	26	65	70.769	4
7/9/2021	21	hard	27	55	58.182	2
7/12/2021	21	hard	28	61	47.541	2
7/13/2021	21	easy	29	60	61.667	5
7/15/2021	21	easy	30	77	70.13	6

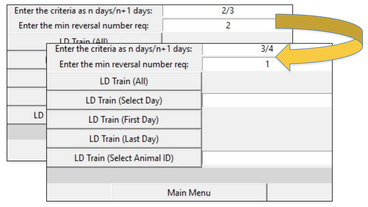
Sorting by animal ID will show all the data rows for the selected animal ID.

Interchangeable Test Criterias

Certain test paradigms such as Punish Incorrect and LD Train have the ability to change their test criterias.



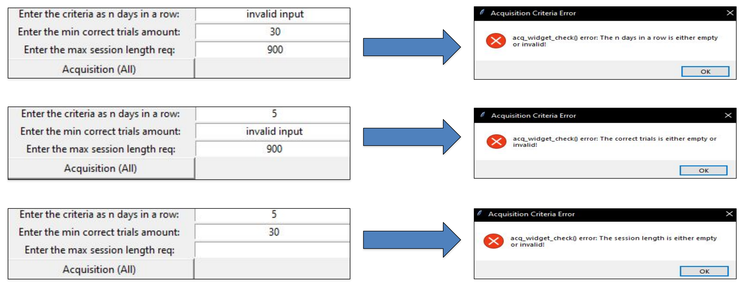
For Punish Incorrect, the minimum trials, percent correctness each day are interchangable.



For LD Train, the set of required days out of total days and the reversal number required are interchangeable.

Explicit User Input Error-Handling

If the user inputs an invalid criteria, an error message will indicate to explain the erroneous input.



Conclusions/Future Directions

- A formal automated process for cleaning touchscreen data can significantly improve productivity within the touchscreen community.
- The creation of a manuscript to introduce the automated process with the rest of the touchscreen community.
- Sharing the application and code with the open-science touchscreen data platform (i.e. MouseBytes)
- The code is open-source and available for anyone on my personal GitHub (raymon-shi)

ACKNOWLEDGEMENTS/REFERENCES

We acknowledge the support of former and current members of the Eisch Lab. SY was supported by an NIH Institutional Training Grant (MH076690, PI: CA Tamminga), NNX15AE09G (PI AJE), MH107945 (PI AJE), a 2018 PENN McCabe Pilot grant, a 2019 IBRO travel grant, and is currently supported by 2019 NARSAD Young Investigator Grant from the Brain and Behavior Research Foundation, 2020 a PENN Undergraduate Research Foundation grant, and a 2021 NASA HERO grant (80NSSC21K0814).

References:

- Palmer, Daniel, et al. "Touchscreen cognitive testing: Cross-species translation and co-clinical trials in neurodegenerative and neuropsychiatric disease." *Neurobiology of Learning and Memory* 182 (2021): 107443.
- Oomen, C., Hvoslef-Eide, M., Heath, C. *et al.* The touchscreen operant platform for testing working memory and pattern separation in rats and mice. *Nat Protoc* **8,** 2006–2021 (2013). https://doi.org/10.1038/nprot.2013.124
- Beraldo, Flavio H, et al. "MouseBytes, an Open-Access High-Throughput Pipeline and Database for Rodent Touchscreen-Based Cognitive Assessment." *ELife*, vol. 8, 2019, doi:10.7554/elife.49630.
- Lafayette Instruments, . *80614/LD - Location Discrimination (LD) Task for Mouse Touch Screen Systems and ABET II (2 x 6 Mask Version)*. 2011.