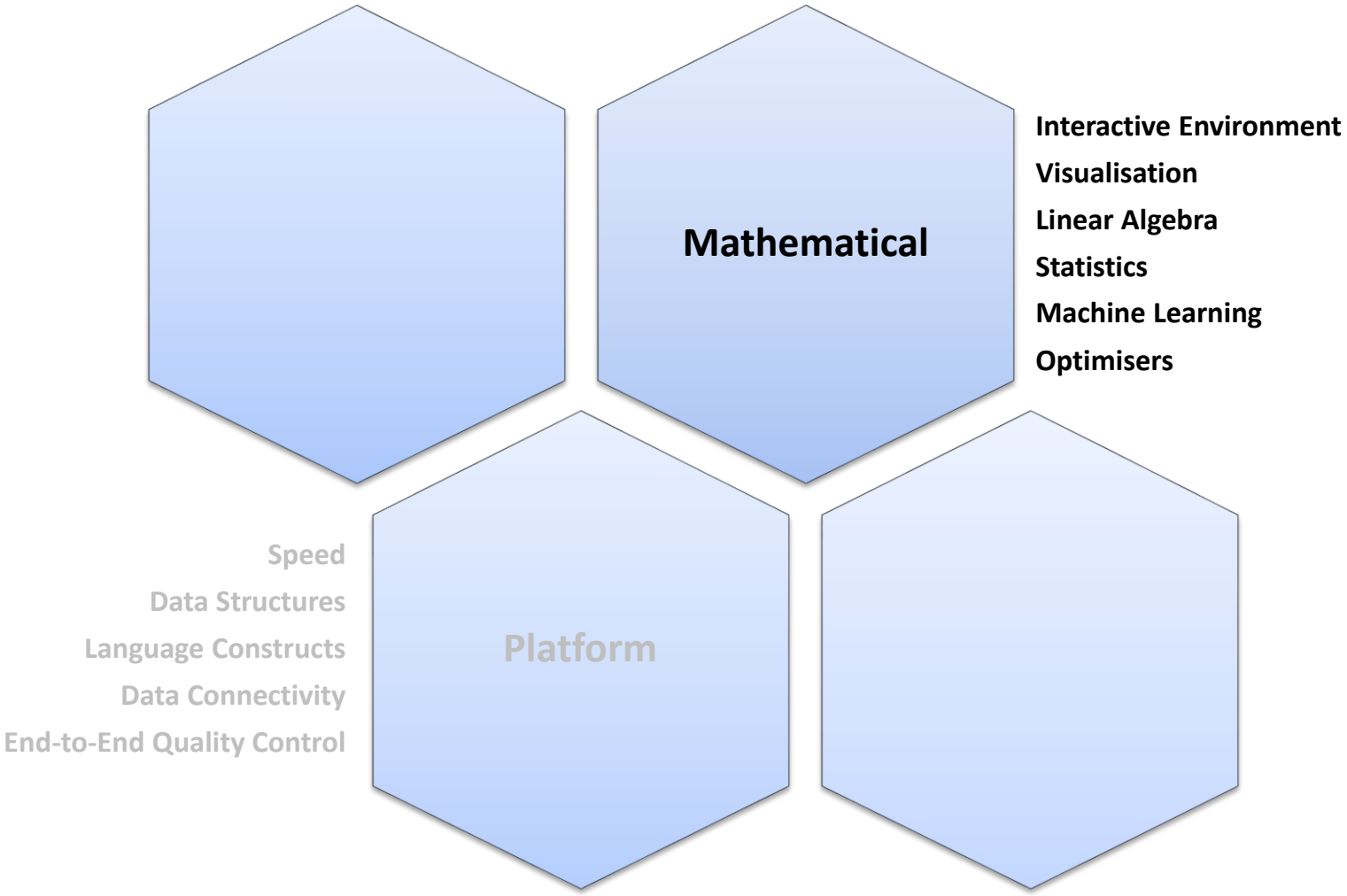
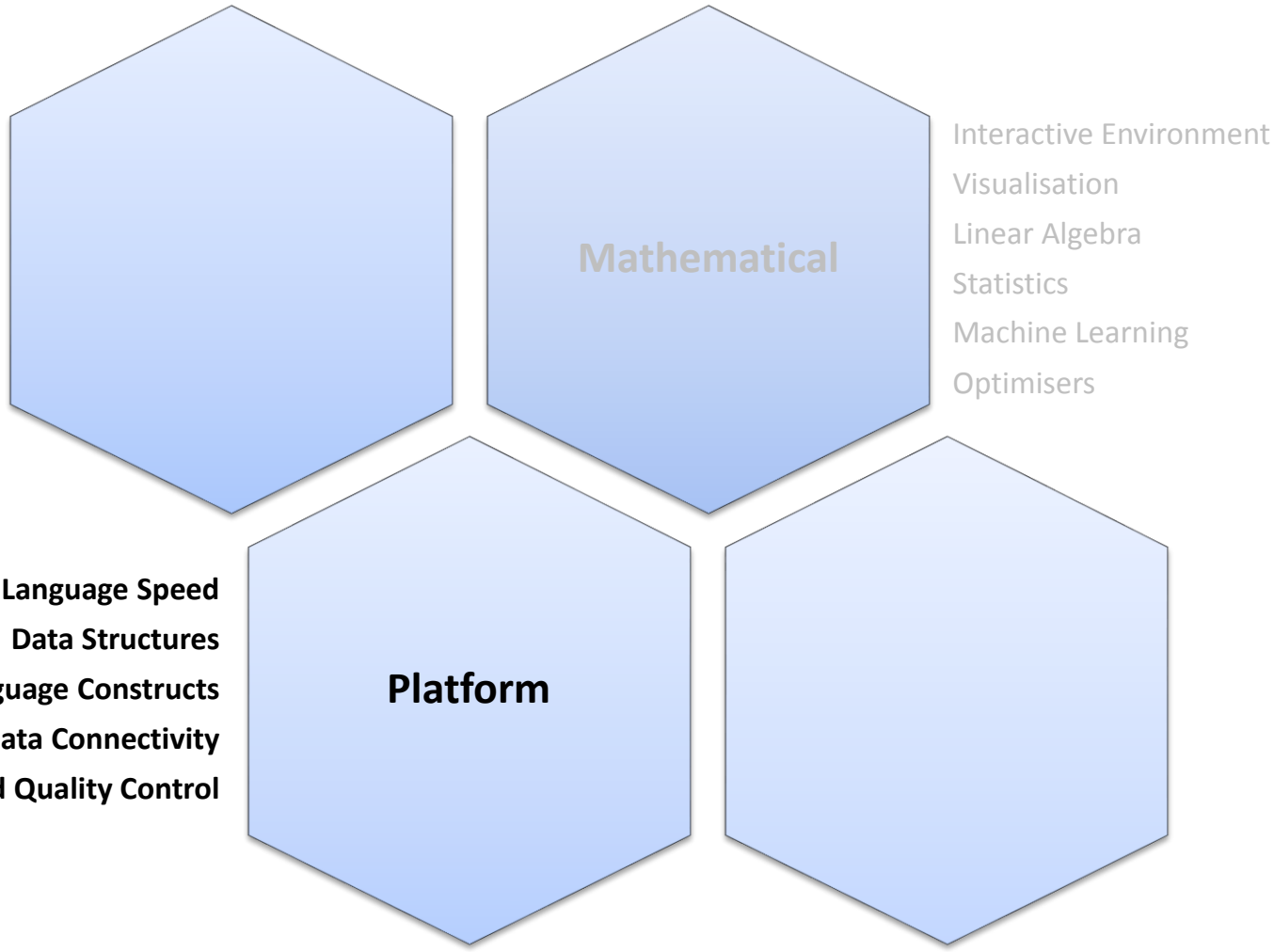
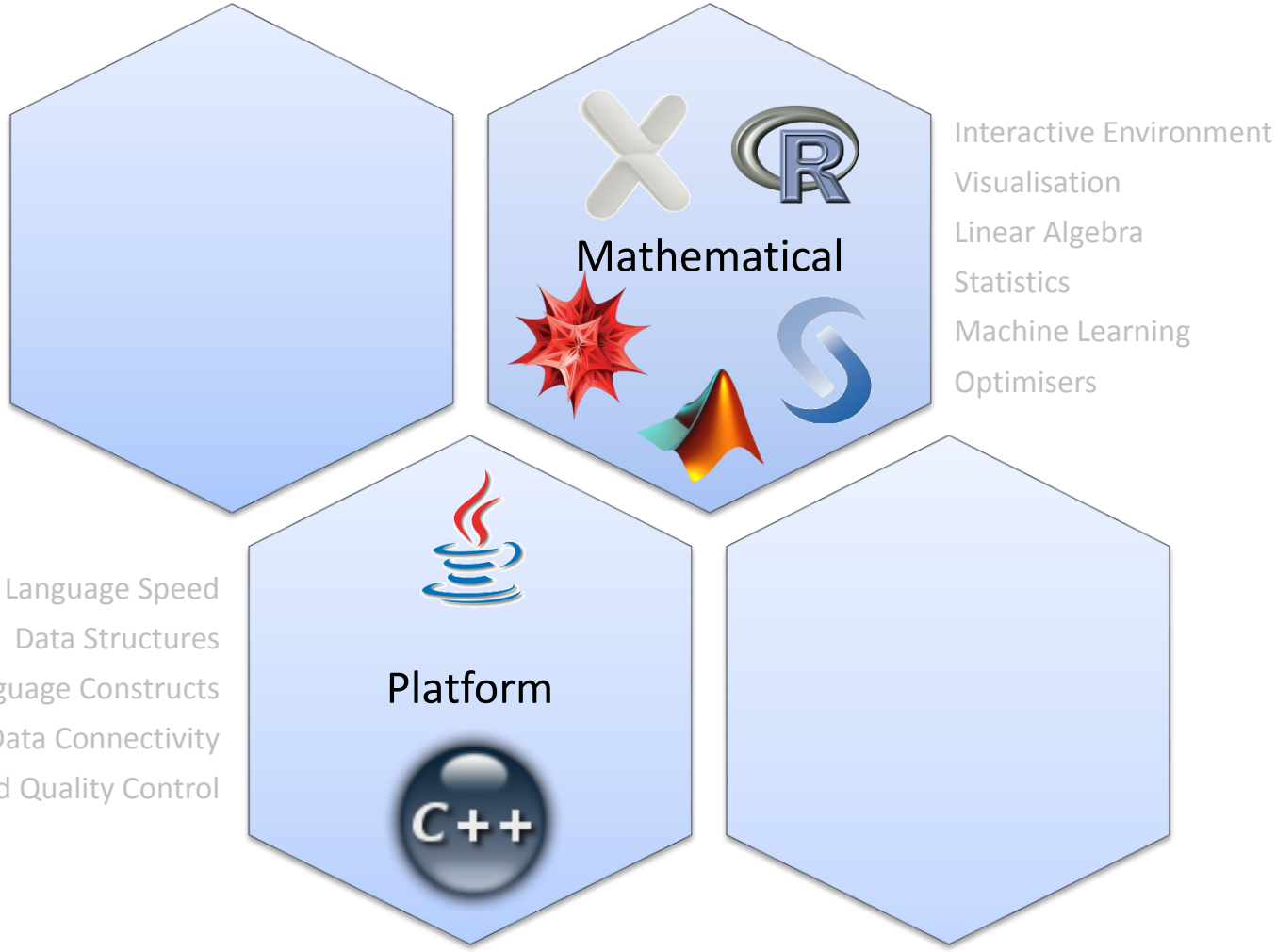

