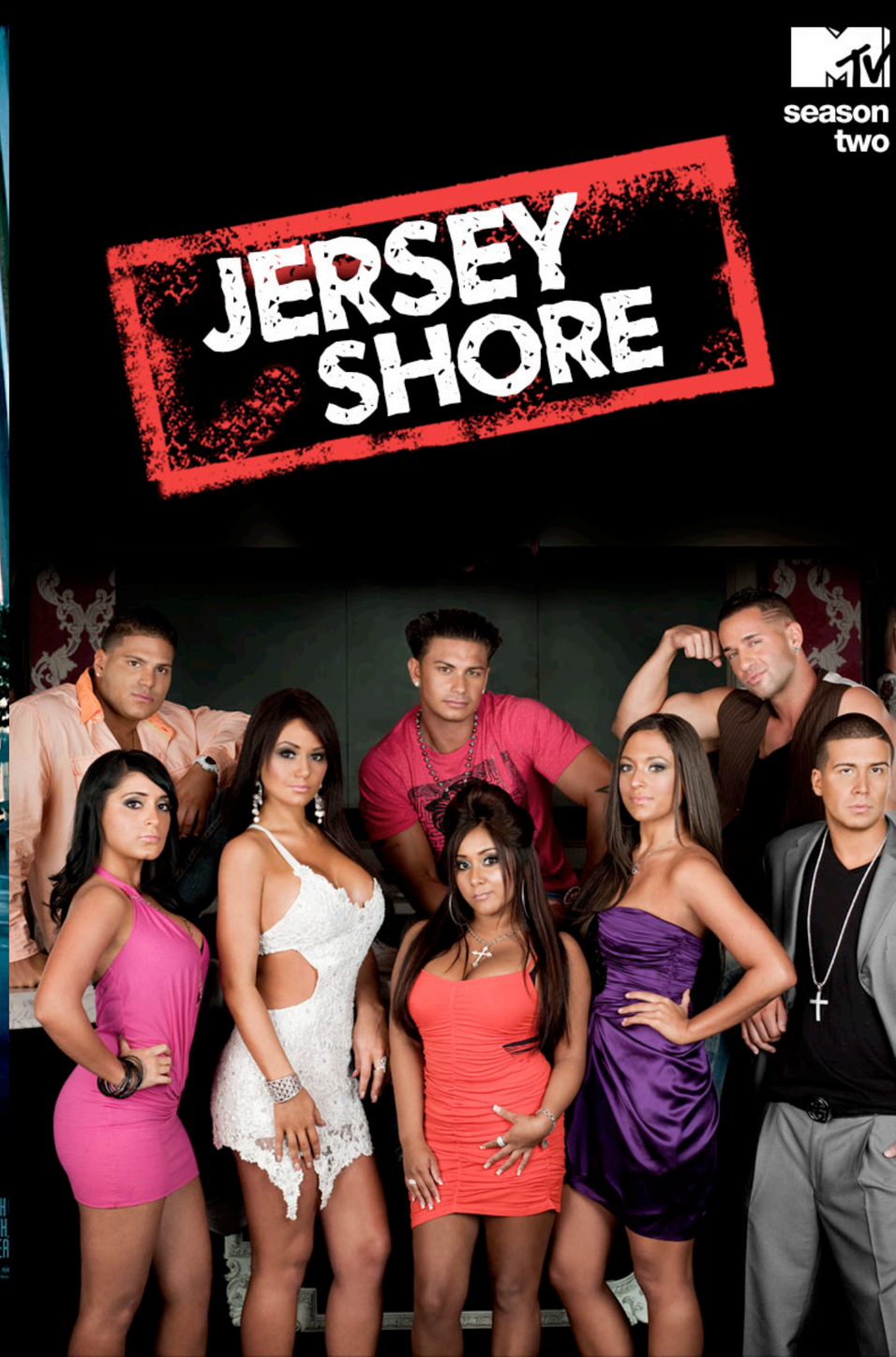
Data to the People?

Reflections on Trying to Help People Struggle Less with Data

Jeffrey Heer @jeffrey_heer
University of Washington

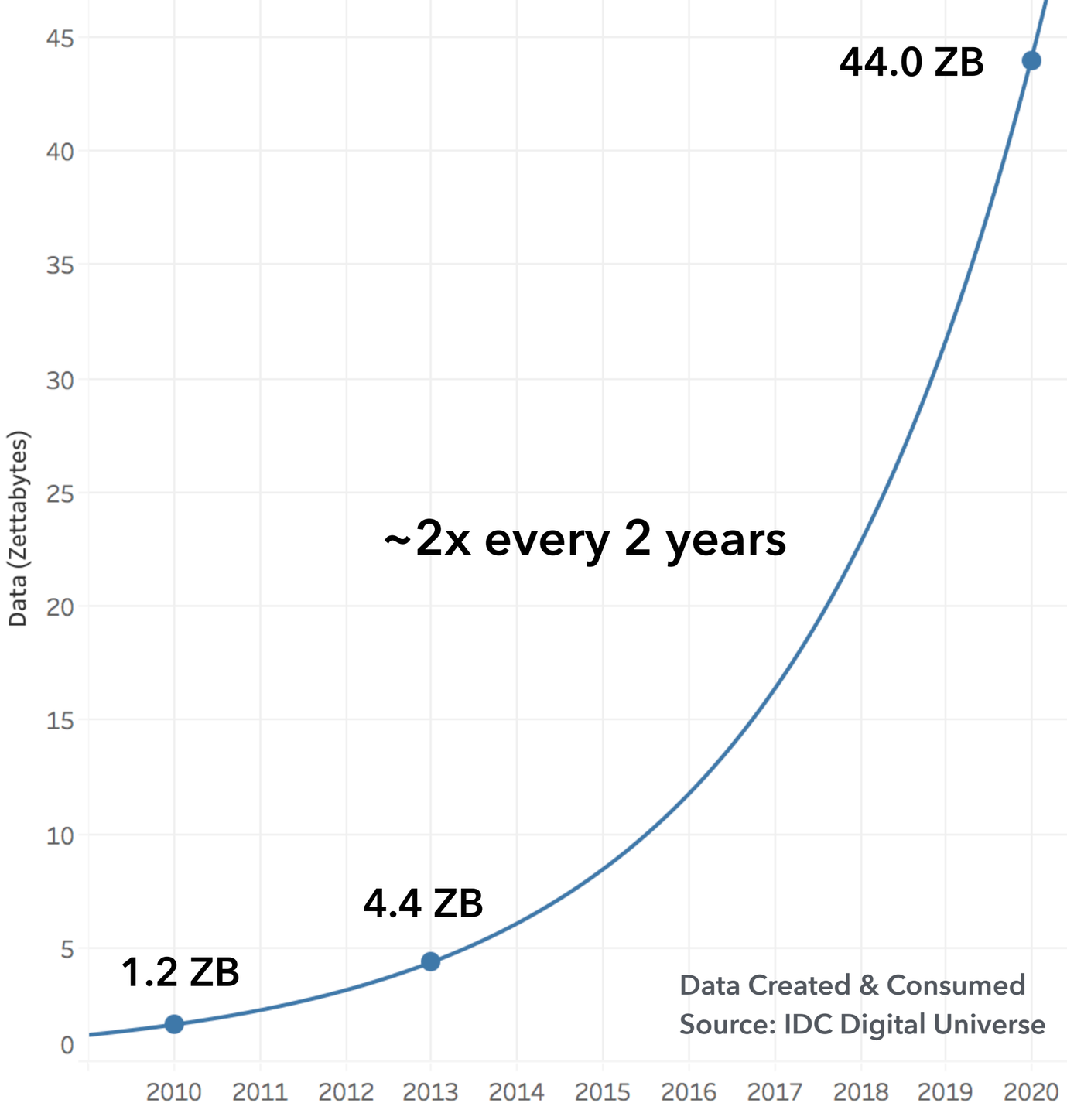
2010


LEONARDO DICAPRIO
KEN WATANABE
JOSEPH GORDON-LEVITT
MARION COTILLARD
ELLEN PAGE
TOM HARDY
CILLIAN MURPHY
TOM BERENGER
AND MICHAEL CAINE
YOUR MIND IS THE SCENE OF THE CRIME.
A FILM BY CHRISTOPHER NOLAN
INCEPTION
FROM THE DIRECTOR OF THE DARK KNIGHT
WARNER BROS. PICTURES PRESENTS
IN ASSOCIATION WITH LEGENDARY PICTURES A SYNCOPY PRODUCTION A FILM BY CHRISTOPHER NOLAN LEONARDO DiCAPRIO "INCEPTION" KEN WATANABE JOSEPH
MARION COTILLARD ELLEN PAGE TOM HARDY CILLIAN MURPHY TOM BERENGER AND MICHAEL CAINE MUSIC BY HANS ZIMMER EDITED BY LEE SMITH
PRODUCTION DESIGNER GUY HENDRIX DYAS DIRECTOR OF PHOTOGRAPHY WALLY PFISTER, A.S.C. EXECUTIVE PRODUCERS CHRIS BRIGHAM THOMAS TULL PRODUCED BY EMMA THOMAS CHRISTOPHER
WRITTEN AND DIRECTED BY CHRISTOPHER NOLAN
inceptionmovie.com
EXPERIENCE IT JULY 16 IN THEATERS AND IMAX


season two
JERSEY SHORE


"Big Data"

2010: 1,200 exabytes & exponential growth



The ability to take data—to be able to **understand** it, to **process** it, to **extract value** from it, to **visualize** it, to **communicate** it—that's going to be a hugely important skill in the next decades, … because now we really do have essentially free and ubiquitous data. So the complimentary **scarce factor** is the ability to **understand** that data and **extract value** from it.

Hal Varian, Google's Chief Economist
The McKinsey Quarterly, Jan 2009

Four major influences act on data analysis today:

1. The formal theories of statistics.
2. Accelerating developments in computers and display devices.
3. The challenge, in many fields, of more and larger bodies of data.
4. The emphasis on quantification in a wider variety of disciplines.

1966 Tukey & Wilk, Data Analysis & Statistics

Accordingly, both approaches and techniques need to be structured so as to **facilitate human involvement and intervention**.

Some implications for effective analysis are: (1) it is essential to have convenience of **interaction of people and intermediate results** and (2) at all stages of data analysis, the outputs need to be **matched to the capabilities of the people who use it and want it.**

1966 Tukey & Wilk, Data Analysis & Statistics

dp = data to the people

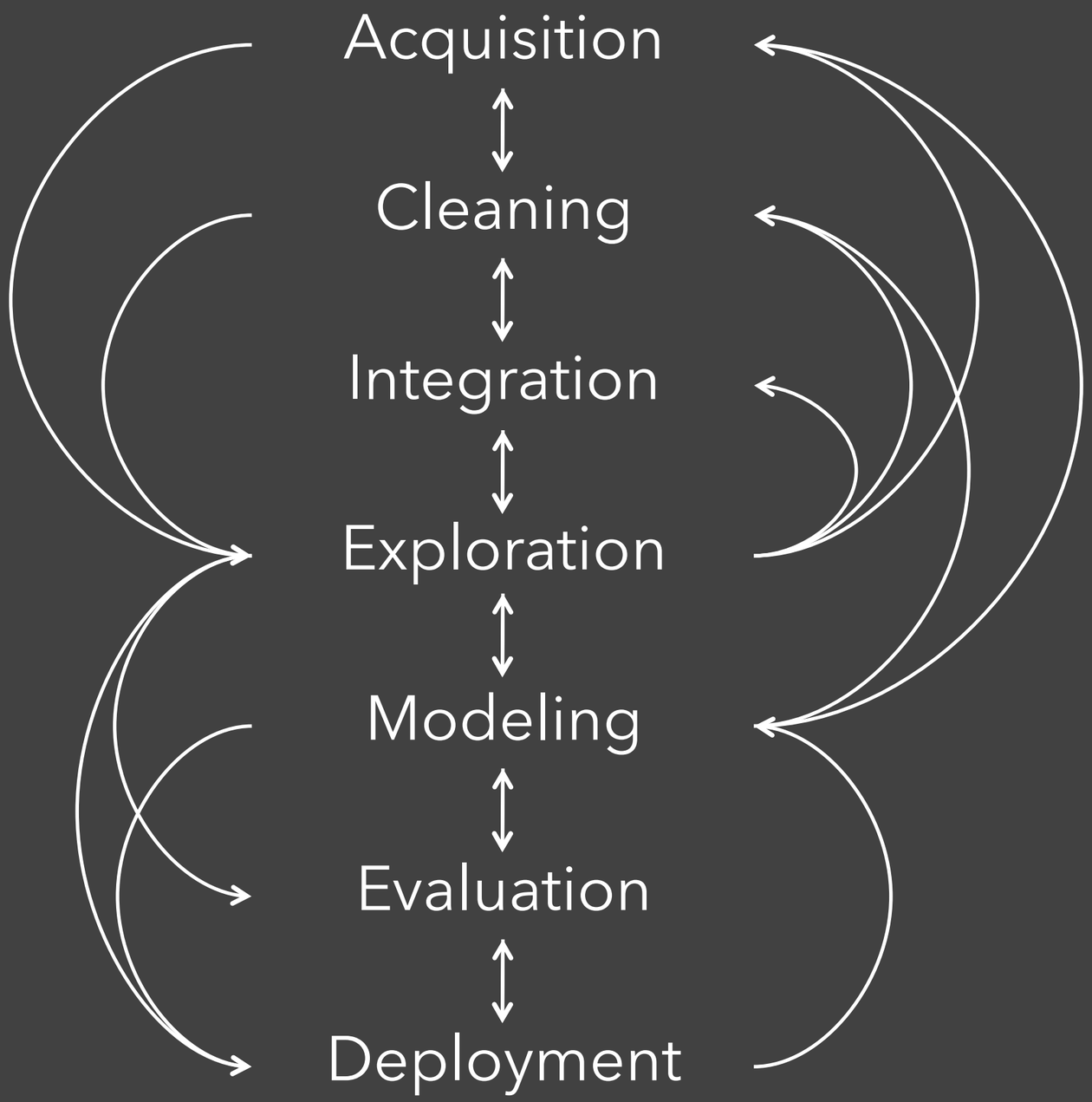
Facilitating interactions between people and data throughout the analytic lifecycle.

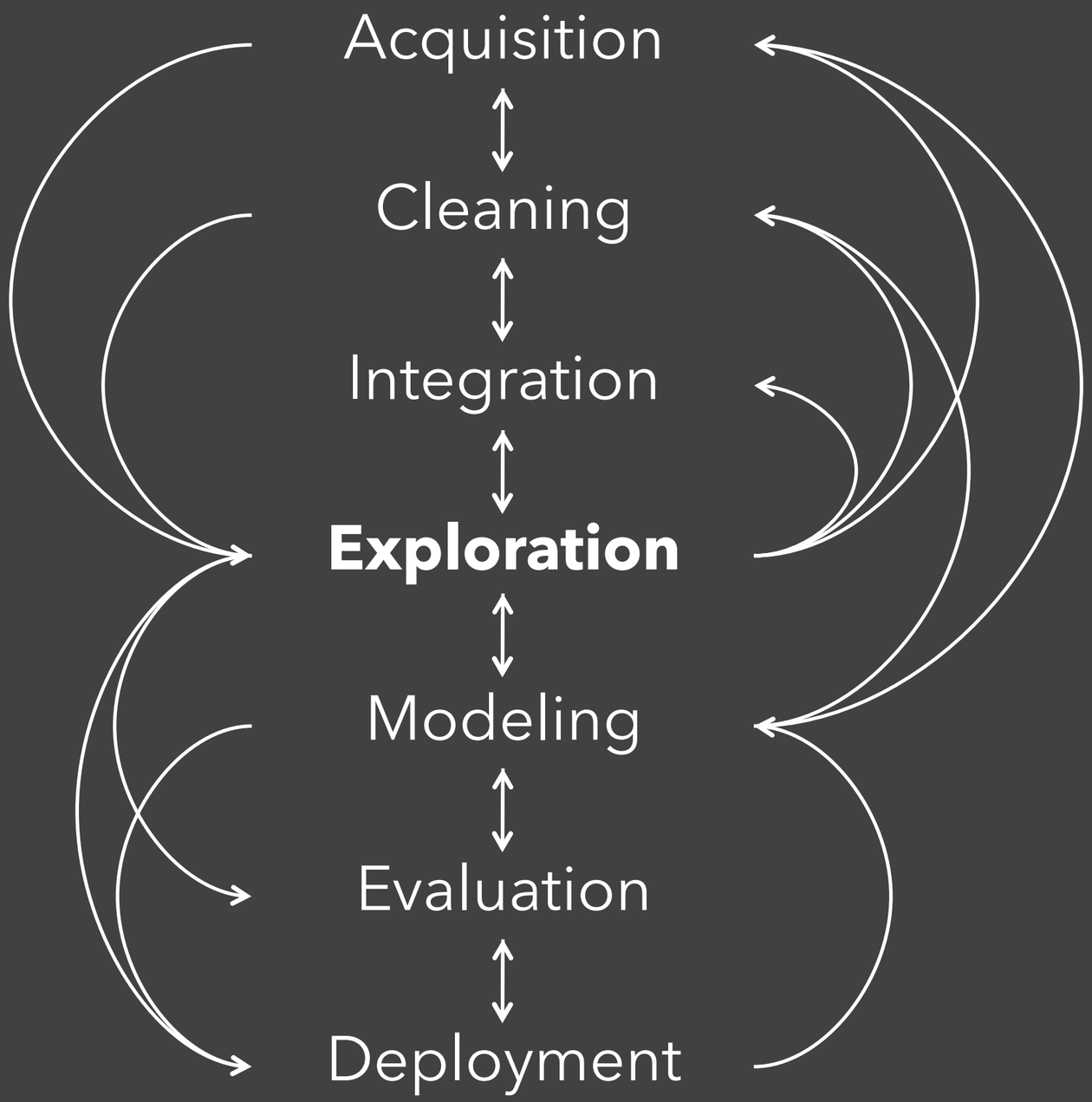
Jeff Heer
Stanford

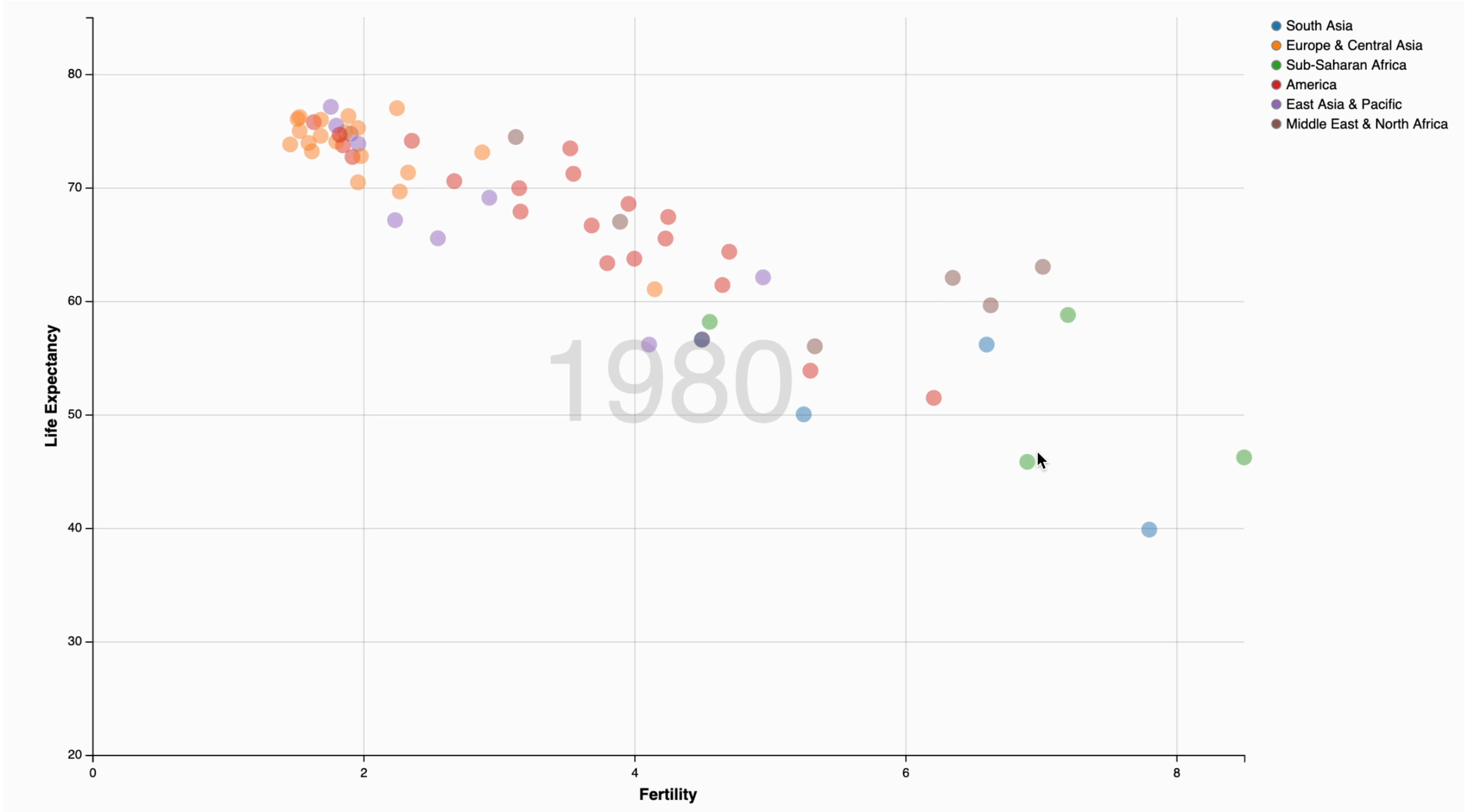
Maneesh Agrawala
Berkeley

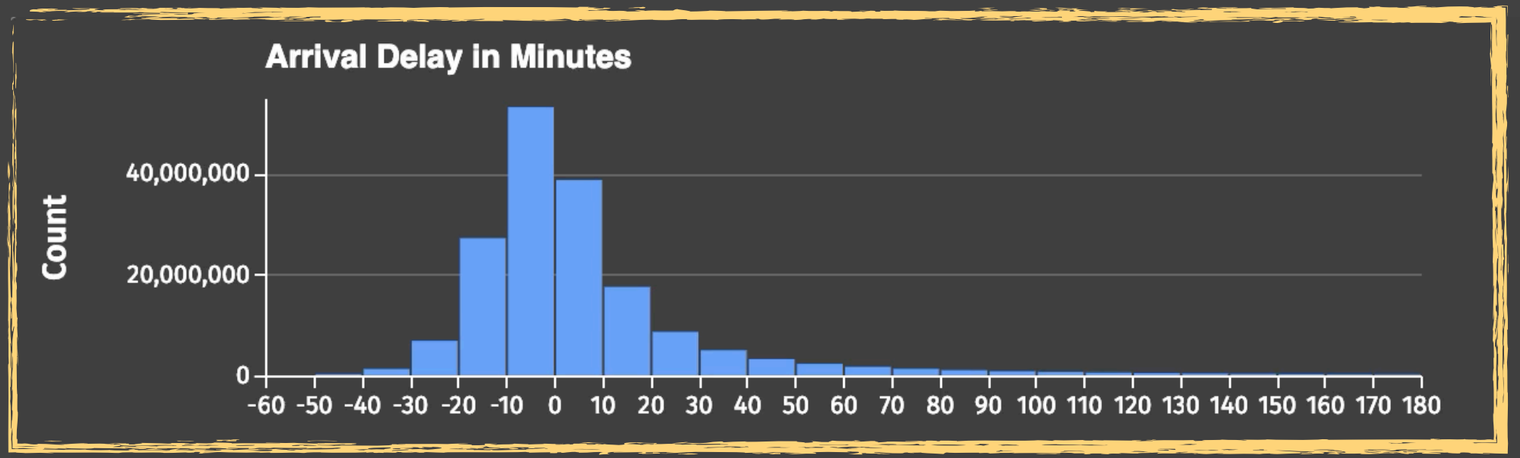
Joe Hellerstein
Berkeley

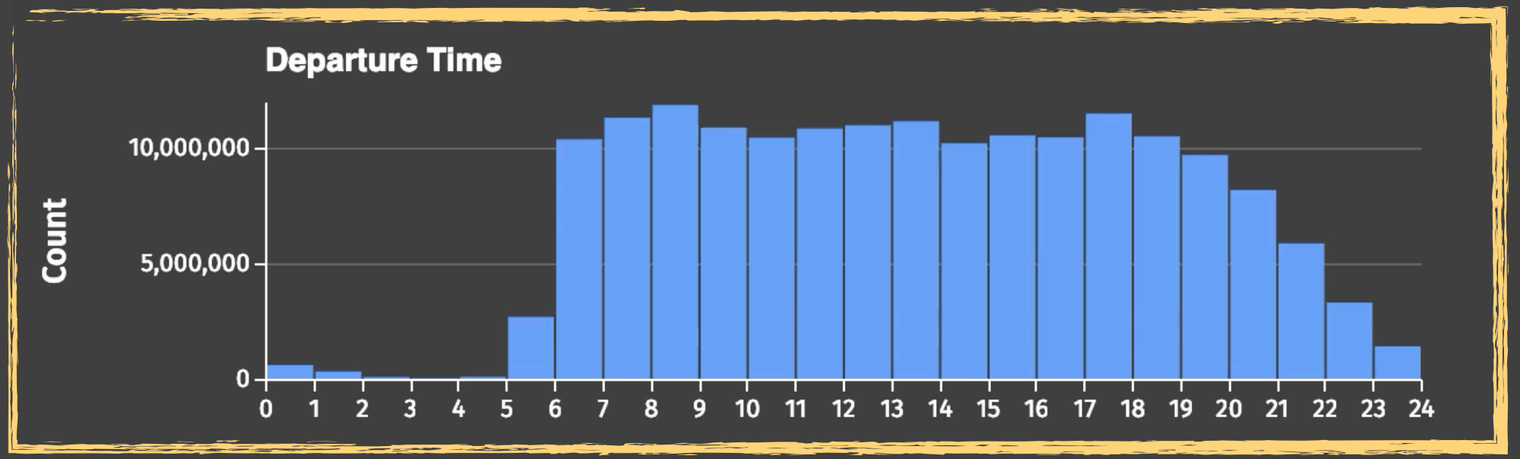
Tapan Parikh
Berkeley

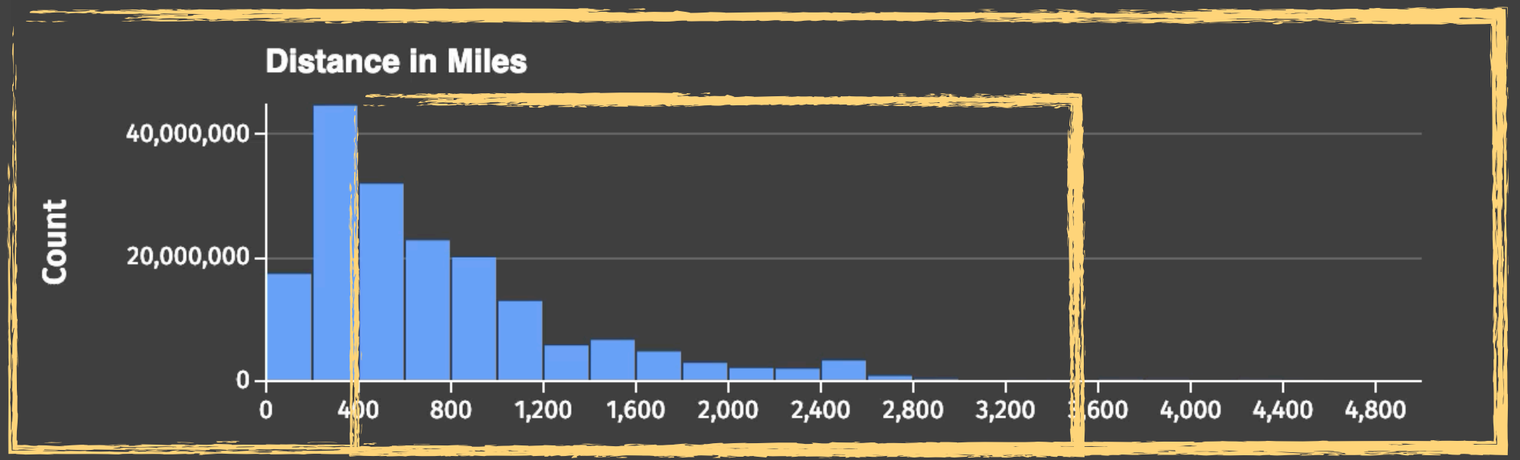

Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment


Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment

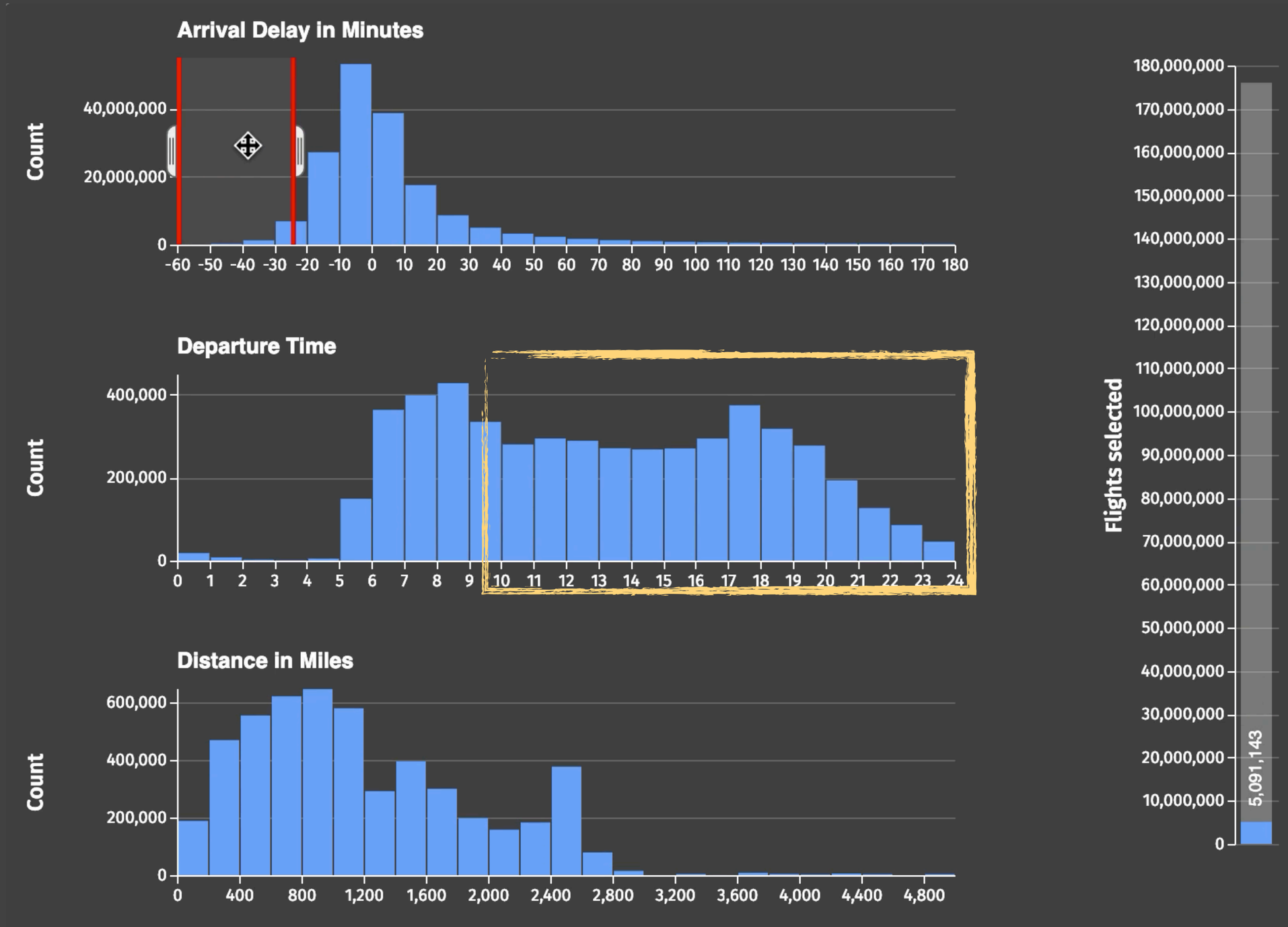

South Asia
Europe & Central Asia
Sub-Saharan Africa
America
East Asia & Pacific
Middle East & North Africa
1980
Life Expectancy
Fertility
80
70
60
50
40
30
20
0
2
4
6
8

