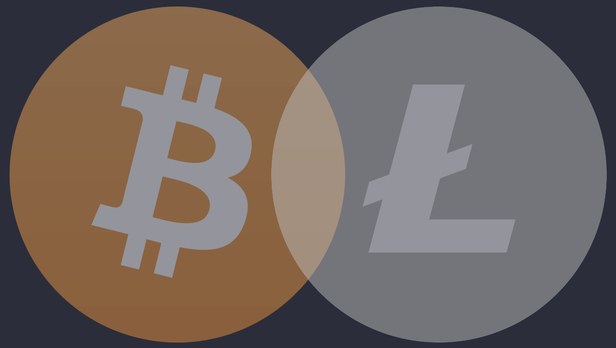
CRYPTOCURRENCY PRICE PREDICTION

Bitcoin & Litecoin Prices vs their Google Search Trend

David Racine



OVERVIEW

- This project will explore the relationship between the “closing price” of two established cryptocurrencies bitcoin(BTC) and litecoin(LTC) vs the respective frequency of their Google search terms "bitcoin price" and "litecoin price" respectively. A third search trend ("donald trump") is used to validate the method of comparing the Google search term trend for a cryptocurrency coin and its respective closing price.

- The presentation will be broken up into the functional sub sections: Data Wrangling, Inferential Statistics, and Machine Learning.

- The price data will be pulled from Crypto Compare using the "crycompare" python library. This uses the Crypto Compare API to pull the most up-to-date coin information. The Google search trend data will be obtained using the "pytrends" python library. This uses some unofficial Google APIs to query the relative frequency, by week, of specific search terms.

- Once all the data is in memory and cleaned, trends and correlations are explored in further detail.

- Finally, machine learning techniques are applied to attempt to predict the price.

01 DATA WRANGLING

PRICE DATA

- Price data for both bitcoin and litecoin was gathered using the crycompare Python library. The crycompare backend is utilizing the Crypto Compare API (https://www.cryptocompare.com/api/) for price history.

- Once obtained the data was placed in a pandas dataframe for storage. This can be seen on the right.

```
In [7]: #
        # Print the dataframe info.
        #
        df_coins.info()

        <class 'pandas.core.frame.DataFrame'>
        DatetimeIndex: 2682 entries, 2010-07-17 to 2017-11-18
        Freq: D
        Data columns (total 12 columns):
        btc_close       2682 non-null float64
        btc_high        2682 non-null float64
        btc_low         2682 non-null float64
        btc_open        2682 non-null float64
        btc_volfrom     2682 non-null float64
        btc_volto       2682 non-null float64
        ltc_close       1487 non-null float64
        ltc_high        1487 non-null float64
        ltc_low         1487 non-null float64
        ltc_open        1487 non-null float64
        ltc_volfrom     1487 non-null float64
        ltc_volto       1487 non-null float64
        dtypes: float64(12)
        memory usage: 272.4 KB
```

01

DATA WRANGLING

PRICE DATA - CONT.

- To the right, the relative price data can be seen for both bitcoin and litecoin.
- At first glance the two time series look very similar.

01 DATA WRANGLING

GOOGLE SEARCH TREND

- The Google search trend data will be obtained using the "pytrends" python library. This uses some unofficial Google APIs to query the relative frequency, by week, of specific search terms (https://trends.google.com/trends/)

- Once obtained the data was placed in the same data frame as the price data. This can be seen on the right.

```
In [59]: #
         # Print the head of the keyword columns.
         #
         df_coins[['btc_kwrd', 'ltc_kwrd', 'trump_kwrd']].head()
```

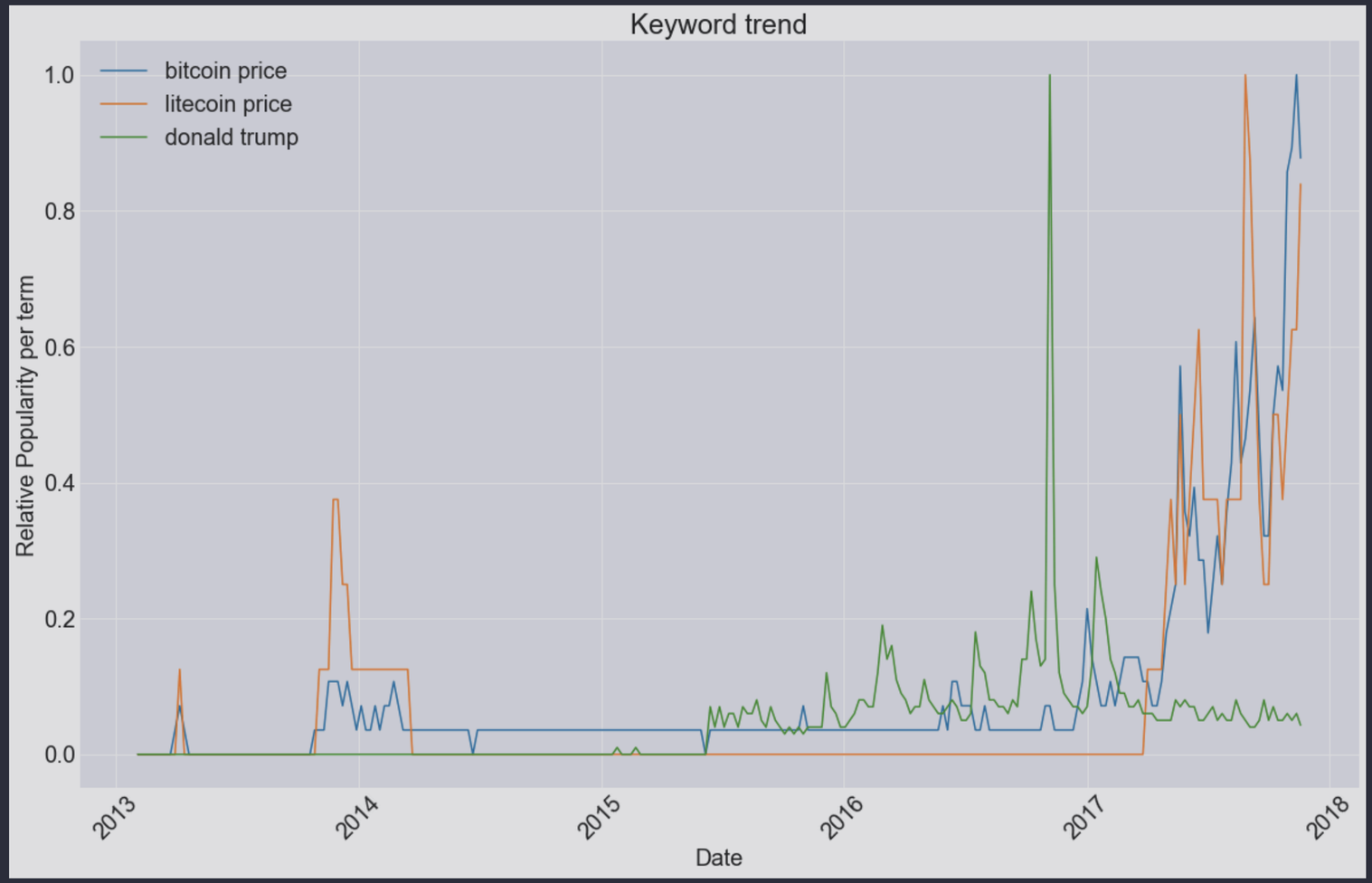
Out[59]:

time	btc_kwrd	ltc_kwrd	trump_kwrd
2013-10-24	0.571429	0.000000	0.0
2013-10-25	0.714286	0.000000	0.0
2013-10-26	0.857143	0.000000	0.0
2013-10-27	1.000000	0.000000	0.0
2013-10-28	1.000000	0.142857	0.0

01 DATA WRANGLING

GOOGLE SEARCH TREND - CONT.

- To the right, the relative trend data can be seen for “bitcoin price”, “litecoin price” as well as “donald trump”
- It becomes very apparent that both the bitcoin and litecoin trends are similar while the trump search appears to have little to no correlation with the other two search trends.



