

Designing and Deploying Online Field Experiments

Eytan Bakshy*

Dean Eckles*

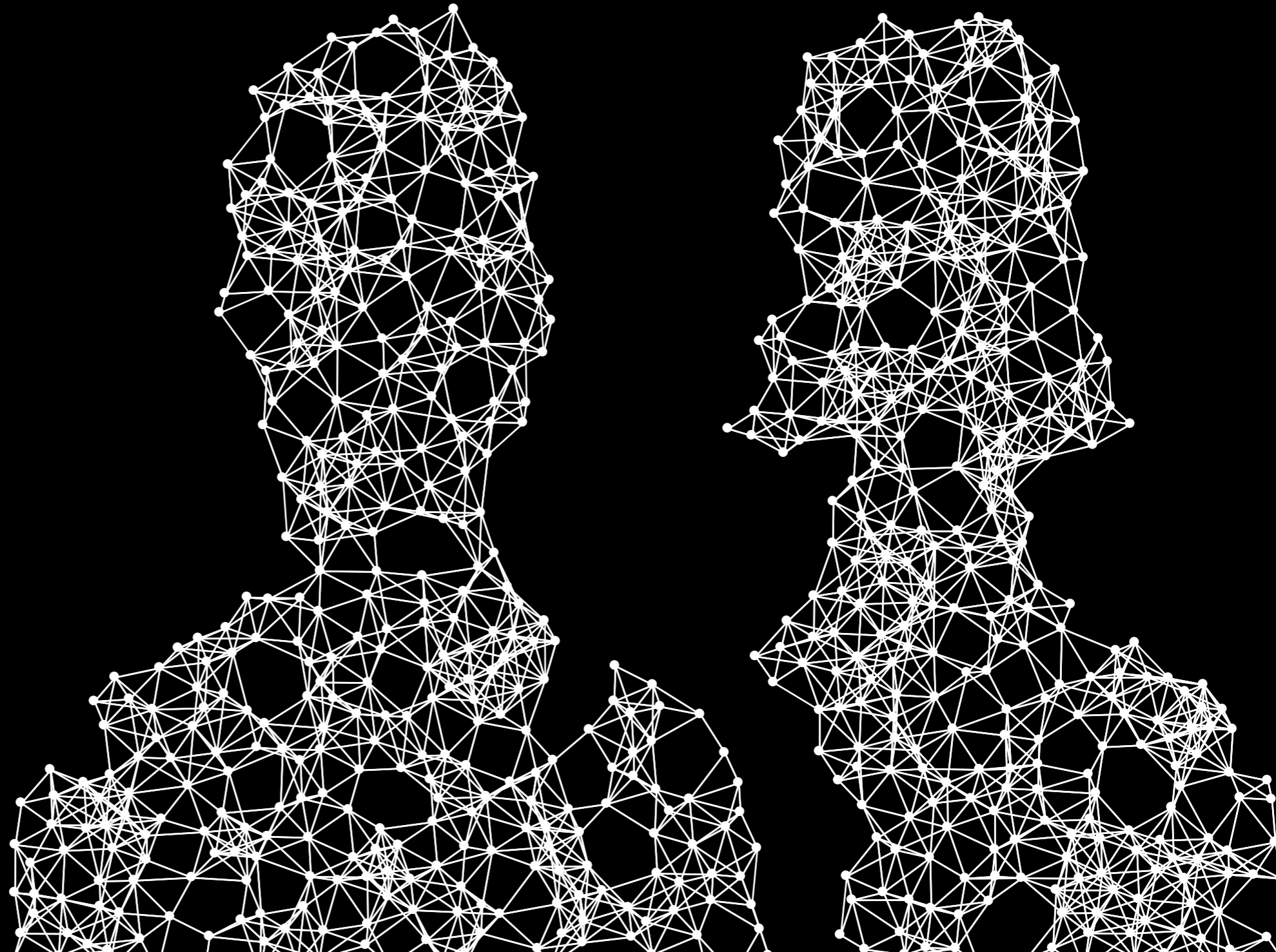
Michael S. Bernstein**

***Facebook**

****Stanford**

WWW 2014

April 10, 2014



Outline

- Motivation
 - Why run experiments?
 - Problems with running experiments
- Designing experiments with PlanOut
- Deploying experiments:
 - Logging and analysis
 - Management and iterative experimentation
- Discussion

