KARAN GODA

Project Stage 2

Data Analysis

Prepared by: **Karan Goda**, Undergraduate Bachelors in Computer science and technology

SID: 460496371

Part 1

Introduction

My interest in cryptocurrency

Cryptocurrencies were a relatively new field when I discovered it on my mobile because of its recent 1000% rise in a couple of weeks. There was a lot of mixed sentiment surrounding bitcoin because the **vast majority were against Bitcoin claiming it to be a "Ponzi scam"** or more commonly as a "market bubble". However, those that were loyal to Bitcoin defended it to the very core including the Bitcoin foundation going so far to claim Satoshi Nakamoto as their saviour and modern day Jesus Christ. My views were somewhere in the middle because I did not fully believe in the Bitcoin hype, however I was not against the idea of a decentralized currency either. This allowed me to explore Bitcoin more as an investment, instead of the code behind Bitcoin because **I was very passionate about economics back in high school (and still am).** I was only 16 when I discovered Bitcoin. Since it was an unregulated market I could move in and out small funds that were liquid enough to not get squeezed by the market. **Most of my analysis during this time was based on greed and dreams. I was not very data savvy.** It had cost me heavily financially, academically and in terms of health. However, just as a chart bounces back up so did I, and here I am conducting analysis on my dataset of cryptocurrencies as a research topic.

I have done qualitative analysis such as technical analysis of the structure of the chart of the cryptocurrencies, however the tools Python, SQL and Unix allowed me to delve deeper in my analysis and do quantitative analysis as I was able to do use some mathematical functions as well as conduct some aggregation analysis on my data. This helped me know the total volume of each of the different price points in my data which allowed me to know where the most money flowed inside, and outside of Bitcoin.

The origin of the dataset I had used to create my final dataset

I had **used two datasets** to do a more complex and meaningful analysis as it allowed me to display objectivity and allowed me to do more complex analysis. The data sources are:

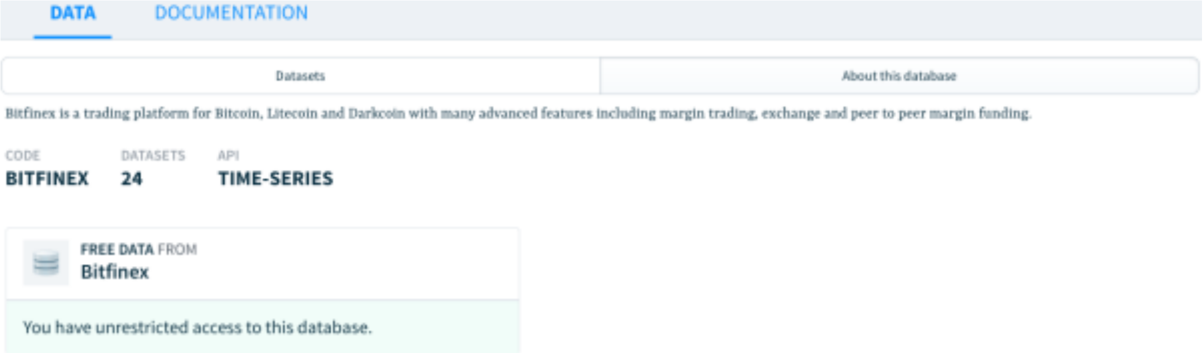
- http://bitcoincharts.com/charts/btcmarketsAUD
- https://www.quandl.com/collections/markets/bitcoin-data

Bitcoin charts has been a very amazing tool to use to obtain the BTCAUD rates because the data source had given me the power on what dates I can choose as well as the currency involved. Therefore, I chose the Australian dollar vs Bitcoin pair because it would make more sense as Australia conducts its dealings in the Australian dollar, and not the US Dollar. However the lack of cryptocurrency pairs was this website's limitation as I wanted my dataset to be a theme on pairs involving BTC and LTC.

