Code Contribution and Authorship

Eva Maxfield Brown   Nicholas Weber

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur eget porta erat. Morbi consectetur est vel gravida pretium. Suspendisse ut dui eu ante cursus gravida non sed sem. Nullam sapien tellus, commodo id velit id, eleifend volutpat quam. Phasellus mauris velit, dapibus finibus elementum vel, pulvinar non tellus. Nunc pellentesque pretium diam, quis maximus dolor faucibus id. Nunc convallis sodales ante, ut ullamcorper est egestas vitae. Nam sit amet enim ultrices, ultrices elit pulvinar, volutpat risus.

1 Random Citations

- may be interesting / useful: https://doi.org/10.1145/1772690.1772766

2 Introduction

- Contemporary scientific research fundamentally depends on specialized software tools and computational methods (Edwards et al. 2013; Mayernik et al. 2017; Howison et al. 2015).
  - define scientific software (analysis scripts, research tools, computational infrastructure) (Hasselbring et al. 2024)
  - software enables reproducible research and large-scale experiments (Krafczyk et al. 2019; Trisovic et al. 2021)
  - code serves as a detailed log of research methodology (Ram 2013)
  - due to all of the above, code is increasingly being shared alongside research articles (Cao et al. 2023; Trujillo, Hébert-Dufresne, and Bagrow 2022)
- The development and maintenance of scientific software requires substantial contribution, yet faces persistent challenges in receiving academic recognition (Muna et al. 2016).
  - software contributions often receive only acknowledgments rather than authorship (Philippe et al. 2019)

  - lack of formal credit affects career advancement in academia (Carver et al. 2022; Biagioli and Galison 2014)
  - other general discussion of software citations and credit systems (Merow et al. 2023; Westner et al. 2024; Katz et al. 2020)

- Recent initiatives to expand academic credit systems, while promising, have not fully addressed the challenges of recognizing software contributions.

  - describe the Contributor Roles Taxonomy (CRediT) (Brand et al. 2015)
  - previous research using CRediT (and prior systems) to understand research labor distribution (Larivière, Pontille, and Sugimoto 2020; Larivière et al. 2016; Sauermann and Haeussler 2017; K. Li, Zhang, and Larivière 2023; Lu et al. 2019)
  - CRediT research is still centered on traditional author lists (historic and systematic bias, self-reporting without verification, etc.) (Haeussler and Sauermann 2013; Gøtzsche et al. 2007; Ni et al. 2021)

- Our novel predictive model addresses these challenges by enabling systematic matching between scientific article authors and source code developer accounts.

  - we use predictive modeling due to the lack of standardized identifiers (i.e. ORCID) for developers (Haak et al. 2012)
  - further, lack consistency in naming and email overlap [GET CITATION FOR SORTING HAT FROM GOGGINS??]
  - semantic models handle subtle variations in identity information (general entity matching has moved to transformers and semantic embeddings) (Y. Li et al. 2020; Brunner and Stockinger 2020)

- By applying our model across a corpus of 138596 paired research articles and repositories, we provide unique insight into the dynamics of code contribution within research teams, the impact of code contribution on research outcomes, and an understanding of the authors who are and who aren't code contributors.

  - move from self-reporting to verifiable source code repository commit histories
  - provide preliminary quantitative evidence of exclusion of code contributors from academic authorship
  - model article level impact metrics as a function of software development dynamics to show the benefit code contributors have on research
  - find that first authors are more likely to be code contributors than not
  - find that code-contributing authors have reduced individual level impact metrics compared to their non-coding counterparts

- These findings not only illuminate the relationship between code contribution and scientific impact but also provide an empirical foundation for reforming academic credit systems to better recognize software development contributions in research.

3 Background

- The relationship between scientific software development and academic credit systems represents a complex intersection of traditional academic practices and modern research requirements.
  - academic credit traditionally focuses on analytical, theoretical, and experimental contributions (Larivière et al. 2016; X. Liu, Zhang, and Li 2023)
  - software development historically viewed as technical rather than scholarly work
  - growing recognition that research software development requires deep domain expertise (Heroux 2022; Carver et al. 2022)
  - increased emphasis on large scale (big data) projects has resulted in larger need for software development (Jin et al. 2015; Hampton et al. 2013; Fan, Han, and Liu 2014)
  - understanding this relationship requires examining both team-level dynamics and individual contributions
- (H1) Modern research increasingly depends on collaborative software development, yet we lack systematic evidence of how code contribution patterns affect research outcomes.
  - existing research focuses primarily on general team size and diversity (Franceschet and Costantini 2010; Larivière et al. 2014; AlShebli et al. 2024; Yang, Ding, and Liu 2024; L. Liu et al. 2021; Naik et al. 2023)
  - software engineering literature shows correlation between team size and code quality (many eyes make all bugs shallow) (Wyss, De Carli, and Davidson 2023; Meirelles et al. 2010)
  - limited understanding of how code contribution is associated to research impact
  - need to understand relationship between code contributors and citation metrics to understand the value of these technical, potentially uncredited, contributions
  - we believe that more code contributors may signal a more technical research project and that technical complexity may be rewarded with more citations
- (H2) Despite formal taxonomies like CRediT attempting to standardize contribution recognition, the criteria for granting authorship to technical contributors remain inconsistent and poorly understood across research communities.
  - existing contribution frameworks provide definitions for software development roles (K. Li, Zhang, and Larivière 2023; Ding et al. 2021)
  - however, these frameworks may not capture the full spectrum of technical contributions
  - repository histories allow us to examine how sustained technical engagement relates to authorship status (Ram 2013)
  - we believe that longer project involvement increases likelihood of authorship recognition

- specifically, we hypothesize that projects with longer durations will show higher proportions of author-developers compared to non-author developers

- (H3 and H4) Academic authorship conventions signal both intellectual contribution and project responsibilities, yet their relationship to software development remains poorly understood.
  - first authors traditionally responsible for primary intellectual and experimental contributions (Larivière et al. 2016; Larivière, Pontille, and Sugimoto 2020; Júnior et al. 2016)
  - corresponding authors serve as primary points of contact and often maintain research artifacts
  - varying expectations across academic disciplines regarding technical contributions (E. Smith 2023)
  - limited research examining how these authorship roles relate to direct code contributions
  - potential insights into how software development responsibilities are distributed within research teams
  - we believe that first authors and corresponding authors will have higher proportions of code contribution than not.
  - conversely, middle and last authors and non-corresponding authors will have lower proportions of code contribution than not.

- (H5) Academic career advancement has historically depended on traditional impact metrics, creating potential tension for researchers who dedicate significant time to software development.
  - lack of formal software citation standards and adoption may directly result in software and methodological focused researchers to experience lower citation counts
  - more generally, theoretical contributions may be more widely shared and cited than technical and methodological contributions
  - potential career implications for researchers who prioritize coding (Hannay et al. 2009; Heroux 2022; A. M. Smith, Norman, and Cruz 2019; Cosden, McHenry, and Katz 2022)
  - need to understand relationship between code contributions and academic impact
  - we believe that code contributing researchers will have lower individual level impact metrics than non-coding researchers, while we can't be clear on mechanism, we believe it may be do to some combination of lack of formal citation, or methodological contributions being less widely shared unless they represent fundamental shifts in the field
    - trying to get to the idea that software is iterative and minor optimizations don't make big waves

- Understanding these relationships is crucial for developing equitable academic credit systems that recognize the full spectrum of research contributions.

- findings will inform policy making around academic credit
- importance of large-scale quantitative evidence for understanding current credit systems
- implications for academic hiring and promotion decisions
- potential to develop new impact metrics that capture software contributions

4 Data and Methods

4.1 Linking Scientific Articles and Source Code Repositories

- Modern scientific research increasingly requires the public sharing of research code, creating unique opportunities to study the relationship between academic authorship and software development.
  - many journals and platforms now require or recommend code and data sharing (Stodden, Guo, and Ma 2013; Sharma et al. 2024)
  - this requirement creates traceable links between publications and code
  - these links enable systematic study of both article-repository and author-developer relationships (Hata et al. 2021; Kelley and Garijo 2021; Stankovski and Garijo 2024; Milewicz, Pinto, and Rodeghero 2019)
- Our data collection process leverages multiple complementary sources of linked scientific articles and code repositories to ensure comprehensive coverage.
  - PLOS: Traditional research articles with code requirements
  - JOSS and SoftwareX: Specialized software-focused publications
  - Papers with Code / ArXiv: Capturing pre-print landscape
  - to reduce the complexity of dataset processing and enrichment, we filter out any article-source-code-repository pairs which store code somewhere other than GitHub, note: we do this for simplicity of processing but recognize that there is work elsewhere to understand code outside of GitHub (Trujillo, Hébert-Dufresne, and Bagrow 2022)
- Through integration of multiple data sources, we extract detailed information about both the academic and software development aspects of each project.
  - specifically we utilize the Semantic Scholar API for article DOI resolution to ensure that we find the latest version for each article.
  - this is particularly important for working with preprints as they may have been published in a journal since their inclusion in the Papers with Code dataset
  - we then utilize the OpenAlex API to gather publication metadata (i.e. open access status, domain, publication date), author details (i.e. name, author position, corresponding author status), and article- and individual-level metrics (i.e. citation count, FWCI, h-index).

