

## 一、研究背景與動機

決定匯率的因素有非常多種。在基本面上，可以是即期利率、遠期利率、通貨膨脹率、失業率、非農就業人口等等；另外一方面，技術面也有移動平均線、KD 指標、平滑移動平均線 (MACD)、相對強弱指數 (RSI)。許多專家及學者<sup>1</sup>認為，技術指標是根據歷史股價研究出的一套反映投資人心理的理論，因此能夠更準確預測匯率的短期走勢；相對的，基本面則是透過分析幾項外匯市場長期發展的幾項指標，來預測中長期的匯價發展。

其中，技術分析仰賴投資者的計算及良好的投資策略，在自動化交易上，我們時常仰賴 Python 程式語言。Python 的套件隨著時間演進，經過更多工程師的貢獻，越趨完整，可以幫助我們進行各項種類的資料分析，例如時間序列分析、文字探勘等等。而我們在這學期學習各種 Numpy、Scipy、Pandas 套件下的功能，也讓我能夠更順利的進行交易策略的撰寫。

以下我主要會透過 Numpy 及 Pandas 計算匯率在技術分析上的各項指標，並且建構自己的交易策略，並進行回測的動作。最後也了解到，如果單純將自己的交易策略及邏輯以程式碼的方式寫成 Python 的程式碼，未必能夠通用於所有情境，因此也希望透過 scikit-learn 的 RandomForestClassifier 分析匯率漲跌幅。

總的來說，以上兩種交易策略在進行簡單的回測時，都能為我賺一點小錢，然而策略上還只是最小可行性方法 (MVP)，皆有改進的空間。

## 二、資料與資料來源

### 1. 各國匯率：

- (1) 資料種類：台幣及美元、歐元、英鎊、日幣、人民幣等五大特別提款權 (SDR) 貨幣之間的匯率
- (2) 資料區間：自 2000 年 1 月 1 日至 2021 年 10 月 31 的匯率日資料
- (3) 資料來源：Investing.com
- (4) 資料集：參見附錄 (1)

2. 技術面指標：利用資料來源為前述的各國匯率，並將 Python 的 Numpy 及 Pandas 套件計算各技術指標的公式解，求得 2000 至今的技術指標數據，後續將會依序介紹這份研究使用的技術指標。

---

<sup>1</sup> The Deutsche Bank Guide of FX determination

### 三、資料前處理 (DataLoader.CurrencyDataLoader 建議程式碼參閱)

#### 1. 台幣間接匯率 `indirect(df, columns)`：

由於從 Investing.com 下載下來的匯率資料，我們使用的是各國貨幣與美元之間的匯率，為了方便將台幣當做一種商品來計算台幣價格，我們首先將台幣匯率進行倒數。

#### 2. 計算日資料的報酬率 `price_pct_change(df)`：

原始資料及僅包括收盤價、開盤價、最高點、最低點，我們透過 Pandas 簡單計算日資料的報酬率。

#### 3. 建立各類貨幣重新命名的功能 `rename_currency_data(df, symbol)`：

為了後續程式進行的方便，利用 `Pandas.DataFrame.rename()` 的功能，將各種貨幣的匯率資訊，依照外匯代號 (Symbol) 進行區分

#### 4. 合併匯率資料表 `dataframe_combine(df, currency_list)`：

建立功能，將原始的 USD\_TWD 的資料表，與使用者想要合併的所有貨幣的資料表進行合併，並回傳合併資料表。其中，由於各類貨幣市場交易的日期可能略有不同，因此在進行合併資料表時，我使用 Left join 的方式對齊使用者提供的原始資料表，若再出現空值，則簡單的以內插法補值。

