Operational Evidence and Outcomes
Volume 1 · Number 1 · 2026

Visible Patch Sets and Merge Latency in a Bounded Gerrit Cohort

Edward Hayes[1] · Ethan Reed[2] · James Harrison[3]

[1] Canada · [2] United States · [3] Australia

Original publication: November 8, 2026 | Article 575

Abstract

Final patch size does not describe the history through which a change reaches submission. Among 486 merged changes from a capped Gerrit query, visible patch-set count accompanied longer creation-to-submission intervals after final edit volume and calendar adjustment. Its log coefficient was 1.132 in the mean regression and 1.200 in the median regression. The association therefore persisted beyond an analysis focused on average log time. However, a longer open interval can itself allow more revisions, and the API query selected recently updated matching records. The evidence supports revision history as a descriptive workflow signal, not a causal estimate of review delay or a measure of coding-assistant quality.

Observational software-repository draft. Results describe public change metadata; causal and publication review is pending.

Visible Patch Sets and Merge Latency in a Bounded Gerrit Cohort

Observational software-repository draft. Results describe public change metadata; causal and publication review is pending.

Abstract

Final patch size does not describe the history through which a change reaches submission. Among 486 merged changes from a capped Gerrit query, visible patch-set count accompanied longer creation-to-submission intervals after final edit volume and calendar adjustment. Its log coefficient was 1.132 in the mean regression and 1.200 in the median regression. The association therefore persisted beyond an analysis focused on average log time. However, a longer open interval can itself allow more revisions, and the API query selected recently updated matching records. The evidence supports revision history as a descriptive workflow signal, not a causal estimate of review delay or a measure of coding-assistant quality.

1. The final patch omits a process

Two merged changes can alter the same number of lines yet have very different development histories. One may be accepted after a single upload; another may be revised repeatedly over weeks. A final-diff measure captures the submitted artifact but omits that sequence. We examine how strongly visible patch-set count accompanies elapsed creation-to-submission time after adjustment for final edit volume and basic calendar characteristics in one public Gerrit project.

Modern code-review research distinguishes the expectations and outcomes of review from simple artifact inspection (Bacchelli and Bird, 2013). Mining repository records can help describe that process, but the warnings in Bird et al., 2009 are pertinent: recorded timestamps and histories reflect platform conventions and selection mechanisms. Our analysis uses a bounded public API return, not a complete account of all engineering work. It does not infer that an observed patch set corresponds to one reviewer request or one independent attempt at a solution.

Preference-based fine-tuning for coding assistants, such as Tao and Kamata, 2026, motivates understanding which changes are useful in a workflow. The current study supplies no preference pairs and trains no assistant. Its narrower contribution is to show that final patch size and visible revision history contain different descriptive information. Treating the revision count as a direct measure of assistant mistakes would exceed the available data because these changes do not carry a verified assistant-use indicator.

2. A workflow association, not a productivity ranking

Software-engineering reporting guidance in Kitchenham et al., 2002 supports defining the observational unit and sampling frame precisely. The unit here is a merged change in the Gerrit project's master branch. The relevant response is elapsed clock time from recorded creation to submission. It includes waiting, author availability, automated checks, and other unobserved stages. It is not developer effort, active review time, or a direct productivity measure.

Issue and change labels can also be unreliable proxies for substantive work. Herzig et al., 2013 shows why classification choices matter in software mining. We therefore do not classify these changes as bug fixes, features, or refactors from unverified subject text. The absence of that classification limits adjustment, but

is preferable to inventing a taxonomy. Change size remains the sum of recorded insertions and deletions in the final revision, with no claim that it measures semantic complexity completely.

SWE-bench (Jimenez et al., 2023), MBPP (Austin et al., 2021), and rigorous code-generation evaluation in Liu et al., 2023 address program behavior under different task designs. They are useful contrasts to this repository-time outcome. Passing tests, matching preferences, and reaching submission are related concerns but are not interchangeable endpoints. A system could generate functionally correct code that takes time to integrate, or produce a small final patch after substantial earlier revision.

The same distinction applies to representation and adaptation methods. GraphCodeBERT (Guo et al., 2020), InCoder (Fried et al., 2022), StarCoder (Li et al., 2023), and DeepSeek-Coder (Guo et al., 2024) concern code modeling capabilities. Direct Preference Optimization (Rafailov et al., 2023) and LoRA (Hu et al., 2022) concern learning or adaptation. None provides a reason to label a historical patch-set count as a direct assessment of those capabilities when model usage is unobserved.

