



Fast Disk Analysis with Random Sampling

Simson L. Garfinkel

Associate Professor, Naval Postgraduate School

CENIC 2010 — March 9, 2010 - 5:06pm

<http://domex.nps.edu/deep/>

NPS is the Navy's Research University.



Location: Monterey, CA

Campus Size: 627 acres

Students: 1500

- US Military (All 5 services)
- US Civilian (Scholarship for Service & SMART)
- Foreign Military (30 countries)

Digital Evaluation and Exploitation:

- *Research* computer forensics.
- *Develop* “corpora” for use in research & education.
- *Identify* limitations of current tools & opportunities for improvement.
- <http://domex.nps.edu/deep/>

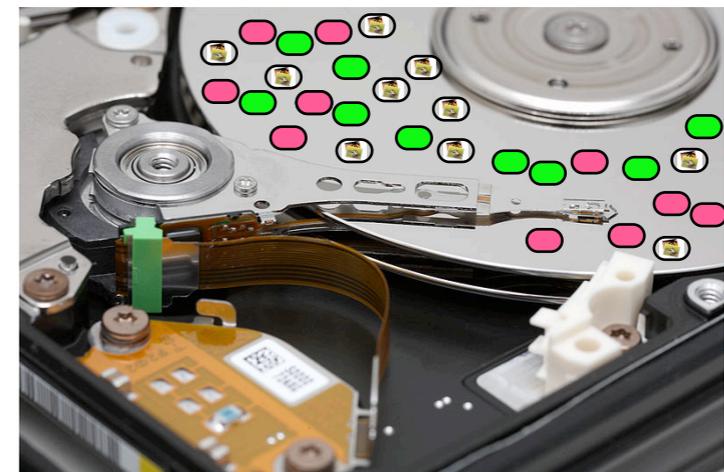


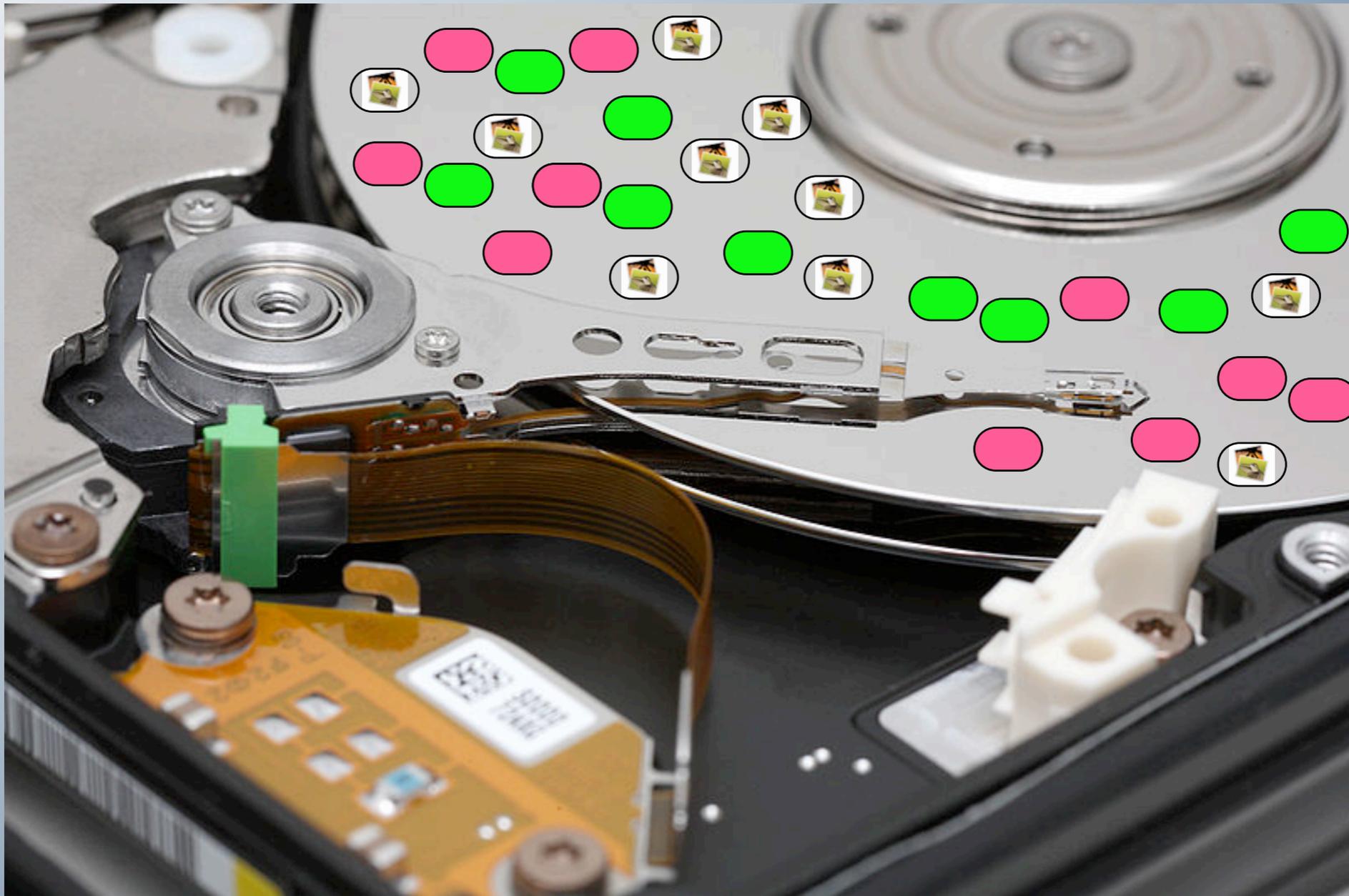
“The views expressed in this presentation are those of the author and do not necessarily reflect those of the Department of Defense or the US Government.”



This talk is about a breakthrough in digital forensics.

How to analyze a 1TB drive in 2 minutes.





Instant Drive Analysis with Statistical Sampling



Question: Can we analyze a 1TB drive in a minute?

What if US agents encounter a hard drive at a border crossing?



Or a search turns up a room filled with servers?



It takes 3.5 hours to read the contents of a 1 TB drive. What can you learn in 1 minute?

