Group 21

ROBO-ADVISOR

指導教授: 石百達、蔡芸琤
組員: 林依蓁、賴奕辰、鄒峻安

ETF

最佳投資組合 理財機器人

市面上ETF百百種，該如何挑選?

利用機器人自動配置

32組別 >1000支ETF

涵蓋股票、債券及各種地區和類別

資料時間

2017/1 – 2019/1

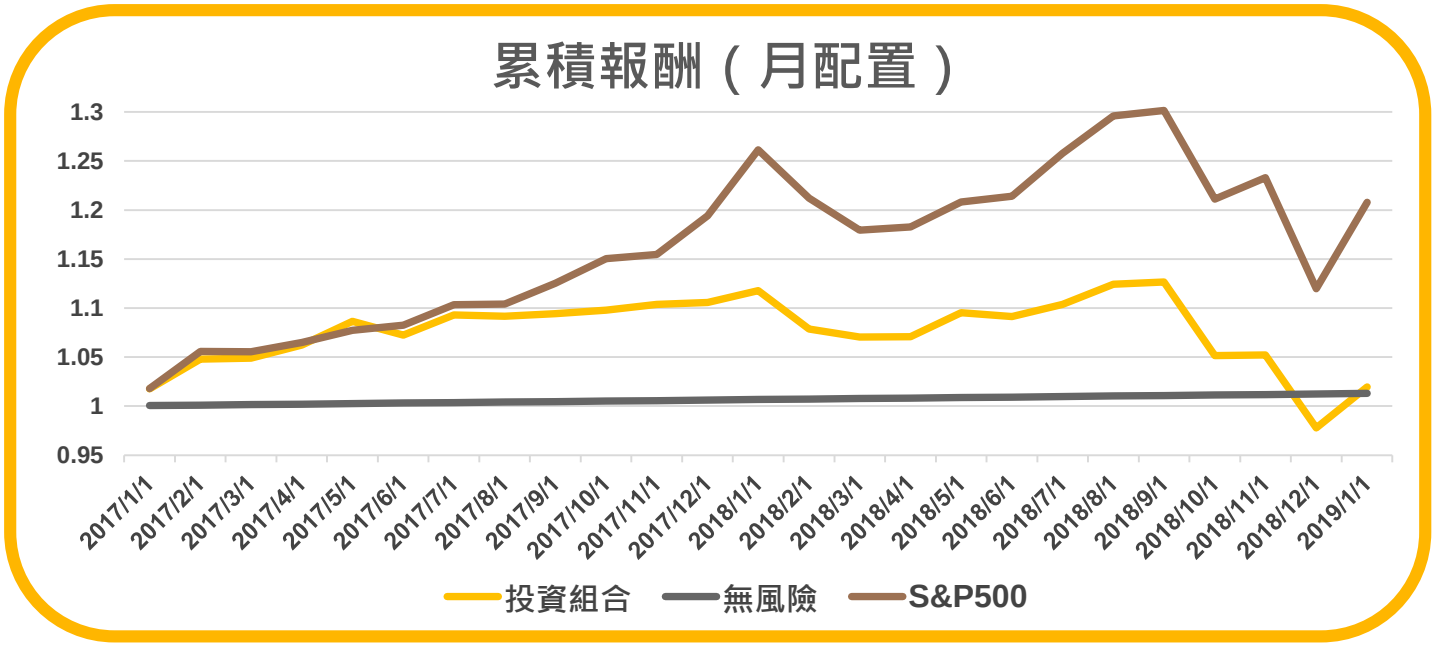
前5名

最佳的候選人

1

成果展示

績效圖

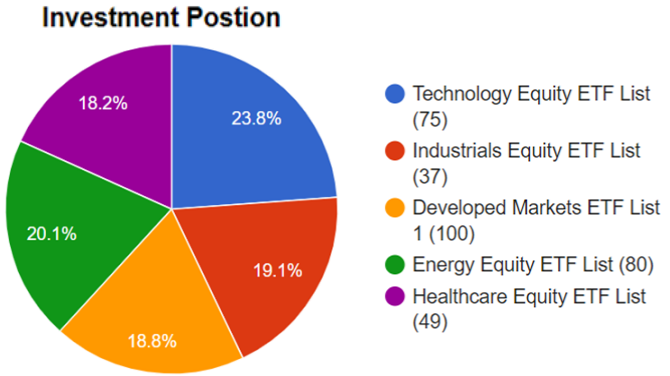

累積報酬（月配置）
1.3
1.25
1.2
1.15
1.1
1.05
1
0.95
2017/1/1
2017/2/1
2017/3/1
2017/4/1
2017/5/1
2017/6/1
2017/7/1
2017/8/1
2017/9/1
2017/10/1
2017/11/1
2017/12/1
2018/1/1
2018/2/1
2018/3/1
2018/4/1
2018/5/1
2018/6/1
2018/7/1
2018/8/1
2018/9/1
2018/10/1
2018/11/1
2018/12/1
2019/1/1
投資組合
無風險
S&P500

