

# Cryptocurrency Spy Process book

u1431623 Yintong Shang, u1431587 Haoyang Shi

## Related Work & Acknowledgments

Bitcoin, as the first cryptocurrency, is introduced by a pseudonymous person [Satoshi Nakamoto](#) with the paper titled "[Bitcoin: A Peer-to-Peer Electronic Cash System](#)".

Blockchain.com (<https://www.blockchain.com/explorer>) uses a conventional line chart and treemap to show the overview of cryptocurrencies. It also has a fantastic block explorer to show the information within the blockchain, e.g., the transactions, amount, rewards, and accounts.

The github repo blockchain-etl ([blockchain-etl/bitcoin-etl: ETL scripts for Bitcoin, Litecoin, Dash, Zcash, Doge, Bitcoin Cash. Available in Google BigQuery https://goo.gl/oY5BCQ \(github.com\)](#)) provides us with the right tool to dig through the bulk data and turn it into the format required by the network layout.

Students from UCB created this amazing project which has a network layout similar to our design. We did not, however, manage to find the working demo as the link to the project page is broken. (<https://www.ischool.berkeley.edu/projects/2021/cryptocurrency-visualization>)

## Data Sources and Processing

Transaction data:

[Tracing the 10,000 BTC pizza | Kaggle.](#)

Market share data:

[https://download.data.world/file\\_download/pmohun/complete-historical-cryptocurrency-financial-data/consolidated\\_coin\\_data.csv](https://download.data.world/file_download/pmohun/complete-historical-cryptocurrency-financial-data/consolidated_coin_data.csv)

Price data & volume data:

[Home - Coin Metrics](#)

## Current Design

2022/11/10

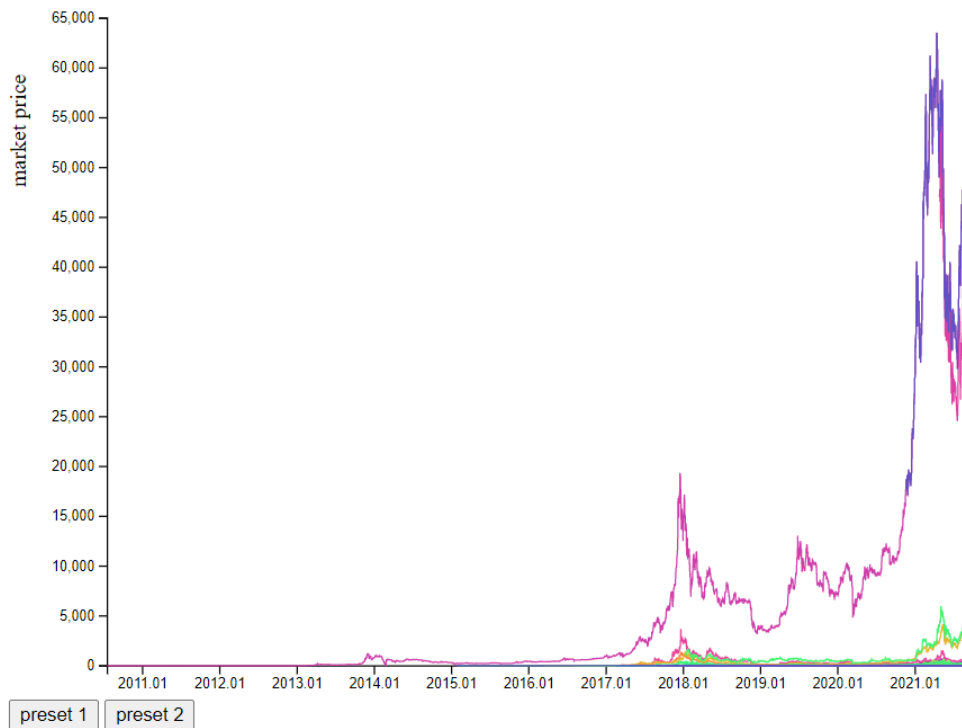
We planned to include views of the treemap, brush, line chart, and transaction network.

So far, our website has two split views overall, left and right. The left view includes a treemap and a force-directed map. The right view includes a line chart, a name-selection checkbox array and a time-selection grid.

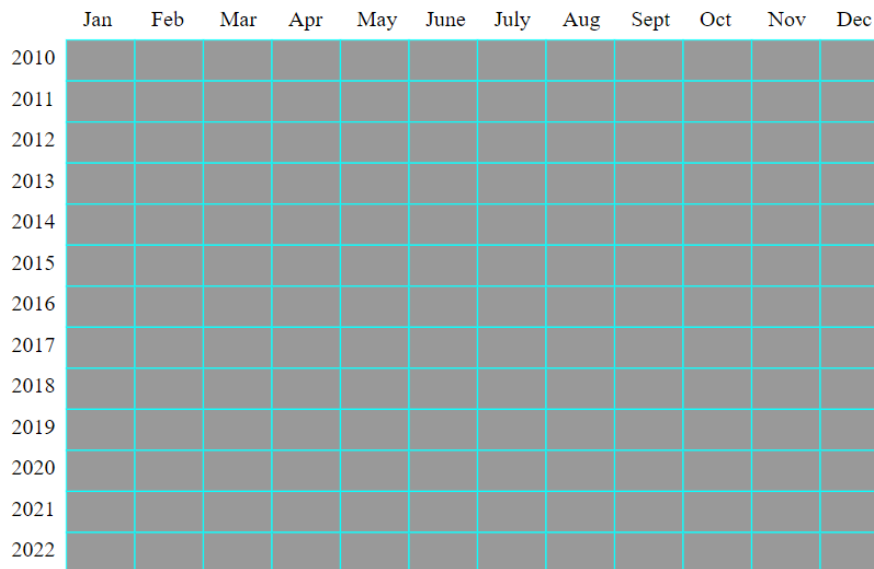
A snapshot of data used for the right view is shown below. (file 'combined\_prices.csv') We use the first three columns now, each of which is, the name of the cryptocurrency, the date, and the price of this currency on that date.

	A	B	C	D	E	F	G	H	I	J	K
1	Currency_Name	Date	Price	Open	High	Low	Vol	Change			
2	Aave	2018/1/30	0.15	0.17	0.17	0.14	530470	-7.95			
3	Aave	2018/1/31	0.14	0.15	0.15	0.13	396050	-11.1			
4	Aave	2018/2/1	0.11	0.14	0.14	0.11	987260	-17.46			
5	Aave	2018/2/2	0.1	0.11	0.11	0.08	1810000	-8.32			
6	Aave	2018/2/3	0.11	0.1	0.12	0.09	1200000	6.85			
7	Aave	2018/2/4	0.09	0.11	0.12	0.09	1040000	-18.16			
8	Aave	2018/2/5	0.07	0.09	0.09	0.06	756000	-24.39			
9	Aave	2018/2/6	0.09	0.07	0.09	0.05	819460	26.28			
10	Aave	2018/2/7	0.08	0.09	0.09	0.07	890850	-10.06			
11	Aave	2018/2/8	0.09	0.08	0.09	0.08	211470	15.81			
12	Aave	2018/2/9	0.1	0.09	0.1	0.09	746800	5.99			
13	Aave	2018/2/10	0.08	0.1	0.1	0.08	684670	-13.93			
14	Aave	2018/2/11	0.08	0.08	0.08	0.08	292770	0			
15	Aave	2018/2/12	0.09	0.08	0.09	0.08	225650	0			
16	Aave	2018/2/13	0.08	0.09	0.09	0.08	589310	-7.4			
17	Aave	2018/2/14	0.1	0.08	0.11	0.08	1370000	26.47			

The functionality of the right view is shown below.