The other data source **Quandl** I had used was more limited in terms of Bitcoin pairs because it did not have the specific pairing (BTCAUD) I had chosen. However, Quandl was a collection of knowledge when it came to providing altcoin pairs in the cryptocurrency environment, especially the more important ones which are ETHUSD, ETHBTC, ETCBTC, LTCBTC, XMRBTC, XMRUSD, ZECBTC, DSHUSD, and DSHBTC. Out of these, I had chosen only LTC because the other currencies are newer in general and I was basing my research on the years 2014-2016 where altcoins like ZECBTC did not exist for example, or there may not be enough data to do reliable analysis on it. LTCBTC provided the only viable option.

Permissions to use this data

Both of these websites **have given me appropriate rights to use the data** as mentioned in their about pages.



Bitfinex is a trading platform for Bitcoin, Litecoin and Darkcoin with many advanced features including margin trading, exchange and peer to peer margin funding.

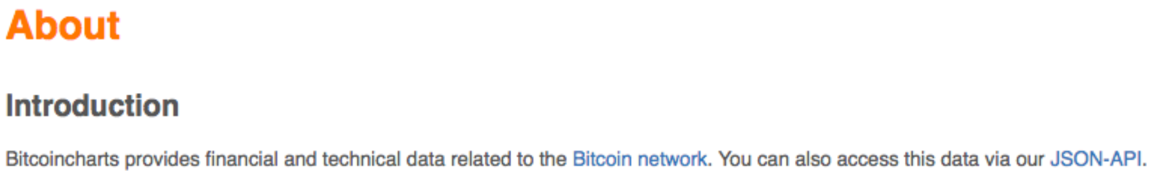
CODE BITFINEX

DATASETS 24

API TIME-SERIES

FREE DATA FROM Bitfinex

You have unrestricted access to this database.



About

Introduction

Bitcoincharts provides financial and technical data related to the Bitcoin network. You can also access this data via our JSON-API.

The **header** of the **final merged dataset**

The columns are:

1. **Date:** The date is the common element between the cryptocurrency pairs LTCBTC, and BTCAUD. It is to track the prices of either currency at any given point in time between 2014 - 2016.
2. **LTCBTC:** LTCBTC shows the pair's prices on the 1D (daily) timeframe between 2014-2016 in order to provide more accuracy in finding a day with a specific price point for example.
3. **Open:** The opening price for BTCAUD on the market
4. **High:** The high price in the range for BTCAUD on the market
5. **Low:** The low price in the range for BTCAUD on the market
6. **Close:** The closing price of the daily candle for BTCAUD on the market
7. **vol_btc:** The volume of BTCAUD in which the quantity of dollar flow is measure in terms of BTC
8. **vol_aud:** The volume of BTCAUD in which the quantity of dollar flow is measure in terms of AUD
9. **weighted_price:** The weighted price of BTCAUD by averaging the high, low and close with

most weigh upon whichever had the highest volume (which would give it the highest importance).

Analysis overview

The analysis should tell you a lot of things which I, the author found useful. For example, how many rows there are in the dataset and their means, the total number of columns, as well as the main data analysis conducted. **Limited images will be uploaded here**, however **links to the graphical Jupyter notebook and Excel are provided below which do display the data in a interpretable form with a lot more images.**

My analysis has revealed to me that Bitcoin and Litecoin did not go up at the same time. They both shared a common trend of going up, however after LTC had spiked, there was a quick demise as displayed from the dataset which was graphed.

Since the dataset itself was just price points which could have gone up or down, there was no linear equation to fit the dataset. Ultimately the dataset used other calculations to find the significant information. These other calculations where for example the mean price of Bitcoin, or the total volume in Australian dollars at each important price of Bitcoin, etc. The Jupyter Notebook, and other tools go more in-depth of the analysis done.

