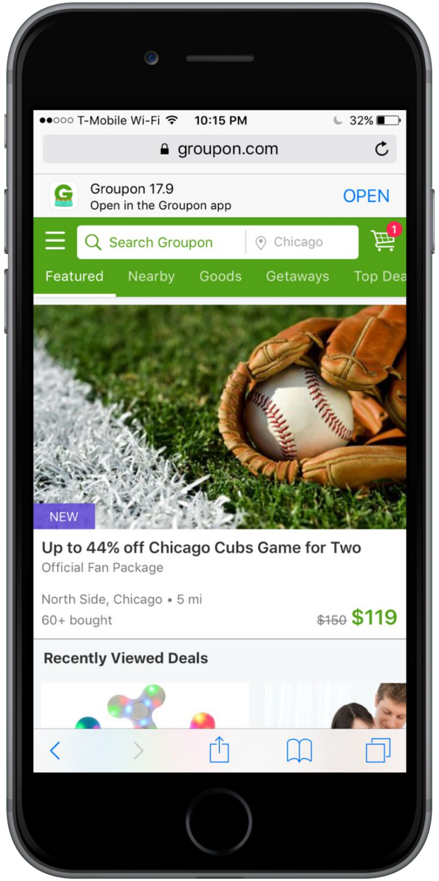
Data-Driven Product @ Groupon

Laura Hamilton
June 7, 2017

Mission: To become the daily habit in local commerce

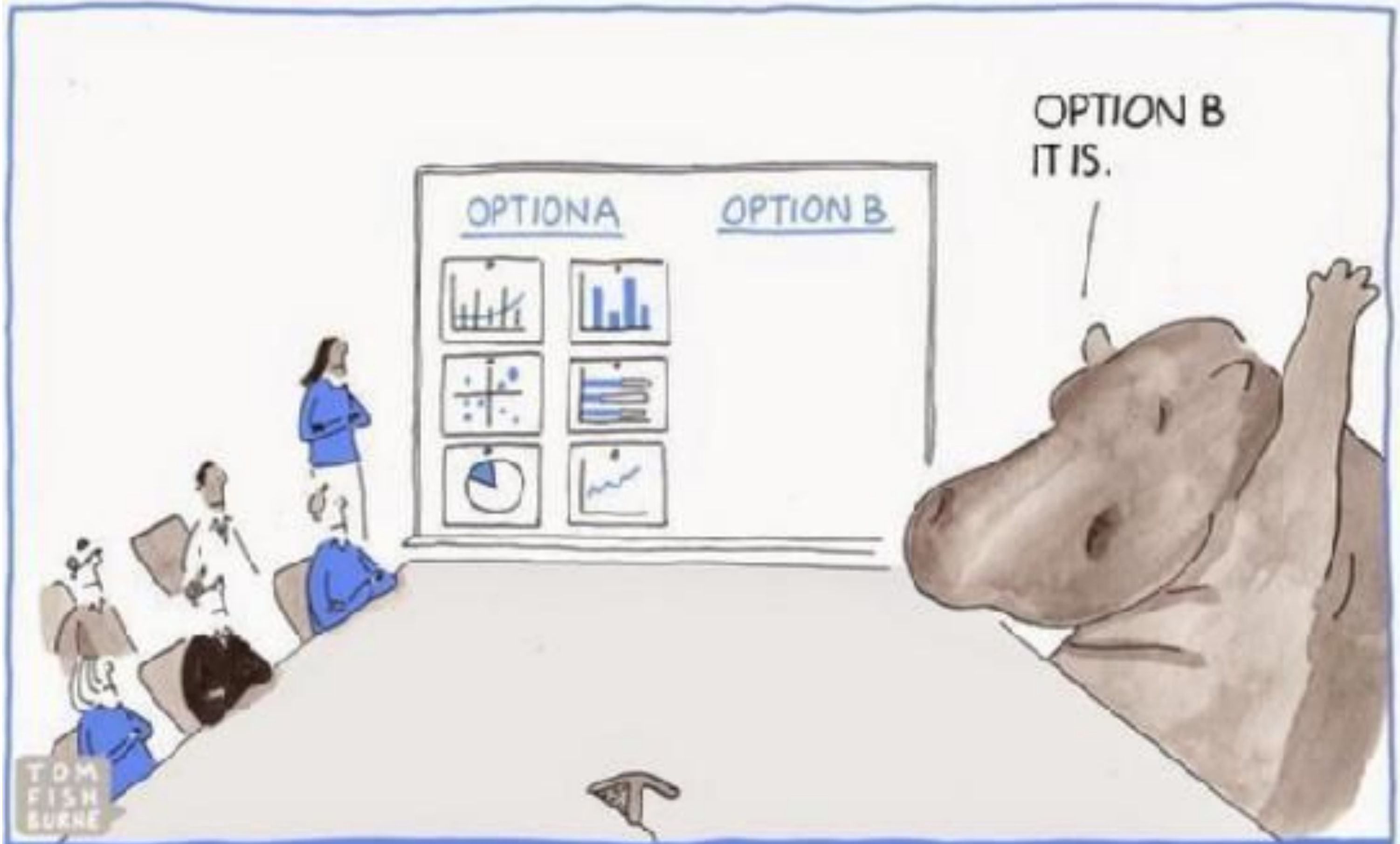
Groupon Scale

- **One million merchants** worked with to date
- More than **6,000 employees** globally
- **49.6 million active customers**
- **177 million downloads** of the mobile app
- Nearly **1.5 billion** Groupons sold
