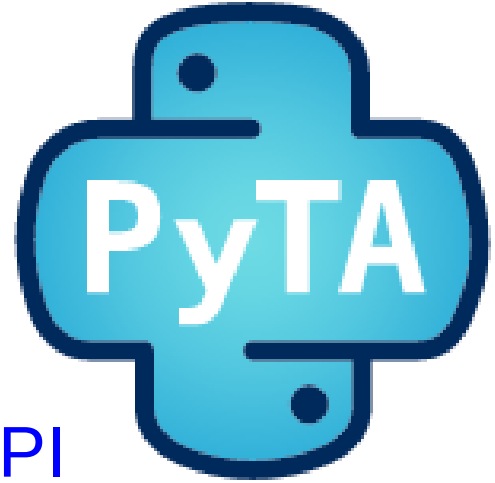
A Static Analysis Tool in CS1: Student Usage and Perceptions of PythonTA

David Liu, Jonathan Calver & Michelle Craig

Computer Science
UNIVERSITY OF TORONTO

PythonTA



- Free open-source Python package
- Wraps around pylint and pycodestyle
- Run in the terminal or use its Python API
- Customizable

```
PyTA Report for: C:\Users\David\Documents\my_file.py
= Code errors/forbidden usage (fix: high priority) =
E9996 (one-iteration)  Number of occurrences: 1.
  [Line 7] This loop will only ever run for one
        iteration
     5  def search(lst: list, item: Any) -> bool:
     6      """Return whether item is in lst."""
     7      for x in lst:
     8          if x == item:
     9              return True
    10          else:
    11              return False
    12
    13

= Style/convention errors (fix: before submission) =
No problems detected, good job!
```

❐ E9996 (one-iteration) (Learn More) 1 occurrence.

[Line 7] This loop will only ever run for one iteration

```
5  def search(lst: list, item: Any) -> bool:
6      """Return whether item is in lst."""
7      for x in lst:
8          if x == item:
9              return True
10         else:
11             return False
```

Our CS1 Course

a large public research-oriented university

11 weekly programming exercises and 3 large programming assignments (39%)

- PythonTA part of grading for 10-20% per item
- 1% deduction per PythonTA error
- Fully available to students during development

Research Questions

RQ1 Do students use PythonTA when working on their programming assignments?

RQ2 Do students perceive PythonTA feedback to be useful for their learning and their self-efficacy?

RQ3 Do the answers to RQ1 and RQ2 change for subgroups of students based on prior programming experience or other student characteristics?

Our Study

Surveys in week 5 and after week 12 (.5% each)

- usage and perceptions of PythonTA
  - Student demographics: gender, prior experience, ethnic background, aspirations in CS, comfort reading English

Participants

1168 consented but only 896 completed both surveys

42.5% men 39.3% women 1% non-binary 17.2% declined

70% planned to take more CS

46% considering enrolling in CS program

For quantitative analysis removed another 78 leaving 818

Prior Experience

Self-reported on a number of statements

We combined to

NONE "I had no programming experience before…" 401

COURSE "took course in school or Uni before this one" 264

OTHER everyone else

153

PythonTA Usage

How have you used PythonTA up to this point?

- frequently "... frequently as I work on an assignment"
- before_final "... before making my final submission"
- fix_all "I fix all the errors PythonTA reports before I submit"
- check_grade "I check my PythonTA grade after marks are released"

13 Perceptions Survey Questions

7-point Likert scale (-3 strongly disagree to +3 strongly agree)

- I find PythonTA error messages to be confusing and hard to understand
- PythonTA helps me fix logical errors in my code
- Using PythonTA has supported my learning in this course

