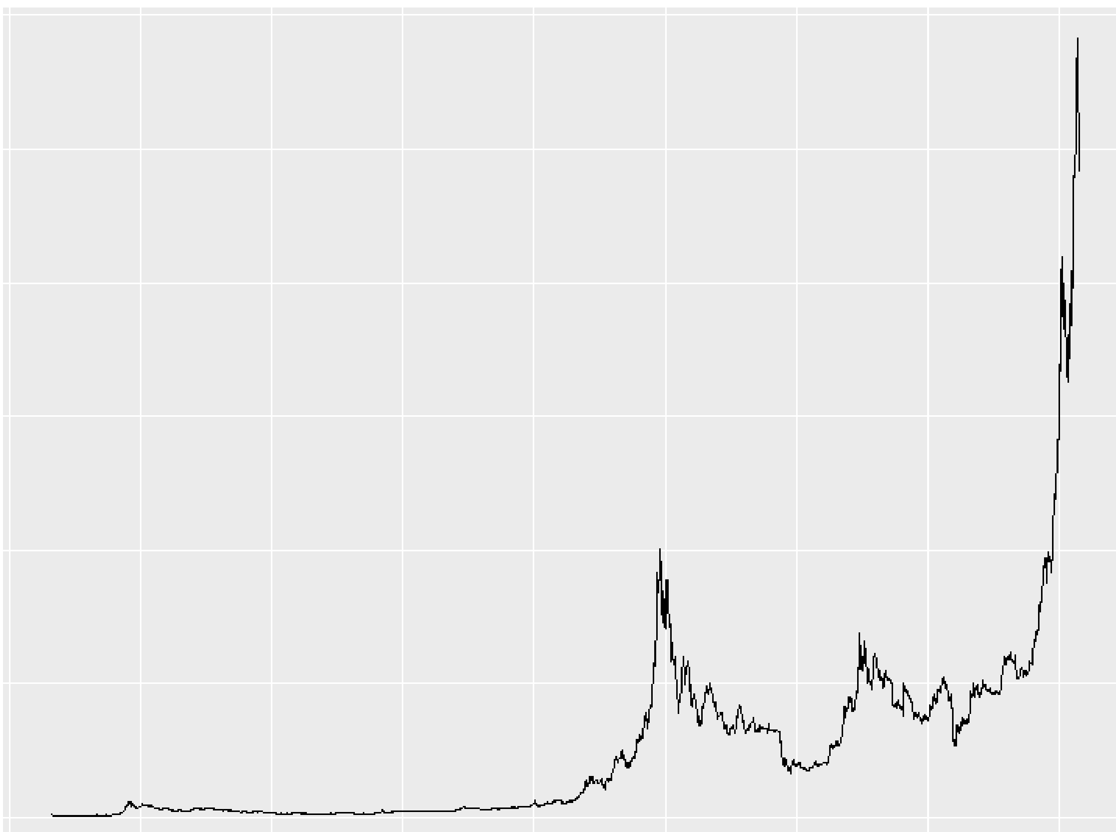
Understanding Cryptocurrency Volatility

Exploring Correlations and Building Predictive Models



Syracuse Master's of Applied Data Science Program

Huang, Cliff
Onyeugbo, Glory

Table of Contents

I. INTRODUCTION

High Level Summary

Cryptocurrencies are digital assets designed to be mediums of exchange that use files as money. Cryptocurrencies use decentralized control as opposed to centralized electronic money and central banking systems.The first and most popular cryptocurrency, Bitcoin, was invented in 2008. Following the success of Bitcoin, alternatives to the currency were launched and given the term 'altcoins', which refers to any cryptocurrencies other than Bitcoin. Cryptocurrencies can be transferred from one owner to another directly, or indirectly via an exchange, which functions like a stock market exchange.

Dataset

Subset of Cryptocurrency Historical Prices, which are directly pulled from coinmarketcap (see subset details below).

Number of Data Points

202,770 (20,277 observations of 10 variables)

Subset Details

Each dataset represents a different cryptocurrency and contains observations of the same 10 variables, which include price, volume and market capitalization over time.

Bitcoin: 2862 observations of 10 variables
Cardano: 1245 observations of 10 variables
Dogecoin: 2631 observations of 10 variables
EOS: 1337 observations of 10 variables
Ethereum: 2031 observations of 10 variables
Iota: 1355 observations of 10 variables
Litecoin: 2682 observations of 10 variables
Monero: 2473 observations of 10 variables
Stellar: 2398 observations of 10 variables
Tron: 1263 observations of 10 variables

Scope and Context of Analysis

The goal of this project is to understand the factors that impact the price movement of several cryptocurrencies in exchanges. We will focus our analysis on ten currencies: Bitcoin, Cardano, Dogecoin, EOS, Ethereum, Iota, Litecoin, Monero, Stellar, and Tron. We will limit our analysis to relatively few variables, which include date, price, volume, market capitalization, and a few calculated variables derived from price and volume.

Business Questions

1. What correlations can we find between the prices of Bitcoin and altcoins?
2. Is there a relationship between trading volume and price?
3. Can we develop a model to predict the price of a coin over a given period of time?
4. Are there outliers in price movements? If so, what can we identify about them?

II. DATA ACQUISITION AND MUNGING

R Studio script is located in the Appendix

Data Acquisition

We selected this source because the data sets are freely available to use, current as of early 2021, and highly organized. Of the 23 data sets available, we focused on a subset of 9 because we are particularly interested in cryptocurrencies with high lifetime returns, high market capitalization, and/or high volatility. We also chose these data sets based on personal interest in each coin's underlying technology and value proposition.

We downloaded the datasets in .csv format from the aforementioned URL; the compiler of the data used Python to access coinmarketcap's API to pull historical data on select coins. We then read each .csv into R Studio for cleansing, transforming, and munging.

Data Dictionary

Attribute	Data Type	Definition
SNo	integer	Observation identity
Name	character	Cryptocurrency name
Symbol	character	Cryptocurrency trading symbol
Date	character	Datetime of observation, in UTC
Open	numeric	opening price on given day
High	numeric	highest price on given day
Low	numeric	lowest price on given day
Close	numeric	closing price on given day (define the exact time)
Volume:	numeric	Volume of transactions on the given day
Market Cap	numeric	Market Capitalization in USD Market Cap = High * Circulating Supply
Daily Return	numeric	Percent return over 24 hours Daily Return = 100 * ((Close - Open)/Open)
Daily Volume Change	numeric	Percent change in volume over 24 hours Value = 100 * ((Volume - (Previous Day's Volume))/Previous Day's Volume
Big Day	numeric	Day in which daily return exceeds 5%

