

リアルタイム・データサイエンスへ向けたはじめの一步：データの可視化 ～Windows フリーソフトを使ってビットコイン価格をグラフ化してみる～

総合情報基盤センター 准教授 滝谷 弘

本報では、時々刻々と更新されていく情報、つまりリアルタイムのデータセットに対するデータ解析（リアルタイム・データサイエンス）と Web スクレイピング等を念頭において、学生や研究者などがお手持ちの Windows 環境ですぐにでもはじめられる実時間で取得した時系列データのグラフ作成方法を Windows フリーソフトで実現するひとつの試みをご披露したい。ここでいう Windows 用フリーソフトとはプログラミング言語 Python と高機能グラフ作成ツール gnuplot である。リアルタイム・データセットとしては、ビットコインの売値と買値をとりあげ、これら値の取得には Python からビットコイン取引所の API と対応させることにより実装する。特に、24 時間周期で日付をまたぐ厄介な時系列データのグラフ化を gnuplot の優れた CUI に基づくバッチ処理により実現する。

キーワード：データサイエンス、リアルタイム・データ、Python、gnuplot、ビットコイン

1. はじめに

近年、データサイエンス^(1, 2)がさまざまなシーンに到来している。本学でも学部の文理を問わず、学生たちにデータサイエンスの修学や実践の機会を設けることにより、ビジネスや研究の現場などで即戦力となるデータサイエンティストの育成を目標に掲げている。ところで、データサイエンスといえば、電磁記憶媒体へのランダムアクセスや Web スクレイピング⁽³⁾等により、過去から蓄積された膨大な情報に対するデータ分析を連想する。ところが、昨今では社会経済情勢や気候環境変動など森羅万象の至るところで、その系全体がドラスティックに変貌を遂げており⁽⁴⁾、データサイエンスの観点からすれば、次世代のデータ解析にて扱うべき代物は、ダイナミックなデータセットへと変わりつつある。

そこで、本報では、時々刻々と更新されていくデータセット、つまりリアルタイム・データセットに対するデータ解析を念頭において、学生や研究者などがお手持ちの Windows 環境で手頃にはじめられる時系列データのグラフ作成方法をフリーソフトで実現するひとつの方法を解説する。ここでいう Windows 用フリーソフトとはプログラミング言語 Python⁽²⁾と高機能グラフ作成ツール

gnuplot^(5, 6)である。gnuplot は、フリーの高機能グラフ作成ツール（ソフトウェア）である。特に Linux OS 環境マシンで研究開発を行ってこられた方々にとっては、gnuplot はなじみ深いものと思われる。最近では、Windows マシンでもプログラミングや研究開発の環境が整いやすくなり、Windows にても (Mac OS でも) 動作する gnuplot がインターネット上⁽⁶⁾から取得してフリーライセンスとして使えるようになった。Windows ユーザーからすれば、マイクロソフト社の Excel スプレッドシートのグラフ作成が王道のように思われるかもしれないが、学術論文に掲載できるほどのハイクオリティーなグラフを作成するには、細かい設定のできる CUI (Character User Interface) ベースの gnuplot を越えるフリーソフトはないと著者は思っている。それから、Python であるが、データサイエンスにも親和性が高く、多くのデータ解析用モジュールが用意されているため、もはやデータサイエンスのためのデファクト・スタンダードともいえるプログラミング言語となりつつある。なお、本報で扱うリアルタイム・データセットとしては、ビットコインの売値と買値を取り上げ、Python コードからビットコイン取引所 bitFlyer の API と対応させることにより実装する。

2. 日付をまたぐ24時間周期の時系列データをいかにgnuplotでグラフ化するか

ここでは、2次元データとして横軸に時間、縦軸に実数値からなる時系列データをグラフ化することを考える。なお、本報では解説の冗長を避けるため、折れ線グラフの描画に絞って解説する。時間（横軸）は、西暦の年月日および時刻（時分秒）から構成され、縦軸をとる折れ線には実数値を扱う。なお、本報では実数値としてビットコインの売値 `bit`（買い気配）と買値 `ask`（売り気配）の二つを扱う。ビットコインのデータセット取得には Python のコード中、`CCXT` ライブラリを用いることで、複数の仮想通貨/ビットコイン取引所の API（Application Programming Interface）と対応させることができる。なお、`CCXT`（Crypto Currency eXchange Trading）は、ビットコインやアルトコインといった仮想通貨の売買を自動化するための Python/JavaScript/PHP 向けライブラリである。ところで、横軸に時分秒（年月日を除いたもの）のみを表示したいとき、24時間周期として日付が変わる時系列データのグラフ描画には、周期境界の処理で骨が折れる場合が多い。というのも、この24時間周期の問題を放置すれば、幾重にも塗りつぶされた多価関数グラフの様相を呈する。

