QEST 2019

End-User Probabilistic Programming

Andrew D. Gordon (MSR and University of Edinburgh)
Based on joint work with Judith Borghouts, Advait Sarkar, and Neil Toronto (MSR)

Pop Quiz

Your project is comprised of four independent, parallel subprojects. Each subproject will take between 1 and 2 months to complete, and each has an average completion time of 1.5 months.

On average, how long will your project take to complete?

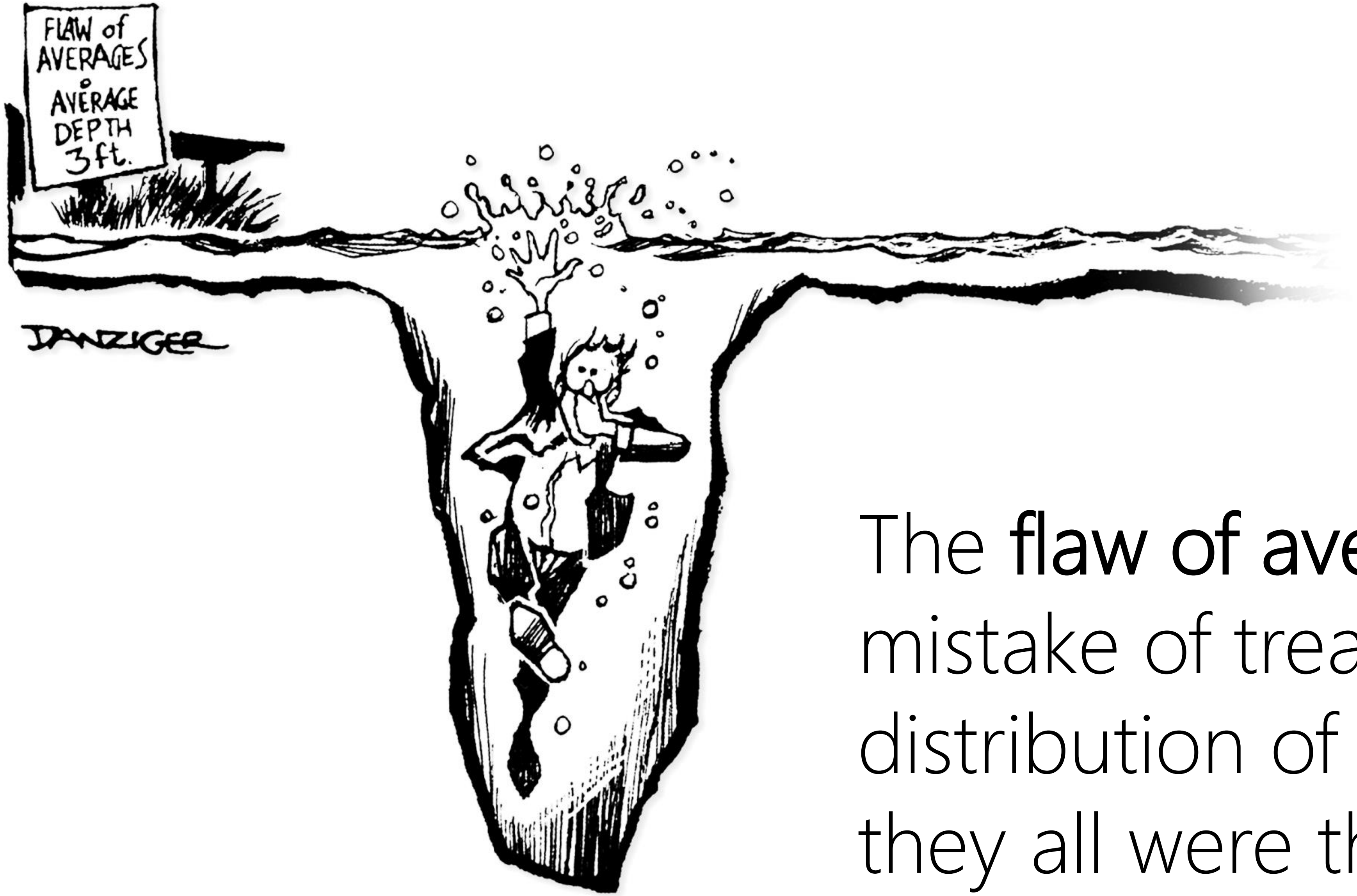
A. Less than 1.5 months
B. 1.5 months
C. More than 1.5 months

Quiz Answer

C. More than 1.5 months

Why? Mathematical answer: Jensen's inequality says so.

Why? Intuitive account: You can't finish *your project* until *all of your subprojects* are finished. Finishing all of them in ≤ 1.5 months is as likely as flipping four coins and having them all come up heads (1 in 16).



The **flaw of averages** is the mistake of treating a distribution of things as if they all were the average

Sam Savage *The Flaw of Averages* (2009)

My talk – End-User Probabilistic Programming

Probabilistic programming for spreadsheet formula authors

- Flaw of averages – good reason for EUPP
- Risk versus uncertainty – fundamental limitation of EUPP
- Trustworthy communication – next chapter for EUPP research