### 四、技術分析指標(DataLoader.TechnicalAnalysis)

#### 1. RSI 相對強弱指標 `calculate_RSI(df, symbols)`：

相對強弱指標長以 7 天或 14 天計算商品價格的強弱度，顯示投資者與匯價的投資情緒，匯價無法保有無限的漲幅或無限的跌幅，當匯價持續漲到一定的水位後，便會有壓力讓價格下跌，反之亦然，因此我們在匯率的交易策略設定一定的門檻值，便於進場或出場。RSI 指標公式如下：

$$RSI = \frac{\text{期間內的絕對漲幅}}{\text{期間內的絕對漲幅} + \text{期間內的絕對跌幅}} * 100$$

數值介於 0 至 100 之間接近 80 時適合放空 (Sell)；RSI 指標接近 20 時適合做多(Buy)。

#### 2. KD 指標 `Calculate_KD(df, symbol)`：

KD 指標是由 George C. Lane，並於其著作說明：「他觀察上漲時，當日收盤價總是朝向當日價格波動最高價接近；反之，當股價下跌時，當日收盤價總是朝向當日價格波動最低價接近。」在這項指標表示，以當天標的的價格與選

定前 N 天(window)比較，標的價格之強弱勢，常用的策略中，window 設置為 9 天。其公式如下：

$$RSV = \frac{\text{今日收盤價} - \text{近 } N \text{ 天最低價}}{\text{近 } N \text{ 天最高價} - \text{近 } N \text{ 天最低價}} * 100$$

$$K \text{ 值} = \frac{2}{3} * \text{昨天的 } K \text{ 值} + \frac{1}{3} * \text{今天的 } RSV \text{ 值}$$

$$D \text{ 值} = \frac{2}{3} * \text{昨天的 } D \text{ 值} + \frac{1}{3} * \text{今天的 } K \text{ 值}$$

主要比較 K 值與 D 值之間指標的變動，K 值又稱快線，D 值又稱慢線，當 K 值 > D 值則為上漲行強，適合作多；相對地，K 值 < D 值則為下跌行情，適合做空。

### 3. 移動平均數 moving\_average(df, symbols)：

計算一段時間長短的匯價平均數，策略中這訂為 5 天，捕捉極短期的平均價格，當價格高於 5 日平均線時，適合進場 (Buy)；反之則適合退場 (Sell)

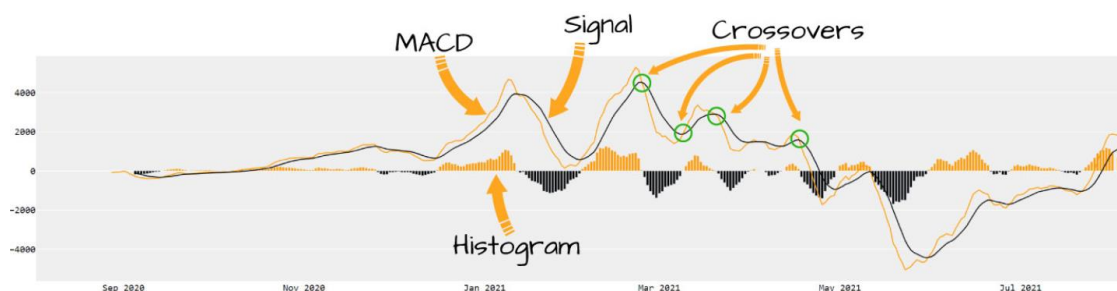
### 4. 指數平滑異同平均線(MACD) calculate\_MACD(df, symbols)：

屬於均線理論的延伸，以指數平滑的方式確立中長期的波段走勢，計算公式中有許多參數需要計算，

(1) MACD: 26 天的指數平均 - 12 天的指數平均

(2) Signal: 短期的指數平均，常用為 9 天

(3) Difference: MACD - Signal，差值可以求出漲跌幅的趨勢，正向代表匯價未來看漲，負向則看跌。



資料來源：Alpharithms.com

圖（一）MACD 示意圖

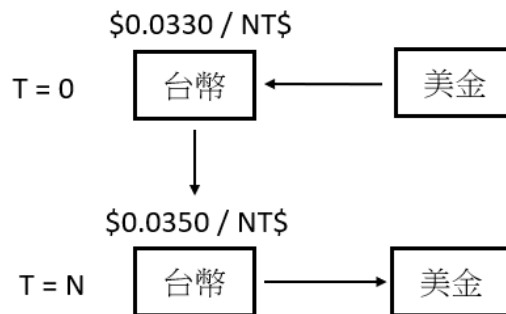
## 五、交易策略與回測

我的交易策略分為兩種，一種是簡單的使用 Pandas 運算，並設立技術指標的買賣點，幫助我進行套利。而第二種則是將時間序列的資料分成訓練資料集(Training dataset)以及測試資料集(Testing dataset)，利用 Scikit-Learn 的 RandomForest 根據訓練資料集的匯價漲跌(Binary)建立模型，再使用測試資料集評估模型的準確性，並進行回測，進而達到獲利，為了計算簡便，將簡單的以美元兌台幣進行套利的示範。

