Every Chemist a Programmer

K Cowtan
VICEPHEC 2021

UNIVERSITY of York

Department of Chemistry

History...

Nearly 2 decades of teaching programming to chemists...

- 2004-2012: **Python** Y4 option module, 6 lectures + 4 workshops
- 2014-2019: **R** Y3 option module, 4 lectures + 3 workshops
- 2021- : **Python** Y1 skills, 90 mins mini lectures + exercises

Approach: Initially very traditional, building up from basic elements.

Outcomes: <20% of students would get to the point where they could use programming in their projects/labs. That subset biased male.

New course context...

Plan:

- Delivered to all Y1 chemists
- Limited to 6 hours of contact
  - 2 x 1 hour lectures
  - 2 x 2 hour workshops

Covid context

- Delivered electronically
- 12 short videos
  - (total 90 mins)
- Online exercises supported by quizzes and Q&A forums

Spyder (Python 3.8)

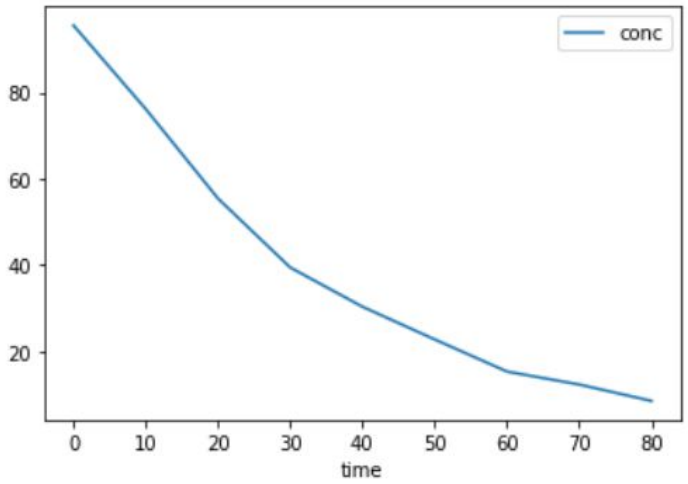
File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Kevin\Courses\Python

C:\Users\Kevin\Courses\Python\demo1.py

demo1.py demo2.py demo3.py demo4.py

```
1  import pandas as pd
2
3  concs = pd.read_csv("reaction1.csv")
4  print( concs.describe() )
5  concs.plot(x="time")
```



Variable explorer Help Plots Files

Console 1/A

```
              time        conc
count    9.000000    9.000000
mean    40.000000   39.541572
std     27.386128   30.389325
min      0.000000    8.513256
25%     20.000000   15.325440
50%     40.000000   30.345580
```

