e^i Ventures – Solana Study

Sam Reeves

Task: Pick your favorite crypto project and produce a report or analysis on your insights for the project.

Describe your overall methodology in terms of data collection, data cleaning, and visualizations.

In a general high-level view, I approach all studies with two questions in mind: What question do we need to answer? and What information which could contain this truth is readily available? One of these two questions will provide the limitations for the other, and it will be clear which path to take.

For data collection, ideally a massive amount of information which is either quantitative or which can be quantified is a good starting position. Nearly everything can be quantified in some way, by ordering factors, onehot encoding, etc. The next logical question is something like: How does this information interact to produce the truth that I am looking for?

After these considerations, data cleaning is a routine effort: 1) combine the information so that individual observations are given in rows, with each detail in the columns. 2) Deal with missing information in a reasonable way. This could mean throwing out observations, substituting null values with a mean or simulating realistic information using other similar data. 3) Normalizing the data between 0 and 1. 4) For nonlinear models, discover some power transformation so that a linear model can be fitted, or, in the case of timeseries data, remove seasonal or cyclical elements and normalize for a trend line.

Visualizations are useful for explaining the importance of an observational study or experiment, but they definitely shouldn't be the starting point for a study. Often times, there will be many many dimensions of a model, and trying to visualize more than two at a time is not helpful.

Give an overview of the project, as well as some high-level statistics

Honestly, I do not have a favorite crypto project. I was extremely excited when the initial BTC whitepaper was published, and I loved the process of mining on my Raspberry Pi 2 with ASIC chips and a little fan. I have created some NFTs but not sold many. My uncle is invested in a few altcoins, and he suggested I poke around at Solana.

I gather that this coin has gone from ~$20 to about ~$200 this year. Just from reading about it, I imagine that this has something to do with FTX selecting it for their "Serum", however, many people seem excited about its "smart" functionality, high transaction rate, and comparatively low fees. Maybe, it can be considered a realistic alternative to ETH.

I will take a look at the candlestick chart data and try to determine what events could have influenced price fluctuations. I expect that fitting a linear or ARIMA model (to the naked OHLC data alone) will not work very well. The price fluctuations of crypto assets often have more to do with new players moving in and big players moving out than anything else. As such, each movement changes the supply/demand scenario which generates prices.

What might be fruitful is using Google Trends data to predict fluctuations in publix interest. If we can establish some leading indicator for these big market movements, then classical momentum trading activities would probably be successful.

For example, if we know ahead of time what the momentum or rate of change of volume of trades may be, then we can use MACD and stochastic indicators to know which direction the price will shift during the period of increased trading. A sharp downward turn in momentum often indicates what economists have called "the dead cat bounce".

Collect some data about the project

There is an R package that functions as a wrapper for the Cryptowatch REST API: https://docs.cryptowat.ch/rest-api

As always, we capture the data and load it from a file so we don't max out our API call allowance.

```
#sol <- get_assets('sol')
#sol_usd <- get_ohlc(pair = 'solusd')

#write.csv(sol_usd, 'sol_usd.csv')

sol_usd <- read.csv('sol_usd.csv') %>%
  select(-1)

sol_usd$CloseTime <- lubridate::as_date(sol_usd$CloseTime)

head(sol_usd)
```

```
##    CloseTime OpenPrice HighPrice LowPrice ClosePrice    Volume QuoteVolume
## 1 2021-06-18     40.23     40.57    38.30      39.13  6007.618    236628.4
## 2 2021-06-19     39.14     39.36    35.00      36.62 13557.357    502474.7
## 3 2021-06-20     36.82     37.35    35.00      35.45 16986.164    609327.8
## 4 2021-06-21     35.09     35.89    31.48      35.28 38681.266   1306981.5
## 5 2021-06-22     35.08     35.08    26.00      26.55 41903.613   1257696.4
## 6 2021-06-23     26.43     28.69    20.20      26.18 48006.849   1193925.0
```

Here, we have OHLC data from the last 201 days. The default period is one data point per day.

We've also got some data from google trends representing the number of searches of "Solana" per week during the same time period.

