Modern Concepts in Python: Spring 2026

by Eric Rying

Module 3: Database Queries Assignment Experiment

Feb 2026

Written Assignment Requirements

Having carried out this assignment, please write two paragraphs about the inherent limitations of carrying out analytics over anonymously submitted data items. Did the analytic responses surprise you? How does this different from standards? For example, the average GRE quantitative reasoning score was

157 for 2023-2023 and was nearly 165 for grad school entries submitted (see sample output). Why do you think that is? What might cause this to occur? Please place your essay into a file called limitations.pdf

Answer:

Limitations of the GradCafe Data and Scraping Pipeline

Analytics performed on anonymously submitted data are inherently constrained by issues of data quality, completeness, and representativeness. Because participation is voluntary and unverified, users choose what information to share, how to format it, and whether to report sensitive metrics such as GPA or GRE scores. This leads to inconsistent coverage across fields, missing values, and occasional exaggeration or misreporting. The anonymity of platforms like GradCafe also introduces self-selection bias: applicants who post tend to be more anxious, more stats-focused, or more competitive than the general applicant pool. As a result, the dataset reflects the behavior and characteristics of people who choose to self-report online, not the broader population of graduate applicants. These limitations mean that any statistical patterns observed should be interpreted as informal community signals rather than rigorous or generalizable findings.

The analytic results did not surprise me once these biases were considered, but they differ sharply from standardized national metrics. For example, the average GRE Quantitative score in my scraped dataset was close to 165, far above the ETS national mean of roughly 157. This inflation is expected: applicants with unusually strong scores are more likely to post them publicly, while those with weaker scores often omit them or choose not to post at all. Additionally, many programs have made the GRE optional, so the only people reporting GRE scores on GradCafe tend to be those who performed well and feel confident sharing them. This combination of self-selection, survivorship bias, and incomplete reporting naturally pushes the averages upward. These patterns illustrate why analytics over anonymously submitted data must be interpreted cautiously: the dataset reflects who choos to contribute to GradCafe, not the full population.

A significant technical limitation emerged during data collection: GradCafe does **not** provide a functional API for retrieving detailed applicant information from what I could find in my testing. Although a /api/result/{id} endpoint is documented online, all attempts to access it returned HTML error pages rather than JSON, indicating that the API is deprecated or inaccessible. To obtain detail fields such as GPA, GRE scores, citizenship, and application term, the scraper was forced to rely on **HTML parsing and regex extraction** from individual result pages. This approach is inherently fragile because the HTML structure varies across entries, and many users omit these fields entirely. As a result, coverage varied significantly by field:

- GPA achieved 57% coverage (578 of 1,000 entries),
- citizenship status reached 99%,
- while GRE scores remained sparse at 3-6% coverage.

This variability reflects both the inconsistent HTML structure across pages and users' selective reporting—many applicants omit standardized test scores, particularly as more programs adopt test-optional policies. While this level of completeness is sufficient for exploratory analysis, it limits the statistical strength of any conclusions drawn.

One field—**application term (Fall/Spring + year)**—could not be reliably extracted at all. GradCafe's HTML pages do not present term information in a consistent or machine-readable format, and no regex pattern produced dependable matches. Consequently, all entries contain null values for the term field. This prevents term-specific filtering (e.g., "Fall 2026 applicants"), and several assignment questions that depend on term segmentation necessarily return **N/A**. This limitation highlights a broader challenge in web scraping: when data sources are unstructured and user-generated, certain fields may simply be unavailable regardless of scraper sophistication.