CRYPTO MARKET ANALYSIS

Geordi Reiner, David Xie, Tien Tran
SJSU
CS161 - Ghofraniha
Spring 18

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

Special thanks to our CS161 instructor Jahan Ghofraniha for the in-class assistance and supplementary materials used for our value at risk model.
Special thanks to this Keggle article for our csv files:
https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory
Lastly, special thanks to anyone in our class that gave us development feedback.

EXECUTIVE SUMMARY

Cryptocurrency trading has become an increasingly lucrative market over the last year. The current market evaluation is listed at around 450 billion dollars, a sharp rise from its 20 billion dollar evaluation in 2015. Being a relatively unregulated, frontier method of trade, there are numerous reports of scams, ponzi schemes, and faked market data looking to deceive potential investors. Cryptocurrency data analysis is similar to the stock market's in that most consumer-side results are either inconclusive, far-reaching, or unconsolidated. Furthermore, there is a scarcity of risk assessment modelling in a market rife with unpredictability. Recent scandals such as the Bitconnect ponzi scheme shows that there is a clear customer need for proper data analysis, most specifically value at risk modelling. Our team has developed a minimal cryptocurrency data analysis web application that allows a user to upload an appropriate .csv file and receives output of a graphical linear regression and a breakdown of a Bitcoin-limited value at risk model. We hope that projects like ours are only the start of proper data science ventures in the cryptomarket.

BACKGROUND

The theme for our particular section of our software project class was data science and machine learning. We were encouraged to include at least some aspect of these fields in our projects, whether with heavy or minimal impact. Our project leans on the heavier side of this spectrum with a specific emphasis on data science techniques. For the entire team, this project was our first foray in data science and all its subfields (data pre-processing, modelling, etc.). As a result, we made some sub-optimal technology choices and experienced numerous developmental and conceptual hiccups along the way, with value at risk modelling being the most notable offender. Even so, we learned much and completed what we initially set out to do.

PROBLEM STATEMENT

Besides the blockchain, the cryptomarket's legal framework is relatively unregulated. Along with cryptocurrency's reputation for clandestine transactions, numerous scams and ponzi schemes have developed in recent years. Some of these issues may be mitigated by using appropriate data science techniques, such as value at risk. Value at risk is a known method for determining potential capital loss in stock market investment and can be applied in the cryptomarket environment appropriately. Furthermore, the ongoing rise of cryptocurrency's value supplements a clear market need for more investor data science, of which there is a significant dearth in value at risk modelling.

MOTIVATIONS

Cryptocurrency is the world's fastest rising commodity, rising from 270 billion at the start of this project to 450 billion today. However, outside the blockchain, which guarantees the validity of a given transaction, legal frameworks are still relatively foggy. Initial coin offerings that result in ponzi schemes would not exist if the world had a better idea of how to regulate the atmosphere surrounding the transactions themselves. During the Bitconnect ponzi scheme in February of this year, investors could not believe the incredibly high rate of return on Bitconnect coins. The likelihood of investment return eclipsed that of most other cryptocurrencies on the market, and begs the question: Could an anomalous pattern like Bitconnect's been predicted through value at risk modelling? Value at risk goes beyond simply observing if investment is "too good to be true", but confirms it through loss prediction that takes all historical data into account.

Our motivation is driven by an investor need, a lack of value at risk modelling in the cryptomarket, and the sharp uptick of most of the market in recent years.