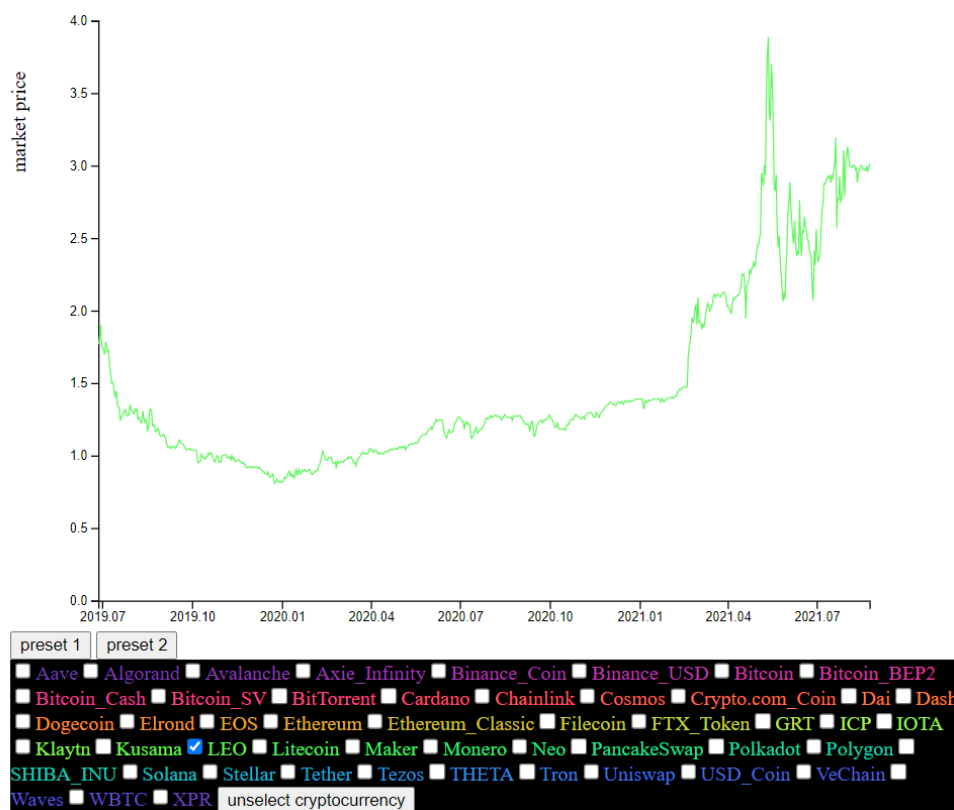
☐ Aave ☐ Algorand ☐ Avalanche ☐ Axie\_Infinity ☐ Binance\_Coin ☐ Binance\_USD ☐ Bitcoin ☐ Bitcoin\_BEP2  
☐ Bitcoin\_Cash ☐ Bitcoin\_SV ☐ BitTorrent ☐ Cardano ☐ Chainlink ☐ Cosmos ☐ Crypto.com\_Coin ☐ Dai ☐ Dash  
☐ Dogecoin ☐ Elrond ☐ EOS ☐ Ethereum ☐ Ethereum\_Classic ☐ Filecoin ☐ FTX\_Token ☐ GRT ☐ ICP ☐ IOTA  
☐ Klaytn ☐ Kusama ☐ LEO ☐ Litecoin ☐ Maker ☐ Monero ☐ Neo ☐ PancakeSwap ☐ Polkadot ☐ Polygon ☐  
☐ SHIBA\_INU ☐ Solana ☐ Stellar ☐ Tether ☐ Tezos ☐ THETA ☐ Tron ☐ Uniswap ☐ USD\_Coin ☐ VeChain ☐  
☐ Waves ☐ WBTC ☐ XPR

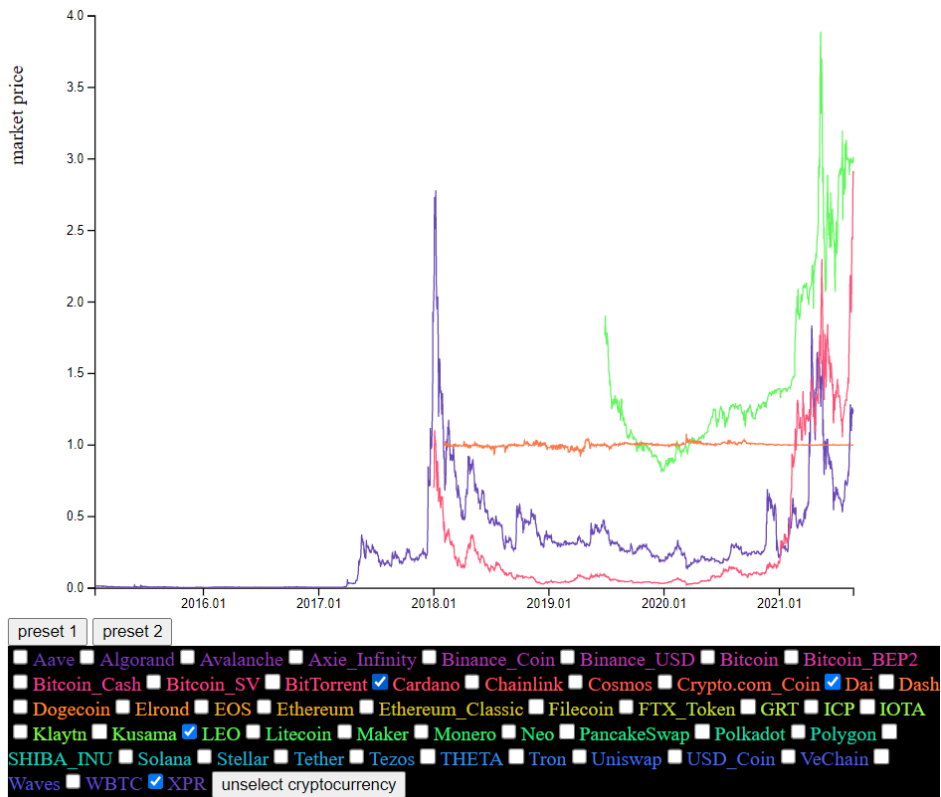
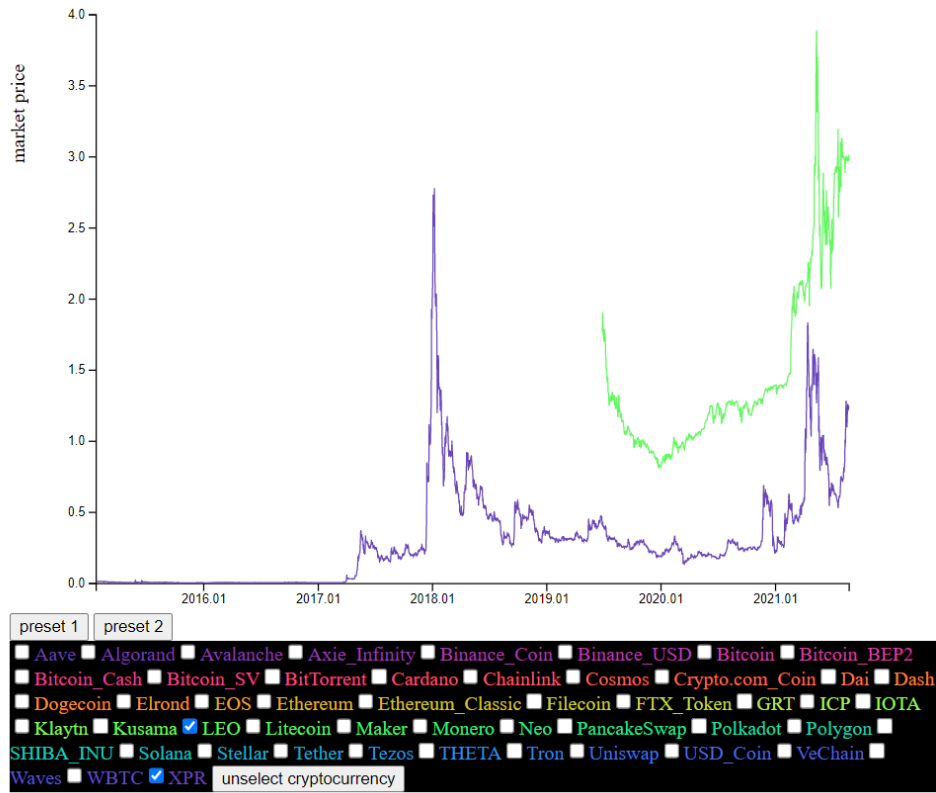


We have tried different color schemes for different cryptocurrencies (applied on lines in the line chart and texts in the checkbox group), such as `d3.schemeCategory10` and `d3.schemeAccent`. We noticed that these ordinal color schemes have a limited number of categories (most of them have only 10 colors each). However, there are many cryptocurrencies (the data set we use here includes the top 50 cryptocurrencies) we need to display. For now, we use a continuous color scheme `d3.interpolateRainbow` to sample 50 colors out of it as our ordinal color scheme. The shortcoming of this method is that adjacent currencies have similar and even indistinguishable colors, such as between 'Aave' and 'Algorand', or 'Bitcoin\_Cash' and 'Bitcoin\_SV'. We may be able to deal with this problem with a better color scheme, adding an interactive highlighting, or adding tooltips to each path.

The preset buttons are designed to show a few preset data sets with the time ranges and currency selections, which we will prepare in advance for the user.

