Institute and Faculty of Actuaries

How to be more productive: Using programming

Shaun Lazzari

01 December 2016

Agenda for this session

- What it mean to be productive, and why you should care
- The fundamentals of programming
- Why these skills important for you and the profession
- What good programming look like
- Examples of using programming at work
- How to go about building programming skills

Institute and Faculty of Actuaries

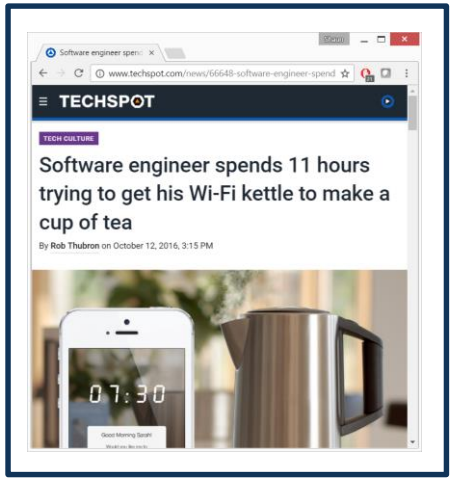
Non-goals of this session

- ✗ How to use any specific programming language
- ✗ Programming within actuarial systems and platforms

Institute and Faculty of Actuaries

Before we start – gauging your existing capabilities

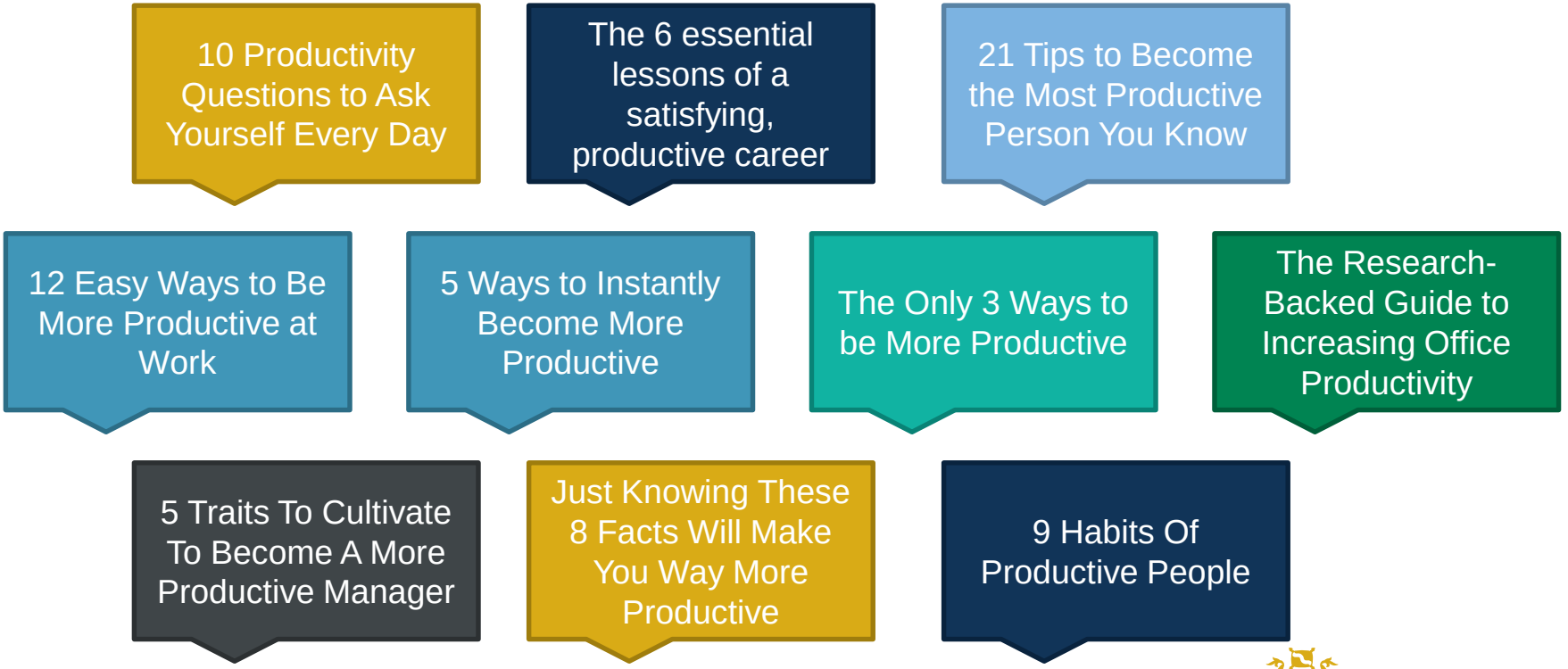
1. I carry out most of my work using pen, paper, and judgement
2. I'm a user of computer programs, e.g. Excel
3. I've recorded a few VBA macros
4. I occasionally use coding as part of my role
5. Coding is (or has been) a key part of my role
6. I am the subject of this article



