

# Measuring Difficulty of Introductory Programming Tasks

Tomáš Effenberger

Jaroslav Čechák

Radek Pelánek

*Masaryk University Brno, Czech Republic*

## Why study difficulty?

- improved student skill estimates
- task sequencing
- content authoring and management

## Obstacles

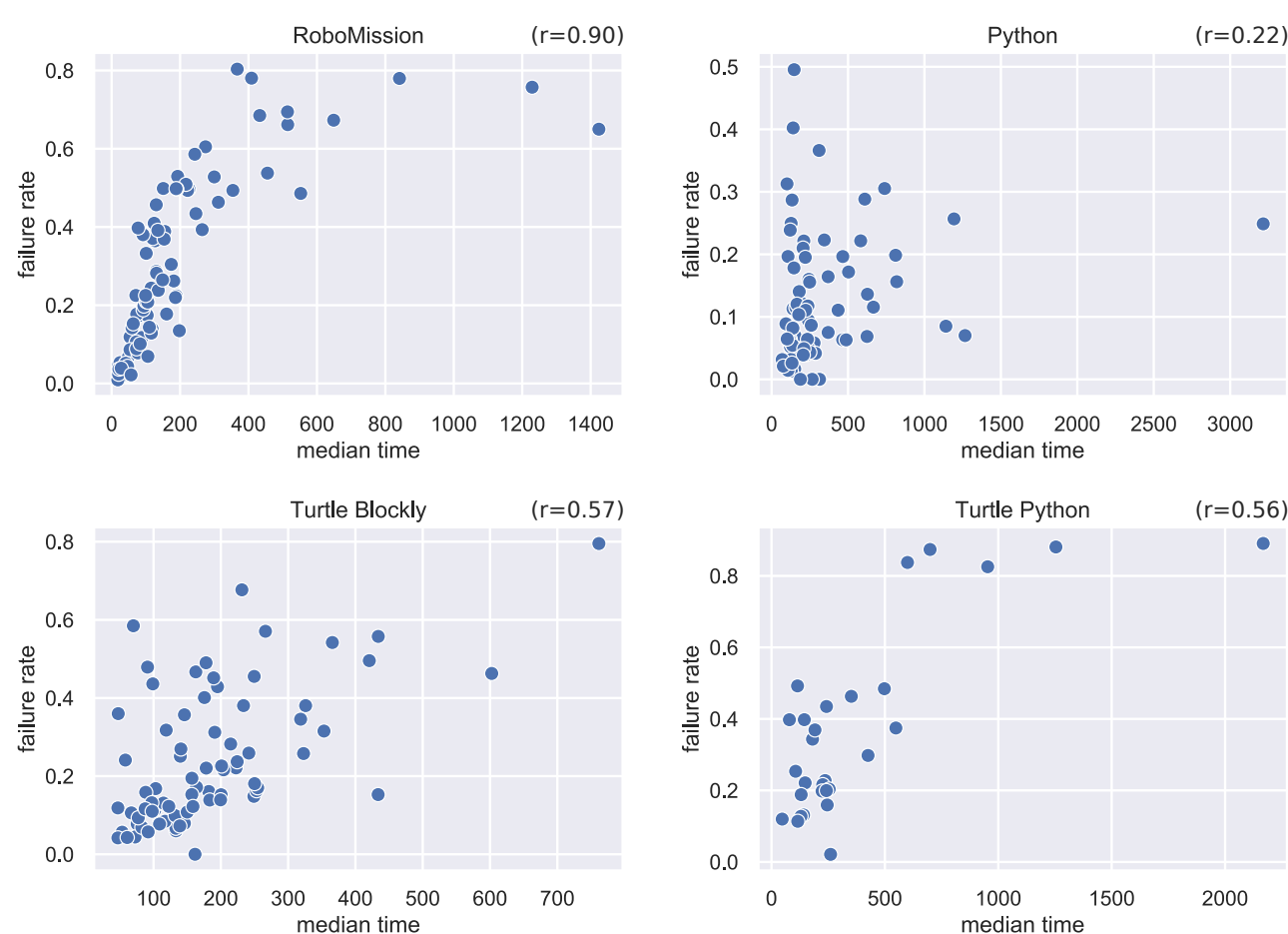
- many measures
- biases in data collection
- variations across exercises

