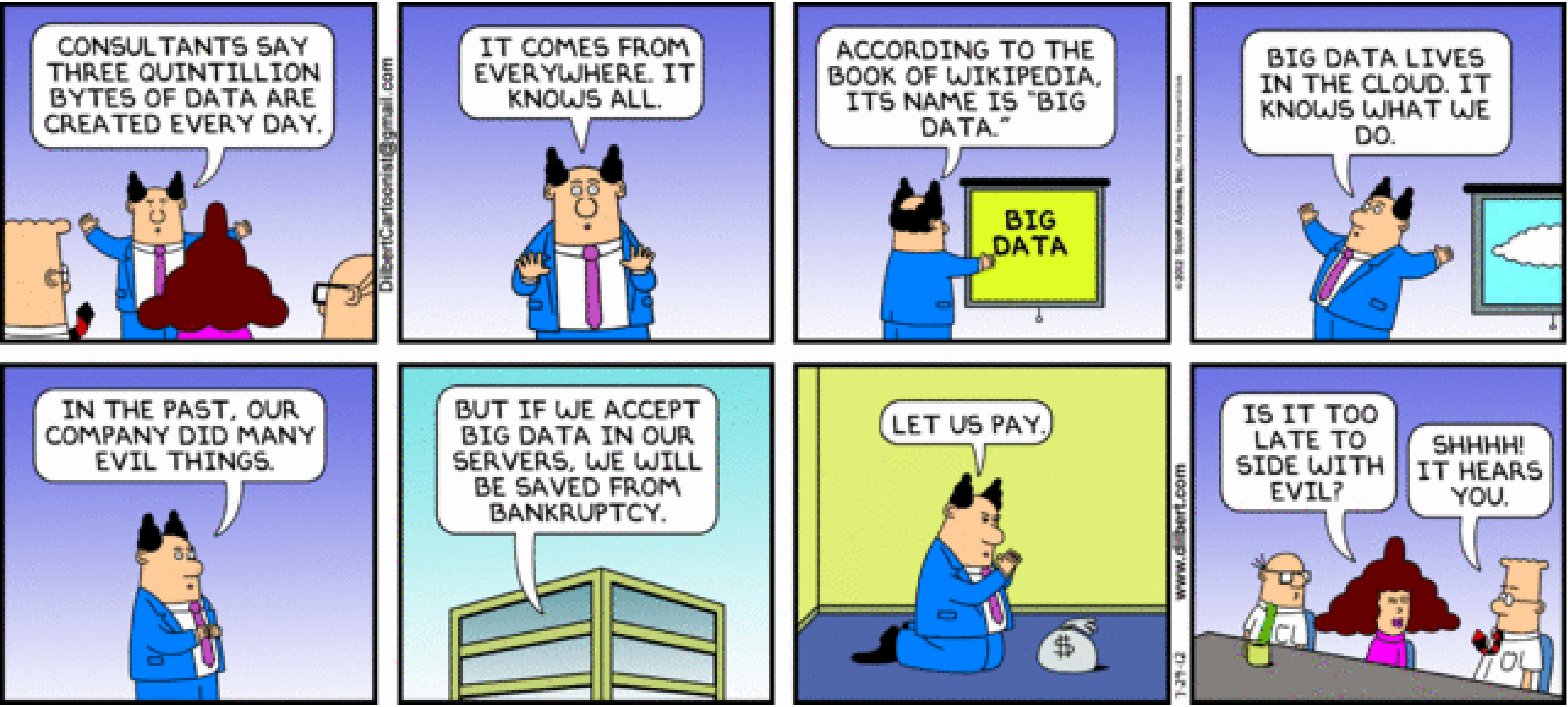
Big Data

CONSULTANTS SAY THREE QUINTILLION BYTES OF DATA ARE CREATED EVERY DAY.
DilbertCartoonist@gmail.com
IT COMES FROM EVERYWHERE. IT KNOWS ALL.
ACCORDING TO THE BOOK OF WIKIPEDIA, ITS NAME IS "BIG DATA."
BIG DATA
BIG DATA LIVES IN THE CLOUD. IT KNOWS WHAT WE DO.
IN THE PAST, OUR COMPANY DID MANY EVIL THINGS.
BUT IF WE ACCEPT BIG DATA IN OUR SERVERS, WE WILL BE SAVED FROM BANKRUPTCY.
LET US PAY.
$
www.dilbert.com
IS IT TOO LATE TO SIDE WITH EVIL?
SHHHH! IT HEARS YOU.

My Point

- Not social impact
- Not wonderful gadgets or services
- But an illustration of computational thinking
  - Invisible ideas enable science, technology, and communication
  - Clever, ingenious, and surprising solutions to specific problems pertaining to information

My Case Study: Spam (Counting)

- Spam: unsolicited commercial email (UCE)
- We will call the other email “ham” – the good stuff!

The Enron Corpus

The Enron Corpus is a large database of over 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission during its investigation after the company's collapse. A copy of the database was subsequently purchased for $10,000 by Andrew McCallum, a computer scientist at the University of Massachusetts Amherst. He released this copy to researchers, providing a trove of data that has been used for studies on social networking and computer analysis of language. The corpus is "unique" in that it is one of the only publicly available mass collections of "real" emails easily available for study, as such collections are typically bound by numerous privacy and legal restrictions which render them prohibitively difficult to access.