Data Science in Clojure
@edmundjackson


ALAN TURING


The weak shall perish
Dum spiramus
tuebimur
The strong shall triumph
Emitte lucem et
veritatem
Ftn
C++
Ex animo
Prevail over my sword
Esto perpetue
And I shall grant it to thee

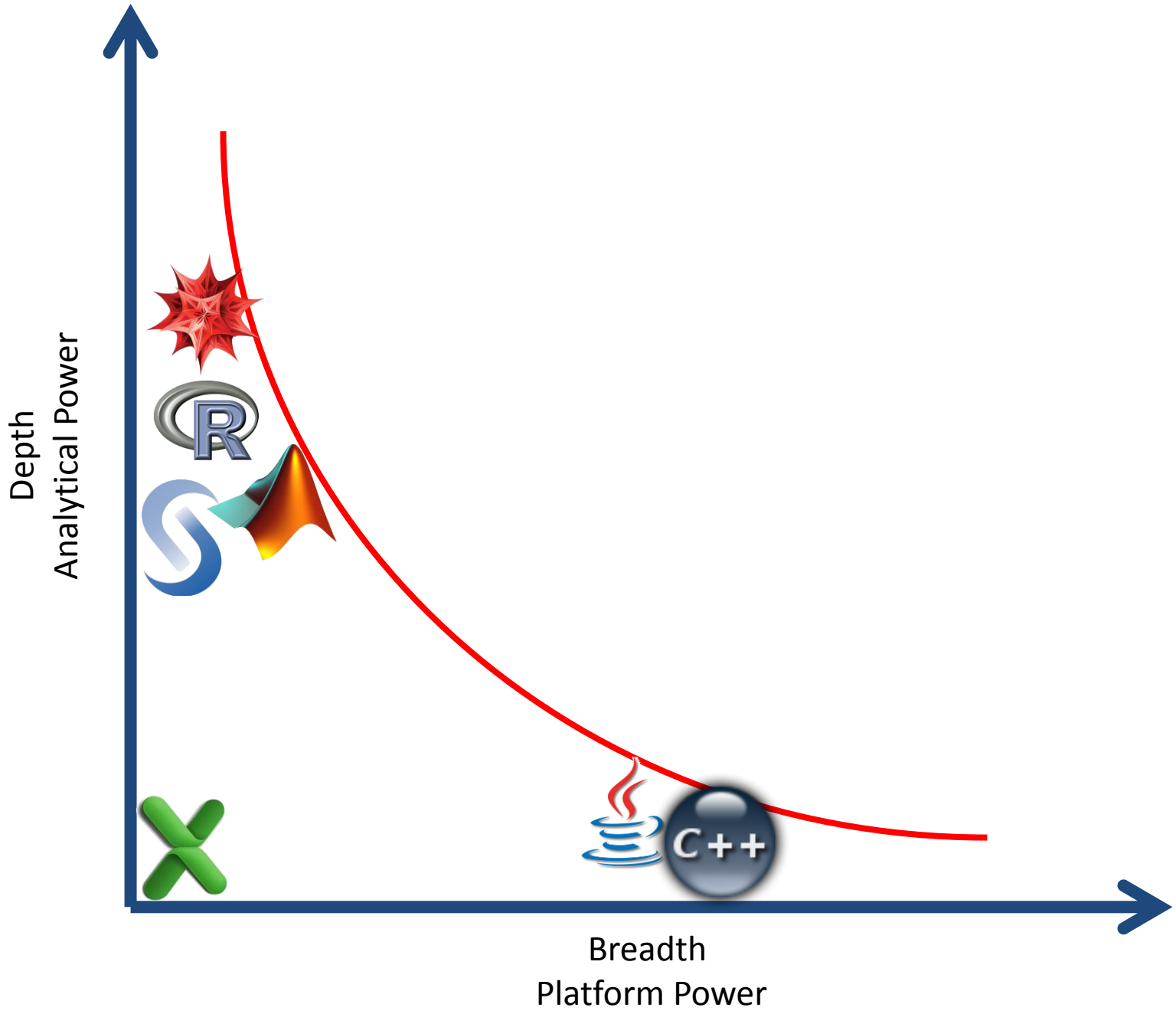

Mathematical
Interactive Environment
Visualisation
Linear Algebra
Statistics
Machine Learning
Optimisers
Platform
Speed
Data Structures
Language Constructs
Data Connectivity
End-to-End Quality Control

