RSAConference2020
Asia Pacific & Japan

A Virtual Learning Experience | 15–17 July

HUMAN ELEMENT

SESSION ID: DPS-W02

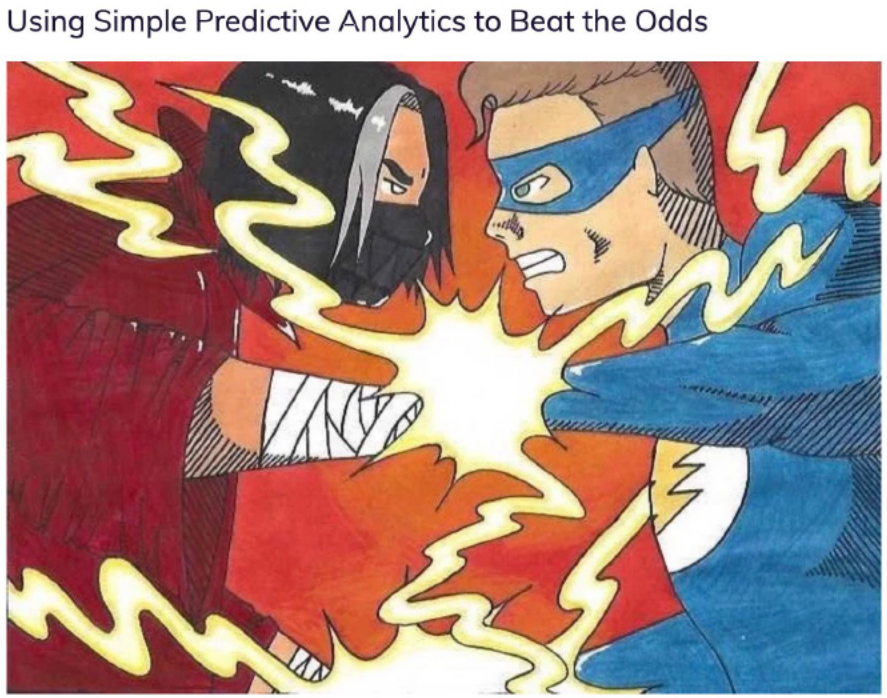
Security Outgunned: Measuring Software Defined Attack Surface

Richard Seiersen

CEO / Author
www.soluble.ai
@richardseiersen

A tutorial for this talk (and code) can be found here:

https://www.soluble.ai/blog/three-steps-to-getting-better-security-roi



RSAConference2020 APJ
A Virtual Learning Experience

Part 1: I Believe Therefore I Measure

An Approach To Reducing Attack Surface By Reducing Errors

I Believe Therefore I Measure

> "We are prone to overestimate how much we understand about the world and to underestimate the role of chance in events."
> — Daniel Kahneman, Thinking, Fast and Slow

I Believe Therefore I Measure

You're outgunned by developers 100:1

You have little time, little data and little budget

Your first step – reduce external facing vulnerabilities

That means getting the solutions with the least errors and the least cost

I Believe Therefore I Measure

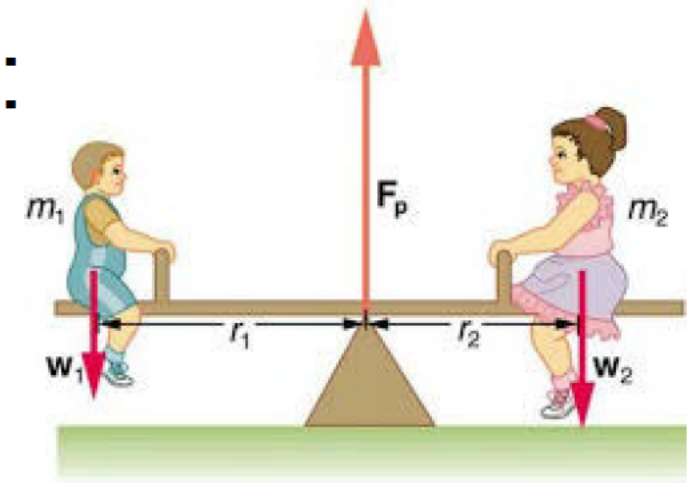
Our first step is to **naively measure** your beliefs about errors rates

To **quantify your beliefs**, I will require **two probabilities** from you

The first is the **median error rate**. The second is the **90% boundary rate**...

