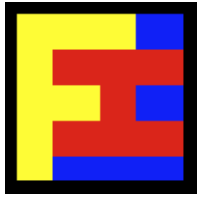


FOUNTAINHEAD

“Math for Big Data”

~ Big Data, Small Data, No Data ~

Baruch College  
Thursday 26<sup>th</sup> January 2017



FOUNTAINHEAD

# Three Talks in One!

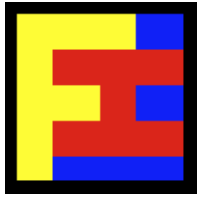
This talk is really three talks in one:

1. Complexity, Fallibility and Finance.
2. Big Data in Finance (BDiF). The MFE course.
3. Mathematics for Big Data, with examples.



FOUNTAINHEAD

# #1 Complexity, Fallibility and Finance

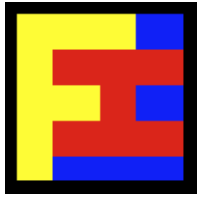


FOUNTAINHEAD

# Complexity

Pretty much any problem can be classified as one of four problem types:

1. Simple. Easily knowable.
2. Complicated. Not simple, but still knowable and predictable.
3. Complex. Not fully knowable, limited predictability.
4. Chaotic. Neither knowable nor predictable.



FOUNTAINHEAD

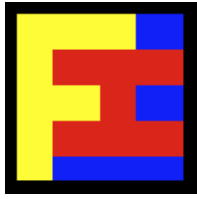
# Fallibility

There are only two reasons for making mistakes:

1. Ignorance. Don't have the knowledge, experience and skills.
2. Incompetence. Have them, but misapply them.

Also:

- (And, maybe, "Act of God". Though, surprisingly, that argument doesn't work very often!)
- (Test, test, test! Oh, libc let me down! Won't fly.)

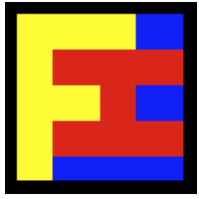


FOUNTAINHEAD

# Finance

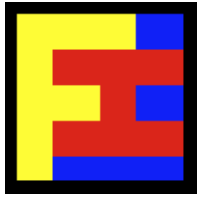
How is all this relevant to what we do outside the classroom?

- In finance, we have simple, complicated and complex problems.
- The first thing you must do is identify what type of problem you are trying to solve.
- If and when you fail, you must identify why: ignorance, or incompetence?
- Much of what professors do—as teachers—is to help you with these key issues.



FOUNTAINHEAD

## #2 Big Data in Finance Course Outline



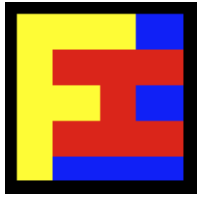
FOUNTAINHEAD

# BDiF Course Logistics

This is the fourth year the course is running and it's been substantially revamped:

- Seven classes.
- Focus is on practical hands-on skills.
- Three hands-on assignments. (Previously six, plus project!)
- One final exam.
- Revamped based on feedback, technology change, jobs market.



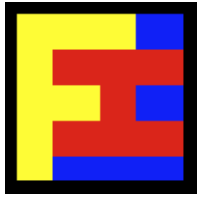


FOUNTAINHEAD

# BDiF Course Topics

BDiF follows a methodology for tackling Big Data projects:

- D1: Data Exploration ~ Know thy data.
- D2: Data Programming ~ Get the plumbing right.
- D3: Data Analysis ~ Use the right analysis tools.
- D4: Data Insights ~ Apply domain expertise to profit from data.



FOUNTAINHEAD

# BDiF Course Data

BDiF will expose you to all the key characteristics of Big Data:

- V1: Volume.
- V2: Velocity.
- V3: Value.
- V4: Variety.
- V5: Veracity.

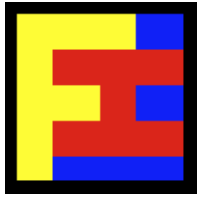


FOUNTAINHEAD

# BDiF Course Models

With BDiF you will tackle various types of Big Data models:

- M1: Big Data in → Small Data out. E.g. CVA, VaR.
- M2: Small Data in → Big Data out. E.g. Bitcoin.
- M3: Big Data in → Big Data out. E.g. Sorting.
- M4: No Data in → Big Data out. E.g. Theoretical models.
- M5: Small Data in → Big Data out over time. E.g. Bitcoin.
- M6: Streaming Data in → Streaming Data out. E.g. Real-time.

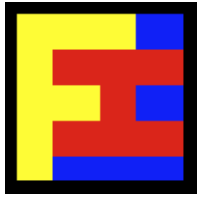


FOUNTAINHEAD

# BDiF Course Skills

Students who complete BDiF will walk away with a number of useful (and bankable!) hands-on skills, such as:

- Spark and no SQL.
- Parallel programming and running at scale.
- Graphs over Big Data.
- Data visualization.
- CVA, Bitcoin, Pattern Matching (with & without machine learning).
- Programming patterns. Big Data patterns.

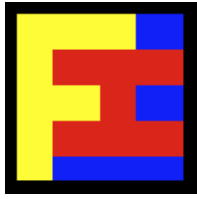


FOUNTAINHEAD

## BDiF Big Data

BDiF: Will expose you to real-world financial data:

- Absolute need to get your hands dirty and work with real financial Big Data.
- Know the tools.
- Need to know what tool to apply to what problem. Match capabilities to requirements. (Capability matrix.)
- All real data sets are living, breathing things. None are static.

