Client-Side Multivariate Testing

Howard Abrams
@howardabrams

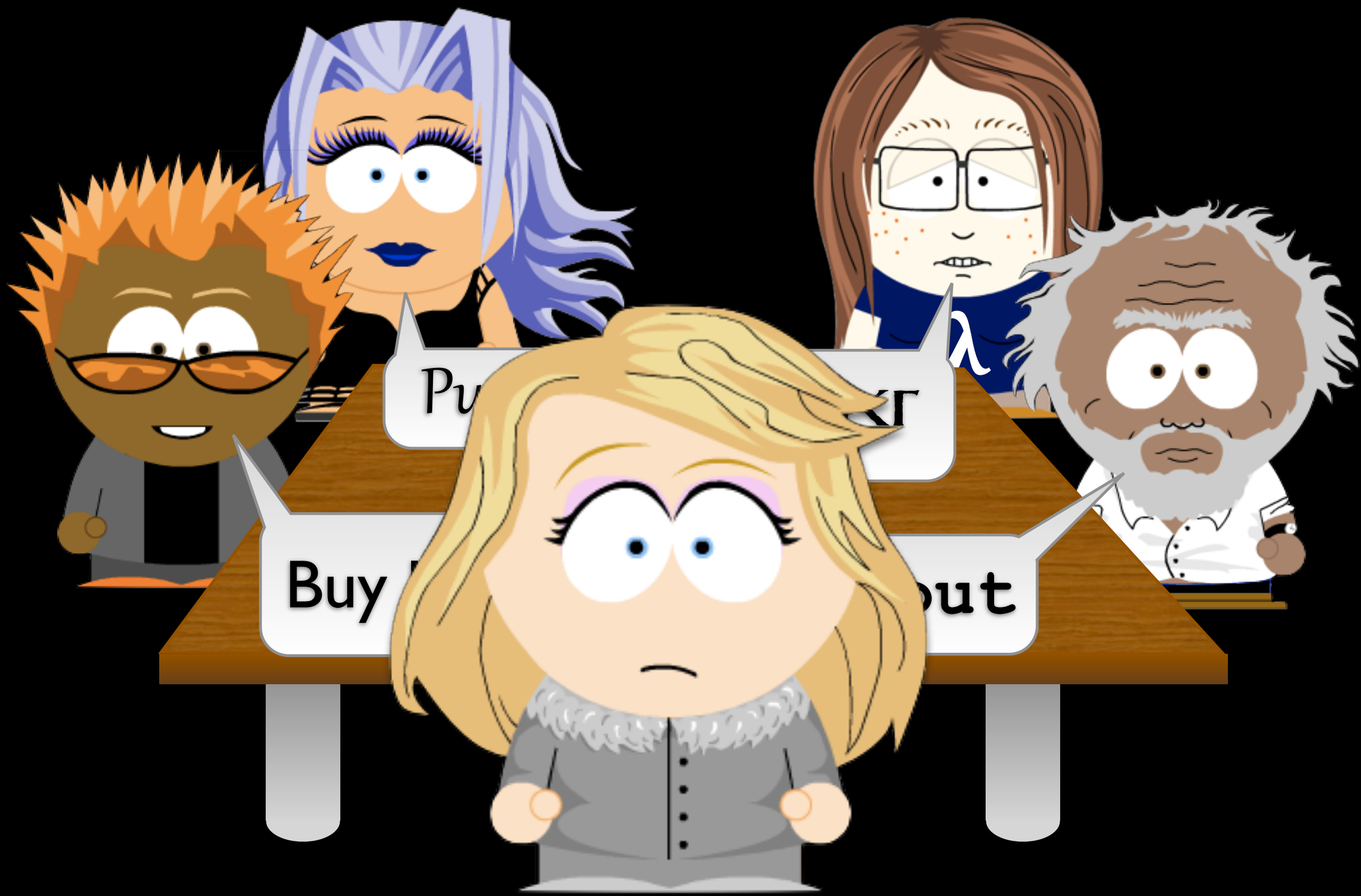
http://github.com/howardabrams/labrats

Client-Side Multivariate Testing

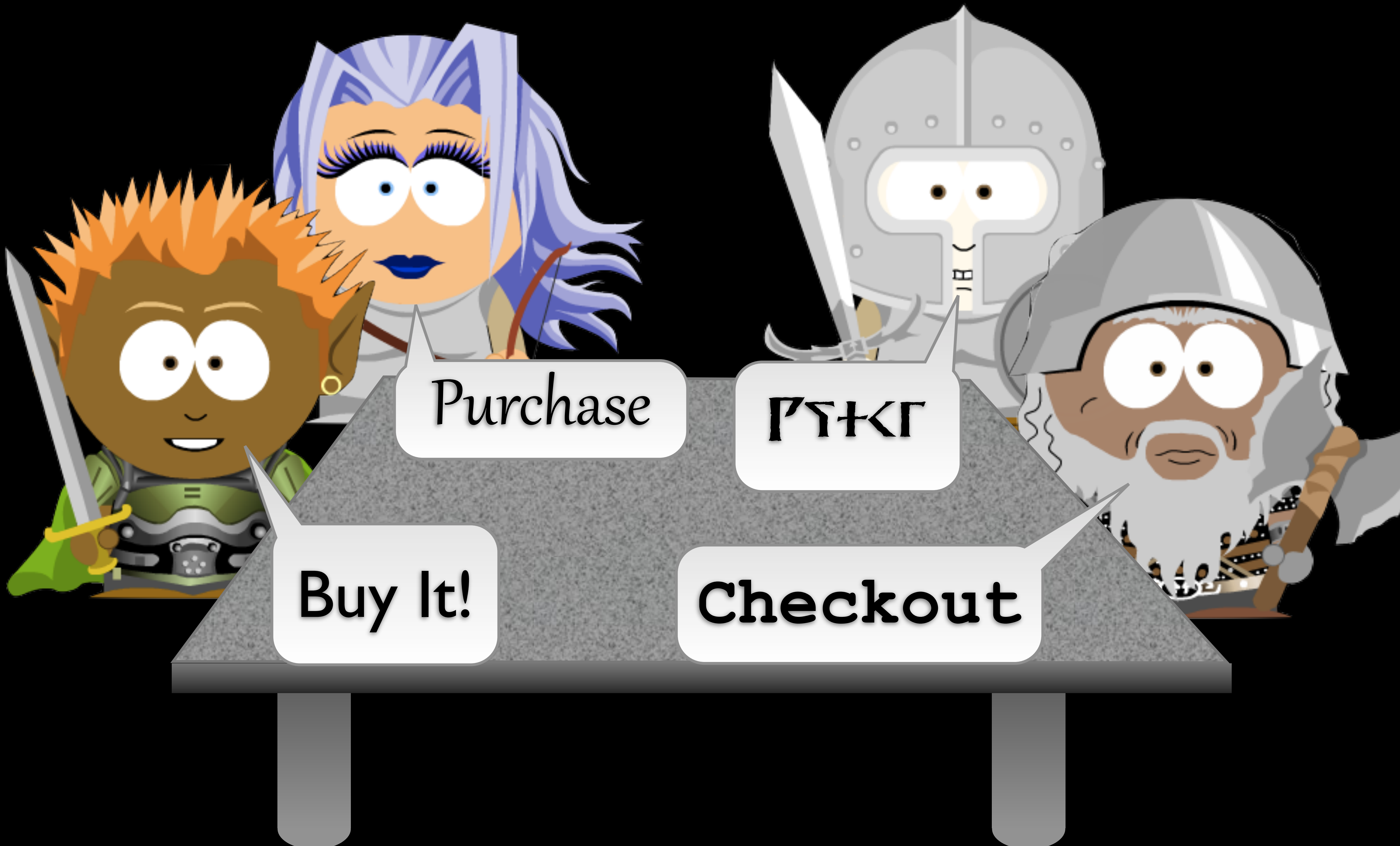
Experimenting on your Customers like Lab Rats for Fun & Profit


λ


Purchase
Buy It!
Checkout


Pu
λ
Buy
out

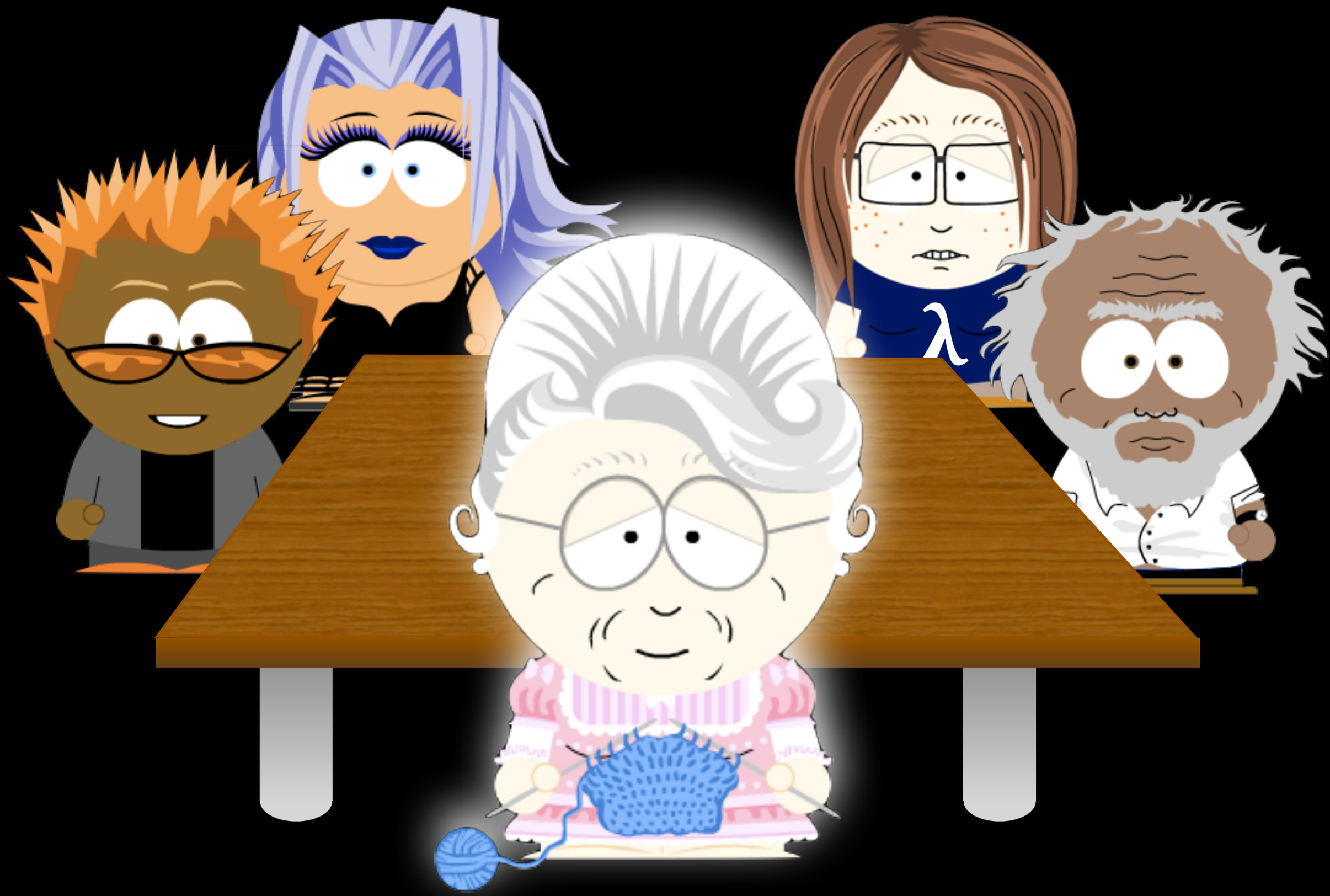

Purchase
Buy It!
Checkout


Buy It!
Purchase
Checkout


λ


λ
Buy Now!
Checkout


λ


λ

Win A Free Las Vegas Vac

Take our survey and be entered
to win a Las Vegas Vacation
Package on us

- Round-trip airfa
- Hotel stay at th
- $400 Restaurar

Help Us Help You!

We need your feedback on enhancing your
Money Progra 09.

Click N

survey

NO
n it at my office

Please take
a few
minutes to
give us your
opinion.
PLEASE!

Take Our F
Site Survey
Win a $500

willing to complete a brief (5 minute) ques-
tionnaire?

It's easy to participate:

Click the "Yes" button below.

Yes, I will!

No thanks

Win A Free

Take our survey and
to win a Las Vegas
Package on us

- Round-trip airfa
- Hotel stay at th
- $400 Restaura

TAKE OUR SURVEY & RECEIVE A

FREE BAG!

Help Us Help You!

feedback on enhancing your

ey

NO ▸
it at my office

a few
minutes to
give us your
opinion.
PLEASE!

Take Our
Site Survey
Win a $500

willing to complete a brief (3 minutes) ques-
tionnaire?

It's easy to participate:

Click the "Yes" button below.

Yes, I will!

No thanks

	Very Strongly Agree	Strongly Agree	Agree	Disagree	Strongly Disagree	Very Strongly Disagree
Stores are conveniently located.	○	○	○	○	○	○
Store hours are convenient for my shopping needs.	○	○	○	○	○	○
Store atmosphere and decor are appealing.	○	○	○	○	○	○
A good selection of products was present.	○	○	○	○	○	○
(Store) has the lowest prices in the area.	○	○	○	○	○	○
Merchandise sold is of the highest quality.	○	○	○	○	○	○
The merchandise sold is a good value for the money.	○	○	○	○	○	○
Merchandise displays are attractive.	○	○	○	○	○	○
Advertised merchandise was in stock.	○	○	○	○	○	○
Overall, I am very satisfied with the store.	○	○	○	○	○	○
I am very satisfied with the price I paid for what I bought.	○	○	○	○	○	○
I am very satisfied with the merchandise I bought.	○	○	○	○	○	○

Big Data Borat
@BigDataBorat

Follow

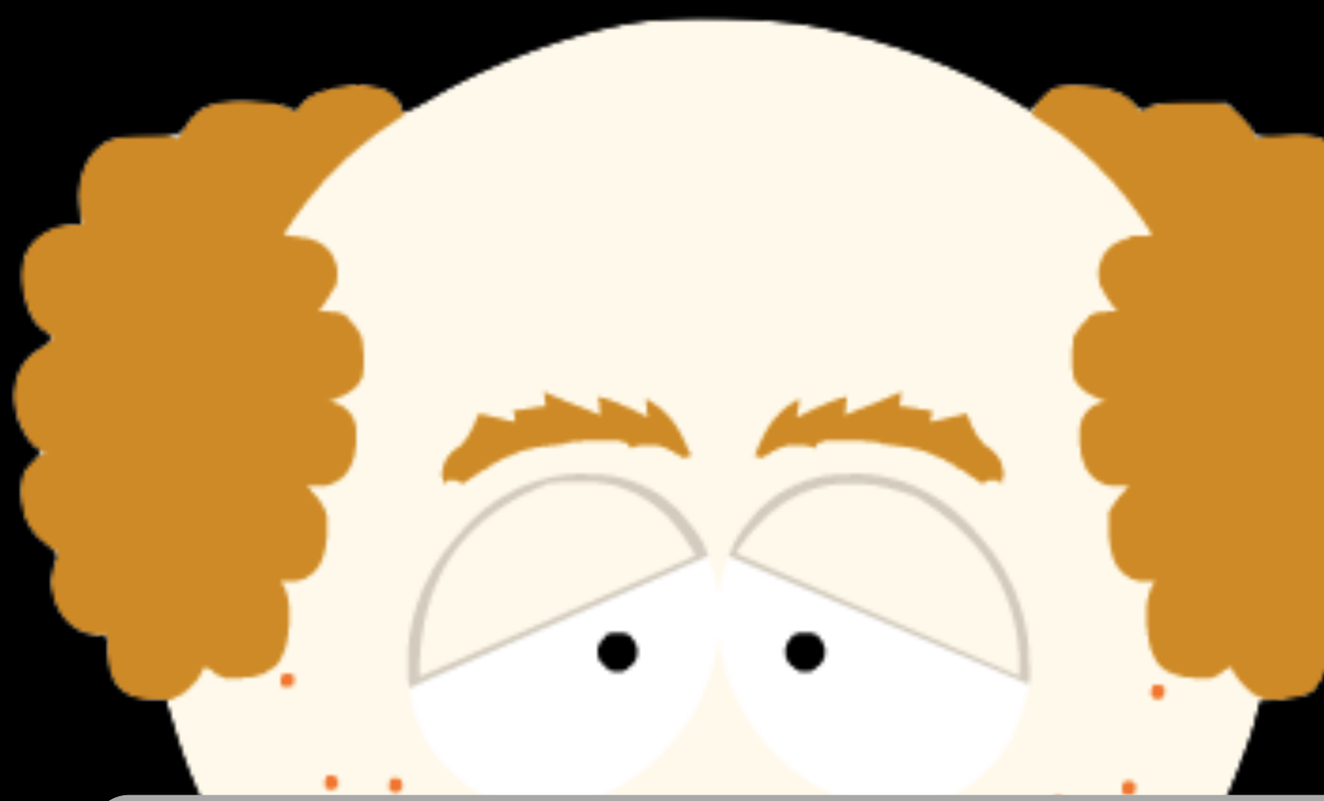
9 of 10 Data Scientist in survey agree, survey no does represent statistically significant sample size.