Factor Analysis

Exploratory Factor Analysis on random subset of 300

- Two-factor model
- Removed two items and assigned others to factors

Helpfulness: helpful to one's learning

PyTA Self-efficacy: one's ability to respond to PythonTA messages

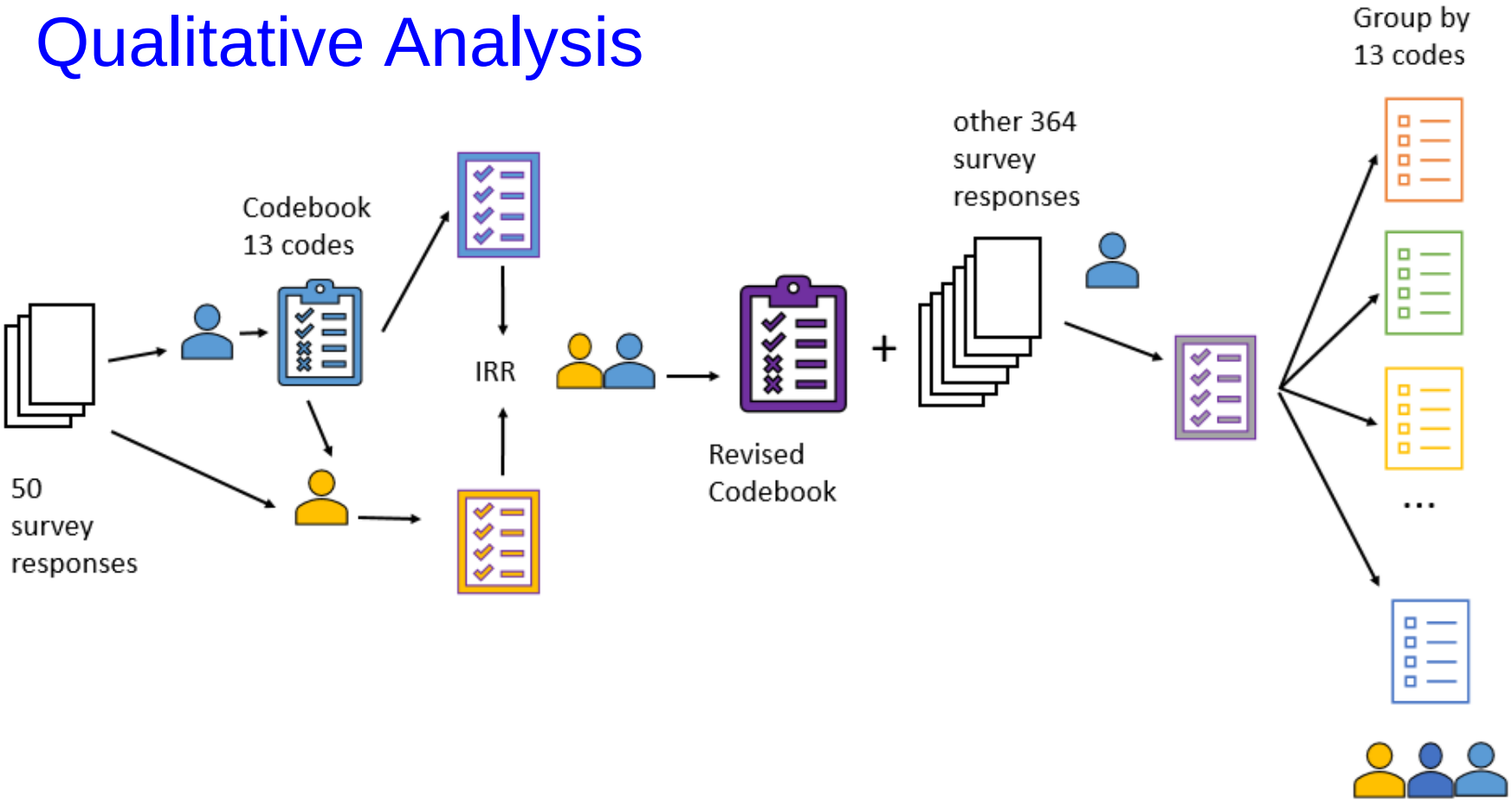
Confirmatory Factor Analysis on remaining 1336 surveys

Qualitative Analysis

“Is there anything else you would like us to know about your experience using PythonTA this term? If so, please explain in a couple of sentences”