### 策略一：根據常規設立技術指標買賣點

#### 1. 套利邏輯：

本策略利用台幣的間接匯率作為匯價的表達方式，原因在於使用間接匯率，投資人可以直接將貨幣視為一種商品，判斷漲跌幅上會比較快速一點。若是投資人預期台幣升值，如下圖（二）所示，我們則可以在第  $T=0$  期的時候就先買入數單位的台幣，於第  $T=N$  期賣出，便可以從中賺取匯差。而我們的策略一也是圍繞在這個邏輯進行套利。



圖（二）預期台幣升值的交易策略

#### 2. 利用 Pandas 設立技術指標買賣點：

具備套利邏輯後，我們則是要思考什麼時間點是適合進場( $T=0$ )，以及什麼時間點適合出場( $T=N$ )。以下將透過技術分析指標判斷進出場時刻。

程式碼針對測試資料集 (2017-01-01 ~ 2021-10-29) 的美元兌台幣進行技術分析的交易策略。

##### (1) 設立 RSI 強度 `strategy_for_rsi(df, symbol)`：

當 RSI 指標介於 70 ~ 80 之間時，設立價格強度為 1，價格大於 80 時，價格強度為 2，此時價格水位偏高，預期未來匯價下跌，建議作空  
當 RSI 指標介於 20 ~ 30 之間時，設立價格強度為 -1，價格小於 20 時，價格強度為 2，此時價格水位偏低，預期未來匯價上漲，建議作多

##### (2) 設立 KD 強度 `strategy_for_kd(df, symbol)`：

當 K 線恰好突破 D 線，預期未來會有更大的漲幅，因此設立強度為 2，其餘 K 值大於 D 值的情況則設為 1；相對的，當 K 線恰好跌破 D

線，預期未來會有更大的跌幅，因此設立強度為 -2，其餘 K 值小於 D 值的情況則設為 -1。

(3) 設立 MA 趨勢 `strategy_for_ma(df, symbol)`：

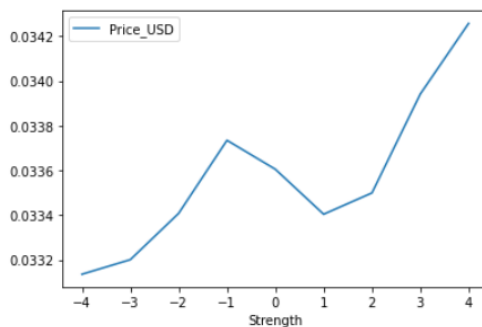
當匯價恰好由下往上突破 5 日均線，預期未來會有更大的漲幅，因此設立趨勢為 2，其餘匯價大於 5 日均線的情況則設為 1；相對地，當匯價恰好跌破 5 日均線，預期未來會有更大的跌幅，因此設立趨勢為 -2，其餘匯價小於 5 日均線的情況則設為 -1

(4) 設立 MACD 趨勢 `strategy_for_macd(df, symbol)`：

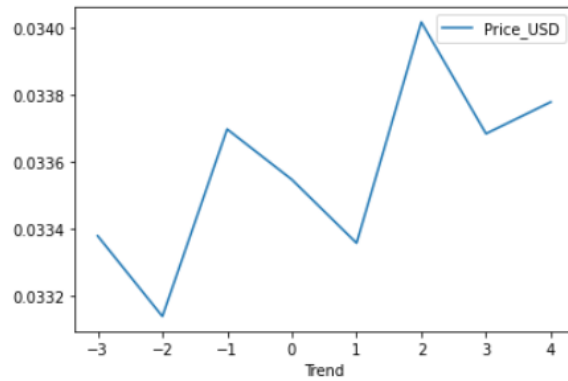
MACD 的趨勢屬於中長期的 MACD 與 9 日短期的訊號 (Signal) 進行相減的值，當差值 (Difference) 正好由負轉正時，代表短期的價格優於中長期價格，預期未來還會有更大的漲幅，則趨勢設為 2，其餘設為 1；相反地，當由正轉負則將趨勢設為 -2，其餘設 -1

