

# Big Data Analytics

Juliana Freire & Cláudio Silva  
Computer Science & Engineering  
Visualization, Imaging and Data Analysis Center (VIDA)  
Center for Data Science (CDS)

# Big Data: What is the Big deal?



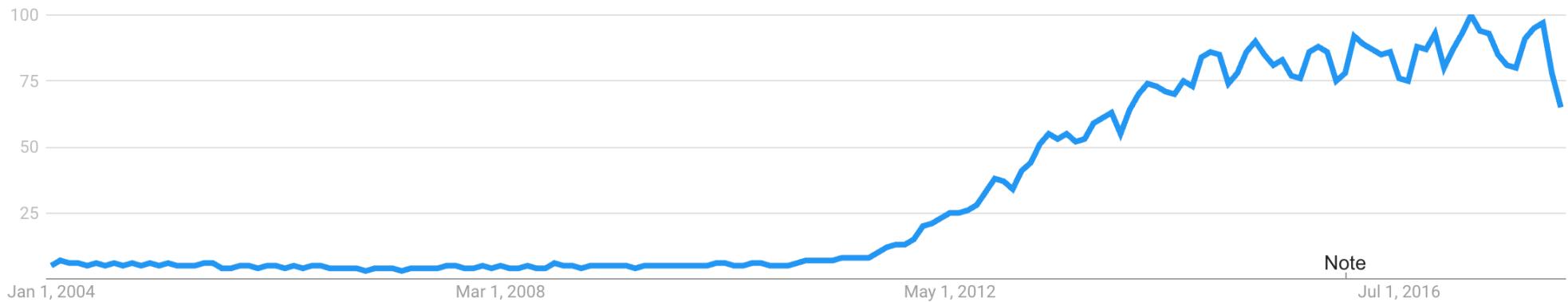
NYU

TANDON SCHOOL  
OF ENGINEERING



# Google Trends: Big Data

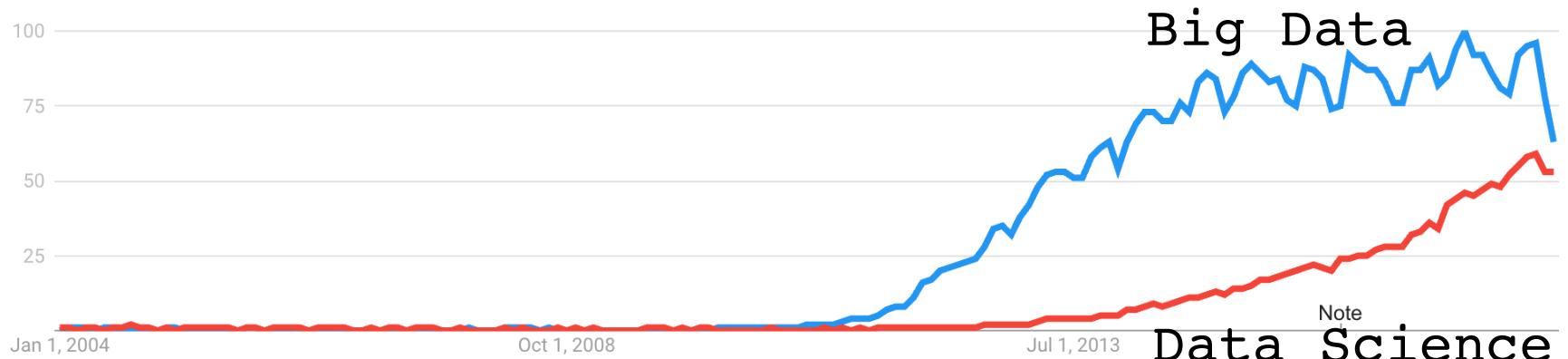
Interest over time 



<https://www.google.com/trends/explore?date=all&q=big%20data>

# Google Trends: Big Data vs. Data Science

Interest over time (?)



<https://www.google.com/trends/explore?date=all&q=data%20science,big%20data>

# Big Data: What is the Big deal?

- Many success stories
  - Google: many billions of pages indexed, products, structured data
  - Facebook: 2.07 billion users each month
  - Twitter: 330 million monthly active users, 500 million tweets/day
- This has changed society!



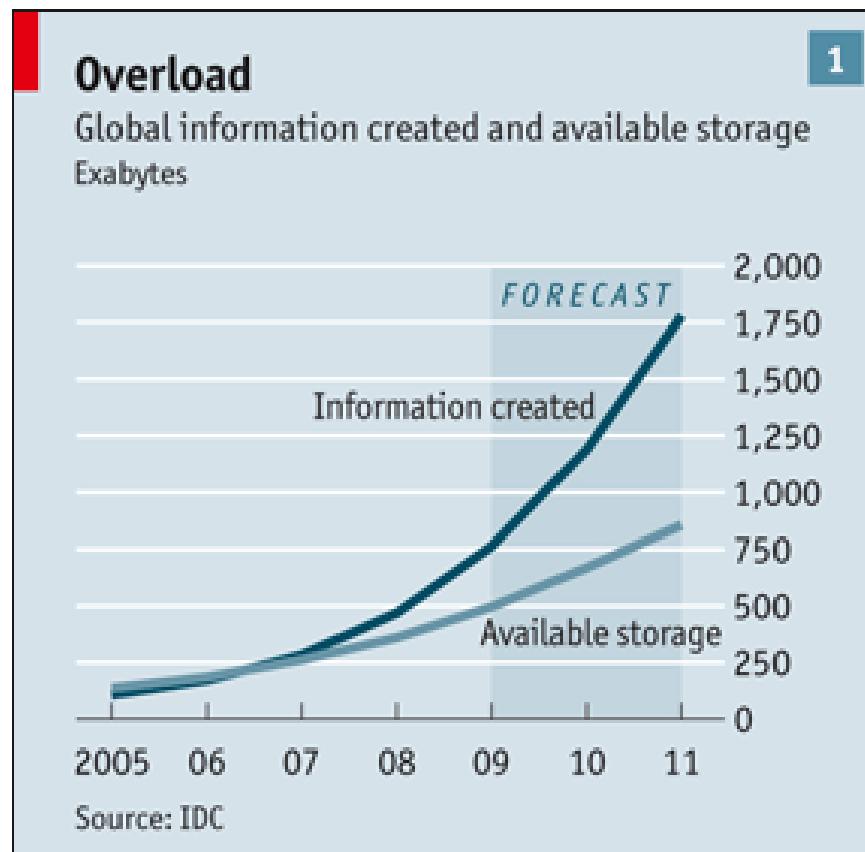
Google Search

I'm Feeling Lucky

<https://www.omnicoreagency.com/twitter-statistics>  
<https://www.omnicoreagency.com/facebook-statistics/>

# Data Volumes are Exploding

- We are producing more data than we are able to store (or analyze!)



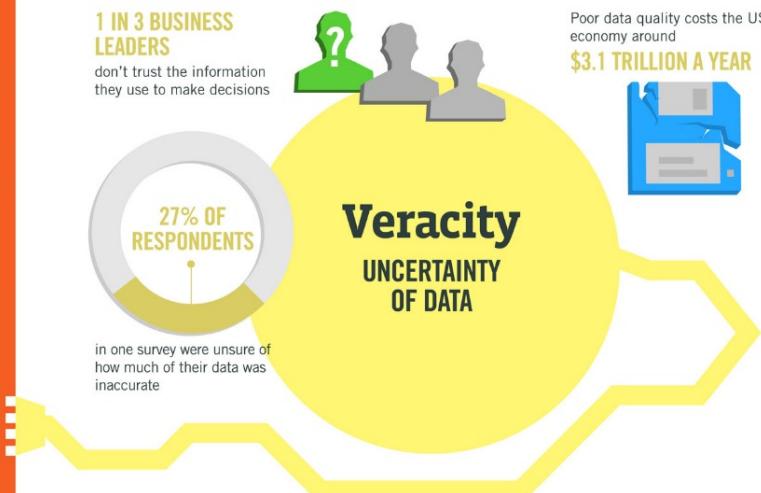
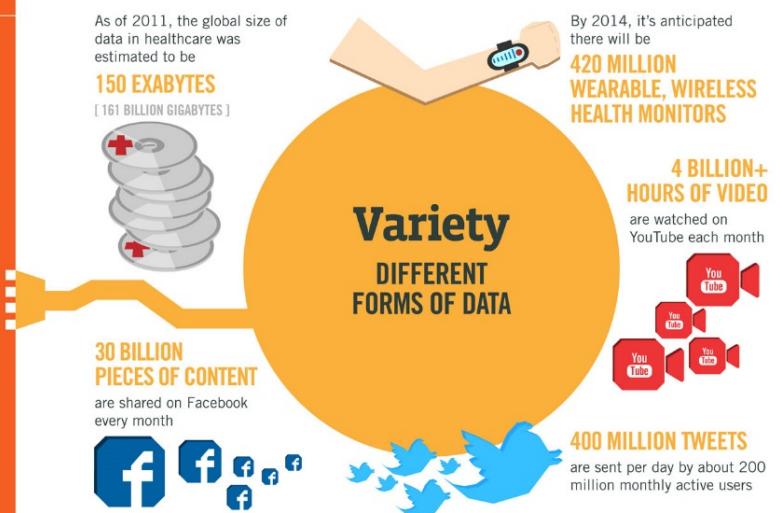
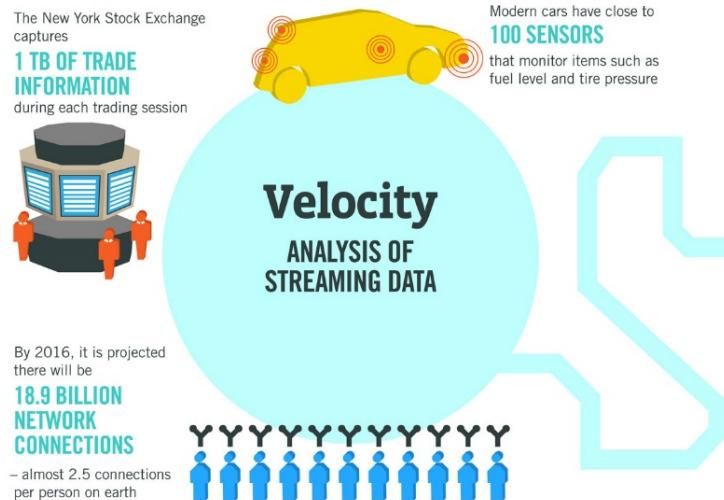
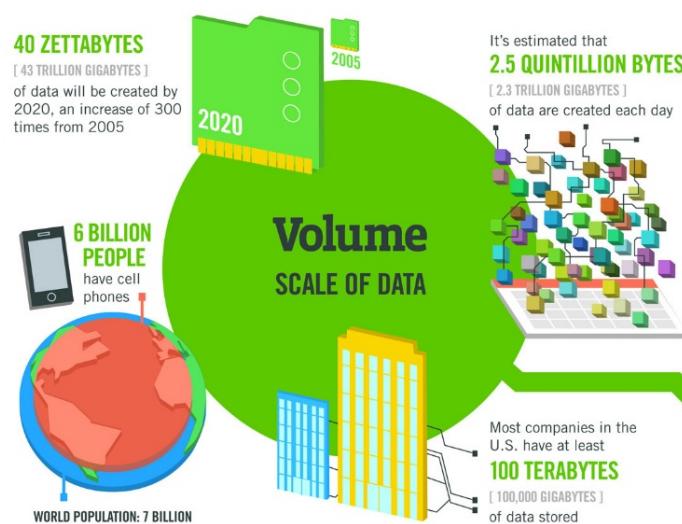
Economist, 2010

# New Data and Applications

---

- Search engines – large-scale crawls, indexing, many users consuming data
  - 20 PB a day circa 2008
- Social networks – many users producing data
  - 2.5 PB of user data + 15 TB/day circa 4/2009
- Urban data – lots of sensors and data, need to clean, integrate, and analyze data
  - NYC Open Data: 1,200+ data sets
- Online shopping – many users consuming and producing data
- Science: Petabytes of data generated each day, e.g., Australian radio telescopes, Large Hadron Collider, climate,...
- Many new and diverse requirements for data management, no one-size-fits-all solution!

# 4 Vs of Big Data

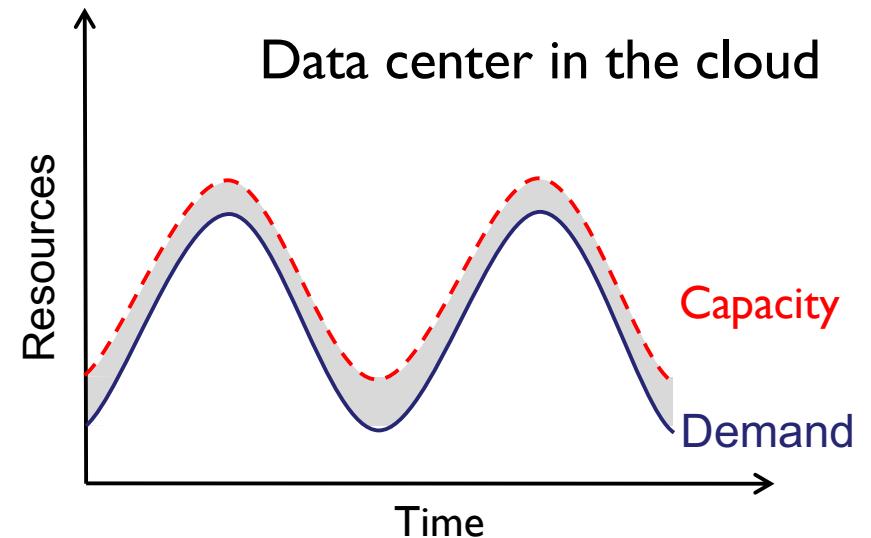
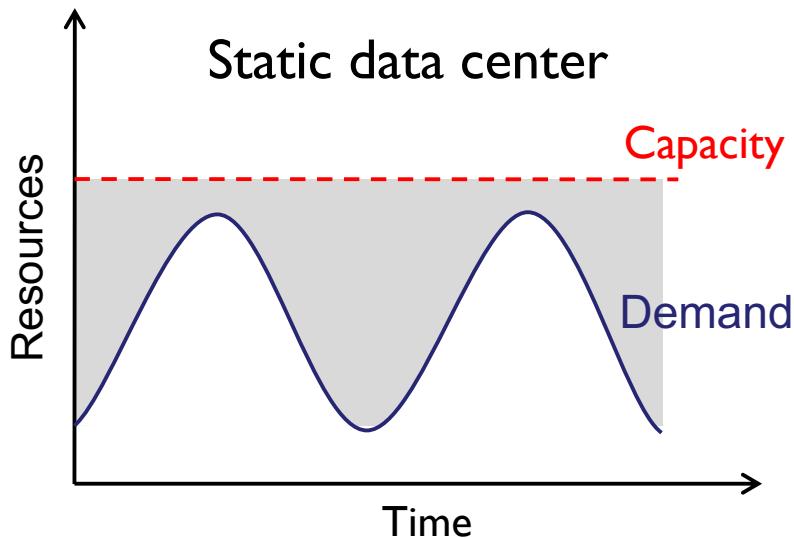


Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

# New Computing Infrastructure

- Meet the cloud!
- “[Hardware, Infrastructure, Platform] as a service”
- Utility Computing: pay-as-you-go computing
  - Illusion of infinite resources
  - No up-front cost
  - Fine-grained billing (e.g., hourly)

 Unused resources



# Warehouse Scale Computing

---

Google's data center in Oregon



**16 Million Nodes per building**



**NYU**

TANDON SCHOOL  
OF ENGINEERING

Agrawal et al., VLDB 2010 Tutorial

 **VIDA**  
VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Data Lifecycle

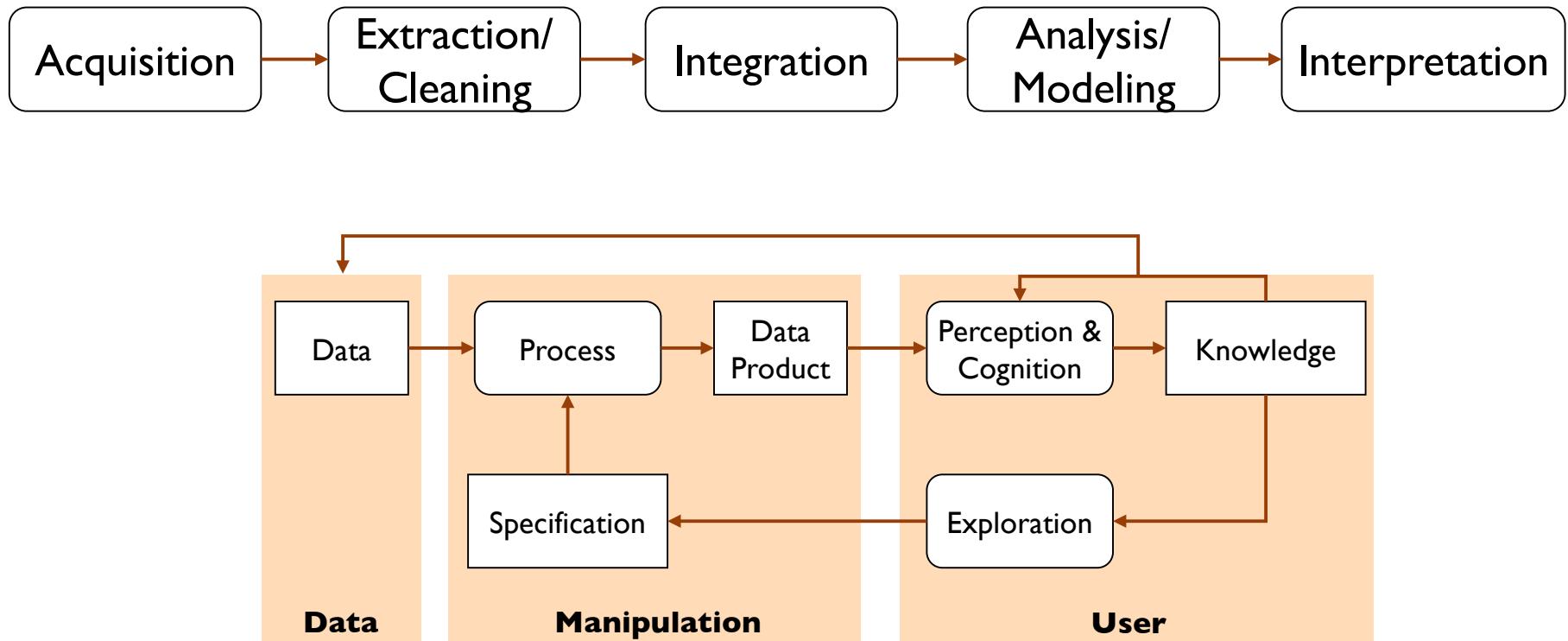
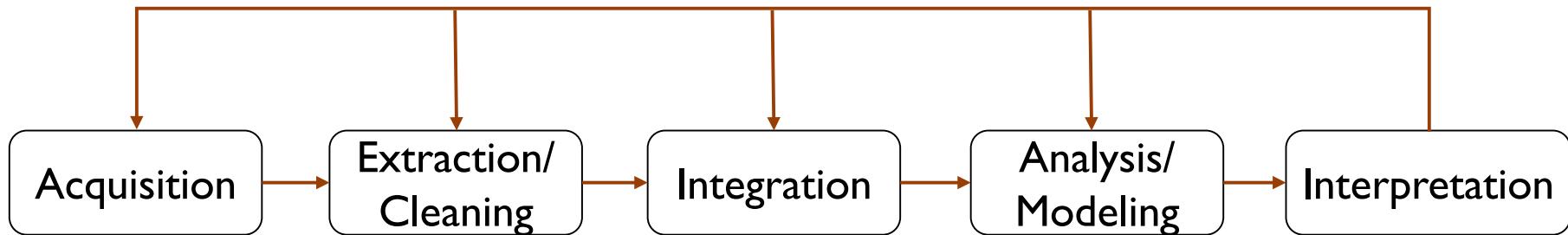


Figure modified from J. van Wijk, IEEE Vis 2005

# Big Data Lifecycle

---



- The 4 Vs lead to new challenges and opportunities
- Data discovery: how to find relevant data?
- Large-scale cleaning and integration: 10s-100s of data sets
- Interactive analysis: how to handle very large data? How to find interesting ‘features’?
- Explainability and transparency

# Outline

---

- Introduction to Programming with Big Data
  - Map Reduce vs. Parallel Databases
- Running Example: Big Urban Data
- Data Cleaning
  - Overview and Challenges
  - Cleaning the NYC Taxi Data: A Case Study
- Data exploration
  - Usability and Interactivity
  - Finding Interesting Features
  - Using Data to Discover and Explain Data
- Transparency and Reproducibility

# New Programming Infrastructure: Map Reduce

# MapReduce: Motivation

---

- Dean & Ghemawat, OSDI 2004

*“many ... at Google have implemented hundreds of special-purpose computations that process large amounts of raw data, such as crawled documents, web request logs, etc., to compute various kinds of derived data, such as inverted indices, ... summaries of the number of pages crawled per host, the set of most frequent queries in a day...”*

*Most such computations are conceptually straightforward... issues of how to parallelize the computation, distribute the data, and handle failures conspire to obscure the original simple computation with large amounts of complex code to deal with these issues.”*

# Parallelization Challenges

---

- How do we assign work units to workers?
- What if we have more work units than workers?
- What if workers need to share partial results?
- How do we aggregate partial results?
- How do we know all the workers have finished?
- What if workers die?
- What if hardware fails?

What is the common theme of all of these problems?

# Map Reduce: Big Ideas

---

- Abstraction
  - Hide system-level details from the developers
  - Developer specifies the computation that needs to be performed and the execution framework (“runtime”) handles actual execution
- Scale out: Large number of commodity low-end servers is more effective for data-intensive applications
  - 8 128-core machines vs. 128 8-core machines
- Cope with common failures
  - Automatic task restarts
  - Store files multiple times for reliability
- Move processing to data
- Avoid random access: organize computations into long streaming operations

# Map Reduce: Big Ideas

---

- Abstraction
  - Hide system-level details from the developers
  - Developer specifies the computation that needs to be performed and the execution framework (“runtime”) handles actual execution
- Scale out: Large number of commodity low-end servers is more effective for data intensive computations
  - 1 TB database containing  $10^{10}$  100 byte records
  - Random access: each update takes ~30ms (seek, read, write)
- Co-partitioned
  - Updating 1% of the records takes ~35 days
  - Sequential access: 100MB/s throughput
  - Reading the whole database and rewriting all the records, takes 5.6 hours
- Move processing to data
- Avoid random access: organize computations into long streaming operations

# Typical Big-Data Problem

---

- Iterate over a large number of records
- Extract something of interest from each record
- Sort and shuffle intermediate results
- Aggregate intermediate results
- Generate final output

Key idea in Map Reduce: provide a functional abstraction  
for these two operations

(Dean and Ghemawat, OSDI 2004)

# Typical Big-Data Problem in Map Reduce

---

- Iterate over a large number of records
- **Map:** Extract something of interest from each record  
**map**  $(k, v) \rightarrow \langle k', v' \rangle^*$

*There is one Map call for each  $(k, v)$  pair*

- Sort and shuffle intermediate results
- **Reduce:** Aggregate intermediate results  
**reduce**  $(k', \langle v' \rangle^*) \rightarrow \langle k', v'' \rangle^*$

*There is one Reduce per unique key  $k'$*

- Generate final output

*Structure remains the same, Map and Reduce functions change to fit the problem*

(Dean and Ghemawat, OSDI 2004)

# Word Count in Python

---

```
def word_count_dict(filename):
    """Returns a word/count dict for this filename."""
    # Utility used by count() and Topcount().
    word_count = {} # Map each word to its count
    input_file = open(filename, 'r')
    for line in input_file:
        words = line.split()
        for word in words:
            word = word.lower()
            # Special case if we're seeing this word for the first time.
            if not word in word_count:
                word_count[word] = 1
            else:
                word_count[word] = word_count[word] + 1
    input_file.close() # Not strictly required, but good form.
    return word_count
```

<https://github.com/mlafeldt/google-python-class/blob/master/basic/solution/wordcount.py>

# Word Count in Map Reduce

---

Map(String docid, String text):  
for each word w in text:  
    Emit(w, 1);

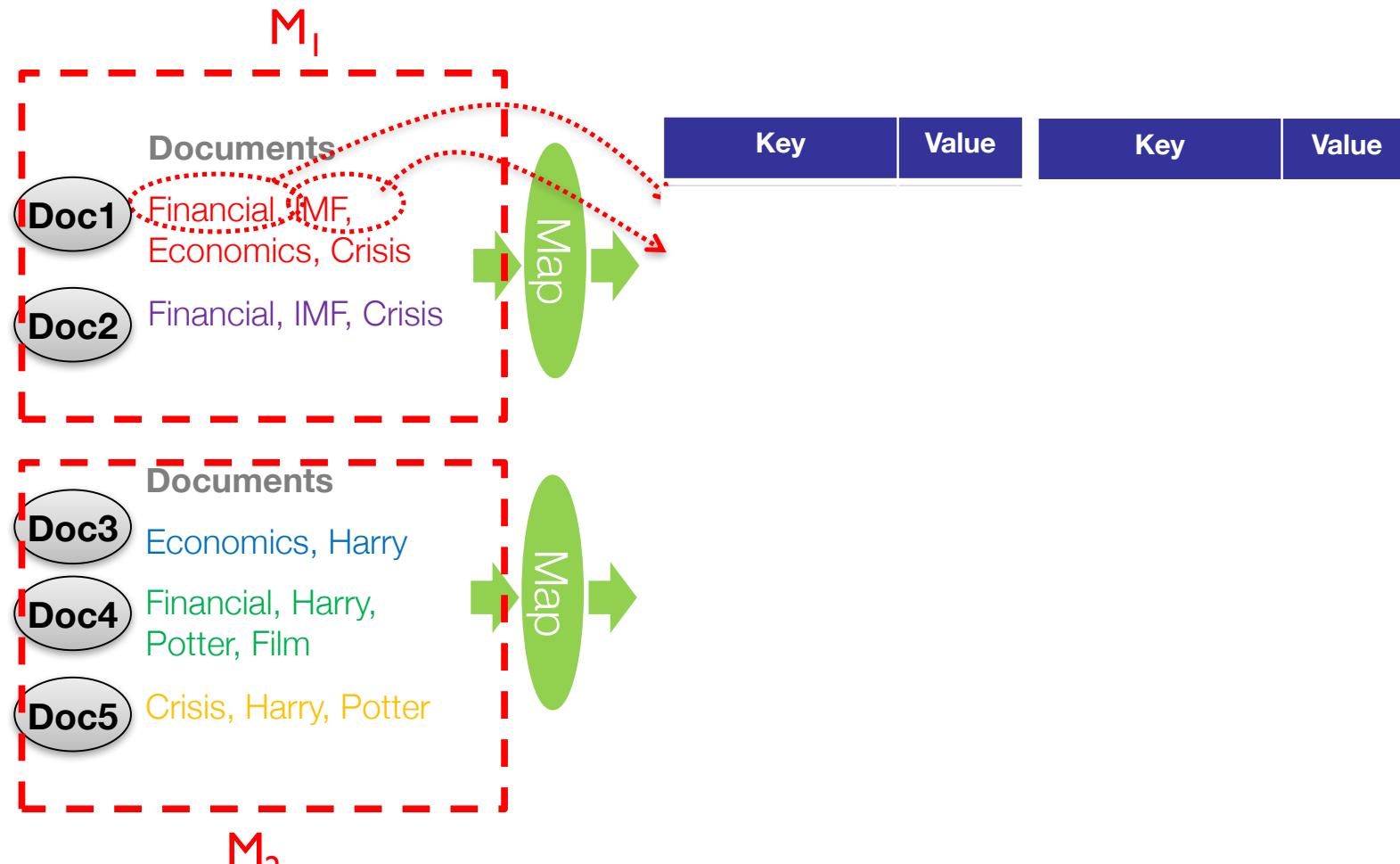
$\text{map} (docid, text) \rightarrow [(word, 1)]$

Reduce(String term, Iterator<Int> values):

int sum = 0;  
for each v in values:  
    sum += v;  
Emit(term, value);

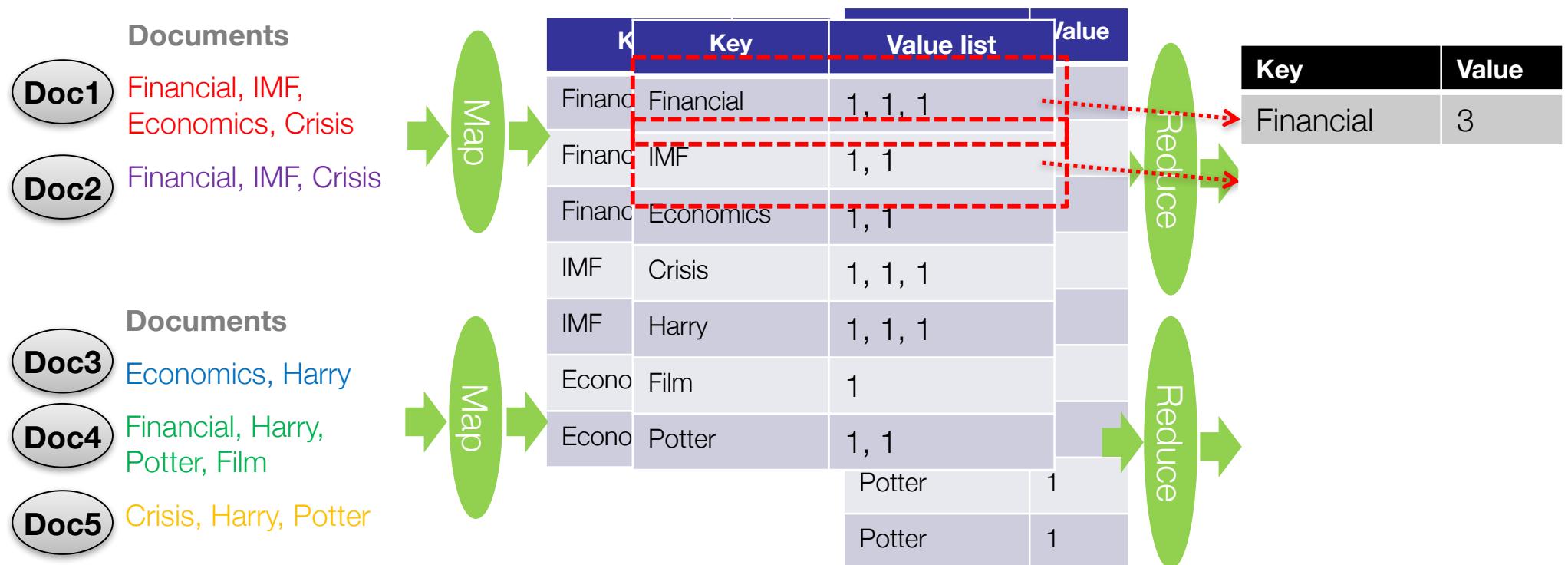
$\text{reduce} (word, [v_1, v_2, v_3]) \rightarrow [(word, count)]$

# Word Counting with MapReduce



# Word Counting with MapReduce

© Kyuseok Shim (VLDB 2012 TUTORIAL)



Before reduce functions are called,  
for each distinct key, a list of associated values is  
generated



NYU

TANDON SCHOOL  
OF ENGINEERING



VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Challenge Question

---

What is the average time spent per URL?

```
SELECT url, AVG(time)  
FROM visits  
GROUP BY url
```

URL	User	Time
<a href="http://www.google.com">www.google.com</a>	Juliana	39
<a href="http://www.twitter.com">www.twitter.com</a>	DJT	1456
www.twitter.com	Mary	300
<a href="http://www.cnn.com">www.cnn.com</a>	John	123

How would you write this query in Map Reduce?

- Map over tuples, emit time, keyed by url
- Framework *automatically groups values by keys*
- Compute average in reducer

# Designing Algorithms for Map Reduce

---

- Need to adapt to a restricted model of computation
- Goals
  - Scalability: adding machines will make the algorithm run faster
  - Efficiency: resources will not be wasted
- The translation some algorithms into MapReduce isn't always obvious
- But there are useful *design patterns* that can help
- All complexities related to parallelism and fault tolerance are automatically handled by the environment!