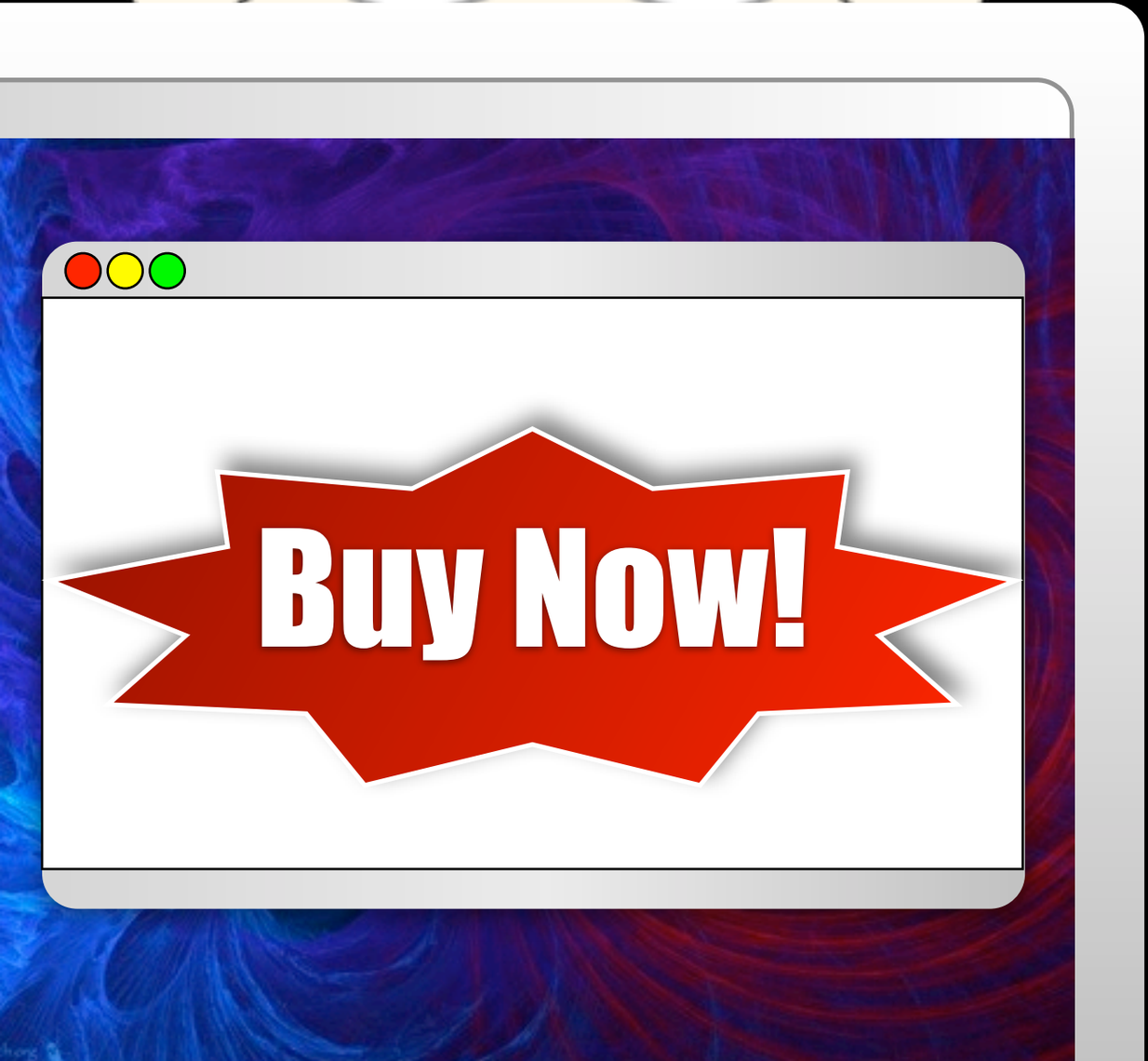
Reply Retweet Favorite More

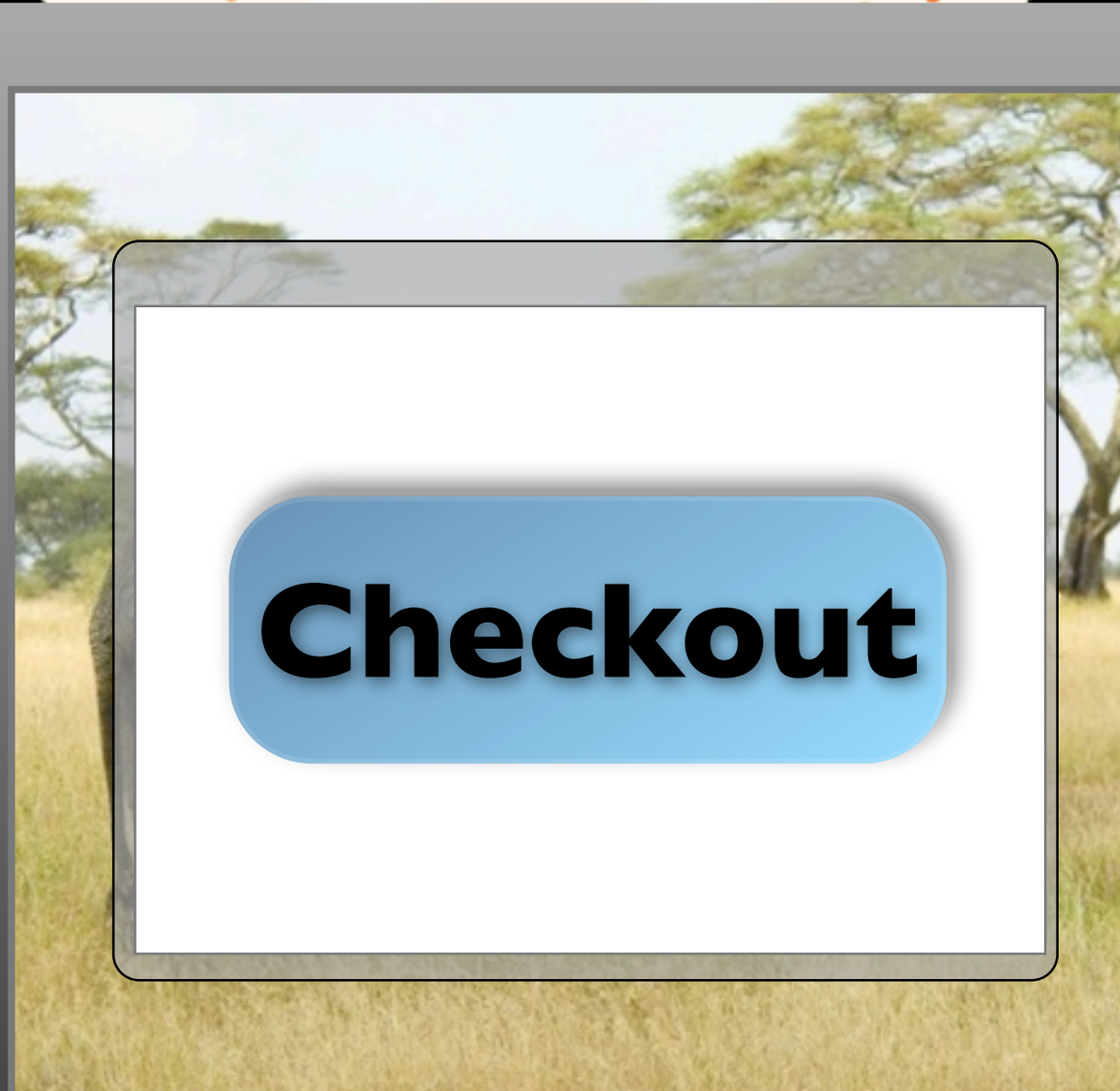
27	2
RETWEETS	FAVORITES

2:24 PM - 17 Apr 13

Reply to @BigDataBorat




Buy Now!


Checkout

Contents

- Goals and Plans
- User Distribution into test groups
- Reporting and Analysis

Contents

- Goals and Plans
- User Distribution into test groups
- Reporting and Analysis

Testing Variations • Tools • Tips

Contents

- Goals and Plans ← *Concepts*
- User Distribution into test groups
- Reporting and Analysis

Testing Variations • Tools • Tips

Contents

- Goals and Plans ← *Concepts*
- User Distribution into test groups ← *Code*
- Reporting and Analysis ← *Code*

Testing Variations • Tools • Tips

Disclaimer


N(μ,σ2)


N(μ,σ2)

Goals and Plans

Planning Tests

- Full company on board?
  Wired's article: *The A/B Test*
- What problems are you facing?
  Focus on the biggest concerns
- Do your tools actually work?
  Run an A/A test first!

Test Effectiveness

Efficiency is doing things right;
effectiveness is doing the right things.

—Peter F. Drucker

Goal

Goal

Test

Answer me, these questions three...

Answer me, these questions three...

- **What is the primary goal?**

Answer me, these questions three...

- **What is the primary goal?**
- **What is the immediate goal?**

Answer me, these questions three...

- **What is the primary goal?**
- **What is the immediate goal?**
- **How will you measure?**

Be Specific

Effective Tests

Quality is not an act, it is a habit.
—Aristotle

Effective Tests

Quality is not an act, it is a habit.
—Aristotle

Test Types

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolor minim

Donec id quis pur nsectetur

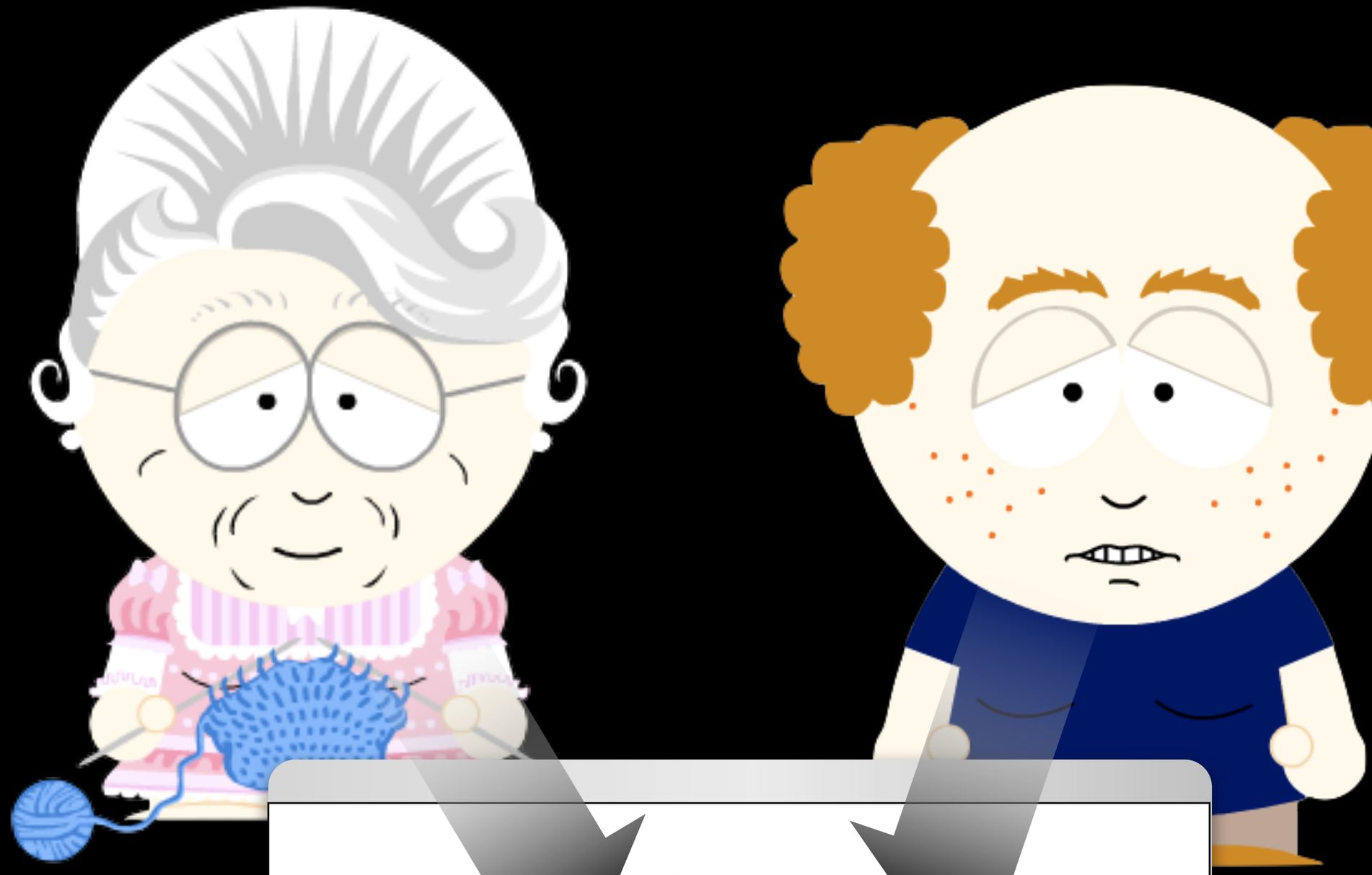
Buy Now!

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim

Donec id elit quis pur consectetur

Checkout

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Nulla nec tortor. Donec id elit quis purus consectetur consequat.



Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labo m ad minim ve lamco labo sequat. N rus co sequat.

