

Implementing GitHub Actions continuous integration to reduce error rates in ecological data collection

Albert Y. Kim^{1,2}  | Valentine Herrmann²  | Ross Barreto³ | Brianna Calkins² | Erika Gonzalez-Akre²  | Daniel J. Johnson³  | Jennifer A. Jordan²  | Lukas Magee³  | Ian R. McGregor⁴  | Nicolle Montero³ | Karl Novak³ | Teagan Rogers²  | Jessica Shue⁵  | Kristina J. Anderson-Teixeira^{2,6} 

¹Program in Statistical and Data Sciences; Smith College, Northampton, Massachusetts, USA; ²Conservation Ecology Center, Smithsonian National Zoo & Conservation Biology Institute, Front Royal, Virginia, USA; ³School of Forest, Fisheries, & Geomatics Sciences, University of Florida, Gainesville, Florida, USA; ⁴Center for Geospatial Analytics, North Carolina State University, Raleigh, North Carolina, USA; ⁵Smithsonian Environmental Research Center, Edgewater, Maryland, USA and ⁶Forest Global Earth Observatory, Smithsonian Tropical Research Institute, Panama, Republic of Panama

Correspondence

Albert Y. Kim
Email: akim04@smith.edu

Funding information

NSF Macrosystems, Grant/Award Number: DEB-2106015

Handling Editor: Aldo Compagnoni

[Corrections added on 12 October 2022, after first online publication: The author name has been corrected from "Ross Barreto" to "Ross Barreto".]

Abstract

1. Accurate field data are essential to understanding ecological systems and forecasting their responses to global change. Yet, data collection errors are common, and data analysis often lags far enough behind its collection that many errors can no longer be corrected, nor can anomalous observations be revisited. Needed is a system in which data quality assurance and control (QA/QC), along with the production of basic data summaries, can be automated immediately following data collection.
2. Here, we implement and test a system to satisfy these needs. For two annual tree mortality censuses and a dendrometer band survey at two forest research sites, we used GitHub Actions continuous integration (CI) to automate data QA/QC and run routine data wrangling scripts to produce cleaned datasets ready for analysis.
3. This system automation had numerous benefits, including (1) the production of near real-time information on data collection status and errors requiring correction, resulting in final datasets free of detectable errors, (2) an apparent learning effect among field technicians, wherein original error rates in field data collection declined significantly following implementation of the system, and (3) an assurance of computational reproducibility—that is, robustness of the system to changes in code, data and software.
4. By implementing CI, researchers can ensure that datasets are free of any errors for which a test can be coded. The result is dramatically improved data quality, increased skill among field technicians, and reduced need for expert oversight. Furthermore, we view CI implementation as a first step towards a data

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](#) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.
© 2022 The Authors. *Methods in Ecology and Evolution* published by John Wiley & Sons Ltd on behalf of British Ecological Society.

collection and analysis pipeline that is also more responsive to rapidly changing ecological dynamics, making it better suited to study ecological systems in the current era of rapid environmental change.

KEY WORDS

continuous integration, ecological forecasting, field research, forest ecology, Forest Global Earth Observatory, GitHub Actions, quality assurance and quality control, reproducibility

1 | INTRODUCTION

In an era when Earth's ecological systems are increasingly altered by pervasive human impacts (Vitousek et al., 2008), including but not limited to climate change, ecological systems are changing rapidly and at an accelerating pace (Alstad et al., 2016; Masson-Delmotte et al., 2021; McDowell et al., 2020). Long-term data are particularly valuable for understanding changing ecological dynamics (Anderson-Teixeira et al., 2015), and their consistency through time is critical to accurately assess such changes (e.g. Welti et al., 2021). The scientific challenges of documenting and understanding these changes, and forecasting future dynamics, call for increasingly efficient collection, analysis, and publishing of ecological field data (Dietze et al., 2018; Wolkovich et al., 2012). It is also critical that data be of high quality, with thorough documentation and minimal errors, ideally preserved in a stable open-access environment (Wolkovich et al., 2012).

Challenges to efficient and accurate collection and analysis of high-quality ecological field data are abundant. Field researchers, regardless of expertise level, are prone to data collection errors and equipment malfunctions during collection and often lack the time or expertise to conduct thorough quality assurance and quality control (QA/QC) checks before the end of the data collection period (Ferretti, 2009; Michener, 2018; Yenni et al., 2019). When field researchers lack the expertise to thoroughly check the collected data, the QA/QC process requires input from more experienced researchers who are often over-committed and struggle to provide timely data review. Sometimes, it can take years from the time of data collection to the time it is analysed for publication (Hampton et al., 2013; Marx, 2013). At that point, it is too late to fix errors that require field re-visits or reliable memories of the field experience. Moreover, a particular long-term research challenge is that field data are often collected by field technicians in transitory positions (e.g. interns, undergraduate and graduate students, temporary field technicians, postdoctoral researchers), thus many long-term data records are collected by a variety of researchers with differing background knowledge, prior experience, and attention to accuracy and detail. Systematic data collection differences among researchers and across years are likely to arise, particularly if training or QA/QC are inadequate, and data inconsistencies across years can bias scientific analyses and conclusions (Didham et al., 2020; Welti et al., 2021). A system is needed for rapid, thorough, and objective QA/QC of ecological field data, which could ensure that field researchers are promptly alerted to their errors and that final data products conform

to predetermined standards, free of detectable errors (Michael C. Dietze et al., 2018; Yenni et al., 2019).

Furthermore, given the increasingly computational nature of ecological research, issues of computational irreproducibility are increasing in frequency and severity (Alston & Rick, 2021; Milcu et al., 2018; Powers & Hampton, 2019). Data collection, pre-processing, and analysis pipeline steps are becoming increasingly dependent on correct digital data formatting, functioning statistical software code scripts, and ever-updating external software packages and extensions (McNutt, 2014; Peng, 2011). Breakdowns in this pipeline can easily occur and, if left unchecked, can hamper scientific progress. Needed are safeguards to test the reproducibility of this pipeline that both provide timely notifications of breakdowns and require a minimal number of active human interventions (Beaulieu-Jones & Greene, 2017; Stodden et al., 2016; Stodden & Miguez, 2014; Wilson et al., 2014).

Here, we design and test such a system, using a continuous integration (CI) data pipeline in GitHub using GitHub Actions. GitHub is an online, cloud-based service that can help researchers track, organize, discuss, share and collaborate on software and other materials related to research production, including data, code for analyses, and protocols (Crystal-Ornelas et al., 2022). CI is a process used by software engineers to automate the integration of code changes from multiple contributors working on a single software project; GitHub Actions is GitHub's implementation of CI.

At its simplest, whenever a user makes changes to code in a repository, a series of tests and checks called 'jobs' are automatically triggered to ensure code quality, to detect code compilation errors, and to perform unit testing. If a job fails, an alert is promptly issued to the user. Our system is modified from Yenni et al. (2019) and White et al. (2019), but adds the dimension of using the system for minimizing data collection errors by field technicians. We tested this system on three field surveys contributing to long-term data records at two large forest dynamics plots of the Forest Global Earth Observatory network (ForestGEO, Anderson-Teixeira et al., 2015; Davies et al., 2021), evaluating its potential to develop final datasets that are free of detectable errors. In addition, we examined temporal trends in data collection error rates throughout the field season, evaluating the hypothesis that by making field technicians aware of their earlier errors, implementation of the CI system reduces their subsequent data collection error rates.

Other advanced frameworks for developing robust pipelines and workflows for automatic data collection, checking, integration, and ingestion exist, such as OpenRefine (<https://openrefine.org>.

org/), Frictionless Data (<https://frictionlessdata.io/>), and the validate R package (Fowler et al., 2018; van der Loo & de Jonge, 2021). However, we opted for our system because it dovetailed well with existing resources: our data were already stored on GitHub, and much of our data wrangling was already done using highly customizable R statistical software language code. Accordingly, we emphasize that while our implementation is specific to our forest ecology data, the overall structure we detail in the next section can be generalized and transferred to data of any kind.

2 | MATERIALS AND METHODS

2.1 | Continuous integration system

We repurposed the typical software engineering CI framework to one that automates data QA/QC. In our system, whenever new data are pushed to a GitHub repository, a series of jobs ensuring the quality of the data are triggered. If data collection errors exist, a report is generated and an alert is issued. A similar report is generated for warnings: newly collected observations that should be reviewed, but that do not necessarily merit an error alert. Lastly, a dashboard of reports on completion status and other relevant information is also produced automatically.

Several platforms exist for CI, including Travis CI (<https://www.travis-ci.com/>), Jenkins (<https://www.jenkins.io/>), GitLab (<https://about.gitlab.com/>), and GitHub Actions (<https://github.com/features/actions>). We used GitHub Actions since our data and code were already stored in GitHub repositories, allowing for their easy centralization with CI procedures and outputs. Furthermore, at the time of this article's publication, GitHub Actions had a generous monthly allocation of free compute-time minutes for publicly available open-source projects such as ours. (On the other hand, private GitHub repositories had a more restrictive monthly allocation of 3000 compute-time minutes.) Furthermore, under the free GitHub plan, each GitHub Actions job can run for up to 6 hours of execution time and can consume up to 14GB of free disk space during execution. Lastly, no file in any of our repositories exceed the 100MB maximum file size allowed on GitHub; workarounds to this restriction include the arrow, pins, and piggyback R packages (Boettiger, 2022; Richardson et al., 2022; Wickham & Luraschi, 2021).

