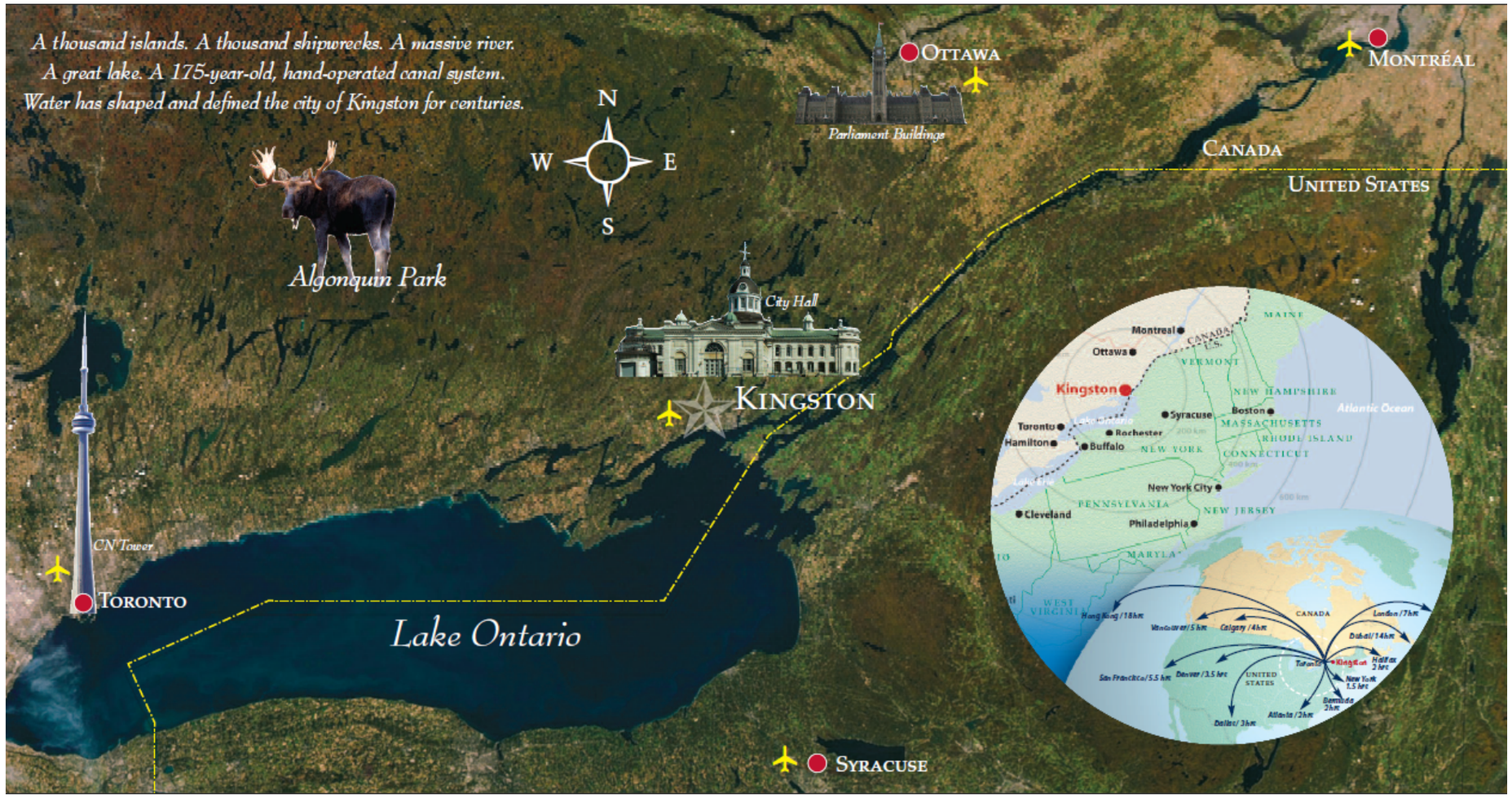
Empirical Evaluations in Software Engineering Research: A Personal Perspective

Ahmed E. Hassan

Canada Research Chair in Software Analytics
Blackberry Ultra Large Scale Systems Chair
School of Computing, Queen's University
Kingston, Canada

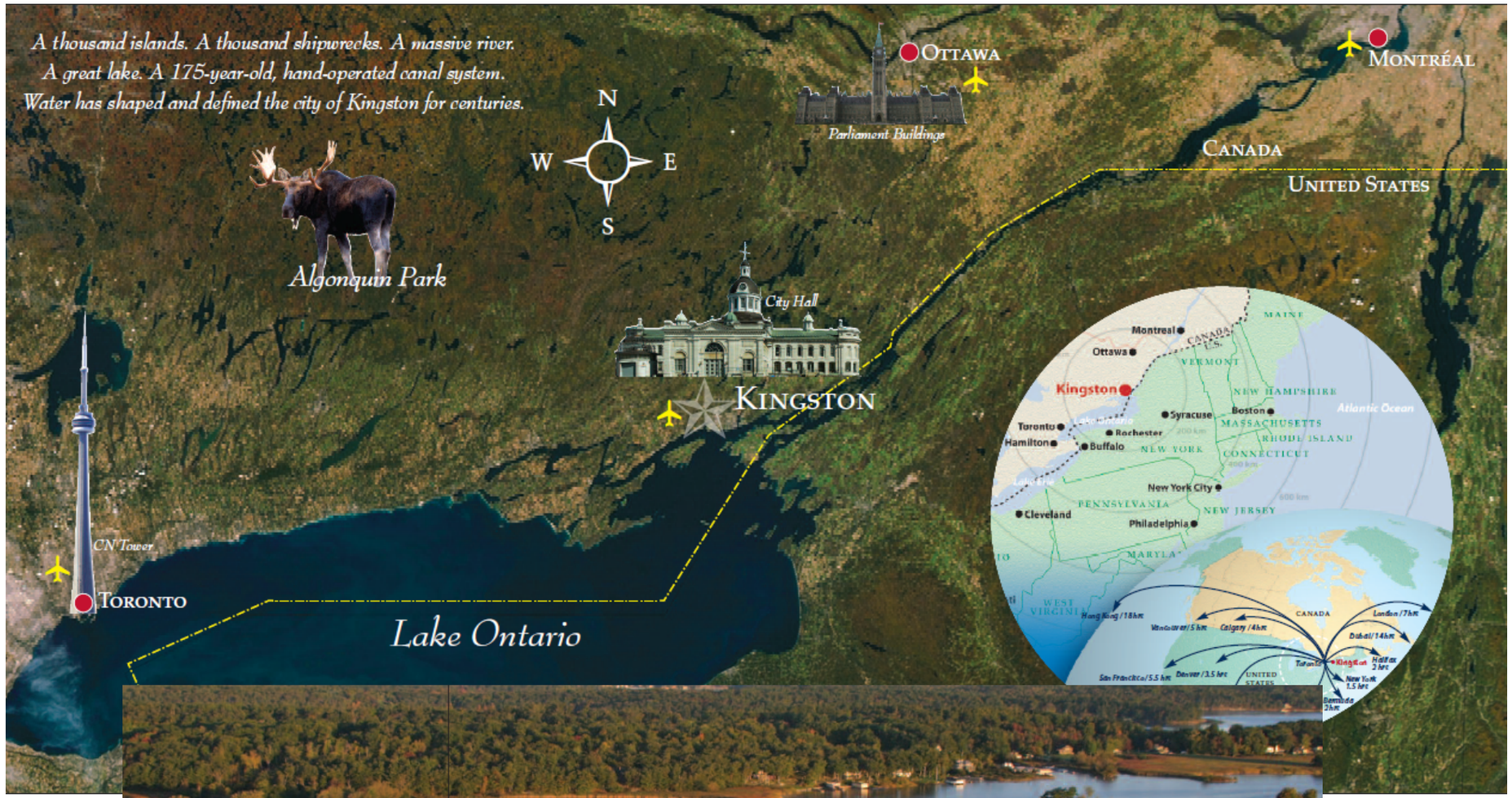
Today's goal:
Inspire change
Give concrete ideas

Kingston
Home of the 1,000 Islands



CANADA 150

Kingston
Home of the 1,000 Islands



CANADA 150

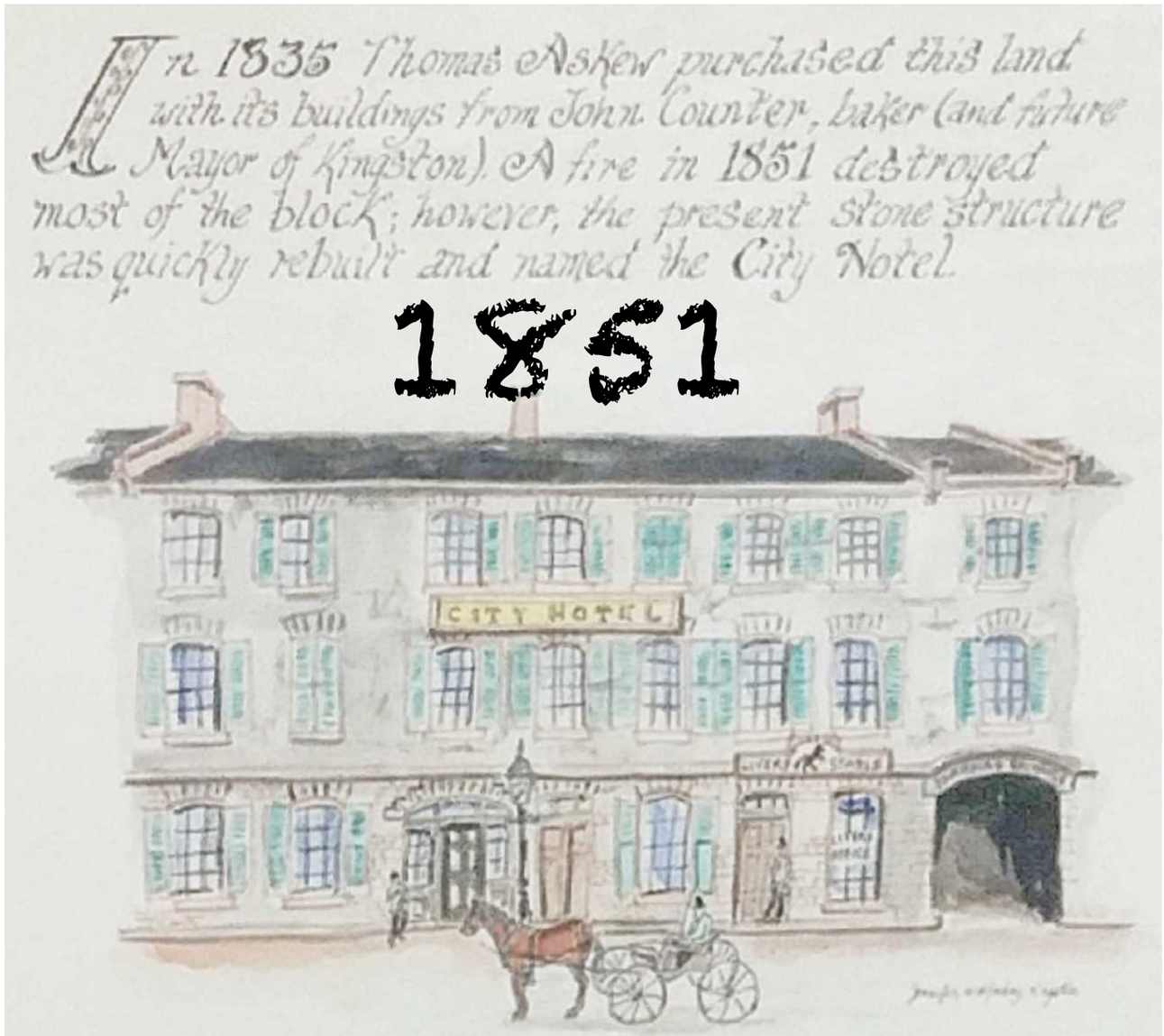
Queen's
175
YEARS

Intelligence throughout the lifetime of a software system

from inception to production

http://sail.cs.queensu.ca

SAIL's new home:
A 5,000 sq. ft. historical building



It prospered under a number of hotel keepers: Bamford, Stenson, Irwin, Johnson, Randolph... The City Hotel was renamed the Hotel Randolph in the late 19th c. and the Hotel La Salle in the 20th. In 1976 Dacon Corporation preserved the City Hotel by converting it into the La Salle Mews

SAIL's new home:
A 5,000 sq. ft. historical building

In 1835 Thomas Askew purchased this land with its buildings from John Counter, baker (and future

Stenson, Irwin, Johnson, Randolph.... The City Hotel was renamed the Hotel Randolph in the late 19th c. and the Hotel La Salle in the 20th. In 1976 Dacon Corporation preserved the City Hotel by converting it into the LaSalle Mews

SAIL's new home:
A 5,000 sq. ft. historical building

~15 years of Mining Software Repositories

http://msrconf.org

Early MSR Days

Early Day Miner

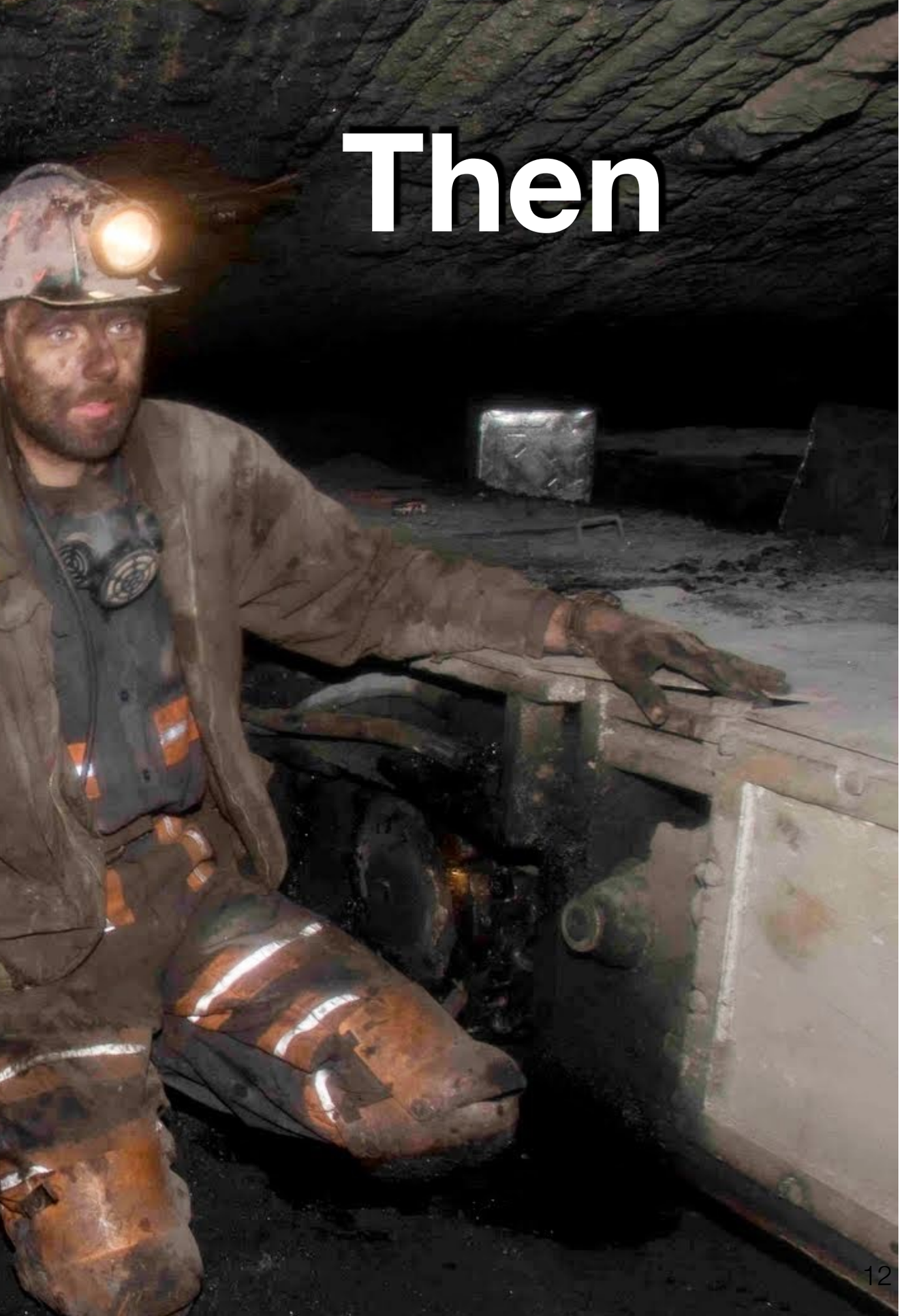
MSR Today: Easy Peasy!

Lots of data!

GHTorrent

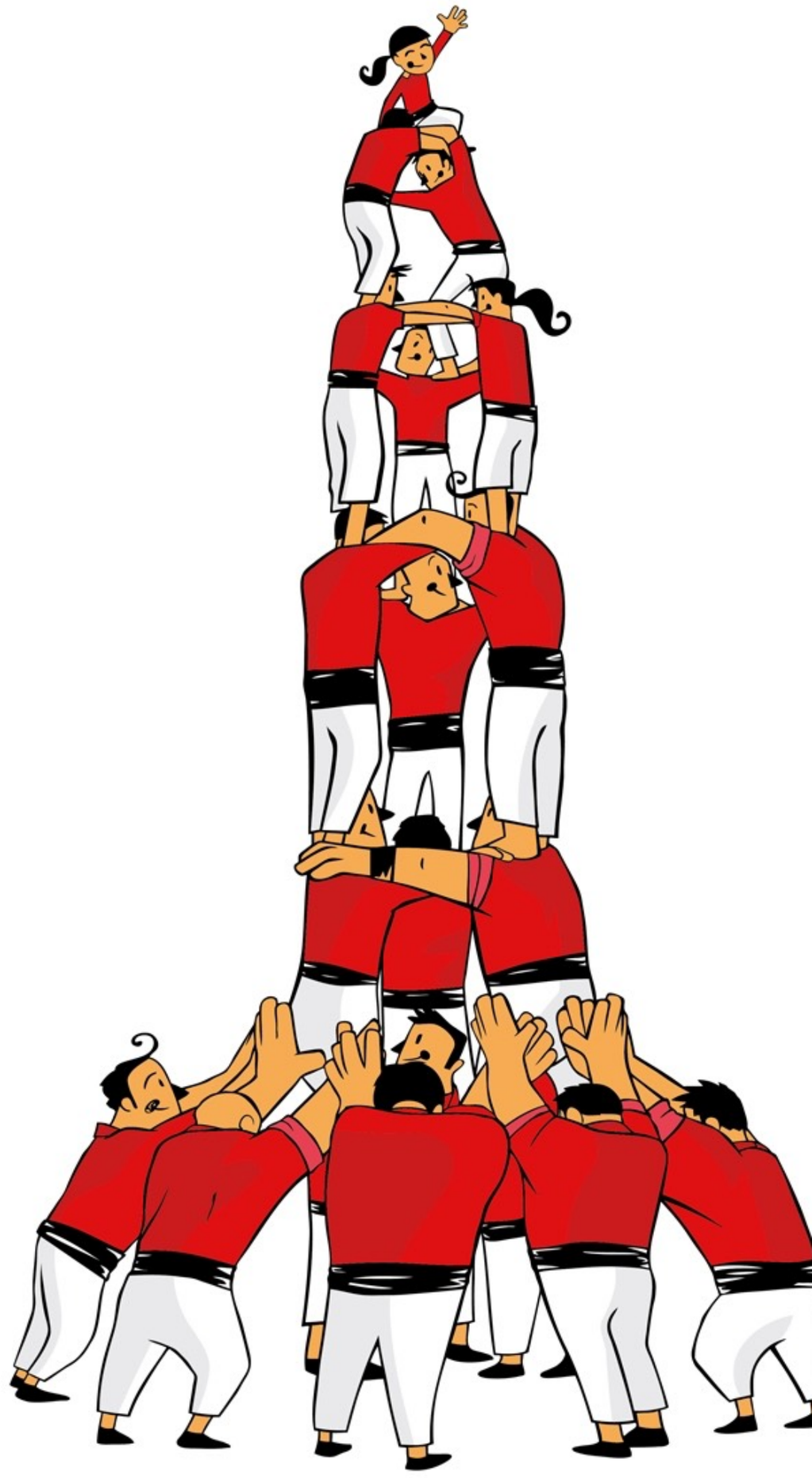
Data track

Then



Now

Thanks to a large scale community effort!



Yet low practitioner interest!!

Even though our friendly reviewers are pleased!

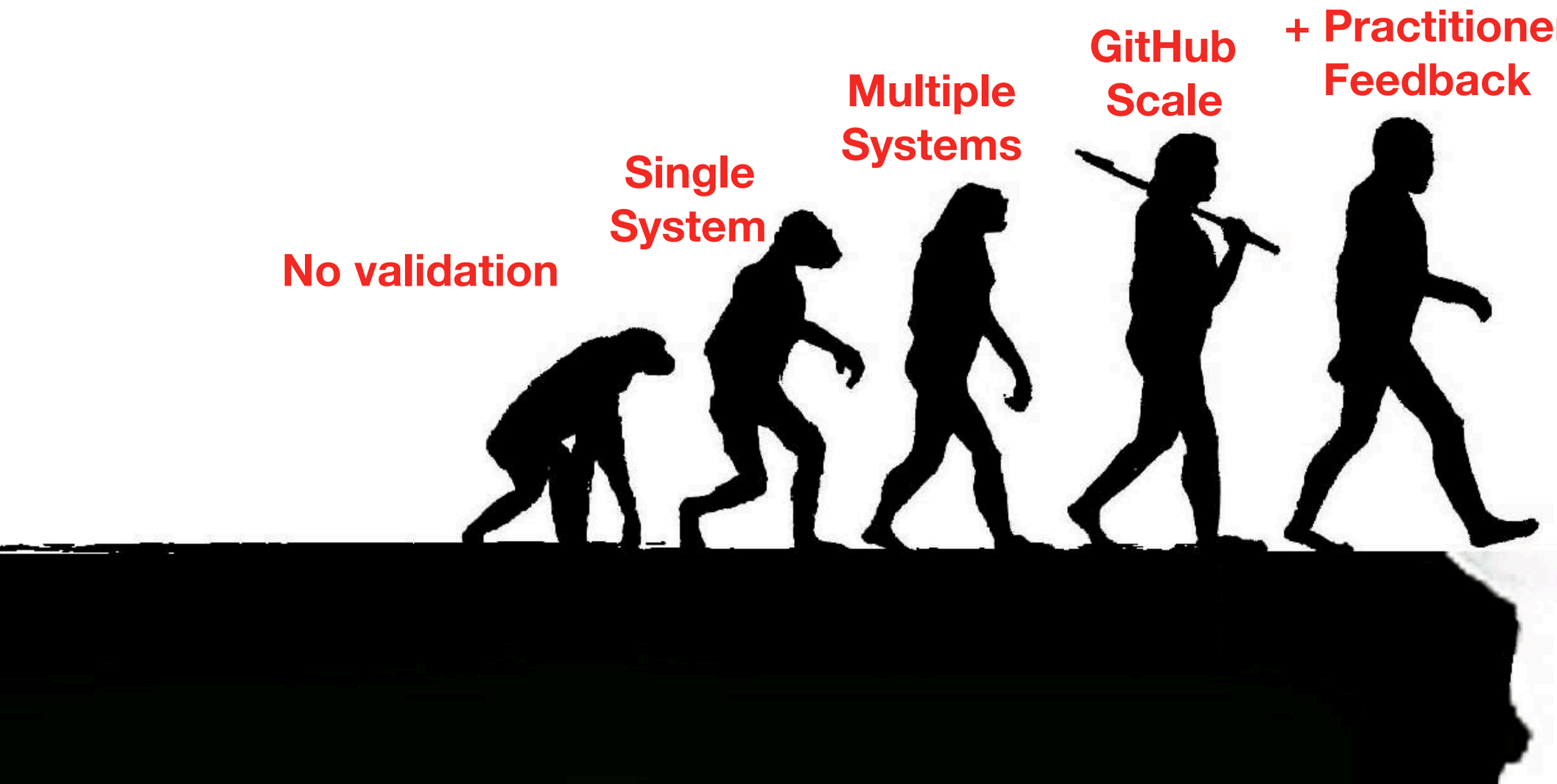
Why?!?

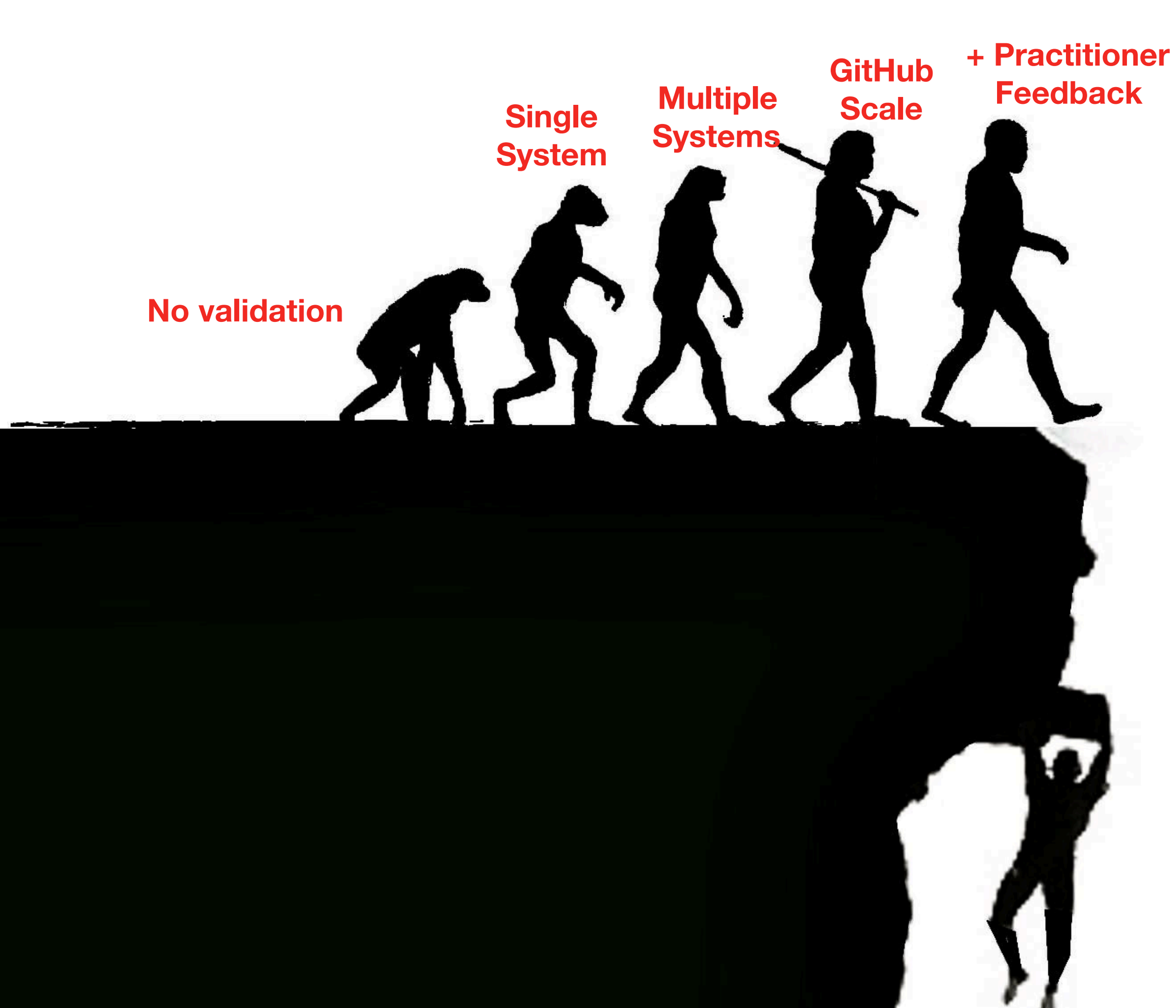
Why?!?