Buy out

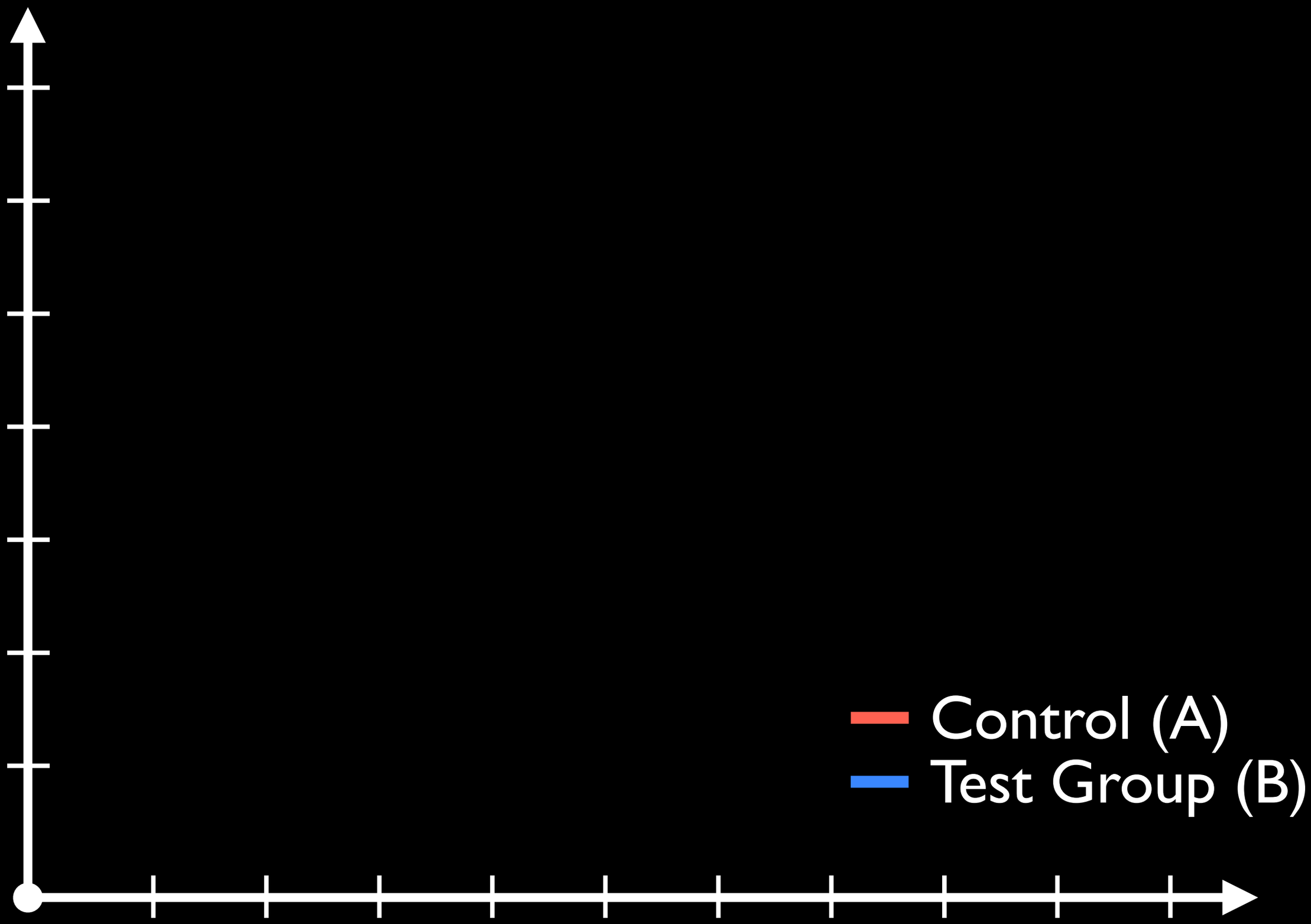
Test Length

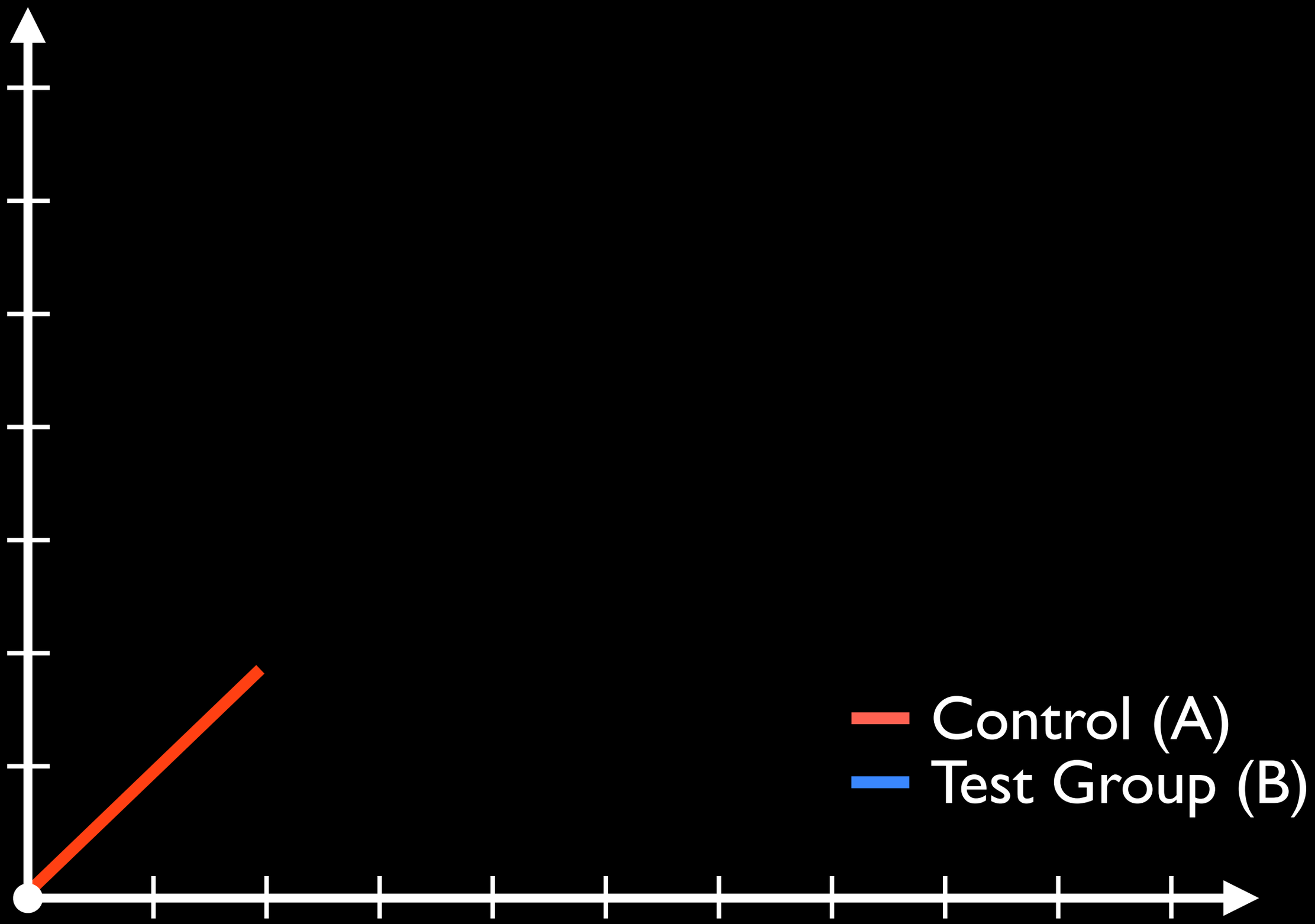
?

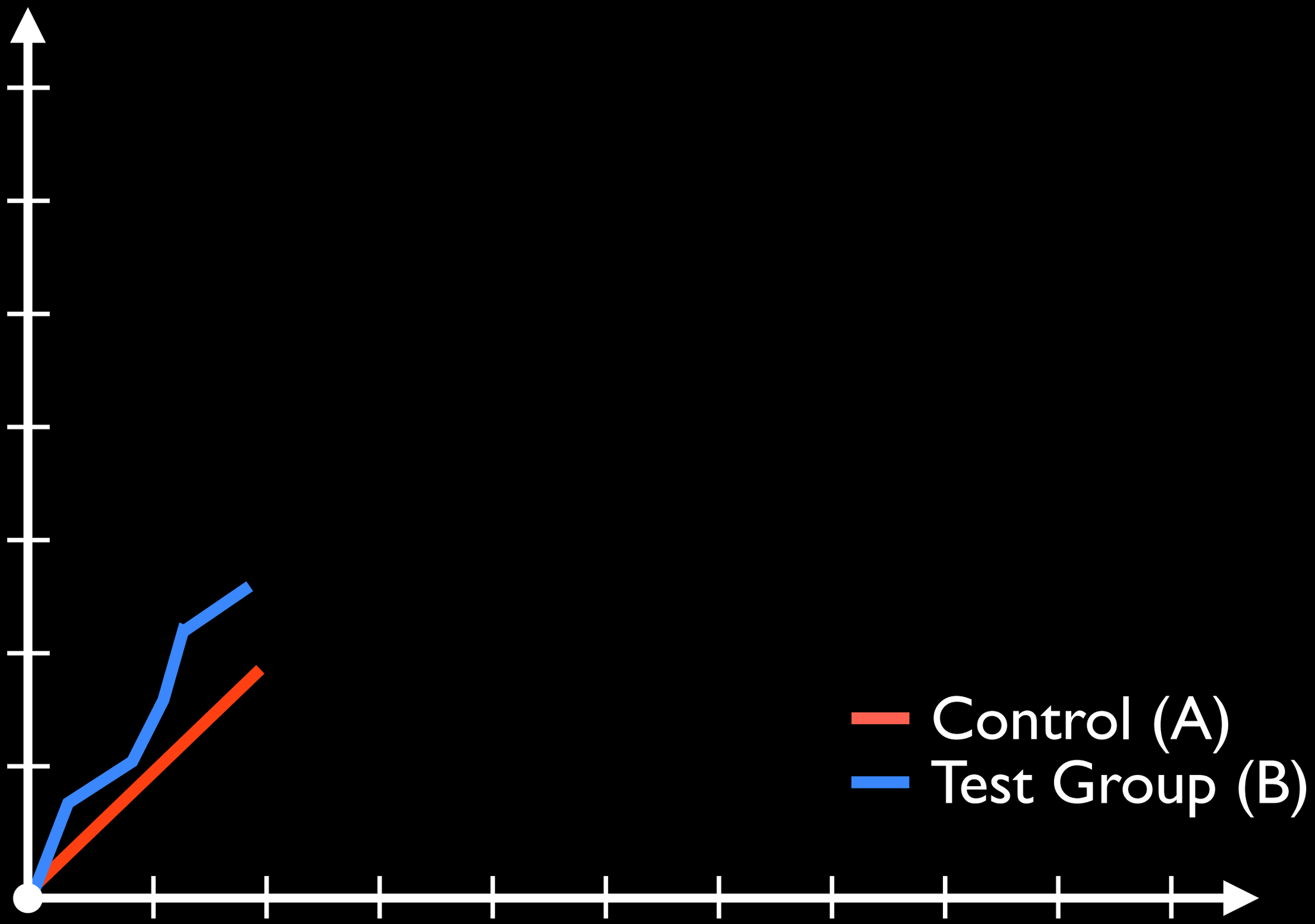
?

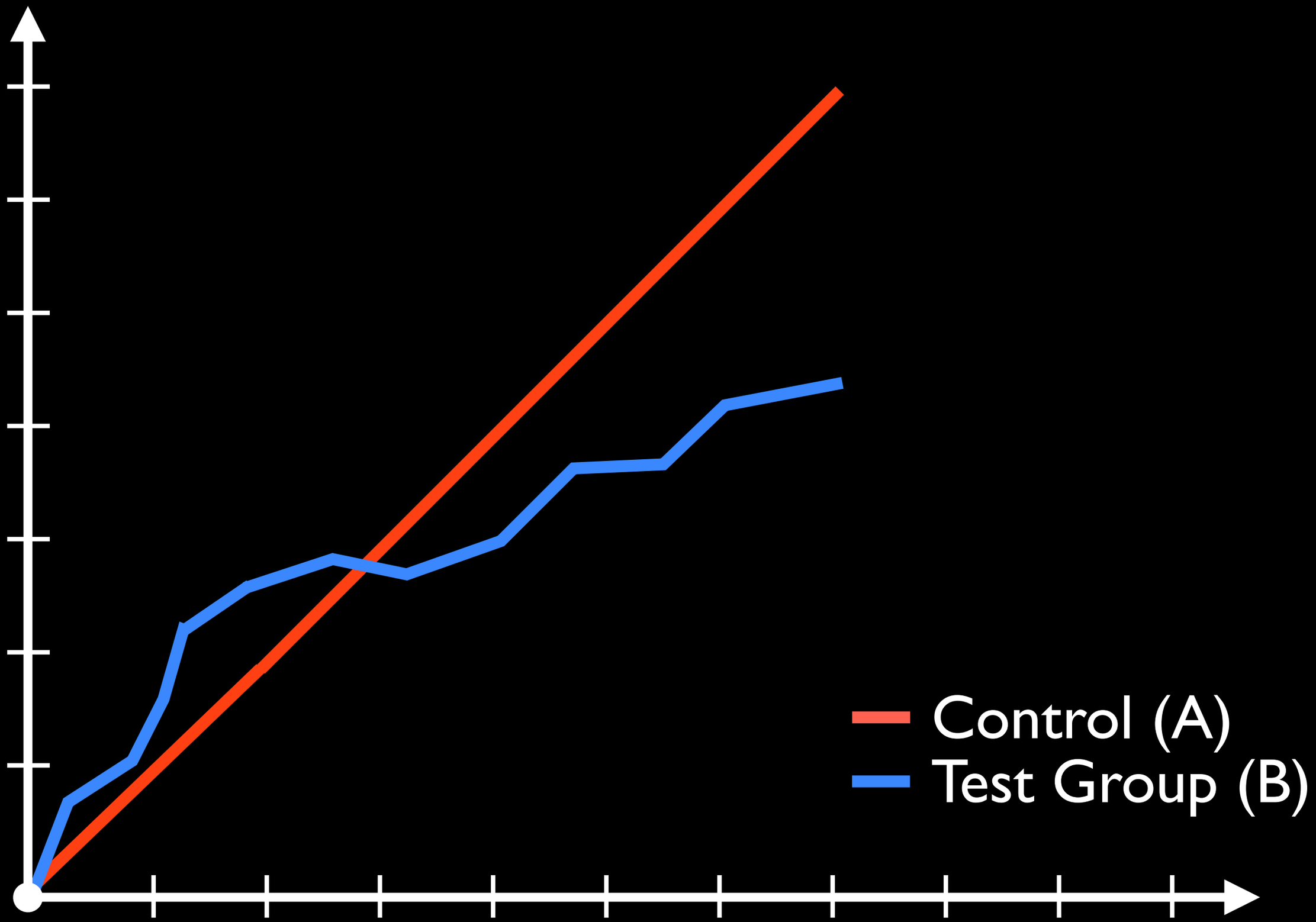
How much data is needed?

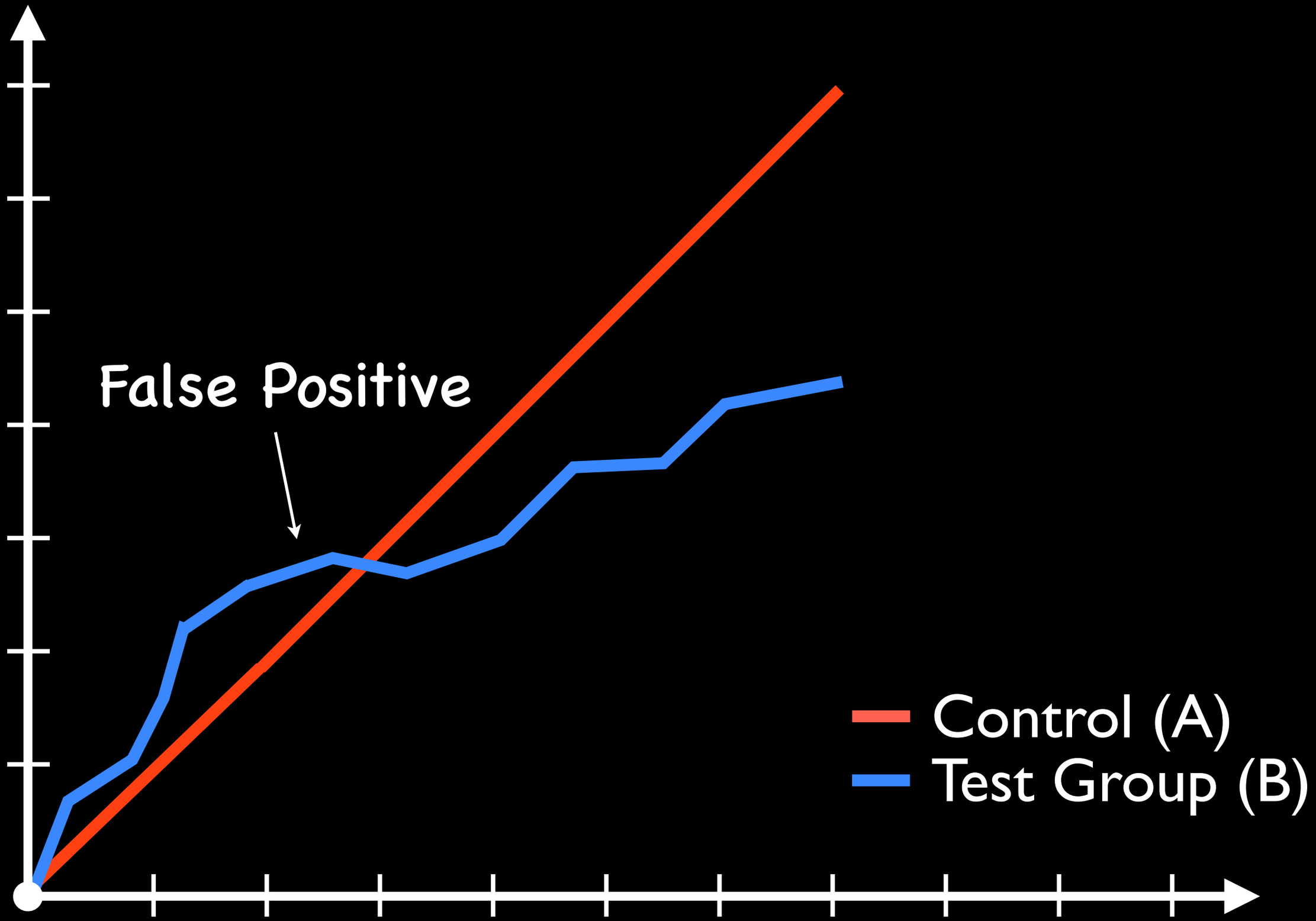
How long should experiment run?