Using RAND() for Monte Carlo simulation is close to probabilistic programming

Add-ins support actual probabilistic programming

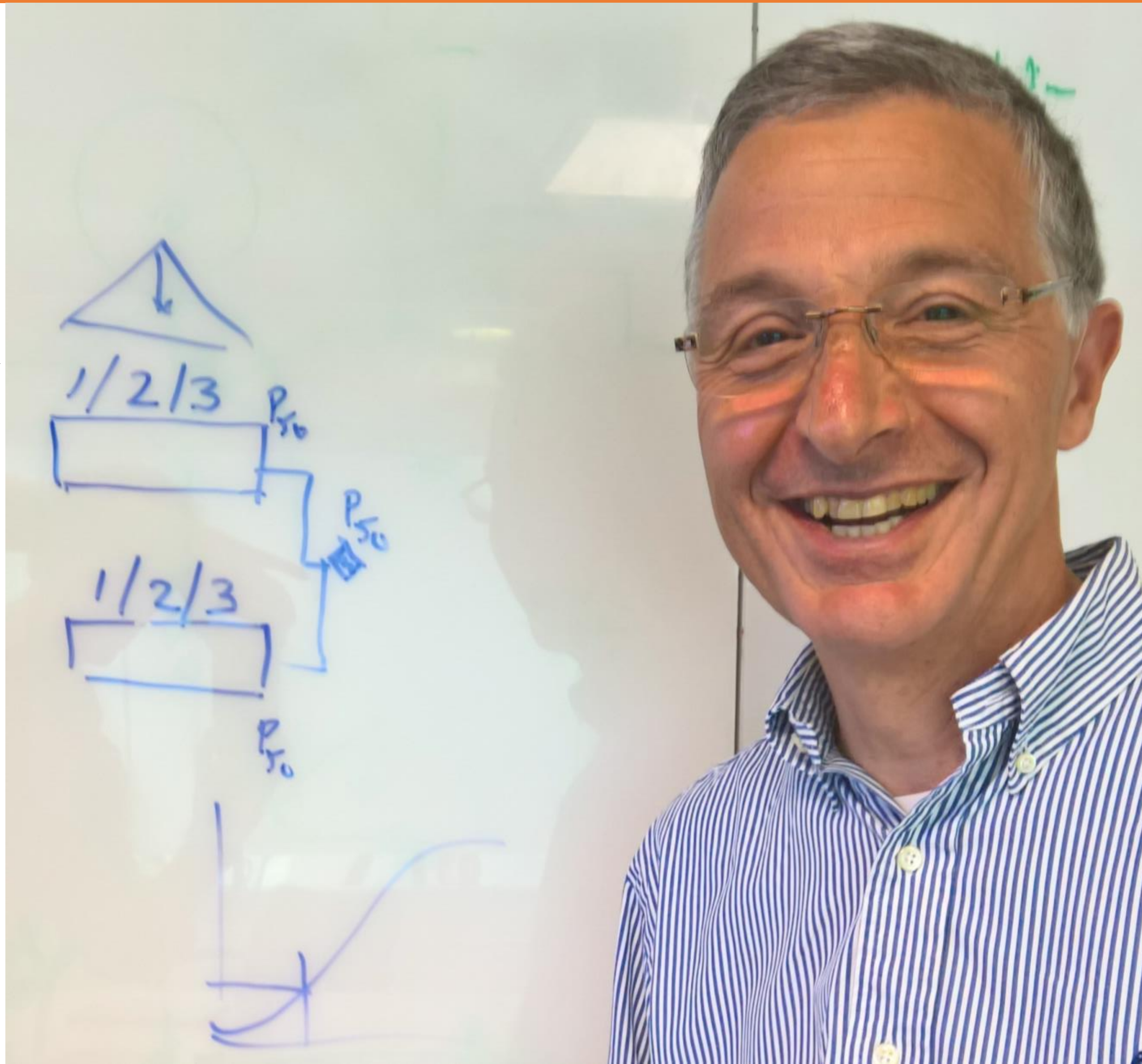
Insights from @Risk Users

We interviewed Peter Adlington, a consultant who helps companies make risk assessments using @Risk.

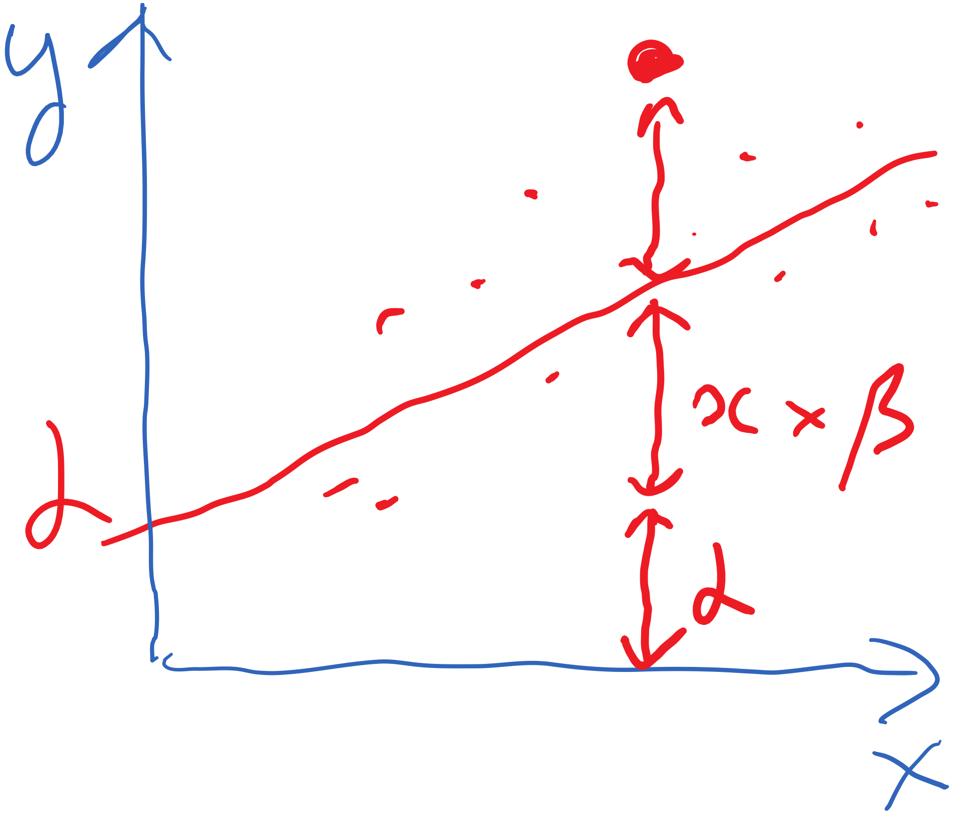
One key idea is the use of triples such as 1/2/3 to specify **triangular** distributions, with 1 being the **minimum**, 2 the **most likely**, and 3 the **maximum value**.

