Analysis of class surveys

Part 1: Is self-reported statistics proficiency associated with the number of PSTAT courses completed?

Justin Zhou, Ella Yang, Lucy Cao, Cecilia Jiang, Janice Jiang, Wendy Zhu (Equal C

2025-10-19

Table of contents

1 Executive summary

For **Question 1**, we analyzed anonymized intake-survey responses from two sections to answer one question: **Is self-reported statistics proficiency associated with the number of PSTAT courses completed?** Using descriptive plots and a one-way ANOVA with Tukey's post-hoc test (implemented in `Question_1.Rmd`), we find evidence of a statistically significant association: respondents reporting higher statistics proficiency have, on average, completed more PSTAT courses. Because the responses are a **sample** of offered-seat students, we emphasize clear visuals and center-focused summaries appropriate for modest (n).

For **Question 2**, we analyzed anonymized intake-survey responses from two PSTAT 197A sections to answer one question: **Does self-reported programming proficiency increase with the number of CS courses completed?** Using descriptive visualizations and a simple linear regression model (implemented in `Question_2.Rmd`), we find a clear positive relationship: students who have taken more CS courses tend to report higher programming proficiency levels. Boxplots and scatterplots both indicate that advanced students generally have greater course exposure than intermediate or beginner peers, though some overlap suggests that informal or cross-disciplinary learning (e.g., PSTAT programming courses) also contributes to proficiency. Because the dataset represents a modest sample of enrolled students, our analysis focuses on clear graphical summaries and interpretable trends rather than formal inference.

1.1 Data description

- **How obtained.** Google Forms intake survey; exported and shared as two section files with identifiers removed plus a metadata file.

- **Sample.** Respondents from two course sections; treated as a **sample** from all offered-seat students (not necessarily a census).

- **Measurements.** Demographics; self-assessed **proficiency** and **comfort** (Likert 1–5); indicators for prior coursework; and project preferences.
  - For Q1, we use `stat.prof` (statistics proficiency; ordered beginner → intermediate → advanced) and `pstat_courses_total` (sum of PSTAT course indicators per student).
  - For Q2, we use `prog.prof` (programming proficiency; ordered beginner → intermediate → advanced), `prog.comf` and `cs_total` (sum of CS course indicators per student, including `CS9`, `CS16`, `CS130`, `CS165`, `CS5`, and programming-intensive PSTAT courses such as `PSTAT100`, `PSTAT131`, and `PSTAT134`).

1.2 Question of interest

1. **Q1.** Does self-reported statistics proficiency relate to the number of PSTAT courses completed?
2. **Q2.** How will the number of Computer Science courses that a student takes affects the students' programming proficiency?

2 Part 1: Is self-reported statistics proficiency associated with the number of PSTAT courses completed?

What we did. We summarized the distribution of proficiency levels (Beginner, Intermediate, Advanced) and compared total PSTAT coursework across groups using descriptive plots (composition chart, boxplot, and mean ± SD bar chart with a trend view). We then ran a one-way ANOVA and Tukey's HSD post-hoc comparisons using the code in `results/Question_1.Rmd`.