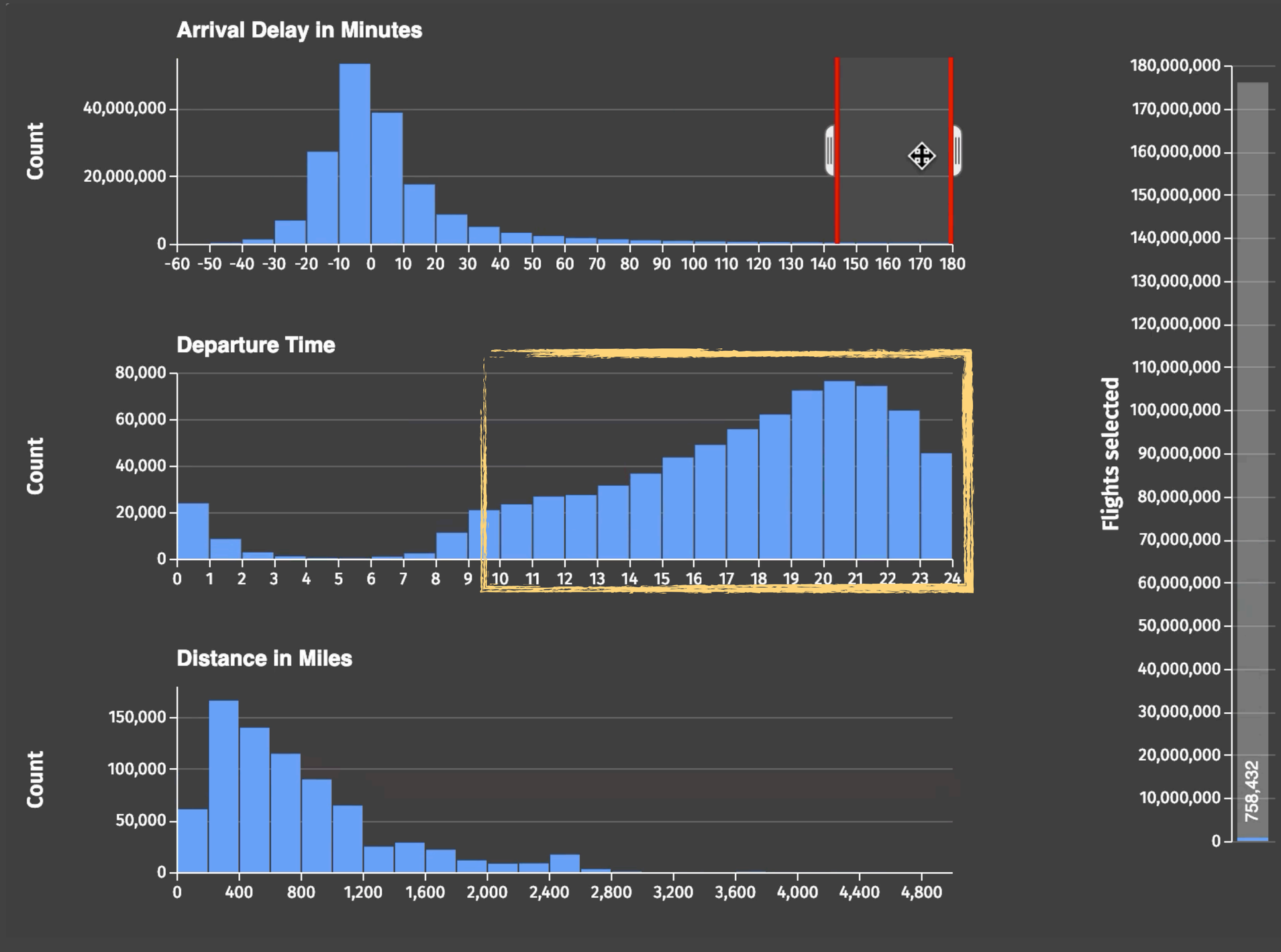

Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180

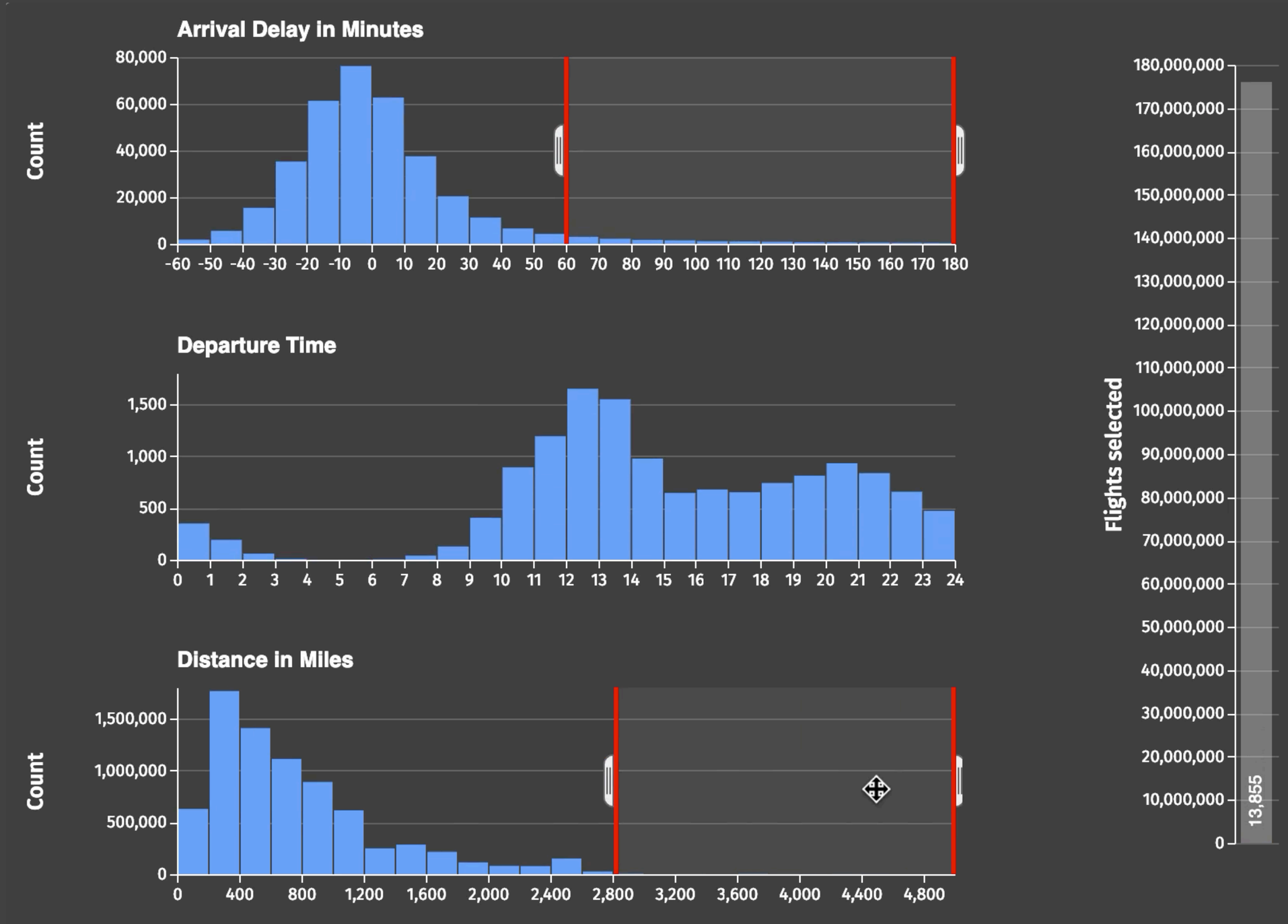

Departure Time
Count
10,000,000
5,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24


Distance in Miles
Count
40,000,000
20,000,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800


Flights selected
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
176,063,415


Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
400,000
200,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
600,000
400,000
200,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
5,091,143


Arrival Delay in Minutes
Count
40,000,000
20,000,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
80,000
60,000
40,000
20,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
150,000
100,000
50,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
758,432


Arrival Delay in Minutes
Count
80,000
60,000
40,000
20,000
0
-60 -50 -40 -30 -20 -10 0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150 160 170 180
Departure Time
Count
1,500
1,000
500
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Distance in Miles
Count
1,500,000
1,000,000
500,000
0
0 400 800 1,200 1,600 2,000 2,400 2,800 3,200 3,600 4,000 4,400 4,800
Flights selected
180,000,000
170,000,000
160,000,000
150,000,000
140,000,000
130,000,000
120,000,000
110,000,000
100,000,000
90,000,000
80,000,000
70,000,000
60,000,000
50,000,000
40,000,000
30,000,000
20,000,000
10,000,000
0
13,855

How might we **flexibly author interactive visualizations?**

prefuse

Protovis

Vega

Data-Driven Documents

Visualization System Evolution

2002-09 Object-Oriented Libraries (Prefuse, Flare)

Requires software engineering to use and extend

2009-12 JavaScript Embedded DSLs (Protovis, D3)

Rich composable primitives, but specific to JavaScript

Interaction via standard event callbacks

2012-?? Declarative DSLs (Vega, Vega-Lite)

Formal specification of visualization and interactions

APIs in various programming languages

DB Concepts

Columnar storage

Query optimization

Closed language

Data join between marks and data

Declarative language

Query plan / runtime

Stream management

Query Evaluation Techniques for Large Datab

GOETZ GRAEFE

Portland State University, Computer Science Department, P.O. Box 75

Database management systems will continue to manag
efficient algorithms for accessing and manipulating larg
required to provide acceptable performance. The advent o
database systems will not solve this problem. On the co
exacerbate the problem: In order to manipulate large
efficiently as today's database systems manipulate sim
algorithms and software will become more complex, a
algorithm and architectural issues is essential for the de
software.

This survey provides a foundation for the design a
execution facilities in new database management syste
practical query evaluation techniques for both relation
systems, including iterative execution of complex query
sort- and hash-based set-matching algorithms, types o
their implementation, and special operators for emergi

Categories and Subject Descriptors: E.5 [**Data**]: Files; H
Systems—*query processing*

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Complex query ev
evaluation plans; extensible database systems, iterato
systems, operator model of parallelization, parallel a
systems, set-matching algorithms, sort-hash duality

INTRODUCTION

Effective and efficient management of large data volumes is necessary in virtually all computer applications, from business data processing to library information retrieval systems, multimedia applications with images and sound, computer-aided design and manufacturing, real-time process control, and scientific computation. While database management systems are standard tools in business data processing, they are only slowly being introduced to all the other areas.

In mos
mains, c
have tra
reasons.
and ma
applicati
nance u
into ser
models
grammi
ing this
to accep

TelegraphCQ: Continuous Dataflow P
Uncertain World+

Sirish Chandrasekaran, Owen Cooper, Amol Deshpande, Michael J. F
Wei Hong*, Sailesh Krishnamurthy, Sam Madden, Vijayshankar Rama

University of California, Berkeley
*Intel Berkeley Laboratory
**IBM Almaden Research Center
http://telegraph.cs.berkeley.edu

Abstract

Increasingly pervasive networks are leading towards a world where data is constantly in motion. In such a world, conventional techniques for query processing, which were developed under the assumption of a far more static and predictable computational environment, will not be sufficient. Instead, query processors based on adaptive dataflow will be necessary. The Telegraph project has developed a suite of novel technologies for continuously adaptive query processing. The next generation Telegraph system, called TelegraphCQ, is focused on meeting the challenges that arise in handling large streams of continuous queries over high-volume, highly-variable data streams. In this paper, we describe the system architecture and its underlying technology, and report on our ongoing implementation effort, which leverages the PostgreSQL open source code base. We also discuss open issues and our research agenda.

1 INTRODUCTION

The deployment of pervasive communications infrastructure ranging from short-range wireless *ad hoc* sensor networks to globe-spanning intra- and internets has enabled new applications that process, analyze, and react to disparate data in a near real-time manner. Examples include: event-based business processing, profile-based data dissemination, and query processing over streaming data sources such as network monitors, sensors, and mobile devices. Such applications present challenges that cannot be met by existing database and data management technology. These challenges stem from their large scale, their deeply-netwo
environment, and t

In emerging
commodity of inte
is realized only wh
contrast to traditi
data can be as
locations, data i
moving and char
management for
processing that
information as th

1.1 Data M

Traditional
inappropriate a
processing supp

Streamin
systems wher
"pulling" data
our target app
data is contin
subtle differe
of the quer
traditional qu
and handlin
must instead
data streams
constraints
data must b
spooled to

Further
detailed sta
statistics f
Finally, it
time and/o
over strea
than queri

Cont
applicatio
which q
arrives a
of active

+ This work was funded in part by the NSF under grants IIS-0086057, SI-0122599, and EIA-0207603, by the IBM Faculty Partnership Award program, and by research funds from Intel, Microsoft, and the UC MICRO program.

The Design of the Borealis Stream Processing Engine

Daniel J. Abadi[1], Yanif Ahmad[2], Magdalena Balazinska[1], Uğur Çetintemel[2], Mitch Cherniack[3], Jeong-Hyon Hwang[2], Wolfgang Lindner[1], Anurag S. Maskey[3], Alexander Rasin[2], Esther Ryvkina[3], Nesime Tatbul[2], Ying Xing[2], and Stan Zdonik[2]

[1]MIT
Cambridge, MA

[2]Brown University
Providence, RI

[3]Brandeis University
Waltham, MA

Abstract

Borealis is a second-generation distributed stream processing engine that is being developed at Brandeis University, Brown University, and MIT. Borealis inherits core stream processing functionality from Aurora [14] and distribution functionality from Medusa [51]. Borealis modifies and extends both systems in non-trivial and critical ways to provide advanced capabilities that are commonly required by newly-emerging stream processing applications.

In this paper, we outline the basic design and functionality of Borealis. Through sample real-world applications, we motivate the need for dynamically revising query results and modifying query specifications. We then describe how Borealis addresses these challenges through an innovative set of features, including revision records, time travel, and control lines. Finally, we present a highly flexible and scalable QoS-based optimization model that operates across server and sensor networks and a new fault-tolerance model with flexible consistency-availability trade-offs.

1 Introduction

Over the last several years, a great deal of progress has been made in the area of stream processing engines (SPE). Several groups have developed working prototypes [1, 4, 16] and many papers have been published on detailed aspects of the technology such as data models [2, 5, 46], scheduling [8, 15], and load shedding [9, 20, 44]. While this work is an important first step, fundamental mismatches remain between the requirements of many streaming applications and the capabilities of first-generation systems.

This paper is intended to illustrate our vision of what second-generation SPE's should look like. It is driven by our experience in using Aurora [10], our own prototype, in several streaming applications including the Linear Road Benchmark [6] and several commercial opportunities. We present this vision in terms of our own design considerations for Borealis, the successor to Aurora, but it should be emphasized that the issues raised here represent general challenges for the field as a whole. We present specifics of our design as concrete evidence for why these problems are hard and as a first cut at how they might be approached. We envision the following three fundamental requirements for second-generation SPEs:

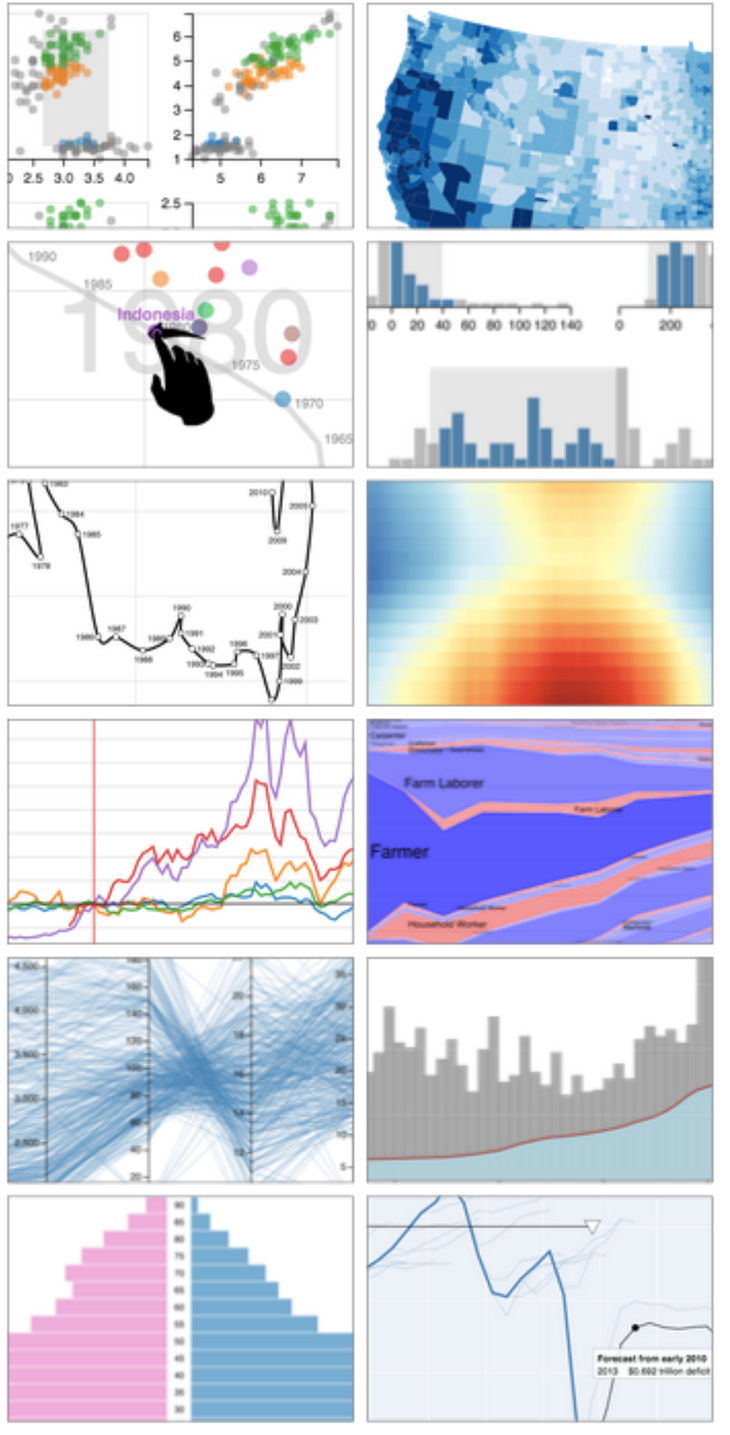
1. Dynamic revision of query results: In many real-world streams, corrections or updates to previously processed data are available only after the fact. For instance, many popular data streams, such as the Reuters stock market feed, often include so-called *revision records*, which allow the feed originator to correct errors in previously reported data. Furthermore, stream sources (such as sensors), as well as their connectivity, can be highly volatile and unpredictable. As a result, data may arrive late and miss its processing window, or may be ignored temporarily due to an overload situation [44]. In all these cases, applications are forced to live with imperfect results, unless the system has means to revise its processing and results to take into account newly available data or updates.

2. Dynamic query modification: In many stream processing applications, it is desirable to change certain attributes of the query at runtime. For example, in the financial services domain, traders typically wish to be alerted of *interesting* events, where the definition of "interesting" (i.e., the corresponding filter predicate) varies based on current context and results. In network monitoring, the system may want to obtain more precise results on a specific subnetwork, if there are signs of a potential Denial-of-Service attack. Finally, in a military stream application from Mitre, they wish to switch to a "cheaper" query when the system is overloaded. For the first two applications, it is sufficient to simply alter the operator parameters (e.g., window size, filter predicate), whereas the last one calls for altering the operators that compose the running query. Although current SPEs allow applications to substitute query networks with others at runtime, such manual substitutions impose high overhead and are slow to take effect as the new query network starts with an empty state. Our goal is to support low overhead, fast, and automatic modifications.

Another motivating application comes again from the financial services community. Universally, people working on trading engines wish to test out new trading strategies as well as debug their applications on historical data before they go live. As such, they wish to perform "time travel" on input streams. Although this last example can be supported

Vega is a *Visualization Grammar*

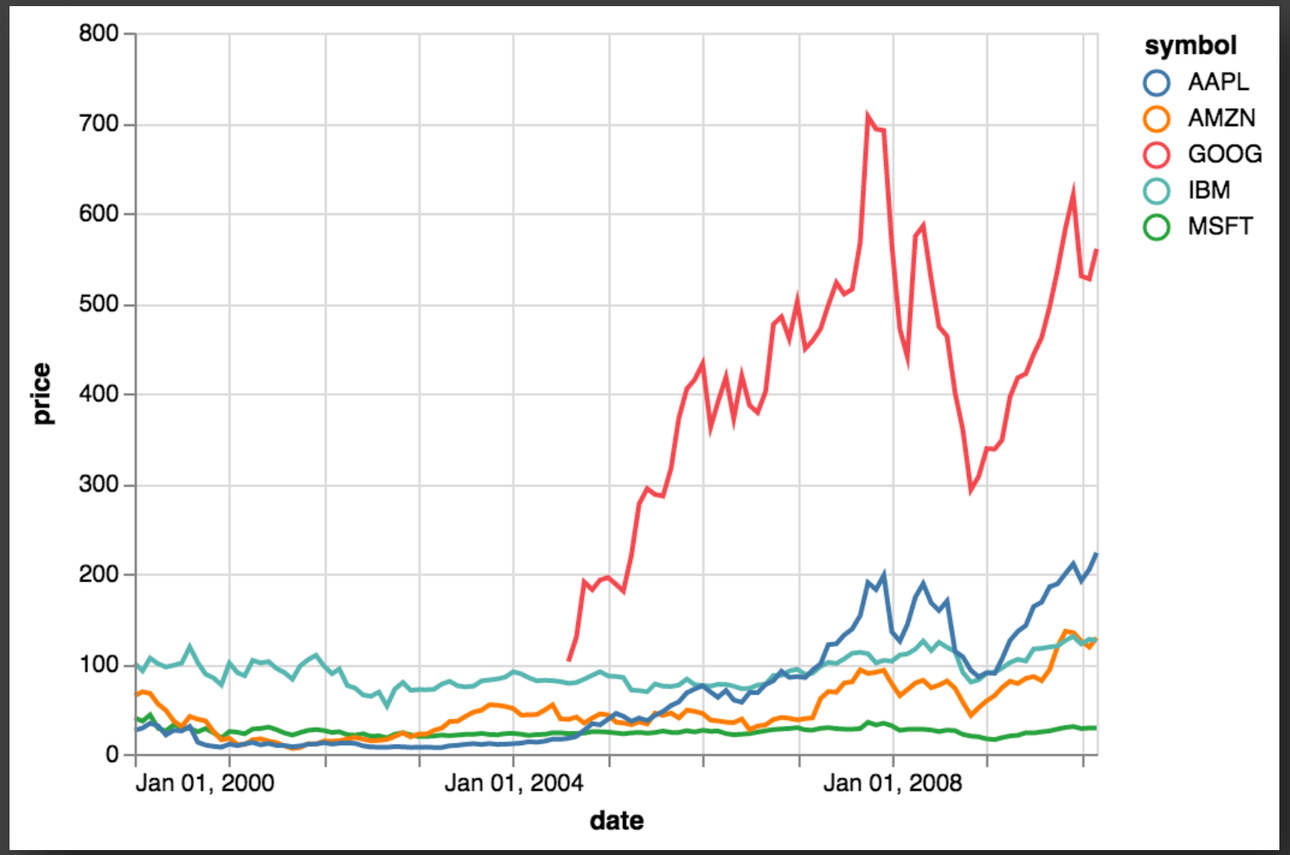
Similar in spirit to how SQL provides a language for expressing database queries, Vega is a high-level language for describing visualizations.



Vega is a *Visualization Grammar*

Vega Altair Python API

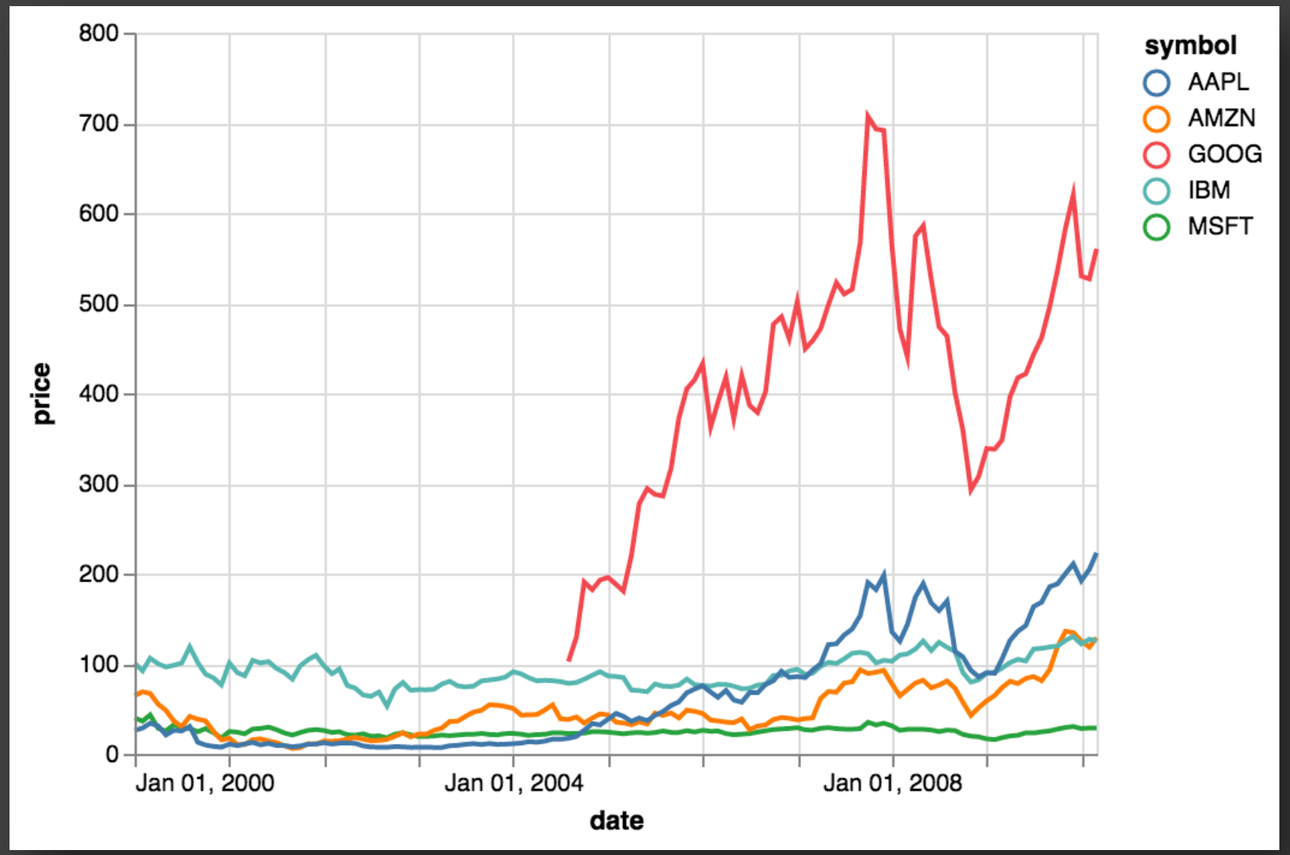
```
alt.Chart('stocks.csv')
  .mark_line()
  .encode(
    x='date',
    y='price',
    color='symbol'
  )
```



Vega is a *Visualization Grammar*

Vega-Lite JavaScript API

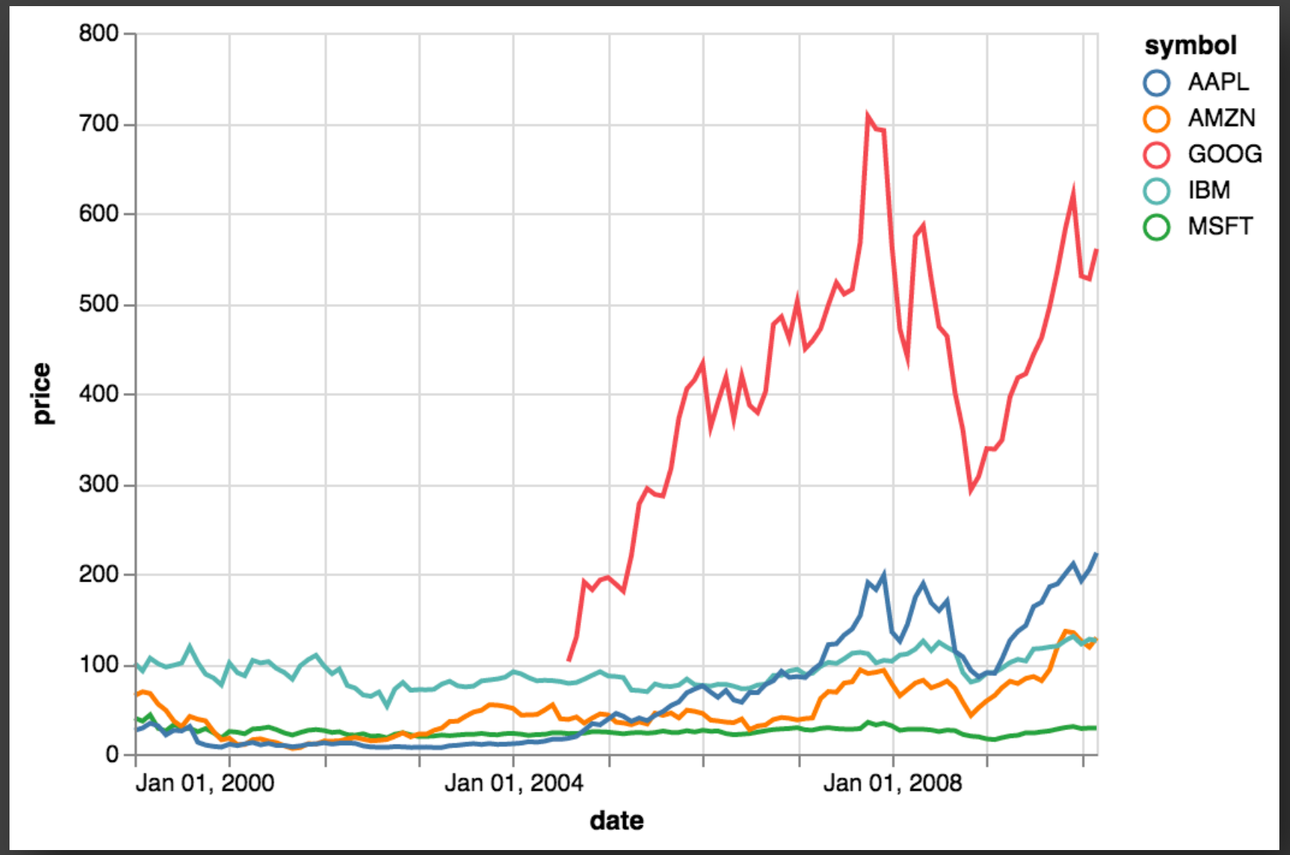
```
vl.data('stocks.csv')
  .markLine()
  .encode(
    vl.x().fieldT('date'),
    vl.y().fieldQ('price'),
    vl.color().fieldN('symbol')
  )
```



Vega is a *Visualization Grammar*

Vega-Lite JSON Specification

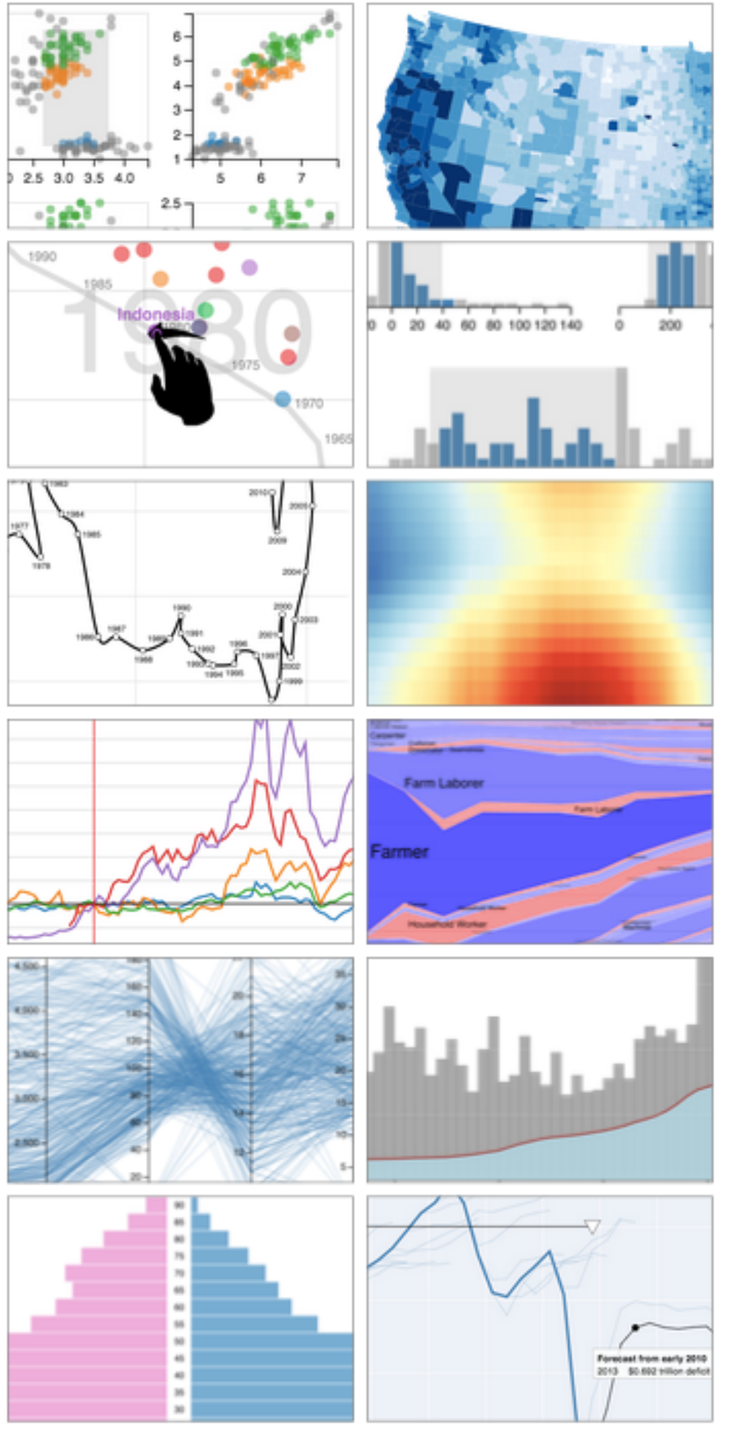
```
{
  data: {url: "stocks.csv"},
  mark: "line",
  encoding: {
    x: {
      type: "temporal",
      field: "date"
    },
    y: {
      type: "quantitative",
      field: "price"
    },
    color: {
      type: "nominal",
      field: "symbol"
    }
  }
}
```



Vega is a *Visualization Grammar*

Similar in spirit to how SQL provides a language for expressing database queries, Vega is a high-level language for describing visualizations.