Institute and Faculty of Actuaries

It seems a lot of people want to be more productive…

- Some titles from around the world wide web:



Institute and Faculty of Actuaries

There's two levers to productivity

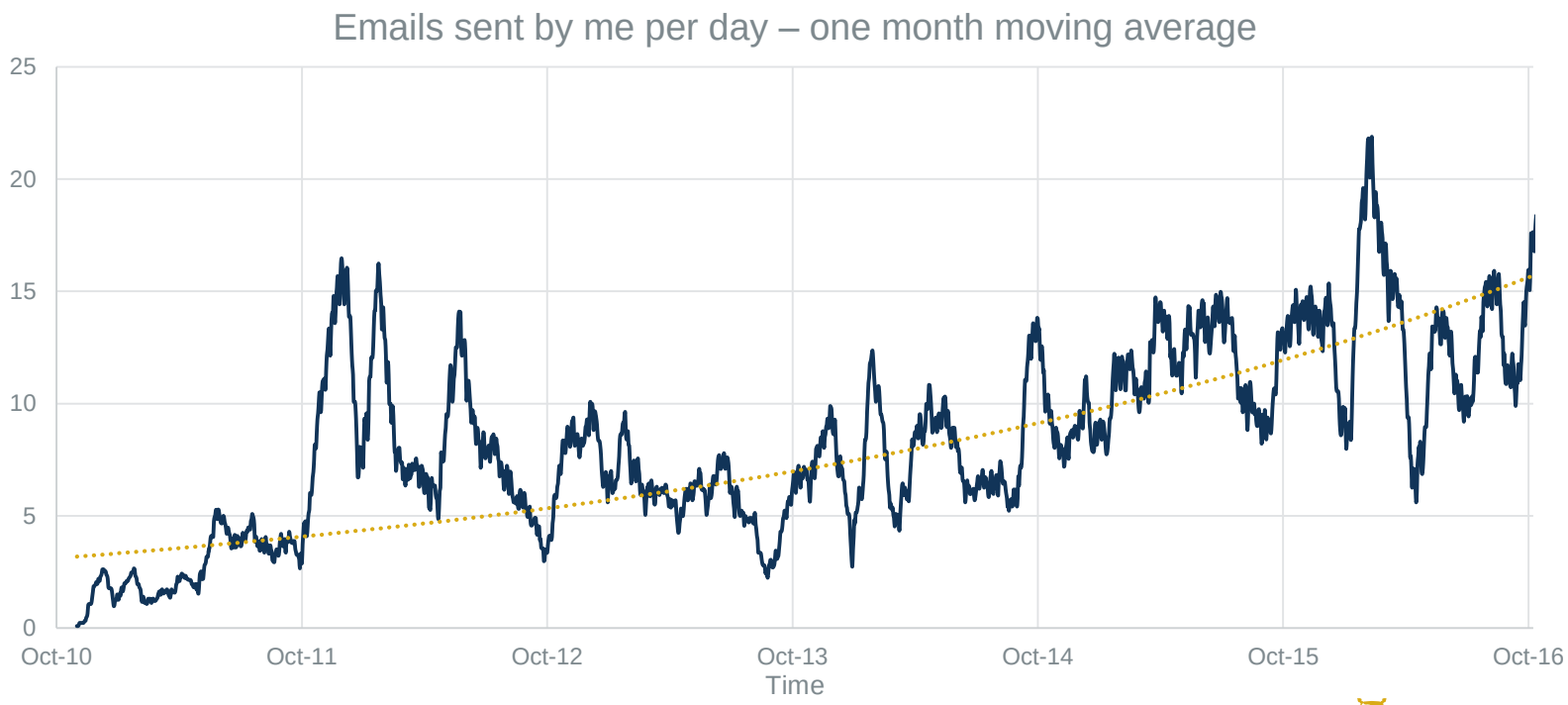
Value of output / **Effort inputted**

- The denominator is important, but don't forget the numerator…

Institute and Faculty of Actuaries

You're likely at a point in your career where productivity really begins to matter

- There's ever-increasing demands upon your time…



Institute and Faculty of Actuaries

You're likely at a point in your career where productivity really begins to matter