His customers don't initially appreciate that average inputs don't result in average outputs – the **Flaw of Averages**.

@Risk said to have ½ million users.



Probabilistic Programming Languages



$$\textbf{let } \alpha = \text{Gaussian}(0, {s_{large}}^2) \textbf{ in}$$
$$\textbf{let } \beta = \text{Gaussian}(0, {s_{large}}^2) \textbf{ in}$$
$$\textbf{let } \pi = \text{Gamma}(1, \lambda_{small}) \textbf{ in}$$
$$((\alpha, \beta), [\textbf{for } z < \underline{students} \rightarrow \alpha + x[z] \times \beta + \text{Gaussian}(0, 1/\pi)])$$

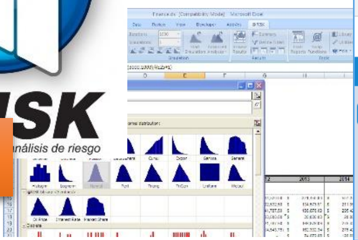
coefficients

predicted values of the y variable

Probabilistic Programming since the 80s

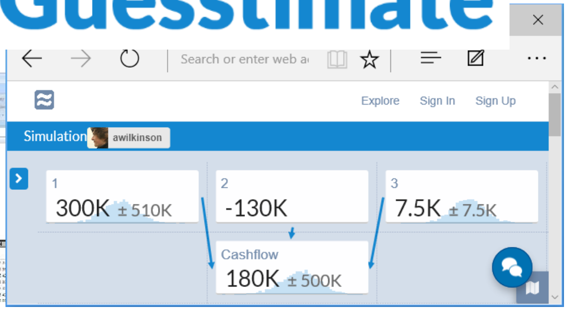
@RISK
análisis de riesgo

1987



Guesstimate

2016



Players			
Name	string	input	
Skill	real	latent	Gaussian(25.0,0.01)
Matches			
Player1	link(Players)	input	
Player2	link(Players)	input	
Perf1	real	latent	Gaussian(Player1.Skill,1.0)
...eal		latent	Gaussian(Player2.Skill,1.0)
...ool		output	Perf1 > Perf2

2014

INVREA | BETTER DECISIONS

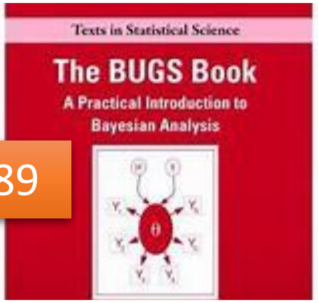
2016

CRYSTAL BALL

Spreadsheets since 1987
(@RISK launched)

PyMC3

2013



1989

infer.net

2004

2011

Stan

Church: a language for generative models

Noah D. Goodman, Vikash K. Mansinghka; Daniel M. Roy, Keith Bonawitz & Joshua B. Tenenbaum
MIT BCS/CSAIL
Cambridge, MA 02139

2007

Languages since 1989
(BUGS started)

Estimated Usage

EU Programming (writing formulas) in spreadsheets
(tens to hundreds of millions of people)

>>>

EU Probabilistic Programming (without conditioning) in spreadsheets
(hundreds of thousands of people)

>

Probabilistic Programming in probabilistic programming languages
(tens of thousands of people)

Interview study

Braun, V., Clarke, V.: **Using thematic analysis in psychology**. Qualitative research in psychology 3(2), 77-101 (2006)

Borghouts, J., Gordon, A.D., Sarkar, A., O'Hara, K.P., Toronto, N.: **Somewhere around that number: An interview study of how spreadsheet users manage uncertainty.** arXiv preprint arXiv:1905.13072 (2019)

Research Questions and Method

What types of uncertainty do spreadsheet users deal with?

How do they manage these various types of uncertainty?

We conducted semi-structured interviews of 11 participants, who walked us through their spreadsheets.

We analysed the audio transcripts of these interviews, identifying six qualitatively different **types of uncertainty**, and six **categories of strategies** used to cope with the uncertainty.

(Thematic analysis is not quantitative.)

Spreadsheet Roles

Database: *'The vast bulk of users use it as a way of storing, analysing, capturing data. (…) it becomes the record of the corporate memory'* (P1).

Template to repeat analysis: *'Even sometimes when you use past examples, you still think: There's still something not quite right about that figure'* (P4).