- How can we detect spam?
- One approach is to look for certain words in the email that might be indicators of spam
- If we know which words appear frequently in spam, then we tend to believe that an email with that word is spam

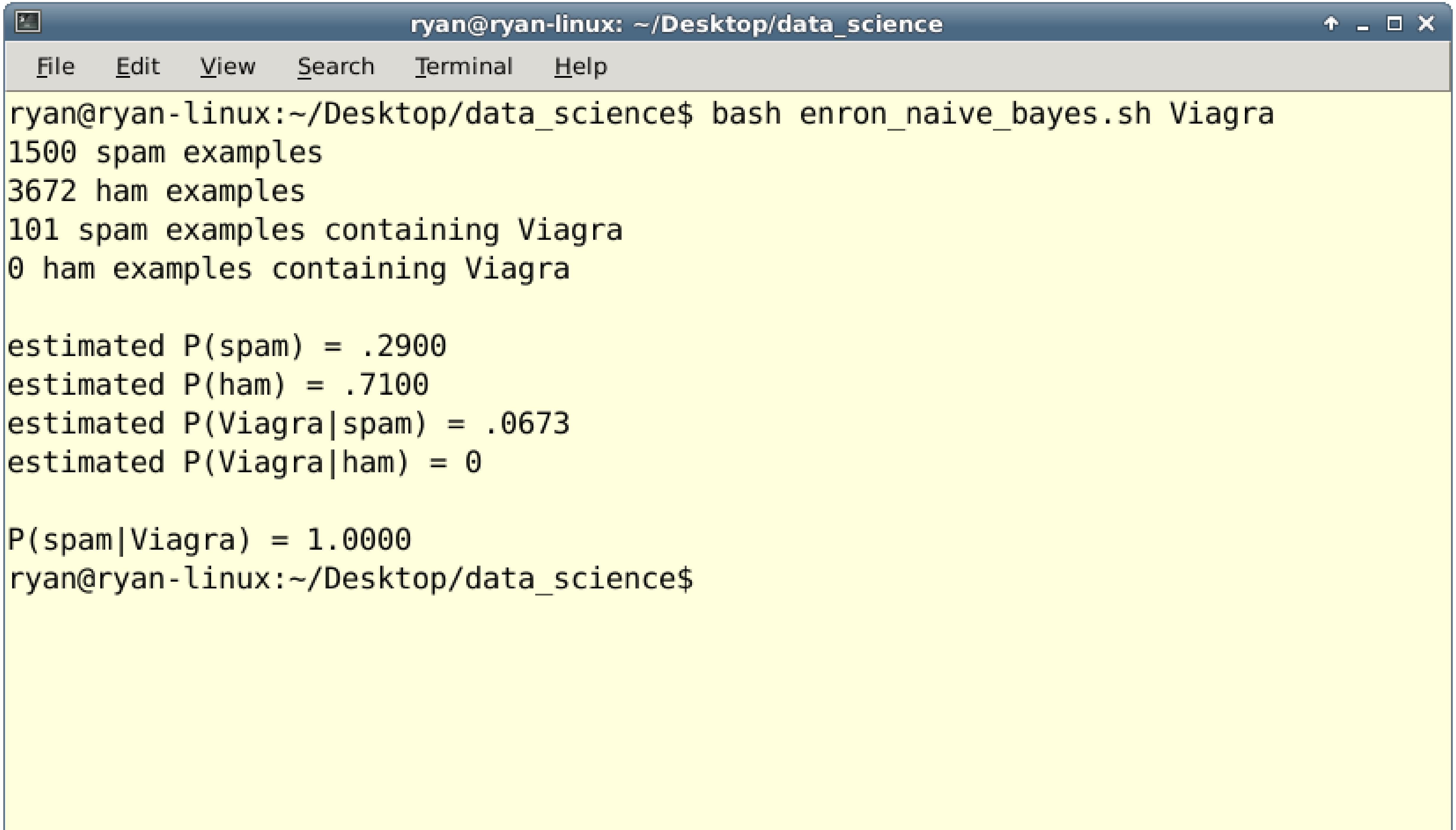
Bayes' Theorem

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

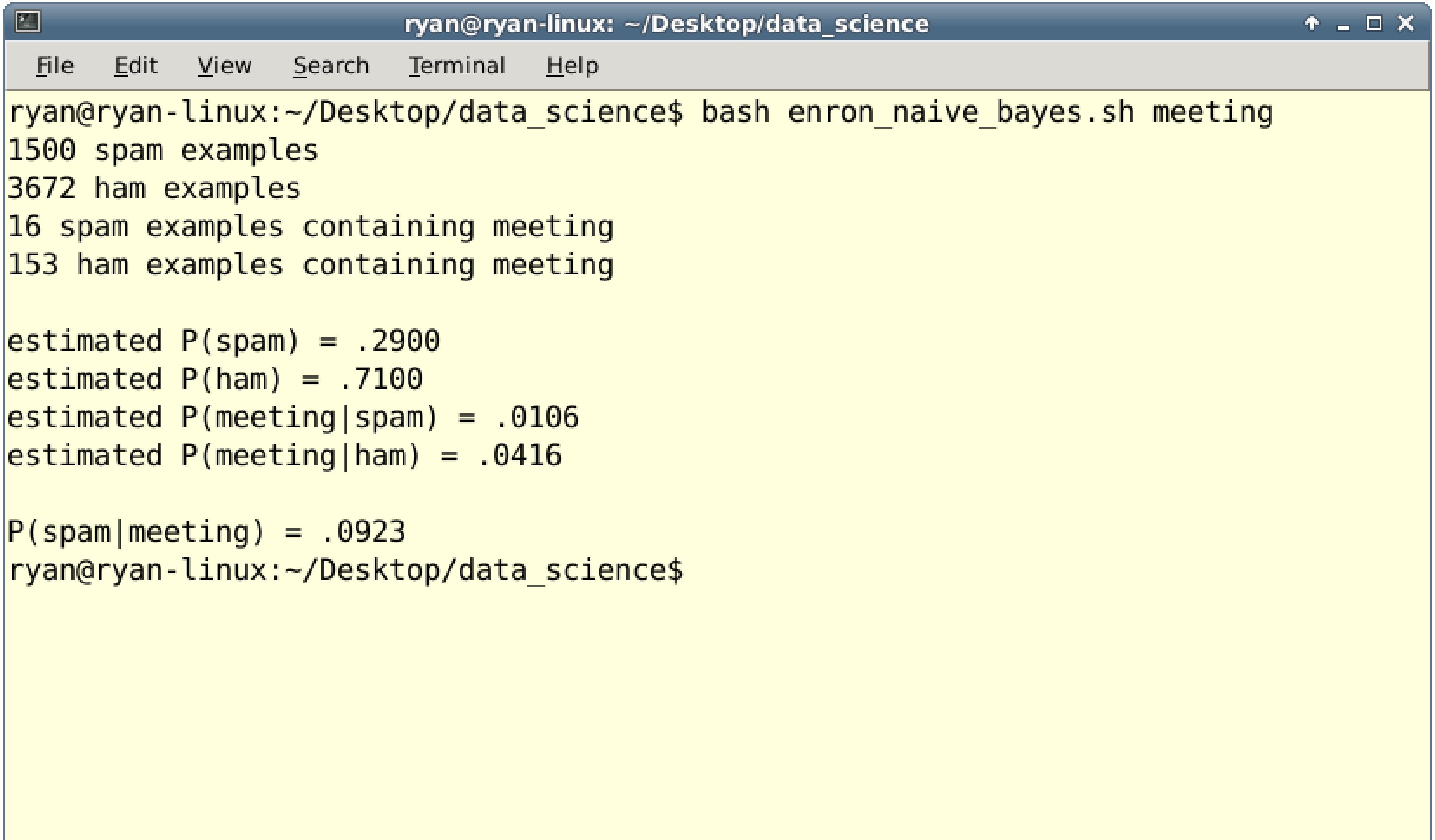
- The probability that the e-mail is spam given that it contains the word “word” is
