
Talent vs Luck
演算法實作與視覺化資料分析
24 彭聖全

目錄

- 專題動機
- 程式實作
- 問題解決
- 心得感想

專題動機

成功取決於幸運還是才能

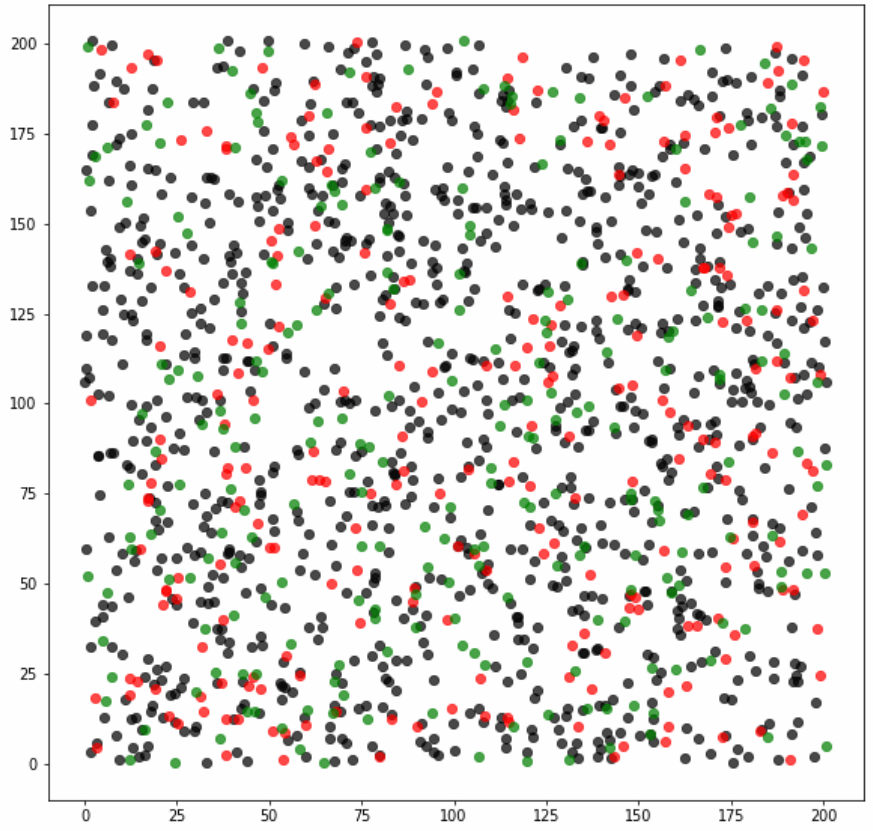
- 2022 Ig Nobel Prize for Economics
- Talent vs Luck: the role of randomness in success and failure
  - https://arxiv.org/abs/1802.07068
- 虛擬統計模型 vs 人生真實狀況
- 老高與小茉　Mr & Mrs Gao
  - https://www.youtube.com/watch?v=qzIfQ5_gYzc

程式實作

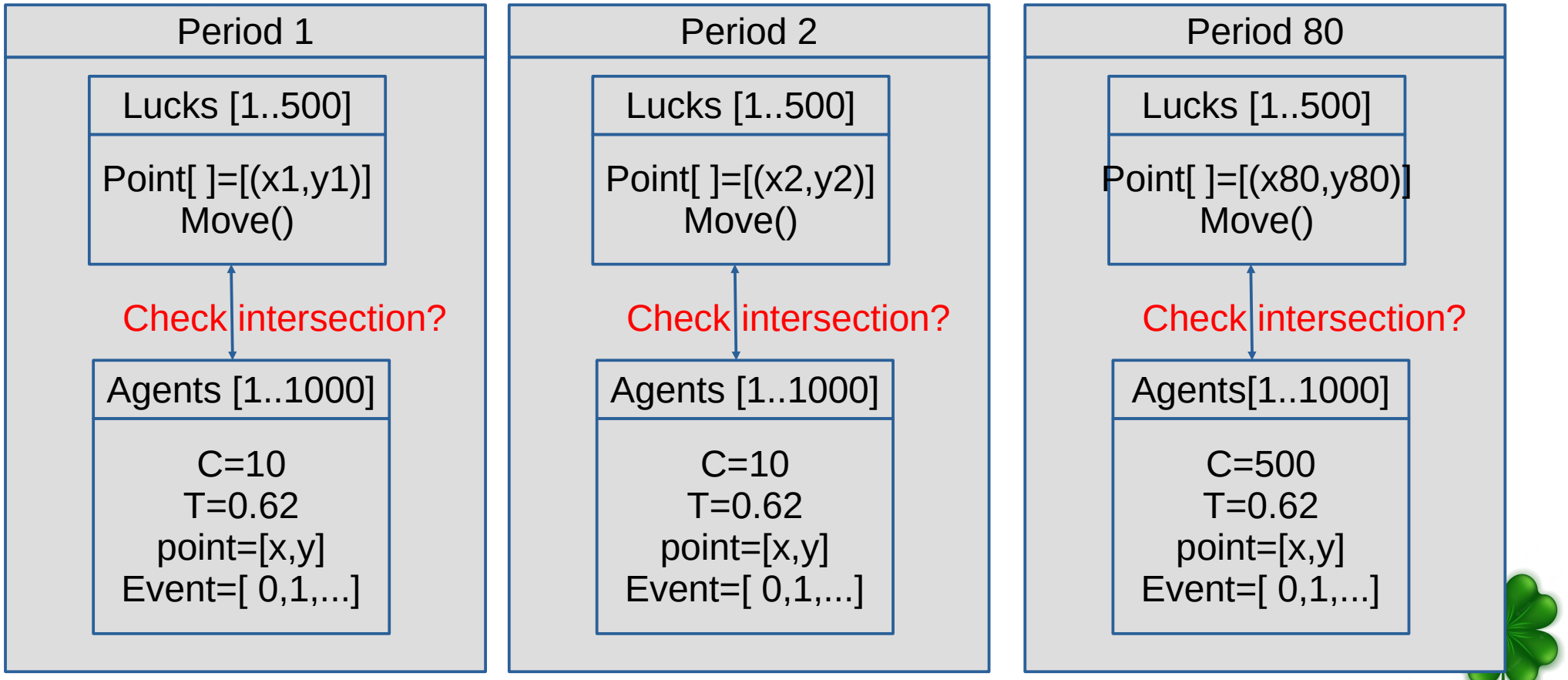
統計模型演算法實作

- 每個人一開始有成本 10 單位
- 1000 個人的才能隨機分布，所有人位置隨機散布在 201x201 的地圖上
- 250 好運球和　250 壞運球初始隨機分布，運氣球每個週期以距離為 2 的方向隨機移動
- 總共 80 個周期，一個周期假設為半年，總共人生 40 年
- 人在壞運球半徑 1 單位內，成本直接減半
- 人在好運球半徑 1 單位內，隨機產生一數值 (0<n<1)，若才能大於 n 值則成本培增，反之則產財不變
- 統計分析 1000 個人最後產財狀況

