Learning through creation and revision of student-developed R Shiny Apps in an applied statistics course

Dr. Heze Chen, University of Virginia

Heze Chen is an assistant professor in the Center for Applied Mathematics at the University of Virginia. He is involved in teaching several applied mathematics courses at the Engineering School. His research focuses on enhancing the mathematical learning experience for engineering students and developing numerical simulation methods in structural engineering.

Dr. Meiqin Li, University of Virginia

Dr. Meiqin Li is an Associate Professor at the University of Virginia. She joined UVA in 2017 after earning her Ph.D. in Applied Mathematics from Texas A&M University–College Station. Dr. Li has a strong interest in STEM education, particularly in curriculum development and redesign, the integration of instructional technologies to support student success, the creation of inclusive and diverse learning environments, and the use of a variety of pedagogical approaches to foster student learning. In addition to her education-focused work, her research interests include numerical computation, optimization, nonlinear analysis, and data science.

Anne M Fernando, University of Virginia

Anne Fernando (amf2cb@virginia.edu) received her PhD in Computational and Applied Mathematics from Old Dominion University. She holds an MS in Statistics, an MS in Applied Mathematics from the Georgia Institute of Technology. She was formerly Professor and Chair at Norfolk State University, currently Professor and Director for APMA in SEAS at the University of Virginia. Her research areas include Numerical Analysis, epidemiological modeling and enjoys studying mathematics with students.

Learning through Creation and Revision of Student-Developed Shiny Apps in an Applied Statistics Course

1. Introduction

In engineering courses that require programming, it is increasingly common for students to be proficient in coding to produce computational outputs that appear correct yet not understand what the results imply. They learn to utilize formulae and reproduce code patterns, yet the translation from computation to statistical reasoning is not usually reflected in their solutions, so for example, conceptual variability, ideas around uncertainty and inference, may remain abstract. Developing students' statistical reasoning can be a persistent instructional challenge, and strong performance in traditional coursework may not indicate deep conceptual understanding [1]. One response in applied statistics instruction has been to incorporate interactive web applications that allow students to manipulate inputs and immediately observe changes in outputs, creating a more direct bridge between computational outcomes and statistical ideas by removing the intermediate coding details and allowing the student to directly engage with a statistical model to bring about their conceptual understanding of the observed behavior [2] [3] [4]. Shiny is particularly practical for this purpose because it supports interactive, web-based applications built directly from statistical code in R or Python, without requiring separate web-development knowledge in HTML, CSS, or JavaScript [5]. It utilizes a reactive programming model to update outputs automatically based on user inputs. In studies of classroom adoption, Shiny resources are commonly framed as a way to support exploratory learning through their hands-on interaction rather than instructor passive demonstration [6]. Instructors' perceptions also suggest that Shiny resources can serve as accessible and reusable materials for data-centered instruction, especially when they are integrated into student course activities rather than treated as standalone examples [7].

Consistent with this shift toward interactive instruction, Shiny Applications (Apps) have also been integrated as reusable teaching modules across a range of university teaching settings and course contexts. In ecology curricula, Shiny based teaching modules have been used to support guided exploration of ecological forecasting concepts and workflows [8]. In computational social science, Li [9] reports interactive open education resources that combine Shiny with companion instructional packages to deliver introductory materials and practice activities across offerings in sequential semesters. Shiny Apps have also been adopted in subject-focused quantitative courses. For example, Neyhart and Watkins [10] describe a quantitative genetics module where simulation-based Apps allow students to test theoretical ideas through controlled parameter changes and immediate visualization. Within statistics and data science education, Shiny Apps have been paired with active learning activities to support more advanced topics, including support vector classifiers, by helping students connect model behavior to tuning choices and learning features [11], as well as to scaffold Bayesian data analysis lessons by providing an interactive environment for students to visualize uncertainty therefore connecting modeling assumptions to posterior outcomes [12].

Others have shown that students can create Shiny resources rather than just use them. For example, Wang et al. [13] reported student-developed Shiny Apps for teaching statistics, including a field test of short-term effectiveness and an emphasis on open-source code that supports reuse and adaptation. In an earlier implementation of student-developed Shiny Apps in an applied mathematics course [14], students described the work as challenging but rewarding and reported gains in both conceptual understanding and R programming skills. Also, Hanč et al. [7] discussed instructors' views of Shiny as a tool that can support open education and sharing of interactive materials. Together, these studies support the feasibility of student-developed application ecosystems and reinforce the idea that interactive modules can be treated as course resources rather than one-time creation. However, most published examples still center on instructor-selected Apps or one-time student assignments. In particular, there is less evidence on what happens when student-built Apps become part of a sustained instructional loop, where Apps persist across semesters, are reused by later cohorts as learning resources, and are explicitly revised through evaluation, feedback and redesign.

This study evaluates a multi-semester instructional process implemented in "From Data to Knowledge", an introductory applied statistical course. Across four semesters, student groups developed topic-aligned R Shiny Apps that map to core units in the course. After each semester, selected Apps and supporting materials were imported into a shared repository and then reused by later cohorts as interactive learning modules. In the final semester of the study, this instructional loop with Shiny Apps was closed through revision projects: each group selected a Shiny App from the repository, reconstructed its logic, identified limitations, and delivered an improved version with documentation. This study builds on our earlier work in the same course context and shifts the focus from initial implementation to the full cycle of creation, classroom use, critique, and revision as a sustainable model for active learning.

The rest of this study is organized as follows. Section 2 provides a quick overview of "From Data to Knowledge" and the R Shiny App integration, including the course context, the instructional process, and how the App repository was used by later cohorts. Section 3 describes the study design and methods, including data sources and analytic approaches. Section 4 presents the results, combining quantitative outcomes and qualitative patterns from student reflections and project artifacts. Finally, Section 5 concludes this study and outlines next steps.

2. A Quick Overview of the Course

"From Data to Knowledge" is a course designed for engineering undergraduate students that uses a case study approach to learn applied statistics and statistical learning techniques via R programming language in the RStudio environment. Its core statistical topics include confidence intervals, hypothesis testing, regression, and analysis of variance, followed by supervised learning methods for regression and classification and unsupervised learning methods such as clustering and principal component analysis. Students are expected to have prior exposure to probability and introductory statistics, but no prior programming experience is required. This section provides the course context needed to interpret the instructional design examined in this paper. It informs what students were expected to know, what they routinely did in the course, and why the creation and revision of R Shiny Apps is a natural extension of that workflow.

2.1 Course objectives and instructional approach

This course is designed to help students implement statistical ideas in code, interpret outputs, and clearly communicate results. The objectives emphasize effective use of R language, including writing, compiling, and debugging code both during class activities and outside of class assignments. They also include transferring statistical techniques to real applications, recognizing theoretical and technical limitations, developing analysis plans, and drawing evidence-driven conclusions. Instruction is organized around a sequence of case studies. Each case begins with a dataset and a motivating research question. Statistical techniques or methods are introduced as tools for answering that question rather than as individual topics in the sequence from the textbooks. The course uses a mix of widely adopted case studies such as Stat grade, the Vietnam War draft lottery, University of California Berkeley admissions, American roulette simulation, Old Faithful geyser eruptions, S&P 500 returns, along with additional datasets such as Boston housing and violent crime rates by U.S. state. Course materials combine textbook readings with instructor-developed resources distributed through the university learning management system.