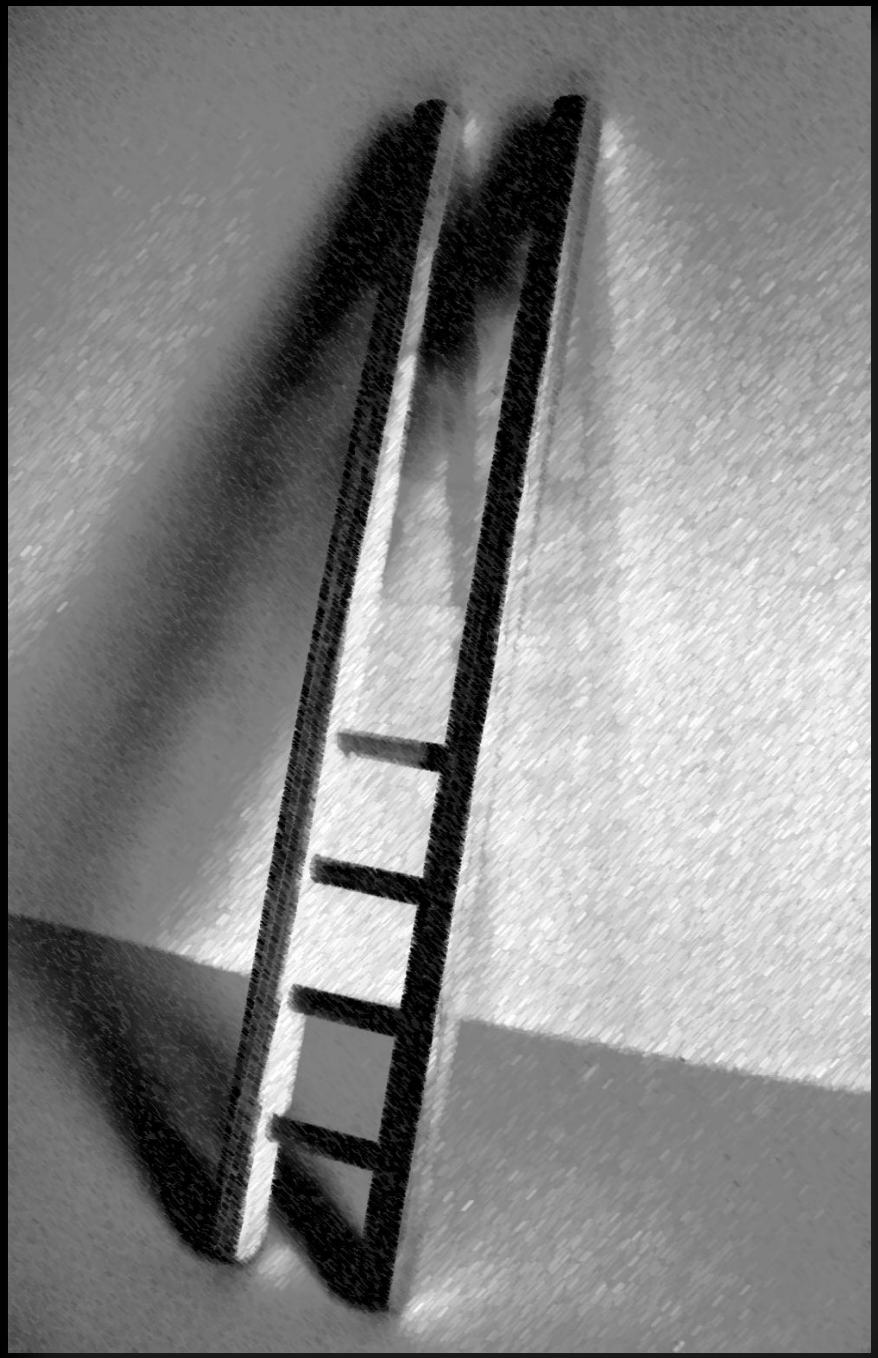

Mathematical
Interactive Environment
Visualisation
Linear Algebra
Statistics
Machine Learning
Optimisers
Native Language Speed
Data Structures
Language Constructs
Data Connectivity
End-to-End Quality Control
Platform

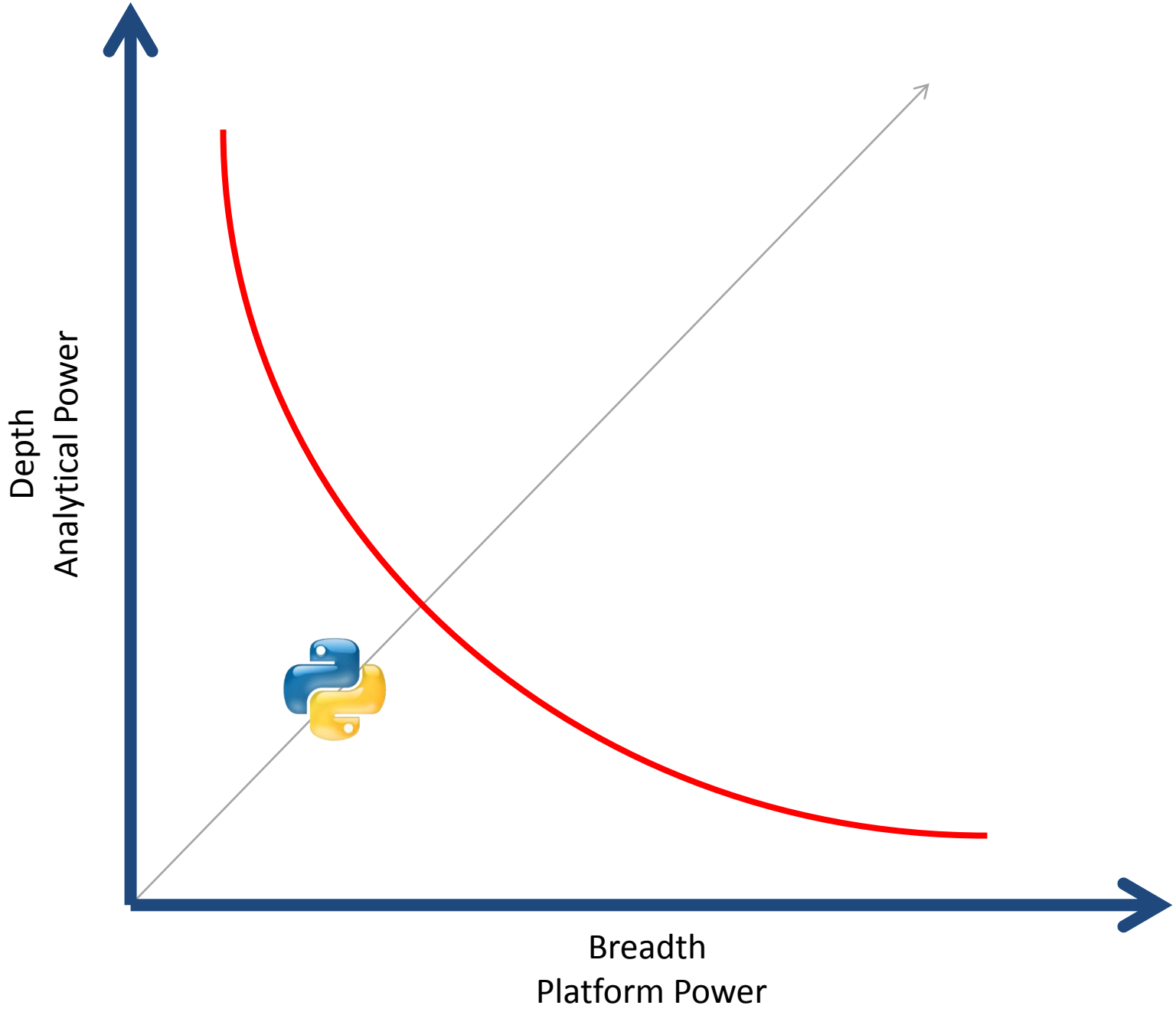


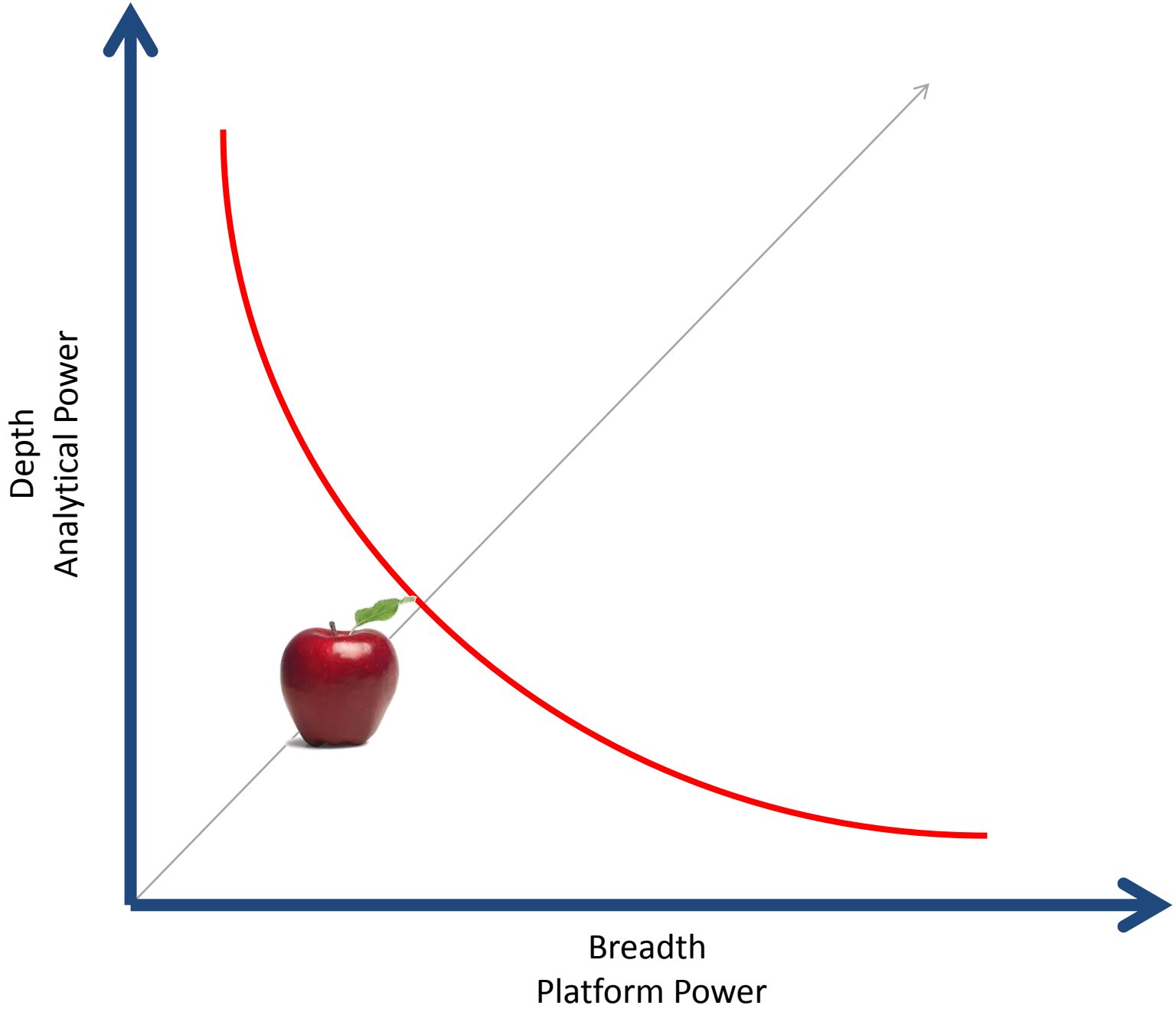
Interactive Environment
Visualisation
Linear Algebra
Statistics
Machine Learning
Optimisers

