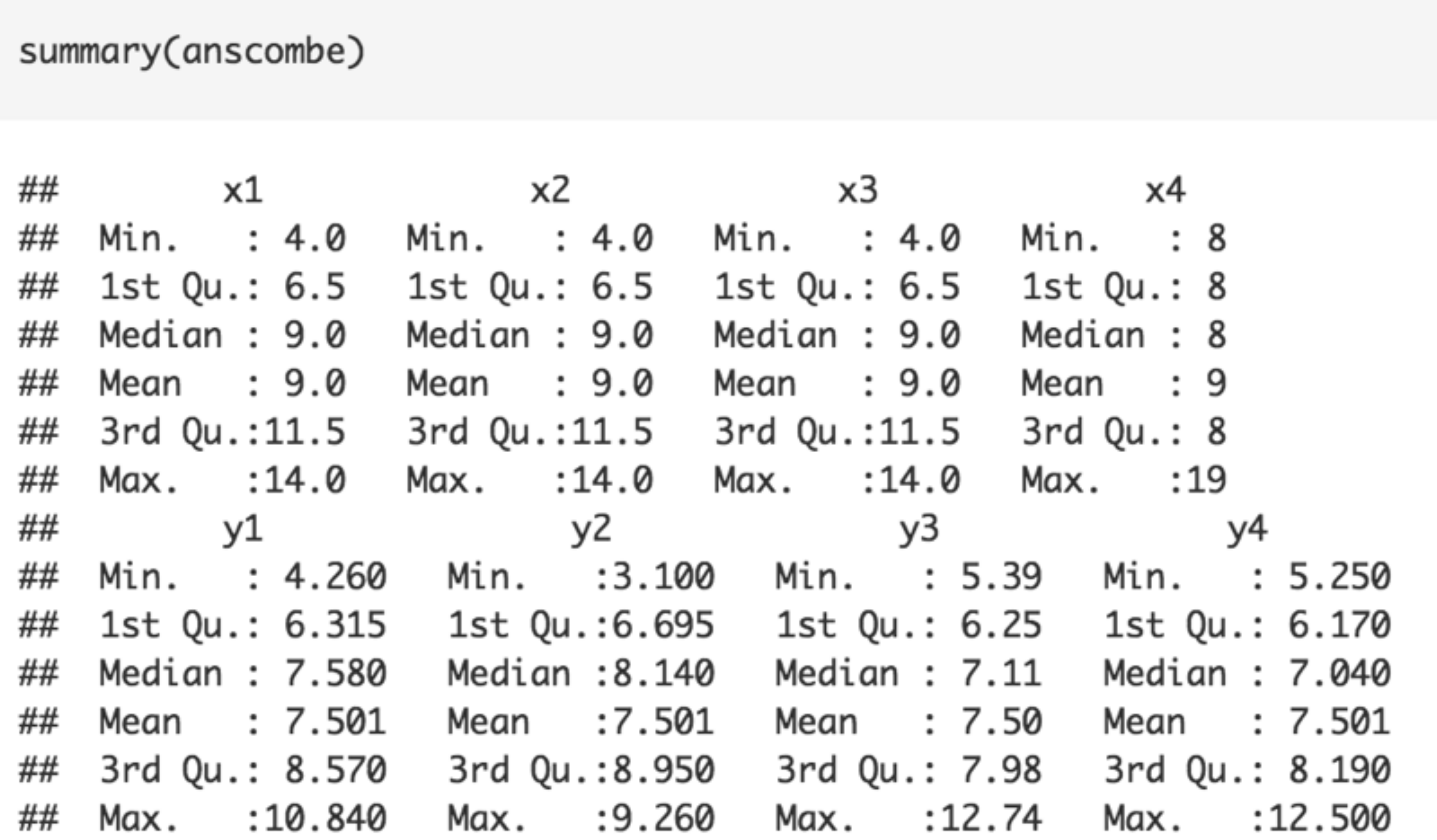
How do you look at a billion data points?

Carlos Scheidegger, Assistant Professor
Dept. of Computer Science - University of Arizona

NOAO Meeting, tools for Big Data

As computational methods get better, so must our **understanding**

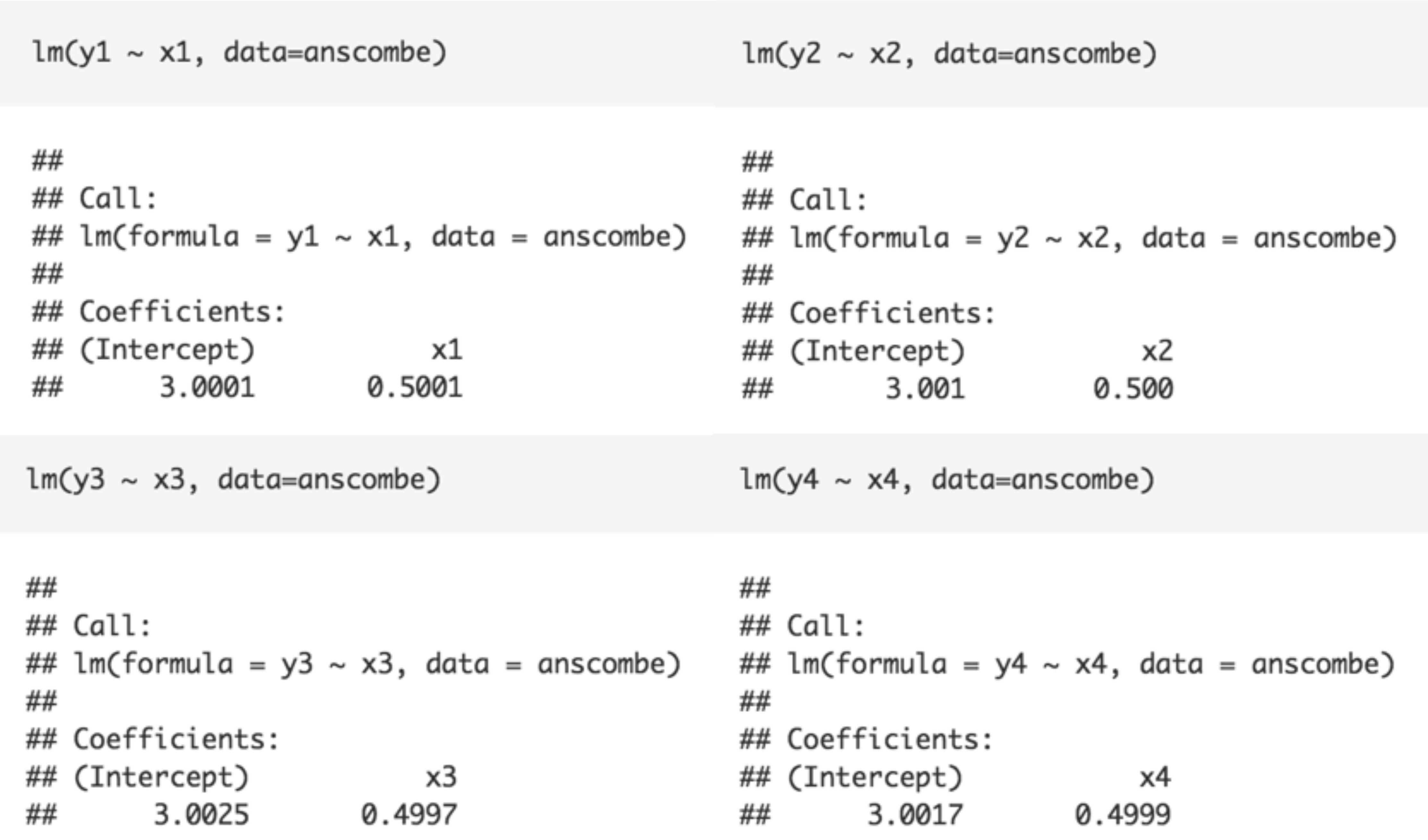
Why look at data at all?