- P(spam|”word”) = P(“word”|spam) P(spam) / P(“word”)
- Where P(“word”)=P(“word”|spam) P(spam) + P(“word”|ham) P(ham)

We have boiled it down to a counting exercise:

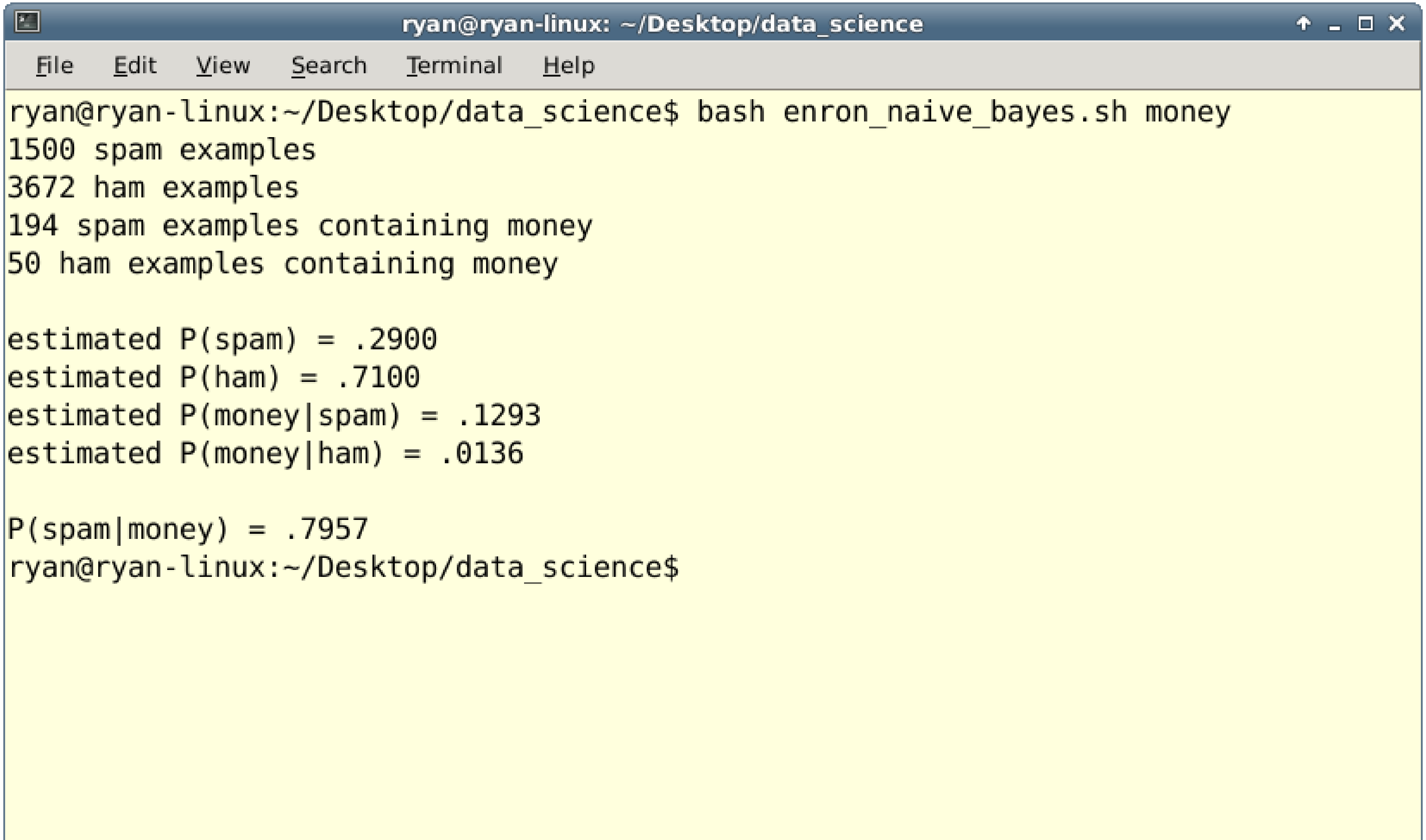
- *P (spam)* counts spam emails versus all emails,
- *P("word"| spam)* counts the prevalence of those spam emails that contain "word," and
- P("*word"| ham)* counts the prevalence of the ham emails that contain "word."


