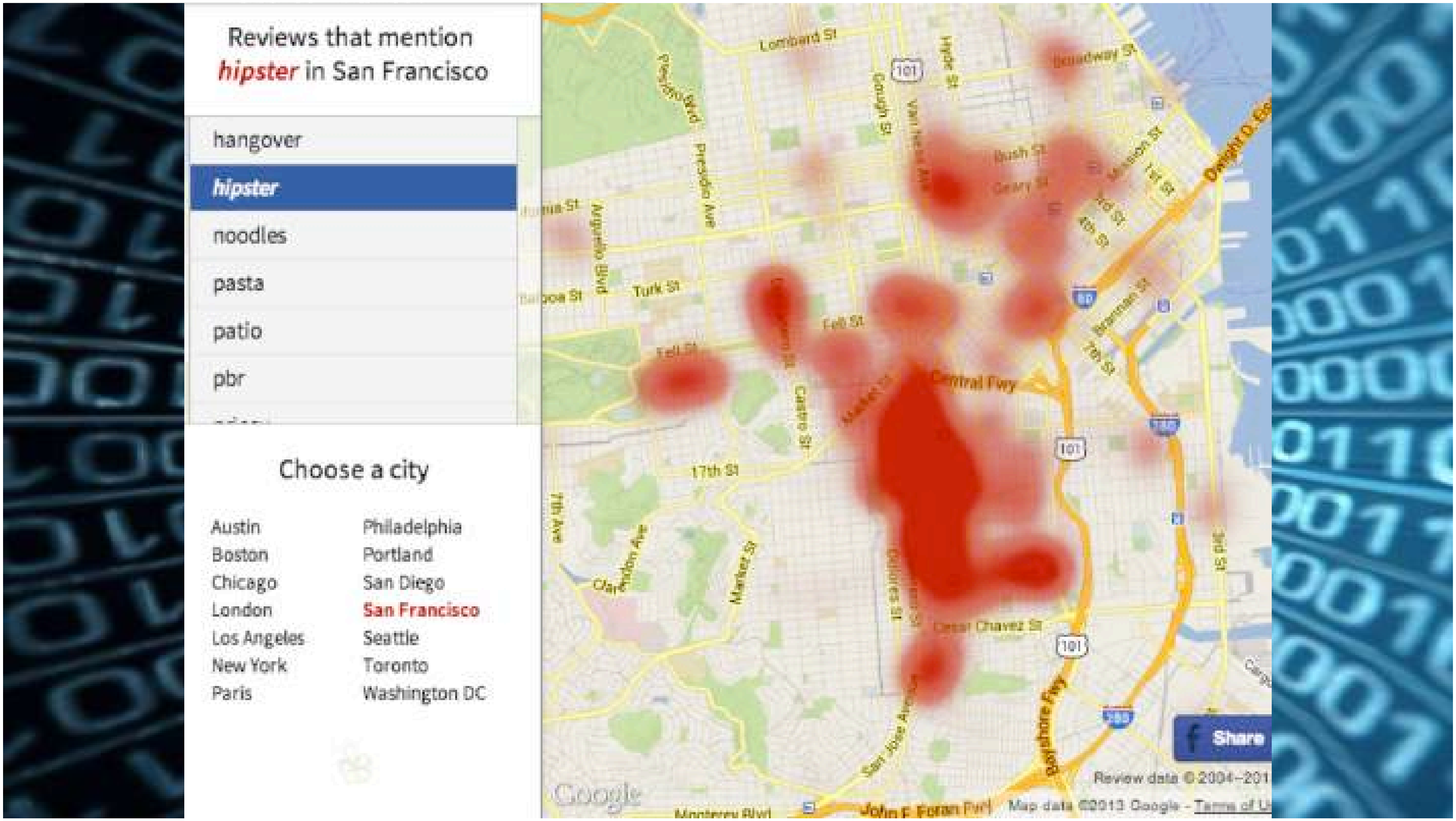
Data Science in a Spreadsheet

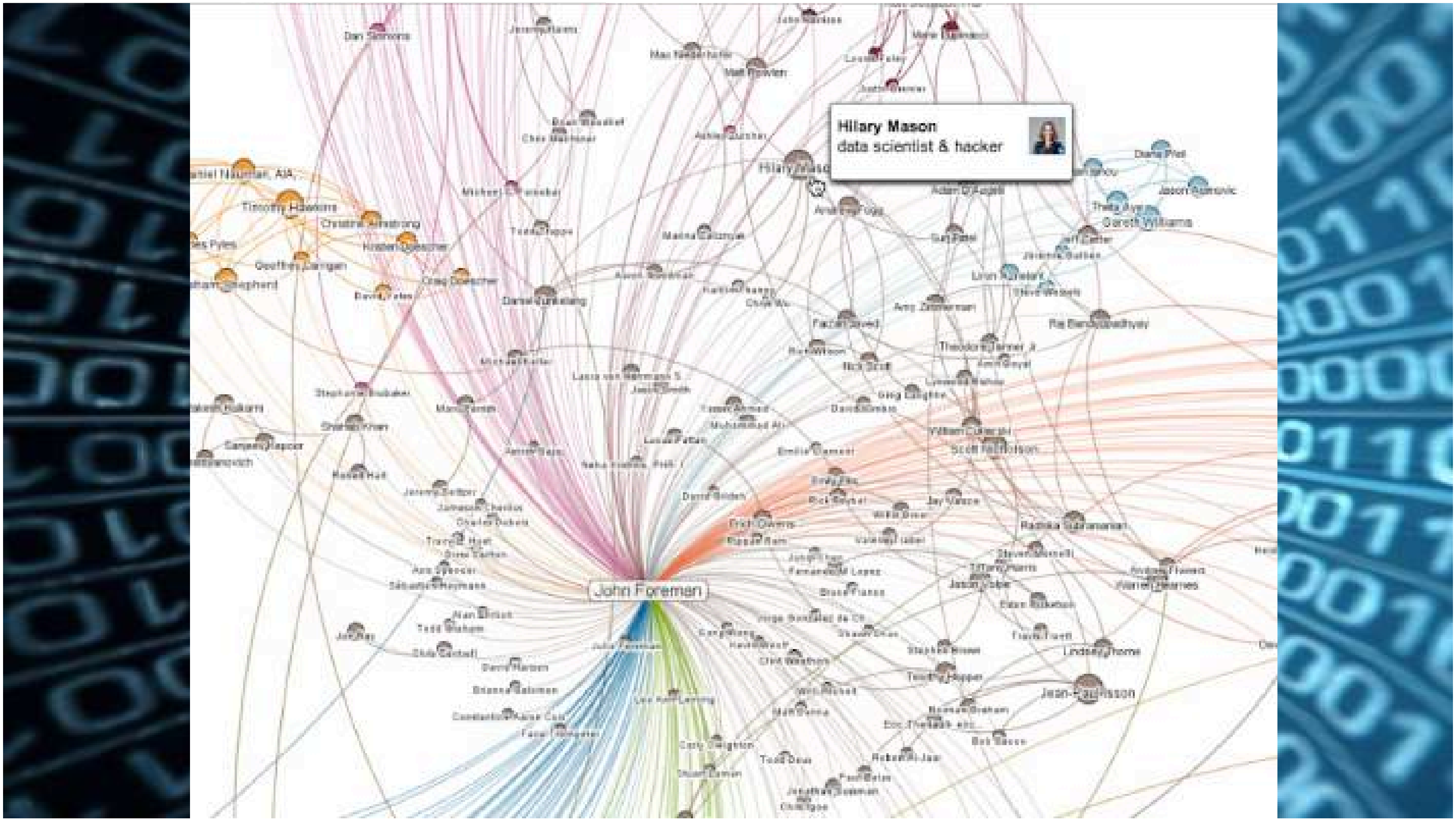
John Foreman
Data Scientist, MailChimp.com

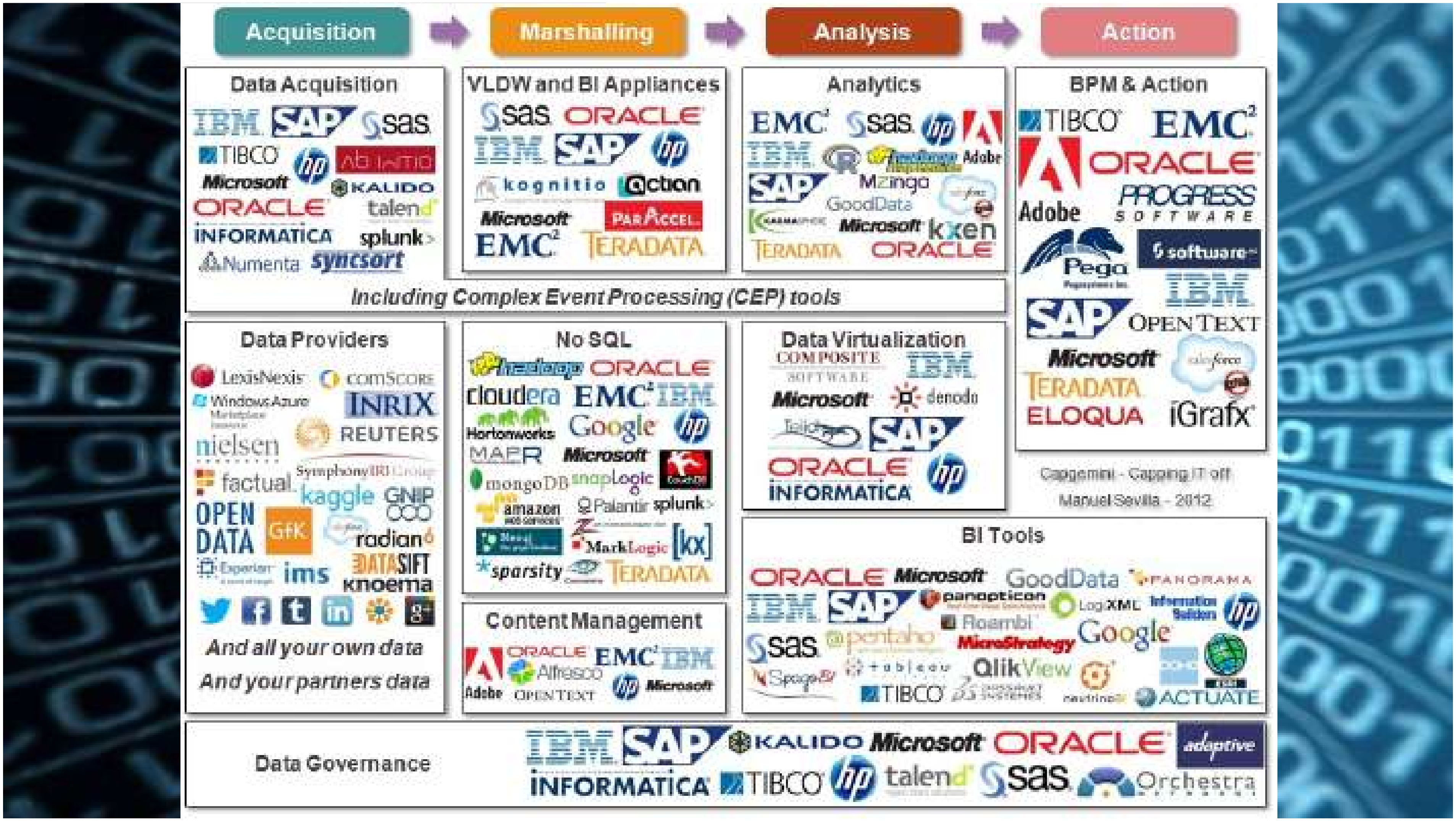
2014


We should be doing big data.

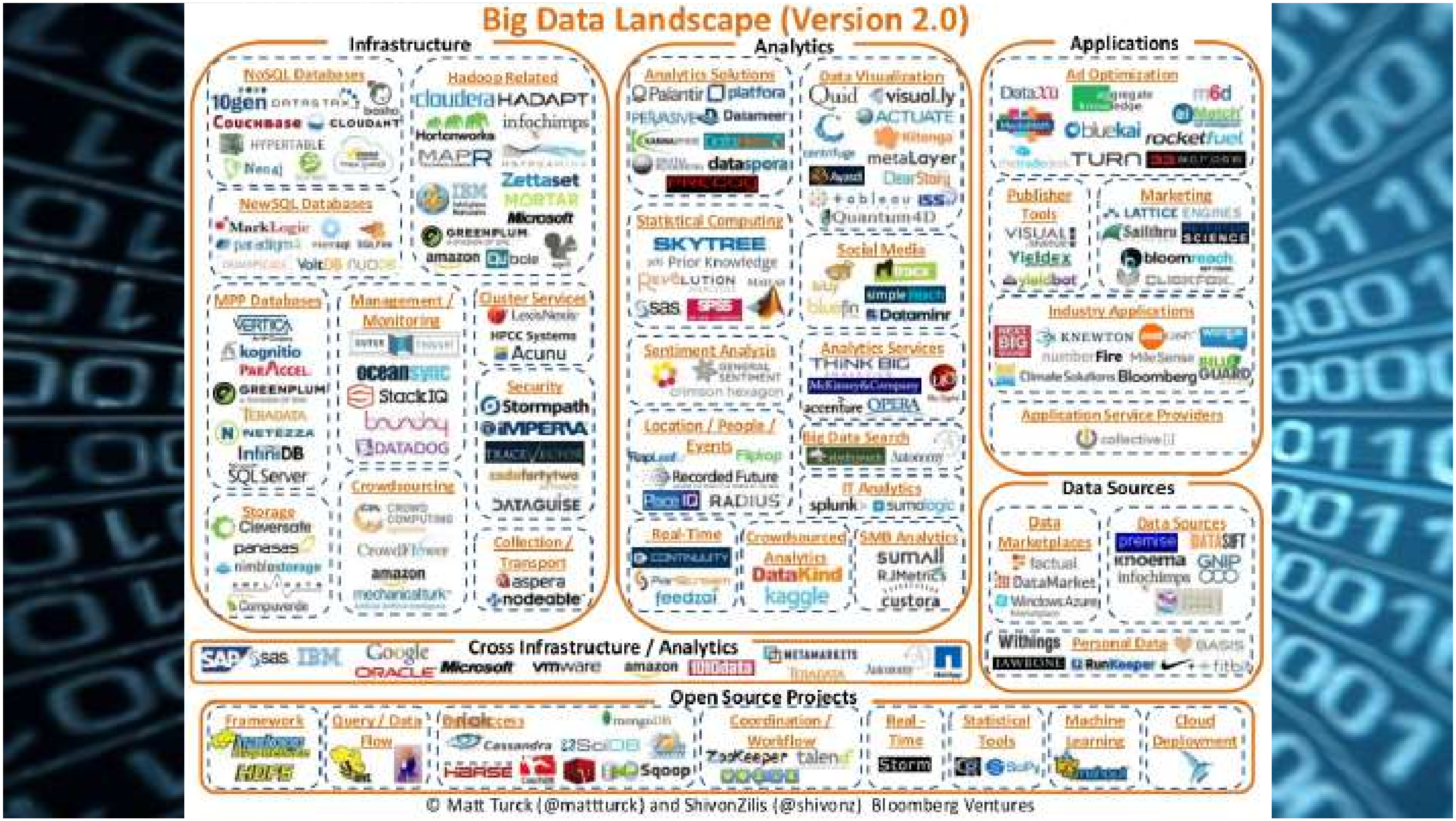


Reviews that mention
hipster in San Francisco
hangover
hipster
noodles
pasta
patio
pbr
Choose a city
Austin
Boston
Chicago
London
Los Angeles
New York
Paris
Philadelphia
Portland
San Diego
San Francisco
Seattle
Toronto
Washington DC
Lombard St
Broadway St
Bush St
Geary St
Turk St
Fell St
Central Fwy
17th St
Cesar Chavez St
Share
Map data ©2013 Google