```
summary(anscombe)
```

```
##        x1             x2             x3             x4    
##  Min.   : 4.0   Min.   : 4.0   Min.   : 4.0   Min.   : 8  
##  1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 6.5   1st Qu.: 8  
##  Median : 9.0   Median : 9.0   Median : 9.0   Median : 8  
##  Mean   : 9.0   Mean   : 9.0   Mean   : 9.0   Mean   : 9  
##  3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.:11.5   3rd Qu.: 8  
##  Max.   :14.0   Max.   :14.0   Max.   :14.0   Max.   :19  
##        y1               y2               y3              y4        
##  Min.   : 4.260   Min.   :3.100   Min.   : 5.39   Min.   : 5.250  
##  1st Qu.: 6.315   1st Qu.:6.695   1st Qu.: 6.25   1st Qu.: 6.170  
##  Median : 7.580   Median :8.140   Median : 7.11   Median : 7.040  
##  Mean   : 7.501   Mean   :7.501   Mean   : 7.50   Mean   : 7.501  
##  3rd Qu.: 8.570   3rd Qu.:8.950   3rd Qu.: 7.98   3rd Qu.: 8.190  
##  Max.   :10.840   Max.   :9.260   Max.   :12.74   Max.   :12.500  
```

Why look at data at all?



```
lm(y1 ~ x1, data=anscombe)
```

```
##
## Call:
## lm(formula = y1 ~ x1, data = anscombe)
##
## Coefficients:
## (Intercept)           x1
##      3.0001       0.5001
```

```
lm(y2 ~ x2, data=anscombe)
```

```
##
## Call:
## lm(formula = y2 ~ x2, data = anscombe)
##
## Coefficients:
## (Intercept)           x2
##       3.001        0.500
```

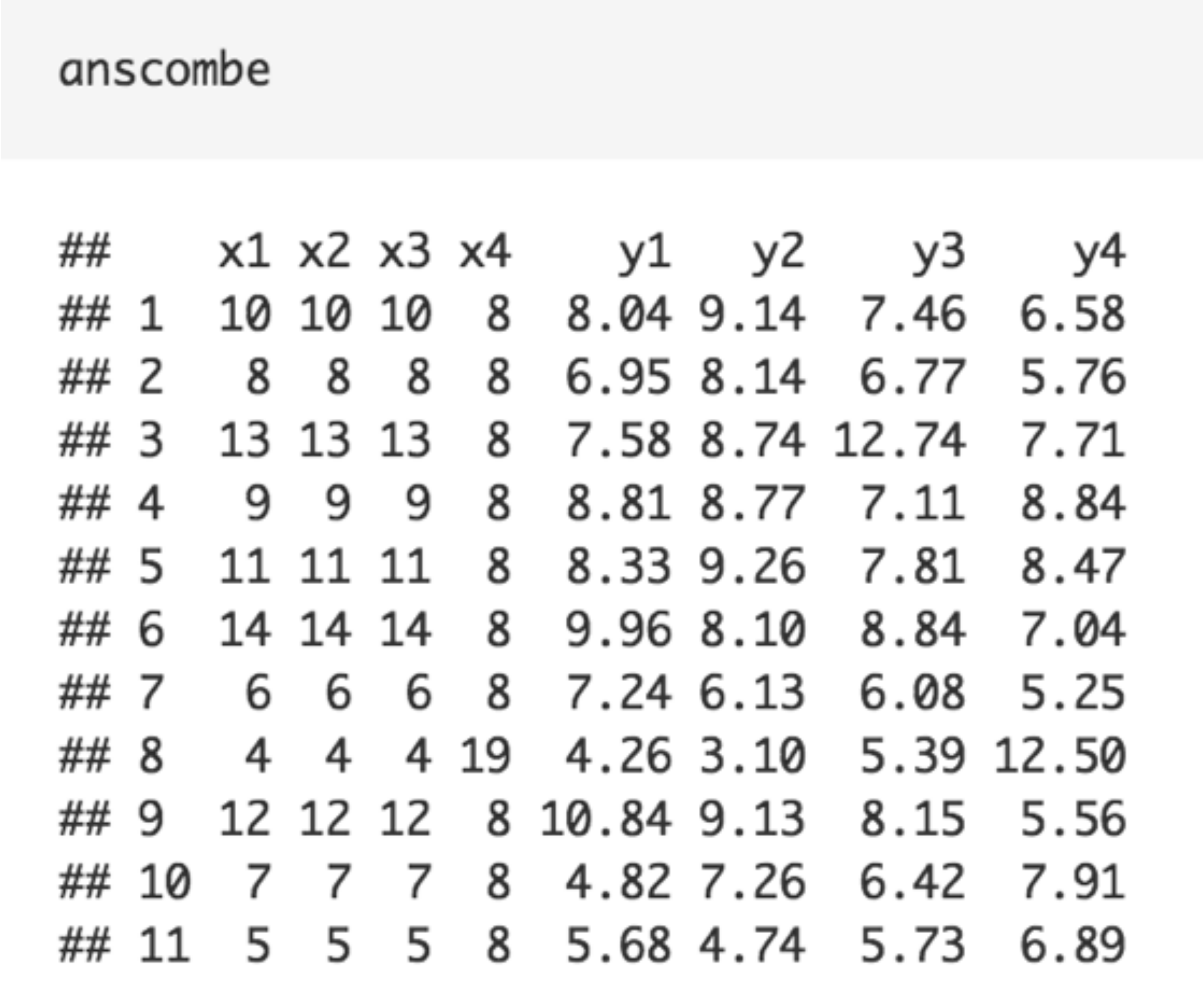
```
lm(y3 ~ x3, data=anscombe)
```

```
##
## Call:
## lm(formula = y3 ~ x3, data = anscombe)
##
## Coefficients:
## (Intercept)           x3
##      3.0025       0.4997
```

```
lm(y4 ~ x4, data=anscombe)
```

```
##
## Call:
## lm(formula = y4 ~ x4, data = anscombe)
##
## Coefficients:
## (Intercept)           x4
##      3.0017       0.4999
```

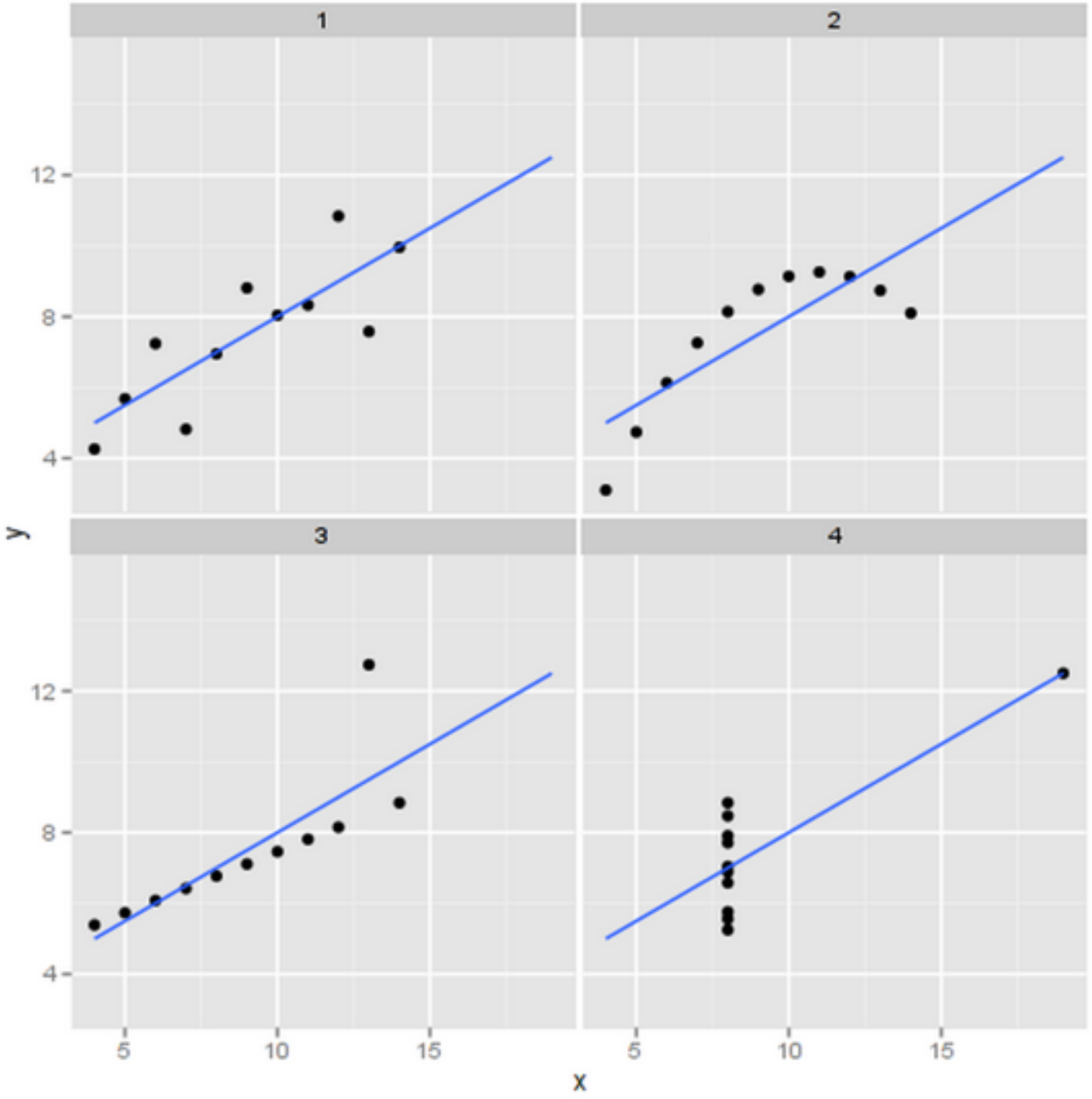
Why look at data at all?



```
anscombe
```

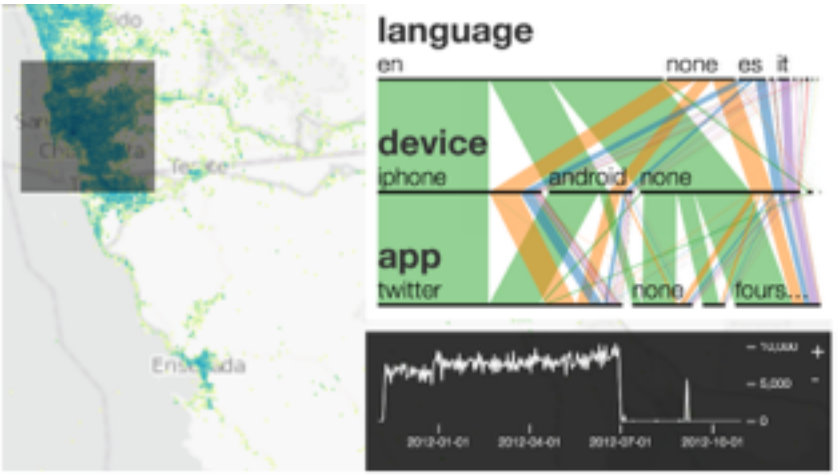
```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

Why look at data at all?

