PUBG 遊戲數據分析

資料科學第8組

黎濟毅、賴晨禾

郭冠廷、張順益、孫崇斌

Contents

1、遊戲介紹

2、資料介紹

3、分析目標及方法

4、資料預處理

5、模型建立

6、分析困難

7、分析結論

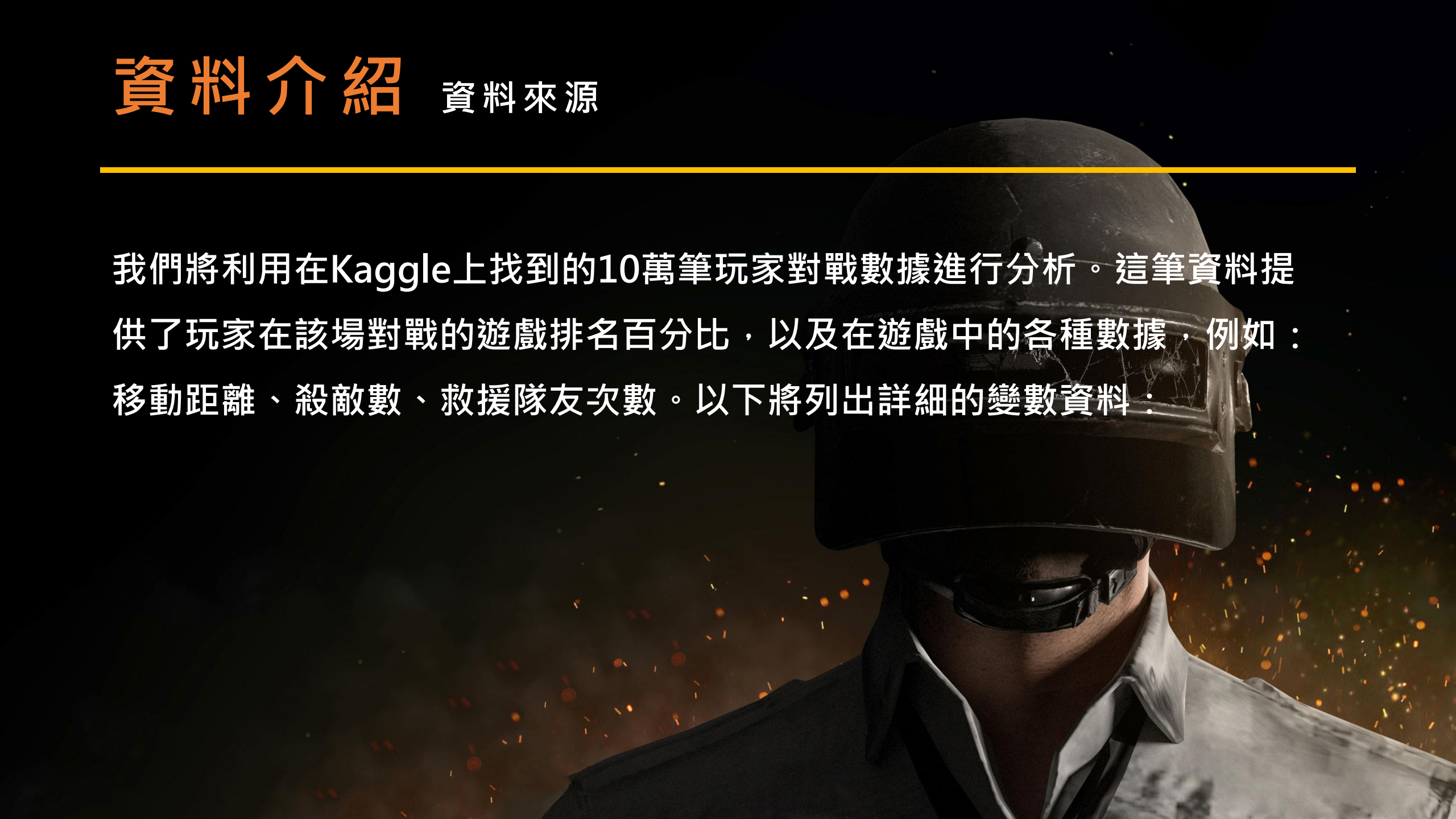
遊戲介紹

每場遊戲都會有約100位的玩家，並且玩家可在進入遊戲前選擇1人、2人、4人...等組隊模式進行對戰。遊戲開始時，每位玩家身上都沒有任何武器，需透過撿取地圖上隨機散落的武器來殺敵，能夠活到最後的人或隊伍就能獲得吃雞的稱號（第一名）。

此外遊戲中還分為**一般休閒模式**與**牌位積分模式**，在一般休閒模式下，玩家的輸贏不會有懲罰；而在排位積分模式下，積分可以讓玩家獲得遊戲牌位，而玩家透過贏得對戰來獲得積分，且過低的排名可能會造成積分的下降。

資料介紹 資料來源

我們將利用在Kaggle上找到的10萬筆玩家對戰數據進行分析。這筆資料提供了玩家在該場對戰的遊戲排名百分比，以及在遊戲中的各種數據，例如：移動距離、殺敵數、救援隊友次數。以下將列出詳細的變數資料：



