





Fluxtools: Reproducible QA/QC for Eddy Covariance Data Aligned with AmeriFlux Submission Standards

Kesondra Key ¹

¹ Paul H. O'Neill School of Public and Environmental Affairs, Indiana University, Bloomington, Indiana, USA 

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Open Journals](#) 

Reviewers:

- [@openjournals](#)

Submitted: 01 January 1970

Published: unpublished

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Eddy covariance data processing requires extensive quality control (QA/QC) to identify and remove implausible or erroneous half-hourly flux data before submission to public data repositories such as [AmeriFlux](#) ([AmeriFlux Management Project, 2025](#)). [Fluxtools](#) ([Key, 2025](#)) is an R ($\geq 4.5.0$; ([2025](#))) [Shiny](#) ([Chang et al., 2024](#)) application built with [Plotly](#) ([Sievert et al., 2024](#)) and [dplyr](#) ([Wickham et al., 2023](#)) packages designed to streamline this workflow by providing interactive visualization, year-based filtering, and on-the-fly R code generation for specified data removal. Users can visually flag anomalous data points (i.e., periods of sensor failure, physically implausible data), accumulate multiple cleaning steps, inspect R^2 values before and after data cleaning via base R's [lm\(\)](#) function, and export a zipped folder containing a cleaned .csv file and a full R script that records every decision. [Fluxtools](#) significantly accelerates the QA/QC workflow, ensuring transparent, reproducible, and shareable data cleaning suitable for final dataset preparation and repository submission.

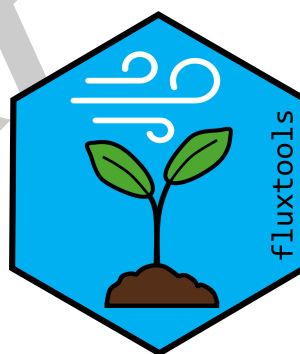


Figure 1: *Fluxtools* hex logo

Key features:

- **Interactive Plotly Scatterplots:** Plot any numeric or time variable; hover mouse over data points to see timestamps and values; export plots as .png directly from the app
- **Flexible point selection:** Box-select, lasso, or apply standard-deviation (σ) cutoffs to mark selected points. *Fluxtools* automatically generates removal ready-to-copy R snippets (`dplyr::case_when(... ~ NA)`) in the *Current* code pane. See *Fig 2* for interface and data selection example
- **On-the-fly R code generation:** After point selection, clicking *Flag Data* automatically highlights chosen points in yellow and appends the corresponding removal code into the *Accumulated* code panel for easy and continuous data selection

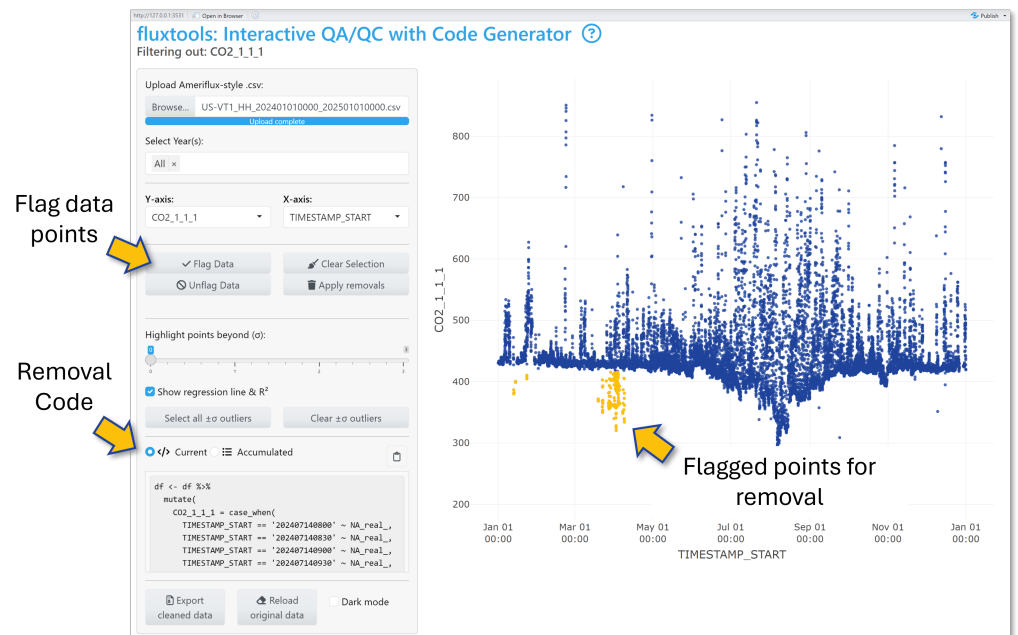


Figure 2: Example of the *Fluxtools* interface and data selection

- Before/after R² diagnostics: For any numeric variable comparison, *Fluxtools* fits a linear regression model and reports its R² value. Selecting points re-computes R² as if those points were removed, allowing for easy comparison. Fig 3 shows this process in *Fluxtools* using the $\pm\sigma$ outliers selection tool: The top (red) R² uses all data, while the bottom R² (orange) omits selected points from the linear regression.