2.2 Previous learning activities, assessments, and workflow

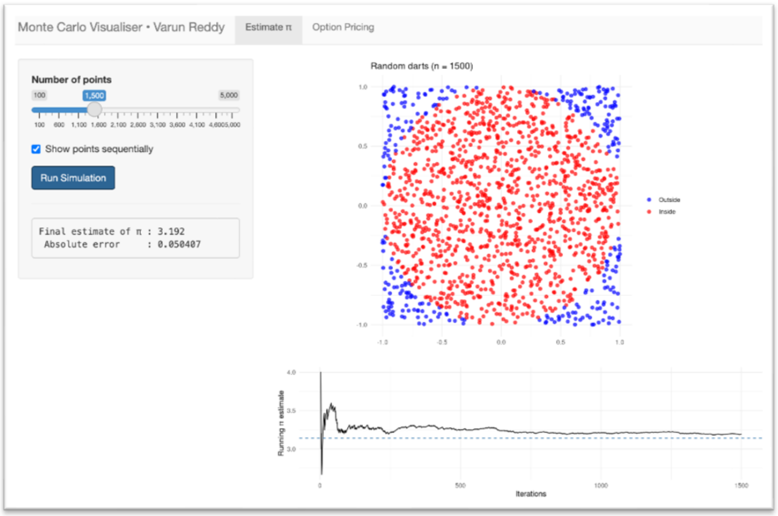
Before the R Shiny App introduction described in this paper, students developed statistical programming skills through assignments and course projects that emphasized implementing methods in R, interpreting results, and communicating conclusions in writing via:

- Practice sheets: most class lectures include a structured coding activity in a preset RMarkdown template. Each practice sheet typically includes several short problems or a fewer number of longer tasks. Students may work individually or collaboratively after class, but submissions are completed individually.
- Weekly group homework: weekly homework assignments are completed and submitted in groups. These assignments require students in group to write code largely from scratch, with a rough RMarkdown template provided to standardize formatting and reporting.
- Exams: three exams assess both conceptual understanding and the ability to implement and interpret statistical methods in R language.
- Final research project: it is a half-semester-long group research report in RMarkdown that requires students to apply course methods to a dataset of interests. Students select a publicly available dataset, formulate a focused research question, and use at least two techniques from the course to support their conclusions with appropriate visualizations and statistical evidence.

These activities establish the traditional workflow of the course, which later supports the R Shiny based projects described next.

2.3 R Shiny App integration in the course

Beginning in Spring 2024, R Shiny Apps were introduced in “From Data to Knowledge” and gradually developed by enrolled students across later offerings of the course. Each offering involved a different cohort of students, and the role of the Apps changed over time. In the early stage, enrolled students created R Shiny Apps as extra-credit work, to illustrate core statistical concepts or datasets from the course. These Apps were reviewed by peers and instructors using criteria such as user friendliness, responsiveness, topic relevance, functionality, and aesthetics, and selected Apps were added to a shared course repository. Figure 1 shows an example of a student-developed R Shiny App on "Monte Carlo Simulation," which was created by a student and peer-reviewed in Spring 2024.



What does the app do?
This app offers two side-by-side Monte Carlo demos that turn probability theory into moving pictures:

- Estimate π (Dartboard method): Random points are thrown uniformly at a square enclosing a unit circle. The fraction landing inside the circle, multiplied by 4, yields a running estimate of π. A scatterplot shows every dart.
- Option Pricing (European call): Thousands of stock-price paths that follow geometric Brownian motion are simulated, discounted payoff max(St - K, 0) for each path is computed, and are averaged to obtain the option's fair value. Its plot tracks convergence vs. iteration count.

Figure 1. Example student-developed R Shiny App with an excerpt of its project documentation

As the repository grew, those selected Apps were used as learning resources during class activities and assignments. This accumulated repository set the stage for a redesign in Fall 2025, when there were enough Apps that visualize and apply statistical topics, including hypothesis testing, confidence intervals, regression, clustering, and probability distribution behavior (see the categorical summary in Table 1 of the Appendix). The instructional goal shifted from creating an App to learning by exploring, critiquing, and improving an existing App. All students enrolled in the course during that semester completed a mid-semester revision project, and students could also complete an optional second revision project. The two projects were structured as follows:

- *Project 1 (required, group, mid semester): revise and improve an existing App.* Students in groups selected one student-developed App from the course list, analyzed what it was designed to teach and how it worked, and then produced an improved version. The work was organized around two tasks: *(i)* investigate the selected App, including its purpose, use, strengths, and limitations, and *(ii)* implement and document a revised App based on that investigation. Deliverables included a short report (maximum two pages) with required components (abstract, App investigation, App improvement process, improved App description, remaining limitations, and a brief reflection) and two videos: an App building video documenting the development process step by step, and an App user guide video demonstrating all major features.
- *Project 2 (optional, group/individual, end of semester): extend and deploy a polished revision.* Students selected a different student-developed App from the same repository, focused on at least one post midterm topic, and produced a revised version that was polished and fully functional. The project also required deployment as an R Shiny App and submission of the App link, along with a short user guide video (about one minute) stating the App's purpose, demonstrating inputs and outputs, and showing one concrete example of what a user can learn from the App.

Table 2 below summarizes this progression across course offerings, including the motivation for each stage, the number of Apps developed, the instructional role of the Apps, the primary student learning task, and the data sources available for analysis described in Section 3.

Table 2. Evolution of R Shiny App integration and data sources across course semesters

	Spring 2024	**Fall 2024**	**Spring 2025**	**Summer 2025**	**Fall 2025**
Stage	Pilot App development	Early App integration	Expanded integration	Quasi-experimental comparison	Revision-cycle implementation
Motivation	Test whether student-created Apps could support course concepts	Begin using selected Apps as course learning resources	Expand App use across more course topics	Compare outcomes across sections with different App exposure	Close the instructional loop through critique and revision of existing Apps
Developed App numbers	5	13	10	N/A	20 (not including the required mid-semester projects)
App instructional role	Pilot examples shared through class presentation	Supporting visualizations for key topics	Integrated across more units and assignments	Systematic use within case studies	Apps treated as revisable learning resources
General student learning task	Review Apps	Use and explore Apps to interpret outputs	Use and explore Apps + reflect using Apps	Use Apps as assigned within case studies	Critique + revise an existing App
Quantitative data	N/A	Closed-ended survey responses	Closed-ended survey responses	Three common exams + closed-ended survey responses	App scores + closed-ended survey responses
Qualitative data	N/A	Open-ended survey responses	Open-ended survey responses	Open-ended survey responses	Open-ended survey responses + artifacts (code, report, peer feedback, video)

3. Study Design and Methods

The primary evidence comes from end-of-semester Qualtrics surveys that capture student perceptions, experiences, and perceived learning value of R Shiny Apps. These survey results are complemented by course performance measures tied to concepts aligned with R Shiny Apps, and project artifacts from the required App revision project and the optional extension for Fall 2025. Together, these quantitative and qualitative sources, support a mixed methods analysis of both outcomes and mechanisms.

3.1 Study design in the past and current semesters

This study uses a mixed-methods design across four course offerings of “From Data to Knowledge” from Fall 2024 to Fall 2025, with the course offerings and data sources summarized in Table 2. Our analysis follows the Interactive Tools Impact Assessment approach described in [14] by pairing student perception data with topic-aligned outcomes and examining differences across exposure conditions with R Shiny Apps. In particular, Summer 2025 provides a two-section contrast because both sections used the same exams, while Fall 2025 includes additional artifacts (code, reports, peer feedback, and videos) that support qualitative analysis of the App revision.

3.2 Participants and data sources

Participants were students enrolled in “From Data to Knowledge” across four semesters during the experiment period. This study draws on three data sources. First, end-of-semester Qualtrics surveys reflect student perceptions of R Shiny Apps, and include both Likert-scale items and open-response questions. Second, course performance in Summer 2025 measures is extracted from assessments that align with the statistical concepts represented in the R Shiny Apps. Third, for Fall 2025, we analyze project artifacts from the App revision project and the optional project, including App web links, written reports, peer-review responses, and short videos documenting both the development process and intended App use. When building projects in Fall 2025, we intentionally incorporated the components of learning mechanism (refer to Section 4) into the project description that we could utilize students' project artifacts to investigate RQ3 (refer to Section 4).

