(tentative title) Covid projects on GitHub

Mitchell Joseph[1] and James Bagrow[1,2,*]

[1]Vermont Complex Systems Center, University of Vermont, Burlington, VT, United States
[2]Mathematics & Statistics, University of Vermont, Burlington, VT, United States
[*]Corresponding author. Email: james.bagrow@uvm.edu, Homepage: bagrow.com

April 10, 2025

Abstract Millions of developers utilize GitHub's platform to build, deploy, and maintain open source software everyday. On March 11, 2020 when the World Health Organization (WHO) declared COVID-19 to be a global pandemic, there was a noticeable shift in projects being hosted on GitHub's platform [1]. People from around the world began working together in a concerted effort to develop new tools and services to combat the disease. In this paper we explore how COVID themed teams differed from non-COVID themed teams during the same time period by analyzing metrics like team focus, workload, experiential diversity, and apply survival analysis to assess the longevity of these projects.

1 Introduction

Researchers have been fascinated with the life cycle of open source projects for many years [2]. Since the earlier days of the open source software revolution people were determined to understand the evolution of these projects [3]. Questions naturally arose around the motivating factors that led people to join these open source teams and volunteer their time given that there are no monetary incentives [4] [5]. While other studies have been dedicated to looking into what makes someone a long-term contributor [6].

At the same time that researchers were diving into the motivational factors, others were focusing on analyzing the social networks of these open source projects. Similar to [3], by focusing on communication patterns [7] were able to develop centrality measures to better understand the social structures of teams. This work was then expanded upon by [8] which took a longitudinal approach to assess how these social structures changed over time. [9] further looked into successful open source projects and analyzed the self organizing properties that exist. However, it should be noted that just because an open source project has an evolved network structure does not necessarily correlate to better performance [10].

Coming up with metrics to measure the success and performance of a project has been made easy on sites like GitHub since every repository has an associated star rating, where every user can bookmark a repository they find interesting by clicking on a star. The number of people who have bookmarked a project are sometimes referred to as stargazers, and it turns out that counting the number of stargazers is actually an effective metric for evaluating project success [11]. [12] expanded upon this by looking into the factors that go into making certain repositories popular like choice of programming language. [13] were able to show that certain Java database frameworks were more likely to survive than others. However, it should be noted that success comes in many forms. [14] looked looked at team support with regards to how projects handle issues reported by users to measure project quality. Meanwhile, [15] focused on how certain open source platforms are better equipped than others at creating a more inclusive space for all genders. These are important factors to consider since increases in gender diversity has been associated with higher levels of productivity [16]. Expertise is another factor that has been shown to positively affect team success [17].

2 Background

2.1 GitHub

Github is a website for hosting public and private projects and is built around the version control system Git. Users can update files through what is called a push, which some have used as a proxy for measuring the amount of work done by an individual [18]. However, it should be noted that a single push can capture multiple edits or changes and so it is not a perfect representation. While GitHub is typically associated with being used primarily by software engineers, many projects are actually not involved with software development [19]. This was further backed up by [1] who found that most of the projects that emerged during the COVID-19 pandemic were focused purely on data aggregation and visualization.

2.2 Survival Analysis

Survival analysis is a common statistical application in biostatistics, and is often used to measure time until event data [20] [21]. In our case we are interested in using it to compare the longevity of COVID themed projects versus non-COVID themed projects. A key feature of survival analysis for our use case is its ability to handle right censored data. This occurs when the event we are interested in occurs either outside the time frame of our study or is unobservable during our study. This is significant for our purposes since all of our data is right censored.

One of the biggest challenges when dealing with open source software from a survival analysis perspective is determining when a project is actually dead. When [22] looked to see how losing core team members could affect the

survival of a project, they ended up surveying developers to see which projects were actually abandoned. To further complicate matters, when [23] looked at project abandonment they determined that many developers end up taking breaks over the course of a projects lifetime. This can make it difficult to establish when a project is truly dead, since, after several months of inactivity a single commit can make that repository 'alive' again. [24] approached this by claiming that any repository that has less than two commits per month is considered inactive, and if a project is inactive for two consecutive months then it is considered abandoned.

[25] developed an algorithm that observed the activity data of a project and fit a polynomial logistic regression to that data to find a death threshold. [26] claimed a project was abandoned if it had no activity at all. They essentially had a hold out data set containing information on the activity of projects after their observable period and determined that a project would be censored if it had activity after the cutoff period. The remaining projects in their study were considered inactive if they had a sudden stop in activity by the end of the observable period. These results were then successfully recreated by [27].

Other research have focused on when individual developers are likely to leave a project [28]. [29] used a threshold of 180 days since a users last commit to determine if a person had left the repository. The choice of their threshold was motivated by [30]. However, given the lack of consensus and other papers using different values they also repeated their study using thresholds of 30 days and 90 days. However, they found that even with the different thresholds they still witnessed the same trends. Meanwhile [31] used a 12 month window in their paper to determine whether a developer had disengaged from a project.

While it is tempting to use an approach similar to [25], they noted that when they applied their algorithm to a sample dataset containing 300 open source projects, a 22 month death threshold was established. This is more than double the length of our entire observable data and so it is unlikely that we can obtain meaningful inferences regarding resurgent rates from our limited data. Further, since there doesn't appear to be a general consensus within the literature, we started by implementing a methodology most similar to that of [24] and [29].

3 Materials and Methods

3.1 Dataset

Our data ranges from January 2020 through September of the same year. By using Google BigQuery and filtering for words like 'covid', 'virus', and 'sars-cov-2' we were able to obtain information on $n = 50,308$ public COVID-themed GitHub repositories along with $n = 20,371,460$ public non-COVID GitHub repositories. The data includes the username of an individual, both the name of the repositories that user contributed to on a monthly basis and a unique

repository id, along with the number of pushes made to each repository by month. [19] showed that most projects are personal, usually only having one committer. To avoid these instances we limited ourselves to only look at projects with at least two members, which reduced our dataset to $n = 14,582$ and $n = 3,111,307$ for the COVID and non-COVID themed teams respectively. For the remainder of the paper we assume the minimum allowable team size is two unless otherwise specified.

3.2 Feature Engineering

[18] found that successful open source teams were significantly correlated with higher levels of diversity and also had smaller effective team sizes. We will summarize the methods from their paper and incorporate their metrics in our analysis. They let M be the number of team members in a project and W the number of pushes, this then acts as a proxy for workload. An information theoretic measure of perplexity was then defined to calculate the effective team size $m = 2^H$ where $H = -\sum_{i=1}^{M} f_i \log_2 f_i$ and $f_i = w_i/W$ with w_i the number of pushes made by team member i. Additionally, by defining R_i to be the set of projects that user i has pushed to the experiential diversity of a project can be measured as $D = \frac{\bigcup_i R_i}{\sum_i R_i}$. The last measure used, which we will also be implementing, is the focus of a team. This is evaluated as the effective team size over the total team size. Therefore, teams that share the workload evenly amongst members will have $m = M$ and hence $m/M = 1$.

3.3 Survival Analysis

3.3.1 Technical Review

A survival curve is defined as $S(t) = Pr(T > t)$, which is to say the probability of observing an alive subject past time t. Kaplan-Meier estimates are a non-parametric method that allow us to approximate the survival curve using the function

$$\hat{S}(d_k) = \prod_{j=1}^{k} \left(\frac{r_j - q_j}{r_j} \right)$$

where $d_1 < d_2 < \cdots < d_k$ represent the K unique death times among non-censored patients, q_k are the number of patients who died at time d_k, and r_k are the number of patients at risk.

Other tools from survival analysis include the hazard function $h(t)$, which is the instantaneous failure rate at time t, given survival past that time. If we let $f(x)$ be the probability density function associated with T then we can show that

$$h(t) = \frac{f(t)}{S(t)}$$

The cumulative hazard function is another useful metric, it's simply the integral of the hazard function from 0 to

t. We can use the cumulative hazard function to draw inferences about the hazard function since the rate of change of the cumulative hazard will give us an estimate of the hazard function. We'll be using the Nelson Aalen estimator, a non-parametric method, from the Lifelines[1] package in python to estimate the cumulative hazard function given by

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{n_i}$$

where d_i is the number of deaths at time t_i and n_i is the number of susceptible individuals.

In addition to the Nelson-Aalen estimator we will also be fitting a Weibull model, which is a parametric model that can be used to estimate both the survival curve and the cumulative hazard. All the survival models fit to our data have been built using the Lifelines python package [32].