01 DATA WRANGLING

CLEANING

- The data from Google is by week. However, the price data is by day.
- To remedy this problem the Google trend data will be interpolated between weeks to obtain a daily estimate.
- The 'isPartial' feature is not needed so is removed from the dataframe to keep it uncluttered.¶

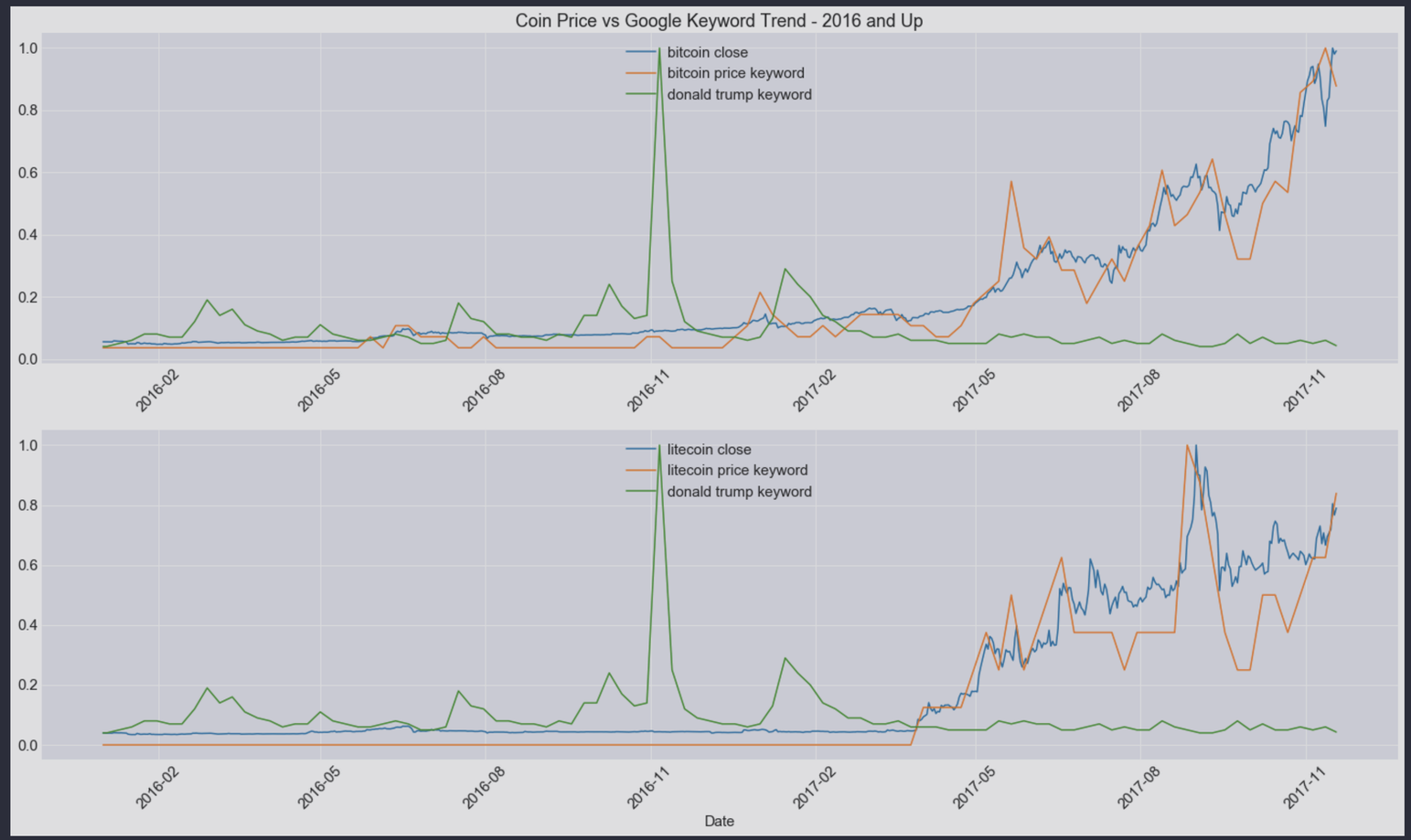
In [13]:
```
#
# Resample by day.
#
df_kwrd_bitcoin = df_kwrd_bitcoin.resample('D').interpolate(method='linear')
df_kwrd_litecoin = df_kwrd_litecoin.resample('D').interpolate(method='linear')
df_kwrd_trump = df_kwrd_trump.resample('D').interpolate(method='linear')

#
# Filter out anything newer than the cutoff day.
#
df_kwrd_bitcoin = pd.DataFrame(df_kwrd_bitcoin[:cutoff_day])
df_kwrd_litecoin = pd.DataFrame(df_kwrd_litecoin[:cutoff_day])
df_kwrd_trump = pd.DataFrame(df_kwrd_trump[:cutoff_day])
```

In [14]:
```
#
# Remove the 'isPartial' column.
#
del df_kwrd_bitcoin['isPartial']
del df_kwrd_litecoin['isPartial']
del df_kwrd_trump['isPartial']
```

01 DATA WRANGLING

TIME RANGE

- The python libraries will query the most up to date data as available. Due to this a maximum date is chosen to be 11/18/2017.
- This allows repeatability when running the code.
- As seen from the plot on the previous slide there are distinct “sections” in the data.
- After some exploratory analysis, the minimum year is set to 2017. The updated data sets can been seen on the right.



01

DATA WRANGLING

SUMMARY

- Data relatively easy to wrangle thanks to some python libraries.
- Minimal cleaning and manipulation necessary to clean the data.
- Initial visualizations look promising.

02

INFERENTIAL STATISTICS

OVERVIEW

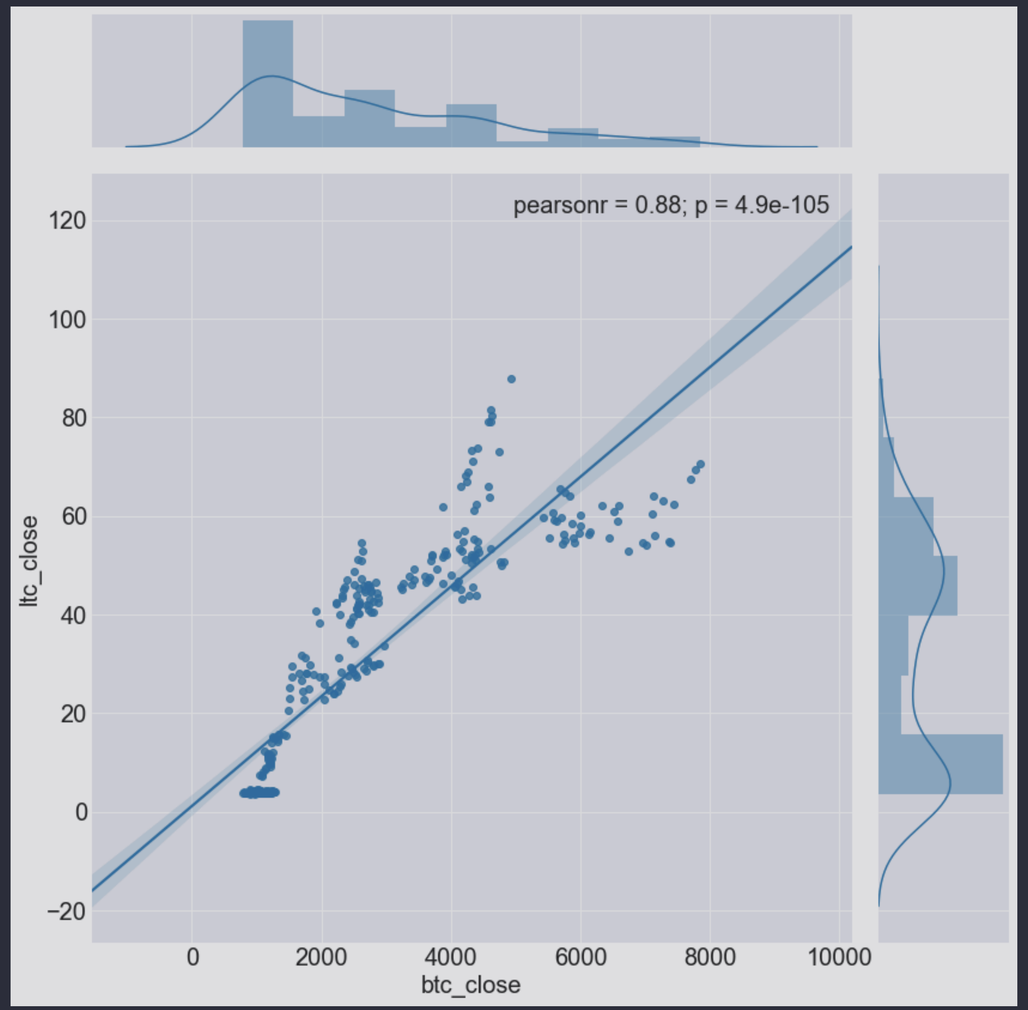
- Correlation will be explored.
- Data is shifted to understand how correlation shifts with the shifted data.
- Data is broken up by quarter and explored.

02

INFERENTIAL STATISTICS

BITCOIN CORRELATION WITH LITECOIN

- The correlation between the bitcoin and litecoin datasets is explored and found to be highly correlated with a Pearson correlation coefficient of 0.88.



