110 年學年度第二學期

寶可夢捕捉率與屬性之分析預測

學號：U10811003、U10811011、U10811029

系級：數學系三

姓名：朱嘉翎、蔡君彤、顏伯諭

指導老師：李美賢 助理教授

111 年 6 月 23 日

目錄

壹、前言

寶可夢前一陣子在全世界掀起了一波熱潮，所有人都在玩，那要怎麼樣才能在眾多玩家之中脫穎而出，成為寶可夢大師呢？

知己知彼百戰百勝，如果我能對寶可夢這個遊戲更加了解的話，那麼我的寶可夢是不是就能比其他人更多、更厲害，並能在最短的時間內，了解這隻寶可夢的強項，讓我能戰無不勝攻無不克。

貳、研究目的

一、 利用寶可夢的身高、體重和能力值來預測捕捉率。

二、 利用寶可夢的屬性特色來做分類。

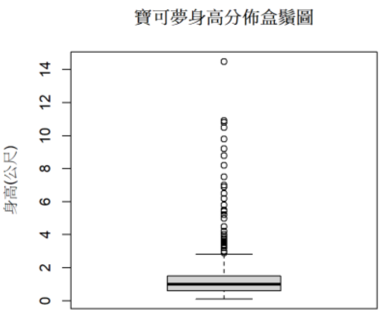
參、資料簡介

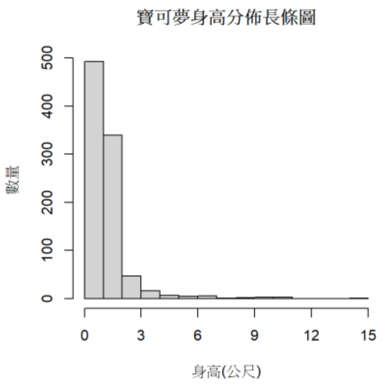
一、 變數說明

變數	類型	意義及數值範圍
身高(height_m)	連續	區間：0~100m
體重(weight_kg)	連續	區間：0~1000kg
血量(hp)	連續	區間：1~300
攻擊(attack)	連續	區間：1~200
防禦(defense)	連續	區間：1~250
特攻(sp_attack)	連續	區間：1~200
特防(sp_defense)	連續	區間：1~250
速度(speed)	連續	區間：1~200
捕捉率(catch_rate)	連續	區間：1~260
總能力值(total_point)	連續	區間：175~1125 attack+defense+sp_attack+spdefense+hp+speed
屬性(type)	類別	十八類：Water、Bug、Dark、Dragon、Electric、Fairy、Fighting、Fire、Flying、Ghost、Grass、Ground、Ice、Normal、Poison、Psychic、Rock、Steel

二、 寶可夢身高分佈分析

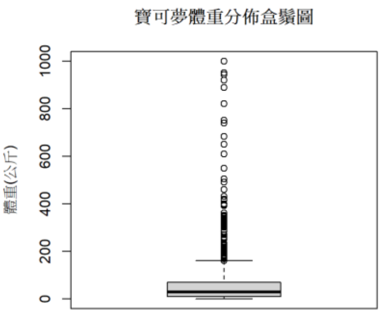
寶可夢身高主要分佈於 0~2 公尺，平均值為 1.264 公尺，且離群值非常多。

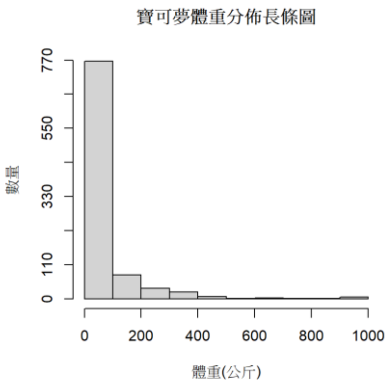




三、 寶可夢體重分佈分析

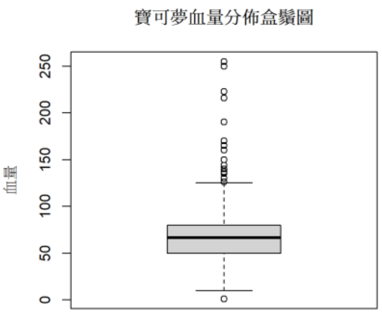
寶可夢體重主要分佈於 0~100 公斤，也有少部分分佈在 100~400 公斤，平均值為 68.774 公斤，且離群值非常多。

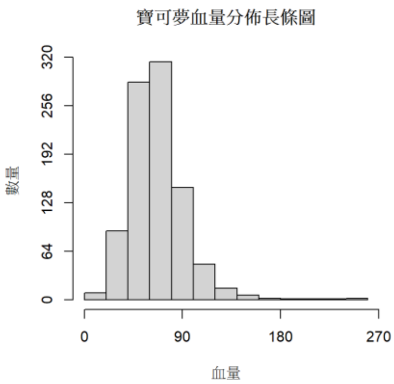




四、 寶可夢血量分佈分析

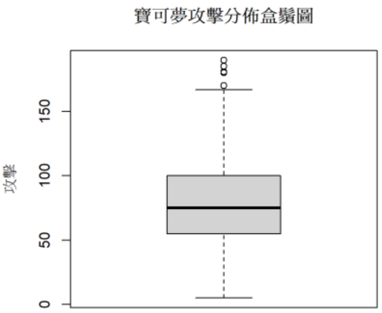
寶可夢血量主要分佈於 40~80，平均值為 69.49，且離群值非常多。

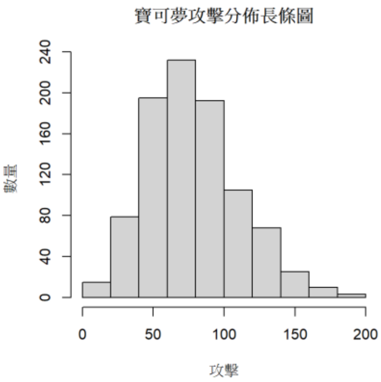




五、 寶可夢攻擊分佈分析

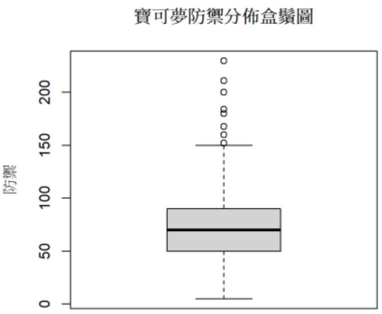
寶可夢攻擊主要分佈於 40~100，平均值為 80.02，且資料集中。

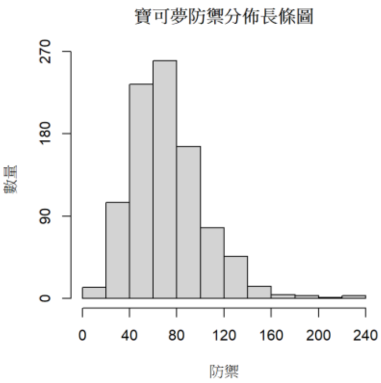




六、 寶可夢防禦分佈分析

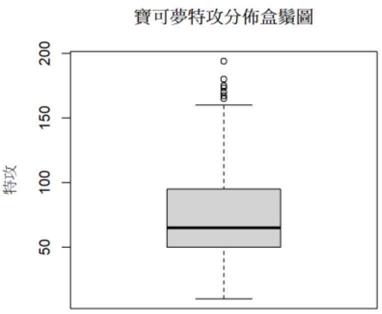
寶可夢防禦主要分佈於 40~100，平均值為 74.44，且資料集中。

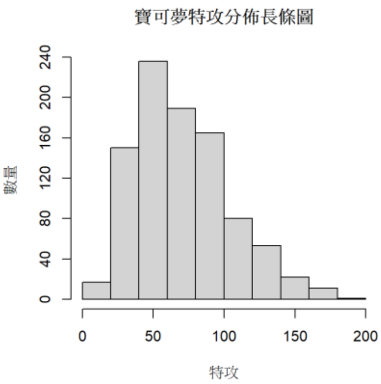




七、 寶可夢特攻分佈分析

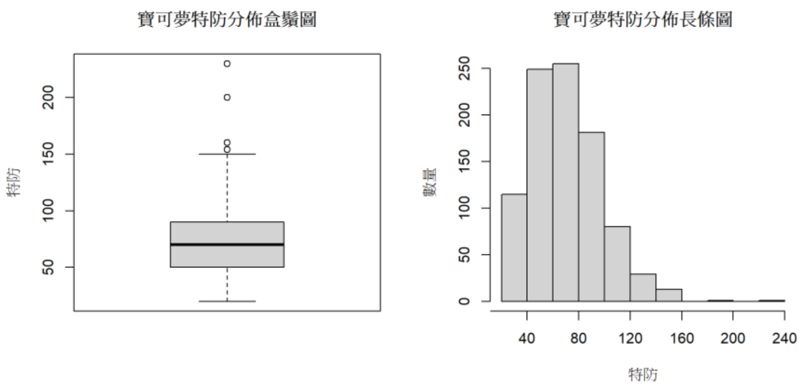
寶可夢特攻主要分佈於 40~100，平均值為 73.14，且資料集中。





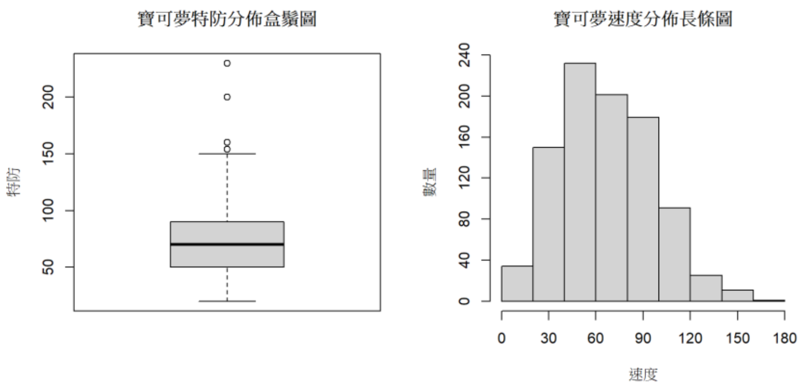
八、 寶可夢特防分佈分析

寶可夢特防主要分佈於 40~100，平均值為 72.25，且資料集中。



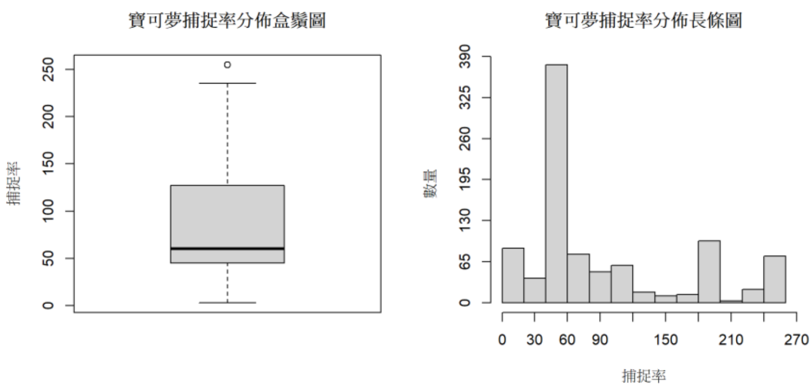
九、 寶可夢速度分佈分析

寶可夢速度主要分佈於 40~90，平均值為 68.59，且資料集中。



十、 寶可夢捕捉率分佈分析

寶可夢捕捉率主要分佈於 40~60，平均值為 92.98。



肆、 資料分析與方法

一、 利用寶可夢的身高、體重和能力值來預測捕捉率

1.不分群，用線性迴歸模型預測捕捉率

（1）在 R 中讀取寶可夢資料

（2）檢查資料是否有 NA 值

（3）將資料標準化

（4）以捕捉率為應變數，並將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數當自變數，建立線性迴歸模型

（5）使用 Cook's distance 法，偵測資料中的影響點，並移除它

（6）使用 VIF 法，檢測變數之間是否有共線性

（7）使用共變異數法，檢測變數之間是否有強相關性

（8）使用 CP 值法，選擇較合適的線性迴歸模型

（9）使用 stepwise 法，選擇較合適的線性迴歸模型

（10）隨機抽取 7 成資料當作訓練資料，剩下 3 成當測試資料

（11）使用線性迴歸模型預測測試資料的捕捉率並計算平均估計誤差

2.分群後，各群分別建立線性迴歸模型預測捕捉率

（1）在 R 中讀取寶可夢資料

（2）檢查資料是否有 NA 值

（3）將資料標準化

（4）使用 k-means 法，將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數做分群

（5）使用 K-means++法，將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數做分群

（6）使用 kernel k-means 法，將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數做分群

（7）比較 3 種方法的組內差距值

（8）對各群建立各自的線性迴歸模型。

（9）分別計算分 3 群、分 4 群、分 5 群的平均估計誤差

（10）與不分群的平均估計誤差做比較

二、 利用寶可夢的屬性特色來做分類

1.新增屬性 1 和屬性 2 來做分群

（1）在 R 中讀取寶可夢資料

（2）查看各屬性和總能力值的關係

（3）檢查資料是否有 NA 值

（4）屬性 1 和屬性 2 對總能力值的影響

（5）畫熱圖，檢視各能力的關係

（6）使用 Elbow Method，將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數考慮分幾群

（7）使用 k-means 法，將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數做分群

（8）查看身高、體重、血量、攻擊、防禦、特攻、特防、速度、總能力值和屬性 1 分群的關係

（9）查看身高、體重、血量、攻擊、防禦、特攻、特防、速度、總能力值和屬性 2 分群的關係

（10）檢視影響第一、二主成分的變數

（11）確定保留多少維度

（12）檢視影響第一、二維度變數

（13）不同分群中的差別

（14）畫雷達圖以方便觀察各群特性

伍、研究結論與討論

一、利用寶可夢的身高、體重和能力值來預測捕捉率

1. 對於整體而言

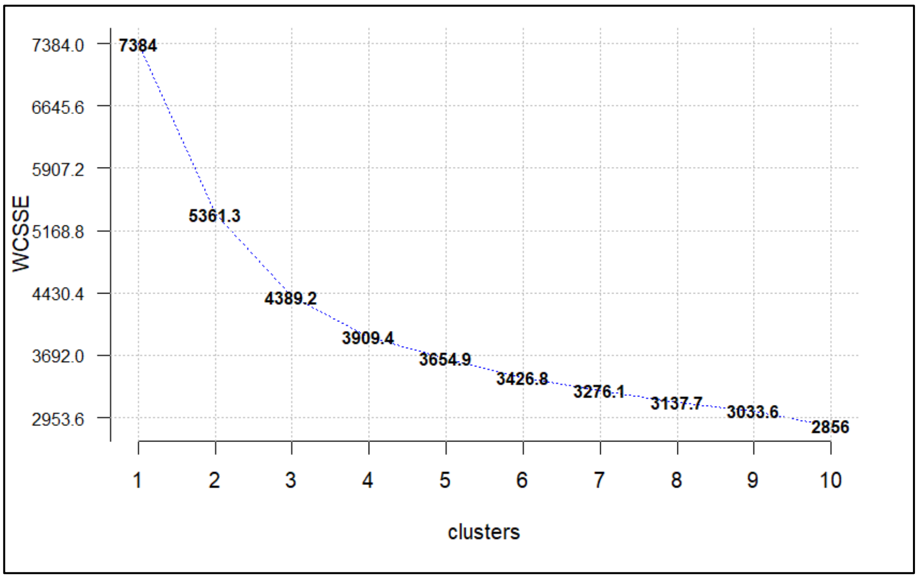
（1）身高不顯著影響捕捉率

（2）體重和捕捉率非線性相關

```
> summary(model)

Call:
lm(formula = catch_rate ~ speed + defense + hp + sp_attack +
    sp_defense + attack + weight_kg, data = traindata)

Residuals:
     Min       1Q   Median       3Q      Max
-2.07126 -0.34517 -0.06476  0.43042  2.88803

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.004965   0.026352   0.188    0.851
speed       -0.190885   0.032012  -5.963 4.10e-09 ***
defense     -0.204777   0.035898  -5.704 1.79e-08 ***
hp          -0.204830   0.033545  -6.106 1.77e-09 ***
sp_attack   -0.222504   0.034650  -6.421 2.64e-10 ***
sp_defense  -0.158351   0.036446  -4.345 1.62e-05 ***
attack      -0.175991   0.035412  -4.970 8.62e-07 ***
weight_kg    0.053616   0.036540   1.467    0.143
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

2. 將身高、體重、血量、攻擊、防禦、特攻、特防、速度等連續型變數做分群。

參考下圖的轉折點，建議分成 3、4、5 群



根據下面的表格，選擇 k-means 法最適合。

分 3 群	k-means	K-means++	kernel k-means
組內誤差	4389.2	4389.2	5335.6

分 4 群	k-means	K-means++	kernel k-means
組內誤差	3909.4	4088.9	5806.4

分 5 群	k-means	K-means++	kernel k-means
組內誤差	3654.9	3656.6	5605.3

3. 不分群與分 3、4、5 群的預測準確度比較

不分群的預測準確度最差，分 5 群的預測準確度最好。

k-means	不分群	分 3 群	分 4 群	分 5 群
平均預測誤差	39.19	35.35	35.36	34.72

4. 用 k-means 法分 5 群

(1) 第一群

a. 身高、特防、速度顯著影響捕捉率

b. 速度呈線性相關