		
Minutes	208	1
Max Data Read	1 TB	4.8 GB

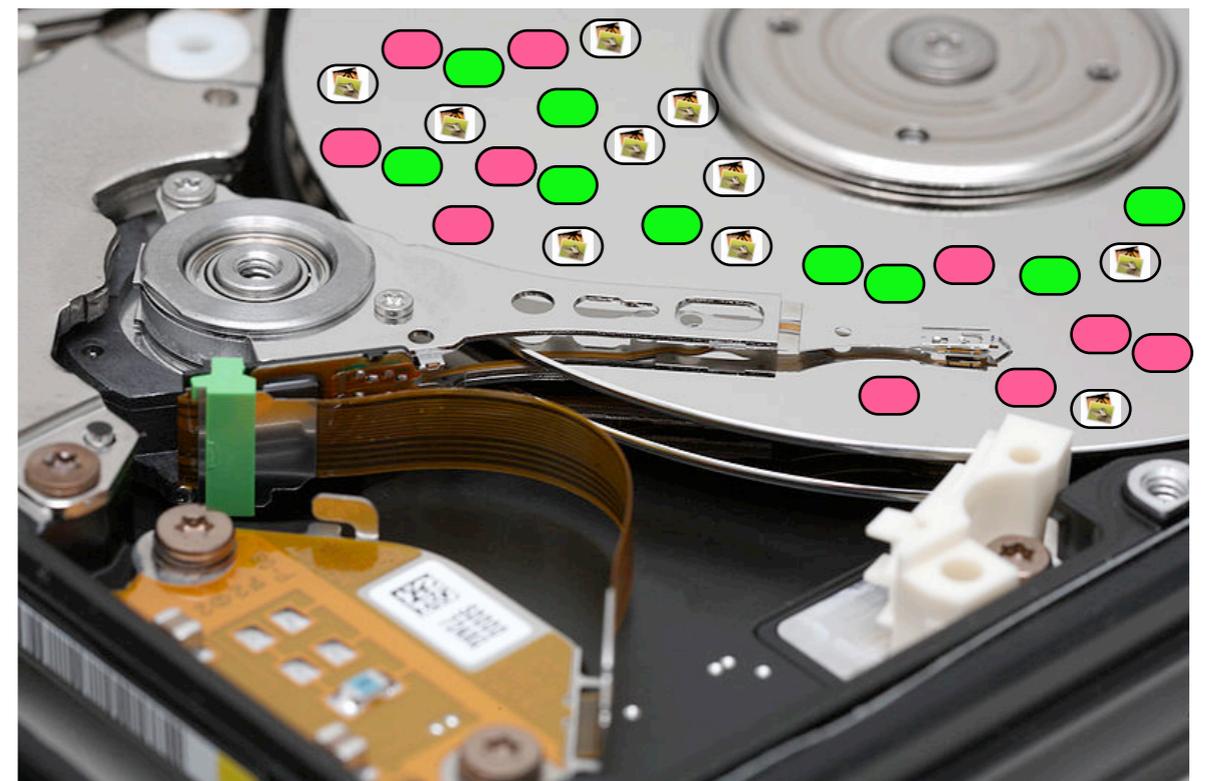
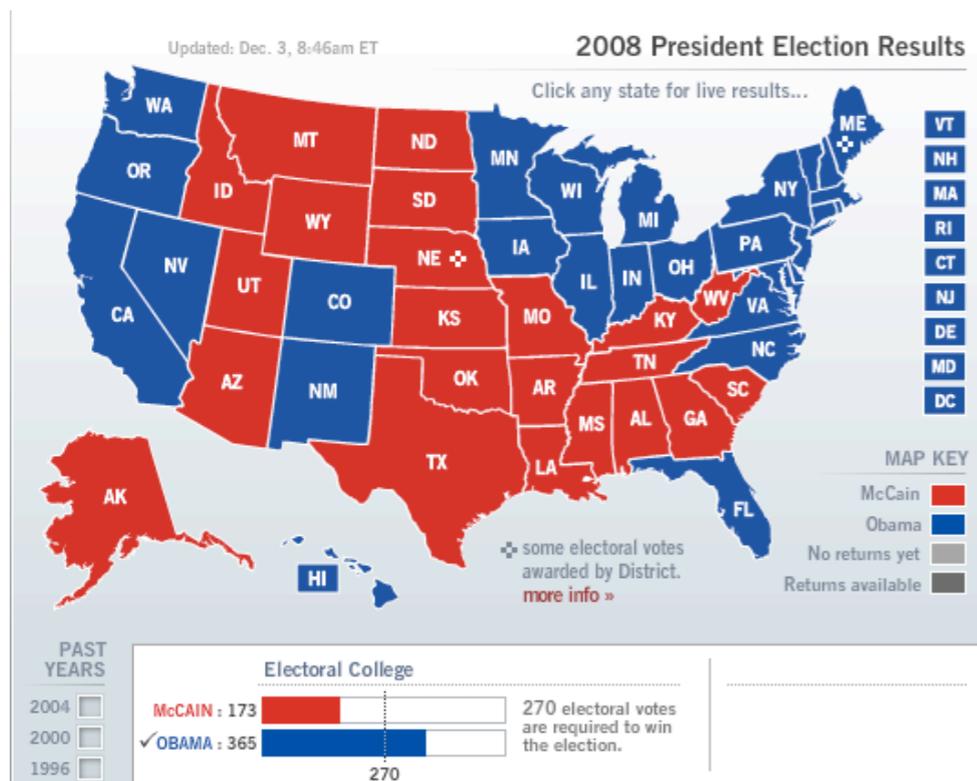
4.8 GB (0.48%) is a tiny fraction of the disk.

But 4.8 GB is a lot of data!

Hypothesis: The contents of the disk can be predicted by identifying the contents of randomly chosen sectors.