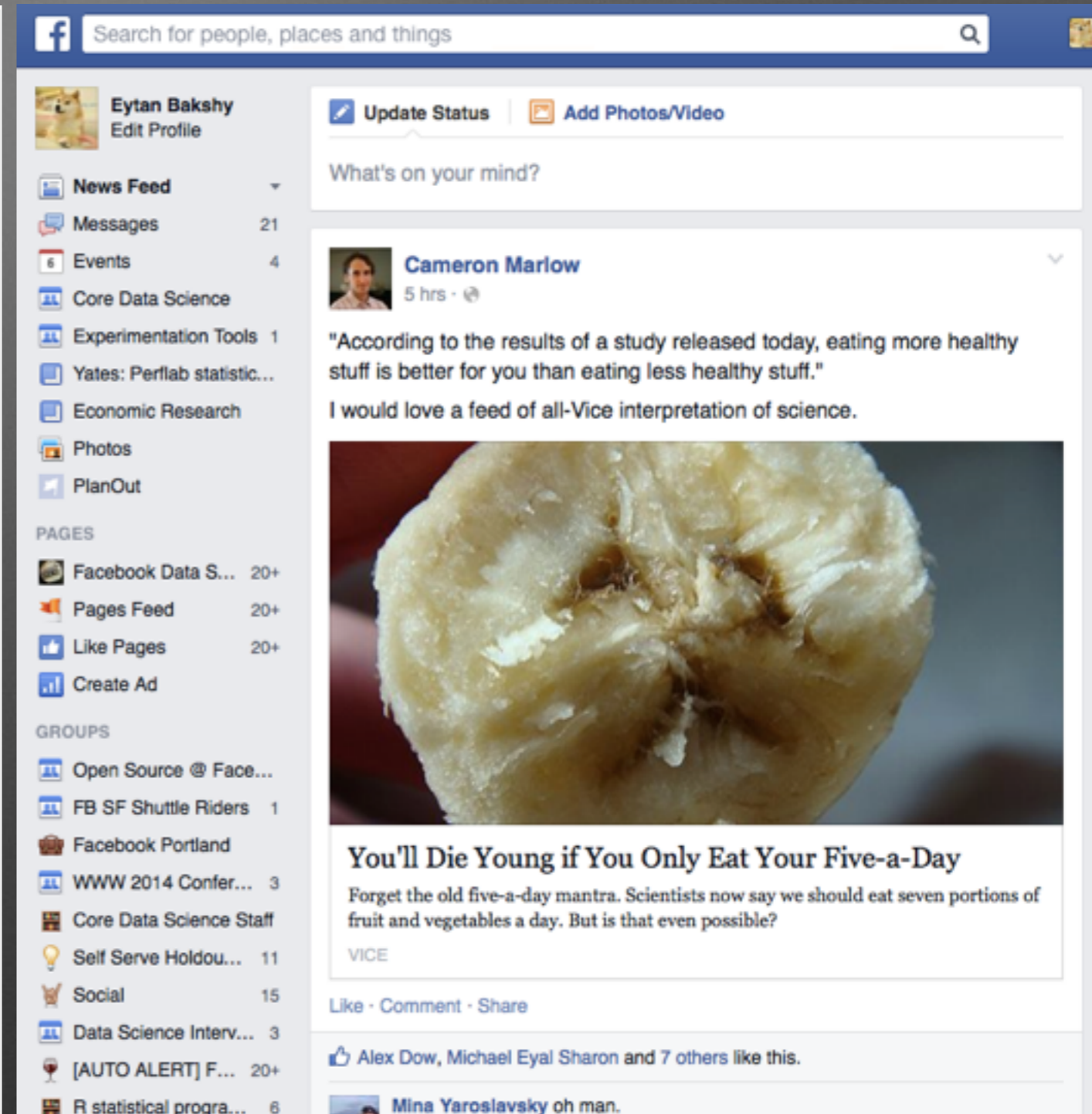
Motivation

Goals

Test complete