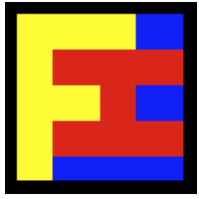


FOUNTAINHEAD

# BDiF Programming

BDiF: What does it mean to be a good programmer?

- Understand the problem and domain.
- Know what users want.
- Iterate to a good solution.
- Know the language.
- Know the tools (Github, etc.).
- Build a development toolchain.
- Have a development methodology (story about John O'Leary).



FOUNTAINHEAD

# BDiF Portfolio of Projects

BDiF: Why have a personal portfolio of Big Data projects?

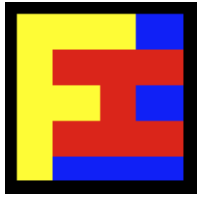
- There is a world of difference between talking about something and showing something.
- Show your boss - get a raise!
- Switching jobs (either internally or somewhere else).
- Starting points for more ambitious projects.
- Tip: extend your portfolio → contribute to open source projects.



FOUNTAINHEAD

# #3 Mathematics for Big Data





FOUNTAINHEAD

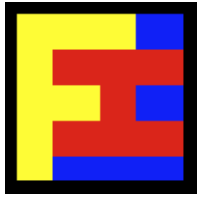
# What is Big Data?

Informal definition:

- End user: anything that doesn't fit in Excel.
- Technologist: anything for which the first thought is "Hadoop".

More formally:

- Anything that is "Big" in one or more of the "V"-axes.



FOUNTAINHEAD

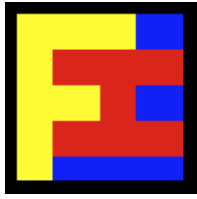
## Data on Big Data

Some Big Data data points (just a snapshot as of 2014):

- 1 TB: Structured trading data collected by the NYSE each day.
- 90% of stored data generated in the last 2 years [IBM].
- MapReduce: 1 PB of 100-byte records sorted in 33 min versus 6 hours in 2008.
- \$600 hard drive will store all the world's music [McKinsey].

For comparison:

- Entire works of William Shakespeare is about 5 MB of text data.
- Human brain holds around 1.25 TB.
- Human brain performs at roughly 100 teraflops.
- 340,000 people working on Big Data in 2018 [McKinsey].

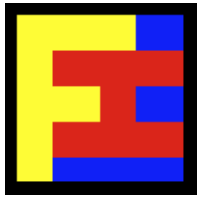


FOUNTAINHEAD

# Why Math for Big Data?

Why is thinking mathematically about Big Data important?

- Gives us answers before we begin.
- Allows formalization of the problem and the data. Data models.
- Allows validation of the models.
- Points to underlying processes (power outages  $\rightarrow$  power laws).
- Provides a big chest of useful tools.
- Helps with data selection.
- Helps with data representation.
- Counterbalances human tendency to apophenia.



FOUNTAINHEAD

# Math for Big Data ~ A Survey

Where is mathematics currently being best applied to Big Data?

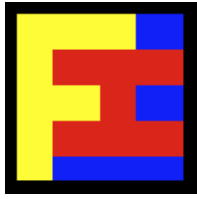
- Benford's law. Fraud.
- Power laws (80-20).
- Bayesian statistics.
- Optimization.
- Machine learning and AI. Why did Google give away TensorFlow?
- Transforms (Fourier, Laplace, etc.).
- Queuing and network theorems.
- Graphs.



FOUNTAINHEAD

V1: Volume

Sparse Sampling



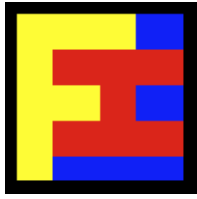
FOUNTAINHEAD

# V1: Sparse Sampling

Definition:

“Via optimized sampling, a signal can be recovered from fewer samples than required by the Shannon-Nyquist theorem.”

Also known as “Compressed sensing”, “Compressive sensing”, or “Compressive sampling”.

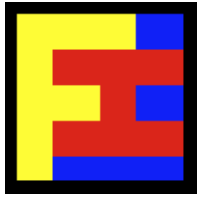


FOUNTAINHEAD

# V1: Sparse Sampling

Key characteristics of sparse sampling:

- Shannon-Nyquist sampling theorem: must sample at least twice as fast as the highest frequency present in the signal.
- Sparsity: Information rate may be much smaller than suggested by its bandwidth. Sparse in the sense its information is compressible.
- Incoherence: Data sparse in their own domain may be spread out in the domain in which they are acquired.
- Non-intuitive!



FOUNTAINHEAD

# V1: Sparse Sampling

Key benefits of sparse sampling:

- Selective and efficient data sampling can capture most of the information in a signal and condense it into a small amount of data.
- Non-adaptive. Signal independent (one-time set up).
- No need to comprehend the whole signal.
- Can reconstruct the full-length signal from the small amount of collected data.





FOUNTAINHEAD

## V1: Sparse Sampling

The sensing problem:

$$y_k = \langle f, \varphi_k \rangle$$

$$k = 1, \dots, m$$

The signal is obtained by linear functionals to record values. Correlate the object we wish to acquire with the waveforms ( $\varphi$ ).



FOUNTAINHEAD

## V1: Sparse Sampling

Sparcity:

$$f(t) = \sum_{i=1}^n x_i \psi_i(t)$$

When a signal has a sparse expansion, you can discard terms with small coefficients without much information loss.