```
> summary(model1)

Call:
lm(formula = catch_rate ~ height_m + sp_defense + speed, data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2889 -0.7332  0.2724  0.4631  1.6512 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -0.03584    0.07457  -0.481   0.6320  
height_m     0.11468    0.08183   1.401   0.1646  
sp_defense  -0.12077    0.07942  -1.521   0.1319  
speed       -0.16315    0.07124  -2.290   0.0244 *
```

（2）第二群

a. 體重、特防、攻擊不顯著影響捕捉率

b. 防禦、特攻、速度與捕捉率呈線性相關

```
> summary(model2)

Call:
lm(formula = catch_rate ~ height_m + hp + defense + sp_attack +
    speed, data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max
-0.6836 -0.3265 -0.1750  0.2136  1.4521

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05221    0.13420   0.389  0.70119
height_m     0.25547    0.13864   1.843  0.07953 .
hp          -0.26486    0.13163  -2.012  0.05721 .
defense     -0.69684    0.23831  -2.924  0.00811 **
sp_attack   -0.44970    0.16035  -2.804  0.01062 *
speed       -0.43785    0.20920  -2.093  0.04868 *
```

（3）第三群

a. 身高不顯著影響捕捉率

b. 防禦、血量、特防、速度與捕捉率呈線性相關

```
> summary(model3)

Call:
lm(formula = catch_rate ~ defense + hp + attack + sp_defense +
    speed + weight_kg + sp_attack, data = traindata)

Residuals:
     Min       1Q   Median       3Q      Max
-1.61543 -0.36081 -0.04767  0.26227  2.87551

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.13994    0.05208  -2.687  0.00817 **
defense     -0.20011    0.07268  -2.753  0.00676 **
hp          -0.17960    0.06757  -2.658  0.00886 **
attack      -0.08968    0.06438  -1.393  0.16607
sp_defense  -0.25338    0.06191  -4.093 7.50e-05 ***
speed       -0.21693    0.05241  -4.139 6.29e-05 ***
weight_kg   -0.02400    0.05965  -0.402  0.68812
sp_attack   -0.05400    0.05587  -0.967  0.33559
```

（4）第四群

a. 體重、速度不顯著影響捕捉率

b. 特攻與捕捉率呈線性相關

```
> summary(model4)

Call:
lm(formula = catch_rate ~ hp + attack + sp_attack + sp_defense +
    height_m + defense, data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max
-2.4513 -0.5833  0.2356  0.6779  1.6304

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03570    0.06108   0.584  0.55954
hp          -0.12317    0.06704  -1.837  0.06759 .
attack      -0.12979    0.07311  -1.775  0.07728 .
sp_attack   -0.20356    0.06744  -3.018  0.00285 **
sp_defense  -0.11517    0.08045  -1.432  0.15376
height_m    -0.11023    0.06791  -1.623  0.10604
defense     -0.03247    0.06816  -0.476  0.63425
```

（5）第五群

a. 攻擊、防禦、特攻、特防顯著影響捕捉率

b. 攻擊、防禦、特攻和捕捉率呈線性相關

```
> summary(model5)

Call:
lm(formula = catch_rate ~ attack + defense + sp_attack + sp_defense,
    data = traindata)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6007 -0.6576 -0.2231  0.2893  4.9251

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.06343    0.08215   0.772   0.4412
attack      -0.25080    0.10466  -2.396   0.0177 *
defense     -0.18661    0.09284  -2.010   0.0461 *
sp_attack   -0.21355    0.09390  -2.274   0.0243 *
sp_defense  -0.09586    0.09653  -0.993   0.3222
```

5. 模型係數圖與討論

	常數	height_m	weight_kg	hp	attack	defense	sp_attack	sp_defense	speed
沒分群	0.005		0.054	-0.205	-0.176	-0.205	-0.223	-0.158	-0.191
第一群	-0.036	0.115						-0.121	-0.163
第二群	0.052	0.255		-0.265		-0.697	-0.450		-0.438
第三群	-0.140		-0.024	-0.180	-0.090	-0.200	-0.054	-0.253	-0.217
第四群	0.036	-0.110		-0.123	-0.130	-0.032	-0.204	-0.115	
第五群	0.063				-0.251	-0.187	-0.214	-0.096	

（紅色字代表係數為正、灰色底代表不顯著線性相關）

（1）**高大且笨重的寶可夢看起來好有壓迫感，一定很難捕捉？**

根據不分群的線性迴歸模型，首先身高並沒有顯著影響捕捉率，代表說寶可夢的高矮不顯著影響捕捉牠的難易度，且模型中體重的係數為 0.054，此數值相較於血量、攻擊、防禦、特攻、特防、速度等係數而言是低的，代表捕捉率主要不受體重影響。綜上所述，若寶可夢高大且笨重，不代表她很難捕捉。

（2）**捕捉率是隱藏數值，有甚麼辦法可以知道寶可夢好不好抓嗎？**

以不分群的方式探討捕捉率，可以由總能力值來判斷好不好抓，這也很容易理解，能力總和越高，則越難捕捉。

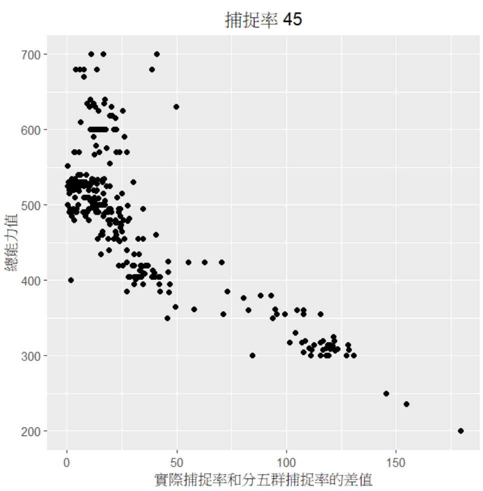
6. 同捕捉率的寶可夢，有時估計值卻天差地遠？

例如：

水水獺：實際捕捉率為 45，分 5 群的預測捕捉率為 156.65

甲賀忍蛙：實際捕捉率為 45，分 5 群的預測捕捉率為 55.97

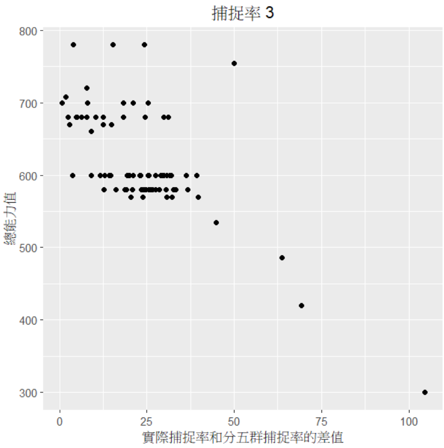
我們發現寶可夢的總能力值會影響捕捉率的估計誤差。
對於捕捉率為 45 的寶可夢，總能力值在 400~700 間的估計誤差會小於 50，反之，總能力值越遠離區間的估計誤差則會越來越大。
因為水水獺的總能力值為 308，位於區間外，而甲賀忍蛙的總能力值為 530，位在區間內，故甲賀忍蛙的預測準確度較水水獺精準。

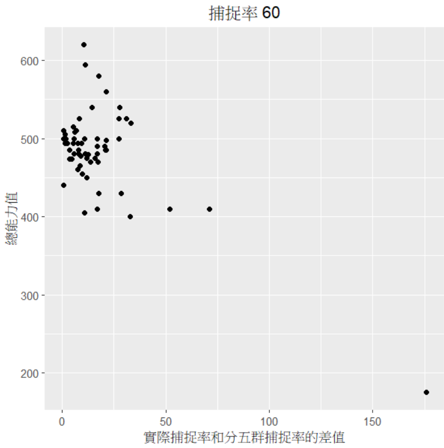


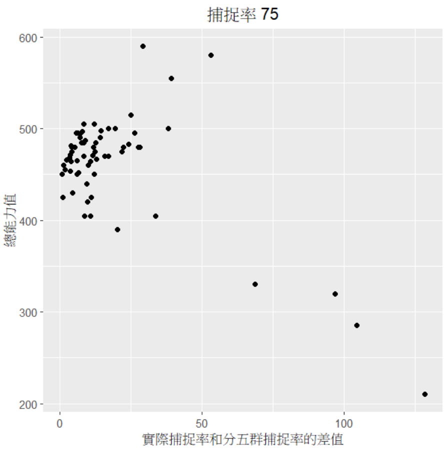
其他捕捉率也可以發現相同的規律：

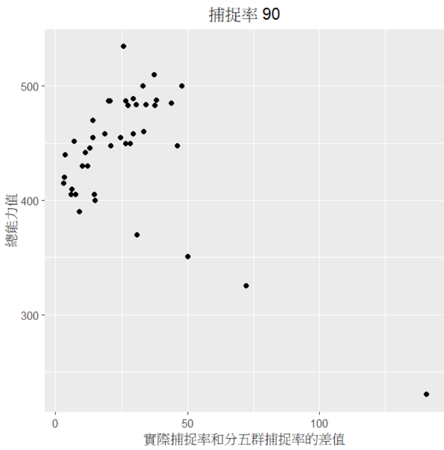
例如對於捕捉率 3 的寶可夢，估計誤差小於 50 的總能力值區間為 500~800；對於捕捉率 90 的寶可夢，估計誤差小於 50 的總能力值區間為 300~600。

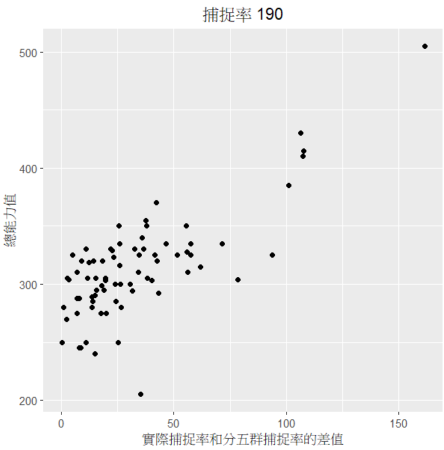
可以發現到隨著捕捉率提高，估計誤差小於 50 的總能力值區間也會逐漸降低。

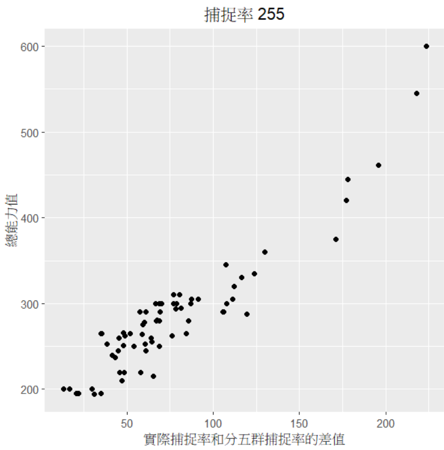








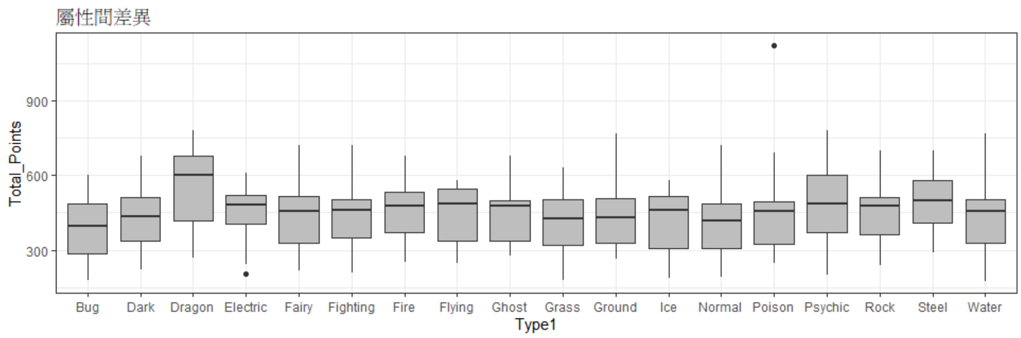




二、 利用寶可夢的屬性特色來做分類

1. 寶可夢各屬性間的差異

（1）除了龍屬性較高、蟲屬性較低外，其他屬性間並沒有明顯區別



2. 檢查資料中是否有 NA 值？

（1）屬性 2 有 486 筆 NA 值

```
> colSums(is.na(poke))
pokedex_number           name          type1          type2       height_m
    weight_kg              hp         attack
             0              0              0            486              0
            0              0              0
       defense      sp_attack     sp_defense          speed   total_points
             0              0              0              0              0
```

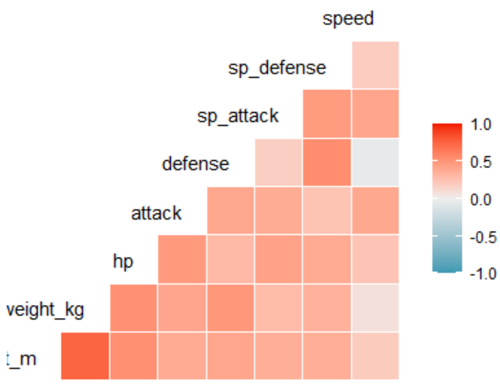
（2）將有 NA 值的資料設為另一變數