DIFFERENTIATOR

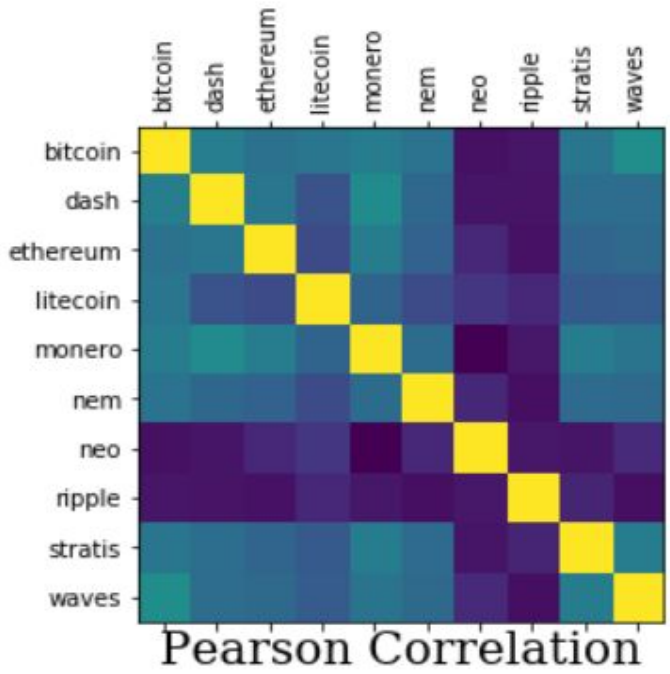
As stated in previous sections, cryptocurrency's rise is a very recent trend. During our preliminary research before proposing the project, we found not just a lack in general data science when compared with the stock market, but also a significantly large lack of value at risk modelling. Furthermore, our team could find no such web application for the presentation of cryptocurrency data analysis. Granted, our scope is

quite limited, but a fully deployed, all cryptocurrencies enabled, live data streaming version of our project would be a great need to potential investors. Hosting a popular web application is also much more lucrative than compiling data science articles for investment columns. This is something that not only differentiates us from the cryptomarket, but physical currency trade markets as a whole.

METHODOLOGY

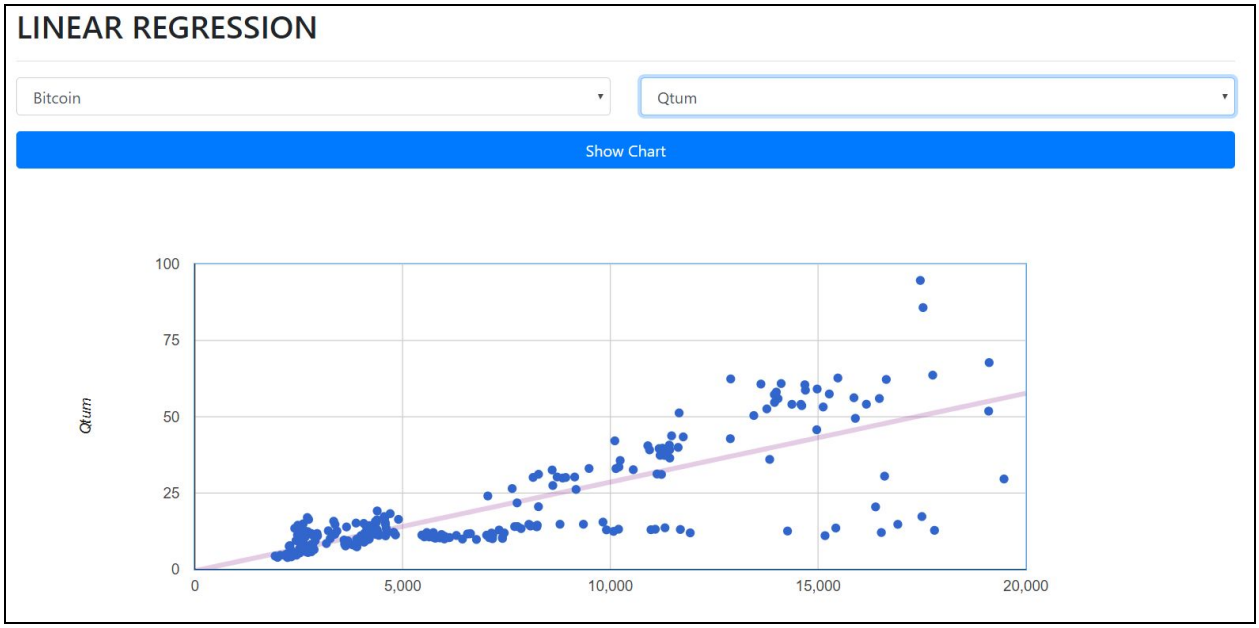
Linear Regression

The Linear Regression model is based on the correlation between the top cryptocurrencies. From our data set, we were able to output the following correlations:



bitcoin 1.000000
bitconnect 0.965897
dash 0.962461
ethereum_classic 0.872567
ethereum 0.933576
litecoin 0.941633
monero 0.919344
nem 0.951953
neo 0.839645
omisego 0.908620
qtum 0.490948
ripple 0.798784
stratis 0.881124
waves 0.924165

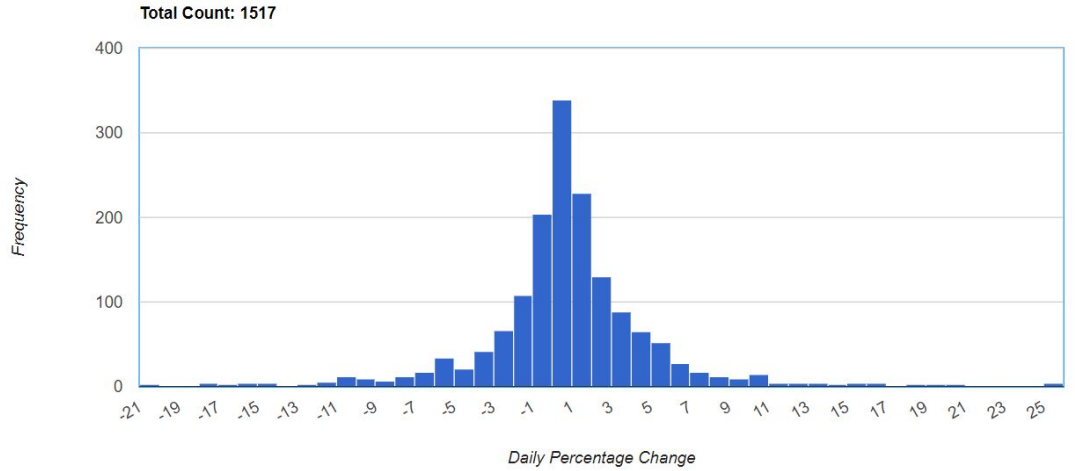
The least correlated cryptocurrency relative to Bitcoin is Qtum with a correlation of 0.49 only. For this project we used a simple Linear Regression model to to graph the price of Bitcoin against the price of other top cryptocurrencies such as Ethereum, Litecoin. Based on the result, we were able to determine that Bitcoin had a relatively high correlation with other top cryptocurrencies. Therefore, we use bitcoin as the primary cryptocurrency for our next model, Value at Risk.



Value at Risk

The Value at risk model implemented was based on the normal distribution of the daily moving average. We convert the USD values for daily change in price to percentages in order to normalize the data. By normalizing the data, we can use the 90%, 95%, 99% confidence interval to approximate the potential loss in investment based on the selected confidence interval. The left edge of the confidence interval is then converted back into USD so the user can have a better understanding of the values at risk. By approximating the left most 10%, 5%, 1% of the confidence interval, we can calculate the potential loss but not limited to that value. Which means even if the left most 5% of a 95% confidence interval is say $2,000, it does not guarantee that the loss will be $2,000 but rather the average of all the losses that leads to $2,000. The leftmost 5% is simply the average of potential losses based on this model.