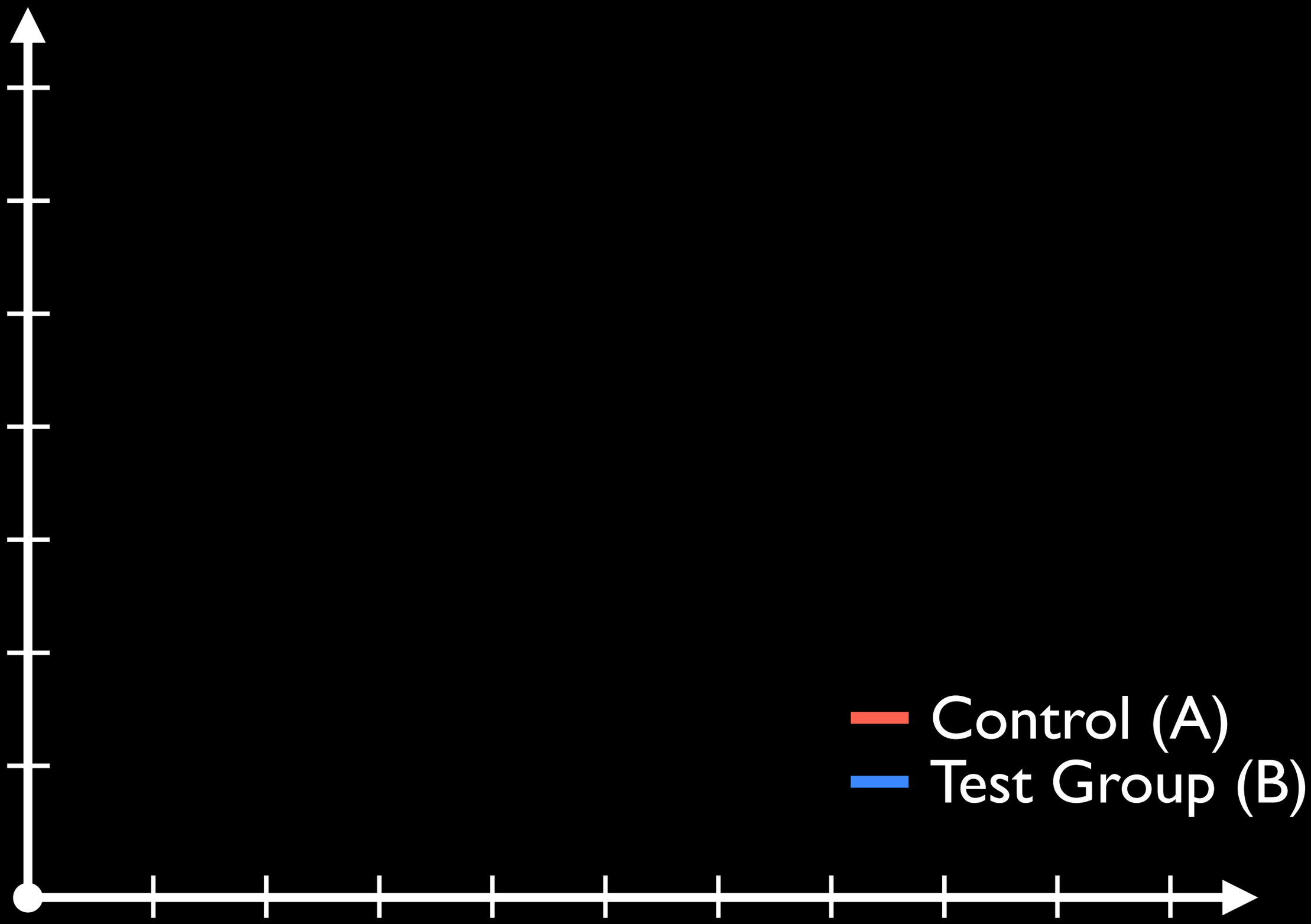

Control (A)
Test Group (B)

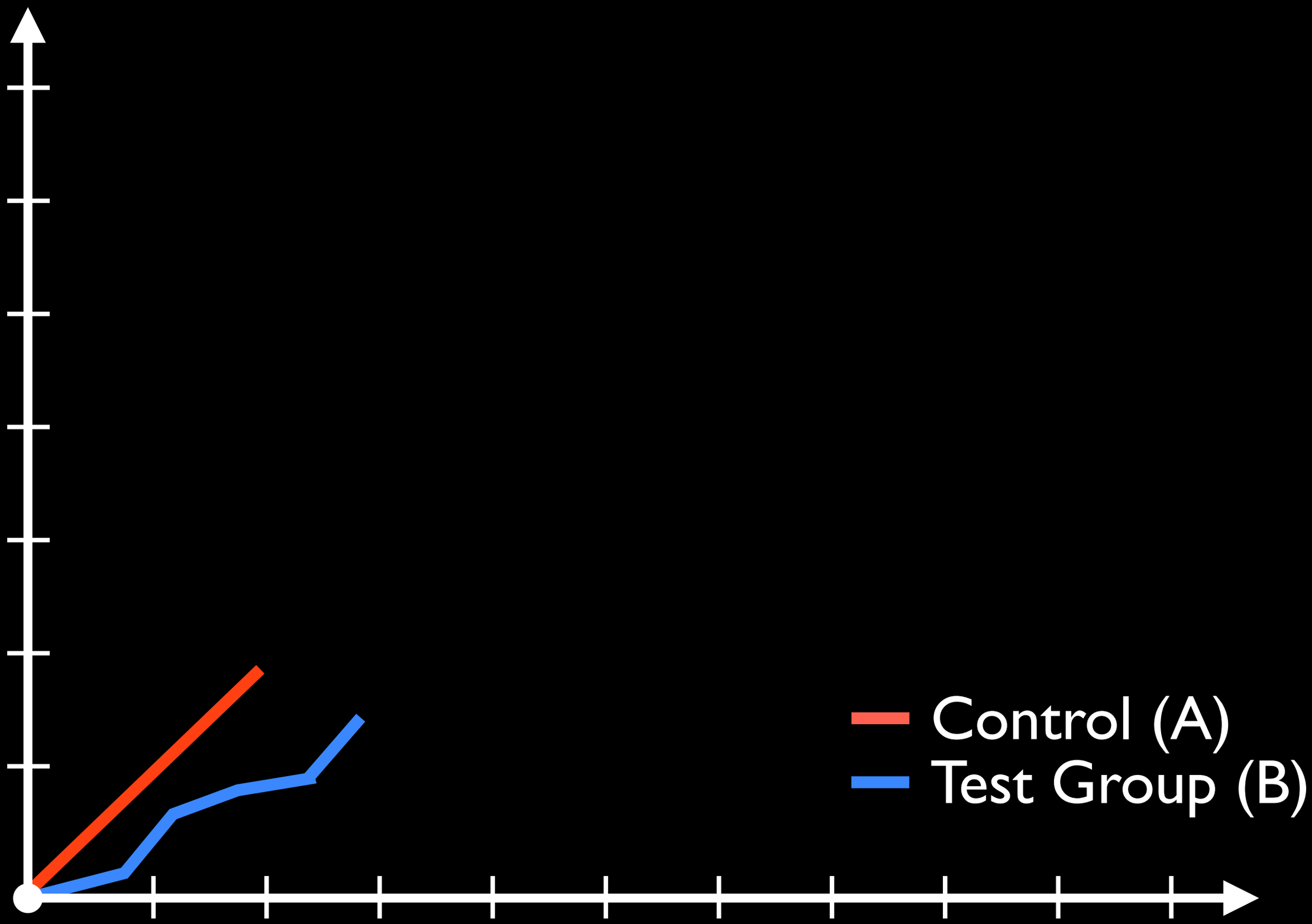

Control (A)
Test Group (B)

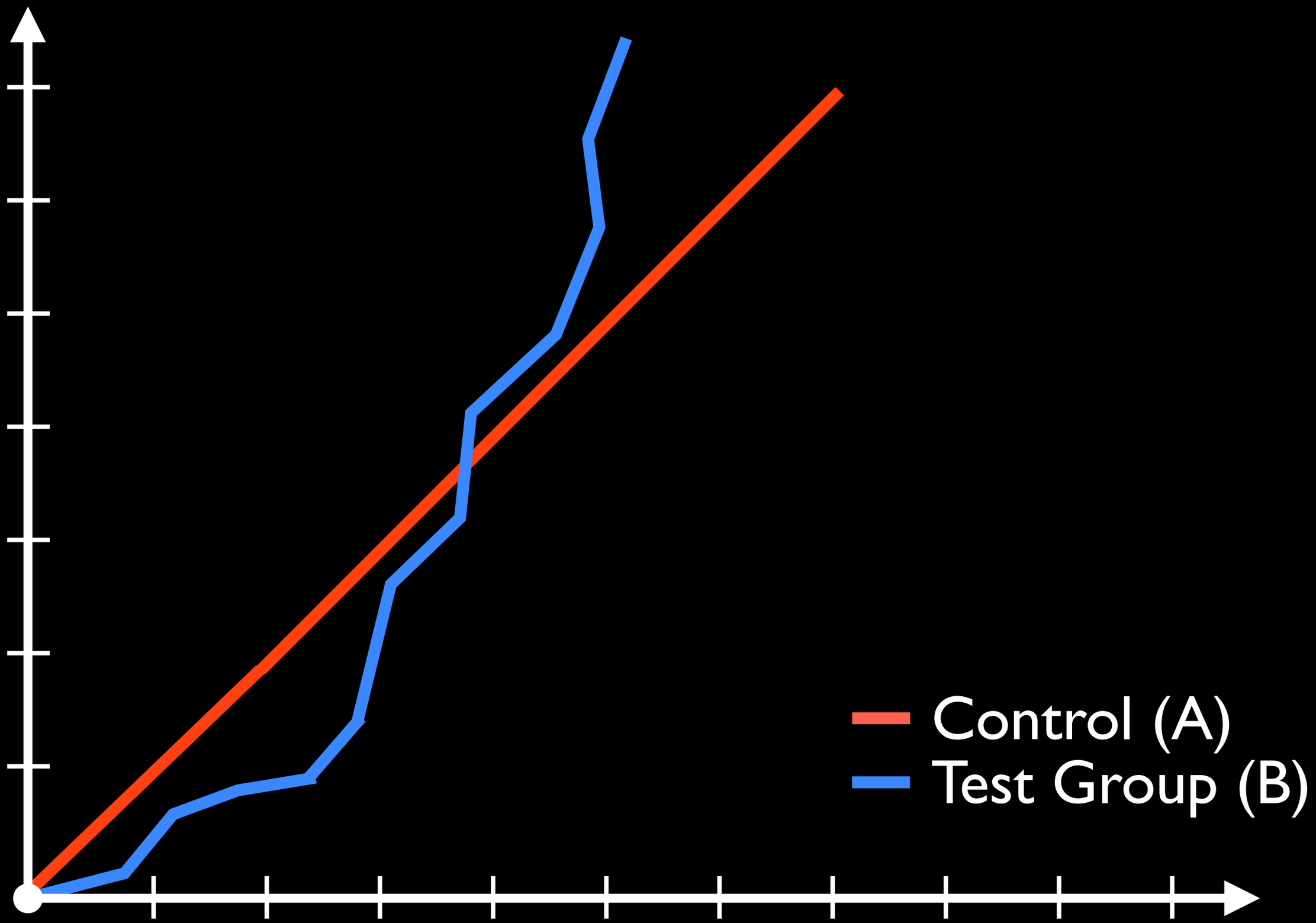

Control (A)
Test Group (B)

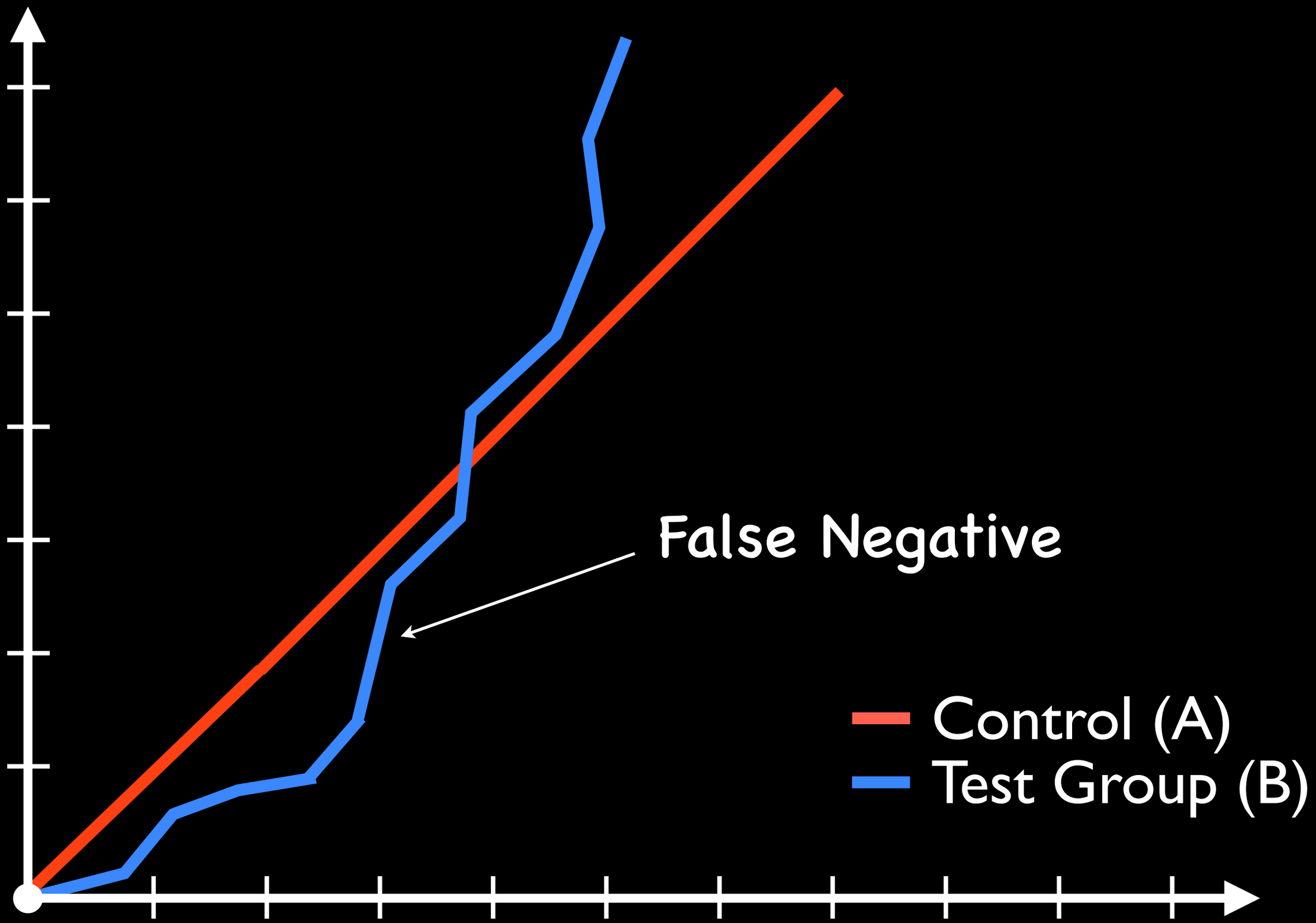

Control (A)
Test Group (B)

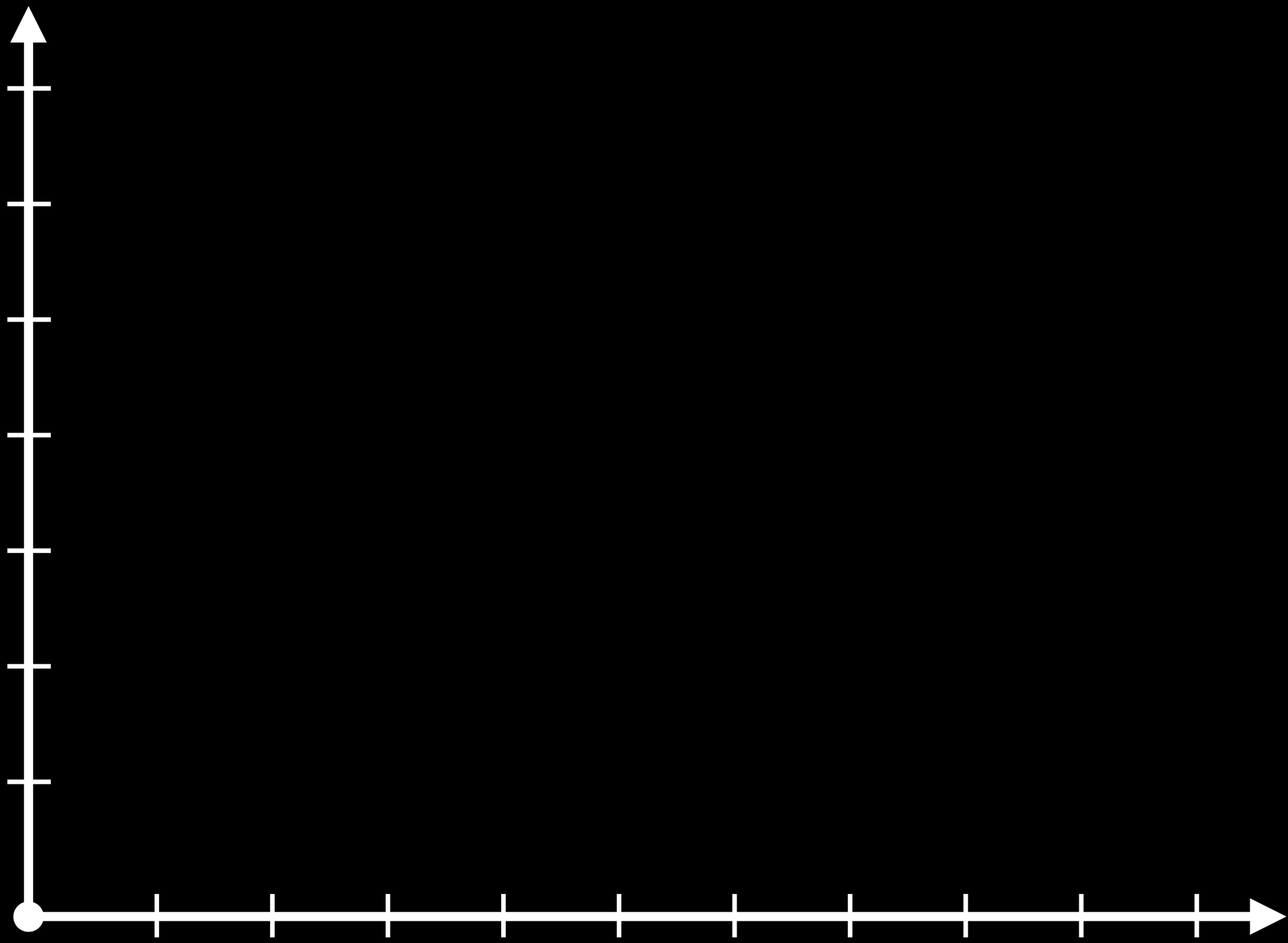

False Positive
Control (A)
Test Group (B)

