



Zhong Ting Xu

- Education

- ✓ NCCU
- ✓ NCCU

Candidate of Master of Business Administration
Bachelor of Science in Statistics

- Work experience

- ✓ DBS Taiwan
- ✓ Asian Pacific Research

Audit IT team intern
Analyst

- Skillset

- ✓ Statistic software : SPSS 、 SAS EG
- ✓ Language : Python 、 R
- ✓ Database : SQL 、 ACL 、 Bloomberg
- ✓ Visualization : PowerBI 、 Tableau

- License

- ✓ TOEIC : 930
- ✓ Senior Securities Specialist (TSA)

- Project Experiences

- ✓ TMBA x JihSun Securities :
Develop stock trading strategy using MultiChart & Python
- ✓ NCCU x President Securities :
Develop customer clustering model using Python
- ✓ Machine learning marathon challenge :
100 days ML challenge with Python

✉ hndsmhsu@gmail.com
in <https://www.linkedin.com/in/benny0624/>
github <https://github.com/Benny0624>

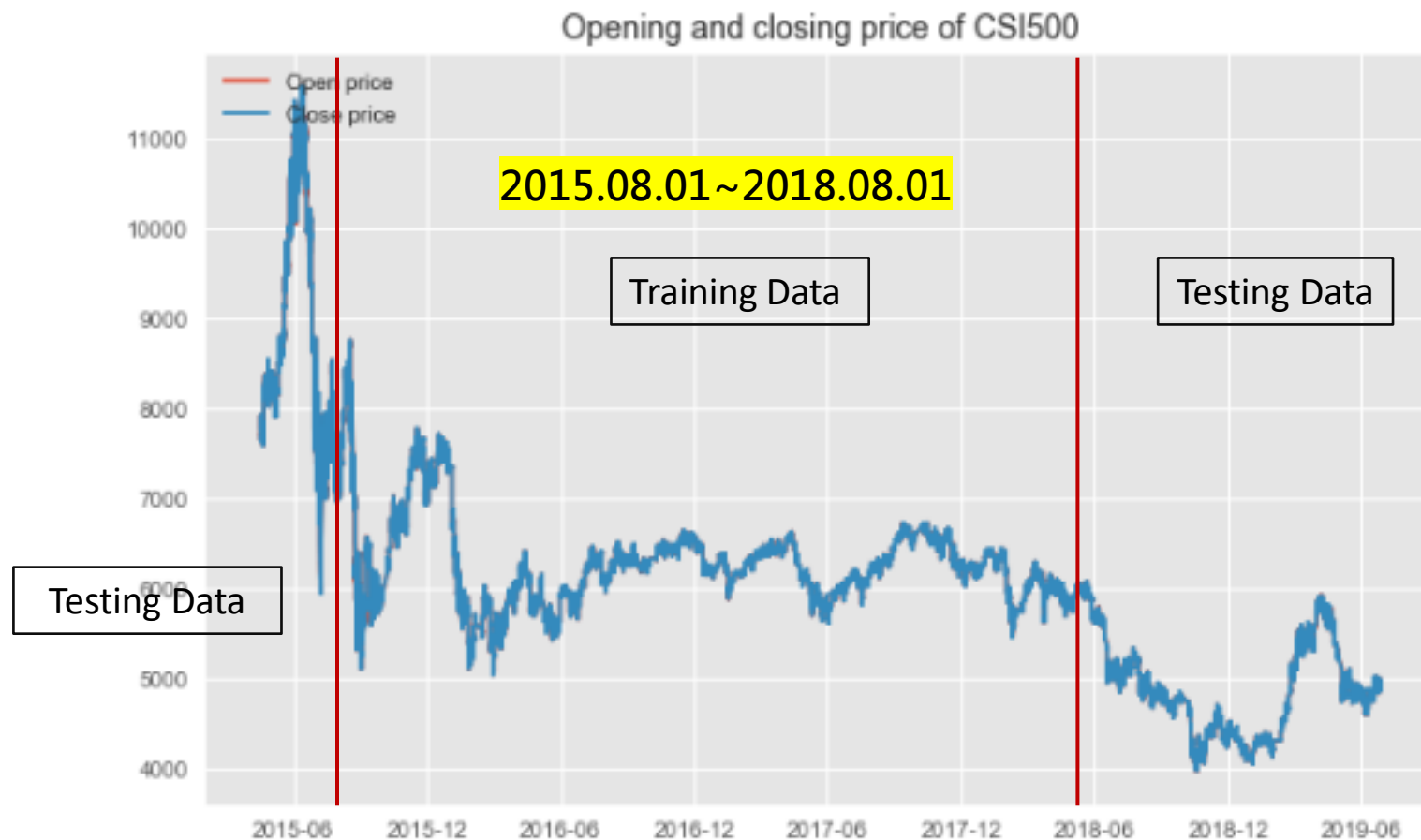
策略分享

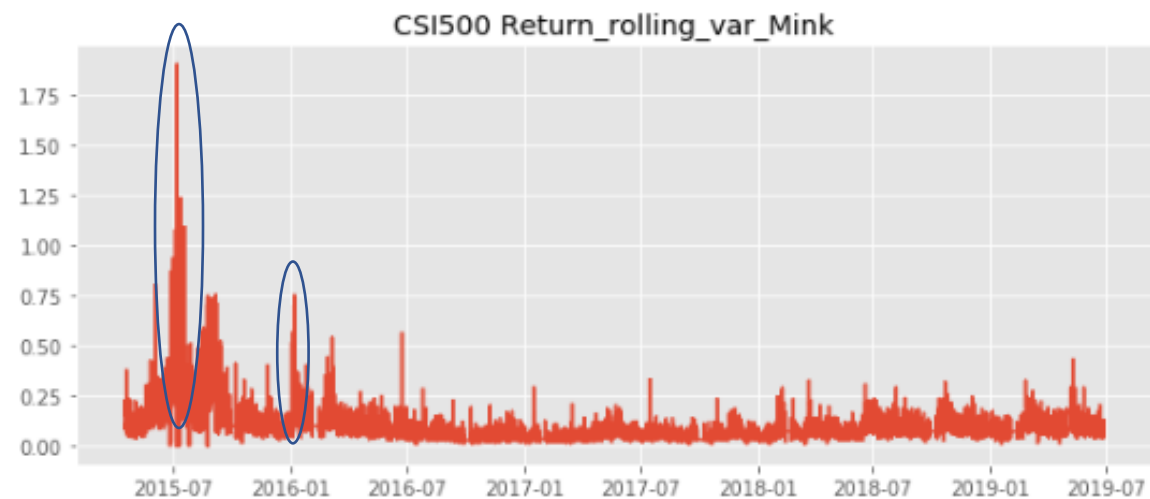
Day Trading – Target : CSI 500

Data period : 2015.01 ~ 2019.06

Underlying Bond	CSI 500 Index
Contract Multiplier	CNY 200
Unit	Index point
Tick Size	0.2 point
Contract Months	Monthly: current month, next month, next two calendar quarters (four total)
Trading Hours	09:30 am - 11:30 am, 01:00 pm - 03:00 pm
Limit Up/Down	±10% of the settlement price on the previous trading day
Minimum Margin Requirement	8% of the contract value
Last Trading Day	Third Friday of the contract month, postponed to the next business day if it falls on a public holiday
Delivery Day	Third Friday, same as "Last Trading Day"
Settlement Method	cash settlement
Transaction Code	IC
Exchange	China Financial Futures Exchange