alternatives

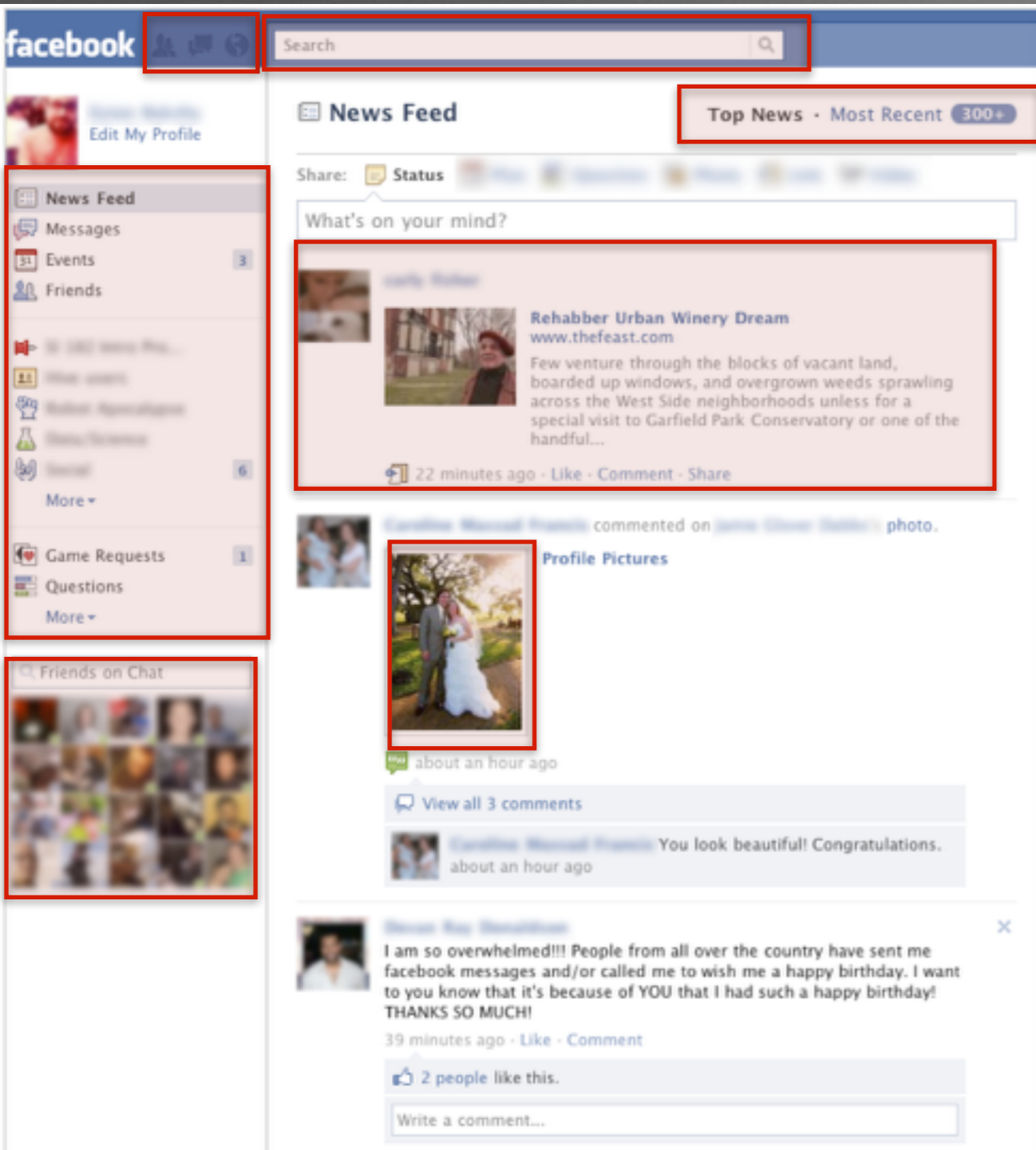


News Feed (2011)