ryan@ryan-linux: ~/Desktop/data_science
File Edit View Search Terminal Help

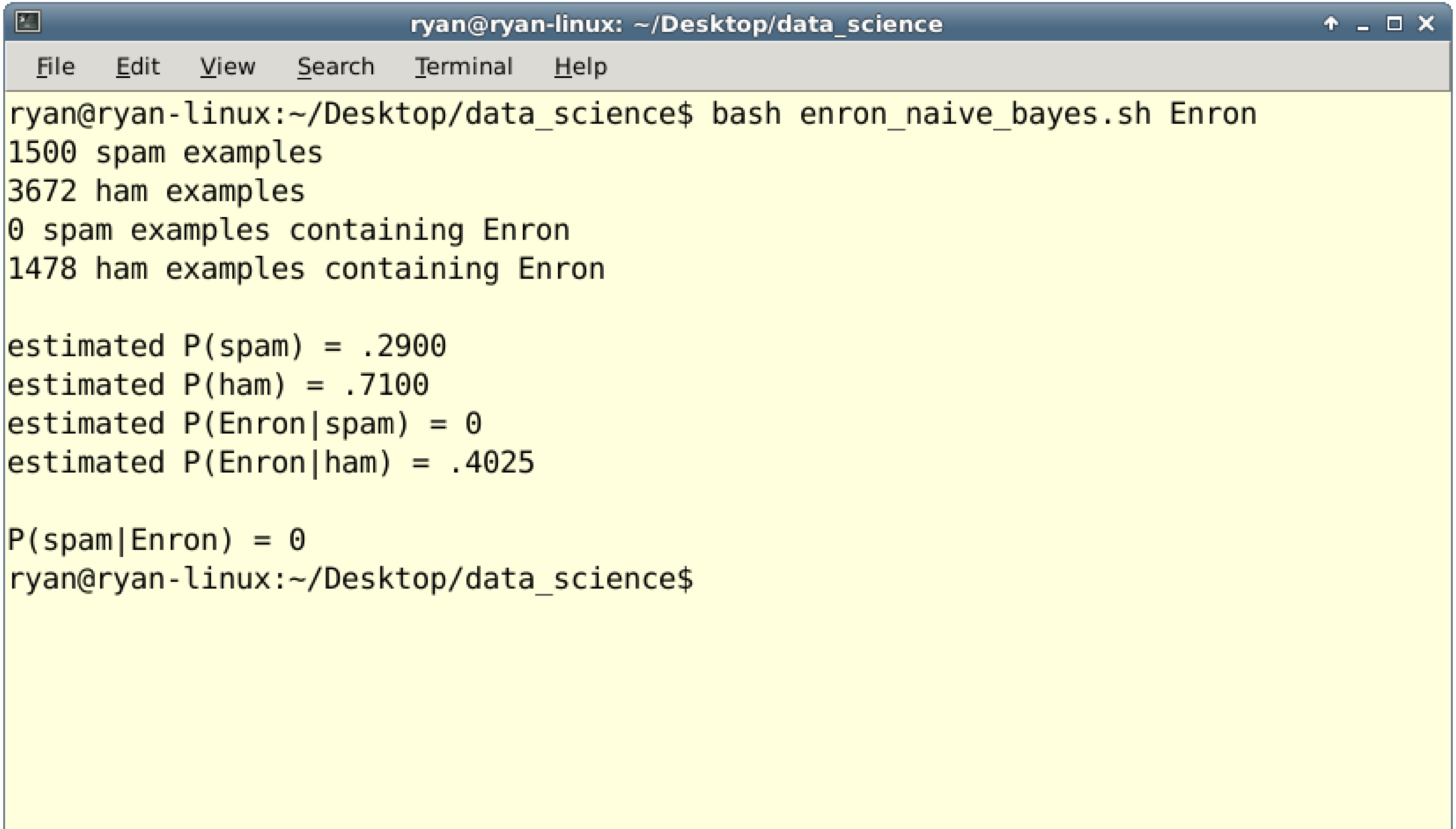
```
ryan@ryan-linux:~/Desktop/data_science$ bash enron_naive_bayes.sh Viagra
1500 spam examples
3672 ham examples
101 spam examples containing Viagra
0 ham examples containing Viagra

estimated P(spam) = .2900
estimated P(ham) = .7100
estimated P(Viagra|spam) = .0673
estimated P(Viagra|ham) = 0

P(spam|Viagra) = 1.0000
ryan@ryan-linux:~/Desktop/data_science$
```