由以上的步驟，我們可以求得單一貨幣的貨幣強度(Strength)及價格趨勢(Trend)，若將貨幣強度的兩項指標進行相加，則強度介於 -4~4 之間；同樣的，若將趨勢的兩項指標進行相加，則趨勢介於 -4~4 之間，我們將從中找到適合的進出場時刻。

若進行價格相對於強度與趨勢之間的繪圖，如下圖（二）、圖（三）所示，可以發現強度與匯價大致成正相關。而為求買低賣高進行貨幣的套利，當我們的匯價強度很高時，則買進貨幣，當強度及低時，則買進貨幣。同樣的邏輯也呈現在匯價的趨勢上。



圖（二）匯率與其強度成正相關



圖（三）匯率與趨勢成正相關

### 3. 買賣的交易量：

從上圖（二）、圖（三）也可以觀察到，強度與趨勢越趨向於兩極，也就是 -4 與 +4，匯率會有強大的壓力進行漲跌幅，以下則利用這項發現去訂定買賣的交易量。

首先將強度及趨勢兩個指標於 -1 ~ 1 之間的訊息當作雜訊，僅保持持有 (Hold) 的狀態，不做買進賣出，其餘則依據強度與趨勢數值的高低進行做多 (Buy) 與作空 (Sell) 的動作，如下表示：

強度與趨勢	買進 / 賣出的交易量
4	作空 3,000,000 台幣，以現價買進美元
3	作空 2,000,000 台幣，以現價買進美元
2	作空 1,000,000 台幣，以現價買進美元
-2	持美元以現價買進 1,000,000 台幣，並在 n 天後換回
-3	持美元以現價買進 2,000,000 台幣，並在 n 天後換回
-4	持美元以現價買進 3,000,000 台幣，並在 n 天後換回

表（一）交易策略買賣量

### 4. 回測與獲利

在建立交易策略後，我利用第 1 點的套利邏輯，於 2017 年至 2021 年的時間區間進行回測，並設定獲利了結的時間  $n = 7$ ，代表每一次的套利行為將會以一周為一個循環。最終獲利為 324,312 美元。

```

@staticmethod
def back_test(df: pd.DataFrame, symbol: str, days: int) -> float:
    profits = 0.0
    for index, price in enumerate(df["Price_"+symbol].tolist()):
        if index < len(df["Price_"+symbol].tolist()) - days:
            profits += (df["buy_sell"].tolist()[index] / price *
                        df["Price_"+symbol].tolist()[index + days] - df["buy_sell"].tolist()[index])
    return profits

# determine buy or sell
tec_df = TechnicalStrategy.buy_sell(tec_df)
#print out total profits for Technical analysis
print("Total profits for Technical analysis:" + str(round(TechnicalStrategy.back_test(tec_df, "USD", 7), 0)))

if __name__ == '__main__':
    main

```

Total profits for Technical analysis: 324312.0

圖（四）技術分析策略回測與獲利

後續我也使用了 Scipy 提供的 Optimization.minimize 負向的最小化（也就是最大化）利潤，最佳化出場的時間點，優化獲利。

## 策略二：RandomForest 演算法判斷進場點

1. 首先我們將 2000 年至 2021 年 10 月底的所有匯率資料依據 8:2 的方式分成訓練資料集以及測試資料集，再以利用訓練資料集建立隨機樹 (RandomForest) 預測漲跌的模型。特別說明使用隨機樹的原因在於，我在進行策略一的時候，其實發現利用技術指標來判斷漲跌是具有一定規則的，例如 RSI 應屆於 20~80 之間，否則會出現漲跌的壓力等等，這樣的運作模式其實與隨機樹很像，且隨機樹是所有機器學習模型應用於本資料表現最好的一個。
2. 套利的運作模式，是先以隨機樹預測測試資料集（2017 年至 2021 年的匯率日資料）的漲跌，並依據漲跌做美元兌台幣的套利，每次進場單位為 1,000,000 美元，在一周 7 天為間距的套利獲利為 231,157 美元。
3. 缺陷：在進行測試資料集的評價(evaluation)時，我發現了這個模型的精準度 (Accuracy) 與召回值 (Recall) 是相對低的，代表模型表現較差，而如果使用找尋最大獲利點則會發現是在間隔 275 天進行套利，獲利為 1000 萬元，模型解釋上是不合理的。