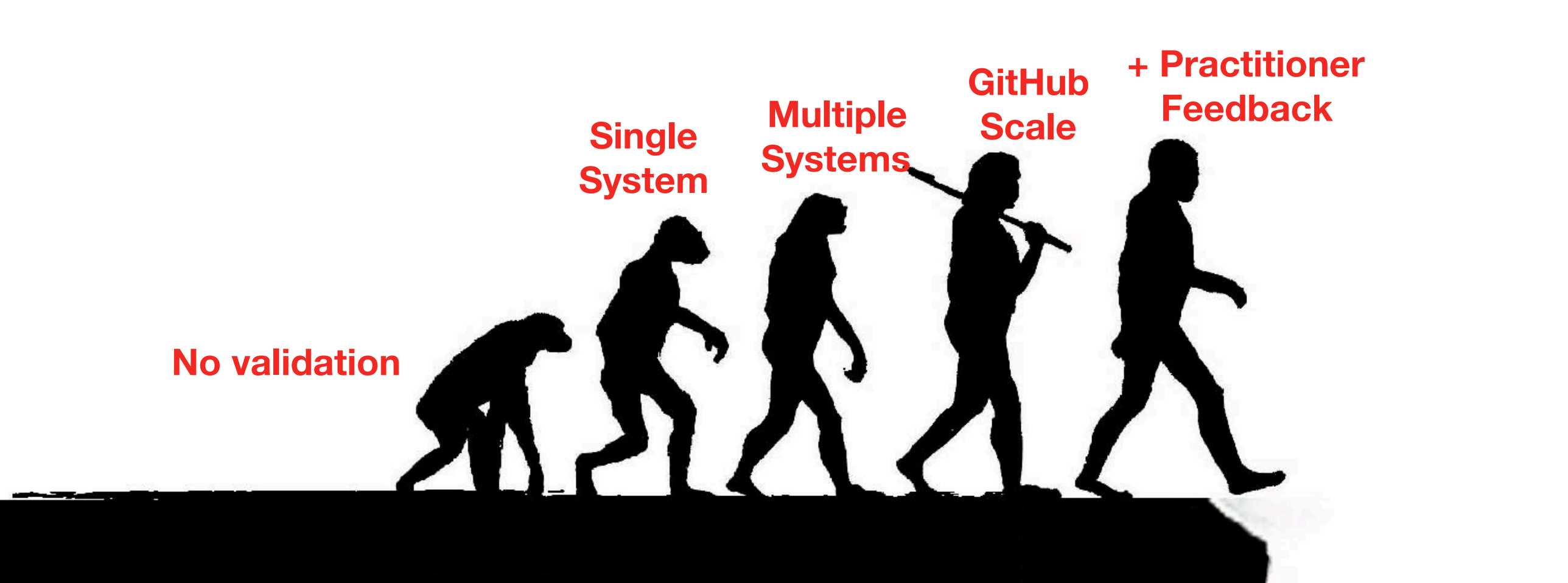
We continue to raise the bar for studies

Empirical evaluations continues to mature and evolve

Empirical evaluations continues to mature and evolve

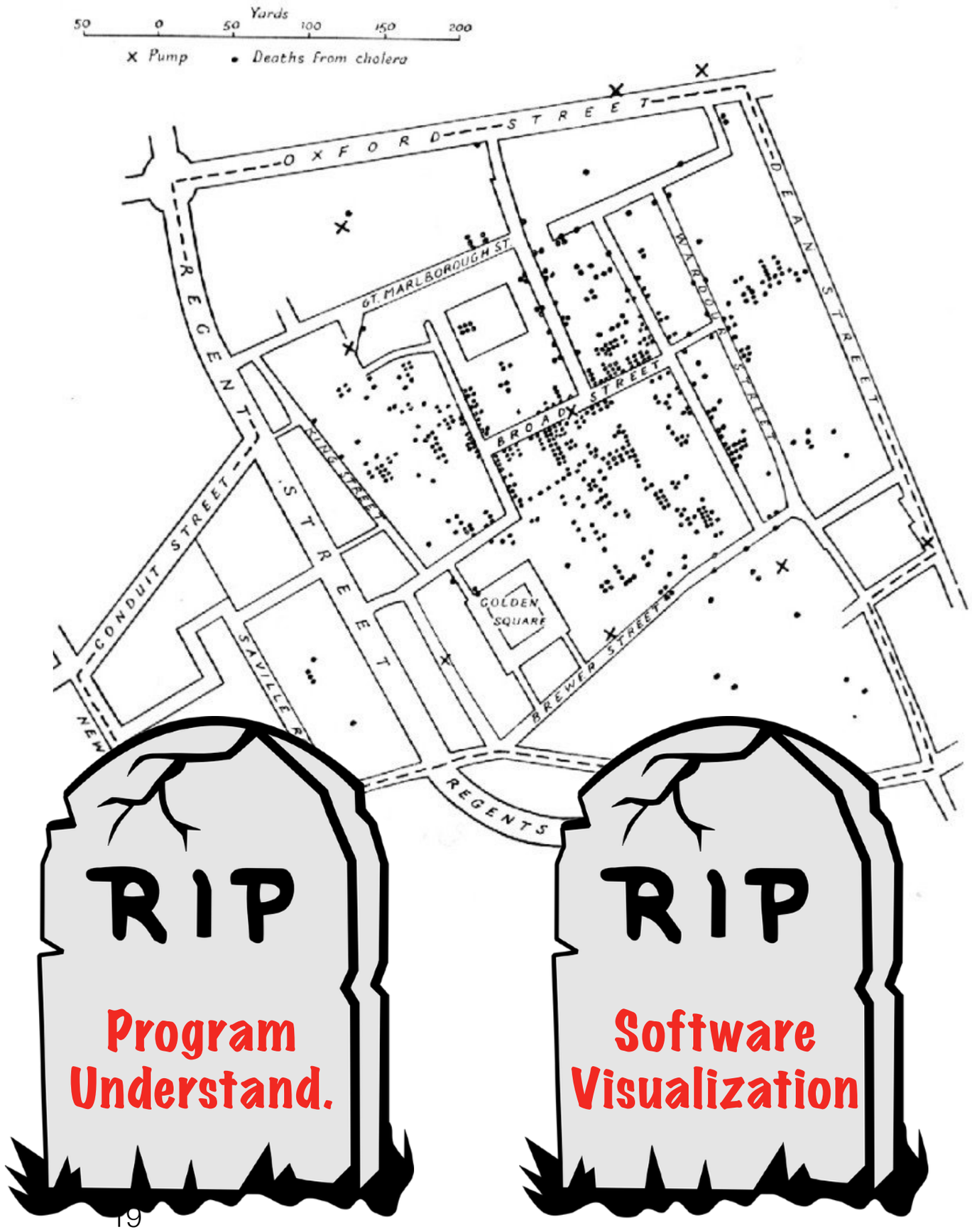



No validation
Single System
Multiple Systems
GitHub Scale
+ Practitioner Feedback

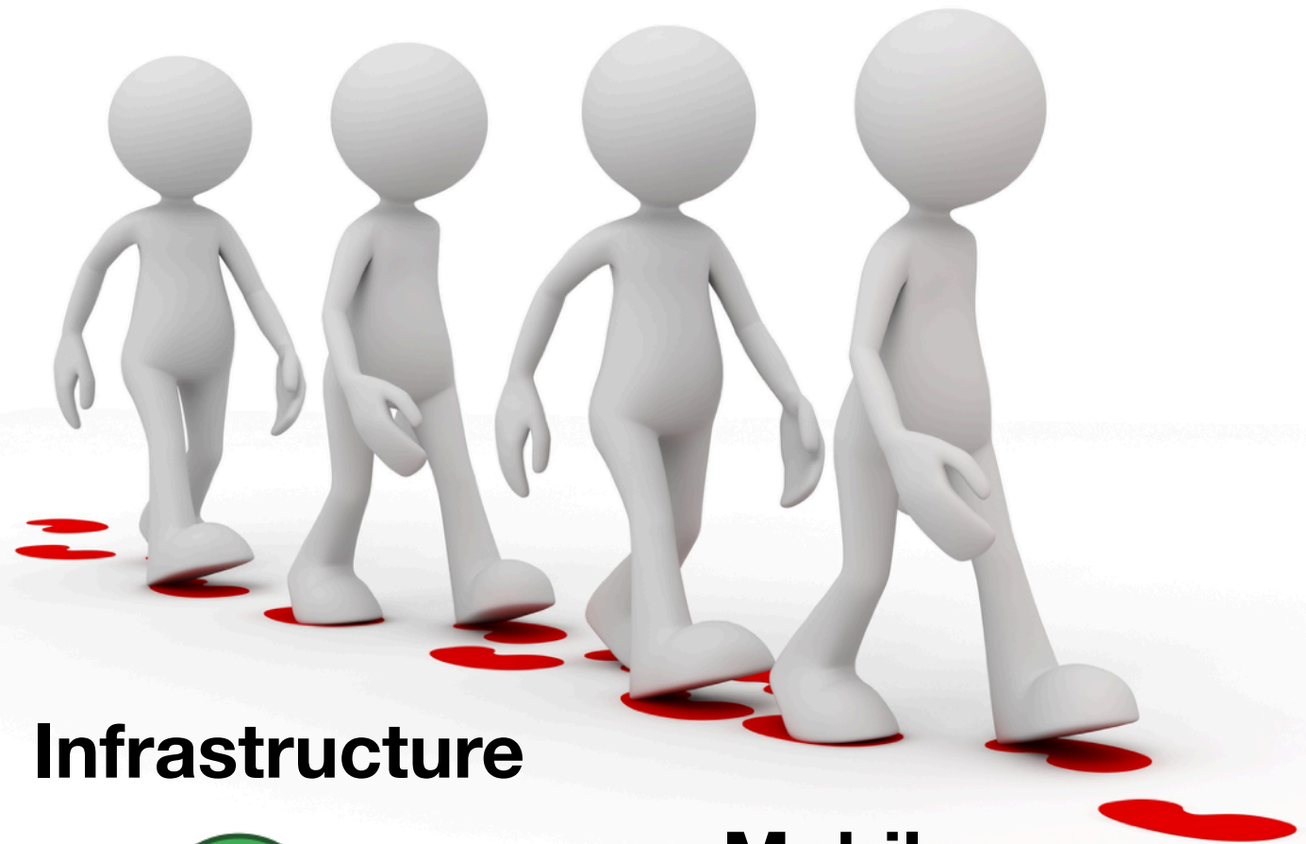


Are things this bad?
What can we do to prevent this?

We are killing research areas



We are following industry instead of leading!!



Infrastructure

open source

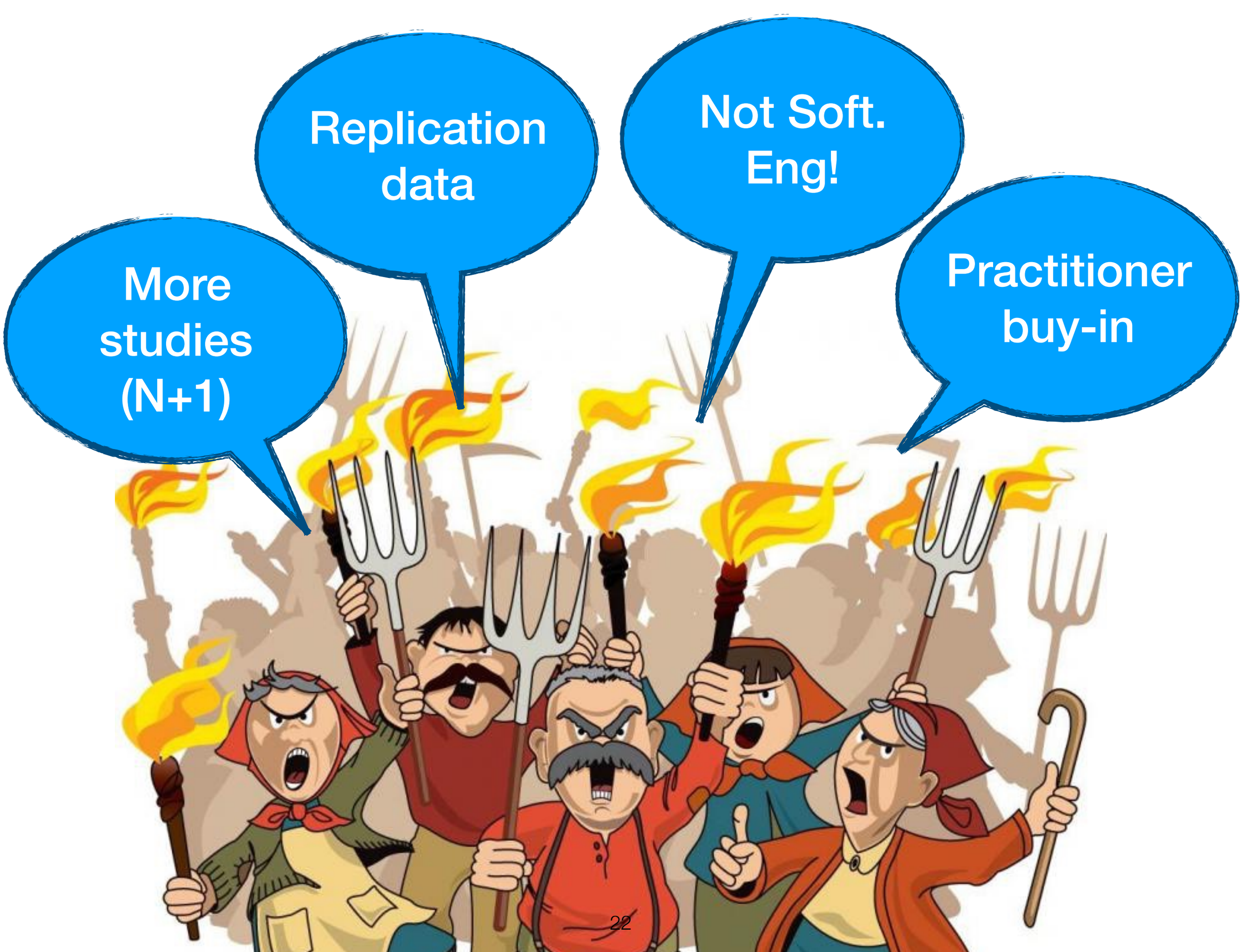
Mobile Apps

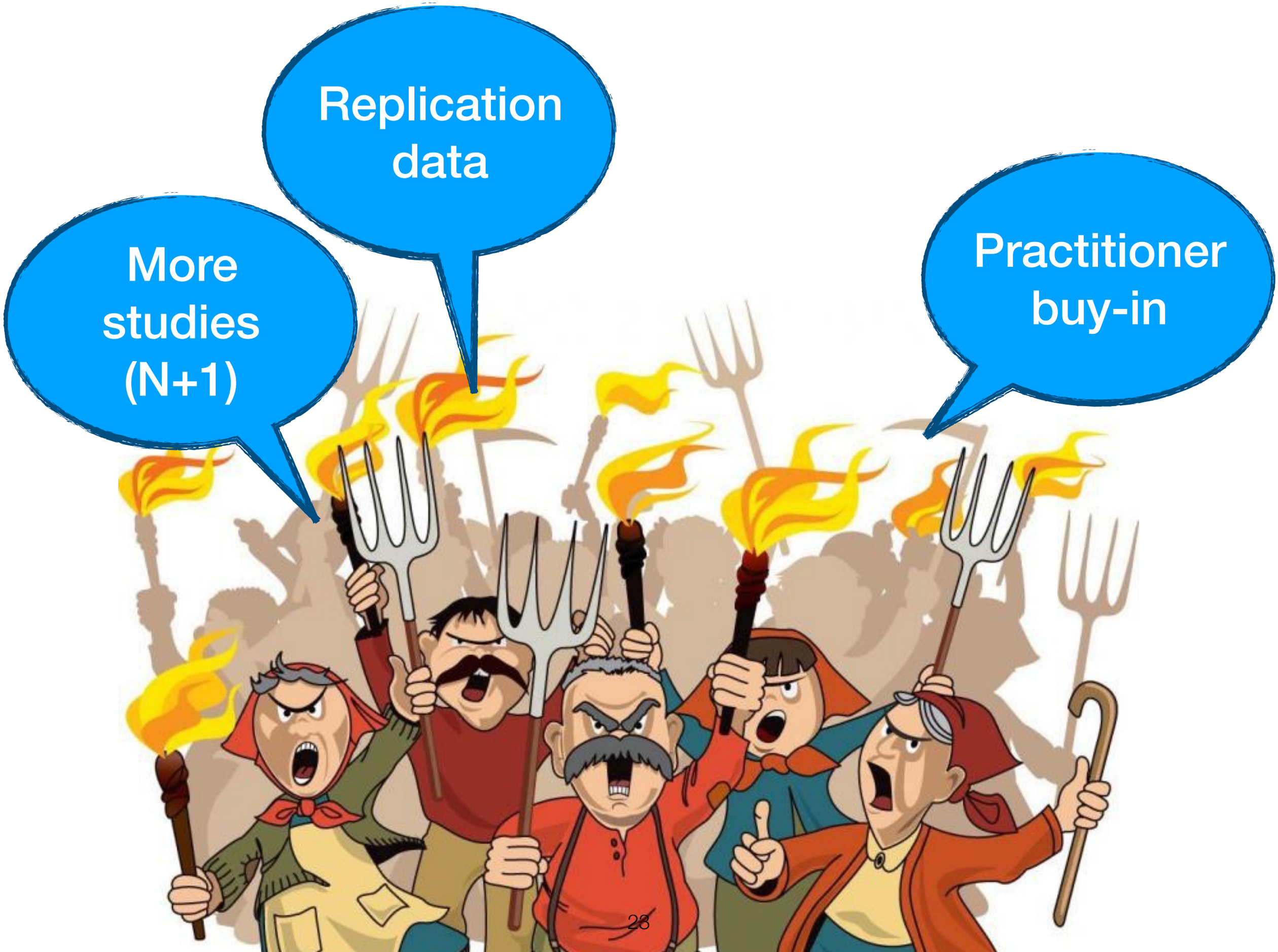
IoT

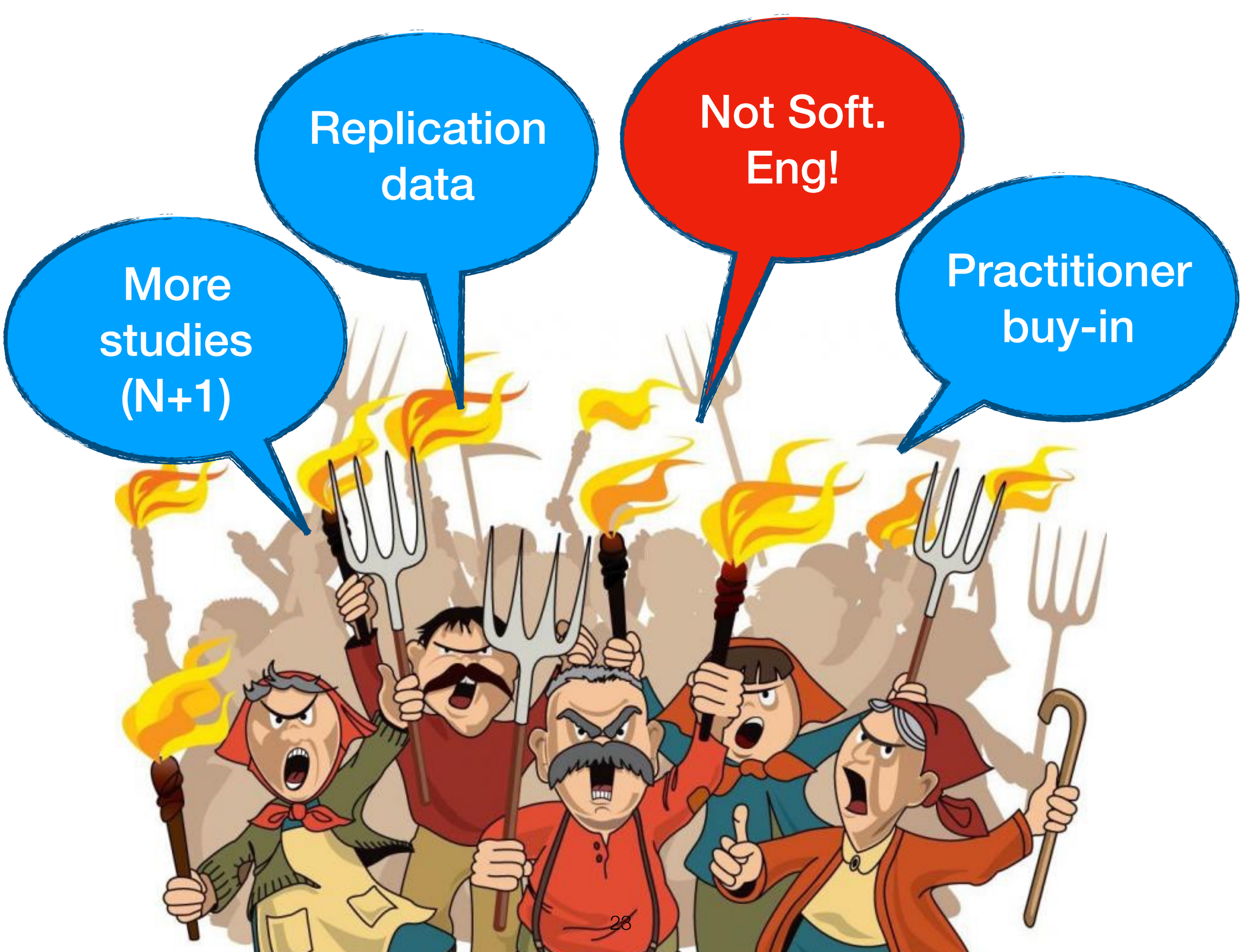
Web Apps (e.g, AJAX)

Data Analytics Scripting

We are doing this to ourselves!!


More studies (N+1)
Replication data
Not Soft. Eng!
Practitioner buy-in


Replication data
More studies (N+1)
Practitioner buy-in


Replication data
Not Soft. Eng!
More studies (N+1)
Practitioner buy-in

Not Software Enginering

Not Software Enginering

Mobile Apps

Logs

Build Systems

Load Testing

NoSQL Systems

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full text and hyperlink database of at least 24 million pages is available at http://google.stanford.edu/

To engineer a search engine is a challenging task. Search engines index tens to hundreds of millions of web pages involving a comparable number of distinct terms. They answer tens of millions of queries every day. Despite the importance of large-scale search engines on the web, very little academic research has been done on them. Furthermore, due to rapid advance in

Google!

5 Results and Performance

The most important measure of a search engine is the quality of its search results. While a complete user evaluation is beyond the scope of this paper, our own experience with Google has shown it to produce better results than the major commercial search engines for most searches. As an example which illustrates the use of PageRank, anchor text, and proximity, Figure 4 shows Google's results for a search on "bill clinton". These results demonstrates some of Google's features. The results are clustered by server. This helps

Query: bill clinton

http://www.whitehouse.gov/
100.00% (no date) (0K)
http://www.whitehouse.gov/
Office of the President
99.67% (Dec 23 1996) (2K)
http://www.whitehouse.gov/WH/EOP/OP/html/OP_Home.html
Welcome To The White House
99.98% (Nov 09 1997) (5K)
http://www.whitehouse.gov/WH/Welcome.html
Send Electronic Mail to the President
99.86% (Jul 14 1997) (5K)
http://www.whitehouse.gov/WH/Mail/html/Mail_President.html

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

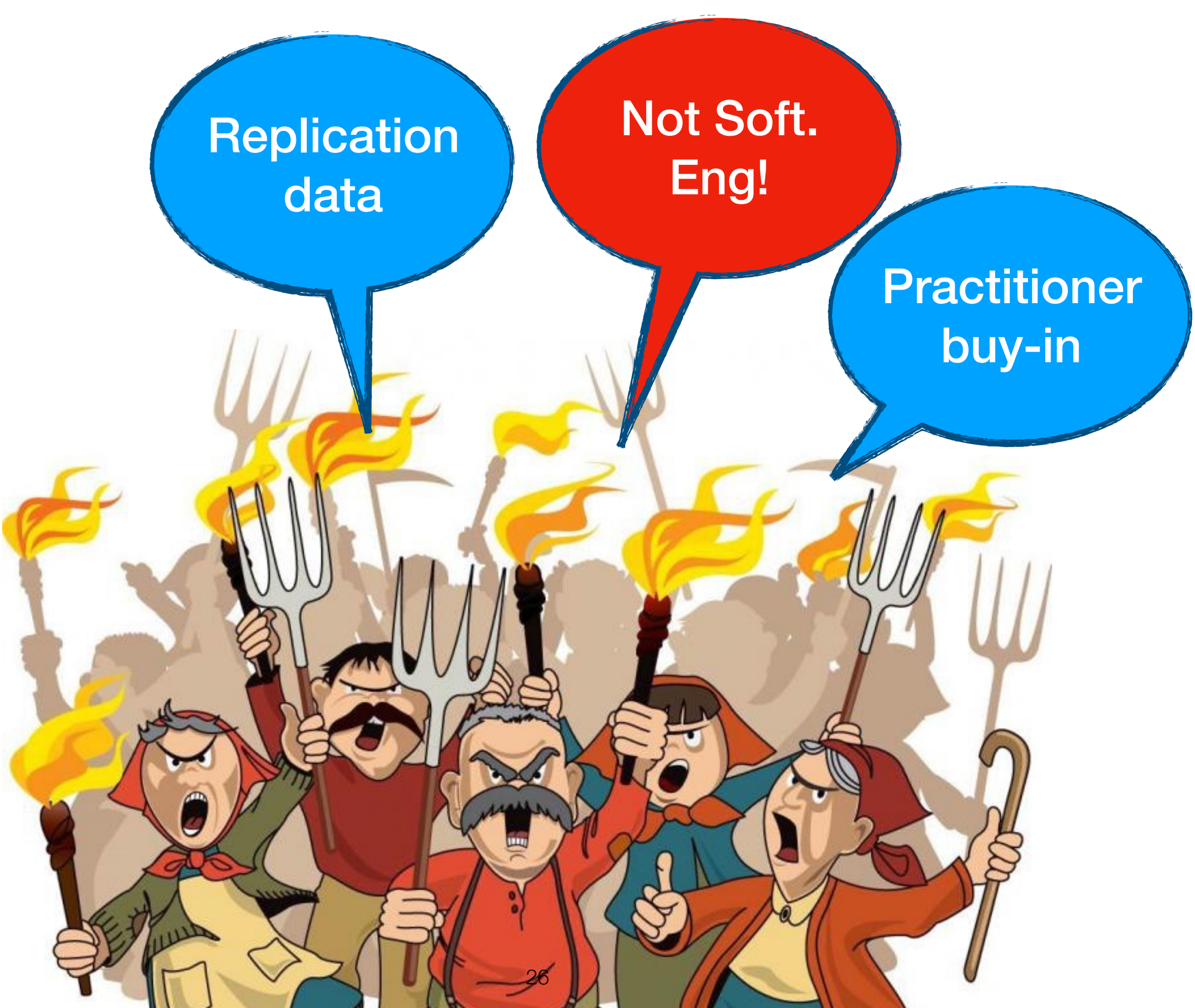
In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more
text and hyperlink data
To engineer a search e
millions of web pages
millions of queries eve
very little academic re

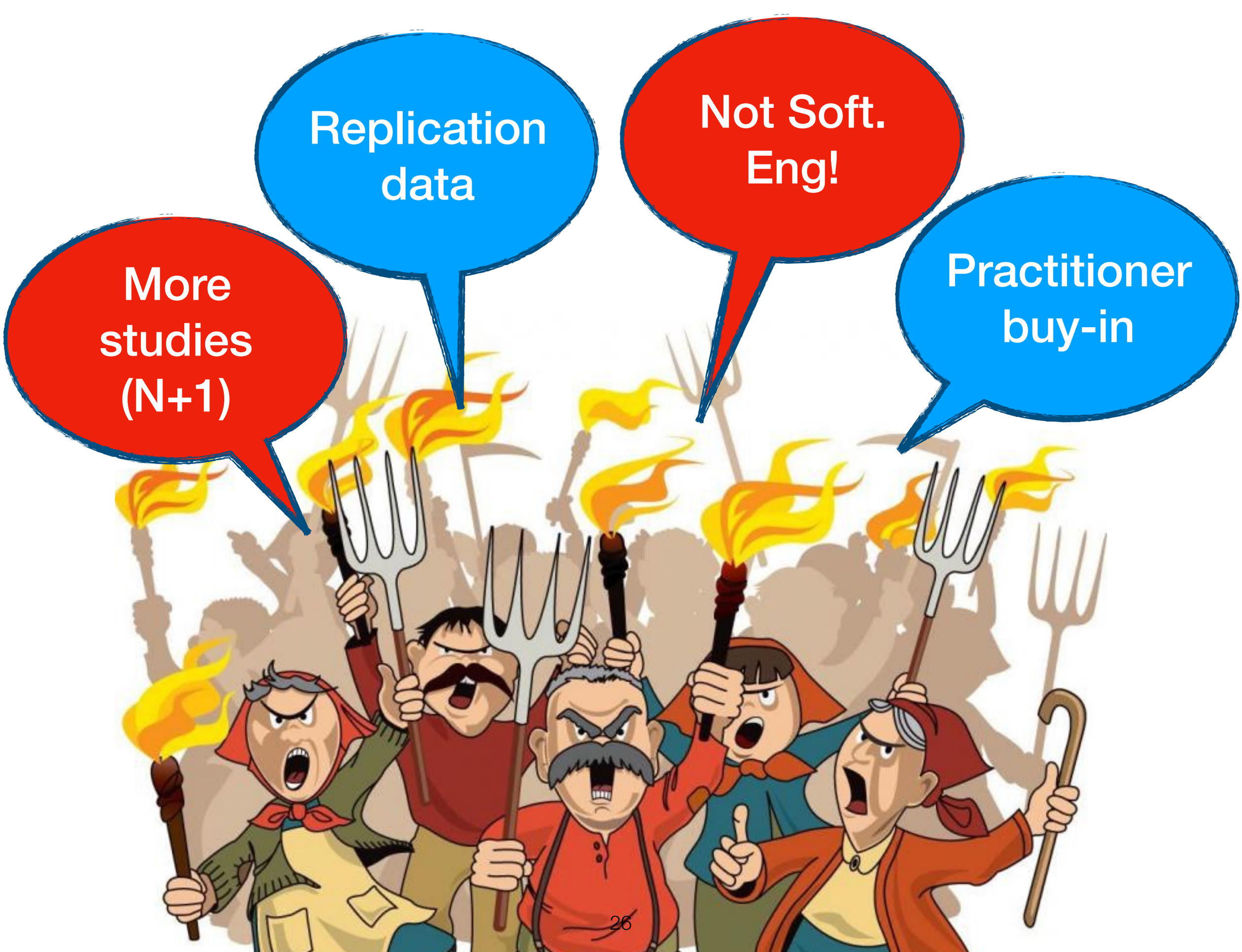
"I found the overall presentation disjointed…. This needs to focus more on the IR issues and less on web analysis." SIGIR Rejection 98

Google!

engine is the quality of its search results.
While a complete user evaluation is beyond the scope of this paper, our own experience with Google has shown it to produce better results than the major commercial search engines for most searches. As an example which illustrates the use of PageRank, anchor text, and proximity, Figure 4 shows Google's results for a search on "bill clinton". These results demonstrates some of Google's features. The results are clustered by server. This helps

http://www.whitehouse.gov/
100.00% (no date) (0K)
http://www.whitehouse.gov/
Office of the President
99.67% (Dec 23 1996) (2K)
http://www.whitehouse.gov/WH/EOP/OP/html/OP_Home.html
Welcome To The White House
99.98% (Nov 09 1997) (5K)
http://www.whitehouse.gov/WH/Welcome.html
Send Electronic Mail to the President
99.86% (Jul 14 1997) (5K)
http://www.whitehouse.gov/WH/Mail/html/Mail_President.html


Replication data
Not Soft. Eng!
Practitioner buy-in


Replication data
Not Soft. Eng!
More studies (N+1)
Practitioner buy-in

We can early-detect brain cancer!!!