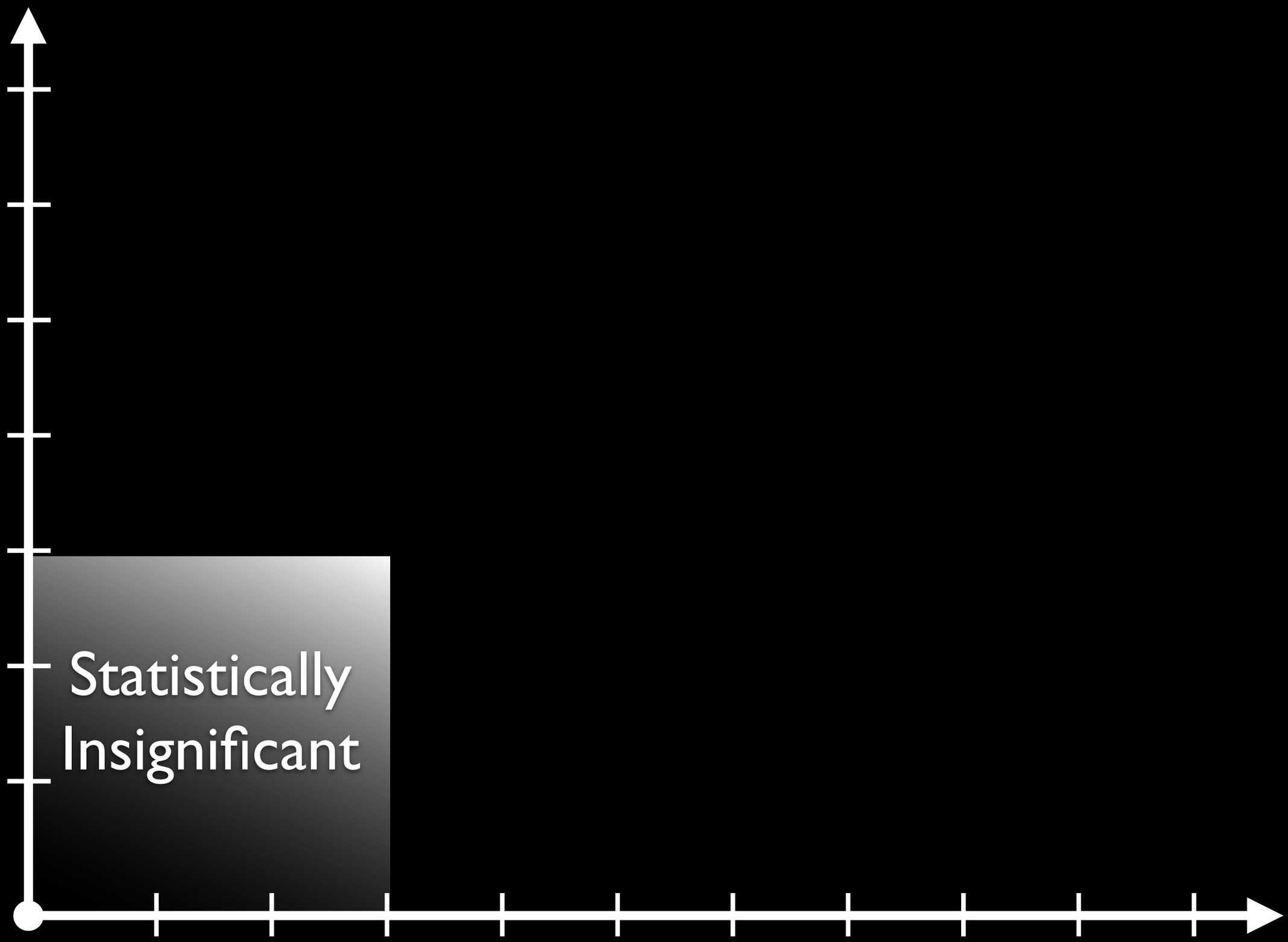

Control (A)
Test Group (B)

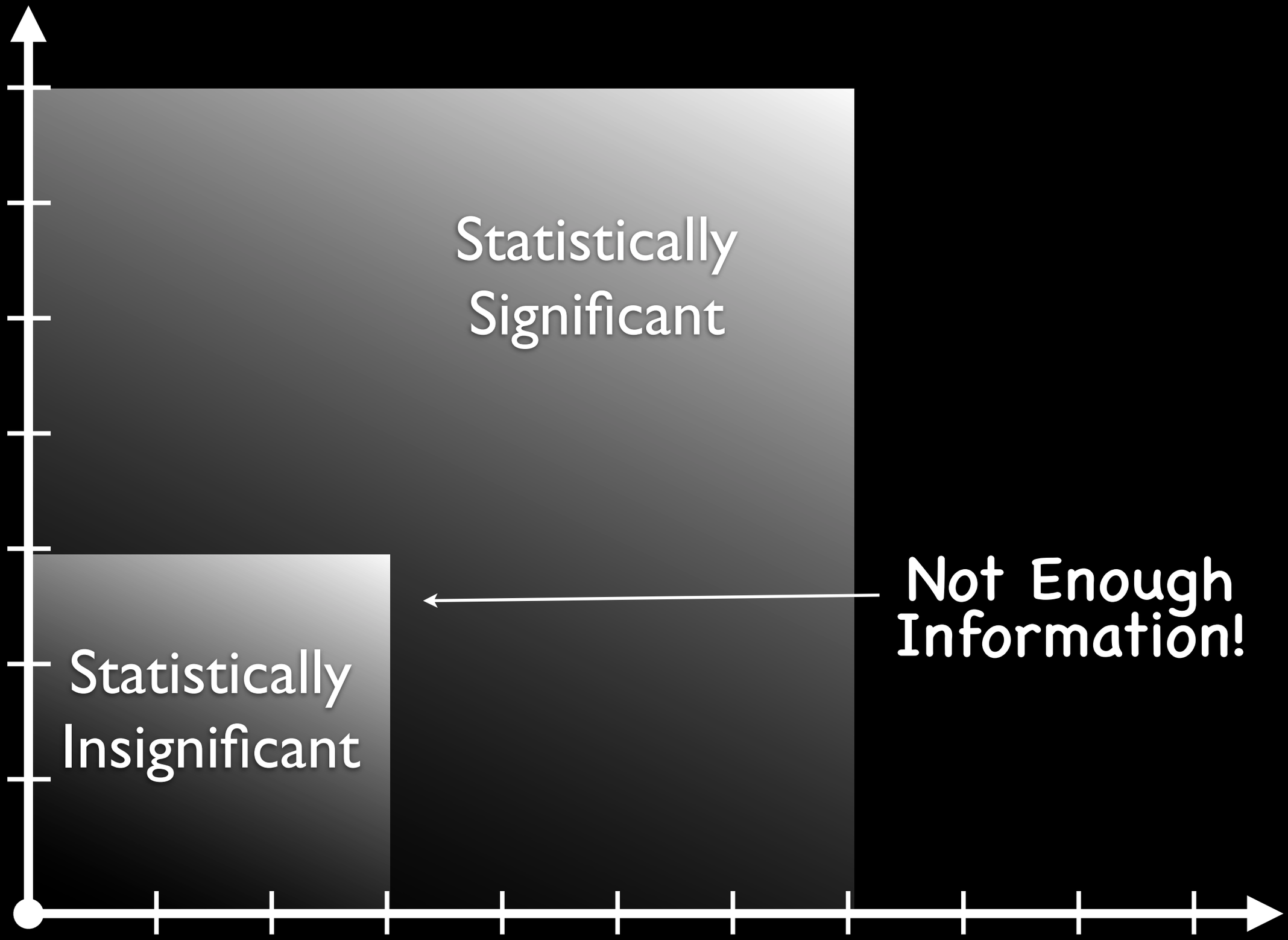

Control (A)
Test Group (B)


Control (A)
Test Group (B)


False Negative
Control (A)
Test Group (B)




Statistically
Insignificant


Statistically
Significant
Statistically
Insignificant
Not Enough
Information!

NO PEEKING!

