Repository Approaches to Improving the Quality of Shared Data and Code

Citation	Trisovic, Ana, Katherine Mika, Ceilyn Boyd, Sebastian Feger, Merce Crosas. "Repository Approaches to Improving the Quality of Shared Data and Code." Data 6, no. 2 (2021): 15. DOI: 10.3390/data6020015
Citable link	https://nrs.harvard.edu/URN-3:HUL.INSTREPOS:37367589
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

data

MDPI

Article

Repository Approaches to Improving the Quality of Shared Data and Code

Ana Trisovic [1,*], Katherine Mika [1], Ceilyn Boyd [1], Sebastian Feger [2,3] and Mercè Crosas [1]

[1] Institute for Quantitative Social Science, Harvard University, 1737 Cambridge St, Cambridge, MA 02138, USA; kmika@g.harvard.edu (K.M.); ceilyn_boyd@harvard.edu (C.B.); mcrosas@g.harvard.edu (M.C.)
[2] European Organization for Nuclear Research (CERN), 1, Esplanade des Particules, CH-1217 Meyrin, Switzerland; sebastian.feger@ifi.lmu.de
[3] LMU Munich, 1, Geschwister-Scholl-Platz, 80539 Munich, Germany
* Correspondence: anatrisovic@g.harvard.edu

Abstract: Sharing data and code for reuse has become increasingly important in scientific work over the past decade. However, in practice, shared data and code may be unusable, or published results obtained from them may be irreproducible. Data repository features and services contribute significantly to the quality, longevity, and reusability of datasets. This paper presents a combination of original and secondary data analysis studies focusing on computational reproducibility, data curation, and gamified design elements that can be employed to indicate and improve the quality of shared data and code. The findings of these studies are sorted into three approaches that can be valuable to data repositories, archives, and other research dissemination platforms.

Keywords: data quality; data repository; digital libraries; data curation; fair principles; open data; open code; gamification

check for updates

Citation: Trisovic, A.; Mika, K.; Boyd, C.; Feger, S.; Crosas, M. Repository Approaches to Improving the Quality of Shared Data and Code. *Data* **2021**, *6*, 15. https://doi.org/10.3390/data6020015

Academic Editor: Maurizio Lenzerini
Received: 22 December 2020
Accepted: 28 January 2021
Published: 3 February 2021

1. Introduction

Research data, defined as collected or generated information used as evidence for original research findings [1], have become a vital component of the scholarly record and a primary asset of new inquiry. The increased value of scientific data and concerns over a reproducibility crisis [2–4] have led funders and journals to require data sharing as conditions of grant funding and publication. Data repositories are considered a primary venue for data sharing as they implement systematic stewardship to foster curation, dissemination, access, and preservation of research data [5–7]. However, published data will only be reused if a researcher trusts in its quality [8].

Data quality is a broad term that includes many elements and can be defined from many different perspectives. In 2015 Cai & Zhu developed a framework for evaluating data quality along five axes: availability, usability, reliability, relevance, and presentation quality [9]. While some of the dimensions defined in their framework largely refer to intrinsic qualities of data files such as accuracy, integrity, and completeness, several important features can be improved by data repositories. These features include presentation quality, documentation, metadata, and accessibility, which contribute to the overall quality of a published dataset. Similarly, Martin et al. enumerate a series of data quality properties that emphasizes the importance of intrinsic data quality, contextual data quality, metadata quality, characteristics of data and data users, platform promotion and user training [10]; many of which are within the influence of data repositories. Table 1 shows an approximate alignment of these properties described in Refs. [9,10], together with examples of typical data repository features and functionalities. This paper presents three approaches, or categories of repository feature enhancements, beyond those highlighted in Table 1 that repository staff can apply to improve the overall quality of data published on their platforms.

Table 1. An approximate alignment of data quality properties described in Refs. [9,10], together with typical data repository features and functionalities.

Cai & Zhu	**Martin et al.**	**Examples of Common Data Repository Features**
Availability: accessibility, timeliness, authorization	Accessibility, timeliness, representational consistency, visibility, platform functionality	Capturing data citation information, minting DOIs
Usability: documentation, credibility, metadata	Intended use, subject matter expertise, technical skills, metadata quality (standards & consistency); learnability, believability & reputation, confidentiality, etc.	Supporting documentation, reuse licensing, terms of access/restrictions
Reliability: accuracy, integrity, consistency, completeness, auditability	Data accuracy, validity, reliability, completeness, missing data, timing and frequency, collection methods, format & layout, sample size & method, representation, study design, unit of analysis, etc.	Metadata standards, variable level metadata support
Relevance: fitness	Relevancy, value added	Reuse metrics, granular description
Presentation quality: readability & structure	Concise representation, ease of understanding, ease of manipulation, user-friendliness	Preview options, UI/UX reviews
	Platform promotion and user training: availability of information, capacity to respond to feedback, financial resources, legal protections and interpretations, platform training and promotion, policies and regulation, political support for developing and releasing data	Support services, preservation policies, governance, and legal policies

Data repositories are designed to meet the needs of different scientific communities, and as such, can be broadly classified as either domain-specific or general-purpose. The former is located within domain communities, such as physics (CERN Analysis Preservation (CAP)) or genetics (GenBank). The latter emerged with the increased demand for data repositories to support the "long-tail" of science, where a large number of relatively small labs and individual researchers collectively produce the majority of results [5,11,12]. Examples of generalist repositories are Harvard Dataverse, Figshare, Dryad, and Zenodo.