EDA - Sampling

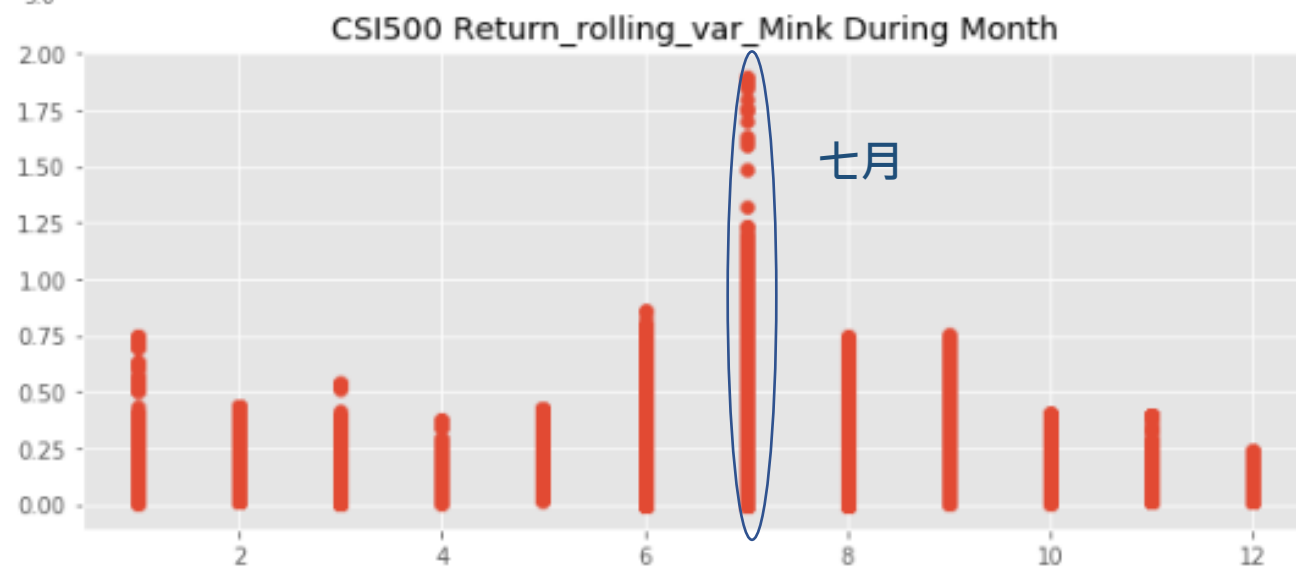
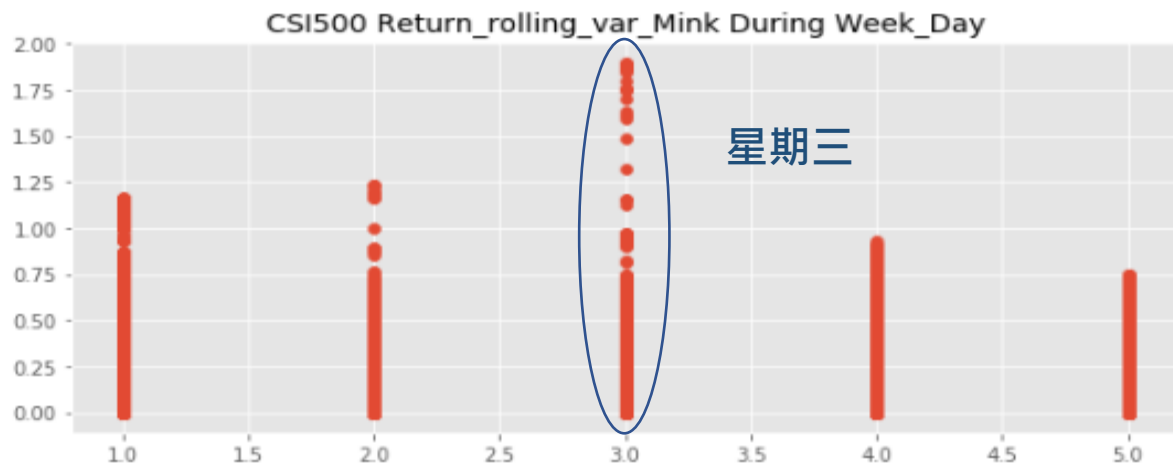




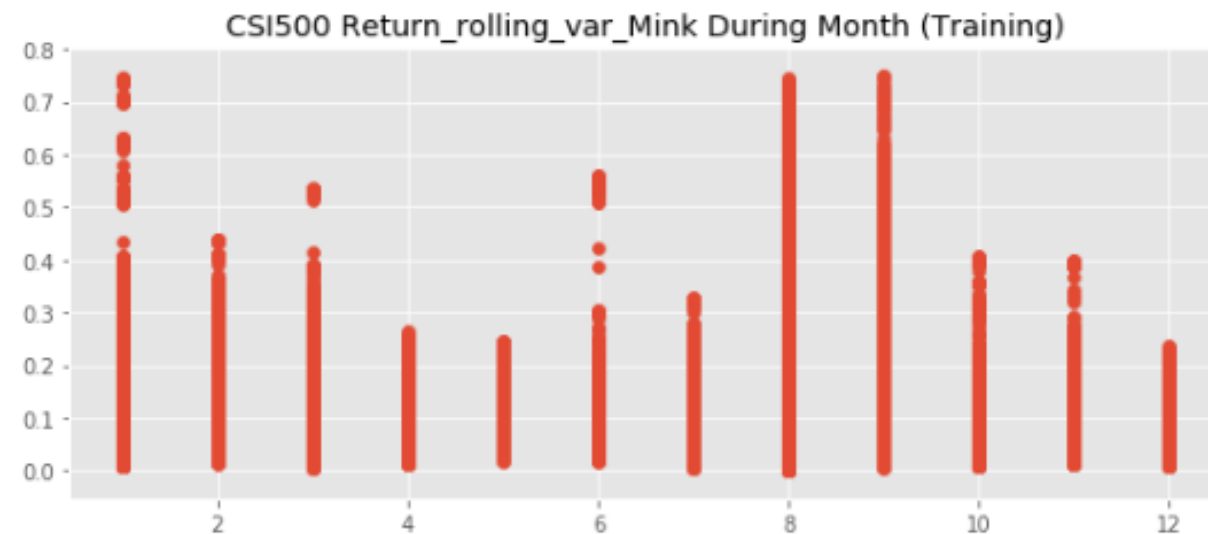
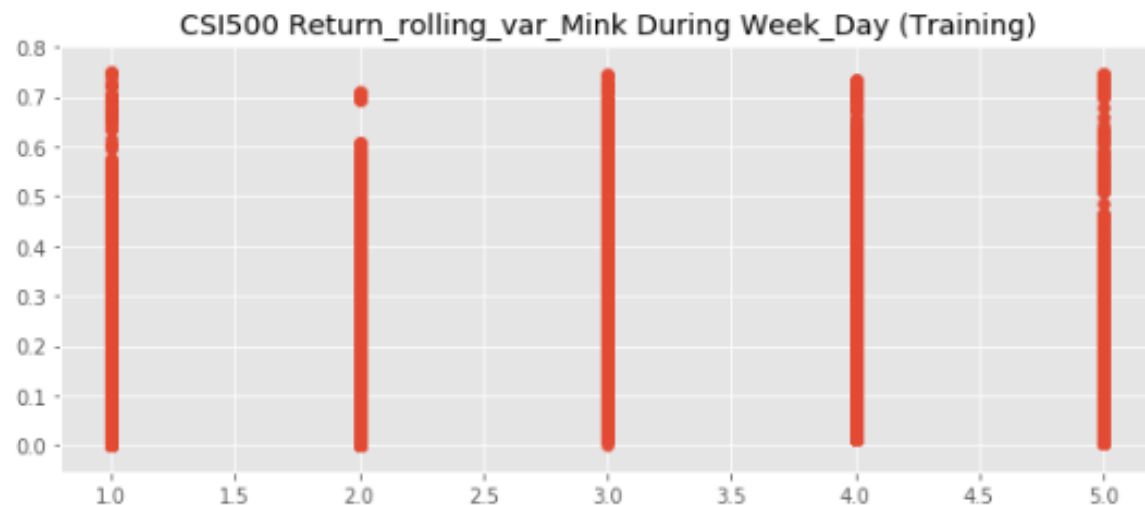
實體k棒報酬率: $(\text{Close} - \text{Open}) / \text{Open}$