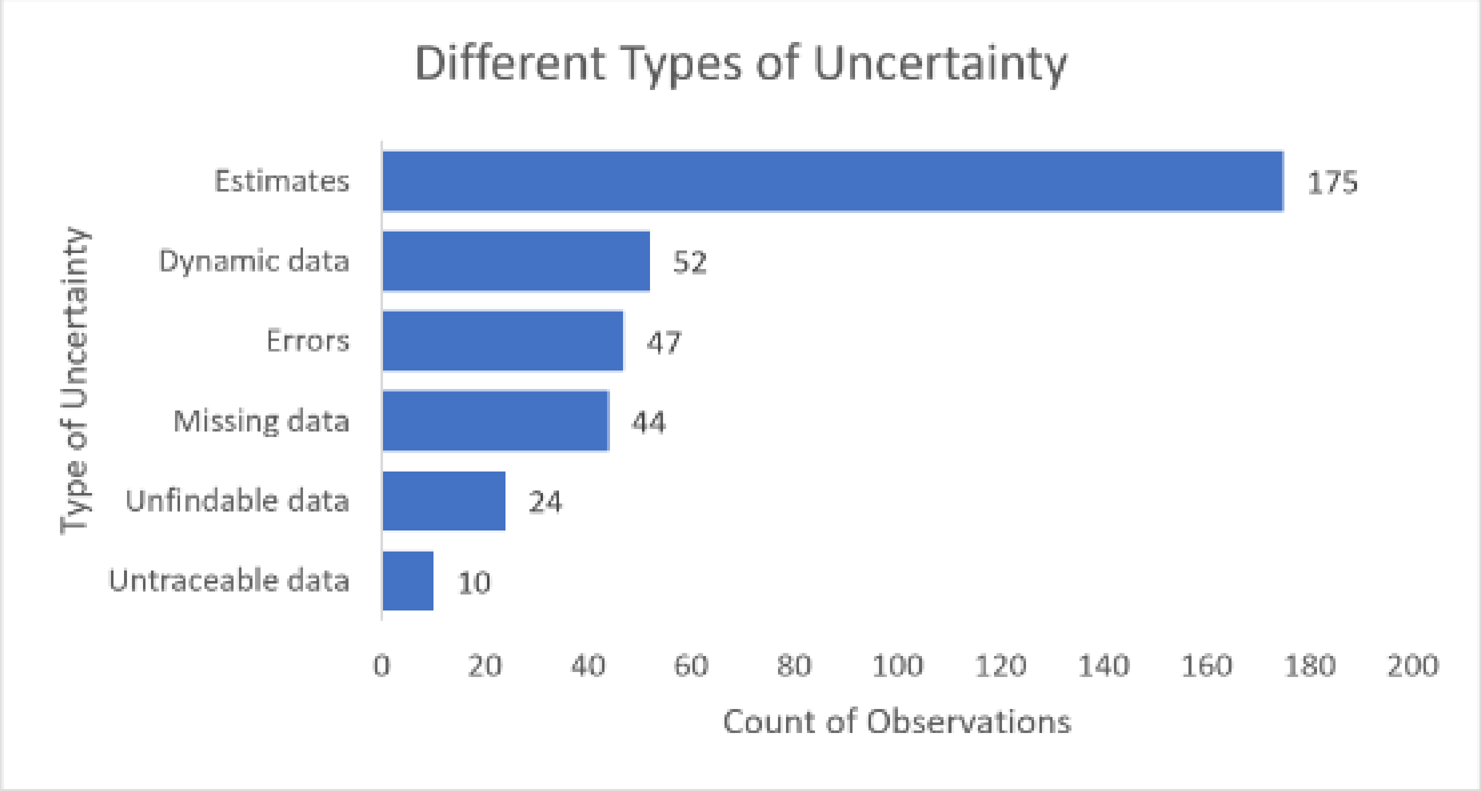
Calculator and analysis tool: P2 and P5 used spreadsheets to build prediction models with formulas.

Notepad: P4 used spreadsheets to get a better understanding of how long a project would approximately take.

Element	Code	Nature of Work	Time Days	Dependencies	Confidence %
Main Site	MS::C	Content	2	None	80
Vocabulary Builder	C::VB::C	Content	3	None	10

Data exploration tool: P10 and P11 dealt with large datasets and used statistical analysis tools to analyse these.

Observed Uncertainty in Spreadsheets



An 'observation' is a concrete instance in our transcripts where a participant mentioned encountering a type of uncertainty in a spreadsheet. We recorded 32 observations on average per participant (min: 16, max 55).

Estimates were the most common type of uncertainty among participants.

Approximations where the precise value is not known, such as the expected profit of a project.

In a world of **known risks**, everything, including the probabilities is known for certain.

Here, statistical thinking and logic are sufficient to make good decisions.

In an **uncertain world**, not everything is known, and one cannot calculate the best option.

Here, good rules of thumb and intuition are also required.

Distinction due to Frank Knight *Risk, Uncertainty, and Profit* (1921)
Images and quotation from Gerd Gigerenzer *Risk Savvy* (2014)

Probabilistic versus Possibilistic

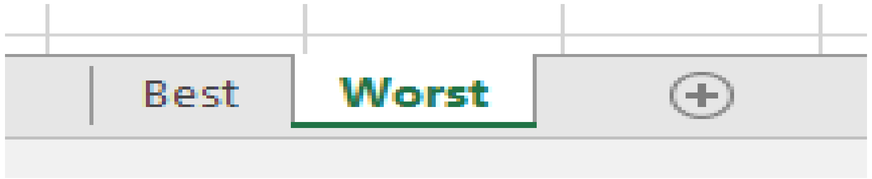
Found widespread use of **estimates** in spreadsheets, where precise value is unknown.

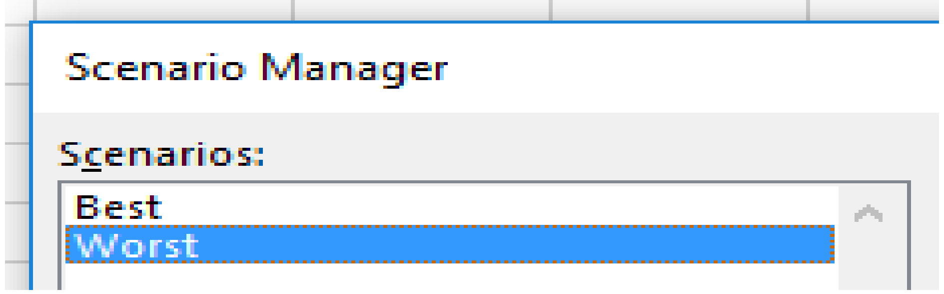
Probabilistic programming helps when the **risks are known**, or can easily be computed. (This situation is **Bayesian uncertainty**.)

Still, won't help with **Knightian uncertainty**, when probabilities not practically known.

A heuristic is to consider **multiple scenarios** – **possibilistic** programming.

