

新米データサイエンティストのFXブログ

PythonでFXのトレンドラインを引いてみる

🕒 2018年8月15日 🔄 2018年12月21日

使っているFX会社はどこ？

自動売買用（1万円キャッシュバック中） → [インヴァスト証券](#)

Python API用（FX APIはここしかない） → [OANDA ジャパン](#)

FXのトレンドラインをPythonを使って引く方法のまとめ。為替レート of データを処理して抵抗線と支持線をチャートに表示させます

こんばんは、新米データサイエンティスト（@[algon_fx](#) [🔗](#)）です。今月はTwitterを使ったFX予想を頑張るという目標を掲げながらも、勾配ブースティングが楽しすぎて・・・までTwitterまで着手できていません（笑）

勾配ブースティングの初心者向けチュートリアルも、また後日更新する予定です。今日は別ネタで、為替レートを処理してトレンドラインをPythonで作る方法をまとめます。



新米データサイエンティスト
@algon_fx

前に少しブログでも書きましたが、やっぱり、為替レート of データには細かい線形回帰が非常に有効ですね。

言ってもトレンドラインって、やっぱり重要ですよ。

7 0:09 - 2018年8月15日

[新米データサイエンティストさんの他のツイートを見る](#)

上記でツイートした通りですが、機械学習トレード（機械学習を活用したFXトレードを勝手に命名）では、線形回帰が思った以上に役に立ちます。

そもそもFXトレードの原則はやはりトレンドな訳であり、ニューラルネットワークのような複雑なアルゴリズムで為替レートを学習するのも重要ですが、**単純に線形回帰でトレンドをしっかり把握するのも重要**です。（参考：[TensorFlowでニューラルネットワークを使ってFX予想をする方法](#)）

目次 [閉じる](#)

1. そもそもトレンドラインとは？
2. STEP1 ライブラリとデータのインポート
3. STEP2 データの簡単な前処理
4. STEP3 上昇トレンドラインを算出
5. STEP4 下降トレンドラインを算出
6. STEP5 チャートに落としてみる
7. まとめ

｜ そもそもトレンドラインとは？

トレンドラインはテクニカル指標の最も初歩的な指標と言っても過言ではありません。トレンドラインには「上昇トレンドライン」と「下降トレンドライン」の**2種類**あります。

上昇トレンドラインとは、ローソク足の安値同士を結んで引く線です。対して下降トレンドラインは高値同士を結んで線を引きます。

チャートを見ると一目瞭然です。下の赤い線が上昇トレンドラインです。赤丸の部分の安値を基軸として、直線を引いたものです。一目瞭然でレートが上昇トレンドとなっているのが確認できます。



対して緑色が下降トレンドラインです。これは緑丸の高値2点を直線で結んだもので、その後のレートが下降トレンドを示しているのが解ります。

トレンドラインですが、冒頭でも話した通り最も初歩的なテクニカル指標であり、トレードラインを参考にトレードしている方は非常に多いです。そのため、トレンドライン付近では売買が盛んに行われるため、結果としてトレンドラインにある程度協調したレートが形成されていく訳です。

構造はとってもシンプルなトレンドラインですが、Pythonを使って実際に計算を試みましょう。

STEP1 ライブラリとデータのインポート

まずはいつも通りライブラリとデータをインポートしましょう。今回も私がAPI経由から取得したドル円の1日足のレートを使います。

データは下記のURLからCSVファイルをダウンロードできます。

[usd_jpy_api.csv](#)

まずは基本的なライブラリをインポートしてあげましょう。

```
1 # ライブラリのインポート
2 import pandas as pd
3 import numpy as np
4 import matplotlib.pyplot as plt
5 from scipy.stats import linregress
6
7
```