VALUE AT RISK (Bitcoin)



Bitcoin's daily change in percentage

Mean	Variance	Standard Deviation
0.2566	15.8286	3.9785

Monthly Percentage Change

Start Date	End Date	Total Change	Daily Change
2017-02-01	2017-03-01	25.64%	0.92%
2017-03-01	2017-04-01	-10.54%	-0.34%
2017-04-01	2017-05-01	25.79%	0.86%
2017-05-01	2017-06-01	53.2%	1.72%
2017-06-01	2017-07-01	17.45%	0.58%
2017-07-01	2017-08-01	14.53%	0.47%
2017-08-01	2017-09-01	52.57%	1.70%
2017-09-01	2017-10-01	-2.73%	-0.09%
2017-10-01	2017-11-01	40.67%	1.31%
2017-11-01	2017-12-01	51.35%	1.71%
2017-12-01	2018-01-01	36.97%	1.19%
2018-01-01	2018-02-01	-17.42%	-0.56%

Confidence Interval

Confidence	Total Percentage Change	Change in USD Values (Assuming you bought one bitcoin on Feb-20-2018)
90%	-6.1%	-$685.14
95%	-8.06%	-$905.28
99%	-11.73%	-$1317.49

IMPLEMENTATION & RESULTS

This project was implemented on PHP with ease of access for full stack development. The backend and frontend of the project was coded in PHP which deviated from our original plan. Integration of python scripts within the PHP was a failed experiment by our team. We were only able to call simple python scripts from the command line within a php function. Any package dependencies would not execute properly in the PHP domain. Which meant we were not able to use any of the function calls in python for our project such as numpy, pandas, sklearn etc.

Our project reads from a csv and imports all the data into the database. Our project have several queries that would parse and store the data according to the datatype. Using the data from our database we were able to graph the model for Linear Regression. Using the Linear Regression model, we decided to do value at risk for Bitcoin only because of the strong correlation that it has with other cryptocurrencies.

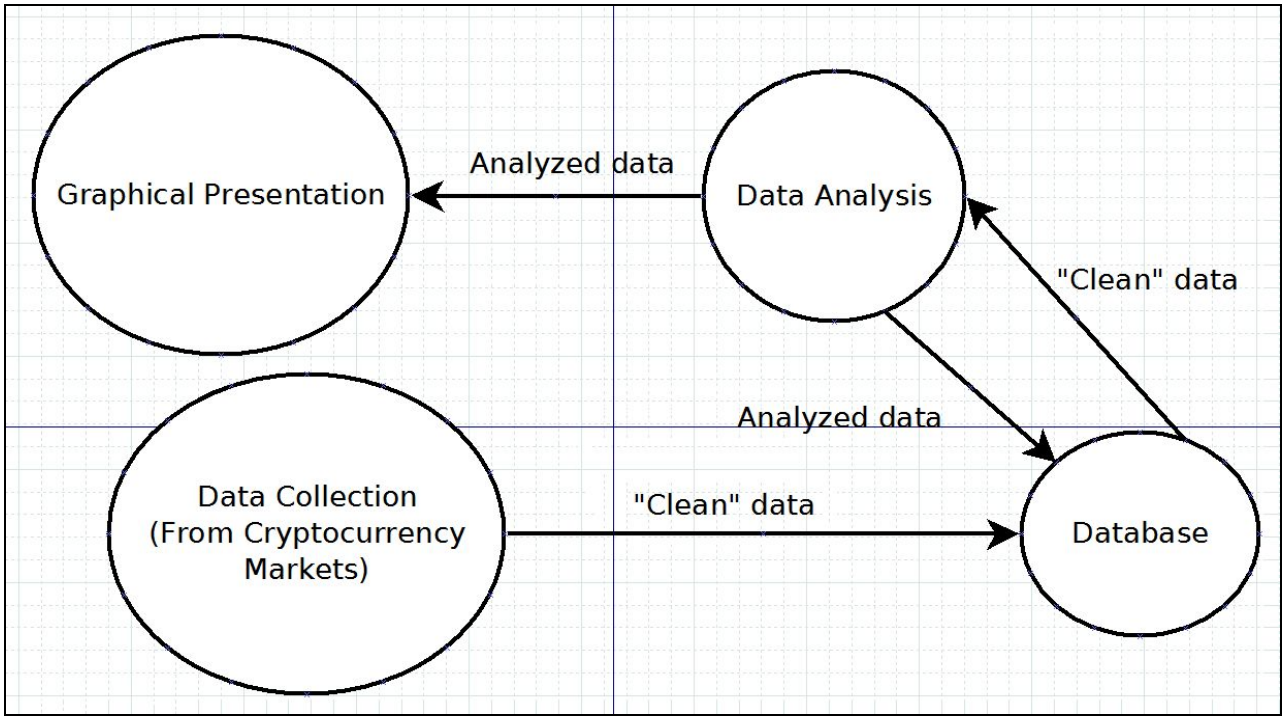
For work distribution, everyone did a little of everything, however these were the points of focus:

- Geordi primarily did management, documentation, and front-end related tasks
- David primarily did research, algorithm construction, and backend related tasks
- Tien primarily did database related tasks.

Technologies Used

PHP calculation engine with SQL queries
Javascript (Google charts)
Apache web server
MySQL database
HTML, CSS, Javascript through Bootstrap framework
XAMPP control panel (for ease of development purposes)

System Analysis



Development tasks were divided roughly as follows:

1. Back-end
    a. Data collection (from file)
    b. Passing “clean” and analyzed data to database
    c. Data analysis on “clean” data
2. Front-end
    a. Receive analyzed data from database
    b. Graphical representation of data
        i. Linear regression
        ii. Risk assessment
3. Database and web server setup
    a. Passes analyzed data to front-end

Functional Requirements

R1: Core Functionality

R 1.1

The application must allow upload of active cryptocurrency trade data.

R 1.2

The application must store “clean” data in a database.

R2: Data Analysis

R 2.1

The application must graphically display a binary-factored linear regression model for cryptocurrencies.

R 2.2

The application should graphically display investment risk assessment for Bitcoin.

R3: Usability

R 3.1

The application must allow users to toggle between linear regression models for different cryptocurrencies.

R 3.2

The application should indicate important results on graphs (ex. Color highlighting on graphs).

Non-Functional Requirements

NR1: Graphical User Interface (GUI)

NR 1.1

The application must have a file upload button.

