

Loglime: A Workflow for for Collecting and Processing Log-data

Tomasz Żółtak

Institute of Philosophy and Sociology, Polish Academy of Sciences

UseR! 2026, 06.07.2026

- Widespread use of computer testing in large scale assessments in education (LSAEs)
 - Interactive items emulating real-world tasks (science experiments, IT tasks, etc.)
- Growing importance of self-report modes of survey collection in the developed countries
 - European Social Survey announced to use primarily web and postal questionnaires from round 13 (year 2027) on
- Popularization of web-based interfaces in psychological studies

Limitations of the previous solutions

- Heerwegh (2003), Kaczmirek (2009), Stieger and Reips (2010):
 - JavaScript applets enabling to detect respondent's behavior (clicks, cursor moves);
 - Difficult implementation, requiring basic knowledge of web page development and deployment
 - No tools enabling further processing of the collected paradata
- Schlosser (2016), Schlosser & Höhne (2020):
 - Applet integrated with EFS-Survey/Unipark on-line survey platform makes it easy to design and deploy a survey
 - No tools enabling further processing of the collected paradata
 - No information about hovering

Limitations of the previous solutions (cont.)

- PIAAC Log Data analyzer, (Goldhammer et al., 2020), LOGAN (Reis Costa & Leoncio, 2019) and ProcData (Tang et al., 2021) R packages:
 - Designed to work with a specific study/studies (PIAAC, PISA)
 - Focused on predicting test-takers performance in cognitive tasks
- OpenSesame/OSWeb environment (Mathôt et al., 2012) accompanied by the *mousetrap* R package; Mathur and Reichling (2019) Qualtrics template and R scripts:
 - Designed specifically for the *category-competition* type of experiments

Limitations of the previous solutions (cont.)

- Kroehne's LogFSM (Kroehne & Goldhammer, 2018):
 - Very powerful framework
 - But one needs to have paradata collected and pre-processed to use it
 - Defining different *states* and matrix of transitions between *states* given respondent's action may be challenging

logLime's design

- Covers the entire process starting with paradata collection, through pre-processing to computing most popular process indicators
 - *logdataLimeSurvey* JavaScript applet - paradata collection
 - Log-data streams stored as responses to (hidden) text questions
 - *logLime* R package - preprocessing paradata, computing indicators
- Integration with (quite) popular on-line survey platform to make survey design and deployment easy
 - Lime Survey:
 - Matured and quite popular platform
 - Easy integration of JavaScript applets using survey *themes*
 - Asynchronous data transfer
 - In its HTML code it provides names for almost all of the survey interface elements

Using *logdata*LimeSurvey

- 1 Modify (copy-paste) two files using the LimeSurvey *theme* manager
- 2 Set your survey to use a modified *theme*
- 3 Add an open-ended (*long free text*) question to each survey screen on which you want to collect log-data and assign it a CSS class *logdata_container*

Log-data will be stored in the same way as answers to the regular questions and they can be exported the same way

Limitations of *logdataLimeSurvey*

- Longer log-data collection may cause user interface lags if internet connection is slow
 - This also increases the risk of data corruption during the asynchronous writing it to the database
- Log-data length limit of 524,288 characters on each screen

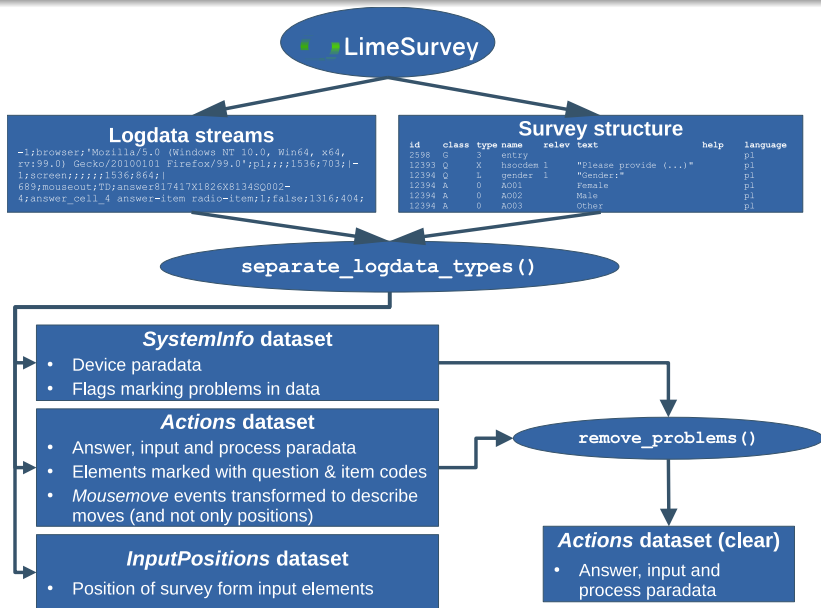
Ways to mitigate the problem:

- Setting time limit on log-data collection (by default 2 minutes)
- Limiting the frequency of collecting *mousemove* events (by default 100 milliseconds)
- Disabling the collection of some types of events
 - By commenting lines (that start appropriate event listeners) of the applet code using *LimeSurvey* theme manager

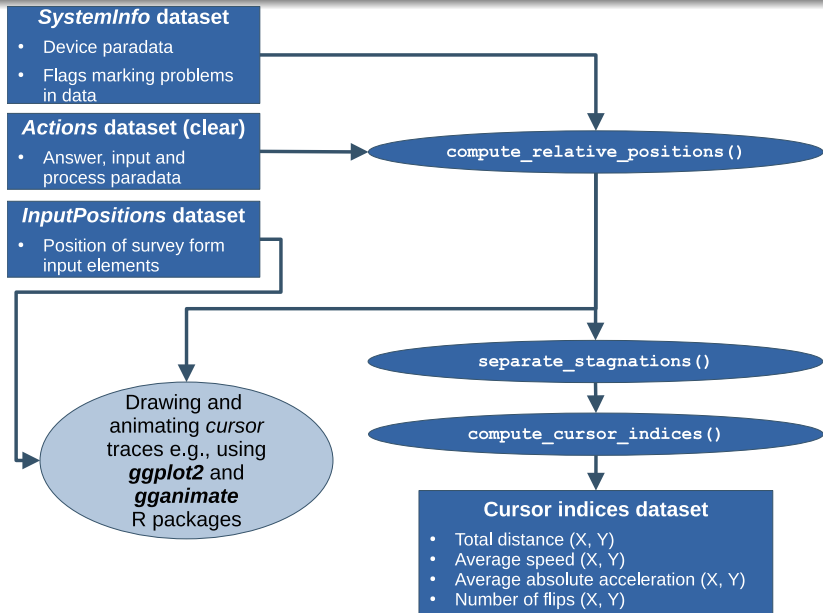
Using *logLime*

- ① Exporting data from *limeSurvey*
- ② Reading data into R
 - Solutions enabling fast reading big text files, like packages *arrow*, *data.table* or *vroom*, are very helpful
- ③ Paradata preprocessing
 - Transforming log-data to tabular form
 - Labelling user interface elements
 - Removing problematic observations
- ④ Deriving analytically useful data
 - Computing *process indicators*
 - Preparing data for visualization
 - Preparing data for analysis in other software

Paradata preprocessing



Processing cursor moves



Drawing heatmaps

SystemInfo dataset

- Device paradata
- Flags marking problems in data

Actions dataset (clear)