統計模型演算法實作



統計模型演算法實作


Period 1
Lucks [1..500]
Point[ ]=[(x1,y1)]
Move()
Check intersection?
Agents [1..1000]
C=10
T=0.62
point=[x,y]
Event=[0,1,...]
Period 2
Lucks [1..500]
Point[ ]=[(x2,y2)]
Move()
Check intersection?
Agents [1..1000]
C=10
T=0.62
point=[x,y]
Event=[0,1,...]
Period 80
Lucks [1..500]
Point[ ]=[(x80,y80)]
Move()
Check intersection?
Agents[1..1000]
C=500
T=0.62
point=[x,y]
Event=[0,1,...]

統計模型演算法實作

```python
# define class Agent
class Agent:
    def __init__(self, ind, C, T, Axis_x, Axis_y):
        self.Num = ind
        self.C = [-1]*(PERIOD+1);     # Capital
        self.T = T;           # Talent
        self.Event = [0]*(PERIOD+1); # 1:lucky, -1:unlucky
        self.Point = [Axis_x,Axis_y]
        self.TouchPoint =[[-1,-1,-1,-1]] #period, ball no, touch ball x, touc
        self.C[0] = C # init capital


    def __repr__(self):
        #return f"Agent_Ind:{self.Num} C:{self.C} T:{self.T} Event:{self.Even
        #return f"{self.Num},{self.C},{self.T},{self.Event},{self.Point},{sel
        if self.Num == (AGENT_NUM-1):
            return "{"+f"\"Agent_Ind\":{self.Num},\"T\":{self.T},\"Point\":{s
        else:
            return "{"+f"\"Agent_Ind\":{self.Num},\"T\":{self.T},\"Point\":{s

    def Outoput(self):
        test = self.Point[0],self.Point[1]
        print(self.Point[0],self.Point[1])
        return test

# define Luck class
class Luck:
    def __init__(self, ind, Event, Axis_x, Axis_y):
        self.Event = Event
        self.Point = [[Axis_x,Axis_y]]
        self.Num = ind
    def __repr__(self):
        if self.Num == (LUCKY_BALL_NUM/2-1) and self.Event == 1:
            return "{"+f"\"No\":{self.Num},\"Event\":{self.Event},\"Point\":{
        else:
            return "{"+f"\"No\":{self.Num},\"Event\":{self.Event},\"Point\":{
```

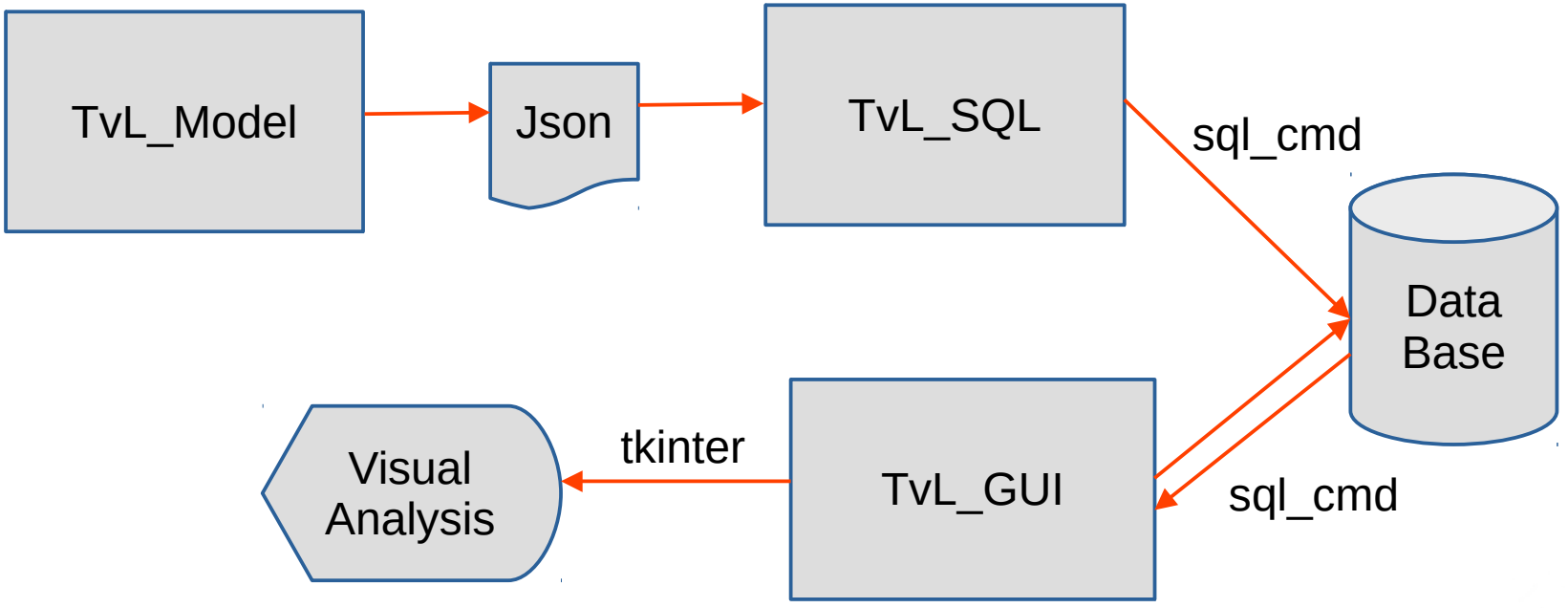
統計模型演算法實作

```
def Luckys_move(period, circle):

    lucky_circle_r = 2

    next_x = round(random.uniform(circle.Point[period-1][0]-1, circle.Point[period-1][0]+1), 2)

    if next_x <= 0 :
        next_x = 0

    if next_x >= BOUNDARY_X_Y :
        next_x = BOUNDARY_X_Y

    # next_y = round(((lucky_circle_r**2-(next_x-circle.Point[period-1][0])**2)**(0.5)+circle.Point[period-1][1]),2)

    if(0.5 >= ran_num(1)):
        next_y = round((circle.Point[period-1][1]+(lucky_circle_r**2-(next_x-circle.Point[period-1][0])**2)**(0.5)),2)
    else:
        next_y = round((circle.Point[period-1][1]-(lucky_circle_r**2-(next_x-circle.Point[period-1][0])**2)**(0.5)),2)

    if next_y <= 0 :
        next_y = 0
    if next_y >= BOUNDARY_X_Y :
        next_y = BOUNDARY_X_Y

    if (next_x == 0) or (next_x == BOUNDARY_X_Y) or (next_y == 0) or (next_y == BOUNDARY_X_Y):
         # print(next_x,next_y)
         dis = ((next_x - circle.Point[period-1][0])**2 + (next_y- circle.Point[period-1][1])**2)**(0.5)
         # print(dis)

    dis = ((next_x - circle.Point[period-1][0])**2 + (next_y- circle.Point[period-1][1])**2)**(0.5)
    circle.Point.append([next_x,next_y])
```