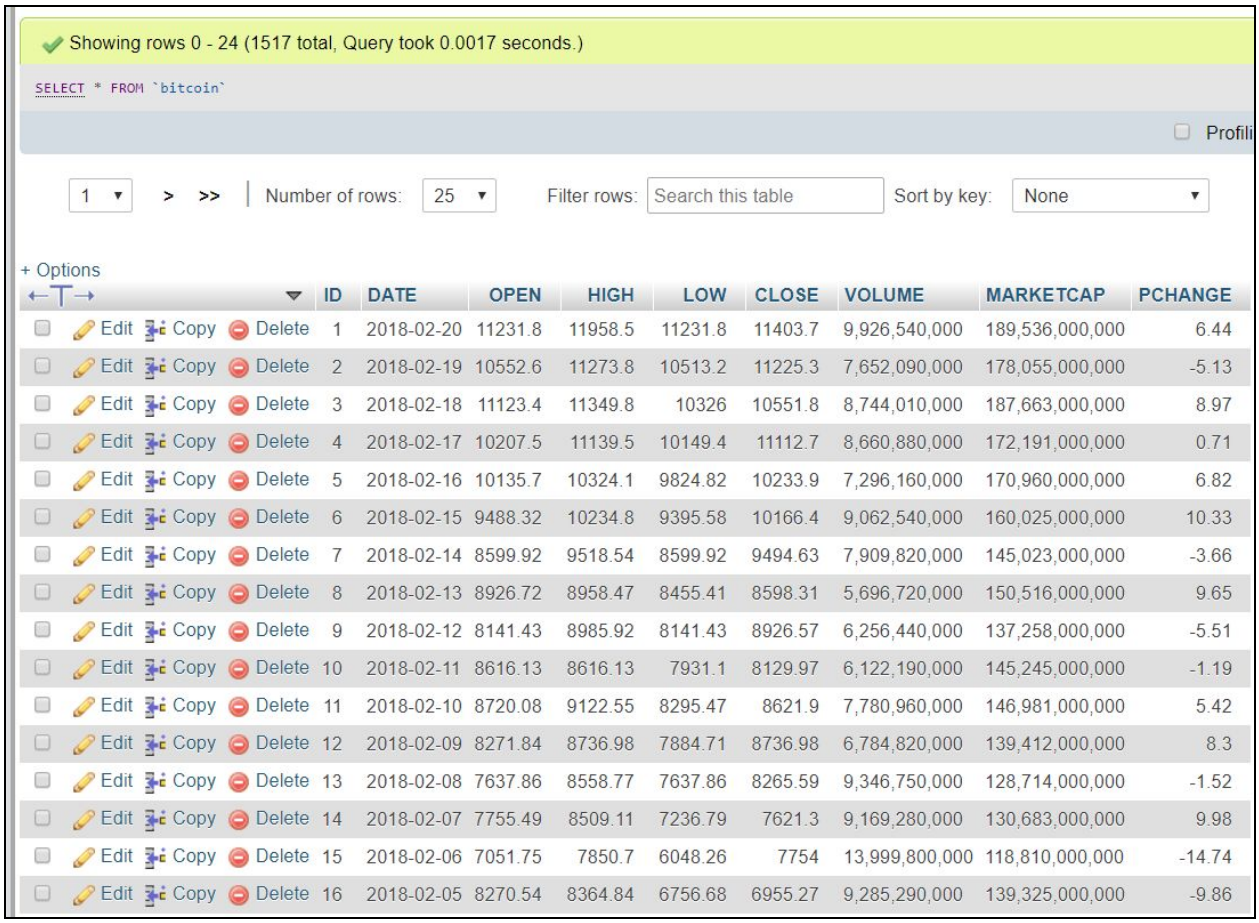
NR 1.2

The application must allow users to switch cryptocurrency tokens from a drop-down menu.

NR 1.3

The application must allow users to switch views between file upload, linear regression, and risk assessment zones.

Example Results & Database



Showing rows 0 - 24 (1517 total, Query took 0.0017 seconds.)

```
SELECT * FROM `bitcoin`
```

Profili

1 > >> | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

	ID	DATE	OPEN	HIGH	LOW	CLOSE	VOLUME	MARKETCAP	PCHANGE
Edit Copy Delete	1	2018-02-20	11231.8	11958.5	11231.8	11403.7	9,926,540,000	189,536,000,000	6.44
Edit Copy Delete	2	2018-02-19	10552.6	11273.8	10513.2	11225.3	7,652,090,000	178,055,000,000	-5.13
Edit Copy Delete	3	2018-02-18	11123.4	11349.8	10326	10551.8	8,744,010,000	187,663,000,000	8.97
Edit Copy Delete	4	2018-02-17	10207.5	11139.5	10149.4	11112.7	8,660,880,000	172,191,000,000	0.71
Edit Copy Delete	5	2018-02-16	10135.7	10324.1	9824.82	10233.9	7,296,160,000	170,960,000,000	6.82
Edit Copy Delete	6	2018-02-15	9488.32	10234.8	9395.58	10166.4	9,062,540,000	160,025,000,000	10.33
Edit Copy Delete	7	2018-02-14	8599.92	9518.54	8599.92	9494.63	7,909,820,000	145,023,000,000	-3.66
Edit Copy Delete	8	2018-02-13	8926.72	8958.47	8455.41	8598.31	5,696,720,000	150,516,000,000	9.65
Edit Copy Delete	9	2018-02-12	8141.43	8985.92	8141.43	8926.57	6,256,440,000	137,258,000,000	-5.51
Edit Copy Delete	10	2018-02-11	8616.13	8616.13	7931.1	8129.97	6,122,190,000	145,245,000,000	-1.19
Edit Copy Delete	11	2018-02-10	8720.08	9122.55	8295.47	8621.9	7,780,960,000	146,981,000,000	5.42
Edit Copy Delete	12	2018-02-09	8271.84	8736.98	7884.71	8736.98	6,784,820,000	139,412,000,000	8.3
Edit Copy Delete	13	2018-02-08	7637.86	8558.77	7637.86	8265.59	9,346,750,000	128,714,000,000	-1.52
Edit Copy Delete	14	2018-02-07	7755.49	8509.11	7236.79	7621.3	9,169,280,000	130,683,000,000	9.98
Edit Copy Delete	15	2018-02-06	7051.75	7850.7	6048.26	7754	13,999,800,000	118,810,000,000	-14.74
Edit Copy Delete	16	2018-02-05	8270.54	8364.84	6756.68	6955.27	9,285,290,000	139,325,000,000	-9.86