3.3.2 Censoring the Data

For our analysis We started by determining the starting month of activity within a project, which is defined as the first instance of a push being recorded. Similarly the end month of a repository was the last month of observed activity. We began by ensuring that there was at least two months of activity in a repository, otherwise the project would be marked as dead. However, we needed to account for the different starting months of these projects. For example, if a project was started during the last month of our observable period it would be default be marked as dead. We correct for this by introducing a cutoff month, such that all repositories that were started during or after this cutoff were marked as censored. The last step was to attempt to account for inactivity. Currently we look to see if the last month of activity is before the cutoff month. If it is, then we mark that project as dead. By defining our cutoff to be two months prior to our last observed month, we can get a similar recreation to the study done by [24], albeit slightly more flexible, as we do allow for projects with stretches of longer than two months of inactivity to remain alive as long as they have activity before the cutoff threshold is reached.

After labelling projects as either dead or censored using the methodology listed above we then calculated Kaplan-Meier curves for both covid and non-covid teams to compare their survival. We did this analysis two times once where we restricted the minimum team size to be at least two people and five people respectively. The choice of restricting teams to be at least five members was motivated by the research done by [29] who noted that "when a group consists of very few items (i.e. contributors in our case), the survival analysis might become inaccurate and does not have much meaning". However, the trends we observed were similar in both instances. Likewise, we also adjusted our cutoff threshold to be both two months before our last recorded month of collection and again with three months. No noticeable change was visible in the trends.

[1] https://lifelines.readthedocs.io/en/latest/index.html

4 Results

4.1 Team Focus

We started by taking a temporal look at the differences between COVID and non-COVID themed projects. As mentioned in §3.2 we use a measure of focus as defined by [18] to assess how team dynamics change on a monthly basis. If we look at each month as independent from all the other months we begin to see that COVID-themed projects start out as more focused in March and April, but then become less focused as the months continue as shown in Fig. 1. Recall that a focus of 1 only happens when $m = M$, which would indicate that the workload is being evenly distributed across all members. However, when we look at these same plots in aggregate of one another, i.e. months are no longer independent of one another, we actually see that the focus for COVID-themed projects is continually decreasing over time (Fig. 2).

These seemingly contradictory results can actually be explained by one person taking on all the responsibility of an entire project. If only one person works on a project in a given month, then by default the focus would be 1. However, in aggregate we are seeing that one person is slowly taking on more and more of the responsibility. This result supports the analysis done by [18], which also found that project workloads are primarily carried out by only a small number of individuals on a team. Plotting the CCDF of the aggregated data in Fig. 3 allows us to see more clearly that COVID-themed teams tend to be more focused than their non-COVID counterparts.

While visibly it seems apparent that there is a difference between the two curves in Fig. 3, we follow up by providing summary statistics in Table 2. From the T-test we can reject the null hypothesis that COVID and non-COVID teams share the same expected value. The Mann-Whitney U rank test also gives strong evidence that the two groups don't come from the same distribution. We used a two-sided version of the Kolmogorov-Smirnov test, which assumes that the two distributions are identical but given our results we can reject that hypothesis as well. The last metric included is Cohen's D to capture the effect size of the difference between the means of the two groups. The value indicates that the two groups differ by around 0.26 standard deviations. We also provide another way of visualizing the focus of teams in 4. From this we can see that COVID themed projects are consistently more focused than their non-COVID counterparts. Note that the large spikes in February's plot are a byproduct of a small number of samples in that period. Addtitional summary statistics and CCDF plots corresponding to the non-aggregated data can be found in Appendix A. **RQ1**: COVID-themed projects are more focused than non-COVID projects.

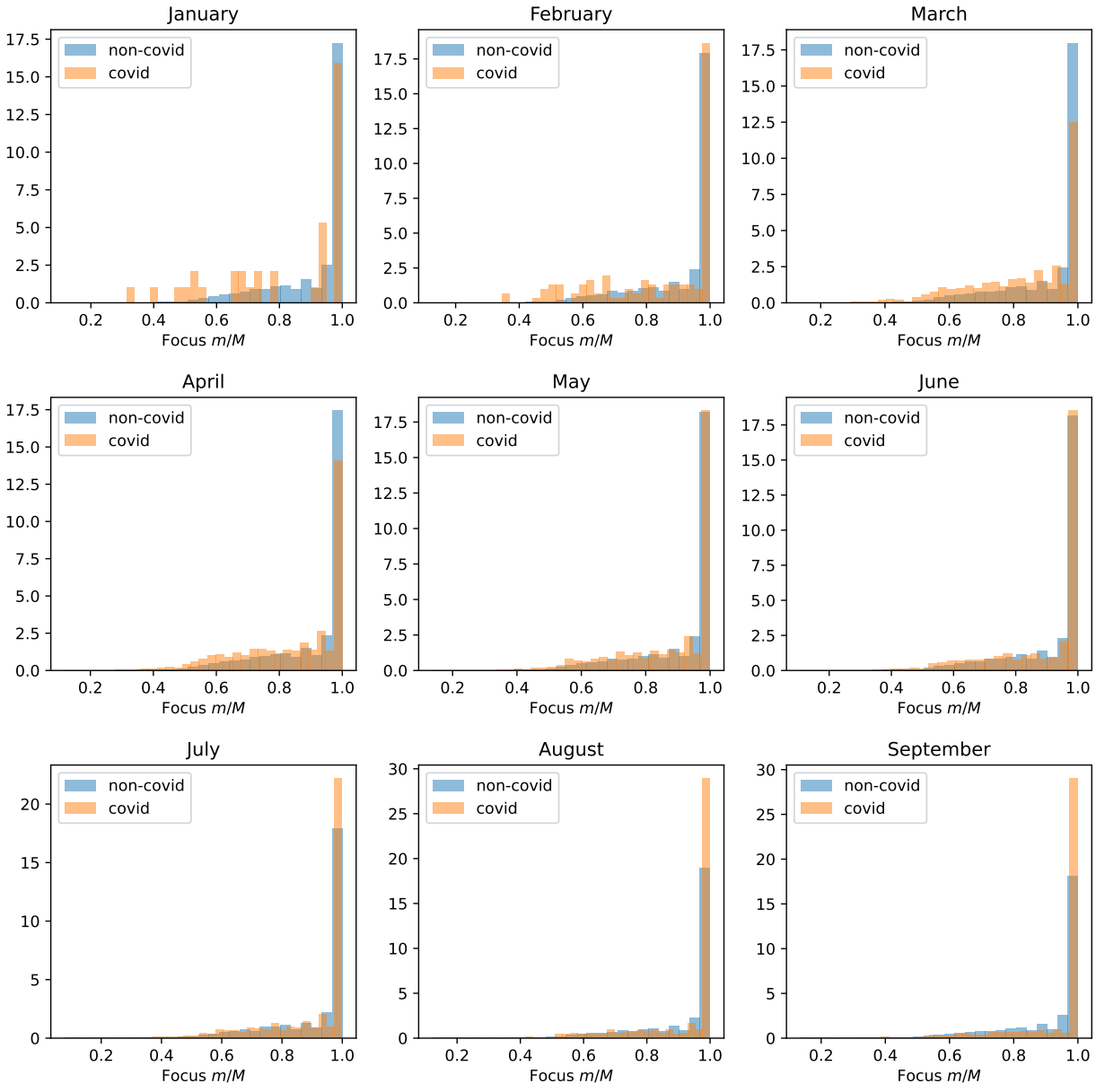

Figure 1: A comparison of team focus between covid and non-covid teams on GitHub by month where each month is independent from the previous months.