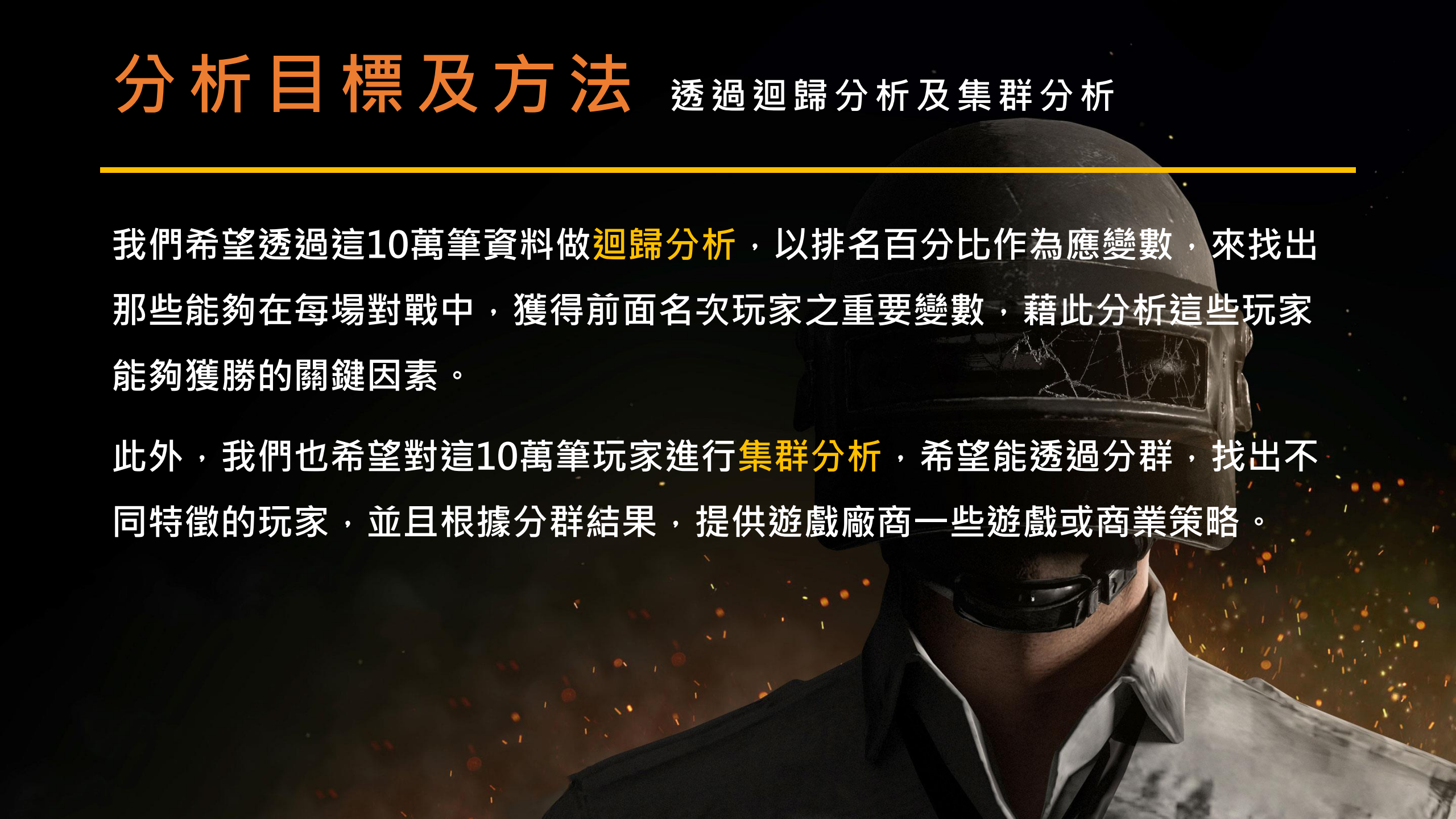
資料介紹 變數介紹

matchType	遊戲進行模式(分為第一、三人稱，單人、雙人、四人)	DBNOs	被擊倒的敵方玩家數量
rankPoints	一種隱藏分數的制度，目前已停用	assists	對敵方玩家造成傷害但最後由隊友所擊殺
revives	救活隊友的次數	boosts	使用補品的個數(能量飲料、止痛藥、腎上腺素)
rideDistance	使用交通工具移動的總距離	damageDealt	造成的總傷害量
roadKills	在交通工具上所完成的擊殺	headshotKills	爆頭擊殺的數量
swimDistance	以游泳方式所移動的總距離	heals	使用補包的數量(繃帶、急救包、急救箱)
teamKills	擊殺隊友的次數	killPlace	該局遊戲以擊殺數所形成的排名
vehicleDestroys	破壞交通工具的次數	killPoints	以擊殺數去參考的外部排名
walkDistance	以走路方式所移動的總距離	killStreaks	短時間內所造成的最大連續擊殺
weaponsAcquired	拾取的武器數量	kills	擊殺敵方玩家的數量
winPoints	以獲勝去參考的外部排名(即存活到最後)	longestKill	兩人之間最長距離所形成的擊殺
groupId	小組id	matchDuration	單局遊戲進行時間
numGroups	參與遊戲中的小組數量	matchId	遊戲房間名稱
maxPlace	遊戲中的最差排名，這可能與numGroups不匹配，因為有時數據會跳過展示位置。	winPlacePerc	預測的目標。 這是一個百分位獲勝位置，其中1對應於第一名，0對應於比賽中的最後一名

分析目標及方法 透過迴歸分析及集群分析

我們希望透過這10萬筆資料做**迴歸分析**，以排名百分比作為應變數，來找出那些能夠在每場對戰中，獲得前面名次玩家之重要變數，藉此分析這些玩家能夠獲勝的關鍵因素。