Vega provides a formal model for enumerating and reasoning about visualization designs.

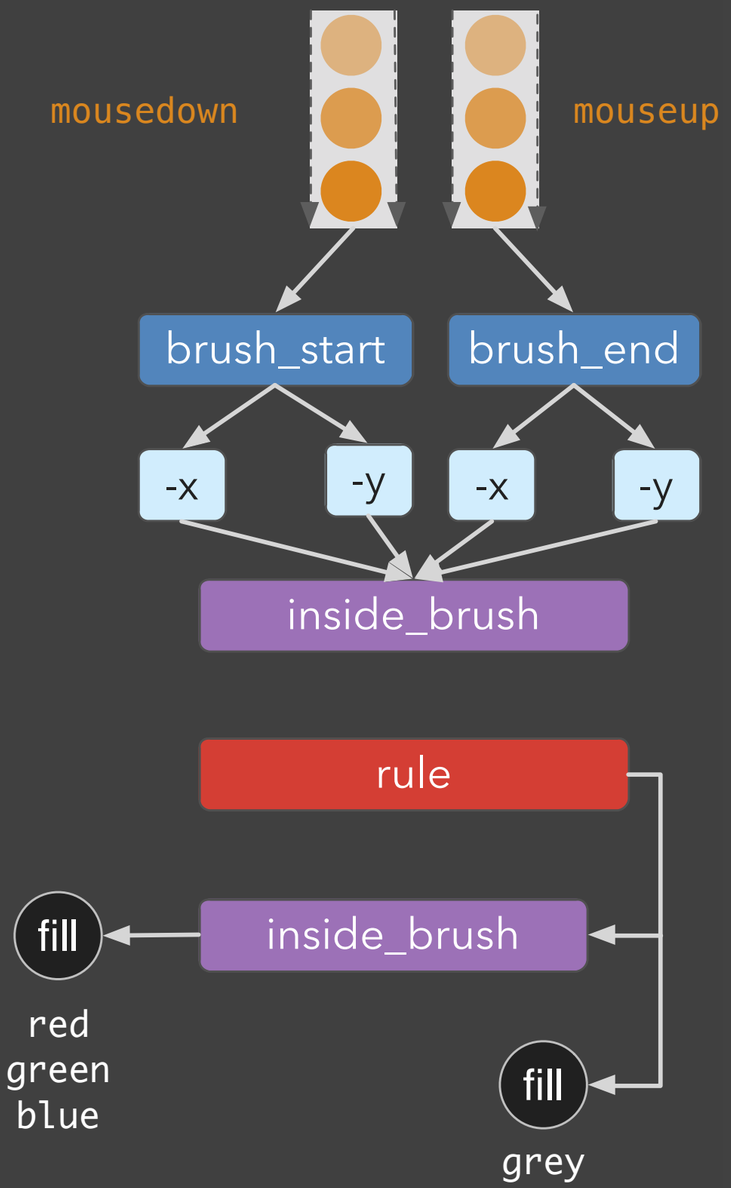


Vega is a *Visualization Grammar*

Similar in spirit to how SQL provides a language for expressing database queries, Vega is a high-level language for describing visualizations.

Vega provides a formal model for enumerating and reasoning about visualization designs.

First comprehensive approach for declarative specification of interaction techniques.



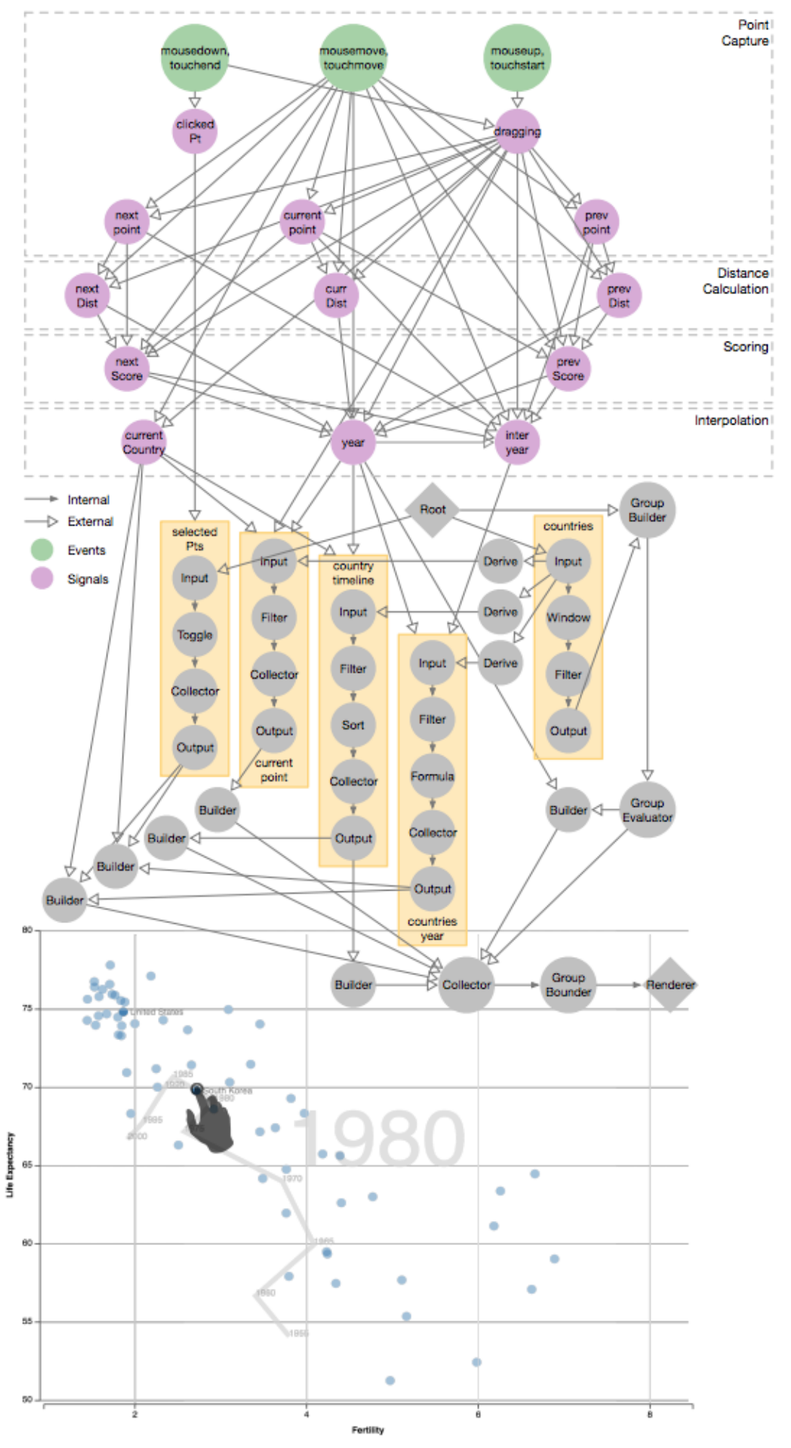
Vega is a *Visualization Grammar*

Similar in spirit to how SQL provides a language for expressing database queries, Vega is a high-level language for describing visualizations.

Vega provides a formal model for enumerating and reasoning about visualization designs.

First comprehensive approach for declarative specification of interaction techniques.

Compiles JSON description to a reactive dataflow graph with efficient, incremental processing.



Vega is a *Visualization Grammar*

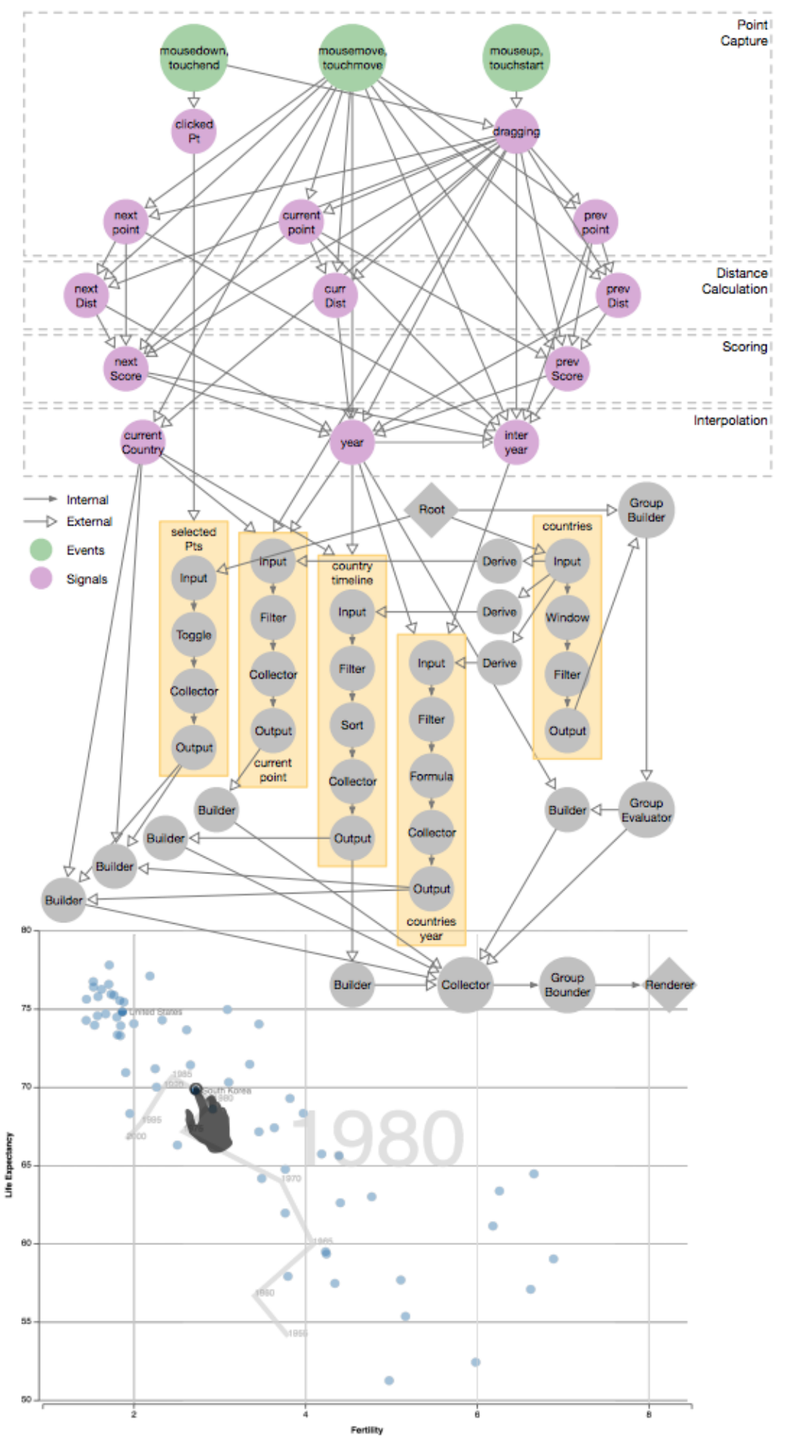
Similar in spirit to how SQL provides a language for expressing database queries, Vega is a high-level language for describing visualizations.

Vega provides a formal model for enumerating and reasoning about visualization designs.

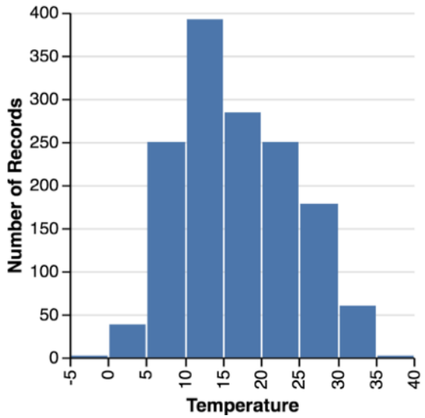
First comprehensive approach for declarative specification of interaction techniques.

Compiles JSON description to a reactive dataflow graph with efficient, incremental processing.

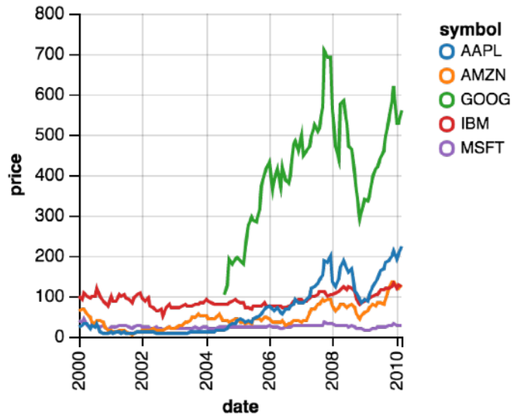
Vega's dataflow graph generates a scenegraph that is then rendered using Canvas or SVG.



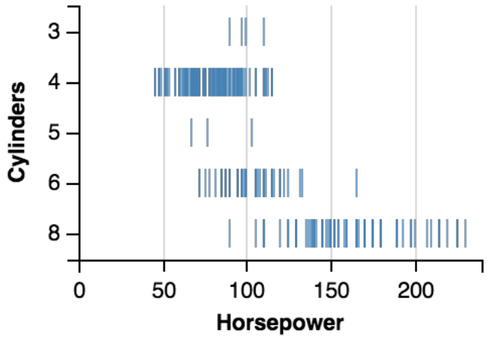
Histogram



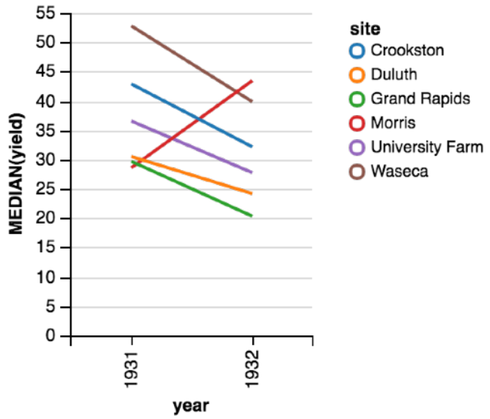
Line Chart



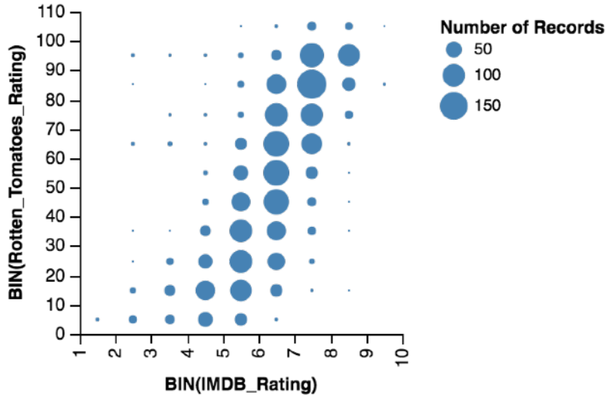
Strip Plot



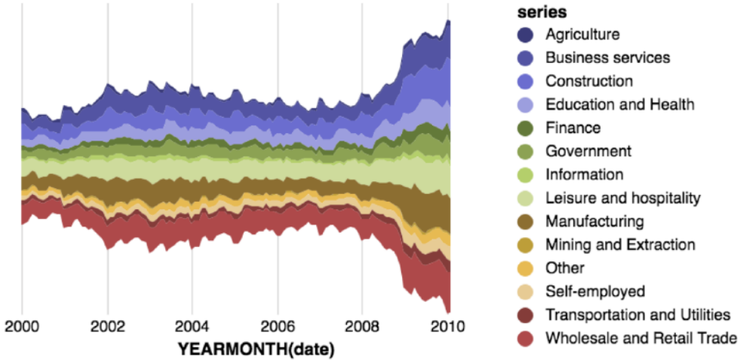
Slope Graph



Binned Scatter Plot



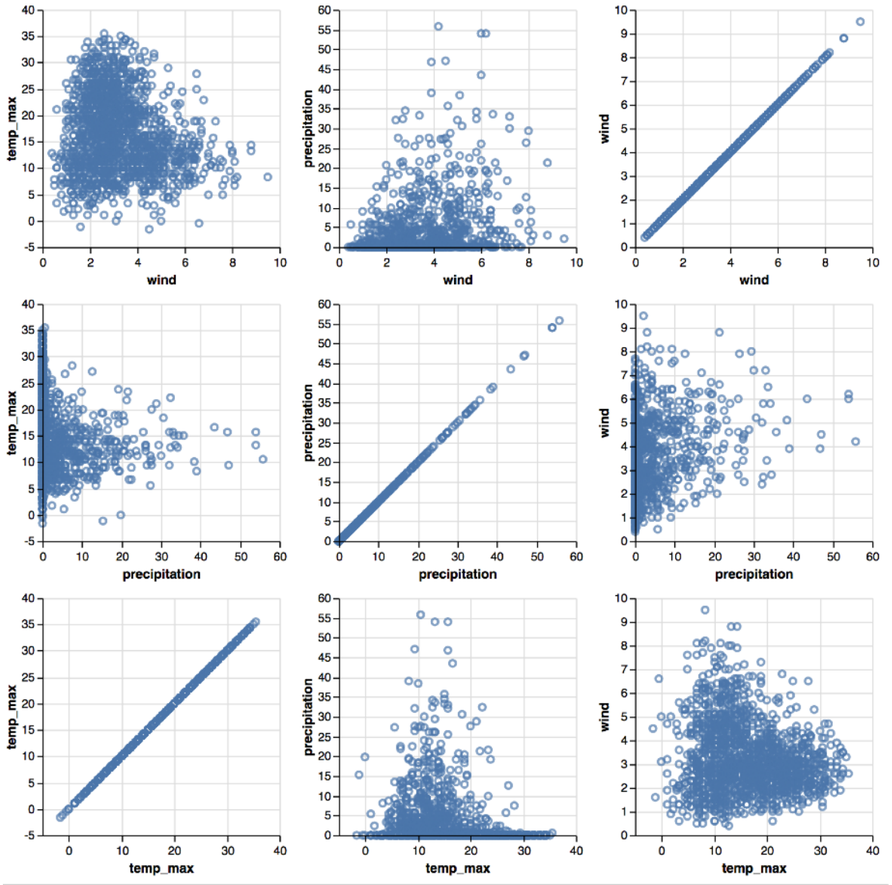
Area Chart



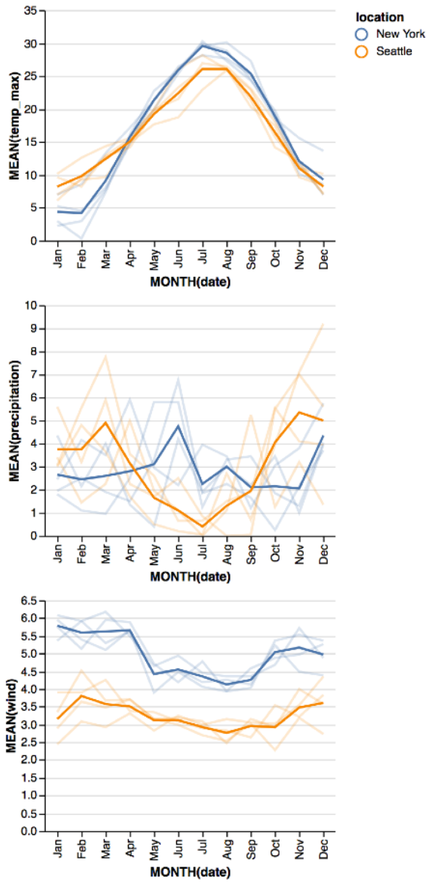
A Grammar of Graphics

[Satyanarayan et al. 2017]

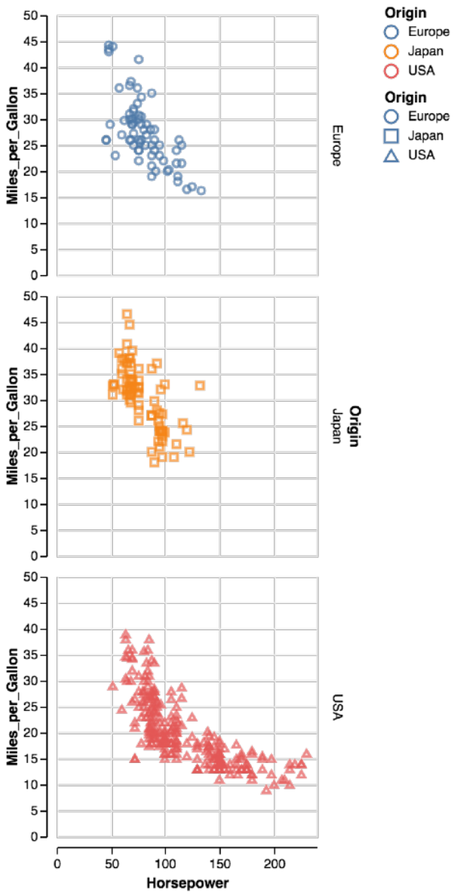
Scatter Plot Matrix



Concatenated & Layered View



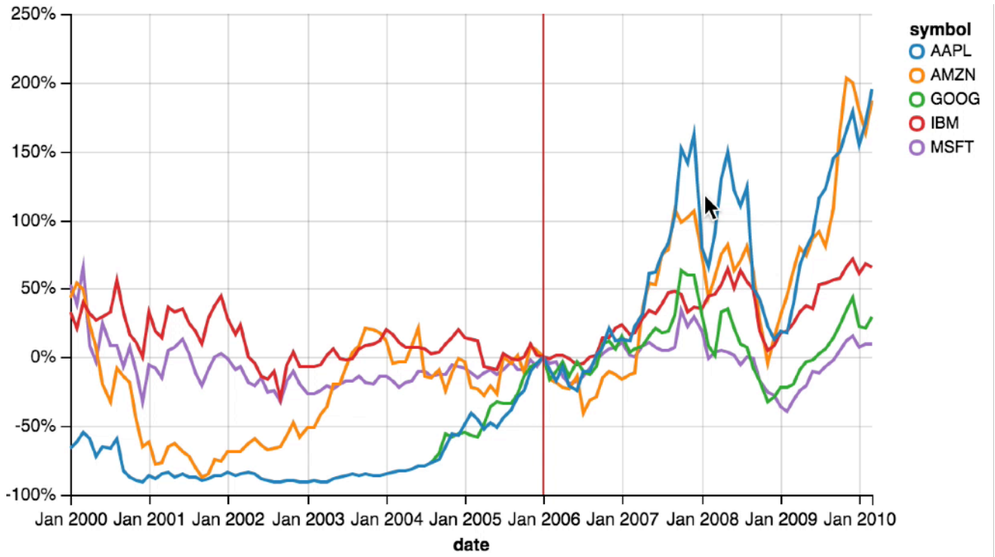
Faceted View



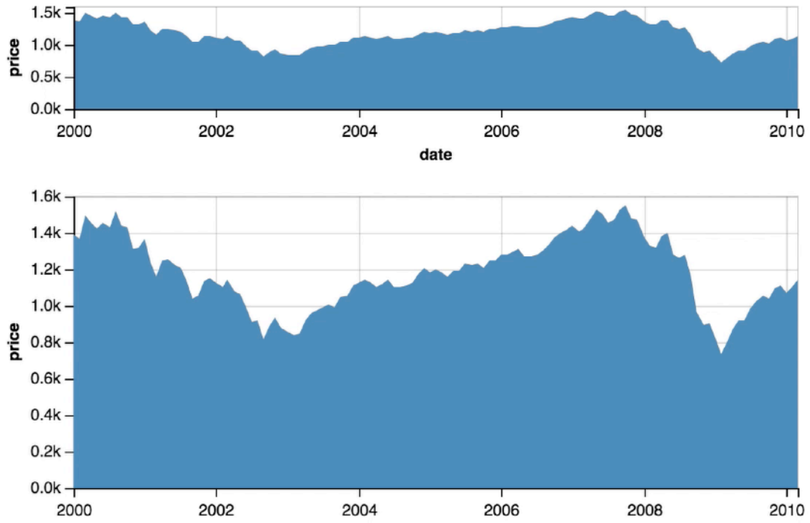
A Grammar of Multi-View Graphics

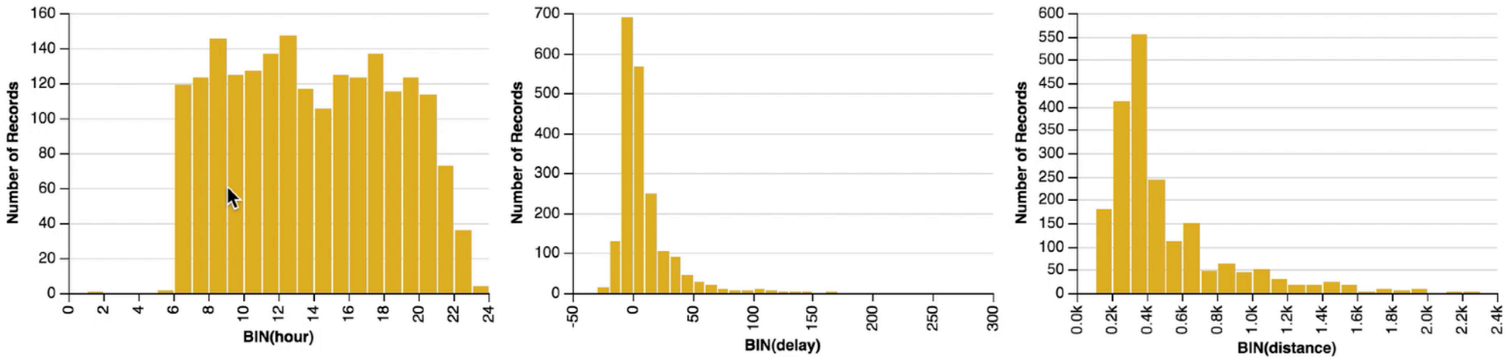
[Satyanarayan et al. 2017]

Indexed Chart



Focus + Context

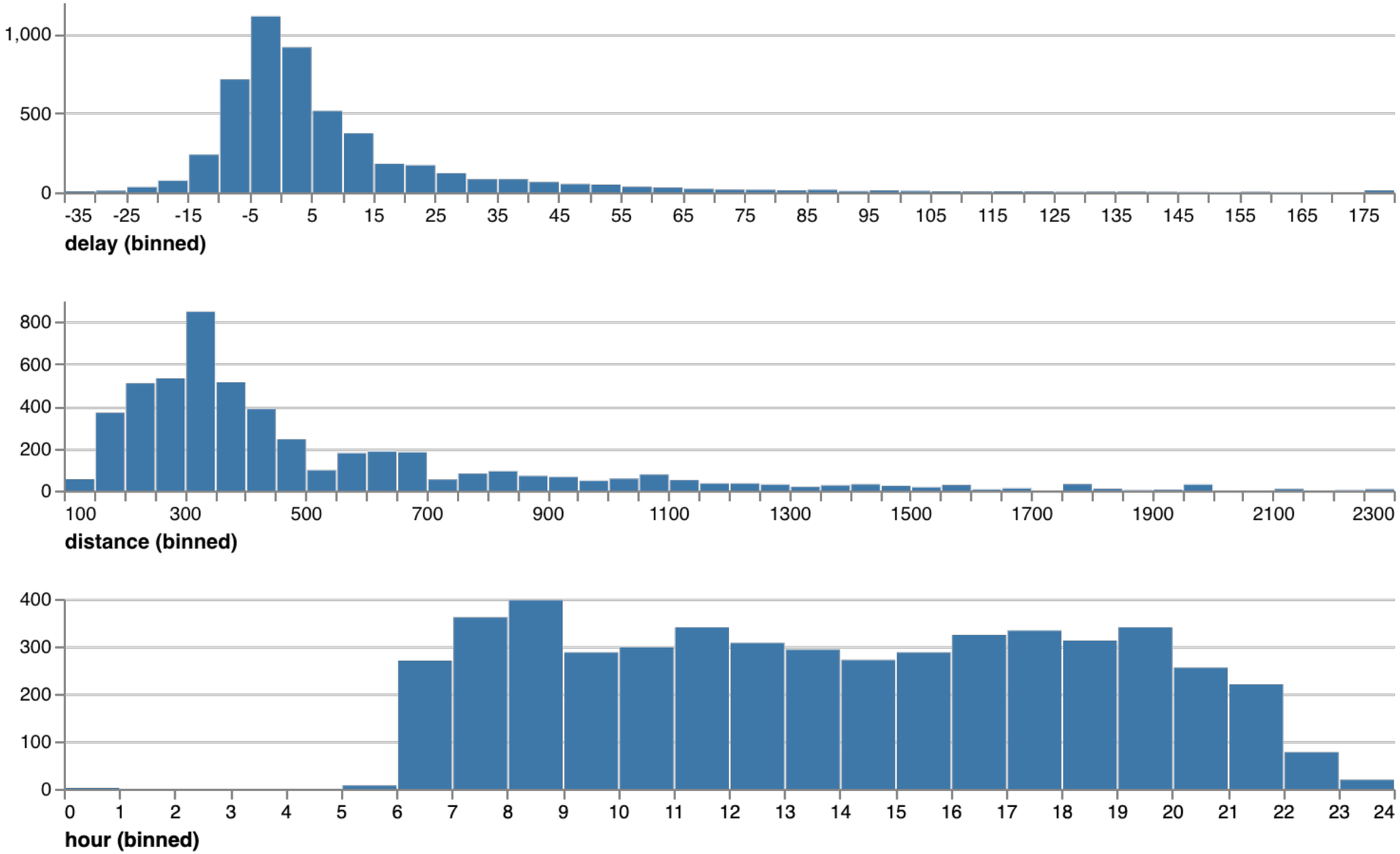




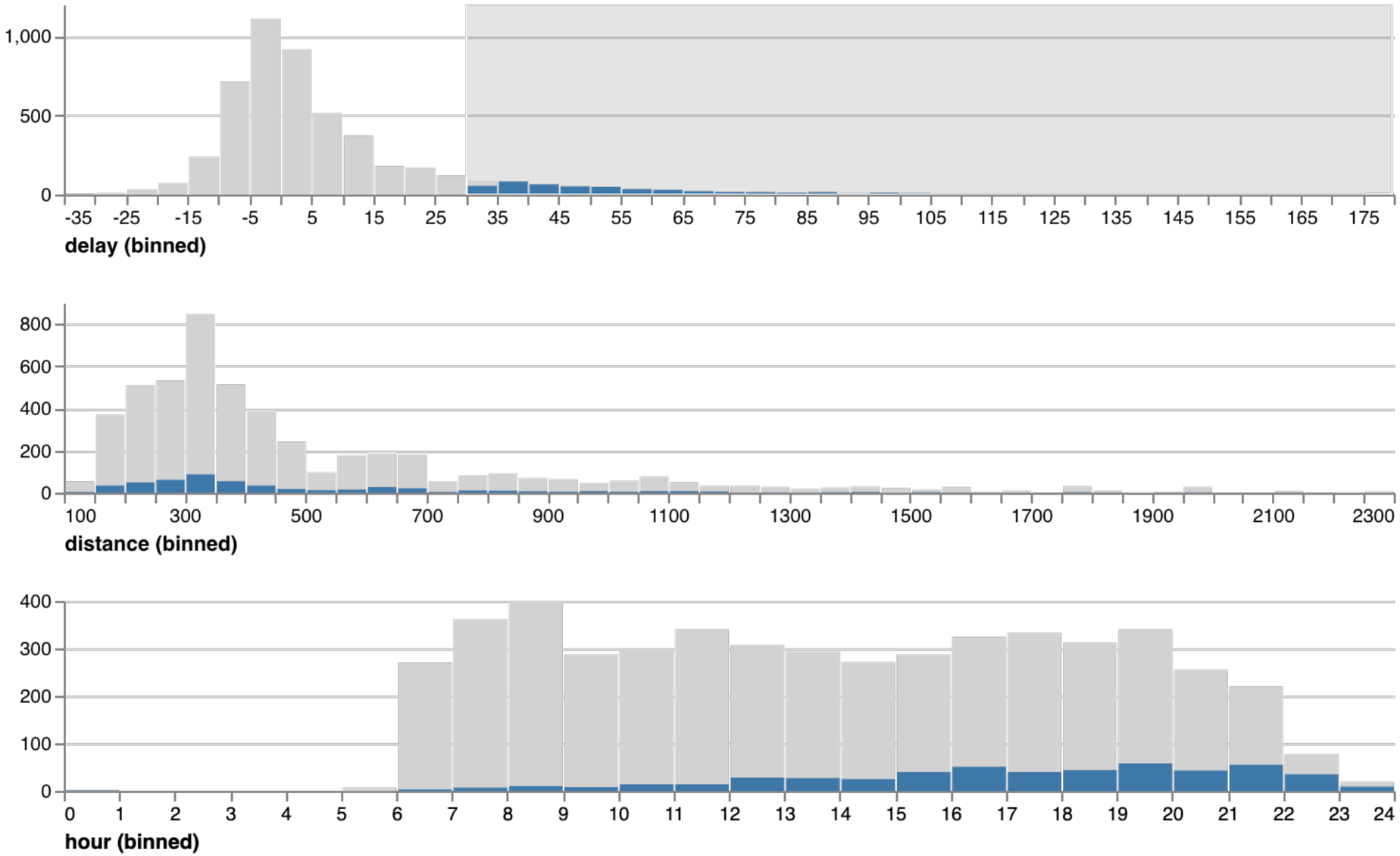
Cross-Filtering

A Grammar of Interactive Multi-View Graphics

Cross-Filtering in Vega-Lite

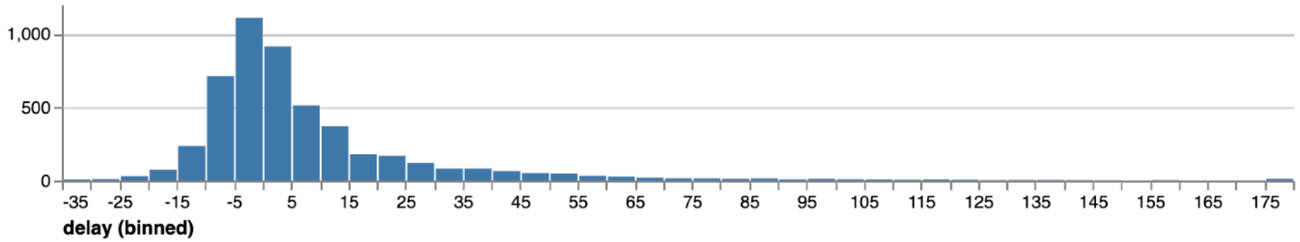


Cross-Filtering in Vega-Lite



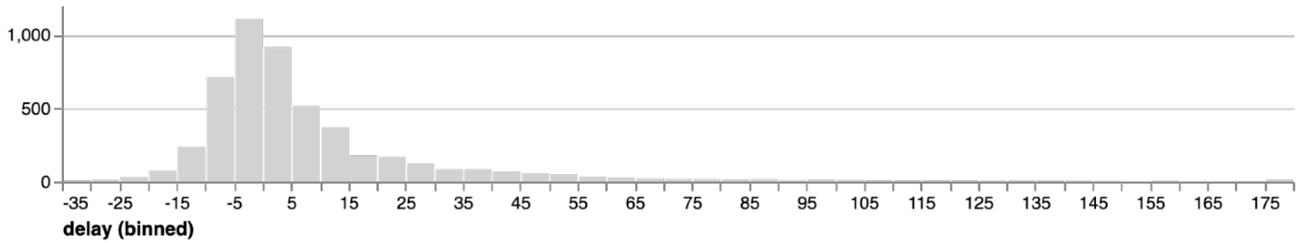
Cross-Filtering in Vega-Lite

```
markBar().encode(
  x().fieldQ('delay').bin(true),
  y().count()
).data('data/flights.json')
```



