InsightVis: Peer Review #1
Background - The class

- CPSC 310 is a project-heavy course, and a requirement of the Computer Science Major
- Roughly 180 or 360 students per term working in pairs, meaning we have 90 to 180 teams
Background - The project

- Students are tasked to build a simple data storage and query language system
- Students are marked by their project’s ability to pass a suite of automated tests
Background - The data

- We have records of test results for all the students commits (100MB for one term)
- We also have their git repositories, which means entire project histories (separately on GitHub)
- One major challenge will be pre-processing this data
Target use cases

- Enable course staff/TAs to get an overview of how the class is doing
- Identify exceptional or struggling teams
- Find correlations between project qualities and performance
- Provide an overview of how an individual team is doing to guide TA help
- (Stretch) Learn more about the test suite used in the course to identify improvements
Class and Individual views
Test View

Test 1
Test 2
Test 3

Test pass (Teams)

Test fail (Teams)

Test skipped (Teams)

Sort option: Pass/fail/skipped
Process Data
Extract data containing D1 & D2 because these deliverables consist of writing test cases

Data Derivation
Derive regression score

Clone project repos
Clone all the project repos to extract various commit metrics and derive more data.

Read result.json
Contains data about all the deliverables D1, D2, D3 & D4

Final data format
Structure the original & derived data to specific format for visualization using d3.js
Derived Data: Regression Score

Example for Team A & deliverable D1:

Commit #1  Passed Tests: [Test1, Test2, Test3]  Failed Tests: [Test4, Test5, Test6]

Regression Score = number of times a test went from pass to fail or vice versa / number of commits

Regression Score = (2 + 2) / 2 = 2
Challenges

- Did not have access to Github API in the old github.ugrad.com server
- Downloaded all the project team repos
- Extracted all the commit metrics using git command by looping through all project directory
- Our initial data does not save deliverables end time so manually went over commits in github to approximate end times for deliverables