3. Research model and propositions

Our first proposition is that the logarithm of visible revision count has a positive conditional association with log one plus elapsed hours. The logarithmic transformation recognizes that an increase from one to two visible uploads need not be equivalent to an increase from twenty to twenty-one. The second proposition asks whether the association persists in a conditional median regression, which responds differently to a long elapsed-time tail than ordinary least squares.

Both regressions include log one plus final changed lines, an indicator for weekend creation, and creation-month indicators. These are descriptive adjustments, not a complete confounder set. Revision count is measured over the change history, so a longer elapsed interval can create more opportunities for revision. The causal-diagram perspective in Greenland et al., 1999 makes this reverse direction central. Matching concepts in Stuart, 2010 would not remove it without a better-defined intervention and temporal design.

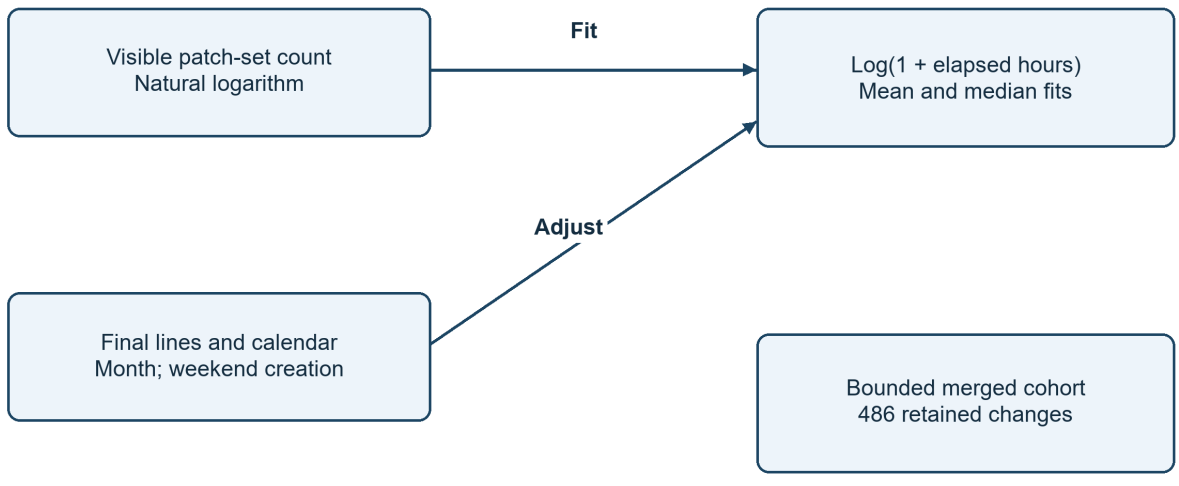


Figure 1. Visible revision count, final edit volume, and calendar terms describe mean and median log elapsed time. The diagram represents fitted associations rather than a directional intervention on revision behavior.

4. Query return and eligibility

The source is Gerrit's public changes endpoint, whose documentation defines returned fields and ordering (Gerrit Code Review, 2026). The query requested merged changes in project gerrit, branch master, after 1 January 2025 and before 1 July 2025, with a limit of 500 and all visible revision metadata. Gerrit returns matching changes ordered by most recent update. The cap therefore creates an update-ordered subset, not a random sample or complete six-month census.

We check project, branch, and merged status in every returned record. We then require the actual submitted timestamp to fall between 1 January inclusive and 1 July exclusive, and the creation-to-submission interval to be nonnegative. Fourteen records fail the timestamp-window or nonnegative-duration restriction, leaving 486 changes. The additional local restriction is necessary because a server search predicate and the exact response field used as the outcome endpoint should not be assumed identical without verification.

Visible revisions are counted from the returned revision mapping. In this acquired cohort, the highest revision number equals the number of visible revision entries for every retained change. That consistency check does not prove that every real-world work step is represented. Private drafts, local edits, discussions outside the platform, or unavailable historical details can remain absent. We call the measure visible patch sets throughout, avoiding the stronger phrase review rounds.

The median count is three, the mean is about four, and the maximum is forty-two. Elapsed hours have median 11.74, mean 78.97, and maximum 2081.66. The large gap between mean and median motivates the declared response transformation and the separate median specification. It does not justify removing the long-duration observations merely because they complicate a simple productivity narrative.