Structure and Summary of Raw Data

```
> str(bitcoin)
'data.frame':   2862 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Bitcoin" "Bitcoin" "Bitcoin" "Bitcoin" ...
 $ Symbol   : chr  "BTC" "BTC" "BTC" "BTC" ...
 $ Date     : chr  "2013-04-29 23:59:59" "2013-04-30 23:59:59" "2013-05-01 23:59:59" "2013-05-02 23:59:59" ...
 $ High     : num  147 147 140 126 108 ...
 $ Low      : num  134 134.1 107.7 92.3 79.1 ...
 $ Open     : num  134 144 139 116 106 ...
 $ Close    : num  144.5 139 117 105.2 97.8 ...
 $ Volume   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Marketcap: num  1.60e+09 1.54e+09 1.30e+09 1.17e+09 1.09e+09 ...
> summary(bitcoin)
      SNo             Name              Symbol              Date               High
 Min.   :   1.0   Length:2862        Length:2862        Length:2862        Min.   :   74.56
 1st Qu.: 716.2   Class :character   Class :character   Class :character   1st Qu.:  426.05
 Median :1431.5   Mode  :character   Mode  :character   Mode  :character   Median : 1197.33
 Mean   :1431.5                                                            Mean   : 4974.04
 3rd Qu.:2146.8                                                            3rd Qu.: 8138.05
 Max.   :2862.0                                                            Max.   :58330.57
      Low               Open             Close              Volume            Marketcap
 Min.   :   65.53   Min.   :   68.5   Min.   :   68.43   Min.   :0.000e+00   Min.   :7.784e+08
 1st Qu.:  415.68   1st Qu.:  421.2   1st Qu.:  420.99   1st Qu.:2.786e+07   1st Qu.:5.989e+09
 Median : 1164.17   Median : 1180.1   Median : 1182.81   Median :3.302e+08   Median :1.924e+10
 Mean   : 4695.10   Mean   : 4836.3   Mean   : 4852.09   Mean   :8.978e+09   Mean   :8.592e+10
 3rd Qu.: 7703.36   3rd Qu.: 7924.6   3rd Qu.: 7926.70   3rd Qu.:1.297e+10   3rd Qu.:1.388e+11
 Max.   :55672.61   Max.   :57532.7   Max.   :57539.94   Max.   :3.510e+11   Max.   :1.072e+12
```

```
> str(cardano)
'data.frame':   1245 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Cardano" "Cardano" "Cardano" "Cardano" ...
 $ Symbol   : chr  "ADA" "ADA" "ADA" "ADA" ...
 $ Date     : chr  "2017-10-02 23:59:59" "2017-10-03 23:59:59" "2017-10-04 23:59:59" "2017-10-05 23:59:59" ...
 $ High     : num  0.0301 0.0274 0.0228 0.0222 0.0215 ...
 $ Low      : num  0.02 0.0207 0.0209 0.0209 0.0184 ...
 $ Open     : num  0.0246 0.0258 0.0209 0.022 0.0214 ...
 $ Close    : num  0.0259 0.0208 0.0219 0.0215 0.0185 ...
 $ Volume   : num  57641300 16997800 9000050 5562510 7780710 ...
 $ Marketcap: num  6.29e+08 5.40e+08 5.69e+08 5.57e+08 4.81e+08 ...
> summary(cardano)
      SNo           Name              Symbol              Date               High
 Min.   :   1   Length:1245        Length:1245        Length:1245        Min.   :0.02105
 1st Qu.: 312   Class :character   Class :character   Class :character   1st Qu.:0.04557
 Median : 623   Mode  :character   Mode  :character   Mode  :character   Median :0.08351
 Mean   : 623                                                            Mean   :0.14605
 3rd Qu.: 934                                                            3rd Qu.:0.14931
 Max.   :1245                                                            Max.   :1.47665
      Low              Open             Close             Volume            Marketcap
 Min.   :0.01762   Min.   :0.01841   Min.   :0.01854   Min.   :1.739e+06   Min.   :4.807e+08
 1st Qu.:0.04298   1st Qu.:0.04429   1st Qu.:0.04433   1st Qu.:4.642e+07   1st Qu.:1.149e+09
 Median :0.07763   Median :0.08087   Median :0.08095   Median :1.012e+08   Median :2.102e+09
 Mean   :0.12934   Mean   :0.13776   Mean   :0.13876   Mean   :4.730e+08   Mean   :3.788e+09
 3rd Qu.:0.13810   3rd Qu.:0.14309   3rd Qu.:0.14357   3rd Qu.:2.577e+08   3rd Qu.:3.780e+09
 Max.   :1.23414   Max.   :1.24215   Max.   :1.32486   Max.   :1.914e+10   Max.   :4.122e+10
```

```
> str(dogecoin)
'data.frame':   2631 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Dogecoin" "Dogecoin" "Dogecoin" "Dogecoin" ...
 $ Symbol   : chr  "DOGE" "DOGE" "DOGE" "DOGE" ...
 $ Date     : chr  "2013-12-16 23:59:59" "2013-12-17 23:59:59" "2013-12-18 23:59:59" "2013-12-19 23:59:59" ...
 $ High     : num  0.000866 0.000289 0.000362 0.00152 0.001143 ...
 $ Low      : num  0.00015 0.000116 0.000205 0.000328 0.000662 ...
 $ Open     : num  0.000299 0.000207 0.000267 0.000395 0.001143 ...
 $ Close    : num  0.000205 0.000269 0.000362 0.001162 0.000704 ...
 $ Volume   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Marketcap: num  1509085 2169688 3188943 11150339 7284337 ...
> summary(dogecoin)
      SNo             Name              Symbol              Date               High
 Min.   :   1.0   Length:2631        Length:2631        Length:2631        Min.   :8.919e-05
 1st Qu.: 658.5   Class :character   Class :character   Class :character   1st Qu.:2.288e-04
 Median :1316.0   Mode  :character   Mode  :character   Mode  :character   Median :1.645e-03
 Mean   :1316.0                                                            Mean   :2.566e-03
 3rd Qu.:1973.5                                                            3rd Qu.:2.810e-03
 Max.   :2631.0                                                            Max.   :8.495e-02
      Low                Open               Close               Volume            Marketcap
 Min.   :8.547e-05   Min.   :8.715e-05   Min.   :8.727e-05   Min.   :0.000e+00   Min.   :1.509e+06
 1st Qu.:2.197e-04   1st Qu.:2.241e-04   1st Qu.:2.243e-04   1st Qu.:2.339e+05   1st Qu.:2.315e+07
 Median :1.387e-03   Median :1.523e-03   Median :1.523e-03   Median :4.485e+06   Median :1.553e+08
 Mean   :2.243e-03   Mean   :2.394e-03   Mean   :2.410e-03   Mean   :1.063e+08   Mean   :2.875e+08
 3rd Qu.:2.625e-03   3rd Qu.:2.710e-03   3rd Qu.:2.714e-03   3rd Qu.:3.864e+07   3rd Qu.:3.299e+08
 Max.   :6.853e-02   Max.   :7.877e-02   Max.   :7.883e-02   Max.   :2.540e+10   Max.   :1.011e+10
```

```
> str(ethereum)
'data.frame':   2031 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Ethereum" "Ethereum" "Ethereum" "Ethereum" ...
 $ Symbol   : chr  "ETH" "ETH" "ETH" "ETH" ...
 $ Date     : chr  "2015-08-08 23:59:59" "2015-08-09 23:59:59" "2015-08-10 23:59:59" "2015-08-11 23:59:59" ...
 $ High     : num  2.8 0.88 0.73 1.13 1.29 ...
 $ Low      : num  0.715 0.629 0.637 0.663 0.884 ...
 $ Open     : num  2.794 0.706 0.714 0.708 1.059 ...
 $ Close    : num  0.753 0.702 0.708 1.068 1.217 ...
 $ Volume   : num  674188 532170 405283 1463100 2150620 ...
 $ Marketcap: num  45486894 42399573 42818364 64569288 73645011 ...
> summary(ethereum)
      SNo             Name              Symbol              Date               High         
 Min.   :   1.0   Length:2031        Length:2031        Length:2031        Min.   :   0.483  
 1st Qu.: 508.5   Class :character   Class :character   Class :character   1st Qu.:  13.222  
 Median :1016.0   Mode  :character   Mode  :character   Mode  :character   Median : 189.089  
 Mean   :1016.0                                                            Mean   : 267.068  
 3rd Qu.:1523.5                                                            3rd Qu.: 342.239  
 Max.   :2031.0                                                            Max.   :2036.286  
      Low                Open               Close              Volume             Marketcap        
 Min.   :   0.4209   Min.   :   0.4316   Min.   :   0.4348   Min.   :1.021e+05   Min.   :3.221e+07  
 1st Qu.:  12.5151   1st Qu.:  12.8582   1st Qu.:  12.8661   1st Qu.:2.849e+07   1st Qu.:1.086e+09  
 Median : 180.3172   Median : 184.9434   Median : 185.0287   Median :1.844e+09   Median :1.973e+10  
 Mean   : 246.9864   Mean   : 257.4906   Mean   : 258.1423   Mean   :5.430e+09   Mean   :2.699e+10  
 3rd Qu.: 317.6260   3rd Qu.: 332.7165   3rd Qu.: 333.3705   3rd Qu.:7.944e+09   3rd Qu.:3.280e+10  
 Max.   :1896.6845   Max.   :1959.9030   Max.   :1960.1647   Max.   :6.073e+10   Max.   :2.249e+11  

> str(eos)
'data.frame':   1337 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "EOS" "EOS" "EOS" "EOS" ...
 $ Symbol   : chr  "EOS" "EOS" "EOS" "EOS" ...
 $ Date     : chr  "2017-07-02 23:59:59" "2017-07-03 23:59:59" "2017-07-04 23:59:59" "2017-07-05 23:59:59" ...
 $ High     : num  2.88 5.4 4.19 3.52 3.87 ...
 $ Low      : num  0.823 2.632 2.933 2.73 3.008 ...
 $ Open     : num  0.997 2.717 4.098 3.356 3.014 ...
 $ Close    : num  2.71 4.09 3.37 3 3.36 ...
 $ Volume   : num  3.20e+08 4.15e+08 2.19e+08 1.24e+08 1.77e+08 ...
 $ Marketcap: num  0.00 6.55e+08 5.50e+08 5.01e+08 5.68e+08 ...
> summary(eos)
      SNo           Name              Symbol              Date               High        
 Min.   :   1   Length:1337        Length:1337        Length:1337        Min.   : 0.5187  
 1st Qu.: 335   Class :character   Class :character   Class :character   1st Qu.: 2.6612  
 Median : 669   Mode  :character   Mode  :character   Mode  :character   Median : 3.4979  
 Mean   : 669                                                            Mean   : 4.7148  
 3rd Qu.:1003                                                            3rd Qu.: 5.6617  
 Max.   :1337                                                            Max.   :22.8904  
      Low              Open             Close             Volume             Marketcap        
 Min.   : 0.4802   Min.   : 0.4928   Min.   : 0.4932   Min.   :4.557e+06   Min.   :0.000e+00  
 1st Qu.: 2.5285   1st Qu.: 2.6005   1st Qu.: 2.6025   1st Qu.:6.466e+08   1st Qu.:2.403e+09  
 Median : 3.2239   Median : 3.3539   Median : 3.3613   Median :1.501e+09   Median :3.083e+09  
 Mean   : 4.2981   Mean   : 4.5173   Mean   : 4.5182   Mean   :1.725e+09   Mean   :3.795e+09  
 3rd Qu.: 5.3369   3rd Qu.: 5.4843   3rd Qu.: 5.4710   3rd Qu.:2.405e+09   3rd Qu.:4.870e+09  
 Max.   :18.8774   Max.   :21.6391   Max.   :21.5426   Max.   :1.001e+10   Max.   :1.777e+10  

> str(iota)
'data.frame':   1355 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "IOTA" "IOTA" "IOTA" "IOTA" ...
 $ Symbol   : chr  "MIOTA" "MIOTA" "MIOTA" "MIOTA" ...
 $ Date     : chr  "2017-06-14 23:59:59" "2017-06-15 23:59:59" "2017-06-16 23:59:59" "2017-06-17 23:59:59" ...
 $ High     : num  0.606 0.543 0.448 0.444 0.426 ...
 $ Low      : num  0.496 0.3 0.31 0.414 0.394 ...
 $ Open     : num  0.592 0.528 0.353 0.427 0.421 ...
 $ Close    : num  0.529 0.364 0.411 0.42 0.406 ...
 $ Volume   : num  14194900 10300400 6920690 3100660 2514450 ...
 $ Marketcap: num  1.47e+09 1.01e+09 1.14e+09 1.17e+09 1.13e+09 ...
> summary(iota)
      SNo             Name              Symbol              Date               High       
 Min.   :   1.0   Length:1355        Length:1355        Length:1355        Min.   :0.1247  
 1st Qu.: 339.5   Class :character   Class :character   Class :character   1st Qu.:0.2727  
 Median : 678.0   Mode  :character   Mode  :character   Mode  :character   Median :0.3498  
 Mean   : 678.0                                                            Mean   :0.6975  
 3rd Qu.:1016.5                                                            3rd Qu.:0.6201  
 Max.   :1355.0                                                            Max.   :5.6900  
      Low               Open            Close            Volume             Marketcap        
 Min.   :0.07962   Min.   :0.1103   Min.   :0.1102   Min.   :2.335e+06   Min.   :3.063e+08  
 1st Qu.:0.25387   1st Qu.:0.2651   1st Qu.:0.2650   1st Qu.:9.420e+06   1st Qu.:7.367e+08  
 Median :0.31729   Median :0.3336   Median :0.3331   Median :1.696e+07   Median :9.258e+08  
 Mean   :0.61568   Mean   :0.6590   Mean   :0.6591   Mean   :4.642e+07   Mean   :1.832e+09  
 3rd Qu.:0.55826   3rd Qu.:0.5858   3rd Qu.:0.5876   3rd Qu.:4.089e+07   3rd Qu.:1.633e+09  
 Max.   :4.86136   Max.   :5.3711   Max.   :5.3663   Max.   :2.127e+09   Max.   :1.492e+10  
```

```
> str(litecoin)
'data.frame':	2862 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Litecoin" "Litecoin" "Litecoin" "Litecoin" ...
 $ Symbol   : chr  "LTC" "LTC" "LTC" "LTC" ...
 $ Date     : chr  "2013-04-29 23:59:59" "2013-04-30 23:59:59" "2013-05-01 23:59:59" "2013-05-02 23:59:59" ...
 $ High     : num  4.57 4.57 4.36 4.04 3.45 ...
 $ Low      : num  4.23 4.17 3.52 3.01 2.4 ...
 $ Open     : num  4.37 4.4 4.29 3.78 3.39 ...
 $ Close    : num  4.38 4.3 3.8 3.37 3.04 ...
 $ Volume   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Marketcap: num  75388964 74020918 65604596 58287979 52694847 ...
> summary(litecoin)
      SNo             Name              Symbol              Date                High        
 Min.   :   1.0   Length:2862        Length:2862        Length:2862        Min.   :  1.345  
 1st Qu.: 716.2   Class :character   Class :character   Class :character   1st Qu.:  3.815  
 Median :1431.5   Mode  :character   Mode  :character   Mode  :character   Median : 25.121  
 Mean   :1431.5                                                            Mean   : 43.628  
 3rd Qu.:2146.8                                                            3rd Qu.: 60.116  
 Max.   :2862.0                                                            Max.   :375.286  
      Low               Open              Close              Volume            Marketcap        
 Min.   :  1.114   Min.   :  1.153   Min.   :  1.157   Min.   :0.000e+00   Min.   :3.789e+07  
 1st Qu.:  3.682   1st Qu.:  3.753   1st Qu.:  3.753   1st Qu.:2.121e+06   1st Qu.:1.550e+08  
 Median : 22.683   Median : 23.853   Median : 23.934   Median :4.492e+07   Median :6.311e+08  
 Mean   : 40.058   Mean   : 41.907   Mean   : 41.956   Mean   :1.119e+09   Mean   :2.429e+09  
 3rd Qu.: 56.966   3rd Qu.: 58.384   3rd Qu.: 58.476   3rd Qu.:1.616e+09   3rd Qu.:3.599e+09  
 Max.   :335.675   Max.   :359.125   Max.   :358.336   Max.   :1.799e+10   Max.   :1.948e+10  

> str(monero)
'data.frame':	2473 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Monero" "Monero" "Monero" "Monero" ...
 $ Symbol   : chr  "XMR" "XMR" "XMR" "XMR" ...
 $ Date     : chr  "2014-05-22 23:59:59" "2014-05-23 23:59:59" "2014-05-24 23:59:59" "2014-05-25 23:59:59" ...
 $ High     : num  2.19 3.43 4.01 4.04 3.76 ...
 $ Low      : num  1.36 2.05 2.62 2.8 2.31 ...
 $ Open     : num  1.59 2.05 2.92 4.04 3.22 ...
 $ Close    : num  2.1 2.96 3.7 3.14 3.02 ...
 $ Volume   : num  132918 266852 248028 283545 220148 ...
 $ Marketcap: num  1855608 2692180 3456658 2993339 2946133 ...
> summary(monero)
      SNo           Name              Symbol              Date                High         
 Min.   :   1   Length:2473        Length:2473        Length:2473        Min.   :  0.2518  
 1st Qu.: 619   Class :character   Class :character   Class :character   1st Qu.:  1.5139  
 Median :1237   Mode  :character   Mode  :character   Mode  :character   Median : 49.4574  
 Mean   :1237                                                            Mean   : 65.8049  
 3rd Qu.:1855                                                            3rd Qu.: 95.2534  
 Max.   :2473                                                            Max.   :495.8360  
      Low               Open               Close               Volume            Marketcap        
 Min.   :  0.213   Min.   :  0.2201   Min.   :  0.2235   Min.   :7.900e+03   Min.   :1.280e+06  
 1st Qu.:  1.308   1st Qu.:  1.4372   1st Qu.:  1.4457   1st Qu.:1.434e+05   1st Qu.:6.721e+06  
 Median : 46.010   Median : 47.6853   Median : 47.8302   Median :1.993e+07   Median :7.802e+08  
 Mean   : 60.210   Mean   : 63.1191   Mean   : 63.1908   Mean   :1.440e+08   Mean   :1.036e+09  
 3rd Qu.: 88.952   3rd Qu.: 92.1754   3rd Qu.: 92.2993   3rd Qu.:9.051e+07   3rd Qu.:1.550e+09  
 Max.   :453.391   Max.   :470.2870   Max.   :469.1980   Max.   :2.896e+10   Max.   :7.274e+09  

> str(stellar)
'data.frame':	2398 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "Stellar" "Stellar" "Stellar" "Stellar" ...
 $ Symbol   : chr  "XLM" "XLM" "XLM" "XLM" ...
 $ Date     : chr  "2014-08-06 23:59:59" "2014-08-07 23:59:59" "2014-08-08 23:59:59" "2014-08-09 23:59:59" ...
 $ High     : num  0.0034 0.00304 0.00324 0.00371 0.00762 ...
 $ Low      : num  0.00227 0.00245 0.00249 0.00287 0.00336 ...
 $ Open     : num  0.00237 0.00269 0.00249 0.00288 0.0035 ...
 $ Close    : num  0.00266 0.0025 0.00288 0.00348 0.0043 ...
 $ Volume   : num  35820 142864 93708 233579 352737 ...
 $ Marketcap: num  966370 1079148 1260109 1645185 2216012 ...
> summary(stellar)
      SNo             Name              Symbol              Date                High         
 Min.   :   1.0   Length:2398        Length:2398        Length:2398        Min.   :0.001509  
 1st Qu.: 600.2   Class :character   Class :character   Class :character   1st Qu.:0.002495  
 Median :1199.5   Mode  :character   Mode  :character   Mode  :character   Median :0.043371  
 Mean   :1199.5                                                            Mean   :0.087763  
 3rd Qu.:1798.8                                                            3rd Qu.:0.108561  
 Max.   :2398.0                                                            Max.   :0.938144  
      Low                Open               Close               Volume            Marketcap        
 Min.   :0.001227   Min.   :0.001352   Min.   :0.001357   Min.   :4.910e+02   Min.   :9.664e+05  
 1st Qu.:0.002280   1st Qu.:0.002374   1st Qu.:0.002373   1st Qu.:4.307e+04   1st Qu.:1.240e+07  
 Median :0.038482   Median :0.040443   Median :0.040761   Median :3.359e+07   Median :6.580e+08  
 Mean   :0.078909   Mean   :0.083358   Mean   :0.083527   Mean   :1.788e+08   Mean   :1.590e+09  
 3rd Qu.:0.102993   3rd Qu.:0.105978   3rd Qu.:0.106007   3rd Qu.:1.906e+08   3rd Qu.:2.078e+09  
 Max.   :0.677496   Max.   :0.892399   Max.   :0.896227   Max.   :1.041e+10   Max.   :1.602e+10  
```

```
> str(tron)
'data.frame':	1263 obs. of  10 variables:
 $ SNo      : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Name     : chr  "TRON" "TRON" "TRON" "TRON" ...
 $ Symbol   : chr  "TRX" "TRX" "TRX" "TRX" ...
 $ Date     : chr  "2017-09-14 23:59:59" "2017-09-15 23:59:59" "2017-09-16 23:59:59" "2017-09-17 23:59:59" ...
 $ High     : num  0.00248 0.00183 0.01767 0.00388 0.00344 ...
 $ Low      : num  0.0013 0.00109 0.00137 0.00178 0.00253 ...
 $ Open     : num  0.00201 0.00142 0.00182 0.00184 0.00332 ...
 $ Close    : num  0.00143 0.0018 0.00183 0.00332 0.00326 ...
 $ Volume   : num  58770 26475 54535 124998 159158 ...
 $ Marketcap: num  0 0 0 0 0 0 0 0 0 0 ...
> summary(tron)
      SNo             Name              Symbol              Date               High         
 Min.   :   1.0   Length:1263        Length:1263        Length:1263        Min.   :0.001831  
 1st Qu.: 316.5   Class :character   Class :character   Class :character   1st Qu.:0.016729  
 Median : 632.0   Mode  :character   Mode  :character   Mode  :character   Median :0.024156  
 Mean   : 632.0                                                            Mean   :0.028420  
 3rd Qu.: 947.5                                                            3rd Qu.:0.032845  
 Max.   :1263.0                                                            Max.   :0.300363  
      Low               Open              Close              Volume            Marketcap        
 Min.   :0.001091   Min.   :0.001419   Min.   :0.001427   Min.   :2.648e+04   Min.   :0.000e+00  
 1st Qu.:0.015794   1st Qu.:0.016193   1st Qu.:0.016201   1st Qu.:1.790e+08   1st Qu.:1.080e+09  
 Median :0.022816   Median :0.023491   Median :0.023523   Median :5.624e+08   Median :1.562e+09  
 Mean   :0.025369   Mean   :0.026909   Mean   :0.026931   Mean   :7.331e+08   Mean   :1.808e+09  
 3rd Qu.:0.029708   3rd Qu.:0.031294   3rd Qu.:0.031351   3rd Qu.:1.092e+09   3rd Qu.:2.175e+09  
 Max.   :0.176712   Max.   :0.224499   Max.   :0.220555   Max.   :6.184e+09   Max.   :1.450e+10  
```

Initial Quality Assessment

There are no NAs in these datasets, and most variables are formatted correctly. However, for some coins, there are dates where trading is taking place, as indicated by volume and/or price changes, but there is no market cap, or no volume, usually early in the life of the coin. It is unclear why, and this appears to be an issue with coinmarketcap's data directly. We will take care to avoid such observations.

We are going to keep all but one variable, SNo. SNo is the observation identity, and does not provide us with any value. We will transform date from character—it is currently a string—into a date datatype. We are especially interested in tracking changes in price and volume over time, within and among coins, in short, medium and long-term. To this end, we want to add some calculated fields, including daily return, daily change in volume, and volatile trading days, to look for interesting correlations and build our models.

Munging

We started with ten data frames, one for each cryptocurrency. Since we wanted to analyze relationships between cryptocurrencies, and our data are time-sensitive, we combined all the data frames into one, and changed the date from character to date datatype. To calculate the dimensions of this new dataframe, we identified our cryptocurrency dataset with the fewest—Cardano—and removed every row from all other cryptocurrency datasets with dates earlier than Cardano's first observation date of 2017-10-02. We merged the dataframes, which now had the same dimensions, into one large dataframe with one date column, no SNo columns, and every other column from the separate datasets.

We then created three functions which we then use to create our calculated columns for daily percent change in price (daily_return), daily percent change in volume, (daily_volume_change) and days with larger than 5% daily returns (big_day) as follows:

```
# create a function to calculate the percent change over 24 hours between the values in any two columns
daily_change <- function(current_value, previous_value) {
  DailyPercentChange <- 100*(current_value - previous_value)/previous_value
  return(DailyPercentChange)
}

# create a function to calculate daily percent change in volume
daily_volume_change <- function(coin.vol){
  volume_change <- daily_change(coin.vol, c(coin.vol[1], coin.vol[-length(coin.vol)]))
  return(volume_change)
}

# create a function to define 'big day' as a day with at least 5% returns
isBigDay <- function(col){
  returncol <- c()
  for(i in 1:length(col)){
    if(col[i]>=5){
      returncol[i] <- 1
    } else {
      returncol[i] <- 0
    }
  }
  return(returncol)
}
```

We also created a second dataframe in order to do specific analysis on the volatility of price by coin. We melted the newly created merged dataframe into a new dataframe with three variables: date, cryptocurrency name as factor, and daily return (listed as “value”).

Starting on the next page is a summary of the new data sets, followed by some interesting initial discoveries we found in the data before performing further analysis.

