3-2020

How does status affect performance and learning from failure? Evidence from online communities

Tengjian ZOU
Singapore Management University, tengjianzou.2015@pbs.smu.edu.sg

HOW DOES STATUS AFFECT PERFORMANCE AND LEARNING FROM FAILURE? EVIDENCE FROM ONLINE COMMUNITIES

TENGJIAN ZOU

SINGAPORE MANAGEMENT UNIVERSITY
2020

How Does Status Affect Performance and Learning from Failure?
Evidence from Online Communities

Tengjian Zou

Submitted to Lee Kong Chian School of Business

in partial fulfilment of the requirements for the

Degree of Doctor of Philosophy in Business (Strategic Management & Organisation)

Dissertation Committee:

Gokhan Ertug (Supervisor / Chair)

Professor of Strategic Management

Singapore Management University

Ilya Cuypers

Associate Professor of Strategic Management

Singapore Management University

Reddi Kotha

Associate Professor of Strategic Management

Singapore Management University

Gianluca Carnabuci

Associate Professor of Organizational Behavior

ESMT Berlin

Singapore Management University

2020

I hereby declare that this PhD dissertation is my original work
and it has been written by me in its entirety.
I have duly acknowledged all the sources of information
which have been used in this dissertation.

This PhD dissertation has also not been submitted for any degree
in any university previously.

邹腾剑

Tengjian Zou

31 March 2020

How Does Status Affect Performance and Learning from Failure?
Evidence from Online Communities

Tengjian Zou

Abstract

This dissertation is composed of two essays. In the first essay, I investigate the factors that can alleviate the detrimental effect of hierarchy on team performance. I first show that hierarchy negatively impacts team performance, which is consistent with recent meta-analytic evidence. One mechanism that drives this negative effect is that hierarchy prevents low-ranking members from voicing their potentially valuable insights. Then I propose that team familiarity is one factor that can encourage low-ranking team members to speak up. I contend that team familiarity can be established either by team members' prior experience in working with one another or can be built by team members' prior experience in working in hierarchical teams, such that they are familiar with hierarchical working relationships. Using data collected from an online crowdsourcing contest community, I find that team members' familiarity with each other and their familiarity with hierarchical working relationships can alleviate the detrimental effect of hierarchy on team performance. By illuminating the moderating effect of team familiarity on the hierarchy-performance relationship, this study advances current understandings of how to reduce the detrimental effect of hierarchy on performance, and offers insights about how teams should be organized to improve performance.

In the second essay, I examine what factors drive learning from failure. In answering this question, I bring status theory into the literature on learning from failure and propose that status can drive people's learning from their failures. I propose that failure feedback given by a higher-status source is more likely to drive a focal individual to learn from her failures than failure feedback given by a lower-status source. This is because people pay more attention to and are more engaged with failure feedback given by a higher-status source than failure feedback given by a lower-status source. Data collected from an online programming contest community provides support to my prediction that failure feedback given by higher-status peers has stronger effect in driving learning from failure than failure feedback given by lower-status peers. By demonstrating that status is a driver of learning from failure, I expand experiential learning theories by incorporating status theory.

Table of Contents

Acknowledgement

I would like to express the deepest appreciation to my supervisor and committee chair Professor Gokhan Ertug. His friendly guidance and expert advice have been invaluable throughout all stages of this work. Without his guidance and persistent help this dissertation would not have been possible. It has been a great pleasure and honor to have him as my supervisor.

I would like to thank my committee members, Professor Ilya Cuypers, Professor Reddi Kotha, and Professor Gianluca Carnabuci, for their invaluable advice and feedback on this research.

I am deeply thankful to my family for their love, support, and sacrifices. Without them, this dissertation would never have been written. This last word of acknowledgment I have saved for my dear wife Han Shi, who has been with me all these years and has made them the best years of my life.

1. Introduction for Both Essays

Status is a pervasive social phenomenon that leads to both positive and negative outcomes. Both essays of this dissertation are related to the consequences of status. Whereas the first essay of this dissertation investigates the negative outcome of status, the second essay examines the positive side of status. According to status theory, high-status individuals command deference from low-status individuals. I contend that this has negative effect as well as positive effect. One the one hand, because low-status individuals defer to high-status individuals, low-status individuals are less likely to speak up and contribute their views when they work with high-status individuals. This is harmful if a team is composed of both high-status members and low-status members, because low-status members are less likely to speak up. Such lack of knowledge sharing from low-status members could reduce the novel ideas that are generated by the team, which detracts from team performance. On the other hand, because high-status individuals receive deference from low-status individuals, this could benefit low-status individuals' learning, especially when learning from their own failures. Literature on learning from failure suggests that failures are usually unpleasant experiences, which people tend to react defensively against, such that they might not pay attention to these failures. This undermines learning because people cannot learn from failures that they do not pay attention to. However, if the failure feedback is given by high-status peers, then according to status theory, the focal individuals are more likely to pay attention and be more engaged with the failures, which makes them more likely to learn from failure feedback given by higher-status peers than failure feedback given by lower-status peers. Therefore, from the perspective of learning, feedback from high-status others can drive people to learn from their failures.

Nowadays, people are increasingly using online communities (i.e., virtual organizations) to interact with others and to conduct certain activities. More and more online communities are implementing status hierarchy systems to differentiate their members. Therefore, it is important to understand the pros and cons of hierarchy systems in online communities. The data of both my essays are collected from online communities. The data for my first essay is collected from Kaggle, which is an online community for crowdsourcing contests. The data for my second essay is collected from an online community for programming contests. The findings in my essays offer practical implications for online communities regarding the positive side and negative side of status hierarchy systems. Specifically, whereas status hierarchy systems could hinder team performance, this system could also promote individuals' learning from their failures.

I contend in my first essay that status hierarchical differentiation leads low-status team members to be less likely to voice their possibly valuable insights, which can undermine team performance. This result is consistent with the recent meta-analytic evidence. In light of the negative effect of status hierarchical differentiation on team performance, I then investigate the factors that can alleviate the detrimental effect of hierarchy on team performance. I propose that team familiarity that emerges from team members' past shared experience and team members' familiarity with hierarchical relationships can reduce the negative relationship between hierarchy and team performance. By illuminating the moderating effect of team familiarity on the hierarchy-performance relationship, this study shows how to reduce the detrimental effect of hierarchy on performance, and offers insights about how teams should be organized to improve performance.

In the second essay, I show the positive effect of status. Specifically, I bring status theory into the literature on learning from failure and propose that status can drive people's learning from their failures. I propose that failure feedback given by higher-status peers is more likely to drive a focal individual to learn from her failures than failure feedback given by lower-status peers. This is because people pay more attention to and are more engaged with failure feedback given by higher-status peers than to failure feedback given by lower-status peers. By demonstrating that status is a driver of learning from failure, I expand experiential learning theories by incorporating status theory.

In summary, my dissertation advances the research of status at the team level as well as the individual level. At the team level, my results suggest that an overall detrimental effect of hierarchy on team performance is not unavoidable. Teams with more familiarity among their members, either with each other or with hierarchical working relationships, suffer less from the negative effect of hierarchy on team performance. At individual level, my results reveal that even though learning from failure is not straightforward for individuals, the status of feedback providers can drive them to learn from failures.

2. Hierarchy, Team Familiarity, and Performance in Online Crowdsourcing Teams

2.1. Introduction

Hierarchy is ubiquitous in social groups (Magee & Galinsky, 2008). In light of this, it is concerning that a recent meta-analysis provides aggregated evidence that, on net, hierarchy negatively impacts team performance (Greer et al., 2018). One mechanism that drives this negative effect is that hierarchy prevents low-status members from voicing their potentially valuable insights (Anicich, Swaab, & Galinsky, 2015; Keum & See, 2017). Specifically, hierarchical differentiation leads higher-status members to expect deference from lower-status members, which makes it difficult for lower-status members to disagree and express their views. In consideration of the detrimental effect of hierarchy on performance, it is important to understand what factors can encourage low-status members to voice their insights and thus alleviate the detrimental effect of hierarchy on team performance.

I propose that one factor that can facilitate information exchange between high-status and low-status team members is team familiarity. Team familiarity could emerge in two ways. One way that team familiarity can be established is by team members' prior experience in working with one another, such that they know which individual has what knowledge within the team (Huckman & Staats, 2011). This understanding of who knows what within the team can encourage low-status members to speak up for two reasons. First, if low-status members have no idea of what other team members know, they are more likely to be fearful that their propositions will be dismissed by other team members, so that these low-status members end up being less likely to speak up.

In comparison, if they know what other team members know, low-status members are more likely to speak up because they can be more confident that their perspective will be valuable to the team. Second, if high-status members are not aware of what low-status members know, they are less likely to give low-status members chance to speak up, because they might consider low-status members as less knowledgeable and less likely to have something valuable to share. By contrast, when they know what low-status members know, high-status members are more likely to encourage low-status members to speak up and contribute their ideas when necessary.

A second way that team familiarity can emerge is from team members' prior experience in working in hierarchical teams, such that they are familiar with hierarchical working relationships. This familiarity with hierarchical working relationship can promote voice from low-status members for two reasons. First, if low-status members have worked in hierarchical teams before, they are more likely to understand how to voice their insights in ways that are more likely to be considered by other team members. In comparison, without such familiarity low-status members would have more uncertainty about how, when, or whether their views will be considered by others. This uncertainty would lead low-status members to make safer choices in voicing their opinion or less likely to speak up. Second, if high-status members have worked in hierarchical teams before, they are more likely to be aware that low-status members might keep silent or voice safe opinions only because they are afraid of making mistakes or being dismissed. This understanding is valuable because high-status members can accordingly find ways to reduce low-status members' concern such that they can voice their insights.

Overall, I propose that team members' familiarity with each other and their familiarity with hierarchical working relationships can drive low-status members to voice their insights more, which can alleviate the detrimental effect of hierarchy. I test my theory by collecting data from an online crowdsourcing contest community that relies on a formal hierarchy system. The large number of teams I can trace, the objective performance metric available, and the different degrees of team familiarity that results from their past interactions with one another, make this setting well-suited to test my predictions. By illuminating the moderating effect of team familiarity on the hierarchy-performance relationship, this study advances current understandings of how to soothe the detrimental effect of hierarchy, and offers insights into how teams should be organized to improve team performance.

2.2. Literature Review

Hierarchy refers to a rank ordering of individuals with respect to a valued social dimension (Magee & Galinsky 2008). Extant literature offers two divergent views on the relationship between hierarchy and team performance (Anderson & Willer, 2014; Greer et al., 2018). On the one hand, some scholars posit that hierarchy is functional[1] because it provides incentives for team members to work hard and facilitates coordination and decision making within the team (Magee & Galinsky, 2008). For example, Halevy and colleagues (2012) found that hierarchical differentiation enhanced team performance in professional basketball teams. In a study of 75 teams drawn from a range of industries, Bunderson and colleagues (2016) found that more hierarchically

[1] In line with the explicit or implicit use in the literature on hierarchy and team performance (e.g., Anderson & Brown, 2010), I use "functional" when talking about a positive relationship between hierarchy and team performance, and "dysfunctional" in reference to a negative relationship between hierarchy and team performance.

differentiated teams enjoyed better team performance and greater member satisfaction. In three experimental studies, De Kwaadsteniet and Van Dijk (2010) demonstrated that in hierarchical groups, low-ranked individuals are inclined to defer to the preferences of high-ranked individuals, thereby facilitating coordination. Based on their findings from a study of 72 work teams across diverse business settings, Cantimur, Rink, and van der Vegt (2016) showed that hierarchy increases team performance in teams working on tasks with low complexity. Similarly, Simpson, Willer, and Ridgeway (2012) proposed that hierarchies help organize collective actions within teams by coordinating how much and when individual members should contribute to group efforts.

In contrast with this view, the dysfunctional view of hierarchy suggests a negative relationship between hierarchical differentiation and team performance. According to this perspective, introducing formal hierarchical differentiation within a team is detrimental because hierarchical markers come with an expectation that high-status members should command influence and that lower-status members defer to them. This may hinder team performance by preventing lower-ranked members from voicing their potentially valuable perspectives and insights (Anicich et al., 2015). For example, using a sample of 56 organizational work teams from a range of industries, Oedzes and colleagues (2019) found that more hierarchically differentiated teams displayed lower team creativity. Studies in the setting of professional sports also found cases where hierarchical differentiation may impair team performance. For example, Coates, Frick, and Jewell (2016) showed that hierarchy in Major League Soccer teams is negatively related to team production. Similarly, using U.S. Major League

Baseball data, Jewell and Molina (2004) demonstrated that teams organized more hierarchically were less successful.

A recent meta-analysis consolidates the empirical evidence the literature and suggests that, in the net, hierarchy negatively impacts team performance (Greer et al., 2018), which provides support to the dysfunctional view of hierarchy. In consideration of this detrimental effect of hierarchy, it is important to understand what factors can encourage low-status members to voice their insights and thus alleviate the detrimental effect of hierarchy on team performance. In the next section, I first theorize why hierarchy is detrimental for team performance, then I propose two factors that are related to team similarity, which can alleviate the negative effect of hierarchy on team performance.

2.3. Theory and Hypotheses

Knowledge sharing within the team is key to enhance performance because increased knowledge sharing leads to a more comprehensive consideration of alternatives and better utilization of existing knowledge within a team, which increases the chance of making better decision (Choi, Lee, & Yoo, 2010; Lee et al., 2010; Srivastava, Bartol, & Locke, 2006). The evidence that knowledge sharing increases team performance is abundant in past research. For example, Srivastava, Bartol, and Locke (2006) found that knowledge sharing within the management teams in 102 hotel properties in the U.S. is positively related to management team performance. In a different setting that involves 139 on-going teams from two major firms in South Korea, knowledge sharing improves team performance by increasing knowledge utilization (Choi, Lee, & Yoo, 2010). Based on surveys collected from 34 engineering project team, Lee and colleagues (2010) also showed that team knowledge sharing significantly predicts team performance.

However, the hierarchy literature suggests that hierarchical differentiation hinders the knowledge sharing between high-status team members and low-status team members (Anicich, Swaab, & Galinsky, 2015; Keum & See, 2017). Status is the extent to which an individual is respected or admired by others (Anderson & Kilduff, 2009; Piazza & Castellucci, 2014). Past research in status literature suggests that high-status people are more influential than lower-status ones. Numerous experimental and observational research finds that, in task-oriented groups, the ideas of high-status members carry greater weight in determining the solutions a group adopts and the directions it takes (Berger, Rosenholtz, & Zelditch, 1980; De Kwaadsteniet & Van Dijk, 2010; Eckel & Wilson, 2007). In addition, hierarchical differentiation on the basis of status comes with an expectation that high-status actors should command deference from low-status ones (Berger, Cohen, & Zelditch, 1972; Gould, 2002), which prevents low-status members from voicing their potentially valuable insights (Anicich, Swaab, & Galinsky, 2015; Keum & See, 2017). This view is supported by empirical evidences in the hierarchy literature. For example, in explaining why there is higher mortality in climber teams from more hierarchical countries, Anicich, Swaab, and Galinsky (2015) used an experimental study to show that the mechanism is that low-status members are less likely to share their knowledge with high-status members in hierarchical teams than in egalitarian teams. Keum and See (2017) documented a similar finding in a different setting, finding that employees in more hierarchical organizations are less likely to voice their new ideas.

Because hierarchy prevents low-status members from sharing their potentially valuable perspectives, which undermines the knowledge utilization within the team, thereby adversely impacting performance, in line with the findings from the recent meta-analysis, I start with the baseline hypothesis that:

> ***Hypothesis 1***: *Hierarchy is negatively related team performance.*

Since the mechanism that drives the negative relationship between hierarchy and team performance is a lack of knowledge sharing between high-status members and low-status members, I propose that one factor that can promote information sharing between high-status and low-status team members is team familiarity. Team familiarity could emerge in two ways. One way that team familiarity can be established is by team members' prior experience in working with one another, such that they know which individual has what knowledge within the team (Huckman & Staats, 2011). This familiarity with who knows what within the team can encourage low-status members to speak up for two reasons. A second way that team familiarity can emerge is from team members' prior experience in working in hierarchical teams, such that they are familiar with hierarchical working relationships. In the rest of this section, I will elaborate in turn how these two types of team familiarity alleviate the negative relationship between hierarchy and team performance.

First, knowledge is distributed among team members (Lewis, 2004). Before using new knowledge, a team member must first locate it (Staats, 2012). If team members have worked with each other before, they might develop a Transactive Memory System (TMS) that includes a representation of which person knows what (Lewis, 2003). If team members have not worked with each other before, then low-status members have little idea of what other team members know, and they are more likely to be fearful that their propositions will be dismissed by other team members, which is a threat to low-status members because they might be considered by other team members to be incompetent. Past research suggests that low-status individuals show a psychological threat response when they are being evaluated by others (Scheepers & Ellemers, 2005) and this threat leads low-status individuals to keep silent (Kish-Gephart et al., 2009). As a result of these concerns, low-status members end up being less likely to speak up.

In comparison, if low-status members know what other team members know, they are more likely to speak up because they can be more confident that their perspective will be valuable or novel to the team. Individuals who are able to combine knowledge are more likely to come up with novel ideas (Taylor & Greve, 2006), which is less likely to be dismissed by other team members. Therefore, team familiarity encourages low-status members to speak up because it reduces their uncertainty and anxiety about social acceptance and being dismissed (Harrison et al., 2003; Hinds et al., 2000)

Second, status serve as a cue for team coordination, which means that low-status individuals are inclined to defer to the preferences of high-status individuals (De Kwaadsteniet & Van Dijk, 2010). For this reason, high-status members play a central role in directing the actions of low-status members (Edmondson, 2003; Sauer, 2011). If high-status members are not aware of what low-status members know, they are less likely to provide opportunities for low-status members to speak up, because they might consider low-status members to be less knowledgeable and less likely to have something valuable to share (Oldmeadow & Fiske, 2007). By contrast, past research suggests that team leaders can promote team members to speaking up by reducing their concerns about status difference (Edmondson, 2003). Therefore, when high-status members know what low-status members know, high-status members are more likely to encourage low-status members to speak up and contribute their ideas when necessary by providing suitable opportunities.

In summary, I argue that team familiarity improves low-status members' confidence to speak up and also enables high-status members to provide more opportunities to low-status members to voice their insights. These two arguments lead me to hypothesize that:

> ***Hypothesis 2***: *Team familiarity established by prior shared working experience weakens the negative relationship between hierarchy and team performance, such that the relationship between hierarchy and team performance is less negative for teams with more familiarity and more negative for teams with less familiarity.*

In Hypothesis 2, I build on the idea that team familiarity can be built by team members' prior experience of working with one another. A second way that team familiarity can emerge is from team members' prior experience in working in hierarchical teams, such that they are familiar with hierarchical working relationships. Experiential learning theory suggests that people learn from their experience, which as a result influences their current behavior and performance (Kolb, Boyatzis, & Mainemelis, 2001). Based on experiential learning theory, I argue that people learn from their prior experience in working in hierarchical teams, such that this experience makes them better at dealing with hierarchical working relationships. The familiarity with hierarchical working relationship can promote voice from low-status members for two reasons.