5. Estimation choices

The mean model fits ordinary least squares to log one plus elapsed hours. Its covariance clusters changes by creation date, giving 132 day clusters, with a finite-sample correction and t-based intervals. Changes created on one day can share release pressures, staff availability, or system conditions. Cameron and Miller, 2015 motivates allowing their residuals to co-vary; White, 1980 provides the broader heteroskedasticity-consistent foundation. This does not address all dependence among authors or review teams.

The median model uses the same covariates and a quantile of one half, following the conditional-quantile framework of Koenker and Bassett, 1978. It converges in 63 iterations under a declared tolerance and iteration limit. Its reported robust asymptotic covariance is not clustered by creation day. Consequently, agreement in coefficient direction is a response-tail sensitivity result, not a claim that the two confidence intervals rest on the same dependence assumption.

Repeated-measure approaches such as Laird and Ware, 1982 could be relevant if a verified author or team hierarchy were modeled. We do not build that hierarchy from names or infer personal characteristics. The robust-estimation contributions of Huber, 1964, Hampel, 1974, and Rousseeuw, 1984 also differ from conditional median regression. They are cited to clarify alternative notions of robustness; we do not claim that every outlier-related concern has been solved by adding one median fit.

Table 1. Selected covariates in mean and median log elapsed-time regressions. Both include creation-month indicators. Mean-model intervals cluster by creation day; median-model intervals use unclustered robust quantile covariance.

Model	Term	Estimate (SE)	95% CI	N
M1	Log(1 + final lines)	0.1036 (0.0500)	[0.0048, 0.2025]	486
M1	Log visible patch sets	1.1317 (0.1087)	[0.9168, 1.3467]	486
M1	Weekend creation	1.1306 (0.2970)	[0.5431, 1.7181]	486
M1: fit	R-squared = 0.3702	-	-	486
M2	Log(1 + final lines)	0.1847 (0.0606)	[0.0657, 0.3038]	486
M2	Log visible patch sets	1.1999 (0.1320)	[0.9405, 1.4593]	486
M2	Weekend creation	1.5206 (0.3276)	[0.8768, 2.1643]	486
M2: fit	Quantile pseudo R-squared = 0.2576	-	-	486

Model labels: M1 = Mean log elapsed time; M2 = Median log elapsed time.

6. Revision history remains informative after final size

In the mean model, log visible revisions has coefficient 1.1317, standard error 0.1087, and interval 0.9168 to 1.3467. The log final-line coefficient is 0.1036, with interval 0.0048 to 0.2025. The weekend-creation coefficient is 1.1306, with interval 0.5431 to 1.7181. The full model R-squared is 0.3702. These are conditional log-scale associations and should not be translated directly into saved or wasted engineering hours.

The median regression gives a revision coefficient of 1.1999, with interval 0.9405 to 1.4593. Its changed-line coefficient is 0.1847, with interval 0.0657 to 0.3038. The weekend coefficient is 1.5206, with interval 0.8768 to 2.1643. Quantile pseudo R-squared is 0.2576, which is not directly comparable to the ordinary model's explained-variance statistic. Both models use the same 486 records but summarize different conditional features of their time distribution.

The similar revision coefficients make it unlikely that the observed direction is only a consequence of fitting the ordinary mean to a few very long changes. That statement remains limited: the median model can still be affected by covariate leverage, functional-form error, and selection. Influence analysis as developed by Cook, 1977 could guide further diagnostics, but a stable sign across two estimators does not identify the causal cost of uploading another patch set.

7. Selection and temporal interpretation

Only merged changes are analyzed. Abandoned or still-open changes may have different revision histories and durations. We do not treat them as absent failures, impute their eventual submission times, or fit a survival model to a sample that excludes them. Missing-data reasoning in Little and Rubin, 2002 is useful for recognizing that outcome-based availability changes the population. The model describes retained merged records, not all proposed work.

The update-order cap creates another selection layer. A change submitted during the target interval can receive a later update and appear nearer the top of the query return. Other eligible changes may be omitted once 500 records are reached. Without complete pagination and a defined extraction protocol, the cohort

cannot support a project-wide six-month rate estimate. Its usefulness is as a reproducible bounded example of how final artifact size and visible process history differ.

Measurement-error concepts in Carroll et al., 2006 also caution against interpreting final lines as a sufficient complexity control. Large conceptual changes can require few lines, while mechanical replacements can require many. The revision coefficient can therefore absorb unobserved task difficulty, review standards, and author availability. An intervention study would need to define what reducing revisions means without simultaneously changing those factors, and would need a workflow outcome beyond elapsed clock time alone.