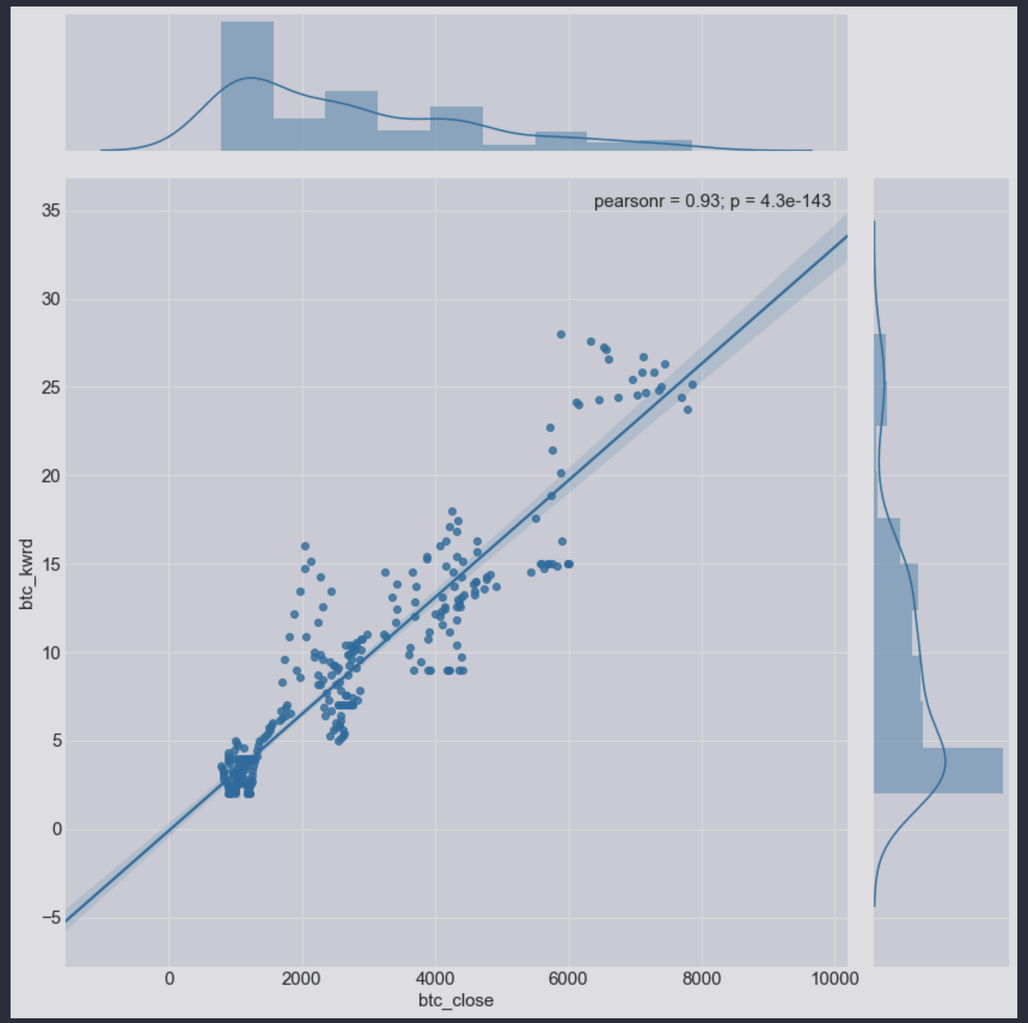
02

INFERENTIAL STATISTICS

BITCOIN CORRELATION WITH GOOGLE SEARCH

- The correlation between the bitcoin and the google search trend for “bitcoin price” is explored and found to be highly correlated with a Pearson correlation coefficient of 0.93.

- This is great news as this may be able to be used for prediction.



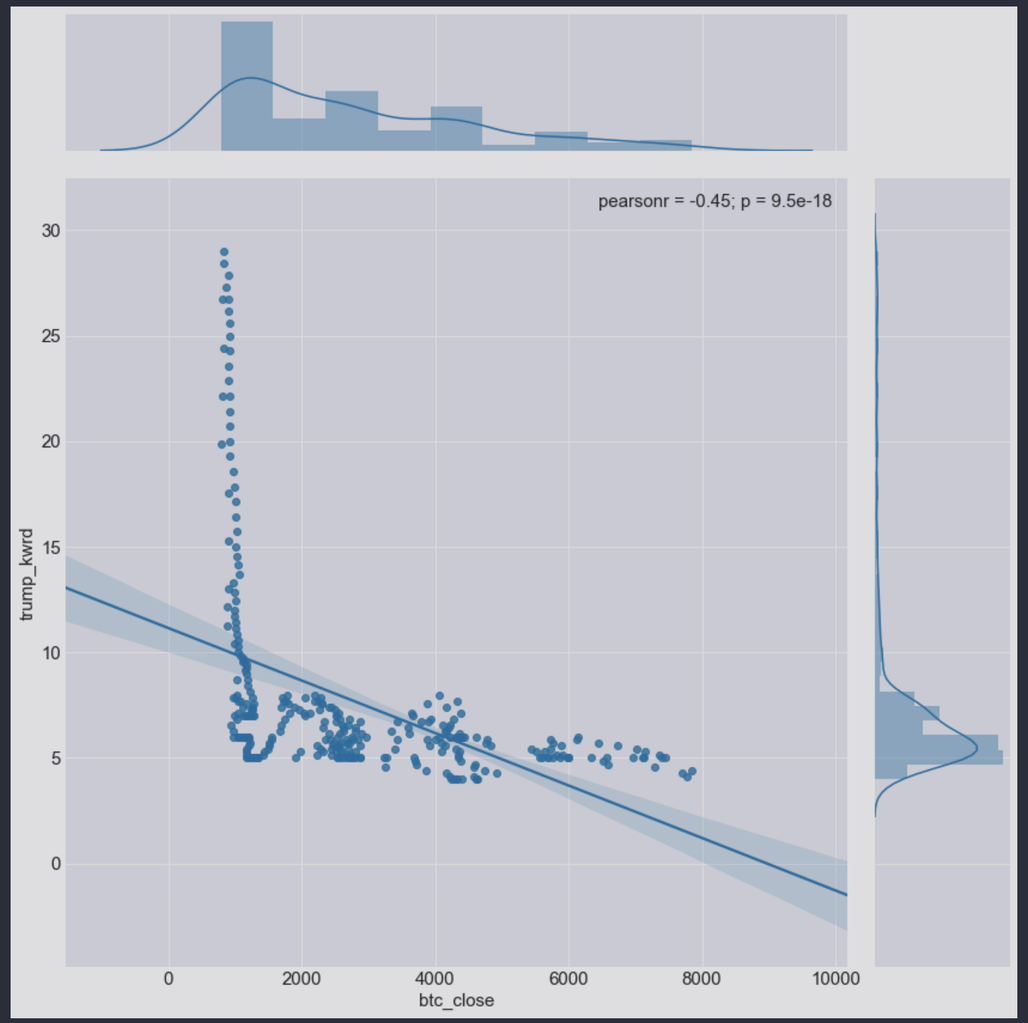
02

INFERENTIAL STATISTICS

BITCOIN CORRELATION WITH TRUMP SEARCH

- The correlation between the bitcoin and the google search trend for “donald trump” is explored and found to be very poorly correlated with a Pearson correlation coefficient of -0.45.

- This is also great news as it shows there is some validity to using Google keyword search trends.