統計模型演算法實作

```
def Intersection_Lucks_Agents(period, circle, agent):

    Touch_dis = 1

    if agent.C[period] == -1: # first assign
        agent.C[period] = agent.C[period-1]

    if (circle.Point[period][0] > agent.Point[0]) and ((circle.Point[period][0]-agent.Point[0]) > 1.3):
        return agent.C
    if (circle.Point[period][0] < agent.Point[0]) and ((agent.Point[0] - circle.Point[period][0]) > 1.3):
        return agent.C

    if ((circle.Point[period][1] > agent.Point[1])) and ((circle.Point[period][1]-agent.Point[1]) > 1.3):
        return agent.C
    if ((circle.Point[period][1] < agent.Point[1])) and ((agent.Point[1] - circle.Point[period][1]) > 1.3):
        return agent.C
    dis = ((circle.Point[period][0]-agent.Point[0])**2+(circle.Point[period][1]-agent.Point[1])**2)**(0.5)

    if(dis <= Touch_dis):

        agent.TouchPoint.append([period, circle.Num, circle.Point[period][0],circle.Point[period][1]])
        if(circle.Event == 0x2): # touch the lucky ball
            #print("Touch Red")
            if(agent.T >= ran_num(1)): # check talent and randon value
                agent.C[period] = 2*(agent.C[period])
                # Lucky and get
                agent.Event[period] = agent.Event[period]|0x2;
            else:
                # lucky but not get
                agent.Event[period] = agent.Event[period]|0x4;

        else: # touch the unlucky ball
                agent.C[period] = (agent.C[period])/2
                # nulucky
                agent.Event[period] = agent.Event[period]|0x1;
                if(agent.C[period] < 1):
                    agent.C[period] = 0;
    #else:
    #    print("no touch")


    return agent.C
```

程式碼實作架構


TvL_Model
Json
TvL_SQL
sql_cmd
Data
Base
Visual
Analysis
tkinter
TvL_GUI
sql_cmd

Model 產生 Json 檔案資料

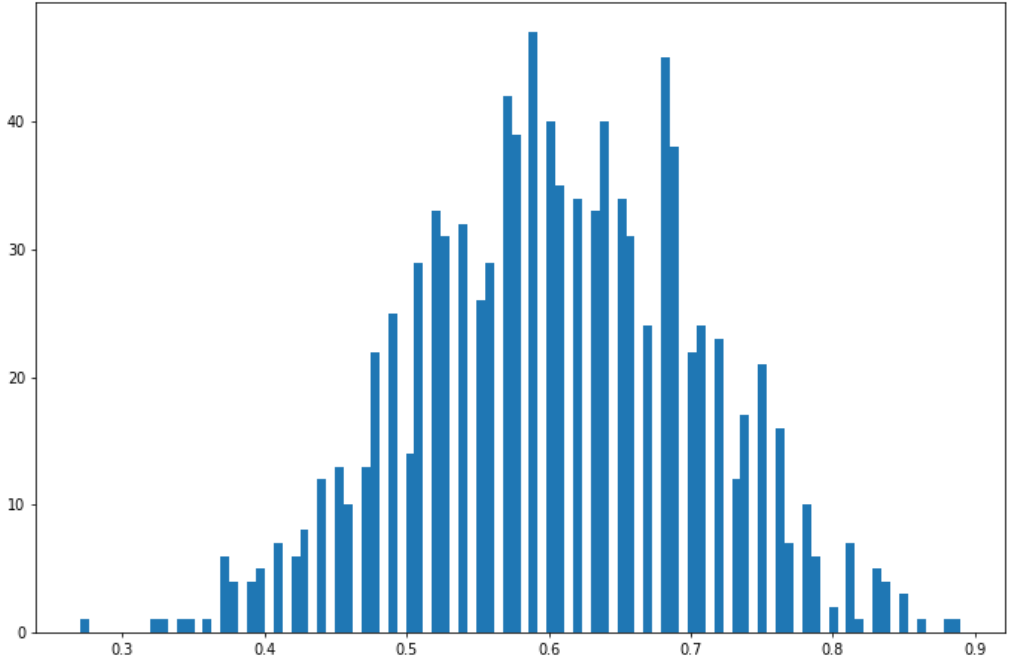

```
[{"Agent_Ind":0,
  "T":0.54,
  "Point":[152.73, 43.77],
  "C":[10, 10, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20,
  "Event":[0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  "TouchPoint":[[-1, -1, -1, -1], [2, 186, 152.73, 43.25], [41, 170, 153.24
{"Agent_Ind":1,"T":0.57,"Point":[29.38, 157.72],"C":[10, 10, 10, 10, 10, 10
{"Agent_Ind":2,"T":0.54,"Point":[185.29, 128.88],"C":[10, 10, 20, 20, 20, 2
{"Agent_Ind":3,"T":0.52,"Point":[72.82, 175.13],"C":[10, 10, 10, 10, 10, 10
{"Agent_Ind":4,"T":0.64,"Point":[4.17, 54.52],"C":[10, 10, 10, 10, 10, 10,
{"Agent_Ind":5,"T":0.67,"Point":[1.37, 136.99],"C":[10, 10, 10, 10, 10, 10,
{"Agent_Ind":6,"T":0.62,"Point":[131.84, 6.36],"C":[10, 10, 10, 10, 10, 10,
{"Agent_Ind":7,"T":0.57,"Point":[168.06, 60.23],"C":[10, 10, 10, 10, 10, 10
{"Agent_Ind":8,"T":0.54,"Point":[189.05, 31.9],"C":[10, 10, 10, 10, 10, 10,
{"Agent_Ind":9,"T":0.58,"Point":[77.72, 80.48],"C":[10, 10, 10, 10, 10, 10,
{"Agent_Ind":10,"T":0.64,"Point":[194.44, 98.36],"C":[10, 10, 10, 10, 10, 1
{"Agent_Ind":11,"T":0.64,"Point":[184.98, 144.77],"C":[10, 10, 10, 10, 10,
{"Agent_Ind":12,"T":0.57,"Point":[51.71, 29.34],"C":[10, 10, 10, 10, 10, 10
{"Agent_Ind":13,"T":0.66,"Point":[178.21, 100.22],"C":[10, 10, 10, 10, 10,
{"Agent_Ind":14,"T":0.5,"Point":[195.98, 68.14],"C":[10, 10, 10, 10, 10, 10
{"Agent_Ind":15,"T":0.51,"Point":[168.47, 93.53],"C":[10, 10, 10, 10, 10, 1
{"Agent_Ind":16,"T":0.5,"Point":[184.55, 3.91],"C":[10, 10, 10, 10, 10, 10,
{"Agent_Ind":17,"T":0.6,"Point":[183.09, 13.77],"C":[10, 10, 10, 10, 10, 10
```