ryan@ryan-linux: ~/Desktop/data_science
File Edit View Search Terminal Help

```
ryan@ryan-linux:~/Desktop/data_science$ bash enron_naive_bayes.sh meeting
1500 spam examples
3672 ham examples
16 spam examples containing meeting
153 ham examples containing meeting

estimated P(spam) = .2900
estimated P(ham) = .7100
estimated P(meeting|spam) = .0106
estimated P(meeting|ham) = .0416

P(spam|meeting) = .0923
ryan@ryan-linux:~/Desktop/data_science$
```


ryan@ryan-linux: ~/Desktop/data_science
File Edit View Search Terminal Help

```
ryan@ryan-linux:~/Desktop/data_science$ bash enron_naive_bayes.sh money
1500 spam examples
3672 ham examples
194 spam examples containing money
50 ham examples containing money

estimated P(spam) = .2900
estimated P(ham) = .7100
estimated P(money|spam) = .1293
estimated P(money|ham) = .0136

P(spam|money) = .7957
ryan@ryan-linux:~/Desktop/data_science$
```


ryan@ryan-linux: ~/Desktop/data_science
File Edit View Search Terminal Help

```
ryan@ryan-linux:~/Desktop/data_science$ bash enron_naive_bayes.sh Enron
1500 spam examples
3672 ham examples
0 spam examples containing Enron
1478 ham examples containing Enron

estimated P(spam) = .2900
estimated P(ham) = .7100
estimated P(Enron|spam) = 0
estimated P(Enron|ham) = .4025

P(spam|Enron) = 0
ryan@ryan-linux:~/Desktop/data_science$
```

- It is not difficult to count the number times a word appears in a collection of e-mail
- But imagine 100,000 different words in a trillion different e-mail messages!
- There are many others ways to detect UCE (spam)

