A Reproducible Computational Pipeline for Modelling Sequential Decision Thresholds from Stopping-Rule Data

Alex Kolesnikov
Eton College
Windsor, Berkshire, United Kingdom
alexkolesnikov2010@gmail.com

Abstract

Stopping-rule experiments ask participants to move through ordered stages until they decide to act. These designs are useful for studying intervention thresholds, but raw survey exports are usually stored in wide form, while correct modelling requires scenario-level thresholds and stage-level at-risk rows. This paper presents a runnable R artifact for reconstructing and validating stopping-rule data. The artifact uses a JSON configuration layer, modular R source files, toy data, a second structurally different toy configuration, automated testthat tests, adversarial validation checks, reference outputs, licensing and citation metadata. In a de-identified sequential risk-decision case study, the same reconstruction logic maps 1,108 potential scenario observations from 277 consenting participants, flags 25 ambiguous response sequences, retains 1,083 clean scenario-level observations, and generates 3,047 stage-level at-risk rows. The evaluation documents local execution, automated test coverage, end-to-end preprocessing behaviour, a second configuration demonstration, adversarial failure behaviour and expected linear scaling for larger survey exports. The contribution is a reproducible data-engineering system that bridges raw stopping-rule survey exports and standard event-history or repeated-measures modelling tools, while making exclusion decisions auditable rather than hidden in manual recoding.

Keywords

Stopping-Rule Data, Sequential Decision-Making, Computational Pipeline, R Software, Reproducibility, Data Validation, Data Engineering, Discrete-Time Hazard Models, Right-Censoring, Decision-Support Systems

ACM Reference Format:
Alex Kolesnikov. 2026. A Reproducible Computational Pipeline for Modelling Sequential Decision Thresholds from Stopping-Rule Data. In *Proceedings of International Journal of Secondary Computing and Applications Research (IJSCAR VOL. 3, ISSUE 3)*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.67149/yhjs2024.5/3s7gdh1q

IJSCAR VOL. 3, ISSUE 3

1 Introduction

Many applied decision problems are sequential rather than static. A person, monitoring system or AI assistant may wait while evidence remains ambiguous and act only when accumulating information crosses a threshold. This structure appears in symptom checkers, safety monitors, cybersecurity alerts, financial-risk systems and user-facing decision-support tools. In each setting, the important question is not only what information is present, but when the information becomes action-relevant.

Stopping-rule surveys provide a compact experimental design for studying this behaviour. Participants see a sequence of stages and choose either to keep waiting or to act. Once action is selected, the scenario stops. The first action point defines the decision threshold. If no action occurs by the final stage, the true threshold is right-censored: it is known only to exceed the final displayed level.

The computational difficulty is that common survey platforms export this data in wide form: one row per participant and separate columns for possible scenario-stage responses. Standard modelling tools expect long, validated event-history data. Between those two forms lies a non-trivial reconstruction problem. Researchers must map survey columns to scenario structures, enforce the stopping rule, identify ambiguous response sequences, reconstruct scenario-level thresholds and expand clean cases into stage-level at-risk rows.

This preprocessing step matters because errors are easy to hide. A manual wide-to-long conversion can accidentally count post-intervention stages as if the participant were still at risk, treat right-censored observations as observed final-stage interventions, or silently retain inconsistent response sequences. Such errors can change the estimated timing of action even when the later statistical model is correctly specified. The present artifact therefore treats preprocessing as part of the scientific method, not as clerical preparation before analysis.

This paper introduces a runnable R artifact that performs this transformation. The contribution is not a new statistical estimator. It is a reproducible preprocessing and modelling pipeline that produces auditable datasets ready for existing event-history, GEE or mixed-model tools.

2 Related Work and Tool Gap

Discrete-time event-history models represent event occurrence across ordered opportunities and are well established

in statistics and social-science methodology [1, 2]. Generalised estimating equations provide population-average inference for clustered binary data [3, 4]. Software ecosystems such as R survival, Python lifelines, geepack and statsmodels support related model fitting once data have already been transformed into the required long form [4–7].

However, these tools do not solve the prior data-engineering problem caused by stopping-rule survey exports. They do not validate whether a response sequence obeys a first-action stopping rule, decide which post-intervention stages should be excluded, generate exclusion logs or create both scenario-level and stage-level files from a wide survey export. The present artifact fills that preprocessing gap and is designed to complement, not replace, existing modelling packages.