The r-lib/actions library of GitHub Actions for the R language, available at <https://github.com/r-lib/actions>, provides an interface between GitHub Actions and R (Hester, 2021); it is this library that makes it possible to program our system in R. Furthermore, we leveraged existing unit testing infrastructure in the testthat package to issue data collection error alerts (Wickham, 2011). While the combination of r-lib/actions and testthat is typically used to ensure that R packages saved in a GitHub repository pass all QA/QC checks for R package submission to the Comprehensive R Archive Network (CRAN), we repurposed it to run our data QA/QC.

A general overview of our CI system is visualized as a data pipeline in Figure 1; specific details pertaining to each of the example three surveys are discussed in Section 2.2.

The system starts with field technicians collecting field data. At this stage, data can be collected either on paper or electronically, and no internet connection is assumed. After returning from the field, data technicians (often the same individuals as the field technicians) create a spreadsheet containing new measurements, which we call data.csv. This step may require some consolidation of the individual field technicians' measurements into a single spreadsheet and/or entering the values recorded on paper into a computer.

The data technicians then push the data.csv spreadsheet to the GitHub repository, at which point GitHub Actions CI is automatically triggered. We make a distinction between files saved in the repository (enclosed in darker blue shade; Figure 1) and the sequence of automated tasks performed by GitHub Actions (enclosed in lighter orange shade; Figure 1). The sequence of tasks are specified in a .yaml file included in the .github/workflows/ folder of the repository; .yaml (Yet Another Markup Language) is a human-readable data-serialization language and is commonly used for configuration files and in applications where data are being stored or transmitted.

In our data-tests.yaml file we

1. Specify the branches, folders, and/or files that, when modified, will trigger GitHub Actions;
2. Specify the operating system(s) used for testing (Windows, macOS, or linux);
3. Import the appropriate r-lib/actions GitHub Actions for the R language, specifically, the r-lib/actions/setup-r action to set up an R environment on GitHub servers;
4. Install all R package dependencies listed in DESCRIPTION using the REMOTES package (Hester et al., 2021);
5. Run all relevant R scripts to
 - a. detect any data collection errors via generate_reports.R,
 - b. (optionally) produce data collection warnings via generate_warnings.R and
 - c. produce a dashboard via generate_dashboard.R;
6. Commit and push any new or modified files to the repository; and
7. Issue any data collection alerts/notifications specified in testthat.R via the testthat package.

In Appendix S2, we detail an example data-tests.yaml file (which can be found at <https://github.com/SCBI-ForestGEO/SCBImortality/blob/main/.github/workflows/data-tests.yaml>) and describe a YouTube screencast tutorial describing all the elements of this file available at <https://youtu.be/JKF6aXtdLHs>.

2.1.1 | Step 1: Instal all necessary R packages

In order for GitHub Actions to run all the relevant R scripts remotely on GitHub servers, all packages used must be specified in the

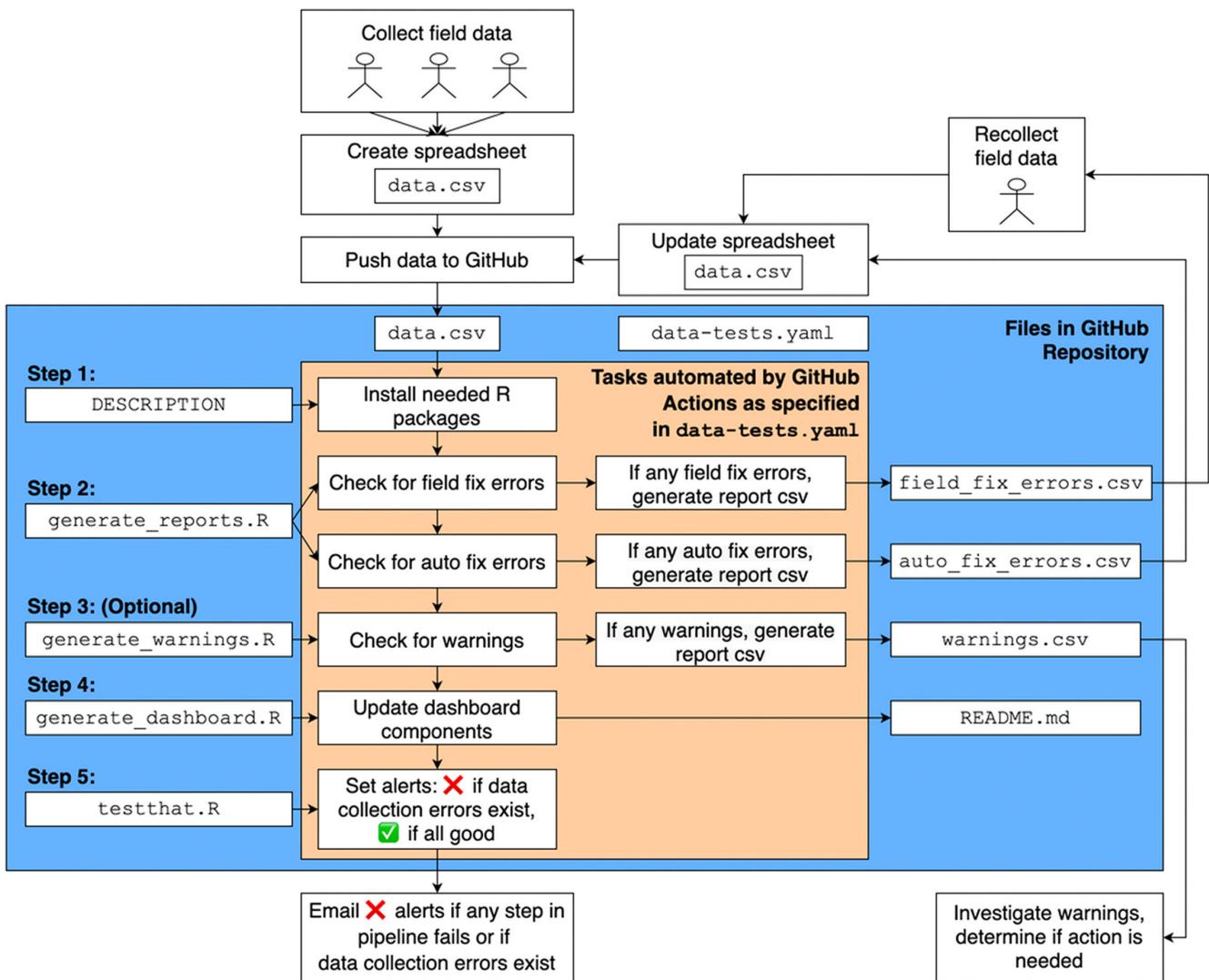


FIGURE 1 Continuous integration data pipeline used in this study to automate data quality assurance and control of field collected forest data. The process begins with field technicians collecting data, continues with data technicians committing and pushing a data spreadsheet to GitHub, which will trigger GitHub Actions, and ends with email alerts being sent automatically to the user if any steps in the process fail or if any programmed 'field fix' or 'auto fix' data collection errors exist. Upon receiving data collection error alerts, either field technicians recollect data to resolve field fix errors, or data technicians update the spreadsheet to resolve auto fix errors. Similarly, warnings that do not require field revisits are also generated.

Imports section of the DESCRIPTION file included in the root directory of the repository. This DESCRIPTION file is identical in structure and function to the DESCRIPTION file included in all CRAN R packages. For example, it is where developers specify R version and R package requirements. The r-lib/actions library will in turn call the remotes package to automatically install all specified R packages from scratch, thus testing whether the data pipeline breaks down due to differences in package versions; this default behaviour can be overridden by having packages cached so that they aren't reinstalled at each GitHub Actions run.

In our case, while we specified all packages used in DESCRIPTION, we did not specify any package version numbers, nor perform any software dependency management. It is important to note that neglecting these tasks could lead to issues caused

by changes in software versions, especially as system complexity increases. Solutions to safeguard against such issues range from simpler solutions like using the renv package or its packrat predecessor for dependency management, to more involved but more robust solutions such as using Docker containers with R via the rocker package (Boettiger & Eddelbuettel, 2017; Merkel, 2014; Ushey, 2022; Ushey et al., 2021).

2.1.2 | Step 2: Detect any data collection errors

A script named generate_reports.R is then run on data.csv to check for data collection errors. In theory, this system can run any test that can be programmed in R. In practice, tests should be carefully

defined based on the properties of the data being collected. Here, we define 'detectable errors' as those potential errors for which tests can and have been programmed in this script.

We divided errors into 'field fix' and 'auto fix' errors. 'Field fix' errors require field technicians to return to the field and recollect measurements. If a field fix error is detected, a `field_fix_errors.csv` error report is automatically generated and saved to the repository. Equipped with this error report, a field technician (usually the one who recorded the original error) returns to the field and retakes the appropriate measurement. The new data are then consolidated and used to update the `data.csv` spreadsheet. Example field fix errors include when tree information was not recorded, such as missing crown position values, or a stem being marked as unhealthy or dead but no factor associated with decline was recorded.

Alternatively, 'auto fix' errors do not require field technicians returning to the field to retake measurements. If an auto fix error is detected, an `auto_fix_errors.csv` error report is similarly generated and saved to the repository, at which point the data technician can correct these errors by merely updating the `data.csv` spreadsheet. Example auto fix errors include when a tree is marked as having a wound but is not marked unhealthy.

We detail more examples of both field fix and auto fix errors for the mortality census and dendroband surveys in Sections 2.2.1 and 2.2.2, as well as in the complete list of errors in Tables S1 and S2.

In the presence of either type of error, an error alert is emailed to the data technician, directing them to the Actions tab of the GitHub repository homepage for error identification and handling; this is represented by the lowest block in Figure 1. We detail this further in the forthcoming Section 2.1.5.