2.0.1 Proficiency composition

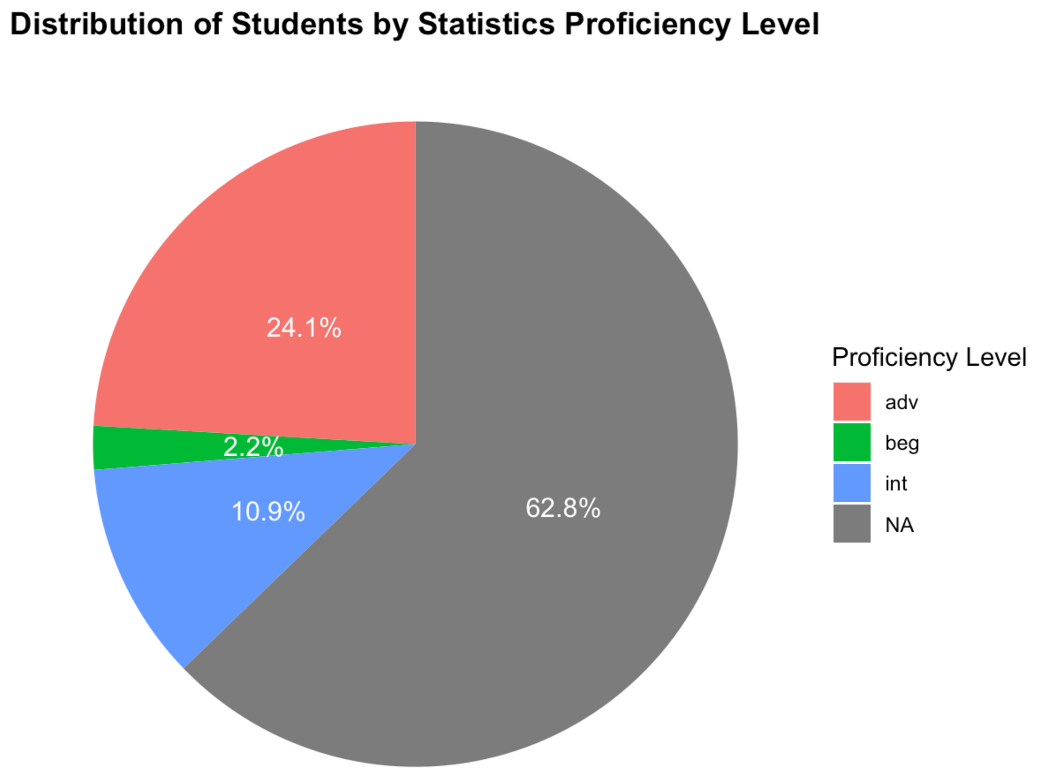


Figure 1: Distribution of Students by Statistics Proficiency Level

Most respondents did not record a proficiency level (NA is the largest slice). Among those who did, Advanced is more common than Intermediate, and Beginner is the smallest group. This matters for two reasons: (1) our comparisons are essentially between a larger Advanced group and smaller Intermediate/Beginner groups, and (2) inferential tests should exclude the NA category since it is missing-by-design.