IPython console History

LSP Python: ready conda (Python 3.8.8) Line 5, Col 21 ASCII CRLF RW Mem 71%

Type here to search

16°C Clear ENG 20:41 10/08/2021

Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

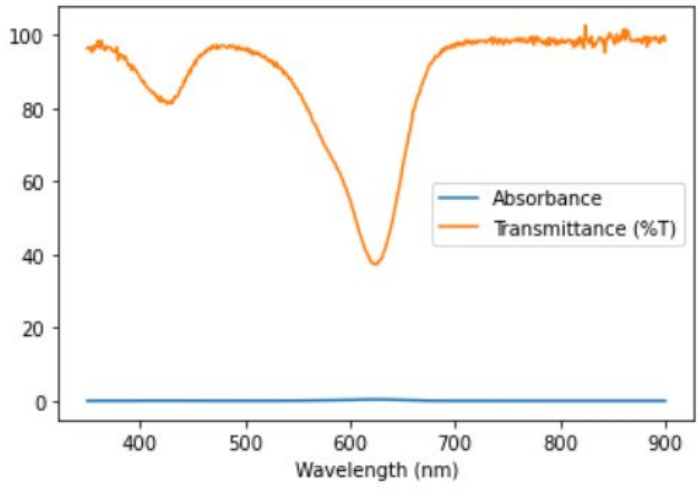
C:\Users\Kevin\Courses\Python

C:\Users\Kevin\Courses\Python\demo2.py

demo1.py | demo2.py | demo3.py | demo4.py

```
3   # photometer file is simple CSV
4   d1 = pd.read_csv("Photometer_0.025_M.csv")
5   d1.plot(x="Time_s")
6
7   # IR data has no headers
8   d2 = pd.read_csv("CTLIRPC1_20201011_154920.c
9   names=["Wavelength","Transmittance"])
10  d2.plot(x="Wavelength")
11
12  # UVJ data need to skip 8 header rows
13  d3 = pd.read_csv("DATA_UVJ_20190502_170757.c
14  d3.plot(x="Wavelength (nm)")
15
16  # JCamp-dx file - space separator and # for
17  d4 = pd.read_csv("Iron_Complex.dx",sep="\s+"
18  names=["Wavelength","Absorbance"])
19  d4.plot(x="Wavelength")
20
21  # ElNino data is fixed format
22  d5 = pd.read_fwf("ELNINO_wksst8110.for",
23  skiprows=3,widths=[10,9,4,9,4,9,4,9,4])
24  d5.plot(x="Week")
25  print(d5.describe())
```

135 %



Variable explorer | Help | Plots | Files

Console 1/A

```
max       29.200000      4.600000  ...     30.400000
1.800000

[8 rows x 8 columns]

In [8]:
```

IPython console | History

LSP Python: ready | conda (Python 3.8.8) | Line 16, Col 1 | ASCII | CRLF | RW | Mem 72%

Type here to search

16°C Clear ENG 20:41 10/08/2021

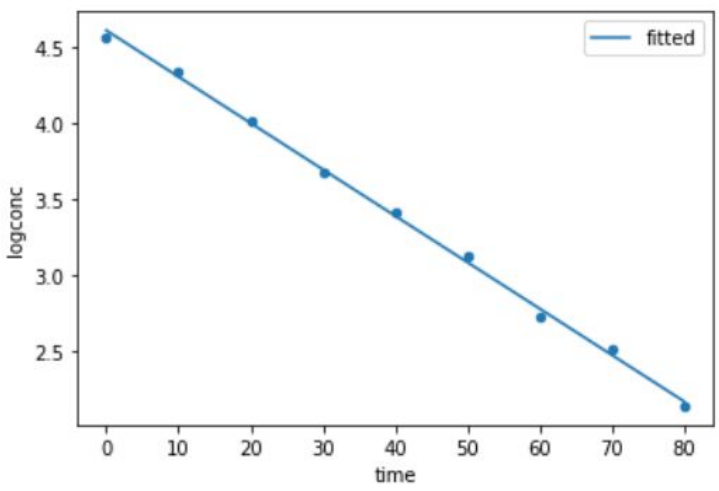
Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Kevin\Courses\Python\demo3.py

demo1.py demo2.py demo3.py demo4.py

```
1  import pandas as pd
2  import numpy as np
3  import statsmodels.formula.api as sm
4
5  # read data
6  concs = pd.read_csv("reaction1.csv")
7  concs.plot(x="time")
8
9  # linearize for first order
10 concs["logconc"] = np.log(concs["conc"])
11
12 # perform ols fit
13 fit = sm.ols("logconc ~ time", concs).fit()
14 print(fit.summary())
15
16 # plot
17 concs["fitted"] = fit.fittedvalues
18 ax1 = concs.plot.scatter(x="time",y="logconc"
19 concs.plot(x="time",y="fitted",ax=ax1)
```



Variable explorer Help Plots Files

Console 1/A

```
Notes:
[1] Standard Errors assume that the covariance matrix of
the errors is correctly specified.

In [9]:
```

IPython console History

LSP Python: ready conda (Python 3.8.8) Line 7, Col 1 ASCII CRLF RW Mem 70%

Type here to search

16°C Clear ENG 20:41 10/08/2021

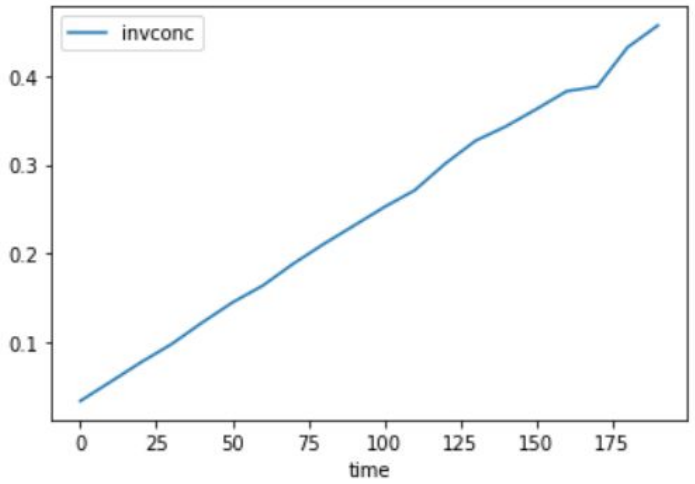
Spyder (Python 3.8)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\Kevin\Courses\Python\demo4.py

