Big Data Analytics: What is Big Data?

H. Andrew Schwartz

CSE545
Spring 2020

Big Data, what is it?

traditional
computer science

data that will not fit
in main memory.

Big Data, what is it?

traditional
computer science

data that will not fit in main memory.

SSD Sequential Read:
~500 MB/s

For example...

busy web server access logs

graph of the entire Web

all of Wikipedia

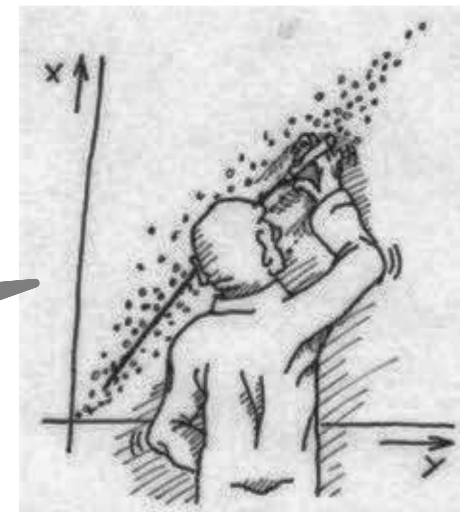
daily satellite imagery over a year

Big Data, what is it?

traditional
computer science

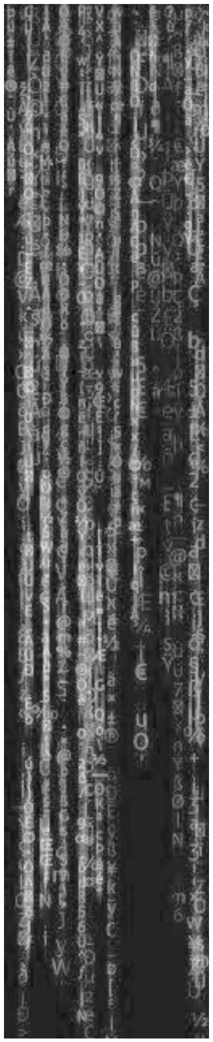
data that will not fit in main memory.

data with a *large* number of observations and/or features.



statistics

Big Data, what is it?

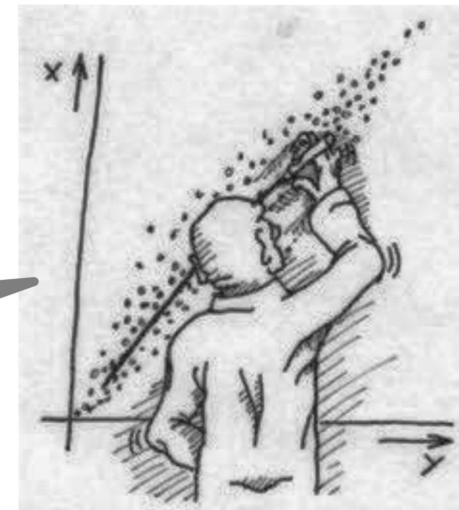


Tall data:

edge list of a large graph

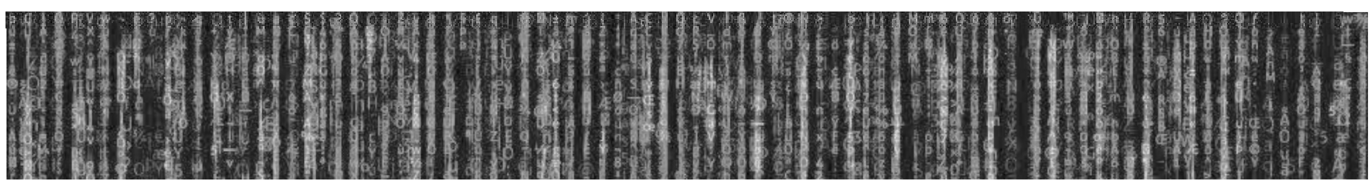
rgb values per pixel location in large images

data with a *large* number of observations and/or features.



statistics

Wide data: *mobile app usage statistics of 100 people*

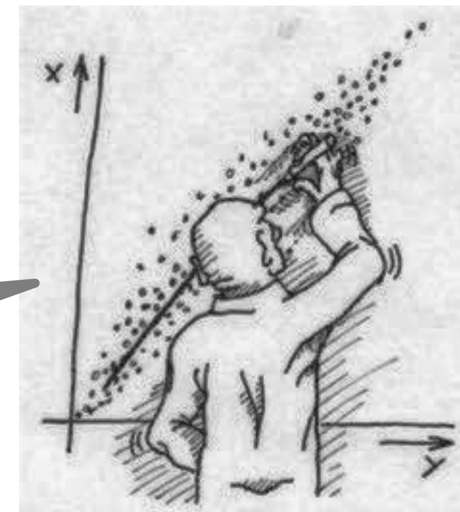


Big Data, what is it?

traditional
computer science

data that will not fit
in main memory.

data with a *large*
number of observations
and/or features.

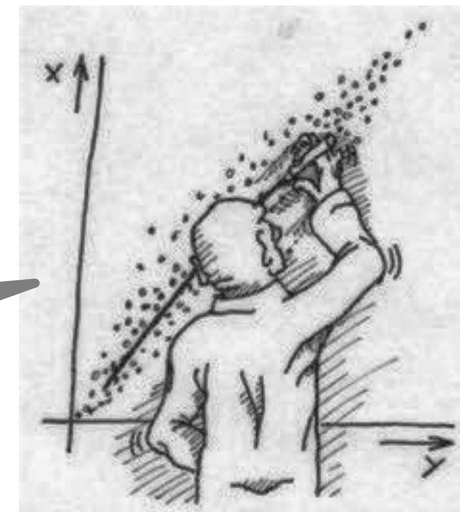


statistics

Big Data, what is it?

traditional
computer science

data that will not fit in main memory.



statistics

data with a *large* number of observations and/or features.

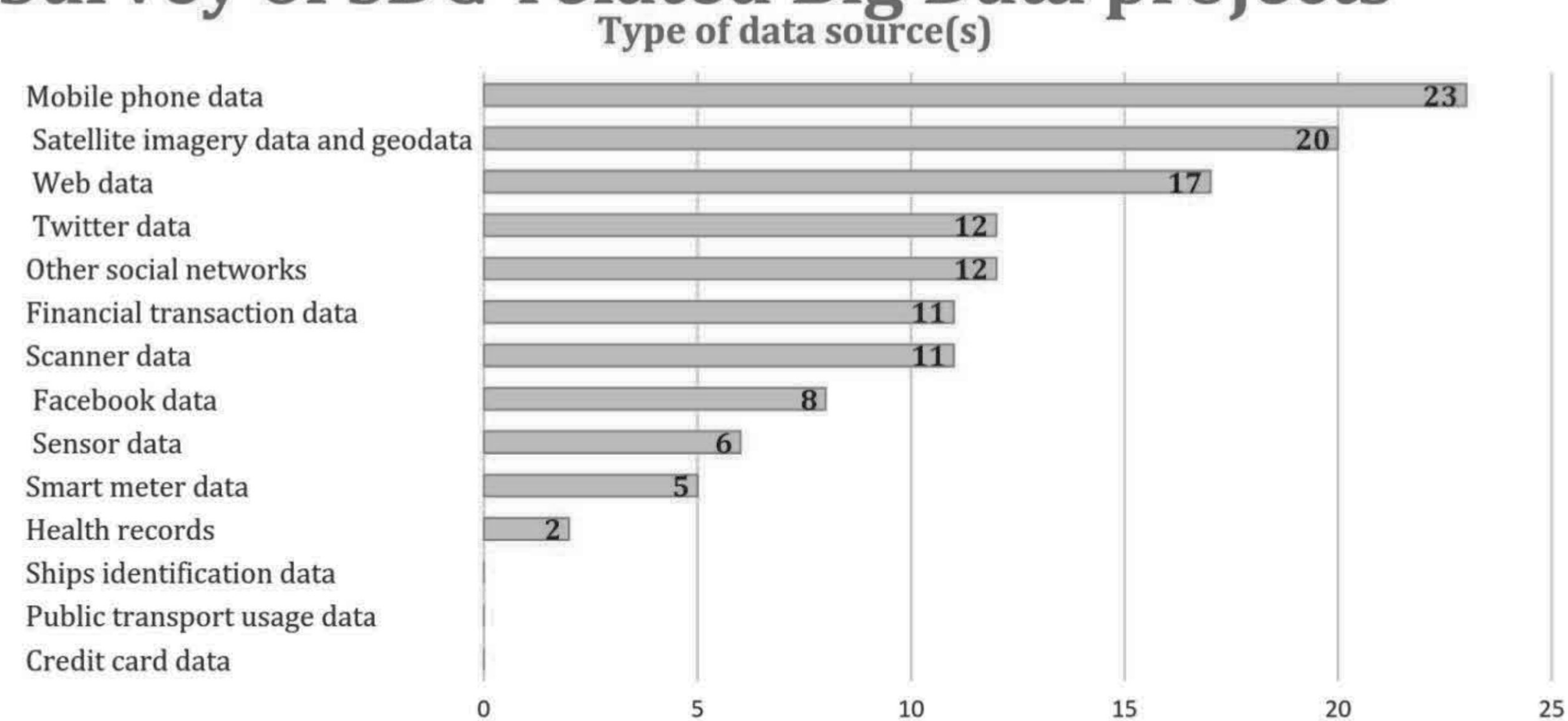
other fields

non-traditional sample size (i.e. > 100 subjects); can't analyze in stats tools (Excel).