2.0.2 PSTAT courses by proficiency (boxplot)

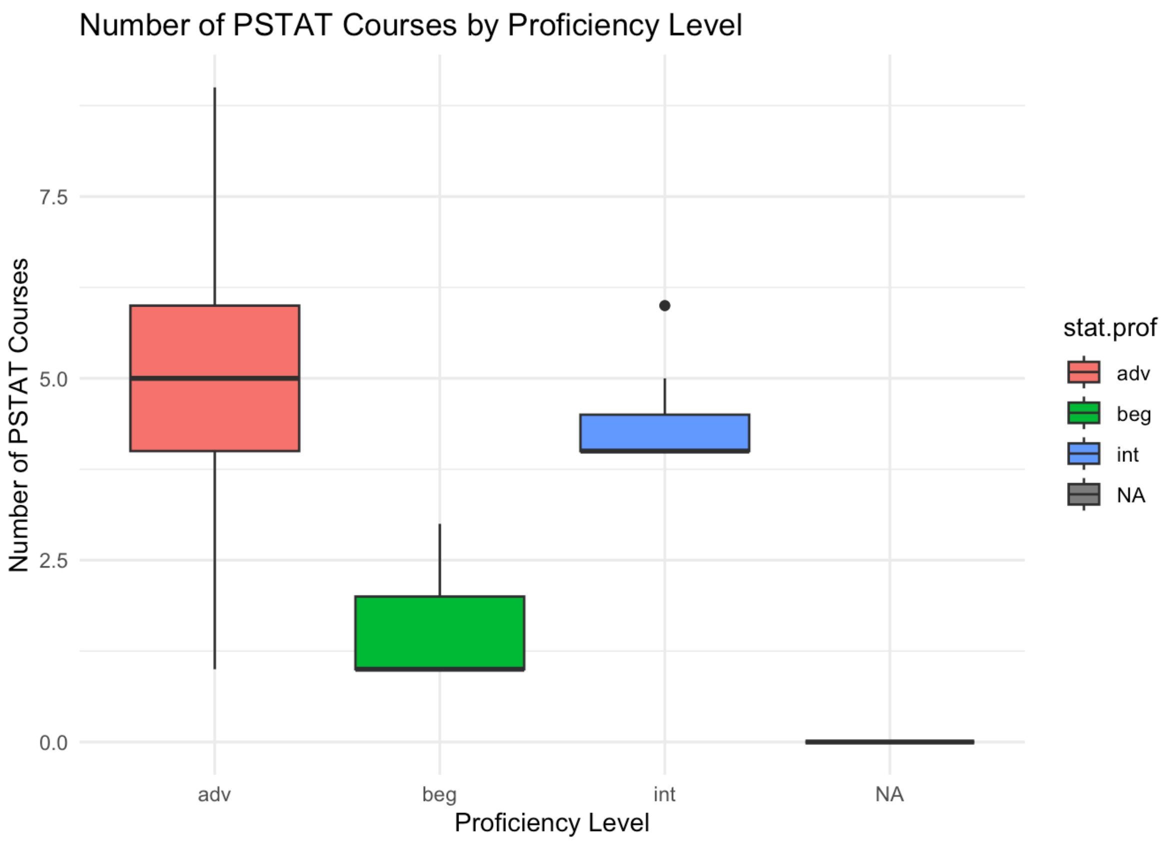
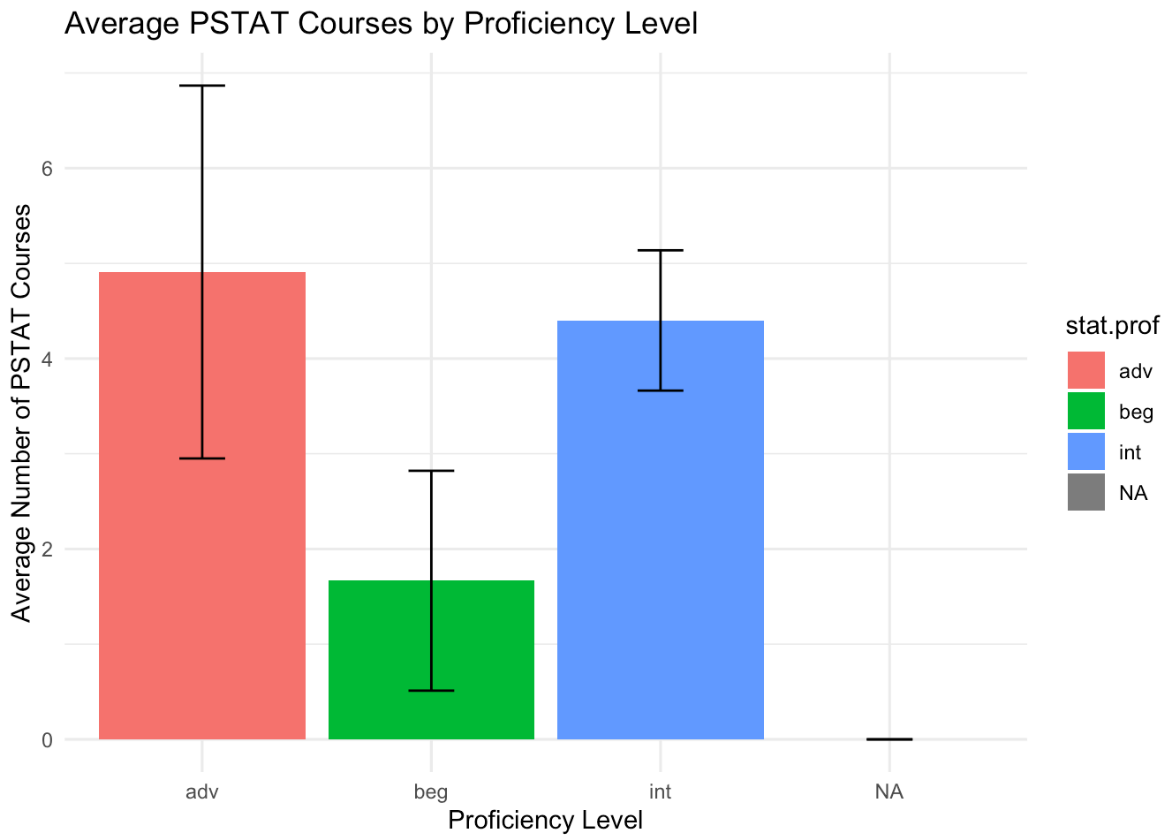


Figure 2: Number of PSTAT Courses by Proficiency Level

The center and spread of completed PSTAT courses shift upward from Beginner -> Intermediate -> Advanced. Beginners cluster at the low end; Intermediate has a higher median with modest spread; Advanced shows both a higher median and a longer right tail (some students have many courses). The NA group appears at zero because proficiency was not reported; we do not use NA in tests. Overlapping IQRs are limited, which is consistent with a real difference in central tendency.