Structure and Summary of Munged Data

```
> summary(cryptos)
      date             btc.symbol           btc.high        btc.low         btc.open        btc.close        btc.vol              btc.mktcap
 Min.   :2017-10-02   Length:1245        Min.   : 3275   Min.   : 3191   Min.   : 3236   Min.   : 3237   Min.   :9.069e+08   Min.   :5.640e+10
 1st Qu.:2018-08-09   Class :character   1st Qu.: 6644   1st Qu.: 6450   1st Qu.: 6551   1st Qu.: 6550   1st Qu.:5.800e+09   1st Qu.:1.132e+11
 Median :2019-06-16   Mode  :character   Median : 8722   Median : 8267   Median : 8522   Median : 8551   Median :1.585e+10   Median :1.497e+11
 Mean   :2019-06-16                      Mean   :10484   Mean   : 9893   Mean   :10193   Mean   :10226   Mean   :2.039e+10   Mean   :1.833e+11
 3rd Qu.:2020-04-22                      3rd Qu.:10842   3rd Qu.:10266   3rd Qu.:10578   3rd Qu.:10583   3rd Qu.:2.804e+10   3rd Qu.:1.893e+11
 Max.   :2021-02-27                      Max.   :58331   Max.   :55673   Max.   :57533   Max.   :57540   Max.   :3.510e+11   Max.   :1.072e+12
  ada.symbol           ada.high           ada.low            ada.open           ada.close           ada.vol             ada.mktcap
 Length:1245        Min.   :0.02105   Min.   :0.01762   Min.   :0.01841   Min.   :0.01854   Min.   :1.739e+06   Min.   :4.807e+08
 Class :character   1st Qu.:0.04557   1st Qu.:0.04298   1st Qu.:0.04429   1st Qu.:0.04433   1st Qu.:4.642e+07   1st Qu.:1.149e+09
 Mode  :character   Median :0.08351   Median :0.07763   Median :0.08087   Median :0.08095   Median :1.012e+08   Median :2.102e+09
                    Mean   :0.14605   Mean   :0.12934   Mean   :0.13776   Mean   :0.13876   Mean   :4.730e+08   Mean   :3.788e+09
                    3rd Qu.:0.14931   3rd Qu.:0.13810   3rd Qu.:0.14309   3rd Qu.:0.14357   3rd Qu.:2.577e+08   3rd Qu.:3.780e+09
                    Max.   :1.47665   Max.   :1.23414   Max.   :1.24215   Max.   :1.32486   Max.   :1.914e+10   Max.   :4.122e+10
 doge.symbol          doge.high            doge.low             doge.open            doge.close            doge.vol            doge.mktcap
 Length:1245        Min.   :0.001047   Min.   :0.0009381   Min.   :0.0009804   Min.   :0.0009907   Min.   :1.072e+06   Min.   :1.104e+08
 Class :character   1st Qu.:0.002414   1st Qu.:0.0022978   1st Qu.:0.0023550   1st Qu.:0.0023588   1st Qu.:1.590e+07   1st Qu.:2.827e+08
 Mode  :character   Median :0.002802   Median :0.0026360   Median :0.0027124   Median :0.0027161   Median :4.136e+07   Median :3.310e+08
                    Mean   :0.004863   Mean   :0.0042421   Mean   :0.0045303   Mean   :0.0045636   Mean   :2.224e+08   Mean   :5.567e+08
                    3rd Qu.:0.003660   3rd Qu.:0.0033903   3rd Qu.:0.0035337   3rd Qu.:0.0035328   3rd Qu.:9.483e+07   3rd Qu.:4.283e+08
                    Max.   :0.084945   Max.   :0.0685255   Max.   :0.0787726   Max.   :0.0788251   Max.   :2.540e+10   Max.   :1.011e+10
  eos.symbol           eos.high           eos.low            eos.open           eos.close            eos.vol             eos.mktcap
 Length:1245        Min.   : 0.5187   Min.   : 0.4802   Min.   : 0.4928   Min.   : 0.4932   Min.   :4.730e+06   Min.   :2.090e+08
 Class :character   1st Qu.: 2.7409   1st Qu.: 2.5976   1st Qu.: 2.6641   1st Qu.: 2.6641   1st Qu.:7.465e+08   1st Qu.:2.478e+09
 Mode  :character   Median : 3.6850   Median : 3.4589   Median : 3.5830   Median : 3.5860   Median :1.615e+09   Median :3.268e+09
                    Mean   : 4.9459   Mean   : 4.5166   Mean   : 4.7430   Mean   : 4.7442   Mean   :1.849e+09   Mean   :4.048e+09
                    3rd Qu.: 5.8777   3rd Qu.: 5.4669   3rd Qu.: 5.7407   3rd Qu.: 5.7366   3rd Qu.:2.497e+09   3rd Qu.:4.933e+09
                    Max.   :22.8904   Max.   :18.8774   Max.   :21.6391   Max.   :21.5426   Max.   :1.001e+10   Max.   :1.777e+10
  eth.symbol           eth.high           eth.low            eth.open           eth.close            eth.vol             eth.mktcap
 Length:1245        Min.   :  85.34   Min.   :  82.83   Min.   :  84.28   Min.   :  84.31   Min.   :2.537e+08   Min.   :8.751e+09
 Class :character   1st Qu.: 181.67   1st Qu.: 173.13   1st Qu.: 177.34   1st Qu.: 177.34   1st Qu.:2.226e+09   1st Qu.:1.911e+10
 Mode  :character   Median : 257.36   Median : 241.80   Median : 247.74   Median : 247.52   Median :6.551e+09   Median :2.719e+10
                    Mean   : 397.50   Mean   : 368.22   Mean   : 383.60   Mean   : 384.43   Mean   :8.734e+09   Mean   :4.064e+10
                    3rd Qu.: 475.16   3rd Qu.: 449.52   3rd Qu.: 463.70   3rd Qu.: 463.45   3rd Qu.:1.182e+10   3rd Qu.:4.758e+10
                    Max.   :2036.29   Max.   :1896.68   Max.   :1959.90   Max.   :1960.16   Max.   :6.073e+10   Max.   :2.249e+11

iota.symbol          iota.high          iota.low           iota.open          iota.close          iota.vol            iota.mktcap
Length:1245        Min.   :0.1247   Min.   :0.07962   Min.   :0.1103   Min.   :0.1102   Min.   :2.896e+06   Min.   :3.063e+08
Class :character   1st Qu.:0.2697   1st Qu.:0.25217   1st Qu.:0.2617   1st Qu.:0.2617   1st Qu.:9.923e+06   1st Qu.:7.274e+08
Mode  :character   Median :0.3312   Median :0.30674   Median :0.3188   Median :0.3187   Median :1.763e+07   Median :8.859e+08
                   Mean   :0.7106   Mean   :0.62872   Mean   :0.6719   Mean   :0.6720   Mean   :4.901e+07   Mean   :1.868e+09
                   3rd Qu.:0.6094   3rd Qu.:0.55859   3rd Qu.:0.5797   3rd Qu.:0.5756   3rd Qu.:4.213e+07   3rd Qu.:1.600e+09
                   Max.   :5.6900   Max.   :4.86136   Max.   :5.3711   Max.   :5.3663   Max.   :2.127e+09   Max.   :1.492e+10
 ltc.symbol           ltc.high          ltc.low           ltc.open          ltc.close          ltc.vol             ltc.mktcap
Length:1245        Min.   : 23.75   Min.   : 22.82   Min.   : 23.46   Min.   : 23.46   Min.   :5.179e+07   Min.   :1.398e+09
Class :character   1st Qu.: 49.28   1st Qu.: 46.75   1st Qu.: 48.00   1st Qu.: 48.01   1st Qu.:4.525e+08   1st Qu.:3.021e+09
Mode  :character   Median : 63.07   Median : 59.49   Median : 61.24   Median : 61.30   Median :2.178e+09   Median :3.816e+09
                   Mean   : 87.71   Mean   : 80.67   Mean   : 84.32   Mean   : 84.40   Mean   :2.523e+09   Mean   :5.081e+09
                   3rd Qu.:104.84   3rd Qu.: 97.31   3rd Qu.:101.11   3rd Qu.:101.15   3rd Qu.:3.482e+09   3rd Qu.:6.241e+09
                   Max.   :375.29   Max.   :335.68   Max.   :359.12   Max.   :358.34   Max.   :1.799e+10   Max.   :1.948e+10
 xmr.symbol           xmr.high          xmr.low           xmr.open          xmr.close          xmr.vol             xmr.mktcap
Length:1245        Min.   : 36.65   Min.   : 26.70   Min.   : 33.08   Min.   : 33.01   Min.   :1.016e+07   Min.   :5.770e+08
Class :character   1st Qu.: 64.82   1st Qu.: 61.70   1st Qu.: 63.17   1st Qu.: 63.15   1st Qu.:4.486e+07   1st Qu.:1.096e+09
Mode  :character   Median : 92.35   Median : 86.56   Median : 89.66   Median : 89.55   Median :8.699e+07   Median :1.521e+09
                   Mean   :119.67   Mean   :109.76   Mean   :114.98   Mean   :115.05   Mean   :2.801e+08   Mean   :1.910e+09
                   3rd Qu.:136.53   3rd Qu.:127.53   3rd Qu.:132.02   3rd Qu.:132.63   3rd Qu.:1.471e+08   3rd Qu.:2.249e+09
                   Max.   :495.84   Max.   :453.39   Max.   :470.29   Max.   :469.20   Max.   :2.896e+10   Max.   :7.274e+09
 xlm.symbol           xlm.high           xlm.low            xlm.open           xlm.close            xlm.vol             xlm.mktcap
Length:1245        Min.   :0.01254   Min.   :0.01133   Min.   :0.01188   Min.   :0.01185   Min.   :1.670e+06   Min.   :1.966e+08
Class :character   1st Qu.:0.07197   1st Qu.:0.06778   1st Qu.:0.07006   1st Qu.:0.07019   1st Qu.:8.499e+07   1st Qu.:1.408e+09
Mode  :character   Median :0.10623   Median :0.09968   Median :0.10315   Median :0.10312   Median :1.805e+08   Median :2.041e+09
                   Mean   :0.16348   Mean   :0.14719   Mean   :0.15540   Mean   :0.15571   Mean   :3.418e+08   Mean   :3.019e+09
                   3rd Qu.:0.23327   3rd Qu.:0.21316   3rd Qu.:0.22301   3rd Qu.:0.22347   3rd Qu.:3.814e+08   3rd Qu.:4.197e+09
                   Max.   :0.93814   Max.   :0.67750   Max.   :0.89240   Max.   :0.89623   Max.   :1.041e+10   Max.   :1.602e+10
 trx.symbol           trx.high            trx.low             trx.open            trx.close             trx.vol             trx.mktcap
Length:1245        Min.   :0.002006   Min.   :0.001684   Min.   :0.001795   Min.   :0.001783   Min.   :1.160e+05   Min.   :9.786e+07
Class :character   1st Qu.:0.016934   1st Qu.:0.015949   1st Qu.:0.016384   1st Qu.:0.016411   1st Qu.:1.883e+08   1st Qu.:1.092e+09
Mode  :character   Median :0.024339   Median :0.022961   Median :0.023637   Median :0.023623   Median :5.711e+08   Median :1.573e+09
                   Mean   :0.028759   Mean   :0.025711   Mean   :0.027266   Mean   :0.027288   Mean   :7.437e+08   Mean   :1.834e+09
                   3rd Qu.:0.033013   3rd Qu.:0.029912   3rd Qu.:0.031476   3rd Qu.:0.031539   3rd Qu.:1.107e+09   3rd Qu.:2.182e+09
                   Max.   :0.300363   Max.   :0.176712   Max.   :0.224499   Max.   :0.220555   Max.   :6.184e+09   Max.   :1.450e+10

btc.daily_return   ada.daily_return   doge.daily_return  eos.daily_return    eth.daily_return    iota.daily_return  ltc.daily_return
Min.   :-37.1869   Min.   :-39.5344   Min.   :-39.7983   Min.   :-39.57363   Min.   :-42.30885   Min.   :-41.9808   Min.   :-36.15815
1st Qu.: -1.4318   1st Qu.: -2.9174   1st Qu.: -2.0728   1st Qu.: -2.38190   1st Qu.: -2.00623   1st Qu.: -2.9117   1st Qu.: -2.54885
Median :  0.1722   Median :  0.1393   Median : -0.0491   Median :  0.00708   Median :  0.08765   Median : -0.1128   Median : -0.06894
Mean   :  0.2743   Mean   :  0.6224   Mean   :  0.6912   Mean   :  0.34240   Mean   :  0.24506   Mean   :  0.2574   Mean   :  0.23647
3rd Qu.:  1.9165   3rd Qu.:  3.2734   3rd Qu.:  1.8569   3rd Qu.:  2.61217   3rd Qu.:  2.58864   3rd Qu.:  3.1189   3rd Qu.:  2.52993
Max.   : 25.4702   Max.   :136.5243   Max.   :355.6254   Max.   : 41.95707   Max.   : 25.95129   Max.   : 45.9443   Max.   : 48.80591
xmr.daily_return   xlm.daily_return    trx.daily_return    btc.daily_volume_change ada.daily_volume_change doge.daily_volume_change
Min.   :-38.9748   Min.   :-33.62227   Min.   :-40.75704   Min.   :-86.9188        Min.   :-74.49          Min.   : -84.840
1st Qu.: -2.3752   1st Qu.: -2.78938   1st Qu.: -2.74860   1st Qu.:-10.8031        1st Qu.:-20.58          1st Qu.: -14.272
Median :  0.1047   Median : -0.06619   Median :  0.01799   Median : -0.8359        Median : -2.59          Median :  -1.052
Mean   :  0.2042   Mean   :  0.53404   Mean   :  0.58403   Mean   :  2.7529        Mean   :  9.16          Mean   :  13.975
3rd Qu.:  2.8492   3rd Qu.:  2.61560   3rd Qu.:  2.92014   3rd Qu.: 12.3013        3rd Qu.: 23.92          3rd Qu.:  15.171
Max.   : 27.6401   Max.   : 94.75207   Max.   :119.10428   Max.   :543.9003        Max.   :505.43          Max.   :5257.273
eos.daily_volume_change eth.daily_volume_change iota.daily_volume_change ltc.daily_volume_change xmr.daily_volume_change
Min.   :-65.600         Min.   :-66.1533        Min.   :-82.284          Min.   :-56.695         Min.   :  -98.997
1st Qu.:-15.031         1st Qu.:-11.3838        1st Qu.:-20.654          1st Qu.:-11.579         1st Qu.:  -13.483
Median : -1.786         Median : -0.6393        Median : -2.725          Median : -1.129         Median :   -0.432
Mean   :  5.635         Mean   :  2.8218        Mean   :  8.351          Mean   :  3.596         Mean   :   26.810
3rd Qu.: 15.613         3rd Qu.: 11.2360        3rd Qu.: 22.973          3rd Qu.: 11.177         3rd Qu.:   14.810
Max.   :706.465         Max.   :192.4079        Max.   :520.575          Max.   :494.822         Max.   :13804.304
xlm.daily_volume_change trx.daily_volume_change
Min.   : -72.460        Min.   :-64.2571
1st Qu.: -15.985        1st Qu.:-11.8092
Median :  -1.208        Median : -0.7027
Mean   :   9.526        Mean   :  5.4367
3rd Qu.:  16.202        3rd Qu.: 11.7665
Max.   :3119.044        Max.   :845.4179

btc.big_day ada.big_day doge.big_day eos.big_day eth.big_day iota.big_day ltc.big_day xmr.big_day xlm.big_day trx.big_day
0:1138      0:1031      0:1099       0:1060      0:1088      0:1049       0:1076      0:1070      0:1059      0:1047
1: 107      1: 214      1: 146       1: 185      1: 157      1: 196       1: 169      1: 175      1: 186      1: 198
```

```
> summary(cryptos.molten)
      date                          variable          value
 Min.   :2017-10-02   btc.daily_return :1245   Min.   :-42.3088
 1st Qu.:2018-08-09   ada.daily_return :1245   1st Qu.: -2.4079
 Median :2019-06-16   doge.daily_return:1245   Median :  0.0295
 Mean   :2019-06-16   eos.daily_return :1245   Mean   :  0.3992
 3rd Qu.:2020-04-22   eth.daily_return :1245   3rd Qu.:  2.6335
 Max.   :2021-02-27   iota.daily_return:1245   Max.   :355.6254
                      (Other)          :4980
> str(cryptos.molten)
'data.frame':   12450 obs. of  3 variables:
 $ date    : Date, format: "2017-10-02" "2017-10-03" "2017-10-04" "2017-10-05" ...
 $ variable: Factor w/ 10 levels "btc.daily_return",..: 1 1 1 1 1 1 1 1 1 1 ...
 $ value   : num  0.307 -2.064 -2.084 2.329 1.072 ...
> unique(cryptos.molten$variable)
 [1] btc.daily_return  ada.daily_return  doge.daily_return eos.daily_return  eth.daily_return
 iota.daily_return ltc.daily_return  xmr.daily_return  xlm.daily_return  trx.daily_return
Levels: btc.daily_return ada.daily_return doge.daily_return eos.daily_return eth.daily_return
 iota.daily_return ltc.daily_return xmr.daily_return xlm.daily_return trx.daily_return
```