The closer methodological neighbours are therefore not only survival-analysis libraries but also work on reproducible workflows, tidy data and research data organisation. Tidy-data principles emphasise that each variable, observation and observational unit should be represented consistently before analysis [12]. Good-enough practices in scientific computing similarly argue that project organisation, scripted processing, versioned outputs and clear documentation are necessary for reliable computational research [13]. Data-organisation guidance for spreadsheets highlights the risk of analysis errors when raw data structures are convenient for entry but not for computation [14]. The present pipeline applies these general principles to a specific experimental structure: first-action stopping-rule surveys.

General data-validation frameworks such as validate and pointblank support reusable column-level, row-level and cross-column rules, including checks on values, missingness and consistency [15, 16]. They can therefore identify many generic data-quality problems that fall outside the scope of this artifact. However, they do not provide built-in first-action stopping-rule semantics for reconstructing thresholds, distinguishing right-censoring from ambiguous sequences, or generating stage-level at-risk rows. The present artifact is narrower and domain-specific: it performs those stopping-rule transformations and returns threshold, censoring, exclusion-log and at-risk-row outputs.

The design also follows principles from reproducible computational research: code, intermediate data products, logs and tests should make a workflow inspectable rather than relying on undocumented manual recoding [8–11]. Its specific gap is the lack of a small, reusable bridge between survey exports and event-history-ready stopping-rule datasets.

3 System

The artifact contains modular R files, JSON configuration files, toy datasets, testthat tests, reference output files, adversarial validation checks and software metadata. It is intentionally small so that its logic can be inspected directly and adapted to new stopping-rule designs by editing configuration rather than rewriting code. The system is organised around three design choices: study-specific structure belongs in configuration; response-sequence validity is checked before modelling; and all transformations produce auditable output files.

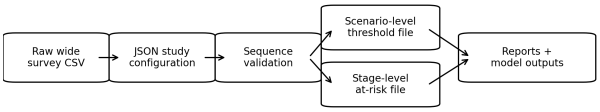

Figure 1: R pipeline from raw wide survey export to auditable model outputs.

3.1 Configuration-Driven Mapping

The primary file `config/study_config.json` specifies the participant identifier, response labels, covariates, scenario identifiers, condition labels, ordered response columns and stage scores. JSON was used because the mapping is structured, human-readable, language-independent and easy to inspect without running R. Keeping this mapping outside the R functions prevents study-specific column names from being hard-coded into the validation and expansion logic. A second file, `config/study_config_3scenario_7stage.json`, uses three scenarios, seven stages, different column names and different stage scores to demonstrate that the same reconstruction functions can be reused without source-code edits.

3.2 Sequence Validation

For each participant-scenario sequence, `validate_sequences.R` classifies the observation as a clean intervention, clean right-censored non-intervention, or ambiguous sequence. A clean intervention contains wait responses before the first action and no later responses after the action. A clean censored sequence contains wait responses through the final observed stage. A sequence is ambiguous if it contains unknown labels, non-wait responses before action, responses after action, missing values inside an otherwise observed sequence, or malformed stage information. This taxonomy separates errors that are analytically meaningful from ordinary right-censoring. The goal is to fail loudly or log exclusions rather than silently manufacture model-ready rows from invalid input.

3.3 Scenario and Stage Expansion

Clean sequences are reconstructed into scenario-level rows containing the intervention indicator, action stage, action score and censoring status. `expand_stage_level.R` then creates one row per at-risk decision opportunity. If action occurs at stage 3, only stages 1 to 3 are included; stages 4 and 5 are structurally excluded because the participant is no longer at risk. If no action occurs, all observed stages are retained with event = 0. This distinction is essential because event-history models estimate the conditional probability of acting at a stage given that action has not already occurred.

Table 1: Scope relative to existing modelling and workflow tools.

Tool or approach	Primary function	Remaining gap for stopping-rule surveys
survival / lifelines	Event-history modelling	Requires long-form event data already reconstructed
geepack / statsmodels GEE	Clustered binary modelling	Requires at-risk rows and event indicators
General workflow tools	Project organisation and reproducibility	Do not encode first-action stopping-rule logic
Spreadsheet or survey exports	Store raw wide responses	Do not directly represent scenario thresholds or right-censoring
This R pipeline	Validation and reconstruction	Produces scenario-level and stage-level model-ready files with exclusion logs

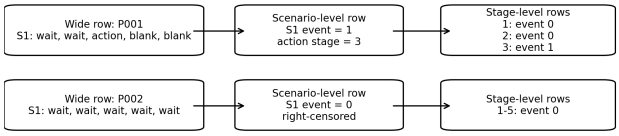

Figure 2: Conversion from wide stopping-rule responses to scenario-level and stage-level data.