Native Language Speed
Data Structures
Language Constructs
Data Connectivity
End-to-End Quality Control


Depth
Analytical Power
R
C++
Breadth
Platform Power

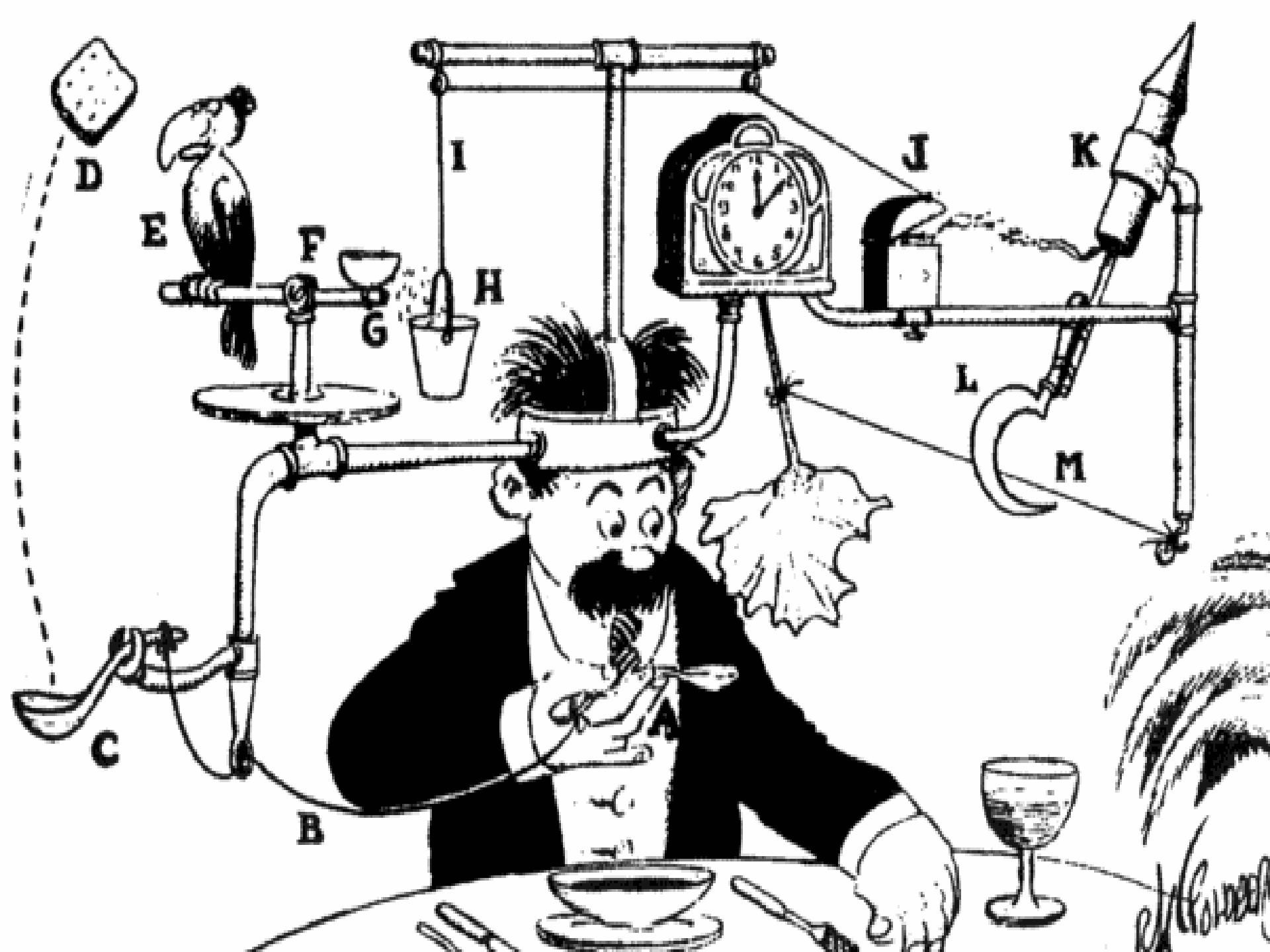




Depth
Analytical Power
Breadth
Platform Power


Depth
Analytical Power
Breadth
Platform Power

$$x = x + 1$$

$$X = X + 1$$

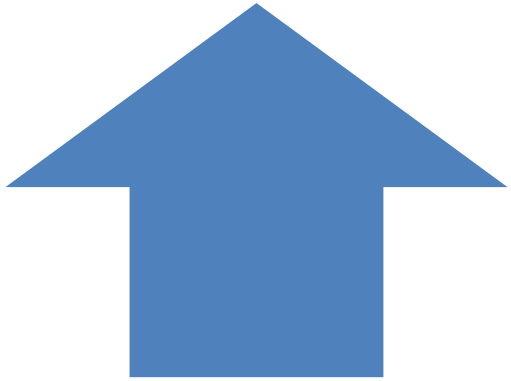

D
E
F
G
H
I
J
K
L
M
A
B
C

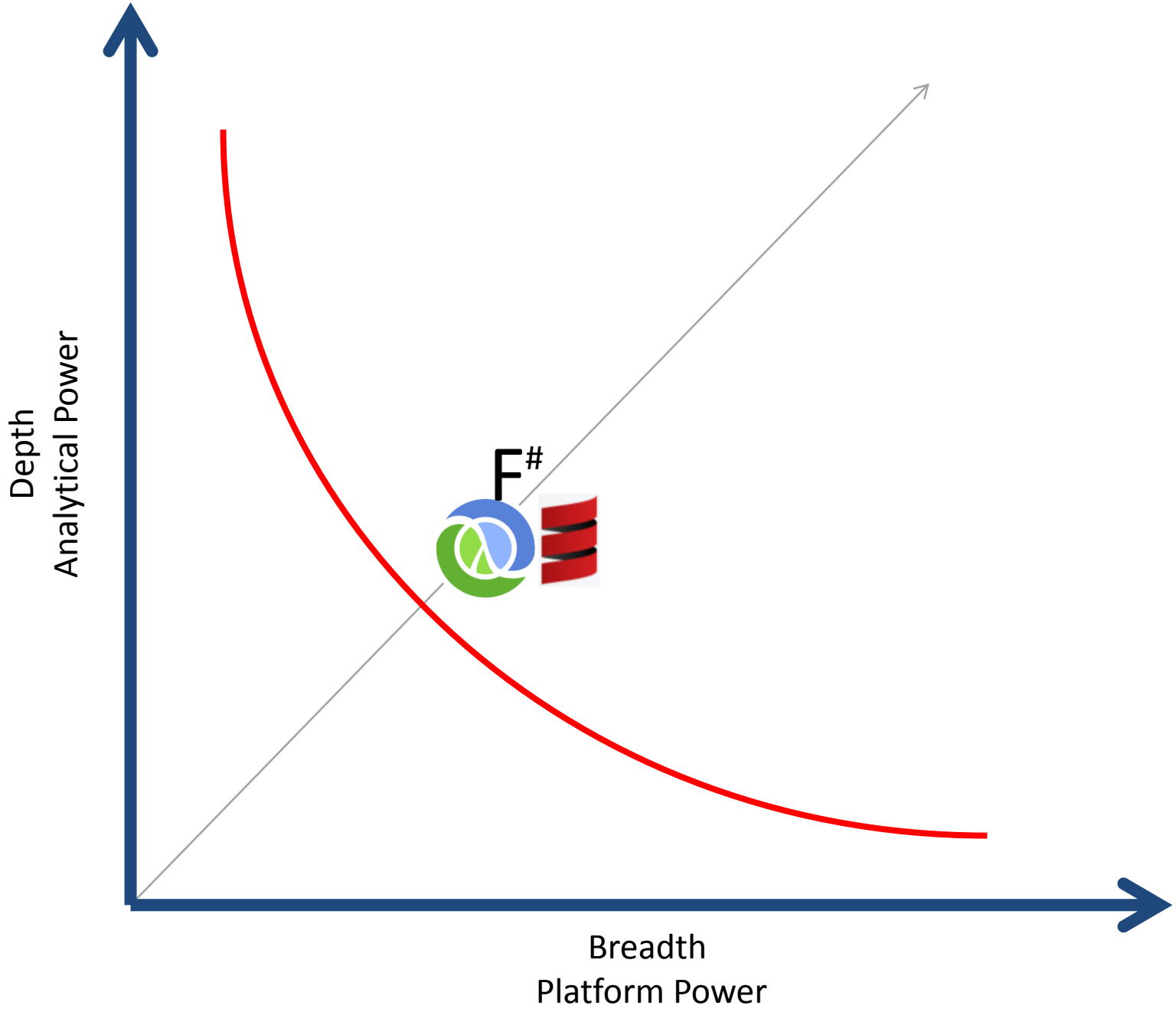
Function Programming

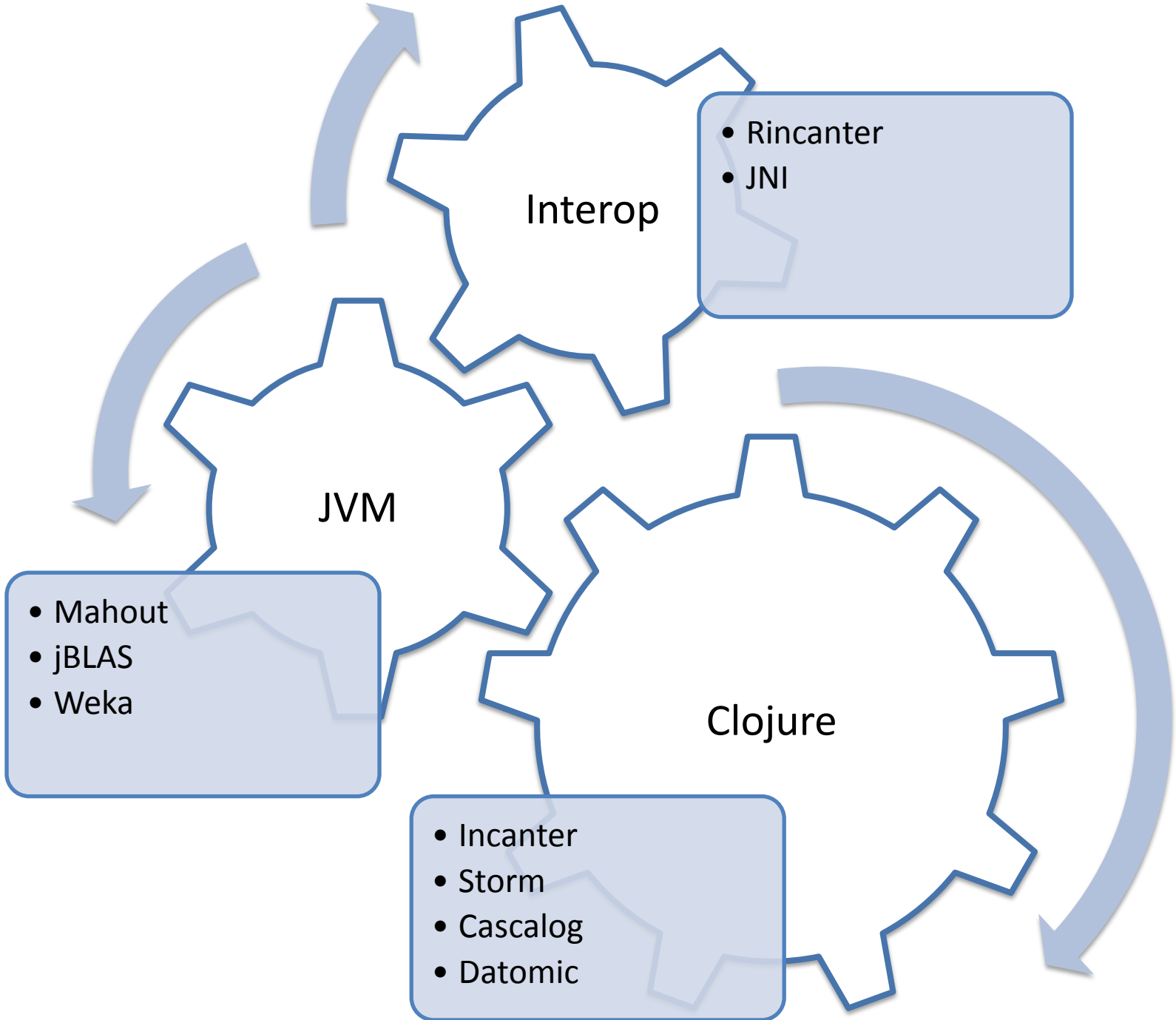
- Immutability
- Close to the Maths
- Serious Data Abstractions

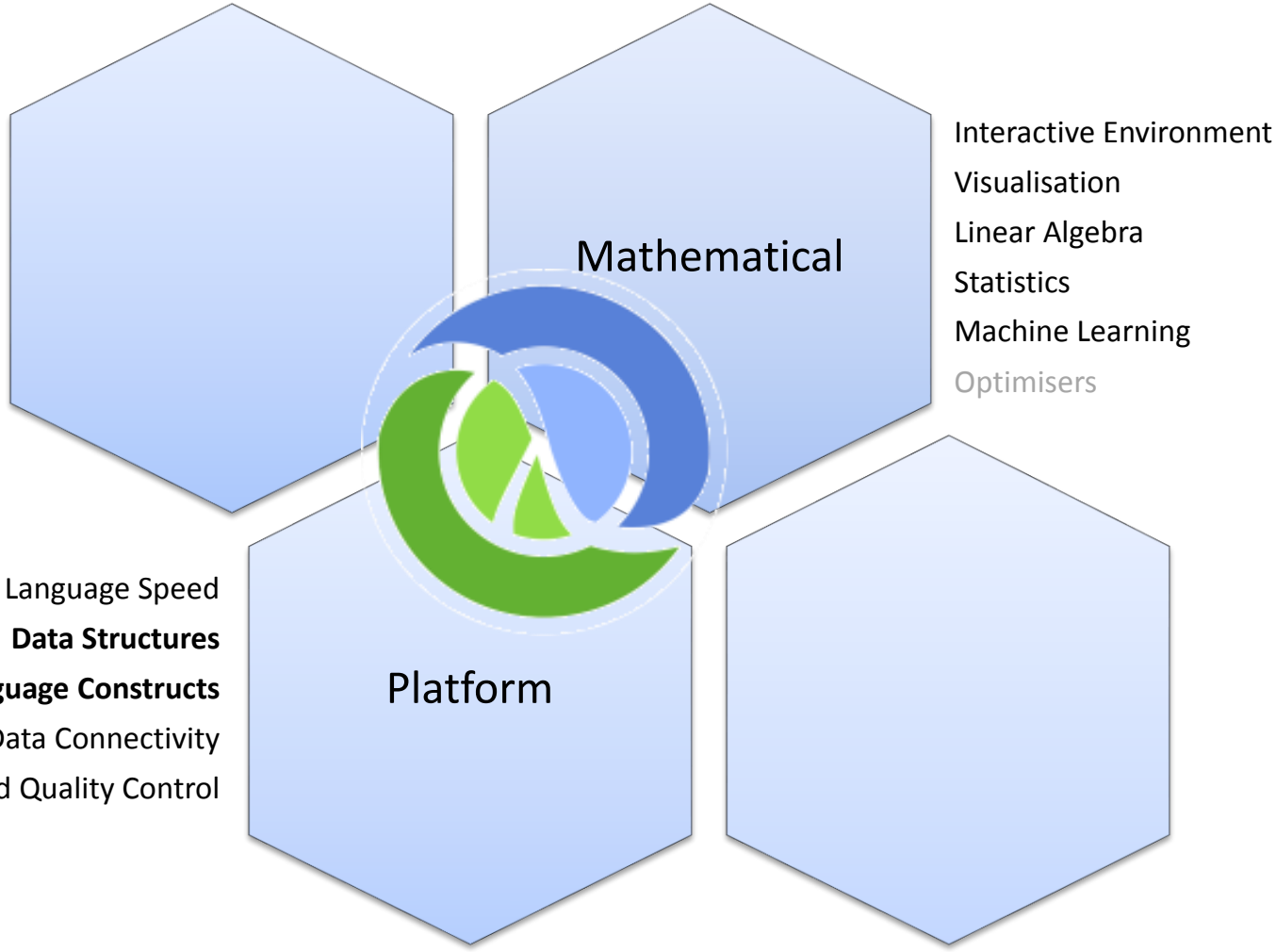
Platform

- In Business Process
- Data Sources
- Testing, CI, Deployment




Depth
Analytical Power
F#
Breadth
Platform Power


Interop
• Rincanter
• JNI
JVM
• Mahout