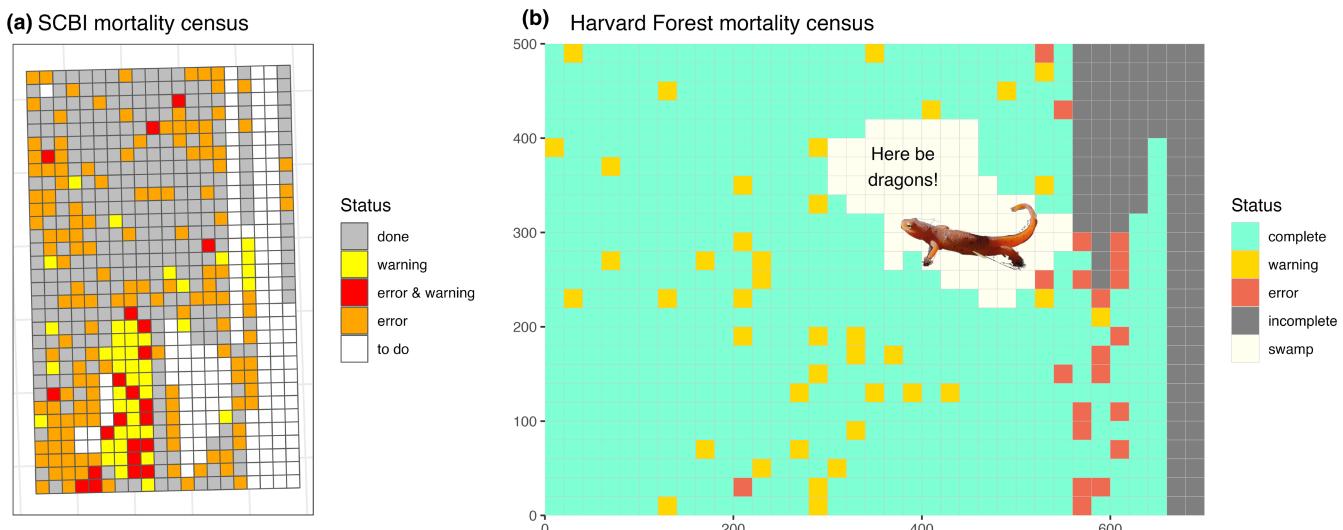


FIGURE 2 Snapshots of SCBI and Harvard Forest mortality census repository dashboards: Quadrat maps illustrating census completion status. For each 20×20 m quadrat at both sites, the dashboard reports whether the quadrat (1) has been fully and successfully censused; (2) has warnings of note but do not require action; (3) has data collection errors that require either field or data technician intervention; (4) has both warnings and data collection errors; or (5) has not been censused yet. The images represent mortality census quadrat completion statuses as of July 26, 2021.

2.1.3 | (Optional) Step 3: Run other scripts

Other optional scripts are run at this point; for example, further data wrangling and analysis scripts. In our case, a `generate_warnings.R` script is run to produce warnings. As opposed to data collection errors which necessitate some form of action, warnings do not necessarily merit error alerts, but identify observations that may still be of interest for further investigation. An example warning would be if a tree previously marked as dead was later recorded as alive. If such warnings exist, a `warnings.csv` file is automatically generated and saved to the repository.

2.1.4 | Step 4: Produce dashboard

Next, a `generate_dashboard.R` script is run to automatically produce a dashboard whose elements quickly provide reports, such as images and data relating to progress and results, completion maps, error rate tracking, and other summaries. In our case, dashboard elements such as images and tables are saved to the GitHub repository and then displayed on the repository `README.md` cover page. We present example dashboard elements for the three surveys in Section 2.2 and in Figures 2 and 3. More sophisticated dashboards could include external websites, automatically published using GitHub Pages (<https://pages.github.com/>) for example (White et al., 2019; Yenni et al., 2019).

Another step that could be performed at this point is data archiving, whereby a timestamped permanent archive of the data is stored. For example, in their continuous integration pipeline White et al. (2019) included an R script that, every time data are pushed to the repository, automatically publishes a repository version on the

general-purpose open repository site, Zenodo (<https://zenodo.org/>), and obtains a corresponding Digital Object Identifier (DOI) number. In our case however, we only periodically manually update our data repositories' versions on Zenodo and include a Zenodo badge on the repository README.md cover page.

2.1.5 | Step 5: Issue any alerts or notifications

Finally, our CI system can issue alerts or notifications, typically of one of two types: a 'red' alert indicating a failed process or a 'green' alert indicating a successful one. In the event of a failed process, GitHub Actions will email a notification to the data technician who triggered GitHub Actions. In our implementation, a failed process email notification alert is issued to the data technician who originally pushed data to GitHub in one of two cases.

First, failed process alerts are generated if any step in the data pipeline, visualized in [Figure 1](#), fails due to errors in the code. Examples include errors caused by changes to the code, as well as errors arising from differences between different versions of the same software. Any such failed processes require prompt fixing of all coding errors so that the entire process in [Figure 1](#) can be completed. Such failed process alerts act as safeguards, ensuring code reproducibility and the robustness of the pipeline against all code and software changes.

Second, failed process alerts are also programmed to be generated when any of the data collection errors from [Section 2.1.2](#) are detected. As opposed to failed process alerts due to errors in the code, failed process alerts due to data collection errors must be explicitly programmed by the data technician as described in [Section 2.1.2](#) and then operationalized using the `testthat` package (Wickham, 2011). The typical use of the `testthat` package is for R package validation, whereby the parameters that define R package compilation error alerts are defined in a `testthat.R` file. In our case, however, the `testthat.R` file in [Figure 1](#) defines when a data collection error alert should be triggered—specifically, when a `field_fix_errors.csv` error report gets created in Step 2.

2.2 | Deployment of the continuous integration system

The CI system described above was implemented for three ecological surveys at two ForestGEO large forest dynamics research sites: the Smithsonian Conservation Biology Institute (SCBI; near Front Royal, VA, USA, 113 km [70 miles] west of Washington, DC, USA; latitude 38.8935, longitude -78.1454) and Harvard Forest (HF; near Petersham, MA, USA, 113 km [70 miles] west of Boston, MA, USA; latitude 42.5388, longitude -72.1755). SCBI is a 25.6 ha (400 × 640 m) plot dominated by broadleaf deciduous tree species including *Liriodendron tulipifera* L. and several species of *Quercus*. Harvard Forest is a 35 ha (700 × 500 m) plot dominated by *Tsuga canadensis* (L.) Carrière, *Acer rubrum* L., and *Quercus rubra* L. The

three surveys included two annual tree mortality censuses, one at each site, and a dendrometer band survey at SCBI, as detailed below.

2.2.1 | SCBI and Harvard Forest mortality censuses

Beginning in 2014, annual tree mortality censuses were established at SCBI (Gonzalez-Akre et al., 2016). In 2021, the annual tree mortality census protocols of SCBI and other ForestGEO sites (Arellano et al., 2021) were standardized and applied across five temperate ForestGEO sites, including SCBI and HF. This was the eighth annual tree mortality census at SCBI, and the first at HF. Each census included all stems ≥10 cm diameter at breast height (DBH; 1.3 m above the ground) in the most recent ForestGEO core census (2013 or 2018 at SCBI, 2019 at HF) where every stem ≥1 cm DBH was mapped, tagged, identified to species, and measured at DBH (Anderson-Teixeira et al., 2015; Condit, 1998). Data collected as part of the tree mortality census included status (alive, alive unhealthy, dead standing or dead fallen), crown position, crown condition and signs of damage, disease or decay (full protocol available in the SCBI mortality GitHub repository at <https://github.com/SCBI-ForestGEO/SCBImortality>). The SCBI census included add-ons to capture the impacts of emerald ash borer (EAB, *Agrilus planipennis* Fairmaire) invasion, wherein all stems ≥1 cm DBH of host tree species (*Fraxinus americana*, *F. nigra*, *F. pennsylvanica*, and *Chionanthus virginicus*) were censused, and EAB-specific symptoms of tree decline assessed.

At SCBI, the 2021 mortality census was conducted from June 2 to August 18 by a full-time intern, two other part-time interns who provided assistance (all with minimal prior experience in forest field work), and a graduate student with prior experience including conducting the same mortality census earlier in the year at other ForestGEO sites (Smithsonian Environmental Research Center and the Ordway Swisher Forest Dynamics Plot). Training and occasional help was provided by an experienced field technician and a principal investigator. At HF, the 2021 mortality census was conducted from July 8 to July 29 by three field technicians who had previously conducted the census at other sites, three forest technicians, and a guest surveyor.

We activated the CI system to check for errors on July 6 at SCBI, which was approximately mid-way through the mortality census there, and on July 8 at HF, which was immediately after the first day of data collection. Thus, the first time field technicians were equipped with feedback on their previous data collection was the next field collection date: July 7 at SCBI and July 9 at HF.