US elections can be predicted by sampling a few thousand households:

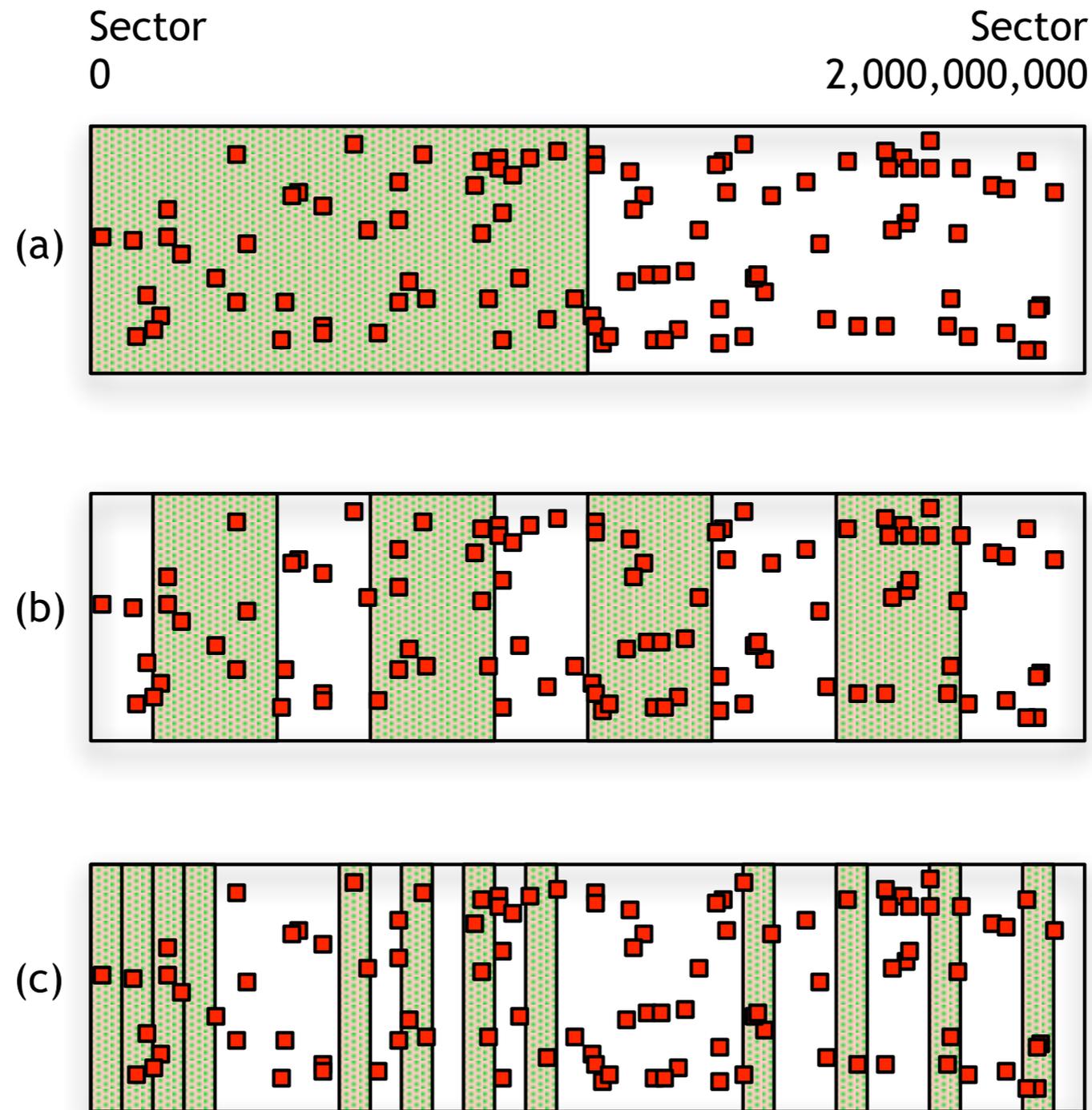
Hard drive contents can be predicted by sampling a few thousand sectors:



The challenge is identifying *likely voters*.

The challenge is *identifying the content* of the sampled sectors.

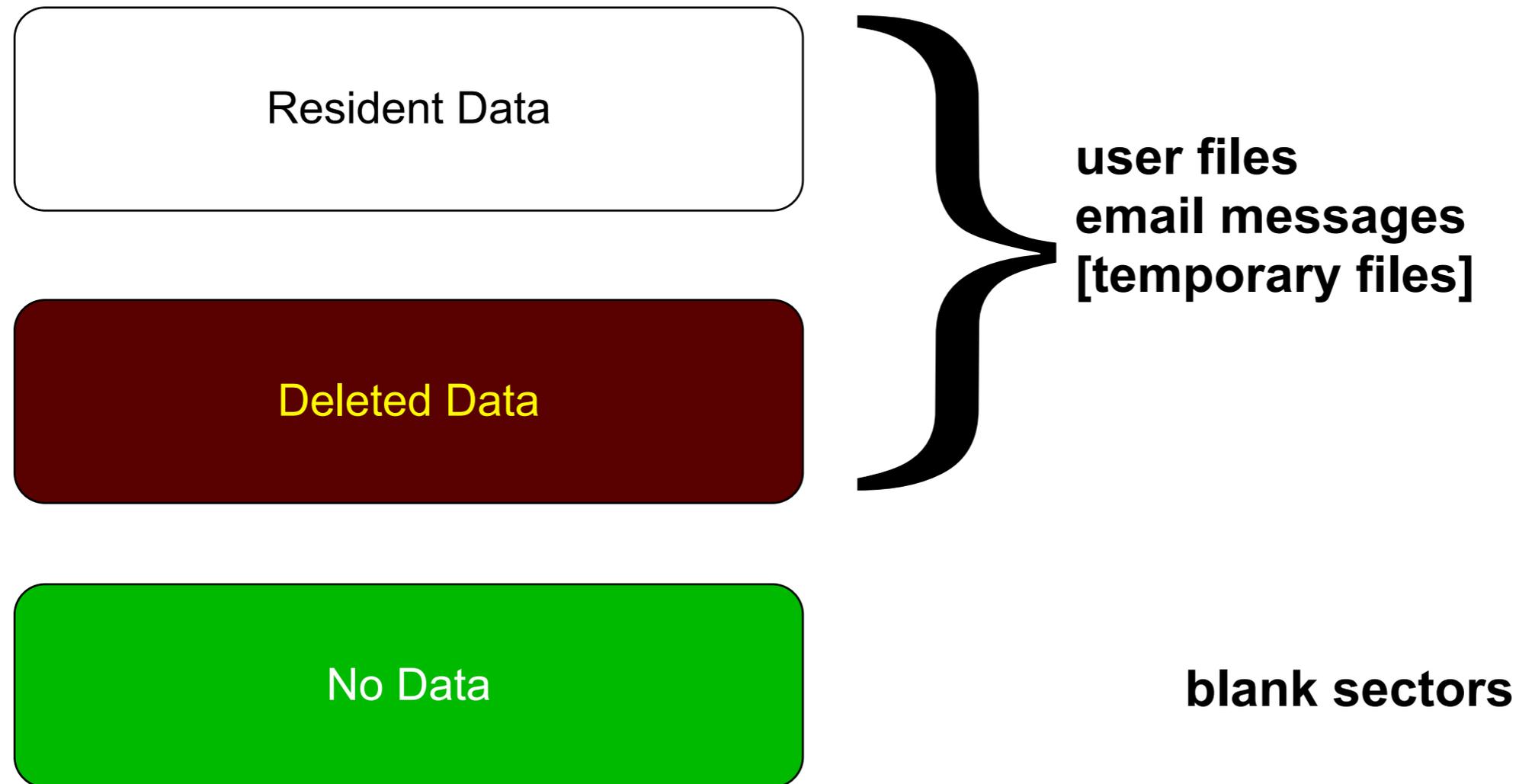
We use random sampling;
any other approach could be exploited by an adversary.