Big Data, what is it? *Government View*

THE WORLD BANK
IBRD • IDA | WORLD BANK GROUP (2016)

1. Survey of SDG-related Big Data projects

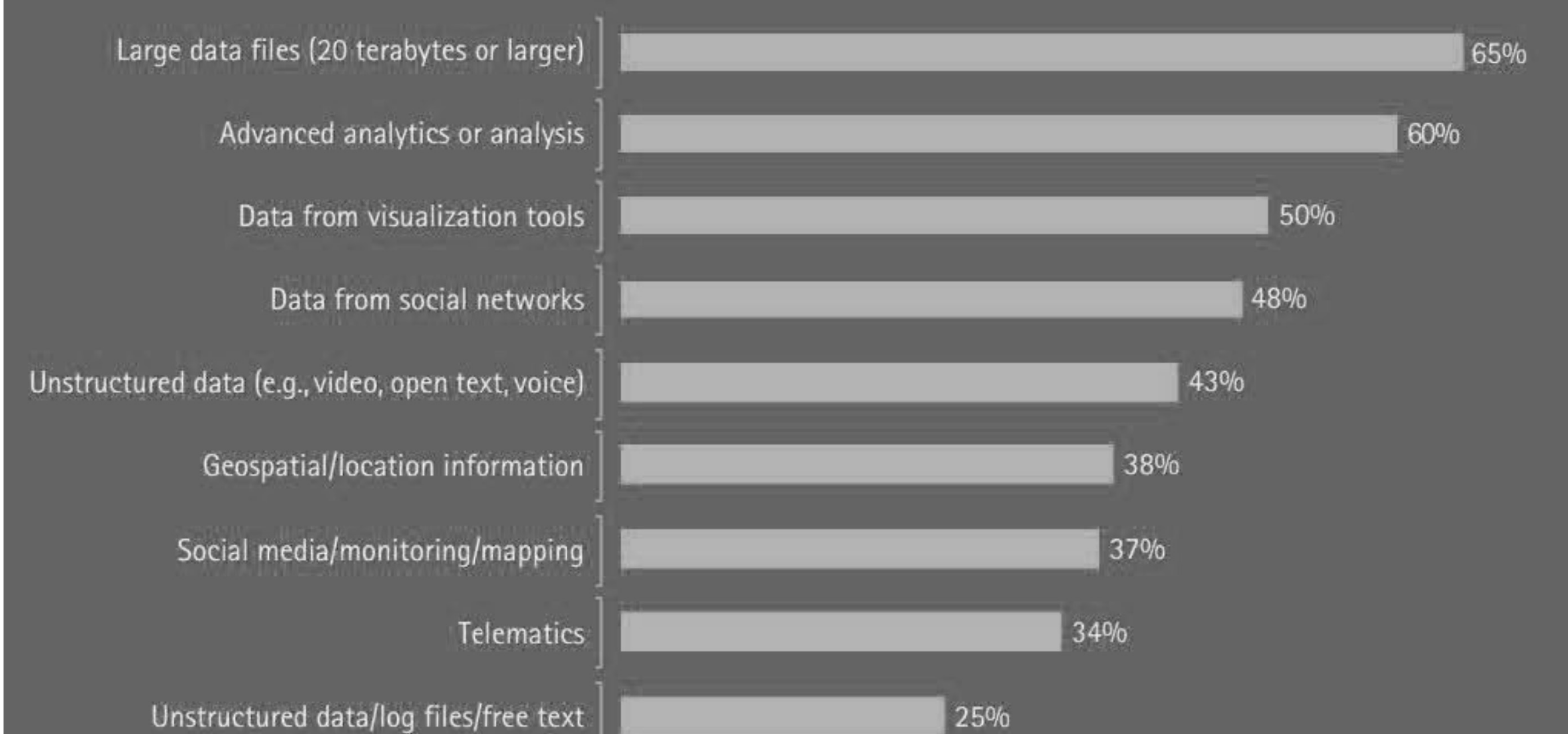


- Mobile (23), Satellite imagery (20) and social media (12+12+8) are the most prominent sources

Big Data, what is it? *Industry View*

Figure 2: Sources of big data

Which of the following do you consider part of big data (regardless of whether your company uses each)?