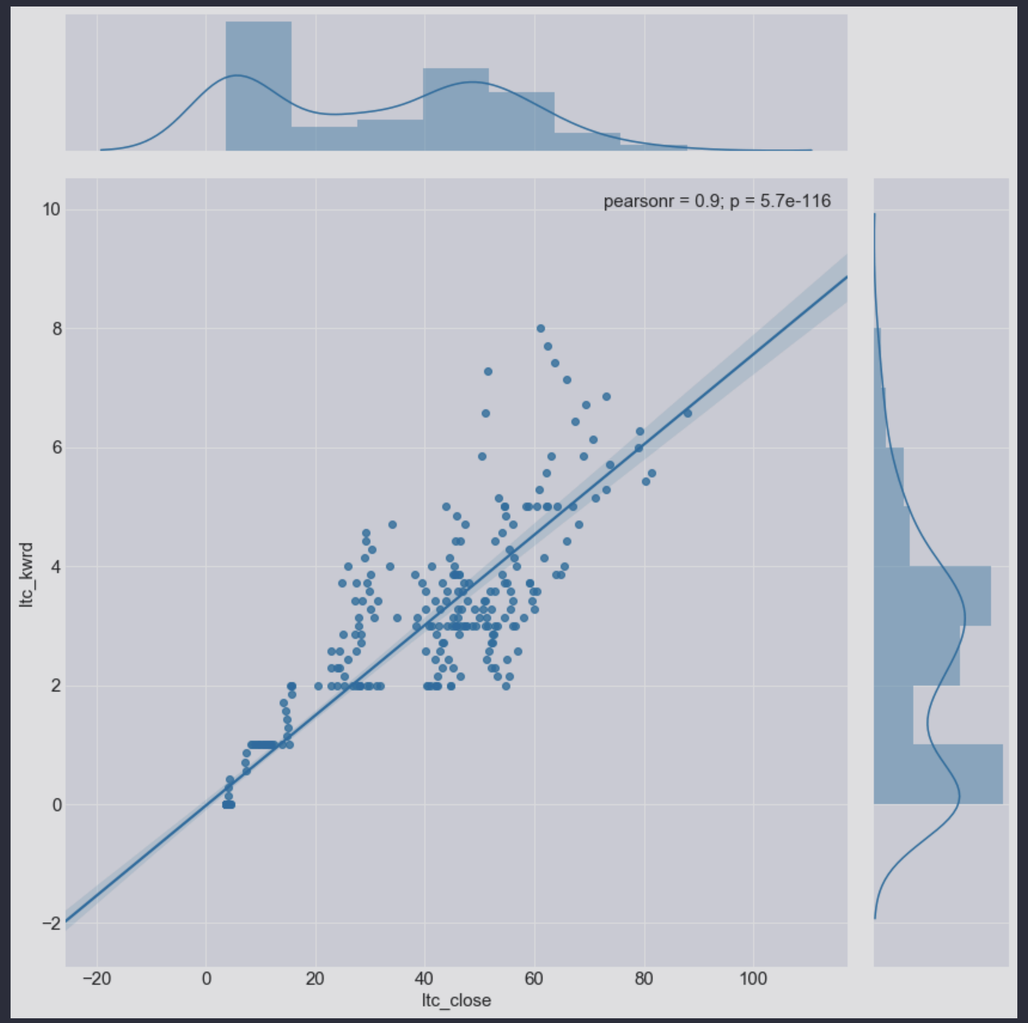
02

INFERENTIAL STATISTICS

LITECOIN CORRELATION WITH GOOGLE SEARCH

- The correlation between the litecoin and the google search trend for “litecoin price” is explored and found to be highly correlated with a Pearson correlation coefficient of 0.9.

- This is again great news as this may be able to be used for prediction.

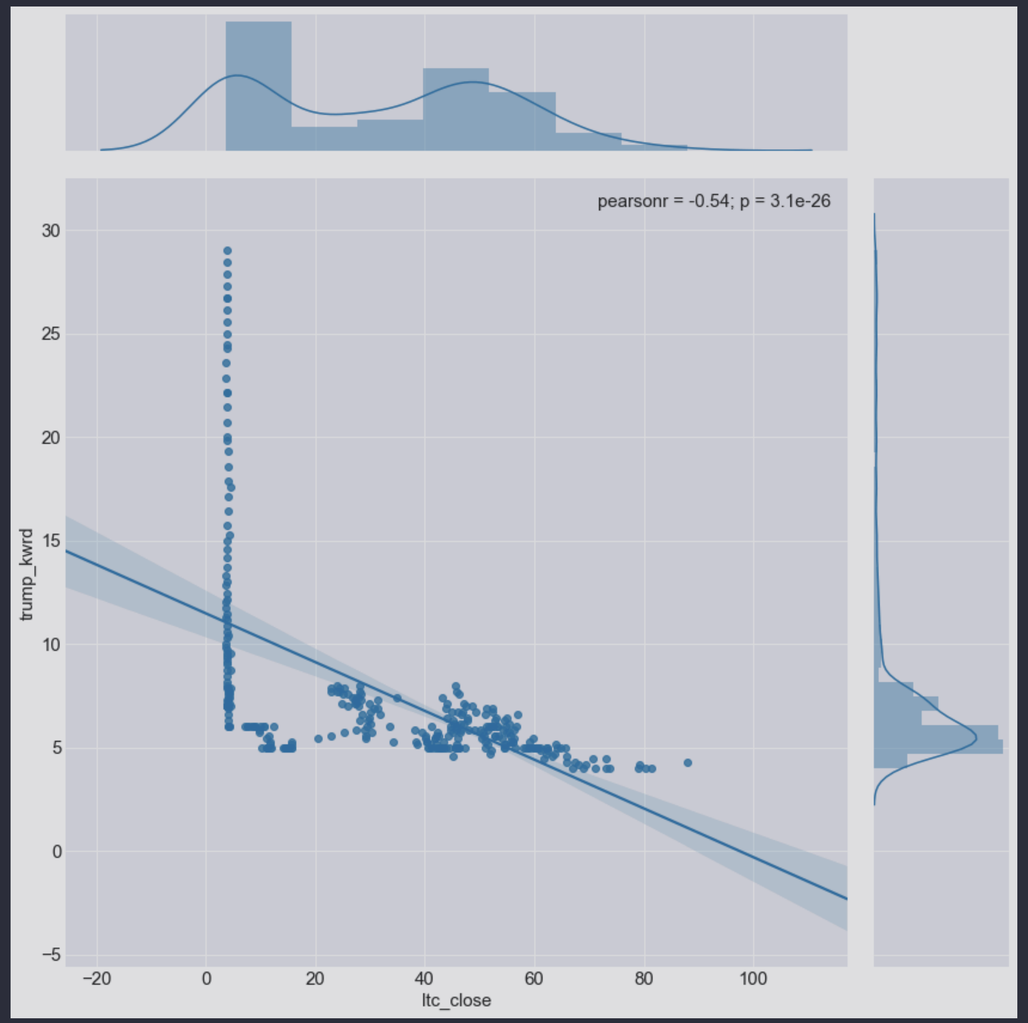


02

INFERENTIAL STATISTICS

LITECOIN CORRELATION WITH TRUMP SEARCH

- The correlation between the litecoin and the google search trend for “donald trump” is explored and found to be very poorly correlated with a Pearson correlation coefficient of -0.54.

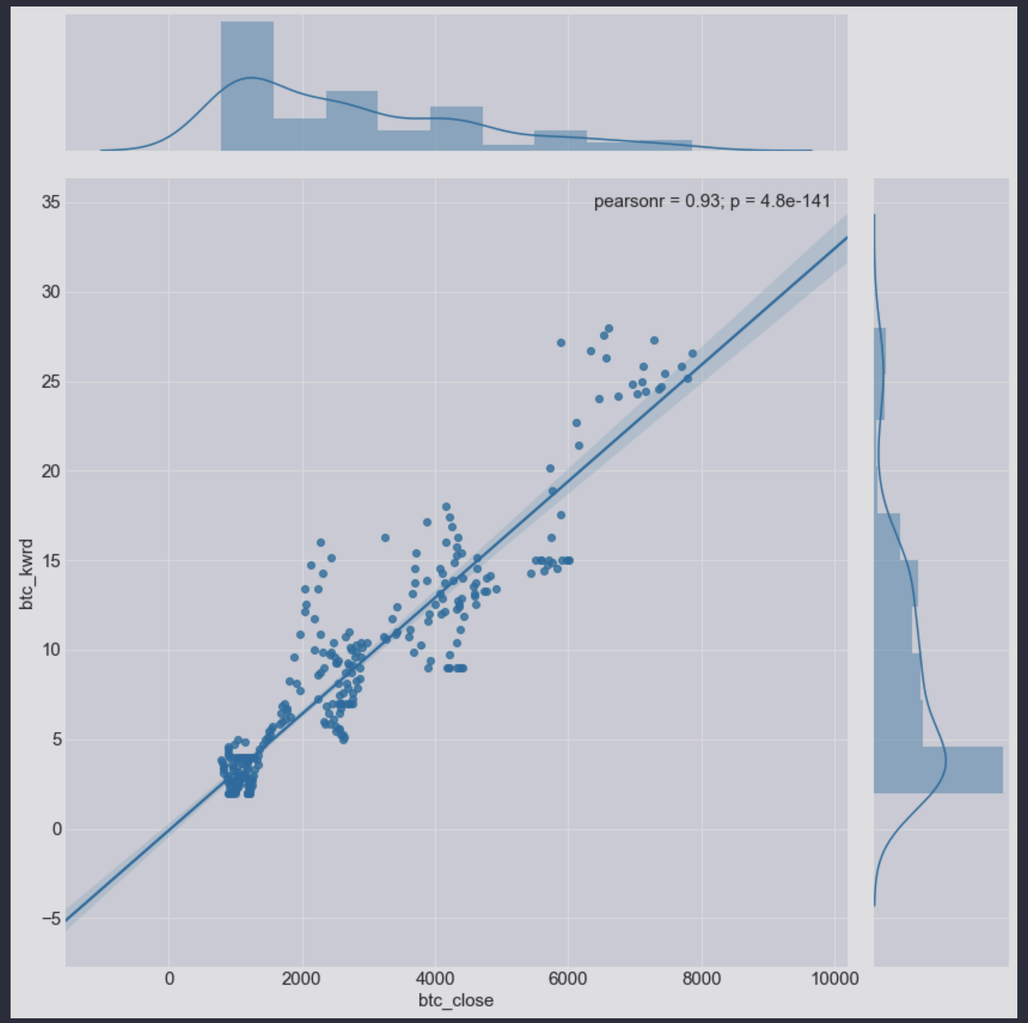


02

INFERENTIAL STATISTICS

BITCOIN CORRELATION WITH SHIFTED GOOGLE

- The google search trend data for “bitcoin price” is shifted by one day in an effort to understand if its worth exploring using yesterdays google search trend to predict the closing price for bitcoin today.
- The correlation between the bitcoin and the shifted google search trend for “bitcoin price” is very high at 0.93.
- In other words, 86.49% of the variance can be explained from the Google search trend!