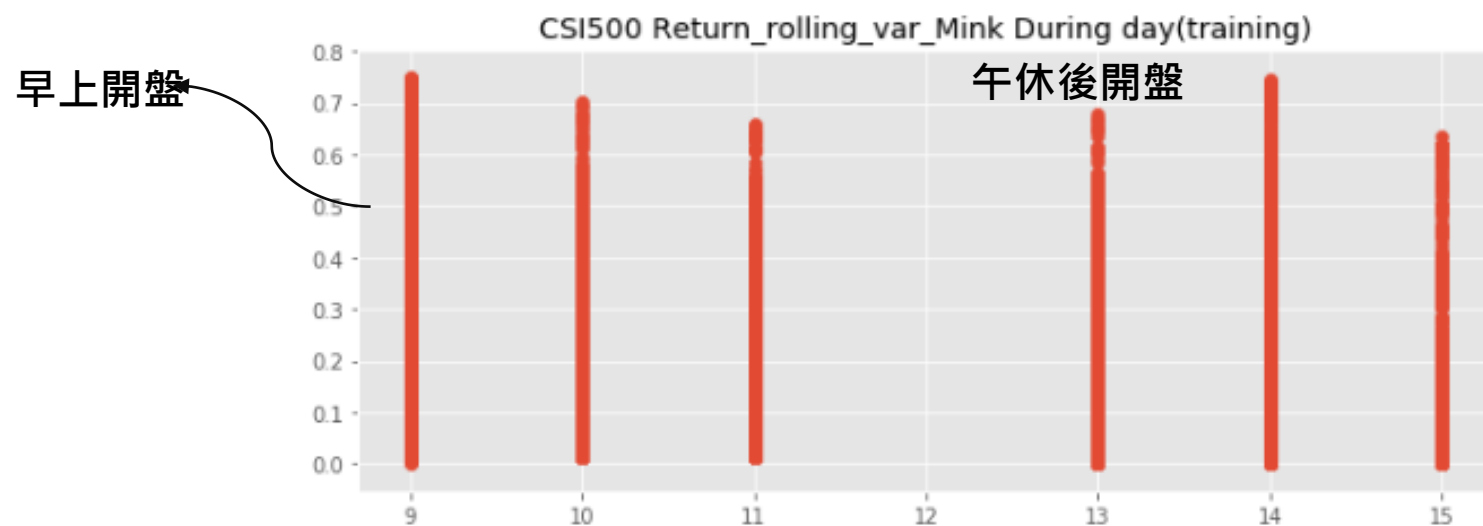
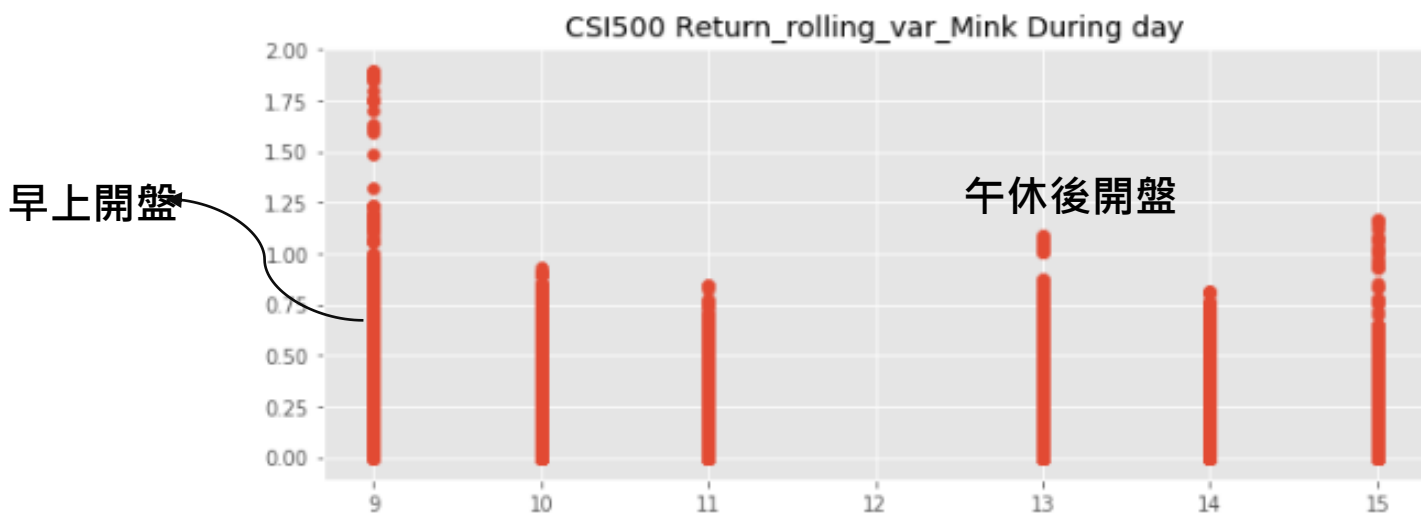
EDA - Rolling variance of month & weekday daily return



EDA - Rolling variance of month & weekday daily return (Training data)



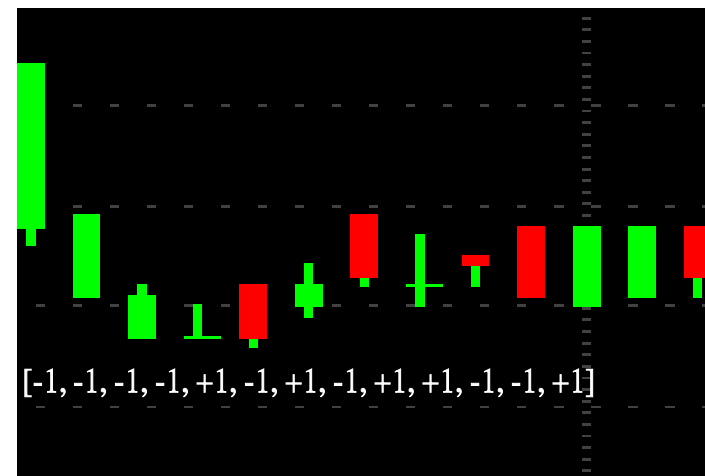
EDA - Rolling variance of hourly return



EDA – Trend distribution

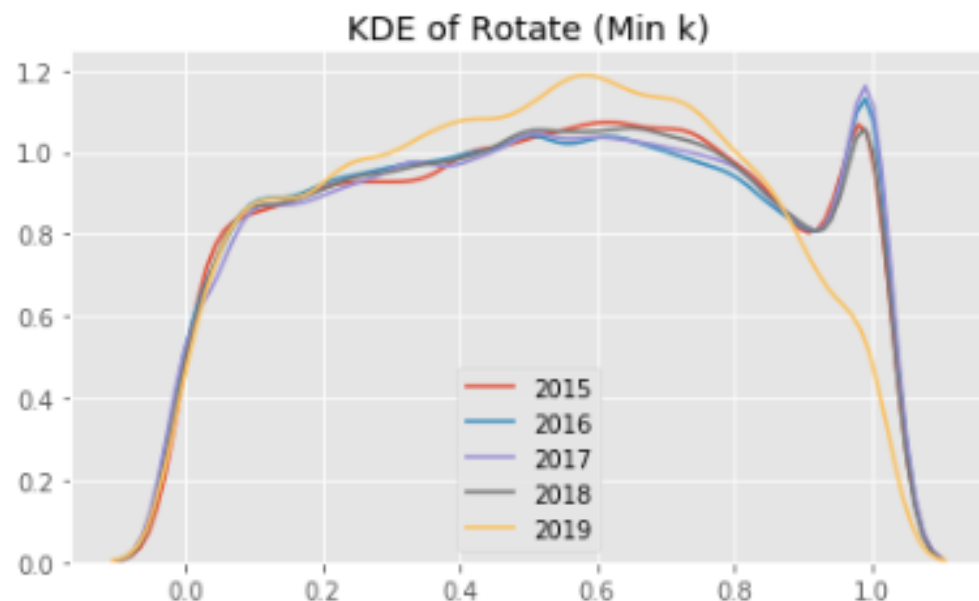
```
def Go_up_num(Series):
    positive_list = []
    longest = 0
    current = 0
    for i in Series:
        if i == 1:
            current += 1
        else:
            longest = max(longest, current)
            positive_list.append(longest)
            current = 0
    return positive_list

def Go_down_num(Series):
    negative_list = []
    longest = 0
    current = 0
    for i in Series:
        if i == -1:
            current += 1
        else:
            longest = max(longest, current)
            negative_list.append(longest)
            current = 0
    return negative_list
```



