**Crypto-currencies Spot vs. Futures**

**1. CASE STUDY:**

Our aim is to analyze the crypto-currency market datasets provided by ICE, understand the gaps (if any) in volume or price distribution , build programs to detect anomalies or outliers in such data sets, find out comparative time stamp relationships from the data sets and study the behavior of crypto futures vis-à-vis spot. There are different types of data feeds like daily data feed which gives an aggregate trading information of a day in form of closing price and total trading volume of the day, similarly there are data feeds which provide 1-minute bars of trading information. The data feed on which we will be working on is high frequency trading (HFT) data feed which provide huge trading information in millions of rows for a single trading day.

**2. BROAD SCOPE OF STUDY:**

1. Perform data classification by source (exchange).
2. Filter the data feed i.e. remove the outliers from a given feed.
3. Perform analysis of data based on frequency, volatility & volume patterns.
4. Identify gaps in the data and perform comparative analysis and provide parameters such as update frequencies, trading activity, order book, etc.
5. Study the relation and movement pattern of Crypto Futures vis-à-vis Spot.

**3. ICE DATA FEED:**

**3.1 Data Coverage**

ICE offers multi-asset and multi-venue data feed from crypto-currency exchanges globally. The data is captured from more than 15 crypto-currency exchange venues globally and normalized to create a consolidated database so that each data row contains a unique sequence number, details on where the trade took place, and other relevant order book data such as quantity, price, currency and timestamp.

Broad coverage includes seven digital currencies from global trading venues, markets and exchanges around the world:

* Bitcoin
* Ether
* Litecoin
* Tether
* Dash
* Ripple
* Bitcoin Cash

The above seven digital currencies are paired against USD and other major currencies. The Bitcoin(XBT)/USD pair averages over 1,200,000 updates per day. Also, the feed normally contains more than 4,000,000 updates per day across all digital pairs.

You can find many other crypto-currencies too recently added to the consolidated feed.

**3.2 Data Description**

Below table provides a brief description of traded pair symbol (Ticker Symbol) for Spot files. Some new ticker symbols (other than present in the below list) can be present in the data feed depending on the type of data file:

|  |  |
| --- | --- |
| **Ticker Symbol** | **Description** |
| X:SXBTXRP | Bitcoin (b) vs Ripple Spot (XBT/XRP) |
| X:SXBTUST | Bitcoin (b) vs USD Tether Spot (XBT/UST) |
| X:SXBTCAD | Bitcoin (b) vs Canadian Dollar Spot (XBT/CAD) |
| X:SXBTCNY | Bitcoin (b) vs China Yuan Renminbi Spot (XBT/CNY) |
| X:SXBTEUR | Bitcoin (b) vs Euro Spot (XBT/EUR) |
| X:SXBTJPY | Bitcoin (b) vs Japanese Yen Spot (XBT/JPY) |
| X:SXBTMXN | Bitcoin (b) vs Mexican Peso Spot (XBT/MXN) |
| X:SXBTGBP | Bitcoin (b) vs Pound Sterling Spot (XBT/GBP) |
| X:SXBTRUB | Bitcoin (b) vs Russian Ruble Spot (XBT/RUB) |
| X:SXBTSGD | Bitcoin (b) vs Singapore Dollar Spot (XBT/SGD) |
| X:SXBTUSD | Bitcoin (b) vs United States Dollar Spot (XBT/USD) |
| X:SETHXBT | Ether (b) vs Bitcoin Spot (ETH/XBT) |
| X:SETHUST | Ether (b) vs USD Tether Spot (ETH/UST) |
| X:SETHCAD | Ether (b) vs Canadian Dollar Spot (ETH/CAD) |
| X:SETHCNY | Ether (b) vs China Yuan Renminbi Spot (ETH/CNY) |
| X:SETHEUR | Ether (b) vs Euro Spot (ETH/EUR) |
| X:SETHGBP | Ether (b) vs Pound Sterling Spot (ETH/GBP) |
| X:SETHJPY | Ether (b) vs Japanese Yen Spot (ETH/JPY) |
| X:SETHMXN | Ether (b) vs Mexican Peso Spot (ETH/MXN) |
| X:SETHUSD | Ether (b) vs United States Dollar Spot (ETH/USD) |
| X:SLTCXBT | Litecoin (b) vs Bitcoin Spot (LTC/XBT) |
| X:SLTCUST | Litecoin (b) vs USD Tether Spot (LTC/UST) |
| X:SLTCCNY | Litecoin (b) vs China Yuan Renminbi Spot (LTC/CNY) |
| X:SLTCEUR | Litecoin (b) vs Euro Spot (LTC/EUR) |
| X:SLTCGBP | Litecoin (b) vs Pound Sterling Spot (LTC/GBP) |
| X:SLTCJPY | Litecoin (b) vs Japanese Yen Spot (LTC/JPY) |
| X:SLTCMXN | Litecoin (b) vs Mexican Peso Spot (LTC/MXN) |
| X:SLTCUSD | Litecoin (b) vs United States Dollar Spot (LTC/USD) |
| X:SXRPXBT | Ripple (b) vs Bitcoin Spot (XRP/XBT) |
| X:SXRPEUR | Ripple (b) vs Euro Spot (XRP/EUR) |
| X:SXRPJPY | Ripple (b) vs Japanese Yen Spot (XRP/JPY) |
| X:SXRPMXN | Ripple (b) vs Mexican Peso Spot (XRP/MXN) |
| X:SXRPUSD | Ripple (b) vs United States Dollar Spot (XRP/USD) |
| X:SDAHXBT | Dash (b) vs Bitcoin Spot (DAH/XBT) |
| X:SDAHEUR | Dash (b) vs Euro Spot (DAH/EUR) |
| X:SDAHGBP | Dash (b) vs Pound Sterling Spot (DAH/GBP) |
| X:SDAHUSD | Dash (b) vs United States Dollar Spot (DAH/USD) |
| X:SBCHXBT | Bitcoin Cash (b) vs Bitcoin Spot (BCH/XBT) |
| X:SBCHETH | Bitcoin Cash (b) vs Ether Spot (BCH/ETH) |
| X:SBCHUST | Bitcoin Cash (b) vs USD Tether Spot (BCH/UST) |
| X:SBCHEUR | Bitcoin Cash (b) vs Euro Spot (BCH/EUR) |
| X:SBCHJPY | Bitcoin Cash (b) vs Japanese Yen Spot (BCH/JPY) |
| X:SBCHGBP | Bitcoin Cash (b) vs Pound Sterling Spot (BCH/GBP) |
| X:SBCHUSD | Bitcoin Cash (b) vs United States Dollar Spot (BCH/USD) |