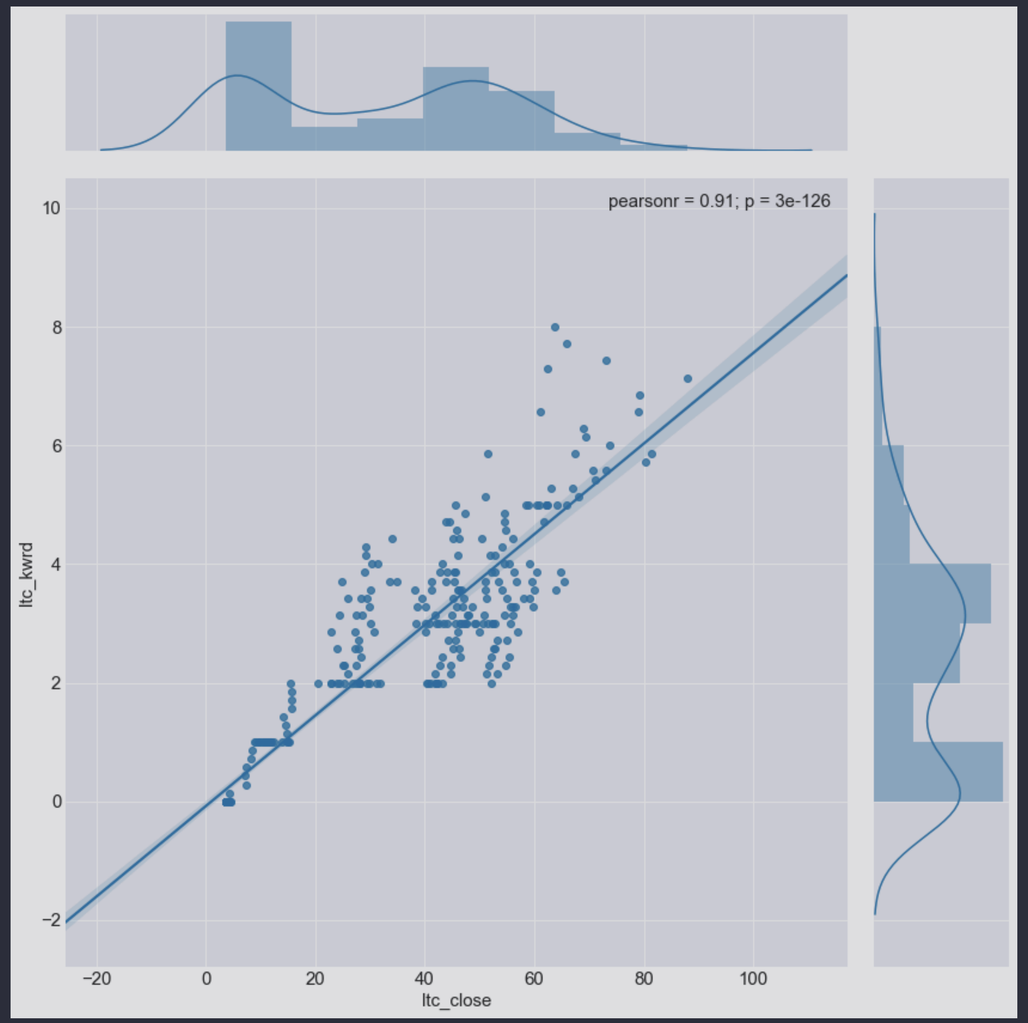


02

INFERENTIAL STATISTICS

LITECOIN CORRELATION WITH SHIFTED GOOGLE

- As before, google search trend data for “litecoin price” is shifted by one day in an effort to understand if its worth exploring using yesterdays google search trend to predict the closing price for litecoin today.
- The correlation between the litecoin and the shifted google search trend for “litecoin price” is very high at 0.91.
- In other words, 82.81% of the variance can be explained from the Google search trend!



02

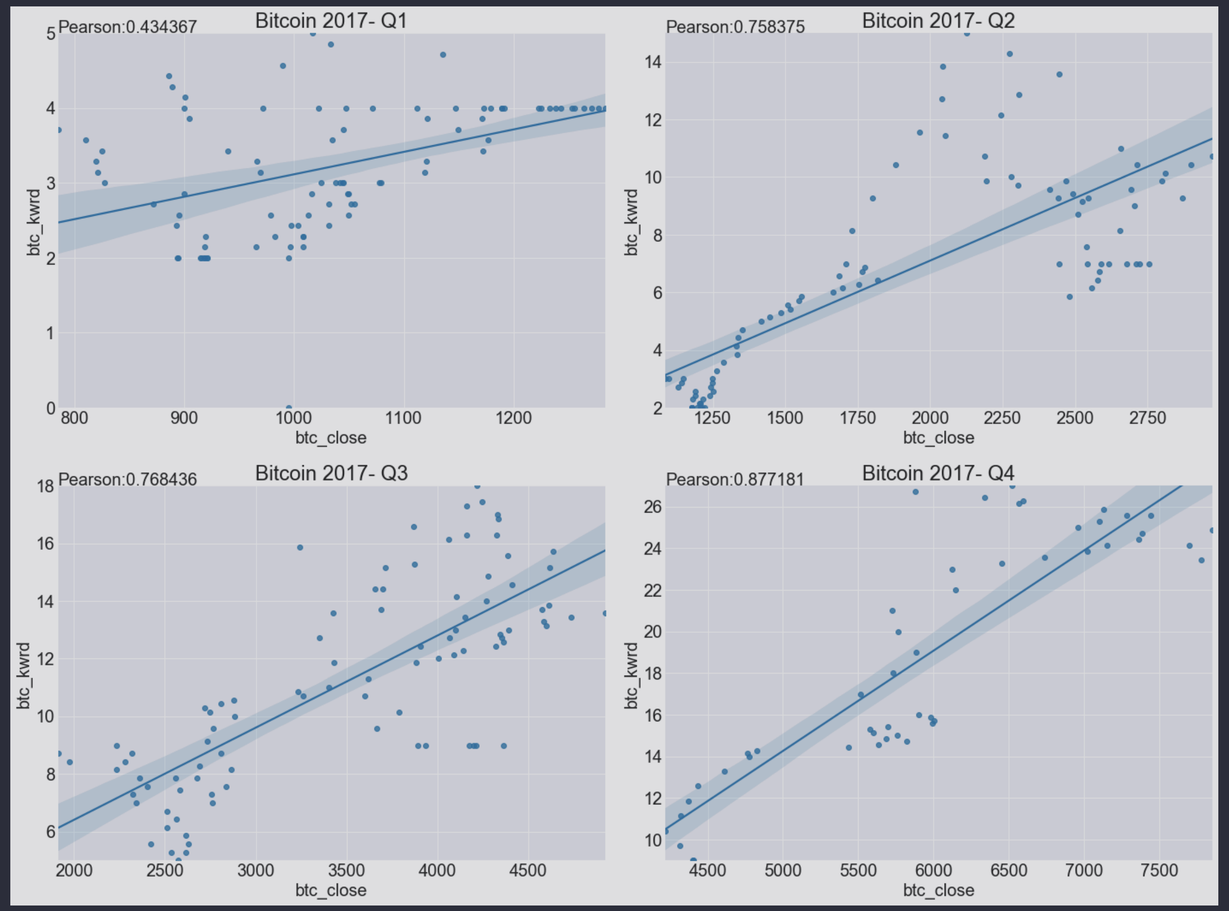
INFERENTIAL STATISTICS

BITCOIN BY QUARTER

- To understand how the keyword correlates throughout the year, the dataset is broken down by quarter and the correlations are computed quarterly.
- The results show that the correlation varies greatly by quarter (0.43 - 0.87)
- The plotted results can be seen on the next slide.