```
colSums(is.na(poclean))
  height_m   weight_kg          hp      attack     defense
         0           0           0           0           0
 sp_attack  sp_defense       speed total_points       name
         0           0           0           0           0
     type1       type2
         0           0
```

3. 各能力值兩兩之間的關係

（1）速度越快防禦越低

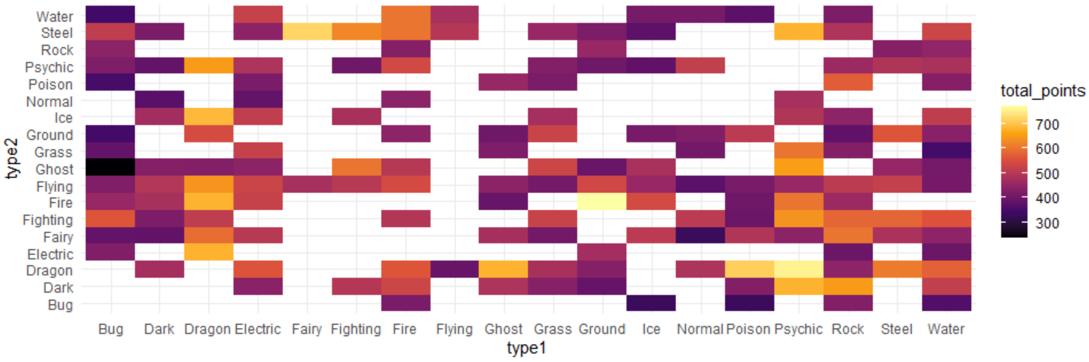
（2）防禦和特防有高度相關



4. 屬性 1 和屬性 2 間總能力值比較

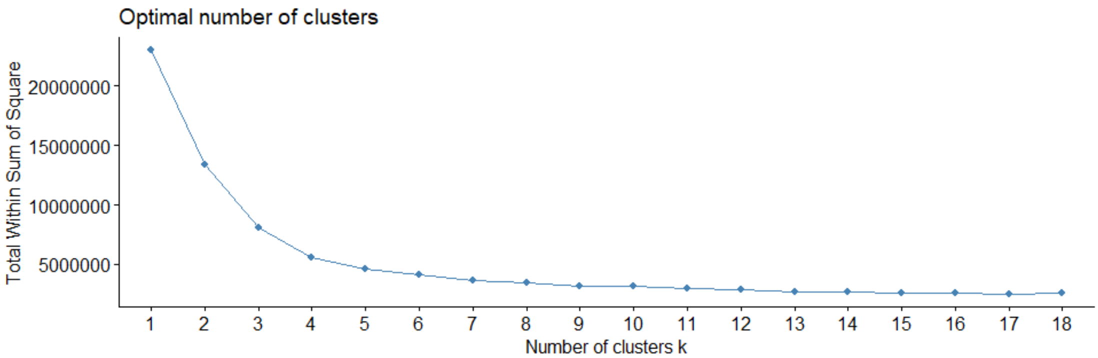
（1）屬性 1 是龍屬性的整體能力都較強的，而蟲屬性不論跟什麼搭配，其能力都較弱。

（2）精靈屬性和飛行屬性不太有屬性 2 的影響，造成能力較弱。



5. 分幾群比較好？

（1）Elbow Method 主要看轉折點，第 5 群之後斜率漸趨平緩，代表分成 6 群所能給出的資料量，和分成 5 群並沒有差很多，因此考慮分成 5 群。



6.K-means 檢視分群結果

（1）從結果上可看出第 3 群整體數值偏低，第一群整體狀況不錯

```
> kmeans <- kmeans(poclean3, centers = 5)
> kmeans
K-means clustering with 5 clusters of sizes 13, 228, 149, 95, 57

Cluster means:
   height_m weight_kg        hp    attack   defense sp_attack sp_defense    speed
1 7.8769231 828.34615 133.76923 119.46154 142.92308 106.00000  111.92308 83.84615
2 1.1723684  46.47105  73.00000  83.19298  82.48246  76.09649   79.44737 73.32456
3 0.5825503  16.03960  49.47651  54.24161  54.04027  47.43624   50.32886 50.12081
4 1.6347368  66.27474  82.93684 109.75789  91.72632 112.13684   92.52632 95.65263
5 3.3877193 284.36842  93.40351 117.21053 107.91228  99.87719   95.24561 72.75439
  total_points
1     697.9231
2     467.5439
3     305.6443
4     584.7368
5     586.4035

Clustering vector:
  [1] 3 2 4 4 4 4 4 2 3 3 2 2 3 3 2 4 3 2 3 2 2 3 2 2 2 2 3 2 3 2 3 2 2 3 2 3 2 3 2 2
 [41] 3 3 2 3 2 3 3 2 2 5 5 2 3 2 4 3 2 3 3 2 2 3 2 4 3 2 2 4 2 3 4 5 2 2 3 2 2 2 2 2
 [81] 2 4 5 5 5 3 2 3 2 2 4 4 4 4 5 4 3 2 3 2 3 2 4 3 2 3 2 3 2 4 3 2 3 3 2 2 3 2 2 2
[121] 2 2 2 5 1 2 2 4 2 2 4 2 2 3 2 2 3 5 2 3 2 4 4 3 3 2 5 5 5 5 4 4 2 4 4 2 4 4 3 2
[161] 2 3 3 3 2 3 2 3 2 3 2 3 3 2 4 3 2 2 3 2 3 3 3 2 3 2 3 2 5 3 2 2 2 3 2 4 3 5 5 3
[201] 2 2 3 2 4 2 2 3 2 2 3 2 3 2 3 2 2 3 2 4 2 4 4 3 2 5 1 4 4 4 4 1 5 5 4 5 2 4 4 3
[241] 3 2 2 3 2 3 2 2 2 2 2 3 2 2 3 2 4 2 3 2 3 5 3 2 2 3 2 4 4 4 4 3 2 3 2 3 3 2 5 2
[281] 5 5 4 2 2 5 2 4 5 2 2 2 2 2 2 2 1 5 5 1 1 4 4 2 4 3 3 2 3 2 2 4 3 2 3 3 2 3 3 5
```

（2）對應回原始資料，看每一個寶可夢被分到哪一群