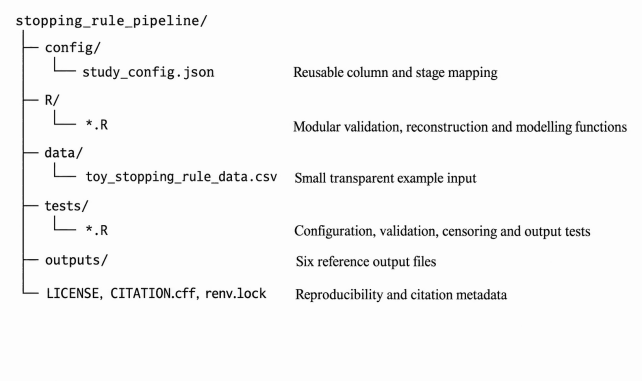

Figure 3: Directory layout of the R software artifact supplied with the paper.

3.4 Outputs and Packaging

The output module writes six files for each configured run: scenario-level data, stage-level data, an exclusions log, a reconstruction summary, a runtime report and model coefficients. The repository also includes LICENSE, CITATION.cff, .gitignore and renv.lock files so that the artifact behaves like research software rather than loose analysis scripts. Table 2 summarises the artifact structure in accessible form, and Figure 3 shows the corresponding directory layout.

4 Case Study

The pipeline is demonstrated using a de-identified sequential risk-decision dataset collected by the author in April and May 2026 during a broader empirical study of intervention thresholds in adolescents and adults. The original survey asked participants to move through hypothetical health-risk scenarios and choose either to continue monitoring or seek help. Participants were recruited through a convenience sample of UK secondary-school pupils aged approximately 15–18, recorded in age categories, and adult contacts, including parents and teachers. The school-age sample included participants from several independent and state secondary schools. The survey was administered online through Google Forms. The substantive empirical analysis of the Part B dataset has been published separately [17]. The case study in the present paper uses the reconstruction problem from that dataset; it does not reuse the published article's hypothesis tests as the contribution of this article.

As an independent, school-supervised project rather than university or clinical research, the study was not submitted to a university or external research ethics committee. Survey distribution and data collection involving school-age participants were reviewed and approved before distribution by the Deputy Head Pastoral and Designated Safeguarding Lead at the participating schools. Participation was voluntary, and only submissions with electronic consent were included in the analysis; two submissions without consent were excluded. The project was considered low risk because it used hypothetical scenarios, involved no intervention or medical procedure, collected no clinical or diagnostic data, and allowed participants to stop at any time. Responses were collected anonymously: no names, contact details, email addresses, IP addresses, school identifiers, personal medical histories or clinical records were included in the research dataset. Demographic information was retained only in broad categories. After collection, the twelve survey-version exports were combined into a de-identified analytical dataset. Original timestamps, submission order, row numbers, consent responses and potentially identifying metadata were removed. Each consenting participant was assigned a random anonymous identifier that did not encode survey version, timestamp or original row order, and the combined dataset was shuffled before sharing.

Table 2: Structure of the R software artifact supplied with the paper.

Artifact component	Purpose
R/	Source functions for configuration loading, validation, reconstruction, stage-level expansion, optional modelling and exporting
config/	Primary and second JSON mappings for participant identifiers, response labels, scenario-stage columns and stage scores
data/	Primary and second toy wide-format stopping-rule inputs used for demonstration and tests
tests/	Automated checks for configuration loading, validation, expansion, output-file generation, second-configuration reuse and assumption violations
outputs/	Reference outputs, second-configuration summary and adversarial stress-test matrix
local_records/	Local verification records, R session information, console logs and output-file lists
LICENSE, CITATION.cff, renv.lock	Software licensing, citation metadata and dependency record

Table 3: Case-study reconstruction counts.

Quantity	Count
Consenting participants	277
Potential scenario observations	1,108
Ambiguous sequences flagged	25
Clean scenario-level observations	1,083
Stage-level at-risk rows	3,047

The case-study dataset is used here only to demonstrate the computational reconstruction pipeline. The present article does not make substantive behavioural claims about age, escalation trajectory or information format. Its contribution is the software artifact and the auditable conversion from wide stopping-rule survey exports to scenario-level and stage-level modelling datasets.

