寶可夢捕捉率與屬性之分析預測

DESIGNER : 數學系三 朱嘉翎 蔡君彤 顏伯諭

June, 23, 2022

目　錄

研究動機

動機

我想成為寶可夢大師

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。

II

研究目的

目的

目的一

利用寶可夢的身高、體重和能力值來預測捕捉率

目的二

利用寶可夢的屬性特色來做分類

III

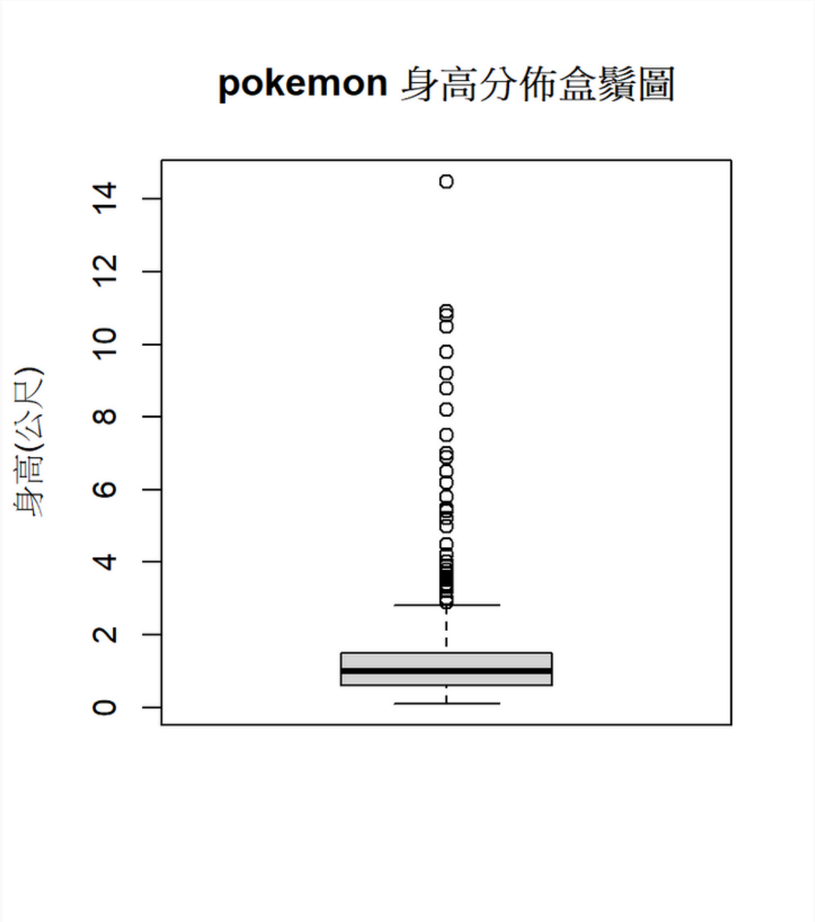
資料簡介

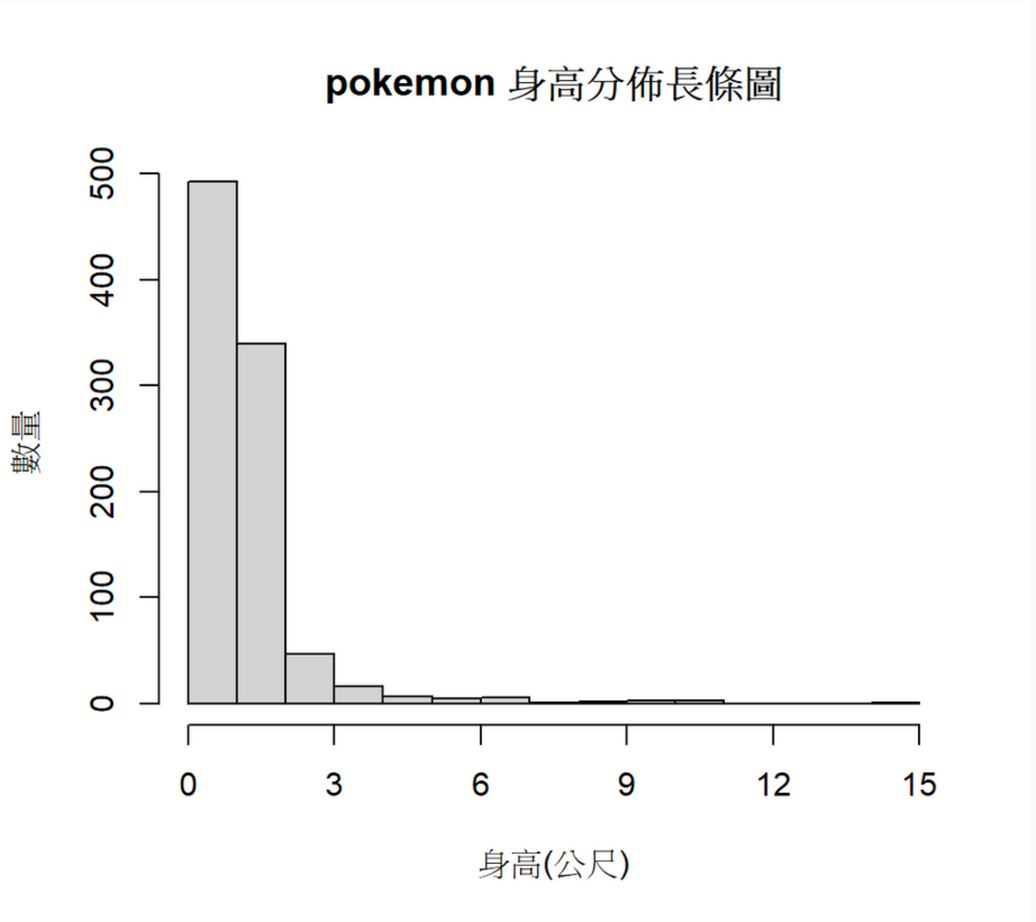
變數	類型	意義及數值範圍
身高	連續	區間：0~100 m
體重	連續	區間：0~1000 kg
血量	連續	區間：1~300
攻擊	連續	區間：1~200
防禦	連續	區間：1~250
特攻	連續	區間：1~200

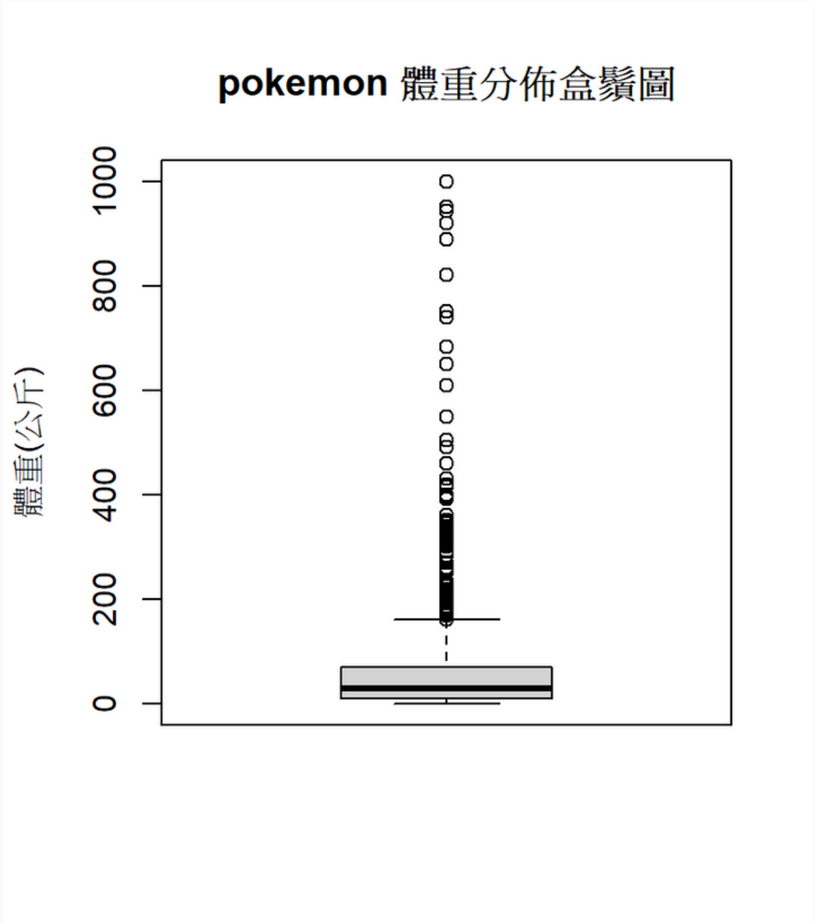
變數	類型	意義及數值範圍
特防	連續	區間：1~250
速度	連續	區間：1~200
捕捉率	連續	區間：1~260
總能力值	連續	total_points： attack+defense+sp_attack+spdefense+hp+speed 175~1125
屬性	類別	十八類：Water、Bug、Dark、Dragon、Electric、Fairy、Fighting、Fire、Flying、Ghost、Grass、Ground、Ice、Normal、Poison、Psychic、Rock、Steel