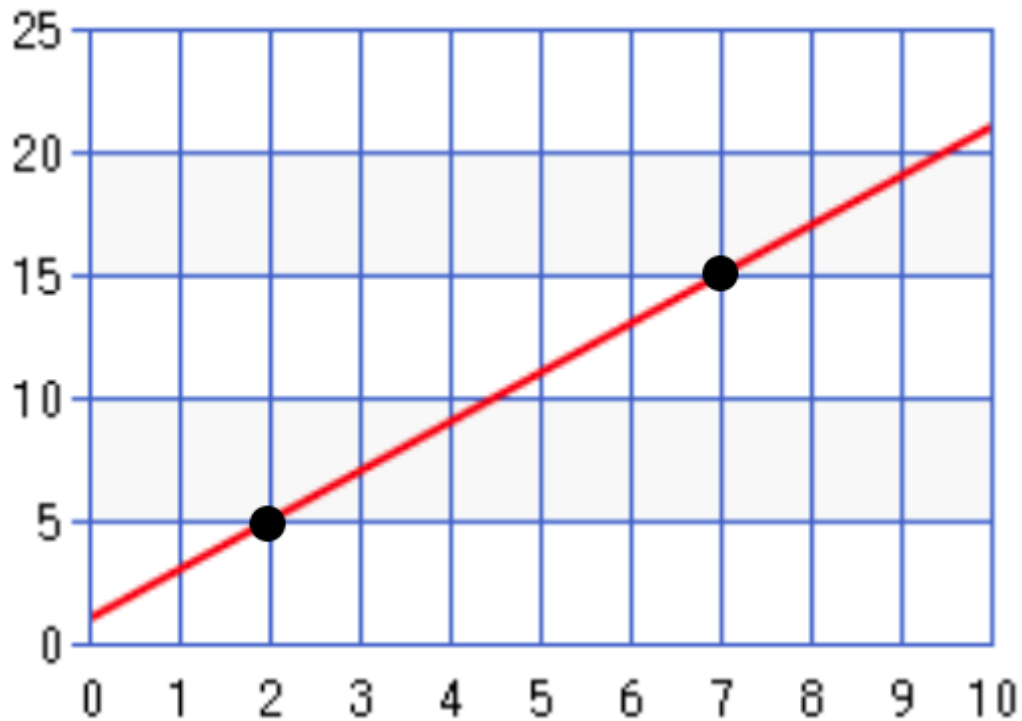
トレンドラインの処理のために、scipy.statsのlinregressをインポートしています。scipy.statsとは、**統計関数をまとめたモジュール**でちょっとした処理を行うときは重宝します。

また、今回のトレンドラインを引く主役とも言えるのが「linregress」です。これはXとYの値を渡すと、単回帰直線を求めてくれる非常にハンディーな関数です。

単回帰直線とは、与えられた2つのデータポイントを通る直線をさします。下の図を見てください。これは(2, 5)のデータと(7, 15)のデータです。

この二つを通る直線ですが、 $y = 2x + 1$ と求めることが可能です。xに掛かっている「2」は直線の傾きを表しています。実際に下の直線の傾きを計算してみると「2」が出てくると思います。

対して「+1」はY切片を表しており、これは単純に直線がY軸と交わる時のXの値です。下の図をみると赤い直線がY軸と交わっている部分が1なのが解ります。



linregress()はXとYを渡すことで、この直線を導き出してくれます。簡単にPythonを使ってやって見ましょう。まずは上の黒丸の2つのポイントをNumpy配列で作ります。

```
1 x = np.array([2, 7])
2 y = np.array([5, 15])
3
4
```

では、この2つのポイントの直線を導き出してみましょう。

```
1 sample_lin = linregress(x = x, y = y)
2 print(sample_lin)
3
4 -- 出力
5 LinregressResult(slope=2.0, intercept=1.0, rvalue=1.0, pvalue=0.0, stderr=0.0)
6
7
```

ご覧通り、slope（傾き）が2.0、intercept（Y切片）が1.0と出ています。では、上の表にはありませんが、仮にXが10とした時のYの値を直線を使って計算をしてみましょう。

```

1 # Xが10の時のYは?
2 z = 10
3 sample_lin[0] * z + sample_lin[1]
4
5 -- 出力
6 21.0
7
8

```

21と算出できました。上のグラフで見ても、Xが10の時のYはおおよそ21の付近になるのが確認できます。

すでに勘が鋭い方であれば気づいたかもしれませんが、**単回帰直線の傾きはトレンドの強さ**を表しているのが判ると思います。トレンドが強ければ強いほど、直線の傾きは増えます。

前奥が長くなりましたが、次はCSVファイルを読み込んでデータフレームに変換してあげます。

```

1 # CSVファイルから過去レートを読み込む
2 df = pd.read_csv("../tensorflow/usd_jpy_api.csv")
3
4 # データの最初の5行を表示
5 df.head()
6
7