2.0.3 Average PSTAT courses (mean ± SD)



Group means increase with proficiency: Beginners average the fewest PSTAT courses, Intermediate is higher, and Advanced is highest. Error bars ($±$1 SD) overlap somewhat between Intermediate and Advanced, suggesting those two groups are similar on average. This aligns with the Tukey HSD results: both Intermediate and Advanced differ from Beginners, while Intermediate and Advanced are not statistically different from each other.

2.0.4 Average PSTAT courses (trend)

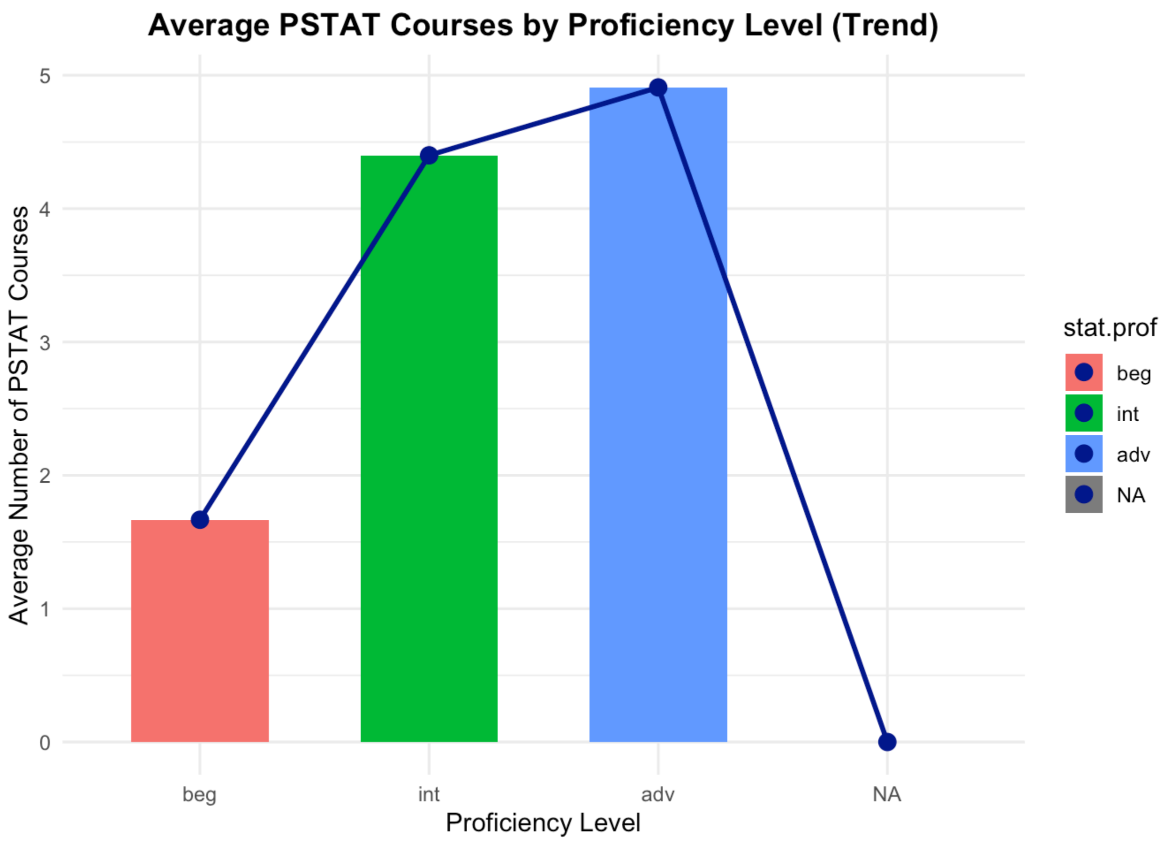


Figure 3: Average PSTAT Courses by Proficiency Level (Trend)

The trend line provides a compact view of the monotone relationship: more completed PSTAT courses are associated with higher self–reported proficiency. The drop to zero at NA reflects missing proficiency rather than a true value and should not be interpreted substantively. Overall, the trend corroborates the boxplot and mean $\pm$ SD views.