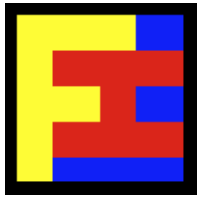
FOUNTAINHEAD

## V1: Sparse Sampling

Incoherence:

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k, j \leq n} \left| \langle \varphi_k, \psi_j \rangle \right|$$

Coherence measures the largest correlation between any two elements of the sensing basis ( $\phi$ ) and representation basis ( $\psi$ ). If they contain correlated elements, the coherence is large.



FOUNTAINHEAD

## V1: Sparse Sampling

Undersampling and sparse signal recovery:

$$y_k = \langle f, \varphi_k \rangle$$

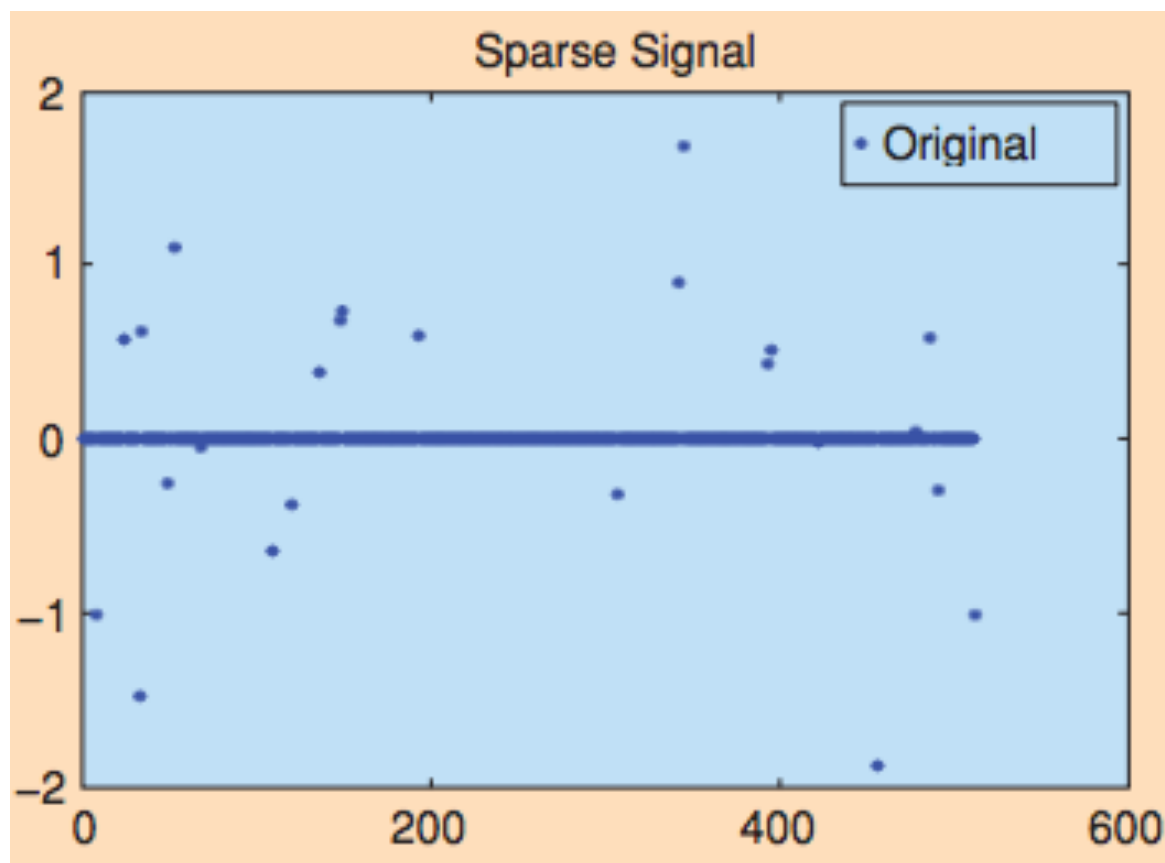
$$k \in M, M \subset \{1, \dots, n\}, m < n$$

Ideally, would like to measure all the  $n$  coefficients of  $f$ , but we only get to observe a subset of these when collecting data. Can recover signal by optimization.