The heterogeneous nature of collected data contributes to this long-tail effect [13,14], and creates a need for versatile data repositories such as the Harvard Dataverse repository. Harvard Dataverse is a multi-disciplinary research data repository that allows members of the worldwide scientific community to deposit, publish, and share their datasets. The repository infrastructure supports various file formats and requires that depositors provide citation-level metadata, including a dataset title, author, and date. Additional features, including support for subject-specific metadata built on community and domain standards, file versioning, persistent object identifiers, and custom rights statements, contribute to the quality of published data by making it easier to discover, understand, and reuse them. There are more than 60 independent Dataverse repository installations worldwide at the time of writing this article. Each installation runs a version of the open-source Dataverse software platform developed and maintained by Harvard's Institute for Quantitative Social Sciences (IQSS) and open source contributors.

Domain-specific repositories are often required for sharing large or very complex data that generalist repositories may not have the infrastructure, specialized curatorial skills, or domain expertise to support. Optimizations for data description, file formats, storage, and exploration features are not feasible or necessary for generalist repositories to support a heterogeneous collection. Large-scale "Big Science" datasets (measured in terabytes and petabytes) are often produced by large coordinated teams with extensive instrumentation and are designed to be shared with many researchers for a variety of purposes and projects. CERN Analysis Preservation (CAP) is a good example of such a domain-tailored repository that offers specialized research data management services [15]. The platform maps research workflows of its four largest experiments in customized analysis description and submission templates, thereby easing and supporting documentation, sharing, and reusing research conducted within those experiments.

One of the major benefits of publishing research datasets in a domain repository is the built-in designated community of users who understand jargon, descriptions and metadata, collection protocols, and potential errors or flaws in a given dataset. Users looking for data within a specific domain can better evaluate a dataset described with precise domain metadata than those described in general terms or with specific descriptions from an unfamiliar field. Nevertheless, communities and audiences are porous and often ill-defined. As research becomes increasingly interdisciplinary, researchers establish their agendas at the nexus of multiple disciplines and communities. Communities of practice are formed around data, but also software packages, methodologies (i.e., computational, experimental, theoretical), geographic regions, and more [16]. Therefore, it is important to consider more intersectional research communities and interdisciplinary standards to enable data reuse in a larger variety of research contexts.

Published research data are routinely used by these diverse communities for calibration, control, comparison, testing, and conducting meta-analyses [17], but they are not always well-documented, understandable, and reusable. A lack of data-sharing conventions, incentives, and infrastructure to support publishing for long-tail, heterogeneous data, have historically affected the quality of data as a research output. [11,18–20]. Additionally, irreproducibility of published results is often caused by missing files or documentation and has increased the importance of transparent and available research datasets, which recently have come to include data, code, documentation, and other Supplementary Files [2–4,21].

This paper addresses the following questions: how can data repositories improve data and code quality, and how can they signal data and code quality to external researchers? While there are important elements of data quality that repositories cannot affect, we focus on specific features of published research datasets: including data, code, metadata, documentation, and their presentation that data repositories can identify, strengthen, and highlight. Improving and signaling the quality of research data along the axes identified by Cai & Zhu and Martin, et al. by enhancing data curation, code completeness, and data publishing incentives in data repositories will contribute to the transparency, reproducibility, and reuse of research products.

2. Approaches for Advancing Dataset Quality

We present three approaches to improve and signal the quality of published datasets in research data repositories based on a combination of original and secondary studies from computer science, information science, and human-computer interaction. In this paper we use the term "approach" to collect a set of activities into a general strategy designed to improve the quality of a dataset. We analyze data from these studies to evaluate effects on data and code quality. These approaches are designed to support data repository managers and curators in identifying and effectively stewarding high-quality datasets. In particular, we explore applying our proposed approaches to Dataverse repositories.

2.1. *Ensure Research Code Completeness*

Shared research code is increasingly a common element of many datasets, and it should be comprehensible, executable, and reusable to be of high quality. However, disseminating such code can be complex as it is often written for a specific environment, like software, operating system, or hardware, meaning that it will not execute unless all required dependencies are available. Even a small change in these dependencies can sometimes result in errors or discrepancies in the execution outputs. Computing methods and artifacts are often not sufficiently documented in data repositories, which later hinders reproducibility and reuse.

To illustrate the challenge of code re-execution that is necessary for reproducibility and reuse, we conducted an original study in which we re-execute Python code files from Harvard Dataverse [22]. We successfully retrieved 92 publicly available replication datasets that contain Python files.[1] The re-execution study was executed in a clean anaconda environment in Docker containers running Debian GNU/Linux 10 in the following steps:

1. We look for files such as "requirements.txt" or "environment.yml" inside the dataset because these filenames are common conventions for documenting needed code dependencies for Python. If such files were not found, we scan the Python code looking for the used libraries, and create a new requirements file. We attempt to install all libraries from the requirements file.
2. We automatically (naively) re-execute the Python files first with Python 2.7 and then with Python 3.5 with a time limit of 10 minutes per each Python file. If the file executes successfully in the allocated time, we record a success; if it crashes with error, we record the error; and if it exceeds the allocated time, we record 'time limit exceeded' (TLE) or null result (which are ignored in the success analysis as we cannot be certain whether the file would successfully execute or not).

Our results show that about 27% of the files (102 out of 379) are re-executable using either Python 2.7 or Python 3.5. The success rate with each of the versions independently is lower than the combined result (see Figure 1), showcasing the importance of reusing the code with the right software version. In particular, it is likely that older code was more compatible with Python 2.7 and recent with Python 3.5. In that vein, we observe that the most common errors in Python 3.5 execution were Syntax Error (missing parentheses in print or invalid syntax) that appeared in 110 cases or 28%, and Import Error. The most common errors in Python 2.7 execution were Import Error (unavailable library) and Syntax Errors. The type of errors further emphasizes the differences between the two Python versions (notably syntax) and the importance of recording it in the preserved code.