2.0.5 Inferential check (supports the visuals)

- **One-way ANOVA:** p = 0.008 (alpha = 0.05) -> mean PSTAT courses differ across proficiency groups.
- **Tukey's HSD pairwise comparisons:**

  - Advanced vs Beginner: p = 0.0062 (significant)
  - Intermediate vs Beginner: p = 0.033 (significant)
  - Intermediate vs Advanced: p = 0.59 (not significant)

Conclusion from tests: The main differences are between **Beginners** and the other two groups; **Intermediate** and **Advanced** are not statistically different in this sample.

2.0.6 Conclusion

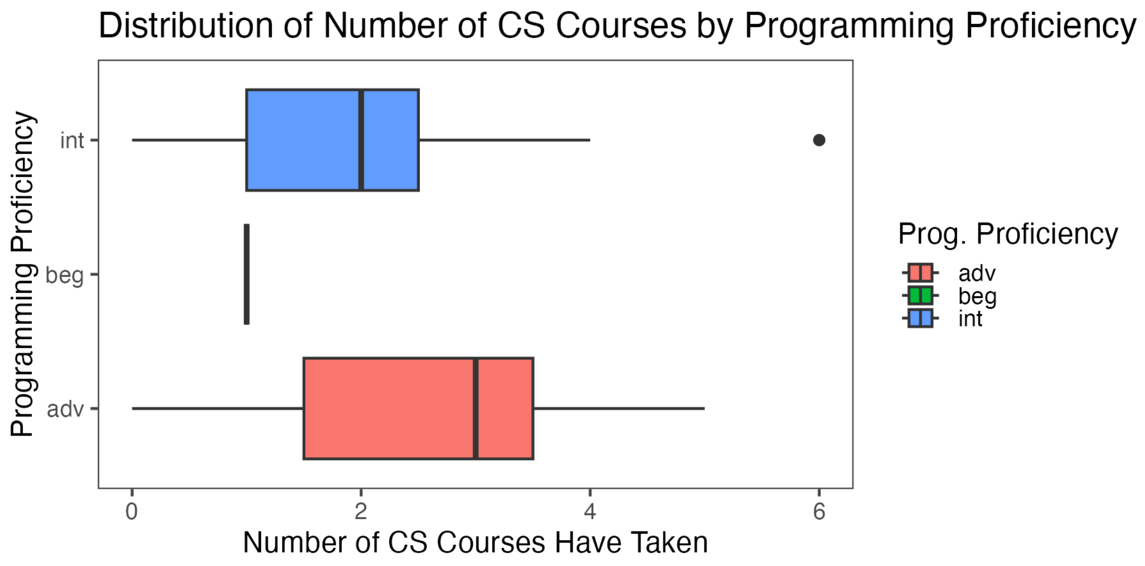
Within this sample of students offered a seat, **greater PSTAT coursework is associated with higher self-reported statistics proficiency**, especially when comparing **Beginners** to **Intermediate/Advanced**. Because this is a respondent sample with some item nonresponse, we treat results as **descriptive evidence** rather than a population claim.

Possible extensions. Report group counts and medians in the text; check robustness with a non-parametric Kruskal–Wallis test; examine whether particular PSTAT sequences (e.g., 120/122/126) align most with higher self-ratings.

3 Part 2: How will the number of Computer Science courses that a student takes affects the students' programming proficiency?

What we did. We summarized the distribution of programming proficiency levels (Beginner, Intermediate, Advanced) and compared total CS coursework (and alao some pstat cs related courses) across groups using descriptive visualizations (boxplot, grouped bar chart, and regression scatterplot with a fitted trend line). We then fit a simple linear model to quantify the relationship between the number of CS courses taken, programming comfort, and programming proficiency, using the code in `results/Question_2.Rmd`.