demo1.py demo2.py demo3.py demo4.py

```
16
17      # perform ols fits
18      fit0 = sm.ols("conc ~ time", concs).fit(
19      fit1 = sm.ols("logconc ~ time", concs).f
20      fit2 = sm.ols("invconc ~ time", concs).f
21
22      # gather the rsquared values
23      rsq0 = fit0.rsquared
24      rsq1 = fit1.rsquared
25      rsq2 = fit2.rsquared
26
27      # test for 0th order
28      if rsq0 > rsq1 and rsq0 > rsq2:
29          ax = concs.plot(x="time",y="conc")
30          print("Reaction is 0th order")
31          print("Rate is ",-fit0.params["time"
32
33      # test for 1st order
34      if rsq1 > rsq0 and rsq1 > rsq2:
35          ax = concs.plot(x="time",y="logconc"
36          print("Reaction is 1st order")
37          print("Rate is ",-fit1.params["time"
38
39      # test for 2nd order
```



Variable explorer Help Plots Files

Console 1/A

```
Reaction is 1st order
Rate is  0.016691681472875263
Reaction is 0th order
Rate is  1.9721765553946267
Reaction is 2nd order
Rate is  0.002193416293750461
```

IPython console History

LSP Python: ready conda (Python 3.8.8) Line 28, Col 1 ASCII CRLF RW Mem 71%

Type here to search

16°C Clear ENG 20:41 10/08/2021

12 short videos

1. Introduction
2. Python and Spyder
3. History, psychology and sociology of programming
4. Using Spyder
5. Writing a useful program
6. Pulling it to bits
7. Bugs and debugging
8. Reading data
9. Arithmetic with numbers and columns
10. Fitting a line to a graph
11. Conditions
12. Loops

12 short videos

1. Introduction
2. Python and Spyder
3. History, psychology and sociology of programming
4. Using Spyder
5. Writing a useful program
6. Pulling it to bits
7. Bugs and debugging
8. Reading data
9. Arithmetic with numbers and columns
10. Fitting a line to a graph
11. Conditions
12. Loops

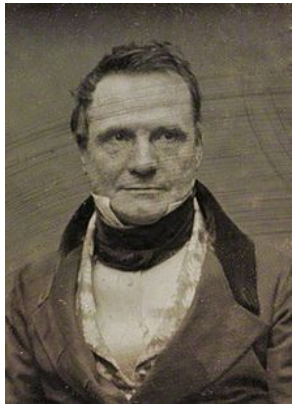
History, psychology and sociology of programming

Aims: Understand social factors which might get in the way of learning to program

History of computer programming...

Charles Babbage (1791-1871) built what is considered to be the first mechanical computer

Ada Lovelace (1815-1852) developed the ideas for programming it.



Early computation...