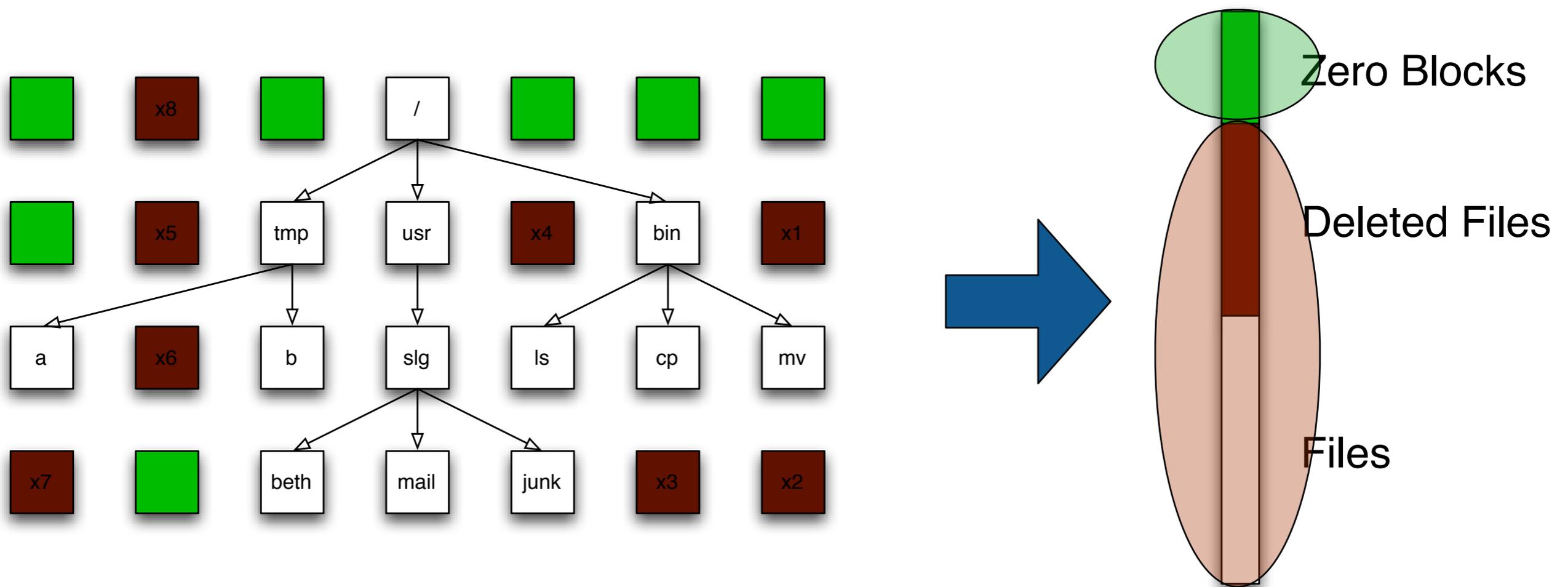
But sampling has an important limitation...



Data on hard drives divides into three categories:

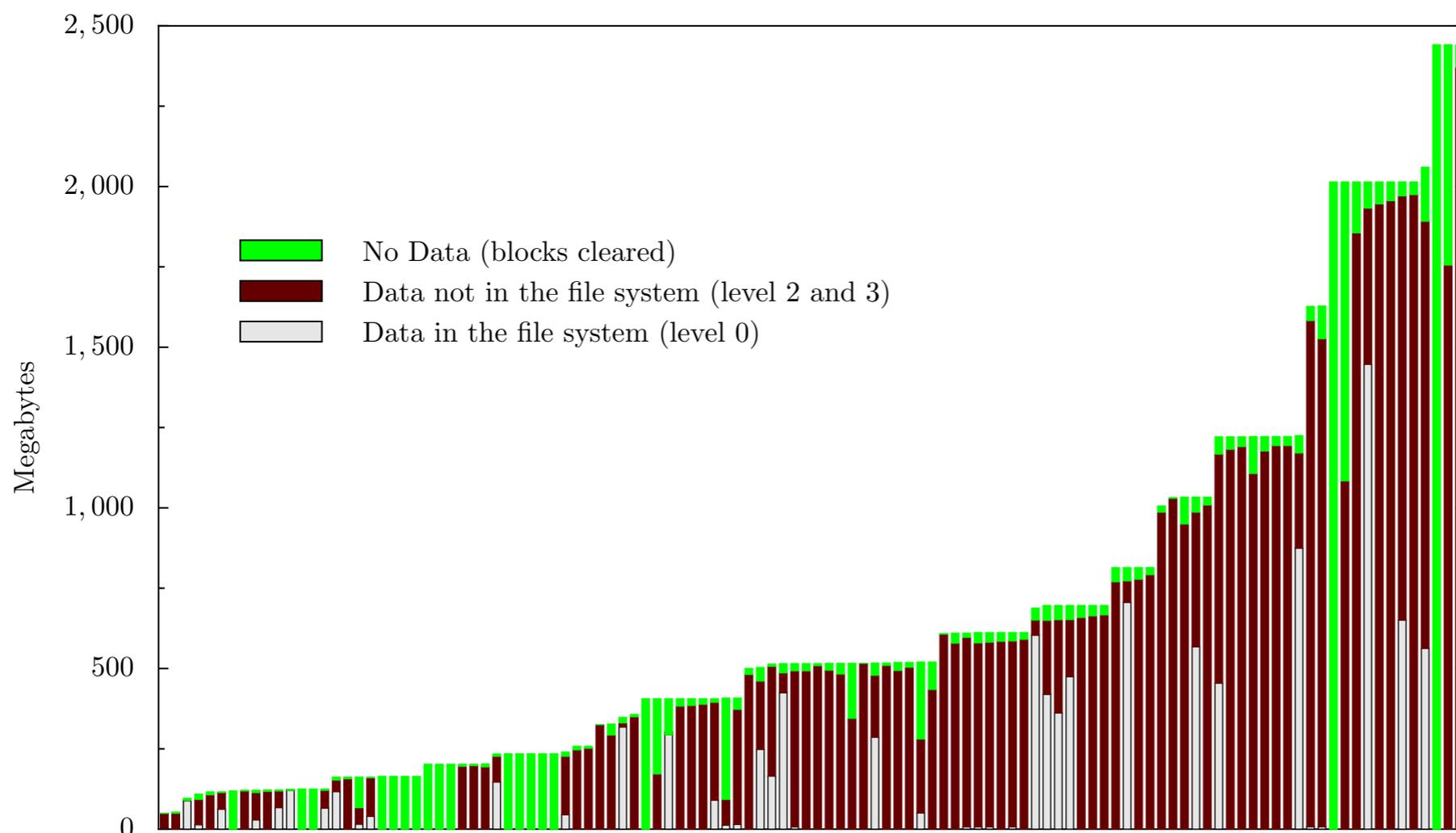


Sampling can distinguish between "zero" and data. It can't distinguish between resident and deleted.



Let's simplify the problem. Can we use statistical sampling to verify wiping?

I bought 2000 hard drives between 1998 and 2006.
Most of were not properly wiped.



It should be easy to use random sampling to distinguish a properly cleared disk from one that isn't.

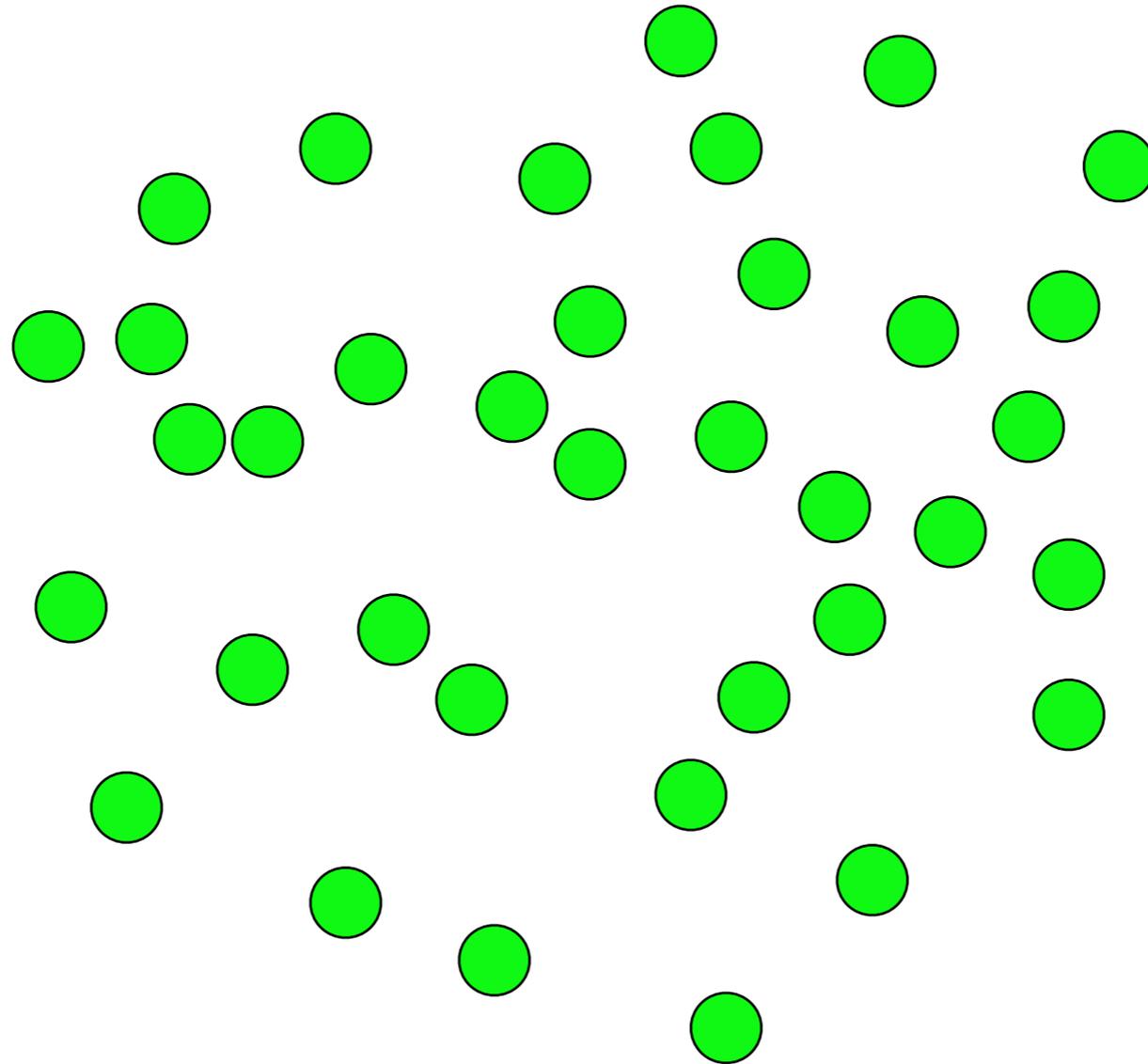


Let's try reading 10,000 random sectors and see what happens....

