SHARP

LABORATORIES OF AMERICA

Predictive Analytics War Stories

Feb 13, 2015

Hobson Lane

slides.com/hobsonlane/data-analytics-war-stories/live bit.ly/pawsvote

Choose Your Story

7707-2-TOTAL

(770) 728-6825

1. Only Nyquist Knows
2. The Meaning of Mean
3. Data Dearth
4. Question the Question
5. Deep Net Runs Aground
6. Escape the Maze

When your vehicle is out of control...

1. Only Nyquist Knows

Photo by
Eric Cutright
Public Domain

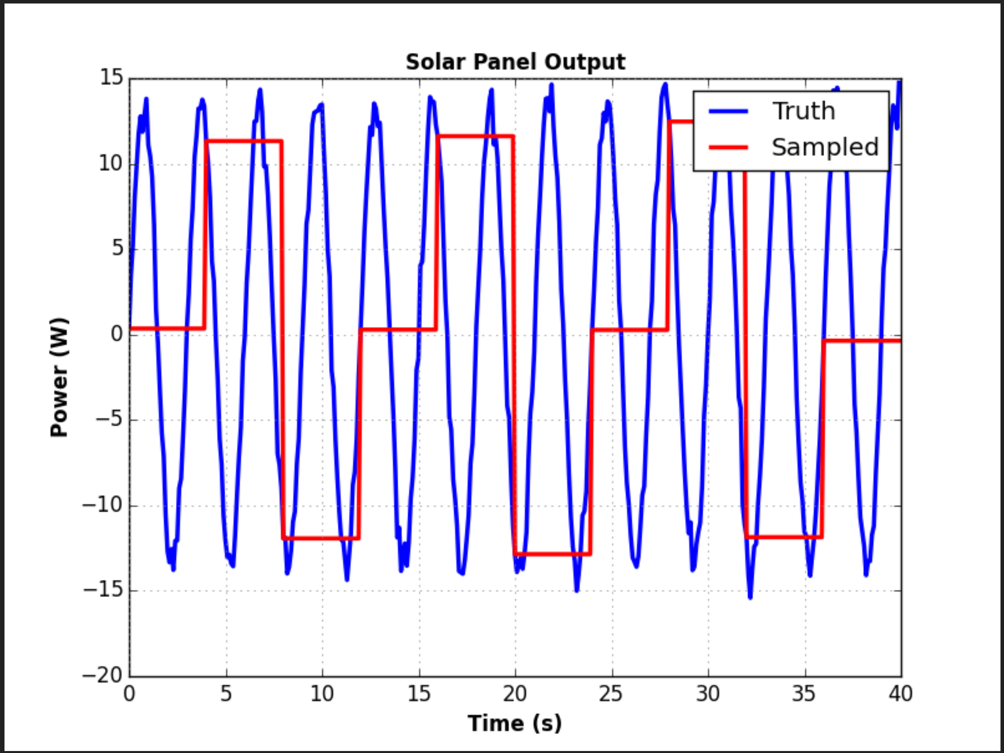
Photo by
Public Domain

Photo by
US Secret
Service

SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: **"1"**, "2", "3", "4", "5", or "6"

1. Only Nyquist Knows

- Nav sensors (gyro., accel) are "pegged"
- All you know is solar power:

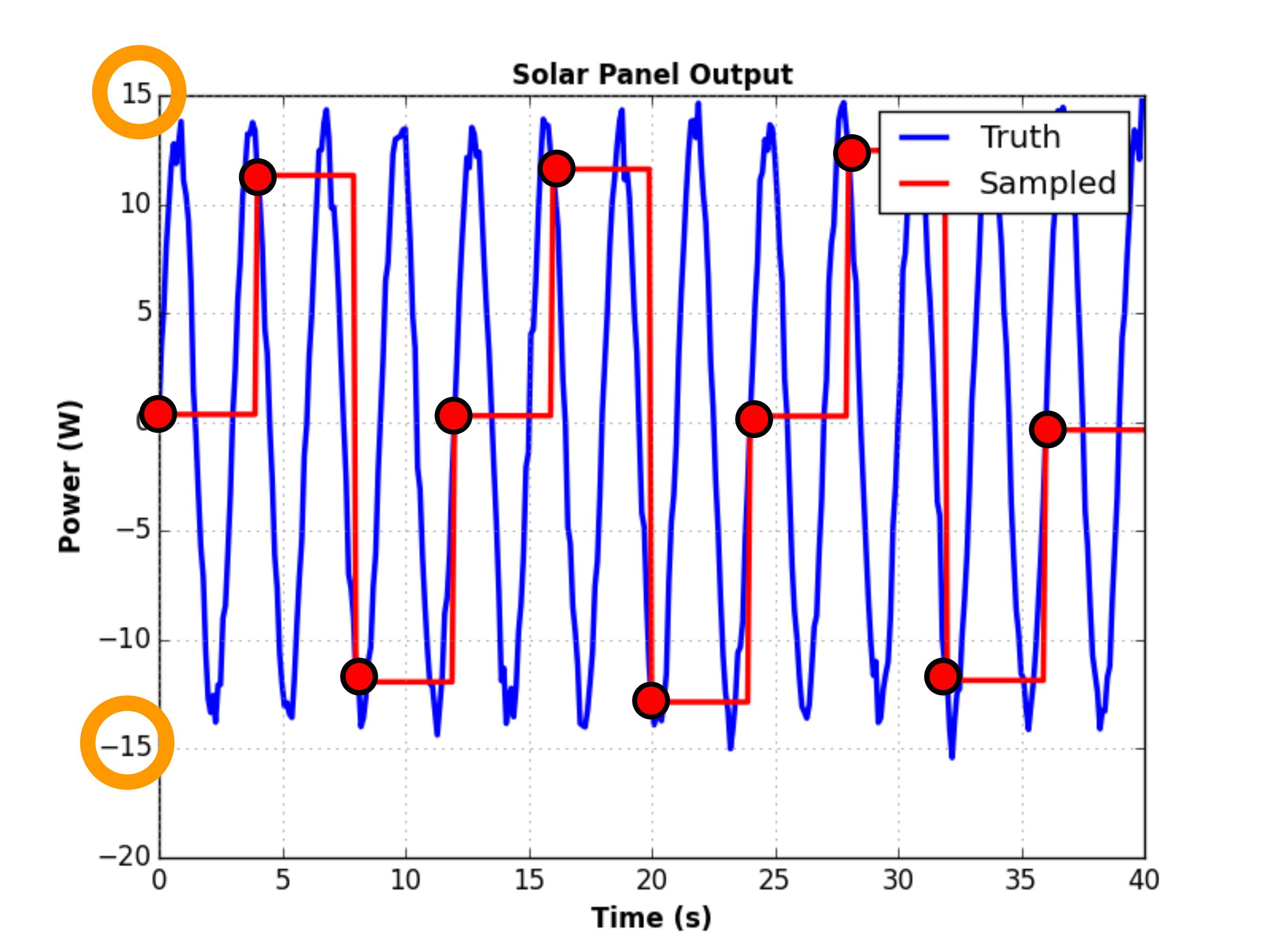


How fast is the tumble?

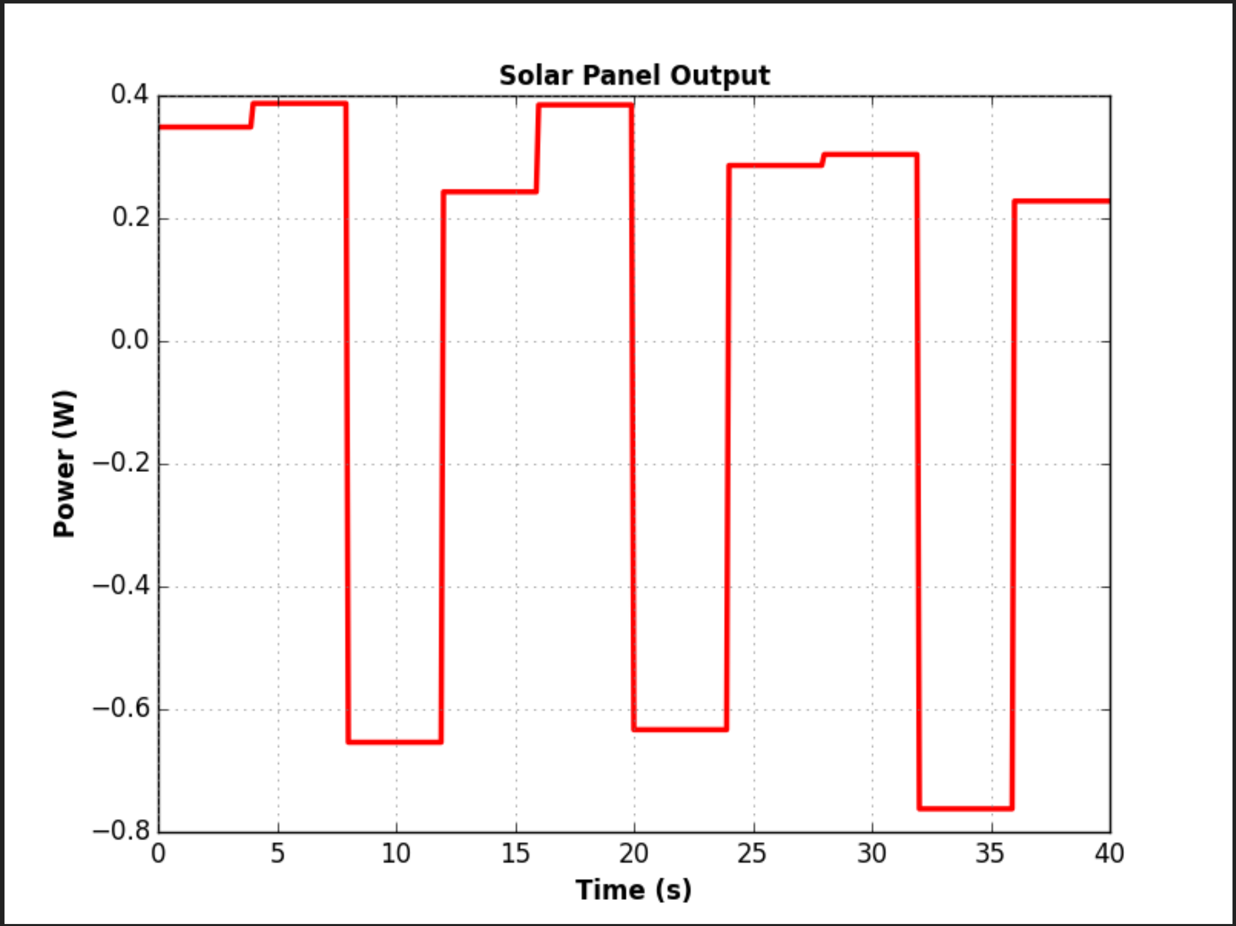
4 sec !

SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: **"1"**, "2", "3", "4", "5", or "6"

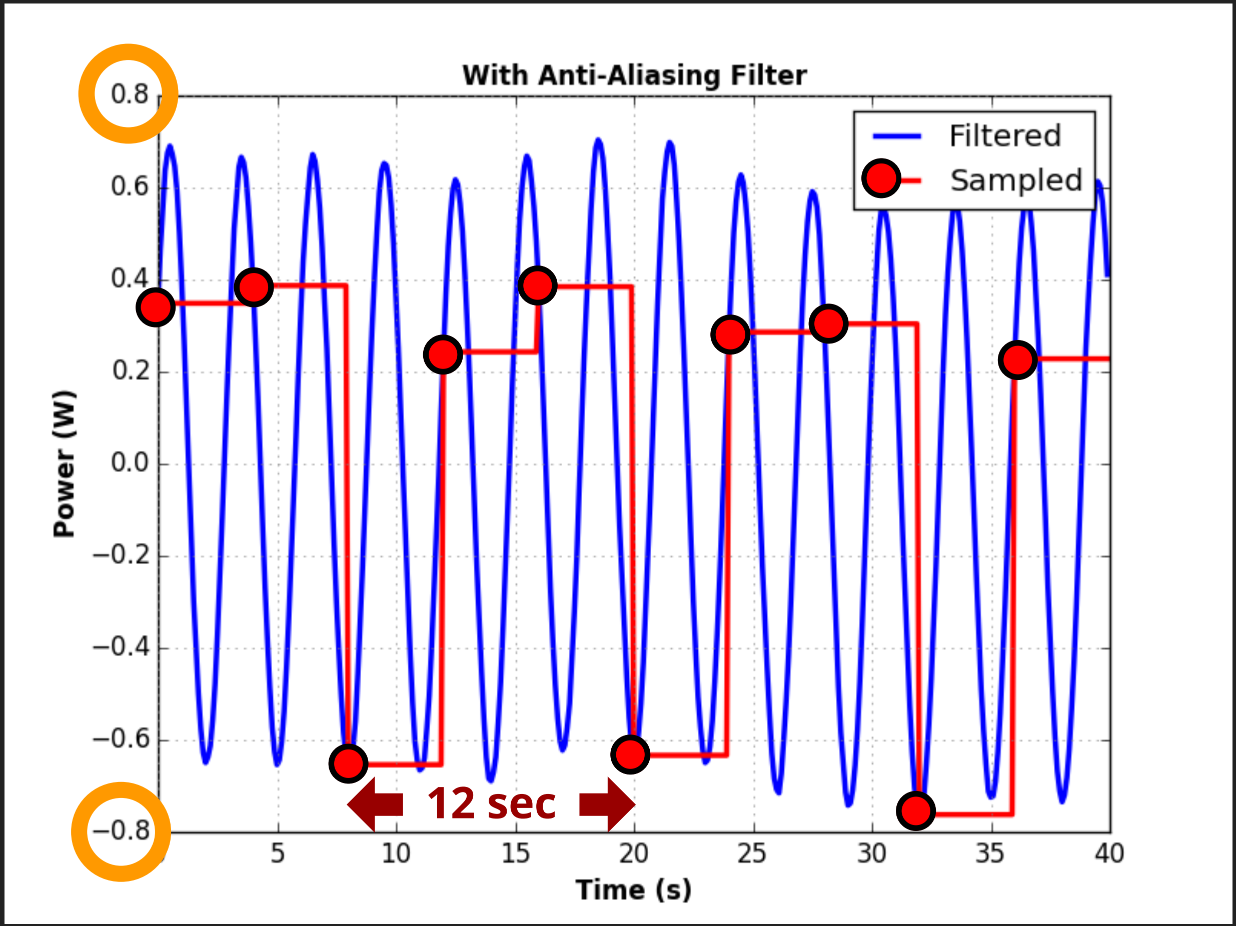
1. Only Nyquist Knows



Try an Anti-Aliasing Filter



Fail: Only Nyquist Knows



Workarounds

If Nyquist sampling (2x faster than truth) isn't possible....

- Use a different sensor
  - Postprocess existing signal (radio doppler)
- Sample **irregularly**!
  - Captures higher frequencies
  - Lomb-Scargle to post-process

```
spectrum = scipy.signal.lombscargle(sample_times, samples, frequencies)
```

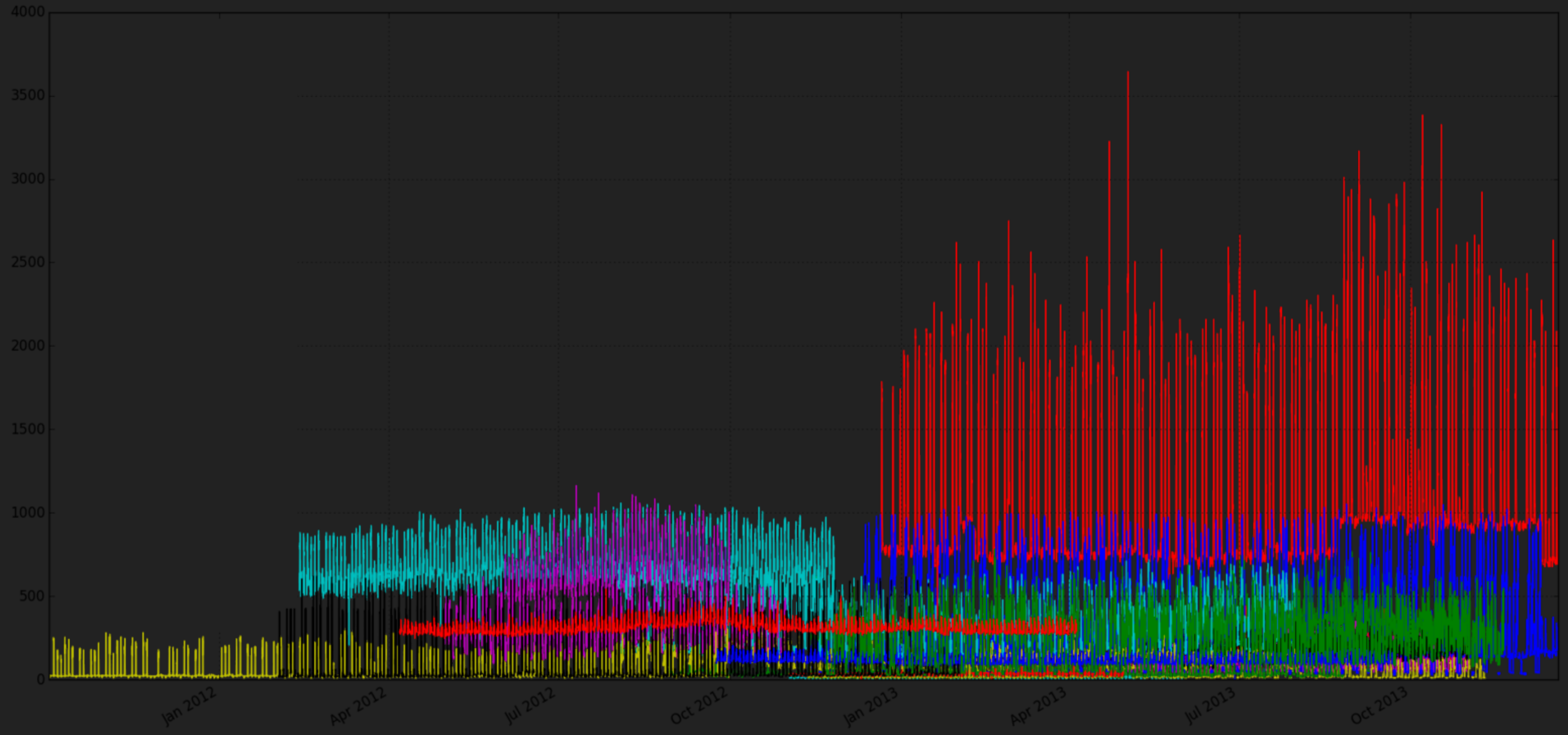
- Probabilistic modeling
  - Great for overwhelming data volume (**IoT**)

2. The Meaning of Mean

- Means don't tell the whole story
- Consider both μ and σ
- Meaning may be found in the means for each...
  - **group**, cluster, or class
- For us we started with grouping by **time of day**, but that wasn't enough...

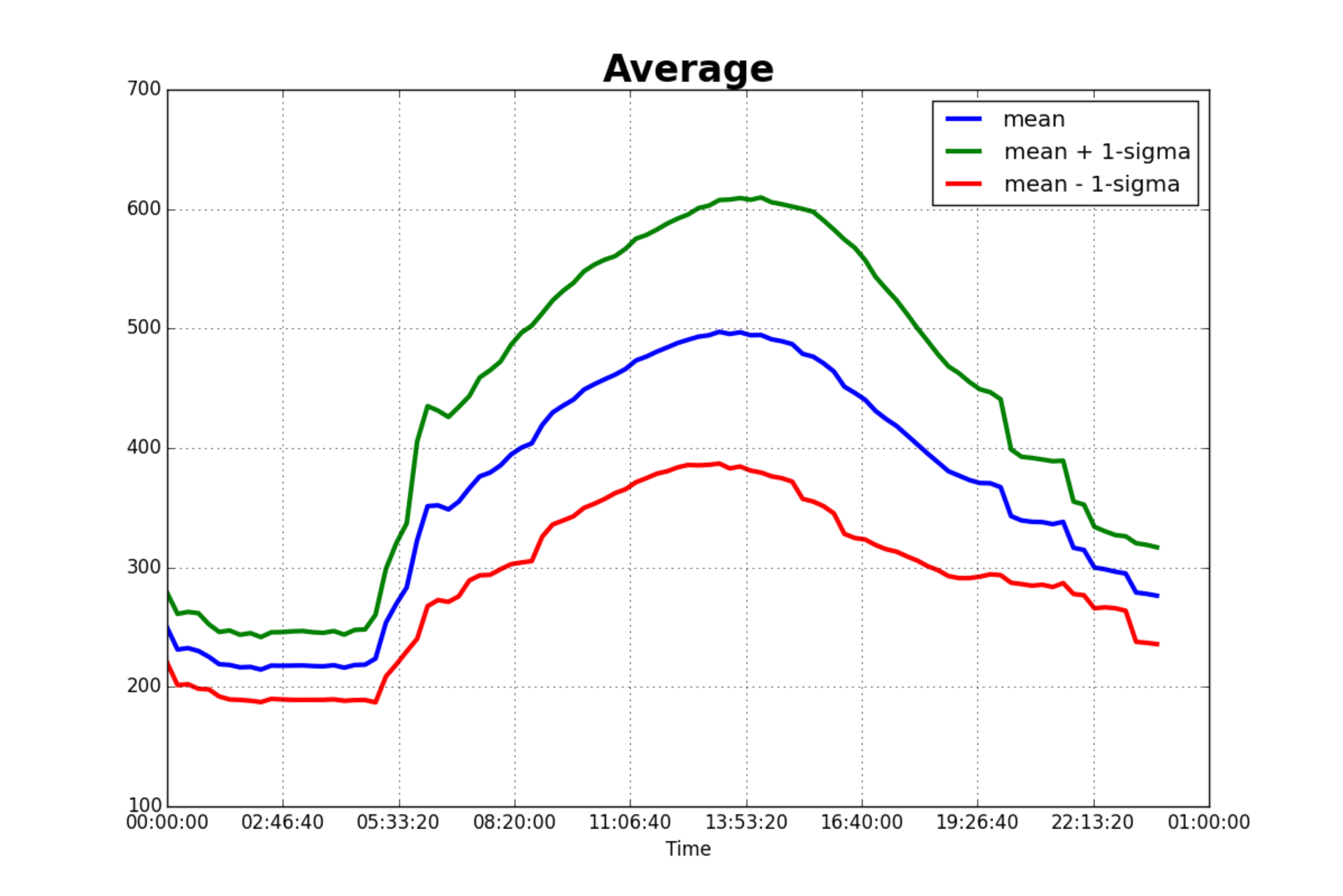
2. The Meaning of Mean

- Regression and classification required
- Many "fundamental frequencies"