The analysis that I have done using Unix, Python, SQL and Excel matters because it is not hypothetical speculation in a trading environment, but actual analysis done using actual data from outsource free information outlets. This raw data allowed me to do many interesting things with the graph because I had the freedom in what columns I selected, what specific ports of the data I analysed as well as the purpose on my analysis. Cryptocurrency is one of my favourite areas of the market, so this allowed me to work with even more motivation as what I would do would be meaningful to me as well as my peers and tutors alike.

Some short comings in the dataset I found was that some ‘date’ rows were missing. This meant that I would be missing the ‘volume’, ‘high’, ‘low’, ‘close’ and ‘LTCBTC’ values in my data. This meant that my data may not be able to display the full picture as some percentage of data was missing. However, this flaw in the dataset was overlooked because more than 90% of the data was properly captured and formatted for SQL, Python, Unix and Excel use.

This data is very reliable as it comes from the reputable sources **Quandl** and **Bitcoin charts.** The accuracy of this data has been verified by the author by validating the date compared to their respective exchange dates and checking that the prices were equal during these points.

The author hopes that the reader (you) find meaning in his work and analysis.

Analysis by the author:

1. Jupyter Notebook (Source is in main directory as well as a webpage):
   (Located in Edstem.com.au workspace of user ‘kgod6253’)
2. SQL (DML and DDL statements are in subfolder called ‘SQL’ in main directory)
3. Unix (Bash script is in is in subfolder called ‘Unix in main directory)
4. Excel (Excel file is in subfolder called ‘Excel’ in main directory)

Part 2

An IT approach to Data analysis

I believe that to fully look at data from all angles, different tools must be used accordingly. This is because no one tool can do all jobs in an easy manner; each tool has its specific job and that is exactly what it should be used for. Imagine using a spoon for digging or a chainsaw for cutting a sandwich. This is impractical, and very time consuming. A better practice would be to separate (partition) the files and folders into easy, accessible modules, and then operating the tools on the respective files.

Jupyter notebook

I used Python's library Jupyter notebook over Excel on most of the data visualisation and interpretation aspects because Jupyter notebook allows me to modify data to greater lengths and it is much easier getting the functionality I want over Excel in the aspects related to the data itself. Beautifying the data's graphs is a bigger challenge on Jupyter because of all the different functions you have to use from the Jupyter notebook library as well as some other custom libraries if your graphs have extra functionality.

My Jupyter notebook does analysis on:

Analysis by the author:

1. **Finding the total volume in each of the respective important price points.** For example, say I want to know the total volume of BTC where it was between $400-500, but for all price points. Using Unix commands; this would take me a while because I would have to repeat the process for each of the ranges. However in Jupyter notebook, I was able to do this in less than 10 lines.
2. **The line graph of LTCBTC.** This helps the viewer visualise what the data looks like.
3. **The top ten LTCBTC prices.** This helps the user of this notebook because the user can then locate the price points and observe in hindsight what exactly happened at those values.
4. **The candlestick graph of BTCAUD.** A financial dataset must have this functionality as candlesticks can play a major role throughout a trader's career.
5. **Other analysis**