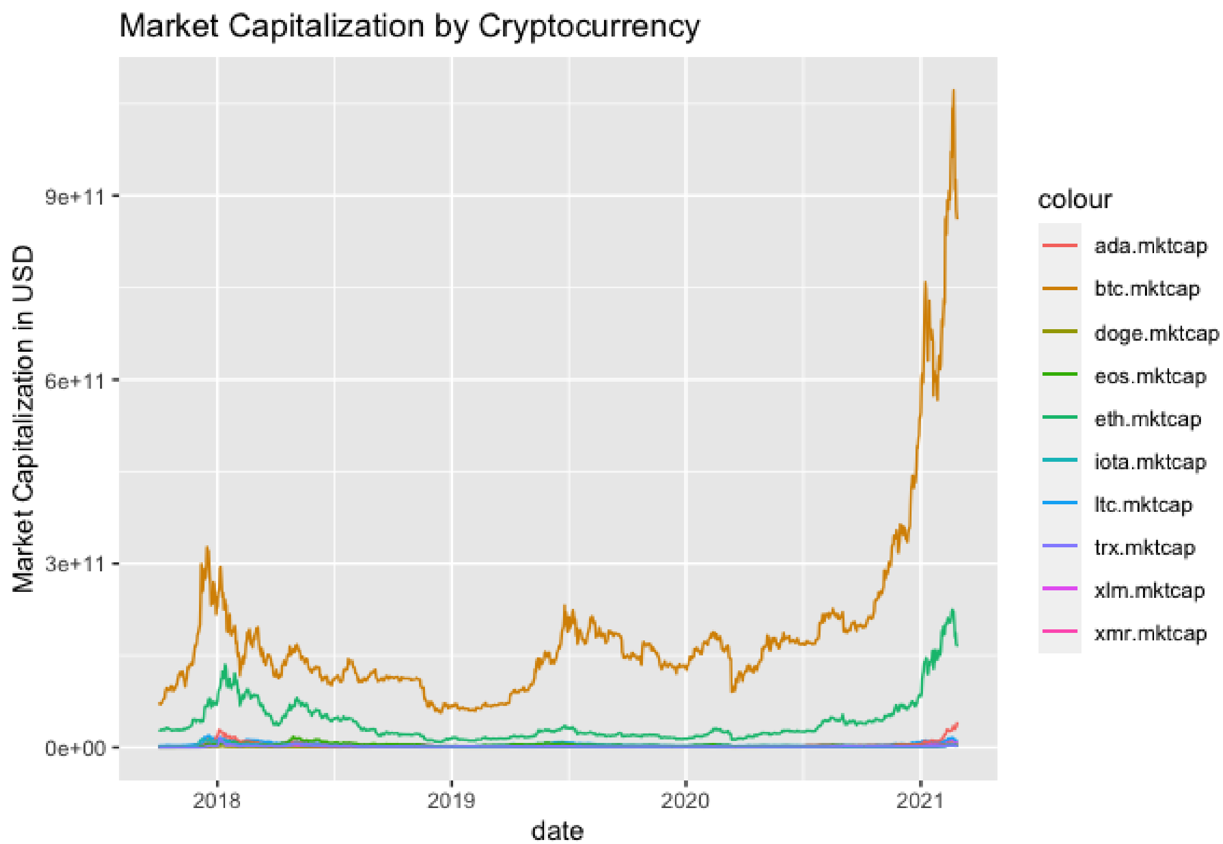
```
> str(cryptos)
'data.frame':   1245 obs. of  101 variables:
 $ date                   : Date, format: "2017-10-02" "2017-10-03" "2017-10-04" ...
 $ btc.symbol             : chr  "BTC" "BTC" "BTC" "BTC" ...
 $ btc.high               : num  4470 4432 4352 4363 4413 ...
 $ btc.low                : num  4377 4259 4210 4164 4321 ...
 $ btc.open               : num  4396 4408 4319 4230 4324 ...
 $ btc.close              : num  4409 4317 4229 4328 4371 ...
 $ btc.vol                : num  1.43e+09 1.29e+09 1.12e+09 1.16e+09 1.07e+09 ...
 $ btc.mktcap             : num  7.32e+10 7.17e+10 7.02e+10 7.19e+10 7.26e+10 ...
 $ ada.symbol             : chr  "ADA" "ADA" "ADA" "ADA" ...
 $ ada.high               : num  0.0301 0.0274 0.0228 0.0222 0.0215 ...
 $ ada.low                : num  0.02 0.0207 0.0209 0.0209 0.0184 ...
 $ ada.open               : num  0.0246 0.0258 0.0209 0.022 0.0214 ...
 $ ada.close              : num  0.0259 0.0208 0.0219 0.0215 0.0185 ...
 $ ada.vol                : num  57641300 16997800 9000050 5562510 7780710 ...
 $ ada.mktcap             : num  6.29e+08 5.40e+08 5.69e+08 5.57e+08 4.81e+08 ...
 $ doge.symbol            : chr  "DOGE" "DOGE" "DOGE" "DOGE" ...
 $ doge.high              : num  0.00113 0.00112 0.00112 0.00108 0.00108 ...
 $ doge.low               : num  0.001055 0.001004 0.000938 0.000979 0.000995 ...
 $ doge.open              : num  0.00108 0.00108 0.00112 0.00102 0.001 ...
 $ doge.close             : num  0.00109 0.00111 0.00102 0.001 0.00106 ...
 $ doge.vol               : num  3145110 2044110 2119210 2656730 1505160 ...
 $ doge.mktcap            : num  1.21e+08 1.23e+08 1.14e+08 1.12e+08 1.18e+08 ...
 $ eos.symbol             : chr  "EOS" "EOS" "EOS" "EOS" ...
 $ eos.high               : num  0.714 0.669 0.634 0.621 0.609 ...
 $ eos.low                : num  0.636 0.616 0.585 0.576 0.591 ...
 $ eos.open               : num  0.708 0.653 0.632 0.599 0.607 ...
 $ eos.close              : num  0.655 0.631 0.602 0.606 0.6 ...
 $ eos.vol                : num  8299880 7114830 5449300 8038660 6785670 ...
 $ eos.mktcap             : num  2.50e+08 2.42e+08 2.32e+08 2.35e+08 2.33e+08 ...
 $ eth.symbol             : chr  "ETH" "ETH" "ETH" "ETH" ...
 $ eth.high               : num  303 300 299 298 309 ...
 $ eth.low                : num  295 288 291 289 295 ...
 $ eth.open               : num  302 297 293 293 295 ...
 $ eth.close              : num  297 292 293 296 309 ...
 $ eth.vol                : num  3.39e+08 3.22e+08 2.58e+08 2.54e+08 3.19e+08 ...
 $ eth.mktcap             : num  2.82e+10 2.78e+10 2.78e+10 2.81e+10 2.93e+10 ...
 $ iota.symbol            : chr  "MIOTA" "MIOTA" "MIOTA" "MIOTA" ...
 $ iota.high              : num  0.622 0.582 0.571 0.568 0.563 ...
 $ iota.low               : num  0.565 0.516 0.52 0.521 0.493 ...
 $ iota.open              : num  0.618 0.577 0.561 0.547 0.543 ...
 $ iota.close             : num  0.575 0.562 0.547 0.541 0.503 ...
 $ iota.vol               : num  13928300 13709400 9152780 7592760 12478500 ...
 $ iota.mktcap            : num  1.60e+09 1.56e+09 1.52e+09 1.50e+09 1.40e+09 ...
 $ ltc.symbol             : chr  "LTC" "LTC" "LTC" "LTC" ...
 $ ltc.high               : num  54.8 53.6 53 51.9 52.6 ...
 $ ltc.low                : num  53.3 51.6 50.8 50.4 51.5 ...
```

```
 $ ltc.low                : num  53.3 51.6 50.8 50.4 51.5 ...
 $ ltc.open               : num  54.5 53.4 52.3 51.4 51.6 ...
 $ ltc.close              : num  53.4 52.3 51.4 51.7 52.1 ...
 $ ltc.vol                : num  95474600 88516096 77473400 69594400 59355900 ...
 $ ltc.mktcap             : num  2.84e+09 2.79e+09 2.74e+09 2.75e+09 2.78e+09 ...
 $ xmr.symbol             : chr  "XMR" "XMR" "XMR" "XMR" ...
 $ xmr.high               : num  93.8 94.8 93.7 92.4 92.9 ...
 $ xmr.low                : num  91 87.2 89 88.1 90.2 ...
 $ xmr.open               : num  93.6 91.5 92.9 90.5 91.9 ...
 $ xmr.close              : num  91.5 93 90.4 91.9 91.2 ...
 $ xmr.vol                : num  27534700 44289800 29568100 27697400 24504100 ...
 $ xmr.mktcap             : num  1.39e+09 1.41e+09 1.37e+09 1.39e+09 1.38e+09 ...
 $ xlm.symbol             : chr  "XLM" "XLM" "XLM" "XLM" ...
 $ xlm.high               : num  0.0137 0.0129 0.0125 0.0153 0.0192 ...
 $ xlm.low                : num  0.0126 0.0113 0.0116 0.0117 0.0147 ...
 $ xlm.open               : num  0.0137 0.0129 0.0125 0.0119 0.0147 ...
 $ xlm.close              : num  0.013 0.0124 0.0119 0.0146 0.0184 ...
 $ xlm.vol                : num  2362700 2577540 1670080 9616220 24002100 ...
 $ xlm.mktcap             : num  2.15e+08 2.06e+08 1.97e+08 2.42e+08 3.05e+08 ...
 $ trx.symbol             : chr  "TRX" "TRX" "TRX" "TRX" ...
 $ trx.high               : num  0.00318 0.00319 0.00352 0.00349 0.00335 ...
 $ trx.low                : num  0.00177 0.00213 0.0024 0.00294 0.00281 ...
 $ trx.open               : num  0.00185 0.00275 0.00247 0.00327 0.00332 ...
 $ trx.close              : num  0.0028 0.00245 0.00325 0.00332 0.003 ...
 $ trx.vol                : num  887622 822107 1072200 704167 478330 ...
 $ trx.mktcap             : num  1.12e+08 9.79e+07 1.30e+08 1.33e+08 1.20e+08 ...
 $ btc.daily_return       : num  0.307 -2.064 -2.084 2.329 1.072 ...
 $ ada.daily_return       : num  5.38 -19.18 5.12 -2.11 -13.2 ...
 $ doge.daily_return      : num  0.923 2.198 -8.613 -1.442 5.297 ...
 $ eos.daily_return       : num  -7.49 -3.31 -4.82 1.14 -1.13 ...
 $ eth.daily_return       : num  -1.6553 -1.6875 -0.0328 1.0554 4.5512 ...
 $ iota.daily_return      : num  -6.92 -2.68 -2.47 -1.18 -7.32 ...
 $ ltc.daily_return       : num  -2.147 -1.949 -1.692 0.522 0.938 ...
 $ xmr.daily_return       : num  -2.153 1.573 -2.734 1.508 -0.783 ...
 $ xlm.daily_return       : num  -5.37 -4.2 -4.95 22.69 25.55 ...
 $ trx.daily_return       : num  51.71 -10.99 31.64 1.56 -9.79 ...
 $ btc.daily_volume_change : num  0 -10.04 -13.3 4.03 -7.9 ...
 $ ada.daily_volume_change : num  0 -70.5 -47.1 -38.2 39.9 ...
 $ doge.daily_volume_change: num  0 -35.01 3.67 25.36 -43.35 ...
 $ eos.daily_volume_change : num  0 -14.3 -23.4 47.5 -15.6 ...
 $ eth.daily_volume_change : num  0 -5.23 -19.83 -1.61 25.58 ...
 $ iota.daily_volume_change: num  0 -1.57 -33.24 -17.04 64.35 ...
 $ ltc.daily_volume_change : num  0 -7.29 -12.48 -10.17 -14.71 ...
 $ xmr.daily_volume_change : num  0 60.85 -33.24 -6.33 -11.53 ...
 $ xlm.daily_volume_change : num  0 9.09 -35.21 475.79 149.6 ...
 $ trx.daily_volume_change : num  0 -7.38 30.42 -34.33 -32.07 ...
 $ btc.big_day            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ ada.big_day            : Factor w/ 2 levels "0","1": 2 1 2 1 1 2 1 2 1 1 ...
 $ doge.big_day           : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 1 1 2 1 ...
 $ eos.big_day            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
 $ eth.big_day            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ iota.big_day           : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 1 1 1 1 ...
 $ ltc.big_day            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ xmr.big_day            : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
  [list output truncated]
```

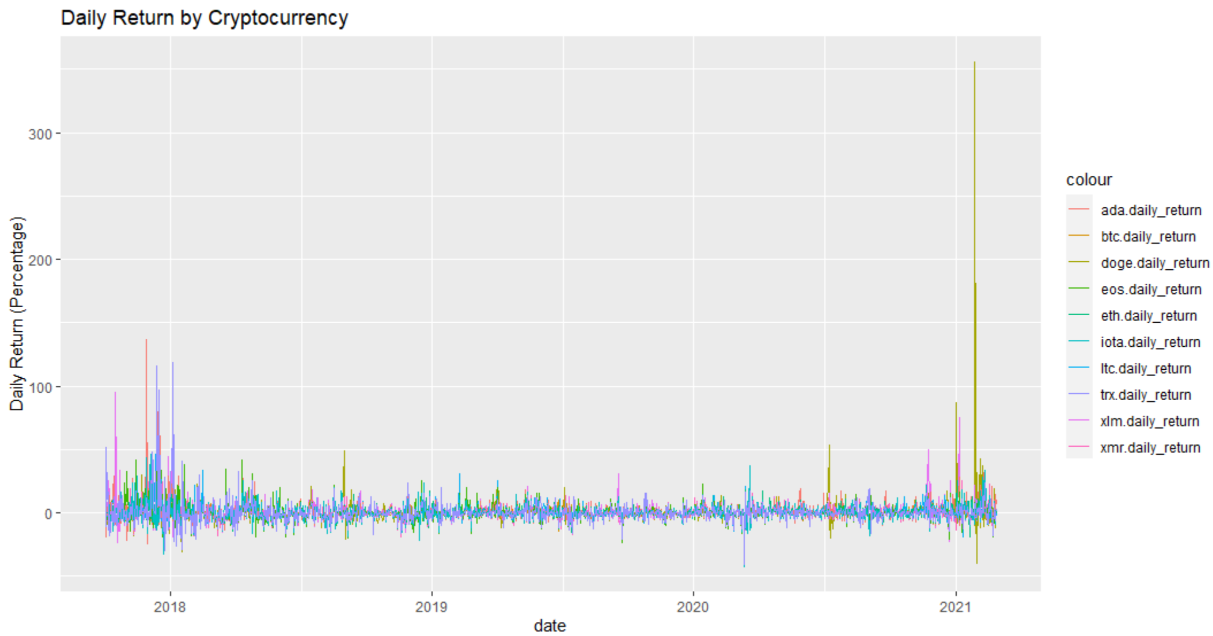
We now have a large data set that is now well-suited to our needs. We have successfully removed the undesired identity variable, merged our separate dataframes into one master dataframe, *cryptos*, created our desired calculated variables, and melted *cryptos* into *crypto.molten*.

Initial Interesting Findings after Munging

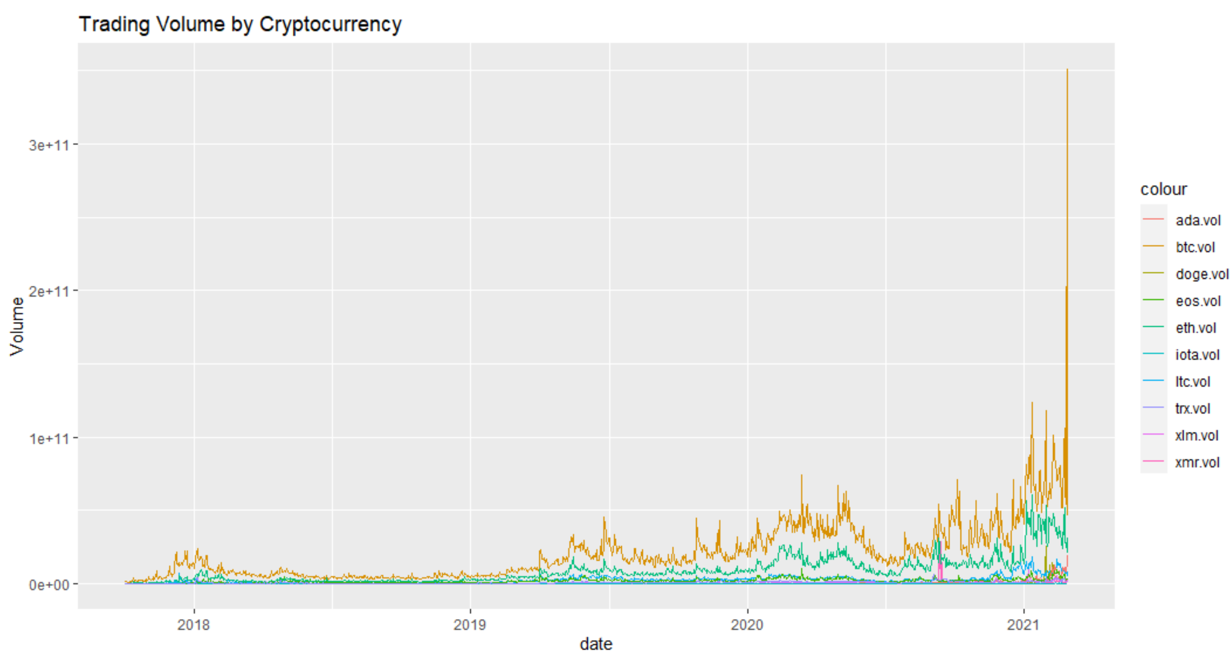
Cryptocurrency exchanges were modeled after stock exchanges, which have opening and closing times, and therefore daily highs and daily lows. The exchanges for cryptocurrencies never close, however, so the daily high/low and open/close are based on the last 24-hour frame of reference (if looking at live data) or some arbitrary 24-hour block of time (in this case, each 24-hour trading day starts at 00:00:00 UTC).



