PyCoin: Process Book

[Website](#) | [Github Repo](#)

By Raul Jordan and Jacques van Rhyn

Our Project

We live in a period of unprecedented change, where even our most fundamental beliefs about the world and society are being challenged. We’re particularly interested in cryptocurrencies, a phenomenon that has the potential to change the way we do transactions between each other forever.

As of now, there is no single cryptocurrency to invest in, no gold standard in the world of online currencies: cryptocurrencies like Dogecoin, Bitcoin and Litecoin are currencies that are prominent but they are currencies that constantly fluctuate. But besides the big three, there are many more lesser-known cryptocurrencies, from (365) to Zetacoin. The big issue about all of these currencies is that no-one really knows which cryptocurrency is going to be the next big cryptocurrency, the next viable alternative to the US Dollar.

We want to create a visualization that compares all of the current publicly traded cryptocurrencies. Sure, this already exists, but we believe that we can do it better than anything available online.

The following are the top three biggest websites that enable a user to look up the exchange rates between cryptocurrencies:
http://coinmarketcap.com/
http://www.coinwarz.com/cryptocurrency/coins
https://www.cryptonator.com/

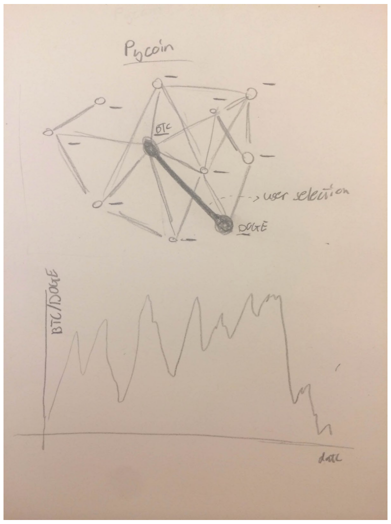
None of these enable a user to lookup the historical exchange rate between two markets. This is important and useful data. From deciding which currency to invest in to figuring out arbitrage opportunities, having a way to display the historical exchange rate data between any two cryptocurrencies would help greatly to further the field of alternative currencies.

Log Book

Setting up the repository
[entry by jacques 3.23.2015]

Today we set up the repo for our final project from Raul Jordan's github account. Even though we decided that we would do a lot of the coding together, we also decided that we would use separate branches in both Jacques' and Rauls' computers to avoid the same kind of issues that we had experienced in CS51 with commit issues. It's also just a lot neater

Initial design idea
[entry by jacques 4.3.2015]



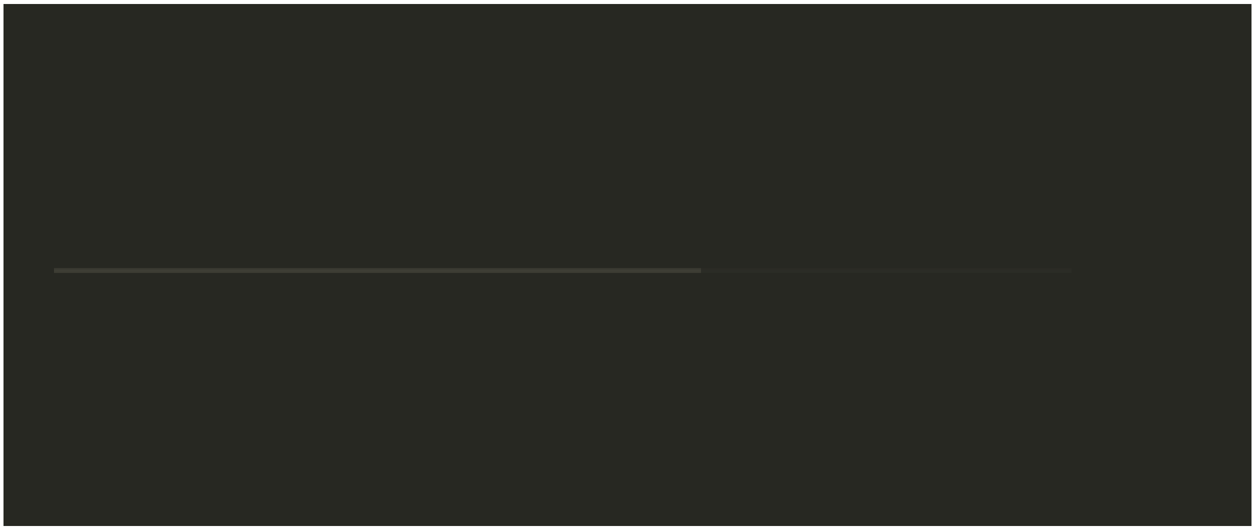
For our homework 4 suggestion, our idea was to have a network of all the cryptocurrencies display at the top of the visualization. At the bottom, we want to have the exchange rate history between the two cryptocurrencies selected by the user. The user will select the cryptocurrencies to display by hovering over the edge that connects them