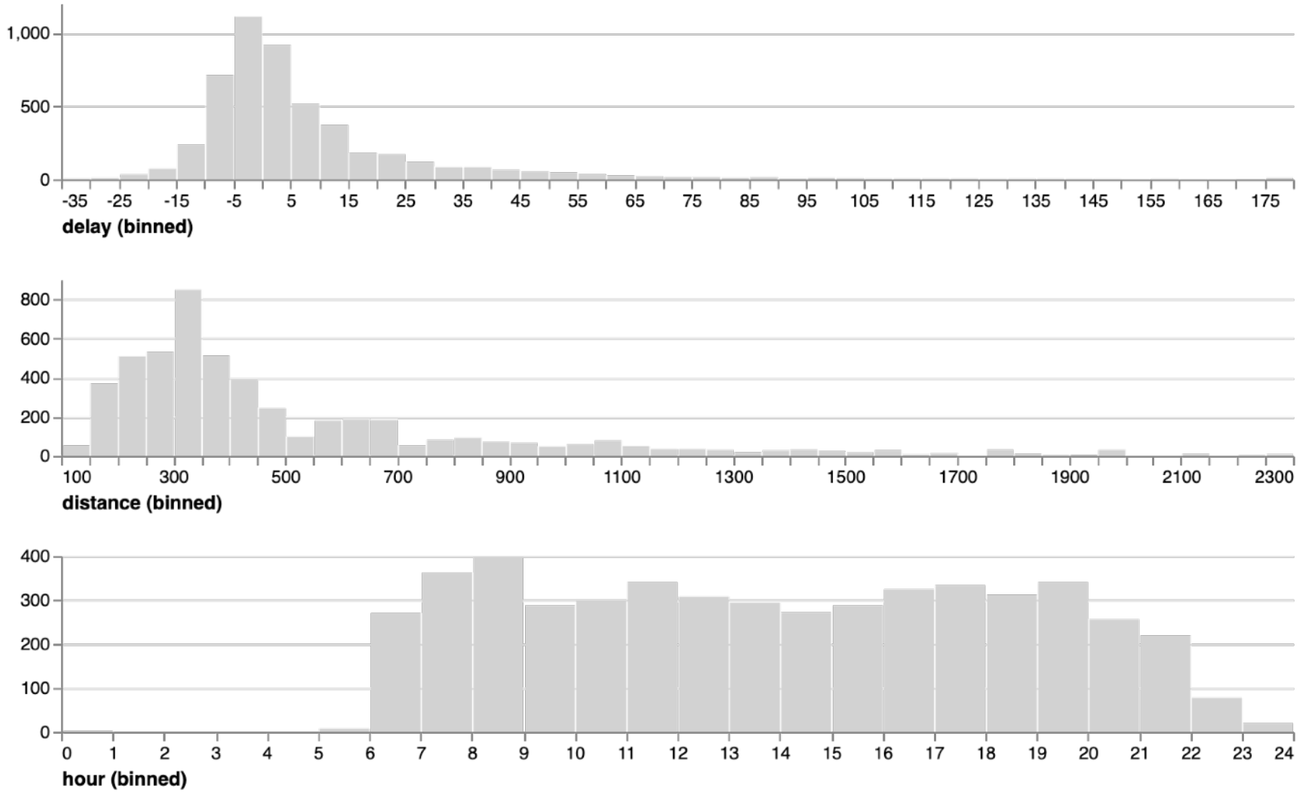
Cross-Filtering in Vega-Lite

```
markBar().encode(
  x().fieldQ('delay').bin(true),
  y().count(),
  color().value('lightgrey')
).data('data/flights.json')
```



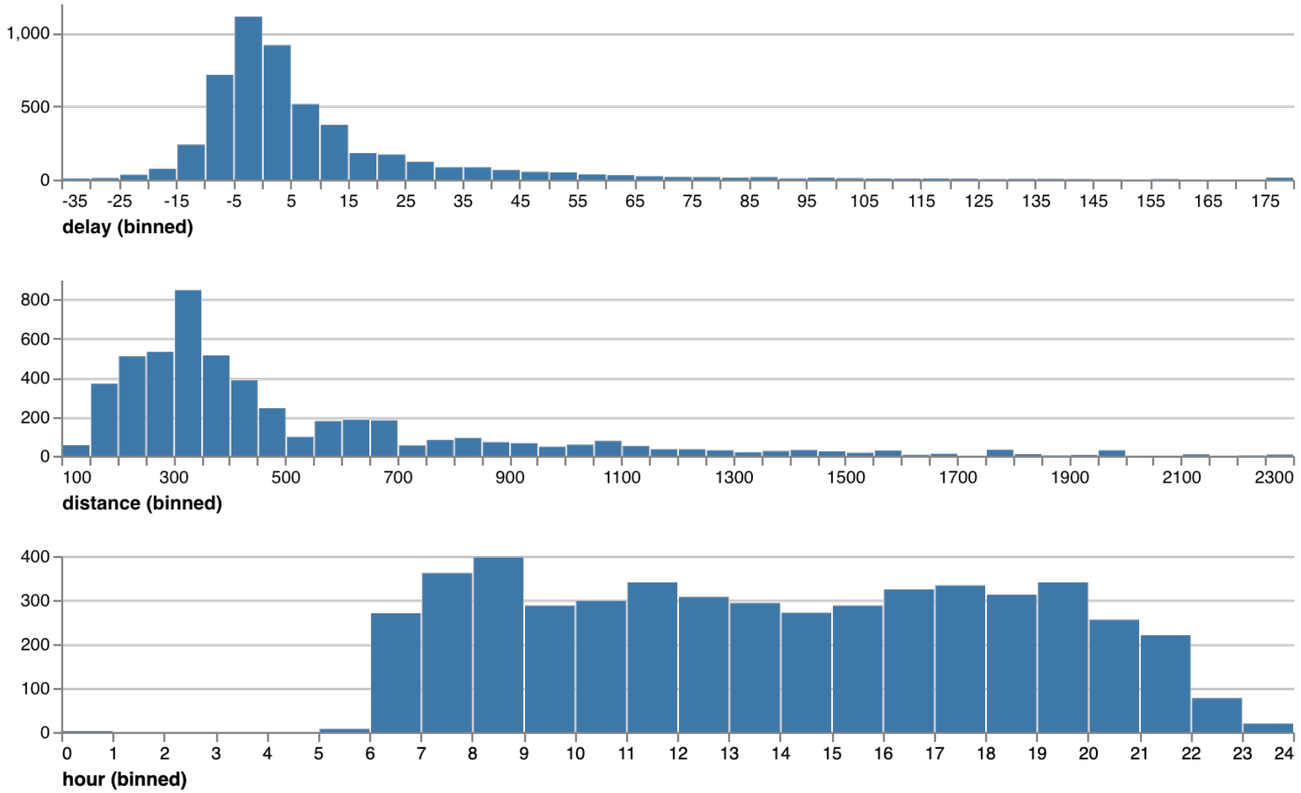
Cross-Filtering in Vega-Lite

```
markBar().encode(
  x().fieldQ(repeat('row').bin(true),
  y().count(),
  color().value('lightgrey')
)
.repeat({row: ['delay', 'distance', 'hour']})
.data('data/flights.json')
```



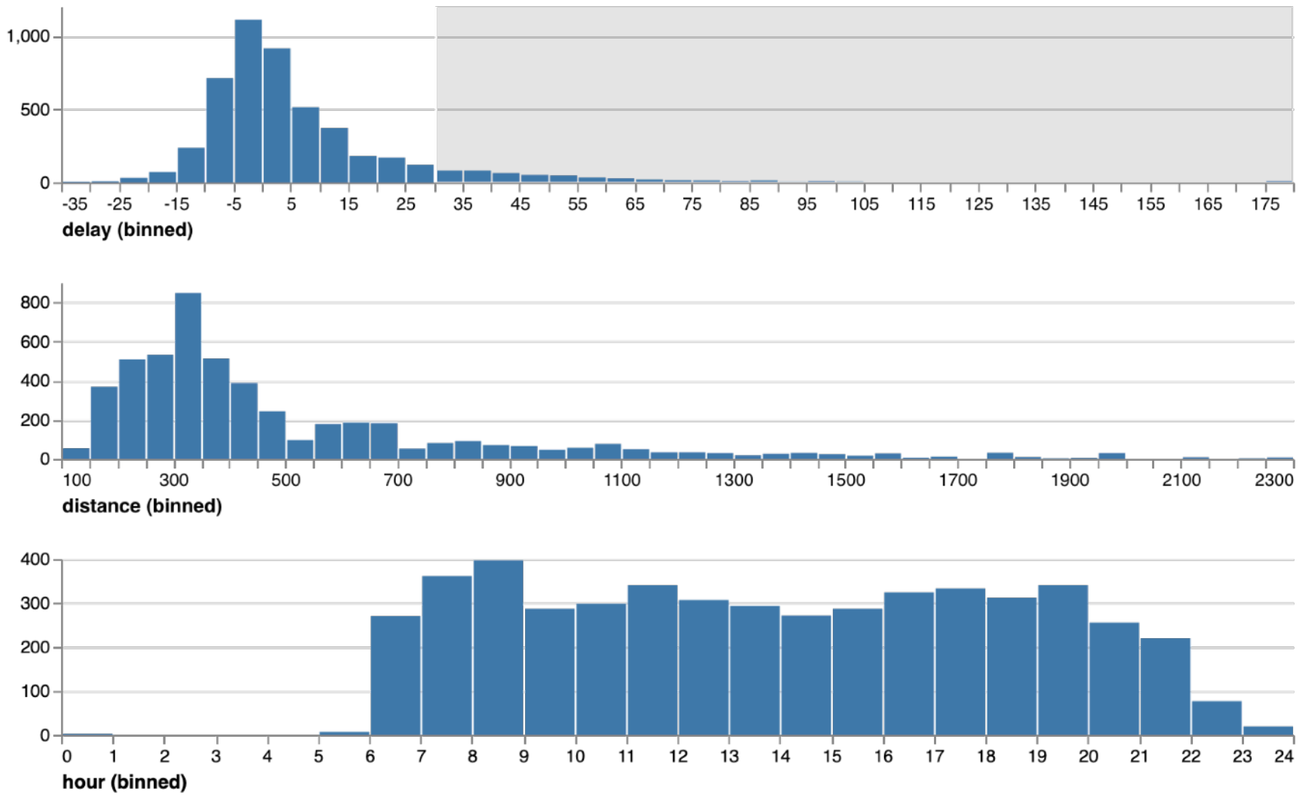
Cross-Filtering in Vega-Lite

```
layer(
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count(),
    color().value('lightgrey')
  ),
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count()
  )
)
.repeat({row: ['delay', 'distance', 'hour']})
.data('data/flights.json')
```



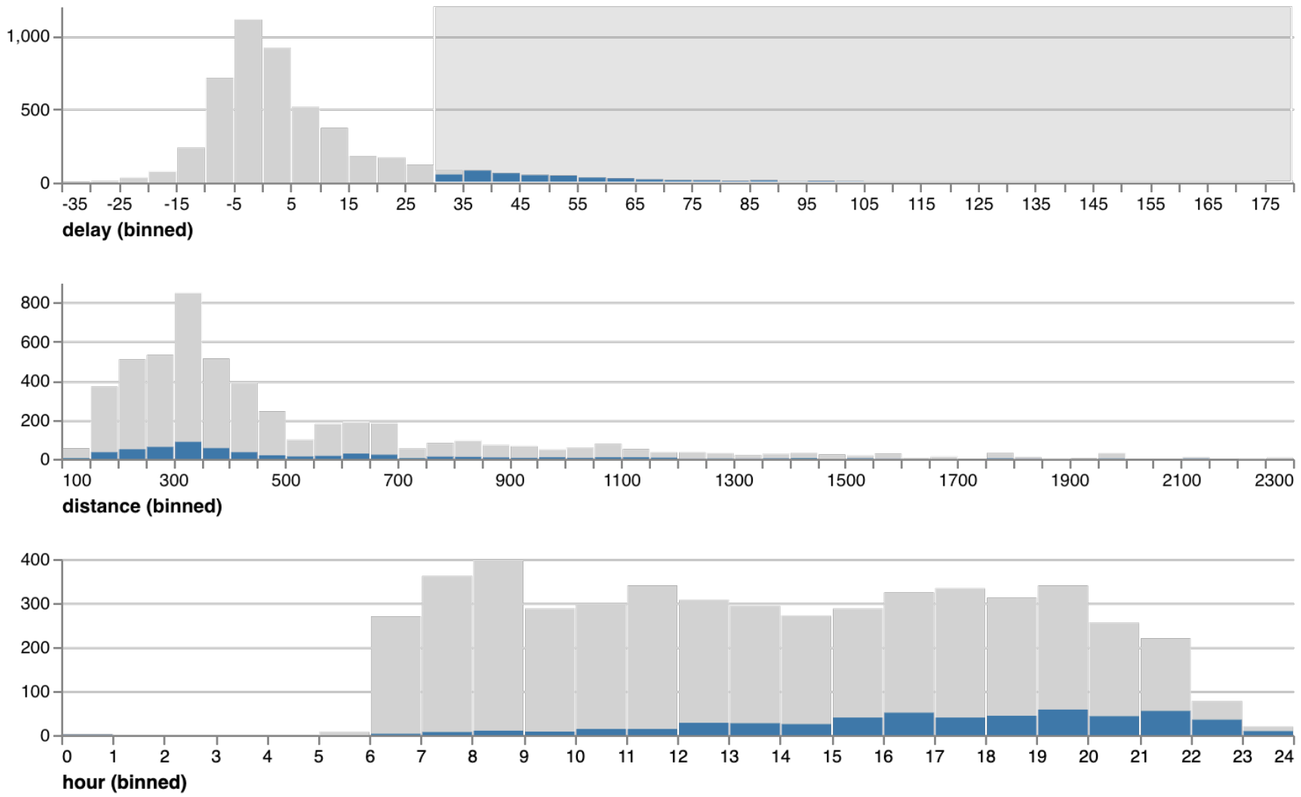
Cross-Filtering in Vega-Lite

```
brush = selectInterval().encodings('x')

layer(
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count(),
    color().value('lightgrey')
  ).select(brush),
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count()
  )
)
.repeat({row: ['delay', 'distance', 'hour']})
.data('data/flights.json')
```



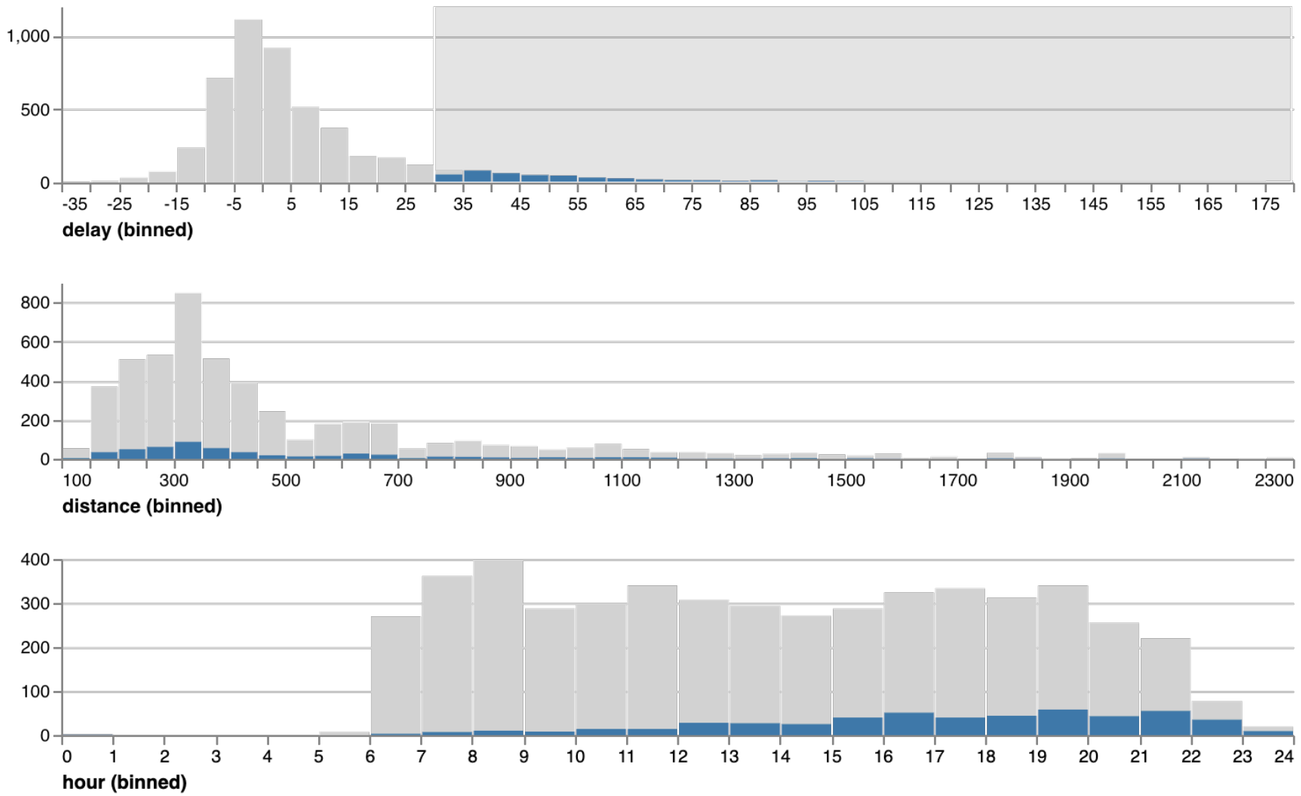
Cross-Filtering in Vega-Lite

```
brush = selectInterval.encodings('x')

layer(
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count(),
    color().value('lightgrey')
  ).select(brush),
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count()
  ).transform(filter(brush))
)
.repeat({row: ['delay', 'distance', 'hour']})
.data('data/flights.json')
```



Cross-Filtering in Vega-Lite

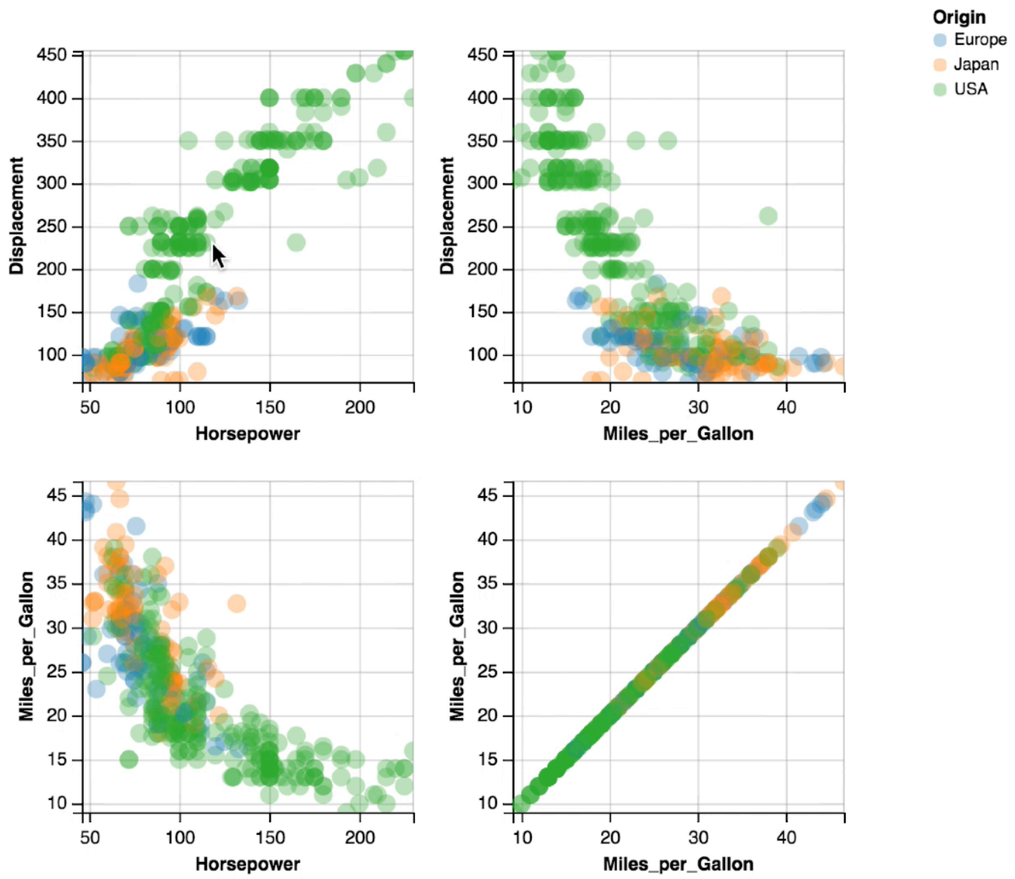
```
brush = selectInterval.encodings('x')

layer(
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count(),
    color().value('lightgrey')
  ).select(brush),
  markBar().encode(
    x().fieldQ(repeat('row')).bin(true),
    y().count()
  ).transform(filter(brush))
)
.repeat({row: ['delay', 'distance', 'hour']})
.data('data/flights.json')
```



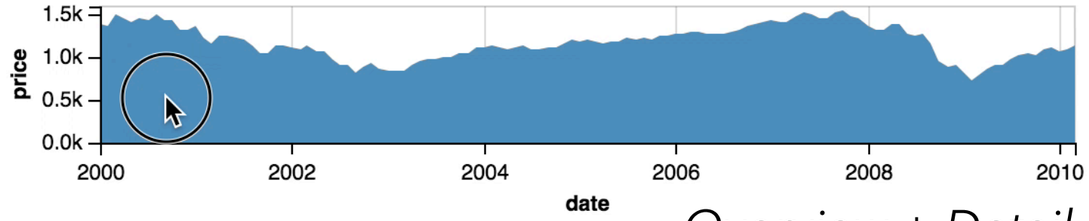
Multi-view interactive graphics in ~10 lines of code!

Interactive Selections

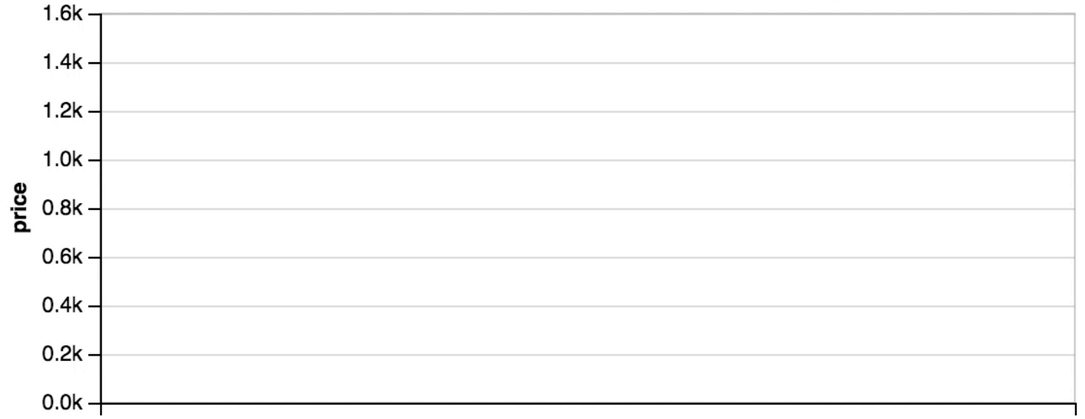
Selections *invert* scales and *parameterize* graphics

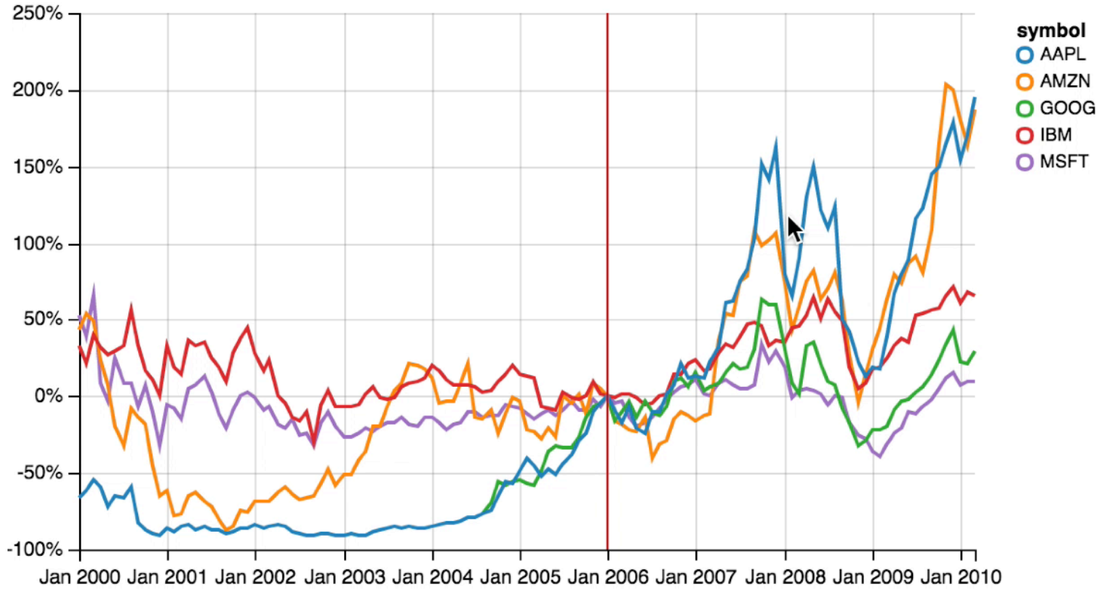


Bind selection to scale domains:
Synchronized Pan & Zoom!



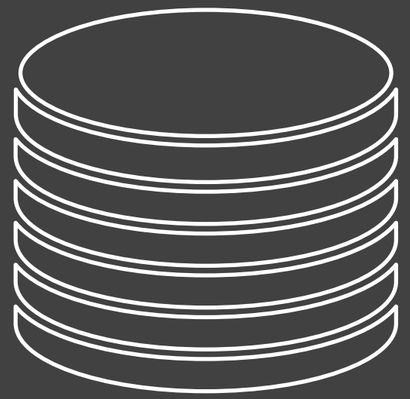
Overview + Detail



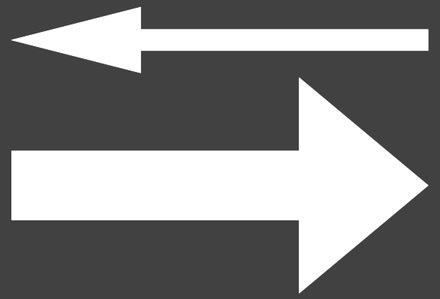


Parameterized Transformations

How might we enable **scalable, real-time visual queries?**



Database



Latency?

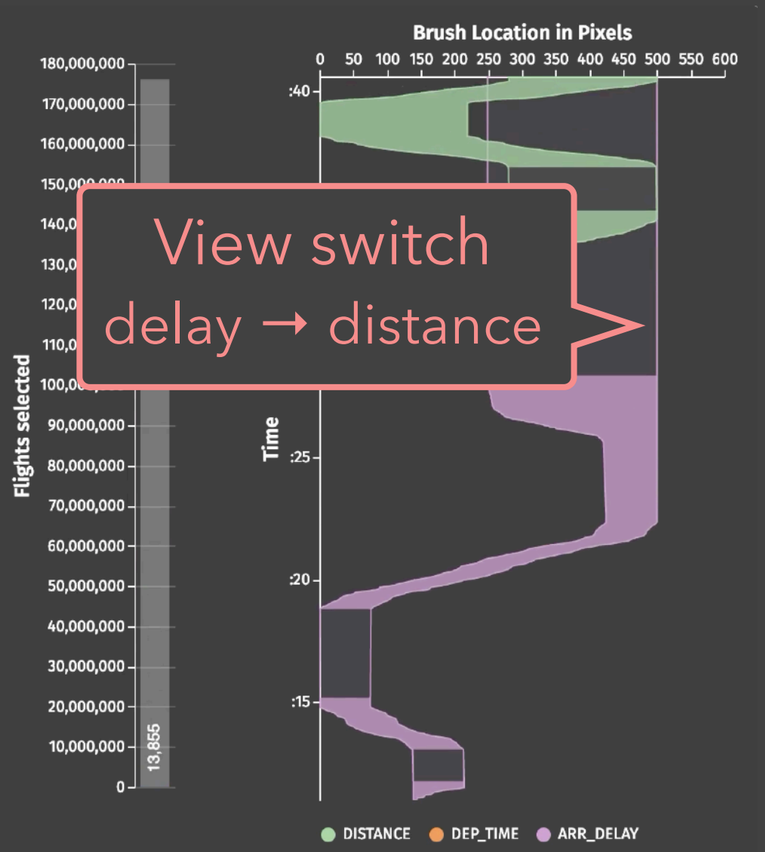
Vega

JavaScript

Milliseconds matter for latency-sensitive interactions like brushing!

[Liu & Heer 2014, Zgraggen et al. 2017]

Cross-Filtering Interaction Log



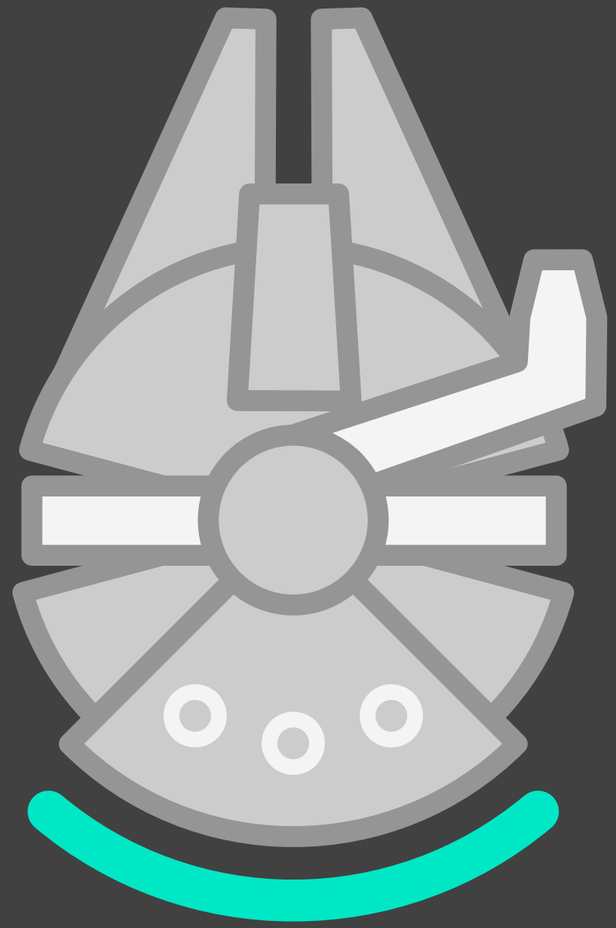
View switch
delay → distance

5x speedup

Brushing interactions

👁️ Brushing is more common and people are sensitive to latencies.