- More than **$29 billion** saved by consumers
- **Tens of billions** of user actions per month
- Decisions made in **fractions of a second**



Up to 44% off Chicago Cubs Game for Two
Official Fan Package
North Side, Chicago • 5 mi
60+ bought
~~$150~~ $119

Recently Viewed Deals

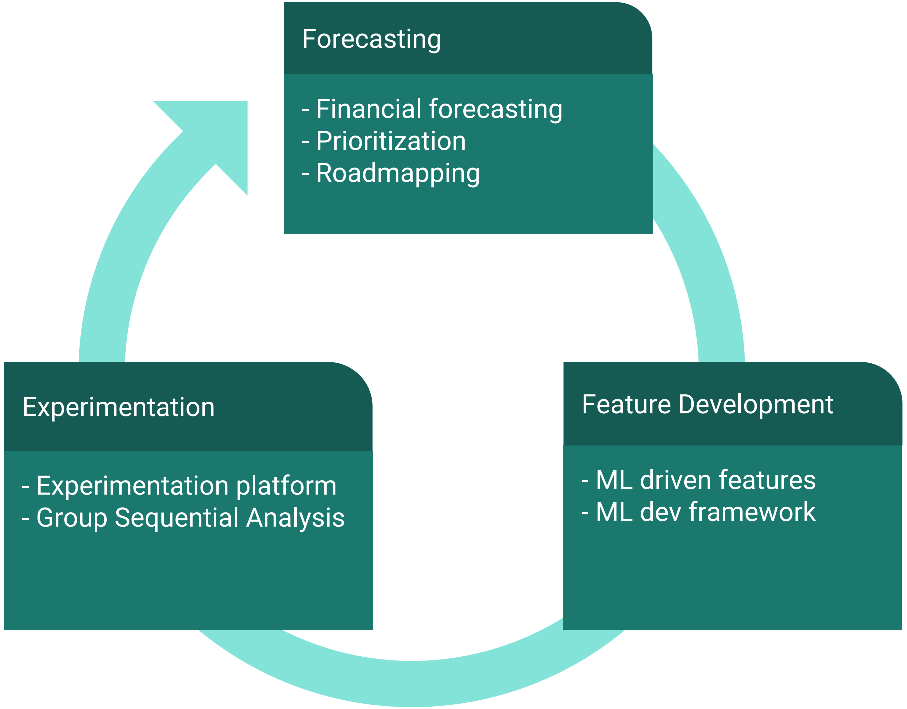


Let data drive decisions, not the HIghest Paid Person's Opinion.