Day of research
[entry by jacques 4.5.2015]

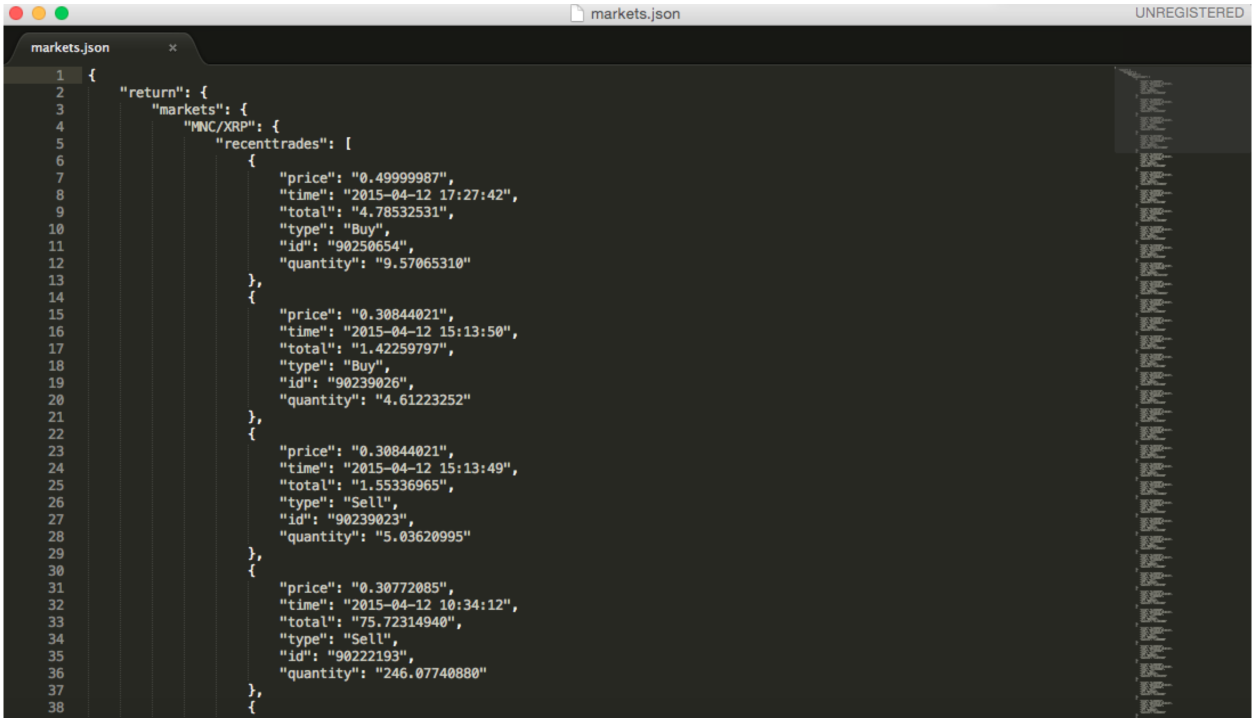
Where on earth would we find trading data between different currencies? After realizing that there's no big convenient file of trading data on the internet, we realized that scraping was the best way to obtain the data we wanted for the project. We spent the whole of today looking at and figuring out the different API’s of cryptocurrency websites. Finally, we decided that we wanted to use the API on cryptsy, a HUGE database of trading data between different cryptocurrencies.