The observed high rate of Import Errors attests to how hard it is to reconstruct a working runtime environment even when all used libraries are pre-identified. This is because the version of the used libraries is essential for code re-execution. We observe a significantly higher re-execution success rate of 38% (17 out of 44 files) in datasets where the requirements file (likely containing the library versions) was present. However, files that could accurately reconstruct the runtime environments (such as environment.yml, requirements.txt, and Dockerfile) were rarely present (6 out of 92, 6%).

Another possible explanation for the low re-execution rate might be due to the order in which Python files should be executed. Sometimes, each Python file in a dataset recreates one published result or figure, and it can be executed independently. However, in some cases, it is necessary to execute multiple Python files in a sequence to obtain the right result. In our study, the files were executed in a random order, which would favor the first type of datasets. We aggregate the obtained results and label a dataset successful if at least one Python file executes with success. The success rate of about 44% signals that likely some of the Python files in the datasets were meant to be executed in a sequence. It also shows that about 56% of the datasets do not contain a single Python file that is easily re-executable. This result points to the lack of code support in data repositories, the existence of common code errors (like fixed paths), or the need for another version of Python.

[1] There were additional 15 datasets that contained Python files and were visible through the API but could not be retrieved due to restricted authorization or connection error.

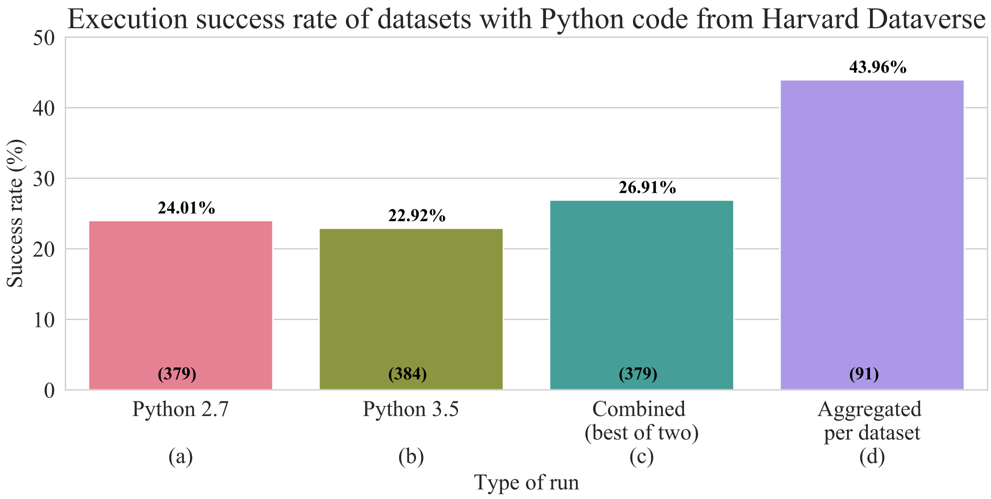


Figure 1. Python re-execution results. The upper number in the bars is the success rate and the lower number is the total file count. Total file counts differ because "time limit exceeded (TLE)" results were ignored as we cannot be certain whether the file would successfully execute or not. (**a**) Re-execution rate for Python 2.7. (**b**) Re-execution rate for Python 3.5. (**c**) Re-execution rate with both Python 2.7 and 3.5 (i.e., success is recorded if it runs with either of the versions) (**d**) Success rate aggregated per dataset, i.e., 43.96% of the datasets have at least one Python file that runs (or conversely 56.04% of the datasets have no Python files that executed successfully in this study.)

Finally, we examined other code quality indicators, like documentation, within the replication dataset. Data and code are likely to be more understandable and reusable if a user-friendly instructions file is available. We can observe that 57 out of 92 (62%) replication datasets contain a README, codebook, or an instructions file (Table 2). Half of the datasets (46 out of 92) contained code in other programming languages (Stata, R, Java, C++, Matlab, SAS, Ruby), and we observe a higher re-execution rate in datasets containing only Python code (68 out of 196 files or about 35%), than in those containing files in other languages (34 out of 183 or about 18%). In this case, Python code might depend on the output of the code in other programming languages, which may explain the difference in the re-execution rates. The average number of files in a replication dataset is 43, and the average dataset size was 248 MB.

Table 2. Presence of environment-capturing files in datasets.

File	Count (Out of 92)
environment.yml	0
requirement.txt	6
Dockerfile	0
README, instructions or codebook	57

Our Python study shows that further support is needed to adequately document research code and their computing environments when publishing them in data repositories. Several approaches could help facilitate reproducibility of research code. First, we observe an increase in re-execution rate if a requirements file is present in the datasets. Therefore it would be helpful to encourage storing such a file to capture needed dependencies. Second, we observe that a documentation file is sometimes missing, which could be improved in the repository. In practice, data repositories could support depositing these files (like requirements, environment, readme, codebook, and others) either through the User Guide or a pop-up window if they detect Python files. Finally, reproducibility could be achieved by using virtual containers that were deemed irrepressible for capturing the runtime environment. Reproducibility platforms such as Code Ocean, Whole Tale, or Renku natively provide research portability through virtual containers and the cloud. These platforms automatically capture the runtime environment and often facilitate code automation. These proposed approaches are being considered by the Dataverse open-source community, and there is already ongoing work in the Dataverse software project that aims to capture virtual containers through integration with these reproducibility platforms [23].