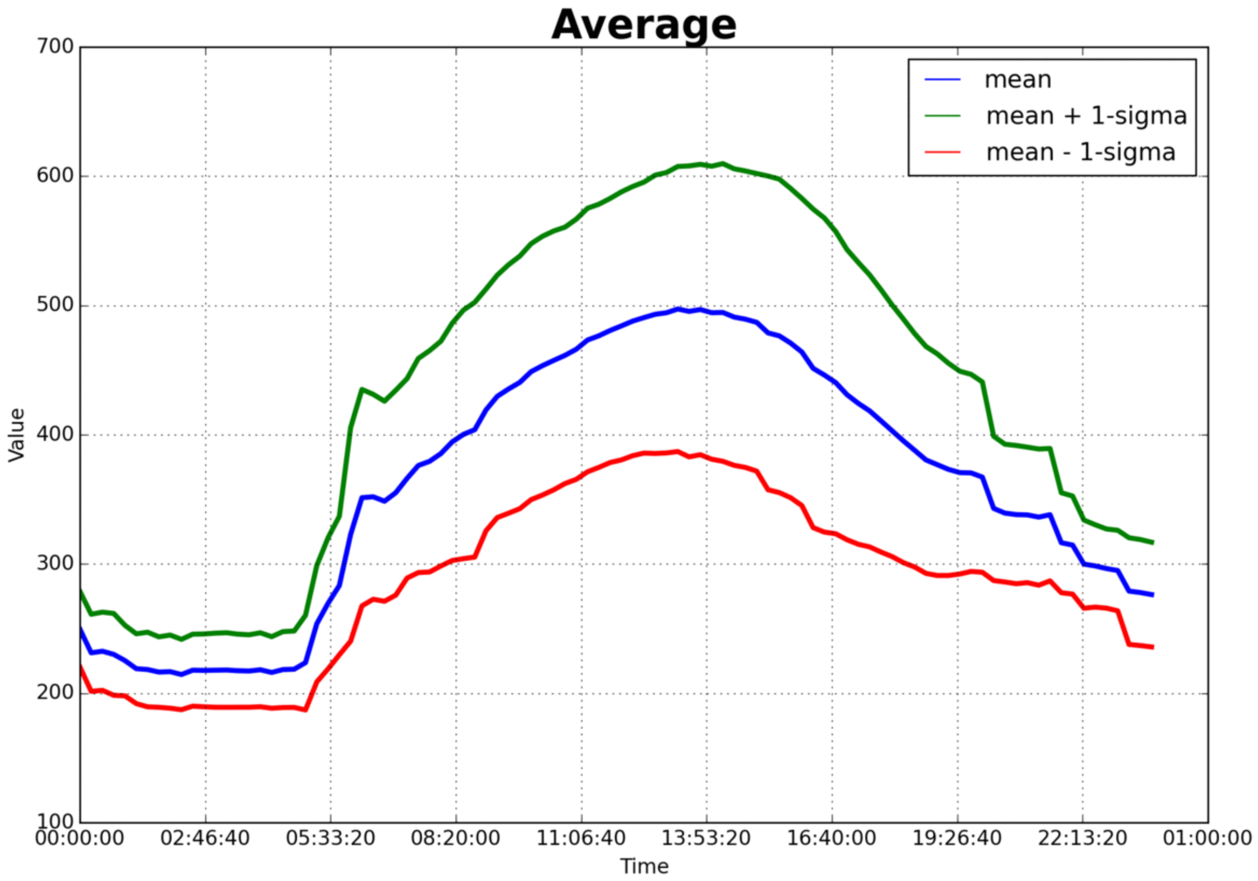


SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", **"2",** "3", "4", "5", or "6"

Mean for Each Time of Day



Classify Before Getting Mean

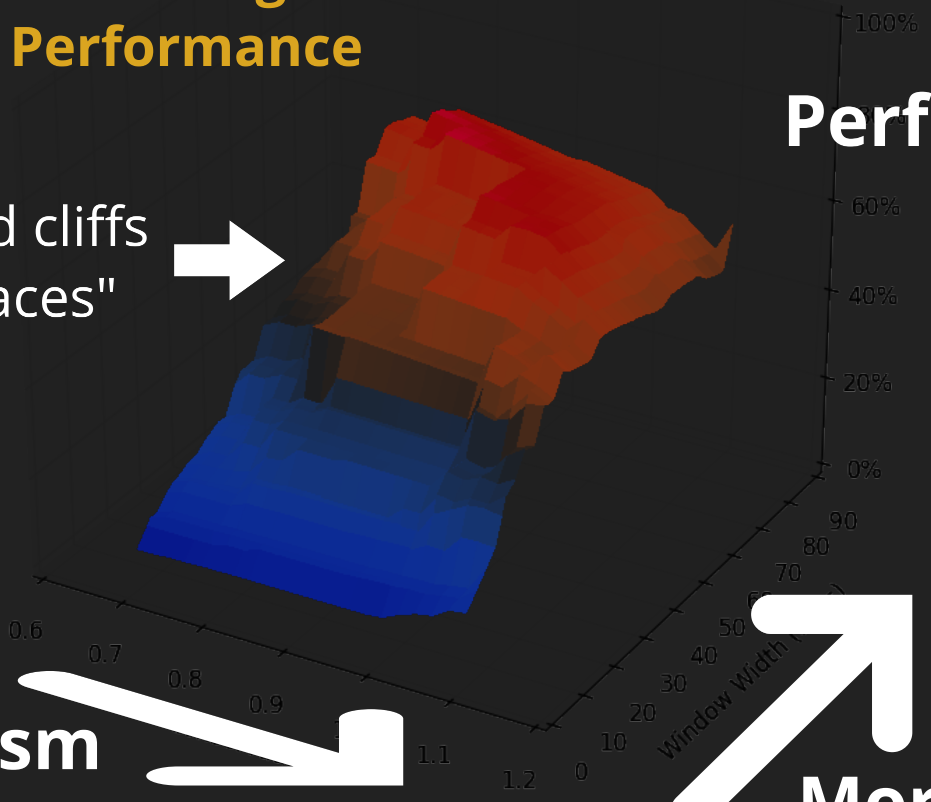


3. Data Dearth

- Tuning a 2-DOF predictive filter for performance
- More data gives algorithm more to work with
  - Less **Overfitting**
  - More **Performance**

Anticlined cliffs
or "terraces"

Performance
($)