At both sites, measurements were digitally recorded using the FastField app loaded on iPads. FastField forms (<https://www.fastfieldforms.com/>) are customizable, version-controlled forms that facilitate mobile but internet connection-free data collection. A data technician first built the FastField form using a template Excel file to populate the necessary fields. They then programmed the form with first-stage data QA/QC (e.g. making certain data fields be required, limiting valid numerical ranges, presenting categorical choices for categorical variables, and other metadata considerations) and conditional statements to make certain fields appear when specific

Dendrobands dashboard: tree tags with measurements alerted to be anomalous
 Stars = alerted measures (vertical lines = alert date), circles = erroneous measures, dots = correct measures
 Solid lines = correct database, dashed lines = (hypothetical) uncorrected database

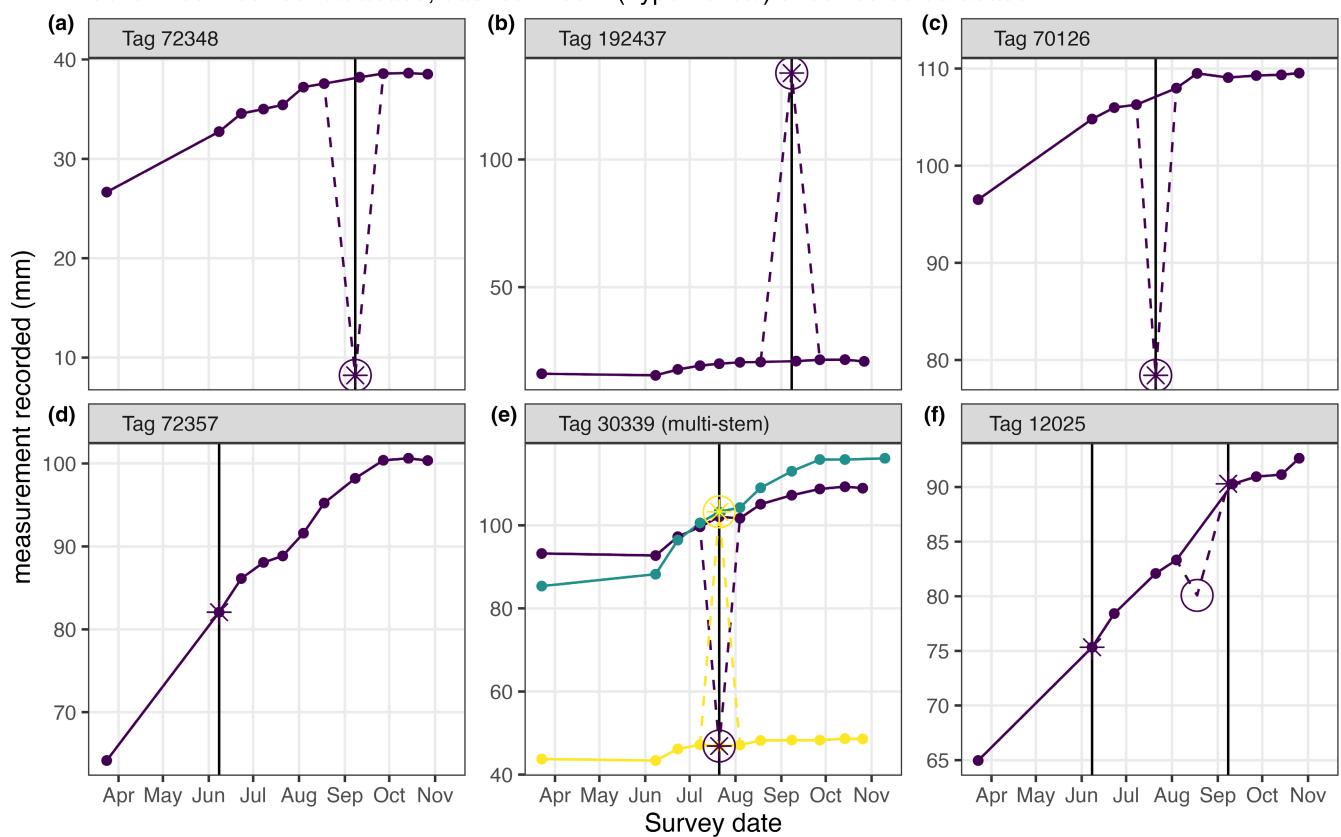


FIGURE 3 Snapshot of SCBI dendroband survey repository dashboard: Time series plots displaying anomalous dendroband measurements identified using continuous integration for six trees. Examples include (a, b) two cases where measurements >10 mm different from the previous measurement were detected and corrected the next day, (c) a case where an anomaly was detected but could not be re-measured quickly enough, and so simply removed from the data record, (d) a false alert where a tree's circumference increased by >10 mm between the spring survey and the next measurement in early June (a nontypical time gap between measurements caused by COVID-19), (e) an alert issued when measurements were accidentally transposed between two stems of a multistem tree, and (f) an instance where an anomalous measurement passed the screening immediately after it was taken, but was identified and retroactively deleted when an alert was issued for the subsequent measurement. Similarly to the example in (d), this record also contains a false alert that occurred in early June.

conditions were met (e.g. only display fields required for dead trees if the tree was recorded dead). Other fields were programmed to auto-populate (e.g. date and time). We note that FastField Forms are one of multiple potential data collection interfaces and were not specifically required for our CI system. For example, Google Sheets (<https://www.google.com/sheets/about/>) allows for offline access, whereby synchronizing with the cloud occurs as soon as an internet connection is reestablished. Furthermore, while a few first-stage data QA/QC checks can be programmed into FastField forms, users are limited to a set of generic form input validations; the more sophisticated data QA/QC checks required by our projects necessitated custom R code, which in turn was automated by our CI system.

FastField survey forms were populated with a list of trees to be included in the survey, sorted into established 20×20m quadrats, and with existing data including stem coordinates, species, prior DBH and prior status (e.g. 'dead standing'). Field technicians were assigned individual 20×20m quadrats (out of 640 at SCBI and out of 875 at HF)

where they censused each tree ≥ 10 cm DBH. After completing each quadrat, field technicians submitted data within the FastField Forms app. Individual submissions were assigned a field technician ID indicating who collected the data, a timestamp, and a unique submission ID number. Upon return from the field, field technicians connected iPads to the internet and synchronized their individual FastField submissions to the cloud via the FastField website. A data technician then downloaded a single Excel file containing all the individual submissions; this Excel file corresponds to data.csv in Figure 1 and contains a collated and cumulative record of all submissions.

Data files were then pushed to GitHub repositories corresponding to each site: SCBImortality for SCBI and HarvardMortality for HF (see Data Availability Statement). The CI systems for the two mortality censuses are very similar since the HarvardMortality repository started as a clone of the pre-existing SCBImortality repository, with slight modifications to adjust for census add-ons and minor differences in plot setup.

For each site, a series of automated tests for data collection errors and warnings were applied ($n = 31$ at SCBI and 23 at HF). Automated tests included an assessment of census progress, completion checks testing whether all stems in censused quadrats had been visited and whether all required data were present, consistency checks testing for internal consistency within each new data record, and consistency checks comparing new records with prior data (e.g. new DBH measurements were consistent with the previous census measurements; see Table S1 for a complete list of automated tests). As described in Section 2.1.2, all data collection errors were classified as either 'field fix' or 'auto fix' errors.

The CI system reported field fix errors in `field_fix_errors.csv`. Example field fix errors include (1) stem not censused, (2) missing crown position, and (3) status recorded as unhealthy or dead, but no factors associated with decline recorded. The CI system similarly reported auto fix errors in `auto_fix_errors.csv`. Example auto fix errors included (1) a wound was recorded, but the tree was classified as healthy (classification changed to unhealthy, by definition), and (2) a wound severity level was recorded, but wound was not selected from a list of factors associated with decline. Finally, the system produced warning alerts in `warnings.csv`. Example warnings included (1) a tree was marked as alive but was recorded as dead in a previous census and (2) a newly dead tree DBH measurement was >2 cm different from the previous census measurement. We note that the errors in this final category could have arisen from a previous census error and did not necessarily indicate a new data collection error.

After producing all necessary error alerts and warnings, the repository `README.md` dashboards were populated with an overall completion percentage and a map of all quadrats illustrating the corresponding error, warning, and completion statuses (Figure 2). Using the map and error reports as a guide, field technicians returned to the field to correct problems until all errors had been resolved, resulting in a highly efficient method to obtain a final dataset that passed all programmed data QA/QC checks.

2.2.2 | SCBI dendrometer band surveys

Dendrometer bands have been used for fine-scale measurement of tree diameter growth at SCBI beginning in 2010. Metal dendrometer bands were constructed following the protocol of Muller-Landau and Dong (2010) and installed within 20 cm of breast height (full protocol available in the SCBI Dendrobands GitHub repository at <https://github.com/SCBI-ForestGEO/Dendrobands>). Bands were initially installed following this schedule: 243 in 2010, 505 in 2011, 515 in 2012, and 566 in 2013. The number of bands has since fluctuated slightly as trees have died or bands were otherwise lost and new bands were eventually added to compensate for these losses. Once bands were installed, precise measurements of the band gap or window were taken to a hundredth of a mm using digital callipers. All bands are measured biannually ($n = 549$ in 2021): once at the beginning of the growing season (in the spring, around March) and once at the end (in the fall, around November). A subset of these

bands are further measured biweekly throughout the growing season starting ($n = 146$ in 2021; Dow et al., 2022).

In 2021, the spring biannual census was conducted on March 23 and 24 by an experienced field technician, 10 biweekly censuses were conducted between June 8 and October 26 by interns trained by an experienced field technician, and the fall biannual census was conducted on October 26 and 27 by an experienced field technician, a principal investigator, and an intern. We activated the CI system for the SCBI dendrometer band survey mid-way through the 2021 growing season on July 21, 2021, which was after the spring biannual survey and three biweekly surveys.

As opposed to both tree mortality censuses, where field data were collected using the FastField app, dendrometer band data were entered into .csv files either directly on iPads or on paper forms first and then later transcribed to a computer. Spreadsheets were then assigned a unique survey identification value of form YEAR-SURVEYID (e.g. 2021-03 indicates the third survey for the 2021 growing season). This spreadsheet corresponds to `data.csv` in Figure 1.

Data files were then pushed to the GitHub repository for dendrometer band data (see Data Availability Statement). Before running any data collection error tests, the Dendrobands CI system was programmed to collate independent spreadsheets for each census into an annual summary spreadsheet, which required checking that individual file column structure, number, and types matched.