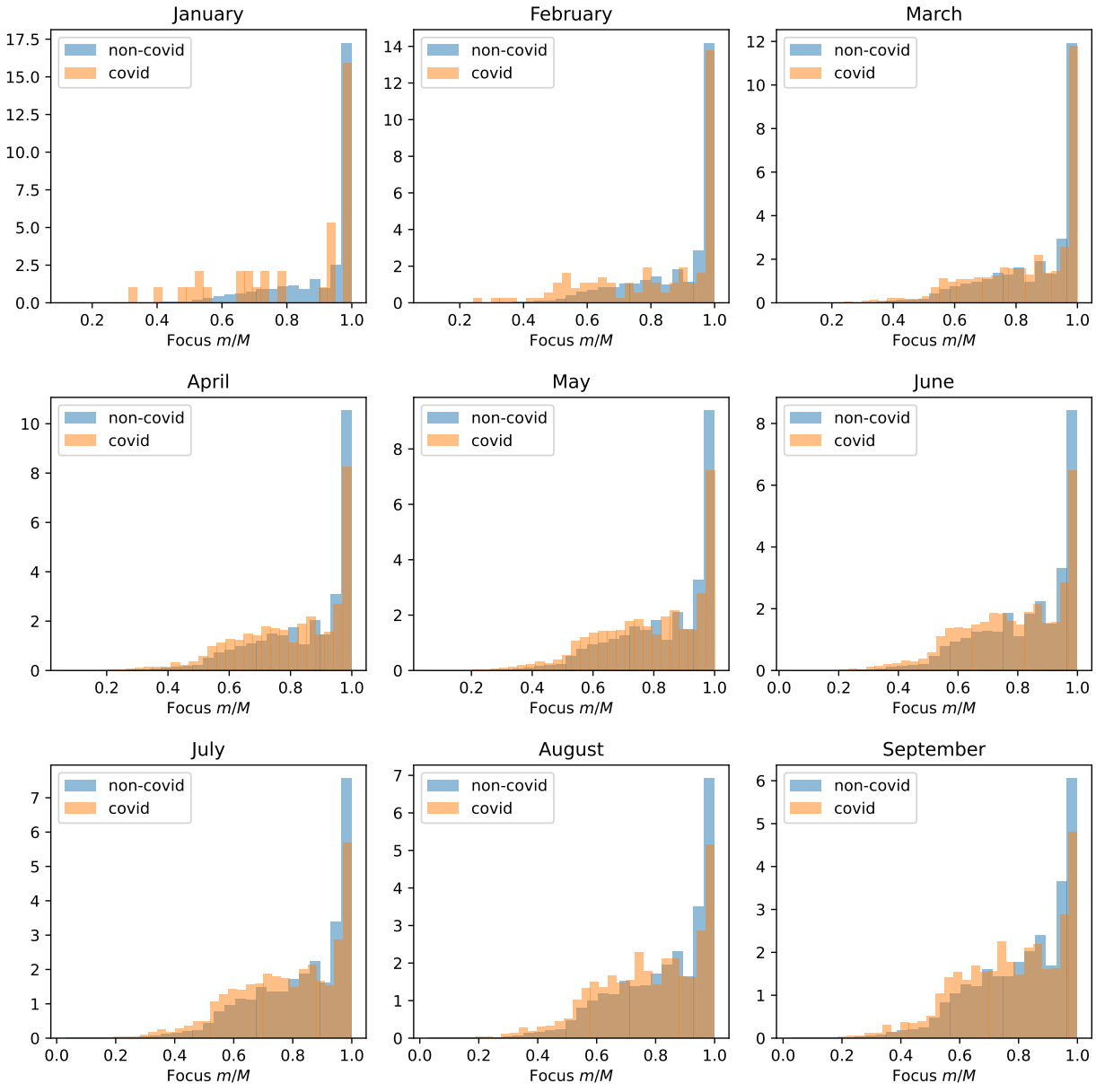

Figure 2: A comparison of team focus between covid and non-covid teams on GitHub by month where each month is dependent on the previous months.

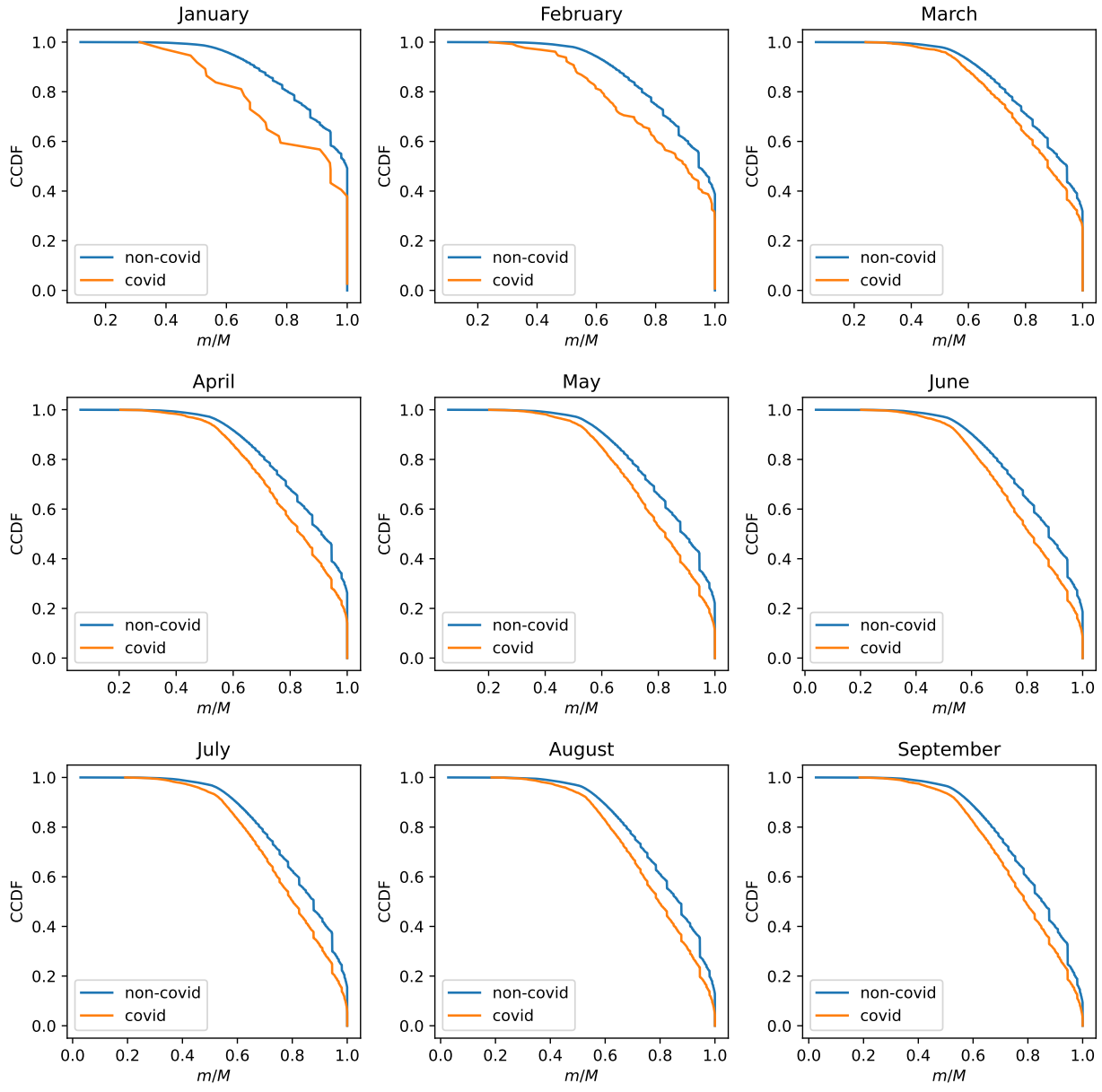

Figure 3: CCDF plots of Fig. 2 which looks at the focus of GitHub teams where previous months history is carried forward into the preceding months. Median values for non-covid were 0.853 ($n = 1,149,380$) and covid 0.79 ($n = 5567$)