The guidance of Wasserstein and Lazar, 2016 supports reporting the magnitude, interval, and design together instead of turning positive coefficients into a universal claim about inefficiency. No author ranking or team-performance judgment follows from this analysis. The public response contains account-related fields, but the processed research table retains only the nonpersonal identifiers and derived workflow variables required for reproduction.

8. Evidence preserved for a narrower conclusion

The creation-month adjustment is tied to when a change enters the recorded workflow, rather than the month when it finally merges. These dates can differ, especially for the long-duration tail. Month indicators can absorb broad calendar differences in the retained sample, but cannot reconstruct holidays, working hours, reviewer availability, or release deadlines. The weekend indicator has the same limitation: it labels the creation timestamp's day, not the amount of weekend work performed or the time zone of every contributor.

Log one plus elapsed hours is used because the response includes short positive intervals and a long tail. A coefficient in this model does not equal a constant increase in raw hours at every starting duration. Translating it into an expected raw-time change would require assumptions about the conditional residual distribution and the relevant covariate values. We therefore retain the log-scale coefficients and the original-hour descriptive summaries instead of offering a deceptively precise estimate of how many engineering hours one additional upload costs.

The visible revision count can also mix qualitatively different events. An upload may address review feedback, incorporate a rebase, update generated content, or reflect an author's own revision before a reviewer responds. The current extraction does not classify those events. A more detailed process study could use messages and revision timestamps to distinguish them, with a reviewed coding scheme and explicit treatment of automated activity. That would materially improve interpretation, but it would be a new data collection and annotation task rather than information already contained in the present coefficient.

The positive association is therefore potentially useful for workload description or for selecting cases for qualitative process review. It is much less suitable for individual performance assessment. Penalizing a contributor for many patch sets could encourage larger, less transparent local edits or discourage useful iteration. The study does not evaluate those incentives, and it supplies no normative optimum for revision count. Its role is to make a process signal visible while preserving uncertainty about why the signal and elapsed time move together.

The provenance record follows Gebru et al., 2021 by identifying the endpoint, query cap, output ordering, and timestamp exclusions. Wilkinson et al., 2016 motivates retaining the public snapshot, processed change table, formulas, and numerical results with local hashes. This documentation allows another

analyst to reproduce the bounded cohort even if the live server later changes. It does not imply that the public source grants unrestricted redistribution of every account-related field.

A more informative future design would capture complete query pagination, distinguish active review from waiting, and record when an assistant contributed to a change. Those additions could support a properly defined evaluation of preference-trained assistants. For now, the evidence shows that visible revision history has a substantial association with elapsed submission time after final line count and calendar adjustment. It remains a workflow descriptor whose causal interpretation is unresolved.

References

1. Rafailov, Rafael; Sharma, Archit; Mitchell, Eric; Ermon, Stefano; Manning, Christopher D.; Finn, Chelsea. (2023). Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290. https://arxiv.org/abs/2305.18290.

2. Gebru, Timnit, et al. "Datasheets for datasets" *Communications of the ACM*, 2021. https://doi.org/10.1145/3458723.

3. Li, Raymond; Allal, Loubna Ben; Zi, Yangtian; Muennighoff, Niklas; Kocetkov, Denis; Mou, Chenghao; Marone, Marc; Akiki, Christopher; Li, Jia; Chim, Jenny; Liu, Qian; Zheltonozhskii, Evgenii; Zhuo, Terry Yue; Wang, Thomas; Dehaene, Olivier; Davaadorj, Mishig; Lamy-Poirier, Joel; Monteiro, João; Shliazhko, Oleh; Gontier, Nicolas; Meade, Nicholas; Zebaze, Armel; Yee, Ming-Ho; Umapathi, Logesh Kumar; Zhu, Jian; Lipkin, Benjamin; Oblokulov, Muhtasham; Wang, Zhiruo; Murthy, Rudra; Stillerman, Jason; Patel, Siva Sankalp; Abulkhanov, Dmitry; Zocca, Marco; Dey, Manan; Zhang, Zhihan; Fahmy, Nour; Bhattacharyya, Urvashi; Yu, Wenhao; Singh, Swayam; Luccioni, Sasha; Villegas, Paulo; Kunakov, Maxim; Zhdanov, Fedor; Romero, Manuel; Lee, Tony; Timor, Nadav; Ding, Jennifer; Schlesinger, Claire; Schoelkopf, Hailey; Ebert, Jan; Dao, Tri; Mishra, Mayank; Gu, Alex; Robinson, Jennifer; Anderson, Carolyn Jane; Dolan-Gavitt, Brendan; Contractor, Danish; Reddy, Siva; Fried, Daniel; Bahdanau, Dzmitry; Jernite, Yacine; Ferrandis, Carlos Muñoz; Hughes, Sean; Wolf, Thomas; Guha, Arjun; von Werra, Leandro; de Vries, Harm. (2023). StarCoder: may the source be with you!. arXiv:2305.06161. https://arxiv.org/abs/2305.06161.