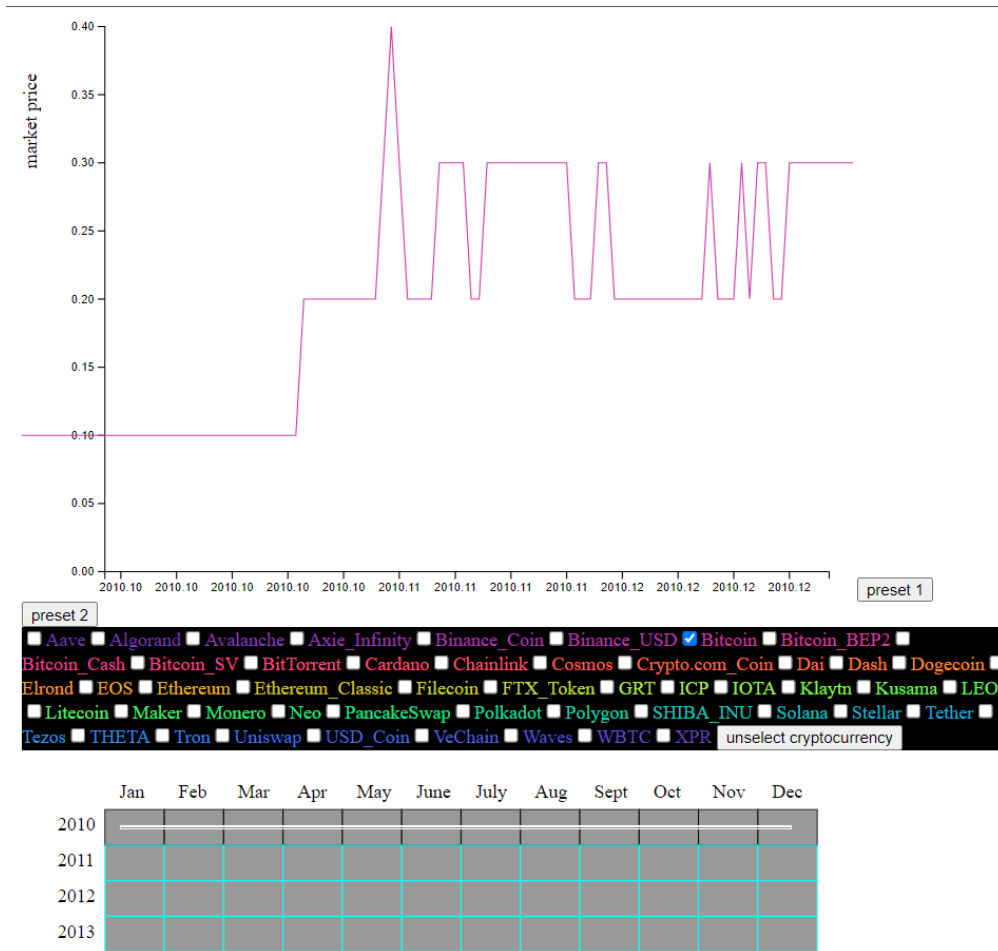
With checkboxes checked, currencies can be selected, allowing users to focus on some currencies they are interested in. With the 'unselect cryptocurrency' button clicked, we remove all currency selections. A few examples are shown below.





By brushing on the grid, a time range can be selected, allowing users to focus on a specific time range they want to explore. With the 'unselect time' button clicked, we remove time range selection. A few examples are shown below.

In year 2010, there was only one cryptocurrency, BitCoin, as below.



Bitcoin changes its name to WBTC in 2020, as below.



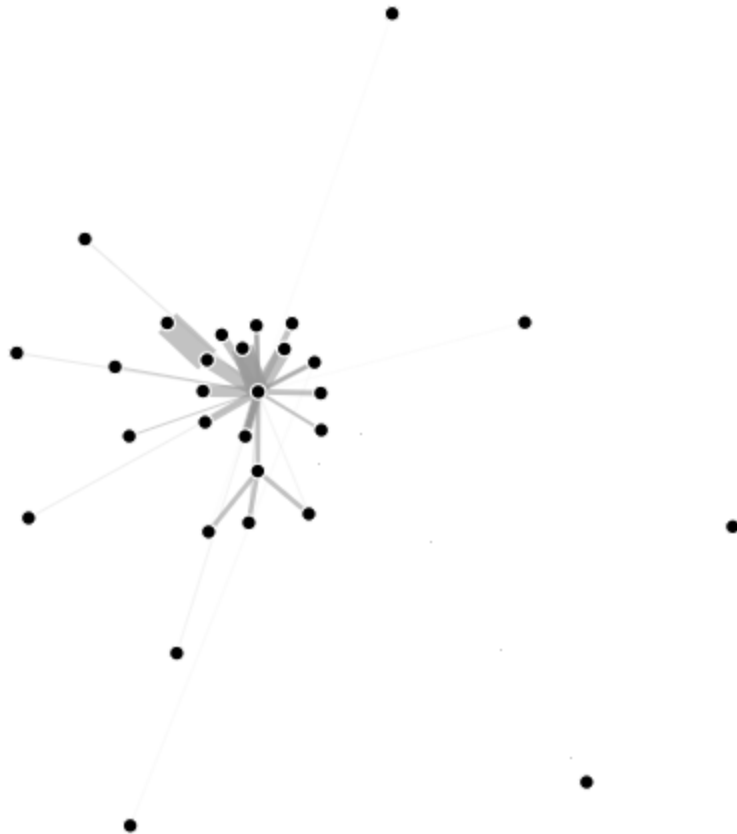


The force-directed tree is plotted from the data of bitcoin transactions in the early days (allegedly somebody purchased a pizza with 10,000 bitcoins). Currently we just use a small sample of the data as the whole dataset is 94MB and we are still figuring out a way to responsively read and visualize it. The data is in the following csv format:

```
timestamp,input_key,output_key,satoshis
1280859762000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,12hEZssdwGLS8UsavJzYAUgxiXLA4x4bu7,36000000
1511333577000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,13ejC8wyUmYG7wzznASEgEEPurWqpe1fcE,10000000000
1289859983000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,13gWLBmdqyftxUw7mvtn4FQdGaAAWkErgF,500000000
1281069800000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,13pVVjBntbheW93GNCWKA1ZMdbQrRopNsp,3212000000
1279119052000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,156mDTc51R4k9yL3J5GGapkr2o83KQShWz,100000000000
1271619803000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,157fRrqAKrDyGHR1Bx3yDxeMv8Rh45aUet,100000000
1272225849000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,157fRrqAKrDyGHR1Bx3yDxeMv8Rh45aUet,1500000000
1271646596000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,157fRrqAKrDyGHR1Bx3yDxeMv8Rh45aUet,1000000
1273901908000,1XPTgDRhN8RFnzniWCddobD9iKZatrvH4,157fRrqAKrDyGHR1Bx3yDxeMv8Rh45aUet,1200000000
```

We sample a small amount of the data and implement the force-directed tree with dragging interaction. The strength and width of the link are determined by the amount of transactions so that the nodes will create a cluster. Most of its functions are inherited from the d3 example, but we tend to add zoom and filter functionality in the future. Potentially, coloring can be decided by the clustering of the nodes.



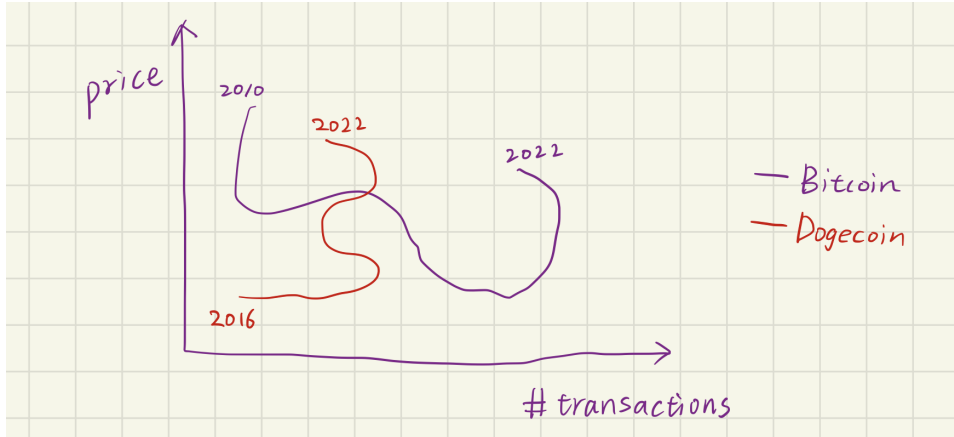


## Next Step

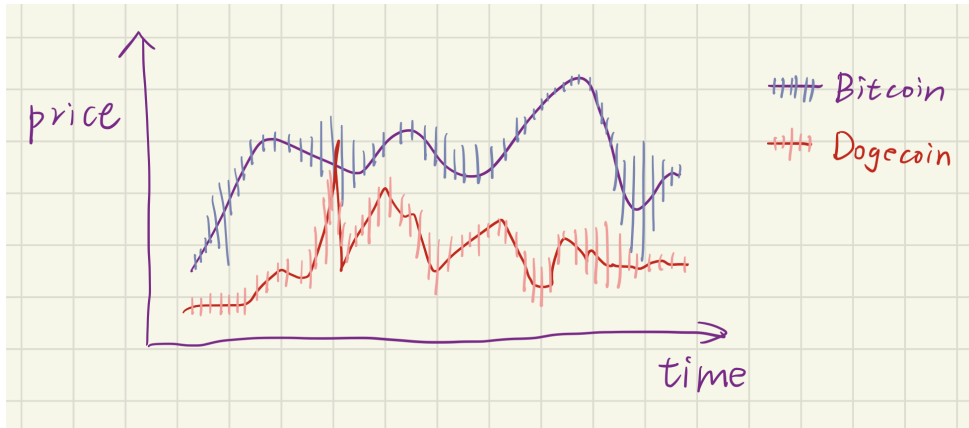
We are adding richer interactions between views. By design, we plan to use the brush as an overall control panel to filter the time range and each view will update in response to the brush selection. We will also get rid of the dense checkboxes and use the on-click method of the treemap to select the bitcoins plotted in the line chart.

## Ideas on Optional Further Improvement

For the line chart, we come up with an idea to improve it by changing the x-axis from time to the number of transactions ("price vs. transactions" line chart), plotting to scatter points on the chart and connecting them together to encode changes of both prices and transactions over time, as sketched below.