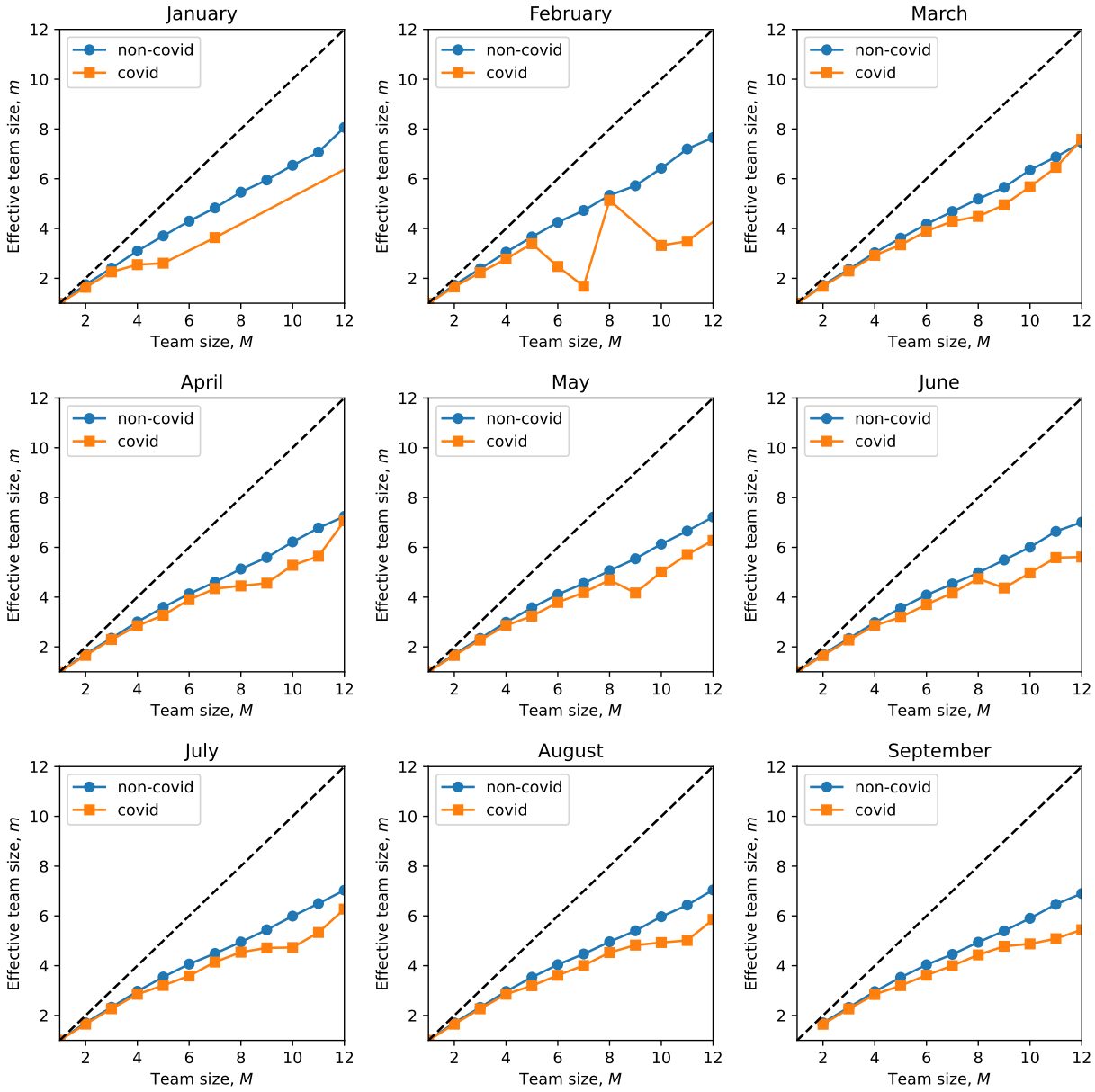

Figure 4: Average effective team size per month for covid and non-covid teams where months are dependent on each other

Statistical Test	statistic	p-value
T-test	19.20126	1.23420e-79
Mann Whitney U	3698538661.5	3.83893e-90
Kolmogorov-Smirnov	0.11556	6.58192e-65
Cohen's D	0.26739	

Table 1: Aggregated monthly data.

Table 2: Statistical tests comparing the differences between COVID and non-COVID themed teams on GitHub for the final month observed (i.e. September) for Fig. 3.

4.2 Team Workload

By using the number of pushes made as a proxy for workload done by an individual we can assess how the two groups compare. However, it should be noted that as with any lines of code based metrics, there are limitation with regards to what pushes actually captures. However, we believe that it is an acceptable metric for the purposes of our study.

By creating a CCDF of the workload done by COVID and non-COVID themed projects we can compare the two groups, both on a month by month basis where in one, each month is treated independent of those that preceded it (Fig. 5) and in the other on an aggregated basis where each month builds off of the the previous months (Fig. 6). Fig. 6 makes it clear to see that COVID themed projects are consistently pushing more, thereby producing more work than non-COVID teams. However, when we look at each month independently as shown in Fig. 5 we see that while the same trend persists early on, they begin to trend towards one another from June onward. This trend is highlighted further if we focus on the median number of pushes made in each month as shown in Fig. 7. This shows us that even though workload slowed in the later part of our observable period for COVID-themed projects, the work done in the earlier months was so great that they still outworked their non-COVID counterparts on the whole. **RQ2**: COVID teams did more work than non-COVID teams.

4.3 Team Diversity

We started by comparing the experiential diversity of teams, as defined in §3.2, before COVID along with an accumulation of before and during the pandemic (Fig. 8). We can note from the plots that COVID teams had higher levels of diversity compared to their non-COVID counterparts both before and during the pandemic.

From this we further explored the differences in diversity from before and after COVID shown in Fig. 9. We can see here that while both experienced some change, there doesn't seem to be a large difference between COVID and non-COVID teams. The median for non-COVID teams is -0.0588 ($n = 906508$) versus -0.0867 ($n = 5095$) for COVID teams. However, it should be noted that this is a statistically significant difference as seen in Table 3. We also explored how the diversity after COVID would compare to a null model with the assumption that there would be no

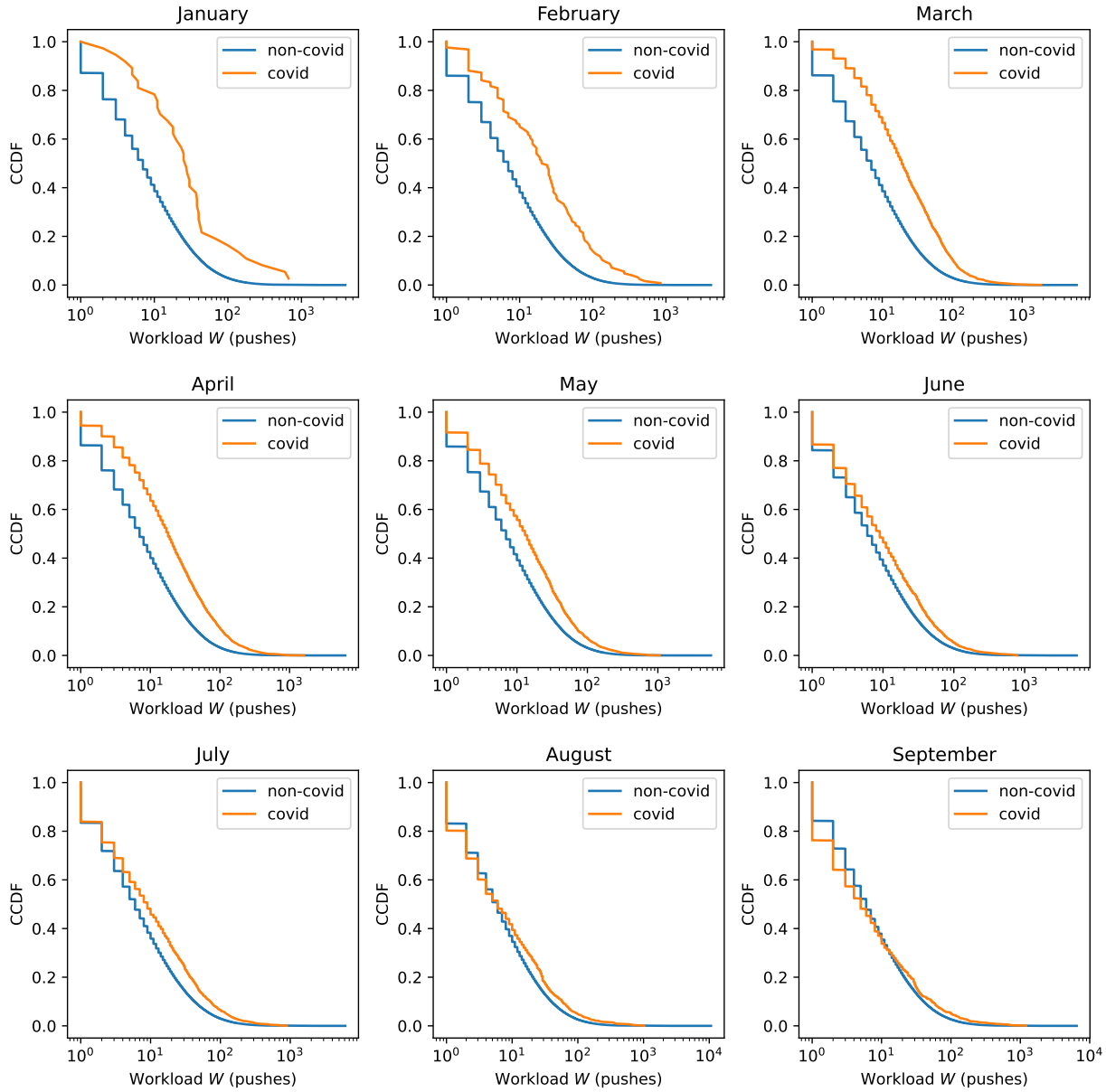

Figure 5: CCDF of workload where months are independent from one another.