#HowGoogleWorks HowGoogleWorks.net

“Companies that make HIPPO decisions rather than data-driven decisions are at a massive competitive disadvantage.”

Agenda



Forecasting

- Financial forecasting
- Prioritization
- Roadmapping

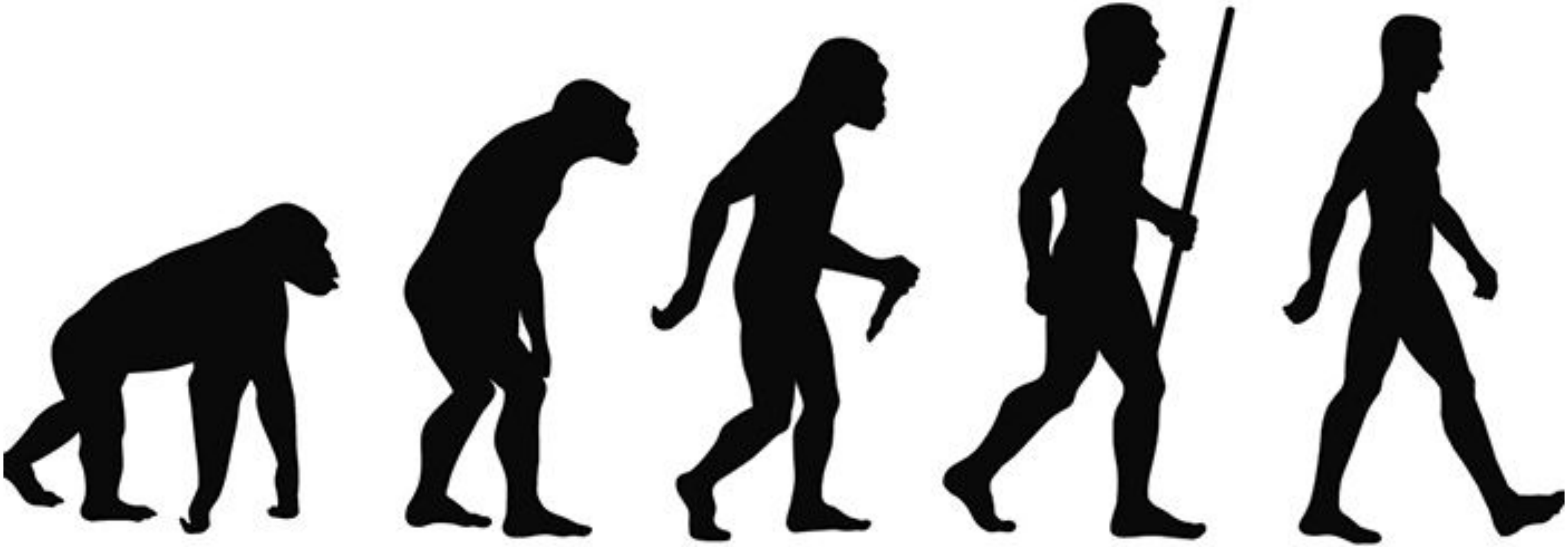
Feature Development

- ML driven features
- ML dev framework

Experimentation

- Experimentation platform
- Group Sequential Analysis

Groupon Journey



- Product/market fit
- Pre/post analysis
- Weekly business reviews

- A/B testing
- ROI-based roadmapping
- Forecasting
- Machine learning
- EDW

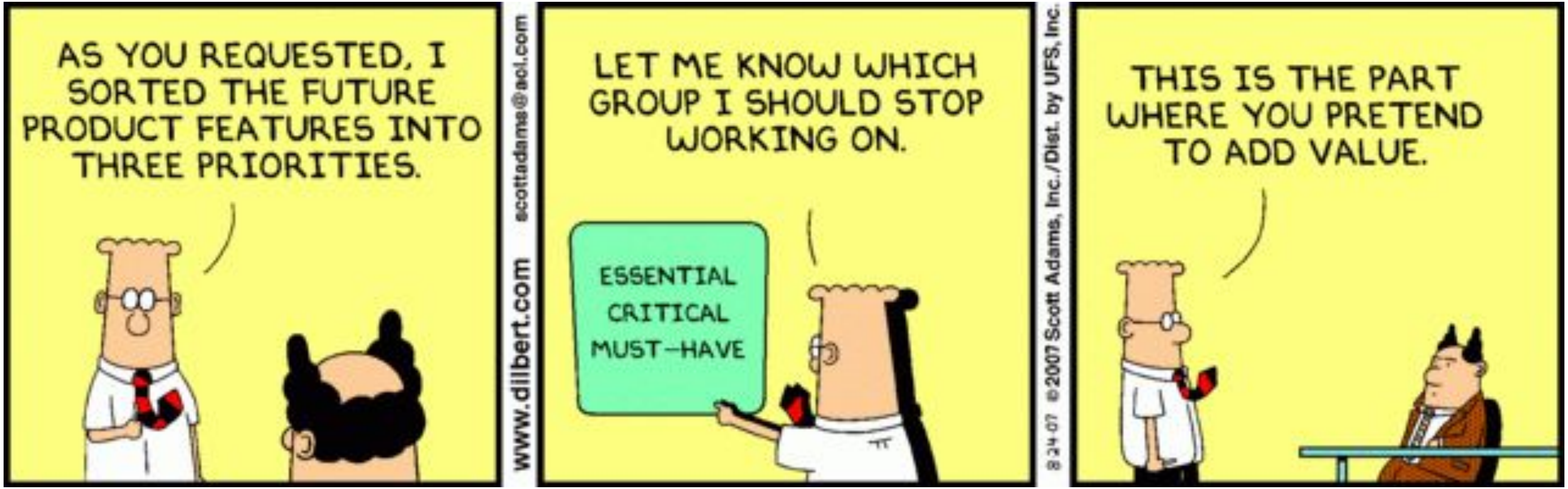
- Automatic experimentation
- ML framework

- Codeless experiments
- Group Sequential Analysis

- Image processing
- AI chatbots

Forecasting

Prioritization


AS YOU REQUESTED, I SORTED THE FUTURE PRODUCT FEATURES INTO THREE PRIORITIES.
www.dilbert.com scottadams@aol.com
ESSENTIAL
CRITICAL
MUST-HAVE
LET ME KNOW WHICH GROUP I SHOULD STOP WORKING ON.
8-24-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.
THIS IS THE PART WHERE YOU PRETEND TO ADD VALUE.

Source: Dilbert.com

Revenue Forecasting

feature_revenue_forecast = expected_lift x platform_factor x success_probability x platform_revenue

- **feature_revenue_forecast** is the expected revenue from the feature
- **expected_lift** is the increase in conversions we expect from users in the treatment group vs. users in the control group
- **platform_factor** is what percent of all users of the platform (whether iOS, android, mobile web, or desktop web) are part of the experiment
- **success_probability** is a haircut we apply to take into account that not all experiments will succeed
- **platform_revenue** is the total revenue generated by the platform. For example, the platform_revenue for iOS is the total revenue from orders placed via the iOS app.

ROI Calculation

ROI = feature_revenue_forecast / level_of_effort

ML Feature Development

Machine Learning

Discovery and personalization - Laura likes tacos, poke, and emoji pillows

Supply Intelligence - There are millions of merchants we could call at any time to get onto our platform…how do we pick the best ones?

Fraud prevention - Fighting the bad guys, in real time

Image recognition - Identify the best user-generated images with neural networks

Logistics - Get ahead of order rush by sending extra inventory to the warehouse in advance of big demand