First day of coding
[entry by jacques 4.12.2015]

Today, we sat down and figured out how to scrape data from cryptsy using Python with the requests package. After a lot of trial and error we were able to tease out vast amounts of data from the website. There’s nothing more satisfying than having a data file so big that it takes a while for sublime to open it.



loading…



```
{
    "return": {
        "markets": {
            "MNC/XRP": {
                "recenttrades": [
                    {
                        "price": "0.49999987",
                        "time": "2015-04-12 17:27:42",
                        "total": "4.78532531",
                        "type": "Buy",
                        "id": "90250654",
                        "quantity": "9.57065310"
                    },
                    {
                        "price": "0.30844021",
                        "time": "2015-04-12 15:13:50",
                        "total": "1.42259797",
                        "type": "Buy",
                        "id": "90239026",
                        "quantity": "4.61223252"
                    },
                    {
                        "price": "0.30844021",
                        "time": "2015-04-12 15:13:49",
                        "total": "1.55336965",
                        "type": "Sell",
                        "id": "90239023",
                        "quantity": "5.03620995"
                    },
                    {
                        "price": "0.30772085",
                        "time": "2015-04-12 10:34:12",
                        "total": "75.72314940",
                        "type": "Sell",
                        "id": "90222193",
                        "quantity": "246.07740880"
                    },
                    {
```

It lives!

This file contains exactly what we need it: it sorts every trade between two different currencies together. As you can see in the example above, MNC/XRP contains all the trades between MNC (MinCoin) and XRP (Ripple). The data in our file is between 1 March and 13 April, 2015

We also decided to have a look at the different ways with which we could implement the website: we decided that we wanted to apply design principles not just to our visualization but to our graph itself. This is a bit tangential but after looking at Google's material design principles, we also found a really cool framework called materialize.

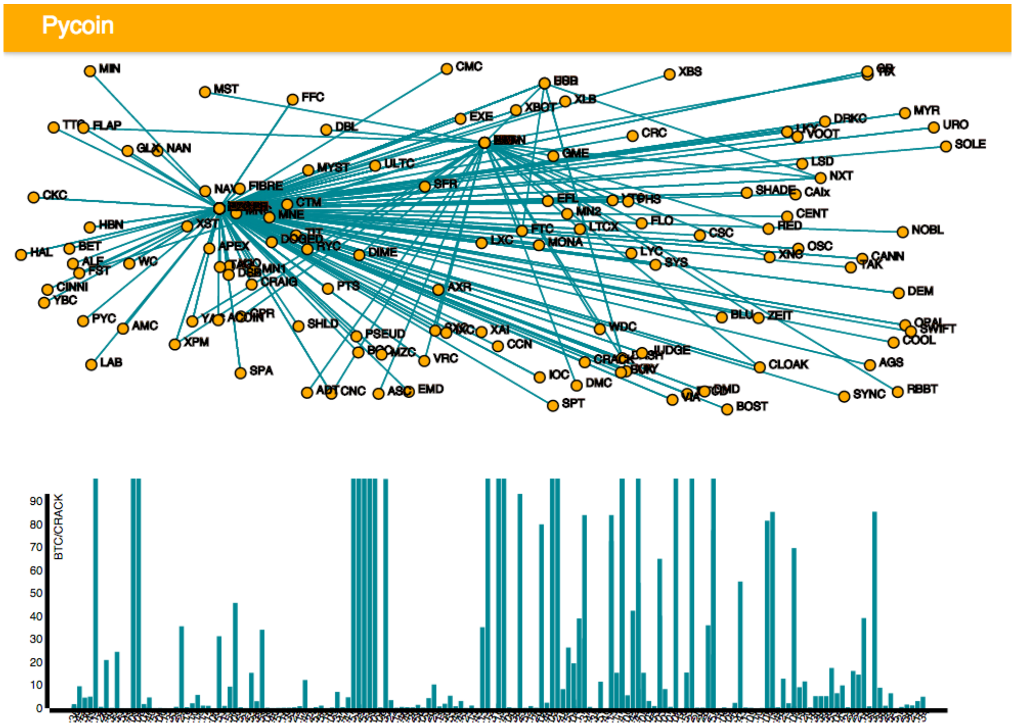
We decided on a color scheme (#ffab00 for orange and) that we wanted to follow for the rest of the project and after tinkering with materialize and heroku for a while, we had a beautiful heroku website, http://pycoin.herokuapp.com/

--- image of the home page ---

Second big day of coding
[entry by Jacques 4.16.2015]

We built our prototype today - it’s not that pretty and not fully functional but it’s a step into the right direction.