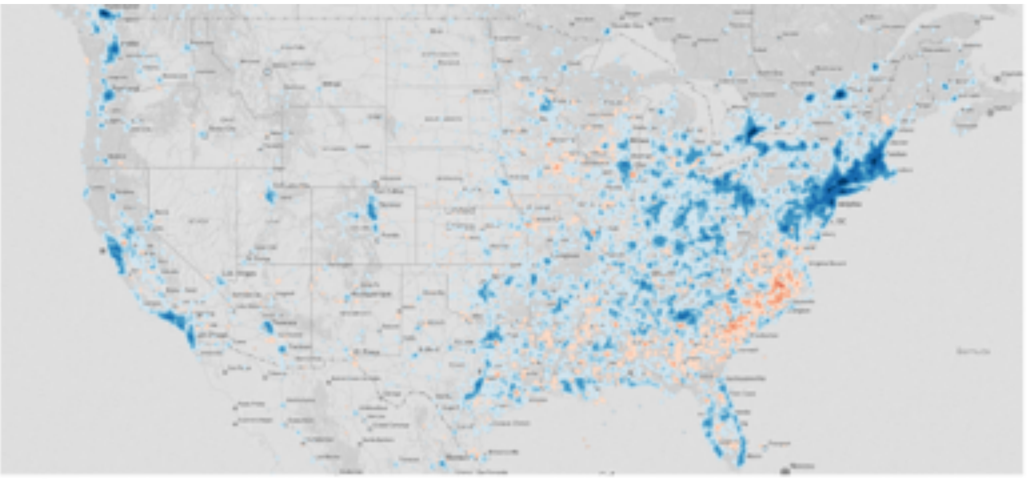


NOAO Meeting, tools for Big Data, March 16 2015

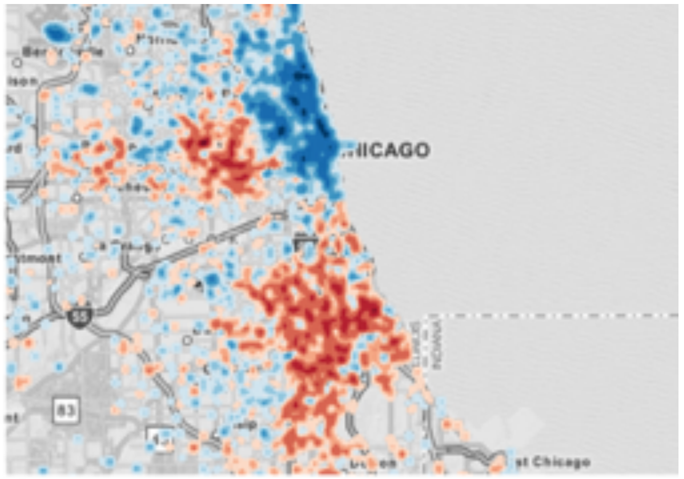
If it’s bad with 11 points, imagine 1 billion



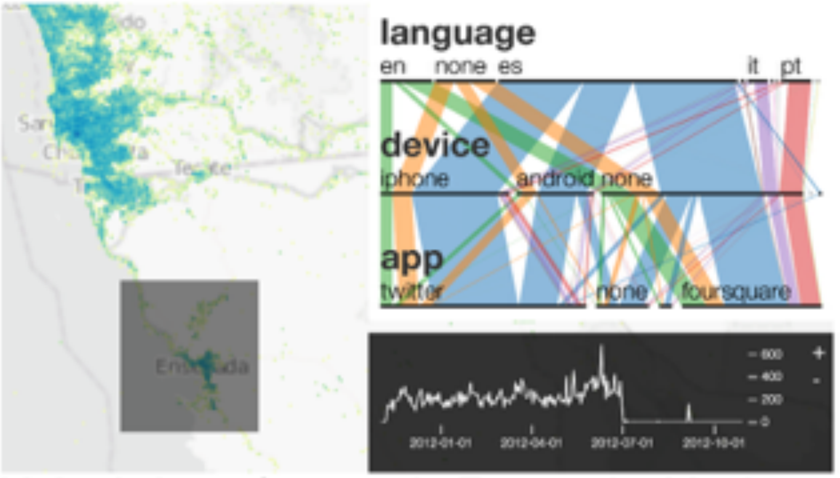
Linked view of tweets in San Diego, US



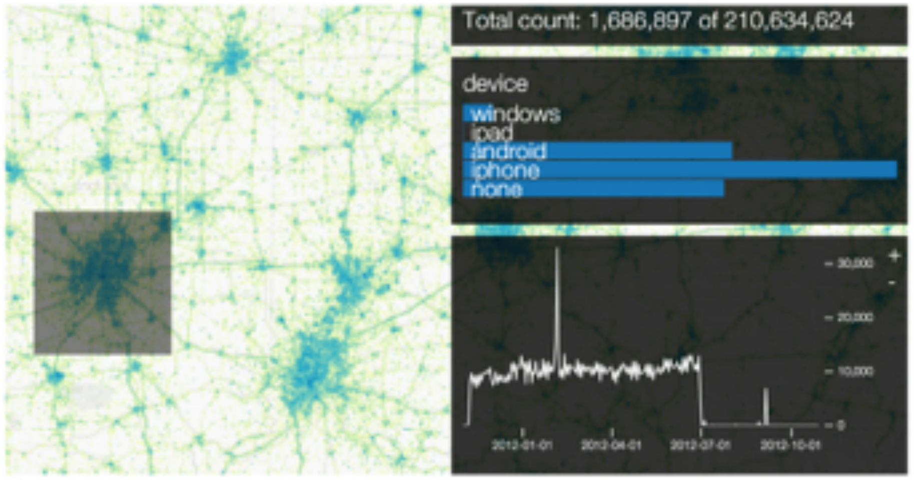