**3.3 File Type**

The files given to us are Consolidated Tick Data Files which include trade and quote data (and Best Bid and Offer (BBO) data too in case of Futures) for a given day from multiple trading venues for multiple ticker symbols (as described above). There are four types of file types given to us whom we can name as: Spot-1, Spot-2, Futures-1 and Futures-2. File names for all types are in below format:

Spot-1: PLUSTICK\_<Source ID>\_yyyymmdd.txt

where

PLUSTICK = a constant for all consolidated tick data files

<srcID> = numeric code for the source of data feed being reported, in Spot-1 files the ID is 1619 representing **ICE Data Services: Digital Currencies Streaming**

yyyymmdd = the date in New York at the time the file is generated

.txt = file extender showing that the file is in .txt format

Spot-2: PLUSTICK\_FI\_<Source ID>\_yyyymmdd.txt

where

PLUSTICK = a constant for all consolidated tick data files

FI = Fixed Income

<srcID> = numeric code for the source of data feed being reported, in Spot-2 files the ID is 1356 representing **Blockstream : Crypto Data Feed Streaming Level 1**

yyyymmdd = the date in New York at the time the file is generated

.txt = file extender showing that the file is in .txt format

Futures-1: PLUSTICK\_FUTURES\_<Source ID>\_yyyymmdd.txt

where

PLUSTICK = a constant for all consolidated tick data files

<srcID> = numeric code for the source of data feed being reported, in Fututres-1 files the ID is 666 representing **Cboe: Futures Exchange Level 1**

yyyymmdd = the date in New York at the time the file is generated

.txt = file extender showing that the file is in .txt format

Futures-2: PLUSTICK\_FUTURES\_<Source ID>\_yyyymmdd.txt

where

PLUSTICK = a constant for all consolidated tick data files

<srcID> = numeric code for the source of data feed being reported, in Fututres-2 files the ID is 680 representing **Chicago Mercantile Exchange: Globex Level 1**

yyyymmdd = the date in New York at the time the file is generated

.txt = file extender showing that the file is in .txt format

**3.4 File Format**

There are different types of metadata rows in the four files, and some data fields corresponding to a common metadata row might be different. The fields which are NOT common in two types of Spot files or Futures files have been highlighted in BLUE.