Customer Service - AI-powered chatbots serve customers quickly using NLP & ML

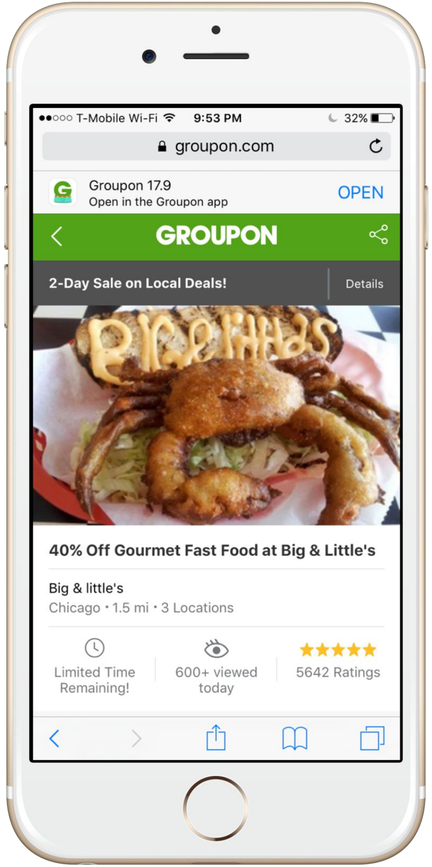
Plush Emoji Pillow

~~$10.99~~ $5.99

Image: Groupon.com

Discovery and personalization

- Personalize browse feed based on product views, clicks, purchases, and other features
- **Naïve Bayes** model used to predict the probability that a user will be interested in a particular deal
- **Collaborative filtering** used to group users with similar preferences together and personalize suggestions
- Freshness algorithm penalizes multiple reimpressions



ML Frameworks

2015: Duct tape and string

The task: Predict the potential $$ performance of every merchant that could run on Groupon

Implementation:

- ETLs! (**Extract**, **Transform**, **Load**)
- Tables built on tables built on tables, glued together with bash scripts and cron jobs
- Tightly coupled? You bet.

It worked! (most of the time)

...but most of the time is way worse than all of the time

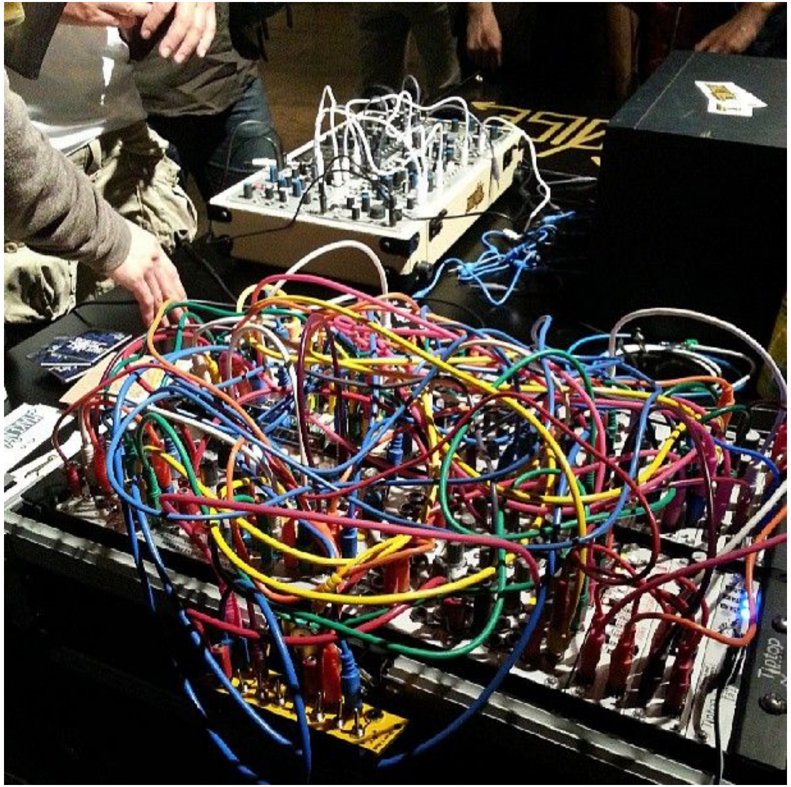


Image: Wikimedia Commons

Two Big Challenges

Clean Data in Production

How do we untangle the ETLs into separate features that we can monitor and quality-check independently?

- Subtle changes in a single data field can seriously impact model performance
- Nuances in your data set can look fine to tests, but fail in the real world

Swappable and Testable Models

How do we allow Data Scientists to test new versions of the model without rebuilding everything from scratch?