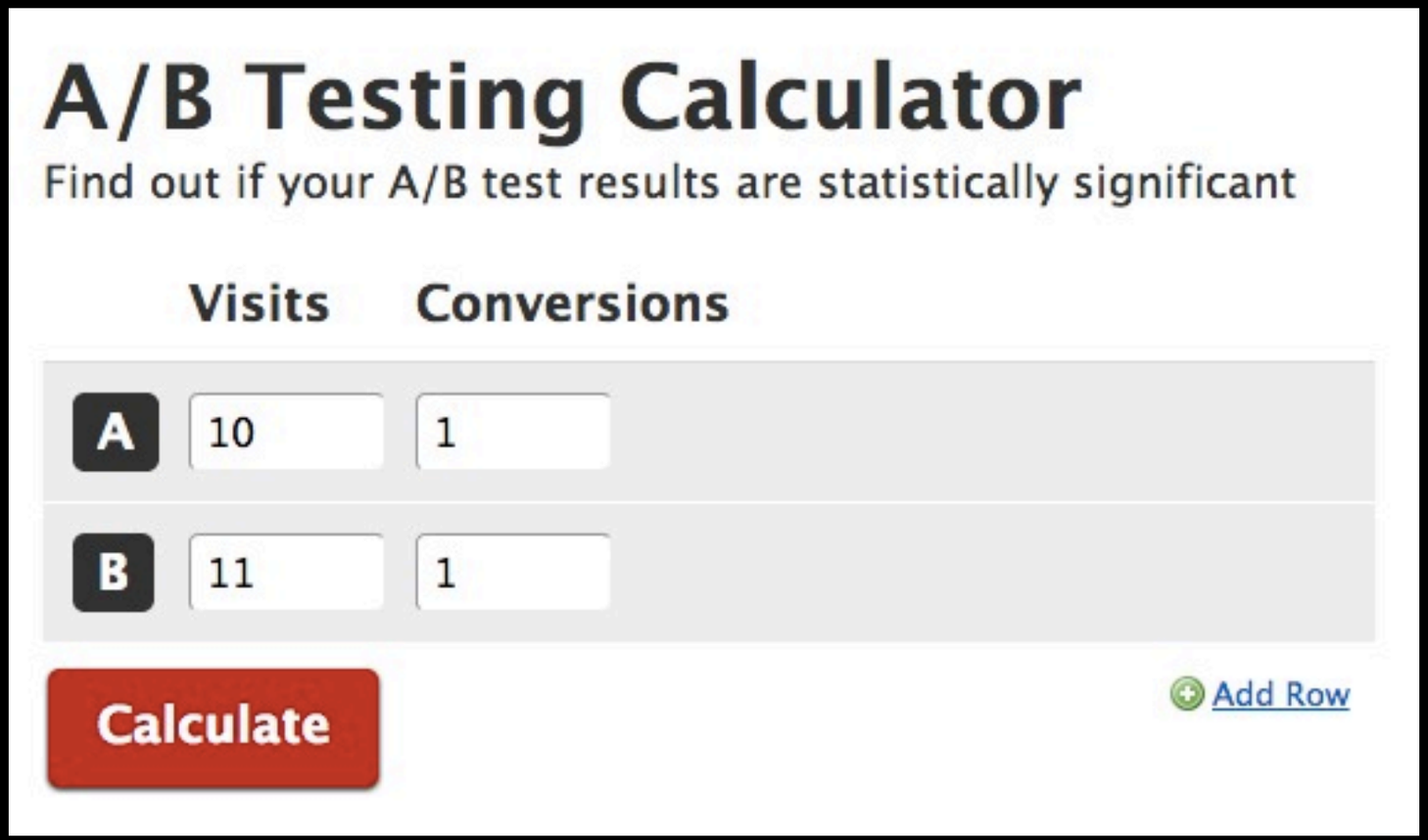
$$n = \frac{16\sigma^2}{\delta^2}$$

σ is sample variance you expect

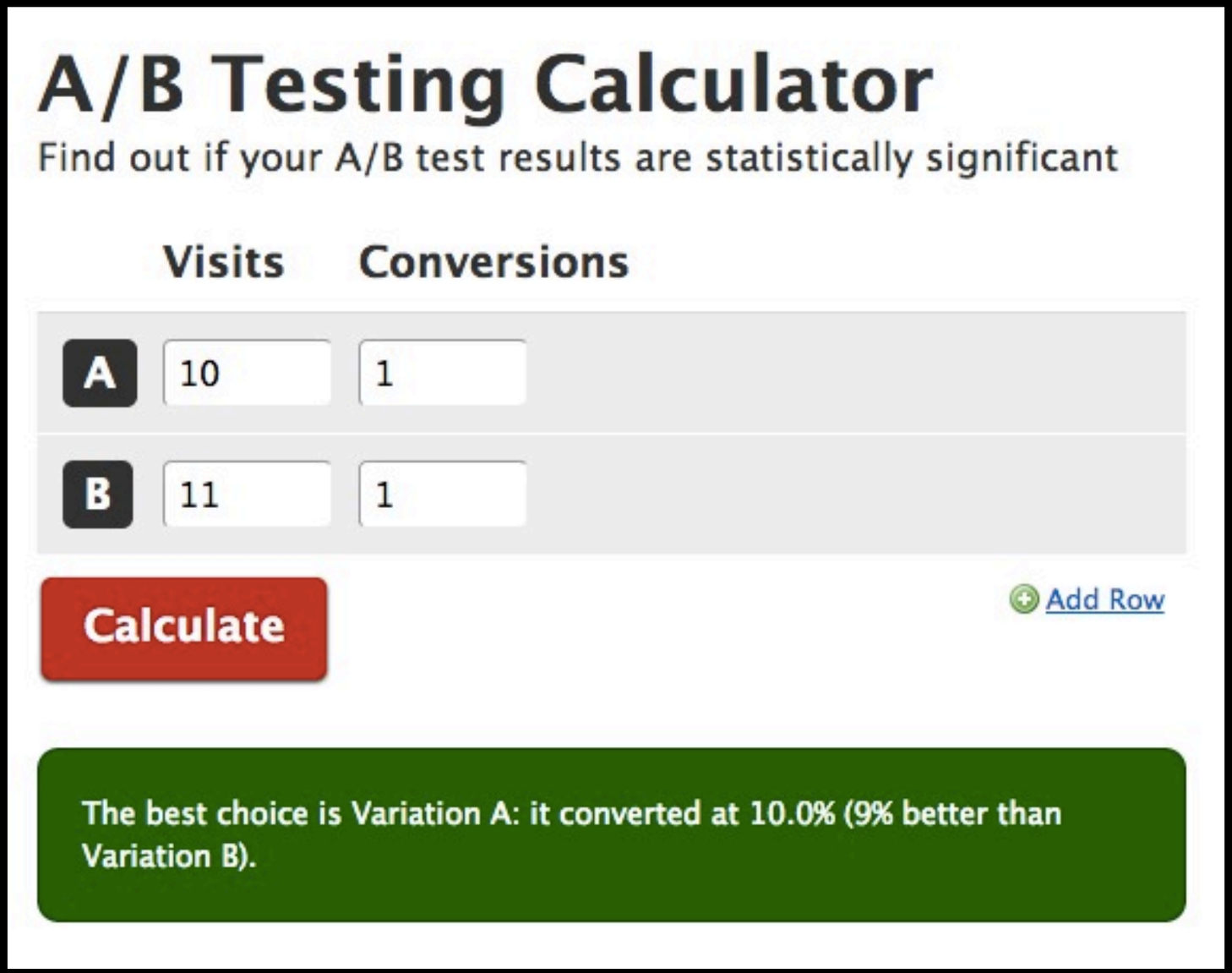
δ is minimum effect to test

$$\sigma^2 = p(1\text{-}p)$$

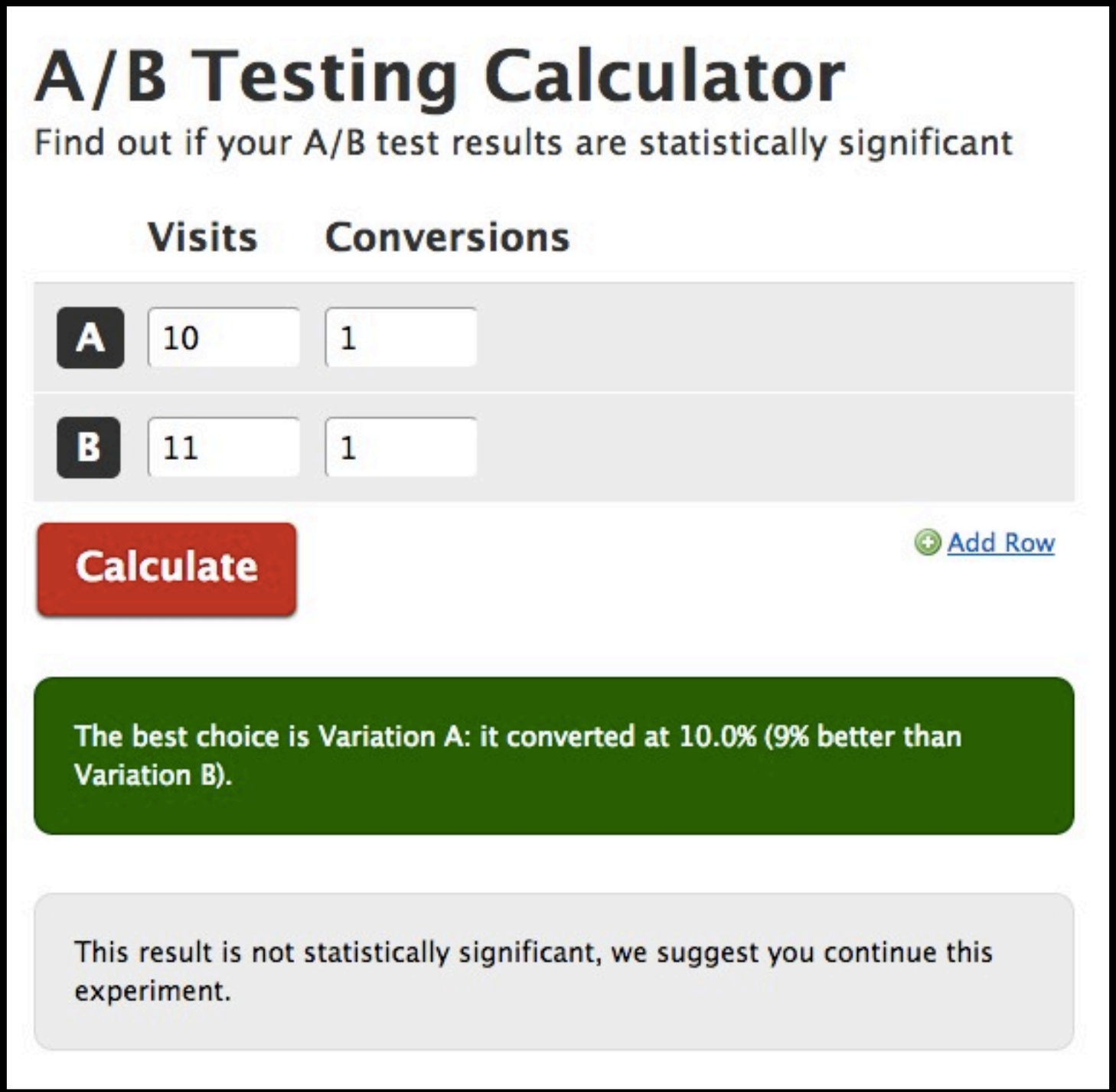
Use Online Calculators



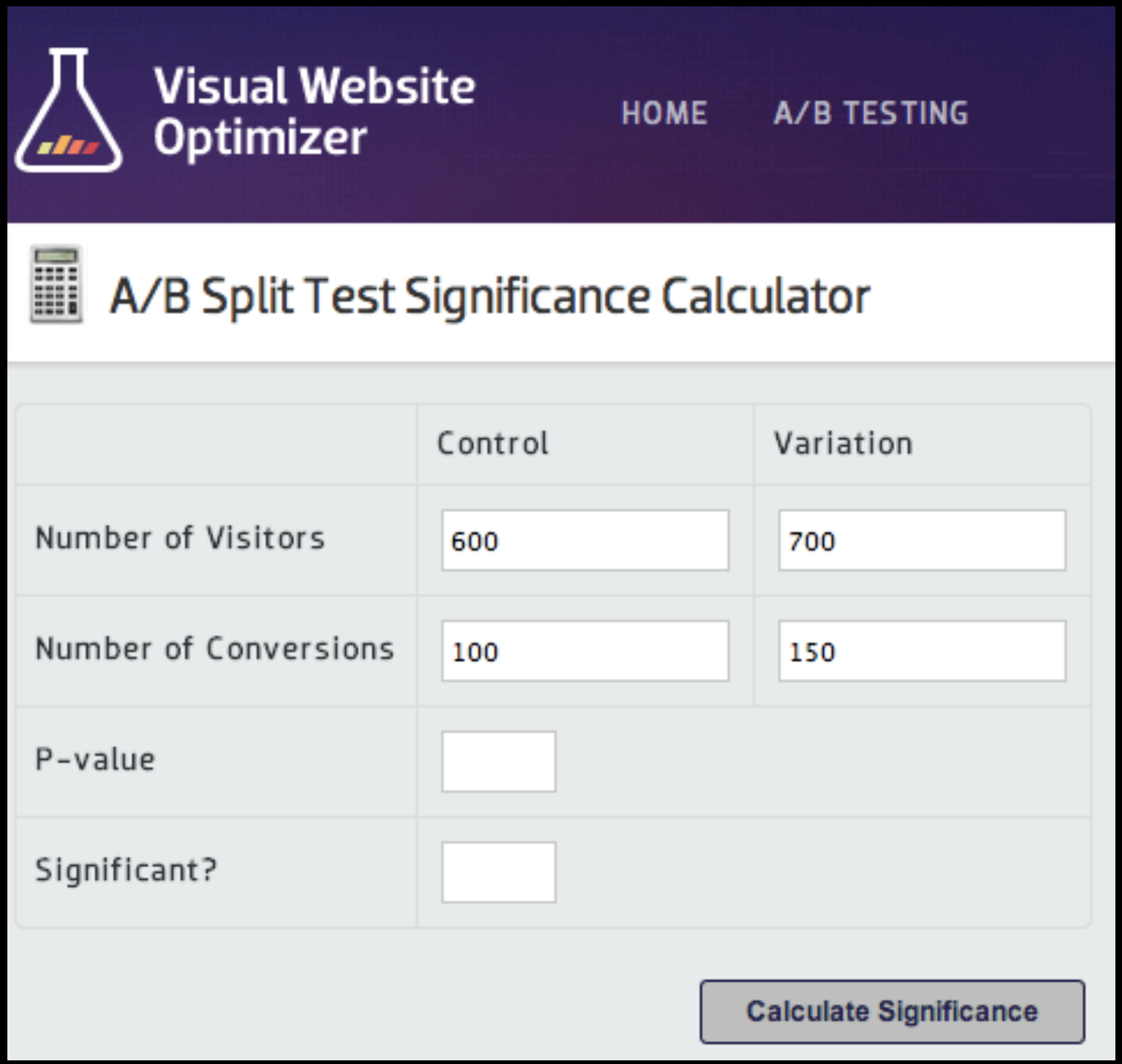
Use Online Calculators



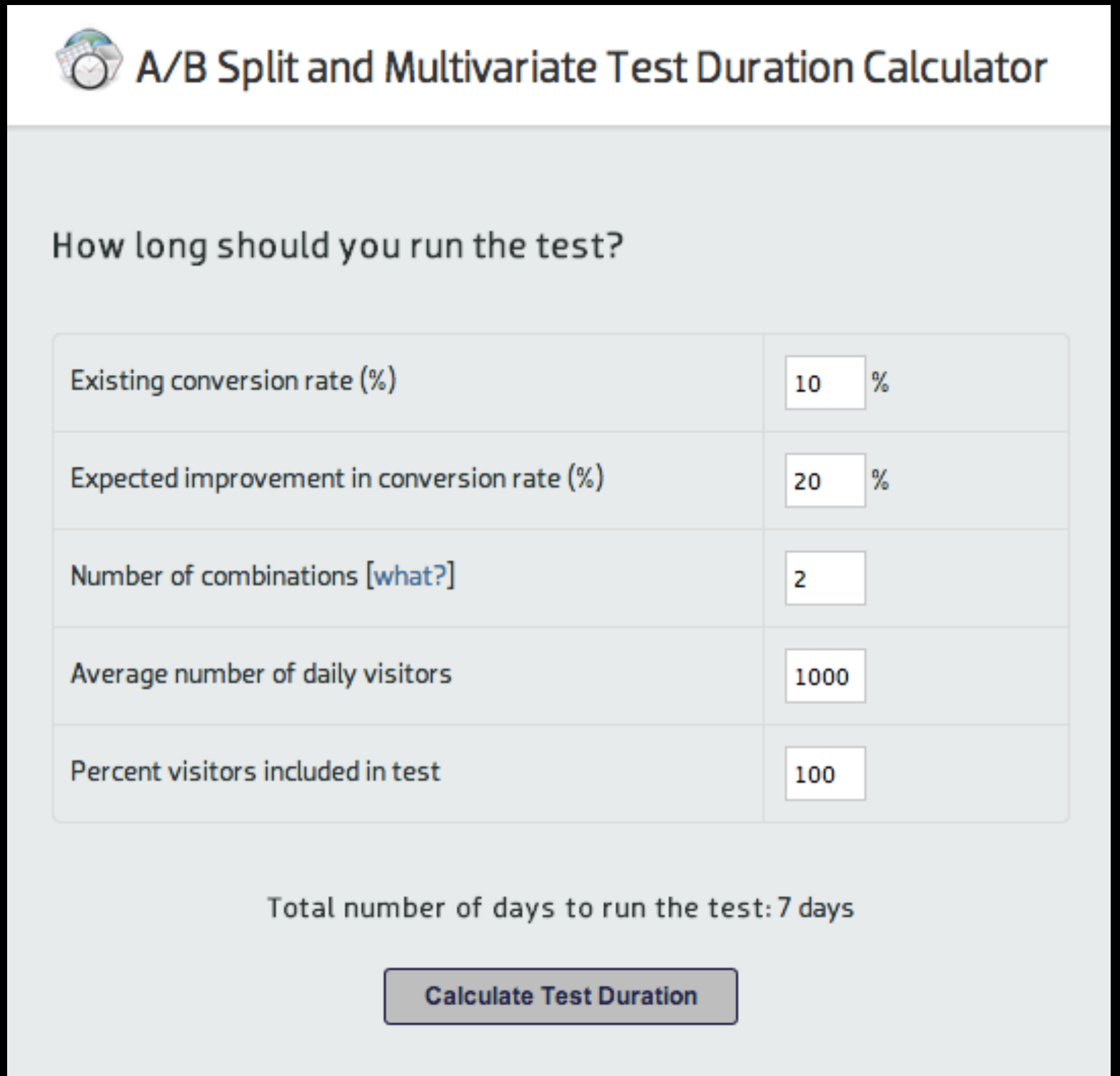
Use Online Calculators



Use Online Calculators



Use Online Calculators



Distribution

Customer Distribution

Act of assigning your customers to *groups*

1. Convert customer to a number
2. Combine number with test label
3. Modulus with the number of groups

Customer Distribution

Act of assigning your customers to *groups*

1. **Convert customer to a number**
2. Combine number with test label
3. Modulus with the number of groups

Customer Distribution

Act of assigning your customers to *groups*

1. **Convert customer to a number**
   - Don't care much about *uniqueness*.
   - Care more about *spread*.
2. Combine number with test label
3. Modulus with the number of groups

String to Numbers

```
function hash(key) {
    var results = 0;
    for (c in key) {
        results += key.charCodeAt(c);
    }
    return Math.abs(results);
}
```

String to Numbers

```
function hash(key) {
    var results = 0;
    for (c in key) {
        results += key.charCodeAt(c);
    }
    return Math.abs(results);
}

hash("hello") => 532
hash("24DE5F01-E0F8-4DB6-8921-125B64AAF3B7") => 2030
```

String to Numbers

```
function hash(key) {
    var results = 0;
    for (c in key) {
        results += key.charCodeAt(c);
    }
    return Math.abs(results);
}

hash("hello") => 532
hash("24DE5F01-E0F8-4DB6-8921-125B64AAF3B7") => 2030
```

Seems a bit... uhm,
simplistic to be good.

String to Numbers

Part 2

```
function hash(key) {
    var results = 0;
    for (c in key) {
        results += ( key.charCodeAt(c) << c);
    }
    return Math.abs(results);
}
```

String to Numbers

Part 2

```
function hash(key) {
    var results = 0;
    for (c in key) {
        results += ( key.charCodeAt(c) << c);
    }
    return Math.abs(results);
}

hash("hello") => 3378
hash("24DE5F01-E0F8-4DB6-8921-125B64AAF3B7") => 1223556814
```

String to Numbers

Part 2

```
function hash(key) {
    var results = 0;
    for (c in key) {
        results += ( key.charCodeAt(c) << c);
    }
    return Math.abs(results);
}

hash("hello") => 3378
hash("24DE5F01-E0F8-4DB6-8921-125B64AAF3B7") => 1223556814
```

So? How did this algorithm play out?

57844
36986
79094
86960
3152
86532
86178
12130
83592
60162
29680
175
21522
82582

kimberlysgarcia@einrot.com
alyssagtoupin@gustr.com
shirleyrwilliams@superrito.com
lauraaparker@cuvox.de
marywtaylor@cuvox.de
philipwallen@teleworm.us
patriciaabarringer@einrot.com
gulelmartinez@teleworm.us
earllgladden@gustr.com
leahbpeterson@dayrep.com
susantruth@teleworm.us
georgejcardona@dayrep.com
kathleenjmayfield@gustr.com
janaburns@teleworm.us

placentoid
Encrinidae
eyefulproturan
unevolutionary
Hippoboscidae
kenopremotion
physiquedsyriac
outshakepunger
Pituitrinheaving
Myxogasteres
defycourtlike
unnarrowheadcheese
extoolitic
overexpansive
becrosspremorial
revoltingintegrally
ureterointestinal
shutdownabjectness
hypoalkalinity
biostaticfilamentary
frowyElizabethanism
secularization
rectovesical
predespondency
antifaction

Edwin Brunson
Samuel Marshall
Dara Foxwell
James Henshaw
Emma Hammer
Eusebio Bush
Tina Yun
Patricia Padua
Lonnie Drakes
Bruce Updegraff
June Macdougall
Ilda Smith

17f88680-8250-4498-b7ff-b5df5bf05f49
95af91ed-52ca-4cc5-a195-838badbce810
bd9ace7d-a64b-4af3-961e-9bdf66bad6e8
42da06c6-f037-4a07-956e-bf23db654345
7f673ece-83ef-44ba-a421-a862e467ab17
58672fae-b283-4649-84d3-742ee9e68eb6
92b93fe2-80ef-4d37-9ffa-80b85cafece2
f9850617-f20d-47f0-a6bc-54d791d04586
afd7b826-111d-43cc-abdd-9b3c0c3f197d
12ae19e0-96a5-4687-9deb-151964f1785a
07f34164-dc49-4710-a0a3-6bbb3b86896d
f1b8d68d-8895-49b3-afba-8a24cc3314b8
7e646065-ce17-4534-b87c-96c4399ce4ce
08e75204-26d1-4eda-bcd2-3a15051a2097
8257545b-22e0-4924-8f10-9b8f1b4f705b
905eed70-ac4b-420a-a929-24605cd3cba0
45b328ee-07cf-4389-a472-df5e2c762002
77963e13-ad04-4e5a-9c32-19eb95c542cb
6fa14634-d5f7-41b6-9fd4-78a7adedcd23
a7ee4ff6-d090-4794-be7c-7770b2c7be78
8749256c-3d29-4a1a-af77-197bddace5e9
74294588-3c74-4369-a4a8-3525e98bca06
32cdac95-33c2-48cf-858b-61796ef0b2c0
4b8ac70f-561f-4c34-8528-d2b9e95d06a6
cd0b8524-8ed1-412e-9422-40206be97fc1
492a15e0-b61c-4187-8ef4-cd628cd2a926
715bdb8a-bd55-4334-a2fd-3082178231d5
759401ff-1d17-43fb-9957-b46aea63df5a
743735da-c989-41c3-a889-59f9c502543f
d7882e98-3a44-4082-925a-815358cb6319