此外，我們也希望對這10萬筆玩家進行**集群分析**，希望能透過分群，找出不同特徵的玩家，並且根據分群結果，提供遊戲廠商一些遊戲或商業策略。

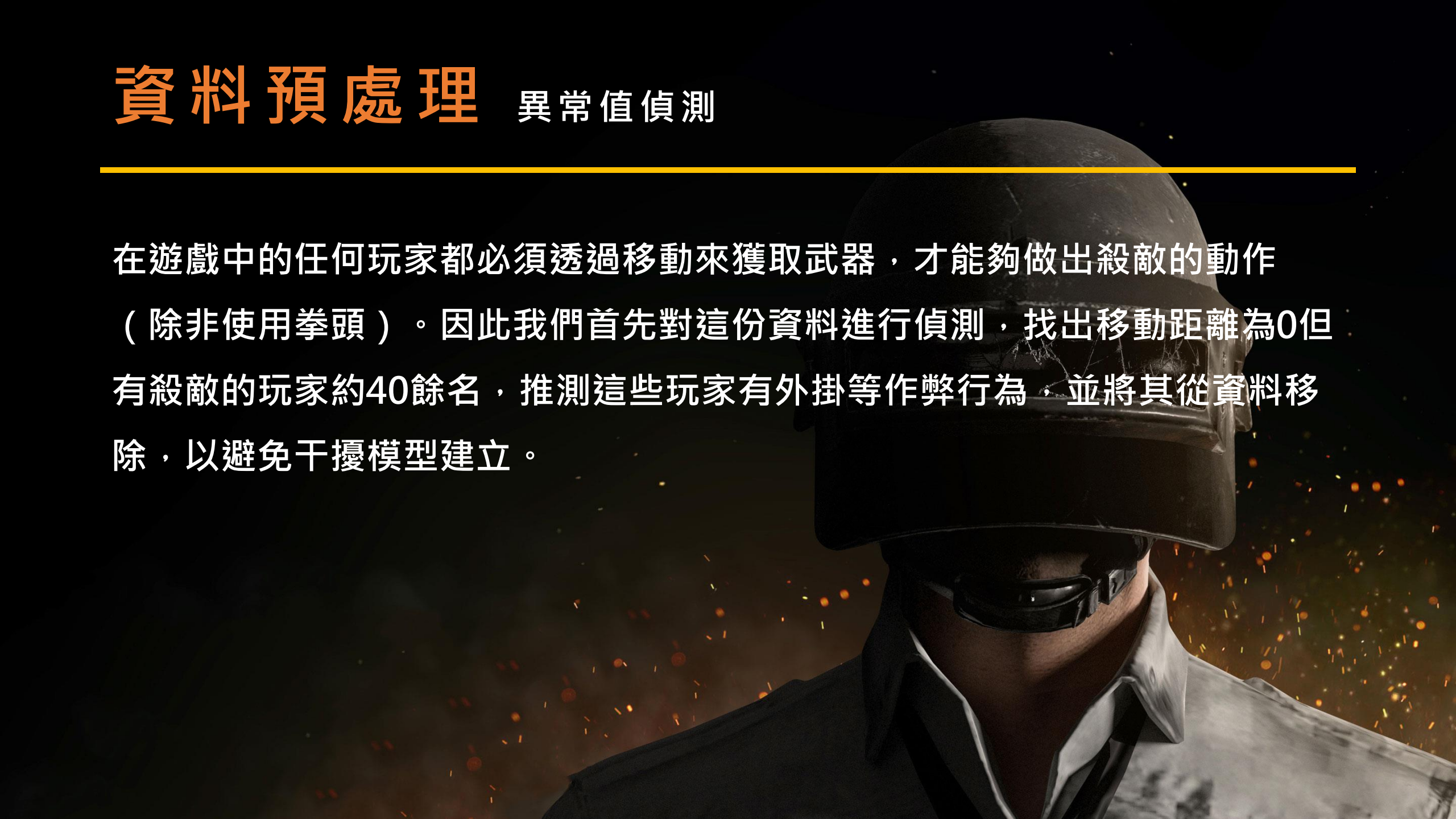


資料預處理 缺失值檢測

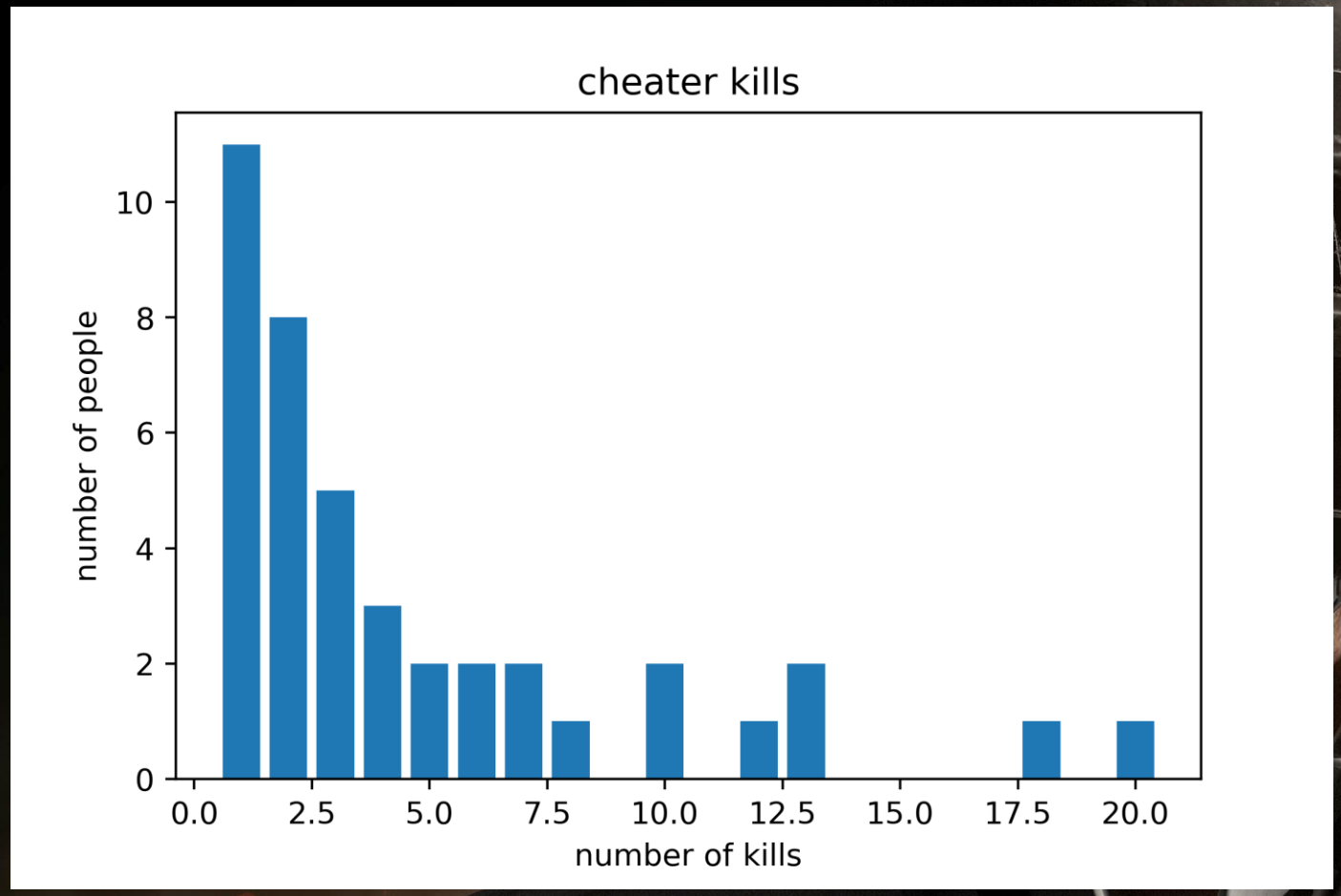
本次的資料在缺失值的表現上面十分乾淨，我們並未發現任何的缺失值。

資料預處理 異常值偵測

在遊戲中的任何玩家都必須透過移動來獲取武器，才能夠做出殺敵的動作（除非使用拳頭）。因此我們首先對這份資料進行偵測，找出移動距離為0但有殺敵的玩家約40餘名，推測這些玩家有外掛等作弊行為，並將其從資料移除，以避免干擾模型建立。



資料預處理 異常值偵測


cheater kills
number of people
0
2
4
6
8
10
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
number of kills

資料預處理 比例標準化

由於我們稍後將使用Kmeans進行集群分析，因此先對資料進行MinMax的標準化，將所有變數資料控制在0~1的範圍內，以防Kmeans因為變數尺度的不同而有偏誤。

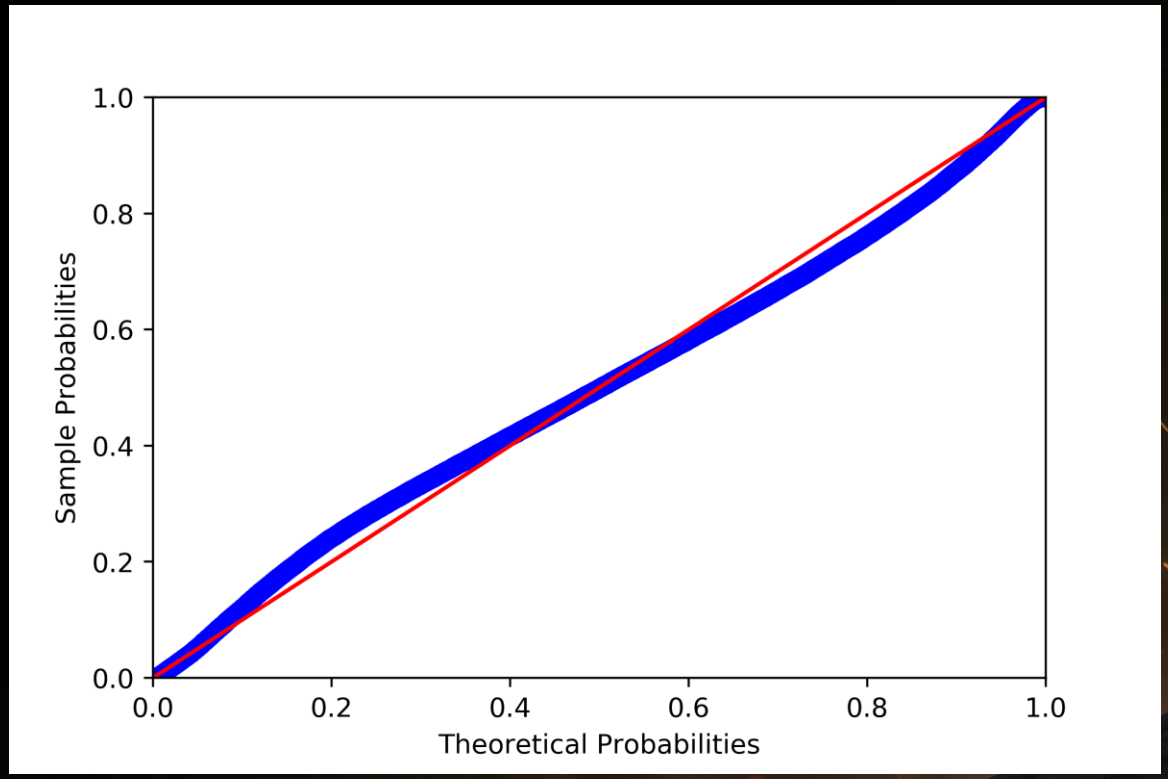
Out[320]:

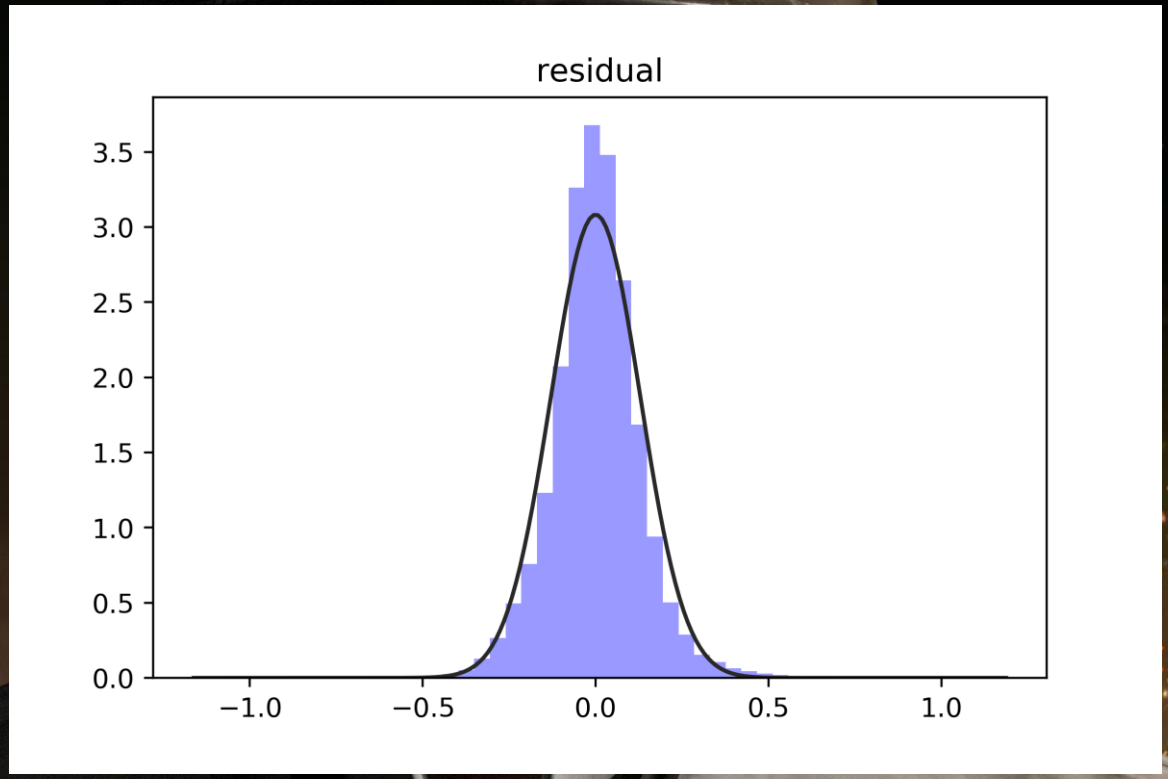
	assists	boosts	...	winPoints	winPlacePerc
0	0.000000	0.0	...	0.743408	0.4444
1	0.000000	0.0	...	0.000000	0.6400
2	0.076923	0.0	...	0.000000	0.7755
3	0.000000	0.0	...	0.000000	0.1667
4	0.000000	0.0	...	0.000000	0.1875