How about skin/lung cancer?!!!

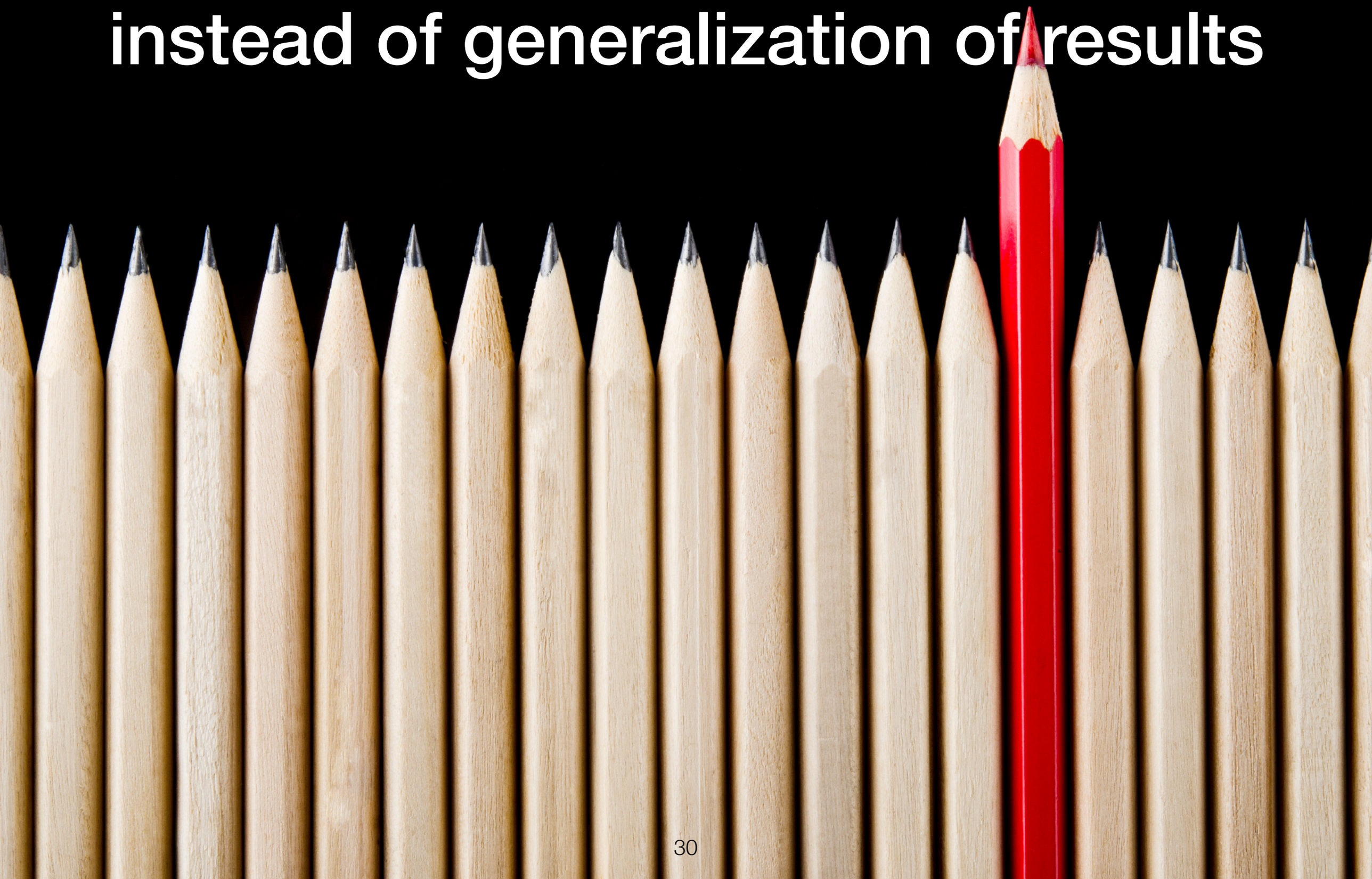
The “N+1” Critique

encourages

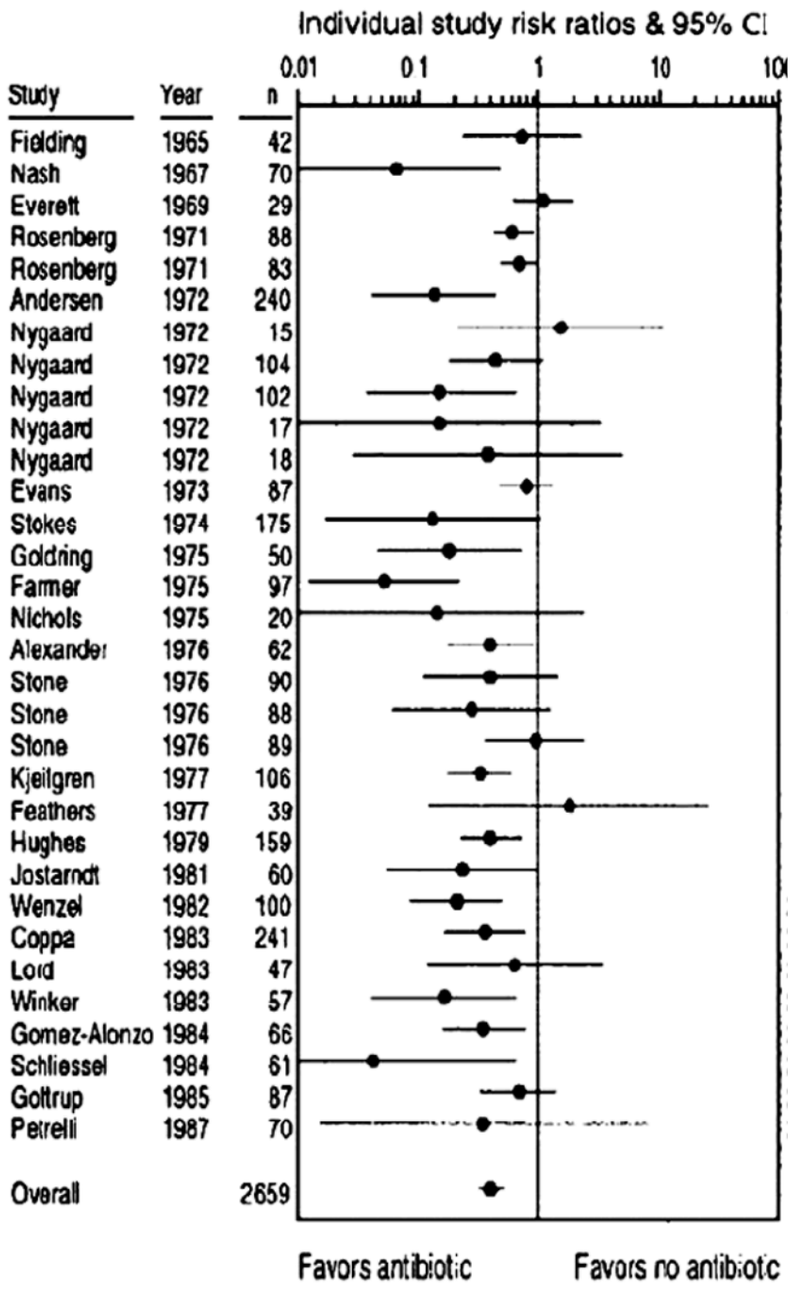
Case Study Fishing

Shallow Analysis

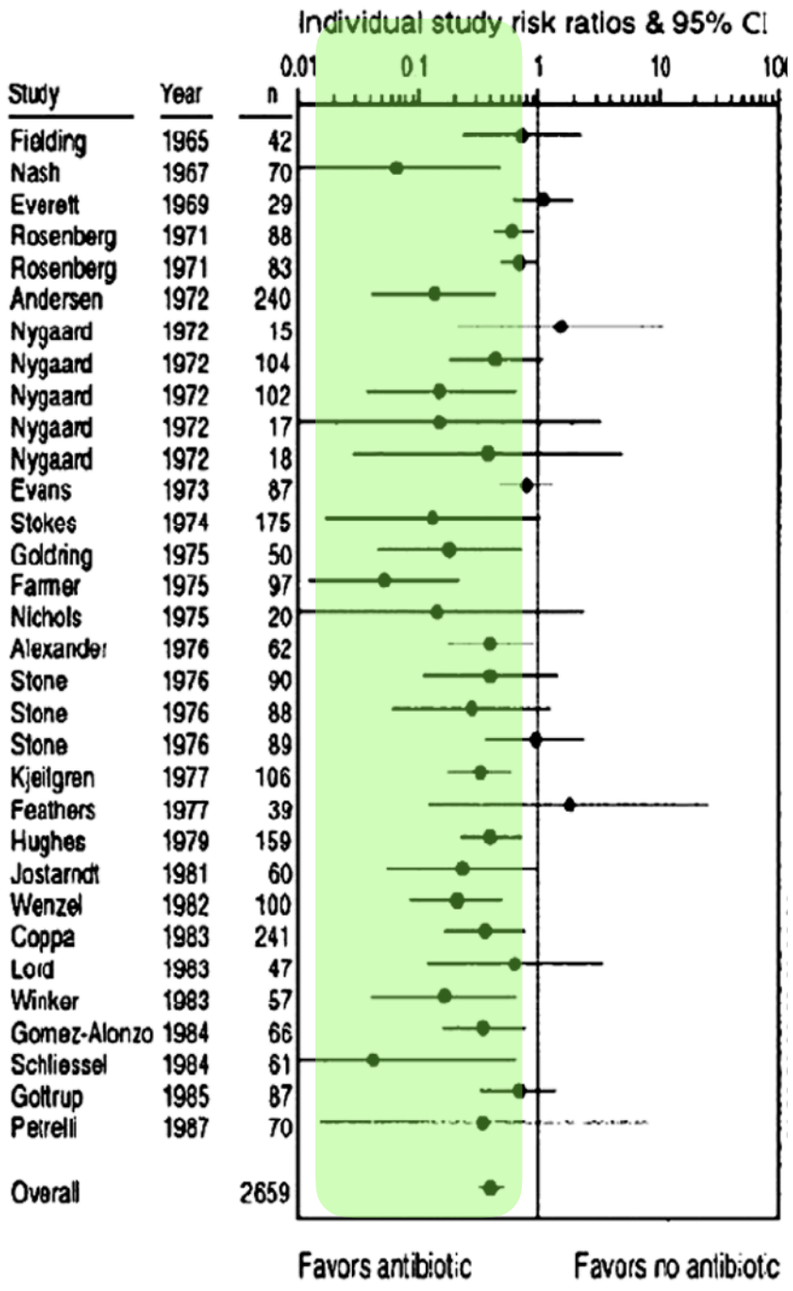
Generalization of approaches instead of generalization of results