3.3 Survey instrument and key measures

The surveys combine Likert-scale items, multiple choice questions, and short open questions. Questions were designed to obtain whether R Shiny Apps help students connect computational implementation to statistical concepts, interpret outputs, and remain engaged. The surveys also collect background information such as academic level, major, and prior experience with R programming or statistics to support interpretation across cohorts. Key measures are organized around recurring survey categories, including participation, perceived learning outcomes, skill development, engagement and motivation, and challenges and barriers. For cross-semester comparisons, analysis emphasizes items that are shared across all the semesters from Fall 2024 to Fall 2025 except Summer 2025. The Fall 2025 survey additionally includes Shiny-specific questions related to the App selected for the revision project and the revision project itself, which supports linking student perceptions to the revision-based implementation examined in this study.

3.4 Analysis procedure

Our analysis integrates quantitative comparisons with qualitative analysis. We first construct a semester-level dataset that tags each semester by R Shiny exposure condition and treats Fall 2025 as a distinct revision-cycle implementation because students were required to critique and improve an existing App and some continued through the optional extension.

For quantitative outcomes, we use two sources. *(i)* Topic-aligned performance is measured with scores on targeted assessment items tied to concepts represented in the R Shiny Apps, or composite scores, rather than overall exam totals. The performance comparison is Summer 2025, because the two sections used the same exams, and we analyze section differences with independent-samples t-tests. *(ii)* Perception and engagement outcomes come from the surveys. We report item-level summaries built from shared Likert items. Because exams differ across semesters, cross-semester comparisons are only limited to common survey measures and are tested with one-way ANOVA statistical test.

For qualitative analysis, the data are open-response survey questions and Fall 2025 project artifacts. Such data are analyzed through a thematic analysis approach based on Braun and Clarke's framework [15], which supports systematic identification and interpretation of patterns within qualitative data. The analysis employs a combined inductive and deductive coding process [16], inductive coding captured themes aligned with RQ3 (refer to Section 4), while deductive coding allows for the inclusion of additional insights that emerge during the analysis, consistent with robust hybrid thematic analysis practices. Themes are refined through iterative review and cross-checked against quantitative session ratings to enhance consistency and ensure alignment with reflexive thematic analysis principles.

4. Results and Discussion

In the previous section, Table 2 summarizes how R Shiny Apps were integrated across semesters, including the Summer 2025 two-section contrast and the Fall 2025 revision cycle. The results are organized around three research questions below:

1. How do students describe R Shiny Apps' role in helping them connect statistical concepts to computation, interpret outputs, and stay engaged? (Fall 2024, Spring 2025, Fall 2025)
2. To what extent is access to/reuse of R Shiny Apps associated with higher performance on topic-aligned assessment items? (Summer 2025)
3. In the Fall 2025 revision cycle, what learning mechanisms appear in student reflections and project artifacts (e.g., conceptual clarity, debugging workflow, reusability), and what challenges persist? (Fall 2025)

The first two research questions focus on student perceptions and topic-aligned performance, while the third research question examines the learning processes visible in the Fall 2025 revision cycle. We use learning mechanisms to describe the ways students' reflections and project artifacts show learning through revision, including how they connected statistical ideas to App features, rebuilt and tested code, identified limitations, overcame challenges, and considered how the Apps could be used by future students.

4.1 *Research Question 1: How do students describe R Shiny Apps' role in helping them connect statistical concepts to computation, interpret outputs, and stay engaged?*

Two sets of questions (A & B) from end-of-semester survey responses were compared across three independent student cohorts: Fall 2024 (n=59), Spring 2025 (n=7), and Fall 2025 (n=61), where n is the number of student respondents.

(A) Please rate the following statements regarding our extra credit R Shiny assignment:

1. *"I found the process of creating R Shiny applications challenging."*
2. *"Working on R Shiny applications improved my skills in R programming."*
3. *"Working on R Shiny applications improved my understanding of the statistical concepts taught in this course."*
4. *"Working on R Shiny applications has increased my interest in exploring data science further."*
5. *"I found the process of creating R Shiny applications rewarding."*

(B) Even if you did not complete the extra credit assignment, please rate the following statements regarding our R Shiny Apps:

1. *"R Shiny apps could make complex topics more accessible and understandable."*
2. *"I am likely to engage more in class if R Shiny apps are used regularly."*
3. *"I would like to see more R Shiny apps integrated into our classroom teaching."*

All items used a 5-point Likert scale (1-5). For each item, we tested for differences in mean ratings across semesters using Welch's one-way ANOVA. It is appropriate here because group sizes are highly unbalanced (especially Spring 2025), and we do not want conclusions to depend on an equal-variance assumption. Figure 2 shows the mean rating in each semester for Survey question (A).

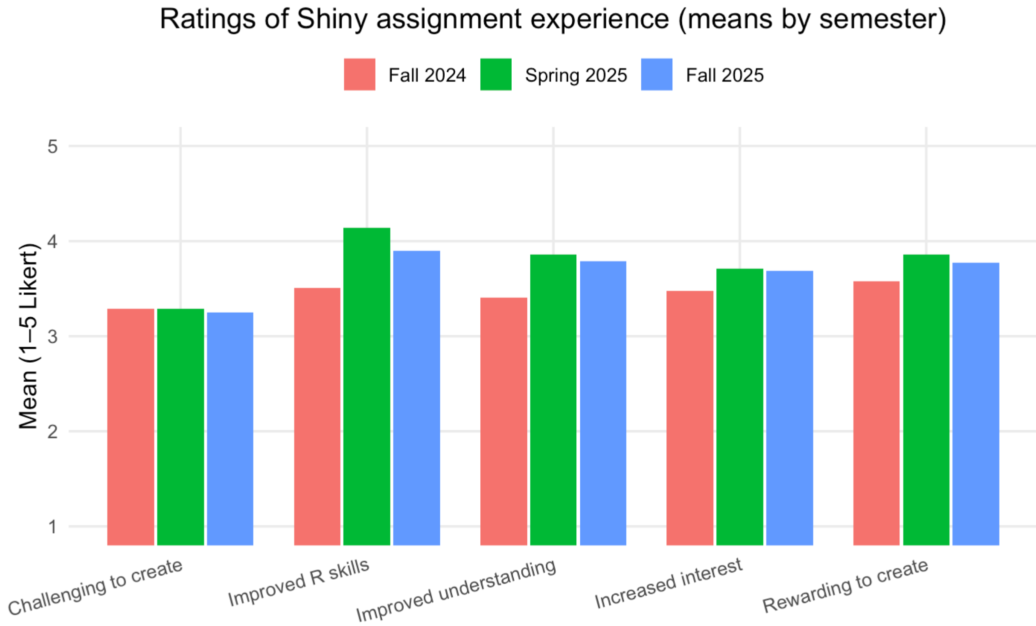


Figure 2. Extra-credit R Shiny App assignment mean ratings - Survey question (A)

Across the three cohorts, perceptions of the difficulty of building R Shiny Apps were stable, where the mean ratings for "challenging" were essentially unchanged, and Welch's one-way ANOVA showed no semester effect ($p = 0.968$). Interest in exploring data science was positive across terms but did not differ significantly ($p = 0.291$). Students also rated the experience as rewarding in all semesters, with only a modest upward shift after Fall 2024 that was not statistically significant ($p = 0.356$). The clearest cross-semester differences occurred in the two learning-gain items: students in Spring 2025 and Fall 2025 reported higher perceived gains in both R programming skills ($p = 0.011$) and statistical understanding ($p = 0.025$) than in Fall 2024, and these were the only items with statistically significant differences. Because Spring 2025 has a small sample ($n = 7$), we treat its means descriptively, even though they are the highest. Focusing on the larger cohorts (Fall 2024 vs. Fall 2025), the pattern still suggests stronger perceived gains in R skills and statistical understanding in Fall 2025, consistent with the expanded Shiny activity and a richer App repository.