See **Data-Intensive Text Processing with MapReduce**  
Jimmy Lin and Chris Dyer, 2010

# Platforms for Large-Scale Data Analysis

---

- **Parallel DBMS technologies**

- Proposed in the late eighties
- Matured over the last two decades
- Multi-billion dollar industry: Proprietary DBMS Engines intended as Data Warehousing solutions for very large enterprises

- **Map Reduce**

- Data-parallel *programming model*
- Works on the cloud
- Pioneered by Google, and popularized by Yahoo! (open-source Hadoop)
- [Dean et al., OSDI 2004, CACM Jan 2008, CACM Jan 2010]

Agrawal et al., VLDB 2010 Tutorial

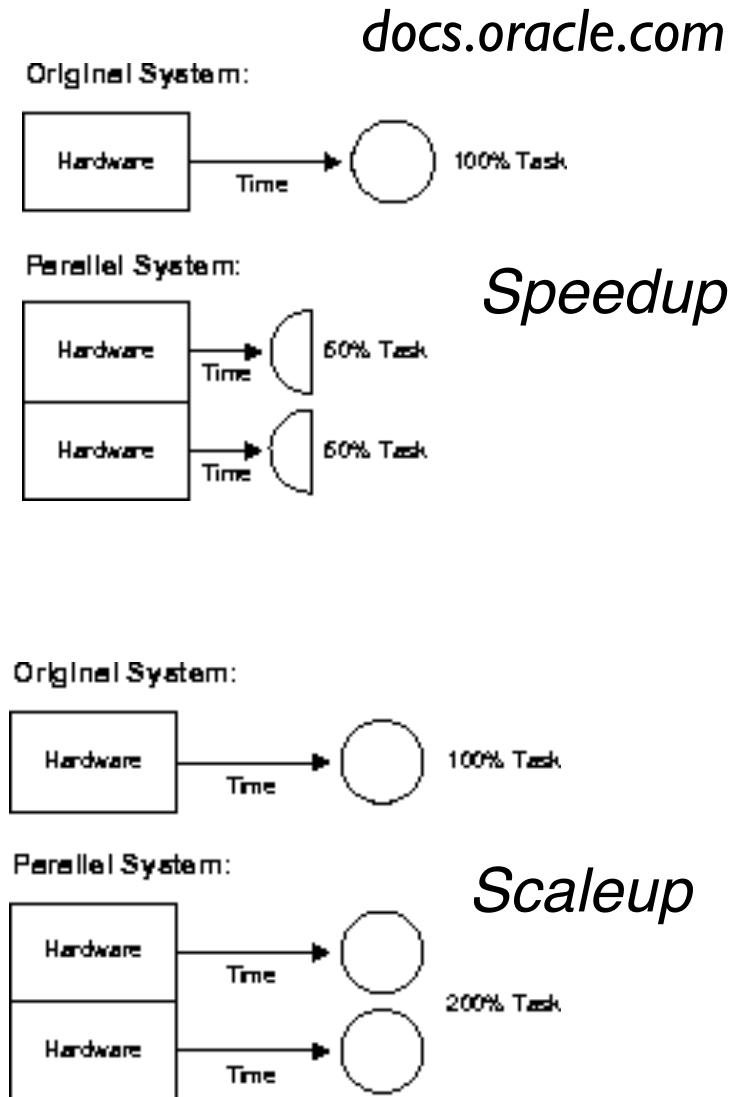
# Parallel DBMS technologies

The image is a collage of screenshots from four different company websites, each showcasing a different parallel DBMS technology:

- Oracle:** Shows the Oracle Data Warehouse page with the headline "Oracle is a Leader in Data Warehousing". It includes a mention of the Gartner Magic Quadrant for Data Management Solutions for Analytics.
- IBM:** Shows the IBM Analytics page under the Technology section, specifically the Data management platform > Data warehousing category. It features a 3D diagram illustrating data flow between a server, a bar chart, and a laptop.
- Microsoft:** Shows the SQL Server 2016 page, highlighting its use for building intelligent, mission-critical applications. It includes a call-to-action button "Watch the overview video >".
- VIDA:** Shows the VIDA logo and the tagline "VISUALIZATION IMAGING AND DATA ANALYSIS CENTER".

# Parallel DBMSs

- Aim to improve performance by executing operations in parallel
- Benefit: easy and cheap to scale
  - Add more nodes, get higher speed
  - Reduce the time to run queries
  - Higher transaction rates
  - Ability to process more data
- Challenges: minimize overheads and efficiently deal with contention



# Parallel DBMS technologies

---

- Research Projects: Gamma, Grace – many papers!
- Popularly used for more than two decades
  - Commercial: Multi-billion dollar industry
- All the advantages of relational databases
  - Relational Data Model
  - Familiar SQL interface: declarative queries hide complexities from users
  - Advanced query optimization
  - Indexing
  - Data integrity and security
- *Well understood and studied*
- *Very reliable, easy for the end user*
- **Very expensive**

Modified from Agrawal et al., VLDB 2010 Tutorial

# The Debate Starts...

This is Google's cache of [http://databasecolumn.vertical.com/2008/01/mapreduce\\_a\\_major\\_step\\_back.html](http://databasecolumn.vertical.com/2008/01/mapreduce_a_major_step_back.html). It is a snapshot of the page as it appeared on Sep 27, 2009 00:24:13 GMT. The [current page](#) could have changed in the meantime. [Learn more](#)

These search terms are highlighted: **search** These terms only appear in links pointing to this page: [hi en&safe off&q](#)

[Text-only version](#)

## The Database Column

A multi-author blog on database technology and innovation.

## MapReduce: A major step backwards

By David DeWitt on January 17, 2008 4:20 PM | [Permalink](#) | [Comments \(44\)](#) | [TrackBacks \(1\)](#)

*[Note: Although the system attributes this post to a single author, it was written by David J. DeWitt and Michael Stonebraker]*

On January 8, a Database Column reader asked for our views on new distributed database research efforts, and we'll begin here with our views on [MapReduce](#). This is a good time to discuss it, since the recent trade press has been filled with news of the revolution of so-called "cloud computing." This paradigm entails harnessing large numbers of (low-end) processors working in parallel to solve a computing problem. In effect, this suggests constructing a data center by lining up a large number of "jelly beans" rather than utilizing a much smaller number of high-end servers.

For example, IBM and Google have announced plans to make a 1,000 processor cluster available to a few select universities to teach students how to program such clusters using a software tool called MapReduce [1]. Berkeley has gone so far as to plan on teaching their freshman how to program using the MapReduce framework.

As both educators and researchers, we are amazed at the hype that the MapReduce proponents have spread about how it represents a paradigm shift in the development of scalable, data-intensive applications. MapReduce may be a good idea for writing certain types of general-purpose computations, but to the database community, it is:

1. A giant step backward in the programming paradigm for large-scale data intensive applications
2. A sub-optimal implementation, in that it uses brute force instead of indexing
3. Not novel at all -- it represents a specific implementation of well known techniques developed nearly 25 years ago
4. Missing most of the features that are routinely included in current DBMS
5. Incompatible with all of the tools DBMS users have come to depend on



# The Debate Continues...

---

- A comparison of approaches to large-scale data analysis. Pavlo et al., SIGMOD 2009
- <http://dl.acm.org/citation.cfm?id=1559865>;  
<http://dsl.serc.iisc.ernet.in/~course/DBMS/papers/benchmarks-sigmod09.pdf>
- *Parallel DBMS beats MapReduce by a lot!*
- Many were outraged by the comparison
- MapReduce: A Flexible Data Processing Tool. Dean and Ghemawat, CACM 2010
  - Pointed out inconsistencies and mistakes in the comparison
  - <http://cacm.acm.org/magazines/2010/1/55744-mapreduce-a-flexible-data-processing-tool/fulltext>
- MapReduce and Parallel DBMSs: Friends or Foes? Stonebraker et al., CACM 2010
  - Toned down claims...
  - <http://cacm.acm.org/magazines/2010/1/55743-mapreduce-and-parallel-dbmss-friends-or-foes/fulltext>

# Map Reduce: Data Analytics in the Cloud

---

- Programming model that supports scalability for large data volumes: Divide-And-Conquer (i.e., *data partitioning*)
  - It is not a database!
- Cost-efficiency:
  - Commodity nodes (cheap, but unreliable)
  - Commodity network
  - Automatic fault-tolerance (fewer admins)
  - *Easy to use* (fewer programmers)

# Complex Functions

---

- MapReduce was designed for *complex* tasks that manipulate diverse data:
  - Extract links from Web pages and aggregating them by target document
  - Generate inverted index files to support efficient search queries
  - Process all road segments in the world and rendering map images
- These data do not fit well in the relational paradigm
  - SQL is not Turing-complete!
- Parallel databases support
  - Tabular data
  - SQL operations
  - User Defined Functions (UDF), but these have limitations

# Heterogeneous Systems

---

- MapReduce can **support different storage backends** and it can be used to combine data from different sources
  - Production MapReduce environments use a plethora of storage systems: files, RDBMS, Bigtable, column stores
  - <http://static.googleusercontent.com/media/research.google.com/en//archive/bigtable-osdi06.pdf>
- Parallel databases require all data to be pre-loaded and in the **same system**
  - Would you use a ParDB to load Web pages retrieved by a crawler and build an inverted index?

# Structured Data and Schemas

- Schemas were key to the success of database: enable structured queries, efficiency, data sharing
- The original version of Hadoop had no support for schemas!
- Google's MapReduce implementation supports the Protocol Buffer format: high-level language is used to describe the input and output types
  - Platform-neutral, extensible mechanism for serializing structure data

<https://developers.google.com/protocol-buffers/>

- Compiler-generated code hides the details of encoding/decoding data
- Use optimized binary representation --- compact and faster to encode/decode; huge performance gains

# Fault Tolerance

---

- Pull model is necessary to provide fine-grained fault tolerance
- It may lead to the creation of many small files on disk
- Use implementation tricks to mitigate these costs
  - Keep this in mind when writing your MapReduce programs!

# Architectural Elements: ParDB vs. MR

---

- Schema support:
  - Relational paradigm: rigid structure of rows and columns
  - Flexible structure, but need to write parsers and challenging to share results
  - Use protocol buffers!
- Indexing
  - B-trees to speed up access
  - No built-in indexes --- programmers must code indexes
  - Can use MR-based storage systems
- Programming model
  - Declarative, high-level language
  - Imperative, write programs
  - SQL and other higher-level languages are supported
- Data distribution
  - Use knowledge of data distribution to automatically optimize queries
  - Programmer must optimize the access

# Architectural Elements: ParDB vs. MR

---

- Execution strategy and fault tolerance:
  - Pipeline operators (push), failures dealt with at the transaction level – if a single node fails, the entire query must be restarted
  - Write intermediate files (pull), provide fault tolerance
  - Can use Spark instead of MR

# Map Reduce: Takeaway

---

- MapReduce's data-parallel programming model hides complexity of distribution and fault tolerance
- Principal philosophies:
  - *Make it scale*, so you can throw hardware at problems
  - *Make it cheap*, saving hardware, programmer and administration costs (but requiring fault tolerance)
- Map Reduce is not suitable for all problems, but when it works, it may save you a lot of time

# Beyond Map Reduce

---

- Spark: leverage main memory to support streaming + analytics + iterative computation “Spark can be as much as 10 times faster than MapReduce <https://spark.apache.org/>
- DB features added to the ecosystem for batch processing and up to 100 times faster for in-memory analytics”
  - Shark: large-scale data warehouse system
    - SQL API <https://amplab.cs.berkeley.edu/>
    - Now <https://spark.apache.org/sql/>
  - HadoopDB: hybrid of DBMS and MapReduce technologies that targets analytical workloads
  - Apache Hbase: random, real-time read/write access to Big Data
  - Apache Accumulo (<https://accumulo.apache.org>): distributed key/value store – similar to BigTable
  - ElasTraS: An elastic, scalable, and self-managing transactional database for the cloud

# Map Reduce Implementations

---

- Google has a proprietary implementation in C++
  - Bindings in Java, Python
- Hadoop is an open-source implementation in Java
  - Development led by Yahoo, used in production
  - Now an Apache project
  - Rapidly expanding software ecosystem
- Lots of custom research implementations
  - For GPUs, cell processors, etc.

# Map Reduce: References

---

- You can get free time on AWS!
- Several online tutorials
- <http://hadoop.apache.org>
- Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large Clusters. Usenix OSDI, 2004:  
[http://www.usenix.org/events/osdi04/tech/full\\_papers/dean/dean.pdf](http://www.usenix.org/events/osdi04/tech/full_papers/dean/dean.pdf)
- David DeWitt, Michael Stonebraker, MapReduce: A major step backwards,  
[craighenderson.blogspot.com](http://craighenderson.blogspot.com)  
[http://scienceblogs.com/goodmath/2008/01/databases\\_are\\_hammers\\_mapreduc.php](http://scienceblogs.com/goodmath/2008/01/databases_are_hammers_mapreduc.php)
- Mining of Massive Data Sets (version 2.1), by Anand Rajaraman, Jure Leskovec and Jeff Ullman. <http://www.mmds.org>
- Data-Intensive Text Processing with MapReduce, by Jimmy Lin and Chris Dyer. <http://lintool.github.com/MapReduceAlgorithms/index.html>

Free!

# Running Example: Big Urban Data

# Urban Data: What is the Big deal?

---

- Cities are the loci of economic activity
- 50% of the world population lives in cities, by 2050 the number will grow to 70%
- Growth leads to problems, e.g., transportation, environment and pollution, housing, infrastructure
- Good news: Lots of data being collected from traditional and *unsuspecting* sensors

# Data Exhaust from Cities

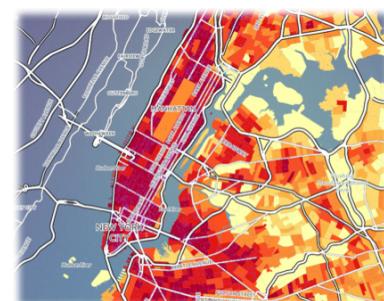
*Infrastructure*

Condition,  
Operations



*Environment*

Meteorology, pollution,  
noise, flora, fauna



*People*

Relationships,  
economic  
activities,  
health, nutrition,  
opinions,...

**flickr**

**twitter**



**NYU**

TANDON SCHOOL  
OF ENGINEERING

**VIDA**

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Data Exhaust from Cities

The screenshot shows the Beijing Open Data Portal homepage. It features a large banner for the 'Tourism Scenic Spot Comfort Index' (游览舒适度指数) with a smiley face icon and a bar chart. Below the banner, there's a search bar and navigation links for '首页', '数据', '接口', '定向数据', '应用', '工具', and '互动交流'. A '网站统计' button is also present. The main content area displays a map of Beijing with scenic spots highlighted.

The screenshot shows the Paris Data portal homepage. It features a header with 'PARISDATA' and 'MAIRIE DE PARIS'. Below the header, there are links for 'Les données', 'L'API', 'La licence', 'La démarche', and 'Cartographie'. The main content area includes a 'Bienvenue sur Paris Data' section and a 'Disponibilité des Autolib aux stations' section with a map showing car availability. There are also sections for 'Активный гражданин', 'Архитектура и строительство', 'Безопасность', 'Государственные услуги', and 'Дороги и транспорт'. On the right side, there's a sidebar with tweets about the portal and a 'Fabrice Benaut' profile.

The screenshot shows the NYC Open Data portal homepage. It features a 'NYC OpenData' logo and navigation links for 'Home', 'Data', 'About', 'Learn', and 'Alerts'. Below the navigation, there's a section titled 'Yokers' with a question about finding public Wi-Fi. A search bar at the bottom says 'Search Open Data for things like 311, Build...' and a 'Translate' button is visible.

## Dados em Destaque

Three cards are displayed under the 'Dados em Destaque' heading:

- 1746 - Histórico - Aplicativo Móvel**: 1746 - Histórico - Aplicativo Móvel
- Frequência e índice de aprovação escolares**: Frequência e índice de aprovação das escolas, por ano letivo.
- Bairros do Rio de Janeiro**: Conjunto de dados com uma lista dos bairros do Rio de Janeiro, contendo apenas o nome do bairro e um ponto de...

# Urban Data: Success Stories



**OneBusAway**  
Serving up fresh real-time transit information for the  
region.

<http://onebusaway.org>

- Real-time arrival predictions
- 94% reported increased or greatly increased satisfaction with public transit
- Significant decrease in actual wait time per user, and an even greater decrease in *perceived* wait time

- 78% of riders reported increased walking – a significant public health benefit

Benefit residents

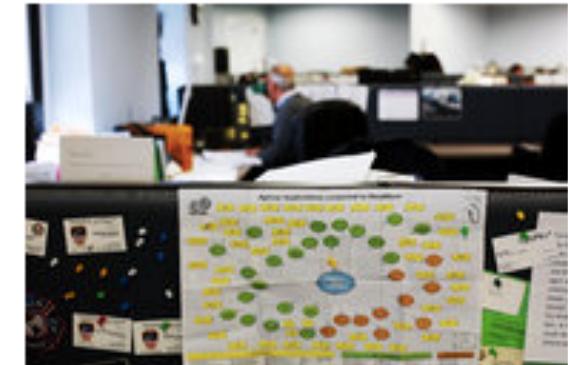
# Urban Data: Success Stories

- NYC gets 25,000 illegal complaints a year and **inspectors** to handle major fire hazards, cauldrons of crime, drugs, disease, and pest infestation
- Data-driven approach
  1. Integrated information from 19 different agencies that provided indication of issues in buildings, e.g., late taxes, foreclosure proceedings, service cuts, ambulance visits, rodent infestation, crime
  2. Compared with 5 years of fire data
  3. Created a prediction system
- Result: hit rate for inspections went from 13% to 70%



Todd Heisler/The New York Times  
Michael Flowers, right, oversees a small group of tech-savvy and civic-minded statisticians working across from City Hall.

[Enlarge This Image](#)



Todd Heisler/The New York Times  
"All we do," Mr. Flowers said, is "process massive amounts of information and use it to do things more effectively."

Make City more efficient

# Urban Data: What is hard?

## Infrastructure



Condition, operations

## Environment



Meteorology, pollution,  
noise, flora, fauna

## People



Relationships,  
economic activities, health,  
nutrition, opinions, ...

- City components interact in complex ways over space and time
- Need to analyze the city *data exhaust* to understand these interactions



NYU

TANDON SCHOOL  
OF ENGINEERING

# Urban Data: What is hard?

- There are lots of data, but how can we **discover** what is relevant for a given question/analysis?
- Data are **dirty and heterogeneous**
- **Data analysis and modeling** is challenging even for skilled data scientists

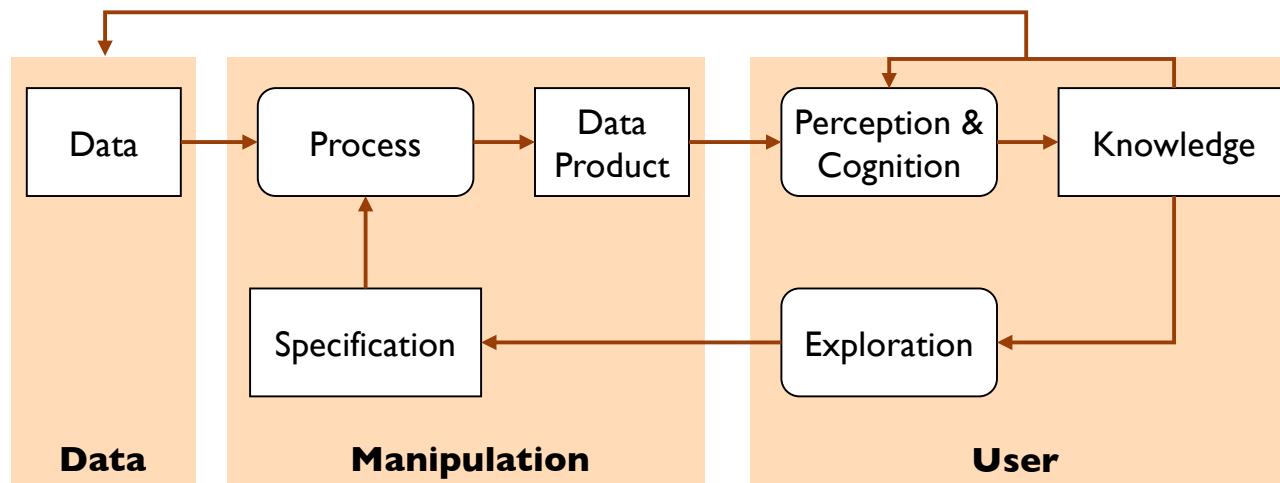
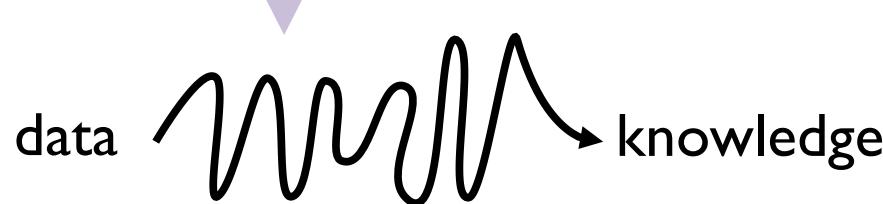
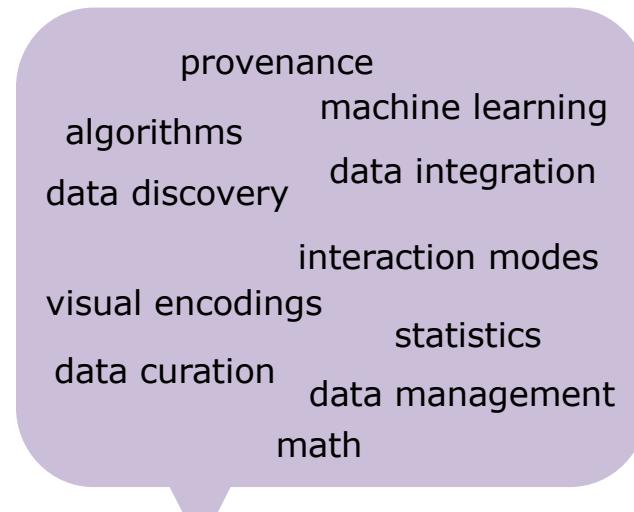


Figure modified from J. van Wijk, IEEE Vis 2005

# Urban Data: What is hard?

- There are lots of data, but how can we **discover** what is relevant for a given question/analysis?
- Data are **dirty and heterogeneous**
- **Data analysis and modeling** is challenging even for skilled data scientists



# Urban Data Analysis: Desiderata

---

- Scalable tools and techniques that help *domain experts* find, clean, integrate, *interactively* explore and explain data
- Cater to different kinds of users with little or no CS training
- *Automate* tedious tasks as much as possible
- Guide users in the exploration process

Data analysis for all!

# Big Data Analysis: Desiderata

---

- Scalable tools and techniques that help *domain experts* find, clean, integrate, *interactively* explore and explain data
- Cater to different kinds of users with little or no CS training
- *Automate* tedious tasks as much as possible
- Guide users in the exploration process

Data analysis for all!

Skip to outline

# Open Urban Data (as of 2014)

- Study: 20 cities in North America, 9,000 data sets
- Investigated
  - Nature of the data
  - Opportunities for integration

*“People are tribal, but data doesn’t care”*

Mike Flowers

[Barbosa et al., Big Data 2014]



Downloaded from online��lebertpub.com by 108.29.63.241 on 09/20/14. For personal use only.

## STRUCTURED OPEN URBAN DATA: Understanding the Landscape

Luciano Barbosa,<sup>1</sup> Kien Pham,<sup>2</sup> Claudio Silva,<sup>2,3</sup>  
Marcos R. Vieira,<sup>1</sup> and Juliana Freire<sup>2,3</sup>

### Abstract

A growing number of cities are now making urban data freely available to the public. Besides promoting transparency, these data can have a transformative effect in social science research as well as in how citizens participate in governance. These initiatives, however, are fairly recent and the landscape of open urban data is not well known. In this study, we try to shed some light on this through a detailed study of over 9,000 open data sets from 20 cities in North America. We start by presenting general statistics about the content, size, nature, and popularity of the different data sets, and then examine in more detail structured data sets that contain tabular data. Since a key benefit of having a large number of data sets available is the ability to fuse information, we investigate opportunities for data integration. We also study data quality issues and time-related aspects, namely, recency and change frequency. Our findings are encouraging in that most of the data are structured and published in standard formats that are easy to parse; there is ample opportunity to integrate different data sets; and the volume of data is increasing steadily. But they also uncovered a number of challenges that need to be addressed to enable these data to be fully leveraged. We discuss both our findings and issues involved in using open urban data.

### Introduction

FOR THE FIRST TIME IN HISTORY, more than half of the world’s population lives in urban areas<sup>1</sup>; in a few decades, the world’s population will exceed 9 billion, 70% of whom will live in cities. The exploration of urban data will be essential to inform both policy and administration, and enable cities to deliver services effectively, efficiently, and sustainably while keeping their citizens safe, healthy, prosperous, and well-informed.<sup>2–4</sup>

While in the past, policymakers and scientists faced significant constraints in obtaining the data needed to evaluate their policies and practices, recently there has been an explosion in the volume of open data. In an effort to promote transpar-

ency, many cities in the United States and around the world are publishing data collected by their governments (see, e.g., refs.<sup>5–8</sup>).

Having these data available creates many new opportunities. In particular, while individual data sets are valuable, by integrating data from multiple sources, the integrated data are often more valuable than the sum of their parts. The benefits of integrating city data have already led to many success stories. In New York City (NYC), by combining data from multiple agencies and using predictive analytics, the city increased the rate of detecting dangerous buildings, as well as improved the return on the time of building inspectors looking for illegal apartments.<sup>2</sup> Policy changes have also been triggered by studies that, for example, showed correlations

<sup>1</sup>IBM Research, Rio de Janeiro, Brazil.

<sup>2</sup>Department of Computer Science and Engineering, NYU School of Engineering, Brooklyn, New York.

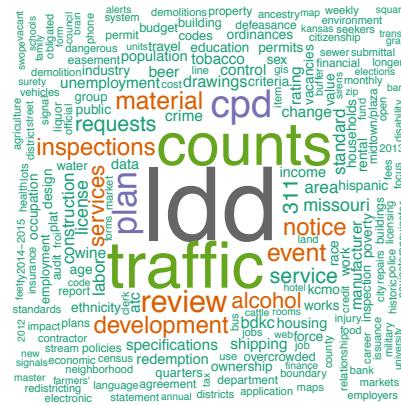
<sup>3</sup>NYU Center for Urban Science and Progress, Brooklyn, New York.

# Some Findings

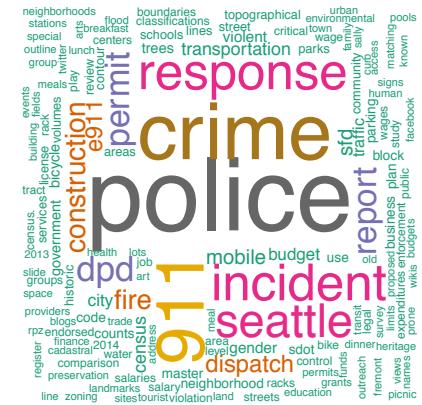
- 75% of the data sets are available in tabular formats, e.g., CSV: *ability to pose ‘complex’ queries and re-use data cleaning/integration techniques*
  - Many topics are covered: potential to connect different data



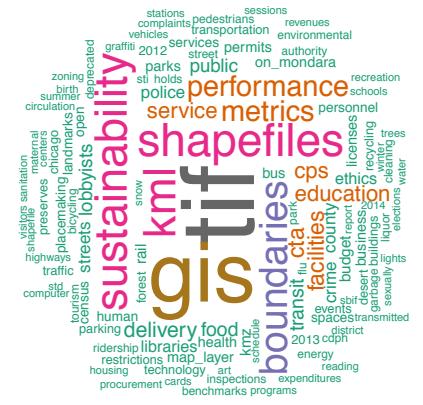
(a) NYC



(b) Kansas City



(c) Seattle



(d) Chicago

# Some Findings

- Most data are available in tabular formats, e.g., CSV
- Many topics are covered
- Number of data sets is growing
  - In 2013, more data sets were added than in the 3 previous years combined
- *Data are small:* 70GB for all cities
  - Compare against 1 year of taxi data: 50GB/year
- There are big and small tables



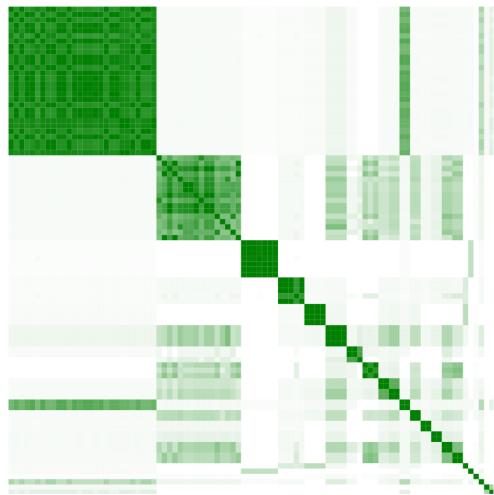
No. of records	Percentage of total
0–1K	65.3
1K–10K	17.0
10K–100K	11.7
100K–1M	5.5
1M–10M	0.3

# Some Findings

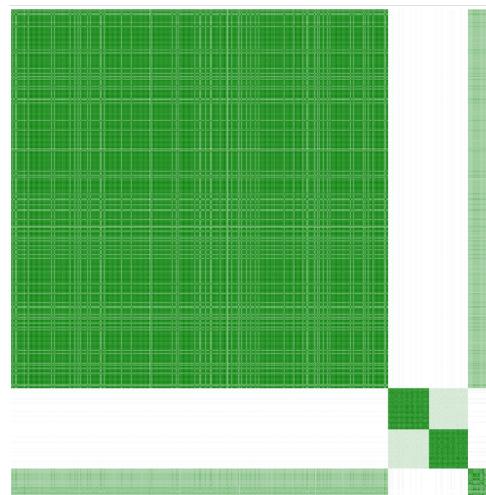
---

- Most data are available in tabular formats, e.g., CSV
- Many topics are covered
- Number of data sets is growing
  - In 2013, more data sets were added than in the 3 previous years combined
- *Data is small*: 70GB for all cities
  - Compare against 1 year of taxi data: 50GB/year
- There are big and small tables
- Lots of *spatio-temporal data*
  - Over 50% of the tables have lat+long and over 40% have date
- There is ample opportunity for integration – significant overlap across tables: schema and spatial!

