

0811253 程珣 期末報告

● 報告介紹

Youtube 連結: <https://youtu.be/eW-TOWYTIO4>

報告主要使用了 `yfinance` 取得股票資訊，`numpy` 做部分計數運算，`matplotlib` 做畫股票趨勢圖部分，`datetime` 取出當天日期印於圖片上

主要資料：`^IXIC`, `TSLA`, `AAPL`, `AMZN`, `NVDA`, `MSFT`

資料來源：Yahoo Finance

程式主要功能：

啟動程式後會輸出依畫面，包含 6 支股票的 當天 1 分 K 走勢、5 分 K 均線、成交量，在成交量輸出部分，若分 K 的成交量為當天平均成交量的 3 倍以上，輸出為深色，其餘成交量則輸出為淺紅 or 淺綠色

程式參數：

可自由設定要取得哪六支股票

可自由設定要幾分均線，均線顏色、線寬

成交量的深淺參數可以自由設定(大於幾倍，要輸出什麼顏色)

程式應用：

可將 6 支相關類股放在一途上記錄當天盤面變化，有助於紀錄、觀看的方便性，在圖片上可利用成交量的深淺來看出大戶在進出的時候後是盤面的變化(`AAPL`、`MSFT` 的爆量常常是盤面的相對高點低點)，6 支股票排列可觀察哪個時段由哪支股票來帶領盤面。

● 程式介紹

程式大綱：

參數設定

下載當天股票資訊

整理股票資訊

畫圖

設定畫布

消除 xY 軸刻度

畫趨勢圖

均線

畫成交量

畫整體時間線切割

主要程式：

```
import yfinance as yf
import numpy as np
import matplotlib.pyplot as plt
from datetime import date, timedelta

# 可以觀看當天的六支股票走勢比對圖
# 走勢 = 1 分線
# 可設定 1. 六支股票 2. 均線 3. 大量成交量顏色，大量倍數
# 3 條時間軸，分別為 10:00 、 10:30 、 14:30

# 參數設定 (股票、均線、成交量)
# 設定要看哪六隻，由上至下
stock = ['^IXIC', 'TSLA', 'AAPL', 'AMZN', 'NVDA', 'MSFT']
# 均線參數設定
# ma, 線寬 顏色
pa = [5, 0.5, 'orange']
# 成交量參數設定 (當成交量 > 平均量兩5 倍 畫成 深綠、深紅)
# 深綠 淺綠 深紅 淺紅 倍數
par = ['g', 'lightgreen', 'r', 'mistyrose', 3]

# 下載當天股票資訊
df = [0,0,0,0,0,0]
t = ['1d', '1m'] # 週期，頻率
df[0] = yf.download(stock[0], period=t[0], interval=t[1]) # 下載一支股票當天的資
訊，頻率1 為分鐘(1 分 K 線)
df[1] = yf.download(stock[1], period=t[0], interval=t[1])
df[2] = yf.download(stock[2], period=t[0], interval=t[1])
df[3] = yf.download(stock[3], period=t[0], interval=t[1])
df[4] = yf.download(stock[4], period=t[0], interval=t[1])
df[5] = yf.download(stock[5], period=t[0], interval=t[1])

# 載到 csv 檔(檢查資料正確用)
df[1].to_csv('df[1].csv')
print('success')
```

```

# 整理股票資訊
# 刪除檔案第一列與最後 2 列 ( 搓合成交量影響比例 )
df[0] = df[0].drop(df[0].index[[0, -1, -2]])
df[1] = df[1].drop(df[1].index[[0, -1, -2]])
df[2] = df[2].drop(df[2].index[[0, -1, -2]])
df[3] = df[3].drop(df[3].index[[0, -1, -2]])
df[4] = df[4].drop(df[4].index[[0, -1, -2]])
df[5] = df[5].drop(df[5].index[[0, -1, -2]])

# ( 檢查資料擷取正確與否用 )
print(df[1])

# 畫圖
# 設定畫布
left = 0.024 # 刻度顯示到百位
fig = plt.figure(figsize=(24, 8))
ax = fig.add_axes([left, 0.88, 0.97, 0.1]) # 第一條界線( 控制畫佈的左, 下, 寬度, 高度 )
ax2 = fig.add_axes([left, 0.82, 0.97, 0.06]) # 第二條界線
ax3 = fig.add_axes([left, 0.72, 0.97, 0.1])
ax4 = fig.add_axes([left, 0.66, 0.97, 0.06])
ax5 = fig.add_axes([left, 0.56, 0.97, 0.1])
ax6 = fig.add_axes([left, 0.5, 0.97, 0.06])
ax7 = fig.add_axes([left, 0.4, 0.97, 0.1])
ax8 = fig.add_axes([left, 0.34, 0.97, 0.06])
ax9 = fig.add_axes([left, 0.24, 0.97, 0.1])
ax10 = fig.add_axes([left, 0.18, 0.97, 0.06])
ax11 = fig.add_axes([left, 0.08, 0.97, 0.1])
ax12 = fig.add_axes([left, 0.02, 0.97, 0.06])

# 消除 xY 軸刻度( 使圖表好看 )
ax2.yaxis.set_visible(False)
ax3.yaxis.set_visible(False)
ax4.yaxis.set_visible(False)
ax5.yaxis.set_visible(False)
ax6.yaxis.set_visible(False)