• jBLAS
• Weka
Clojure
• Incanter
• Storm
• Cascalog
• Datomic


Mathematical
Interactive Environment
Visualisation
Linear Algebra
Statistics
Machine Learning
Optimisers
Platform
Native Language Speed
Data Structures
Language Constructs
Data Connectivity
End-to-End Quality Control

Part 2: Entropy Always Wins in the End

60%

40%

H

60%

P =0.6

H H

60% 60%

P =0.6 * 0.6

= 0.36

60% 40%

H H H

60% 60% 60%

P =0.6 * 0.6 * 0.6

= 0.216

H H H H H H H H H H

60% 60% 60% 60% 60% 60% 60% 60% 60% 60%

P =0.6 * 0.6 * 0.6 * 0.6 * 0.6 * 0.6 * 0.6 * 0.6 * 0.6 * 0.6

= 6E-3

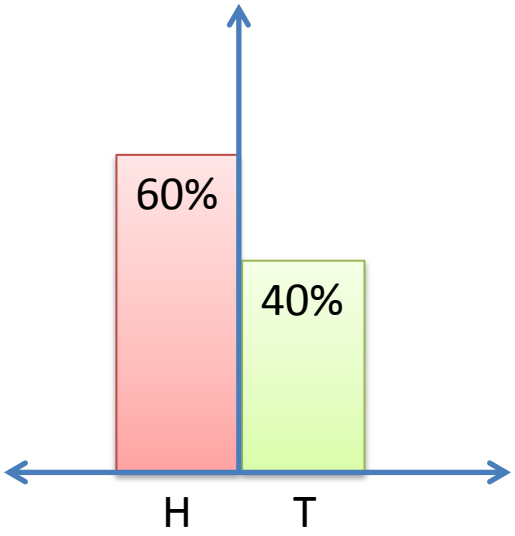
H H H H H T H H H H

60% 60% 60% 60% 60% 40% 60% 60% 60% 60%

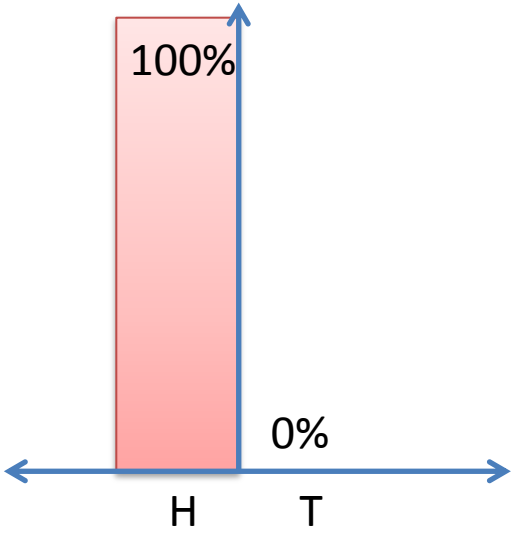
P =0.6 * 0.6 * 0.6 * 0.6 * 0.6 * 0.4 * 0.6 * 0.6 * 0.6 * 0.6