First, Rentsch, Heffner, and Duffy (1994) suggests that team members with more experience conceptualized teamwork more concisely, and also that individuals who have more experience can express consistently what they understand about teamwork, compared to individuals with less experience. In line with this reasoning, if low-status members have worked in hierarchical teams before, they are more likely to understand how to voice their insights in ways that are more likely to be taken seriously/properly considered by other team members. In comparison, without such familiarity, low-status members would have more uncertainty about how, when, or whether their views will be taken into proper consideration by others. This uncertainty would lead low-status members to make safer choices in voicing their opinion or be less likely to speak up on the whole.

Second, if high-status members have worked in hierarchical teams before, they are more likely to be aware that low-status members might keep silent or voice safe opinions only because they are afraid of making mistakes or being dismissed. This understanding is valuable because high-status members can accordingly find ways to reduce low-status members' concern such that they can voice their insights.

In summary, I argue that both low-status members and high-status members learn from their prior experience in working in hierarchical teams. Such familiarity with hierarchical working relationships ends up leading to conditions that enable low-status members to voice their insights. Based on these arguments, I predict that:

> ***Hypothesis 3***: *Team members' familiarity with hierarchical working relationships weakens the negative relationship between hierarchy and team performance, such that the relationship between hierarchy and team performance is less negative for teams with more familiarity and more negative for teams with less familiarity.*

2.4. Data and Method

2.4.1. Empirical Setting

My data is collected from an online crowdsourcing contest platform for machine learning contests called Kaggle. Kaggle was founded in April 2010. The contests are hosted by different types of organizations, including major ones such as Google, the University of Melbourne, the U.S. Department of Homeland Security, and Quora. On average, contests last for 87 days and attract 873 teams. Participants can join contests alone or as part of a team; in either case each submitting "actor" is referred to as a team on Kaggle. The contest host prepares the data and a description of the problem. Kaggle offers a service to help the host in this preparation, as well as in framing the contest, anonymizing the data, and selecting an evaluation metric (e.g., area under receiver operating characteristic curve, logarithmic loss, root mean squared error) to measure the performance of the submitted results by contestants. Most contests offer monetary rewards to top teams (top three teams for most contests). On average, this monetary reward is 33,627USD. Teams are formed on a voluntary basis. For those who want to team up with others when a new contest is launched, they can contact others (through the private messaging system or on Kaggle's message boards) to ask if they are willing to form a team to join the contest, and other members can then choose to follow up on, accept, or reject such invitations.

Kaggle contests require participants to try multiple machine learning models (e.g., logistic regression, decision trees, support vector machines, and neural networks) to train a model that has the best predictive power or accuracy. The model training process involves cleaning the data, visualizing and observing the pattern in the data, extracting proper feathers (in computer science this is called feather engineering, which is to create the independent variables that have causal relationships with the dependent variable), experimenting with different machine learning models, and fine-tuning the parameters of the models.

The contest host provides two sets of data. One is called the training dataset, and the other is called the test dataset. A training dataset is a dataset of examples used for training the machine learning models. A training dataset includes a set of predictors and one outcome variable, which can be thought as a set of independent variables and one dependent variable, respectively. The participants can use this training set to train the models that they consider suitable for the setting. The test dataset is a dataset that is separate from the training dataset, but which follows the same probability distribution as the training dataset. The test dataset has the same number of predictors as the training dataset, and the naming (labelling) of the predictors are also the same as in the training dataset. However, the information for the outcome variable (the actual values) is left blank/empty for the contest participant to predict. These predicted values for the outcome variables are then used to assess the performance, or the generalizability of the models trained by the contest participants. The test dataset has one more column of unique identifiers for each row (observation) of data, which is then included in the submission for the identification and evaluation of the submitted results. During the contest, participants can make multiple submissions to get feedback on their performance of their trained models.

A submission entails submitting a CSV (comma separated values) file with two columns of data, the first column includes the unique identifiers for each row (observation) of data (as mentioned above), and the second column is the predicted results generated by the participants' trained models. The system will randomly pick a certain percentage (e.g., 40%, this percentage varies by contest, but within each contest the chose random portion is the same for all the participants and does not change during the contest) of data from this submission and use the evaluation metric to calculate a performance score. Participants can see the performance score immediately once the submission is uploaded into the system. Based on the feedback of the performance of their submissions, participants can continue to work on their models and make more submissions. On the last day of the contest, participants can pick two submissions in the system for the evaluation of their final performance. This final performance is calculated using the rest of the test dataset (e.g., in my example, with 40% being used for the evaluation of submissions made during the contest, this would be the remaining 60% of the test dataset, this portion is again the same for all the participants within a contest). The system will pick the submission (out of the up to two submissions selected by each participant) with the better performance as indicating the final performance of the participant. The goal in dividing the test dataset into two parts (e.g., the 40%

and the 60%) is to avoid over-fitting and to emphasize the generalizability (out of sample predictive properties) of the models.

For example, in a contest about Click-Through Rate Prediction, the description of the problem is “In online advertising, click-through rate (CTR) is a very important metric for evaluating ad performance. As a result, click prediction systems are essential and widely used for sponsored search and real-time bidding. For this contest, we have provided 11 days of data to build and test prediction models. Can you find a strategy that beats standard classification algorithms?” The training data includes 10 days of click-through data, ordered chronologically. The training data includes predictors such as time stamp, advertiser ID, banner position, site category, device type, and app category, and an outcome variable, which in this case is binary (0 or 1), with 0 representing that the advertisement is not clicked and 1 representing that the advertisement is clicked. The test dataset includes 1 day of data that is sampled from a different time point than the training dataset. In the test dataset, the number of predictors is the same as that in the training dataset, but information about the outcome variable is left blank/empty for the participants to predict. The evaluation metric for this contest is logarithmic loss. Logarithmic loss measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The smaller the logarithmic loss, the better the fit of the model. Logarithmic loss increases as the predicted probability diverges from the actual value. This contest awards 15,000 USD to the top three teams; 10,000 USD for first place team, 3,000 for second place team, and 2,000 for third place team.

2.4.2. Sample

My sample covers the archived Kaggle data from April 2010, when Kaggle was launched, to June 2019. My starting data includes 285 contests. I put "team" in quotes because this is the labelling that Kaggle uses, whereby each team has at least one member (Kaggle labels each and every contest entering entity as a team, whether the team consists of only one person, two people, or three or more people). Following prior team level research that is conducted using field data (e.g., Groysberg et al., 2011; Hu & Judge, 2017; Madrid et al., 2016), I define a team as having two or more members.[2] This is also consistent with my theorizing because hierarchical differentiation can form between two (e.g., Greer & van Kleef, 2010) or more individuals (e.g., Halevy et al., 2012). I exclude all "teams" (as Kaggle labels it) with only one member because there is one sole individual actor in these cases and no implication for hierarchy. Based on these considerations, my estimation sample consists of 19,937 contest-team observations, where all teams have two or more members.

2.4.3. Dependent Variable

Team performance. I measure *Team performance* using rank in contests. Rank has been widely used in management research to measure individuals' relative performance (e.g., Bhattacharjee et al., 2007; Groysberg et al., 2008; Kuhnen & Tymula, 2012). For each contest there is a final rank list based on teams' final performance. The contest host uses an evaluation metric, such as logarithmic loss, to measure the accuracy or performance of the machine learning models developed by the teams. Rank number 1 is given to the team

[2] My definition of team is consistent with a recent meta-analytic review on team tenure and team performance (Gonzalez-Mulé et al., 2020: 151), in which that the authors note that "organizations have sought to accomplish work tasks through the use of teams, defined as interdependent groups of two or more people who contribute to the parent organization's performance."

with the best performance on this metric. As in all ranked contests, the larger this “rank”, the worse the performance, i.e. the lower-ranked is the actor. The range of rank is from 1 (for the best performing team) to the total number of teams that have entered the contest. Because the total number of teams in each contest varies, I normalize the rank of each team to 0-1 range using a min-max normalization. The smaller the normalized rank, the better the team performance. Then, for easier interpretation, I reverse code this normalized rank by subtracting it from 1 and then multiplying it by 100. As a result, team performance ranges from 0 to 100, where the larger the number, the better the performance. The way I normalize the rank and calculate the team performance is shown in formula (1) below.

$$Team\,performance = \left(1 - \frac{rank - \min(rank)}{\max(rank) - \min(rank)}\right) * 100\,, \tag{1}$$

where *rank* is focal team’s rank in a contest. *min(rank)* and *max(rank)* are the minimum and maximum rank, respectively, in a contest across all teams. In my setting, *min(rank)* is 1 and *max(rank)* is the total number of teams in a contest. After min-max normalization, the best rank becomes 0 and the worst rank is transformed to 1, then I subtract it from 1 to reverse code it, such that the best rank is 1 and the worse rank is 0. Finally, these numbers are multiplied by 100 such that the best rank is 100 and worst rank is 0.

The hierarchy system in Kaggle. Kaggle has a formal hierarchy system. In line with the meritocratic ideal, members’ rank within this hierarchy is supposed to reflect their merit. Specifically, Kaggle members obtain their tiers based on their past performance and the medals they have earned. In most cases, the members of top 10 teams receive gold medals, top 11-50 teams receive silver medals, and top 51-100 teams receive bronze medals. Before July 2016, Kaggle used a slightly different hierarchy system than the one currently in use. My data and analyses span across both hierarchy systems. Below is a summary of the two systems, which I dub "old" and "new" for ease of reference.

The old hierarchy system had three hierarchical levels (from low to high): *Novice, Kaggler*, and *Master*. *Novices* are those members who have registered in the system, with no further requirement to earn this tier. *Kagglers* are members who have completed at least one contest, regardless of their performance in that contest. *Masters* are those members who have at least two top 10% finishes in contests, and where at least one of those two finishes are also in the top 10 rank, in an absolute sense.

The new hierarchy system, which has been in effect since July 2016, has five hierarchical levels (from low to high): *Novice*, *Contributor*, *Expert*, *Master*, and *Grandmaster*. *Novices* are those members who have registered in the system, with no further requirement. *Contributors* are members who have registered, but have additionally also input information about their biography, location, occupation, organization, and phone number, and ran at least one script,[3] made at least one comment, casted at least one upvote,[4] and made at least one contest submission. Kaggle notes that *Contributors* are members who have completed their profiles and have fully explored Kaggle's platform. *Experts* are members who have received at least 2 bronze medals. *Masters* are members who have

[3] A script is a piece of Python or R code written by Kaggle members to demonstrate how they analyze the data or build the statistical models. Many such scripts are shared publicly in Kaggle forums for Kaggle members to learn from each other. When browsing the code, Kaggle members can click the "fork" button in the script and run the script to see the output of the script. "Fork" is a function provided by Kaggle for members to produce a copy of someone else's script for their own use. Hence for a *Novice* to run a script, she just needs to "fork" a script and run it.

[4] When browsing the posts in Kaggle forum, Kaggle members can click a button alongside the post to cast an upvote, if they like a post, or a downvote, if they do not like the post. On the one hand, casting an upvote or downvote can improve interactions among Kaggle members. On the other hand, the number of upvotes is a signal of the quality of the post, which makes it easier for Kaggle members to find useful information.

received at least 1 gold medal and 2 silver medals. *Grandmasters* are members who have received at least 5 gold medals, where at least one of these is a solo gold medal (meaning that it is a gold medal won in a contest in which this member entered on his/her own).

2.4.4. Moderating Variables

Team familiarity established by prior shared working experience. I measure this variable as the number of actual (sometimes also referred to as "realized") ties between a team' members divided by the number of possible ties between these members (Reagans & Zuckerman, 2001). This network metric is also known as "network density" in network parlance. I assume that a tie exists between two team members if they worked with each other at least once in the 3-years that precede the focal contest (a tie between A and B exists if they entered a contest together, as part of the same team, at least once in the 3-years that precede the focal contest). I choose a 3-year moving window because on average 96.6 percent of the contest participation of Kaggle members in my estimation sample happened within a 3-year period before a given contest.

Team members' familiarity with hierarchical working relationships. For team members, I count their past experience in hierarchical teams (a team is considered to be hierarchical if it is not flat, or, alternatively, if it has any hierarchical differentiation), as normalized by their total experience. I take the average of this normalized number for all team members to measure *Team members' familiarity with hierarchical working relationships*. As I did for *Team familiarity established by prior shared working experience*, I use 3-year moving window to calculate this variable.

2.4.5. Independent Variable

Hierarchy. While some research has conceptualized hierarchy as inequality in valued characteristics or outcomes (e.g., Halevy et al., 2012), recent studies build on ethological and social network traditions to advance a view of hierarchy as cascading relations of dyadic influence (i.e. acyclicity) (Bunderson et al., 2016). I adopt this view and use acyclicity to measure *Hierarchy* in a team (Bunderson et al., 2016; Carnabuci et al., 2018). The measure of acyclicity was originally developed by Krackhardt (1994), and is calculated as shown in formula (2):

$$1 - [V/\max(V)], \qquad (2)$$

where V is the number of pairs in the team where influence is symmetric (A influences B and B also influences A, which in my case means that A and B have the same tier) and max(V) is the total number of pairs. Acyclicity values can range from 0 (where all pairs of team members are in the same tier) to 1 (where all pairs of team members are hierarchically ordered along distinct tiers).

Insert Figure 1 about here

I illustrate how this *Hierarchy* measure works in Figure 1. Team 1 has a *Hierarchy* score of 1 (the highest possible score), in which *Grandmaster* is hierarchically above *Master* and *Expert*, and *Master* is above *Expert*. Team 2 has a medium level of *Hierarchy*. Although *Grandmaster* is hierarchically above the two *Experts*, the two *Experts* belong to the same hierarchical layer. According to formula (2), *Hierarchy* in team 2 is equal to 0.667. In team 3, there is no hierarchical differentiation, as all the members belong to the same tier. According to formula (2), *Hierarchy* in team 3 is equal to 0.

2.4.6. Control Variables

I control for a number of team attributes that could contribute to *Team performance*.

Team quality. I take the average past performance of the members in a team to measure *Team quality*. Each member's past performance is measured by the average performance in the past contests that she has joined. I control for this variable because team quality is an important contributor of team performance. I calculate this variable using 3-year moving window.

Team experience. Similar to the measure of *Team quality*, I took the average of team members' experience to measure *Team experience*. Each member's experience is a count of contests that she has joined. Team members with more experience are more likely to have accumulated knowledge in past contests, which might contribute to team performance in the current contest. I use a log transformation of this measure to adjust for the skewed distribution. I calculate this variable using a 3-year moving window.

Team specialization. Across the contests in my sample, there are 78 different evaluation metrics that are used. A given contest has only one evaluation metric. To calculate a team's specialization, I use the type of evaluation metric (e.g., logarithmic loss) that a contest host uses to measure performance. Similar to the measure of *Team quality* and *Team experience*, I aggregate all contests that a team member has attended before, individually or as part of a team. Then I calculated the proportion of these contests that used the same evaluation metric as that in the focal contest. This measure ranges from 0, if a team member never attended a contest that has the same evaluation metric as the focal contest, to 1, if all the contests previously attended by the team member have the same evaluation metric as focal contest. I control for this variable because teams with higher specialization are more likely to have deeper knowledge required for the focal contest, which may give them an advantage. I calculate this variable using a 3-year moving window.

Team size. I measure *Team size* as the number of members in the team. I control for team size because prior research suggest that larger teams might be more likely to obtain more resources such as time, energy, and expertise from team members (Stewart, 2006).

Number of submissions. This variable is intended to account for how many different approaches a team experiments in a given contest. For example, a team might have tried a logistic model and a random forest model in the contest and made one submission for each method, so they have a total of two submissions. I control this variable because each contest gives the teams about three months to experiment with different methods, and in the experimentation process they can explore new solutions, therefore teams who explore more should have better performance on average. Since the duration, the difficulty, and the allowed maximum daily submission varies across contests, I normalize this variable by the maximum submissions of all teams in a given contest. After normalization, this variable ranges from 0 to 1.

All team members only appeared once. There are some teams whose members came together and attended only one contest in Kaggle. Before that one contest, none of these members have attended any other contest, individually or with others, and after that one contest, none of them attend any other contest, individually or with others. It is possible that these teams join Kaggle for that one specific contest, or they might wish to experience one contest and never intend to stay in Kaggle beyond this, or they might be discouraged by their experience or performance in that one contest. These are speculations and I do not know what the reason(s) might have been. Regardless of the reasons, these teams may confound my analyses because (a) have no hierarchical differentiation in the team (since they are all of the same, lowest, tier), and (b) they are likely to have a low performance. Therefore, I use a dummy variable *All team members only appeared once* to control for their possible confounding effects. This variable is coded 1 if all the members in a team appear only once in my sample and before and after that contest none of these members enter any other contests in Kaggle.

2.4.7. Method

I use ordinary least squares (OLS) regressions with contest fixed effects to estimate my models. I used contest fixed-effects estimations based on the results of a Hausman test ($p < 0.001$). Because of this fixed-effects estimation, control variables such as the number of teams in the contest, the duration of the contest, and the reward of the contest are not included in the model, since they are invariant within the contest and would not vary for all the observations. I use robust standard errors, clustered at the contest level to take into account the possible non-independence of observations within contests.

2.5. Results

2.5.1. Main Results

I present descriptive statistics and correlations in Table 2. I observe that the correlation between *Hierarchy* and *Team performance* is weakly positive ($r = 0.19$). For the two moderators, *Team familiarity established by prior shared working experience* and *Team members' familiarity with hierarchical working* *relationships* show positive associations with *Team performance* (r = 0.17 and 0.19).

Insert Tables 1 and 2 about here

The results of regression are shown in Table 2. Model 1 includes all the control variables. In Model 2, I add *Hierarchy*, which has a negative and significant coefficient (β = -2.196, p < 0.001) on *Team performance*. This lends support to my Hypothesis 1 that *Hierarchy* is negatively related to *Team performance*. I checked the effect size for the effect of *Hierarchy* on *Team performance*. The results suggest that one standard deviation increase in *Hierarchy* can undermine *Team performance* by 0.81. Because *Team performance* is measured by rank and normalized to a 0-100 scale, the implication is that a one standard deviation increase in *Hierarchy* lowers a team’s relative standing in a contest by 0.81 percent. Given that on average a contest has 873 teams, a 0.81 percent decrease in rank lowers a team’s standing by 7 (873 * 0.81%) positions.

In Model 3 I add the interaction between *Hierarchy* and *Team familiarity established by prior shared working experience*. The interaction effect is positive and significant (β = 3.748, p < 0.01), suggesting that *Team familiarity established by prior shared working experience* weakens the negative relationship between *Hierarchy* and *Team performance*. This provides support to my Hypothesis 2, such that the relationship between *Hierarchy* and *Team performance* is less negative for teams with more familiarity and more negative for teams with less familiarity.

I checked the effect size for the interaction effect between *Hierarchy* and *Team familiarity established by prior shared working experience*. For teams with less familiarity, one standard deviation increase in *Hierarchy* can undermine *Team performance* by 1.11. Because *Team performance* is measured by rank and normalized to a 0-100 scale, the implication is that a one standard deviation increase in *Hierarchy* lowers a team's relative standing in a contest by 1.11 percent. Given that on average a contest has 873 teams, a 1.11 percent decrease in rank lowers a team's standing by 10 (873 * 1.11%) positions. In comparison, for teams with more familiarity, one standard deviation increase in *Hierarchy* only lowers a team's standing by 4 (873 * 0.42%) positions.