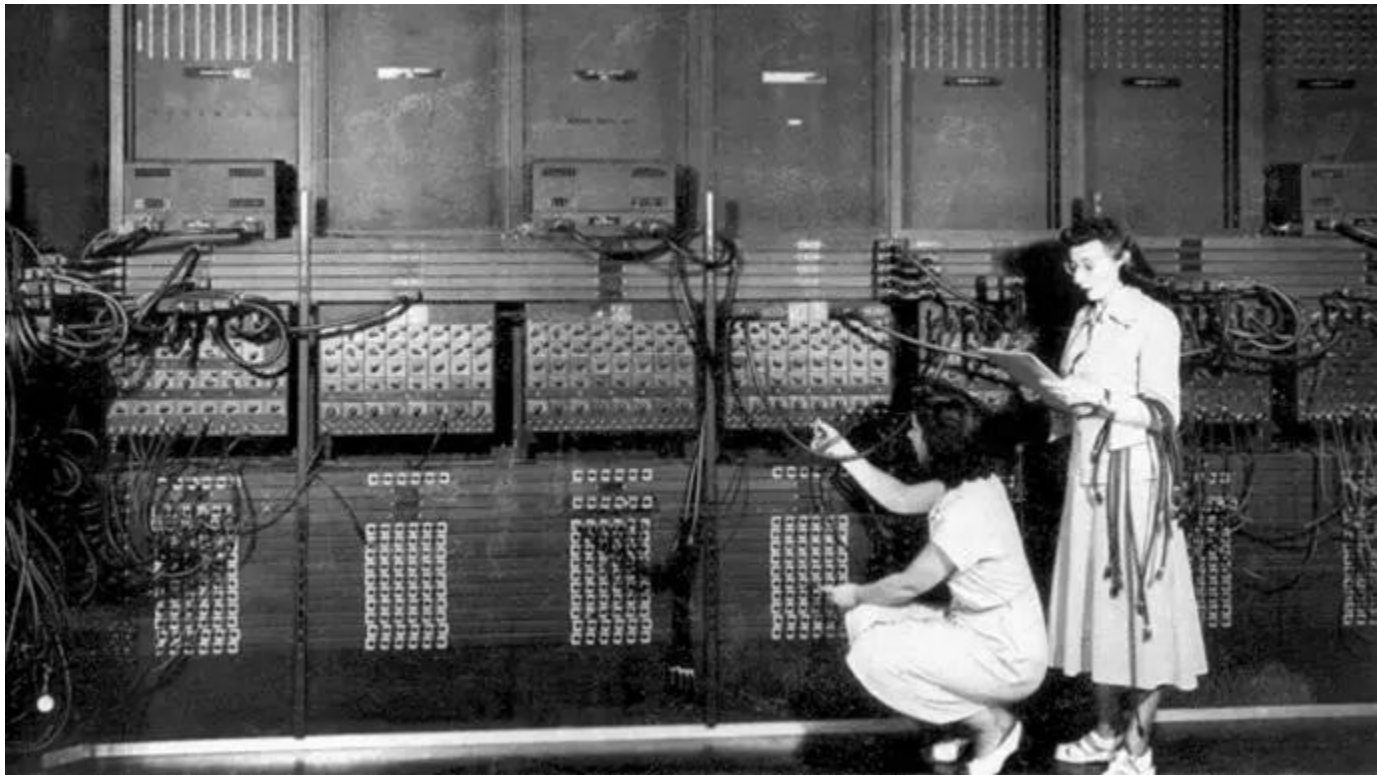
The "Harvard Computers",
Harvard observatory, 1875

Early computation...

US treasury 1920s

Electronic computers...

ENIAC, 1946



Computers becoming programmers...

NASA space program...

Computer programming...

Grace Hopper (1906-1992) developed COBOL, the first modern computer language.

“Programming requires patience and the ability to handle detail. Women are ‘naturals’ at computer programming.”

Margaret Cowtan...

1960s: Vauxhall, Brush

Today...

- Only 2% of top contributors to the main python code sharing site are women
- Computing and computer programming are now *perceived* to be aptitudes associated with a very small, mostly male minority.
- This is not supported by history, by experience in schools or in other departments at York.

Freemantle/Channel 4

https://blog.revolutionanalytics.com/2018/12/women-and-r.html

https://blog.revolutionanalytics.com/2016/06/programmers-gender.html

How did this happen?

Hiring policies changed to favour male applicants.

Computer advertising was targeted at boys

- The status of the job increased
- It was increasingly done by men

No longer a "pink collar" job.

Apple][
1985

Review of "The computer boys take over",
https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6479433

When computer coding was a women's job:
https://www.history.com/news/coding-used-to-be-a-womans-job-so-it-was-paid-less-and-undervalued

Impacts: some online spaces can be exclusionary...

Stack Overflow (top computing Q&A site) users were asked "What would you like to change?"