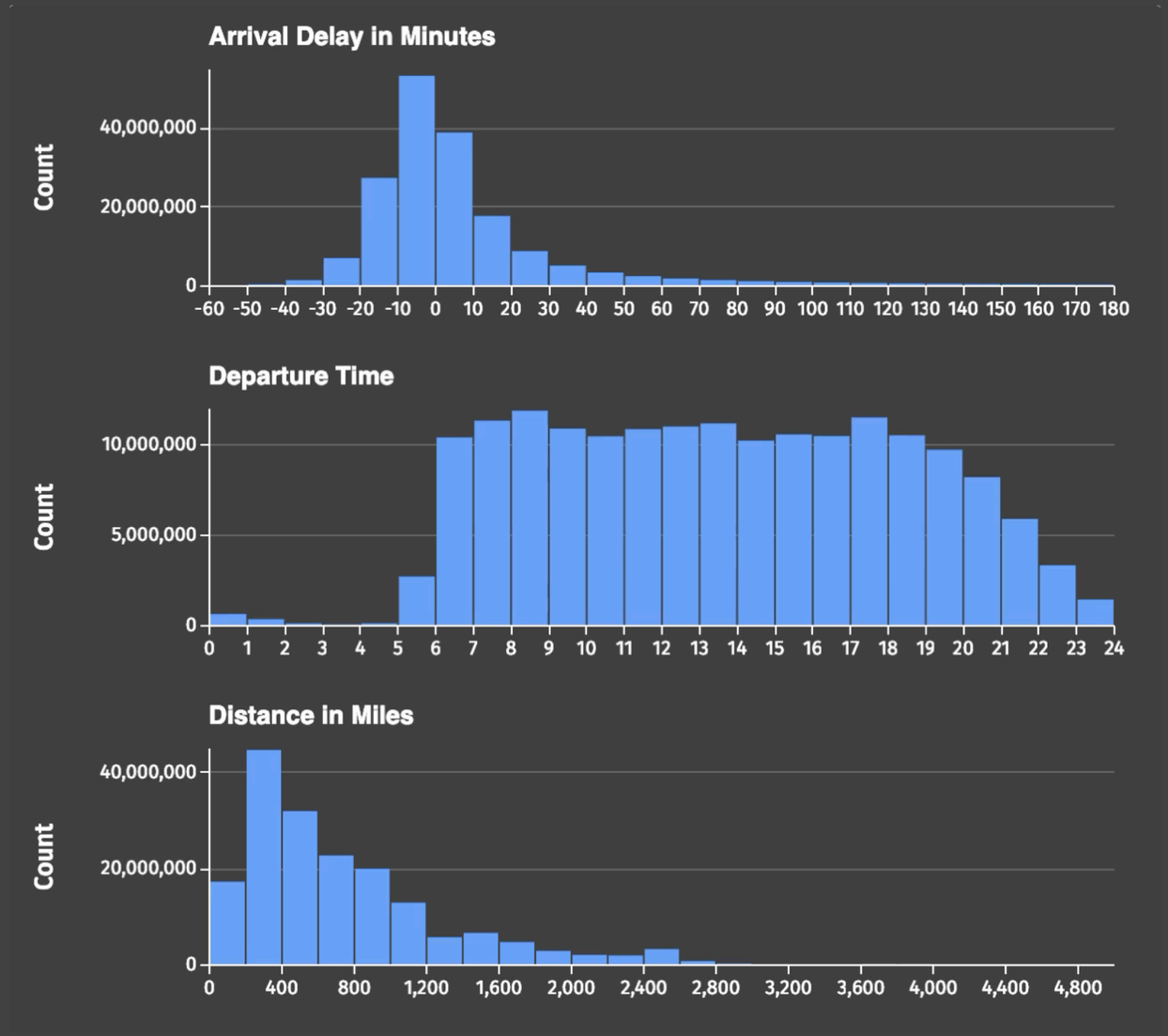
💡 Prioritize brushing latency over view switching latency.



Falcon

uwdata.github.io/falcon

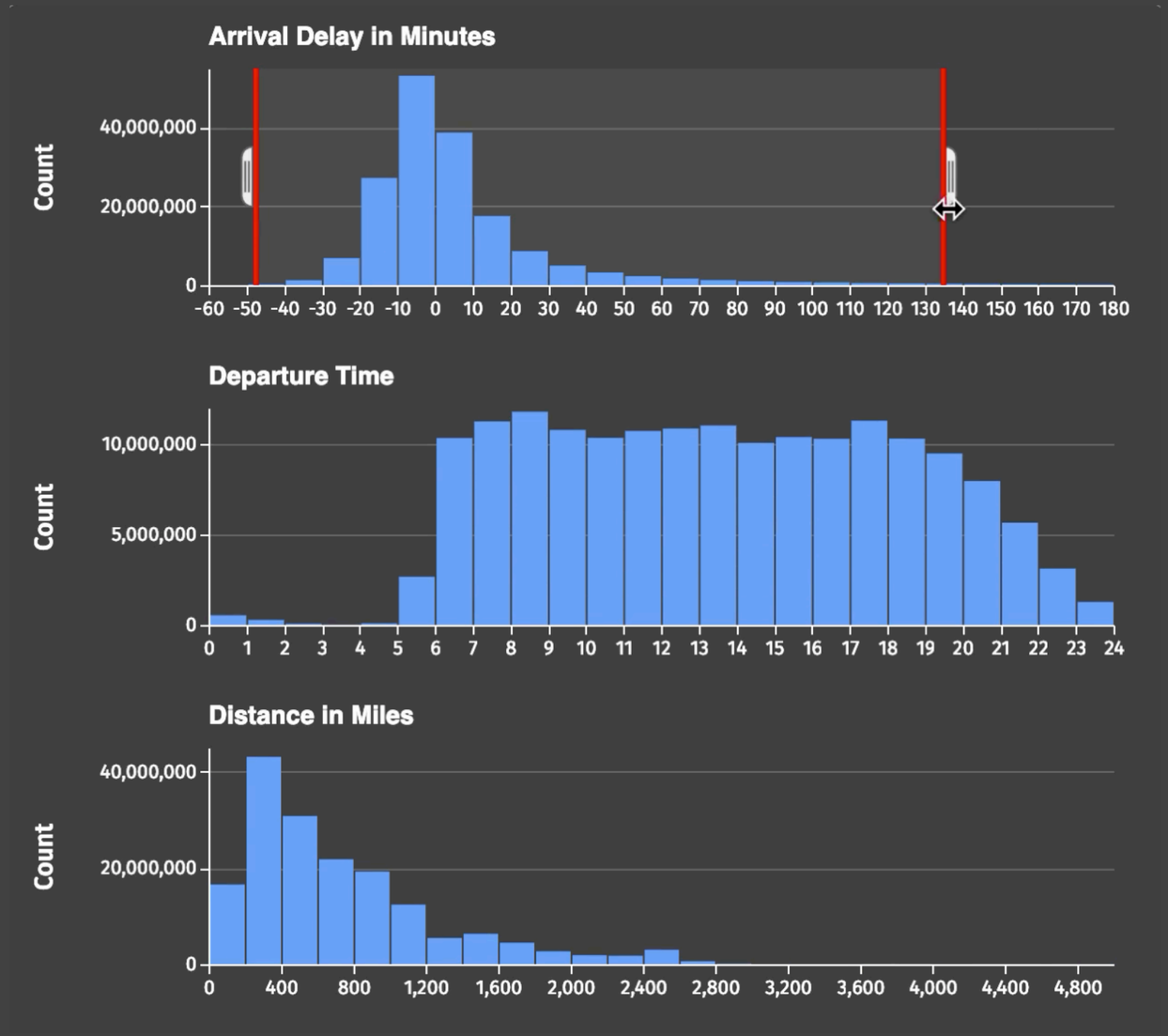
[Moritz et al., CHI 2019]



brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997.*



brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997.*

interacts with a new view

query for new data cubes

Constant data & time.
Client only.

brushes in the precomputed view

serves requests from a data cube

Data Cube. Gray et al. *1997.*

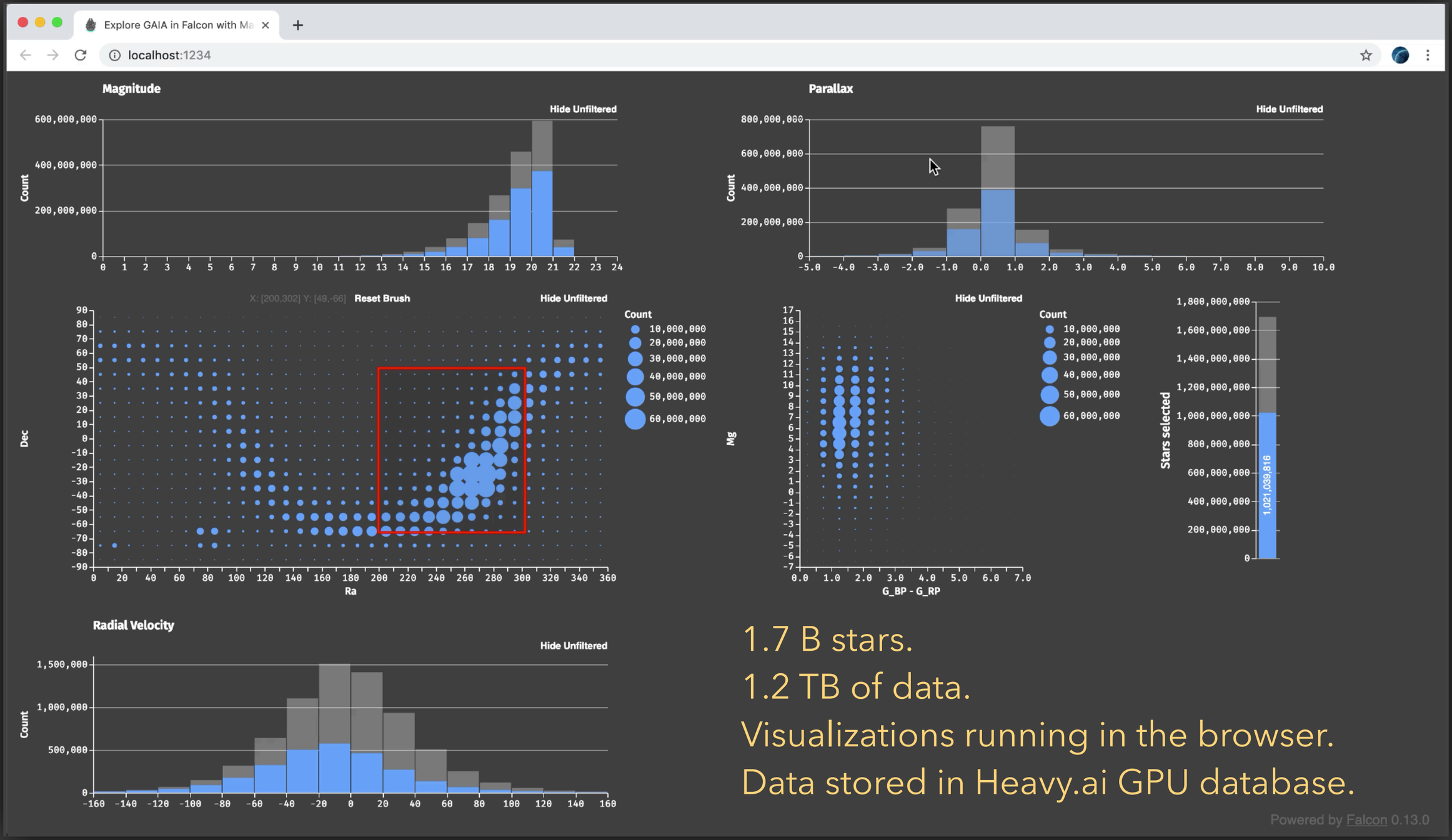
💡 Aggregation decouples interactions from queries over the raw data.

Requires one pass
over the data.

interacts with a new view

query for new data cubes

💡 View switches are **rare** and users are **not as latency sensitive** with them.


Explore GAIA in Falcon with Ma
localhost:1234
Magnitude
Hide Unfiltered
Count
600,000,000
400,000,000
200,000,000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
Parallax
Hide Unfiltered
Count
800,000,000
600,000,000
400,000,000
200,000,000
0
-5.0 -4.0 -3.0 -2.0 -1.0 0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
X: [200,302] Y: [49,-66]
Reset Brush
Hide Unfiltered
Count
10,000,000
20,000,000
30,000,000
40,000,000
50,000,000
60,000,000
Dec
90 80 70 60 50 40 30 20 10 0 -10 -20 -30 -40 -50 -60 -70 -80 -90
0 20 40 60 80 100 120 140 160 180 200 220 240 260 280 300 320 340 360
Ra
Hide Unfiltered
Count
10,000,000
20,000,000
30,000,000
40,000,000
50,000,000
60,000,000
Mg
17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0 -1 -2 -3 -4 -5 -6 -7
0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0
G_BP - G_RP
Stars selected
1,800,000,000
1,600,000,000
1,400,000,000
1,200,000,000
1,000,000,000
800,000,000
600,000,000
400,000,000
200,000,000
0
1,021,039,816
Radial Velocity
Hide Unfiltered
Count
1,500,000
1,000,000
500,000
0
-160 -140 -120 -100 -80 -60 -40 -20 0 20 40 60 80 100 120 140 160
1.7 B stars.
1.2 TB of data.
Visualizations running in the browser.
Data stored in Heavy.ai GPU database.
Powered by Falcon 0.13.0

How might we support more **effective data exploration?**

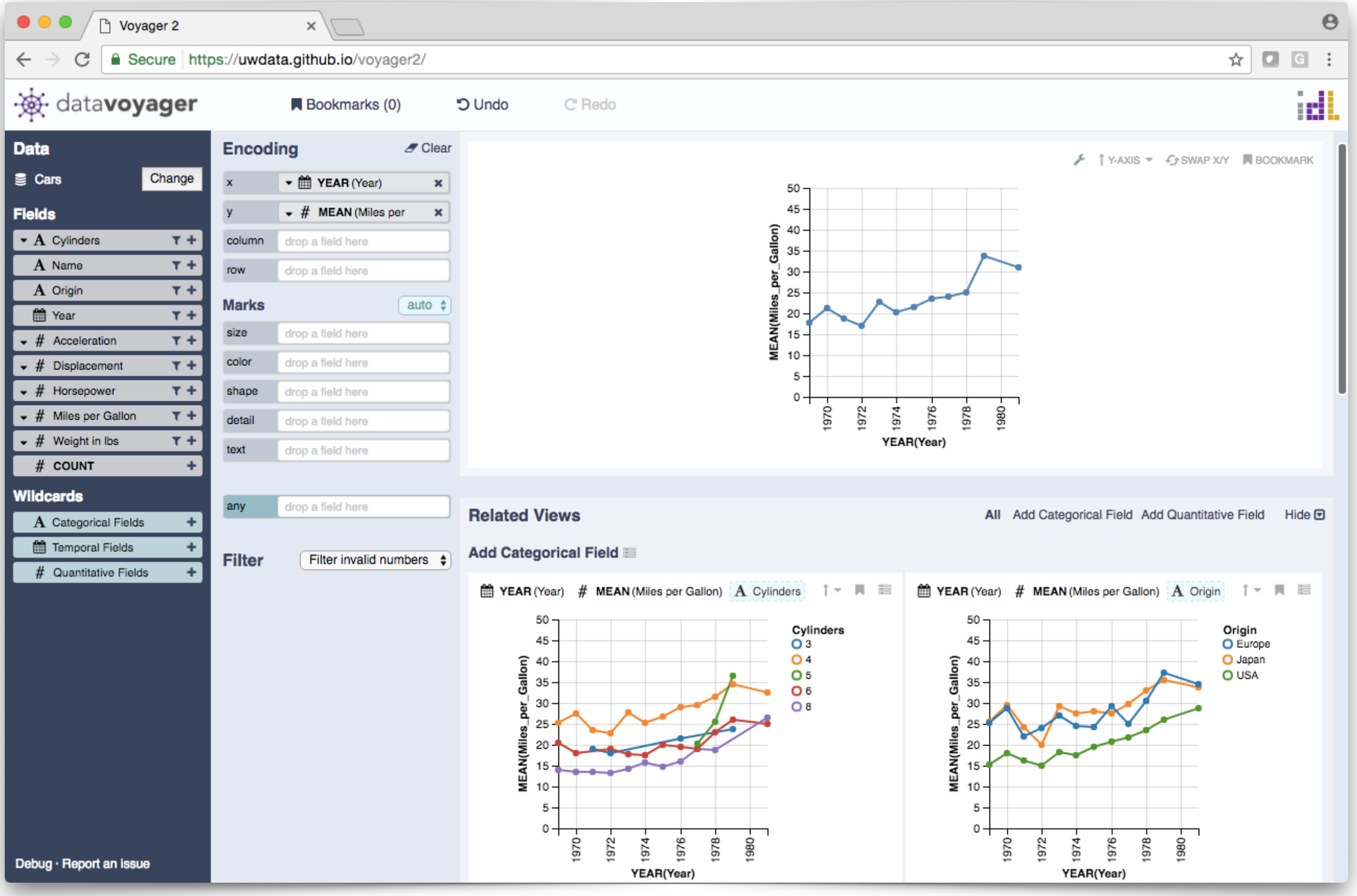
Common exploration pitfalls:

Overlook data quality issues

Fixate on specific relationships

Plus many other biases…

[Heuer 1999, Kahneman 2011, …]



Voyager: Combine Specification and Recommendation [Wongsuphasawat 2017]

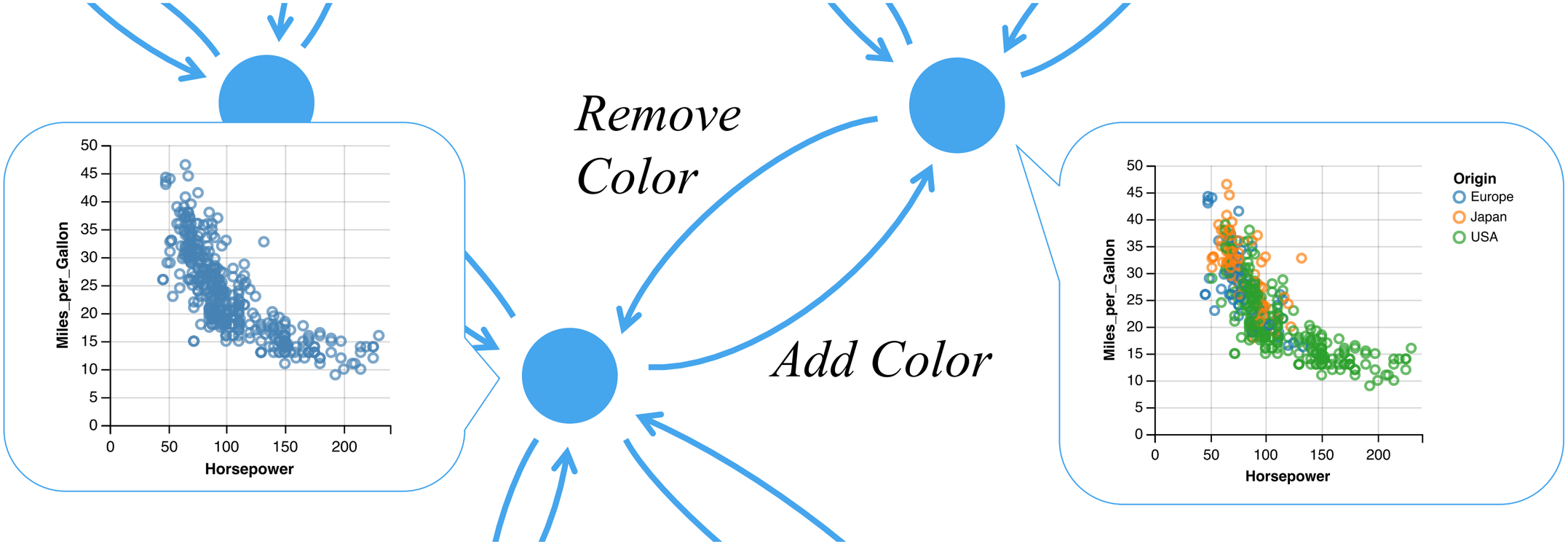
Key Idea: Augment manual exploration with visualization recommendations sensitive to the user's current focus.

The ultimate goal is to support *systematic consideration* of the data, without exacerbating *false discovery*.

To model a user's search frontier, we *optimize for related chart specifications*, seeded by the user's current focus.

Candidate charts are pruned and ranked using a formal model of *design constraints* and *perceptual effectiveness,* which can be trained from perception experiment results.

A Formal Design Space of Visualizations

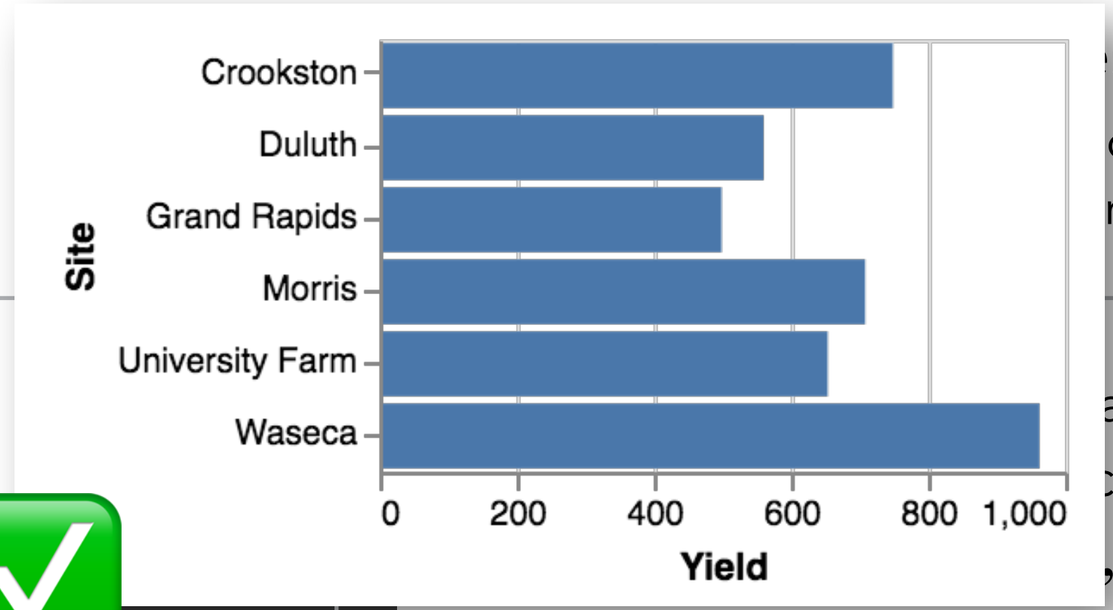


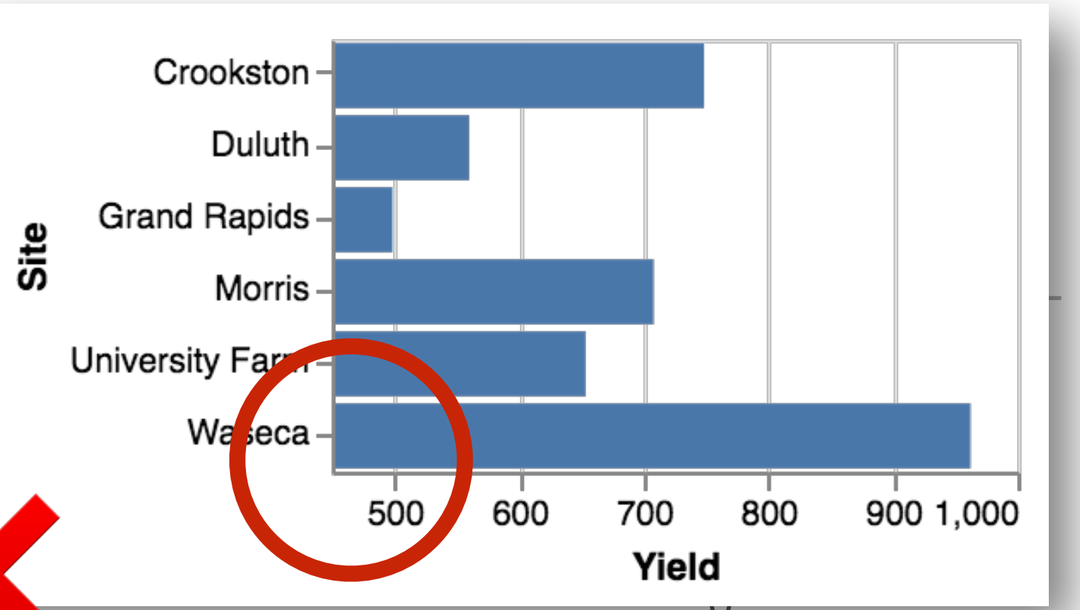
Enumerate **Vega-Lite specifications** and transformations among them.

Search the space using logic programming methods.

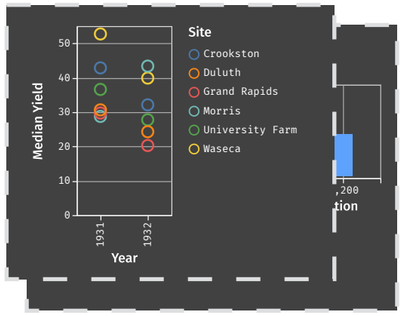
[Kim et al. 2017]

Learn Design Trade-Offs from Experimental Data





$$\arg\max_{w} \sum_{i \in 0 \ldots k} w_i \, (u_i - v_i)$$



👎 Negative example

Feature Vector

$$[v_1, v_2, \ldots, v_k]$$

v_i : the number of violations of constraint .

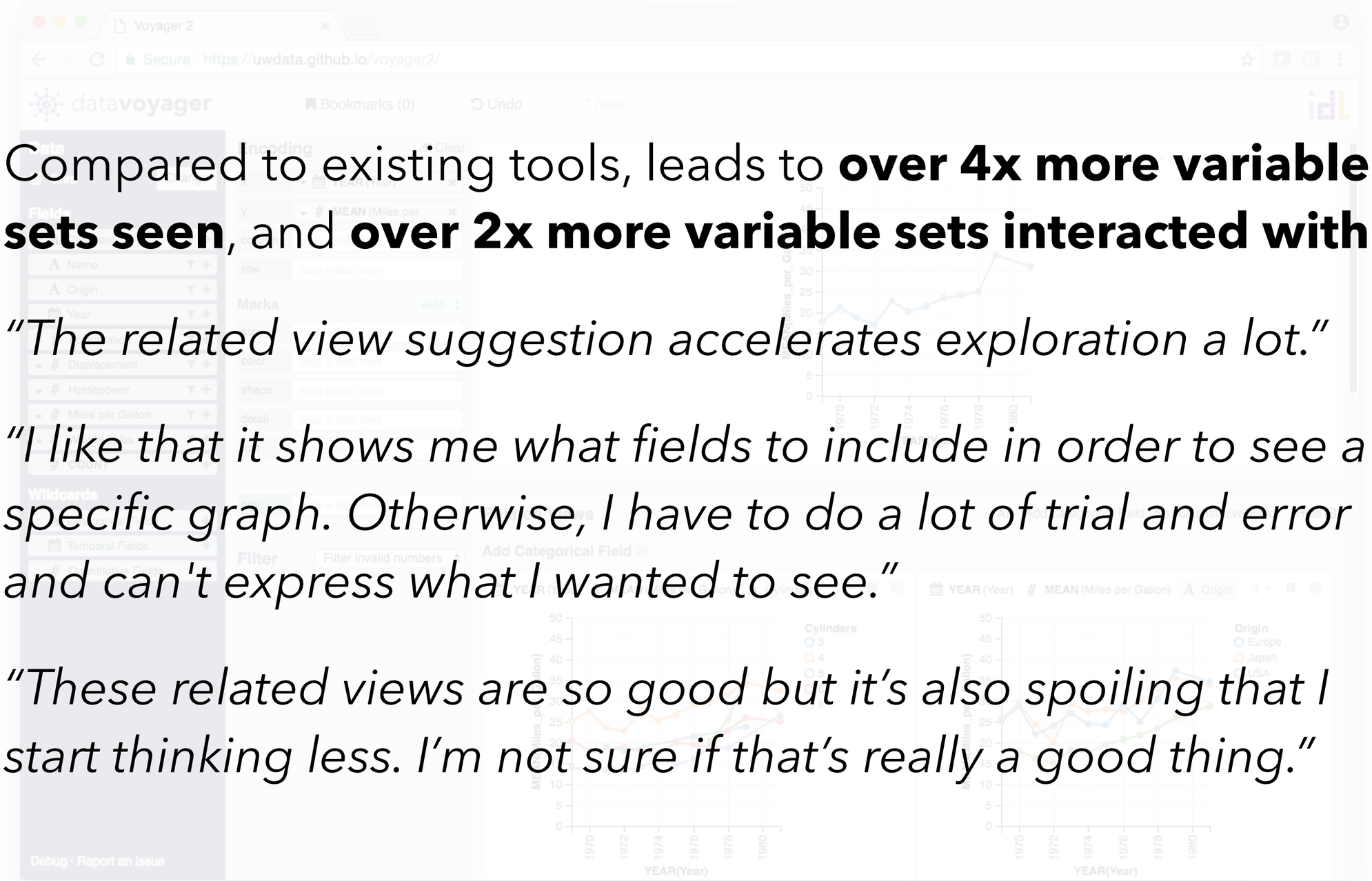
[Moritz et al. 2019]

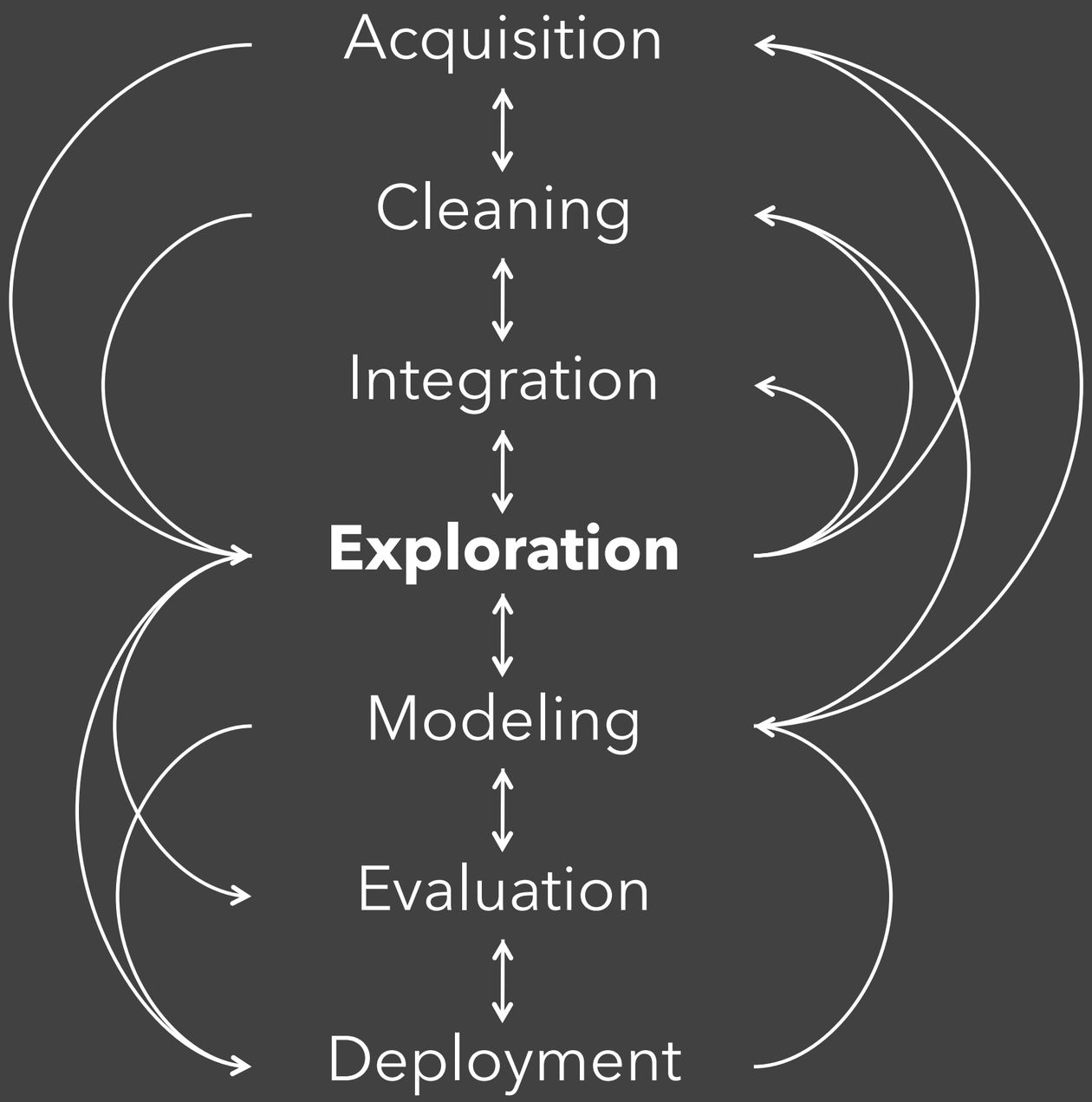
Compared to existing tools, leads to **over 4x more variable sets seen**, and **over 2x more variable sets interacted with**.