- the GitHub API provides similar information for source code repositories, including repository metadata (i.e. name, description, languages, creation date), contributor details (i.e. username, name, email), and repository-level metrics (i.e. star count, fork count, issue count).

- while the majority of our data is sourced from Papers with Code, our additional collection from PLOS, JOSS, and SoftwareX as well as the enrichment from GitHub and OpenAlex together form one of the largest collections of linked, metadata enriched, datasets of paired scientific articles and associated source code repositories.
  - in total, we collect and enrich data for 163292 article-repository pairs

4.2 A Predictive Model for Matching Article Authors and Source Code Contributors

4.2.1 Annotated Dataset Creation

- The development of an accurate author-developer matching model requires high-quality labeled training data that captures the complexity of real-world identity matching.
  - entity matching between authors and developers is non-trivial
  - multiple forms of name variation and incomplete information
  - add figure showing example matches/non-matches
- We developed an annotation process to create a robust training dataset while maximizing efficiency and accuracy.
  - focus on JOSS articles to increase positive match density
  - we create author-developer pairs for annotation by creating all possible combinations of authors and developers within a single JOSS article-repository pair
  - we take a random sample of 3000 pairs from the full set and have two independent annotators label each
  - after all 3000 pairs are annotated, we resolve any disagreements between the two annotators
- The resulting annotated dataset provides a comprehensive foundation for training our predictive model while highlighting common patterns in author-developer identity matching.
  - after resolution of all annotated pairs, our annotated dataset contains 451 (15.0%) positive and 2548 (85.0%) negative author-developer-account pairs
  - there are 2027 unique authors and 2733 unique developer accounts within this annotated set
  - however, not all developer accounts contain complete information, in our set 2191 (80.2%) have associated names and 839 (30.7%) have associated emails

4.2.2 Training and Evaluation

- To optimize our predictive model for author-contributor matching, we evaluate a variety of Transformer-based base models and input features.
  - specifically, we fine-tune from three different base transformer models:
    * deberta-v3-base (He, Gao, and Chen 2021; He et al. 2021)
    * bert-base-multilingual-cased (Devlin et al. 2018)
    * distilbert-base-uncased (Sanh et al. 2019)
  - these three models are all variations or built-upon BERT, and while significant time has passed since BERT was first introduced, BERT-based models remain a strong base for many NLP tasks across a number of domains while being relatively "small" compared to the much larger decoder transformer relatives (GPT, Llama, etc.) (Tran et al. 2024; Yu et al. 2024; Jeong and Kim 2022)
- We employed a systematic evaluation to identify optimal combination of base models and input features.
  - first, to ensure that there was no data leakage, we split our dataset into training and test sets
  - specifically, we created two random sets of 10% of all unique authors and 10% of all unique developers, any pairs containing either the author or developer were placed into the test set
  - in doing so, we ensured that the model was never trained on any author or developer information later used for evaluation
  - due to the fact that each author and developer-account can be included in multiple annotated pairs, our final training set contains 2442 (81.4%) and our test set contains 557 (18.6%) author-developer-account pairs
  - we fine-tuned each of our three base models using all combinations of available developer-account features, from including only the developer account username to including the developer's username, name, and email.
  - to avoid overfitting and ensure generalizability, we fine-tuned each of the base models for only a single training epoch.
  - model evaluation was performed using standard classification metrics, including accuracy, precision, recall, and F1 score
- After extensive model comparison we find that fine-tuning from Microsoft's deberta-v3-base and including the developer's username and name achieves the best performance for author-developer matching.
  - our best model achieves a binary F1 score of 0.944, with an accuracy of 0.984, precision of 0.938, and recall of 0.95 (see Figure 1 for a confusion matrix of model predictions on the test set).
  - analysis of feature importance

* note that the addition of developer's name has a "larger effect" on model performance but that could simply be because of how many more developers have a name available than an email
* also note that there is a model that performs just as well as this one using bert-multilingual and includes the developers email however we choose to use the deberta and name only version for its simplicity as well as the fact that deberta is a much more recently developed and released model which was pre-trained on a much larger dataset.
* considering that in most cases, deberta out-performs bert-multilingual, we believe that while the overall evaluation metrics between the top two performing models are the same, the deberta based model will generalize to other unseen data better than the bert-multilingual model

- all model and feature set combination results are available in Table 4

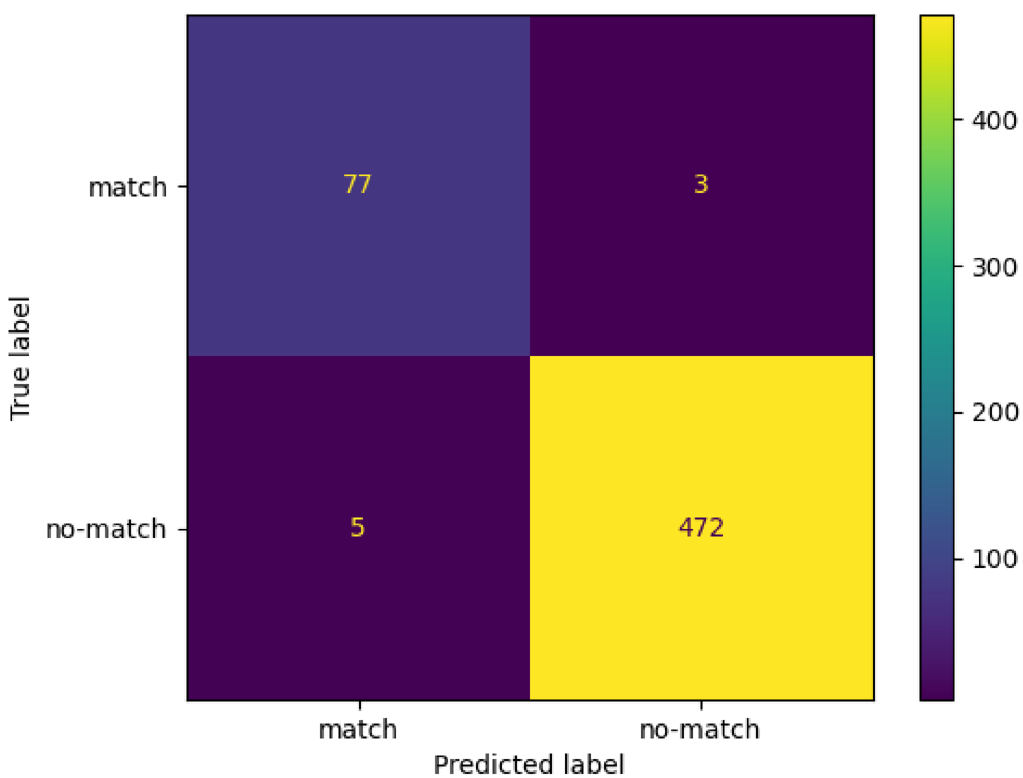

Figure 1: Confusion Matrix Produced From Evaluation of Best Performing Model (deberta-v3 with developer username, developer name, and author name).