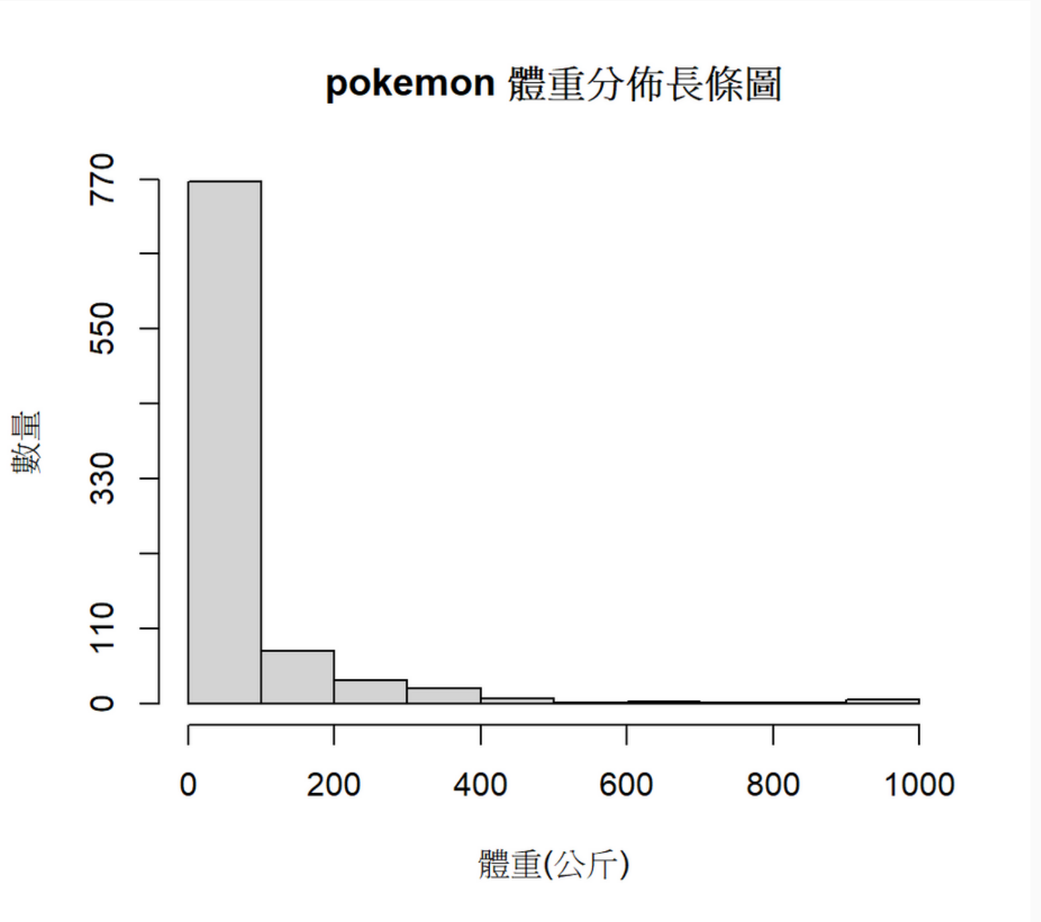
IV

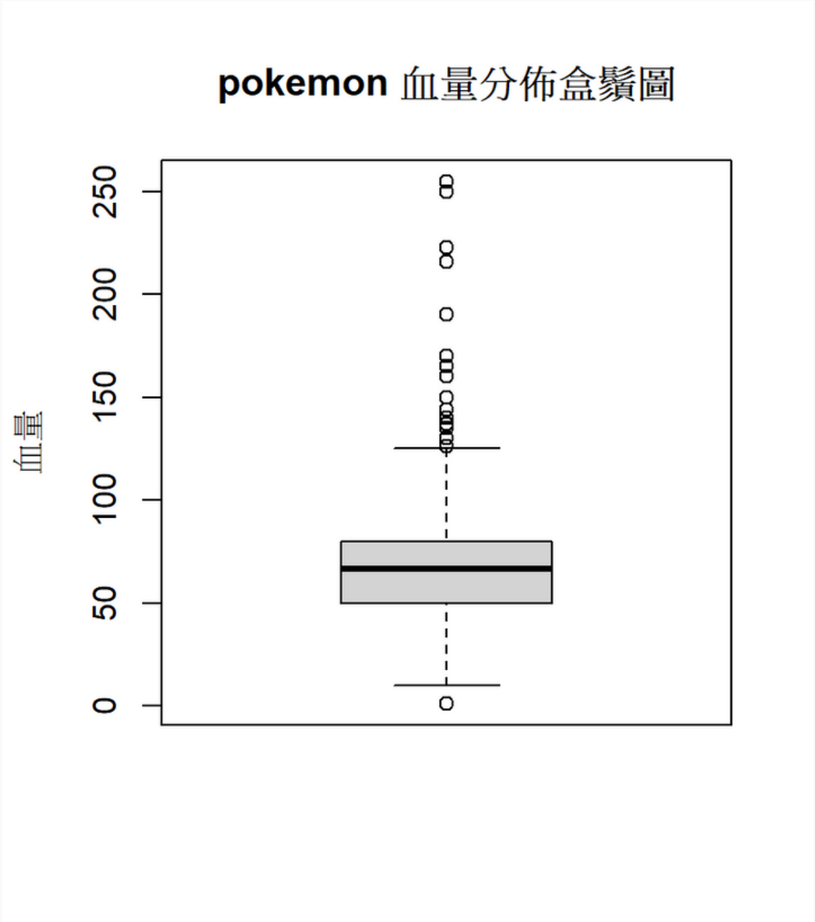
圖表分析

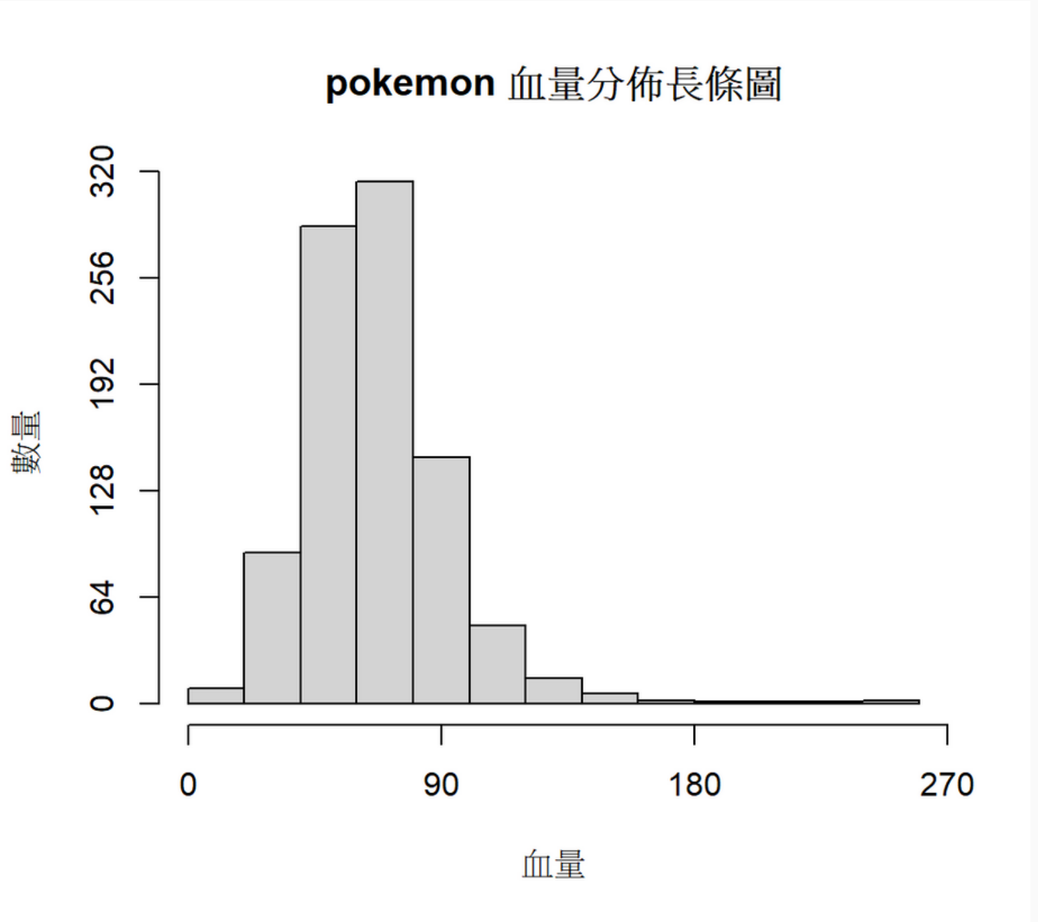

pokemon 身高分佈盒鬚圖
身高(公尺)
0
2
4
6
8
10
12
14

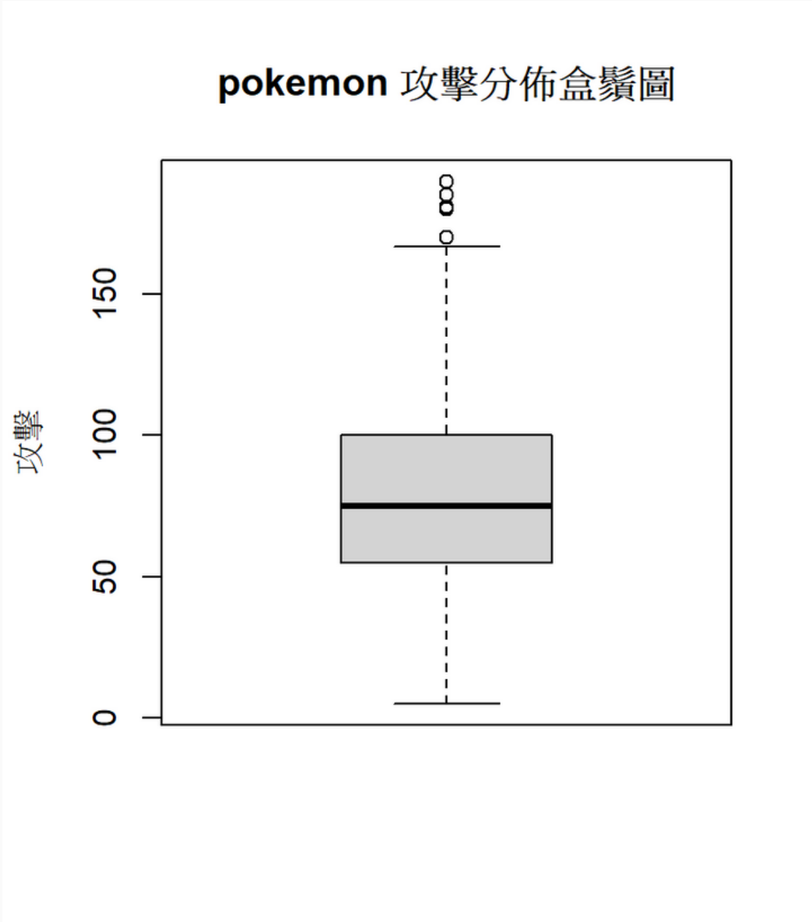

pokemon 身高分佈長條圖
數量
0
100
200
300
400
500
0
3
6
9
12
15
身高(公尺)

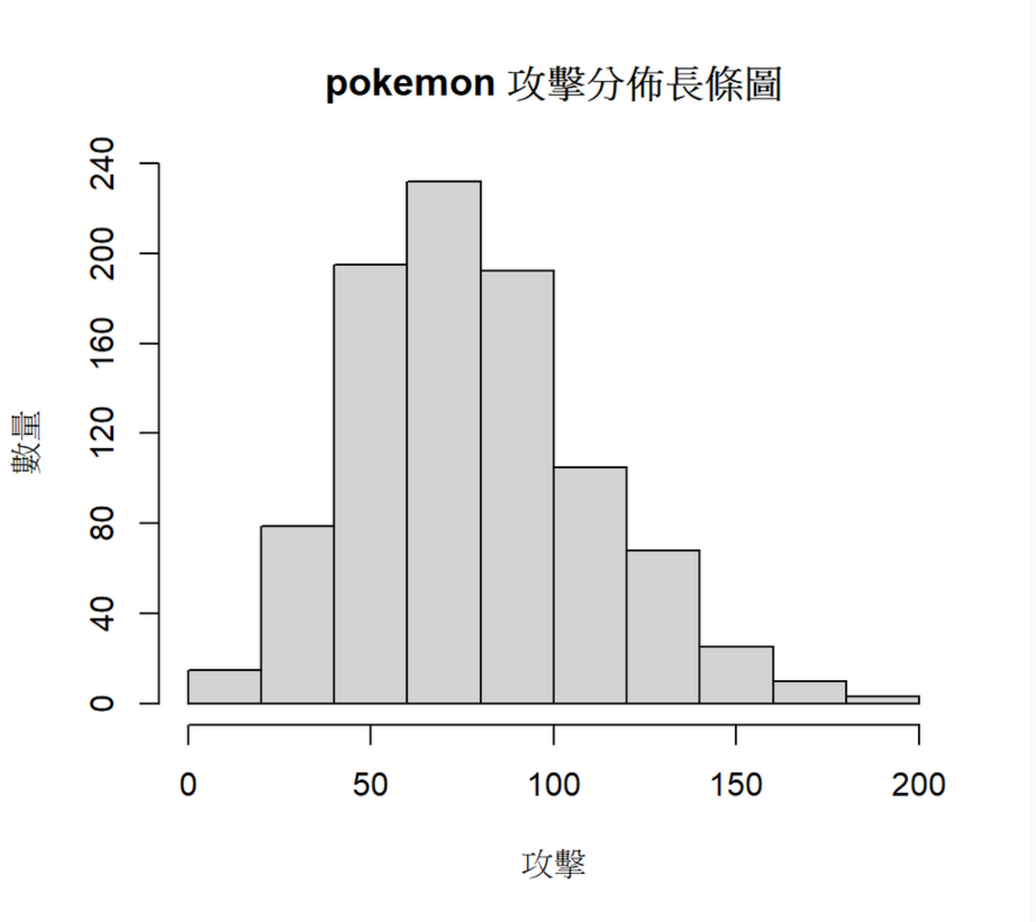

pokemon 體重分佈盒鬚圖
體重(公斤)
0
200
400
600
800
1000

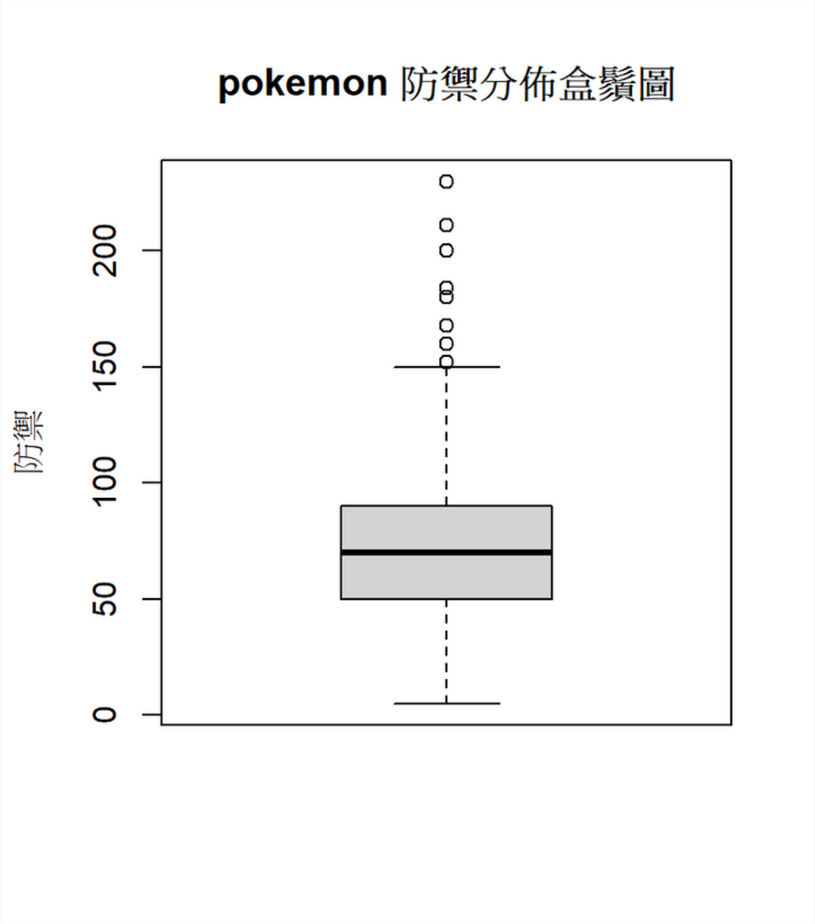

pokemon 體重分佈長條圖
數量
0
110
330
550
770
0
200
400
600
800
1000
體重(公斤)