Another idea is that we may encode the change in price into the line chart by adding a vertical line to each data point on the current “price vs. time” line chart, encoding the number of transactions, as sketched below.



These two ideas pose a bigger challenge to the color scheme and the interactive highlighting effect mentioned earlier.

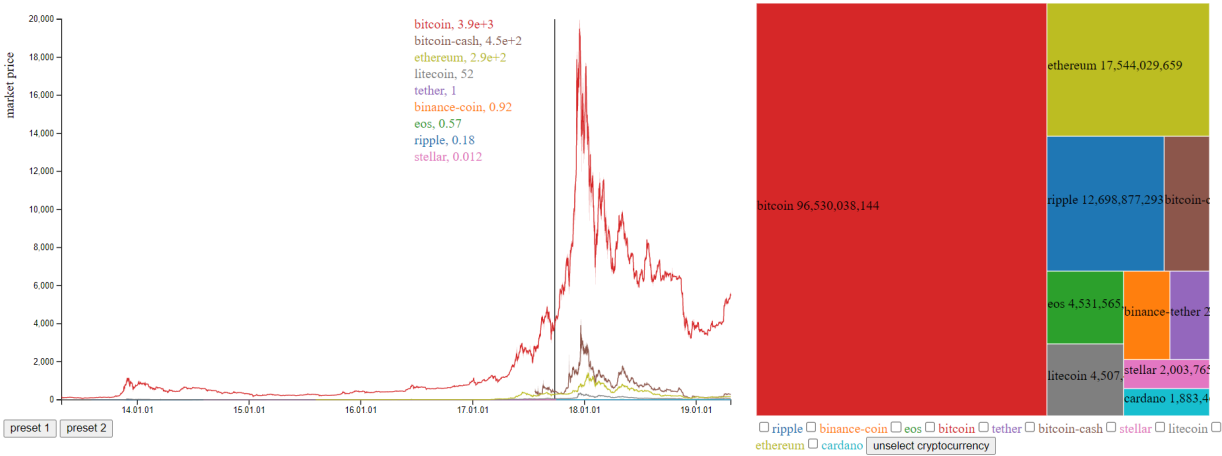
2022/11/22

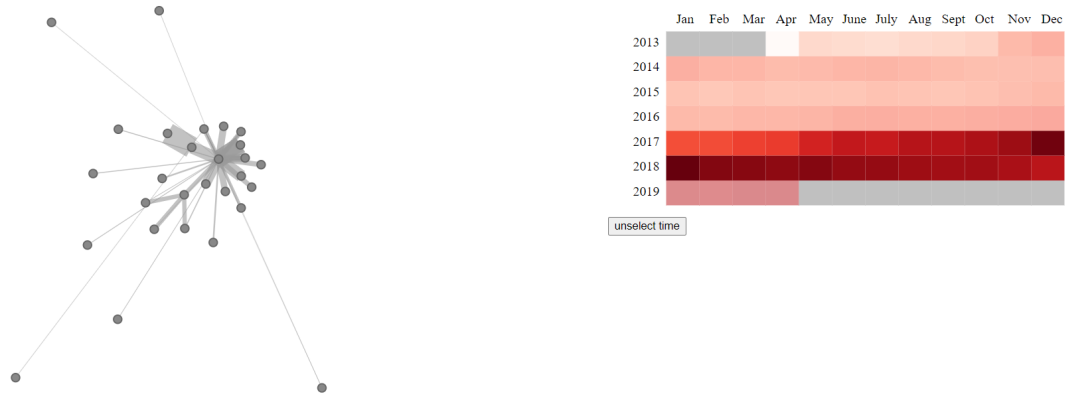
Improved network layout: upon hovering on the nodes, it will highlight its connected ingoing links in green and outgoing links in red. We also exploited the possibility of encoding the transaction amount and time via channels of width and opacity.



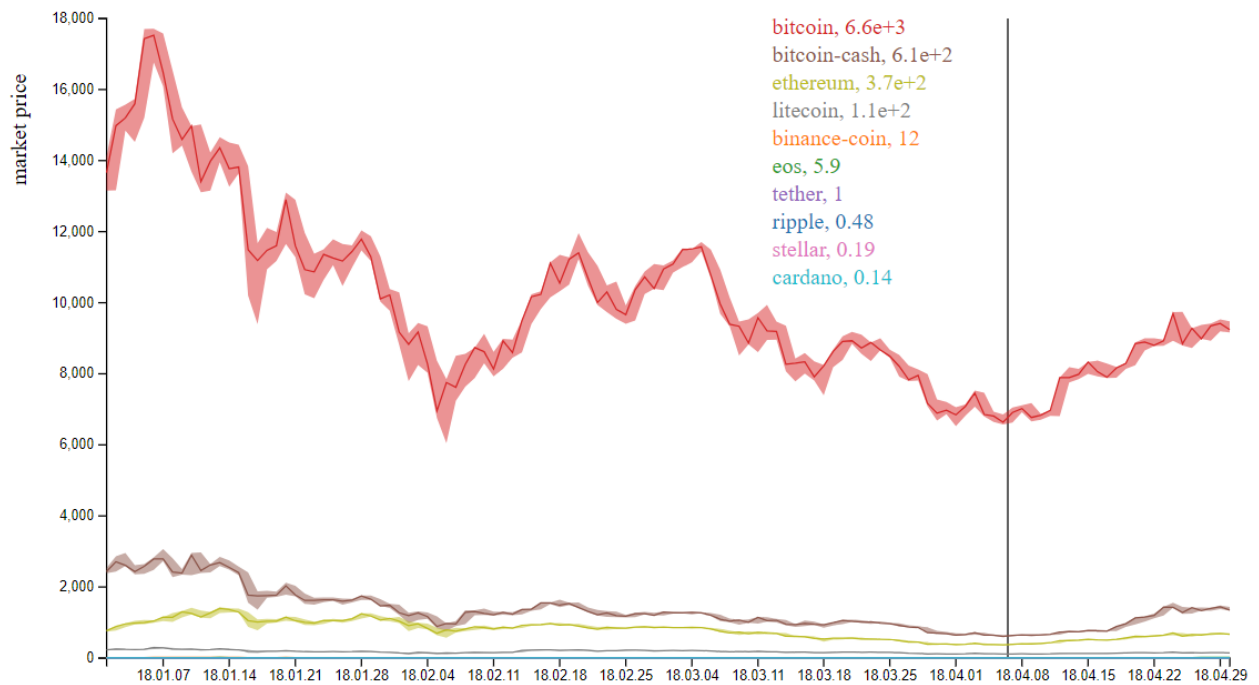
Modify the whole layout

Cryptocurrency Spy Name: Yintong Shang; E-Mail: u1431623@utah.edu; UID: u1431623; Name: Haoyang Shi; E-Mail: u1431587@umail.utah.edu; UID: u1431587

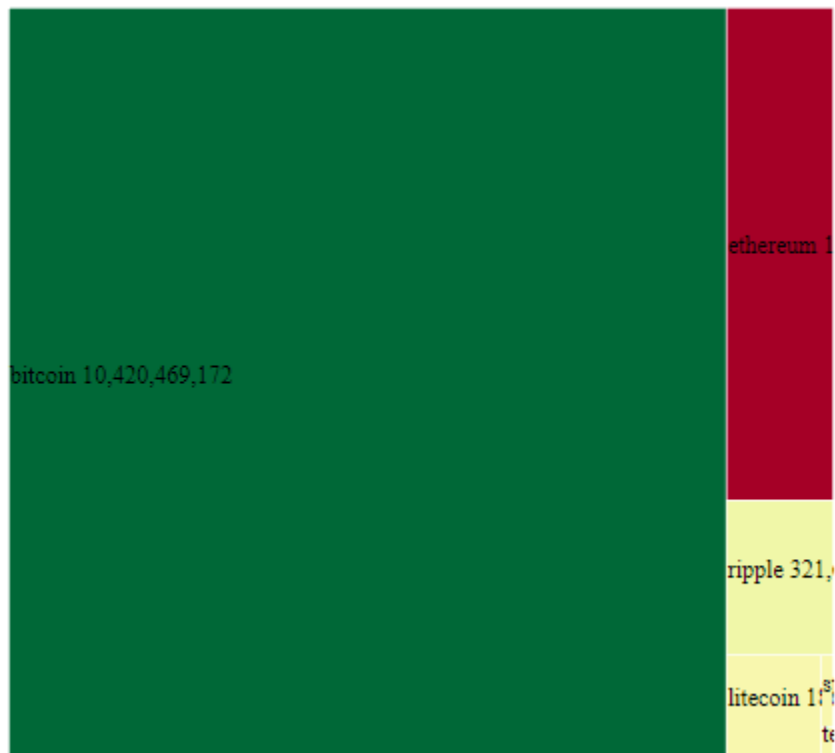
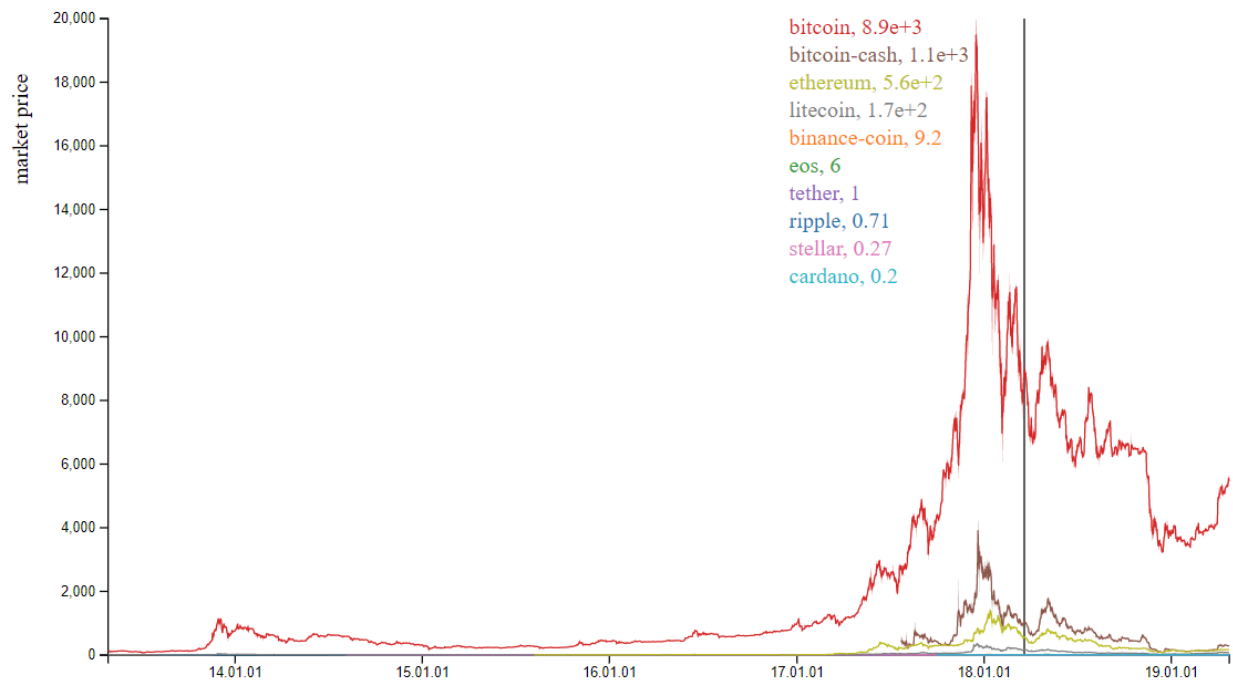