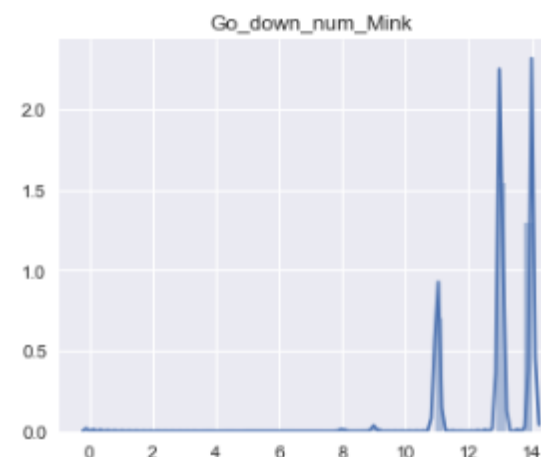
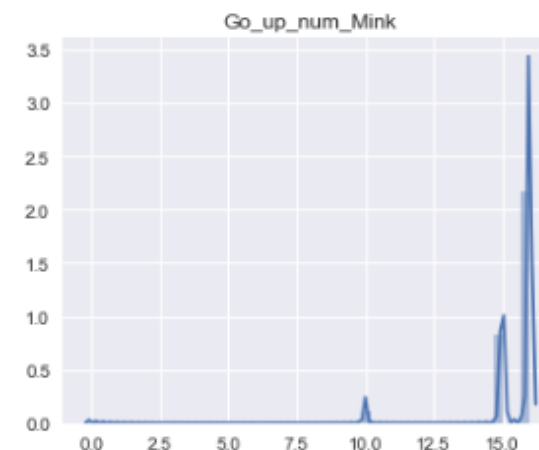
- 計算連續紅棒、綠棒次數，以上圖為例
- -1: [4, 1, 1, 2]
- +1: [1, 1, 2, 1]
- 0: 開收盤價相同則填入0

EDA – Trend distribution



```
CSI500['Range'] = CSI500['High'] - CSI500['Low']  
CSI500['Diff'] = CSI500['Close'] - CSI500['Open']  
CSI500['Rotate'] = abs(CSI500['Diff']) / CSI500['Range']
```

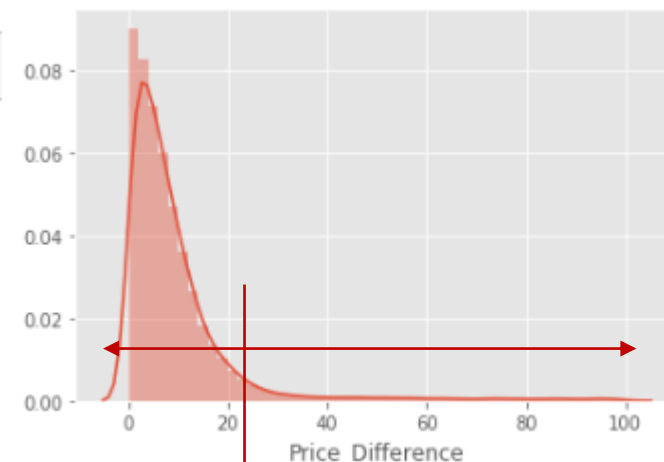
- 計算連續向上、向下k棒分布，發現都呈現右偏**順勢**分布
- 最高最低距離和以實體k棒的比率，去繪製KDE圖
- 從圖中可以看出，它們是靠向右端，表示向上甩高、向下甩低的比例很多，是**順勢**的好兆頭



EDA – Contango & Backwardation

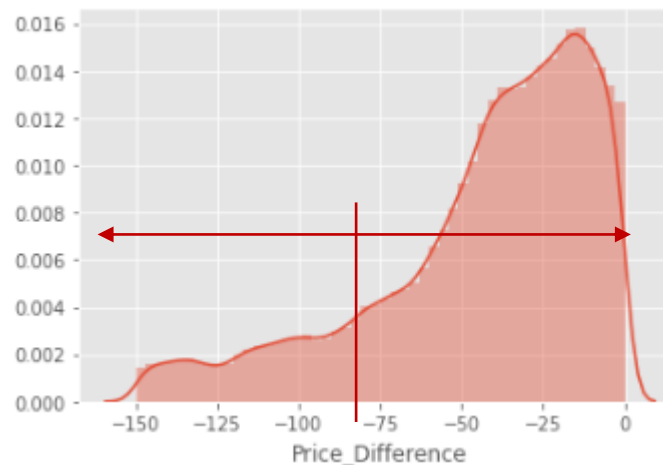
```
In [12]: CSI500_SSE.loc[CSI500_SSE['Price_Difference'] > 0, 'Price_Difference'].describe()
```

```
Out[12]: count    25471.000000  
mean      23.058546  
std       60.564933  
min        0.010000  
25%        3.010000  
50%        6.680000  
75%       12.740000  
max       564.620000  
Name: Price_Difference, dtype: float64
```



```
In [13]: CSI500_SSE.loc[CSI500_SSE['Price_Difference'] < 0, 'Price_Difference'].describe()
```

```
Out[13]: count    217854.000000  
mean     -79.912086  
std      100.807606  
min     -889.950000  
25%     -95.470000  
50%     -42.280000  
75%     -20.410000  
max       -0.010000  
Name: Price_Difference, dtype: float64
```



Strategy logic

期貨

現貨

```
INPUT: DataSeries1(C OF DATA1), DataSeries2(C OF DATA2), UP(100), DN(100),  
       PercentB(1.0), PercentS(1.0), PercentB2(1.0), PercentS2(1.0), BEGINTIME(0930), ENDTIME(1200);  
VARS: RANGE(0);
```

```
// Value Setting
```

```
RANGE = HIGHD(1) - LOWD(1);
```

```
VALUE1 = OPEND(0) + (RANGE * PercentB);
```

```
VALUE2 = OPEND(0) - (RANGE * PercentS);
```

```
VALUE3 = DataSeries1 - DataSeries2;
```

```
VALUE4 = OPEND(0) + (RANGE * PercentB2);
```

```
VALUE5 = OPEND(0) - (RANGE * PercentS2);
```

名稱	數值
DataSeries1	c of data1
DataSeries2	c of data2
UP	105
DN	94
PercentB	0.5
PercentS	0.2
PercentB2	1
PercentS2	0.9
BEGINTIME	930
ENDTIME	1200

Strategy logic

```
// Time Condition
CONDITION1 = TIME > BEGINTIME;
CONDITION2 = TIME < ENDTIME;

// Price Diff Condition
CONDITION3 = VALUE3 CROSS OVER DN AND VALUE3 < UP;
CONDITION4 = VALUE3 CROSS UNDER -DN AND VALUE3 > -UP;

// Price Diff Enter
IF CONDITION1 AND CONDITION3 AND ENTRIES TODAY (DATE) = 0 THEN BUY ("BPD") NEXT BAR MARKET;
IF CONDITION1 AND CONDITION4 AND ENTRIES TODAY (DATE) = 0 THEN SELLSHORT ("SPD") NEXT BAR MARKET;

// ORB Enter
IF CONDITION1 AND CONDITION2 AND ENTRIES TODAY (DATE) = 0 THEN BEGIN
    BUY ("BORB") NEXT BAR AT VALUE1 STOP;
    SELLSHORT ("SORB") NEXT BAR AT VALUE2 STOP;
END;
```

