Professional Data - Wrestling Techniques Using Elasticsearch's Aggregation Framework

Mark Harwood @elasticmark
18/6/2015

elastic

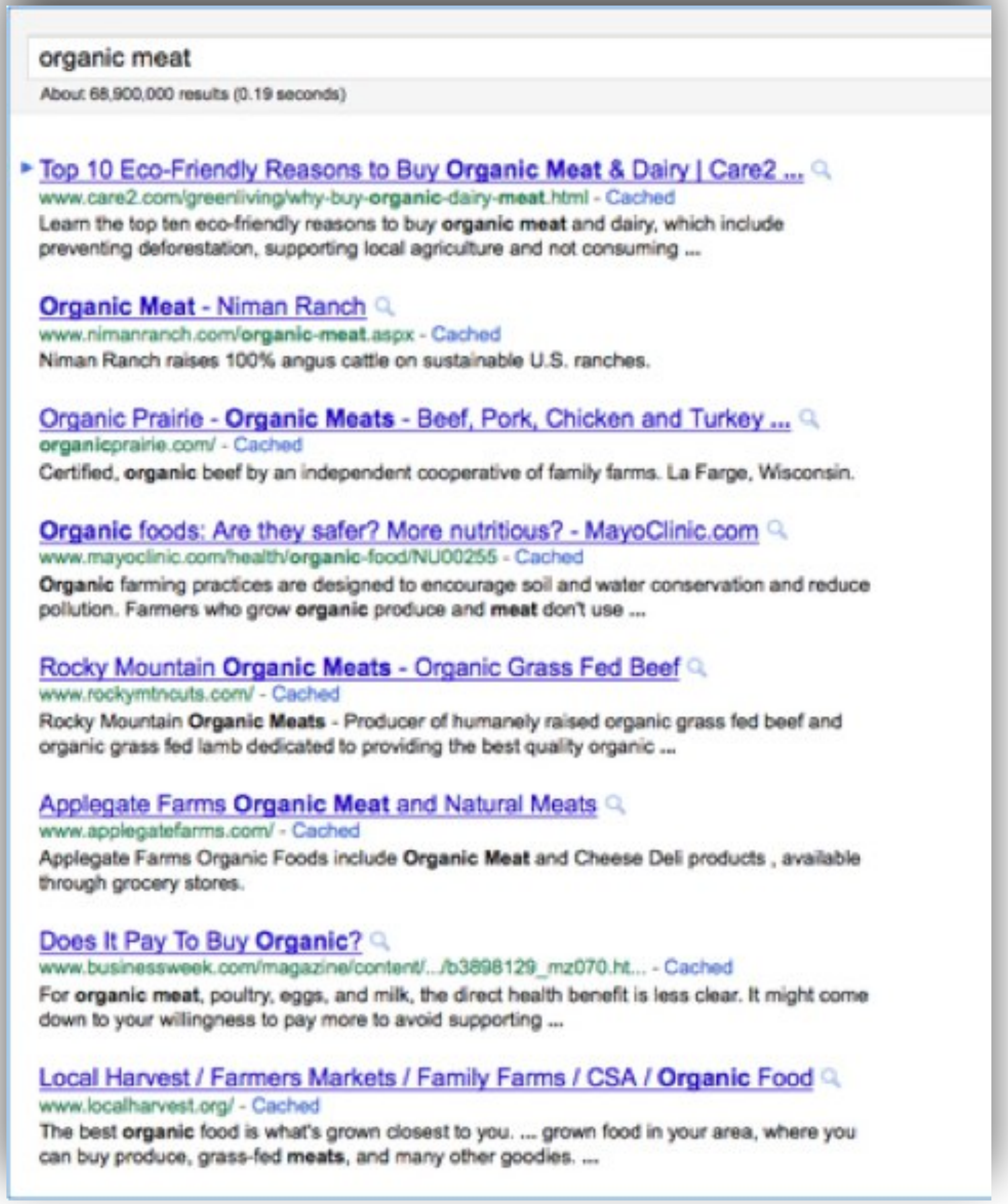
Some brief background

How search moved into analytics

Search interface 1.0

search box

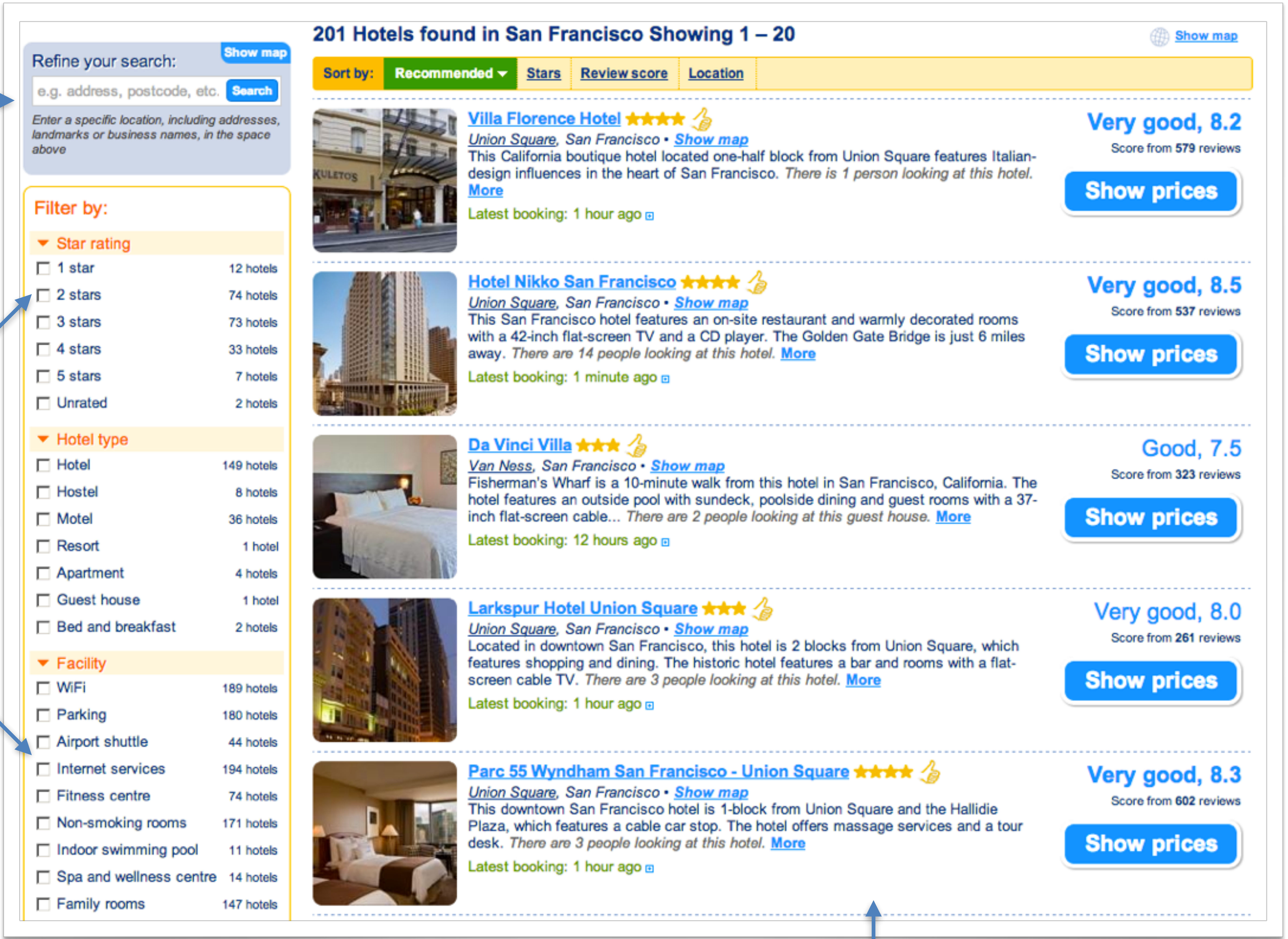
“10 blue links”



Search interface 2.0

search box

Facets - *fast* summaries of all the stuff beyond page 1



"10 blue links"

Analytics interface of today

search selections

"aggregations"

Optimised for real-time:

	Search	*Analytics*
Business question	***"Help me find the best documents"***	***"Collectively, what do these documents tell me about my business?"***
Enablers	***Fuzzy matching, relevance ranking, auto-complete, filtering (time/geo..) highlighting..***	***Summaries, patterns, trends, outliers, visualization***

Caution: performing analytics on fuzzy sets can cause issues….

Worked examples

Practical uses of aggregations

UK Housing data

400k documents of this form:


```
{
    "town": "COVENTRY",
    "status": "A",
    "location": {
        "lat": 52.401126222,
        "lon": -1.568220925
    },
    "district": "COVENTRY",
    "locality": "",
    "price": 119000,
    "housetype": "T",
    "oldnew": "N",
    "county": "WEST MIDLANDS",
    "duration": "F",
    "street": "STANDARD AVENUE",
    "postcode": "CV49BT",
    "date": "2014-01-31 00:00",
    "paon": "121",
    "saon": ""
}
```


Postcodes to lat/lon: http://data.gov.uk/dataset/code-point-open
House sale prices: https://www.gov.uk/government/statistical-data-sets/price-paid-data-downloads

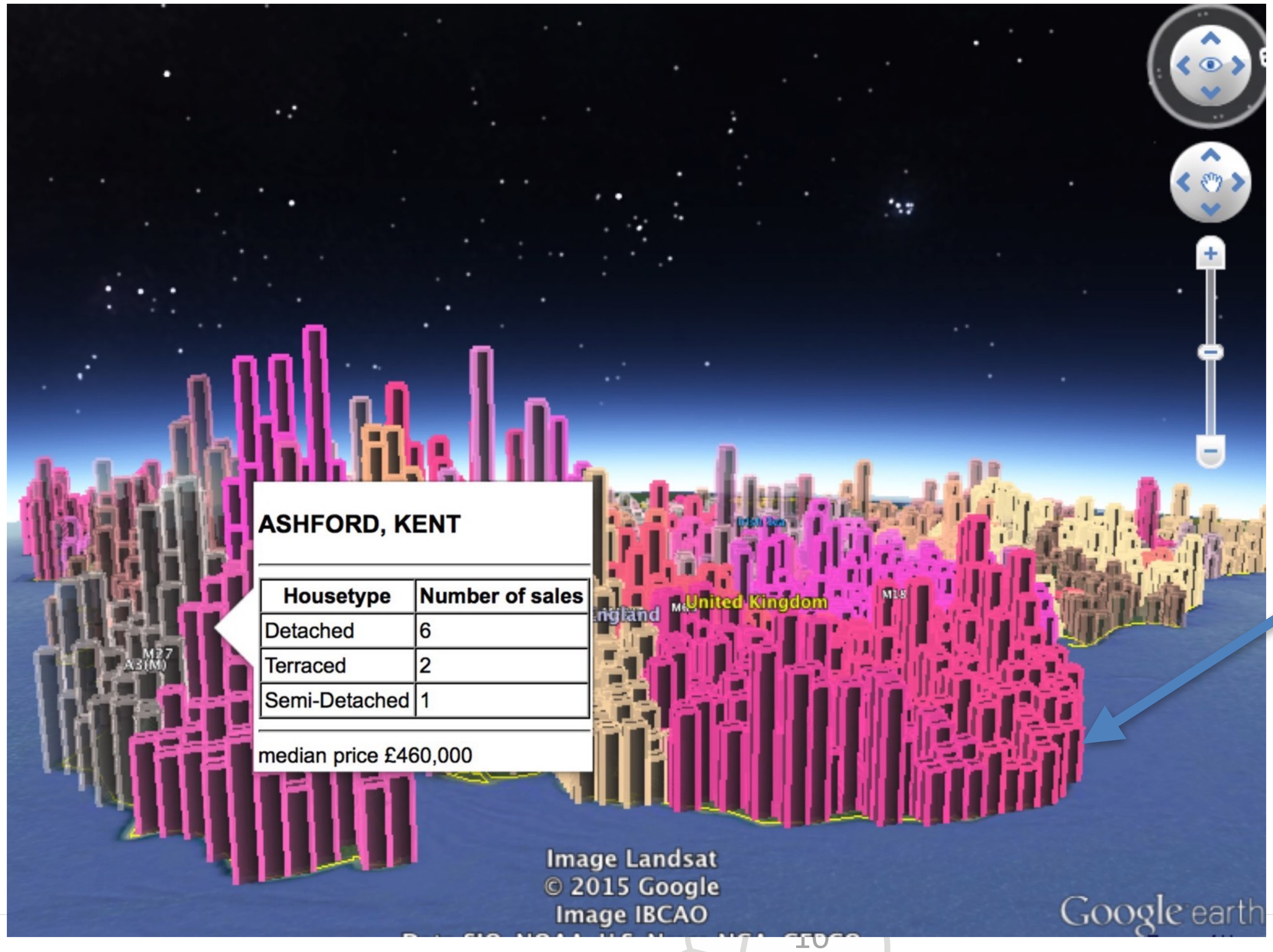
UK Housing data

```
{
    "aggregations" : {
        "map" : { "geohash_grid" : { "field":"location", "precision":5},
                  "aggregations":{
                        "priceBands":{"percentiles":{"field":"price"}},
                        "county":{"terms":{"field":"county.raw", "size":1}},
                        "town":{"terms":{"field":"town.raw", "size":1}},
                        "housetype":{"terms":{"field":"housetype.raw", "size":10}
                  },
            }
        }
    }
}
```