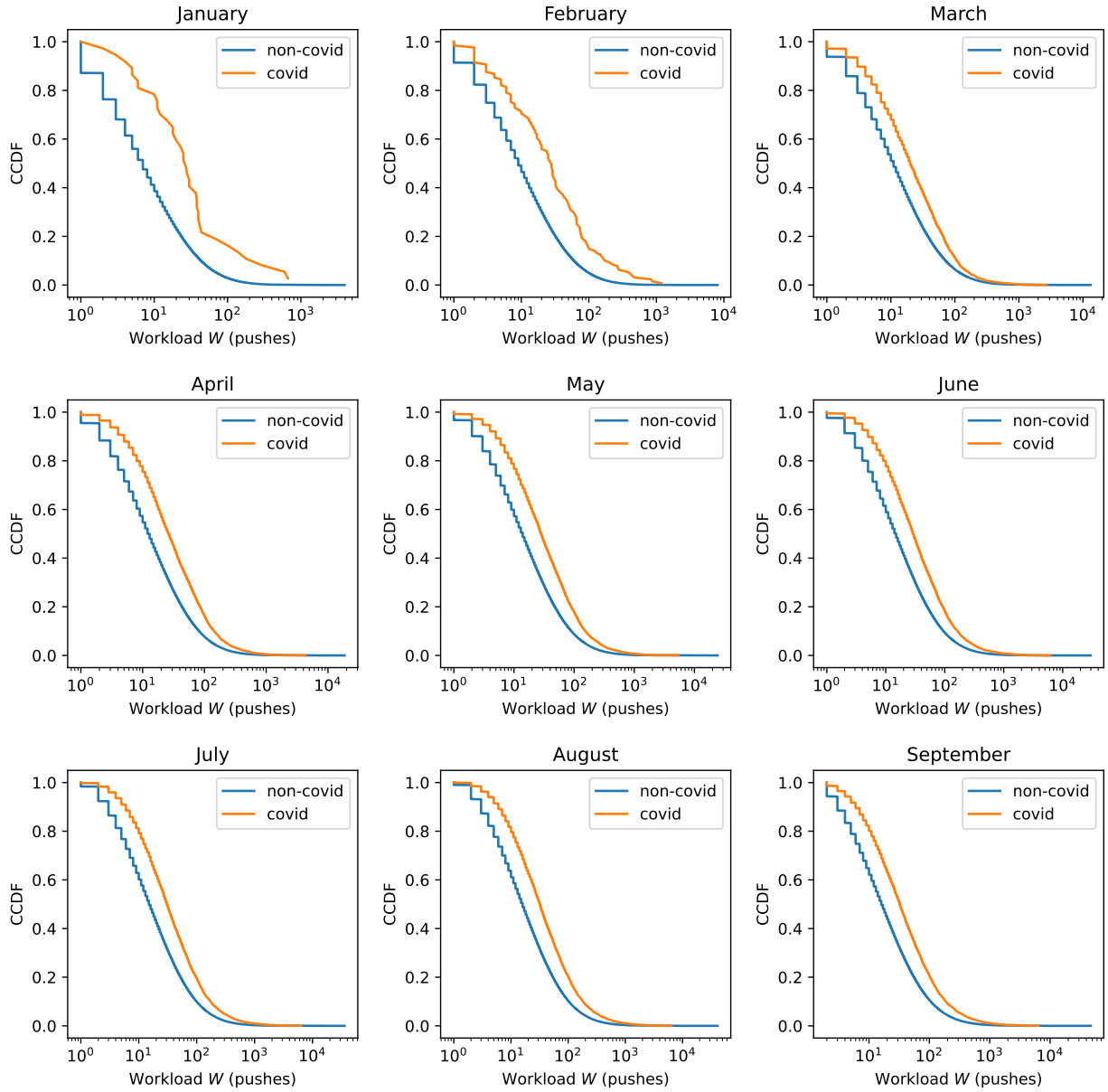

Figure 6: CCDF of workload where months are dependent on one another.

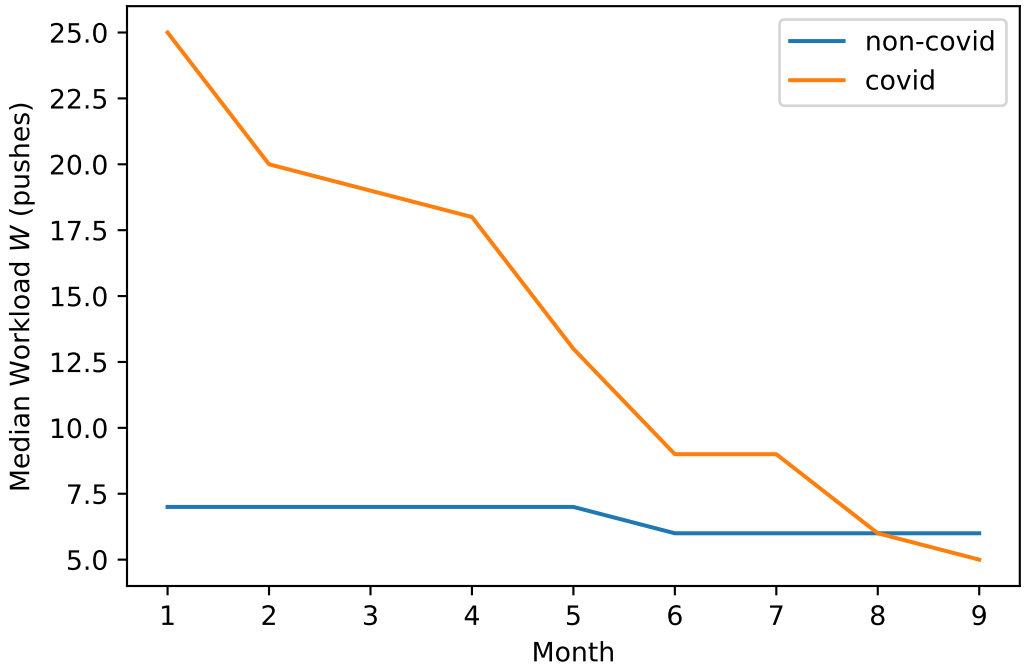

Figure 7: Median work done, as measured in pushes, for COVID and non-COVID teams.

change. The results from this comparison can be seen in Fig. 10 and statistical tests (Table 4) show that there is a significant difference.

If we let Da be the diversity of a team after COVID and $DaNull$ be the expected diversity of a team after COVID under the null hypothesis, then we can calculate excess bonding ($Da < DaNull$), expected bonding ($Da = DaNull$), and excess bridging ($Da > DaNull$) of COVID and non-COVID teams. Fig. 11 depicts this comparison on a monthly basis, starting in January 2020. We can note from this that non-covid teams maintain a relatively constant proportion throughout the entire time frame, with only a slight increase in expected bonding over time. COVID teams on the other hand seem to have a constant decrease in excess bonding (blue), a growth in expected bonding (orange) and an increase in excess bridging (green) until March and then a bit of leveling off until it starts to decrease around July. **RQ3**: COVID teams are more diverse than non-COVID teams.

4.4 Team Survival

As mentioned in the methodology we marked projects as dead if they had less than two months of activity or hadn't been updated by out cutoff month, which was set to be two months prior to our last month of observations. We

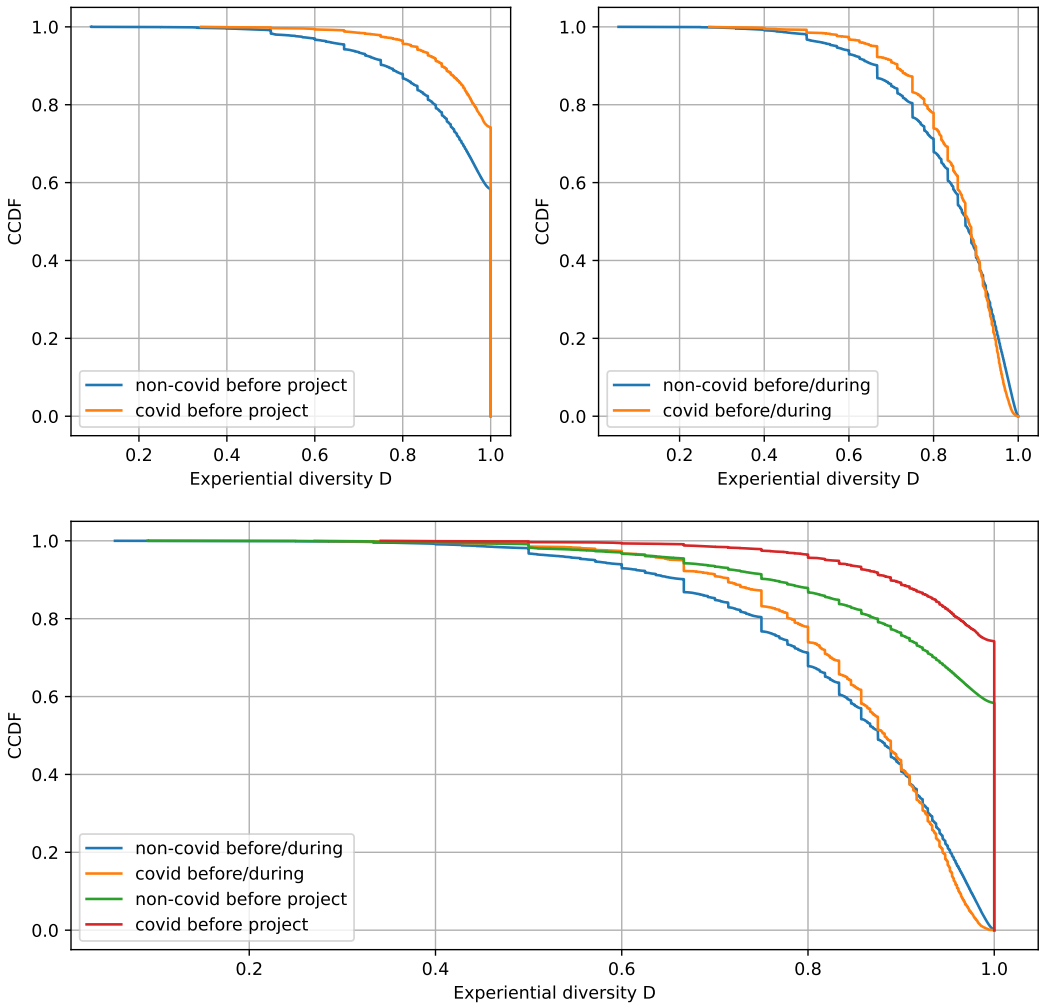

Figure 8: Comparison of experiential diversity of covid and non-covid teams before project (top left), before and during (top right) and full comparison (bottom).

