[320] Web 5: A/B Testing

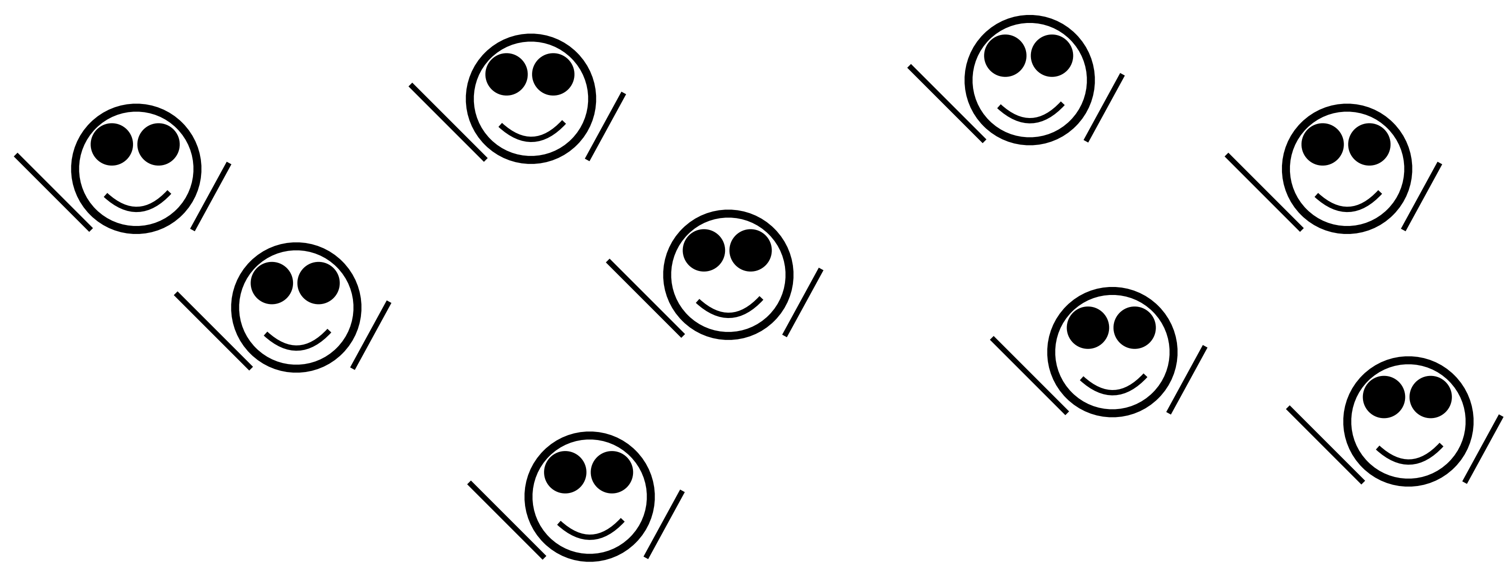
Tyler Caraza-Harter

Source for Examples/Lessons

Ronny Kohavi Keynote Talk at KDD conference (Knowledge Discovery and Data Mining)
Title: Online Controlled Experiments: Lessons from Running A/B/n Tests for 12 years
Video: https://exp-platform.com/kdd2015keynotekohavi/

Experiment Design:
Does Coffee Improve Programming Ability?

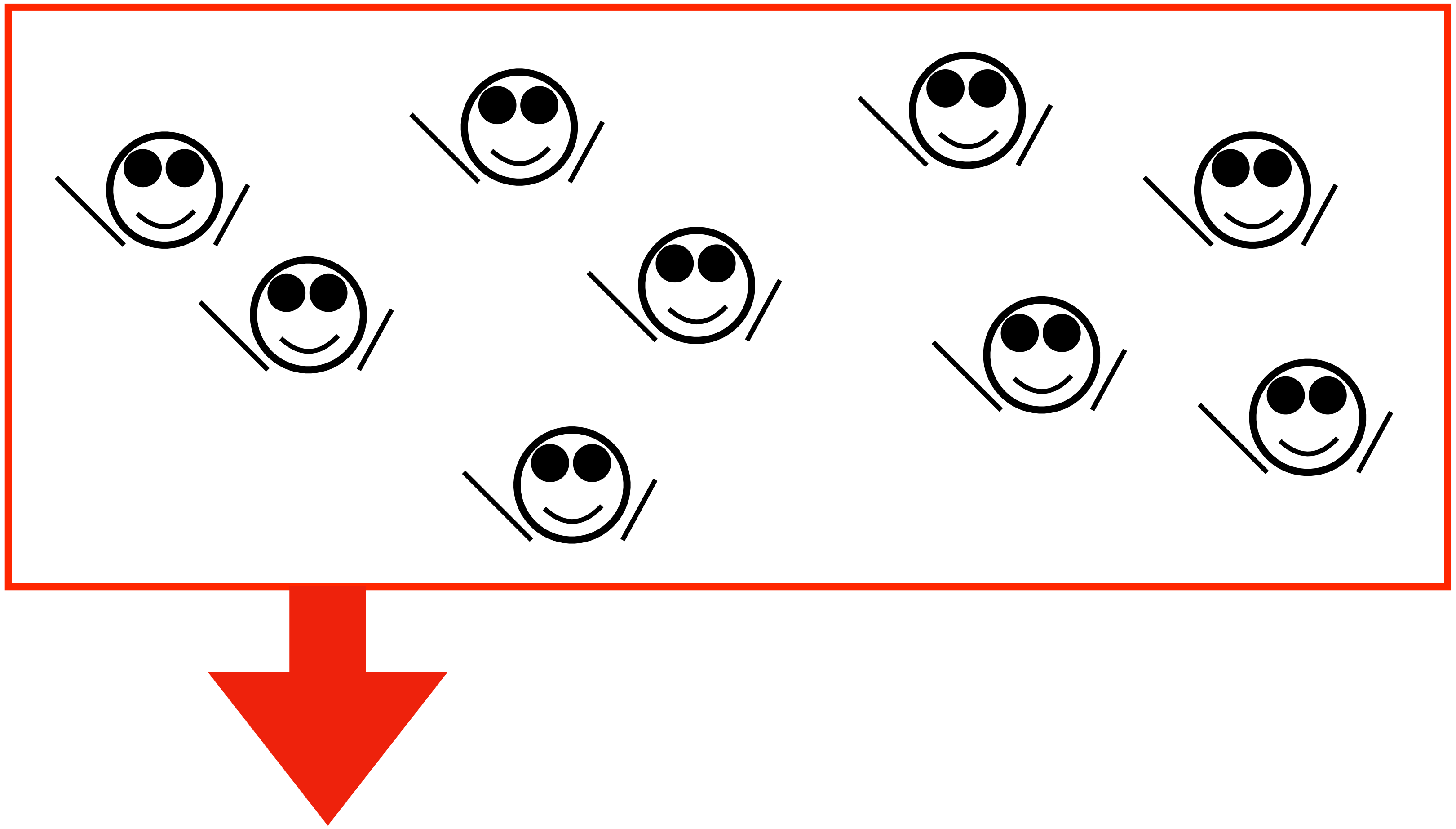
programmers:



Experiment Design:
Does Coffee Improve Programming Ability?

Design 1: before and after

programmers:

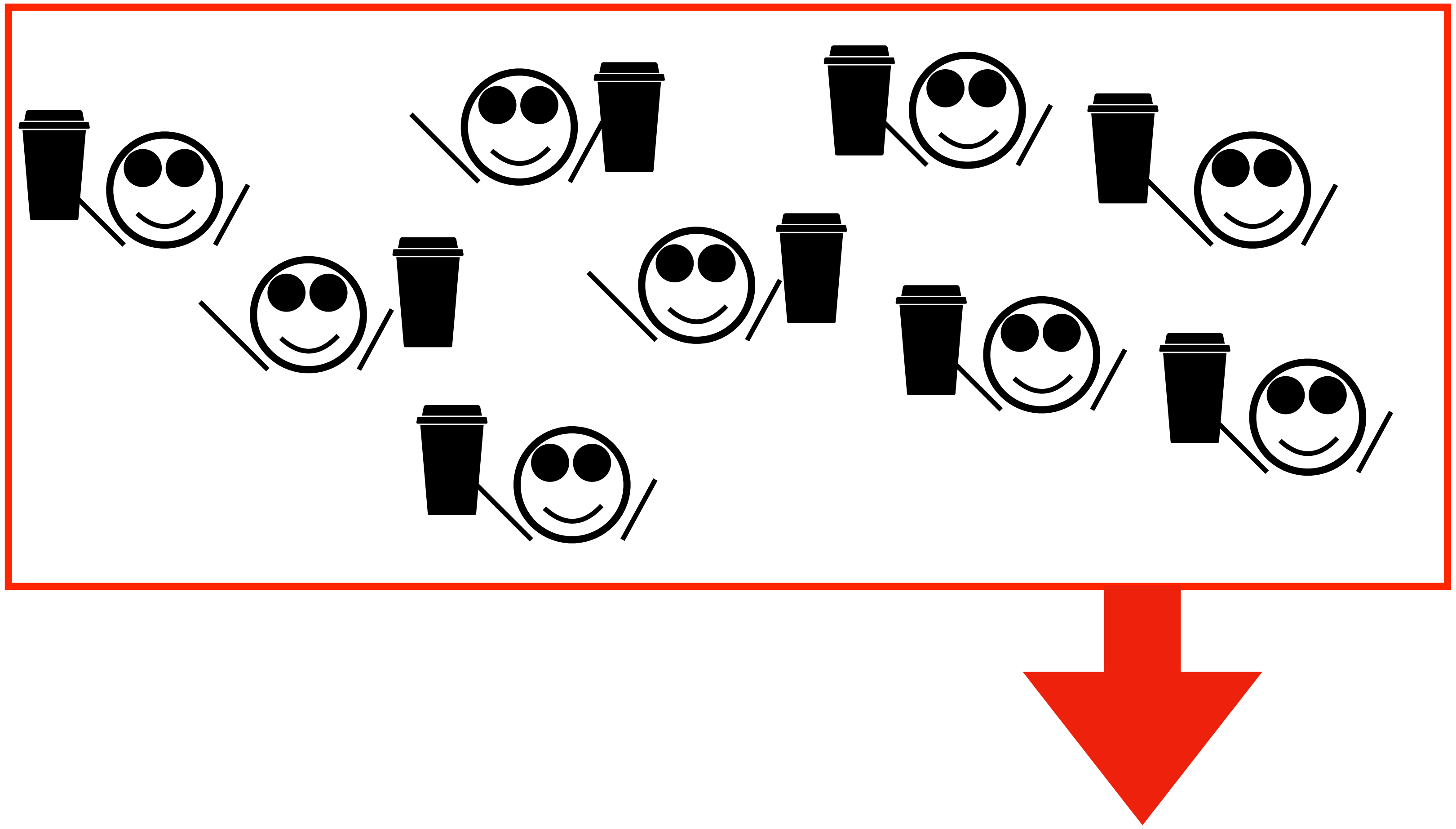


average of 16 hours
for the project before
(no coffee)

Experiment Design:
Does Coffee Improve Programming Ability?

Design 1: before and after

programmers:



average of 16 hours
for the project before
(no coffee)

average of 8 hours
for the project after
(with coffee)

Experiment Design:
Does Coffee Improve Programming Ability?

Design 1: before and after

programmers:

concerns???

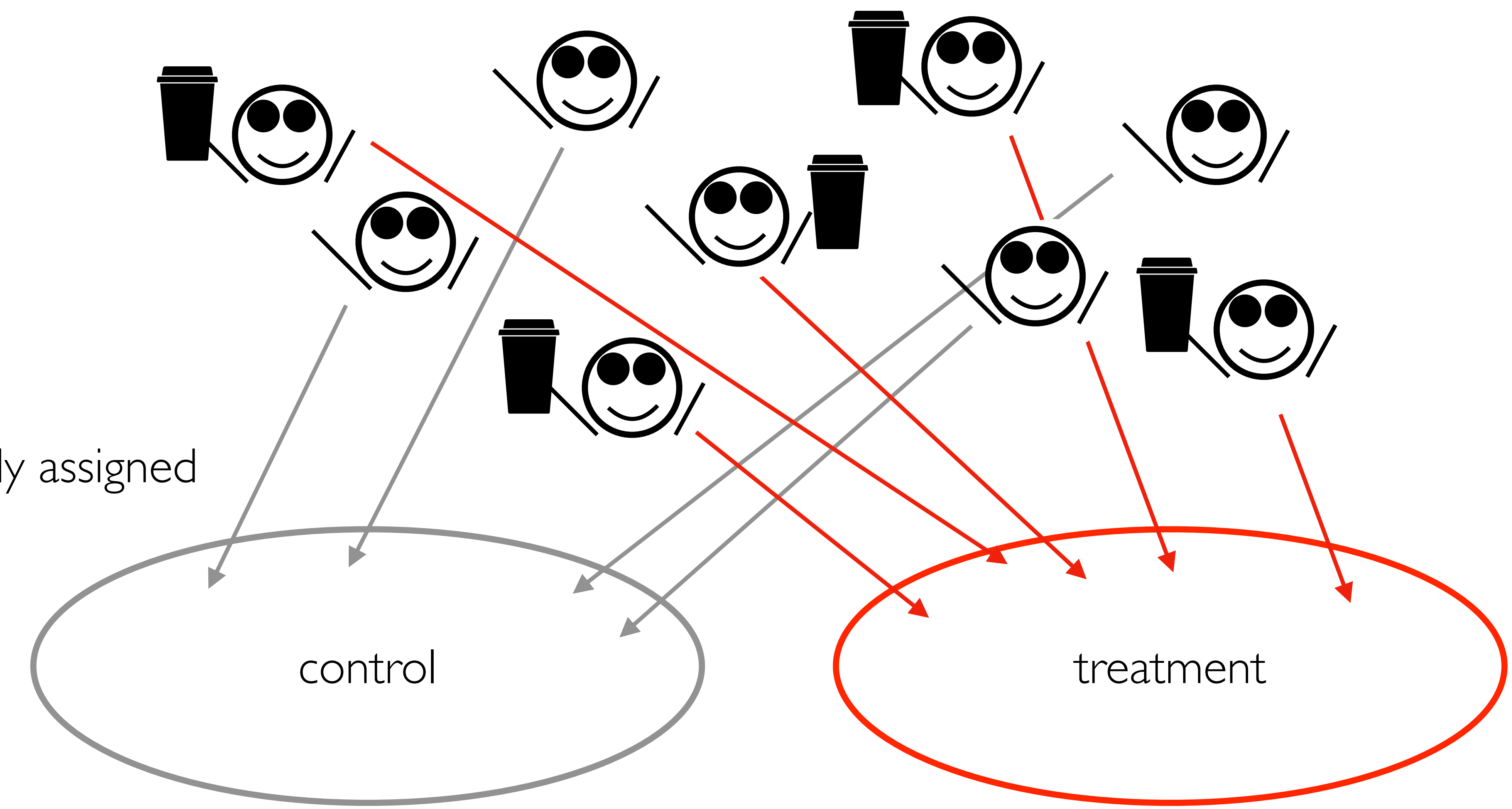
average of 16 hours
for the project before
(no coffee)

average of 8 hours
for the project after
(with coffee)

Experiment Design:
Does Coffee Improve Programming Ability?