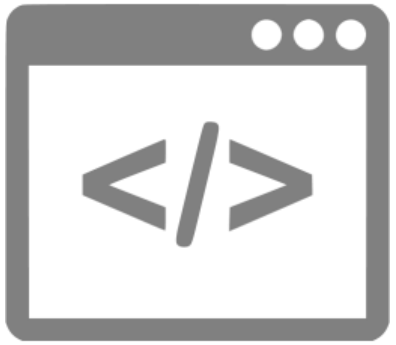
- …and scope for your work to have increasing levels of impact for you and your employer
- Some role descriptions for nearly/newly qualified actuaries

Source: www.theactuaryjobs.com

Institute and Faculty of Actuaries

Fundamentally, what is programming?

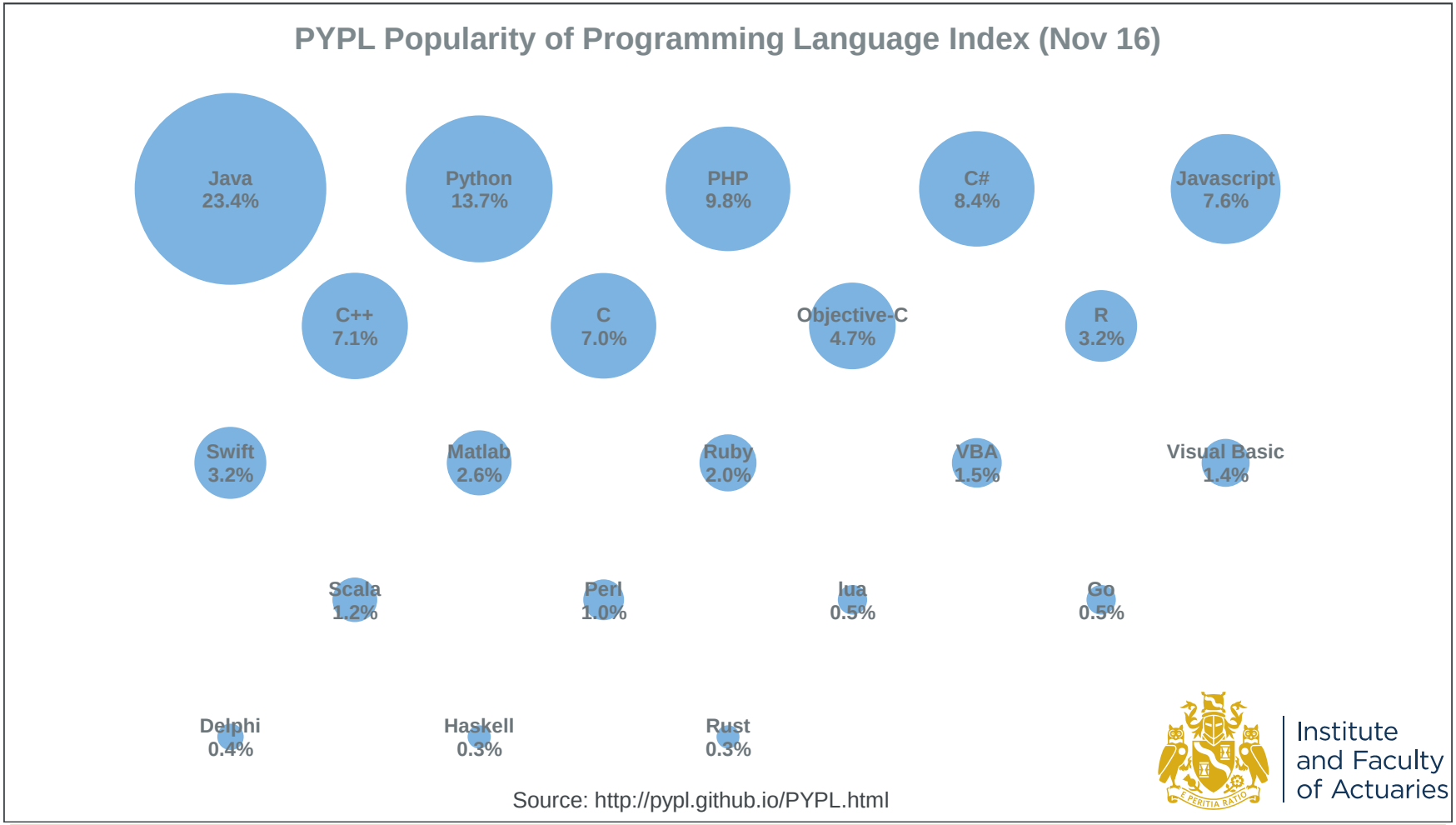
- Constructing an unambiguous model of a process or algorithm (“code”), and specifying it in a way that is interpretable to a computer