- To enable future research, we have made our trained model and supporting application library publicly available.
  - Python library implementation: sci-soft-models
  - HuggingFace model deployment: dev-author-em-clf

4.3 Linking Authors and GitHub Developer Accounts

- Our trained entity-matching model enables comprehensive identification of author-developer relationships while accounting for the complex realities of academic software development practices.
  - in practice, to fill out our dataset, we apply our trained model to all possible author and developer-account combinations within each article-repository pair
  - The presence of multiple developer accounts per individual reflects common practices in academic software development that must be accommodated in our analysis.
  - developers often maintain separate accounts for different projects or institutions
  - account transitions are common as researchers move between roles
- Further, while our model performs well, we note that there are some limitations to our approach.
  - in most cases predictions are trivial due to minor differences in text (spelling of author name to username)
  - however we do observe a few cases in which our model may not perform as well
  - namely, shorter names, articles and repositories which have contributors with the same last name (i.e. siblings or other relationship), and "organization" accounts (i.e. research lab GitHub accounts used for management, administration, and documentation or a project)
  - TODO: should we take a sample and estimate how widespread these problems are?
  - we include appropriate filtering during analysis to ensure that we do not include author-developer pairs which are unlikely to be the same individual
- Our final dataset provides unprecedented scale and scope for analyzing the relationship between academic authorship and software development contributions.
  - Specifically, our dataset contains 138596 article-repository pairs, 295806 distinct authors, and 152170 distinct developer accounts.
  - From the 295806 distinct authors and 152170 distinct developer accounts we are able to create 108754 annotated author-developer pairs
  - a detailed breakdown of these counts by data source, domain, document type, and open access status is available in Table 1

Table 1: Counts of Article-Repository Pairs, Authors, and Developers by Data Sources, Domains, Document Types, and Access Status.

Category	Subset	Article-Repository Pairs	Authors	Developers
By Domain	Physical Sciences	116600	240545	130592
	Social Sciences	8838	29269	14043
	Life Sciences	7729	31649	12150
	Health Sciences	5172	25979	7248
By Document Type	preprint	72177	170301	87311
	research article	63528	173183	78935
	software article	2891	9294	12868
By Access Status	Open	132856	286874	147831
	Closed	5740	23668	9352
By Data Source	pwc	129615	262889	134926
	plos	6090	30233	8784
	joss	2336	7105	11362
	softwarex	555	2244	1628
Total		**138596**	**295806**	**152170**

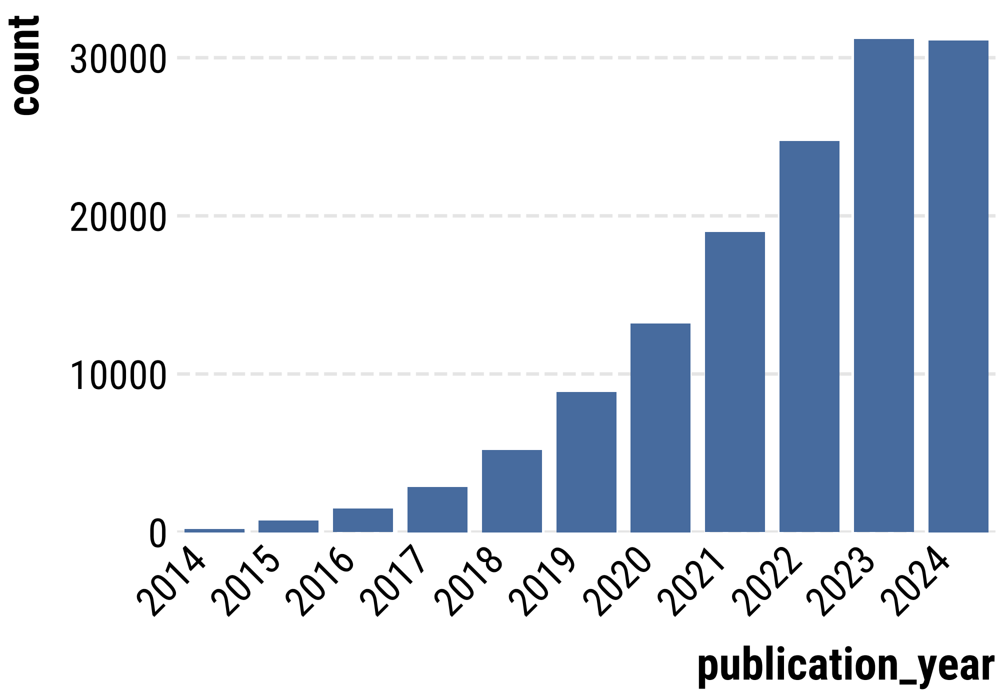

Figure 2: Number of articles by publication year. Only publication years with 100 or more articles are included.

5 Preliminary Analysis Code Contributor Authorship and Development Dynamics of Research Teams

5.1 Software Development Dynamics Within Research Teams

- We begin by measuring the distributions of different coding and non-coding contributors across all of the article-code-repository pairs within our dataset.
  - individuals in our dataset can fall into three categories:
    * Code-Contributing Authors (CC-A): authors for which our model predicted a match with at least one developer account which contributed code to the associated repository
    * Non-Code-Contributing Authors (NCC-A): authors for which our model did not predict any matches with developer accouts which contributed code to the associated repository
    * Code-Contributing Non-Authors (CC-NA): developer accounts which contributed code to the associated repository but were not predicted to be a match with any author
  - within our dataset, we find that papers on average have 4.9 ± 1.9 total authors, 3.9 ± 2.0 non-code-contributing authors, 1.0 ± 0.7 code-contributing authors, and 0.4 ± 1.7 code-contributing non-authors (see Figure 3 for a visualization of these distributions).
  - Table 2 provides a detailed breakdown of these distributions by domain, article type, and open access status.
  - Our finding of, on average, only a single code-contributing author on a paper, is similar to previous work in understanding distributions of labor in knowledge production from Larivière, Pontille, and Sugimoto (2020) which found that the CRediT tasks of "Data Curation", "Formal Analysis", "Visualization", and "Software" were all predominantly performed by a first author.
  - However, we do also see that code-contributing non-authors are present in the data, albeit, on average, with less than one code-contributing non-author per article.

- Next we investigate how these distributions have changed over time and how they change by the total number of authors on a paper.
  - Figure 4 shows thats both, the mean number of code-contributing authors and code-contributing non-authors has remained relatively stable over time and across different total author sizes.
  - these descriptive statistics suggest that code-contributing non-authors are an inconsistent feature of research teams, and while their exclusion from authorship is not a recent phenomenon, their exclusion does not appear to be getting worse over time.

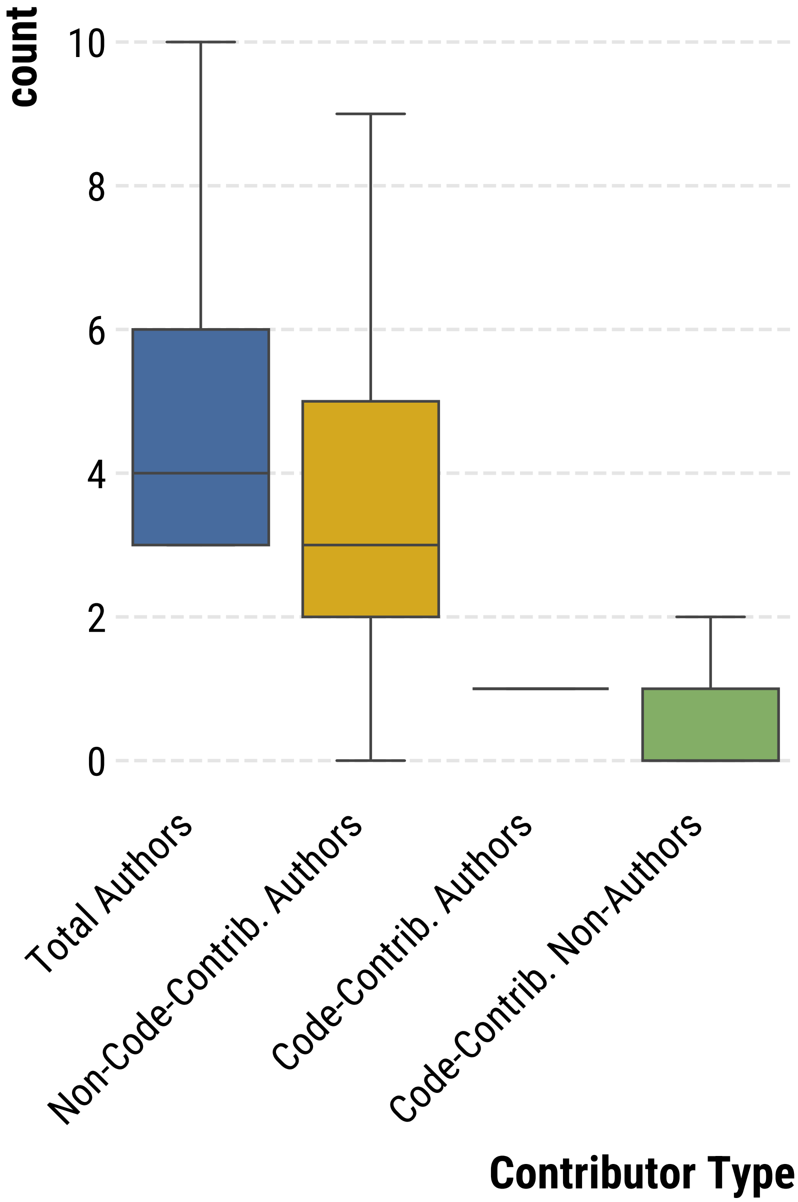

Figure 3: Distribution of the number of Total Authors, Non-Code-Contributing Authors, Code-Contributing Authors, and Code-Contributing Non-Authors across all article-repository pairs. Only includes article-repository pairs with a most recent commit no later than 90 days after publication.