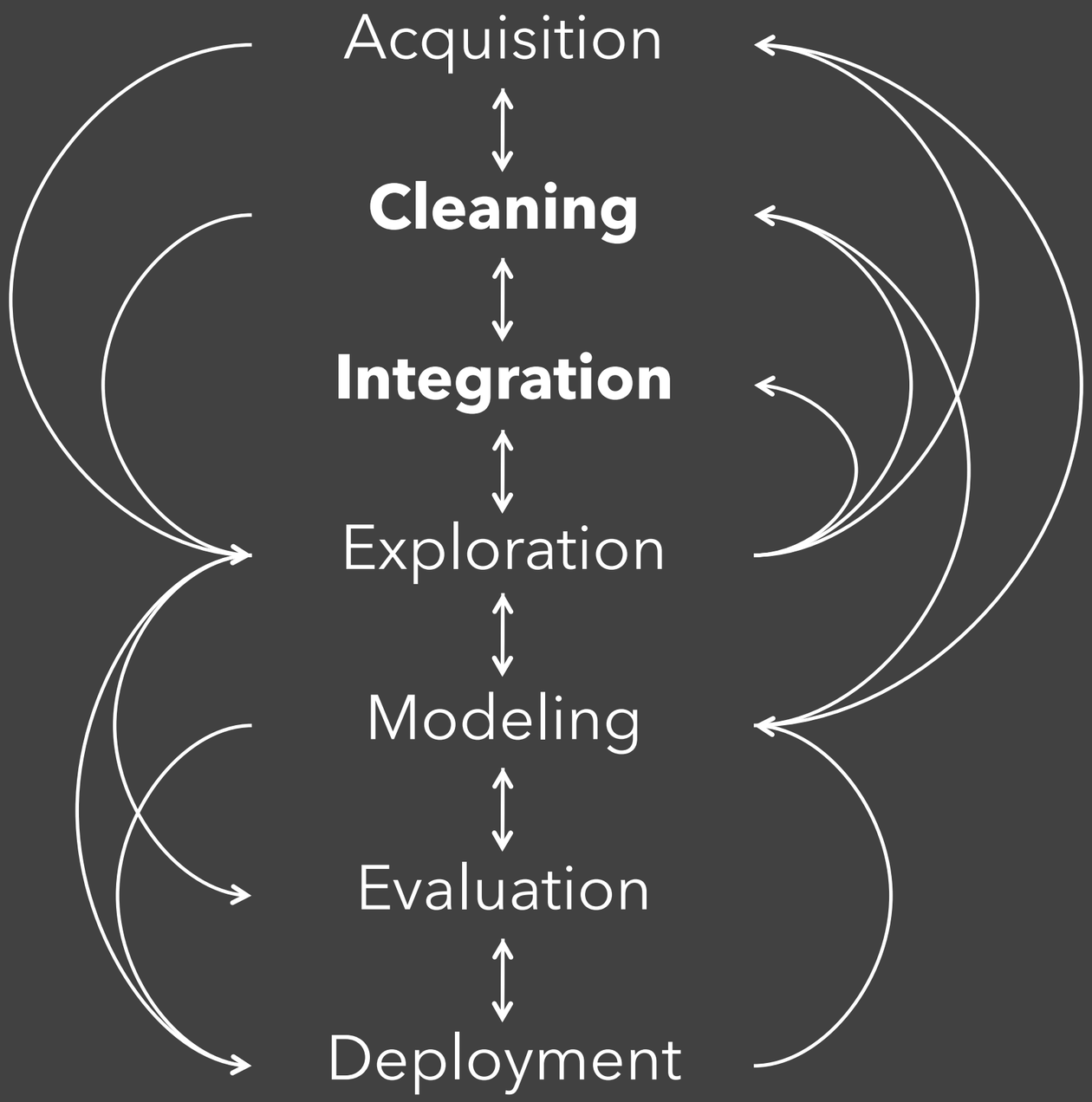
"The related view suggestion accelerates exploration a lot."

"I like that it shows me what fields to include in order to see a specific graph. Otherwise, I have to do a lot of trial and error and can't express what I wanted to see."

"These related views are so good but it's also spoiling that I start thinking less. I'm not sure if that's really a good thing."




Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment


Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment

I spend more than half of my time integrating, cleansing and transforming data without doing any actual analysis. Most of the time I'm lucky if I get to do any "analysis" at all.

Anonymous Data Scientist
from our 2012 interview study

Big Data Borat

@BigDataBorat

Following

In Data Science, 80% of time spent prepare data, 20% of time spent complain about need for prepare data.

How might we assist the process of **data transformation**?

Bureau of Justice Statistics - Data Online
http://bjs.ojp.usdoj.gov/

Reported crime in Alabama

Year	Population		Property crime rate		Burglary rate	Larceny-theft rate	Motor vehicle theft rate
2004	4525375	4029.3	987	2732.4	309.9		
2005	4548327	3900	955.8	2656	289		
2006	4599030	3937	968.9	2645.1	322.9		
2007	4627851	3974.9	980.2	2687	307.7		
2008	4661900	4081.9	1080.7	2712.6	288.6		

Reported crime in Alaska

Year	Population		Property crime rate		Burglary rate	Larceny-theft rate	Motor vehicle theft rate
2004	657755	3370.9	573.6	2456.7	340.6		
2005	663253	3615	622.8	2601	391		
2006	670053	3582	615.2	2588.5	378.3		
2007	683478	3373.9	538.9	2480	355.1		
2008	686293	2928.3	470.9	2219.9	237.5		

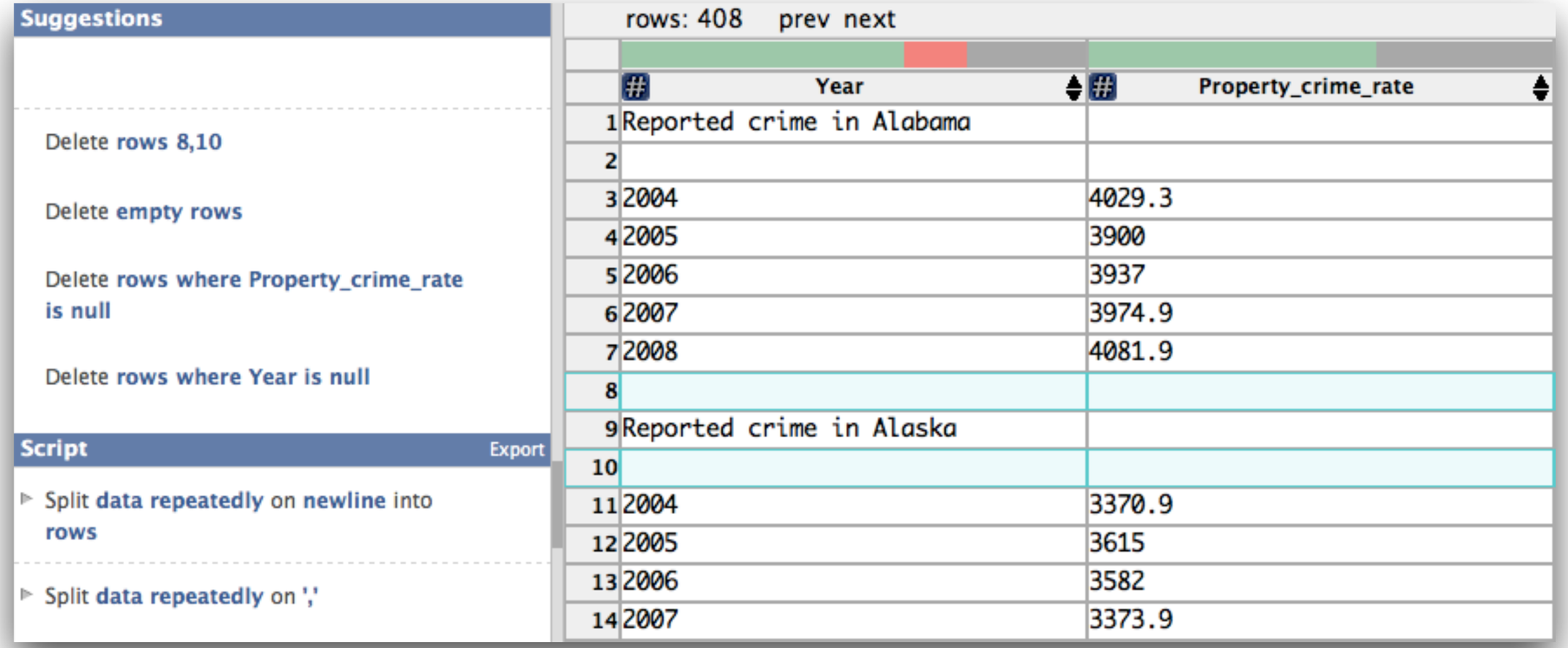
Reported crime in Arizona

Year	Population		Property crime rate		Burglary rate	Larceny-theft rate	Motor vehicle theft rate
2004	5739879	5073.3	991	3118.7	963.5		
2005	5953007	4827	946.2	2958	922		
2006	6166318	4741.6	953	2874.1	914.4		
2007	6338755	4502.6	935.4	2780.5	786.7		
2008	6500180	4087.3	894.2	2605.3	587.8		

Reported crime in Arkansas

Year	Population		Property crime rate		Burglary rate	Larceny-theft rate	Motor vehicle theft rate
2004	2750000	4033.1	1096.4	2699.7	237		
2005	2775708	4068	1085.1	2720	262		
2006	2810872	4021.6	1154.4	2596.7	270.4		
2007	2834797	3945.5	1124.4	2574.6	246.5		
2008	2855390	3843.7	1182.7	2433.4	227.6		

DataWrangler



Wrangler: Interactive Visual Specification of Data Transformation Scripts

Sean Kandel, Joseph Hellerstein et al. *CHI'11*

Reduces specification time, promotes the use of reusable, scalable transformations rather than idiosyncratic edits.

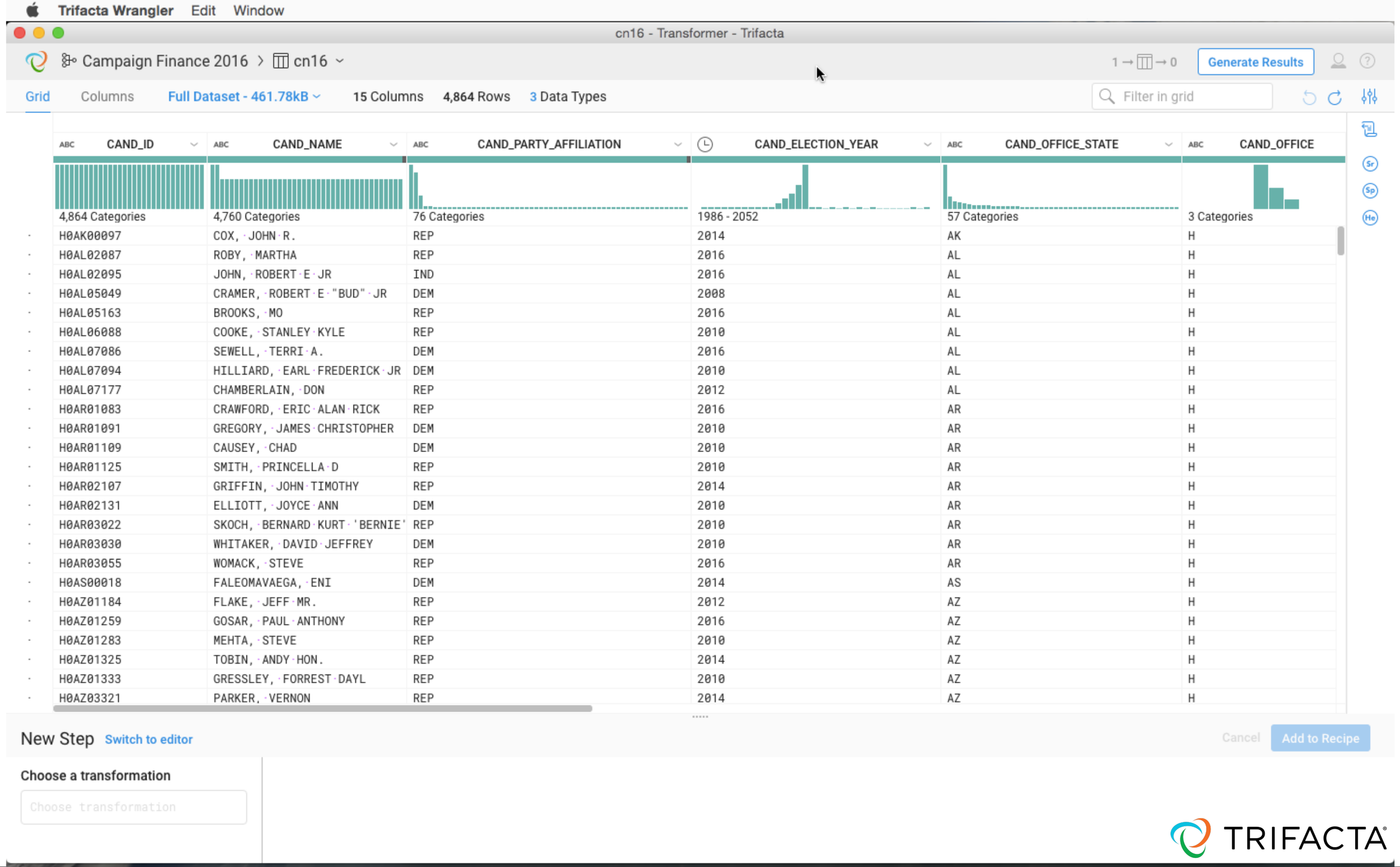
Complementarity: Suggestions good for tasks people found hard (extraction patterns, table reshaping). People good in cases where inference is less tractable (arbitrary formulas).

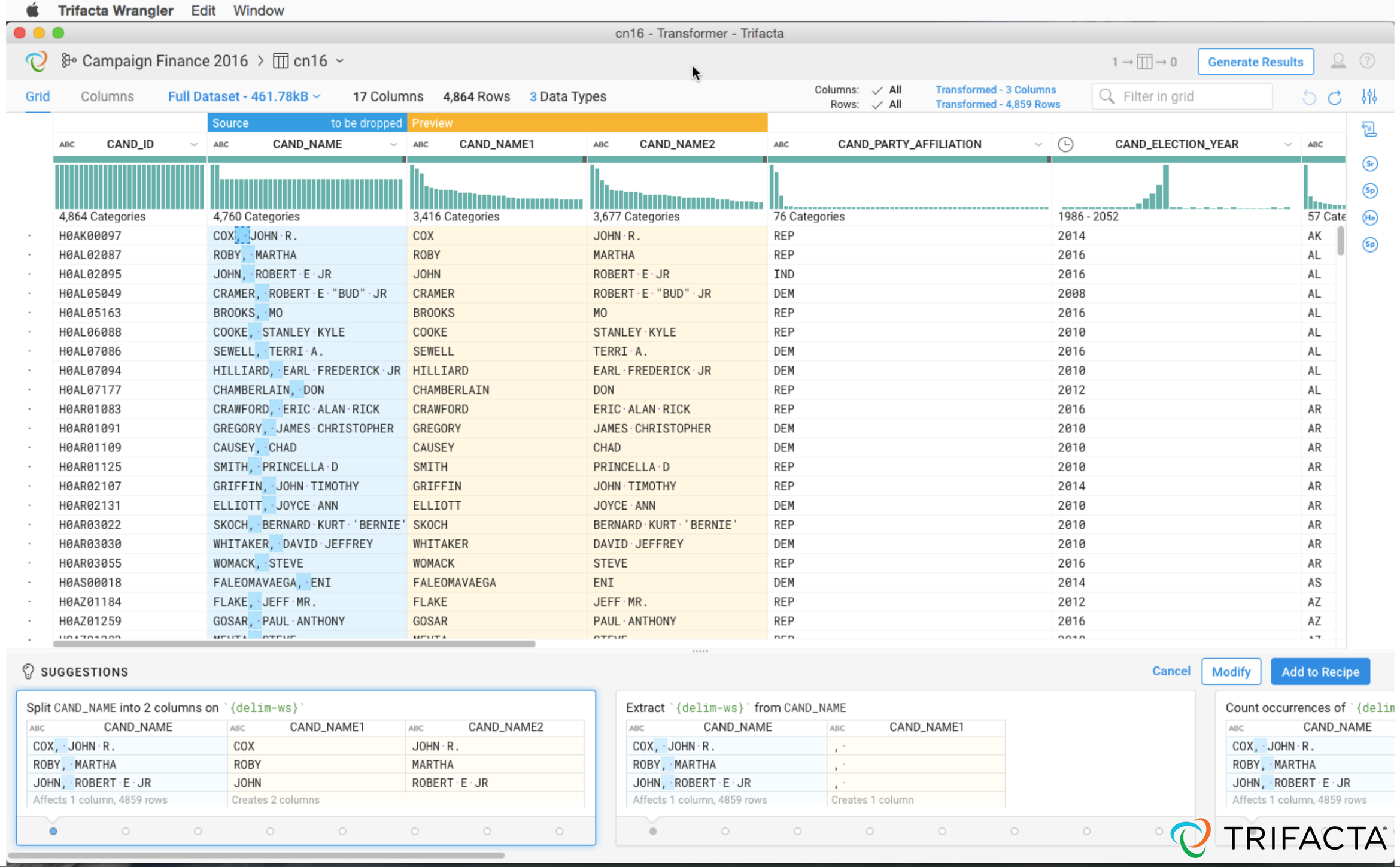
Agency: Users appreciated suggestions in response to an initiating interaction, but did not always act on *proactive* assistance, often preferring to maintain the initiative.

Wrangler: Interactive Visual Specification of Data Transformation Scripts

Kandel et al. [CHI 2011]

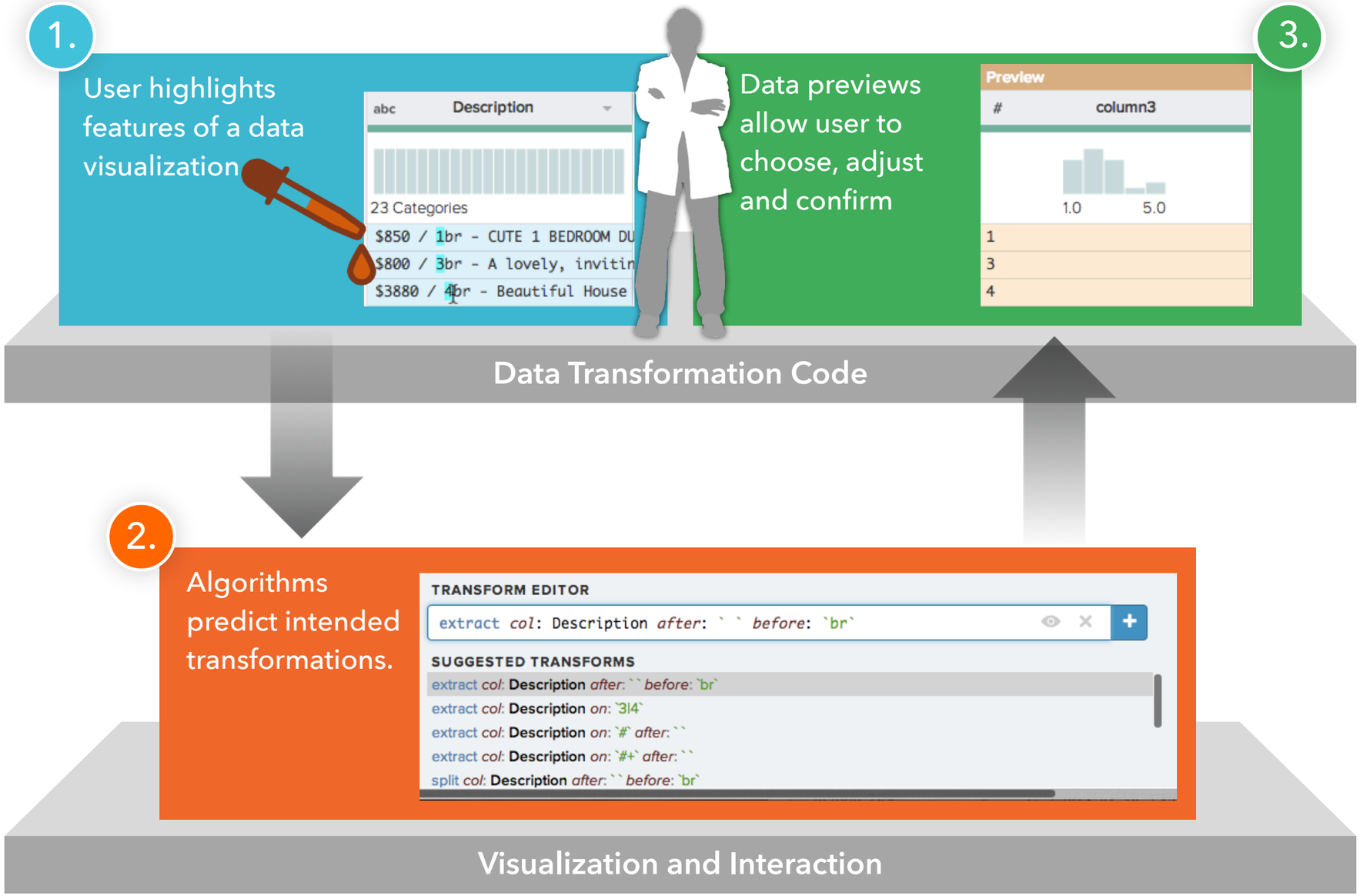

TRIFACTA®


Trifacta Wrangler Edit Window
cn16 - Transformer - Trifacta
Campaign Finance 2016 › cn16
1 → → 0
Generate Results
Grid Columns Full Dataset - 461.78kB 15 Columns 4,864 Rows 3 Data Types
Filter in grid
ABC CAND_ID
ABC CAND_NAME
ABC CAND_PARTY_AFFILIATION
CAND_ELECTION_YEAR
ABC CAND_OFFICE_STATE
ABC CAND_OFFICE
4,864 Categories
4,760 Categories
76 Categories
1986 - 2052
57 Categories
3 Categories
H0AK00097 COX, JOHN R. REP 2014 AK H
H0AL02087 ROBY, MARTHA REP 2016 AL H
H0AL02095 JOHN, ROBERT E JR IND 2016 AL H
H0AL05049 CRAMER, ROBERT E "BUD" JR DEM 2008 AL H
H0AL05163 BROOKS, MO REP 2016 AL H
H0AL06088 COOKE, STANLEY KYLE REP 2010 AL H
H0AL07086 SEWELL, TERRI A. DEM 2016 AL H
H0AL07094 HILLIARD, EARL FREDERICK JR DEM 2010 AL H
H0AL07177 CHAMBERLAIN, DON REP 2012 AL H
H0AR01083 CRAWFORD, ERIC ALAN RICK REP 2016 AR H
H0AR01091 GREGORY, JAMES CHRISTOPHER DEM 2010 AR H
H0AR01109 CAUSEY, CHAD DEM 2010 AR H
H0AR01125 SMITH, PRINCELLA D REP 2010 AR H
H0AR02107 GRIFFIN, JOHN TIMOTHY REP 2014 AR H
H0AR02131 ELLIOTT, JOYCE ANN DEM 2010 AR H
H0AR03022 SKOCH, BERNARD KURT 'BERNIE' REP 2010 AR H
H0AR03030 WHITAKER, DAVID JEFFREY DEM 2010 AR H
H0AR03055 WOMACK, STEVE REP 2016 AR H
H0AS00018 FALEOMAVAEGA, ENI DEM 2014 AS H
H0AZ01184 FLAKE, JEFF MR. REP 2012 AZ H
H0AZ01259 GOSAR, PAUL ANTHONY REP 2016 AZ H
H0AZ01283 MEHTA, STEVE REP 2010 AZ H
H0AZ01325 TOBIN, ANDY HON. REP 2014 AZ H
H0AZ01333 GRESSLEY, FORREST DAYL REP 2010 AZ H
H0AZ03321 PARKER, VERNON REP 2014 AZ H
New Step Switch to editor
Cancel
Add to Recipe
Choose a transformation
Choose transformation
TRIFACTA

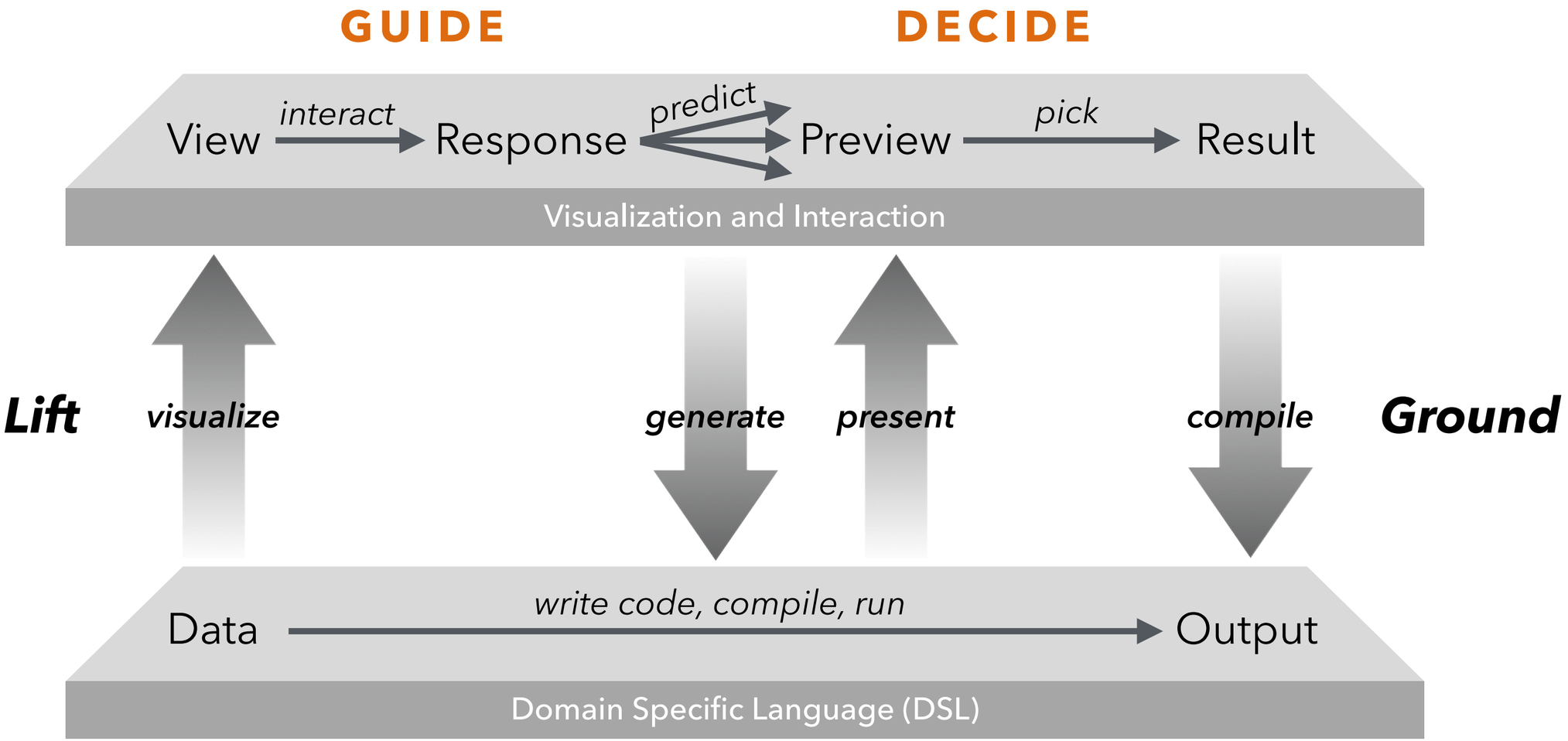

Trifacta Wrangler Edit Window
cn16 - Transformer - Trifacta
Campaign Finance 2016 › cn16
Generate Results
Grid Columns Full Dataset - 461.78kB 17 Columns 4,864 Rows 3 Data Types
Columns: All Rows: All
Transformed - 3 Columns
Transformed - 4,859 Rows
Filter in grid
Source to be dropped
Preview
CAND_ID CAND_NAME CAND_NAME1 CAND_NAME2 CAND_PARTY_AFFILIATION CAND_ELECTION_YEAR
4,864 Categories 4,760 Categories 3,416 Categories 3,677 Categories 76 Categories 1986 - 2052 57 Cate
H0AK00097 COX, JOHN R. COX JOHN R. REP 2014 AK
H0AL02087 ROBY, MARTHA ROBY MARTHA REP 2016 AL
H0AL02095 JOHN, ROBERT E JR JOHN ROBERT E JR IND 2016 AL
H0AL05049 CRAMER, ROBERT E "BUD" JR CRAMER ROBERT E "BUD" JR DEM 2008 AL
H0AL05163 BROOKS, MO BROOKS MO REP 2016 AL
H0AL06088 COOKE, STANLEY KYLE COOKE STANLEY KYLE REP 2010 AL
H0AL07086 SEWELL, TERRI A. SEWELL TERRI A. DEM 2016 AL
H0AL07094 HILLIARD, EARL FREDERICK JR HILLIARD EARL FREDERICK JR DEM 2010 AL
H0AL07177 CHAMBERLAIN, DON CHAMBERLAIN DON REP 2012 AL
H0AR01083 CRAWFORD, ERIC ALAN RICK CRAWFORD ERIC ALAN RICK REP 2016 AR
H0AR01091 GREGORY, JAMES CHRISTOPHER GREGORY JAMES CHRISTOPHER DEM 2010 AR
H0AR01109 CAUSEY, CHAD CAUSEY CHAD DEM 2010 AR
H0AR01125 SMITH, PRINCELLA D SMITH PRINCELLA D REP 2010 AR
H0AR02107 GRIFFIN, JOHN TIMOTHY GRIFFIN JOHN TIMOTHY REP 2014 AR
H0AR02131 ELLIOTT, JOYCE ANN ELLIOTT JOYCE ANN DEM 2010 AR
H0AR03022 SKOCH, BERNARD KURT 'BERNIE' SKOCH BERNARD KURT 'BERNIE' REP 2010 AR
H0AR03030 WHITAKER, DAVID JEFFREY WHITAKER DAVID JEFFREY DEM 2010 AR
H0AR03055 WOMACK, STEVE WOMACK STEVE REP 2016 AR
H0AS00018 FALEOMAVAEGA, ENI FALEOMAVAEGA ENI DEM 2014 AS
H0AZ01184 FLAKE, JEFF MR. FLAKE JEFF MR. REP 2012 AZ
H0AZ01259 GOSAR, PAUL ANTHONY GOSAR PAUL ANTHONY REP 2016 AZ
SUGGESTIONS
Cancel
Modify
Add to Recipe
Split CAND_NAME into 2 columns on `{delim-ws}`
CAND_NAME CAND_NAME1 CAND_NAME2
COX, JOHN R. COX JOHN R.
ROBY, MARTHA ROBY MARTHA
JOHN, ROBERT E JR JOHN ROBERT E JR
Affects 1 column, 4859 rows
Creates 2 columns
Extract `{delim-ws}` from CAND_NAME
CAND_NAME CAND_NAME1
COX, JOHN R. ,
ROBY, MARTHA ,
JOHN, ROBERT E JR ,
Affects 1 column, 4859 rows
Creates 1 column
Count occurrences of `{delim
CAND_NAME
COX, JOHN R.
ROBY, MARTHA
JOHN, ROBERT E JR
Affects 1 column, 4859 rows
TRIFACTA