Strategy logic

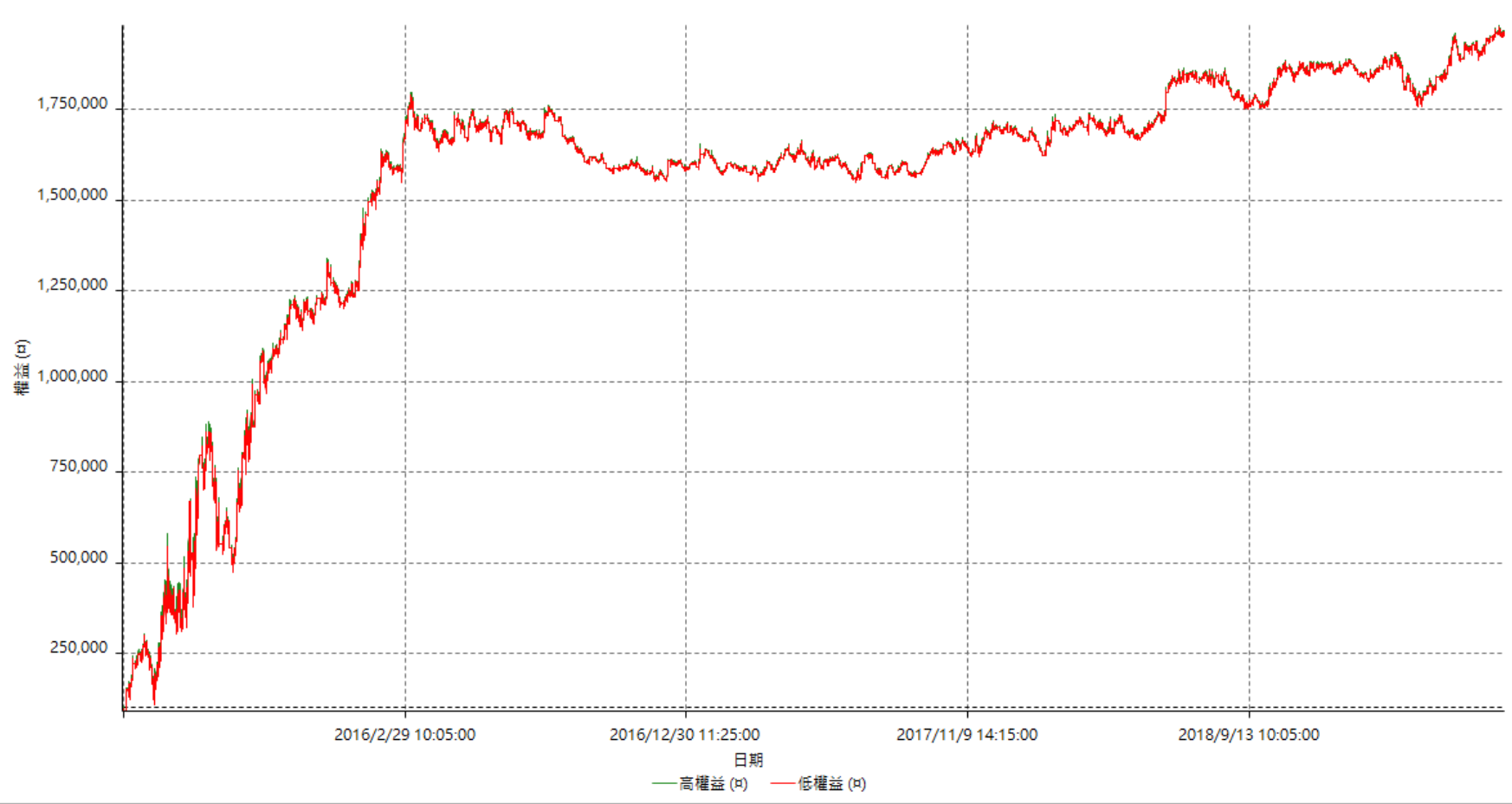
```
// Exit
IF MARKETPOSITION <> 0 THEN BEGIN
    SELL NEXT BAR AT VALUE5 STOP;
    BUYTOCOVER NEXT BAR AT VALUE4 STOP;
END;
```

```
// Day Close Exit
IF TIME >= 1455 THEN BEGIN
    SELL NEXT BAR MARKET;
    BUYTOCOVER NEXT BAR MARKET;
END;
```