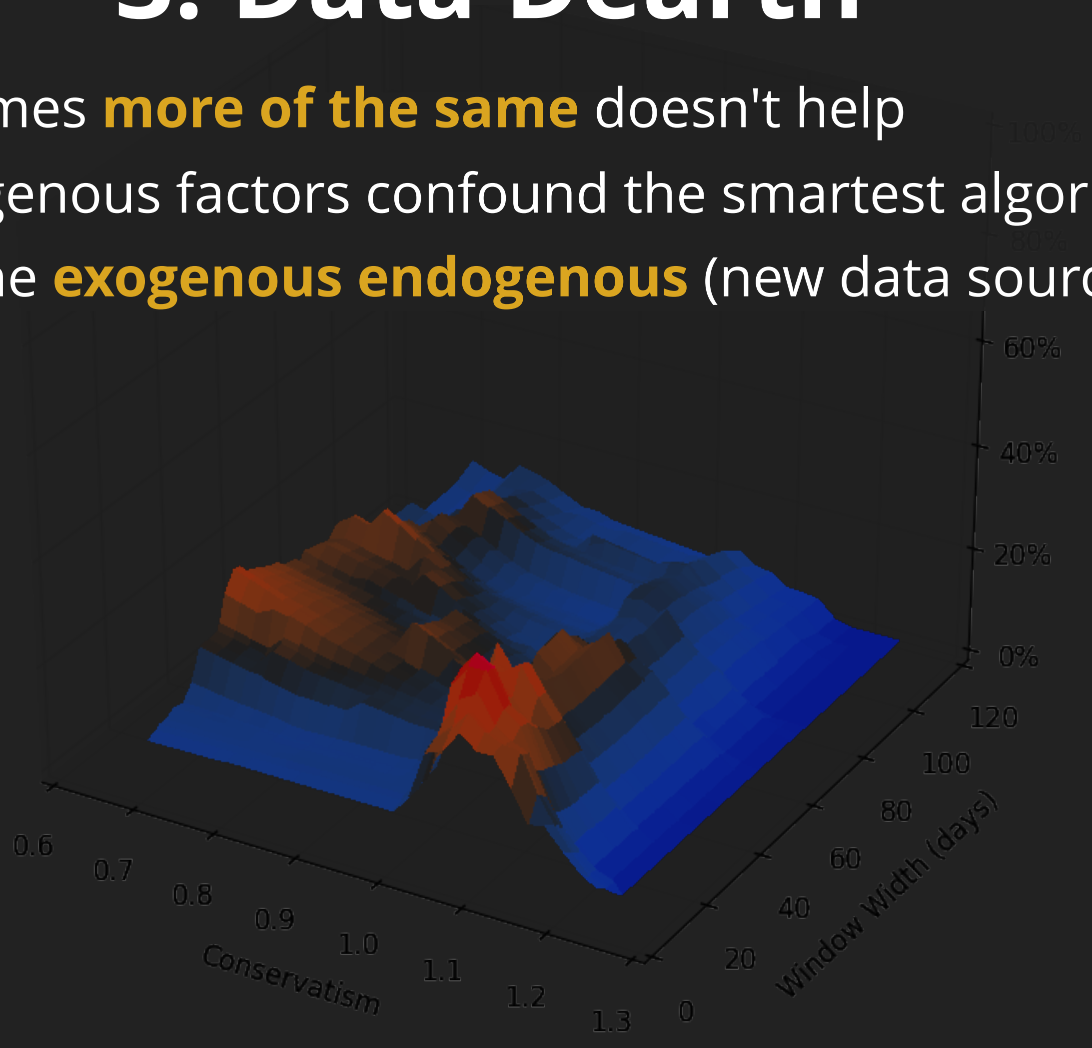


Conservatism

More Data

3. Data Dearth

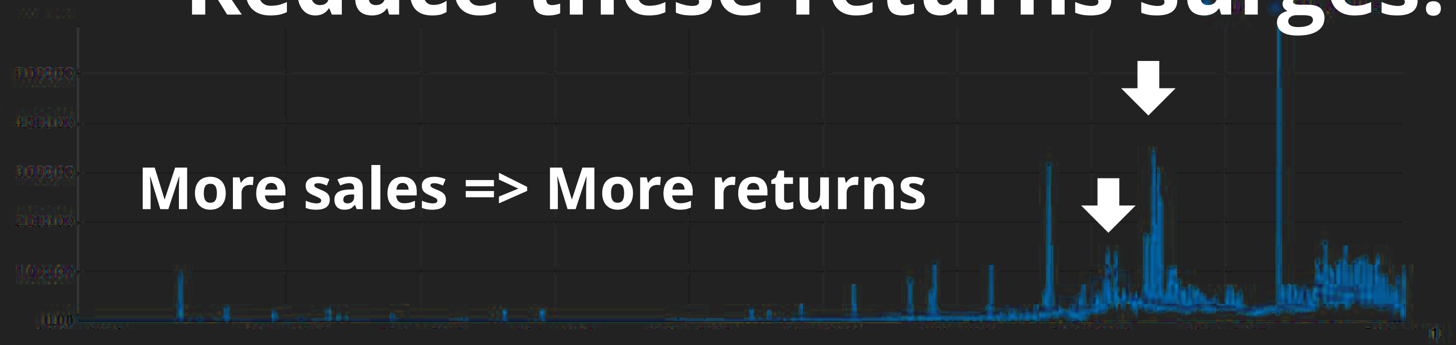
- Sometimes **more of the same** doesn't help
  - Exogenous factors confound the smartest algorithm
- Make the **exogenous endogenous** (new data source)



SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", **"3"**, "4", "5", or "6"

4. Question the Question

Reduce these returns surges!



Correlation != Causation

(a. la. Tyler Vigen)

Multiple interracting causes

Normalize return rate for sales (lag-compensated)

4. Question the 6σ Question

"Cost of quality"

"Customer reject rate"

"Defect rate"

$$\text{Reject rate} = \frac{\text{Rejects (last quarter)}}{\text{Sales (last quarter)}}$$

Simple equation everyone can agree on

But it's Wrong!

And it's Late!

4. Better "Question"

$$\text{Reject rate} = \frac{\text{Rejects (last quarter)}}{\text{Sales (qtr before last)}}$$

Even Better

$$\text{Reject rate} = \frac{\text{Rejects (last quarter)}}{\text{Sales (estimate lagged quarter)}}$$

Correct

Reject rate = $\dfrac{\text{Rejects (last week)}}{\text{Sales (integral of lagged sales)}}$

$$r_r = \Sigma_k \alpha s_{n-k}$$

"Birth-Death Process"

Sale | Lag | Reject

$$S(t) \Rightarrow \boxed{H(t, \tau)} \Rightarrow R(t)$$

Sale: Product enters "pipeline" arbitrarily

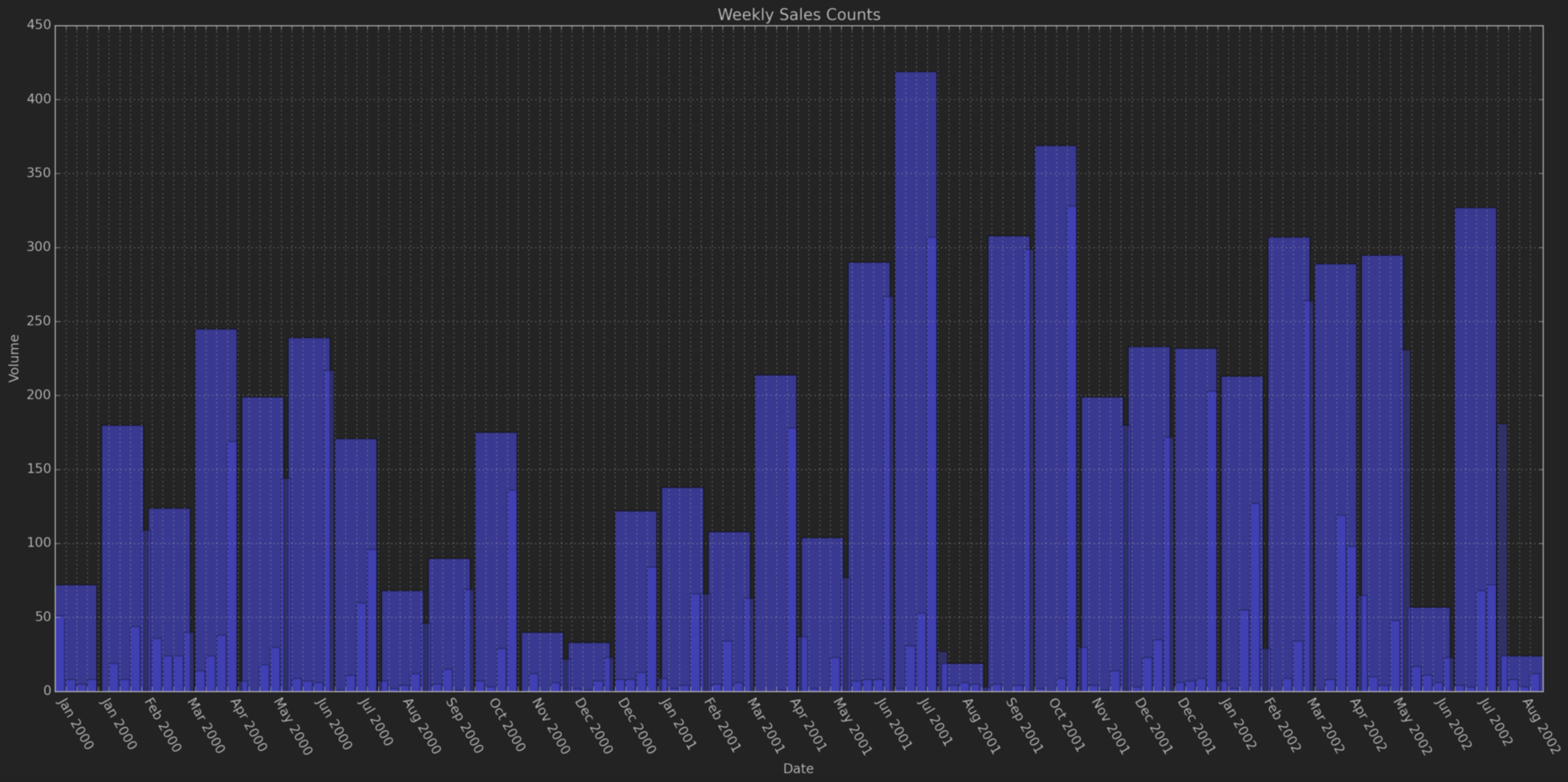
Lag: Flow rate (Reject rate)

Reject: All products "die", Question is **when** And the portion that happens too soon