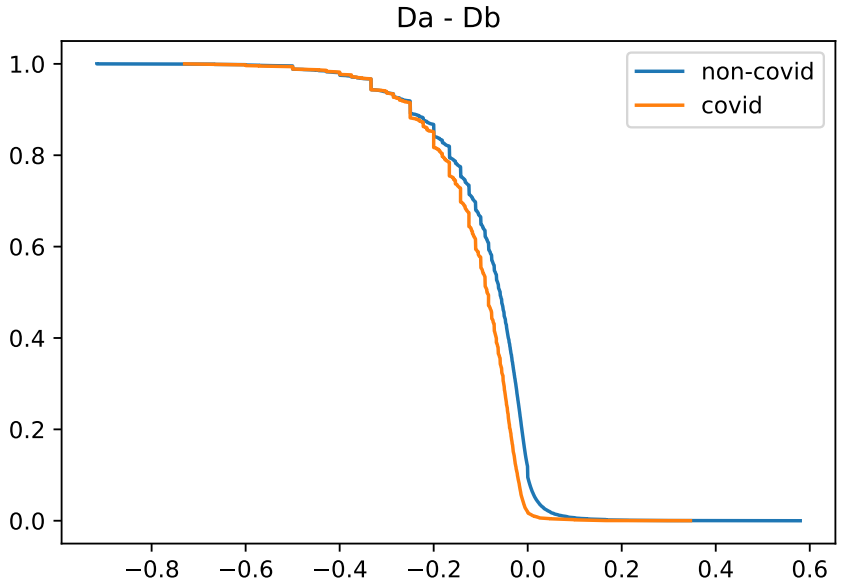

Figure 9: CCDFs comparing the change in diversity before covid and after covid for both non-covid and covid themed teams.

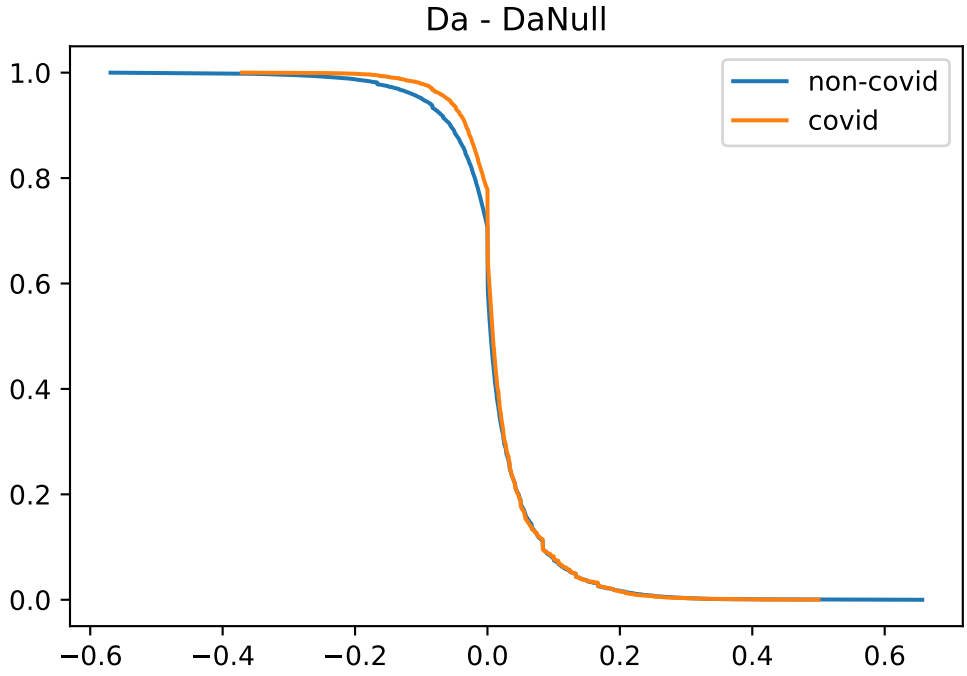

Figure 10: CCDFs comparing the difference of the diversity after covid versus the diversity we'd expect to see after covid under a null model for both covid and non-covid themed teams.

Test	Statistic	P-value
T-test	15.58975	1.40e-53
Mann Whitney U	2759668784.5	9.78e-128
Kolmogorov-Smirnov	0.18423	5.32e-151
Cohens D	0.21010	

Table 3: Statistical tests comparing the change in diversity for covid and non-covid teams of Fig. 9. A statistical significance was observed across all tests.

Test	Statistic	P-value
T-test	-9.75848	2.64e-22
Mann Whitney U	2159156964.5	1.04e-15
Kolmogorov-Smirnov	0.07087	1.41e-22
Cohens D	-0.12320	

Table 4: Statistical tests comparing the change in diversity for teams after covid against the null model Fig. 10. A statistical significance was observed across all tests.

compared the survival of COVID and non-COVID themed projects, once where the allowable minimum team size was two members and again where the minimum allowable team size was five. We found no difference in the trends when we changed the team size however the median survival did increase for both COVID and non-COVID projects going from team sizes of two to five. In this section we only include figures for when the minimum team size is five, however, the other plots can be found in the Appendix.

It is clear from Fig. 12 that non-COVID teams have a higher probability of survival compared to COVID teams. By performing a log-rank test we can show that this difference is in fact significant (85.82, p-value=$1.97e-20$).

The cumulative hazard functions reinforce these findings, indicating that COVID teams had a higher hazard rate than non-COVID teams. **RQ4**: Non-COVID teams have higher levels of survival than COVID teams.

5 Discussion

Throughout this paper we determined that while COVID themed projects tended to have higher levels of focus, diversity, and did more work than non-COVID teams, they tended to be more short lived on average. This brings up questions regarding what types of COVID projects tend to be the most robust to abandonment.

By continuing to restrict ourselves to team sizes with a minimum of five members, we filtered for the top 100 active and abandoned repositories based off of the number of pushes made. Using the taxonomy developed by [1] we manually assessed each repository and labeled them as either contact tracing (c), data collection/visualization (d), detection and diagnosis (dd), forecasting and simulation (f), toolkits (t), or other helpful measures (o). The results from these classifications can be seen in Fig. 13. We noticed from this that most of the active repositories tended to

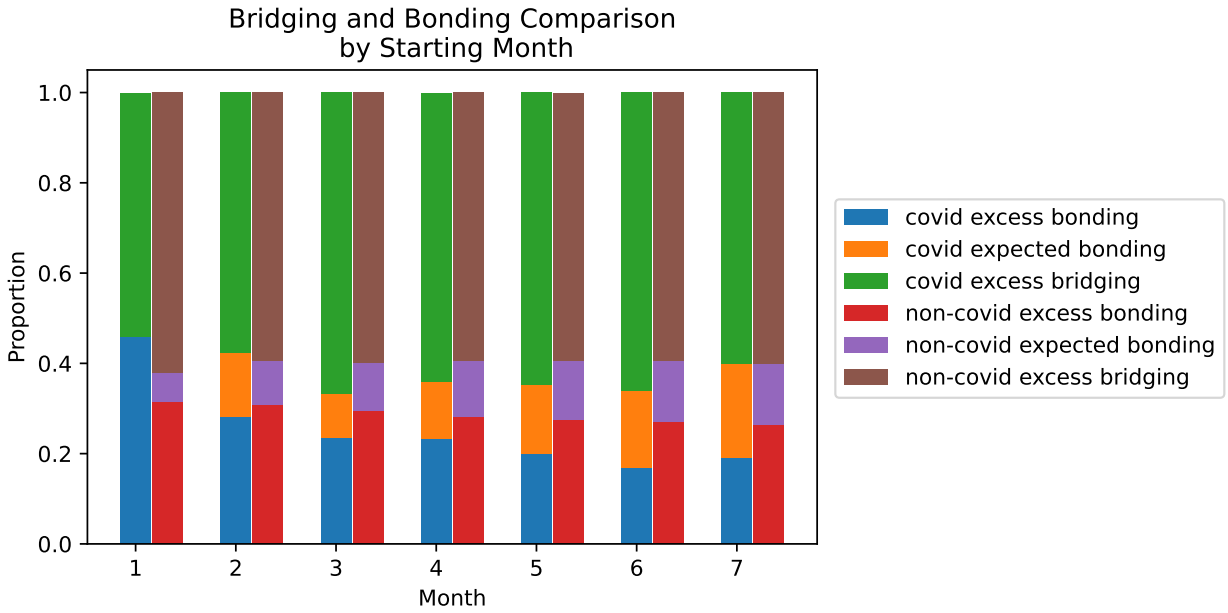

Figure 11: Comparison of the bridging and bonding for covid and non-covid teams.

be focused on data collection and visualization methods, whereas most of the abandoned projects tended to be focused more on other means of providing help.