- Programming is performed in a chosen **language**

Institute and Faculty of Actuaries

There are a lot of different programming languages

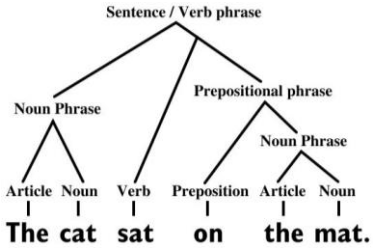


Source: http://pypl.github.io/PYPL.html

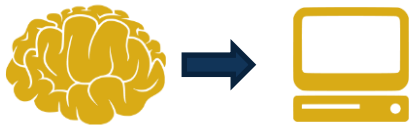
Institute and Faculty of Actuaries

There's a lot of parallels between programming and "natural" languages

Tools for communication	Syntax	Semantics



"Egg rides jelly bicycle"



```
for ( init; condition; increment ) {
   statement(s);
}
```

Institute and Faculty of Actuaries

There is one notable difference compared to natural language…

- Programming languages are **unambiguous**
- i.e. they can't grasp context or tone

Institute and Faculty of Actuaries

So why is programming important for actuaries?

- From the IFoA website:

What are the skill sets of an actuary?

- Actuaries use financial and statistical techniques to **solve business problems**… Actuaries have sufficient technical understanding to solve very demanding financial and risk management problems.
- It is essential that actuaries have **excellent communication skills** to enable them to communicate actuarial ideas to non-specialists in a way that meets the needs of the audience.

Institute and Faculty of Actuaries

Some benefits of being able to program

- Helps you to communicate with your computer, so you can use it in the most flexible way to solve your business problems
- This includes both:
  - Carrying out difficult maths and spotting relationships that would be impossible for a human to spot
  - Automating procedures to reduce processing timescales and the risk of human error

Institute and Faculty of Actuaries

Some benefits of being able to program

- Having a programmer's mindset can also help in your broader work:

Clear specification of problems and ideas

Creative, optimally structured solutions

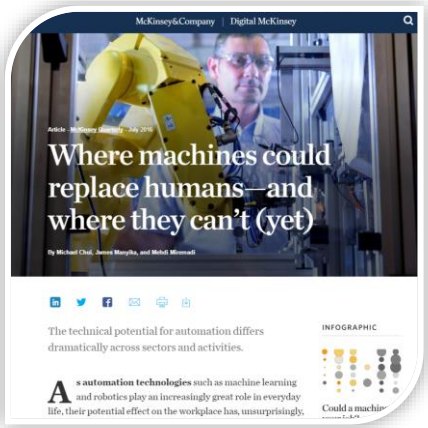
Collaboration

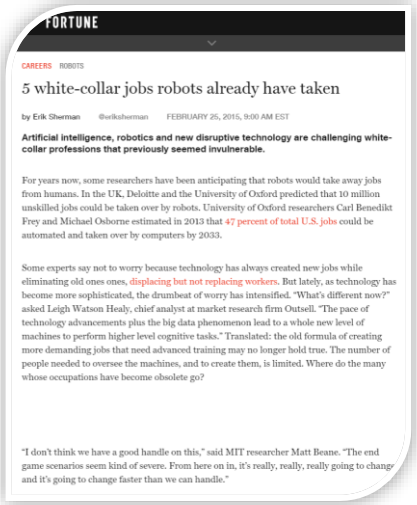
Understanding technology and better interaction with technology team