Figure 3 summarizes mean ratings for in-class R Shiny App use by semester. Overall, students viewed classroom R Shiny Apps favorably, and the strongest consistency was in perceived conceptual support. Ratings for "making complex topics more accessible and understandable" remained high across cohorts, and Welch's one-way ANOVA found little evidence of a semester effect ($p = 0.577$). Perceptions tied to participation and demand for more Apps showed greater movement across semesters. "Likely to engage more" was lower in Fall 2025 than Fall 2024 in the means, but the cohort differences were not statistically reliable under

Welch's ANOVA (p = 0.236). "Wanting more Shiny integration" showed the largest apparent separation among semesters, but the ANOVA result was only suggestive rather than significant at conventional thresholds of 0.05 (p = 0.073). Given the small Spring 2025 sample (n = 7), we interpret that cohort cautiously; taken together, the results suggest that perceived support for understanding is stable, whereas engagement and appetite for expanded use are more sensitive to cohort context and implementation.

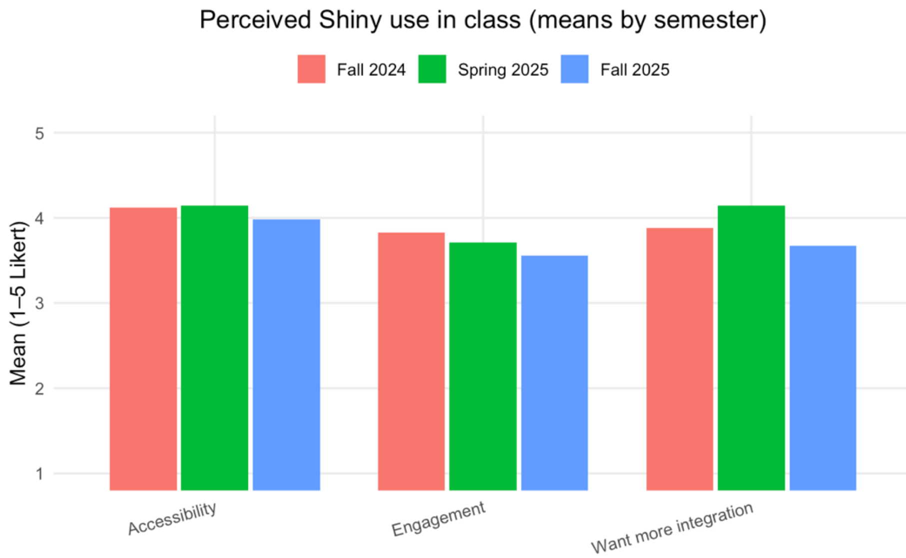


Figure 3. R Shiny App classroom use mean ratings - Survey question (B)

Across those three semesters, the item-level results show a stable signal with a few focused shifts. Several perceptions change little from time to time, though students consistently rate R Shiny Apps as making complex topics more accessible, and they describe the app-building experience as challenging but more interesting and rewarding. The clearest movement appears in the two learning-gain items: later offerings show higher self-reported gains in both R programming skills and statistical understanding than the earlier cohort. In contrast, the engagement-related items (e.g., being more likely to engage in class and wanting more R Shiny integration) fluctuate more across semesters, suggesting these perceptions are more sensitive to cohort context and implementation details than to the presence of R Shiny Apps alone.

4.2 *Research Question 2: To what extent is access to/reuse of R Shiny Apps associated with higher performance on topic-aligned assessment items?*

In Summer 2025, the two sections (001 & 002) in two summer sessions took the same set of assessments, but differed in R Shiny App exposure. In Section 001 (without Apps; n = 19), instruction did not integrate R Shiny Apps as structured learning resources. In Section 002 (with Apps; n = 26), R Shiny Apps were used repeatedly as interactive support across topics (normal distribution, central limit theorem, simulation, inference, regression diagnostics, chi-square goodness of fit, clustering, etc.), including student-developed apps from previous semesters.

The research question in this section only uses the Summer 2025 semester's data because it provides a two-section contrast with a common assessment instrument. Student-level scores contain section identifiers (Section 001 vs Section 002) and problem-level or subpart-level scores for Exam 1 to Exam 3. Subpart scores were recorded but aggregated to problem totals when needed. The analysis focused on a series of topic-aligned problems from those three exams, which are listed in the subfigure titles in Figure 4. They also correspond to the specific topics supported by R Shiny Apps used in Section 002 (see Table 3 in Appendix).

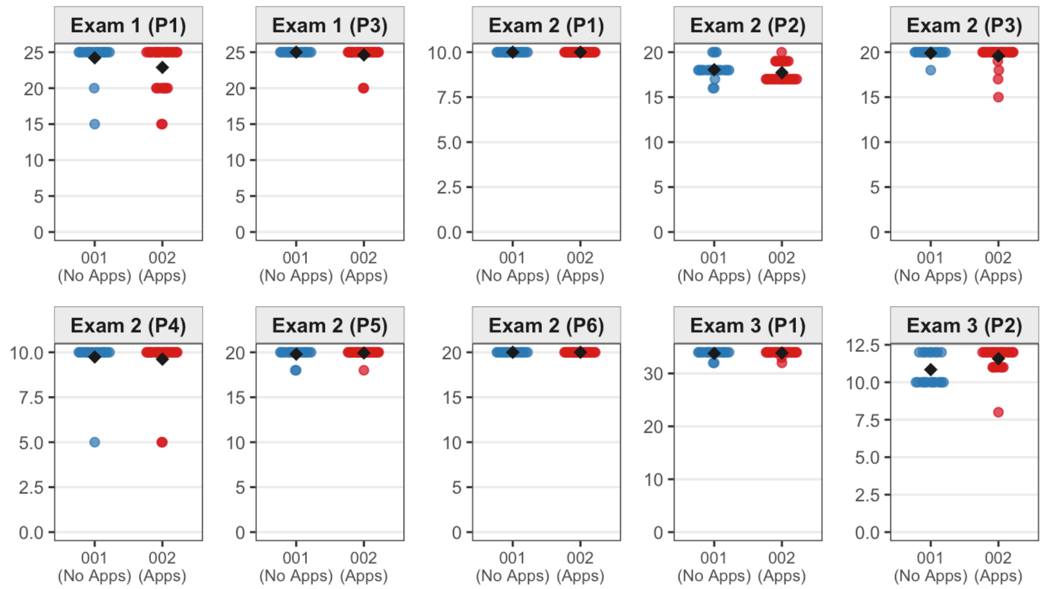


Figure 4. Summer 2025 exam scores by section
(001: blue dots; 002: red dots; section average: black diamond)

For each topic-aligned problem, the distributions of scores by section were summarized above and compared using independent-sample Welch's t tests (the results are shown in Table 4 in Appendix). Standard deviations (SD) and confidence intervals (CI) were computed to supplement p-values. In addition to problem-level comparisons, a composite topic-aligned score was formed by averaging the selected problem totals, and the same independent-samples comparison was applied to the composite.

Both sections scored near the top of the problem score ranges, and the score distributions overlapped substantially. Several problems, including Exam 2 (P1), Exam 2 (P4) (P5) (P6), and Exam 3 (P1), had most students earn full or near-full credit. The topic-aligned composite score (percent across those 10 problems) was similarly high in both sections: Section 001 averaged 97.26% (SD = 2.37) and Section 002 averaged 96.84% (SD = 2.20), with trivial evidence of a section-level difference ($t(43) = 0.61$, $p = 0.545$; mean difference = -0.42, with 95% CI [-1.80, 0.97]). At the problem level, section differences were generally small and mixed in direction. Most problems showed similar score distributions and comparable means across sections, and there was no sign of a consistent performance advantage associated with R Shiny App exposure.

One exception was Exam 3 (P2), the chi-square goodness-of-fit problem, where Section 002 (with Apps) performed higher on average. Figure 5 provides useful context by showing the two-part exam problem alongside the chi-square R Shiny App ("Understanding Chi-Square Densities") used in class. That App makes the chi-square distribution more intuitive by linking degrees of freedom and significance level to the critical value and shaded tail area, which mirrors what students must do on the exam (clearly stating the null hypothesis and the alternative hypothesis, computing the test statistic, and interpreting evidence using a p-value or rejection region).