$$r_r = \Sigma_k \alpha s_{n-k}$$

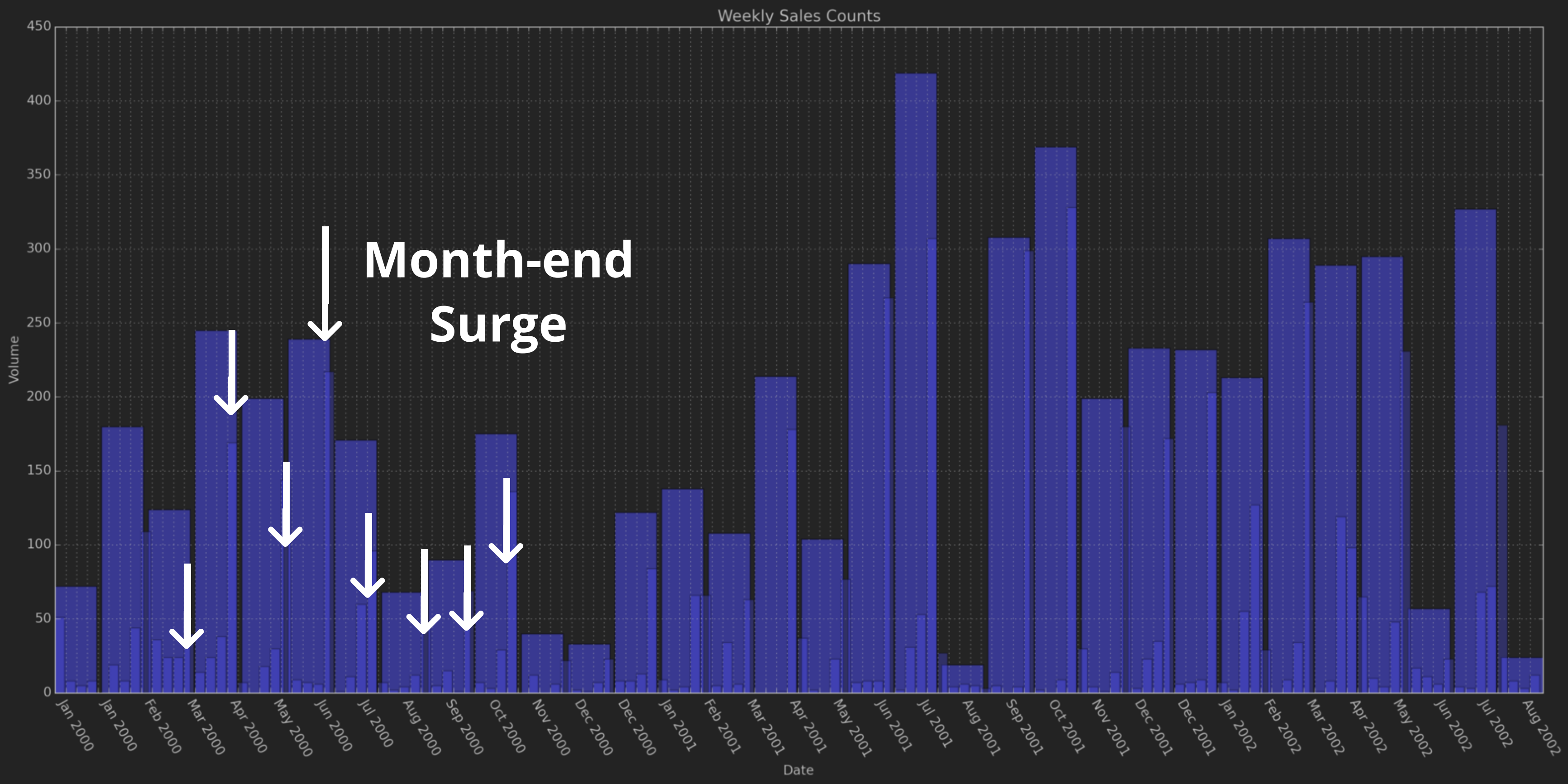
4. Question the Question

Histogram reveals trend and seasonality



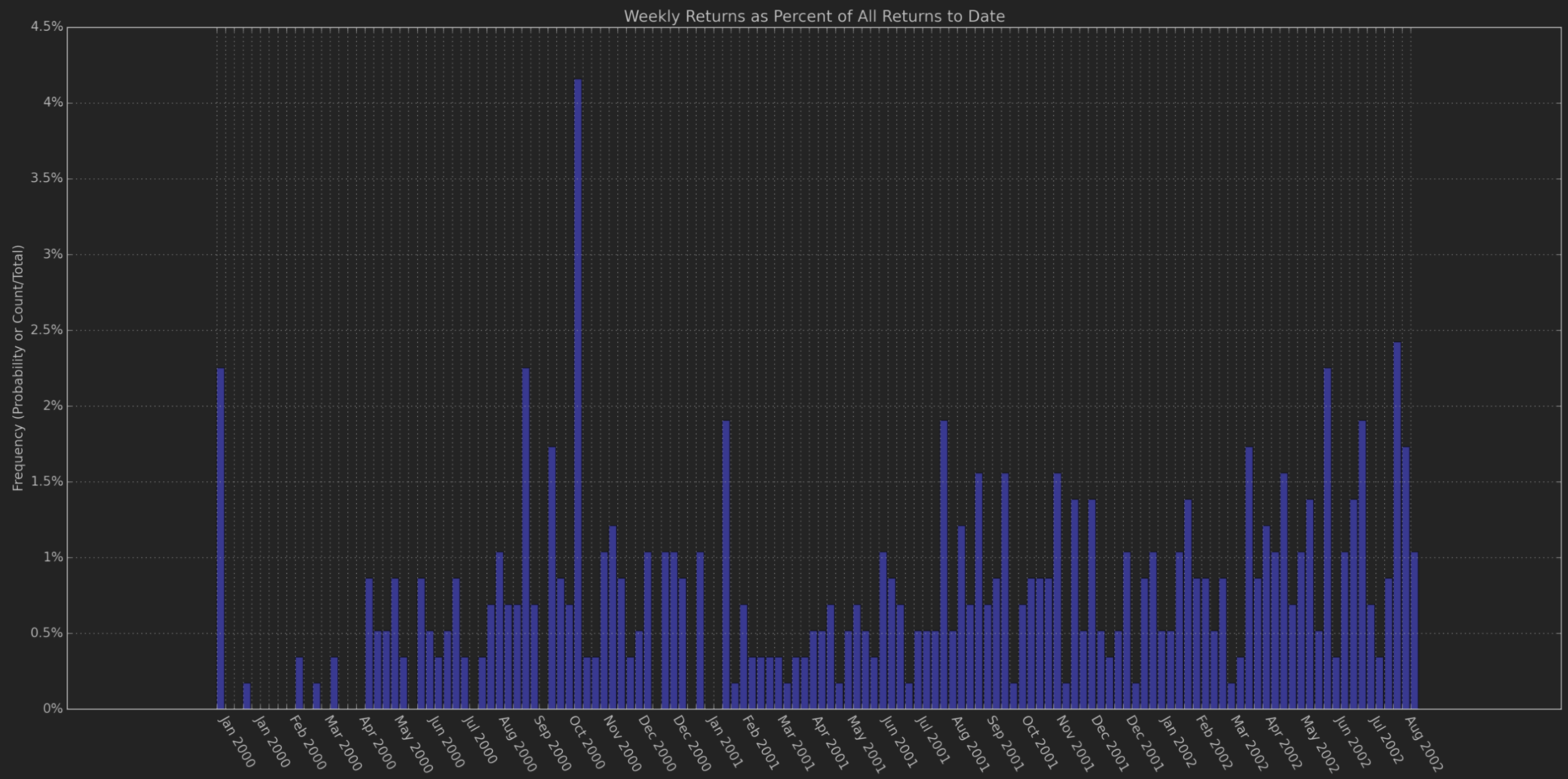
SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", "3", **"4"**, "5", or "6"

Sales

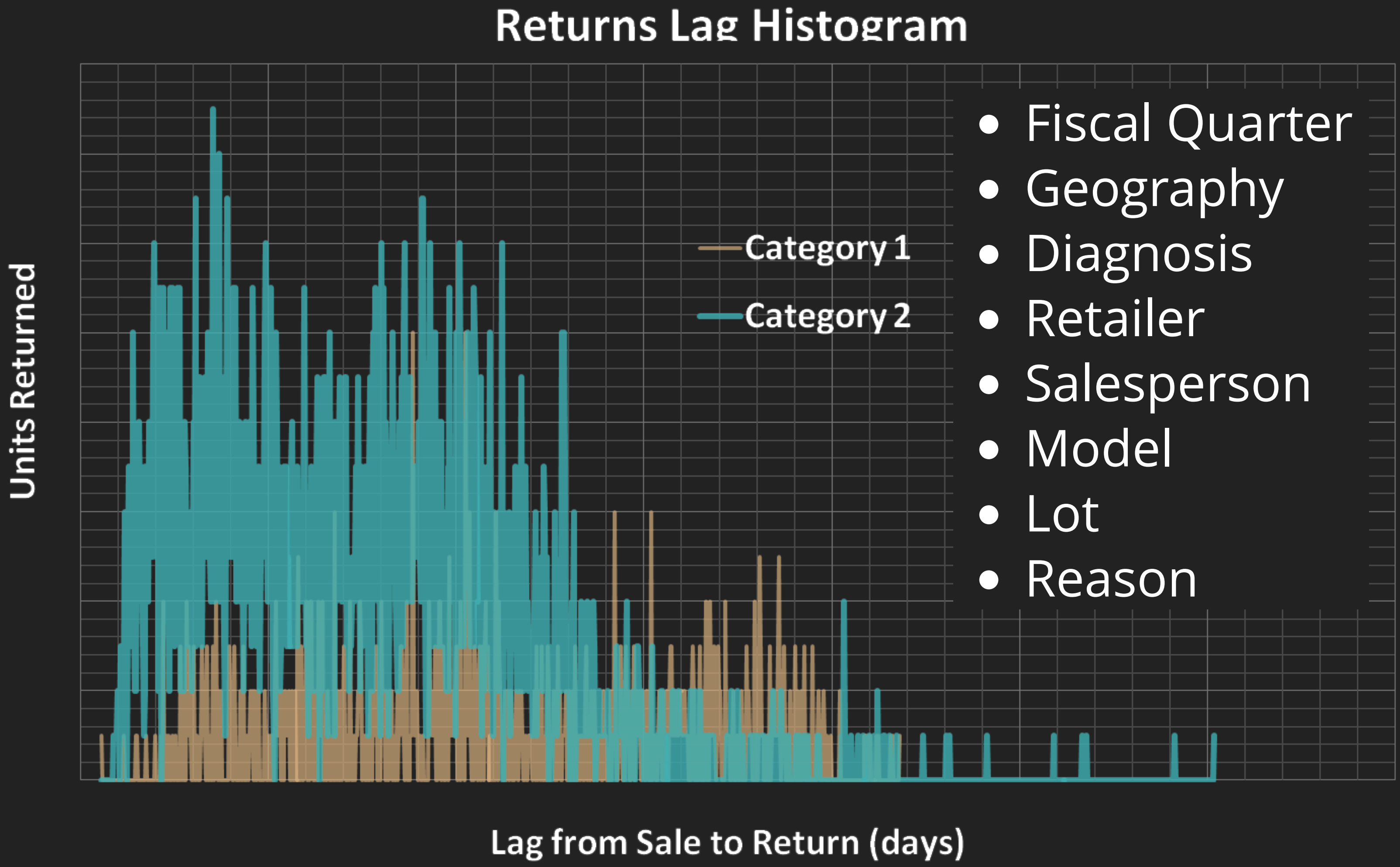


Weekly Sales Counts
Volume
Date
0
50
100
150
200
250
300
350
400
450
Jan 2000
Jan 2000
Feb 2000
Mar 2000
Apr 2000
May 2000
Jun 2000
Jul 2000
Aug 2000
Sep 2000
Oct 2000
Nov 2000
Dec 2000
Dec 2000
Jan 2001
Feb 2001
Mar 2001
Apr 2001
May 2001
Jun 2001
Jul 2001
Aug 2001
Sep 2001
Oct 2001
Nov 2001
Dec 2001
Dec 2001
Jan 2002
Feb 2002
Mar 2002
Apr 2002
May 2002
Jun 2002
Jul 2002
Aug 2002
Month-end
Surge

Rejects


Weekly Returns as Percent of All Returns to Date
Frequency (Probability or Count/Total)
4.5%
4%
3.5%
3%
2.5%
2%
1.5%
1%
0.5%
0%
Jan 2000
Jan 2000
Feb 2000
Mar 2000
Apr 2000
May 2000
Jun 2000
Jul 2000
Aug 2000
Sep 2000
Oct 2000
Nov 2000
Dec 2000
Dec 2000
Jan 2001
Feb 2001
Mar 2001
Apr 2001
May 2001
Jun 2001
Jul 2001
Aug 2001
Sep 2001
Oct 2001
Nov 2001
Dec 2001
Dec 2001
Jan 2002
Feb 2002
Mar 2002
Apr 2002
May 2002
Jun 2002
Jul 2002
Aug 2002