Add area encoding price 'high' and price 'low' to line chart

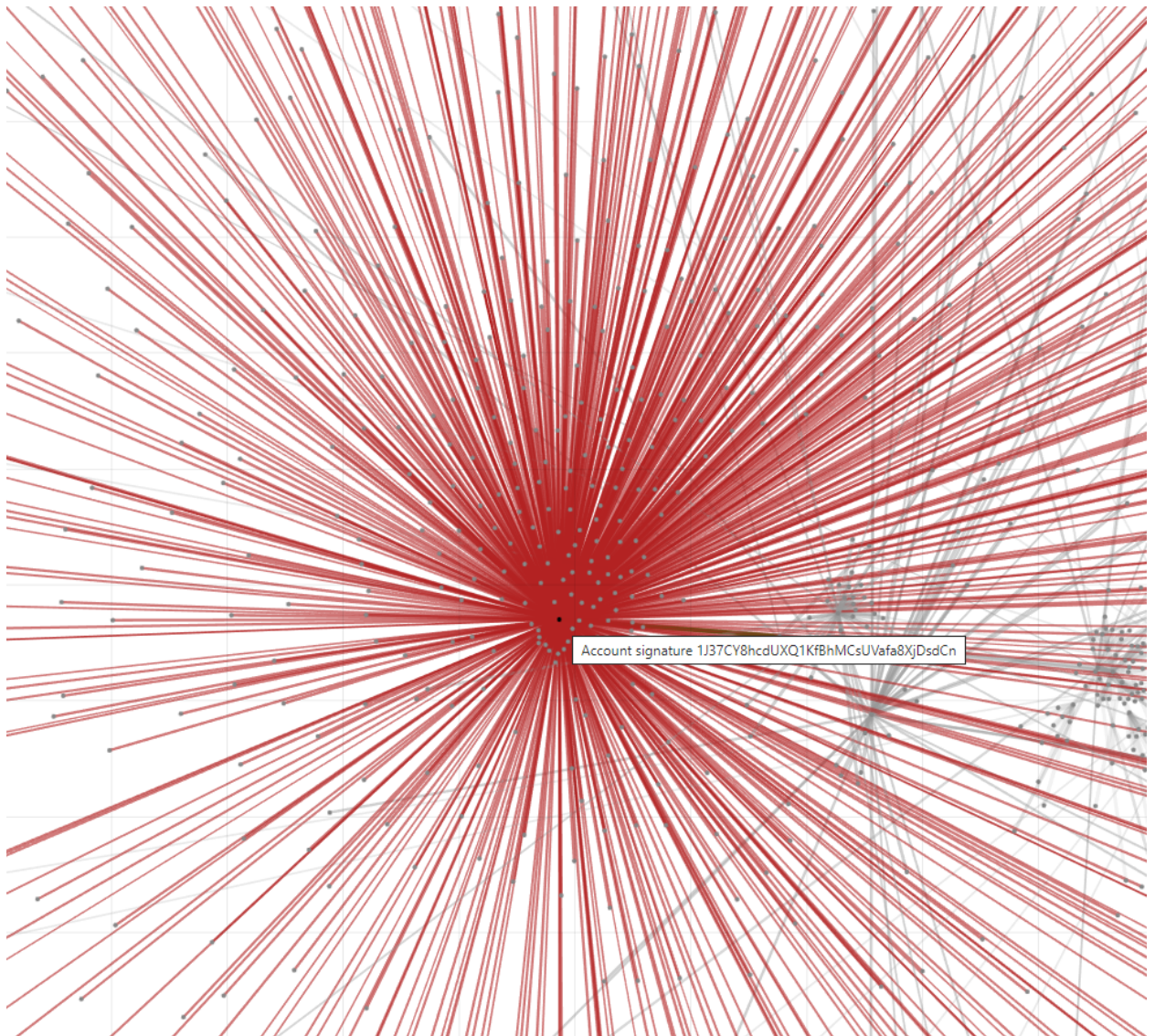


Add overlay to the line chart

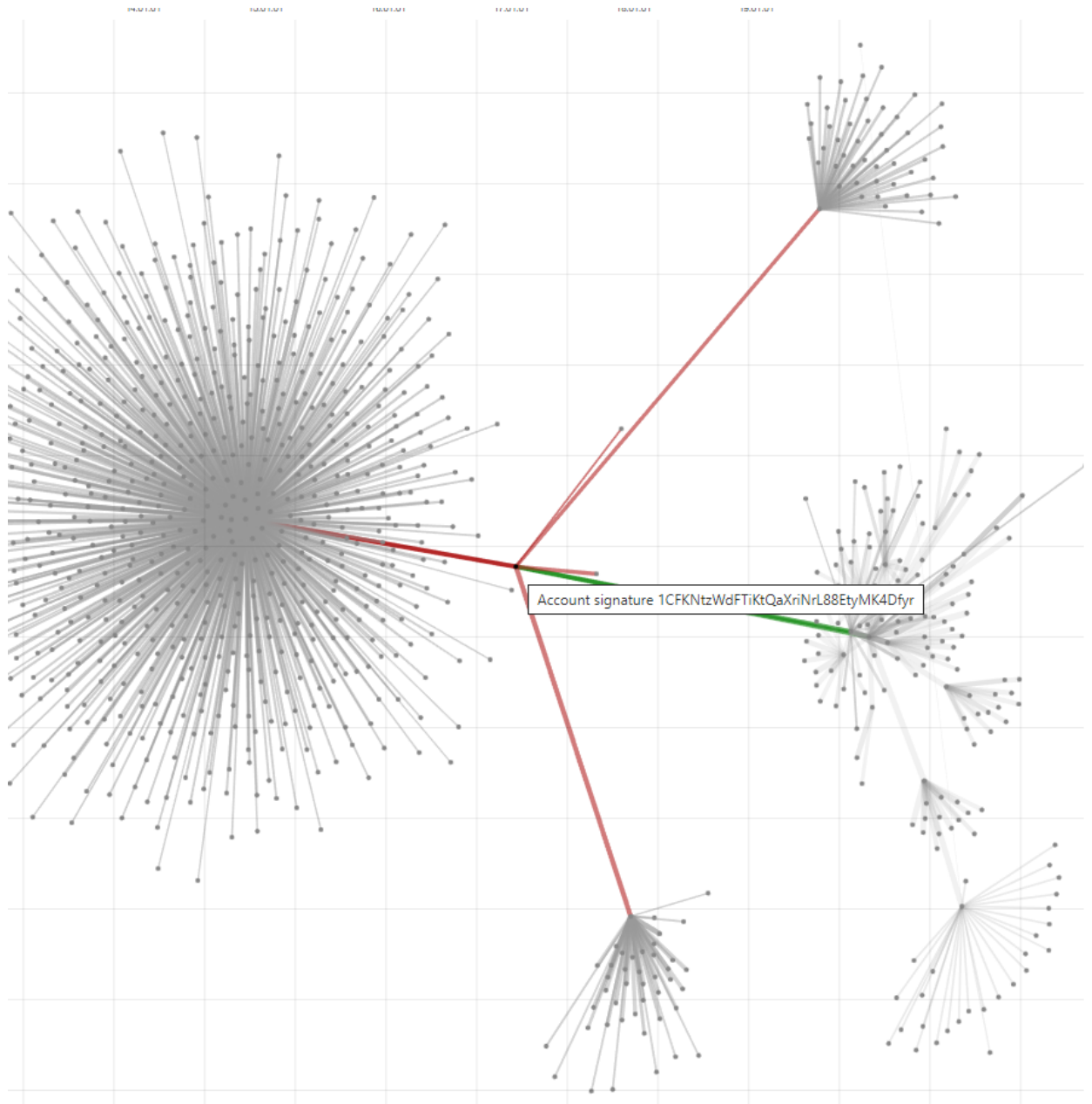


We applied the diverging red-green color map. There are very small tiles that don't have enough room to accommodate the caption. We plan on using the zoomable treemap ([Zoomable Treemap / D3 | Observable \(observablehq.com\)](#)) to resolve this issue, putting all the small tiles into a zoomable "others" tile.