```
> df_clust <- poke %>% na.omit() %>% bind_cols(cluster = as.factor(kmeans$cluster)) %>% sele
ct(cluster, 1:14)
> df_clust
# A tibble: 542 x 14
   cluster pokedex_number name      type1 type2 height_m weight_kg    hp attack defense
   <fct>            <dbl> <chr>     <fct> <fct>    <dbl>     <dbl> <dbl>  <dbl>   <dbl>
 1 3                    1 Bulbasaur Grass Pois~      0.7       6.9    45     49      49
 2 2                    2 Ivysaur   Grass Pois~      1        13      60     62      63
 3 4                    3 Venusaur  Grass Pois~      2       100      80     82      83
 4 4                    3 Mega Ven~ Grass Pois~      2.4     156.     80    100     123
 5 4                    6 Charizard Fire  Flyi~      1.7      90.5    78     84      78
 6 4                    6 Mega Cha~ Fire  Drag~      1.7     110.     78    130     111
 7 4                    6 Mega Cha~ Fire  Flyi~      1.7     100.     78    104      78
 8 2                   12 Butterfr~ Bug   Flyi~      1.1      32      60     45      50
 9 3                   13 Weedle    Bug   Pois~      0.3       3.2    40     35      30
10 3                   14 Kakuna    Bug   Pois~      0.6      10      45     25      50
# ... with 532 more rows, and 4 more variables: sp_attack <dbl>, sp_defense <dbl>,
#   speed <dbl>, total_points <dbl>
```

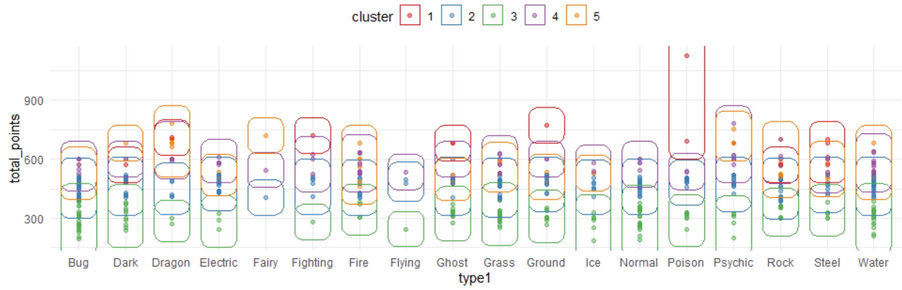
7. 屬性 1 和各項能力值分群的比較

（1）屬性 1 和總能力值

a. 第 3 群整體能力都較弱，而第 1 群較強

b. 精靈屬性和飛行屬性沒有屬性 2 加持，整體能力都較弱，沒有能被分進第 1 群中

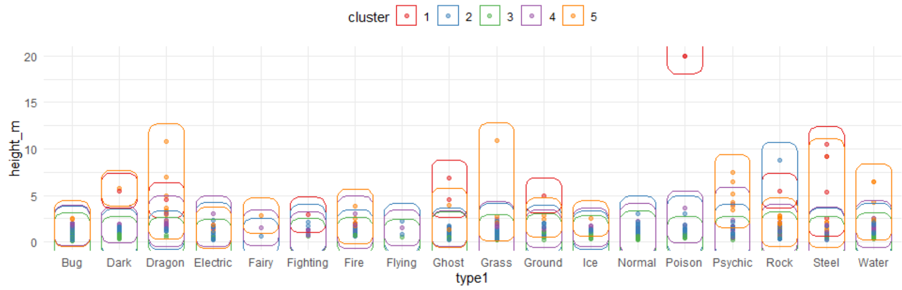
c. 毒屬性在第 1 群中有極端值存在，其他屬性基本上不論屬於哪一群，總能力值都較相近



（2）屬性 1 和身高

a. 毒屬性在身高中也是有極端值存在

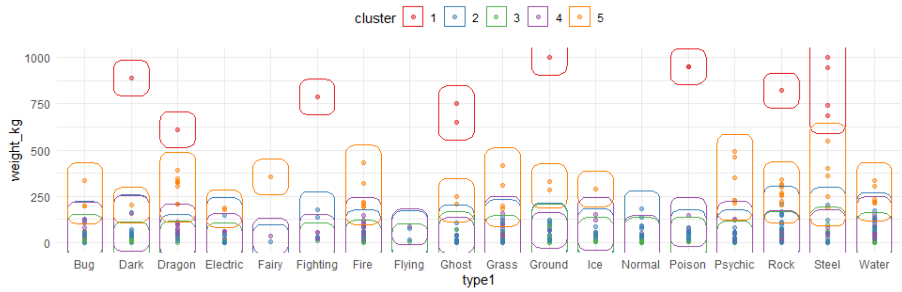
b. 除了毒屬性的極端值以外，其他各屬性各群身高差異不大



（3）屬性 1 和體重

a. 第 1 群體重都比其他群來的重很多

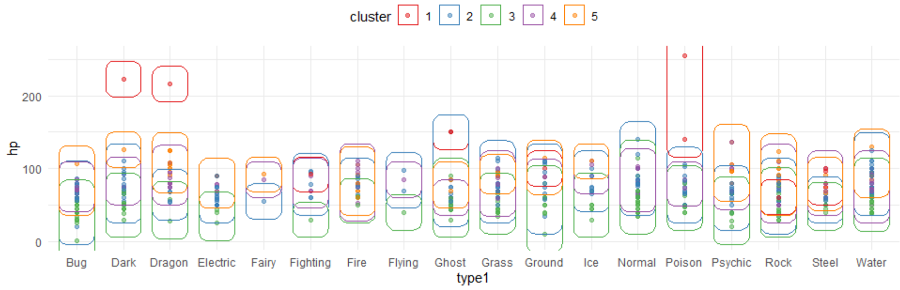
b. 第 5 群比第 2、3、4 群來的重一些



（4）屬性 1 和血量

a. 除了毒屬性的極端值和暗屬性、龍屬性以外，其他寶可夢血量差距不大

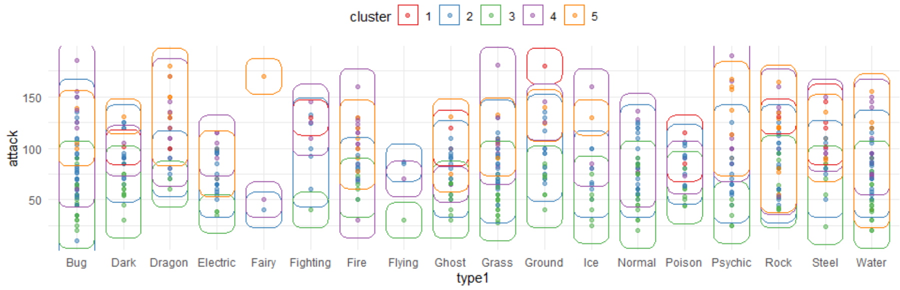
b. 第 3 群的血量比其他群來的低



（5）屬性 1 和攻擊

a. 精靈屬性在第 5 群中攻擊特別高

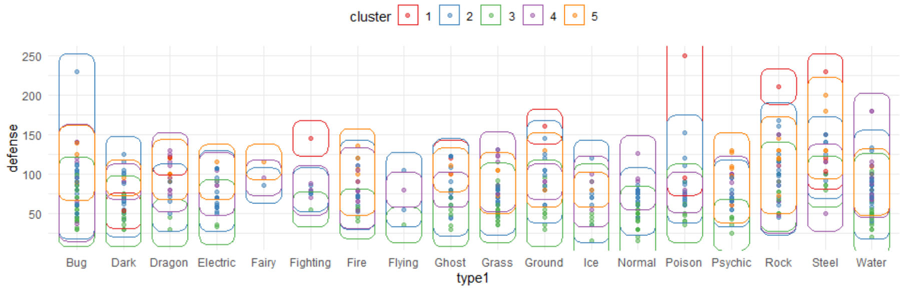
b. 第 3 群攻擊較其他群弱



（6）屬性 1 和防禦

a. 精靈屬性和飛行屬性沒有屬性 2 屬性加持，整體防禦都較弱，沒有能被分進第 1 群中

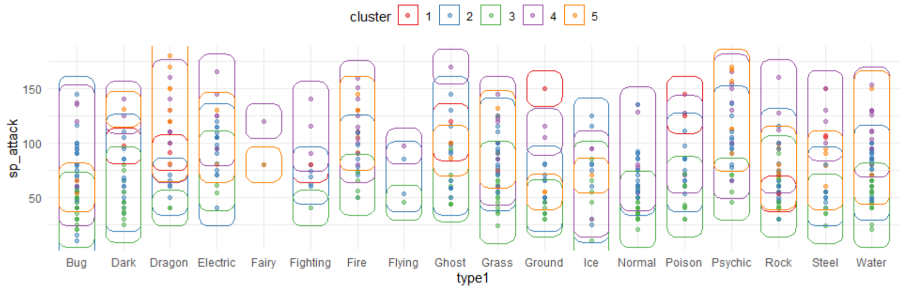
b. 各屬性防禦最強的並沒有特別被分入哪一群中，代表有些屬性防禦強但其他能力較不好



（7）屬性 1 和特攻

a. 精靈屬性和飛行屬性沒有屬性 2 屬性加持，整體防禦都較弱，沒有能被分進第 1 群中

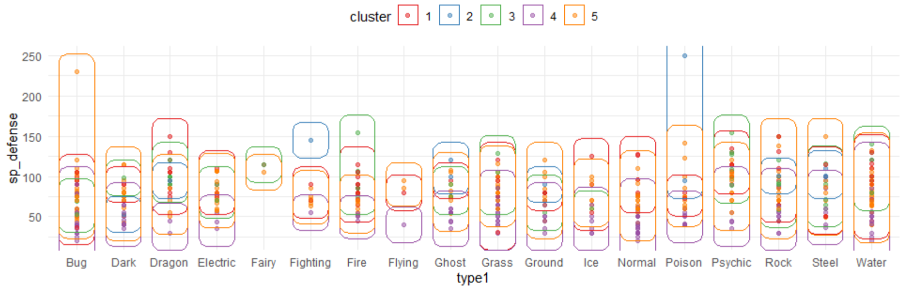
b. 一般屬性、蟲屬性特攻能力都比較弱，因此都沒被分入第一群。



（8）屬性 1 和特防

a. 蟲屬性特攻雖弱，但其中有幾隻寶可夢特防有突出的表現

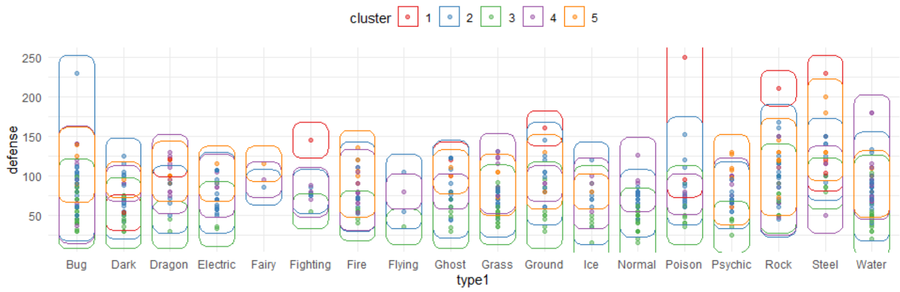
b. 毒屬性依舊存在極端值



（9）屬性 1 和速度

a. 暗屬性、龍屬性、電屬性、精靈屬性、格鬥屬性、火屬性、飛行屬性、鬼屬性、草屬性、土屬性、冰屬性和一般屬性速度較慢。

b. 毒屬性依舊存在極端值

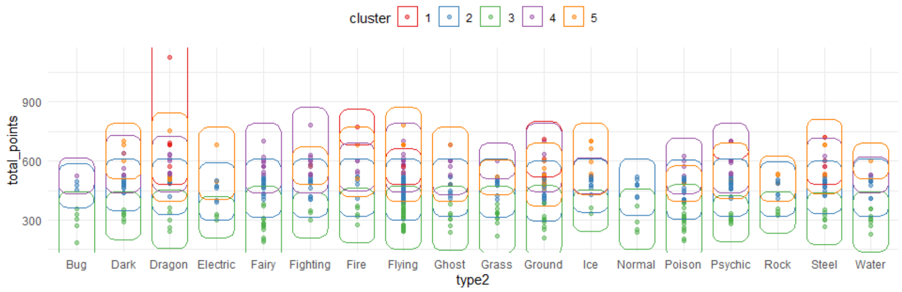


8. 屬性 2 和各項能力值分群的比較

（1） 屬性 2 和總能力值

a. 龍屬性第一群存在極端值

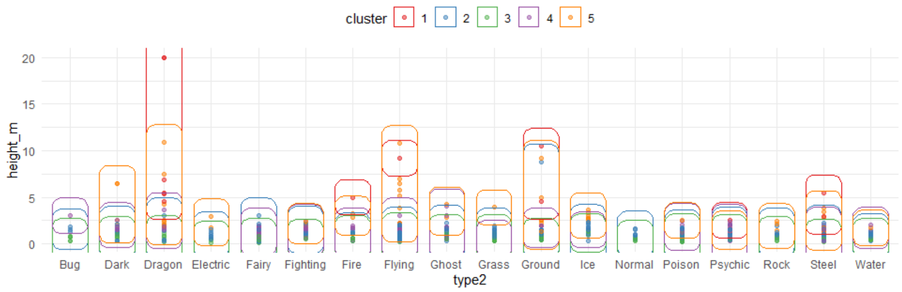
b. 第 3 群總能力值偏低



（2） 屬性 2 和身高

a. 龍屬性身高偏高

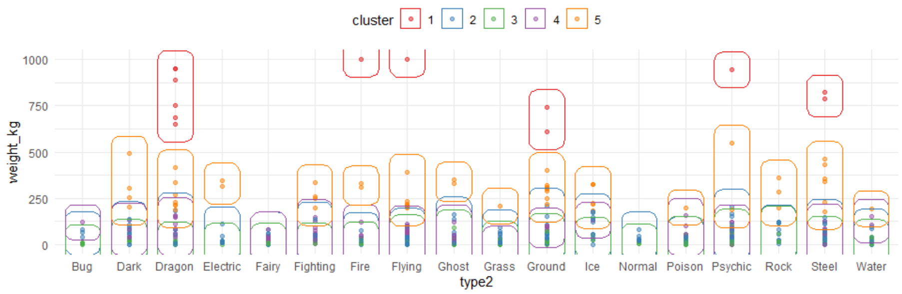
b. 蟲屬性、電屬性、精靈屬性、格鬥屬性、一般屬性、毒屬性、精神屬性、石屬性、水屬性身高集中偏低



（3） 屬性 2 和體重

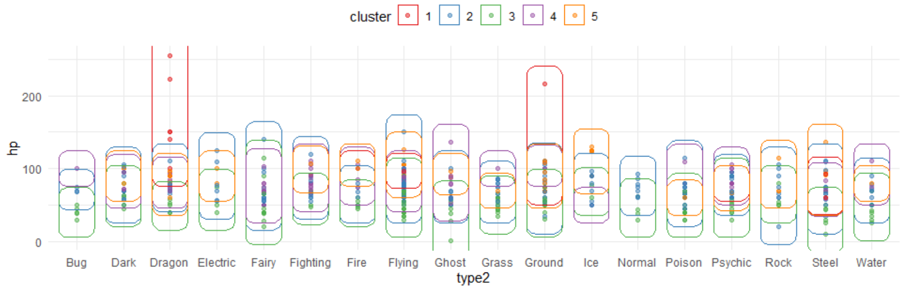
a. 第 1 群、第 5 群體重較高其他群都較輕

b. 蟲屬性、精靈屬性、一般屬性只有第 2、3、4 群



（4）屬性 2 和血量

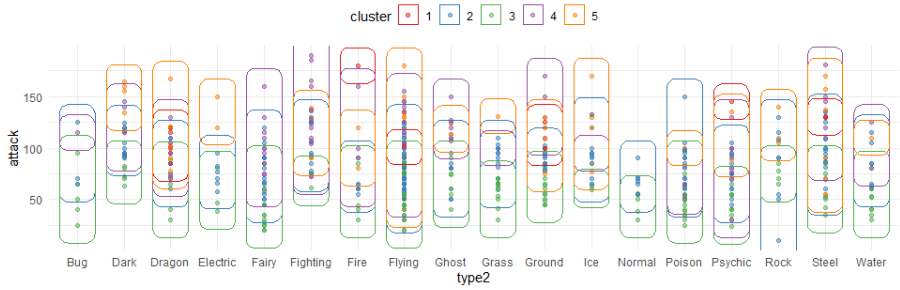
a. 龍屬性、土屬性第 1 群血量較高



（5）屬性 2 和攻擊

a. 格鬥屬性整體攻擊偏高

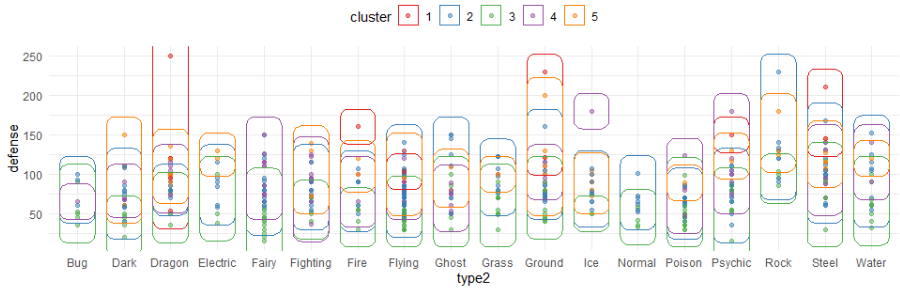
b. 一般屬性整體攻擊偏低



（6）屬性 2 和防禦

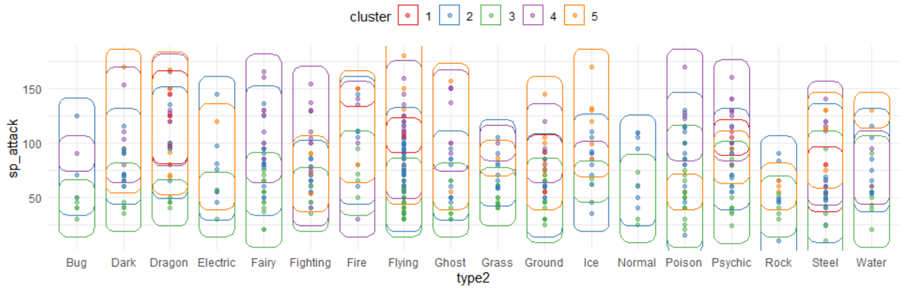
a. 龍屬性第 1 群存在極端值

b. 石屬性整體防禦偏高



（7）屬性 2 和特攻

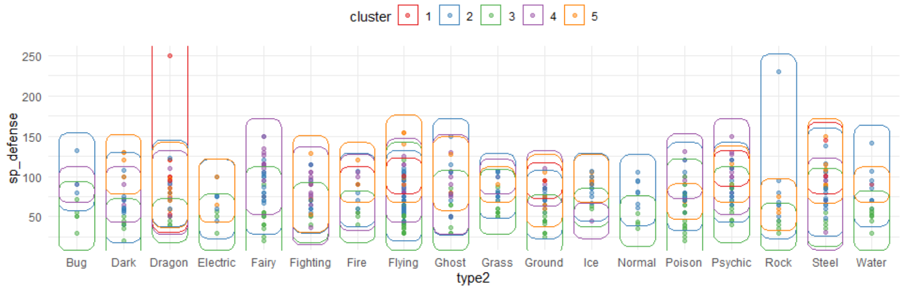
a. 各屬性的特攻皆存在極端值



（8）屬性 2 和特防

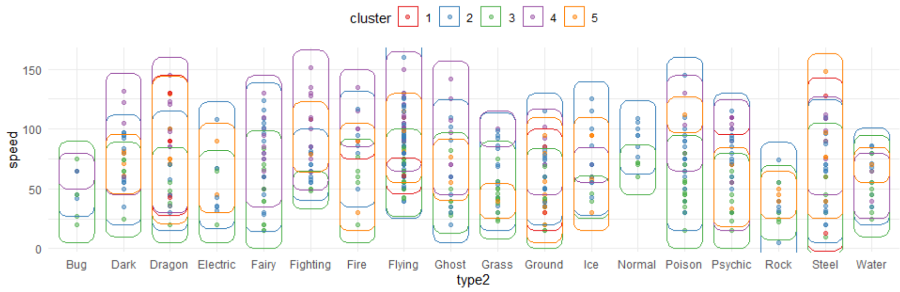
a. 龍屬性、石屬性有極端值存在

b. 其他屬性特防都滿集中的



（9）屬性 2 和速度

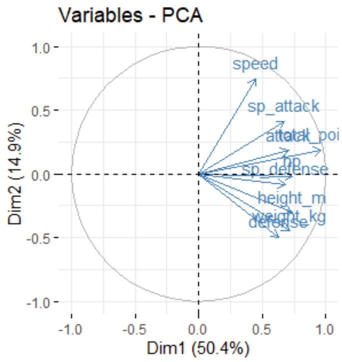
a. 蟲屬性速度極低，一般屬性速度偏快



9. 第一主成分和第二主成分

（1）速度、攻擊、血量、防禦、特攻和特防都與第一主成分相關

（2）只有速度、特攻和攻擊與第二個主成分相關，其餘變量皆為負相關或不相關

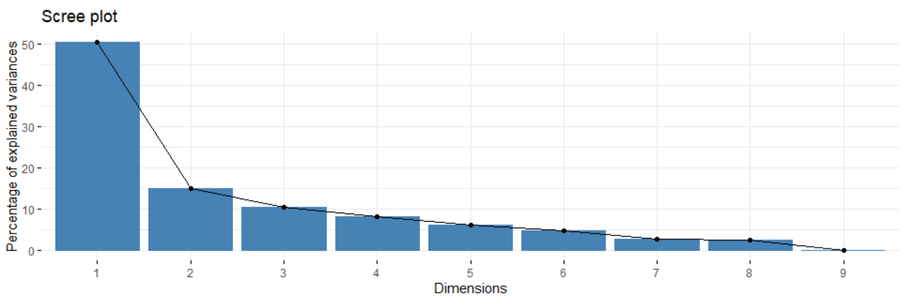


10. 該保留多少維度呢？

（1）只有第一維度和第二維度變異數大於 1，保留第一維度和第二維度即可，第三維度變異數雖然沒有大於 1，但也滿接近的，畫圖的時候可以考慮納入

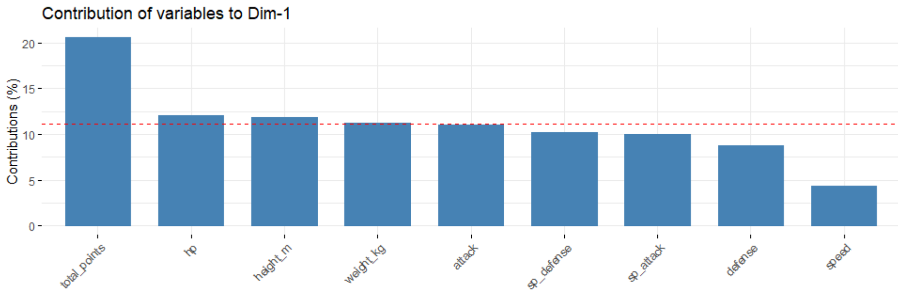
（2）第一維度提供 50.447%，第二維度提供 14.888%

```
> summary(poke_pca)

Call:
PCA(X = poclean3 %>% select(-cluster), scale.unit = T, ncp = 9)