Examples of data collection error tests for the two biannual and 10 biweekly surveys included (1) incorrect census number, (2) measurement not recorded and, of most interest to us, (3) measurements that were 'anomalous' as compared to previous data. For the biweekly surveys, these were measurements that differed from the previous ones by ≥ 10 mm. For the stems that were only measured at the spring and fall biannual surveys but not biweekly, we tested whether their overall 2021 growth was within ± 3 standard deviations of the previous year's species-specific distribution of growth; for example comparing the 2021 growth of all biannual *Liriodendron tulipifera* stems to the distribution of 2020 growth for all conspecific stems. If a particular stem's overall 2021 growth fell outside ± 3 standard deviations, its fall measurement was flagged as anomalous. All anomalous measurements were displayed in time series plots posted to the repository `README.md` dashboard (Figure 3, which we discuss further in Section 3.2). All our data collection tests are listed in Table S2.

The CI system also produced warning alerts in `warnings.csv`. In particular, a warning was generated when a dendrometer band needed repair or replacement because the calliper measurement window on the dendrometer band was too close to the minimum or maximum measurable length, or if a field technician coded the need for replacement.

2.3 | Evaluation of the continuous integration system

We present the methods used to evaluate the CI system's effectiveness at ensuring data QA/QC and reducing original error rates.

2.3.1 | SCBI and Harvard Forest mortality censuses

For both tree mortality censuses, we evaluated (1) the numbers and types of errors detected and fixed, and (2) changes in the original error rates over time (i.e. before and after implementing CI at SCBI, and throughout the HF census).

To support both analyses, we programmed the CI system to create cumulative trace reports that logged the date all stems were censused and information on all data collection errors and warnings. We then analysed these logs to compute, for given time intervals, (1) the number of stems censused and (2) the error rate (defined as the number of errors divided by the number of stems censused). Given the different number of field technicians and survey completion times between SCBI and Harvard Forest, we report the number of stems censused and error rates at different time scales: weekly values for SCBI and daily values for Harvard Forest.

For the SCBI mortality census, where the CI system started mid-way through the census, we define the 'pre-CI' initial error rate as the error rate for the time period from the start of the survey to the day prior to CI activation. Correspondingly, we define the 'post-CI' final error rate as the error rate over the time period from CI activation to the last day of the census. We assessed the effect of the CI system on data QA/QC and field technician learning by comparing the two separate error rates. To test the significance of any change in error rate between the pre-CI to the post-CI period, we conducted a nonparametric two-sample hypothesis test using a two-sample permutation test based on 10,000 permutations under a null hypothesis of no true difference in error rates; the two samples consisted of the stems censused pre-CI and post-CI respectively. We performed this test for field fix errors and auto fix errors separately.

2.3.2 | SCBI dendrometer band surveys

For the dendrometer band survey, we evaluated the number and types of errors detected and fixed. As with the tree mortality census, we programmed the CI system to create cumulative trace reports that logged information on all data collection errors and warnings. We did not test for whether implementation of the system reduced original error rates because we had a lower field measurement sample size, and because dendrometer band measurements were conceptually more straightforward than the tree mortality census measurements.

3 | RESULTS

For all three surveys, the CI system implementation resulted in final data that were error-free for the checks we programmed. Below, we describe error detections and, for the mortality censuses, temporal trends in original error rates over the course of the surveys.

3.1 | SCBI and Harvard Forest tree mortality censuses

The SCBI tree mortality census included 8393 stems, censused between June 2 and August 18. The CI activation date of July 6 divides this period into pre-CI and post-CI periods, with 4814 and 3579 censused stems, respectively (Figure 4a). During the pre-CI period, a total of 461 field fix errors and 452 auto fix errors were detected, yielding original error rates of 0.096 and 0.094 errors stem^{-1} , respectively. During the post-CI period, a total of 73 field fix errors and 140 auto fix errors were detected, yielding original error rates of 0.02 and 0.039 errors stem^{-1} , respectively (Figure 4c). This corresponded to declines from the pre-CI to the post-CI period in field fix error rates of 0.075 errors stem^{-1} and in auto fix error rates of 0.055 errors stem^{-1} (both statistically significant at $p < 0.001$).

To put the declining trend of the error rate for the 2021 SCBI mortality censuses into perspective, we attempted to conduct a similar, retroactive analysis for the previous 2014–2020 mortality censuses at SCBI. However, this could not be achieved in a manner that would be scientifically robust or that would yield meaningful comparisons because the 2021 protocol was a lot more strict and thorough than previous years (regardless of CI), including changes in the criteria determining which variables would be measured on each tree. Additionally, starting in 2021 electronic FastField forms were used instead of paper forms. Lastly, many post-hoc checks for errors and fixes did occur on previous years' data, albeit in a nonsystematic and less immediate fashion. Thus reverting the previous years' data to an appropriate state for comparison with the 2021 data was impossible.

The Harvard Forest mortality census included 30,946 stems, censused between July 8 and July 29 (Figure 4b). During this period a total of 628 field fix errors and 355 auto fix errors were detected, yielding original error rates of 0.02 and 0.011 errors stem^{-1} , respectively (Figure 4d). A large spike of field fix errors occurred on July 8, 2021, which was the first day of data collection. After the field technicians started receiving feedback from the CI system, the error rate declined steeply and subsequently remained low.

3.2 | SCBI dendrometer band surveys

The 2021 dendrometer band survey at SCBI included a total of 2412 measurements on 549 bands between March 23 and November 15. (Note that while the final biannual survey took place on October 26 and 27, eight measurements that were originally identified as anomalies were re-measured on November 15).

The system detected a total of 89 data collection errors, where 44 measurements were alerted to be anomalies, the incorrect survey identification value was entered four times, an incorrect status code was entered nine times, and no measurement was recorded 11 times. The field fix errors relating to measurements necessitated a field technician returning to the field and verifying the measurement (if this could be done reasonably soon after the alert); otherwise the alerted

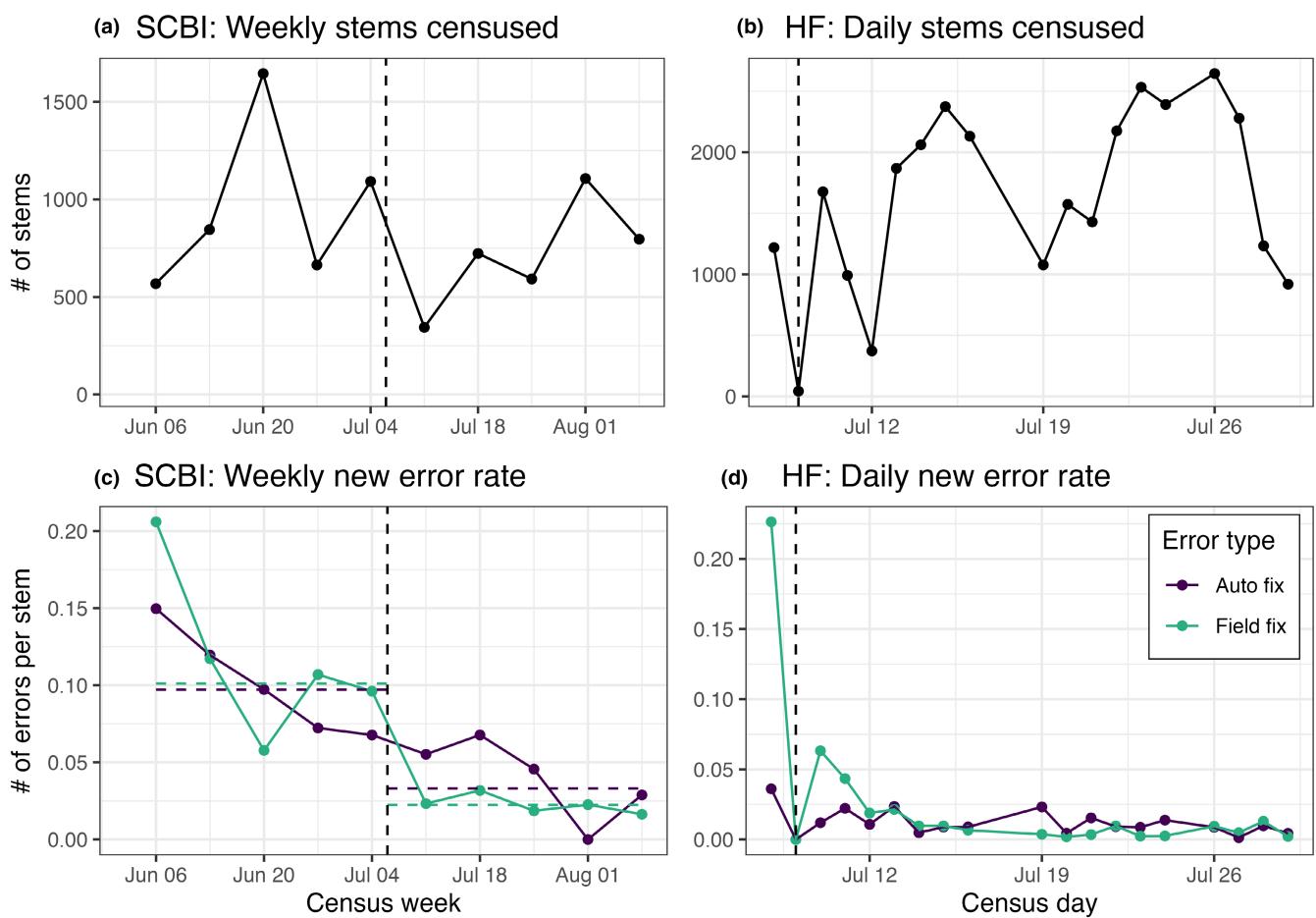


FIGURE 4 Tree mortality census stem counts, field fix error rates, and auto fix error rates for SCBI (a, c) and Harvard Forest (b, d). In (a, c), data were aggregated weekly while in (b, d) data were aggregated daily, reflecting the differing time lengths it took to complete the two surveys. Continuous integration was activated at SCBI on July 6, 2021, which was mid-way through the census, and at HF on July 7, 2021, which was the first data collection date. Field technicians were thus first equipped with feedback from the continuous integration system on the first subsequent data collection date, July 7 at SCBI and July 8 at HF (marked with dashed vertical lines). For the SCBI time series plots, the dashed horizontal lines mark overall pre-CI and post-CI error rates. The differences in pre versus post error rates for both types of errors were statistically significant ($p < 0.001$).

value was deleted. Auto fix errors relating to the survey identification value and code were fixed in the database by a data technician.