Design 2: randomly assigned control and treatment groups

randomly assigned



average 16.3 hours

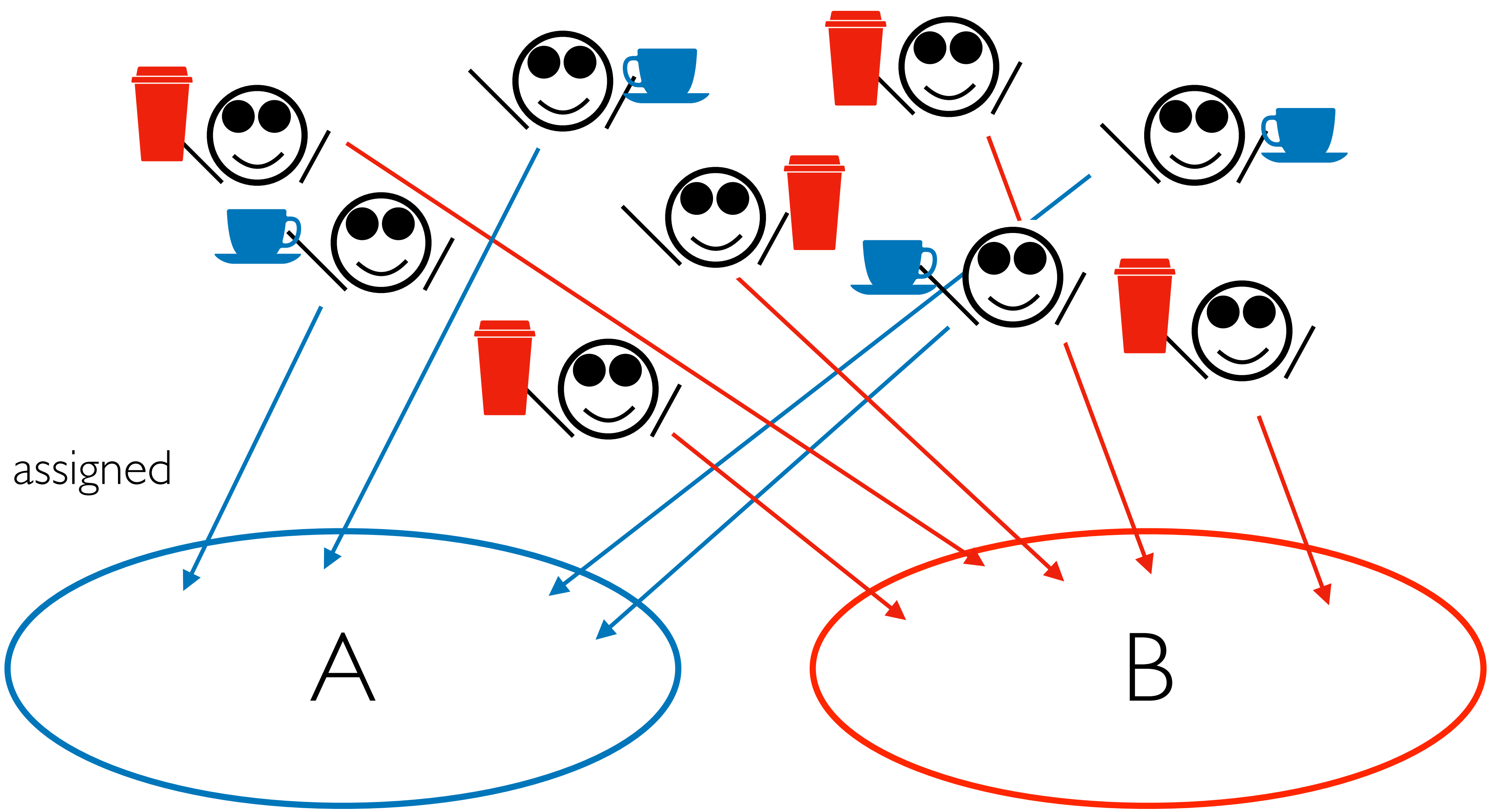
average 15.4 hours

significant difference?

Experiment Design:
Is coffee or tea better for programming?

A/B Testing

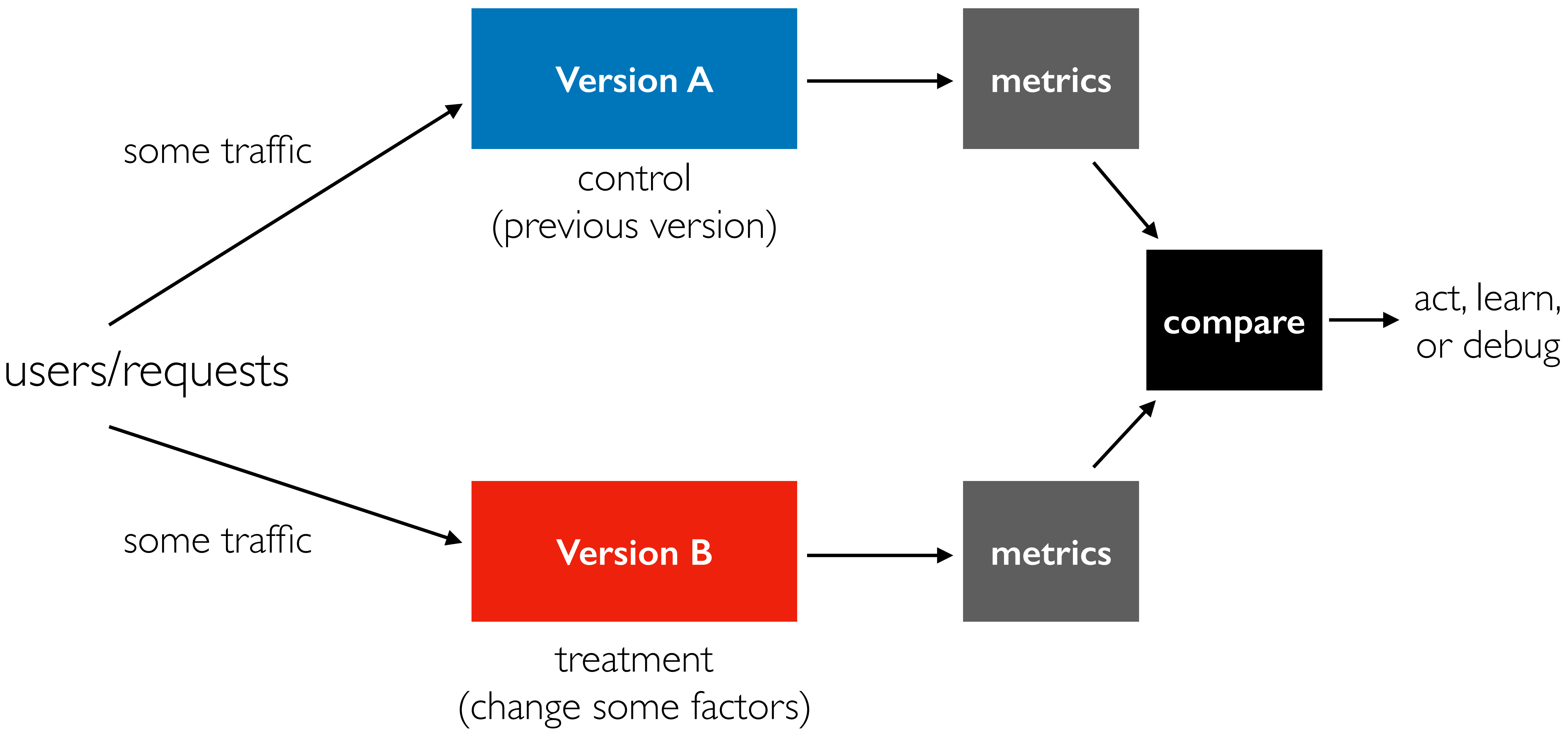
randomly assigned



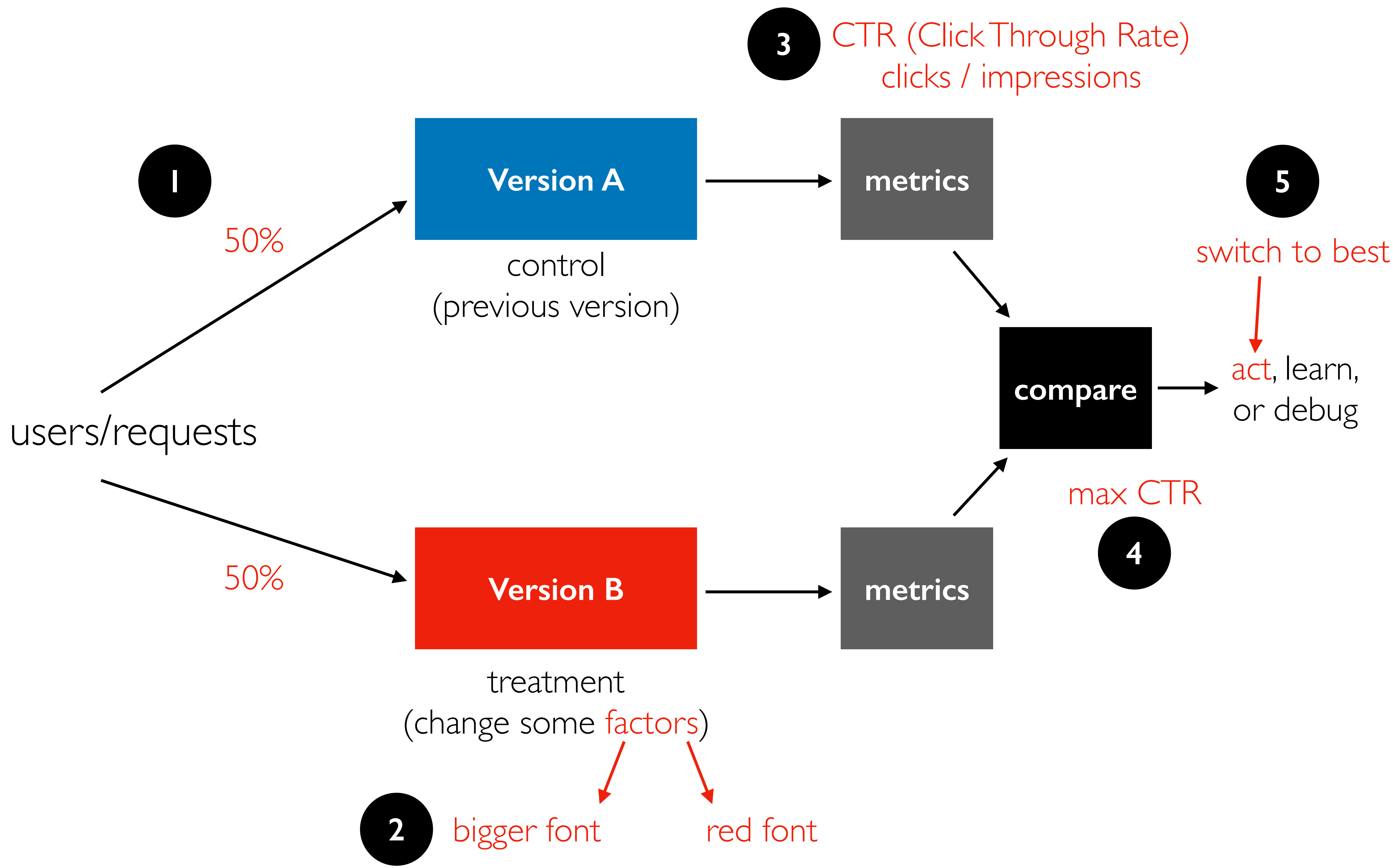
average 16.3 hours

average 15.4 hours

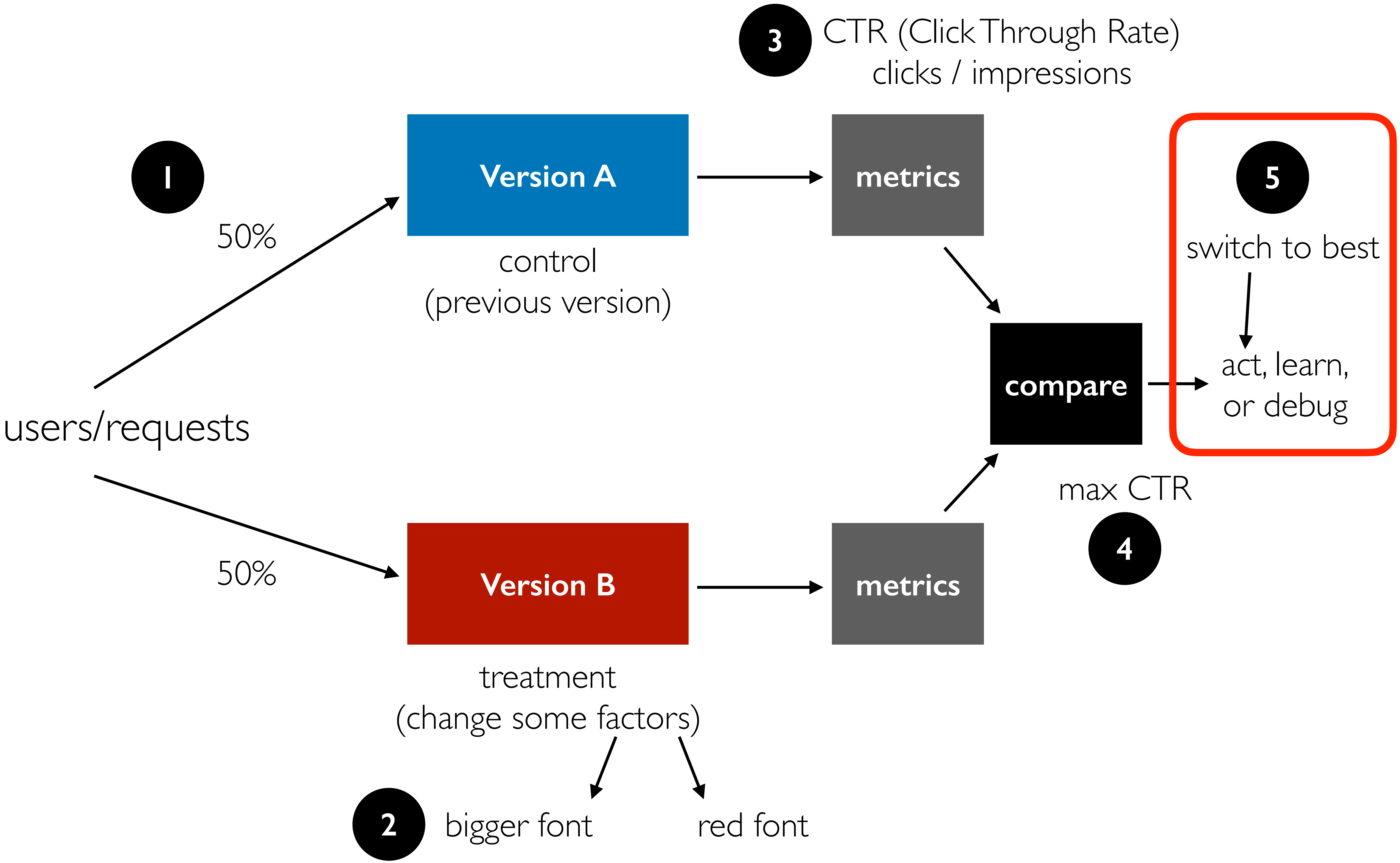
A/B Test Overview (for web applications!)