2.2. Encourage Use of Curation Features and Pre-Submission Dataset Review

Though anyone can deposit data in an unmediated, self-service fashion to their Dataverse collection in Harvard Dataverse, groups such as journals, laboratories, and project teams often restrict who can contribute to their collections and actively curate new datasets deposited to those collections. The Dataverse software supports pre- and post-publication data curation workflows that allow curators to ensure that deposited datasets meet group-defined expectations for characteristics such as metadata completeness, approved file formats, or accompanying code or documentation. A number of academic journals perform reproducibility verification through the curation workflow. For instance, the American Journal of Political Science (AJPS) requires their authors to provide all necessary research materials for verification and research reproducibility.[2] Upon a paper's acceptance, the research datasets are reviewed to confirm that they produce the reported analytic results before they are published at the AJPS collection at Harvard Dataverse.

The Harvard Dataverse features for data curation can be classified into three categories (Table 3) for convenience of discussion. The Dataverse platform automatically enforces a baseline of curation (Category I) through features such as required metadata fields to support data citations and smart defaults for components like data use agreements. Dataverse tools do not, however, automatically inspect the contents of data files to, for instance, confirm that data values are valid or are not missing, leaving that responsibility to data depositors. Data depositors may also use optional features to improve data curation quality from basic to Category II, which includes the use of custom metadata blocks, dataset versioning, and supporting documentation. The use of managed data curation processes, together with the reputation and reliability of a repository, can influence researchers' perception of data quality [24]. Therefore, the extent to which depositors and data curators use optional repository features can be considered an indicator of the overall quality of a dataset. The more extensive the use of optional features, the more FAIR (Findable, Accessible, Interoperable, and Reusable) [25] the dataset is likely to be. Category III data curation requires that the dataset contents be inspected either manually or using software tools to ensure that they meet subject-area standards for data sharing and reuse, as demonstrated by the AJPS data curation workflow mentioned above.

To learn about the impact of review and curation services, we conduct an analysis of previously collected data [26], that captures the presence of the following characteristics [27] in the Harvard Dataverse datasets:

1. Optional metadata blocks. A well-curated dataset should have at least one optional metadata block to support its discoverability and reuse.
2. Keywords. A well-curated dataset should also have at least one keyword.
3. Description. A well-curated dataset should have a description. Like keywords, descriptions help to facilitate its discovery and reuse.
4. Open file formats. A well-curated dataset should use open file formats, where possible.
5. Discipline standard file formats. Not all disciplines use open standards, but at minimum, datasets should adhere to best practices for discipline file formats.
6. Supplemental Files. A well-curated dataset should have either a codebook or a readme file that provides insight into the datasets' internals, such as descriptions of its variables.
7. Submission review. A well-curated dataset may undergo an additional review by the collection owner prior to publication. In contrast to the previous six, this characteristic might be considered a direct indicator of dataset quality [24].

[2] AJPS Data Policy: https://ajps.org/ajps-verification-policy/.

Table 3. Summary of the characteristics associated with the three Dataverse data curation categories.

Category I: Basic	**Category II: Enhanced**	**Category III: Comprehensive**
Default curation support supplied or enforced by the Dataverse software	Optional curation support provided by Dataverse software	Research-dependent data quality characteristics
E.g., Default rights statements, Required metadata fields, Reliable storage and access, Persistent identifiers (DOIs), Data citations	E.g., File versioning, Optional keywords, Optional description, Optional metadata blocks, Optional rights statements, Optional supporting documentation	E.g., Comprehensible variable names, Confirmed valid data values, Up-to-date codebook, Well-documented code

The presence of these characteristics suggests that a dataset depositor or curator has taken additional steps to facilitate its sharing and reuse, and therefore it indirectly signals that the dataset may be more FAIR, has higher dataset quality, and greater fitness for use. For instance, we observe that most datasets had a text description (n = 24,661, 84.2%), though this field became mandatory in 2015, leading to a drastic increase in its use of 99% and 100% in the subsequent years. The keywords field remains optional, explaining why the number of datasets with keywords (n = 14,593, 49.8%) is close to those without keywords (n = 14,702, 50.2%). A summary of all data characteristics is shown in Table 4.

Table 4. Harvard Dataverse dataset characteristics between 2007 and October 2019.

Characteristic	**Value (n)**	**% of Total**
Total published datasets (N)	29,295	100%
Total file count in datasets	383,685	100%
Contain optional metadata blocks	8380	28.6%
Contain keywords	14,593	49.8%
Contain description	24,661	84.2%
Dataset linked to a publication	6742	23%
Required review before before publishing	25,938	89%
Affiliation		
- Associated with groups	15,368	52.5%
- Associated with individuals	6125	20.9%
- Uncategorized	7802	26.6%

By examining the datasets that had a prior review, we find lengthier descriptions, higher keywords count, increased number of versions, and higher use of optional metadata than in datasets released without review (Figure 2). In prior review (or submission review), the collection owner or manager may inspect datasets' metadata for completeness, ensure that Supplementary Materials, data files, and code adhere to best practices, or assess how well the datasets meet other established publication criteria. For example, both metadata records and descriptive data fields [28] are essential inputs to indexers and web crawlers used by search engines like Google's Dataset Search to make data discoverable across domains. Therefore, our result suggests that, on average, a prior review effectively improved the curation quality of a deposited dataset.