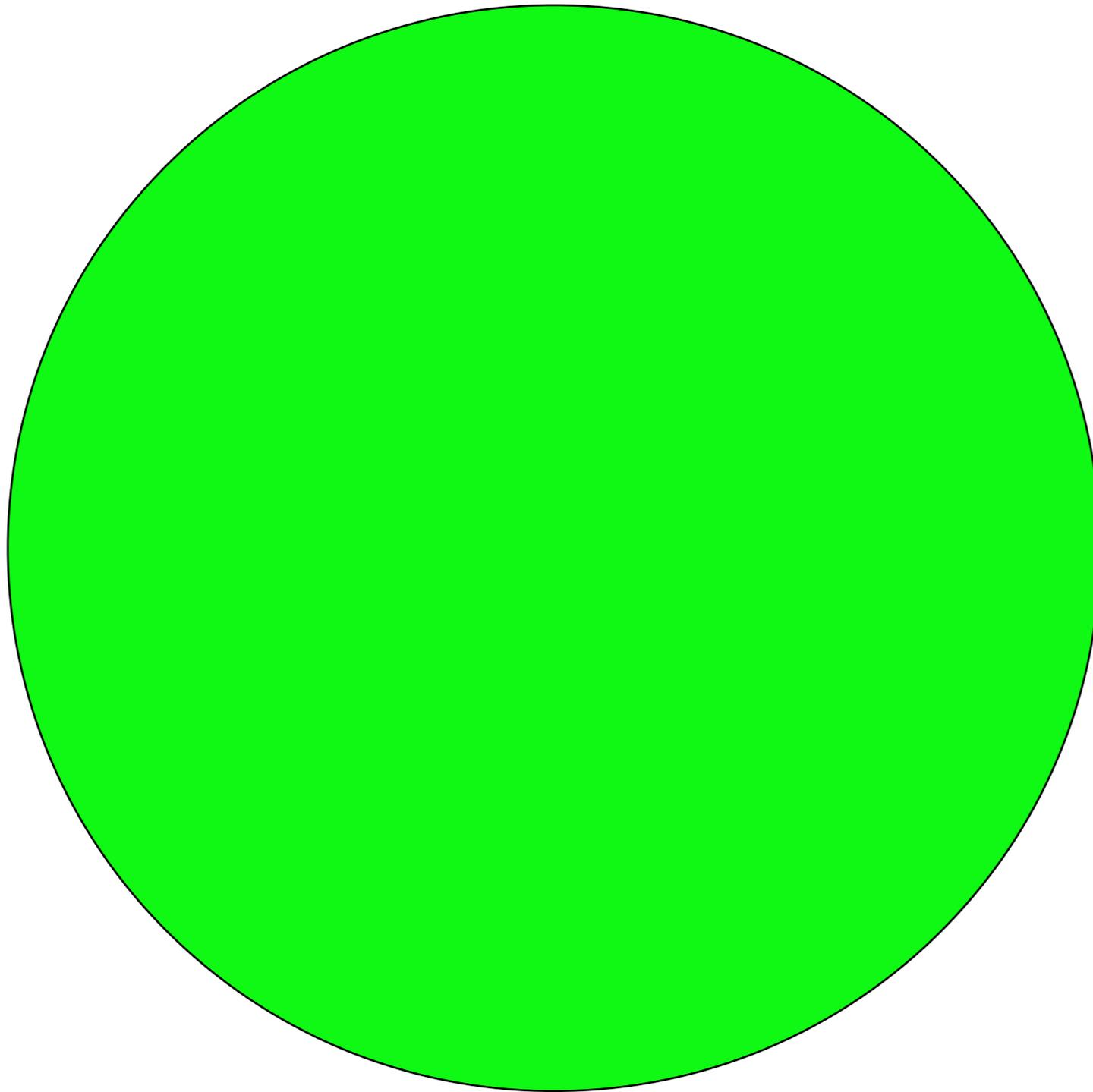
We read 10,000 randomly-chosen sectors ...
and they are all blank



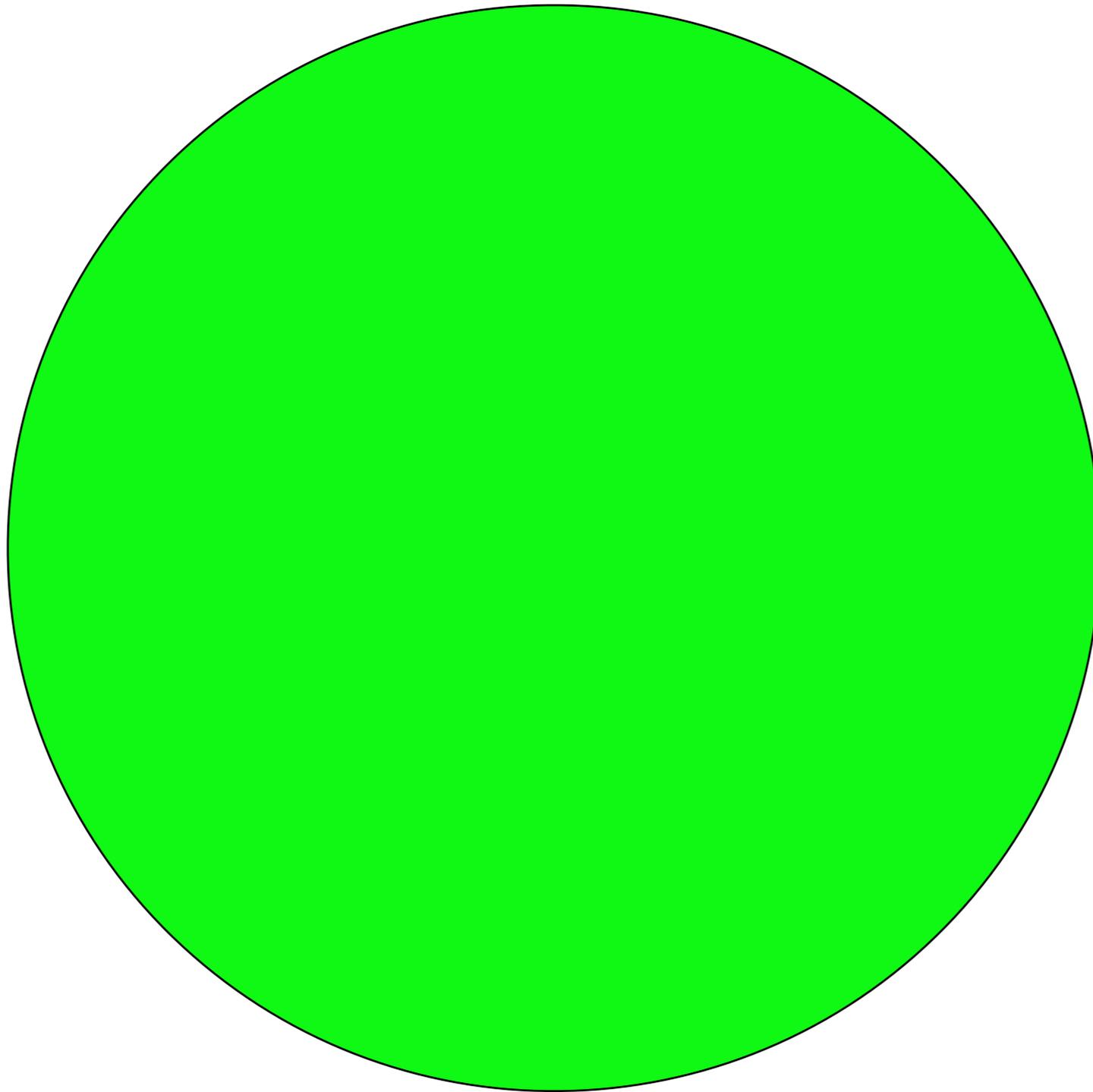
We read 10,000 randomly-chosen sectors ...
and they are all blank