3.0.1 Relationship Between CS Coursework and Programming Proficiency



Description This boxplot displays the distribution of the total number of CS courses taken for each programming proficiency group (beginner, intermediate, advanced). The x-axis represents the number of CS courses taken, while the y-axis categorizes students by their self-reported proficiency.

Interpretation: From the graph, we can observe that:

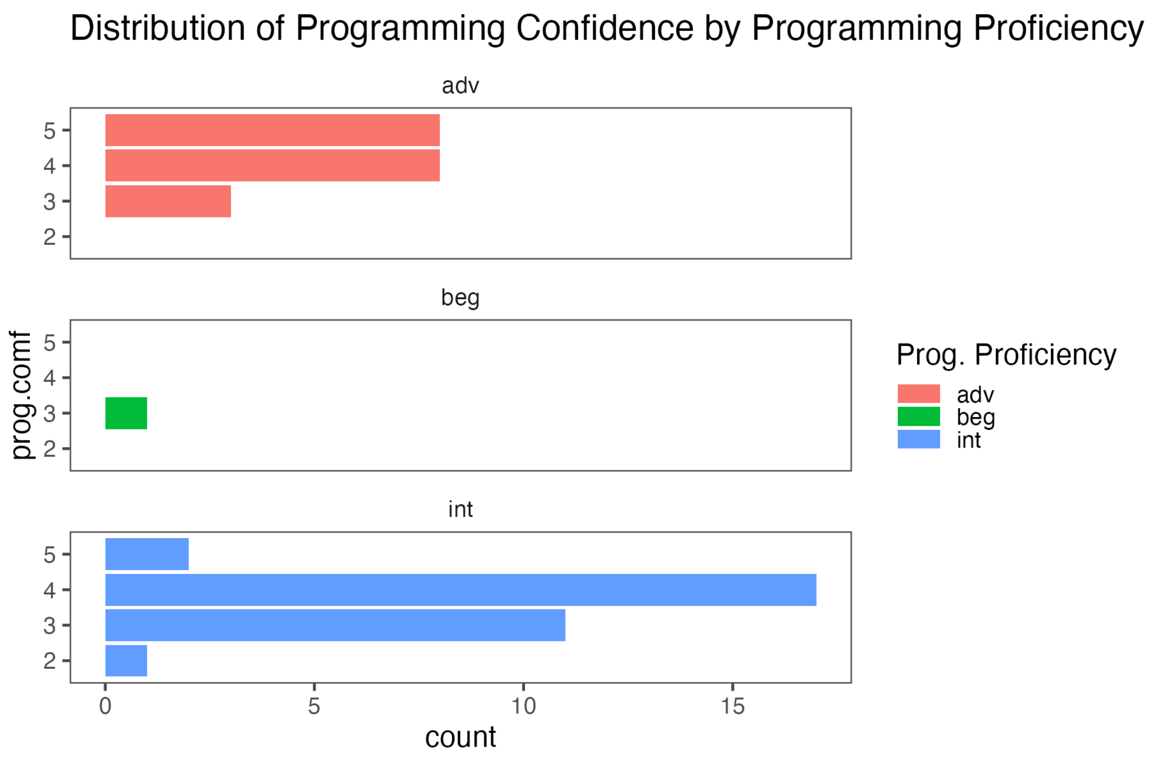
Advanced students (red) tend to have taken more CS courses on average than intermediate or beginner students. Their interquartile range is higher, and their median is around 2–3 courses.

Intermediate students (blue) show a slightly lower median but also a wider spread, ranging from 1 to 5 courses, with a few outliers who have taken 6 or more.

Beginners (green) have taken very few or no CS courses, with their distribution concentrated near 0–1.

Overall, this pattern **supports the hypothesis that taking more CS courses generally corresponds to higher programming proficiency.** However, the overlap between intermediate and advanced groups suggests that coursework alone does not fully determine skill level—experience outside formal classes might also influence proficiency.

3.0.2 Distribution of Programming Confidence by Proficiency Level



Description: This bar chart shows how students' self-reported confidence levels (1–5 scale) vary across different proficiency groups. Each bar indicates the count of students who selected a given confidence rating.

Interpretation: Students in the advanced group mostly report high confidence levels, typically 4 or 5, indicating consistency between their proficiency and self-assessment.