755 responses -> removed “no” -> 414

Qualitative Analysis



Results: PythonTA Usage

Usage	Survey 1	Survey 2	p value
frequently	34.7	45.5	< .001
before_final	69.9	66.0	.073
fix_all	63.7	53.9	<.001
check_grade	22.2	32.3	<.001

Results: PythonTA Usage

before_final	Survey 1	Survey 2	p value
All students	69.9	66.0	.073
NONE	67.3	63.8	.310
COURSE	73.1	73.5	1
OTHER	71.2	58.8	.018

Results: Student Perceptions

		Survey 1	Survey 2	
NONE	helpfulness	1.24	1.21	
	PyTA self-efficacy	0.17	0.355	p < .01*
COURSE	helpfulness	1.15	1.17	
	PyTA self-efficacy	0.93	0.92	.
OTHER	helpfulness	1.16	1.17	
	PyTA self-efficacy	0.78	0.95	p < .1

Results: Student Perceptions

		Survey 1	Survey 2	
NONE	helpfulness	1.24	1.21	
	PyTA self-efficacy	0.17	0.355	p < .01*
COURSE	helpfulness	1.15	1.17	
	PyTA self-efficacy	0.93	0.92	.
OTHER	helpfulness	1.16	1.17	
	PyTA self-efficacy	0.78	0.95	p< .1

Results: Student Perceptions

		Survey 1	Survey 2	
NONE	helpfulness	1.24	1.21	
	PyTA self-efficacy	0.17	0.355	p < .01*
COURSE	helpfulness	1.15	1.17	
	PyTA self-efficacy	0.93	0.92	.
OTHER	helpfulness	1.16	1.17	
	PyTA self-efficacy	0.78	0.95	p < .1

Results: Qualitative Analysis

Helped Improve Students' Code Style

Earlier I would write programming as if I were writing an essay, but after PyTA it looks like programming

PythonTA has allowed me to reorganize and rewrite my codes in a more appropriate and brief manner

Tedious & Time-Consuming but Helpful & Habit Changing

Fixing PythonTA errors takes a lot of time, but if I successfully fixed, my code looks much better.

PythonTa has been tremendously helpful … as I now find myself, almost subconsciously avoiding those common errors that are raised on PythonTA.

Not everyone was convinced of the importance of style

Need lots of time to fix style errors, which do not necessarily improve my skills in python programming.

Some of the style conventions are incredibly stupid and often make code more difficult to understand.

A Common Complaint

it is a very irritating feeling to have all your code work then plug it into pythonTA only to have it return 50 instances of “trailing whitespace”

Increase Confidence by Helping Find Errors

As a novice coder, PythonTA has been a tremendous help in allowing me to submit assignments more confidently and avoid common errors when writing code

Increased Anxiety Levels for Some

Python TA hurts my feelings and scares me because it is so daunting

it does give me anxiety because there are so many errors and not enough explanations

Fixing the Errors Can Be a Challenge

I like how it pointed out every style issue I had, but figuring out how to fix the style issue was difficult.

Sometimes it gets too overwhelming and I don’t understand what exactly [I’m] doing wrong

PythonTA Helpful “By The End”

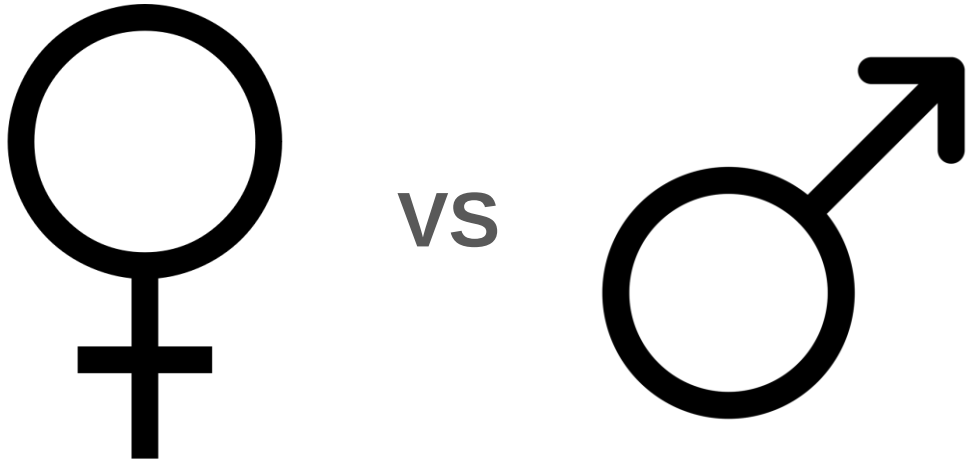
PythonTA was intimidating at first, but once I got used to read- ing the different kinds of errors, it was extremely helpful.