CONCLUSIONS

Our team learned a great deal, as this was not only our first foray into data science, but also our first foray into web development. Overall, we would quantify this project as a success because we achieved what we set out to complete. However, there is much room for improvement. Using a PHP oriented development stack is certainly not optimal for data science related tasks. If we were to repeat this project, we would definitely choose Python for its access to a variety of data science packages. Further development may also include:

- Chart.js instead of Google charts (Google charts are easier to use, but are not fully responsive)
- Live ticker API integration for live data streaming (we considered doing this during the project, but the two options that we found were only available via paid subscription)
- Support for more cryptocurrencies
- Full deployment online

APPENDIX

Github Repository

https://github.com/davidxie95/cs161-project/tree/temp

Sprint Details & Updates

Note: we did not use an agile project management tool, we kept our backlogs in an excel sheet

Task	**Status**	**Assign to**	**Point Weight**
Website connects to web server	Complete	Tien	1
Website connects to database	Complete	Tien	1
Website has rough UI	Complete	Tien	1
Read .csv and cut to year	Complete	Tien	3
Generate variables for database storage	Complete	Tien	3
Collaboration tool setup (Discord, Taiga, and Github)	Complete	Geordi	1
Documentation	Complete	Geordi	3
System Testing	Complete	Geordi	3
Database schema setup	Complete	Geordi	3
Store tuples in database	Complete	Geordi	3
Selecting Data Source	Complete	David	3
Selecting Cryptocurrencies	Complete	David	3
Selecting Trading sites	Complete	David	1
Researching analyzing algorithms	Complete	David	2
Sprint 1 - Week 1 & 2			**Total**
			31
Task	**Status**	**Assign to**	**Point Weight**

System Testing (from Sprint 1)	Complete	Geordi	3
Documentation	Complete	Geordi	3
Retrieve graphing variables from database	Complete	David	1
Create Google chart with graphing variables	Complete	David	3
Chart can switch between volume, market cap, or price	Complete	Tien	3
Chart has tabs for linear regression and risk assessment	Complete	Tien	3
Sprint 2 - Week 3 & 4			**Total**
			16
Task	**Status**	**Assign to**	**Point Weight**
System Testing (from Sprint 2)	Complete	Geordi	3
GUI research and testing	Complete	Geordi	3
Live data API selection?	Complete	Geordi	2
Risk assessment algorithm research	Complete	Geordi	2
Documentation	Complete	Geordi	1
Calculate and pass regression variables to database	Complete	Tien	3
Create Google chart displaying linear regression	Complete	Tien	3
Chart tabs between price and linear regression	Complete	David	3
Sprint 3 - Week 5 to 7			**Total**
			20
Task	**Status**	**Assign to**	**Point Weight**
System Testing (from Sprint 3)	Complete	Geordi	3
GUI test version	Complete	Geordi	3
Project can run Python scripts (through Flask or any other means)	Complete	David	2
VAR (Value at risk) model Python script	Complete	David	3
Linear regression is graphically displayed	Complete	Tien	3
Documentation	Complete	Geordi	1

Sprint 4 - Week 8 & 9			**Total**
			18
Task	**Status**	**Assign to**	**Point Weight**
Value at risk completed, fully communicates with database	Complete	David	3
Linear regression completed, fully communicates with database	Complete	Tien	3
Frontend completed, fully communicates with necessary modules	Complete	Geordi	3
Documentation (Final update, report, and test plan)	Complete	Geordi	5
Final presentation slides	Complete	Geordi	2
Sprint 5 - Week 10 to 13			**Total**
			18

Sprint 1 Conclusion Update:

3/8/18 Update - Sprint 1

Cryptocurrency Data Analysis

Geordi Reiner, David Xie, Tien Tran
CS161 - Ghofraniha

Sprint 1 (2 weeks):
Setup of database schema, website framework, collaboration tools, and research on data analysis techniques.