Table 1: *Types of metadata rows present in Spot-1 (1619) daily data feeds along with data fields.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Row** | **Opening Identifier** | **Total No. of fields** | **Data Fields** |
| Header | #D=H | 10 | #D=H|<ENUM.SRC.ID>|<SYMBOL.TICKER>|<ABRV.CURRENCY>|<ISIN>|<SEDOL>|<CUSIP>|<ENUM.INSTR.TYPE>|<CONTRACT.SIZE>|<VARIABLE.TICK.SIZE>|<FX.ADJUSTMENT.FACTOR> |
| Quote | #D=Q | 12 | #D=Q|<TAS.SEQ>|<ACTIVITY.DATETIME>|<BID.PRICE>|<BID.SIZE>|<ASK.PRICE>|<ASK.SIZE>|<QUOTE.COND\_1>|<CONTRIBUTOR.ID>|<REGION.CODE>|<CITY.CODE>|<QUOTE.DATETIME>|<EXCH.MESSAG  E.TIMESTAMP> |
| Trade | #D=T | 18 | #D=T|<TAS.SEQ>|<ACTIVITY.DATETIME>|<TRADE.PRICE>|<TRADE.SIZE>|<TRADE.COND\_1>|<CONTRIBUTOR.ID>|<REGION.CODE>|<CITY.CODE>|<TRADE.DATETIME>|<EXCH.MESSAGE.TIMESTAMP>|<TRADE  .COND\_2>|<TRADE.COND\_3>|<TRADE.OFFICIAL.TIME>|<TRADE.COND\_4>|<TRADE.COND\_5>|<EXTENDED.TRADE.COND>|<TRADE.OFFICIAL.DATE>|<RETRANSMISSION.FLAG> |
| Status | #D=S | 5 | #D=S|<TAS.SEQ>|<ACTIVITY.DATETIME>|<TRADABLE.STATUS>|<MARKET.PHASE>|<EXCH.MESSAGE.TIMESTAMP> |

Table 2: *Types of metadata rows present in Spot-2 (1356) daily data feeds along with data fields.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Row** | **Opening Identifier** | **Total No. of fields** | **Data Fields** |
| Header | #D=H | 8 | #D=H|<ENUM.SRC.ID>|<SYMBOL.TICKER>|<ABRV.CURRENCY>|<FX.ADJUSTMENT.FACTOR>|<CONTRACT.SIZE>|<ISIN>|<ENUM.INSTR.TYPE>|<VARIABLE.TICK.SIZE> |
| Quote | #D=Q | 14 | #D=Q|<TAS.SEQ>|<ACTIVITY.DATETIME>|<ASK.PRICE>|<BID.PRICE>|<QUOTE.DATETIME>|<QUOTE.OFFICIAL.TIME>|<EXCH.MESSAGE.TIMESTAMP>|<BID.SIZE.DEC>|<ASK.SIZE.DEC>|<CONTRIBUTOR.ID  >|<QUOTE.OFFICIAL.DATE>|<REGION.CODE>|<CITY.CODE>|<QUOTE.COND\_4> |
| Trade | #D=T | 13 | #D=T|<TAS.SEQ>|<ACTIVITY.DATETIME>|<TRADE.PRICE>|<TRADE.DATETIME>|<EXCH.MESSAGE.TIMESTAMP>|<TRADE.SIZE.DEC>|<TRADE.VOL.DEC>|<CONTRIBUTOR.ID>|<TRADE.OFFICIAL.DATE>|<TRAD  E.OFFICIAL.TIME>|<REGION.CODE>|<CITY.CODE>|<RETRANSMISSION.FLAG> |
| Status | #D=S | 5 | #D=S|<TAS.SEQ>|<ACTIVITY.DATETIME>|<TRADABLE.STATUS>|<MARKET.PHASE>|<EXCH.MESSAGE.TIMESTAMP> |

Table 3: *Types of metadata rows present in Futures-1 (666) daily data feeds along with data fields.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Row** | **Opening Identifier** | **Total No. of fields** | **Data Fields** |
| Header | #D=H | 20 | #D=H|<ENUM.SRC.ID>|<SYMBOL.TICKER>|<ABRV.CURRENCY>|<ISIN>|<SEDOL>|<CUSIP>|<DISPLAY.PRECISION>|<TRADE.OPEN>|<OPEN\_INT>|<YEST.TRADE.CLOSE>|<YEST.TRADE.VOL>|<SETTLE.DATE>|<SETTLE.PRICE>|<EXPIRATION.DATE>|<FRONT.MONTH.CONTRACT>|<MARKET.ID>|<ENUM.INSTR.TYPE>|<LOT.SIZE>|<CONTRACT.SIZE>|<VARIABLE.TICK.SIZE> |
| Quote | #D=Q | 11 | #D=Q|<TAS.SEQ>|<RNR.END.EXCH.SEQ>|<ACTIVITY.DATETIME>|<BID.PRICE>|<BID.SIZE>|<ASK.PRICE>|<ASK.SIZE>|<QUOTE.COND\_1>|<PART.CODE>|<QUOTE.DATETIME>|<EXCH.MESSAGE.TIMESTAMP> |
| Trade | #D=T | 18 | #D=T|<TAS.SEQ>|<RNR.END.EXCH.SEQ>|<ACTIVITY.DATETIME>|<TRADE.PRICE>|<TRADE.SIZE>|<TRADE.COND\_1>|<PART.CODE>|<VWAP>|<TRADE.DATETIME>|<EXCH.MESSAGE.TIMESTAMP>|<TRADE.COND\_2>|<TRADE.COND\_3>|<TRADE.OFFICIAL.TIME>|<TRADE.COND\_4>|<TRADE.COND\_5>|<EXTENDED.TRADE.COND>|<TRADE.OFFICIAL.DATE>|<RETRANSMISSION.FLAG> |
| Status | #D=S | 8 | #D=S|<TAS.SEQ>|<ACTIVITY.DATETIME>|<MKT.PHASE>|<EXCH.MESSAGE.TIMESTAMP>|<MARKET.PHASE>|<TRADABLE.STATUS>|<INSTR.STATUS>|<STATE.FLAG> |
| BBO | #D=B | 10 | #D=B|<TAS.SEQ>|<RNR.END.EXCH.SEQ>|<ACTIVITY.DATETIME>|<BID.PRICE>|<BID.SIZE>|<BID.PART.CODE>|<ASK.PRICE>|<ASK.SIZE>|<ASK.PART.CODE>|<EXCH.MESSAGE.TIMESTAMP> |
| Block Trade | #D=L | 3 | #D=L||<TAS.SEQ>|<LIMIT.HIGH>|<LIMIT.LOW> |