	A	B	C
1		*Best*	*Worst*
2	**Widgets**	75000	125000
3	**Cost**	$6.50	$7.50
4	**Total**	$487,500.00	$937,500.00
5	**Budget**	$750,000.00	$750,000.00
6	**Over?**	FALSE	TRUE





Uncertain Values

Based on Streit's proposal of uncertain values that propagate through a sheet

Alexander Streit, "Encapsulation and abstraction for modeling and visualizing information uncertainty" (March 2008), Queensland University of Technology

Clara’s Budget

Clara is an end-user confident to model with simple formulas

	A	B	C	D	E	F
1					**Budget for Jan 2018**	
2						
3					**Income**	2000
4						
					Expenses	
					Rent	1100
					Commute	85
					Sofa!	700
					Utilities	
					Total expenses	1885
					Balance	115

E1 = "Budget for Jan 2018"
E3 = "Income"; F4 = 2000
E5 = "Expenses"
E6 = "Rent"; F6 = 1100
E7 = "Commute"; F7 = 85
E8 = "Sofa!"; F8 = 700
E9 = "Utilities"; F9 = ???
E10 = "Total expenses"; F10 = SUM(F6:F9)
E12 = "Balance"; F12 = F4−F10

Streit's Uncertain Values in Cells

1. The user can input uncertainty information into cells.
    - Input can be via a textual notation, such as a numeric range, or as parameters of a probability distribution.
    - Assisted by some UI that constructs an underlying formula.
2. Uncertainty information propagates through formula evaluation.
3. The presence of uncertainty information is indicated within cell.
    - A most likely value might be displayed, for example.

Essentially, a **computational effect** for uncertainty in spreadsheets

Streit considered estimates, intervals, probability distributions

Probabilistic (weighted sets of values)

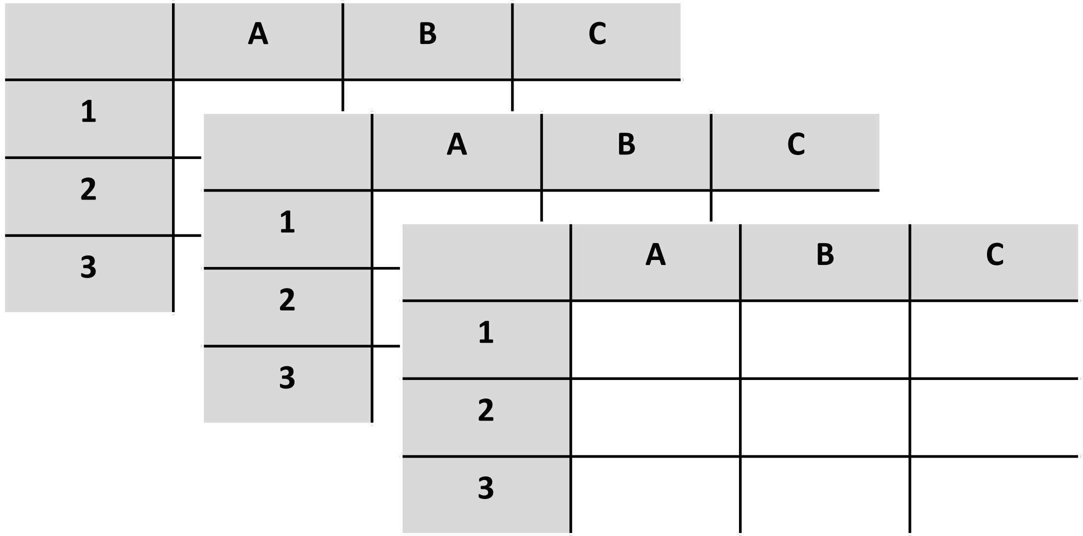
The function DIST.TRIANG(*a*,*b*,*c*) constructs a value distributed according to a **triangular distribution**, with lower bound *a*, upper bound *b*, and mode *c*, where $a < b$ and $a \leq c \leq b$.

E1 = "Budget for Jan 2018"
E3 = "Income"; F4 = 2000
E5 = "Expenses"
E6 = "Rent"; F6 = 1100
E7 = "Commute"; F7 = 85
E8 = "Sofa!"; F8 = 700
E9 = "Utilities"; F9 = DIST.TRIANG(50,150,100)
E10 = "Total expenses"; F10 = SUM(F6:F9) // value: DIST.TRIANG(1935,2035,1985)
E12 = "Balance"; F12 = F4−F10 // value: DIST.TRIANG(−35,65,15)
E13="Chance of overdraft"; F13=F12<0 // value: 25%

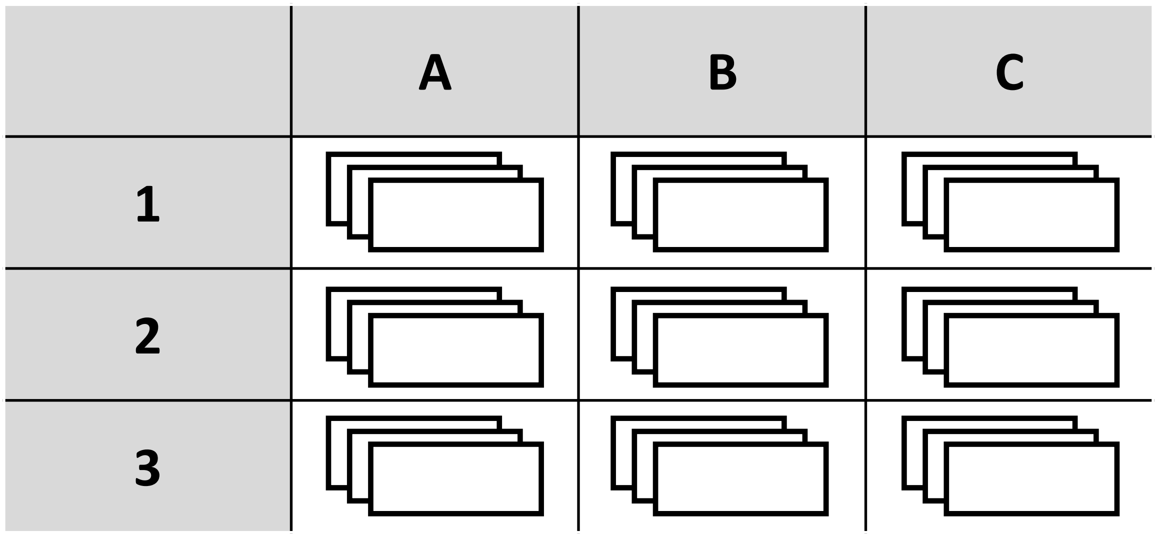
Idea: Store Array of Scenarios in Each Cell

Means the same as mashing F9 on a spreadsheet with random values…

Acts like this



Stored like this



…except each sequence of "random" numbers is repeatable

How Many Valentine Cards to Order?