In Model 4 I add the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships*. The coefficient is positive and significant (β = 11.304, p < 0.001), indicating that *Team members' familiarity with hierarchical working relationships* alleviates the negative relationship between *Hierarchy* and *Team performance*. This provides support to my Hypothesis 3, such that the relationship between *Hierarchy* and *Team performance* is less negative for teams with more familiarity and more negative for teams with less familiarity.

I checked the effect size for the interaction effect between *Hierarchy* and *Team members' familiarity with hierarchical working relationships*. For teams with less familiarity, one standard deviation increase in *Hierarchy* can undermine *Team performance* by 1.2. Because *Team performance* is measured by rank and normalized to a 0-100 scale, the implication is that a one standard deviation increase in *Hierarchy* lowers a team's relative standing in a contest by 1.2 percent. Given that on average a contest has 873 teams, a 1.2 percent decrease in rank lowers a team's standing by 11 (873 * 1.2%) positions. In comparison, for teams with more familiarity, one standard deviation increase in *Hierarchy* only lowers a team's standing by 4 (873 * 0.47%) positions.

I plot the effect of the interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* in Figure 2. The solid line shows the relationship between *Hierarchy* and *Team performance* for teams with more familiarity (i.e., 1 standard deviation above the mean) and the dashed line shows the relationship between *Hierarchy* and *Team performance* for teams with less familiarity (i.e., 1 standard deviation below the mean).

Insert Figures 2 and 3 about here

The plot for the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* is shown in Figure 3. The solid line depicts the relationship between *Hierarchy* and *Team performance* for teams with more familiarity (i.e., 1 standard deviation above the mean) and the dashed line shows the relationship between *Hierarchy* and *Team performance* for teams with less familiarity (i.e., 1 standard deviation below the mean).

2.5.2. Additional Analyses

In theorizing why hierarchy is detrimental to team performance, my argument is that hierarchy makes it less likely for low-status members to voice their possibly useful insights. My data does not allow me to test this mechanism directly. Specifically, from the data I am not able to observe whether hierarchy makes it less likely for low-status members to speak up in team discussions in this setting. However, an indirect way to test this mechanism is to see whether teams with hierarchy have more or less submissions during a contest. In other words, what I can see is whether hierarchy is positively or negatively associated with number of submissions that a team makes during a contest. My assumption is that, all else equal, if low-status members speak up and contribute their insights, the team should generate more ideas, such that they would make more submissions during a contest.

If hierarchy is positively associated with number of submissions that the teams make during a contest, then the implication is that hierarchy is functional and makes the team more productive, which means that the mechanism that I propose might not be operating. However, if hierarchy is negatively related to number of submissions that the teams make during a contest, then one possibility is that because hierarchy prevents low-status members from voicing their insights, teams with hierarchy should generate fewer ideas compared to teams without hierarchy, such that teams with hierarchy end up making fewer submissions. In summary, the indirect way to test my argument is to see whether number of submissions mediates the relationship between hierarchy and team performance.

Insert Table 3 about here

In order to test this mediating relationship, I conducted a Sobel test using the command *sgmediation* in STATA to test whether the number of submissions mediates the relationship between hierarchy and team performance. The results suggest that the number of submissions significantly mediates the relationship between hierarchy and team performance ($z = 7.66$, $p < 0.001$). In Table 3, I use *Hierarchy* to predict *Number of submissions*. In Model 2 I see that *Hierarchy* is negatively associated with the *Number of submissions* ($p < 0.001$). In addition, the results in Table 2 show that *Number of submissions* is positively related to *Team performance* (p < 0.001). Finally, I calculated the proportion of total effect that is mediated. The results suggest that 30.4% of the direct effect between *Hierarchy* and *Team performance* is mediated by *Number of submissions*. These results provide indirect evidence to support my argument that hierarchy hurts team performance by preventing low-status members to voice their insights such that teams with hierarchy make less submissions compared to teams without hierarchy.

2.5.3. Robustness Checks

I conducted several checks to ensure my findings are robust. First, in Models 3 and 4 in Table 2, I enter the interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* separately in two models. If I enter these two interactions in the same model, the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant (p = 0.015), however, the interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* becomes nonsignificant (p = 0.231). The correlation between these two interactions is high (r = 0.698). Even though the VIF numbers for these two interactions is less than 10, which is used as rule of thumb in management research, the high correlation between these two interactions suggests that it is improper to include both of them in the same model. As Kalnins (2018) suggests, if two variables are highly corrected and adding one variable substantially impacts the magnitudes of the other variable, which is true in my case, then the recommendation is to show separate regressions rather than putting them together in one model.

Second, because team size varies in my setting, and even though I control for team size, I also added team size fixed effects in all models to control for different dynamics of teams of different size. My results still hold. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Third, in my operationalization of *Team familiarity established by prior shared working experience*, tie strength for each dyad is either 0 (i.e., the two actors did not work with each other before) or 1 (i.e., the two actors worked with each other before). If I weigh the strength of each tie (i.e., number of times that they worked with each other before) using the maximum tie strength between each actor and any of her contacts, then for each dyad, the tie strength will be weighed by the two actors' maximum tie strength. Finally, I sum the weighted tie strength and divide it by two to get the weighted tie strength for this dyad. If I calculate *Team familiarity established by prior shared working experience* using this weighted tie strength, my results still hold. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Fourth, if I calculate *Team familiarity established by prior shared working experience* by taking the average of tie strength (i.e., number of times that the two actors in the dyad worked with each other before), my results still hold. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Fifth, some prior studies operationalize hierarchy as inequality (e.g., Halevy et al., 2012) rather than acyclicity, as I do in this study. Accordingly, I also tried to control for inequality among team members. As consistent with previous work, I measure inequality as the standard deviation of Kaggle points normalized by the mean of Kaggle points in the team. Kaggle points are accumulated on the basis of members' experience and performance, and are calculated using a particular way to discount, as detailed on the Kaggle website, so that recent accomplishments are weighted more. My results still hold after adding this additional control variable to account for inequality. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Sixth, prior research indicates that too many high-status members in the team might lead to status competition, which can impair team performance (Groysberg et al., 2011). In my setting, because *Master* and *Grandmaster* are considered to be high-status tiers, when they work together in a team, they may generate conflicts in making decisions in team tasks, which – based on the above supposition – can harm team performance. I address this concern by controlling for the proportion of *Master* and *Grandmaster* in a team and my results still hold. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Seventh, in the main estimations that I reported, the variables related to past activities (e.g., past performance, experience, and team familiarity) are calculated using a 3-year moving window. My results still hold if I calculate all the relevant variables using all experience (meaning using no cut off point). The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Eighth, I alternatively measure *Team familiarity established by prior shared working experience* using a ratio, which is the proportion of team members who worked in hierarchical team before. This variable is measured by a count of team members who have experience in working in hierarchical teams before, as normalized by team size. My results still hold when I use this different way to capture familiarity. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity* *established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

Ninth, because the distributions of *Team familiarity established by prior shared working experience* and *Team members' familiarity with hierarchical working relationships* are bimodal, if I re-operationalize these two variables using two dummy variables, such that they are equal to 0 if their original values are 0 and equal to 1 if their original values are larger than 0, my results still hold. The main effect of *Hierarchy* remains negative and significant ($p < 0.001$). The interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* remains positive and significant ($p < 0.01$) and the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* remains positive and significant ($p < 0.001$).

2.6. Discussion

Extant theory and empirical findings suggest that hierarchy could be beneficial for or detrimental to team performance. Consistent with recent meta-analytic evidence, I find that hierarchy undermines team performance. My contribution is that I propose two factors that can alleviate the negative relationship between hierarchy and team performance. These two factors are related to team familiarity. I propose that team familiarity could be built by team members' past shared working experience or by team members' familiarity with hierarchical working relationships. My findings reveal that these two types of familiarity can weaken the negative relationship between hierarchy and team performance.

The findings of this study offer practical implication regarding how individuals should team up to improve performance. If they are not familiar with each other or have no prior experience in working in hierarchical teams, it is better for individuals to team up with same-status partners, such that they are more likely to voice their insights. In my setting that majority of teams are homophilous, meaning that the majority of teams are composed of members with the same status. This phenomenon of status homophily indicates that people in this online community might realize that they have more freedom to speak up in status homophilous teams. However, for those teams with hierarchical differentiation, my additional analyses suggest that low-status members may be given fewer opportunities to express their perspectives. This raises a question regarding why these low-status people pair up with high-status partners. My speculation is that these low-status people connect to high-status partners for two reasons. First, in line with the argument in status literature, association with high-status people can improve the focal individuals' visibility and elevates the focal individual's standing in the hierarchy system (Sauder, Lynn, & Podolny, 2012). Second, the connection also serves as a channel of learning, such that the low-status members can learn from high-status members. Another insight from my findings is that people can learn from their networking relationships. Experiential learning theory suggests that people learn from their prior task-related experience. For example, surgeons learn from their experience in surgery, entrepreneurs learn from their experience in past start-ups, and students learn from their exercise. My results suggest that people learn from their experience in working hierarchical relationships. Hierarchical relationships differ from egalitarian relationships in that low-status individuals defer to higher-status individuals such that higher-status individuals command more influence. In comparison, in egalitarian relationships, people have more leeway to express themselves. My findings suggest that people learn from their experience in working in hierarchical relationships, such that they are more adept at dealing with current hierarchical relationships.

2.7. Conclusion

Using data from an online crowdsourcing contest community, I found that hierarchy jeopardizes team performance, which is consistent with recent meta-analytic evidence. Moreover, I showed that team familiarity that is emerged from team members' past shared working experience and their familiarity with hierarchical working relationships can alleviate the negative relationship between hierarchy and team performance. My findings offer practical implications regarding how people should be organized into teams to minimize the detrimental effect of hierarchy.

2.8. References

Anderson, C., & Kilduff, G. J. 2009. The pursuit of status in social groups. ***Current Directions in Psychological Science***, 18(5): 295-298.

Anderson, C., & Willer, R. 2014. Do status hierarchies benefit groups? A bounded functionalist account of status. In ***The psychology of social status*** (pp. 47-70). New York, NY: Springer.

Anicich, E. M., Swaab, R. I., & Galinsky, A. D. 2015. Hierarchical cultural values predict success and mortality in high-stakes teams. ***Proceedings of the National Academy of Sciences***, 112(5): 1338-1343.

Berger, J., Cohen, B. P., & Zelditch Jr, M. 1972. Status characteristics and social interaction. ***American Sociological Review***, 37(3): 241-255.

Berger, J., Rosenholtz, S. J., & Zelditch Jr, M. 1980. Status organizing processes. ***Annual Review of Sociology***, 6(1): 479-508.

Bhattacharjee, S., Gopal, R. D., Lertwachara, K., Marsden, J. R., & Telang, R. 2007. The effect of digital sharing technologies on music markets: A survival analysis of albums on ranking charts. ***Management Science***, 53(9): 1359-1374.

Bunderson, J. S., Van Der Vegt, G. S., Cantimur, Y., & Rink, F. 2016. Different views of hierarchy and why they matter: Hierarchy as inequality or as cascading influence. ***Academy of Management Journal***, 59(4): 1265-1289.

Carnabuci, G., Emery, C., & Brinberg, D. 2018. Emergent leadership structures in informal groups: A dynamic, cognitively informed network model. ***Organization Science***, 29(1): 118-133.

Choi, S. Y., Lee, H., & Yoo, Y. 2010. The impact of information technology and transactive memory systems on knowledge sharing, application, and team performance: A field study. ***MIS Quarterly***, 34(4): 855-870.

Coates, D., Frick, B., & Jewell, T. 2016. Superstar salaries and soccer success: The impact of designated players in Major League Soccer. ***Journal of Sports Economics***, 17(7): 716-735.

De Kwaadsteniet, E. W., & Van Dijk, E. 2010. Social status as a cue for tacit coordination. ***Journal of Experimental Social Psychology***, 46(3): 515-524.

Eckel, C. C., & Wilson, R. K. 2007. Social learning in coordination games: does status matter? ***Experimental Economics***, 10(3): 317-329.

Edmondson, A. C. 2003. Speaking up in the operating room: How team leaders promote learning in interdisciplinary action teams. ***Journal of Management Studies***, 40(6): 1419-1452.

Gonzalez-Mulé, E., S. Cockburn, B., W. McCormick, B., & Zhao, P. Team tenure and team performance: A meta-analysis and process model. Personnel Psychology. ***Forthcoming***.

Gould, R. V. 2002. The origins of status hierarchies: A formal theory and empirical test. ***American Journal of Sociology***, 107(5): 1143-1178.

Greer, L. L., de Jong, B. A., Schouten, M. E., & Dannals, J. E. 2018. Why and when hierarchy impacts team effectiveness: A meta-analytic integration. ***Journal of Applied Psychology***, 103(6): 591-613.

Groysberg, B., Lee, L. E., & Nanda, A. 2008. Can they take it with them? The portability of star knowledge workers' performance. ***Management Science***, 54(7): 1213-1230.

Groysberg, B., Polzer, J. T., & Elfenbein, H. A. 2011. Too many cooks spoil the broth: How high-status individuals decrease group effectiveness. ***Organization Science***, 22(3): 722-737.

Halevy, N., Chou, E. Y., Galinsky, A. D., & Murnighan, J. K. 2012. When hierarchy wins: Evidence from the national basketball association. ***Social Psychological and Personality Science***, 3(4): 398-406.

Harrison, D. A., Mohammed, S., McGrath, J. E., Florey, A. T., & Vanderstoep, S. W. 2003. Time matters in team performance: Effects of member familiarity, entrainment, and task discontinuity on speed and quality. ***Personnel Psychology***, 56(3): 633-669.

Hinds, P. J., Carley, K. M., Krackhardt, D., & Wholey, D. 2000. Choosing work group members: Balancing similarity, competence, and familiarity. ***Organizational Behavior and Human Decision Processes***, 81(2): 226-251.

Huckman, R. S., & Staats, B. R. 2011. Fluid tasks and fluid teams: The impact of diversity in experience and team familiarity on team performance. ***Manufacturing & Service Operations Management***, 13(3): 310-328.

Jewell, R. T., & Molina, D. J. 2004. Productive efficiency and salary distribution: The case of US Major League Baseball. ***Scottish Journal of Political Economy***, 51(1): 127-142.

Kalnins, A. 2018. Multicollinearity: How common factors cause Type 1 errors in multivariate regression. ***Strategic Management Journal***, 39(8): 2362-2385.

Keum, D. D., & See, K. E. 2017. The influence of hierarchy on idea generation and selection in the innovation process. ***Organization Science***, 28(4): 653-669.

Kish-Gephart, J. J., Detert, J. R., Treviño, L. K., & Edmondson, A. C. 2009. Silenced by fear: The nature, sources, and consequences of fear at work. ***Research in Organizational Behavior***, 29: 163-193.

Kolb, D. A., Boyatzis, R., & Mainemelis, C. 2001. Experiential learning theory: Previous research and new directions. In Sternberg R. & Zhang L. (Eds.), ***Perspectives on cognitive learning, and thinking styles***: 228-247. Mahwah, NJ: Erlbaum.

Krackhardt, D. 1994. Graph theoretical dimensions of informal organizations. Carley, K., & Prietula, M. (Eds.). ***Computational Organizational Theory***. Hillsdale, NJ: Lawrence Erlbaum.

Kuhnen, C. M., & Tymula, A. 2012. Feedback, self-esteem, and performance in organizations. ***Management Science***, 58(1): 94-113.

Lee, P., Gillespie, N., Mann, L., & Wearing, A. 2010. Leadership and trust: Their effect on knowledge sharing and team performance. ***Management Learning***, 41(4): 473-491.

Lewis, K. 2003. Measuring transactive memory systems in the field: scale development and validation. ***Journal of Applied Psychology***, 88(4): 587-604.

Lewis, K. 2004. Knowledge and performance in knowledge-worker teams: A longitudinal study of transactive memory systems. ***Management Science***, 50(11): 1519-1533.

Magee, J. C., & Galinsky, A. D. 2008. Social hierarchy: The self-reinforcing nature of power and status. ***Academy of Management Annals***, 2(1): 351-398.

Oedzes, J. J., Rink, F. A., Walter, F., & Van Der Vegt, G. S. 2019. Informal hierarchy and team creativity: The moderating role of empowering leadership. ***Applied Psychology***, 68(1): 3-25.

Oldmeadow, J., & Fiske, S. T. 2007. System-justifying ideologies moderate status= competence stereotypes: Roles for belief in a just world and social dominance orientation. ***European Journal of Social Psychology***, 37(6): 1135-1148.

Piazza, A., & Castellucci, F. 2014. Status in organization and management theory. ***Journal of Management***, 40(1): 287-315.

Reagans, R., & Zuckerman, E. W. 2001. Networks, diversity, and productivity: The social capital of corporate R&D teams. ***Organization Science***, 12(4): 502-517.

Rentsch, J. R., Heffner, T. S., & Duffy, L. T. 1994. What you know is what you get from experience: Team experience related to teamwork schemas. ***Group & Organization Management***, 19(4): 450-474.

Sauder, M., Lynn, F., & Podolny, J. M. 2012. Status: Insights from organizational sociology. ***Annual Review of Sociology***, 38: 267-283.

Sauer, S. J. 2011. Taking the reins: The effects of new leader status and leadership style on team performance. ***Journal of Applied Psychology***, 96(3): 574-587.

Scheepers, D., & Ellemers, N. 2005. When the pressure is up: The assessment of social identity threat in low and high status groups. ***Journal of Experimental Social Psychology***, 41(2): 192-200.

Simpson, B., Willer, R., & Ridgeway, C. L. 2012. Status hierarchies and the organization of collective action. ***Sociological Theory***, 30(3): 149-166.

Srivastava, A., Bartol, K. M., & Locke, E. A. 2006. Empowering leadership in management teams: Effects on knowledge sharing, efficacy, and performance. ***Academy of Management Journal***, 49(6): 1239-1251.

Staats, B. R. 2012. Unpacking team familiarity: The effects of geographic location and hierarchical role. ***Production and Operations Management***, 21(3): 619-635.

Stewart, G. L. 2006. A meta-analytic review of relationships between team design features and team performance. ***Journal of Management***, 32(1): 29-55.

Taylor, A., & Greve, H. R. 2006. Superman or the fantastic four? Knowledge combination and experience in innovative teams. ***Academy of Management Journal***, 49(4): 723-740.