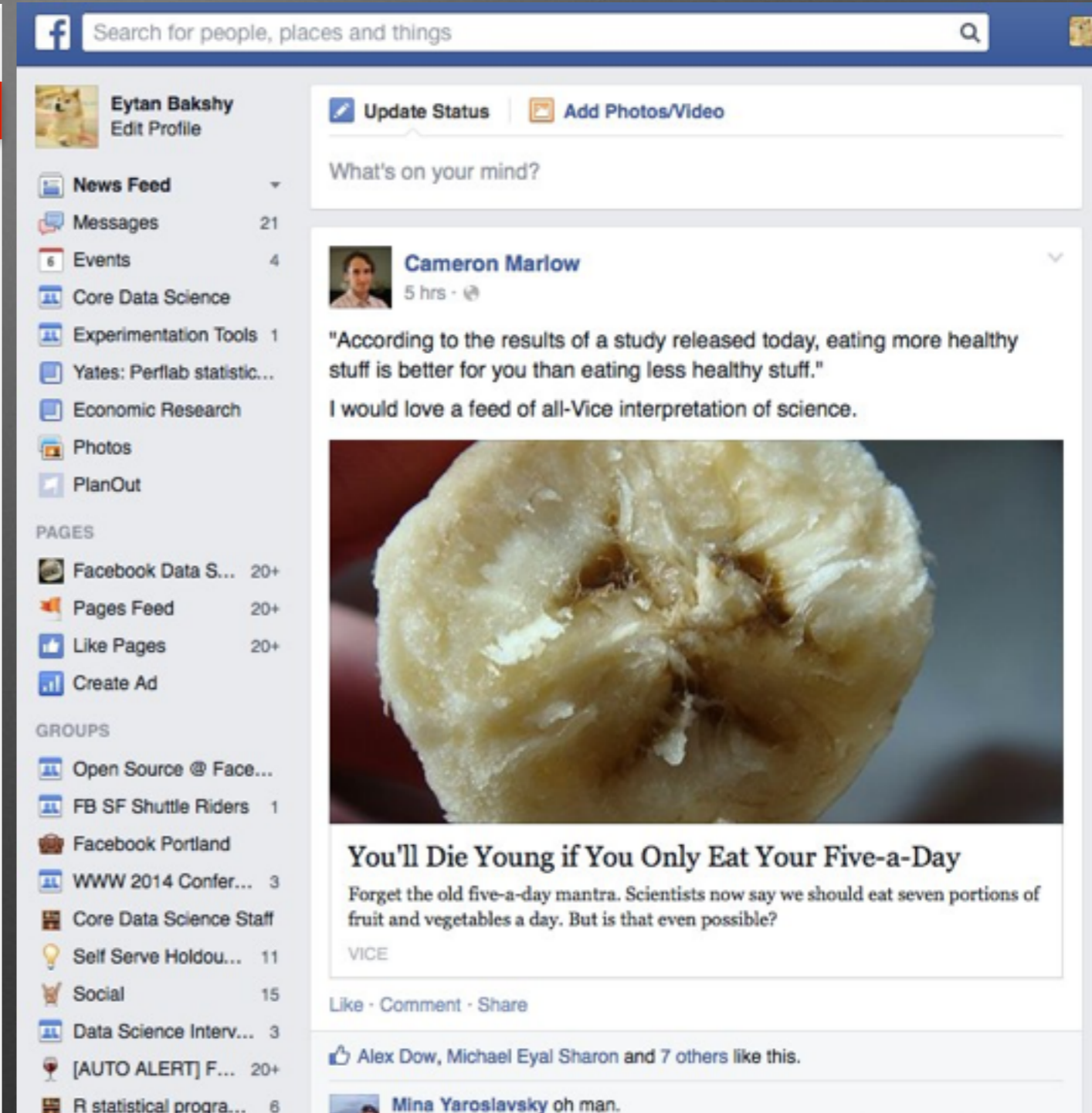


News Feed (2014)

Explore a design space



News Feed (2011)



News Feed (2014)

Understand tradeoffs



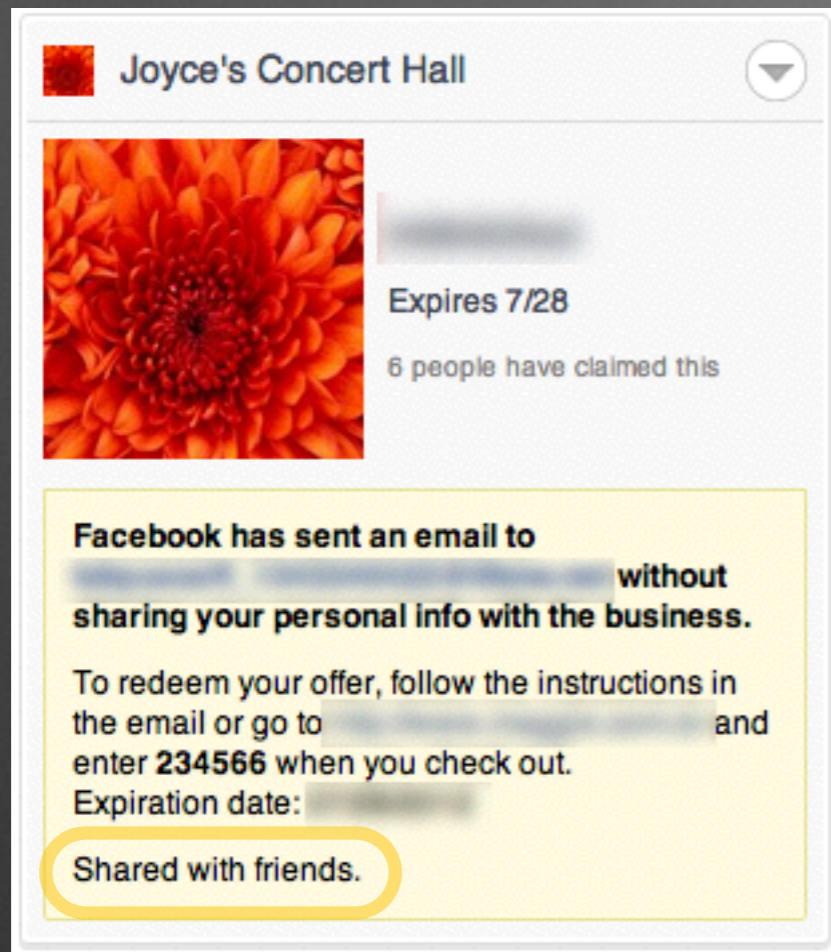
 **Solomon Messing** claimed an offer from Catakshy Home.
Yesterday at 23:24