Year	Sold			Cost per card	Price per card	
2010	833924			$1.47	$1.99	
2011	1299434					
2012	760148		Ordered	Cost	Gross Profit	Net Profit
2013	1083010		400000	$588,000.00	$796,000.00	$208,000.00
2014	923113		550000	$808,500.00	$1,094,500.00	$286,000.00
2015	1216995		700000	$1,029,000.00	$1,393,000.00	$364,000.00
2016	861598		850000	$1,249,500.00	$1,691,500.00	$442,000.00
			1000000	$1,470,000.00	$1,983,808.83	$513,808.83
Average	996889		1150000	$1,690,500.00	$1,983,808.83	$293,308.83
			1300000	$1,911,000.00	$1,983,808.83	$72,808.83

Today, we'd model 2017 demand by exactly the average of previous years' demand.

How Many Valentine Cards to Order?

Year	Sold			Cost per card	Price per card	
2010	833924			$1.47	$1.99	
2011	1299434					
2012	760148		Ordered	Cost	Gross Profit	Net Profit
2013	1083010		400000	$588,000.00	~$795,792.39	~$207,792.39
2014	923113		550000	$808,500.00	~$1,092,363.04	~$283,863.04
2015	1216995		700000	$1,029,000.00	~$1,379,405.37	~$350,405.37
2016	861598		850000	$1,249,500.00	~$1,634,464.79	~$384,964.79
			1000000	$1,470,000.00	~$1,823,660.99	~$353,660.99
Average	996889		1150000	$1,690,500.00	~$1,929,667.49	~$239,167.49
Stdev	205648		1300000	$1,911,000.00	~$1,971,125.46	~$60,125.46
Demand	~996899		=DIST.NORM(N,B12,B13)			

After (850000 row)

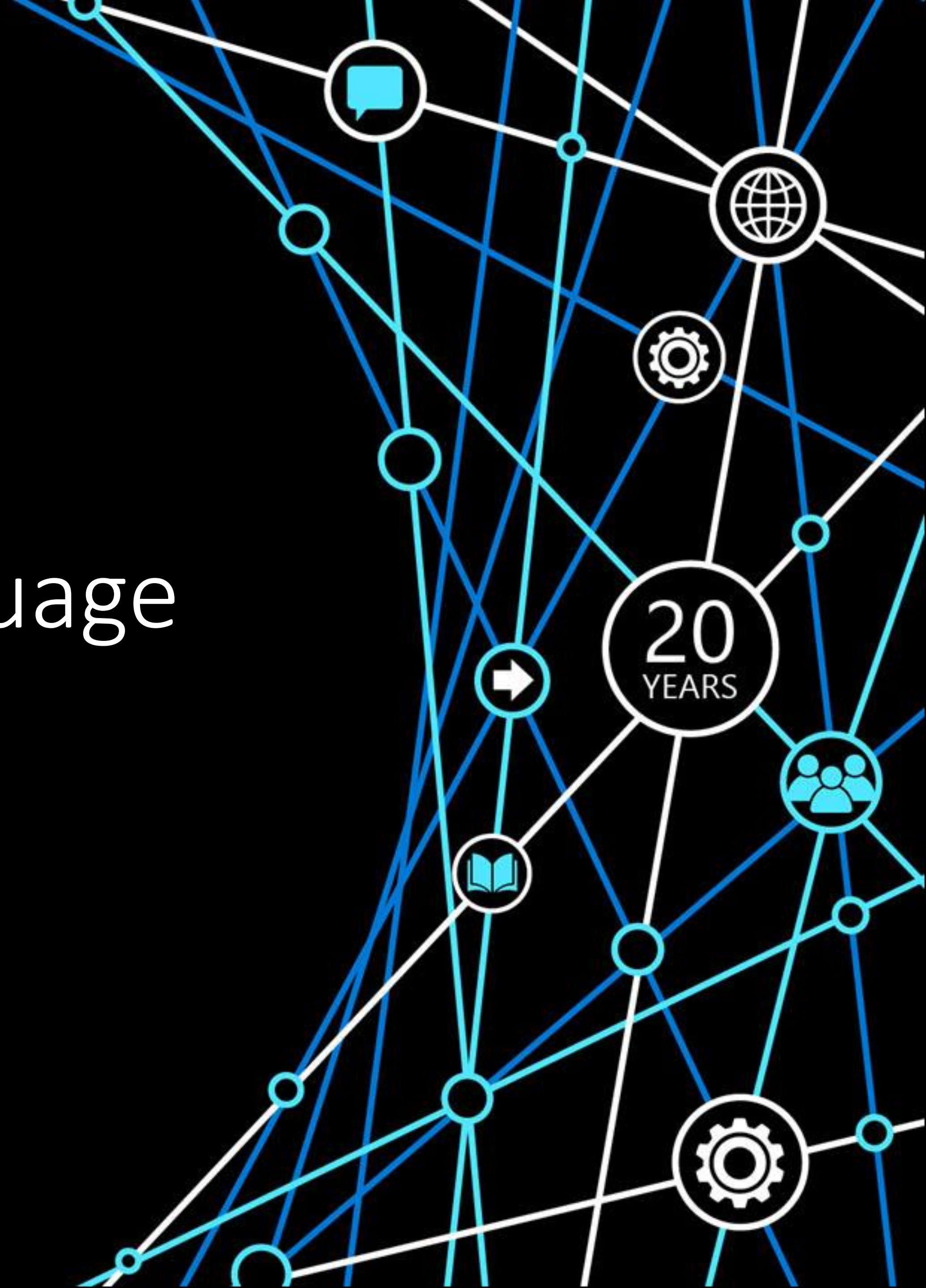
Before (1000000 row)

Expected net profits are lower when we model distributions.
Surprisingly, the optimal decision is to order below the average demand.