My hope is that this will serve as a statistically relevant leading indicator for the price movements. This data doesn't tell us exactly how many searches occurred during the week, but it is already normalized between 0 and 100. Convenient.

```
sol_search <- read.csv('solana_searches.csv')
sol_search$week <- as_date(sol_search$week)

sol_search <- sol_search %>%
  filter(week > min(sol_usd$CloseTime) - 7) %>%
  rename(CloseTime = week)

data <- merge(sol_usd, sol_search,
              by = intersect(names(sol_usd), names(sol_search)),
              all.x = TRUE)
```

From the initial Google Trends data, we can see that the first two null values in the searches column are equal to 15. Since the weekly data is denominated by the first day of each week, we should fill the values forward.

There is a different function to fill with proportionate data in between two non-null values, but we cannot use it because of two reasons: 1) we don't know if the changes in search volume are consistent and linear, and 2) this is equivalent to looking into the future.

```
data$searches[1:2] <- 15
data$searches <- zoo::na.locf(data$searches, fromLast = FALSE)
```

For plotting trends and training models, it's helpful to normalize the data in terms of the minima and maxima of each column, placing everything except for the date (which really is just an index) between 0 and 1.

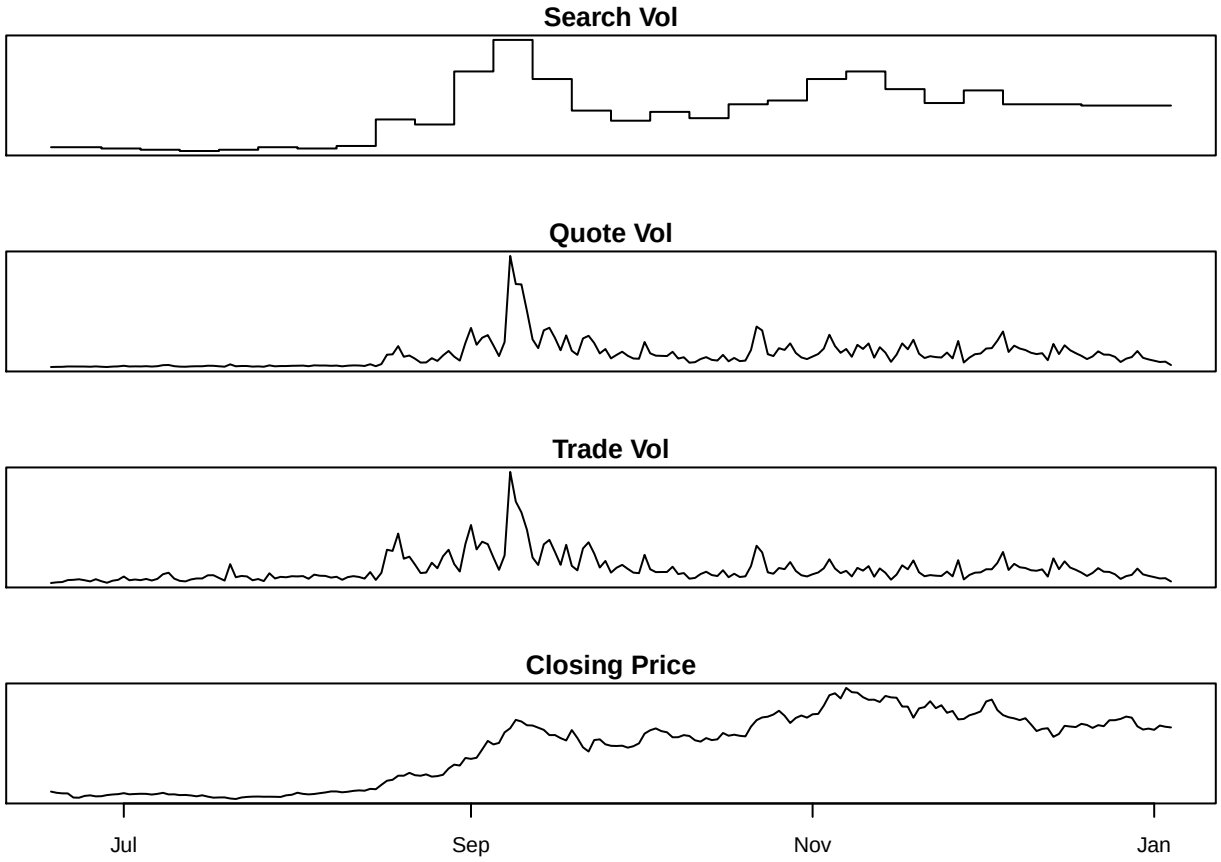
```
normalize <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}

normalize.all <- function(x) {
  date <- data$CloseTime

  tb <- lapply(x[-1], normalize) %>%
    as_tibble() %>%
    cbind(date)

  return(tb)
}

data.norm <- normalize.all(data)
```

Describe what insights can be obtained from the data.