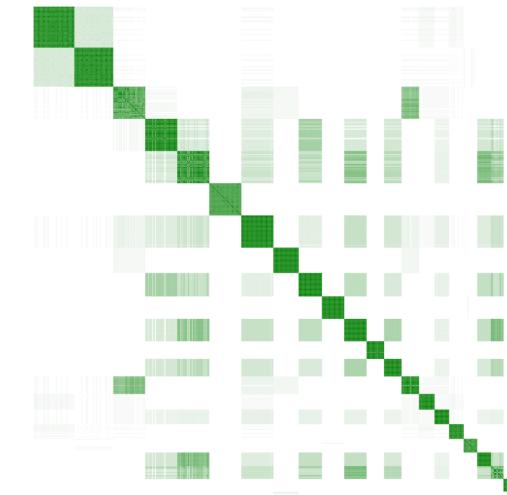
# Integration Opportunities



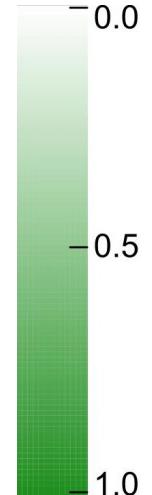
(a) Boston



(b) 4 largest NYC clusters



(c) NYC without 311 data set



(d) Similarity Scale

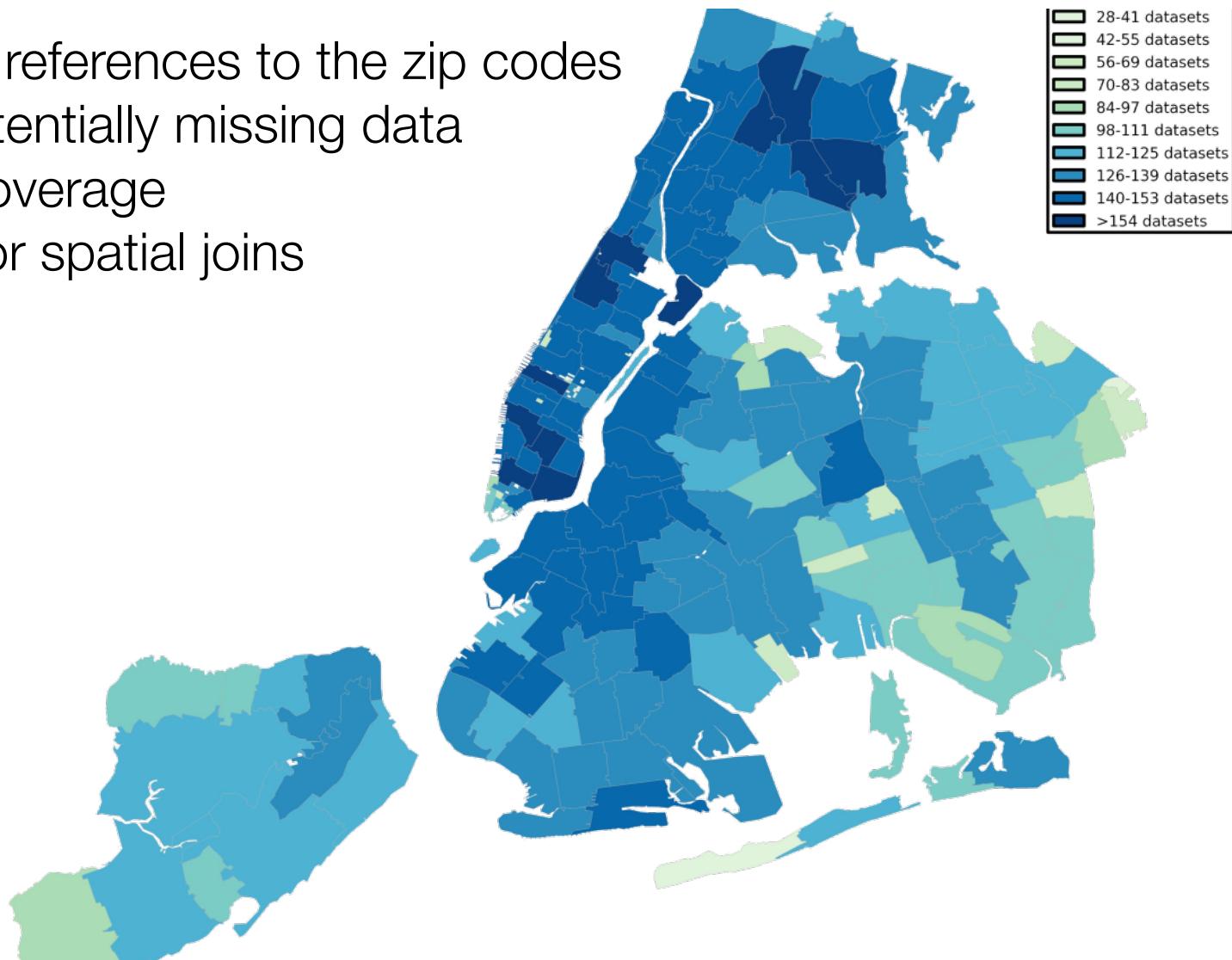
Attribute overlap among tables

- Potential for joining tables
- Hints about horizontally partitioned tables

# Integration Opportunities

Frequency of references to the zip codes

- Identify potentially missing data
- Quantify coverage
- Potential for spatial joins



Geographical coverage and overlap

# It's not all roses...

---

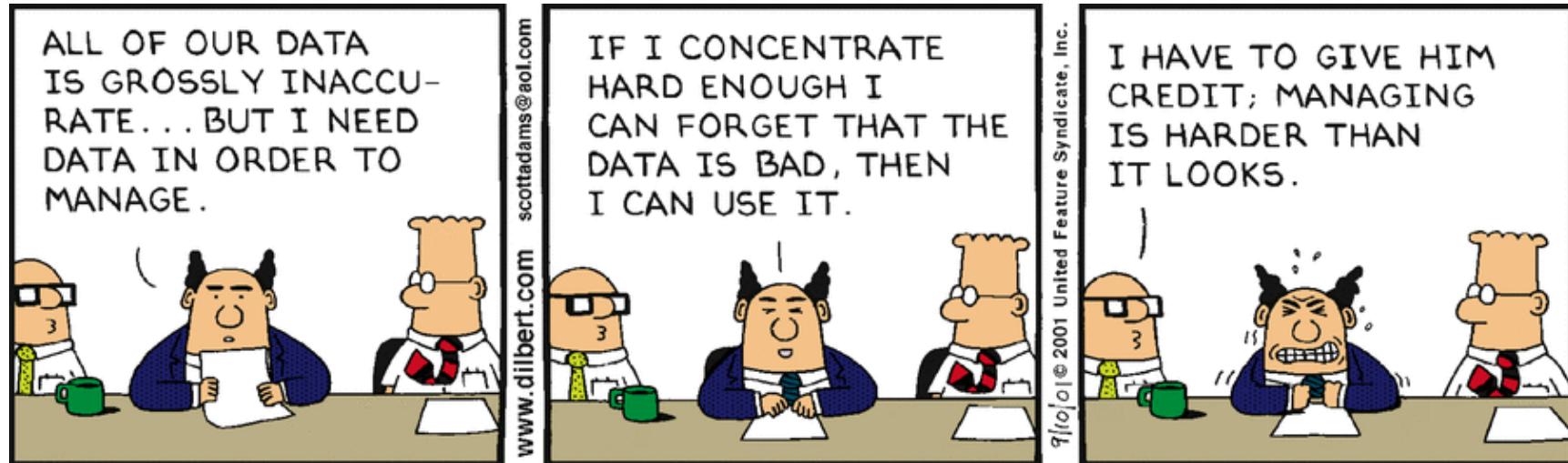


# Outline

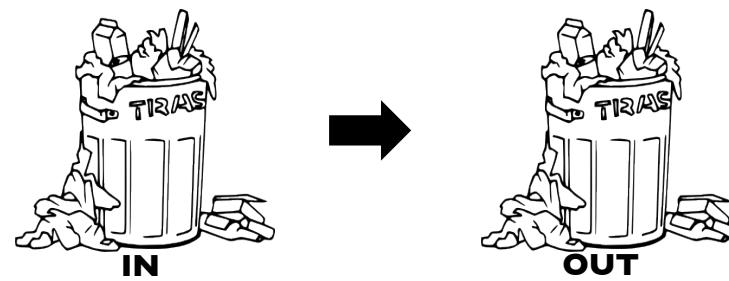
---

- Introduction to Programming with Big Data
  - Map Reduce vs. Parallel Databases
- Running Example: Big Urban Data
- Data Cleaning
  - Overview and Challenges
  - Cleaning the NYC Taxi Data: A Case Study
- Data exploration
  - Usability and Interactivity
  - Finding Interesting Features
  - Using Data to Discover and Explain Data
- Transparency and Reproducibility

# Data Quality → Analysis Quality



- **Data is a critical resource** that supports analytics and decision making
- As data volumes increase, so does the complexity of managing it and the **risks of poor data quality**.



Modified from H. Müller

# The Impact of Data Quality

*Because of poor data quality ...*

- 88% of data integration projects fail to meet budgets
- 75% of organizations have additional costs due to poor data quality
- 33% of organizations delayed or canceled projects because of poor data quality
- \$611bn per year is lost in the US alone

In [Marsh 2005] summarizing reports by **Gartner Group, Warehousing Institute**

**Bad Data Costs the U.S. \$3 Trillion Per Year**

by Thomas C. Redman  
SEPTEMBER 22, 2016

SAVE | SHARE | COMMENT (8) | TEXT SIZE | PRINT | \$8.95 BUY COPIES



Consider this figure: \$136 billion per year. That's the [research firm IDC's estimate](#) of the size of the big data market, worldwide, in 2016. This figure should surprise no one with an interest in big data.

But here's another number: [\\$3.1 trillion, IBM's estimate](#) of the yearly cost of poor quality data, in the US alone, in 2016. While most people who deal in data every day know that bad data is costly, this figure stuns.

While the numbers are not really comparable, and there is considerable variation around each, one can only conclude that right now, improving data quality represents the far larger data opportunity. Leaders are well-advised to develop a deeper appreciation for the opportunities improving data quality present and take fuller advantage than they do today.

The reason bad data costs so much is that decision makers, in the US and elsewhere, datacentric as they may be, must accommodate it in their everyday work. And doing so is both time-consuming and expensive. The data they need has plenty of holes.

# Big Data Cleaning: Big Problem

≡ SECTIONS    HOME    SEARCH



Rise of Bitcoin Competitor Ripple Creates Wealth to Rival Zuckerberg



What You Need to Do Because of Flaws in Computer Chips



CRITIC'S NOTEBOOK  
How HQ Trivia Became the Best Worst Thing on the Internet



BITS  
Farhad and Mike's Week in Tech: Another Huge Security Flaw



TECH TIP  
Sharing Online A With Google Phc

The New York Times



A strong network helps you ensure scalability, agility and security.



TECHNOLOGY

***For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights***

<https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html>

# Big Data Cleaning: Big Problem

MAR 23, 2016 @ 09:33 AM 15,078 VIEWS

## Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says



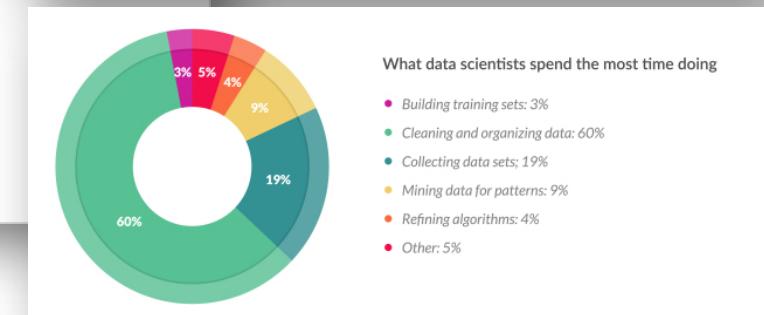
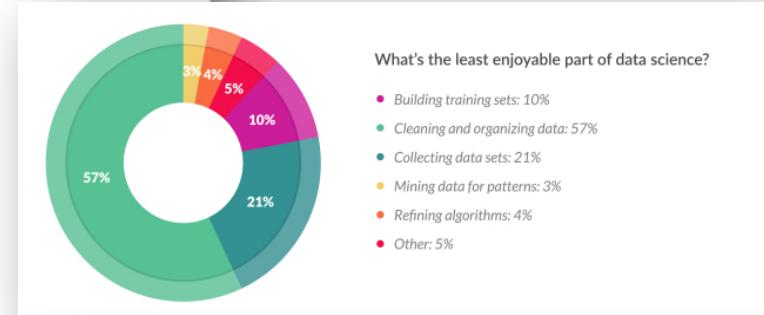
 **Gil Press, CONTRIBUTOR**  
I write about technology, entrepreneurs and innovation. [FULL BIO](#) ▾  
Opinions expressed by Forbes Contributors are their own.

**TWEET THIS**

- data scientists found that they spend most of their time massaging rather than mining or modeling data.
- 76% of data scientists view data preparation as the least enjoyable part of their work

A new survey of data scientists found that they spend most of their time massaging rather than mining or modeling data. Still, most are happy with having the sexiest job of the 21<sup>st</sup> century. The survey of about 80 data scientists was conducted for the second year in a row by CrowdFlower, provider of a “data enrichment” platform for data scientists. Here are the highlights:

- *Least enjoyable part of Data Science?*
  - Collecting data (21%)
  - Cleaning and organizing data (57%)
- *Spend most time doing*
  - Collecting data (19%)
  - Cleaning and organizing data (60%)



<https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says>

Modified from H. Müller

# Data Cleaning

---

*“the process of starting with raw data from one or more sources and maintaining reliable quality for your applications”*

*“the exercise of detecting errors, and possibly modifying the database, such that the data conforms to a set of data quality rules”*

- Steps:
  1. **Identify errors:** different types of errors require different detection techniques
  2. **Fixing errors:** approaches can be automated or require the user in the loop

# Data Cleaning Mechanisms and Operations

---

- Integrity constraints
- Similarity join
- Clustering
- Parsing
- More during the lab

# Integrity Constraints

- Usually declared as part of the database schema

CREATE TABLE Account

```
(ID          integer,  
State       character NOT NULL,  
ZIP         integer,  
PRIMARY KEY (ID));
```

ID	State	ZIP
1	NULL	85376
2	AZ	85376
3	NY	90012

ID	State	ZIP
1	NY	85376
1	AZ	85376
3	NY	90012

# Integrity Constraints

- Have been increasingly used as data quality rules
  - Checking the validity of the data upon addition or update – data ingestion
  - Cleaning the dirty data at various points during the processing pipeline

ZIP → State

ID	State	ZIP
1	NY	85376
2	AZ	85376
3	NY	90012

Either tuple 1 or tuple 2 is incorrect!

# Integrity Constraints

- Have been increasingly used as data quality rules
  - Checking the validity of the data upon addition or update
  - Cleaning the dirty data at various points during the processing pipeline
  - Derive missing values

ZIP → State

ID	State	ZIP
1	NY	85376
2	AZ	85376
3	NY	90012
4	NULL	90012

What is the State for tuple 4?

- In Databases, one defines the constraints when the database is created
- For Big Data, often the constraints must be inferred

# Data Cleaning Operations

- Similarity join: join similar data [Chaudhuri et al., ICDE 2006]
  - Useful for record matching as well as deduplication

```
SELECT *  
FROM R, S  
WHERE sim(R.A,S.B,a)
```

ID	Name	Street	City	Phone
r1	Sweetlegal Investments Inc	202 North	Redmond	425-444-5555
r2	ABC Groceries Corp	Amphitheatre Pkwy	Mountain View	4081112222
r3	Cable television services	One Oxford Dr	Cambridge	617-123-4567
s1	Sweet legal Invesments Incorporated	202 N	Redmond	
s2	ABC Groceries Corpn.	Amphitheetre Parkway	Mountain View	
s3	Cable Services	One Oxford Dr	Cambridge	6171234567

# Similarity Functions: Edit Distance

---

- The *edit distance* of two strings A and B is the number of inserts and deletes of characters needed to turn A into B.
- Edit similarity is  $(1 - \text{Edit Distance}) / \max(|A|, |B|)$
- Good for typographical errors

$\text{ED}(\text{"John Doe"}, \text{"Jon Doe"}) = ?$

$\text{ED}(\text{"Sweet}\mathbf{l}\text{egal Investments}\mathbf{s"}, \text{"Swee}\mathbf{l}\text{egal Investment."}) = ?$

- ED fails to capture the similarity between two strings that use the same set of tokens with different ordering, e.g., "Sweetlegal Investment", "Investment Sweetlegal"

# Similarity Functions: Jaccard

---

- The *Jaccard similarity* between two sets A and B is the ratio of the size of the intersection over the size of the union
$$|A \text{ intersection } B| / |A \cup B|$$
- Need to transform strings into sets – use q-grams
- For A = “Sweet”, B = “Sweat.”, and q = 2  
$$Q(s1) = \{\mathbf{Sw}; \mathbf{we}; ee; et\}$$
 and 
$$Q(s2) = \{\mathbf{Sw}; \mathbf{we}; ea; at\}$$
$$\text{Jaccard}(Q(s1), Q(s2)) = 2/6 = 1/3$$
- Q-grams based Jaccard similarities are very useful for longer strings, such as addresses or documents

# Similarity Functions: Cosine

---

- The *cosine similarity* is a vector-based similarity measure
- Need to transform strings into high-dimensional vectors
  - E.g., each token becomes a dimension For A = “Sweet Inc”, B =“Sweet”  
 $v(s1) = \{1,1\}$  and  $v(s2) = \{1,0\}$   
 $\text{cosine}(v(s1), v(s2)) = \text{cosine}(45) = 0.707$
- Useful for larger strings such as web documents, addresses, or text records
- To avoid high-dimensionality and noise due to irrelevant words, stop-words (such as “the,” “an,” etc.), and commonly occurring words are often eliminated

# Similarity Functions: Soundex

---

- Phonetic approach for measuring the similarity between strings
- Convert any string to some code based on the pronunciation of the word, and then compare the codes of the strings
- For American English see  
<http://en.wikipedia.org/wiki/Soundex>

# Similarity Functions: Summary

---

- No single string similarity function is uniformly good – choice usually depends on the data and application domain
  - E.g., Matching products – slightly different representations are meaningless
    - “Apple iPhone 7 , GSM Unlocked, 32GB”
    - “Apple iPhone 7 (GSM Unlocked) 32Gigabytes”
  - E.g., Matching addresses – slightly different representations are significant
    - “148th Ave” and “147th Ave”
- Use a classifier to learn and determine whether two fields are similar by combining multiple similarity measures

# Data Cleaning Operations

- Clustering
  - Group similar records for de-duplication: pair-wise comparison only within a group
  - Grouping similar values to find 'dirt'
  - Use similarity measures we just discussed
- Parsing: transform input data into the structure defined in the application

Transform “15633 148th Ave Bellevue WA 98004” into

**House number:** 15633

**Street name:** 148<sup>th</sup> Ave

**City:** Bellevue

**State:** WA

**Zip:** 98004

Player Name	Grand Slams Won
R. Federer	13
D. Ferrer	4
Rafa Nadal	6
Rafael Nadal	3

Player ID	Names of Players
1	R. Federer
2	D. Ferrer
3	Rafa Nadal, Rafael Nadal

# Data Cleaning Tools: Verticals

---

- Data cleaning functionality for specific domains, e.g., <http://www.trilliumsoftware.com>
- By incorporating domain knowledge, solutions can be comprehensive and easier to deploy
  - Example: a data cleaning package for addresses can incorporate the format of a typical address record, and lookup tables (\*zipcodes, city, state) to standardize the format
- Downside: solutions are not generic and cannot be ported to other domains
  - A solution developed for U.S. addresses cannot be applied to the domain of electronic products, or to addresses in Brazil

# Data Cleaning Tools: Generic Platforms

---

- Provide a suite of operators for cleaning
- ETL Platforms: Microsoft SQL Server Integration Services and IBM Websphere Information Integration
  - Extensible: applications can plug in their own custom operators
  - A data transformation and cleaning solution is built by composing the default and custom operators to obtain an operator tree or a graph
  - E.g., But need to have a DB backend...
- Interactive tools: e.g., OpenRefine, Trifacta Wrangler, TAMR
  - User in the loop: focus on making it easy to clean data
  - We will use OpenRefine during the lab

<https://www.tamr.com>, <https://www.trifacta.com/products/wrangler>,  
<http://openrefine.org>

# Cleaning Small Data

---

- What to do:
  - Remove errors
  - Fill in missing information
  - Transform units and formats
  - Map and align columns
  - Remove duplicates records
  - Fix integrity constraint violations
  - Specify domain knowledge as integrity constraints
- Very rich literature and many tutorials
- Some tools are available

Modified from Chu & Ilyas



# Big Data + Data Quality: Challenges

- Constraints are not known a priori – they need to be discovered *Complete Knowledge infeasible*
- Size and complexity: huge volume of heterogeneous data from multiple sources *Need scalable solutions*
- Speed: dynamic data, collected and analyzed at high velocity *Domain knowledge becomes obsolete*
- Evolution: considerable variability of data, semantics over time
- Active area of research
  - Learn/infer constraints (semantics) from the data
  - Automatically identify data glitches
- Need (semi) automated methods and toolkits
  - Get ready to build your own!

Modified from D. Srivastava

# Quality Issues in Urban Data

# Challenge: Inconsistent Metadata

*Hard to understand semantics*

Data Dictionary – Yellow Taxi Trip Records																			September 28, 2015	Page 1 of 1
This data dictionary describes yellow taxi trip data. For dictionaries describing green taxi and FHV data, please visit <a href="http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml">http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml</a> .																				
<b>Field Name</b>																			<b>Description</b>	
<b>VendorID</b>																			A code indicating the TPEP provider that provided the record.	
1= Creative Mobile Technologies, LLC; 2= VeriFone Inc.																				
<b>tpep_pickup_datetime</b>																			The date and time when the meter was engaged.	
<b>tpep_dropoff_datetime</b>																			The date and time when the meter was disengaged.	
<b>Passenger_count</b>																			The number of passengers in the vehicle.	
This is a driver-entered value.																				
<b>Trip_distance</b>																			The elapsed trip distance in miles reported by the taximeter.	
<b>Pickup_longitude</b>																			Longitude where the meter was engaged.	
<b>Pickup_latitude</b>																			Latitude where the meter was engaged.	
<b>RateCodeID</b>																			The final rate code in effect at the end of the trip.	
1= Standard rate 2=JFK 3=Newark 4=Nassau or Westchester 5=Negotiated fare 6=Group ride																				
<b>Store_and_fwd_flag</b>																			This flag indicates whether the trip record was held in vehicle memory before sending to the vendor, aka "store and forward," because the vehicle did not have a connection to the server.	
Y= store and forward trip N= not a store and forward trip																				
<b>Dropoff_longitude</b>																			Longitude where the meter was disengaged.	
<b>Dropoff_latitude</b>																			Latitude where the meter was disengaged.	
<b>Payment_type</b>																			A numeric code signifying how the passenger paid for the trip.	
1= Credit card 2= Cash 3= No charge 4= Dispute 5= Unknown 6= Voided trip																				
<b>Fare_amount</b>																			The time-and-distance fare calculated by the meter.	
<b>Extra</b>																			Miscellaneous extras and surcharges. Currently, this only includes the \$0.50 and \$1 rush hour and overnight charges.	
<b>MTA_tax</b>																			\$0.50 MTA tax that is automatically triggered based on the metered rate in use.	
																			Click to sort largest first	
Showing 135,335,924 out of 135,335,924 rows																				
vendorid	tpep_pickup_datetime	dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	ratecodeid	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount				
CMT	08/08/2011 11:49:24 PM	08/08/2011 11:57:45 PM	1	2.5	-74.00563	40.740509	1	N	-74.016117	40.711134	CRD	8.5	0.5	0.5	0	0				
CMT	09/15/2011 03:54:48 AM	09/15/2011 03:58:54 AM	1	1.2	0	0	1	N	0	0	CRD	5.3	0.5	0.5	1.89	0				
CMT	07/05/2011 04:03:40 PM	07/05/2011 04:15:53 PM	1	1.7	-73.978989	40.736481	1	N	-73.977367	40.754893	CSH	8.1	1	0.5	0	0				
CMT	09/03/2011 07:47:57 PM	09/03/2011 07:59:32 PM	3	1.3	-73.992899	40.729957	1	N	-74.008172	40.716759	CSH	7.7	0	0.5	0	0				
CMT	01/10/2011 09:07:22 AM	01/10/2011 09:13:06 AM	1	1	-73.990681	40.734814	1	N	-73.998295	40.74209	CRD	5.7	0	0.5	0.93	0				
CMT	04/29/2011 09:41:47 PM	04/29/2011 10:08:26 PM	1	3.1	-73.989607	40.731801	1	N	-73.982571	40.757293	CSH	14.1	0.5	0.5	0	0				
CMT	06/25/2011 02:58:07 PM	06/25/2011 03:05:53 PM	2	1.4	-73.96	40.7816	1	N	-73.9669	40.7642	CSH	6.5	0	0.5	0	0				
CMT	01/26/2011 01:10:25 PM	01/26/2011 01:18:59 PM	1	2.2	-73.946743	40.772331	1	N	-73.97567	40.760758	CSH	7.7	0	0.5	0	0				
CMT	06/16/2011 12:12:11 AM	06/16/2011 12:29:45 AM	1	5.8	-73.983032	40.726829	1	N	-73.977406	40.789431	CRD	15.7	0.5	0.5	3.34	0				
CMT	01/27/2011 09:00:40 PM	01/27/2011 09:07:52 PM	1	1.4	-73.98935	40.741689	1	N	-73.981712	40.758483	CRD	6.1	0.5	0.5	2	0				
CMT	09/02/2011 01:45:10 PM	09/02/2011 01:55:59 PM	1	1	-73.990709	40.755332	1	N	-73.974292	40.751193	CSH	7.3	0	0.5	0	0				



NYU

TANDON SCHOOL  
OF ENGINEERING



VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Challenge: Evolving Data

*Analyses may break for new data*

Removed data

VendorID	tpep_pickup_datetime	tpep_dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	RatecodeID	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount
3	2016 Jul 17 06:46:27 PM	2016 Jul 17 06:53:03 PM	1	0.87			1	N			1	6	0	0.5	0.68	
3	2016 Jul 02 12:48:30 AM	2016 Jul 02 01:03:52 AM	1	2.63			1	N			2	12	0.5	0.5	0	
3	2016 Sep 11 12:17:32 AM	2016 Sep 11 12:29:47 AM	1	1.81			1	N			1	9.5	0.5	0.5	2.16	
3	2016 Jul 13 05:18:46 PM	2016 Jul 13 05:28:25 PM	1	0.91			1	N			1	7.5	1	0.5	2.33	
3	2016 Jul 06 03:17:09 PM	2016 Jul 06 03:20:45 PM	1	0.55			1	N			1	4.5	0	0.5	1.06	
3	2016 Jul 01 11:11:24 PM	2016 Jul 01 11:31:05 PM	1	2.06			1	N			1	13.5	0.5	0.5	0	
3	2016 Jul 17 03:19:41 PM	2016 Jul 17 03:33:01 PM	1	1.71			1	N			2	10	0	0.5	0	
3	2016 Jul 02 11:22:15 AM	2016 Jul 02 11:29:26 AM	1	1.5			1	N			2	7.5	0	0.5	0	
3	2016 Jul 16 01:37:03 PM	2016 Jul 16 02:00:16 PM	1	4.33			1	N			1	19	0	0.5	0	
3	2016 Aug 07 06:33:27 PM	2016 Aug 07 06:58:23 PM	1	3.15			1	N			1	17	0	0.5	2.67	
3	2016 Sep 15 12:12:21 PM	2016 Sep 15 12:17:48 PM	1	0.49			1	N			1	5.5	0	0.5	1.26	

vendorid	tpep_pickup_datetime	dropoff_datetime	passenger_count	trip_distance	pickup_longitude	pickup_latitude	ratecodeid	store_and_fwd_flag	dropoff_longitude	dropoff_latitude	payment_type	fare_amount	extra	mta_tax	tip_amount	tolls_amount
CMT	08/08/2011 11:49:24 PM	08/08/2011 11:57:45 PM	1	2.5	-74.00563	40.740509	1	N	-74.016117	40.711134	CRD	8.5	0.5	0.5	0	0
CMT	09/15/2011 03:54:48 AM	09/15/2011 03:58:54 AM	1	1.2	0	0	1	N	0	0	CRD	5.3	0.5	0.5	1.89	0
CMT	07/05/2011 04:03:40 PM	07/05/2011 04:15:53 PM	1	1.7	-73.978989	40.736481	1	N	-73.977367	40.754893	CSH	8.1	1	0.5	0	0
CMT	09/03/2011 07:47:57 PM	09/03/2011 07:59:32 PM	3	1.3	-73.992899	40.729957	1	N	-74.008172	40.716759	CSH	7.7	0	0.5	0	0
CMT	01/10/2011 09:07:22 AM	01/10/2011 09:13:06 AM	1	1	-73.990681	40.734814	1	N	-73.998295	40.74209	CRD	5.7	0	0.5	0.93	0
CMT	04/29/2011 09:41:47 PM	04/29/2011 10:08:26 PM	1	3.1	-73.989607	40.731801	1	N	-73.982571	40.757293	CSH	14.1	0.5	0.5	0	0
CMT	06/25/2011 02:58:07 PM	06/25/2011 03:05:53 PM	2	1.4	-73.96	40.7816	1	N	-73.9669	40.7642	CSH	6.5	0	0.5	0	0
CMT	01/26/2011 01:10:25 PM	01/26/2011 01:18:59 PM	1	2.2	-73.946743	40.772331	1	N	-73.97567	40.760758	CSH	7.7	0	0.5	0	0
CMT	06/16/2011 12:12:11 AM	06/16/2011 12:29:45 AM	1	5.8	-73.983032	40.726829	1	N	-73.977406	40.789431	CRD	15.7	0.5	0.5	3.34	0
CMT	01/27/2011 09:00:40 PM	01/27/2011 09:07:52 PM	1	1.4	-73.98935	40.741689	1	N	-73.981712	40.758483	CRD	6.1	0.5	0.5	2	0
CMT	09/02/2011 01:45:10 PM	09/02/2011 01:55:59 PM	1	1	-73.990709	40.755932	1	N	-73.974292	40.751193	CSH	7.3	0	0.5	0	0

Different data types

Rows 1-11 out of 135,335,924

# Challenge: Data Quality Issues

---

**DOHMH New York City Restaurant Inspection Results**

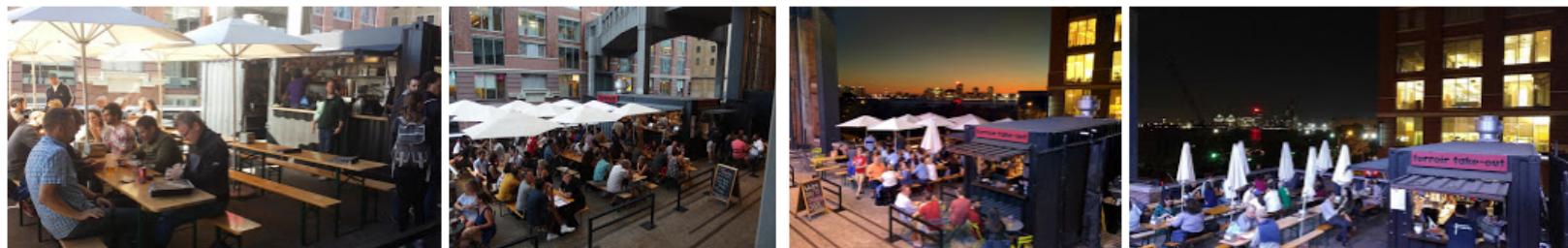
DBA	STREET	BUILDING
MADANGSUI	WEST 35 STREET	35
@NINE	9 AVENUE	592
TACO HUT	BROADWAY	3210

<https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j>

# Challenge: Data Quality Issues

**DOHMH New York City Restaurant Inspection Results**

DBA	STREET	BUILDING
MADANGSUI	WEST 35 STREET	35
@NINE	9 AVENUE	592
TACO HUT	BROADWAY	3210
<b>TERROIR AT THE PORCH</b>	<b>W 15th Street @ 10th Ave</b>	<b>HIGHLINE</b>



<https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j>

# Challenge: Data Quality Issues

---

**DOHMH New York City Restaurant Inspection Results**

DBA	STREET	BUILDING
MADANGSUI	WEST 35 STREET	35
@NINE	9 AVENUE	592
TACO HUT	BROADWAY	3210
<b>TERROIR AT THE PORCH</b>	<b>W 15th Street @ 10th Ave</b>	<b>HIGHLINE</b>

People that generate data get ‘creative’ to fit information to data models.

Lack of provenance information means we have to attempt to understand their decisions and the data generation process.

<https://data.cityofnewyork.us/Health/DOHMH-New-York-City-Restaurant-Inspection-Results/43nn-pn8j>

# Challenge: Data Quality Issues

- Columns containing Telephone Numbers in NYC Open Data
- Think of a (simple) way to distinguish the ‘Good’ from the ‘Bad’ and to transform the bad into good.

.

0

212 NEW YORK

311

511

911

0000000000

1111111

1111111111

1212669311

2012162746

2015954606

2033631907

9737924762

9737924769

Fax7189801021

Fax:7189187823

(000) 000-0000

(201) 368-1000

(201) 373-9599

(718) 206-1088

(718) 206-1121

(718) 206-1420

(718) 206-4420

(718) 206-4481

(914) 681-6200

(718) 868-2300 x206

(718) 206-0545 / (718) 298-0117

(718) 262-9072 / (718) 658-1537

(718) 297-4708/c: (347) 806-4588

(888) 8NYC-TRS

(888)-VETS-NYS

1-800-CUNY-YES

800-624-4143

# Challenge: Data Quality Issues

- Columns containing Boroughs, Cities, Neighborhoods in NYC Open Data
- Cities, neighborhoods and boroughs all mixed: how to fix this?

borough (0)	city (1)	manhattan neighborhood (2)
BRONX	ASTORIA	CHELSEA
BROOKLYN	BRONX	CHINATOWN
MANHATTAN	BROOKLYN	CLINTON
QUEENS	CHELSEA	HARLEM
STATEN ISLAND	CLINTON	SOHO
	FLUSHING	TRIBECA
	HARLEM	
	JAMAICA	
	QUEENS	
	MANHATTAN	
	NEW YORK	
	STATEN ISLAND	

# Challenge: Data Quality Issues

- Assumption about valid values in a column, i.e., the domain  
Data Type (INT, DECIMAL, TEXT, DATE)
- Semantic constraints often not explicitly documented  
ZIP Code is a 5 digit number between 10000 and 99999  
Monetary value in US\$  
Date in format YYYY-MM-DD  
Name in format <first> <last>
- Pairs of records that contradict each other or violate a functional dependency  
 $\text{ZIP} \rightarrow \text{City}$

*Attribute:  
illegal and  
missing values*

ZIP	City
10003	NYC
10003	Chicago

- Uniqueness violations, conflicting values, missing records

# Toolbox of a Data Cleaner

---

- *External (High Quality) Data Sources*
  - E.g., lookup tables for city names and ZIP codes
- *Integrity Constraints*
  - Define and enforce constraints that high quality data adhere to
- *Regular Expressions*
  - Define format of values
- *String Similarity Functions and Clustering*
  - Identify typos at data entry
  - Find records that represent the same entity (duplicates)
- *Conflict Resolution Functions*
  - Resolve contradicting information



[Skip to case study](#)

Modified from H. Müller

# Find Attribute Outlier Values

---

- Sort attribute values in alphabetical order
- ‘Interesting’ values often appear at the beginning and end of list

The following examples are from the **DOB Permit Issuance** dataset  
in **NYC Open Data**

## owner\_s\_business\_name

(JOANNE H. SIEGMUN 2ND OWNER)

(PERSONAL RESIDENCE)

(PRIVATE RESIDENCE)

(TENANT IN COMMON)

(TENANTS IN COMMON)

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

+++++

-

--

---

-----

.

..

[...]

[...]

\_\_\_\_\_N/A

altered state restoration

c/o Bowery Hotel

c/o Cooper Square Realty

c/o Leibovitz Studio

individual

mtp investment

n/a

na

new hempstead home for the adult

none

not applicable

owner

renaissanc

same

sierra realty corp.

wm maidmanfamily lp

# Outliers in Alphabetical Order

city

(646)4396000

, FLORAL PARK

,ELMSFORD

.

1

10012

10013

10452

10462

105

A large number of quality problems are a result of ‘parsing errors’ or invalid file formats (e.g., too many or missing column delimiters in CSV file).

QUEENS|4144683|147-57 | 78 AVE |421156046|01|A1||06688|00040

|408|11367|1|YES|||PL|ISSUED|RENEWAL|PL|02| | |NOT APPLICABLE

|11/06/2016|11/06/2016|11/06/2017|11/10/2015|CONSTANTINE | KOUMPAROULIS

|ARIANA CONTRACTING INC |7187215018|MASTER PLUMBER |0001101| | | | | |

| | |INDIVIDUAL ||N/A |ARTUR |KHAIMOV |147-57 | 78TH AVENUE |KEW

GARDENS |NY|11367 |6464022132|11/07/2016

# Find Attribute Outlier Values

---

- Sort attribute values in alphabetical order
  - ‘Interesting’ values often appear at the beginning or end of list.
- Frequency outliers
  - NULL values sometimes have significantly different frequency (high or low) compared to other column values.