Though only 23% of the datasets were linked to a publication (had the "related publication" field), datasets with prior review were again better performing than the rest.[3] A paper publication is often seen as primary documentation for open data and as "the official version of record, as officially peer-reviewed and published, that will explain background, context, methodology, and possibilities for further analysis in the best possible way, and express the intentions of the person who helped collect the data" [29]. Without the

[3] In practice, a publication reference is often placed in the dataset description field.

unstructured but necessary context provided by the literature, researchers may reject data rather than risk misinterpretation [30]. Therefore, a reference to the original publication is essential for external researchers when evaluating data reuse.

We find that data depositors often do not adequately document their datasets. Prior review and mandatory fields can improve the quality of curation and, thus, likely the quality of deposited datasets. Though dataset description, keywords, and optional metadata are not ubiquitously used in Harvard Dataverse, this could also be improved with curation review. Harvard Dataverse provides advanced curation services (shown in the three categories) and publication curation workflows that other repositories can look up to. It is important to establish a curation baseline, to ensure that all published datasets comply with a certain quality standard.

Finally, we find that articles linked to published data often include contextual information that metadata cannot sufficiently capture and transmit. Also, academic literature remains the primary avenue through which researchers find and evaluate secondary data [16,29]. By building adequate citation infrastructure, repositories can encourage bidirectional linking between publications and datasets to facilitate direct access between them. Therefore, citing datasets across the scholarly record makes them both more findable and better documented.

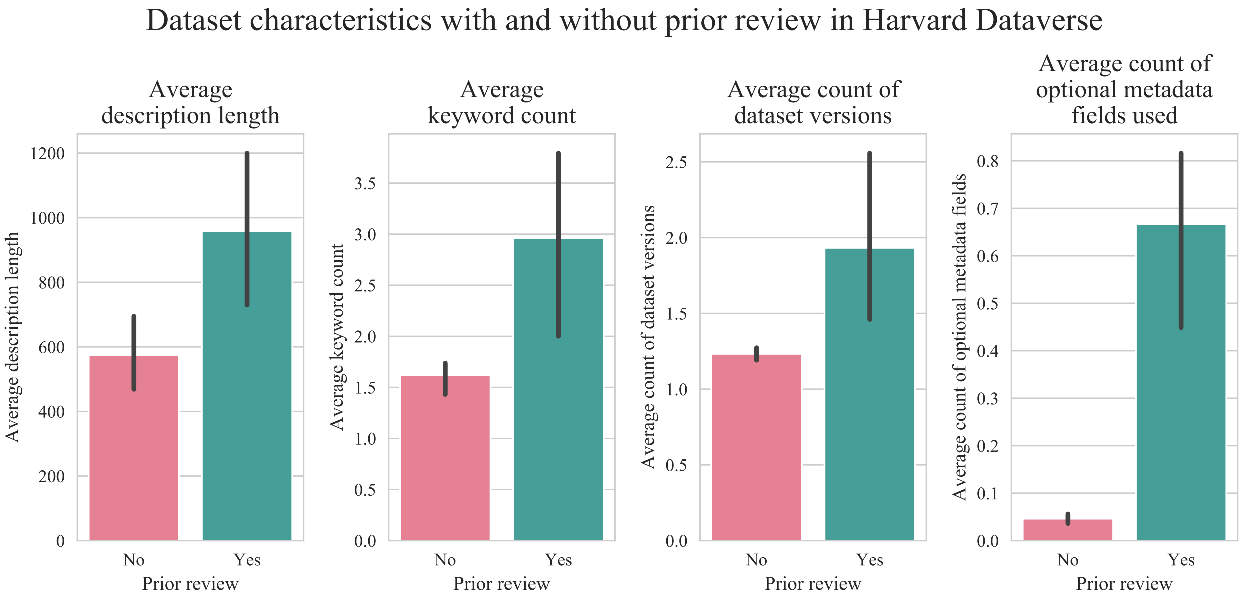
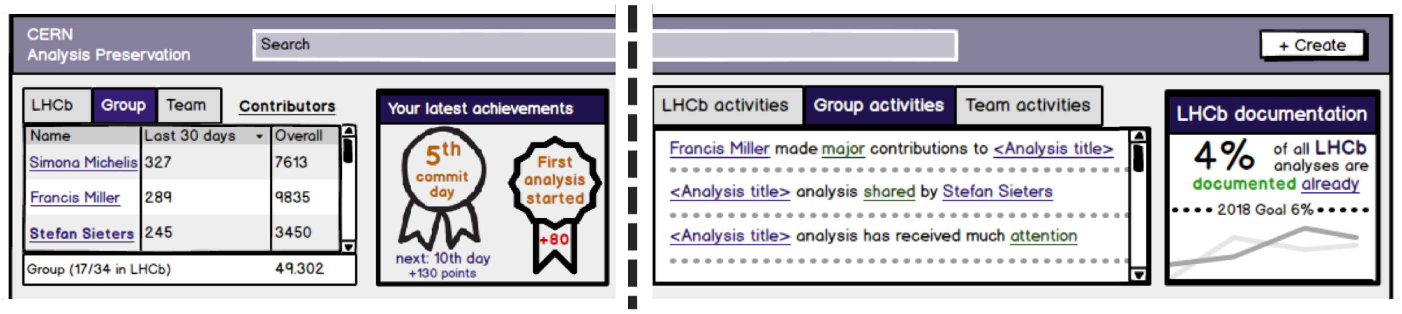


Figure 2. Presence of dataset characteristics with and without prior submission review in Harvard Dataverse. Data is from 2015 onward to reflect changes in making fields mandatory.