I Believe Therefore I Measure

For example, a practitioner might say:

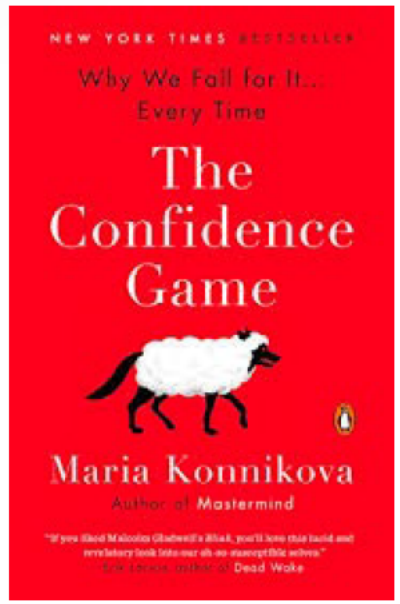


*Our median error rate is around **25%**.*

*That means I believe the true error rate **is just as likely** to be **below** 25% as it is to be **above**.*

I Believe Therefore I Measure

"In terms of my 90% boundary rate. I'm 90% confident the true rate is below 45%."

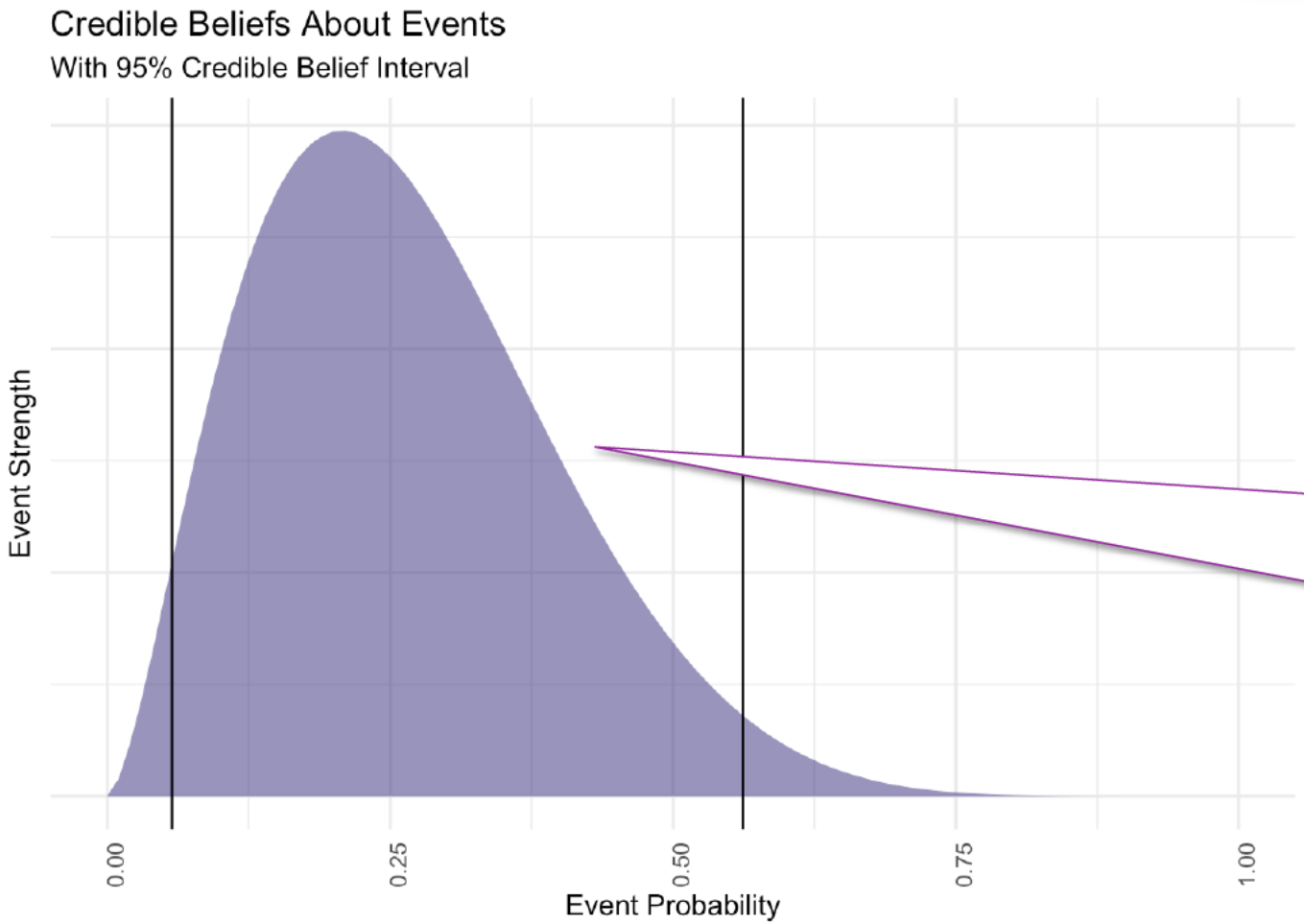


"After all, I have never in my life experienced a 50% error rate – but it might happen!"

I Believe Therefore I Measure

```
# Beliefs about error counts prior to testing
event_priors <- list()
event_priors["median_errors"] <- 0.25  # True rate is just as likely above/below this
event_priors["edge_errors"] <- 0.45    # 90% sure true value is below this
```

I Believe Therefore I Measure



If this graph could speak it would say,

"Given your beliefs you should expect the true rate is likely between 6% and 56%."