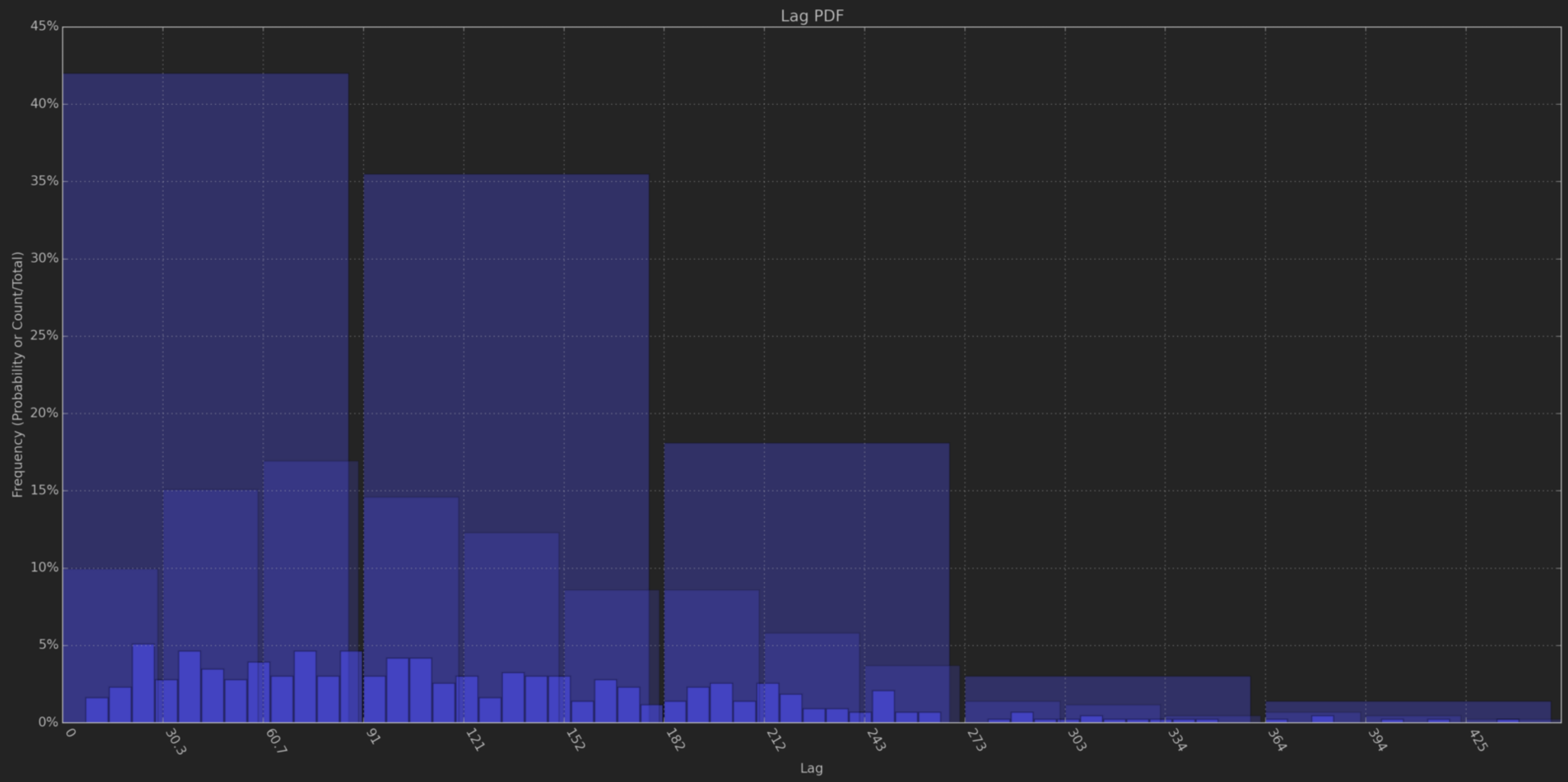
4. Question the Question



- Fiscal Quarter
- Geography
- Diagnosis
- Retailer
- Salesperson
- Model
- Lot
- Reason

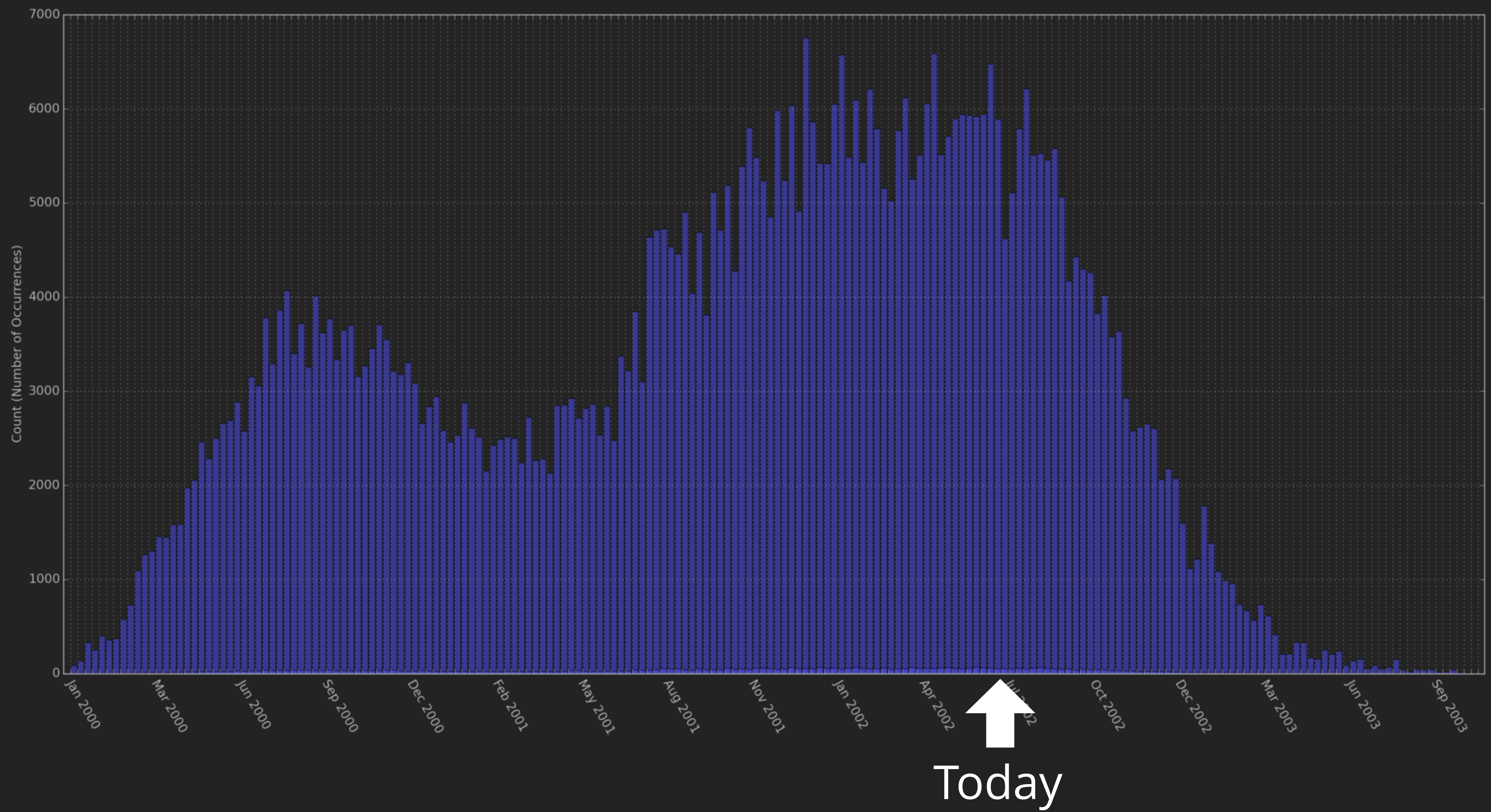
SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", "3", **"4"**, "5", or "6"

Lag



Lag PDF
45%
40%
35%
30%
25%
20%
15%
10%
5%
0%
Frequency (Probability or Count/Total)
0
30.3
60.7
91
121
152
182
212
243
273
303
334
364
394
425
Lag

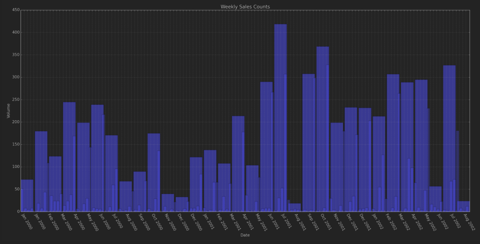
Lagged Sales



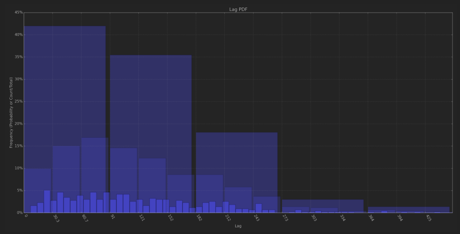
Predicted Returns

Sales **Lag Process** **Rejects**

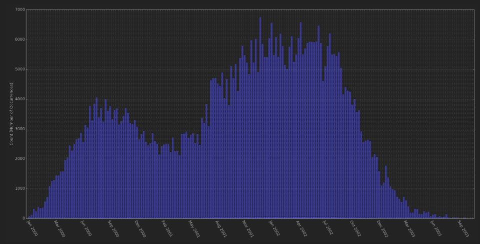
$$S(t) \Rightarrow \boxed{H(t,\tau)} \Rightarrow R(t)$$