Eigenvalues
                       Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
Variance               4.540   1.340   0.936   0.730   0.543   0.434   0.247
% of var.             50.447  14.888  10.402   8.112   6.030   4.822   2.739
Cumulative % of var.  50.447  65.335  75.736  83.849  89.879  94.700  97.440
                       Dim.8   Dim.9
Variance               0.230   0.000
% of var.              2.560   0.000
Cumulative % of var. 100.000 100.000
```

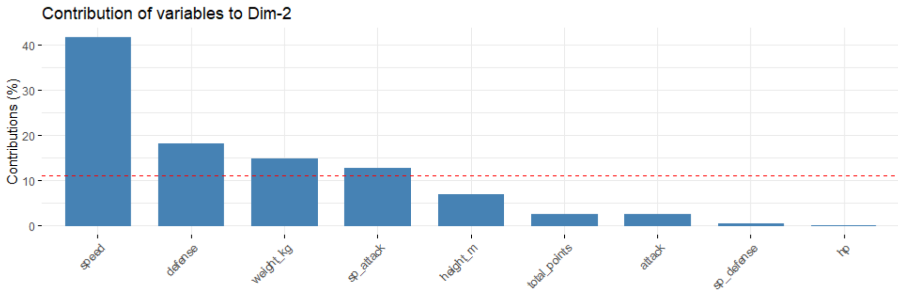


11. 第一維度和第二維度主成分貢獻度

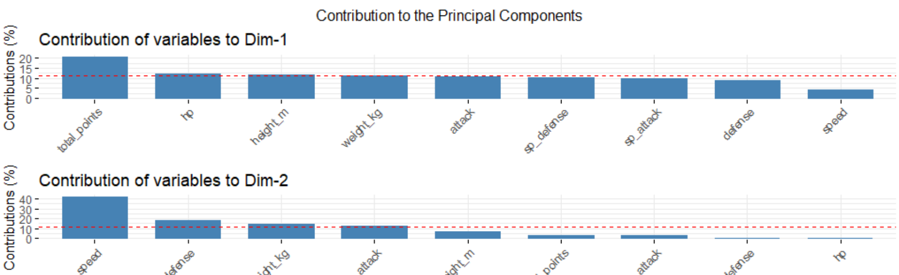
（1）第一維度是由總能力值跟血量所組成

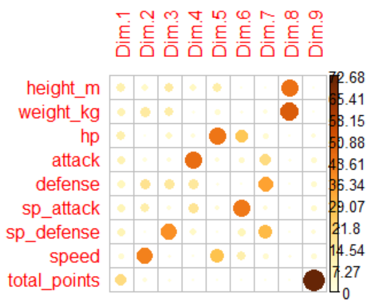


（2） 第二維度是由速度、防禦和特攻所組成



12. 第一維度和第二維度

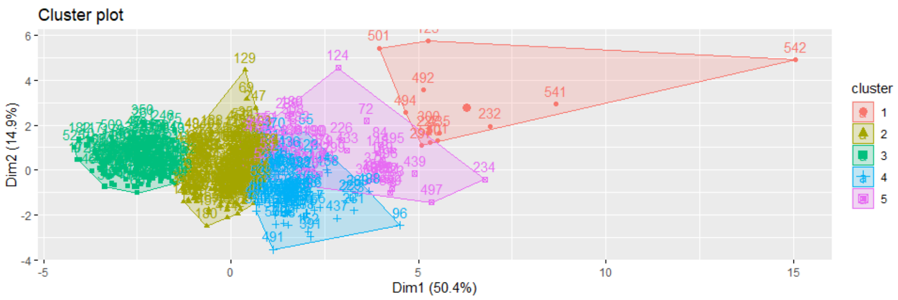




13. 成功分群

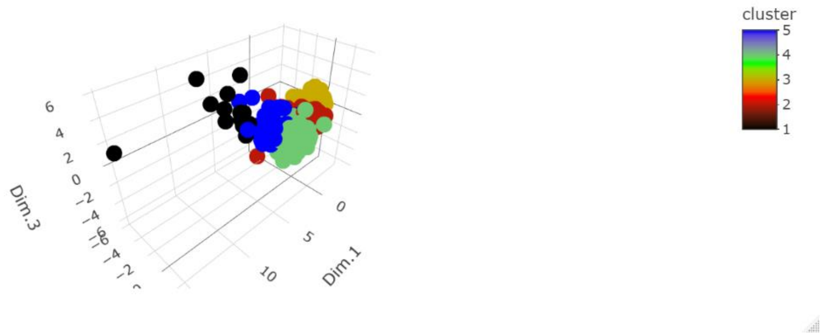
（1）平面看分群

a. 因為我們其實有 9 個維度，雖然被我們降至 2 個維度，但實際上還是有影響在裡面，所以會不容易看出，不同群的差異



（2）立體看分群

a. 增加第三維度形成一立體空間，可較清楚看出第 1 群較分散，可能存在極端值，第 2、3、4 群集中，數據較相似



14. 不同群的差異

（1）第一群整體數據較高

（2）第三群整體數據較低

（3）第四群速度快

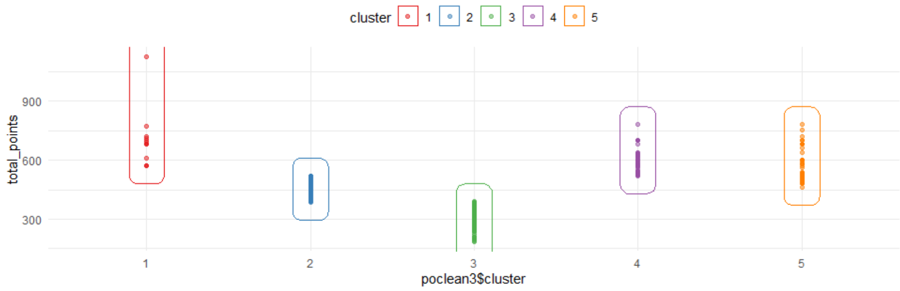
```
> cluster_all
# A tibble: 5 x 10
  cluster height_m weight_kg    hp attack defense sp_attack sp_defense speed total_points
  <fct>      <dbl>     <dbl> <dbl>  <dbl>   <dbl>     <dbl>      <dbl> <dbl>        <dbl>
1 1          7.88      828.  134.   119.    143.      106        112.   83.8         698.
2 2          1.17       46.5  73     83.2    82.5      76.1       79.4  73.3         468.
3 3          0.583      16.0  49.5   54.2    54.0      47.4       50.3  50.1         306.
4 4          1.63       66.3  82.9  110.     91.7     112.        92.5  95.7         585.
5 5          3.39      284.   93.4  117.    108.       99.9       95.2  72.8         586.
```

15. 5 群和各能力值的比較

（1）總能力值

a. 第 1 群總能力值高，第 3 群總能力值低

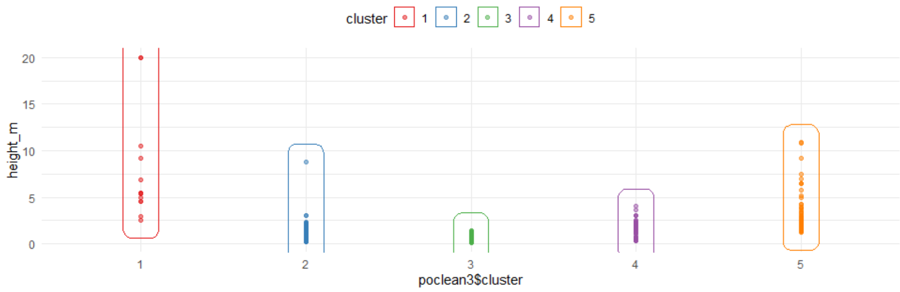
b. 第 2 群總能力值集中，代表第 2 群中沒有特別厲害或特別弱的



（2）身高

a. 第 1 群有非常高的也有非常矮的

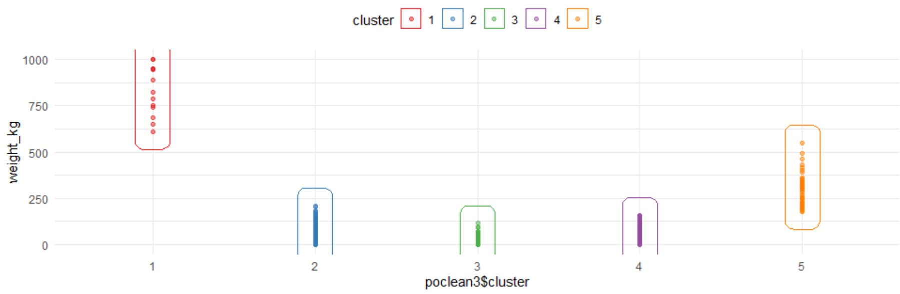
b. 第 3 群身高都偏矮



（3）體重

a. 第 1 群大部分體重都很重

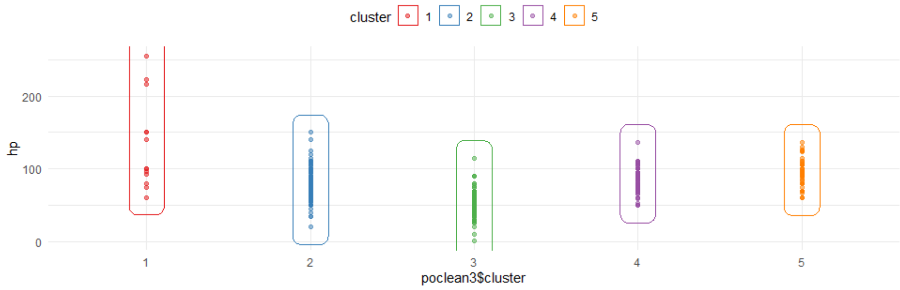
b. 第 2、3、4 群體重幾乎差不多



（4）血量

a. 第 1 群有血量偏低，也有高血量

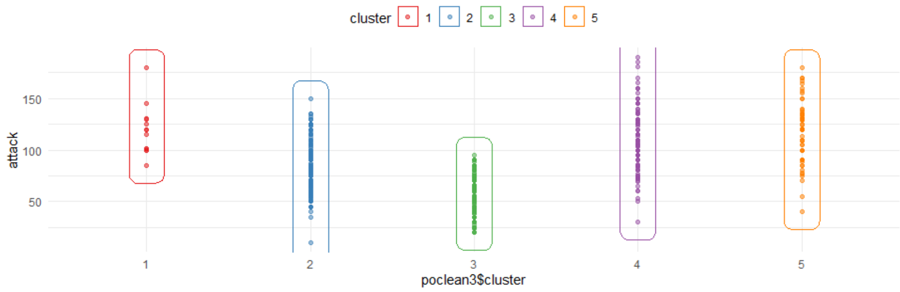
b. 第 4、5 群血量幾乎一致



（5）攻擊

a. 第 1 群攻擊集中偏高，第 3 群集中偏低

b. 第 4 群攻擊很分散，代表資料有極端值存在



（6）防禦

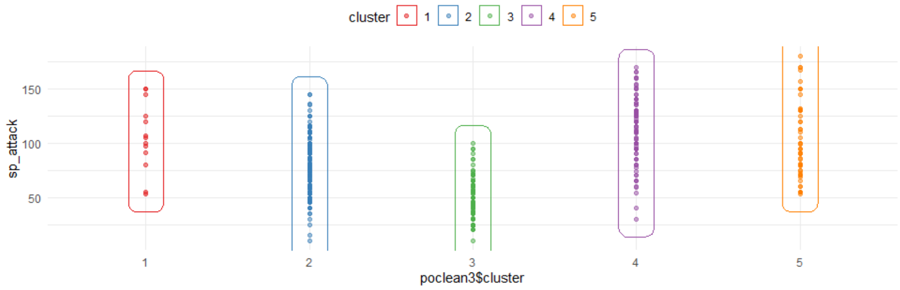
a. 第 3 群防禦偏低

b. 第 2、4、5 群防禦都有極端值存在

（7） 特攻

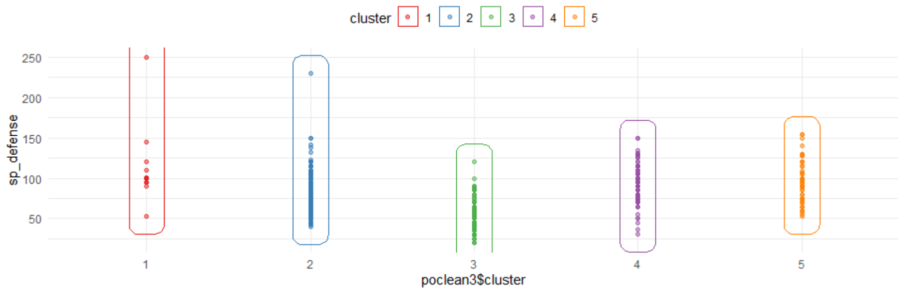
a. 第 5 群特攻集中偏高、第 3 群集中偏低

b. 不論是哪一群特攻值都較集中，代表資料內極端值較少



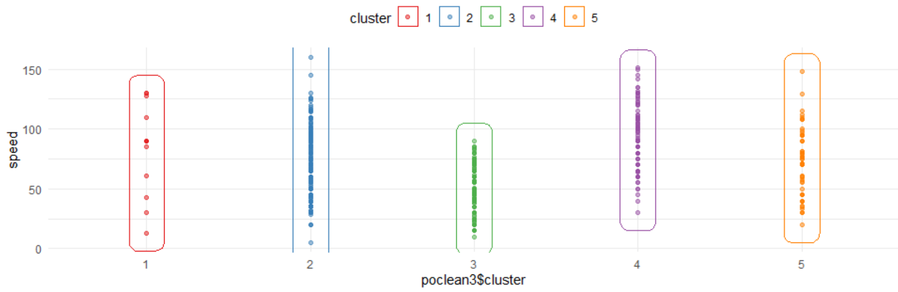
（8）特防

a. 第 1、2 群特防有極端值存在



（9）速度

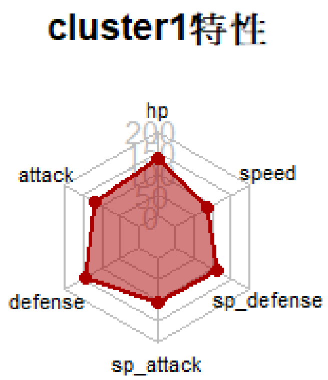
a. 第 4 群整體速度較快



16. 各群雷達圖觀察

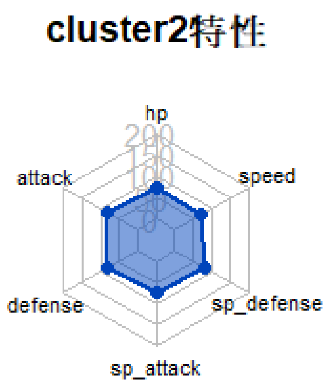
（1）第 1 群雷達圖

a. 壓倒性小隊：又大又重，除了特攻、速度以外其餘皆最高



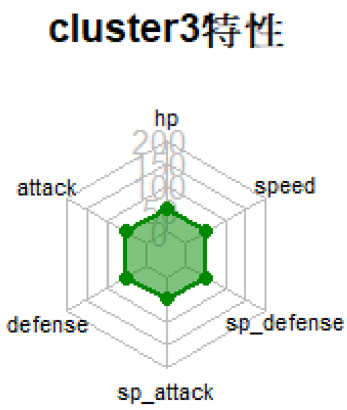
（2）第 2 群雷達圖

a. 中規中矩小隊：普通的防禦和攻擊



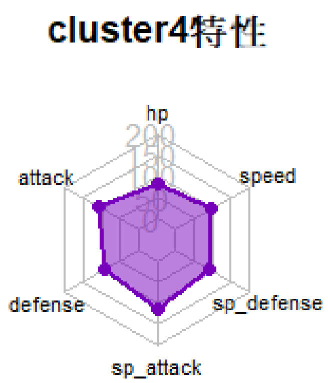
（3）第 3 群雷達圖

a. 弱小無助小隊：全部數值都偏低



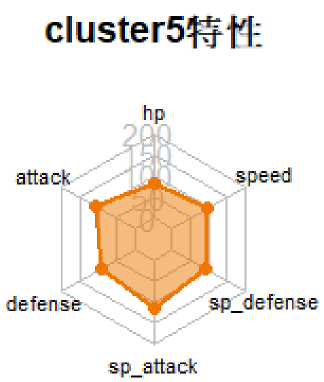
（4）第 4 群雷達圖

a. 高速小隊：特攻較強，防禦較弱



（5）第 5 群雷達圖

a. 快打小隊：高攻擊，高特攻



17. 寶可夢介紹

（1）第 1 群：壓倒性小隊

（2）第 2 群：普通小隊

（3）第 3 群：弱小無助小隊

（4）第 4 群：高速小隊

（5）第 5 群：快打小隊

陸、參考資料

一、Complete Pokemon Dataset
https://www.kaggle.com/datasets/mariotormo/complete-pokemon-dataset-updated-090420

柒、附錄

一、程式碼

```
#安裝 package
library(tidyverse)
library(lubridate)
library(cluster)
library(factoextra)
library(ggforce)
library(GGally)
library(scales)
library(cowplot)
library(FactoMineR)
library(plotly)
library(readxl)
library(corrplot)
library(gridExtra)
library(grid)
library(ggplot2)
library(lattice)
library(concaveman)
library(fmsb)
library(car)
library(reshape2)
library(leaps)
library(ClusterR)
library(kernlab)
library(dplyr)
```

#目標一 利用寶可夢的身高、體重和能力值來預測捕捉率-不分群

#1.讀取資料

```
pokemon_Rdata <- read_excel("寶可夢數據.xlsx")
pokemon_Rdata <- pokemon_Rdata[,3:11]
pokemon_Rdata <- data.frame(pokemon_Rdata)
```

#2.檢查 NA 值?