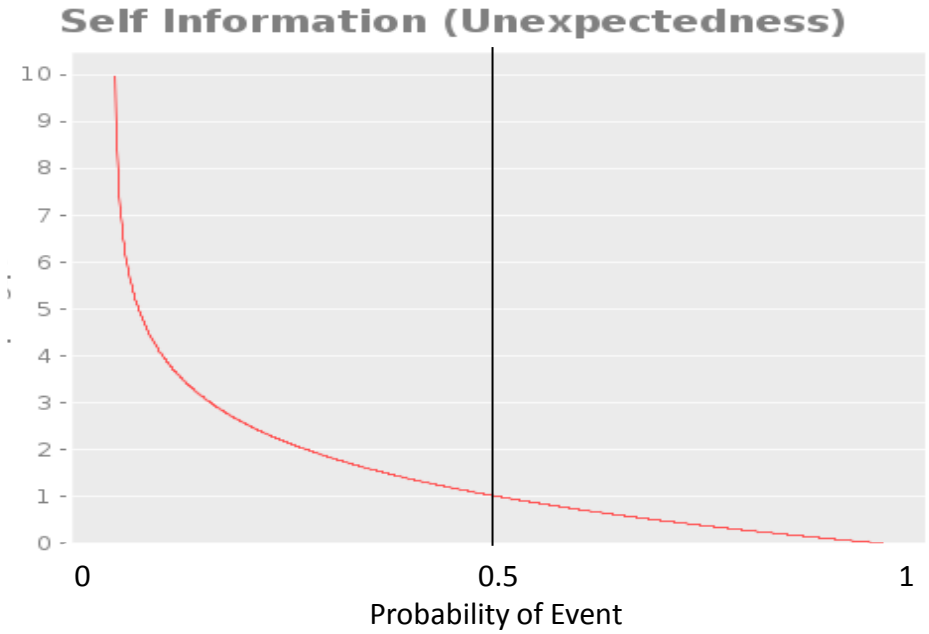
= 4E-3

Generative Distribution (what you expected)

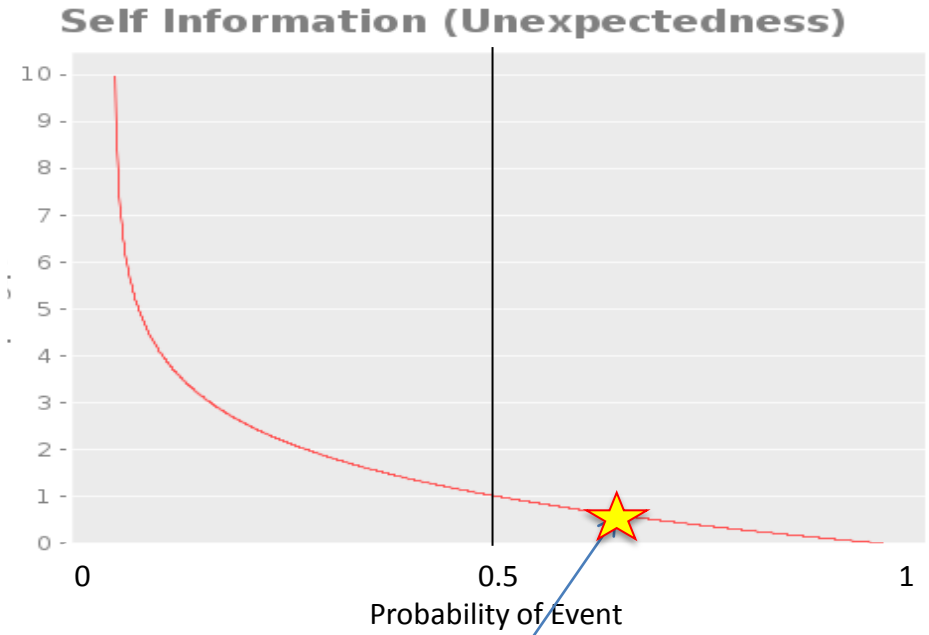


Empirical Distribution (what you got)

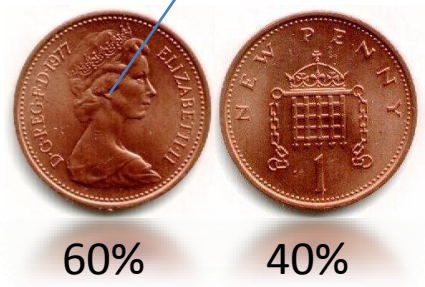


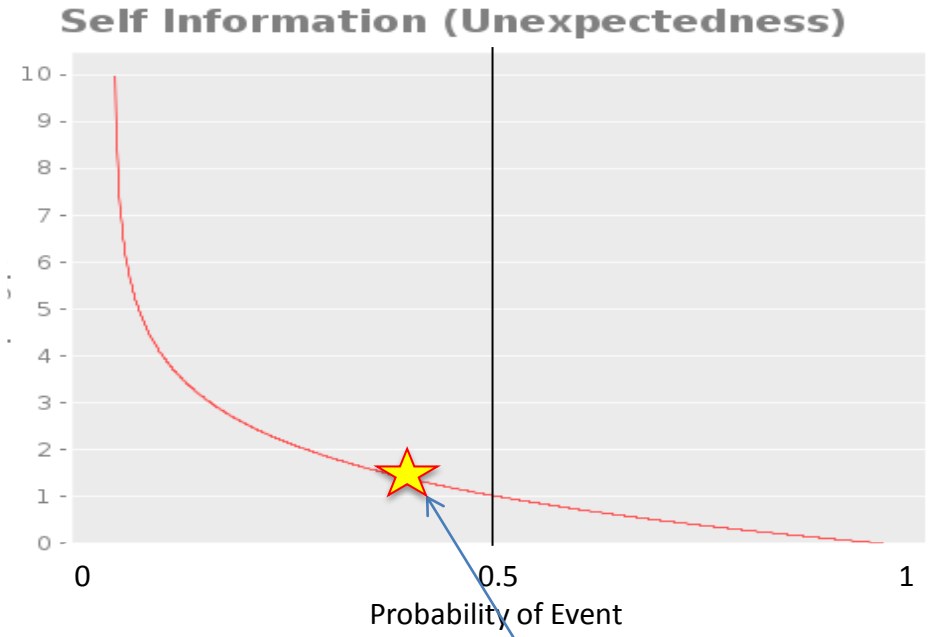

Self Information (Unexpectedness)
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
Probability of Event

$$-\log_2(p)$$

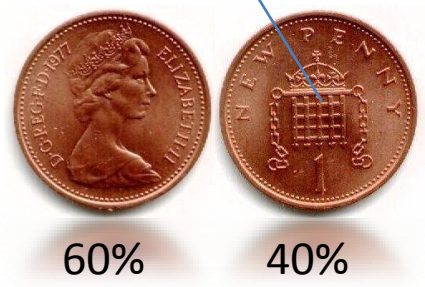

Self Information (Unexpectedness)
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
Probability of Event