These plots are given in (apparent) order of determination.

```
par(mfrow = c(4,1), cex=0.7, mai=c(0.3,0.1,0.2,0.1))
plot(data.norm$searches, x = data$CloseTime,
     main = 'Search Vol', type = 's',
```

```
     xaxt = 'n', yaxt = 'n')
plot(data.norm$QuoteVolume, x = data$CloseTime,
     main = 'Quote Vol', type = 'l',
     xaxt = 'n', yaxt = 'n')
plot(data.norm$Volume, x = data$CloseTime,
     main = 'Trade Vol', type = 'l',
     xaxt = 'n', yaxt = 'n')
plot(data.norm$ClosePrice, x = data$CloseTime,
     main = 'Closing Price', type = 'l',
     yaxt = 'n')
```



Based on these simple plots, it appears that the massive spike in volume and price which happened in September follows shortly after the huge spike in google searches for Solana. The volume of trades then normalizes, and the price normalizes around $200, with greater variance in trading Volume coinciding with healthy price fluctuations.

Prescriptive: What are your suggestions for the project?

https://www.fxstreet.com/cryptocurrencies/news/solana-price-is-on-the-verge-of-dipping-30-further-by-next-week-202109191409

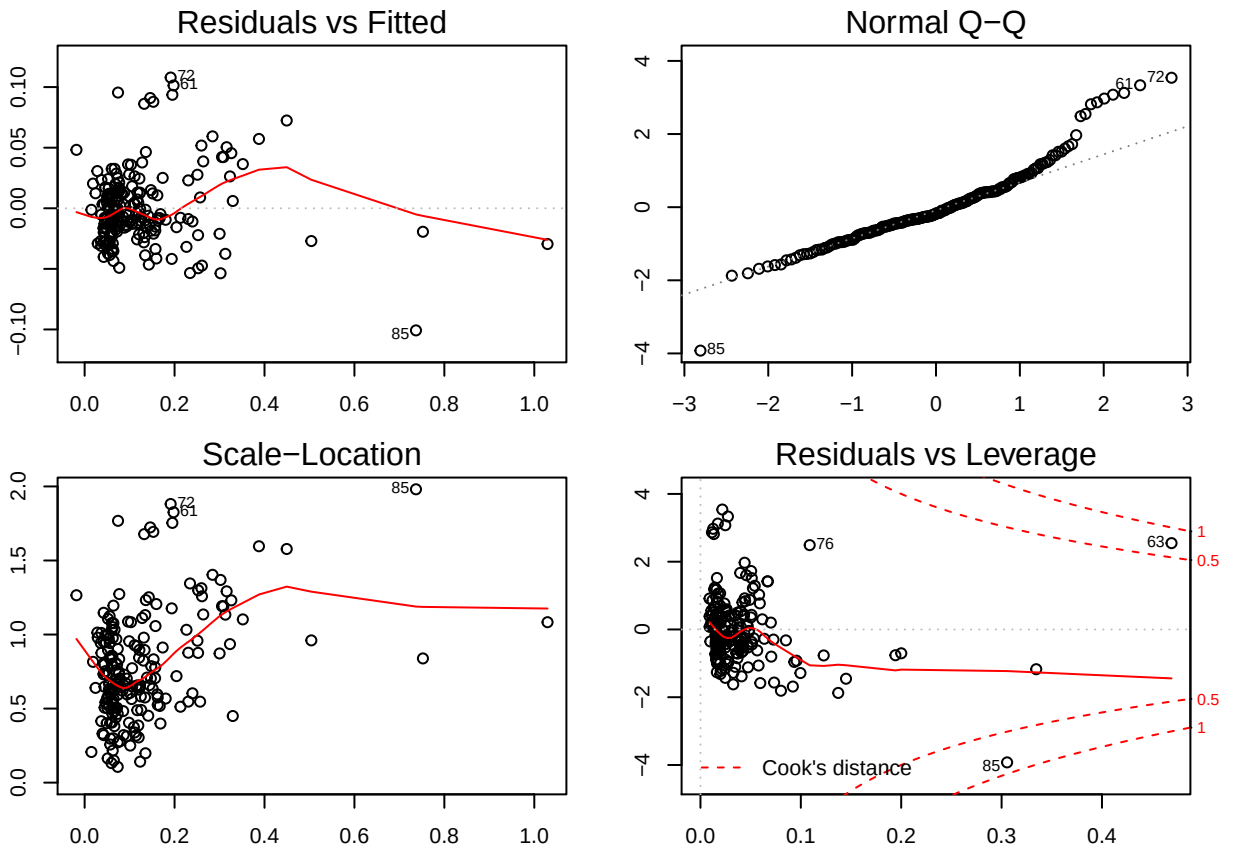
According to this article published on September 19, the crazy run had finished after reaching $223. I guess the magic number that caused players to exit the currency was a factor of 10x. A few things are apparent:

1. The Google search volume does appear to slightly lead the trade and quote volumes.
2. Quote volume is a bit sharper than trade volume.
3. During the September bull run, lots of people decided to buy and hold. The volume of trades drops back to previous levels, and the closing price normalizes considerably.

Predictive: what trends do you see in the data? Can these trends be validated statistically?

The purpose of this whole project was to interpret the trading history of Solana and to try to find a predictive model for trading volume. We will use all the data available as inputs for the trading volume in a linear model, and take a look at what the residuals can show us. Here, the residuals are the differences between the linear model fit against all the data and the real trading volume:

```
volume.linear <- lm(Volume ~ ., data.norm)

par(mfrow = c(2,2), cex=0.7, mai=c(0.3,0.3,0.3,0.3))
plot(volume.linear)
```



The first plot here shows us that there is not a direct linear relationship. However, a linear model is not useless. There is a place in the middle (this is the September trading spike) which drastically alters the cycle. The suggestion is that the most recent trading volume is appropriate or perhaps a bit low (!).

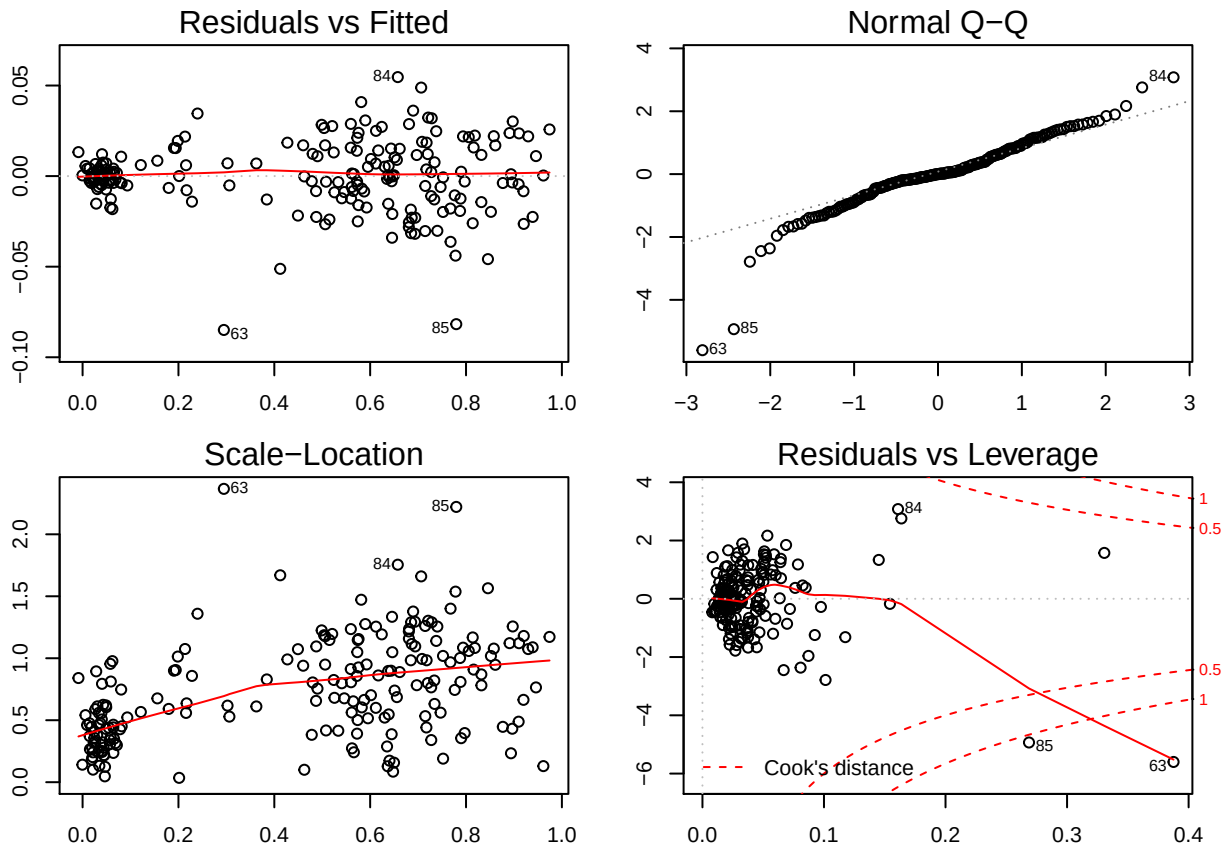
The Normal Q-Q plot is a representation of the theoretical quantiles versus the actual quantiles. If the relationship is visibly linear, then we know that the trained model coincides with reality.