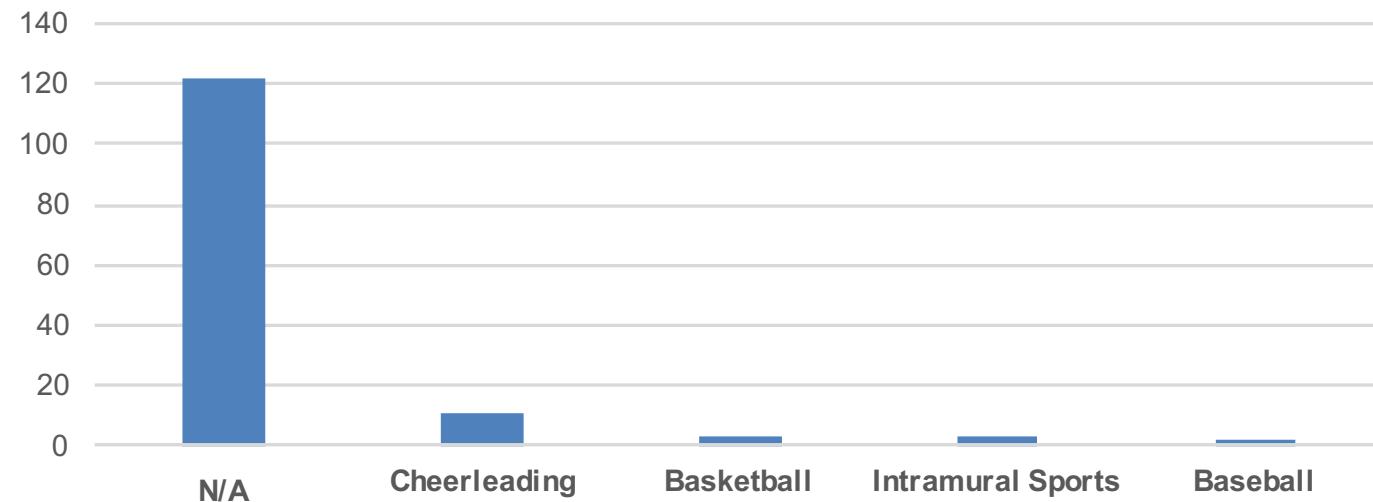
# Frequency Outliers

---

**DOE High School Directory 2013-2014**

**NYC Open Data**

**school\_sports**



# Frequency Outliers (cont.)

---

- *Values that frequently occur as high frequency outliers*
  - Values that occur with frequency >50% in + 15,000 columns of NYC Open Data datasets

Value	Number of Columns
0	(x 262)
N/A	(x 71)
UNSPECIFIED	(x 67)
S	(x 57)
-	(x 50)
<b>0.00</b>	<b>(x 47)</b>
NY	(x 38)
1	(x 25)
0 . 0	(x 20)
IND	(x 12)
CLOSED	(x 10)
100	(x 8)
NOT AVAILABLE	(x 8)
0 UNSPECIFIED	(x 6)
NONE	(x 5)

# Find Attribute Outlier Values

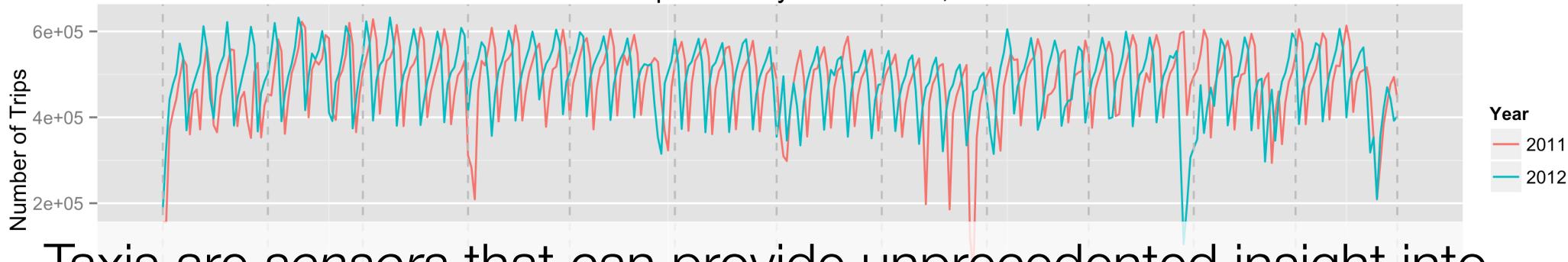
---

- Sort attribute values in alphabetical order
  - ‘Interesting’ values often appear at the beginning or end of list
- Frequency outliers
  - NULL values sometimes have significantly different frequency (high or low) compared to other column values
- Regular expressions
  - Find values that do not match the expected format of a column
- Often identify outliers and potential problems during data exploration

# Data Quality: A Case Study using NYC Taxi Trips

# NYC Taxis

Number of Trips for the years of 2011, and 2012



Taxis are sensors that can provide unprecedented insight into city life: economic activity, human behavior, mobility patterns

*“What is the average trip time from Midtown to the airports during weekdays?”*

*“How was traffic affected during the Macy’s Parade?”*

*“Where are the popular night spots?”*

*“Which neighborhoods are being gentrified?”*



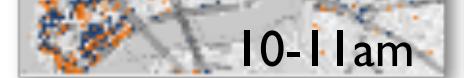
7-8am



8-9am



9-10am



10-11am

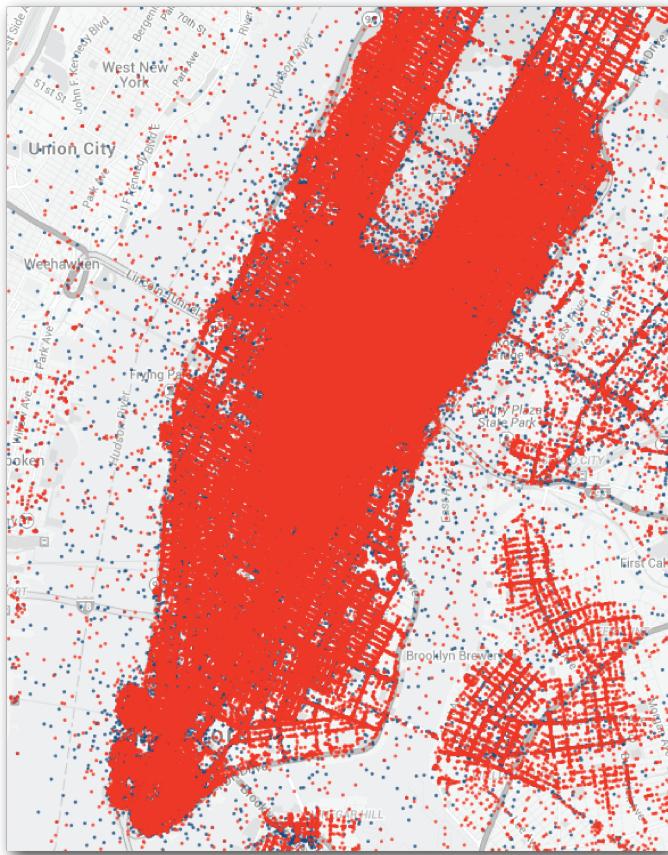
# Taxi Data: What to Clean and not to Clean

Dataset	Statistic	Trip Duration (min)	Trip Distance (mi)	Fare Amount (US\$)	Tip Amount (US\$)
2008	Min	0.00	0.00	0.00	0.00
	Avg	16.74	2.71	0.09	0.10
	Max	1440.00	50.00	10.00	8.75
2009	Min	0.00	0.00	2.50	0.00
	Avg	7.75	6.22	6.04	0.38
	Max	180.00	180.00	200.00	200.00
2010	Min	-1,760.00	-21,474,834.00	-21,474,808.00	-1,677,720.10
	Avg	6.76	5.89	9.84	2.11
	Max	1,322.00	16,201,631.40	93,960.07	938.02
2011	Min	0.00	0.00	2.50	0.00
	Avg	12.35	2.80	10.25	2.22
	Max	180.00	100.00	500.00	200.00
2012	Min	0.00	0.00	2.50	0.00
	Avg	12.32	2.88	10.96	2.32
	Max	180.00	100.00	500.00	200.00

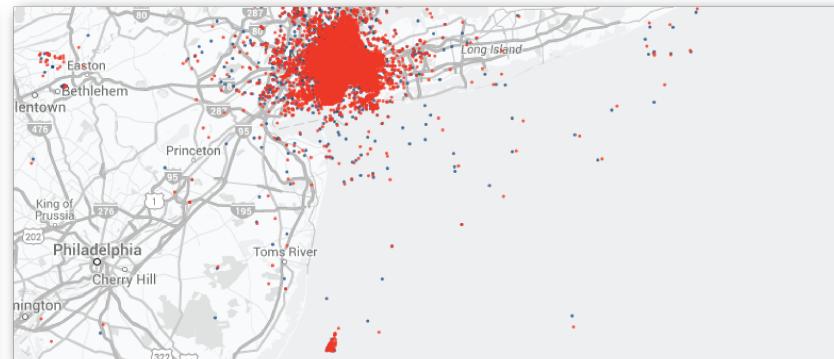
Negative values are clearly errors.  
But high tip may not be an error...

Different processes were used to process data in different years,  
but no provenance information is provided

# Taxi Data: What to Clean and not to Clean



(a)



(b)



(c)

Need to consider spatial constraints:  
Trips in rivers, ocean and Central America



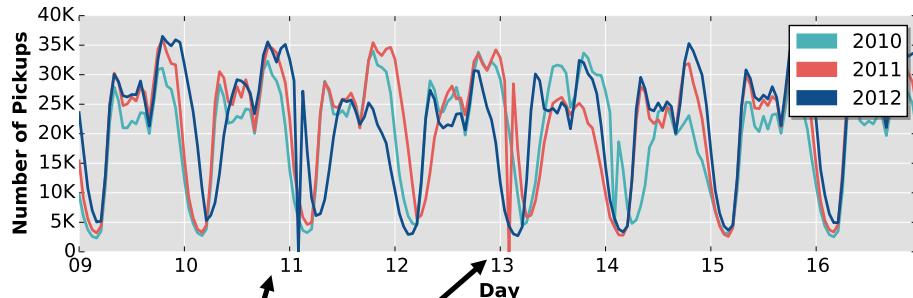
NYU

TANDON SCHOOL  
OF ENGINEERING

[Freire et al., IEEE DEB 2016]

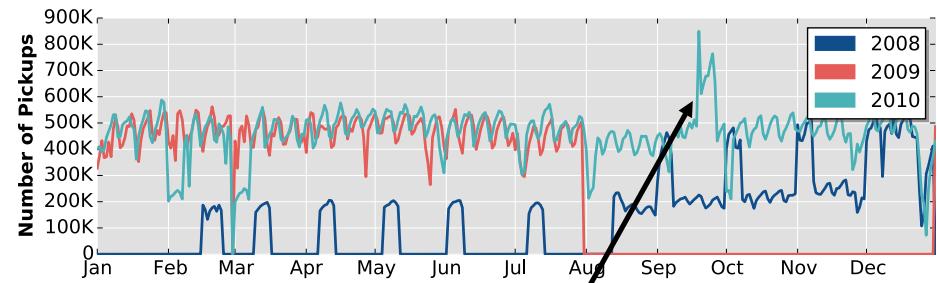
VIDA  
VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Taxi Data: What to Clean and not to Clean



No trips at 2am

Daylight savings:  
March 13, 2011  
March 11, 2012



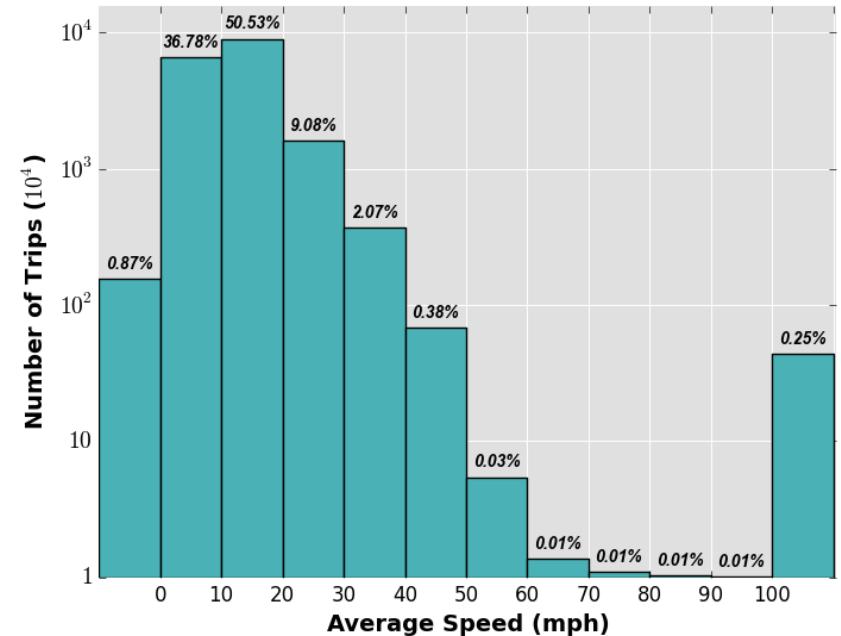
Missing data  
in 2008

Missing data  
in 2009

Unusually large number  
of consecutive and  
extremely short trips  
(lasting less than a  
minute)

# Taxi Data: What to Clean and not to Clean

- Ghost trips
  - Overlapping trips for the same taxi, i.e., for a given taxi, a new trip starts before the previous trip has ended
- Speed too high or too low
  - Incorrect values can negatively impact predictive models which rely on average speeds
  - Speed = 0, easily an error
  - But what about high speeds?



# Takeaway: Big Data Cleaning

---

- Data cleaning has been performed as a pre-processing step  
$$\textit{Dirty Data} \rightarrow \textit{Clean Data}$$
- Cleaning is an integral part of data exploration: constraints that should be checked in the cleaning function, and which might not be evident at first, are naturally discovered
- Different question/analyses require different cleaning strategies  
$$\textit{DirtyData} \times \textit{UserTask} \rightarrow (\textit{CleanData}, \textit{Explanation})$$
- Map Reduce provides a scalable framework for executing cleaning scripts
  - Need a scalable data cleaning library – analogous to sklearn for data cleaning

# Takeaway: Big Data Cleaning (cont.)

---

- Spatio-temporal data adds a new set of constraints and issues that need to be considered
- Visualization is essential!
- Traditional cleaning techniques are useful
- Need domain knowledge
- Cleaning a large number of data sets is hard
- It is not always clear what is dirt and what is a feature
- Promising *research* direction: New techniques that leverage multiple data sets
  - Holistic data cleaning and integration
  - Use data to explain data (more soon!)

# Data Cleaning References

---

- Tutorial: *Data Cleaning: Overview and Emerging Challenges*  
[http://sigmod2016.org/sigmod\\_tutorial1.shtml](http://sigmod2016.org/sigmod_tutorial1.shtml)
- Tutorial: Knowledge curation and knowledge fusion: challenges, models, and applications (SIGMOD 2015)  
[http://lunadong.com/talks/KFTutorial\\_sigmod.pptx](http://lunadong.com/talks/KFTutorial_sigmod.pptx)
- *Profiling relational data: a survey.* [VLDB J. 24\(4\)](#): 557-581 (2015)
- Data Cleaning: A Practical Perspective. Venkatesh Ganti and Anish Das Sarma (2013)
- **Trends in Cleaning Relational Data: Consistency and Deduplication. Ihab Illias and Xu Chu (2015)**
-

# Exploring Data: Usability and Interactivity

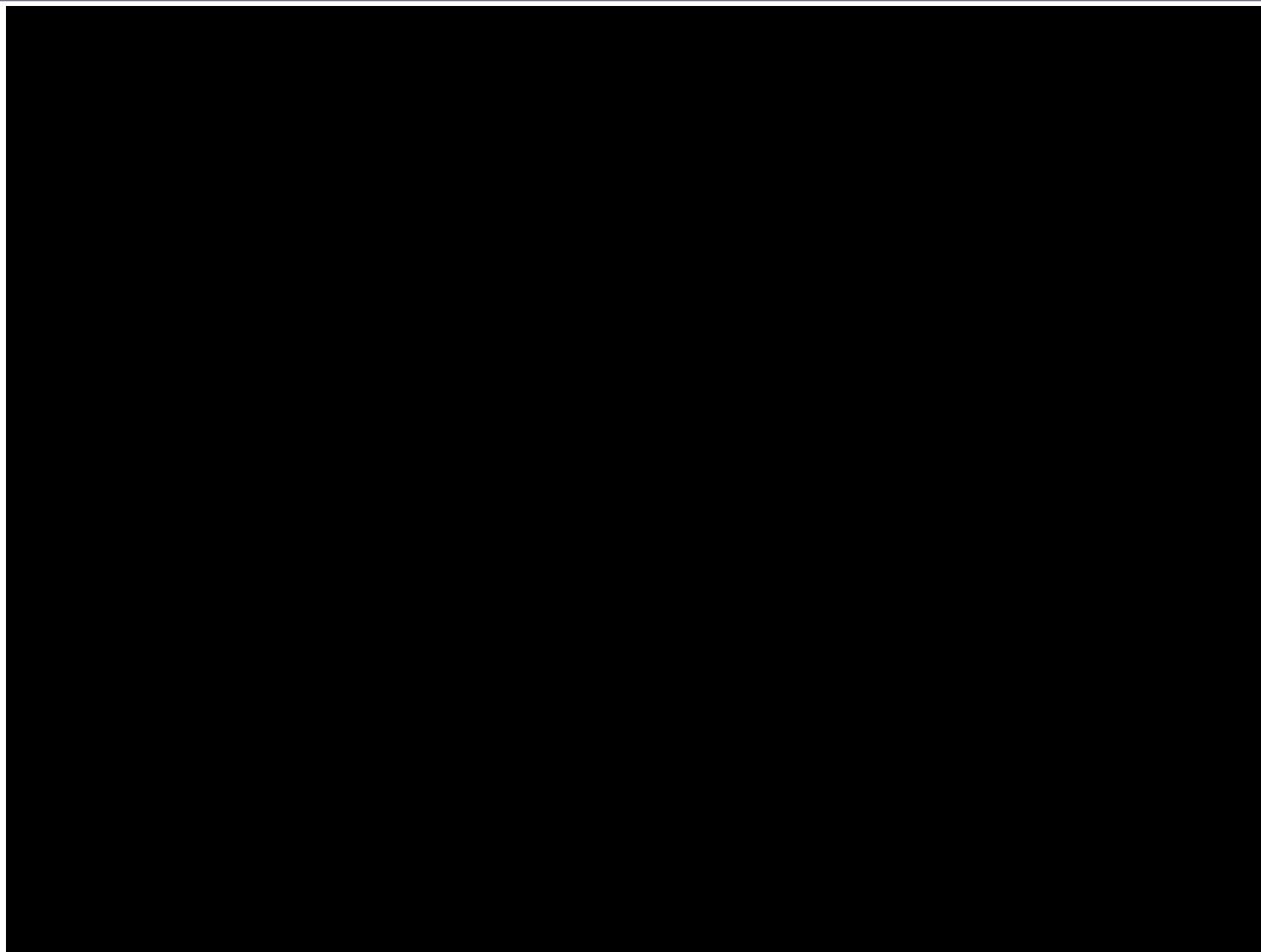
# Exploring Taxi Data: Challenges

---

- Data: ~500k trips/day; 868 million trips in 5 years
  - *spatio-temporal*: pick up + drop off
  - *trip attributes*: e.g., distance traveled, fare, tip
- Government, policy makers and scientists are unable to *interactively explore the whole data*
  - Too many data slices to examine
- Our goal: Design a *usable* interface, efficiently support *interactive + exploratory* queries

# Exploring Taxi Data

---



# Usability through Visual Operations

Users select a data slice by specifying spatial, temporal  
and attribute constraints

```
SELECT *  
FROM trips  
WHERE pickup_time in (5/1/11,5/7/11)  
AND dropoff_loc in "Times Square"  
AND pickup_loc in "Gramercy"
```

Data selection and result  
exploration are unified



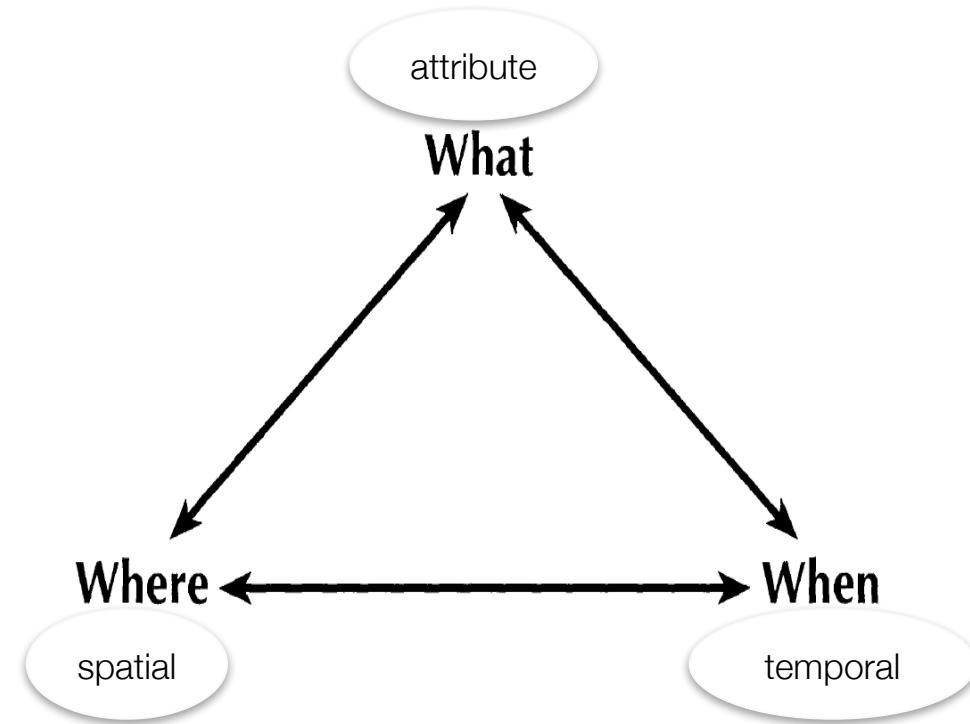
NYU

TANDON SCHOOL  
OF ENGINEERING

# Visual Query Model

## Expressiveness:

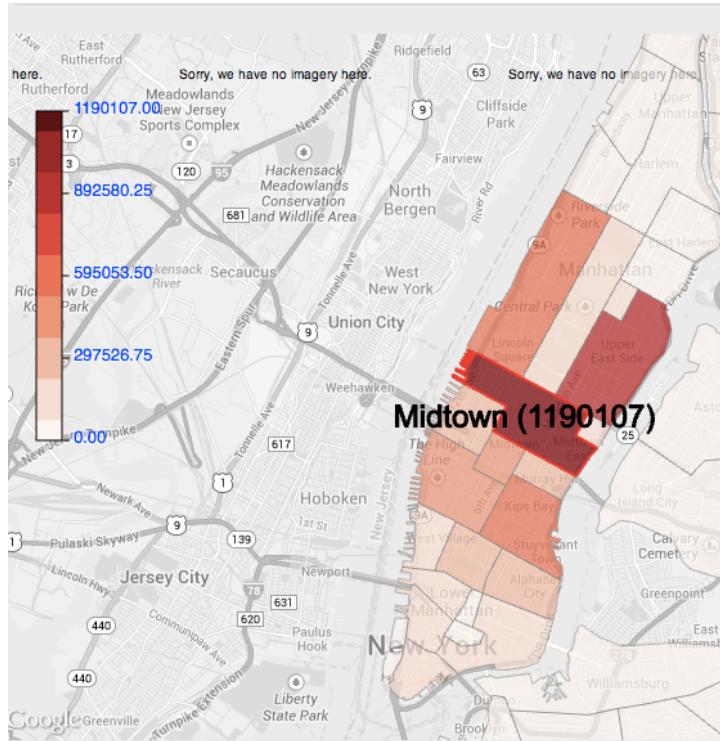
- when + where → what: “*What is the average trip time from Midtown to the airports during weekdays?*”
- when + what → where: “*Where are the hot spots in Manhattan in weekends?*”
- where + what → when: “*When were activities restored in Lower Manhattan after the Sandy hurricane?*”



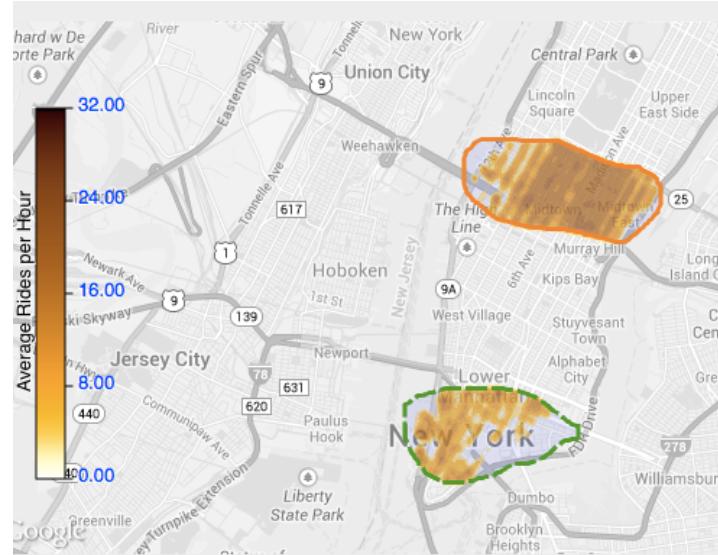
Peuquet's Triad

Model is also able to express other types of queries, including *when* → *what + where*, *where* → *when + what*, and *what* → *where + when*

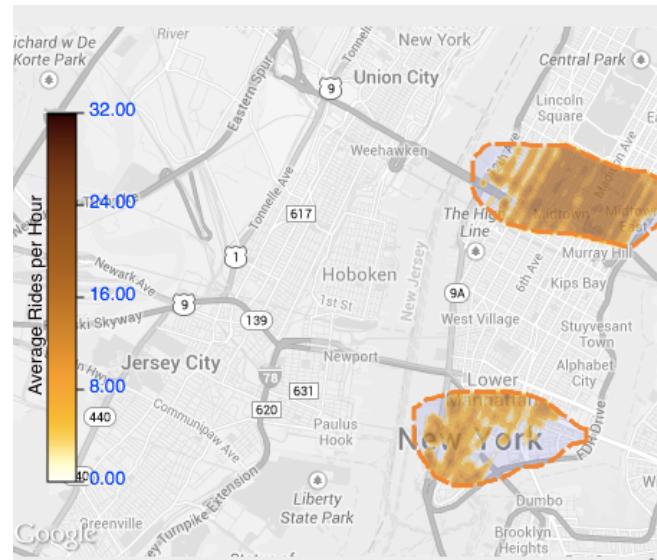
# Selecting Regions – Spatial Constraints



Predefined polygons, e.g.,  
zip, neighborhoods, etc



Free  
selection



Group  
regions



NYU

TANDON SCHOOL  
OF ENGINEERING

VIDA

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Selecting Time – Temporal Constraints

*Time interval*

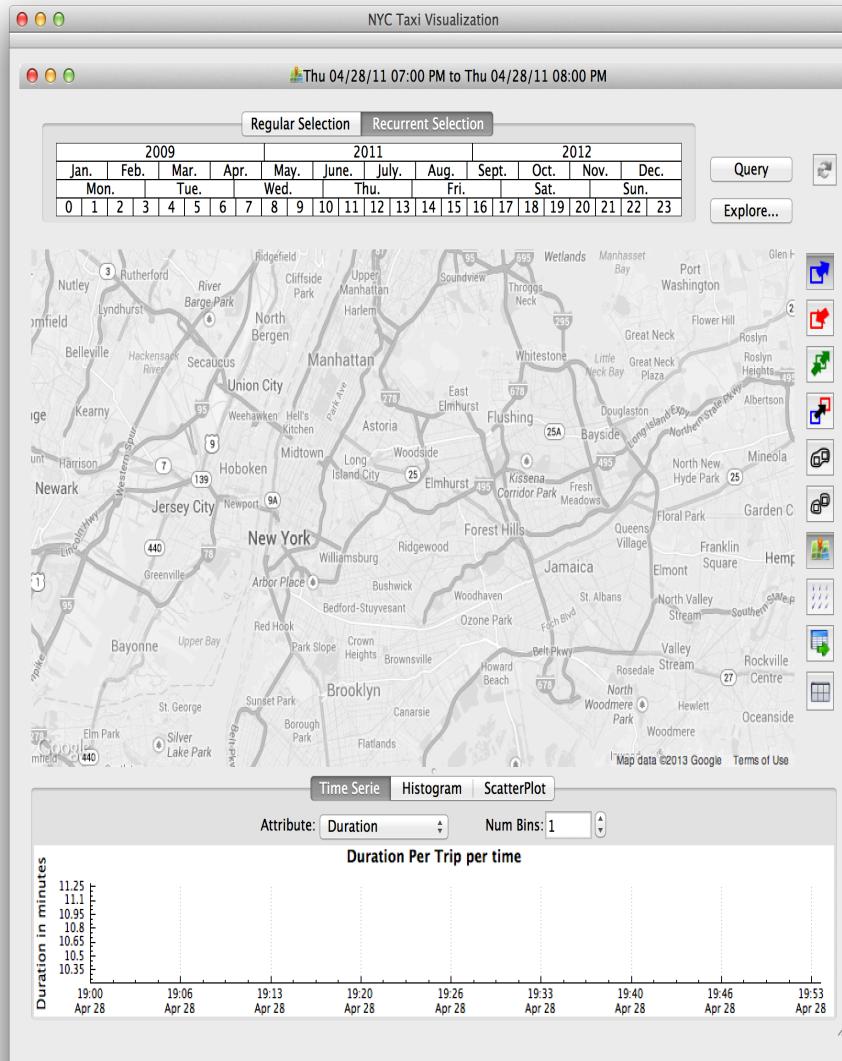
Start Time	Step Size	End Time
<input type="button" value="◀"/> Sun 05/01/11 00:00 <input type="button" value="▶"/>	1 hour	<input type="button" value="◀"/> Sun 05/01/11 01:00 <input type="button" value="▶"/>

2009				2011								2012											
Jan.	Feb.	Mar.	Apr.	May.	June.	July.	Aug.	Sept.	Oct.	Nov.	Dec.	Mon.	Tue.	Wed.	Thu.	Fri.	Sat.	Sun.					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

*Recurrent time patterns*

# When + Where → What

*“What is the average trip time from Midtown to the airports during weekdays?”*



# Composing Queries

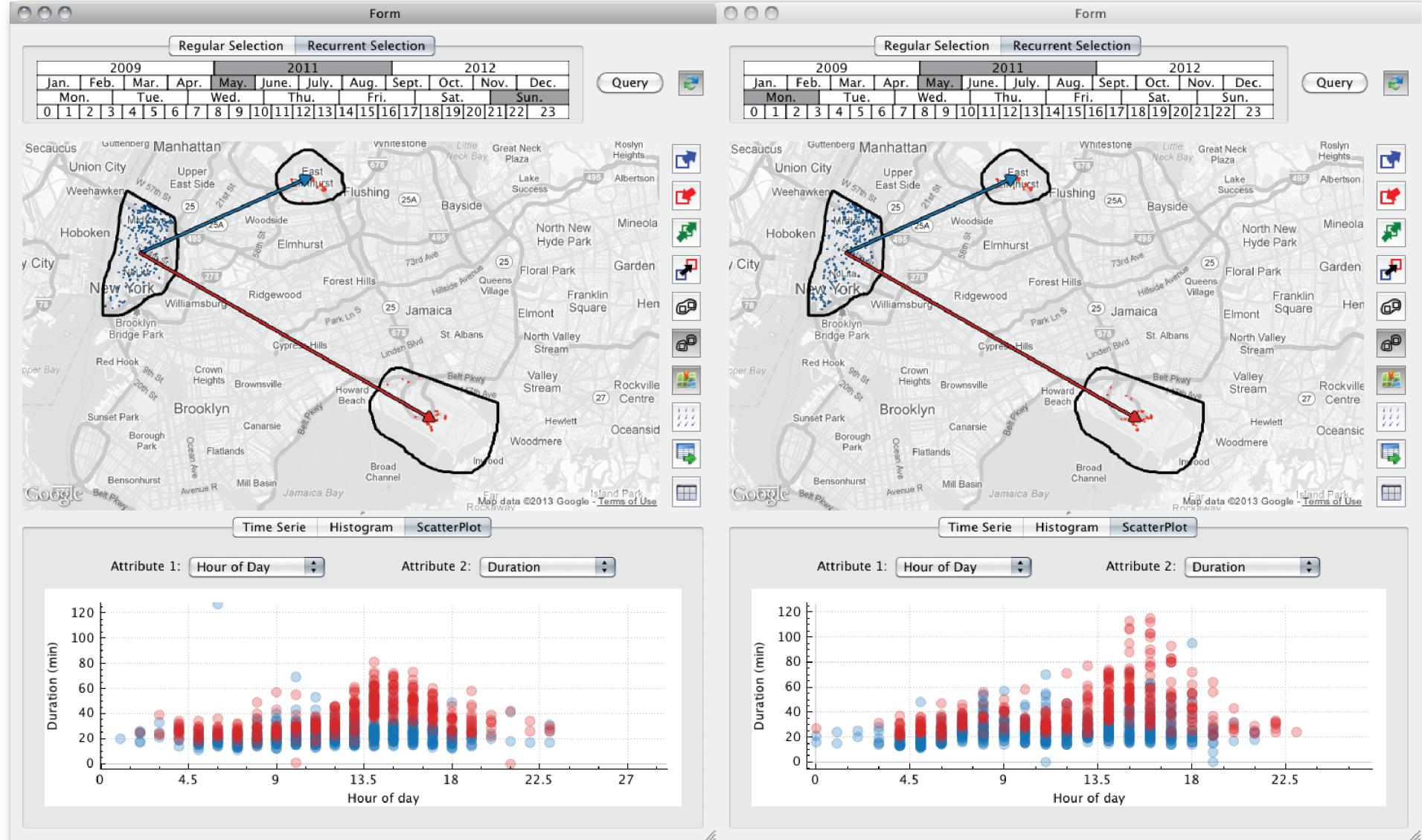
A query is associated with the set of trips contained in its results – queries can be composed.