2.9. Tables

Table 1. Descriptive statistics and correlations (N = 19,937)

		Mean	S.D.	1	2	3	4	5	6	7	8	9
1	Team performance	60.86	29.73									
2	Hierarchy	0.20	0.37	0.19								
3	Team members' familiarity with hierarchical working relationships	0.05	0.13	0.19	0.33							
4	Team familiarity established by prior shared working experience	0.16	0.35	0.17	0.09	0.43						
5	Team experience	0.79	1.16	0.38	0.55	0.47	0.37					
6	Team quality	25.91	33.66	0.37	0.57	0.42	0.34	0.82				
7	Team size	2.74	1.38	0.09	-0.03	0.04	-0.05	0.09	0.05			
8	Team specialization	0.02	0.07	0.13	0.22	0.17	0.13	0.36	0.31	0.01		
9	Number of submissions	0.12	0.19	0.39	0.17	0.16	0.09	0.37	0.30	0.14	0.13	
10	All team members only appeared once	0.28	0.45	-0.29	-0.34	-0.22	-0.28	-0.43	-0.48	-0.05	-0.17	-0.20

Note: Correlations above |0.03| are significant at $p < 0.05$.

Table 2. Contest fixed effects OLS models predicting *Team performance*.

	(1)	(2)	(3)	(4)
Team experience	1.312***	1.513***	1.525***	1.586***
	(0.319)	(0.315)	(0.316)	(0.315)
Team quality	0.125***	0.132***	0.136***	0.136***
	(0.010)	(0.010)	(0.010)	(0.010)
Team size	0.444**	0.394*	0.390*	0.389*
	(0.158)	(0.157)	(0.157)	(0.156)
Team specialization	-1.351	-1.204	-1.071	-1.144
	(2.972)	(2.964)	(2.987)	(3.001)
Number of submissions	66.285***	65.964***	65.928***	65.897***
	(2.031)	(2.046)	(2.045)	(2.042)
All team members only appeared once	-8.468***	-8.638***	-8.782***	-8.734***
	(0.692)	(0.697)	(0.701)	(0.702)
Team members' familiarity with hierarchical working relationships	0.606	1.498	0.056	-4.059+
	(1.435)	(1.474)	(1.576)	(2.184)
Team familiarity established by prior shared working experience	3.608***	3.091***	2.243**	2.888***
	(0.725)	(0.737)	(0.844)	(0.731)
Hierarchy		-2.196***	-3.073***	-3.346***
		(0.642)	(0.748)	(0.731)
Team familiarity established by prior shared working experience * Hierarchy			3.748**	
			(1.378)	
Team members' familiarity with hierarchical working relationships * Hierarchy				11.304***
				(3.015)
Constant	49.039***	49.370***	49.537***	49.495***
	(0.537)	(0.551)	(0.560)	(0.551)
N	19,937	19,937	19,937	19,937
R^2	0.287	0.287	0.287	0.288

Robust standard errors, clustered at contest level, are in parentheses.
$^{+}$ $p < 0.10$, * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. Two-tailed tests.

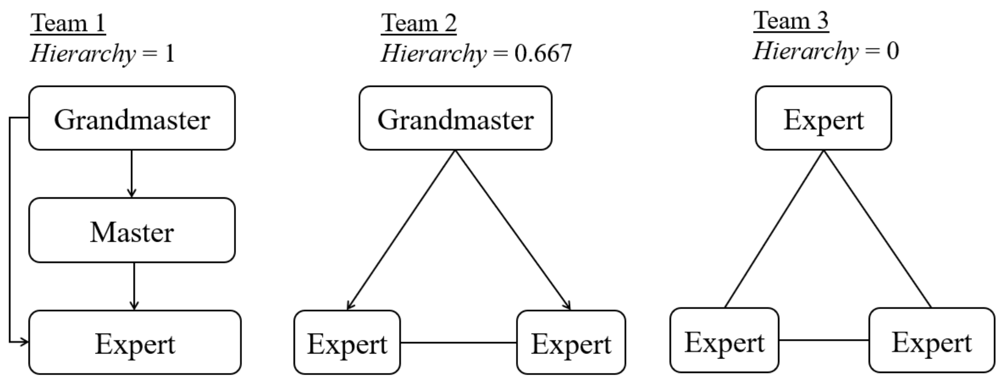
<u>Table 3. Contest fixed effects OLS models predicting *Number of submissions*.</u>

	(1)	(2)
Team members' familiarity with hierarchical working relationships	-0.004	0.011
	(0.011)	(0.011)
Team familiarity established by prior shared working experience	-0.035^{***}	-0.044^{***}
	(0.005)	(0.005)
Team experience	0.060^{***}	0.063^{***}
	(0.004)	(0.004)
Team quality	0.000	0.000^{*}
	(0.000)	(0.000)
Team size	0.014^{***}	0.013^{***}
	(0.002)	(0.002)
Team specialization	-0.033	-0.030
	(0.022)	(0.022)
All team members only appeared once	-0.016^{***}	-0.019^{***}
	(0.003)	(0.003)
Hierarchy		-0.039^{***}
		(0.005)
Constant	0.046^{***}	0.051^{***}
	(0.005)	(0.005)
N	19,937	19,937
R^2	0.192	0.197

Robust standard errors, clustered at contest level, are in parentheses.
* $p < 0.05$, *** $p < 0.001$. Two-tailed tests.

2.10. Figures

<u>Figure 1. Illustration of the *Hierarchy* measure.</u>



Note: A line with an arrow indicates the direction of the influence, from a higher-status team member to a lower-status team member. A solid line without an arrow means that the two team members are of the same status, such that they do not have influence over each other.

Figure 2. Plot of the interaction between *Hierarchy* and *Team familiarity established by prior shared working experience* on *Team performance* (using model 3 in Table 2)

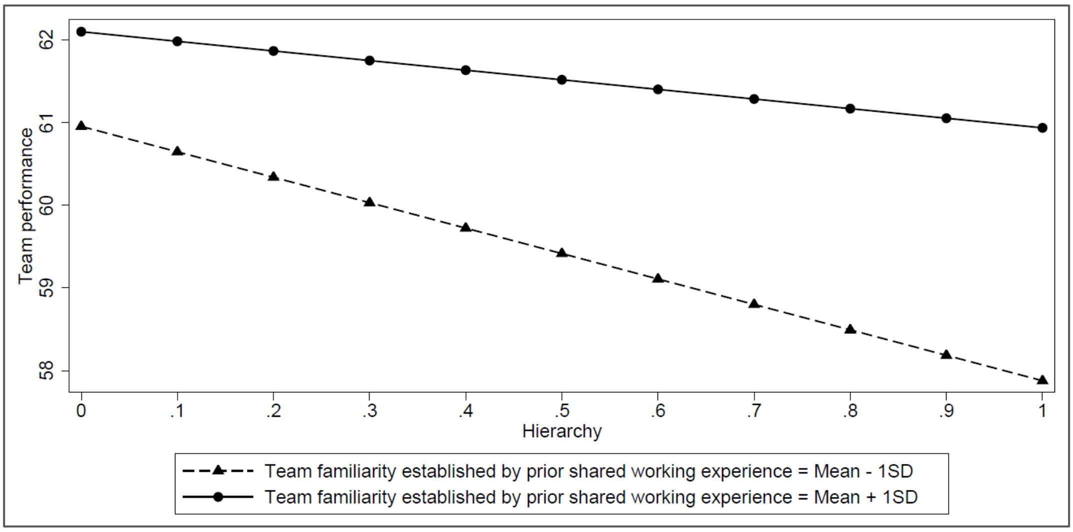
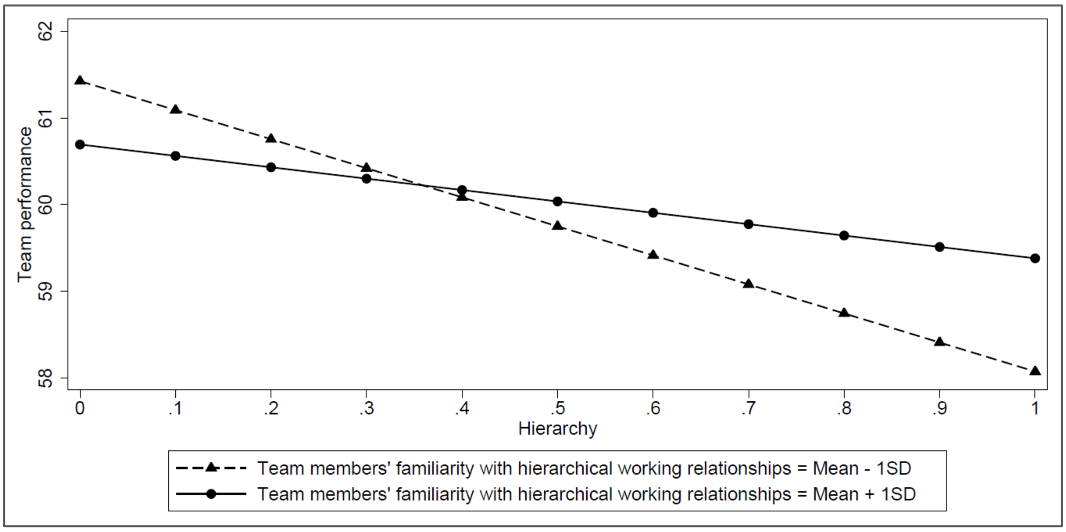


Figure 3. Plot of the interaction between *Hierarchy* and *Team members' familiarity with hierarchical working relationships* on *Team performance* (using model 4 in Table 2)



3. Does Status Drive Learning from Failure? Evidence from an Online Programming Contest Community

3.1. Introduction

Learning is a process whereby knowledge is created through the transformation of prior experience (Kim, 1993), potentially translating this experience into superior performance (Kolb, 1984). A key dimension of prior experience is whether an outcome was a success or a failure (KC, Staats, & Gino, 2013). Failures provide valuable learning opportunities for individuals to correct their practices, and learning from failure is critical to improve performance (Sitkin, 1992). However, failures are also unpleasant experiences, which people tend to react defensively against (Brown & Dutton, 1995; Shepherd, Patzelt, & Wolfe, 2011; Sitkin, 1992; Taylor, 1991). These kinds of reactions might disengage people from their failure experience, so that they pay no attention to it (Eskreis-Winkler & Fishbach, 2019). Such disengagement undermines learning because people cannot learn from their failures that they do not pay attention to.

Since it is difficult for people to learn from their failures, I ask what factors drive learning from failure. In answering this question, I bring status theory into the literature on learning from failure, and propose that status can drive people's learning from their failures. I consider status to be a pertinent factor because status is a relational attribute that colors people's attention to and evaluation of each other (Magee & Galinsky, 2008). I propose that failure feedback given by a higher-status source is more likely to drive a focal individual to learn from her failures than failure feedback given by a lower-status source. Status literature offers two mechanisms that I use to support this proposition. First, status elicits attention, such that people pay more attention to failure feedback given by a higher-status source than to failure feedback given by a lower-status source (Azoulay, Stuart, & Wang, 2013; Kovács & Sharkey, 2014; Simcoe & Waguespack, 2011). Second, status biases evaluations because evaluators perceive that higher-status individuals are more competent and thus evaluators are more likely to defer more to them (Berger, Cohen, & Zelditch, 1972), holding constant quality. As a result, failure feedback given by a higher-status source will lead to more engagement with such feedback by the focal individual than failure feedback given by a lower-status source. Such engagement drives people to exert greater effort to scrutinize the causes of their failure and adapt their behavior to avoid failure again.

I test my predictions using data collected from an online programming contest community, where contestants could receive failure feedback given by peers. The large number of individuals I can trace, the objective failure and performance metrics, and the status information available on the individuals and peers make this setting suitable to test my predictions. By demonstrating that status is a driver of learning from failure, I expand experiential learning theories by incorporating status theory. My findings indicate that individuals are likely to pay more attention to failure feedback given by higher-status peers and that they are likely to pay less attention to failure feedback given by lower-status peers, which influences their learning accordingly. With this illustration, I open a broad avenue for future research to investigate other characteristics of the sources of failure feedback that can facilitate or inhibit the focal individuals' learning from their failures.

3.2. Literature Review

Failure is performance that falls short of a desired outcome (Cannon & Edmondson, 2001). For example, an employee's creative idea might be rejected by an organization's evaluation committee (Wilhelm et al., 2019), a surgeon's cardiac procedure performed on a patient might be unsuccessful (KC et al., 2013), a research scientist might not find the expected outcome in her research project (Shepherd, Patzelt, & Wolfe, 2011), or an entrepreneur's new venture might not survive (Ucbasaran et al., 2013). Learning from failure is a process by which individuals identify failure events, analyze such events to find their causes, and search for and implement solutions to prevent similar failures in the future (Dahlin et al., 2018). Although failures offer valuable opportunities to learn (Sitkin, 1992), they are also unpleasant experiences that dent individuals' self-esteem (Brown & Dutton, 1995), and therefore tend to trigger defensive reactions (Schneider & Turkat, 1975; Shepherd et al., 2011). These kinds of reactions create motivational shutdown, which impairs people's motivation to attend to the failures (Eskreis-Winkler & Fishbach, 2019). Such disengagement from failures would undercut individuals' learning from these failures because people cannot learn from failures that they do not pay attention to.

Even though past studies have looked into whether individuals learn from their failures, the findings are mixed. Some studies found that individuals learn from their failures. For example, in a quasi-field experiment, Ellis and Davidi (2005) showed that the performance of soldiers performing successive navigation exercises improved significantly when they were debriefed on their failures after each training day. Similarly, Ellis, Mendel, and Nir (2006) reported that students' performance is improved after they experienced failed events in a laboratory study. Likewise, Shepherd and colleagues (2011) found, in their study of 257 research scientists in 12 different research institutes in Germany, that individuals learnt from the failures of their projects.

In contrast, another body of work suggests that learning from failure is not straightforward. For example, Bennett and Snyder (2017) used data on liver transplantation and found no direct evidence of learning from failure. KC, Staats, and Gino (2013) used 10 years of data from 71 cardiothoracic surgeons who completed more than 6,500 procedures to show that surgeons do not learn from their own failures. In five experimental studies, Eskreis-Winkler and Fishbach (2019) find that participants do not learn from their failures.

Even though prior studies did not specify how their failures are revealed to the focal individuals, in many cases, learning from failure is mediated through external parties who provide feedback (Ilgen & Davis, 2000). One of those external sources is peer feedback and, in the case of learning from failures, failure feedback, i.e., when peers identify and make the focal individual's failure visible to this individual. For example, within organizations, the failure feedback could come from the managers or other colleagues (i.e., peers). As Grote (2015) noted "But at the moment when an unhappy colleague is telling you loudly that the project plan you created left out some obvious key components, or your boss is taking you to task for the stumbles you made in running an important meeting, it's hard to recall these valid pointers, move them to the front of your mind, and actually act on them."[5] Failure feedback is a form of information-based opportunity to learn (Dahlin et al., 2018): it provides the learner with information about how to improve future performance. Failure feedback reminds people who receive the feedback that their performance is substandard (Taggar & Brown, 2006). In sum, failure feedback can offer a strong information-based opportunity to link an individual's failure and her future performance.

Recent research advances a view of social context as an integral part of learning from failure (Wilhelm et al., 2019) and calls for incorporating social context further into investigations of learning from failure. In response to this call, I bring insights from the literature on status into the literature on learning from failure and suggest that peers' status plays an important role in determining whether an individual learns from failure feedback given by peers. In the next section, I link status theory and literature on learning from failure to theorize how status drives learning from failure.

3.3. Theory and Hypothesis

Status is a generalized evaluation that goes beyond performance and signals an individual's position in a social hierarchy (Magee & Galinsky, 2008; Piazza & Castellucci, 2014). Peers' status is pertinent in deepening our understanding of learning from failure because (i) status is a pervasive social phenomenon that influences people's attention and evaluation of each other (Magee & Galinsky,

[5] Source: https://hbr.org/2015/08/how-to-handle-negative-feedback

2008; Sauder et al., 2012) and thus the evaluation of the information emanating from other sources and (ii) status orderings serve a motivational function that provides incentives for individuals to try to ascend to higher positions (Magee & Galinsky, 2008) and thus creates a basis for social comparison.

Ilgen and Davis (2000) argue that individuals have the choice to act on or ignore failure feedback from peers. In the former situation, how much individuals invest determines how much they can learn from the failure feedback and improve their future performance. I propose that failure feedback given by higher-status peers is more effective in driving a focal individual's learning from her own failure. Status literature offers two mechanisms that I use to support this proposition. First, higher status source elicits more attention from audiences. Prior studies in various settings lend support to this relationship. For example, Simcoe and Waguespack (2011) found, in a natural experiment, that working papers from high-status authors receive more attention on electronic discussion boards. By comparing reader reviews of 64 books that won or were short-listed for awards between 2007 and 2011, Kovács and Sharkey (2014) showed that prizewinning books attract more readers' attention following the announcement of an award than only short-listed books. In a different setting, Azoulay, Stuart, and Wang (2013) used a matched sample (i.e., quality is matched between the treatment and control groups) and difference-in-difference estimates to show that high-status scientists receive more citations. Consistent with this reasoning and these findings, I argue that failure feedback given by higher-status peers will engage more of a focal individual's attention to scrutinize these failure events, to identify problems, and to search for solutions. In contrast, I expect that the focal individual pays less attention to failure feedback given by lower-status peers, such that these failures are likely to be downplayed, minimizing the focal individual's engagement with them, therefore being less likely to trigger learning.

Second, status theory suggests that status biases evaluations, because evaluators perceive that higher-status individuals are more competent and thus defer more to them (Berger, Cohen, & Zelditch, 1972; Kim & King, 2014), holding constant quality.[6] In line with this reasoning, I contend that failure feedback given by higher-status peers will be engaged with more thoroughly by the focal individual. This is because higher-status peers command more deference from the focal individual, and therefore, such feedback is more likely to be scrutinized carefully by the focal individual (Halperin et al., 1976). As a result, this is more likely to promote the focal individual's learning.

When a focal individual's failure is pointed out by another individual, the challenge for the focal individual is to accept her responsibility for substandard performance (Ilgen & Davis, 2000: 561). Halperin and colleagues (1976) suggest that status has a significant effect on people's acceptance of negative feedback from others. Specifically, in a laboratory experiment, they found that negative feedback given by a higher-status source (i.e., a PhD clinical psychologist) generated more acceptance for the subjects than negative feedback given by a lower-status source (i.e., an undergraduate who had recently earned a mental health technician degree

[6] I conducted a survey to show that the failure feedback given by higher-status peers and lower-status peers in my setting does not significantly differ in quality. I detail the survey procedures and findings in Appendix 2.

at a nearby junior college). As a result, when receiving failure feedback from higher-status peers, because the negative feedback is engaged with and scrutinized more thoroughly by the focal individual, the focal individual would be more likely to spend time and effort understanding the causes of failure and revising the task practices or routines to avoid failing again.

In summary, people pay more attention to failure feedback given by higher-status peers and are more engaged with failure feedback given by higher-status peers. These two arguments lead me to hypothesize that:

> ***Hypothesis:*** *An individual learns more from prior failure feedback given by higher-status peers than from prior failure feedback given by lower-status peers, such that prior failure feedback given by higher-status peers has a greater effect on the individual's performance than does prior failure feedback given by lower-status peers*.

3.4. Data and Method

I collected data from CPC,[7] an online community that hosts competitive programming contests. My data covers from February 2010, when CPC was launched, to October 2018, with a total of 988 contests. On average, CPC organizes a contest every 4 days, and about 1,775 contestants are registered in each contest. Contests last for 4.8 hours on average, with 75 percent of them lasting for 2 hours.

[7] The online community where I collected data from is available upon request.