The reconstruction logic maps 1,108 potential Part B scenario observations from 277 consenting participants. It flags 25 ambiguous response sequences, retains 1,083 clean scenario-level observations and expands these into 3,047 stage-level at-risk decision rows. These counts show the difference between raw possible observations, valid threshold observations and model-ready at-risk rows.

5 Evaluation

The artifact is evaluated as a computational reconstruction system rather than as a new estimator. The evaluation asks whether the system has a concrete accessible software object, whether its mapping is configuration-driven, whether output files match the paper claims, whether the automated tests cover the main intended behaviours and whether the resulting data products reduce the risk of common manual preprocessing errors.

5.1 Local Verification and Public Artifact

The R artifact is available in a dedicated project OSF repository: https://osf.io/gyvmz/. The repository contains the submitted software artifact and verification materials, including modular R source code, JSON configurations, toy wide-format stopping-rule datasets, automated testthat files, six reference outputs, a second configuration demonstration, adversarial stress-test materials, software metadata and local verification records. The final primary artifact was tested locally in RStudio on R version 4.6.0. The expanded testthat suite completed successfully with 41 passes, 0 failures, 0 warnings and 0 skipped tests across test groups covering configuration loading, output-file generation, sequence validation, stage-level expansion, second-configuration reuse and assumption violations. The demonstration pipeline also completed successfully, generated all six expected output files, and recorded a runtime of 0.1449 seconds on the primary toy dataset.

Because the toy demonstration dataset is intentionally small, the model-fitting module first attempts the full GEE specification and then falls back to prespecified simpler formulas if the full model matrix is rank deficient. In the local verification run, the score-plus-trajectory exchangeable GEE model was used successfully and estimates were exported to `gee_coefficients.csv`. This fallback behaviour is part of the optional modelling wrapper rather than a change to the empirical reconstruction logic.

5.2 Reusability Through Configuration

The artifact now demonstrates reusability through a second structurally different toy configuration. The primary toy design uses two scenarios and five stages; the additional configuration uses three scenarios and seven stages, with a different participant identifier, different response columns and different stage scores. The same validation, reconstruction and stage-expansion functions are used for both designs. The second configuration test verifies that no source-function changes are required and reconstructs 9 clean scenario observations, 3 logged ambiguous observations and 42 stage-level at-risk rows from the alternative toy export. This does not prove universal generality, but it converts the reusability claim from an architectural argument into a checked example of configuration-driven adaptation.

5.3 Automated Checks and Failure Behaviour

The evaluation distinguishes successful intended behaviour from the failure mode that matters most in stopping-rule preprocessing: silent generation of apparently valid stage-level rows from invalid response sequences. The included tests cover configuration loading, presence of the six reference output files, recognition of clean intervention sequences, recognition of clean right-censored sequences, classification of action-followed-by-later-response sequences as ambiguous, exclusion of ambiguous observations from the reconstructed scenario-level dataset, stopping stage-level expansion after action, and retention of all observed stages for right-censored scenarios. The revised artifact also includes adversarial checks for a missing configured stage column, an unknown response label, a response after action, a missing value inside an otherwise observed sequence, a malformed export missing the participant identifier, and a zero-byte input file. These cases document whether the pipeline fails loudly, logs an exclusion or completes safely rather than silently producing wrong model-ready rows.

5.4 End-to-End Comparison With Manual Preprocessing

A manual preprocessing workflow for the toy data requires a researcher to inspect each scenario sequence, identify the first action, decide whether later responses should be ignored or treated as invalid, create a scenario-level threshold row, expand the sequence into at-risk rows and maintain a separate exclusion record. The pipeline makes those decisions explicit and repeatable. In the demonstration run, the software generated the scenario-level file, stage-level file and exclusion log from the same configured rules. The practical advantage is qualitative rather than a claim of statistical superiority: it reduces researcher discretion, makes exclusions inspectable, and prevents common event-history mistakes such as retaining post-action stages or assigning final-stage scores to right-censored non-interventions.

