We explored trends in student programming assignment data. In our first three programming courses (CS1/CS2/CS3), collected from electronic submissions using Web-CAT, an open-source automated grading tool. Web-CAT provides rapid feedback on style, coding conventions, and correctness. It allows the option of grading students on how thoroughly they test their own code. Students typically go through many edit/submit/revise cycles on each assignment.

5 years (10 semesters) of the first three courses. Every submission (89,879) by every student. 1,101 different students, 105 different assignments, 6,001 program assignment grades.
Factors of interest

- When did they start submitting?
- When did they finish?
- Total elapsed time?
- How many submission attempts?
- How much code did they write?
- How thoroughly did they test?
- How did they score?
- ... earlier investigations on similar issues exist (Jadud, ClockIt, Retina, ...)

We began with clustering

- A hierarchical clustering analysis
- Using factors on previous slide
- The top-level split in the dendogram partitioned students by assignment score at approximately 80%

We then looked for differences between students scoring ≥ 80% and those scoring less

Overall score distribution

Submission attempts are similar
Finally, they test more thoroughly
• Using a composite coverage measure (methods + statements + decisions)
• A/B solutions averaged: 94.2%
• C/D/F solutions averaged: 79.7%
• F(502,1598) = 2.50, p < .0001
• 0.78 σ
• A/B-only group: 93.3%
• C/D/F-only group: 70.8%

And also at initial submissions
• Looking at how close the student is to their final solution (in size):

<table>
<thead>
<tr>
<th>Group</th>
<th>Proportion of solution code present at first try</th>
<th>Proportion of test code present at first try</th>
<th>Test coverage achieved on first try</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B-only</td>
<td>89.5%</td>
<td>73.1%</td>
<td>68.4%</td>
</tr>
<tr>
<td>A/B</td>
<td>90.1%</td>
<td>75.5%</td>
<td>70.0%</td>
</tr>
<tr>
<td>C/D/F</td>
<td>87.5%</td>
<td>75.4%</td>
<td>55.5%</td>
</tr>
<tr>
<td>C/D/F-only</td>
<td>89.8%</td>
<td>83.9%</td>
<td>50.2%</td>
</tr>
</tbody>
</table>

So we looked at code size again
• Looking at normalized solution code size:

<table>
<thead>
<tr>
<th>Group</th>
<th>Solution Size</th>
<th>Size of Tests</th>
</tr>
</thead>
<tbody>
<tr>
<td>A/B-only</td>
<td>1.02</td>
<td>1.12</td>
</tr>
<tr>
<td>A/B</td>
<td>1.01</td>
<td>1.01</td>
</tr>
<tr>
<td>C/D/F</td>
<td>0.98</td>
<td>0.90</td>
</tr>
<tr>
<td>C/D/F-only</td>
<td>0.87</td>
<td>0.73</td>
</tr>
</tbody>
</table>

We also looked at score trends as the deadline approaches

Different behaviors emerge if you look at instantaneous scores

The “dip” appears to be due to late starters
Conclusions

- Larger scope than previous studies
- Removed the high and low performers from the equation
- When students received higher scores, they started earlier and finished earlier
- They also wrote more code, and tested it more thoroughly
- We've only scratched the surface of the data at this point