```
anyNA(pokemon_Rdata)
```

#3.資料標準化

```
pokemon_Rdata1 <- scale(pokemon_Rdata[,1:9])
pokemon_Rdata1 <- data.frame(pokemon_Rdata1)
```

#4.建立線性迴歸模型

```
attach(pokemon_Rdata)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
```

#5.偵測資料中的影響點(Cook's distance)

```
which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(pokemon_Rdata)
-9))
which(as.matrix(abs(dfbetas(model)))>1)%%924
which(as.vector(abs(dffits(model)))>1)
```

#6.是否有共線性?(VIF)

```
vif(model)
mean(vif(model))
```

#7.是否有相關性?(共變異數)

```
cor(pokemon_Rdata[,1:8])
ggplot(melt(cor(pokemon_Rdata[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue",mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1),axis.title = element_blank())
cor(pokemon_Rdata[, 1:8])
```

#8.模型選擇(CP 值法)

```
model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(pokemon_Rdata[,c(1:8)],y=pokemon_Rdata$catch_rat
e, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(4,9),ylim=c(0,200))
abline(0, b=1)
```

#9.模型選擇(stepwise 值法)

```
step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)
```

#10.隨機抽樣

```
n <- nrow(pokemon_Rdata1)
set.seed(12345)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- pokemon_Rdata1[subset,]
testdata <- pokemon_Rdata1[ - subset,]
```

#11.用線性迴歸模型預測捕捉率

```
model <- lm(catch_rate ~
height_m+hp+attack+defense+sp_attack+sp_defense+speed,traindata)
future <- predict(model,testdata)
future <- as.data.frame(future)
final <- cbind(testdata,future)
for (i in 1:9) {
  final[,i] <- (final[,i] * sd(pokemon_Rdata[,i])) +
mean(pokemon_Rdata[,i])
}
final[,10] <- (final[,10] * sd(pokemon_Rdata[,9])) +
mean(pokemon_Rdata[,9])
```

```
summary(model)
realdiff <- abs(final$catch_rate-final$future)
avg_realdiff <- sum(realdiff)/length(realdiff)
avg_realdiff

model <- lm(catch_rate ~ speed + defense + hp + sp_attack +
sp_defense + attack + weight_kg,traindata)
future <- predict(model,testdata)
future <- as.data.frame(future)
final <- cbind(testdata,future)
for (i in 1:9) {
  final[,i] <- (final[,i] * sd(pokemon_Rdata[,i])) +
mean(pokemon_Rdata[,i])
}
final[,10] <- (final[,10] * sd(pokemon_Rdata[,9])) +
mean(pokemon_Rdata[,9])
summary(model)
realdiff <- abs(final$catch_rate-final$future)
avg_realdiff <- sum(realdiff)/length(realdiff)
avg_realdiff
```

#目標一：再利用寶可夢的深、體重和能力值來預測捕捉率-分群

#1.讀取資料

```
pokemon_Rdata <- read_excel("寶可夢數據.xlsx")
pokemon_Rdata <- pokemon_Rdata[,3:11]
pokemon_Rdata <- data.frame(pokemon_Rdata)
```

#2.檢查 NA 值?

```
anyNA(pokemon_Rdata)
```

#3.標準化

```
pokemon_Rdata_scale <- scale(pokemon_Rdata[,1:8])
pokemon_Rdata_scale <- as.data.frame(pokemon_Rdata_scale)
```

#4.比較 3 種方法的組內差距值

```
##使用 k-means 法做分群
opt_km = Optimal_Clusters_KMeans(pokemon_Rdata_scale, max_clusters
= 10, initializer = 'random',criterion = "WCSSE",plot_clusters = T)
```

```
##使用 K-means++做分群
opt_km = Optimal_Clusters_KMeans(pokemon_Rdata_scale, max_clusters
= 10, initializer = 'kmeans++',criterion = "WCSSE",plot_clusters =
T)

##使用 kernel k-means 做分群
pokemon_Rdata_scale_matrix <- as.matrix(pokemon_Rdata_scale)
kkmeans <- kkmeans(pokemon_Rdata_scale_matrix, centers = 3, kernel
= "rbfdot", kpar = "automatic",alg="kkmeans")

kkmeans_final3 <- cbind(pokemon_Rdata,kkmeans@.Data)
kkmeans3_SSE <- withinss(kkmeans)
sum(kkmeans3_SSE)
kkmeans <- kkmeans(pokemon_Rdata_scale_matrix, centers = 4, kernel
= "rbfdot", kpar = "automatic",alg="kkmeans")
kkmeans_final4 <- cbind(pokemon_Rdata,kkmeans@.Data)
kkmeans4_SSE <- withinss(kkmeans)
sum(kkmeans4_SSE)
kkmeans <- kkmeans(pokemon_Rdata_scale_matrix, centers = 5, kernel
= "rbfdot", kpar = "automatic",alg="kkmeans")
kkmeans_final5 <- cbind(pokemon_Rdata,kkmeans@.Data)
kkmeans5_SSE <- withinss(kkmeans)
sum(kkmeans5_SSE)
```

#8.kmeans 分 3 群，去建構預測模型

```
km3 = KMeans_rcpp(pokemon_Rdata_scale, clusters = 3, num_init = 5,
max_iters = 100, initializer = 'random')
k_means_SSE3 <- km3$WCSS_per_cluster
sum(k_means_SSE3)
km3_out <- as.data.frame(km3$clusters)
k_means_final3 <- cbind(pokemon_Rdata,km3_out)
kmeans1 <- subset(k_means_final3,k_means_final3[10] == 1 )
kmeans2 <- subset(k_means_final3,k_means_final3[10] == 2 )
kmeans3 <- subset(k_means_final3,k_means_final3[10] == 3 )
```

#9.kmeans1 模型(分 3 群)

```
kmeans1_subset <- scale(kmeans1[,1:9])
kmeans1_subset <- data.frame(kmeans1_subset)
```

```
attach(kmeans1_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans1_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans1_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans1_subset <- kmeans1_subset[-14, ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans1_subset <- kmeans1_subset[-c(42,45), ]

model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans1_subset <- kmeans1_subset[-c(14,42), ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans1_subset <- kmeans1_subset[-45, ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans1_subset <- kmeans1_subset[-39, ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))

cor(kmeans1_subset[,1:8])
ggplot(melt(cor(kmeans1_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans1_subset[, 1:8])
```

```
model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans1_subset[,c(1:8)],y=kmeans1_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(3,9),ylim=c(0,20))
abline(0, b=1)

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ weight_kg + defense + speed,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1
```

```
n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~
height_m+hp+sp_attack+sp_defense+speed+weight_kg,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~
height_m+hp+sp_attack+sp_defense+speed+weight_kg+defense,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1
```

#10.kmeans2 模型(分 3 群)

```
kmeans2_subset <- scale(kmeans2[,1:9])
```

```
kmeans2_subset <- data.frame(kmeans2_subset)

attach(kmeans2_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans2_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans2_subset)
which(as.vector(abs(dffits(model)))>1)

vif(model)
mean(vif(model))
cor(kmeans2_subset[,1:8])
ggplot(melt(cor(kmeans2_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans2_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans2_subset[,c(1:8)],y=kmeans2_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
```

```
summary(model_stepwise)

n <- nrow(kmeans2_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans2_subset[subset,]
testdata <- kmeans2_subset[ - subset,]
model2 <- lm(catch_rate ~
height_m+defense+hp+attack+sp_attack+sp_defense+speed,traindata)
future <- predict(model2,testdata)
future <- as.data.frame(future)
final2 <- cbind(testdata,future)
for (i in 1:9) {
  final2[,i] <- (final2[,i] * sd(kmeans2[,i])) + mean(kmeans2[,i])
}
final2[,10] <- (final2[,10] * sd(kmeans2[,9])) + mean(kmeans2[,9])
realdiff2 <- abs(final2$catch_rate-final2$future)
avg_realdiff2 <- sum(realdiff2)/nrow(final2)
avg_realdiff2

n <- nrow(kmeans2_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans2_subset[subset,]
testdata <- kmeans2_subset[ - subset,]
model2 <- lm(catch_rate ~
height_m+hp+attack+sp_attack+sp_defense+speed,traindata)
future <- predict(model2,testdata)
future <- as.data.frame(future)
final2 <- cbind(testdata,future)
for (i in 1:9) {
  final2[,i] <- (final2[,i] * sd(kmeans2[,i])) + mean(kmeans2[,i])
}
final2[,10] <- (final2[,10] * sd(kmeans2[,9])) + mean(kmeans2[,9])
realdiff2 <- abs(final2$catch_rate-final2$future)
avg_realdiff2 <- sum(realdiff2)/nrow(final2)
avg_realdiff2
```

#11.kmeans3 模型

```
kmeans3_subset <- scale(kmeans3[,1:9])
kmeans3_subset <- data.frame(kmeans3_subset)

attach(kmeans3_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans3_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans3_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans3_subset <- kmeans3_subset[-124, ]
model=lm(data=kmeans3_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans3_subset <- kmeans3_subset[-c(316,478), ]
model=lm(data=kmeans3_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))

cor(kmeans3_subset[,1:8])
ggplot(melt(cor(kmeans3_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans3_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
```

```
out.all=regsubsets(kmeans3_subset[,c(1:8)],y=kmeans3_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_attack+sp_defense+speed+weight_kg,traindata)
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_attack+sp_defense+speed+height_m,traindata)
```

```
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_attack+sp_defense+speed,traindata)
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3
```

#12.全部預測誤差平均(分 3 群)

```
(sum(realdiff1)+sum(realdiff2)+sum(realdiff3))/(length(realdiff1)+
length(realdiff2)+length(realdiff3))

pokemon_testdata <- read_excel("寶可夢數據.xlsx")
x <- pokemon_testdata[,3:11]
distance_1 <- as.data.frame(1:nrow(x))
distance_2 <- as.data.frame(1:nrow(x))
distance_3 <- as.data.frame(1:nrow(x))
```

```
predict_1 <- as.data.frame(1:nrow(x))
predict_2 <- as.data.frame(1:nrow(x))
predict_3 <- as.data.frame(1:nrow(x))

for (j in 1:nrow(x)) {
  a=0
  b=0
  c=0
  for (i in 1:8) {
    a <- a + (x[j,i]-mean(final1[,i]))^2
    b <- b + (x[j,i]-mean(final2[,i]))^2
    c <- c + (x[j,i]-mean(final3[,i]))^2
  }
  distance_1[j,1] <- a^(1/2)
  distance_2[j,1] <- b^(1/2)
  distance_3[j,1] <- c^(1/2)
  x1=0*x
  x2=0*x
  x3=0*x
  for (i in 1:8) {
    x1[j,i] <- (x[j,i]-mean(kmeans1[,i]))/sd(kmeans1[,i])
    x2[j,i] <- (x[j,i]-mean(kmeans2[,i]))/sd(kmeans2[,i])
    x3[j,i] <- (x[j,i]-mean(kmeans3[,i]))/sd(kmeans3[,i])
  }
  A1 <- predict(model1,x1[j,])
  A2 <- predict(model2,x2[j,])
  A3 <- predict(model3,x3[j,])
  (predict_1[j,1] <- (A1 * sd(kmeans1[,9])) + mean(kmeans1[,9]) )
  (predict_2[j,1] <- (A2 * sd(kmeans2[,9])) + mean(kmeans2[,9]) )
  (predict_3[j,1] <- (A3 * sd(kmeans3[,9])) + mean(kmeans3[,9]) )
}

Final_prediction <- cbind(pokemon_testdata,distance_1,distance_2
                        ,distance_3,predict_1,predict_2,predict_3)
names(Final_prediction) <-
c("name","rare","height_m","weight_kg","hp","attack" ,"defense","sp
_attack","sp_defense","speed","catch_rate","distance_1","distance_2
","distance_3","predict_1","predict_2","predict_3")
```

```
write.csv(Final_prediction, file="分3群全部預測.csv", row.names = TRUE)
```

#13.kmeans 分 4 群，去建構預測模型

```
km4 = KMeans_rcpp(pokemon_Rdata_scale, clusters = 4, num_init = 5,
                  max_iters = 100, initializer = 'random')
k_means_SSE4 <- km4$WCSS_per_cluster
sum(k_means_SSE4)
km4_out <- as.data.frame(km4$clusters)
k_means_final4 <- cbind(pokemon_Rdata,km4_out)
kmeans1 <- subset(k_means_final4,k_means_final4[10] == 1 )
kmeans2 <- subset(k_means_final4,k_means_final4[10] == 2 )
kmeans3 <- subset(k_means_final4,k_means_final4[10] == 3 )
kmeans4 <- subset(k_means_final4,k_means_final4[10] == 4 )
```

#14.kmeans1 模型(分 4 群)

```
kmeans1_subset <- scale(kmeans1[,1:9])
kmeans1_subset <- data.frame(kmeans1_subset)

attach(kmeans1_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans1_subset)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans1_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans1_subset <- kmeans1_subset[-186, ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))
```

```
cor(kmeans1_subset[,1:8])
ggplot(melt(cor(kmeans1_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
scale_fill_gradient2(low = "red", high = "blue",  mid = "white",
midpoint = 0) +guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans1_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans1_subset[,c(1:8)],y=kmeans1_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(3,9),ylim=c(0,20))
abline(0, b=1)

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ hp + attack + defense + sp_attack +
sp_defense + speed,traindata)
future <- predict(model1,testdata)
```

```
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ hp + attack + defense + sp_attack +
sp_defense + speed + height_m,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ hp + attack + defense + sp_attack +
sp_defense + speed + weight_kg,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1

#15.kmeans2 模型(分 4 群)
kmeans2_subset <- scale(kmeans2[,1:9])
kmeans2_subset <- data.frame(kmeans2_subset)

attach(kmeans2_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans2_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans2_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans2_subset <- kmeans2_subset[-c(68,166,254), ]
model=lm(data=kmeans2_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans2_subset <- kmeans2_subset[-92, ]
model=lm(data=kmeans2_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans2_subset <- kmeans2_subset[-91, ]
model=lm(data=kmeans2_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))
```

```
cor(kmeans2_subset[,1:8])
ggplot(melt(cor(kmeans2_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans2_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans2_subset[,c(1:8)],y=kmeans2_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans2_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans2_subset[subset,]
testdata <- kmeans2_subset[ - subset,]
model2 <- lm(catch_rate ~
height_m+defense+hp+attack+sp_attack+sp_defense+speed,traindata)
future <- predict(model2,testdata)
future <- as.data.frame(future)
final2 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final2[,i] <- (final2[,i] * sd(kmeans2[,i])) + mean(kmeans2[,i])
}
final2[,10] <- (final2[,10] * sd(kmeans2[,9])) + mean(kmeans2[,9])
realdiff2 <- abs(final2$catch_rate-final2$future)
avg_realdiff2 <- sum(realdiff2)/nrow(final2)
avg_realdiff2
```

#16.kmeans3 模型(分 4 群)

```
kmeans3_subset <- scale(kmeans3[,1:9])
kmeans3_subset <- data.frame(kmeans3_subset)

attach(kmeans3_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans3_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans3_subset)
which(as.vector(abs(dffits(model)))>1)

vif(model)
mean(vif(model))

cor(kmeans3_subset[,1:8])
ggplot(melt(cor(kmeans3_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans3_subset[, 1:8])
```

```
model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans3_subset[,c(1:8)],y=kmeans3_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_attack+sp_defense+speed+height_m,traindata)
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3
```

#17.kmeans4 模型(分 4 群)

