CrySP
Jeremy Clark & Urs Hengartner
University of Waterloo

On the use of financial data as a random beacon

Overview

- We examine using the closing prices of stocks as a source for a true random seeds
- This approach has been used in binding E2E elections
- We conservatively estimate that over one trading day, the stocks in the Dow Jones have over 200 unpredictable bits
- We find the level of randomness is sufficient

Randomness in elections

- The detection of errors or fraud in elections can be achieved with audits
- In traditional elections, precincts can be randomly selected for manual recounts
- In end-to-end verifiable (E2E) elections, random challenges can prove the tally is correctly computed from a verifiable set of privacy-preserving receipts
- If the challenges were known in advance, the proof could be faked

Random challenges

- Two systems that require external randomness are Scantegrity II and Punchscan
- Both have run binding elections and both used financial market data for generating a seed
- The seed (or its pseudorandom expansion) is formatted to create challenges

- What properties should a random seed have for E2E elections?
  - Each bit should have a uniform probability of 0 or 1
  - Generated at the appropriate time
  - Appropriate length
  - Generation is observable by anyone
  - A high level of mathematics is tolerable

Price manipulation

- Since the price is determined by trades and anyone can trade, can't anyone manipulate the closing price?
- In theory, yes, but…
- Widely considered to be difficult for liquid stocks on established exchanges
- There is empirical evidence for this
- Barrier options continue to be written, held and traded
- Other complexities: see paper

Method

Financial Model

- Choose a model to represent stock price movements

Historic Data

- Fit historic data to the model to estimate parameters

Monte Carlo Simulations

- Run simulations of price movements forward in time

Entropy Estimation

- Measure the resulting entropy

Extraction

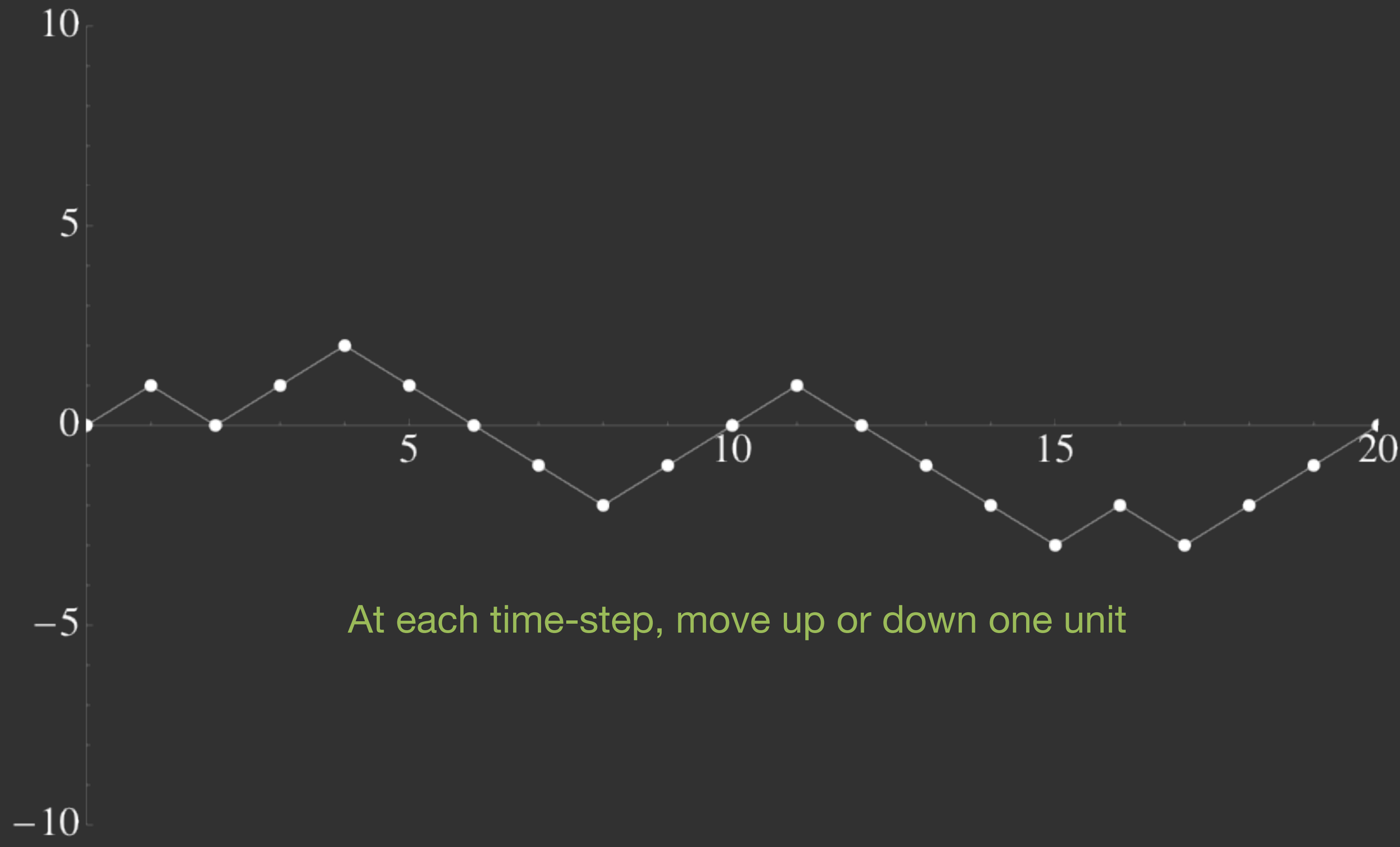
- Determine how to extract random bits from prices