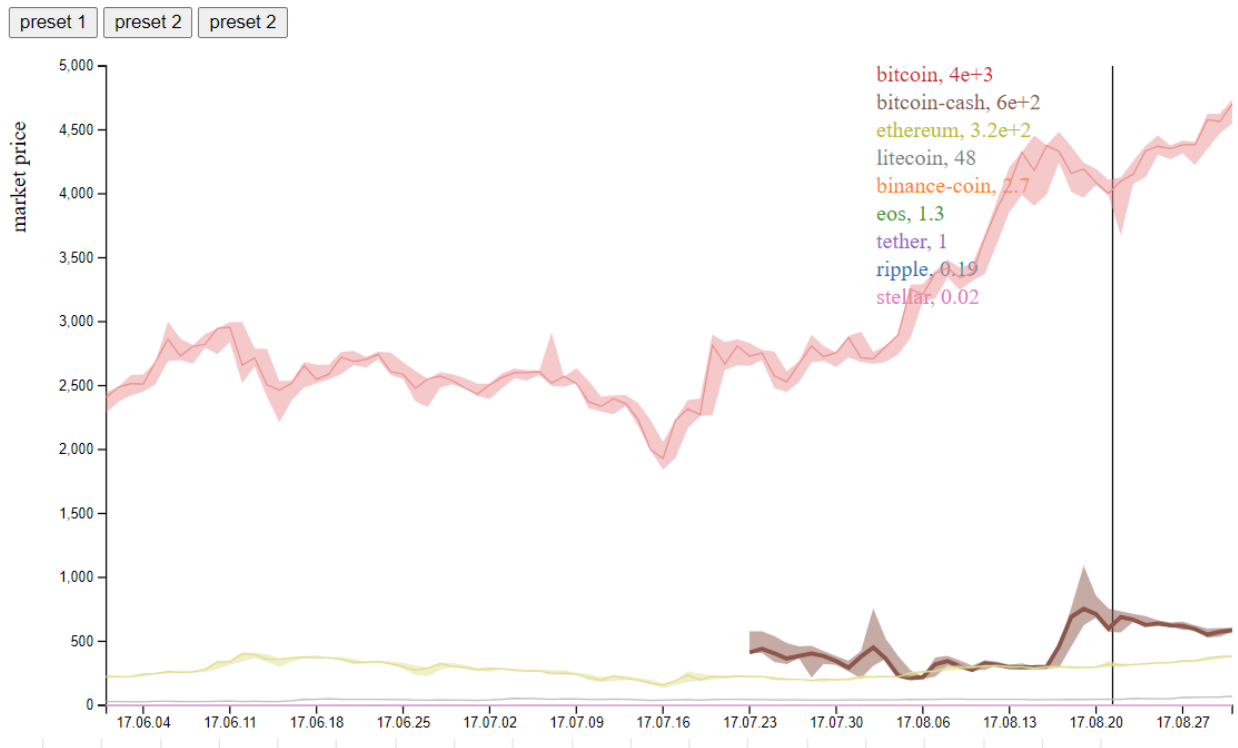
Applied zoom and pan for network view. It eases users so much when the graph is huge.



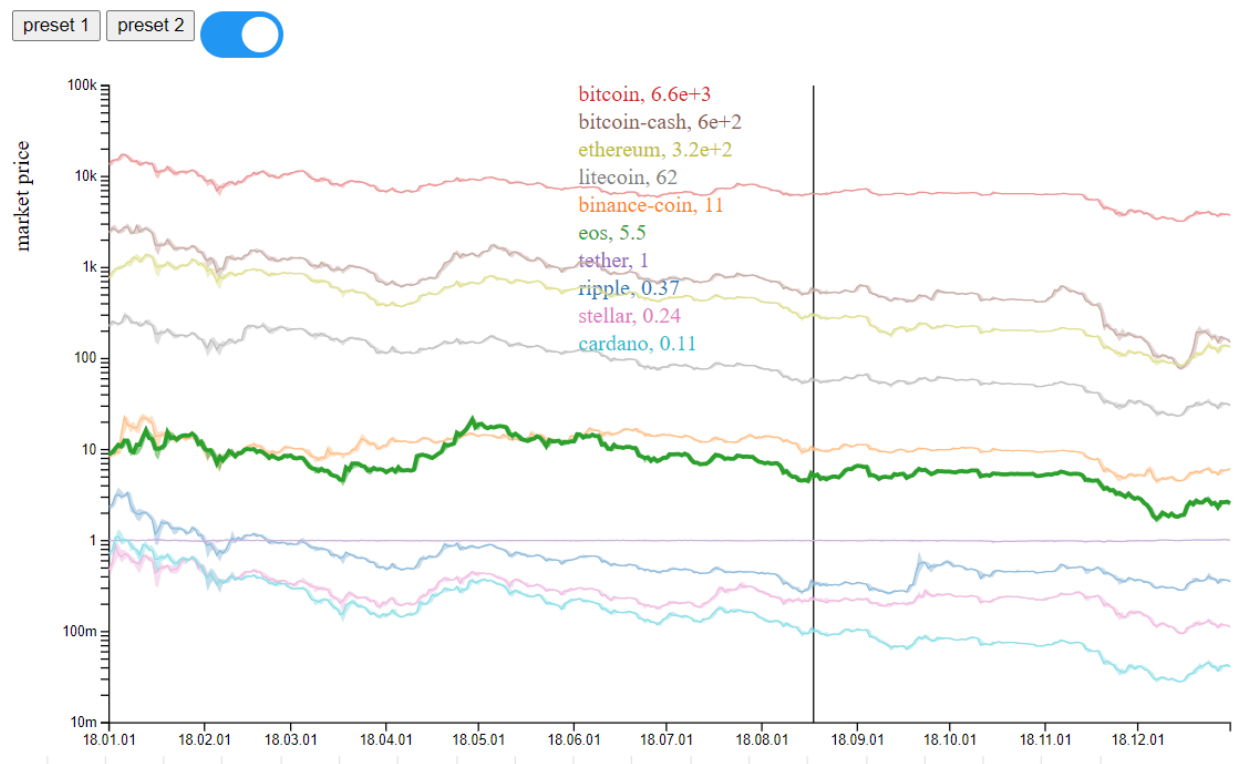
After loading 1000 edges the network is already overcrowded. We did the above experiments under the settings that link force strength is the square root of the transaction value. We tried several scales and noticed that the log scale gives the best clustering