02

INFERENTIAL STATISTICS



02

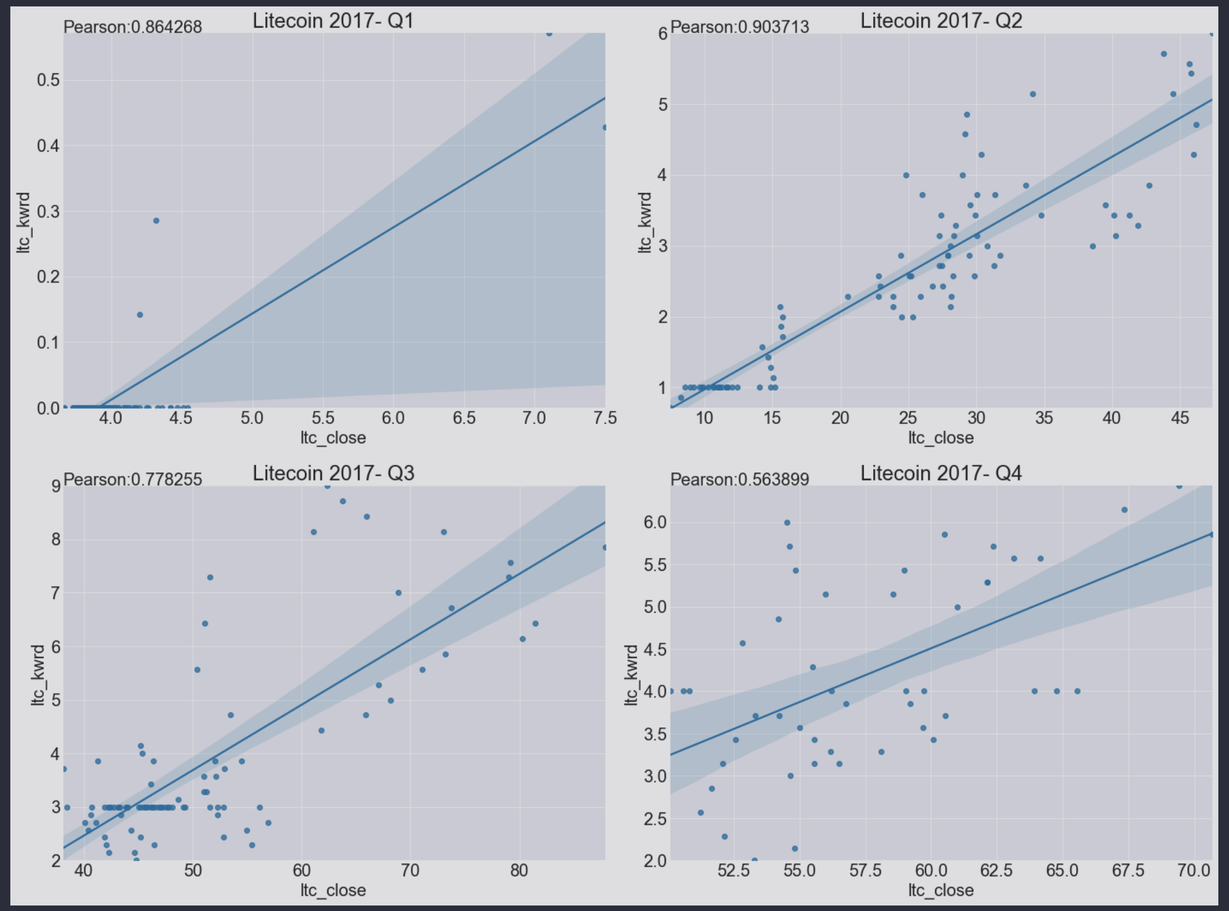
INFERENTIAL STATISTICS

LITECOIN BY QUARTER

- Again, to understand how the keyword correlates throughout the year, the dataset is broken down by quarter and the correlations are computed quarterly.
- The results show that the correlation varies greatly by quarter (0.56 - 0.90).
- There is a range in the confidence interval in Q1. This is likely due to the ‘flatlined’ Google search results during that time period.
- The plotted results can be seen on the next slide.

02

INFERENTIAL STATISTICS



02

INFERENTIAL STATISTICS

SUMMARY

- Correlations between closing price and Google trend for both coins look to have promise.
- No strong correlation exists between the Google search trend for “donald trump” and either coin price.
- Correlation varies widely by quarter between coin price and Google search trend.

03

MACHINE LEARNING

OVERVIEW

- Stationarity is explored.
- An ARMA model is used for litecoin prediction.
- Google trend and bitcoin price are used in the model in an attempt reduce error.

03
MACHINE LEARNING

PREDICTION

- Now that the data and its trends are understood in greater detail, a logical next step would be to attempt to predict the closing price or at least the direction of the coin.
- To achieve this an Autoregressive–moving-average model (ARMA). This is chosen over the autoregressive integrated moving average model (ARIMA) due to the seasonality component ("I") that will not be explored here.
- Next stationarity will be discussed.

03
MACHINE LEARNING

STATIONARITY

- The two most common ways to make a non-stationary time series curve stationary are differencing and transforming. Log transform is probably the most commonly used transformation, if you are seeing a diverging time series. Therefore, a log transformation will be used.

- The coin price data is not very interesting before May of 2017 so in this portion of the exercise, any data before 05/2017 is filtered out.

- To the right is the result of the log transform on both the bitcoin and litecoin closing price values.

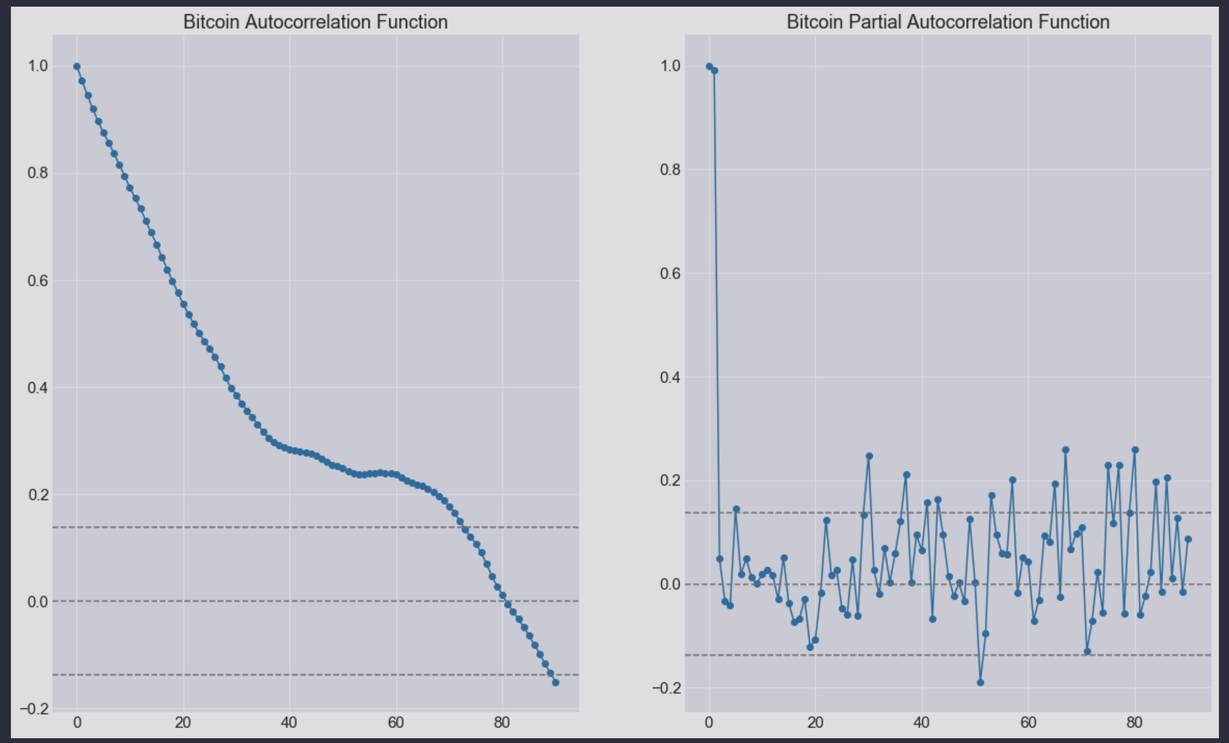
03
MACHINE LEARNING

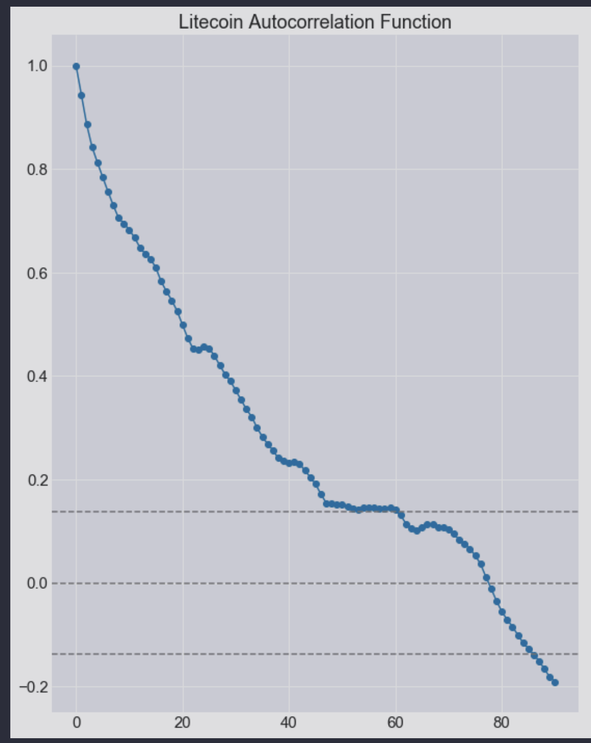
AUTOCORRELATION FUNCTION(ACF) AND PARTIAL AUTOCORRELATION FUNCTION(PACF)