- It’s hard to test ML models deeply embedded in code
- Data Scientists have to throw the model over the wall to engineers to reimplement

Think like engineers! Separate the concerns, unite them with clean interfaces!

Solutions @ Groupon - QED

QED is “Quantum Engineered Data”— it’s an ETL management platform that reads data from any source and has built in cleaning, error correction, and anomaly detection

Tenets:

- Avoid monolithic ETLs with catastrophic failure scenarios
- Preserve clean data; make it available as a “feature catalog”
- Handles failures smartly—can we fall back to yesterday? Do we fail the entire process?
- Plugs into any source of truth—streams, warehouse tables, JSON endpoints
- Automatically measure accuracy and drift over time
- “Built-in” anomaly detection and alerting (e.g., monitoring number of null features)
- Treat data as a first-class citizen: Data source failures = production failures

Image credit: AppDynamics and TistaTech

Solutions @ Groupon - Flux

We built a generic, extensible machine learning platform called **Flux.**

Flux is the “Rosetta Stone” between data scientists and engineers

Keep production ML model in a state data scientists can easily understand

- Data scientists work primarily in R
- Python is the “glue” that connects R and Java
- Flux models written in Java and Clojure for stability and speed
- Run on Groupon’s large Hadoop cluster

Image: Wikimedia Commons

Experimentation

Experimentation @ Groupon Scale

- 100 teams running experiments
- 200 experiments running at a given time
- 2,500 total experiments run to date

2014: Mayhem

Photo credit: thetaxhaven / Flickr

2016: Finch Express

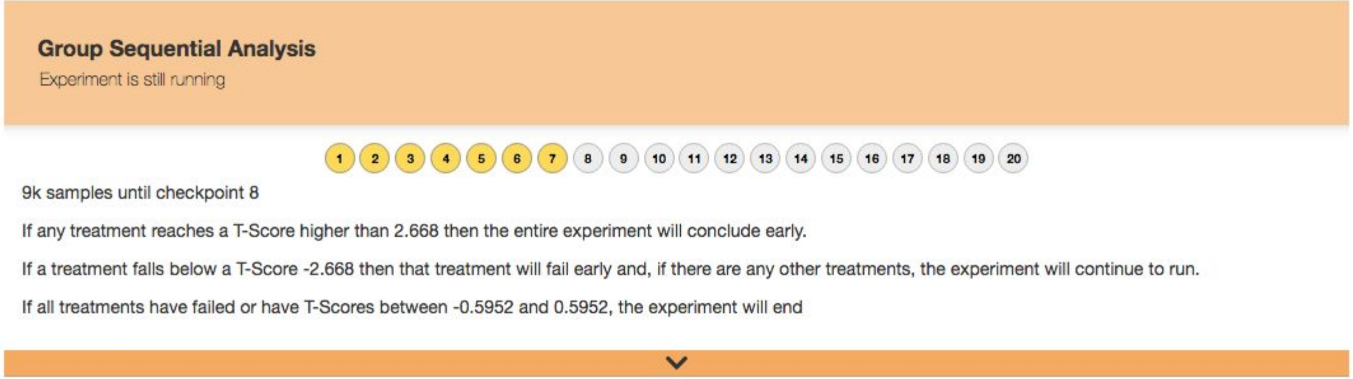
- Bespoke platform called "Finch Express"
- Dedicated engineering team ("Optimize")
- Ruby on Rails, Node.js, Ember.js, Python, R, and Hadoop/Hive

Photograph by Chris Murphy

Finch Express

- Support for code-less experiments
- Dynamic lift sensitivity analysis
- Automatic analysis
- Auto rollout & auto rollback
- Mix-shift detection
- Store key lessons for future generations
- Peeking Prevention
- Group Sequential Analysis

Group Sequential Analysis



Group Sequential Analysis

Experiment is still running

9k samples until checkpoint 8

If any treatment reaches a T-Score higher than 2.668 then the entire experiment will conclude early.

If a treatment falls below a T-Score -2.668 then that treatment will fail early and, if there are any other treatments, the experiment will continue to run.