# FOUNTAINHEAD

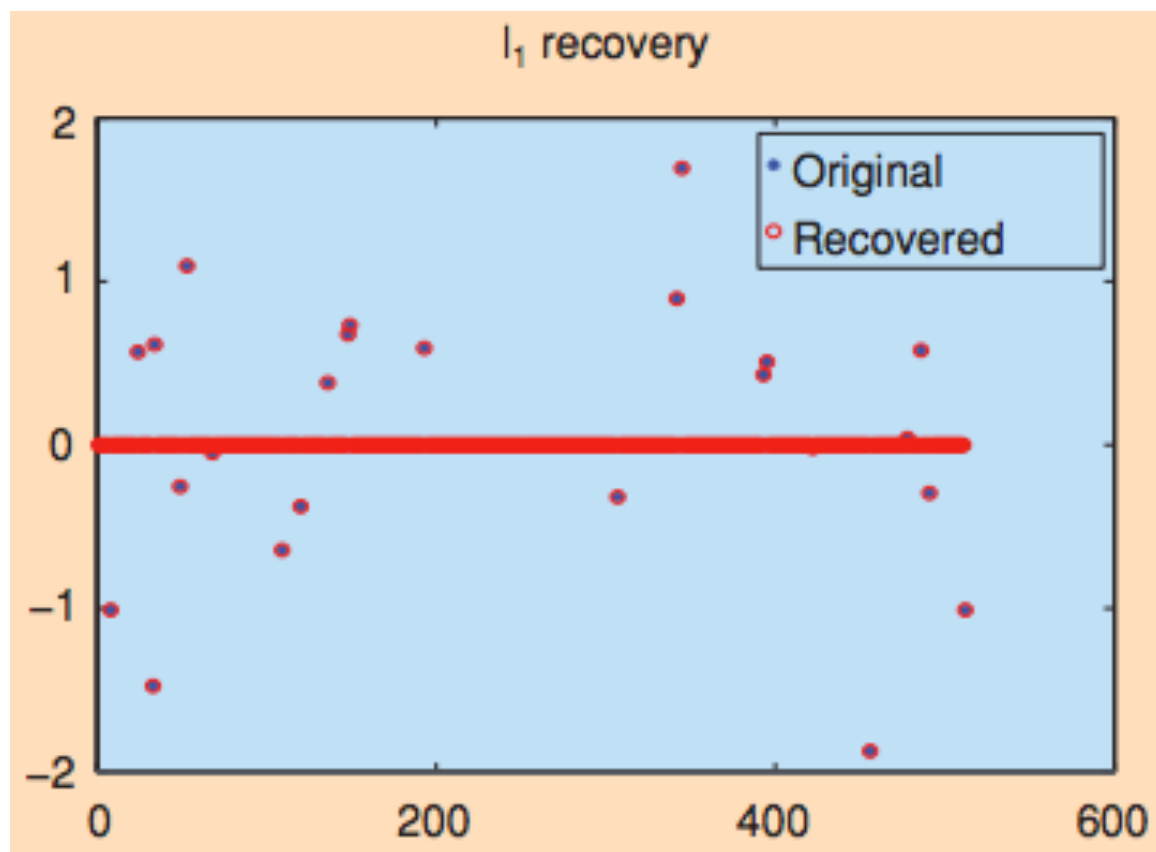
## V1: Sparse Sampling

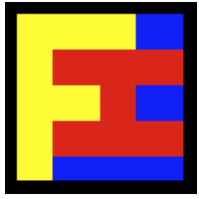




# FOUNTAINHEAD

## V1: Sparse Sampling





FOUNTAINHEAD

V2: Velocity

Bitcoin Transactions



FOUNTAINHEAD

## V2: Bitcoin

Definition:

“Bitcoin is a P2P distributed value storage and transactional system centered on a public ledger called the blockchain.”

Bitcoin is part of a wider group of emerging cryptocurrencies.





FOUNTAINHEAD

## V2: Bitcoin Stats

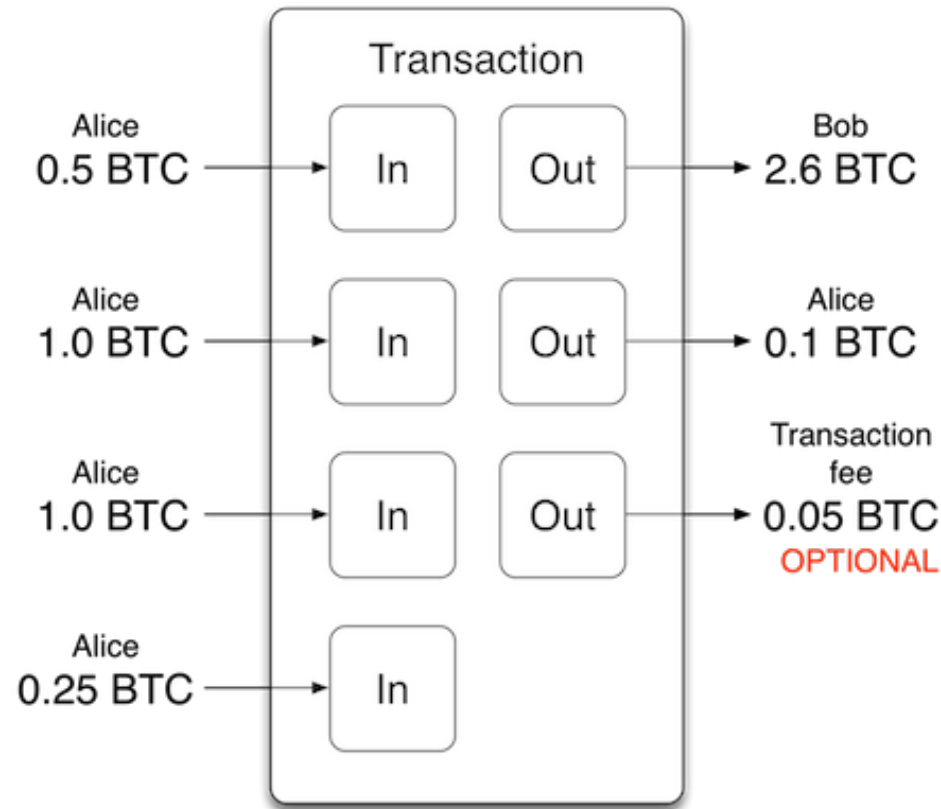
Some bitcoin stats (as of Jan 27 2016):

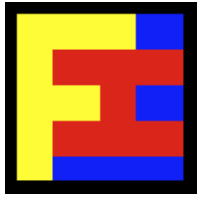
- Market cap: \$5.9 billion.
- BTC/USD: \$390.
- Network hashing rate: 967,800,000 Thash/sec. (>> Top500)
- Daily transactions: 225,000.
- Daily transaction volume: \$150 million.
- Largest recent transaction: 135,800 BTC (~\$53 million)
- Average transaction confirmation time: ~8.9 minutes.