The intermediate group also shows a large cluster of high confidence scores (mostly 3–5), which suggests that many intermediate-level students also very comfortable with programming.

The beginner group is small, but its few members report lower confidence levels, typically 2–3, as expected.

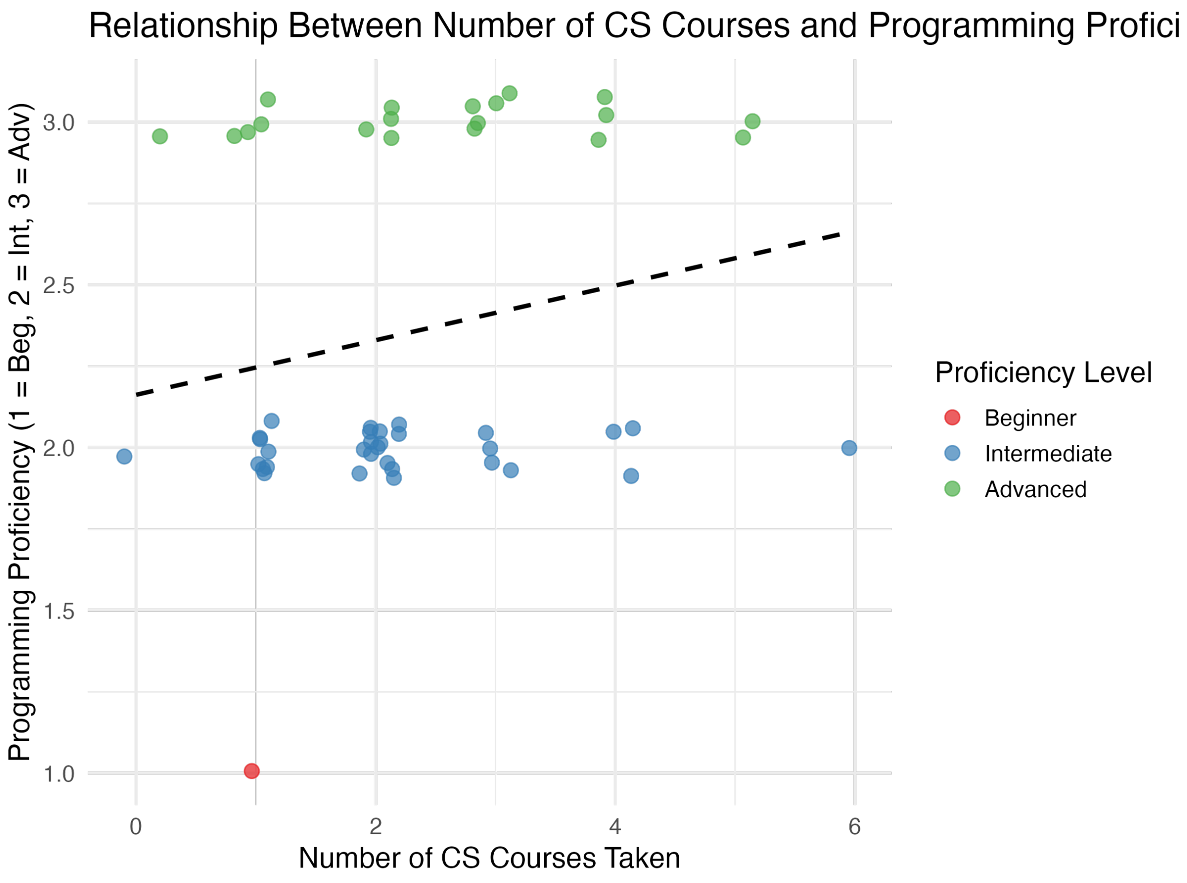
Analysis: The visualization shows that programming confidence generally increases with proficiency, but the relationship is not perfectly linear.

Advanced students show consistently high confidence, as expected.

Intermediate students' confidence overlaps considerably with the advanced group, indicating subjectivity in self-perceived skill levels.

The variation also highlights that confidence does not always correspond directly to technical ability—students' previous experiences (e.g., PSTAT courses, self-study, or non-CS exposure) may also shape how confident they feel when coding.