Example 1: Link to Donation Page

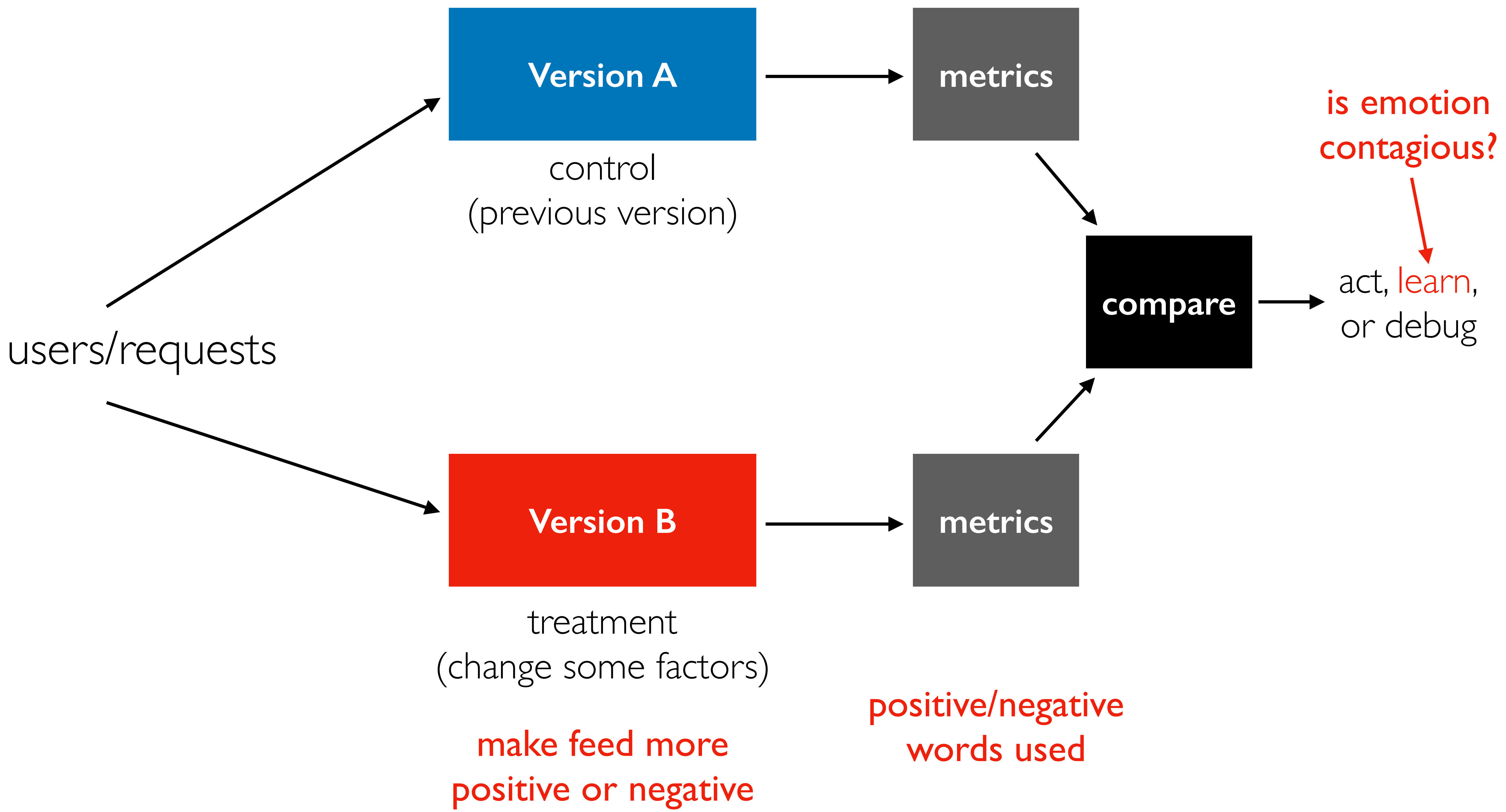


Lecture Outline



Example 2: Facebook Emotional Contagion Study

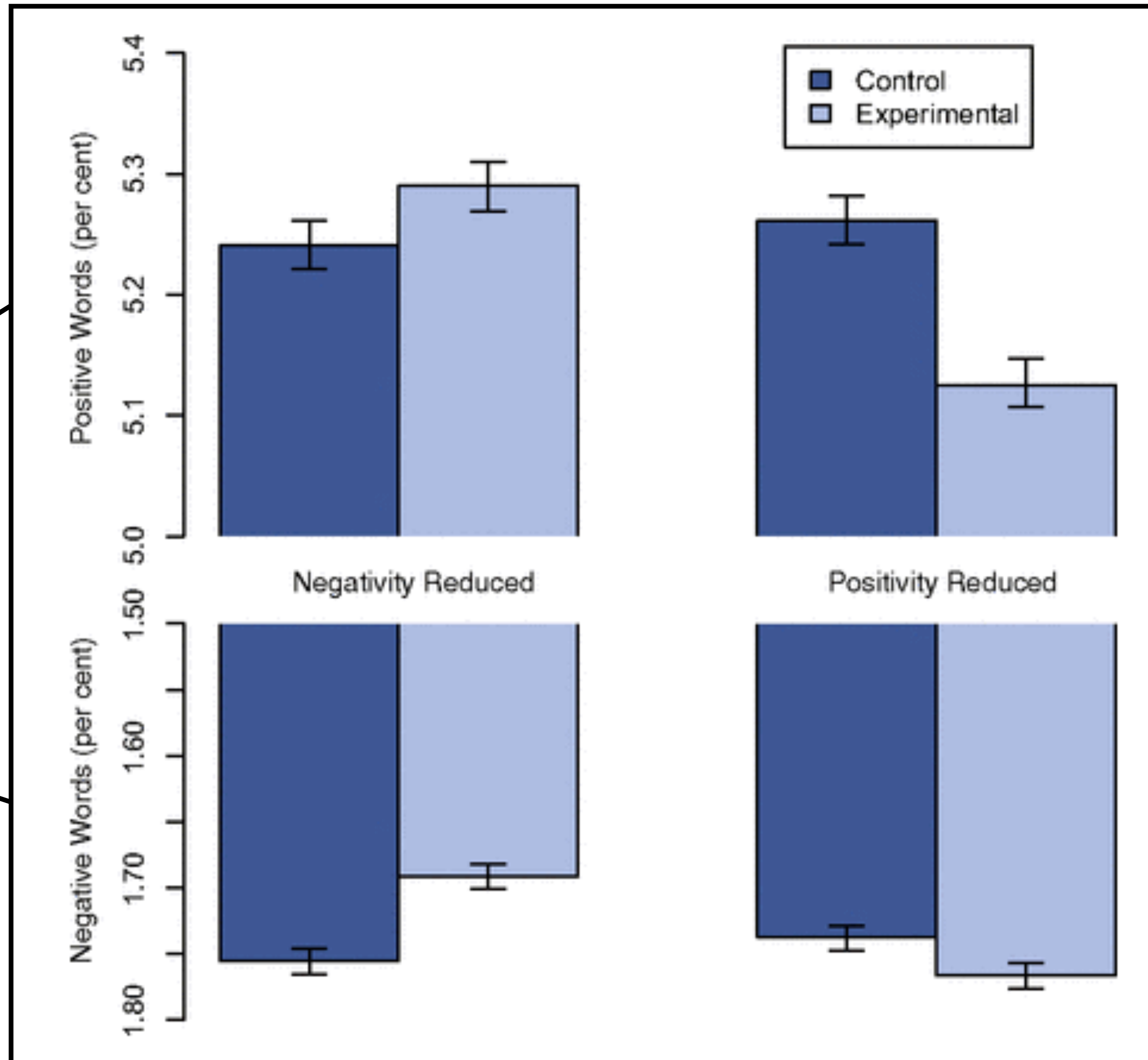
Reading: https://techcrunch.com/2014/06/29/ethics-in-a-data-driven-world/



Example 2: Facebook Emotional Contagion Study

Reading: https://techcrunch.com/2014/06/29/ethics-in-a-data-driven-world/

users/requests



(change some factors)

mpare

is emotion contagious?

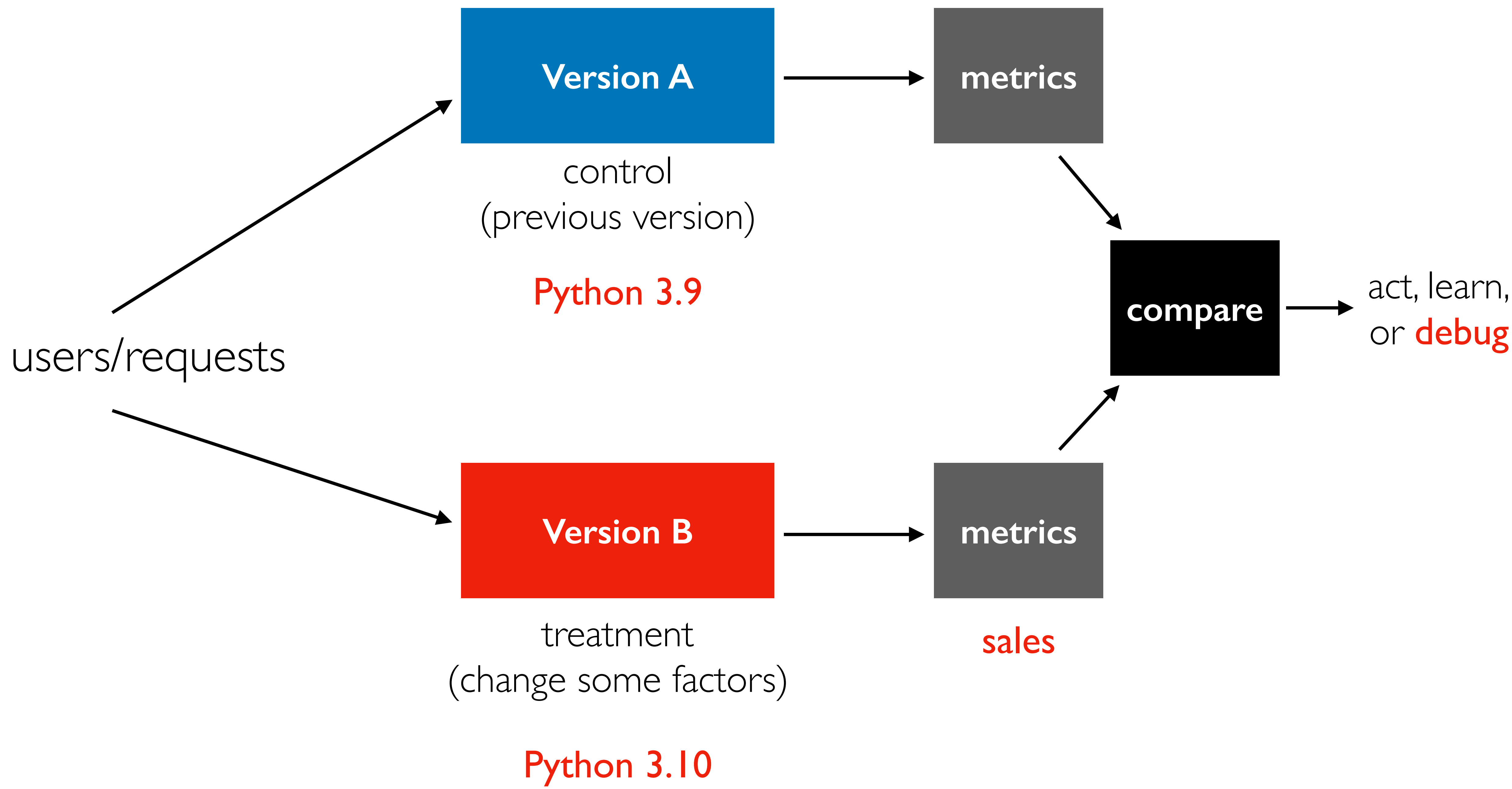
act, learn, or debug

make feed more positive or negative

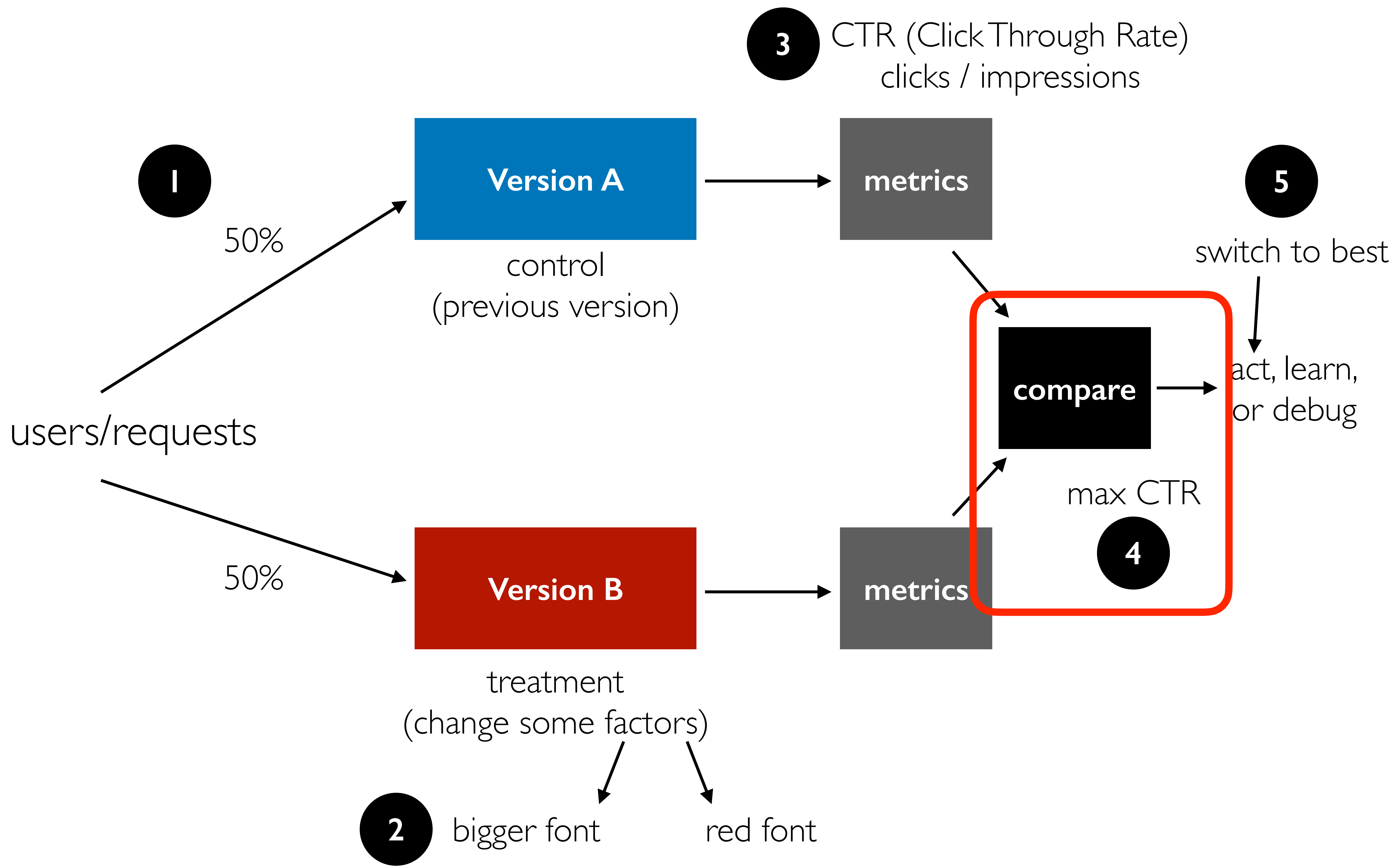
positive/negative words used

didn't need to submit to the IRB (Institutional Review Board) -- *when should it be required?*

Example 3: Update Python Version



Lecture Outline



Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

	click	no-click
A	12	68
B	6	14

df: contingency table

how many B **impressions** were there?
what was B's **CTR**?

Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

	click	no-click
A	12	68
B	6	14

df: contingency table

how many B **impressions** were there? 20
what was B's **CTR**? 6/20 = 30%

Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

	click	no-click
A	12	68
B	6	14

df: contingency table

```
df["click"] / (df["click"] + df["no-click"])
```

```
A    0.15
B    0.30
dtype: float64
```

is the improvement noise?

Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

	click	no-click
A	12	68
B	6	14

```
df["click"] / (df["click"] + df["no-click"])
```

```
A    0.15
B    0.30
dtype: float64
```

df: contingency table

pip3 install scipy

```
import scipy.stats as stats
_, pvalue = stats.fisher_exact(df)
pvalue
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.fisher_exact.html

```
0.1886443478471497
```

Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

p-value is probability of seeing a difference this extreme (or more) if both ratios were generated by the same underlying process (the one most likely to generate this)

	click	no-click
A	12	68
B	6	14

df: contingency table

"**significant**" means p-value is less than some threshold (e.g., 5%)

false positive means it is significant even though underlying process is same

```
import scipy.stats as stats
_, pvalue = stats.fisher_exact(df)
pvalue
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.fisher_exact.html

```
0.18864434784714974
```

Comparisons

out of 200 neutral changes, how many will falsely show up as significant if we set our p-value threshold to 5%?

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

p-value is probability of seeing a difference this extreme (or more) if both ratios were generated by the same underlying process (the one most likely to generate this)

	click	no-click
A	12	68
B	6	14

df: contingency table

"significant" means p-value is less than some threshold (e.g., 5%)

false positive means it is significant even though underlying process is same

```
import scipy.stats as stats
_, pvalue = stats.fisher_exact(df)
pvalue
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.fisher_exact.html

```
0.18864434784714 97
```

Comparisons

out of 200 neutral changes, how many will falsely show up as significant if we set our p-value threshold to 5%?

10

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

p-value is probability of seeing a difference this extreme (or more) if both ratios were generated by the same underlying process (the one most likely to generate this)

	click	no-click
A	12	68
B	6	14

df: contingency table

"significant" means p-value is less than some threshold (e.g., 5%)

false positive means it is significant even though underlying process is same

```
import scipy.stats as stats
_, pvalue = stats.fisher_exact(df)
pvalue
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.fisher_exact.html

0.1886443478471497

Comparisons

out of 200 neutral changes, how many will falsely show up as significant if we set our p-value threshold to 5%?

10

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

occasionally run A/A tests to make sure the system is working (false positive rate should be as expected)

	click	no-click
A	12	68
B	6	14

df: contingency table

```
import scipy.stats as stats
_, pvalue = stats.fisher_exact(df)
pvalue
```

https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.fisher_exact.html

```
0.1886443478471497
```

Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

	click	no-click
A	12	68
B	6	14

df: contingency table

3 outcomes, based on CTRs and significance

- A is significantly better
- B is significantly better
- *neither wins*

what to do?

- ideas???

Comparisons

Example Metric: **CTR** (Click-Through Rate)

CTR = clicks / impressions

"Impression" means user saw it

	click	no-click
A	12	68
B	6	14

df: contingency table

3 outcomes, based on CTRs and significance

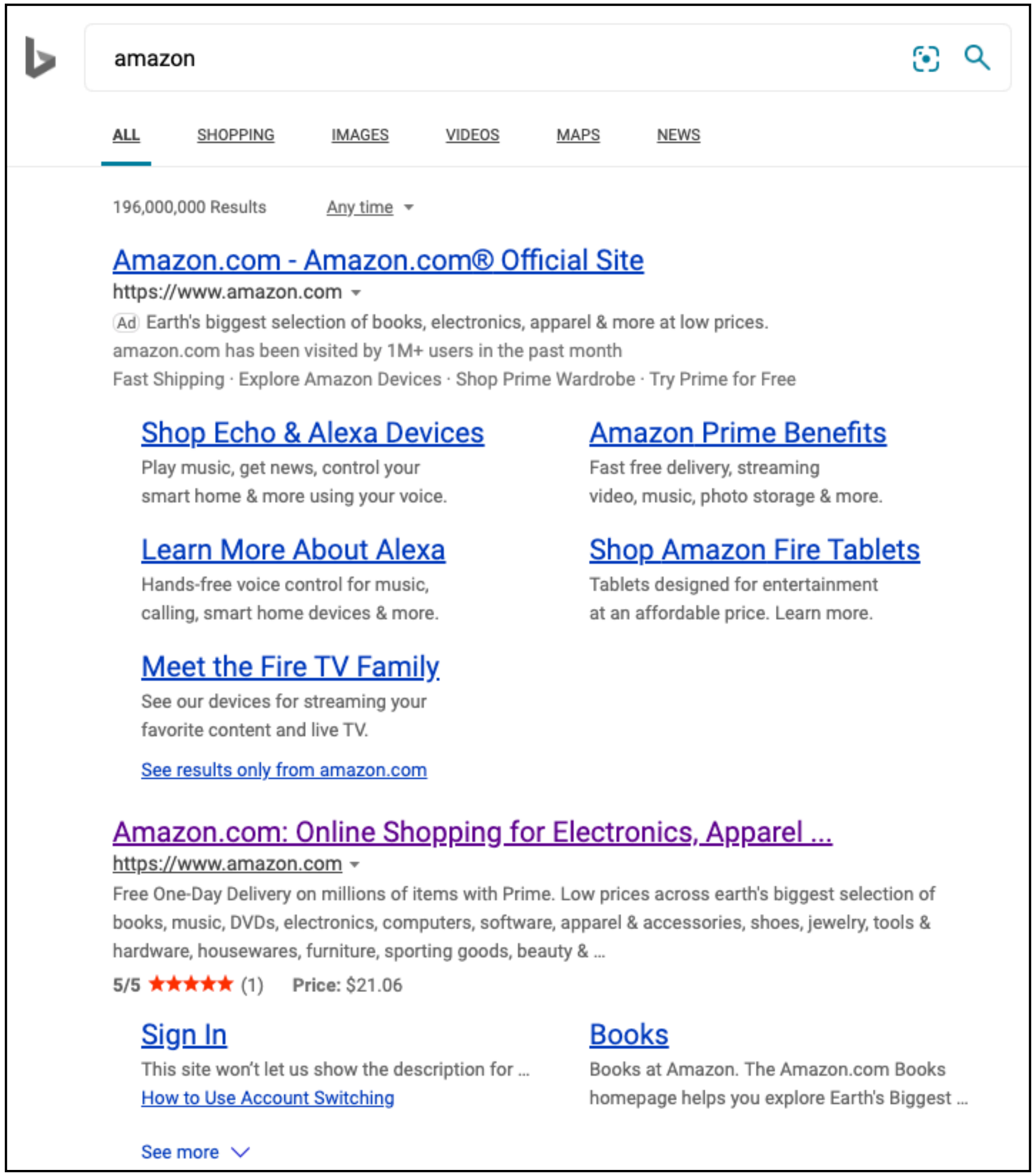
- A is significantly better
- B is significantly better
- *neither wins*

what to do?

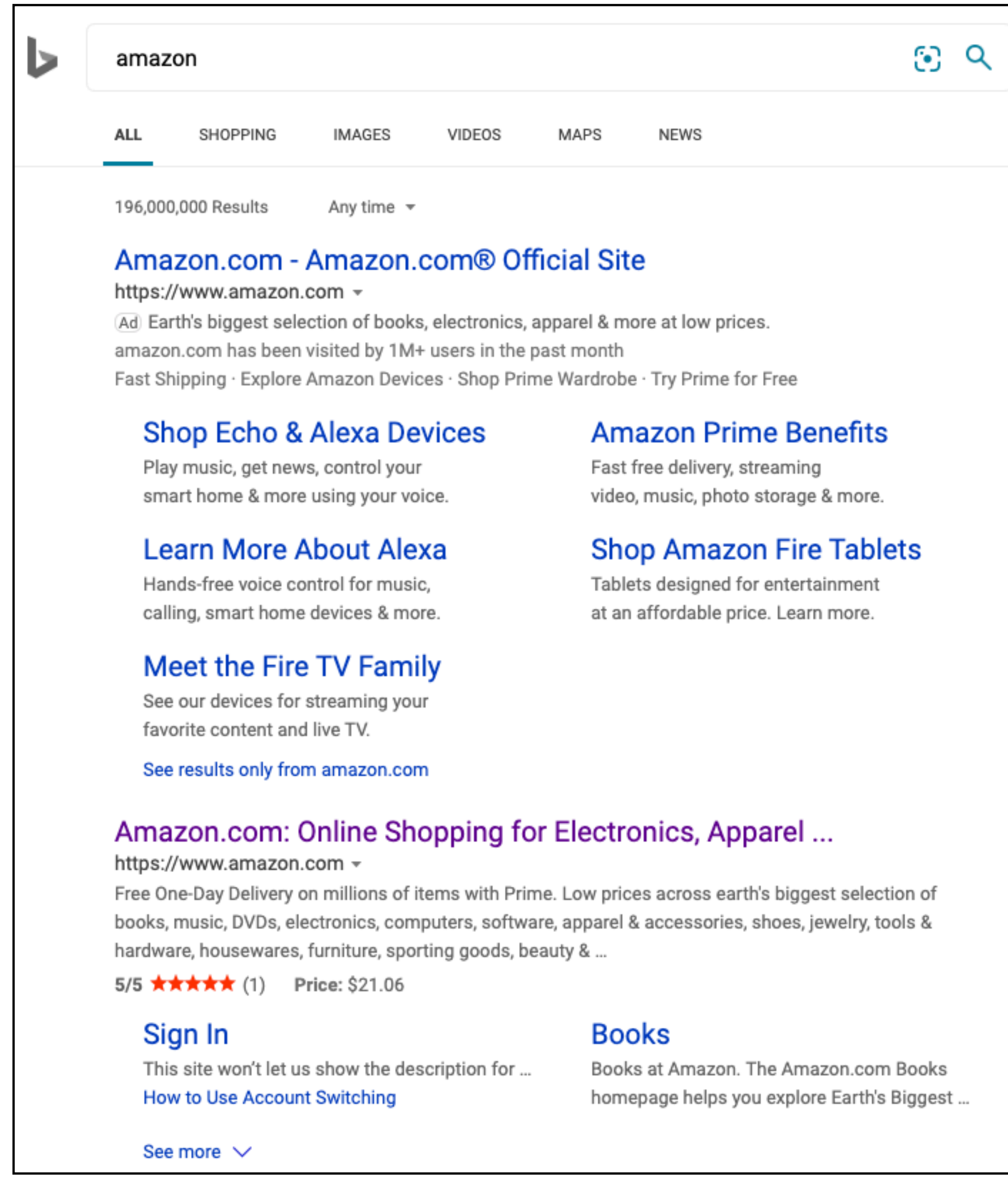
- collect more data
- ignore significance, just look at CTR (indecision may be the worst decision)
- choose previous version A (probably fewer bugs)
- choose new version B (for simplicity or other merits)

Which Version Has Higher Whole-page CTR?