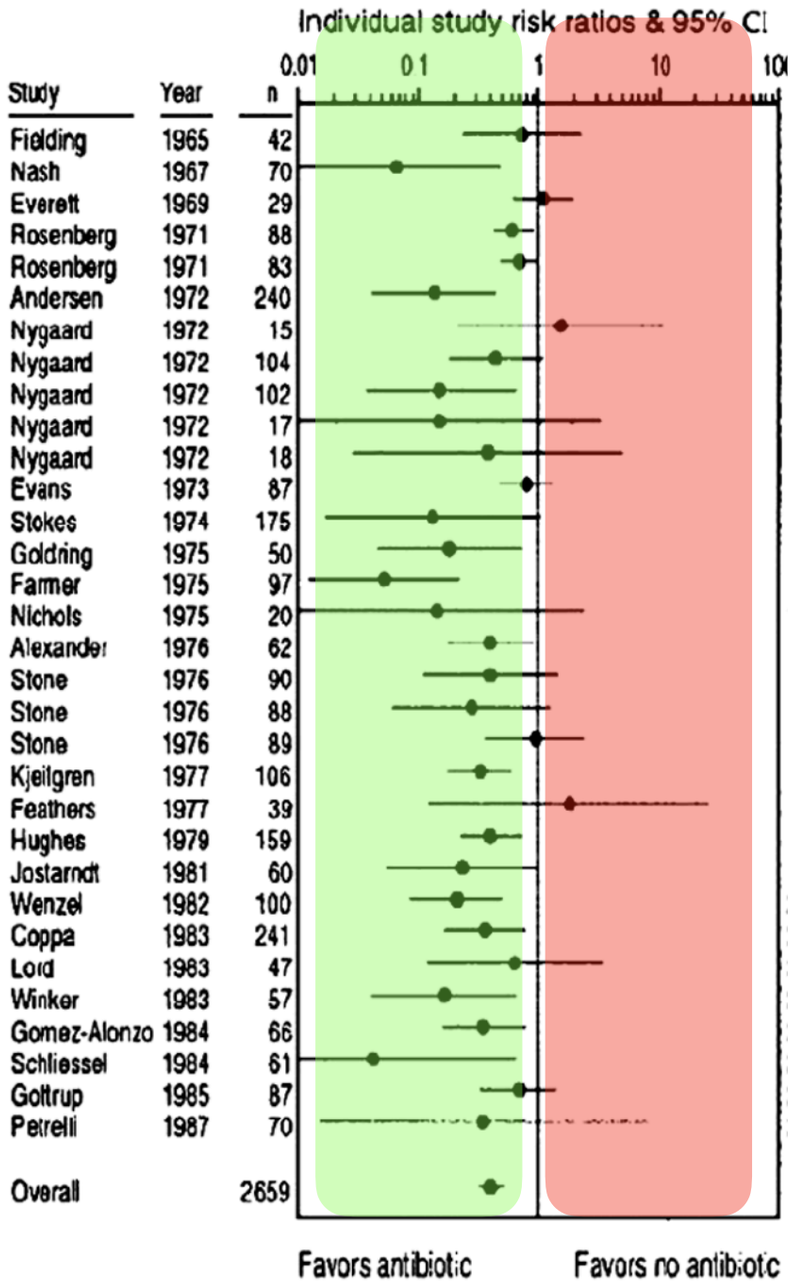
Encourage Meta Analysis



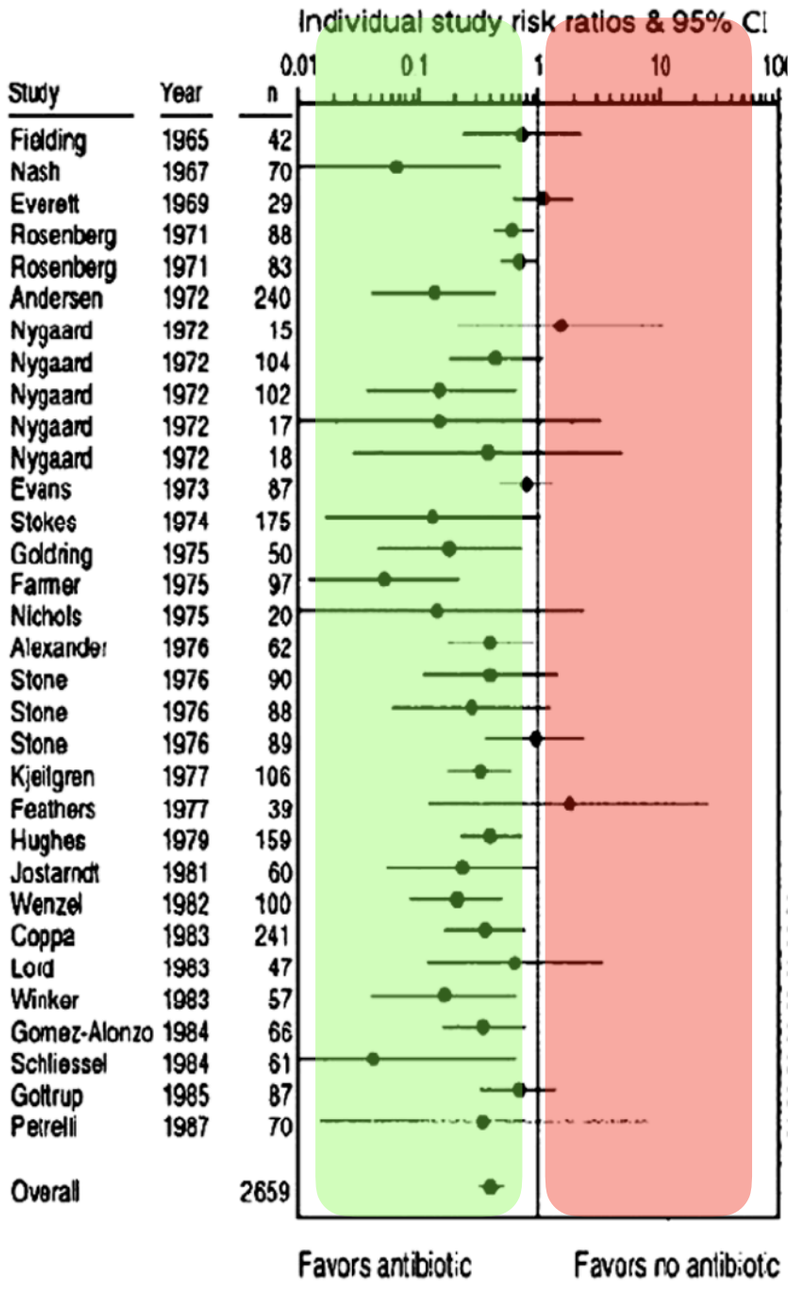
Encourage Meta Analysis



Encourage Meta Analysis



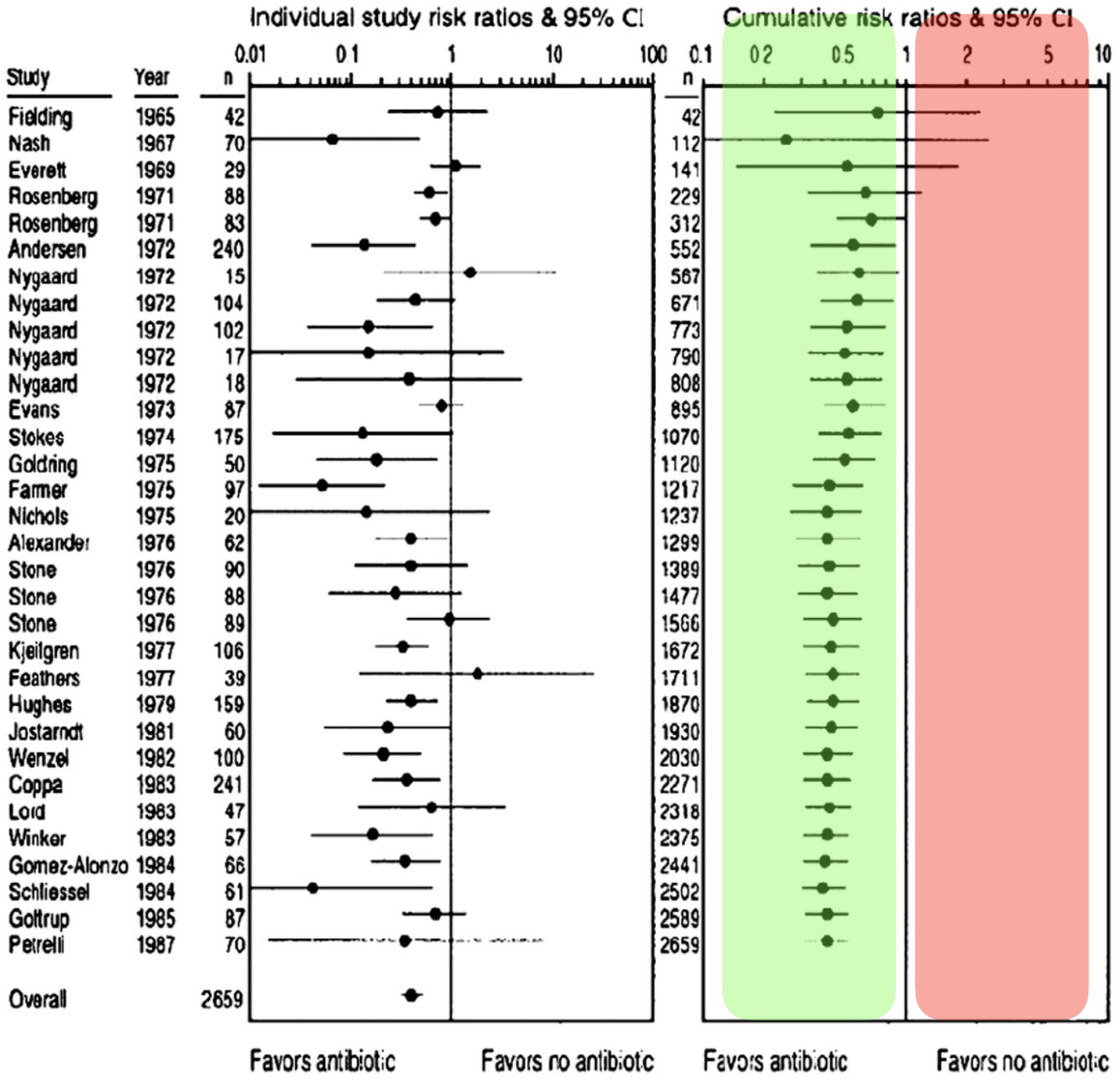
Encourage Meta Analysis



N=15,241

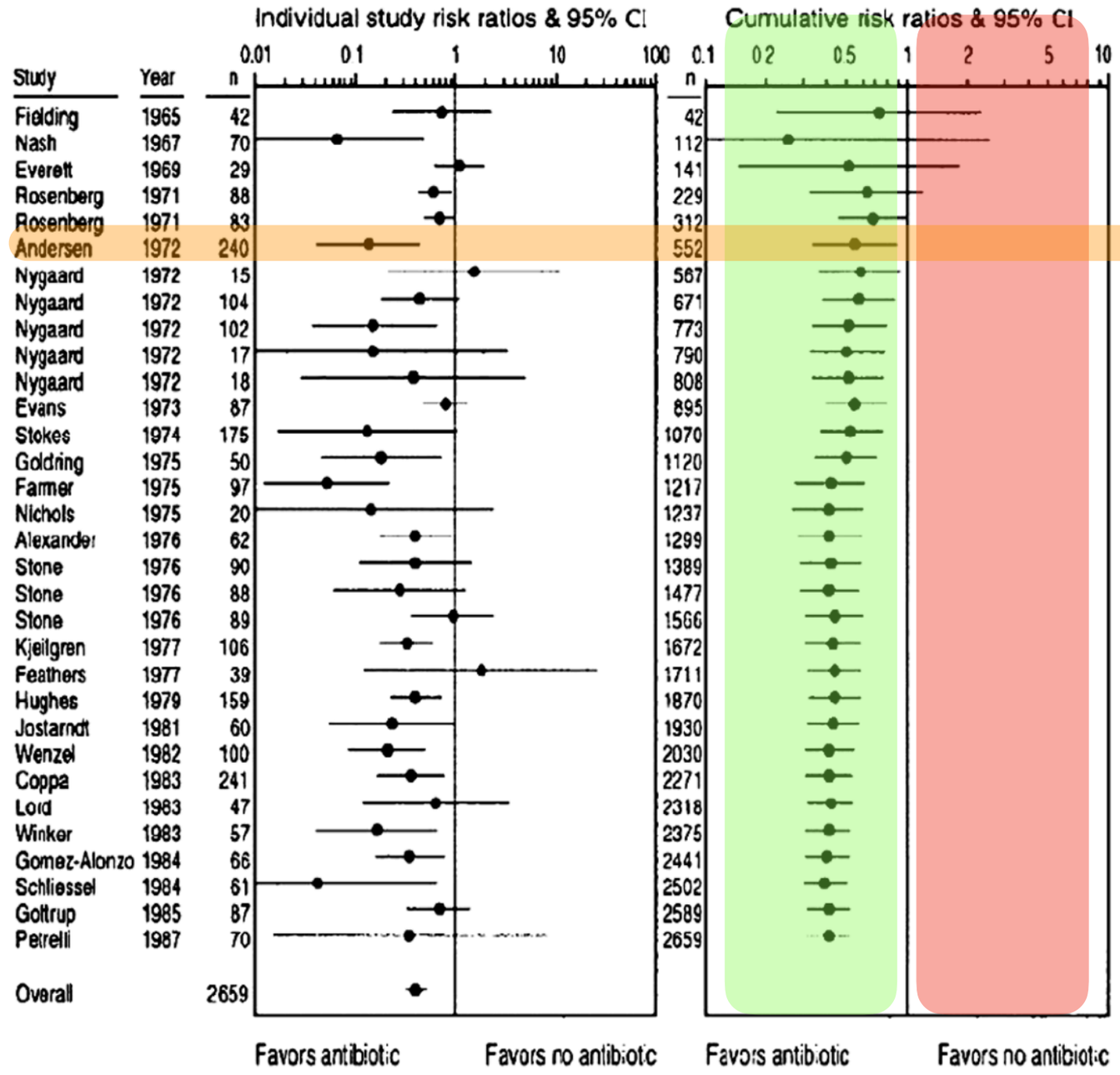
Total N=
2,659

Encourage Meta Analysis

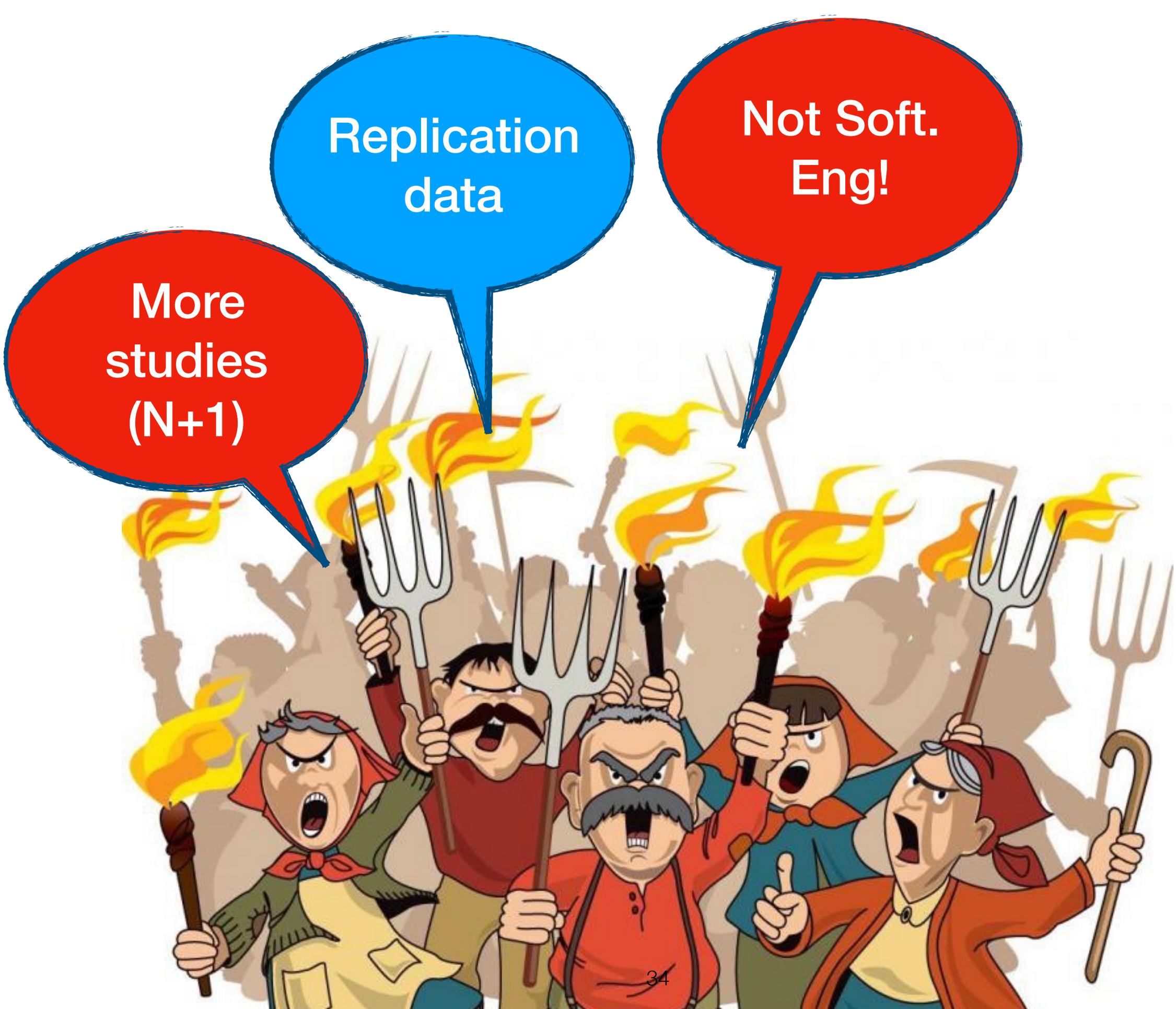


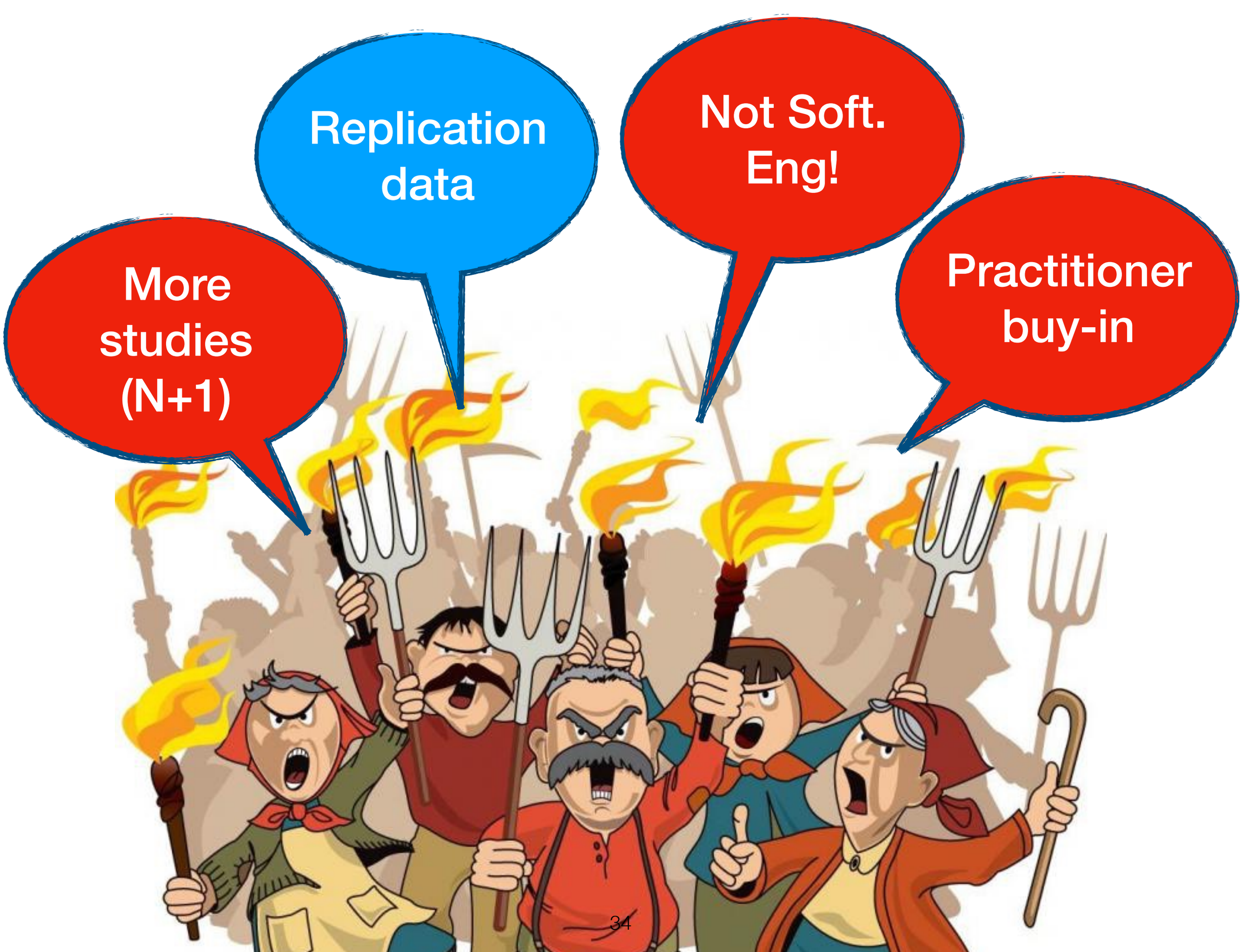
Encourage Meta Analysis

1972



More is less!


Replication data
Not Soft. Eng!
More studies (N+1)


Replication data
Not Soft. Eng!
More studies (N+1)
Practitioner buy-in

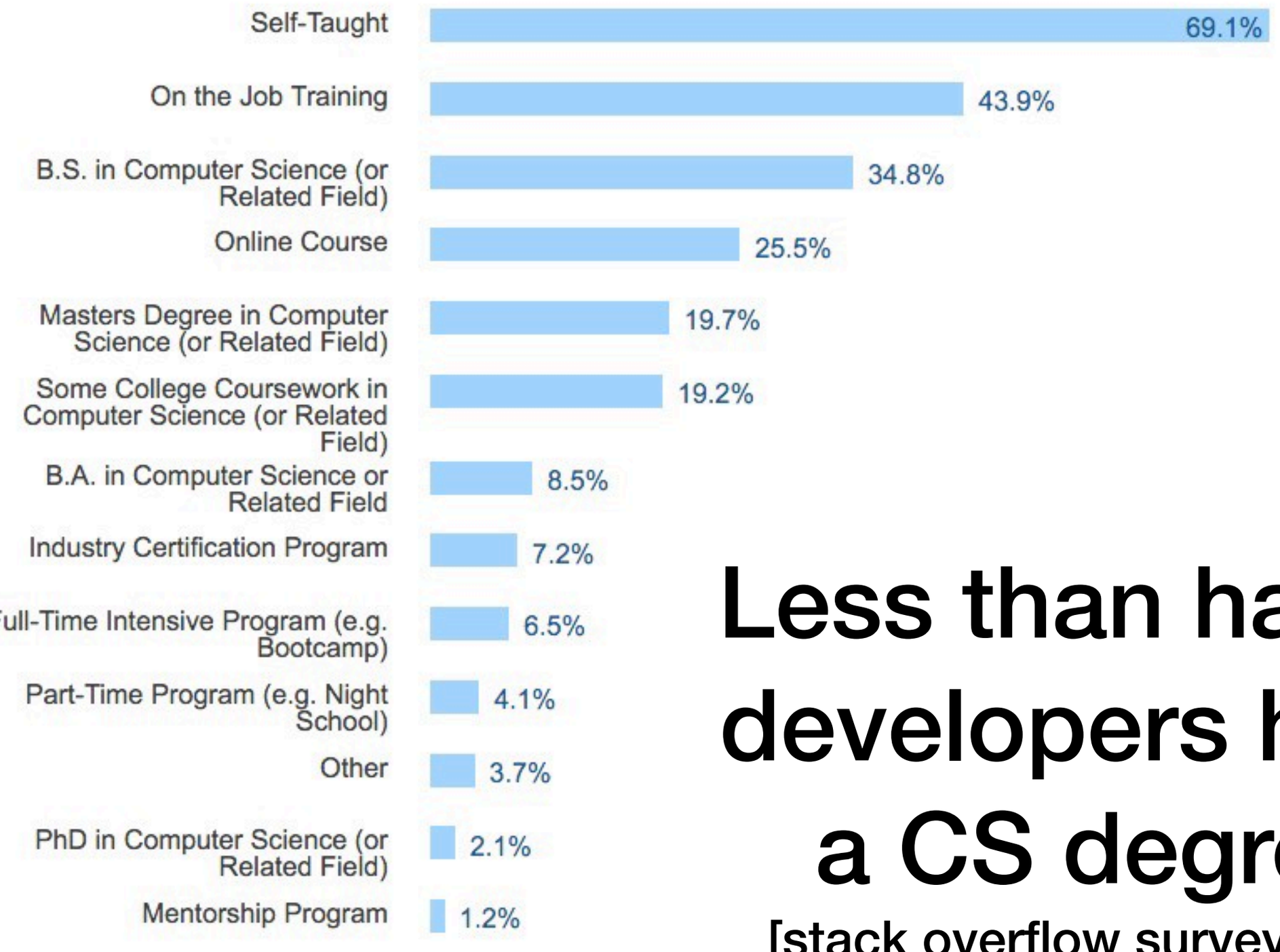
Smoking kills!

What do smokers think?!

54.9% of the time patients did not receive the recommended care for their condition

[New England Journal of Medicine 2003]

VIII. Education



40,183 responses from non-student developers

Less than half of developers have a CS degree

[stack overflow survey 2016]

(Often) People don't understand much.

- Our initial naïve expectation: People who write code for money understand it. Instead:
  - **"To build, I just press this button…"**
  - **"I'm just the security guy"**
  - **"That bug is in 3rd party code"**
  - **"Is it a leak? Author left years ago…"**
- People don't understand compilers.
  - **"Static" analysis? What is the performance overhead?**
  - **Business card at customer site: "Static analyzer" (?!)**
  - **"We use purify, why do we need your tool?"**
  - **Anything that finds bugs = testing.**
  - **"Think of it as super compiler warnings"**

coverity®
A Synopsys Company

"A couple billion lines of code later: static checking in the real world"
Dawson Engler

More general: A too-hard bug didn't happen.

- In fact, can be worse.
  - **People don't want to look stupid.**
  - **If they don't understand error, what will they do?**
- Social has *major* big impact on technical.
  - **User not same as tool builder.**
  - **Uninformed. Inattentive. Cruel.**
  - **HUGE problem. Prevents getting many things out in world.**
- Give up on error classes that need too much sophistication.
  - **statistical inference,**
  - **race conditions,**
  - **heap tracking**
  - **globals.**
  - **In some ways, checkers lag much behind our research ones.**

(Often) People don't understand much.

- Our initial naïve expectation: People who write code for money understand it. Instead:
  - **"To build, I just press this button…"**
  - **"I'm just the security guy"**
  - **"That bug is in 3rd party code"**
  - **"Is it a leak? Author left years ago…"**
- People don't understand compilers.
  - **"Static" analysis? What is the performance overhead?**
  - **Business card at customer site: "Static analyzer" (?!)**
  - **"We use purify, why do we need your tool?"**
  - **Anything that finds bugs = testing.**
  - **"Think of it as super compiler warnings"**

coverity®
A Synopsys Company

"A couple billion lines of code later: static checking in the real world"
Dawson Engler

More general: A too-hard bug didn't happen.

- In fact, can be worse.
  - **People don't want to look stupid.**
  - **If they don't understand error, what will they do?**
- Social has *major* big impact on technical.
  - **User not same as tool builder.**
  - **Uninformed. Inattentive. Cruel.**
  - **HUGE problem. Prevents getting many things out in world.**
- Give up on error classes that need too much sophistication.
  - **statistical inference,**
  - **race conditions,**
  - **heap tracking**
  - **globals.**
  - **In some ways, checkers lag much behind our research ones.**

(Often) People don't understand much.

- Our initial naïve expectation: People who write code for money understand it. Instead:
  - **"To build, I just press this button…"**
  - **"I'm just the security guy"**
  - **"That bug is in 3rd party code"**
  - **"Is it a leak? Author left years ago…"**
- People don't understand compilers.
  - **"Static" analysis? What is the performance overhead?**
  - **Business card at customer site: "Static analyzer" (?!)**
  - **"We use purify, why do we need your tool?"**
  - **Anything that finds bugs = testing.**
  - **"Think of it as super compiler warnings"**

coverity®
A Synopsys Company

"A couple billion lines of code later: static checking in the real world"
Dawson Engler

More general: A too-hard bug didn't happen.

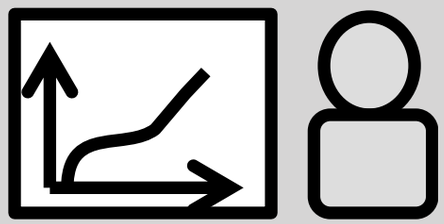
- In fact, can be worse.
  - **People don't want to look stupid.**
  - **If they don't understand error, what will they do?**
- Social has *major* big impact on technical.
  - **User not same as tool builder.**
  - **Uninformed. Inattentive. Cruel.**
  - **HUGE problem. Prevents getting many things out in world.**
- Give up on error classes that need too much sophistication.
  - **statistical inference,**
  - **race conditions,**
  - **heap tracking**
  - **globals.**
  - **In some ways, checkers lag much behind our research ones.**

Do practitioners agree with each other?

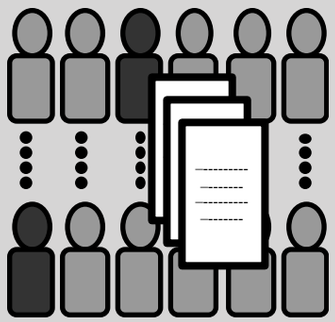
SONY


SONY
10:35
Google
10 35
Thursday • September 12
Walkman
Album
Movies
PlayStation

Practitioners do not agree with each other

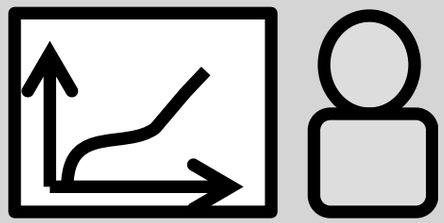


Presentation

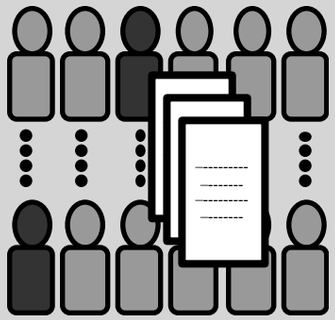


Survey
(93 stakeholders)

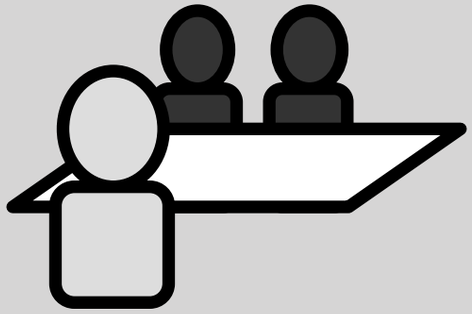
Practitioners do not agree with each other



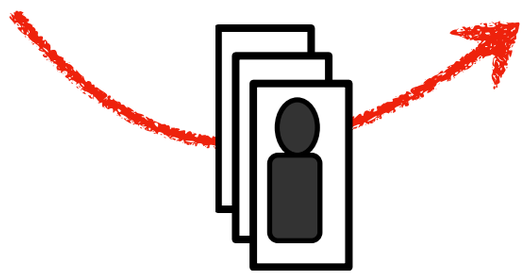
Presentation



Survey
(93 stakeholders)

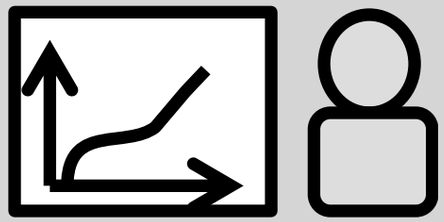


Semi-structured interviews
(15 Key engineers)

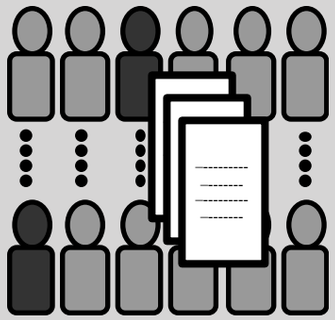


Interviewee’s list

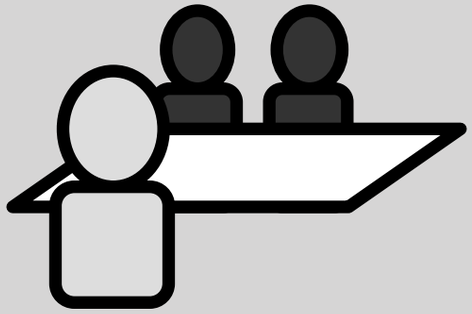
Practitioners do not agree with each other



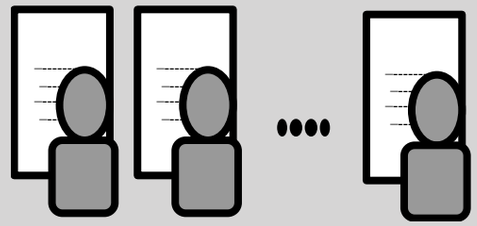
Presentation



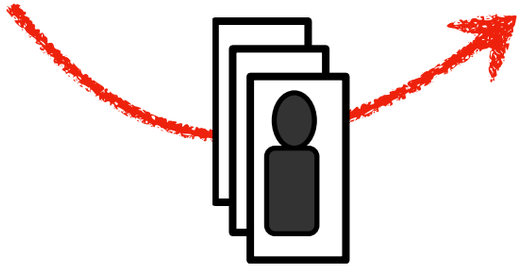
Survey
(93 stakeholders)



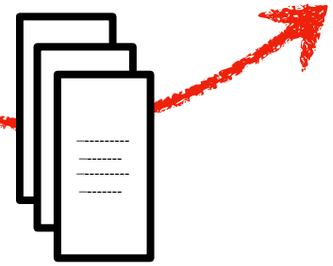
Semi-structured interviews
(15 Key engineers)



Validation survey Interviews
(25 Senior engineers)

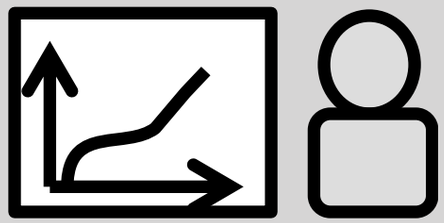


Interviewee’s list

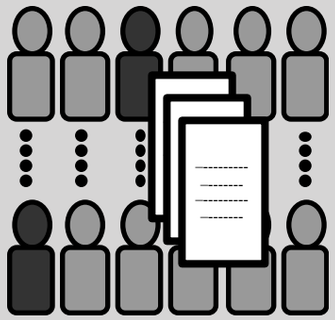


Implications

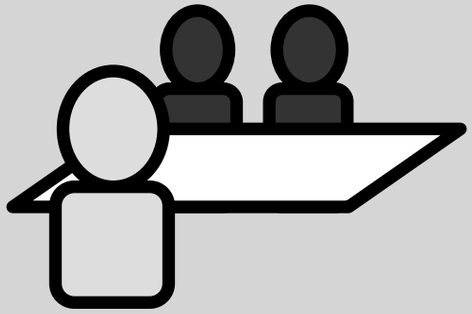
Practitioners do not agree with each other



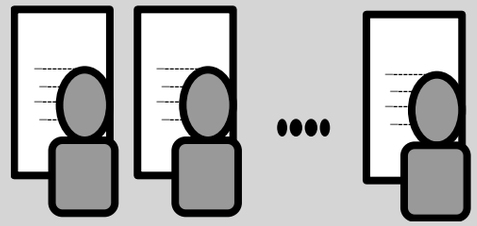
Presentation



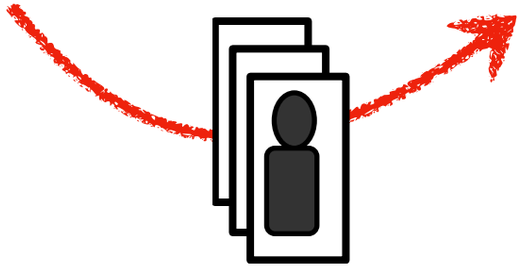
Survey
(93 stakeholders)



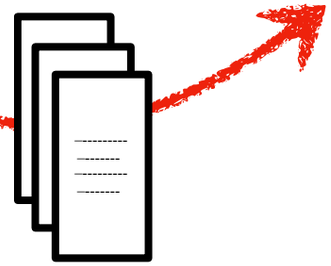
Semi-structured interviews
(15 Key engineers)



Validation survey Interviews
(25 Senior engineers)



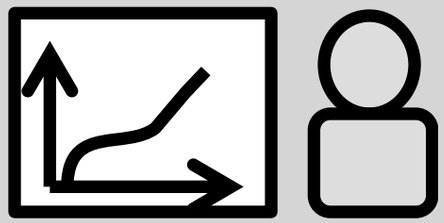
Interviewee's list



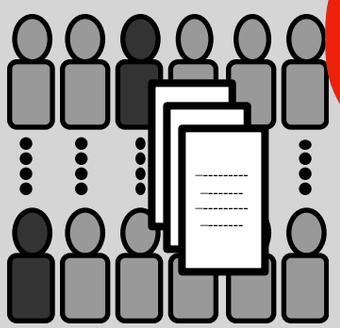
Implications

Verified implications

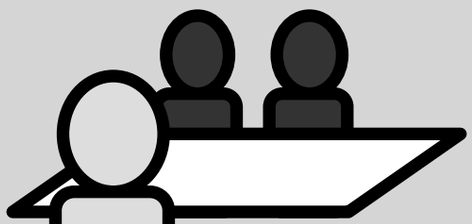
Practitioners do not agree with each other



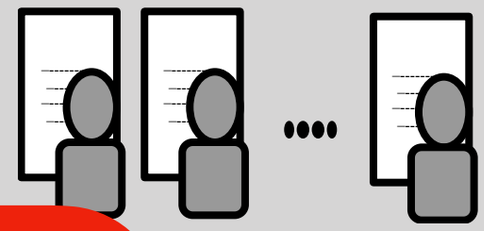
Presentation



Survey
(93 stakeholders)

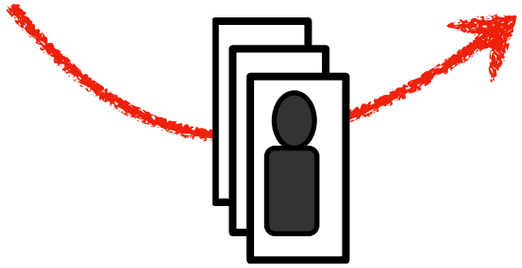


interviews
(15 Key engineers)



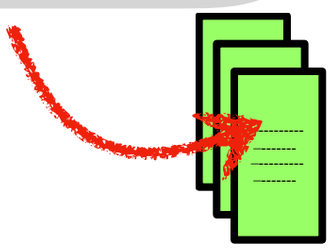
dation survey
Interviews
(25 Senior engineers)