UK Housing data : geohash_grid

```
{
    "aggregations" : {
        "map" : { "geohash_grid" : { "field":"location", "precision":5},
                    "aggregations":{
                        "priceBands":{"percentiles":{"field":"price"}},
                        "county":{"terms":{"field":"county.raw", "size":1}},
                        "town":{"terms":{"field":"town.raw", "size":1}},
                        "housetype":{"terms":{"field":"housetype.raw", "size":10}
                    },
                }
            }
        }
    }
```

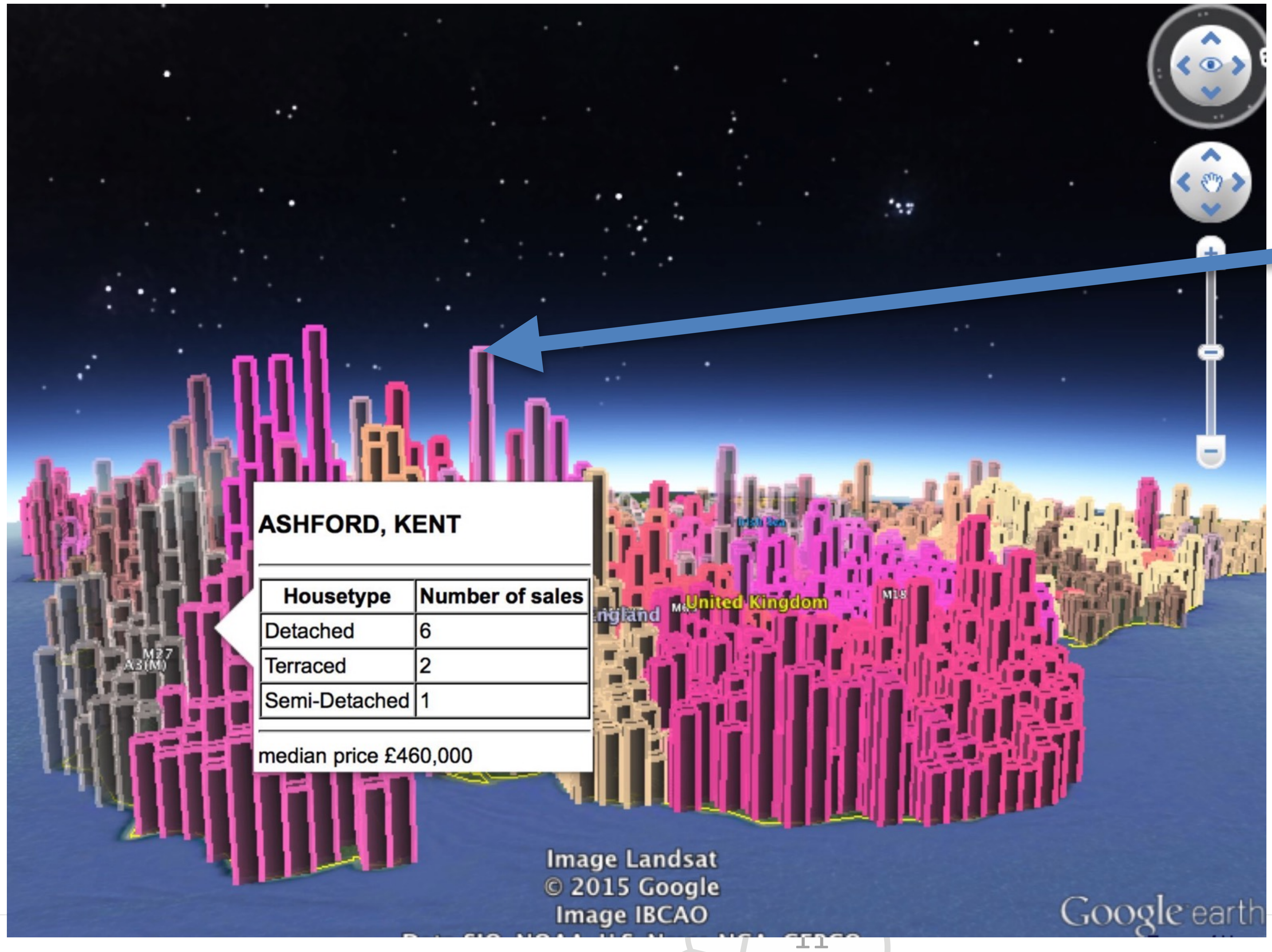
Geohash precision determines width of cells used to organize results



Visualization code and KML: https://goo.gl/WkWKmh

UK Housing data: percentiles

```
{
    "aggregations" : {
        "map" : { "geohash_grid" : { "field":"location", "precision":5},
                  "aggregations":{
                      "priceBands":{"percentiles":{"field":"price"}},
                      "county":{"terms":{"field":"county.raw", "size":1}},
                      "town":{"terms":{"field":"town.raw", "size":1}},
                      "housetype":{"terms":{"field":"housetype.raw", "size":10}
                  },
            }
        }
    }
}
```

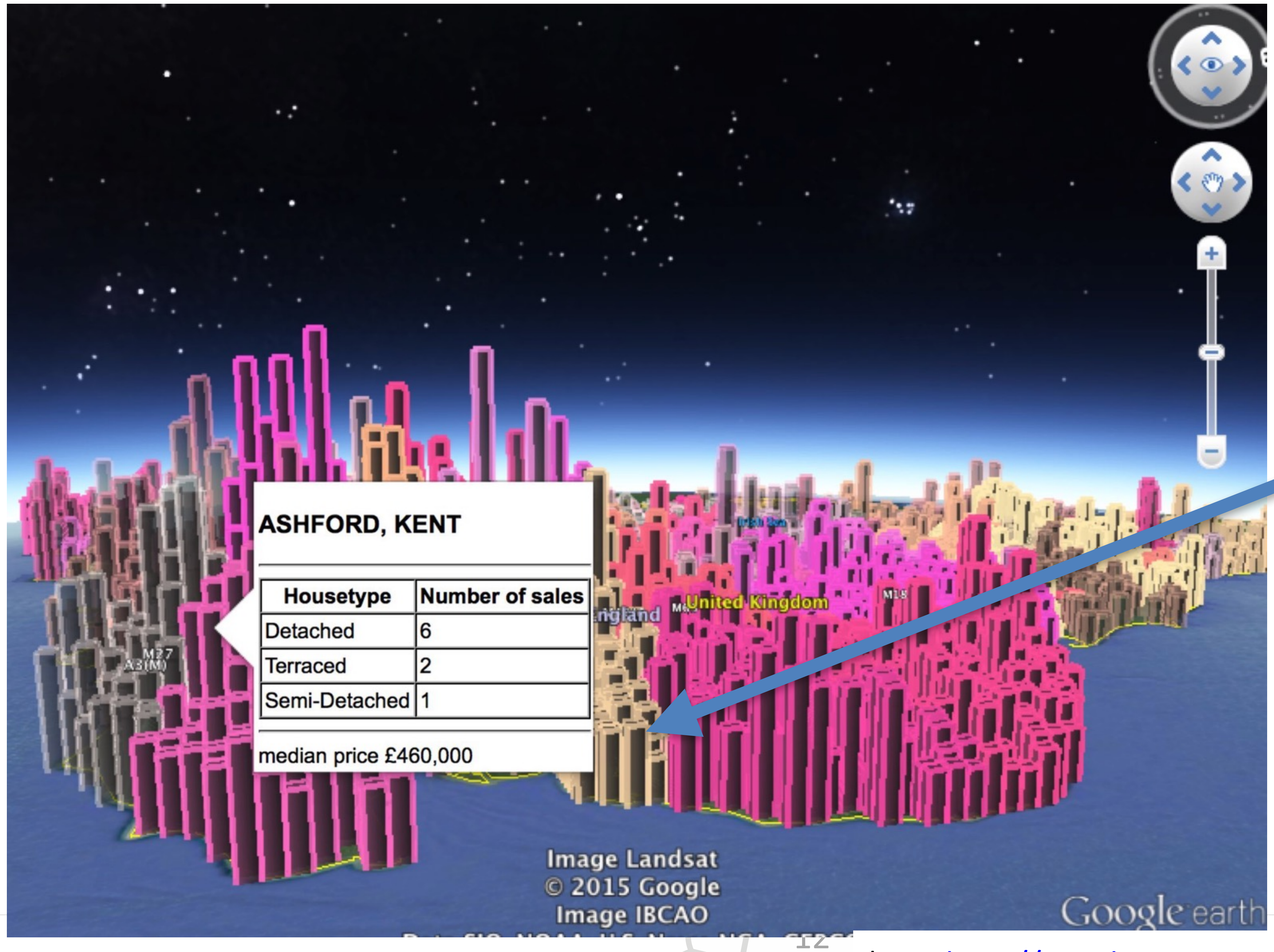


Median house price used for cell height
Avoids avg skew by outliers *

https://www.elastic.co/blog/averages-can-dangerous-use-percentile

UK Housing data: terms

```
{
    "aggregations" : {
        "map" : { "geohash_grid" : { "field":"location", "precision":5},
                  "aggregations":{
                      "priceBands":{"percentiles":{"field":"price"}},
                      "county":{"terms":{"field":"county.raw", "size":1}},
                      "town":{"terms":{"field":"town.raw", "size":1}},
                      "housetype":{"terms":{"field":"housetype.raw", "size":10}
                  },
            }
        }
    }
}
```