Different visualizations can be applied to query results

Lines in plot are linked to the queries by their color.



# TaxiVis: Studying Mobility



[Ferreira et al., IEEE TVCG 2013]



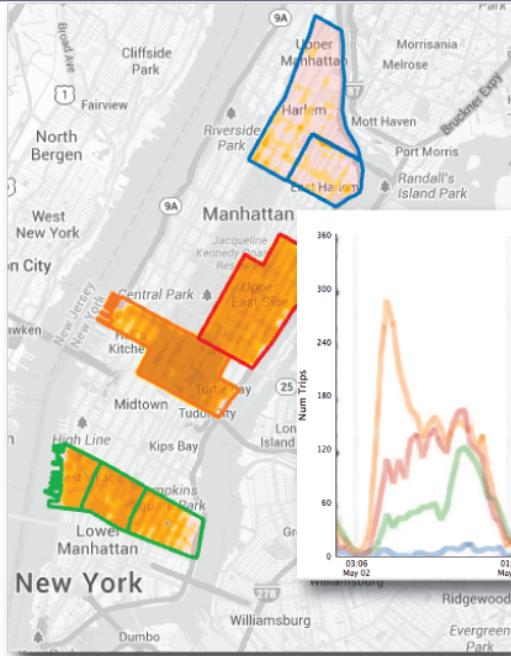
NYU

TANDON SCHOOL  
OF ENGINEERING

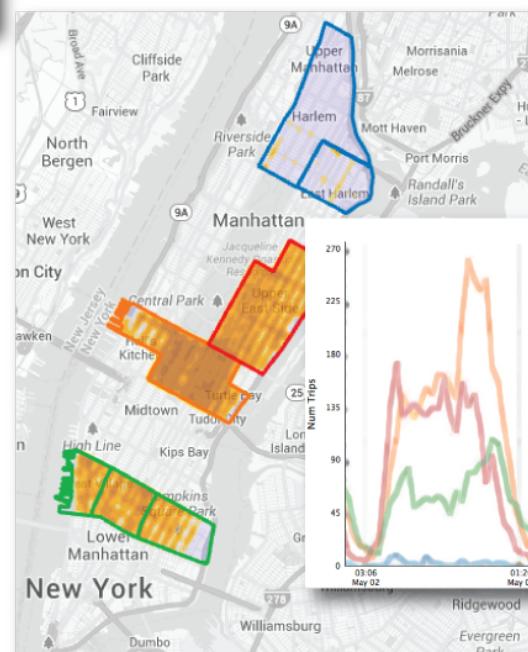
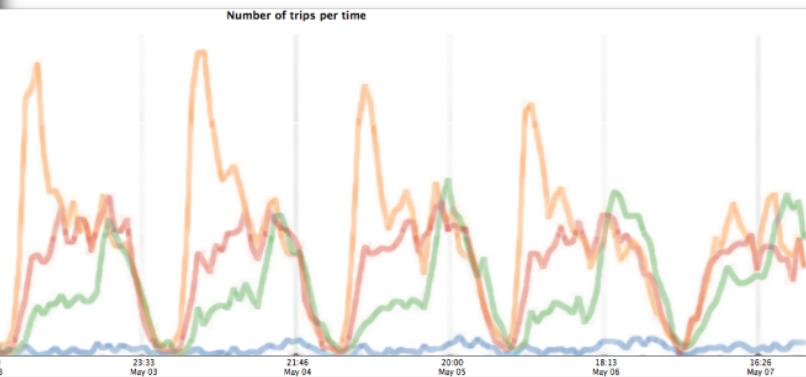


VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

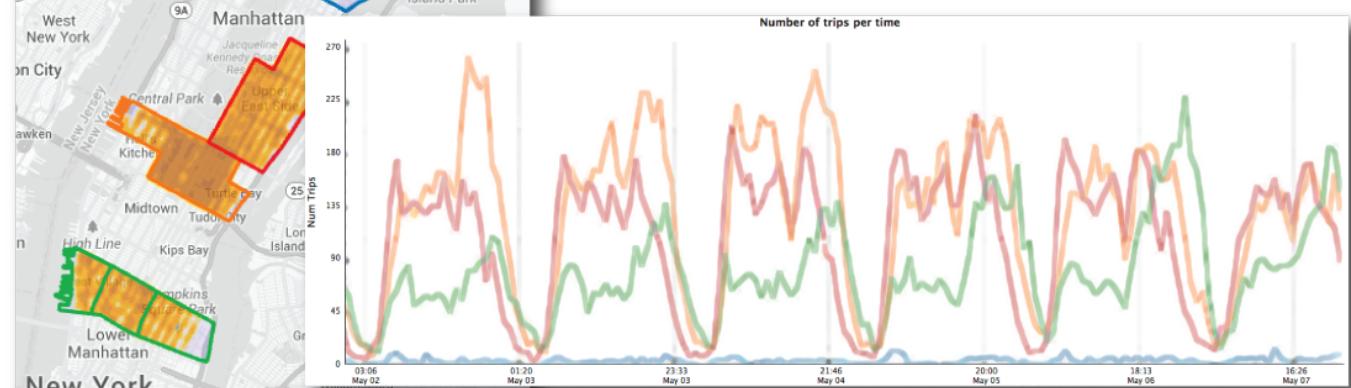
# TaxiVis: Comparing Neighborhoods



*dropoffs*



*pickups*



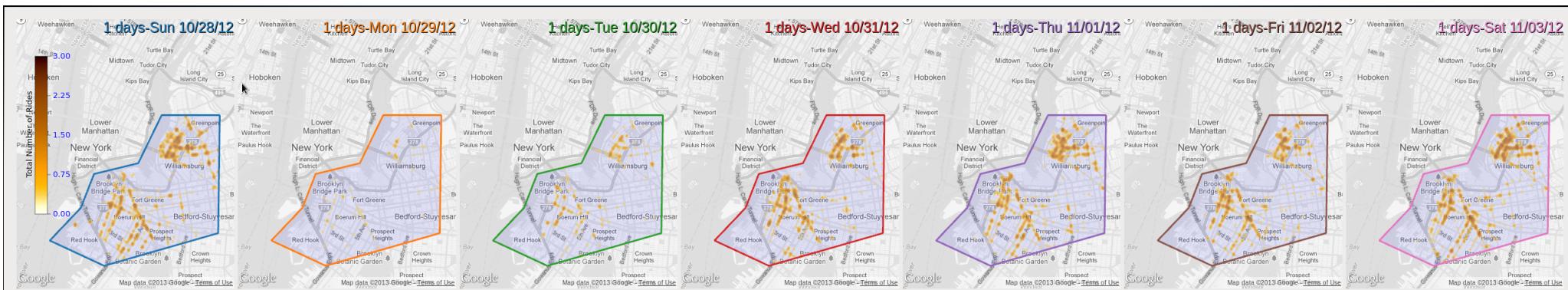
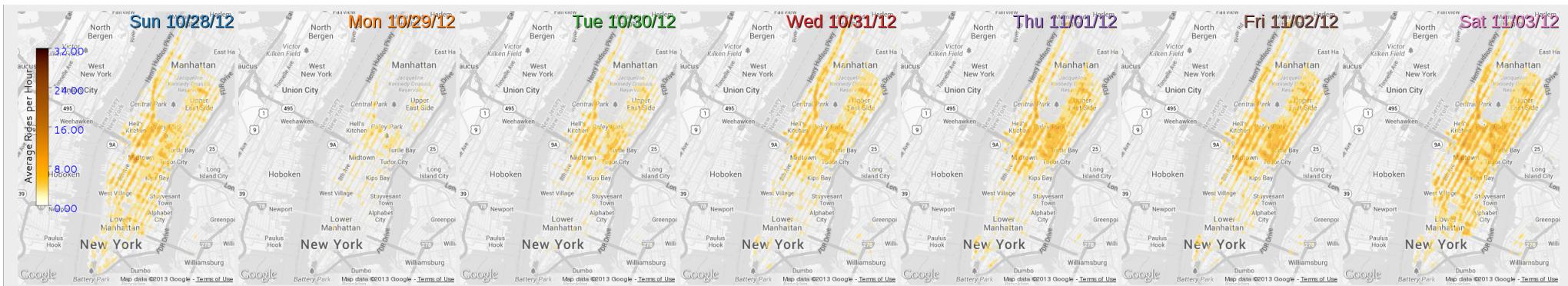
NYU

TANDON SCHOOL  
OF ENGINEERING

VIDA

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Exploring the Effect of Major Events: Sandy



NYU

TANDON SCHOOL  
OF ENGINEERING



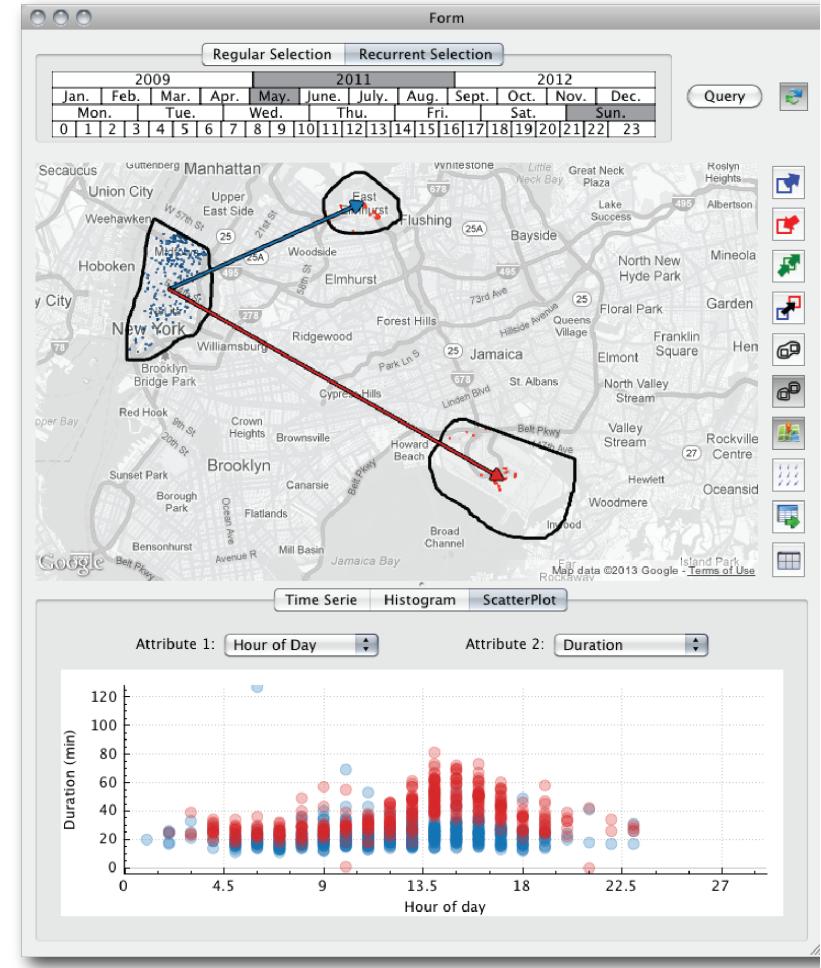
VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Challenge: *Interactive Query Evaluation*

- Typical query:

*Find all trips that occurred between lower Manhattan and the two airports, JFK and LGA, during all Sundays in May 2011*

Query time (sec)	PostgreSQL	ComDB
	503.9	20.6



*“increased latency reduces the rate at which users make observations, draw generalizations and generate hypotheses”*

[Liu and Heer, IEEE TVCG 2014]

# Challenge: *Interactive Query Evaluation*

- Typical query:

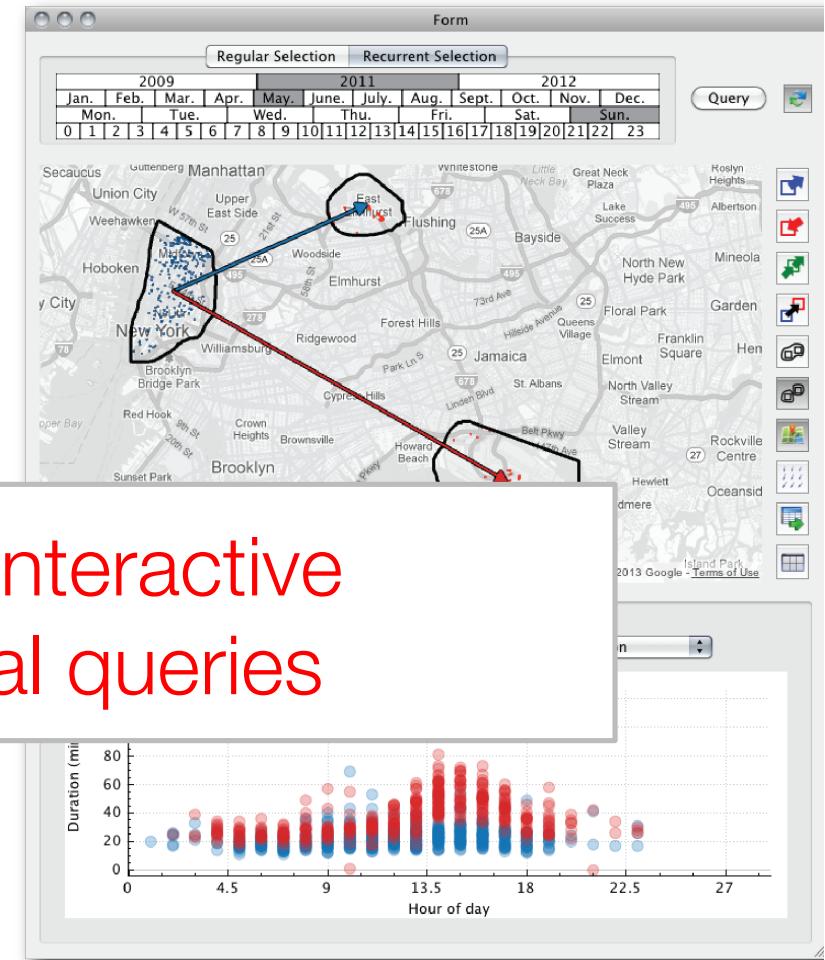
*Find all trips that occurred between lower Manhattan and the two airports, JFK and LGA, during all Sundays in May 2011*

Goal: Support interactive spatio-temporal queries

Query time  
(sec)

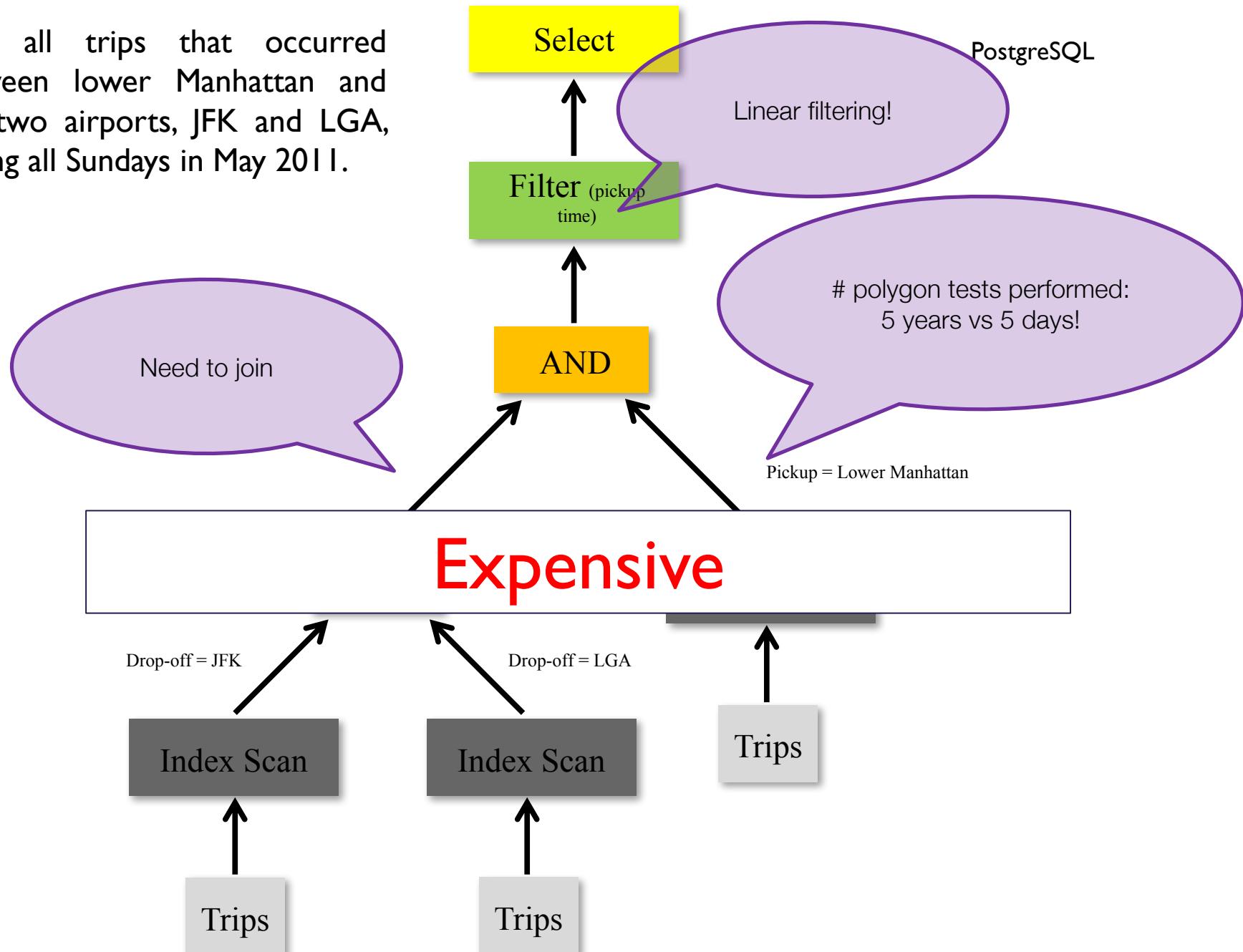
503.9

20.0



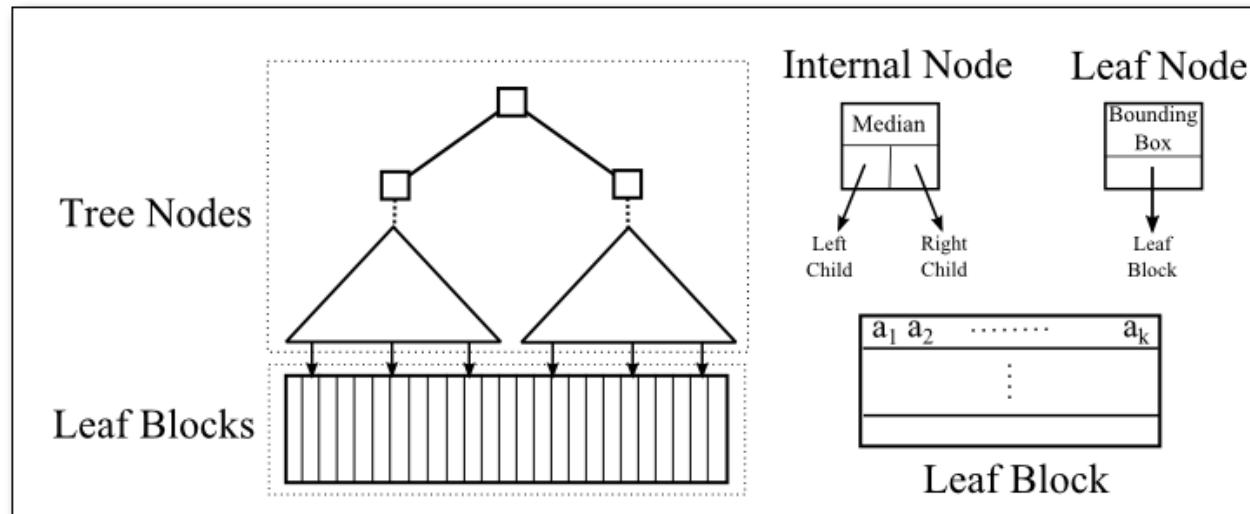
*“increased latency reduces the rate at which users make observations, draw generalizations and generate hypotheses”*

Find all trips that occurred between lower Manhattan and the two airports, JFK and LGA, during all Sundays in May 2011.



# STIG: Supporting Interactive Queries

- Spatio-temporal index based on out-of-core kd-tree using GPUs (STIG)
- Can index and simultaneously filter multiple attributes: avoid joins and reduce the number of point-in-polygon (PIP) tests
- Block-based kd-tree
  - Tree nodes store kd-tree
  - Leaf nodes represent a set of  $k$ -dimensional nodes: Point to a leaf block containing records that satisfy the path constraints



# STIG: Supporting Interactive Queries (cont.)

---

- Spatio-temporal index based on out-of-core kd-tree using GPUs
- Can index and simultaneously filter multiple attributes: avoid joins and reduce the number of point-in-polygon (PIP) tests
- Block-based kd-tree
  - Tree nodes store kd-tree
  - Leaf nodes represent a set of *k*-dimensional nodes: Point to a leaf block containing records that satisfy the path constraints
  - Create *big* blocks – tree is small and fits in memory
  - Use GPU to search the blocks in parallel – speeds up PIP tests
- Source code available at  
<https://github.com/harishd10/mongodb>

# Performance Evaluation

---

Setup:

- 12-core Xeon processor @2.4 GHz
- 8 TB storage
- 256 GB memory
- 3 x NVIDIA GeForce TITAN
- 6 GB memory

# Performance: Taxi Data

Find all trips between Lower Manhattan and  
the two airports, JFK and LGA, during all  
Sundays in May 2011.

Query	MongoDB	PostgreSQL		ComDB	
	Time	Time	Speed up	Time	Speed up
1		503.9		20.6	
2		501.9		23.3	
3		437.8		21.6	
4		437.1		32.6	

Time in Seconds

868 million trips; ~13k results/query

# Performance: Taxi Data

Find all trips between Lower Manhattan and the two airports, JFK and LGA, during all Sundays in May 2011.

Query	MongoDB	PostgreSQL	ComDB		
	Time	Time	Speed up	Time	Speed up
1	0.075	503.9	6718	20.6	274
2	0.080	501.9	6273	23.3	291
3	0.067	437.8	6534	21.6	322
4	0.070	437.1	6244	32.6	465

Time in Seconds  
868 million trips; ~13k results/query

# Performance: Twitter Data

---

Query	MongoDB	PostgreSQL		ComDB	
	Time	Time	Speed up	Time	Speed up
1	0.246	161.2	655	109.6	445
2	0.288	151.2	525	157.7	547
3	0.558	286.0	512	216.8	388

Time in Seconds

1.1 billion tweets; 130k-370k results/query

# TaxiVis: Status

- Demoed to NYC DOT and TLC

----- Forwarded message -----

From: [REDACTED]@tlc.nyc.gov>

Date: Thu, Oct 24, 2013 at 4:58 PM

Subject: NYC taxi data

To: "Claudio Silva (csilva@nyu.edu)" <csilva@nyu.edu>, "Huy Vo (huy.vo@nyu.edu)" <huy.vo@nyu.edu>, "Caryn Joy Knutsen (caryn.knutsen@nyu.edu)" <caryn.knutsen@nyu.edu>, "Kim Alfred (kim.alfred@nyu.edu)" <kim.alfred@nyu.edu>  
[REDACTED]>

Hi all,

First, I would like to thank you all for coming to data. We were truly blown away! In fact, we have product like the one you've demonstrated to us  
[REDACTED]  
[REDACTED]

for us on Monday. We think that could be a great future use for our data in combination with othe

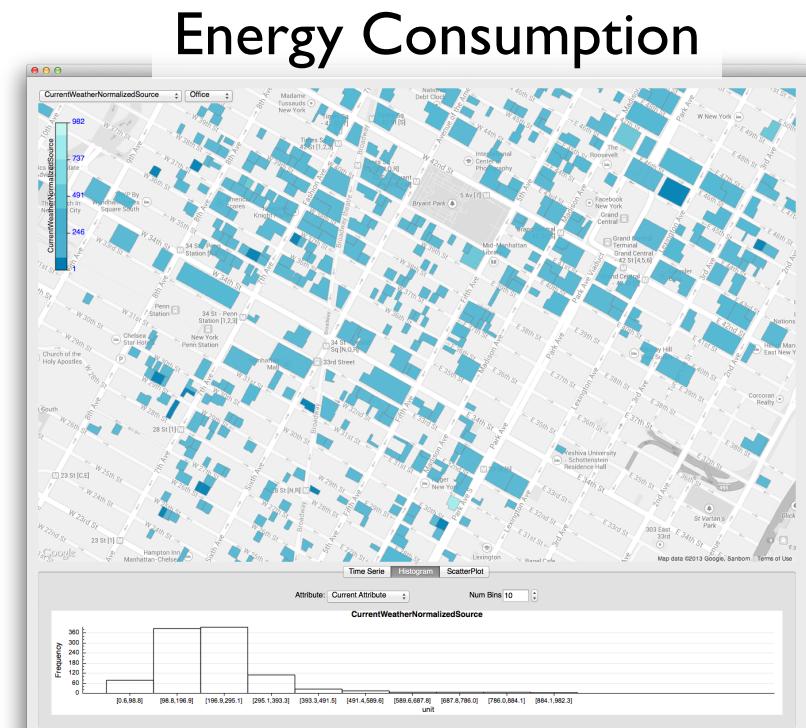
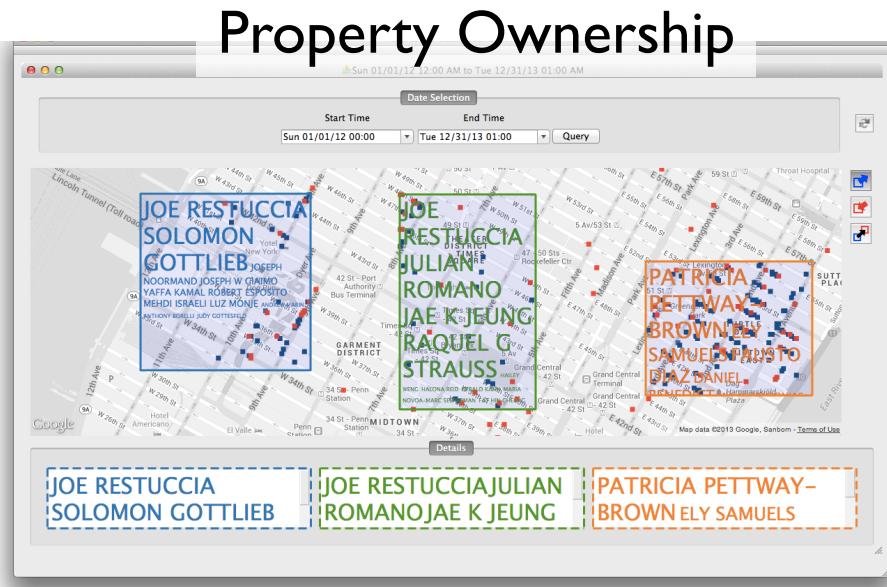
*"The speed at which the tool permits us to work has saved multiple hours of staff time and has dramatically improved the unit's output and capabilities."*

Assistant Commissioner, DoT

Cheers,

# TaxiVis: Status

- Demoed to NYC DOT and TLC
- System is open source
  - <https://github.com/ViDA-NYU/TaxiVis>
- Used to explore different data sets



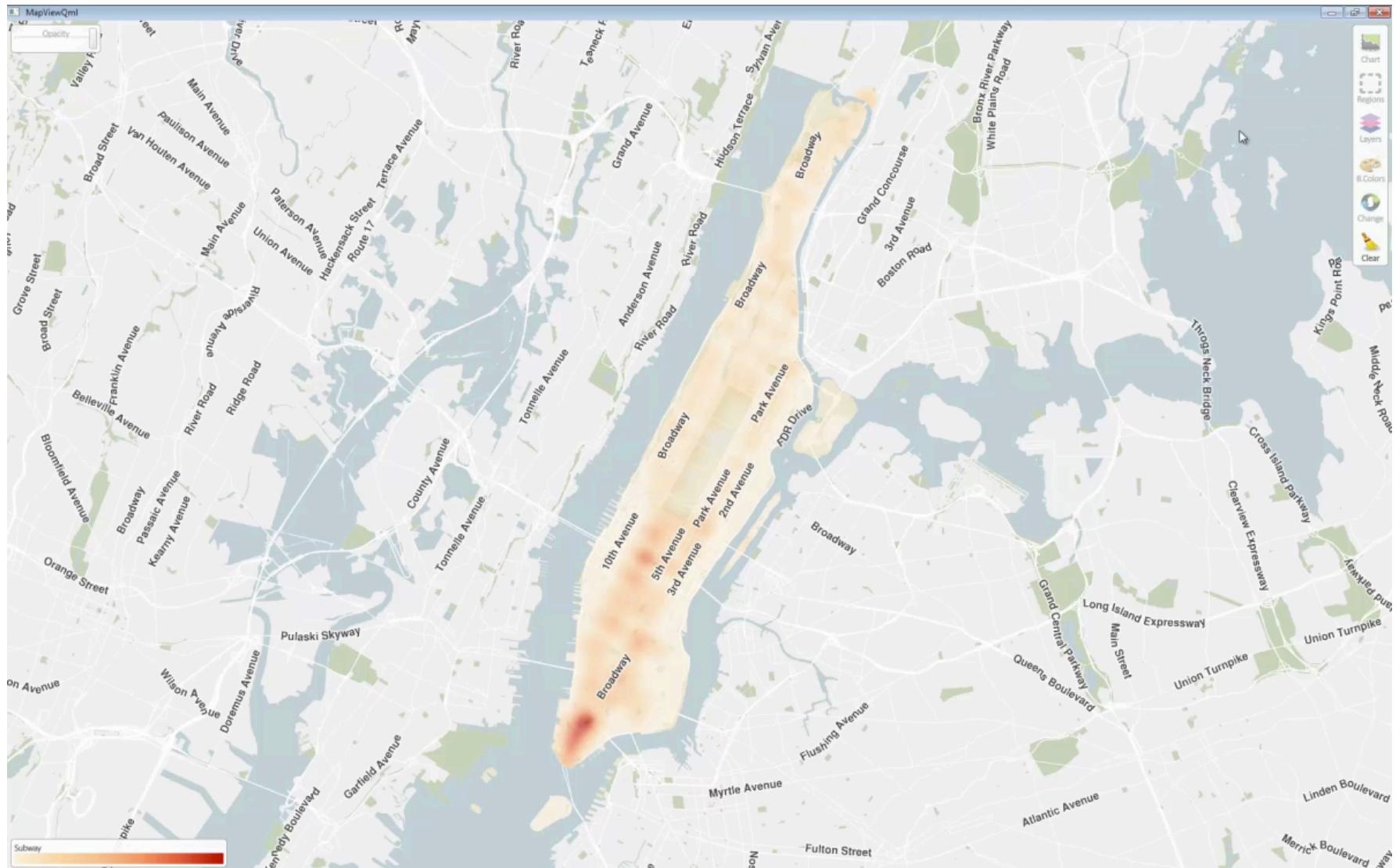
NYU

TANDON SCHOOL  
OF ENGINEERING

ViDA

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# What Next: Urbane



[https://www.youtube.com/watch?v=\\_B35vxCgDw4&feature=youtu.be](https://www.youtube.com/watch?v=_B35vxCgDw4&feature=youtu.be)

[Ferreira et al., IEEE VAST 2015]