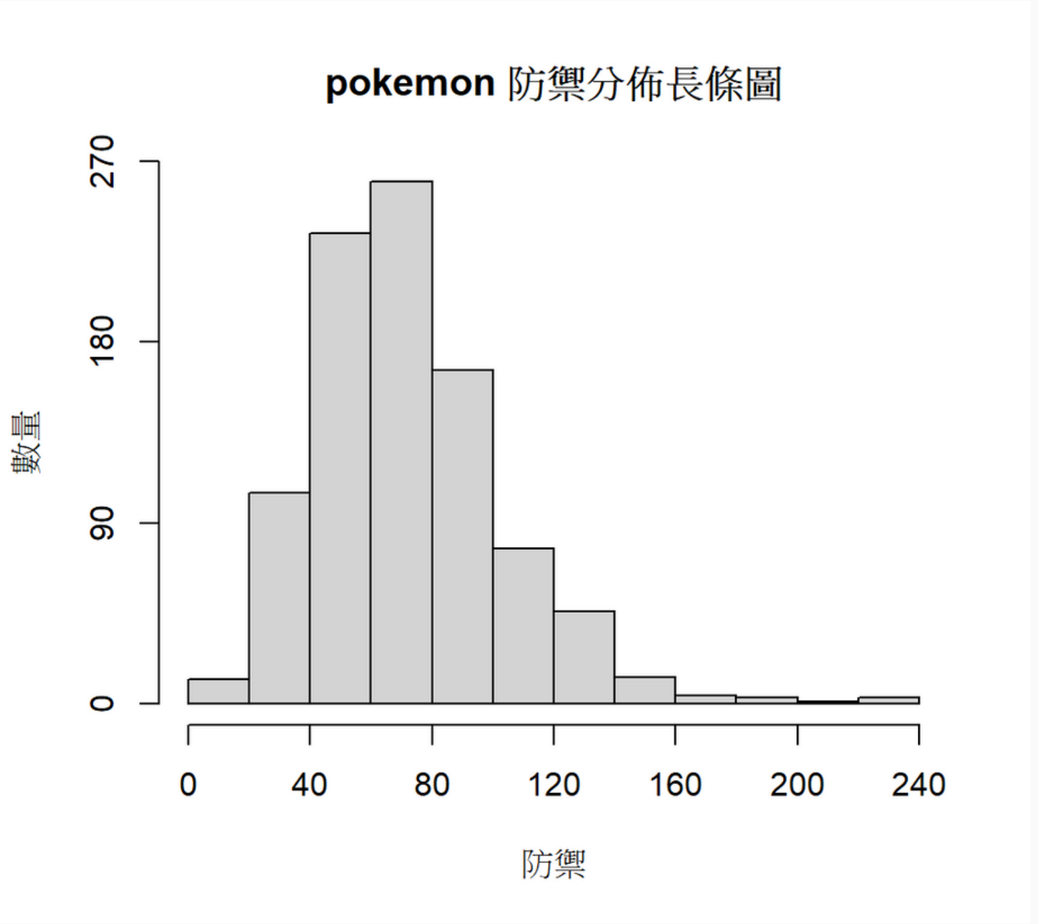

pokemon 血量分佈盒鬚圖
血量
0
50
100
150
200
250

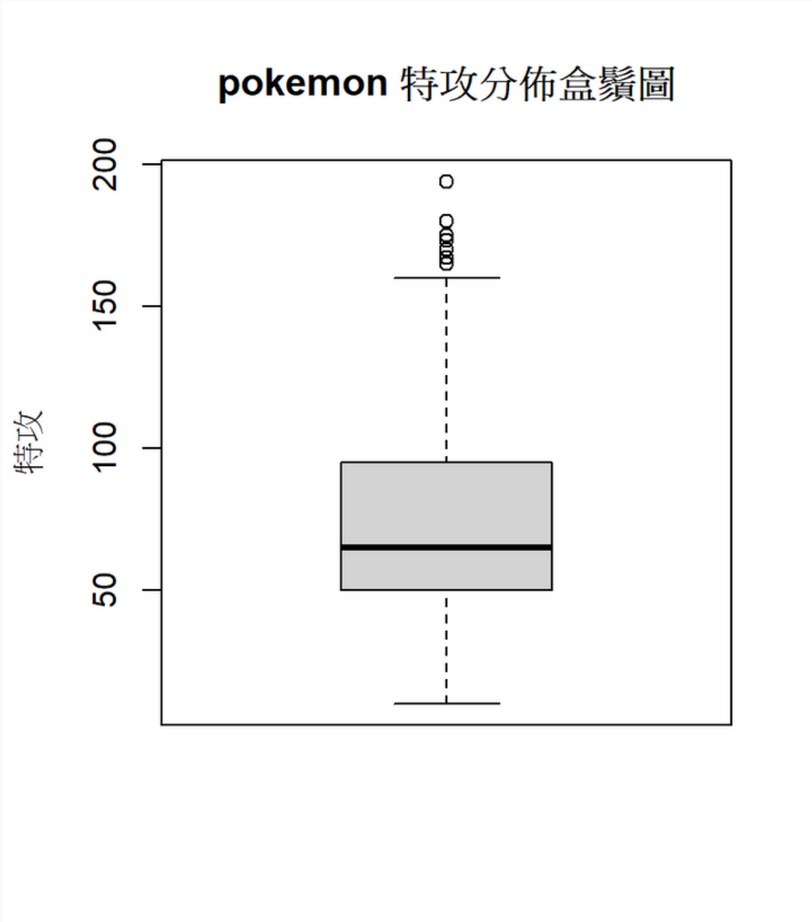

pokemon 血量分佈長條圖
數量
0
64
128
192
256
320
0
90
180
270
血量

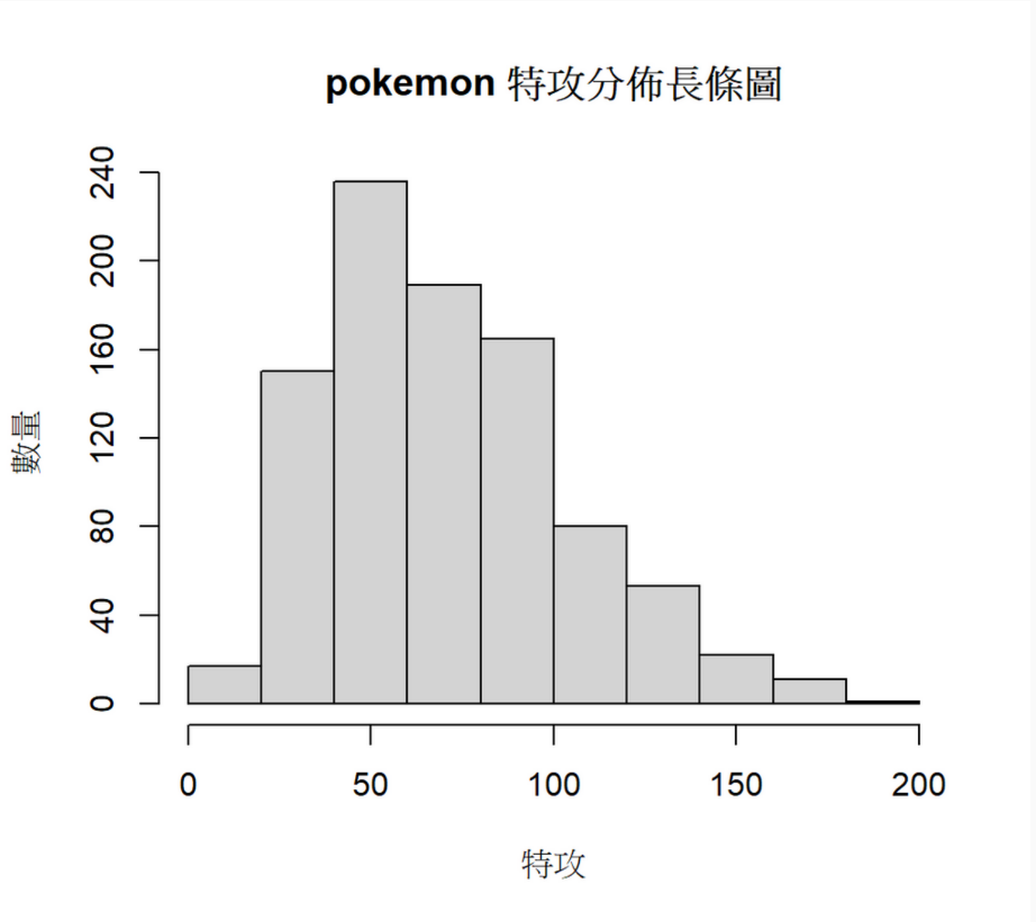

pokemon 攻擊分佈盒鬚圖
攻擊
0
50
100
150

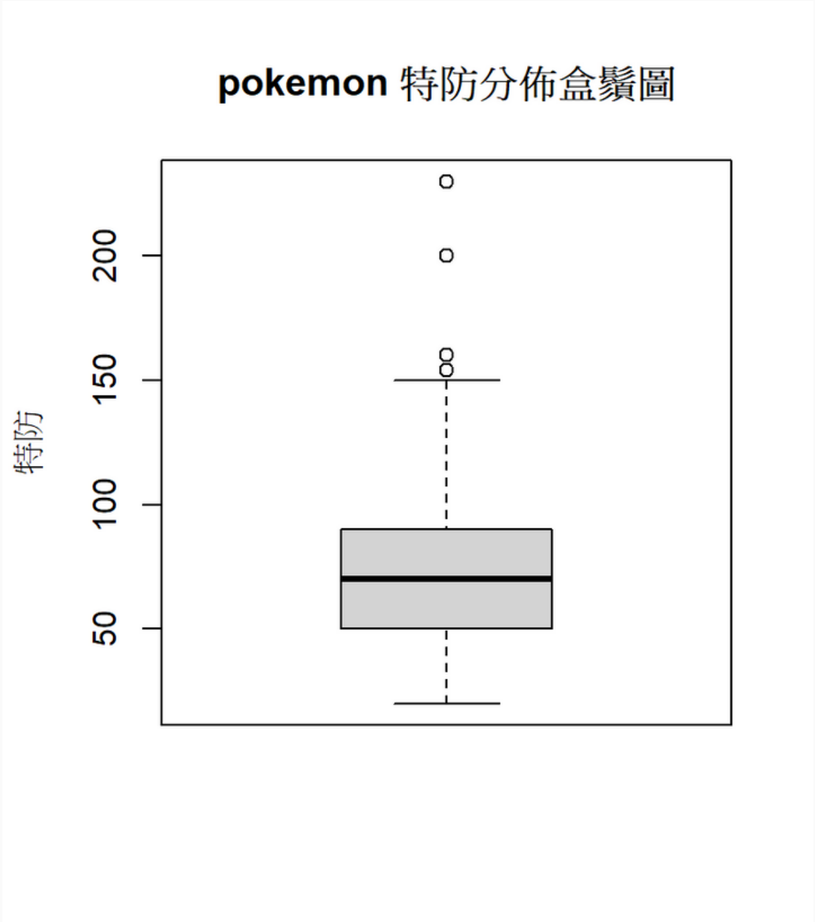

pokemon 攻擊分佈長條圖
數量
0
40
80
120
160
200
240
0
50
100
150
200
攻擊

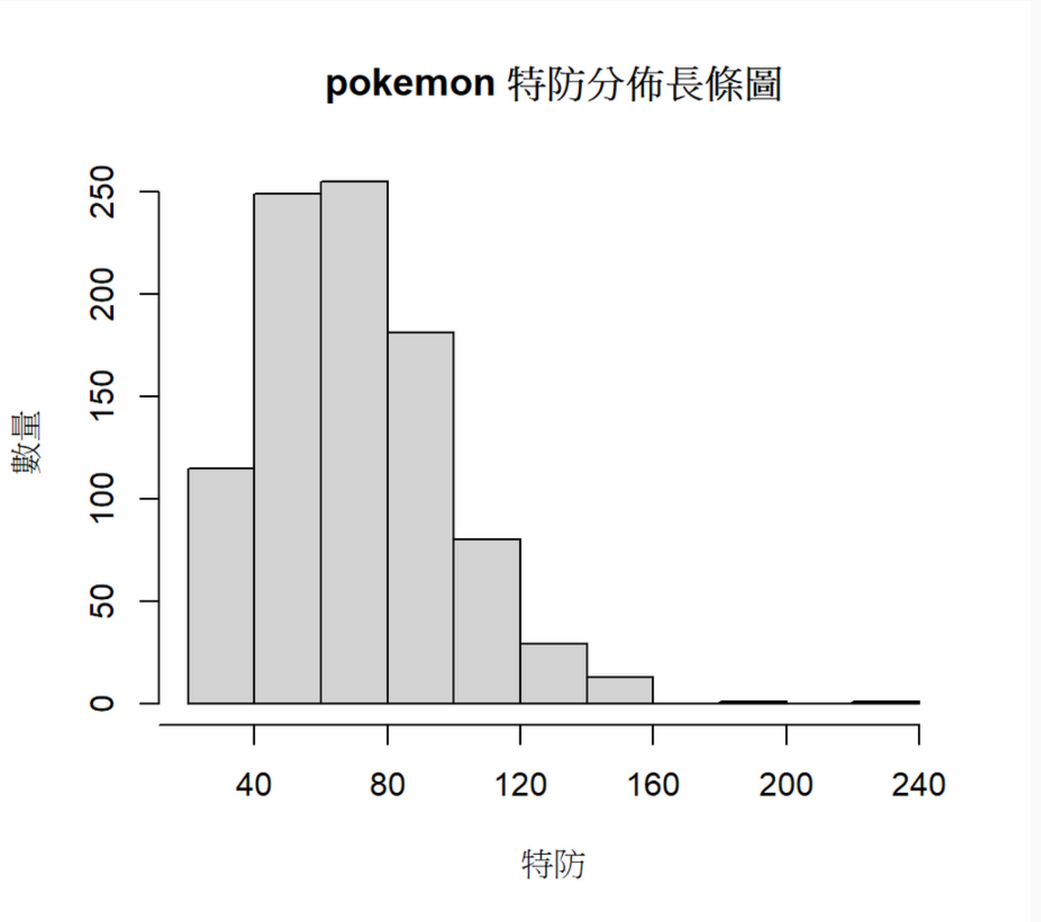

pokemon 防禦分佈盒鬚圖
防禦
0
50
100
150
200

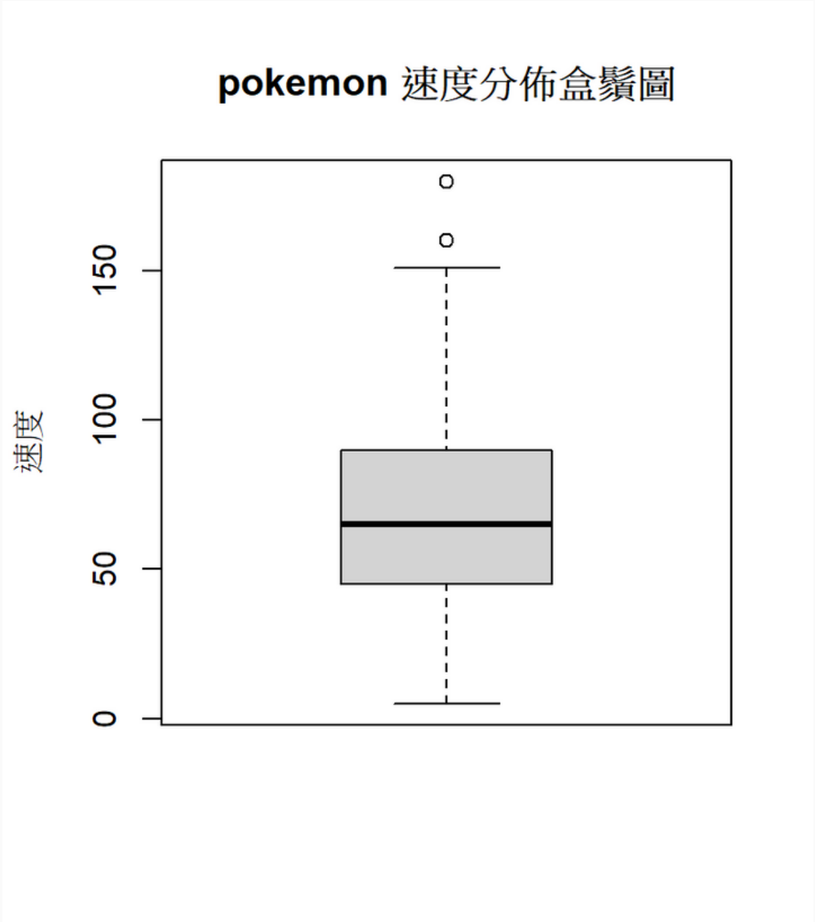

pokemon 防禦分佈長條圖
數量
0
90
180
270
0
40
80
120
160
200
240
防禦

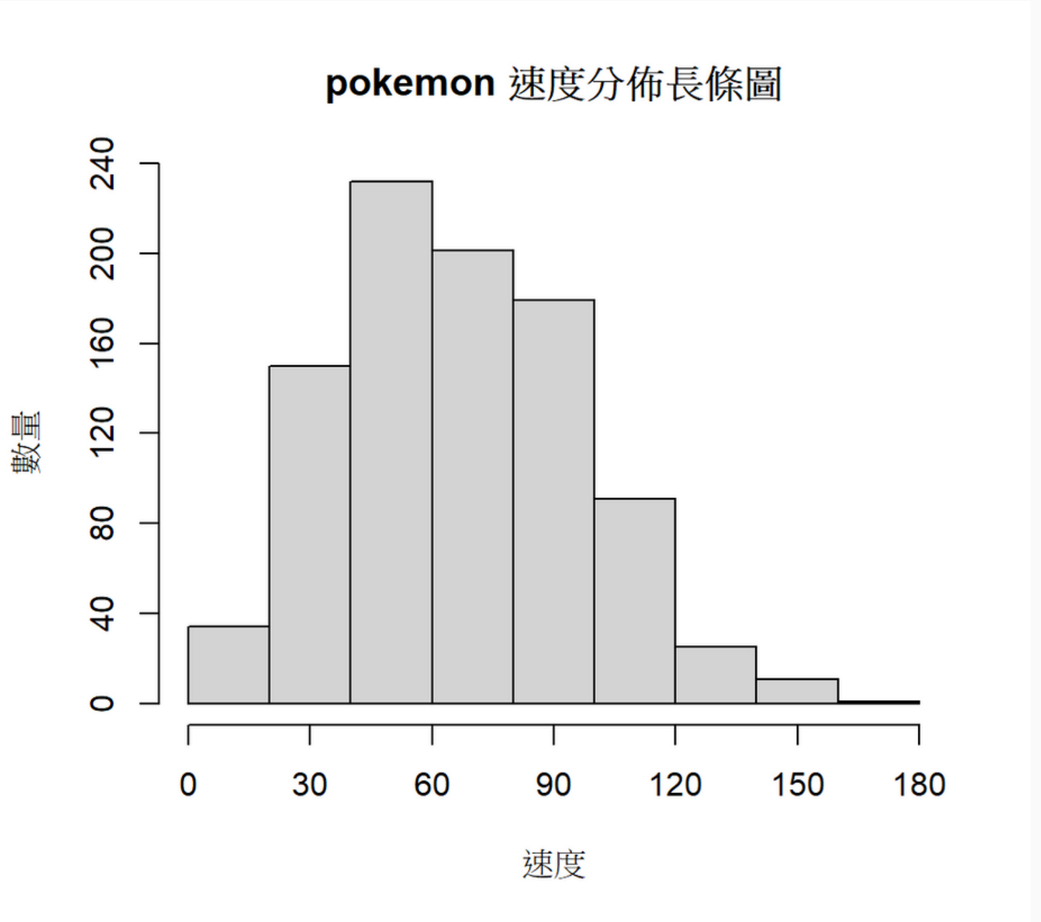

pokemon 特攻分佈盒鬚圖
特攻
200
150
100
50

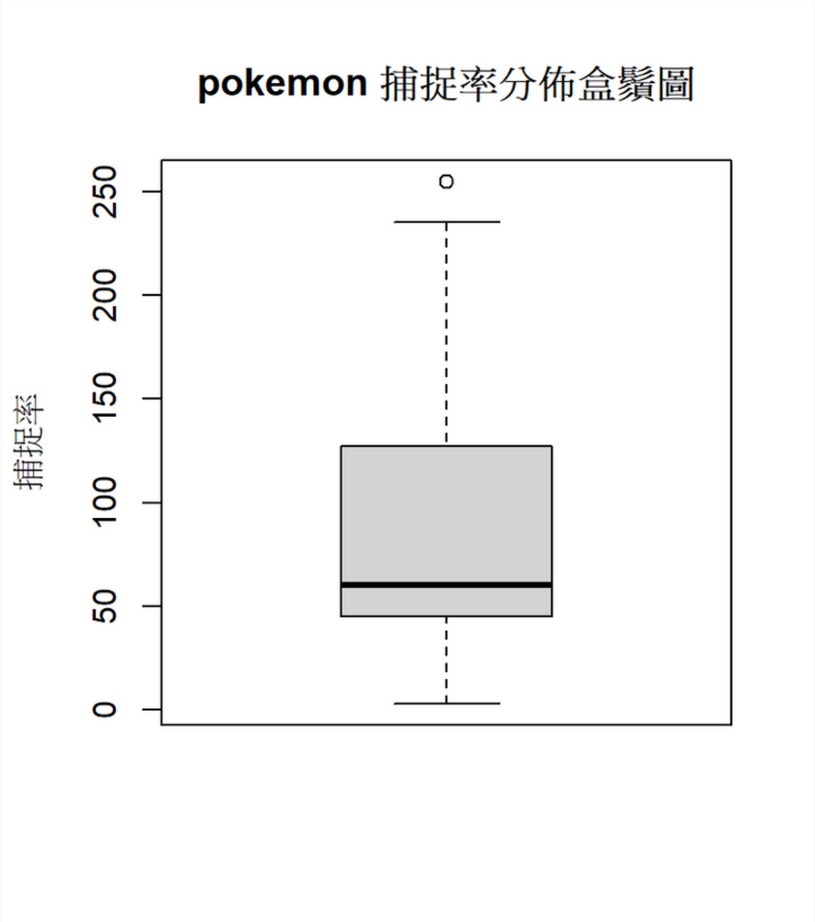

pokemon 特攻分佈長條圖
數量
240
200
160
120
80
40
0
0
50
100
150
200
特攻

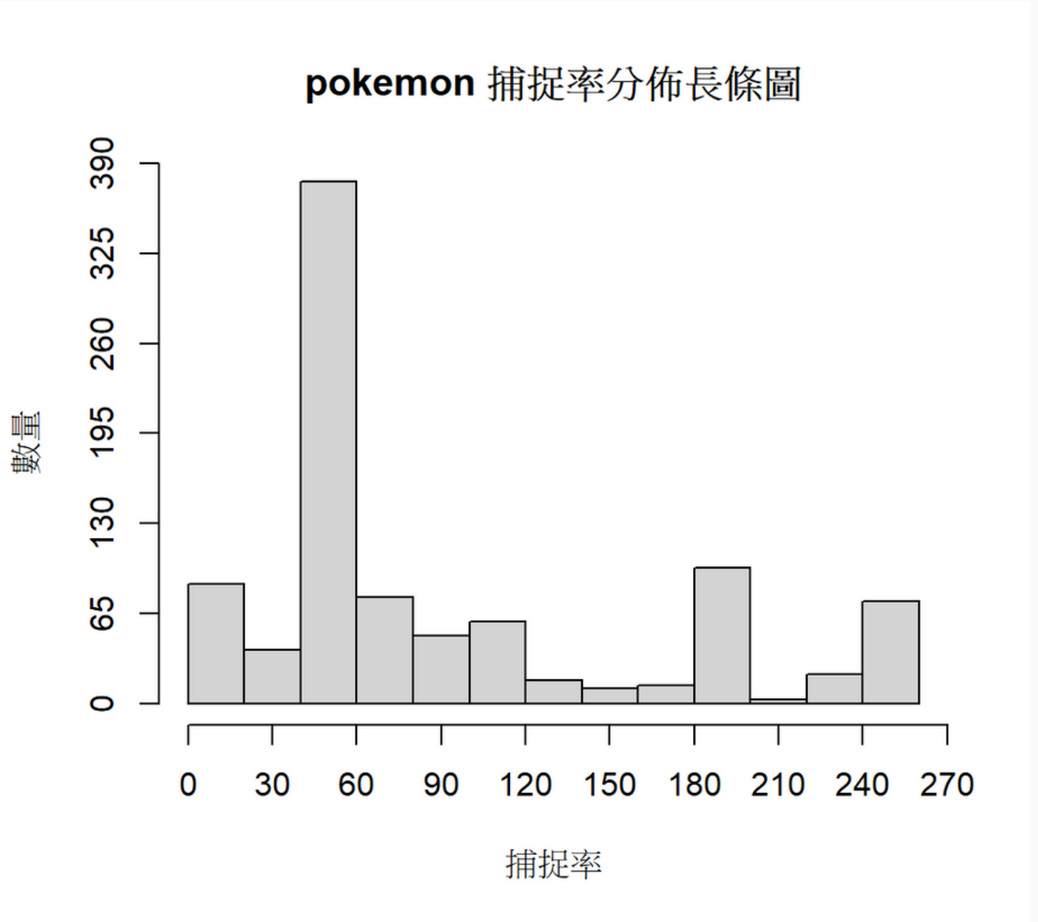

pokemon 特防分佈盒鬚圖
特防
50
100
150
200


pokemon 特防分佈長條圖
數量
0
50
100
150
200
250
40
80
120
160
200
240
特防


pokemon 速度分佈盒鬚圖
速度
0
50
100
150


pokemon 速度分佈長條圖
數量
0
40
80
120
160
200
240
0
30
60
90
120
150
180
速度


pokemon 捕捉率分佈盒鬚圖
捕捉率
0
50
100
150
200
250


pokemon 捕捉率分佈長條圖
數量
0
65
130
195
260
325
390
0
30
60
90
120
150
180
210
240
270
捕捉率

V

分析方法與結論

"Pokémon Catch Rate Prediction."

目標一：
利用寶可夢身高、體重和能力值
來捕捉率預測

利用到哪些課堂知識？

- Model selection：
  在給定數據的情況下，在各個候選模型中選定最優模型的過程。

- Diagnostics for multiple regression：
  通常異常的資料為離群值(outliers)、極端值(extreme value)、影響點(Influential)，因為在資料的分佈中，這些值會遠離其他的資料點，並對模型的參數估計值造成嚴重影響。

- Cluster：
  按照資料之間的差異性，將資料分群。
  常用的分群方法為：K-means ＆ K-means++ ＆ kernel k-means

"Garbage in, garbage out."

- **是否有NA值：**
  我們發現這個資料部分是有的，而且剛好為我們要預測的捕捉率。

- **如何處理NA值：**
  我們發現在1028筆中占有104筆是NA值，占整體的10%，代表說資料其實已經相當空洞了，所以不建議用補值的方法。

- **刪除NA值：**
  因為NA值是我們要預測的變數，同時比例占整體的10%，為了避免"Garbage in, garbage out."，我們選擇留下好數據去做預測，故刪除NA值。

就你最特別-影響點

如果此點對模型的參數估計值影響出現了比例失衡，那麼我們稱之爲強影響點。

```
> ##通常以F(0.5, p, n-p)查表值當作比較的門檻值
> which(as.vector(cooks.distance(model))>qf(0.5,9,
924-9))
integer(0)
> ##DFBETAS與Cook's D一樣是以迴歸係數估計變化量的大小當作影響點的偵測指標，若DFBETAS值大於1，則相對的觀察值可能是影響點
> which(as.matrix(abs(dfbetas(model)))>1)%%924
numeric(0)
> ##DFFITS衡量觀察值被移除後，對於應變數估計值的影響，若DFFITS值大於1，則相對的觀察值可能是影響點
> which(as.vector(abs(dffits(model)))>1)
integer(0)
```

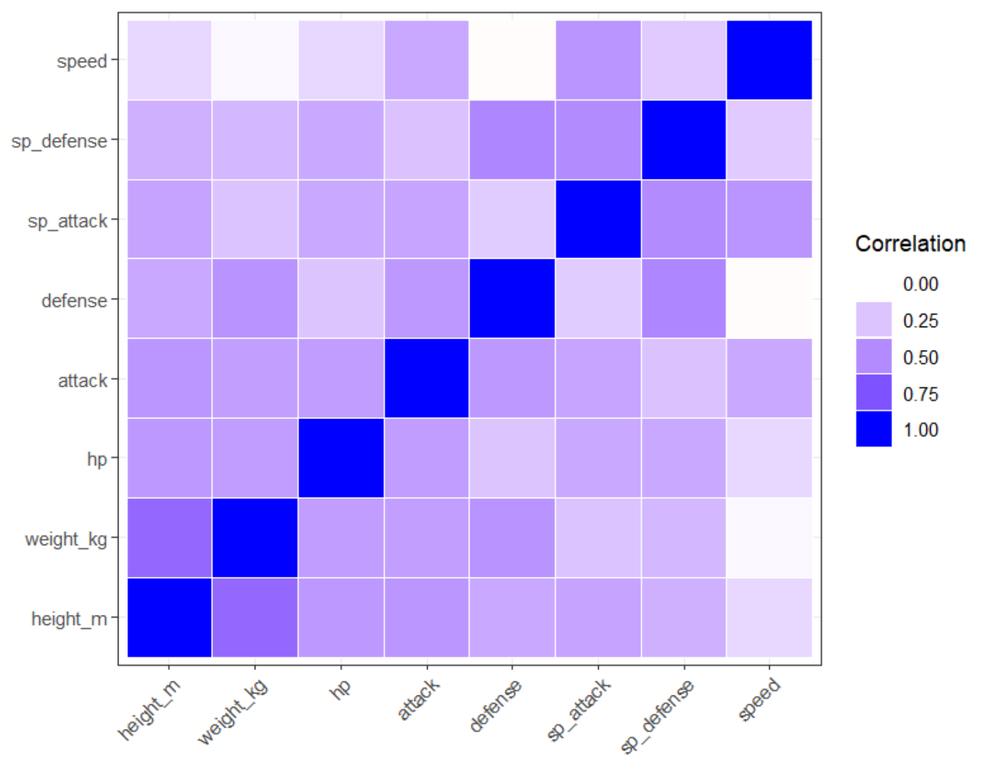
MultiCollinearity

- **多元共線性：**
  存在多元共線性會使得迴歸係數的估計值變得很不精確。

✓ VIF<10：
若VIF值大於10，則可能有共線性問題。

✓ -0.9 <**相關係數** <0.9：
如果相關係數的絕對值大於0.9的話，代表兩個變數間有高度相關。

```
vif(model)
height_m  weight_kg          hp      attack
2.054487   2.060649    1.508789    1.870366
