

The elephant in the room

Richard Bornat and Saeed Dehnadi
School of Computing Science, Middlesex University

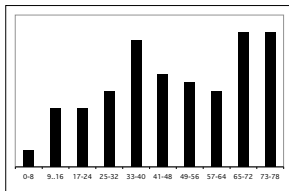
21st February 2007



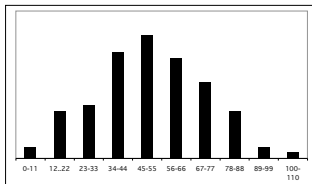
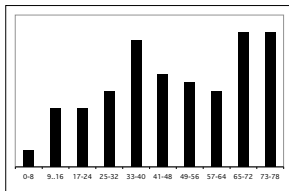
The double hump



The double hump



The double hump



Programming teaching is hard



Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.



Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).



Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).
- ▶ In particular, maths A-level is a weak predictor of success.

Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).
- ▶ In particular, maths A-level is a weak predictor of success.
- ▶ This has been a problem since the subject started



Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).
- ▶ In particular, maths A-level is a weak predictor of success.
- ▶ This has been a problem since the subject started
- ▶ – and we've tried **everything**.

Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).
- ▶ In particular, maths A-level is a weak predictor of success.
- ▶ This has been a problem since the subject started
- ▶ – and we've tried **everything**.
- ▶ We aren't teaching the wrong stuff.

Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).
- ▶ In particular, maths A-level is a weak predictor of success.
- ▶ This has been a problem since the subject started
- ▶ – and we've tried **everything**.
- ▶ We aren't teaching the wrong stuff.
- ▶ We aren't teaching it the wrong way (beware the Deductive Fallacy).

Programming teaching is hard

- ▶ It appears that some people can learn, and others cannot.
- ▶ We are unable to select university entrants, in any university, by previous educational attainment (i.e. A levels).
- ▶ In particular, maths A-level is a weak predictor of success.
- ▶ This has been a problem since the subject started
- ▶ – and we've tried **everything**.
- ▶ We aren't teaching the wrong stuff.
- ▶ We aren't teaching it the wrong way (beware the Deductive Fallacy).
- ▶ And although our students are still adolescent, motivation isn't the real problem.

Programming is utterly formal



Programming is utterly formal

- ▶ The meaning (denotation) of a program is irrelevant to its execution.

Programming is utterly formal

- ▶ The meaning (denotation) of a program is irrelevant to its execution.
- ▶ Only the meaning as calculation-elaboration counts.



Programming is utterly formal

- ▶ The meaning (denotation) of a program is irrelevant to its execution.
- ▶ Only the meaning as calculation-elaboration counts.
- ▶ This is like formal logical proof, like nothing else, and is **extremely** abstract.

There are no good predictors



There are no good predictors

- ▶ Wilson and Shrock found ‘comfort level’, maths background and attribution of success to luck were good predictors (the last negative).



There are no good predictors

- ▶ Wilson and Shrock found ‘comfort level’, maths background and attribution of success to luck were good predictors (the last negative).
- ▶ Extensive game playing was a negative predictor.

There are no good predictors

- ▶ Wilson and Shrock found ‘comfort level’, maths background and attribution of success to luck were good predictors (the last negative).
- ▶ Extensive game playing was a negative predictor.
- ▶ Simon et al. found a deep approach to learning a predictor, and some evidence for map-drawing style.

There are no good predictors

- ▶ Wilson and Shrock found ‘comfort level’, maths background and attribution of success to luck were good predictors (the last negative).
- ▶ Extensive game playing was a negative predictor.
- ▶ Simon et al. found a deep approach to learning a predictor, and some evidence for map-drawing style.
- ▶ But the Wilson & Shrock result was mostly post-hoc;

There are no good predictors

- ▶ Wilson and Shrock found ‘comfort level’, maths background and attribution of success to luck were good predictors (the last negative).
- ▶ Extensive game playing was a negative predictor.
- ▶ Simon et al. found a deep approach to learning a predictor, and some evidence for map-drawing style.
- ▶ But the Wilson & Shrock result was mostly post-hoc;
- ▶ Simon’s predictors are weak (as is maths achievement);

There are no good predictors

- ▶ Wilson and Shrock found ‘comfort level’, maths background and attribution of success to luck were good predictors (the last negative).
- ▶ Extensive game playing was a negative predictor.
- ▶ Simon et al. found a deep approach to learning a predictor, and some evidence for map-drawing style.
- ▶ But the Wilson & Shrock result was mostly post-hoc;
- ▶ Simon’s predictors are weak (as is maths achievement);
- ▶ and I have never seen an ‘aptitude test’ for programming that is more than a general IQ test.

Programming hurdles



Programming hurdles

- ▶ assignment and sequence seem trivial, but trip up many;

Programming hurdles

- ▶ assignment and sequence seem trivial, but trip up many;
- ▶ choice is not much of an additional hurdle;



Programming hurdles

- ▶ assignment and sequence seem trivial, but trip up many;
- ▶ choice is not much of an additional hurdle;
- ▶ iteration / recursion is hard, and is the cliff at which the remaining failures pause;

Programming hurdles

- ▶ assignment and sequence seem trivial, but trip up many;
- ▶ choice is not much of an additional hurdle;
- ▶ iteration / recursion is hard, and is the cliff at which the remaining failures pause;
- ▶ concurrency is reached by only a tiny percentage of programmers in their lifetime.



We decided to keep it simple



We decided to keep it simple

- ▶ It ought to be possible to find out the mental models that are used in assignment and sequence, and track them over time.

We decided to keep it simple

- ▶ It ought to be possible to find out the mental models that are used in assignment and sequence, and track them over time.
- ▶ Saeed Dehnadi's inspiration was to look at those mental models **before** instruction begins.



We decided to keep it simple

- ▶ It ought to be possible to find out the mental models that are used in assignment and sequence, and track them over time.
- ▶ Saeed Dehnadi's inspiration was to look at those mental models **before** instruction begins.
- ▶ Surprisingly, there was a regularity: some used the same mental model throughout the test; some changed models according to their interpretation of the question; some refused to answer.

We decided to keep it simple

- ▶ It ought to be possible to find out the mental models that are used in assignment and sequence, and track them over time.
- ▶ Saeed Dehnadi's inspiration was to look at those mental models **before** instruction begins.
- ▶ Surprisingly, there was a regularity: some used the same mental model throughout the test; some changed models according to their interpretation of the question; some refused to answer.
- ▶ Which group did best in the exam?

Consistency counts

Consistent subjects, who used the same (mechanical) mental model of assignment and sequence no matter what the question, did markedly better.



Consistency counts

Consistent subjects, who used the same (mechanical) mental model of assignment and sequence no matter what the question, did markedly better.

Almost all consistent subjects pass; about half of non-consistent subjects fail.



Consistency counts

Consistent subjects, who used the same (mechanical) mental model of assignment and sequence no matter what the question, did markedly better.

Almost all consistent subjects pass; about half of non-consistent subjects fail.

This is a very strong result. We are only beginning to understand it and its implications.