Exploring Engineers' Understanding of Uncertainty

Riya Aggarwal, Mira Flynn, Zachary del Rosario

Abstract

There are long-standing examples of how engineers' education in probability and statistics has not been sufficient. This work presents a novel theoretical framework to help teach and study statistical variability. Using this framework, we developed an interview guide and deductive coding scheme to use in interviews with engineering students. Early results from these interviews support our initial hypothesis of a slight induced variability bias.

Introduction

Every aircraft ever flown has been designed using probabilistically-flawed, potentially dangerous criteria (del Rosario et al. 2021). These long-standing criteria have been recognized to be inaccurate, speaking to the difficulties in teaching these concepts to engineers. More generally, statistics education has been brought into the spotlight as a vital but difficult discipline to teach well (Hogg 1991) leaving many engineers on shaky ground when it comes to processing uncertainties in data they face. Our team aimed to bridge this gap by analyzing how engineering students study uncertainty in data, with a focus on improving engineering practice and training.

Solving this problem required understanding our audience's educational background and developing a software tool. An education team worked to design and conduct interviews with engineering students to learn more about the mental models they used when considering uncertainty through data analysis, while a software team continued work on the software package py-grama, designed to support scientists and engineers learning to handle uncertainty. This report will focus mainly on the takeaways from the education sub-team.

Theoretical Framework

Our theoretical framework for this work consists of a two-axis, four-category framing to distinguish types of variability. (See Figure 1) This framework, developed by Prof. del Rosario (Aggarwal et al. 2021), is summarized in this section.

Axes: Cause (Chance-Assignable) and Source (Induced-Real)

Our framework is defined by two main axes, cause (*chance* or *assignable*) and source (*induced* or *real*).

The concepts of chance and assignable cause were formulated in the context of high-volume manufacturing (Shewhart 1931). Cause is a way to distinguish variation that can be attributed to something specific versus variation that cannot. While much unknown variation may be explained through careful analysis, there are many situations in which trying to control those causes of variation isn't possible or feasible. One way to think about the difference is that an assignable cause of variation is something that one could try to fix, while chance variation is a cause that is not possible or worthwhile to fix. For instance, effort spent attempting to find a cause for variability in manufacturing is effort not used to create useful products, in which case the remaining uninvestigated variability is treated as chance.

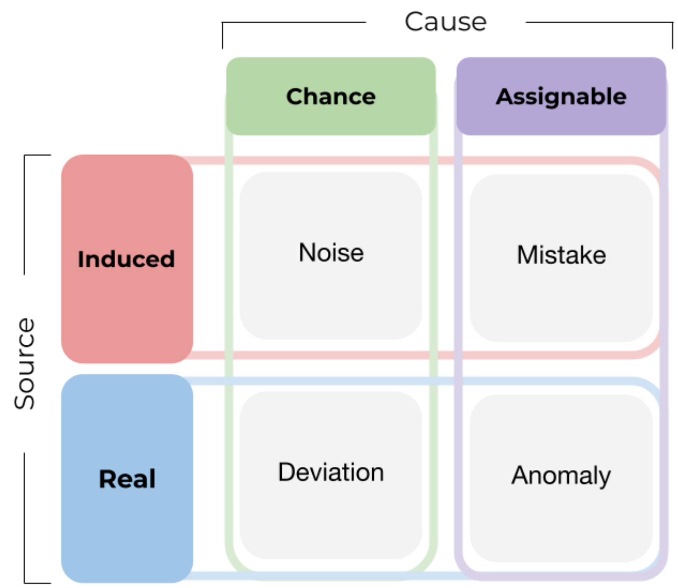

Figure 1: Diagram showing relationship between the four quadrants

Source is a way to distinguish where variation was introduced into a process. Wild and Pfannkuch (1999) separated variability into the concepts of *real* and *induced,* and we more specifically defined these terms for our work. In our framing, induced source variability is introduced after the item has been created, and real source variability is introduced within the process of the item's creation. For example, in a manufacturing scenario, an induced source of variability may be due to miscalibration of the strength measurement device, while a real source of variability may be due to impurities in the metal itself.

The overlap of these axes create cause-source quadrants, which organize variability into four distinct categories.

Four Quadrants: Noise, Deviation, Mistake, Anomaly

Overall, these four quadrants are: *noise* (induced and chance), *deviation* (real and chance), *mistake* (induced and assignable), and *anomaly* (real and assignable). These quadrants are illustrated through the following examples, one technical and one non-technical.