The third plot shows fitted values on the x axis and the square root of the standardized residuals on the y axis. This tells us about the absolute value of the error across data points. In the line plots of the raw information from the above section, the story is corroborated: variance does increase after the spike, but apparently it averages out to a workable level if we consider a trading period longer than a day.

The final plot here shows us that points 63 and 85 are highly influential to the model.

Perhaps there is more. Let's try fitting a linear model to predict closing price:

```
price.linear <- lm(ClosePrice ~ ., data.norm)

par(mfrow = c(2,2), cex=0.7, mai=c(0.3,0.3,0.3,0.3))
plot(price.linear)
```



This is interesting, not at all what I expected.

The variance in the closing price is shockingly normal (according to the first plot), and the theoretical quantiles show us that the linear relationship predicting closing price is far better than the one predicting trading volume.

The Cook's distance plot suggests that the obvious outliers, points 63, 84, and 85, are highly influential to the model. If you removed these, the coefficients would change drastically.

Conclusions

It is absolutely possible to predict the trading volume and closing price of this cryptocurrency and perhaps many more by combining the historical information with a small amount of information pertaining to general interest. By collecting information from Reddit, Twitter, Meta, or other mass-participation media sources, we can observe what momentum traders might describe as investor emotion.

```
data[c(63, 85), c(1,5,6,8)]
```

```
##     CloseTime ClosePrice   Volume searches
## 63 2021-08-19       72.8 561324.5       37
## 85 2021-09-10      187.6 799965.1      100
```

Datapoints 63 and 85 correspond to August 19, the day following departure from a flat line, and September 10, the day following the big spike. At this last moment, trading volume increased by a factor of 10, and the price followed. Trading volume then went back down by a factor of 10 to its "normal" levels, but the price had been baked in.

```
aov(price.linear)
```

```
## Call:
##    aov(formula = price.linear)
##
## Terms:
##                 OpenPrice HighPrice  LowPrice    Volume QuoteVolume  searches
## Sum of Squares  19.337488  0.140693  0.040990  0.000090    0.002764  0.000961
## Deg. of Freedom         1         1         1         1           1         1
##                       date Residuals
## Sum of Squares   0.000615  0.072469
## Deg. of Freedom         1       193
##
## Residual standard error: 0.01937755
## Estimated effects may be unbalanced
```

Training a neural net on the same data, the coefficients converge and stabilize instantly. They yield exactly the same results:

```
price.nn <- multinom(ClosePrice ~ ., data.norm,
                  MaxNWts = 10000,
                  maxit = 100000)
```

```
## # weights:  1791 (1584 variable)
## initial  value 1063.954270
## iter  10 value 1063.195816
## iter  10 value 1063.195810
## final  value 1063.195793
## converged
```

```
aov(price.nn)
```

```
## Call:
##    aov(formula = price.nn)
##
## Terms:
##                 OpenPrice HighPrice  LowPrice    Volume QuoteVolume  searches
## Sum of Squares  19.337488  0.140693  0.040990  0.000090    0.002764  0.000961
## Deg. of Freedom         1         1         1         1           1         1
##                      date Residuals
## Sum of Squares   0.000615  0.072469
## Deg. of Freedom         1       193
##
## Residual standard error: 0.01937755
## Estimated effects may be unbalanced
```

The standard error is less than 2%. This model is overtrained, but with more data at a finer resolution, we could definitely make a proper generalized model. A period of news info shorter than 1 week would help immensely.

I did not preform a train/test split with predictions because I was interested in finding out what kind of relationships existed inside the dataset. It was outside of my scope. Considering these results, I would go ahead and make a predictive model using more data, training and test sets, and automated hyperparameter tuning.