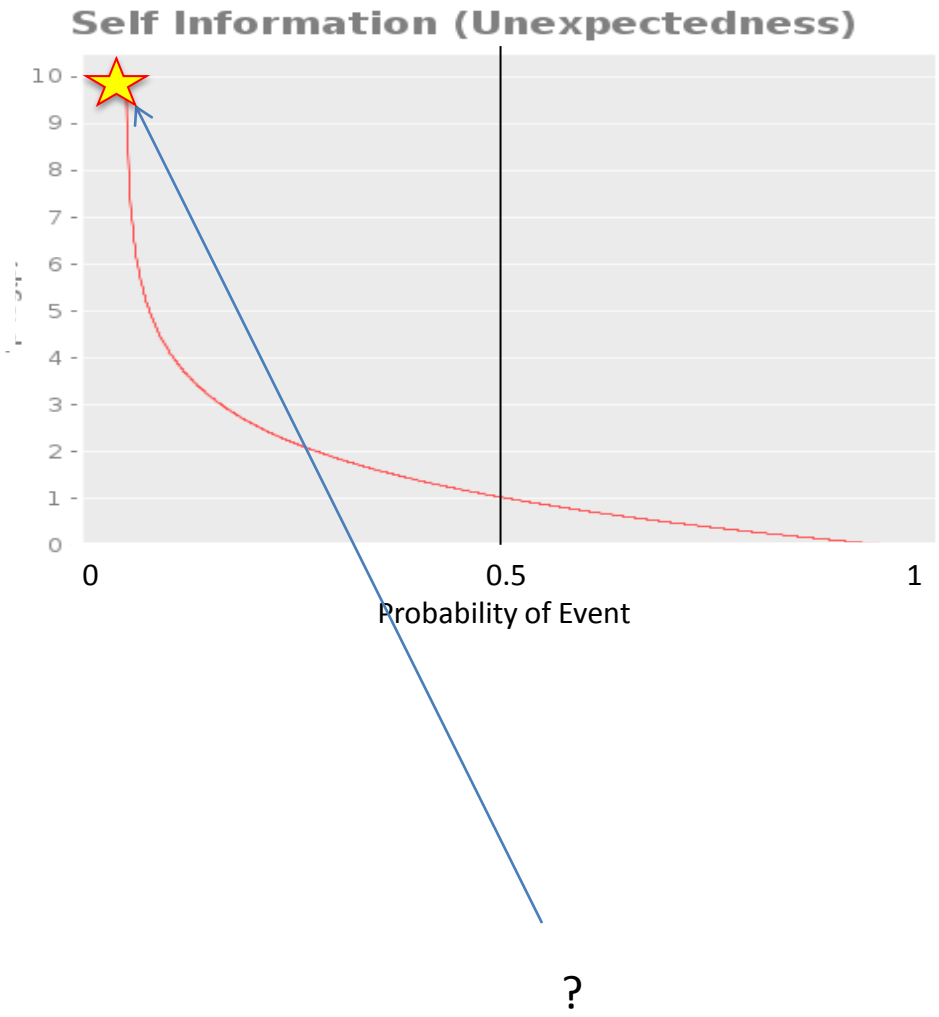
$$-\log_2(p)$$


60%
40%


Self Information (Unexpectedness)
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
Probability of Event

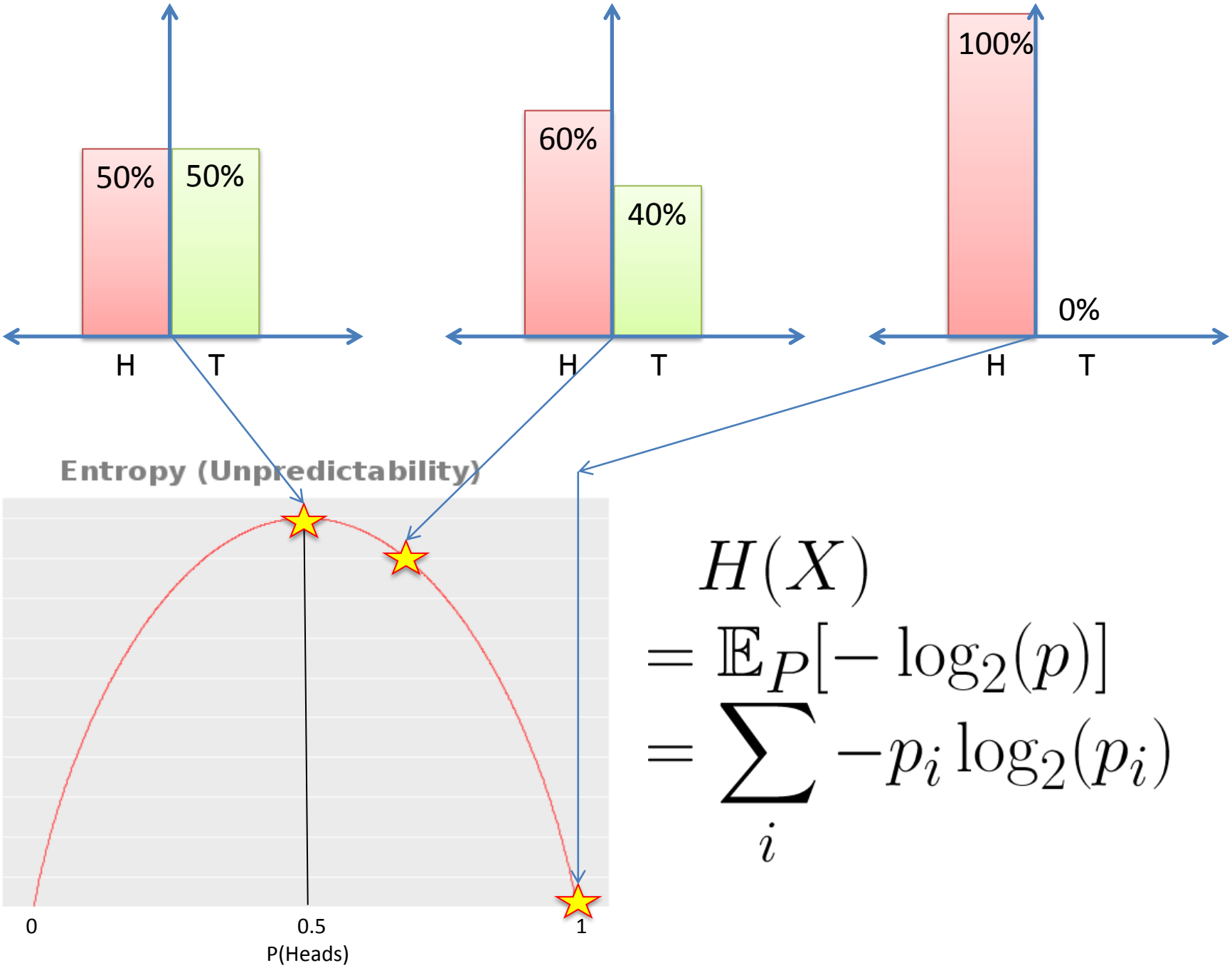
$$-\log_2(p)$$


60%
40%


Self Information (Unexpectedness)
10
9
8
7
6
5
4
3
2
1
0
0
0.5
1
Probability of Event

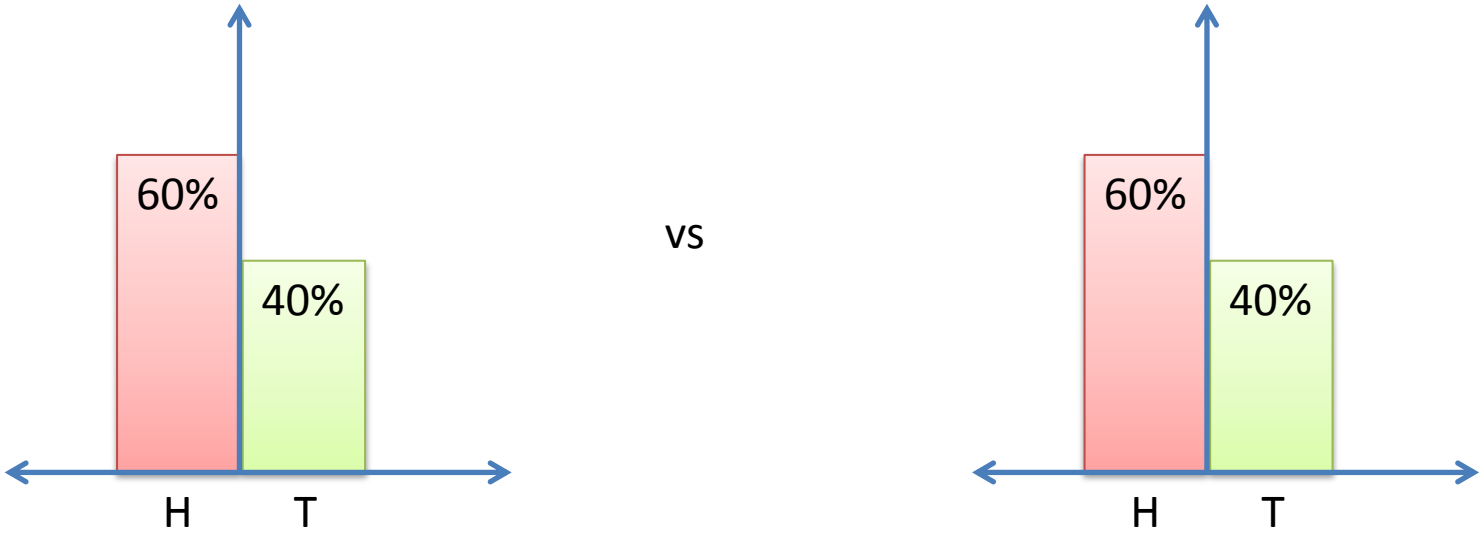
$$-\log_2(p)$$

?