 Come on by the Catakshy Home for a free home-cooked meal!
6 claimed this

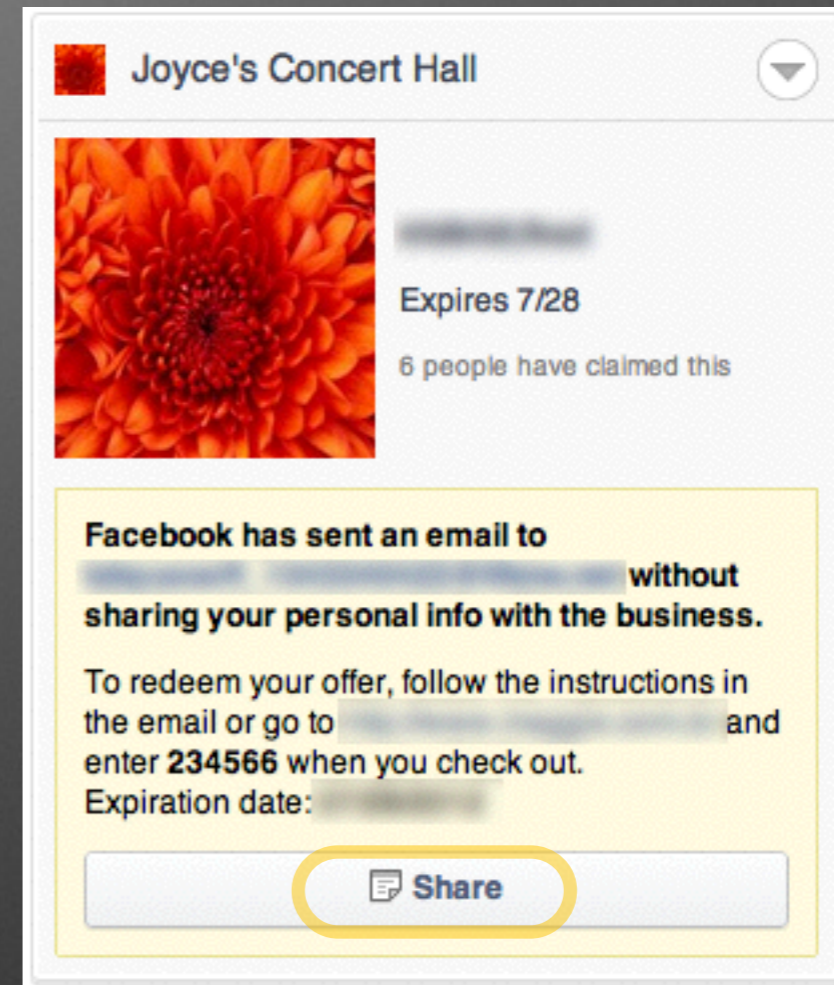
Get Offer

Like • Comment

Understand tradeoffs



Passive sharing
(old sharing model)



Active sharing
(new sharing model)

Attribute outcomes to causes

Healthy Meals, No Cooking
munchery.com



Too busy to cook? Order wholesome prepared meals from top SF chefs for same-day delivery.

non-social ad

History

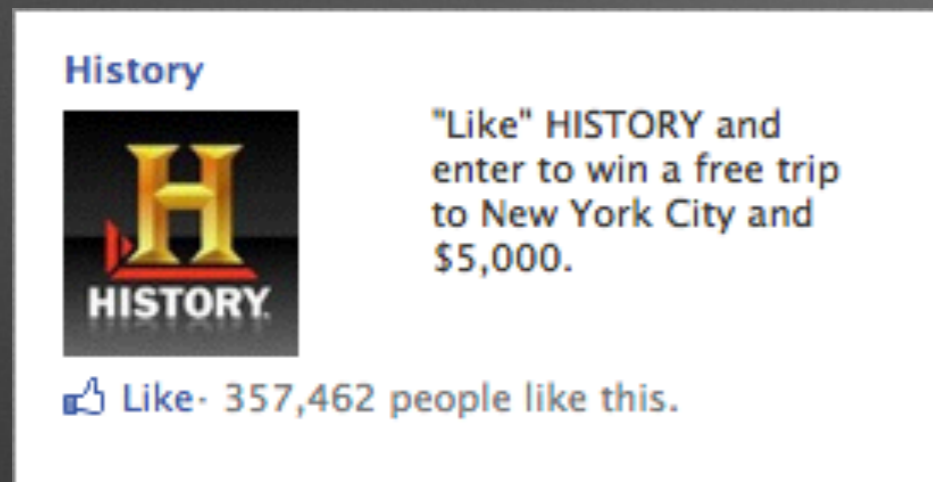


"Like" HISTORY and enter to win a free trip to New York City and \$5,000.