The majority of the 89 dendrometer band measurements flagged as anomalous were incorrect and either re-measured within a few days or removed from the database (e.g. Figure 3a-c,e). However, our coarse anomaly detection algorithm for the biweekly survey ($>10\text{mm}$ difference from previous measurement) had two shortcomings.

First, false alarms of anomalies occurred for four trees whose growth exceeded 10mm between pre-growing season biannual measurements in late March and the start of biweekly measurements in early June (Figure 3d,f). This measurement gap of nearly two and a half months during the peak survey season was caused by COVID-19 induced staffing problems, and thus would not have occurred under normal operations.

Second, a more nuanced shortcoming occurred where an anomaly was not detected in real time at the time of measurement, but rather only during a subsequent survey. This initially undetected anomaly (Figure 3f) occurred for tag 12,025 at the August 18th

measurement. This measurement was only identified as an anomaly after collecting the subsequent September 8th measurement and performing a visual trend analysis. The original August 18th measurement was thus retroactively removed from the database.

4 | DISCUSSION

The implementation of our CI system led to reductions in data errors, resulting in final datasets free of detectable errors. Moreover, in mortality censuses at both SCBI and HF, error rates declined precipitously, and with a large effect size, immediately after the CI implementation (Figure 4c,d). Additionally, the continuously updated dashboards (e.g. Figures 2 and 3) were useful for all team members to assess progress. Finally, use of this system acted as a reproducibility safeguard by quickly alerting computer technicians to failures in the data pipeline due to changes in code, data, and software.

While this approach makes an enormous advance in guaranteeing that final datasets are free of 'detectable' errors, there remain two classes of errors that may go undetected. First, some errors can never be detected based on review of data because no test can be written. An example would be if a field technician fails to notice a wound on a tree and incorrectly records the tree as unwounded and healthy. Second, and more importantly, errors that are theoretically detectable will only be caught if appropriate tests have been defined and appropriately coded. An omission or oversight will allow such errors through. For example, after completing the HF census, data technicians realized that a couple minor wording differences between the HF and SCBI censuses had resulted in some tests failing to run, the result being that the dataset included some errors that had to be revisited after the end of the census. Thus, test definition merits careful consideration and review, and the corresponding code should be structured to ensure all tests work on the particular dataset being collected, for example by taking a previous year's dataset and deliberately injecting known errors into it.

While the tests applied here were fairly simple (completion tests, consistency tests, and rudimentary tests for consistency with previous data), there is vast potential to increase test sophistication. For example, our anomaly detection algorithm for biweekly dendrometer band measurements (Figure 3) used a crude classification rule: 'Does the new measurement differ by 10 mm or more than the previous measurement?' This approach can be improved by using a classification rule based on species growth rates over time instead of absolute differences in growth. This would help lower the false alarm rate and improve immediate detection of anomalies (e.g. avoiding errors in Figure 3d,f). In the long run, we can also use existing Bayesian models of tree growth that fuse dendrometer band data with other data sources (e.g. census and tree-ring data) to forecast reasonable dendrometer band readings and produce warnings when newly measured values fall outside a prescribed confidence/prediction interval (Clark et al., 2007; Heilman et al., 2022; Schliep et al., 2014). The latter approach would be more sensitive at detecting measurement anomalies; it would likely have caught the anomalous dendrometer band reading that incorrectly passed our rudimentary test for growth outliers at the time it was taken and was only later identified as an anomaly (Figure 3f). Thus, while the datasets collected here have far fewer errors than would be possible without implementation of such a system, test refinement could produce even more reliable data.

In both mortality censuses, field-fix error rates declined immediately and dramatically following the implementation of the CI system (Figure 4c,d). At SCBI, where the CI system was implemented midway through the census, the post-CI implementation mean field-fix error rate declined dramatically to 21.3% of the pre-CI mean ($p < 0.001$). While this suggests that implementation of the system directly reduced data collection error rates, a direct causal statement on the learning effect engendered by the CI system is difficult to make, chiefly due to the nonexperimental nature of the pre-CI versus post-CI comparison. However, there are three reasons why we believe these declines were still attributable to initiation of the

CI system. First, the timing of declines differs at the two sites and are coincident with the implementation of the CI system. Specifically, field-fix error rates declined to a relatively stable, low value much earlier during the census at HF, where the field team was notified of errors immediately after the first day of data collection (Figure 4d), compared to SCBI, where the census had been in progress for 1 month prior to CI implementation (Figure 4c). Second, field-fix error rates, which required field technicians to go back to the field, declined more dramatically following implementation of CI than did auto-fix errors, which were fixed via computer code by data technicians (Figure 4c,d). This observation suggests that the error rate reduction was linked to the field technicians' level of awareness of past errors. Finally, and perhaps most significantly, all field technicians and principal investigators responsible for project oversight anecdotally reported that after receiving feedback on their data collection errors, they were much more mindful of their data collection technique during subsequent field visits. Thus, while a direct experimental test for a causlearning effect was not conducted here, the evidence strongly suggests that such an effect occurred.

The value of a CI system, such as that applied here, lies not just in the system and its outputs in and of themselves, but also the inherent thought and adherence to best practices required for its design and implementation. First, designing and writing tests requires clearly defined variables, variable codes, and allowable value ranges, and other such metadata. Second, for CI to function, an absolutely strict threshold of code reproducibility is necessary, right down to the robustness to package version changes (Beaulieu-Jones & Greene, 2017; Stodden et al., 2016; Stodden & Miguez, 2014; Wilson et al., 2014). Third, because the system requires automation from end-to-end, the number of manual interventions must be minimized; for example, the number of variables that are coded statically as opposed to dynamically (e.g. dates, file names). Fourth, all the steps in the data pipeline are enumerated in the GitHub Actions .yaml files located in the .github/workflows/ folder, facilitating clear communication of all necessary data processing and wrangling steps to current and future collaborators. Data centralization, and the scripts that check and manipulate them in a single stable repository accessible to any collaborator, ensure clear organization, version control, and tractability. Finally, use of the CI system ensures that everyone involved in data collection and quality control adhere to best practices, and any failure would be readily detected and could be promptly addressed.

The CI system described here proved successful for three surveys at two sites, and expansion to other sites and census types—while almost certainly worthwhile—will require a few considerations. First, while the CI system is compatible with any data collection system, its effectiveness would be maximized by a system that allows pushing newly collected data to GitHub directly from the field. This would provide field technicians with an assessment of newly collected data within minutes of data submission, allowing them to address any problems before leaving the data collection site. However, in situations where this is not possible (e.g. no internet connection or inappropriate device configuration), as was the case in this study,

the system is still enormously valuable. A CI system could be used at remote sites or with data collection on paper, with its value proportional to the frequency with which data could be entered (or synchronized) and checked. Frequent data checking, especially near the beginning of censuses, is particularly valuable given the apparent learning effects produced by using the CI system (Figure 4c,d).

An additional consideration pertains to use of this CI system across sites for the same census type. Even deploying the mortality census to just two sites was a difficult task of coordination: changes in the R code tests in one repository had to be manually mirrored in the other. In the future, such standardization can be done by having all tests coded as functions in a shared R package. Using a CRAN based repository version of the package would not be appropriate as frequent updating of CRAN packages is not permitted. Rather, as part of the CI routine, a development version of the package could be installed at each push using the `install_github()` function from the `remotes` package (Hester et al., 2021).

As mentioned earlier, our system is a modification of Yenni et al. (2019) and White et al. (2019)'s system for collecting and analysing rodent population data. We did so by mirroring the overall GitHub repository and GitHub Actions structure detailed in Figure 1, but replacing all rodent specific R scripts with tree mortality and dendroband specific R scripts: `generate_reports.R`, `generate_warnings.R`, and `generate_dashboard.R`. This same mirroring can be replicated to establish a CI system for QA/QC of data of any kind, in particular if researchers have pre-existing R scripts that can be reused. We have included resources for those new to GitHub Actions in Appendix S1, including a description of a quickstart tutorial available at <https://www.updatingdata.org/githubactions/>.

In a broader sense, this system opens the door not only to improved QA/QC, but also to a scientific inquiry system wherein the cycle of hypothesis formation, data collection, hypothesis testing, and result dissemination is greatly accelerated (Dietze et al., 2018; Dietze, 2017). Combined with a Bayesian system for generating near-term iterative forecasts of some ecological phenomenon along with their corresponding uncertainty estimates, this system would allow automated hypothesis testing (White et al., 2019), even as data are still being collected. Data analysis relative to current model expectations in near real-time can quickly alert researchers to unexpected dynamics or gaps in understanding, giving them opportunities to adjust data collection and test new hypotheses before the relevant data collection window closes. For example, an unexpectedly high mortality rate of a particular tree species, which could be revealed while a tree mortality census is still in progress (Figure 4), could alert researchers to the potential of a taxa-specific forest insect or pathogen (Anderson-Teixeira et al., 2021). Surveyors could then diagnose and document taxa-specific symptoms of decline during the current census. Thus, we view this CI implementation as the first step towards a system of data collection that is not only more accurate and efficient, but also more responsive to rapidly changing ecological dynamics, making it better suited to study ecological systems in the current era of rapid environmental change.