Add hover highlight and fading on line chart

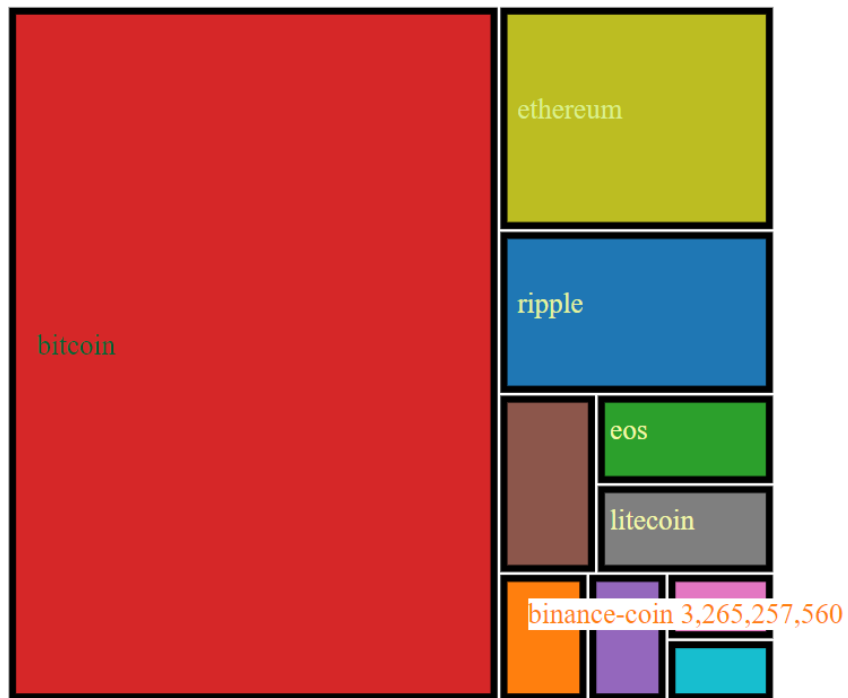


Add log scale toggle switch

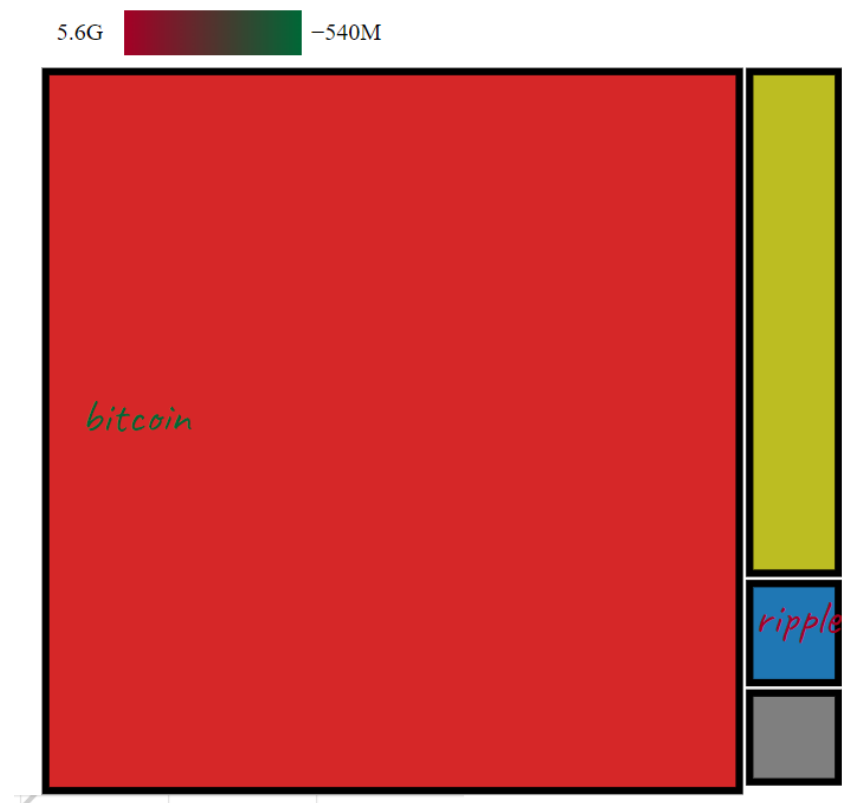




Hide texts for small rectangles in the treemap, and add a tooltip for all rectangles for info display.

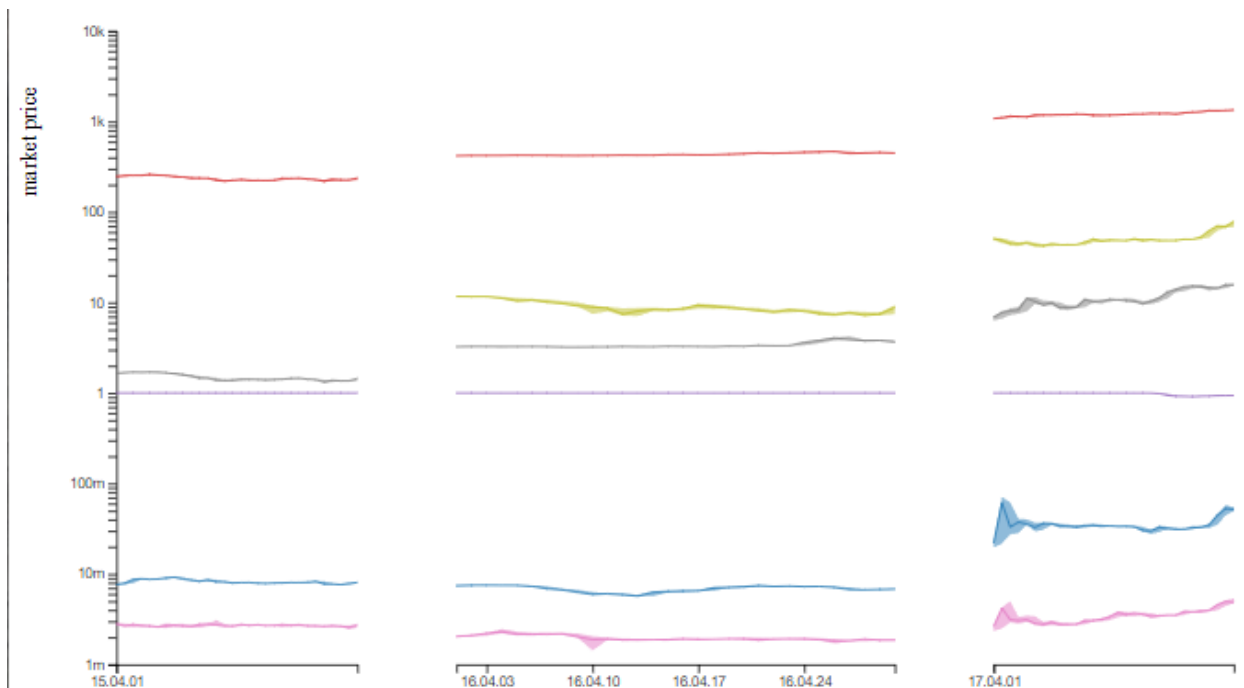
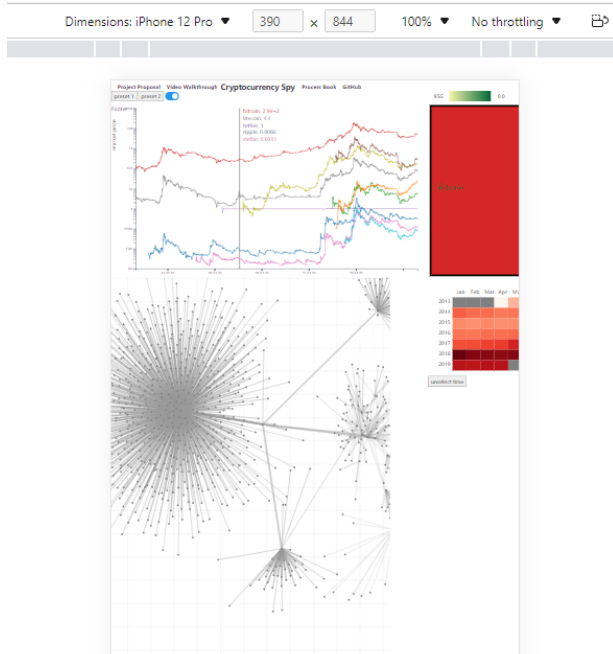


Add a color legend for the treemap



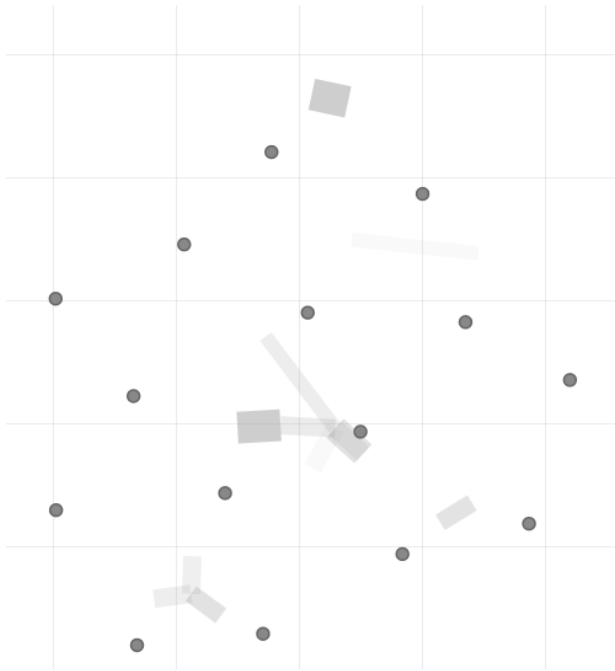
2022/11/26

Deployed a stable branch at [Cryptocurrency Spy \(cryptocurrency-spy.github.io\)](https://cryptocurrency-spy.github.io). The layout needs some refinement under mobile settings.



When using the brush to select interested seasons, we emit the unselected intervals to optimize the information-ink ratio and lead users' attention.