Let us say a company is manufacturing steel components, which they aim to use as structural components. For structural components to be used safely, the manufactured parts must be sufficiently strong. During the manufacturing process, there will be chance variation that causes random variations within the microstructure of the steel (deviation) as well as possible manufacturing slip-ups that cause incorrect proportions of materials to be mixed into the steel mixture (anomaly). After the steel has been cast, the company must test the tensile strength of the alloy to determine how strong it is. They may choose to perform a test with a machine such as an Instron. This could introduce chance variation through external electrical interference affecting the machine's output (noise) as well as assignable variations, such as an engineer forgetting to calibrate the machine prior (mistake).

Another example would be an ice cream shop, where the goal is to make each scoop taste the same as the last. While the ice cream is being made, it may vary in how it crystallizes naturally (deviation) and in the blending time the employee chose (anomaly). Once it is in its serving containers at the store, flavor variation can be introduced through trace cross-contamination from the common ice cream scoop (noise) or perhaps accidentally mislabeled tubs (mistake).

Method

Our education team focused on exploring how engineering students reasoned about uncertainty. Our software team worked on developing an open-source package designed to help engineers process uncertainty through probabilistic models.

Education

Interview Guide

Our education team's overall goal for the summer research period was to analyze how engineers process uncertainty, and we decided to begin by interviewing engineering students at Olin. All the education team members went through IRB human subjects training and then began designing an interview guide. The team decided to conduct interviews as opposed to a written examination because we felt that there would be a significant need for follow-up questions to truly understand the participant's uncertainty mindset. The interview guide was intended to provide the interview with structure, but we were also mindful of leaving space for the interviewer to explore the participant's answers with them.

The interview guide is split into two main parts. The first part is heavily structured, with defined scenarios and follow-ups and focused on tables and numerical data. The second half is much more open-ended and allows the participant to bring in their own context. With this split, we were able to gather data that was common across participants and also provide an opportunity for participants to explain their own experiences as an engineer that we may miss through our structure.

In the first part, questions were designed to progressively emphasize safety and catastrophic failure. Given prior studies on current aircraft design procedures (del Rosario et al. 2021), we hypothesized that without explicit prompting, interview participants would default to an induced style of analysis and reasoning. Conversely, we expected questions that emphasize safety to lead participants to analyze and reason more in terms of real variability. Additionally, we expected an overall larger fraction of induced than real thinking among participants.

Interviews

Over the summer period we conducted seven interviews with engineering students, selected from a pool of twenty-seven volunteers.

Through the interview process, we faced multiple constraints while trying to learn about a participant's uncertainty mental model. One major issue was timing. In our first few interviews, we found ourselves going over our scheduled time of 45 minutes because we spent too long on certain questions or continued to follow up even after we had gotten a sufficient answer since we weren't sure of what to look for. This improved after our pilot interviews, but we also built in additional structured follow-up questions into the interview guide to maintain a steady rhythm. Another roadblock was interviewee comfort. The questions consisted of scenarios many interviewees hadn't previously considered, and it led to both discomfort and a lack of confidence in their answers. We tackled this from multiple avenues. One of our first steps was to use selection criteria to ensure that any participants we interviewed had sufficient mechanical engineering background, so the scenarios we showed them would feel relatively comfortable prior to our addition of uncertainty. Our second fix was to add a warm-up section to the beginning of the interview. We started the interview by asking participants to list off words related to variability in engineering, warming them up to start chatting about these concepts while still giving them control of the context.

Qualitative Coding

Once we completed our interviews, we began qualitative coding on the interview transcripts. As we developed our coding scheme, we were looking to separate the participants' thoughts into *analysis* or *reasoning*. We defined analysis as the mathematical methods they used to analyze the data presented and reasoning as the causes the participant defined to explain the variation in said data. For reasoning, we would look if the possible sources the participant listed fell into induced or real.

For the second, more open-ended portion of our interview, we applied the theoretical framework described above. For each source of variation the participant listed, we coded the item as noise, deviation, mistake, or anomaly by looking at its source and cause.

As we developed the coding scheme, we realized that some participants had combined both real and induced reasoning into single answers. As a result, we had to code these lines as both induced and real. In future iterations of the interview guide, we will aim to lead participants to separate these aspects themselves during interviews to better allow us to code answers as induced or real independently.