```

```
ax7.yaxis.set_visible(False)
ax8.yaxis.set_visible(False)
ax9.yaxis.set_visible(False)
ax10.yaxis.set_visible(False)
ax11.yaxis.set_visible(False)
ax12.yaxis.set_visible(False)
ax12.xaxis.set_visible(False)
```

畫趨勢圖

```
ax.plot(df[0]['Adj Close'], label=stock[0])
ax.legend(loc=1) # 圖標標於右上角
ax.grid(linestyle='-', linewidth=1, axis='y')
```

```
ax3.plot(df[1]['Adj Close'], label=stock[1])
ax3.legend(loc=1)
```

```
ax5.plot(df[2]['Adj Close'], label=stock[2])
ax5.legend(loc=1)
```

```
ax7.plot(df[3]['Adj Close'], label=stock[3])
ax7.legend(loc=1)
```

```
ax9.plot(df[4]['Adj Close'], label=stock[4])
ax9.legend(loc=1)
```

```
ax11.plot(df[5]['Adj Close'], label=stock[5])
ax11.legend(loc=1)
```

均線

```
ax.plot(df[0]['Close'].rolling(pa[0]).mean(), linewidth=pa[1], color=pa[2]) # 對每分5
分k 收盤價 做一次平均， 取5ma
ax3.plot(df[1]['Close'].rolling(pa[0]).mean(), linewidth=pa[1], color=pa[2])
ax5.plot(df[2]['Close'].rolling(pa[0]).mean(), linewidth=pa[1], color=pa[2])
ax7.plot(df[3]['Close'].rolling(pa[0]).mean(), linewidth=pa[1], color=pa[2])
ax9.plot(df[4]['Close'].rolling(pa[0]).mean(), linewidth=pa[1], color=pa[2])
ax11.plot(df[5]['Close'].rolling(pa[0]).mean(), linewidth=pa[1], color=pa[2])
```