- Men*: “official, complex, algorithm”
- Women*: "condescending, rude, assholes"

1. There are distinct gendered communication styles (MacCoby)
2. Male communication is typified by higher levels of social dominance signalling (MacCoby)
3. Male communication styles tend to become dominant in anonymous mixed contexts (Brooke)

Gender and relationships: a developmental account (1990) E. MacCoby
https://webs.wofford.edu/nowatkacm/Abnormal%20Child/Maccoby1990.pdf

Framing gender and hostility on Stack Overflow. (2019) S. Brooke
https://www.aclweb.org/anthology/W19-3519.pdf

* based on self-declared gender for binary respondents

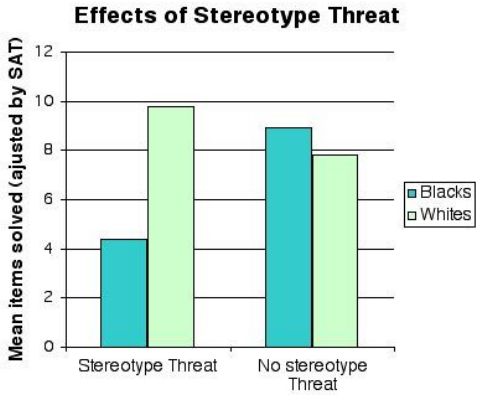
Implications...

If you do not fit the stereotype of a computer programmer, social pressures are working to make computing harder for you than it is.

- *Stereotype threat* increases anxiety levels
- *Performance anxiety* reduces performance

This is *particularly* true for women and gender minorities, racial and ethnic minorities and people with disabilities, but *also* for many men.

This serves to maintain the exclusivity of the activity.
(How much this affects you will vary with personality.)



Is there a solution?

9.3% of top contributors to the main R code sharing site are women - nearly 5 times better than python.

- R is taught differently (data science background)
- R has women as core developers
- R ladies (including gender minorities)
- R forwards (including other minorities)

But python is more used in chemistry...

What can we do?

- Be aware of the social pressures.
  - This helps a bit - limited by *G.I. Joe bias*
- Exploit group learning - but pick your groups!
  - Use discord, messenger, google meet
  - Find a group that works for you both intellectually *and* psychologically
- Find problems you are interested in.
  - Find fun data and analyze it (#TidyTuesday). Plot data in interesting ways. Write games
- We will try and enable approaches to learning which...
  - Reach more people and break down stereotypes
  - Reduce inequalities due to different approaches to learning

Did it work?

- "I thought the info on the history of coding was interesting and useful in making the course seem less scary to start."
- "it was nice that he gave some background lectures as well."
- "The video about the history of coding and improving access for women and minorities was excellent as it really **helped me to believe that I was capable** and it was worth trying properly instead of thinking it just wasn't my thing- **turns out I really enjoy coding**!"

Why the gender disparity?

- Gendered learning styles?
  - Gregorc style delineator?

British Journal of Educational Technology Vol 40 No 4 2009 696–712
doi:10.1111/j.1467-8535.2008.00847.x

Exploring the effects of gender and learning styles on computer programming performance: implications for programming pedagogy

Wilfred W. F. Lau and Allan H. K. Yuen

Wilfred W.F. Lau is working on his PhD dissertation to explore the relationship among gender, learning styles, mental model and programming performance in learning to program. Allan H. K. Yuen is an Associate Professor of the Faculty of Education, The University of Hong Kong. He is also a Deputy Director of the Centre for Information Technology in Education (CITE) at the university. Yuen has engaged in a number of research and development projects on information technology in education. Address for correspondence: Wilfred W.F Lau, Faculty of Education, The University of Hong Kong, Pokfulam Road, Hong Kong. Tel: 852 28592540; Fax: 852 25406360; email: wilfredlau@hkusua.hku.hk

Abstract
Computer programming has been taught in secondary schools for more than two decades. However, little is known about how students learn to program. From the curriculum implementation perspectives, learning style helps address the issue of learner differences, resulting in a shift from a teacher-centred approach to a learner-focused approach. This study aims to investigate the effects of gender and learning styles on computer programming performance. The Gregorc Style Delineator (GSD) was employed to measure learning styles. A test was administered to assess students' programming performance. Two hundred and seventeen secondary school students of age from 14 to 19 participated in this study. Results indicated that no gender differences in programming performance were found after controlling for the effect of student ability. Academic ability had a differential effect on programming knowledge. Sequential learners in general performed better than random learners. These results suggest the importance of the ordering dimension of the GSD in influencing programming performance. Implications of the findings in relation to programming pedagogy are discussed in this paper.