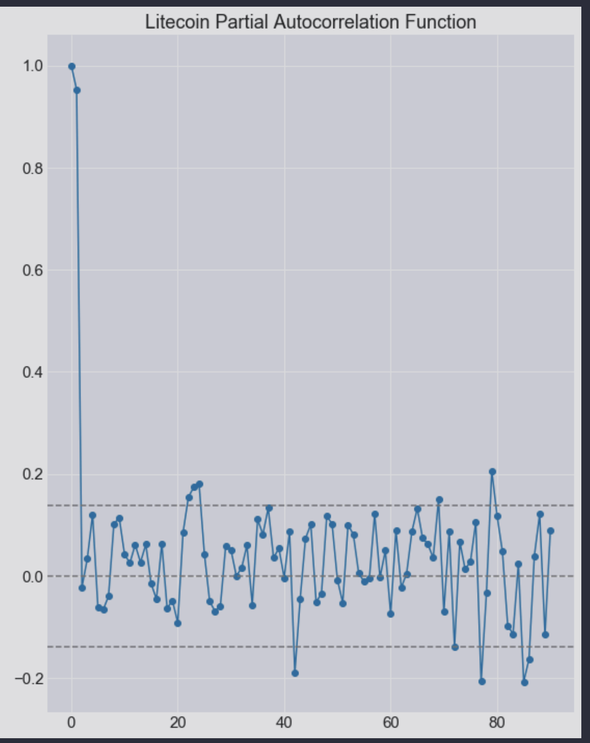
- The autocorrelation function (ACF) is the correlation of a time series signal with a delayed copy of itself as a function of delay.
- The partial autocorrelation function (PACF) gives the correlation of a time series with its own lagged values, controlling for the values of the time series at all shorter lags.
- The bitcoin and litecoin ACF and PACF can be seen on the next slide.

03
MACHINE LEARNING

- Notice that both coins have a very high correlation with the lag 1 point meaning that todays closing price has a very high correlation with yesterdays closing price. Also, the ACF gradually decays. Both of these indicators suggest that a large portion of the variance can be explained using only the “AR” piece of the ARIMA(ARMA) model.







03
MACHINE LEARNING

AUGMENTED DICKEY-FULLER TEST

- The Augmented Dickey-Fuller test is a simple test to run for checking stationarity.
- After performing this test on bitcoin and litecoin, bitcoin is found to not be stationary with a p-value of 0.761847.
- However, litecoin is found to be stationary with a p-value of 0.038118.
- While the bitcoin dataset does not appear to be stationary (even after breaking it up) with a p-value of 0.761847, the litecoin dataset does appear to be stationary with a p-value of 0.038118. We could perform a single order difference on bitcoin to move towards stationarity but for now we have at least one dataset that passes the Dickey-Fuller Test.

03
MACHINE LEARNING

BASIC MODEL - TRAINING

- Let's now create the ARMA model for litecoin using the Python library statsmodels.

- The dataset is broken up into train and test sets. The train set is every datapoint from 5/2017 to 11/14/2017. The test set consists of only 3 data points from 11/15/2017 to 11/18/2017.

- To the right are the model results:
  - The AR component is chosen to be 1 used on the PACF. This has a significant contribution to the model.

  - There is no “I” term due to a lack of observable seasonality.

  - The “MA” term was chosen to be 3 though from the model its clear that they do not add much.

ARMA Model Results

Dep. Variable:	ltc_close	No. Observations:	199
Model:	ARMA(1, 3)	Log Likelihood	232.463
Method:	css-mle	S.D. of innovations	0.075
Date:	Wed, 31 Jan 2018	AIC	-452.926
Time:	23:11:38	BIC	-433.166
Sample:	05-01-2017 - 11-15-2017	HQIC	-444.929

	coef	std err	z	P>|z|	[0.025	0.975]
const	3.6282	0.364	9.960	0.000	2.914	4.342
ar.L1.ltc_close	0.9904	0.011	94.063	0.000	0.970	1.011
ma.L1.ltc_close	0.0489	0.072	0.680	0.497	-0.092	0.190
ma.L2.ltc_close	-0.0176	0.072	-0.245	0.807	-0.158	0.123
ma.L3.ltc_close	-0.1149	0.070	-1.642	0.102	-0.252	0.022

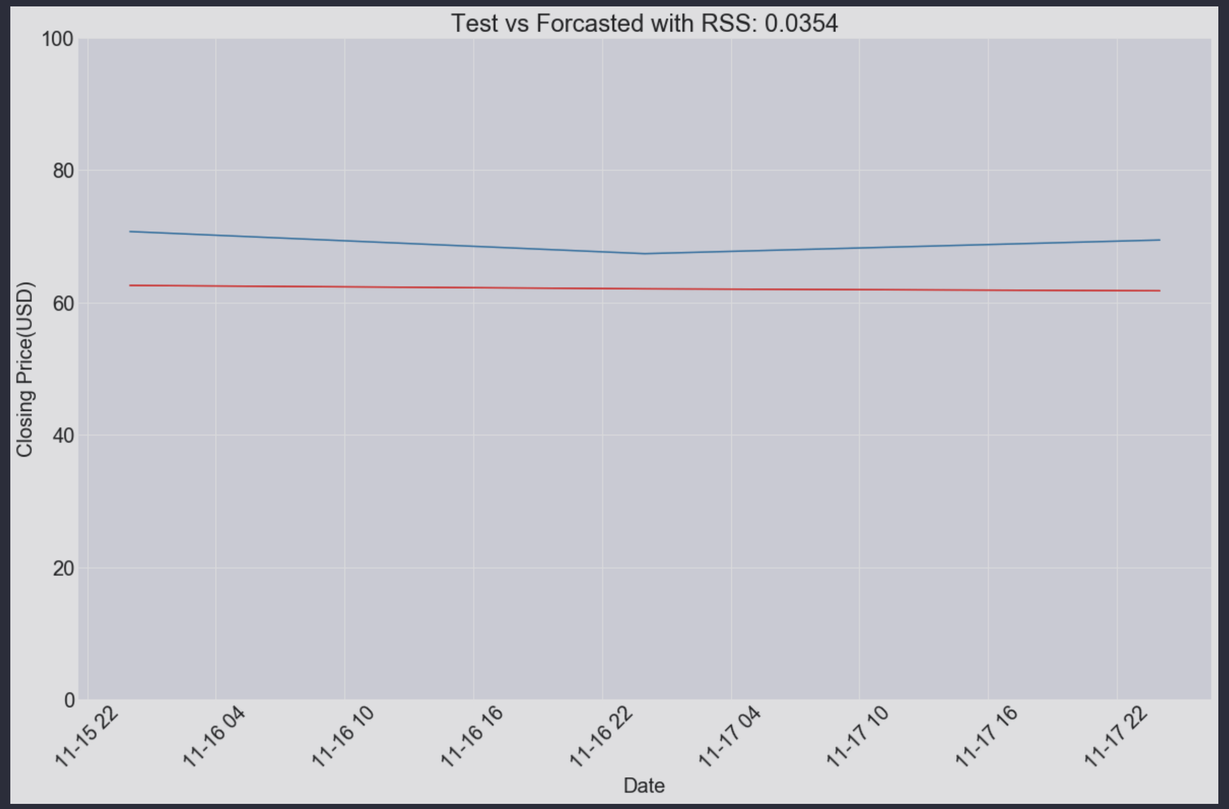
Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0097	+0.0000j	1.0097	0.0000
MA.1	-1.1138	-1.7191j	2.0484	-0.3415
MA.2	-1.1138	+1.7191j	2.0484	0.3415
MA.3	2.0745	-0.0000j	2.0745	-0.0000

03
MACHINE LEARNING

BASIC MODEL - PREDICTION

- The next step is to use the model for litecoin price prediction.
- To the right are the model results of a 3 point forecast using the using the forecast method on the model.
- The model performed fairly well with a residual sum of squares(RSS) of 0.0354.
- Can this error be improved by including either the Google search trend or the other coin?