US-wide choropleth map of relative device popularity



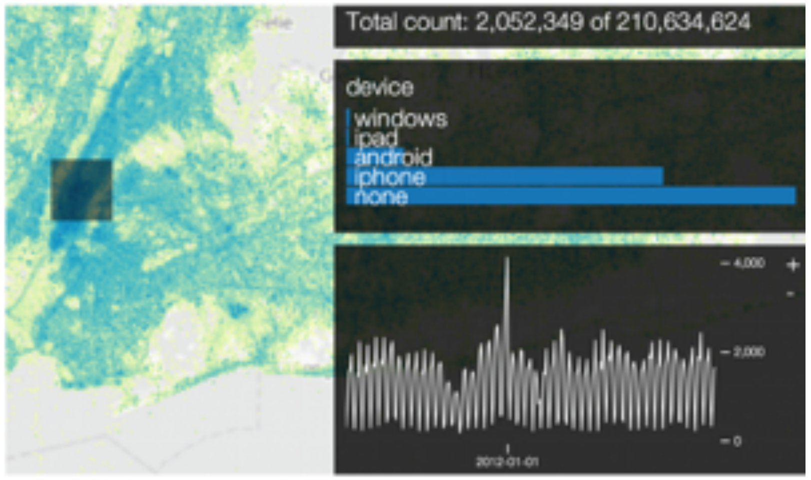
Close-up view of Chicago



Linked view of tweets in Ensenada, Mexico



Superbowl, Indianapolis



New Year's Eve, Midtown Manhattan

Nanocubes

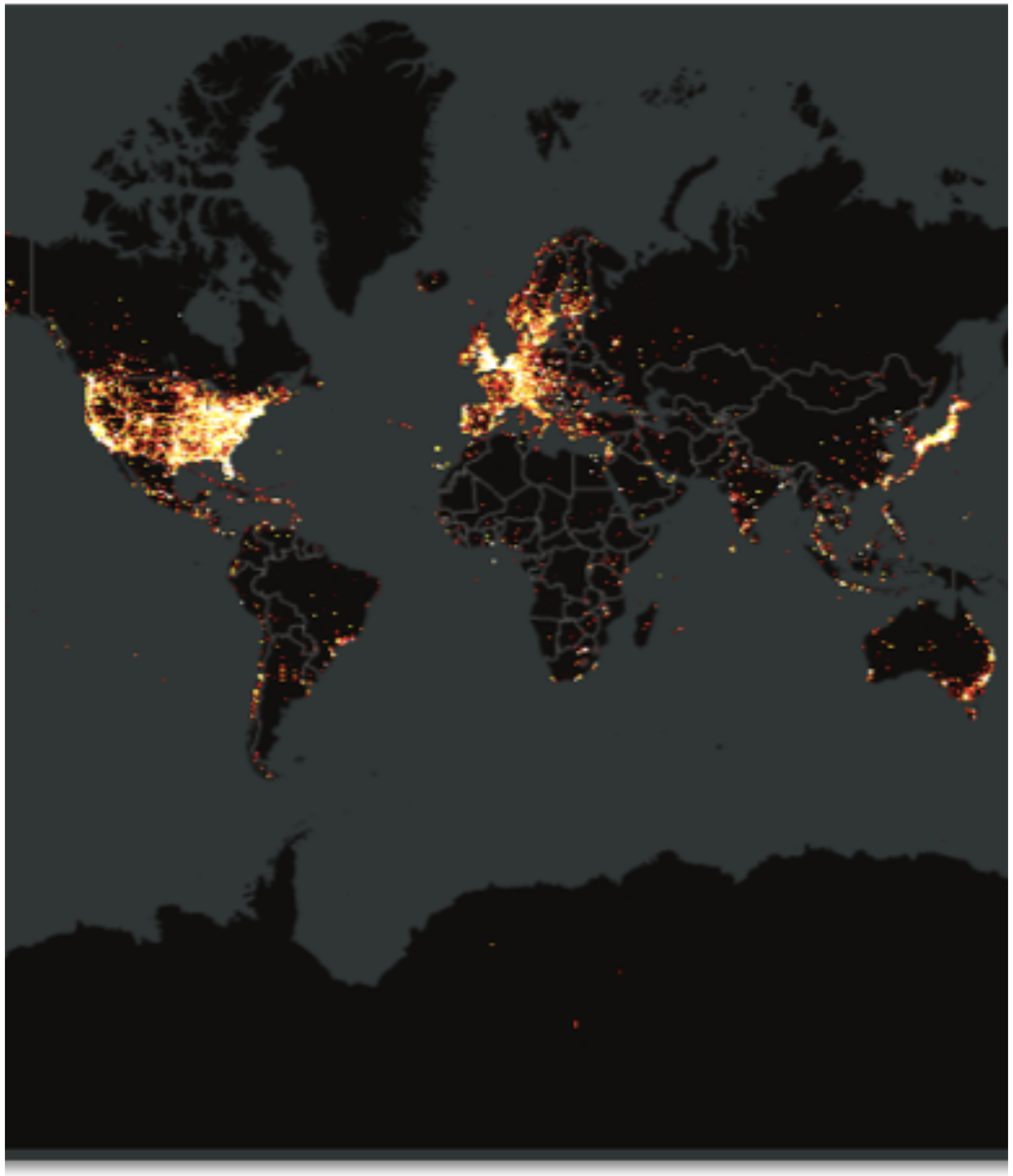
Lins, Scheidegger, Klosowski, IEEE TVCG 2013