As low as 68% agreement

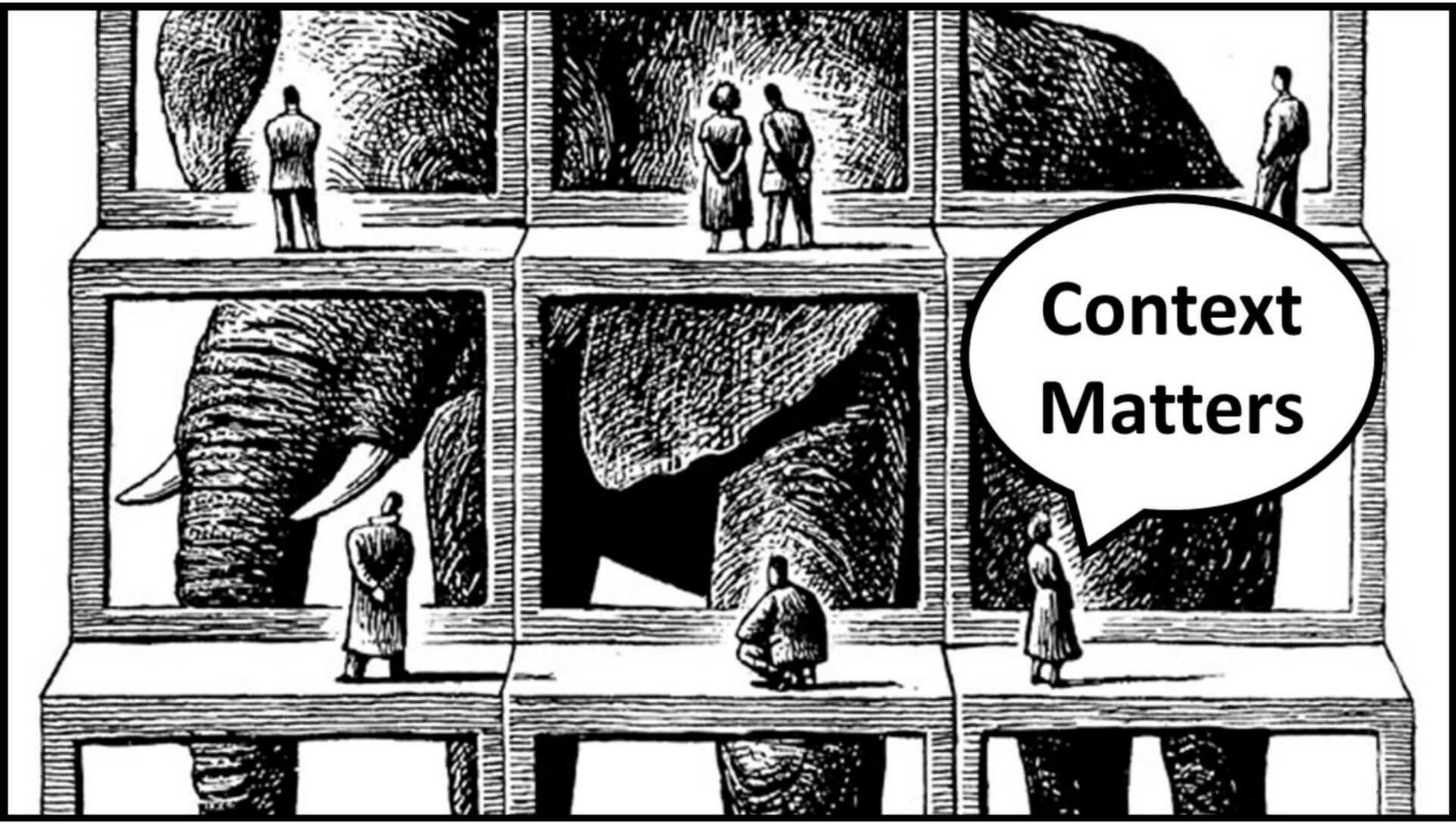


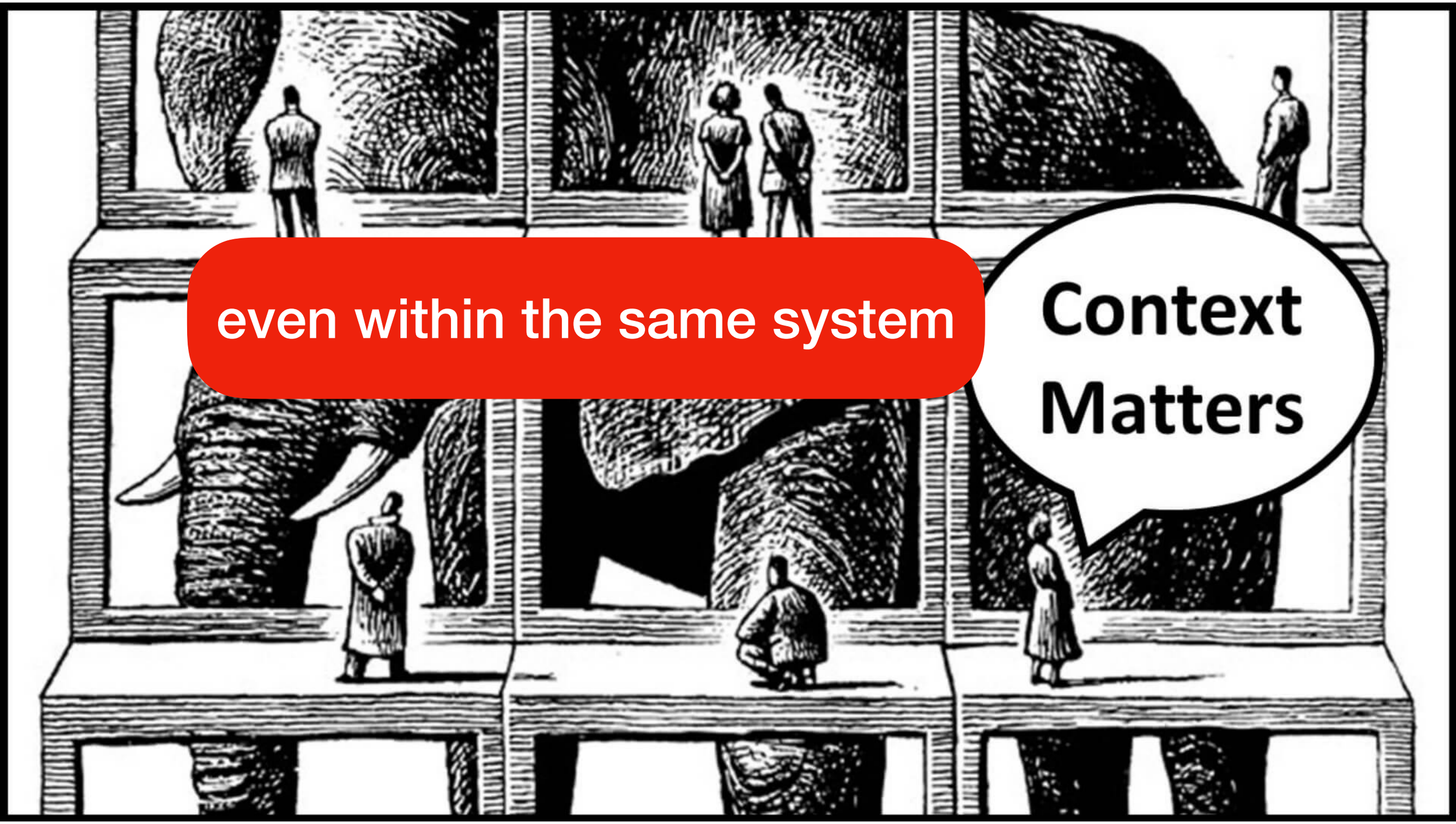
Interviewee's list

Implications

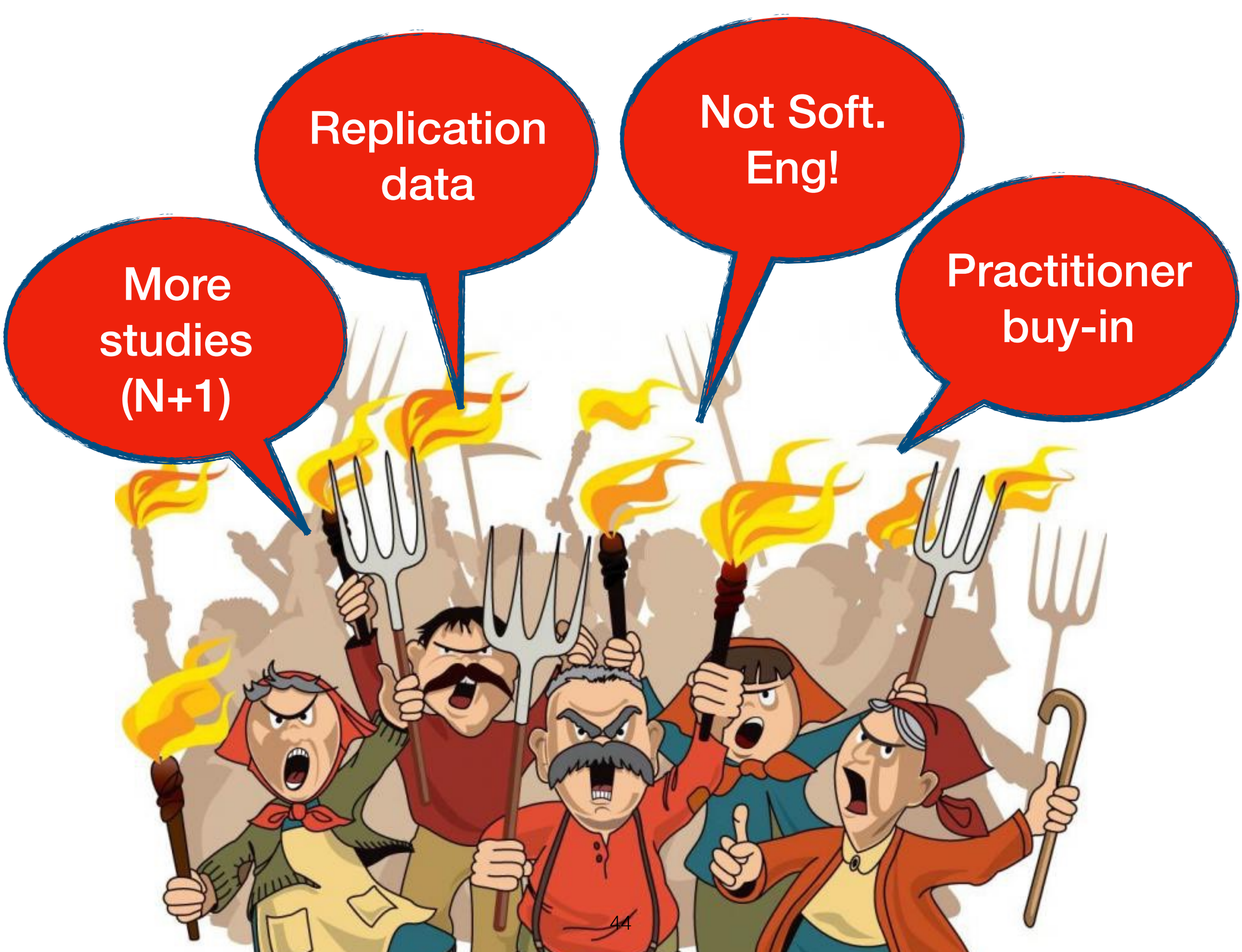


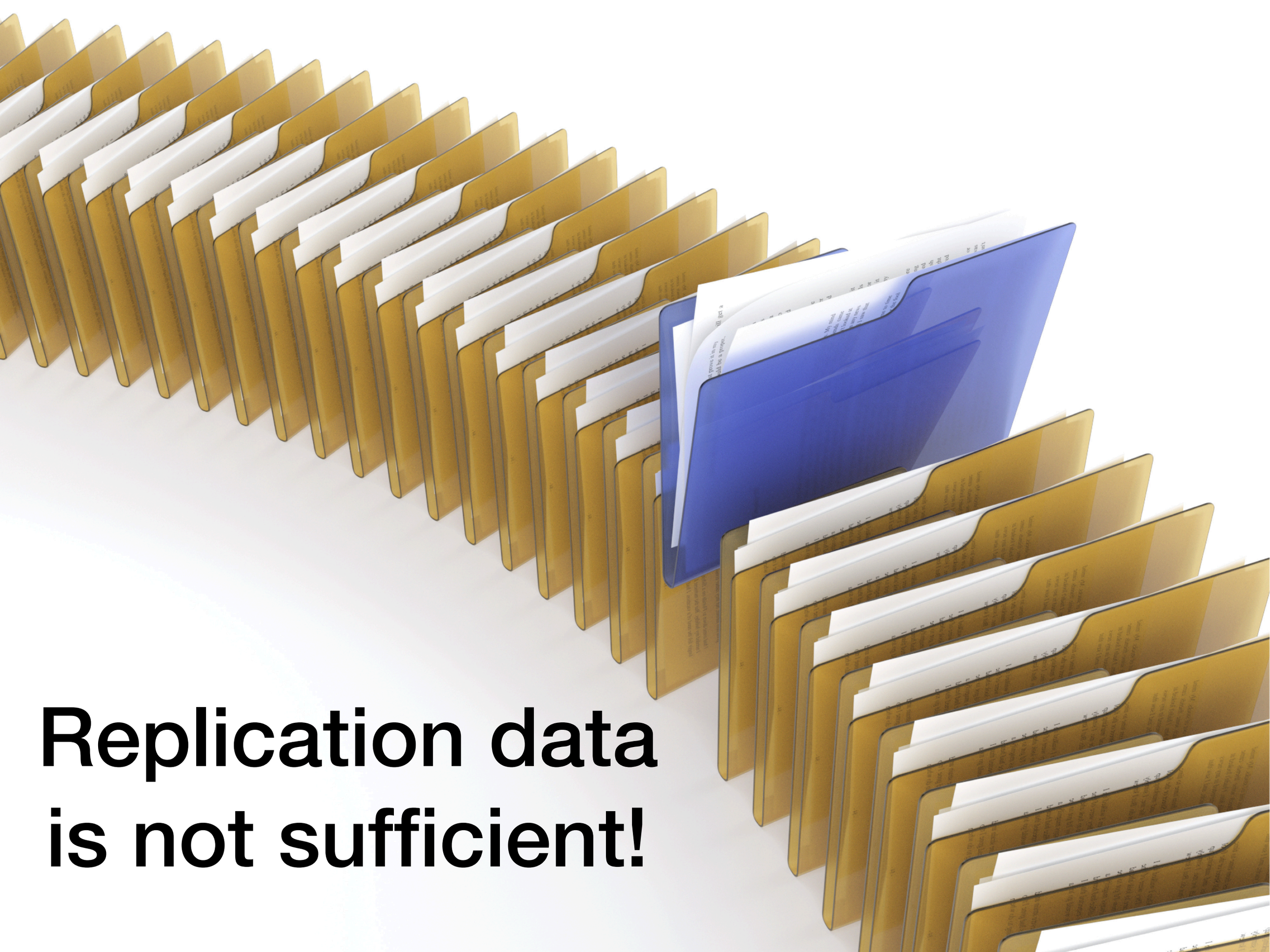
Verified implications


Context
Matters


even within the same system
Context Matters

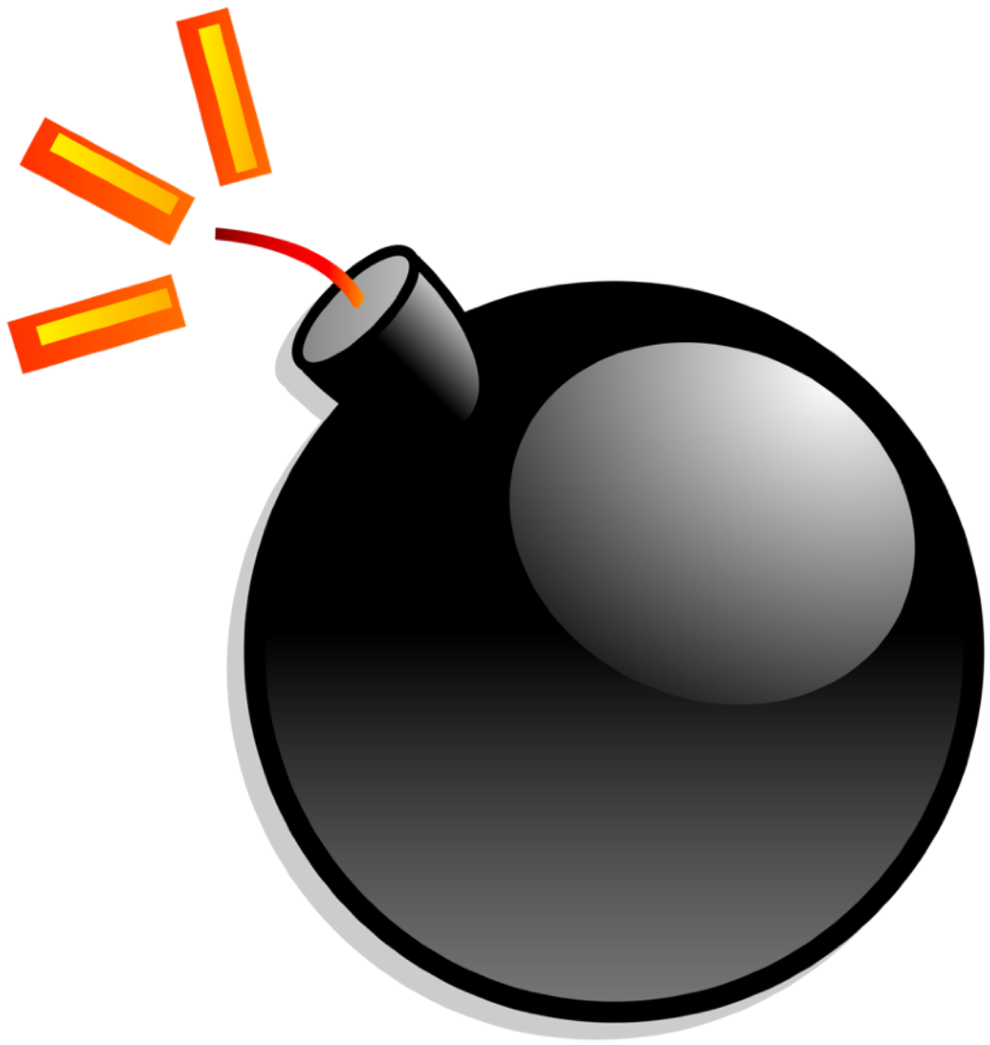

More studies (N+1)
Not Soft. Eng!
Practitioner buy-in


Replication data
Not Soft. Eng!
More studies (N+1)
Practitioner buy-in



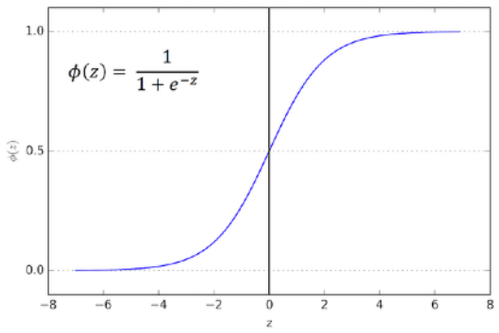
Replication data is not sufficient!

MSR Today: Easy Peasy!



Limited knowledge of machine learning is a serious risk

A Simple Example



Logistic Regression

A Simple Example

Bugs ~ f(LOC, CC_max)

NOT about predicting bugs

Empirical Hypothesis Testing
(is *complex code more risky?*)

Including correlated metrics

Model M1: Bugs ~ f(CC_max, CC_avg, PAR_max, FOUT_max)
Model M2: Bugs ~ f(CC_avg, CC_max, PAR_max, FOUT_max)

Including correlated metrics

Model M1: Bugs ~ f(CC_max, CC_avg, PAR_max, FOUT_max)
Model M2: Bugs ~ f(CC_avg, CC_max, PAR_max, FOUT_max)

	M1 (AUC=0.78)		**M2** (AUC=0.78)	
Metrics	**Position**	**ANOVA**	**Position**	**ANOVA**
CC_max	[1]	74%	[2]	19%
CC_avg	[2]	2%	[1]	58%
PAR_max	[3]	16%	[3]	16%
FOUT_max	[4]	7%	[4]	7%

Including correlated metrics

Model M1: Bugs ~ f(CC_max, CC_avg, PAR_max, FOUT_max)
Model M2: Bugs ~ f(CC_avg, CC_max, PAR_max, FOUT_max)

	M1 (AUC=0.78)		**M2** (AUC=0.78)	
Metrics	**Position**	**ANOVA**	**Position**	**ANOVA**
CC_max	[1]	74%	[2]	19%
CC_avg	[2]	2%	[1]	58%
PAR_max	[3]	16%	[3]	16%
FOUT_max	[4]	7%	[4]	7%

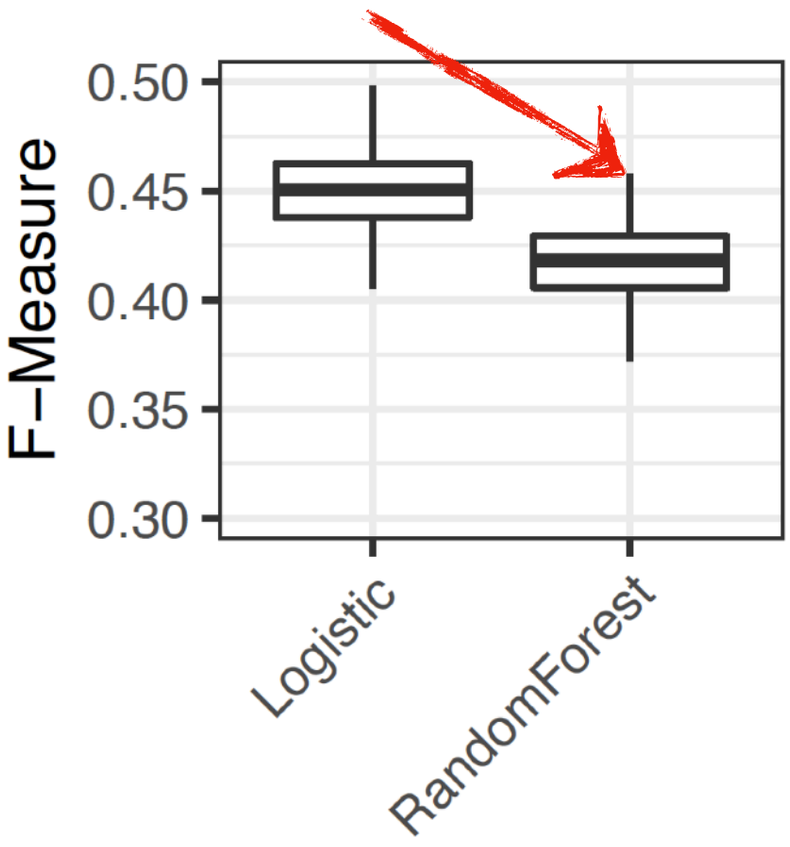
Inc cs

60% of 2000-2011 studies do not handle correlated metrics [Shihab 2012]

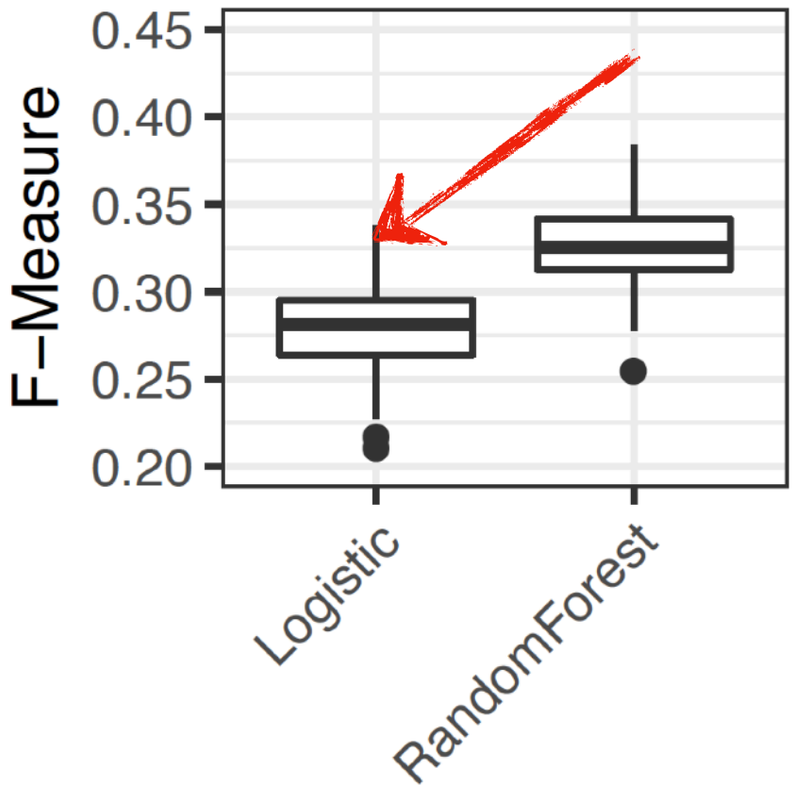
Model M1: Bugs ~ f(CC_max, CC_avg, PAR_max, FOUT_max)
Model M2: Bugs ~ f(CC_avg, CC_max, PAR_max, FOUT_max)

	M1 (AUC=0.78)		**M2** (AUC=0.78)	
Metrics	**Position**	**ANOVA**	**Position**	**ANOVA**
CC_max	[1]	74%	[2]	19%
CC_avg	[2]	2%	[1]	58%
PAR_max	[3]	16%	[3]	16%
FOUT_max	[4]	7%	[4]	7%