Modeling stock prices

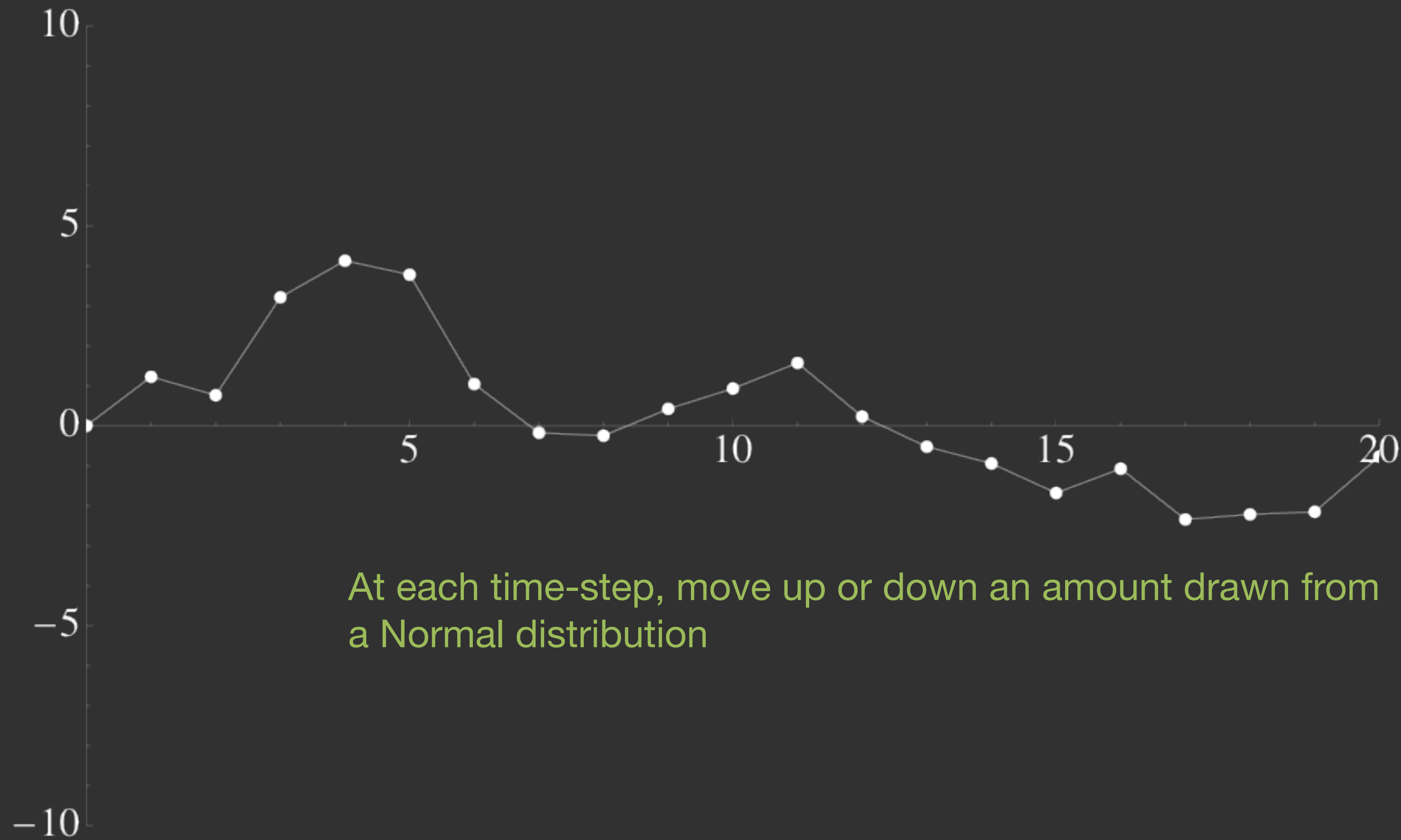
- To estimate the randomness in a closing price, we need to assume a mathematical model holds for stock prices
- These models do not predict prices
- Models are used in real-life by banks to hedge against risky assets