Hilary Mason
data scientist & hacker
John Foreman

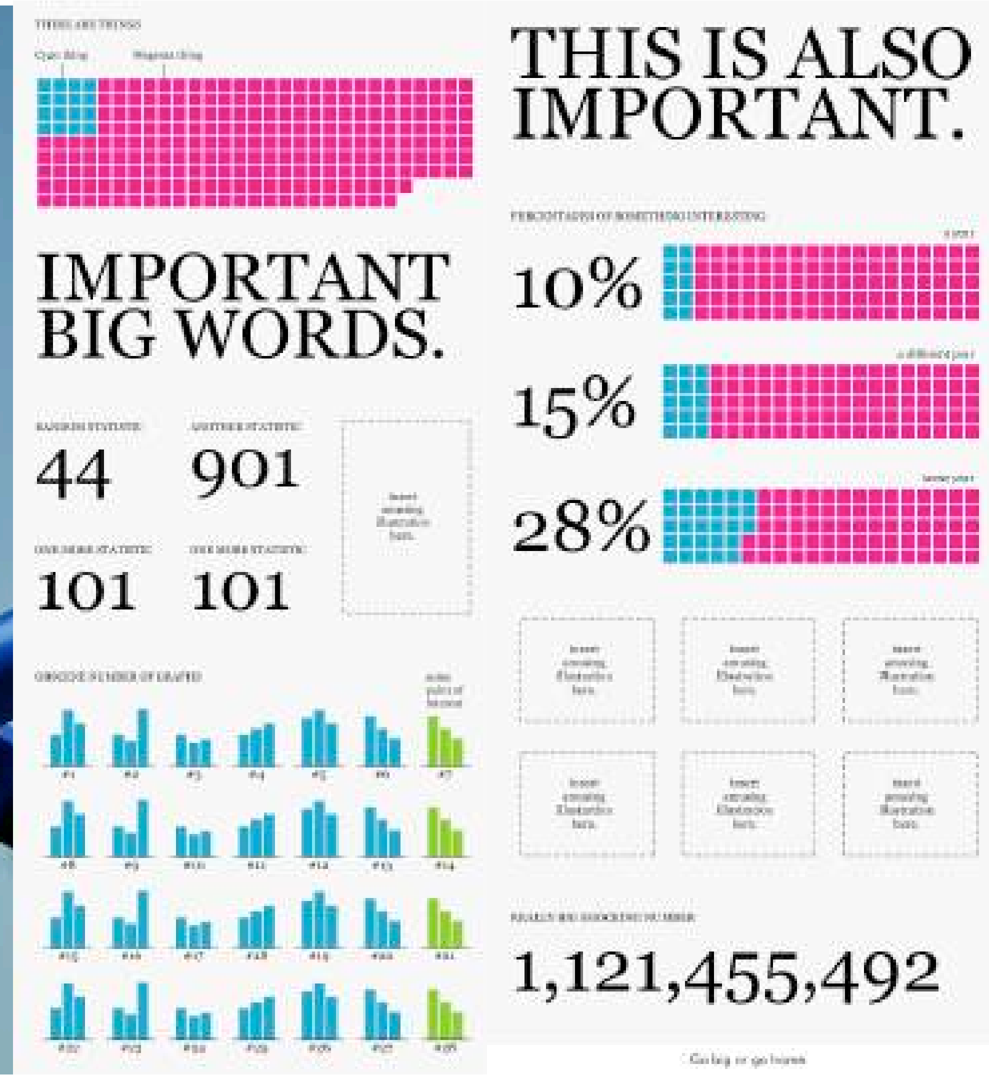


Acquisition
Marshalling
Analysis
Action
Data Acquisition
VLDW and BI Appliances
Analytics
BPM & Action
Including Complex Event Processing (CEP) tools
Data Providers
No SQL
Data Virtualization
And all your own data
And your partners data
Content Management
BI Tools
Capgemini - Capping IT off
Manuel Sevilla - 2012
Data Governance


Big Data Landscape (Version 2.0)
Infrastructure
NoSQL Databases
Hadoop Related
NewSQL Databases
MPP Databases
Management / Monitoring
Cluster Services
Security
Storage
Crowdsourcing
Collection / Transport
Analytics
Analytics Solutions
Data Visualization
Statistical Computing
Social Media
Sentiment Analysis
Analytics Services
Location / People / Events
Big Data Search
IT Analytics
Real-Time
Crowdsourced Analytics
SMB Analytics
Applications
Ad Optimization
Publisher Tools
Marketing
Industry Applications
Application Service Providers
Data Sources
Data Marketplaces
Data Sources
Personal Data
Cross Infrastructure / Analytics
Open Source Projects
Framework
Query / Data Flow
Databases
Coordination / Workflow
Real-Time
Statistical Tools
Machine Learning
Cloud Deployment
© Matt Turck (@mattturck) and ShivonZilis (@shivonz) Bloomberg Ventures

- Choose tools first
- Know a fraction of what’s possible
- Flail about

- Choose tools first
- Know a fraction of what's possible
- Flail about
- Create Infographic



- Know what's possible
    - Data
    - Techniques
    - Technologies
- Identify problems & opportunities
- Choose what solves the problem

- Know what’s possible
  - Data
  - Techniques
  - Technologies
- Identify problems & opportunities
- Choose what solves the problem

What are we doing here?

- We’re here to learn to:
  - Differentiate
  - Prototype
- Why Excel?

WARNING: I’ve never done this before. There will be stammering. We’ll get through it.

WARNING #2: There’s math ahead. And formulas.

Agenda

- Supervised Machine Learning
- Forecasting
- Optimization

- FOR LATER:
  - @John4man
  - John.4man@gmail.com

Naïve Bayes

- A naïve Bayes model is a **supervised AI model**
  - Takes in past data (in our case, word usage by category) and uses it to classify future observations
  - But in order to use naïve Bayes, we need to learn probability

Introduction to Probability

- *p(Michael Bay's next film will be terrible) = 1 = 100%*
- *p(I eat wings today) = .5 = 50%*

Conditional Probability

- *p(John Foreman will ever go vegan) = 0.0000001*
- *p(John Foreman will go vegan | you pay him* $1B) = *1*

Law of Total Probability

- *p(vegan) = p($1B) * p(vegan | $1B) +*
  p(not $1B) p(vegan | not $1B)*

- *p(vegan) = 0*1 +*
  1.0000001* = .0000001

Joint Probability

- *p(John eats Taco Bell) = .2*
- *p(John listens to cheesy electronic music) = .8*
- *p(John eats Taco Bell, John listens to cheese)* = ?

The Chain Rule:

- *p(John eats Taco Bell, John listens to cheese)*
  = *p(John eats Taco Bell) **
  p(John listens to cheese | *John eats Taco Bell)*

But these are independent:

- *p(John listens to cheese* | *John eats Taco Bell) =*
  p(John listens to cheese)
- ***p(John eats Taco Bell, John listens to cheese) =***
  p(John eats Taco Bell) * p(John listens to cheese) *=.2*.8 = .16*

What happens in a dependent situation?

- *p(John listens to cheese) = .8*
- *p(John listens to Depeche Mode) = .3*
- *p(John listens to cheese | John listens to Depeche Mode) = 1*
- *p(John listens to cheese, John listens to DM) =*
  *p(John listens to Depeche Mode) * p(John listens to cheese |*
  John listens to Depeche Mode) =
  *.3 * 1*
  = .3

Bayes' Rule

- *p(John listens to cheese) = .8*
- *p(John listens to Depeche Mode)* = .3
- *p(John listens to cheese | John listens to Depeche Mode) = 1*

But what about:

- *p(John listens to Depeche Mode | John listens to cheese)*

Bayes' Rule allows you to flip what you know around:

- $p(b)*p(a \mid b) = p(a)*p(b \mid a)$
- $\boldsymbol{p(a \mid b) = p(a) * p(b \mid a) / p(b)}$
- $p(DM \mid cheese) = p(DM) * p(cheese \mid DM) / p(cheese)$
  $= .3 * (1/.8) = .375$

Using Bayes Rule to create an AI model

*We care about **comparing**:*

- *p(app | word1, word2, word3, ...)*
- *p(other | word1, word2, word3, ...)*

Bayes:

- *p(app | word1, word2, ...) =*
  p(app) p(word1, word2, ...| app) /
  p(word1, word2, ...)
- *p(other | word1, word2, ...) =*
  p(other) p(word1, word2, ...| other) /
  p(word1, word2, ...)

Drop the denominator!

Using Bayes' Rule to create an AI model

Let's get stupid and compare:

- *p(app) p(word1, word2,* ...| *app)* =
  p(app) p(word1| *app) p(word2*| *app) p(word3*| *app)…*
- *p(other) p(word1*| *other) p(word2*| *other) p(word3*| *other)…*

High-level class probabilities are often assumed to be equal.

So we need only compare:

- *p(word1*| *app) p(word2*| *app)* ... >=
  p(word1| *other) p(word2*| *other)* …

Using Bayes Rule to create an AI model

So what is p(word | app):

- p(“spark” | app) = sum of “spark” in training app tweets divided by total number of words in app tweets

Rare Words

p(word1| app) p(word2| app) ... >=
p(word1| other) p(word2| other) …

But what if we've never seen one of the words? That's a problem. (Shortened links, new handles, etc.)

The solution: Additive smoothing.

- Give it a 1.
- And add 1 to all the counts!

Floating point underflow

p(word1| app) p(word2| app) ... = .00001 * .000073 * .0000002 * … = BARF

Instead, take the log:

ln(p(word1| app) p(word2| app) ...) =

ln(p(word1| app)) + ln(p(word2| app)) … =

-11.5 + -9.5 + … = A nice looking negative number

Everybody take a break

Stretch

Drink coffee

Escape out the back

Forecasting (and a little simulation and optimization)

Forecasting is a lot like machine learning. Take past data and turn it into a future prediction. E.g. demand, supply, weather, population …

In machine learning though, usually you have lots of features. In forecasting, you generally only have a **time series**. A time series is a collection of values over time:
(80s: 2 comic book movies, 90s: 10 comic book movies, 00s: 111 comic book movies, …)

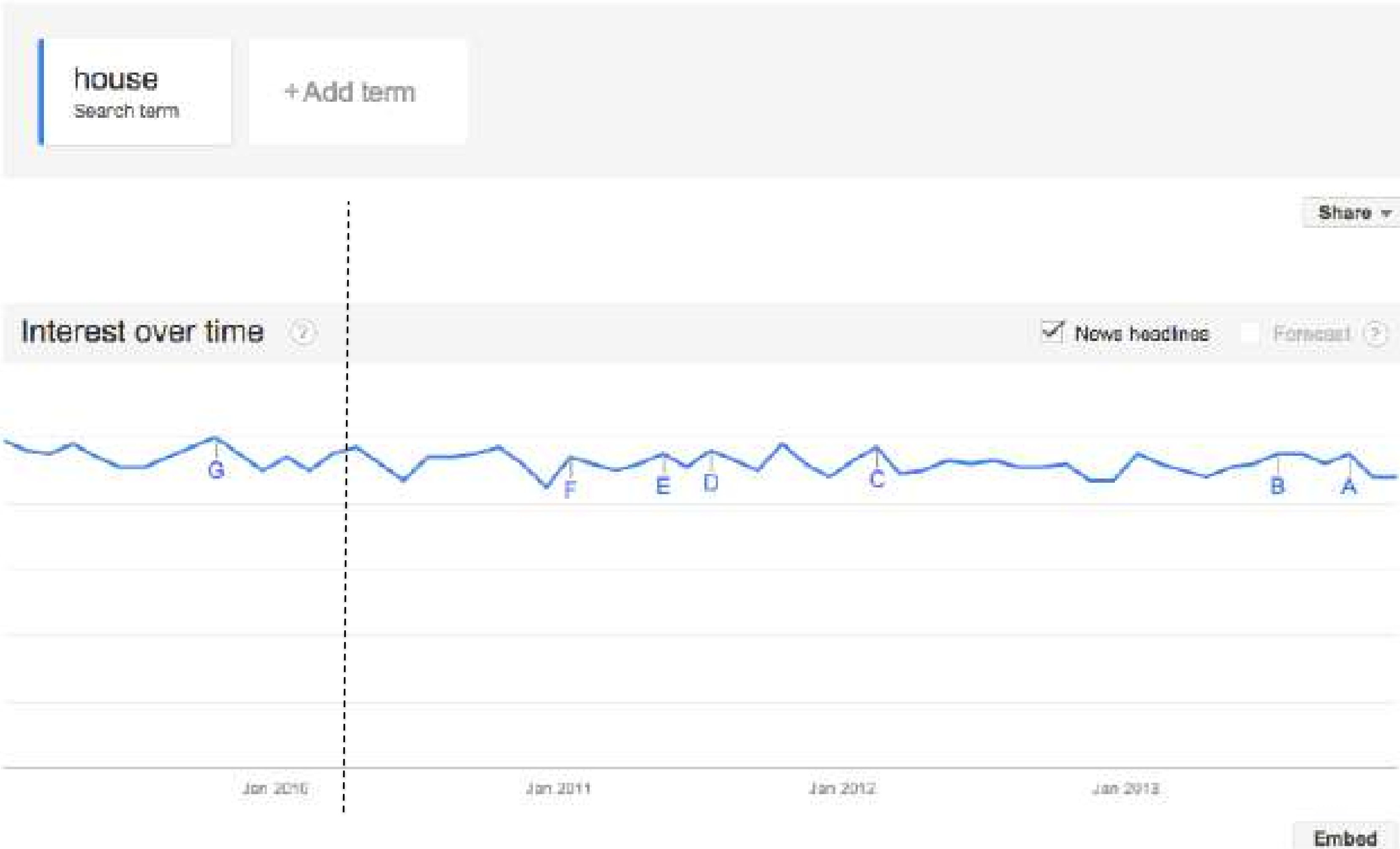
Time series analysis has been around for forever, but it's gotten some new life thanks to Google, Twitter, etc. (trend and anomaly detection)