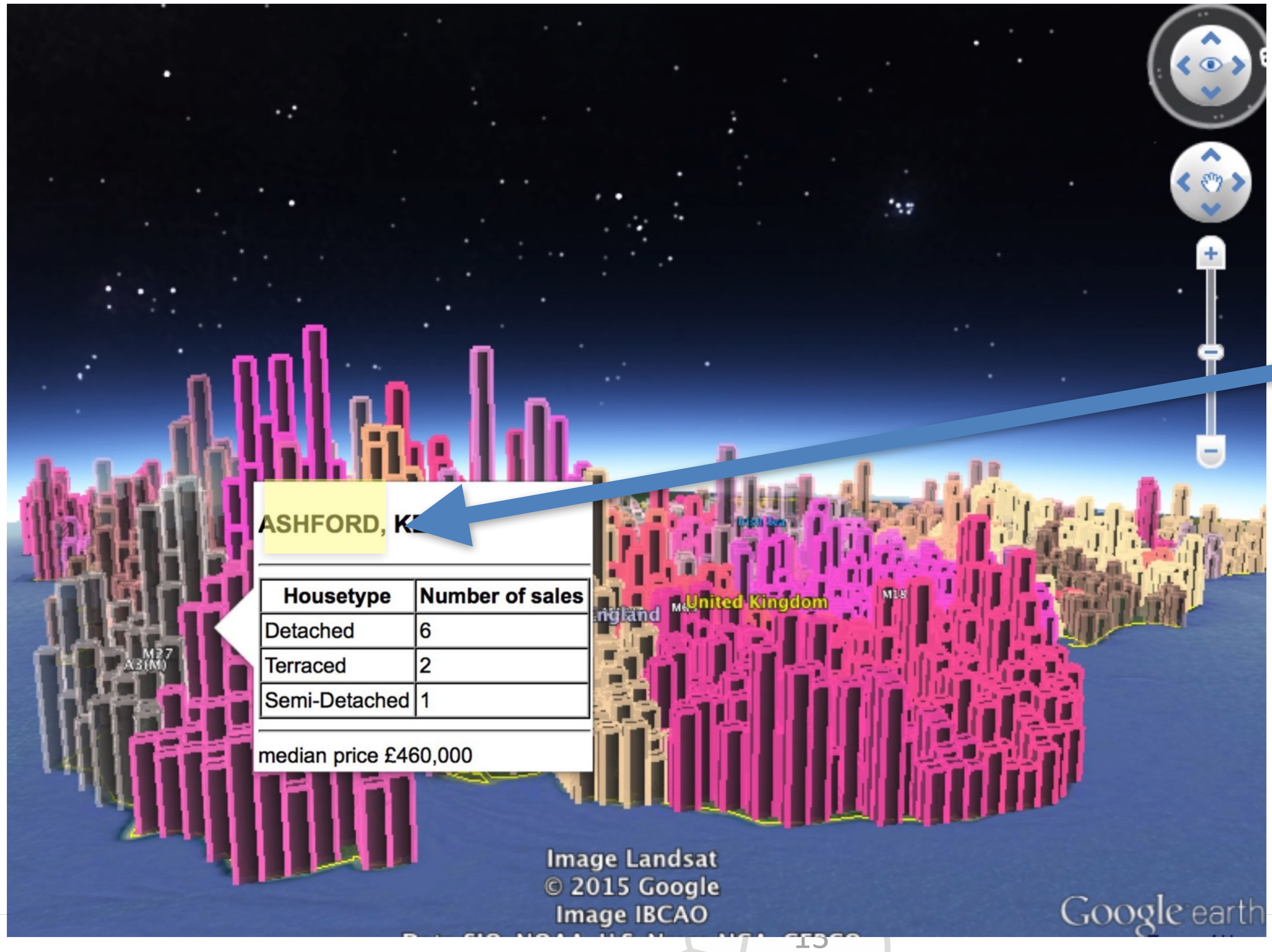


Most popular county name is used to pick a colour - reveals county boundaries *

* see http://googlemapsmania.blogspot.co.uk/2015/05/crowdsourcing-neighborhood-boundaries.html

UK Housing data: terms

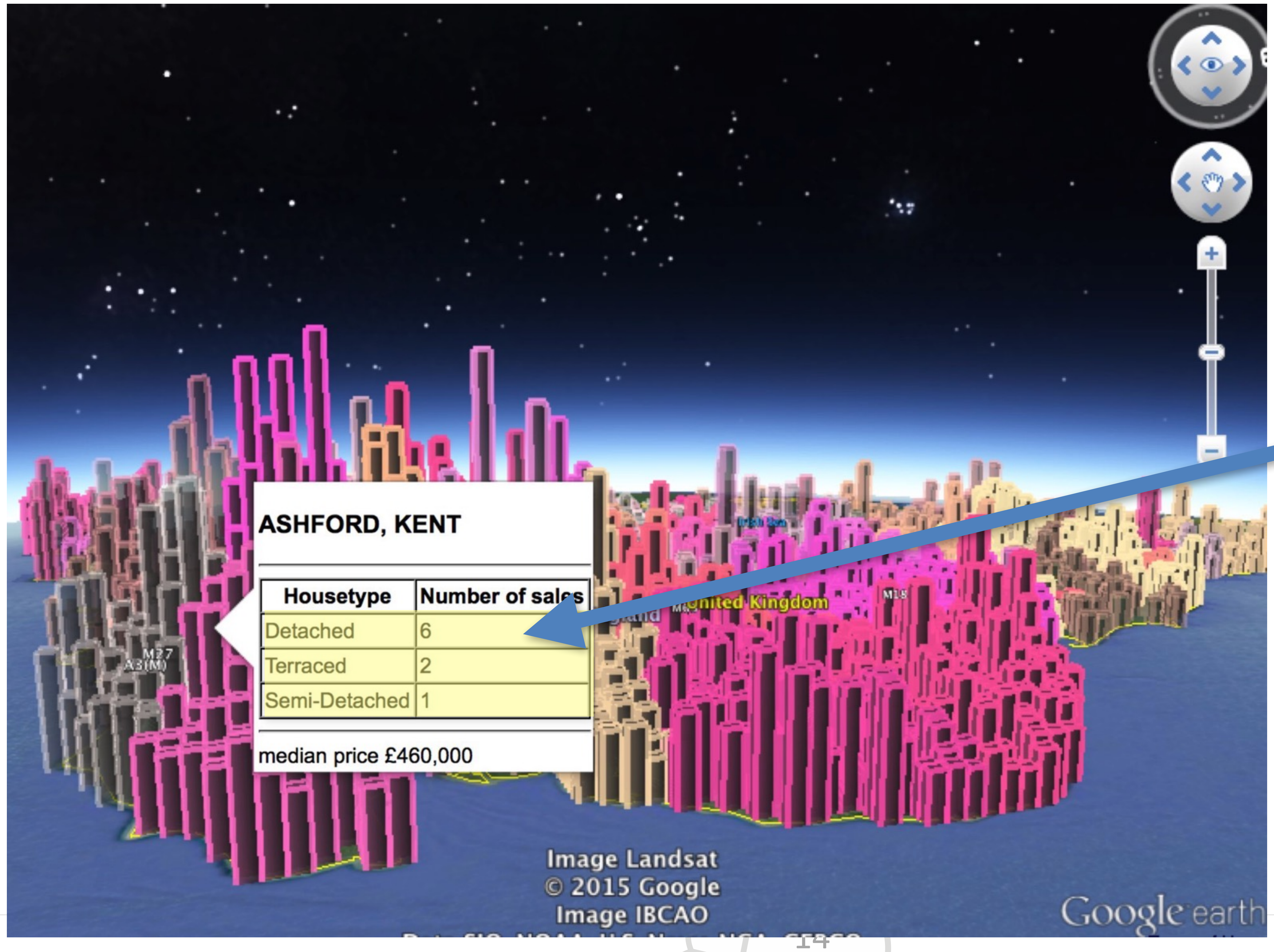
```
{
    "aggregations" : {
        "map" : { "geohash_grid" : { "field":"location", "precision":5},
                  "aggregations":{
                      "priceBands":{"percentiles":{"field":"price"}},
                      "county":{"terms":{"field":"county.raw", "size":1}},
                      "town":{"terms":{"field":"town.raw", "size":1}},
                      "housetype":{"terms":{"field":"housetype.raw", "size":10}
                  },
              }
          }
      }
 }
```



Most popular town name reveals most-likely-to-be-useful label

UK Housing data: terms

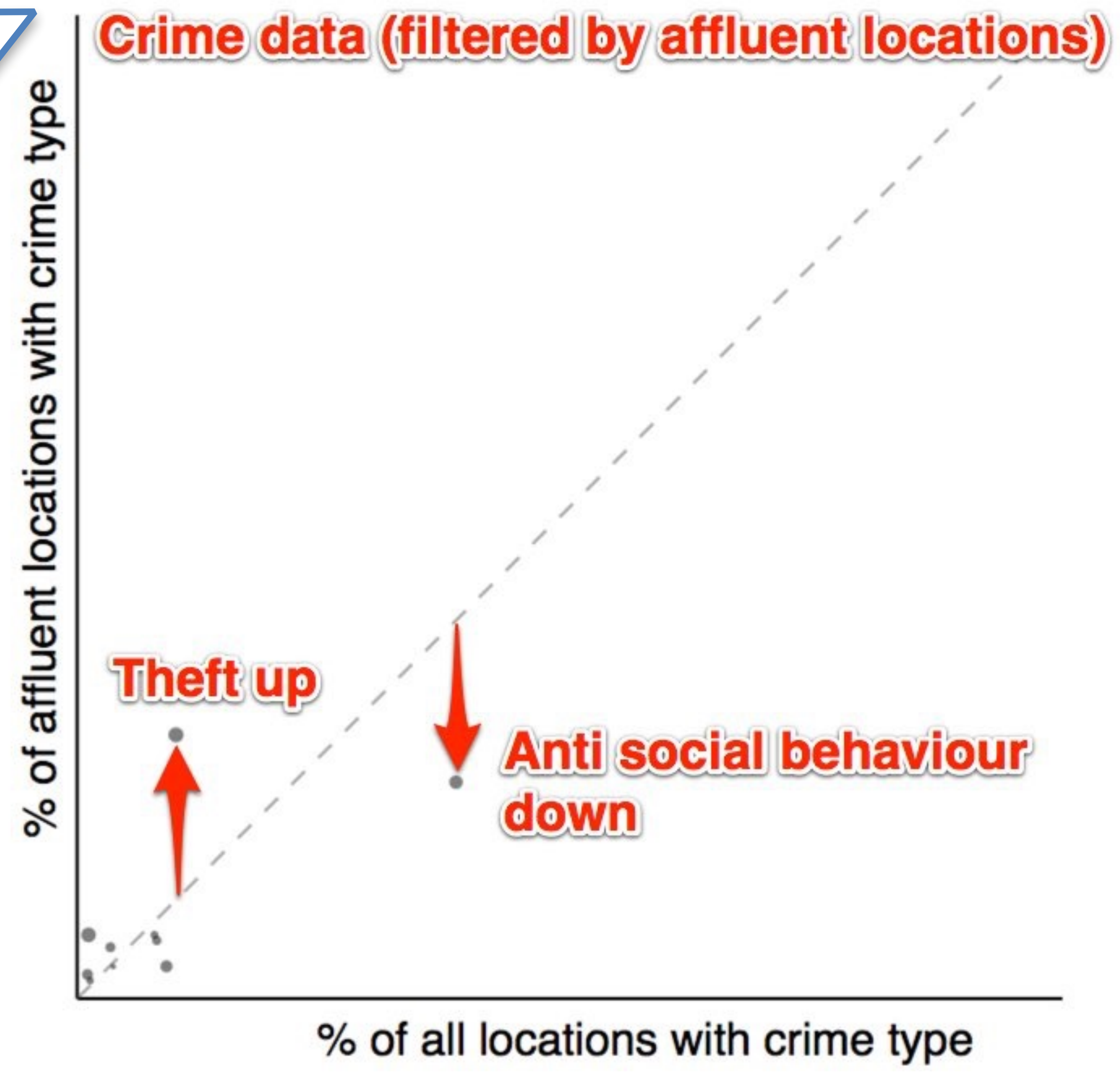
```
{
    "aggregations" : {
        "map" : { "geohash_grid" : { "field":"location", "precision":5},
                  "aggregations":{
                       "priceBands":{"percentiles":{"field":"price"}},
                       "county":{"terms":{"field":"county.raw", "size":1}},
                       "town":{"terms":{"field":"town.raw", "size":1}},
                       "housetype":{"terms":{"field":"housetype.raw", "size":10}
                  },
            }
        }
    }
 }
```