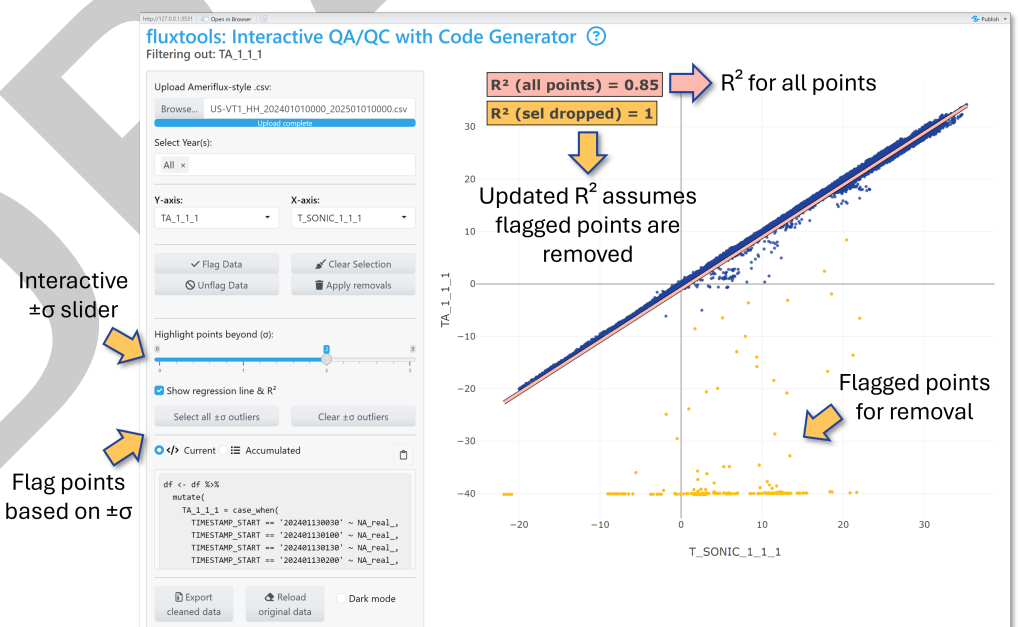


Figure 3: Example of R² diagnostics using the $\pm\sigma$ outliers cutoffs and selection

- Export cleaned .csv file and R script: Apply *removals* in-app (converting data points into NAs for selected timestamps) then *Export cleaned data* to download a cleaned .csv file and a comprehensive R script documenting each data removal step

Statement of need

High-frequency (10 Hz; data recorded 10 times per second) eddy covariance measurements generate large datasets that must be aggregated into half-hourly fluxes, using careful quality assurance and quality control (Burba, 2021). At this high frequency, intermittent periods of sensor drift or failure are common, making manual data cleaning an integral part of the workflow. Tools like *EddyPro* (LI-COR Biosciences, 2021) convert raw 10Hz data into half-hourly fluxes, while R packages like *REddyProc* (Wutzler et al., 2024), and Python tools like *PyFluxPro* (Isaac, 2021), automate u^* -threshold filtering, gap-filling, and flux partitioning. These tools excel at bulk data processing but offer no interactive means to inspect or carefully remove outliers that require a human eye.

In practice, data managers resort to custom scripts, extensive manual visualization, and fragmented documentation to detect and remove erroneous data points caused by sensor drift, malfunction, or calibration issues. These procedures are labor-intensive, prone to errors, challenging to reproduce, and lack transparency. *Fluxtools* addresses this challenge by pairing an interactive scatterplot-based interface with on-the-fly R code generation. Users can visually flag implausible half-hourly data points, automatically generate the exact `case_when(... ~ NA) dplyr` code snippets (or apply removals automatically in the app) and export a .zip file containing a cleaned .csv file plus a comprehensive R script documenting each data removal step that captures every user-made QA/QC decision.

Fluxtools streamlines and clarifies this QA/QC workflow by combining interactivity with code-based reproducibility. It promotes transparent documentation of decisions, reduces manual effort, and accelerates the preparation of flux data for repository uploads such as individual site submissions to AmeriFlux. Ultimately, *Fluxtools* lowers the barriers to robust and reproducible QA/QC workflows, enabling researchers to devote less time to manual anomaly detection and more time to scientific analysis.

Code Example

Fluxtools can be installed from [Github](#).