2.3. *Incorporate Gamified Design Elements*

Gamification, defined as the use of game design elements in non-game contexts [31], is a promising approach that can be used to improve data sharing and signal high-quality data. Badges, points, and leaderboards are some of the most common game design elements [32]. They are used to motivate actions and behaviors across a wide range of domains, from health applications [33] to work environments [34]. Gamification in science was traditionally used in teaching [35] and in citizen science [36,37]. However, when the gamified design is implemented in concert with researchers' values and interests, it could be a powerful tool in encouraging open science practices such as dataset sharing [38,39].

To showcase the potential of gamification on quality of shared research data, we conduct a secondary interpretation of published results. A study at CERN [40], carried out to drive the design of the CAP portal, investigated how scientists perceive the use of various gamification elements. Two interactive prototypes were designed for the study (Figure 3), one (Simple Game Elements Design) making use of the most common game design elements, including points and leaderboards, and the other (Rational-Informative Design) focusing on communication (i.e., group activities log), resource sharing, and providing an overall research dataset management status of the group. The study found that some of the gaming elements were more desirable than others. In particular, several participants opposed the use of leaderboards, as they could encourage comparisons and competition. The gamified badges were identified as the most suitable elements to incentivize dataset sharing. Such a result is corroborated by the successful use of the Open Science Badges (OSB)[4] that proved to incentivize data sharing for submissions to a medical and health journal [41]. Rowhani-Farid et al. [42] even concluded, based on their systematic review in the health and medical domain, that OSB is the "only one evidence-based incentive to promote data sharing." Therefore, gamified badges allow promoting best practices that are considered highly important in the community while still providing attainable goals for the authors. They represent an incentive for researchers as the papers with such rewards (badges) might have improved visibility and higher citation rate.



Name	Last 30 days	Overall
Simona Michelis	327	7613
Francis Miller	289	9835
Stefan Sieters	245	3450
Group (17/34 in LHCb)		49.302

Francis Miller made major contributions to <Analysis title>

<Analysis title> analysis shared by Stefan Sieters

<Analysis title> analysis has received much attention

Figure 3. Dashboards of the two gamified research dataset management prototypes (left: Simple Game Elements Design, right: Rational-Informative Design) that make use of contrasting design strategies [40].

Game design elements such as badges motivate research dissemination by providing recognition to the authors, but they can also be used to identify resources of high quality within an available resource pool. An example of this type of gamification application is the GitHub repository star system, where software developers "star" a repository if they find it valuable, or in contrast, assess its quality based on the number of existing stars [43]. A similar design element could be implemented in data repositories that would provide a more nuanced peer assessment of a dataset. For example, such an element would allow a dataset to be rated 'novel', 'educational', 'fundamental', or similar.

In a similar vein, Harvard Dataverse displays a number of downloads for each dataset, which may appear as a comparable peer-enabled reuse metric. Reuse, however, is a flawed method for assessing the quality of a dataset. A high number of downloads signals the popularity of a dataset and may seem to confer high scientific, educational, or reuse value. Data products from Big Science endeavors are significant investments and are intended to serve a broad audience, yet small datasets that are much more common may only be reused by a few specialists. Download and reuse counts do not reliably measure value or quality for a given dataset. Improving the reusability of data improves its quality, but reuse metrics cannot give a complete picture of the quality or value of an individual dataset. This problem will be improved when the data repository community starts using reliable and standard counts of data citations for their datasets. This effort is being addressed by the Make Data Count project, including collaboration with DataCite and CrossRef. [5] But currently, there is no yet widespread scholarly practice of using data citations in published articles, so data citation counts do not yet reflect how a dataset is reused.

In addition to providing data citation counts, data repositories could improve the quality of datasets by implementing gamification elements developed to create incentives

[4] https://www.cos.io/badges.

[5] https://makedatacount.org and https://datacite.org.

for researchers to be more open and thorough when sharing data. Through our secondary study, we find that scientific badges have high potential, as they not only motivate the author to obtain them but also present a positive signal, or quality indicator, to external researchers looking to reuse data. Finally, elements that allow peer assessment, such as the ′star′ system on GitHub, can also be viewed as a resource quality indicator. Such a system could be employed at a later stage after the resource is published, as it does not require direct input from the original authors.

3. Conclusions

Data repository features and services can contribute significantly to the quality and reusability of shared datasets. They may also advertise datasets to multiple communities through various quality indicators proposed in this paper. The three presented approaches for data repositories are based on three different studies that provide guidance for how datasets may be improved. Runtime environment components for code can be encouraged by the repository infrastructure to improve research reproducibility. Repositories can support a deposit workflow with prior review of dataset submissions, which we have shown often results in better-curated data. Finally, including gamification elements like badges and peer-assessments in a repository system promote data sharing by providing recognition for authors and useful metrics for data reusers. When authors are incentivised to share data in a repository and are held accountable for its quality through open metrics and peer-evaluation, the resulting data products are often better quality.

We defined data quality along a number of axes, highlighting the importance of both intrinsic elements and features that data repositories can affect. Each study investigated a suite of strategies, combined into three more general "approaches," in order to determine whether the activities impacted the overall dataset quality. The approaches discussed identify three categories of repository features and services that improve the overall quality of a published dataset: code reproducibility, data curation, and quality incentives. Developing strategies to implement aspects of these approaches depends on various repository constraints and community needs, but each likely contributes to improved dataset quality. As data repositories and data sharing practices continue to evolve, the connection between data quality and repository infrastructure presents significant possibilities for further research.