網頁平台

- ETF Optimized Profolio -

Jan 2019

Portfolio Pie Chart



- ETF Investment Profolio -

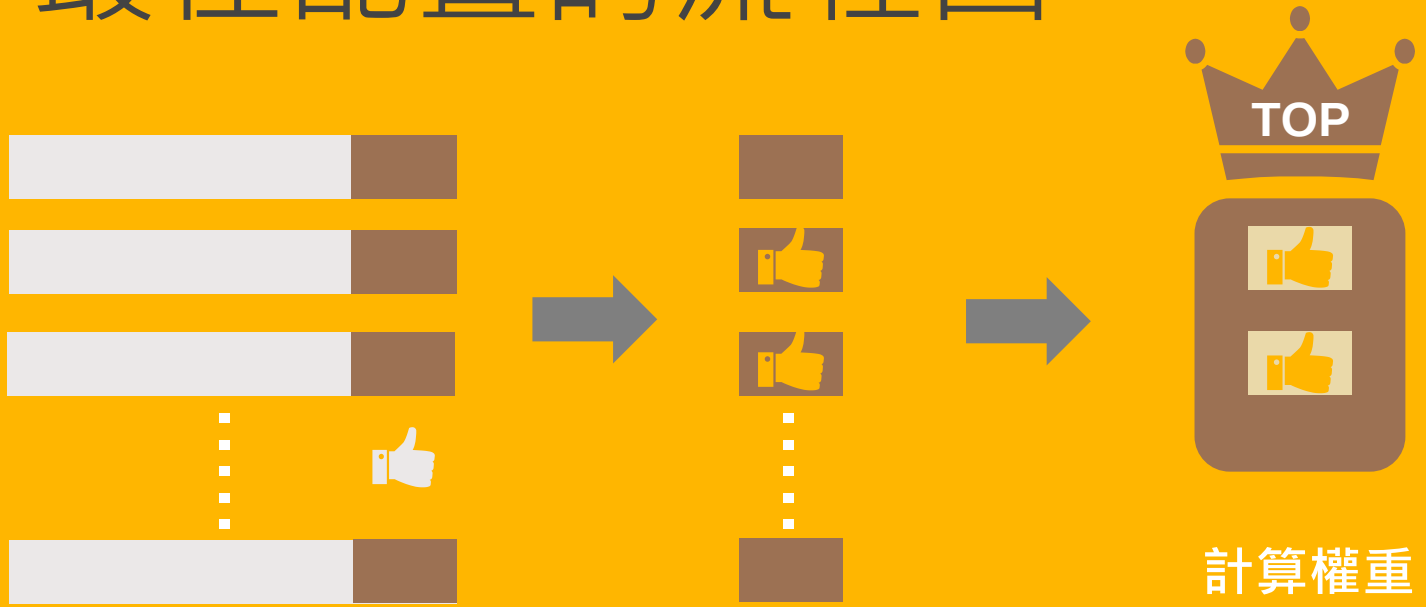
Group Changed Weight (2019-01)

	ETF Group	Last Month Weight (%)	This Month Weight (%)	Changed Weight (%)
1	Technology Equity ETF List (75)	23,026	23,849	0.823
2	Industrials Equity ETF (37)	21,913	19,093	-2.82
3	Developed Markets ETF List 1 (100)	15,907	18,788	2.881
4	Energy Equity ETF List (80)	19,679	20,503	0.374
5	Healthcare Equity ETF List (49)	19,475	18,217	-1.258