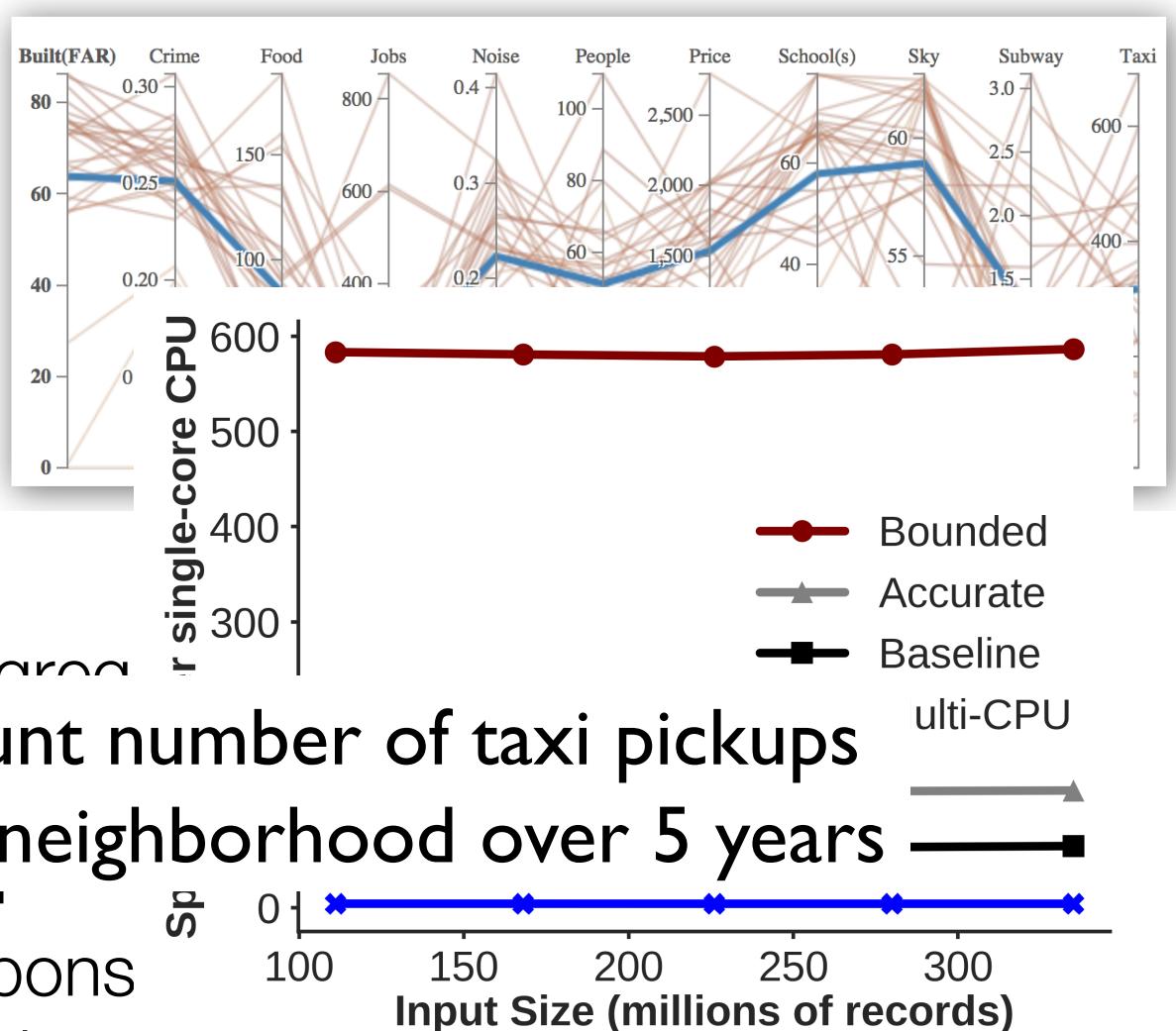
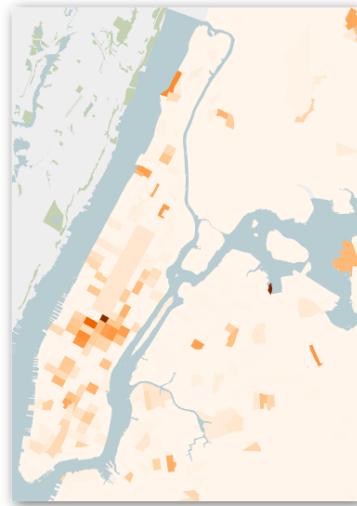
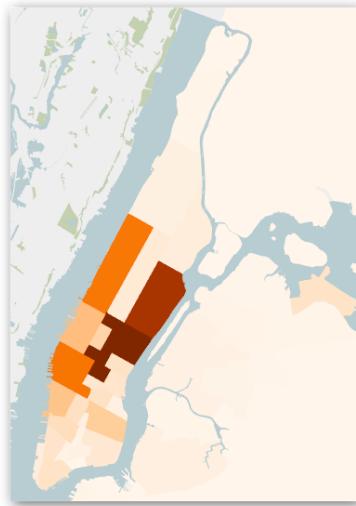
NYU

TANDON SCHOOL  
OF ENGINEERING



VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Spatial Aggregation



- Convert spatial join + aggregate operations to count number of taxi pickups in each NYC neighborhood over 5 years
- Exploit spatial locality
- Support interactive responses on commodity laptops/desktops

# A Unified Index for Spatio-Temporal Keyword Queries

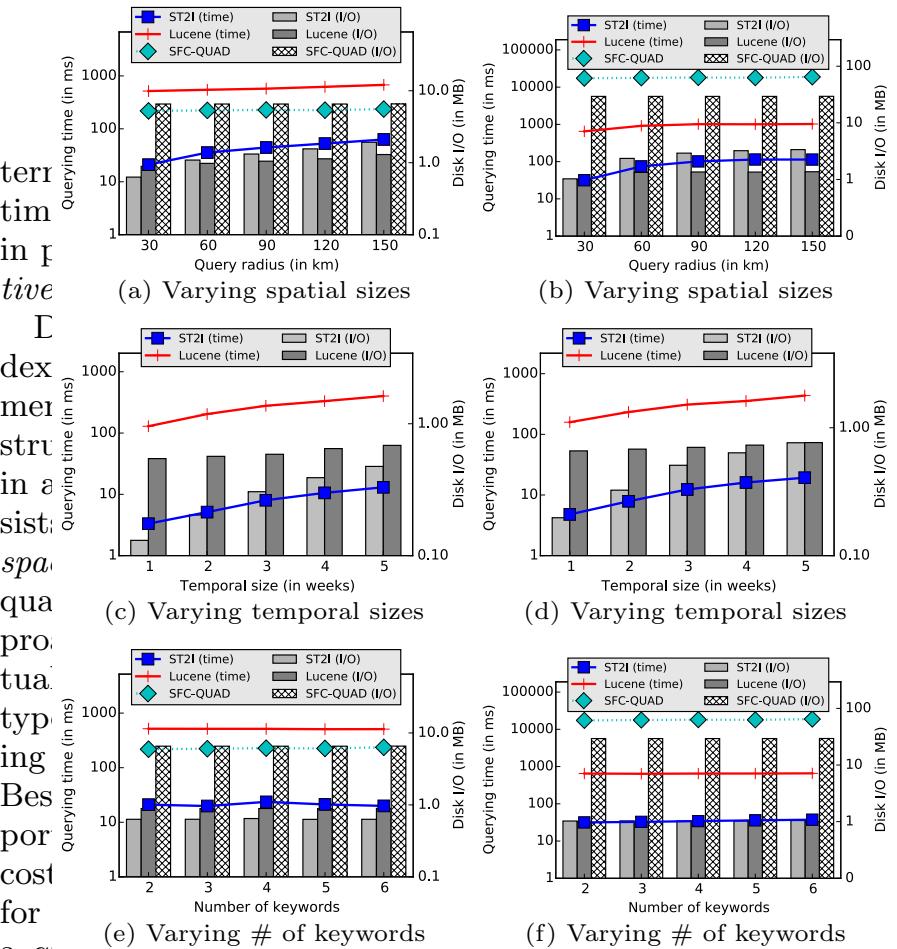
Tuan-Anh Hoang-Vu  
New York University  
tuananh@nyu.edu

Huy T. Vo  
The City College of New York  
hvo@cs.ccny.cuny.edu

Juliana Freire  
New York University  
juliana.freire@nyu.edu

## ABSTRACT

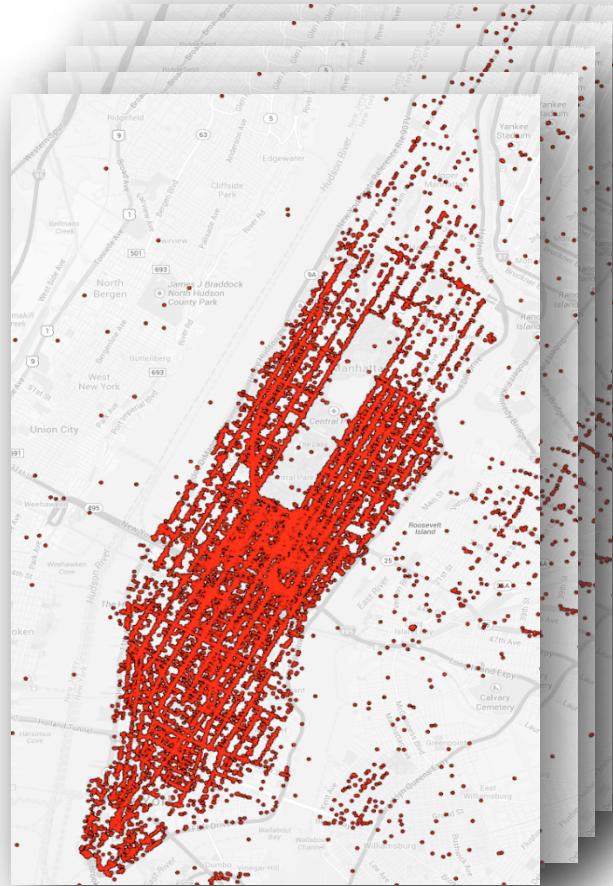
From tweets to urban data sets, there has been an explosion in the volume of textual data that is associated with both temporal and spatial components. Efficiently evaluating queries over these data is challenging. Previous approaches have focused on the spatial aspect. Some used separate indices for space and text, thus incurring the overhead of storing separate indices and joining their results. Others proposed a combined index that either inserts terms into a spatial structure or adds a spatial structure to an inverted index. These benefit queries with highly-selective constraints that match the primary index structure but have limited effectiveness and pruning power otherwise. We propose a new indexing strategy that uniformly handles text, space and time in a single structure, and is thus able to efficiently evaluate queries that combine keywords with spatial and temporal constraints. We present a detailed experimental evaluation using real data sets which shows that not only our index attains substantially lower query processing times, but it can also be constructed in a fraction of the time required by state-of-the-art approaches.



a query was posted at a given location, containing a term and that was posted at given location

# Exploring Data: Finding Interesting Features

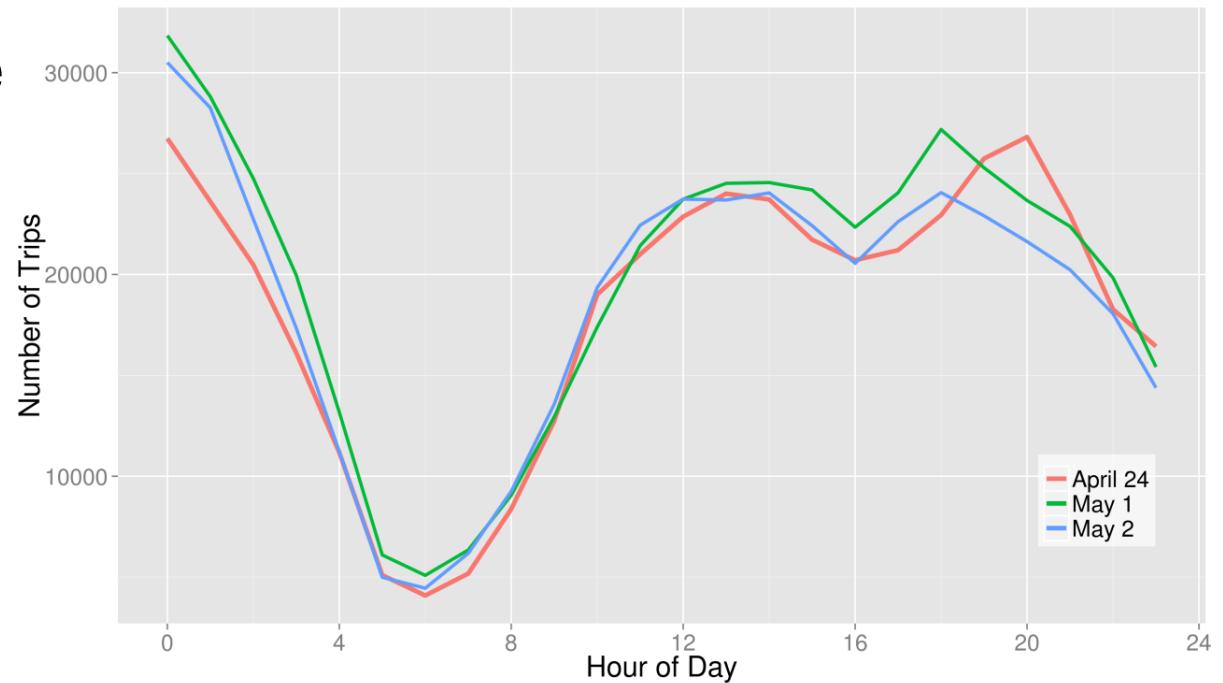
# Taxi Data: Too Many Slices



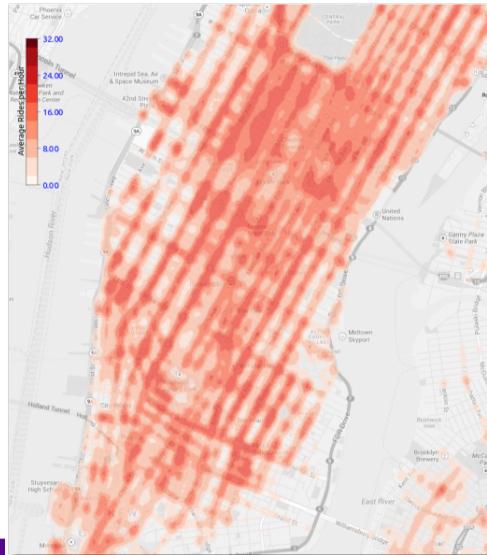
- 365\*24 1-hour slices in one year
- Which slices are interesting?

# Reducing the Number of Slices

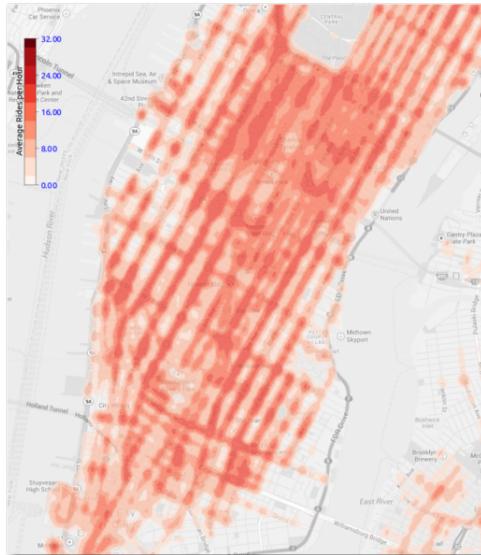
*Aggregate over space*



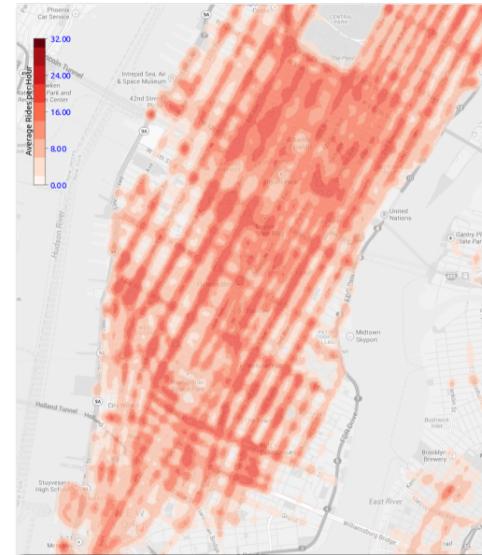
*Aggregate over time*



April 24



May 1



May 8



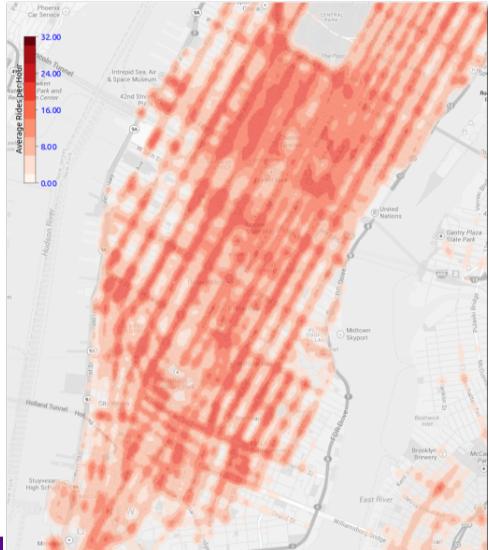
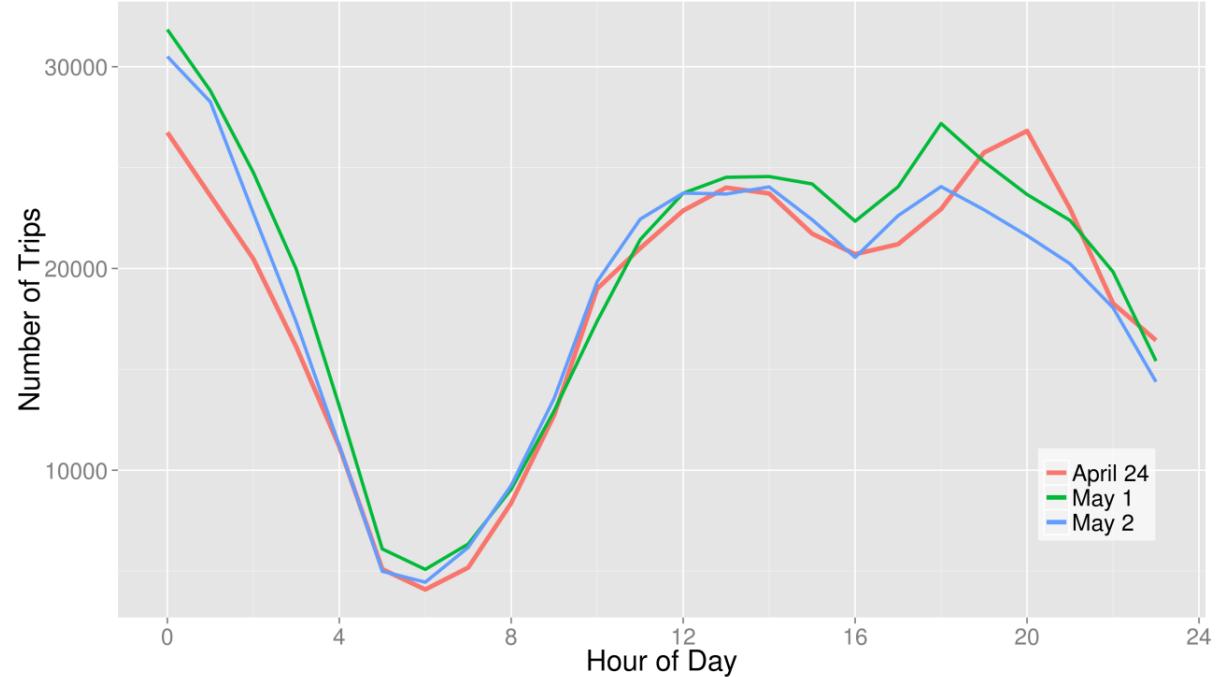
NYU

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

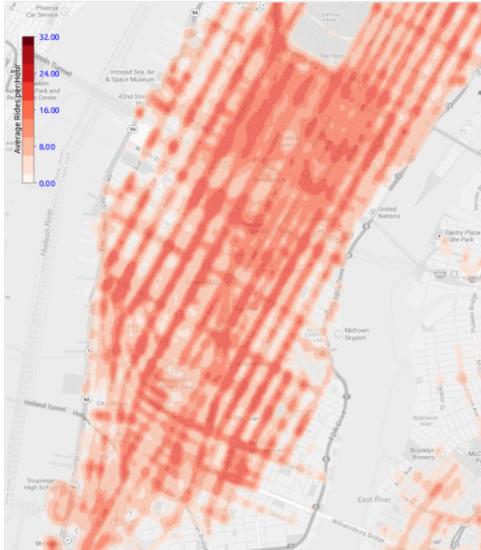
# Miss Interesting Slices



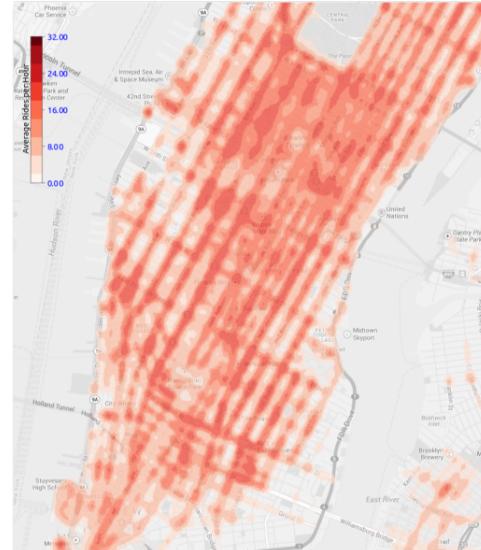
May 1 (8-9am)



April 24



May 1



May 8

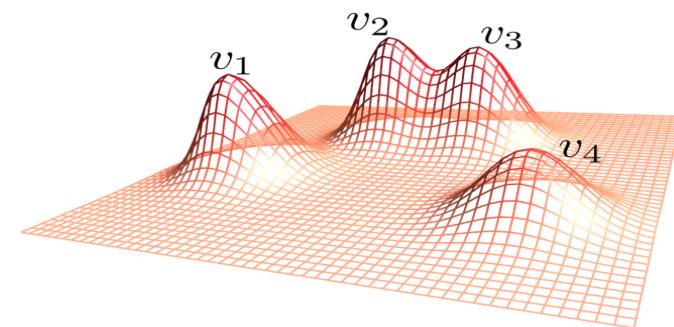


NYU

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

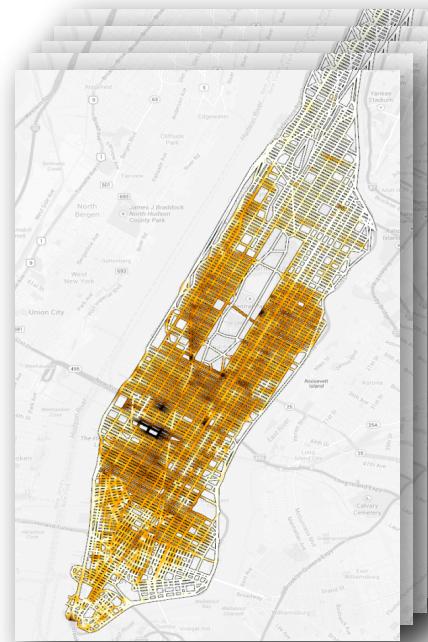
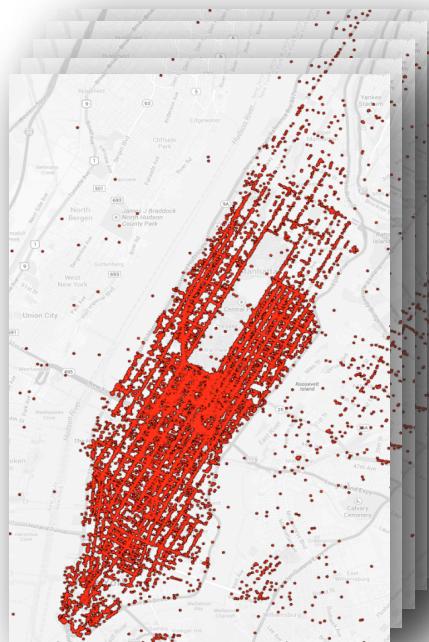
# Finding Interesting Slices

- Desiderata: automatically identify *events* with arbitrary spatial structure and at multiple temporal scales
- Our solution: Use computational topology techniques to efficiently discover events
  - Model data as a time-varying scalar function
  - Identify regions of maxima and minima: a region is *interesting* if its behavior differs from that of its neighborhood



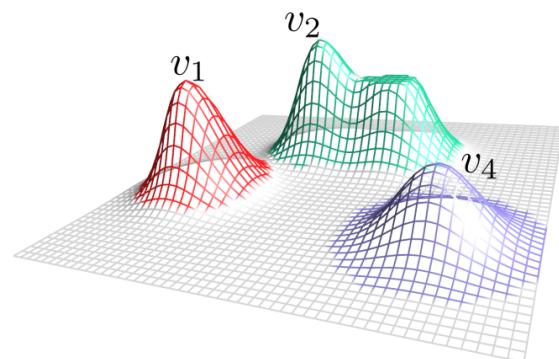
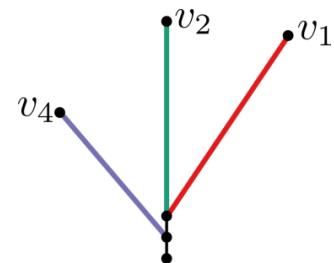
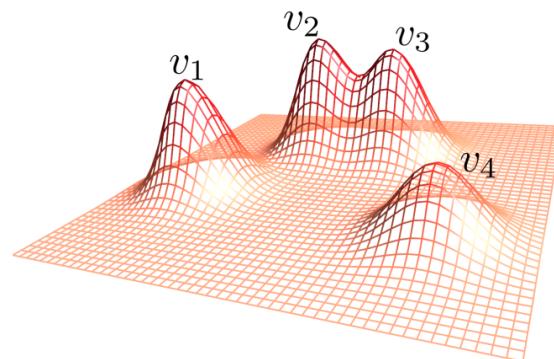
# Identifying Potential Events

- Model data as a time-varying scalar function defined on a graph
  - $f : G \rightarrow \mathbb{R}$
  - Taxi data: Graph = road network; Function = density of taxis
  - Subway data: Graph = track network; Function = delay of trains



# Identifying Potential Events

- Use Merge Trees to efficiently identify events in each time step
- Compute the regions corresponding to the set of *maxima* and *minima* – *the set of potential events*
  - Intuition: a region is interesting if its behavior differs from that of its neighborhood
  - Unimportant events can be simplified



# Taxi Data: Potential Events

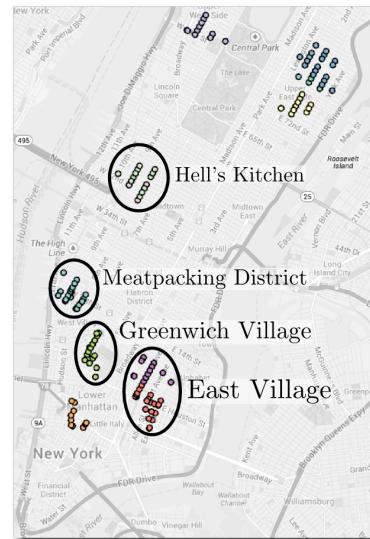
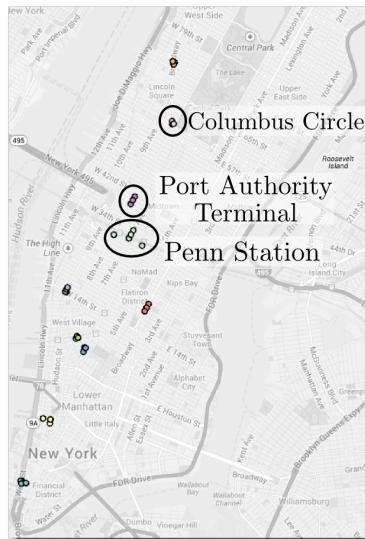
- Minima: lack of taxis
  - Regions where density is lower than local neighborhood
  - Could denote road blocks, e.g., Macy's parade



*Scalar function corresponding  
to the time step 10 am-11 am  
on 24 November 2011*

# Taxi Data: Potential Events

- Minima: lack of taxis
  - Regions where density is lower than local neighborhood
  - Could denote road blocks, e.g., Macy's parade
- Maxima: popular taxi locations
  - Regions where density is higher than local neighborhood
  - Could denote tourist locations, train stations



NYU

TANDON SCHOOL  
OF ENGINEERING

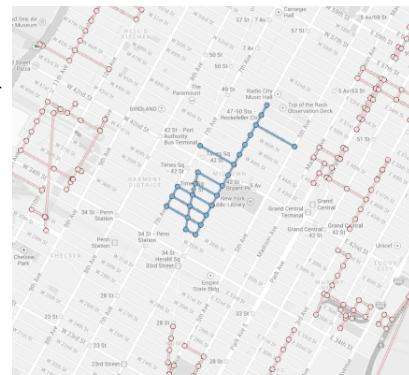
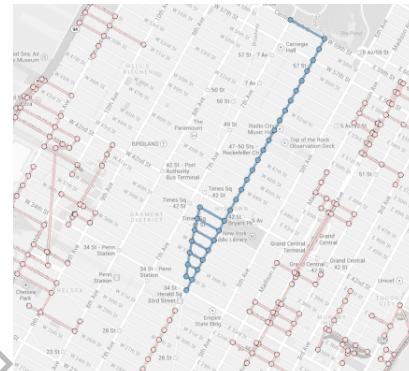
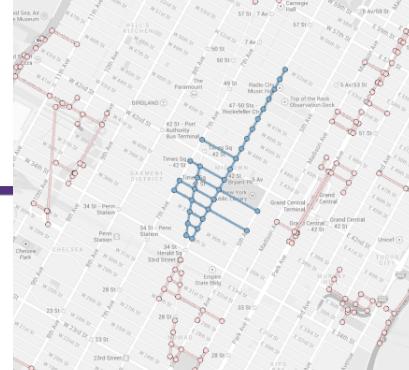
# Querying Events



5 Borough Bike  
Tour 2011  
(1 May 2011)



Query



Dominican Day Parade 2011  
(14 August 2011)



5 Borough Bike Tour 2012  
(6 May 2012)



Dominican Day Parade 2012  
(12 August 2012)



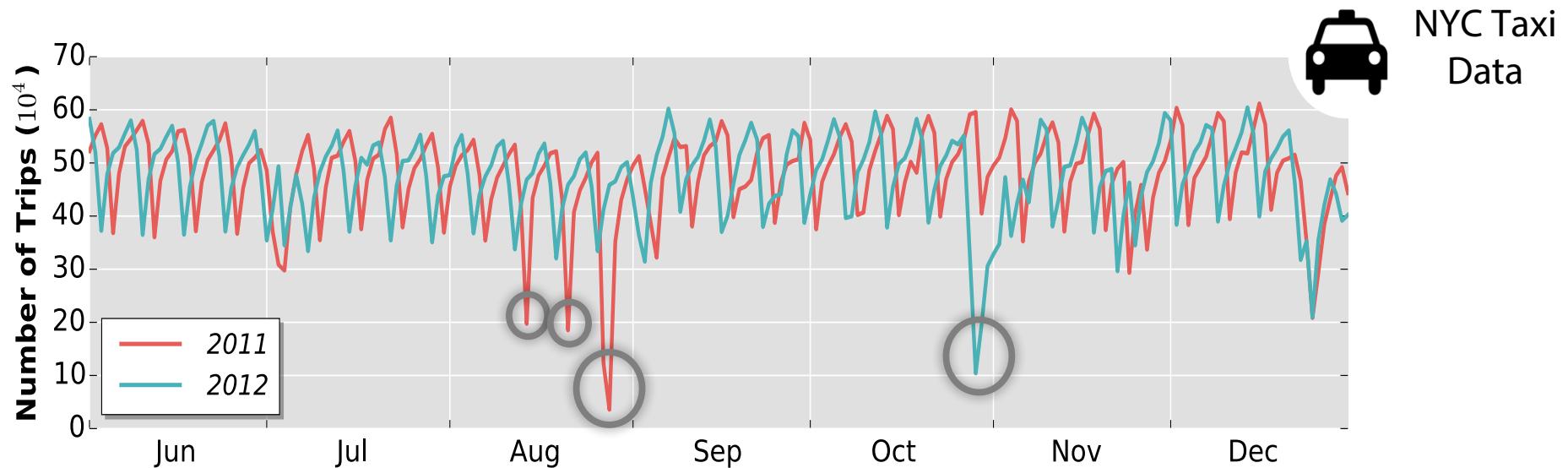
Gaza Solidarity Protest NYC  
(18 November 2012)



NYU | TANDON SCHOOL  
OF ENGINEERING

# Using Data to Explain Data

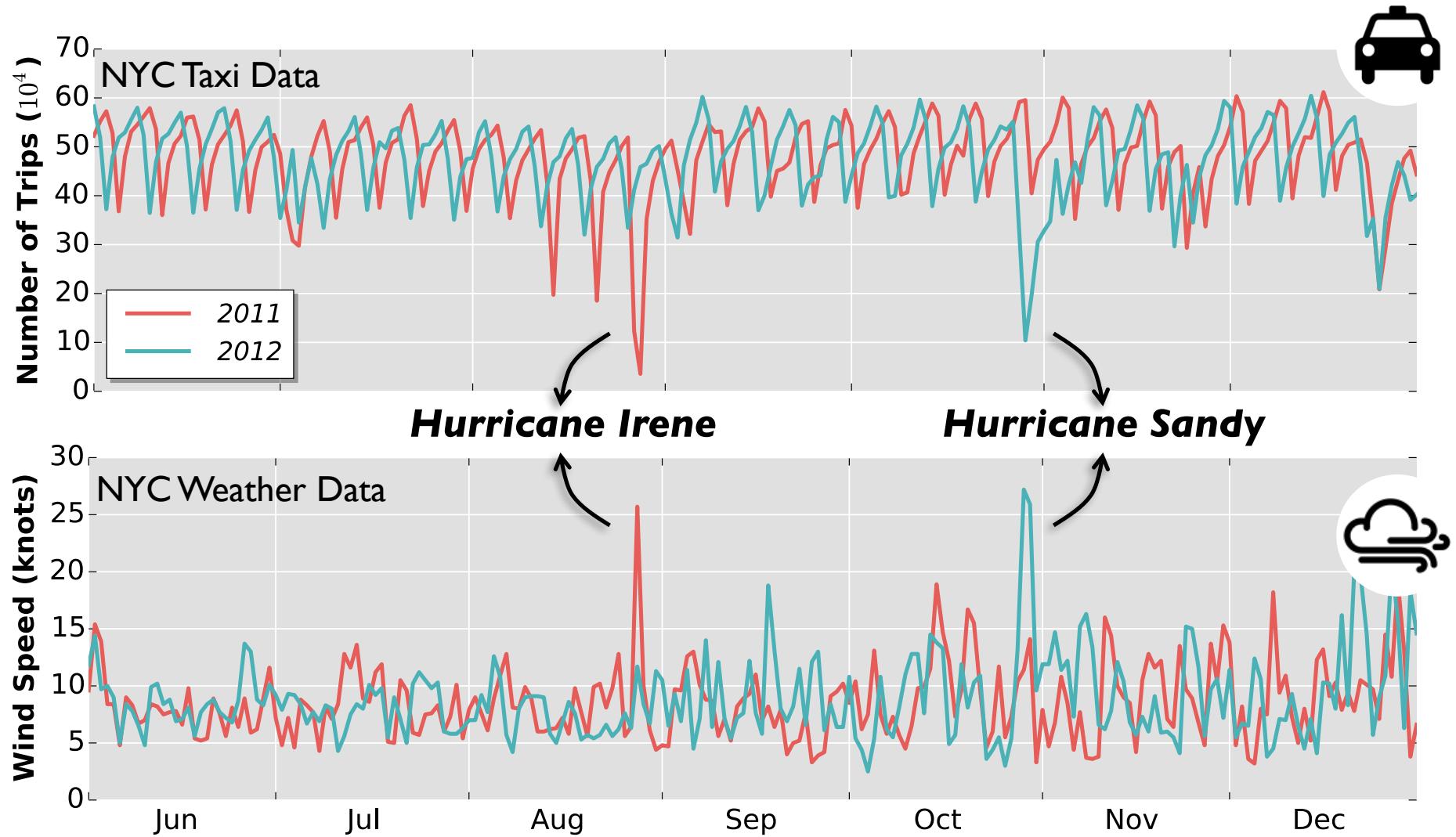
# Explaining Events



- Are these big drops data quality issues in the data?
- Or do they correspond to *real* events?