4. Wilkinson, Mark D.; Dumontier, Michel; Aalbersberg, IJsbrand Jan; Appleton, Gabrielle; Axton, Myles; Baak, Arie; Blomberg, Niklas; Boiten, Jan-Willem; da Silva Santos, Luiz Bonino; Bourne, Philip E.; Bouwman, Jildau; Brookes, Anthony J.; Clark, Tim; Crosas, Mercè; Dillo, Ingrid; Dumon, Olivier; Edmunds, Scott; Evelo, Chris T.; Finkers, Richard; Gonzalez-Beltran, Alejandra; Gray, Alasdair J.G.; Groth, Paul; Goble, Carole; Grethe, Jeffrey S.; Heringa, Jaap; 't Hoen, Peter A.C; Hooft, Rob; Kuhn, Tobias; Kok, Ruben; Kok, Joost; Lusher, Scott J.; Martone, Maryann E.; Mons, Albert; Packer, Abel L.; Persson, Bengt; Rocca-Serra, Philippe; Roos, Marco; van Schaik, Rene; Sansone, Susanna-Assunta; Schultes, Erik; Sengstag, Thierry; Slater, Ted; Strawn, George; Swertz, Morris A.; Thompson, Mark; van der Lei, Johan; van Mulligen, Erik; Velterop, Jan; Waagmeester, Andra; Wittenburg, Peter; Wolstencroft, Katherine; Zhao, Jun; Mons, Barend. (2016). The FAIR Guiding Principles for scientific data management and stewardship. Scientific Data. https://doi.org/10.1038/sdata.2016.18.

5. White, Halbert. "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity." *Econometrica*, vol. 48, no. 4, 1980, pp. 817-838. https://doi.org/10.2307/1912934.

6. Herzig, Kim; Just, Sascha; Zeller, Andreas. (2013). It's not a bug, it's a feature: How misclassification impacts bug prediction. https://doi.org/10.1109/ICSE.2013.6606585.

7. Tao, Hongmo; Kamata, Hiroyuki. (2026). Teaching Coding Assistants What to Change: Fine-Tuning from Pairwise Preferences. Proceedings of the Second Workshop for REsearch on Agent Language Models (REALM 2026), 1363-1374. https://doi.org/10.18653/2026.realm-1.95.

8. Guo, Daya; Ren, Shuo; Lu, Shuai; Feng, Zhangyin; Tang, Duyu; Liu, Shujie; Zhou, Long; Duan, Nan; Svyatkovskiy, Alexey; Fu, Shengyu; Tufano, Michele; Deng, Shao Kun; Clement, Colin; Drain, Dawn; Sundaresan, Neel; Yin, Jian; Jiang, Daxin; Zhou, Ming. (2020). GraphCodeBERT: Pre-training Code Representations with Data Flow. arXiv:2009.08366. https://arxiv.org/abs/2009.08366.

9. Hu, Edward J., et al. "LoRA: Low-Rank Adaptation of Large Language Models." *International Conference on Learning Representations*, 2022. https://openreview.net/forum?id=nZeVKeeFYf9.

10. Carroll, Raymond J., David Ruppert, Leonard A. Stefanski, and Ciprian M. Crainiceanu. "Measurement Error in Nonlinear Models." 2006. https://doi.org/10.1201/9781420010138.

11. Laird, Nan M. and James H. Ware. "Random-Effects Models for Longitudinal Data." *Biometrics*, vol. 38, no. 4, 1982, pp. 963. https://doi.org/10.2307/2529876.

12. Kitchenham, Barbara A.; Pfleeger, Shari Lawrence; Pickard, Lesley M.; Jones, Peter W.; Hoaglin, David C.; El Emam, Khaled; Rosenberg, Jarrett. (2002). Preliminary guidelines for empirical research in software engineering. https://doi.org/10.1109/TSE.2002.1027796.