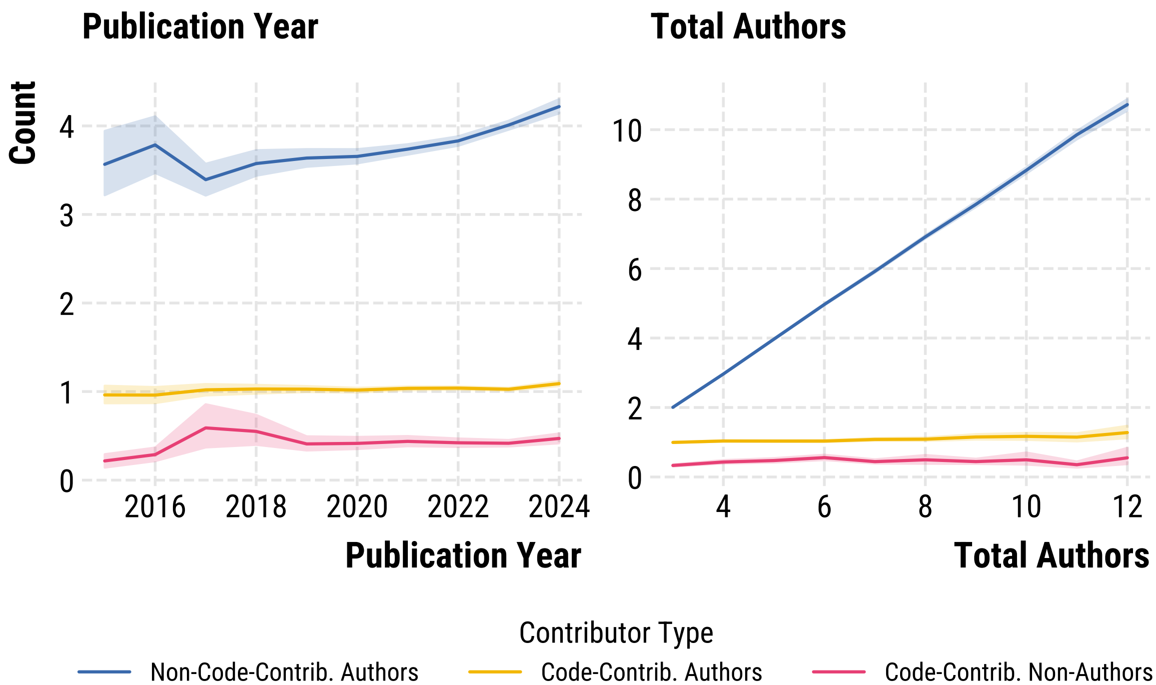

Figure 4: Mean number of Non-Code-Contributing Authors, Code-Contributing Authors, and Code-Contributing Non-Authors by Publication Year and by Total Number of Authors. Only includes article-repository pairs with a most recent commit no later than 90 days after publication and excludes research-teams which are in the top 3% of total author sizes for publication years with 50 or more articles.

Table 2: Mean and Standard Deviation of Non-Code-Contributing Authors (NCC-A), Code-Contributing Authors (CC-A), and Code-Contributing Non-Authors (CC-NA) Research Team Members by Domain, Article Type, and Open Access Status. Only includes research teams from article-repository pairs with a most recent commit no later than 90 days after publication and excludes research teams which are in the top 3% of total author sizes.

Control	Subset	Total Authors	NCC-A	CC-A	CC-NA
OA Status	Closed	5.1 ± 1.9	4.0 ± 1.9	1.1 ± 0.7	0.5 ± 2.1
	Open	4.9 ± 1.9	3.9 ± 2.0	1.0 ± 0.7	0.4 ± 1.7
Domain	Health Sciences	6.1 ± 2.5	5.1 ± 2.5	1.0 ± 0.6	0.4 ± 1.2
	Life Sciences	5.2 ± 2.1	4.2 ± 2.2	1.0 ± 0.7	0.4 ± 1.2
	Physical Sciences	4.8 ± 1.8	3.8 ± 1.9	1.0 ± 0.7	0.5 ± 1.8
	Social Sciences	4.5 ± 1.7	3.5 ± 1.8	1.1 ± 0.7	0.3 ± 1.1
Article Type	preprint	4.8 ± 1.8	3.8 ± 1.9	1.1 ± 0.7	0.5 ± 2.2
	research article	4.9 ± 1.9	3.9 ± 2.0	1.0 ± 0.7	0.4 ± 1.6
	software article	4.7 ± 1.9	3.2 ± 1.9	1.5 ± 1.4	0.9 ± 1.1

5.1.1 Modeling Citations

- Building upon previous work which discuss the effects of team size and team diversity on scientific impact and software quality (see Section 3), we examine how the number of code contributors within a research team may be associated with an article's research impact.
  - We hypothesize that more code contributors may signal greater technical complexity in research, which may be associated with higher citation counts as the community builds upon more technically sophisticated works.
- However, after analyzing our data, we find few significant associations between the number of code-contributing authors and non-authors, and article citations.
  - Without controlling for any domain, open access, or article type differences (**?@tbl-article-composition-overall**), we find a positive association between the number of code-contributing authors and article citations with each code-contributing author being associated with a 4.5% increase in article citations ($p < 0.001$).
  - Controlling for article type (Table 7), we find that for preprints, each code-contributing non-author is associated with a statistically significant 2.9% decrease in expected citations, holding other variables constant ($p < 0.01$).
  - For research articles, we find a significant positive association between the number of code-contributing authors and citations ($p < 0.001$). However, we cannot estimate the precise magnitude of this effect due to the non-significant main effect in the model.

- Finally, we additionally find a statistically significant relationship for code-contributing non-authors for research articles, specifically finding that each code-contributing non-author is associated with a 0.8% increase in expected citations ($p < 0.001$).
  - Overall, while we find some statistically significant associations between code contributions and citation counts, these effects are relatively modest in magnitude.

5.1.2 Team Composition and Project Duration

- Building upon previous work examining standardized contribution frameworks and authorship attribution practices (see Section 3), we investigate how project duration may influence the distribution of coding roles between authors and non-authors.
  - We hypothesize that projects with longer development durations may show higher proportions of author-developers compared to non-author developers, as sustained technical engagement could increase the likelihood of receiving authorship recognition.
- However, our analysis finds no evidence to support this hypothesis:
  - When examining the relationship between a repository's commit duration and the percentage of developers who receive authorship recognition, we find no significant correlation ($r = -0.00$, $p =$ n.s.).
  - This suggests that the length of time a project has been in development has no meaningful relationship with the proportion of developers who are recognized as authors.
  - While this suggests authorship recognition may be driven by other factors, a more precise analysis would examine individual contribution durations rather than overall project length.

5.2 Characteristics of Scientific Code Contributors

5.2.1 Author Positions of Code Contributing Authors

- Building upon previous work examining the relationship between authorship position and research contributions, we investigate how author position may relate to code contribution patterns.
  - We hypothesize that first authors, who traditionally contribute the bulk of intellectual and experimental work, would be most likely to contribute code to a project, while middle and last authors, who often provide oversight and guidance, would be less likely to contribute code.
- Our analysis supports this hypothesis:

- In every case tested, we find that first authors are statistically significantly more likely to be code contributors than non-code contributors across most scenarios both overall, and regardless of Domain, Article Type, and Open Access Status.
- Conversely, both middle authors and last authors are statistically significantly more likely to be non-code contributors.
- These patterns align with traditional academic authorship conventions, where first authors often take primary responsibility for both intellectual and technical aspects of the research.
- Table 8 provides a detailed breakdown of Bonferronic-correct p-values from post-hoc binomial tests against the counts of author positions of code-contributing authors.