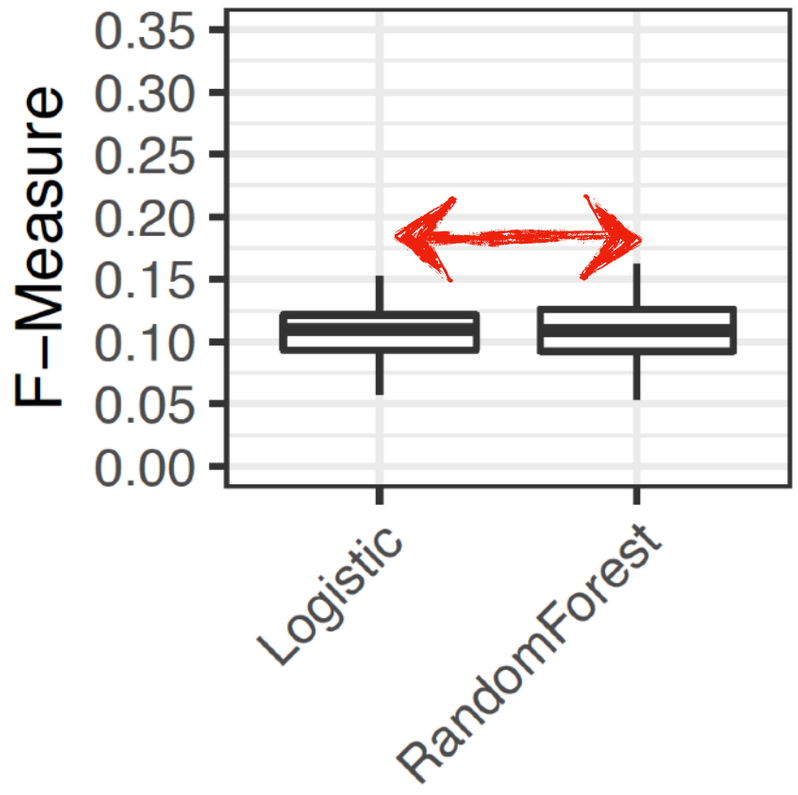
Using F-measure in studies



Threshold = 0.2

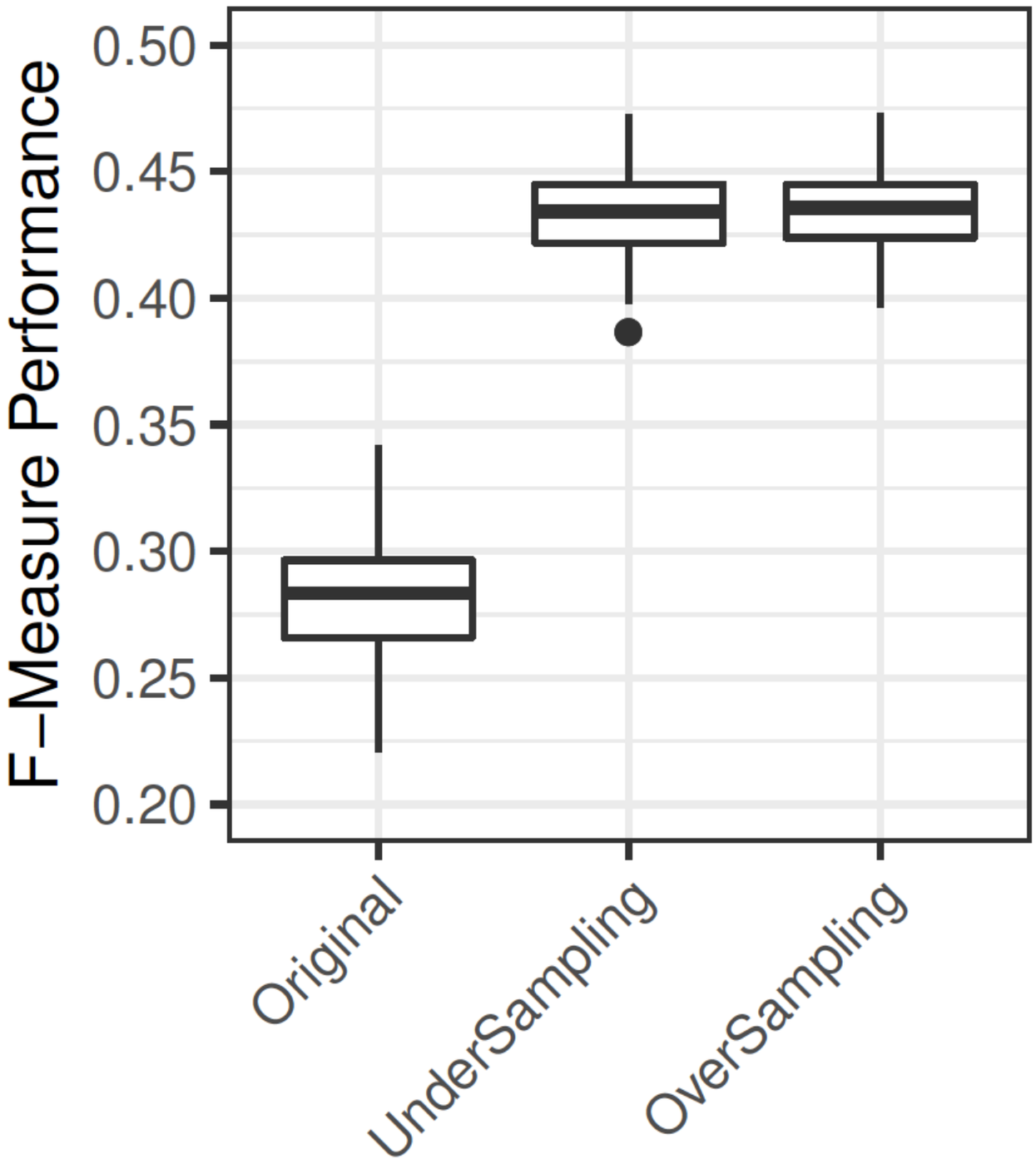


Threshold = 0.5

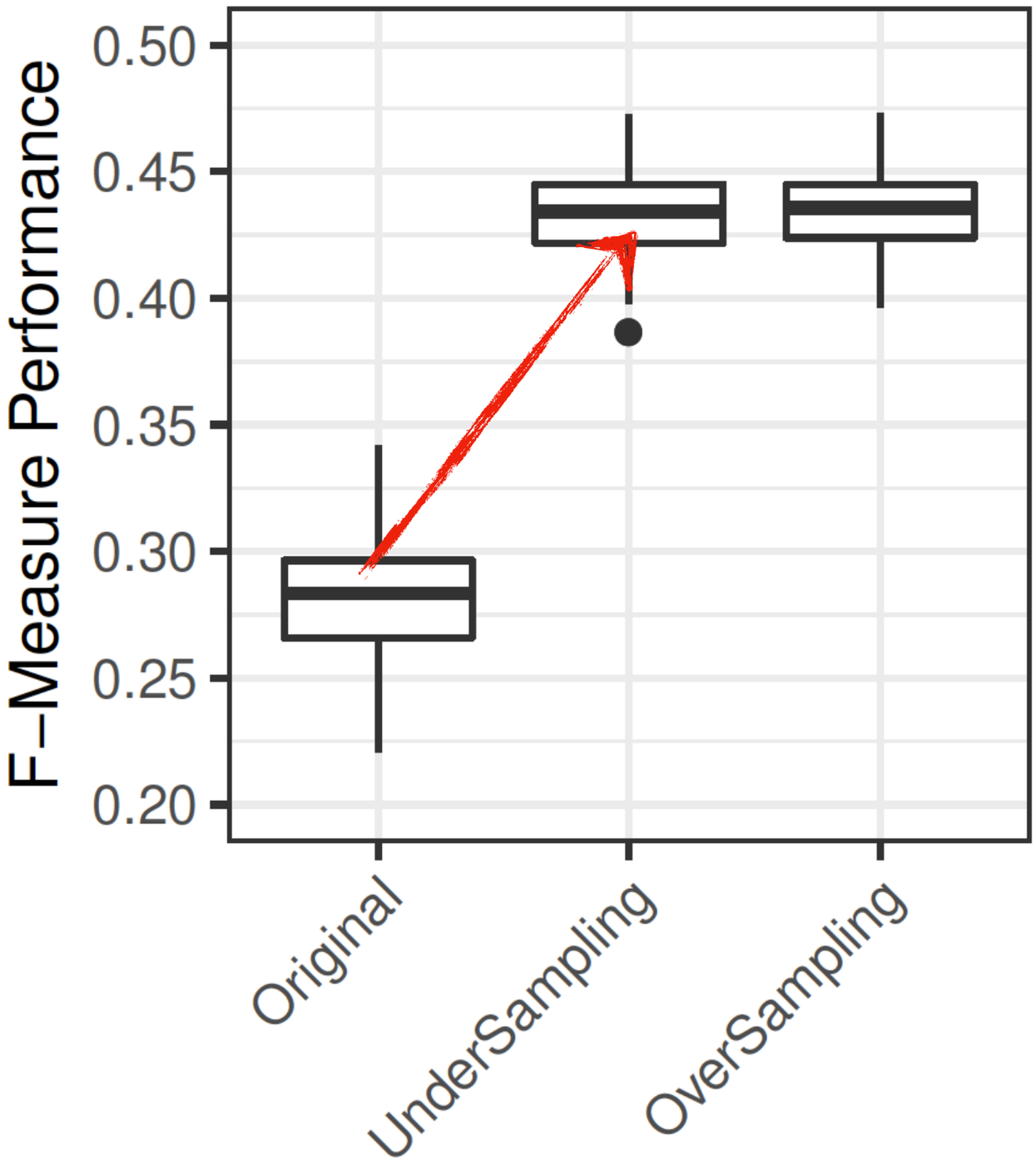


Threshold = 0.8

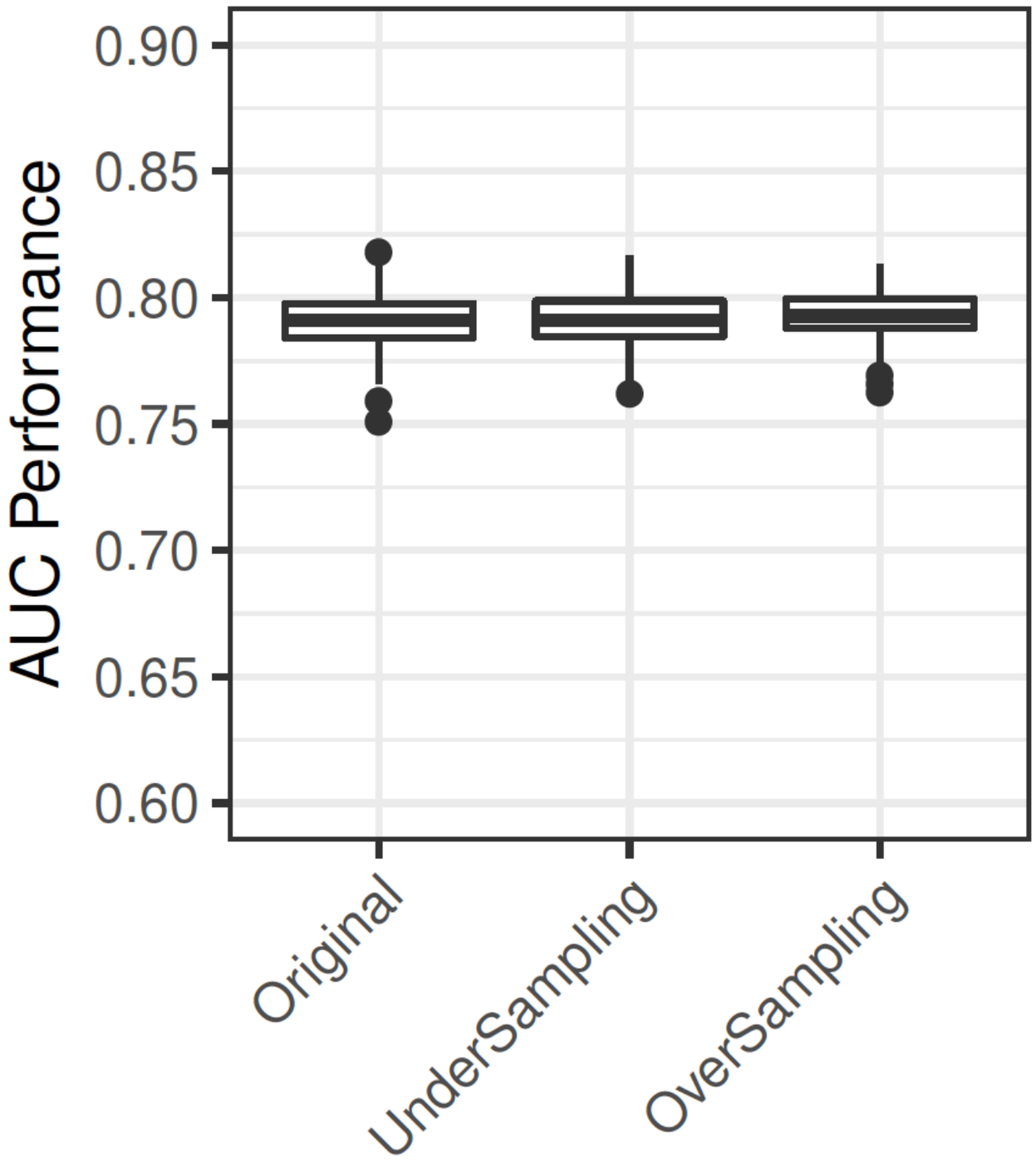
Resampling imbalanced data

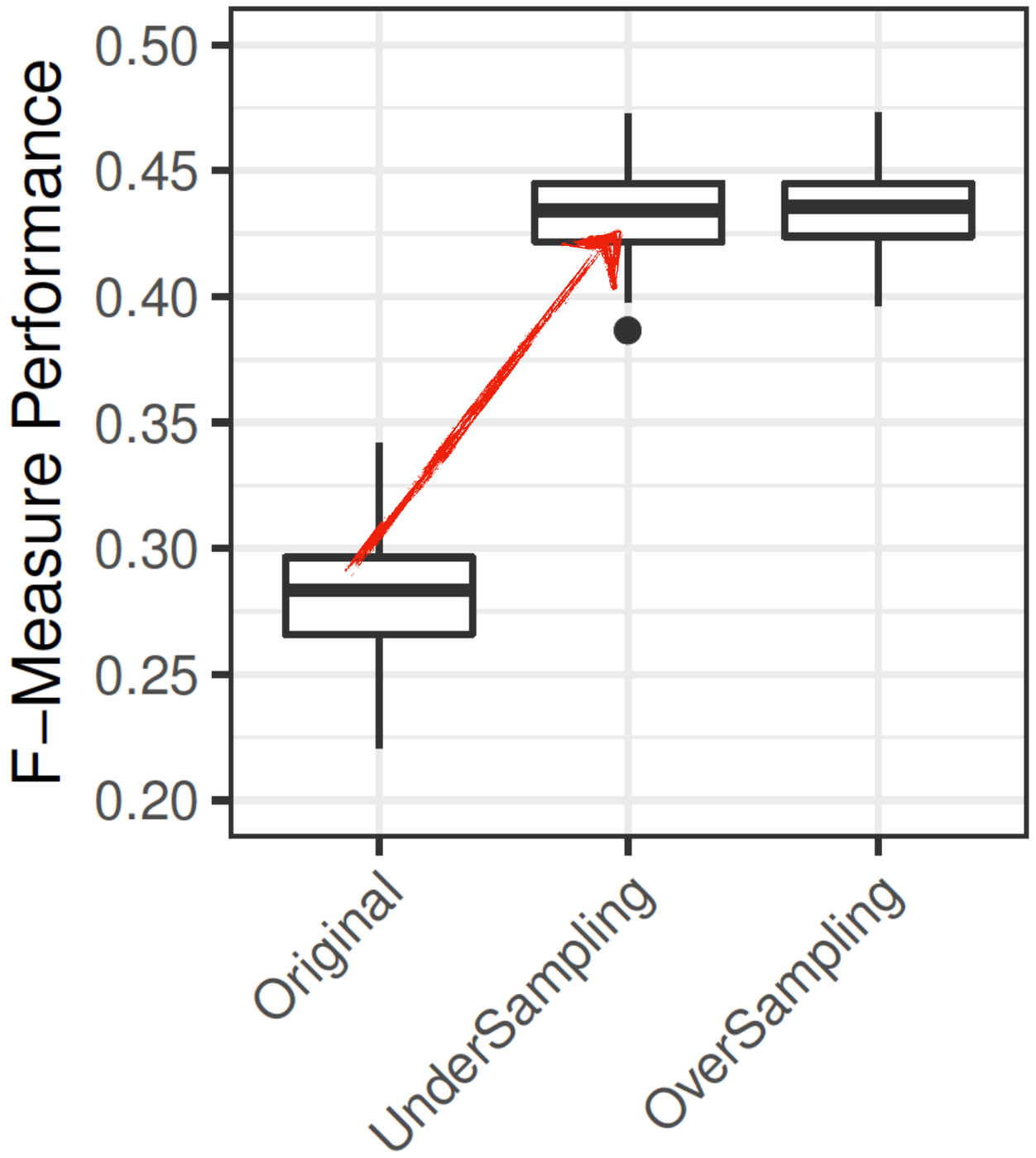


Resampling imbalanced data

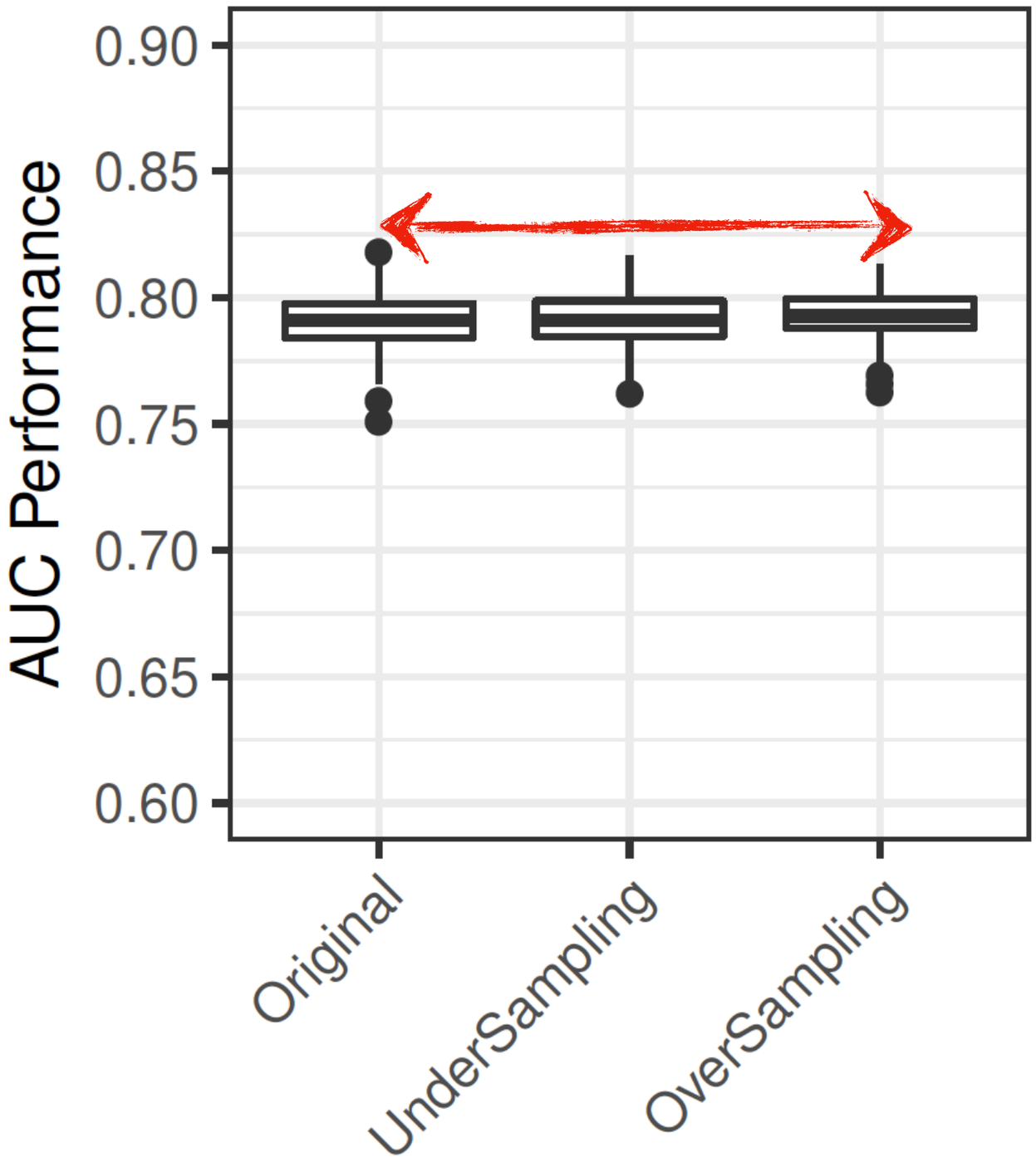


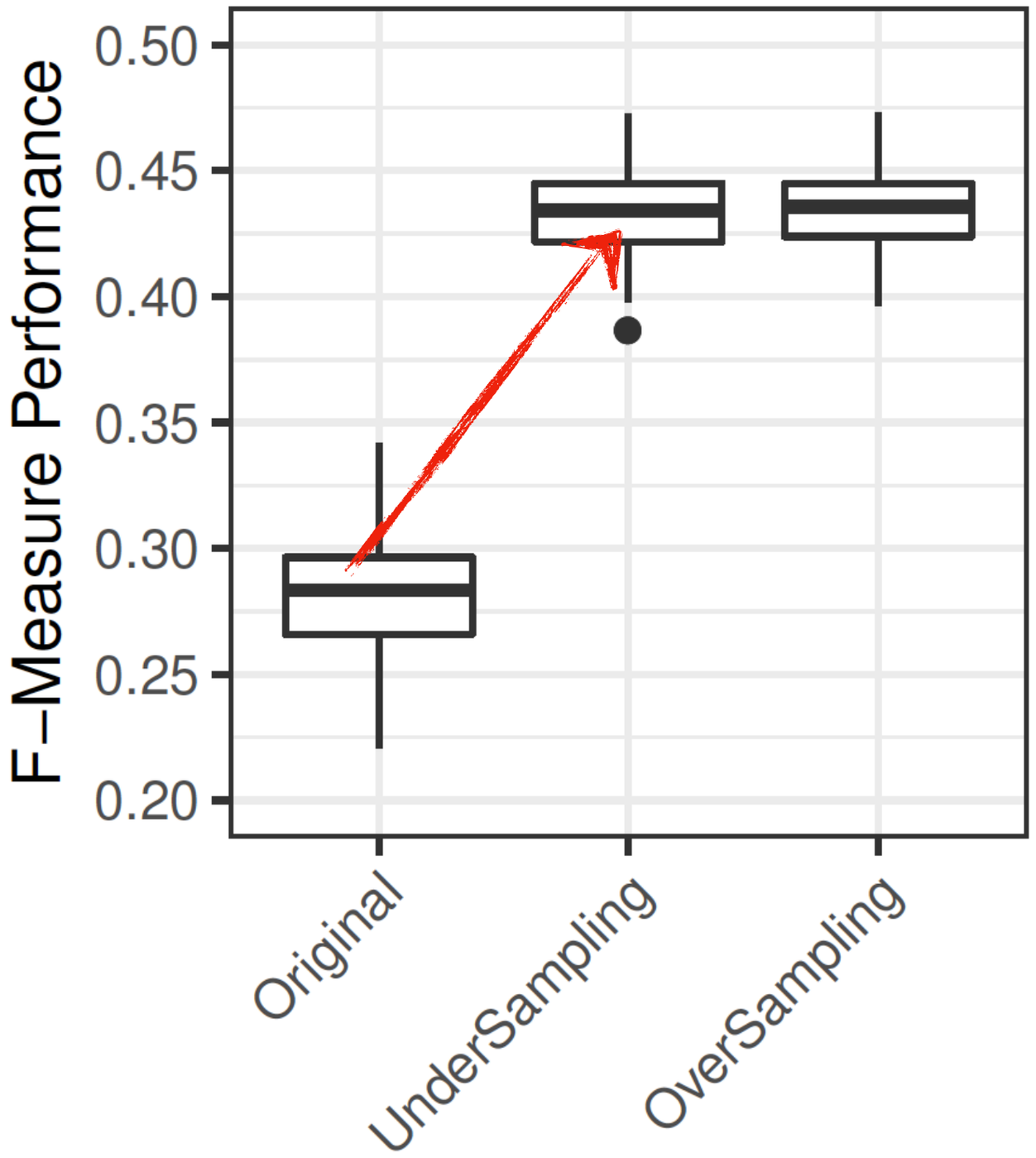
Resampling imbalanced data



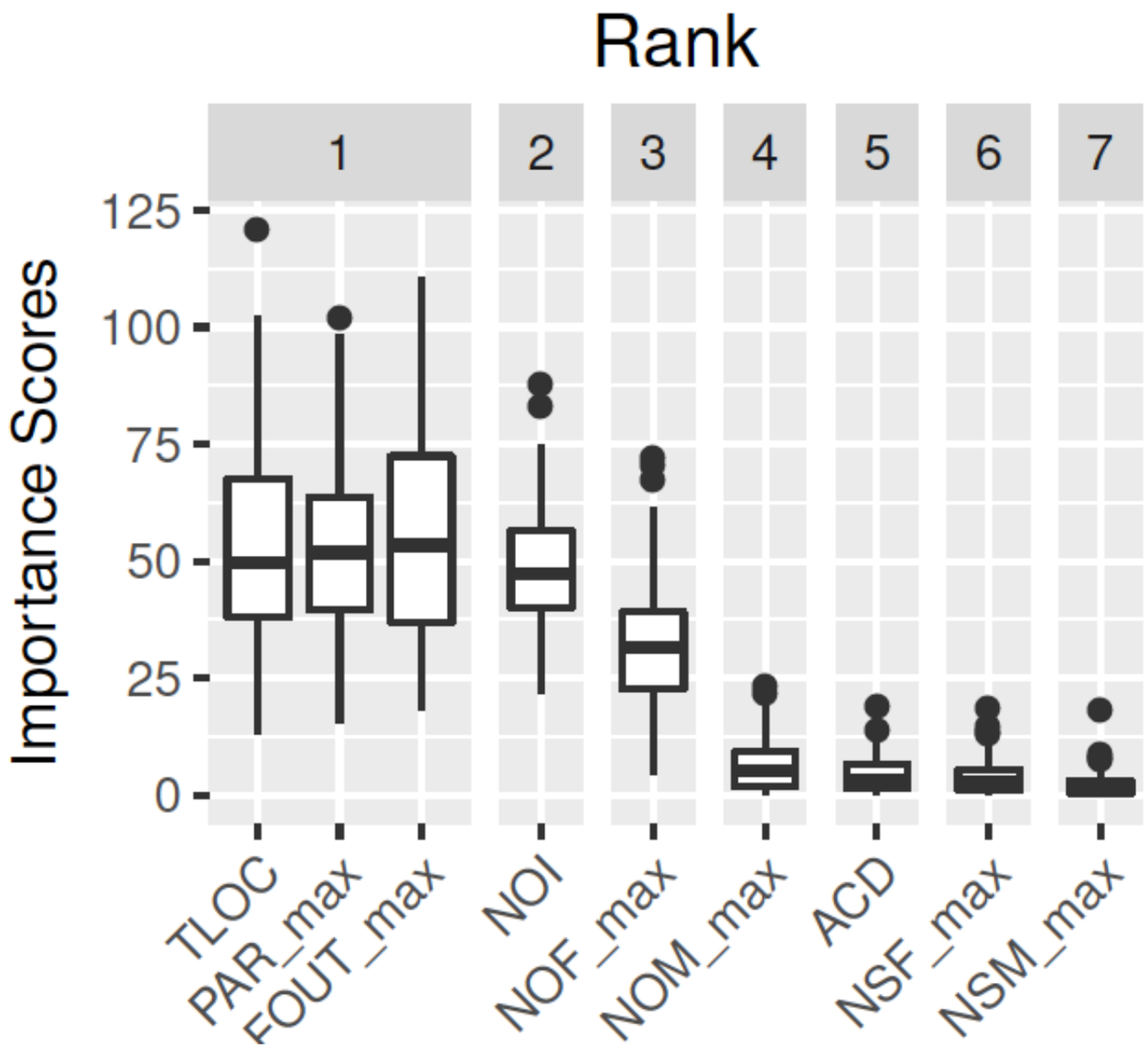


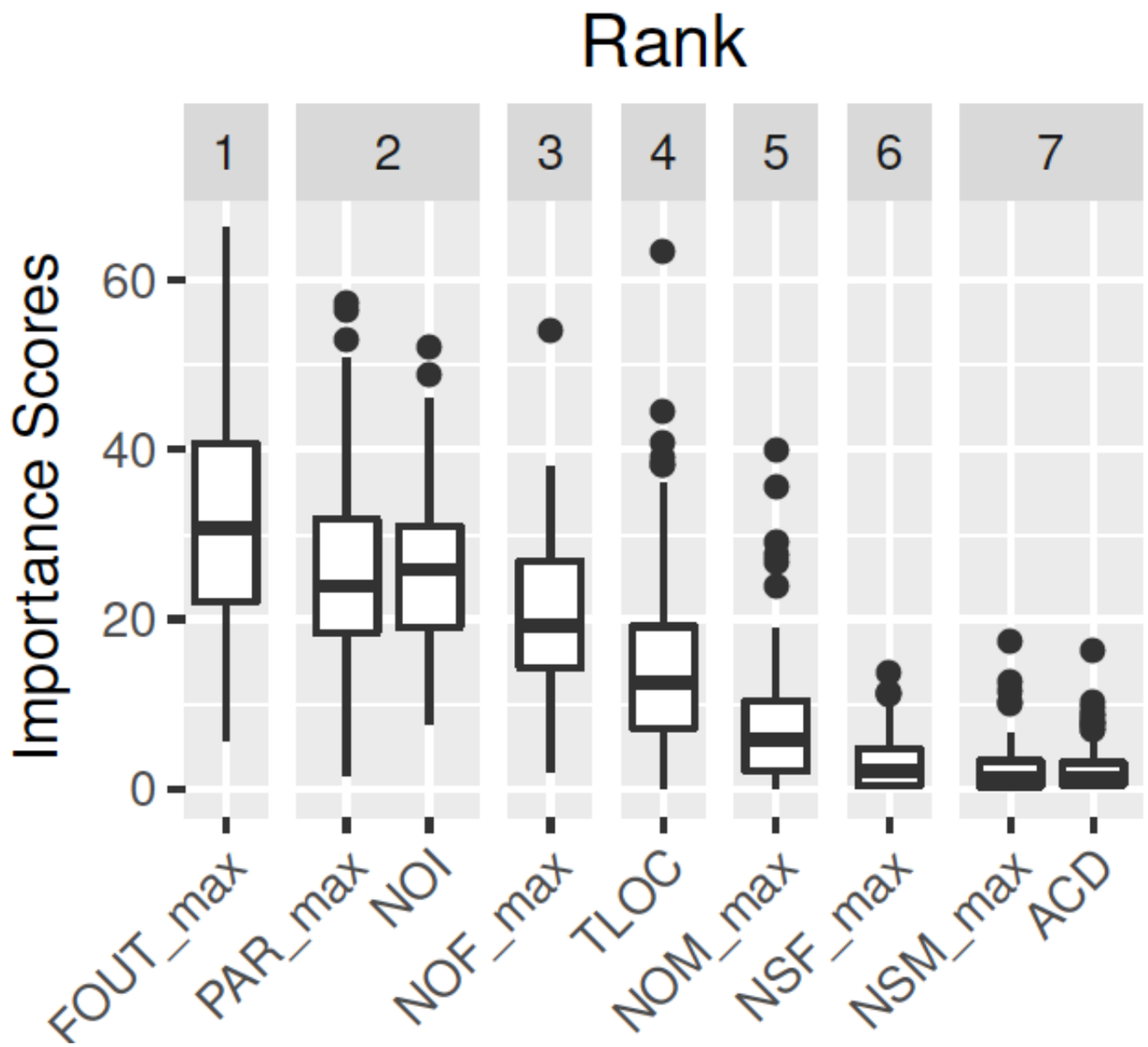
Resampling imbalanced data



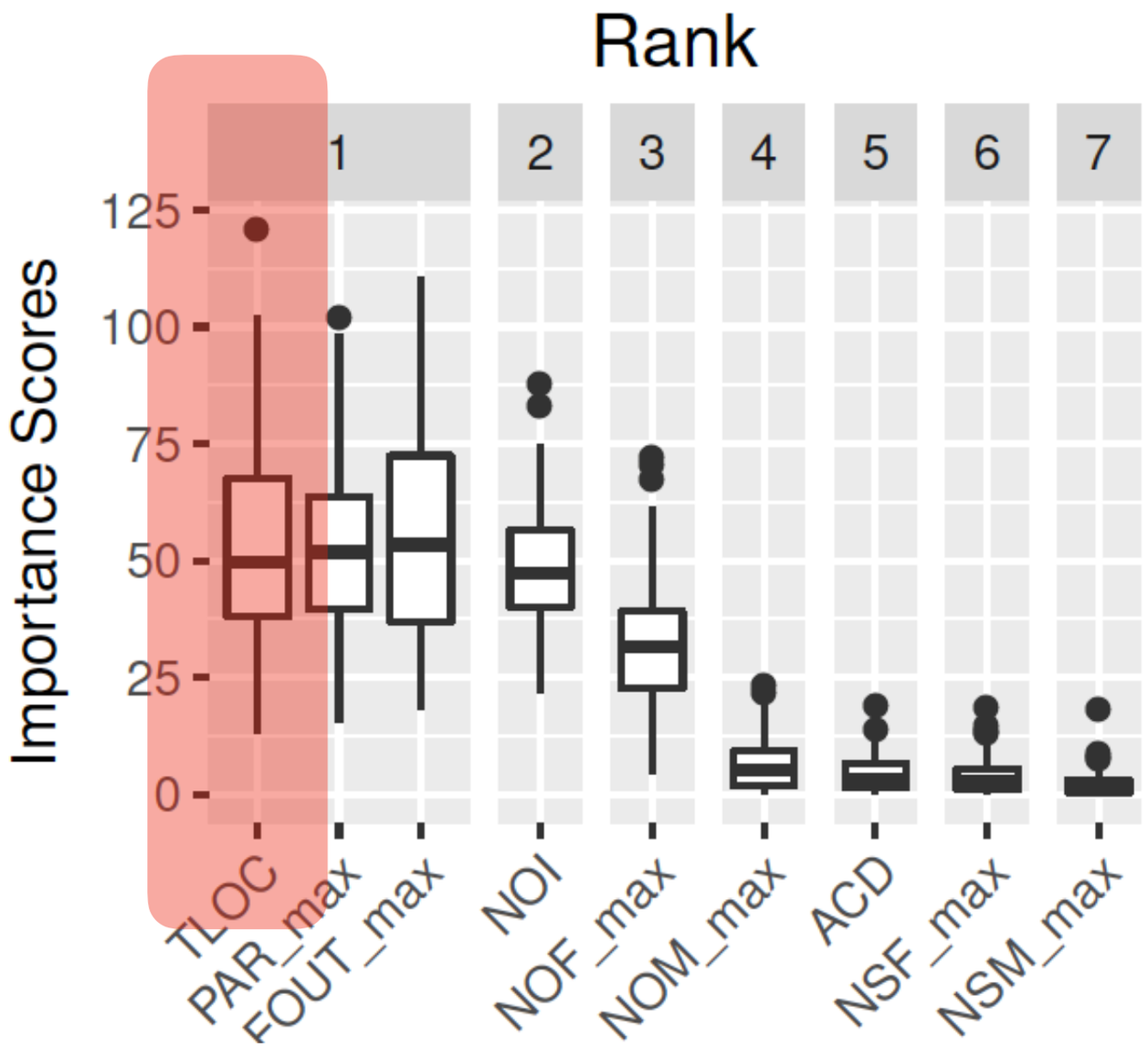


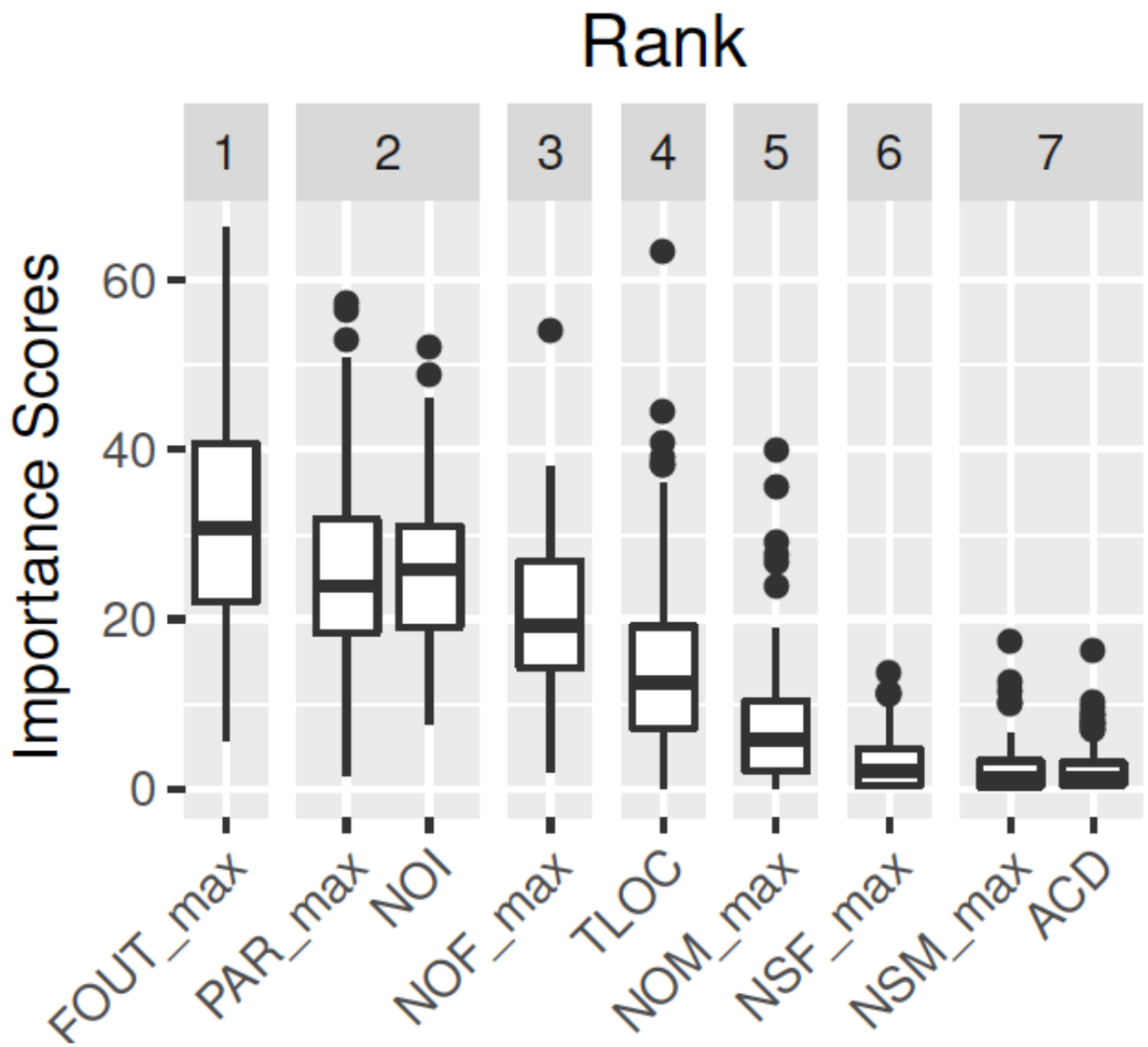
Variable importance: Original vs under sampled data