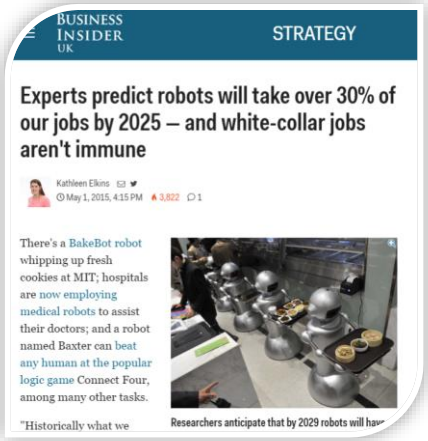
Institute and Faculty of Actuaries

Some benefits of being able to program

- Get computers to work for you, instead of them one day replacing you







Institute and Faculty of Actuaries

Some benefits of being able to program

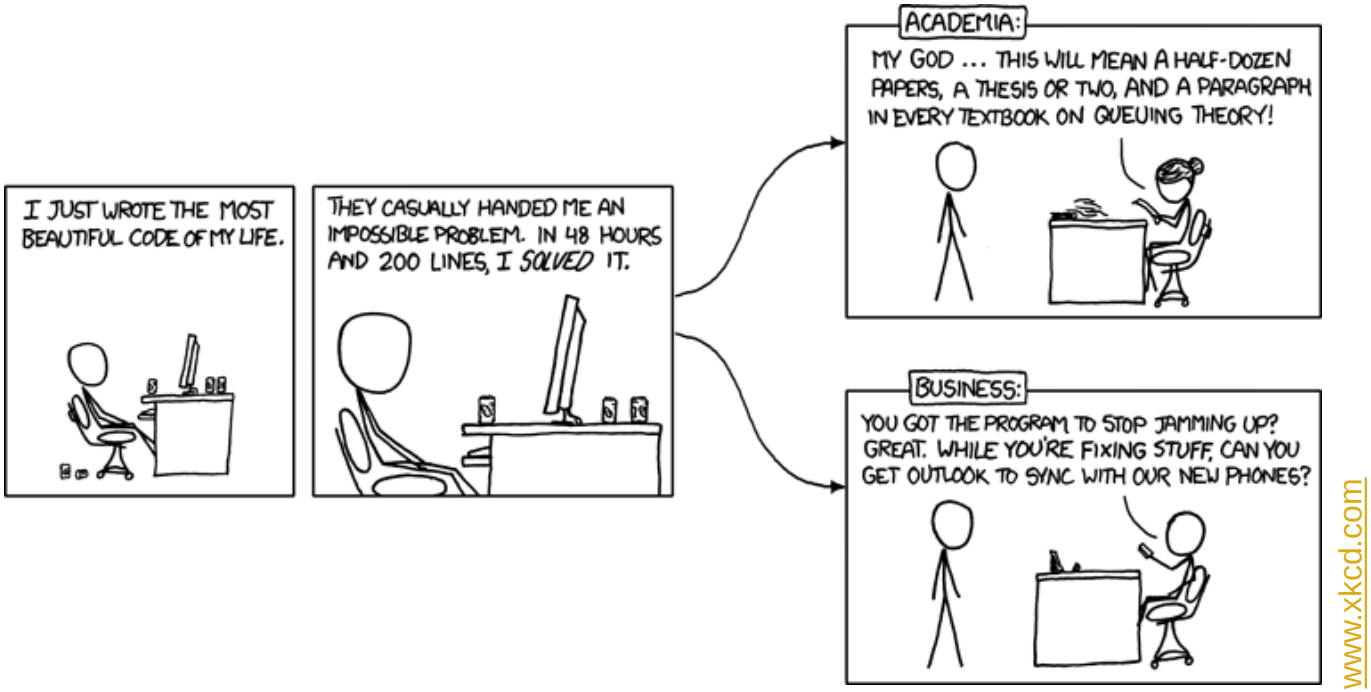
“Our policy at Facebook is literally to hire as many talented engineers as we can find. There just aren't enough people who are trained and have these skills today.”

“Learning to write programs stretches your mind, and helps you think better, creates a way of thinking about things that I think is helpful in all domains.”

Source: www.code.org

Institute and Faculty of Actuaries

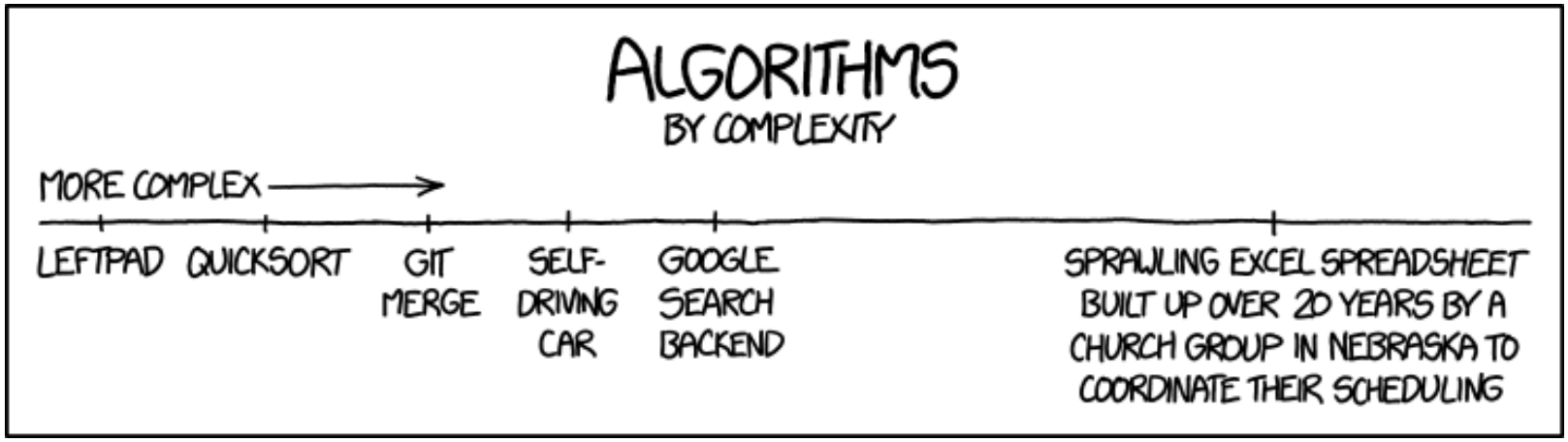
For actuaries, programming is a means to an end…