 defense   sp_attack sp_defense       speed
1.972999   1.797862    1.926450    1.488559
```



"How to choose a good model?"

- **CP值法：**
  CP的值與參數個數值愈近愈好。

- Stepwise：
  是向前選取法與向後選取法的結合。

CP 值法

	(Intercept)	height_m	weight_kg	hp	attack	defense	sp_attack	sp_defense	speed	rsq	adjr2	rss	cp	bic
1	1	0	0	0	0	0	1	0	0	0.29	0.289	3709118	477.949	-303.004
1	1	0	0	0	0	0	0	1	0	0.277	0.276	3777607	503.762	-286.098
1	1	0	0	0	1	0	0	0	0	0.27	0.27	3811970	516.713	-277.731
2	1	0	0	0	1	0	0	1	0	0.435	0.434	2953546	195.177	-506.651
2	1	0	0	0	1	0	1	0	0	0.404	0.402	3116669	256.658	-456.978
2	1	0	0	0	0	1	1	0	0	0.4	0.399	3133599	263.038	-451.973
3	1	0	0	0	1	0	1	1	0	0.476	0.475	2736164	115.247	-570.462
3	1	0	0	0	1	0	0	1	1	0.469	0.467	2773929	129.481	-557.796
3	1	0	0	1	0	1	0	0	1	0.466	0.464	2791673	136.168	-551.904
4	1	0	0	1	0	1	1	0	1	0.508	0.506	2569214	54.325	-621.805
4	1	0	0	1	1	0	0	1	1	0.497	0.495	2625883	75.683	-601.646
4	1	0	0	1	1	0	1	1	0	0.495	0.493	2639300	80.739	-596.937
5	1	0	0	1	0	1	1	1	1	0.519	0.516	2515493	36.077	-634.502
5	1	0	0	1	1	1	1	0	1	0.517	0.515	2521927	38.502	-632.141
5	1	0	0	1	1	1	0	1	1	0.515	0.512	2534124	43.099	-627.683
6	1	0	0	1	1	1	1	1	1	0.534	0.531	2437265	8.594	-656.864
6	1	0	1	1	1	1	1	0	1	0.519	0.516	2513884	37.471	-628.264
6	1	0	1	1	0	1	1	1	1	0.519	0.516	2513905	37.479	-628.257
7	1	0	1	1	1	1	1	1	1	0.535	0.531	2429995	7.853	-652.8
7	1	1	0	1	1	1	1	1	1	0.535	0.531	2430123	7.902	-652.75
7	1	1	1	1	1	1	1	0	1	0.519	0.516	2512325	38.883	-622.009
8	1	1	1	1	1	1	1	1	1	0.535	0.531	2427730	9	-646.829

Stepwise

```
Step:  AIC=7292.21
catch_rate ~ weight_kg + hp + attack + defense + sp_attack +
    sp_defense + speed

             Df Sum of Sq     RSS    AIC F value    Pr(>F)
<none>                    2429995 7292.2
- weight_kg   1      7270 2437265 7293.0  2.7406   0.09817 .
+ height_m    1      2264 2427730 7293.3  0.8534   0.35583
- sp_defense  1     83889 2513884 7321.6 31.6225 2.486e-08 ***
- attack      1     83911 2513905 7321.6 31.6305 2.476e-08 ***
- sp_attack   1    100196 2530191 7327.5 37.7696 1.186e-09 ***
- defense     1    115134 2545129 7333.0 43.4005 7.513e-11 ***
- hp          1    137927 2567922 7341.2 51.9922 1.166e-12 ***
- speed       1    141782 2571777 7342.6 53.4455 5.792e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

預測結果

- **候選模型一：**

  height_m 、weight_kg (X)、hp、attack、defense、sp_attack、sp_defense、speed