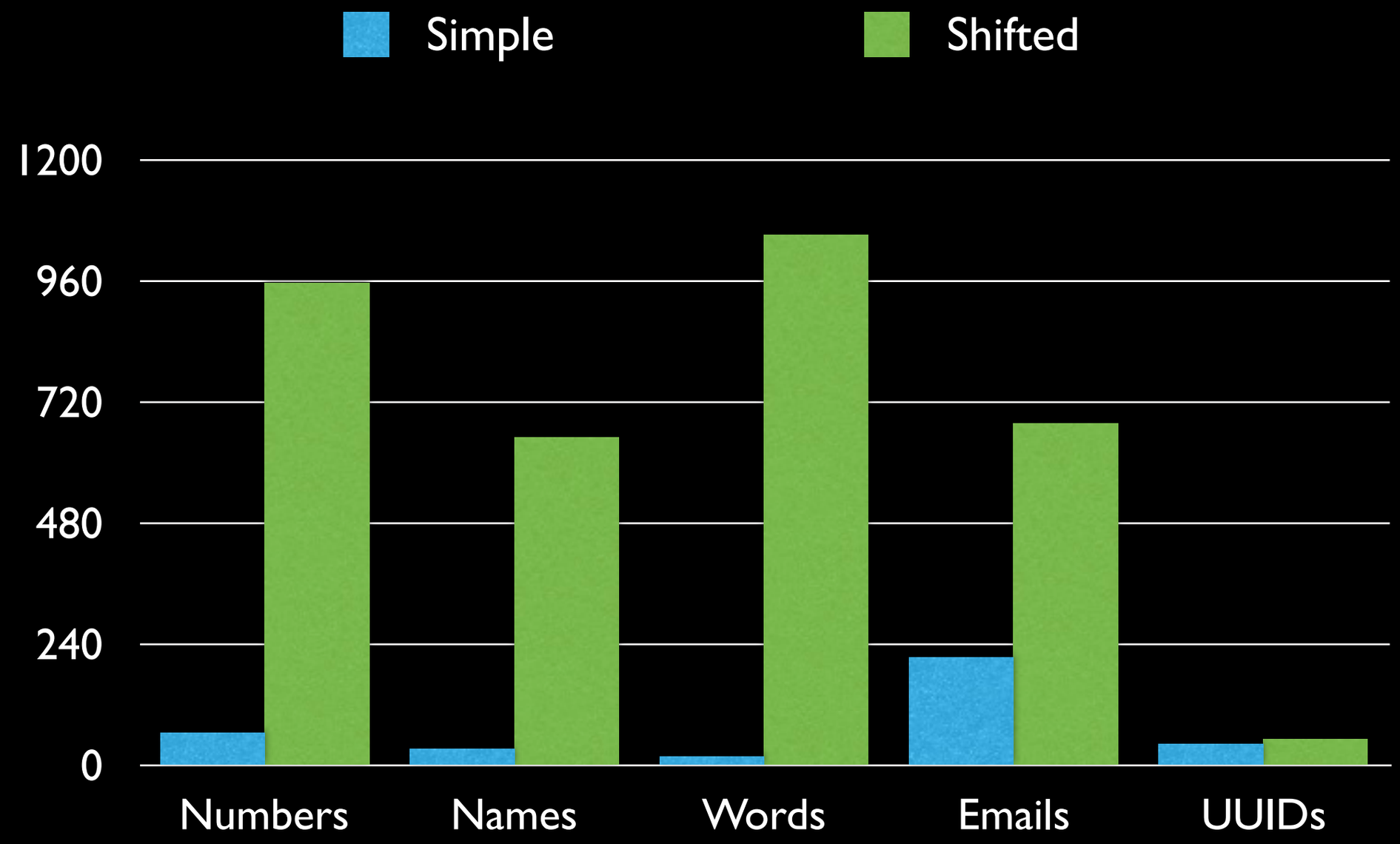
Simple Hash Results

Data Type	Group A	Group B	Delta
numbers	4,968	5,032	64
names	4,984	5,016	32
words	4,991	5,009	18
emails	4,893	5,107	214
uuids	4,979	5,021	42

Shifted Hash Results

Data Type	Group A	Group B	Delta
numbers	4,522	5,478	956
names	4,675	5,325	650
words	4,474	5,526	1,052
emails	4,661	5,339	678
uuids	4,974	5,026	52

Simple is Better



Customer Distribution

Act of assigning your customers to *groups*

1. **Convert customer to a number**
2. Combine number with test label
3. Modulus with the number of groups

Customer Distribution

Act of assigning your customers to *groups*

1. Convert customer to a number
2. **Combine** number with **test label**
3. Modulus with the number of groups

Customer Distribution

Act of assigning your customers to *groups*

1. Convert customer to a number
2. **Combine** number with **test label**
    - Most tests are split (A/B)
    - A is often the control
    - Fred is ID of 123 is always being tested!
3. Modulus with the number of groups

Combine ID and Test

```
var group = hash( userID + testName ) % 2;
```

Combine ID and Test

```
var group = hash( userID + testName ) % 2;
```

```
var useridA = 10, useridB = 11;
var testid1  = 1, testid2  = 2;

useridA + testid1; //-> 11
useridB + testid1; //-> 12

useridA + testid2; //-> 12
useridB + testid2; //-> 13
```

Combine ID and Test

```
var group = hash( userID + testName ) % 2;
```

```
var useridA = 10, useridB = 11;
var testid1  = 1, testid2  = 2;

useridA + testid1; //-> 11
useridB + testid1; //-> 12

useridA + testid2; //-> 12
useridB + testid2; //-> 13
```

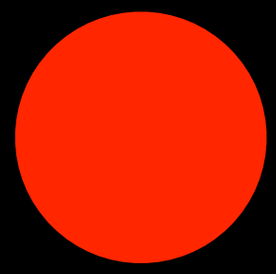
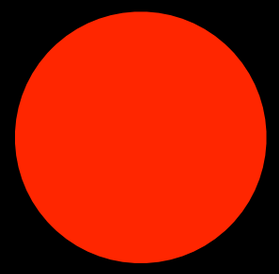
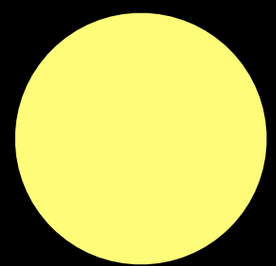
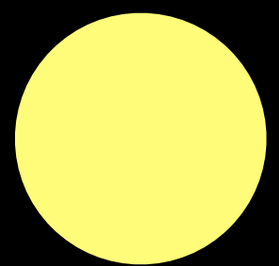
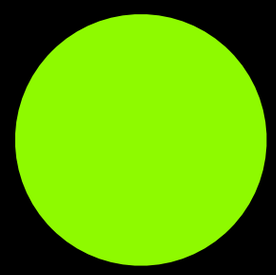
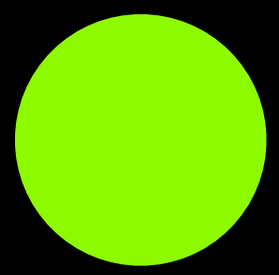
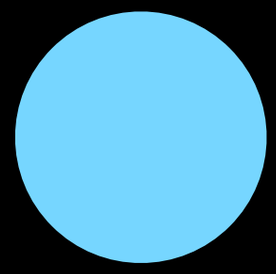
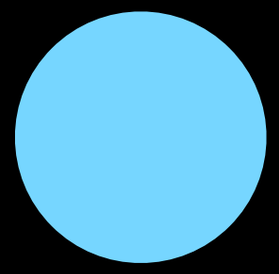
- Even IDs are always in the same group!

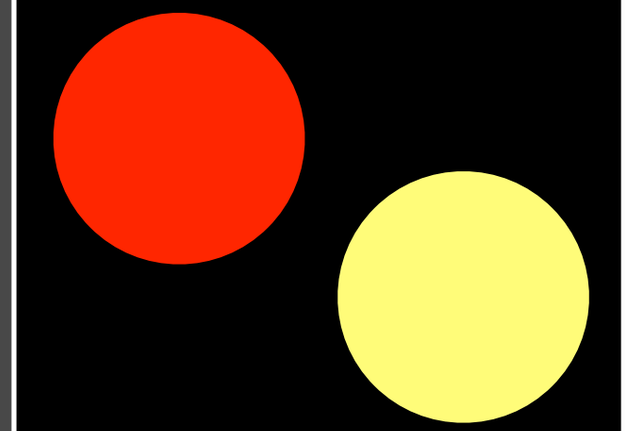
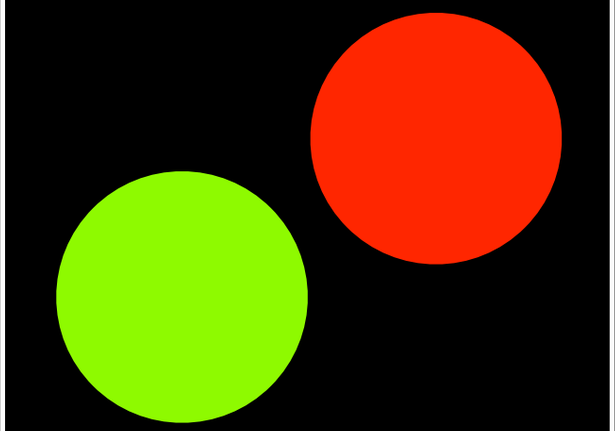
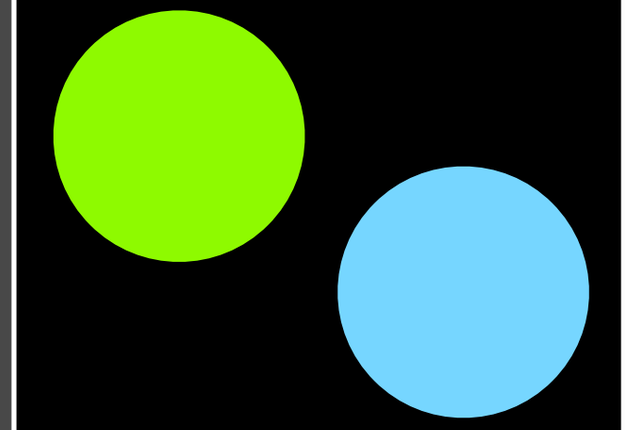
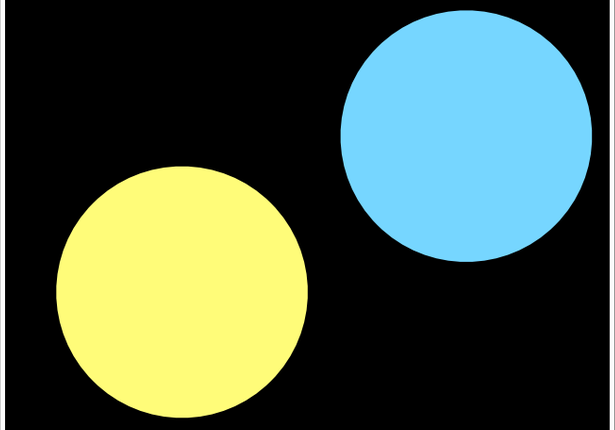
1: Fred

2: Wilma

3: Barney

4: Betty

	Test 1	Test 2
1: Fred		
2: Wilma		
3: Barney		
4: Betty		

	Test 1	Test 2
Group A		
Group B		

Blend on High...

```
function combine(id, name) {
    var results = "";
    for (var i = 0; i < Math.min( id.length, name.length ); i++) {
        results += id[i] + name[i];
    }
    return results + id.substr(i) + name.substr(i);
}
```

Blend on High...

```
function combine(id, name) {
    var results = "";
    for (var i = 0; i < Math.min( id.length, name.length ); i++) {
        results += id[i] + name[i];
    }
    return results + id.substr(i) + name.substr(i);
}
```

Gah... The critical last letter will always go to the longest string...

Blend on High...

```
function combine(id, name) {
    var results = "";
      for (var i = id.length-1, j = name.length-1;
                i >= 0 && j >= 0; i--, j--) {
          results = id[i] + name[j] + results;
      }
      return id.substr(0, i+1) + name.substr(0, j+1) + results;
}
```

Blend on High...

```
function combine(id, name) {
    var results = "";
      for (var i = id.length-1, j = name.length-1;
                i >= 0 && j >= 0; i--, j--) {
          results = id[i] + name[j] + results;
      }
      return id.substr(0, i+1) + name.substr(0, j+1) + results;
}
```

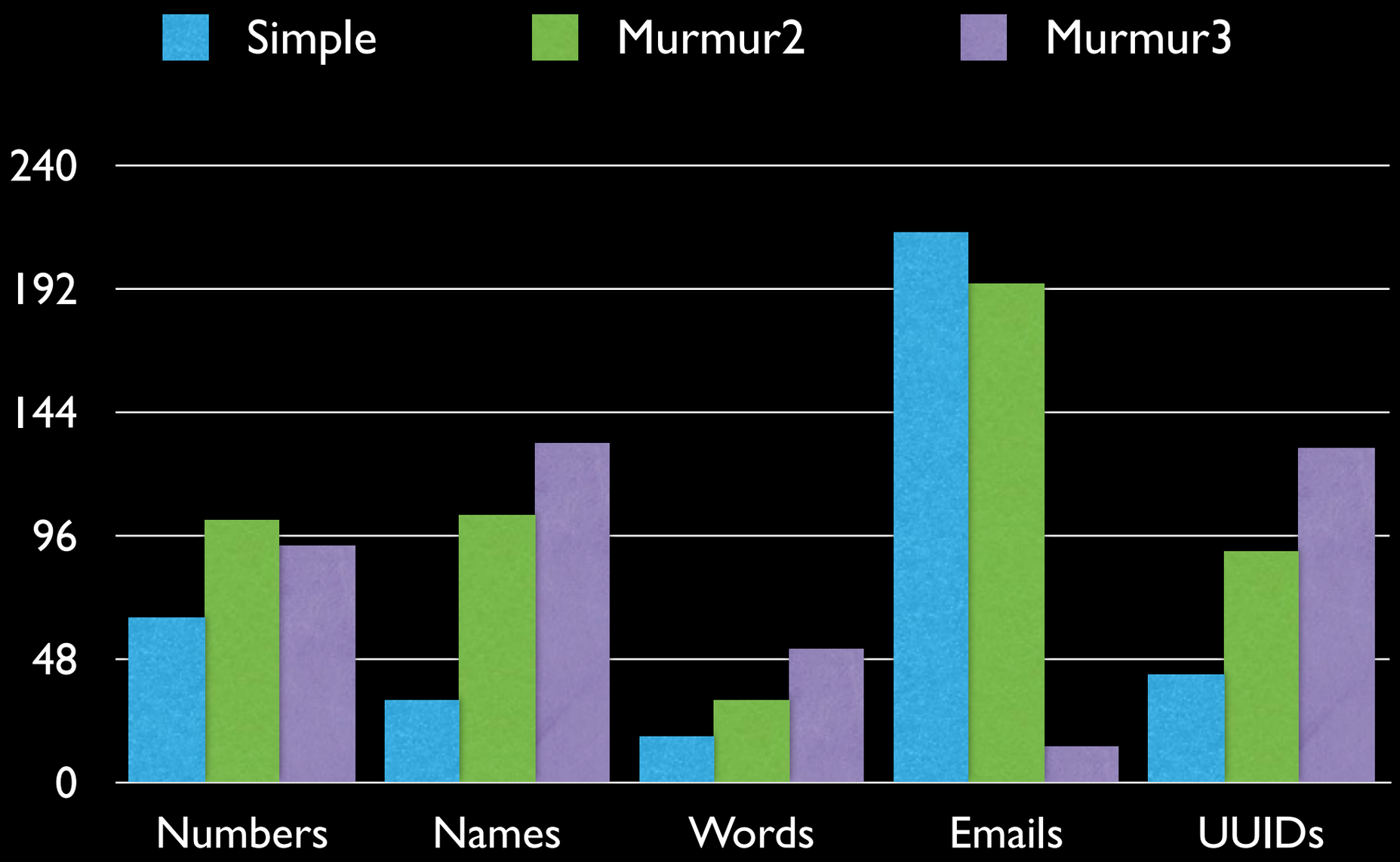
Gah… The critical last letter will now always go to the test name string…

Murmur Hash

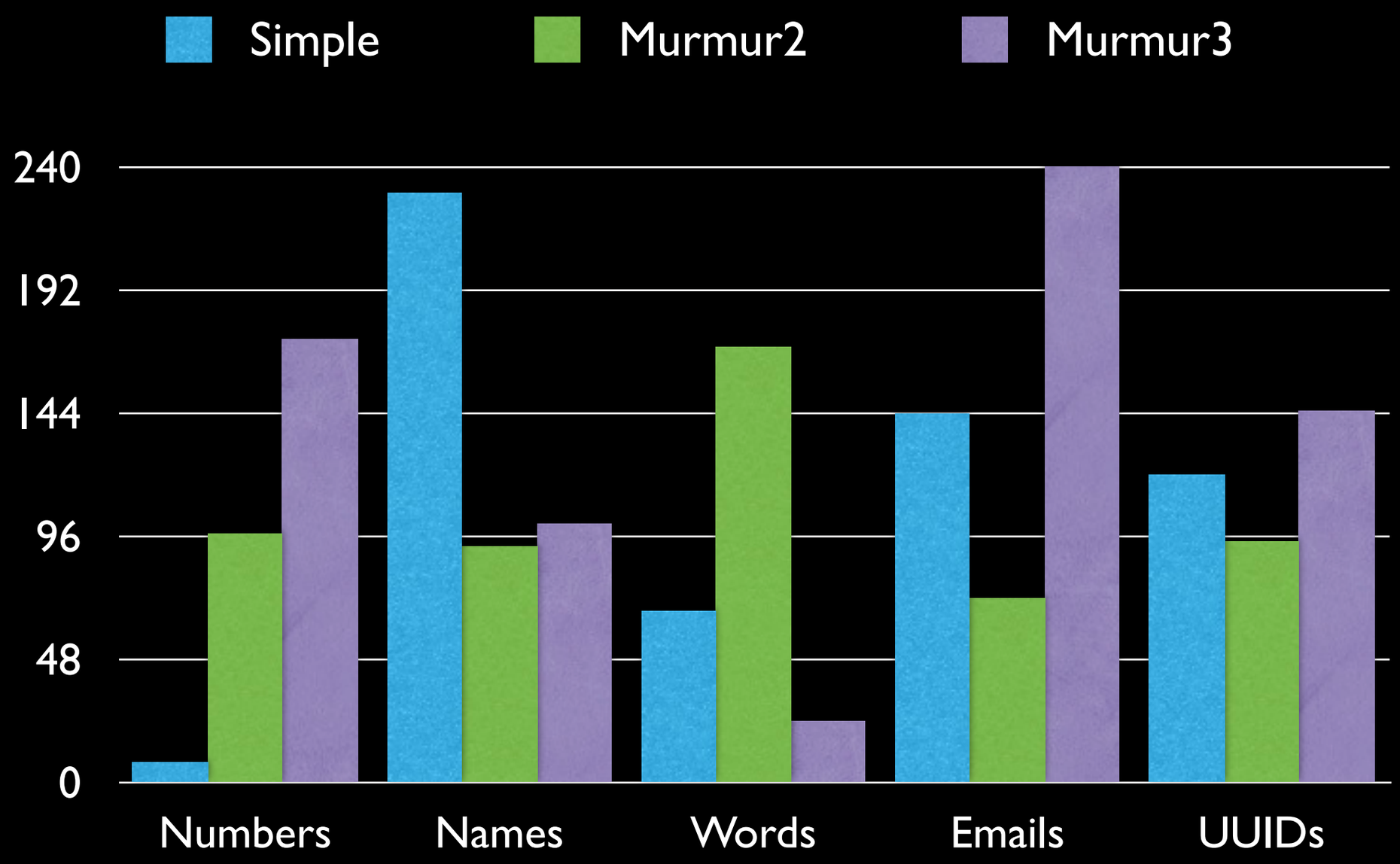
MurmurHash is a non-cryptographic hash function suitable for general hash-based lookup. It was created by Austin Appleby in 2008, and exists in a number of variants, all of which have been released into the public domain. When compared to other popular hash functions, MurmurHash performed well in a *random distribution of regular keys*.