Black-Scholes

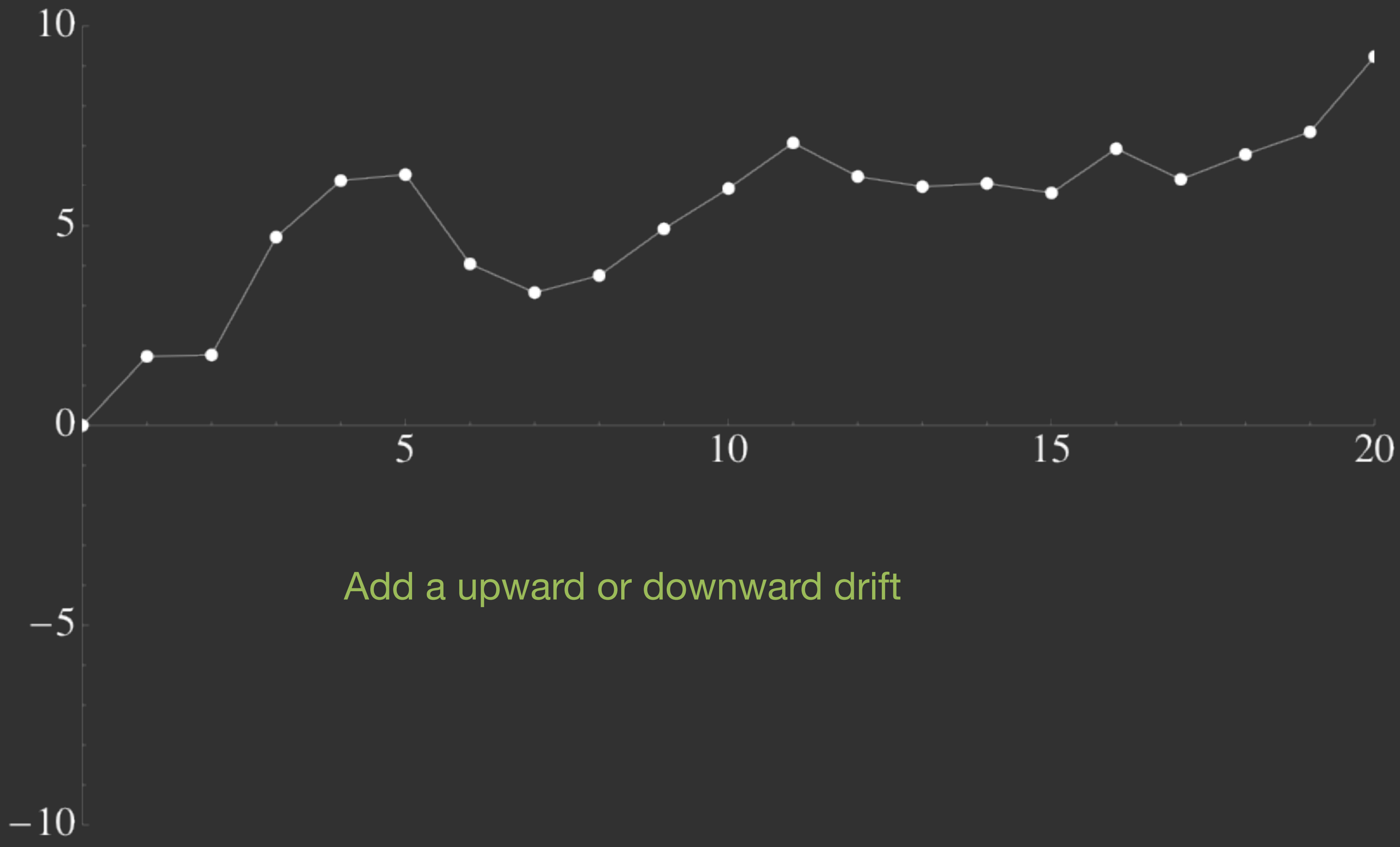
- We use the Black-Scholes model
- This model is now widely considered to under-estimate market volatility: bad for banks when pricing options, good for us in estimating a lower-bound on the randomness in a closing price
- Black-Scholes assumes that stock prices follow a stochastic process called geometric Brownian motion (GBM)