さて、時系列データセットには、次のようなフォーマットのテキストデータを `gnuplot` で可視化（折れ線グラフ描画）していくことを考える。

```
# 年月日 時刻 bit price ask price
2021/07/30 16:06:57 4366081.3482 4367501.0
2021/07/30 16:07:04 4366081.3482 4367655.0
2021/07/30 16:07:34 4364624.4939 4366551.0
# 中略
2021/07/30 16:07:55 4364452.5111 4367312.0
2021/07/30 16:08:01 4364452.5111 4367592.0
```

図1. 時系列データの形式例（本報ではファイル名を `output.txt` として話を進めている）

なお、参考ではあるが、このようなフォーマットのデータセットをテキストファイル（ファイル名を `output.txt`）に出力する Python プログラムは次のようにコーディングできる。

```
fout = open('./output.txt', 'a+')
timenow = datetime.datetime.now()
print(timenow.strftime('%Y/%m/%d %H:%M:%S'), bid_price, ask_price, file = fout)
```

ここで、コード中の変数 `ask_price` と `bid_price` には、それぞれ、取引所 `bitFlyer` からリアルタイムで取得した売値と買値の実数値を格納する変数である。年月日と時刻そして買値と売値をテキストファイル（`output.txt`）に落としている。この時系列データを `gnuplot` にてリアルタイムに取得したデータのグラフを実時間で描くことを考える。

さて、コマンドライン（コマンドプロンプト）から読み込ませる `gnuplot` 用のバッチファイルを次のように作成する（ファイル名 `gpl.cmd`）。なお、事前に Windows 環境変数に `gnuplot.exe` が実行できるよう `PATH` を通しておく必要がある。

```
# gnuplot 用のバッチファイル gpl.cmd
set xdata time
set timefmt "%Y/%m/%d %H:%M:%S"
set format x "%H:%M:%S"
set grid
set ytics nomirror
set y2tics
set tics out
set autoscale y
set autoscale y2
#グラフ中の文字のフォントはお好みで
set xlabel font "Arial,10"
set y2label font "Arial,10"
set ylabel font "Arial,10"
set tics font "Arial, 10"
set key font "Arial,10"
# tail コマンドの実行
!tail.bat
plot 'tmp.txt' using 1:3 t 'pl 1' w l l
w 2 lc rgb "red" axes x1y2, 'tmp.txt' u
sing 1:4 t 'pl 2' w l lw 2 lc rgb "blue"
axes x1y2
pause 0.5
reread
```

図2. `gnuplot` のバッチファイル（ファイル名を `gpl.cmd` として話を進めている）



図3 コマンドラインから gnuplot の実行

このとき、図3のようにWindowsコマンドプロンプトからバッチファイル `gpl.cmd` を `gnuplot` で実行すればよい。図2にあるバッチファイル `gpl.cmd` のスクリプトに少し解説を加えるならば、

```
set xdata time
set timefmt "%Y/%m/%d %H:%M:%S"
set format x "%H:%M:%S"
```

とすることで、時系列データの入力フォーマット西暦/月/日 時刻(日のあとは空白)と合わせ、グラフ横軸として時:分:秒と表示させるように設定できる。なお、

```
# gnuplot 用のバッチファイル gpl.cmd
```

など、行の初めに#を付すと、その行はコメントとして処理される。そして、次の行、

```
set grid
```

はグラフを見やすくするため、グラフ背面にグリッド補助線を描くための命令である。さらに、

```
set ytics nomirror
set y2tics
set tics out
set autoscale y
set autoscale y2
```

とすることで、縦軸バーの数値表示位置が右側となり、売値と買値の数値レンジを `gnuplot` のほうで時々刻々と自動調整してくれる。時系列データの更新プロット位置はグラフ右側先端の縦軸付近にあるため、こうしたほうが目視で確認しやすいからである。そして、エクスクラメーション・マークから始まる次のスクリプト、

```
# tail コマンドの実行
!tail.bat
```

については後述するが、Pythonプログラムの動作によってリアルタイムに出力されるファイル(ここでは `output.txt`) から、グラフ描画したい時系列数の分量にまで縮小したデータセットを取り出すスクリプトである(後述)。なお、このようにして取り出したデータセットのファイル名を `tmp.txt` とした。実時間でデータセットを取得したとき、そのデータ容量は際限なく増大していくため、グラフ描画時に古いデータは切り捨てていく。そのため、`gnuplot` のバッチファイルでは、エクスクラメーションのあとに実行したいスクリプトのファイル名を記述している。そして、

```
plot 'tmp.txt' using 1:3 t 'pl 1' w l lw
2 lc rgb "red" axes x1y2, 'tmp.txt' using
1:4 t 'pl 2' w l lw 2 lc rgb "blue" axes
x1y2
```

であるが、縮小されたリアルタイムデータファイル `tmp.txt` を呼び出して折れ線グラフにしている(`plot` コマンドの詳しい使い方は別途書籍^⑥を参考にしてください。ざっくりといえは折れ線の色や太さを設定しています)。そして、バッチファイル `gpl.cmd` の最後の二行

```
pause 0.5
reread
```

では、0.5秒間隔でグラフ描画の繰り返し(`reread`)処理をしている。本邦ではこの時間間隔を0.5秒としたが、これは時系列データの性質と描画のリフレッシュ速度に合わせて秒数を調整されたい。