Table 4: *Types of metadata rows present in Futures-2 (680) daily data feeds along with data fields.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Type of Row** | **Opening Identifier** | **Total No. of fields** | **Data Fields** |
| Header | #D=H | 19 | #D=H|<ENUM.SRC.ID>|<SYMBOL.TICKER>|<ABRV.CURRENCY>|<ISIN>|<SEDOL>|<CUSIP>|<DISPLAY.PRECISION>|<TRADE.OPEN>|<OPEN\_INT>|<YEST.TRADE.CLOSE>|<YEST.TRADE.VOL>|<SETTLE.DATE>|<SETTLE.PRICE>|<EXPIRATION.DATE>|<FRONT.MONTH.CONTRACT>|<ENUM.INSTR.TYPE>|<LOT.SIZE>|<CONTRACT.SIZE>|<VARIABLE.TICK.SIZE> |
| Quote | #D=Q | 11 | #D=Q|<TAS.SEQ>|<RNR.END.EXCH.SEQ>|<ACTIVITY.DATETIME>|<BID.PRICE>|<BID.SIZE>|<ASK.PRICE>|<ASK.SIZE>|<QUOTE.COND\_1>|<PART.CODE>|<QUOTE.DATETIME>|<EXCH.MESSAGE.TIMESTAMP> |
| Trade | #D=T | 18 | #D=T|<TAS.SEQ>|<RNR.END.EXCH.SEQ>|<ACTIVITY.DATETIME>|<TRADE.PRICE>|<TRADE.SIZE>|<TRADE.COND\_1>|<PART.CODE>|<VWAP>|<TRADE.DATETIME>|<EXCH.MESSAGE.TIMESTAMP>|<TRADE.COND\_2>|<TRADE.COND\_3>|<TRADE.OFFICIAL.TIME>|<TRADE.COND\_4>|<TRADE.COND\_5>|<EXTENDED.TRADE.COND>|<TRADE.OFFICIAL.DATE>|<RETRANSMISSION.FLAG> |
| Status | #D=S | 6 | #D=S|<TAS.SEQ>|<ACTIVITY.DATETIME>|<INSTR.STATUS>|<EXCH.MESSAGE.TIMESTAMP>|<MARKET.PHASE>|<TRADABLE.STATUS> |
| BBO | #D=B | 10 | #D=B|<TAS.SEQ>|<RNR.END.EXCH.SEQ>|<ACTIVITY.DATETIME>|<BID.PRICE>|<BID.SIZE>|<BID.PART.CODE>|<ASK.PRICE>|<ASK.SIZE>|<ASK.PART.CODE>|<EXCH.MESSAGE.TIMESTAMP> |

The description of these data fields can be found in an Information Guides (.PDF files) present in the same directory where data files are present (path: **/space/data/new**). Please have a look at the guides for better understanding of Data Fields.

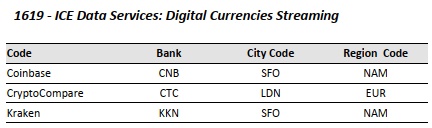
**NOTE:** On source ID 1356 (Spot-2 files), the level 1 venue status messages using MARKET.PHASE and TRADABLE.STATUS are delivered using the special symbol .EXCHANGE.ADMIN with the contributor ID appended. For example,

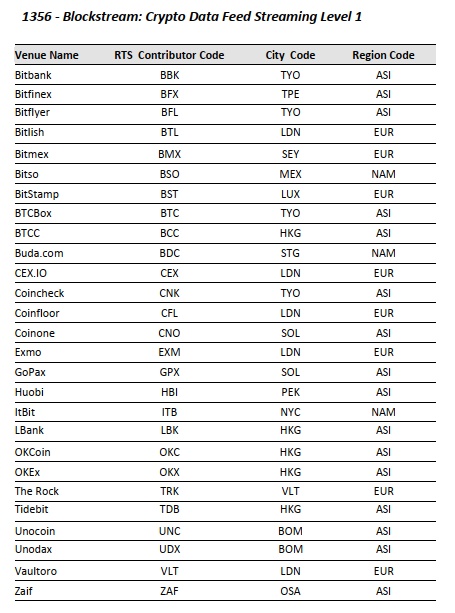
.EXCHANGE.ADMIN@BFX|1708=1|1709=4

To get more information, go through pages 12 and 13 of ‘Digital\_Currencies\_Developer \_Supplement\_2019-01-30\_ICEDataServices’.