3.4.1. Empirical Setting

Problem. Contestants have to register to participate in a contest. The registration opens six hours before the start of the contest and closes five minutes before it starts. Typically, contestants are given five problems to solve in a contest. The five problems are indexed from A to E. The difficulty of problems increases from A to E. I show one of the simplest problems in a contest in CPC in Figure 1. The problem is usually about one to two pages in length, which includes a description of the problem, a description of input data, a description of the expected output data, time limit per test, and memory limit per test.

Insert Figure 1 here

Here I use a simple made-up example to explain a problem in a contest. If the problem were to ask the contestants to calculate the area of a circle, then the description would be detailed information about how to calculate the area of a circle, the input data would be numerous sets of radiuses, and the expected output data is the calculated areas corresponding to the given radiuses. The contestants are expected to write code using certain programming languages to realize the calculation of area of a circle. The time limit per test is the longest time that the code can take to finish the calculation of the output data for one set of radiuses when it is executed by the system, and the memory limit per test is the maximum computer memory that the code can use during the calculation for one set of radiuses when it is executed by the system. To solve the problem, the contestants need to employ knowledge in various domains, such as algorithms, data structure, and mathematics. The required knowledge for a problem is denoted by the tags of that problem, which is similar, for example, to the genre of a movie, book, or video game. CPC has a total of 36 tags, and the top five most frequently used tags are implementation, mathematics, dynamic programming, greedy algorithm, and data structure.[8]

As shown in the bottom of Figure 1, the examples of input data and output data are provided following the problem description. The problems in CPC contests are usually practically meaningful. For example, the problem in Figure 1 is to require the participants to convert information from one format to another format, which is similar to what management scholars do when converting date information from string format (e.g., 15Jan2008) to integer format (e.g., 17546) using the function *date* in STATA. For the problem in Figure 1, contestants are required to write code using certain programming language (e.g., C, C++, Java, and Python) to realize the function of converting the information from one format to another format.

Problem solution. Before the contest begins, all registered contestants are randomly split into rooms.[9] On average, a room contains 26 contestants. The

---

[8] *Implementation* is the realization of an idea using code. For example, a problem may describe an idea to measure cultural distance using Mahalanobis distance and give the formula, then contestants need to realize or implement it using code. *Mathematics* problems require contestants to employ some mathematics knowledge, such as calculus, algebra, and statistics, to solve the problem. *Dynamic programming* refers to simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner. A *greedy algorithm* is an algorithmic paradigm that follows the problem-solving heuristic of making the locally optimal choice at each stage with the intent of finding a global optimum. *Data structure* is a data organization, management and storage format that enables efficient access and modification of data.

[9] I note that in 18.7% of the contests, contestants are not assigned randomly into rooms. These include two types of contests. In the first type of contests, the hosts aim to simulate traditional on-site contests, where contestants only compete on solving problems and are not allowed to hack each other. Because hacks are not allowed, room randomization is not implemented. The second type of contests is known as educational contests, in which contestants are again not allowed to hack during contests. Instead, they are given 12 hours or 24 hours to hack each other *after* the contests. Because each contestant is allowed to

problems are displayed as soon as the contest begins, which means that all the contestants have access to the problems at exactly the same time. During the contest, the contestants submit solutions to each problem. In almost all cases, the contestants solve the problems in order of difficulty (i.e., from A to E). A problem solution is a piece of code written in one of the programming languages (e.g., C, C++, Java, or Python), as indicated by CPC in the contest rules. An example of problem solution written in Python is shown in Figure 2. The function of the code shown in Figure 2 is to read the input data (as given by the system) and convert it to another format as described in the problem shown in Figure 1. I note that high quality code does not necessarily need to be long (in terms of number of rows) and complex, because – typically – the longer and more complex code is usually less efficient, which is more likely to exceed the time limit and memory limit as set in the problem. Elegant code is usually the result of careful analysis of the problem and finding an algorithm and design that simplifies the code. A high-quality coder writes code that looks like it was easy and straightforward to produce.

Insert Figure 2 here

Test. In the code shown in Figure 2, the function *input()* in the first two rows is to read the input data as given by the system, then the rest of the code realizes the function of converting a number from one format to another format as required by the problem shown in Figure 1, and the *print* function in the code is to generate

hack any contestant after the contest, again room randomization is not implemented. In other words, all the contestants are in one big room in these contests.

the output. The output will be used by the system to automatically evaluate whether the solution is successful or failed. This automatic assessment involves a process of comparing the output generated by a contestant's solution with the preset correct output. If a solution produces an output that is exactly the same as the output that is preset by the system, then the solution is successful, otherwise it is failed. This judgement process is referred as test in CPC. The test includes a large number of test cases that were preset by contest organizers. The test cases are designed carefully by experts such that they can identify almost all of the possible issues that might appear in contestants' solutions. Each test case includes the input data and corresponding output data. This judgment process is done automatically by the system. The system refers to a computer that is used to test the solutions submitted by contestants. Once a solution is submitted, it will be automatically sent to this computer for test. The test result is immediately sent to the contestant when the test for that solution is done. The test cases are randomly divided into two sets. One set is used to test the solutions during contest, which is called *pretests*. The other set is used to test the solutions when the contest is over, which is called *final tests*.

If a solution passed all *pretests*, the contestant is shown a message that says "Pretests passed." Otherwise, if a solution has failed in *pretests*, the contestant is shown a corresponding error information, such as "Memory limit exceeded," "Time limit exceeded," "Runtime error," or "Wrong answer." A contestant can submit solutions to the same problem multiple times, regardless of whether the prior solutions passed *pretests* or not. If a contestant submits multiple solutions that pass *pretests*, then only the last solution is considered as the contestant's verified solution for that problem. Once the solution to a problem passes *pretests*, the problem is considered pre-solved by the contestant and the system automatically calculates the contestant's preliminary points for the given problem.

The points are calculated based on the following scheme. Each minute taken by the contestant to solve the problem decreases the problem's value: the value decreases by $X/250$ points per minute (where X is a problem's initial points or full points). The number of points a contestant gets for a problem equals the current value of the problem in points minus penalty. The penalty is determined as the number of this contestant's previous solutions for this problem, multiplied by 50 points. Thereby the intuition here is that, in order to maximize the points, a contestant should submit a solution as soon as she can and make sure the solution will pass the pretests in one shot. For example, if a problem has initial value of 500 points, and a contestant solves the problem in the third trial in two minutes, then her points for this problem is 500 – (500/250) * 2 – 2 *50 = 396.

Hack. A contestant can "lock" any of her pre-solved problems (i.e., the problems that passed *pretests*). Once the problem is locked, the contestant loses the right to resubmit solutions to this problem. At the same time, the contestant now gains the right to view the original codes of submitted solutions for this problem by other contestants who share the same room with her. After viewing another person's code, the contestant can suggest a test (i.e., a set of input data) on which, she thinks, the given solution will fail. This procedure is called "hacking" another contestant's solution. Once the test input data by a hack giver is submitted, the system automatically judges if a hack receiver's solution can pass the test. If the solution passes the test, the hack giver is notified that the hack attempt is unsuccessful. If the solution does not pass the test, then the hack giver is notified that the hack is successful, and in this case, the hack receiver is also notified at the same time that her solution has been hacked. The hacker earns 100 points for a successful hack and loses 50 points for an unsuccessful hack. For the hack receiver, after a successful hack, her solution is considered unsuccessful by the system, her preliminary score for that problem is reset to 0.

Performance. During the contest, contestants' scores are rank ordered based on *pretests* results and are visible to all the contestants. After a contest is over, the contestants' solutions go through *final tests*. A problem is considered solved by the contestant if the contestant's solution (i) passed all *pretests*, (ii) passed all *final tests*, and (iii) was not hacked during the contest. The points for contestants are recalculated after the *final tests*. The points are the sum of two parts. The first part is the sum of points earned by solving problems, and the second part is the total points calculated by successful and unsuccessful hacks. Based on the recalculated points, the contestants are rank ordered in the leaderboard, which indicates the final performance ranking in that contest. In cases when multiple contestants obtain the same points, their rank are the same in the leaderboard. An illustration of the process of a contest is shown in Figure 3.

Insert Figure 3 here

Status hierarchy system. Members in CPC are ranked based on their ratings. CPC uses an Elo rating system to rate its members (Elo, 1978).[10] Before a contest, each registered contestant' expected standing is calculated based on her rating compared to rest of the registered contestants, then after the contest, the contestant's rating goes up if her actual standing is higher than the expected standing, goes down if her actual standing is lower than the expected standing, and remains unchanged if actual standing is the same as the expected standing. If it is the very first contest by a contestant, her rating before this contest is set by CPC to 1,500 by default, which approximates the mean rating of all members in CPC.

Practice. CPC provides four ways for members to practice their programming skills. The first way is called practice, which is to directly submit solutions to problems in past contests. In this scenario, there are no time constraints for the contestants, which means that they can take as long as they wish. The solutions will be judged by system using all the tests (both pretests and final tests), and the contestants are informed whether their solutions are successful or not. However, as different from formal contests, these contestants are not ranked in the contest leaderboard, their rating will not be affected, and the function of hacking others is disabled.

The second way is called a virtual contest. A virtual contest provides a way to take part in past contests, as close as possible to participation in a formal contest, where time constraint is imposed. Contestants can start a virtual contest at any time they like. In this scenario, contestants have time constraints to solve the problems

[10] Elo rating system is widely used in sports and games, most famously in chess, but also in tennis, as well as in some video games.

in the contests, which are the same constraints as in real contests. However, as different from formal contests, these contestants are not ranked in the contest leaderboard, their rating will not be affected, and the function of hacking others is disabled.

The third way is to directly participate in a contest that is not in a contestant's rating range, which is called unqualified participation. Contests on CPC are designed for contestants with a certain rating range. Only contestants in that rating range will be ranked in the leaderboard and can gain or lose rating after the contest. Contestants who are not in that rating range will not be ranked in leaderboard and their rating remains unchanged, regardless of their "performance" in the contest. For example, some contests are for contestants' whose rating is above 1,900, which means that only those contestants whose rating is above 1,900 will be ranked in leaderboard. Other contests are for contestants' whose rating is below 2,100, which means only those contestants whose rating is below 2,100 will be ranked in the leaderboard. And, there are also contests are for all contestants, regardless of their ratings. For the 34,842,669 solutions submitted in the contests covered in my sample, 25.5% are submitted in formal contests, 67.3% are submitted in practice, 5.7% are submitted in virtual contests, and 1.4% are submitted in unqualified participation.

The fourth way is to "practice in the gym." CPC collects a list of problems in past international programming contests (e.g., The ACM International Collegiate Programming Contest) for its members to practice, which is called a gym in CPC. In the gym, contestants can participate in formal contests, practice, and virtual contests, but not in unqualified participation, as the contests in the gym are open to all contestants regardless of their ratings. The function of hacking others is disabled in the gym. For the 5,204,583 solutions submitted in the gym, 6.0% are submitted in formal contests, 37.1% are submitted in practice, and 56.9% are submitted in virtual contests. Even though contestants who participate in a formal gym contest are ranked in a gym leaderboard, unlike the leaderboard of formal contests, they do not gain or lose rating in these gym contests. The distribution of solutions for different participation types are shown in Table 11 in Appendix 1.

3.4.2. Dependent Variable

Performance. I measure *Performance* using rank in the contests, which reflects a contestant's relative performance compared to others in these contests. Rank has been widely used in management research to measure individuals' relative performance (e.g., Bhattacharjee et al., 2007; Groysberg et al., 2008; Kuhnen & Tymula, 2012). As I introduced before, contestants are ranked based on their total points obtained in the contest. The points are composed of two parts. The first part is obtained by solving the problems. The points for each problem are calculated based on the following scheme. Each minute taken by the contestant to solve the problem decreases the problem's value: the value decreases by $X/250$ points per minute (where X is a problem's initial points or full points). The number of points a contestant gets for a problem equals the current value of the problem in points minus penalty. The penalty is determined as the number of this contestant's previous solutions for this problem, multiplied by 50 points. Thereby the intuition here is that, in order to maximize the points, a contestant should submit a solution as soon as she can and make sure the solution will pass the *pretests*, not being hacked by others, and pass the *final tests*. For example, if a problem has initial value of 500 points, and a contestant solves the problem in the third trial in two minutes, then her points obtained by solving this problem is 500 – (500/250) * 2 – 2 *50 = 396. The second part is by hacking others. According to the contest rules, a contestant earns 100 points for a successful hack and loses 50 points for an unsuccessful hack. Contestants' final points are sum of points obtained in both parts. For example, if a contestant solved three problems in a contest and obtained 236, 358, and 532 for each problem. Besides, this contestant made two successful hacks and one unsuccessful hack, then her final points in this contest is 1,276 (236 + 358 + 532 + 100*2 – 50*1). The highest rank is 1 and lowest rank is equal to the number of contestants in the contest. As the total number of contestants varies across contests, I normalize the rank to 0-1 range using a min-max normalization. The smaller the normalized rank, the better the performance. Then, for easier interpretation, I reverse code this normalized rank by subtracting it from 1 and then multiplying it by 100. As a result, performance ranges from 0 to 100, where the larger the number, the better the performance. The way I normalize the rank and calculate the team performance is shown in formula (1) below.

$$Performance = \left(1 - \frac{rank - \min(rank)}{\max(rank) - \min(rank)}\right) * 100\,, \tag{1}$$

where *rank* is focal contestant's rank in a contest. *min(rank)* and *max(rank)* are the minimum and maximum rank, respectively, in a contest across all contestants. In my setting, *min(rank)* is 1 and *max(rank)* is the total number of contestants in a contest. After min-max normalization, the best rank becomes 0 and the worst rank is transformed to 1, then I subtract it from 1 to reverse code it, such that the best rank is 1 and the worse rank is 0. Finally, these numbers are multiplied by 100 such that the best rank is 100 and worst rank is 0.

3.4.3. Independent Variables

Prior failure feedback given by machine. System tests are pre-designed by experts to test the solutions submitted by contestants. As I introduced before, systems tests include *pretests* and *final tests*. Whereas *pretests* are used to test the solutions during a contest, *final tests* are used to test the solutions when the contest is over. If a solution does not pass *pretests* or *final tests*, the system displays an error message (e.g., memory limit exceeded, wrong answer, or runtime error) to the contestant, which indicates that the contestant's failure is identified by the system. I measure *Prior failure feedback given by machine* by counting the number of times that the focal contestant's prior solutions failed in system tests, in either formal contests or during practice. A logarithmic transformation was used to enhance the normality of this measure.

Prior failure feedback given by peers. I count the number of times that a contestant was successfully hacked by other contestants in prior contests to proxy *Prior failure feedback given by peers*. I use a log transformation to compensate for the skewed distribution of this measure. Because the contestant's solution could be hacked by higher-status peers, those whose rating is higher than focal contestant, or lower-status peers, those whose rating is lower than focal contestant, I split this into two variables: *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers*. *Prior failure feedback given by* *higher-status peers* is number of times that a contestant was successfully hacked by peers whose ratings were higher than or equal to the contestant's rating at the time when the hack happened. *Prior failure feedback given by lower-status peers* is number of times that a contestant was successfully hacked by peers whose ratings were lower than the contestant at the time when the hack happened. A logarithmic transformation was used for both measures.

3.4.4. Control Variables

Past performance. I take the average of the performance in prior contests that the contestant participated to measure her past performance. I control for this variable to account for contestants' quality. If a contestant performed well in past contests, she is highly likely to continue to perform well in focal contest.

Status. This variable is proxied by a contestant's Elo rating when she joined the contest. A person's status has been defined as her position in the hierarchy of a social system (Anderson & Kilduff, 2009; Piazza & Castellucci, 2014). Elo ratings capture status because they indicate a contestant's relative standing compared to other contestants. I control for it because status is a signal of quality and quality is positively related to performance. As the highest rating in my sample is 3,739, I divided this measure by 100 to facilitate the reporting of the coefficients and standard errors.

Knowledge specialization. Every problem has a list of tags (e.g., dynamic programming, data structure) that specify the knowledge required to solve the problem. I use tags to measure a contestant's *Knowledge specialization* in focal contest. Specifically, I first aggregate the tags for all problems solved by the contestant in past contests, then I sum number of times each tag in focal contest appears in the aggregated tag list. *Knowledge specialization* is measured as number of times that tags in focal contest appear in the aggregated list normalized by the total length of the aggregated list. For example, if the aggregated list of tags for a contestant is [a, a, a, b, b, c, d, e, f, g, h, h] and the tags in focal contest is [a, b, c, d, e], then the contestant's *Knowledge specialization* is (3 + 2 + 1 + 1 + 1) / 12 = 0.667. I control for *Knowledge specialization* because a contestant with high level of *Knowledge specialization* may have deeper knowledge that is required to solve the problems in focal contest, which can positively affect the contestant's performance.

Prior unsuccessful hacks from peers. For my independent variables, I only count number of times that a contestant was successfully hacked by peers in prior contests. As a hack could also be unsuccessful, I also control for *Prior unsuccessful hacks from peers*, which is number of times that a contestant was unsuccessfully hacked by peers in prior contests. A logarithmic transformation was used to enhance the normality of this measure. I control for this variable because if a contestant was unsuccessfully hacked by peers in prior contests, she might become confident about her solution, which can positively affect the performance in current contest, or she might become complacent, which can negatively affect the performance in current contest.

Experience of hacking peers in prior contests. While a contestant can be hacked by peers, she can also hack peers and the hack may be successful or unsuccessful. I use two variables to control for the focal contestant's experience of hacking peers in prior contests. The first variable is *Prior successful hacks to peers*, which is number of times that a contestant successfully hacked peers in prior contests. The other variable is *Prior unsuccessful hacks to peers*, which is number of times that a contestant unsuccessfully hacked peers in prior contests. I use a log transformation to compensate for the skewed distribution of these two measures. I control for *Prior successful hacks to peers* because successful hacks to peers can make the contestant become confident about her skill, which can positively affect the performance in current contest, or become complacent, which can negatively affect the performance in current contest. I control for *Prior unsuccessful hacks to peers* because this experience can discourage the contestant and make her feel less confident about her knowledge or skill, which could negative affect the performance in current contest, or the contestant might learn from this unsuccessful experience, and therefore her performance might be improved in the current contest.

Influence of hacks in focal contest. In a focal contest, a contestant can hack others or being hacked by others. I use four variables to control for the influence of hacking others or being hacked by others in the focal contest on a contestant's performance. The first variable is *Number of successful hacks sent by focal contestant in focal contest*, which is measured as number of times the contestant hacked others successfully in focal contest. I control for this variable because a contestant can earn 100 points for a successful hack, which positively contributes to final performance. The second variable is *Number of unsuccessful hacks sent by focal contestant in focal contest*, which is measured as number of times the contestant hacked others unsuccessfully in focal contest. I control for this variable because a contestant can lose 50 points for a successful hack, which detracts from final performance. The third variable is *Number of successful hacks received by a contestant in focal contest*, which is measured as number of times the contestant was hacked successfully by others in focal contest. I control for this variable because a contestant's solution is marked as failed after successfully hacked by others, which leads to loss of points. The fourth variable is *Number of unsuccessful hacks received by a contestant in focal contest*, which is measured as number of times the contestant was hacked unsuccessfully by others in focal contest. I control for this variable because if a contestant's solution is hacked unsuccessfully by others, she may become more confident, which can positively influence her final performance. A logarithmic transformation was used to enhance the normality of these four measures.