After finalizing our scheme, we computed our inter-rater reliability, or IRR, by independently coding our interview transcripts and comparing our levels of agreement. The percent agreement for the Part 1 coding scheme was 87%, which is comfortably above the standard 80% IRR threshold for a valid coding scheme. The percent agreement in Part 2 was only 55%, indicating that further work is necessary on Part 2 of the interview protocol and coding scheme.

Software

This paper also covers two of the software projects completed throughout the summer. The first project was code coverage, a software metric used to see how much of the code written in a package is covered by corresponding tests. This industry-standard procedure ensures code quality by guaranteeing that a percentage of a code base is covered by a test. In this way, the team can ensure that any proposed code added to the package is tested for correctness prior to adding it to the main code branch on our GitHub repository. The best avenue for this was through implementing an automatic scan on any pull requests made to the repository to ensure the overall code coverage wouldn't decrease if the proposed code changes were added.

The second project was on a circular import issue within two of Grama's main files, core.py, and tools.py. The files were both importing the full Grama package within them and attempts at importing only specific methods were unsuccessful due to dependencies on code that was not yet created.

Results

Interviews and Coding

Our preliminary overall results, as seen below in Table 1, agree with our hypotheses that participants start with a bias towards induced variability. We came to this conclusion as interviewees gave more responses that aligned with induced variability (12) than real variability (11) for the Aluminum question.

Question	**Induced**	**Real**
Aluminum	12	11
Steel	7	12
Compare	8	7
Critique	5	12

Table 1. Counts of responses that aligned with induced and real variability for all n=7 participants, out of 14 possible counts

For more thorough analysis and discussion of the results, please see Aggarwal et. al., 2021.

Software

For the first project, we implemented the code coverage by using the software Codecov (https://codecov.io), their GitHub app, and by modifying our existing GitHub Actions code. Our final result was an inline report from the Codecov app on the proposed code modifications and the net code coverage difference. From there, our team could decide to accept the pull request or request changes, such as additional test code, before accepting it.

For the second project, we worked our way through both files and were able to isolate some external methods that provided core functionality to the package and move them into the core.py or tools.py file itself. We also discovered that an additional main functionality of Grama, marginals, was spread among the two files, causing them to be tied together unnecessarily, so we separated all related code into an additional base file, marginals.py. Once this transfer was complete, we were able to entirely remove importing the full package within all of these base files.

Future Work

One major takeaway was the setup of our interview guide. Our team began work on the interview guide early in the summer, prior to the expansion of our theoretical framework. As a result, the current interview guide focuses more heavily on the ideas of noise and deviation than mistake and anomaly. In addition, as we worked through the creation of the coding scheme, our team found the separation of the participants' answers into “analysis” and “reasoning” to be useful, though this hadn't been intentionally considered while writing the questions. Overall, we found we had framed our interview guide to be open enough that it encompasses these ideas, but it would be valuable to consider these points more carefully in a future rewrite.

Our preliminary results support our hypotheses about context effects and an induced-over-real bias, but future work with wider populations of engineers is necessary. Namely, future work should target engineering students from multiple institutions outside Olin to avoid exposure confounds.

Acknowledgments

With thanks to Sam Daitzman, Oscar de la Garza, and Dr. Diane Lam for their contributions to this work and Dr. Emily W. Tow and the Clare Boothe Luce Research Scholars Program for their advising and sponsorship of this research.

References

Aggarwal, R., & Flynn, M., & Daitzman, S., & Lam, D., & del Rosario, Z. R. (2021), *A Qualitative Study of Engineering Students' Reasoning About Statistical Variability.* Paper presented at 2021 Fall ASEE Middle Atlantic Section Meeting, Virtually Hosted by the section. https://peer.asee.org/38421

del Rosario, Z., Fenrich, R. W., & Iaccarino, G. (2021). When are allowables conservative? *AIAA Journal*, *59*(5), 1760–1772. https://doi.org/10.2514/1.j059578

Hogg, R. V. (1991). Statistical education: Improvements are badly needed. *The American Statistician*, *45*(4), 342. https://doi.org/10.2307/2684473

Shewhart, W. A. (1931). *Economic control of quality of manufactured product*. Van Nostrand.

Wild, C. J., & Pfannkuch, M. (1999). Statistical thinking in empirical enquiry. *International Statistical Review*, *67*(3), 223–248. https://doi.org/10.1111/j.1751-5823.1999.tb00442.x