**4. SPOT SOURCE IDs (1619, 1356)**:

Spot-1 file is associated with source ID of 1619 representing ‘**ICE Data Services: Digital Currencies Streaming**’ and Spot-2 file is associated with source ID of 1356 representing ‘**Blockstream : Crypto Data Feed Streaming Level 1’**. Both the file types derive their data from multiple venues. For example, Spot-1 file derive their data from 3 venues – Coinbase (CNB), CryptoCompare (CTC), and Kraken (KKN). The data from these 3 venues cover 2 regions (NAM and EUR) and 2 cities (SFO and LDN). Similarly one can see from below tables that Spot-2 file type derive their data from 27 venues and 3 regions. In metadata rows the venue code, region code and city code are marked by tokens CONTRIBUTOR.ID, REGION.CODE and CITY.CODE respectively. Also, the electronic currencies traded on the sources and venues detailed in Spot-1 and Spot-2 file types are traded 24/7.





Quote row (Q) or Trade row (T) etc. can belong to either of venue code and associated region and city codes depending on the file type. The venue-wise and region-wise distribution of quotes & trades might be very helpful for comparative analysis.

**5. FUTURES SOURCE IDs (666, 680):**

i) The source ID of Fututres-1 file type is 666 and it represents **Cboe: Futures Exchange Level 1**. CBOE’s regular trading hours are from 8:30 AM to 3:00 PM Central Time.

Field SYMBOL.TICKER for Futures in 666 type files is in below format:

F2:cc[cccc]\myy

Where:

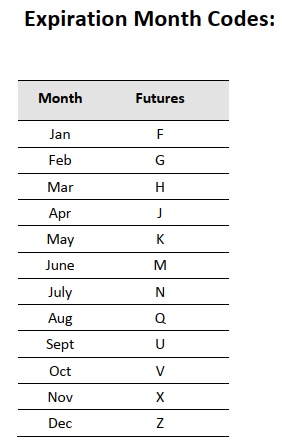
F2: Electronic traded future

cc[cccc]: Up to six-digit contract code

m: One digit month code

yy: Two digit year

where month codes are mentioned below



Example:

F2:GV\F12

Where:

F2: = Electronic traded future

GV = GV contract

F12 = January, 2012 expiration

ii) The source ID of Fututres-2 file type is 680 and it represents **Chicago Mercantile Exchange: Globex Level 1**. CME’s trading hours are not fixed throughout the year.

Field SYMBOL.TICKER for Futures in 680 type files is in below format:

Fn:CommCode\myy

Where:

F = futures contract

n = session indicator (1 = Regular trading, 2 = Electronic trading)

CommCode = Commodity code as provided by the exchange

m = Alphabetic month code (month codes are same as that of 666 type files)

yy = Two-digit year code

Example of Futures Symbology

F1:S\U04

Where:

F = Futures contract

1 = Regular trading

S = Soy beans

U = September

04 = 2004

**6. PROJECT TASK LAYOUT:**

Project tasks have been divided into 3 phases as described below. For each task in all phases, teams need to maintain their programming codes i.e. the functions required for a group of tasks should be stored in one python module & all python modules should be stored in a separate directory. To make the code explainable, proper comments should be put along code lines. Also, make copies of original data files for testing of the programs.

**PHASE 1: Data Classification & Formatting**

First observe the data feed of different file types and list out the ticker symbols of Spot and Futures files. After this match and make a list of the ticker symbols of Futures files (both type) with underlying crypto-currencies’ ticker symbols of Spot-1 and Spot-2 file types. This will help us for further analysis and comparison.

Now for the initial research, for all the four type of files (Spot-1, Spot-2, Futures-1, and Futures-2), build common programs as mentioned below:

1. **Program-1**: Develop a program to plot the trading price data for the required ticker symbols. The program should ask the user to input the ‘Date’, ‘File Type’ and the ‘Ticker Symbol’ (out of available ticker symbols in a proper format) for which plotting needs to be done. On the x-axis, there should be time in 12 hour AM/PM format and on the y-axis, there should be trading prices at which crypto-currency was traded and the title of the plotted image should contain the ‘Ticker Symbol’, ‘File Type’ and ‘Date’ of the plotted data. Include proper legends in the plot. The plots will help one to find out whether there are any data gaps. Also, store the images in ‘.PNG’ format. **(Refer NOTE 2 for conversion of UNIX timestamp to actual date-time format).**
2. **Program-2**: Develop a program to plot the quote price data (bid price in RED and ask price in BLUE on the same plot) for the required ticker symbols. The program should ask the user to input the ‘Date’, ‘File Type’ and the ‘Ticker Symbol’ (out of available ticker symbols in a proper format) for which plotting needs to be done. On the x-axis, there should be time in 12 hour AM/PM format and on the y-axis, there should be bid and ask prices for a crypto-currency and the title of the plotted image should contain the ‘Ticker Symbol’, ‘File Type’ and ‘Date’ of the plotted data. Include proper legends in the plot. The plots will help one to find out whether there are any data gaps. Also, store the images in ‘.PNG’ format**. (Refer NOTE 2 for conversion of UNIX timestamp to actual date-time format and Refer Appendix A-1 for sample graph – made for a stock).**
3. **Program-3**: Develop a program to plot the BBO price data (bid price in RED and ask price in BLUE on the same plot) for the required ticker symbols (of Futures files). The program should ask the user to input the ‘Date’, ‘File Type’ and the ‘Ticker Symbol’ (out of available ticker symbols) for which plotting needs to be done. On the x-axis, there should be time in 12 hour AM/PM format and on the y-axis, there should be bid and ask prices for a crypto-currency and the title of the plotted image should contain the ‘Ticker Symbol’, ‘File Type’ and ‘Date’ of the plotted data. Include proper legends in the plot. The plots will help one to find out whether there are any data gaps. Also, store the images in ‘.PNG’ format**. (Refer NOTE 2 for conversion of UNIX to actual date-time format and Refer Appendix A-1 for sample graph – made for a stock).**

NOTE: The above three programs can be merged to have a single program that asks a user about the price type to plot- Trade price, Quote price or BBO price.

1. **Program-4**: Develop a program which asks the user to enter the date in ‘yyyymmdd’ format and ‘File type’ to choose a specific file from the given data files. Program then should output an ‘excel’ file containing all ticker symbols as different rows and trading activity in form of different columns such as no. of quotes (Q), no. of trades (T), no. of BBOs (B) and traded volume (V) on ‘Overall market’ (ALL) as well as at ‘Venue’ and ‘Region’ levels. The sample table for Spot-1 file type is mentioned below.

**NOTE**: In Futures files, volume may be mentioned in unit of ‘lot’. Convert the lot into crypto-currencies units.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ticker Symbol | Q- ALL | T-ALL | B-ALL | V-ALL | Q-CNB | .. | T-CNB | … | V-CNB | … | Q-NAM | … |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |

1. **Program-5**: Develop a program to plot trade, quote and BBO price (as per input provided by user) data separately for separate venues (if multiple venues are available) on the same graph (with different colors) for a required pair. The program should ask the user to input the ‘Date’, ‘File type’, ‘Ticker Symbol’ (out of available ticker symbols), ‘Data type’ (Quote or BBO) and ‘Field Type’ (Bid or Ask prices) for which plotting needs to be done. On the x-axis, there should be time in 12 hour AM/PM format and on the y-axis, there should be quote prices and the title of the plotted image should contain the ‘Ticker Symbol’, ‘File type’, ‘Data type’, ‘Field type’ and ‘Date’ of the plotted data. Include proper legends in form of ‘venue codes’ as different curves will be from different exchanges (in Spot files). This will provide a venue-wise view of price movement. Store the images in ‘.PNG’ format. **(Refer Appendix A-2 for sample graph – made for a stock).**
2. **Program-6**: Develop a program to plot the ‘spread’ i.e. the difference between Crypto Futures’ Trade Price and the underlying Crypto Spot’s Trade price. The program should ask the user to input the ‘Date’ and ‘Ticker Symbol’ (out of available ticker symbols) for which plotting needs to be done. If a crypto instrument is common to both ‘Futures’ files or both ‘Spot’ files then program should display the available file types present for that ticker symbol to the user and should ask the user for the ‘File types’ for which spread need to be plotted. Also, in case of multiple expiry dates for a Futures symbol, program is to tell the user about all such symbols and ask the user to enter an expiry date of the symbol for which comparison needs to be done. On the x-axis, there should be time in 12 hour AM/PM format and on the y-axis, there should be spread for a crypto-currency and the title of the plotted image should contain the ‘Ticker Symbol’, ‘File types’ and ‘Date’ of the plotted data. Include proper legends in the plot. Also, store the images in ‘.PNG’ format**.**
3. **Program-7**: Develop a program to plot the ‘daily spread’ i.e. the difference between Crypto Futures’ Closed Trade Price and the underlying Crypto Spot’s Closed Trade price. The program should ask the user to input the ‘Ticker Symbol’ (out of available ticker symbols) for which plotting needs to be done. If a crypto instrument is common to both ‘Futures’ files or both ‘Spot’ files then program should display the available file types present for that ticker symbol to the user and should ask the user for the ‘File types’ for which spread need to be plotted. Also, in case of multiple expiry dates for a Futures symbol, program should plot the ‘daily spread’ for all expiry dates on the same plot. On the x-axis, there should be time in 12 hour AM/PM format and on the y-axis, there should be spread for a crypto-currency and the title of the plotted image should contain the ‘Ticker Symbol’ and ‘Date’ of the plotted data. Include proper legends in the plot. Also, store the images in ‘.PNG’ format**.**