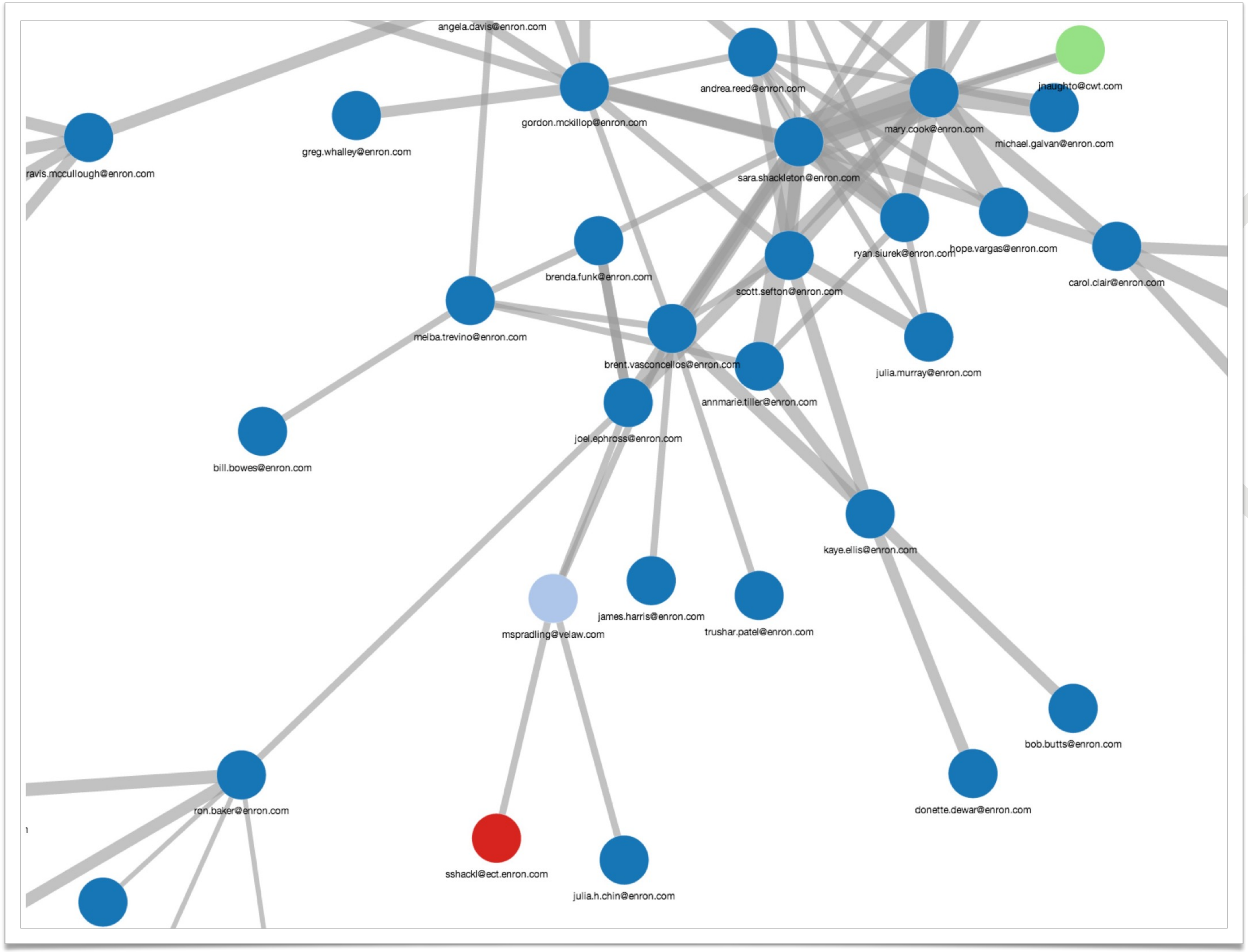
League-table of most popular house types in each cell.

Geo as a common link between datasets: housing ->crime


```
"query":{
    "filtered" : {
        "filter" : {
            "bool":{
                "should":
            [{"geo_bounding_box":{"location":{"top_left":{"lat":51.50390625,"lon":-0.17578125},"bo
            {"geo_bounding_box":{"location":{"top_left":{"lat":51.3720703125,"lon":-0.3955078125},
            {"geo_bounding_box":{"location":{"top_left":{"lat":51.416015625,"lon":-0.615234375},"b
            ...
            ]
        }
    }
}
"aggs" : {
            "keywords":{"significant_terms":{"field":"crimeType"} }
}
```



Connected data: Enron emails

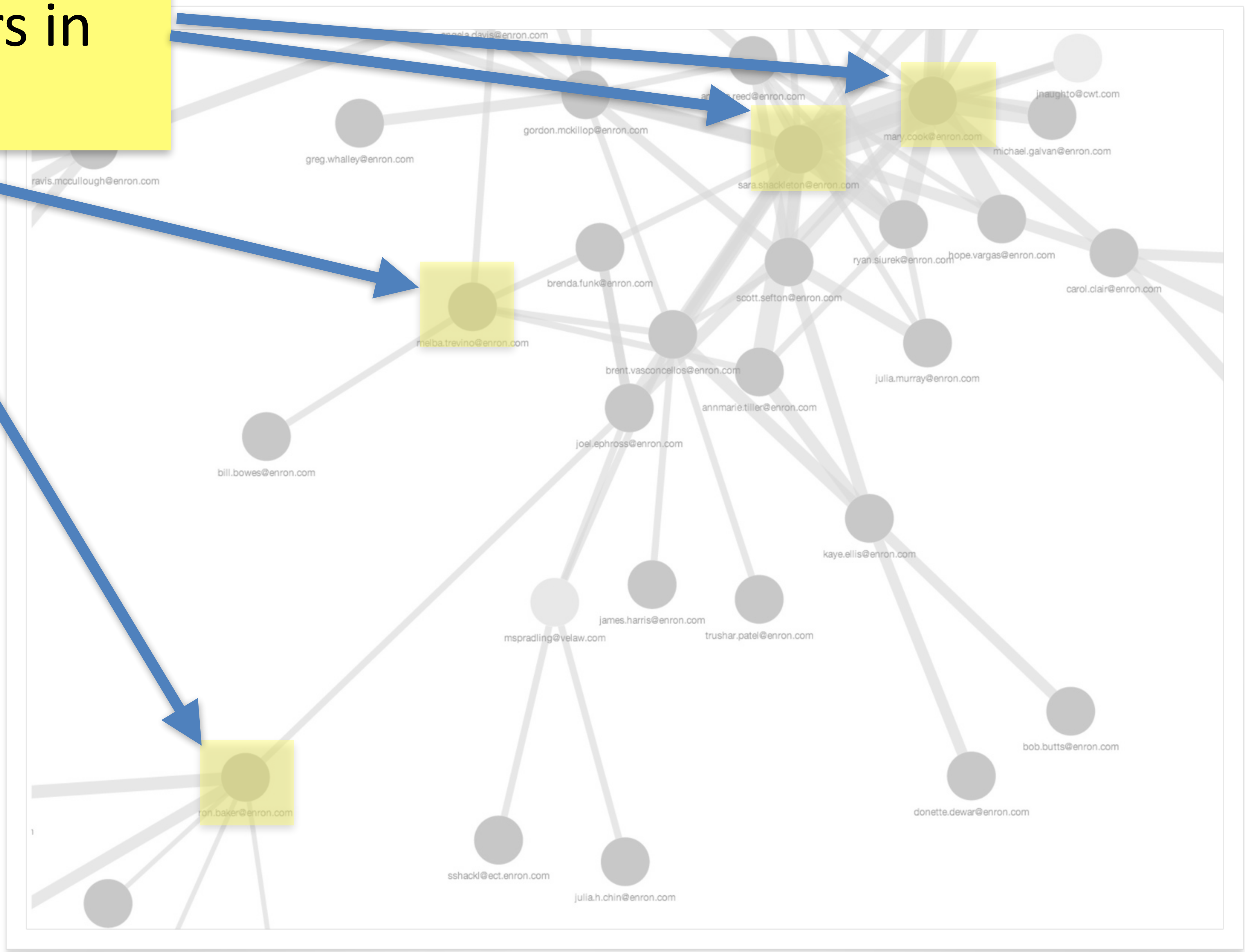


https://www.cs.cmu.edu/~./enron/

Connected data: Enron emails

```
{
  "aggregations": {
    "senders": {
      "terms": {
        "field": "from",
        "size": 50,
        "collect_mode": "breadth_first"
      },
      "aggs": {
        "recipients": {
          "terms": {
            "field": "to",
            "size": 50
          }
        }
      }
    }
  }
}
```

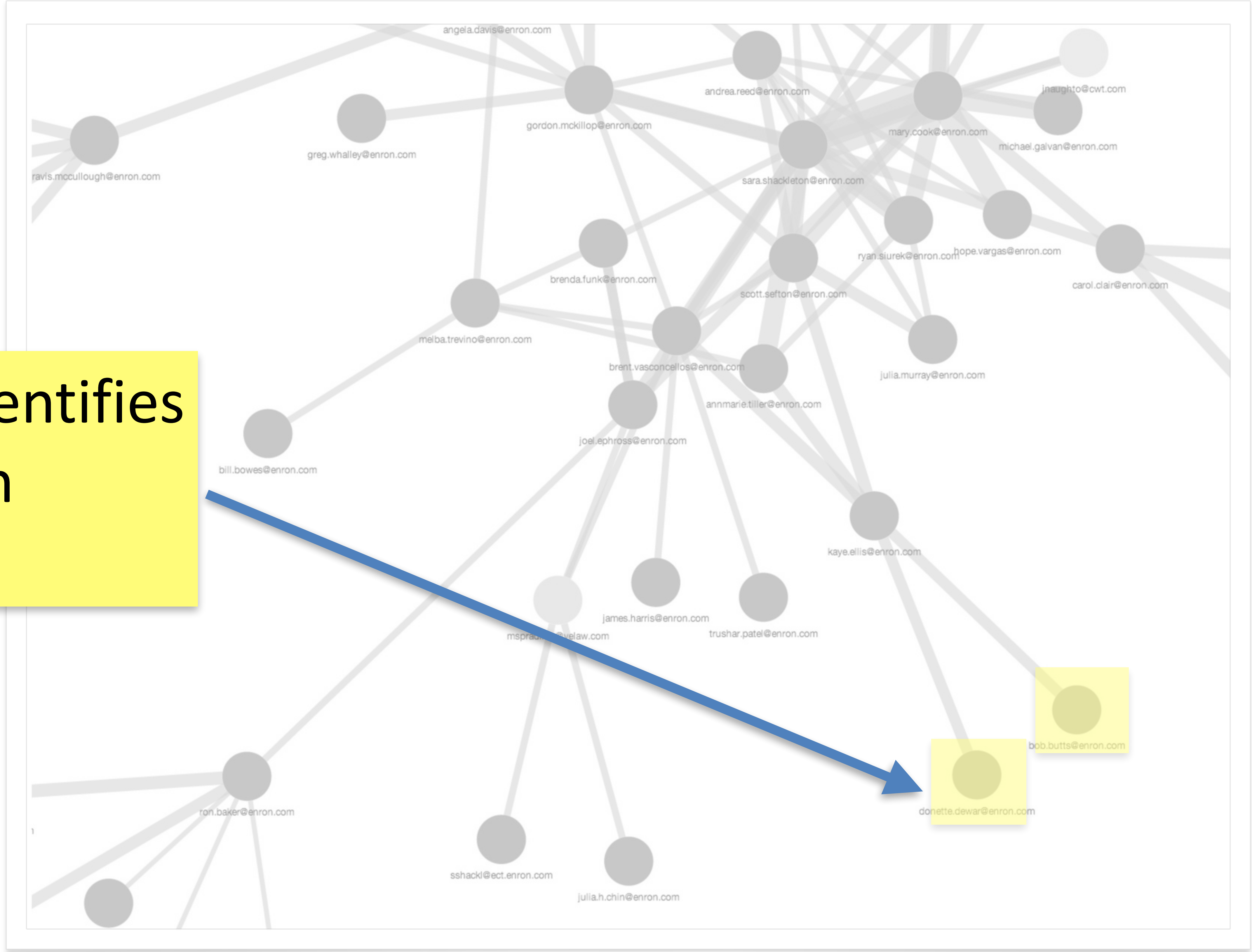
Identifies the 50 most prolific email senders in our index



Connected data: Enron emails

```
{
  "aggregations": {
    "senders": {
      "terms": {
        "field": "from",
        "size": 50,
        "collect_mode": "breadth_first"
      },
      "aggs": {
        "recipients": {
          "terms": {
            "field": "to",
            "size": 50
          }
        }
      }
    }
  }
}
```