At each time-step, move up or down one unit



At each time-step, move up or down an amount drawn from a Normal distribution



Add a upward or downward drift

Geometric Brownian motion

- If we make it continuous in time, we get:

$$\mathrm{d}S_t = \mu S_t \mathrm{d}t + \sigma S_t \mathrm{d}W_t$$

S_t Stock price at a given time

μ Drift term / rate of return / interest rate

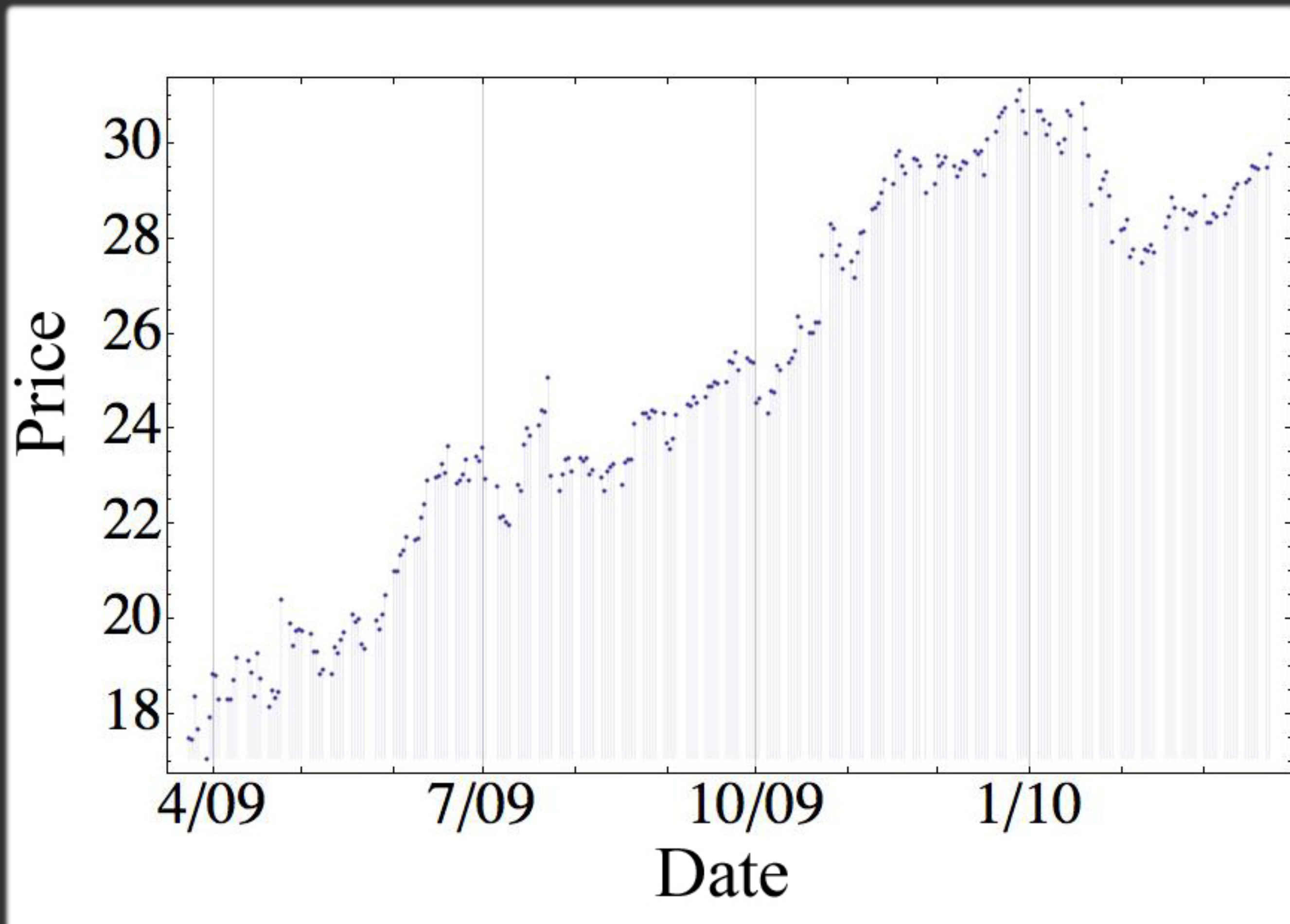
σ Diffusion term / volatility

$\mathrm{d}W_t$ Increment of a Weiner process / stochastic term

Geometric Brownian motion

- With a series of prices for a specific stock, we can estimate its daily drift and diffusion rates
- Example: Microsoft over one year
- From March 23, 2009 (at $17.95) until March 23, 2010 (at $29.88)