Code

```
# Importing the libraries needed
import pandas as pd
from ipykernel import kernelapp as app
import os
import math
import matplotlib
import numpy as np
import datetime as dt
import matplotlib.pyplot as plt
import matplotlib.ticker as mticker
from matplotlib.finance import candlestick_ohlc
import matplotlib.dates as mdates
import seaborn as sns
%matplotlib inline

# I want to know the available fonts
print(plt.style.available)

# The file and path to locate the dataset
ltc_btc = ('mergeLBAUD.csv')
file = os.path.join(ltc_btc)

# The csv file is stored as a dataframe in pandas.
file_opened = pd.read_csv(ltc_btc)
dataframe = pd.DataFrame(file_opened)

# Rows, columns
dataframe.shape

# The column names of the file
dataframe.columns

# The mean values of each attribute in the dataset
dataframe.mean()

# The median (middle) values of each attribute in the dataset
dataframe.median()

# The maximum values of each attribute in the dataset
dataframe.max()

# The total volumns of BTC and AUD respectively
vol_col = dataframe.loc[:,['vol_btc','vol_aud']]
vol_col.sum()

# Displays the data where date is the index
dataframe.set_index('Date')[:3]
```

```
# Find the ten highest LTCBTC prices
# If ascending is false, that means that descending is true
dataframe.sort_values(by='LTCBTC',ascending=False)[:10]

# Removes the grid in all the graphs
sns.set_style("whitegrid", {'axes.grid' : False})

# Plot the ten highest LTCBTC prices
# If ascending is false, that means that descending is true
ltc_highs = dataframe.sort_values(by='weighted_price',ascending=False)[:10]
ltc_highs.set_index('Date').plot(y = 'LTCBTC', lw = 3, kind = 'line', style='b-',
figsize=(18,10))
plt.xlabel('Date', fontsize=25, fontname='DejaVu Sans')
plt.ylabel('Price', fontsize=25, fontname='DejaVu Sans')
plt.title('Ten highest LTC prices', fontsize=30, fontname='DejaVu Sans')
plt.setp(plt.gca().get_xticklabels(), rotation=45, horizontalalignment='right',
fontsize=18)
plt.setp(plt.gca().get_yticklabels(), fontsize=18)
plt.style.use('ggplot')

# Bar Graph for BTC prices vs AUD volume
# Groups specific columns of the data by the weighted_price
# Finds the sum of the BTC volume
# This allows us to know where BTC trades occurred the most.

columns = dataframe.loc[:,['vol_aud', 'weighted_price']]
columns = (columns / 100)
columns = columns.round(0) * 100
btc_occurrences = columns.groupby(['weighted_price']).sum()

# Point of control: aka most voluminous areas of the BTC prices
fig, ax = plt.subplots()
ax.plot(btc_occurrences)
ax.set_facecolor('white')
ax.ticklabel_format(useOffset=False, style='plain')
plt.xlabel('Weighted Price of BTC', fontsize=25, fontname='DejaVu Sans')
plt.ylabel('Volume in AUD', fontsize=25, fontname='DejaVu Sans')
plt.title('Total AUD volume at different BTC prices between 2014-2016',
fontsize=30, fontname='DejaVu Sans')
plt.setp(plt.gca().get_xticklabels(), rotation=45, horizontalalignment='right',
fontsize=18)
plt.setp(plt.gca().get_yticklabels(), fontsize=18)
```

```
btc_occurrences.plot(style='b', ax=ax, kind = 'bar', figsize=(18,10))
plt.style.use('ggplot')

# BTC graph
graph = dataframe.set_index('Date').plot(y = 'weighted_price', lw = 2,
style='black', figsize=(18,10))
plt.xlabel('Date', fontsize=25, fontname='DejaVu Sans')
plt.ylabel('Price', fontsize=25, fontname='DejaVu Sans')
plt.title('BTC vs AUD between 2014-2016', fontsize=30, fontname='DejaVu
Sans')
plt.setp(plt.gca().get_xticklabels(), rotation=45, horizontalalignment='right',
fontsize=18)
plt.setp(plt.gca().get_yticklabels(), fontsize=18)
plt.style.use('ggplot')


# LTCBTC graph
ltc_graphed = dataframe.set_index('Date')
ltc_graphed.plot(y = 'LTCBTC', kind = 'area', color = 'purple',
figsize=(18,10))
plt.setp(plt.gca().get_xticklabels(), rotation=45, horizontalalignment='right',
fontsize=18)
plt.setp(plt.gca().get_yticklabels(), fontsize=18)
pass
plt.xlabel('Date', fontsize=25, fontname='DejaVu Sans')
plt.ylabel('Price', fontsize=25, fontname='DejaVu Sans')
plt.title('The price of LTC vs BTC between 2014-2016', fontsize=30,
fontname='DejaVu Sans')
plt.style.use('ggplot')

# BTCAUD candlestick plot
candles_btc = dataframe.drop(['LTCBTC','vol_aud','vol_btc',
'weighted_price'], 1)[0:591]

candles_btc.columns = ['Date', 'Open', 'High', 'Low', 'Close']
candles_btc['Date'] = candles_btc['Date'].map(lambda x: pd.to_datetime(x,
dayfirst=True)).map(mdates.date2num)

fig = plt.figure(figsize=(18,10))
ax1 = plt.subplot2grid((6,1), (0,0), rowspan=6, colspan=1)

#Converts raw mdate numbers to dates
ax1.xaxis_date()
```

```
ax1.set_facecolor('white')
#Making candlestick plot
candlestick_ohlc(ax1,candles_btc.values,width=3, colorup='g', colordown='r',alpha=0.75)
#plt.legend('BTCAUD')
plt.xlabel('Date', fontsize=25, fontname='DejaVu Sans')
plt.ylabel('Price', fontsize=25, fontname='DejaVu Sans')
plt.title('BTC vs AUD between 2014-2016', fontsize=30, fontname='DejaVu Sans')
plt.style.use('ggplot')
#plt.text(1.1,12, r'$y = x^2$', fontsize=15, bbox={'facecolor':'yellow'})
plt.setp(plt.gca().get_xticklabels(), rotation=45, horizontalalignment='right', fontsize=18)
plt.setp(plt.gca().get_yticklabels(), fontsize=18)
pass
```