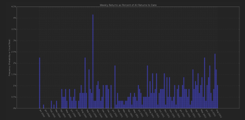
*



=

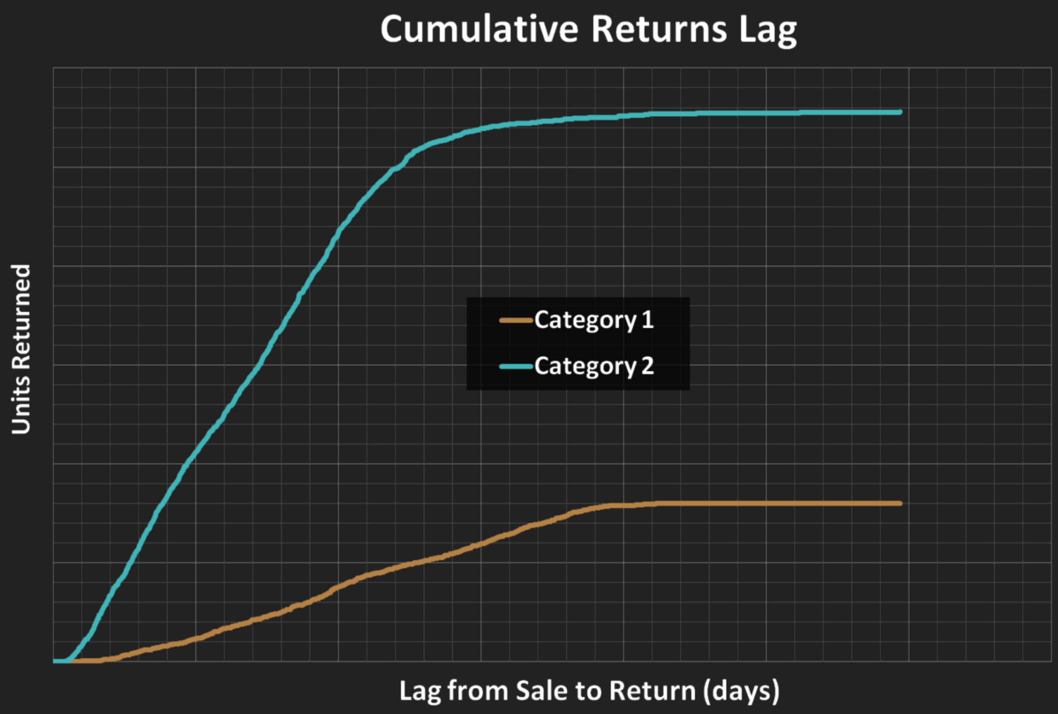


÷



4. Analyze the Question

Cumulative histograms focus attention on final total

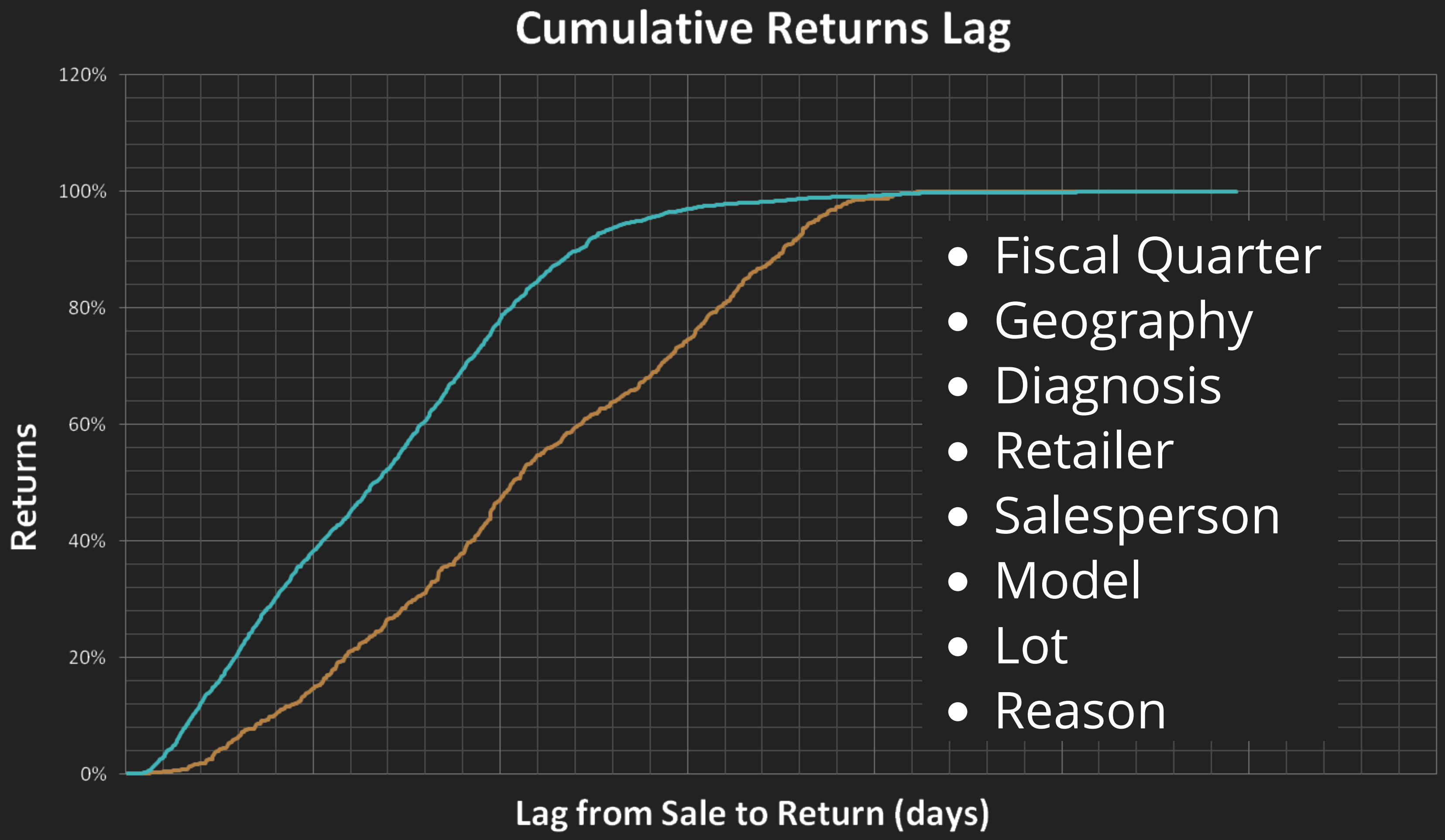


Product returns stop when...

- You stop accepting returns
- You stop counting
- You stop selling

SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", "3", **"4"**, "5", or "6"

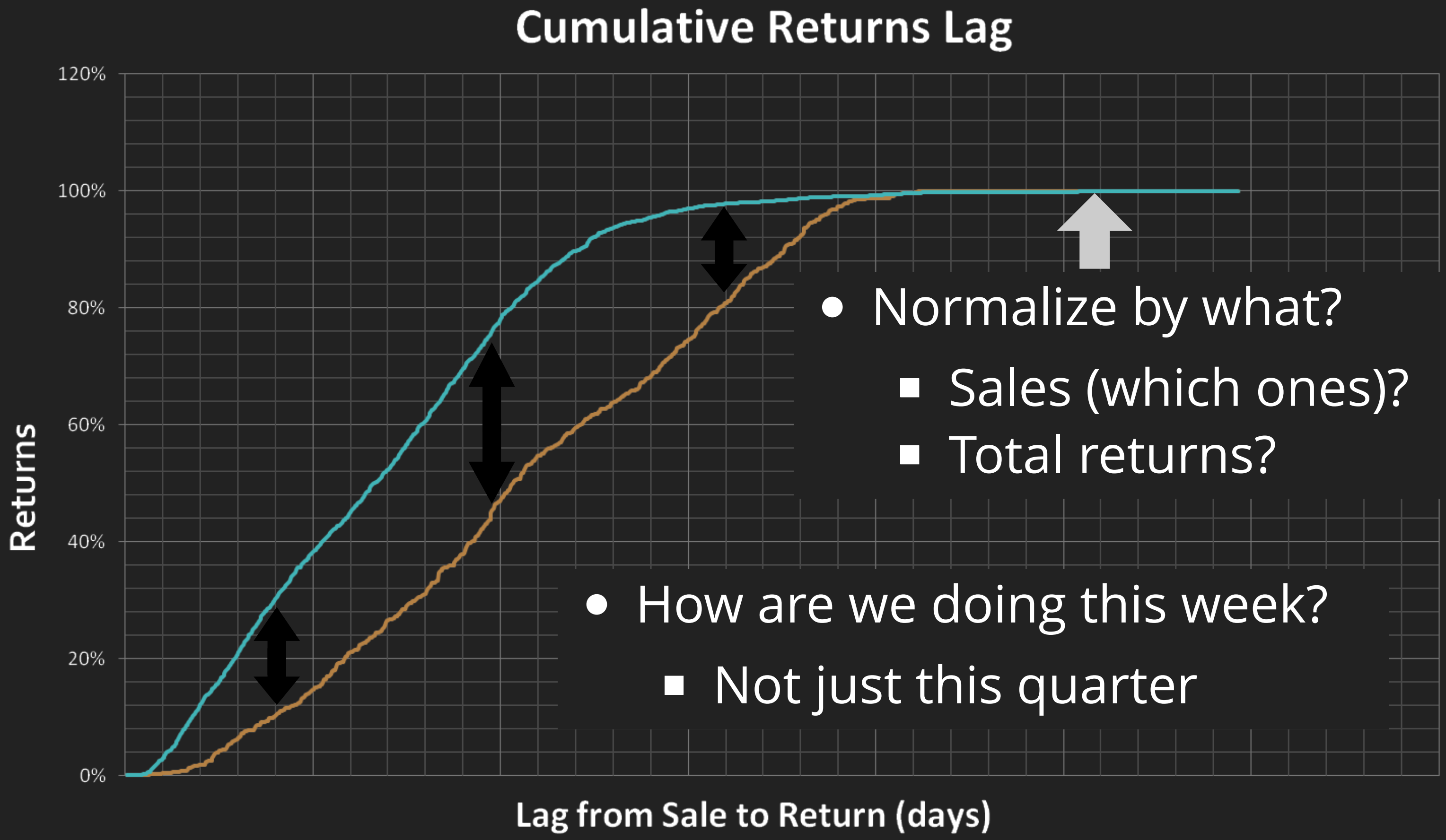
4. Normalize & Compare



- Fiscal Quarter
- Geography
- Diagnosis
- Retailer
- Salesperson
- Model
- Lot
- Reason

4. Analyze the Question

Normalize histograms to compare categories



- Normalize by what?
  - Sales (which ones)?
  - Total returns?

- How are we doing this week?
  - Not just this quarter

SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", "3", **"4"**, "5", or "6"

4. Question the Question

Unsupervised natural language processing?

President inaugural speeches

Target category = political party

4. Question the Question

What are the US Presidents' political parties based on speeches?

4. Question the Question

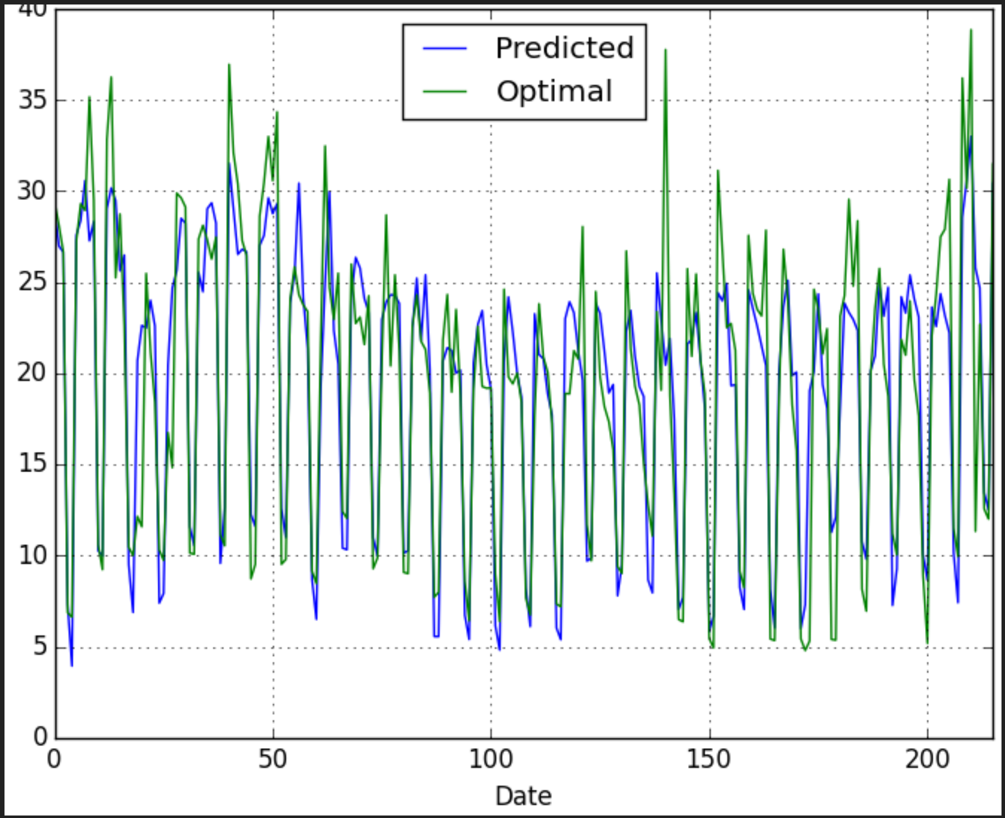
What are the US Presidents' political parties based on speeches?