5.5 Scalability

The reconstruction algorithm is linear in the number of participant-scenario-stage cells that must be inspected. If P is the number of participants, S the number of scenarios per participant and K the maximum number of stages, the validation step is $O(P{\cdot}S{\cdot}K)$, and the stage-level expansion produces at most $P{\cdot}S{\cdot}K$ rows. Memory use is also dominated by the resulting long-form stage-level table. This scaling is appropriate for typical survey exports because K is usually small and fixed by design. The reported 0.1449-second runtime should therefore be read only as a local toy-data verification result, not as a benchmark for all future uses. For larger studies, the main practical limit is expected to be the size of the expanded stage-level file rather than the sequence-validation loop.

6 Discussion

The main contribution is the separation of stopping-rule reconstruction from model fitting. Existing modelling packages are powerful after the analyst has created the correct long-form data structure. The fragile step occurs earlier, when wide survey responses must be interpreted as thresholds, censoring decisions and at-risk rows. By making that step scripted, configured and logged, the artifact turns an often invisible preprocessing stage into an inspectable computational object.

The configuration-driven approach is useful because stopping-rule studies may vary in the number of scenarios, number of stages, stage scores and column names while sharing the same first-action logic. Separating these design features from the R functions makes adaptation easier and makes errors more visible: if a configured column is absent or a response label is unknown, this should be treated as a data-integrity problem rather than silently resolved by analyst judgement.

The artifact is deliberately modest. It is not a replacement for survival, geepack, statsmodels or lifelines. Its purpose is to produce the data structures those tools require. Its novelty is therefore practical rather than mathematical: it packages validation, reconstruction, exclusion logging and stage-level expansion for a class of sequential survey designs that otherwise invite ad hoc recoding.

7 Limitations and Future Work

The artifact currently assumes ordered stages and a first-action stopping rule. Designs that allow repeated actions, reversible decisions, competing event types or multiple possible interventions would require additional state logic. For example, a study in which participants can first choose to call a friend, later call a doctor and later call emergency services would need a competing-risk or multi-state extension rather than the single first-action rule used here.

The case-study dataset has also been used in a published empirical analysis of information-format effects on intervention thresholds in the same sequential health-decision task [17]. That article examines substantive behavioural associations between risk-information format and intervention timing, whereas the present article addresses the distinct computational problem of reproducible stopping-rule reconstruction, validation, exclusion logging and stage-level expansion. The dataset is therefore shared between the two articles, while the analyses, research questions and results are entirely separate.

The evaluation remains limited. The automated tests address the main intended behaviours and a small set of adversarial failure cases, but they do not prove robustness to every possible survey-platform export. Future work could turn the script-based artifact into a formal R package with continuous integration, vignettes, more bundled configuration templates and a public issue tracker. Larger simulated benchmarks would also be useful for measuring runtime and

Table 4: Automated checks and adversarial stress-test behaviour.

Automated or adversarial check	Observed or required behaviour	Why it matters
Configuration loading	`study_config.json` loads and exposes scenario-stage mapping	Prevents analysis from proceeding without an explicit study map
Second configuration	3-scenario × 7-stage configuration uses same source functions and reconstructs 9 clean scenarios, 3 exclusions and 42 stage rows	Demonstrates configuration-driven reuse rather than only asserting it
Reference outputs	All six expected primary output files are present	Confirms that the runnable artifact produces the documented files
Clean intervention sequence	First action is recognised and action stage is reconstructed	Defines the observed decision threshold
Right-censored sequence	All-wait sequence is recognised as non-intervention rather than final-stage action	Preserves right-censoring rather than inventing an event
Response after action	Sequence is classified as ambiguous and logged	Prevents impossible post-action rows from entering the at-risk dataset
Missing value inside sequence	Sequence is classified as ambiguous with reason `missing_inside_sequence`	Prevents interpolation or silent compression of stages
Unknown response label	Sequence is classified as ambiguous with reason `unknown_response`	Prevents unrecognised survey labels from entering model-ready data
Missing configured stage column	Reconstruction fails loudly rather than guessing absent data	Protects against malformed survey exports
Malformed export missing participant identifier	Reconstruction fails loudly rather than producing model-ready output	Prevents structurally incomplete survey exports from being processed as valid data
Zero-byte input file	Reconstruction fails loudly with "no lines available in input" rather than producing model-ready output	Prevents an empty survey export from being treated as valid input
Malformed configuration	Configuration loading fails when required fields are absent	Prevents reconstruction without a valid study map
Stage expansion	Expansion stops after action and retains all stages for censored scenarios	Creates valid event-history rows

memory use across thousands of participants, many scenarios and longer stage sequences.

