

# Approaches to programming in first year Physics

Yuliyana Stefanov

2476572S

## Introduction

- Its been 70 years since computing became a taught degree and the right teaching method remains unclear.[1]
- Computer Science drop out rates are the highest of any degree (~50%)[1].
- The problem is too complex and ambiguous to tackle as is.
- We must understand student's feelings around programming and uncover the **most relevant** problems with it.

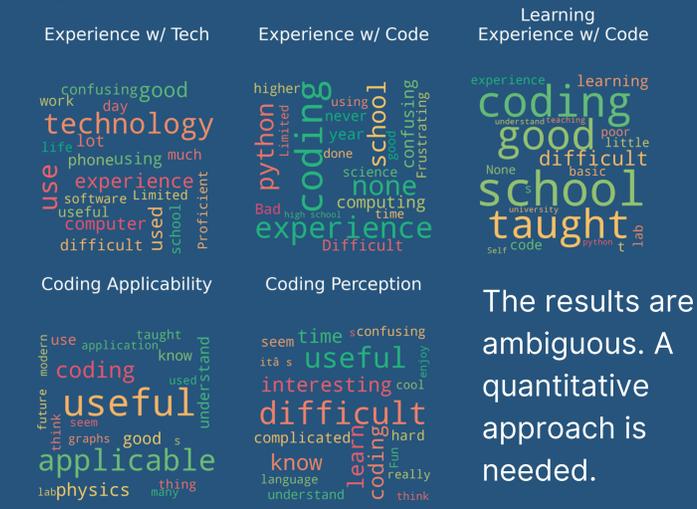
## Background

- Success in programming means ability to divide problems into chunks, solve and organize accordingly.
- Novices tend to think linearly, while experts think of strategies.
- Reasoning and strategy are more important than syntax for beginners.
- **PBL (Problem Based Learning)** proven to be a more effective strategy than traditional methods when applied to coding.[2]
- Not enough "good"[3] PBL problem sets.

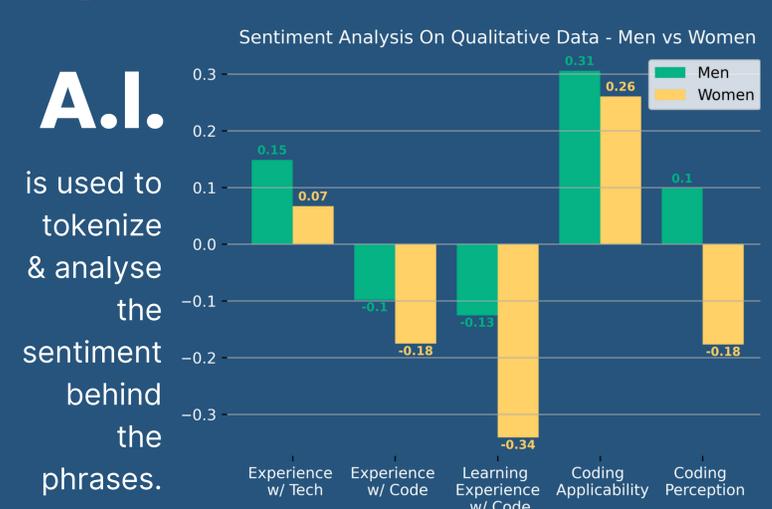
## Methods

- Questionnaire - 21 quantitative and 5 qualitative questions. Based on categories\* of skill/experience and **self-perception**.
- Questions constructed on the basis of IOP Physics Deg. criteria [4], UofG Graduate Attributes.[5]
- Analyse qualitative answers by Wordcloud collection and AI sentiment analysis for finer detail. (fig.1 & 2)
- Analyse quantitative answers by examining the correlation between categories (fig.3), analysing their principal factors (PCA) (fig.4) and diluting them into a better set of variables.