MSFT closing prices



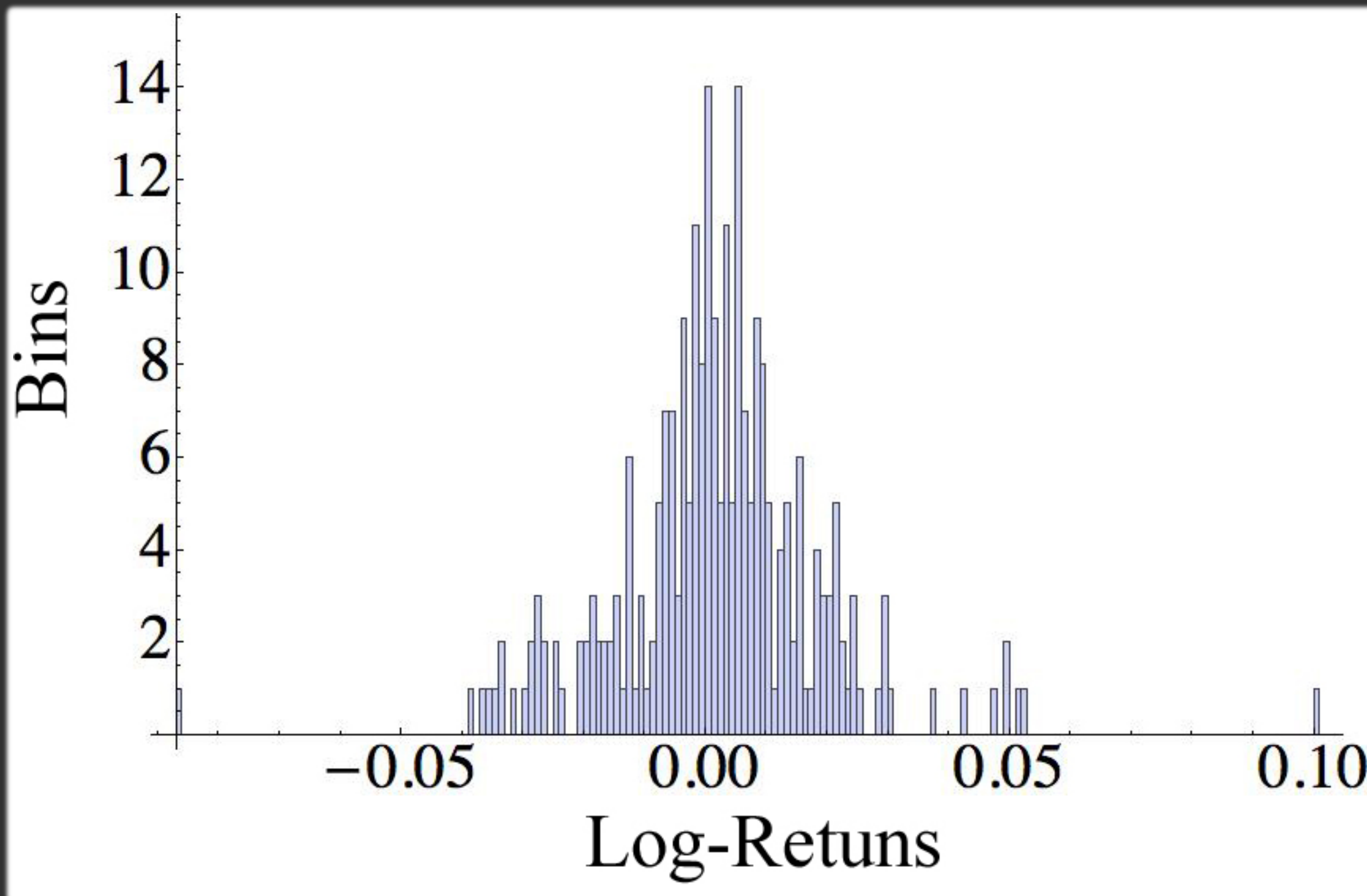
Logarithmic returns

- We are interested in the relative changes in the price, and need to fit it to an exponent
- For each price, we calculate its logarithmic return from the previous price:

$$R_i = \ln\left(\frac{S_{i+1}}{S_i}\right),\ 0 \leq i \leq T-1$$

where T is the number of prices in the period (T=251)

Histogram of log-returns



Estimator for drift/diffusion

- Under GBM, the log-returns should be normally distributed as:

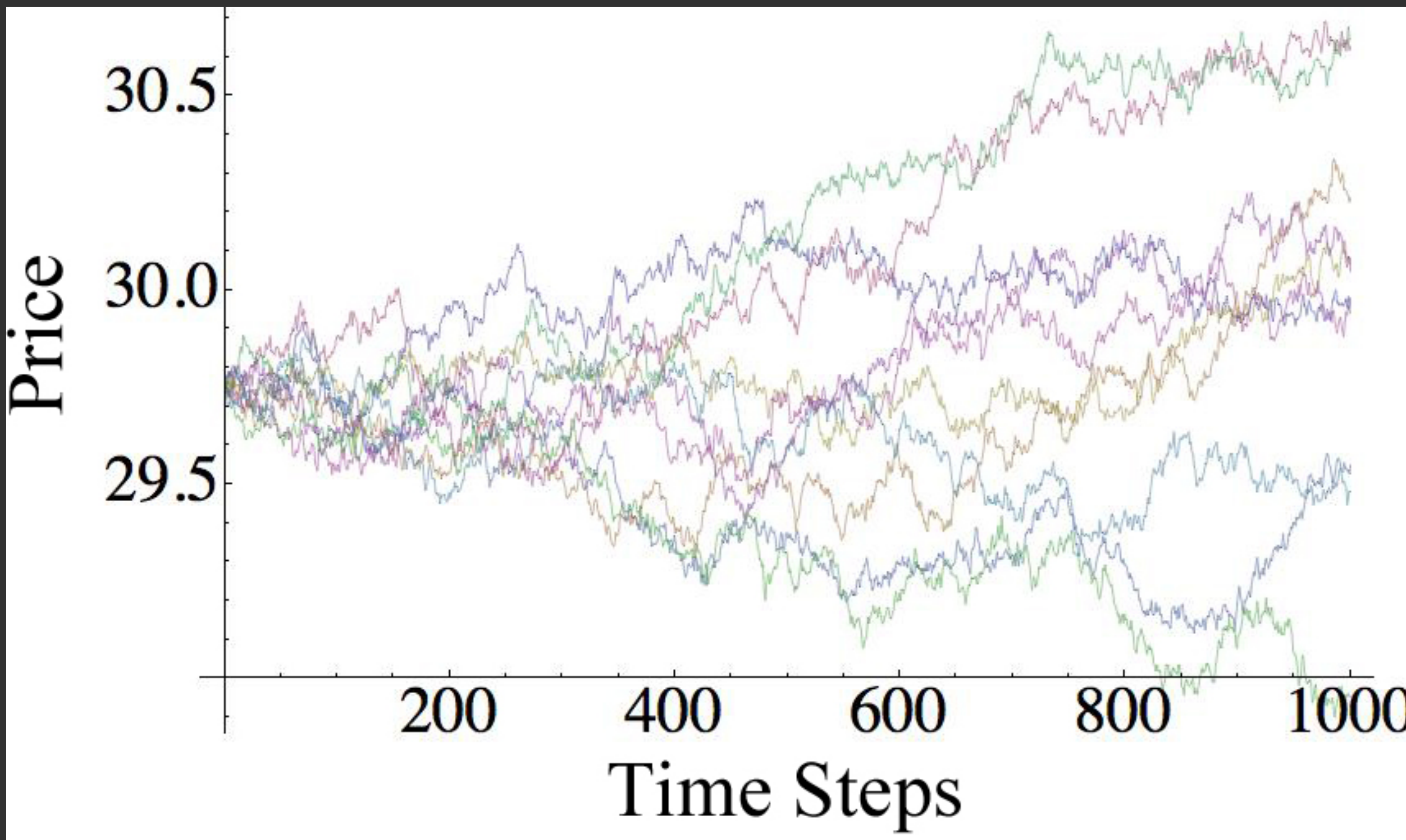
$$R_i \sim \mathrm{N}\left(\left(\mu - \frac{\sigma^2}{2}\right)\Delta t, \sigma^2 \Delta t\right)$$

- We can fit our historic data
- For MSFT during this period, daily drift was 0.23% and daily diffusion was 1.77%.