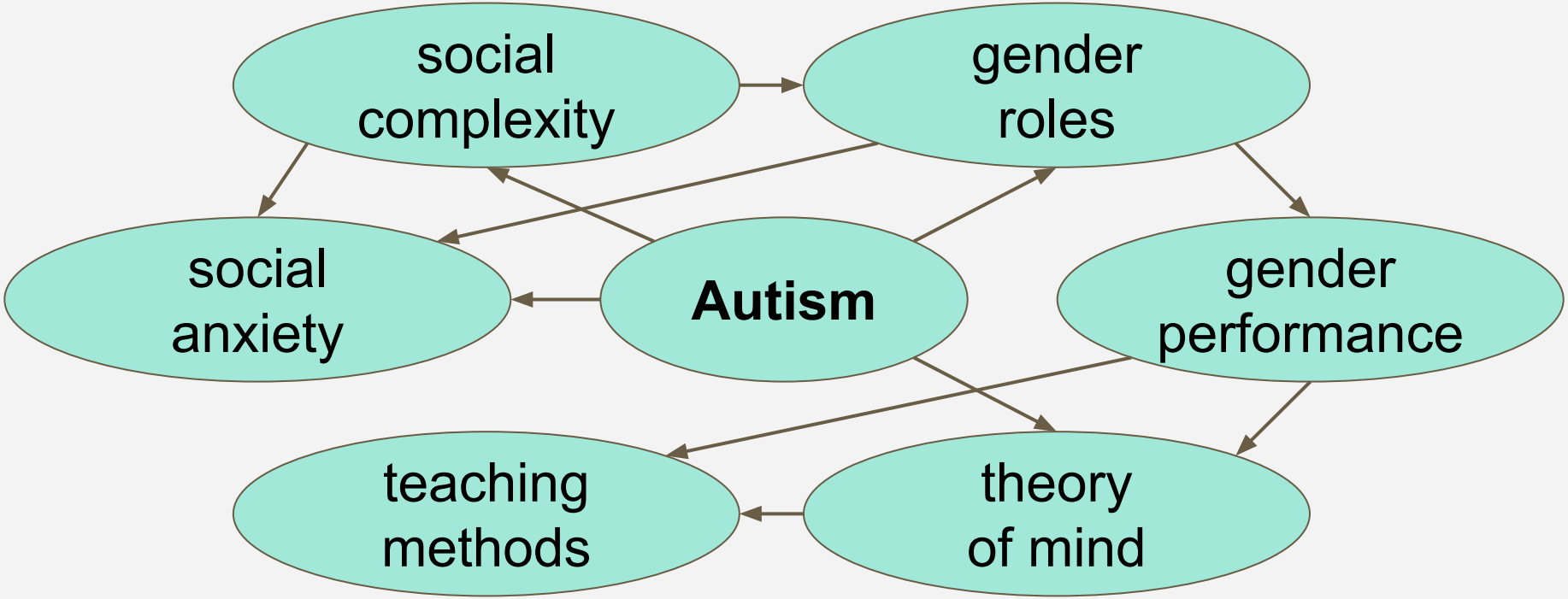
Introduction
The development of educational computing has emerged over the last two decades. Although advanced technology tools can be integrated into different curricula, learning to program still plays a role in promoting information literacy in technology education. Despite the fact that learner differences are one of the major concerns in recent

© 2008 The Authors. Journal compilation © 2008 British Educational Communications and Technology Agency. Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

Why the gender disparity?

- Because traditional teaching methods, tools, and online communities are structured to make programming hard. Only those who society has primed to believe they can and should be programmers can ever overcome the energy barrier.
  - Solution: draw attention to the barriers, and teach in a way which weakens them.

Personal context...



Questions and suggestions?

UNIVERSITY of York

Thanks to:

- David Pugh,
- Nigel Bailey,
- Mat Evans,
- Aneurin Kennedy,
- Lucy Wheeler,
- Caterina Mapelli,
- Nick Yates,
- Paul Bond,
- Tom Webb,
- Emma Rand
- Julie Wilson
- Susannah Cowtan

And everyone I've forgotten (sorry)!