5.2.2 Corresponding Status of Code Contributing Authors

- Building upon our analysis of author position, we next examine how corresponding author status relates to code contribution patterns.
  - We hypothesize that corresponding authors, who traditionally maintain research artifacts and serve as primary points of contact, would be more likely to contribute code compared to non-corresponding authors.
- However, our analysis reveals patterns contrary to this hypothesis:
  - Both corresponding and non-corresponding authors are statistically significantly less likely to be code contributors than would be expected by chance ($p < 0.001$) in most scenarios (Table 9).
  - This pattern holds true across almost all conditions with the only exceptions found in software articles and closed access articles, where corresponding authors show no significant difference in their likelihood to contribute code.
  - These findings suggest that the responsibility for maintaining research artifacts may not typically extend to direct code contributions, contrary to traditional assumptions about corresponding author roles.

5.2.3 Modeling H-Index

- Building upon previous work examining career implications for researchers who prioritize software development (see Section 3), we investigate how varying levels of code contribution relate to scholarly impact through citation metrics.
  - We hypothesize that researchers who contribute more code may show lower citation counts, potentially due to a lack of formal software citation or more general attitudes towards methodological vs theoretical contributions.

Table 3: Counts of Researcher Coding Status Used in H5

Control	Subset	Any Coding	Majority Coding	Always Coding	Total
Freq. Author Pos.	First	1689	4952	3322	11444
	Middle	12184	3327	701	31911
	Last	2413	568	191	10188
Freq. Domain	Health Sciences	345	202	86	1501
	Life Sciences	369	241	129	1436
	Physical Sciences	15289	8179	3821	49311
	Social Sciences	283	225	178	1295
Freq. Article Type	Preprint	9572	4963	2214	28948
	Research Article	6669	3765	1871	24217
	Software Article	45	119	129	378

- While we can't confirm the underlying mechanisms, we do find strong evidence to support our general claim that code-contributing researchers are associated with lower h-indices compared to their non-coding peers.
  - Without any controls (Table 10), we find increasingly negative h-index effects as coding frequency increases compared to non-coding authors with scientists who occasionally make code contributions being associated with ~27.3% lower h-indices than their non-coding counterparts ($p < 0.001$), followed by ~53.5% lower h-indices for majority coders ($p < 0.001$), and ~62.1% lower h-indices for always coding authors ($p < 0.001$).
  - When controlling for author position (Table 11), we find again find a general pattern that more frequent code contribution is associated with reduced h-indices compared to non-coding peers with a few exceptions.
    * Occasional coding first authors are associated with a ~14.9% higher h-index ($p < 0.001$) while always coding first authors are associated with a ~21.6% reduction in h-index compared to non-coding first authors ($p < 0.001$).
    * Occasional coding middle authors are associated with a ~26.6% lower h-index ($p < 0.001$) while always coding middle authors are associated with a ~52.9% lower h-index compared to non-coding middle authors ($p < 0.001$).
    * Occasional coding last authors are associated with a ~13.1% lower h-index ($p < 0.001$) while always coding last authors are associated with a ~45.7% lower h-index compared to non-coding last authors ($p < 0.001$).
  - When controlling for domain (Table 12), majority coding scientists in health science domains are associated with a ~76.5% reduction in h-index compared to non-coding peers ($p < 0.001$), while majority coding scientists in life science see a ~47.1% reduction ($p < 0.001$), physical science majority coding scientists see a ~52.3% reduction ($p < 0.001$), and majority coding scientists in social sciences see a ~51.4% reduction in h-index compared to non-coding peers ($p < 0.001$).

- Finally, when controlling for an authors common article type (Table 13), we find that:
    - for authors whose most common article type is preprints, occasionally coding authors are associated with a ~25.6% lower h-index, majority coding authors are associated with a ~53.5% lower h-index, and always coding authors are associated with a ~62.9% lower h-index compared to their non-coding peers.
    - while for authors whose most common article type is software articles, majority coding authors are associated with a ~33.1% lower h-index, and always coding authors are associated with a ~33.0% lower h-index compared to their non-coding peers.

- Taken as a whole, these findings tend to indicate that the more frequently an author contributes code, the lower their h-index is likely to be relative to their peers, with one exception being first authors who occasionally contribute code.

6 Discussion

RE: article citations

- Our findings that code contribution and distribution among research teams has only a modest relationship with citations might reflect several underlying factors.
    - First, the relationship between code contributions and research impact may be complex and indirect - while code might enhance research reproducibility or utility, this may not directly translate into increased citations
    - Second, current citation practices may not adequately capture or credit code contributions, as authors might reference papers without explicitly acknowledging the associated code.
    - Third, the quality and significance of code contributions likely varies substantially across papers, making it difficult to detect strong aggregate effects.
    - Finally, while we have attempted to find and match as many code contributing members of research teams as possible, we must acknowledge that there may be two code-centric reasons why we have not found more code contributors:
        - First, the code may have been developed by multiple individuals but only a single individual uploaded or committed the code to a repository, thereby removing the opportunity for us to link the code to the other contributors.
        - Second, we are primarily analyzing "analysis" repositories, which may not contain the full codebase for a project. This is particularly true for projects which separate repositories for tools, libraries, and infrastructures, than the single analysis code repository.

7 References

AlShebli, Bedoor, Shahan Ali Memon, James A Evans, and Talal Rahwan. 2024. "China and the US Produce More Impactful AI Research When Collaborating Together." *Scientific Reports* 14 (1): 28576.

Biagioli, Mario, and Peter Galison. 2014. *Scientific Authorship: Credit and Intellectual Property in Science*. Routledge.

Brand, Amy, Liz Allen, Micah Altman, Marjorie Hlava, and Jo Scott. 2015. "Beyond Authorship: Attribution, Contribution, Collaboration, and Credit." *Learned Publishing* 28 (2).

Brunner, Ursin, and Kurt Stockinger. 2020. "Entity Matching with Transformer Architectures - a Step Forward in Data Integration." In *International Conference on Extending Database Technology*.

Cao, Hancheng, Jesse Dodge, Kyle Lo, Daniel A. McFarland, and Lucy Lu Wang. 2023. "The Rise of Open Science: Tracking the Evolution and Perceived Value of Data and Methods Link-Sharing Practices." *ArXiv* abs/2310.03193.

Carver, Jeffrey C., Nic Weber, Karthik Ram, Sandra Gesing, and Daniel S. Katz. 2022. "A Survey of the State of the Practice for Research Software in the United States." *PeerJ Computer Science* 8.

Cosden, Ian A., Kenton McHenry, and Daniel S. Katz. 2022. "Research Software Engineers: Career Entry Points and Training Gaps." *Computing in Science & Engineering* 24 (6): 14–21. https://doi.org/10.1109/MCSE.2023.3258630.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *CoRR* abs/1810.04805. http://arxiv.org/abs/1810.04805.

Ding, Jingda, Chao Liu, Qiao Zheng, and Wei Cai. 2021. "A New Method of Co-Author Credit Allocation Based on Contributor Roles Taxonomy: Proof of Concept and Evaluation Using Papers Published in PLOS ONE." *Scientometrics* 126: 7561–81. https://api.semanticscholar.org/CorpusID:237206568.

Edwards, Paul N, Steven J Jackson, Melissa K Chalmers, Geoffrey C Bowker, Christine L Borgman, David Ribes, Matt Burton, and Scout Calvert. 2013. "Knowledge Infrastructures: Intellectual Frameworks and Research Challenges."

Fan, Jianqing, Fang Han, and Han Liu. 2014. "Challenges of Big Data Analysis." *National Science Review* 1 (2): 293–314. https://doi.org/10.1093/nsr/nwt032.

Franceschet, Massimo, and Antonio Costantini. 2010. "The Effect of Scholar Collaboration on Impact and Quality of Academic Papers." *J. Informetrics* 4: 540–53. https://api.semanticscholar.org/CorpusID:17315541.