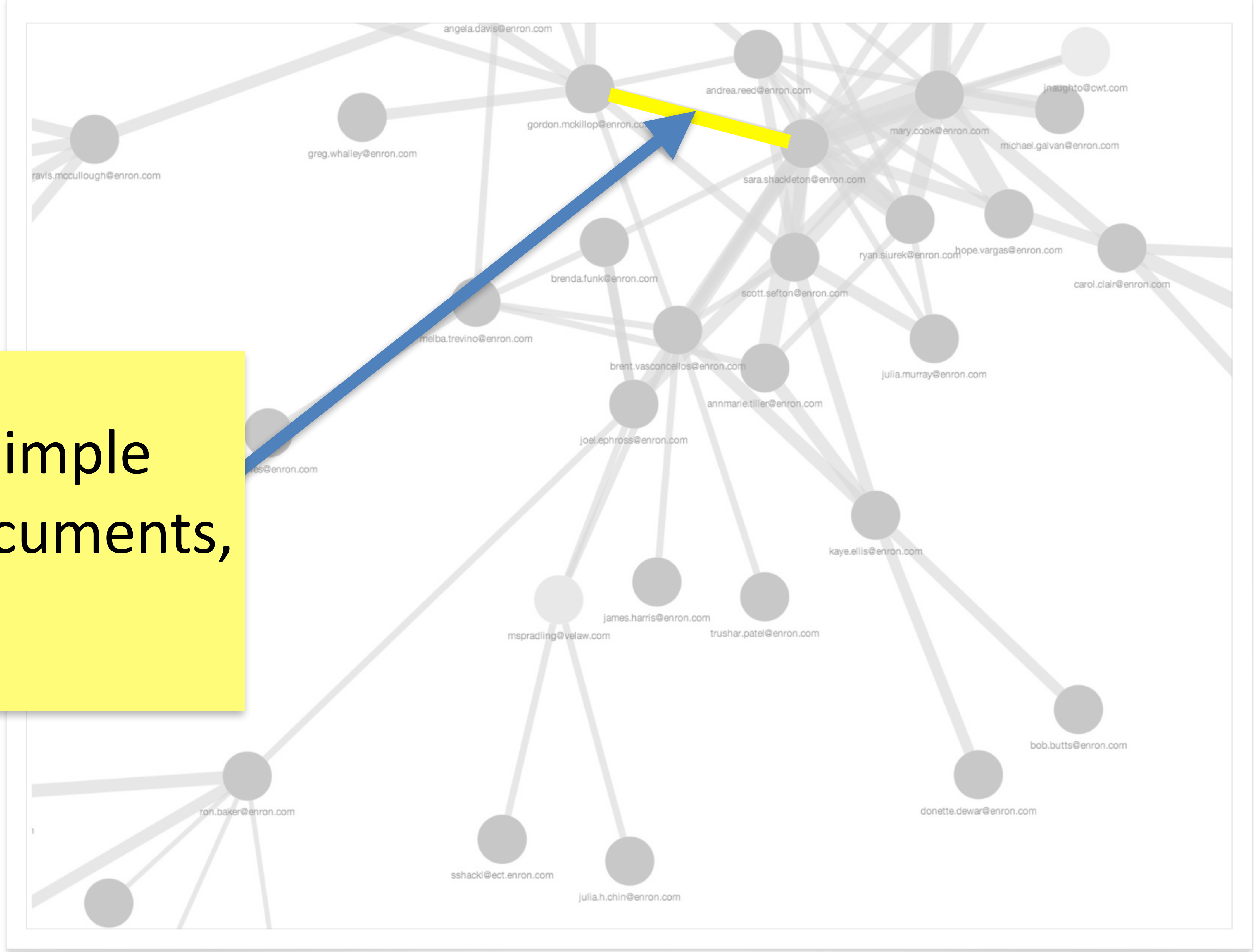
For each sender, identifies their most common correspondents



Connected data: Enron emails

```
{
  "aggregations": {
    "senders": {
      "terms": {
        "field": "from",
        "size": 50,
        "collect_mode": "breadth_first"
      },
      "aggs": {
        "recipients": {
          "terms": {
            "field": "to",
            "size": 50
          }
        }
      }
    }
  }
}
```

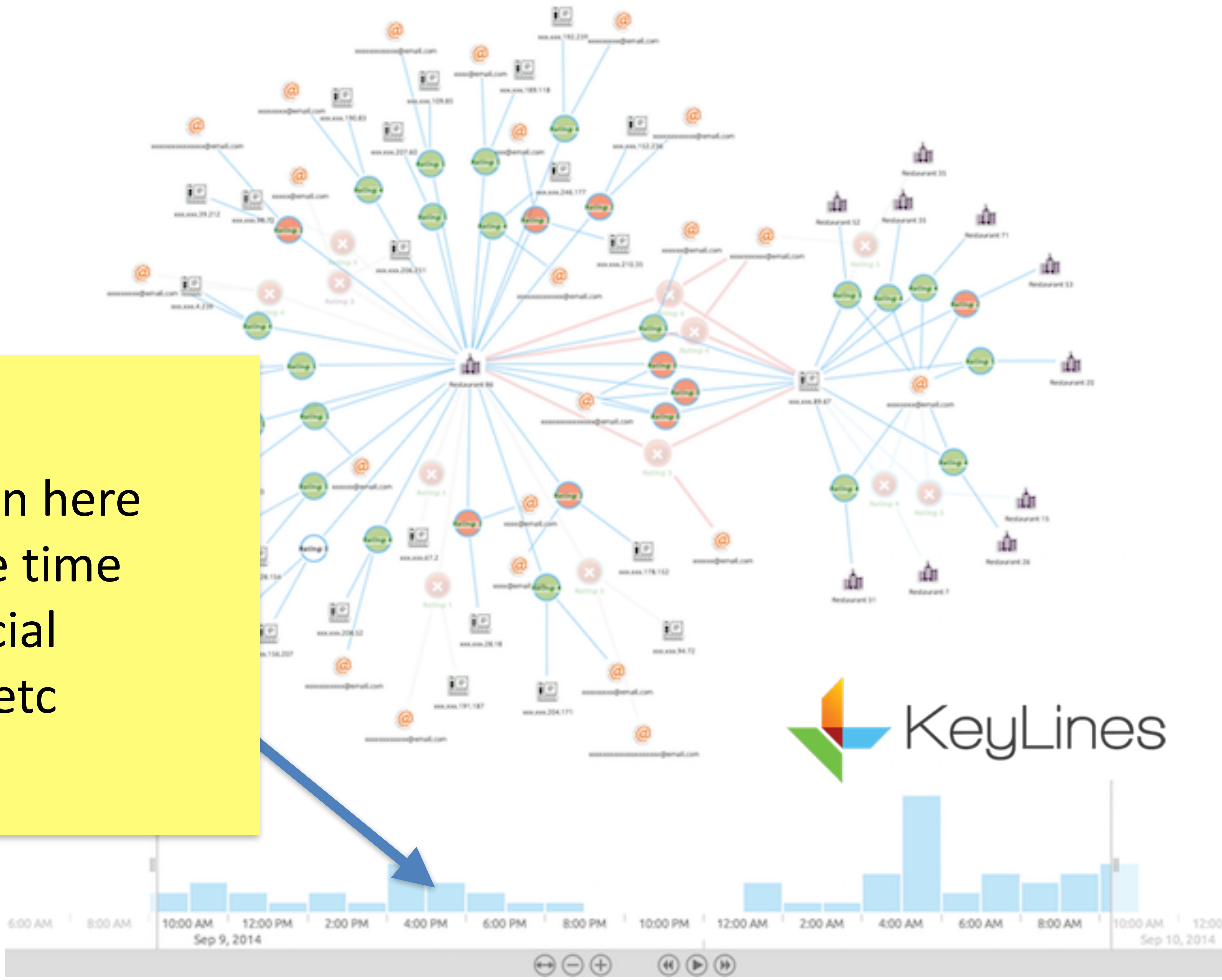
Line thickness is a simple count of shared documents, but there's more…



Connected data: Enron emails

```
{
  "aggregations": {
    "senders": {
      "terms": {
        "field": "from",
        "size": 50,
        "collect_mode": "breadth_first"
      },
      "aggs": {
        "recipients": {
          "terms": {
            "field": "to",
            "size": 50
          } "aggs":{…}
        }
      }
    }
  }
}
```

A sub aggregation here could summarise time periods or financial volumes traded etc



Connected data: Enron emails

```
{
  "aggregations": {
    "senders": {
      "terms": {
        "field": "from",
        "size": 50,
        "collect_mode": "breadth_first"
      },
      "aggs": {
        "recipients": {
          "terms": {
            "field": "to",
            "size": 50
          }
        }
      }
    }
  }
}
```

Important optimisation!
This line is the difference between:

1) Building a network of the whole business, then pruning selections or
2) Finding the top 50 email senders first *then* gathering only their connections.

The final results are the same (50 senders x 50 recipients) but the interim working state is vastly reduced.

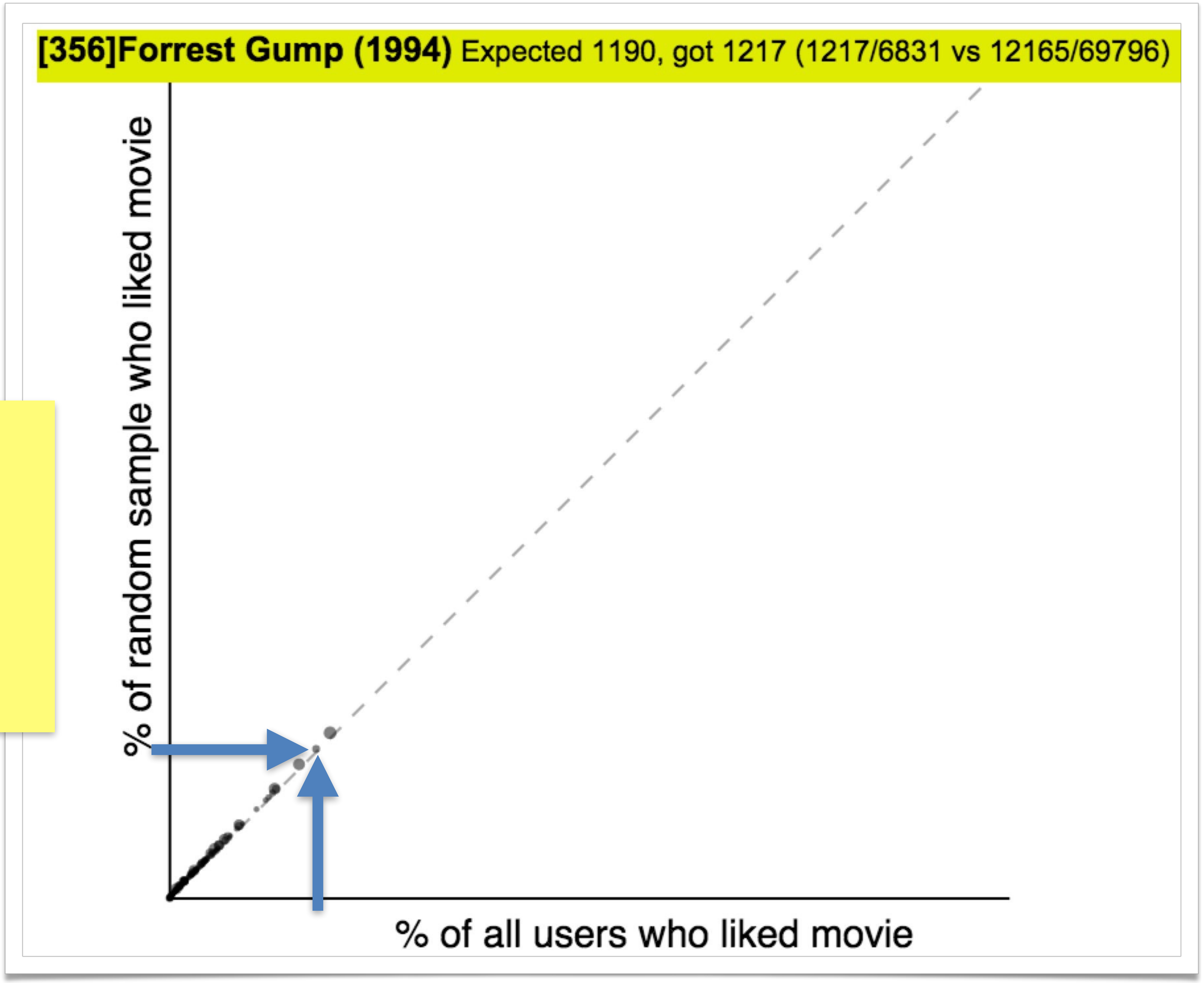
Recommendations: MovieLens data

```
{
    "movie": [260, 500, 1080...]
    "user": 8353
}
```

http://files.grouplens.org/datasets/movielens/ml-10m-README.html

Random samples should hold no surprises

- 17% of all people like "Forrest Gump"
- In a random sample of people, 17% of them will also like "Forrest Gump"



Dull. But in non-random samples something interesting happens.....