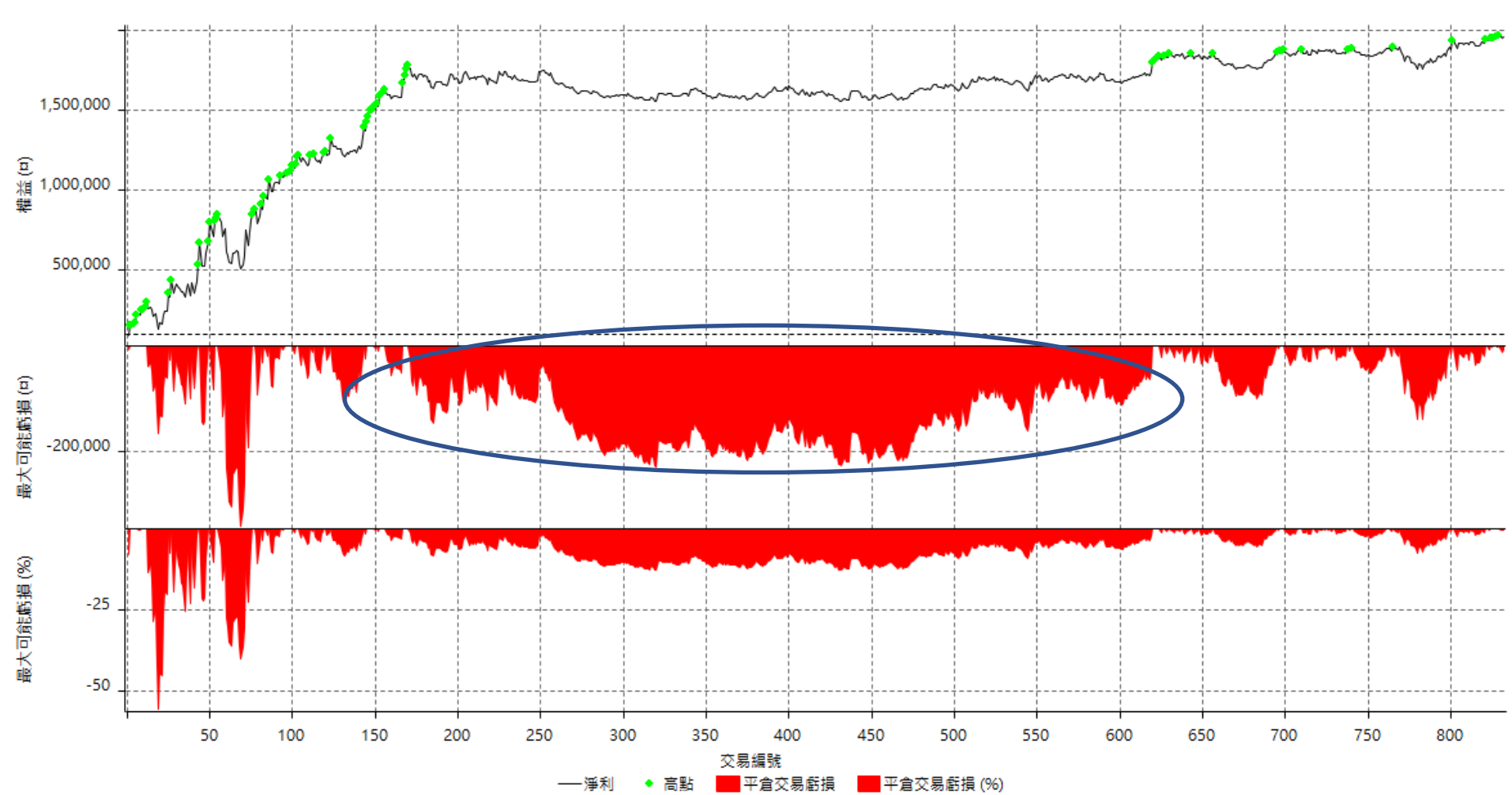
Performance

	所有交易	多單	空單
淨利	¥1856640	¥528200	¥1328440
毛利	¥7036640	¥1798440	¥5238200
毛損	(¥5180000)	(¥1270240)	(¥3909760)
調整後淨利	¥1256367	¥242077.4	¥798948.81
調整後毛利	¥6689125.12	¥1636280.13	¥4928998.83
調整後毛損	(¥5432758.12)	(¥1394202.73)	(¥4130050.02)
特定淨利	¥463200	¥388240	¥74960
特定毛利	¥3870600	¥1230920	¥2639680
特定毛損	(¥3407400)	(¥842680)	(¥2564720)
帳戶所需金額	¥340680	¥215760	¥206760
帳戶報酬	544.98%	244.81%	642.5%
初始資本報酬	1856.64%	528.2%	1328.44%
最大策略虧損	(¥413080)	(¥294280)	(¥298360)
最大策略虧損 (%)	(64.38%)	(61.08%)	(101.62%)
最大平倉交易虧損	(¥340680)	(¥215760)	(¥206760)
最大平倉交易虧損 (%)	(55.4%)	(52.53%)	(89.13%)
最大的策略虧損報酬	4.49	1.79	4.45
獲利因子	1.36	1.42	1.34
調整獲利因子	1.23	1.17	1.19
特定獲利因子	1.14	1.46	1.03
最大持有契約數量	1	1	1
滑價支付	¥832000	¥229000	¥603000
佣金支付	¥0	¥0	¥0
未平倉部位損益	n/a	n/a	n/a
年報酬率	443.16%	126.08%	317.09%
月報酬率	36.93%	10.51%	26.42%
買進持有績效	(¥36606.5)	(¥36606.5)	(¥41572.39)

Performance



Performance

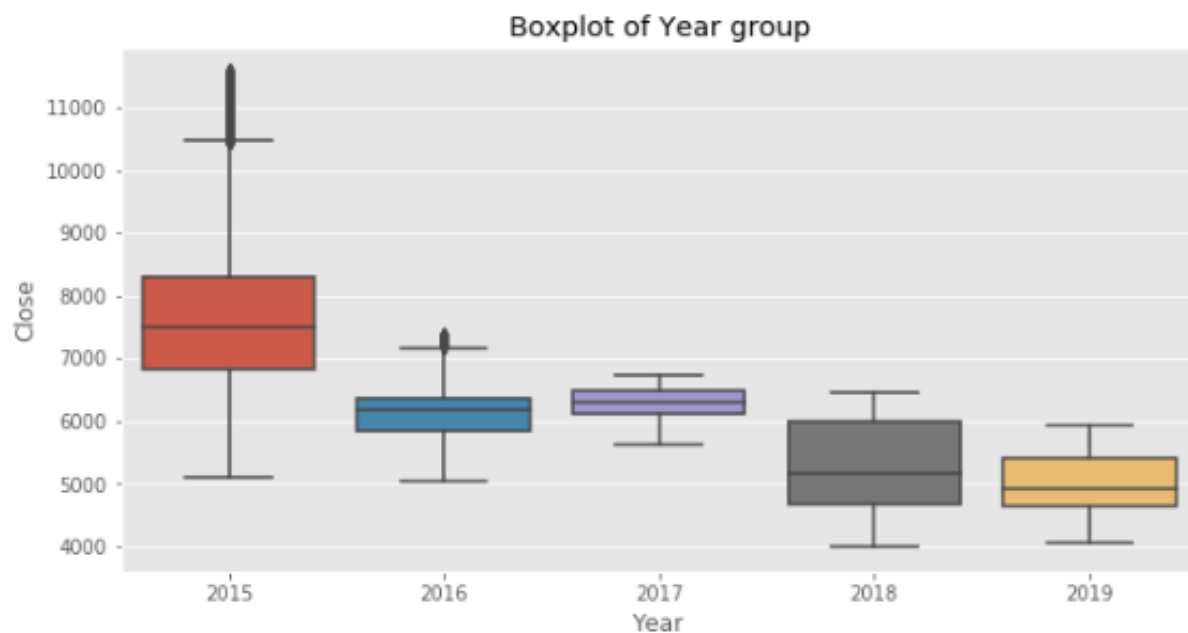


Performance

期間	獲利		毛利	毛損	交易次數	勝率
	¥	%				
2019	¥80400	4.29%	¥597440	(¥517040)	95	44.21%
2018	¥179120	10.55%	¥1112280	(¥933160)	208	52.88%
2017	¥109880	6.92%	¥877640	(¥767760)	196	51.02%
2016	¥310480	24.32%	¥1359360	(¥1048880)	191	42.93%
2015	¥1176760	1176.76%	¥3089920	(¥1913160)	142	53.52%

	所有交易	多單	空單
交易總次數	832	229	603
未平倉交易總數量	0	0	0
獲利交易次數	410	123	287
虧損交易次數	420	105	315
勝率	49.28%	53.71%	47.6%
平均交易(獲利 虧損)	¥2231.54	¥2306.55	¥2203.05
平均獲利交易	¥17162.54	¥14621.46	¥18251.57
平均虧損交易	(¥12333.33)	(¥12097.52)	(¥12411.94)
平均獲利/平均虧損 比率	1.39	1.21	1.47
最大的交易獲利	¥168000	¥109720	¥168000
最大的交易虧損	(¥146040)	(¥146040)	(¥143520)
平倉交易的平均K棒數	40.3	38.8	40.9
獲利平倉交易的平均K棒數	42.5	39.8	43.6
虧損平倉交易的平均K棒數	38.2	37.6	38.4
平倉交易間的平均K棒數	42	43	41
獲利平倉交易間的平均K棒數	77.3	359.5	127.5
虧損平倉交易間的平均K棒數	78.7	430.2	117.5

Strategy review



問題

1. 2016,2017 盤很黏，策略吃不到肉
2. 跳空、價差幅度過度依賴最佳化，可能過度配適
3. 最大回檔(MDD)有點大

解決

1. 多策略portfolio配置其他市場(非中國)，盤黏就不要做，倚靠其他市場賺錢
2. 跳空(價差)幅度以降低最大回檔(MDD)為首要原則，少吃一點也不要跌出場
3. 最大回檔(MDD)出現在2015(股災)，以後再出現機會不大(中國法規限制)

附錄 - 機器學習(Decision tree)

Daily Return

紅黑 k 報酬 (日內)

```
In [9]: # Red k return
return_mask = period_stock_data['diff_1D'] > 0
period_stock_data.loc[return_mask, 'return_1D'].describe()
```

```
Out[9]: count    525.000000
        mean      0.012921
        std       0.014898
        min       0.000030
        25%       0.003879
        50%       0.008030
        75%       0.016751
        max       0.099990
        Name: return_1D, dtype: float64
```