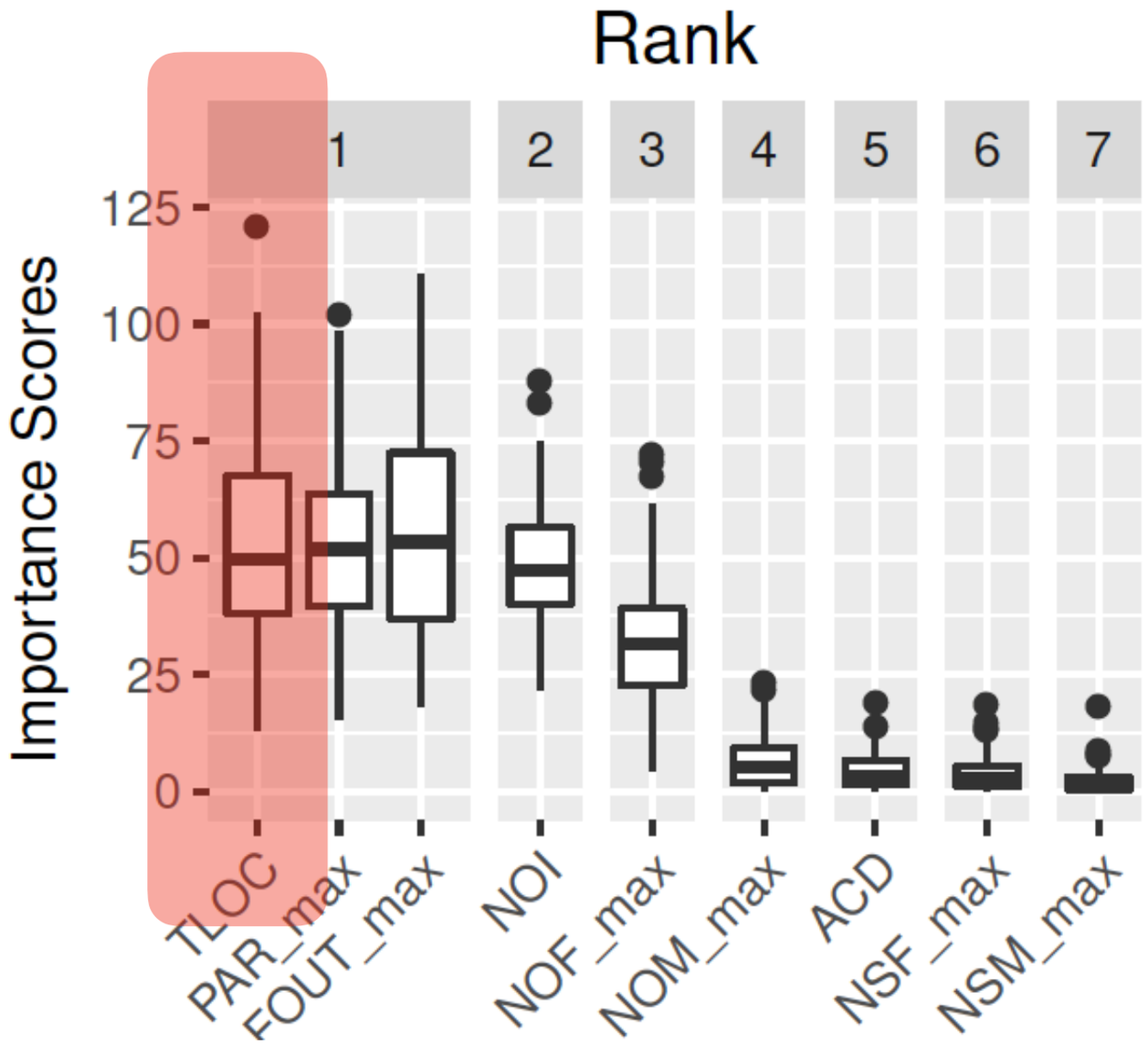


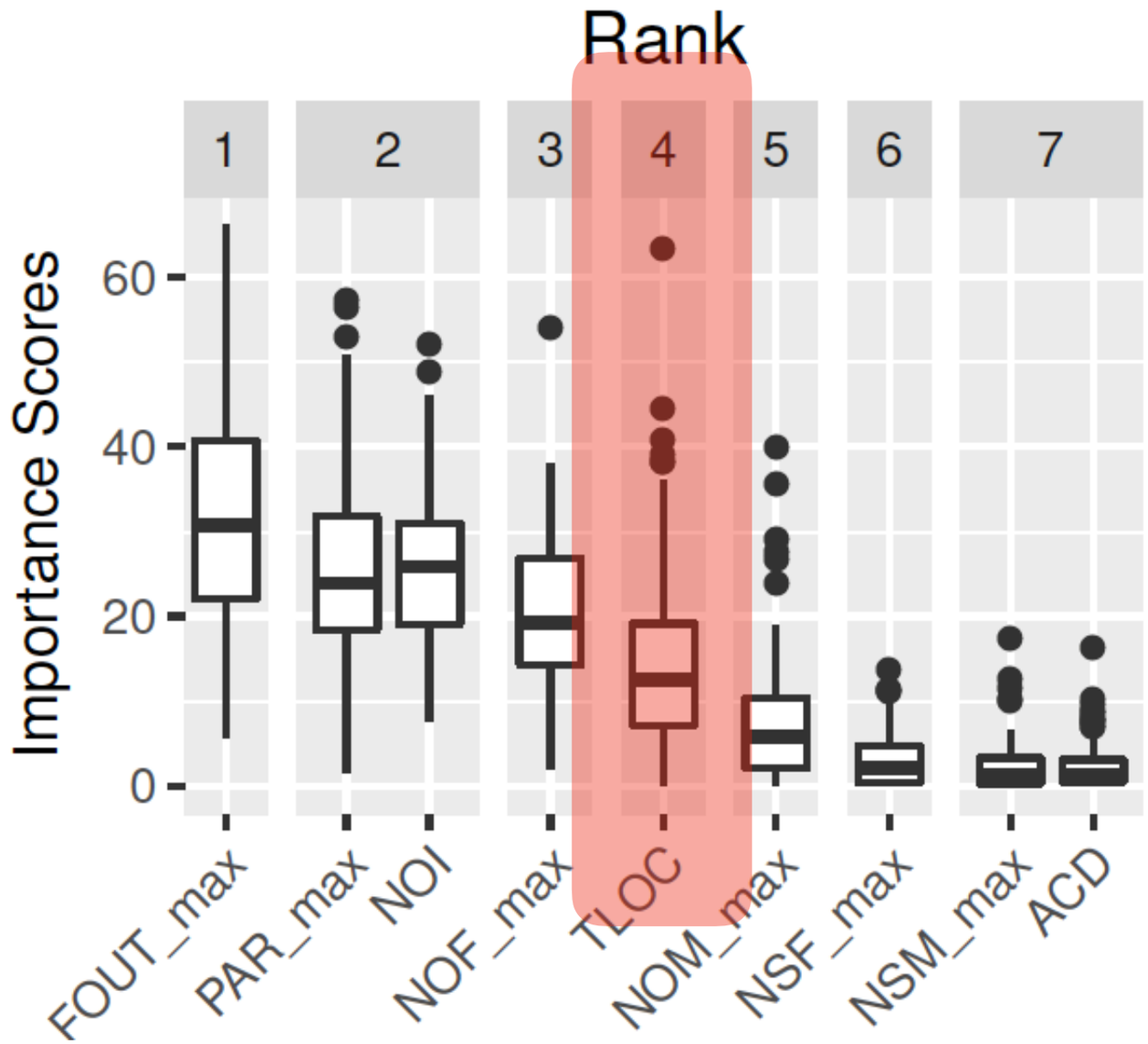
Variable importance: Original vs under sampled data





Variable importance: Original vs under sampled data





Not including control metrics

Model M1: Bugs ~ f(, CC_max, PAR_max, FOUT_max)
Model M2: Bugs ~ f(TLOC, CC_max, PAR_max, FOUT_max)

Not including control metrics

Model M1: Bugs ~ f(, CC_max, PAR_max, FOUT_max)
Model M2: Bugs ~ f(TLOC, CC_max, PAR_max, FOUT_max)

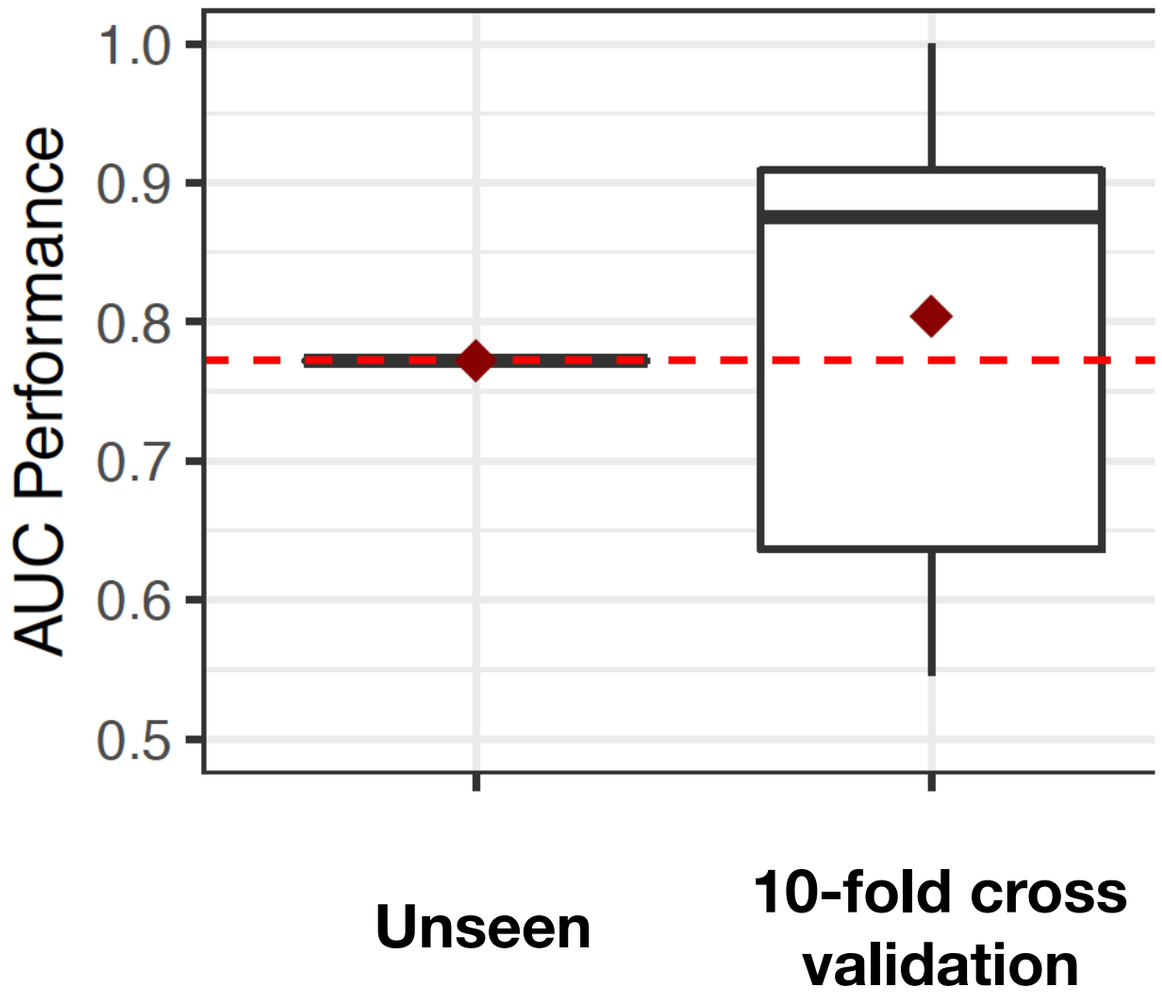
	M1	M2
Metrics	AUC=0.78	AUC=0.79
Control Metric		
TLOC	-	82%
Studied Metrics		
CC_max	76%	9%
PAR_max	17%	7%
FOUT_max	8%	2%

Not including control metrics

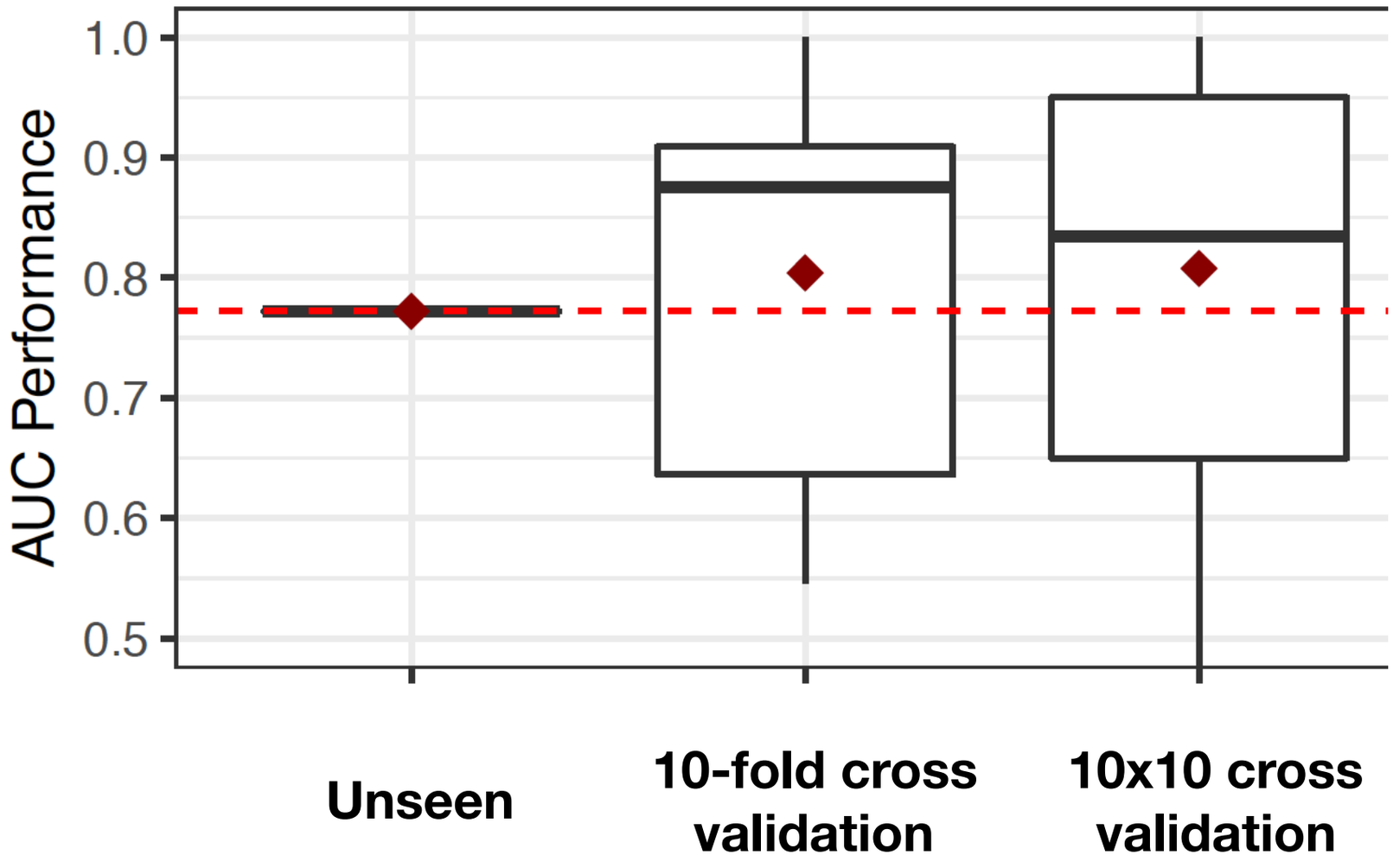
Model M1: Bugs ~ f(, CC_max, PAR_max, FOUT_max)
Model M2: Bugs ~ f(TLOC, CC_max, PAR_max, FOUT_max)

	M1	M2
Metrics	AUC=0.78	AUC=0.79
Control Metric		
TLOC	-	82%
Studied Metrics		
CC_max	76%	9%
PAR_max	17%	7%
FOUT_max	8%	2%

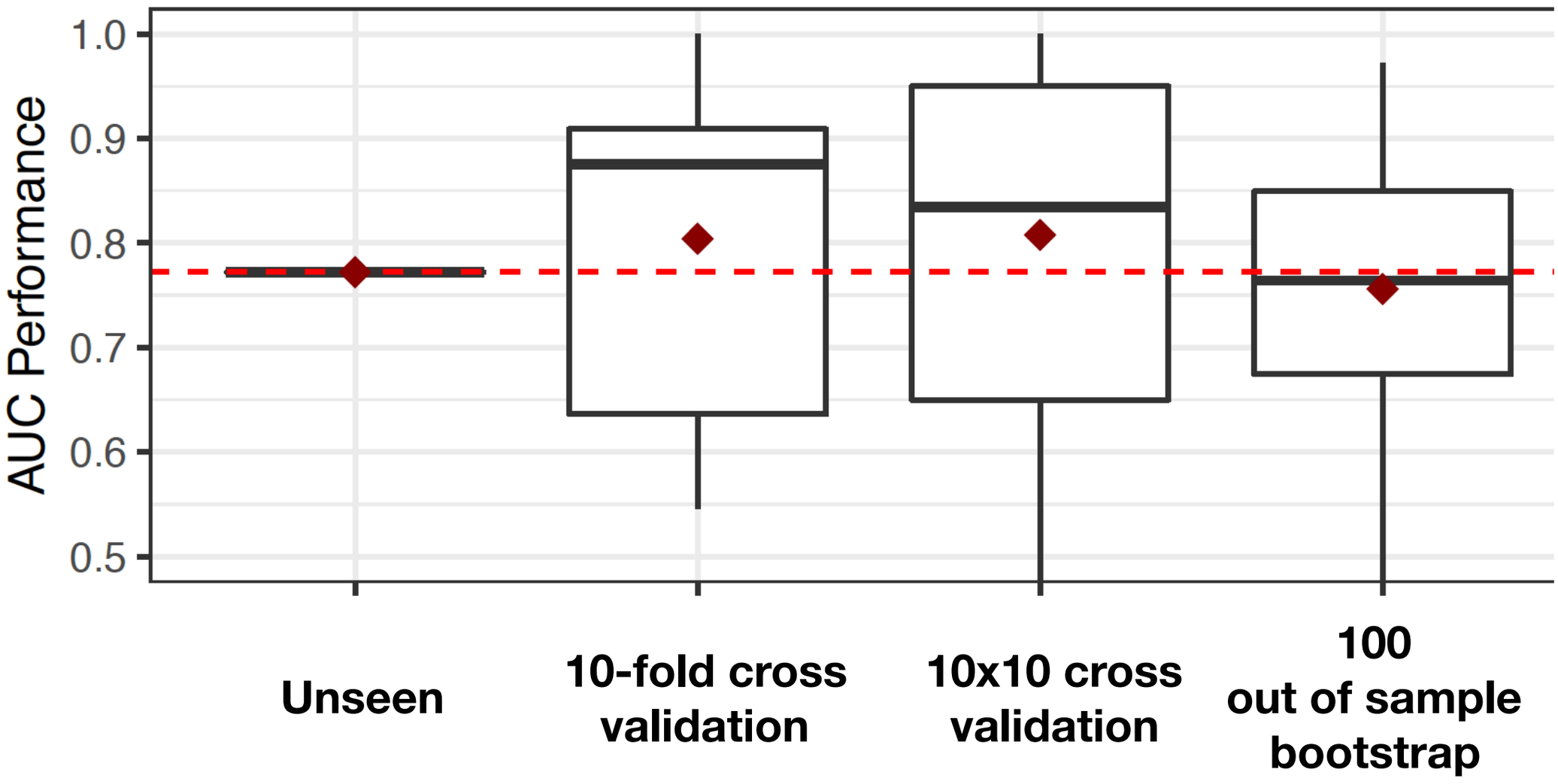
Using 10 fold cross validation



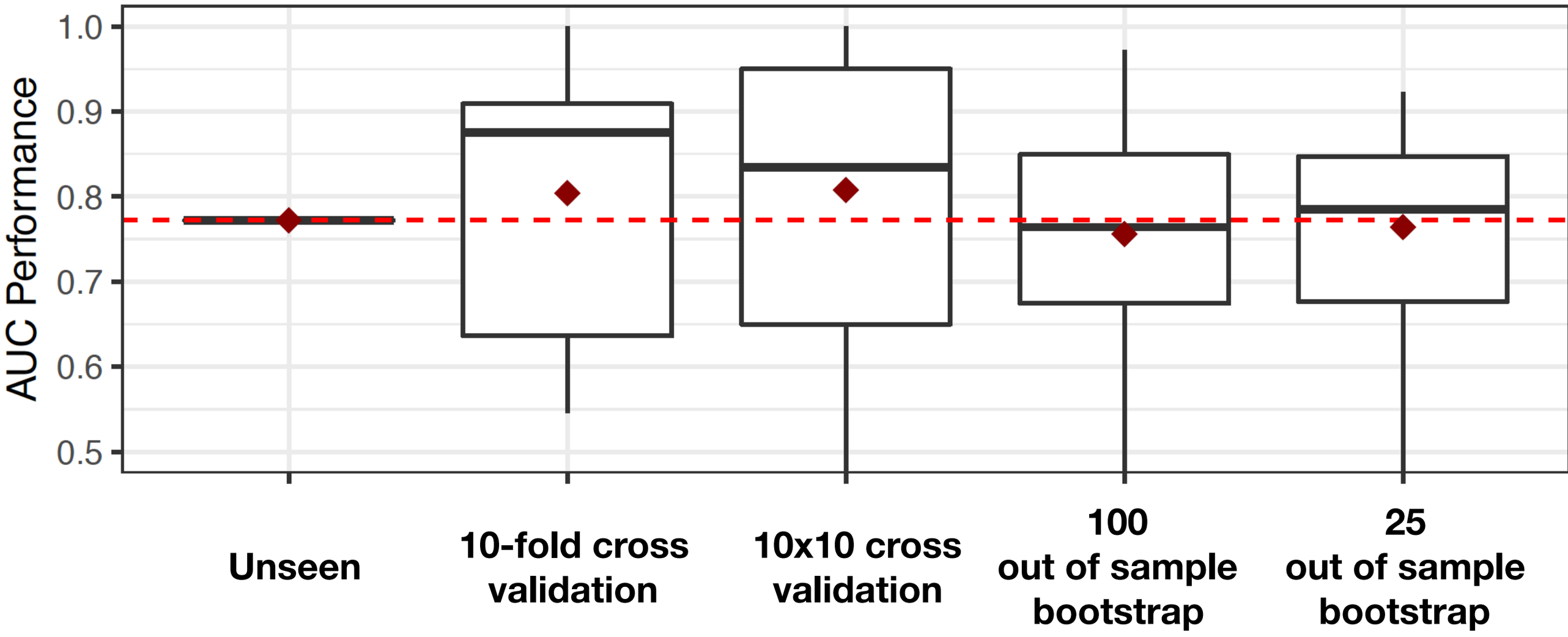
Using 10 fold cross validation



Using 10 fold cross validation



Using 10 fold cross validation



Using default ANOVA

Model M1: Bugs ~ f(TLOC, ...)