AUTHOR CONTRIBUTIONS

Albert Y. Kim, Valentine Herrmann and Kristina J. Anderson-Teixeira conceived the ideas and designed the CI system; Albert Y. Kim, Valentine Herrmann, Erika Gonzalez-Akre, Daniel J. Johnson, and Ian R. McGregor wrote code for the CI system implemented in the GitHub repositories; Albert Y. Kim, Valentine Herrmann, Ross Barreto, Brianna Calkins, Daniel Johnson, Jennifer Jordan, Lukas Magee, Nicolle Montero, Karl Novak, Teagan Rogers, Jessica Shue, and Kristina Anderson-Teixeira collected or oversaw collection of data; Albert Y. Kim and Kristina J. Anderson-Teixeira led the writing of the manuscript; All authors contributed critically to the drafts and gave final approval for publication.

ACKNOWLEDGEMENTS

We thank Ethan White and four anonymous reviewers for their helpful input, and to William McShea and all other researchers who contributed to field data collection. This research was funded by an NSF Macrosystems grant (NSF DEB-2106015) to Daniel J. Johnson, Kristina J. Anderson-Teixeira and collaborators.

CONFLICT OF INTEREST

The authors have no conflicts of interest to declare.

PEER REVIEW

The peer review history for this article is available at <https://publon.com/publon/10.1111/2041-210X.13982>.

DATA AVAILABILITY STATEMENT

Smithsonian Conservation Biology Institute tree mortality census data are publicly available via GitHub repository: <https://github.com/SCBI-ForestGEO/SCBImortality>, which is periodically archived in Zenodo (Herrmann et al., 2021). Harvard Forest tree mortality census data are publicly available via GitHub repository: <https://github.com/SCBI-ForestGEO/HarvardMortality>, which is periodically archived in Zenodo (Kim et al., 2022). Smithsonian Conservation Biology Institute dendrometer band survey data are publicly available via GitHub repository: <https://github.com/SCBI-ForestGEO/Dendrobands>, which is periodically archived in Zenodo (McGregor et al., 2021).

ORCID

Albert Y. Kim  <https://orcid.org/0000-0001-7824-306X>
Valentine Herrmann  <https://orcid.org/0000-0002-4519-481X>
Erika Gonzalez-Akre  <https://orcid.org/0000-0001-8305-6672>
Daniel J. Johnson  <https://orcid.org/0000-0002-8585-2143>
Jennifer A. Jordan  <https://orcid.org/0000-0001-5521-360X>
Lukas Magee  <https://orcid.org/0000-0003-3124-2275>
Ian R. McGregor  <https://orcid.org/0000-0002-5763-021X>
Teagan Rogers  <https://orcid.org/0000-0002-8828-9640>
Jessica Shue  <https://orcid.org/0000-0003-3026-5838>
Kristina J. Anderson-Teixeira  <https://orcid.org/0000-0001-8461-9713>

REFERENCES

Alstad, A. O., Damschen, E. I., Givnish, T. J., Harrington, J. A., Leach, M. K., Rogers, D. A., & Waller, D. M. (2016). The pace of plant community change is accelerating in remnant prairies. *Science Advances*, 2(2), e1500975. <https://doi.org/10.1126/sciadv.1500975>

Alston, J. M., & Rick, J. A. (2021). A beginner's guide to conducting reproducible research. *The Bulletin of the Ecological Society of America*, 102(2), e01801. <https://doi.org/10.1002/bes2.1801>

Anderson-Teixeira, K. J., Davies, S. J., Bennett, A. C., Gonzalez-Akre, E. B., Muller-Landau, H. C., Joseph Wright, S., Salim, K., Almeyda Zambrano, A. M., Alonso, A., Baltzer, J. L., Basset, Y., Bourg, N. A., Broadbent, E. N., Brockelman, W. Y., Bunyavejchewin, S., Burslem, D. F. R. P., Butt, N., Cao, M., Cardenas, D., ... Zimmerman, J. (2015). CTFS-ForestGEO: A worldwide network monitoring forests in an era of global change. *Global Change Biology*, 21(2), 528–549. <https://doi.org/10.1111/gcb.12712>

Anderson-Teixeira, K. J., Herrmann, V., Cass, W. B., Williams, A. B., Paull, S. J., Gonzalez-Akre, E. B., Helcoski, R., Tepley, A. J., Bourg, N. A., Cosma, C. T., Ferson, A. E., Kittle, C., Meakem, V., McGregor, I. R., Prestipino, M. N., Scott, M. K., Terrell, A. R., Alonso, A., Dallmeier, F., & McShea, W. J. (2021). Long-term impacts of invasive insects and pathogens on composition, biomass, and diversity of forests in Virginia's Blue Ridge Mountains. *Ecosystems*, 24(1), 89–105. <https://doi.org/10.1007/s10021-020-00503-w>

Arellano, G., Zuleta, D., & Davies, S. J. (2021). Tree death and damage: A standardized protocol for frequent surveys in tropical forests. *Journal of Vegetation Science*, 32(1), e12981. <https://doi.org/10.1111/jvs.12981>

Beaulieu-Jones, B. K., & Greene, C. S. (2017). Reproducibility of computational workflows is automated using continuous analysis. *Nature Biotechnology*, 35(4), 342–346. <https://doi.org/10.1038/nbt.3780>

Boettiger, C. (2022). Piggyback: Managing larger data on a GitHub repository. Man. version 0.1.4. (pp. 1–11). <https://cran.r-project.org/web/packages/piggyback/piggyback.pdf>

Boettiger, C., & Eddelbuettel, D. (2017). An introduction to rocker: Docker containers for R. *The R Journal*, 9(2), 527–536. <https://doi.org/10.32614/RJ-2017-065>

Clark, J. S., Wolosin, M., Dietze, M., Ibáñez, I., LaDoux, S., Welsh, M., & Kloepfel, B. (2007). Tree growth inference and prediction from diameter censuses and ring widths. *Ecological Applications*, 17(7), 1942–1953. <https://doi.org/10.1890/06-1039.1>

Condit, R. (1998). Tropical Forest census plots: Methods and results from Barro Colorado Island. In *Panama and a comparison with other plots*. Environmental intelligence unit. Springer-Verlag. <https://doi.org/10.1007/978-3-662-03664-8>

Crystal-Ornelas, R., Edwards, B. P., Hébert, K., Hudgins, E. J., Reyes, L. L. S., Scott, E. R., Grainger, M., Foroughirad, V., Binley, A., Brookson, C., Gaynor, K., Sabet, S. S., Güncan, A., Hillemann, F., Weierbach, H., Gomes, D. G. E., & Braga, P. H. P. (2022). Not just for programmers: How GitHub can accelerate collaborative and reproducible research in ecology and evolution. *MetaArXiv*, <https://doi.org/10.31222/osf.io/x3p2q>

Davies, S. J., Abiem, I., Abu Salim, K., Aguilar, S., Allen, D., Alonso, A., Anderson-Teixeira, K., Andrade, A., Arellano, G., Ashton, P. S., Baker, P. J., Baker, M. E., Baltzer, J. L., Basset, Y., Bissengou, P., Bohlman, S., Bourg, N. A., Brockelman, W. Y., Bunyavejchewin, S., ... Zuleta, D. (2021). ForestGEO: Understanding forest diversity and dynamics through a global observatory network. *Biological Conservation*, 253(January), 108907. <https://doi.org/10.1016/j.biocon.2020.108907>

Didham, R. K., Basset, Y., Collins, C. M., Leather, S. R., Littlewood, N. A., Menz, M. H. M., Müller, J., Packer, L., Saunders, M. E., Schönrogge, K., Stewart, A. J. A., Yanoviak, S. P., & Hassall, C. (2020). Interpreting insect declines: Seven challenges and a way forward. *Insect Conservation and Diversity*, 13(2), 103–114. <https://doi.org/10.1111/icad.12408>

Dietze, M. C. (2017). *Ecological forecasting*. Princeton University Press.

Dietze, M. C., Fox, A., Beck-Johnson, L. M., Betancourt, J. L., Hooten, M. B., Jarnevich, C. S., Keitt, T. H., Kenney, M. A., Laney, C. M., Larsen, L. G., Loescher, H. W., Lunch, C. K., Pijanowski, B. C., Randerson, J. T., Read, E. K., Tredennick, A. T., Vargas, R., Weathers, K. C., & White, E. P. (2018). Iterative near-term ecological forecasting: Needs, opportunities, and challenges. *Proceedings of the National Academy of Sciences*, 115(7), 1424–1432. <https://doi.org/10.1073/pnas.1710231115>

Dow, C., Kim, A. Y., D'Orangeville, L., Gonzalez-Akre, E. B., Helcoski, R., Herrmann, V., Harley, G. L., Maxwell, J. T., McGregor, I. R., McShea, W. J., McMahon, S. M., Pederson, N., Tepley, A. J., & Anderson-Teixeira, K. J. (2022). Warm springs alter timing but not total growth of temperate deciduous trees. *Nature*, 608(7923), 552–557. <https://doi.org/10.1038/s41586-022-05092-3>

Ferretti, M. (2009). Quality Assurance in Ecological Monitoring towards a unifying perspective. *Journal of Environmental Monitoring*, 11(4), 726–729. <https://doi.org/10.1039/B902728A>

Fowler, D., Barratt, J., & Walsh, P. (2018). Frictionless data: Making research data quality visible. *International Journal of Digital Curation*, 12(May), 274–285. <https://doi.org/10.2218/ijdc.v12i2.577>