模型建立

線性迴歸　R -squared:0.841

lm <- lm(winPlacePerc ~ ., data = df)　模型殘差大致為常態

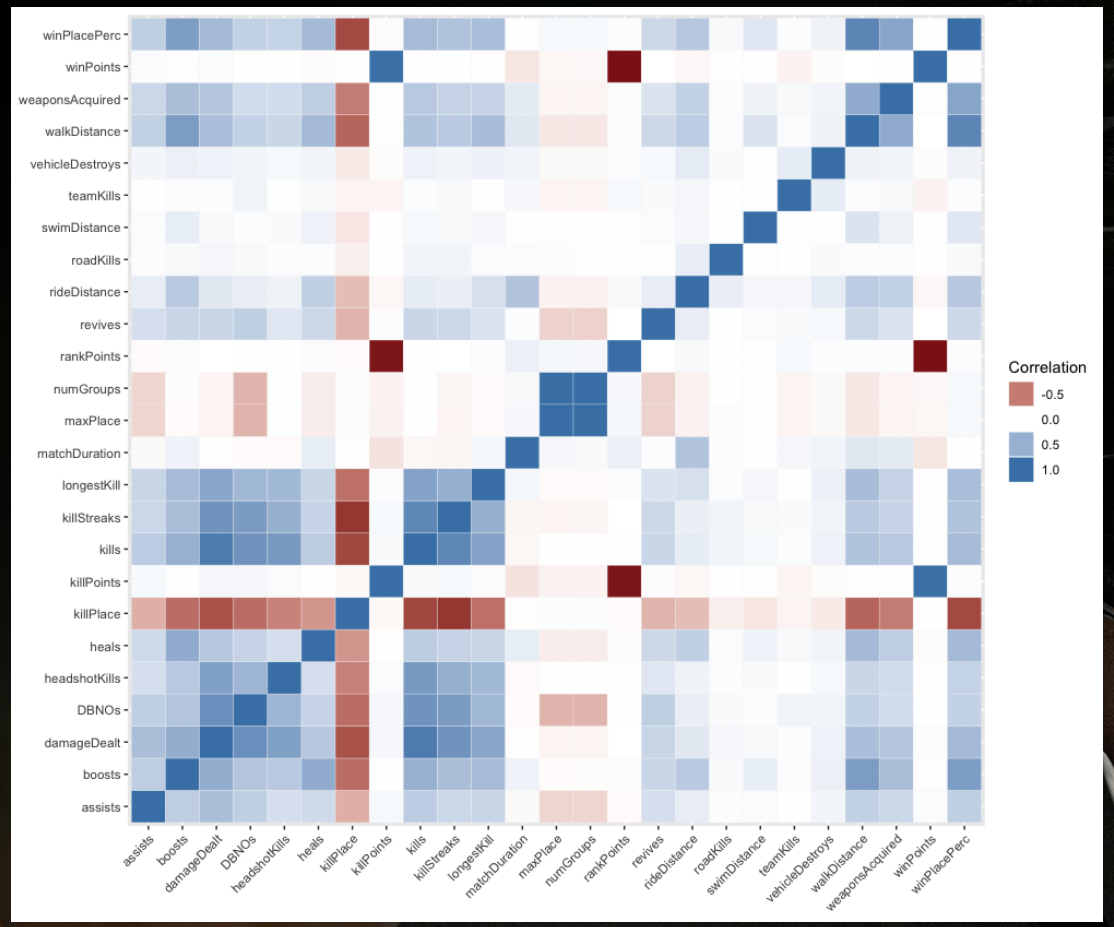




模型建立

線性迴歸　R -squared:0.841

lm <- lm(winPlacePerc ~ ., data = df)　有共線性問題！

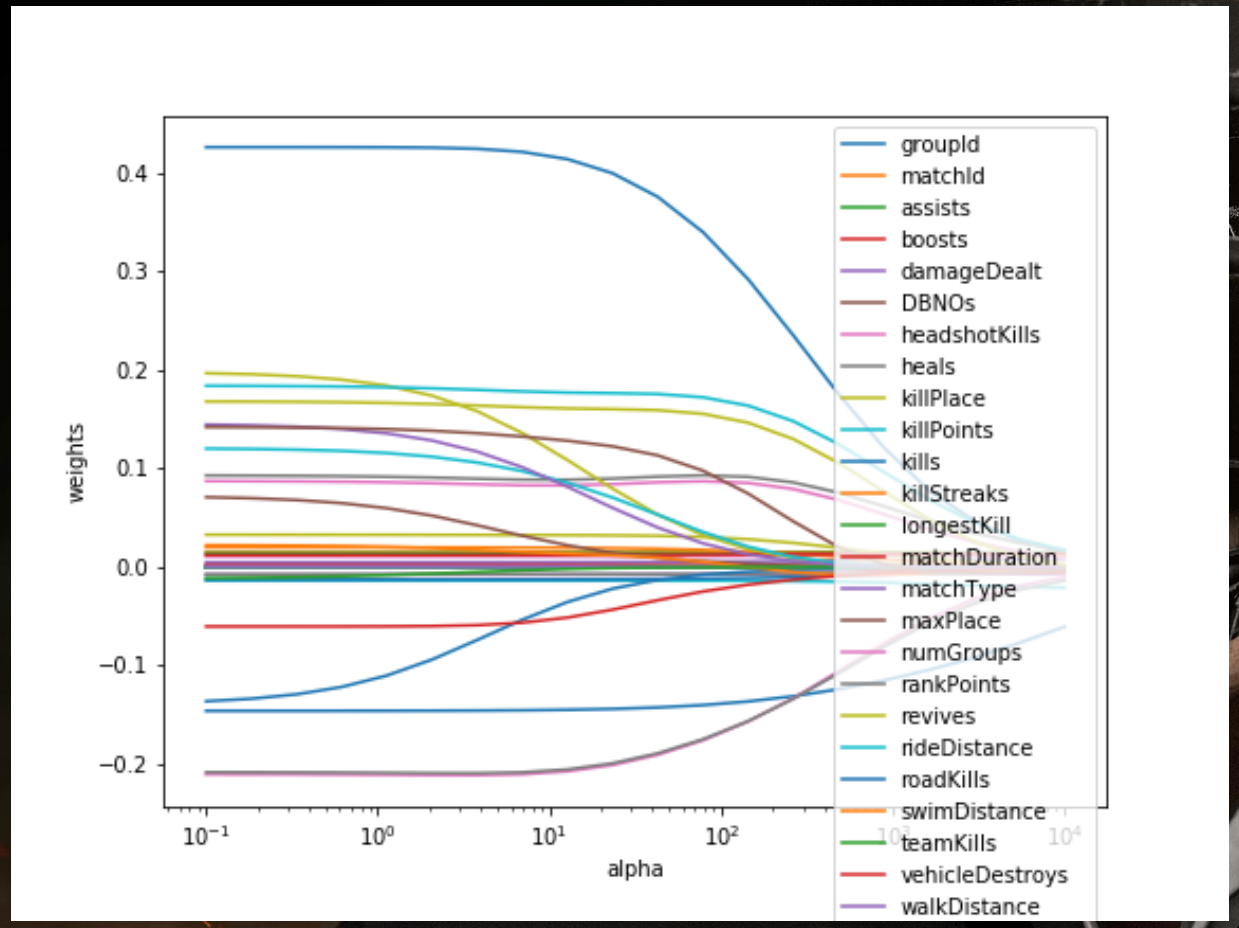


	VIF Factor	features
0	1.502852	assists
1	3.287822	boosts
2	9.750098	damageDealt
3	3.787694	DBNOs
4	2.104449	headshotKills
5	1.892824	heals
6	21.230310	killPlace
7	54.630058	killPoints
8	11.134582	kills
9	7.855826	killStreaks
10	2.105305	longestKill
11	48.257315	matchDuration
12	1643.125230	maxPlace
13	1235.214333	numGroups
14	149.802846	rankPoints
15	1.313373	revives
16	1.637316	rideDistance
17	1.046947	roadKills
18	1.061758	swimDistance
19	1.049279	teamKills
20	1.044416	vehicleDestroys
21	5.061685	walkDistance
22	5.400638	weaponsAcquired
23	165.675919	winPoints
24	1.016494	crashtpp
25	23.645416	duo
26	74.386356	duo-fpp
27	1.049936	flarefpp
28	1.115896	flaretpp
29	1.014292	normal-duo
30	1.364464	normal-duo-fpp
31	1.007931	normal-solo
32	1.131714	normal-solo-fpp
33	1.058973	normal-squad
34	2.083180	normal-squad-fpp
35	25.445464	solo
36	75.887966	solo-fpp
37	40.254240	squad
38	111.585242	squad-fpp