- In the age of Big Data, counting becomes a surprising challenge

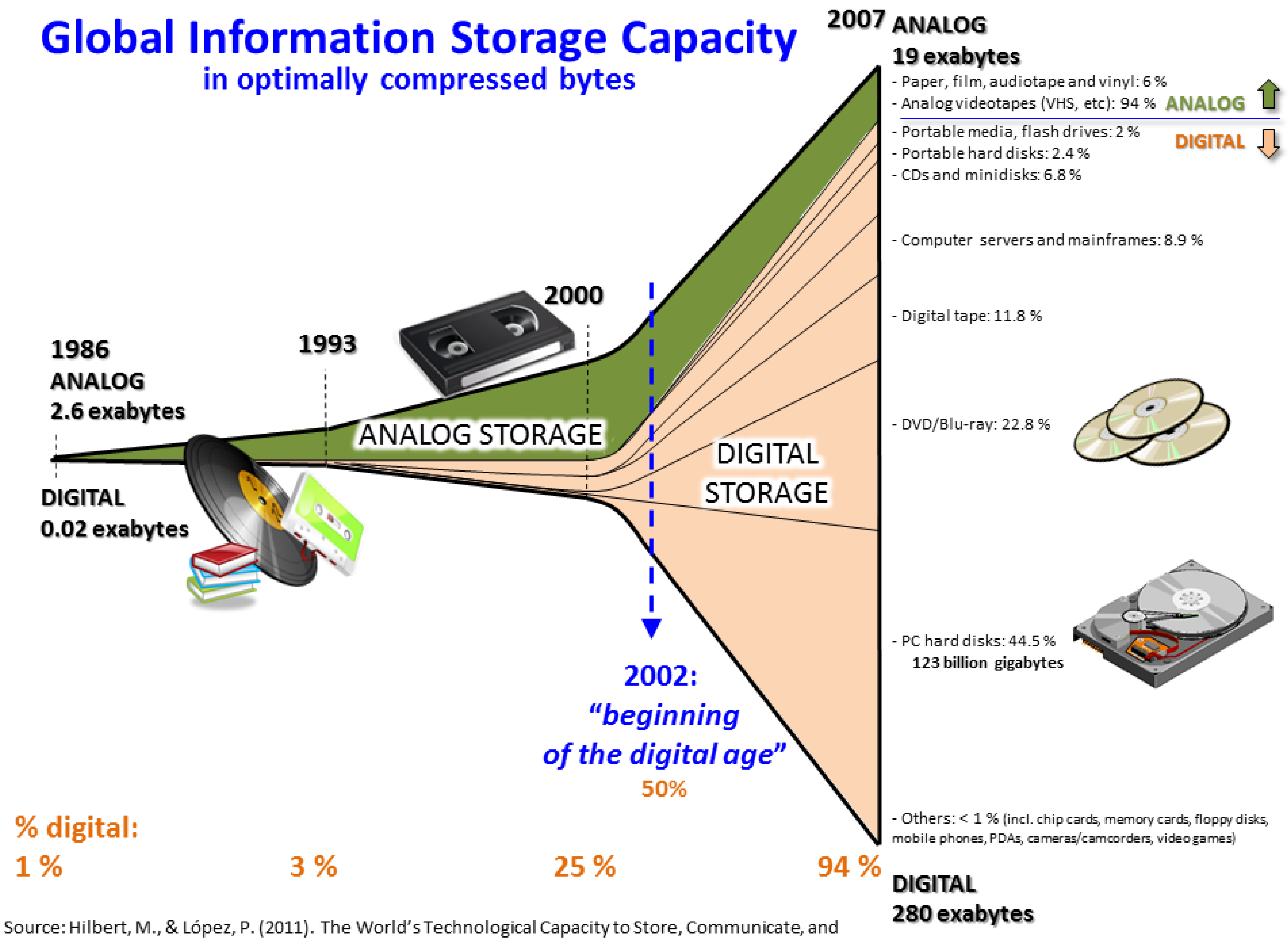
0, 1, 2, 3, 4, 5, …

BIG DATA

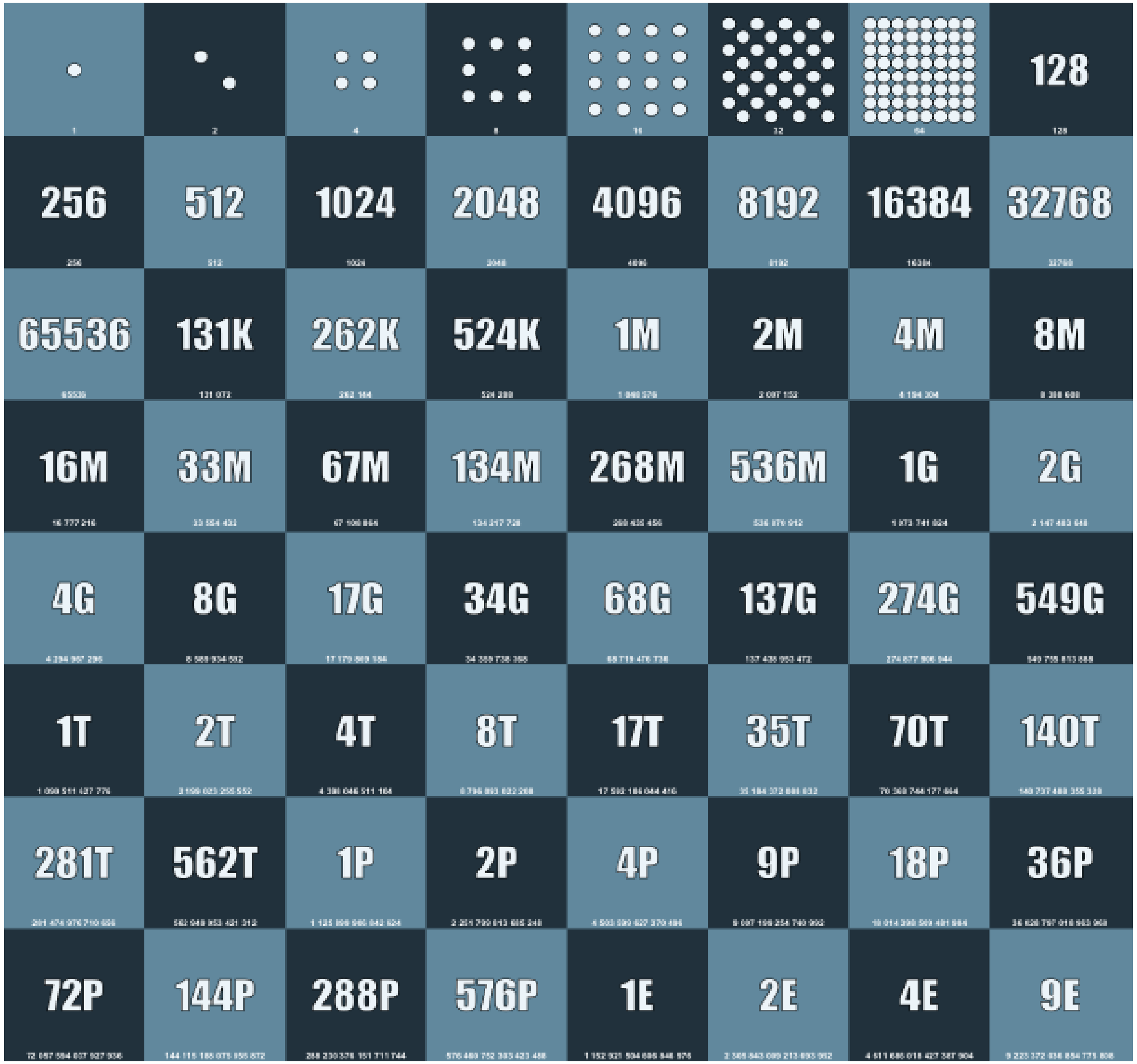
- Estimates by the Radicati Group put the number of emails sent per day to be around 294 billion in 2010.
- According to a Radicati Group study from May 2009, there are about 1.9 billion e-mail users worldwide. That makes more than one in every five persons on the earth use e-mail. In June 2012, Google reported 425 million users worldwide.