If all treatments have failed or have T-Scores between -0.5952 and 0.5952, the experiment will end

- Goal: Minimize downside risk & maximize upside opportunity
- α spending function allows statistically rigorous “peeking” at designated checkpoints
- No need to spend all our α at the end! We can budget it.
- Results: Experiments concluded an average of 57.5% earlier compared to single checkpoint
- Pioneered in heart valve clinical trials (Lan & DeMets 1983)

No more mayhem (well, less anyway)

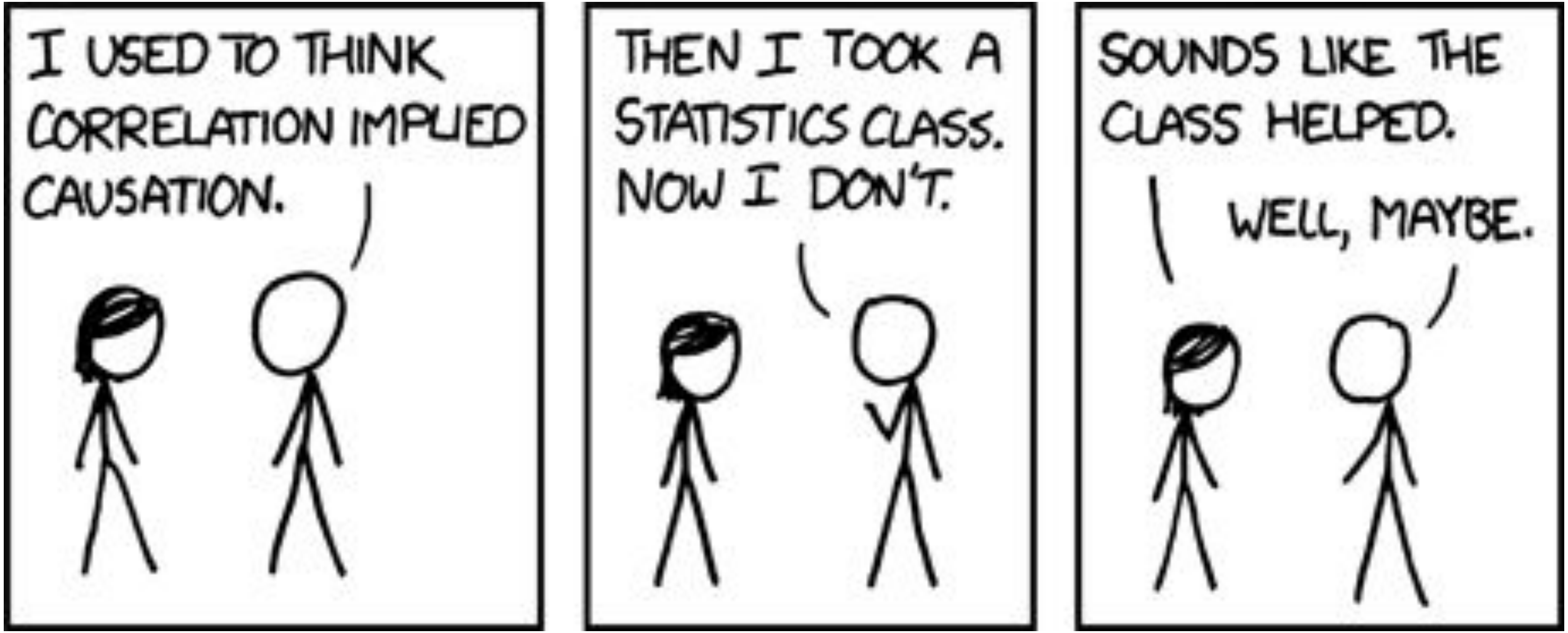

I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Image credit: XKCD

“A/B testing is table stakes for any internet or mobile business.”

The circle completes

- Capitalize on past learnings to inform future iterations
- Winners are exciting
- Big losers are exciting too!
- Failure embraced as part of the process
- Apply 50% incrementality haircut to successes when feeding into forecasts

Questions?