```
> realdiff <- abs(final$catch_rate-final$future)
> avg_realdiff <- sum(realdiff)/length(realdiff)
> avg_realdiff
[1] 39.32689
```

- **候選模型二：**

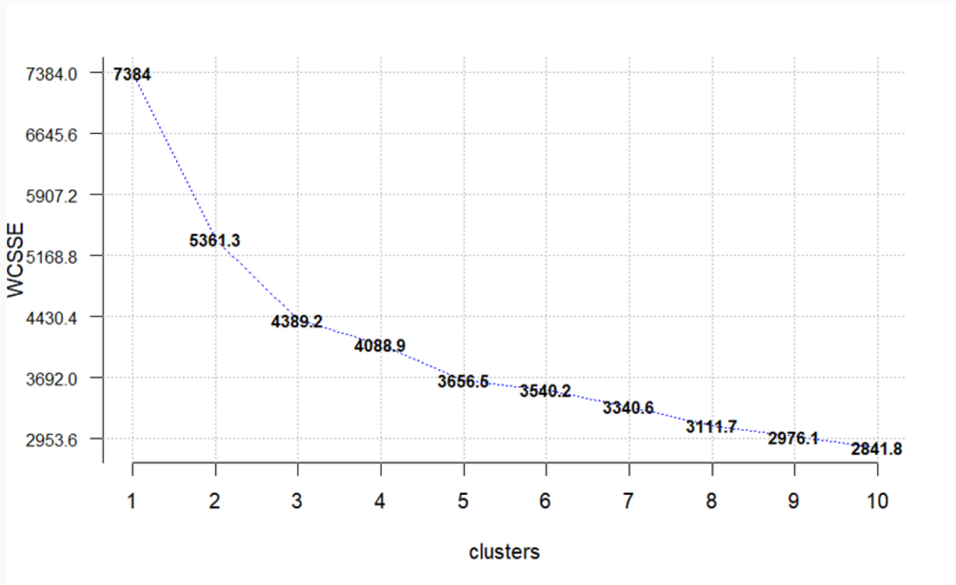
  height_m (X) 、weight_kg、hp、attack、defense、sp_attack、sp_defense、speed