- Answer, input and process paradata

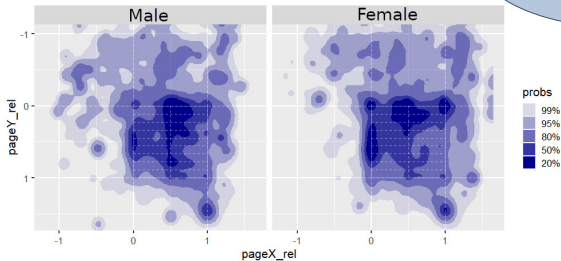
InputPositions dataset

- Position of survey form input elements

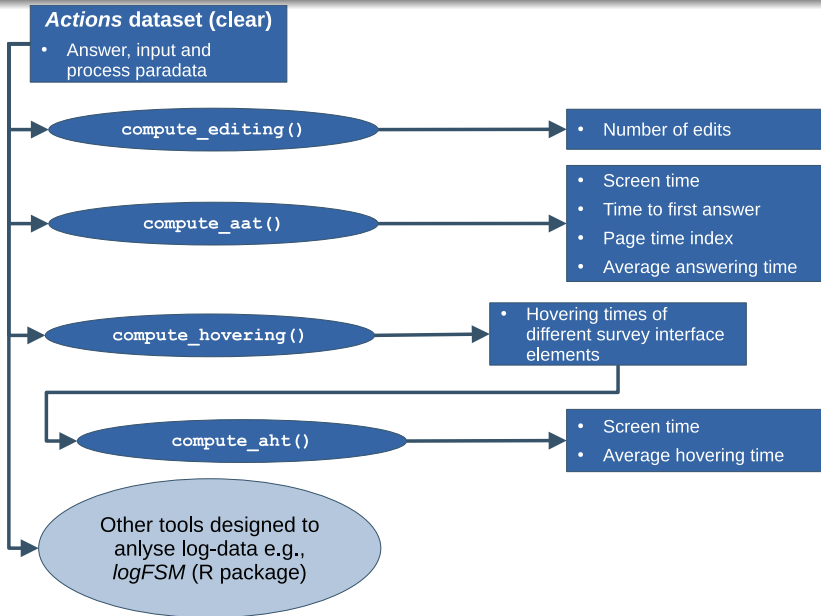
`compute_relative_positions()`

`compute_cursor_positions()`

Drawing heatmaps
e.g., using
ggplot2 and
ggdensity
R packages



Computing answering times and other indicators



Summary

- Presented software solution makes collecting and using paradata for analysis **simple and widely available**
- It is integrated with reliable and quite popular on-line survey platform *Lime Survey*
 - It can be potentially adapted to work with other platforms as long as they:
 - Support using survey admin. provided JavaScript applets
 - Name survey interface elements in its generated HTML code (using a reasonable convention)
- Enables to compute most widely used *process indicators*
 - It can be further developed to cover additional indicators
- Enables to prepare data for drawing plots
- Enables to prepare data for analysis with other software

Cited literature (1)

Goldhammer, F., Hahnel, C., & Kroehne, U. (2020). Analysing Log File Data from PIAAC. In D. B. Maehler & B. Rammstedt (Eds.), *Large-Scale Cognitive Assessment* (pp. 239–269). Springer International Publishing.

https://doi.org/10.1007/978-3-030-47515-4_10

Heerwegh, D. (2003). Explaining response latencies and changing answers using client-side paradata from a web survey. *Social Science Computer Review*, 21(3), 360–373.

Kaczmarek, L. (2009). Human-survey interaction: Usability and nonresponse in online surveys. von Halem.

Kroehne, U., & Goldhammer, F. (2018). How to conceptualize, represent, and analyze log data from technology-based assessments? A generic framework and an application to questionnaire items. *Behaviormetrika*, 45(2), 527–563.

<https://doi.org/10.1007/s41237-018-0063-y>

Mathôt, S., Schreij, D., & Theeuwes, J. (2012). OpenSesame: An open-source, graphical experiment builder for the social sciences. *Behavior Research Methods*, 44(2), 314–324. <https://doi.org/10.3758/s13428-011-0168-7>

Mathur, M. B., & Reichling, D. B. (2019). Open-source software for mouse-tracking in Qualtrics to measure category competition. *Behavior research methods*, 51(5), 1987–1997. <https://doi.org/10.3758/s13428-019-01258-6>

Cited literature (2)

Reis, C. D., & Leoncio, W. (2019). *LOGAN: An R package for log file analysis in international large-scale assessments*. <https://cran.r-project.org/package=LOGAN>

Stieger, S., & Reips, U.-D. (2010). What are participants doing while filling in an online questionnaire: A paradata collection tool and an empirical study. *Computers in Human Behavior*, 26(6), 1488–1495.

Schlosser, S. (2016). *Embedded Client Side Paradata (ECSP)*.
<https://doi.org/10.5281/zenodo.55169>

Schlosser, S., & Höhne J. K. (2020). *ECSP – Embedded Client Side Paradata*.
<https://doi.org/10.5281/zenodo.3782592>

Tang, X., Zhang, S., Wang, Z., Liu, J., & Ying, Z. (2021). ProcData: An R Package for Process Data Analysis. *Psychometrika*, 86(4), 1058–1083.
<https://doi.org/10.1007/s11336-021-09798-7>

Thank you for your attention!

tomasz.zoltak@ifispan.edu.pl

Understanding response styles in self-report data: consequences, remedies and sources [NCN/Opus/17/HS6/00937]



NATIONAL SCIENCE CENTRE
POLAND