3.4.5. Method

I use ordinary least squares (OLS) regressions with contest, contestant, and room fixed effects to estimate my models. I used fixed-effects estimations based on the results of a Hausman test ($p < 0.001$). Because of these fixed-effects, contest characteristics (e.g., number of participants, duration), contestant attributes (e.g., gender, country), and room characteristics (e.g., room size) are not separately controlled for with individual variables in my estimations, since their effects do not vary within the given fixed-effects set in my observation period, and therefore cannot be estimated. I use robust standard errors, clustered at the contest, contestant, and room level to take into account the possible non-independence of observations within contests, contestants, and rooms.

3.5. Results

3.5.1. Main Results

In Table 1, I present the descriptive statistics and correlations. The correlation between *Prior failure feedback given by higher-status peers* and *Performance* and the correlation between *Prior failure feedback given by lower-status peers* and *Performance* are both positive (r = 0.21 and 0.31), which are consistent with my predictions in the hypothesis.

Insert Tables 1 and 2 here

My estimation results are shown in Table 2. Model 1 includes all the control variables. In Model 2, I add *Prior failure feedback given by peers* and I see that the coefficient of this variable is positive and significant (β = 2.387, p < 0.001), which means that individuals learn from failure feedback given by peers, and their subsequent performance is improved. In order to test my hypothesis, I split *Prior failure feedback given by peers* into two components: *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers*. In Model 3, I add *Prior failure feedback given by higher-status peers* but do not add *Prior failure feedback given by lower-status peers*. I see that the coefficient of this variable is positive and significant (β = 2.835, p < 0.001), which means that failure feedback given by higher-status peers contributes to performance. In Model 4, I add *Prior failure feedback given by lower-status peers* but do not add *Prior failure feedback given by higher-status peers*. I see that the coefficient of this variable is not significant (β = 0.015, p = 0.907), which means that failure feedback given by lower-status peers does not contribute to performance.

In order to test my hypothesis, in Model 5 I add both *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers*. The results are consistent with the results in Models 3 and 4, whereas the coefficient for *Prior failure feedback given by higher-status peers* remains positive and significant (β = 2.836, p < 0.001), the coefficient for *Prior failure feedback given by lower-status peers* remains nonsignificant (β = 0.034, p = 0.791). A Wald test reveals that the coefficients for *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* are significantly different from each other (F = 231.9, p < 0.001), which yields support for my hypothesis (that an individual learns more from prior failure feedback given by higher-status peers than prior failure feedback given by lower-status peers, such that the individual's prior failure feedback given by higher-status peers have a greater effect on the individual's performance than does the individual's prior failure feedback given by lower-status peers).

I checked the effect sizes for the significant independent variable (i.e., *Prior failure feedback given by higher-status peers*) in Model 5 of Table 2. The effect size for *Prior failure feedback given by higher-status peers* is such that one standard deviation increase in *Prior failure feedback given by higher-status peers* improves *Performance* by 2.4 units. Because *Performance* is measured by rank and is normalized to a 0-100 scale, this means that one standard deviation increase in *Prior failure feedback given by higher-status peers* can boost a focal contestant's

Performance in a contest by 2.4%. Given that on average each contest has 1,775 contestants, this means that one standard deviation increase in *Prior failure feedback given by higher-status peers* elevates a focal contestant's *Performance* in a contest by 43 (1,775 * 2.4%) positions.

3.5.2. Additional Analyses

Face validity of my arguments

In the theorizing of my hypothesis, I argue that people learn more from failure feedback given by higher-status peers than failure feedback given by lower-status peers because people pay more attention to and are more engaged with failure feedback given by higher-status peers than failure feedback given by lower-status peers. In order to test the face validity of my argument, I use failure feedback given by machine to interact with failure feedback given by peers to see whether failure feedback given by machine enhances the effect of failure feedback given by peers on performance. In my setting, contestants receive failure feedback from machine and peers. My rationale to investigate this interaction is that if my argument is true, then as contestants receive more failure feedback from machine, they will be more aware that they need to learn and improve their programming skills, such that they will pay more attention to and be more engaged with failure feedback given by peers.

Insert Table 3 here

I present the interaction results in Table 3. In Model 4 I see that the interaction between *Prior failure feedback given by machine* and *Prior failure* *feedback given by peers* is positive and significant ($p < 0.001$). Similarly, in Models 5 and 6, the interaction between *Prior failure feedback given by machine* and *Prior failure feedback given by higher-status peers* and the interaction between *Prior failure feedback given by machine* and *Prior failure feedback given by lower-status peers* are also positive and significant ($p < 0.001$). The plots for these three interactions are shown Figures A1-A3 in Appendix 3. These results of these three interactions suggest that as contestants receive more failure feedback given by machine, they are more aware that they need to learn, such that they pay more attention to and are more engaged in learning from failure feedback from peers, regardless of their status. In another additional analysis that I do not report in this study, I interact *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers*. I see that this interaction is also positive and significant ($p < 0.001$), suggesting that receiving failure feedback from either higher-status peers or lower-status peers also drive contestants to pay more attention to and be more engaged in learning from failure from one another. Taken together, these results suggest that the mechanism that drives the different effects for failure feedback given by higher-status peers and failure feedback given by lower-status peers is that failure feedback given by higher-status peers attracts more attention and engagement from focal individuals than failure feedback given by lower-status peers.

Alternative measure of performance

Insert Table 4 here

In the tests of my hypothesis, I measure learning using contestants' ranking in contests, such that the unit of analysis contest-contestant. As a contestant can make multiple submissions during a contest and each submission can be either a success (i.e., pass the test) or a failure (i.e., does not pass the test). Therefore, an alternative measure of learning outcome is to see a contestant's probability of success in her submissions. If a contestant's probability of success is higher, it suggests that she learns more from experience. On the contrary, if a contestant's probability of success is lower, it suggests that she learns less from prior experience. I create an alternative dependent variable, *Probability of success*, to measure the learning outcome, such that the unit of analysis is contest-contestant-submission. The results are presented in Table 4. The sequence that I enter the variables is the same as in Table 2. Overall, the results in Model 5 are consistent with those in Model 5 of Table 2. Specifically, whereas the coefficient for *Prior failure feedback given by higher-status peers* is positive and significant ($p < 0.001$), the coefficient for *Prior failure feedback given by lower-status peers* is not significant ($p = 0.106$). The Wald test for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continues to lend support to my hypothesis ($p < 0.001$). In summary, these results suggest that failure feedback given by higher-status peers have a stronger effect on probability of success than failure feedback given by lower-status peers, which is consistent my hypothesis that people learn more from failure feedback from higher-status peers than failure feedback from lower-status peers.

Alternative measure of status

In the tests of my hypothesis, I measured status using Elo rating, which is a score. If a peer's rating is larger than the focal individual's rating, then I consider the peer's status to be higher. Similarly, if a peers' rating is smaller than the focal individual's rating, then I consider the peer's status to be lower. I note that CPC also uses tiers and colors to indicate individuals' status. Specifically, individuals are differentiated based on ten tiers, in descending order, these are: Legendary Grandmaster, International Grandmaster, Grandmaster, International Master, Master, Candidate Master, Expert, Specialist, Pupil, and Newbie. These tiers are assigned based on Elo ratings. For example, Newbie is for an individual whose Elo rating is between 0 to 1199. In addition, these ten tiers are also grouped into seven colors, which are, in descending order: red, orange, violet, blue, cyan, green, and gray. Red color includes three tiers: Legendary Grandmaster, International Grandmaster, and Grandmaster. Orange color covers two tiers: International Master and Master. Each of the rest of the colors corresponds to one tier. Again, as the tiers are based on Elo ratings, these colors are also primarily determined by an individual's Elo rating. Nevertheless, as I detail below, I also tested my hypothesis using either the tier or the color in determining status differences, rather than the Elo ratings.

Insert Table 5 here

Based on the possibility that the status comparison between the focal individual and peers might not be based on the Elo rating, but rather based on either their tiers or the tier-indicative colors, I recalculate the status related variables using tier and color. Accordingly, I count a focal individual's failures that are identified by higher-tier/higher-color, same-tier/same-color, or lower-tier/lower-color peers. A summary of the results is presented in Table 5. I see the coefficient for *Prior failure feedback given by higher-status peers* remains positive and significant (p < 0.001), and the coefficient for *Prior failure feedback given by lower-status peers* remains not significant (p > 0.550), which are consistent with in the main results in Table 2. The results of the Wald tests for coefficients continue to yield support for my hypothesis (p < 0.001). Lastly, I also note that *Prior failure feedback given by same-status peers* is positive and significant (p < 0.001). Wald tests suggest that the effect of *Prior failure feedback given by same-status peers* is stronger than *Prior failure feedback given by lower-status peers*, but weaker than *Prior failure feedback given by higher-status peers*. I speculate that this is because the higher the peers' status, the more likely it can drive the focal individual to learn from her failures, as consistent with the arguments I discuss in deriving my hypothesis.

Analyses of unsuccessful hacks from peers and hacks to peers

To test my hypothesis, I split only successful hacks from peers into successful hacks from higher-status peers and successful hacks from lower-status peers. Since hacks from peers could be unsuccessful, I conduct further analyses by also splitting unsuccessful hacks from peers into unsuccessful hacks from higher-status peers and unsuccessful hacks from lower-status peers. Moreover, the focal individual can also hack peers and the hacks "sent" by the focal individual in this way could be themselves also be successful or unsuccessful. Accordingly, I split successful hacks sent to peers into successful hacks of higher-status peers' codes and successful hacks of lower-status peers' codes, and split unsuccessful hacks similarly. I present the regression results using these decomposed variables in Table 6.

Insert Table 6 and Figure 4 here

In Table 6, Model 1 includes all the variables before splitting. In Model 2, I split *Prior failure feedback given by peers* into *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers*, which is the same as Model 5 in Table 2. In Model 3, I only split *Prior unsuccessful hacks from peers* into *Prior unsuccessful hacks from higher-status peers* and *Prior unsuccessful hacks from lower-status peers*. In Model 4, I only split *Prior successful hacks to peers* into *Prior successful hacks to higher-status peers* and *Prior successful hacks to lower-status peers*. In Model 5, I only split *Prior unsuccessful hacks to peers* into *Prior unsuccessful hacks to higher-status peers* and *Prior unsuccessful hacks to lower-status peers*. Finally, in Model 6, I split all the variables (i.e., *Prior failure feedback given by peers*, *Prior unsuccessful hacks from peers*, *Prior successful hacks to peers*, and *Prior unsuccessful hacks to peers*) and put all the decomposed variables together in the same model.

First, I see that the coefficient for *Prior failure feedback given by higher-status peers* is positive and significant (β = 2.280, p < 0.001) and that the coefficient for *Prior failure feedback given by lower-status peers* is not significant (β = 0.104, p = 0.386), which are consistent with the results in Model 5 of Table 2. Second, the coefficients for *Prior unsuccessful hacks from higher-status peers* (β = 1.595, p <

0.001) and *Prior unsuccessful hacks from lower-status peers* (β = 0.563, p < 0.001) are positive and significant. I speculate that this is because contestants consider unsuccessful hacks from peers as evidence that their code is robust, which, as a result, increases their confidence and can positively affect their subsequent performance. A Wald test reveals that the effect of *Prior unsuccessful hacks from higher-status peers* on *Performance* is stronger than the effect of *Prior unsuccessful hacks from lower-status peers* on *Performance* (F = 75.3, p < 0.001), which is consistent with my speculation that contestants perceive that higher-status peers are more competent and put more weight on the output from higher-status peers, such that unsuccessful hacks from higher-status peers have a greater effect in promoting a contestant's confidence than unsuccessful hacks from lower-status peers do. Consequently, unsuccessful hacks from higher-status peers have a greater effect in improving a contestant's subsequent performance than unsuccessful hacks from lower-status peers do.

Third, the coefficients for *Prior successful hacks to higher-status peers* (β = 1.223, p < 0.001) and *Prior successful hacks to lower-status peers* (β = 0.284, p < 0.001) are positive and significant, but a Wald test suggests that the coefficients for *Prior successful hacks to higher-status peers* is statistically stronger than the coefficient for *Prior successful hacks to lower-status peers* (F = 30.5, p < 0.001). I speculate that this is because successful hacks to higher-status peers have a greater effect in boosting a contestant's confidence than successful hacks to lower-status peers do, such that successful hacks to higher-status peers contributes more to a focal contestant's subsequent performance than successful hacks to lower-status peers do.

Finally, the coefficients for *Prior unsuccessful hacks to higher-status peers* (β = 0.555, p < 0.001) and *Prior unsuccessful hacks to lower-status peers* (β = -0.274, p < 0.001) are significant, but a Wald test suggests that the coefficients for *Prior unsuccessful hacks to higher-status peers* is statistically stronger than the coefficient for *Prior unsuccessful hacks to lower-status peers* (F = 32.2, p < 0.001). I speculate that this is because unsuccessful hacks to higher-status peers are more likely to drive the focal contestant to investigate why her hacks were failed, which lead to learning and subsequent performance improvement. However, unsuccessful hacks to lower-status peers are less likely to drive the focal contestant to investigate why her hacks were failed, which therefore does not lead to learning and subsequent performance improvement. I note that the coefficient for *Prior unsuccessful hacks to lower-status peers* is negative and significant, which is consistent with KC and colleagues' (2013) finding that if individuals do not learn from their failures, their future performance could possibly exacerbate. A coefficient plot of the eight decomposed variables is shown in Figure 4. Overall, these results suggest that contestants put more weight on information from higher-status peers, regardless of whether the information is positive (unsuccessful hacks from higher-status peers and successful hacks to higher-status peers) or negative (successful hacks from higher-status peers and unsuccessful hacks to higher-status peers). As a result, focal individuals are more likely to accept the information from higher-status peers, and such information is more motivating or more likely to drive these individuals to exert effort to learn.

3.5.3. Alternative Explanations

I consider two alternative explanations for my findings that individuals learn more from failures that are identified by higher-status peers than failures that are identified by lower-status peers. First, hacks from higher-status peers are easier to understand by focal individuals than hacks from lower-status peers, such that individuals learn less from hacks from lower-status peers because it is more difficult for them understand the issues identified by these hacks. Second, hacks from higher-status peers are of higher quality. In other words, hacks from higher-status peers identify more serious issues in the focal individuals' code, such that these hacks are more useful in improving focal individuals' coding skills.

To rule out these two explanations (i.e., understandability and quality), I conducted a survey to investigate whether the understandability and quality of hacks from higher-status and lower-status peers are different. I detail the survey procedures and findings in Appendix 2. In summary, the survey results suggest that hacks from higher-status and lower-status peers do not differ significantly in their understandability or quality ($t < 0.54$, $p > 0.58$). That is to say, the different effects of failures identified by higher-status peers and lower-status peers are driven by the status of the peers rather than the understandability or the quality of hacks from peers.

3.5.4. Robustness Checks

First, because *Prior success feedback given by machine* and *Prior failure feedback given by machine* are highly correlated ($r = 0.94$), a VIF (variance inflation factor) test suggests that the VIF for these two variables are above the threshold of 10 (VIF = 10.7 for *Prior success* and VIF = 10.2 for *Prior failure feedback given by machine*) that is often used a rule of thumb, which means including both of them in my estimations may lead to collinearity issues. Therefore, I did not include *Prior success feedback given by machine* in my estimations. If I add this variable in my estimations, my results still hold. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant (β = 2.836, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant (β = 0.034, p = 0.791). The Wald test for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continues to yield support to my hypothesis ($p < 0.001$).

Second, I included only solo contestants in my current estimations. However, a small number of contests allow teams (a team is composed of two or more contestants) to participate. Excluding the teams in my estimation sample may raise a concern of sampling bias. I address this concern by including the teams in my estimation sample, which increases the estimation sample size by 1.7%. My results still hold after including the teams in the estimation sample. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant (β = 2.701, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant (β = 0.063, p = 0.628). The Wald test for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continues to yield support to my hypothesis ($p < 0.001$).

Third, in current estimations I calculated all the variables using all the data before the focal contest (meaning I used no “window” or a cutoff point). I conducted a robustness check by calculating all the variables using 5-year and 3-year moving windows. My results still hold if I calculate the variables using 5-year moving window. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant ($\beta = 2.615$, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant ($\beta = 0.011$, $p = 0.933$). Similarly, my results still hold if I calculate the variables using 3-year moving window. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant ($\beta = 2.303$, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant ($\beta = 0.039$, $p = 0.744$). The Wald tests for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continue to yield support to my hypothesis ($p < 0.001$).

Fourth, in current operationalization I categorized peers with equal or higher status peers into one group, which I call peers with higher status. I tried to recategorize peers with equal status into another group, which is peers with lower status, such that higher status peers only include those peers whose status is above focal contestant. My results hold after this recategorization. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant ($\beta = 2.602$, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant ($\beta = 0.082$, $p = 0.518$). The Wald test for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continues to yield support to my hypothesis ($p < 0.001$).

Fifth, in most of the contests the hacks happened only during contests and within each room, where the contestants in the room are randomly assigned, however, 6.7% of contests in my sample are open-hack contests, where the hacks happened after the contests and all the contestants are given either 12 or 24 hours to hack any contestants (including the contestants outside of their own rooms). My results still hold if I do not to count the open hacks into my independent variables. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant ($\beta = 2.595$, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant ($\beta = -0.011$, $p = 0.934$). The Wald test for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continues to yield support to my hypothesis ($p < 0.001$).

Sixth, because *Status* and *Past performance* are highly correlated ($r = 0.79$), including both of them in my estimations may lead to collinearity issue. I checked the VIF for these two variables and the VIF for them is below 3.3. Even though the VIF figures do not indicate a collinearity problem, I removed *Past performance* from my estimations. I keep *Status* in the model because my independent variables were calculated based on the comparison of peers' status and focal contestant's status. My results still hold after removing *Past performance*. The coefficient of *Prior failure feedback given by higher-status peers* remains positive and significant

($\beta = 2.771$, $p < 0.001$) and the coefficient of *Prior failure feedback given by lower-status peers* remains nonsignificant ($\beta = 0.087$, $p = 0.501$). The Wald test for the difference of coefficients between *Prior failure feedback given by higher-status peers* and *Prior failure feedback given by lower-status peers* continues to yield support to my hypothesis ($p < 0.001$).

3.6. Discussion

I investigate whether status drives people to learn from failure feedback given peers. I find that whereas individuals learn from failure feedback given by higher-status peers, they learn less (in fact, in most cases not at all) from failure feedback given by lower-status peers. In the rest of this section, I discuss the contributions of my study and the implications of my findings to organizations and individuals.

By showing that failure feedback given by higher-status peers are more likely to trigger learning than failure feedback given by lower-status peers, I demonstrate how the characteristics of the sources of feedback play a crucial role in driving learning from failure. My findings suggest that higher-status peers draw more attention and engagement from individuals. Individuals are more reactive to feedback given by higher-status peers, even if it is through negative feedback. In addition, my findings about the different effects between higher-status peers and lower-status peers in driving learning form failure also add to the broader literature on peer effects (e.g., Hasan & Bagde, 2015; Hasan & Koning, 2019).