	Accuracy	AUC	Precision	Recall	F1	TPR	FNR
values:	0.496437	0.494078	0.482906	0.364516	0.415441	0.364516	0.635484

表（五）隨機樹模型評價

我認為本次使用模型的缺陷在餘，我只有利用技術分析指標捕捉價格，而沒有去查看「交易量」來支撐預測出的漲幅及跌幅。

## 五、金融計算程式使用到的技術：

1. Numpy
2. Pandas & datetime
3. Scipy: Constrained minimization
4. scikit-learn: Random Forest Classifier
5. Python object oriented programming

## 六、未來展望（可以優化的部分）：

1. 資料來源：這次的資料集是手動下載 Investing.com 的歷史資料，若是未來想要擴大成更大的專案，可以串接匯率的 API，市面上 Yahoo finance 與 OANDA 皆有提供 API 串接。
2. 捕捉各國匯率之間的相關性：各國之間的股市、匯市其實多有連動性，譬如美股就時常影響的台股的走向，可以利用各個交易市場之間的時間差來進行套利，這部分也呼應到動態串接 API 的優點，可以將交易策略的時間長短縮減至奈秒 (ns) 的套利
3. 交易策略的優化：這次使用的交易策略以及變數算是非常非場簡單，可能有欠擬和的疑慮，在技術分析指標上，可以再加上「交易量」相關的指標，如此一來能夠在漲跌的預測上更能夠支撐論點。除此之外，在資訊的技術的部份，也可以使用深度學習的 LSTM 進行匯價預測，LSTM 特別適合處理股價、匯價等時間序列的資料。

## 七、附錄：

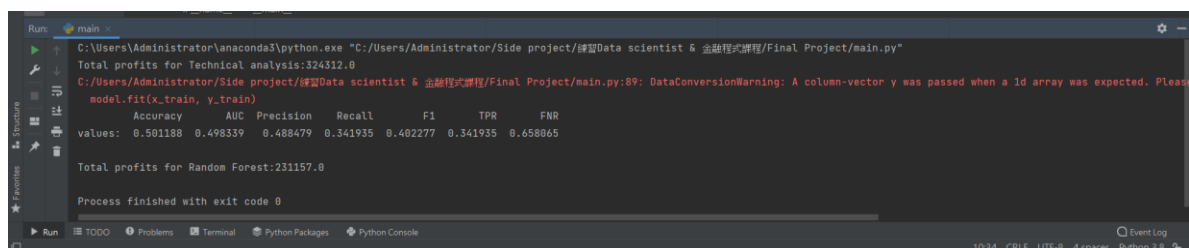
1. 資料集：

[https://drive.google.com/drive/folders/1DrDKy5P3Au6c\\_5AqAvV7n2VWlJzwBINo?usp=sharing](https://drive.google.com/drive/folders/1DrDKy5P3Au6c_5AqAvV7n2VWlJzwBINo?usp=sharing)

2. 程式碼：

<https://drive.google.com/drive/folders/17Ez4J0JkqsTAwKVtg4Tjg-uWzMGzGJtP?usp=sharing>

3. 完整 Pycharm Output 截圖：



```
Run: main
C:\Users\Administrator\anaconda3\python.exe "C:/Users/Administrator/Side project/練習Data scientist & 金融程式課程/Final Project/main.py"
Total profits for Technical analysis:324312.0
C:/Users/Administrator/Side project/練習Data scientist & 金融程式課程/Final Project/main.py:89: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please
model.fit(x_train, y_train)
Accuracy    AUC    Precision    Recall    F1    TPR    FNR
values: 0.501188 0.498339 0.488479 0.341935 0.402277 0.341935 0.658065
Total profits for Random Forest:231157.0
Process finished with exit code 0
```