Improving Error Messages

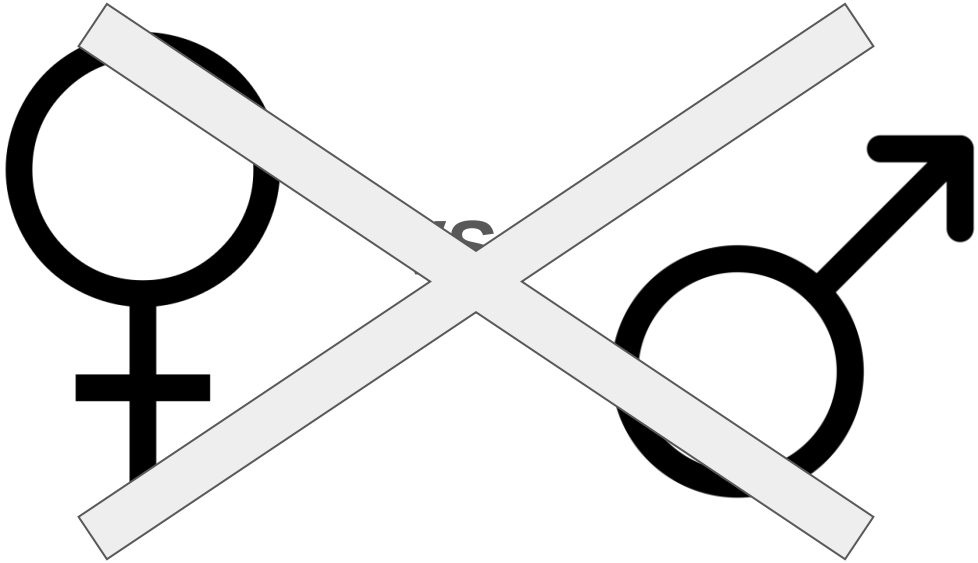
it could be improved by being more clear on what & where is the error, why does the error matters and maybe hint to fix them.

The code which report the error was too complicated, I need to read the full sentences to find the error that I make, I think it can be more convenience to read.

Variations across Subgroups



Variations across Subgroups



Variations across Subgroups

Asked “I am comfortable reading English” (-3 to +3)

Only weak correlations:

PyTA Self-efficacy ($\boldsymbol{\rho}$=0.20, $\boldsymbol{p}$<.001)

Helpfulness ($\boldsymbol{\rho}$=0.09, $\boldsymbol{p}$<.014)

Variations across Subgroups

Asked “I am comfortable reading English” (-3 to +3)

Only weak correlations:

PyTA Self-efficacy ($\boldsymbol{\rho}$=0.20, $\boldsymbol{p}$<.001)

Helpfulness ($\boldsymbol{\rho}$=0.09, $\boldsymbol{p}$<.014)

Variations across Subgroups

Prior Experience: Novices vs all others

Novices less likely to fix all errors before final sub

Equally likely to run PythonTA frequently

Helpfulness: No differences

PyTA Self-efficacy: Novices lower than others

Self-efficacy gap decreases on second survey

Recommendations for Instructors

1. Enable auto-formatting in IDE
2. Provide examples of fixing common errors
3. Customize the tool's error messages
4. Allow students to run the tool themselves repeatedly
5. Explain the importance of good style rules enforced by tool

Conclusions

Static analysis tools can work in an educational context

Even for beginners

Even for students with low language proficiency

With some care

Thank you

david@cs.toronto.edu

calver@cs.toronto.edu

mcraig@cs.toronto.edu