```



```

1      time      close  open   high    low volume
2 0  2016/08/19 06:00:00 100.256 99.919 100.471 99.887 30965
3 1  2016/08/22 06:00:00 100.335 100.832 100.944 100.221 32920
4 2  2016/08/23 06:00:00 100.253 100.339 100.405 99.950 26069
5 3  2016/08/24 06:00:00 100.460 100.270 100.619 100.104 22340
6 4  2016/08/25 06:00:00 100.546 100.464 100.627 100.314 17224
7
8

```

こちらのレートデータはドル円の1日足のデータを私が綺麗にしたものです。為替データの取得を自分でして見たい方は、OANDA APIがオススメです。（参照：[Pythonを使ったOANDA API v1で知っておきたい10の基本操作](#)）

STEP2 データの簡単な前処理

データの前処理と言うほどでもありませんが、簡単な処理が必要です。上の単回帰直線でもやりましたが、直線を導き出すにはXとYの値が必要です。

当然ではありますが、Yは為替レートです。上昇トレンドラインの場合のYは「安値（low）」、逆に下降トレンドラインでは「高値（high）」を使います。

では、Xの値はどうすれば良いでしょうか？色々な算出方法がありますが、最も単純に日付ごとに1を増やした値をXとしましょう。

こちらの為替レートのデータは古い日付から新しい日付でソートされていますので、単純にインデックスに1を加えることで、1日目、2日目・・・といった具合の数値を作れます。

```
1 # 日付スタンプ
2 df['time_id'] = df.index + 1
3 df.head()
4
5
```

```
1 -- 出力
2 time      close   open    high    low volume  time_id
3 0  2016/08/19 06:00:00 100.256 99.919  100.471 99.887  30965    1
4 1  2016/08/22 06:00:00 100.335 100.832 100.944 100.221  32920    2
5 2  2016/08/23 06:00:00 100.253 100.339 100.405 99.950   26069    3
6 3  2016/08/24 06:00:00 100.460 100.270 100.619 100.104  22340    4
7 4  2016/08/25 06:00:00 100.546 100.464 100.627 100.314  17224    5
8
9
```

ご覧通り、データの1日目のtime_idは「1」、それ以降は1ずつ増えているのが確認できます。続いて、上昇トレンド、下降トレンド、表示用のデータを切り分けておきましょう。

```
1 # 元データを目的別に切り分ける
2 df_fin = df.copy()
3 df_high = df.copy()
4 df_low = df.copy()
5
6
```

STEP3 上昇トレンドラインを算出

まずは上昇トレンドラインを算出します。前述した通りscipy.statsの統計モジュールにあるlinregress()＝単回帰直線を使って算出します。

```

1  # 高値のトレンドライン
2  while len(df_high)>3:
3
4      reg_1 = linregress(
5          x = df_high['time_id'],
6          y = df_high['high'],
7      )
8      df_high = df_high.loc[df_high['high'] > reg_1[0] * df_high['time_id'] + reg_1[1]]
9
10 reg_1 = linregress(
11     x = df_high['time_id'],
12     y = df_high['high'],
13 )
14
15 df_fin['high_trend'] = reg_1[0] * df_fin['time_id'] + reg_1[1]
16
17