We know that all futures contracts derive their value from their respective underlying. In case of crypto-currency futures, their prices depend on crypto-currency spot prices, and any move in the latter affects the former. This dependency leads to the prices of the two moving in sync with each other, though there is a difference between the two. The theoretical formula for calculating the futures price from the spot price is as follows:

Crypto-currency Futures Price = Crypto-currency Spot price \* [1+ rf\*(x/365)]

Where rf = risk-free rate on annual-basis

And x = number of days to expiry

So, we can see from above formula that spread depends on Spot price as well as Futures’ no. of days to expiry as below:

Spread = Crypto-currency Spot price \* [rf\*(x/365)]

Make a comparison of the observed spread and theoretical spread for different ticker symbols and different expiry dates for a particular ticker symbol.

**NOTE 1:** While dealing with crypto-currency tick data, format of given data feed can be transformed into Pandas DataFrame, if required, where ticker symbols can be mentioned as different rows and all data fields can be mentioned as different columns. Advantage of storing the data into Pandas DataFrame is that Pandas library offers a lot of functions like ‘groupby’, ‘indexing’ which can be helpful for quick data manipulation and also, if some fields are empty, pandas DataFrame put NaN (Not a Number) in place of those so that numerical operations can be performed on the datasets.

**NOTE 2**: The ACTIVITY.DATETIME field under the ‘Q’, ‘T’ and ‘B’ rows is expressed in Epoch (UNIX) Time. Convert the UNIX timestamps under the field ‘ACTIVITY.DATETIME’ to actual date-time which would be UTC. The UNIX timestamps are the number of seconds passed since January 1, 1970 at 00:00:00 UTC. For example, activity time at Saturday, 1st October 2016, 09:30:00 AM is represented by 1475328600 in UNIX timestamp. Use Python’s DateTime module and pandas.timestamp() function to convert timestamps mentioned in the file to actual date & time. Date and time parameters can also include milliseconds as decimal fractions. To get a feel of conversion factor and to change a particular activity.datetime to normal date time or vice-versa, below website link can be used: <http://www.convert-unix-time.com/>

**PHASE 2: Data Cleaning & Analysis**

Till now we have used only raw data (or say dirty data or unfiltered data) to plot different graphs and observing price movements. Now, in this phase we will try to remove most of the price outliers present in the given feeds using a filtering method called (k, gamma) **(Refer Appendix B).**

Price Outliers – The observations which are located significantly far from their preceding or following observations are considered as outliers (or say price outliers). Whether an observation is outlier or not, can be found by comparing the difference between the observation value and mean value with the standard deviation of nearby observations. Such outliers can be seen clearly in the daily plots of the trade, quote or BBO prices.

1. **Program-8**: First, we will look for one more trend. Develop a program to plot volume and price trends together to check the correlation between the two. Generally if quotes’ bid volume increases more rapidly than quotes’ ask volume then trade price increases. To check this hypothesis, make plots in such a way that the difference between quotes’ bid volume and quotes’ ask volume and the trade price are shown on the same plot (one above the other for the better visualization). （量和价格的关系）
2. **Program-9**: Develop a program to remove the price outliers present in the feed using the filtering logic of (k, γ) method explained in **Appendix B**. The program should ask a user to input the ‘Date’, ‘File Type’, ‘Ticker Symbol’, ‘Data type’ (Trade, Quote or BBO) and ‘Field Type’ (Bid or Ask prices in case of Quote or BBO) and the value of ‘k’ for filtering method. Some values of k which are generally used are 20, 40, 60, 80, 100. The user can enter either of these values or some other value. The program after removing the outliers should plot the original (dirty) as well as cleaned price series **(Refer Appendix C for the sample graph).**

Along with this the program should provide following information in a .CSV file or excel file: (i) all outliers founded by the algorithm (ii) the no. of removed outliers and (iii) the highest and lowest trading prices of original (dirty) series as well as filtered (clean) series. This will help to check the cleaning efficiency and compare the “price range” (i.e. high price - low price) of cleaned series with actual “price range” determined through third party data (can fetch it through exchange websites).

Depending upon the cleaning efficiency, re-run the (k, y) algorithm on cleaned data set. This will help us to check whether cleaning efficiency gets improved after running the filtering algorithm twice on a given feed.

Compare the no. of outliers between different currency pairs.

1. **Program-10**: Similar to Program-8 but remove the outliers venue-wise and plot the price series for all venues/regions in same graph using different colors (**Refer Appendix D for the sample graph**). Along with the program, gather the data as mentioned in Program-8 description in a .CSV file.

Compare the no. of outliers between different venues for multiple currency pairs.