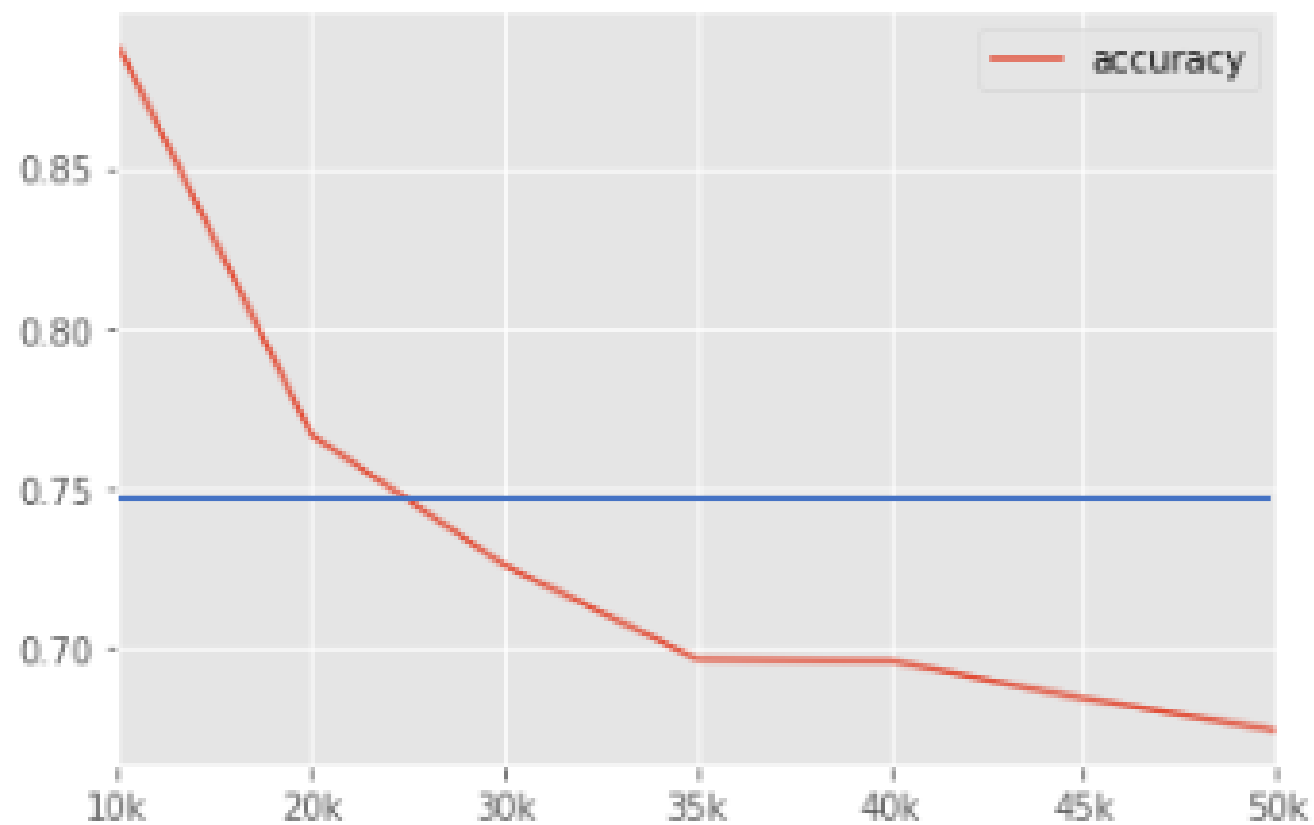
```
In [10]: # Black k return
return_mask = period_stock_data['diff_1D'] < 0
period_stock_data.loc[return_mask, 'return_1D'].describe()
```

```
Out[10]: count    499.000000
         mean     -0.014013
         std      0.018349
         min     -0.116022
         25%     -0.016552
         50%     -0.007695
         75%     -0.003001
         max     -0.000035
         Name: return_1D, dtype: float64
```

Beat the mean return

```
# Future Return cover handling fee (14%)
period_stock_data['future_Return_10_cov'] = period_stock_data['future_Return_10'].map(lambda x: 1 if abs(x) > 0.014 else 0)
period_stock_data['future_Return_20_cov'] = period_stock_data['future_Return_20'].map(lambda x: 1 if abs(x) > 0.014 else 0)
period_stock_data['future_Return_30_cov'] = period_stock_data['future_Return_30'].map(lambda x: 1 if abs(x) > 0.014 else 0)
period_stock_data['future_Return_35_cov'] = period_stock_data['future_Return_35'].map(lambda x: 1 if abs(x) > 0.014 else 0)
period_stock_data['future_Return_40_cov'] = period_stock_data['future_Return_40'].map(lambda x: 1 if abs(x) > 0.014 else 0)
period_stock_data['future_Return_45_cov'] = period_stock_data['future_Return_45'].map(lambda x: 1 if abs(x) > 0.014 else 0)
period_stock_data['future_Return_50_cov'] = period_stock_data['future_Return_50'].map(lambda x: 1 if abs(x) > 0.014 else 0)
```

Model accuracy



Strategy Logic

```
INPUT: DataSeries1(C OF DATA1), DataSeries2(C OF DATA2),  
      PercentB(1.0), PercentS(1.0), PercentB2(1.0), PercentS2(1.0), BEGINTIME(0930), ENDTIME(1200);  
VARS: RANGE(0);  
  
// Value Setting  
RANGE = HIGHD(1) - LOWD(1);  
VALUE1 = OPEND(0) + (RANGE * PercentB);  
VALUE2 = OPEND(0) - (RANGE * PercentS);  
VALUE4 = OPEND(0) + (RANGE * PercentB2);  
VALUE5 = OPEND(0) - (RANGE * PercentS2);  
  
// Time Condition  
CONDITION1 = TIME > BEGINTIME;  
CONDITION2 = TIME < ENDTIME;  
  
// Random Forest Condition  
CONDITION5 = (C OF DATA3) = 1;  
CONDITION6 = (C OF DATA4) = 1;  
CONDITION7 = (C OF DATA5) = 1;
```


Strategy Logic

```
// Random Forest Enter
IF CONDITION1 AND CONDITION5 AND ENTRIES TODAY (DATE) = 0 THEN BEGIN
    BUY("B11") NEXT BAR AT VALUE1 STOP;
    SELLSHORT("S11") NEXT BAR AT VALUE2 STOP;
END;
IF CONDITION1 AND CONDITION6 AND ENTRIES TODAY (DATE) = 0 THEN BEGIN
    BUY("B12") NEXT BAR AT VALUE1 STOP;
    SELLSHORT("S12") NEXT BAR AT VALUE2 STOP;
END;

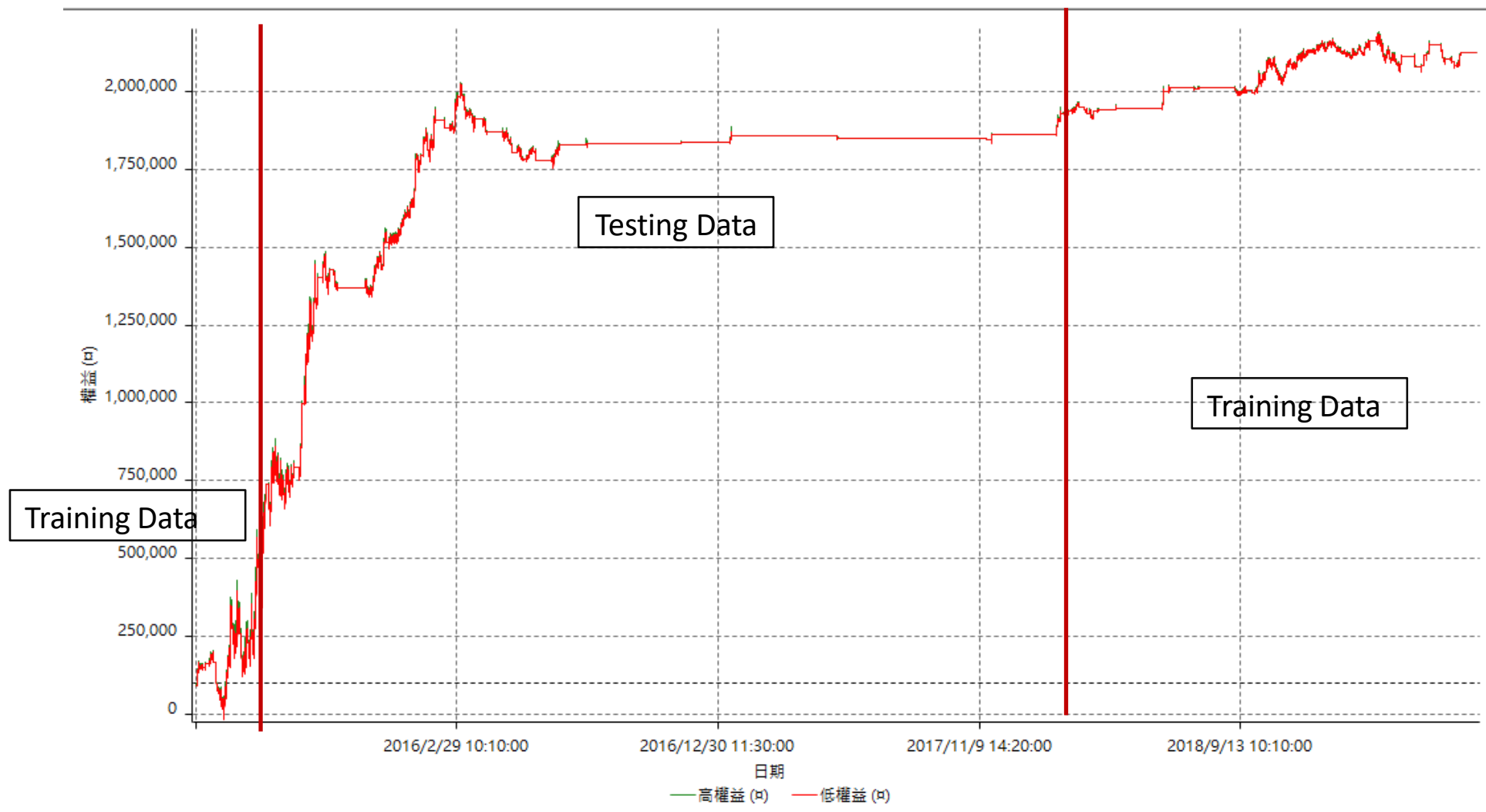
// Exit
IF MARKETPOSITION <> 0 THEN BEGIN
    SELL NEXT BAR AT VALUE5 STOP;
    BUYTOCOVER NEXT BAR AT VALUE4 STOP;
END;

// Day Close Exit
IF TIME >= 1455 THEN BEGIN
    SELL NEXT BAR MARKET;
    BUYTOCOVER NEXT BAR MARKET;
END;
```