- Don’t learn to code, but code to learn!

Institute and Faculty of Actuaries

So what's the problem with Excel?



www.xkcd.com

Institute and Faculty of Actuaries

So what's the problem with Excel? (1/2)

Separation of input, calculations and output

- This doesn't always necessarily happen in spreadsheets
- It's harder to fall into this trap in code

Documentation

- It's often a separate process to record what a spreadsheet does
- Documentation may end up living quite far away from the calculations
- Code can be **self-documenting**

Clarity of calculation flow

- Some complex processes can be hard to step-through in Excel
- Code can make it easier to debug and traverse calculations

Limited maths functionality

- Excel doesn't support all of the actuarial techniques you might require
- Other packages may – or you can build them!

Institute and Faculty of Actuaries

So what's the problem with Excel? (2/2)

Lack of support for data structures more complex than 2D matrices

- Excel is based on cells in spreadsheets, i.e. 2D data arrays
- Programming in object-oriented languages allows you to specify the optimal structure for your data

Controls and testing

- No formal version control within Excel
- Workbook protection can be easily overridden, making it hard to resist making ad-hoc adjustments
- Vast swathes of version control and unit testing systems can be used to manage code

Re-usability

- The same calculations are often done across many spreadsheest - a waste of time which also introduces scope for inconsistency
- Modular nature of code means that operations can be packaged up separately and re-used

Processing speed and memory constraints

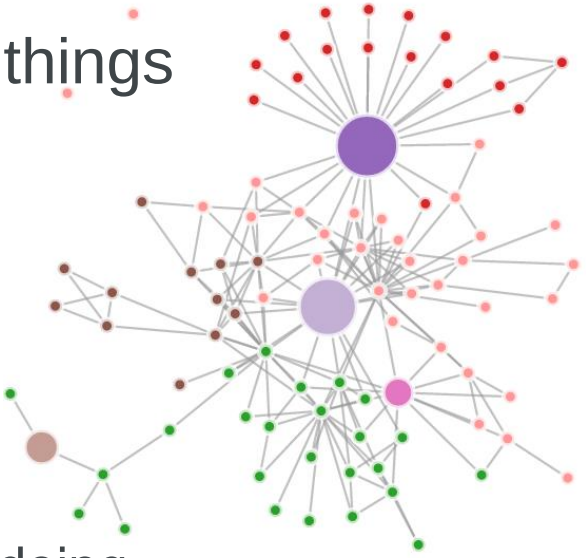
- Excel is not always the fastest option, and may struggle to process large data sets
- Other approaches offer more flexibility in handling usage of processors and memory

Institute and Faculty of Actuaries

What does good code look like?

Componentisation:

- Each routine should do as few distinct things as possible
  - Ideally one thing and one thing only



- So many benefits!
  - Clear what the routine you're looking at is doing
  - Helps you break down a problem into small chunks
  - Able to re-use code and ensure consistency
  - Easier to find and solve any bugs that arise

Institute and Faculty of Actuaries

What does good code look like?