Like · Jina [profile picture] likes this.

social ad

Attribute outcomes to causes



1 liking friend
0 friends shown



1 liking friend
1 friend shown

Experiments can be hard

Organizational problems

- **Multiple simultaneous experiments**
 - **Sometimes need to iterate**
- **Not clear how to manage changes to experiments**
- **Changing experiments breaks randomization**

Engineering problems

- **Experimental logic is embedded in application code**
- **Distributed logic makes experiments brittle**
- **Difficult to understand for those who didn't write the code**
- **Difficult to run experiments on multiple platforms**

Parameterization of user experiences



feed width

image width

font size

padding

number of social cues

PlanOut

**PlanOut scripts are
high-level descriptions
of randomized
parameterizations**

The PlanOut idea

- User experiences are parameterized by experimental assignments
- PlanOut scripts describe assignment procedures
- Experiments are PlanOut scripts plus a population
- Parallel or follow-on experiments are centrally managed

Sample PlanOut script

```
button_color = uniformChoice(  
  choices=["#ff0000", "#00ff00"],  
  unit=userid);
```

```
button_text = uniformChoice(  
  choices=["I'm voting", "I'm a voter"],  
  unit=userid);
```

2x2 factorial design

Compiled PlanOut code

```
{
  "op": "seq",
  "seq": [
    {
      "op": "set",
      "var": "button_color",
      "value": {
        "choices": {
          "op": "array",
          "values": [
            "#ff0000",
            "#00ff00"
          ]
        }
      },
      "unit": {
        "op": "get",
        "var": "userid"
      },
      "op": "uniformChoice"
    }
  ],
  {
    "op": "set",
    "var": "button_text",
    "value": {
      "choices": {
        "op": "array",
        "values": [
          "I'm voting",
          "I'm a voter"
        ]
      },
      "unit": {
        "op": "get",
        "var": "userid"
      },
      "op": "uniformChoice"
    }
  ]
}
```

PlanOut code is portable



"choice"
choice"

PlanOut code can be generated by GUIs

Experiment Design

button_color (Uniform) x

#3c539a	x
#5f9647	x
#b33316	x
enter value	x

Add Choice

button_text (Weighted) x

Sign up	0.8	x
Join now	0.2	x
enter value	enter weight	x

Add Choice

factor_name Uniform Choice ⌵ Add Factor

How does it work?

```
e = getExp('share_dialog', userid=20)
button_color = e.get('button_color')
```

retrieve experiment script

```
button_color = weightedChoice(
    choices=["#ff0000", "#00ff00"],
    weights=[0.2, 0.8], unit=userid);
```

construct hash string

```
share_dialog.button_color.20
```

f(SHA1(str))

```
"#ff0000"
```


Examples of PlanOut experiments

**What is the marginal effect
of social cues on an action?**

Social cues experiment



Social cues experiment



```
num_cues = randomInteger(  
    min=1, max=min(length(liking_friends), 3),  
    unit=[userid, pageid]);
```

Social cues experiment



```
num_cues = randomInteger(  
    min=1, max=min(length(liking_friends), 3),  
    unit=[userid, adid]);
```

```
friends_shown = sample(  
    choices=liking_friends, draws=num_cues,  
    unit=[userid, adid]);
```

How can we increase voting?

**Can we motivate voter turnout
by invoking the self?**

It's Election Day



Tell friends you're voting in the 2012 Election and find out where to vote.

I'm a Voter Find My Polling Place



Adam, Adam and 4 others are voters in this election.

It's Election Day



Tell friends you're voting in the 2012 Election and find out where to vote.

I'm Voting Find My Polling Place

Banner exposure



User action

Intensity of social cues



News Feed exposure

	Has banner	No banner
Has feed stories	2%	1.5%
No feed stories	95%	1.5%
<i>(sum)</i>	<i>97%</i>	<i>3%</i>

```
has_banner = bernoulliTrial(  
    p=0.97, unit=userid);
```

```
cond_probs = [0.5, 0.98];
```

```
has_feed_stories = bernoulliTrial(  
    p=cond_probs[has_banner], unit=userid);
```


**To what extent does
feedback affect content
production?**

Fact:

Collapsing comment boxes modulates feedback

Expanded comment box



Higher interaction rate

Collapsed comment box



Lower interaction rate

Effect of feedback on content production

```
prob_collapse = randomFloat(  
    min=0.0, max=1.0, unit=sourceid);
```


Effect of feedback on content production

```
prob_collapse = randomFloat(  
    min=0.0, max=1.0, unit=sourceid);  
  
collapse = bernoulliTrial(  
    p=prob_collapse, unit=[storyid, viewerid]);
```

- Simple: prob_collapse vs content production
- Fancy (2SLS):
production ~ comments; comments ~ prob_collapse

Logging and analysis

```
e = getExp('share_dialog', userid=20)
```

```
button_color = e.get('button_color')
```

trigger auto exposure log

```
{
  "inputs": {
    "userid": 20
  },
  "name": "sharing_dialog",
  "params": {
    "button_color": "#ff0000",
    "button_text": "Share with others"
  },
  "time": 1396507677,
  "salt": "sharing_dialog",
  "event": "exposure"
}
```