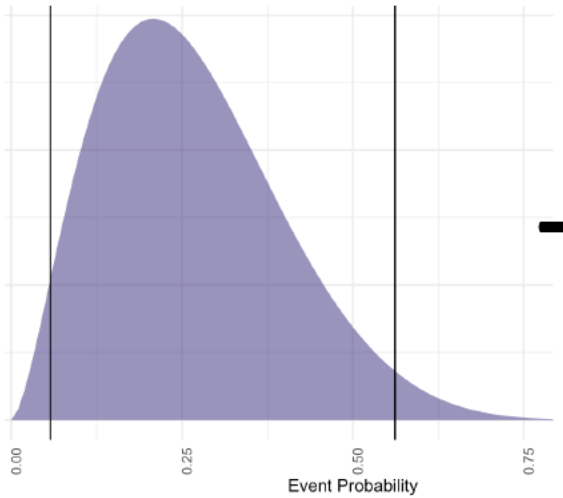
Soluble

RSAConference2020 APJ
A Virtual Learning Experience

Part 2: Munging Beliefs With Data

An Approach To Reducing Attack Surface By Reducing Errors

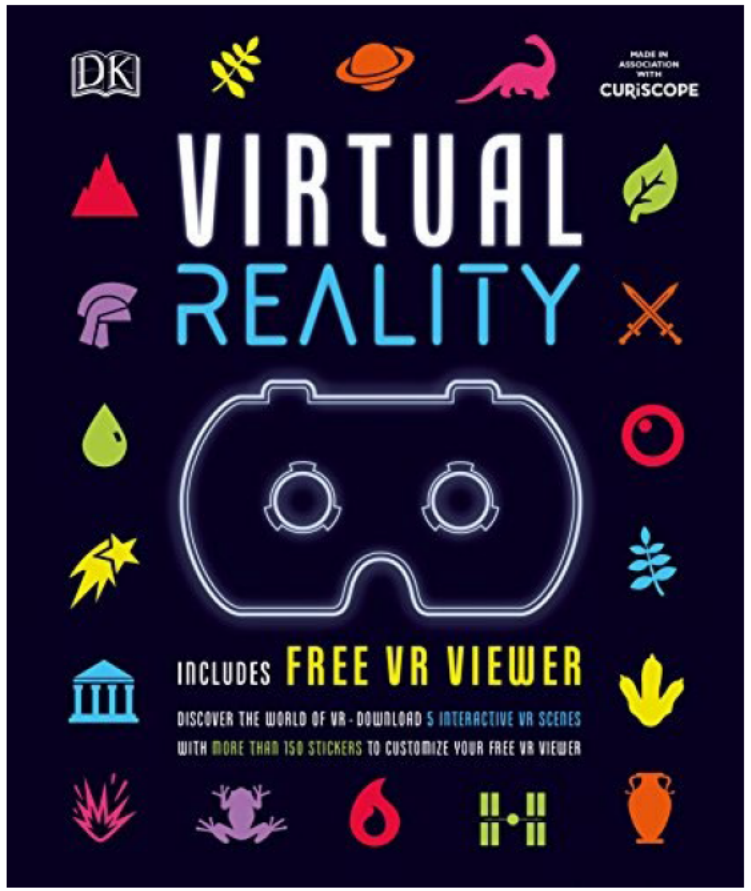
Munging Beliefs With Data



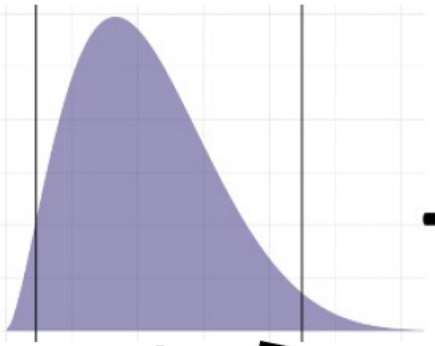
+

```
base_configs <- list()
base_configs["n_draws"] <- 100000
base_configs["a_n_events"] <- 30
base_configs["b_n_events"] <- 37
```

=



Munging Beliefs With Data



```
base_configs <- list()
base_configs["n_draws"] <- 100000
base_configs["a_n_events"] <- 30
base_configs["b_n_events"] <- 37
```

	proportion_events	a_n_errors
1	0.16024930	4
2	0.37131587	10
3	0.16081836	4
4	0.20387551	5
5	0.30250041	5
6	0.04250996	1

	proportion_events	b_n_errors
1	0.16024930	6
2	0.37131587	11
3	0.16081836	4
4	0.20387551	11
5	0.30250041	15
6	0.04250996	1

Munging Beliefs With Data

```
# What was the real count after testing. it's in relation to base_configs["n_events"] above.
event_counts <- list()
event_counts["product_a_errors"] <- 10  # After conducting test, what are the real errors for prod a
event_counts["product_b_errors"] <- 13  # After conducting test, what are the real errors for prod b
```

	proportion_events	a_n_errors
25	0.4345537	10
69	0.3229338	10
75	0.3302515	10
100	0.3975704	10
153	0.2054331	10
154	0.4420882	10

	proportion_events	b_n_errors
55	0.3641264	13
59	0.2843811	13
86	0.4854838	13
98	0.2627091	13
135	0.4033498	13
141	0.2534997	13