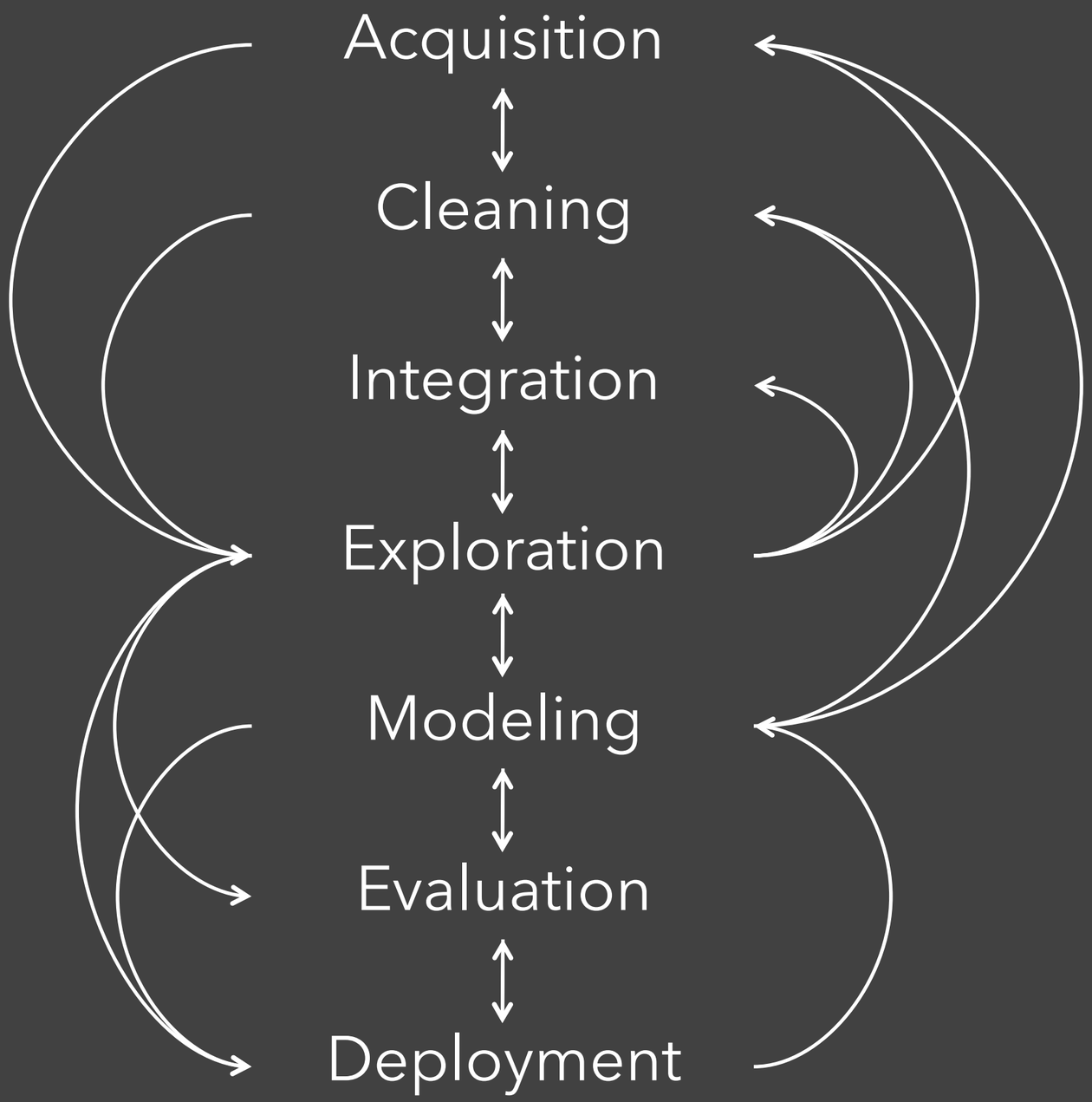
Traditional Specification

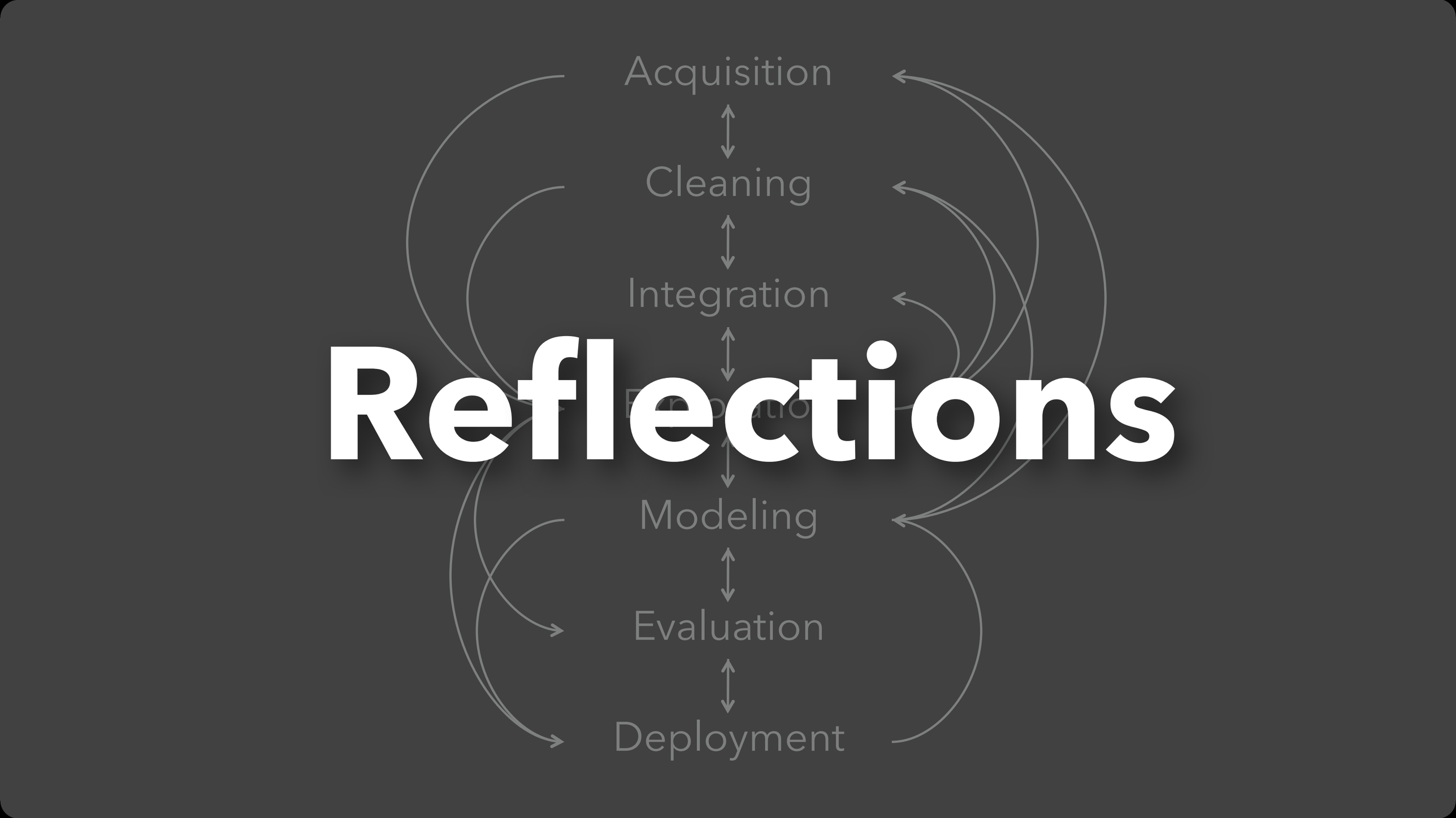
Predictive Interaction



Predictive Interaction [Heer, Hellerstein, Kandel CIDR'15]




Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment



Reflections

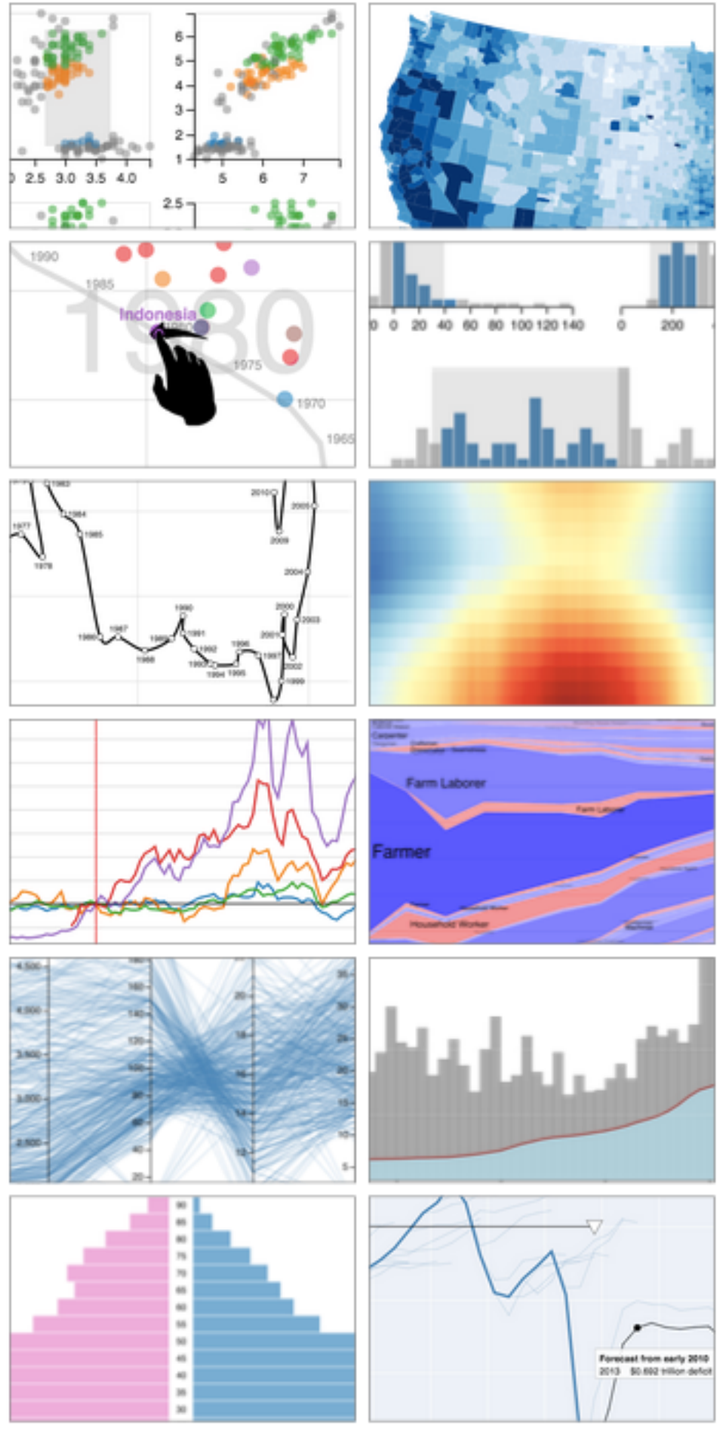
Successes

The unreasonable effectiveness of DSLs

Provide users with high-level specification

Decouple frontend and backend concerns

Model tasks, enable formal reasoning for computational search & recommendation



Successes

The unreasonable effectiveness of DSLs

Provide users with high-level specification

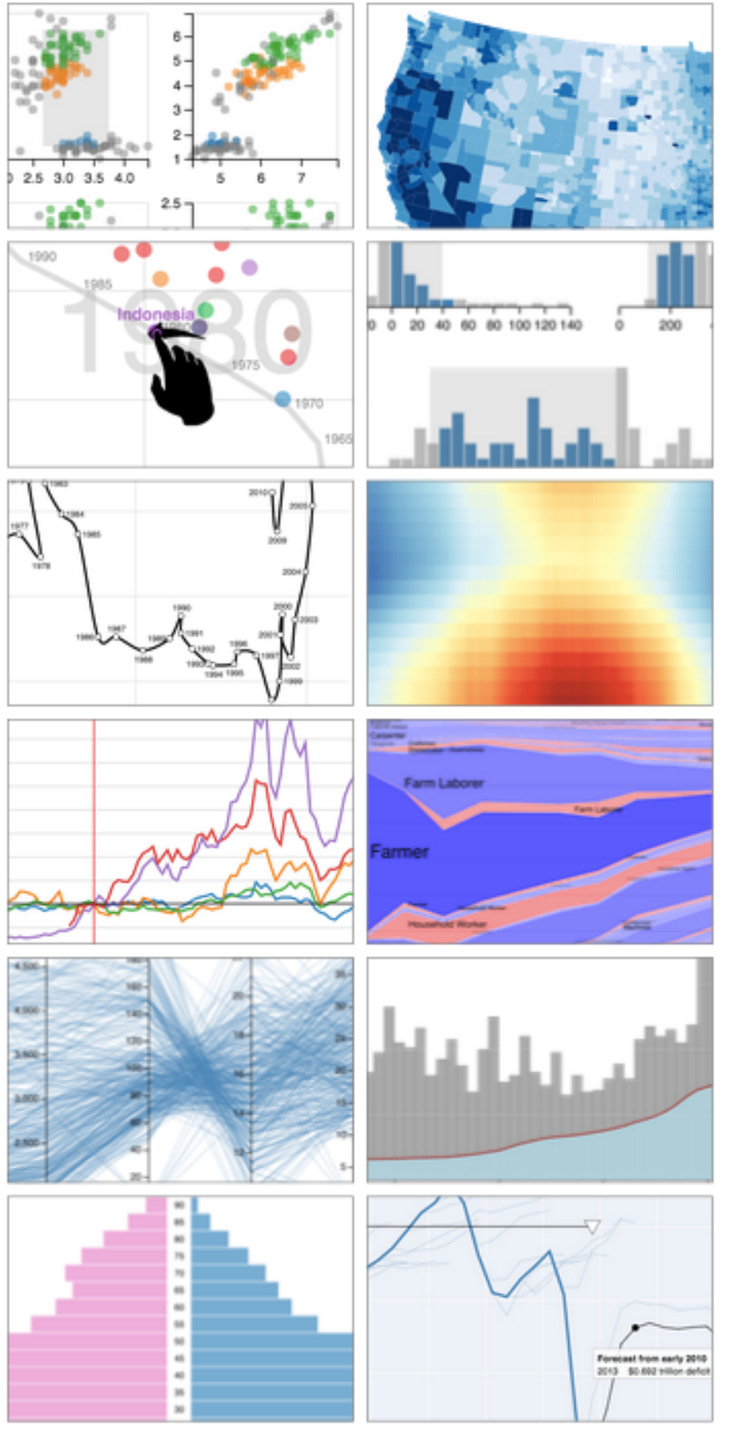
Decouple frontend and backend concerns

Model tasks, enable formal reasoning for computational search & recommendation

AI assistance to bridge user intent

Recommend possible next steps

Aid discovery and learning



Successes

The unreasonable effectiveness of DSLs

Provide users with high-level specification

Decouple frontend and backend concerns

Model tasks, enable formal reasoning for computational search & recommendation

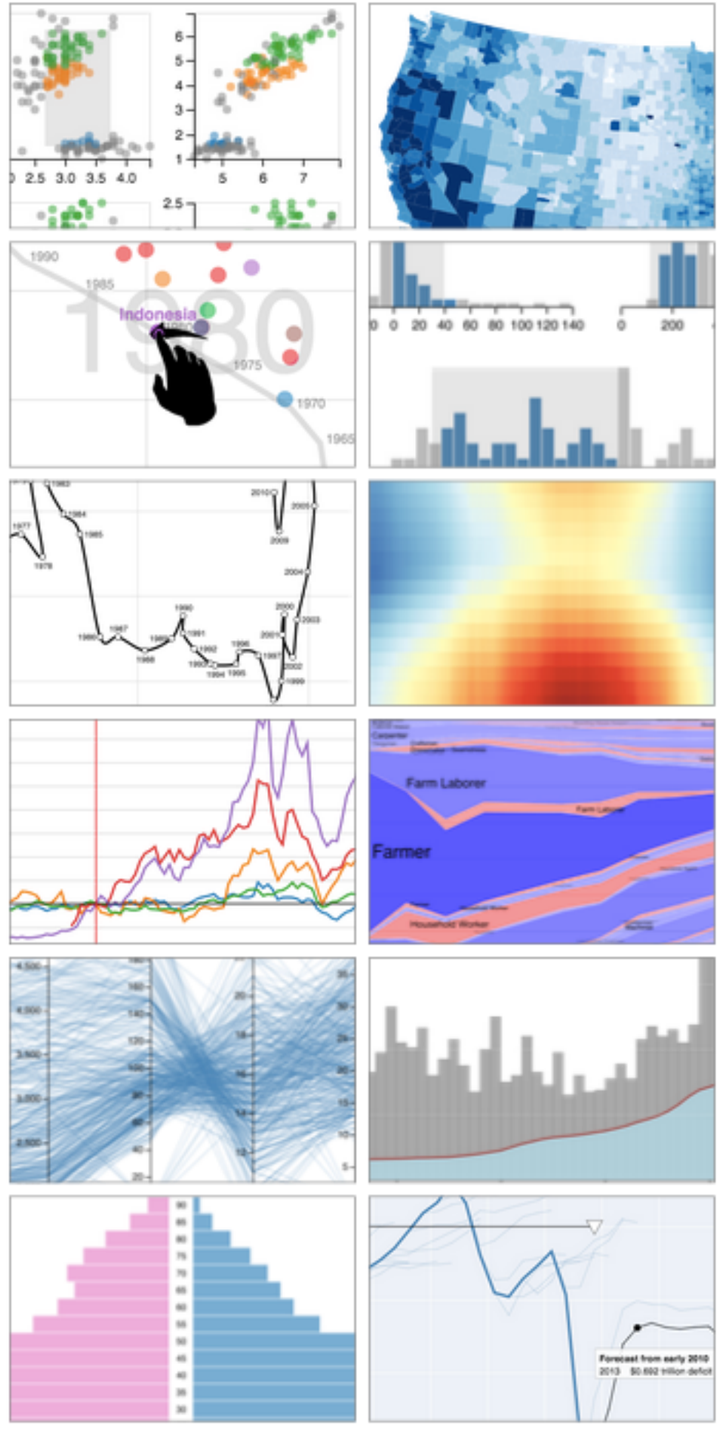
AI assistance to bridge user intent

Recommend possible next steps

Aid discovery and learning

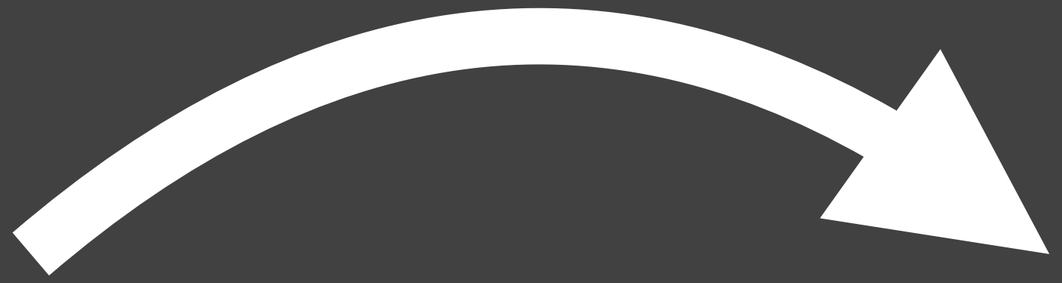
How to balance agency and automation?

Augment, not replace! [Heer, PNAS 2019]



A Virtuous Cycle

Declarativity, Scalability, Query Optimization



Databases **HCI**

New Requirements, Interaction Models, Prefetching Approaches

A Vi

The ability to take data—to be able to **understand** it, to **process** it, to **extract value** from it, to **visualize** it, to **communicate** it—that's going to be a hugely important skill in the next decades, … because now we really do have essentially free and ubiquitous data. So the complimentary **scarce factor** is the ability to **understand** that data and **extract value** from it.

Hal Varian, Google's Chief Economist
The McKinsey Quarterly, Jan 2009

The ability to take data—to be able to **understand** it, to **process** it, to **extract value** from it, to **visualize** it, to **communicate** it—that's going to be a hugely important skill in the next decades, because now we really do have essentially free and ubiquitous data. So the complimentary **scarce factor** is the ability to **understand** that data and **extract value** from it.

"free" to whom?

"ubiquitous" about whom?

…to whose benefit?

Hal Varian, Google's Chief Economist
The McKinsey Quarterly, Jan 2009

Life-size cutouts of Facebook CEO Mark Zuckerberg are displayed by a progressive advocacy group on the lawn of the U.S. Capitol on Tuesday.

Carolyn Kaster / Reuters

My Facebook Was Breached by Cambridge Analytica. Was Yours?

How to find out if you are one of the 87 million victims

ROBINSON MEYER | APR 10, 2018 | TECHNOLOGY

Psychology's Replication Crisis Can't Be Wished Away

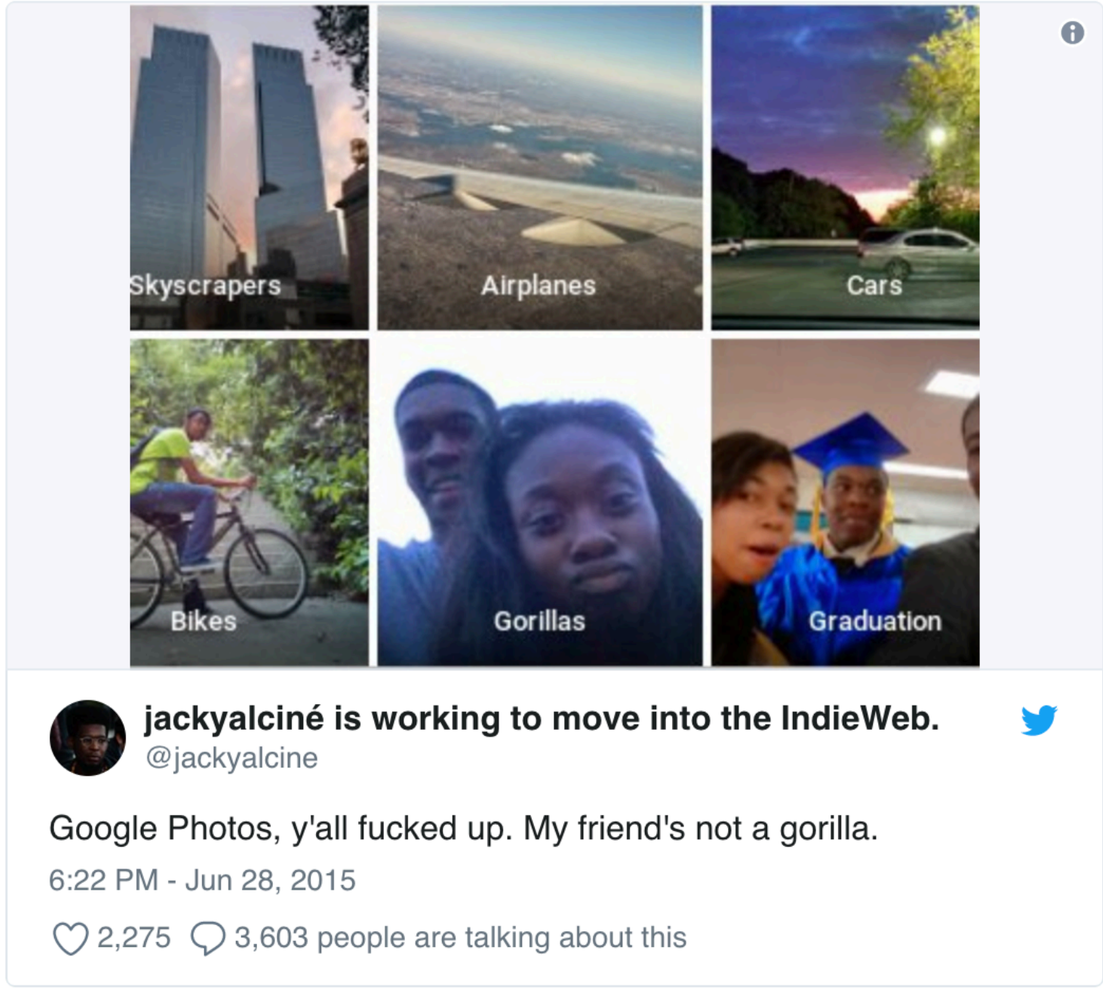
It has a real and heartbreaking cost.

ED YONG | MAR 4, 2016 | SCIENCE

Inequality

Rise of the racist robots – how AI is learning all our worst impulses

There is a saying in computer science: garbage in, garbage out. When we feed machines data that reflects our prejudices, they mimic them – from antisemitic chatbots to racially biased software. Does a horrifying future await people forced to live at the mercy of algorithms?



“The nature, scale, depth and consequences of the data, technical and ethical breaches understood to have occurred thus far … are unlikely to be confined to a single company, technology or industry.”

US ACM Letter to Senate Commerce & Judiciary Committee

People's **interaction with data** is subject to **misuse** and **abuse**.

Examples of Data Analysis Gone Awry

Poor decisions or operations due to faulty inference.

Inaccurate models due to biased or inaccurate data.

Insufficient accountability and review of analysis procedures.

Lack of monitoring and enforcement of data usage patterns.

Examples of Data Analysis Gone Awry

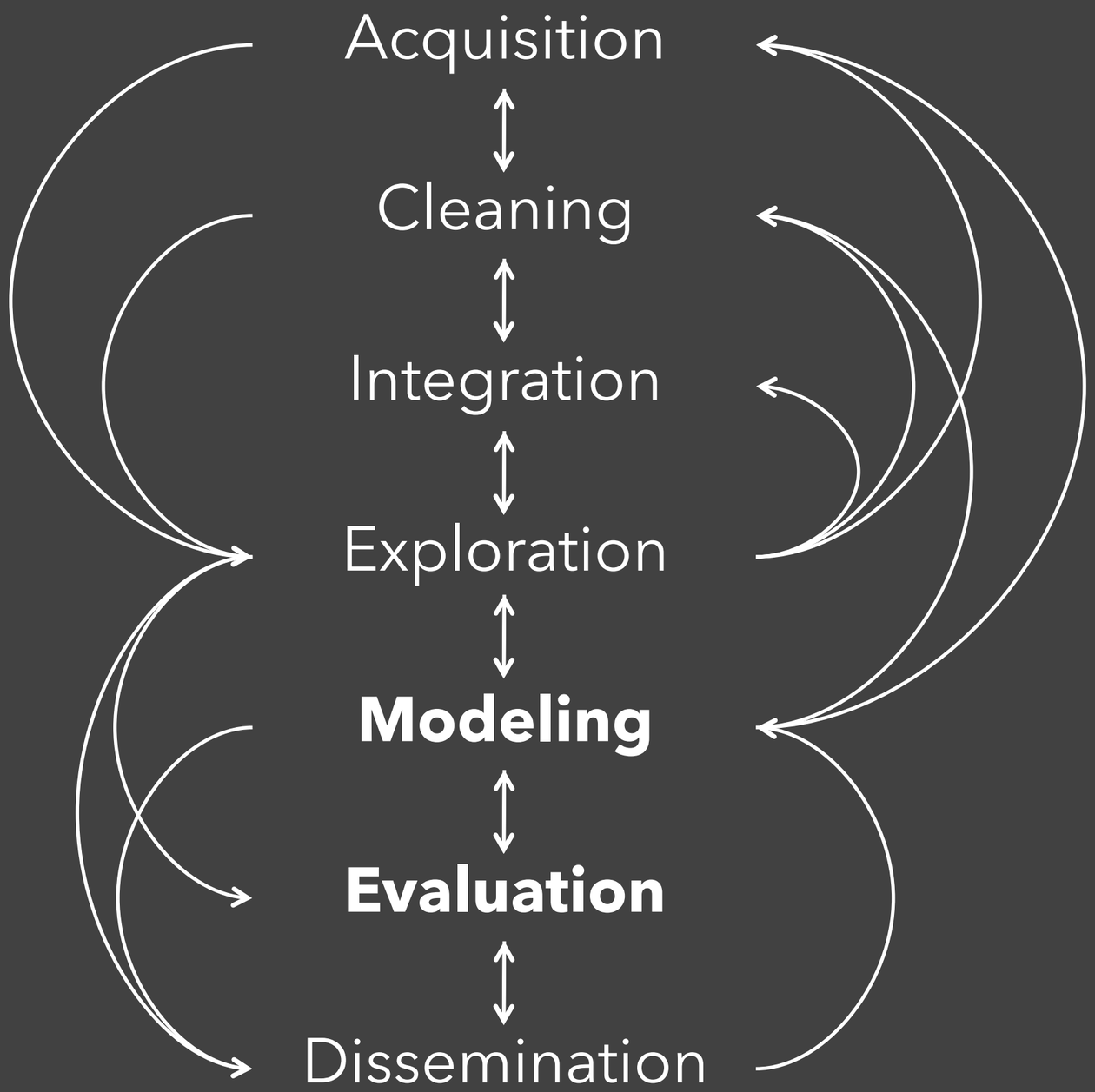
Poor decisions or operations due to faulty inference.

Inaccurate models due to biased or inaccurate data.

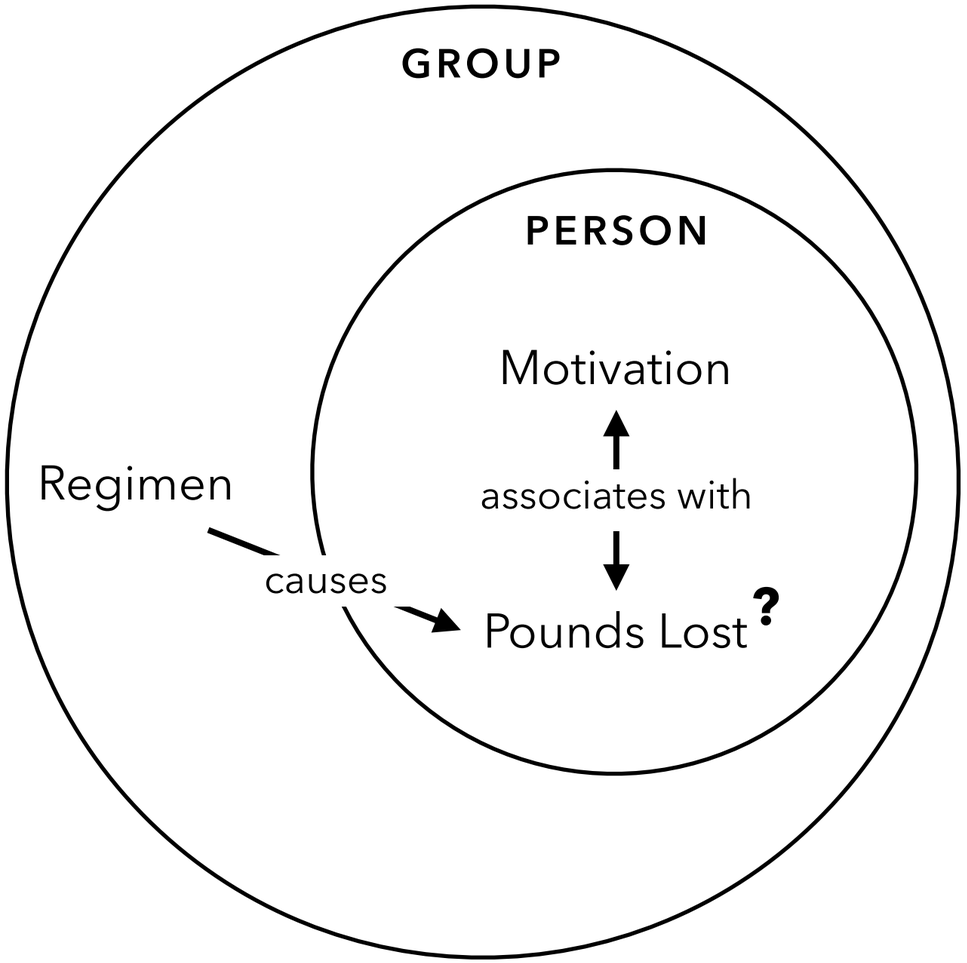
Insufficient accountability and review of analysis procedures.

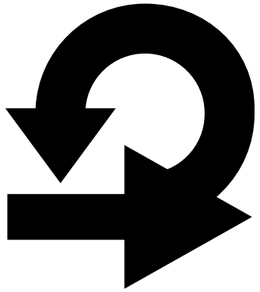
Lack of monitoring and enforcement of data usage patterns.

Problems arise throughout end-to-end analysis workflows!


Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Dissemination

Tisane: From Conceptual Models to Statistical Models

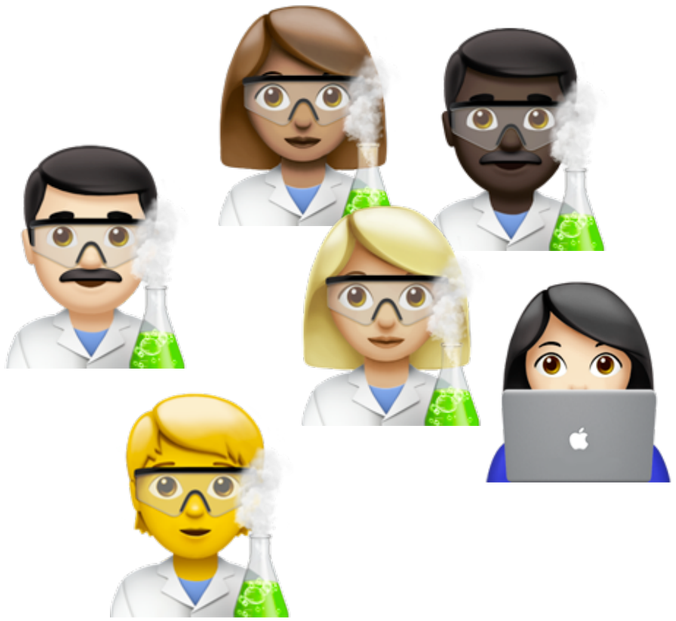




```
lmer(
 pounds_lost ~ regimen
               + motivation
               + (1|group),
 data=df
)
```

Eunice Jun, A. Seo, J. Heer, R. Just. CHI 2022

“Many Analyst” studies



→

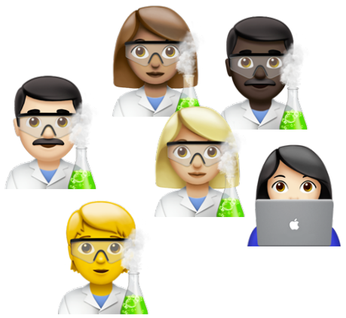
✅ Same dataset,

✅ Same hypothesis

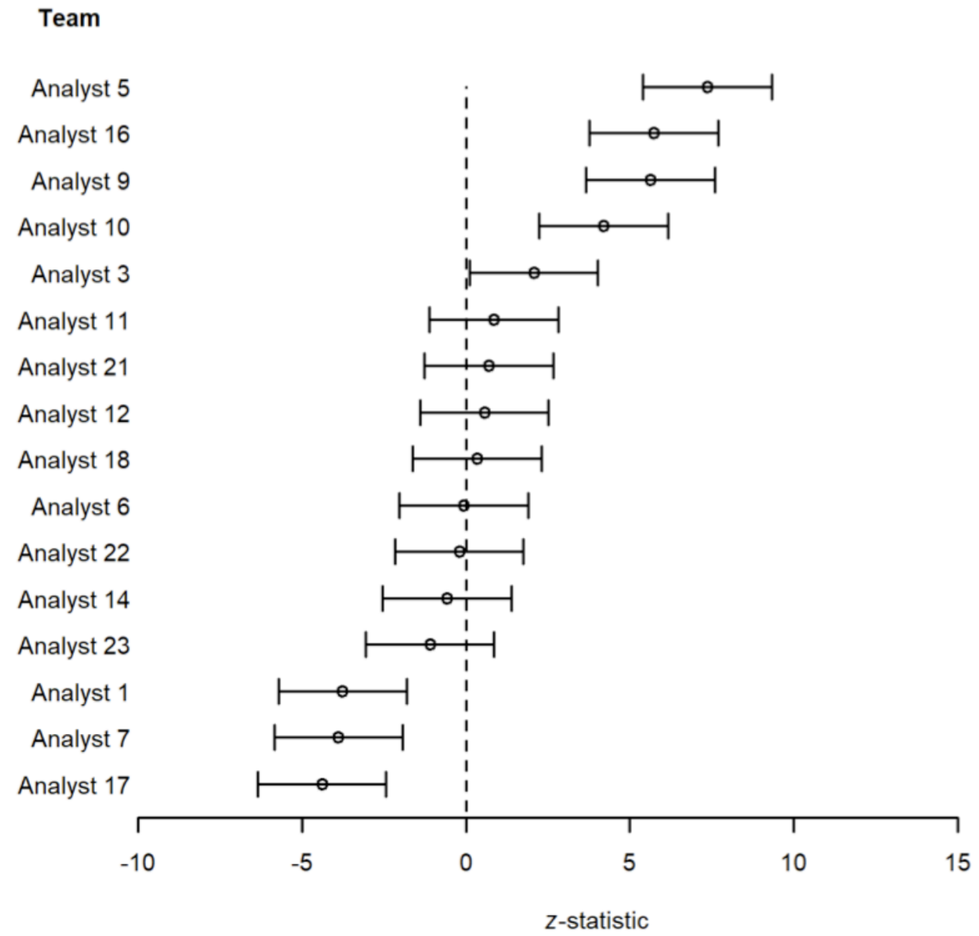
❓ Same results

Silberzahn et al. (2018) “Many analysts, one data set: Making transparent how variations in analytic choices affect results." *AMPPS* 1.3: 337-356.
Schweinsberg et al. (2021) “Same data, different conclusions: Radical dispersion in empirical results when independent analysts operationalize and test the same hypothesis” Organizational Behavior and Human Decision Processes, 165: 228-249.
Huntington-Klein et al. (2021) "The Influence of Hidden Researcher Decisions in Applied Microeconomics" Economic Inquiry. 59 (3): 944-960.

“Many Analyst” studies



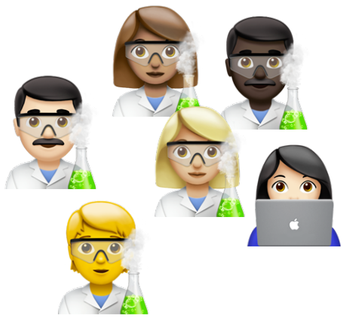
Same dataset,
Same hypothesis



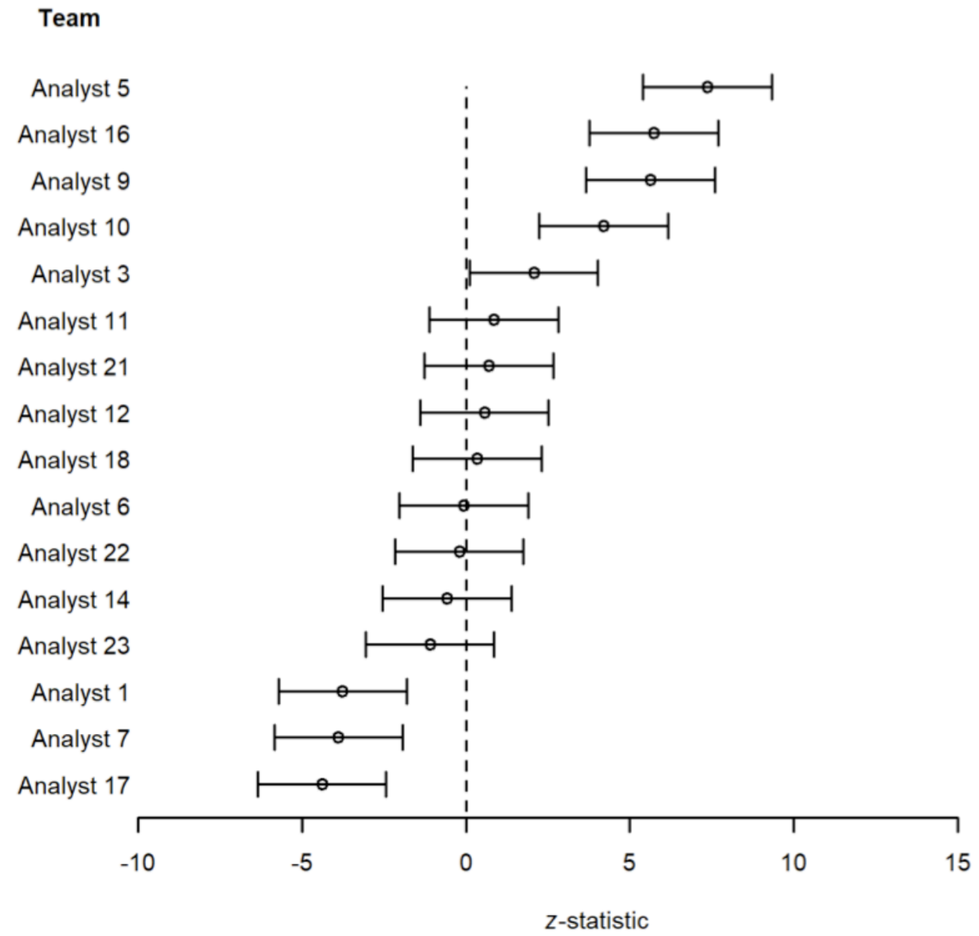
Very different results

Silberzahn et al. "Many analysts, one data set: Making transparent how variations in analytic choices affect results." *AMPPS* 1.3 (2018): 337-356.

“Many Analyst” studies



Same dataset,
Same hypothesis



Variations not explained by

Expertise

Prior belief

Peer-rated quality of the analysis

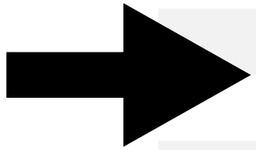
…

Very different results

Silberzahn et al. "Many analysts, one data set: Making transparent how variations in analytic choices affect results." *AMPPS* 1.3 (2018): 337-356.

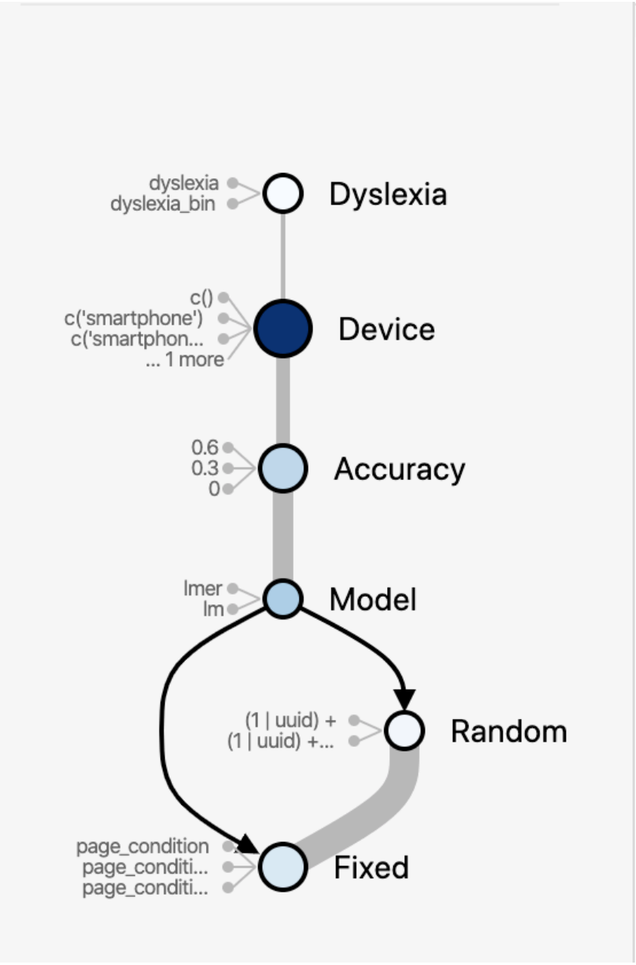
Boba: Authoring & Visualizing Multiverse Analyses

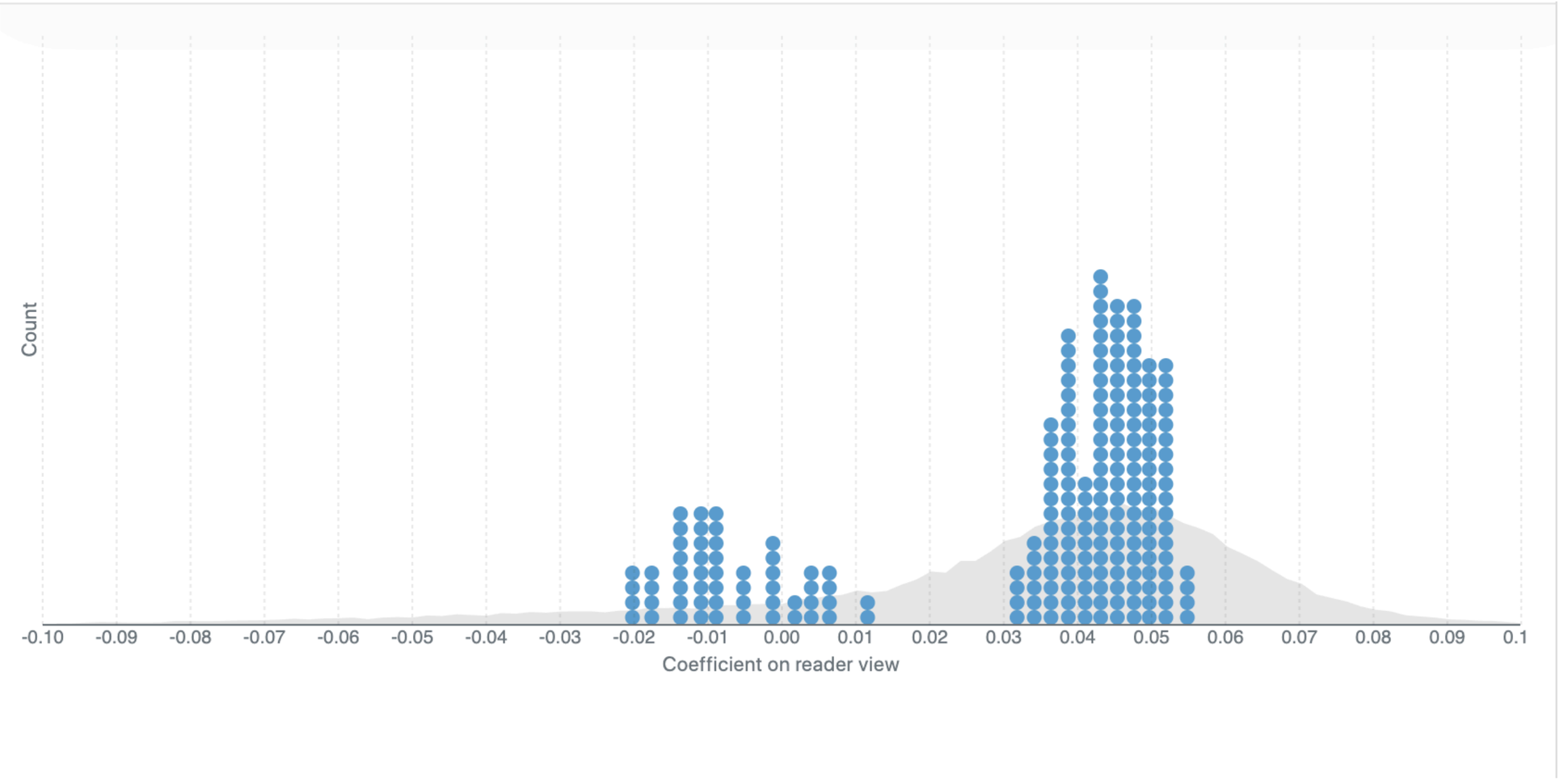
```
# --- (Shared)
df = read_csv("data.csv") %>%
     filter(accuracy > {{cutoff=0, 0.6}})

# --- (Model) lm
model = lm(speed ~ accuracy, data = df)

# --- (Model) lmer
model = lmer(speed ~ accuracy +
        {{random="(1|uid)", "(1|page_id"}})
```



Yang Liu, A. Kale, T. Althoff, J. Heer. IEEE TVCG 2021

Boba: Authoring & Visualizing Multiverse Analyses





Yang Liu, A. Kale, T. Althoff, J. Heer. IEEE TVCG 2021

Errudite: Reproducible & Testable Error Analysis

How many people are in this picture?

groundtruth:**3** *(* 10)*

saaa:**2** vqacounting:**3**

How many brownish peaks are there?

groundtruth:**2** *(* 10)*

saaa:**4** vqacounting:**5**

DID YOU MEAN TO FILTER INSTANCES THAT ARE... Close Now

```
starts_with(question, pattern="how many ADJ")
starts_with(question, pattern="ADV ADJ ADJ")
attr:question_type == "how many"
```

A

See more general suggestions?

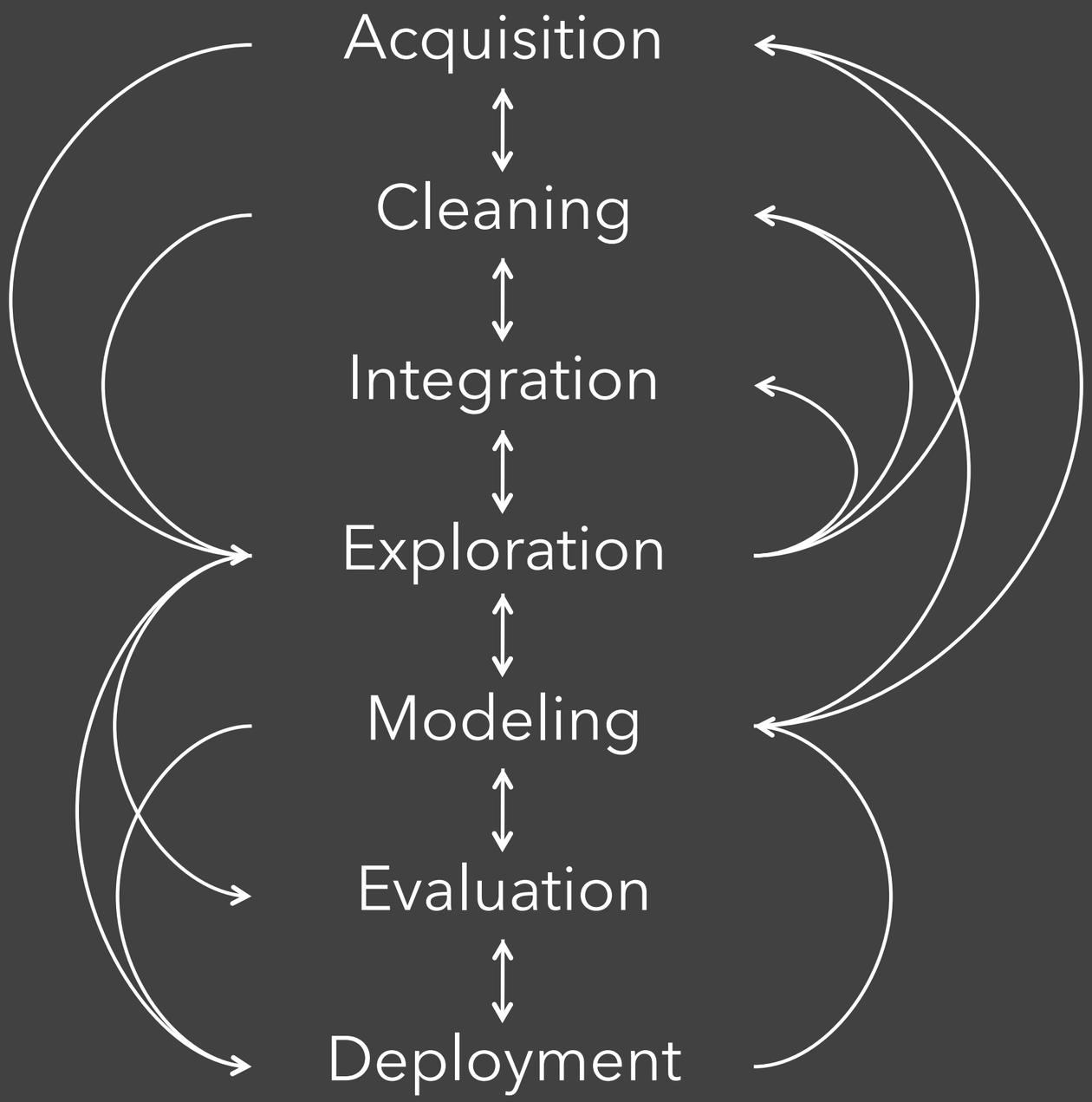
	saaa		vqacounting	
all_instances	121512	43% 57%	121512	39% 61%
how_many_noun	11471	62% 38%	11471	51% 49%
how_many_adj	788	66% 34%	788	63% 37%

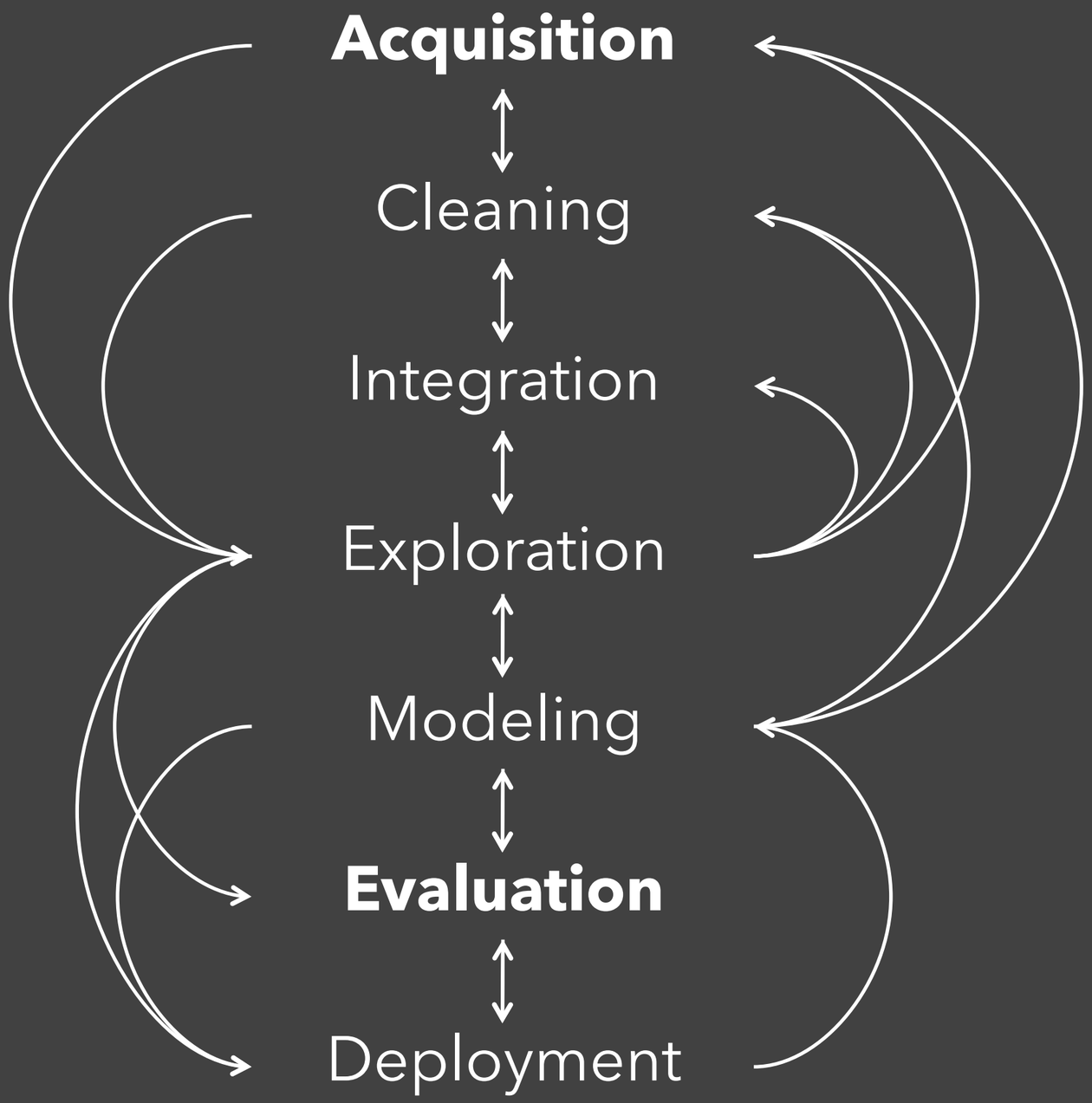
How many ~~brownish~~ peaks are there?

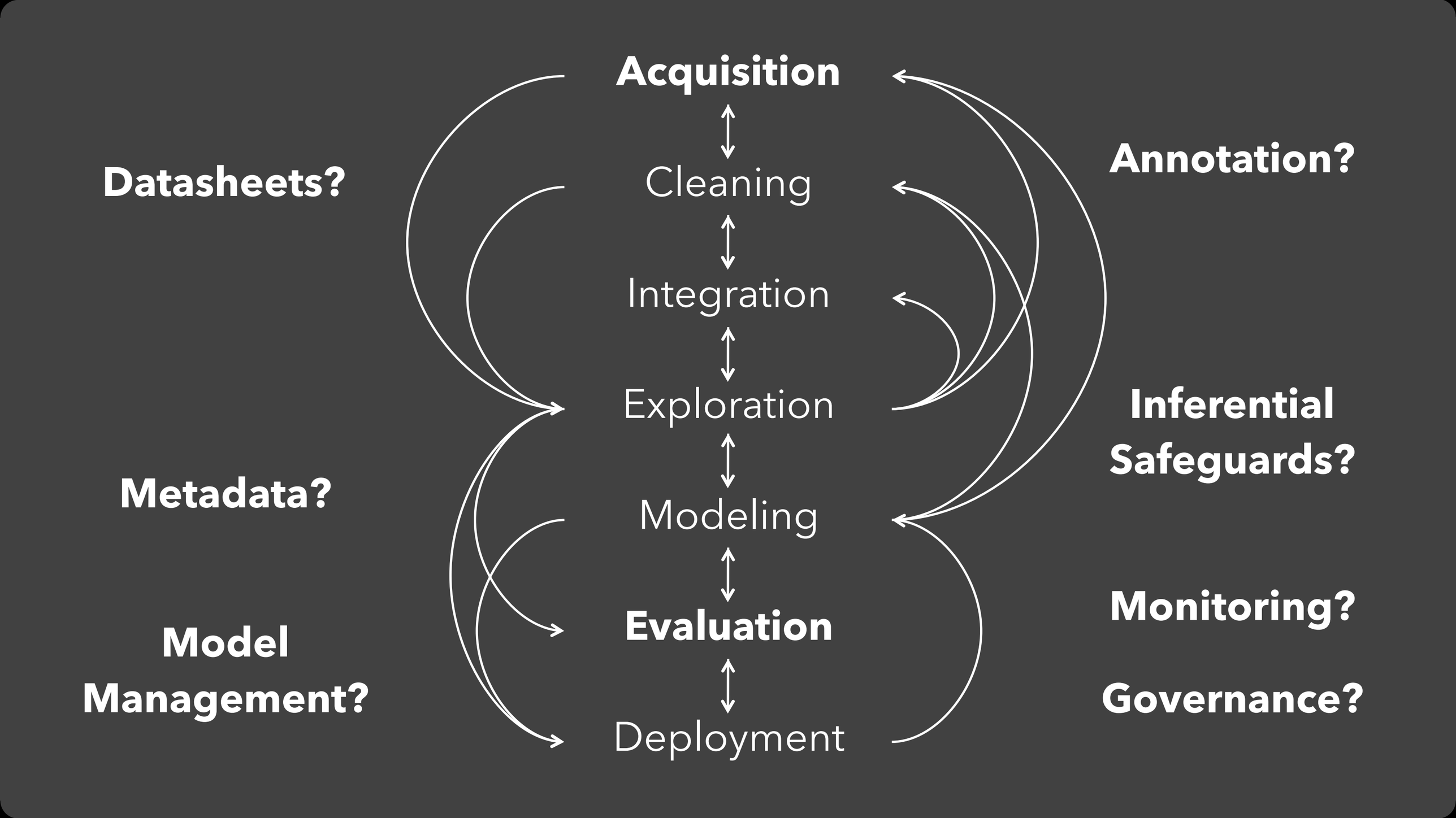
DID YOU WANT TO GENERALIZE TO...

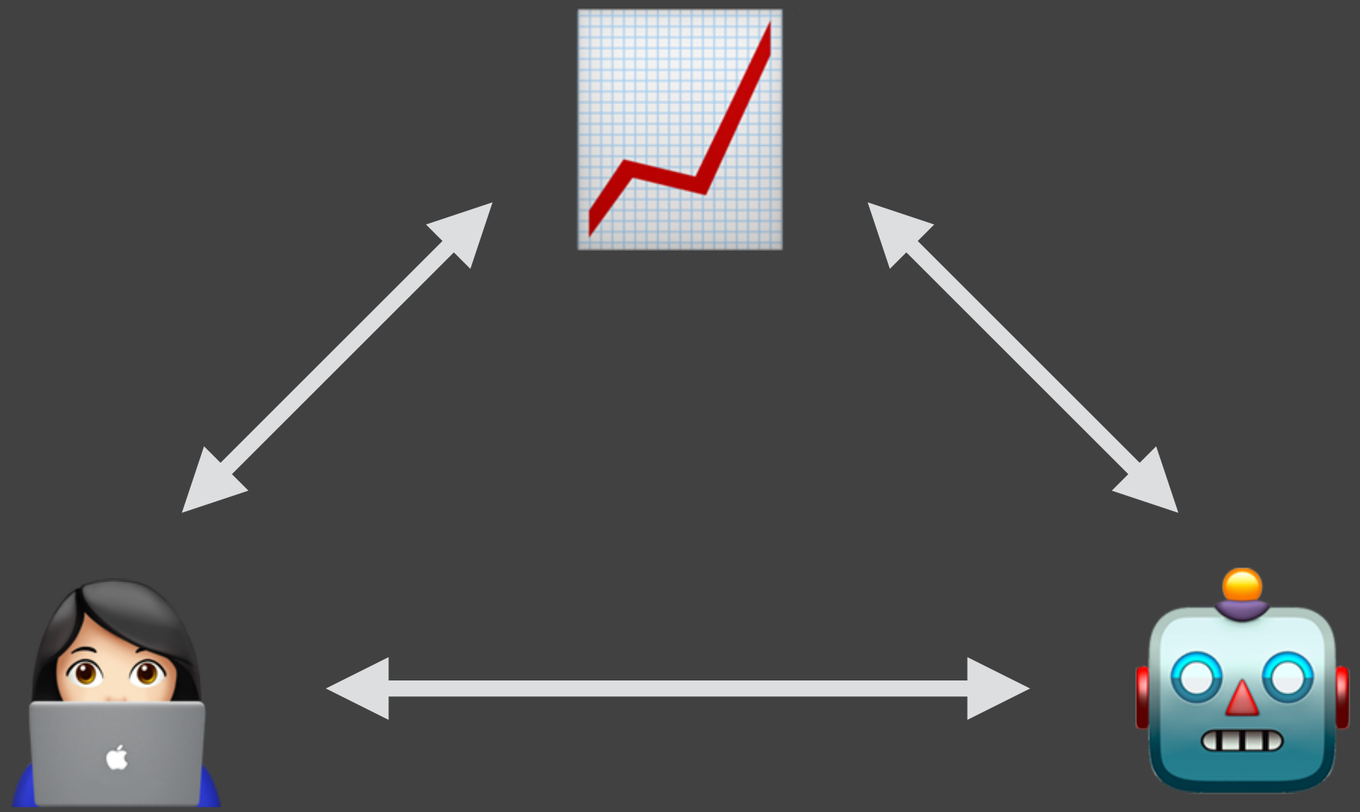
brownish →	☐ keep
brownish peaks → peaks	☐ keep
brownish NOUN → NOUN	☐ keep
ADJ NOUN→ NOUN	☐ keep
how many ADJ→ how many ADJ	☐ keep
how many ADJ NOUN → how many NOUN	☑ keep

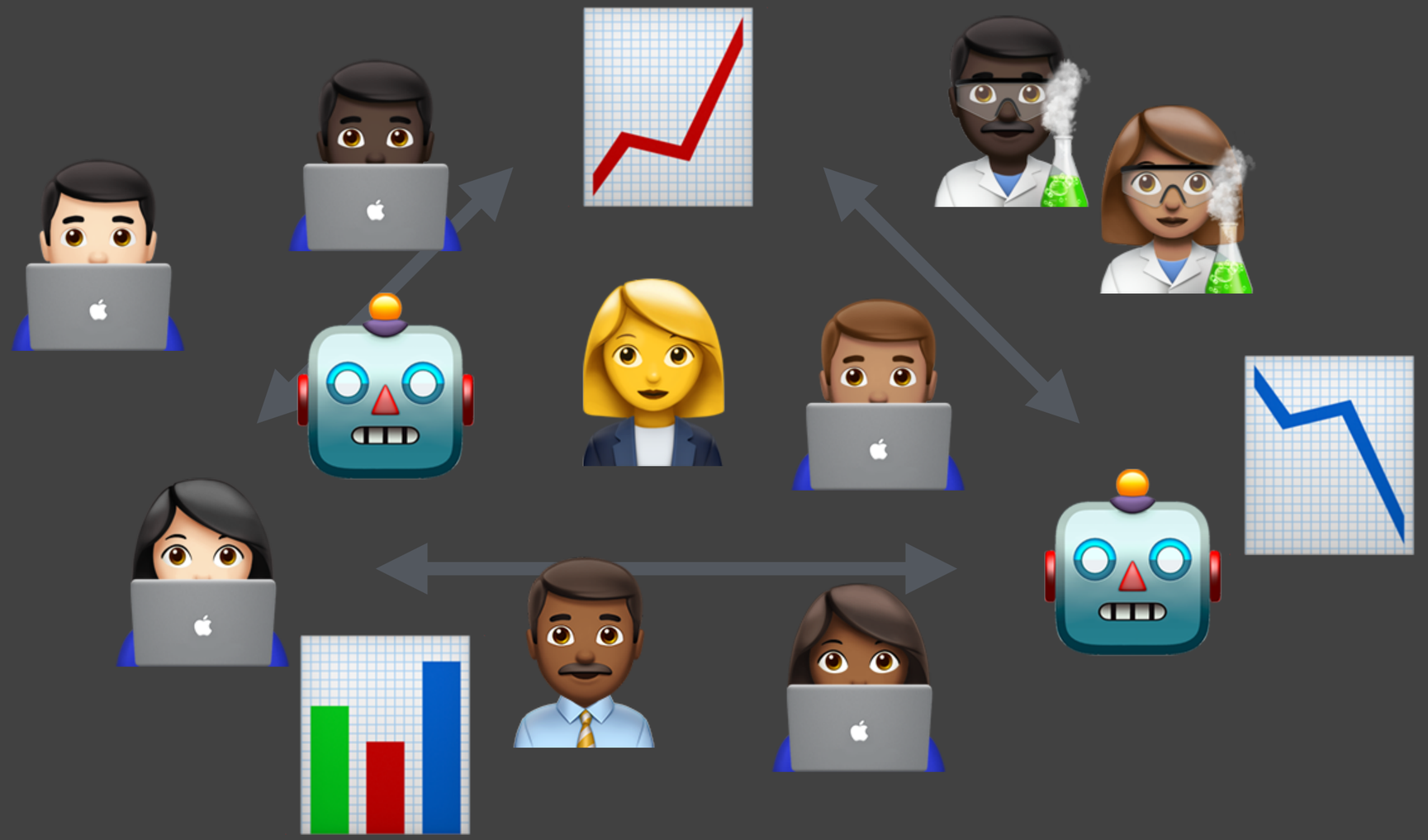
Tongshuang (Sherry) Wu, M. Ribeiro, J. Heer, D. Weld. ACL 2019


Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment


Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment


Acquisition
Cleaning
Integration
Exploration
Modeling
Evaluation
Deployment
Datasheets?
Annotation?
Inferential Safeguards?
Metadata?
Monitoring?
Model Management?
Governance?





idL

NSF

ISTC BIG DATA

DARPA

GORDON AND BETTY MOORE FOUNDATION

THE PAUL G. ALLEN FAMILY FOUNDATION

Data to the People?

Reflections on Trying to Help People Struggle Less with Data

Jeffrey Heer @jeffrey_heer
University of Washington