模型建立

脊迴歸　R -squared:0.8404

脊迴歸是一種修改最小平方法，允許有偏估計量，進而改善多元共線性的方法



模型建立

脊迴歸　R-squared:0.8404

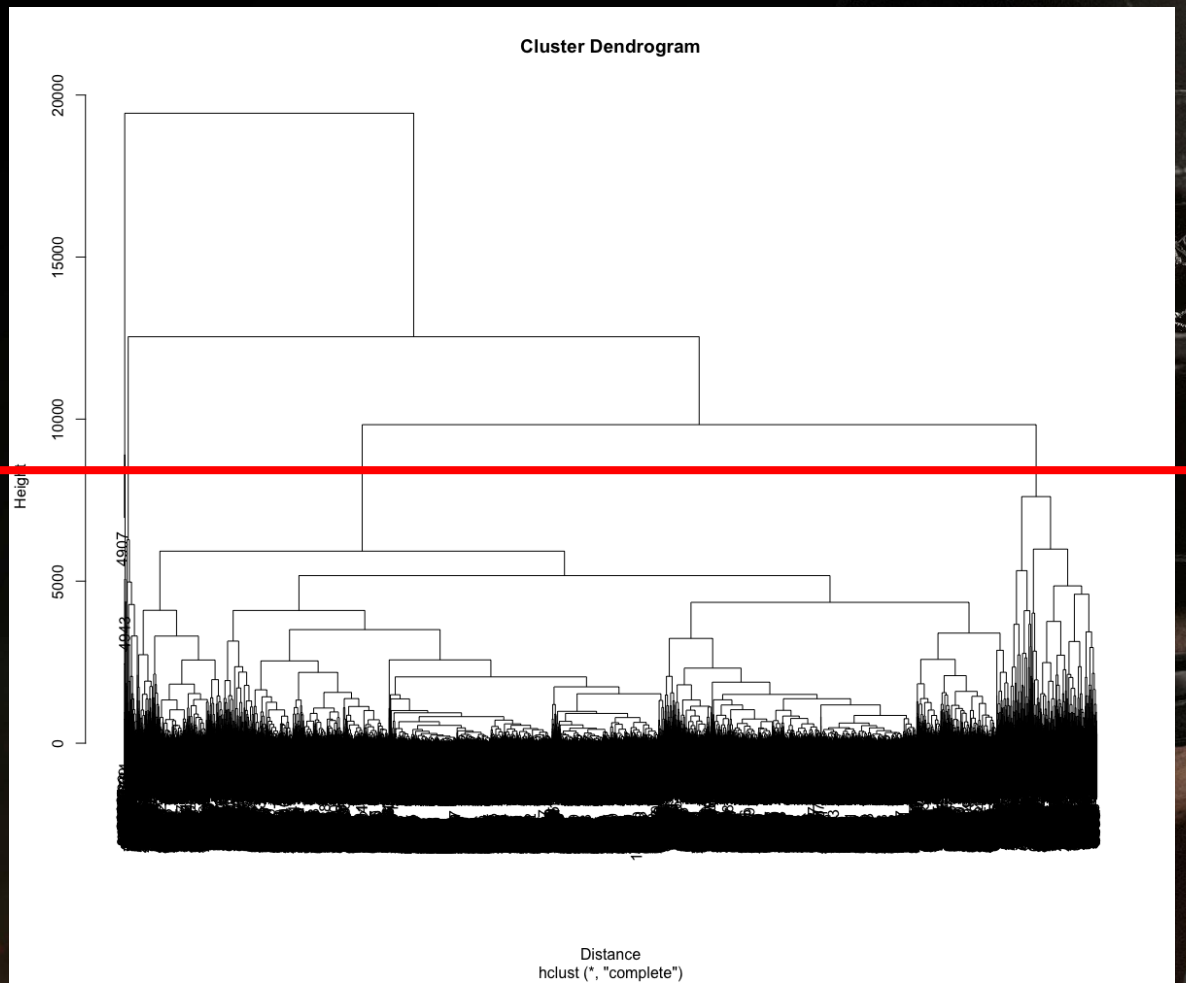
比較特別的發現是，從脊回歸中可以發現與殺敵有關的變數，跟排名大部分都呈現負相關。這樣的結果較違反我們的認知，我們推論可能的原因是，在此遊戲中，**殺人不一定是最好的獲勝策略，一昧的見人就殺反而更容易失敗**；適當的躲避敵人可能會對於玩家的排名更有利。

也就是說，「伏地魔」、「龜到底」這種為人詬病的戰術，的確有可能讓玩家更易致勝？！

0	const	0.455644
1	assists	0.014813
2	boosts	0.014040
3	damageDealt	0.000073
4	DBNOs	-0.007760
5	headshotKills	0.001467
6	heals	0.000822
7	killPlace	-0.007426
8	killPoints	-0.000055
9	kills	-0.013959
10	killStreaks	-0.145786
11	longestKill	-0.000014
12	matchDuration	-0.000164
13	maxPlace	0.001596
14	numGroups	0.004971
15	rankPoints	0.000106
16	revives	0.010716
17	rideDistance	0.000018
18	roadKills	0.030854
19	swimDistance	0.000088
20	teamKills	-0.012843
21	vehicleDestroys	0.019446
22	walkDistance	0.000111
23	weaponsAcquired	0.012011
24	winPoints	0.000151
25	crashtpp	0.019447
26	duo	0.067660
27	duo-fpp	0.073310
28	flarefpp	0.108939
29	flaretpp	0.076575
30	normal-duo	-0.047863
31	normal-duo-fpp	-0.002297
32	normal-solo	-0.005432
33	normal-solo-fpp	-0.068366
34	normal-squad	0.077472
35	normal-squad-fpp	0.111653
36	solo	-0.219639
37	solo-fpp	-0.217451
38	squad	0.143908
39	squad-fpp	0.160038