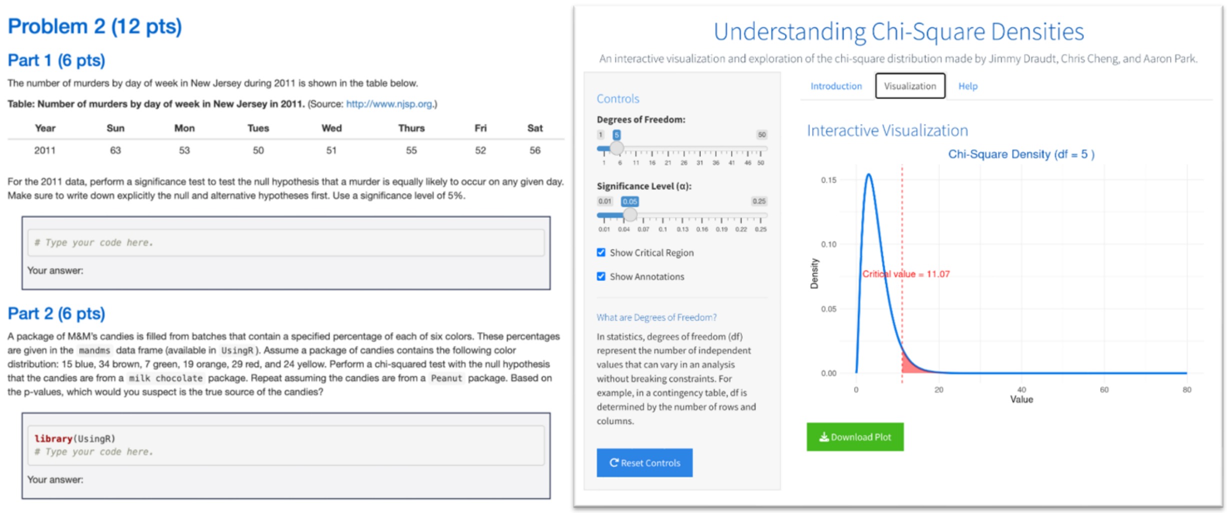


Figure 5. Summer 2025 Exam 3 Problem 2 (left) and the screenshot of the corresponding R Shiny App (right)

Based on the rubric and the pattern of student work for that problem, lost points on this problem were often tied to incorrect or incomplete hypothesis statements (not explicitly stating the null and alternative hypotheses) and errors in p-value calculation or interpretation, even when the code structure was mostly reasonable. That R Shiny App does not automatically fix the statistical setup, but likely helps students "see" what the p-value represents and how it connects to the test statistic. In that sense, the result from Exam 3 Problem 2 can be consistent with a targeted benefit on a topic where visualization directly addresses confusion. It is also worth noting that the quality of the App itself can be a potential factor. The chi-square app used here is one of the strongest R Shiny Apps in the set from instructor evaluation, with positive feedback from students as well. The combination of strong alignment to the assessed skill, interactive feedback that supports interpretation, and a well-designed interface may help explain why this topic stands out compared to others.

The prevalence of ceiling effects across many problems limits what can be detected with these outcomes. When most students already score at or near full credit, there is little room for additional measured improvement, even if apps are supporting learning in ways that are not captured by high exam scores. In summary, the Summer 2025 results do not suggest that R Shiny App exposure consistently raised topic-aligned exam scores across problems. Instead, the results point to high-quality Apps being most useful when they directly support a specific skill.

4.3 *Research Question 3: In the Fall 2025 revision cycle, what learning mechanisms appear in student reflections and project artifacts (e.g., conceptual clarity, debugging workflow, reusability), and what challenges persist?*

We applied a combined inductive-deductive coding approach, as outlined in Section 3.4, to analyze the qualitative data. Driven by the research questions, we read through the open-ended responses and Fall 2025 project artifacts, identified relevant constructs and themes such as interpretability of outputs, conceptual clarity, debugging workflow, team coordination, and perceived reusability, and developed an initial coding framework. We then coded the data using this framework while remaining attentive to new patterns emerging from the data. As coding progressed, we evaluated whether the data supported, refined, or contradicted the initial interpretations and updated the coding scheme accordingly. Representative excerpts were selected to show the common patterns behind those themes, and qualitative findings are then used to explain and validate the quantitative patterns.

Students consistently translated core statistical concepts into interactive features that improved learnability (e.g., comparison overlays, explanations, diagnostics). During Fall 2025, all the enrolled students were required to team up to critique and improve an existing R Shiny App using an "explore → recreate → extend → validate" workflow, shifting apps from demonstrations to systems that must be understood, rebuilt, and made more reliable for others. Selected feedback on the project assignment is summarized in Figure 6 (Appendix). In the same semester, students were asked which Shiny-related experience contributed most to their learning; responses favored the creation/revision work (44.3%), followed by "both equally helpful" (31.1%), with fewer selecting app use alone (13.1%) or neither (11.5%). Across all 32 groups, learning mechanisms clustered around conceptual consolidation through implementation, and the development of adaptive technical workflows, while challenges persisted around reactive complexity and different technical preparation within teams.

4.3.1 Conceptual consolidation in building the R Shiny App

Most groups framed revisions as a way to make statistical ideas visible and testable through interactive controls, and many emphasized making outputs easier to interpret by adding explanations, summaries, or diagnostics. Their reports also commonly documented limits and trade-offs, suggesting that revision work prompted students to reason about what the R Shiny App can potentially support, not only what it can display. Consistent with these observations, Table 5 explicitly lists the most common conceptual consolidation themes that appeared in group reports, along with the number of groups mentioning each theme.

Table 5. Conceptual consolidation themes in project reports

Theme	**Frequency (Groups)**	**Typical Responses** (excerpts are verbatim with light trimming for clarity)
Visualization as a bridge from theory to understanding	23	• "These enhancements transform the app from a simple visualization tool into a comprehensive learning platform that supports comparison, explanation, and practice." • "The interface helps students verify theoretical concepts by directly observing how parameter changes affect shape, spread, and position of the curve."
Clear problem framing & user needs (why this app; who benefits)	21	• "This app is useful because it helps students build visual intuition about probability distributions and quickly explore how parameters change outcomes." • "The tool is especially helpful for teaching, allowing instructors to demonstrate how mean and standard deviation affect the distribution in real time."

Iterative design: from replicating baseline to add targeted improvements	20	• "We began by recreating the original Shiny app so it functioned identically to the base version. Once the base was stable, we incrementally added new features and tested each change before proceeding."
Data-driven insight & transparency (statistics, diagnostics, explanations)	18	• "We added a skewness summary with written interpretations so users can connect visuals to quantitative properties without extra computation." • "Summary tables and explanations clarify how features like confidence bands and correlation coefficients inform interpretation."
Awareness of limits & trade-offs	17	• "Even with improvements, the app is limited to certain distributions and may not handle large or messy datasets." • "The tool remains primarily a visualization aid and does not yet support advanced calculations or multi-dataset comparisons."
Course concept consolidation (deeper mastery through building)	16	• "Developing the app strengthened our understanding of statistical principles by requiring accurate computations and clear explanations." • "Building visualizations deepened our grasp of how parameters influence distributions and model behavior."

These themes are echoed in individual reflections describing concrete concept gains, such as significance level and sample size effects, regression roles and outliers, skewness, and hypothesis testing (see the selected student comments in the end of Appendix). Fall 2025 students also rated how much the R Shiny App projects helped on five learning targets (1-10 scale; item n = 62). Ratings were consistently positive, with means ranging from 6.56 to 7.87 and moderate spread (SD ≈ 1.60-2.02). The highest ratings were for seeing how changing inputs affects results/plots (mean = 7.87), while matching code steps to statistical steps was lowest (mean = 6.56). The figure below shows the distribution of responses across items.