```

コードを見ていただくと解りやすいですが、高値のポイントを計算して、high_trendとしてトレンドラインを算出しています。

STEP4 下降トレンドラインを算出

同じ要領で下降トレンドラインも算出しましょう。上昇トレンドラインでは高値（high）を使って算出しますが、過去トレンドラインは安値（low）を使っている部分に注意してください。

```

1  # 安値のトレンドライン
2  while len(df_low)>3:
3
4      reg_2 = linregress(
5          x = df_low['time_id'],
6          y = df_low['low'],
7      )
8      df_low = df_low.loc[df_low['low'] < reg_2[0] * df_low['time_id'] + reg_2[1]]
9
10 reg_2 = linregress(
11     x = df_low['time_id'],
12     y = df_low['low'],
13 )
14
15 df_fin['low_trend'] = reg_2[0] * df_fin['time_id'] + reg_2[1]
16
17

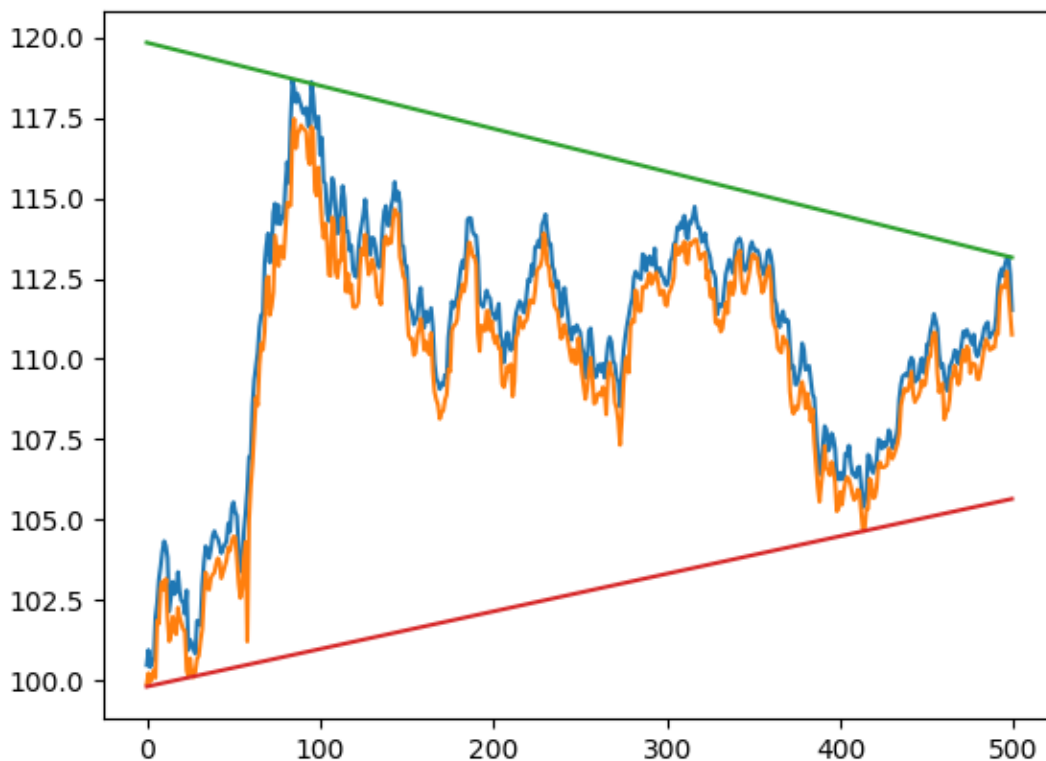
```

これで上昇・下降トレンドラインの算出が完了です。

STEP5 チャートに落としてみる

では、正しくトレンドラインが算出できているか簡単なチャートに落として確認してみましょう。

```
1 #トレンドラインをプロット
2 df_fin['high'].plot()
3 df_fin['low'].plot()
4 df_fin['high_trend'].plot()
5 df_fin['low_trend'].plot()
6
7
```



めちゃくちゃ手抜きコードでごめんなさい（笑）ちゃんとローソク足でトレンドラインチャートを作りたい方は、[Pythonでローソク足チャートを作るこちらの記事](#)をご参照ください。

手抜きではありますが、意図した通り上昇トレンドライン（赤）と下降トレンドライン（緑）がしっかり算出されているのが確認できます。

それではそれぞれのトレンドラインの傾きを確認してみましょう。

```
1 print(reg_1)
2 print(reg_2)
3
4

1 LinregressResult(slope=-0.013401394548201259, intercept=119.86218229940955, rvalue
2 LinregressResult(slope=0.011708465009043387, intercept=99.791812791922567, rvalue=
3
4
```


上昇トレンドラインの傾きは0.0134、下降ラインは0.0117と出ています。加えてY切片（intercept）の値も確認が可能です。

まとめ

今回は非常にシンプルではありますが、為替レートからPythonを使ってトレンドラインを算出する方法をまとめました。

トレードをやっている方であれば、上級者でも初心者でもトレンドラインをひとまずは参考にして決断をしている方も多くいらっしゃるかと思います。

機械学習トレードでも、トレンドラインをその都度算出することで役に立つことが多くあります。参考までにですが、私は様々な期間（例：1分足、5分足、10分足、1時間）など期間別にトレンドラインを算出して使っています。

以上となります！ブログ読んでいただきありがとうございます！[Twitter](#)  でも色々と発信しているので、是非フォローお願いします！

【人気記事】

→[私の機械学習の開発環境&トレード環境](#)

使っているFX会社はどこ？

自動売買用（1万円キャッシュバック中） → [インヴァスト証券](#)

Python API用（FX APIはここしかない） → [OANDA ジャパン](#)