# FOUNTAINHEAD

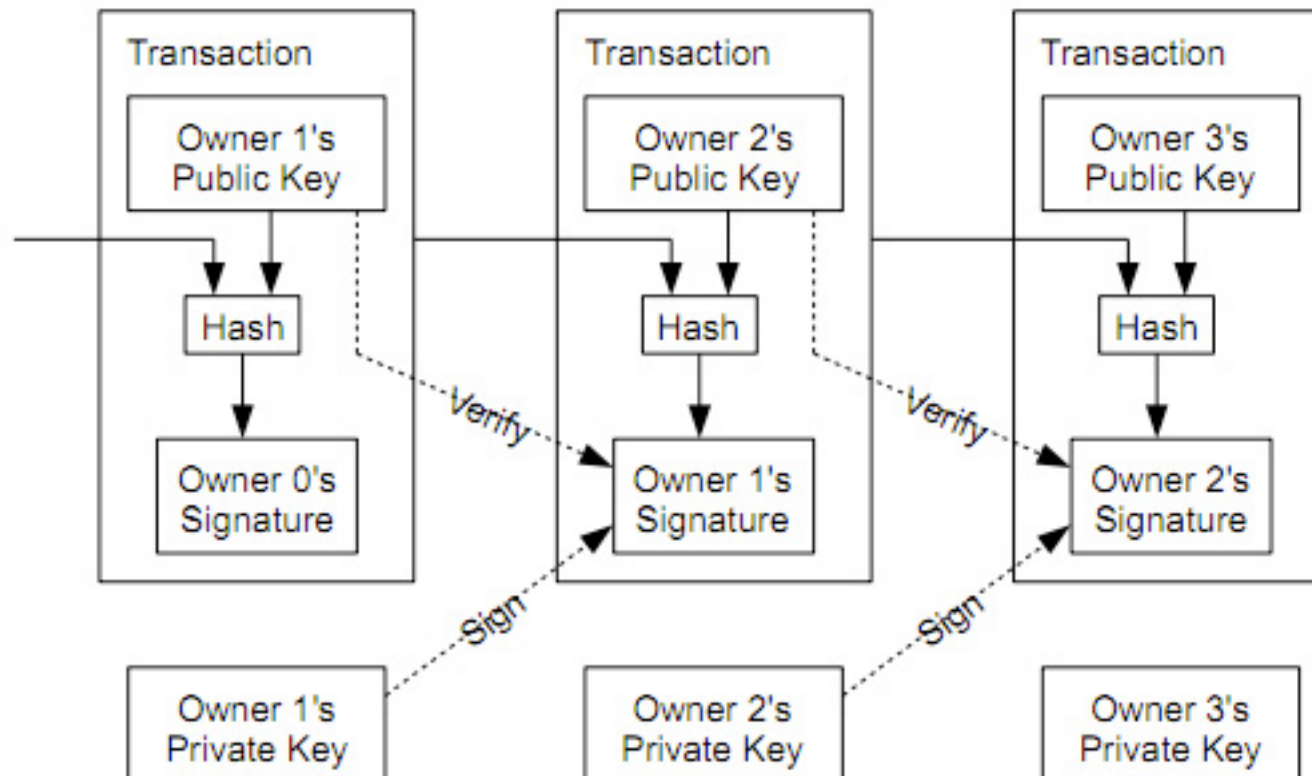
## V2: Bitcoin Transaction

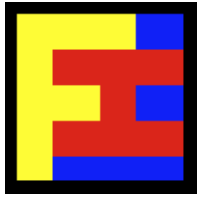




# FOUNTAINHEAD

## V2: Bitcoin Blockchain





FOUNTAINHEAD

## V2: Bitcoin

Bitcoin as a currency:

- Non-fiat. Like gold. 1oz gold = 1 toga + 1 belt + 1 pair of shoes.
- Fixed issuance. Prices float. Deflationary.
- 21 million bitcoins. Each bitcoin divisible into 100 million satoshis.

Bitcoin as a technology:

- Distributed.
- Consensus through proof-of-work. No trust.
- 1000 bytes per transaction. 1MB per block. 1 block every 10 mins.



FOUNTAINHEAD

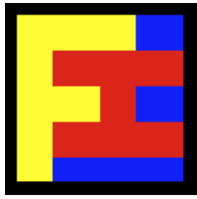
## V2: Bitcoin as Currency

Enough units of bitcoin to go around? (Global currency)

- Total spendable units:  $13,500,000 \times 100,000,000 = 13.5 \times 10^{14}$ .
- Evenly divide among world's population of 7.25 billion.
- Each person would have around 186,000 satoshis.

Compare with the U.S. dollar. (National currency)

- Total spendable units: \$1.29 trillion in circulation.  $1.29 \times 10^{14}$ .
- Evenly divide among world's population of 7.25 billion.
- Each person would have around 17,800 cents. (100,000 all ccys)



FOUNTAINHEAD

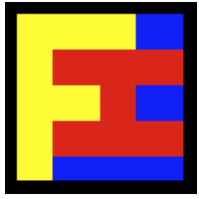
## V2: Bitcoin as Technology

Theoretical transactional bandwidth:

- Every 10 mins, 1MB block added to blockchain.
- 1MB block / 200 bytes per transaction every 10 mins.
- Theoretical transaction rate of ~10 tx per second.
- More detailed analysis (network/queuing) gives ~7 tx per second.

Compare with other payment networks:

- VISA 2,000 tx per second. 47,000 tx per second peak rate.
- PayPal average tx rate is about 115 tx per second.
- Velocity across all transaction processors?



