Where do students struggle on programming tasks in Data Science 101?

Aimee Schwab-McCoy
Senior Manager, Content Authoring
Data Science, Mathematics, and Statistics
zyBooks (A Wiley Brand)
aimee.schwab-mccoy@zybooks.com

Introduction

zyBooks is an online courseware platform offering interactive, web-native textbooks in computer science, data science, engineering, information technology, mathematics, and statistics. Since 2022, zyBooks has offered three versions of an introductory data science book, Data Science Foundations (DSF): Python, R, and non-programming.

In 2024-2025, DSF was used at more than 140 colleges and universities, primarily in the US and Canada. Given this diverse set of students and courses, student performance in DSF gives unique insights into learning in data science at a broader scale.

Challenge activities

Challenge activities (CAs) are used throughout the DSF zyBook to assess student learning at the end of a section. CAs are highly randomized and require students to demonstrate mastery on a concrete task. CAs are tied to section learning goals, and students must pass one level in a CA to progress to the next.

In DSF, two types of CAs are available:

1. Conceptual CAs require students to complete a non-programming task, such as a "by hand" calculation, interpreting a graph, or describing patterns in a dataset. Conceptual CAs are identical in all versions of DSF.
2. Coding CAs require students to complete a programming task using Python or R. Coding CAs are similar, but not identical, in the Python and R versions of DSF.

For this analysis, coding CAs were reviewed and paired based on similar task and complexity. For example, the coding tasks below ask students to create a contingency table from a dataframe.

```python
#Python
freqTable = df.groupby(by=["Total_calls_made"]).size()
```

```r
#R
freqTable <- df |>
  group_by(Total_calls_made) |>
  summarize(n=n())
```

Since the functions and workflow are similar, the CA levels are matched.

Metrics

On the zyBooks platform, we measure student struggle using two metrics:

1. Time spent: Time elapsed between a student's first attempt on a CA and the correct submission. Submission gaps of ten minutes or more are excluded to account for breaks in student activity. Higher time spent on a programming task suggests more struggle.

2. Number of attempts: The number of attempts until a correct submission. More attempts until a correct submission suggests more struggle.

Struggle metrics are recorded at the student level, and aggregated across all courses using the latest releases of DSF between September 2024 and June 2025. Coding tasks with high struggle metrics are labeled as "difficult" and flagged for improvement in a later edition. All student data is completely anonymous and limited to data available in the zyBooks platform.

Limitations

Overall, we see more struggle from students in the R version of DSF than the Python version, especially for mean time spent. But, that doesn't mean R is harder than Python!

- Not every instructor assigns every CA. So, sample sizes vary from 20 students to 2000+ per activity. We dropped CA levels with fewer than 20 students in either version from our analysis.
- The majority of our data science courses are taught in CS departments, which prefer Python. Outliers have a greater impact on struggle metrics in R than Python, since sample sizes are smaller.

We love working with faculty! Let us know how we can support your research in data science, mathematics, and statistics education.

zyBooks
A Wiley Brand

In both Python and R, DS 101 students tend to struggle most when a new syntax or library is first introduced.

Since syntax varies between Python and R, students may struggle at different points in your course depending on what language you're teaching with.

Number of attempts

We consider a task "difficult" if the mean number of attempts to successful completion is greater than three. Tasks that were difficult in one or more language are grouped by category below.

Difficult in R

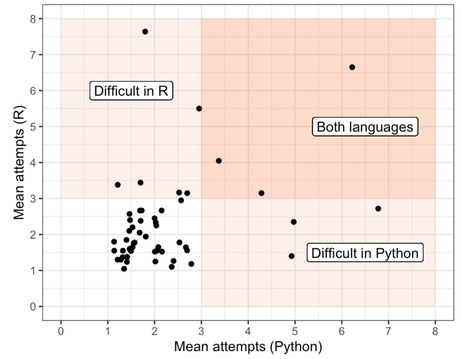
- Probability and statistics: Calculate binomial probabilities, two-sample t-test Data wrangling: Contingency tables
- Modeling: Fit linear regression, fit logistic regression, fit k-nearest neighbors

Difficult in Python

- Probability and statistics: One-sample proportion test
- Modeling: Define cross-validation folds, fit PCA and calculate eigenvalues

Difficult in both

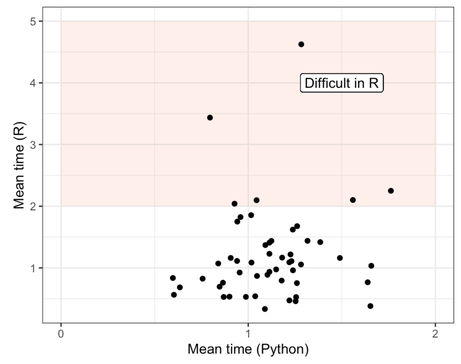
- Data wrangling: Frequency tables, imputation
- Modeling: Find predicted values and slope for linear regression



Time until completion

We consider a task "difficult" if the mean time spent is greater than two minutes. No tasks were difficult in Python based on time spent, but some were difficult in R.

- Probability and statistics: Calculate binomial probabilities, one-sample proportion test
- Data wrangling: Generate bootstrap samples, normalize input features, subset a dataframe
- Modeling: Fit a support vector machine



Where do we see most students struggle?

Overall, students in intro data science courses tend to struggle with:

1. Probabilities and hypothesis testing. These topics are covered only briefly in many data science courses, and are widely accepted as difficult topics in statistics courses. So, students struggling here is not a surprise.
2. Contingency and frequency tables. In both versions of DSF, contingency and frequency tables are some of the earliest introductions to data wrangling packages like `pandas` and `dplyr`.
3. Model fitting. Students tend to struggle more with early modeling tasks - linear regression, logistic regression, and k-nearest neighbors - vs. later tasks.

In our data, programming tasks with more struggle tend to coincide with early introductions of a specific syntax. Tasks using `dplyr` and `tidymodels` tend to take more time in R, but not necessarily more attempts.

Is struggle bad?

No! We want students to work at their own pace. Students may struggle more on topics that weren't covered in class, or on tasks requiring longer code snippets. Some tasks are just difficult, and that's okay.