Approximate
Bayesian
Computation

Soluble

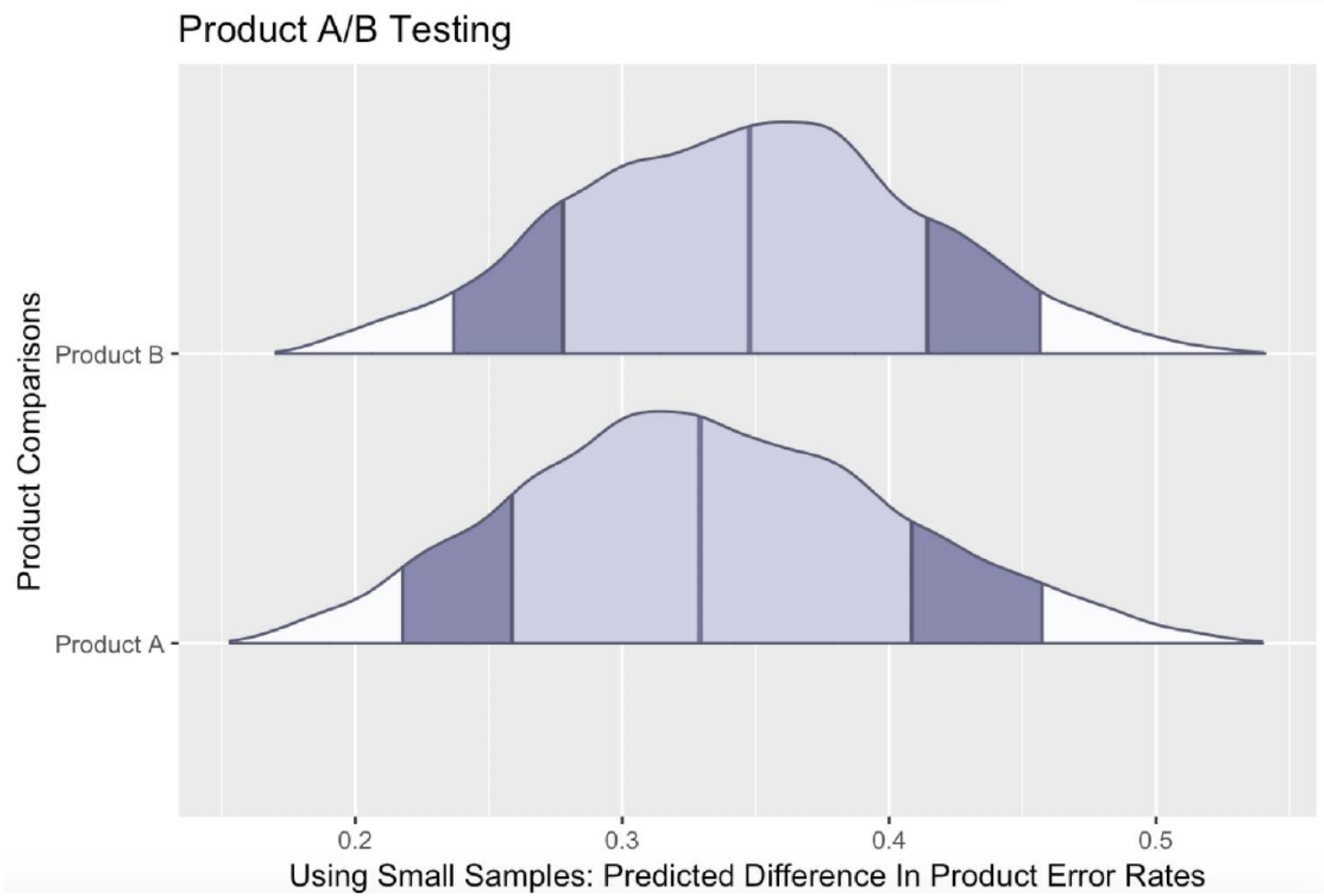
Munging Beliefs With Data

	prod_a	prod_b
1	0.3537220	0.4135234
2	0.3145357	0.3726896
3	0.3561745	0.2975243
4	0.3150756	0.3379805
5	0.2614126	0.2743952
6	0.4905241	0.4176402

Product A SME forecast combined with **30 vulnerabilities** and **10 errors** assumes the real error rate is between: **22%** and **46%** And likely centered near **33%**

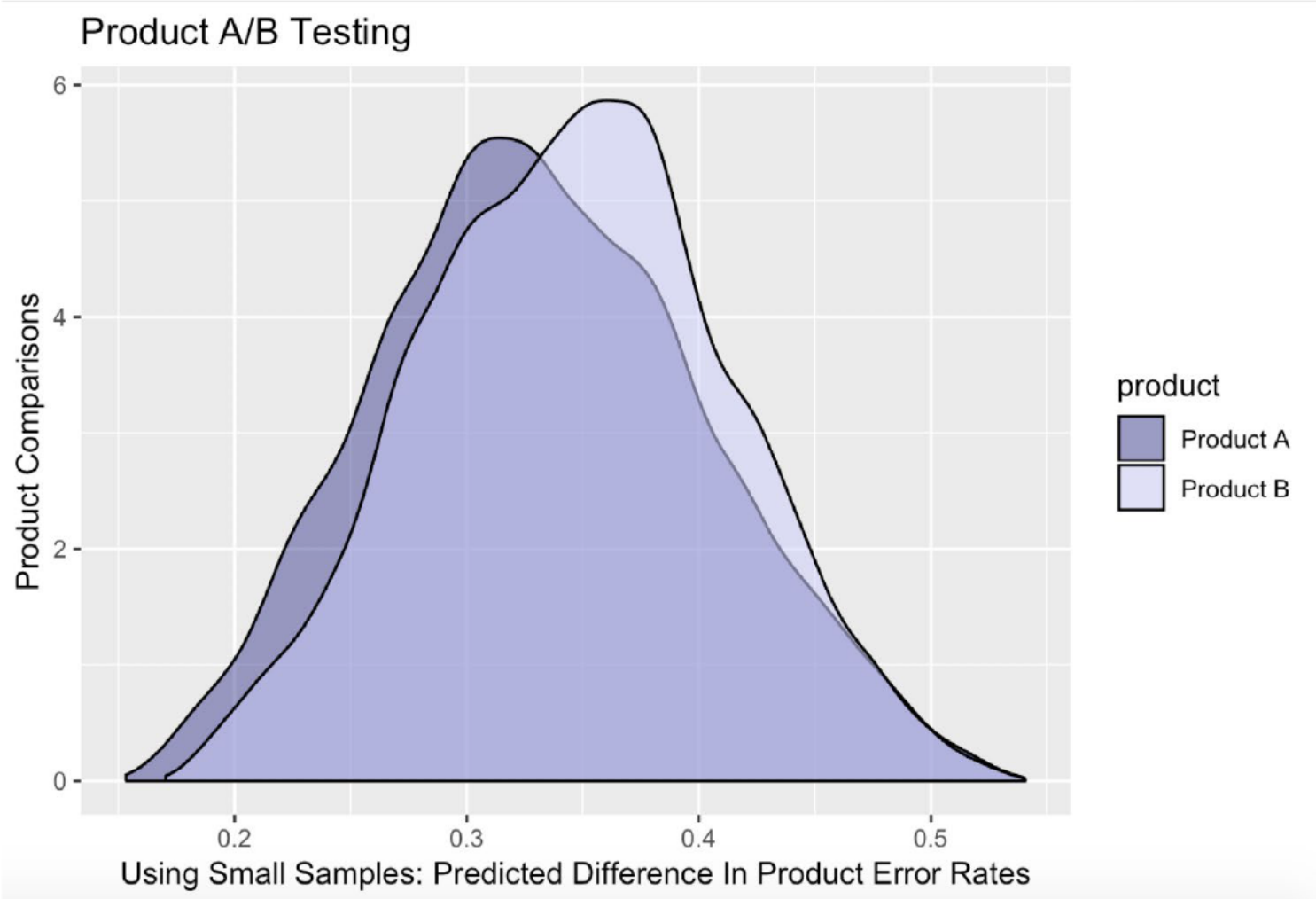
Product B forecast combined with **37 vulnerabilities** and **13 errors** assumes the real error rate is between: **24%** and **46%** And likely centered near **34%**