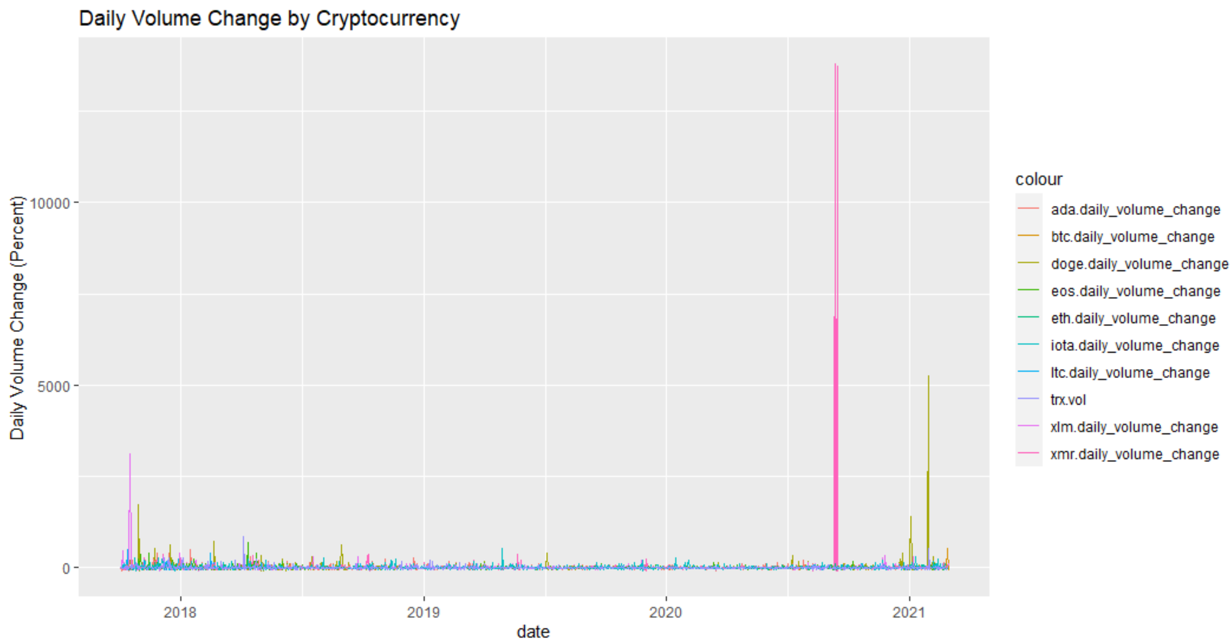
An initial comparison of market capitalizations by cryptocurrency showing Bitcoin's role as the flagship in the cryptocurrency space. Coincidentally, Bitcoin is also by far the most valuable of these assets as of 2021. For these reasons, we will expect to see a lower volatility in Bitcoin's relative price movement compared to altcoins.



Additionally, while Bitcoin has the highest trading volume, other cryptocurrencies also have significantly high volumes that have equaled or surpassed Bitcoin at times, most notably Ethereum, even though Ethereum has historically traded for a fraction of Bitcoin's value.



Finally, while the absolute difference in trading volume appears quite high for some coins, the percent change in volume seems to have less variability. There are some outlier days where volume for certain coins spiked extremely high.


Daily Volume Change by Cryptocurrency
Daily Volume Change (Percent)
10000
5000
0
2018
2019
2020
2021
date
colour
ada.daily_volume_change
btc.daily_volume_change
doge.daily_volume_change
eos.daily_volume_change
eth.daily_volume_change
iota.daily_volume_change
ltc.daily_volume_change
trx.vol
xlm.daily_volume_change
xmr.daily_volume_change

III. DATA SET DESCRIPTIVE STATISTICS AND STRUCTURE

R Studio Script is located in Appendix

Methods

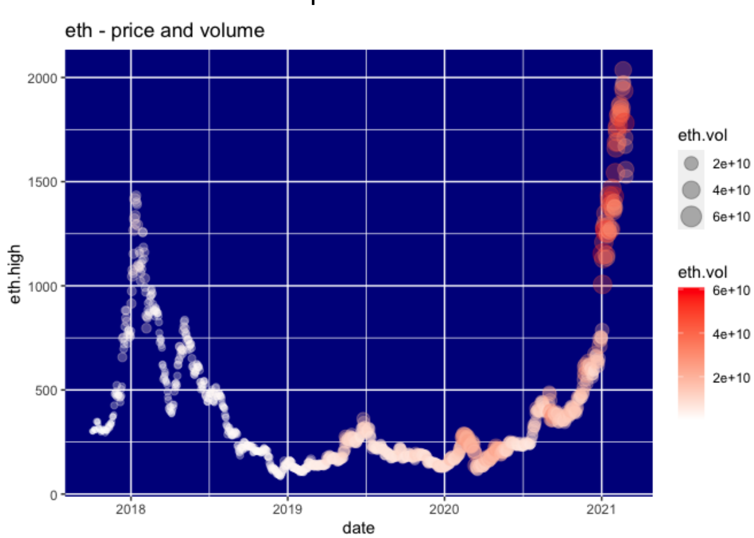
We explored the relationships in our data in more detail after performing initial analysis by:

- Plotting price vs date for each coin, double-encoding volume into size and color to make identifying relationships easier;
- Plotting bitcoin price vs altcoin price, and plotting linear models onto those scatterplots, to examine correlations between Bitcoin price and altcoin price;
- Creating a volatility heatmap to analyze the coins with the highest and lowest returns;
- Creating barcharts and histograms to analyze the most volatile coins.

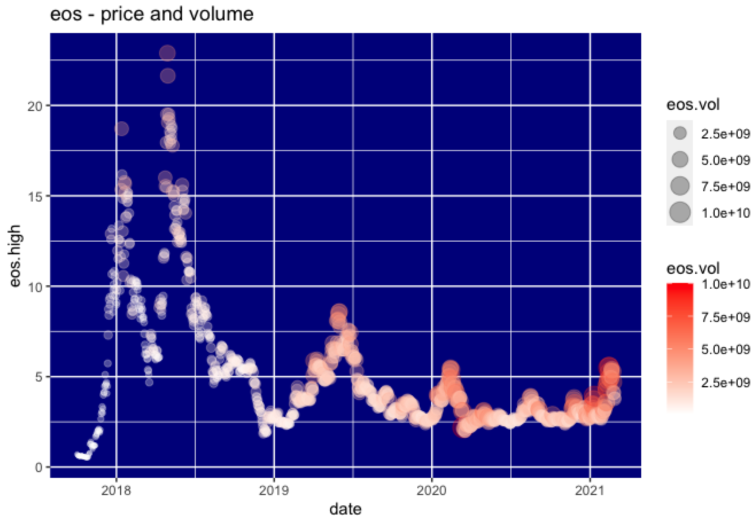
More Interesting Findings

We can see that as the price spikes for Bitcoin, volume also spikes:

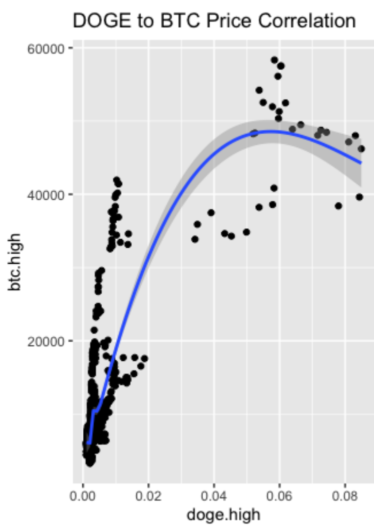
This trend is even more pronounced for Ethereum:

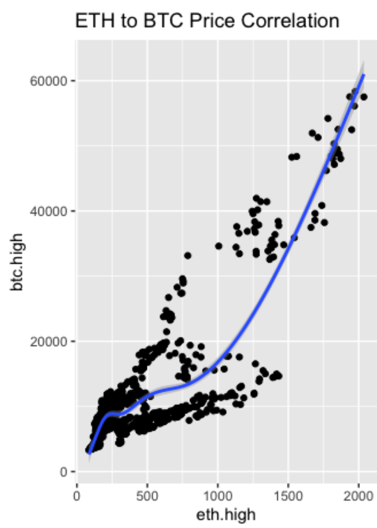


We see a similar correlation in EOS, and a recent increase in volume could be a bullish signal:

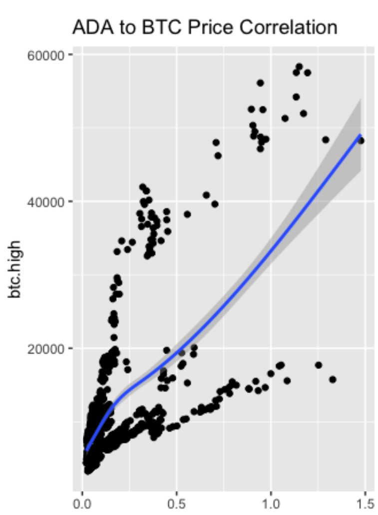


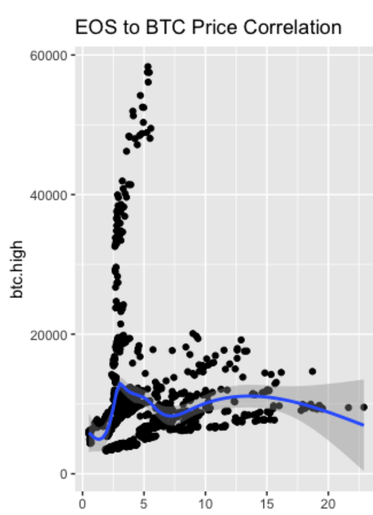
When we examined the relationship between Bitcoin and altcoin prices, we found strong positive correlations in most coins, with one notable exception.



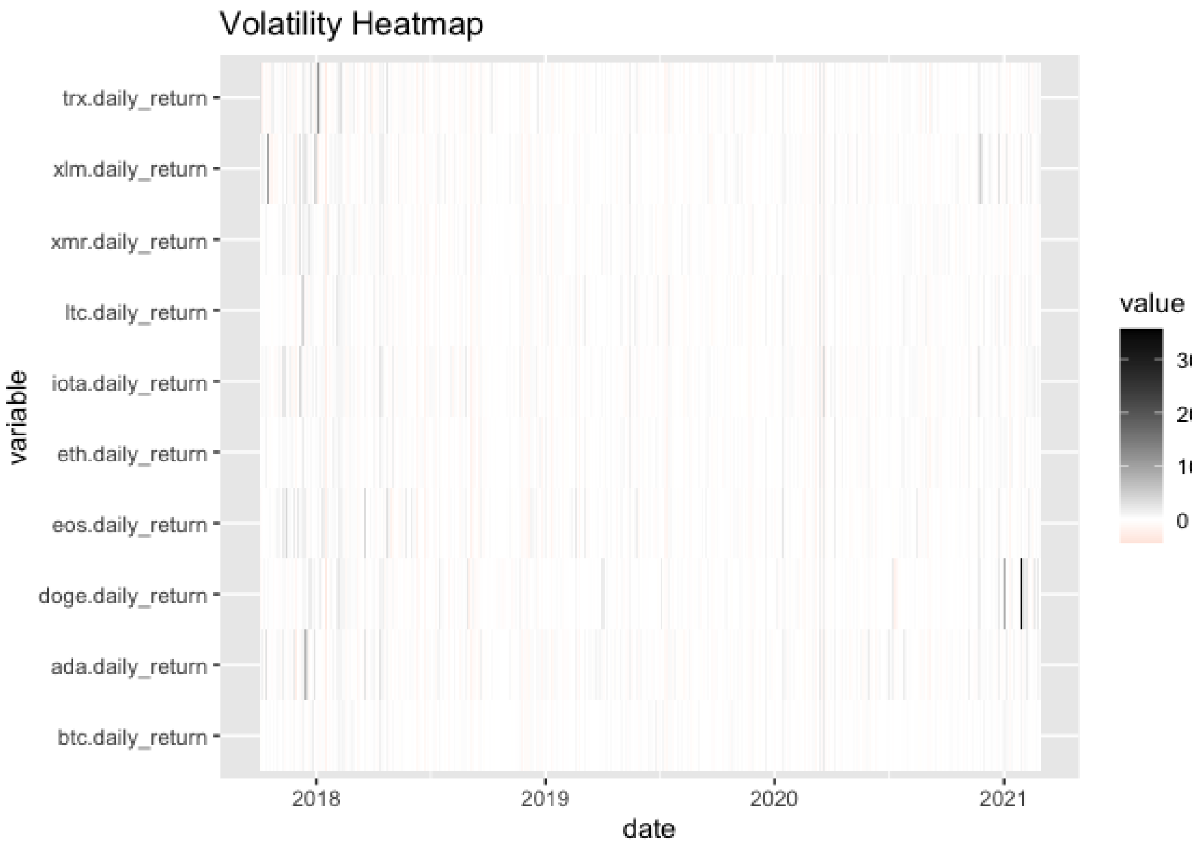


EOS to BTC plot shows that EOS price has not moved up in tandem with BTC's recent spike to $60,000, being more independent in its price movement.





This heatmap helped us visualize days of extreme volatility across all coins, which is where our melted, three-column dataframe came in handy. By mapping days of extreme positive returns to black, zero to white, and negative returns to red, we could quickly scan a broad range of dates for highly volatile trading days.



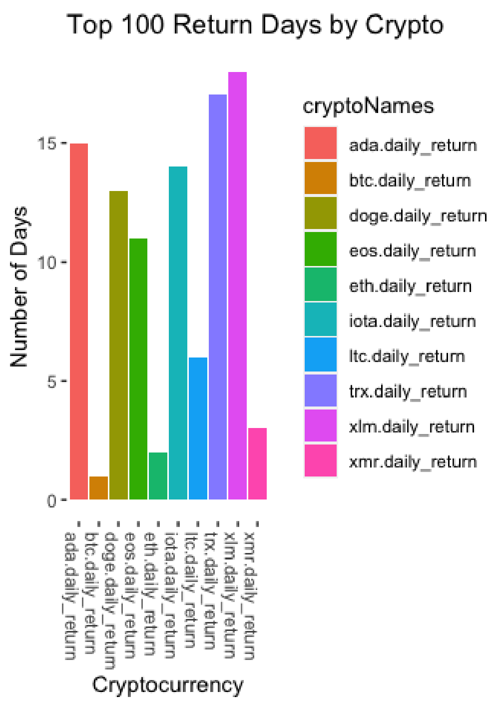
At first glance, we realized that there weren't very many days with values at the extreme minimum and maximum of daily return, so we drilled into our data to understand more.

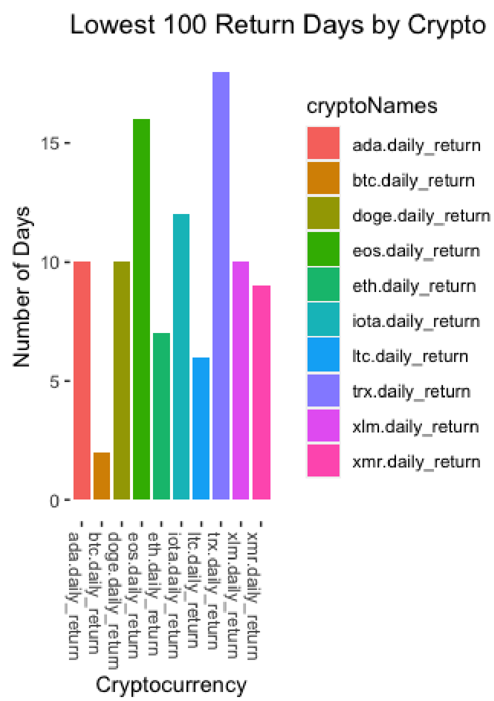
```
> min(cryptos.molten$value)
[1] -42.30885
> max(cryptos.molten$value)
[1] 355.6254
> quantile(cryptos.molten$value, c(0.01, 0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9, 0.99))
         1%         10%         20%         30%         40%         50%         60%         70%         80%         90%         99% 
-15.0854817  -5.9148921  -3.2076757  -1.7793817  -0.8180329   0.0295488   0.8709536   1.9275815   3.5074745   6.4580796  20.8248697 
```

As it turns out, for these 10 coins over this date range, 98% of the daily returns were between -15.09% and 20.82%, making the minimum of -42.31% and the maximum of 355.63% extreme outliers. Given that most days were not very volatile, we wanted to examine the days that were volatile. We ordered this dataframe by return value, then stored the first 100 observations and the last 100 observations. Then we plotted the results in two barcharts. Our results are on the next page.