## Data

Exercise	Interface	Tasks	Students	Attempts
RoboMission	blocks	85	3,800	62,500
Turtle Blockly	blocks	77	11,000	63,600
Turtle Python	text	51	2,400	11,900
Python	text	73	2,000	10,700

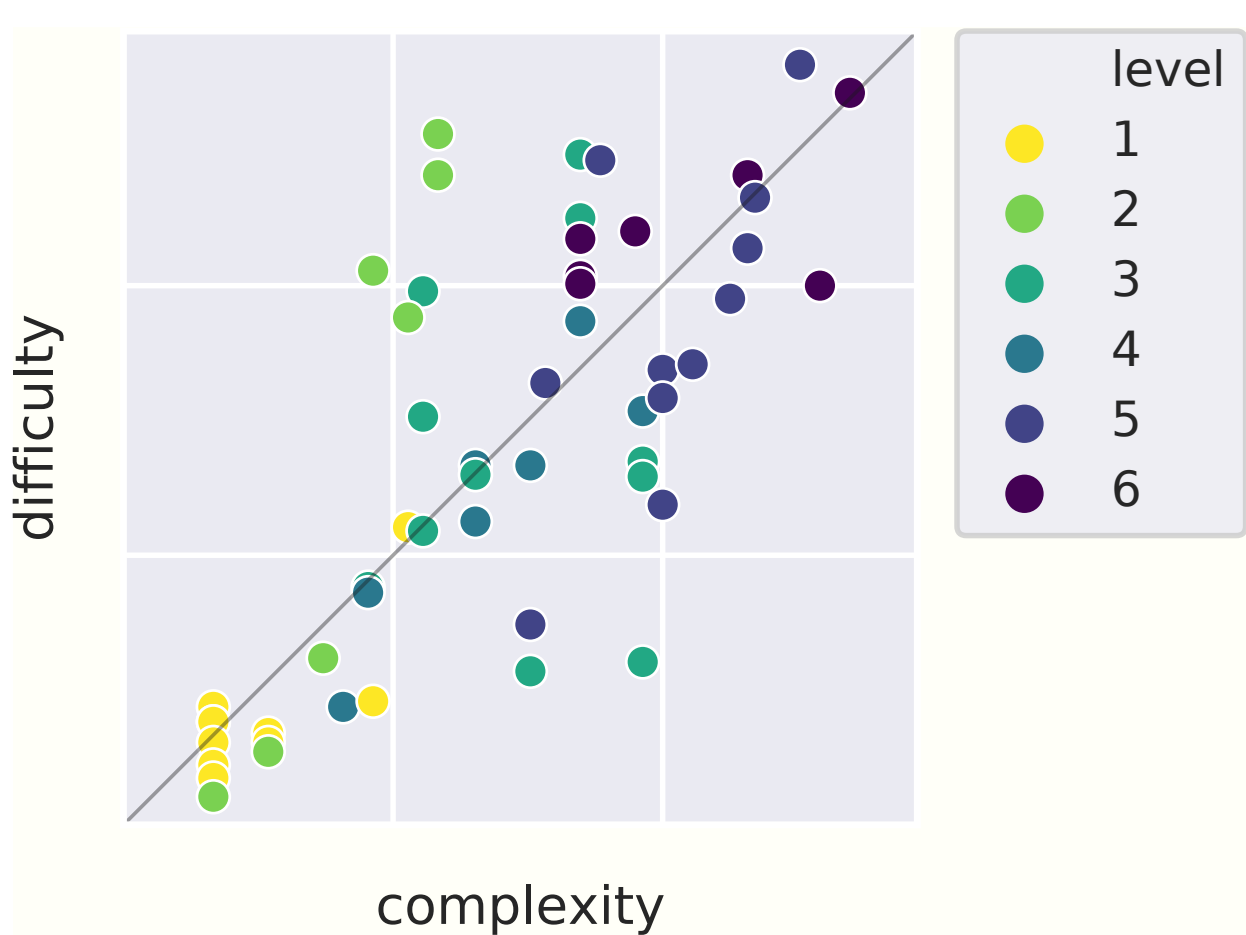
## Performance Aspects

The correlation of the same measures can differ widely between exercises.