A possible explanation could be attributed to an almost Darwinian example of survival of the fittest. As we witness thousands of projects attempting to accomplish similar goals, it's the projects that achieve success first that go on to proliferate in the public view. As this happens, developers who were working on similar projects end up conceding to their faster developing counterparts. However, further research would need to be done to verify these claims.

Future research may wish to dive deeper into the questions of survival and longevity within open source teams during times of duress. It has already been shown that including a contributing.md file has correlated with a 25% to 45% increase in survival [33]. Hence we believe that looking into the metadata of these repositories could provide useful insights into the factor that lead to longer lived projects.

6 Threats to Validity

There are many concerns to be had when using GitHub as a datasource [19]. The first concern is the use of pushes as a metric for workload. While this is common practice within the literature, it's still worth noting that any 'lines of code' or similar metric will be imperfect at capturing the true workload being done. Especially when we factor in that most of the COVID-themed repositories tend to revolve around data collection. This could skew our results if there is

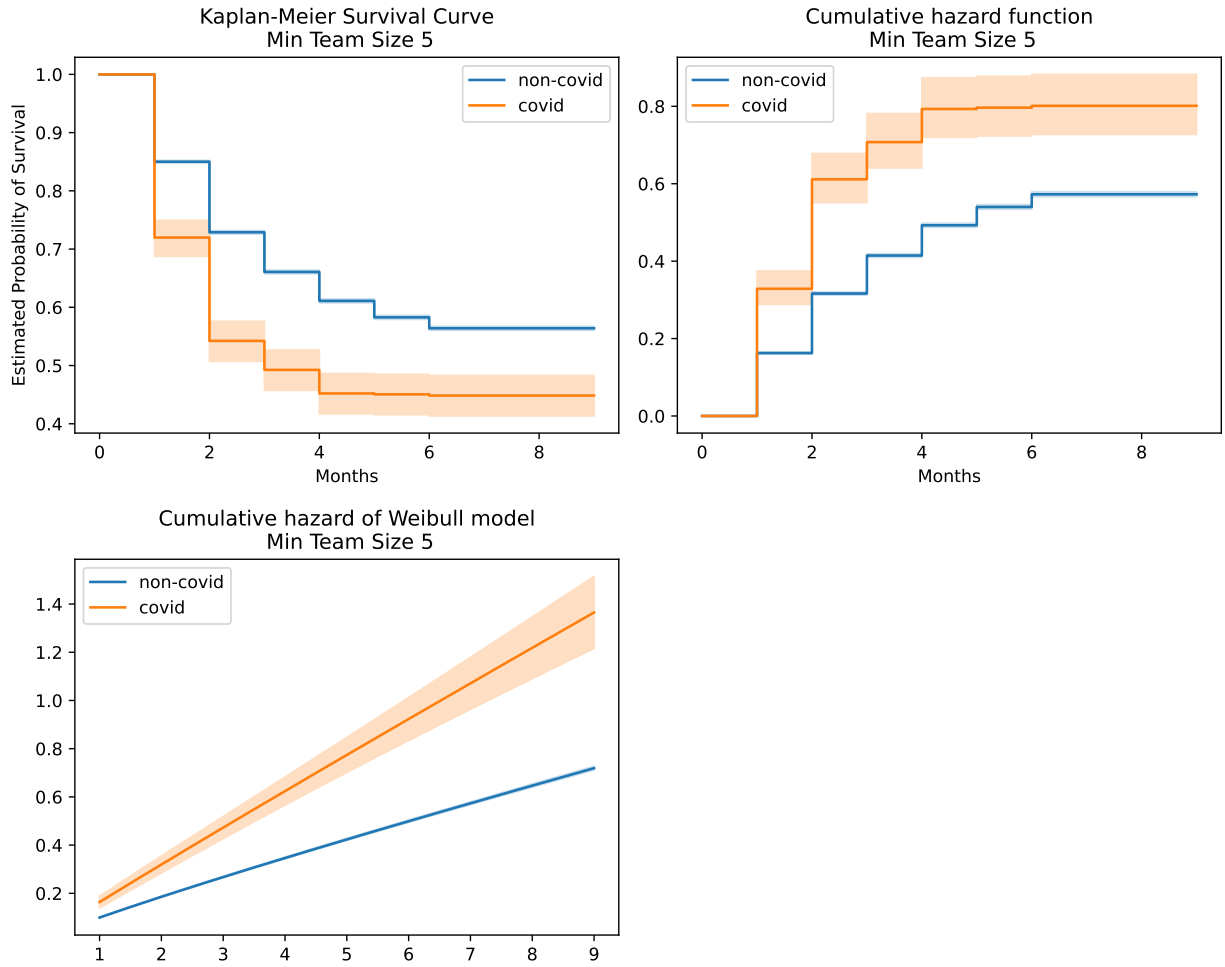

Figure 12: Comparison of covid and non-covid teams with minimum team size of five with 95% confidence intervals displayed. (top left) survival curves, (top right) cumulative hazard using non-parametric Nelson-Aalen, and (bottom) cumulative hazard using parametrc Weibull model.

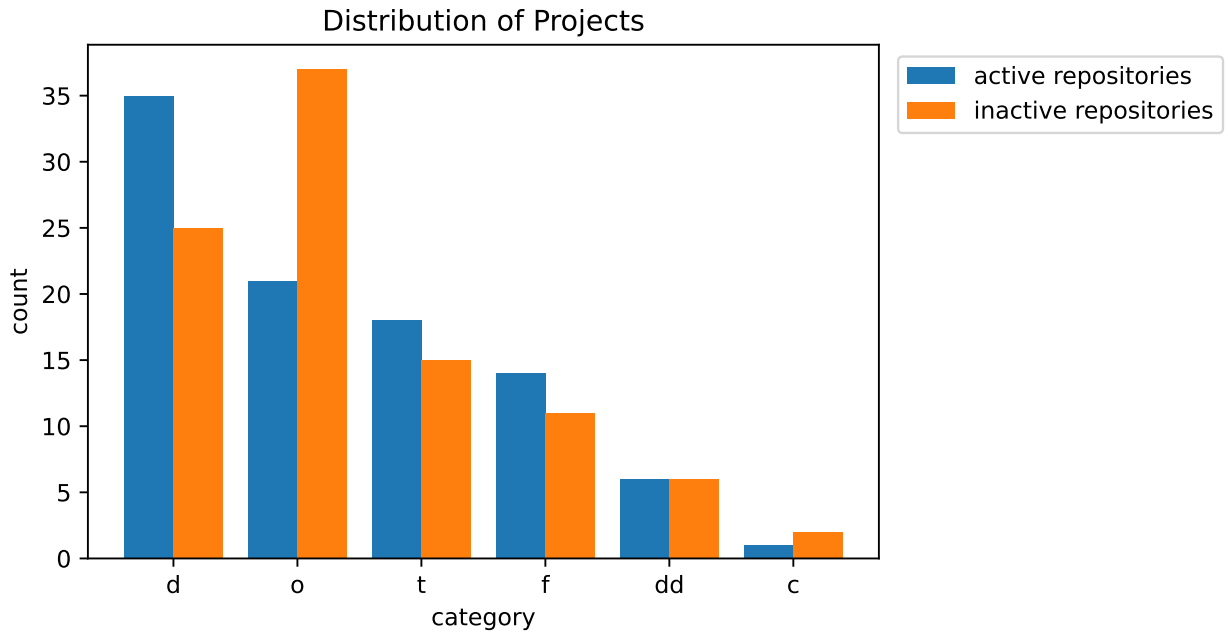

Figure 13: Active COVID-themed repositories were more focused on data collection methods while inactive projects were more focused on alternative methods of helping.

an automatic script scraping the web and updating case statistics within a repository on a daily basis.