畫成交量 (成交量> 平均量兩.5 倍 畫成 深綠、深紅)

```
ax2.bar(np.arange(0, len(df[0].index)), df[0].Volume, color=[
    par[0] if df[0].Open[x] > df[0].Close[x] and df[0].Volume[x] >
    par[4]*df[0]['Volume'].mean() else par[1] if df[0].Open[x] > df[0].Close[x]
    else par[2] if df[0].Open[x] < df[0].Close[x] and df[0].Volume[x] >
    par[4]*df[0]['Volume'].mean() else par[3] for x in range(0,len(df[0].index))
])

ax4.bar(np.arange(0, len(df[1].index)), df[1].Volume, color=[
    par[0] if df[1].Open[x] > df[1].Close[x] and df[1].Volume[x] >
    par[4]*df[1]['Volume'].mean() else par[1] if df[1].Open[x] > df[1].Close[x]
    else par[2] if df[1].Open[x] < df[1].Close[x] and df[1].Volume[x] >
    par[4]*df[1]['Volume'].mean() else par[3] for x in range(0,len(df[1].index))
])

ax6.bar(np.arange(0, len(df[2].index)), df[2].Volume, color=[
    par[0] if df[2].Open[x] > df[2].Close[x] and df[2].Volume[x] >
    par[4]*df[2]['Volume'].mean() else par[1] if df[2].Open[x] > df[2].Close[x]
    else par[2] if df[2].Open[x] < df[2].Close[x] and df[2].Volume[x] >
    par[4]*df[2]['Volume'].mean() else par[3] for x in range(0,len(df[2].index))
])

ax8.bar(np.arange(0, len(df[3].index)), df[3].Volume, color=[
    par[0] if df[3].Open[x] > df[3].Close[x] and df[3].Volume[x] >
    par[4]*df[3]['Volume'].mean() else par[1] if df[3].Open[x] > df[3].Close[x]
    else par[2] if df[3].Open[x] < df[3].Close[x] and df[3].Volume[x] >
    par[4]*df[3]['Volume'].mean() else par[3] for x in range(0,len(df[3].index))
])

ax10.bar(np.arange(0, len(df[4].index)), df[4].Volume, color=[
    par[0] if df[4].Open[x] > df[4].Close[x] and df[4].Volume[x] >
    2.5*df[4]['Volume'].mean() else par[1] if df[4].Open[x] > df[4].Close[x]
    else par[2] if df[4].Open[x] < df[4].Close[x] and df[4].Volume[x] >
    2.5*df[4]['Volume'].mean() else par[3] for x in range(0,len(df[4].index))
])

ax12.bar(np.arange(0, len(df[5].index)), df[5].Volume, color=[
    par[0] if df[5].Open[x] > df[5].Close[x] and df[5].Volume[x] >
    2.5*df[5]['Volume'].mean() else par[1] if df[5].Open[x] > df[5].Close[x]
    else par[2] if df[5].Open[x] < df[5].Close[x] and df[5].Volume[x] >
    2.5*df[5]['Volume'].mean() else par[3] for x in range(0,len(df[5].index))
])
```