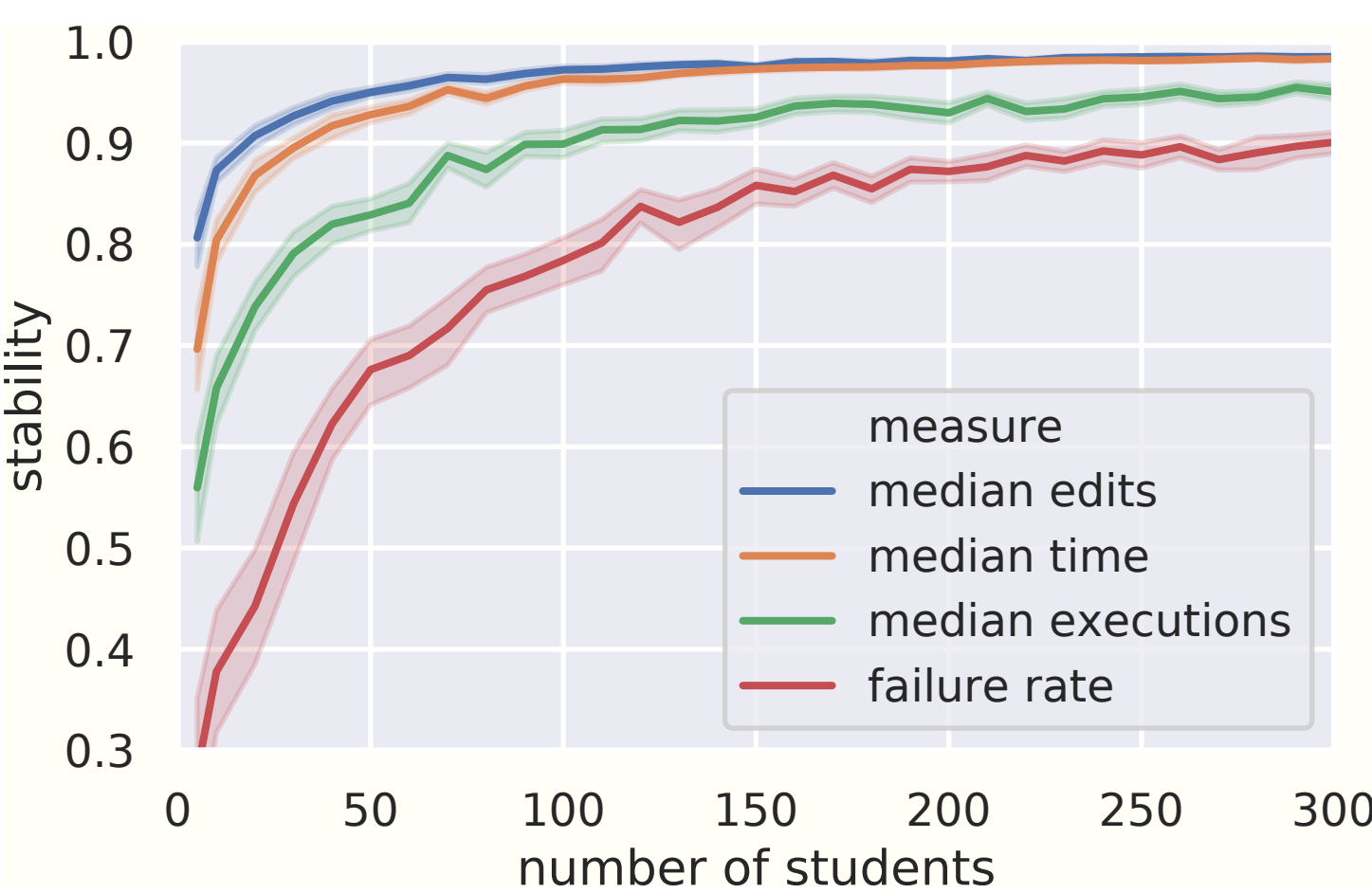
## Difficulty vs. Complexity

Comparing difficulty and complexity of a task can reveal misplaced tasks and "tricky" puzzles.