Global Information Storage Capacity

in optimally compressed bytes



Source: Hilbert, M., & López, P. (2011). The World's Technological Capacity to Store, Communicate, and Compute Information. *Science*, 332(6025), 60 –65. http://www.martinhilbert.net/WorldInfoCapacity.html



10^3	kilo	k
10^6	mega	M
10^9	giga	G
10^{12}	tera	T
10^{15}	peta	P
10^{18}	exa	E

“Things” versus “Events”

- The number of events (transactions, observations) that are and can be made about agents in any period of time grows and grows.
- The events from a pattern in the ocean of information. Individually they may be insignificant.
- If the events are recorded it may be possible to detect the signature of a important precursors to major happenings.
- The data can be used to predict the weather, predict terrorism, predict the stock market, predict the spread of flu, predict crime, the winner of the superbowl, and so on

- Big data has required new solutions
- Of course, one can always buy bigger and bigger hardware, but that is not especially interesting
- **Computational thinking** is about creating solutions (finding algorithms) requiring as little time and hardware as possible.
- In many cases ingenious algorithms have made such a large difference that new science, new technology, and new kinds of communication are possible.

Count Tracking

- The frequency of words in e-mail traffic
- The frequency of items bought by a retailer
- The frequency of shares traded on NSDAQ
- The frequency of logins by gmail users
- Many tasks of massive data distributions such as summarizing, mining, classification, anomaly detection, require count tracking