Exponential smoothing

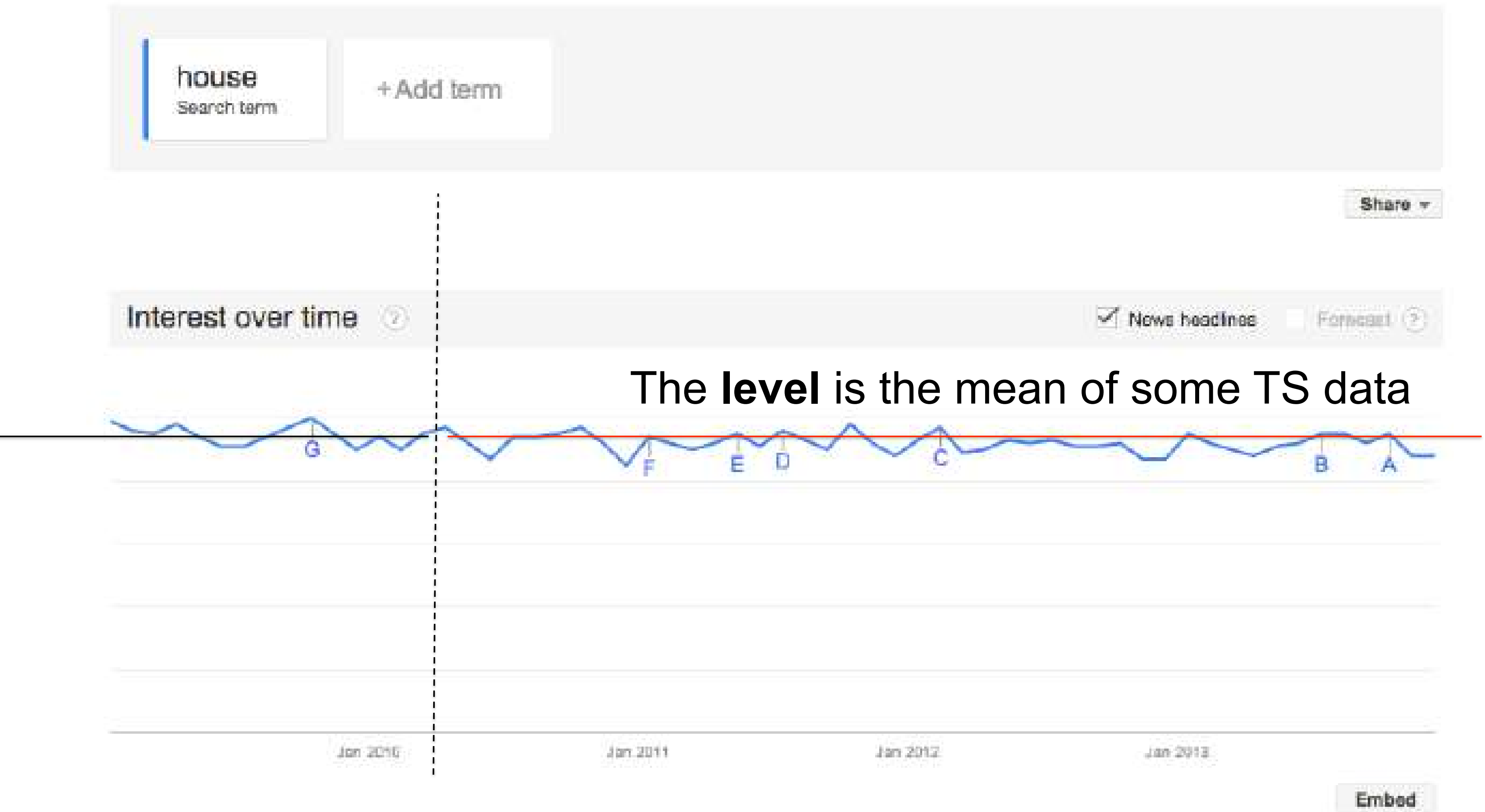
One of the best ways to forecast is via a technique called **exponential smoothing.** In exponential smoothing, you decompose the time series and then use its components to project out.

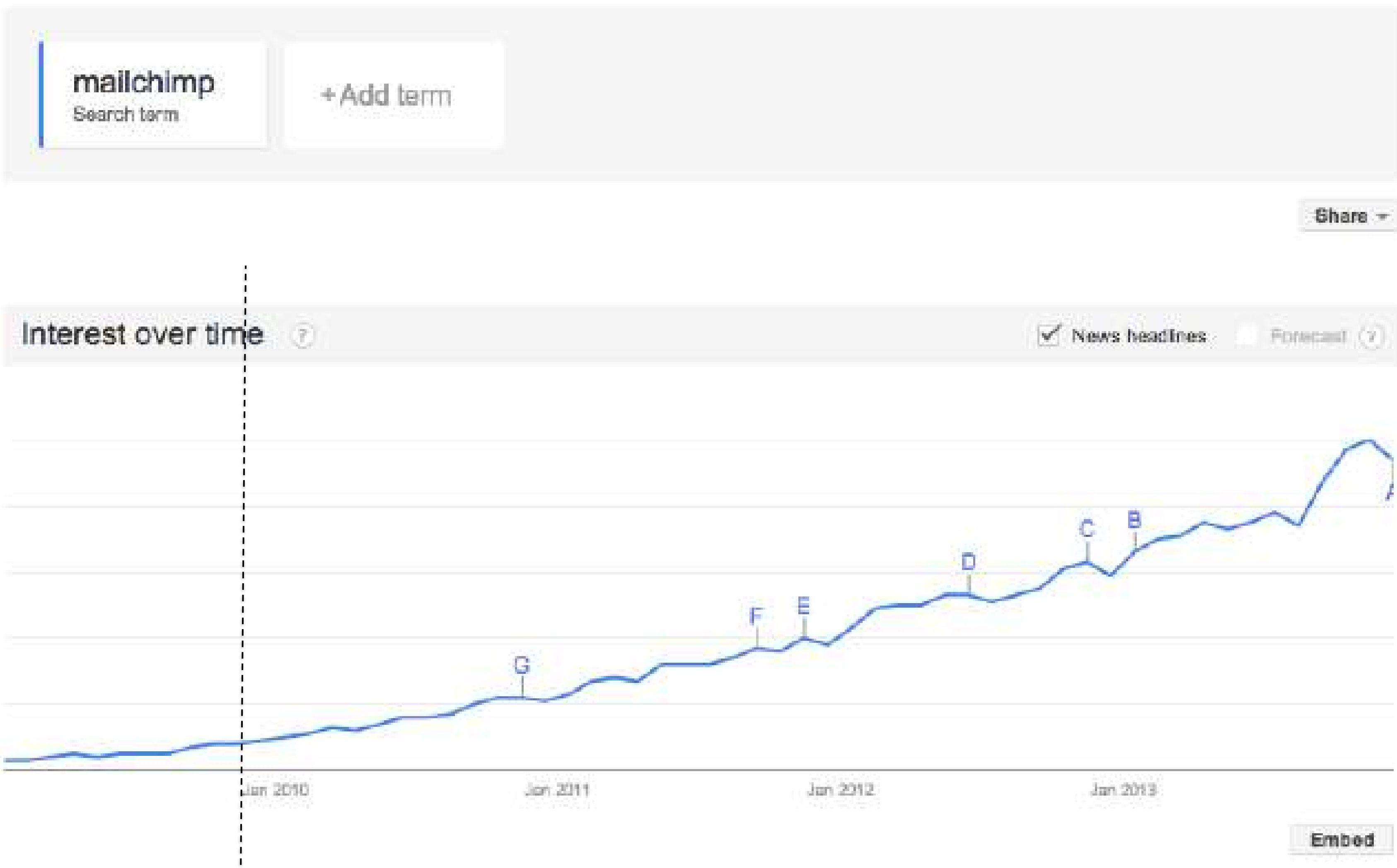
Today we'll learn **Triple Exponential Smoothing with Multiplicative Seasonality.**

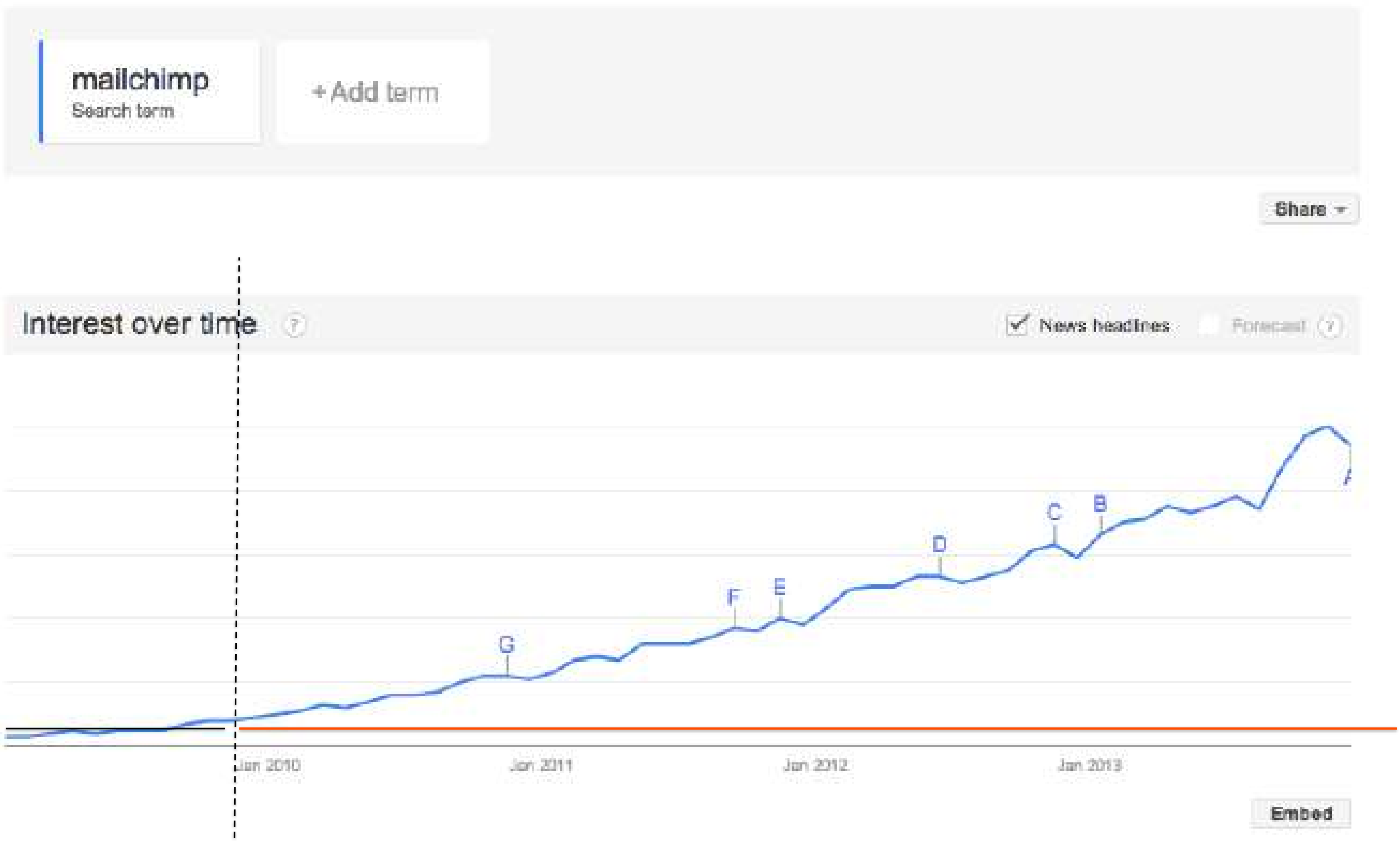
Exponential Smoothing is an industry standard technique used by Fortune 500s and start-ups alike…and it can be implemented in a spreadsheet


house
Search term
+ Add term
Share
Interest over time
News headlines
Forecast
G
F
E
D
C
B
A
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Embed