Version A



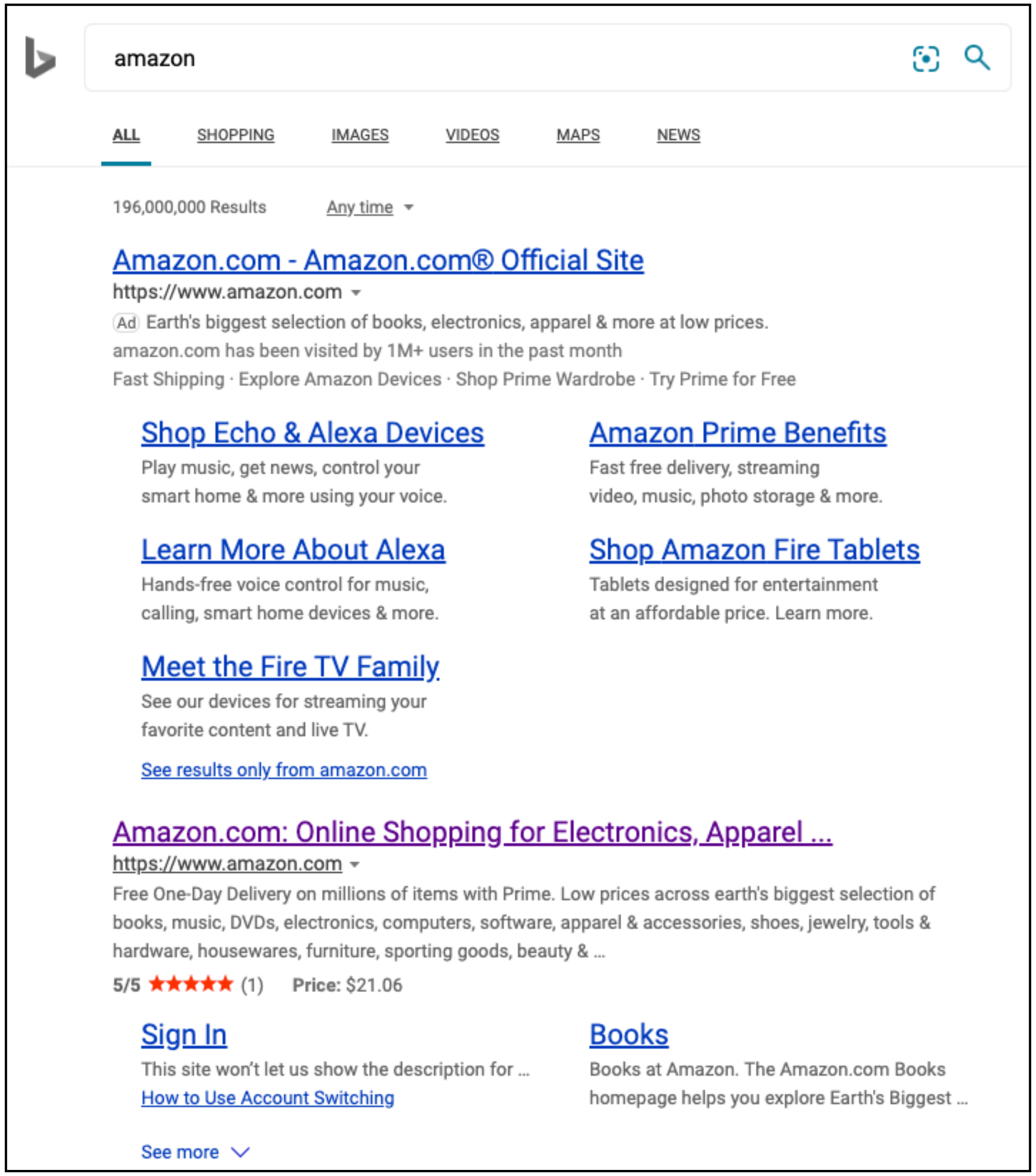
Version B



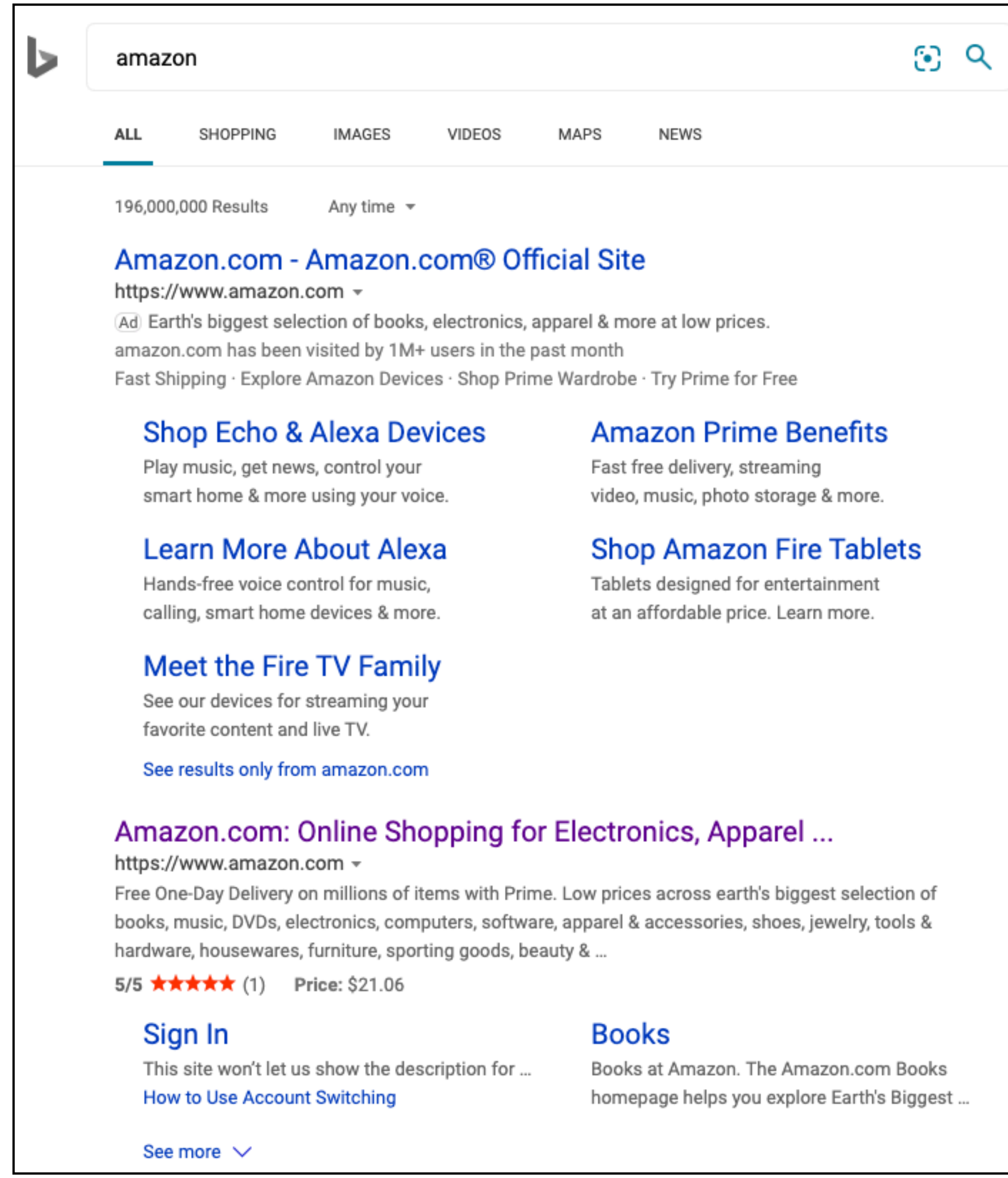
https://youtu.be/qtboCGd_hTA?t=2873

Which Version Has Higher Whole-page CTR?

Version A



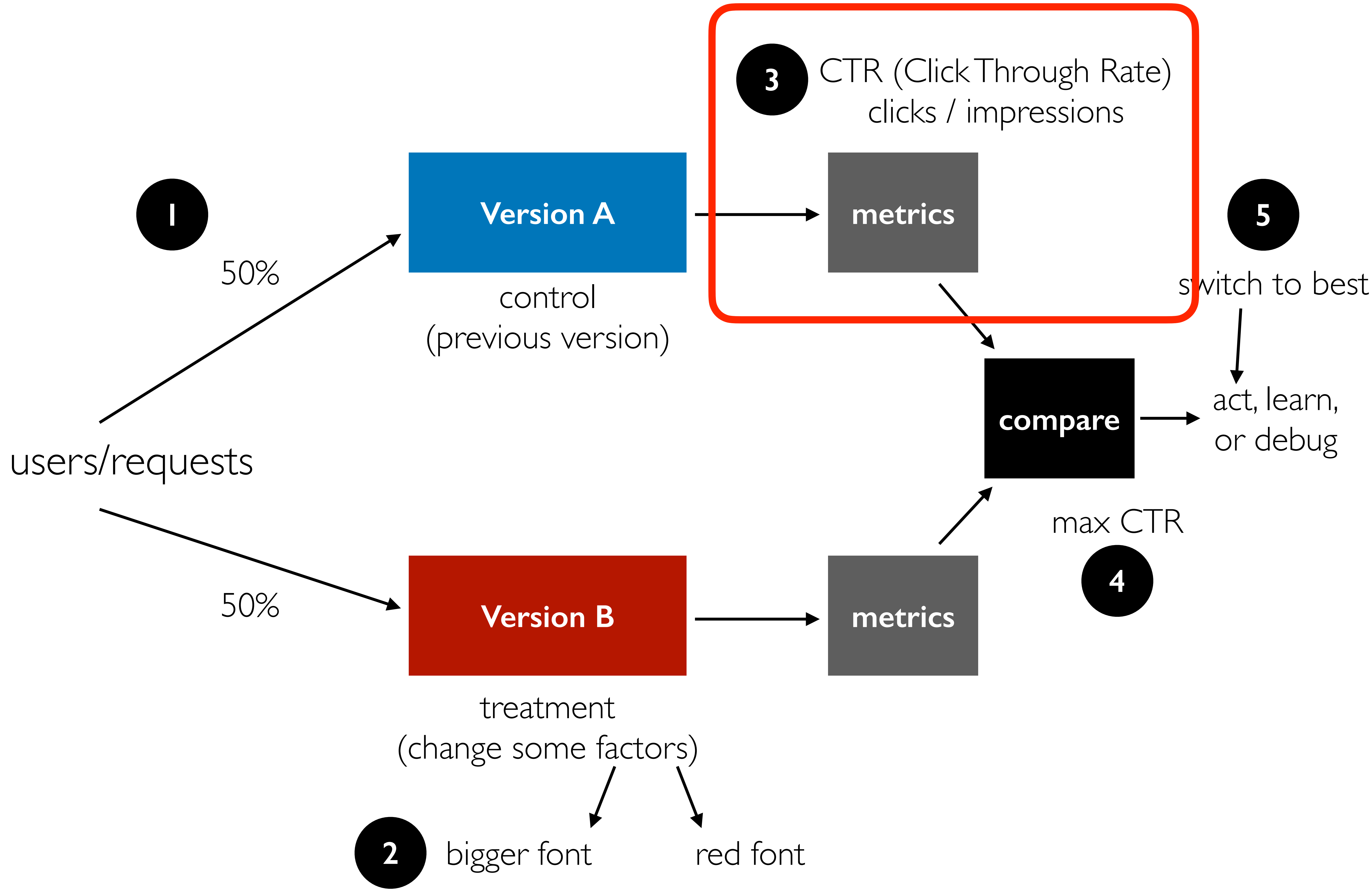
Version B



Lesson: metrics should inform humans, not directly determine decisions

https://youtu.be/qtboCGd_hTA?t=2873

Lecture Outline



Metrics

Things to measure:

- clicks -- when are they bad?

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- other ideas?

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

combos: Bing measures how often people click a result link and don't hit back within 30 seconds

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

combos: Bing measures how often people click a result link and don't hit back within 30 seconds

what is the effect of B?
B is **send twice as many spammy emails**

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

combos: Bing measures how often people click a result link and don't hit back within 30 seconds

what is the effect of B?
B is **send twice as many spammy emails**

what is the effect of B?
B is **remove price from product page link**

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

combos: Bing measures how often people click a result link and don't hit back within 30 seconds

what is the effect of B?
B is **remove price from product page link**

Lesson: it's easy to shift clicks

what is the effect of B?
B is **send twice as many spammy emails**

Lesson: it's hard to measure long-term effects (noisy!), so use common sense

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

combos: Bing measures how often people click a result link and don't hit back within 30 seconds

what is the effect of B?
B is **send twice as many spammy emails**

what is the effect of B?
B is **remove price from product page link**

Decide beforehand on one OEC metric: Overall Experiment Criterion

- Bing has thousands of debug metrics, but only 4 OECs.

Metrics

Things to measure:

- clicks
- scroll (did they read it?)
- subscribe/unsubscribe
- purchases/returns
- hover (did they think about it?)
- shares
- likes/upvotes
- comments

combos: Bing measures how often people click a result link and don't hit back within 30 seconds

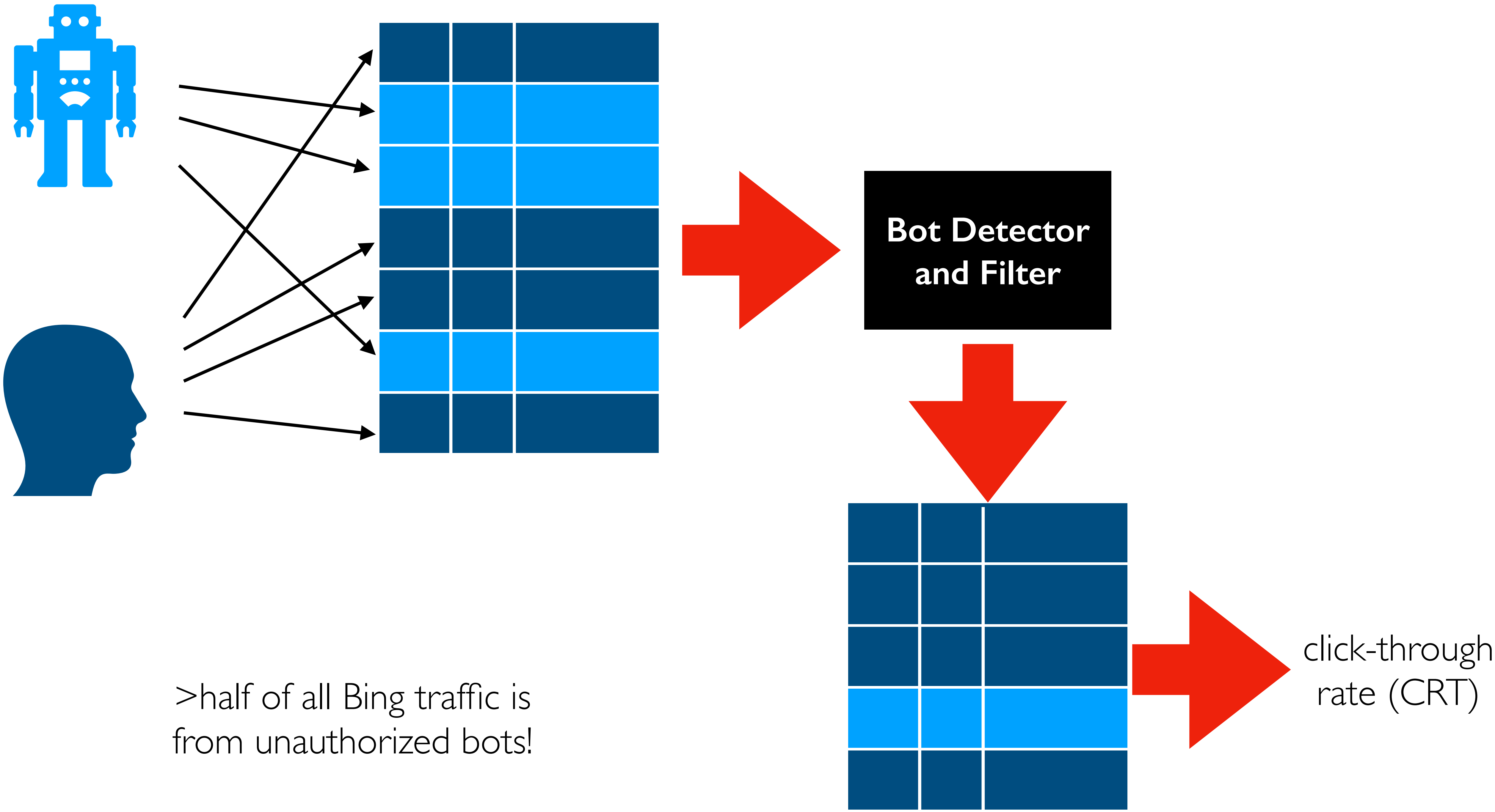
what is the effect of B?
B is **send twice as many spammy emails**

what is the effect of B?
B is **remove price from product page link**

Decide beforehand on one OEC metric: Overall Experiment Criterion

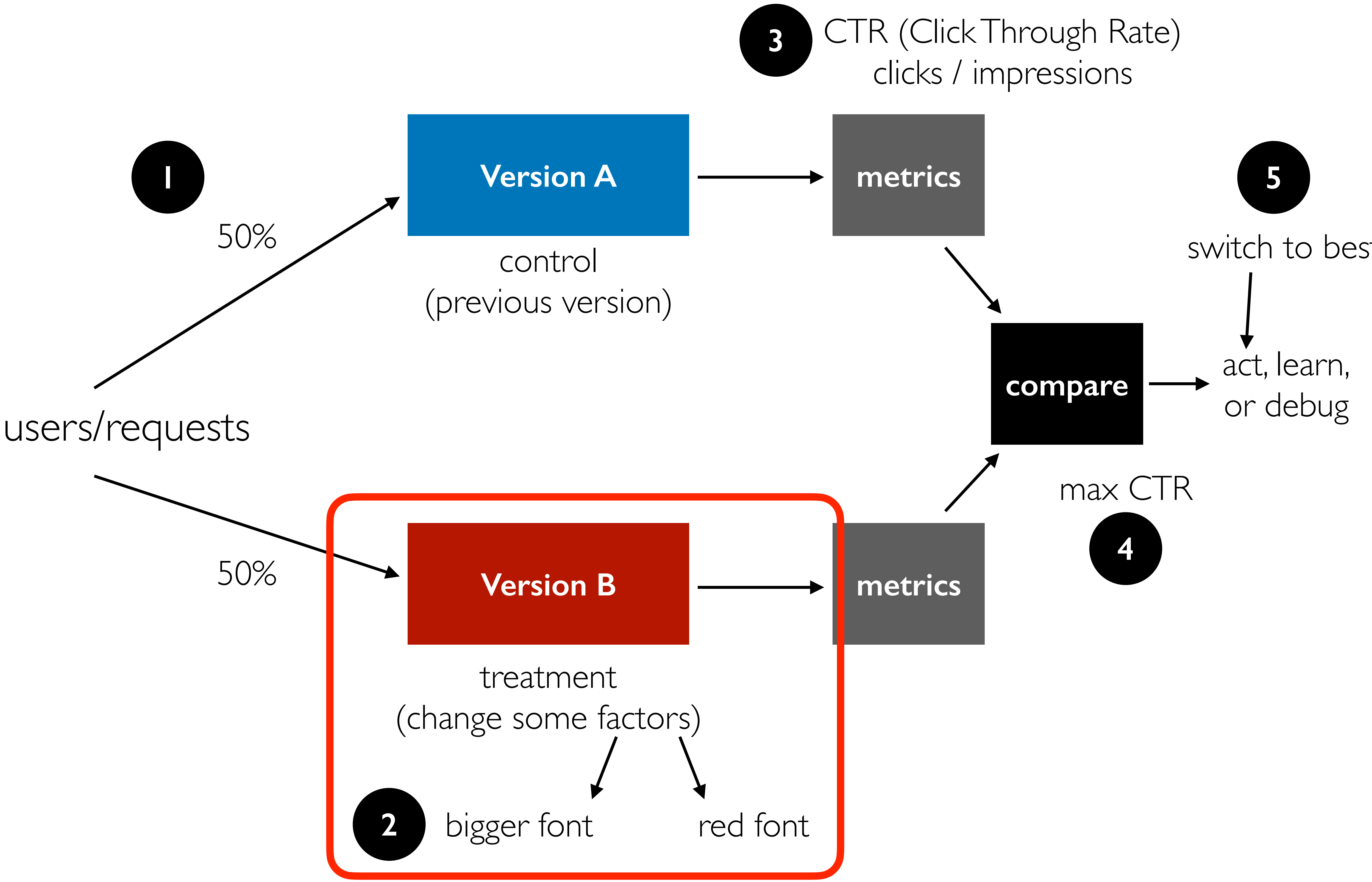
- Bing has thousands of debug metrics, but only 4 OECs. Try to consider cost as well as benefit!
- As a rule of thumb, "*if you make something bigger, more people will click on it*" ~ Ron Kovani
- Making part of the site better could hurt other parts if you have a naive OEC

Metrics Should be on Uniformly Cleaned Data



>half of all Bing traffic is from unauthorized bots!