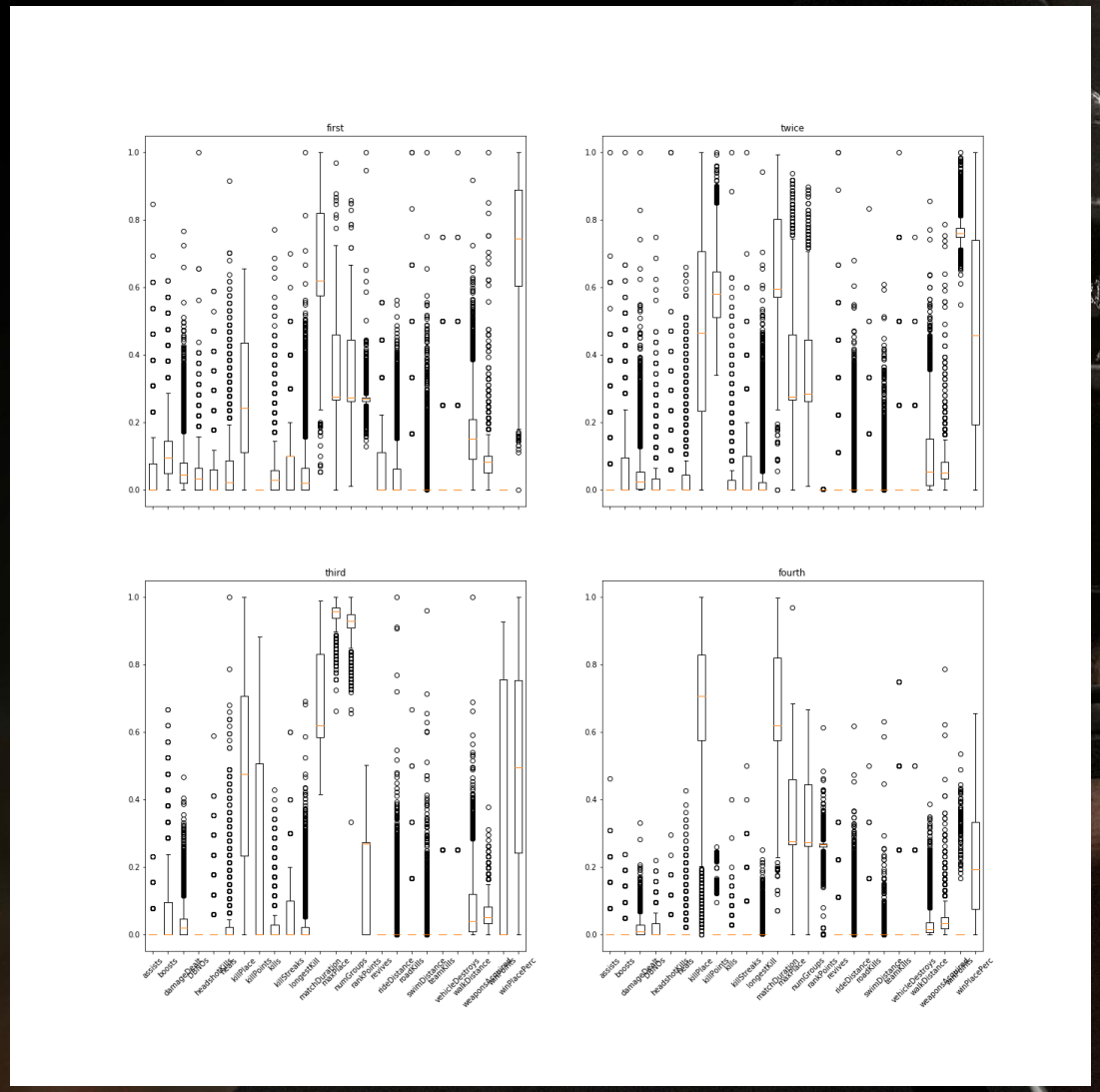
模型建立 集群分析 – 選擇最適分群數

我們首先利用階層式集群分析，來初步判斷適合的分層數，最後判斷約為4群



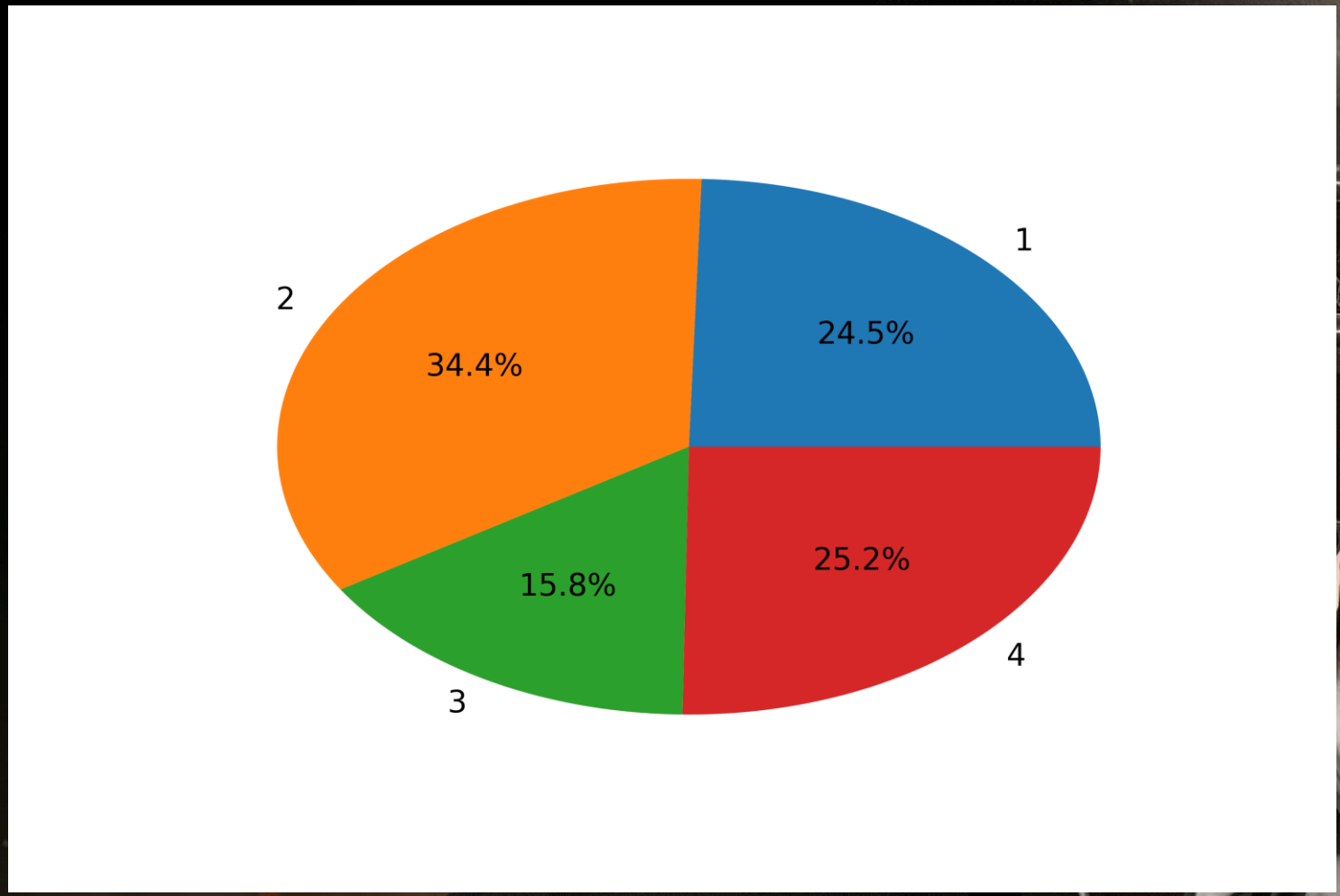
模型建立

集群分析 - 分群結果BoxPlot



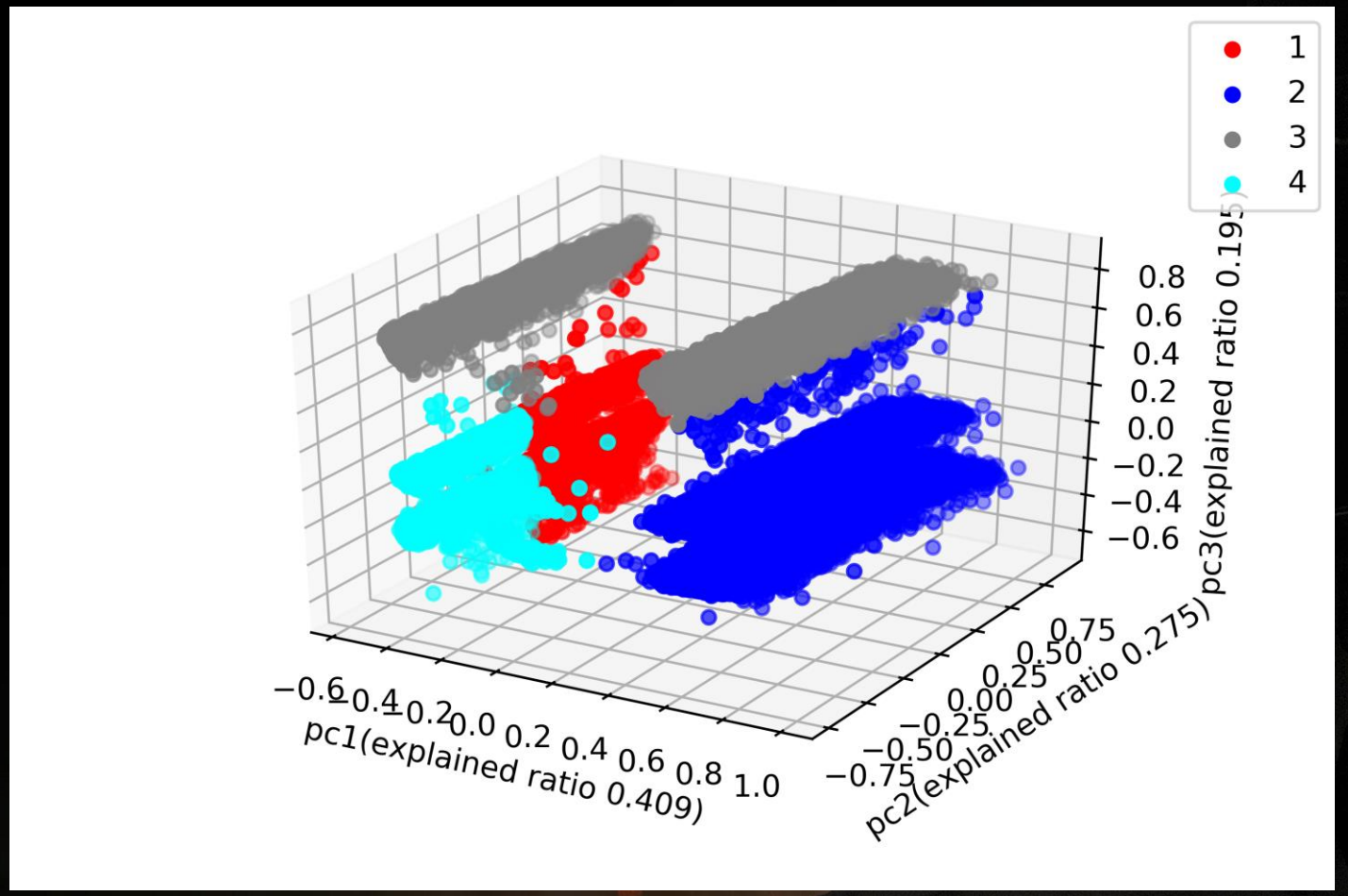
模型建立

集群分析 - 分群比例



模型建立

集群分析 - 分群視覺化

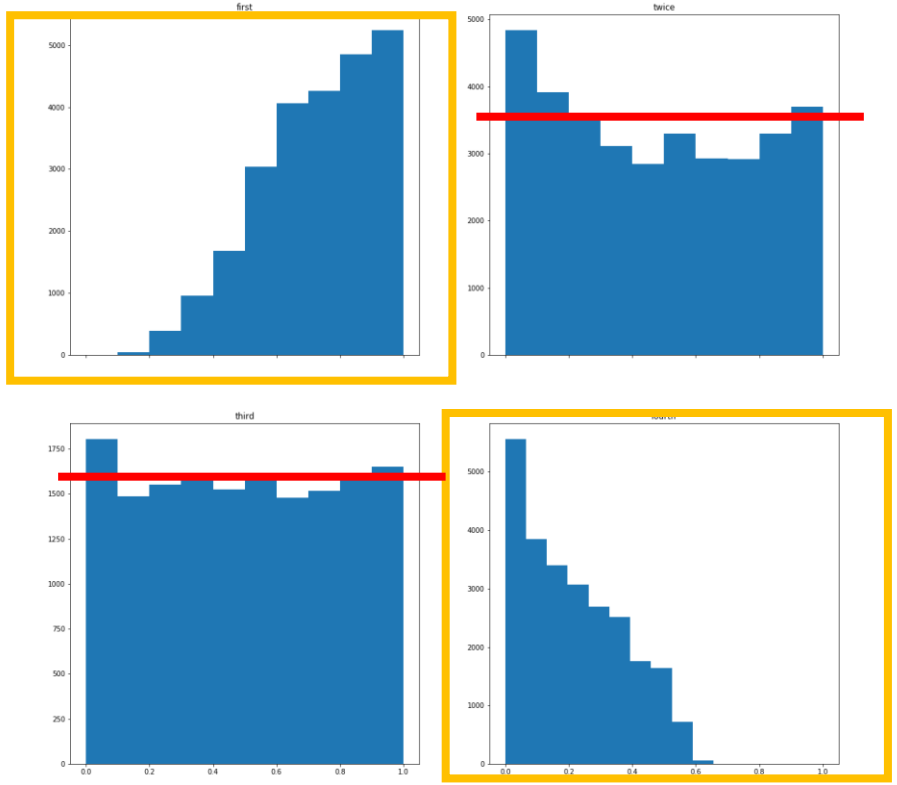


我們先使用主成分分析讓資料降維至三維（可解釋變異達0.88%），藉此來將分群結果視覺化。

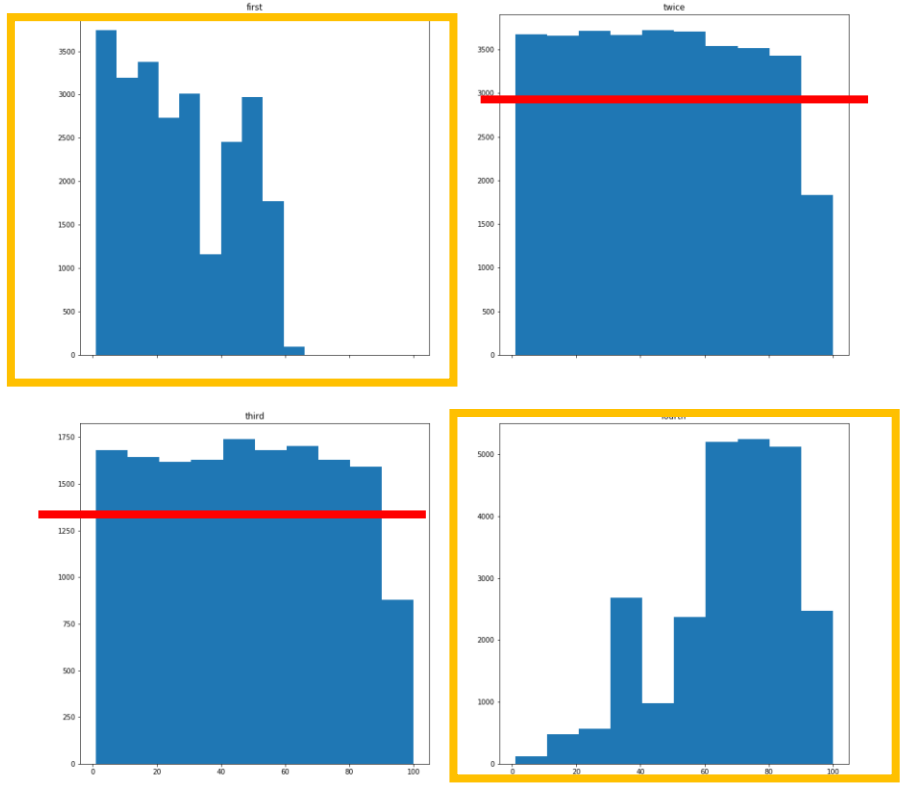