Microsoft

Project Yellow:
Excel as a Programming Language

A Collaboration between MSR and Excel



Aim

Remove the “glass ceiling” that limits the scope and reach of what a domain expert can do with Excel:

- **Make Excel functions reflect the abstractions of our end users**, by allowing end-users to define new functions using an ordinary worksheet.

- **Make Excel’s data values reflect the datatypes of our end users’ domains**, by adding arrays, vectors, records, and even domain-specific data types implemented by third parties.

So far, we've taken Excel beyond text and numbers

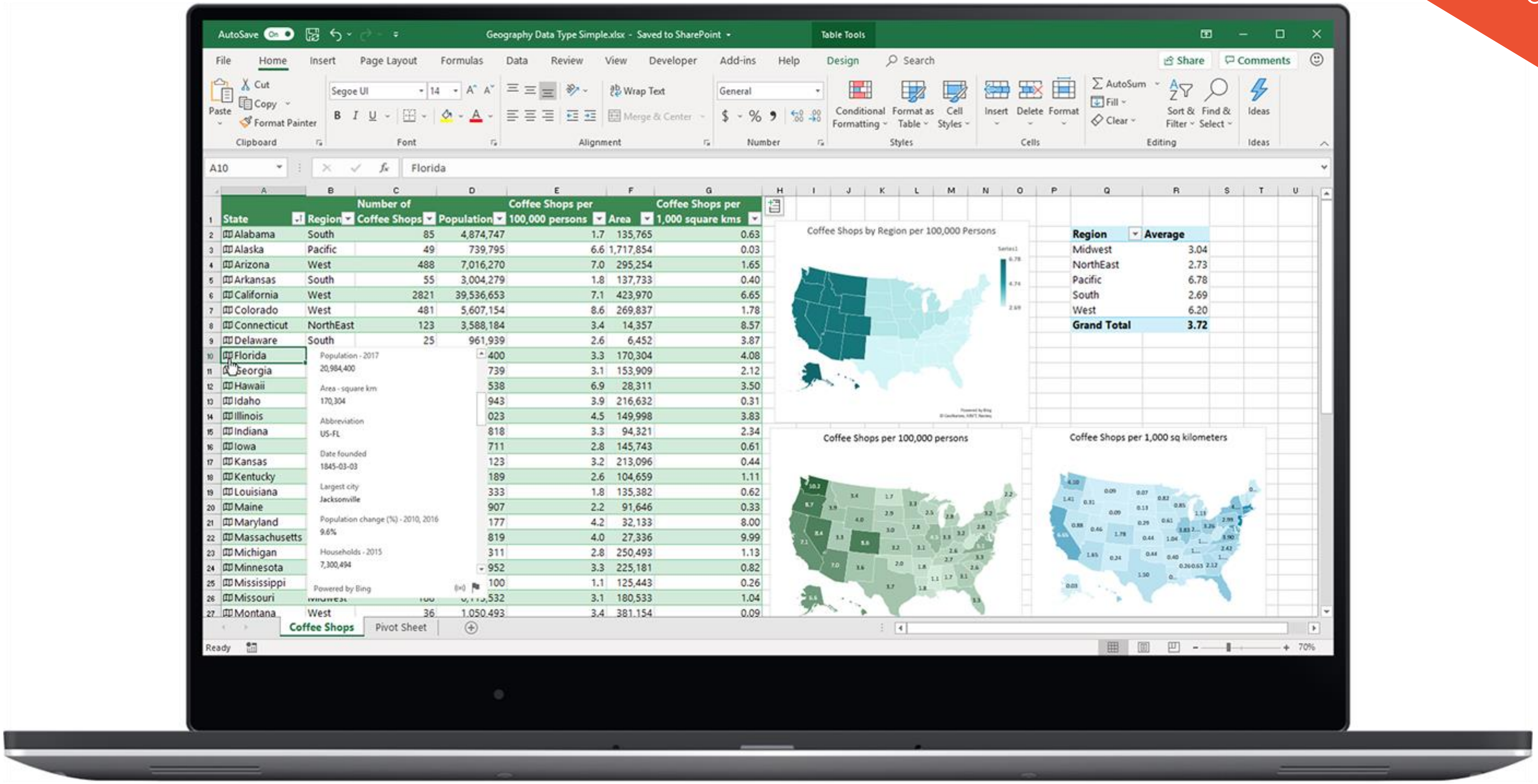
Ignite 2018

Dynamic Arrays

Sort your data		Clean your data		Filter your data		
Names	Sorted	Names	No Dup	Names	Choice	Veg Only
Mary		Peter		Mary	Meat	
Peter		Jane		Peter	Veg	
John		Jane		John	Veg	
Simon		Alison		Simon	Meat	
Charles		Simon		Charles	Veg	
Jane		Mary		Jane	Meat	
Alison		Simon		Alison	Veg	
		Charles				
		Charles				
		John				

So far, we've taken Excel beyond text and numbers

Ignite 2018



aka.ms/popl blog

aka.ms/CalcIntel

Influencing mainstream software—Applying programming language research ideas to transform spreadsheets

Spreadsheets are the world's most widely used programming language, by several orders of magnitude. We asked ourselves whether it would be possible to apply programming language research ideas to make spreadsheets a better programming language? If we could, that would empower a huge user community to do more. One of the joys of working at Microsoft Research is the ability to directly influence mainstream software technologies – in this case, Microsoft Excel. And the Excel...