資料視覺化分析

Data Base

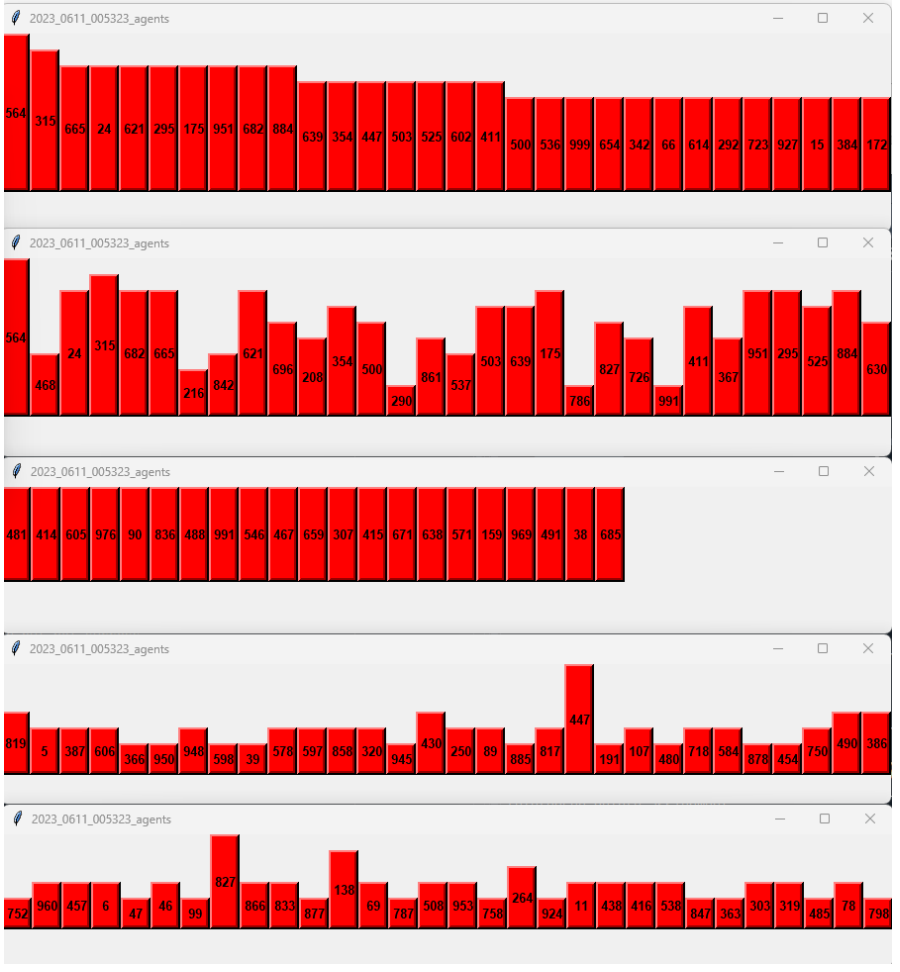
資料視覺化分析

- 1000 人的才能分佈

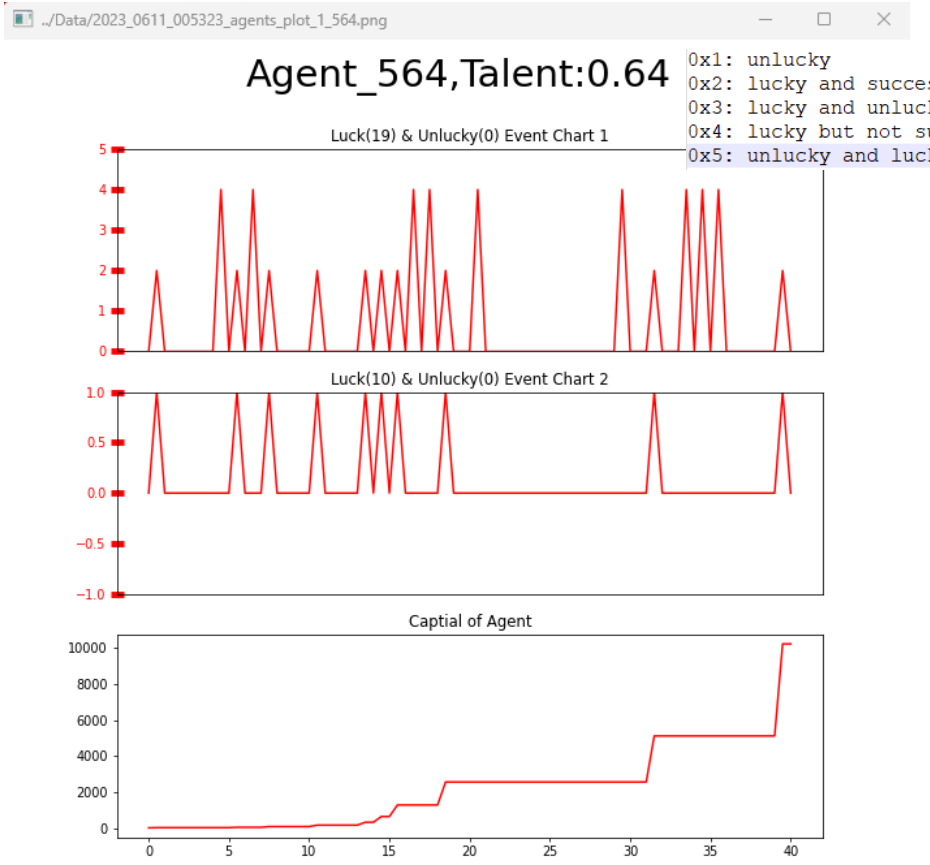


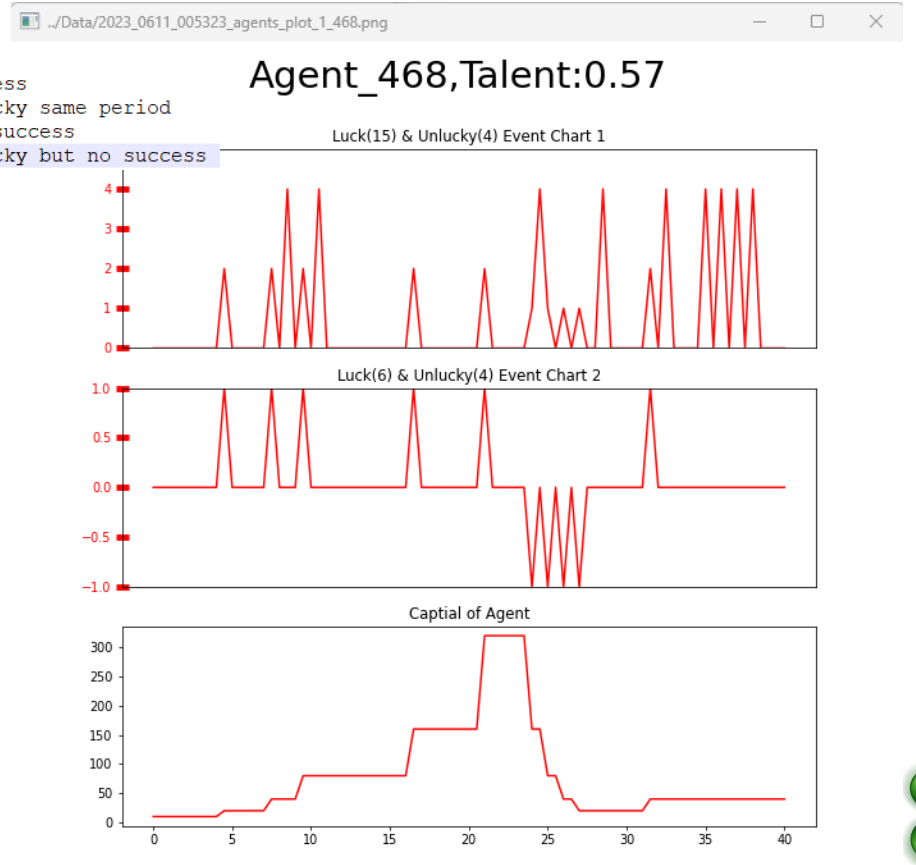
資料視覺化分析


分析圖列表
Rich Top排名
Lucky 排名
Unlucky 排名
Talent Top排名
Talent Bottom排名
Talent 分布圖
20/80 財產分布圖
Lucky & C分佈
Unlucky & C分佈
Lucky Event 分佈
Unlucky Event 分佈
All Run Tops


2023_0611_005323_agents
564 315 665 24 621 295 175 951 682 884 639 354 447 503 525 602 411 500 536 999 654 342 66 614 292 723 927 15 384 172
2023_0611_005323_agents
564 468 24 315 682 665 216 842 621 696 208 354 500 290 861 537 503 639 175 786 827 726 991 411 367 951 295 525 884 630
2023_0611_005323_agents
481 414 605 976 90 836 488 991 546 467 659 307 415 671 638 571 159 969 491 38 685
2023_0611_005323_agents