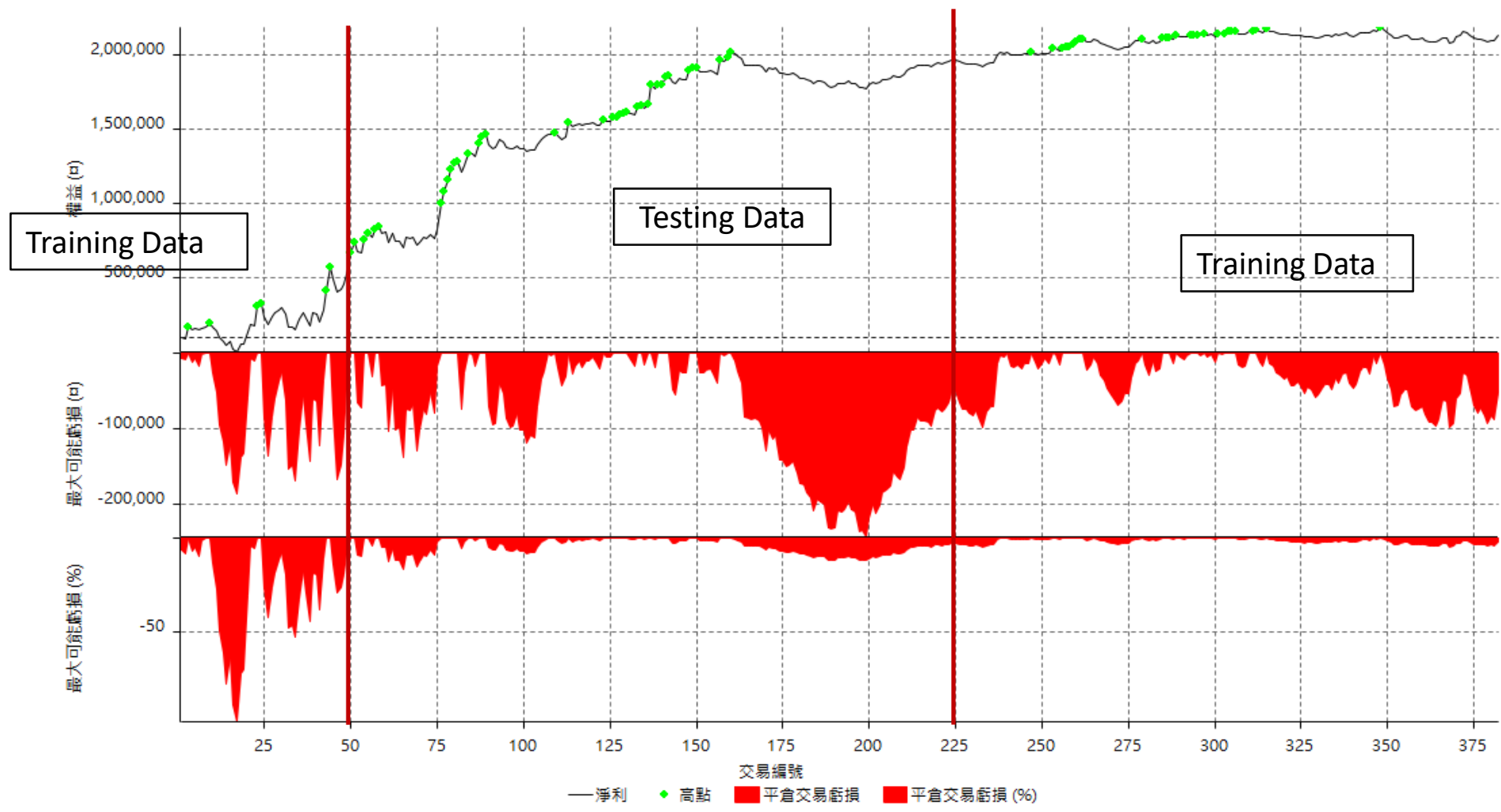
Performance

	所有交易	多單	空單
淨利	¥2023960	¥574840	¥1449120
毛利	¥4907440	¥1402800	¥3504640
毛損	(¥2883480)	(¥827960)	(¥2055520)
調整後淨利	¥1464244.51	¥274670.33	¥975353.98
調整後毛利	¥4563850.48	¥1227450	¥3208443.86
調整後毛損	(¥3099605.96)	(¥952779.67)	(¥2233089.88)
特定淨利	¥902440	¥668360	¥234080
特定毛利	¥2509720	¥1046800	¥1462920
特定毛損	(¥1607280)	(¥378440)	(¥1228840)
帳戶所需金額	¥243840	¥349200	¥225480
帳戶報酬	830.04%	164.62%	642.68%
初始資本報酬	2023.96%	574.84%	1449.12%
最大策略虧損	(¥316400)	(¥376560)	(¥268040)
最大策略虧損 (%)	(108.49%)	(103.33%)	(184.5%)
最大平倉交易虧損	(¥243840)	(¥349200)	(¥225480)
最大平倉交易虧損 (%)	(97.81%)	(99.87%)	(165.52%)
最大的策略虧損報酬	6.4	1.53	5.41
獲利因子	1.7	1.69	1.7
調整獲利因子	1.47	1.29	1.44
特定獲利因子	1.56	2.77	1.19
最大持有契約數量	1	1	1
滑價支付	¥382000	¥108000	¥274000
佣金支付	¥0	¥0	¥0
未平倉部位損益	n/a	n/a	n/a
年報酬率	483.1%	137.21%	345.89%
月報酬率	40.26%	11.43%	28.82%

Performance



Performance



Performance

期間	獲利		毛利	毛損	交易次數	勝率
	¥	%				
2019	(¥32400)	(1.5%)	¥346960	(¥379360)	71	43.66%
2018	¥294400	15.81%	¥629880	(¥335480)	101	58.42%
2017	¥23960	1.3%	¥34320	(¥10360)	4	50%
2016	¥176320	10.61%	¥650400	(¥474080)	70	47.14%
2015	¥1561680	1561.68%	¥3245880	(¥1684200)	136	58.09%

	所有交易	多單	空單
交易總次數	382	108	274
未平倉交易總數量	0	0	0
獲利交易次數	204	64	140
虧損交易次數	178	44	134
勝率	53.4%	59.26%	51.09%
平均交易(獲利 虧損)	¥5298.32	¥5322.59	¥5288.76
平均獲利交易	¥24056.08	¥21918.75	¥25033.14
平均虧損交易	(¥16199.33)	(¥18817.27)	(¥15339.7)
平均獲利/平均虧損 比率	1.49	1.16	1.63
最大的交易獲利	¥170560	¥90400	¥170560
最大的交易虧損	(¥90240)	(¥87720)	(¥90240)
平倉交易的平均K棒數	36.8	35	37.6
獲利平倉交易的平均K棒數	42.3	42.6	42.2
虧損平倉交易的平均K棒數	30.5	23.8	32.7
平倉交易間的平均K棒數	n/a	n/a	n/a
獲利平倉交易間的平均K棒數	198.4	724.8	308.6
虧損平倉交易間的平均K棒數	245.4	1092.5	333.9

*Thank
you!*