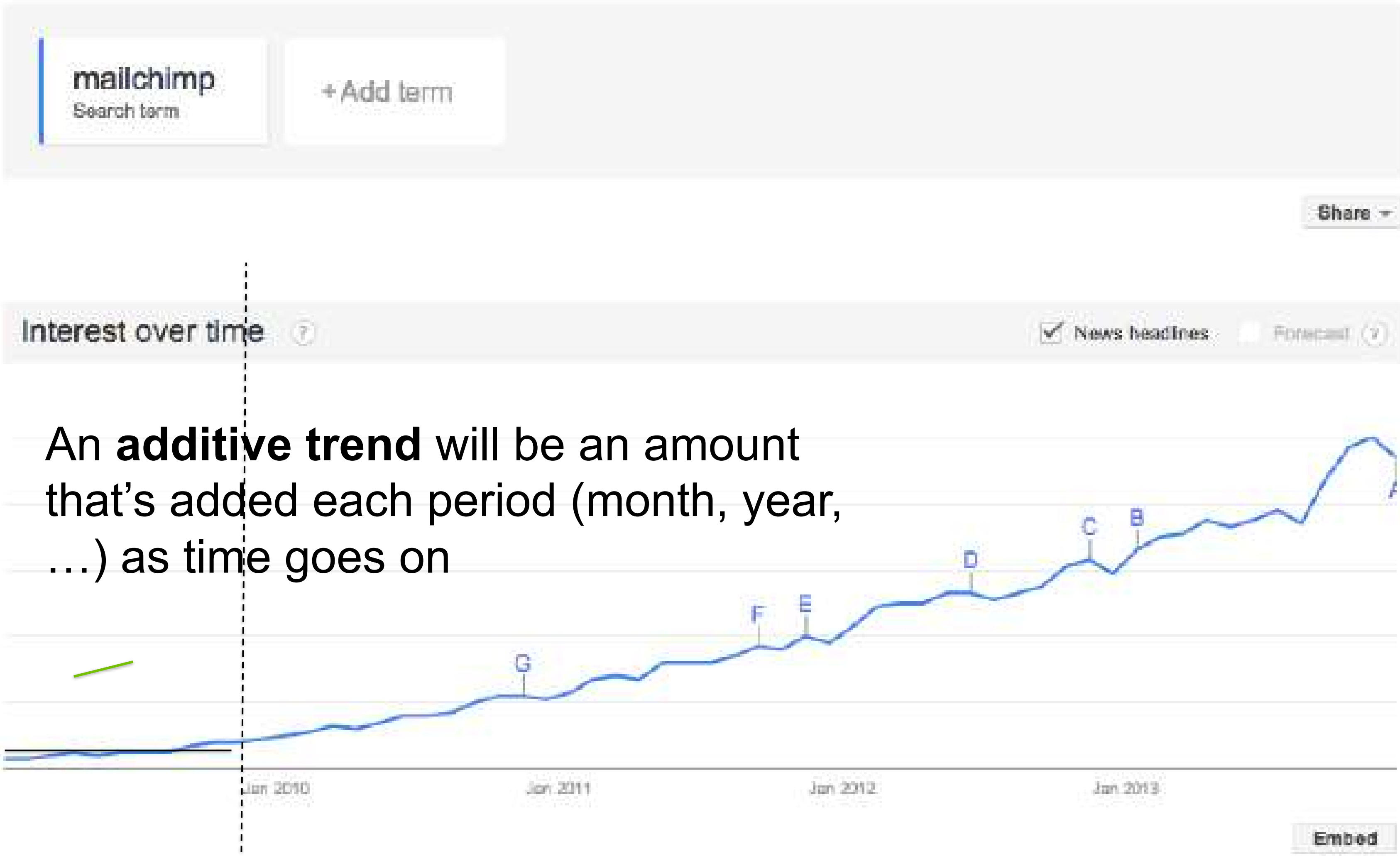
The **level** is the mean of some TS data



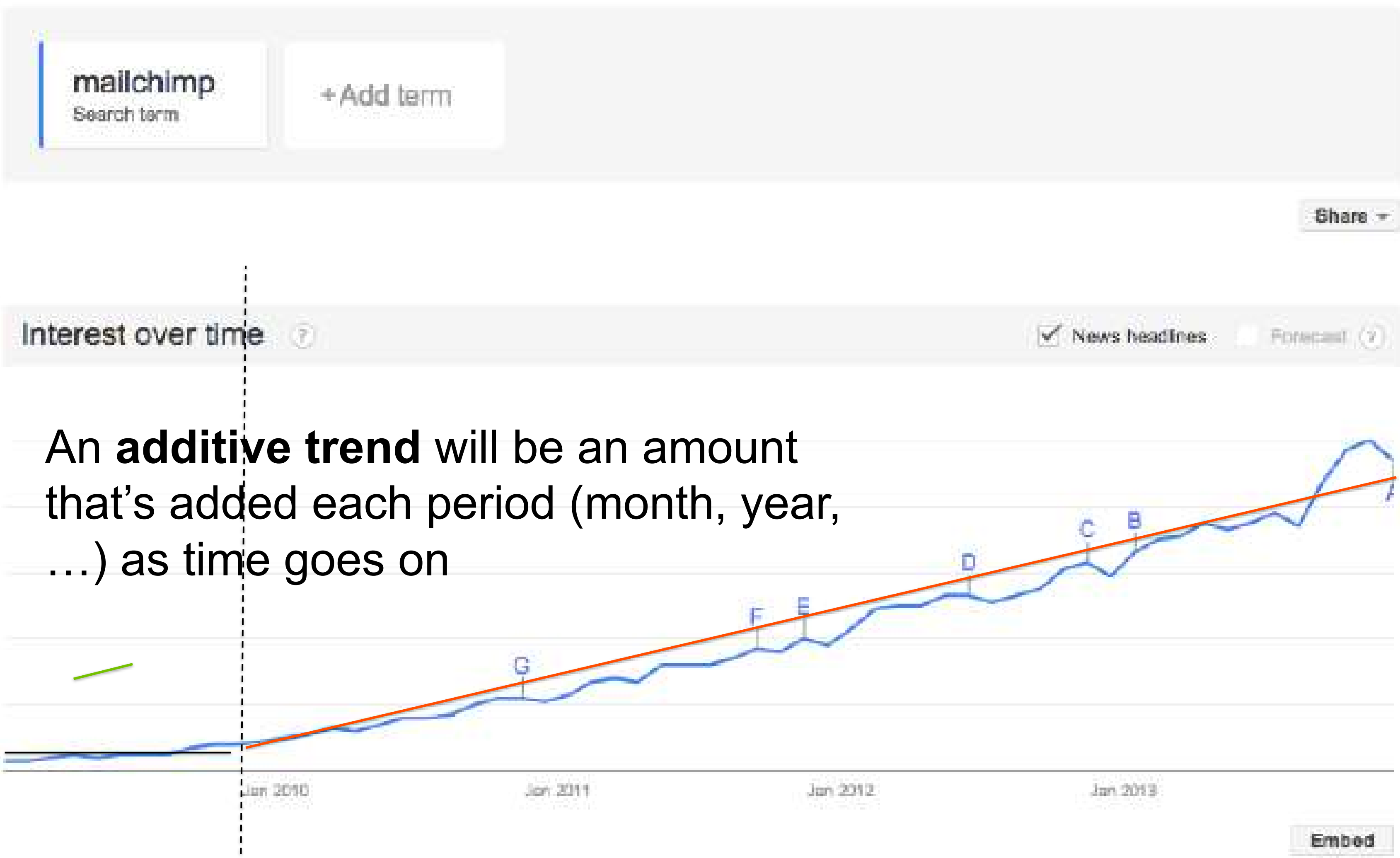

mailchimp
Search term
+ Add term
Share
Interest over time
News headlines
Forecast
G
F
E
D
C
B
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Embed



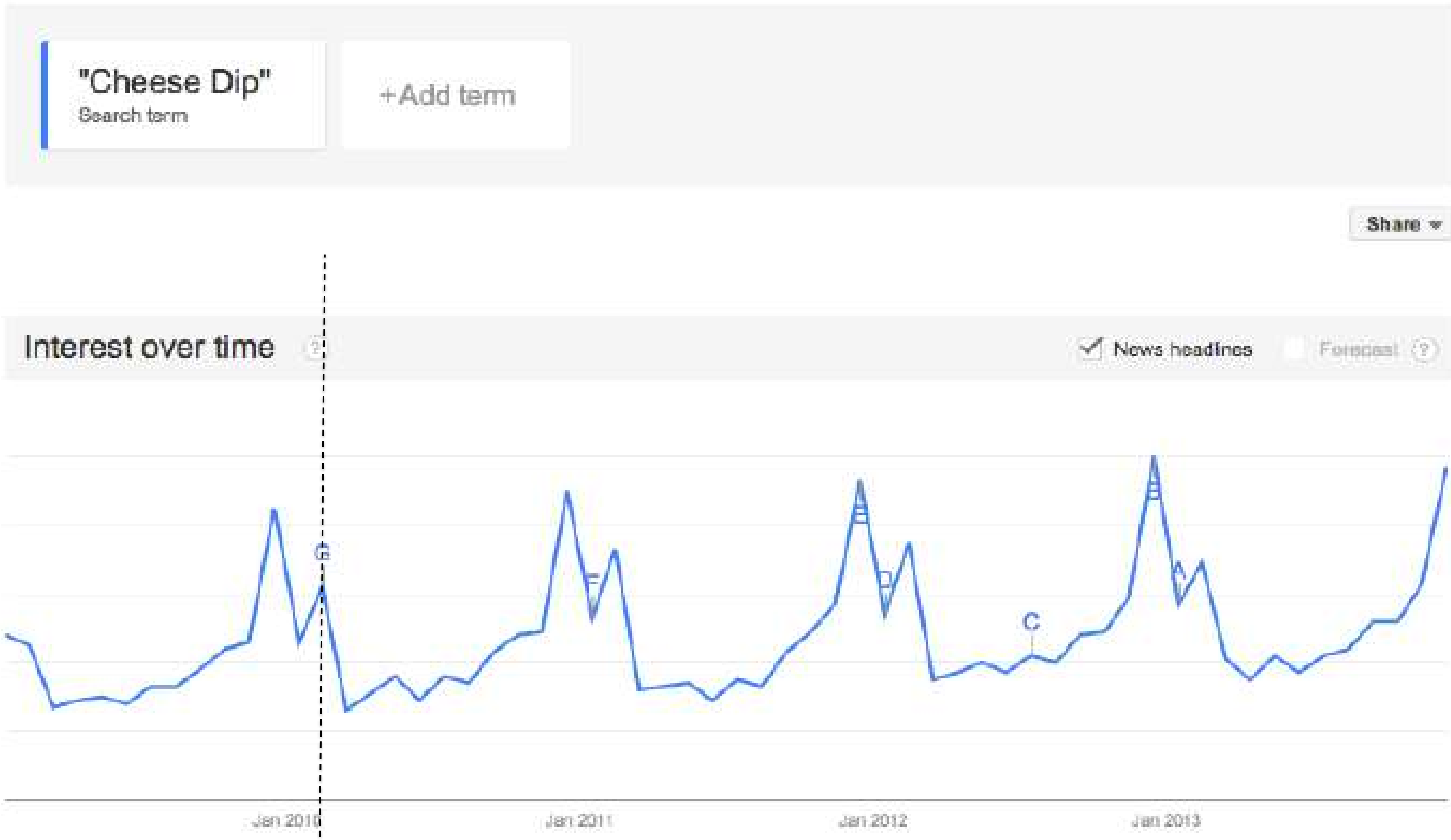
mailchimp
Search term
+Add term
Share
Interest over time
News headlines
Forecast
G
F
E
D
C
B
Jan 2010
Jan 2011
Jan 2012
Jan 2013
Embed

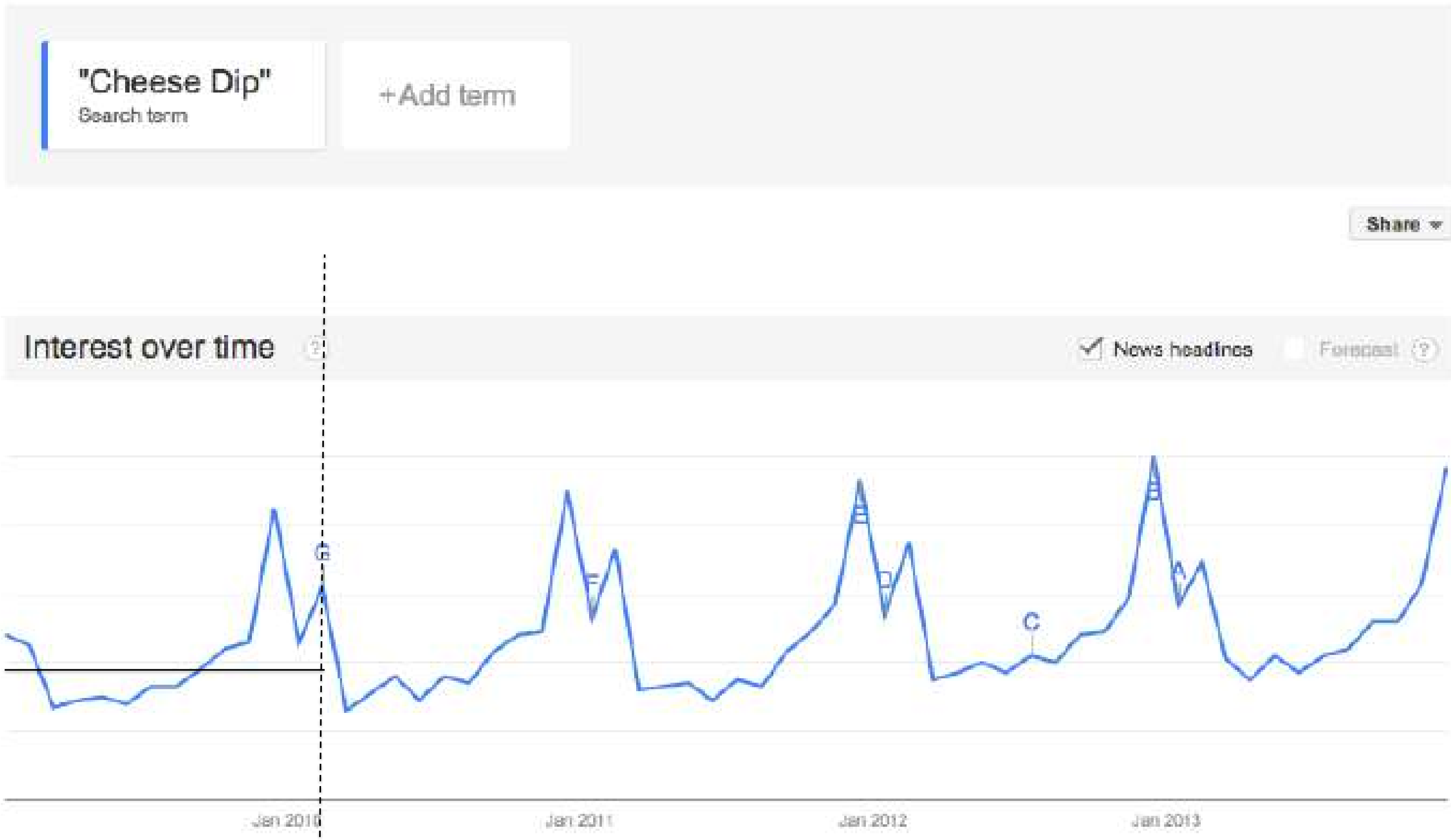


An **additive trend** will be an amount that's added each period (month, year, …) as time goes on