These arguments can be expanded to understand other aspects of learning from failure when mediated by external parties. The characteristics of those external parties, in contrast with those of the focal individual, might influence the amount of credits that the sources of the feedback get. For example, a high-status individual, in my context, will possibly struggle to learn from her own failure and will be facing decreasing returns to learning from failure, as there will be few higher status individuals to provide feedback to her. Inversely, a low-status individual will consider a wider set of sources of feedback, as there will be more higher-status individuals to consider. This suggests that the performance, or the variation in performance, of a crowd would progressively homogenize. In these cases, a third-party can be voluntarily, as an intervention, placed in a position that is above the learning individuals and mediate the learning process (Dahlin et al., 2018), on the condition that it is given credibility and social acceptance.

Even though previous research has investigated positive externalities of social capital and networks (e.g., Burt, 2007; Clement et al., 2018; Galunic et al., 2012), similar research on the externalities of status is lagging. Most work on status has focused on the implications of status for the person whose status is being investigated (e.g., Bowers & Prato 2018; Ertug & Castellucci, 2013; Kim & King, 2014; Simcoe & Waguespack, 2011), or for exchange partners (Castellucci & Ertug, 2010; Flynn et al., 2006; Prato & Ferraro, 2018; Reschke et al., 2018), but not investigated its externalities. The externalities of social capital have been discussed by Adler and Kwon (2002), and the implications for the externalities of status are likewise important. I suggest that even if higher-status peers and the focal individual are not tied to each other (are not network contacts in the standard sense, e.g., Rogan, 2014; Rogan & Mors, 2014) or exchange partners, the failures identified by higher-status peers are more likely to be accepted by the focal individual, which is more likely to motivate the focal individual to learn from her failures and improve her performance subsequently.

A popular human resource practice in organizations is the provision of 360-degree feedback, in which employees receive confidential, anonymous feedback from managers, peers, and direct reports (Brett & Atwater, 2001). However, a debate remains regarding whether the feedback should be kept anonymous (Ghorpade, 2000). As employees are likely to receive some negative feedback from colleagues, and negative feedback usually dents employees' self-esteem, triggers their defensive reactions, and thus stifles their learning from their own failures. in order to promote employees' learning from their own failures, organizations can selectively disclose the feedback giver's status (i) if the feedback is negative and (ii) if the feedback givers' status is higher than that of the receiver.

In terms of the implications of my findings for individuals, recent research has shown convincingly that status biases our evaluations (e.g., Kovács & Sharkey, 2014; Sharkey & Kovács, 2017). My findings suggest that individuals are biased towards feedback givers of different status as well. More specifically, individuals selectively process the information based on the status of feedback givers. This bias could hinder individuals' learning from failures because they are more defensive to negative feedback given by lower-status feedback givers. As failure offers valuable opportunities for learning, individuals should seize these opportunities to investigate the causes of failure and correct problematic routines or practices, regardless of the status of feedback givers.

3.7. Conclusion

I investigate whether peers' status can drive focal individuals to learn from their failures. By analyzing data collected from an online community for programming contest, my findings reveal that even though people might generally react defensively when facing failures and thus disengage from the failures, people pay more attention to and are more engaged with failure feedback given by higher-status peers than failure feedback given by lower-status peers. By showing that status is a driver of learning from failure, I contribute to experiential learning theories by incorporating status theory.

3.8. References

Adler, P. S., & Kwon, S. W. 2002. Social capital: Prospects for a new concept. ***Academy of Management Review***, 27(1): 17-40.

Azoulay, P., Stuart, T., & Wang, Y. 2014. Matthew: Effect or fable? ***Management Science***, 60(1): 92-109.

Bennett, V. M., & Snyder, J. 2017. The empirics of learning from failure. ***Strategy Science***, 2(1): 1-12.

Berger, J., Cohen, B. P., & Zelditch Jr, M. 1972. Status characteristics and social interaction. ***American Sociological Review***, 37(3): 241-255.

Bowers, A., & Prato, M. 2018. The structural origins of unearned status: How arbitrary changes in categories affect status position and market impact. ***Administrative Science Quarterly***, 63(3): 668-699.

Brett, J. F., & Atwater, L. E. 2001. 360° feedback: Accuracy, reactions, and perceptions of usefulness. ***Journal of Applied Psychology***, 86(5): 930-942.

Brown, J. D., & Dutton, K. A. 1995. The thrill of victory, the complexity of defeat: Self-esteem and people's emotional reactions to success and failure. ***Journal of Personality and Social Psychology***, 68(4): 712-722.

Burt, R. S. 2007. Secondhand brokerage: Evidence on the importance of local structure for managers, bankers, and analysts. ***Academy of Management Journal***, 50(1): 119-148.

Cannon, M. D., & Edmondson, A. C. 2001. Confronting failure: Antecedents and consequences of shared beliefs about failure in organizational work groups. Journal of Organizational Behavior: ***The International Journal of Industrial, Occupational and Organizational Psychology and Behavior***, 22(2): 161-177.

Castellucci, F., & Ertug, G. 2010. What's in it for them? Advantages of higher-status partners in exchange relationships. ***Academy of Management Journal***, 53(1): 149-166.

Clement, J., Shipilov, A., & Galunic, C. 2018. Brokerage as a public good: The externalities of network hubs for different formal roles in creative organizations. ***Administrative Science Quarterly***, 63(2): 251-286.

Dahlin, K. B., Chuang, Y. T., & Roulet, T. J. 2018. Opportunity, motivation, and ability to learn from failures and errors: Review, synthesis, and ways to move forward. ***Academy of Management Annals***, 12(1): 252-277.

Ellis, S., & Davidi, I. 2005. After-event reviews: Drawing lessons from successful and failed experience. ***Journal of Applied Psychology***, 90(5): 857-871.

Ellis, S., Mendel, R., & Nir, M. 2006. Learning from successful and failed experience: The moderating role of kind of after-event review. ***Journal of Applied Psychology***, 91(3): 669-680.

Ertug, G., & Castellucci, F. 2013. Getting what you need: How reputation and status affect team performance, hiring, and salaries in the NBA. ***Academy of Management Journal***, 56(2): 407-431.

Eskreis-Winkler, L., & Fishbach, A. 2019. Not Learning From Failure—the Greatest Failure of All. ***Psychological Science***, 30(12): 1733-1744.

Flynn, F. J., Reagans, R. E., Amanatullah, E. T., & Ames, D. R. 2006. Helping one's way to the top: self-monitors achieve status by helping others and knowing who helps whom. ***Journal of Personality and Social Psychology***, 91(6): 1123-1137.

Galunic, C., Ertug, G., & Gargiulo, M. 2012. The positive externalities of social capital: Benefiting from senior brokers. ***Academy of Management Journal***, 55(5): 1213-1231.

Ghorpade, J. 2000. Managing five paradoxes of 360-degree feedback. ***Academy of Management Perspectives***, 14(1): 140-150.

Halperin, K., Snyder, C. R., Shenkel, R. J., & Houston, B. K. 1976. Effects of source status and message favorability on acceptance of personality feedback. ***Journal of Applied Psychology***, 61(1): 85-88.

Hasan, S., & Bagde, S. 2015. Peers and network growth: Evidence from a natural experiment. ***Management Science***, 61(10): 2536-2547.

Hasan, S., & Koning, R. 2019. Prior ties and the limits of peer effects on startup team performance. ***Strategic Management Journal***, 40(9): 1394-1416.

Ilgen, D., & Davis, C. 2000. Bearing bad news: Reactions to negative performance feedback. ***Applied Psychology***, 49(3): 550-565.

KC, D., Staats, B. R., & Gino, F. 2013. Learning from my success and from others' failure: Evidence from minimally invasive cardiac surgery. ***Management Science***, 59(11): 2435-2449.

Kim, D. 1993. The link between individual and organizational learning. ***Sloan Management Review***, 35(1): 37–50.

Kim, J. W., & King, B. G. 2014. Seeing stars: Matthew effects and status bias in major league baseball umpiring. ***Management Science***, 60(11): 2619-2644.

Kolb, D. A. 1984. ***Experiential learning: Experience as the source of learning and development***. Englewood Cliffs, NJ: Prentice-Hall.

Kovács, B., & Sharkey, A. J. 2014. The paradox of publicity: How awards can negatively affect the evaluation of quality. ***Administrative Science Quarterly***, 59(1): 1-33.

Magee, J. C., & Galinsky, A. D. 2008. Social hierarchy: The self-reinforcing nature of power and status. ***Academy of Management Annals***, 2(1): 351-398.

Piazza, A., & Castellucci, F. 2014. Status in organization and management theory. ***Journal of Management***, 40(1): 287-315.

Prato, M., & Ferraro, F. 2018. Starstruck: How hiring high-status employees affects incumbents' performance. ***Organization Science***, 29(5): 755-774.

Reschke, B. P., Azoulay, P., & Stuart, T. E. 2018. Status spillovers: The effect of status-conferring prizes on the allocation of attention. ***Administrative Science Quarterly***, 63(4): 819-847.

Rogan, M. 2014. Executive departures without client losses: The role of multiplex ties in exchange partner retention. ***Academy of Management Journal***, 57(2): 563-584.

Rogan, M., & Mors, M. L. 2014. A network perspective on individual-level ambidexterity in organizations. ***Organization Science***, 25(6): 1860-1877.

Sauder, M., Lynn, F., & Podolny, J. M. 2012. Status: Insights from organizational sociology. ***Annual Review of Sociology***, 38: 267-283.

Schneider, D. J., & Turkat, D. 1975. Self-presentation following success or failure: Defensive self-esteem models. ***Journal of Personality***, 43(1):127-35.

Sharkey, A. J., & Kovács, B. 2018. The many gifts of status: How attending to audience reactions drives the use of status. ***Management Science***, 64(11): 5422-5443.

Shepherd, D. A., Patzelt, H., & Wolfe, M. 2011. Moving forward from project failure: Negative emotions, affective commitment, and learning from the experience. ***Academy of Management Journal***, 54(6): 1229-1259.

Simcoe, T. S., & Waguespack, D. M. 2011. Status, quality, and attention: What's in a (missing) name? ***Management Science***, 57(2): 274-290.

Sitkin, S. B. 1992. Learning through failure: The strategy of small losses. ***Research in Organizational Behavior***, 14: 231-266.

Taggar, S., & Brown, T. C. 2006. Interpersonal affect and peer rating bias in teams. ***Small Group Research***, 37(1), 86-111.

Taylor, S. E. 1991. Asymmetrical effects of positive and negative events: the mobilization-minimization hypothesis. ***Psychological Bulletin***, 110(1): 67-85.

Ucbasaran, D., Shepherd, D. A., Lockett, A., & Lyon, S. J. 2013. Life after business failure: The process and consequences of business failure for entrepreneurs. ***Journal of Management***, 39(1): 163-202.

Wilhelm, H., Richter, A. W., & Semrau, T. 2019. Employee Learning from Failure: A Team-as-Resource Perspective. ***Organization Science***, 30(4): 694-714.

3.9. Tables

Table 1. Descriptive statistics and correlations (N = 1,474,753).

		Mean	S.D.	1	2	3	4	5	6	7	8	9	10	11	12	13
1	Performance	50.25	28.55													
2	Prior failure feedback given by higher-status peers	0.95	0.85	0.21												
3	Prior failure feedback given by lower-status peers	0.38	0.59	0.31	0.52											
4	Prior failure feedback given by machine	4.64	1.48	0.26	0.70	0.54										
5	Prior unsuccessful hacks from peers	1.01	0.99	0.28	0.72	0.62	0.67									
6	Prior successful hacks to peers	0.62	1.13	0.28	0.51	0.59	0.51	0.58								
7	Prior unsuccessful hacks to peers	0.71	1.12	0.25	0.55	0.56	0.54	0.61	0.85							
8	Past performance	45.89	20.96	0.58	0.23	0.44	0.31	0.38	0.40	0.35						
9	Status	14.84	2.87	0.49	0.23	0.54	0.31	0.43	0.50	0.43	0.79					
10	Knowledge specialization	0.58	0.21	-0.11	-0.06	-0.10	-0.12	-0.10	-0.09	-0.08	-0.18	-0.18				
11	Number of successful hacks received by focal contestant in focal contest	0.09	0.24	-0.08	0.02	0.00	0.02	0.01	0.00	0.00	-0.01	-0.01	0.04			
12	Number of unsuccessful hacks received by focal contestant in focal contest	0.07	0.23	0.05	0.03	0.03	0.03	0.04	0.03	0.04	0.04	0.04	0.01	0.17		
13	Number of successful hacks sent by focal contestant in focal contest	0.06	0.28	0.19	0.10	0.13	0.11	0.13	0.26	0.23	0.13	0.14	-0.01	0.05	0.05	
14	Number of unsuccessful hacks sent by focal contestant in focal contest	0.05	0.24	0.09	0.08	0.09	0.09	0.10	0.22	0.23	0.09	0.09	-0.01	0.07	0.08	0.49

Table 2. OLS fixed-effects models predicting *Performance*

	(1)	(2)	(3)	(4)	(5)
Prior failure feedback given by machine	5.454***	4.875***	4.847***	5.454***	4.846***
	(0.073)	(0.070)	(0.069)	(0.074)	(0.070)
Prior unsuccessful hacks from peers	2.286***	1.549***	1.472***	2.284***	1.467***
	(0.068)	(0.068)	(0.067)	(0.070)	(0.067)
Prior successful hacks to peers	0.793***	0.746***	0.796***	0.791***	0.793***
	(0.078)	(0.077)	(0.076)	(0.076)	(0.075)
Prior unsuccessful hacks to peers	0.695***	0.447***	0.407***	0.694***	0.406***
	(0.083)	(0.082)	(0.082)	(0.083)	(0.082)
Past performance	0.049***	0.044***	0.042***	0.049***	0.042***
	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
Status	0.702***	0.707***	0.742***	0.701***	0.741***
	(0.046)	(0.046)	(0.045)	(0.045)	(0.044)
Knowledge specialization	0.489	0.441	0.472	0.489	0.471
	(0.621)	(0.618)	(0.616)	(0.621)	(0.616)
Number of successful hacks received by focal contestant in focal contest	-10.910***	-10.405***	-10.372***	-10.909***	-10.370***
	(0.413)	(0.416)	(0.416)	(0.413)	(0.415)
Number of unsuccessful hacks received by focal contestant in focal contest	4.245***	4.051***	4.032***	4.245***	4.031***
	(0.210)	(0.210)	(0.210)	(0.210)	(0.210)
Number of successful hacks sent by focal contestant in focal contest	8.318***	8.308***	8.311***	8.318***	8.311***

	(0.231)	(0.231)	(0.231)	(0.231)	(0.231)
Number of unsuccessful hacks sent by focal contestant in focal contest	-2.537***	-2.579***	-2.590***	-2.536***	-2.590***
	(0.202)	(0.202)	(0.201)	(0.202)	(0.201)
Prior failure feedback given by peers		2.387***			
		(0.097)			
Prior failure feedback given by higher-status peers			2.835***		2.836***
			(0.107)		(0.107)
Prior failure feedback given by lower-status peers				0.015	0.034
				(0.129)	(0.128)
Constant	9.130***	10.265***	9.944***	9.137***	9.960***
	(0.785)	(0.783)	(0.783)	(0.786)	(0.783)
N	1,474,753	1,474,753	1,474,753	1,474,753	1,474,753
R^2	0.617	0.617	0.617	0.617	0.617

Robust standard errors are in parentheses. All the models include contest, contestant, and room fixed effects. Standard errors are clustered at contest, contestant, and room level. Two-tailed tests. *** $p < 0.001$.

Table 3. Interaction between *Prior failure feedback given by machine* and *Prior failure feedback given by peers*

	(1)	(2)	(3)	(4)	(5)	(6)
Prior failure feedback given by machine	5.454***	4.875***	4.846***	4.887***	4.929***	4.913***
	(0.073)	(0.070)	(0.070)	(0.071)	(0.070)	(0.069)
Prior unsuccessful hacks from peers	2.286***	1.549***	1.467***	1.160***	1.204***	1.446***
	(0.068)	(0.068)	(0.067)	(0.068)	(0.068)	(0.068)
Prior successful hacks to peers	0.793***	0.746***	0.793***	0.471***	0.602***	0.730***
	(0.078)	(0.077)	(0.075)	(0.077)	(0.075)	(0.075)
Prior unsuccessful hacks to peers	0.695***	0.447***	0.406***	0.139	0.131	0.366***
	(0.083)	(0.082)	(0.082)	(0.083)	(0.082)	(0.082)
Past performance	0.049***	0.044***	0.042***	0.044***	0.039***	0.043***
	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
Status	0.702***	0.707***	0.741***	0.536***	0.603***	0.714***
	(0.046)	(0.046)	(0.044)	(0.046)	(0.045)	(0.044)
Knowledge specialization	0.489	0.441	0.471	0.420	0.489	0.467
	(0.621)	(0.618)	(0.616)	(0.618)	(0.613)	(0.618)
Number of successful hacks received by focal contestant in focal contest	-10.910***	-10.405***	-10.370***	-10.504***	-10.548***	-10.410***
	(0.413)	(0.416)	(0.415)	(0.416)	(0.415)	(0.416)
Number of unsuccessful hacks received by focal contestant in focal contest	4.245***	4.051***	4.031***	3.930***	3.948***	4.016***
	(0.210)	(0.210)	(0.210)	(0.210)	(0.209)	(0.210)
Number of successful hacks sent by focal contestant in focal contest	8.318***	8.308***	8.311***	8.260***	8.270***	8.300***
	(0.231)	(0.231)	(0.231)	(0.231)	(0.231)	(0.231)
Number of unsuccessful hacks sent by focal contestant in	-2.537***	-2.579***	-2.590***	-2.597***	-2.613***	-2.583***

focal contest						
	(0.202)	(0.202)	(0.201)	(0.202)	(0.201)	(0.201)
Prior failure feedback given by peers		2.387***		-3.057***		
		(0.097)		(0.255)		
Prior failure feedback given by higher-status peers			2.836***		-3.600***	2.804***
			(0.107)		(0.267)	(0.108)
Prior failure feedback given by lower-status peers			0.034		-0.708***	-3.259***
			(0.128)		(0.128)	(0.467)
Prior failure feedback given by machine * Prior failure feedback given by peers				1.060***		
				(0.048)		
Prior failure feedback given by machine * Prior failure feedback given by higher-status peers					1.223***	
					(0.052)	
Prior failure feedback given by machine * Prior failure feedback given by lower-status peers						0.544***
						(0.081)
Constant	9.130***	10.265***	9.960***	13.103***	12.269***	10.148***
	(0.785)	(0.783)	(0.783)	(0.804)	(0.802)	(0.784)
N	1,474,753	1,474,753	1,474,753	1,474,753	1,474,753	1,474,753
R^2	0.617	0.617	0.617	0.618	0.618	0.618

Robust standard errors are in parentheses. All the models include contest, contestant, and room fixed effects. Standard errors are clustered at contest, contestant, and room level. Two-tailed tests. *** $p < 0.001$.