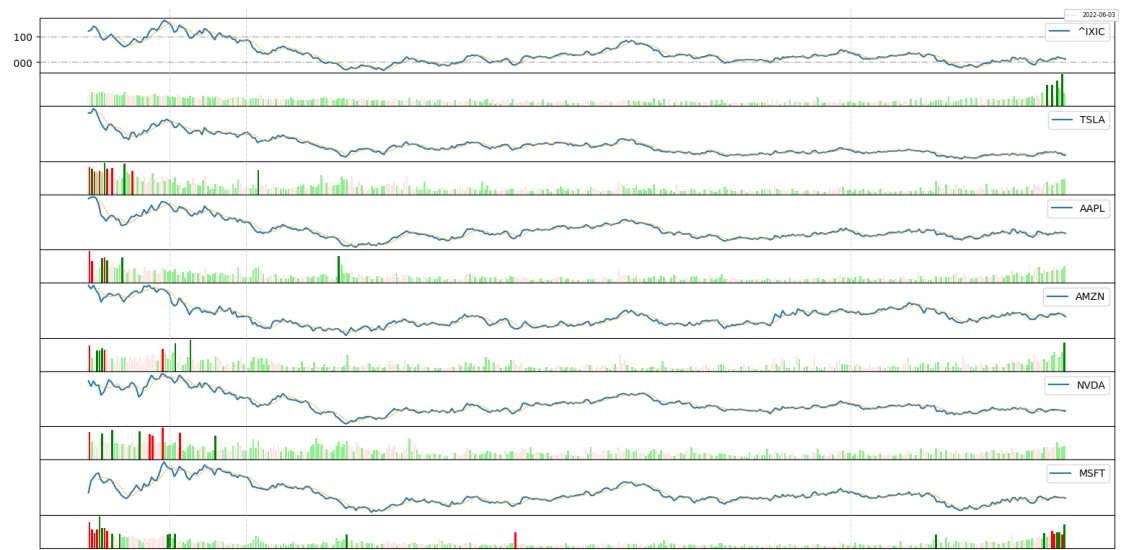
```
# 畫整體時間線切割
    # 設一子畫布占全螢幕
ax_all = fig.add_axes([0, 0, 1, 1])
ax_all.patch.set_alpha(0)    # 背景完全透明
ax_all.axis('off')          # 消除邊框

    # 畫垂直線
# ax4.axvline(x=32, ymin=0, ymax=1)    # 10:00
# ax4.axvline(x=62, ymin=0, ymax=1)    # 10:30 (定基準用)
# ax4.axvline(x=302, ymin=0, ymax=1)    # 14:30

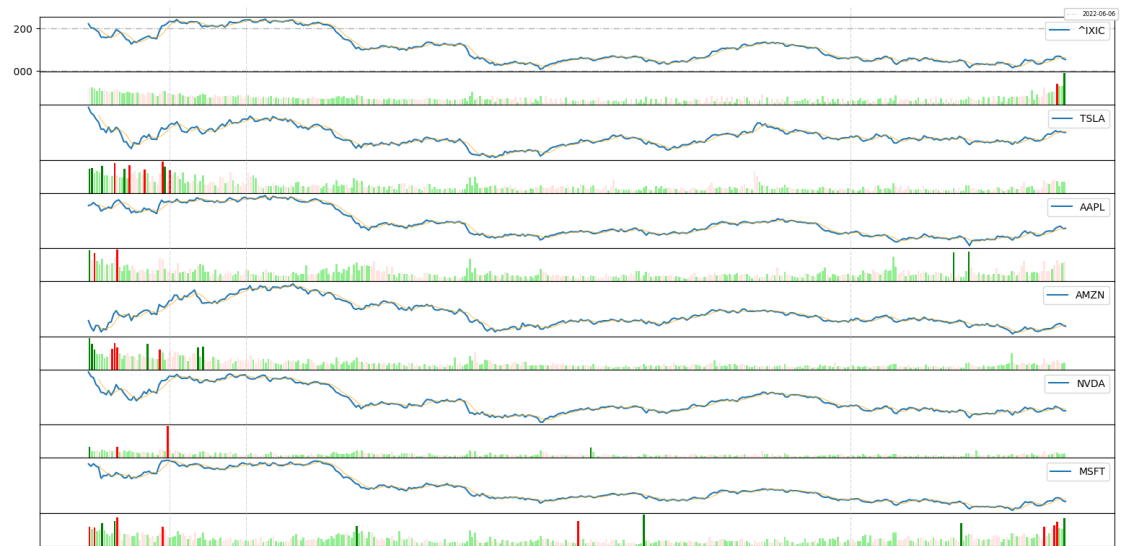
ax_all.axvline(x=0.1411, ymin=0, ymax=1, color='silver', linestyle='-.', linewidth=0.5,
label=date.today() - timedelta(1))    # 10:00
ax_all.axvline(x=0.21, ymin=0, ymax=1, color='silver', linestyle='-.', linewidth=0.5)
# 10:30
ax_all.axvline(x=0.7557, ymin=0, ymax=1, color='silver', linestyle='-.', linewidth=0.5)
# 14:30
ax_all.legend(loc=1, fontsize='xx-small', facecolor='white', framealpha=1)    # 標記時間戳

plt.show()
```