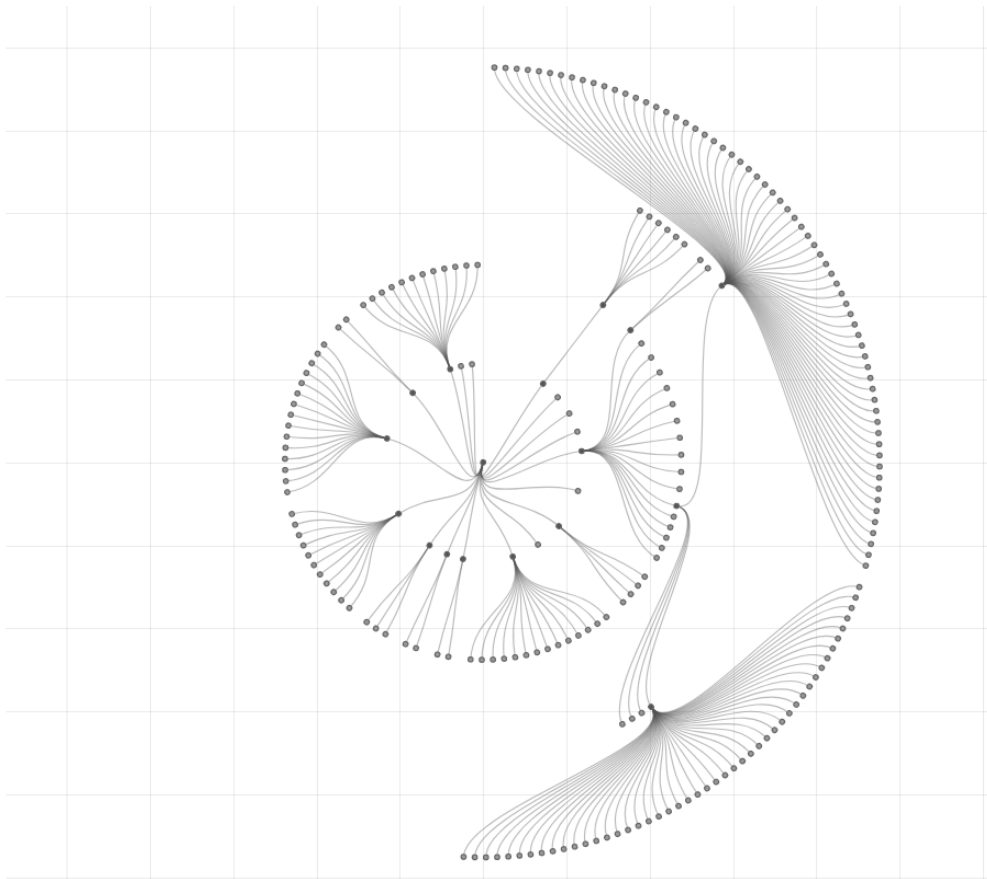
11/28

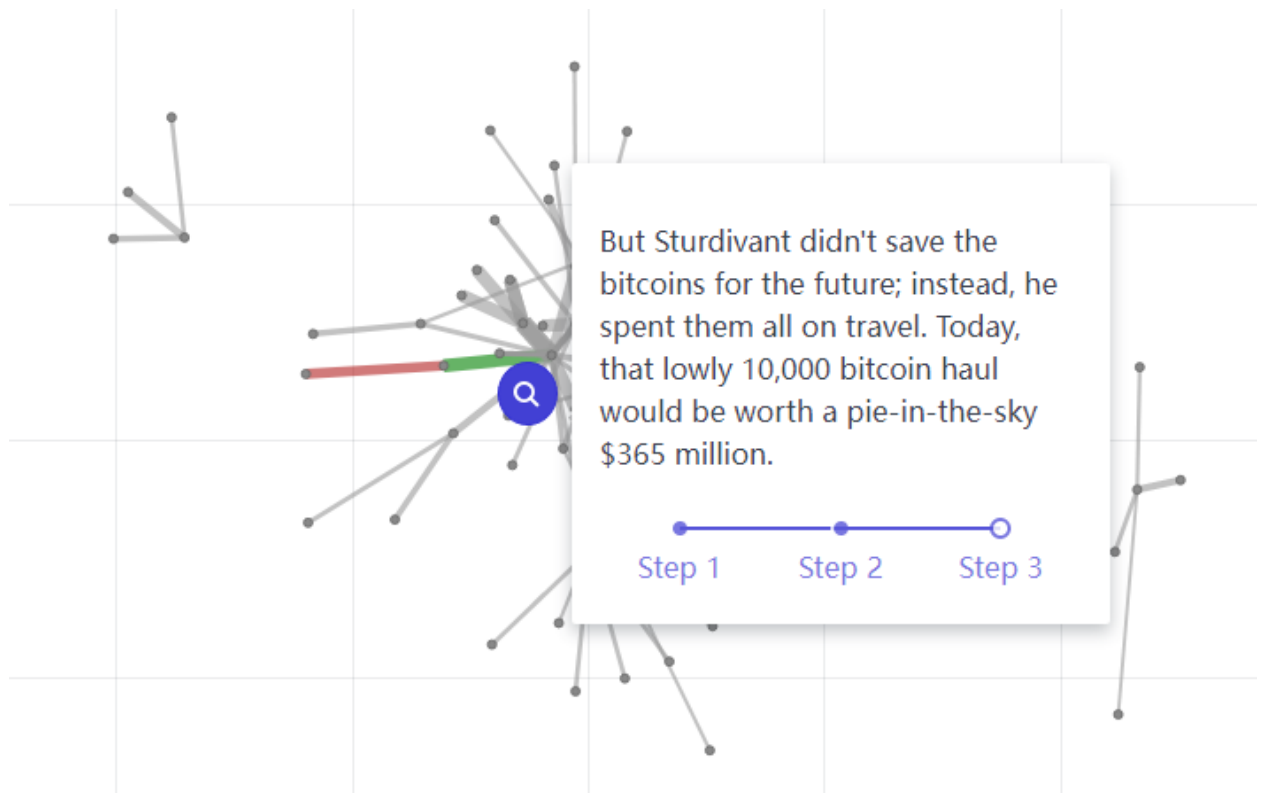


The force layout encounters a major reentrance problem. In particular, even if the whole svg is removed and then executes the same code over, the new result still looks impacted by the leftovers of the last render. Also, its responsiveness isn't great at large scale. We may need to seek other visualizations with better scalability.

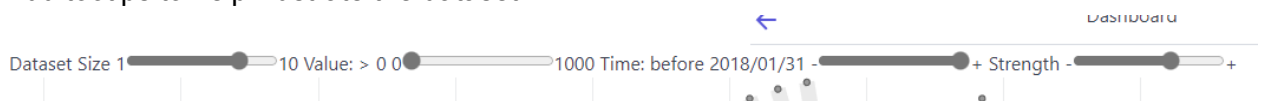
The zoomable radial tidy tree was selected and developed as our backup solution. We lose some fidelity in simplifying the transaction graph to a tree, but because we use a dataset tracking the bitcoin flow of someone buying a pizza with 10,000 bitcoins, this fidelity loss is not horrible.

We choose not to display the node names because that's really the hash of someone's signature and looks really meaningless. We hid the information on the tooltips and show them on hover.





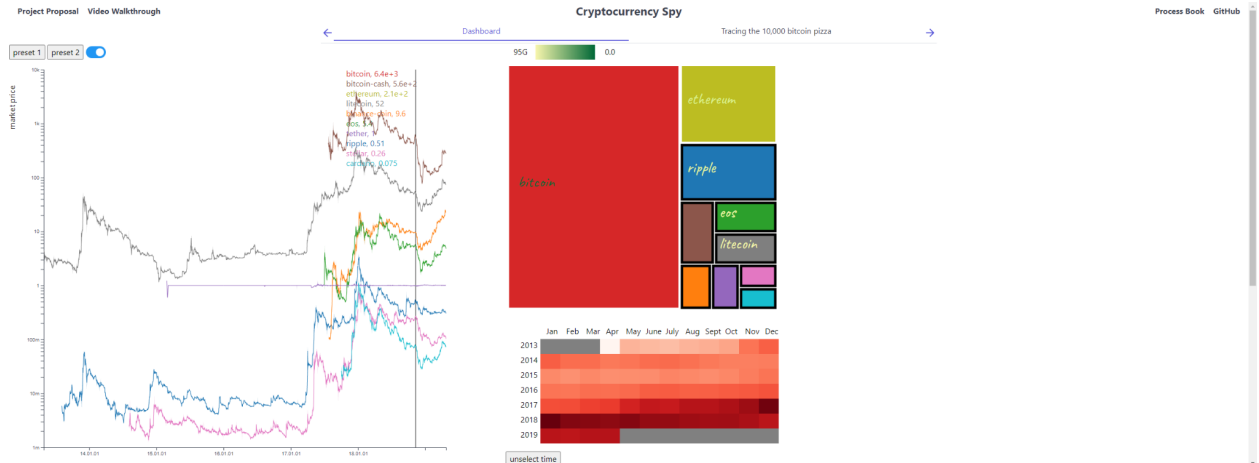
Add tooltips to help illustrate the dataset.



Add toolbars to allow filtering by date and value.

Because the transaction dataset is huge, we load a minimum amount of data at startup. Users can use the slider to adjust the dataset size and load more data on the fly so that the delay of loading a large dataset can be hidden.

Another benefit is that when the dataset size is big enough and force-directed tree becomes less effective, we can automatically switch to the radial tree layout.



Reorganized the views so that each story can have a full page of its own.

This leads to one redundancy that the brush and the time slider both filters time. But splitting them is sensible as the time scopes of the two stories vary to much.

**We decided that the first story is about the cryptocurrency bubble and dramatic up-and-downs in 2017 and 2018, and the second one about someone squandering his bitcoins for pizza in the dawn of cryptocurrency.**