2

配置方式

最佳配置的流程圖



最適配置想法

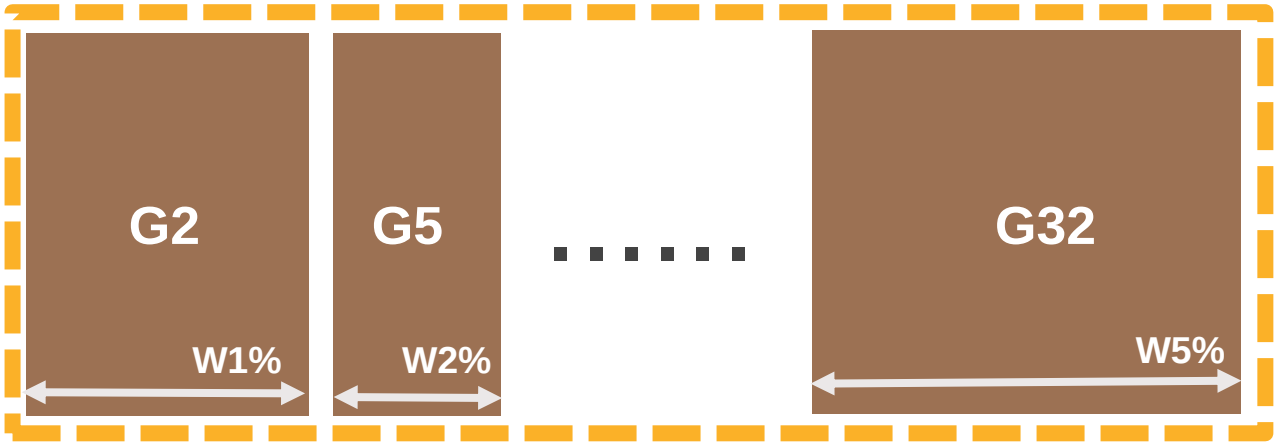
1.找出每組最好的前5名ETF做為該組的代表。

最適配置想法

2.以小組為單位，取最優秀的前五組作為最後的投資組合。

最適配置想法

3.計算投資組合的各組最適的配置比例。



評量指標 Riskiness R

定義：

$$\mathrm{E}e^{-g/R(g)} = 1.$$

性質：

R值越小代表資產越佳。

<範例 >

設資產 3 個月報酬率:x_1, x_2, x_3。

$\frac{1}{3}\sum_{i=1}^{3} e^{-x_i/R} = 1 \Rightarrow$ 我們可以求 R 值

目標: 找尋相同報酬下，風險指標最小的投資組合。

程式運算

利用

python

Fsolve()求解

計算Riskiness R

```python
def f1(x,arr_returns):
    arr_returns = arr_returns
    return np.mean( np.exp(-1*arr_returns/x) ) - 1

def get_riskiness_r(guess , arr_returns):
    while (guess<10**(2)):
        risk2 = fsolve(f1,guess,arr_returns)
        if (risk2 != guess) and (f1(risk2,arr_returns)<0.1):
            break
        guess = guess*10
    return risk2
```

Riskiness R公式

疊代計算
Riskiness R

最適配置的等式關係

利用文獻中最佳資產權重配置下的等式關係，求得最佳權重。
等式亦適用於良好性質的風險指標。

$$\frac{R_i(\alpha^*)}{\sum_{i=1}^{n}\alpha_i^* R_i(\alpha^*)} = \frac{E(z_i) - r_f}{E(\alpha^* \cdot \tilde{\mathbf{z}}) - r_f} = \mathcal{B}_i^{RAS}(\alpha) = \frac{E\left[\exp\left(-\frac{\alpha \cdot \tilde{\mathbf{z}}}{R(\alpha)}\right)\tilde{z}_i\right]}{E\left[\exp\left(-\frac{\alpha \cdot \tilde{\mathbf{z}}}{R(\alpha)}\right)\alpha \cdot \tilde{\mathbf{z}}\right]},$$

N個未知數→N條方程式

計算最佳的權重

```
def penalty(w):
    if min(w)<0 or max(w)>1:
        return 100000
    else:
        return 0

def bestweight(w,all_return):
    r = []
    r.append(penalty(w)+np.abs(sum(w)-1)*1000)
    for i in range(len(w)-1):
        v =  np.abs( risk(w[i]*(all_return.iloc[:,i]))/ sumrisk(w,all_return)  \
            - (np.mean(all_return.iloc[:,i])-rfrate)/(portretmean(w,all_return)-rfrate) ) \
            + np.abs( risk(w[i+1]*(all_return.iloc[:,i+1]))/ sumrisk(w,all_return)  \
            - (np.mean(all_return.iloc[:,i+1])-rfrate)/(portretmean(w,all_return)-rfrate))
        r.append(v)
    print('權重:',w,'誤差值:',v+penalty(w)+np.abs(sum(w)-1))
    return r
```

懲罰項

限制條件:

1. 0<權重<1
2. 權重總和=1

$A=0$、$B=0$ → $A^2+B^2=0$

未來展望

目標: 投組調整自動化

Thanks!

Q&A