Lecture Outline



Treatment

Run two variants side by side: control (A) and treatment (B)

Treatment consists of one or more factors changed:

- wording
- slowdown
- changes "invisible" to user (e.g., software updates)
- what else?

Treatment

Run two variants side by side: control (A) and treatment (B)

Treatment consists of one or more factors changed:
- wording
- slowdown
- changes "invisible" to user (e.g., software updates)
- time of day (for emails sent)
- font, size, color, icons, graphic design in general
- recommendation algorithm used
- sequence of steps necessary to make a purchase
- database that is faster for some queries (and slower for others)

Treatment

Run two variants side by side: control (A) and treatment (B)

Treatment consists of one or more factors **changed:**

- wording
- slowdown
- changes "invisible" to user (e.g., software updates)
- time of day (for emails sent)
- font, size, color, icons, graphic design in general
- recommendation algorithm used
- sequence of steps necessary to make a purchase
- database that is faster for some queries (and slower for others)

many experiments are big time investments (require significant coding)!

Lesson: don't be too attached to your work, be redundant and ready to throw things away

Treatment

Run two variants side by side: control (A) and treatment (B)

Treatment consists of one or more factors changed:

- wording
- slowdown
- changes "invisible" to user (e.g., software updates)
- time of day (for emails sent)
- font, size, color, icons, graphic design in general
- recommendation algorithm used
- sequence of steps necessary to make a purchase
- database that is faster for some queries (and slower for others)

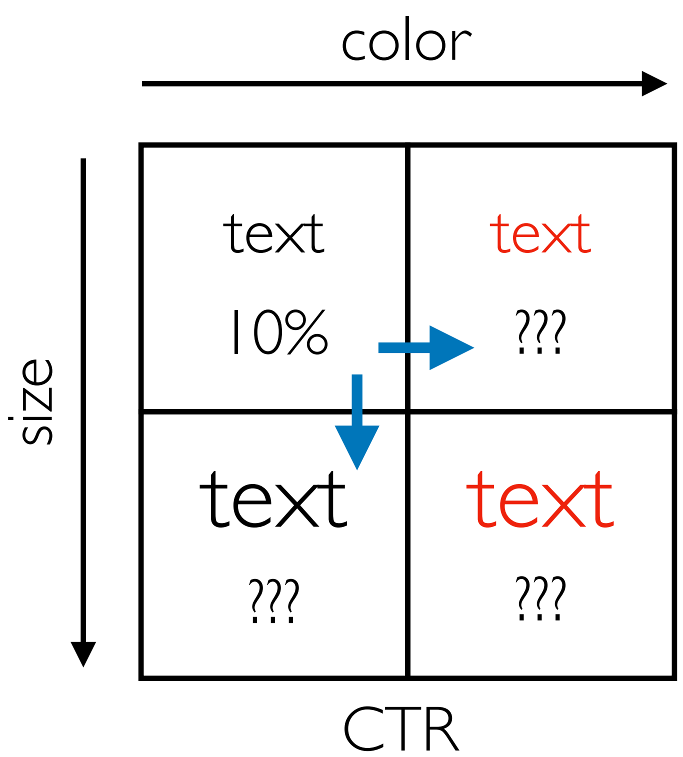
many experiments are big time investments (require significant coding)!

Lesson: don't be too attached to your work, be redundant and ready to throw things away

there's also plenty of low-hanging fruit!

"stop debating, it's easier to get the data" ~ Ron Kohavi

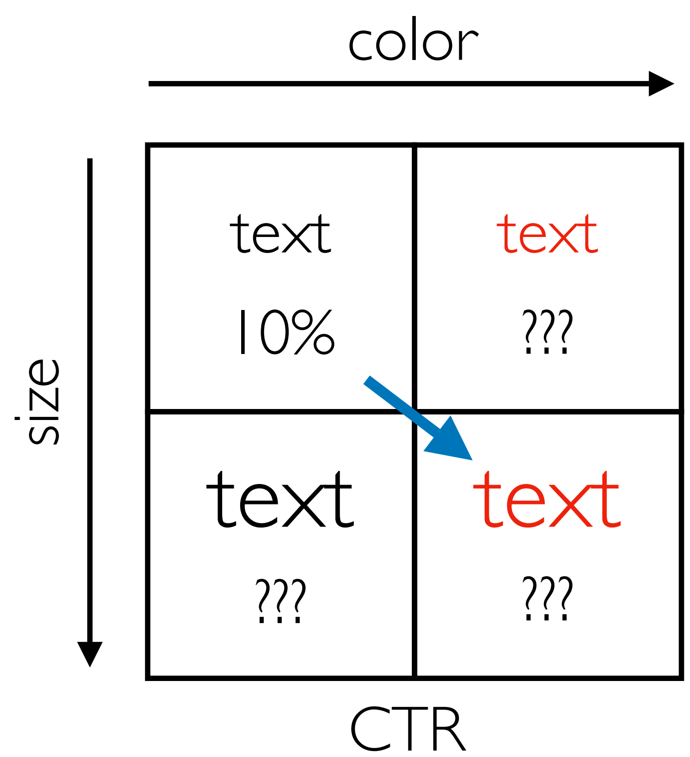
Finding the Best Combination



Option 1: OFAT (one factor at a time)

Hypothesis: large red font will be better

Finding the Best Combination

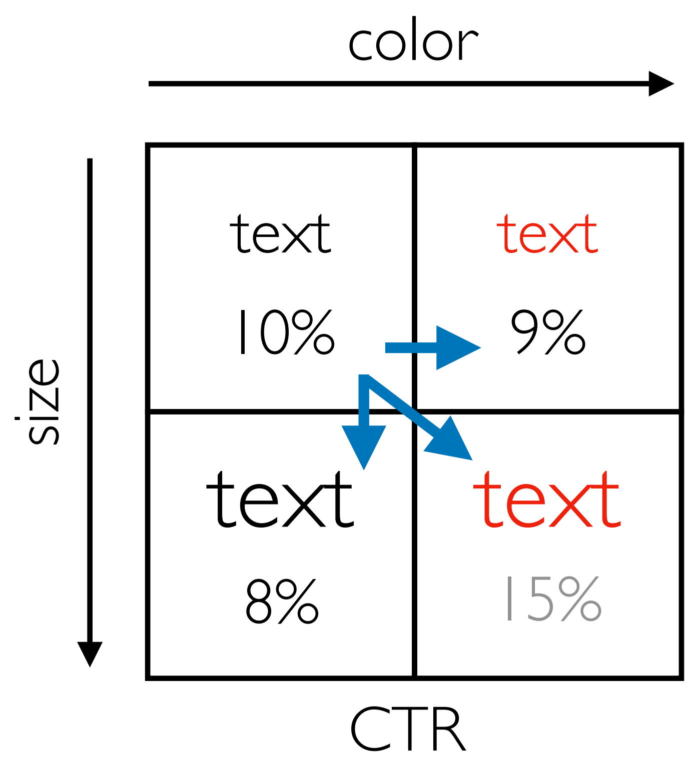


Hypothesis: large red font will be better

Option 1: OFAT (one factor at a time)

Option 2: introduce two factors at once

Finding the Best Combination



Hypothesis: large red font will be better

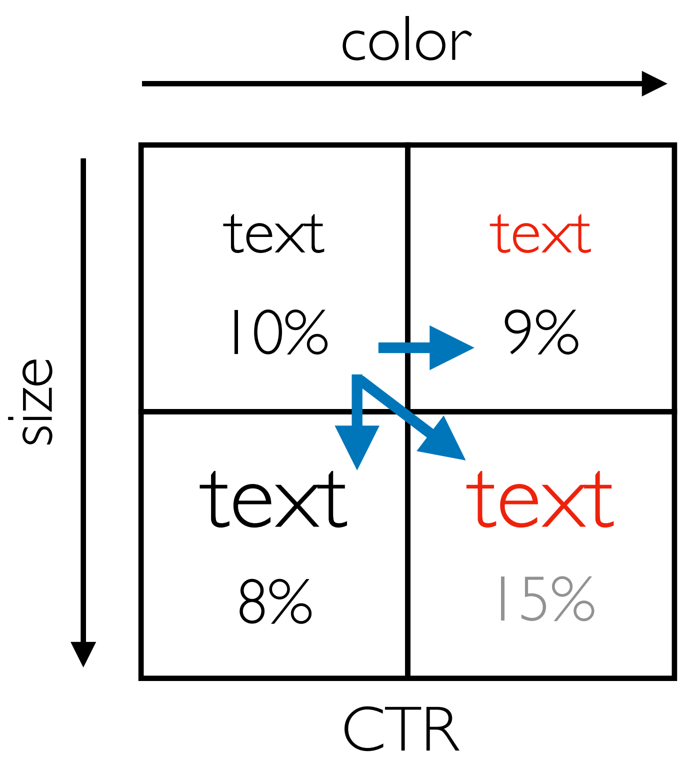
Option 1: OFAT (one factor at a time)

can usually learn more, but will never exploit factor interactions

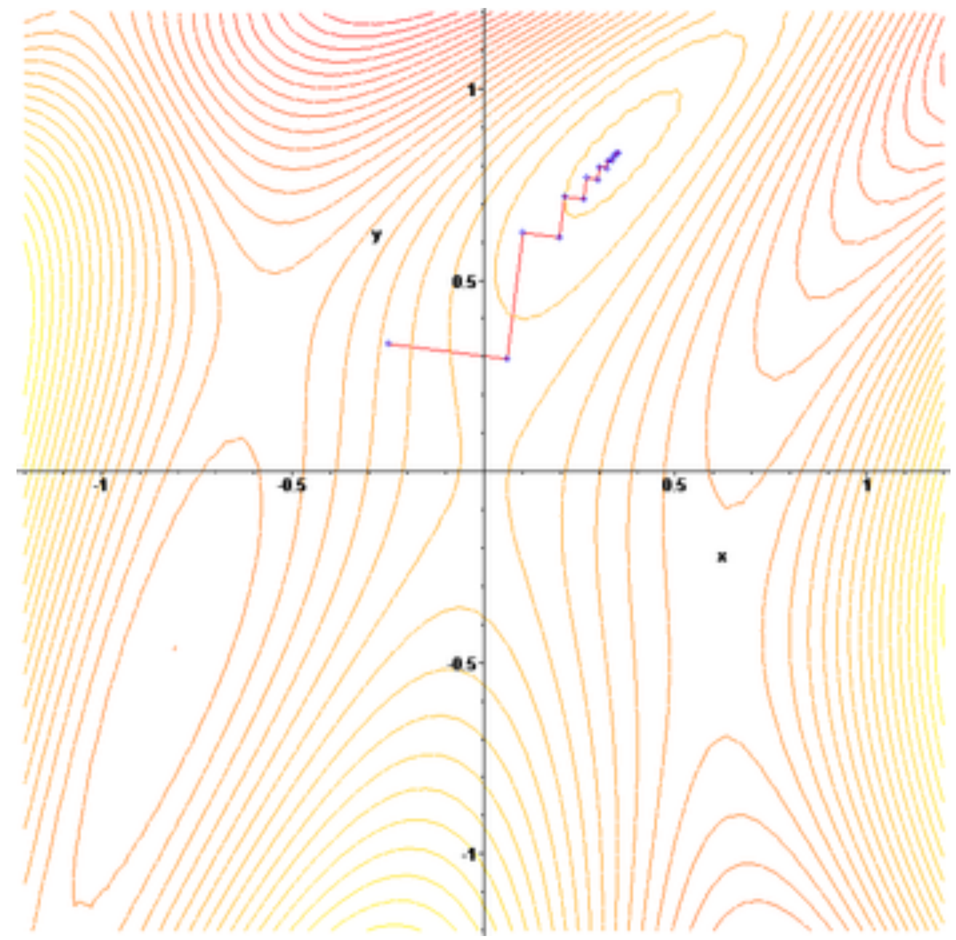
Option 2: introduce two factors at once

can choose a good design, but didn't learn what factors are important

Finding the Best Combination



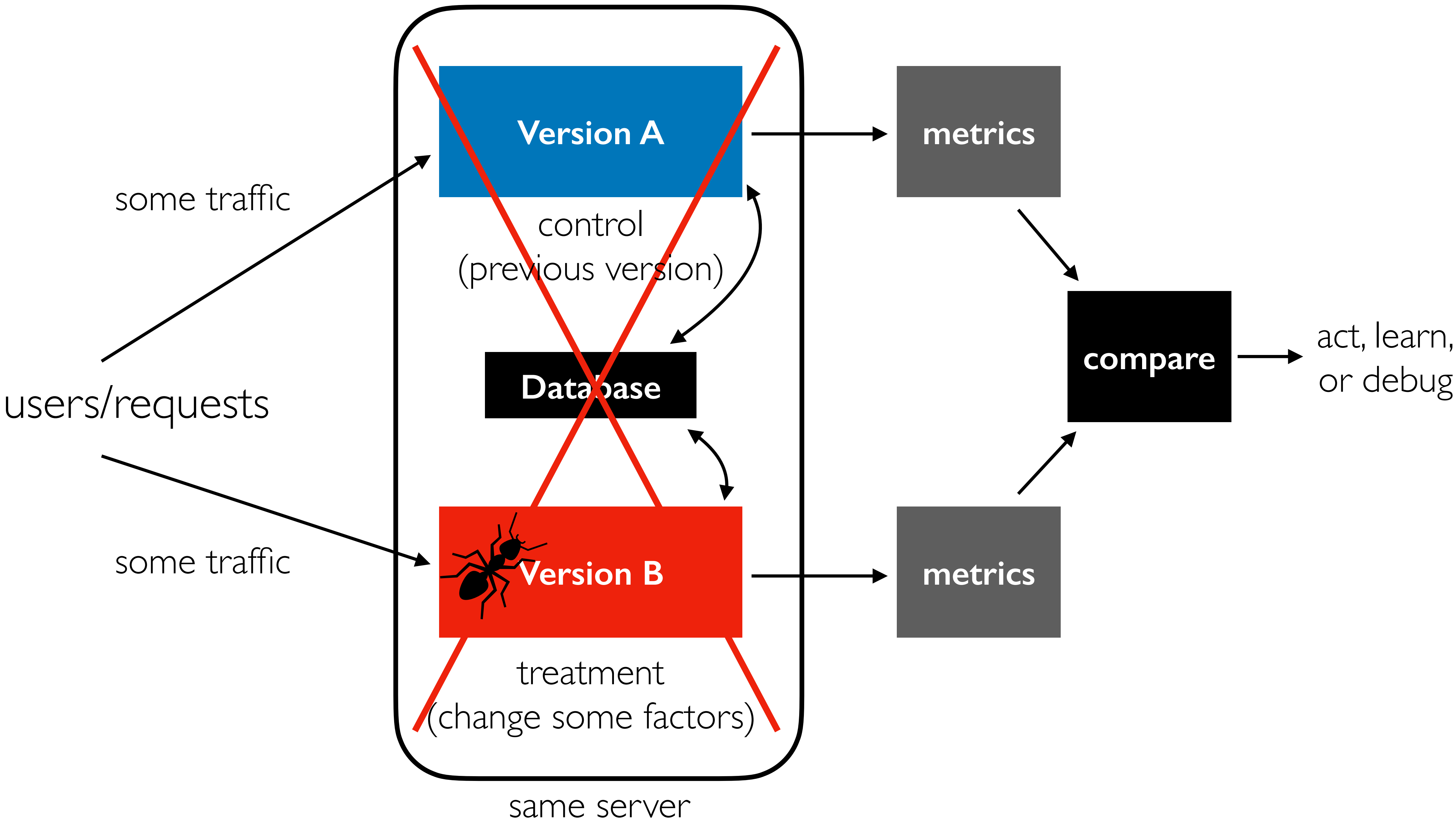
Hypothesis: large red font will be better