—Wikipedia

No Clear Winner



Split into 3 Groups



Distribution?

- Choose hashing algorithm for your IDs
- Combine ID with Test ID *before hashing*
- Verify your assumptions

Reporting

Reporting

- Client can do the testing, but
- Client must report the results to a server
  - Roll yer own
  - Google Analytics

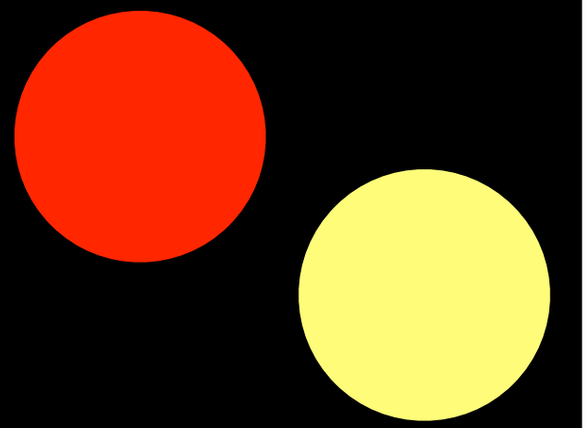
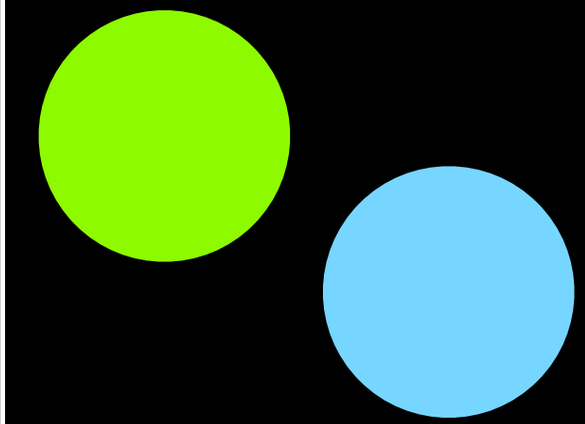
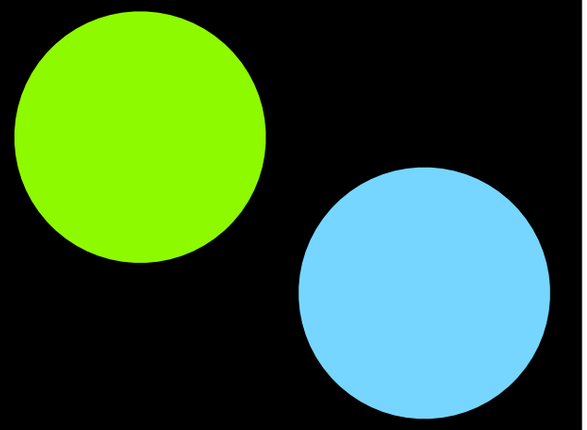
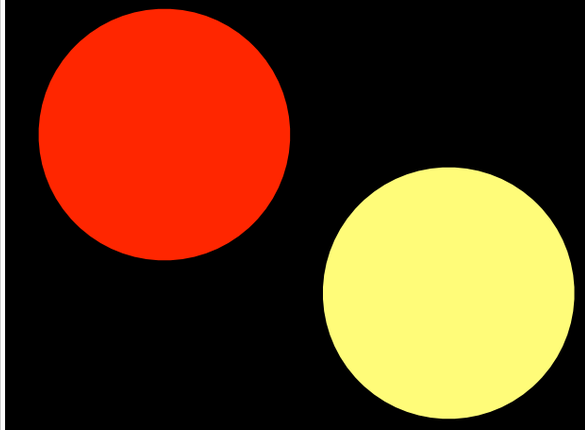
Report Both Events

- Report if the test is shown
- Report if the test is triggered

```
if ( group(userid, "button-test") == 1 ) {
    tracking('shown', 'shiny-red', guid);

    $('#big-button').addClass('shiny-red').click( function(){
        tracking('clicked', 'shiny-red', guid);
    });
}
```

Variations

	Test 1	Test 2
Group A		
Group B		

Slices

Each major test gets "one slice" out of 20:

```
function slice(userID) {
        return hash( userID + "major-slices" ) % 20;
    }
```

Slices

Each major test gets "one slice" out of 20:

```
function slice(userID) {
        return hash( userID + "major-slices" ) % 20;
}


if (slice(userID) == 5)  // 4th slice is our button test
    var group = hash( userID + testName ) % 2;
    // ...
}
```

Epsilon-Greedy

- Called *Multi-arm Bandit*
- 10% of the time, *exploration*:
  - Exploration: choose a random test
- Otherwise, *exploitation*:
  - For each test, calculate its *value* (number of successes / times shown)
  - Use the best test...

```
var tests = [
    {    name: "red",
         show: function() { },
         successes: 1,
         shown: 1
    },
    {    name: "blue",
         show: function() { },
         successes: 1,
         shown: 1
    }
];
```

```
function choose(tests) {
    var bestTest;

    if ( Math.random() < 0.1) {
        // Exploration ... choose a random test to show
        bestTest = tests[ randint(tests.length) ];
    }
    else {
        // Exploitation ... Find best performing test
        var bestValue = 0;
        for (var t in tests) {
            var value = tests[t].successes / tests[t].shown;

            if (value > bestValue) {
                bestTest = tests[t];
                bestValue = value;
            }
        }
    }
    bestTest.shown++;     // increment times test is shown
    // Of course, you have to store test data on the server...
    return show(bestTest);
}
```

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red
10	2/7	29%	1/4	25%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red
10	2/7	29%	1/4	25%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red
10	2/7	29%	1/4	25%	red
11	2/8	25%	1/4	25%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red
10	2/7	29%	1/4	25%	red
11	2/8	25%	1/4	25%	red

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red
10	2/7	29%	1/4	25%	red
11	2/8	25%	1/4	25%	red
12	**2/9**	22%	1/4	25%	

Iteration	Red	Value	Blue	Value	Best Test
1	1/1	100%	1/1	100%	red
2	1/2	50%	1/1	100%	blue
3	1/2	50%	1/2	50%	red
4	1/3	33%	1/2	50%	blue
5	1/3	33%	1/3	33%	red
6	1/4	25%	1/3	33%	blue
7	1/4	25%	1/4	25%	red ★
8	**2/5**	40%	1/4	25%	red
9	2/6	33%	1/4	25%	red
10	2/7	29%	1/4	25%	red
11	2/8	25%	1/4	25%	red
12	**2/9**	22%	1/4	25%	**blue**

Iteration	Red	Value	Blue	Value	Best Test
13	2/9	22%	1/5	20%	red
14	2/10	20%	1/5	20%	red
15	2/11	18%	1/5	20%	blue
16	2/11	18%	1/6	17%	red
17	2/12	17%	1/6	17%	red
18	2/13	15%	1/6	17%	blue
19	2/13	15%	1/7	14%	red
20	2/14	14%	1/7	14%	red
21	2/15	13%	1/7	14%	blue
22	2/15	13%	1/8	13%	red
23	2/16	13%	1/8	13%	red
24	2/17	12%	1/8	13%	blue

Tools

Optimizely

Etsy's *Feature* Project

https://github.com/etsy/feature

```php
$server_config['button_test'] = array(
    'enabled' => array(
        'blue'  => 15,
        'red'   => 15
    ),
);
```

```php
if (Feature::isEnabled('button_test')) {
    switch (Feature::variant('button_test')) {
      case 'blue':
          // ...
          break;
      case 'red':
          // ...
          break;
    }
}
```

Lab Rats

http://github.com/howardabrams/labrats

Step 1. Functions contain Test Group “Feature”

```
function shinyRed() {
    this.addClass('shiny-red');
}

function flashyBlue() {
    this.addClass('flashy-blue');
}
```

Step 2. Call labrats to have “Correct” feature called

```
var buttonTest = { callbacks: [ shinyRed, flashyBlue ] };

$("#logo-test").labrats(buttonTest).click(...);
```

Step 1. Functions contain Test Group “Feature”

```
function shinyRed() {
    this.addClass('shiny-red');
}



function flashyBlue() {
    this.addClass('flashy-blue');
}
```

Step 2. Call labrats to have “Correct” feature called

```
var buttonTest = { callbacks: [ shinyRed, flashyBlue ] };

$("#logo-test").labrats(buttonTest).click(...);
```

Step 1. Functions contain Test Group “Feature”

```
function shinyRed() {
    tracking('shown', 'shiny-red', guid);
    this.addClass('shiny-red').click( function(){
        tracking('clicked', 'shiny-red', guid);
    });
}
function flashyBlue() {
    tracking('shown', 'flashy-blue', guid);
    this.addClass('flashy-blue').click( function(){
        tracking('clicked', 'flashy-blue', guid);
    });
}
```

Step 2. Call labrats to have “Correct” feature called

```
var buttonTest = { callbacks: [ shinyRed, flashyBlue ] };

$("#logo-test").labrats(buttonTest).click(...);
```

Step 1. Functions contain Test Group “Feature”

```
function shinyRed() {
    tracking('shown', 'shiny-red', guid);
    this.addClass('shiny-red').click( function(){
        tracking('clicked', 'shiny-red', guid);
    });
}
function flashyBlue() {
    tracking('shown', 'flashy-blue', guid);
    this.addClass('flashy-blue').click( function(){
        tracking('clicked', 'flashy-blue', guid);
    });
}
```

Step 2. Call labrats to have “Correct” feature called

```
var buttonTest = { name: "Checkout Button",
                   callbacks: [ shinyRed, flashyBlue ] };
$("#logo-test").labrats(buttonTest).click(...);
```

Step 1. Functions contain Test Group “Feature”

```
function shinyRed() {
    tracking('shown', 'shiny-red', guid);
    this.addClass('shiny-red').click( function(){
        tracking('clicked', 'shiny-red', guid);
    });
}
function flashyBlue() {
    tracking('shown', 'flashy-blue', guid);
    this.addClass('flashy-blue').click( function(){
        tracking('clicked', 'flashy-blue', guid);
    });
}
```

Step 2. Call labrats to have “Correct” feature called

```
var buttonTest = { key: guid, name: "Checkout Button",
                   callbacks: [ shinyRed, flashyBlue ] };
$("#logo-test").labrats(buttonTest).click(...);
```

Hashing Algorithm?

```
$.labrats.configure( {
      hash: function(key) {
            return murmurhash3_32_gc(key, 73);
      }
});
```

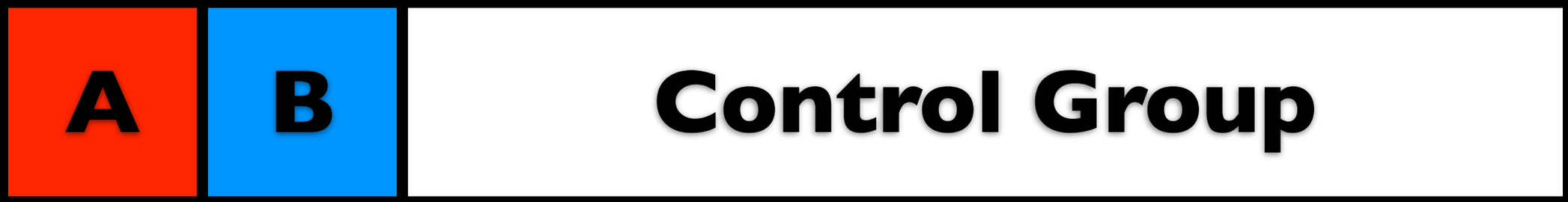
Hashing Algorithm?

```
$.labrats.configure( {
      hash: function(key) {
            return murmurhash3_32_gc(key, 73);
      }
});
```

Seed value

Control Groups

Control Groups



Control Groups

All test groups are in a
10% subset… Rest are control

```
$.labrats( { name: 'Checkout Button', subset: 10,
             callbacks: [ shinyRed, flashyBlue ],
             control: oldButton
} );
```

Control Groups

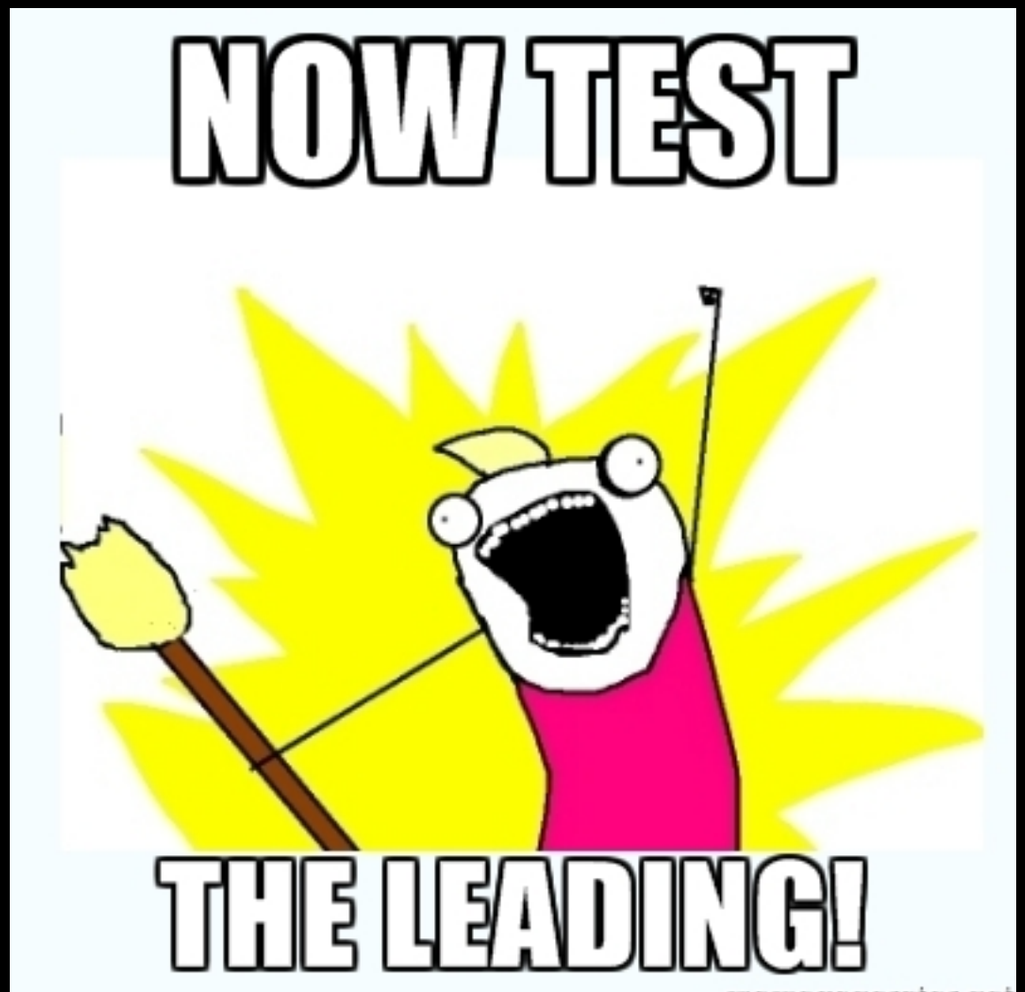
All test groups are in a
10% subset… Rest are control

```
$.labrats( { name: 'Checkout Button', subset: 10,
             callbacks: [ shinyRed, flashyBlue ],
             control: oldButton
} );
```

Tips

Don’t Test Everything

Don’t Test Everything



Calculate Sample Size and Test Duration

Never Test on “Time”

Always test until you reach a specific threshold

Run Test Groups in Parallel

Don’t Surprise Your Reliable Customers

Never put User
in both groups
on the same Test

General

vs.

Targeted

Questions?