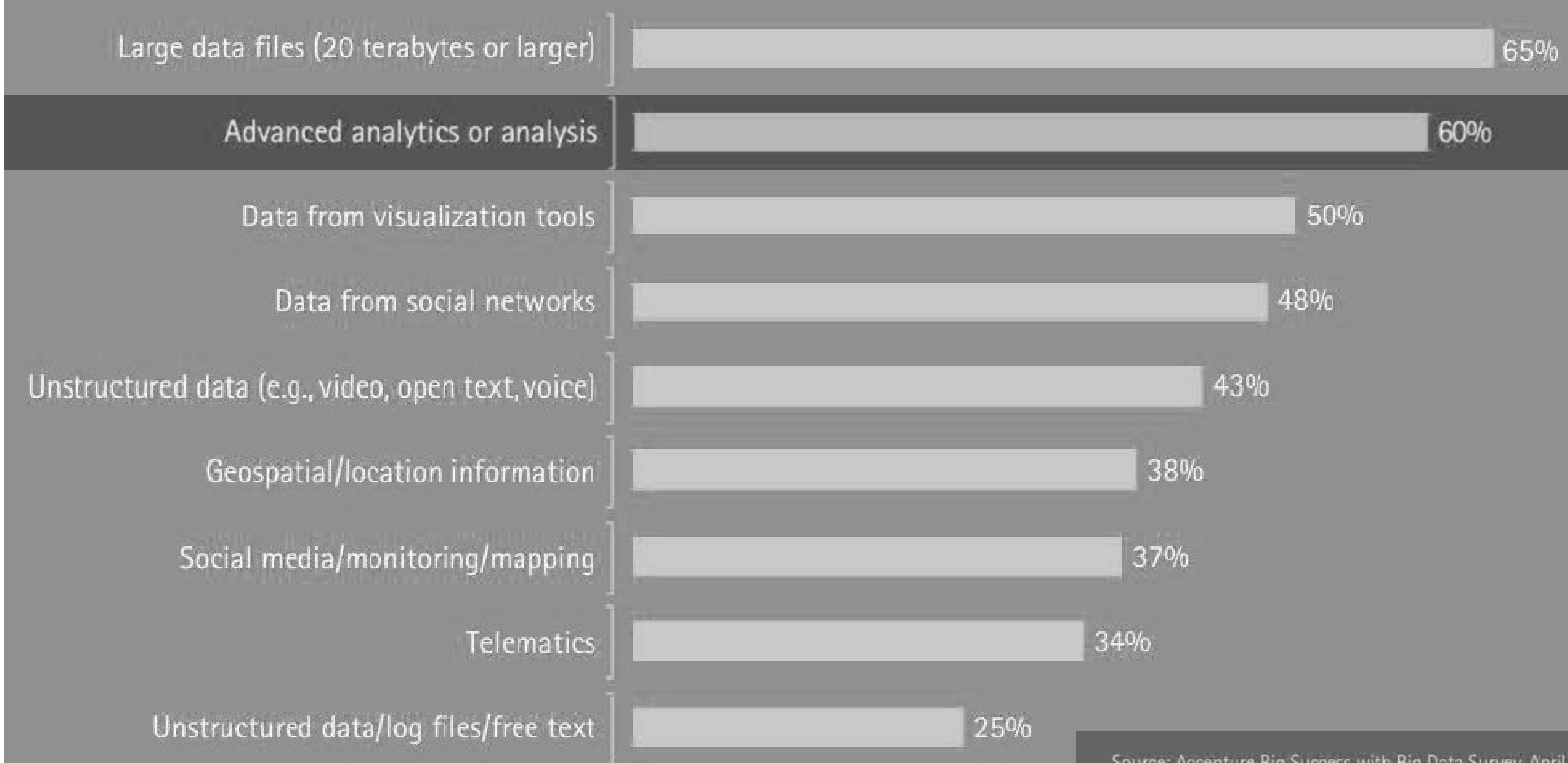


Source: Accenture Big Success with Big Data Survey, April 2014

Big Data, what is it? *Industry View*

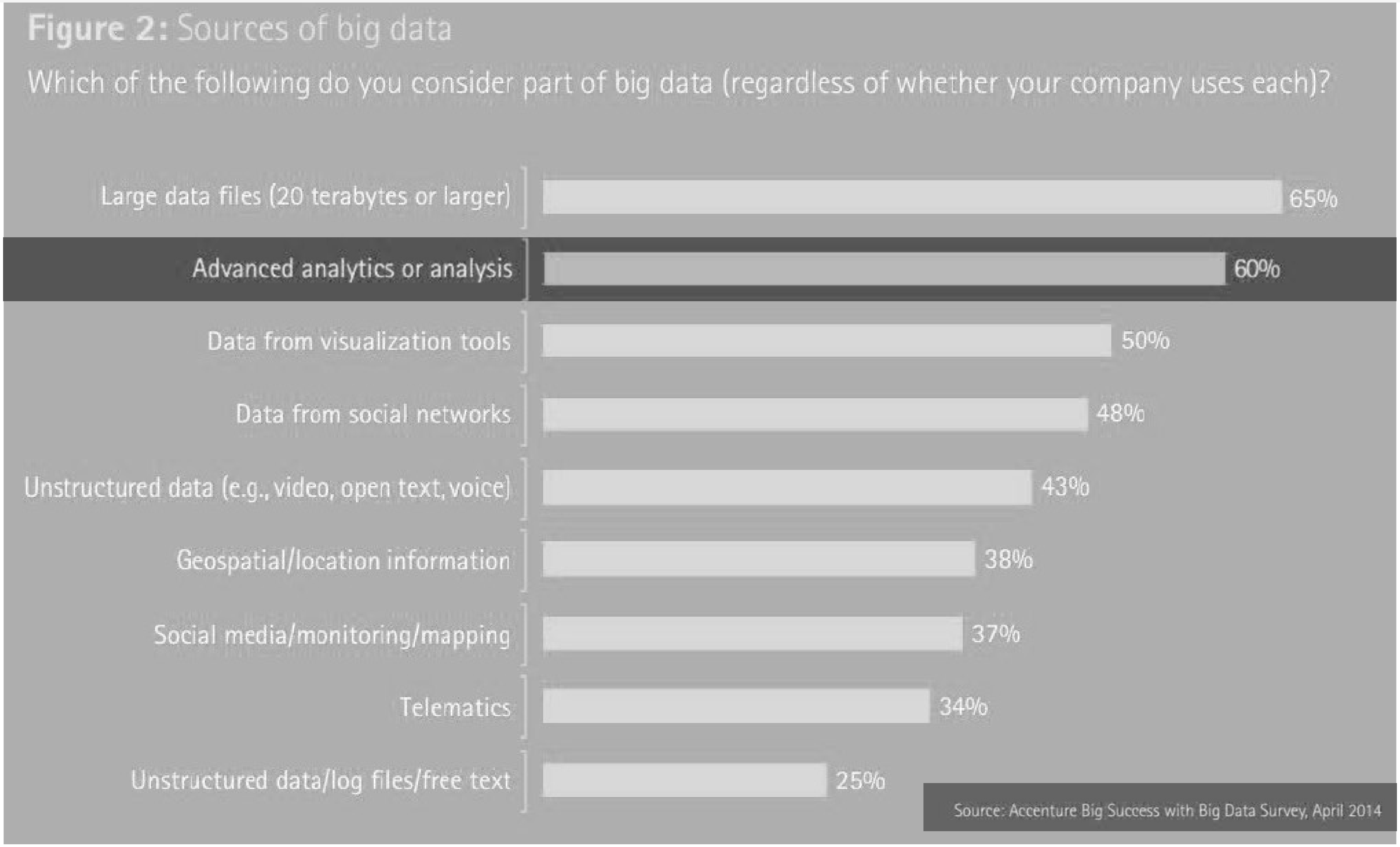
Figure 2: Sources of big data

Which of the following do you consider part of big data (regardless of whether your company uses each)?



Source: Accenture Big Success with Big Data Survey, April 2014

Big Data, a type of analytics



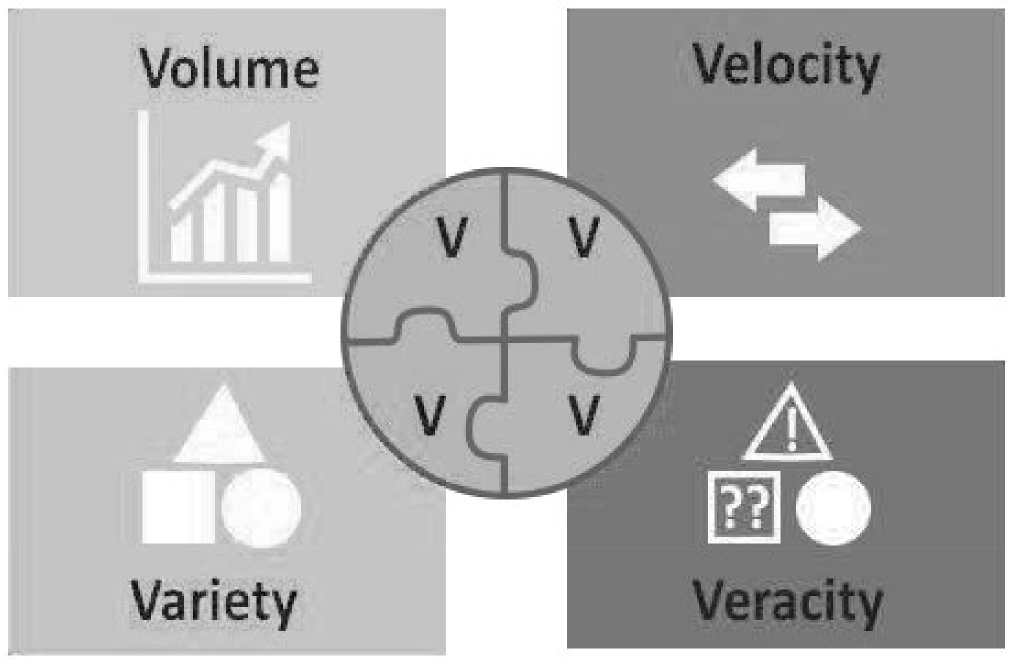
Big Data, a type of analytics

Analyses which can handle the “3 Vs”:

1. Volume - large quantity
2. Volocity - arriving quickly
3. Variety - [un]structed, multi-modal

Big Data, a type of analytics

Analyses which can handle the "~~3~~4 Vs":

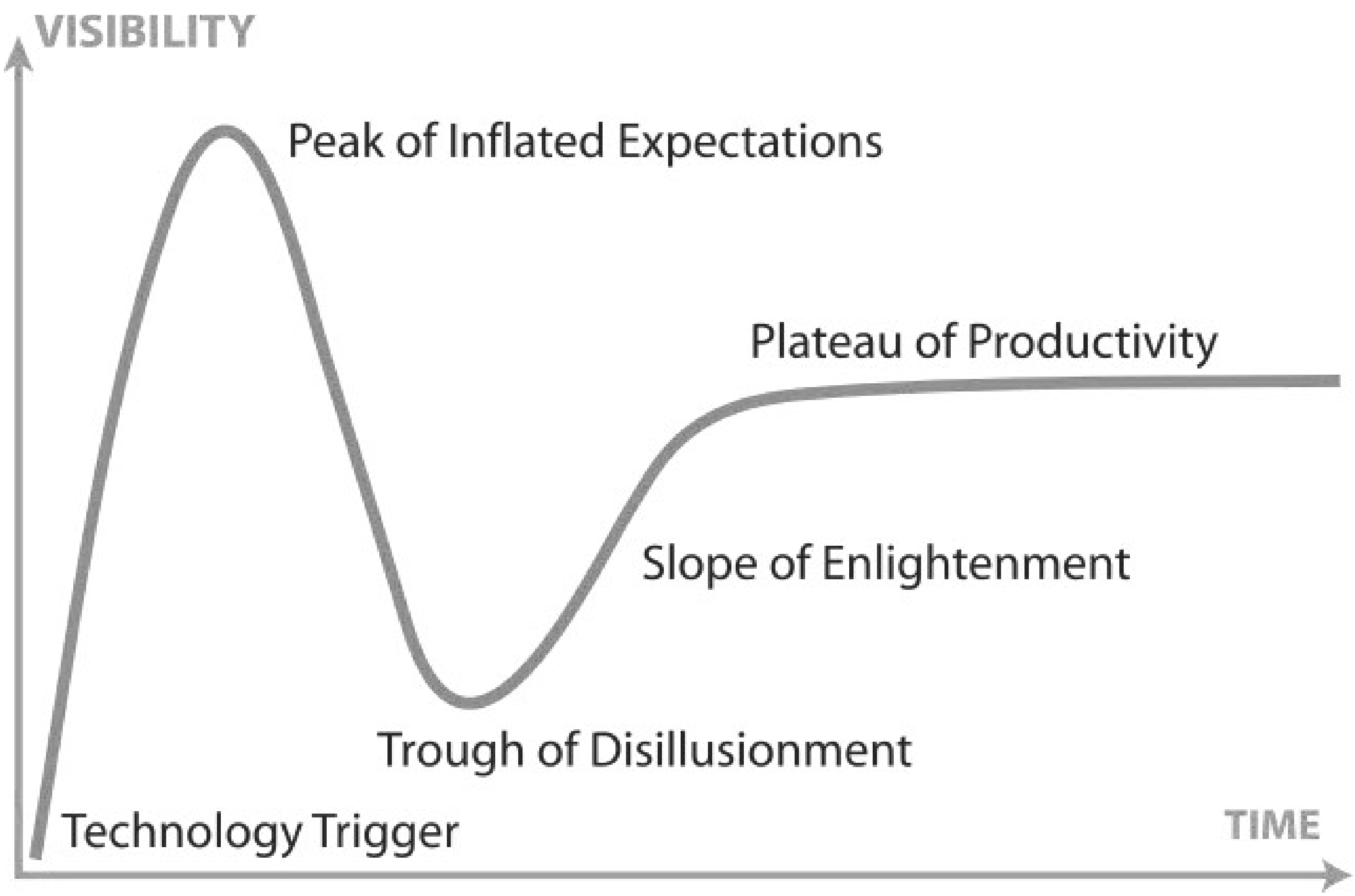


Big Data, a type of analytics

Big Data, a type of analytics

Big Data, a type of analytics

Big Data, a buzz word?



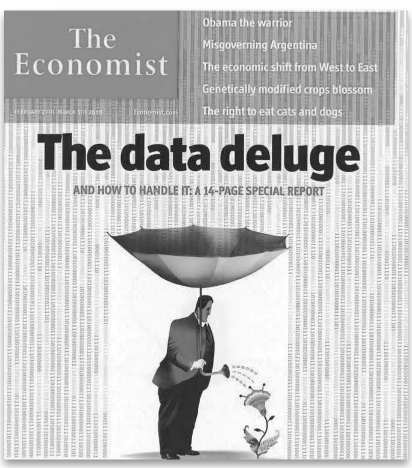
(Gartner Hype Cycle)

Big Data, a buzz word?