13. Wasserstein, Ronald L.; Lazar, Nicole A.. (2016). The ASA Statement on p -Values: Context, Process, and Purpose. The American Statistician. https://doi.org/10.1080/00031305.2016.1154108.

14. Bird, Christian; Rigby, Peter C.; Barr, Earl T.; Hamilton, David J.; German, Daniel M.; Devanbu, Premkumar. (2009). The promises and perils of mining git. https://doi.org/10.1109/MSR.2009.5069475.

15. Gerrit Code Review. (2026). Gerrit Code Review: changes REST API and query output definitions. Official resource, accessed 6 November 2026. https://gerrit-review.googlesource.com/Documentation/rest-api-changes.html.

16. Huber, Peter J.. (1964). Robust Estimation of a Location Parameter. The Annals of Mathematical Statistics. https://doi.org/10.1214/aoms/1177703732.

17. Guo, Daya; Zhu, Qihao; Yang, Dejian; Xie, Zhenda; Dong, Kai; Zhang, Wentao; Chen, Guanting; Bi, Xiao; Wu, Y.; Li, Y. K.; Luo, Fuli; Xiong, Yingfei; Liang, Wenfeng. (2024). DeepSeek-Coder: When the Large Language Model Meets Programming -- The Rise of Code Intelligence. arXiv:2401.14196. https://arxiv.org/abs/2401.14196.

18. Rousseeuw, Peter J.. (1984). Least Median of Squares Regression. Journal of the American Statistical Association. https://doi.org/10.1080/01621459.1984.10477105.

19. Cameron, A. Colin, and Douglas L. Miller. “A Practitioner’s Guide to Cluster-Robust Inference.” *Journal of Human Resources*, vol. 50, no. 2, 2015, pp. 317-372. https://doi.org/10.3368/jhr.50.2.317.

20. Koenker, Roger; Bassett, Gilbert. (1978). Regression Quantiles. Econometrica. https://doi.org/10.2307/1913643.

21. Austin, Jacob; Odena, Augustus; Nye, Maxwell; Bosma, Maarten; Michalewski, Henryk; Dohan, David; Jiang, Ellen; Cai, Carrie; Terry, Michael; Le, Quoc; Sutton, Charles. (2021). Program Synthesis with Large Language Models. arXiv:2108.07732. https://arxiv.org/abs/2108.07732.

22. Hampel, Frank R.. (1974). The Influence Curve and its Role in Robust Estimation. Journal of the American Statistical Association. https://doi.org/10.1080/01621459.1974.10482962.

23. Fried, Daniel; Aghajanyan, Armen; Lin, Jessy; Wang, Sida; Wallace, Eric; Shi, Freda; Zhong, Ruiqi; Yih, Wen-tau; Zettlemoyer, Luke; Lewis, Mike. (2022). InCoder: A Generative Model for Code Infilling and Synthesis. arXiv:2204.05999. https://arxiv.org/abs/2204.05999.

24. Liu, Jiawei; Xia, Chunqiu Steven; Wang, Yuyao; Zhang, Lingming. (2023). Is Your Code Generated by ChatGPT Really Correct? Rigorous Evaluation of Large Language Models for Code Generation. arXiv:2305.01210. https://arxiv.org/abs/2305.01210.

25. Little, Roderick J. A., and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 2002. https://doi.org/10.1002/9781119013563.

26. Jimenez, Carlos E.; Yang, John; Wettig, Alexander; Yao, Shunyu; Pei, Kexin; Press, Ofir; Narasimhan, Karthik. (2023). SWE-bench: Can Language Models Resolve Real-World GitHub Issues?. arXiv:2310.06770. https://arxiv.org/abs/2310.06770.

27. Stuart, Elizabeth A. "Matching Methods for Causal Inference: A Review and a Look Forward." *Statistical Science*, vol. 25, no. 1, 2010, pp. 1-21. https://doi.org/10.1214/09-sts313.

28. Bacchelli, Alberto; Bird, Christian. (2013). Expectations, outcomes, and challenges of modern code review. 2013 35th International Conference on Software Engineering (ICSE). https://doi.org/10.1109/icse.2013.6606617.

29. Greenland, Sander, Judea Pearl, and James M. Robins. “Causal Diagrams for Epidemiologic Research.” *Epidemiology*, vol. 10, no. 1, 1999, pp. 37-48. https://doi.org/10.1097/00001648-199901000-00008.

30. Cook, R. Dennis. (1977). Detection of Influential Observation in Linear Regression. Technometrics. https://doi.org/10.1080/00401706.1977.10489493.