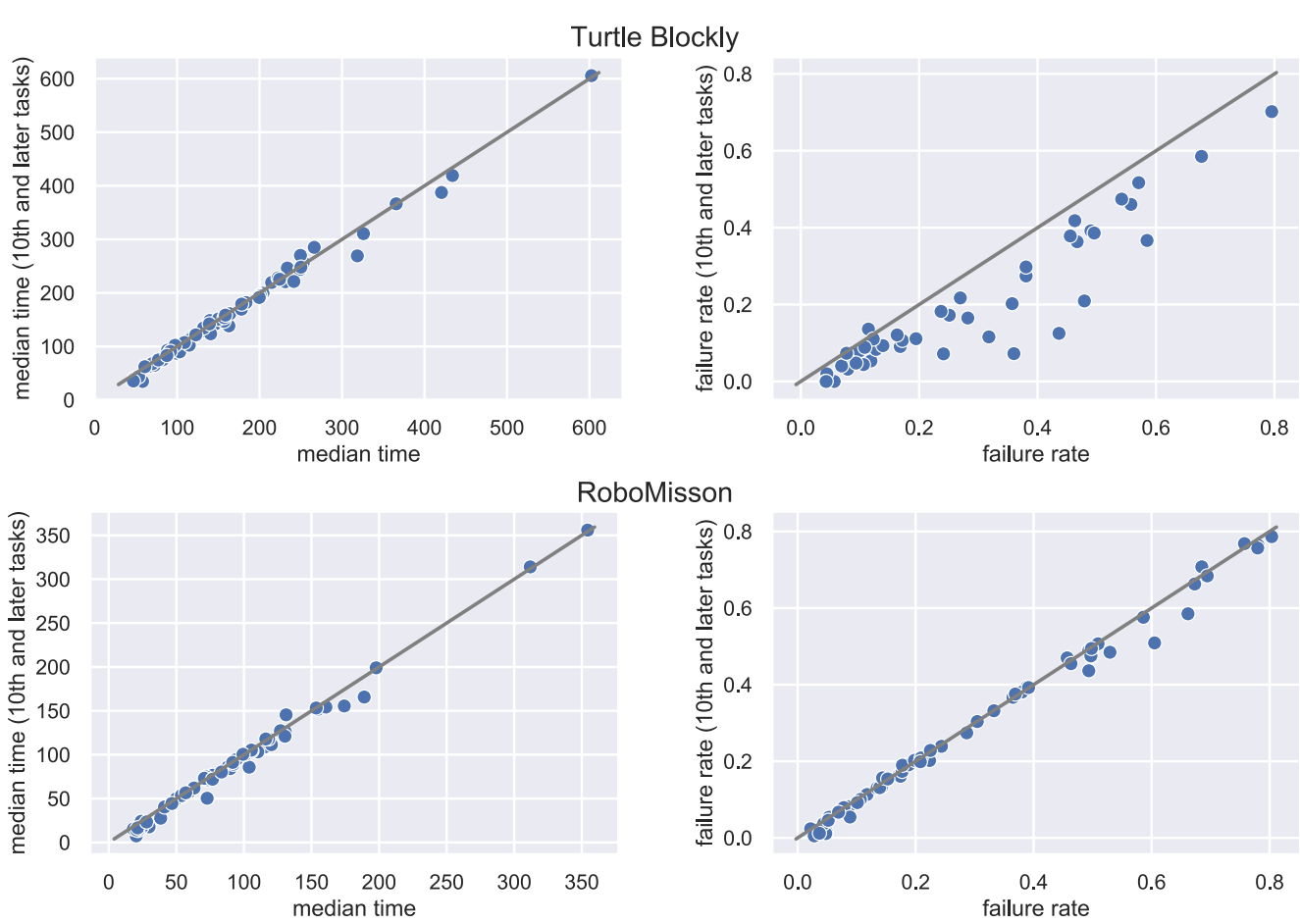
## Stability

Measures using more bits of information from student attempts converge faster.



## Biases and Filtering

Filtering attempts students can reduce biases depending on the learning system.



## Summary

The behavior of difficulty measures varies widely across programming exercises.