Variable naming and operation complexity:

- Descriptive names will make your code easier to follow
- It is rarely necessary to lump lots of operations together

```
ExpectedCashflow = ProbabilityOfClaim * ClaimAmount
DiscountFactor = (1 + RiskFreeRate) ^ -YearsToClaimDate
PresentValueOfPolicy = ExpectedCashflow * DiscountFactor
```

```
pvpol = claimPrb * pol_sa * (1 + rf) ^ -T
```

Institute and Faculty of Actuaries

Understanding programming principles through some high level use cases

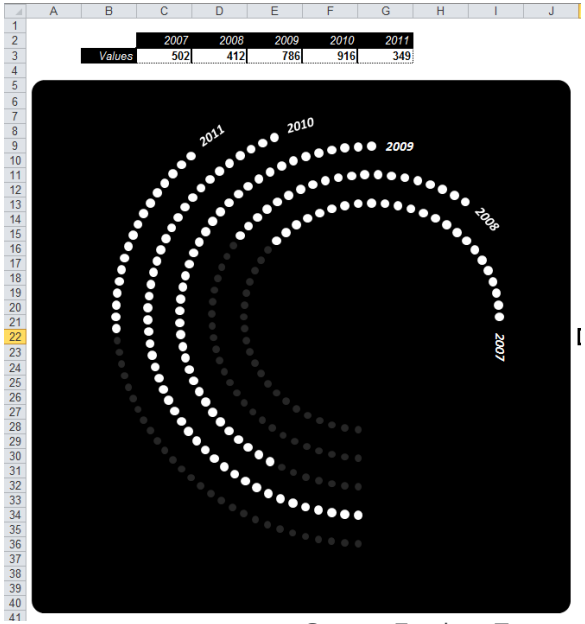
- We'll consider:
  - **Data interpretation:** Charting
  - **Automating data flow:** Documentation
  - **Building models:** Ad-hoc investigations
  - **Analysing and reacting to data:** Email

- This is far from a prescriptive set of uses
  - If something was useful for everyone you wouldn't have to program it, as someone else would have!

Institute and Faculty of Actuaries

Data interpretation: Charting

- Excel is a really powerful charting tool, even for data produced using other systems
- Every aspect of an Excel chart object can be modified programmatically via VBA
- Therefore, you can write some VBA routines to help you and your colleagues quickly make their favourite and/or most obscure charts

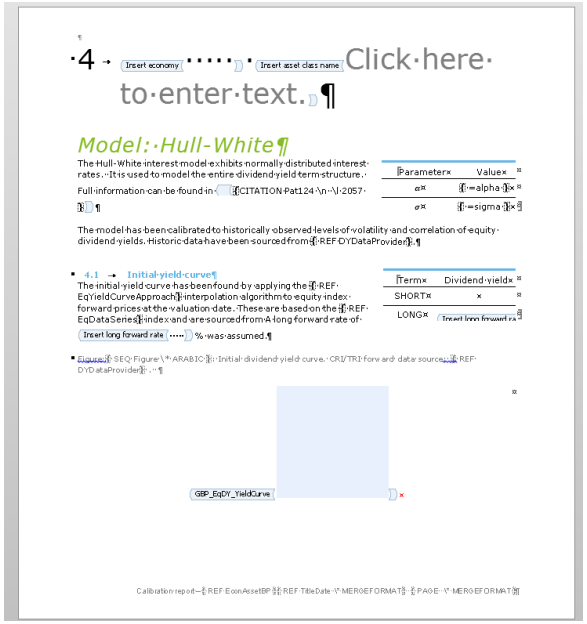


Source: FrankensTeam (https://sites.google.com/site/e90e50charts/)

Institute and Faculty of Actuaries

Automating data flow: Documentation

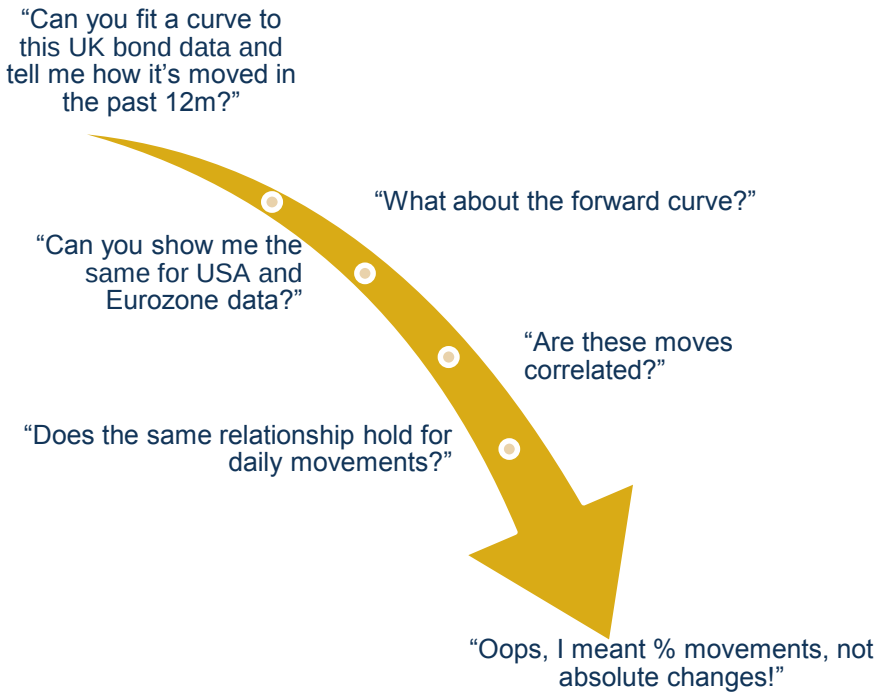
- All the structure, style and contents of a Word document are encoded in the file, and can be modified programmatically
- **Bookmarks** can be used to specify markers within a document so numbers and charts can be imported automatically using VBA
- Focus only on the **content** of your documents. The **style** is almost certainly someone else's concern!