Second is the data collection method. We used keywords to filter for repositories and projects related to COVID-19, however, in our analysis of the data we found that there were some false positives. For example, there was one repository hosted by a student who, in their README.md file explained that they were hosting their work online because of COVID-19, even though the project itself was completely unrelated. However, when we manually looked at 100 of the COVID-themed repositories, this was the only example of such an event occurring. While it might therefore be encouraging to suggest that this repository in particular is an outlier, it is undoubtedly not the only false positive present in our dataset and therefore, caution should be exercised.

Third is the network structure of code repositories. As outlined by [19], a repository is not necessarily a project and can be part of a network of repositories where one acts as a central node. We found this to be true in our case as well. There were several repositories, like alphagov, which provided digital services like vulnerable people forms for citizens of the UK. Alternatively, there were also web applications we found that had multiple repositories for front end and back end development. Unfortunately it is difficult to group these projects together with an algorithm based on the name of a project because there tends to be too much overlap in the naming conventions used. A possible approach might be to parse through the README.md files to better understand the underlying network structure, however, we did not have the time or resources access to that information in our dataset.

Lastly, the use of survival analysis with regards to open source software remains a complicated issue. Given that any project with a long period of inactivity can become 'alive' with a single push at some point in the future, it can be difficult to properly assess whether a project is truly alive or abandoned. We did our best to faithfully represent the survival process chosen for our analysis but we believe that this is an open question within the open source community worth pursuing further. Other possible approaches might be to include a discount rate such that activity towards the end of an observable period are weighted more heavily than those further back.

7 Conclusion

In this paper we looked at how open source teams on GitHub responded in times of duress. We showed that during these times they maintained higher levels of focus, workload, and diversity indicating the power that exists within the open source community for banding together. We also showed that despite these qualities these projects also tended to have higher rates of burnout and abandonment.

Acknowledgments

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

References

[1] L. Wang, R. Li, J. Zhu, G. Bai, and H. Wang, "When the open source community meets covid-19: Characterizing covid-19 themed github repositories," *arXiv preprint arXiv:2010.12218*, 2020. 1, 2, 17

[2] T. Chełkowski, D. Jemielniak, and K. Macikowski, "Free and open source software organizations: A large-scale analysis of code, comments, and commits frequency," *PloS one*, vol. 16, no. 9, p. e0257192, 2021. 1

[3] A. Mockus, R. T. Fielding, and J. D. Herbsleb, "Two case studies of open source software development: Apache and mozilla," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 11, no. 3, pp. 309–346, 2002. 1

[4] G. K. Lee and R. E. Cole, "From a firm-based to a community-based model of knowledge creation: The case of the linux kernel development," *Organization science*, vol. 14, no. 6, pp. 633–649, 2003. 1

[5] J. A. Roberts, I.-H. Hann, and S. A. Slaughter, "Understanding the motivations, participation, and performance of open source software developers: A longitudinal study of the apache projects," *Management science*, vol. 52, no. 7, pp. 984–999, 2006. 1

[6] L. Bao, X. Xia, D. Lo, and G. C. Murphy, "A large scale study of long-time contributor prediction for github projects," *IEEE Transactions on Software Engineering*, vol. 47, no. 6, pp. 1277–1298, 2019. 1

[7] K. Crowston and J. Howison, "The social structure of free and open source software development," *First Monday*, 2005. 1

[8] Y. Long and K. Siau, "Social network structures in open source software development teams," *Journal of Database Management (JDM)*, vol. 18, no. 2, pp. 25–40, 2007. 1

[9] C. Bird, D. Pattison, R. D'Souza, V. Filkov, and P. Devanbu, "Latent social structure in open source projects," in *Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering*, pp. 24–35, 2008. 1

[10] M. J. Palazzi, J. Cabot, J. L. Canovas Izquierdo, A. Solé-Ribalta, and J. Borge-Holthoefer, "Online division of labour: emergent structures in open source software," *Scientific Reports*, vol. 9, no. 1, pp. 1–11, 2019. 1

[11] H. Borges, M. T. Valente, A. Hora, and J. Coelho, "On the popularity of github applications: A preliminary note," *arXiv preprint arXiv:1507.00604*, 2015. 2

[12] H. Borges, A. Hora, and M. T. Valente, "Understanding the factors that impact the popularity of github repositories," in *2016 IEEE international conference on software maintenance and evolution (ICSME)*, pp. 334–344, IEEE, 2016. 2

[13] M. Goeminne and T. Mens, "Towards a survival analysis of database framework usage in java projects," in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pp. 551–555, IEEE, 2015. 2

[14] O. Jarczyk, B. Gruszka, S. Jaroszewicz, L. Bukowski, and A. Wierzbicki, "Github projects. quality analysis of open-source software," in *International Conference on Social Informatics*, pp. 80–94, Springer, 2014. 2

[15] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, representation and online participation: A quantitative study," *Interacting with Computers*, vol. 26, no. 5, pp. 488–511, 2014. 2

[16] B. Vasilescu, D. Posnett, B. Ray, M. G. van den Brand, A. Serebrenik, P. Devanbu, and V. Filkov, "Gender and tenure diversity in github teams," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, pp. 3789–3798, 2015. 2

[17] S. Faraj and L. Sproull, "Coordinating expertise in software development teams," *Management science*, vol. 46, no. 12, pp. 1554–1568, 2000. 2

[18] M. Klug and J. P. Bagrow, "Understanding the group dynamics and success of teams," *Royal Society open science*, vol. 3, no. 4, p. 160007, 2016. 2, 4, 6

[19] E. Kalliamvakou, G. Gousios, K. Blincoe, L. Singer, D. M. German, and D. Damian, "An in-depth study of the promises and perils of mining github," *Empirical Software Engineering*, vol. 21, no. 5, pp. 2035–2071, 2016. 2, 4, 18, 20

[20] G. James, D. Witten, T. Hastie, and R. Tibshirani, "Statistical learning," in *An introduction to statistical learning*, pp. 461–469, Springer, 2021. 2

[21] D. G. Kleinbaum and M. Klein, "Survival analysis," in *Survival Analysis a Self Learning Text*, Springer, 2012. 2

[22] G. Avelino, E. Constantinou, M. T. Valente, and A. Serebrenik, "On the abandonment and survival of open source projects: An empirical investigation," in *2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pp. 1–12, IEEE, 2019. 2

[23] F. Calefato, M. A. Gerosa, G. Iaffaldano, F. Lanubile, and I. Steinmacher, "Will you come back to contribute? investigating the inactivity of oss core developers in github," *Empirical Software Engineering*, vol. 27, no. 3, pp. 1–41, 2022. 3

[24] I. Samoladas, L. Angelis, and I. Stamelos, "Survival analysis on the duration of open source projects," *Information and Software Technology*, vol. 52, no. 9, pp. 902–922, 2010. 3, 5

[25] N. Evangelopoulos, A. Sidorova, S. Fotopoulos, and I. Chengalur-Smith, "Determining process death based on censored activity data," *Communications in Statistics—Simulation and Computation®*, vol. 37, no. 8, pp. 1647–1662, 2008. 3

[26] R. H. Ali, C. Parlett-Pelleriti, and E. Linstead, "Cheating death: A statistical survival analysis of publicly available python projects," in *Proceedings of the 17th International Conference on Mining Software Repositories*, pp. 6–10, 2020. 3

[27] D. Robinson, K. Enns, N. Koulecar, and M. Sihag, "Two approaches to survival analysis of open source python projects," *arXiv preprint arXiv:2203.08320*, 2022. 3

[28] F. Ortega and D. Izquierdo-Cortazar, "Survival analysis in open development projects," in *2009 ICSE Workshop on Emerging Trends in Free/Libre/Open Source Software Research and Development*, pp. 7–12, IEEE, 2009. 3

[29] B. Lin, G. Robles, and A. Serebrenik, "Developer turnover in global, industrial open source projects: Insights from applying survival analysis," in *2017 IEEE 12th International Conference on Global Software Engineering (ICGSE)*, pp. 66–75, IEEE, 2017. 3, 5

[30] M. Foucault, M. Palyart, X. Blanc, G. C. Murphy, and J.-R. Falleri, "Impact of developer turnover on quality in open-source software," in *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*, pp. 829–841, 2015. 3

[31] H. S. Qiu, A. Nolte, A. Brown, A. Serebrenik, and B. Vasilescu, "Going farther together: The impact of social capital on sustained participation in open source," in *2019 ieee/acm 41st international conference on software engineering (icse)*, pp. 688–699, IEEE, 2019. 3

[32] C. Davidson-Pilon, "lifelines: survival analysis in python," *Journal of Open Source Software*, vol. 4, no. 40, p. 1317, 2019. 5

[33] V. Cosentino, J. L. C. Izquierdo, and J. Cabot, "A systematic mapping study of software development with github," *IEEE Access*, vol. 5, pp. 7173–7192, 2017. 18