819 5 387 606 366 950 948 598 39 578 597 858 320 945 430 250 89 885 817 447 191 107 480 718 584 878 454 750 490 386
2023_0611_005323_agents
752 960 457 6 47 46 99 827 866 833 877 138 69 787 508 953 758 264 924 11 438 416 538 847 363 303 319 485 78 798

資料視覺化分析

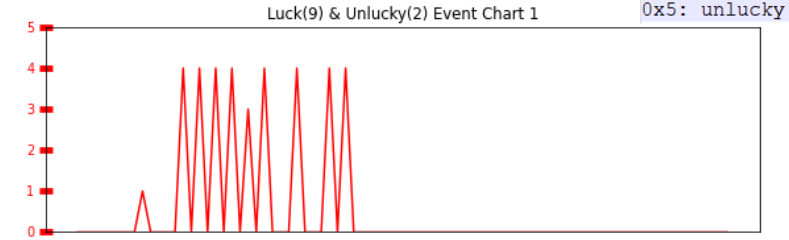

../Data/2023_0611_005323_agents_plot_1_564.png
Agent_564,Talent:0.64
0x1: unlucky
0x2: lucky and success
0x3: lucky and unlucky same period
0x4: lucky but not success
0x5: unlucky and lucky but no success
Luck(19) & Unlucky(0) Event Chart 1
Luck(10) & Unlucky(0) Event Chart 2
Captial of Agent

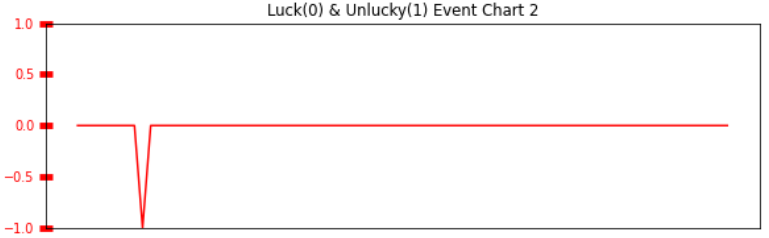

../Data/2023_0611_005323_agents_plot_1_468.png
Agent_468,Talent:0.57
Luck(15) & Unlucky(4) Event Chart 1
Luck(6) & Unlucky(4) Event Chart 2
Captial of Agent