2008

2011

2012

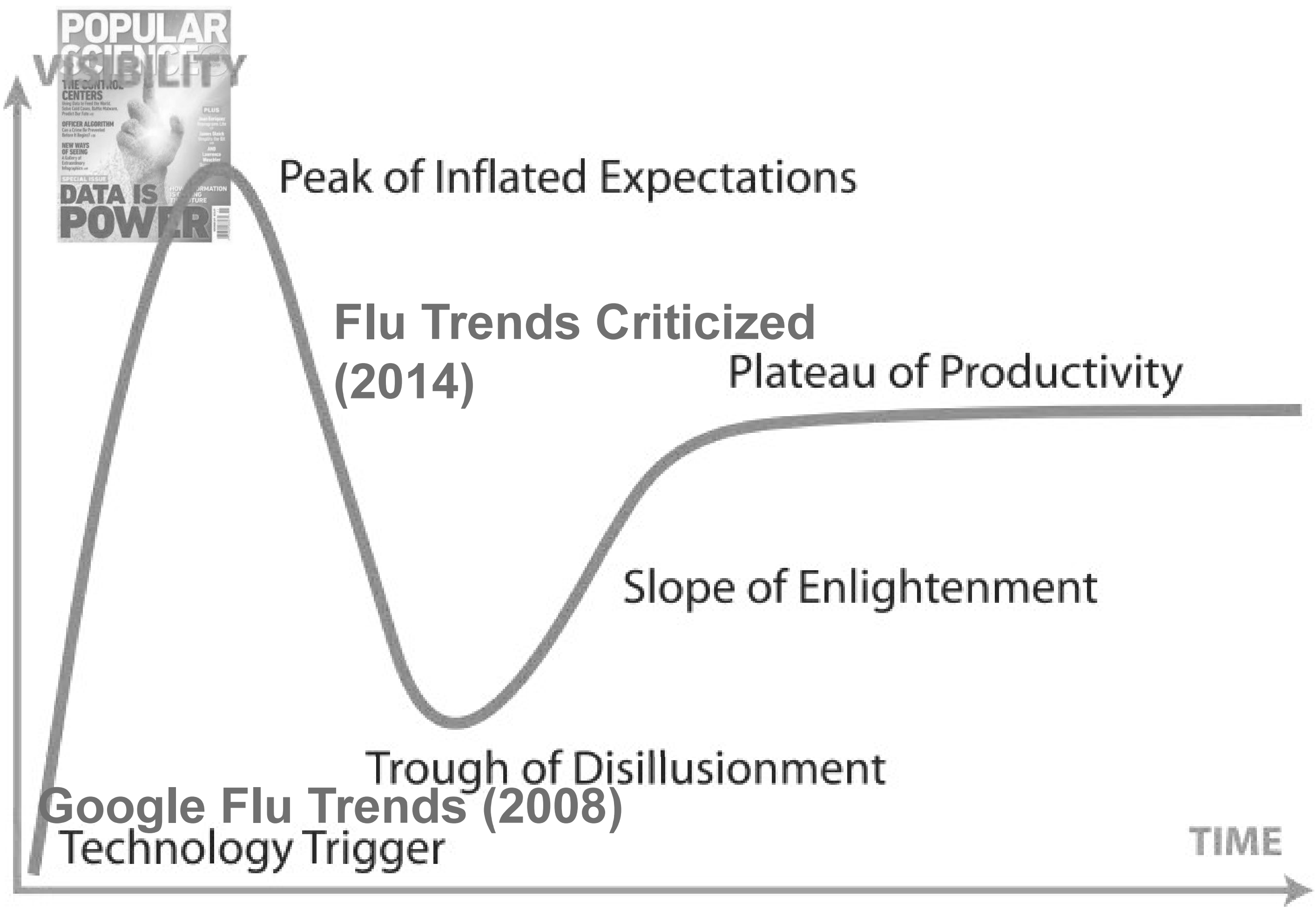


2010

2011

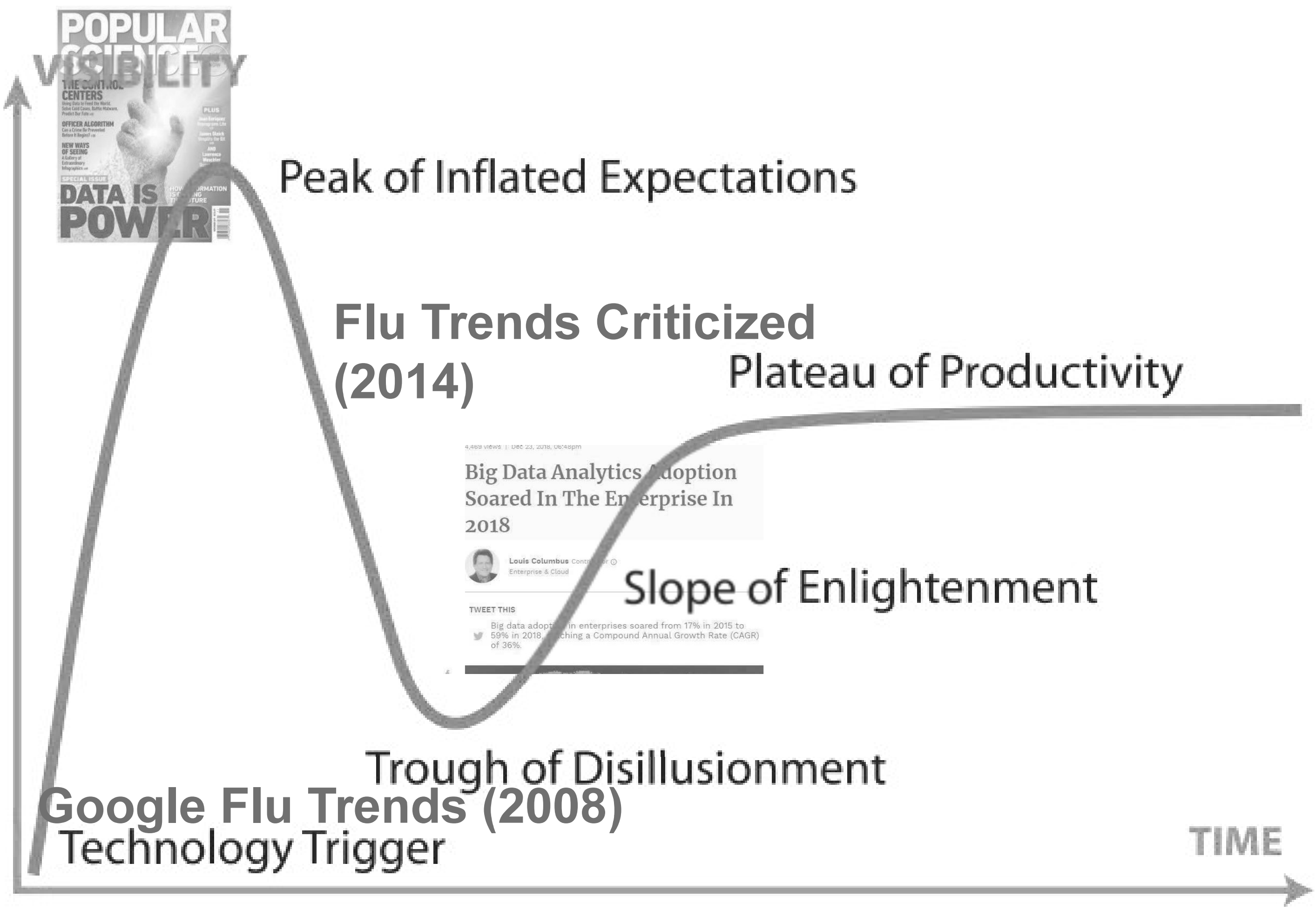
2018

Big Data, a buzz word?



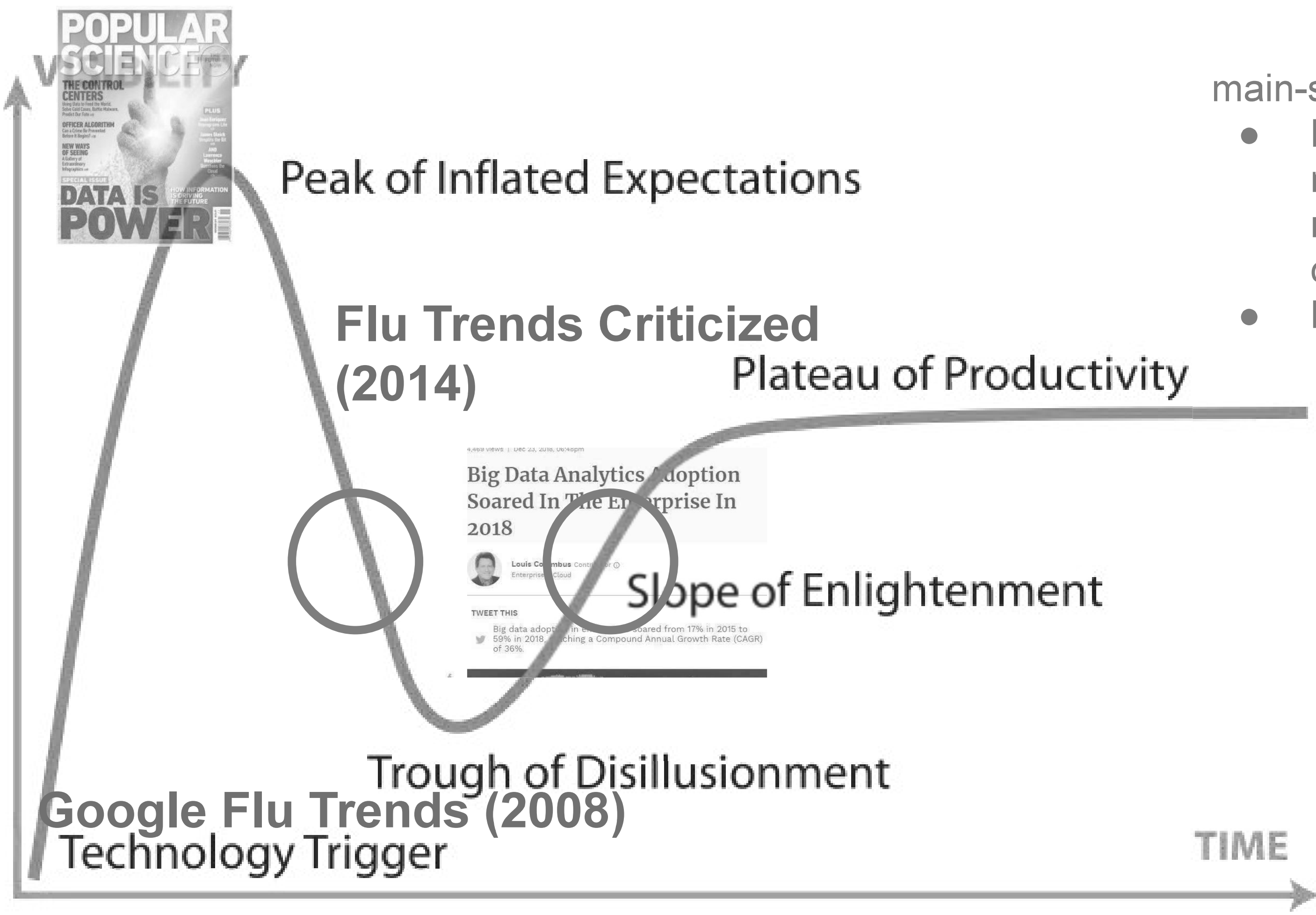
(Gartner Hype Cycle)

Big Data, a buzz word?