Installation

```
# Option 1.) Install from CRAN (Fluxtools R Package submission pending)
#install.packages("fluxtools")
```

```
# Option 2.) Install most recent version from GitHub
library(remotes)
remotes::install_git("https://github.com/kesondrakey/fluxtools.git")
```

Call the Fluxtools App

```
library(fluxtools)

#Set your site's UTC offset (e.g., -5 for Eastern Standard Time)
fluxtools::run_flux_qaqc(-5)
```

Acknowledgments

Fluxtools is an independent project developed in my role as Co-Principal Investigator for the AmeriFlux US-VT1 (Key & Novick, 2025a) and US-VT2 (Key & Novick, 2025b) sites to streamline our QA/QC pipeline. I thank AmeriFlux for maintaining clear, up-to-date

70 data-formatting specifications (see [AmeriFlux Data Variables](#)) and I acknowledge the broader
71 *R* and *Shiny* communities for their efforts in advancing interactive data-visualization tools.

72 I am grateful to Dr. Kim Novick (Indiana University), Housen Chu (AmeriFlux), and Benju
73 Baniya (Texas A&M University) for their invaluable expertise in assisting with the AmeriFlux
74 data processing workflow. Without their support, this data pipeline would not have been
75 possible. I also thank Dr. Mallory Barnes, Dr. Daniel Beverly, and Dr. Xian Wang (all at
76 Indiana University) for their support during development and testing. Special thanks goes to
77 George Burba (LI-COR Biosciences) for his encouragement and for insightful discussions that
78 have helped shape the future of *Fluxtools*.

79 This work was supported by NSF's Division of Environmental Biology (DEB) Dynamics of
80 Socio-Environmental Systems Program (Award 22-06086) and NASA's Carbon Monitoring
81 System (Award 80NSSC23K1254).

82 Partial development of the *Fluxtools* application logic, debugging assistance, and initial code
83 drafting were supported by OpenAI's *GPT-4-mini-high* large language model ([OpenAI, 2025](#)).
84 All final code and revisions were authored and approved by the human author.

85 References

- 86 AmeriFlux Management Project. (2025). *AmeriFlux data portal*. <https://ameriflux.lbl.gov/>
- 87 Burba, G. (2021). *Eddy covariance method - for scientific, regulatory, and commercial*
88 *applications* (p. 702). LI-COR Biosciences. ISBN: 978-0-578-97714-0
- 89 Chang, W., Cheng, J., Allaire, J., Sievert, C., Schloerke, B., Xie, Y., Allen, J., McPherson,
90 J., Dipert, A., & Borges, B. (2024). *Shiny: Web application framework for r*. <https://doi.org/10.32614/CRAN.package.shiny>
- 91 Isaac, P. (2021). *PyFluxPro*. <https://github.com/OzFlux/PyFluxPro>
- 92 Key, K. (2025). *Fluxtools: Interactive QA/QC for AmeriFlux data* (Version 0.3.0). Zenodo.
93 <https://doi.org/10.5281/zenodo.15626645>
- 94 Key, K., & Novick, K. (2025a). *AmeriFlux BASE US-VT1 vermilion tributary paired cropland –*
95 *site 1 (corn/soy; no cover crops)* (1–5 (BASE, CC-BY-4.0)) [Data set]. AmeriFlux Network,
96 Lawrence Berkeley National Laboratory. <https://doi.org/10.17190/AMF/2567994>
- 97 Key, K., & Novick, K. (2025b). *AmeriFlux BASE US-VT2 vermilion tributary paired cropland*
98 *– site 2 (corn/soy; cover crops)* (1–5 (BASE, CC-BY-4.0)) [Data set]. AmeriFlux Network,
99 Lawrence Berkeley National Laboratory. <https://doi.org/10.17190/AMF/2567995>
- 100 LI-COR Biosciences. (2021). *EddyCovariance Processing Software (EddyPro)* (Version 7.0.9).
101 <https://www.licor.com/env/products/eddy-covariance/eddypro>
- 102 OpenAI. (2025). *Introducing OpenAI o3 and o4-mini*. <https://openai.com/index/introducing-o3-and-o4-mini/>
- 103 R Core Team. (2025). *R: A language and environment for statistical computing*. R Foundation
104 for Statistical Computing. <https://www.R-project.org/>
- 105 Sievert, C., Parmer, C., Hocking, T., Chamberlain, S., Ram, K., Corvellec, M., Despouy, P.,
106 Brüggemann, S., & Inc, P. T. (2024). *Plotly: Create interactive web graphics via 'plotly.js'*.
107 <https://doi.org/10.32614/CRAN.package.plotly>
- 108 Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar*
109 *of data manipulation*. <https://doi.org/10.32614/CRAN.package.dplyr>
- 110 Wutzler, T., Reichstein, M., Moffat, A. M., & Migliavacca, M. (2024). *REddyProc: Post*
111 *processing of (half-)hourly eddy-covariance measurements*. <https://doi.org/10.32614/CRAN.package.REddyProc>