```
kmeans4_subset <- scale(kmeans4[,1:9])
kmeans4_subset <- data.frame(kmeans4_subset)

attach(kmeans4_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans4_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans4_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans4_subset <- kmeans4_subset[-c(5,42), ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans4_subset <- kmeans4_subset[-c(12,44), ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans4_subset <- kmeans4_subset[-39, ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans4_subset <- kmeans4_subset[-c(5,13), ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans4_subset <- kmeans4_subset[-c(15,34), ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans4_subset <- kmeans4_subset[-c(5,14), ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans4_subset <- kmeans4_subset[-c(9), ]
model=lm(data=kmeans4_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))
```

```
cor(kmeans4_subset[,1:8])
ggplot(melt(cor(kmeans4_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans4_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans4_subset[,c(1:8)],y=kmeans4_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans4_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans4_subset[subset,]
testdata <- kmeans4_subset[ - subset,]
model4 <- lm(catch_rate ~
hp+attack+sp_attack+sp_defense+speed+height_m+weight_kg,traindata)
future <- predict(model4,testdata)
future <- as.data.frame(future)
final4 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final4[,i] <- (final4[,i] * sd(kmeans4[,i])) + mean(kmeans4[,i])
}
final4[,10] <- (final4[,10] * sd(kmeans4[,9])) + mean(kmeans4[,9])
realdiff4 <- abs(final4$catch_rate-final4$future)
avg_realdiff4 <- sum(realdiff4)/nrow(final4)
avg_realdiff4

n <- nrow(kmeans4_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans4_subset[subset,]
testdata <- kmeans4_subset[ - subset,]
model4 <- lm(catch_rate ~ weight_kg + defense + sp_attack +
sp_defense + speed,traindata)
future <- predict(model4,testdata)
future <- as.data.frame(future)
final4 <- cbind(testdata,future)
for (i in 1:9) {
  final4[,i] <- (final4[,i] * sd(kmeans4[,i])) + mean(kmeans4[,i])
}
final4[,10] <- (final4[,10] * sd(kmeans4[,9])) + mean(kmeans4[,9])
realdiff4 <- abs(final4$catch_rate-final4$future)
avg_realdiff4 <- sum(realdiff4)/nrow(final4)
avg_realdiff4

n <- nrow(kmeans4_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans4_subset[subset,]
testdata <- kmeans4_subset[ - subset,]
model4 <- lm(catch_rate ~
defense+hp+sp_attack+sp_defense+speed+height_m+weight_kg,traindata)
future <- predict(model4,testdata)
future <- as.data.frame(future)
final4 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final4[,i] <- (final4[,i] * sd(kmeans4[,i])) + mean(kmeans4[,i])
}
final4[,10] <- (final4[,10] * sd(kmeans4[,9])) + mean(kmeans4[,9])
realdiff4 <- abs(final4$catch_rate-final4$future)
avg_realdiff4 <- sum(realdiff4)/nrow(final4)
avg_realdiff4
```

#18.全部預測誤差平均(分 4 群)

```
(sum(realdiff1)+sum(realdiff2)+sum(realdiff3)+sum(realdiff4))/(leng
th(realdiff1)+

length(realdiff2)+length(realdiff3)+length(realdiff4))

pokemon_testdata <- read_excel("寶可夢數據.xlsx")
x <- pokemon_testdata[,3:11]
distance_1 <- as.data.frame(1:nrow(x))
distance_2 <- as.data.frame(1:nrow(x))
distance_3 <- as.data.frame(1:nrow(x))
distance_4 <- as.data.frame(1:nrow(x))
predict_1 <- as.data.frame(1:nrow(x))
predict_2 <- as.data.frame(1:nrow(x))
predict_3 <- as.data.frame(1:nrow(x))
predict_4 <- as.data.frame(1:nrow(x))
```

```
for (j in 1:nrow(x)) {
  a=0
  b=0
  c=0
  d=0
  for (i in 1:8) {
    a <- a + (x[j,i]-mean(final1[,i]))^2
    b <- b + (x[j,i]-mean(final2[,i]))^2
    c <- c + (x[j,i]-mean(final3[,i]))^2
    d <- d + (x[j,i]-mean(final4[,i]))^2
  }
  distance_1[j,1] <- a^(1/2)
  distance_2[j,1] <- b^(1/2)
  distance_3[j,1] <- c^(1/2)
  distance_4[j,1] <- d^(1/2)
  x1=0*x
  x2=0*x
  x3=0*x
  x4=0*x
  for (i in 1:8) {
    x1[j,i] <- (x[j,i]-mean(kmeans1[,i]))/sd(kmeans1[,i])
    x2[j,i] <- (x[j,i]-mean(kmeans2[,i]))/sd(kmeans2[,i])
    x3[j,i] <- (x[j,i]-mean(kmeans3[,i]))/sd(kmeans3[,i])
    x4[j,i] <- (x[j,i]-mean(kmeans4[,i]))/sd(kmeans4[,i])
  }
  A1 <- predict(model1,x1[j,])
  A2 <- predict(model2,x2[j,])
  A3 <- predict(model3,x3[j,])
  A4 <- predict(model4,x4[j,])
  (predict_1[j,1] <- (A1 * sd(kmeans1[,9])) + mean(kmeans1[,9]) )
  (predict_2[j,1] <- (A2 * sd(kmeans2[,9])) + mean(kmeans2[,9]) )
  (predict_3[j,1] <- (A3 * sd(kmeans3[,9])) + mean(kmeans3[,9]) )
  (predict_4[j,1] <- (A4 * sd(kmeans4[,9])) + mean(kmeans4[,9]) )
}

Final_prediction <-
cbind(pokemon_testdata,distance_1,distance_2,distance_3,distance_4,
predict_1,predict_2,predict_3,predict_4)
```

```r
names(Final_prediction) <-
c("name","rare","height_m","weight_kg","hp","attack","defense","sp_
attack","sp_defense","speed","catch_rate","distance_1","distance_2"
,"distance_3","distance_4","predict_1","predict_2","predict_3","pre
dict_4")

write.csv(Final_prediction, file="分 4 群全部預測.csv", row.names =
TRUE)
```

#19.kmeans 分 5 群，去建構預測模型

```r
km5 = KMeans_rcpp(pokemon_Rdata_scale, clusters = 5, num_init = 5,
                  max_iters = 100, initializer = 'random')
k_means_SSE5 <- km5$WCSS_per_cluster
sum(k_means_SSE5)
km5_out <- as.data.frame(km5$clusters)
k_means_final5 <- cbind(pokemon_Rdata,km5_out)
kmeans1 <- subset(k_means_final5,k_means_final5[10] == 1 )
kmeans2 <- subset(k_means_final5,k_means_final5[10] == 2 )
kmeans3 <- subset(k_means_final5,k_means_final5[10] == 3 )
kmeans4 <- subset(k_means_final5,k_means_final5[10] == 4 )
kmeans5 <- subset(k_means_final5,k_means_final5[10] == 5 )
```

#20.kmeans1 模型(分 5 群)

```r
kmeans1_subset <- scale(kmeans1[,1:9])
kmeans1_subset <- data.frame(kmeans1_subset)

attach(kmeans1_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans1_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans1_subset)
which(as.vector(abs(dffits(model)))>1)
```

```
kmeans1_subset <- kmeans1_subset[-c(77,130), ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans1_subset <- kmeans1_subset[-50, ]
model=lm(data=kmeans1_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))

cor(kmeans1_subset[,1:8])
ggplot(melt(cor(kmeans1_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue",mid = "white",
midpoint = 0) +guides(fill=guide_legend(title="Correlation")) +
  theme_bw() +theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans1_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans1_subset[,c(1:8)],y=kmeans1_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(3,9),ylim=c(0,20))
```

```
abline(0, b=1)

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ hp + defense + sp_attack + sp_defense +
speed + height_m,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ hp + defense + sp_attack + sp_defense +
speed + height_m + attack,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1
```

```
n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ hp + defense + sp_attack + sp_defense +
speed + height_m + weight_kg,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1

n <- nrow(kmeans1_subset)
set.seed(345132)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans1_subset[subset,]
testdata <- kmeans1_subset[ - subset,]
model1 <- lm(catch_rate ~ height_m + sp_defense + speed,traindata)
future <- predict(model1,testdata)
future <- as.data.frame(future)
final1 <- cbind(testdata,future)
for (i in 1:9) {
  final1[,i] <- (final1[,i] * sd(kmeans1[,i])) + mean(kmeans1[,i])
}
final1[,10] <- (final1[,10] * sd(kmeans1[,9])) + mean(kmeans1[,9])
realdiff1 <- abs(final1$catch_rate-final1$future)
avg_realdiff1 <- sum(realdiff1)/nrow(final1)
avg_realdiff1
```

#21.kmeans2 模型(分 5 群)

```
kmeans2_subset <- scale(kmeans2[,1:9])
kmeans2_subset <- data.frame(kmeans2_subset)

attach(kmeans2_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans2_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans2_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans2_subset <- kmeans2_subset[-c(5,33,34), ]
model=lm(data=kmeans2_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))

cor(kmeans2_subset[,1:8])
ggplot(melt(cor(kmeans2_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue",   mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans2_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans2_subset[,c(1:8)],y=kmeans2_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
```

```
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans2_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans2_subset[subset,]
testdata <- kmeans2_subset[ - subset,]
model2 <- lm(catch_rate ~
height_m+hp+attack+sp_attack+speed+defense+weight_kg,traindata)
future <- predict(model2,testdata)
future <- as.data.frame(future)
final2 <- cbind(testdata,future)
for (i in 1:9) {
  final2[,i] <- (final2[,i] * sd(kmeans2[,i])) + mean(kmeans2[,i])
}
final2[,10] <- (final2[,10] * sd(kmeans2[,9])) + mean(kmeans2[,9])
realdiff2 <- abs(final2$catch_rate-final2$future)
avg_realdiff2 <- sum(realdiff2)/nrow(final2)
avg_realdiff2

n <- nrow(kmeans2_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans2_subset[subset,]
testdata <- kmeans2_subset[ - subset,]
model2 <- lm(catch_rate ~ height_m + hp + defense + sp_attack +
speed,traindata)
future <- predict(model2,testdata)
future <- as.data.frame(future)
final2 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final2[,i] <- (final2[,i] * sd(kmeans2[,i])) + mean(kmeans2[,i])
}
final2[,10] <- (final2[,10] * sd(kmeans2[,9])) + mean(kmeans2[,9])
realdiff2 <- abs(final2$catch_rate-final2$future)
avg_realdiff2 <- sum(realdiff2)/nrow(final2)
avg_realdiff2

#22.kmeans3 模型(分 5 群)
kmeans3_subset <- scale(kmeans3[,1:9])
kmeans3_subset <- data.frame(kmeans3_subset)

attach(kmeans3_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans3_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans3_subset)
which(as.vector(abs(dffits(model)))>1)

kmeans3_subset <- kmeans3_subset[-c(49,68,130), ]
model=lm(data=kmeans3_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)
kmeans3_subset <- kmeans3_subset[-66, ]
model=lm(data=kmeans3_subset,catch_rate~height_m+weight_kg+hp+attac
k+defense+sp_attack+sp_defense+speed)

vif(model)
mean(vif(model))
```

```
cor(kmeans3_subset[,1:8])
ggplot(melt(cor(kmeans3_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans3_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans3_subset[,c(1:8)],y=kmeans3_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_defense+speed+weight_kg ,traindata)
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_defense+speed+weight_kg ,traindata)
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3

n <- nrow(kmeans3_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans3_subset[subset,]
testdata <- kmeans3_subset[ - subset,]
model3 <- lm(catch_rate ~
defense+hp+attack+sp_defense+speed+weight_kg+sp_attack ,traindata)
future <- predict(model3,testdata)
future <- as.data.frame(future)
final3 <- cbind(testdata,future)
```

```
for (i in 1:9) {
  final3[,i] <- (final3[,i] * sd(kmeans3[,i])) + mean(kmeans3[,i])
}
final3[,10] <- (final3[,10] * sd(kmeans3[,9])) + mean(kmeans3[,9])
realdiff3 <- abs(final3$catch_rate-final3$future)
avg_realdiff3 <- sum(realdiff3)/nrow(final3)
avg_realdiff3

#23.kmeans4 模型(分 5 群)
kmeans4_subset <- scale(kmeans4[,1:9])
kmeans4_subset <- data.frame(kmeans4_subset)

attach(kmeans4_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans4_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans4_subset)
which(as.vector(abs(dffits(model)))>1)

vif(model)
mean(vif(model))

cor(kmeans4_subset[,1:8])
ggplot(melt(cor(kmeans4_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue", mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans4_subset[, 1:8])
```

```
model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans4_subset[,c(1:8)],y=kmeans4_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans4_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans4_subset[subset,]
testdata <- kmeans4_subset[ - subset,]
model4 <- lm(catch_rate ~
hp+attack+sp_attack+sp_defense+height_m,traindata)
future <- predict(model4,testdata)
future <- as.data.frame(future)
final4 <- cbind(testdata,future)
for (i in 1:9) {
  final4[,i] <- (final4[,i] * sd(kmeans4[,i])) + mean(kmeans4[,i])
}
final4[,10] <- (final4[,10] * sd(kmeans4[,9])) + mean(kmeans4[,9])
realdiff4 <- abs(final4$catch_rate-final4$future)
avg_realdiff4 <- sum(realdiff4)/nrow(final4)
avg_realdiff4
```

```
n <- nrow(kmeans4_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans4_subset[subset,]
testdata <- kmeans4_subset[ - subset,]
model4 <- lm(catch_rate ~
hp+attack+sp_attack+sp_defense+height_m+defense,traindata)
future <- predict(model4,testdata)
future <- as.data.frame(future)
final4 <- cbind(testdata,future)
for (i in 1:9) {
  final4[,i] <- (final4[,i] * sd(kmeans4[,i])) + mean(kmeans4[,i])
}
final4[,10] <- (final4[,10] * sd(kmeans4[,9])) + mean(kmeans4[,9])
realdiff4 <- abs(final4$catch_rate-final4$future)
avg_realdiff4 <- sum(realdiff4)/nrow(final4)
avg_realdiff4
```

#24.kmeans5 模型(分 5 群)

```
kmeans5_subset <- scale(kmeans5[,1:9])
kmeans5_subset <- data.frame(kmeans5_subset)

attach(kmeans5_subset)
model=lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+
sp_defense+speed)
summary(model)

which(as.vector(cooks.distance(model))>qf(0.5,9,nrow(kmeans5_subset
)-9))
which(as.matrix(abs(dfbetas(model)))>1)%%nrow(kmeans5_subset)
which(as.vector(abs(dffits(model)))>1)

vif(model)
mean(vif(model))
```

```
cor(kmeans5_subset[,1:8])
ggplot(melt(cor(kmeans5_subset[, 1:8])),aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient2(low = "red", high = "blue",mid = "white",
midpoint = 0) + guides(fill=guide_legend(title="Correlation")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 45, hjust =
1, vjust = 1), axis.title = element_blank())
cor(kmeans5_subset[, 1:8])

model01 <-
lm(catch_rate~height_m+weight_kg+hp+attack+defense+sp_attack+sp_def
ense+speed)
summary(model01)
out.all=regsubsets(kmeans5_subset[,c(1:8)],y=kmeans5_subset$catch_r
ate, nbest=3, method="exhaustive")
s.all=summary(out.all)
round(cbind(s.all$which, rsq=s.all$rsq, adjr2=s.all$adjr2,
rss=s.all$rss, cp=s.all$cp, bic=s.all$bic),3)
q=as.vector(rowSums(s.all$which))
plot(q, s.all$cp, xlim=c(1,9),ylim=c(0,200))
abline(0, b=1)

step(model01)
model_stepwise<-step(model01, test="F", direction = "both")
summary(model_stepwise)

n <- nrow(kmeans5_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans5_subset[subset,]
testdata <- kmeans5_subset[ - subset,]
model5 <- lm(catch_rate ~ attack+sp_attack+defense,traindata)
future <- predict(model5,testdata)
future <- as.data.frame(future)
final5 <- cbind(testdata,future)
for (i in 1:9) {
  final5[,i] <- (final5[,i] * sd(kmeans5[,i])) + mean(kmeans5[,i])
}
```