(Gartner Hype Cycle)

Big Data, a buzz word?



main-stream study being established

- Realization of what subfields are really doing “big data” (i.e. data mining, ML, Statistics, computational social sciences).
- Best practices being established.

(Gartner Hype Cycle)

Big Data, in demand?

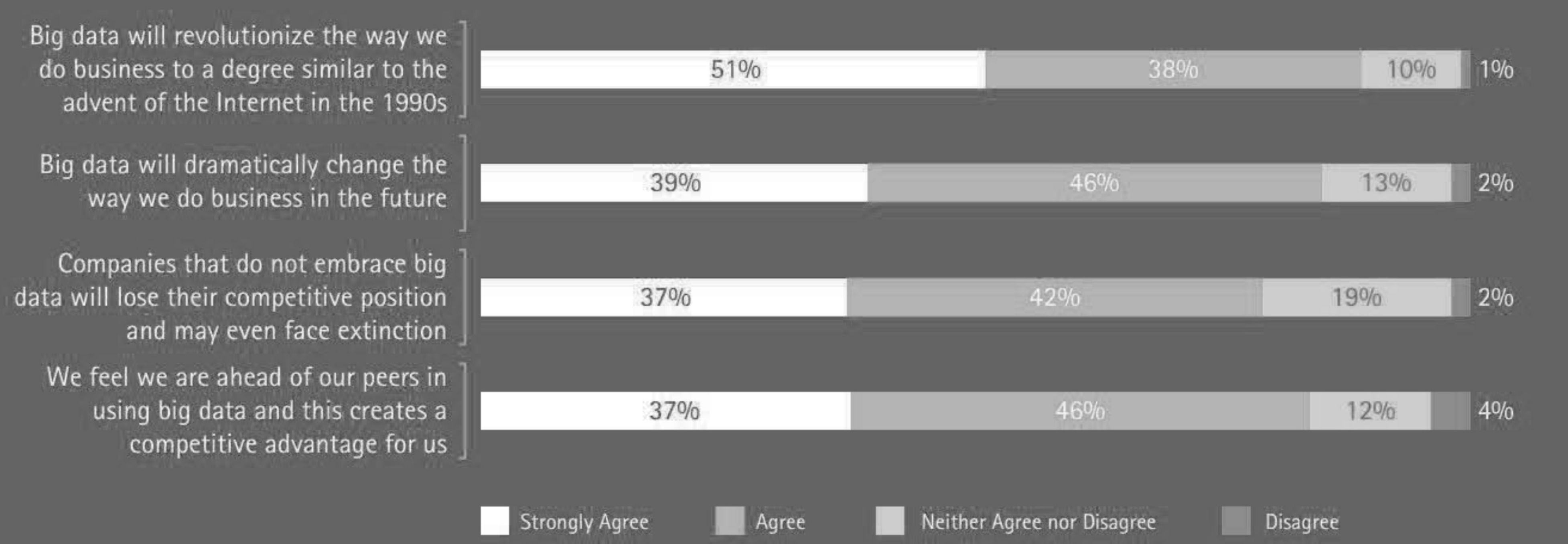
Figure 3: Main challenges with big data projects

What are the main challenges to implementing big data in your company?

Source: Accenture Big Success with Big Data Survey, April 2014

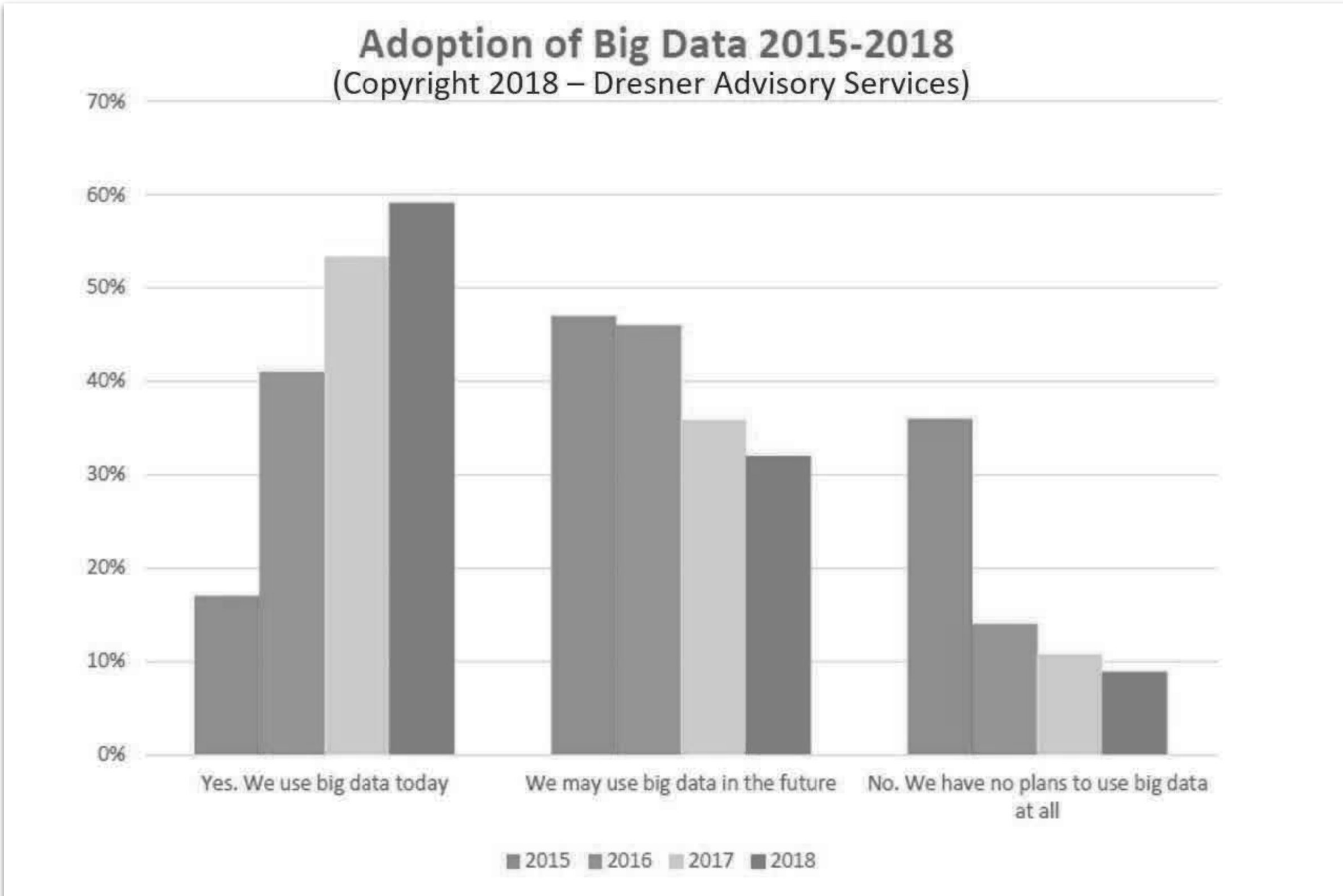
Big Data, in demand?