Unix

I have used the previous analysis that was conducted in ‘Project Stage 1’ and extended my ‘Project Stage 2’ report to include this aspect of my analysis as Unix played a major role in getting me started with my dataset.

Analysis by the author:

1. **Finding the highest BTC price at a fixed date.**
2. **The total number of rows where the opening price of BTC is greater than $700**
3. **The date where both BTC’s weighted_price and LTCBTC's were at their lowest together**
4. **Other analysis**

Code

```
#! /usr/bin/env bash

# The number of lines in the file including the header
wc -l mergeLBAUD.csv

# The number of rows do not have a N/A value for BTC
tail -n+2 mergeLBAUD.csv | egrep -v 'N/A' | wc -l

# The highest weighted price of BTCAUD and the date it was in
cut -d',' -f'1,9' mergeLBAUD.csv | sort -t',' -k'2n' | tail -1

# The total number of rows where the opening price of BTCAUD is greater than $700
cut -d',' -f'3' mergeLBaud.csv | awk '$0 > 700 {print $0}' | wc -l

# Gets the date where LTC was worth 0.008BTC and the BTC high (4rth column)
# at the exchanges was over $315
grep '0.00800' mergeLBAUD.csv | awk -F',' '$4 > 315 {print $1}'

# Gets the largest 'close' value recorded for Bitcoin where
# the 'high' is greater than $350 and the 'low' is less than $300
cut -d',' -f'4,5,6' mergeLBAUD.csv | awk -F',' '$1 > 350 && $2 < 300 { print $3 }' |\
sort -n | tail -1

# The date where both BTCAUD's weighted_price and LTCBTC's were at their lowest
together overall
cut -d',' -f'1,2,9' mergeLBAUD.csv | tail -n+2 |\
sort -t',' -k'3,3n' -k'2,2n' | head -1 | cut -d',' -f'1'
```

SQL

I found from all the tools, that SQL was the easiest in getting results according to specifications. It also could be opened using a professional tool like Postico, or PGAdmin as well as being operated on a bash terminal.

SQL does not graph data**, however in obtaining complex results and then exporting these results to a format readable by pandas, excel, R studio or any other tool** would be done very efficiently by SQL.