We read 10,000 randomly-chosen sectors ...
and they are all blank



We read 10,000 randomly-chosen sectors ...
and they are all blank



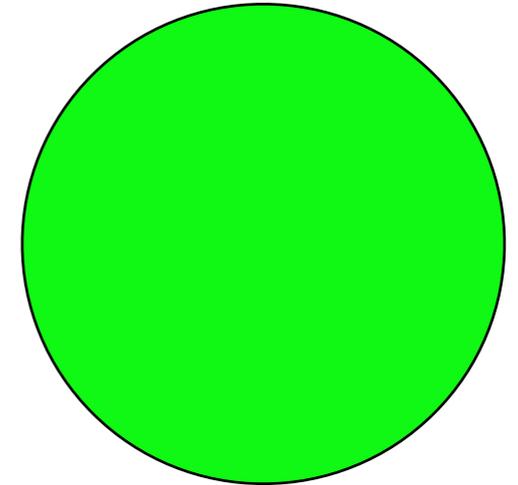
Chances are good that they are all blank.



Random sampling *can't* find a disk with a single sector.

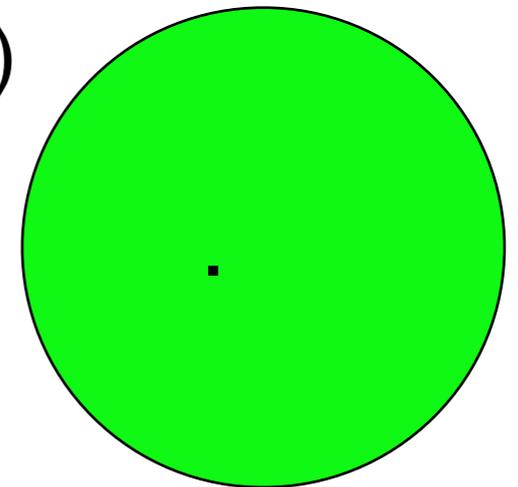
If the disk has 2,000,000,000 blank sectors (0 with data)

- The sample is identical to the population



If the disk has 1,999,999,999 blank sectors (1 with data)

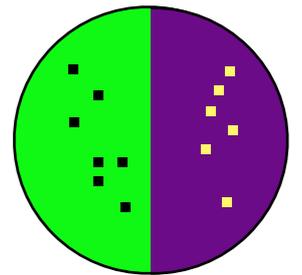
- The sample is representative of the population.
- We will only find that 1 sector using exhaustive search.



What about non-uniform distributions?

If the disk has 1,000,000,000 blank sectors (1,000,000,000 with data)

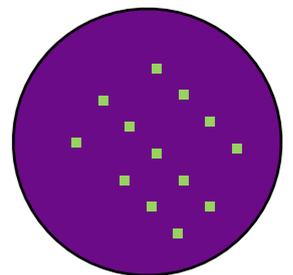
- The sampled frequency should match the distribution.
- *This is why we use random sampling.*



If the disk has 10,000 blank sectors (1,999,990,000 with data)

— and all these are the sectors that we read???

- We are incredibly unlucky.
- ***Somebody has hacked our random number generator!***



Rephrase the problem.

Not a blank disk; a disk with less than 10MB of data.

Sectors on disk: 2,000,000,000 (1TB)

Sectors with data: 20,000 (10 MB)

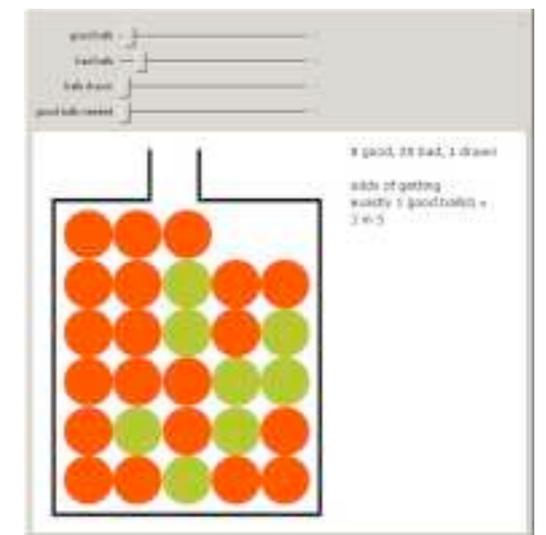
Chose one sector. Odds of missing the data:

- $(2,000,000,000 - 20,000) / (2,000,000,000) = 0.99999$
- You are *very likely* to miss one of 20,000 sectors if you pick just one.

Chose a second sector. Odds of missing the data on both tries:

- $0.99999 * (1,999,999,999 - 20,000) / (1,999,999,999) = .99998$
- You are still *very likely* to miss one of 20,000 sectors if you pick two.

But what if you pick 1000? Or 10,000? Or 100,000?