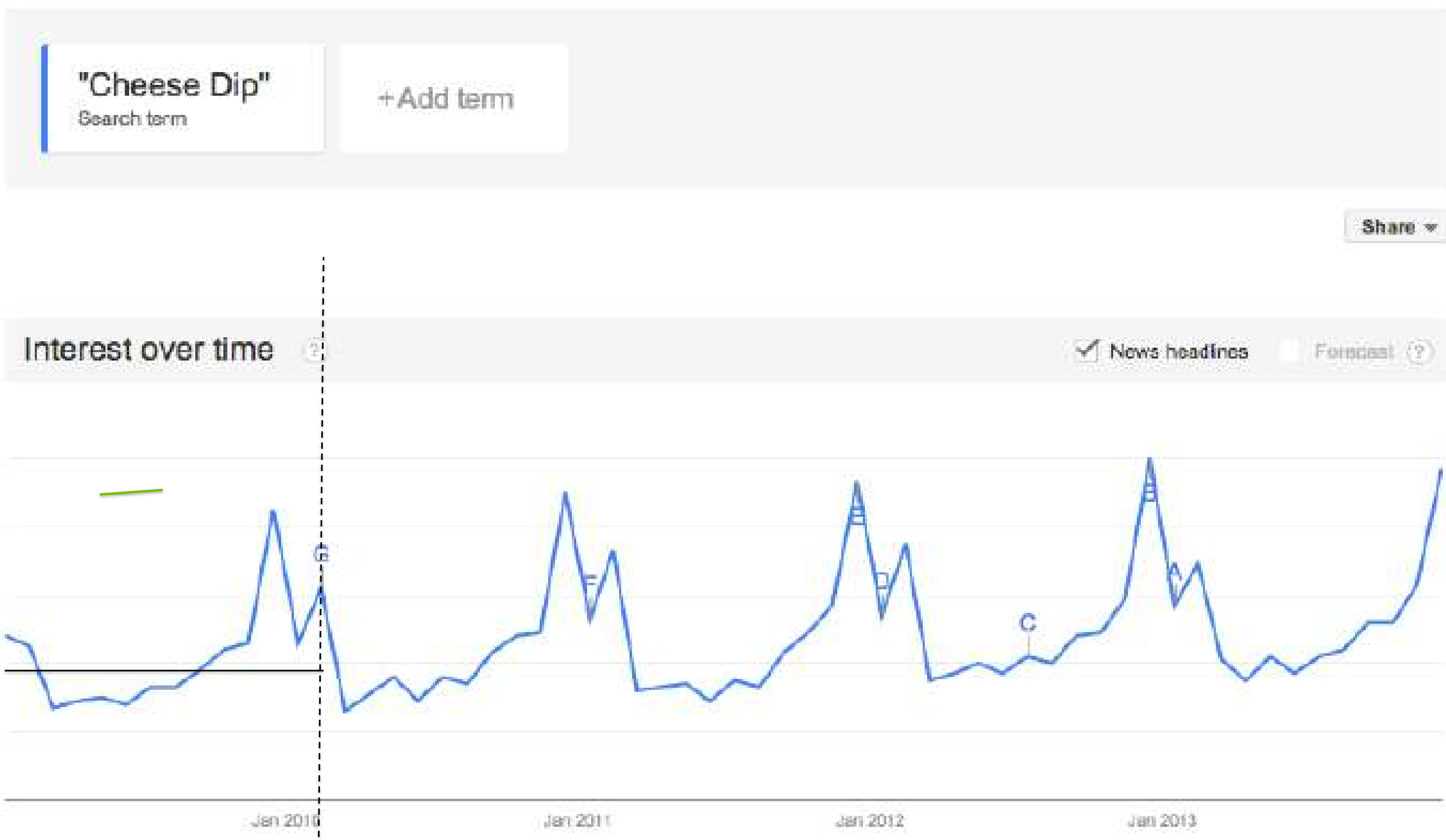


An **additive trend** will be an amount that's added each period (month, year, …) as time goes on

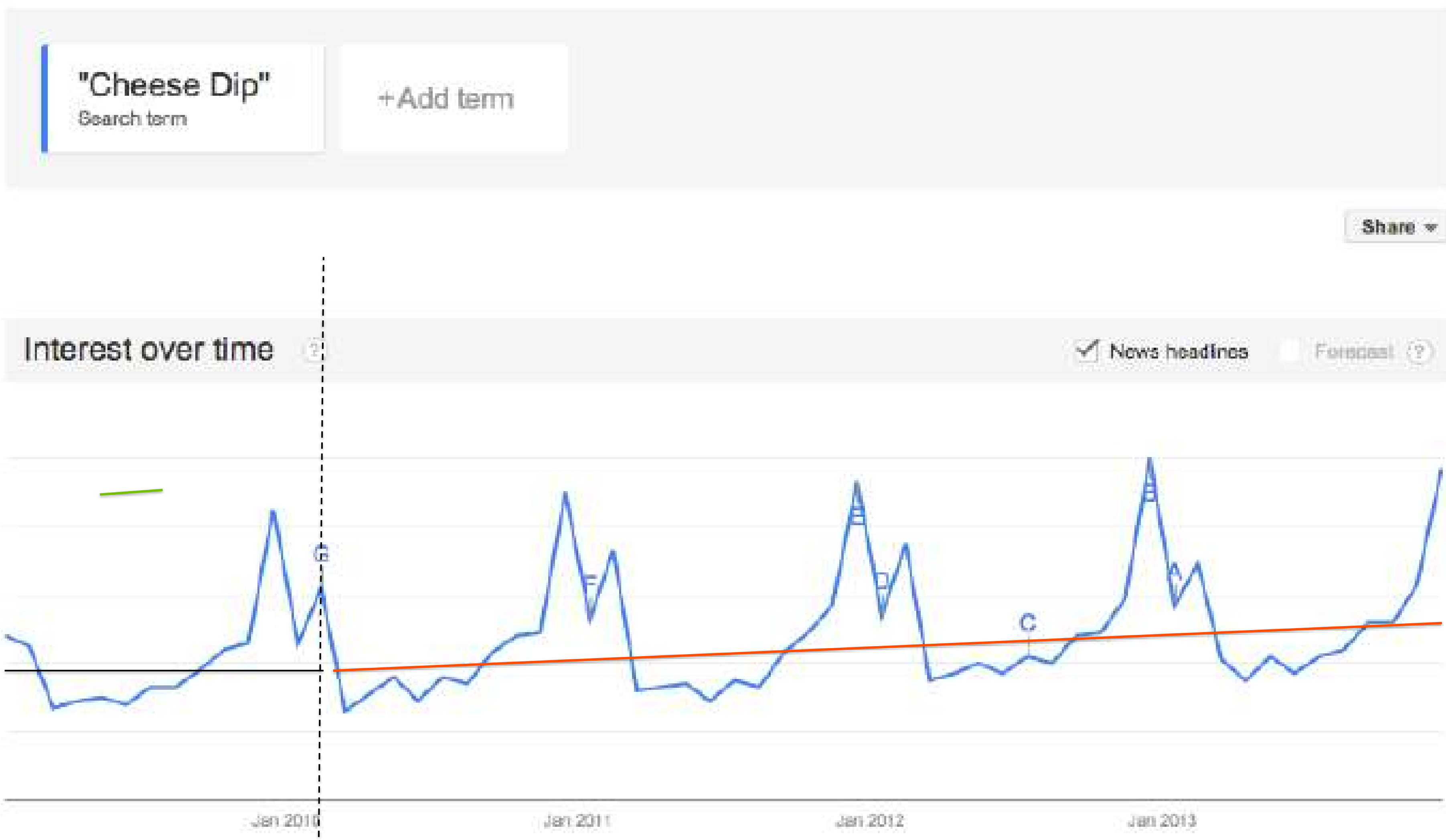

"Cheese Dip"
Search term
+Add term
Share
Interest over time
News headlines
Forecast
Jan 2010
Jan 2011
Jan 2012
Jan 2013



"Cheese Dip"
Search term
+Add term
Share
Interest over time
News headlines
Forecast
Jan 2010
Jan 2011
Jan 2012
Jan 2013

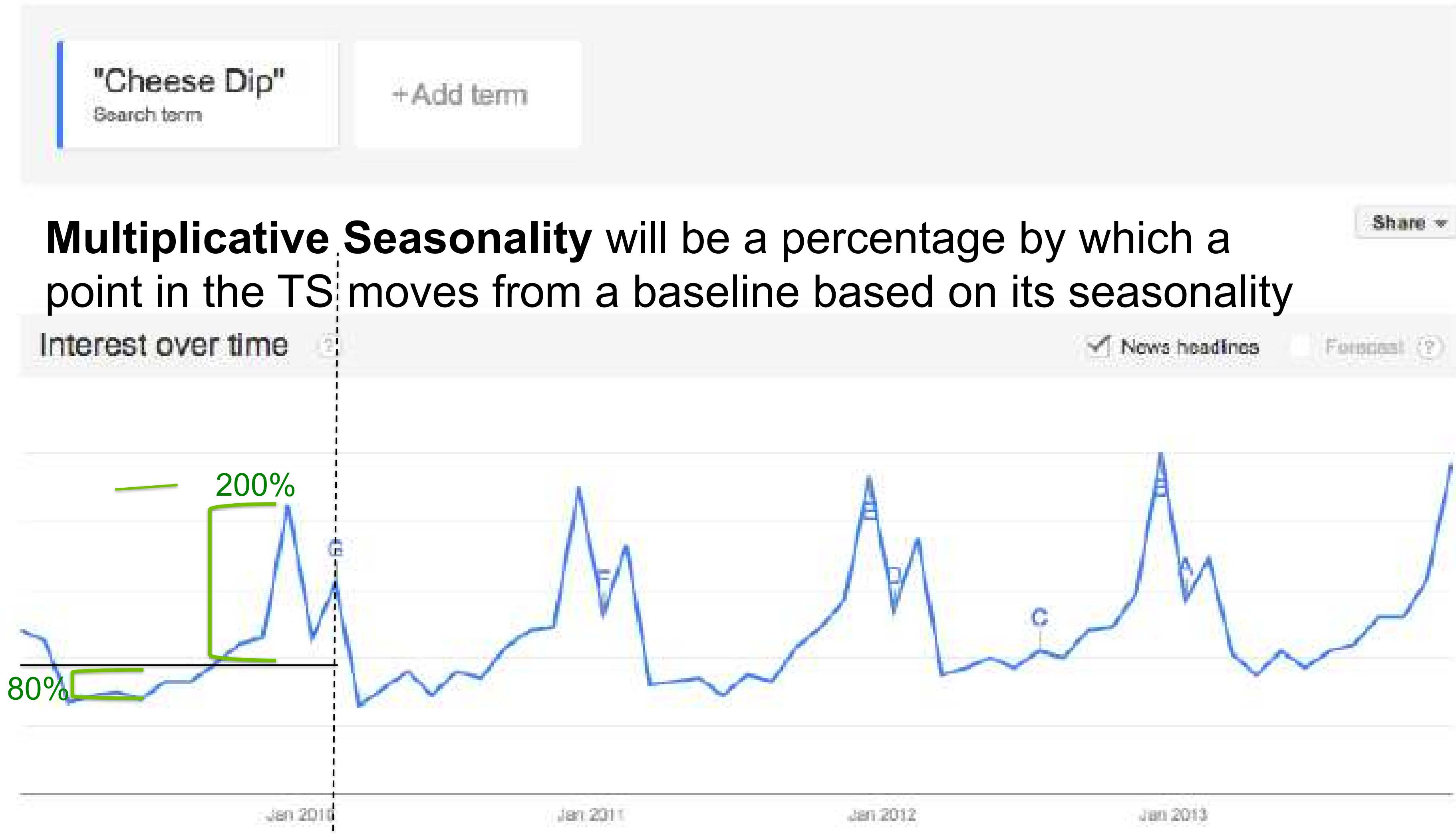


"Cheese Dip"
Search term
+Add term
Share
Interest over time
News headlines
Forecast
Jan 2010
Jan 2011
Jan 2012
Jan 2013