```
> realdiff <- abs(final$catch_rate-final$future)
> avg_realdiff <- sum(realdiff)/length(realdiff)
> avg_realdiff
[1] 39.185
```

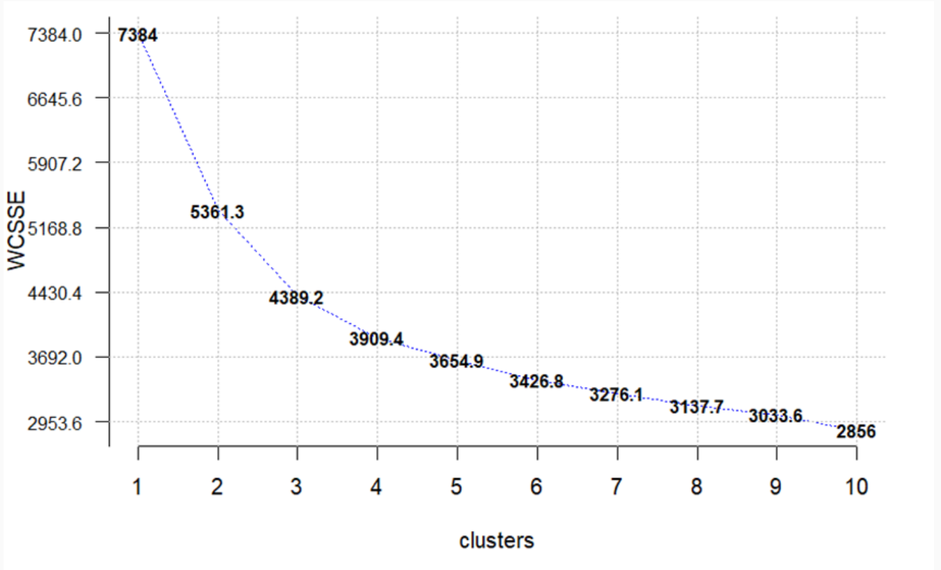
如何更準確

- **轉折點：**
  哪一點之後，圖形逐漸趨緩呢?
  參考兩張圖的轉折點，建議分成3、4、5群。

K-means **分群 組內差異總和圖**



K-means++ **分群 組內差異總和圖**



Clustering Methods

- **分群方法選擇：**
  不論是分3群、分4群、分5群，組內誤差最小都是K-means法。
  所以使用K-means法做分群。

組內誤差	k-means	K-means++	kernel k-means
分3群	4389.2	4389.2	5335.6
分4群	3909.4	4088.9	5806.4
分5群	3654.9	3656.6	5605.3

Best Prediction

- **平均預測誤差最少?**
  分群之後，針對每一群去做線性回歸模型預測捕捉率。
  由下表可知，分5群的預測最準確。
  並且分3、4、5群的預測，都比不分群直接預測還要準確。

- **不分群，平均預測誤差為**39.19
  分3、4、5群的預測，都比不分群直接預測還要準確。

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

模型係數與討論

- 高大且笨重的寶可夢看起來好有壓迫感，一定很難捕捉？
- 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？

	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

同捕捉率
但估計值卻天差地遠

- **是什麼原因影響估計的準確度?**
  我們發現寶可夢的總能力值會影響捕捉率的估計誤差。

實際捕捉率均為45

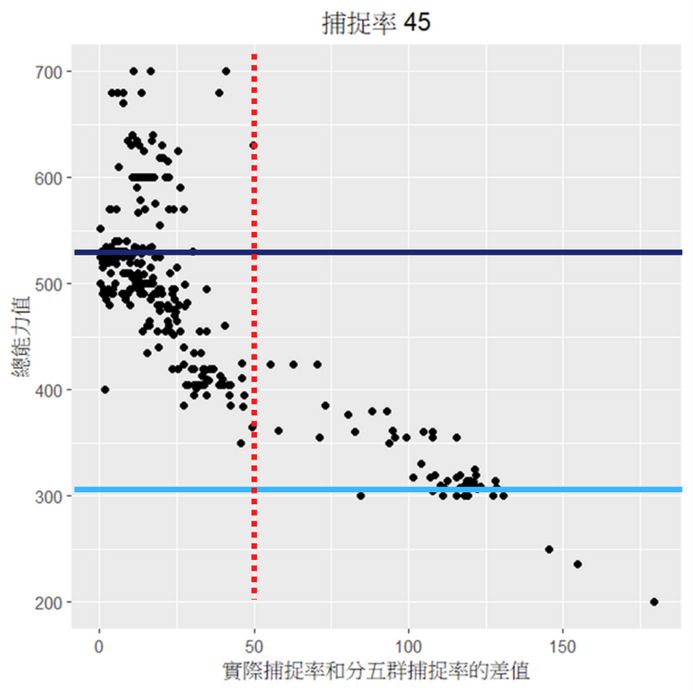
分5群的預測捕捉率

156.65	55.97
實際與預測差很多	實際與預測差不多

同捕捉率
但估計值卻天差地遠

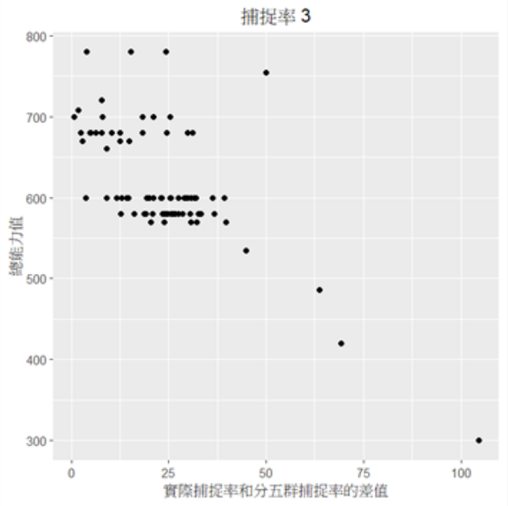
- **寶可夢的總能力值會影響捕捉率的估計誤差**
  對於捕捉率為45的寶可夢，總能力值在400~700間的估計誤差會小於50，反之，總能力值越遠離區間的估計誤差則會越來越大。

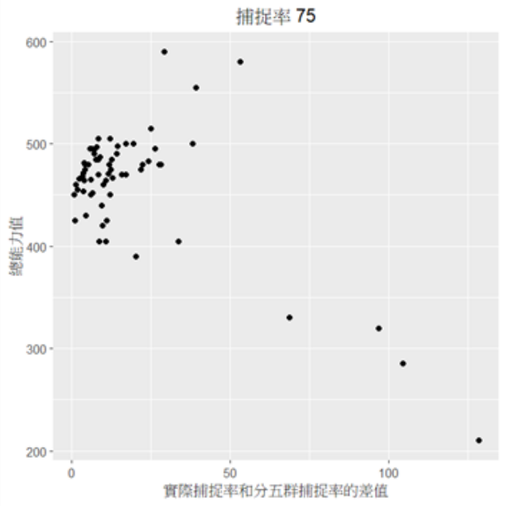
	捕捉率	預測捕捉率	總能力值
水水獺	45	156.65	308
甲賀忍蛙	45	55.97	530

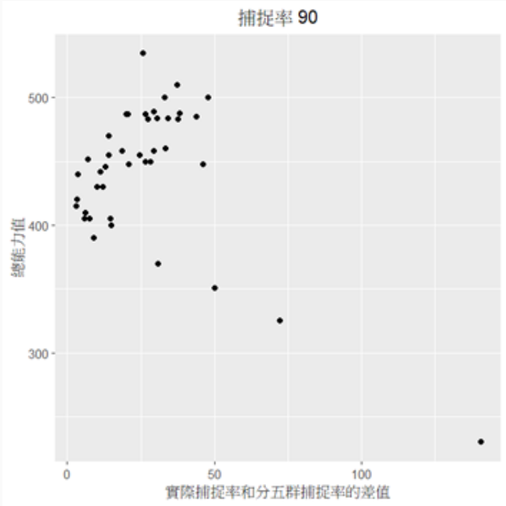


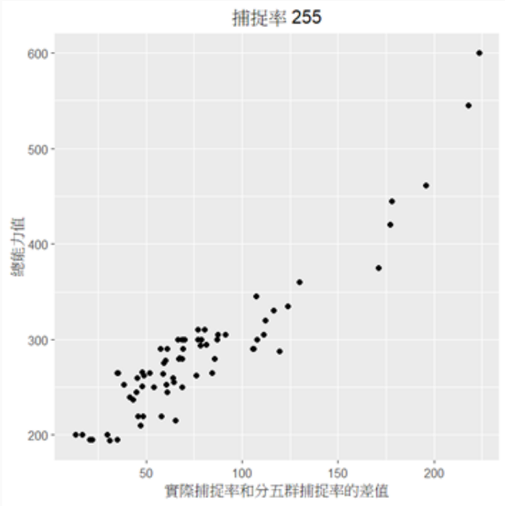
其他捕捉率也有相同的規律

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。









"Clustering of Pokémon ."

目標二:
利用寶可夢的屬性特色來做分類

PCA 分群判斷

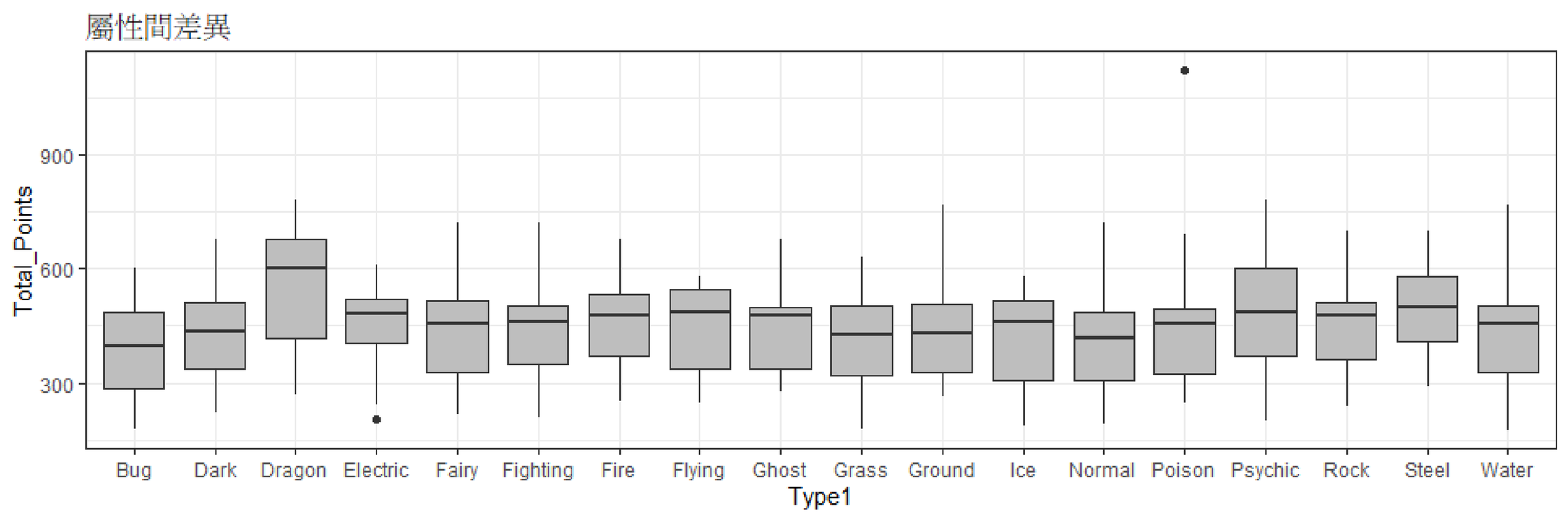
加入新變數(type1、type2)

看能不能成功分群

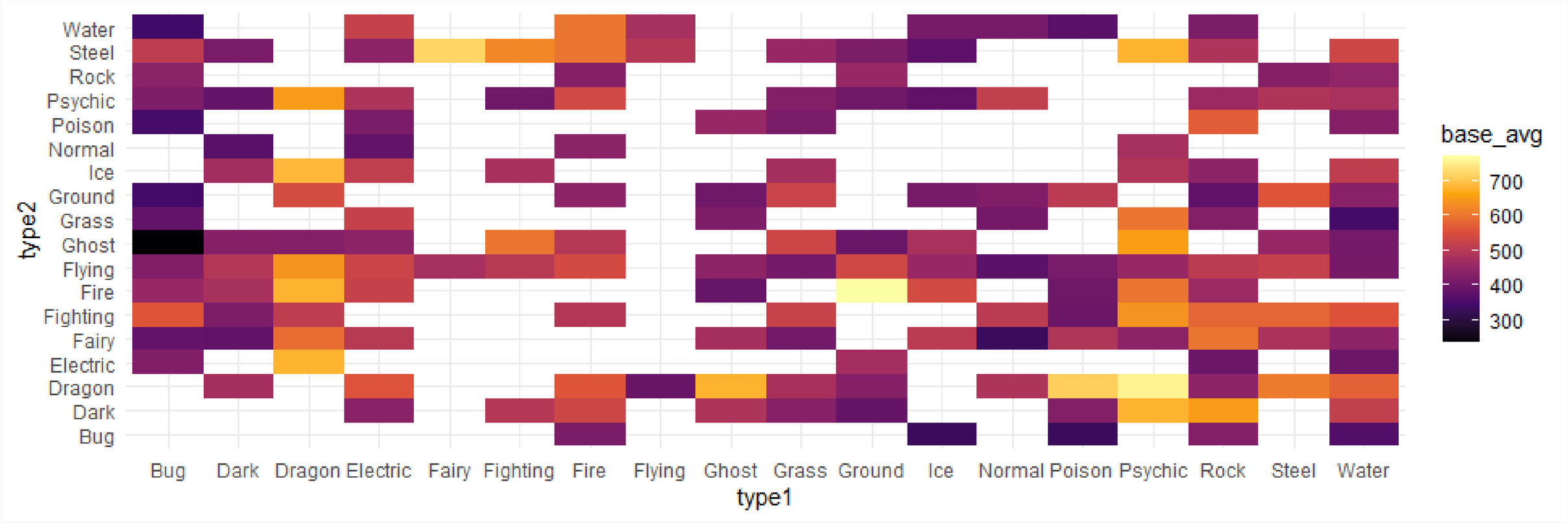
將分群結果歸類