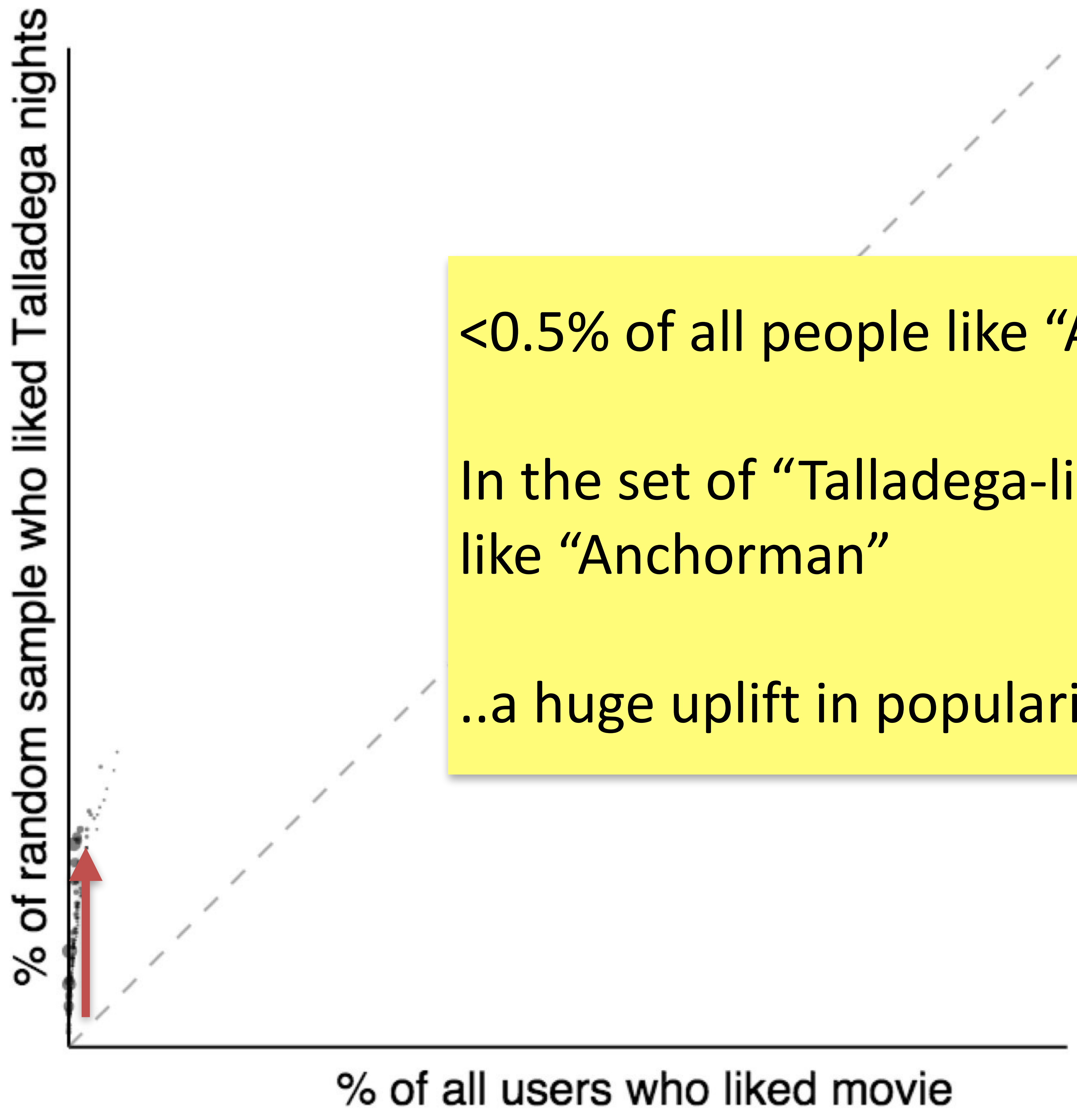
Non-random sample: people who liked “Talladega nights”

Find all people who liked movie #46970

```
{
    "query" :
    {
        "terms":{"movie": [46970] }
    },
    "aggs" : {
        "keywords": {
            "significant_terms": {
                "field": "movie",
                "size": 50
            }
        }
    }
}
```

Summarise how their movie tastes differ from everyone else

[8641]Anchorman: The Legend of Ron Burgundy (2004) Expected 1, got 55 (55/271 vs 374/69796)



<0.5% of all people like “Anchorman”

In the set of “Talladega-likers”, 20% of them like “Anchorman”

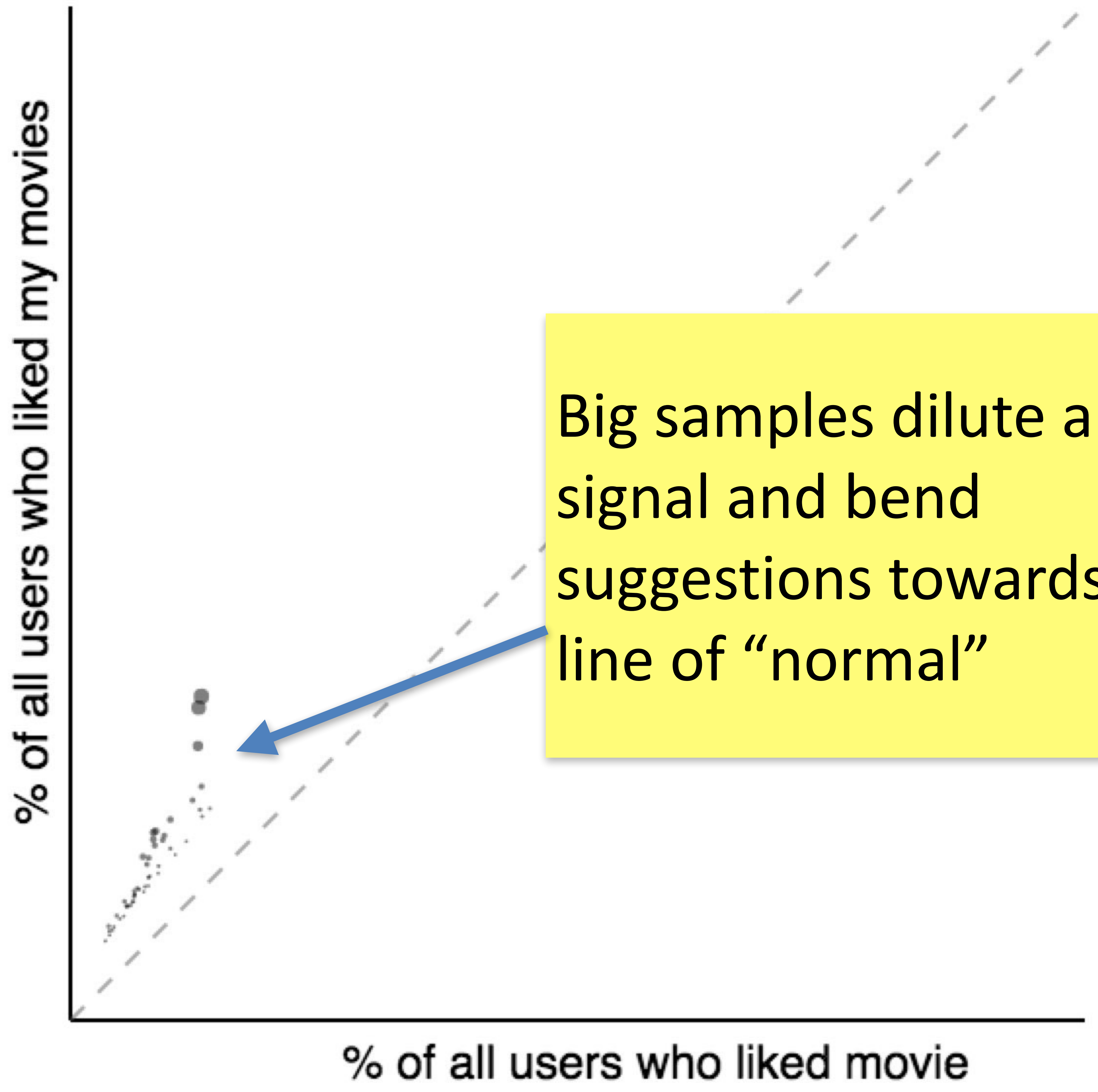
..a huge uplift in popularity from the norm!

Problem: avoid analysis of poorly focused sets

If my movie tastes include StarWars
I'm likely to match a lot of users

```
//talladega, blades of glory and star wars
var movieSelections=[46970, 52245, 260];
var queryJson=
{
    "query" :
    {
        "terms":{"movie": movieSelections }
    },
    "aggs" : {
        "keywords": {
            "significant_terms": {
                "field": "movie",
                "size": 50,
                "exclude": movieSelections
            }
        }
    }
}
```

[1196]Star Wars: Episode V - The Empire Strikes Back (1980)



Big samples dilute any signal and bend suggestions towards the line of "normal"

How do we get a smaller, representative sample of users?

```
//talladega, blades of glory and star wars
var movieSelections=[46970, 52245, 260];
```

Search relevance ranking/information theory to the rescue:

Ranking heuristic	*Effect*
***IDF** (Inverse Document Frequency)*	*People who share my rarer choices (Talladega) are ranked more highly than people who share my mainstream tastes (Star Wars)*
***TF** (Term frequency)*	*People who have watched a movie choice many times are preferable to those who have only watched it once*
***norms** (length normalization)*	*People who have a short list of movies that match are better than those with encyclopaedic lists*
***coord** (coordination factor)*	*People who share many of my choices are better than those with only a few*

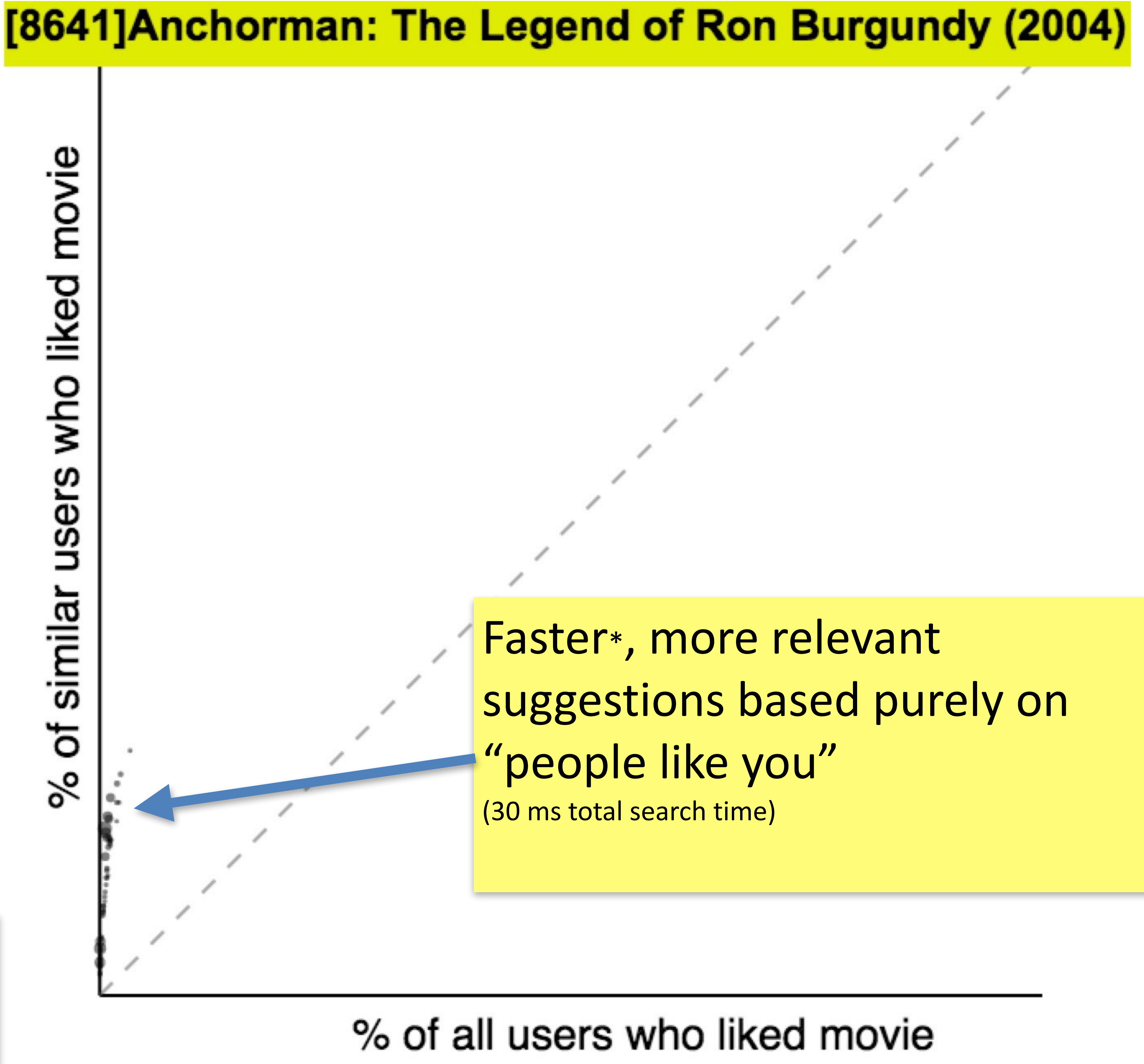
Putting search and analytics together..

```
//talladega, blades of glory and star wars
var movieSelections=[46970, 52245, 260];
var queryJson=
{
    "query" :
    {
        "terms":{"movie": movieSelections }
    },
    "aggs" : {
      "sample": {
        "sampler": {
          "shard_size": 200
        },
        "aggs": {
          "keywords": {
            "significant_terms": {
              "field": "movie",
              "size": 50,
              "exclude": movieSelections
            }
```

2.0 adds relevance ranked search for numeric fields

In 2.0 we can perform analytics on a sample of only the most relevant documents

2.0 exclusion lists are much more efficient



Faster*, more relevant suggestions based purely on “people like you”
(30 ms total search time)

Great, but..