Analysis by the author:

1. **Checking whether the data is outputted properly or not**
2. **Get the values and the dates of the opening price of BTC in the month of April in 2015.**
3. **Get the dates and LTCBTC values of the top ten highest LTCBTC values**
4. **Average the open, high, low, closing values of BTC together and find how many are greater than the average weighted price**
5. **Find the average price of LTCBTC**
6. **Other queries**

Code:

Creating the table

```
-- Makes sure the table is created and the data is inputted in the table. Otherwise rollback to previous state.
START TRANSACTION;

DROP TABLE IF EXISTS cryptocurrency CASCADE;

CREATE TABLE cryptocurrency (
        date DATE,
        ltcbtc NUMERIC,
        open NUMERIC(10),
        high NUMERIC(10),
        low NUMERIC(10),
        close_p NUMERIC(10),
        vol_btc NUMERIC(10),
        vol_aud NUMERIC(10),
        weighted_price NUMERIC(10),
        PRIMARY KEY(DATE)
);

-- Formats date so it can be inputted in the SQL Db
SET datestyle TO DMY;

-- Modify the file location if you wish to test it.
```

```
COPY cryptocurrency(date, ltcbtc, open, high, low, close_p, vol_btc, vol_aud,
weighted_price)
FROM '/Users/kbgoda/Desktop/Y1S1/INFO1903/Assignments/Project Stage
2/mergeLBAUD.csv'
WITH CSV HEADER DELIMITER AS ',';

CREATE INDEX ON cryptocurrency(vol_btc);

COMMIT;
```

The queries

```
-- Analysis on BTCAUD, and LTCBTC
-- Some of these queries are the same as the Unix based ones.
-- However this is done in order to analysis how efficient the tools are compared to each
other.
-- There are other queries which are exclusively done in SQL

-- Checks whether the data is outputted properly
SELECT *
FROM cryptocurrency
LIMIT 4;

-- Columns are date, ltcbtc, open, high, low, close_p, vol_btc, vol_aud, weighted_price in
this table

-- Get the values and the dates of the opening price of BTC in the month of April in 2015.
SELECT date AS Date, open AS Opening_BTCAUD_Price
FROM cryptocurrency
WHERE date BETWEEN '2014-04-01' AND '2014-04-30';

-- Count the total values in the weighted_price column where it is not null
-- I do not need to add a where clause as PSQL automatically removes the null values in
the weighted_price column
SELECT COUNT(weighted_price)
FROM cryptocurrency;

-- Get the dates and LTCBTC values of the top ten highest LTCBTC values
SELECT date, ltcbtc
FROM cryptocurrency
ORDER BY ltcbtc DESC
LIMIT 10;

-- Average the open, high, low, close_p values together and find how many are greater than
the avg weighted price
SELECT round(avg(open), 0) AS open, round(avg(high), 0) AS high, round(avg(low), 0)
AS low, round(avg(close_p), 0) AS close_p
```