Let’s explore the space of solutions

- Prerequisites
  - Support many different queries
  - with small memory usage
  - and fast query times

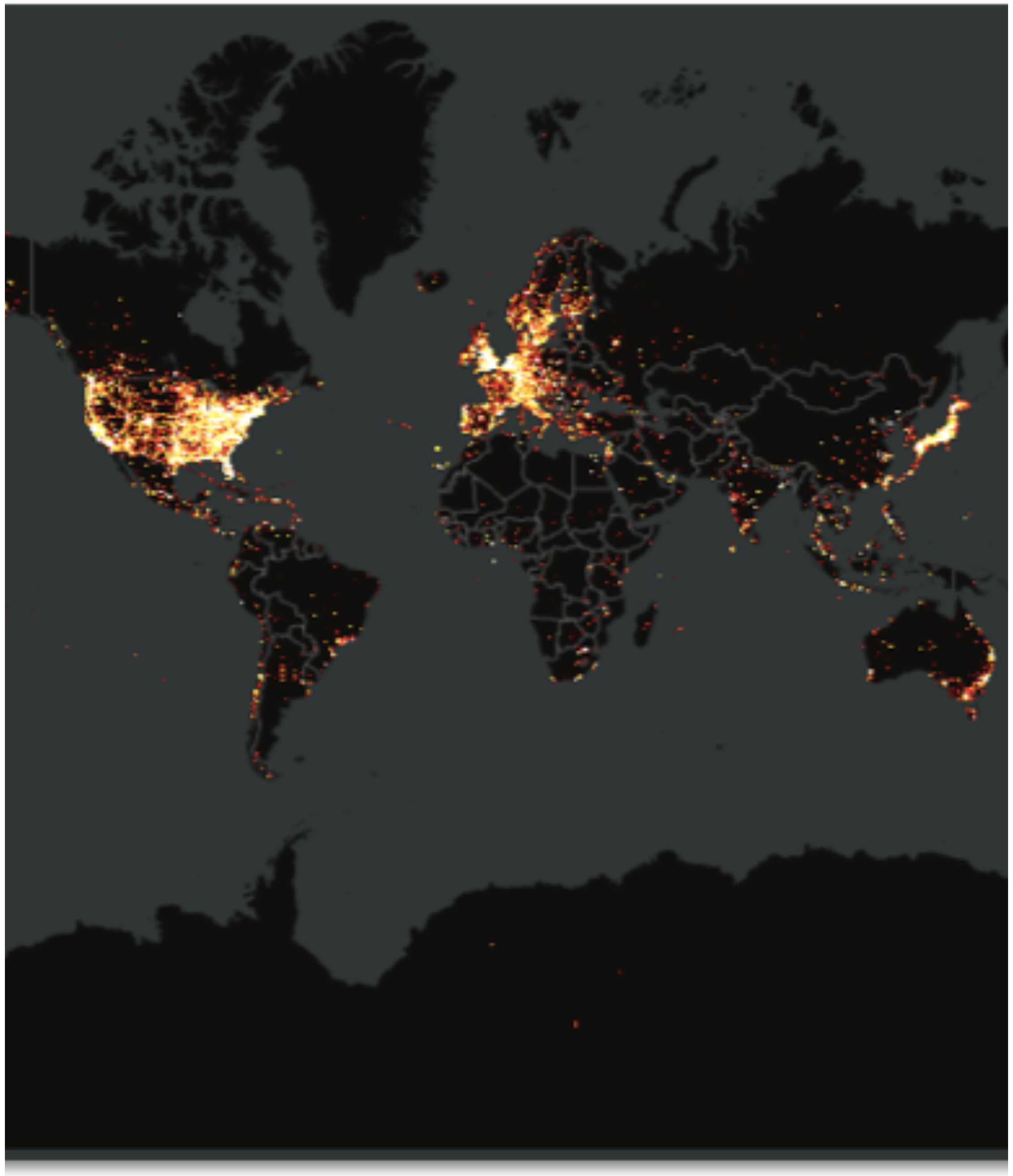
What must it do?

- Query: produce a heatmap of the world



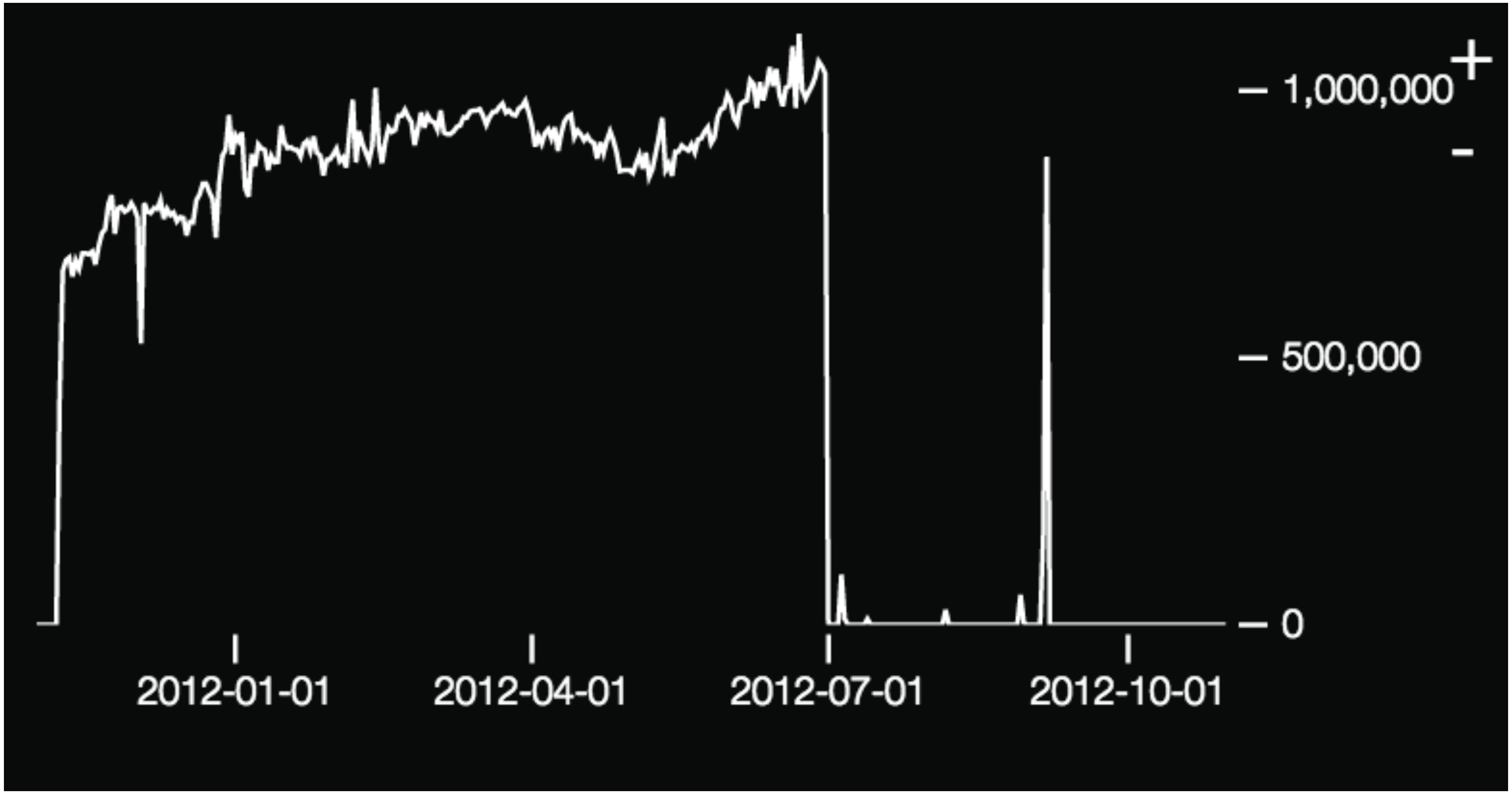
What must it do?