Gøtzsche, Peter C, Asbjørn Hróbjartsson, Helle Krogh Johansen, Mette T Haahr, Douglas G Altman, and An-Wen Chan. 2007. "Ghost Authorship in Industry-Initiated Randomised Trials." *PLoS Medicine* 4 (1): e19.

Haak, Laurel L, Martin Fenner, Laura Paglione, Ed Pentz, and Howard Ratner. 2012. "ORCID: A System to Uniquely Identify Researchers." *Learned Publishing* 25 (4): 259–64.

Haeussler, Carolin, and Henry Sauermann. 2013. "Credit Where Credit Is Due? The Impact of Project Contributions and Social Factors on Authorship and Inventorship." *Research Policy* 42 (3): 688–703. https://doi.org/10.1016/j.respol.2012.09.009.

Hampton, Stephanie E, Carly A Strasser, Joshua J Tewksbury, Wendy K Gram, Amber E Budden, Archer L Batcheller, Clifford S Duke, and John H Porter. 2013. "Big Data and the Future of Ecology." *Frontiers in Ecology and the Environment* 11 (3): 156–62.

Hannay, Jo Erskine, Carolyn MacLeod, Janice Singer, Hans Petter Langtangen, Dietmar Pfahl, and Greg Wilson. 2009. "How Do Scientists Develop and Use Scientific Software?" In *2009 ICSE Workshop on Software Engineering for Computational Science and Engineering*, 1–8. https://doi.org/10.1109/SECSE.2009.5069155.

Hasselbring, Wilhelm, Stephan Druskat, Jan Bernoth, Philine Betker, Michael Felderer, Stephan Ferenz, Anna-Lena Lamprecht, Jan Linxweiler, and Bernhard Rumpe. 2024. "Toward Research Software Categories." https://arxiv.org/abs/2404.14364.

Hata, Hideaki, Jin L. C. Guo, Raula Gaikovina Kula, and Christoph Treude. 2021. "Science-Software Linkage: The Challenges of Traceability Between Scientific Knowledge and Software Artifacts." *ArXiv* abs/2104.05891.

He, Pengcheng, Jianfeng Gao, and Weizhu Chen. 2021. "DeBERTaV3: Improving DeBERTa Using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing." https://arxiv.org/abs/2111.09543.

He, Pengcheng, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. "DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION." In *International Conference on Learning Representations*. https://openreview.net/forum?id=XPZIaotutsD.

Heroux, Michael A. 2022. "Research Software Science: Expanding the Impact of Research Software Engineering." *Computing in Science & Engineering* 24 (6): 22–27. https://doi.org/10.1109/MCSE.2023.3260475.

Howison, James, Ewa Deelman, Michael J. McLennan, Rafael Ferreira da Silva, and James D. Herbsleb. 2015. "Understanding the Scientific Software Ecosystem and Its Impact: Current and Future Measures." *Research Evaluation* 24 (4): 454–70. https://doi.org/10.1093/reseval/rvv014.

Jeong, Yuna, and Eunhui Kim. 2022. "Scideberta: Learning Deberta for Science Technology Documents and Fine-Tuning Information Extraction Tasks." *IEEE Access* 10: 60805–13.

Jin, Xiaolong, Benjamin W Wah, Xueqi Cheng, and Yuanzhuo Wang. 2015. "Significance and Challenges of Big Data Research." *Big Data Research* 2 (2): 59–64.

Júnior, Edilson Anselmo Corrêa, Filipi Nascimento Silva, Luciano da Fontoura Costa, and Diego Raphael Amancio. 2016. "Patterns of Authors Contribution in Scientific Manuscripts." *J. Informetrics* 11: 498–510. https://api.semanticscholar.org/CorpusID:9967627.

Katz, Daniel S., Neil P. Chue Hong, Tim Clark, August Muench, Shelley Stall, Daina R. Bouquin, Matthew Cannon, et al. 2020. "Recognizing the Value of Software: A Software Citation Guide." *F1000Research* 9.

Kelley, Aidan, and Daniel Garijo. 2021. "A Framework for Creating Knowledge Graphs of Scientific Software Metadata." *Quantitative Science Studies* 2: 1423–46.

Krafczyk, Matthew, August Shi, Adhithya Bhaskar, Darko Marinov, and Victoria Stodden. 2019. "Scientific Tests and Continuous Integration Strategies to Enhance Reproducibility in the Scientific Software Context." In *Proceedings of the 2nd International Workshop on Practical Reproducible Evaluation of Computer Systems*, 23–28. P-RECS '19. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3322790.3330595.

Larivière, Vincent, Nadine Desrochers, Benoît Macaluso, Philippe Mongeon, Adèle Paul-Hus, and Cassidy R Sugimoto. 2016. "Contributorship and Division of Labor in Knowledge Production." *Social Studies of Science* 46 (3): 417–35. https://doi.org/10.1177/0306312716650046.

Larivière, Vincent, Yves Gingras, Cassidy R. Sugimoto, and Andrew Tsou. 2014. "Team Size Matters: Collaboration and Scientific Impact Since 1900." *Journal of the Association for Information Science and Technology* 66. https://api.semanticscholar.org/CorpusID:13505127.

Larivière, Vincent, David Pontille, and Cassidy R. Sugimoto. 2020. "Investigating the Division of Scientific Labor Using the Contributor Roles Taxonomy (CRediT)." *Quantitative Science Studies*, 1–18.

Li, Kai, Chenwei Zhang, and Vincent Larivière. 2023. "Are Research Contributions Assigned Differently Under the Two Contributorship Classification Systems in PLoS ONE?" *ArXiv* abs/2310.11687. https://api.semanticscholar.org/CorpusID:264289300.

Li, Yuliang, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang Chiew Tan. 2020. "Deep Entity Matching with Pre-Trained Language Models." *Proceedings of the VLDB Endowment* 14: 50–60.

Liu, Linlin, Jianfei Yu, Junming Huang, Feng Xia, and Tao Jia. 2021. "The Dominance of Big Teams in China's Scientific Output." *Quantitative Science Studies* 2 (1): 350–62.

Liu, Xin, Chengjing Zhang, and Jiang Li. 2023. "Conceptual and Technical Work: Who Will Disrupt Science?" *Journal of Informetrics* 17 (3): 101432. https://doi.org/https://doi.org/10.1016/j.joi.2023.101432.

Lu, Chao, Yingyi Zhang, Yong-Yeol Ahn, Ying Ding, Chenwei Zhang, and Dandan Ma. 2019. "Co-contributorship Network and Division of Labor in Individual Scientific Collaborations." *Journal of the Association for Information Science and Technology* 71: 1162–78. https://api.semanticscholar.org/CorpusID:208267519.

Mayernik, Matthew S, David L Hart, Keith E Maull, and Nicholas M Weber. 2017. "Assessing and Tracing the Outcomes and Impact of Research Infrastructures." *Journal of the Association for Information Science and Technology* 68 (6): 1341–59.

Meirelles, Paulo, Carlos Santos Jr, João Miranda, Fabio Kon, Antonio Terceiro, and Christina Chavez. 2010. "A Study of the Relationships Between Source Code Metrics and Attractiveness in Free Software Projects." In *2010 Brazilian Symposium on Software Engineering*, 11–20. IEEE.

Merow, Cory, Brad L. Boyle, Brian J. Enquist, Xiao Feng, Jamie M. Kass, Brian Salvin Maitner, Brian McGill, et al. 2023. "Better Incentives Are Needed to Reward Academic Software Development." *Nature Ecology & Evolution* 7: 626–27.

Milewicz, Reed, Gustavo Pinto, and Paige Rodeghero. 2019. "Characterizing the Roles of

Contributors in Open-Source Scientific Software Projects." In *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, 421–32. https://doi.org/10.1109/MSR.2019.00069.

Muna, Demitri, Michael Alexander, Alice Allen, Richard Ashley, Daniel Asmus, Ruyman Azzollini, Michele Bannister, et al. 2016. "The Astropy Problem." https://arxiv.org/abs/1610.03159.

Naik, Cian, Cassidy R Sugimoto, Vincent Larivière, Chenlei Leng, and Weisi Guo. 2023. "Impact of Geographic Diversity on Citation of Collaborative Research." *Quantitative Science Studies* 4 (2): 442–65.