Figure 6: Big data's competitive significance



Source: Accenture Big Success with Big Data Survey, April 2014

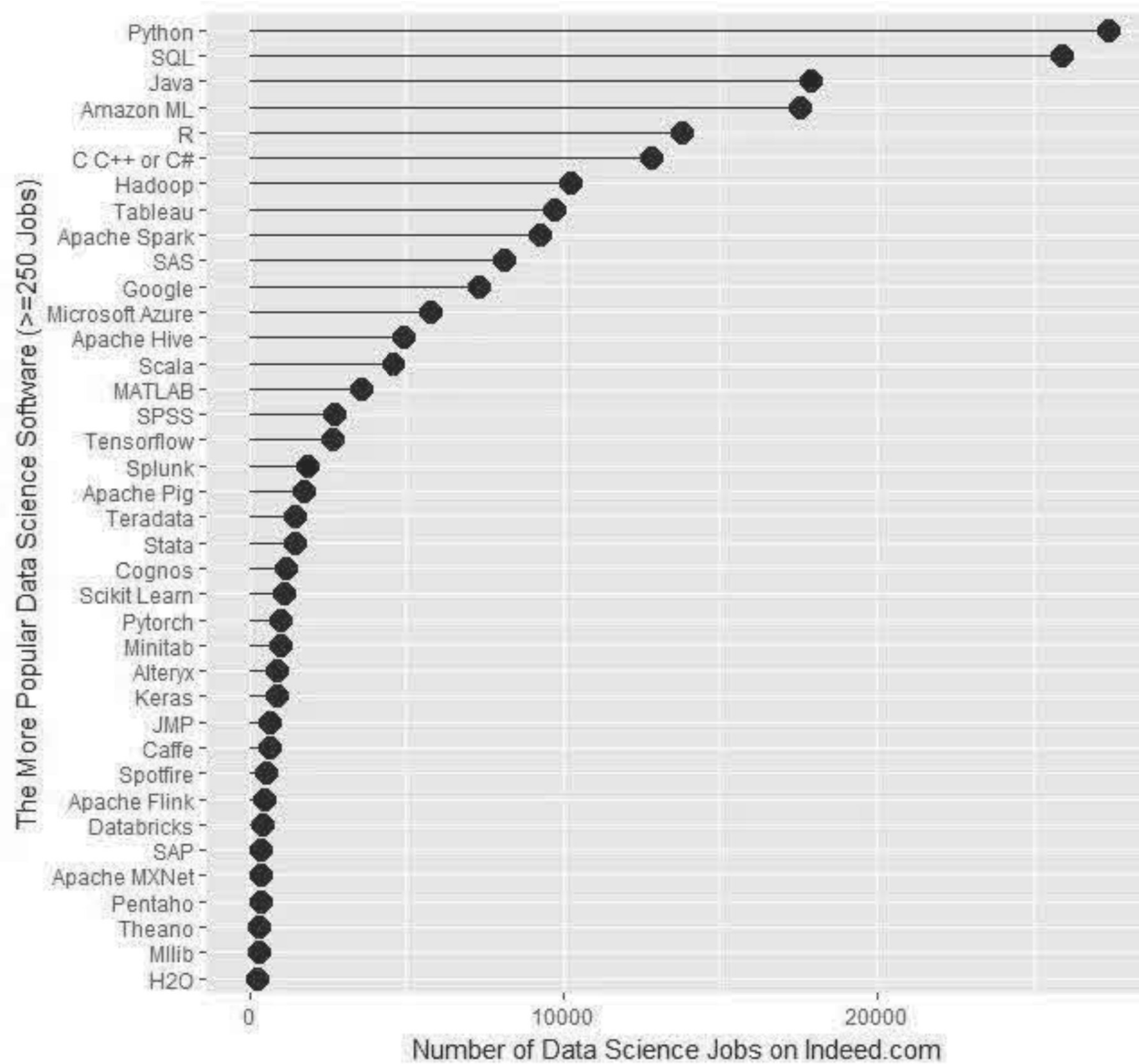
Big Data, in demand?

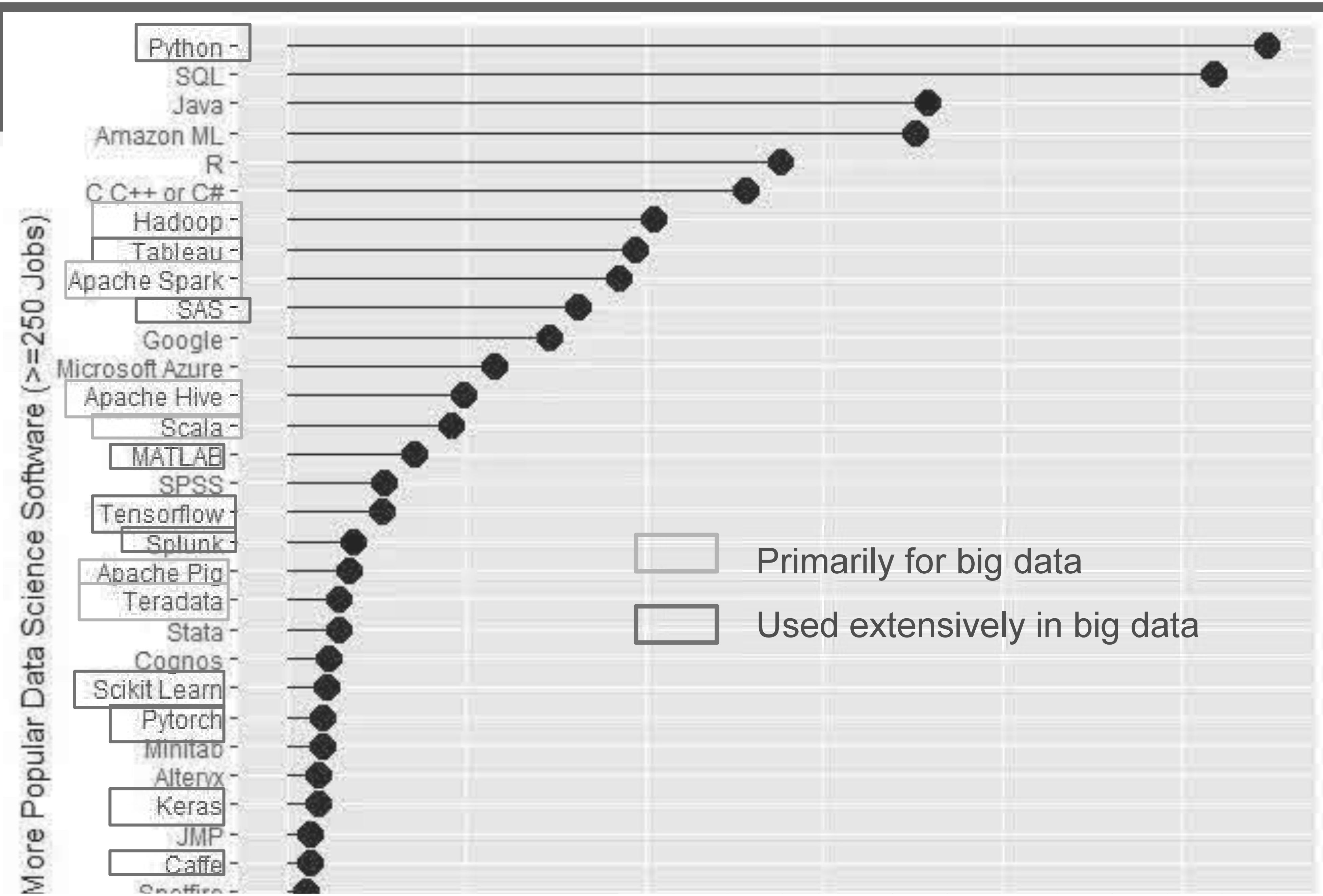


https://www.forbes.com/sites/louiscolumbus/2018/12/23/big-data-analytics-adoption-soared-in-the-enterprise-in-2018/

Big Data, in demand?

By the requirements
in job ads.
(Muenchen,2019)




More Popular Data Science Software (>=250 Jobs)
Python
SQL
Java
Amazon ML
R
C C++ or C#
Hadoop
Tableau
Apache Spark
SAS
Google
Microsoft Azure
Apache Hive
Scala
MATLAB
SPSS
Tensorflow
Splunk
Apache Pig
Teradata
Stata
Cognos
Scikit Learn
Pytorch
Minitab
Alteryx
Keras
JMP
Caffe
Primarily for big data
Used extensively in big data

Big Data, What is it?

Short Answer:

Big Data ≈ Data Mining ≈ Predictive Analytics ≈ Data Science

(Leskovec et al., 2014)

Big Data, What is it?

Short Answer:

Big Data ≈ Data Mining ≈ Predictive Analytics ≈ Data Science

(Leskovec et al., 2014)

CSE545 focuses on:

How to analyze data that is mostly too large for main memory.	Analyses only possible with a *large* number of observations or features.

Big Data, What is it?

Goal: Generalizations
A ***model*** or ***summarization*** of the data.

How to analyze data that is mostly too large for main memory.

Analyses only possible with a *large* number of observations or features.

Big Data, What is it?

Goal: Generalizations
A *model* or *summarization* of the data.

E.g.

- Google's PageRank: *summarizes* web pages by a single number.
- Twitter financial market predictions: *Models* the stock market according to shifts in sentiment in Twitter.
- Distinguish tissue type in medical images: *Summarizes* millions of pixels into clusters.
- Mental health diagnosis in social media: *Models* presence of diagnosis as a distribution (a summary) of linguistic patterns.
- Frequent co-occurring purchases: *Summarize* billions of purchases as items that frequently are bought together.

Big Data, What is it?

Goal: Generalizations
A *model* or *summarization* of the data.

1. Descriptive analytics
   Describe (*generalizes*) the data itself

2. Predictive analytics
   Create something *generalizeable* to new data

Big Data Analytics, The Class

Core Data Science Courses

CSE 519: Data Science Fundamentals

CSE 544: Prob/Stat for Data Scientists

CSE 545: Big Data Analytics

CSE 512: Machine Learning

CSE 537: Artificial Intelligence

CSE 548: Analysis of Algorithms

CSE 564: Visualization

Applications of Data Science

CSE 527:
Computer Vision

CSE 538:
Natural Language Processing

CSE 549:
Computational Biology

...

Big Data Analytics, The Class

Core Data Science Courses

CSE 519: Data Science Fundamentals

CSE 544: Prob/Stat for Data Scientists

CSE 545: Big Data Analytics

CSE 512: Machine Learning

CSE 537: Artificial Intelligence

CSE 548: Analysis of Algorithms

CSE 564: Visualization

Applications of Data Science

CSE 527:
Computer Vision

CSE 538:
Natural Language Processing

CSE 549:
Computational Biology

...

Key Distinction:
Focus on scalability and algorithms / analyses not possible without large data.

Big Data Analytics, The Class

Goal: Generalizations
A *model* or *summarization* of the data.

Data Frameworks

Algorithms and Analyses

Big Data Analytics, The Class

Goal: Generalizations
A *model* or *summarization* of the data.

Data Frameworks

Algorithms and Analyses

Hadoop File System

Spark

Streaming

MapReduce

Tensorflow

Big Data Analytics, The Class

Goal: Generalizations
A *model* or *summarization* of the data.