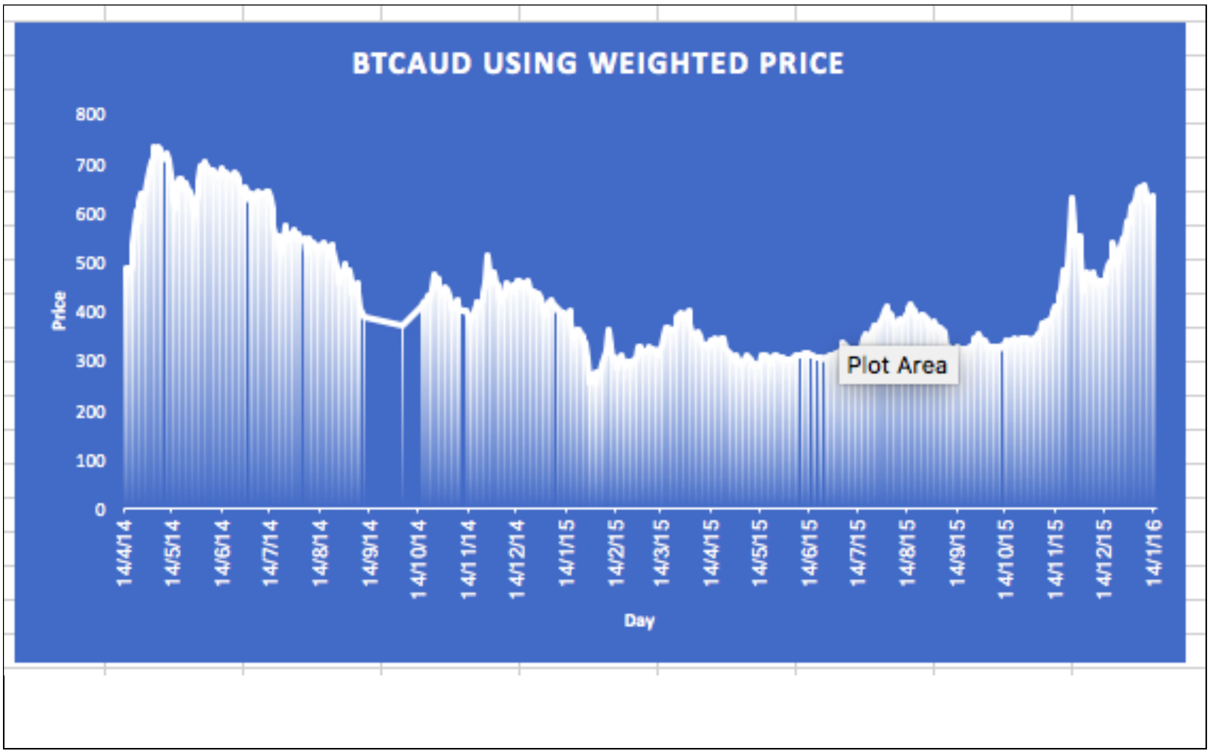
```
FROM cryptocurrency
WHERE open > (SELECT avg(weighted_price) FROM cryptocurrency) AND
            high > (SELECT avg(weighted_price) FROM cryptocurrency) AND
             low > (SELECT avg(weighted_price) FROM cryptocurrency) AND
   close_p > (SELECT avg(weighted_price) FROM cryptocurrency);

-- Count the total volume of BTC where the closing price was $350
SELECT '$' || (SUM(vol_aud)) AS Total_AUD_Volume
FROM cryptocurrency
WHERE close_p = 350;

-- Find the average price of LTCBTC
SELECT trunc(avg(LTCBTC), 4) AS avg_ltcbtc_price
FROM cryptocurrency;

-- Find the value of LTCBTC, weighted BTC price and the date where the high value of
BTCAUD was $300 using a subquery
-- Sort the results in ascending LTCBTC values
SELECT date, weighted_price, ltcbtc
FROM cryptocurrency
WHERE high = (SELECT high WHERE high = 300)
ORDER BY ltcbtc ASC;
```

Excel

The majority of my analysis was not done on Excel, and overall I do not believe I used a lot of Excel's tools. The only reason I used Excel was because I wanted to graph BTCAUD's weighted price so I could compare how the process of how the creation of graphs was compared to Jupyter Notebook.

I was amazed at how quick the process was and how I did not need much complex skills at all. This helped me remove my bias on why Excel is obsolete and instilled a new point of view where I understood that the majority of people in this society **are not very tech savvy.** This is why Excel is such a good tool for these people. It gives them a way to interpret data on their own even if it may not be the most effective way possible.

The visualization using Excel



Conclusion

The analysis conducted shows how we can extrapolate conclusions based on meaningful information obtained using Python, Unix, SQL and Excel from large, clunky datasets. Ultimately it also tells us that no one in hindsight could have predicted the BTC rise. The majority of volume was between $300-400, and in the other price ranges, there was very limited volume. This proves to us that no one was thinking to buy it when it was more expensive in that time. (BTC was $3000 AUD when this report was made in May 2017. The author claims to have purchased BTC during 2014).