Table 4. OLS fixed-effects models predicting *Probability of success*

	(1)	(2)	(3)	(4)	(5)
Prior failure feedback given by machine	0.0447***	0.0396***	0.0396***	0.0446***	0.0396***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Prior unsuccessful hacks from peers	0.0152***	0.0076***	0.0071***	0.0148***	0.0068***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Prior successful hacks to peers	0.0084***	0.0078***	0.0082***	0.0081***	0.0080***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Prior unsuccessful hacks to peers	0.0045***	0.0020*	0.0017	0.0044***	0.0016
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
Status	-0.0003	-0.0003	-0.0001	-0.0004	-0.0001
	(0.001)	(0.001)	(0.001)	(0.001)	(0.000)
Specialization	0.1288***	0.1205***	0.1200***	0.1284***	0.1197***
	(0.007)	(0.007)	(0.007)	(0.007)	(0.007)
Prior failure feedback given by peers		0.0231***			
		(0.001)			
Prior failure feedback given by higher-status peers			0.0265***		0.0265***
			(0.002)		(0.002)
Prior failure feedback given by lower-status peers				0.0029*	0.0021
				(0.001)	(0.001)
Constant	0.2074***	0.2170***	0.2133***	0.2088***	0.2143***
	(0.010)	(0.010)	(0.010)	(0.009)	(0.009)
N	6,237,859	6,237,859	6,237,859	6,237,859	6,237,859
R^2	0.324	0.324	0.324	0.324	0.324

Robust standard errors are in parentheses. All the models include contest, contestant, room, and problem fixed effects. Standard errors are clustered at contest, contestant, room, problem level. Two-tailed tests. * $p < 0.05$, *** $p < 0.001$.

Table 5. Summary of results if status is measured by tier or color

Independent variables	Status is measured by tier		Status is measured by color	
	β	***p***	***β***	***p***
Prior failure feedback given by higher-status peers	2.710	< 0.001	2.719	< 0.001
Prior failure feedback given by same-status peers	0.656	< 0.001	0.693	< 0.001
Prior failure feedback given by lower-status peers	-0.077	0.611	-0.090	0.553

Note: The coefficients reported in Table 5 are estimated based on models that have all of the control variables and fixed effects as in the models that are reported in Table 2.

Table 6. Analyses of unsuccessful hacks from peers and hacks to peers

	(1)	(2)	(3)	(4)	(5)	(6)
Past performance	0.027***	0.025***	0.026***	0.026***	0.027***	0.024***
	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)	(0.005)
Status	0.677***	0.710***	0.686***	0.703***	0.700***	0.732***
	(0.046)	(0.044)	(0.045)	(0.045)	(0.045)	(0.043)
Knowledge specialization	0.708	0.729	0.723	0.737	0.725	0.771
	(0.618)	(0.616)	(0.616)	(0.616)	(0.616)	(0.613)
Number of successful hacks received by focal contestant in focal contest	-10.464***	-10.418***	-10.494***	-10.473***	-10.474***	-10.461***
	(0.416)	(0.415)	(0.416)	(0.416)	(0.416)	(0.415)
Number of unsuccessful hacks received by focal contestant in focal contest	4.020***	3.997***	4.089***	4.021***	4.017***	4.052***
	(0.210)	(0.210)	(0.210)	(0.210)	(0.211)	(0.210)
Number of successful hacks sent by focal contestant in focal contest	8.303***	8.305***	8.300***	8.318***	8.312***	8.318***
	(0.231)	(0.231)	(0.231)	(0.230)	(0.231)	(0.230)
Number of unsuccessful hacks sent by focal contestant in focal contest	-2.582***	-2.593***	-2.588***	-2.589***	-2.577***	-2.607***
	(0.202)	(0.202)	(0.202)	(0.202)	(0.203)	(0.202)
Prior failure feedback given by machine	2.477***	2.478***	2.493***	2.481***	2.477***	2.502***
	(0.118)	(0.118)	(0.118)	(0.118)	(0.118)	(0.118)
Prior failure feedback given by peers	2.111***		1.919***	2.055***	2.060***	
	(0.097)		(0.095)	(0.096)	(0.096)	
Prior unsuccessful hacks from peers	1.435***	1.343***		1.431***	1.414***	
	(0.068)	(0.067)		(0.068)	(0.067)	
Prior successful hacks to peers	0.772***	0.814***	0.781***		0.871***	
	(0.077)	(0.075)	(0.076)		(0.078)	
Prior unsuccessful hacks to peers	0.429***	0.385***	0.367***	0.346***		
	(0.082)	(0.082)	(0.082)	(0.081)		
Prior failure feedback given by higher-status peers		2.594***				2.280***
		(0.106)				(0.100)
Prior failure feedback given by lower-status peers		0.019				0.104

		(0.128)				(0.120)
Prior unsuccessful hacks from higher-status peers			1.881***			1.595***
			(0.080)			(0.077)
Prior unsuccessful hacks from lower-status peers			0.384***			0.563***
			(0.091)			(0.083)
Prior successful hacks to higher-status peers				1.576***		1.223***
				(0.111)		(0.111)
Prior successful hacks to lower-status peers				-0.047		0.284**
				(0.096)		(0.093)
Prior unsuccessful hacks to higher-status peers					1.029***	0.555***
					(0.099)	(0.098)
Prior unsuccessful hacks to lower-status peers					-0.405***	-0.274**
					(0.097)	(0.095)
Constant	11.032***	10.766***	10.905***	10.651***	10.734***	10.321***
	(0.783)	(0.782)	(0.779)	(0.777)	(0.779)	(0.777)
N	1,474,753	1,474,753	1,474,753	1,474,753	1,474,753	1,474,753
R^2	0.618	0.618	0.618	0.618	0.618	0.618

Standard errors are in parentheses. All the models include contest, contestant, and room fixed effects. Standard errors are clustered at contest, contestant, and room level. Two-tailed tests. *** $p < 0.001$.

3.10. Figures

Figure 1. Description of a problem in a CPC contest

B. Spreadsheets

time limit per test: 10 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

In the popular spreadsheets systems (for example, in Excel) the following numeration of columns is used. The first column has number A, the second — number B, etc. till column 26 that is marked by Z. Then there are two-letter numbers: column 27 has number AA, 28 — AB, column 52 is marked by AZ. After ZZ there follow three-letter numbers, etc.

The rows are marked by integer numbers starting with 1. The cell name is the concatenation of the column and the row numbers. For example, BC23 is the name for the cell that is in column 55, row 23.

Sometimes another numeration system is used: RXCY, where X and Y are integer numbers, showing the column and the row numbers respectfully. For instance, R23C55 is the cell from the previous example.

Your task is to write a program that reads the given sequence of cell coordinates and produce each item written according to the rules of another numeration system.

Input

The first line of the input contains integer number n ($1 \le n \le 10^5$), the number of coordinates in the test. Then there follow n lines, each of them contains coordinates. All the coordinates are correct, there are no cells with the column and/or the row numbers larger than 10^6 .

Output

Write n lines, each line should contain a cell coordinates in the other numeration system.

Examples

input (Copy)

```
2
R23C55
BC23
```

output (Copy)

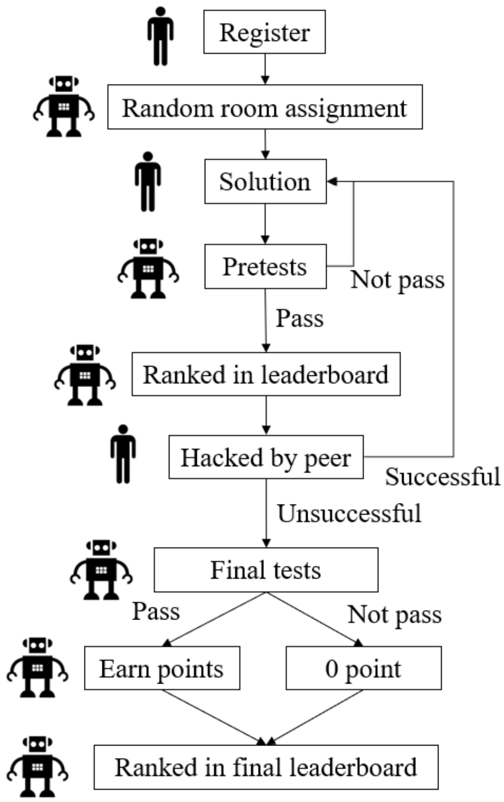
```
BC23
R23C55
```

Figure 2. Example of problem solution written in Python (for the problem in Figure 1)

→ **Source**

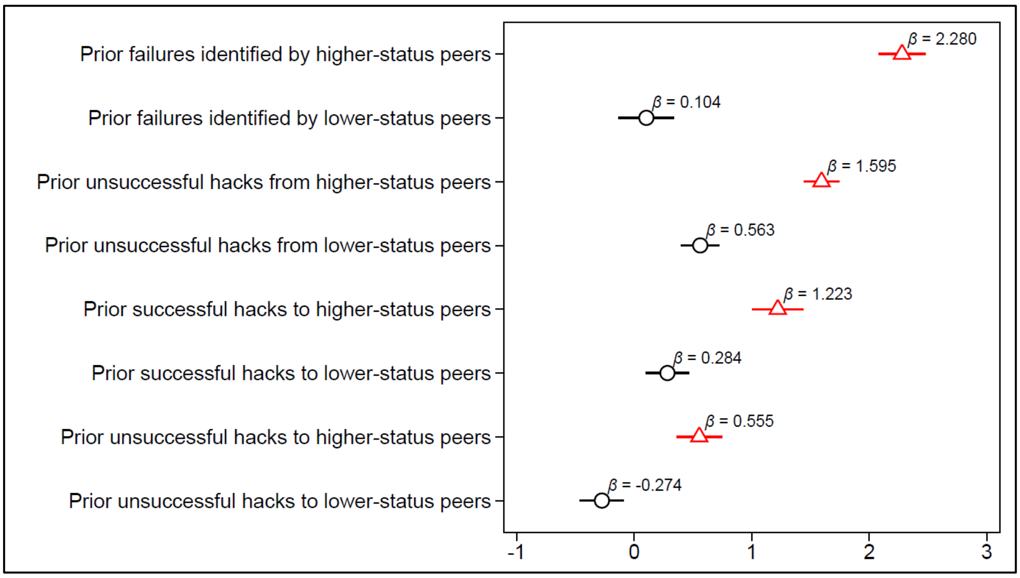
```python
for n in range(int(input())):
    x = input();a=b=0
    for c in x:
        if '0' <= c<='9':b=10*b+int(c)
        elif b:
            a,b=x[1:].split('C');b=int(b);v=""
            while b: b-=1;v=chr(65+b%26)+v;b//=26
            print(v+a);break
        else:a=26*a+ord(c)-64
    else:print("R%dC%d" % (b, a))
```

Figure 3. An illustration of the process of a contest



Note: A human icon on the left of the box indicates that the step is completed by a contestant. A robot icon on the left of the box indicates that the step is performed by the system.

Figure 4. Coefficient plot for the variables of interest in Model 6 of Table 6



Note: Variables related to higher-status peers are plotted using red triangle marker and variables related to lower-status peers are plotted using black circle marker. The marker labels indicate the regression coefficients for these variables.

3.11. Appendix 1

Table A1. Distribution of solutions for different participation types.

Place	Participation type	Percent of solutions	Percent of successful solutions	Percent of unsuccessful solutions
Contest	Formal contest	25.6%	35.5%	64.5%
	Practice	67.3%	40.3%	59.7%
	Virtual contest	5.7%	40.5%	59.5%
	Unqualified participation	1.4%	40.5%	59.5%
Gym	Formal contest	6.0%	33.1%	66.9%
	Practice	37.1%	31.6%	68.4%
	Unqualified participation	56.9%	33.8%	66.2%

3.12. Appendix 2

Survey questions

Understandability (Lee et al., 2002):

Question: Please rate if it is easy to understand what code problem that this hack is pointing to (Scale 1-7, 1 is the hardest to understand and 7 is the easiest to understand). (Your rating is __).

Quality (Li et al., 2010; Steelman et al., 2004):

Question 1: Please rate if this hack is useful in helping the hack receiver to improve his/her coding skills (Scale 1-7, 1 is the least useful and 7 is the most useful). (Your rating is: __).

Question 2: Please rate how critical the code problem is that this hack is pointing to (Scale 1-7, 1 is the least critical and 7 is the most critical). (Your rating is: __).

Survey description

I follow Arts, Cassiman, and Gomez (2018) to invite experts to read the hack and corresponding code that was hacked and rate the understandability and quality of the hack. Experts are CPC members whose rating is above 1,800, which is top 8 percent in CPC. I randomly pick 10 contestants, and for each contestant I randomly pick one of her hacks from higher-status peers and one of her hacks from lower-status peers. An example of hack-code pair is shown below. All the hacks and the corresponding code are put in a document, where the order of the hack-code pairs is randomly decided. The experts are not informed that some hacks are from higher-status peers and some are from lower-status peers. They are only told to rate the understandability and quality of the hacks. I follow Arts, Cassiman, and Gomez (2018) to ask experts to use Likert scale from one to seven to rate their perceived understandability or quality.

A total of 16 experts took the survey, resulting in 160 ratings for hacks from higher-status peers and 160 ratings for hacks from lower-status peers. The survey results are shown in the Table 2elow. The *t* tests suggest that hacks from higher-status peers and lower-status peers are not significantly different in terms of understandability and quality.

Hack	**Variable**	***N***	**Mean**	**Std. Dev.**	**Min**	**Max**	***t*-test**	
							***t*-value**	***p*-value**
from higher-status peers	easy to understand the hack	160	5.56	1.45	1	7	0.54	0.589
from lower-status peers	easy to understand the hack	160	5.48	1.44	1	7		
from higher-status peers	useful for learning	160	5.06	1.52	1	7	0.37	0.714
from lower-status peers	useful for learning	160	4.99	1.52	1	7		
from higher-status peers	hack points out critical issue	160	4.90	1.80	1	7	0.42	0.677
from lower-status peers	hack points out critical issue	160	4.82	1.69	1	7		

Survey question used in survey

Understandability:

Question: Please rate if it is easy to understand what code problem that this hack is pointing to (Scale 1-7, 1 is the hardest to understand and 7 is the easiest to understand). (Your rating is __).

Quality:

Question 1: Please rate if this hack is useful in helping the hack receiver to improve his/her coding skills (Scale 1-7, 1 is the least useful and 7 is the most useful). (Your rating is: __).

Question 2: Please rate how critical the code problem is that this hack is pointing to (Scale 1-7, 1 is the least critical and 7 is the most critical). (Your rating is: __).

Hack:

```
5
1 3 4 6 12
```

Code:

```
#include <bits/stdc++.h>
#define INF 1e9 + 2

using namespace std;
typedef long long ll;

//ios::sync_with_stdio(0);
//cin.tie(0);

set<int> s;
bool used[1234567];
int a[12345];
int gcd(int a, int b)
{
	return ( !b ? a : gcd(b, a % b));
}
int main()
{
	int n;
	cin >> n;
	for(int i = 1; i <= n; i ++)
	{
		cin >> a[i];
		used[a[i]] = 1;
		s.insert(a[i]);
	}
	for(int i = 1; i <= n; i ++)
	for(int j = i; j <= n; j ++)
	{
		int k = gcd(a[i], a[j]);
		if(!used[k])
		{
			cout << -1;
			return 0;
		}
	}
	cout << s.size() << '\n';
	for(auto it : s)
	{
		cout << it << ' ';
	}
}
```

3.13. Appendix 3

Figure A1. Plot of the interaction between *Prior failure feedback given by machine* and *Prior failure feedback given by peers* in Model 4 of Table 3.

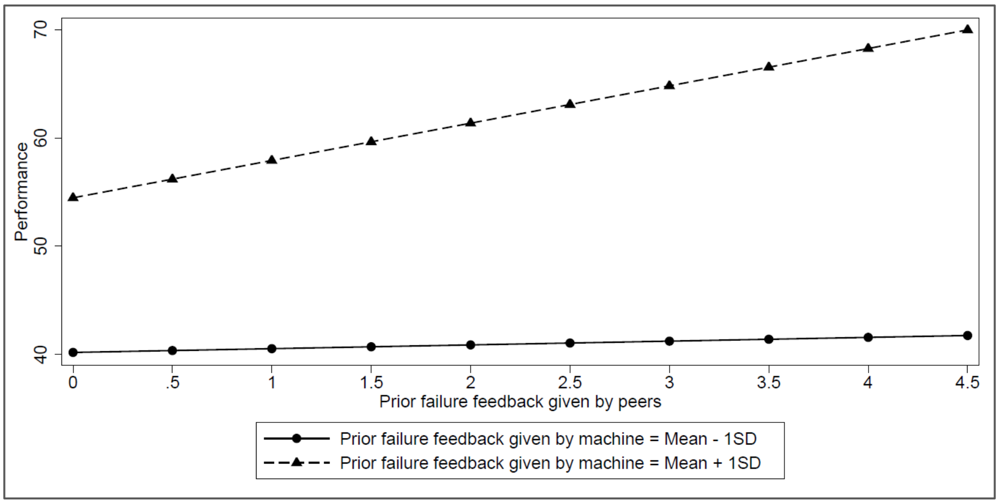
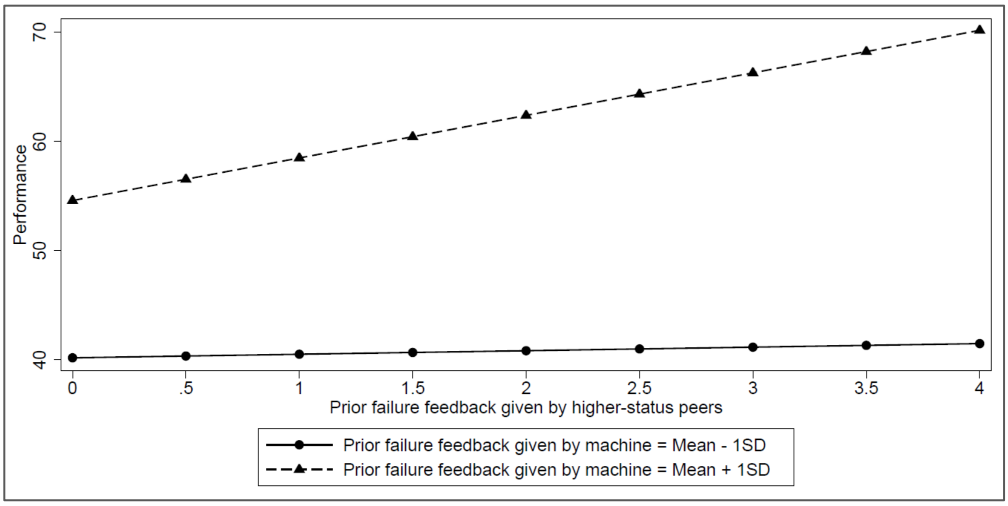


Figure A2. Plot of the interaction between *Prior failure feedback given by machine* and *Prior failure feedback given by higher-status peers* in Model 5 of Table 3.



<u>Figure A3. Plot of the interaction between *Prior failure feedback given by machine* and *Prior failure feedback given by lower-status peers* in Model 6 of Table 3.</u>