```
final5[,10] <- (final5[,10] * sd(kmeans5[,9])) + mean(kmeans5[,9])
realdiff5 <- abs(final5$catch_rate-final5$future)
avg_realdiff5 <- sum(realdiff5)/nrow(final5)
avg_realdiff5

n <- nrow(kmeans5_subset)
set.seed(74527)
subset <- sample(seq_len(n), size = round(0.7 * n))
traindata <- kmeans5_subset[subset,]
testdata <- kmeans5_subset[ - subset,]
model5 <- lm(catch_rate ~ attack + defense + sp_attack +
sp_defense,traindata)
future <- predict(model5,testdata)
future <- as.data.frame(future)
final5 <- cbind(testdata,future)
for (i in 1:9) {
  final5[,i] <- (final5[,i] * sd(kmeans5[,i])) + mean(kmeans5[,i])
}
final5[,10] <- (final5[,10] * sd(kmeans5[,9])) + mean(kmeans5[,9])
realdiff5 <- abs(final5$catch_rate-final5$future)
avg_realdiff5 <- sum(realdiff5)/nrow(final5)
avg_realdiff5
```

#25.全部預測誤差平均(分 5 群)

```
(sum(realdiff1)+sum(realdiff2)+sum(realdiff3)+sum(realdiff4)+sum(re
aldiff5))/(length(realdiff1)+

length(realdiff2)+length(realdiff3)+length(realdiff4)+length(realdi
ff5))

pokemon_testdata <- read_excel("寶可夢數據.xlsx")
x <- pokemon_testdata[,3:11]
distance_1 <- as.data.frame(1:nrow(x))
distance_2 <- as.data.frame(1:nrow(x))
distance_3 <- as.data.frame(1:nrow(x))
distance_4 <- as.data.frame(1:nrow(x))
distance_5 <- as.data.frame(1:nrow(x))
predict_1 <- as.data.frame(1:nrow(x))
```

```
predict_2 <- as.data.frame(1:nrow(x))
predict_3 <- as.data.frame(1:nrow(x))
predict_4 <- as.data.frame(1:nrow(x))
predict_5 <- as.data.frame(1:nrow(x))

for (j in 1:nrow(x)) {
  a=0
  b=0
  c=0
  d=0
  e=0
  for (i in 1:8) {
    a <- a + (x[j,i]-mean(final1[,i]))^2
    b <- b + (x[j,i]-mean(final2[,i]))^2
    c <- c + (x[j,i]-mean(final3[,i]))^2
    d <- d + (x[j,i]-mean(final4[,i]))^2
    e <- e + (x[j,i]-mean(final5[,i]))^2
  }
  distance_1[j,1] <- a^(1/2)
  distance_2[j,1] <- b^(1/2)
  distance_3[j,1] <- c^(1/2)
  distance_4[j,1] <- d^(1/2)
  distance_5[j,1] <- e^(1/2)
  x1=0*x
  x2=0*x
  x3=0*x
  x4=0*x
  x5=0*x
  for (i in 1:8) {
    x1[j,i] <- (x[j,i]-mean(kmeans1[,i]))/sd(kmeans1[,i])
    x2[j,i] <- (x[j,i]-mean(kmeans2[,i]))/sd(kmeans2[,i])
    x3[j,i] <- (x[j,i]-mean(kmeans3[,i]))/sd(kmeans3[,i])
    x4[j,i] <- (x[j,i]-mean(kmeans4[,i]))/sd(kmeans4[,i])
    x5[j,i] <- (x[j,i]-mean(kmeans5[,i]))/sd(kmeans5[,i])
  }
  A1 <- predict(model1,x1[j,])
  A2 <- predict(model2,x2[j,])
  A3 <- predict(model3,x3[j,])
```

```
  A4 <- predict(model4,x4[j,])
  A5 <- predict(model5,x5[j,])
  (predict_1[j,1] <- (A1 * sd(kmeans1[,9])) + mean(kmeans1[,9]) )
  (predict_2[j,1] <- (A2 * sd(kmeans2[,9])) + mean(kmeans2[,9]) )
  (predict_3[j,1] <- (A3 * sd(kmeans3[,9])) + mean(kmeans3[,9]) )
  (predict_4[j,1] <- (A4 * sd(kmeans4[,9])) + mean(kmeans4[,9]) )
  (predict_5[j,1] <- (A5 * sd(kmeans5[,9])) + mean(kmeans5[,9]) )
}

Final_prediction <-
cbind(pokemon_testdata,distance_1,distance_2 ,distance_3,distance_4
,distance_5,predict_1,predict_2,predict_3,predict_4,predict_5)
names(Final_prediction) <-
c("name","rare","height_m","weight_kg","hp","attack","defense","sp_
attack","sp_defense","speed" ,"catch_rate","distance_1","distance_2
","distance_3","distance_4","distance_5","predict_1","predict_2","p
redict_3","predict_4","predict_5")

write.csv(Final_prediction, file="分 5 群全部預測.csv", row.names =
TRUE)
```

#26.畫出分 5 群預測差值圖表

```
catch_rate_3 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
3")
catch_rate_45 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
45")
catch_rate_60 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
60"
catch_rate_75 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
75")
catch_rate_90 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
90")
catch_rate_120 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
120")
catch_rate_190 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
190")
catch_rate_255 <- read_excel("寶可夢捕捉率預測.xlsx", sheet = "捕捉率
255")
```

```
ggplot(catch_rate_3, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 3")+theme(plot.title =
element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕捉
率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_45, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 45")+theme(plot.title =
element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕捉
率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_60, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 60")+theme(plot.title =
element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕捉
率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_75, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 75")+theme(plot.title =
element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕捉
率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_90, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 90")+theme(plot.title =
element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕捉
率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_120, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 120")+theme(plot.title
= element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕
捉率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_190, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 190")+theme(plot.title
= element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕
捉率的差值")+scale_y_continuous("總能力值")
ggplot(catch_rate_255, aes(x=估計誤差,
y=total_point))+geom_point()+ggtitle("捕捉率 255")+theme(plot.title
= element_text(hjust = 0.5))+scale_x_continuous("實際捕捉率和分五群捕
捉率的差值")+scale_y_continuous("總能力值")
```

#目標二：利用寶可夢的屬性特色來做分類

```
options(scipen = 123)
```

#1.屬性間差異圖

```
ggplot(aes(type1,total_points), data=pokemon) +
geom_boxplot(fill="grey") +  ggtitle("屬性間差異") + xlab("Type1") +
ylab("Total_Points") +theme_bw()
```

#2.檢視 NA 值

```
colSums(is.na(poke))
```

#3.分割資料，重新檢視

```
poclean <- poke %>% select_if(~is.numeric(.)) %>% select(-
c(pokedex_number)) %>% mutate(name = pokemon$name,
type1=pokemon$type1, type2=pokemon$type2) %>% na.omit()
colSums(is.na(poclean))
```

#4.屬性 1 和屬性 2 比較

```
poclean  %>% group_by(type1, type2) %>% summarise(total_points =
mean(total_points)) %>%
  ggplot(aes(type1, type2, fill = total_points)) +
scale_fill_viridis_c(option = "B") +
  geom_tile() + theme_minimal()
```

#5.各能力兩兩比較熱圖

```
ggcorr(poclean2,hjust= 1)
```

#6.分群-Elbow Method

```
fviz_nbclust(poclean3, kmeans, method = "wss", k.max = 18)
```

#7.total_points 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,total_points,color = cluster)) + geom_point(alpha
= 0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#8.hp 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,hp,color = cluster)) + geom_point(alpha = 0.5) +
geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#9.height 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,height_m,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#10.weight 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,weight_kg,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#11.attack 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,attack,color = cluster)) + geom_point(alpha = 0.5)
+ geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#12.defense 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,defense,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#13.sp_attack 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,sp_attack,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#14.sp_defense 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,sp_defense,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#15.speed 和 type1 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type1,defense,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#16.total_points 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,total_points,color = cluster)) + geom_point(alpha
= 0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#17.height 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,height_m,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#18.weight 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,weight_kg,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#19.hp 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,hp,color = cluster)) + geom_point(alpha = 0.5) +
geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#20.attack 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,attack,color = cluster)) + geom_point(alpha = 0.5)
+ geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#21.defense 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,defense,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#22.sp_attack 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,sp_attack,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#23.sp_defense 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,sp_defense,color = cluster)) + geom_point(alpha =
0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#24.speed 和 type2 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(type2,speed,color = cluster)) + geom_point(alpha = 0.5)
+ geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#25.影響第一主成分和第二主成分

```
poke_pca <- PCA(poclean3 %>% select(-cluster) , scale.unit = T, ncp
= 9)
fviz_eig(poke_pca)
```

#26.確認維度

```
fviz_eig(poke_pca)
```

```
#27.第一維度影響變數
var <- get_pca_var(poke_pca)
a<-fviz_contrib(poke_pca, "var",axes = 1)
a

#28.第二維度影響變數
var <- get_pca_var(poke_pca)
b<-fviz_contrib(poke_pca, "var",axes = 2)
b

#29.第一、二維度影響變數
grid.arrange(a,b,top='Contribution to the Principal Components')
corrplot(var$contrib, is.corr = F)

#30.二維看分群結果
fviz_cluster(kmeans, poclean3 %>% select(-cluster))

#31.三維看分群結果
plot_ly(df_pca, x = ~Dim.1, y = ~Dim.2, z = ~Dim.3, color =
~cluster, colors = c("black","red","green","blue")) %>%
add_markers() %>% layout(scene = list(xaxis = list(title =
"Dim.1"),yaxis = list(title = "Dim.2"),zaxis = list(title =
"Dim.3")))

#32.cluster 內差別
poclean3$cluster <- as.factor(kmeans$cluster)
cluster_all<-poclean3 %>% group_by(cluster) %>%
summarise_if(is.numeric, 'mean') %>% select(c(cluster, (1:10)))
cluster_all

#33.5 群和 total_oints 的關係
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,total_points,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#34.5 群和 height 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,height_m,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#35.5 群和 weight 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,weight_kg,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#36.5 群和 hp 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,hp,color = cluster)) + geom_point(alpha
= 0.5) + geom_mark_hull() + scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#37.5 群和 attack 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,attack,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#38.5 群和 defense 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,defense,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#39.5 群和 sp_attack 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,sp_attack,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#40.5 群和 sp_defense 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,sp_defense,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#41.5 群和 speed 的關係

```
df_clust %>% mutate(cluster = cluster) %>%
ggplot(aes(poclean3$cluster,speed,color = cluster)) +
geom_point(alpha = 0.5) + geom_mark_hull() +
scale_color_brewer(palette = "Set1") +
  theme_minimal() + theme(legend.position = "top")
```

#42.分群雷達圖

```
cluster <- data.frame(
  row.names = c("cluster1", "cluster2", "cluster3", "cluster4",
"cluster5"),
  hp = c(133.76923,73,49.47651,82.93684,93.40351),
  attack = c(119.46154,83.19298,54.24161,109.75789,117.21053),
  defense = c(142.92308,82.48246,54.04027,91.72632,107.91228),
  sp_attack = c(106,76.09649,47.43624,112.13684,99.87719),
  sp_defense = c(111.92308,79.44737,50.32886,92.52632,95.24561),
  speed = c(83.84615,73.32456,50.12081,95.65263,72.75439))
cluster
max_min <- data.frame(
  hp = c(200, 0), attack = c(200, 0), defense = c(200, 0),
  sp_attack = c(200, 0), sp_defense = c(200, 0), speed = c(200, 0))
rownames(max_min) <- c("Max", "Min")
df <- rbind(max_min, cluster)
df
```

#43.第一群雷達圖

```
cluster1_data <- df[c("Max", "Min", "cluster1"), ]
create_beautiful_radarchart <- function(data, color = color,vlabels
= colnames(data), vlcex = 0.7,caxislabels = NULL, title =
NULL, ...){radarchart(data, axistype = 1,pcol = "#AA0000", pfcol =
scales::alpha("#AA0000", 0.5), plwd = 2, plty = 1,cglcol = "grey",
cglty = 1, cglwd = 0.8,axislabcol = "grey", vlcex = vlcex, vlabels
= vlabels,caxislabels = caxislabels, title = title)}
create_beautiful_radarchart(cluster1_data, caxislabels = c(0, 50,
100, 150, 200),title="cluster1 特性")
```

#44.第二群雷達圖

```
cluster2_data <- df[c("Max", "Min", "cluster2"), ]
create_beautiful_radarchart <- function(data, color = color,vlabels
= colnames(data), vlcex = 0.7,caxislabels = NULL, title =
NULL, ...){radarchart(data, axistype = 1,pcol = "#0044BB", pfcol =
scales::alpha("#0044BB", 0.5), plwd = 2, plty = 1,cglcol = "grey",
cglty = 1, cglwd = 0.8,axislabcol = "grey", vlcex = vlcex, vlabels
= vlabels,caxislabels = caxislabels, title = title)}
create_beautiful_radarchart(cluster2_data, caxislabels = c(0, 50,
100, 150, 200),title="cluster2 特性")
```

#45.第三群雷達圖

```
cluster3_data <- df[c("Max", "Min", "cluster3"), ]
create_beautiful_radarchart <- function(data, color = color,vlabels
= colnames(data), vlcex = 0.7,caxislabels = NULL, title =
NULL, ...){radarchart(data, axistype = 1,pcol = "#008800", pfcol =
scales::alpha("#008800", 0.5), plwd = 2, plty = 1,cglcol = "grey",
cglty = 1, cglwd = 0.8, axislabcol = "grey", vlcex = vlcex, vlabels
= vlabels, caxislabels = caxislabels, title = title)}
create_beautiful_radarchart(cluster3_data, caxislabels = c(0, 50,
100, 150, 200),title="cluster3 特性")
```

#46.第四群雷達圖

```
cluster4_data <- df[c("Max", "Min", "cluster4"), ]
create_beautiful_radarchart <- function(data, color = color,vlabels
= colnames(data), vlcex = 0.7,caxislabels = NULL, title =
NULL, ...){radarchart(data, axistype = 1,pcol = "#7700BB", pfcol =
scales::alpha("#7700BB", 0.5), plwd = 2, plty = 1,cglcol = "grey",
cglty = 1, cglwd = 0.8,axislabcol = "grey", vlcex = vlcex, vlabels
= vlabels,caxislabels = caxislabels, title = title)}
create_beautiful_radarchart(cluster4_data, caxislabels = c(0, 50,
100, 150, 200),title="cluster4 特性")
```

#47.第五群雷達圖

```
cluster5_data <- df[c("Max", "Min", "cluster5"), ]
create_beautiful_radarchart <- function(data, color = color,vlabels
= colnames(data), vlcex = 0.7,caxislabels = NULL, title =
NULL, ...){radarchart(data, axistype = 1,pcol = "#EE7700", pfcol =
scales::alpha("#EE7700", 0.5), plwd = 2, plty = 1,cglcol = "grey",
cglty = 1, cglwd = 0.8, axislabcol = "grey", vlcex = vlcex, vlabels
= vlabels, caxislabels = caxislabels, title = title)}
create_beautiful_radarchart(cluster4_data, caxislabels = c(0, 50,
100, 150, 200),title="cluster5 特性")
```