FOUNTAINHEAD

V3: Value

Half-Life of Data



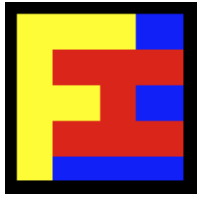
FOUNTAINHEAD

## V3: Half-Life of Data

Definition:

“The half-life of any piece of data is the time it takes for its economic value to half.”

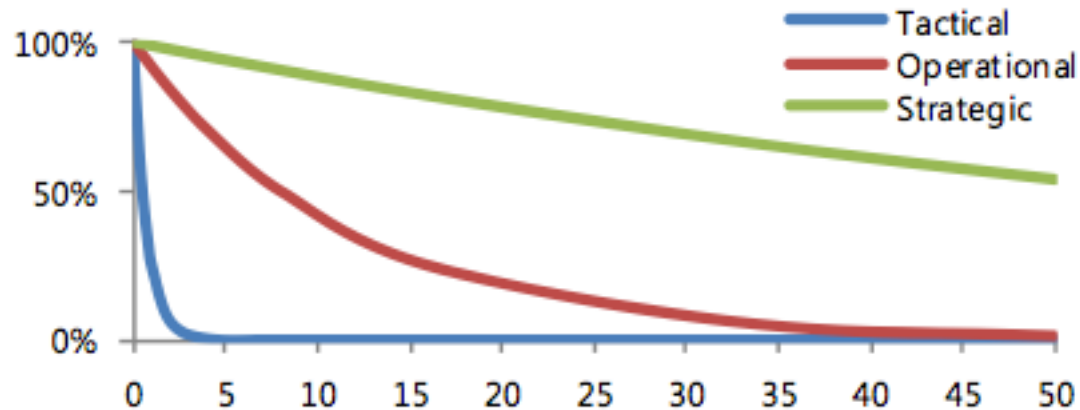




# FOUNTAINHEAD

## V3: Half-Life of Data

### THE HALF LIFE OF DATA





FOUNTAINHEAD

## V3: Half-Life of Data

From a report by Nucleus Research using data from 47 companies:

- Tactical: 30 minutes or less.
- Operational: 1 day. Natural cadence of business operations.
- Strategic: 90 days.

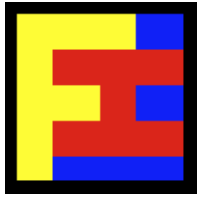


FOUNTAINHEAD

## V3: Half-Life of Data

Half-life formula:

$$V(t) = V_0 \left( \frac{1}{2} \right)^{t/t_{1/2}}$$



FOUNTAINHEAD

## V3: Half-Life of Data

So why is half-life important? Why is it useful?

- Makes you focus on the most valuable data.
- Provides criteria for when to throw away, or ignore, data.
- Helps when making tradeoff between accuracy and speed.

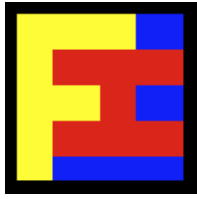
Incidentally, a weighting function for time series data is just another alternative expression of the half-life concept.



FOUNTAINHEAD

V4: Variety

Curse of Dimensionality

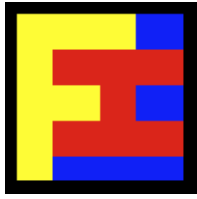


FOUNTAINHEAD

## V4: Curse of Dimensionality

Definition:

“The *curse of dimensionality* refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience.”

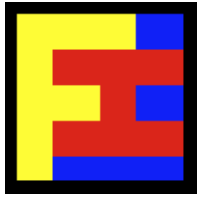


FOUNTAINHEAD

# V4: Curse of Dimensionality

Curse of dimensionality in a nutshell:

- When dimensionality increases, volume of space increases dramatically to make data sparse.
- Sparsity is a problem for any method that requires statistical significance.
- To make results statistically significant, data size grows exponentially with the dimensionality.
- A problem with the data? Or a problem with the algorithms?



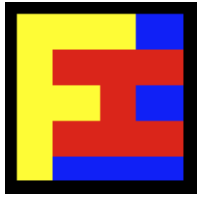
FOUNTAINHEAD

## V4: Curse of Dimensionality

Areas in which the curse of dimensionality is problematic:

- Combinatorics.
- Sampling.
- Optimization.
- Machine learning.
- Bayesian statistics.
- Distance functions.
- K-nearest neighbor classification.
- ... and many more!





FOUNTAINHEAD

# V4: Curse of Dimensionality

Strategies to deal with the curse of dimensionality:

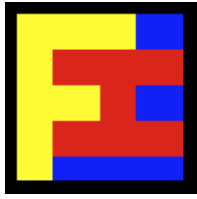
- Pre-processing to a lower dimensional form. c.f. WWII Spitfire.
- Dimension reduction. E.g. principal component analysis.
- Better data organization. Make data less sparse along certain axes.
- Better algorithms.



FOUNTAINHEAD

## V5: Veracity

What is Randomness?