可以發現分群並無過度的重疊，大致上分群得還算恰當。

模型建立 集群分析 - 分群特徵比較

排名百分比

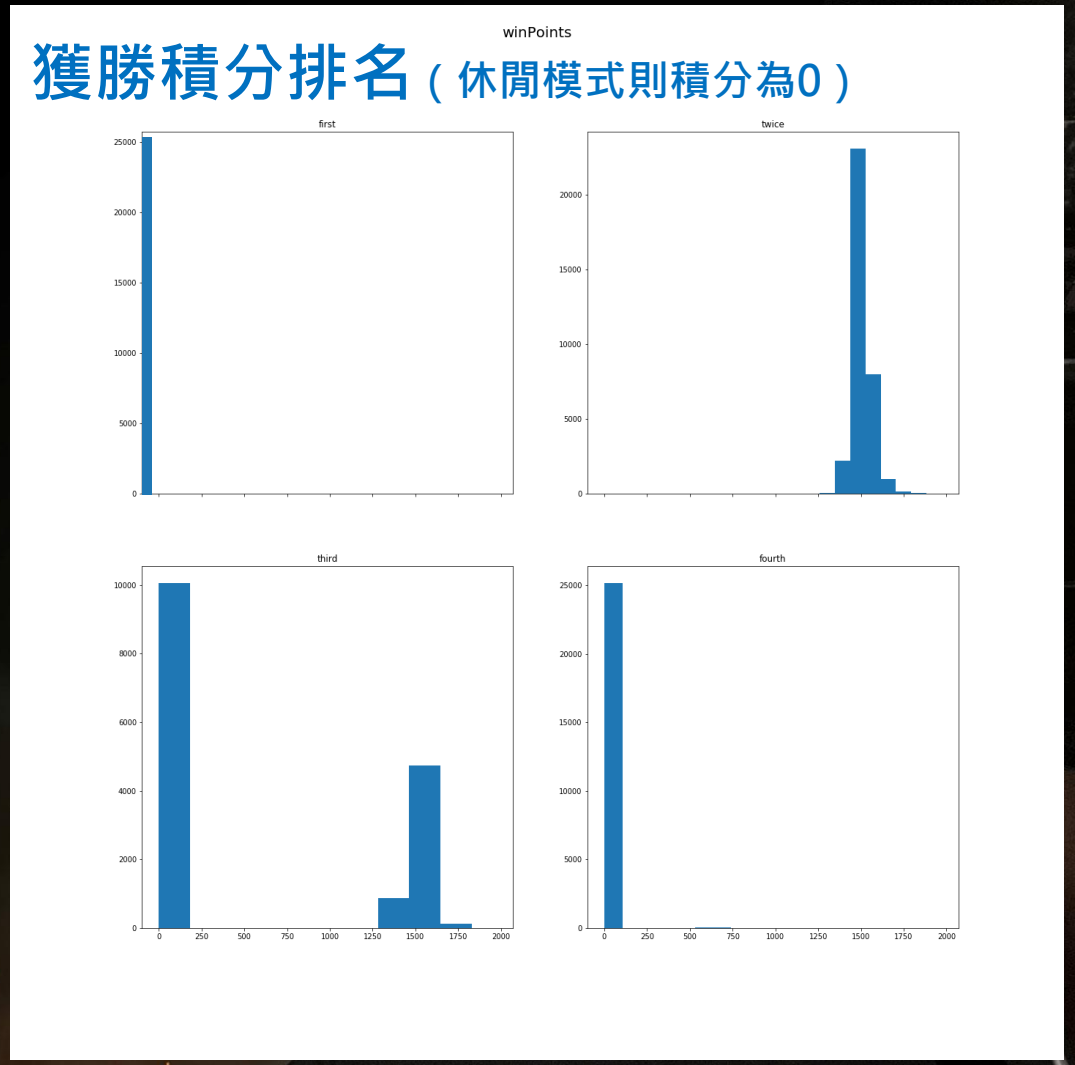


殺敵數排名百分比



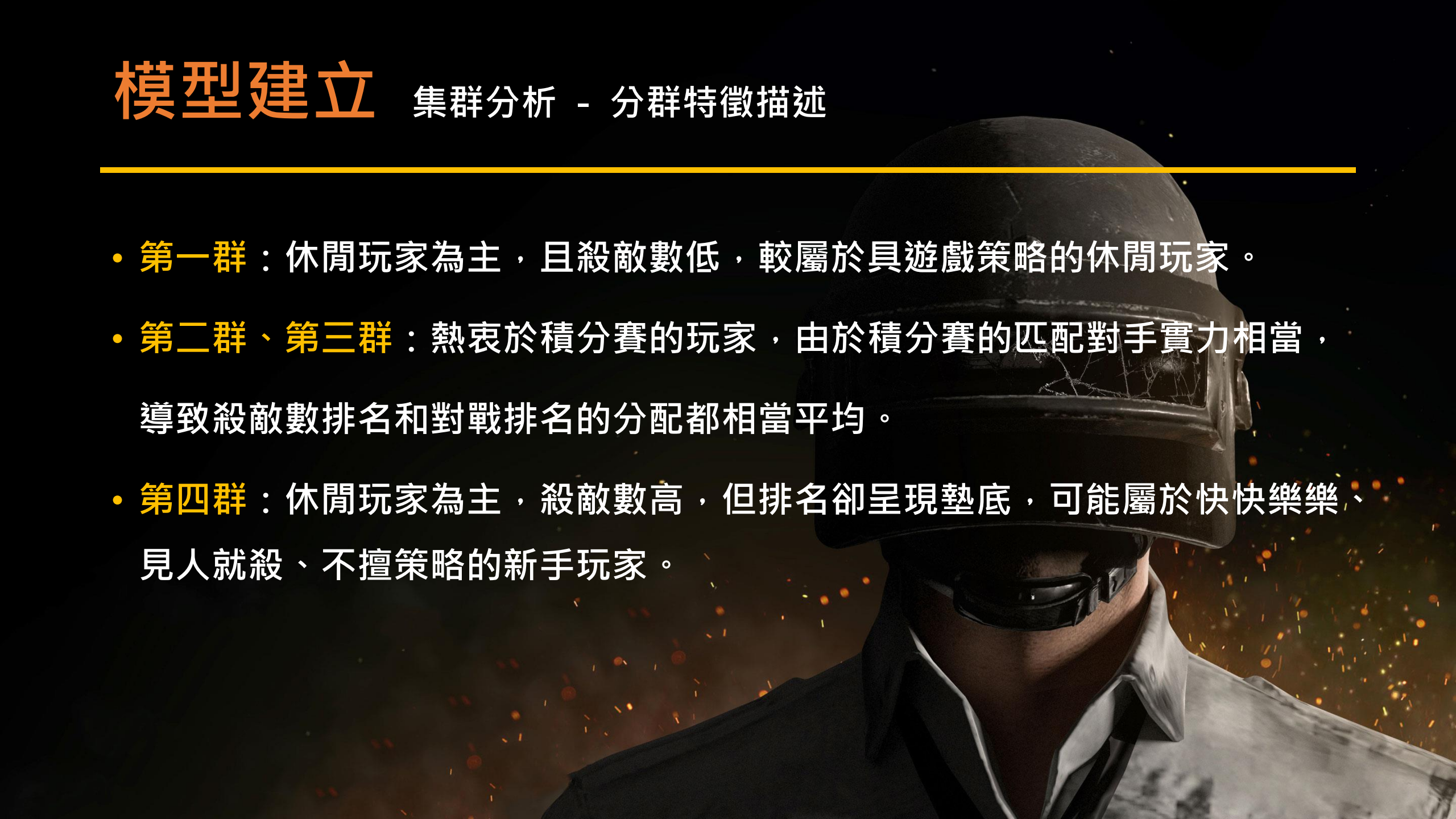
模型建立

集群分析 - 分群特徵比較



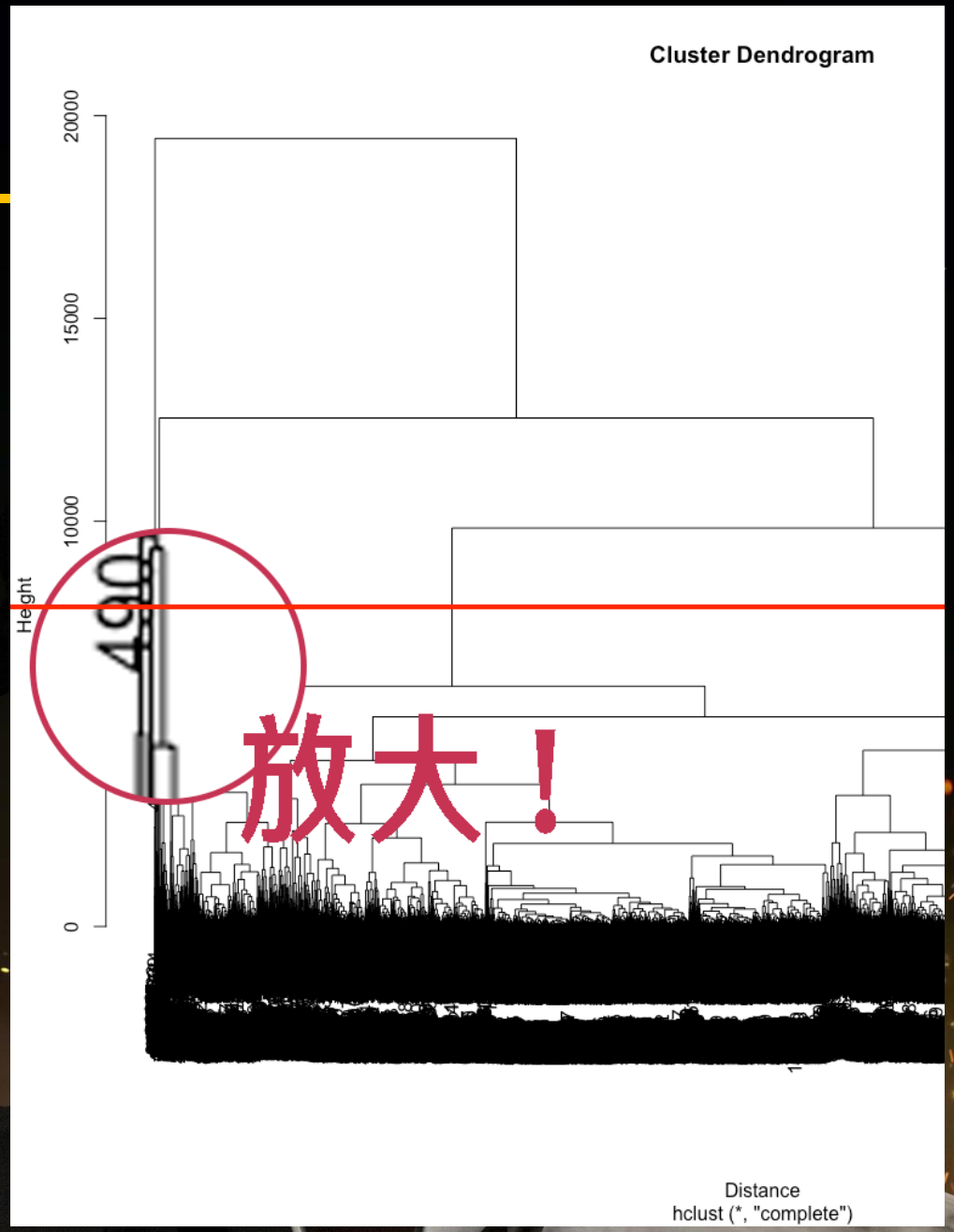
模型建立 集群分析 - 分群特徵描述

- **第一群**：休閒玩家為主，且殺敵數低，較屬於具遊戲策略的休閒玩家。
- **第二群、第三群**：熱衷於積分賽的玩家，由於積分賽的匹配對手實力相當，導致殺敵數排名和對戰排名的分配都相當平均。
- **第四群**：休閒玩家為主，殺敵數高，但排名卻呈現墊底，可能屬於快快樂樂、見人就殺、不擅策略的新手玩家。



分析困難

- 難以選擇適當分群數，來將第二群及第三群分得更開，導致第二及第三群分群不理想。



分析結論

- 針對玩家的建議：若非技術超群，我們建議一般玩家在此遊戲中若想獲得較好的排名，因以擬定策略為優先，切勿把此遊戲作為殺人遊戲，因上述分析顯示，殺敵數與對戰排名可能是背道而馳的。尤其對於新手玩家，多以休閒模式為主，而在休閒模式中殺敵排名越高的玩家，反而越難獲勝。

分析結論

- 針對**遊戲廠商**的建議：在遊戲的玩家匹配制度上，我們**建議可以嘗試將更多的第1和第4群玩家進行匹配**，讓休閒模式的環境能夠保持多樣性，同時也可能有助於第4群玩家在策略面的成長。


Thanks for listening !