Stellar had 18 of the top 100 highest single day returns, followed by Tron (17), Cardano (15), Iota (14), Doge (13) and EOS (11). Bitcoin had only 1 such day.
As far as the top 100 most negative days, Tron had the most (18), then EOS (16), Iota (12), Doge (10), Cardano (10), and Stellar (10). Bitcoin had only 2 such days.

This tells us that Bitcoin is relatively stable in price compared to these coins.

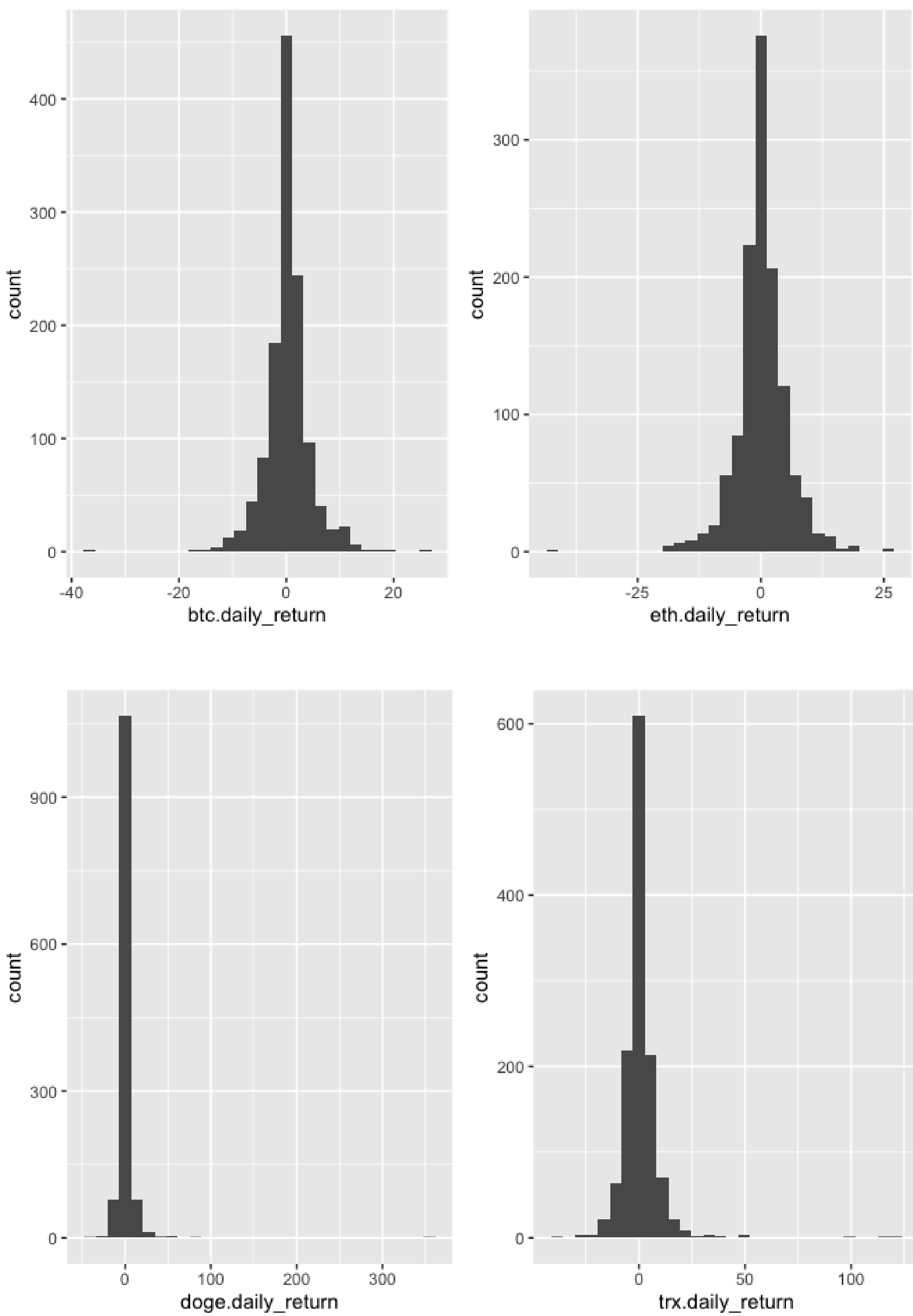





```
> summary(cryptos.100_most_positive_returns$variable)
 btc.daily_return  ada.daily_return doge.daily_return  eos.daily_return  eth.daily_return iota.daily_return  ltc.daily_return
                1                15                13                11                 2                14                 6
 xmr.daily_return  xlm.daily_return  trx.daily_return
                3                18                17
> summary(cryptos.100_most_negative_returns$variable)
 btc.daily_return  ada.daily_return doge.daily_return  eos.daily_return  eth.daily_return iota.daily_return  ltc.daily_return
                2                10                10                16                 7                12                 6
 xmr.daily_return  xlm.daily_return  trx.daily_return
                9                10                18
```

These data tell us that Bitcoin, which had only 1 “extreme” positive day and 2 “extreme” negative days, is relatively stable compared to these other coins, some of which we might describe as particularly volatile—Stellar, Tron, Cardano, Doge, and EOS. This is not surprising given Bitcoin’s high marketcap and price.

If we look at the distributions of daily returns for some of these volatile coins and compare them to more stable coins (next page), we can see that while all distributions appear normally distributed, the volatile coins are positively skewed (since the maximum positive returns are significantly higher than the maximum negative returns).



We decided to focus our models on exploring the particularly volatile coins, along with Bitcoin, and attempt to use correlations we had assessed to predict coin performance.

IV. MODELS

R Studio Script is located in Appendix

Linear Regression

We used linear regression in our initial modeling process. We wanted to determine if the price movement of one coin can explain the price movement of another. The models on the left displays the prediction of the highest price of an altcoin on a given day from the highest price of bitcoin on that day.

In the first model, Bitcoin’s high accounts for approximately 43% of the variability in the Cardano’s high, and the extremely low p-value demonstrates that the predictive variable is statistically significant. This affirms what we saw earlier on our scatterplot of BTC vs ADA.

```
Call:
lm(formula = ada.high ~ btc.high)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.31942 -0.07240 -0.02863  0.01045  1.09870 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.840e-02  6.716e-03   -2.74  0.00624 ** 
btc.high     1.569e-05  5.092e-07   30.81  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1438 on 1243 degrees of freedom
Multiple R-squared:  0.433,	Adjusted R-squared:  0.4325 
F-statistic: 949.1 on 1 and 1243 DF,  p-value: < 2.2e-16
```

Contrast the ADA model with the EOS model: BTC.high explains a negligible proportion of EOS.high, and with a p-value of 0.6, is nowhere near statistically significant. This also affirms what we saw in our scatterplot of BTC vs EOS.

```
Call:
lm(formula = eos.high ~ btc.high)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.4556 -2.1994 -1.2765  0.9308 17.9385 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.012e+00  1.594e-01  31.438   <2e-16 ***
btc.high    -6.313e-06  1.209e-05  -0.522    0.602    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.414 on 1243 degrees of freedom
Multiple R-squared:  0.0002194,	Adjusted R-squared:  -0.0005849 
F-statistic: 0.2728 on 1 and 1243 DF,  p-value: 0.6015
```

But of our highly volatile coins, EOS is unique in that BTC is not a statistically significant predictor of price as seen in the rest of our linear models (next page):

Although BTC only explains 15.57% of the variability in Stellar, it's still a statistically significant predictor of price.

```
Call:
lm(formula = xlm.high ~ btc.high)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.18598 -0.08800 -0.04413  0.07137  0.73949 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 9.331e-02  5.816e-03   16.04   <2e-16 ***
btc.high    6.693e-06  4.409e-07   15.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1245 on 1243 degrees of freedom
Multiple R-squared:  0.1564,	Adjusted R-squared:  0.1557 
F-statistic: 230.4 on 1 and 1243 DF,  p-value: < 2.2e-16
```

Just as it is for Tron, albeit explaining only 5.48% of this coin's variability.

```
Call:
lm(formula = trx.high ~ btc.high)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.029160 -0.011147 -0.004026  0.003389  0.266895 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 2.192e-02  1.006e-03  21.787   <2e-16 ***
btc.high    6.520e-07  7.629e-08   8.547   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.02155 on 1243 degrees of freedom
Multiple R-squared:  0.05551,	Adjusted R-squared:  0.05475 
F-statistic: 73.05 on 1 and 1243 DF,  p-value: < 2.2e-16
```

However, BTC's high explains a majority—59.7% of the variability in Dogecoin.

```
Call:
lm(formula = doge.high ~ btc.high)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.022916 -0.001676  0.000170  0.001933  0.053250 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.579e-03  2.767e-04  -16.55   <2e-16 ***
btc.high     9.007e-07  2.098e-08   42.94   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.005925 on 1243 degrees of freedom
Multiple R-squared:  0.5973,	Adjusted R-squared:  0.597 
F-statistic:  1844 on 1 and 1243 DF,  p-value: < 2.2e-16
```

All-in-all, it doesn't seem like linear models offer the best means to predict price, which makes sense: price doesn't follow a linear pattern. Even the altcoins with prices most strongly correlated to bitcoin's price don't have much of their variability explained by bitcoin's price. Only one coin—Dogecoin—has more than 50% of its variability explained by Bitcoin's price using such a linear model.

So we turned to machine learning in order to construct (hopefully) better models.

Support-Vector Machines

We created support-vector machines to predict daily returns greater than 5% for altcoins (which we defined as “big days”). We chose to iteratively create three models each for Cardano, EOS, and Ethereum. In the first model for each coin, daily return acts as the predictor, whereas daily volume change acts as the predictor in the second model. In the third model, both variables are used to predict volatile days. Once we trained and tested the models, we computed the accuracy to determine which models were the best predictors of volatile days for each coin.

To create our training and testing data, we randomly selected 90% of the indexes from composite cryptos data frame for training and used the remaining 10% for testing. We chose a relatively 9:1 ratio in order to have a higher performance model.

Cardano

The first set of SVMs we created was to predict big days for Cardano.

- SVM 1: daily return, 82.4% prediction accuracy
- SVM 2: daily volume, 81.6% prediction accuracy
- SVM 3: daily return & daily volume 82.4%, prediction accuracy

```
Call:
svm(formula = ada.big_day ~ btc.daily_return, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  463

> ada.svm2 <- svm(ada.big_day~btc.daily_volume_change, data=trainData)
> ada.svm2

Call:
svm(formula = ada.big_day ~ btc.daily_volume_change, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  472

> ada.svm3 <- svm(ada.big_day~btc.daily_volume_change + btc.daily_return, data=trainData)
> ada.svm3

Call:
svm(formula = ada.big_day ~ btc.daily_volume_change + btc.daily_return, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  387
```

```
> ada.svm1.predict.accuracy <- (table(ada.predict1 ,testData$ada.big_day)[1,1]
+                               + table(ada.predict1 ,testData$ada.big_day)[2,2]) / length(testData$ada.big_day)
> ada.svm1.predict.accuracy
[1] 0.824
>
> # model 2 accuracy
> ada.svm2.predict.accuracy <- (table(ada.predict2 ,testData$ada.big_day)[1,1]
+                               + table(ada.predict2 ,testData$ada.big_day)[2,2]) / length(testData$ada.big_day)
> ada.svm2.predict.accuracy
[1] 0.816
>
> # model 3 accuracy
> ada.svm3.predict.accuracy <- (table(ada.predict3 ,testData$ada.big_day)[1,1]
+                               + table(ada.predict3 ,testData$ada.big_day)[2,2]) / length(testData$ada.big_day)
> ada.svm3.predict.accuracy
[1] 0.824
```

EOS

Again we attempted to predict days for this altcoin with returns greater than 5% using Bitcoin data.

- SVM 1: daily return, 88% prediction accuracy
- SVM 2: daily volume, 87.2% prediction accuracy
- SVM 3: daily return & daily volume, 87.2% prediction accuracy

```
svm(formula = eos.big_day ~ btc.daily_return, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  378

> eos.svm2 <- svm(eos.big_day~btc.daily_volume_change, data=trainData)
> eos.svm2

Call:
svm(formula = eos.big_day ~ btc.daily_volume_change, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  408

> eos.svm3 <- svm(eos.big_day~btc.daily_volume_change + btc.daily_return, data=trainData)
> eos.svm3

Call:
svm(formula = eos.big_day ~ btc.daily_volume_change + btc.daily_return, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  341
```

```
> eos.svm1.predict.accuracy <- (table(eos.predict1 ,testData$eos.big_day)[1,1]
+                                + table(eos.predict1 ,testData$eos.big_day)[2,2]) / length(testData$eos.big_day)
> eos.svm1.predict.accuracy
[1] 0.88
>
> # model 2 accuracy
> eos.svm2.predict.accuracy <- (table(eos.predict2 ,testData$eos.big_day)[1,1]
+                                + table(eos.predict2 ,testData$eos.big_day)[2,2]) / length(testData$eos.big_day)
> eos.svm2.predict.accuracy
[1] 0.872
>
> # model 3 accuracy
> eos.svm3.predict.accuracy <- (table(eos.predict3 ,testData$eos.big_day)[1,1]
+                                + table(eos.predict3 ,testData$eos.big_day)[2,2]) / length(testData$eos.big_day)
> eos.svm3.predict.accuracy
[1] 0.872
```

Ethereum

Finally, our last set of SVMs predicted big days for Ethereum in the same way as our other SVMs.

- SVM 1: daily return, 93.6% prediction accuracy
- SVM 2: daily volume, 92.8% prediction accuracy
- SVM 3: daily return & daily volume, 93.6% prediction accuracy

```
svm(formula = eth.big_day ~ btc.daily_return, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  348

> eth.svm2 <- svm(eth.big_day~btc.daily_volume_change, data=trainData)
> eth.svm2

Call:
svm(formula = eth.big_day ~ btc.daily_volume_change, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  415

> eth.svm3 <- svm(eth.big_day~btc.daily_volume_change + btc.daily_return, data=trainData)
> eth.svm3

Call:
svm(formula = eth.big_day ~ btc.daily_volume_change + btc.daily_return, data = trainData)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1

Number of Support Vectors:  296
```

```
> eth.svm1.predict.accuracy <- (table(eth.predict1 ,testData$eth.big_day)[1,1]
+                                + table(eth.predict1 ,testData$eth.big_day)[2,2]) / length(testData$eth.big_day)
> eth.svm1.predict.accuracy
[1] 0.936
>
> # model 2 accuracy
> eth.svm2.predict.accuracy <- (table(eth.predict2 ,testData$eth.big_day)[1,1]
+                                + table(eth.predict2 ,testData$eth.big_day)[2,2]) / length(testData$eth.big_day)
> eth.svm2.predict.accuracy
[1] 0.928
>
> # model 3 accuracy
> eth.svm3.predict.accuracy <- (table(eth.predict3 ,testData$eth.big_day)[1,1]
+                                + table(eth.predict3 ,testData$eth.big_day)[2,2]) / length(testData$eth.big_day)
> eth.svm3.predict.accuracy
[1] 0.936
```

By using only the daily return of Bitcoin to predict days of large returns for altcoins, we could generate highly accurate predictions in our models. Adding the daily volume change did not improve our models, and at times it actually weakened them.

V. CONCLUSIONS

Bullish spikes in price movement in our price and volume charts showed a strong correlation between the two variables. Histograms showed normal distributions for daily percent returns on all coins. Outliers in the daily percent returns were skewed to the right, corroborating the booming state of cryptocurrencies.

The vast majority of the positive return outliers took place in altcoins, with Bitcoin having only one of the top 100 daily returns between 2017-10-02 and early 2021. This finding can, at least, partly reflect the public enthusiasm for altcoins.

Our plots and linear models showed strong correlations between bitcoin and altcoins, with EOS being a notable exception.

We were not surprised that our linear models were not very good predictors of price, but we were surprised that our SVMs were highly accurate in predicting days with high returns—more than 90% accurate in our best model.

While we were able to create simple models to achieve our learning goals, we realize that a model that could solve advanced business problems remains to be completed. In future steps, we could expand the model to take live API data and create more powerful models using more complicated transformations in our computations. Nonetheless, we have focused our efforts in creating visualizations that would be of interest to traders, investors and market analysts.