1. **Other Programs**: Develop a program to check the effect of filtering of the outliers on the difference between theoretical and observed spreads. ‘New spread’ would be the difference between filtered Crypto Futures’ Trade Price and the filtered Crypto Spot’s Trade price. Compare the two observed spreads - new spread (using filtered data feeds) and old spread (using raw data feeds).

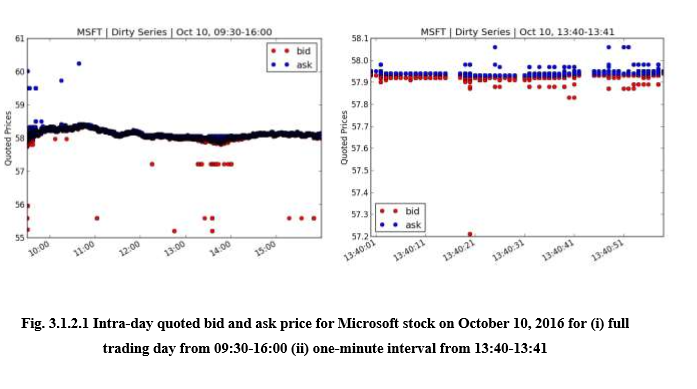
Observe the trends in Futures price movement and underlying crypto-currency’s spot price movement. Plot Futures price and Spot price on the same graph and try to find some relation between two price movements. Find correlations between the two series before filtering and after filtering.

**PHASE 3: Observations & Conclusions**

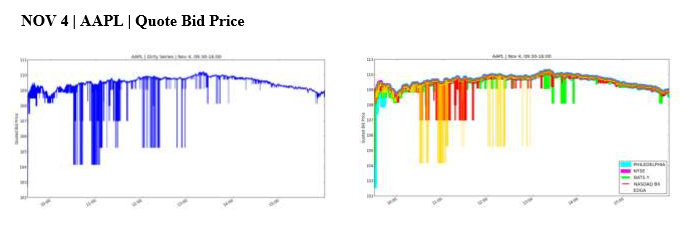
This will be the final phase of the project where we will put together the results/ observations of the work done in previous two phases. Finally, we will arrange the programs in order of their usability in a single directory.

**APPENDIX**

**Appendix A-1**: Below are just the sample graphs which have been made for Microsoft (MSFT) stock. Please replicate the base format of graph i.e. x-axis, y-axis, ‘title’ name, legend, etc.



**Appendix A-2**: Below are the sample graphs for Quoted Bid Price on ‘Overall Basis’ (i.e. Aggregate) as well as at ‘Exchange Level’ and made for Apple (AAPL) stock.

****

**Appendix B:** (k, γ) model proposed by Mineo & Romito in 2007 to be used for removing of price outliers is explained below (This is a modified version of the model proposed by Gallo & Brownlees in 2006)

|现价-均价|< 3\*标准差 + gamma

If {pi}iN=1 is the ordered tick-by-tick price series containing n values where i (= 1, 2, 3 , …., N) is the ith observation or say, trading price of ith transaction then below rulescan be applied to remove the outliers-

True, observation is kept

|pi -pim(k)| < 3 Si(k) + γ = {

False, observation is removed

where,

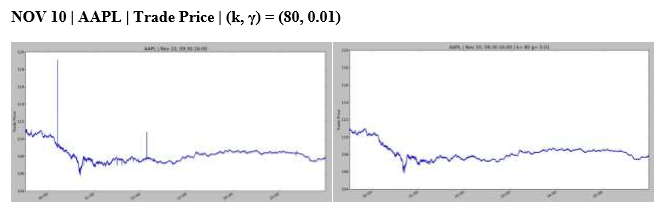
* pim(k) is the sample mean
* Si(k) is the standard deviation
* k is the no. of observations surrounding the ith observation (excluding ith observation)
* And, γ is called the granularity parameter (which is minimum price movement for a crypto-currency pair and can be assumed as 10-10).

Reference:

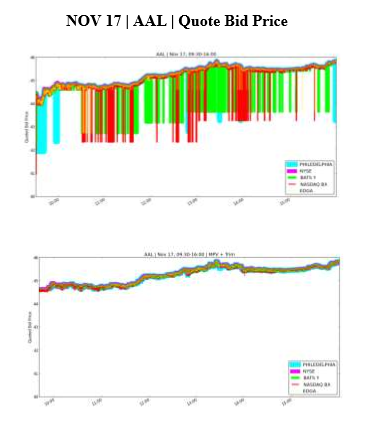
1. A. M. Mineo, F. Romito; *Different Methods to Clean Up Ultra High-Frequency Data*; 2007

2. Christian T. Brownlees, Giampiero M. Gallo; *Financial Econometric Analysis at Ultra–High Frequency: Data Handling Concerns*; 2006

**Appendix C:** Dirty vs. Clean Trade Price Series for AAPL stock using k= 80 & y = 0.01



**Appendix D:** Venue-wise Dirty vs. Clean Bid Price Series for AAL stock using k= 80 & y = 0.01

****