Ni, Chaoqun, Elise Smith, Haimiao Yuan, Vincent Larivière, and Cassidy R. Sugimoto. 2021. "The Gendered Nature of Authorship." *Science Advances* 7 (36): eabe4639. https://doi.org/10.1126/sciadv.abe4639.

Philippe, Olivier, Martin Hammitzsch, Stephan Janosch, Anelda van der Walt, Ben van Werkhoven, Simon Hettrick, Daniel S. Katz, et al. 2019. "Softwaresaved/International-Survey: Public Release for 2018 Results." Zenodo. https://doi.org/10.5281/zenodo.2585783.

Ram, Karthik. 2013. "Git Can Facilitate Greater Reproducibility and Increased Transparency in Science." *Source Code for Biology and Medicine* 8: 1–8.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter." *ArXiv* abs/1910.01108.

Sauermann, Henry, and Carolin Haeussler. 2017. "Authorship and Contribution Disclosures." *Science Advances* 3 (11): e1700404. https://doi.org/10.1126/sciadv.1700404.

Sharma, Nitesh Kumar, Ram Ayyala, Dhrithi Deshpande, Yesha Patel, Viorel Munteanu, Dumitru Ciorba, Viorel Bostan, et al. 2024. "Analytical Code Sharing Practices in Biomedical Research." *PeerJ Computer Science* 10: e2066.

Smith, Arfon M, Dara Norman, and Kelle Cruz. 2019. "Elevating the Role of Software as a Product of the Research Enterprise." *Bulletin of the American Astronomical Society* 51 (7): 52.

Smith, Elise. 2023. "'Technical' Contributors and Authorship Distribution in Health Science." *Science and Engineering Ethics* 29 (4): 22.

Stankovski, Aleksandar, and Daniel Garijo. 2024. "RepoFromPaper: An Approach to Extract Software Code Implementations from Scientific Publications." In *NSLP*.

Stodden, Victoria, Peixuan Guo, and Zhaokun Ma. 2013. "Toward Reproducible Computational Research: An Empirical Analysis of Data and Code Policy Adoption by Journals." *PloS One* 8 (6): e67111.

Tran, Hanh Thi Hong, Nishan Chatterjee, Senja Pollak, and Antoine Doucet. 2024. "DeBERTa Beats Behemoths: A Comparative Analysis of Fine-Tuning, Prompting, and PEFT Approaches on LegalLensNER." In *Proceedings of the Natural Legal Language Processing Workshop 2024*, 371–80.

Trisovic, Ana, Matthew K. Lau, Thomas Pasquier, and Mercè Crosas. 2021. "A Large-Scale Study on Research Code Quality and Execution." *Scientific Data* 9.

Trujillo, Milo Z, Laurent Hébert-Dufresne, and James Bagrow. 2022. "The Penumbra of Open Source: Projects Outside of Centralized Platforms Are Longer Maintained, More Academic

Table 4: Comparison of Models for Author-Developer-Account Matching

Optional Feats.	Model	Accuracy	Precision	Recall	F1
name	deberta	0.984	0.938	0.950	0.944
name, email	bert-multilingual	0.984	0.938	0.950	0.944
name, email	deberta	0.982	0.907	0.975	0.940
name	bert-multilingual	0.982	0.938	0.938	0.938
name	distilbert	0.978	0.936	0.912	0.924
name, email	distilbert	0.978	0.936	0.912	0.924
email	deberta	0.957	0.859	0.838	0.848
email	bert-multilingual	0.950	0.894	0.738	0.808
n/a	deberta	0.946	0.847	0.762	0.803
n/a	bert-multilingual	0.941	0.862	0.700	0.772
n/a	distilbert	0.856	0.000	0.000	0.000
email	distilbert	0.856	0.000	0.000	0.000

and More Collaborative." *EPJ Data Science* 11 (1): 31.

Westner, Britta U., Daniel R. McCloy, Eric Larson, Alexandre Gramfort, Daniel S. Katz, Arfon M. Smith, and invited co-signees. 2024. "Cycling on the Freeway: The Perilous State of Open Source Neuroscience Software." *ArXiv* abs/2403.19394.

Wyss, Elizabeth, Lorenzo De Carli, and Drew Davidson. 2023. "(Nothing but) Many Eyes Make All Bugs Shallow." In *Proceedings of the 2023 Workshop on Software Supply Chain Offensive Research and Ecosystem Defenses*, 53–63. SCORED '23. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3605770.3625216.

Yang, Alex J., Ying Ding, and Meijun Liu. 2024. "Female-Led Teams Produce More Innovative Ideas yet Receive Less Scientific Impact." *Quantitative Science Studies* 5 (4): 861–81. https://doi.org/10.1162/qss_a_00335.

Yu, Liqiang, Bo Liu, Qunwei Lin, Xinyu Zhao, and Chang Che. 2024. "Semantic Similarity Matching for Patent Documents Using Ensemble BERT-Related Model and Novel Text Processing Method." *arXiv Preprint arXiv:2401.06782*.

Table 5: Article Citations by Code Contributorship of Research Team Controlled by Open Access Status

Variable	**coef**	**P>|z|**	**[0.025**	**0.975]**
const ***	**0.61**	**0.00**	**0.49**	**0.73**
Total Authors ***	**0.07**	**0.00**	**0.06**	**0.08**
Code-Contrib. Authors	0.05	0.24	-0.04	0.14
Code-Contrib. Non-Authors	0.00	0.96	-0.03	0.03
Years Since Publication ***	**0.38**	**0.00**	**0.37**	**0.39**
Is Open Access ***	**0.42**	**0.00**	**0.30**	**0.54**
Code-Contrib. Authors * Is Open Access	-0.01	0.82	-0.10	0.08
Code-Contrib. Non-Authors * Is Open Access	-0.00	0.94	-0.03	0.03

Table 6: Article Citations by Code Contributorship of Research Team Controlled by Domain

Variable	coef	P>|z|	[0.025	0.975]
const ***	**0.88**	**0.00**	**0.75**	**1.01**
Total Authors ***	**0.07**	**0.00**	**0.06**	**0.08**
Code-Contrib. Authors	0.03	0.60	-0.08	0.13
Code-Contrib. Non-Authors	0.01	0.61	-0.04	0.07
Years Since Publication ***	**0.40**	**0.00**	**0.39**	**0.41**
Domain Life Sciences ***	**-0.21**	**0.01**	**-0.36**	**-0.06**
Domain Physical Sciences ***	**0.14**	**0.03**	**0.01**	**0.26**
Domain Social Sciences ***	**-0.18**	**0.03**	**-0.34**	**-0.02**
Code-Contrib. Authors * Domain Life Sciences	0.07	0.29	-0.06	0.19
Code-Contrib. Authors * Domain Physical Sciences	0.00	0.93	-0.10	0.11
Code-Contrib. Authors * Domain Social Sciences	0.10	0.14	-0.03	0.23
Code-Contrib. Non-Authors * Domain Life Sciences	-0.04	0.23	-0.11	0.03
Code-Contrib. Non-Authors * Domain Physical Sciences	-0.02	0.57	-0.07	0.04
Code-Contrib. Non-Authors * Domain Social Sciences	-0.04	0.31	-0.11	0.03

Table 7: Article Citations by Code Contributorship of Research Team Controlled by Article Type

Variable	coef	P>|z|	[0.025	0.975]
const ***	**0.53**	**0.00**	**0.45**	**0.61**
Total Authors ***	**0.07**	**0.00**	**0.06**	**0.08**
Code-Contrib. Authors	-0.03	0.20	-0.09	0.02
Code-Contrib. Non-Authors ***	**-0.03**	**0.00**	**-0.05**	**-0.01**
Years Since Publication ***	**0.40**	**0.00**	**0.39**	**0.41**
Article Type Research Article ***	**0.47**	**0.00**	**0.40**	**0.55**
Article Type Software Article ***	**-0.47**	**0.00**	**-0.73**	**-0.22**
Code-Contrib. Authors * Article Type Research Article ***	**0.10**	**0.00**	**0.05**	**0.16**
Code-Contrib. Authors * Article Type Software Article	-0.06	0.37	-0.19	0.07
Code-Contrib. Non-Authors * Article Type Research Article ***	**0.04**	**0.00**	**0.02**	**0.06**
Code-Contrib. Non-Authors * Article Type Software Article	0.09	0.24	-0.06	0.24