Institute and Faculty of Actuaries

Building models: Ad-hoc investigations

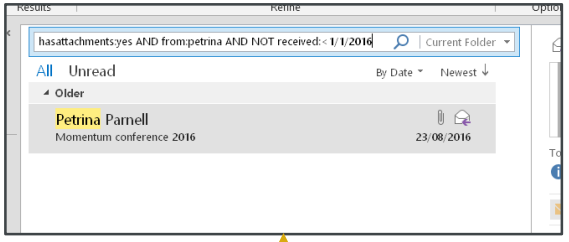
- Often you might have an idea you want to test, or be asked to do a brief analysis of
- In the likely event it grows and evolves, you'll be glad you made use of code



Institute and Faculty of Actuaries

Analysing and reacting to data: Emails

- Emails and calendar form a very rich, constantly evolving data set
- There's both a lot of structured data (From, To, Date etc.), plus the unstructured body data, to learn from
- Good way to understand **Events**
  - Outlook has inbuilt Rules functionality, but you can also code your own
- Search functionality is a form of code, with its own syntax, as well
  - …As is the case in most search tools

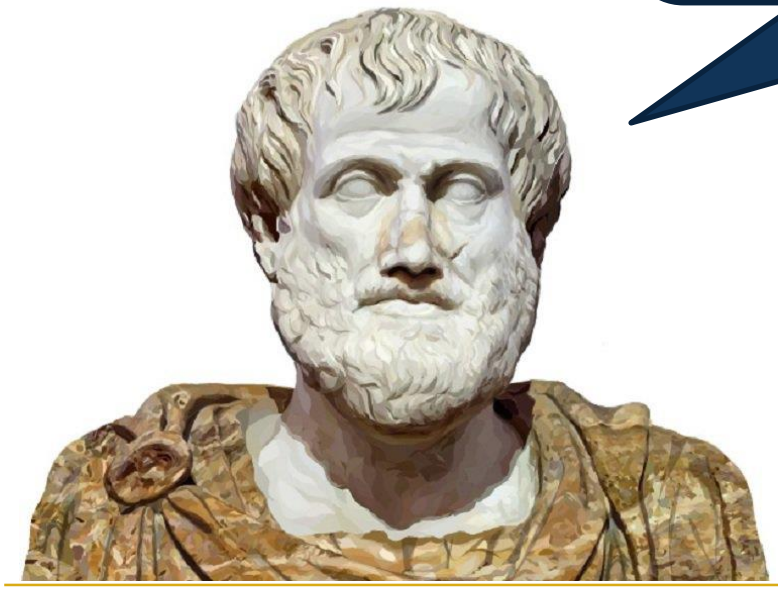


Institute and Faculty of Actuaries

Building your programming skills

"*For the things we have to learn before we can do them, we learn by doing them, e.g. men become builders by building and lyre-players by playing the lyre*"

Aristotle



Institute and Faculty of Actuaries

Building your programming skills

- **Find yourself a problem** that you're motivated to solve
- **Break-down** what you want to do into component parts
- **Have a go** – writing code is an iterative process!
- **Check online** – there are vast communities of people willing to help, and high quality tutorials and guides freely available
  - For VBA, Chip Pearson's site (www.cpearson.com) is outstanding
- **Ask colleagues** and draw upon their knowledge

Institute and Faculty of Actuaries

So, to summarise…

- Having an awareness of programming, and an ability to do so, could help you do your work:
  - more efficiently
  - with greater impact
- Programming is a skill that:
  - enables you to get the most out of technology
  - will further increase in demand industries and working practices evolve
- You learn by doing!

Institute and Faculty of Actuaries

Questions

Comments

Institute and Faculty of Actuaries