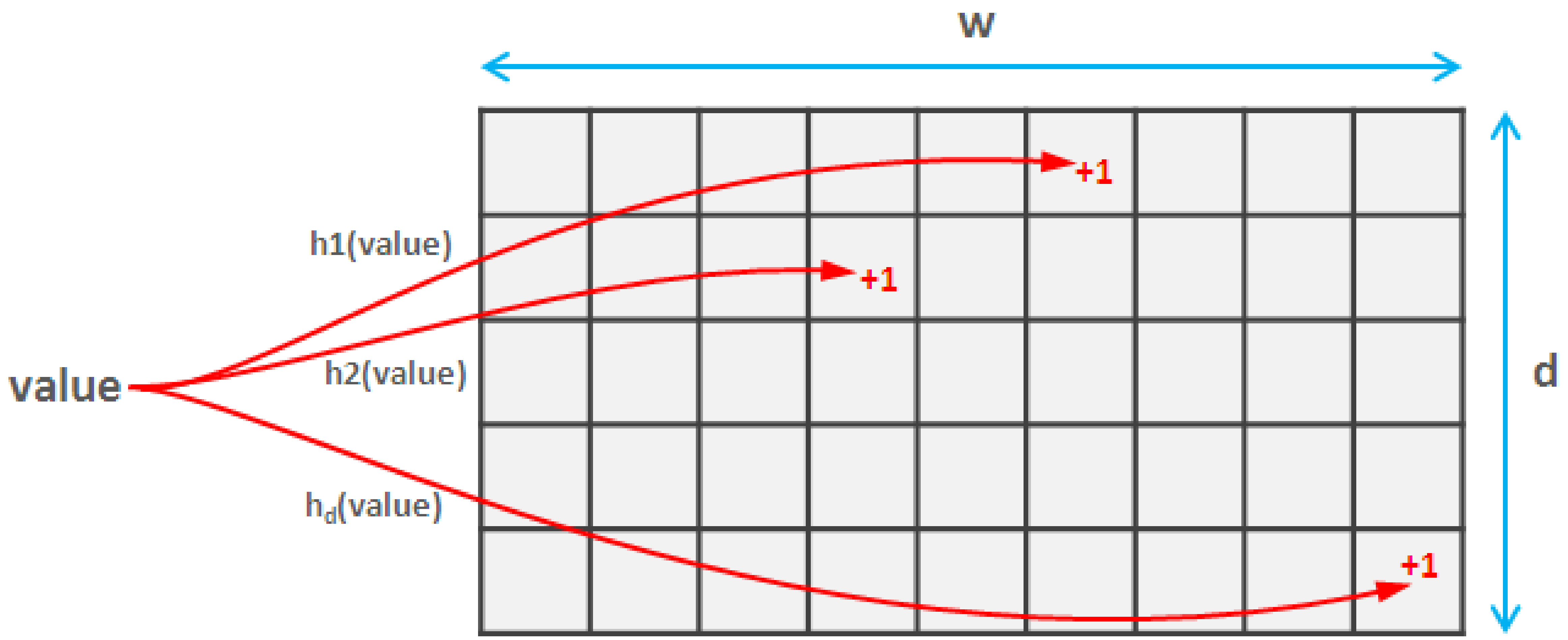
Existing Approaches

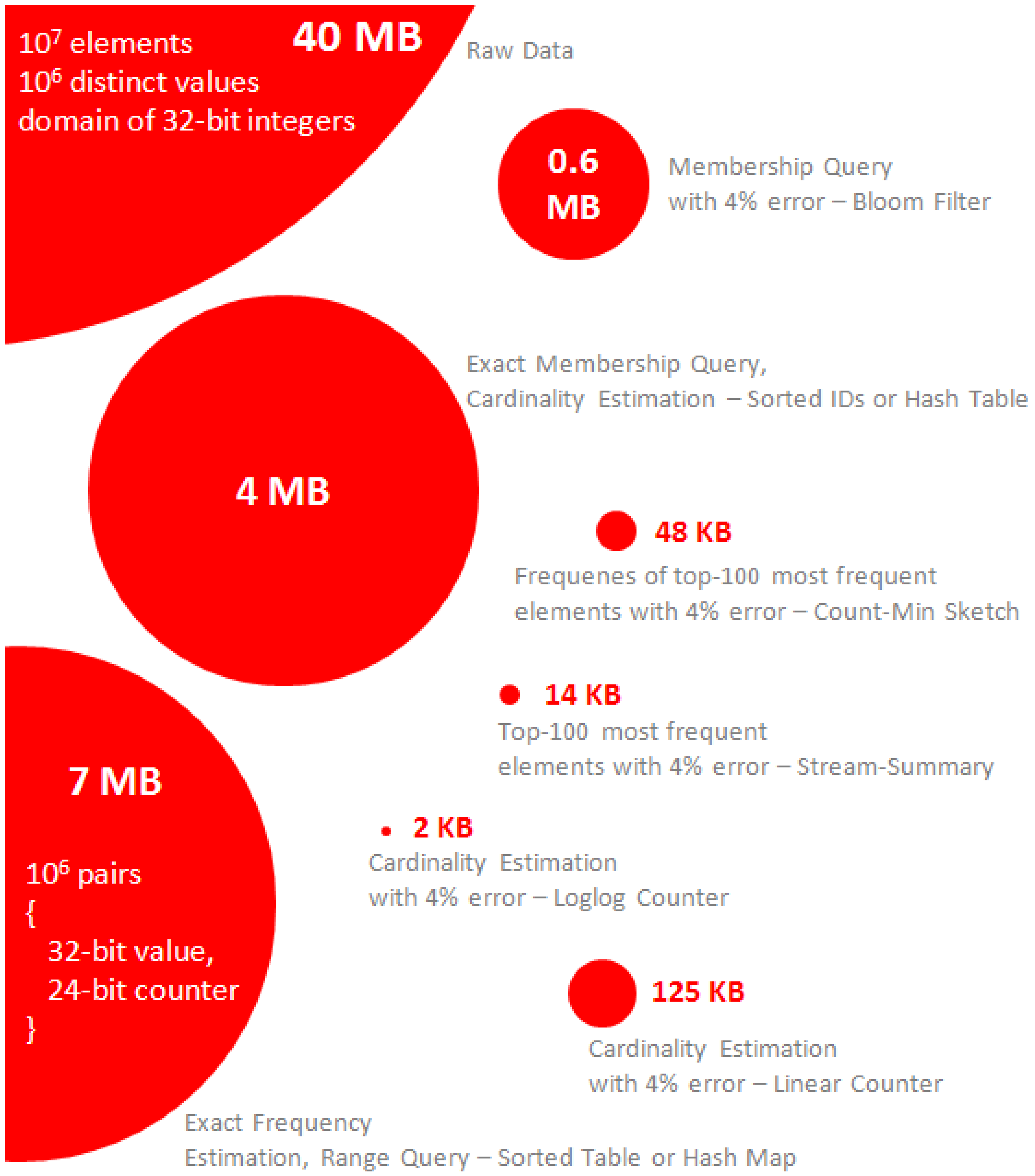
- Many sophisticated approaches are useful here: trees, hashing, etc
- Over the years these have proven quite successful
- In the era of Big Data these approaches are just too big and slow
- In the era of Big Data we can tolerate some lack of exactness
- Is possible to find a computational that is much smaller and faster, but for which there might some well-prescribe amount of error?

“Count-Min Sketch”

- One clever solution is called “count-min sketch” which was created in the last ten years
- It takes advantage of that fact that in most tasks it is reasonable to replace the exact answer with a high-quality approximation
- This beautiful algorithm can be explained fairly easy

Count Tracking




10^7 elements
10^6 distinct values
domain of 32-bit integers
40 MB
Raw Data
0.6 MB
Membership Query
with 4% error – Bloom Filter
4 MB
Exact Membership Query,
Cardinality Estimation – Sorted IDs or Hash Table
48 KB
Frequenes of top-100 most frequent
elements with 4% error – Count-Min Sketch
14 KB
Top-100 most frequent
elements with 4% error – Stream-Summary
2 KB
Cardinality Estimation
with 4% error – Loglog Counter
7 MB
10^6 pairs
{
32-bit value,
24-bit counter
}
125 KB
Cardinality Estimation
with 4% error – Linear Counter
Exact Frequency
Estimation, Range Query – Sorted Table or Hash Map

Additional Nice Properties

- The approach is highly suited for parallelization and distributed computation (very important these days)
- Row-wise on different threads of a multi-processor machine
- Separate domains of local values can be easily maintained on separate, local machines and large summaries can be easily computed

The End

Cormode, Graham; S. Muthukrishnan (2004). "An Improved Data Stream Summary: The Count-Min Sketch and its Applications". *J. Algorithms* **55**: 29–38.

The problem with Big Data

Consider a popular website which wants to keep track of statistics on the queries used to search the site. One could keep track of the full log of queries, and answer exactly the frequency of any search query at the site. However, the log can quickly become very large. This problem is an instance of the count tracking problem. Even known sophisticated solutions for fast querying such as a tree-structure or hash table to count up the multiple occurrences of the same query, can prove to be slow and wasteful of resources. Notice that in this scenario, we can tolerate a little imprecision. In general, we are interested only in the queries that are asked frequently. So it is acceptable if there is some fuzziness in the counts. Thus, we can tradeoff some precision in the answers for a more efficient and lightweight solution. This tradeoff is at the heart of sketches.

Cormode and Muthurishnon, 2011