

DA5020 Term Project Report

Kyle Weise

12/10/2017

Introduction

Cryptocurrencies have received an increasing amount of attention from the media over the past year, none more than Bitcoin. At its core, Bitcoin can be thought of as a peer-to-peer trust network. Although the nuts and bolts of it are a complex mix of cryptography, software security, and network science, the general idea is relatively simple. It is a decentralized, distributed ledger system. When Bob wants to send money to Alice, he broadcasts to the network that he'd like to do. Based on the properties and mathematics behind different cryptographic hashing algorithms, it can be proven that Bob owns as many Bitcoins as he says he does, and that Alice and not some other person will receive them. Although the underlying technology (blockchain) can be applied to practically any industry, Bitcoin has made headlines because it takes on one of, if not **the** biggest industry in the modern world today, finance. Basically, any computer running the Bitcoin software has a copy of every transaction every made, and is constantly being updated with new transactions. The mechanisms by which this work are incredibly interesting but beyond the scope of the report. It's inception a couple of years after the housing market crash of 2008 was no coincidence; Bitcoin was developed in stark contrast to the large, centralized banks that lost Americans thousands and millions of dollars. At the core of many cryptocurrencies (this term has unfortunately expanded to encompass digit assets that do not intend to work as an actual currency) is the idea of decentralization. The idea that we shouldn't have to rely on one giant server owned and operated by one giant corporation or entity to process our information. We should be able to process it ourselves.

Through this project, I wanted to create an application that took a basic look at the prices of Bitcoin over the years since it's inception.

API Creation

Bitcoin price data is available from dozens if not hundreds of different services. I chose CoinDesk because it has a simple API and is a big name in the crypto-space. At the time of my starting this project, there existed no package in R that wrapped up the CoinDesk API endpoints into functions, so I had to make one myself. The source code for this can be found in the `coindeskAPI.R` file, and I think is pretty straightforward to follow. It heavily leverages the `curl`, `httr`, and `jsonlite` packages, some of which we've seen in class before.

Storage

When it comes to storage, I had the advantage of being able to choose between a traditional SQL database (MySQL, Oracle, etc) and a newer document based NoSQL database (MongoDB, Couchbase, etc). I am really only storing the price (an integer) and the Date (a string or datetime), so I decided to go with the more traditional SQL database, although I could have just as easily stored the JSON messages the API returns in a NoSQL database. When I began the project, I wanted initially to implement some kind of machine learning algorithm that took the price/date data as input, and would be able to learn and possibly predict the price levels in the future. Unfortunately that did not come to fruition (discussed later), but it did lead me to the package `sparklyr`, which I used as a storage mechanism. `Sparklyr` is a wrapper API for Apache Spark, which is a distributed cluster computing framework that makes available dozens of different machine learning algorithms. The cool thing about `sparklyr` is that it seamlessly integrates with `dplyr`, so everything we've learned from the tidyverse is still applicable. It also lets the user provide SQL statements if that's more familiar.

Visualization

Plot

The plot visualization I chose to do is a pretty basic one; the x-axis is the date, and the y-axis is the price. Instead just of a static plot, I implemented some interactivity on the side panel that allows the user to specify the date range they wish to look at, as well as the options to display the y-axis as log values and to display a smooth line (using `geom_smooth`) over the data points. The smooth line checkbox opens a slider input that allows the user to specify the degree of smoothness, as well as an animation button that animates the line 0.01 to 1.

Table

The tab over from the plot is the table tab, which shows the data being used to generate the plot, ordered by descending price.

Real-time

Finally, the last tab shows the most recent price of Bitcoin as retrieved from the CoinDesk API. It is invalidated (refreshed) every 5 seconds.

Hosting

One of the major reasons I wanted to create a Shiny app in the first place was so that I would be able to share it with others. The entire Shiny community is kind of rooted in that, and I wanted to contribute. The RStudio team makes it ridiculously easy to host your own Shiny app on the web via their PaaS (Platform as a Service) product shinyapps.io. It's as simply as downloading a package, creating an account on shinyapps, make sure your app behaves as you want it to locally, then press a button and voila, your app is now being hosted on the web. They offer more robust administration and hosting capabilities for a subscription fee, but the free level is sufficient enough for my one app.

Challenges / Improvements / Experience

One of my main disappointments was that I was not able to get a machine-learning algorithm implemented in any fashion. I realize it would have been above and beyond the scope of this project, but I wish I could've got it working. After finishing, I found that the area that caused me the most trouble was working with dates. The `lubridate` package handles them very well, which is all fine for the tidyverse, but for some of the base R functionality it caused some weird errors. Regardless, here are some improvements that could be made and/or learning experiences I've had:

- Implement applicable machine learning algorithm(s)
- Implement real-time data to generate the plot in real time
- General app appearance and layout
- Implement more plot interactivity (i.e. zoom, hover, brush, etc.)
- Learned about creating a custom wrapper for a previously non-existent API endpoint
- Learned how to publish a Shiny app to the web via shinyapps.io

Conclusion

In conclusion, I think Bitcoin as a technology is incredible. As popular news outlets seem to talk about it more and more, I don't see it leaving the public eye anytime soon. The blockchain technology that Bitcoin is based off of is revolutionary, and I think as we push into the future we are going to see much more concern for information privacy and decentralization. My web app can be visited [here](#).

References

CoinDesk API [shinyapps.io](#) Various StackOverflow questions and answers