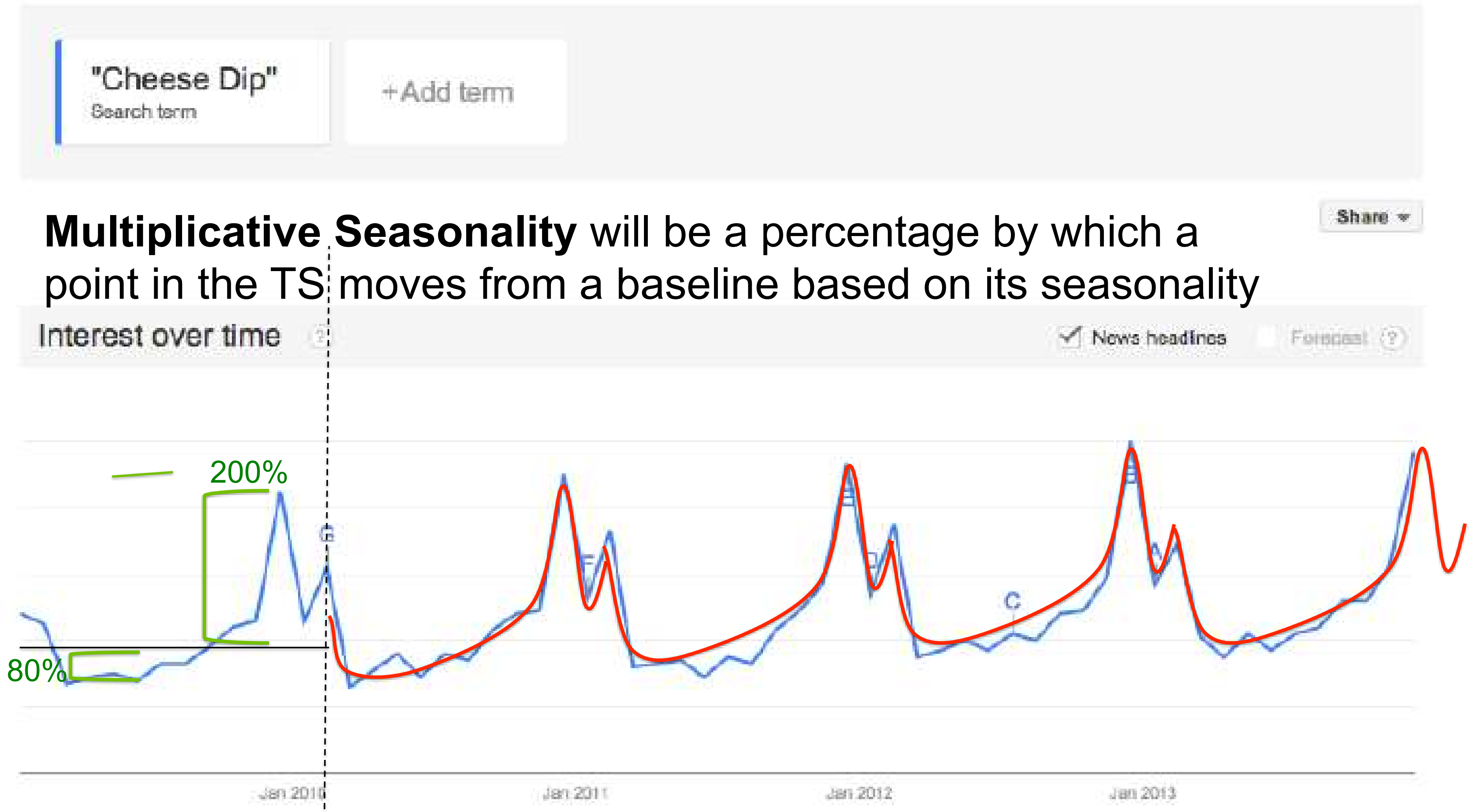


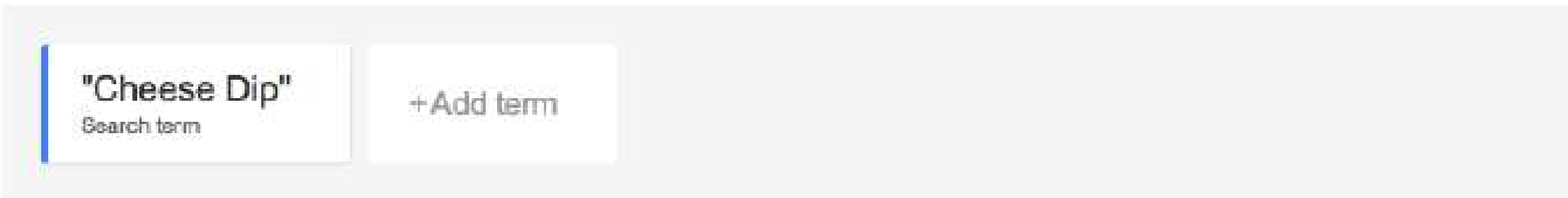
"Cheese Dip"
Search term
+Add term
Share
Interest over time
News headlines
Forecast
Jan 2010
Jan 2011
Jan 2012
Jan 2013

Multiplicative Seasonality will be a percentage by which a point in the TS moves from a baseline based on its seasonality



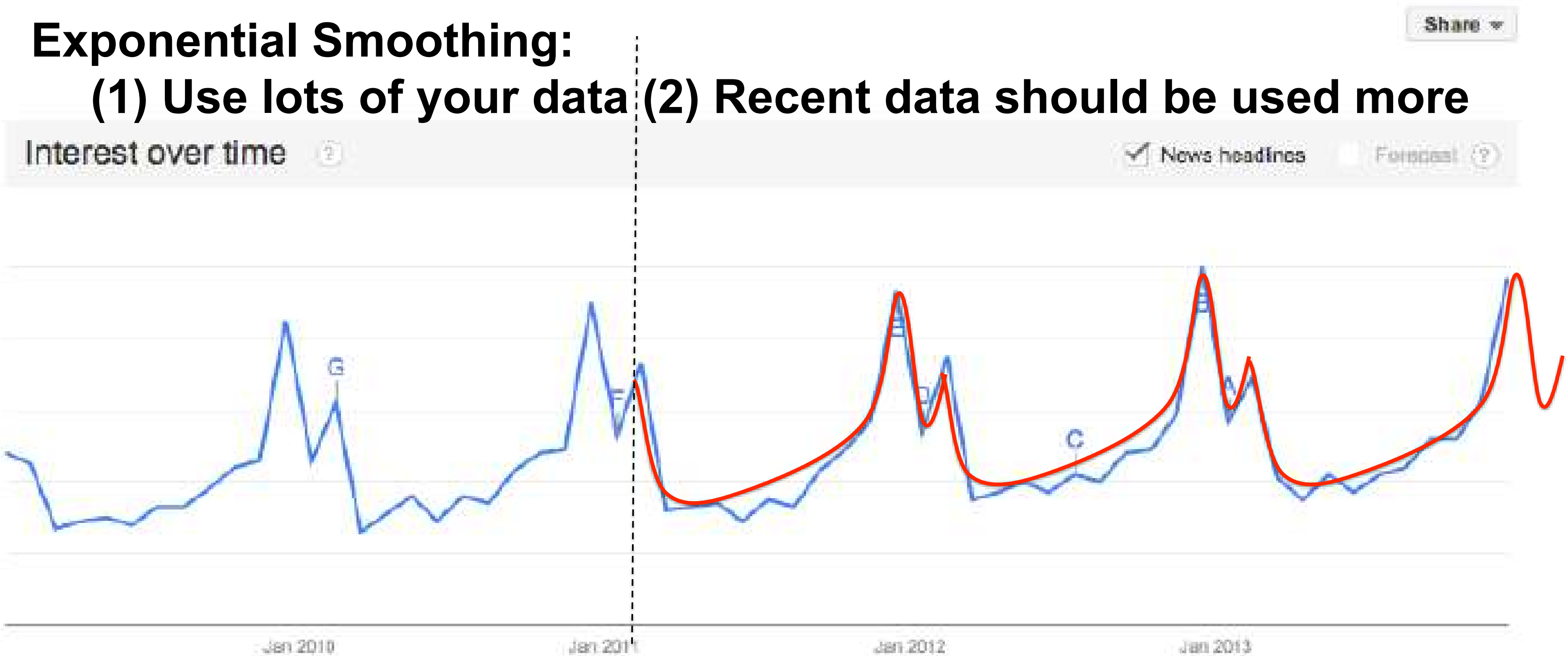
Multiplicative Seasonality will be a percentage by which a point in the TS moves from a baseline based on its seasonality





Exponential Smoothing:

(1) Use lots of your data (2) Recent data should be used more



Everybody take a break

Stretch

Drink coffee

Escape out the back

Optimization – Making good decisions

- Predictive Modeling
  - Given the inputs, what is my output?
  - You can't control anything here
- Optimization Modeling
  - Given the inputs, how do I optimize my output?
  - You can change the future!

- Examples:
  - Scheduling
  - Investment
  - Pricing

Optimization – Why care?

- Optimization can directly touch the bottom line
  - Saves time and money
  - Minimizes risk, maximizes profit
- Optimization is embedded in many data science techniques

Optimization – So how are optimization problems stated and solved?

- Objective
- Decisions
- Constraints

- And if these are *linear* then there's an awesome algorithm for solving these problems

Optimization – An example problem

- **Decisions**:
  - Guns
  - Butter
- **Objective**:
  - Maximize Revenue
    - Guns: \$195
    - Butter: \$150
- **Constraints**:
  - Cellar: 21 Cubic Meters
    - Guns: 0.5 Cubic Meters
    - Butter: 1.5 Cubic Meters
  - Budget: \$1800
    - Guns: \$150 cost
    - Butter: \$100 cost

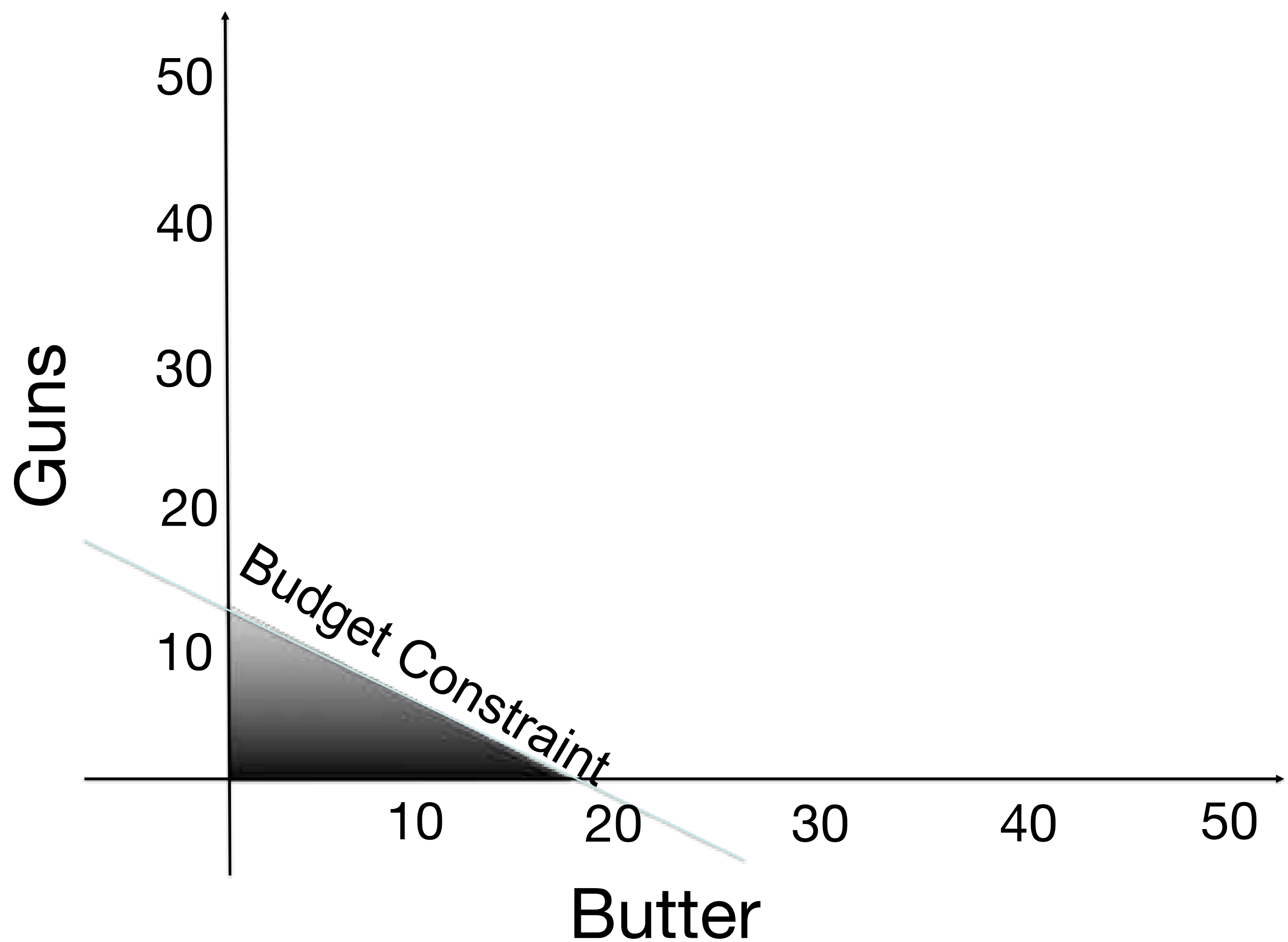
Optimization – Linear?

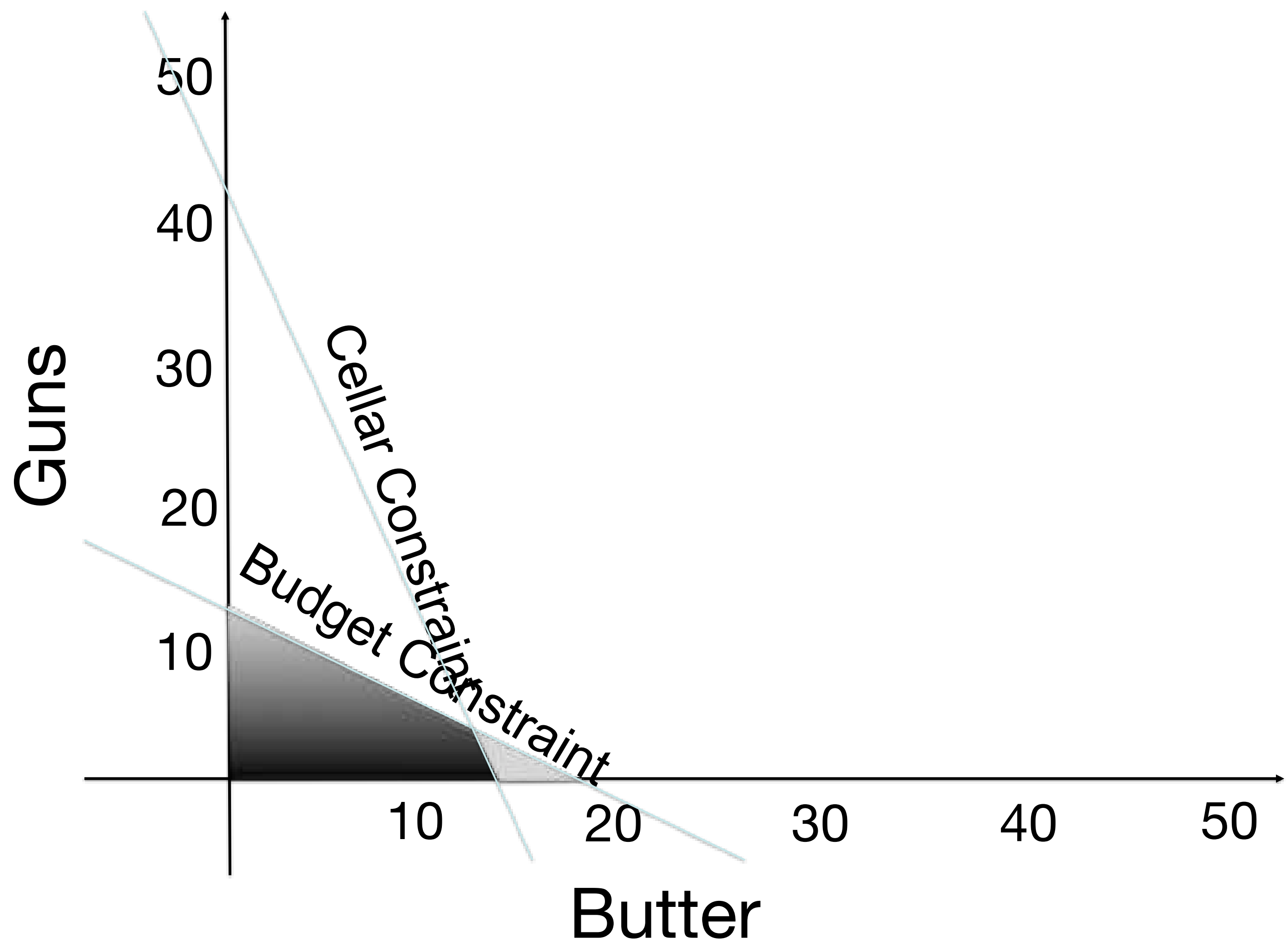
- Objective:
  - Maximize Revenue
    - \$195*Guns + \$150*Butter
- Constraints:
  - Cellar: 0.5*Guns + 1.5*Butter ≤ 21
  - Budget: \$150*Guns + \$100*Butter ≤ \$1800
  - Nonnegative: Guns, Butter ≥ 0

Optimization – Linear?