Gonzalez-Akre, E., Meakem, V., Eng, C.-Y., Tepley, A. J., Bourg, N. A., McShea, W., Davies, S. J., & Anderson-Teixeira, K. (2016). Patterns of tree mortality in a temperate deciduous forest derived from a large forest dynamics plot. *Ecosphere*, 7(12), e01595. <https://doi.org/10.1002/ecs2.1595>

Hampton, S. E., Strasser, C. A., Tewksbury, J. J., Gram, W. K., Budden, A. E., Batcheller, A. L., Duke, C. S., & Porter, J. H. (2013). Big data and the future of ecology. *Frontiers in Ecology and the Environment*, 11(3), 156–162. <https://doi.org/10.1890/120103>

Heilman, K. A., Dietze, M. C., Arizpe, A. A., Aragon, J., Gray, A., Shaw, J. D., Finley, A. O., Stefan Klesse, R., DeRose, J., & Evans, M. E. K. (2022). Ecological forecasting of tree growth: Regional fusion of tree-ring and forest inventory data to quantify drivers and characterize uncertainty. *Global Change Biology*, 28, 2442–2460. <https://doi.org/10.1111/gcb.16038>

Herrmann, V., Anderson-Teixeira, K., Kim, A. Y., Calkins, B., Shue, J., Jordan, J., Magee, L., & GitHub Actions Bot. (2021). SCBI-ForestGEO/SCBIMortality: First official release. Zenodo, <https://doi.org/10.5281/zenodo.5551148>

Hester, J. (2021). GitHub actions for the R language. <https://github.com/r-lib/actions>

Hester, J., Csárdi, G., Wickham, H., Chang, W., Morgan, M., & Tenenbaum, D. (2021). Remotes: R package installation from remote repositories, including 'GitHub', version 2.4.2. (pp. 1–21). <https://cran.r-project.org/web/packages/remotes/remotes.pdf>

Kim, A. Y., Anderson-Teixeira, K., Johnson, D., Herrmann, V., Shue, J., & GitHub Actions Bot. (2022). SCBI-ForestGEO/HarvardMortality: First official release. Zenodo, <https://doi.org/10.5281/zenodo.7019885>

Marx, V. (2013). The big challenges of big data. *Nature*, 498(7453), 255–260. <https://doi.org/10.1038/498255a>

Masson-Delmotte, V., Zhai, P., Connors, S. L., Péan, C., Berger, S., Caud, N., Chen, Y., Goldfarb, L., Gomis, M. I., Huang, M., Leitzell, K., Lonnoy, E., Matthews, J. B. R., Maycock, T. K., Waterfield, T., Yelekçi, O., Yu, R., & Zhou, B. (2021). IPCC, 2021: Climate change 2021: The physical science basis. In *Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press.

McDowell, N. G., Allen, C. D., Anderson-Teixeira, K., Aukema, B. H., Bond-Lamberty, B., Chini, L., Clark, J. S., Dietze, M., Grossiord, C., Hanbury-Brown, A., Hurtt, G. C., Jackson, R. B., Johnson, D. J., Kueppers, L., Lichstein, J. W., Ogle, K., Poulter, B., Pugh, T. A. M., Seidl, R., ... Xu, C. (2020). Pervasive shifts in Forest dynamics

in a changing world. *Science*, 368(6494), eaaz9463. <https://doi.org/10.1126/science.aaz9463>

McGregor, I., Kim, A. Y., Anderson-Teixeira, K., Gonzalez-Akre, E., Dow, C., Terrell, A., Shue, J., Jordan, J., Calkins, B., Herrmann, V., & GitHub Actions Bot. (2021). SCBI-ForestGEO/dendrobands: First official release. Zenodo, <https://doi.org/10.5281/zenodo.5551144>

McNutt, M. (2014). Reproducibility. *Science*, 343, 229. <https://doi.org/10.1126/science.1250475>

Merkel, D. (2014). Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux Journal*, 2014(239), 2.

Michener, W. K. (2018). Quality assurance and quality control (QA/QC). In F. Recknagel & W. K. Michener (Eds.), *Ecological informatics: Data management and knowledge discovery* (pp. 55–70). Springer International Publishing. https://doi.org/10.1007/978-3-319-59928-1_4

Milcu, A., Puga-Freitas, R., Ellison, A. M., Blouin, M., Scheu, S., Freschet, G. T., Rose, L., Barot, S., Cesarz, S., Eisenhauer, N., Girin, T., Assandri, D., Bonkowski, M., Buchmann, N., Butenschoen, O., Devidal, S., Gleixner, G., Gessler, A., Gigon, A., ... Roy, J. (2018). Genotypic variability enhances the reproducibility of an ecological study. *Nature Ecology & Evolution*, 2(2), 279–287. <https://doi.org/10.1038/s4159-017-0434-x>

Muller-Landau, H. C., & Dong, S. X. (2010). Metal band dendrometer protocol. CTFS Global Forest Carbon Research Initiative.

Peng, R. D. (2011). Reproducible research in computational science. *Science*, 334, 1226–1227. <https://doi.org/10.1126/science.1213847>

Powers, S. M., & Hampton, S. E. (2019). Open Science, reproducibility, and transparency in ecology. *Ecological Applications*, 29(1), e01822. <https://doi.org/10.1002/eaap.1822>

Richardson, N., Cook, I., Crane, N., Dunnington, D., François, R., Keane, J., Moldovan-Grünfeld, D., Ooms, J., & Arrow, A. (2022). Arrow: Integration to 'Apache' 'Arrow'. Manual.

Schliep, E. M., Dong, T. Q., Gelfand, A. E., & Li, F. (2014). Modeling individual tree growth by fusing diameter tape and increment Core data. *Environmetrics*, 25(8), 610–620. <https://doi.org/10.1002/env.2324>

Stodden, V., McNutt, M., Bailey, D. H., Deelman, E., Gil, Y., Hanson, B., Heroux, M. A., Ioannidis, J. P. A., & Taufer, M. (2016). Enhancing reproducibility for computational methods. *Science*, 354(6317), 1240–1241. <https://doi.org/10.1126/science.aah6168>

Stodden, V., & Miguez, S. (2014). Best practices for computational science: Software infrastructure and environments for reproducible and extensible research. *Journal of Open Research Software*, 2(1), e21. <https://doi.org/10.5334/jors.ay>

Ushey, K. (2022). Renv: Project environments, version 0.15.5. (pp. 1–65). <https://cran.r-project.org/web/packages/renv/renv.pdf>

Ushey, K., McPherson, J., Cheng, J., Atkins, A., & Allaire, J. J. (2021). Packrat: A dependency management system for projects and their R package dependencies, version 0.8.1. (pp. 1–23). <https://cran.r-project.org/web/packages/packrat/packrat.pdf>

van der Loo, M. P. J., & de Jonge, E. (2021). Data validation infrastructure for R. *Journal of Statistical Software*, 97(10), 1–31. <https://doi.org/10.18637/jss.v097.i10>

Vitousek, P. M., Mooney, H. A., Lubchenco, J., & Melillo, J. M. (2008). Human domination of Earth's ecosystems. In J. M. Marzluff, E. Shulenberger, W. Endlicher, M. Alberti, G. Bradley, C. Ryan, U. Simon, & C. ZumBrunnen (Eds.), *Urban ecology: An international perspective on the interaction between humans and nature* (pp. 3–13). Springer US. https://doi.org/10.1007/978-0-387-73412-5_1

Welti, E. A. R., Joern, A., Ellison, A. M., Lightfoot, D. C., Record, S., Rodenhouse, N., Stanley, E. H., & Kaspari, M. (2021). Studies of insect temporal trends must account for the complex sampling histories inherent to many long-term monitoring efforts. *Nature Ecology & Evolution*, 5(5), 589–591. <https://doi.org/10.1038/s41559-021-01424-0>

White, E. P., Yenni, G. M., Taylor, S. D., Christensen, E. M., Bledsoe, E. K., Simonis, J. L., & Morgan Ernest, S. K. (2019). Developing an automated iterative near-term forecasting system for an ecological study. *Methods in Ecology and Evolution*, 10(3), 332–344. <https://doi.org/10.1111/2041-210X.13104>

Wickham, H. (2011). Testthat: Get started with testing. *The R Journal*, 3(1), 5. <https://doi.org/10.32614/RJ-2011-002>

Wickham, H., & Luraschi, J. (2021). Pins: Pin, discover and share resources. Manual.

Wilson, G., Aruliah, D. A., Brown, C. T., Hong, N. P. C., Davis, M., Guy, R. T., Haddock, S. H. D., Huff, K. D., Mitchell, I. M., Plumley, M. D., Waugh, B., White, E. P., & Wilson, P. (2014). Best practices for scientific computing. *PLoS Biology*, 12(1), e1001745. <https://doi.org/10.1371/journal.pbio.1001745>

Wolkovich, E. M., Regetz, J., & O'Connor, M. I. (2012). Advances in global change research require open science by individual researchers. *Global Change Biology*, 18(7), 2102–2110. <https://doi.org/10.1111/j.1365-2486.2012.02693.x>

Yenni, G. M., Christensen, E. M., Bledsoe, E. K., Supp, S. R., Diaz, R. M., White, E. P., & Morgan Ernest, S. K. (2019). Developing a modern data workflow for regularly updated data. *PLoS Biology*, 17(1), e3000125. <https://doi.org/10.1371/journal.pbio.3000125>

SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Kim, A. Y., Herrmann, V., Barreto, R., Calkins, B., Gonzalez-Akre, E., Johnson, D. J., Jordan, J. A., Magee, L., McGregor, I. R., Montero, N., Novak, K., Rogers, T., Shue, J., & Anderson-Teixeira, K. J. (2022). Implementing GitHub Actions continuous integration to reduce error rates in ecological data collection. *Methods in Ecology and Evolution*, 13, 2572–2585. <https://doi.org/10.1111/2041-210X.13982>