3.0.3 Linear Relationship between Number of CS Courses and Programming Proficiency



Description: This scatterplot displays the relationship between the number of CS courses taken (x-axis) and students' self-reported programming proficiency (y-axis, coded as 1 = Beginner, 2 = Intermediate, 3 = Advanced). Each dot represents one student, color-coded by proficiency group—red for beginners, blue for intermediates, and green for advanced. The black dashed line shows the fitted linear regression trend from the model:

$$Proficiency = \beta_0 + \beta_1(\text{Number of CS Courses}) + \epsilon$$

Interpretation: The plot shows a positive upward trend, suggesting that students who have taken more CS courses tend to report higher programming proficiency levels.

1.Trendline:

The dashed regression line slopes upward, indicating a positive coefficient for cs_total.

This means that, on average, each additional CS course is associated with a higher proficiency score.

2.Group Distribution:

Advanced students (green) cluster toward the upper part of the plot (proficiency   3) and typically have taken multiple CS courses, ranging roughly from 2 to 6.

Intermediate students (blue) form the majority and are spread across the middle range of CS course counts (1–4). Their distribution is relatively flat, reflecting that many have moderate experience regardless of exact course count.

Beginners (red) are few and positioned near the lower left—indicating both low proficiency and minimal course exposure.

3.Overlap & Variability:

Although the general trend supports a positive relationship, there is overlap between intermediate and advanced levels. For instance, some students with 3–4 CS courses still identify as intermediate, while a few advanced students have taken only 1–2.

This overlap suggests that while formal coursework contributes strongly to skill development, other learning pathways (e.g., self-study, statistics programming, or project experience) can also elevate proficiency.

Conclusion: This visualization reinforces the finding that coursework exposure is positively associated with programming proficiency, but not deterministically so. The modest slope of the regression line and the visible group overlap imply that programming ability is multifaceted—it reflects not only classroom experience but also self-learning, cross-disciplinary training (e.g., PSTAT courses), and students' personal engagement with coding practice.

3.0.4 Conclusion

Across all analyses for question two, we find consistent evidence that students who have completed more computer science courses tend to report higher levels of programming proficiency and confidence. The boxplot and regression model both show a positive association between coursework exposure and proficiency level, suggesting that structured academic training contributes meaningfully to skill development. However, the overlapping distributions among intermediate and advanced students, as well as the wide range of confidence levels within each group, indicate that programming ability is not determined by coursework alone. Factors such as self-study, prior experience in statistics or data science classes, and individual motivation also appear to shape both perceived and actual proficiency. Taken together, our findings highlight that while formal CS coursework builds a strong foundation, students develop programming expertise through a combination of classroom learning, applied practice, and independent exploration.

4 Final Conslusion

Taken together, our analyses show a consistent pattern: greater coursework exposure is associated with higher self-reported proficiency in both statistics and programming. In Question 1, **students who completed more PSTAT courses tended to rate themselves as more statistically proficient, a relationship confirmed through descriptive plots and one-way ANOVA tests.** Similarly, in Question 2, **students who had taken more CS courses generally reported higher programming proficiency and confidence, as shown through boxplots and a positive linear regression trend.**

However, across both domains, our findings also reveal meaningful overlap between proficiency groups, suggesting that coursework alone does not fully explain skill development. Students with fewer formal courses sometimes demonstrated high proficiency, likely reflecting experience gained through independent learning, applied projects, or related quantitative classes. Overall, the results highlight that while formal coursework provides a strong foundation for technical growth, true proficiency emerges from a combination of structured study, interdisciplinary exposure, and self-directed practice.

5 Work distribution

Our group divided the tasks based on individual focus areas to ensure an efficient and balanced workflow.

Ella & Cecilia: Analyzed the relationship between the number of PSTAT courses taken and students' statistical proficiency. They cleaned the dataset, computed the total number of PSTAT courses per student, and visualized the corresponding proficiency distributions.

Janice & Wendy: Investigated how the number of CS courses taken affects students' programming proficiency. They generated summary statistics, developed regression and visualization models (including boxplots and scatterplots), and interpreted the results.

Lucy & Justin: Edited the final report, ensuring consistency in format, clarity, and interpretation. They also wrote the executive summary and refined the overall discussion to connect the two sub-questions into a cohesive conclusion.