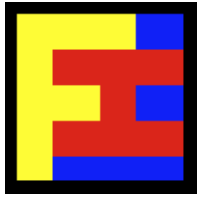


FOUNTAINHEAD

## V5: Randomness

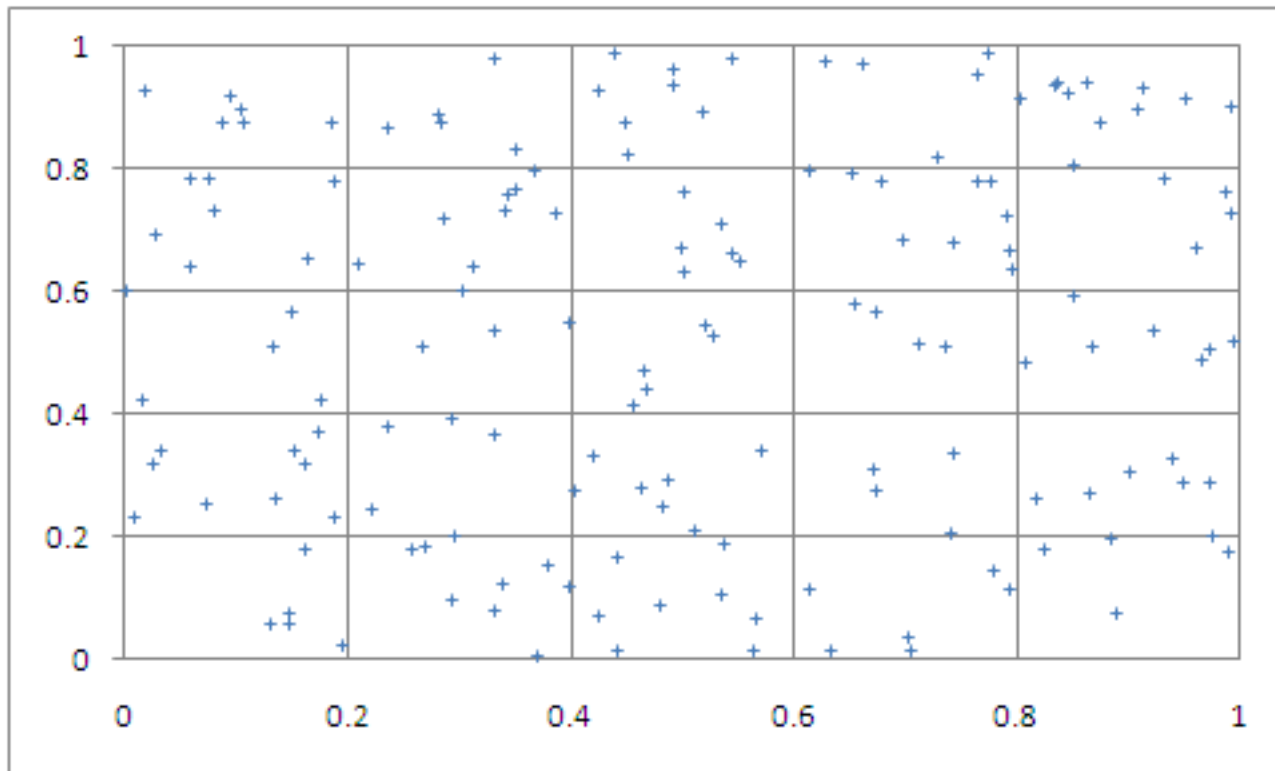
Definition:

“Randomness means lack of pattern or predictability in events. Randomness suggests a non-order or non-coherence in a sequence of symbols or steps, such that there is no intelligible pattern or combination.”