- Produce a heatmap of the world **in 2005**



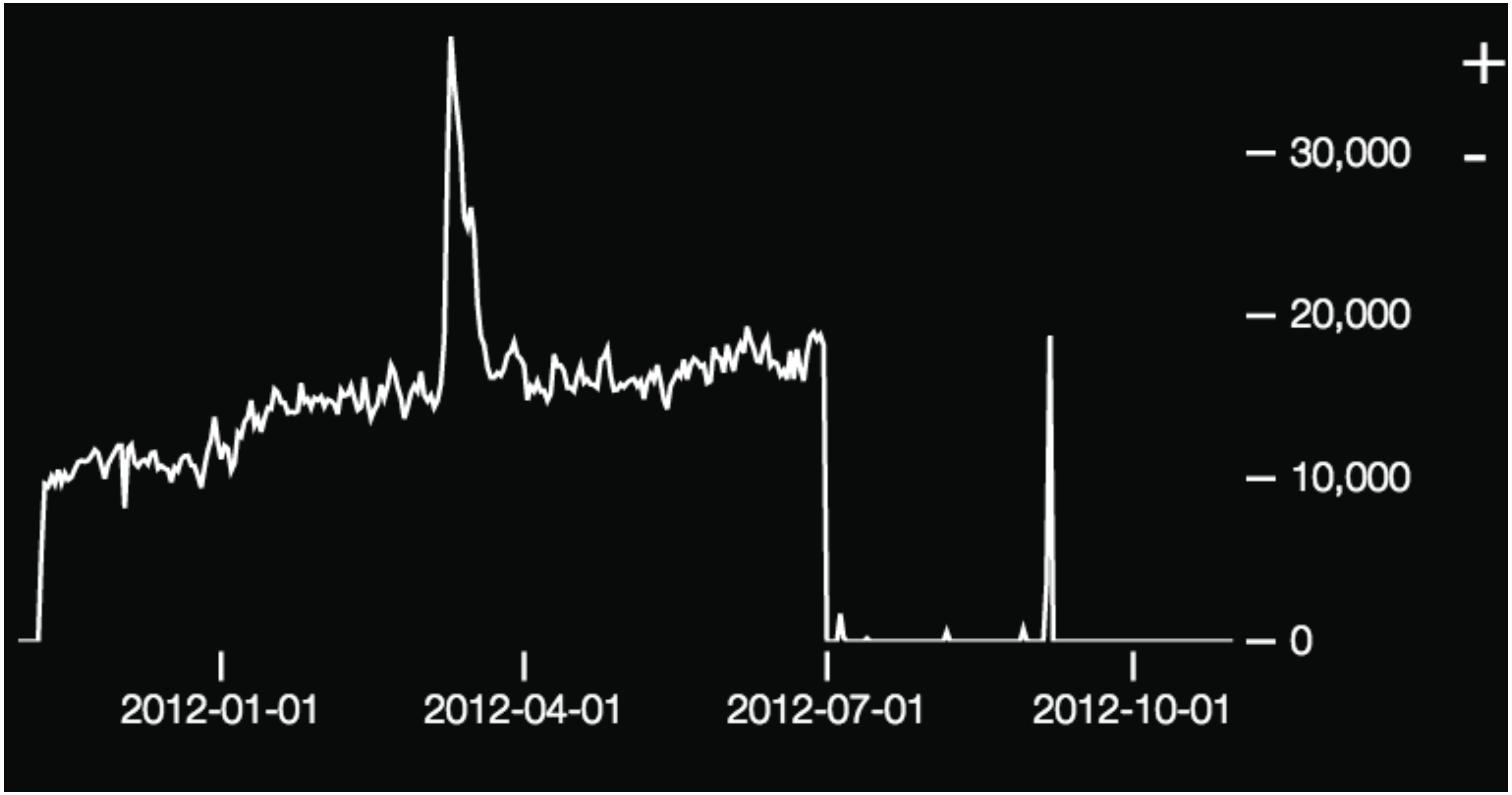
What must it do?