Here’s a screenshot of the first time we got the nodes working with a graph underneath them.

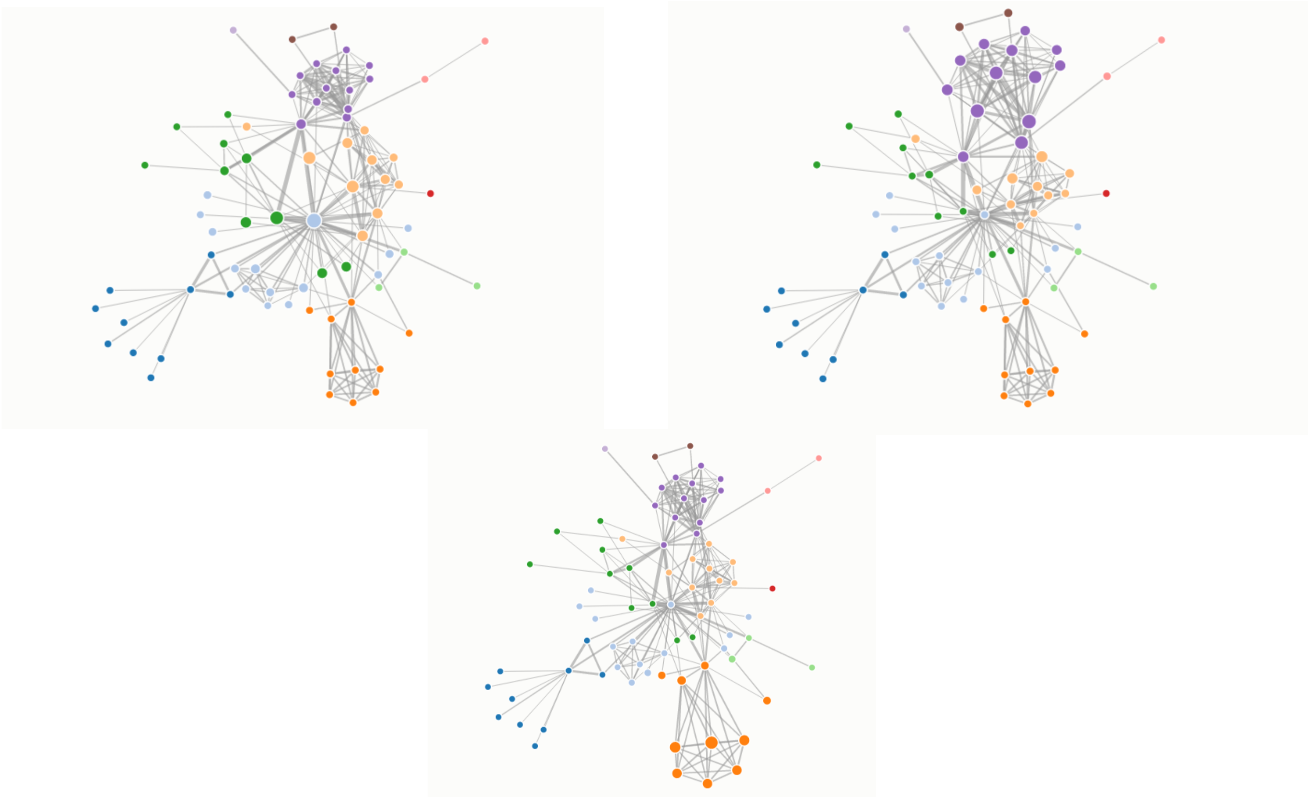


of course, we don’t want it to be a barchart underneath the nodes - what this picture does illustrate though is what we had in mind for the display: on selecting something, a graph gets displayed underneath it. We plan on having a graph that shows change over time in a much better fashion. When a user hovers over a link, we see that the barchart at the bottom changes to encode the recent trade data between those two linked cryptocurrencies.