The more sectors picked, the less likely you are to miss *all* of the sectors that have non-NULL data.

$$P(X = 0) = \prod_{i=1}^n \frac{((N - (i - 1)) - M)}{(N - (i - 1))} \quad (5)$$

Sampled sectors	Probability of not finding data	Non-null data		Probability of not finding data with 10,000 sampled sectors
		Sectors	Bytes	
1	0.99999	20,000	10 MB	0.90484
100	0.99900	100,000	50 MB	0.60652
1000	0.99005	200,000	100 MB	0.36786
10,000	0.90484	300,000	150 MB	0.22310
100,000	0.36787	400,000	200 MB	0.13531
200,000	0.13532	500,000	250 MB	0.08206
300,000	0.04978	600,000	300 MB	0.04976
400,000	0.01831	700,000	350 MB	0.03018
500,000	0.00673	1,000,000	500 MB	0.00673

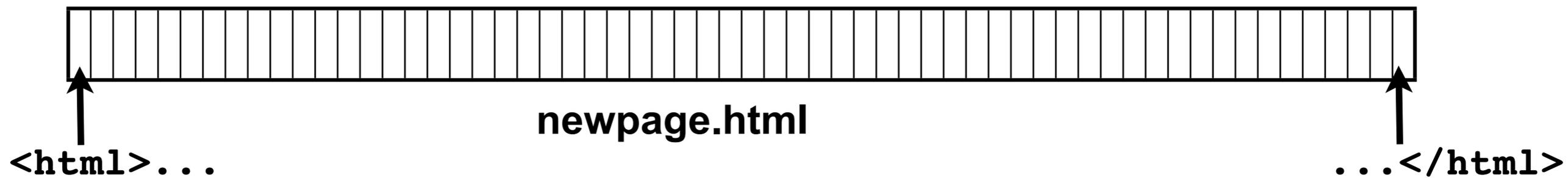
Table 1: Probability of not finding any of 10MB of data on a 1TB hard drive for a given number of randomly sampled sectors. Smaller probabilities indicate higher accuracy.

Table 2: Probability of not finding various amounts of data when sampling 10,000 disk sectors randomly. Smaller probabilities indicate higher accuracy.



Part 2: Can we classify files based on a sector?

A file 30K consists of 60 sectors:



Many file types have characteristics headers and footer:

	header	footer
HTML	<code><html></code>	<code></html></code>
JPEG	<code><FF><D8><FF><E0></code> <code><00><10>JFIF<00></code>	<code><FF><D9></code>
ZIP	<code>PK<03><0D></code>	<code><00><00><00><00></code>

But what about the file in the middle?



JPEGs:

Most FFs are followed by 00 due to “byte stuffing.”

```
Terminal — emacs — 70x27
87654321 0011 2233 4455 6677 8899 aabb ccdd eeff 0123456789abcdef
00006a20: 6b4c cd62 54a0 b214 52ff 0074 ba4f 4622 kL.bT...R..t.0F"
00006a30: d1bf bf4c 67c4 aa2a 4a91 036f f3b3 7ddc ...Lg..*J..o..}.
00006a40: 98d5 f078 7f28 d327 340d a2f2 c916 da4f ...x.(.'4.....0
00006a50: aefa 0cbc e9a6 a580 4b20 952c 17d2 7a09 .....K .,..z.
00006a60: 377b 097c 7395 b7e4 c661 730c 447f 9b5a 7{.ls....as.D..Z
00006a70: 7675 e9d1 e14a 81a8 26a2 2948 93bc 4749 vu...J..&.)H..GI
00006a80: 94fd 8d3f fce2 4a13 e529 2b64 8f31 b961 ...?..J..)+d.1.a
00006a90: 368b 827f 677e 7a64 9a62 60f9 9826 c4e0 6...g~zd.b`..&..
00006aa0: b65e bfa9 97fc 5aa9 6a94 626a 602e 4ac7 .^....Z.j.bj`.J.
00006ab0: 9cb1 0311 3d9d 3e33 e941 482e caf2 8676 .....=>3.AH....v
00006ac0: 240d 43ae ce27 a39e 98d3 f14a 6a23 116a $.C..'.....Jj#.j
00006ad0: af80 dffc 1867 58be 0eaa a9a9 b29f 3331 .....gX.....31
00006ae0: 20b1 9da6 46d3 eb6d 4846 774c 1870 4c98 ...F..mHFwL.pL.
00006af0: 60fd 0f7d 8382 2f04 e2a9 e314 d982 5947 `..}..../.....YG
00006b00: 11ef bef1 7df3 9c6a f0ab 289d 2d99 b6fb .....}..j..(-...
00006b10: ff00 9b6d a903 35aa 8b3c 8014 9240 6006 ..m..5..<...@`.
00006b20: cece 5c3b 9f4d af7f 8934 44d8 bd10 4044 ..\;.M...4D...@D
00006b30: 0124 bd6e b80d 61ff 001d 388c 8b74 aaef .$.n..a...8..t..
00006b40: 32f9 3010 c487 a6fa 681a 4a23 4a8a 5441 2.0.....h.J#J.TA
00006b50: 5b00 3e19 7762 443b 1376 07a1 96c6 5553 [.>.wbD;.v....US
00006b60: 4bbc 285a 7e57 393d e521 e8ce b48a c99a K.(Z~#9=.!.....
00006b70: 69aa 9129 bdab 0361 ba5b 6c36 418d 3e85 i..)...a.[16A.>.
00006b80: 2c2b 5fc4 55c2 162e 0a60 1209 2144 5887 ,+_..U....`...!DX.
00006b90: 20a4 3055 81c3 a566 799d 84b2 1493 28ac .0U...fy.....C.
-:---F1 iStock Privacy.jpg 8% L1714 (Hexl)---8:37PM-----
Mark saved where search started
```



This works!

We identify the *content* of a 160GB iPod in 118 seconds.

Identifiable:

- Blank sectors
- JPEGs
- Encrypted data
- HTML



Report:

- Audio Data Reported by iTunes: 2.42GB
- MP3 files reported by file system: 2.39GB
- Estimated MP3 usage:
 - 2.71GB (1.70%) with 5,000 random samples
 - 2.49GB (1.56%) with 10,000 random samples



Sampling took 118 seconds.

Work to date:

Publications:

- Roussev, Vassil, and Garfinkel, Simson, File Classification Fragment---The Case for Specialized Approaches, Systematic Approaches to Digital Forensics Engineering (IEEE/SADFE 2009), Oakland, California.
- Farrell, P., Garfinkel, S., White, D. Practical Applications of Bloom filters to the NIST RDS and hard drive triage, Annual Computer Security Applications Conference 2008, Anaheim, California, December 2008.

Work in progress:

- Alex Nelson (PhD Candidate, UCSC) summer project
- Using “Hamming,” our 1100-core cluster for novel SD algorithms.
- Similarity Metric



In summary: Statistical disk sampling is a new, powerful technique

We can:

- Rapidly determine if a disk was properly wiped.
- Identify the percentage of:
 - *JPEGs*
 - *MPEGs*
 - *Compressed Data*
 - *Encrypted or Random Data*
- Possible Applications:
 - *Healthcare*
 - *End-of-life auditing*
 - *Privacy Protection*
 - *Boarder Crossing*

Questions?

slgarfin@nps.edu

<http://domex.nps.edu/deep>