VI. APPENDIX

We have appended our R Studio Script in full to the end of this document, starting on the next page. If you would like our .R file, please contact us.

```
##############################################
# Course: IST 687
# Assignment: Final Project Deliverable
# Team Members: Deery-Schmitt, Sam
#               Huang, Cliff
#               Onyeugbo, Glory
##############################################
# libraries
library(ggplot2)
library(dplyr)
library(viridis)
library(Interpol.T)
library(lubridate)
library(ggExtra)
library(tidyr)
library(reshape2)
library(grid)
library(gridExtra)
library(lattice)
library(e1071)
##############################################
# data
# subset of the csv files available from
# 'https://www.kaggle.com/sudalairajkumar/cryptocurrencypricehistory'

#-------------------------------------------------------------------------------
# II. DATA ACQUISITION AND MUNGING
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
# Data Acquisition
#-------------------------------------------------------------------------------

bitcoin <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Bitcoin.csv")
cardano <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Cardano.csv")
dogecoin <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Dogecoin.csv")
eos <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\\IST_687_Applied_Data_Science\
\coin_EOS.csv")
ethereum <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Ethereum.csv")
iota <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\\IST_687_Applied_Data_Science\
\coin_Iota.csv")
litecoin <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Litecoin.csv")
monero <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Monero.csv")
stellar <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\
\IST_687_Applied_Data_Science\\coin_Stellar.csv")
tron <- read.csv("C:\\Users\\SamDe\\Desktop\\MS\\IST_687_Applied_Data_Science\
\coin_Tron.csv")
```

```
#-------------------------------------------------------------------------------
# Structure and Summary of Raw Data
#-------------------------------------------------------------------------------

str(bitcoin)
summary(bitcoin)
str(cardano)
summary(cardano)
str(dogecoin)
summary(dogecoin)
str(ethereum)
summary(ethereum)
str(eos)
summary(eos)
str(iota)
summary(iota)
str(litecoin)
summary(litecoin)
str(monero)
summary(monero)
str(stellar)
summary(stellar)
str(tron)
summary(tron)

#-------------------------------------------------------------------------------
# Munging
#-------------------------------------------------------------------------------
# Creating a composite data frame

# identify the shortest data frame
head(cardano)
# cardano starts at 2017-10-02
# get all data frames to same time frame
bitcoin <- bitcoin[which(bitcoin$Date == '2017-10-02 23:59:59'): dim(bitcoin)
[1],]
dogecoin <- dogecoin[which(dogecoin$Date == '2017-10-02 23:59:59'):
dim(dogecoin)[1],]
eos <- eos[which(eos$Date == '2017-10-02 23:59:59'): dim(eos)[1],]
ethereum <- ethereum[which(ethereum$Date == '2017-10-02 23:59:59'):
dim(ethereum)[1],]
iota <- iota[which(iota$Date == '2017-10-02 23:59:59'): dim(iota)[1],]
litecoin <- litecoin[which(litecoin$Date == '2017-10-02 23:59:59'):
dim(litecoin)[1],]
monero <- monero[which(monero$Date == '2017-10-02 23:59:59'): dim(monero)[1],]
stellar <- stellar[which(stellar$Date == '2017-10-02 23:59:59'): dim(stellar)
[1],]
tron <- tron[which(tron$Date == '2017-10-02 23:59:59'): dim(tron)[1],]

#check to see if dates match
all.equal(bitcoin$Date,
          cardano$Date,
          dogecoin$Date,
          eos.df$Date,
          iota$Date,
          iota.df$Date,
```

```
          litecoin$Date,
          monero$Date,
          stellar$Date,
          tron$Date
)

# create composite data frame
cryptos <- data.frame(bitcoin$Date,
                       bitcoin$Symbol,  bitcoin$High,  bitcoin$Low,
bitcoin$Open,    bitcoin$Close,      bitcoin$Volume,  bitcoin$Marketcap,
                       cardano$Symbol,  cardano$High,  cardano$Low,
cardano$Open,    cardano$Close,      cardano$Volume,  cardano$Marketcap,
                       dogecoin$Symbol, dogecoin$High, dogecoin$Low,
dogecoin$Open,   dogecoin$Close,     dogecoin$Volume, dogecoin$Marketcap,
                       eos$Symbol,      eos$High,      eos$Low,       eos$Open,
eos$Close,          eos$Volume,      eos$Marketcap,
                       ethereum$Symbol, ethereum$High, ethereum$Low,
ethereum$Open,   ethereum$Close,     ethereum$Volume, ethereum$Marketcap,
                       iota$Symbol,     iota$High,     iota$Low,
iota$Open,       iota$Close,         iota$Volume,     iota$Marketcap,
                       litecoin$Symbol, litecoin$High, litecoin$Low,
litecoin$Open,   litecoin$Close,     litecoin$Volume, litecoin$Marketcap,
                       monero$Symbol,   monero$High,   monero$Low,
monero$Open,     monero$Close,       monero$Volume,   monero$Marketcap,
                       stellar$Symbol,  stellar$High,  stellar$Low,
stellar$Open,    stellar$Close,      stellar$Volume,  stellar$Marketcap,
                       tron$Symbol,     tron$High,     tron$Low,
tron$Open,       tron$Close,         tron$Volume,     tron$Marketcap)
# give intuitive column names
colnames(cryptos) <- c('date',
                        'btc.symbol',  'btc.high',  'btc.low',  'btc.open',
'btc.close',  'btc.vol',  'btc.mktcap',
                        'ada.symbol',  'ada.high',  'ada.low',  'ada.open',
'ada.close',  'ada.vol',  'ada.mktcap',
                        'doge.symbol', 'doge.high', 'doge.low', 'doge.open',
'doge.close', 'doge.vol', 'doge.mktcap',
                        'eos.symbol',  'eos.high',  'eos.low',  'eos.open',
'eos.close',  'eos.vol',  'eos.mktcap',
                        'eth.symbol',  'eth.high',  'eth.low',  'eth.open',
'eth.close',  'eth.vol',  'eth.mktcap',
                        'iota.symbol', 'iota.high', 'iota.low', 'iota.open',
'iota.close', 'iota.vol', 'iota.mktcap',
                        'ltc.symbol',  'ltc.high',  'ltc.low',  'ltc.open',
'ltc.close',  'ltc.vol',  'ltc.mktcap',
                        'xmr.symbol',  'xmr.high',  'xmr.low',  'xmr.open',
'xmr.close',  'xmr.vol',  'xmr.mktcap',
                        'xlm.symbol',  'xlm.high',  'xlm.low',  'xlm.open',
'xlm.close',  'xlm.vol',  'xlm.mktcap',
                        'trx.symbol',  'trx.high',  'trx.low',  'trx.open',
'trx.close',  'trx.vol',  'trx.mktcap')

# change from character to date; remove hours, minutes, seconds
cryptos$date <- as.Date(cryptos$date)

# create calculated columns
```

```
attach(cryptos)
# create a function to calculate the percent change over 24 hours between the
values in any two columns
daily_change <- function(current_value, previous_value) {
  DailyPercentChange <- 100*(current_value - previous_value)/previous_value
  return(DailyPercentChange)
}

# calculate daily percent returns for each coin
cryptos$btc.daily_return  <- daily_change(btc.close, btc.open)
cryptos$ada.daily_return  <- daily_change(ada.close,ada.open)
cryptos$doge.daily_return <- daily_change(doge.close, doge.open)
cryptos$eos.daily_return  <- daily_change(eos.close, eos.open)
cryptos$eth.daily_return  <- daily_change(eth.close, eth.open)
cryptos$iota.daily_return <- daily_change(iota.close, iota.open)
cryptos$ltc.daily_return  <- daily_change(ltc.close, ltc.open)
cryptos$xmr.daily_return  <- daily_change(xmr.close, xmr.open)
cryptos$xlm.daily_return  <- daily_change(xlm.close, xlm.open)
cryptos$trx.daily_return  <- daily_change(trx.close, trx.open)

# create a function to calculate daily percent change in volume
daily_volume_change <- function(coin.vol){
  volume_change <- daily_change(coin.vol, c(coin.vol[1], coin.vol[-
length(coin.vol)]))
  return(volume_change)
}

# calculate daily change in volume for each coin
cryptos$btc.daily_volume_change <- daily_volume_change(btc.vol)
cryptos$ada.daily_volume_change <- daily_volume_change(ada.vol)
cryptos$doge.daily_volume_change <- daily_volume_change(doge.vol)
cryptos$eos.daily_volume_change <- daily_volume_change(eos.vol)
cryptos$eth.daily_volume_change <- daily_volume_change(eth.vol)
cryptos$iota.daily_volume_change <- daily_volume_change(iota.vol)
cryptos$ltc.daily_volume_change <- daily_volume_change(ltc.vol)
cryptos$xmr.daily_volume_change <- daily_volume_change(xmr.vol)
cryptos$xlm.daily_volume_change <- daily_volume_change(xlm.vol)
cryptos$trx.daily_volume_change <- daily_volume_change(trx.vol)

# create a function to define 'big day' as a day with at least 5% returns
isBigDay <- function(col){
  returncol <- c()
  for(i in 1:length(col)){
    if(col[i]>=5){
      returncol[i] <- 1
    } else {
      returncol[i] <- 0
    }
  }
  return(returncol)
}

attach(cryptos)

# make a big_days column
cryptos$btc.big_day <- as.factor(isBigDay(btc.daily_return))
```

```
cryptos$ada.big_day <- as.factor(isBigDay(ada.daily_return))
cryptos$doge.big_day <- as.factor(isBigDay(doge.daily_return))
cryptos$eos.big_day <- as.factor(isBigDay(eos.daily_return))
cryptos$eth.big_day <- as.factor(isBigDay(eth.daily_return))
cryptos$iota.big_day <- as.factor(isBigDay(iota.daily_return))
cryptos$ltc.big_day <- as.factor(isBigDay(ltc.daily_return))
cryptos$xmr.big_day <- as.factor(isBigDay(xmr.daily_return))
cryptos$xlm.big_day <- as.factor(isBigDay(xlm.daily_return))
cryptos$trx.big_day <- as.factor(isBigDay(trx.daily_return))


# create a function to check vector for NAs, just in case there are NAs in the
calculated columns
NAcheck <- function(vector) {
  if (sum(is.na(vector)) > 0)
    paste("WARNING: There are", sum(is.na(vector)), "NAs in",
deparse(substitute(vector)))
  else paste("0 NAs")
}

# search every column for NAs
sapply(cryptos, NAcheck)

# melt data frame into 3 columns: date, coin, daily return
# will be used for heat map analysis

cryptos.molten <- melt(data=cryptos, id.vars='date',
measure.vars=colnames(cryptos)[72:81])
# View(cryptos.molten)

#------------------------------------------------------------------------------
# Structure and Summary of Munged Data
#------------------------------------------------------------------------------

summary(cryptos)
str(cryptos)

summary(cryptos.molten)
str(cryptos.molten)
unique(cryptos.molten$variable)

#----------------------------------------------------------------------------------------
# Interesting Findings after Munging
#----------------------------------------------------------------------------------------

# market cap
mktcap.multiLine <- ggplot(cryptos, aes(x = date)) + geom_line(aes (y =
btc.mktcap, color = "btc.mktcap")) +
  geom_line(aes (y = doge.mktcap, color = "doge.mktcap")) + geom_line(aes(y =
xmr.mktcap, color = "xmr.mktcap")) +
  geom_line(aes(y = xlm.mktcap, color = "xlm.mktcap")) + geom_line(aes(y =
ltc.mktcap, color = "ltc.mktcap")) +
  geom_line(aes(y = ada.mktcap, color = "ada.mktcap")) + geom_line(aes(y =
eos.mktcap, color = "eos.mktcap")) +
  geom_line(aes(y = eth.mktcap, color = "eth.mktcap")) + geom_line(aes(y =
iota.mktcap, color = "iota.mktcap")) +
```

```
  geom_line(aes(y = trx.mktcap, color = "trx.mktcap"))
mktcap.multiLine <- mktcap.multiLine + ggtitle("Market Capitalization by
Cryptocurrency") + ylab('Market Capitalization in USD')
mktcap.multiLine

# daily returns
daily_return.multiLine <- ggplot(cryptos, aes(x = date)) + geom_line(aes (y =
btc.daily_return, color = "btc.daily_return")) +
  geom_line(aes (y = doge.daily_return, color = "doge.daily_return")) +
geom_line(aes(y = xmr.daily_return, color = "xmr.daily_return")) +
  geom_line(aes(y = xlm.daily_return, color = "xlm.daily_return")) +
geom_line(aes(y = ltc.daily_return, color = "ltc.daily_return")) +
  geom_line(aes(y = ada.daily_return, color = "ada.daily_return")) +
geom_line(aes(y = eos.daily_return, color = "eos.daily_return")) +
  geom_line(aes(y = eth.daily_return, color = "eth.daily_return")) +
geom_line(aes(y = iota.daily_return, color = "iota.daily_return")) +
  geom_line(aes(y = trx.daily_return, color = "trx.daily_return"))
daily_return.multiLine <- daily_return.multiLine + ggtitle("Daily Return by
Cryptocurrency") + ylab('Daily Return (Percentage)')
daily_return.multiLine

# trading volume
volume.multiLine <- ggplot(cryptos, aes(x = date)) + geom_line(aes (y =
btc.vol, color = "btc.vol")) +
  geom_line(aes (y = doge.vol, color = "doge.vol")) + geom_line(aes(y =
xmr.vol, color = "xmr.vol")) +
  geom_line(aes(y = xlm.vol, color = "xlm.vol")) + geom_line(aes(y = ltc.vol,
color = "ltc.vol")) +
  geom_line(aes(y = ada.vol, color = "ada.vol")) + geom_line(aes(y = eos.vol,
color = "eos.vol")) +
  geom_line(aes(y = eth.vol, color = "eth.vol")) + geom_line(aes(y = iota.vol,
color = "iota.vol")) +
  geom_line(aes(y = trx.vol, color = "trx.vol"))
volume.multiLine <- volume.multiLine + ggtitle("Trading Volume by
Cryptocurrency") + ylab('Volume')
volume.multiLine

# daily volume change
daily_volume_change.multiLine <- ggplot(cryptos, aes(x = date)) +
geom_line(aes (y = btc.daily_volume_change, color =
"btc.daily_volume_change")) +
  geom_line(aes (y = doge.daily_volume_change, color =
"doge.daily_volume_change")) + geom_line(aes(y = xmr.daily_volume_change,
color = "xmr.daily_volume_change")) +
  geom_line(aes(y = xlm.daily_volume_change, color =
"xlm.daily_volume_change")) + geom_line(aes(y = ltc.daily_volume_change, color
= "ltc.daily_volume_change")) +
  geom_line(aes(y = ada.daily_volume_change, color =
"ada.daily_volume_change")) + geom_line(aes(y = eos.daily_volume_change, color
= "eos.daily_volume_change")) +
  geom_line(aes(y = eth.daily_volume_change, color =
"eth.daily_volume_change")) + geom_line(aes(y = iota.daily_volume_change,
color = "iota.daily_volume_change")) +
  geom_line(aes(y = trx.daily_volume_change, color = "trx.vol"))
```

```
daily_volume_change.multiLine <- daily_volume_change.multiLine +
ggtitle("Daily Volume Change by Cryptocurrency") + ylab('Daily Volume Change
(Percent)')
daily_volume_change.multiLine

# price
price.multiLine <- ggplot(cryptos, aes(x = date)) + geom_line(aes (y =
btc.high, color = "btc.high")) +
  geom_line(aes (y = doge.high, color = "doge.high")) + geom_line(aes(y =
xmr.high, color = "xmr.high")) +
  geom_line(aes(y = xlm.high, color = "xlm.high")) + geom_line(aes(y =
ltc.high, color = "ltc.high")) +
  geom_line(aes(y = ada.high, color = "ada.high")) + geom_line(aes(y =
eos.high, color = "eos.high")) +
  geom_line(aes(y = eth.high, color = "eth.high")) + geom_line(aes(y =
iota.high, color = "iota.high")) +
  geom_line(aes(y = trx.high, color = "trx.high"))
price.multiLine <- price.multiLine + ggtitle("Price by Cryptocurrency") +
ylab('Daily High in USD')
price.multiLine

#-----------------------------------------------------------------------------------------
# III. DATA SET DESCRIPTIVE STATISTICS AND STRUCTURE
#-----------------------------------------------------------------------------------------
# price vs volume by coin

#btc - price and volume
btc.chart <- ggplot(data=cryptos, aes(date, btc.high)) +
  geom_point(aes(color=btc.vol, size=btc.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('btc - price and volume')
btc.chart
# we can see that as the price spikes, volume also spikes

#ada - price and volume
ada.chart <- ggplot(data=cryptos, aes(date, ada.high)) +
  geom_point(aes(color=ada.vol, size=ada.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('ada - price and volume')
ada.chart
# the trend looks consistent and even stronger in cardano

#doge - price and volume