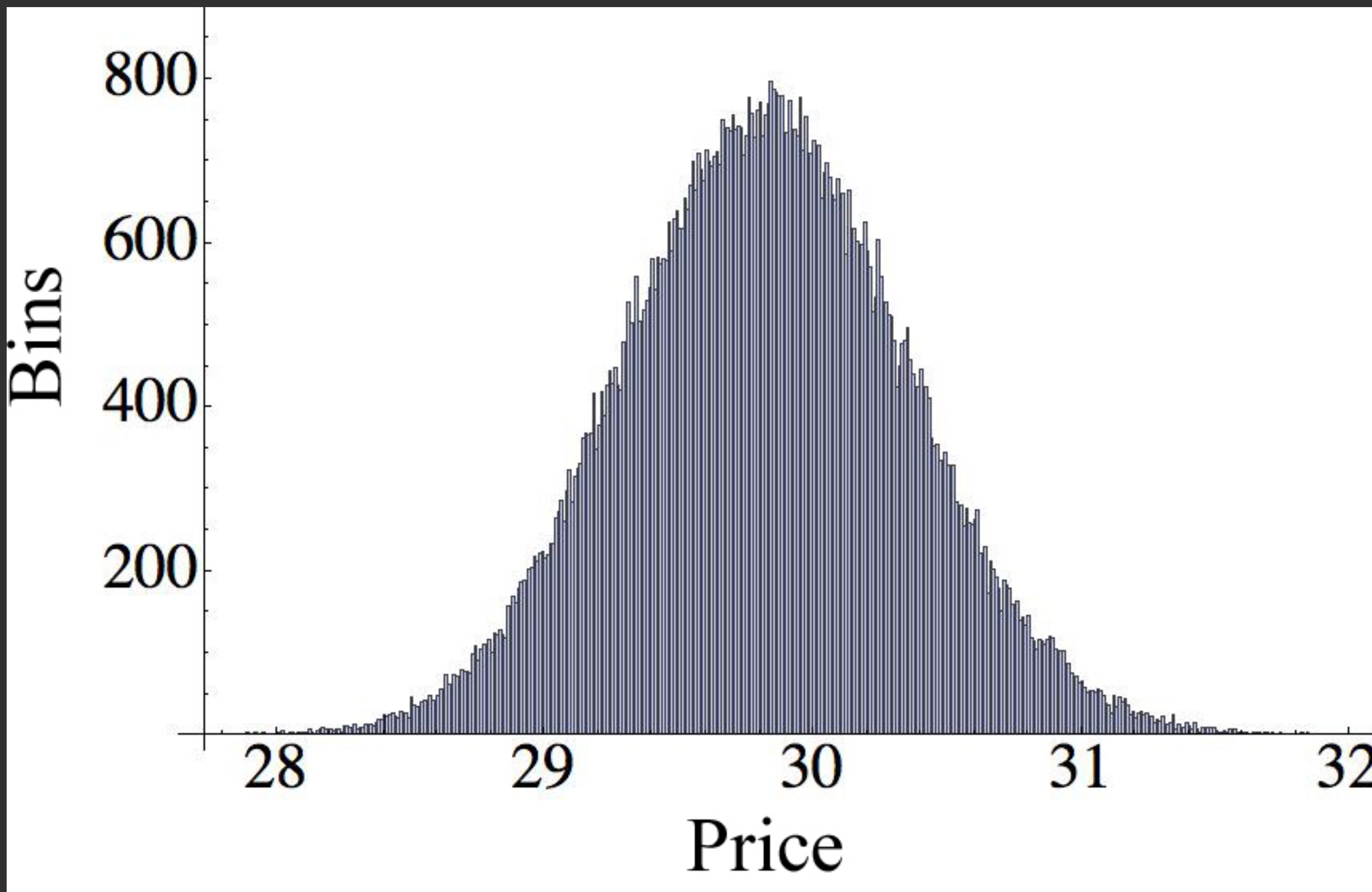
Monte Carlo

- Now that we have estimates for drift and diffusion, we simulate many possible paths for the stock price over the next day
- We round the output price to the nearest cent
- This gives a discrete probability distribution we can use to estimate the randomness
- This approach has some bias: see paper

Monte Carlo simulations



Histogram of outcomes



Entropy

- Randomness can measured: entropy
- A sequence of numbers with *N* bits of (Shannon) entropy contains the same randomness as flipping a coin *N* times
- We can generally extract some these random bits from the sequence but not necessarily all *N* bits
- *M* bits of min-entropy means we can (theoretically) extract $M \leq N$ coin tosses

Entropy Estimation

- Entropy is measured from histogram
- For MSFT over 1 day:
  - 7.76 bits of estimated Shannon entropy
  - 0.02 bits of estimated bias
  - 7.04 bits of estimated min-entropy
- Scantegrity II used the 30 stocks in the Dow Jones