Data Frameworks

- Hadoop File System
- Spark
- Streaming
- MapReduce
- Tensorflow

Algorithms and Analyses

- Similarity Search
- Linear Modeling
- Recommendation Systems
- Graph Analysis
- Deep Learning

Big Data Analytics, The Class

http://www3.cs.stonybrook.edu/~has/CSE545/

Preliminaries

Ideas and methods that will repeatedly appear:

- Bonferroni's Principle
- Normalization (TF.IDF)
- Power Laws
- Hash functions
- IO Bounded (Secondary Storage)
- Unstructured Data

- *Parallelism*
- *Functional Programming*

Statistical Limits. Goal: **Generalization**

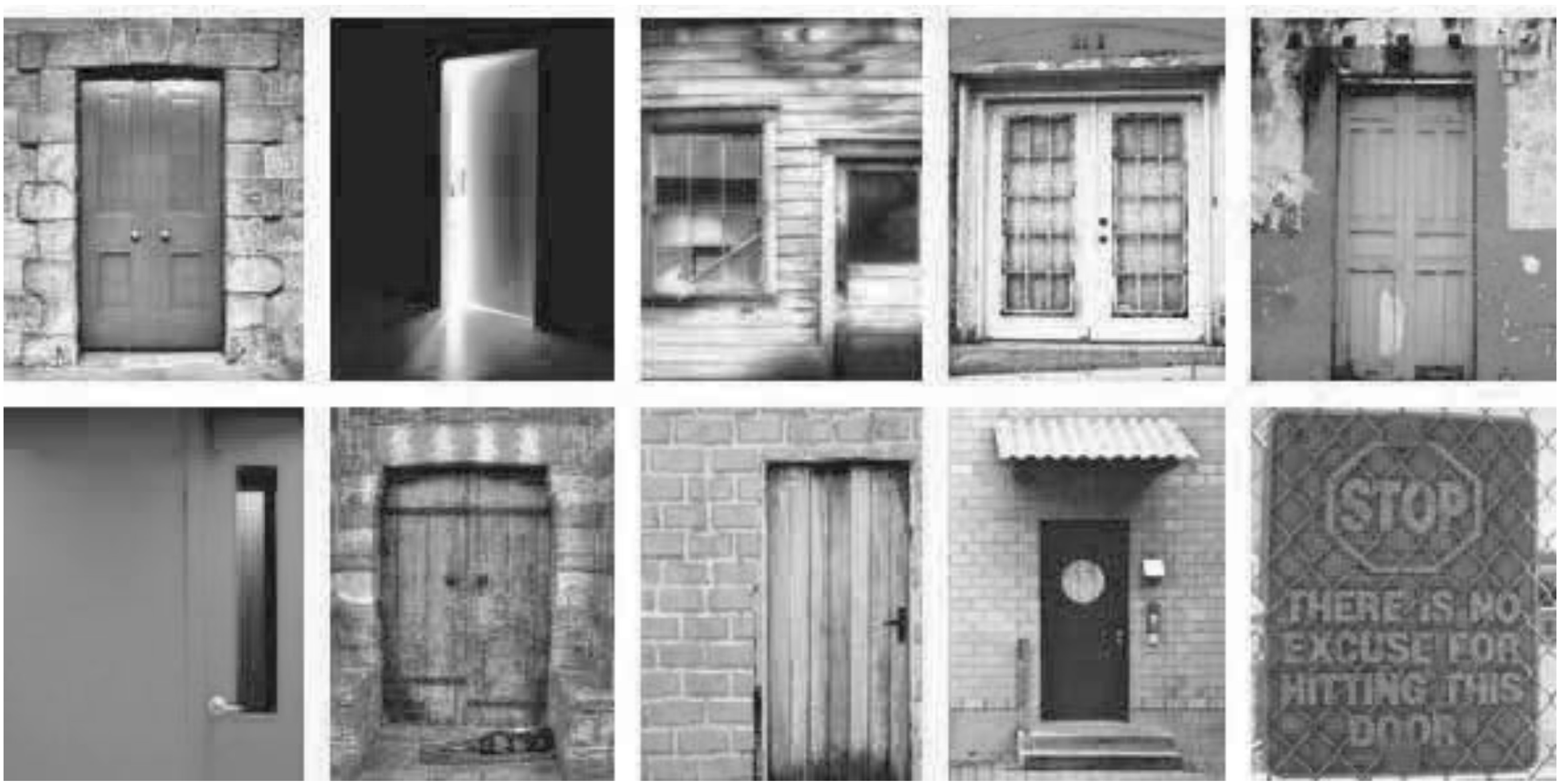
Bonferroni's Principle

A to consider goal of generalization:

Find events that didn’t just happen *by chance*.

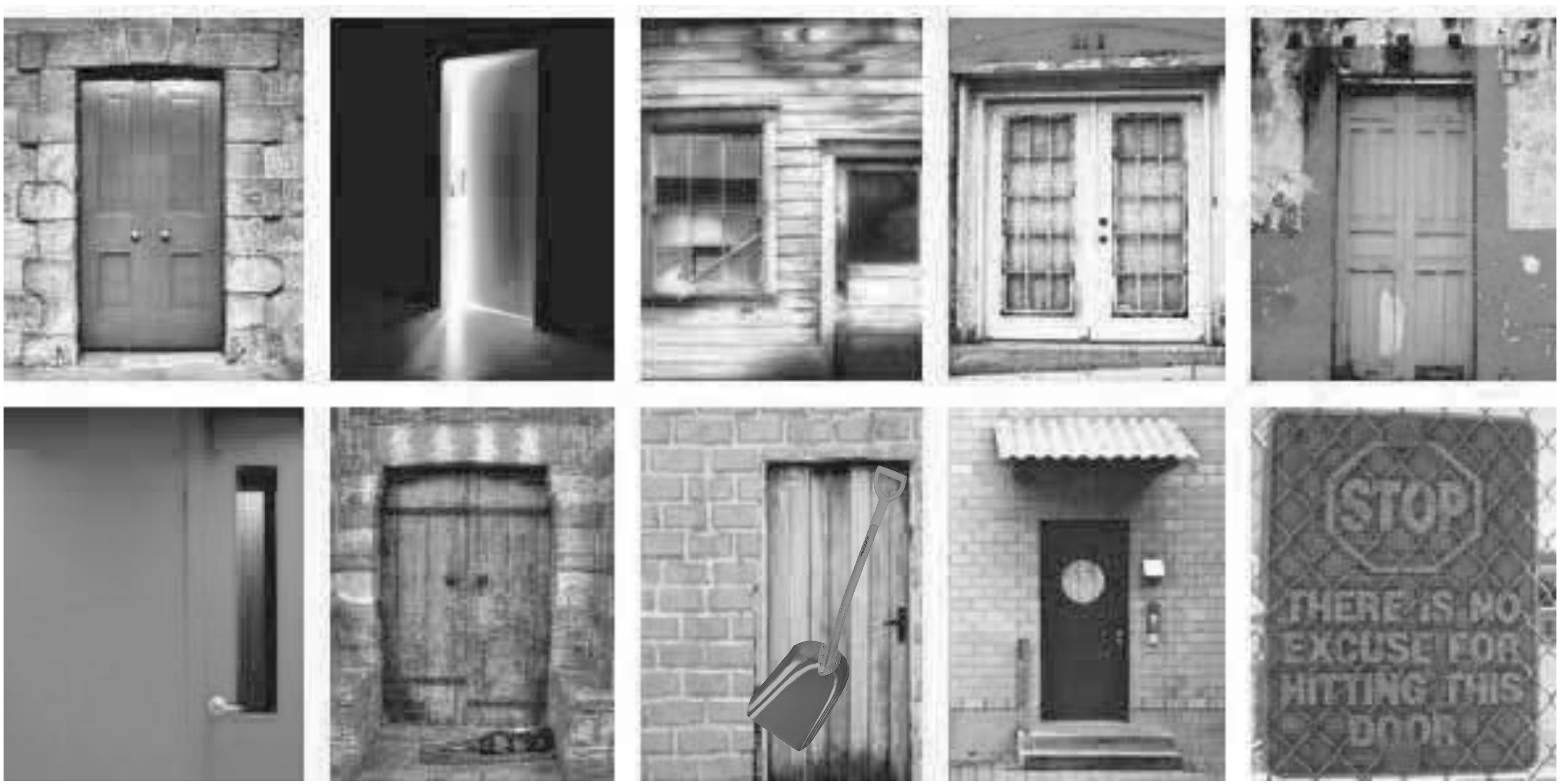
Statistical Limits.

Bonferroni's Principle; an example:



Statistical Limits.

Bonferroni's Principle; an example:



Statistical Limits.

Bonferroni's Principle

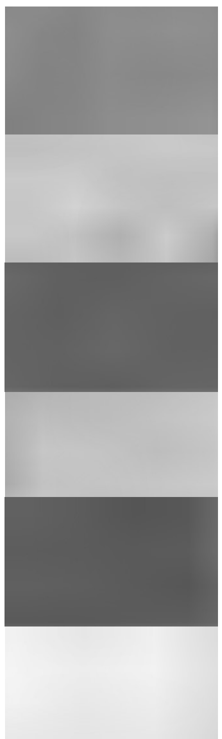
Goal: **Generalization (i.e. not by chance)**

Bonforroni's Principle; Task Example

snazzyphones.com wants to know which case to eliminate.

6 total cases:

Red
Green
Blue
Teal
Purple
Yellow



Bonforroni's Principle; Task Example

snazzyphones.com wants to know which case to eliminate.

6 total cases:

first day, 17 sales:

What is the data telling you?

Bonforroni's Principle; Task Example

snazzyphones.com wants to know which case to eliminate.

6 total cases:

first day, 17 sales:

What is the data telling you?

The blue isn't selling?

Statistical Limits. Goal: **Generalization**

Bonferroni's Principle

Roughly, calculating the probability of any of n *findings* being true requires n times the probability as testing for 1 finding.

https://xkcd.com/882/

In brief, one can only look for so many patterns (i.e. features) in the data before one finds something just by chance (i.e. finding something that does **not** generalize).

"Data mining" is a bad word in some communities!