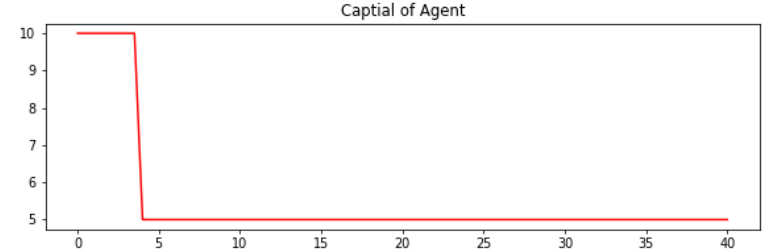
資料視覺化分析

```
0x1: unlucky
0x2: lucky and success
0x3: lucky and unlucky same period
0x4: lucky but not success
0x5: unlucky and lucky but no success
```

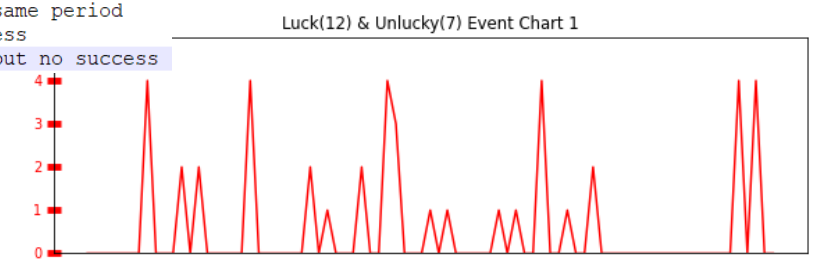
Agent_189,Talent:0.52

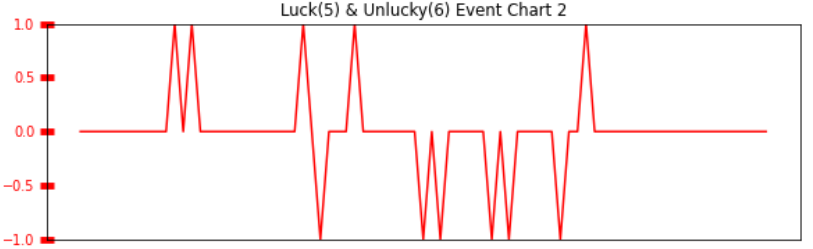


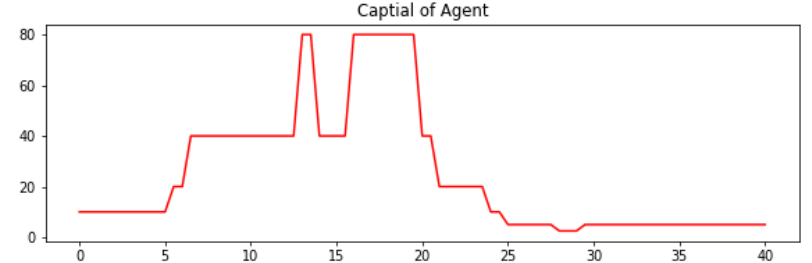




Agent_106,Talent:0.51

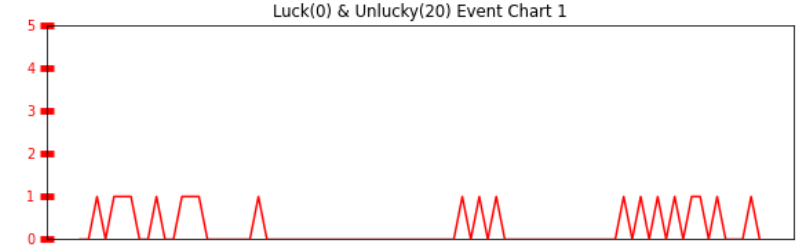


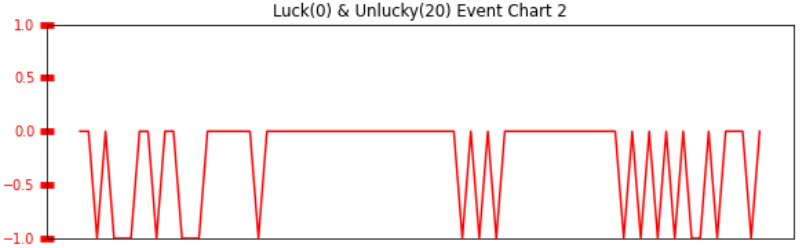


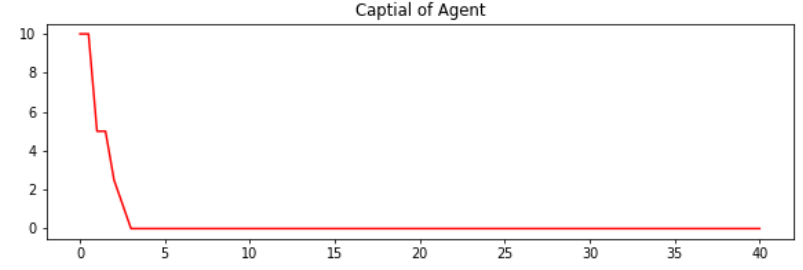


資料視覺化分析

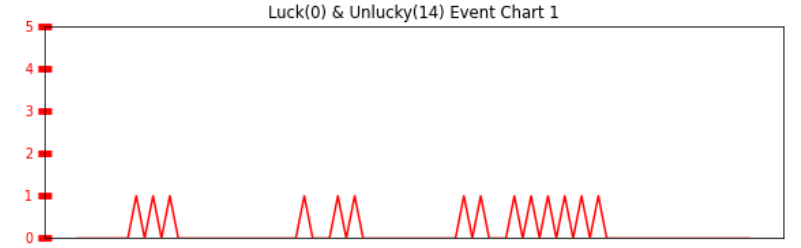