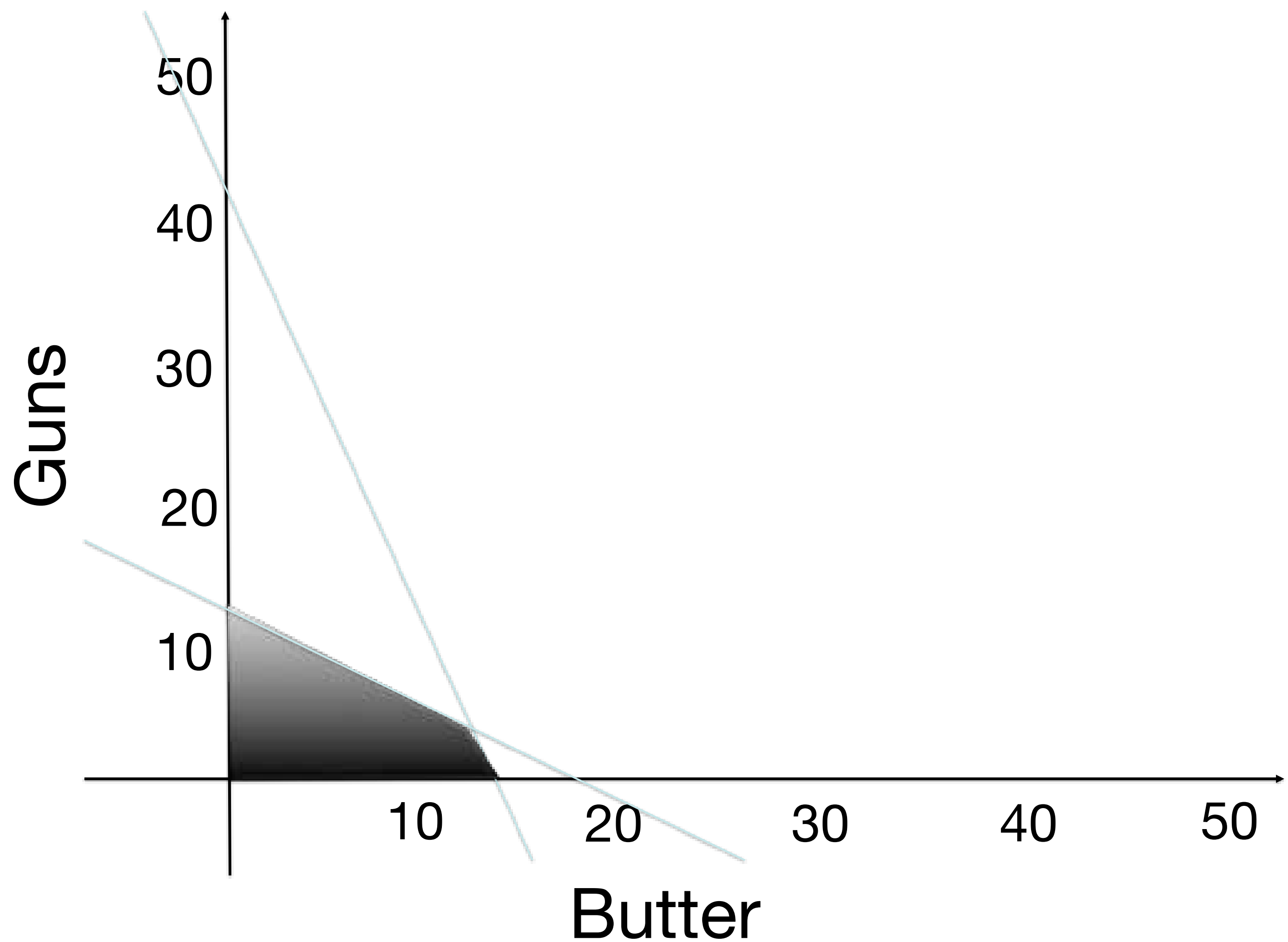
- Objective:
  - Maximize Revenue
    - $195*Guns + $150*Butter
- Constraints:
  - Cellar: $0.5 \cdot \text{Guns} + 1.5 \cdot \text{Butter} \leq 21$
  - Budget: $\$150 \cdot \text{Guns} + \$100 \cdot \text{Butter} \leq \$1800$
  - Nonnegative: Guns, Butter ≥ 0

- Things that are linear in Excel:
  - +/- decisions. */÷ decision by constants
  - SUM() of decision
  - SUMPRODUCT() where decisions are in one range only
  - AVERAGE() or decisions

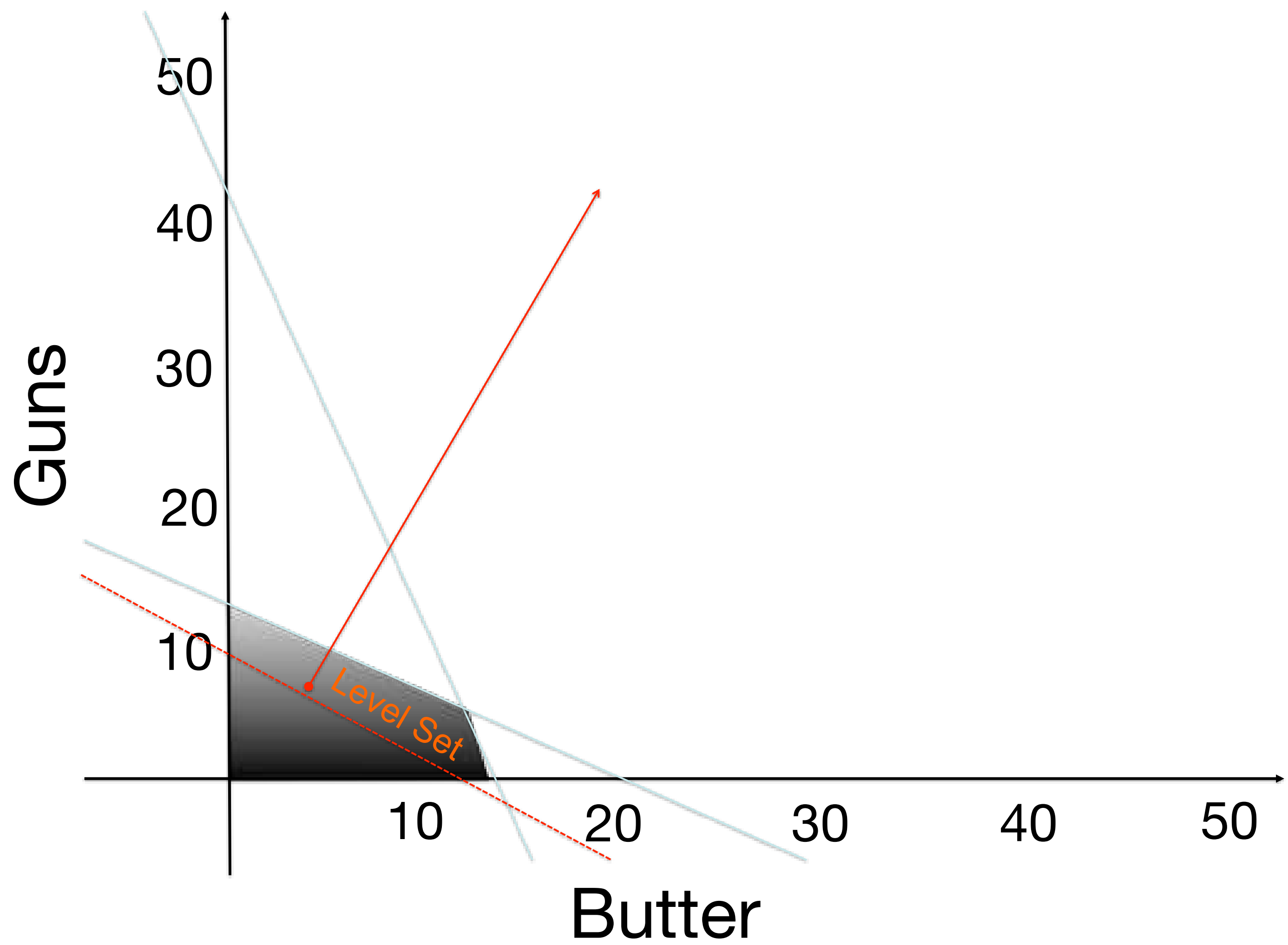

Guns
50
40
30
20
10
Budget Constraint
10
20
30
40
50
Butter



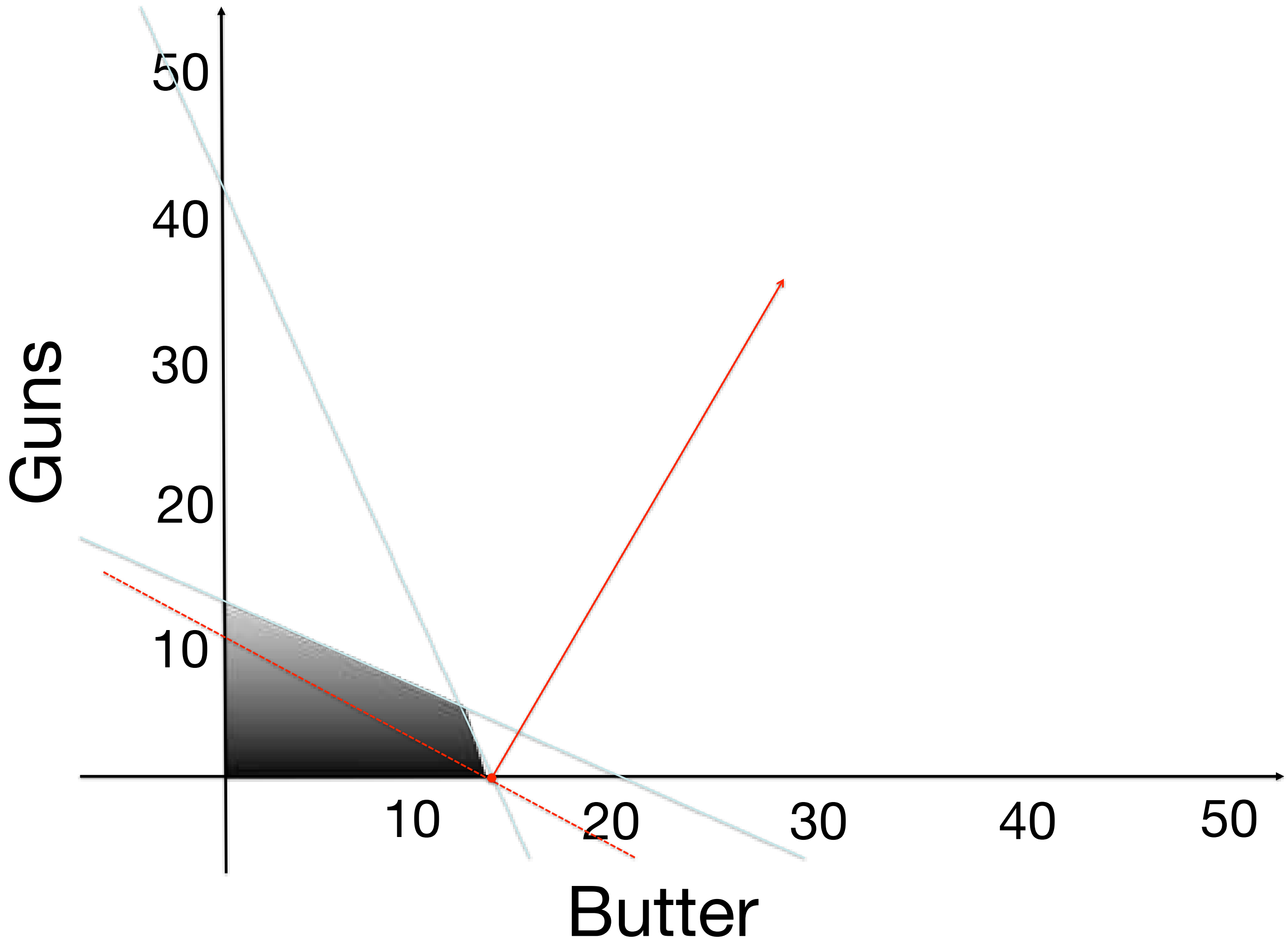
Guns
50
40
30
20
10
Cellar Constraint
Budget Constraint
10
20
30
40
50
Butter