January 2019

Microsoft Research Blog

Judith Borghouts
User researcher

Dany Fabian
Principal Research Software Development Engineer

Andy Gordon
Principal Research Manager

Samin Ishtiaq
Principal Research Software Development Engineer

Anusha Iyer
UX Designer

Simon Peyton Jones
Principal Researcher

Advait Sarkar
Postdoctoral researcher

Neil Toronto
Senior RSDE

Juliana Vicente Franco
Research Software Engineer

Jack Williams
Visiting Researcher

The Next Chapter for EUPP: Trustworthy Communication

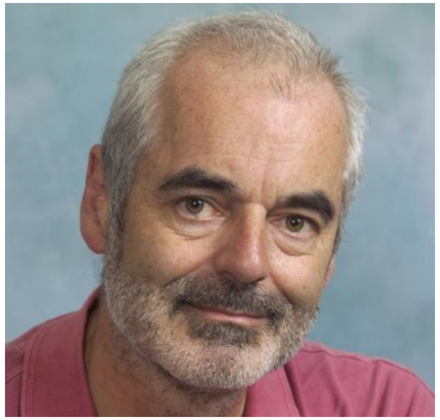
End-users will increasingly generate or receive uncertainty values

From probabilistic programming, from AI algorithms, from experts

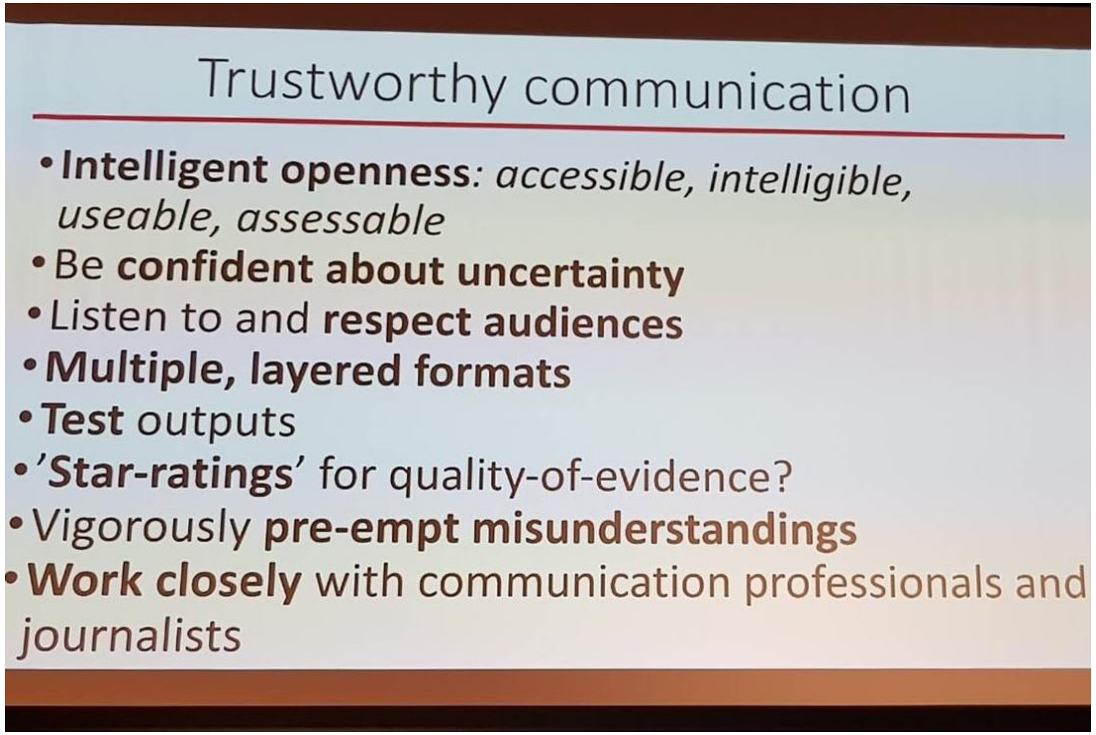
Helping end users comprehend the uncertainty has been **out of scope**.

Trustworthy Communication

Uncertainty is an inherent part of knowledge, and yet in an era of contested expertise, many shy away from openly communicating their uncertainty about what they know, fearful of their audience's reaction.



Talk by David Spiegelhalter
(StanCon, Cambridge, August 2019)
(paper)



Communicating uncertainty about facts, numbers and science

Anne Marthe van der Bles[1,2], Sander van der Linden[1,2], Alexandra L. J. Freeman[1], James Mitchell[3], Ana B. Galvao[3], Lisa Zaval[4] and David J. Spiegelhalter[1]

[1]Winton Centre for Risk and Evidence Communication, Department of Pure Mathematics and Mathematical Statistics, and [2]Cambridge Social Decision-Making Lab, Department of Psychology, University of Cambridge, Cambridge, UK
[3]Warwick Business School, University of Warwick, Coventry, UK
[4]Department of Psychology, Columbia University, New York, NY, USA

AMvdB, 0000-0002-7953-9425; SvdL, 0000-0002-0269-1744; ALJF, 0000-0002-4115-161X; JM, 0000-0003-0532-4568; ABG, 0000-0003-3263-9450; DJS, 0000-0001-9350-6745

Uncertainty Visualization

It's a Moral Imperative

We should…

present **well-calibrated** uncertainty that **cannot be ignored** in ways people can **actually understand**

Matthew Kay (Computer Science, University of Michigan)

http://www.mjskay.com/ (talk at MSR Faculty Summit)

My talk – End-User Probabilistic Programming

Probabilistic programming for spreadsheet formula authors

- Flaw of averages – good reason for EUPP
- Risk versus uncertainty – fundamental limitation of EUPP
- Trustworthy communication – next chapter for EUPP research