Model M2: Bugs ~ f(.....,TLOC)

Using default ANOVA

Model M1: Bugs ~ f(TLOC, …..)
Model M2: Bugs ~ f(……,TLOC)

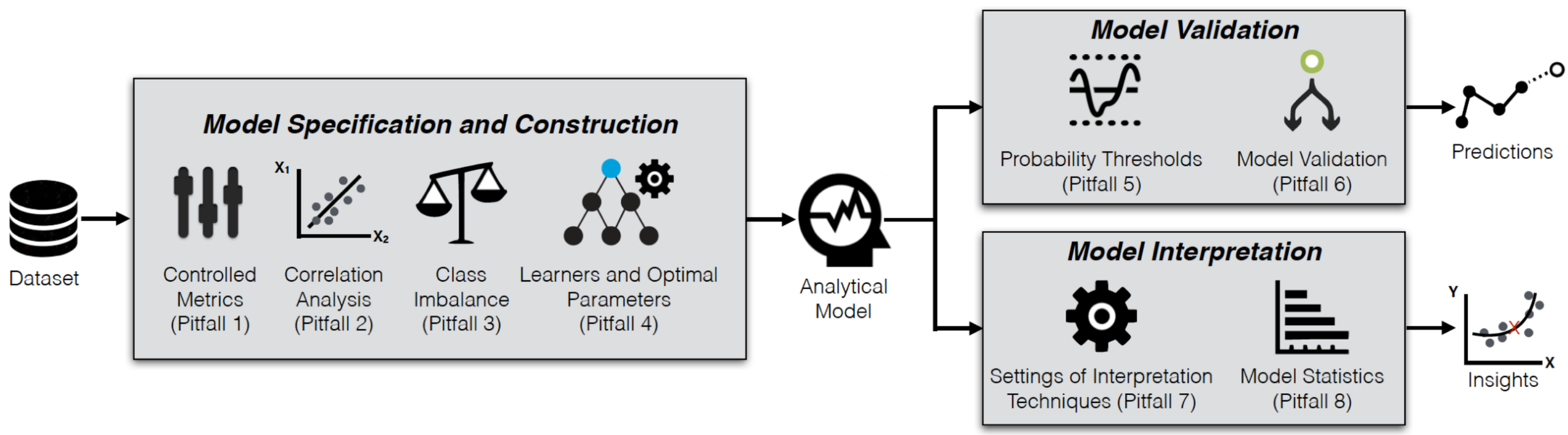
	M1		M2	
Metrics	Type 1	Type 2	Type 1	Type 2
TLOC	76%	22%	6%	22%
PAR_max	7%	21%	10%	21%
FOUT_max	8%	20%	28%	20%
NOI	5%	20%	6%	20%
NOF_max	4%	12%	5%	12%
NOM_max	0%	2%	33%	2%
ACD	0%	1%	5%	1%
NSF_max	0%	1%	5%	1%
NSM_max	0%	0%	2%	0%

Using default ANOVA

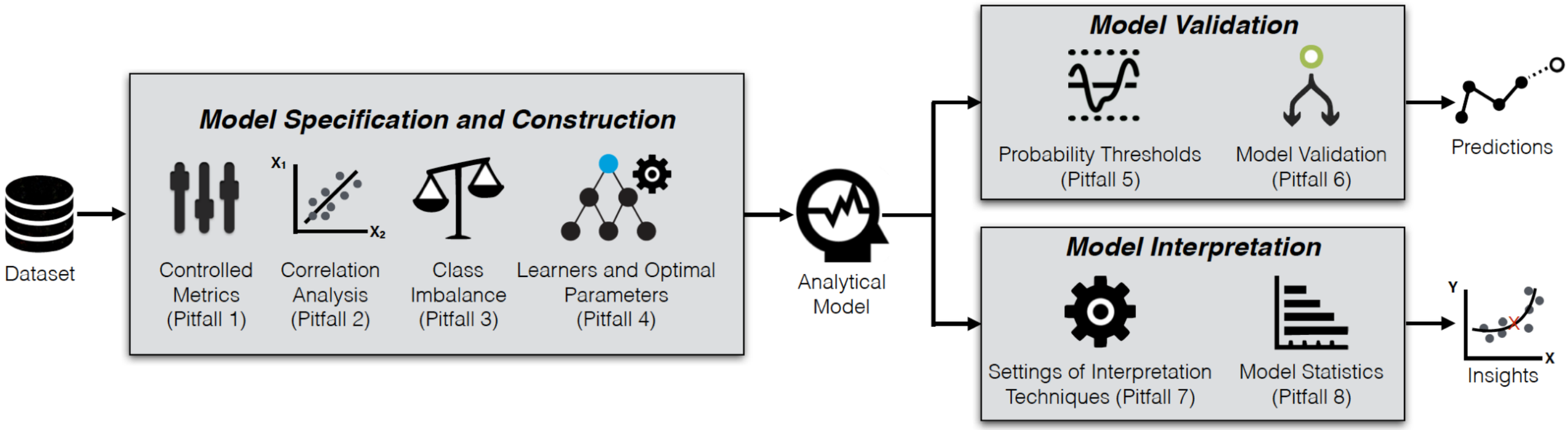
Model M1: Bugs ~ f(TLOC, …..)
Model M2: Bugs ~ f(……,TLOC)

	M1		M2	
Metrics	Type 1	Type 2	Type 1	Type 2
TLOC	76%	22%	6%	22%
PAR_max	7%	21%	10%	21%
FOUT_max	8%	20%	28%	20%
NOI	5%	20%	6%	20%
NOF_max	4%	12%	5%	12%
NOM_max	0%	2%	33%	2%
ACD	0%	1%	5%	1%
NSF_max	0%	1%	5%	1%
NSM_max	0%	0%	2%	0%

Pitfalls in Analytical Modelling

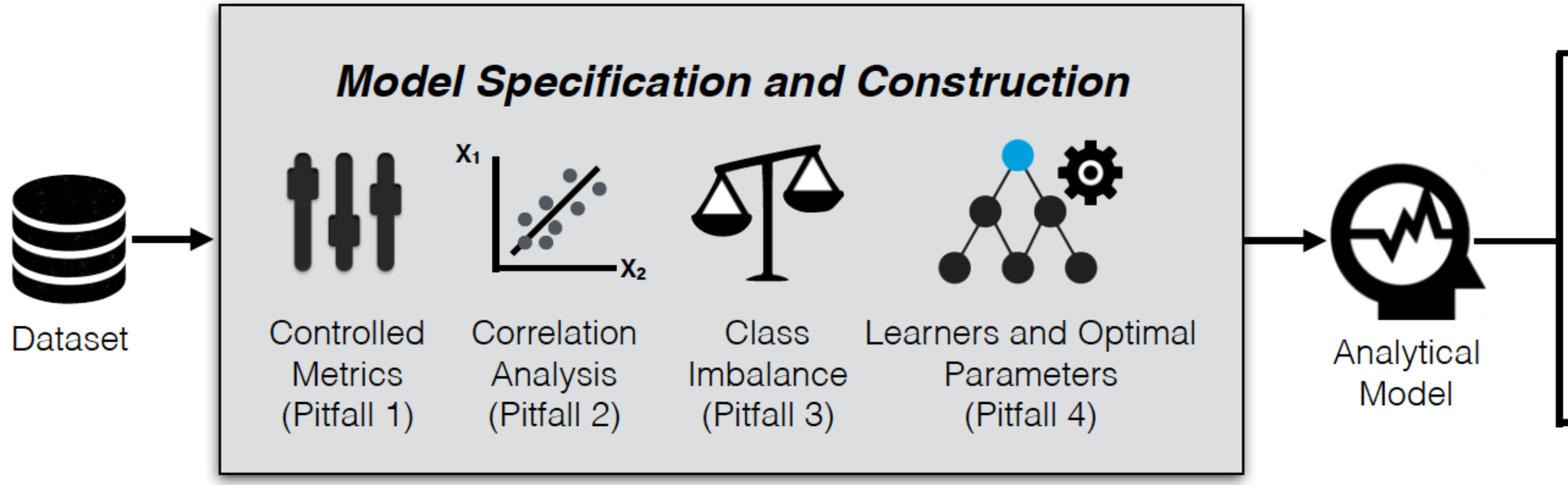


Pitfalls in Analytical Modelling

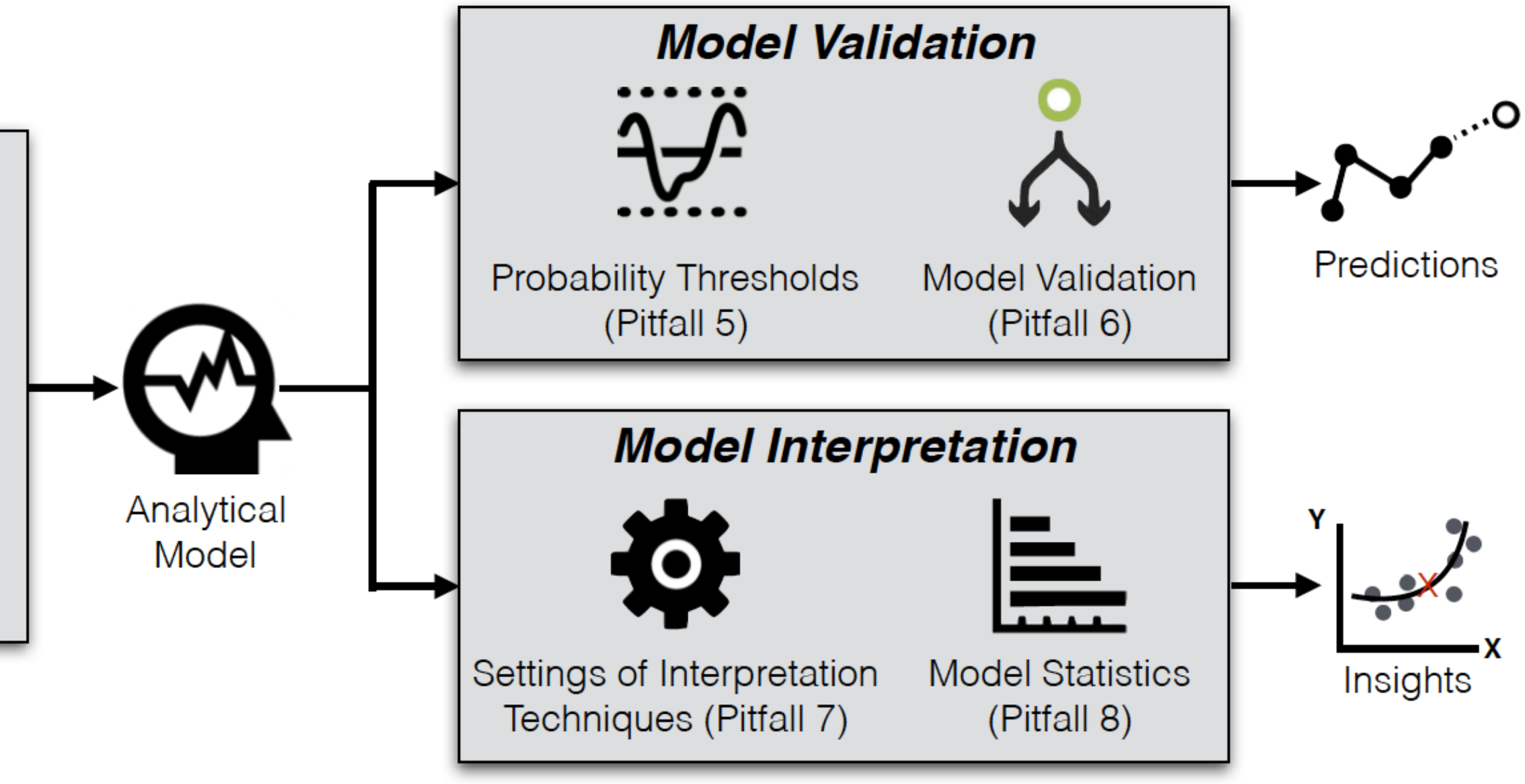


Joint work with
Kla Tantihamthavorn

Pitfalls in Analytical Modelling



Pitfalls in Analytical Modelling



Open Scripts

Open Scripts

Encourages community mentorship

Open Scripts

Works even for industrial data

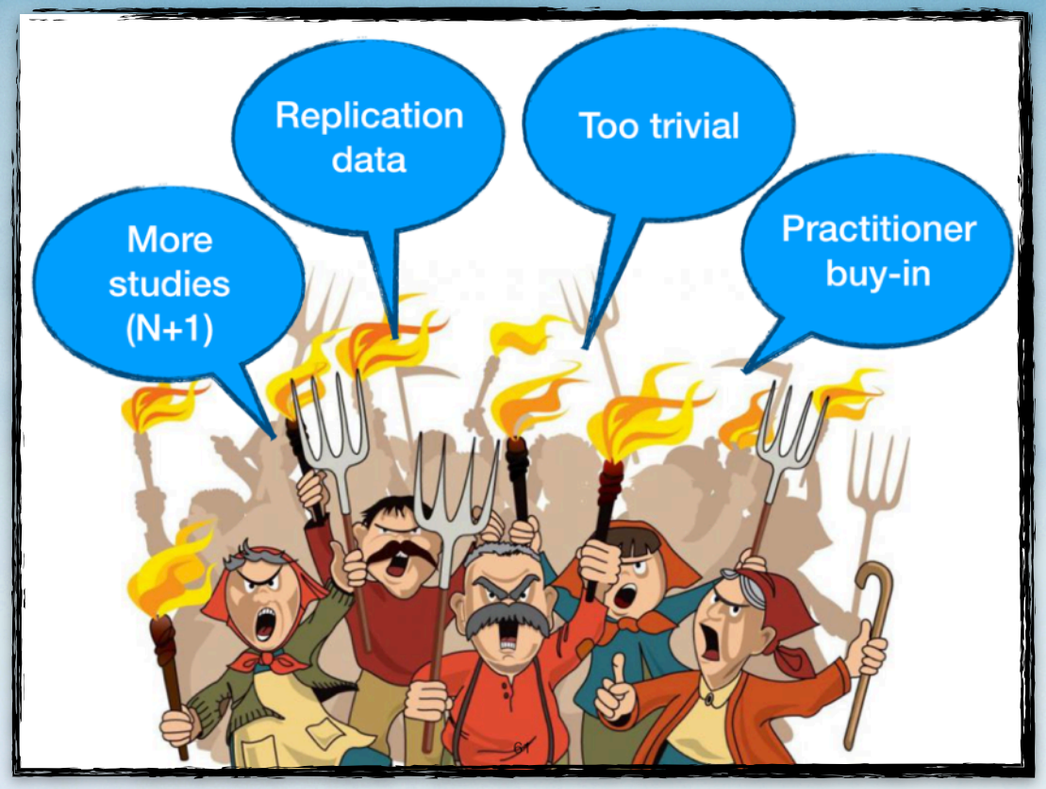
Encourages community mentorship

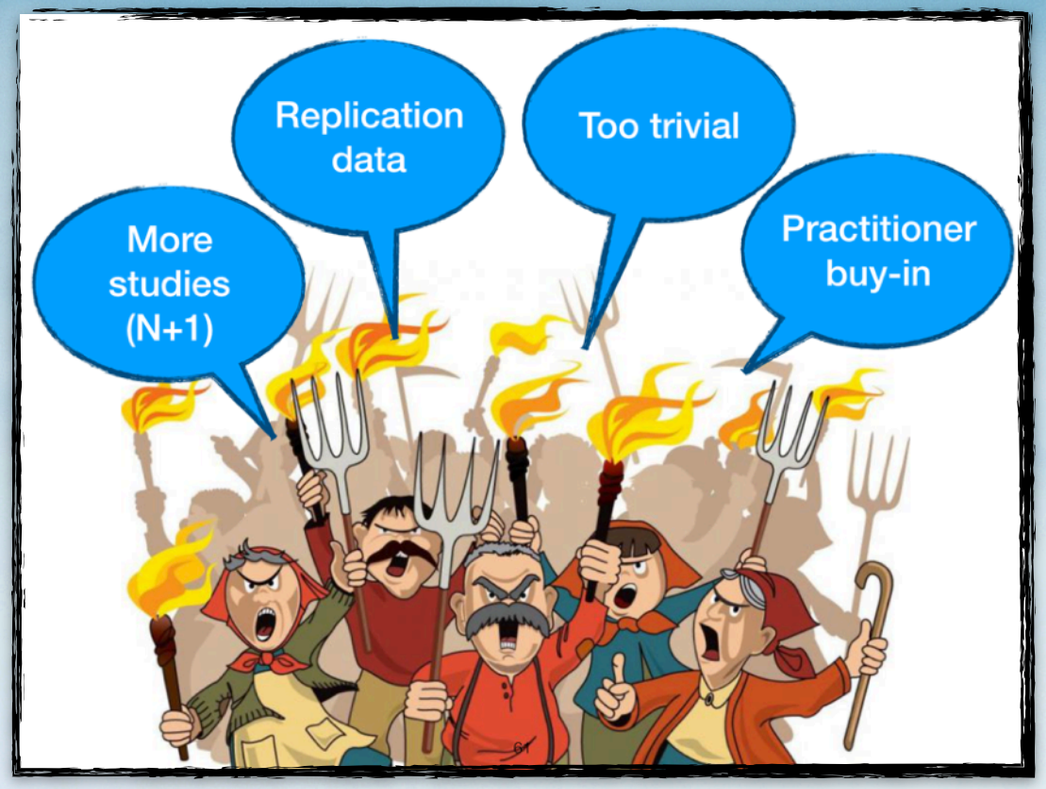
Open Scripts

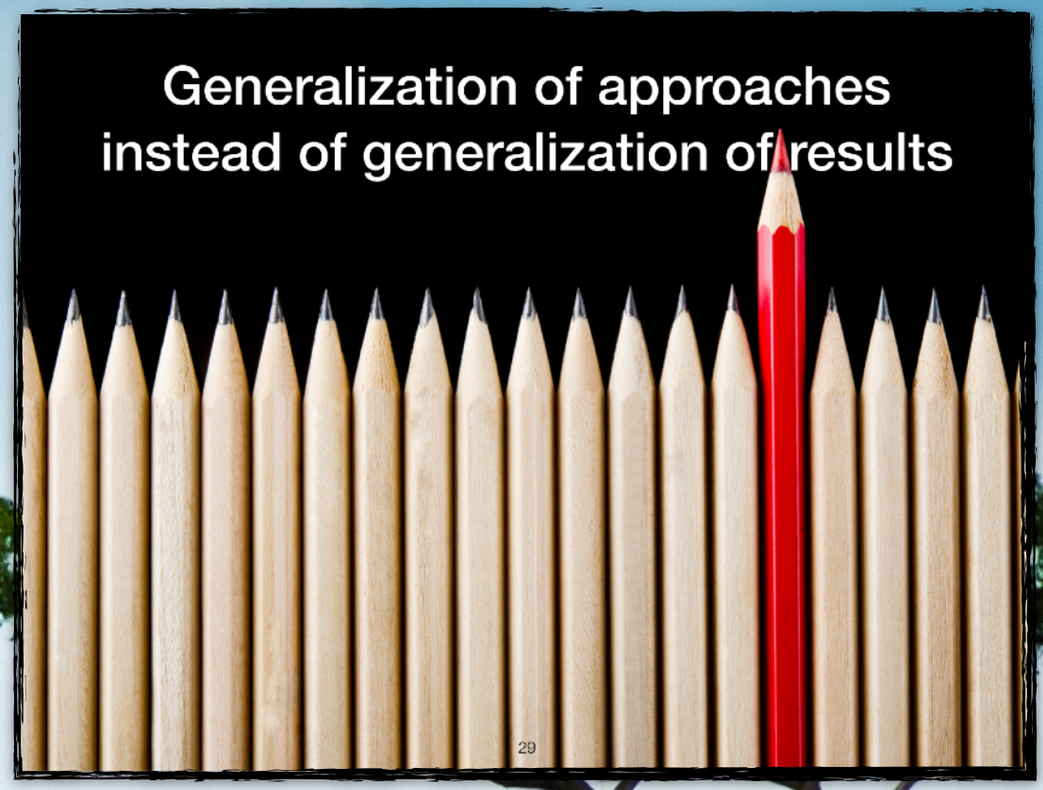
Works even for industrial data

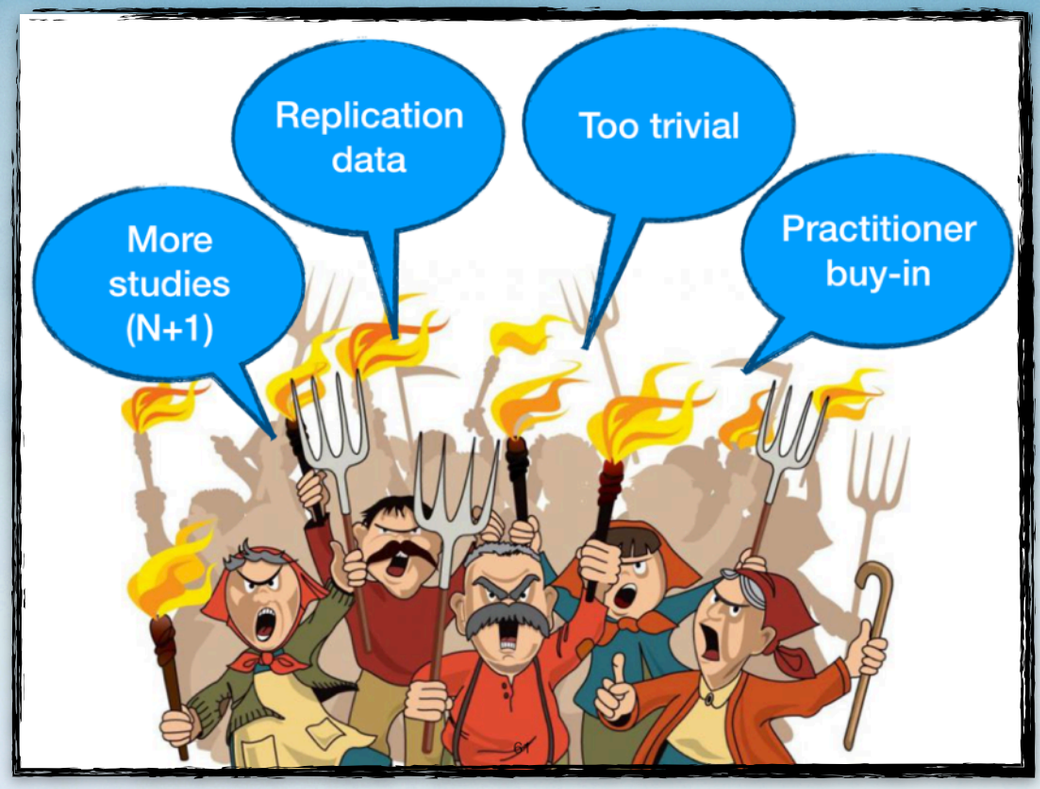
Encourages community mentorship

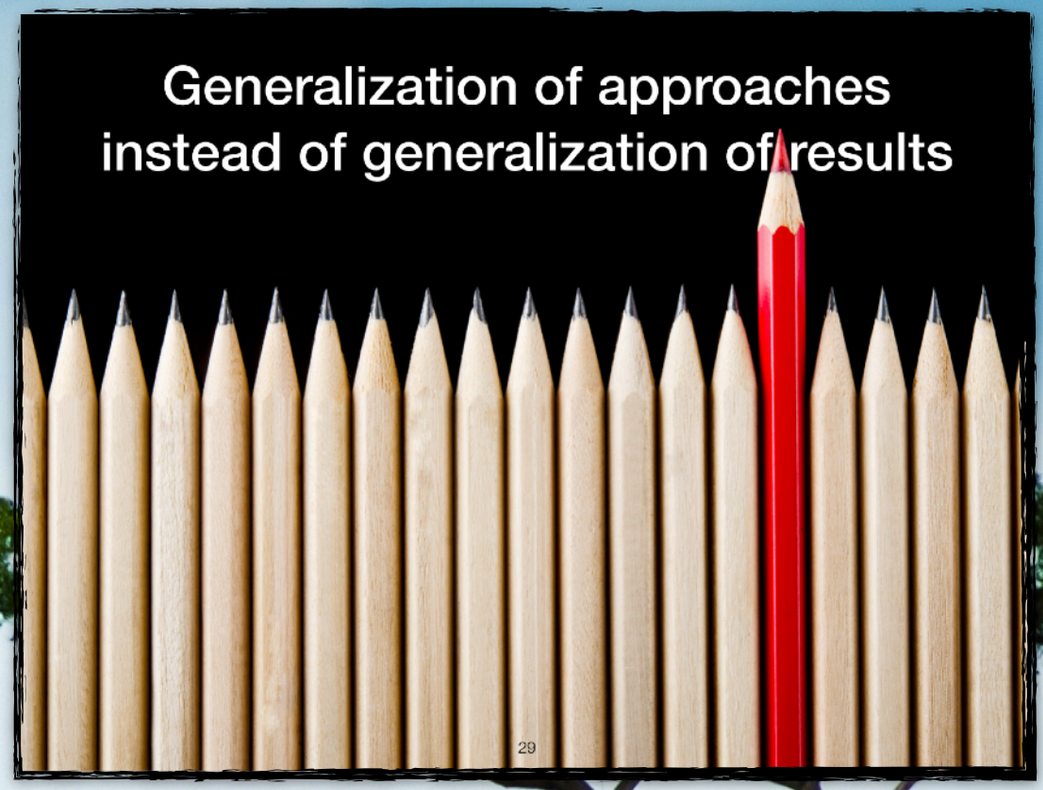
Ensures research transparency


Replication data
Too trivial
More studies (N+1)
Practitioner buy-in


Replication data
Too trivial
More studies (N+1)
Practitioner buy-in


Generalization of approaches instead of generalization of results
29





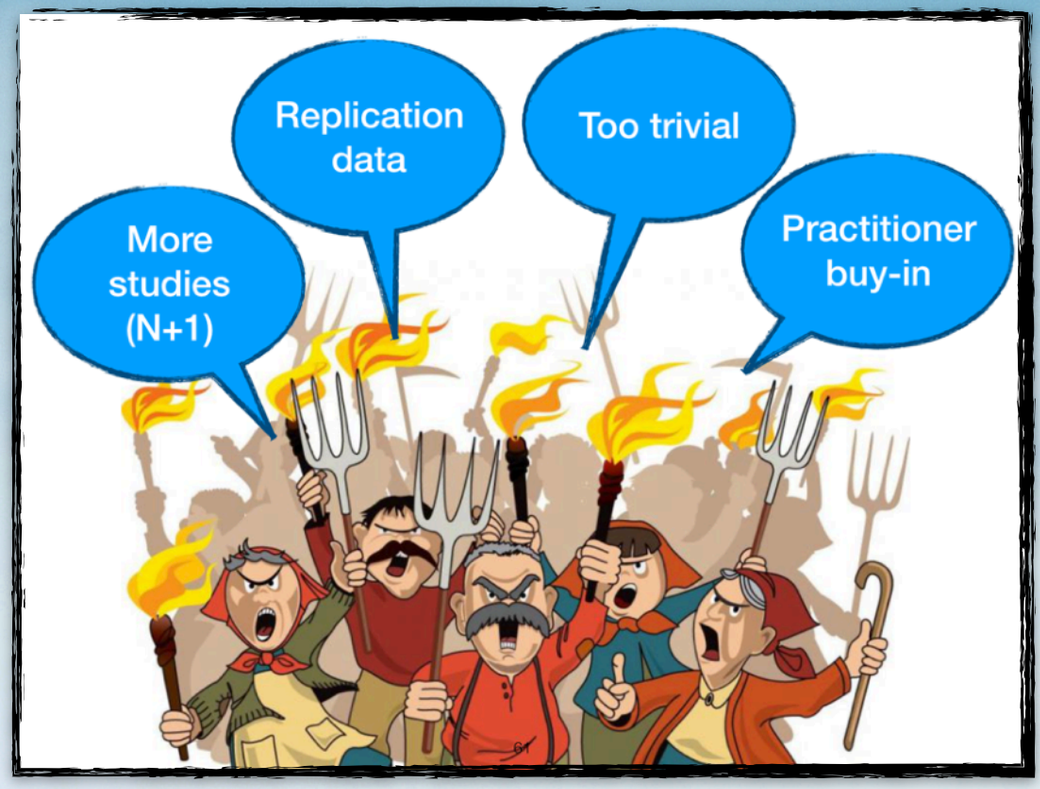
Open Scripts

Works even for industrial data

Encourages community mentorship

Ensures research transparency

38



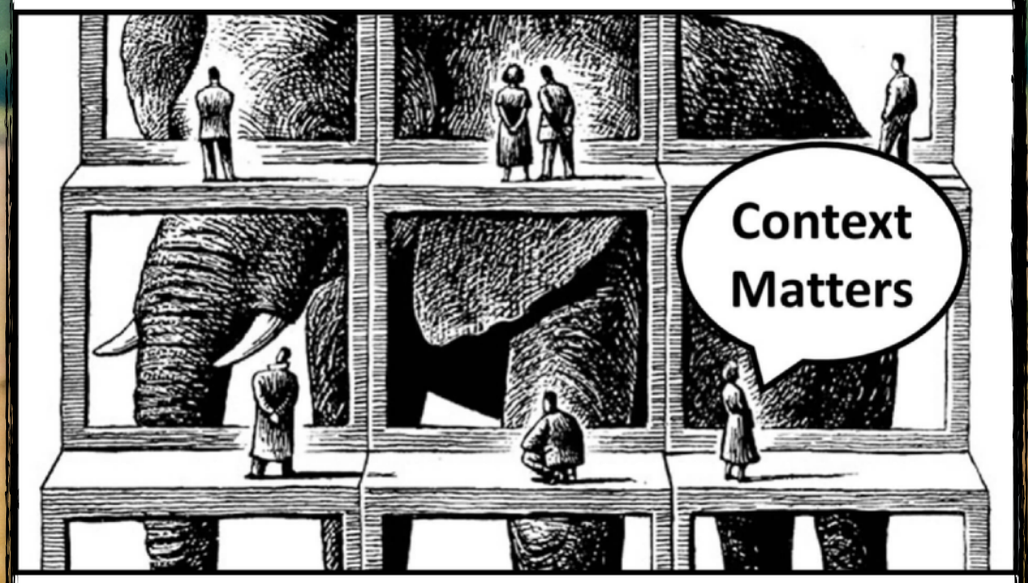
Generalization of approaches instead of generalization of results

Open Scripts

Works even for industrial data

Encourages community mentorship

Ensures research transparency



Where do we go from here?

Parting thoughts

- Trailblazing research should be encouraged and defended
- Inaccurate analysis is a serious and growing threat
- Deeper analysis is more valuable than large scale studies
- Generalization is not feasible nor desired by practitioners
- Industry is a partner not an idol

http://sail.cs.queensu.ca

Think Different!