### 1 ...what are students saying?

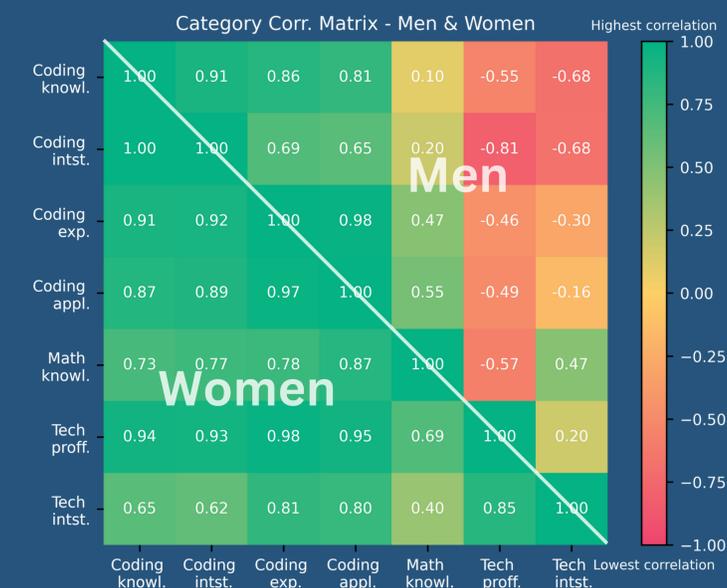


### 2 ...students are unhappy



"...Coding is useful, but I don't like the way its taught."

### 3 What are their abilities, experience and interests?



We see **great differences** in interest & experience between men and women...

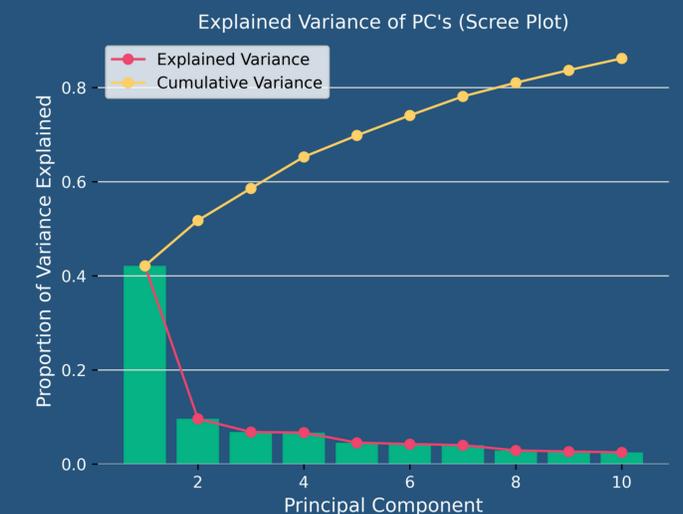
### 5 The Key Performance Indicators

The first 3 principal components account for **60% of the variance**. The most important indicators across the 3 principal components are:

- Technical Familiarity and Comfort
- Perception of Programming Relevance
- Perception of Coding Practicality

### 4 What are the most important categories? (Key Performance Indicators)

Principal Component Analysis (PCA) is used to reduce complexity of a large dataset by identifying the factors that contribute most to a correlation.



A **Scree Plot** shows which **principal components** account for a percentage of the variance.

## \*Categories

- Coding Interest
- Coding Knowledge
- Coding Experience
- Coding Perception
- Mathematical Knowledge
- Technological Interest
- Technological Proficiency

## Results

- Students are **unhappy** with their coding experience.
- Virtually all students believe in the **usefulness** and **applicability** of programming for their career.
- Dimensionality reduction into **3** principal components accounts for **60%** of the explained variance. The factors which account for most of the variance can now be focused on more concretely.

## Conclusion

- New teaching tactics must be employed to tackle improving the **Key Performance Indicators**.
- Some suggestions would include familiarising students with Computing Basics (such as OS file hierarchies) prior coding and increasing awareness of the **necessity** and **applicability** of programming in their careers.
- A strong practical and proven candidate for this would be employing more **PBL strategies**.
- Further research into the differences between men and women in their self perception regarding their technical skills must take place.
- Practices must be put in place to standardise student confidence for a **healthier** and more **productive environment**.

## References

[1] Margulieux, L.E., Morrison, B.B. & Decker, A. Reducing withdrawal and failure rates in introductory programming with subgoal labeled worked examples. *IJ STEM Ed* 7, 19 (2020). <https://doi.org/10.1186/s40594-020-00222-7>

[2] NUUTILA, E., TÖRMÄ, S., KINNUNEN, P. & MALMI, L. 2008. Learning Programming with the PBL Method. In: BENNEDSEN, J., CASPERSEN, M. E. & KÖLLING, M. (eds.) Reflections on the Teaching of Programming Methods and Implementations Berlin: Springer-Verlag

[3] Jackie O'Kelly and J. Paul Gibson. 2006. RoboCode & problem-based learning: a non-prescriptive approach to teaching programming. *SIGCSE Bull.* 38, 3 (September 2006), 217-221. <https://doi.org/10.1145/1140123.1140182>

[4] <https://www.iop.org/sites/default/files/2019-10/the-physics-degree.pdf>

[5] [https://www.gla.ac.uk/media/Media\\_183776\\_smx.pdf](https://www.gla.ac.uk/media/Media_183776_smx.pdf)