Table 8: Counts of Code-Contributing Authors ('Coding') as well as Total Authors by Position and Bonferroni Corrected p-values from Post-Hoc Binomial Tests

Control	Subset	Position	Coding	Total	p
Domain	**Health Sciences**	First	1575	2401	0.000***
		Middle	540	11491	0.000***
		Last	234	2349	0.000***
	Life Sciences	First	2586	3895	0.000***
		Middle	852	11875	0.000***
		Last	491	3784	0.000***
	Physical Sciences	First	28919	41987	0.000***
		Middle	10507	111738	0.000***
		Last	3433	40410	0.000***
	Social Sciences	First	2813	4038	0.000***
		Middle	1021	9249	0.000***
		Last	411	3855	0.000***
Article Type	**Preprint**	First	13421	19523	0.000***
		Middle	5134	52925	0.000***
		Last	1493	18598	0.000***
	Research Article	First	21940	32081	0.000***
		Middle	7345	89991	0.000***
		Last	2909	31188	0.000***
	Software Article	First	532	717	0.000***
		Middle	441	1437	0.000***
		Last	167	612	0.000***
Open Access Status	**Closed Access**	First	2637	3742	0.000***
		Middle	918	10965	0.000***
		Last	279	3642	0.000***
	Open Access	First	33256	48579	0.000***
		Middle	12002	133388	0.000***
		Last	4290	46756	0.000***
Overall	**Overall**	First	35893	52321	0.000***
		Middle	12920	144353	0.000***
		Last	4569	50398	0.000***

Table 9: Counts of Code-Contributing Authors ('Coding') as well as Total Authors by Corresponding Status and Bonferroni Corrected p-values from Post-Hoc Binomial Tests

Control	Subset	Is Corresponding	Coding	Total	p
Domain	**Life Sciences**	Corresponding	1772	8019	0.000***
		Not Corresponding	2157	11535	0.000***
	Physical Sciences	Corresponding	5248	12487	0.000***
		Not Corresponding	37611	181648	0.000***
	Social Sciences	Corresponding	803	2458	0.000***
		Not Corresponding	3442	14684	0.000***
Article Type	**Preprint**	Corresponding	772	1036	0.000***
		Not Corresponding	19276	90010	0.000***
	Research Article	Corresponding	7716	27339	0.000***
		Not Corresponding	24478	125921	0.000***
	Software Article	Corresponding	213	438	1.000
		Not Corresponding	927	2328	0.000***
Open Access Status	**Closed Access**	Corresponding	253	468	0.174
		Not Corresponding	3581	17881	0.000***
	Open Access	Corresponding	8448	28345	0.000***
		Not Corresponding	41100	200378	0.000***
Overall	**Overall**	Corresponding	8701	28813	0.000***
		Not Corresponding	44681	218259	0.000***

Table 10: Code-Contributing Authors H-Index by Coding Status

Variable	coef	P>|z|	[0.025	0.975]
const ***	**3.19**	**0.00**	**3.18**	**3.20**
Works Count ***	**0.00**	**0.00**	**0.00**	**0.00**
Any Coding ***	**-0.32**	**0.00**	**-0.33**	**-0.30**
Majority Coding ***	**-0.77**	**0.00**	**-0.79**	**-0.74**
Always Coding ***	**-0.97**	**0.00**	**-1.02**	**-0.92**

Table 11: Researcher H-Index by Coding Status Controlled by Most Freq. Author Position

Variable	**coef**	**P>|z|**	**[0.025**	**0.975]**
const ***	**2.38**	**0.00**	**2.31**	**2.44**
Works Count ***	**0.00**	**0.00**	**0.00**	**0.00**
Any Coding ***	**0.14**	**0.00**	**0.05**	**0.22**
Majority Coding	-0.07	0.08	-0.14	0.01
Always Coding ***	**-0.24**	**0.00**	**-0.33**	**-0.16**
Common Author Position Last ***	**1.05**	**0.00**	**0.98**	**1.11**
Common Author Position Middle ***	**0.74**	**0.00**	**0.68**	**0.81**
Any Coding * Common Author Position Last ***	**-0.28**	**0.00**	**-0.37**	**-0.19**
Any Coding * Common Author Position Middle ***	**-0.45**	**0.00**	**-0.53**	**-0.36**
Majority Coding * Common Author Position Last ***	**-0.35**	**0.00**	**-0.45**	**-0.26**
Majority Coding * Common Author Position Middle ***	**-0.61**	**0.00**	**-0.69**	**-0.52**
Always Coding * Common Author Position Last ***	**-0.37**	**0.00**	**-0.51**	**-0.22**
Always Coding * Common Author Position Middle ***	**-0.51**	**0.00**	**-0.64**	**-0.38**

Table 12: Researcher H-Index by Coding Status Controlled by Most Freq. Domain

Variable	**coef**	**P>|z|**	**[0.025**	**0.975]**
const ***	**3.32**	**0.00**	**3.29**	**3.36**
Works Count ***	**0.00**	**0.00**	**0.00**	**0.00**
Any Coding ***	**-0.39**	**0.00**	**-0.48**	**-0.31**
Majority Coding ***	**-1.45**	**0.00**	**-1.65**	**-1.25**
Always Coding ***	**-1.22**	**0.00**	**-1.58**	**-0.86**
Common Domain Life Sciences ***	**0.10**	**0.00**	**0.05**	**0.15**
Common Domain Physical Sciences ***	**-0.15**	**0.00**	**-0.18**	**-0.11**
Common Domain Social Sciences ***	**-0.16**	**0.00**	**-0.22**	**-0.10**
Any Coding * Common Domain Life Sciences	0.11	0.07	-0.01	0.22
Any Coding * Common Domain Physical Sciences	0.08	0.09	-0.01	0.17
Any Coding * Common Domain Social Sciences	0.01	0.94	-0.14	0.15
Majority Coding * Common Domain Life Sciences ***	**0.81**	**0.00**	**0.58**	**1.04**
Majority Coding * Common Domain Physical Sciences ***	**0.70**	**0.00**	**0.50**	**0.90**
Majority Coding * Common Domain Social Sciences ***	**0.72**	**0.00**	**0.46**	**0.98**
Always Coding * Common Domain Life Sciences	0.33	0.12	-0.08	0.74
Always Coding * Common Domain Physical Sciences	0.25	0.18	-0.12	0.62
Always Coding * Common Domain Social Sciences	0.30	0.17	-0.13	0.73

8 Appendix

8.1 Full Comparison of Models and Optional Features for Author-Developer-Account Matching

8.2 Linear Models for Software Development Dynamics Within Research Teams

8.2.1 No Controls

8.2.2 Controlled by Open Access Status

8.2.3 Controlled by Domain

8.2.4 Controlled by Article Type

8.3 Post-Hoc Tests for Coding vs Non-Coding Authors by Position

8.4 Post-Hoc Tests for Coding vs Non-Coding Authors by Corresponding Status

8.5 Linear Models for Characterizing Code-Contributing Author H-Index

8.5.1 No Controls

8.5.2 Controlled by Author Position

8.5.3 Controlled by Domain

8.5.4 Controlled by Article Type

Table 13: Researcher H-Index by Coding Status Controlled by Most Freq. Article Type

Variable	**coef**	**P>|z|**	**[0.025**	**0.975]**
const ***	**3.10**	**0.00**	**3.09**	**3.11**
Works Count ***	**0.00**	**0.00**	**0.00**	**0.00**
Any Coding ***	**-0.30**	**0.00**	**-0.32**	**-0.27**
Majority Coding ***	**-0.77**	**0.00**	**-0.81**	**-0.73**
Always Coding ***	**-0.99**	**0.00**	**-1.07**	**-0.92**
Common Article Type Research Article ***	**0.18**	**0.00**	**0.17**	**0.20**
Common Article Type Software Article ***	**0.22**	**0.00**	**0.12**	**0.33**
Any Coding * Common Article Type Research Article	-0.02	0.15	-0.05	0.01
Any Coding * Common Article Type Software Article	0.19	0.06	-0.01	0.39
Majority Coding * Common Article Type Research Article	0.01	0.80	-0.05	0.06
Majority Coding * Common Article Type Software Article ***	**0.36**	**0.00**	**0.19**	**0.54**
Always Coding * Common Article Type Research Article	0.00	0.93	-0.10	0.10
Always Coding * Common Article Type Software Article ***	**0.37**	**0.00**	**0.15**	**0.58**