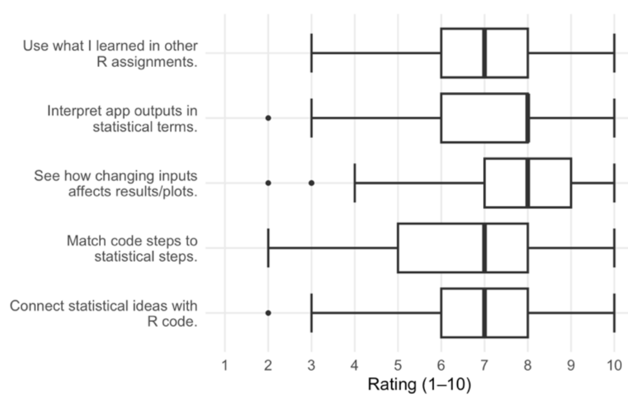


Figure 7. Fall 2025 student ratings of how R Shiny App projects supported learning (1-10 scale)

As one example from the Fall 2025 revision cycle, Figure 8 shows how a student group enhanced a "Distribution Visualizer" R Shiny App originally developed by a prior group to support exploration of probability distributions. Their revisions added a toggle between probability density function and cumulative distribution function views, parameter sliders for real-time updates, and additional distributions (t, chi-square, and exponential). Most groups connected features to learning goals and showed coherent from investigation to improvement, and finally to reflection with explicit limits, which is the evidence of conceptual consolidation.

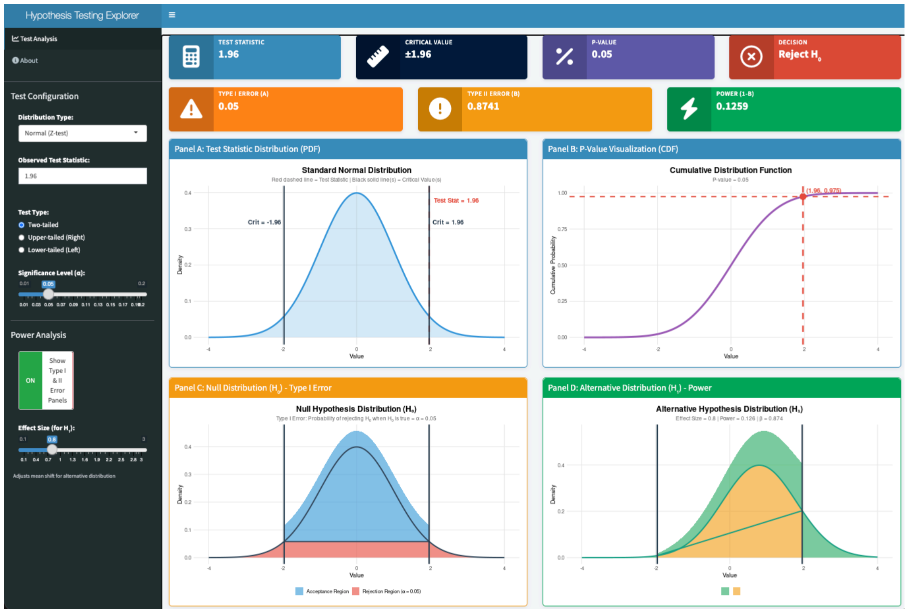


Figure 8. R Shiny App example created and revised by a group of students in Fall 2025

4.3.2 Adaptive workflows for re-creating the App

The students also described the workflow as reproducing the baseline app first, then adding targeted improvements incrementally while testing each step to preserve stability. Many reports describe debugging as iterative and local, with some groups explicitly restructuring reactivity to avoid endless updates and improve responsiveness. Several groups also reported adding guardrails such as input validation and clearer user feedback, reflecting a shift from "making it run" to "making it robust for others". The table below summarizes these adaptive workflow themes that emerged across reports, with the number of groups that mentioned each theme at least once and representative excerpts from their reports.

Table 6. R Shiny App adaptive workflow themes

Theme	**Frequency (Groups)**	**Typical Responses** (excerpts are verbatim with light trimming for clarity)
Baseline replication towards incremental feature additions with testing	20	"First we reproduced the base app to match behavior and layout, then added improvements one by one, testing each step to maintain stability."
Small, testable steps; feature-by-feature integration	18	"We implemented one feature at a time and verified it before proceeding, which reduced breakage and made debugging manageable."
Reactive patterns (event Reactive/observe Event, conditional panels)	8	"We used eventReactive for data generation and observeEvent for user-triggered computations to avoid circular updates and improve responsiveness."
Toolchain & collaboration (GitHub, Posit Cloud/RStudio)	6	"We coordinated edits via a shared repository and merged tested changes, which helped despite single-editor limits in Posit Cloud."
Guardrails & reproducibility (validation, fixed seeds, error handling)	6	"We validated inputs, limited variable choices to numeric columns, added clear messages for invalid data, and fixed seeds for fair comparisons."
Architectural improvements (modules, delayed reactives, performance)	3	"We modularized server logic and delayed expensive reactives to keep the UI responsive during larger computations."

4.3.3 Persisting challenges and benefits

Student group reports identified difficulties in completing the revision project, which are summarized in Table 7. The most common challenges involved R Shiny's reactive model and debugging, particularly when adding dynamic behavior or expanding inputs. Other friction points included plotting or readability issues and data handling when enabling flexible user inputs. Several reports also point to coordination costs and different technical skill levels within groups, which shaped how efficiently teams could implement and validate changes.

Table 7. Persisting challenges and common fixes reported by groups

Theme	**Frequency (Groups)**	**Typical Responses** (excerpts are verbatim with light trimming for clarity)
Learning curve with Shiny/reactive design	12	"Our unfamiliarity with Shiny required time to learn reactive structures, UI/server interactions, and debugging strategies before building features."
Dynamic UI / reactive dependency issues	10	"We initially encountered circular update loops; restructuring with eventReactive and observeEvent resolved these dependencies."
Visualization gotchas (overlays, scaling, opacity, axes)	6	"Overlaid plots obscured each other until we used semi-transparent colors and aligned axes so both distributions remained visible."
Data ingestion & validation (uploads, types, missing)	7	"Adding file upload surfaced type and missing-value issues; we filtered to numeric columns, skipped NAs with notices, and blocked invalid operations."
Collaboration/time constraints	2	"A merged team across sections introduced communication challenges; a group chat and version control helped us coordinate work."
Performance / lag (re-render churn, latency)	3	"New features temporarily broke earlier functionality; we backtracked, simplified redraw logic, and tested incrementally."

Even with these challenges, a number of groups described clear benefits from the revision cycle listed in Table 8. Many reports emphasized practical R/Shiny skill gains, especially in connecting user-interface inputs to reactive server logic, producing interactive graphics, and adding validation and error handling. They also reported deeper conceptual understanding of core topics (e.g., skewness, central limit theorem, robustness, inference) that came from implementing statistical ideas as features and explanations. Finally, many groups framed their revised apps as durable teaching tools that can be reused in future offerings.

Table 8. Benefits when building the applications reported by groups

Theme	Frequency (Groups)	**Typical Responses** (excerpts are verbatim with light trimming for clarity)
Practical R/Shiny skill gains	24	We gained hands-on experience connecting UI inputs to reactive server logic and building interactive graphics for analysis and teaching.
Deeper conceptual understanding (distributions, CLT, regression, tests)	22	Implementing features like skewness summaries, QQ plots, and confidence bands strengthened our statistical reasoning and interpretation.
Creation of durable teaching tools	20	The improved apps turned abstract concepts into interactive learning experiences that instructors and students can reuse.
UX thinking & communication (explanations, guardrails)	8	We wrote explanations, added tooltips, and constrained inputs so users could understand results and avoid common errors.
Teamwork & project management	6	We coordinated editing windows, merged tested code, and iterated collectively to manage scope and timelines.