03
MACHINE LEARNING

BASIC MODEL WITH GOOGLE TREND - TRAINING

- Let's now create the ARMA model for litecoin while also including the Google search trend for the keyword “litecoin price” shifted by one day. This is done so that we are using yesterdays search trend to aid in the prediction of todays litecoin closing price.
- To the right are the model results of this model which was built using the same “AR”, “I” and “MA” coefficients.
- So how does it perform?

ARMA Model Results

Dep. Variable:	ltc_close	No. Observations:	199
Model:	ARMA(1, 3)	Log Likelihood	243.377
Method:	css-mle	S.D. of innovations	0.071
Date:	Wed, 31 Jan 2018	AIC	-472.753
Time:	23:11:39	BIC	-449.700
Sample:	05-01-2017 - 11-15-2017	HQIC	-463.423

	coef	std err	z	P>|z|	[0.025	0.975]
const	3.2136	0.287	11.202	0.000	2.651	3.776
ltc_kwrd	0.3498	0.067	5.188	0.000	0.218	0.482
ar.L1.ltc_close	0.9912	0.010	100.914	0.000	0.972	1.010
ma.L1.ltc_close	-0.0264	0.072	-0.365	0.716	-0.168	0.115
ma.L2.ltc_close	-0.1020	0.077	-1.319	0.189	-0.254	0.050
ma.L3.ltc_close	-0.1822	0.075	-2.422	0.016	-0.330	-0.035

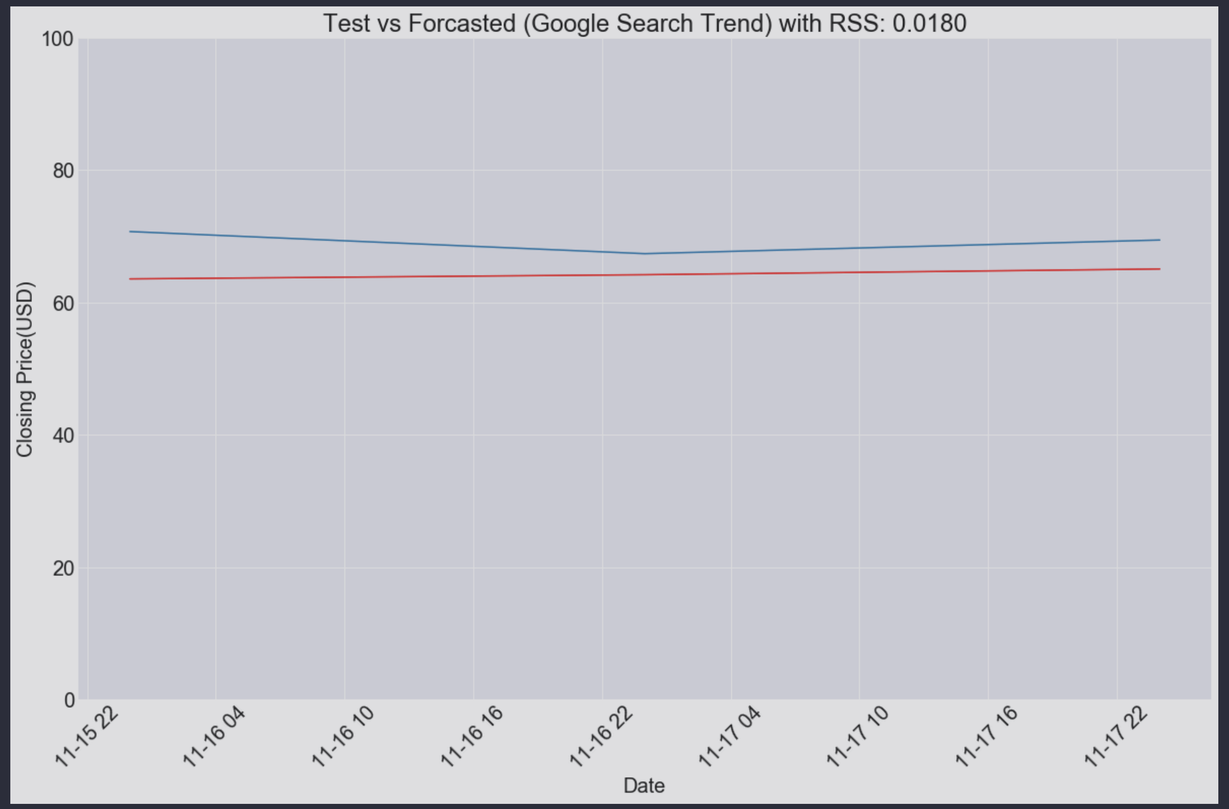
Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0089	+0.0000j	1.0089	0.0000
MA.1	1.5712	-0.0000j	1.5712	-0.0000
MA.2	-1.0656	-1.5355j	1.8690	-0.3466
MA.3	-1.0656	+1.5355j	1.8690	0.3466

03
MACHINE LEARNING

BASIC MODEL WITH GOOGLE TREND - PREDICTION

- To the right are the model results of a 3 point forecast using the using the forecast method on the model.
- The model did very well with a RSS of 0.0180.
- By including the Google search trend the forecasting was improved by 50%!
- Can we get even better though? What happens if we use the bitcoin closing price data instead of the Google search trend data in the model?



03
MACHINE LEARNING

BASIC MODEL WITH BTC CLOSING - TRAINING

- Lastly create the ARMA model for litecoin while also including the bitcoin closing price shifted by one day. As with the Google search trend, this is done so that we are using yesterdays closing price to aid in the prediction of todays litecoin closing price.
- To the right are the model results of this model which was built using the same “AR”, “I” and “MA” coefficients.
- So how does it perform?

ARMA Model Results

Dep. Variable:	ltc_close	No. Observations:	199
Model:	ARMA(1, 3)	Log Likelihood	232.538
Method:	css-mle	S.D. of innovations	0.074
Date:	Thu, 01 Feb 2018	AIC	-451.075
Time:	22:32:10	BIC	-428.022
Sample:	05-01-2017 - 11-15-2017	HQIC	-441.745

	coef	std err	z	P>|z|	[0.025	0.975]
const	4.0156	1.076	3.732	0.000	1.907	6.125
btc_close	-0.0492	0.127	-0.388	0.698	-0.298	0.199
ar.L1.ltc_close	0.9914	0.010	99.472	0.000	0.972	1.011
ma.L1.ltc_close	0.0658	0.084	0.784	0.434	-0.099	0.230
ma.L2.ltc_close	-0.0180	0.072	-0.250	0.803	-0.159	0.123
ma.L3.ltc_close	-0.1150	0.069	-1.660	0.099	-0.251	0.021

Roots

	Real	Imaginary	Modulus	Frequency
AR.1	1.0087	+0.0000j	1.0087	0.0000
MA.1	-1.1263	-1.6972j	2.0369	-0.3432
MA.2	-1.1263	+1.6972j	2.0369	0.3432
MA.3	2.0958	-0.0000j	2.0958	-0.0000

03
MACHINE LEARNING

BASIC MODEL WITH BTC CLOSING - PREDICTION

- To the right are the model results of a 3 point forecast using the using the forecast method on the model.
- The model did very well with a RSS of 0.0397.
- Unfortunately the model preformed worse than the basic model by about 11%.

03

MACHINE LEARNING

SUMMARY

- Both coins (litecoin and bitcoin) have a high correlation with their respective Google search trend of "litecoin price" and "bitcoin price" respectively. This remains the case even when shifted by a day.
- Litecoin passes the augmented Dickey-Fuller test and is therefore the coin chosen for building an ARMA model with.
- The ARMA model utilizing the closing price of litcoin along with the shifted Google search trend for "litecoin price" performed the best at a 3 day closing price forecast. This was based on the RSS between the actual and forecasted values.

04

FUTURE WORK

- The model should be back tested to see how it handles predicting the price of the past data.
- Other features/predictors can be explored and brought into the model in an attempt to improve the RSS.
- The dataset includes opening price as well which can be merged with the closing price for a higher resolution on the price movement.
- A recurrent neural network (RNN) such as a long short term memory network(LSTM) could be trained and tested.
- A simulation could be ran on future data where by each day the model is re-trained with the current days closing price and then used to predict tomorrows closing price.