https://en.wikipedia.org/wiki/Gradient_descent

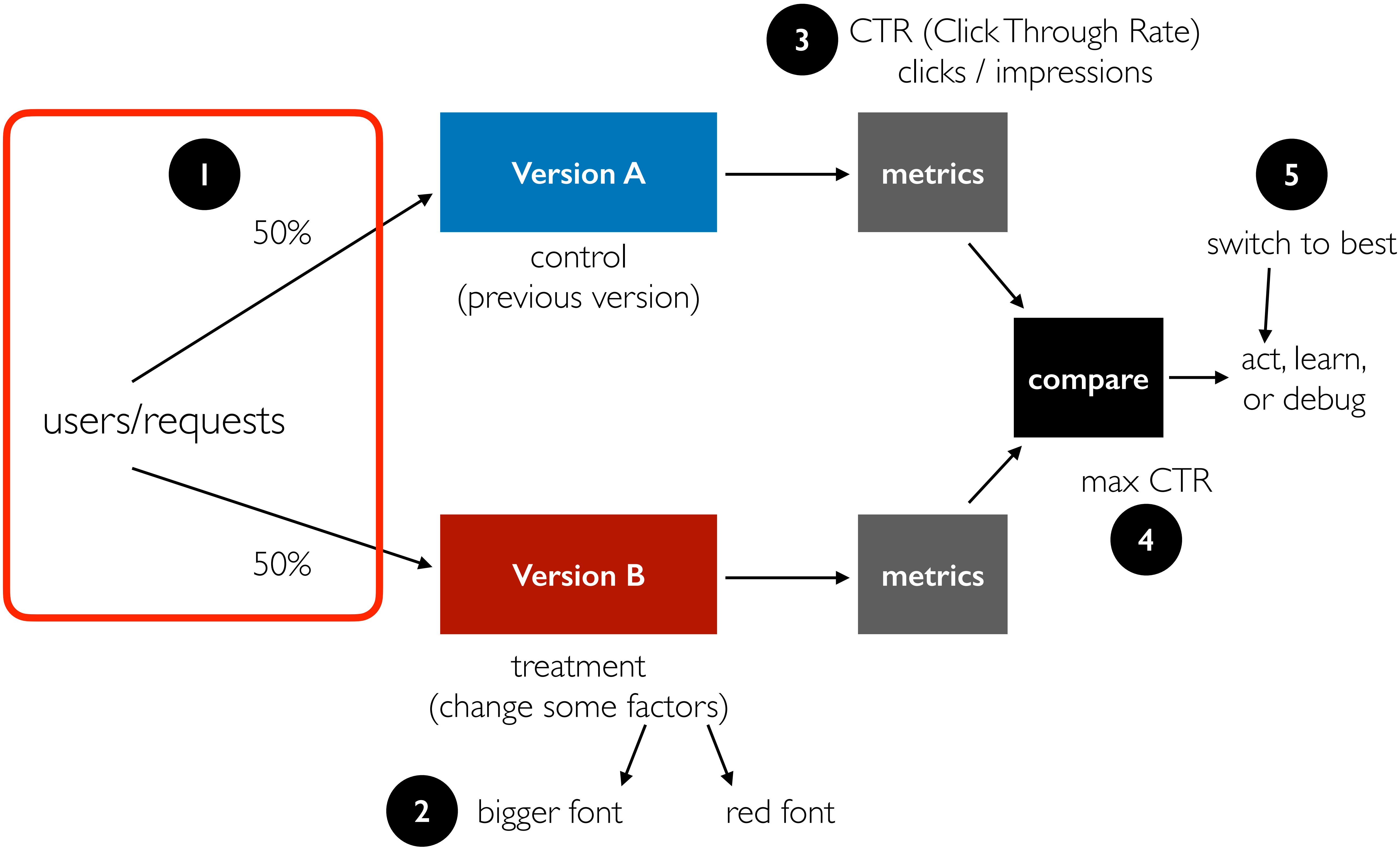
Hill climbing: imagine you're trying to find a peak (representing higher CTR). Taking small steps in the steepest direction is usually best, but not if you reach a local peak/optimimum

Control/Treatment Disruptions



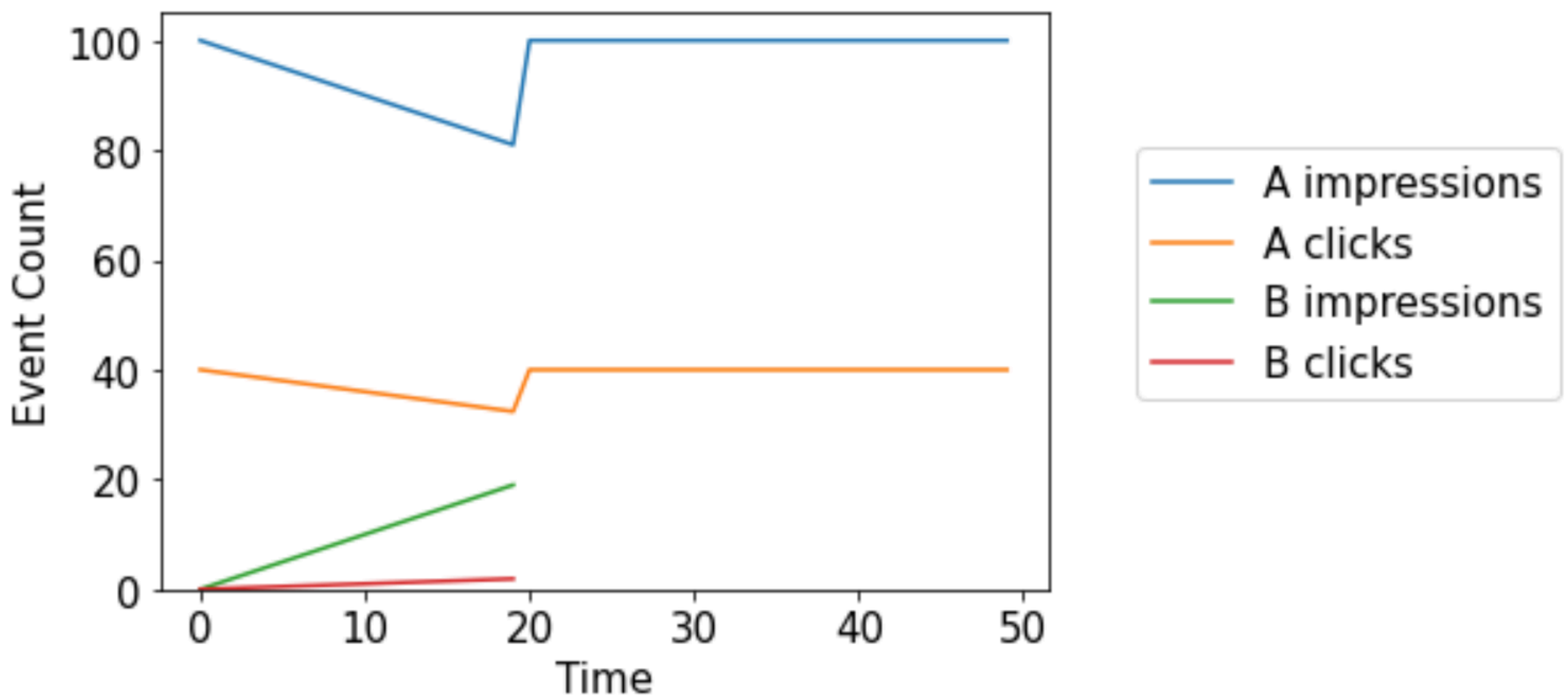
Different variants may save databases/servers, affecting performance of both. Bugs crashing the server will be especially bad! Metrics won't show the true blame.

Lecture Outline

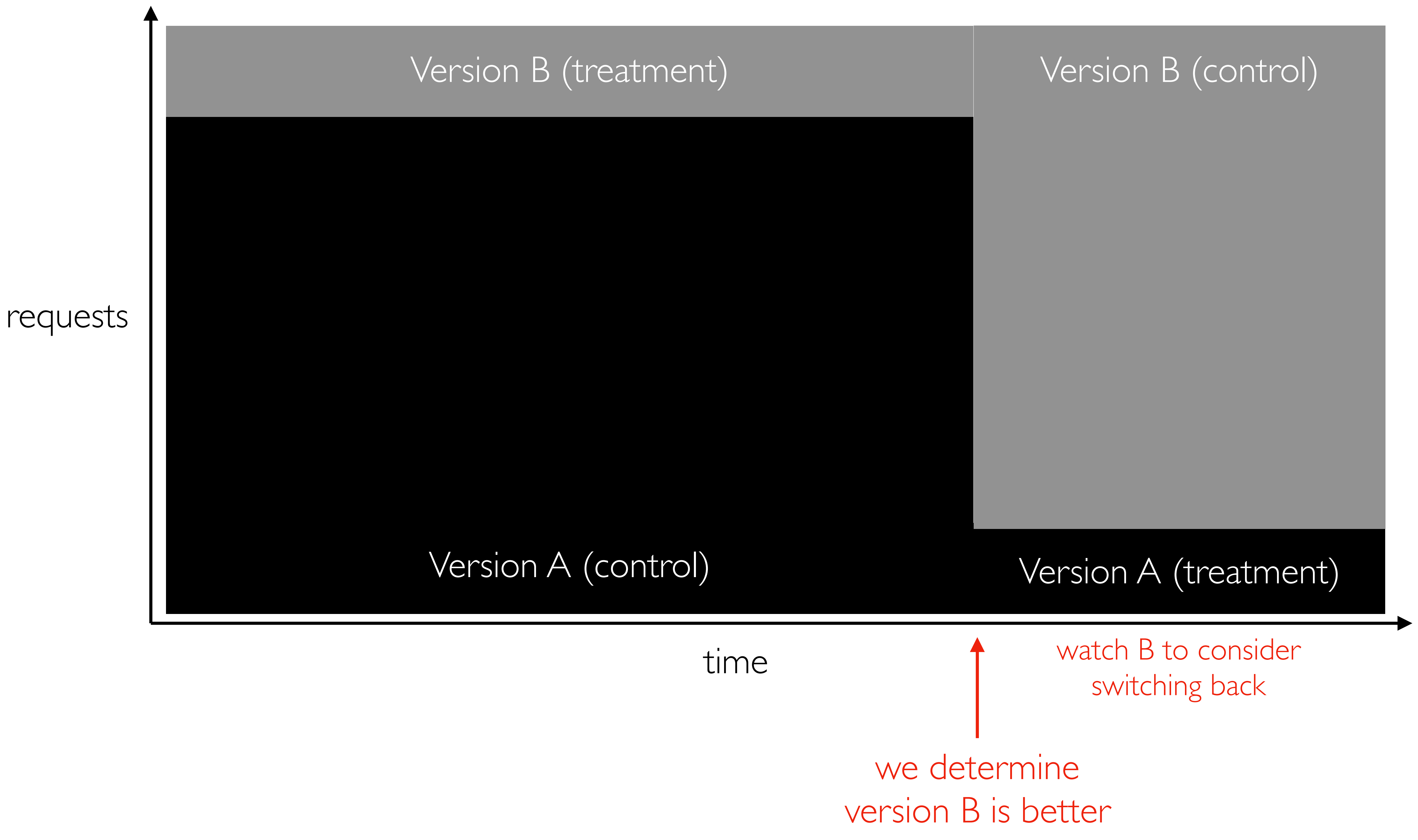


What to split

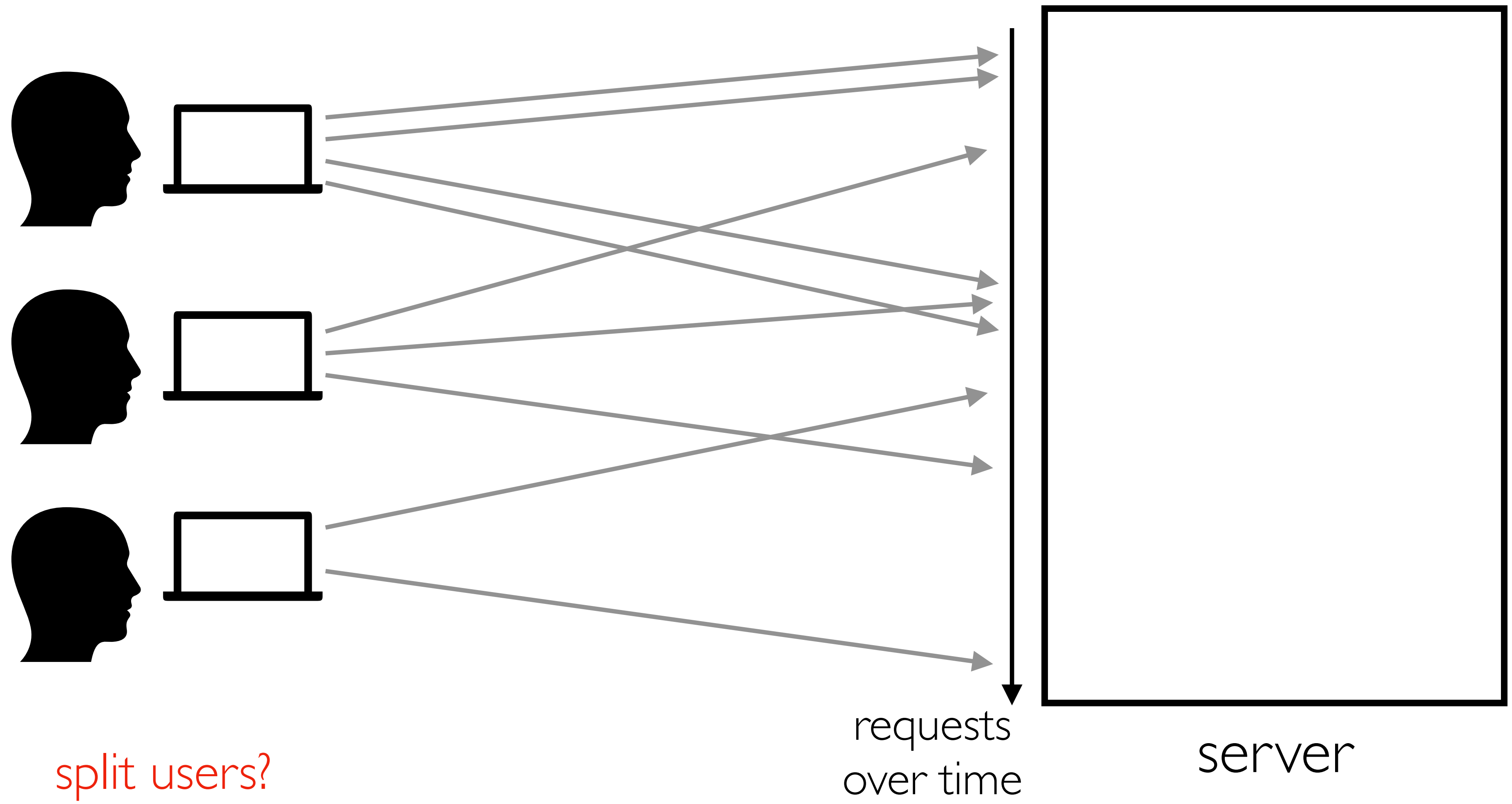
Don't go straight to 50/50!



What if the real factor is **novelty**?



What to split between control+treatment?



split users?