Statistical Limits. Goal: **Generalization**

Note: *Bonferroni's principle* is simply an abstract idea inspired by a precisely defined method of hypothesis testing called "Bonferroni correction".

We will go over this correction method later. The ***principle*** is the more important idea to understand as a big data practitioner.

In brief, one can only look for so many patterns (i.e. features) in the data before one finds something just by chance (i.e. finding something that does **not** generalize).

"Data mining" is a bad word in some communities!

Normalizing

Count data often need *normalizing* -- putting the numbers on the same “scale”.

Prototypical example: TF.IDF

Normalizing

Count data often need *normalizing* -- putting the numbers on the same “scale”.

Prototypical example: TF.IDF of word *i* in document *j:*

Term Frequency:

$$tf_{ij} = \frac{count_{ij}}{\max_k count_{kj}}$$

$$tf.idf_{ij} = tf_{ij} \times idf_i$$

Inverse Document Frequency:

$$idf_i = log_2(\frac{docs_*}{docs_i}) \propto \frac{1}{\frac{docs_i}{docs_*}}$$

where docs is the number of documents containing word *i*.

Normalizing

Count data often need *normalizing* -- putting the numbers on the same “scale”.

Prototypical example: TF.IDF of word *i* in document *j:*

Term Frequency:

$$tf_{ij} = \frac{count_{ij}}{\boxed{\max_k count_{kj}}}$$

$$tf.idf_{ij} = tf_{ij} \times idf_i$$

Inverse Document Frequency:

$$idf_i = \boxed{log_2}(\frac{docs_*}{docs_i}) \propto \frac{1}{\boxed{\frac{docs_i}{docs_*}}}$$

where docs is the number of documents containing word *i*.

Normalizing

Standardize: puts different sets of data (typically vectors or random variables) on the same scale with the same center.

- Subtract the mean (i.e. “mean center”)
- Divide by standard deviation

$$z_i = \frac{x_i - \bar{x}}{s_x}$$

Power Law

Characterized many frequency patterns when ordered from most to least:

County Populations [r-bloggers.com]

links into webpages [Broader et al., 2000]

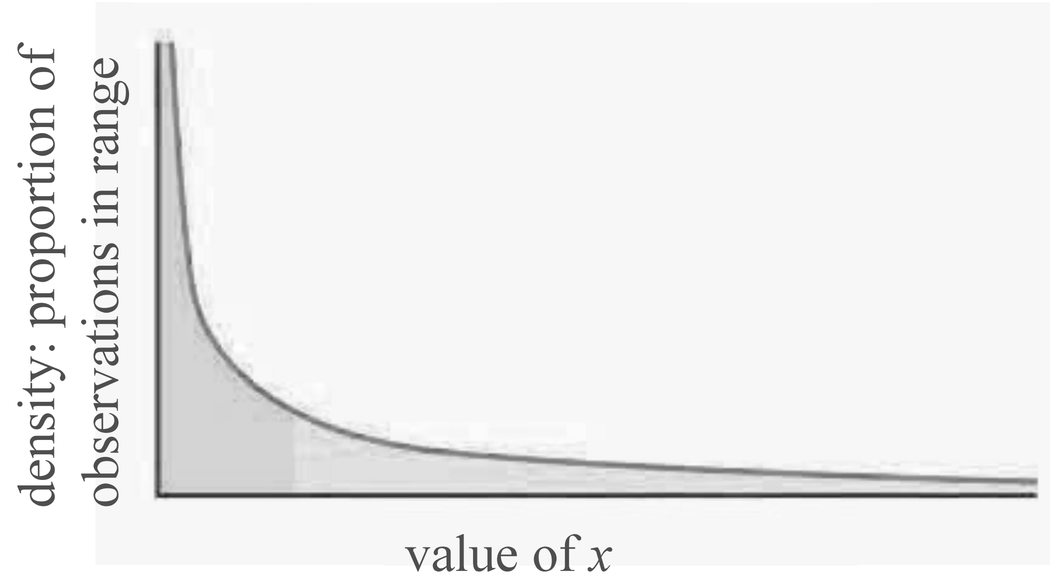
Sales of products [see book]

Frequency of words [Wikipedia, “Zipf’s Law”]

(“popularity” based statistics, especially without limits)

Power Law

$$\log y = b + a \log x$$



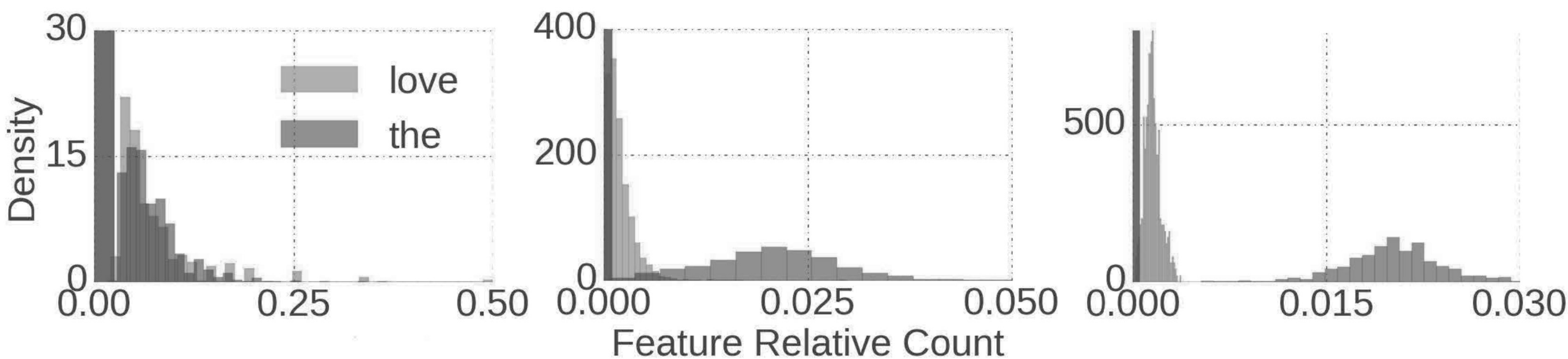
raising to the natural log:

$$y = e^b e^{a \log x} = e^b x^a = cx^a$$

where c is just a constant

Characterizes “the Matthew Effect” -- the rich get richer

Power Law



message-level user-level county-level

Almodaresi, F., Ungar, L., Kulkarni, V., Zakeri, M., Giorgi, S., & Schwartz, H. A. (2017). On the Distribution of Lexical Features at Multiple Levels of Analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (pp. 79-84).

Hash Functions and Indexes

Review:

h: *hash-key -> bucket-number*

Objective: uniformly distribute hash-keys across buckets.

Example: storing word counts.

Hash Functions and Indexes

Review:

h: *hash-key -> bucket-number*

Objective: uniformly distribute hash-keys across buckets.

Example: storing word counts.

$$h(word) = \left(\sum_{char \in word} ascii(char) \right) \% \, \#buckets$$

Hash Functions and Indexes

Review:

h: *hash-key -> bucket-number*

Objective: uniformly distribute hash-keys across buckets.

Example: storing word counts.

$$h(word) = \left(\sum_{char \in word} ascii(char) \right) \% \#buckets$$

Data structures utilizing hash-tables (i.e. O(1) lookup; dictionaries, sets in python) are a friend of big data algorithms! Review further if needed.

Hash Functions and Indexes

Review:

h: *hash-key -> bucket-number*

Objective: uniformly distribute hash-keys across buckets.

Example: storing word counts.

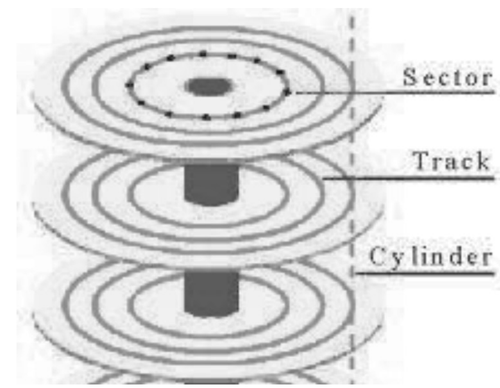
Database Indexes: Retrieve all records with a given *value.* (also review if unfamiliar / forgot)

Data structures utilizing hash-tables (i.e. O(1) lookup; dictionaries, sets in python) are a friend of big data algorithms! Review further if needed.

IO Bounded

Reading a word from disk versus main memory: 10^5 slower!

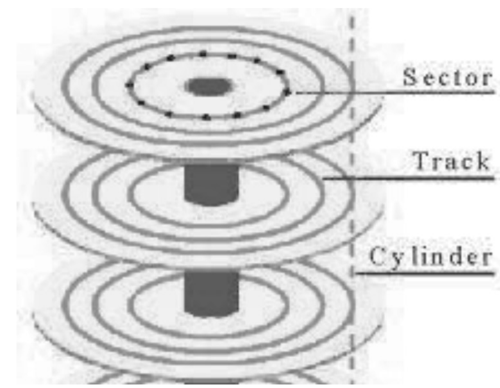
Reading many contiguously stored words
is faster per word, but fast modern disks
still only reach 150MB/s for sequential reads.



IO Bounded

Reading a word from disk versus main memory: 10^5 slower!

Reading many contiguously stored words
is faster per word, but fast modern disks
still only reach 150MB/s for sequential reads.



IO Bound: biggest performance bottleneck is reading / writing to disk.

(starts around 100 GBs; ~10 minutes just to read).

Data

- Unstructured ≈ requires processing to get what is of interest
- Feature extraction used to turn unstructured into structured
- Near infinite amounts of potential features in unstructured data

Data

Structured ⟷ Unstructured

mysql table | email header | satellite imagery | images

vectors matrices | facebook likes | text (email body)

- Unstructured ≈ requires processing to get what is of interest
- Feature extraction used to turn unstructured into structured
- Near infinite amounts of potential features in unstructured data