Munging Beliefs With Data



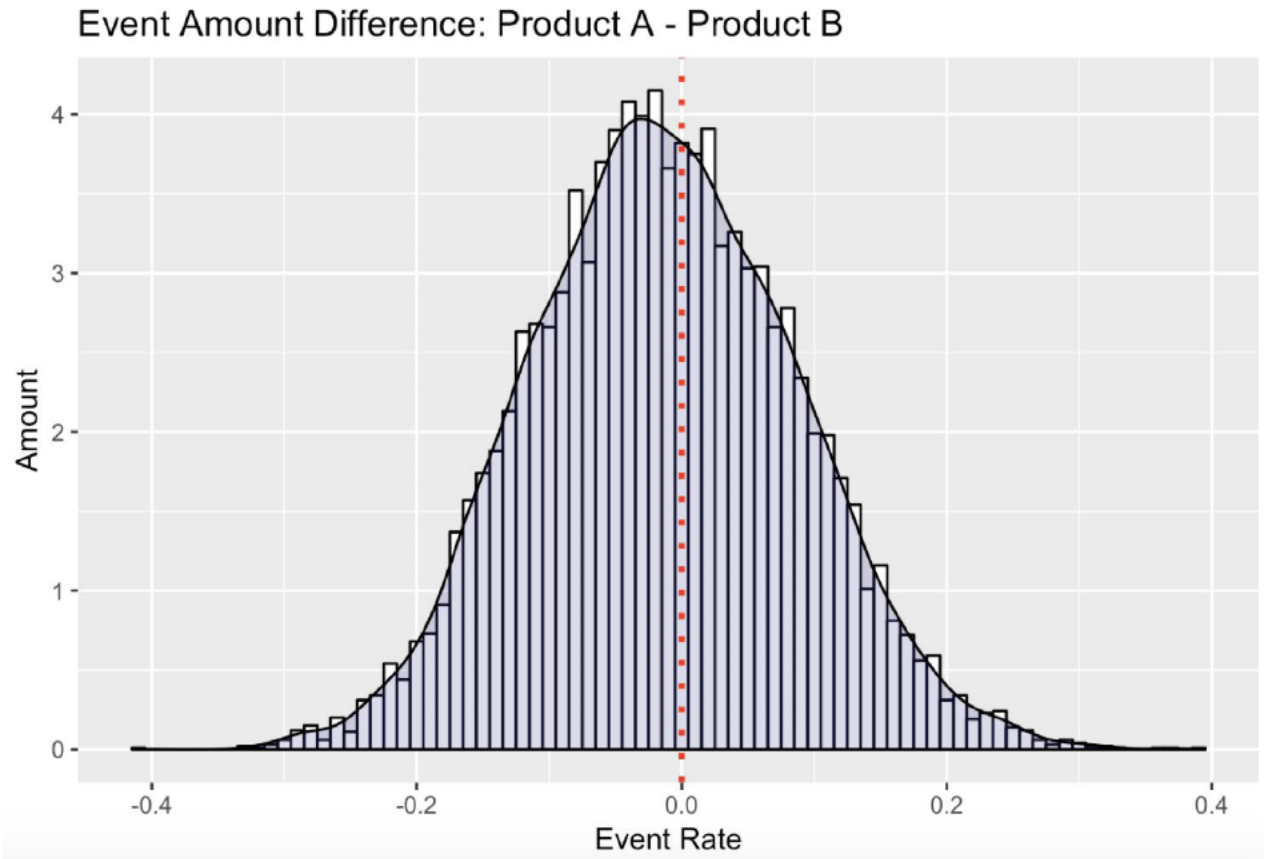
Soluble

Munging Beliefs With Data



Soluble

Munging Beliefs With Data



prod_a	prod_b	prop_diff_events
0.5430354	0.2332028	0.30983260
0.4679103	0.3289690	0.13894132
0.1478495	0.3153235	-0.16747396
0.2982768	0.4216337	-0.12335695
0.3549745	0.3394246	0.01554989
0.2689235	0.2081265	0.06079702
0.3241985	0.3532960	-0.02909743
0.2771185	0.4076813	-0.13056275
0.4246763	0.4519786	-0.02730230
0.2630322	0.3712558	-0.10822356

RSAConference2020 APJ

A Virtual Learning Experience

Part 3: The Financial Impact Of Errors

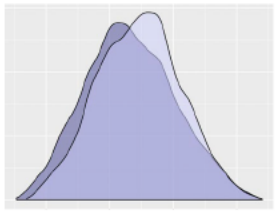
An Approach To Reducing Attack Surface By Reducing Errors

The Financial Impact Of Errors

```
# Forecasted range of error response hours that can happen per event
eng_hour_range <- list()
eng_hour_range["mode"] <- 2  # Expect hours
eng_hour_range["low"] <- 1   # Low hours
eng_hour_range["high"] <- 5 # Max hours
```

```
# Forecasted cost of responding to an error event for ONE HOUR
eng_cost_range <- list()
eng_cost_range["mode"] <- 600  # Expected cost per event
eng_cost_range["low"] <- 200    # Low end cost per event
eng_cost_range["high"] <- 2000  # High end cost per event
```

The Financial Impact Of Errors



X

```
eng_hour_range["mode"] <- 2
eng_hour_range["low"] <- 1
eng_hour_range["high"] <- 5
```