4. Question the Question

- The category you're interested in will not likely be the most important "factor" in the NLP statistics
- Dimension reduction (SVD, PCA) can identify factors
  - Word-sets that are most significant
- These represent the "themes"
  - Interpretation of these "themes" is up to you
  - Statistics $\neq$ Meaning

5. Deep Nets Run Aground

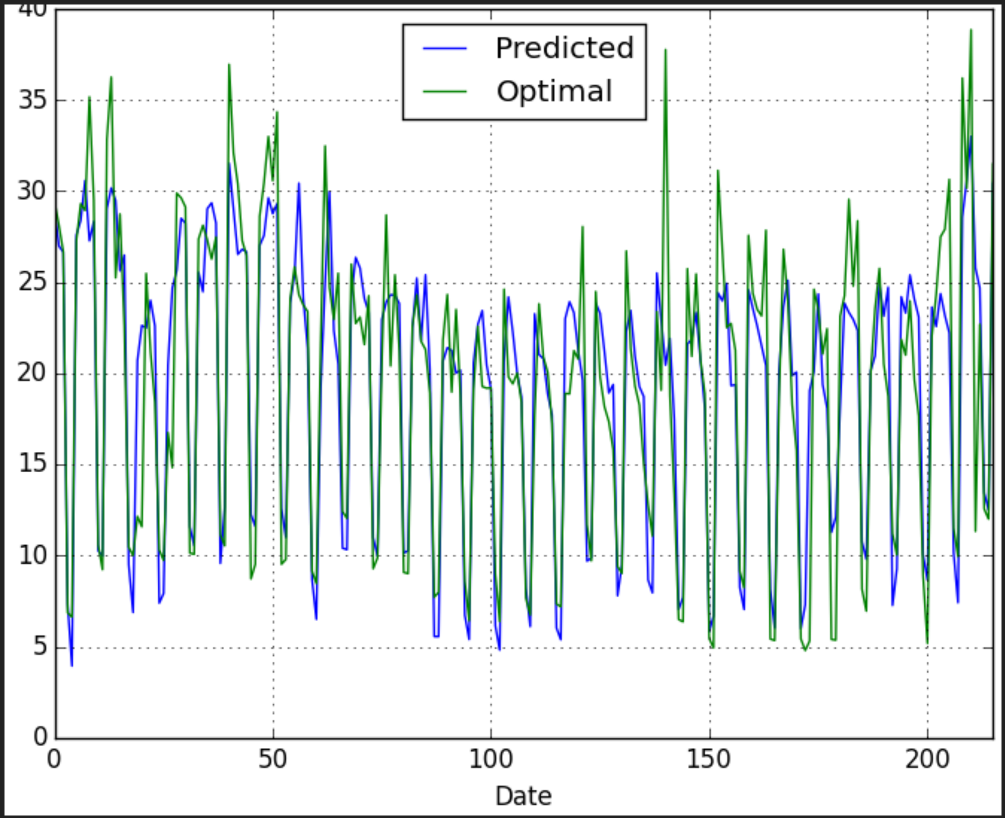
Deep net performs well!



SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", "3", "4", **"5",** "6"

5. Deep Nets Run Aground

Not so fast... it's overfitting



5. Deep Nets Run Aground

$$\mathrm{p} \Rightarrow \mathrm{W}_{S^k, S^{(k+1)}}^{k} \Rightarrow \mathrm{a}$$

$$\mathrm{a} = \mathrm{W}_{S^k, S^{(k+1)}}^{k}\ \mathrm{p}$$

$$\mathbf{W}_{n,m}^{k} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix}$$

- Conventional Hebb rule

$$\mathrm{W}^{new} = W^{old} + \mathrm{t}_q \mathrm{p}_q^T$$

- Hebb "delta" rule

$$\mathrm{W}^{new} = W^{old} + \alpha(\mathrm{t}_q - \mathrm{a}_q)\mathrm{p}_q^T$$

5. Shallow Data

$$\mathrm{p} \Rightarrow \mathrm{W}^{k}_{S^k, S^{(k+1)}} \Rightarrow \mathrm{a}$$

$$\mathrm{a} = \mathrm{W}^{k}_{S^k, S^{(k+1)}}\ \mathrm{p}$$

$$\mathbf{W}^k_{n,m} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix}$$

- Model degree:

$$\sum_k S^k S^{(k+1)}$$

- Training data DOF:

$$S^1 S^3 N_{samples}$$ (independent samples)

5. Shallow Data

$$\mathrm{p} \Rightarrow \mathrm{W}^{k}_{S^k, S^{(k+1)}} \Rightarrow \mathrm{a}$$

$$\mathrm{a} = \mathrm{W}^{k}_{S^k, S^{(k+1)}}\ \mathrm{p}$$

$$\mathbf{W}^{k}_{n,m} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m} \\ \vdots & \vdots & & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,m} \end{bmatrix}$$

- Model degree:

$$S^1 S^2 + S^2 S^3$$

(1 hidden layer)

- Training data DOF:

$$(S^1 + S^3) N_{samples}$$

(**independent** samples)

5. Bottom Line

$$N_{hidden} << N_{training}$$

bit.ly/nntune

6. Escape the Maze
Find Connections
(Actionable Insight)

18 databases
> 10k tables
> 100k fields
> 10M records/table

6. Escape from the Maze

- Tight heuristics vital for efficient graph search
- "Always turn right" is not good enough

SMS: 7707-2-TOTAL or (770) 728-6825 MSGS: "1", "2", "3", "4", "5", or **"6"**

6. Escape from the Maze

- Don't bother with "exhaustive" correlation search

$$\text{complexity} \approx O(\underbrace{M^2}_{10^7}\underbrace{N^2}_{10^5}) \approx 10^{24}$$

- Find db relationships using meta-data
  - min, max, median
  - #records
  - #distinct
  - for reals: mean, std

$$\text{complexity} \approx O(MNlog(N)) \approx 10^{13}$$

Human Heuristics

- **Business knowledge narrows search:**
  - Repair technicians
  - Product designers
  - Factory managers
  - Suppliers
  - Sales channels
  - Call center

Accidental "Experiements"

- **Look for differences in**
  - Model
  - Lot
  - Product
  - Sales Channel
  - Customer Demographic
  - Region/Culture
- **Look for ...**
  - New/deleted features
  - Documentation updates
  - Cost-saving parts changes
  - Production facilities (outsourced vs insourced)

Kruskal's Algorithm
Minimum Spanning Tree

1. Add lowest cost edge with new node
2. Repeat until all nodes accounted for

Produces one graph for each connected subgraph

Built into python graph library (`networkx`):

```
def minimum_spanning_zipcodes():
    zipcode_query_sequence = []
    G = build_graph(api.db, limit=1000000)
    for CG in nx.connected_component_subgraphs(G):
        for edge in nx.minimum_spanning_edges(CG):
            zipcode_query_sequence += [edge[2]['zipcode']]
    return zipcode_query_sequence
```

A* Algorithm
Minimum Path to Goal

Provably optimal and optimally efficient

But typical data relationship graph has large branching factor

```
from networkx.algorithms.shortest_paths import astar_path
astar_path(G, source, target, heuristic=None)
```

Built into python graph library (`networkx`)

A* Algorithm
Minimum Path to Goal

Provably optimal and optimally efficient

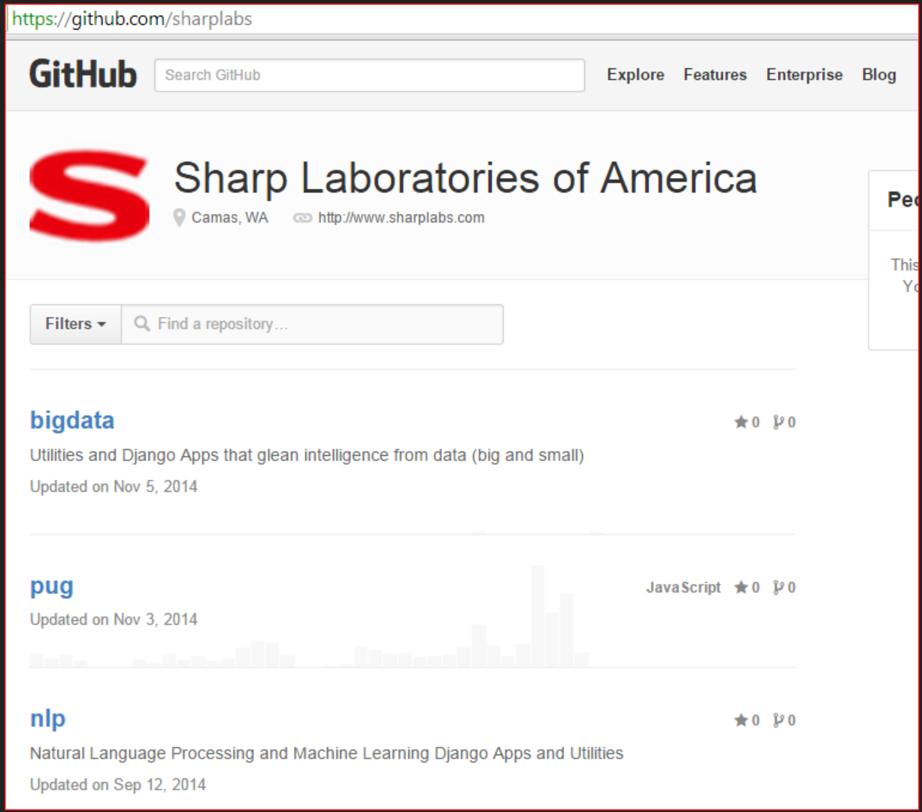
Built into python graph library (`networkx`)

```
from networkx.algorithms.shortest_paths import astar_path
astar_path(G, source, target, heuristic=None)
```

You better have a good heuristic!

It's Open Source!

github.com/sharplabs



bit.ly/pawsvote

Choose Your Story

7707-2-TOTAL

(770) 728-6825

1. Only Nyquist Knows
2. The Meaning of Mean
3. Data Dearth
4. Question the Question
5. Deep Net Runs Aground
6. Escape the Maze

- Consider sample rate
- Classify before mean
- Explore data sources
- Reject rate metric
- data > nodes x inputs
- Lazy correlation

References

- "Data Driven Documents"
  - 2011, Mike Bostock
- "Data Science with `pug`"
  - 2014, Lane, Zen, Kowalski, PDX Python U.G.
- "Neural Network Design"
  - 2014, Hagan, Demuth, et. al., OKSU
- "Forecasting Product Returns"
  - 2001, Toktay, INSEAD
- `scipy.ransac`
  - 2014, Andrew D. Straw
- "Choose Your Own Adventure Presentation"
  - 2014, Matt Makai