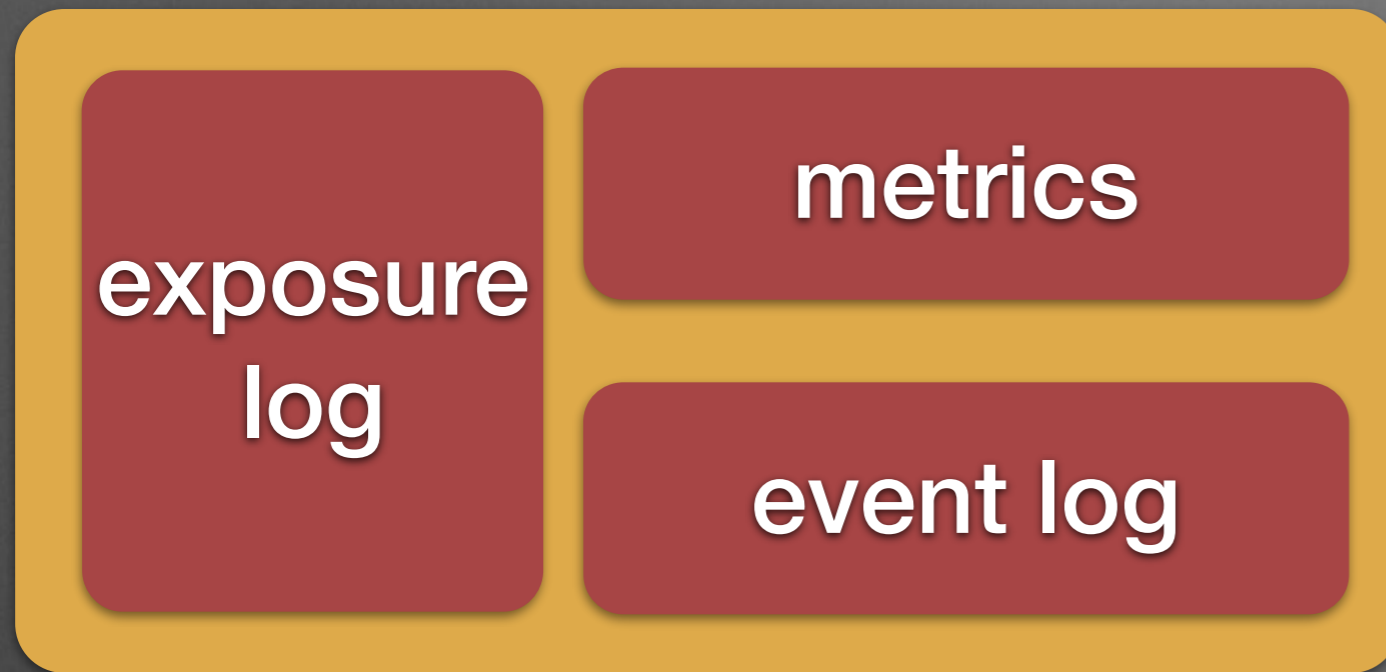
```
e = getExp('share_dialog', userid=20)
```

```
e.log_event('button_click')
```



log event

```
{
  "inputs": {
    "userid": 20
  },
  "name": "sharing_dialog",
  "params": {
    "button_color": "#ff0000",
    "button_text": "Share with others"
  },
  "time": 1396507677,
  "salt": "sharing_dialog",
  "event": "button_click"
}
```



outcomes joined with exposure log



Hive + R



e.g. relative change + confidence intervals



SQL + D3



results can be looked up by experiment name

Managing and analyzing related experiments

Namespaces: a way of managing experiments

```
e = getExp('share_dialog', userid=20)
```

```
button_color = e.get('button_color')
```



Hash userid to a NS-specific segment number



segment to experiment mapping

Exp 1

Exp 2

Default value store



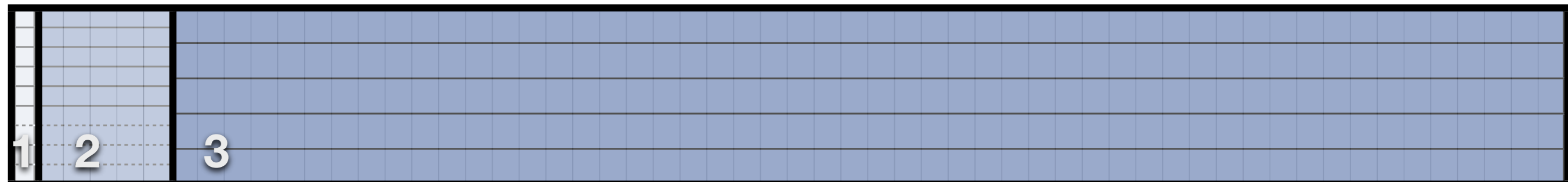
log + return values

return values

Deploying iterative experiments

week 1: launch initial PlanOut script with many conditions

week 2: launch same script with more segments



segments



week 4: launch new script with fewer conditions,
more segments

Best practices for iterative experiments

- Use namespaces for all experiments
- Changes to experiments done through launching new experiments
- Follow-on experiments analyzed separately
- Pool results of related experiments with care*