Figure 9 summarizes how students ranked the revision project components from most to least helpful. Investigating the original App and understanding its purpose was rated most helpful, followed by writing and revising the R Shiny App code and then identifying limitations and selecting improvements. Debugging and testing tended to fall in the middle, while writing the report and creating videos were most often ranked least helpful. These rankings match what groups said in Table 7. Students reported the most learning when they had to understand and rebuild the app's logic, while report or video tasks felt less useful, and debugging and coordination were challenges. Overall, the revision cycle was difficult to carry out but provided strong learning benefits.

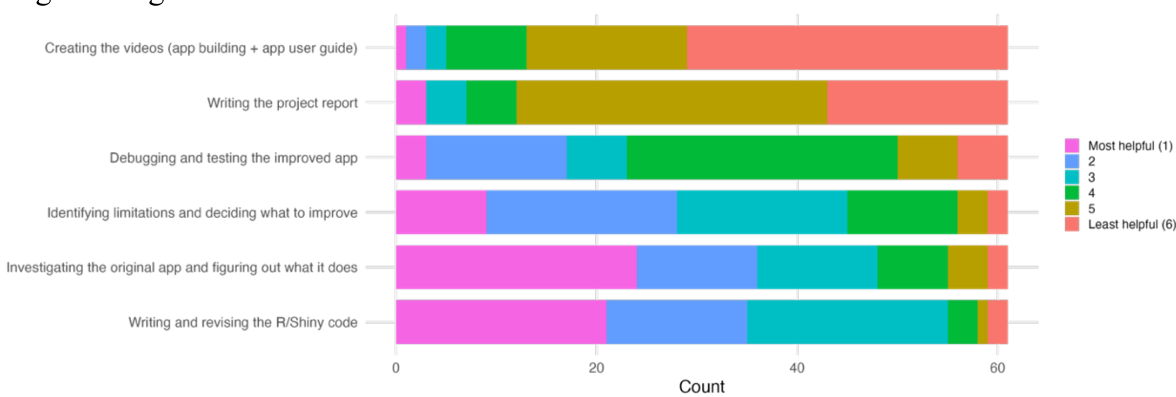


Figure 9. Student rankings of project components from most to least helpful (Fall 2025)

5. Conclusions

This study experimented a complete instructional loop in which student-developed R Shiny Apps are created, reused as course resources, and then revised by a later semester cohort in an applied statistics course with an R programming emphasis. Across semesters, the App repository functioned as part of the growing course infrastructure, as selected student-developed Apps were integrated into class activities and assignments in later semesters, and the Fall 2025 project required students to improve an App from the repository. Evidence indicates that students consistently view Shiny work as challenging yet rewarding, and the clearest cross-semester improvement is in perceived learning gains, with later cohorts reporting higher self-assessed gains in both R programming and statistical understanding. A two-section comparison in

Summer 2025 using common exams did not show a significant advantage associated with App exposure, and most topic-aligned items exhibited ceiling effects, suggesting that score-based impacts may depend on measurement sensitivity and alignment of the R Shiny Apps and assessments. In contrast, the Fall 2025 revision cycle provides stronger evidence about learning mechanisms. The recreation, extension, validation steps in the R Shiny App revision project shifted them from demonstrations to systems that must be understood, rebuilt, and made reliable for others, and students most favored creation or revision work for learning. The revision cycle produced comprehensive benefits and gains in R Shiny App implementation and conceptual understanding through rebuilding and extending an existing App, while challenges centered on reactive debugging, dynamic dependencies, data and plot handling, and coordination under various prior preparation levels.

Since Fall 2025 semester, selected R Shiny Apps have been categorized and compiled into a dedicated repository website to support easier distribution and reuse. At this stage, the repository has primarily been used in “From Data to Knowledge” but the instructional model has potential applicability across other undergraduate majors in the local engineering school. For instance, in systems engineering, the disciplinary work product is not always a physical artifact. Therefore, creating a Shiny App can provide students with a concrete, shareable product while requiring them to apply modeling, data analysis, communication, and design skills that are directly relevant to the discipline. Other engineering areas, such as electrical, chemical, mechanical, and aerospace engineering, may also benefit from this project model because engineering practice broadly relies on statistics, modeling, simulation, and interpretation of results. Student-created Shiny Apps can therefore provide an early, small-scale experience with the type of analytical workflow students may later encounter in research or industry settings. Future work will examine whether this repository and revision-project model can be adapted to other statistics focused courses, second-year major specific engineering courses, and first-year engineering curricula.

Acknowledgments

This work was supported by the Center for Applied Mathematics in the School of Engineering and Applied Science at the University of Virginia. This study was reviewed and approved by University of Virginia Institutional Review Board (IRB) [Protocol #6947]. The authors greatly thank colleagues who provided help and guidance on the IRB process. The authors also appreciate the participation of all the students enrolled in the course described in this paper from Spring 2024 through Fall 2025.

References

[1] J. Garfield, “The challenge of developing statistical reasoning,” *Journal of Statistics Education*, vol. 10, no. 3, 2002, doi: 10.1080/10691898.2002.11910676.

[2] Y. Lu, “Web-Based Applets for Facilitating Simulations and Generating Randomized Datasets for Teaching Statistics,” *Journal of Statistics and Data Science Education*, vol. 31, no. 3, pp. 264–272, 2023, doi: 10.1080/26939169.2022.2146614.

[3] Z. Branson, M. Paz Parra, and R. Yurko, “The Landscape of College-Level Data Visualization Courses, and the Benefits of Incorporating Statistical Thinking,” *Journal of*

Statistics and Data Science Education, vol. 33, no. 4, pp. 390–406, 2025, doi: 10.1080/26939169.2025.2537049.

[4] E. Berkmann, B. Fisseler, L. Schützler, and O. Christ, "Shiny Apps in Distance Education: Do Psychology Students Benefit From Interactive Statistics Applications?," *Journal of Statistics and Data Science Education*, 2025, doi: 10.1080/26939169.2025.2479008.

[5] Posit Software, PBC, "Easy web apps for data science without the compromises," *Shiny*, [Online]. Available: https://shiny.posit.co/ [Accessed: Apr. 27, 2026].

[6] L. Fawcett, "Using Interactive Shiny Applications to Facilitate Research-Informed Learning and Teaching," *Journal of Statistics Education*, vol. 26, no. 1, pp. 2–16, Jan. 2018, doi: 10.1080/10691898.2018.1436999.

[7] J. Hanč, P. Štrauch, E. Paňková, and M. Hančová, "Teachers' perception of Jupyter and R Shiny as digital tools for open education and science," Jul. 2020, [Online]. Available: http://arxiv.org/abs/2007.11262

[8] T. N. Moore, R. Q. Thomas, W. M. Woelmer, and C. C. Carey, "Integrating Ecological Forecasting into Undergraduate Ecology Curricula with an R Shiny Application-Based Teaching Module," *Forecasting*, vol. 4, no. 3, pp. 604–633, Sep. 2022, doi: 10.3390/forecast4030033.

[9] M. Li, "Teaching beginner-level computational social science: interactive open education resources with learnr and shiny apps," *Front. Educ. (Lausanne).*, vol. 8, 2023, doi: 10.3389/feduc.2023.1130865.

[10] J. L. Neyhart and E. Watkins, "An active learning tool for quantitative genetics instruction using R and shiny," *Natural Sciences Education*, vol. 49, no. 1, 2020, doi: 10.1002/nse2.20026.

[11] Q. Wang and X. Cai, "Active-Learning Class Activities and Shiny Applications for Teaching Support Vector Classifiers," *Journal of Statistics and Data Science Education*, vol. 32, no. 2, pp. 202–216, 2024, doi: 10.1080/26939169.2023.2231065.