Agent_757,Talent:0.7

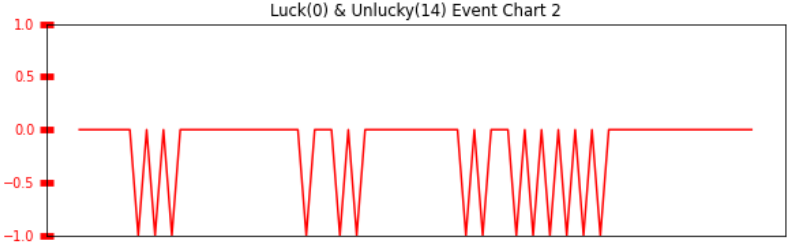


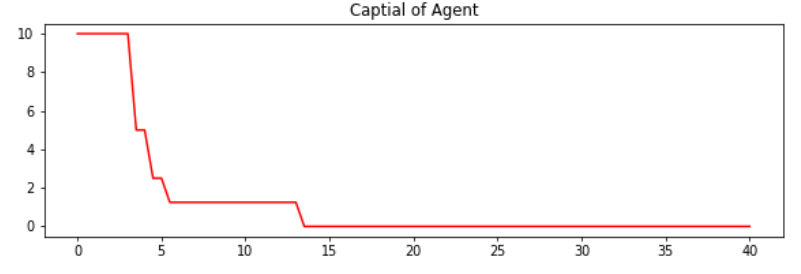




Agent_877,Talent:0.6

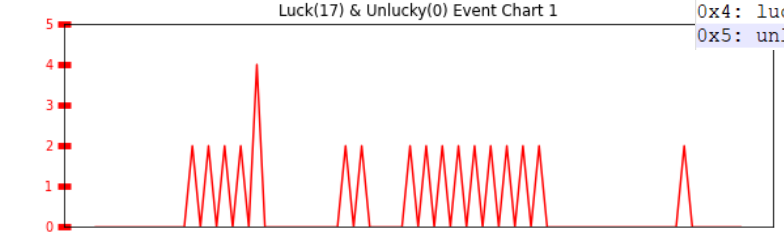


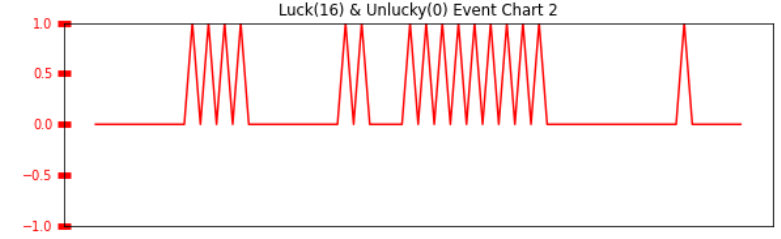


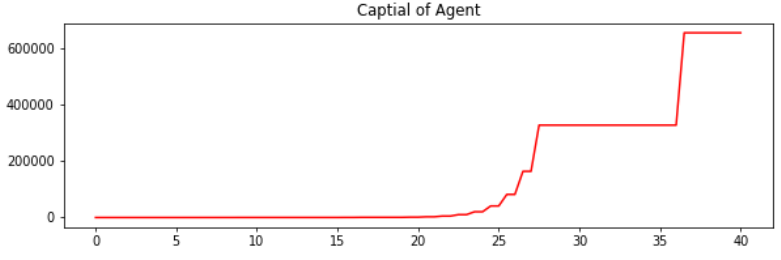


資料視覺化分析

Agent_779,Talent:0.77

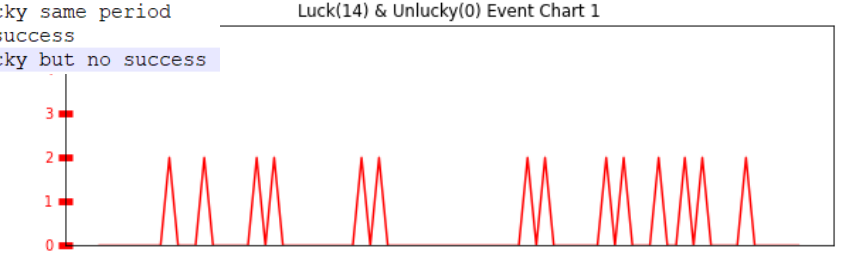


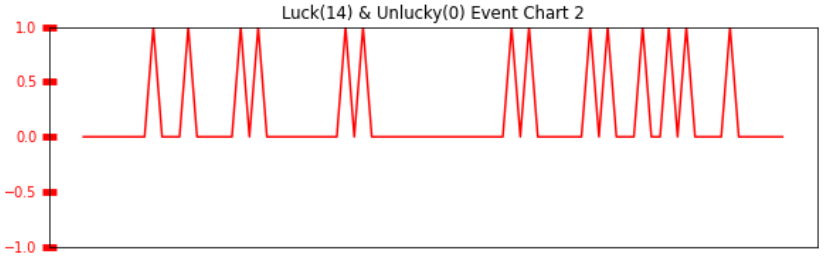


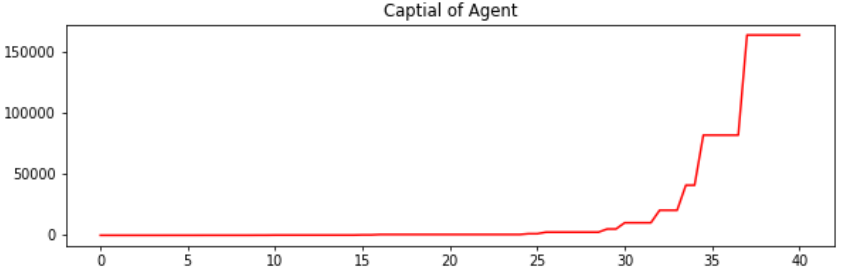



```
0x1: unlucky
0x2: lucky and success
0x3: lucky and unlucky same period
0x4: lucky but not success
0x5: unlucky and lucky but no success
```