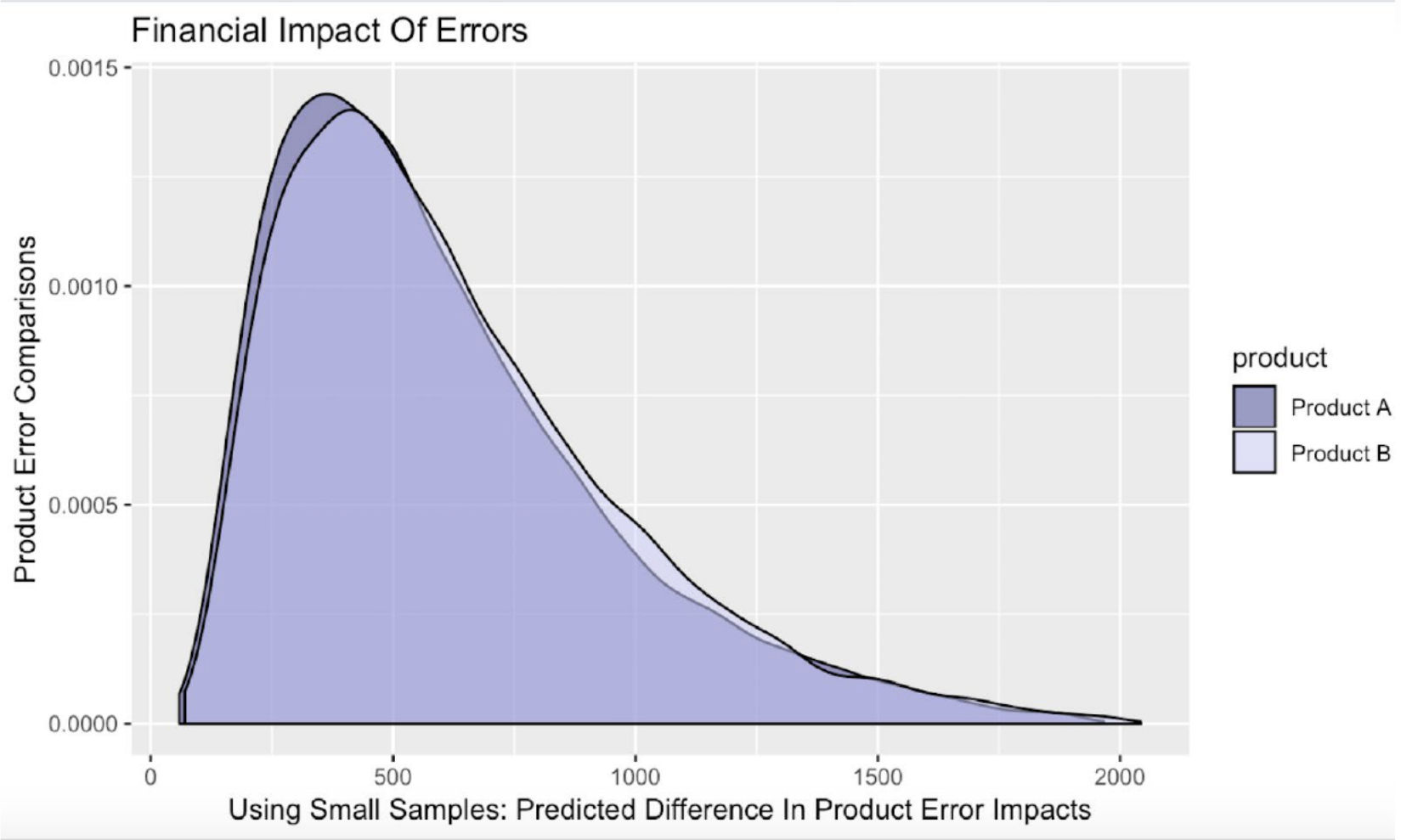
X

```
eng_cost_range["mode"] <- 600
eng_cost_range["low"] <- 200
eng_cost_range["high"] <- 2000
```

	prod_a	prod_b
1	0.4258503	0.4173965
2	0.3910038	0.4240344
3	0.2403929	0.2785301
4	0.3431881	0.5064512
5	0.3103128	0.2710452
6	0.2525325	0.4417854

prod_a_impact	prod_b_impact
950.9384	932.0608
402.8597	436.8919
723.2961	838.0436
756.7992	1116.8273
570.2767	498.1128
543.1028	950.1148

The Financial Impact Of Errors



The Financial Impact Of Errors

Scenario 1: 1 Month

- Product A is 90.8% the total cost of product B.
- Product A (with base price of $72,000) is expected to cost to operate with errors: $148,920 a year.
- Product B (with base price of $65,000) is expected to cost to operate with errors: $163,947.

The Financial Impact Of Errors

Scenario 2: 1 Week

- Product A is 81.1% the total cost of product B
- The expected operation cost of Product A with errors(with base price of $72,000): $372,776
- The expected operation cost of Product B with errors(with base price of $65,000): $459,551

The Financial Impact Of Errors

Scenario 3: 1 Day

- Product A is 76.2% the total cost of product B
- The expected operation cost of Product A with errors(with base price of $72,000): $2,113,258
- The expected operation cost of Product B with errors (with base price of $65,000): $2,772,064

Applications For Measuring Software Defined Attack Surface

- First step, measure your beliefs
- Second step, measure actual errors and munge with beliefs
- Third step, measure costs and munge with error rates
- Consider selecting the solution with the overall least cost.
- **https://www.soluble.ai/blog/three-steps-to-getting-better-security-roi**

RSAConference2020 APJ

A Virtual Learning Experience

Thank you

rich@soluble.ai

@richard.seiersen