doge.chart <- ggplot(data=cryptos, aes(date, doge.high)) +
  geom_point(aes(color=doge.vol, size=doge.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('doge - price and volume')
doge.chart

#eos - price and volume
eos.chart <- ggplot(data=cryptos, aes(date, eos.high)) +
  geom_point(aes(color=eos.vol, size=eos.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
```

```
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('eos - price and volume')
eos.chart
# increase in EOS volume is perhaps a bullish signal based on our inferences

#eth - price and volume
eth.chart <- ggplot(data=cryptos, aes(date, eth.high)) +
  geom_point(aes(color=eth.vol, size=eth.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('eth - price and volume')
eth.chart

#iota - price and volume
iota.chart <- ggplot(data=cryptos, aes(date, iota.high)) +
  geom_point(aes(color=iota.vol, size=iota.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('iota - price and volume')
iota.chart

#ltc - price and volume
ltc.chart <- ggplot(data=cryptos, aes(date, ltc.high)) +
  geom_point(aes(color=ltc.vol, size=ltc.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('ltc - price and volume')
ltc.chart

#xmr - price and volume
xmr.chart <- ggplot(data=cryptos, aes(date, xmr.high)) +
  geom_point(aes(color=xmr.vol, size=xmr.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('xmr - price and volume')
xmr.chart

#xlm - price and volume
xlm.chart <- ggplot(data=cryptos, aes(date, xlm.high)) +
  geom_point(aes(color=xlm.vol, size=xlm.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('xlm - price and volume')
xlm.chart

#trx - price and volume
trx.chart <- ggplot(data=cryptos, aes(date, trx.high)) +
  geom_point(aes(color=trx.vol, size=trx.vol), alpha=0.3) +
  scale_color_gradient(high='red', low='white') +
  theme(panel.background = element_rect(fill = "darkblue")) +
  ggtitle('trx - price and volume')
trx.chart

# Altcoin to BTC price correlation plots

#ada to btc - correlation
```

```
adatobtc.plot <- ggplot(data=cryptos, aes(ada.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('ADA to BTC Price Correlation')
adatobtc.plot

#doge to btc - correlation
dogetobtc.plot <- ggplot(data=cryptos, aes(doge.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('DOGE to BTC Price Correlation')
dogetobtc.plot

#eos to btc - correlation
eostobtc.plot <- ggplot(data=cryptos, aes(eos.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('EOS to BTC Price Correlation')
eostobtc.plot

#eth to btc - correlation
ethtobtc.plot <- ggplot(data=cryptos, aes(eth.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('ETH to BTC Price Correlation')
ethtobtc.plot

#iota to btc - correlation
iotatobtc.plot <- ggplot(data=cryptos, aes(iota.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('Iota to BTC Price Correlation')
iotatobtc.plot
#ltc to btc - correlation
ltctobtc.plot <- ggplot(data=cryptos, aes(ltc.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('LTC to BTC Price Correlation')
ltctobtc.plot
#xmr to btc - correlation
xmrtobtc.plot <- ggplot(data=cryptos, aes(xmr.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('XMR to BTC Price Correlation')
xmrtobtc.plot
#xlm to btc - correlation
xlmtobtc.plot <- ggplot(data=cryptos, aes(xlm.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('XLM to BTC Price Correlation')
xlmtobtc.plot
#trx to btc - correlation
trxtobtc.plot <- ggplot(data=cryptos, aes(trx.high, btc.high)) +
  geom_jitter() + geom_smooth() + ggtitle('TRX to BTC Price Correlation')
trxtobtc.plot

grid.arrange(adatobtc.plot, eostobtc.plot, nrow=1) #eos is the outlier - only
one with p-value that is not near-zero (0.6)
grid.arrange(dogetobtc.plot, ethtobtc.plot, nrow=1)

# heat map with all daily returns

# creating heat map
ggplot(data=cryptos.molten, aes(x=date, y=variable, group=variable,
fill=value)) +
  geom_tile(size=0.1) + ggtitle('Volatility Heatmap') +
  scale_fill_gradient2(low='red', mid='white', high='black', midpoint=0)
# on the heat map, we can see the darkest days as the days with extreme
volatility on the positive side
# anything areas with a shade of red indicates extreme negative volatility

summary(cryptos.molten)
```

```
min(cryptos.molten$value)
max(cryptos.molten$value)
quantile(cryptos.molten$value, c(0.01, 0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,
0.99))
# 98% percent of the percent return values lie between -15.08 and 20.82
percent
# the min and max are -42.3 and 355.62, respectively, indicating some heavy
outliers

# which altcoin had the most days out of the top 100
# which of the coins produced the most positive outliers?
cryptos.100_most_positive_returns <- cryptos.molten[order(-
cryptos.molten$value),]
cryptos.100_most_positive_returns <- cryptos.100_most_positive_returns[1:100,]
cryptos.100_most_positive_returns # view 100 most positive return days
cryptos.100_most_positive_returns <- cryptos.
100_most_positive_returns[order(cryptos.100_most_positive_returns$variable, -
cryptos.100_most_positive_returns$value),]
cryptos.100_most_positive_returns # view 100 most positive return days by
crypto
# only one of the 100 most positive returns belonged to btc
summary(cryptos.100_most_positive_returns$variable)
# stellar had 18 of the top 100 positive outliers, followed by tron (17),
cardano (15), iota (14), doge (13), eos (11)

# which of the coins produced the most negative outliers?
cryptos.100_most_negative_returns <-
cryptos.molten[order(cryptos.molten$value),]
cryptos.100_most_negative_returns <- cryptos.100_most_negative_returns[1:100,]
cryptos.100_most_negative_returns # view 100 most negative return days
cryptos.100_most_negative_returns <- cryptos.
100_most_negative_returns[order(cryptos.100_most_negative_returns$variable,
cryptos.100_most_negative_returns$value),]
cryptos.100_most_negative_returns # view 100 most negative return days by
crypto
summary(cryptos.100_most_negative_returns$variable)
#btc had 2 of the top 100 negative outliers
#tron had the most (18), eos (16), iota (12), doge (10), cardano (10), stellar
(10)

positiveData <- summary(cryptos.100_most_positive_returns$variable)
cryptoNames <- names(positiveData)
names(positiveData) <- NULL
pData <- data.frame(cryptoNames, positiveData)
pData
top100returns <- ggplot(pData, aes(cryptoNames, positiveData)) +
  geom_bar(stat='identity', aes(fill=cryptoNames)) +
  theme(axis.text.x=element_text(angle = -90, hjust = 0)) +
  theme(panel.background = element_rect(fill = 'white')) +
  ylab('Number of Days') + xlab('Cryptocurrency') +
  ggtitle('Top 100 Return Days by Crypto')

negativeData <- summary(cryptos.100_most_negative_returns$variable)
cryptoNames <- names(negativeData)
names(negativeData) <- NULL
nData <- data.frame(cryptoNames, negativeData)
```

```
neg100returns <- ggplot(nData, aes(cryptoNames, negativeData)) +
  geom_bar(stat='identity', aes(fill=cryptoNames)) +
  theme(axis.text.x=element_text(angle = -90, hjust = 0)) +
  theme(panel.background = element_rect(fill = 'white')) +
  ylab('Number of Days') + xlab('Cryptocurrency') +
  ggtitle('Lowest 100 Return Days by Crypto')

grid.arrange(top100returns, neg100returns, nrow=1)

# Histograms

btc.histdata <- cryptos[order(-cryptos$btc.daily_return),
c('date','btc.symbol','btc.high','btc.daily_return')]
btc.hist <- ggplot(data=btc.histdata, aes(x=btc.daily_return)) +
geom_histogram()

ada.histdata <- cryptos[order(-cryptos$ada.daily_return),
c('date','ada.symbol','ada.high','ada.daily_return')]
ada.hist <- ggplot(data=ada.histdata, aes(x=ada.daily_return)) +
geom_histogram()

doge.histdata <- cryptos[order(-cryptos$doge.daily_return),
c('date','doge.symbol','doge.high','doge.daily_return')]
doge.hist <- ggplot(data=doge.histdata, aes(x=doge.daily_return)) +
geom_histogram()

eos.histdata <- cryptos[order(-cryptos$eos.daily_return),
c('date','eos.symbol','eos.high','eos.daily_return')]
eos.hist <- ggplot(data=eos.histdata, aes(x=eos.daily_return)) +
geom_histogram()

eth.histdata <- cryptos[order(-cryptos$eth.daily_return),
c('date','eth.symbol','eth.high','eth.daily_return')]
eth.hist <- ggplot(data=eth.histdata, aes(x=eth.daily_return)) +
geom_histogram()

iota.histdata <- cryptos[order(-cryptos$iota.daily_return),
c('date','iota.symbol','iota.high','iota.daily_return')]
iota.hist <- ggplot(data=iota.histdata, aes(x=iota.daily_return)) +
geom_histogram()

ltc.histdata <- cryptos[order(-cryptos$ltc.daily_return),
c('date','ltc.symbol','ltc.high','ltc.daily_return')]
ltc.hist <- ggplot(data=ltc.histdata, aes(x=ltc.daily_return)) +
geom_histogram()

xmr.histdata <- cryptos[order(-cryptos$xmr.daily_return),
c('date','xmr.symbol','xmr.high','xmr.daily_return')]
xmr.hist <- ggplot(data=xmr.histdata, aes(x=xmr.daily_return)) +
geom_histogram()

xlm.histdata <- cryptos[order(-cryptos$xlm.daily_return),
c('date','xlm.symbol','xlm.high','xlm.daily_return')]
xlm.hist <- ggplot(data=xlm.histdata, aes(x=xlm.daily_return)) +
geom_histogram()
```

```
trx.histdata <- cryptos[order(-cryptos$trx.daily_return),
c('date','trx.symbol','trx.high','trx.daily_return')]
trx.hist <- ggplot(data=trx.histdata, aes(x=trx.daily_return)) +
geom_histogram()

grid.arrange(btc.hist, eth.hist, nrow=1) # we can see the daily percent
returns all follow a fairly normal distribution
grid.arrange(eos.hist, xlm.hist, nrow=1) # some altcoins have lower counts in
the 0 range, around the 400 range
grid.arrange(doge.hist, trx.hist, nrow=1) # some altcoins have very high
counts in the 0 range, in the 900 range

#--------------------------------------------------------------------------------------------
# IV. MODELS
#--------------------------------------------------------------------------------------------

# creating linear models
lm.adafrombtc <- lm(formula=ada.high~btc.high)
summary(lm.adafrombtc)

lm.eosfrombtc <- lm(formula=eos.high~btc.high)
summary(lm.eosfrombtc)
#interesting to see that btc~xlm has a p-value of 0.6
#compared to the near zero p-value in btc~ada

lm.iotafrombtc <- lm(formula=iota.high~btc.high)
summary(lm.iotafrombtc)

lm.xlmfrombtc <- lm(formula=xlm.high~btc.high)
summary(lm.xlmfrombtc)

lm.trxfrombtc <- lm(formula=trx.high~btc.high)
summary(lm.trxfrombtc)

lm.dogefrombtc <- lm(formula=doge.high~btc.high)
summary(lm.dogefrombtc)
#eos is the only coin that seems to have a life of its own, so to speak,
#in regards to its price movement

#--------------------------------------------------------------------------------------------
# creating SVMs
#--------------------------------------------------------------------------------------------
# use daily return, and daily volume change to predict volatile days in
altcoins

attach(cryptos)
#preparing train and test data
set.seed(150)
randIndex <- sample(1:dim(cryptos)[1]) #creating randomized index to ensure
fairness in data selection
cutPoint9_10 <- floor(9 * dim(cryptos)[1]/10) #creating cutpoint to group 9/10
of randomly selected data for training
trainData <-cryptos[randIndex[1:cutPoint9_10],] #create training data set
testData <-cryptos[randIndex[(cutPoint9_10+1):dim(cryptos)[1]],] #create
testing data set
```

```
#Cardano
ada.svm1 <- svm(ada.big_day~btc.daily_return, data=trainData) #creating models
ada.svm1
ada.svm2 <- svm(ada.big_day~btc.daily_volume_change, data=trainData)
ada.svm2
ada.svm3 <- svm(ada.big_day~btc.daily_volume_change + btc.daily_return,
data=trainData)
ada.svm3
ada.predict1 <- predict(ada.svm1, testData) #creating predictions
ada.predict2 <- predict(ada.svm2, testData)
ada.predict3 <- predict(ada.svm3, testData)
ada.error1 <- as.numeric(testData$ada.big_day) - as.numeric(ada.predict1)
#measuring errors
ada.error2 <- as.numeric(testData$ada.big_day) - as.numeric(ada.predict2)
ada.error3 <- as.numeric(testData$ada.big_day) - as.numeric(ada.predict3)

# model 1 accuracy
ada.svm1.predict.accuracy <- (table(ada.predict1 ,testData$ada.big_day)[1,1]
                              + table(ada.predict1 ,testData$ada.big_day)
[2,2]) / length(testData$ada.big_day)
ada.svm1.predict.accuracy

# model 2 accuracy
ada.svm2.predict.accuracy <- (table(ada.predict2 ,testData$ada.big_day)[1,1]
                              + table(ada.predict2 ,testData$ada.big_day)
[2,2]) / length(testData$ada.big_day)
ada.svm2.predict.accuracy

# model 3 accuracy
ada.svm3.predict.accuracy <- (table(ada.predict3 ,testData$ada.big_day)[1,1]
                              + table(ada.predict3 ,testData$ada.big_day)
[2,2]) / length(testData$ada.big_day)
ada.svm3.predict.accuracy

#Eos
eos.svm1 <- svm(eos.big_day~btc.daily_return, data=trainData) #creating models
eos.svm1
eos.svm2 <- svm(eos.big_day~btc.daily_volume_change, data=trainData)
eos.svm2
eos.svm3 <- svm(eos.big_day~btc.daily_volume_change + btc.daily_return,
data=trainData)
eos.svm3
eos.predict1 <- predict(eos.svm1, testData) #creating predictions
eos.predict2 <- predict(eos.svm2, testData)
eos.predict3 <- predict(eos.svm3, testData)
eos.error1 <- as.numeric(testData$eos.big_day) - as.numeric(eos.predict1)
#measuring errors
eos.error2 <- as.numeric(testData$eos.big_day) - as.numeric(eos.predict2)
eos.error3 <- as.numeric(testData$eos.big_day) - as.numeric(eos.predict3)

# model 1 accuracy
eos.svm1.predict.accuracy <- (table(eos.predict1 ,testData$eos.big_day)[1,1]
                              + table(eos.predict1 ,testData$eos.big_day)
[2,2]) / length(testData$eos.big_day)
eos.svm1.predict.accuracy
```

```
# model 2 accuracy
eos.svm2.predict.accuracy <- (table(eos.predict2 ,testData$eos.big_day)[1,1]
                              + table(eos.predict2 ,testData$eos.big_day)
[2,2]) / length(testData$eos.big_day)
eos.svm2.predict.accuracy

# model 3 accuracy
eos.svm3.predict.accuracy <- (table(eos.predict3 ,testData$eos.big_day)[1,1]
                              + table(eos.predict3 ,testData$eos.big_day)
[2,2]) / length(testData$eos.big_day)
eos.svm3.predict.accuracy


#eth
eth.svm1 <- svm(eth.big_day~btc.daily_return, data=trainData) #creating models
eth.svm1
eth.svm2 <- svm(eth.big_day~btc.daily_volume_change, data=trainData)
eth.svm2
eth.svm3 <- svm(eth.big_day~btc.daily_volume_change + btc.daily_return,
data=trainData)
eth.svm3
eth.predict1 <- predict(eth.svm1, testData) #creating predictions
eth.predict2 <- predict(eth.svm2, testData)
eth.predict3 <- predict(eth.svm3, testData)
eth.error1 <- as.numeric(testData$eth.big_day) - as.numeric(eth.predict1)
#measuring errors
eth.error2 <- as.numeric(testData$eth.big_day) - as.numeric(eth.predict2)
eth.error3 <- as.numeric(testData$eth.big_day) - as.numeric(eth.predict3)

# model 1 accuracy
eth.svm1.predict.accuracy <- (table(eth.predict1 ,testData$eth.big_day)[1,1]
                              + table(eth.predict1 ,testData$eth.big_day)
[2,2]) / length(testData$eth.big_day)
eth.svm1.predict.accuracy

# model 2 accuracy
eth.svm2.predict.accuracy <- (table(eth.predict2 ,testData$eth.big_day)[1,1]
                              + table(eth.predict2 ,testData$eth.big_day)
[2,2]) / length(testData$eth.big_day)
eth.svm2.predict.accuracy

# model 3 accuracy
eth.svm3.predict.accuracy <- (table(eth.predict3 ,testData$eth.big_day)[1,1]
                              + table(eth.predict3 ,testData$eth.big_day)
[2,2]) / length(testData$eth.big_day)
eth.svm3.predict.accuracy

#to predict the big return days for altcoins, using only the volatility of btc
was the best performer
# adding the percent change in volume of btc did not improve the model
#at times it actually weakened the model

detach(cryptos)
```