$$H(X) = \mathbb{E}_P[-\log_2(p)] = \sum_i -p_i \log_2(p_i)$$

Relative Entropy



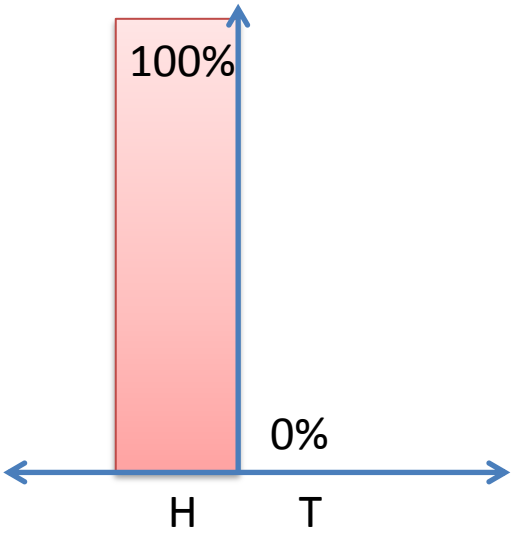
$$\sum_i p_i \log\left(\frac{p_i}{q_i}\right)$$

$$= 0.6 \log\left(\frac{0.6}{0.6}\right) + 0.4 \log\left(\frac{0.4}{0.4}\right)$$

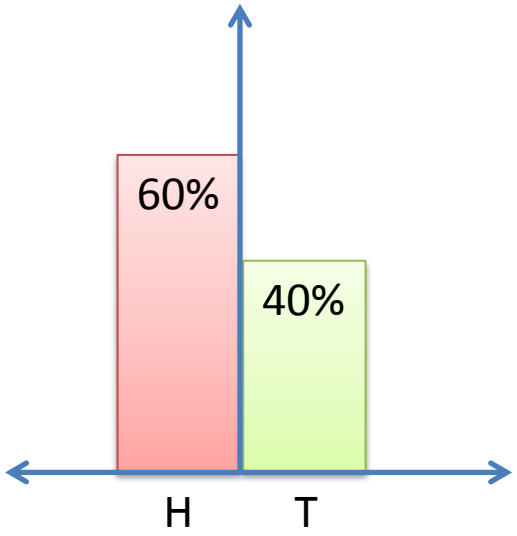
$$= 0.6 \log(1) + 0.4 \log(1)$$

$$= 0$$

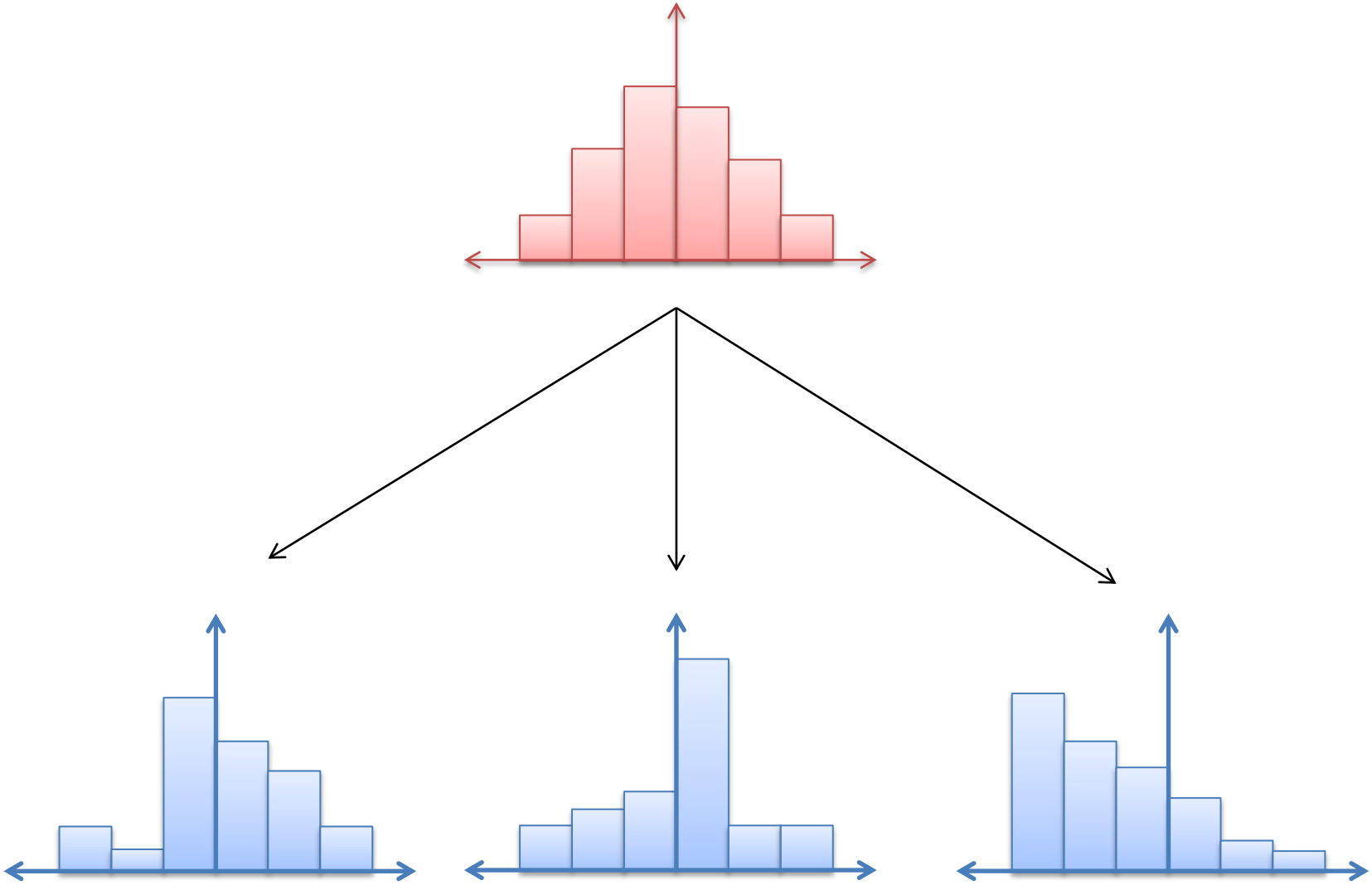
Relative Entropy



vs



$$\sum_i p_i \log\left(\frac{p_i}{q_i}\right)$$
$$= \log(1/0.6) + 0$$
$$= 0.737$$

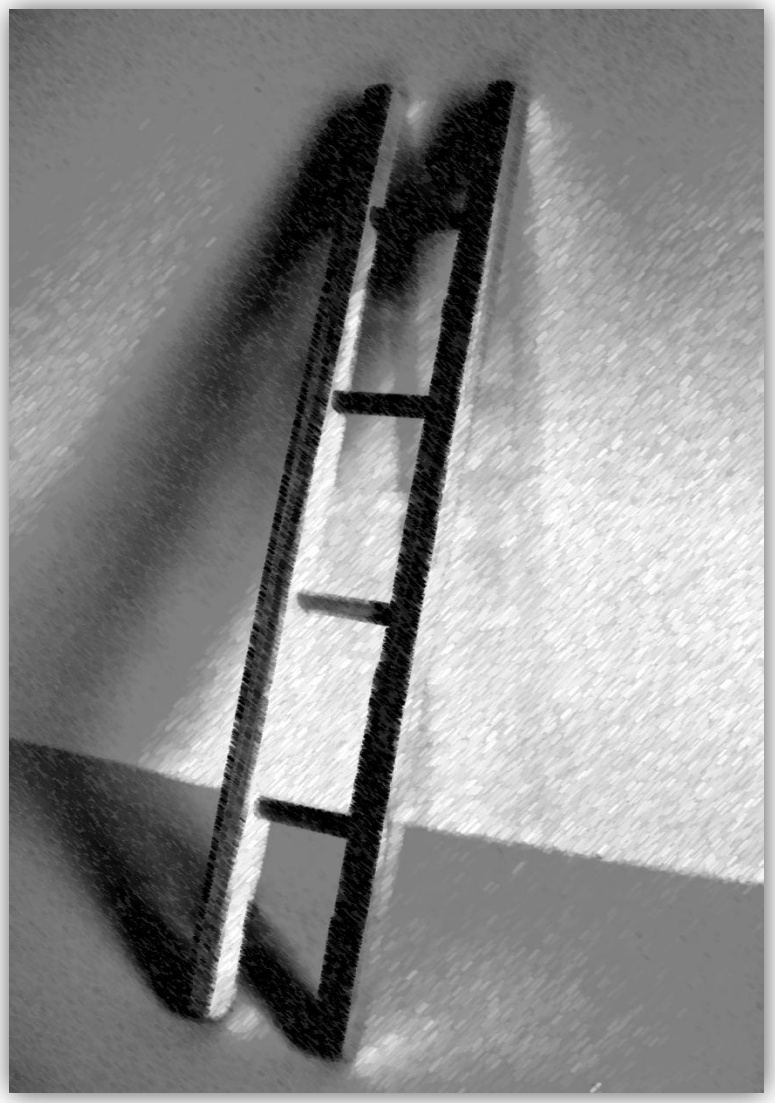


Summary

- Entropy
  - Sum(p*log(p))
  - **amount** of unpredictability

- Relative Entropy
  - Sum(p*log(p/q))
  - **distance** between two distributions

- Native Libraries
  - Linear Algebra
  - Optimisation
  - Statistics
- Numerics
  - No hope



The Future

- Near
  - Libraries & Ports !
  - **Session**
  - Incanter 2.0 ?
- Medium
  - Probabilistic Programming (Church)


ZEN ROBOTICS

@lambdanext

http://www.lambdanext.com