Stock	S_T	μ	σ	$\hat{N}$	$\hat{M}$	$H(\mathcal{P})$	B	$H_B(\mathcal{P})$	$H_\infty(\mathcal{P})$
AA	14.50	0.00338956	0.0354406	386	440	7.72544	0.0022	7.73	6.99
AXP	41.24	0.00512444	0.0365912	1071	1305	9.2823	0.006525	9.29	8.50
BA	72.18	0.00313975	0.0219116	1112	1406	9.34279	0.00703	9.35	8.57
BAC	17.13	0.00455058	0.0468486	588	699	8.37453	0.003495	8.38	7.62
CAT	62.41	0.00352696	0.027272	1173	1540	9.4536	0.0077	9.46	8.69
CSCO	26.64	0.00200486	0.0167037	347	396	7.52981	0.00198	7.53	6.79
CVX	74.77	0.000565046	0.0136131	730	844	8.70798	0.00422	8.71	7.95
DD	38.31	0.0025213	0.0219181	603	751	8.43244	0.003755	8.44	7.69
DIS	34.01	0.00271648	0.0199576	506	596	8.13347	0.00298	8.14	7.39
GE	18.33	0.00264998	0.0239698	335	391	7.50284	0.001955	7.50	6.76
HD	32.59	0.00166033	0.0161739	404	475	7.76791	0.002375	7.77	7.03
HPQ	53.15	0.00234904	0.015783	615	758	8.43501	0.00379	8.44	7.69
IBM	129.37	0.00124652	0.0124436	1121	1460	9.36931	0.0073	9.38	8.60
INTC	22.67	0.0019257	0.0176758	302	352	7.37295	0.00176	7.37	6.63
JNJ	65.36	0.00101973	0.00811278	406	472	7.7723	0.00236	7.77	7.03
JPM	44.58	0.00261719	0.0318482	992	1190	9.18918	0.00595	9.20	8.43
KFT	30.49	0.00134673	0.0129888	314	333	7.35464	0.001665	7.36	6.62
KO	55.30	0.0010976	0.0111199	460	570	7.98678	0.00285	7.99	7.21
MCD	67.35	0.00111279	0.0113681	569	732	8.3043	0.00366	8.31	7.55
MMM	82.35	0.00235099	0.0148201	854	1075	8.97369	0.005375	8.98	8.22
MRK	38.50	0.00162879	0.0166847	486	554	8.05935	0.00277	8.06	7.29
MSFT	29.88	0.0022737	0.0176583	394	449	7.76265	0.002245	7.76	7.04
PFE	17.54	0.00120496	0.01571	216	243	6.82701	0.001215	6.83	6.10
PG	64.53	0.00146004	0.0125241	587	703	8.37914	0.003515	8.38	7.64
T	26.55	0.000357228	0.0121909	251	289	7.05851	0.001445	7.06	6.31
TRV	53.90	0.00154645	0.0188065	734	926	8.7059	0.00463	8.71	7.96
UTX	73.09	0.00232501	0.0159515	835	1015	8.9016	0.005075	8.91	8.14
VZ	30.98	0.000367966	0.0117435	279	320	7.22926	0.0016	7.23	6.48
WMT	55.89	0.000497465	0.010295	431	512	7.89168	0.00256	7.89	7.16
XOM	66.95	0.0000317968	0.012391	604	752	8.41962	0.00376	8.42	7.65

- We also isolated the effect of each parameter on entropy: drift, diffusion, initial price, and elapsed time
- See paper

Correlated stocks

- From chart: MSFT has 7.76 bits and IBM has 9.38 bits
- If we concatenate their prices, do we get 7.76 + 9.38 = 17.14 bits?
- No. The price movements are correlated
- See the paper for modeling correlated stocks

Bottom line

- We estimate the randomness in the DJIA portfolio to have 218 bits of Shannon entropy and 192 bits of min-entropy

Useful form

- Consider taking a set of closing prices and concatenating them together into a large binary string
- Some of the individual bits in this string will be nearly random while others will be almost deterministic
- Can we convert it into a smaller bitstring where each individual bit is uniform random?
- Yes. We require an extractor

Extractors

- Can we just hash it?
- No. A hash function (ideal compression & Merkle-Damgaard) does not make a good extractor [DGHKR'04]
- However we can use a standard cryptographic primitive: block cipher (ideal PRP) in CBC-MAC mode [DGHKR'04]

Producing a seed

- In summary, to make a random seed: take closing prices, concatenate them together, and extract
- This is minimal: seeds rely on only that day and rely fully on the market's randomness
- We present a general protocol for a beacon service provider that offers some additional security properties: see paper

Concluding Remarks

- The approach of using closing prices for post-election audits in E2E elections is sound
- Using a portfolio such as the Dow Jones will produce enough bits for a cryptographically strong seed
- This seed can be used directly or expanded with a PRG

Questions?