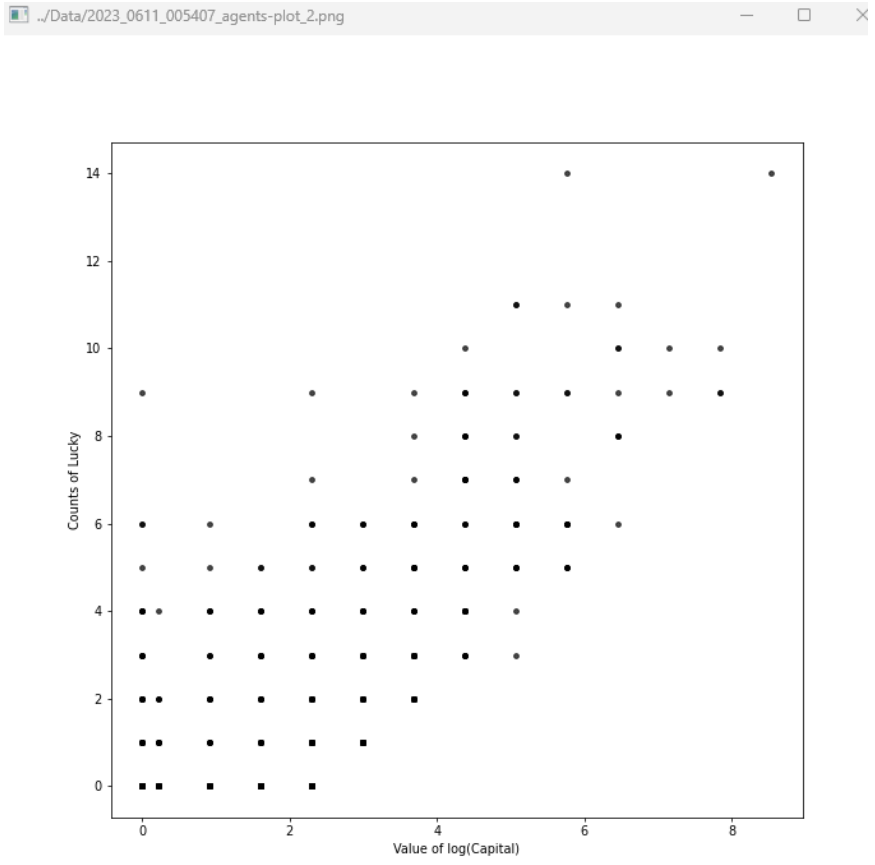
Agent_324,Talent:0.94

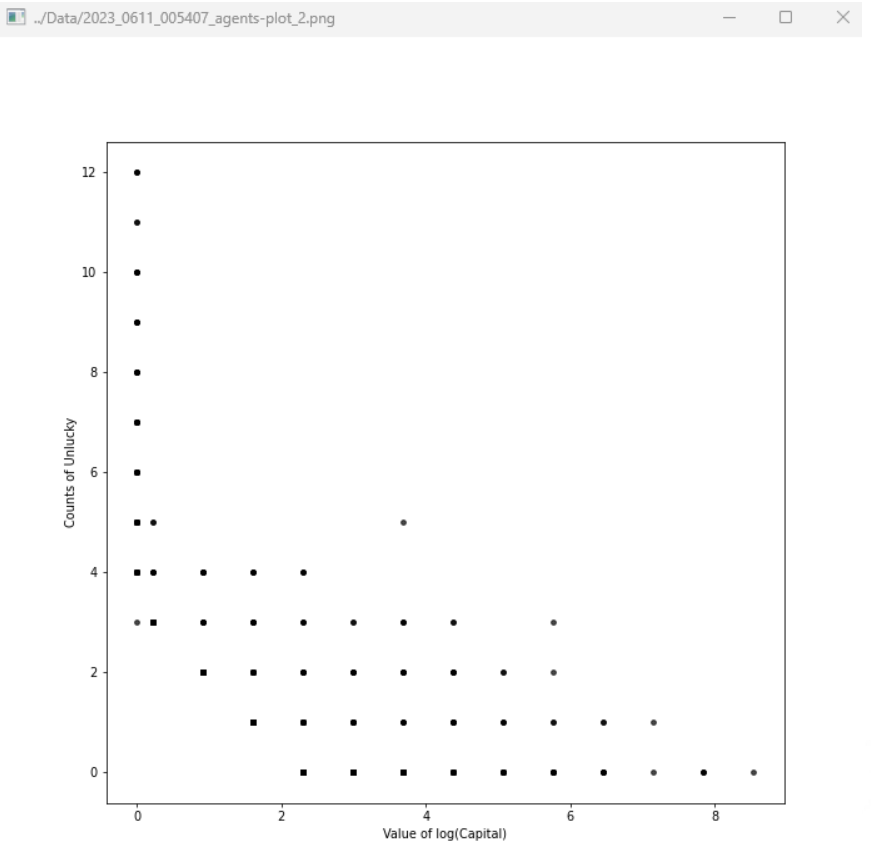






資料視覺化分析


../Data/2023_0611_005407_agents-plot_2.png
Counts of Lucky
Value of log(Capital)
0
2
4
6
8
0
2
4
6
8
10
12
14


../Data/2023_0611_005407_agents-plot_2.png
Counts of Unlucky
Value of log(Capital)
0
2
4
6
8
0
2
4
6
8
10
12

實驗分析觀察結論

- 最有錢的不一定都是最聰明
- 最聰明的人不一定都能很有錢
- 最沒錢的不一定都是最不聰明
- 運氣最差的不一定都是最不聰明
- 運氣最好的不一定都能很有錢，要有一定才能值，才有機會把握好運，讓財富 double

問題解決

遇到的問題 / 解決

- 計算距離的演算法做優化，讓執行時間從 6x 秒變 3x 秒
  - 當 agent 和 luck 計算 interaction 時，先用 x/y 距離大於 1.3 的過濾掉
- Lucky/unlucky event 計算的方法覺得不夠明確，改進論文的計算方式
- TvL_GUI Button 實作一層呼叫一層， callback function 實作比較複雜
- 分析資料都要從 database parse 下來後再做分析，這樣執行速度才會快，所以上傳至 database 的資料可以先統整過
  - 先計算 lucky/unlucky event 的數量，上傳至 DB 增加欄位，供之後分析使用
- 1000 人的才能分佈要達到標準差 0.6 正負 0.1
  - for i in range(1000):

    nums.append(random.normalvariate(mu=0.6, sigma=0.1))

遇到的問題 / 解決

- 程式資料上傳 DB
  - 將 model 的資料製作成 json ，方便之後可用 df 直接存取
- Agent data 裡的 C 和 EVENT 因為週期有 80 個欄位
  - 在寫 SQL cmd 時有點麻煩，應該可以再改進
- 使用 cv2 imshow 無法同時開啟多張圖片
  - cv2 的架構問題，目前還解不了，要再找其他方法

心得感想

心得感想

- Python 作資料分析很方便有效
- 程式架構邊作邊想邊修改，先開始才會有想法，然後一步一步再優化
- 有問題都可以問 GOOGLE 大神，一定有其他人遇到相同的問題
- 可用 GIT HUB 管理上 code 記錄很方便
  - https://github.com/pon0531/my_git/tree/main/Project_8

心得感想

- 三分天注定，七分靠打拼，愛拼才會贏
- 機會是留給準備好的人