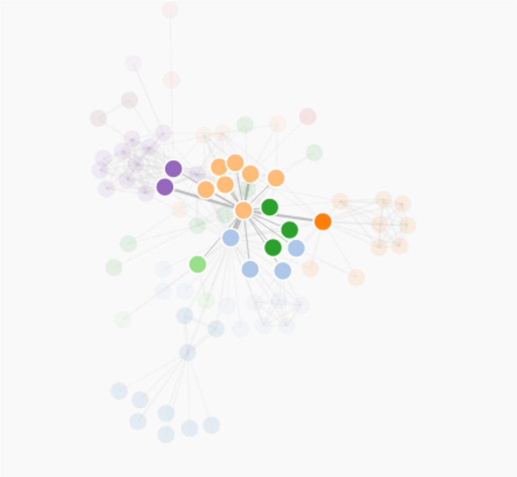
There is definitely still work that we want to do:

Firstly, the currencies are very cluttered in parts of the visualization: it’s sometimes hard to read the labels of the individual nodes.

To solve this problem we want to use the d3 fisheye plugging, allowing the user to see more when the mouse hovers over the nodes.

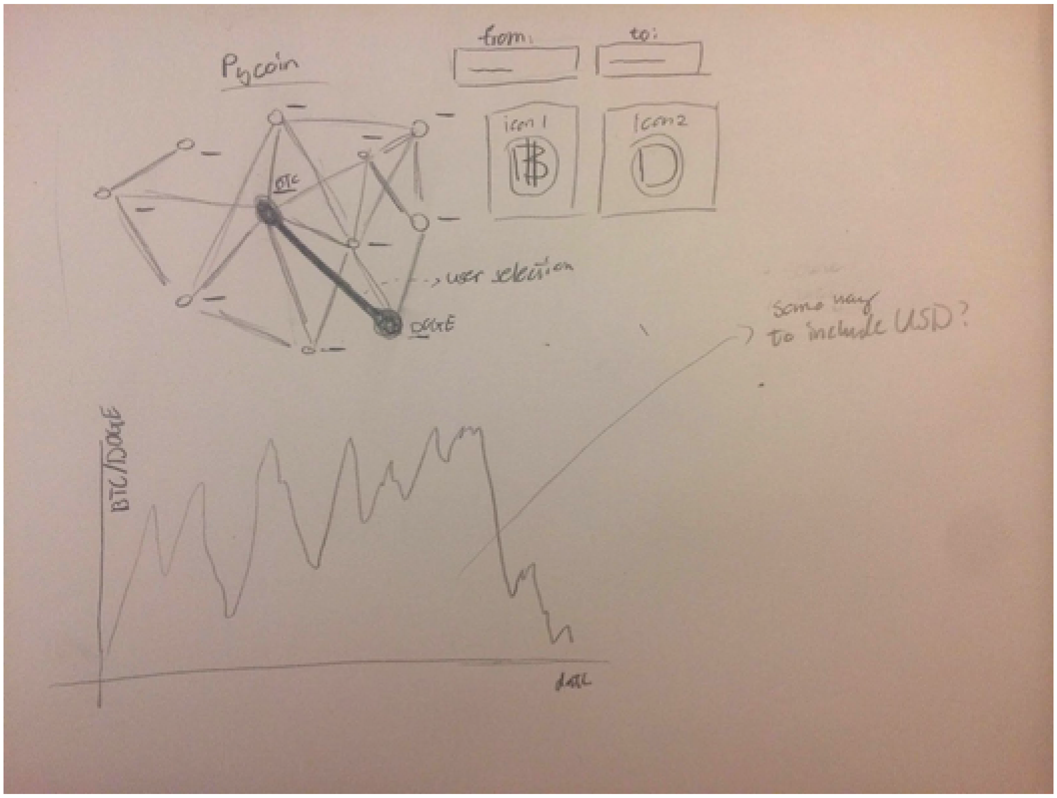


We decided that some kind of highlighting, as in making all non-adjacent nodes to a clicked node, fade away, will also useable.



We also want to make the visualization more useful - we decided that we need a search function that will do highlighting on the found result. We also realized that users might not be familiar with the codes of cryptocurrencies, we like the idea of displaying the full names and icons of icons when the user hovers over the edge connecting them.

Here is a rough idea of some of the additional features we want to implement:



Finally, we decided that we wanted to make the force layout more clever and encode more information. How we're doing this is something we'll decide after having a chat with our TF.