Find all data sets related to the Taxi data set

# Using Data to Explain Events



**NYU**

TANDON SCHOOL  
OF ENGINEERING

**VIDA**  
VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Using Data to Explain and Predict NYC

1. Would a reduction in traffic speed reduce the number of accidents? What other factors contribute to accidents?
2. Why it is so hard to find a taxi when it is raining?

<http://nymag.com/daily/intelligencer/2014/11/why-you-can't-get-a-taxi-when-it's-raining.html>

**MINT Intelligencer**

## Why You Can't Get a Taxi When It's Raining

By Annie Lowrey [Follow @AnnieLowrey](#)



Good luck, lady. Photo: Jacobs Stock Photography/Getty Images

It's pouring rain. You're running late. You desperately want to take a cab to the office. But, of course, there are none to be found. Happens all the time, right? Right, says science — or, to be specific, a new and exhaustive economic analysis of New York City taxi rides and Central Park meteorological data.

# Urban Data Interactions

---

By uncovering **relationships** between data sets, we can

- Better understand a city and how its different components interact
- Discover important attributes that can inform the construction of predictive models

# Where to start?

---

- Data are available!
- Answers are likely in the data
- But there are too many data sets, and even more attributes to consider



**NYC OpenData**

1,200 data sets  
(and counting)

8 attributes  
per data set



**weather**

> 200 attributes

Which data sets to analyze?

# The Data Polygamy Framework

- **Discover relationships** between data sets to better understand urban data and how the different components of city interact
- Each data set can be related to **zero or more** data sets through several attributes

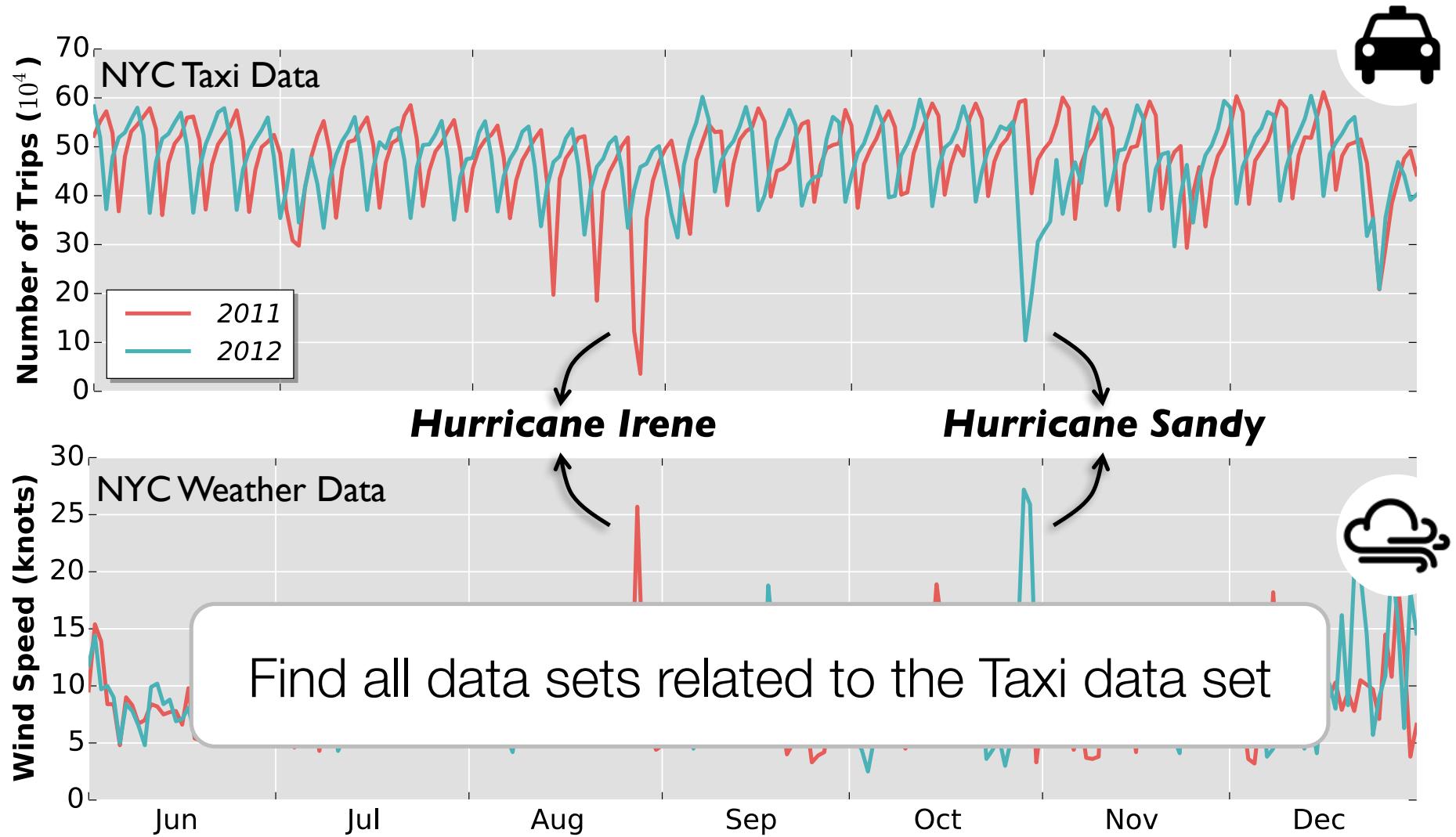
*Data sets are polygamous!*

- Guide users in **data discovery and analysis** by allowing them to pose **relationship queries**

*Find all data sets related to a given data set ID*

- **Support both hypothesis generation and testing**  
[Chirigati et al., ACM SIGMOD 2016]

# Hypothesis Generation



# Hypothesis Testing

- Hypothesis: Taxi drivers set an income goal, and on rainy days they reach the goal faster

*Taxi Fare*



*Precipitation*

**DAILY Intelligencer**

## Why You Can't Get a Taxi When It's Raining

By Annie Lowrey [Follow @AnnieLowrey](#)



Good luck, lady. Photo: Jacobs Stock Photography/Getty Images

It's pouring rain. You're running late. You desperately want to take a cab to the office. But, of course, there are none to be found. Happens all the time, right? Right, says science — or, to be specific, a new and [exhaustive economic analysis](#) of New York City taxi rides and Central Park meteorological data.

<http://nymag.com/daily/intelligencer/2014/11/why-you-can't-get-a-taxi-when-it's-raining.html>



**NYU**

TANDON SCHOOL  
OF ENGINEERING

**VIDA**

VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

# Challenge: Defining a Relationship

---

- Must take both **space** and **time** into account
- How and when are two data sets related?
  - Some relationships become visible through *atypical* behavior
- Conventional techniques (e.g., Pearson's correlation, mutual information, DTW, etc.) take into account the entire data and miss relationships that occur **only at certain times/locations**
  - E.g., most of the time, taxi trips and wind speed are not related

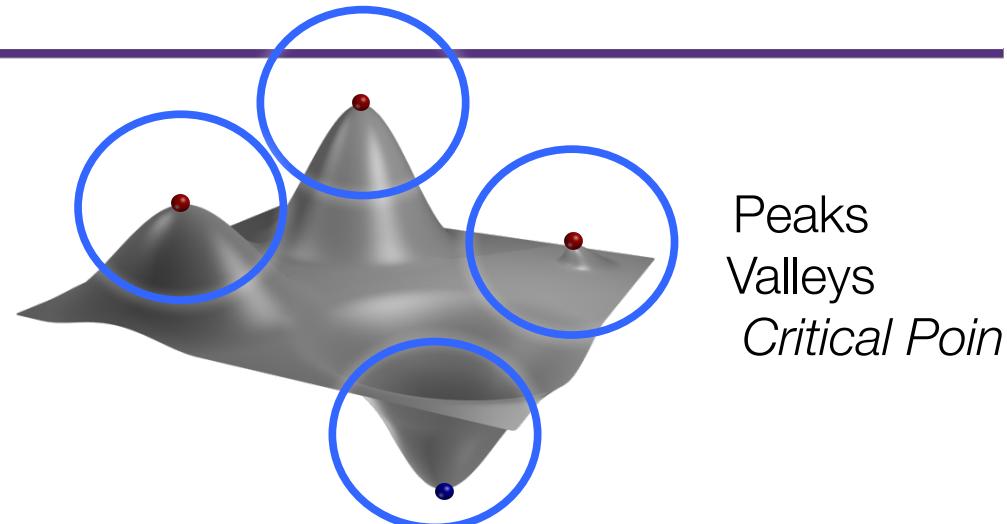
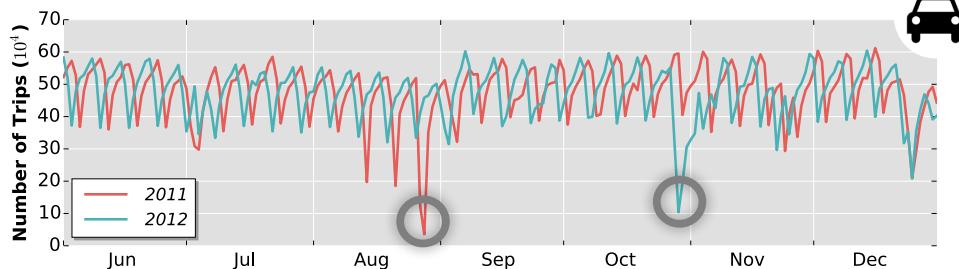
# Challenge: Identifying Meaningful Relationships

- Many data sets, each consisting of many attributes
  - Relationships can be between any of the attributes
  - Weather data: >200 attributes; NYC Open data: 8 attributes per data set on an average
- Data sets can be large
  - Taxi data: 180M trips per year
  - Data at multiple spatio-temporal different resolutions
  - Combinatorially large number of relationships to evaluate
    - ~2.4 million possible relationships among NYC Open Data alone for a single spatio-temporal resolution

meaningful relationship  $\longleftrightarrow$  needle in a haystack

# How and when are two data sets related?

- Topology-based relationships

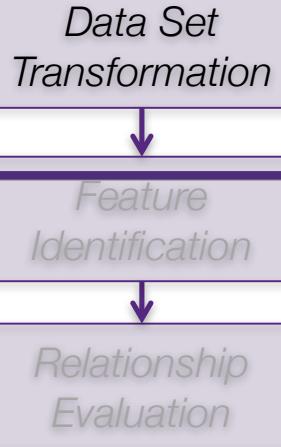


- Topological representation captures **salient features** of the data

A salient feature is a spatio-temporal region whose behavior differs from its neighborhood

- Supports **arbitrary** spatial structures and time intervals
- Efficient feature identification through Merge Tree Index
- **Definition:** Two data sets are related if their **salient features** are related

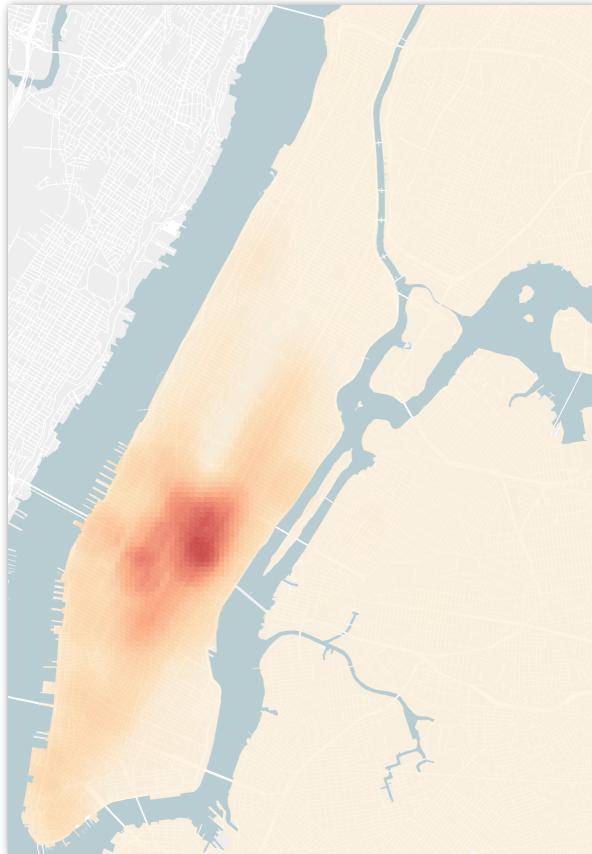
# Data Set to Scalar Functions



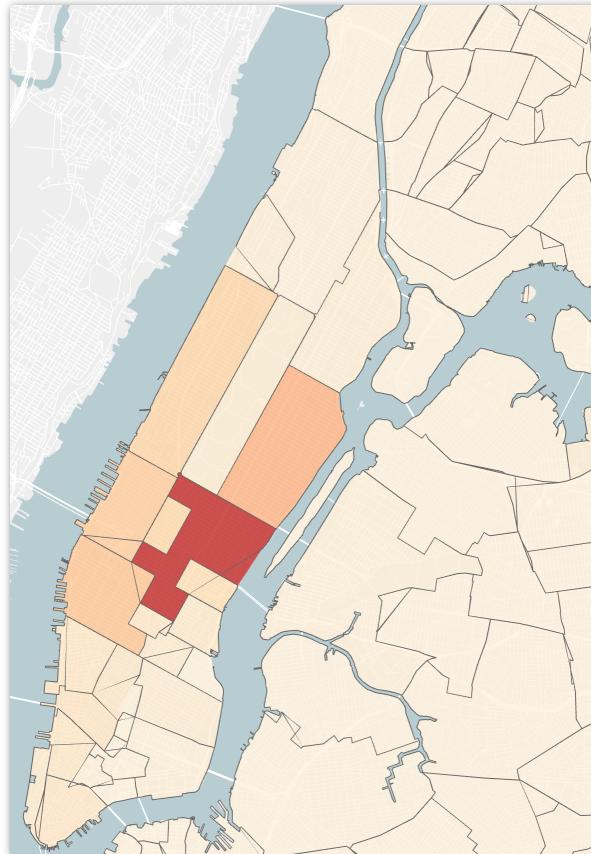
- Each data set represented as a set of time-varying scalar functions
  - $f : [\mathbb{S} \times \mathbb{T}] \rightarrow \mathbb{R}$
  - Maps each point in space and time to a real value
- Different functions can be used
  - Count: Captures the activity of an entity corresponding to the data, e.g., number of trips, number of unique taxis, etc.
  - Attribute: Captures attribute value variation, e.g., average taxi fare
- Functions computed at all possible resolutions

# Data Set to Scalar Functions

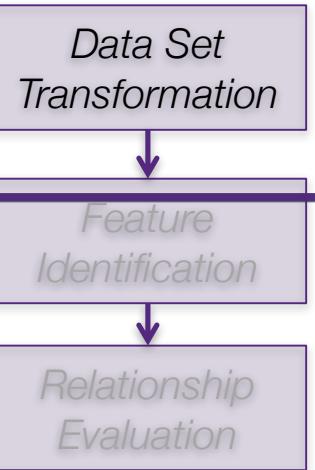
- Example: Density function of Taxi data



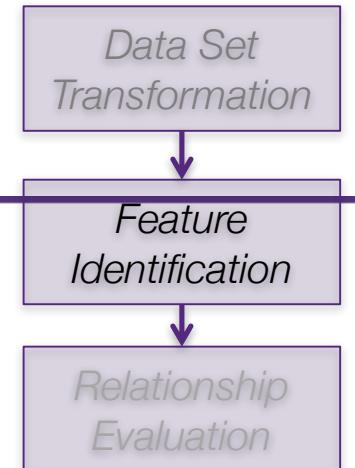
$\mathbb{S}$  : High Resolution Grid



$\mathbb{S}$  : Neighborhood Resolution



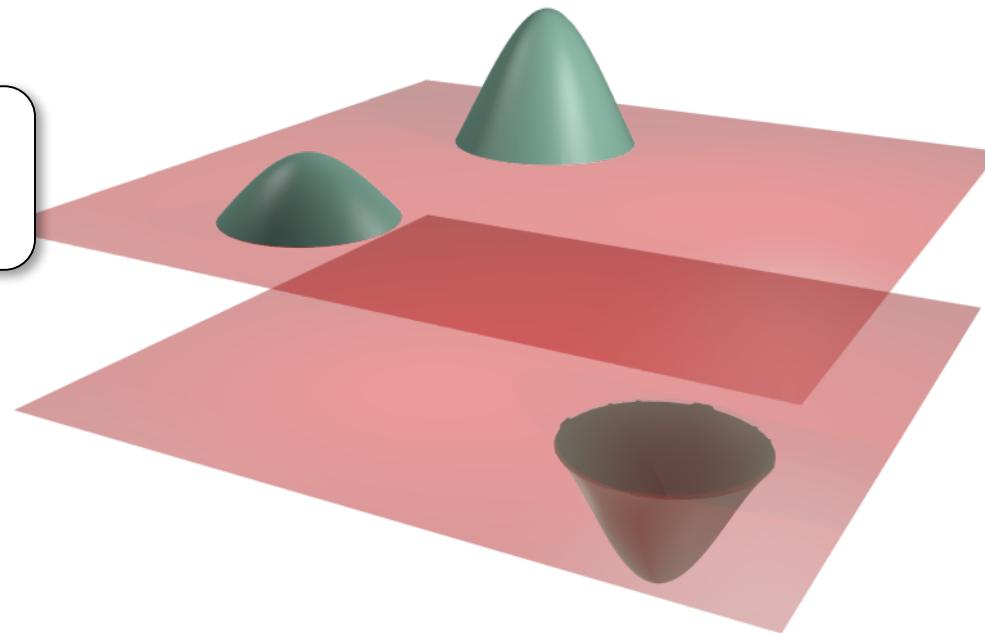
# Identify Salient Features



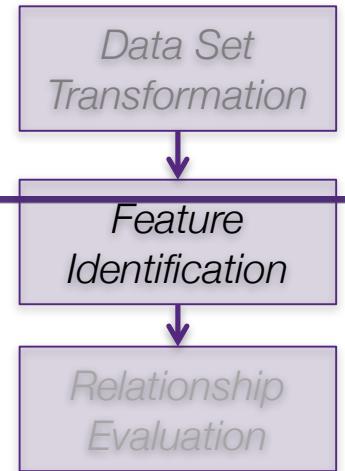
- Topological features of a *scalar function*
  - Neighborhood of critical points
  - Features represented as spatio-temporal points
  - Positive and negative features

*Advantage*

1. Naturally captures *salient* features



# Identify Salient Features



- Topological features of a scalar function
  - Neighborhood of critical points
  - Features represented as spatio-temporal points
  - Positive and negative features

8am - 9am  
May 1 2011  
  
5 Boro Bike Tour



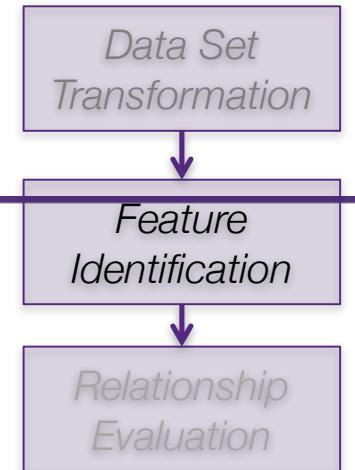
Advantage



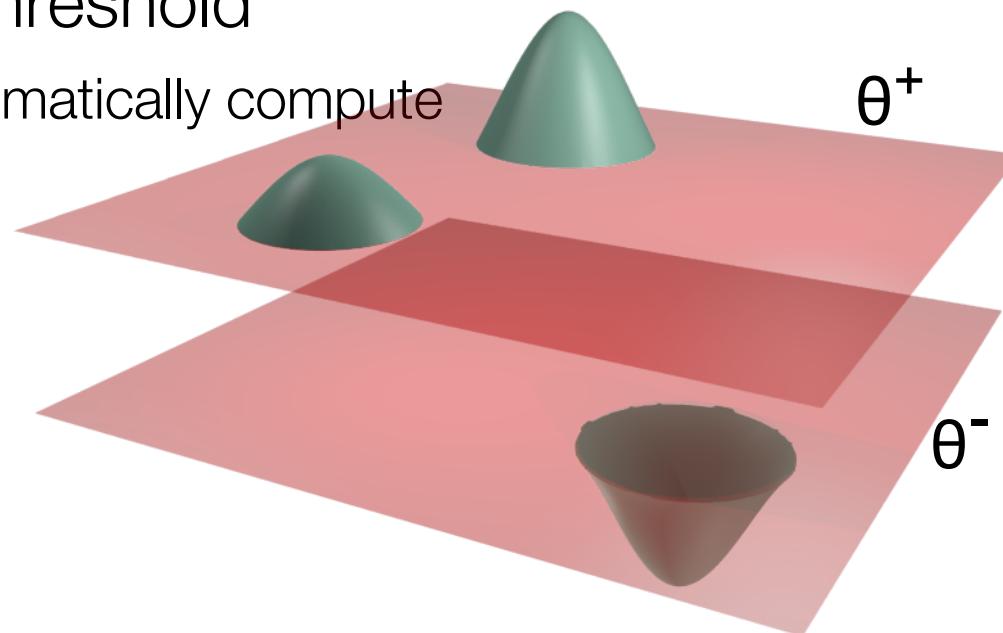
Negative Feature

2. Features can have arbitrary shapes

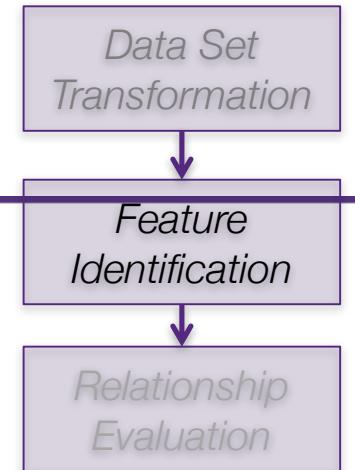
# Identify Salient Features



- Topological features of a *scalar function*
  - Neighborhood of critical points
  - Features represented as spatio-temporal points
  - Positive and negative features
- Neighborhood defined by a threshold
  - Use *topological persistence* to automatically compute thresholds in a data-driven fashion



# Identify Salient Features

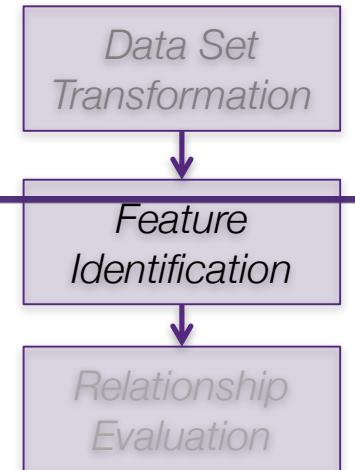


- Topological features of a *scalar function*
  - Neighborhood of critical points
  - Features represented as spatio-temporal points
  - Positive and negative features
- Neighborhood defined by a threshold
  - Use *topological persistence* to automatically compute thresholds in a data-driven fashion

*Advantage*

3. Data driven and robust

# Identify Salient Features

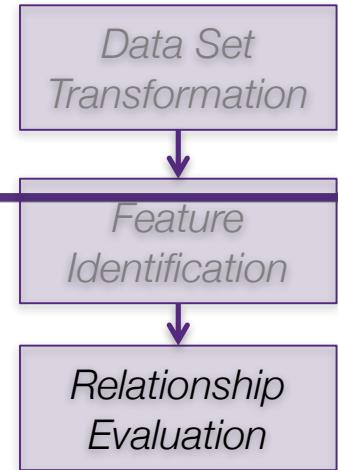


- Topological features of a *scalar function*
  - Neighborhood of critical points
  - Features represented as spatio-temporal points
  - Positive and negative features
- Neighborhood defined by a threshold
  - *Topological persistence* used to automatically compute thresholds
- *Merge Tree Index* used to identify features at all resolutions
  - $O(n \log n)$  to construct
  - Computing features is output sensitive

*Advantage*

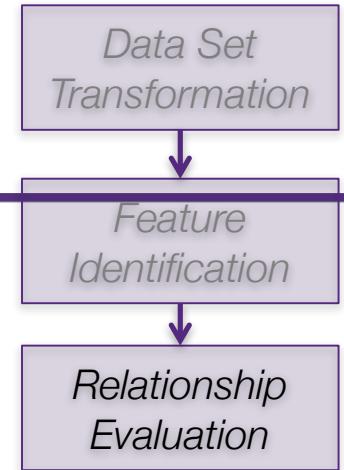
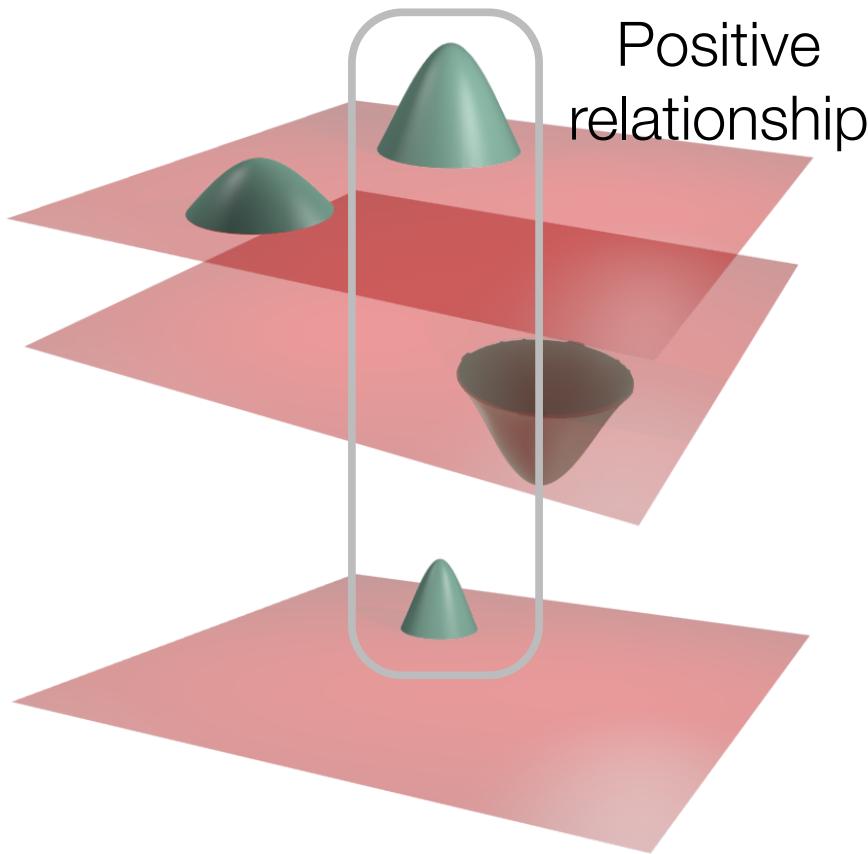
4. Very efficient

# Evaluating Relationships

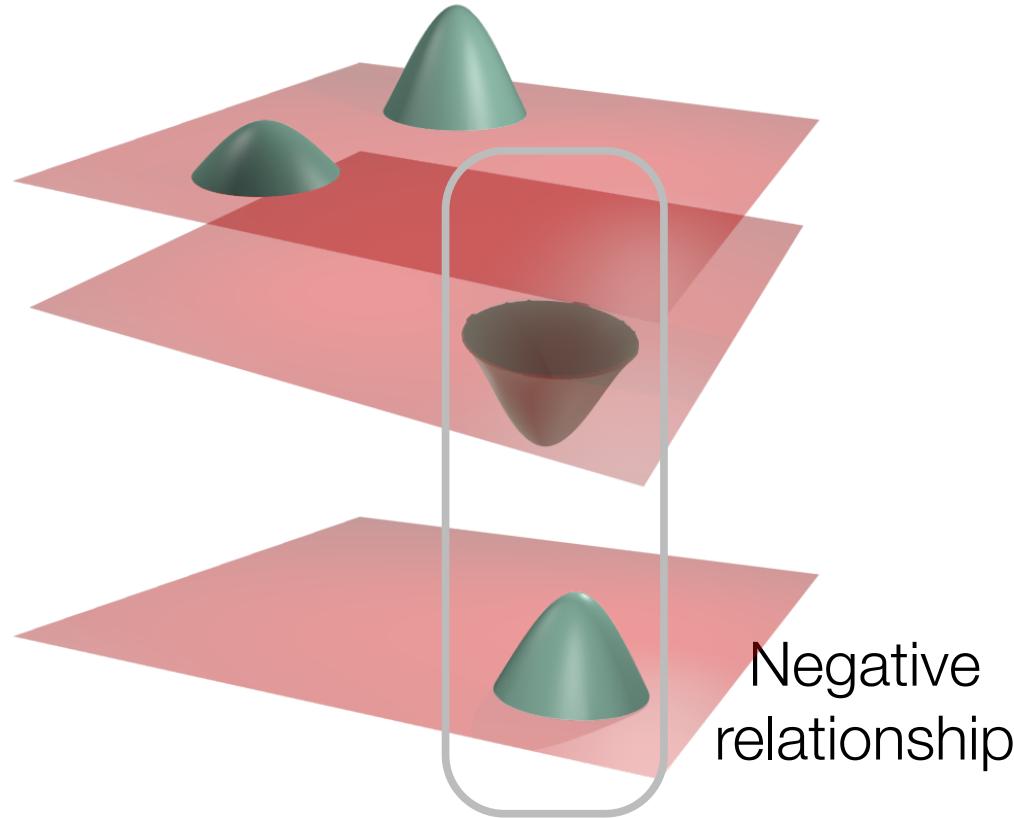
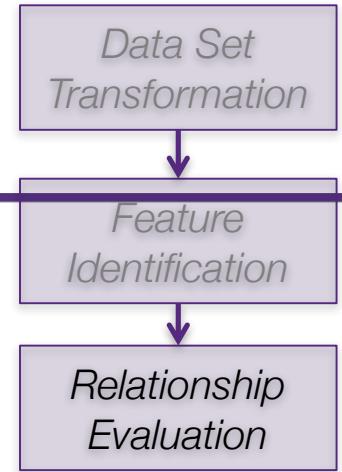


- Relationship between functions  $f$  and  $g$  consists of the set of spatio-temporal points that are features in both functions
- Let  $\Sigma_1$  and  $\Sigma_2$  be the set of features of  $f_1$  and  $f_2$ .  $f_1$  and  $f_2$  are *feature-related* at a spatio-temporal point  $x = (s,t)$  if  $x \in \Sigma_1 \wedge \Sigma_2$ 
  - E.g., for Hurricane Sandy, there is a negative feature in the taxi density function and a positive feature in the wind speed function
- *Relationship Score*: Captures the nature of the relationship – positive or negative

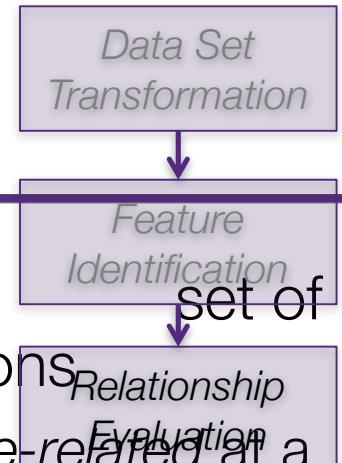
# Identifying Relationships



# Identifying Relationships



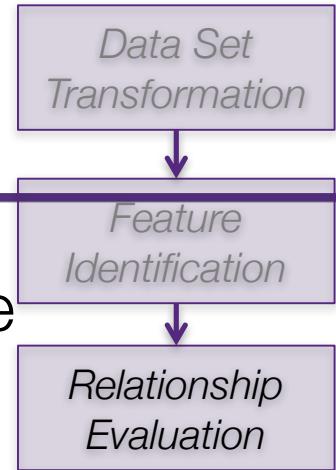
# Evaluating Relationships



- Relationship between functions  $f$  and  $g$  consists of the spatio-temporal points that are features in both functions
- Let  $\Sigma_1$  and  $\Sigma_2$  be the set of features of  $f_1$  and  $f_2$ .  $f_1$  and  $f_2$  are *feature-related* at a spatio-temporal point  $x = (s,t)$  if  $x \in \Sigma_1 \wedge \Sigma_2$ 
  - E.g., for Hurricane Sandy, there is a negative feature in the taxi density function and a positive feature in the wind speed function
- *Relationship Score*: Captures the nature of the relationship – positive or negative
- *Relationship Strength*: How often the functions are related – strong or weak
- Restricted Monte Carlo procedure to test the statistical significance accounting for the spatial and temporal proximity
  - Prune potentially coincidental relationships

[Skip to Experiments](#)

# Evaluating Relationships



- Monte Carlo procedure to test the statistical significance accounting for the spatial and temporal proximity
- Prune potentially coincidental relationships

Relationship between functions  $f$  and  $g$ , with score  $\tau^*$

$H_0$ : The two functions are not related

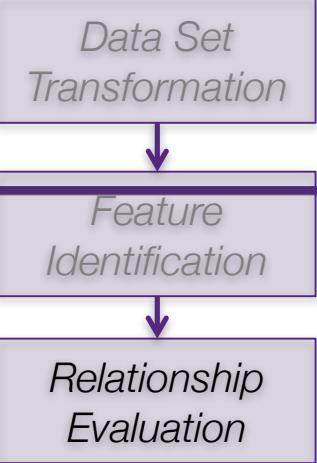
$H_1$ : The two functions are related

$$p = \frac{\sum_i^N I(\tau_i \geq \tau^*)}{N}$$

*N permutations*

Reject the null hypothesis if  
 $p \leq \alpha$

# Evaluating Relationships

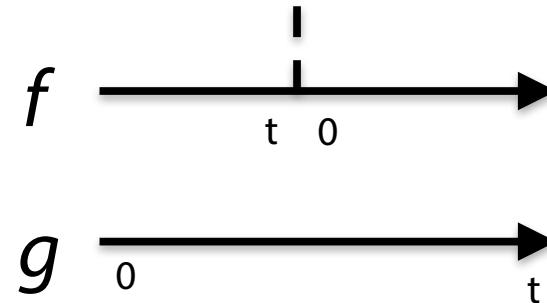


- Monte Carlo procedure to test the statistical significance accounting for the spatial and temporal proximity
- Prune potentially coincidental relationships

## Statistical Significance

Permutations: need to respect spatio-temporal correlations of the data!