找出規律性

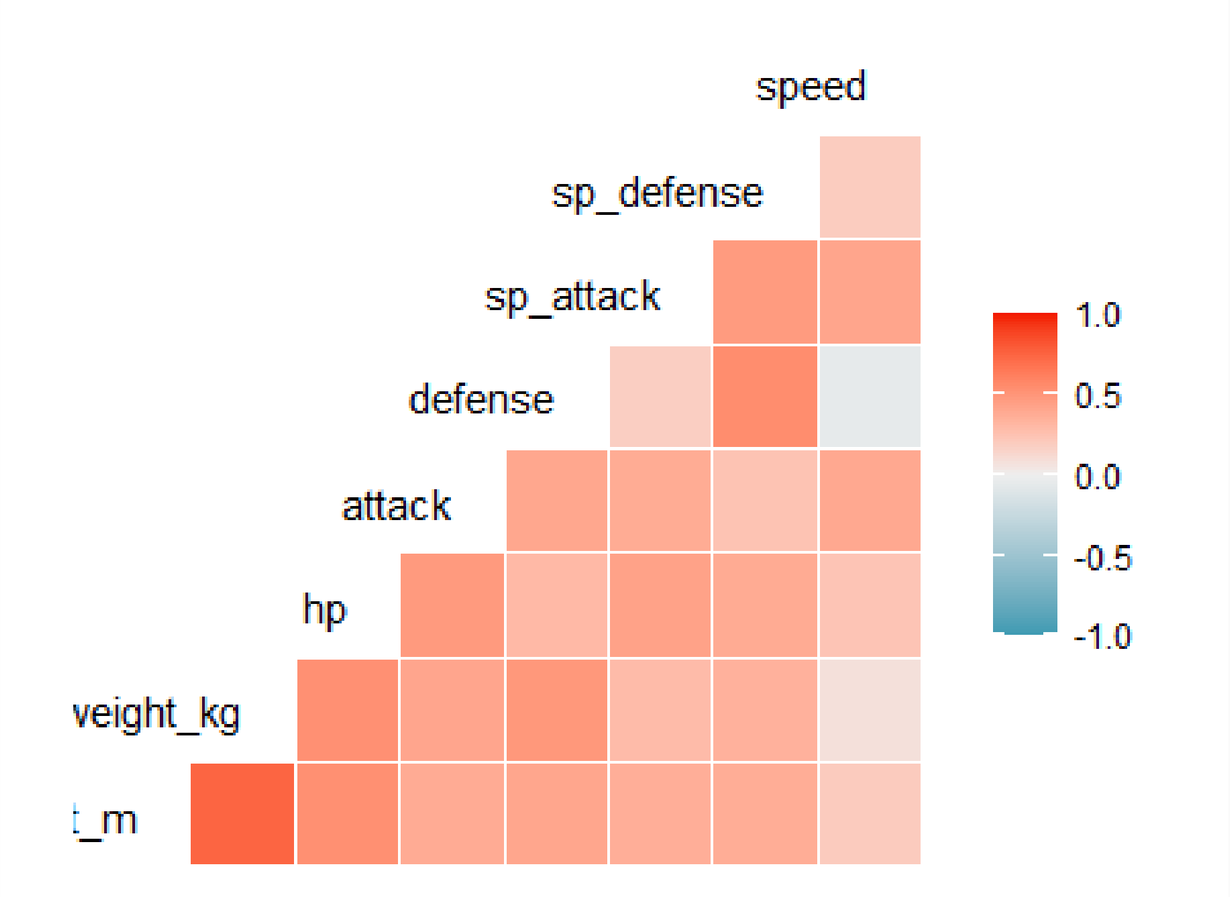
屬性差異


屬性間差異
Total_Points
900
600
300
Bug
Dark
Dragon
Electric
Fairy
Fighting
Fire
Flying
Ghost
Grass
Ground
Ice
Normal
Poison
Psychic
Rock
Steel
Water
Type1

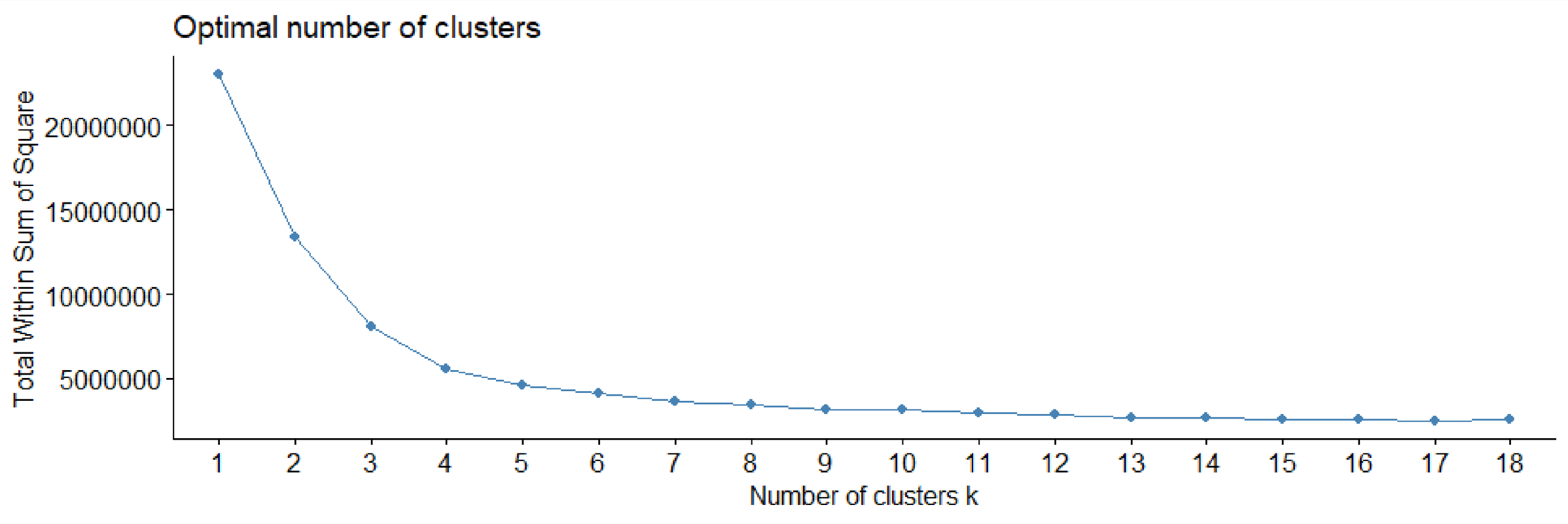
屬性優勢


Water
Steel
Rock
Psychic
Poison
Normal
Ice
Ground
Grass
Ghost
Flying
Fire
Fighting
Fairy
Electric
Dragon
Dark
Bug
type2
Bug
Dark
Dragon
Electric
Fairy
Fighting
Fire
Flying
Ghost
Grass
Ground
Ice
Normal
Poison
Psychic
Rock
Steel
Water
type1
base_avg
700
600
500
400
300

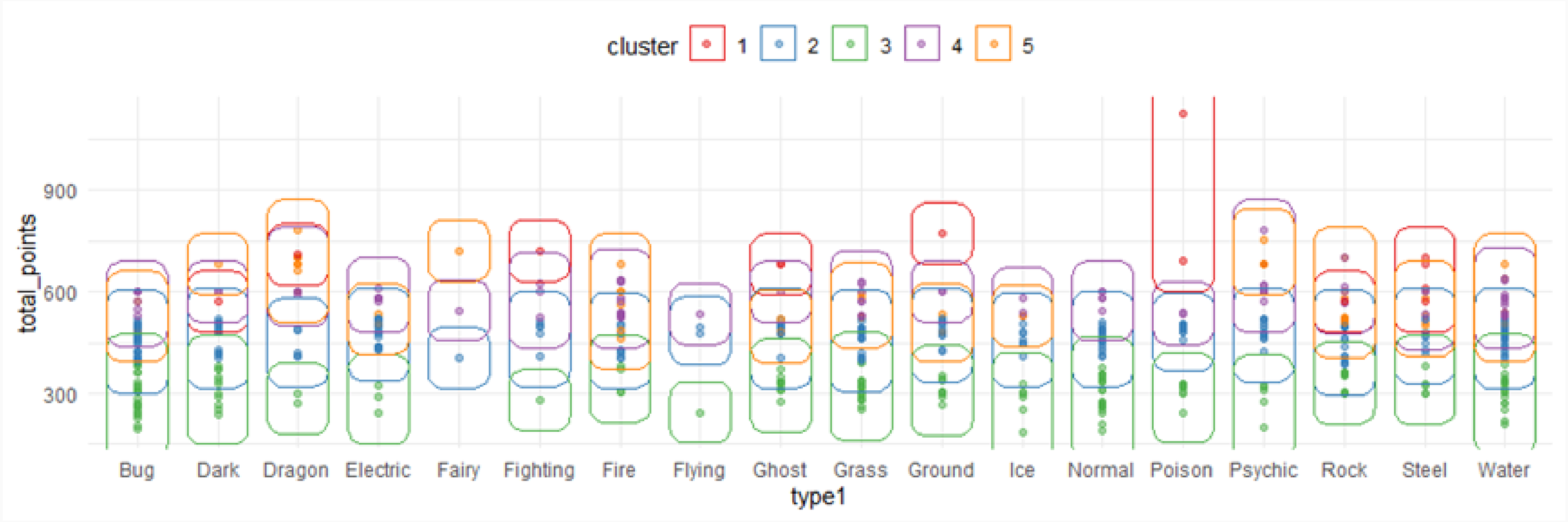
誰造成屬性優勢


speed
sp_defense
sp_attack
defense
attack
hp
veight_kg
t_m
1.0
0.5
0.0
-0.5
-1.0

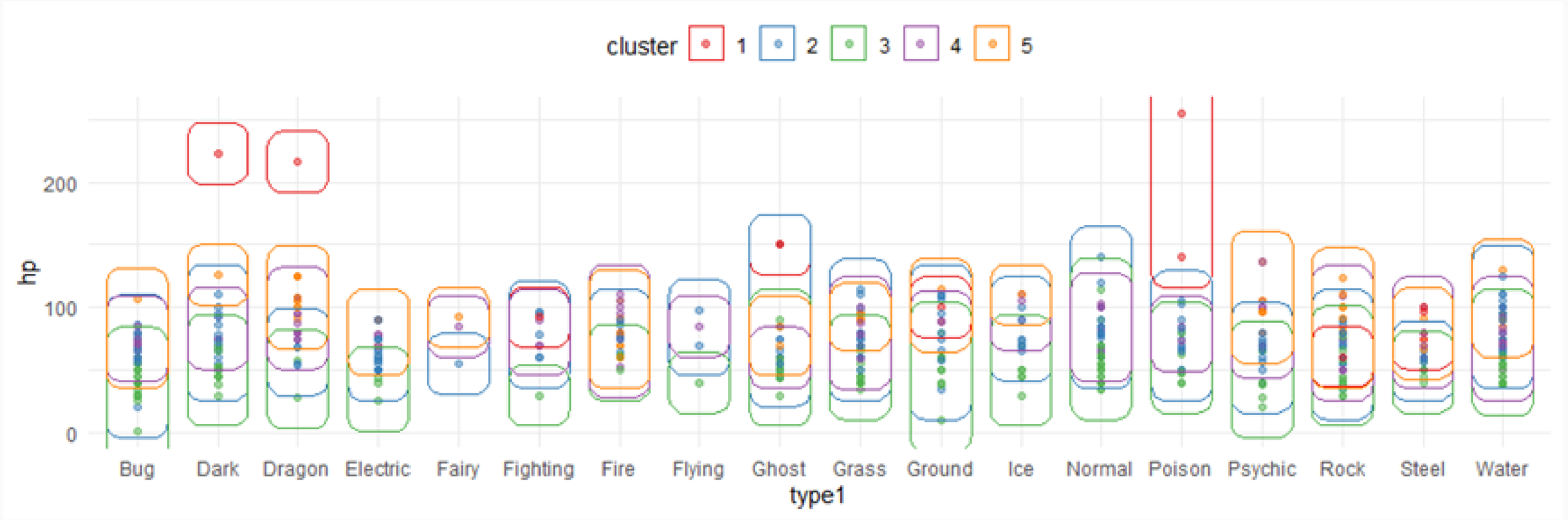
該分幾群


Optimal number of clusters
Total Within Sum of Square
20000000
15000000
10000000
5000000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Number of clusters k

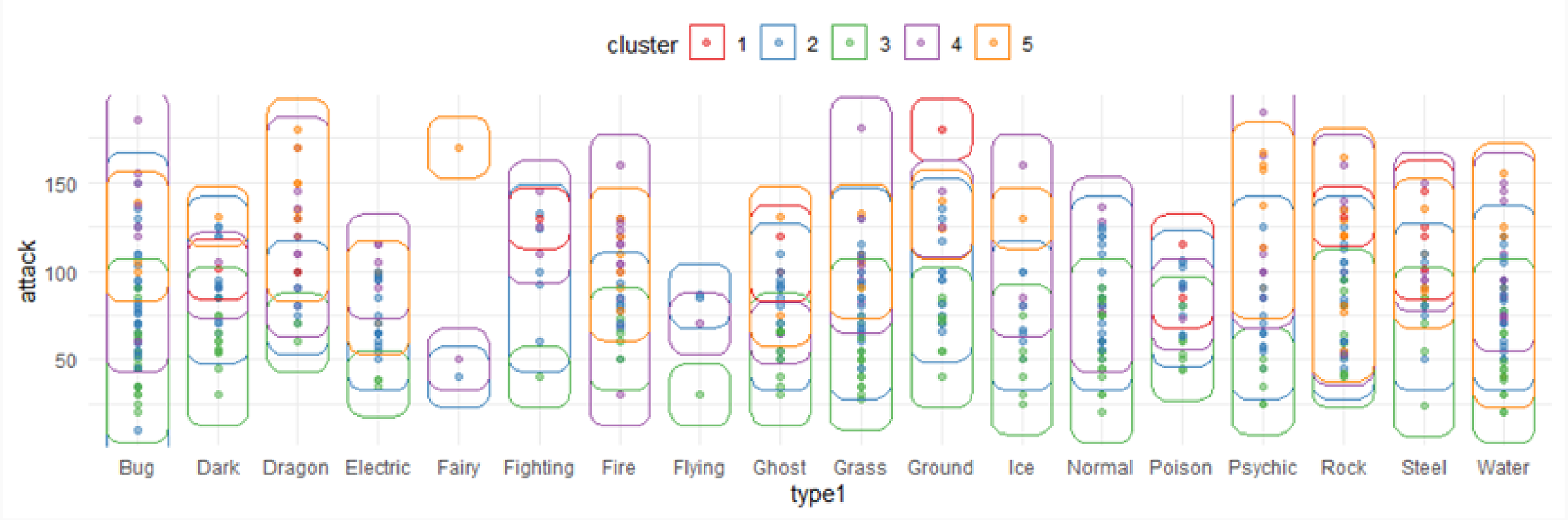
屬性1和總能力值的關係



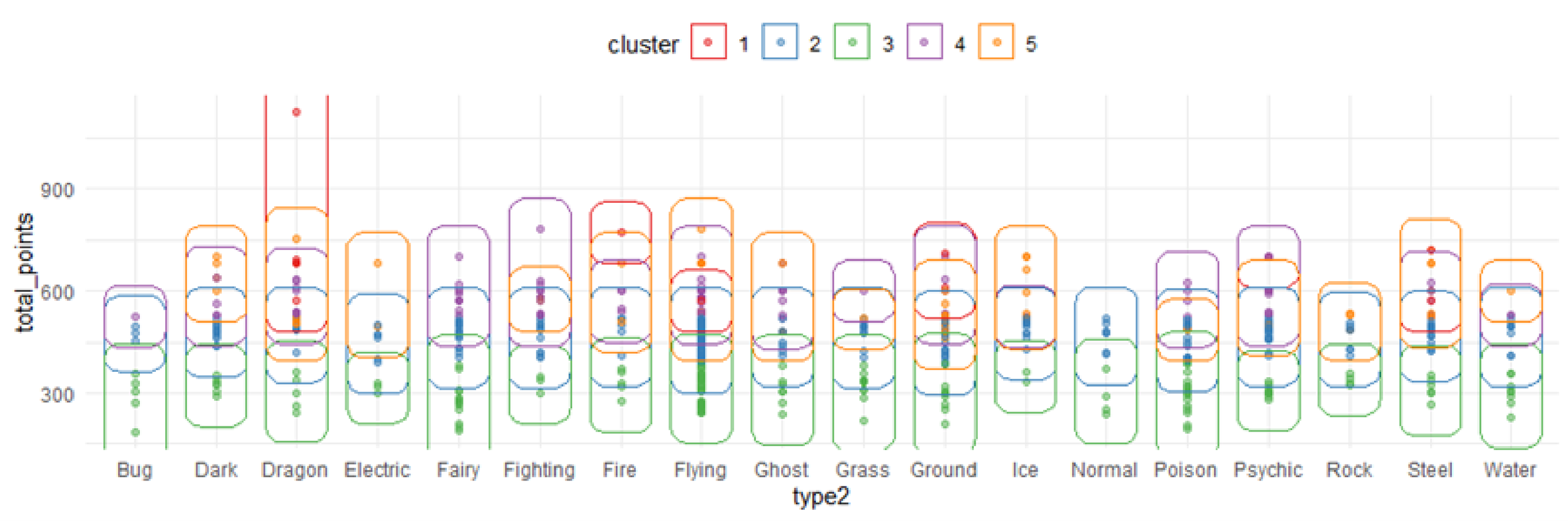
屬性1和血量的關係


cluster
1
2
3
4
5
hp
200
100
0
Bug
Dark
Dragon
Electric
Fairy
Fighting
Fire
Flying
Ghost
Grass
Ground
Ice
Normal
Poison
Psychic
Rock
Steel
Water
type1

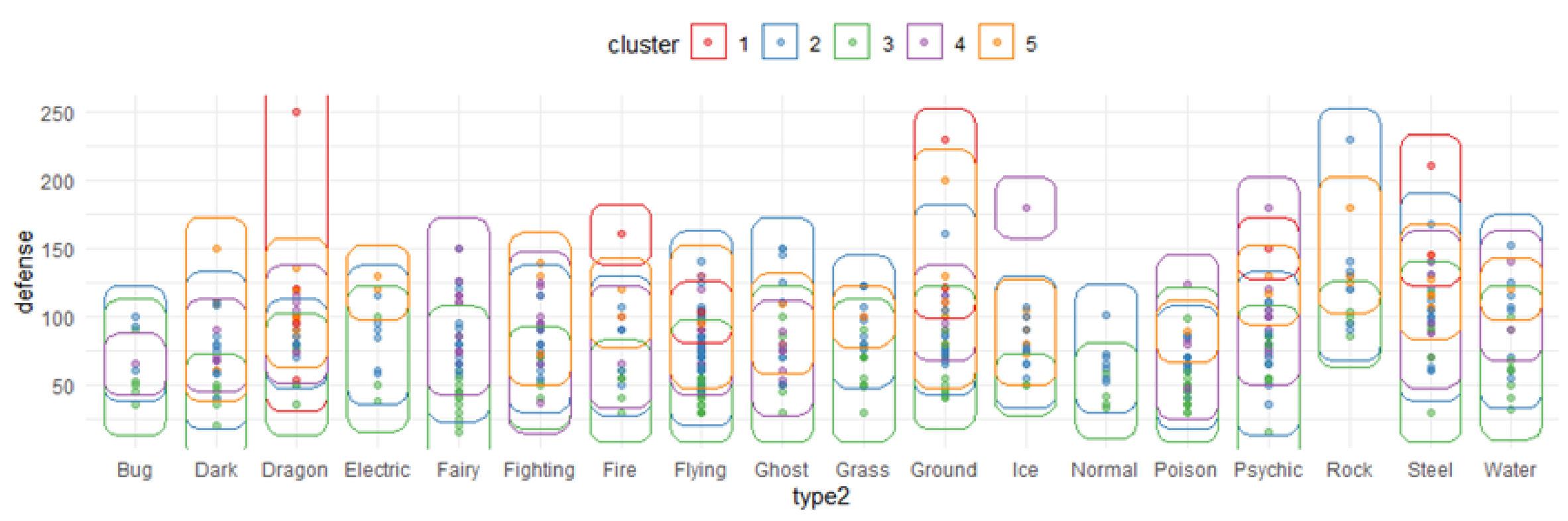
屬性1和防禦的關係


cluster
1
2
3
4
5
attack
150
100
50
Bug
Dark
Dragon
Electric
Fairy
Fighting
Fire
Flying
Ghost
Grass
Ground
Ice
Normal
Poison
Psychic
Rock
Steel
Water
type1

屬性2和總能力值的關係



屬性2和防禦的關係


cluster
1
2
3
4
5
defense
250
200
150
100
50
Bug
Dark
Dragon
Electric
Fairy
Fighting
Fire
Flying
Ghost
Grass
Ground
Ice
Normal
Poison
Psychic
Rock
Steel
Water
type2

降維

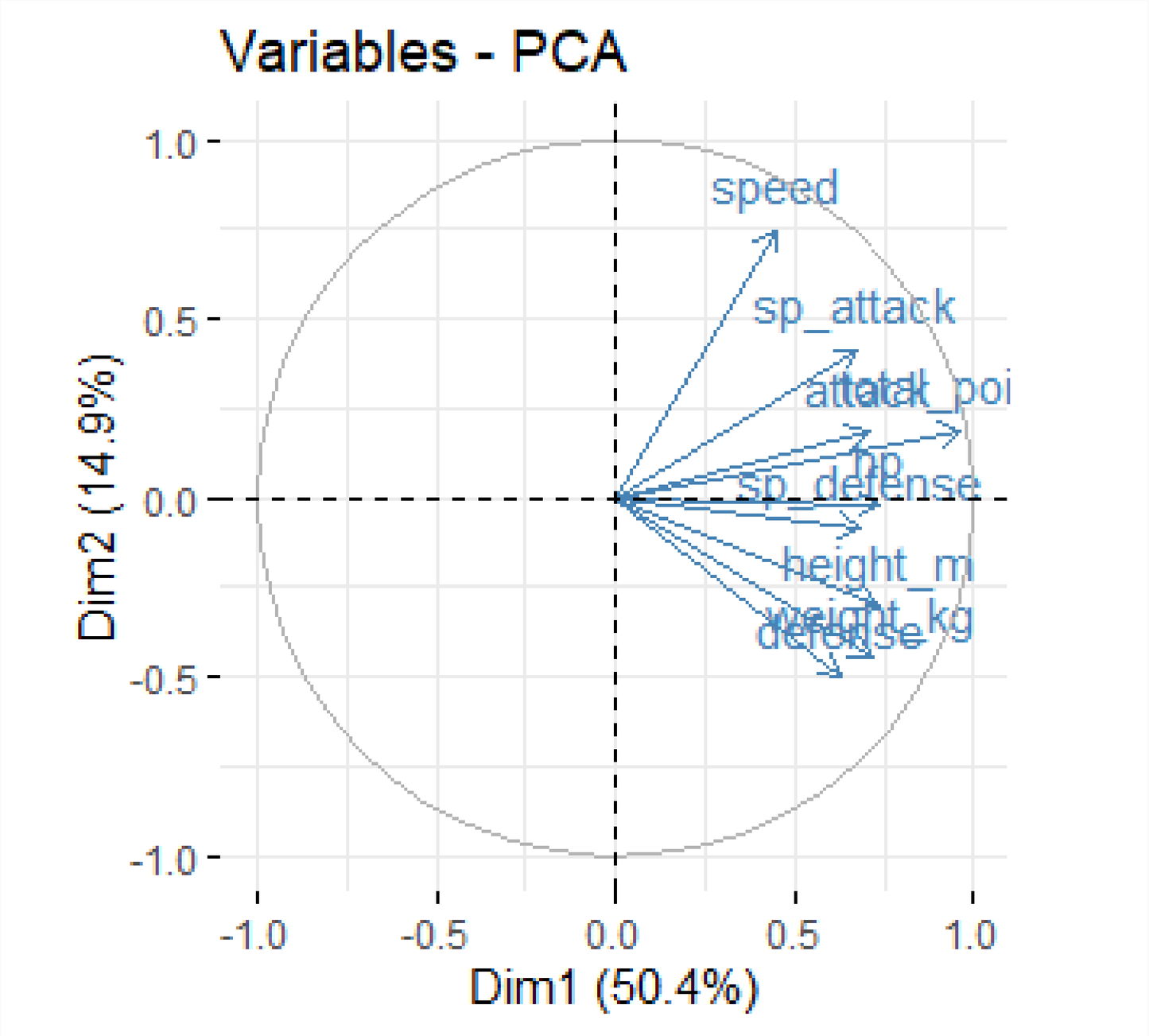
```
> summary(poke_pca)