先ほど `gnuplot` のバッチファイルのなかで用いた更なるバッチファイル `tail.bat` であるが、Linuxではおなじみの `tail` コマンドをWindows上で使うためのものである。この `tail.bat` では単に

```
tail.exe -1000 output.txt > tmp.txt
```

とし、Pythonから出力される `output.txt` から、`reread` 毎に折れ線グラフを描き直すのに必要な分量のデータ(`output.txt` のテキストファイル中最下行から上1,000行まで)を、`gnuplot` の繰り返し処理ごとに逐次テキストファイル `tmp.txt` に上書き処理している。さて、先の説明では飛ばしていた、`tail.bat` の中身を見てみよう。

```
tail.exe -1000 output.txt > tmp.txt
```

tail コマンドは、Linux ではおなじみの便利なコマンドで、ファイルの最終行から（指定した）数行を表示することができる。しかしながら、残念なことに、Windows では tail コマンドは使えない。この tail コマンドはもとより、ほとんどの Linux コマンドが使えないため、MSYS2 (Software Distribution and Building Platform for Windows) のと MinGW (Minimalist GNU for Windows) をインストールして、Linux コマンドが使える Windows 開発環境を整えておく必要がある。

3. ビットコインの売り気配／買い気配の取得

暗号通貨取引所である bitFlyer (ビットフライヤー) から供されている API を用いてビットコイン価格を取得する Python プログラムについて、ざっばくではあるが、コード作成について解説する。CCXT 経由で、bitFlyer の API を使うので、かならず、

```
import ccxt
import json
```

を Python コードの最初にインポートする必要がある。API から取得するデータはほとんどが JSON 形式なので、2 行目もあったほうが便利である。次に、関数 (メソッド) を作る。

```
# bitflyer API
ex_bif = ccxt.bitflyer({'timeout': ErrorTimeOut})
ex_bif.apiKey = '*****'
ex_bif.secret = '*****'
```

ここで、apiKey と secret キーに関しては、自前のものを使っていただく必要があるため、それら文字列は隠してある。引数 timeout に対応する ErrorTimeOut の値、各自で最適と思われるの時間刻み量を設定していただきたい。この HTTP API は、呼出回数を制限され、上限に達すると呼出を一定時間ブロックされる。また、ブロックの解除後も呼出の上限を一定時間引き下げられることにも各自ご注意ください。

次に、ビットコインのいわゆる「板」を取得する Python コードの抜粋である。

```
# bitflyer の板取得
orderbook_bif = ex_bif.fetch_order_book('BTC/JPY')
ask_price = orderbook_bif['asks'][0][0] if len(orderbook_bif['asks']) > 0 else None
bid_price = orderbook_bif['bids'][0][0] if len(orderbook_bif['bids']) > 0 else None
```

それでは、以上に解説した方法を使って、実際に gnuplot で描画したビットコインの売値 bid と買値 ask の折れ線グラフを見てみよう (図 4)。



(a) 24 時手前の折れ線グラフ



(b) 24 時をまたいだ直後の折れ線グラフ

図 4. リアルタイムに取得したビットコインの売値と買値の折れ線グラフ

図 4 の (a) と (b) はそれぞれ 24 時の時刻をまたぐ直前と直後の折れ線グラフであるが、危惧していた問題は生じていない。また、縦軸の数値レンジも売値・買値の値に自動的にアジャストしながら繰り返し描画されていることがわかる。

4. おわりに

本報では、ビットコイン／仮想通貨取引所 bitFlyer が提供する HTTP API を用いることで Python コードから取得したビットコインの売値と買値の折れ線グラフを、gnuplot のバッチファイルに一工夫施すことによって、リアルタイムで描画した。24 時の日付またぎで危惧していた周期境界処理の問題も解消しており、gnuplot の繰り返し (reread) 処理でも数値に応じて縦軸バーの自動スケールリングを確認することができた。次は、リアルタイム・データサイエンスに向けたはじめの二歩、となるべき内容の報告をしたい。

参考文献

- [1] 塚本邦尊ら：東京大学のデータサイエンティスト育成講座 ~Python で手を動かして学ぶデータ分析~, マイナビ出版, 2019.
- [2] Jake VanderPlas (著), 菊池 彰 (翻訳) : Python データサイエンスハンドブック —Jupyter、NumPy、pandas、Matplotlib、scikit-learn を使ったデータ分析、機械学習, オライリージャパン, 2018.
- [3] 加藤 耕太 : Python クローリング&スクレイピング — データ収集・解析のための実践開発ガイド, 技術評論社; 増補改訂版, 2019.
- [4] 斎藤幸平 : 人新世の「資本論」, 集英社, 2020.
- [5] <http://www.gnuplot.info/>
- [6] 山本昌志 : gnuplot の精義 — フリーの高性能グラフ作成ツールを使いこなす, カットシステム; 第二版, 2013.
- [7] <https://www.msys2.org/>