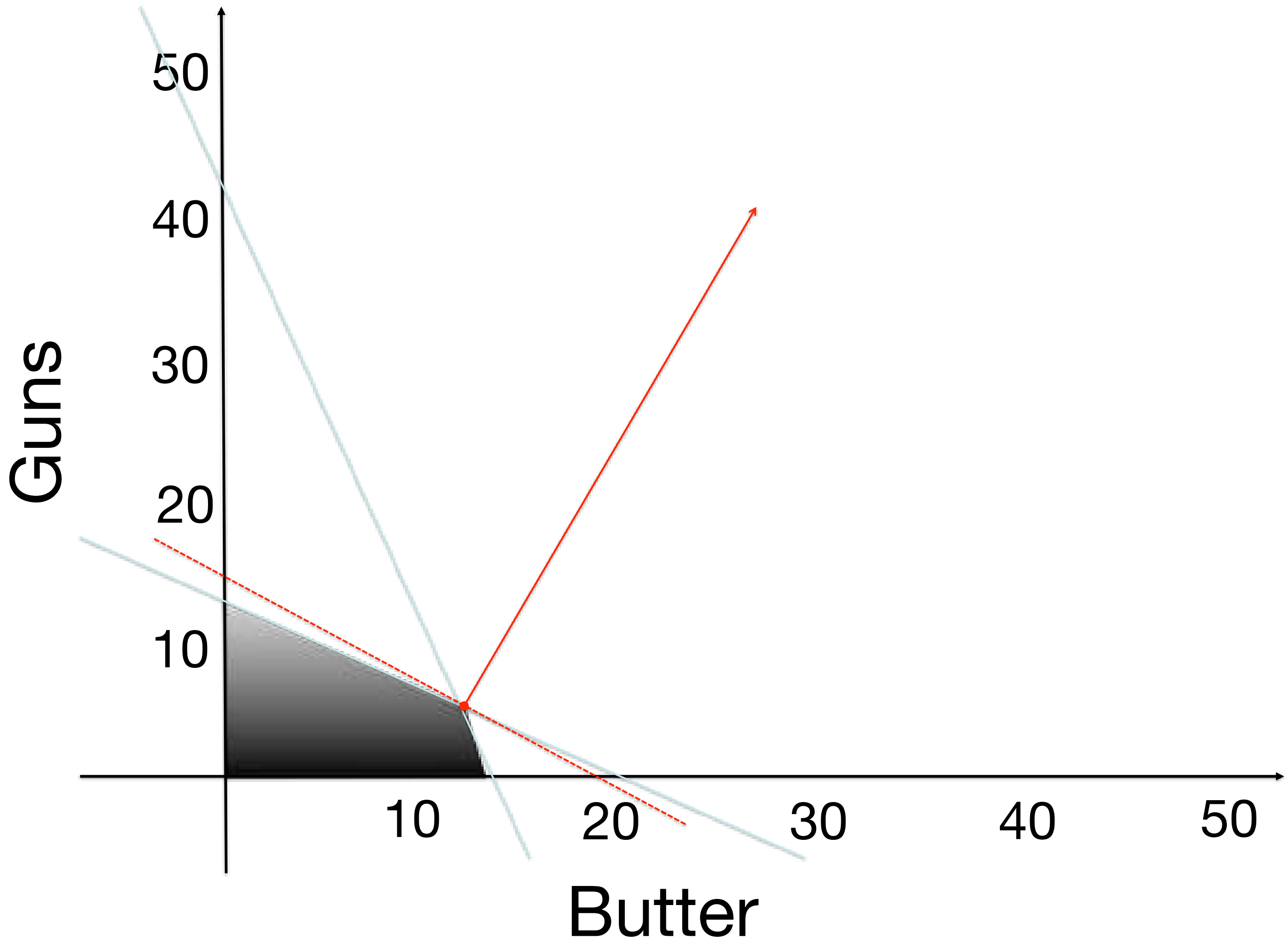
50
40
30
Guns
20
10
10
20
30
40
50
Butter



Guns
50
40
30
20
10
Level Set
10
20
30
40
50
Butter

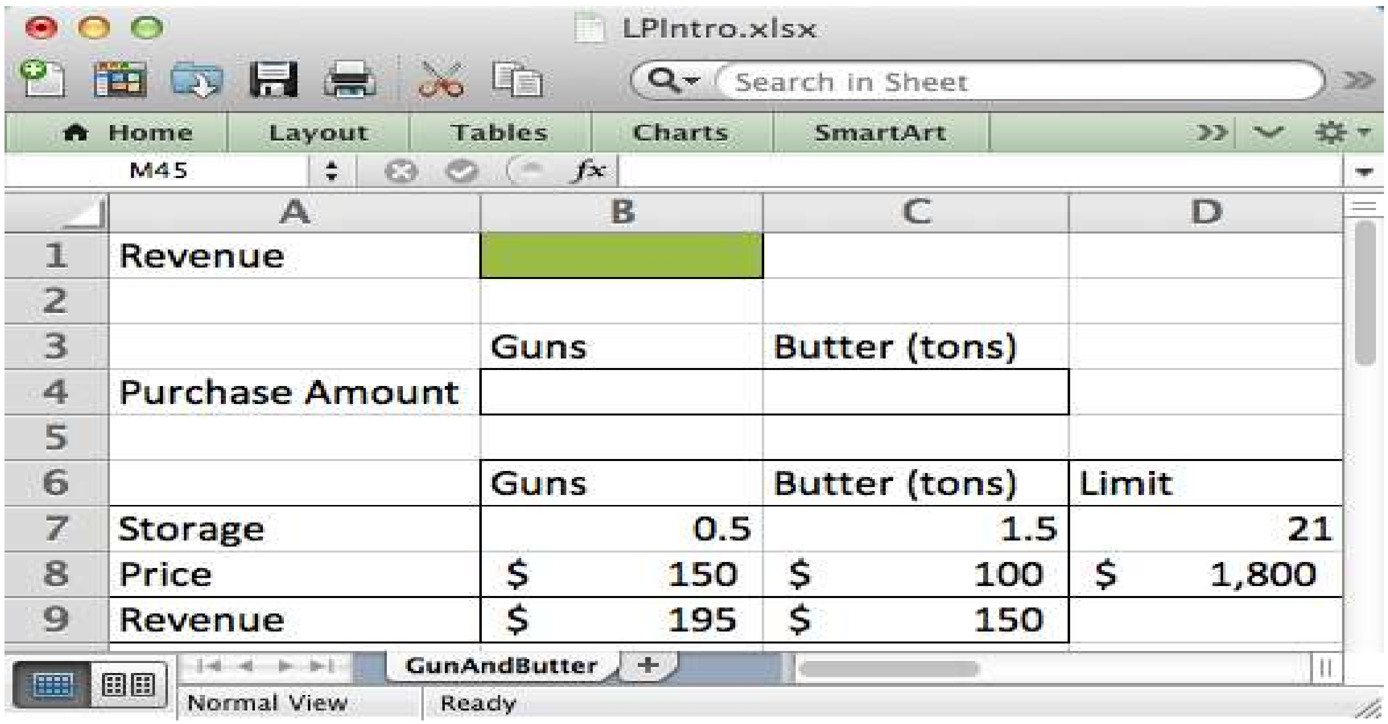


50
40
30
Guns
20
10
10
20
30
40
50
Butter



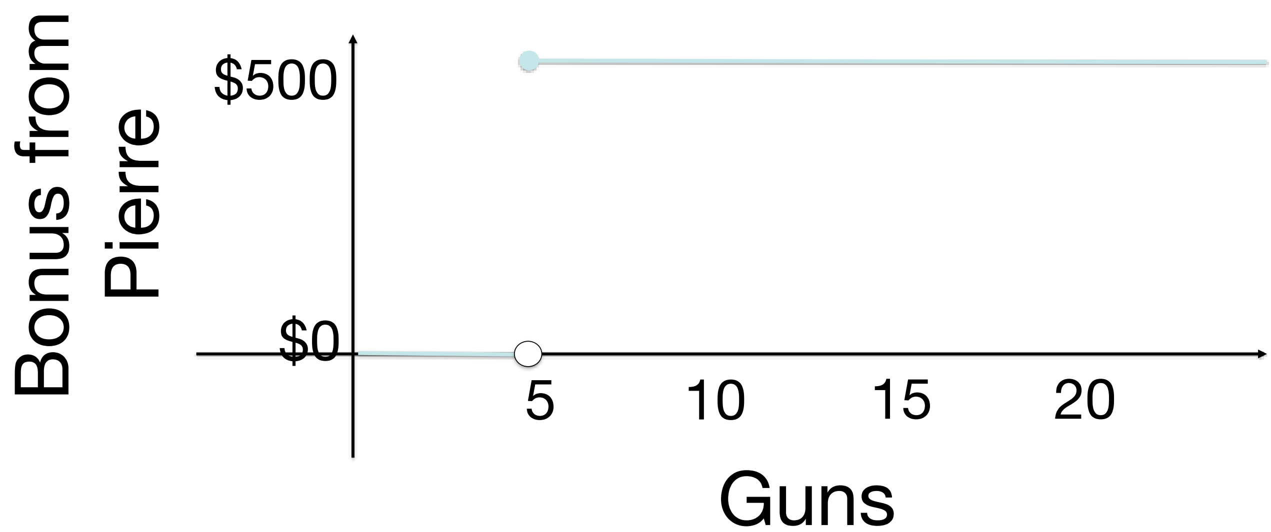
50
40
30
Guns
20
10
10
20
30
40
50
Butter

Setting up a spreadsheet



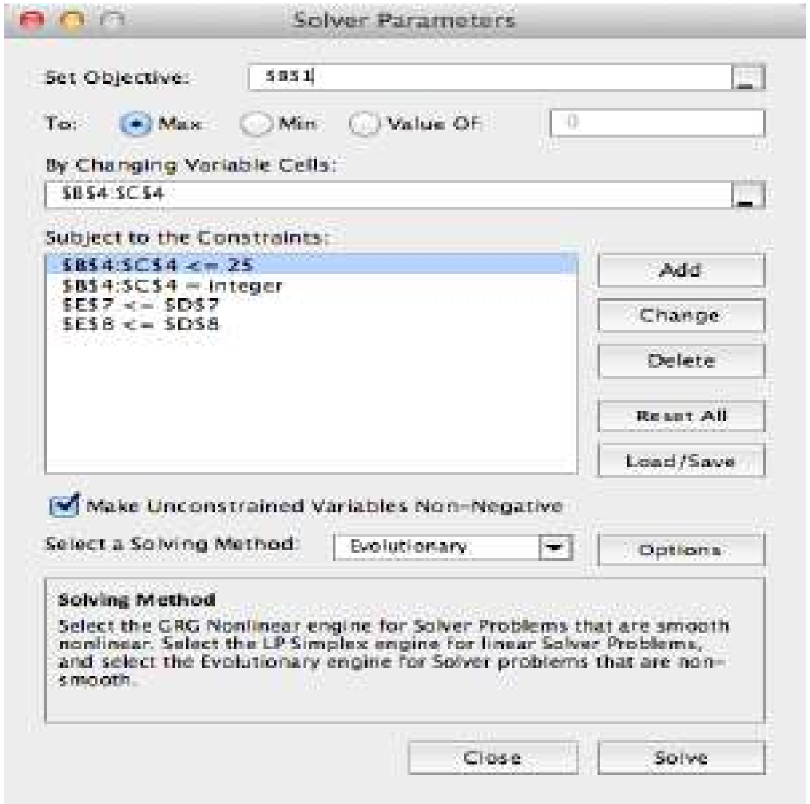
	A	B	C	D
1	Revenue			
2				
3		Guns	Butter (tons)	
4	Purchase Amount			
5				
6		Guns	Butter (tons)	Limit
7	Storage	0.5	1.5	21
8	Price	$ 150	$ 100	$ 1,800
9	Revenue	$ 195	$ 150	

Pierre gives a $500 bonus



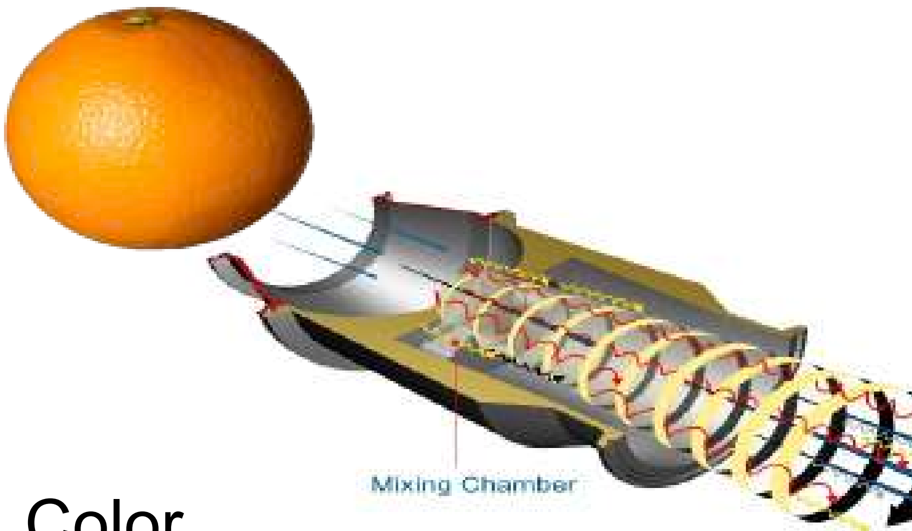
- Not linear in Excel:
- IF(), AND(), OR()
- MAX(), MIN(), LARGE(), PERCENTILE()
- LOOKUP(), INDEX(), MATCH()

Evolutionary Algorithms



- Generate a pool of initial solutions
- Solutions breed through *crossover*
- Solutions *mutate* to create new solutions
- Some amount of *local search* takes place
- *Selection* occurs

Moving on to a larger problem – Orange Juice



- Color
- Brix/Acid
- Acid
- Astringency

Input Data

Varietal	Region	Qty Available (1,000 Gallons)	Brix / Acid Ratio	Acid (%)	Astringency (1-10 Scale)	Color (1-10 Scale)	Price (per 1K Gallons)	Shipping
Hamlin	Brazil	672	10.5	0.60%	3	3	$ 500.00	$ 100.00
Mosambi	India	400	6.5	1.40%	7	1	$ 310.00	$ 150.00
Valencia	Florida	1200	12	0.95%	3	3	$ 750.00	$ -
Hamlin	California	168	11	1.00%	3	5	$ 600.00	$ 60.00
Gardner	Arizona	84	12	0.70%	1	5	$ 600.00	$ 75.00
Sunstar	Texas	210	10	0.70%	1	5	$ 625.00	$ 50.00
Jincheng	China	588	9	1.35%	7	3	$ 440.00	$ 120.00
Berna	Spain	168	15	1.10%	4	8	$ 600.00	$ 110.00
Verna	Mexico	300	8	1.30%	8	3	$ 300.00	$ 90.00
Biondo Commune	Egypt	210	13	1.30%	3	5	$ 460.00	$ 130.00
Belladonna	Italy	180	14	0.50%	3	9	$ 505.00	$ 115.00

Input Data

- Objective:
  - Minimize Cost of Juice Procurement for 3 Months
- Decisions:
  - How much to order of what in which month
- Constraints:
  - Availability
  - Demand: [600, 600, 700]
  - Valencia: 40% for a tax break
  - BAR: 11.5 – 12.5
  - Acid: .0075 - .0100
  - Astringency: 0 – 4
  - Color: 4.5 – 5.5

Big M constraints

- You can only buy from 4 suppliers
- So you need to flick a switch each time you purchase *any amount* and then total up how many switches got flicked