Author Contributions: All authors have contributed to the reported work. The contributions are sorted into the following roles according to the CRediT taxonomy: conceptualization, A.T.; methodology, A.T., K.M., C.B. and S.F.; software, A.T.; formal analysis, A.T., C.B. and S.F.; investigation, K.M.; data curation, A.T.; writing—original draft preparation, A.T., K.M., C.B. and S.F.; writing—review and editing, A.T., K.M., C.B., S.F. and M.C.; visualization, A.T.; supervision, M.C.; project administration, M.C.; funding acquisition, M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially funded by the Sloan Foundation (award number G-2018-11111). Ana Trisovic is funded by the Sloan Foundation (award number P-2020-13988). Ana Trisovic acknowledges a micro-grant award by the Council on Library and Information Resources (CLIR). Sebastian Feger acknowledges CERN Doctoral Student program.

Data Availability Statement: Data and code used in the original study described in this paper are available at Harvard Dataverse at 10.7910/DVN/EA3LC5 under the license CC0.

Acknowledgments: Thank you to the Dataverse and CAP teams for their help and support.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Borgman, C.L. *Big Data, little Data, No Data: Scholarship in the Networked World*; MIT Press: Cambridge, MA, USA, 2015.
2. Baker, M. 1500 scientists lift the lid on reproducibility. *Nature* **2016**, *533*, 452–454. [CrossRef] [PubMed]

3. Stodden, V.; Seiler, J.; Ma, Z. An empirical analysis of journal policy effectiveness for computational reproducibility. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 2584–2589. [CrossRef] [PubMed]
4. Pimentel, J.F.; Murta, L.; Braganholo, V.; Freire, J. A large-scale study about quality and reproducibility of jupyter notebooks. In Proceedings of the 2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR), Montreal, QC, Canada, 25–31 May 2019; pp. 507–517.
5. Assante, M.; Candela, L.; Castelli, D.; Tani, A. Are Scientific Data Repositories Coping with Research Data Publishing? *Data Sci. J.* **2016**, *15*, 6. [CrossRef]
6. Crosas, M. The Dataverse Network®: An Open-Source Application for Sharing, Discovering and Preserving Data. *D-Lib Mag.* **2011**, *17*. [CrossRef]
7. King, G. An Introduction to the Dataverse Network as an Infrastructure for Data Sharing. *Sociol. Methods Res.* **2007**. [CrossRef]
8. Marchionini, G.; Lee, C.A.; Bowden, H.; Lesk, M. *Curating for Quality: Ensuring Data Quality to Enable New Science*; Final Report: Invitational Workshop Sponsored by the National Science Foundation; National Science Foundation: Arlington, VA, USA, 2012.
9. Cai, L.; Zhu, Y. The challenges of data quality and data quality assessment in the big data era. *Data Sci. J.* **2015**, *14*. [CrossRef]
10. Martin, E.G.; Law, J.; Ran, W.; Helbig, N.; Birkhead, G.S. Evaluating the quality and usability of open data for public health research: A systematic review of data offerings on 3 open data platforms. *J. Public Health Manag. Pract.* **2017**, *23*, e5–e13. [CrossRef]
11. Ferguson, A.R.; Nielson, J.L.; Cragin, M.H.; Bandrowski, A.E.; Martone, M.E. Big data from small data: Data-sharing in the 'long tail' of neuroscience. *Nat. Neurosci.* **2014**, *17*, 1442–1447. [CrossRef]
12. Heidorn, P.B. Shedding Light on the Dark Data in the Long Tail of Science. *Libr. Trends* **2008**, *57*, 280–299. [CrossRef]
13. Palmer, C.L.; Cragin, M.H.; Heidorn, P.B.; Smith, L.C. Data Curation for the Long Tail of Science: The Case of Environmental Sciences. In Proceedings of the Third International Digital Curation Conference, Washington, DC, USA, 11–13 December 2007; pp. 11–13.
14. Cragin, M.H.; Palmer, C.L.; Carlson, J.R.; Witt, M. Data sharing, small science and institutional repositories. *Philos. Trans. Math. Phys. Eng. Sci.* **2010**, *368*, 4023–4038. [CrossRef]
15. Chen, X.; Dallmeier-Tiessen, S.; Dasler, R.; Feger, S.; Fokianos, P.; Gonzalez, J.B.; Hirvonsalo, H.; Kousidis, D.; Lavasa, A.; Mele, S.; et al. Open is not enough. *Nat. Phys.* **2019**, *15*, 113–119. [CrossRef]
16. Gregory, K.; Groth, P.; Scharnhorst, A.; Wyatt, S. Lost or Found? Discovering Data Needed for Research. *Harv. Data Sci. Rev.* **2020**. [CrossRef]
17. Pasquetto, I.V.; Borgman, C.L.; Wofford, M.F. Uses and reuses of scientific data: The data creators' advantage. *Harv. Data Sci. Rev.* **2019**, *2019*, *1*.
18. Borgman, C.L.; Wallis, J.C.; Enyedy, N. Little Science Confronts the Data Deluge: Habitat Ecology, Embedded Sensor Networks, and Digital Libraries. *Cent. Embed. Netw. Sens.* **2006**, *7*, 17–30. [CrossRef]
19. Borgman, C.L. The conundrum of sharing research data. *J. Am. Soc. Inf. Sci. Technol.* **2012**, *63*, 1059–1078. [CrossRef]
20. Wallis, J.C.; Rolando, E.; Borgman, C.L. If We Share Data, Will Anyone Use Them? Data Sharing and Reuse in the Long Tail of Science and Technology. *PLoS ONE* **2013**, *8*, e67332. [CrossRef]
21. National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*; National Academies Press: Washington, DC, USA, 2019. [CrossRef]
22. Trisovic, A. Replication Data for: Repository approaches to improving quality of shared data and code. *Harvard Dataverse*, 13 October 2020. [CrossRef]
23. Trisovic, A.; Durbin, P.; Schlatter, T.; Durand, G.; Barbosa, S.; Brooke, D.; Crosas, M. Advancing Computational Reproducibility in the Dataverse Data Repository Platform. In Proceedings of the 3rd International Workshop on Practical Reproducible Evaluation of Computer Systems, P-RECS '20, Stockholm, Sweden, 23 June 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 15–20. [CrossRef]
24. Hense, A.; Quadt, F. Acquiring high quality research data. *D-Lib Mag.* **2011**, *17*. [CrossRef]
25. Wilkinson, M.D.; Dumontier, M.; Aalbersberg, I.J.; Appleton, G.; Axton, M.; Baak, A.; Blomberg, N.; Boiten, J.W.; da Silva Santos, L.B.; Bourne, P.E.; et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **2016**, *3*, 1–9. [CrossRef]
26. Boyd, C. Harvard Dataverse Optional Feature Use Data. *Harvard Dataverse*, 2 October 2020; [CrossRef]
27. Koshoffer, A.; Neeser, A.E.; Newman, L.; Johnston, L.R. Giving datasets context: A comparison study of institutional repositories that apply varying degrees of curation. *Int. J. Digit. Curation* **2018**, *13*, 15–34. [CrossRef]
28. Bishop, B.W.; Hank, C.; Webster, J.; Howard, R. Scientists' data discovery and reuse behavior: (Meta)data fitness for use and the FAIR data principles. *Proc. Assoc. Inf. Sci. Technol.* **2019**, *56*, 21–31. [CrossRef]
29. Smit, E. Abelard and Héloise: Why Data and Publications Belong Together. *D-Lib Mag.* **2011**, *17*. [CrossRef]
30. Faniel, I.M.; Jacobsen, T.E. Reusing Scientific Data: How Earthquake Engineering Researchers Assess the Reusability of Colleagues' Data. *Comput. Support. Coop. Work. (CSCW)* **2010**, *19*, 355–375. [CrossRef]
31. Deterding, S.; Khaled, R.; Nacke, L.E.; Dixon, D. Gamification: Toward a definition. In Proceedings of the CHI 2011 Gamification Workshop Proceedings, Vancouver BC, Canada, 7–12 May 2011; Volume 12.