- Query: produce a time series of tweet counts



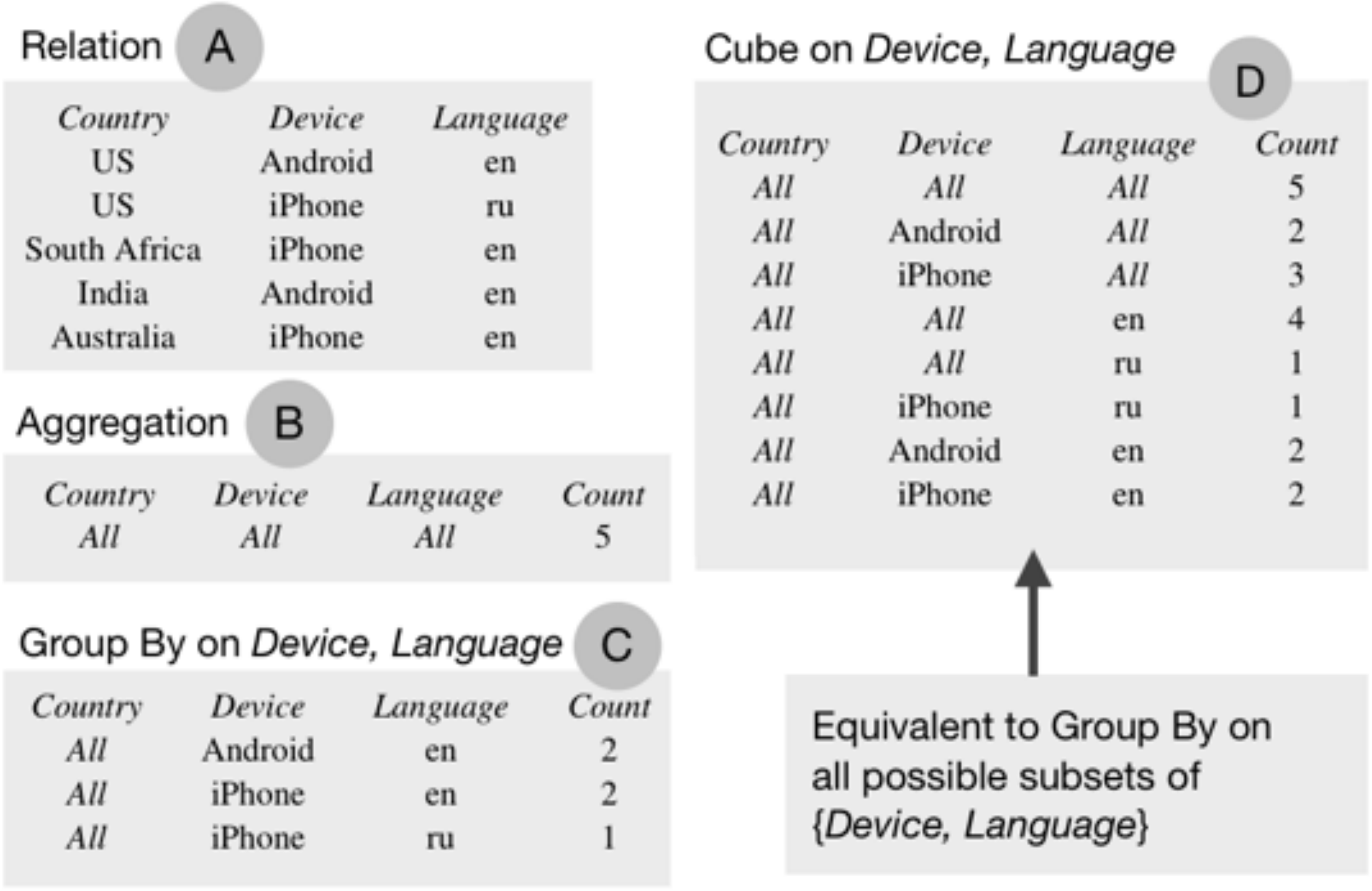
What must it do?

- Query: produce a time series of tweet counts **in Central Texas**



Nanocubes are..

- … multiscale
- … spatiotemporal
- … sparse
- … in-memory
- **data cubes**



Relation A

Country	*Device*	*Language*
US	Android	en
US	iPhone	ru
South Africa	iPhone	en
India	Android	en
Australia	iPhone	en

Aggregation B

Country	*Device*	*Language*	*Count*
All	*All*	*All*	5

Group By on *Device, Language* C

Country	*Device*	*Language*	*Count*
All	Android	en	2
All	iPhone	en	2
All	iPhone	ru	1

Cube on *Device, Language* D

Country	*Device*	*Language*	*Count*
All	*All*	*All*	5
All	Android	*All*	2
All	iPhone	*All*	3
All	*All*	en	4
All	*All*	ru	1
All	iPhone	ru	1
All	Android	en	2
All	iPhone	en	2

Equivalent to Group By on all possible subsets of {*Device, Language*}

(It seems that “Data cubes” means something different to you!)

NOAO Meeting, tools for Big Data, March 16 2015

Demos

How does it work?

- We avoid exponential memory blowup by carefully reusing results of different queries

- eg. Don’t store results twice if query for year=2005 is equal to query for year=2005 and month=January

- Many more ugly, uninteresting data structures tricks

Performance numbers

Build time:

dataset	n	memory	time	keys	cardinality
brightkite	4.5M	1.6GB	3.5m	3.5M	2^74
cust. tix	7.8M	2.5GB	8.47m	7.8M	2^69
flights	121M	2.3GB	31.13m	43.3M	2^75
twitter-small	210M	10.2GB	1.23h	116M	2^53
twitter	210M	46.4GB	5.87h	136M	2^60
cdrs	1B	3.6GB	3.08h	96.3M	2^69

Query time is dominated by network latency and bandwidth (<0.1s)

Preprocessing time is ~100k events/s

Implementation

- C++ backend, HTML5 front-end
    - Program reads data sequentially, then opens a web server
- Open source: https://github.com/laurolins/nanocube
- Runs on cell phones and tablets (!)

Astronomy demos

- (I’m not an astronomer, so apologies in advance!)
    - But imagine an interactive version of the Hertzsprung-Russell diagram
- it would not be hard to create an interactive tool to select/ visualize subsets of stars based on
    - temperature x magnitude x other attributes (sky location, etc).
- Today: two small star catalogs I could find and parse myself

Limitations

- Relatively small number of dimensions (4-8 ideal)
- in-memory for now, so it won't work for arbitrarily-large dataset
    - External memory implementation is coming
    - still, very large ones, 1B daily events with d=5 in production use
- **work-in-progress, usability-wise**

Where do we go from here?

- Store more than counts
    - Anything that behaves like a **monoid:** lots of statistics are monoids
- Rebuild the infrastructure of EDA assuming this is the available backend
    - Clustering, data fitting, modelling
    - Push interactive exploration into the computation infrastructure
- **How to reconcile interaction with the multiple-comparisons problem?**

Thank you!

- http://nanocubes.net for links to paper, source code, documentation
- http://github.com/laurolins/nanocube is the github page