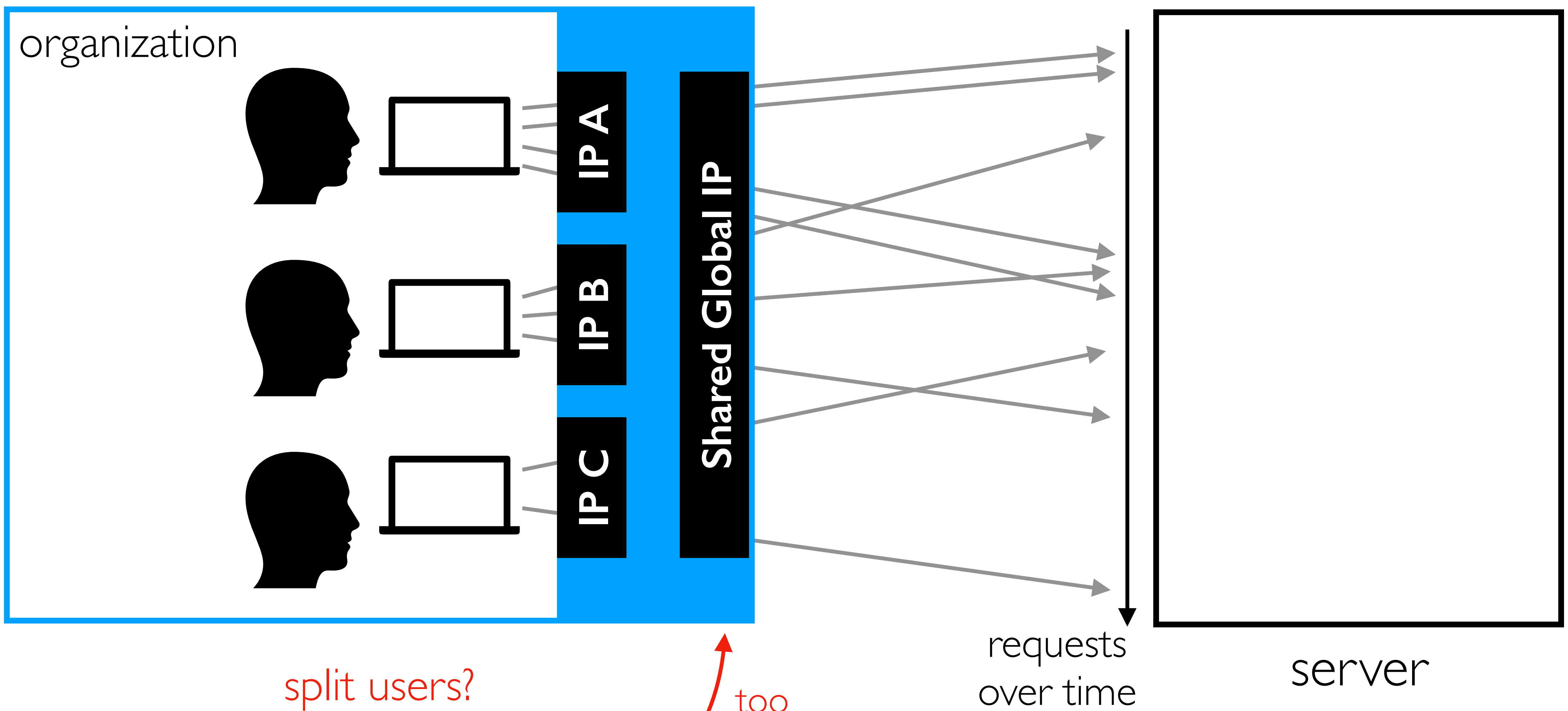
how to identify?

- IP addresses
- signed-in services
- cookies

or requests?

easier, but can't test over-time metrics or provide consistent experience

What to split between control+treatment?



split users?

too
many
share

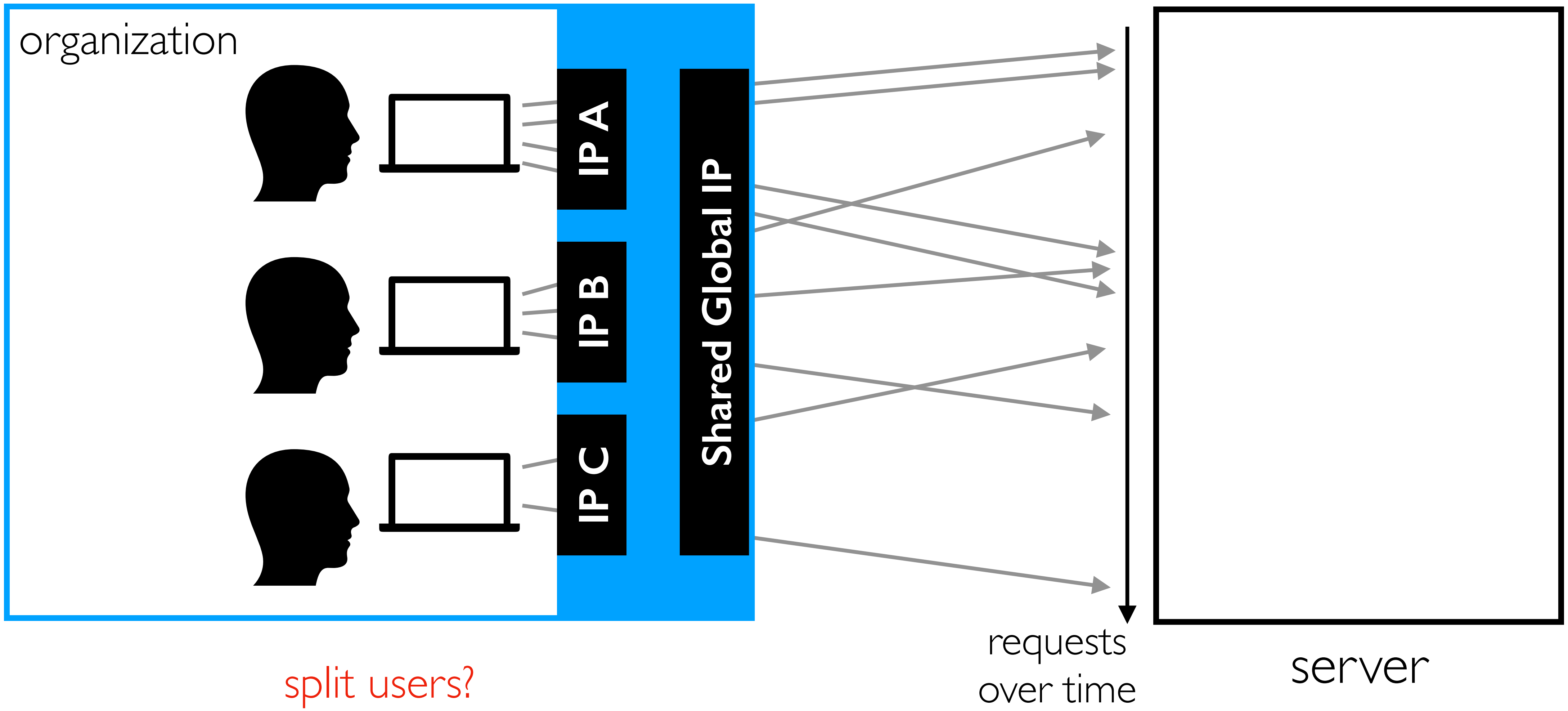
how to identify?

- IP addresses
- signed-in services
- cookies

or requests?

easier, but can't test over-time metrics or provide consistent experience

What to split between control+treatment?



split users?

how to identify?

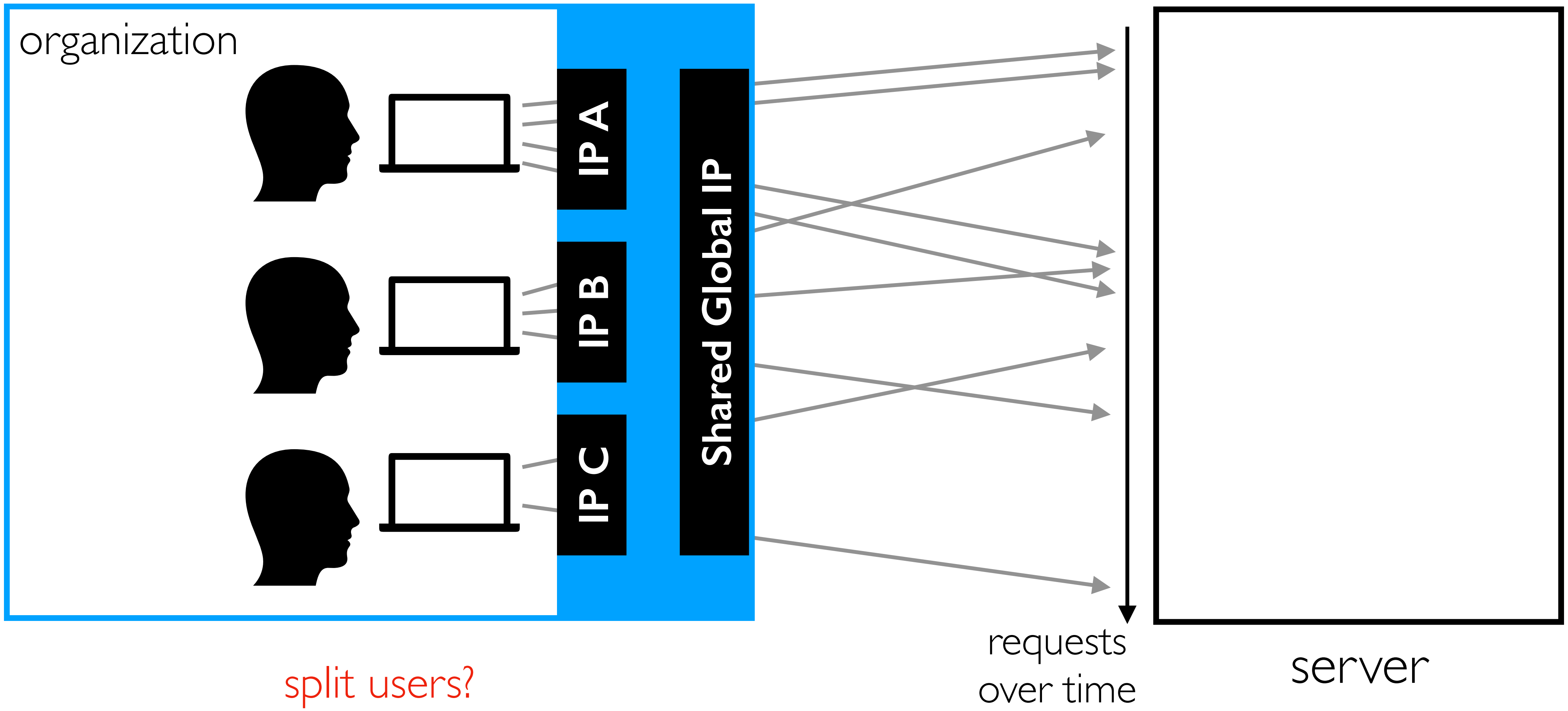
- IP addresses
- signed-in services
- cookies

ideal for when applicable

or requests?

easier, but can't test over-time metrics or provide consistent experience

What to split between control+treatment?



split users?

how to identify?

- IP addresses
- signed-in services
- cookies

or requests?

easier, but can't test over-time metrics or provide consistent experience

Cookies

Cookies are info that sites ask browsers to store locally and upload later.

```
from flask import request, Response, Flask

app = Flask(__name__)

@app.route('/')
def index():
    print(request.cookies)
    user_id = request.cookies.get("user", None)
    if user_id == None:
        user_id = new_id()
    resp = Response("hello")
    resp.set_cookie("user", user_id)
    return resp

def new_id():
    import time
    return str(time.time())

app.run(host="0.0.0.0")
```

dict of cookies

key

key

value

#TODO: get better identifiers

Cookies

Cookies are info that sites ask browsers to store locally and upload later.

dict of cookies

```
from flask import request, Response, Flask

app = Flask(__name__)

@app.route('/')
def index():
    print(request.cookies)
    user_id = request.cookies.get("user", None)
    if user_id == None:
        user_id = new_id()
    resp = Response("hello")
    resp.set_cookie("user", user_id)
    return resp
```

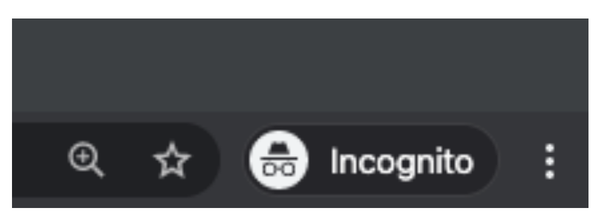
key

key value

```
def new_id():
    import time
    return str(time.time())

app.run(host="0.0.0.0")
```

#TODO: get better identifiers



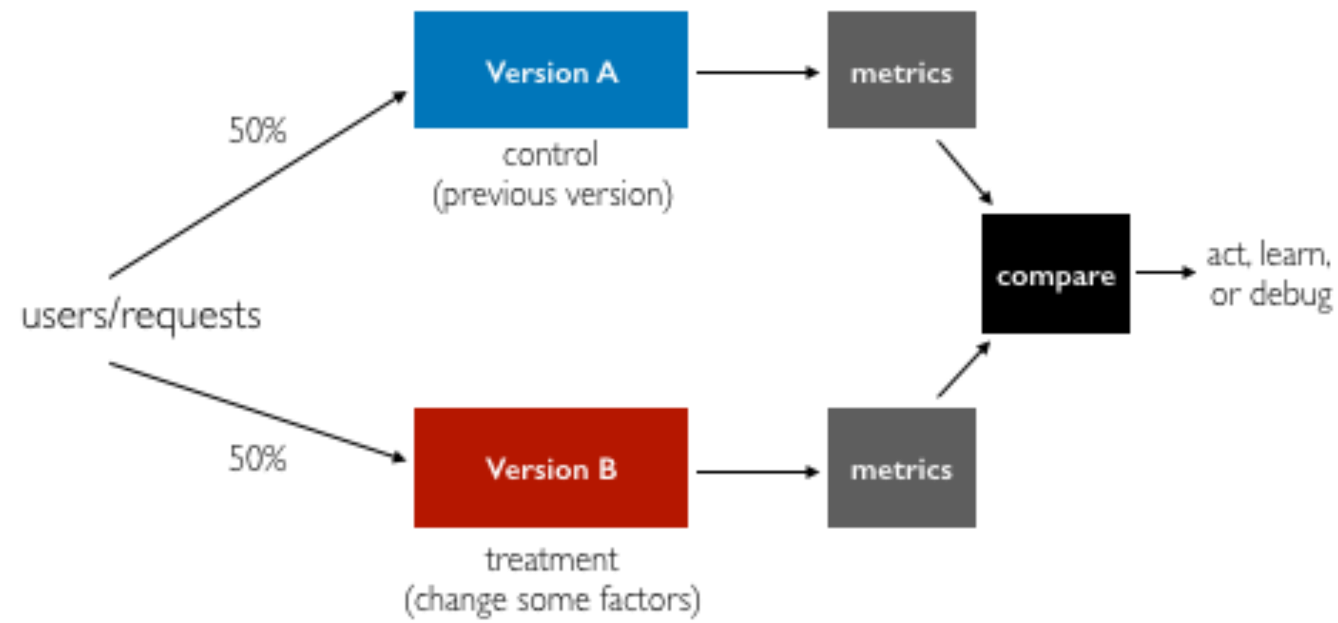
More accurate than IP, but cookie churn, incognito mode, and local laws may limit...

Summary

Goals
- make decisions, learn, debug

Comparisons
- significance testing

Metrics
- simple or combos
- clean uniformly
- choose OEC up front
- think long-term



Treatments
- one or more factors
- factors may require a lot of coding/design work!
- OFAT usually best for learning
- check the novelty factor with a flipped A/B test after decision

Splitting Traffic
- ramp up slowly
- split requests or users (how to distinguish?)