32. Hamari, J.; Koivisto, J.; Sarsa, H. Does gamification work?—A literature review of empirical studies on gamification. In Proceedings of the 2014 47th Hawaii International Conference on System Sciences (HICSS), Waikoloa, HI, USA, 6–9 January 2014; pp. 3025–3034.
33. Knaving, K.; Woźniak, P.W.; Niess, J.; Poguntke, R.; Fjeld, M.; Björk, S. Understanding grassroots sports gamification in the wild. In Proceedings of the 10th Nordic Conference on Human-Computer Interaction, Oslo, Norway, 1–3 September 2018; pp. 102–113.
34. Oprescu, F.; Jones, C.; Katsikitis, M. I PLAY AT WORK—Ten principles for transforming work processes through gamification. *Front. Psychol.* **2014**, *5*, 14. [CrossRef]
35. Ibanez, M.B.; Di-Serio, A.; Delgado-Kloos, C. Gamification for Engaging Computer Science Students in Learning Activities: A Case Study. *IEEE Trans. Learn. Technol.* **2014**, *7*, 291–301. [CrossRef]
36. Eveleigh, A.; Jennett, C.; Lynn, S.; Cox, A.L. "I want to be a captain! I want to be a captain!": Gamification in the old weather citizen science project. In Proceedings of the First International Conference on Gameful Design, Research, and Applications—Gamification '13, Toronto, ON, Canada, 2–4 October 2013; pp. 79–82. [CrossRef]
37. Bowser, A.; Hansen, D.; Preece, J.; He, Y.; Boston, C.; Hammock, J. Gamifying citizen science: A study of two user groups. In Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work and Social Computing, CSCW 2014, Baltimore, MD, USA, 15–19 February 2014; pp. 137–140. [CrossRef]
38. Nicholson, S. A recipe for meaningful gamification. In *Gamification in Education and Business*; Springer: New York, NY, USA, 2015; pp. 1–20. [CrossRef]
39. Feger, S.; Dallmeier-Tiessen, S.; Woźniak, P.; Schmidt, A. Just Not The Usual Workplace: Meaningful Gamification in Science. In Proceedings of the Mensch und Computer 2018-Workshopband, Dresden, Germany, 2–5 September 2018.
40. Feger, S.S.; Dallmeier-Tiessen, S.; Woźniak, P.W.; Schmidt, A. Gamification in Science: A Study of Requirements in the Context of Reproducible Research. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; pp. 1–14. [CrossRef]
41. Kidwell, M.C.; Lazarević, L.B.; Baranski, E.; Hardwicke, T.E.; Piechowski, S.; Falkenberg, L.S.; Kennett, C.; Slowik, A.; Sonnleitner, C.; Hess-Holden, C.; et al. Badges to acknowledge open practices: A simple, low-cost, effective method for increasing transparency. *PLoS Biol.* **2016**, *14*, e1002456. [CrossRef]
42. Rowhani-Farid, A.; Allen, M.; Barnett, A.G. What incentives increase data sharing in health and medical research? A systematic review. *Res. Integr. Peer Rev.* **2017**, *2*, 4. [CrossRef]
43. Borges, H.; Valente, M.T. What's in a GitHub star? understanding repository starring practices in a social coding platform. *J. Syst. Softw.* **2018**, *146*, 112–129. [CrossRef]