Temporal case → Temporal shifts



Spatial case → Spatial shifts

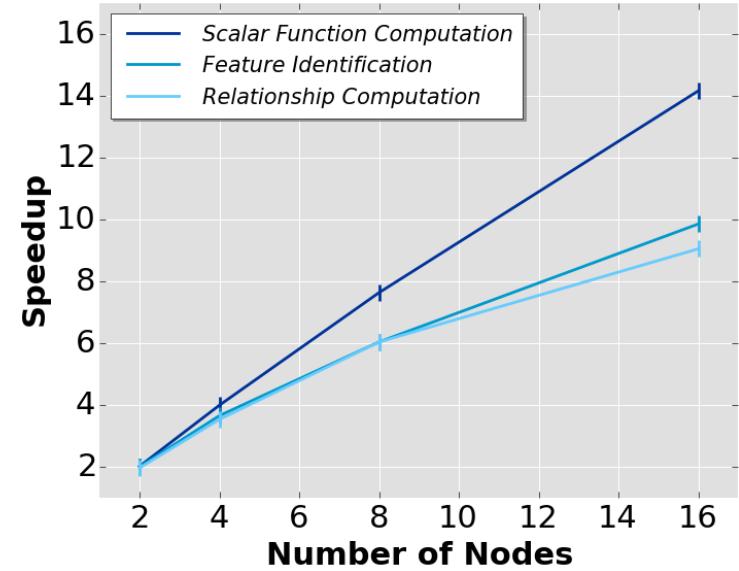
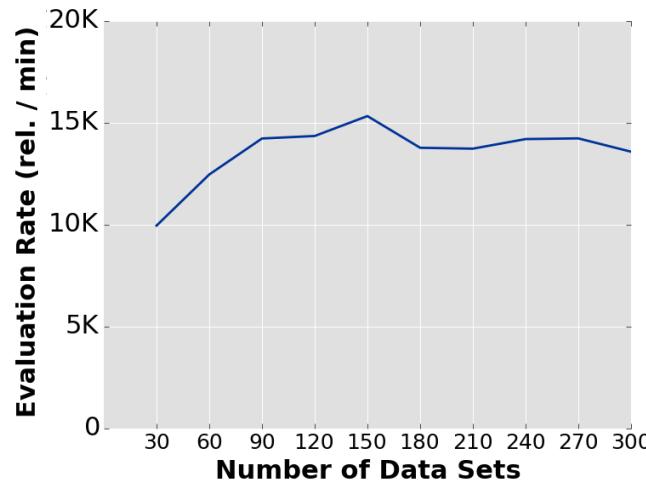
# Experimental Evaluation

---

- Implemented using Map Reduce
  - Feature identification and relationship evaluation are independent operations
  - Two collections of data sets used for experiments
    - NYC Urban: 9 data sets from NYC agencies
    - NYC Open Data: 300 spatio-temporal data sets
  - Setup
    - 20 compute nodes, AMD Opteron(TM) Processor 6272 (4x16 cores) running at 2.1GHz, 256GB of RAM – *for most experiment*
    - Amazon EMR: m1.medium (for master) and r3.2xlarge (for slaves) – *for scalability tests*

# Quantitative Evaluation

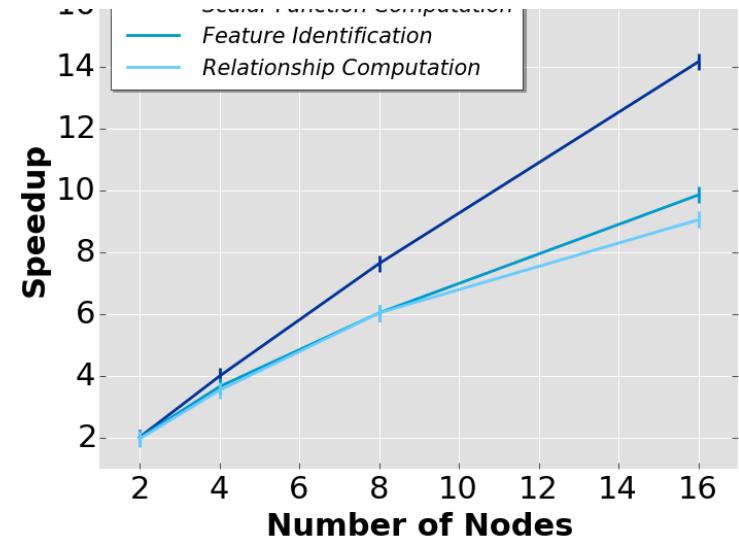
- Approach is efficient: 200 min to compute scalar functions and features for NYC Open Data; and 60 min for NYC Urban
- Query rate: evaluate  $10^4$  relationships per minute
- Scales linearly with number of nodes
- Assessed correctness and robustness



Details in [Chirigati et al., ACM SIGMOD 2016]

# Quantitative Evaluation

- Approach is efficient: 200 min to compute scalar functions and features for NYC Open Data; and 60 min for NYC Urban
- Scales linearly with number of nodes
- Code, data, and reproducible experiments available at:
  - <https://github.com/ViDA-NYU/data-polygamy>



Details in [Chirigati et al., ACM SIGMOD 2016]

# Qualitative Evaluation

---

Does the approach uncover *interesting, non-trivial* relationships?

Details in [Chirigati et al., ACM SIGMOD 2016]

# (Some) Interesting Relationships

1. Would a reduction in traffic speed reduce the number of accidents?

Find all relationships between Collisions and Traffic Speed  
data sets



*Positive relationship between number of collisions and speed*



*Positive relationship between number of persons killed and speed*

The image shows a news article from *NYMag.com* titled "Things to Know About NYC's New 25-Miles-Per-Hour Speed Limit" by Caroline Bankoff. The article URL is <http://nymag.com/daily/intelligencer/2014/11/things-to-know-about-nycs-new-speed-limit.html>. The image includes a speed limit sign reading "SPEED LIMIT 25" and a quote from the article: "Last week, Mayor de Blasio signed a law lowering New York City's 30-miles-per-hour speed limit to 25. The change is the centerpiece of de Blasio's [Vision Zero](#) plan to drastically reduce New York City traffic deaths,".

# (Some) Interesting Relationships

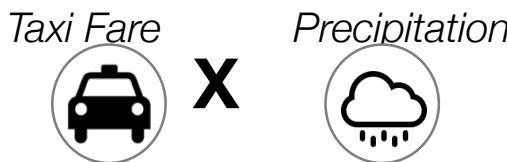
2. Why it is so hard to find a taxi when it is raining?



*Find all relationships between Taxi and Weather data sets*

*Negative relationship* between number of taxis and average precipitation

**Hypothesis:** Taxi drivers are target earners

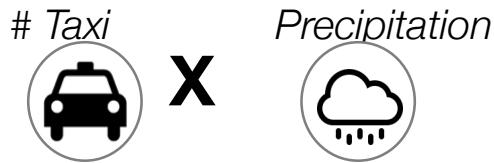


→ *Suggests that hypothesis is true*

*Strong positive relationship* between precipitation and average fare

# (Some) Interesting Relationships

## 2. Why it is so hard to find a taxi when it is raining?



*Find all relationships between Taxi  
and Weather data sets*

This hypothesis had been refuted by [Farber 2014]

- Farber did not find a correlation (using OLS regression) between drivers' earnings and rainfall.
- But (i) he did not take into account the amount of rainfall—instead, he used a binary value indicating whether it rained or not; and (ii) he considered the entire time period—periods with very sparse rainfall are considered equivalent to those having higher rainfall.

It is important to consider salient features

# (Some) Interesting Relationships

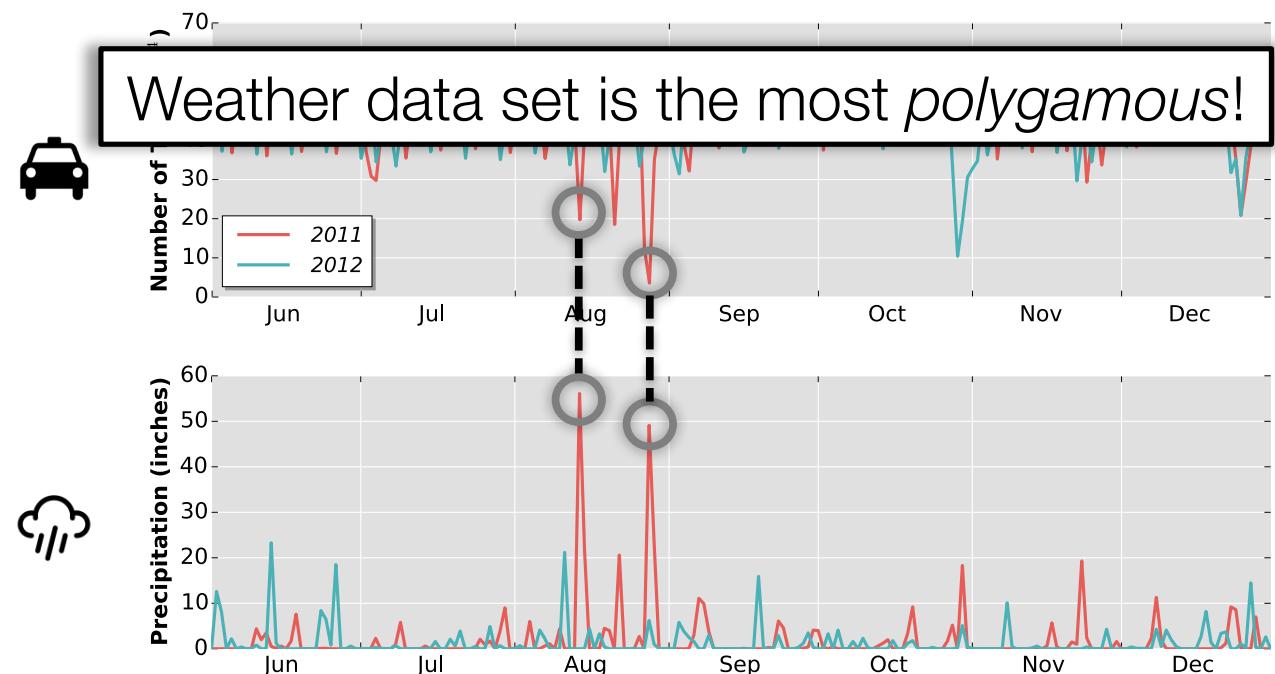
## 3. Why is the number of taxi trips too low?

# Taxi X Precipitation

*Negative relationship* between number of taxis and average precipitation

# Taxi X Wind speed

*Negative relationship* between number of taxis and wind speed



NYU

TANDON SCHOOL  
OF ENGINEERING

# (Some) Interesting Relationships

---

- Citi Bike and Weather

# Citi Bike

stations



Snow



*Negative relationship* between snow precipitation  
and active Citi Bike stations

(day, city) ✓

(hour, city) Ø

# Data Polygamy: Ongoing Work

---

- It's hard to evaluate!
  - No ground truth available
  - Need benchmark
  - Need real use cases from domain experts
- $\sim 100$  *significant* relationships per resolution
  - More relationships (and their implications) can be understood by having domain experts

How to explore and analyze the relationships?

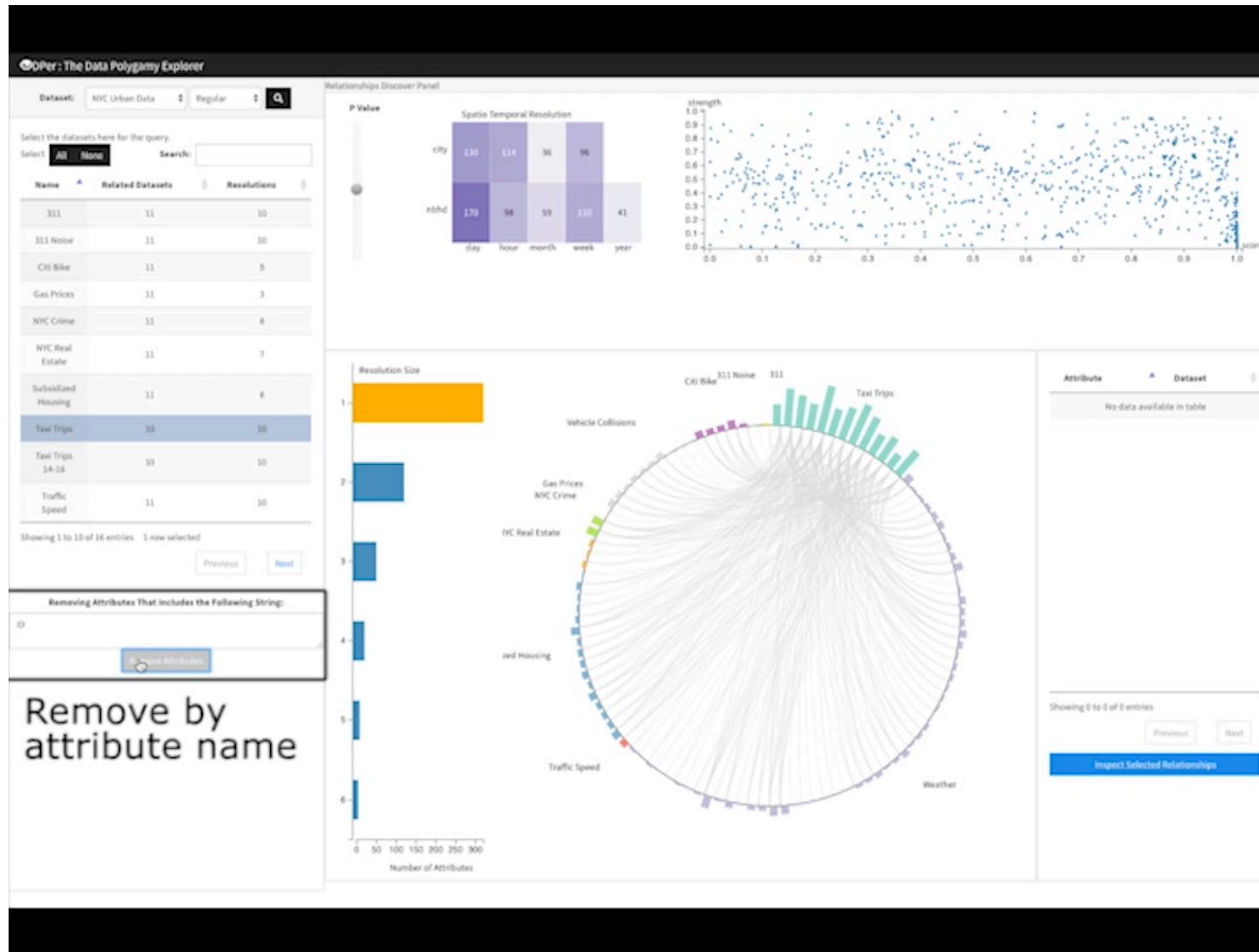
# Visually Exploring Relationships

---

DPer:A Deeper Dive into  
Polygamous Relationships in Urban Data

<https://vgc.poly.edu/~juliana/videos/dper2.mov>

# Visually Exploring Relationships



<https://vgc.poly.edu/~juliana/videos/dper2.mov>

[Chirigati et al., ACM SIGMOD 2017]

# Takeaway: Big Data Exploration

---

- **Usability** is of paramount importance
  - Need to empower domain experts to explore their data
- Exploration requires **interactivity** – improve the rate at which *users make observations, draw generalizations and generate hypotheses*
- Not always Cloud is the answer – **smart algorithms** can go a long way to attain scalability
- **Visualization** is essential
- Need better methods to guide users in exploration
  - Find patterns, outliers, and explain them!
- Warning: a risk with big data is that you will “discover” patterns that are meaningless – watch out for **bogus patterns/events**

# Transparency and Reproducibility

# Science and Reproducibility

- Reproducibility is the cornerstone of science
- *If I have seen further it is by standing on the shoulders of giants.*

Isaac Newton

- If we can't trust previous results, we have to start over from scratch
  - Science is incremental and self-correcting
  - To increase impact, visibility [Vandewalle et al. 2009] and research quality [Begley and Ellis 2012]
  - *Without reproducibility, people die!*

John Wilbanks, AMPS Workshop on Reproducibility 2011



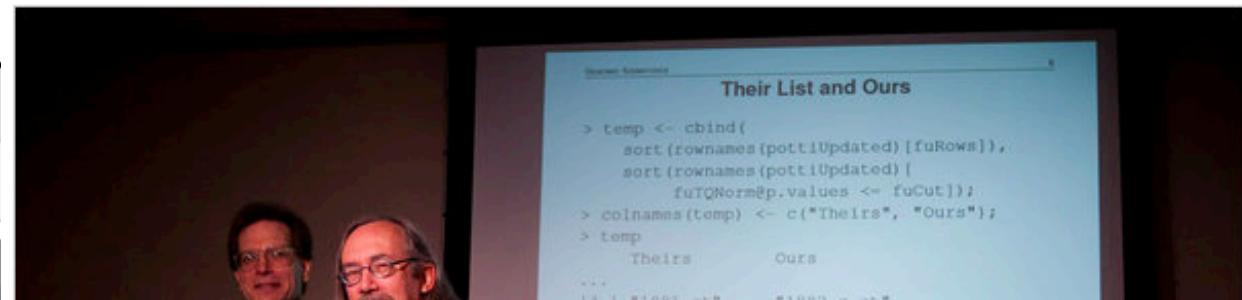
# Science and Reproducibility

The New York Times

WORLD U.S. N.Y. / REGION BUSINESS



## How Bright Promise in Cancer Testing Fell Apart



But the research at Duke turned out to be wrong. Its gene-based tests proved worthless, and the research behind them was discredited. Ms. Jacobs died a few months after treatment, and her husband and other patients' relatives have retained lawyers.

## Nobel Laureate Retracts Prize

By KENNETH CHANG

Published: September 23, 2010



Michael Stravato for The New York Times

Keith Baggerly, left, and Kevin Coombes, statisticians at M. D. Anderson Cancer Center, found flaws in research on tumors.

By GINA KOLATA

Published: July 7, 2011

Linda B. Buck, who shared the 2004 Nobel Prize in Medicine for deciphering the working of the olfactory system, has retracted two scientific papers after she was unable to repeat the findings.

When Juliet Jacobs found out she had lung cancer, she was terrified, but realized that her hope lay in getting the best treatment medicine could offer. So she got a second opinion, then a third. In February of 2010, she ended up at Duke University, where she entered a research study whose promise seemed stunning.

RECOMMEND

TWITTER

COMMENTS (76)

E-MAIL

ON

# Science and Reproducibility

2	3	4	Real GDP growth						11	
			Debt/GDP			30 or less	30 to 60	60 to 90	90 or above	
26						3.7	3.0	3.5	1.7	5.5
27	Minimum					1.6	0.3	1.3	-1.8	0.8
28	Maximum					5.4	4.9	10.2	3.6	13.3
29										
30	US	1946-2009		n.a.	3.4	3.3	-2.0	n.a.		
31	UK	1946-2009		n.a.	2.4	2.5	2.4	n.a.		
32	Sweden	1946-2009		3.6	2.9	2.7	n.a.	6.3		
33	Spain	1946-2009		1.5	3.4	4.2	n.a.	9.9		
34	Portugal	1952-2009		4.8	2.5	0.3	n.a.	7.9		
35	New Zealand	1948-2009		2.5	2.9	3.9	-7.9	2.6		
36	Netherlands	1956-2009		4.1	2.7	1.1	n.a.	6.4		
37	Norway	1947-2009		3.4	5.1	n.a.	n.a.	5.4		
38	Japan	1946-2009		7.0	4.0	1.0	0.7	7.0		
39	Italy	1951-2009		5.4	2.1	1.8	1.0	5.6		
40	Ireland	1948-2009		4.4	4.5	4.0	2.4	2.9		
41	Greece	1970-2009		4.0	0.3	2.7	2.9	13.3		
42	Germany	1946-2009		3.9	0.9	n.a.	n.a.	3.2		
43	France	1949-2009		4.9	2.7	3.0	n.a.	5.2		
44	Finland	1946-2009		3.8	2.4	5.5	n.a.	7.0		
45	Denmark	1950-2009		3.5	1.7	2.4	n.a.	5.6		
46	Canada	1951-2009		1.9	3.6	4.1	n.a.	2.2		
47	Belgium	1947-2009		n.a.	4.2	3.1	2.6	n.a.		
48	Austria	1948-2009		5.2	3.3	-3.8	n.a.	5.7		
49	Australia	1951-2009		3.2	4.9	4.0	n.a.	5.9		
50										
51				4.1	2.8	2.8	=AVERAGE(L30:L44)			

In this paper, we search for historical data on public debt and GDP growth and find no systematic relationship between public debt and GDP growth; average percent loss of GDP is about 1.5 percent lower than the average percent growth in the period between 1946 and 2009. Similar across economies, we find no significant relationship between public debt levels and GDP growth rates as a group. There are some exceptions, however. In contrast, in emerging markets, public debt levels are significantly higher than in developed countries.

Our top concern is whether public debt has been soaring in the wake of the recent global financial maelstrom, especially in the epicenter countries. This should not be surprising, given the experience of earlier severe recessions.

Recovering from a financial crisis is a major challenge for governments and by private entities. For emerging markets, we find that there exists a significantly more severe threshold for total gross external debt (public and private)—which is almost exclusively held by the private sector—than for developed countries. This suggests that the private sector in emerging markets is more exposed to external shocks than the public sector. The same is true for developed countries, although the difference is smaller. This finding is consistent with the observation that private debt in emerging markets is significantly higher than in developed countries. It also suggests that the private sector in emerging markets is more exposed to external shocks than the public sector. The same is true for developed countries, although the difference is smaller. This finding is consistent with the observation that private debt in emerging markets is significantly higher than in developed countries.

# Reproducibility and Data Science

---

- Reproducibility is important not just for science
- Many decisions are made based on results of computational analyses
- Need transparency!

Subsidized housing

Computational hedge funds

Policing and crime prevention

Crime sentencing

Facebook ranking

Loan approval

Taxi fare increases

...

# How?

---

**Provenance:** a key ingredient for reproducibility

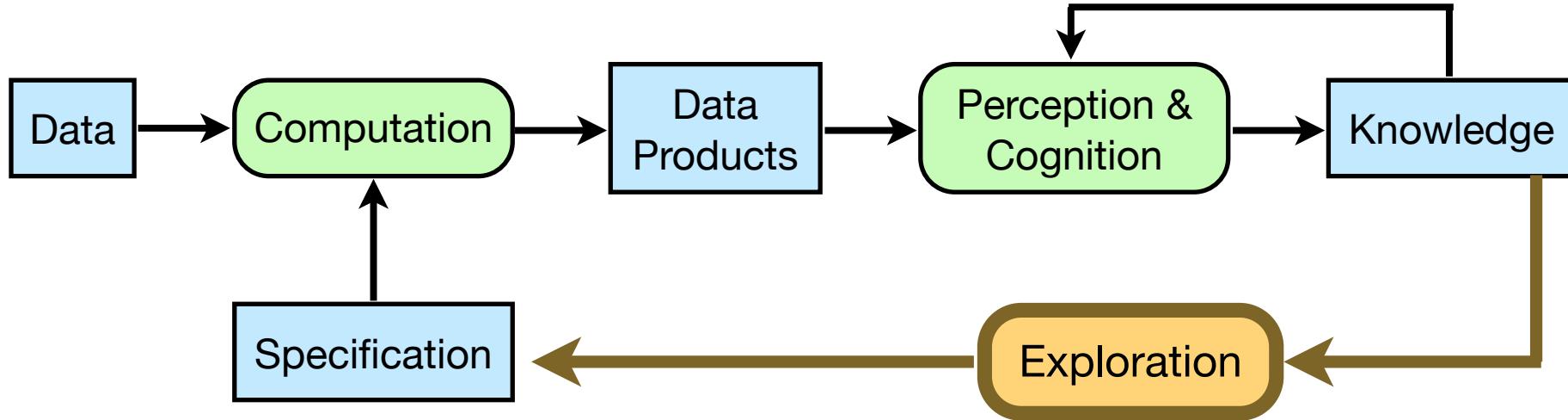
*“The source or origin of an object; its history and pedigree; a record of the ultimate derivation and passage of an item through its various owners.”*

*The Oxford English Dictionary*

Helps determine the value, accuracy and authorship of an object

Used in many fields: works of art and antiques, archives and books, **science**, ...

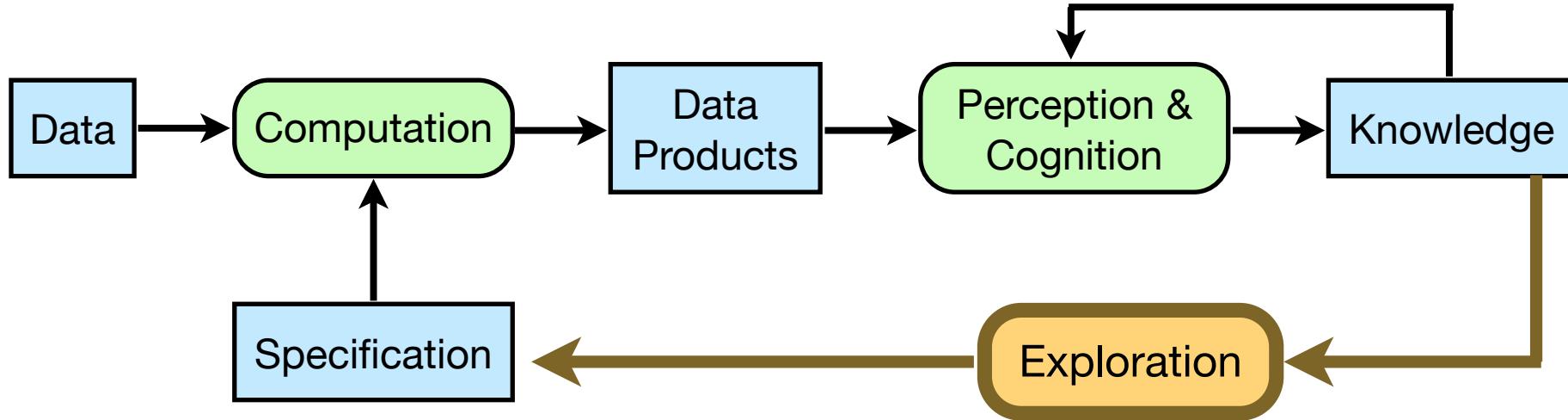
# Data Lifecycle: The Exploration Pipeline



[Modified from Van Wijk, Vis 2005]

- ◆ Iterative process to generate and test hypotheses
- ◆ Easy to get lost---derive a result and not remember how you got there
- ◆ Lots of data: both input and derived
- ◆ Complex processes that encompass multiple tools

# Data Lifecycle: The Exploration Pipeline



[Modified from Van Wijk, Vis 2005]

- ◆ If Need systematic mechanisms to capture provenance
- ◆ Else why not? W
- ◆ Lots of data: both input and derived
- ◆ Complex processes that encompass multiple tools

# Trust, Sharing and Collaboration

---

- Result transparency
  - Show me your work!
  - Allow results to be verified → trust the results

Keep track of what you do and the steps you follow – the provenance of your work
- Hard *data science* problems require people with different expertise to collaborate
  - Need to share *work*, but this can be challenging...
  - *E.g.*, **A** sends their analysis script to **B**, but **B** cannot run it...
    - Missing or incorrect versions of libraries
    - Hard-coded file names: /home/A/myinfile.txt
    - ...

Follow best practices for sharing and reproducibility

# Reproducibility: Best Practices

- Never use spreadsheets!

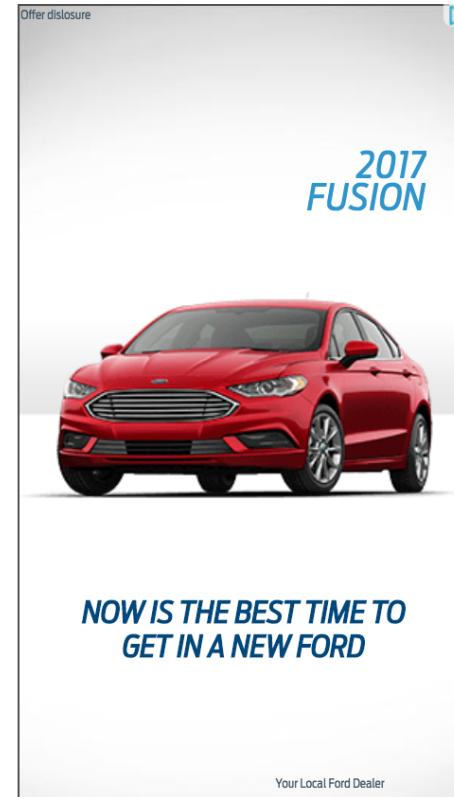
OCT 30, 2017 @ 08:07 PM 958 

The Little Black Book of Billionaire Secrets

## You Can Reduce Business Risk By Phasing Out Spreadsheets For Business



**Meta S. Brown**, CONTRIBUTOR  
I decode analytics for business people. [FULL BIO](#)   
Opinions expressed by Forbes Contributors are their own.



<https://www.forbes.com/sites/metabrown/2017/10/30/you-can-reduce-business-risk-by-phasing-out-spreadsheets-for-business/#d80a9a047a04>

# Reproducibility: Best Practices

---

- Never use spreadsheets!
- Do scripting instead
  - Lots of tools, e.g., Python (and the rich Python environment)
  - Easy to reproduce a script
- Adopt literate programming, e.g., Jupyter Notebook
  - “a prose first approach where exposition with human-friendly text is punctuated with code blocks”
- Keep track of the code/techniques/parameters you tried and the associated results
  - Version control systems are your friends, e.g., git, github
- Watch out for portability -- need provenance for the computing environment
  - ReproZip provides a solution for this: <https://www.reprozip.org/>

# Conclusions

---

- Data has been democratized and the Big Data stack enables us to explore, better understand, and leverage the growing volumes of data
- But there are many challenges
- We covered some of these challenges and a small subset of the solutions
- There are also many open (research) problems in
  - Cleaning and integration
  - Data analysis and exploration
  - Modeling and prediction
- Often new solutions need to be designed – important to learn the foundational aspects of Data Science

# Acknowledgments

- Collaborators: Huy Vo, Harish Doraiswamy, Fernando Chirigati, Theo Damoulas, Nivan Ferreira, Masayo Ota, Jorge Poco, Yeuk Yin Chan
- NYC Taxi & Limousine Commission for providing the data used in this paper and feedback on our results.
- Funding: Google, National Science Foundation, Moore-Sloan Data Science Environment at NYU, and DARPA.



GORDON AND BETTY  
**MOORE**  
FOUNDATION



ALFRED P. SLOAN  
FOUNDATION



NYU

TANDON SCHOOL  
OF ENGINEERING



VISUALIZATION  
IMAGING AND  
DATA ANALYSIS  
CENTER

고맙습니다

Merci

# Thank you

# Obrigada

# благодаря

Kiitos

# धन्यवाद

Tack

Danke

# *Eucharistw*

# Bedankt



NYU

**TANDON SCHOOL  
OF ENGINEERING**


**VIDA** VISUALIZATION  
 IMAGING AND  
 DATA ANALYSIS  
 CENTER