Summary/Recap:
The team has completed all of sprint 1, with the exception of a finished test plan. Testing from sprint 1 will likely continue or be combined with testing in sprint 2. So far, our end-to-end framework is fully connected and we have managed to “clean” and store data from our .csv files. Looking forward we are familiarizing ourselves with Google charts API and data science techniques required for sprints 2 and 3 respectively. Next week’s primary task will be to graph the “cleaned” data using google charts.

Task	Status	Assign to	Point Weight
Website connects to web server	Complete	Tien	1
Website connects to database	Complete	Tien	1
Website has rough UI	Complete	Tien	1
Read .csv and cut to year	Complete	Tien	3
Generate variables for database storage	Complete	Tien	3
Collaboration tool setup (Discord, Taiga, and Github)	Complete	Geordi	1
Documentation	Complete	Geordi	3
System Testing	In Progress	Geordi	3
Database schema setup	Complete	Geordi	3
Store tuples in database	Complete	Geordi	3
Selecting Data Source	Complete	David	3
Selecting Cryptocurrencies	Complete	David	3
Selecting Trading sites	Complete	David	1
Researching analyzing algorithms	Complete	David	2
Sprint 1 - Week 1 & 2			**Total**
			31

Sprint 2, 3, and 4 do not have conclusion updates as they were completed after the mid-semester project evaluation. Here is the penultimate update:

3/15/18 Update - Sprint 2

Cryptocurrency Data Analysis

Geordi Reiner, David Xie, Tien Tran

CS161 - Ghofraniha

Sprint 2 (2 weeks): Present at least one cryptocurrency's price fluctuations and other relevant trading data graphically.

Summary/Recap: Despite being half way through sprint 2's duration, no work was completed on the project, as the team had a high concentration of midterms. Thankfully, sprint 2 is likely our easiest sprint, so its chance of completion is still high. A completed test plan of features from sprint 1 was moved to sprint 2's backlogs to even out work distribution.

Task	Status	Assign to	Point Weight
System Testing (from Sprint 1)	In Progress	Geordi	3
Documentation	In Progress	Geordi	3
Retrieve graphing variables from database	In Progress	David	1
Create Google chart with graphing variables	In Progress	David	3
Chart can switch between volume, market cap, or price	In Progress	Tien	3
Chart has tabs for linear regression and risk assessment	In Progress	Tien	3
Sprint 2 - Week 3 & 4			**Total**
			16

Sprint 5 conclusion update:

Final Update - Sprint 5

Cryptocurrency Data Analysis

Geordi Reiner, David Xie, Tien Tran
CS161 - Ghofraniha

Sprint 5 (2 weeks): Finalize GUI and combine code bases.

Summary/Recap: For this final sprint, most of our effort is being devoted to combining our 3 code bases (Tien's database, linear regression, and backend, David's VaR and test environment, and Geordi's frontend). Last class we announced that we had some issues with implementing a VaR model, so we decided to meet and develop an "experimental Python-less" version for your review. We are still in the process of combining our code, so unforeseen bugs may still arise. Nevertheless, we have learned a lot and have much to present when Thursday comes.

Task	Status	Assign to	Point Weight
Value at risk completed, fully communicates with database	In Progress	David	3
Linear regression completed, fully communicates with database	Complete	Tien	3
Frontend completed, fully communicates with necessary modules	Complete	Geordi	3
Documentation (Final update, report, and test plan)	In Progress	Geordi	5
Final presentation slides	In Progress	Geordi	2

Sprint 5 - Week 10 to 13			**Total**
			18