8 Conclusion

Stopping-rule designs are useful for studying sequential decision thresholds, but they create a specific data-engineering problem. Raw wide survey exports must be mapped, validated, reconstructed and expanded before they can be modelled correctly. This paper presents a runnable R artifact that performs this transformation using configuration-driven mapping, modular reconstruction functions and auditable outputs.

By turning stopping-rule responses into scenario-level thresholds and stage-level at-risk rows, the pipeline provides a bridge between behavioural survey exports and standard event-history or repeated-measures modelling tools. This makes the timing of action under uncertainty easier to analyse reproducibly in digital health, AI warning systems and other decision-support settings.

Data Availability

The complete R software artifact and reproducibility materials are publicly available through the dedicated OSF project at https://osf.io/gyvmz/ and via the persistent DOI https://doi.org/10.17605/OSF.IO/GYVMZ. The repository contains all computational materials supporting this manuscript, including the modular R source code, primary and secondary JSON configurations, toy datasets, automated tests, adversarial validation checks, reference outputs, software metadata and local verification records.

References

[1] Paul D. Allison. 1982. Discrete-time methods for the analysis of event histories. *Sociological Methodology* 13, 61–98.

Table 5: Software-artifact evaluation summary.

Evaluation item	Artifact status
Public accessibility	OSF link supplied for code, toy data, tests, outputs and local verification records
Configurable mapping	Study-specific columns, labels, scenario definitions and stage scores stored in JSON rather than hard-coded in reconstruction functions
Second configuration	Additional 3-scenario × 7-stage toy configuration demonstrates reuse with changed column names, participant identifier and stage scores
Adversarial checks	Stress matrix covers missing stage column, unknown response, post-action response, internal missing value, malformed export and zero-byte input
Core source modules	Configuration loading, validation, scenario reconstruction, stage-level expansion, optional modelling and exporting
Reference outputs	Six primary files exported in outputs/ and verified in local run records
Testing	testthat checks cover primary reconstruction logic, second configuration and assumption-violation behaviour
Existing-tool comparison	Pipeline fills preprocessing gap before survival/geepack/lifelines-style modelling
Scalability	Linear expected scaling in participant-scenario-stage cells; toy runtime reported only as local verification

[2] Judith D. Singer and John B. Willett. 2003. *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford University Press.

[3] Kung-Yee Liang and Scott L. Zeger. 1986. Longitudinal data analysis using generalized linear models. *Biometrika* 73, 1, 13–22.

[4] Ulrich Halekoh, Soren Hojsgaard and Jun Yan. 2006. The R package geepack for generalized estimating equations. *Journal of Statistical Software* 15, 2, 1–11.

[5] Terry M. Therneau. 2024. *A Package for Survival Analysis in R*. R package survival.

[6] Cameron Davidson-Pilon. 2019. lifelines: Survival analysis in Python. *Journal of Open Source Software* 4, 40, 1317.

[7] Statsmodels Developers. 2024. statsmodels: Statistical modelling and econometrics in Python.

[8] Roger D. Peng. 2011. Reproducible research in computational science. *Science* 334, 6060, 1226–1227.

[9] Victoria Stodden, Friedrich Leisch and Roger D. Peng, eds. 2014. *Implementing Reproducible Research*. CRC Press.

[10] Mark D. Wilkinson et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3, 160018.

[11] Karthik Ram. 2013. Git can facilitate greater reproducibility and increased transparency in science. *Source Code for Biology and Medicine* 8, 7.

[12] Hadley Wickham. 2014. Tidy data. *Journal of Statistical Software* 59, 10, 1–23.

[13] Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt and Tracy K. Teal. 2017. Good enough practices in scientific computing. *PLOS Computational Biology* 13, 6, e1005510.

[14] Karl W. Broman and Kara H. Woo. 2018. Data organization in spreadsheets. *The American Statistician* 72, 1, 2–10.

[15] Mark P. J. van der Loo and Edwin de Jonge. 2021. Data Validation Infrastructure for R. *Journal of Statistical Software* 97, 10, 1–31.

[16] Richard Iannone, Mauricio Vargas and June Choe. 2026. pointblank: Data Validation and Organization of Metadata for Local and Remote Tables. R package version 0.12.4. DOI: 10.32614/CRAN.package.pointblank.

[17] Alex Kolesnikov. 2026. A color-coded urgency bar and intervention thresholds in sequential health decisions. *Journal of High School Science* 10, 3, 394–413. https://doi.org/10.64336/001c.166227.