There are limits to aggregations…

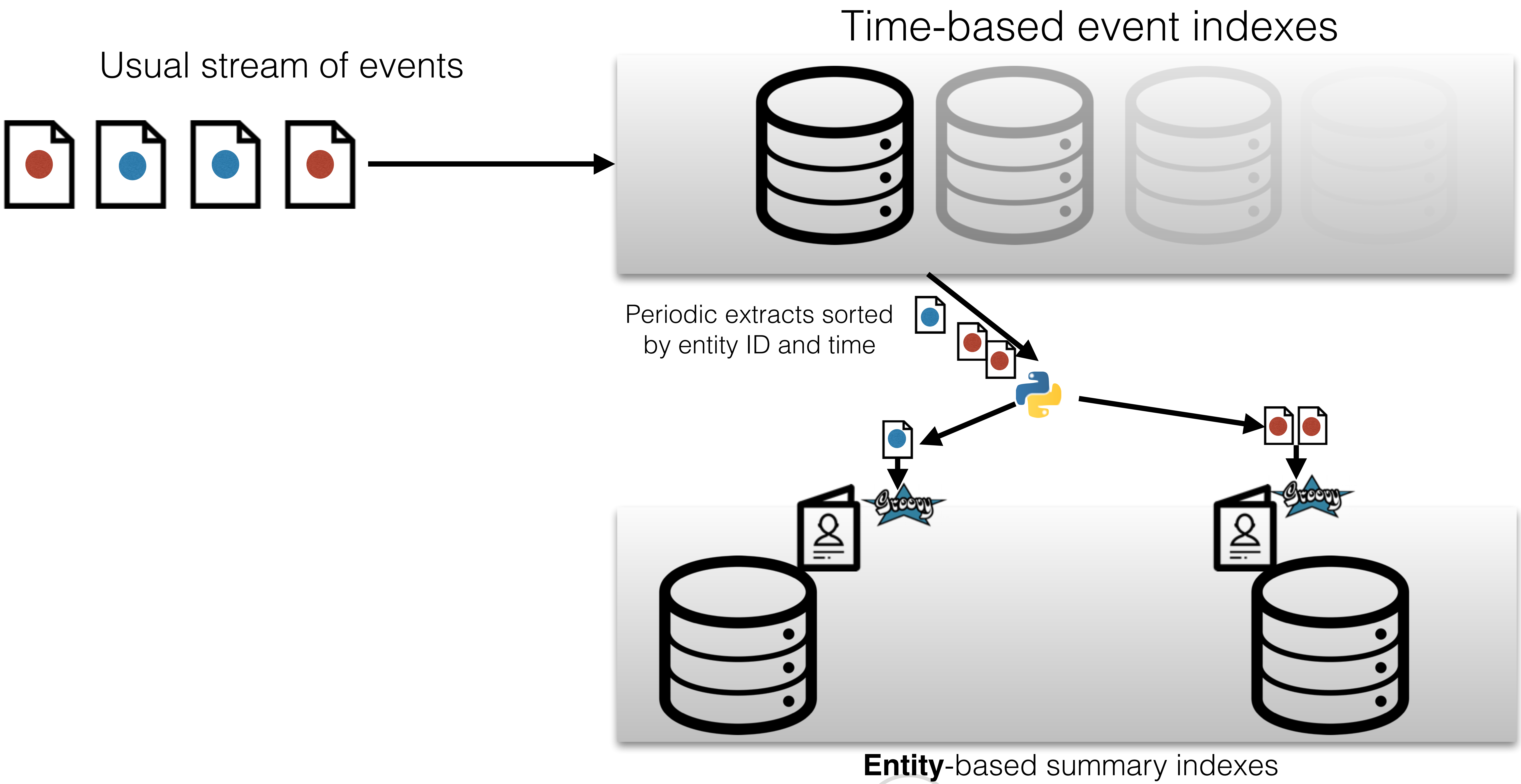
Amazon marketplace reviews

Q: Is this a fraudulent review?


```
{
    "reviewerId": "A3JTCSJWNELJWP",
    "rating": 5,
    "vendorId": "A2YTGD9009QRNWU",
    "reviewText": "Prompt safe delivery of sealed DVD",
    "date": "2006-09-07 18:32",
    "vendorName": "foobardirect"
}
```


A: We can't tell unless we understand people's behaviour over time…

Answer: reorganize content around people

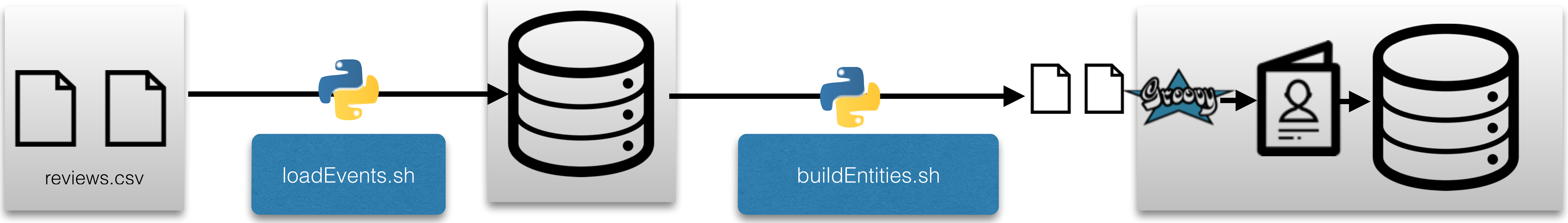


An “entity-centric” model

Play along! Code + data here: bit.ly/entcent

AmazonReviews *(an **event**-centric index)*

Amazon*Reviewers* (an **entity**-centric index)



Review event fields
- rating
- seller
- reviewer
- date

- Drops and creates *reviewers* index.
- Uses Python client to query and scroll list of reviews sorted by reviewerId and time
- Python pushes _update requests to ~400k “Reviewer” documents each containing bundles of their recent reviews using bulk indexing API
- Shard-side Groovy script collapses the multiple reviews into a single reviewer JSON document summarising behaviour

Reviewer entity fields
- positivity
- num sellers reviewed
- last 50 reviews
- profile (“newbie”, “**fanboy**” etc)

Anatomy of an entity indexing groovy script

```
// Extract the doc source to a field
doc = ctx._source;

if("create".equals(ctx.op)){
    //initialize entity state
    doc.profile = "newbie";
    doc.totalNumReviews = 0;
}

// Append the new events into entity summary
for (review in events){
    doc.totalNumReviews++;
    ...
}
// Re-assess reviewer profile
doc.profile = "newbie";
if(doc.totalNumReviews > 5){
    doc.profile = "regular";
    if(doc.positivity > 50){
        if(doc.numVendors == 1){
            doc.profile = "fanboy";
        }
    } else {
        if(doc.numVendors == 1){
            doc.profile = "hater";
        } else {
            doc.profile = "unlucky";
        }
    }
}
```

Load stored state

Initialize if new document

Loop to consolidate latest events

Re-run risk profile logic

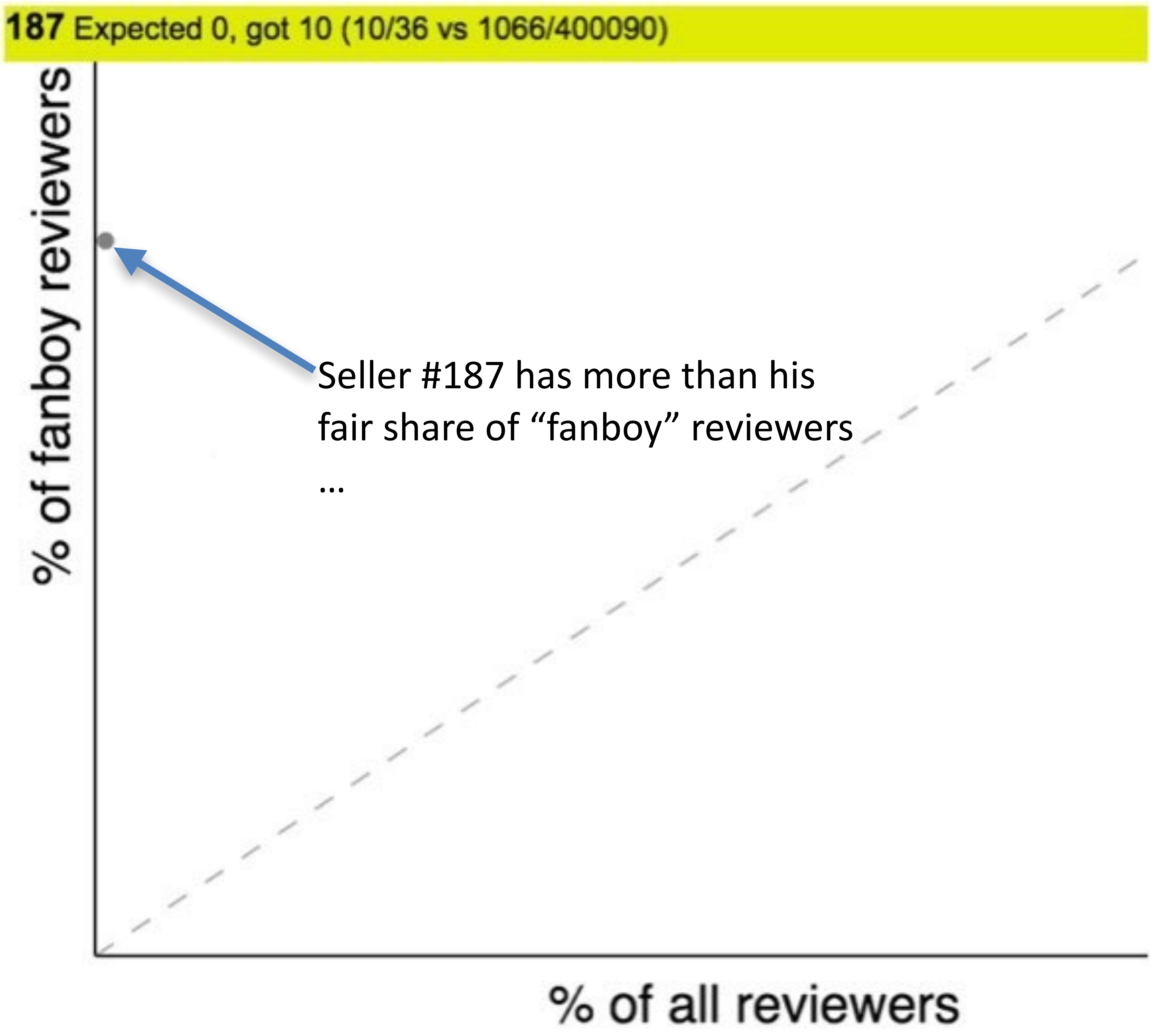
Store the script in `ES_HOME/config/scripts/foo.groovy`

Insight: which sellers have a lot of fanboys?