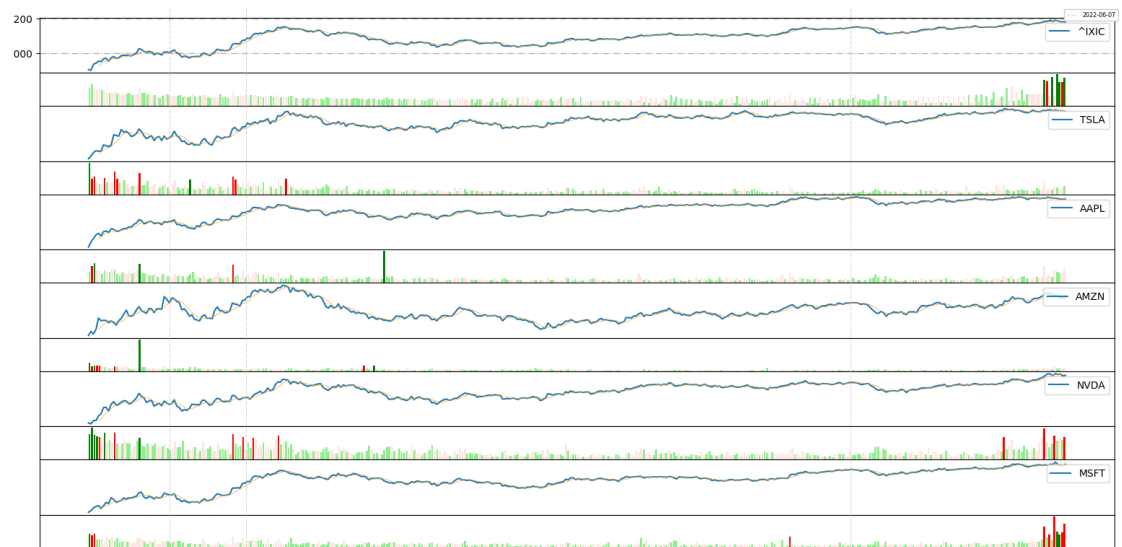
● 執行成果



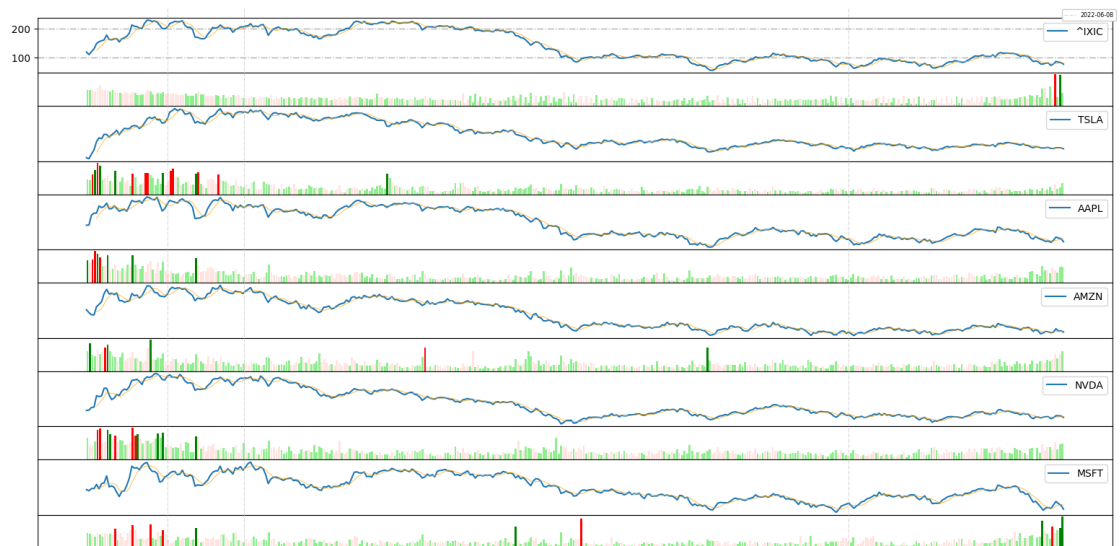
06/03



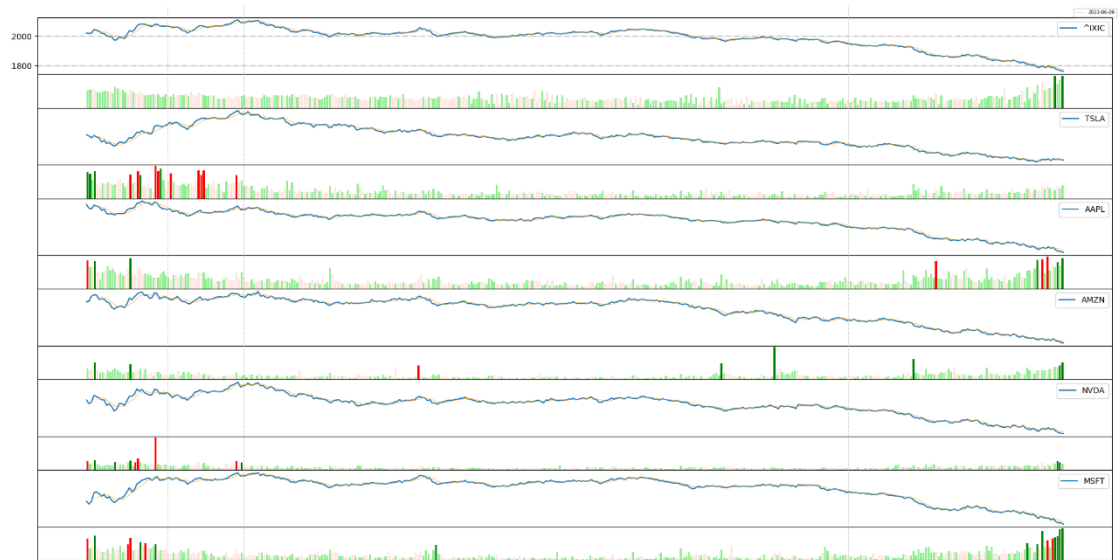
06/06



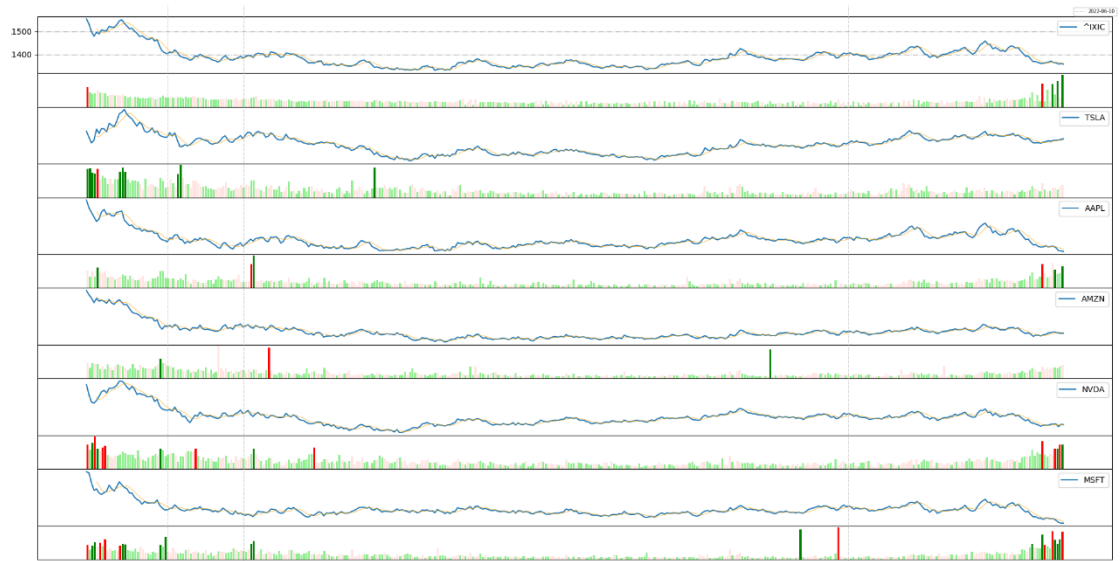
06/07



06/08



06/09



06/10

● 參考資料

<https://ithelp.ithome.com.tw/articles/10206894>

<https://pythonviz.com/finance/yfinance-download-stock-data/>

<https://walkonnet.com/archives/517944>

<https://min.news/zh-tw/technique/e79f74de529f9735a4ad100013f7f448.html>

<https://www.delftstack.com/zh-tw/howto/python-pandas/drop-row-pandas/>

<https://www.delftstack.com/zh-tw/howto/matplotlib/how-to-show-pyplot-grid-in-matplotlib/>

<https://www.delftstack.com/zh-tw/howto/matplotlib/how-to-hide-axis-text-ticks-and-or-tick-labels-in-matplotlib/>

<https://www.delftstack.com/zh-tw/howto/matplotlib/how-to-plot-horizontal-and-vertical-line-in-matplotlib/>

[https://blog.csdn.net/helunqu2017/article/details/78736721?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-](https://blog.csdn.net/helunqu2017/article/details/78736721?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&utm_relevant_index=2)

[2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&depth_1-](https://blog.csdn.net/helunqu2017/article/details/78736721?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&utm_relevant_index=2)

[utm_source=distribute.pc_relevant.none-task-blog-](https://blog.csdn.net/helunqu2017/article/details/78736721?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&utm_relevant_index=2)

[2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-](https://blog.csdn.net/helunqu2017/article/details/78736721?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&utm_relevant_index=2)

[78650339.nonecase&utm_relevant_index=2](https://blog.csdn.net/helunqu2017/article/details/78736721?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7Edefault-1-78736721-blog-78650339.nonecase&utm_relevant_index=2)