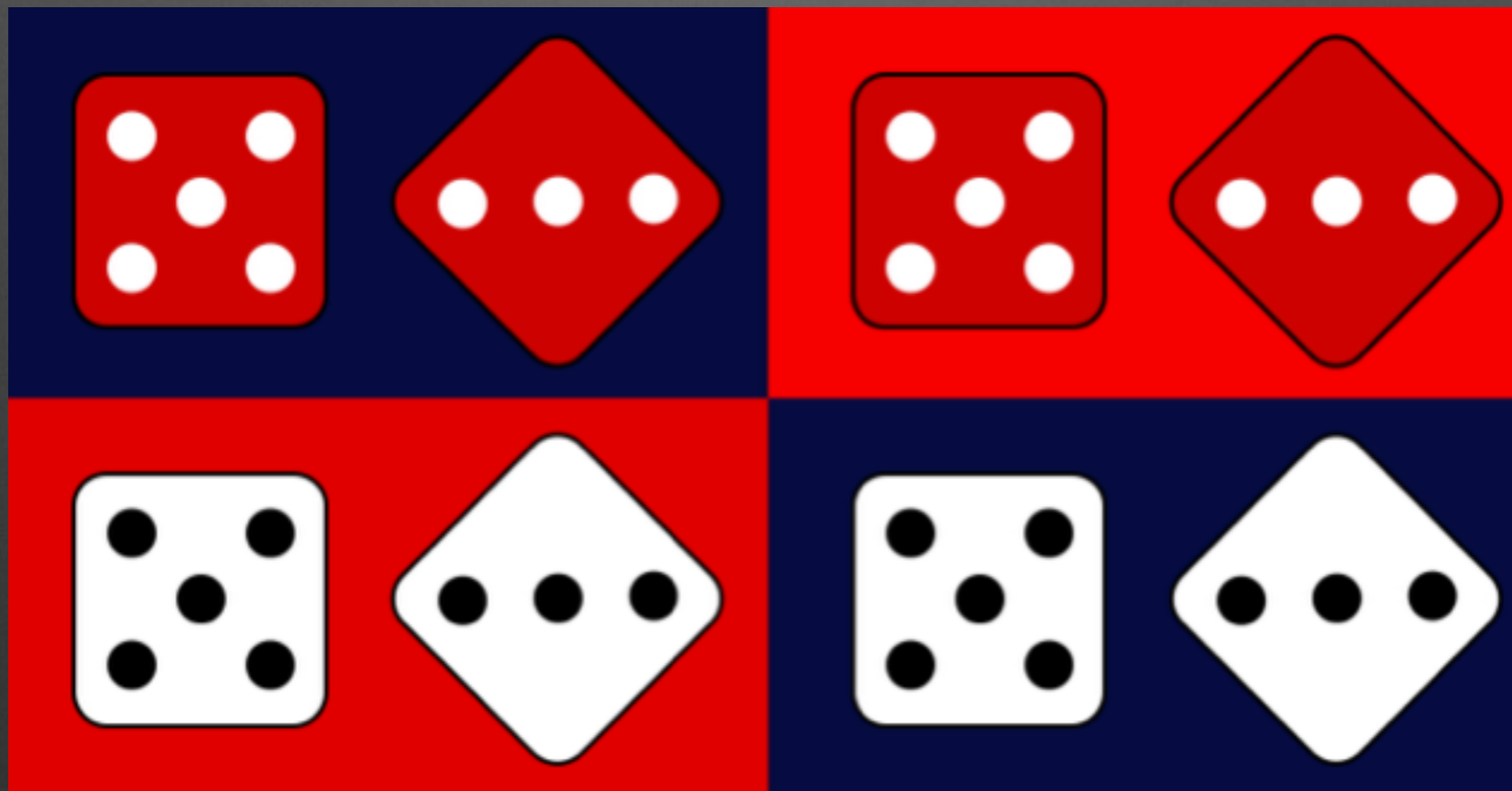
*cf. Statistical Inference in Two-Stage Online Controlled Experiments with Treatment Selection and Validation. Deng, Li, Guo. WWW 2014.

Recap

- Experiments are useful for knowledge building
- PlanOut supports good experiments
 - Focuses experimenters on parameters
 - Makes it easy to run complex studies
- Management frameworks prevent you from shooting yourself in the foot

Use PlanOut

<http://facebook.github.io/planout>



```
> easy_install planout
```


Thank you!

- **Coauthors**



Dean Eckles



Michael Bernstein

- **Special thanks: Daniel Ting, Wojtek Galuba, Breno Roberto, Wesley May**