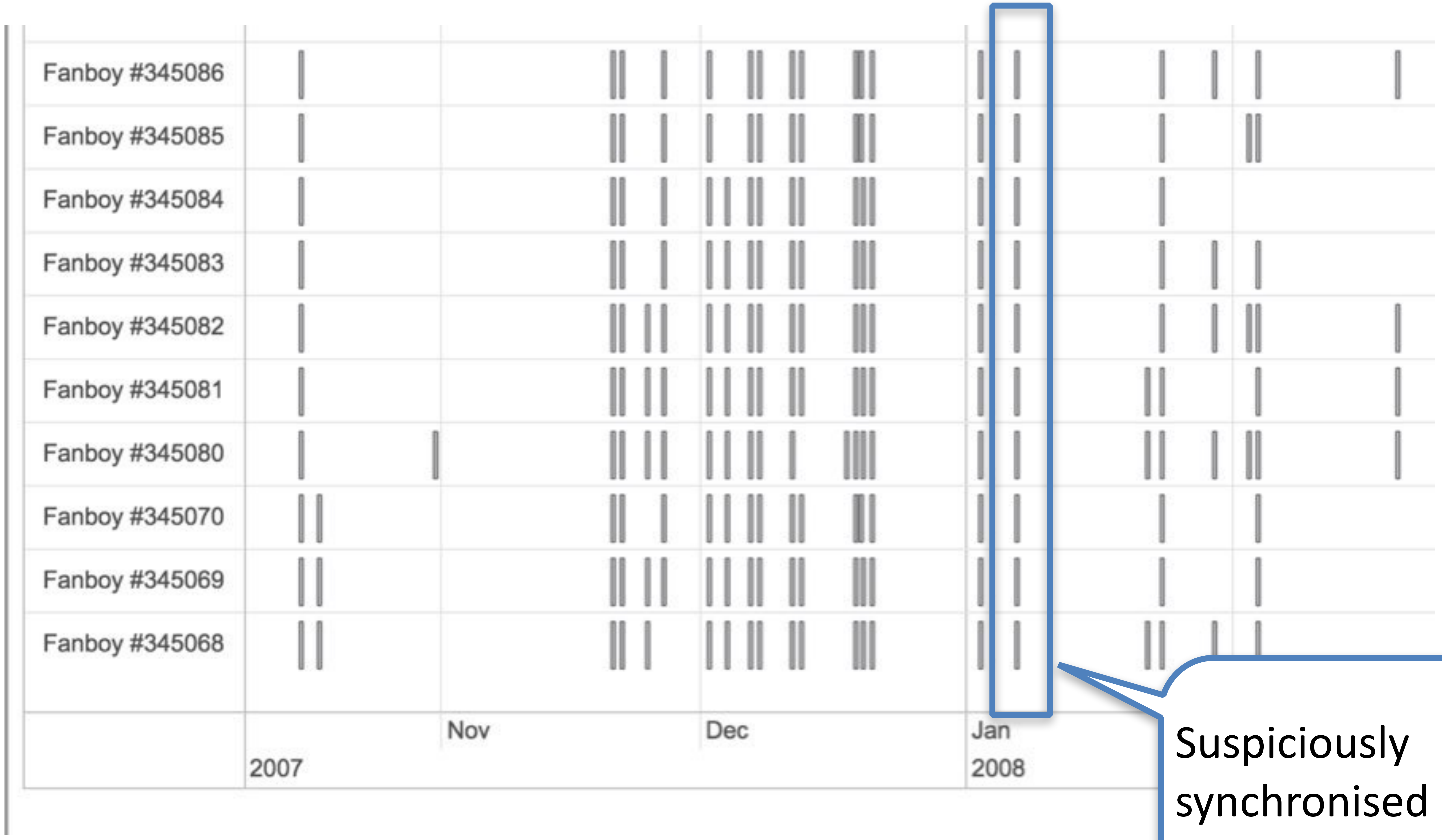
Search the reviewer profiles for all the fanboys

```
{
    "query": {
        "match": {
            "profile": "fanboy"
        }
    },
    "aggs": {
        "interestingSellers": {
            "significant_terms": {
                "field": "recentReviews.vendorId"
            }
        }
    }
}
```

Identify which of the sellers they buy from are unusually represented



Drilling down into seller #187's fanboys



Suspiciously synchronised behaviour

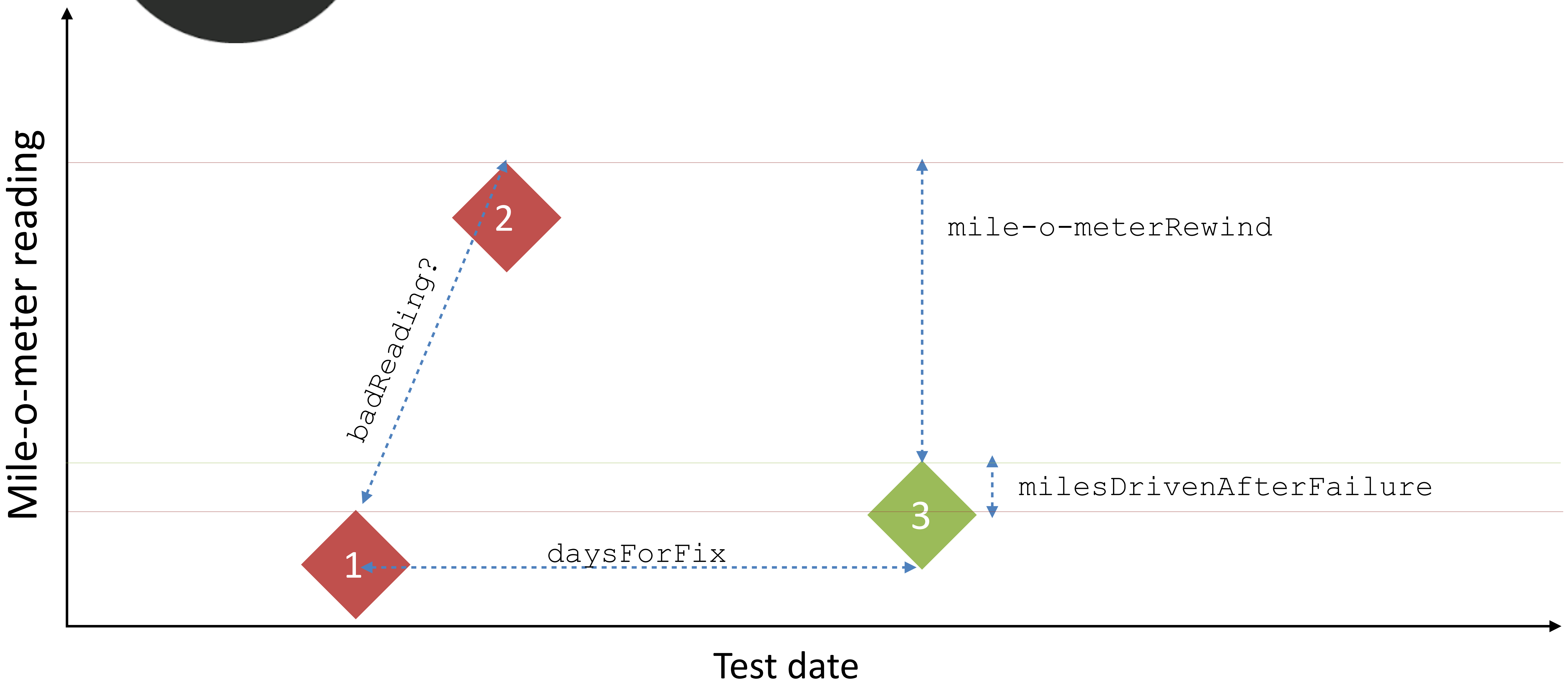
UK car roadworthiness test: raw data


```
{
    "TestClassID": "4",
    "town": [
        "Glasgow"
    ],
    "Colour": "RED",
    "TestResult": "F",
    "VehicleID": "29563",
    "FuelType": "D",
    "Make": "MERCEDES",
    "TestMileage": 80571,
    "CylinderCapacity": "2685",
    "PostcodeArea": "G",
    "location": {
        "lat": "55.869347",
        "lon": "-4.271848"
    },
    "TestDate": "2013-07-31",
    "Model": "CLK270 CDI ELEGANCE A",
    "FirstUseDate": "2005-03-16",
    "TestType": "N",
    "ID": "70605",
    "yearlyMileage": 8952
}
```

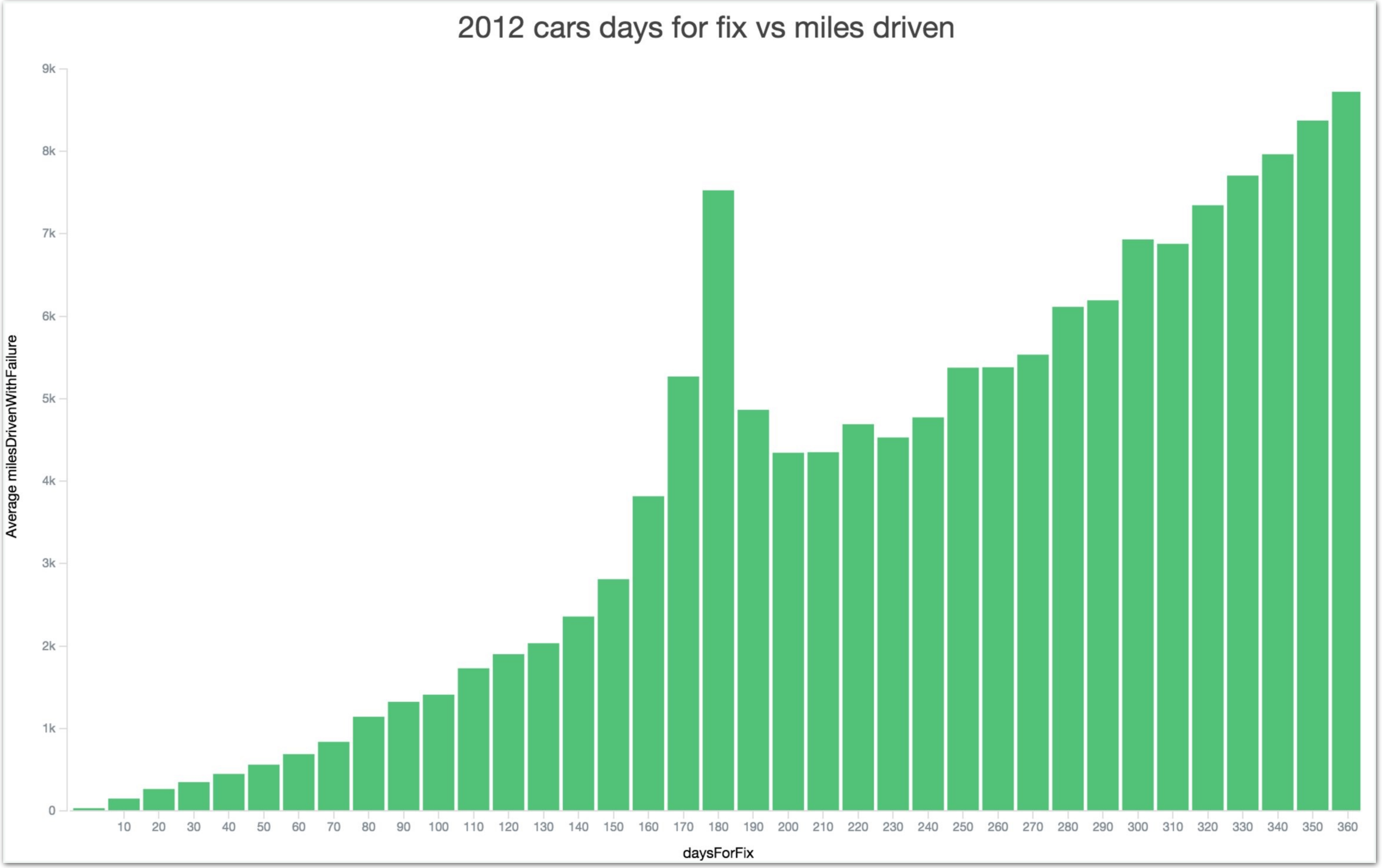

http://data.gov.uk/dataset/anonymised_mot_test

Derived car attributes

Car attributes derived from 3 test result documents



Miles driven vs number of days for fix



Q: Why is there an unexpected peak in milesDrivenWithFailure around the 6-month mark?
A: Taxis

Who drives failed cars?

Table of cars driven long distances after failures

Top 20 unusual terms in MakeModel.raw	Count of documents
VOLKSWAGEN SHARAN S TDI 115 AUTO	247
LONDON TAXIS INT TXII SILVER AUTO	125
LONDON TAXIS INT TX4 SILVER AUTO	107
LONDON TAXIS INT TXII BRONZE AUTO	103
LONDON TAXIS INT TX1 BRONZE AUTO	93
LONDON TAXIS INT TX1 SILVER AUTO	94
LONDON TAXIS INT TX4 BRONZE AUTO	75
FORD GALAXY 16V AUTO	70
LONDON TAXIS INT TX4 GOLD AUTO	45
LONDON TAXIS INT TX1 BRONZE	52

Taxis are the most significant car-makes involved in continuing to drive long distances after MOT failures

In summary

- Averages are misleading - see percentiles
- Consider the fuzziness of the set you analyse
- Significance != popularity
- Consider memory use (breadth_first)
- Re-organise log data into entity-centric indexes for deeper insights into user behaviours

Questions?

@elasticmark