[12] M. Dogucu, S. Kazak, and J. M. Rosenberg, "The Design and Implementation of a Bayesian Data Analysis Lesson for Pre-Service Mathematics and Science Teachers," *Journal of Statistics and Data Science Education*, vol. 33, no. 2, pp. 177–188, 2025, doi: 10.1080/26939169.2024.2362148.

[13] S. L. Wang, A. Y. Zhang, S. Messer, A. Wiesner, and D. K. Pearl, "Student-Developed Shiny Applications for Teaching Statistics," *Journal of Statistics and Data Science Education*, vol. 29, no. 3, pp. 218–227, 2021, doi: 10.1080/26939169.2021.1995545.

[14] H. Chen, M. Li, and A. M. Fernando, "WIP: Integrating Student-developed Applications and In-class Learning Games to Optimize Learning Outcomes: A Case Study in An Introductory Statistical Learning and Programming Course."

[15] G. Stevens, "Braun and Clarke's Approach to Thematic Analysis," *Academic Writing and Research*, 2024.

[16] J. Fereday and E. Muir-Cochrane, "Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development," *International Journal of Qualitative Methods* , vol. 5, no. 1, 2006, doi: 10.1177/160940690600500107.

Appendix

Table 1. R Shiny App repository used for Fall 2025 semester

Category	What the Apps support students in learning	Representative App examples
Distributions & parameter interpretation	How parameters change distribution shape and probabilities.	Distribution Visualizer; Exploring the Normal Distribution; Exploring Normal Distributions
Confidence intervals & uncertainty	How sample size, variability, and confidence level affect interval meaning and width; analytic vs bootstrap CIs.	Confidence Interval Visualizer – Means; Confidence Interval Simulator
Hypothesis testing & decisions	Test setup, p-values/critical regions, and the role of α; interpreting common tests.	The Null Hypothesis and P-Value; Interactive Hypothesis Testing App; Two-Sample t-Test; Understanding Chi-Square Densities
Regression, diagnostics, sensitivity	Fitting and interpreting models; reading diagnostics; effects of outliers/influence.	Linear Regression Model Visualization; Enhanced Linear Regression App; Impact of Outliers on Linear Regression
Discrete distributions & approximation	Binomial PMF/CDF behavior; when/why normal approximation (and continuity correction) works.	Binomial Distribution with Normal Approximation; Binomial Distribution Explorer
Sampling distributions	Sample mean behavior as sample size increases; interpreting histograms and Q-Q plots.	Central Limit Theorem Demo; Illustrating the Central Limit Theorem
Monte Carlo methods	Simulation-based estimation and convergence behavior.	Monte Carlo Visualizer

Table 3. Topic alignment between analyzed exam problems and R Shiny Apps for Summer 2025

Exam problem	Key topic assessed	Title of related R Shiny App used in class
Exam 1 (P1)	Probability distribution shape; numerical summaries	Exploring Normal Distributions
Exam 1 (P3)	Simple linear regression	Linear Regression Diagnostics/Outliers
Exam 2 (P1)	Normal distribution; numerical summaries	Exploring Normal Distributions
Exam 2 (P2)	Monte Carlo simulation	Monte Carlo Visualize, Roulette Simulation
Exam 2 (P3)	Confidence intervals and hypothesis testing with t distribution	Confidence Interval Visualizer, Interactive Hypothesis Testing
Exam 2 (P4)	Binomial distribution; normal approximation/continuity correction	Binomial Distribution & Normal Approximation
Exam 2 (P5)	Proportion confidence intervals	Binomial Distribution & Normal Approximation
Exam 2 (P6)	Proportion hypothesis testing	Interactive Hypothesis Testing
Exam 3 (P1)	Simple linear regression	Linear Regression Diagnostics/Outliers
Exam 3 (P2)	Chi-square goodness-of-fit	Chi-Square Goodness-of-Fit

Table 4. Student exam performance on R Shiny App topic-align problems in Summer 2025

Exam	Big Problem	001 Mean (SD)	002 Mean (SD)	Diff (002-001)	Welch's p-value
Exam 1	P1	24.21 (2.51)	22.88 (3.22)	-1.33	0.1278
Exam 1	P3	25.00 (0.00)	24.62 (1.36)	-0.38	0.1613
Exam 2	P1	10.00 (0.00)	10.00 (0.00)	0.00	NA
Exam 2	P2	18.05 (1.08)	17.73 (1.04)	-0.32	0.3222
Exam 2	P3	19.89 (0.46)	19.58 (1.17)	-0.32	0.2172
Exam 2	P4	9.74 (1.15)	9.62 (1.36)	-0.12	0.7473
Exam 2	P5	19.79 (0.63)	19.92 (0.39)	+0.13	0.4217
Exam 2	P6	20.00 (0.00)	18.00 (0.00)	-2.00	NA
Exam 3	P1	33.79 (0.63)	33.88 (0.43)	+0.10	0.5745
Exam 3	P2	10.84 (1.01)	11.58 (0.86)	+0.73	0.0150

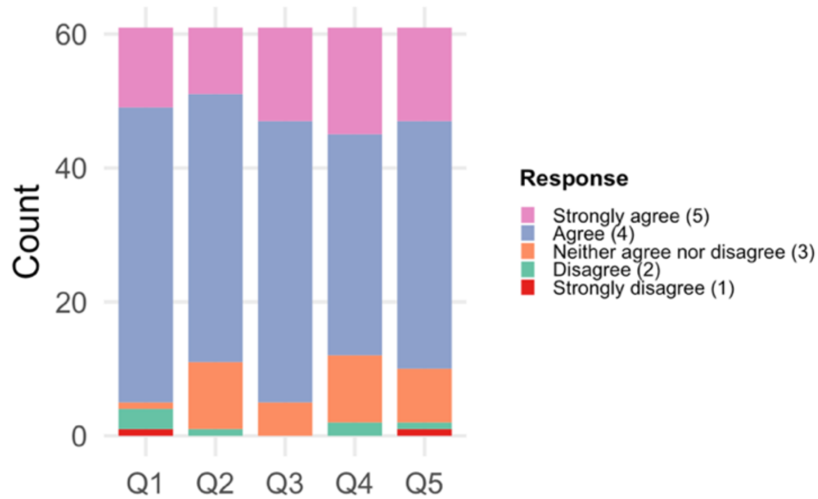


Figure 6. Response rating for a survey about R Shiny Apps list items provided in Fall 2025

<u>Note for Figure 6:</u>
Q1: "*The written descriptions made it clear what each app was designed to teach.*"
Q2: "*The list included at least one app that matched my interests or curiosity.*"
Q3: "*The app(s) we selected were reliable and ran without major errors.*"
Q4: "*Using apps created by previous students made the projects feel more like real work than using textbook or online examples.*"
Q5: "*The app we chose gave us a good starting point for making meaningful improvements.*"

Some feedback of what students understand better now because of their R Shiny App projects:

- *"Due to my Shiny App Work, I have better understood the influence of the significance level on overall t-tests. Also, having different sample sizes can change the outcome of a hypothesis test."*
- *"We did the linear regression model. I had a problem where I kept confusing which variable was response or predictor when relating to x or y, I figured it out while making the Shiny App because when I changed which variable was "response," the "y" axis updated on the graph."*
- *"I understand the impact of extreme outliers on the linear regression line more visually, as the app we worked on allowed us to quickly see the visual changes in the line after placing outliers."*
- *"Initially, I was having trouble visualizing skewness from a histogram. Implementing this in a shiny app helped connect that visual aspect that I was missing and playing around with the settings of the app finally allowed everything to come together."*
- *"I understand more now about null hypothesis and p-value. The null hypothesis represents a default assumption, usually that there is no effect or no difference, that we test against observed data. The p-value tells us how likely it is to see results as extreme as the ones observed if the null hypothesis were actually true. Through the Shiny app, adjusting inputs and immediately seeing how p-values change helped me understand why a small p-value provides evidence against the null hypothesis rather than "proving" an alternative outright."*