## ­­EduViz – Lean Approach

This project lends itself well to *Lean.* The focus on producing code in quick cycles provides an enormous advantage—the opportunity to test, elicit feedback, and debug code quickly without waiting for other tasks to be completed. EduViz is a deeply involved project with strong collaboration between statisticians and developers—this works well with Lean’s emphasis on shared understanding throughout a project.

The idea of *work, learn, adjust* goes hand-in-hand with this visualization initiative. Most of the early stages of data visualization are for exploratory purposes (e.g., to understand the dataset). This gives rise to an opportunity for developers and team members to learn and understand the dataset together, and make appropriate adjustments in visualizations and organization of information as they arise.

## Lean Principles of Interest

In adapting this project to *Lean,* some of the Lean Principles stand out more than other. This section focuses on the most relevant and striking principles as they apply to the project.

### Cross-Functional Teams

The critical players that drive creation of deliverables are the statisticians and the developers. The initial, Waterfall-esque idea of a “hand-off” of datasets or code will not suffice. Rather, statisticians and programmers need to work together; it is critical that the programmers understand the dataset, and that the statisticians understand the way their data will be manipulated and transformed.

### Shared Understanding

On the whole, this project demands unparalleled skills in data analysis, programming, and visualization. It is impossible for any one team member to be an expert in every realm. Having all data, project files, and code available for all team members is integral. Using services like *GitHub*, and *iPython Notebook,* members of the team can more readily discuss and target technical facets of the project. For instance, a statistician might say “I have a question about lines 240-255 in the *vis.js* file.”

A programmer can then pull up the code and discuss it with relative ease, rather than trying to give a long-winded explanation full of computer jargon that might not be comprehensible or user to the statistician.

### Small Batch Size and the “cupcake vs. cake” philosophy

Given the gargantuan dataset (17 years, tens of thousands of rows and 1700 variables), small batch size is essential to making measurable progress. Rather than releasing a tremendous dataset to a hapless programmer, it is sensible to divide the data into a more manageable unit for inspection. For instance, the statisticians may choose to begin with one year of data, and one set of related variables (perhaps completion rates, repayment rates, etc.)

This allows our data viz programmers to generate a smaller visualization in less time. The advantage of this is that:

1. It helps statisticians to understanding a critical component of their dataset
2. It gives a simple visualization that can be more easily understood by members of the team, stakeholders, and the customers
3. It gives initiative and direction to which data elements should be explored next