FOUNTAINHEAD

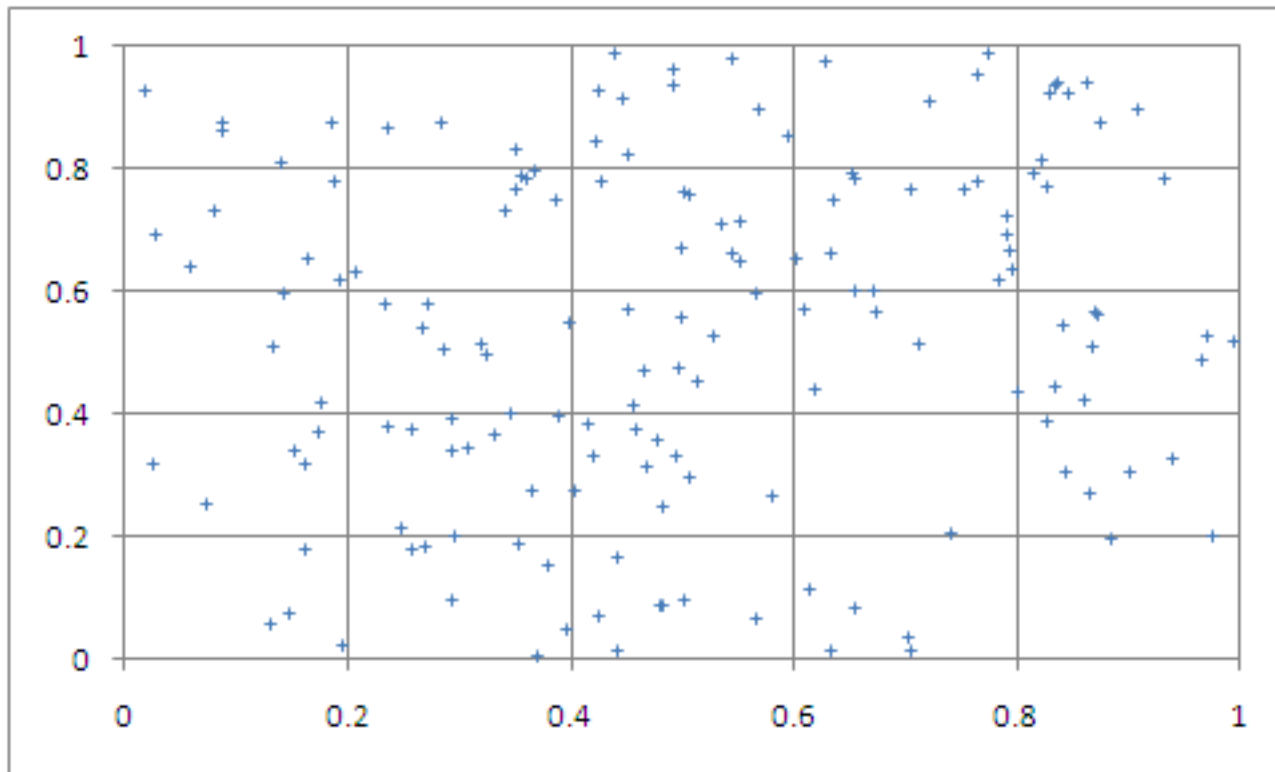
## V5: Randomness – Chart 1

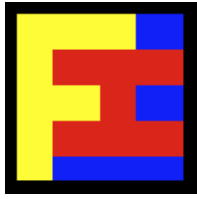




FOUNTAINHEAD

## V5: Randomness – Chart 2



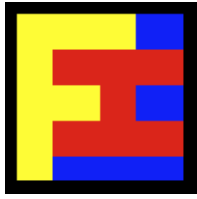


FOUNTAINHEAD

## V5: Randomness

This is how charts 1 and 2 were generated:

- Produce 158 random points  $[a(n), b(n)]$ ,  $n=1, \dots, 158$
- Produce 158 random deviates  $u(n), v(n)$ ,  $n=1, \dots, 158$
- Define  $x(n)$  as follows for  $n > 1$ : if  $u(n) < r$ , then  $x(n) = a(n)$ , else  $x(n) = s \cdot v(n) \cdot a(n) + [1 - s \cdot v(n)] \cdot x(n-1)$ , with  $x(1) = a(1)$
- Define  $y(n)$  as follows for  $n > 1$ : if  $u(n) < r$ , then  $y(n) = b(n)$ , else  $y(n) = s \cdot v(n) \cdot b(n) + [1 - s \cdot v(n)] \cdot y(n-1)$ , with  $y(1) = b(1)$
- Chart 1:  $x(n) = a(n)$ ,  $y(n) = b(n)$
- Chart 2:  $r = 0.5$ ,  $s = 0.5$



FOUNTAINHEAD

## V5: Randomness

Towards a test of randomness:

- Nearest neighbors. But what about points at the boundaries?
- Correlation. Both charts show zero correlation!
- Number of points in sub-squares. Chi-square test. Sparseness?
- Compare with random simulation, then apply tests (e.g. k-th neighbors). “If it walks like a duck and quacks like a duck ...”
- Large number of points in high dimensions. Curse of dimensionality. Apophenia.



FOUNTAINHEAD

## V5: Randomness

Some extra wrinkles:

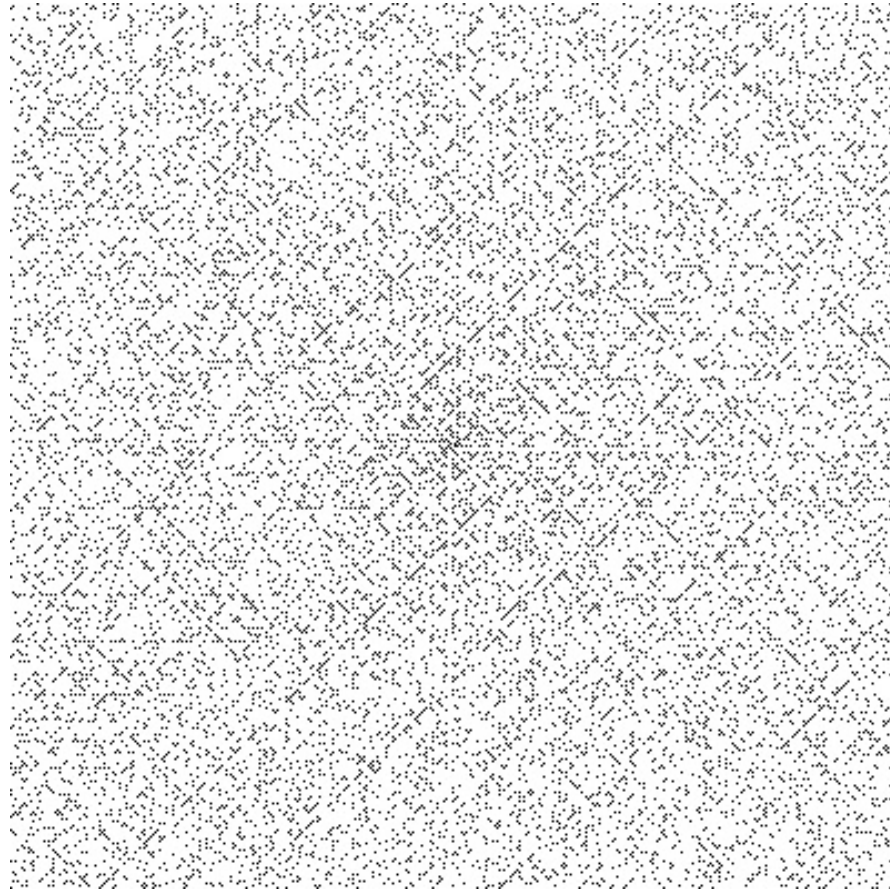
- Is the number 6 random?
- Scale. HHHHH unlikely in a short sequence. Certain in a long one.
- Local randomness.
- No definitive tests. NIST tests made up from 15 separate tests.
- Plot and then view. Ulam Spiral. Apophenia.





FOUNTAINHEAD

## V5: Randomness





FOUNTAINHEAD

## V5: Apopphenia