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
```

降維

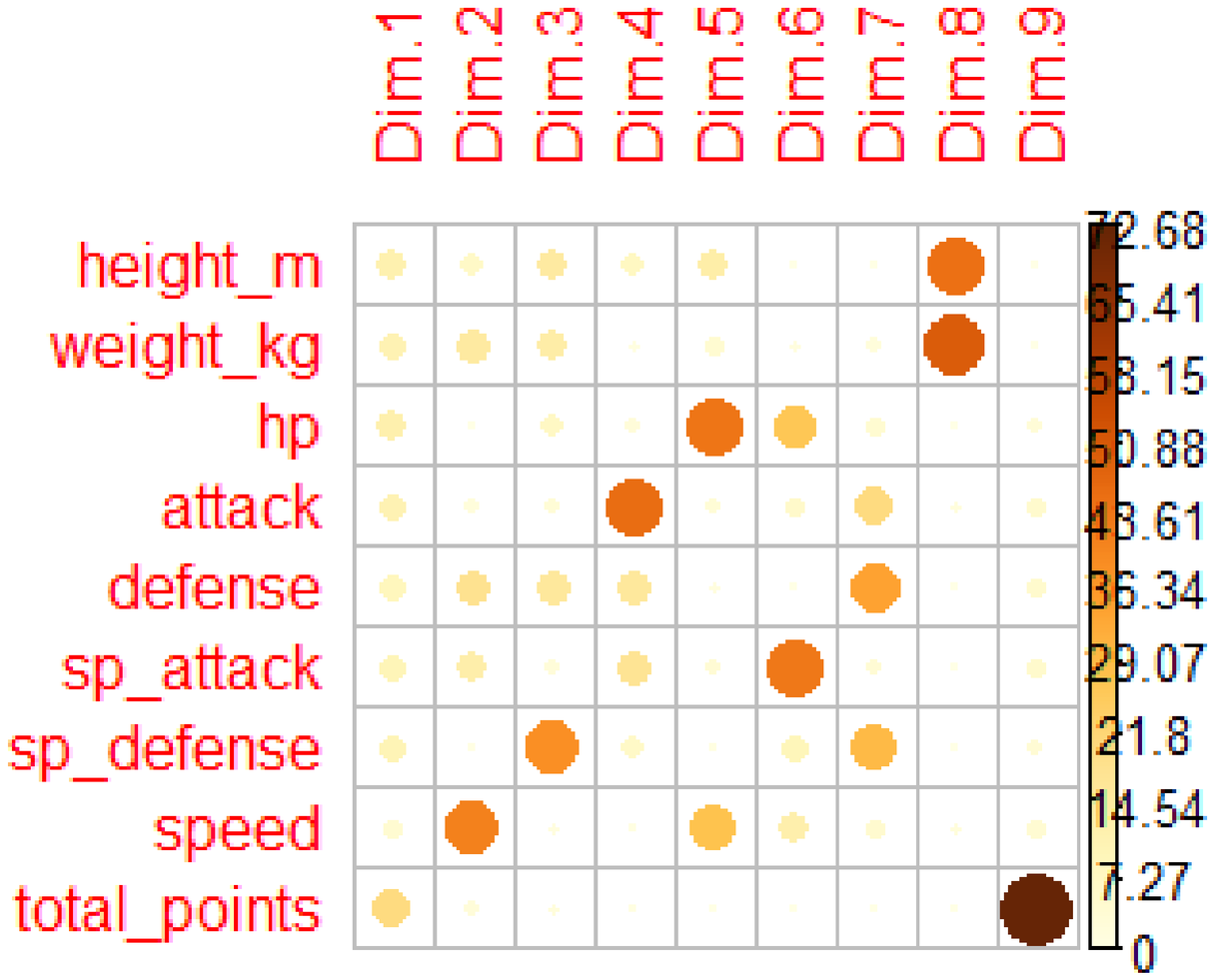
Variables

	Dim.1	ctr	cos2	Dim.2	ctr	cos2	Dim.3	ctr	cos2
height_m	0.732	11.808	0.536	-0.304	6.884	0.092	0.370	14.607	0.137
weight_kg	0.715	11.254	0.511	-0.445	14.798	0.198	0.355	13.479	0.126
hp	0.740	12.074	0.548	-0.019	0.027	0.000	0.261	7.272	0.068
attack	0.706	10.980	0.499	0.184	2.520	0.034	0.181	3.513	0.033
defense	0.630	8.737	0.397	-0.495	18.275	0.245	-0.385	15.805	0.148
sp_attack	0.675	10.033	0.456	0.413	12.726	0.171	-0.159	2.714	0.025
sp_defense	0.681	10.222	0.464	-0.082	0.501	0.007	-0.600	38.403	0.360
speed	0.445	4.368	0.198	0.747	41.645	0.558	0.129	1.789	0.017
total_points	0.965	20.524	0.932	0.188	2.624	0.035	-0.150	2.418	0.023

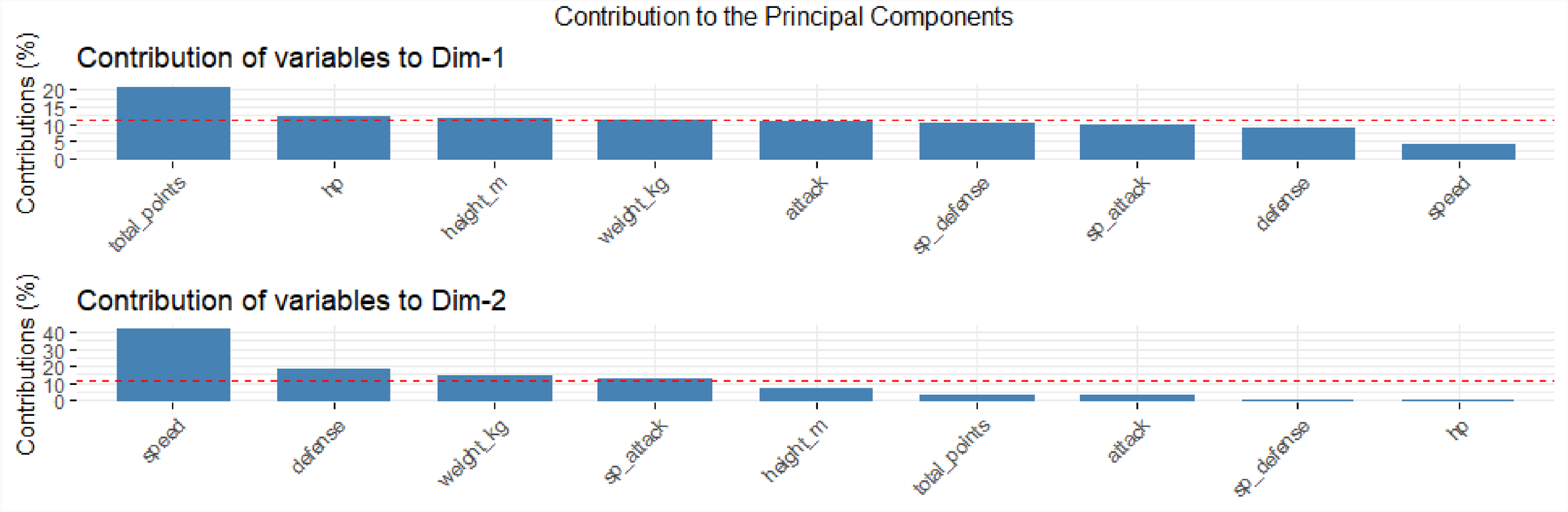
降維


Variables - PCA
1.0
0.5
0.0
-0.5
-1.0
Dim2 (14.9%)
speed
sp_attack
attack
hp
sp_defense
height_m
weight_kg
defense
-1.0
-0.5
0.0
0.5
1.0
Dim1 (50.4%)

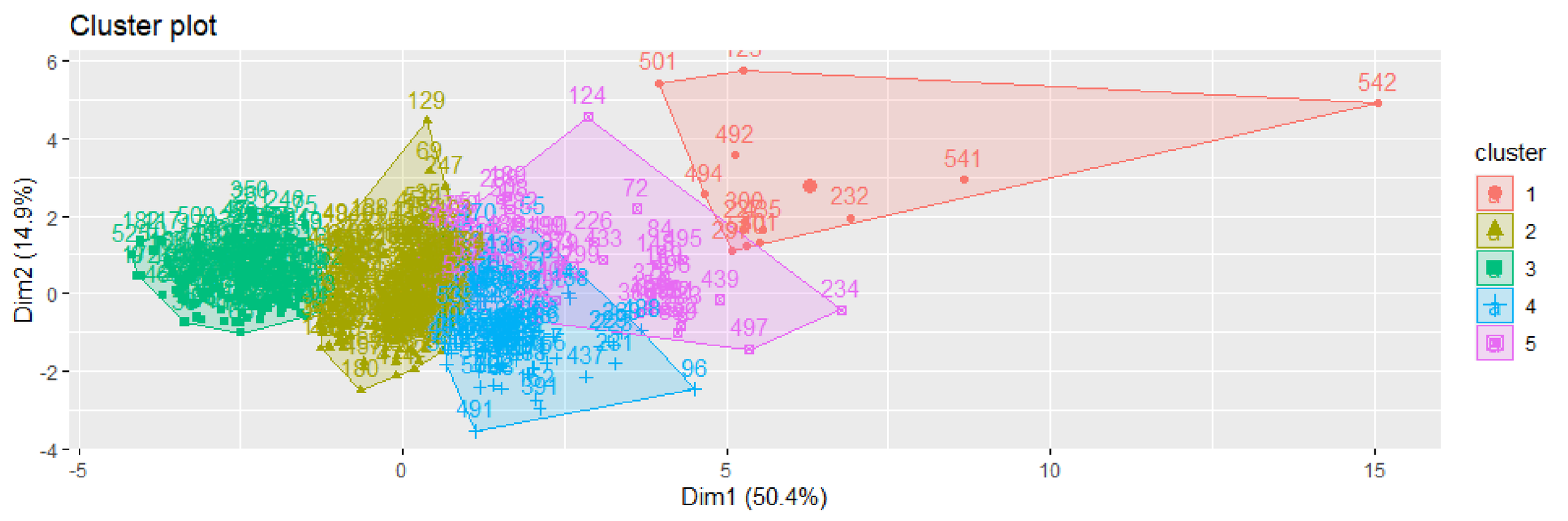
組成維度


Dim.1
Dim.2
Dim.3
Dim.4
Dim.5
Dim.6
Dim.7
Dim.8
Dim.9
height_m
weight_kg
hp
attack
defense
sp_attack
sp_defense
speed
total_points
72.68
65.41
58.15
50.88
43.61
36.34
29.07
21.8
14.54
7.27
0

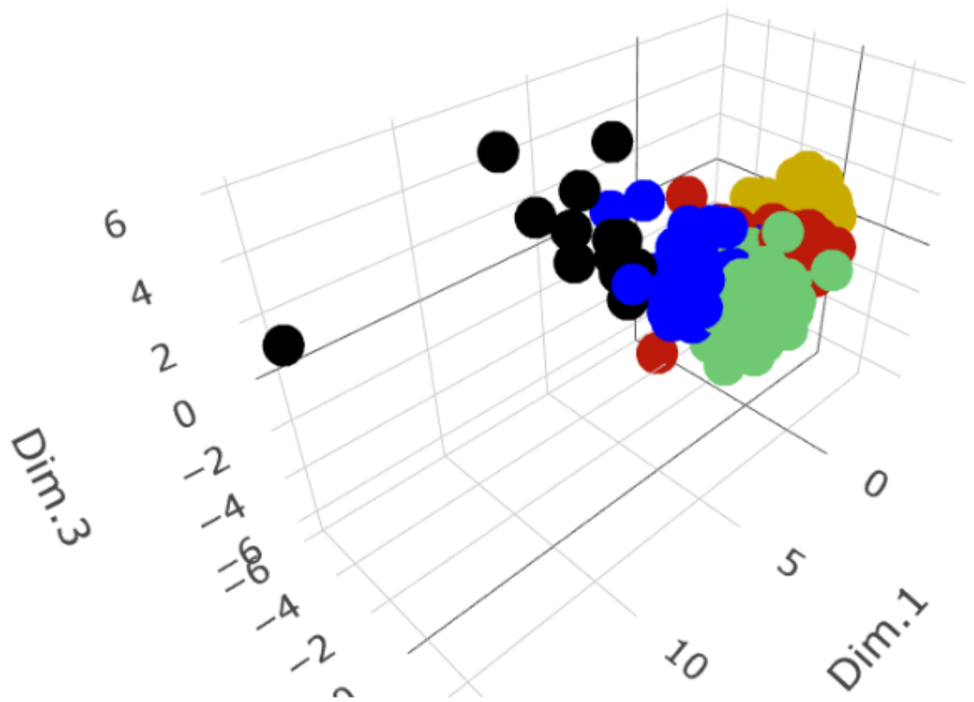
組成維度

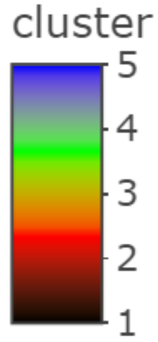

Contribution to the Principal Components
Contribution of variables to Dim-1
Contributions (%)
0
5
10
15
20
total_points
hp
height_m
weight_kg
attack
sp_defense
sp_attack
defense
speed
Contribution of variables to Dim-2
Contributions (%)
0
10
20
30
40
speed
defense
weight_kg
sp_attack
height_m
total_points
attack
sp_defense
hp

2D 看分群

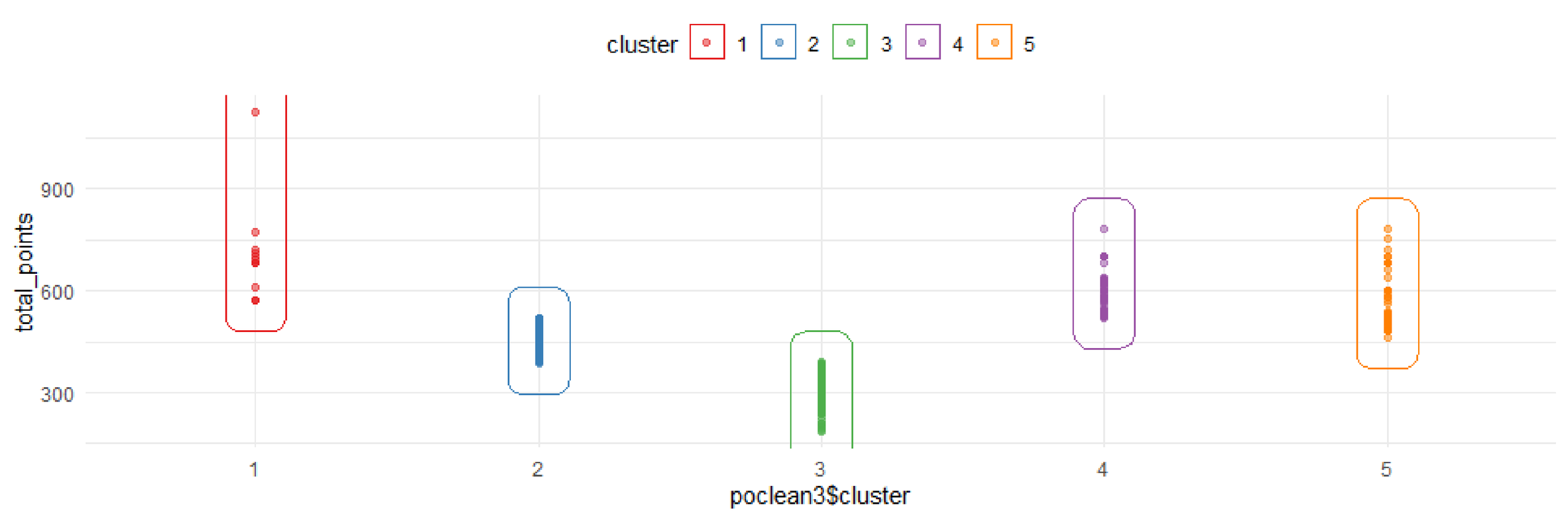

Cluster plot
Dim2 (14.9%)
Dim1 (50.4%)
cluster
1
2
3
4
5
6
4
2
0
-2
-4
-5
0
5
10
15
501
542
129
124
492
541
494
232
72
226
84
439
234
497
437
96
491
180

3D 看分群

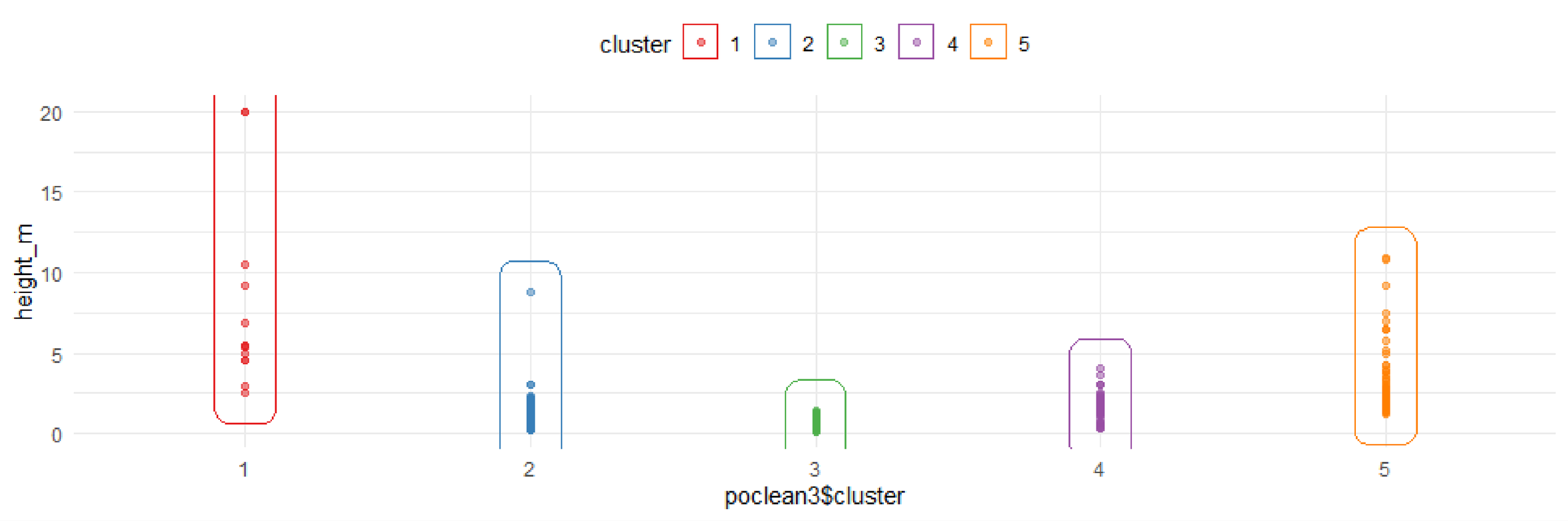




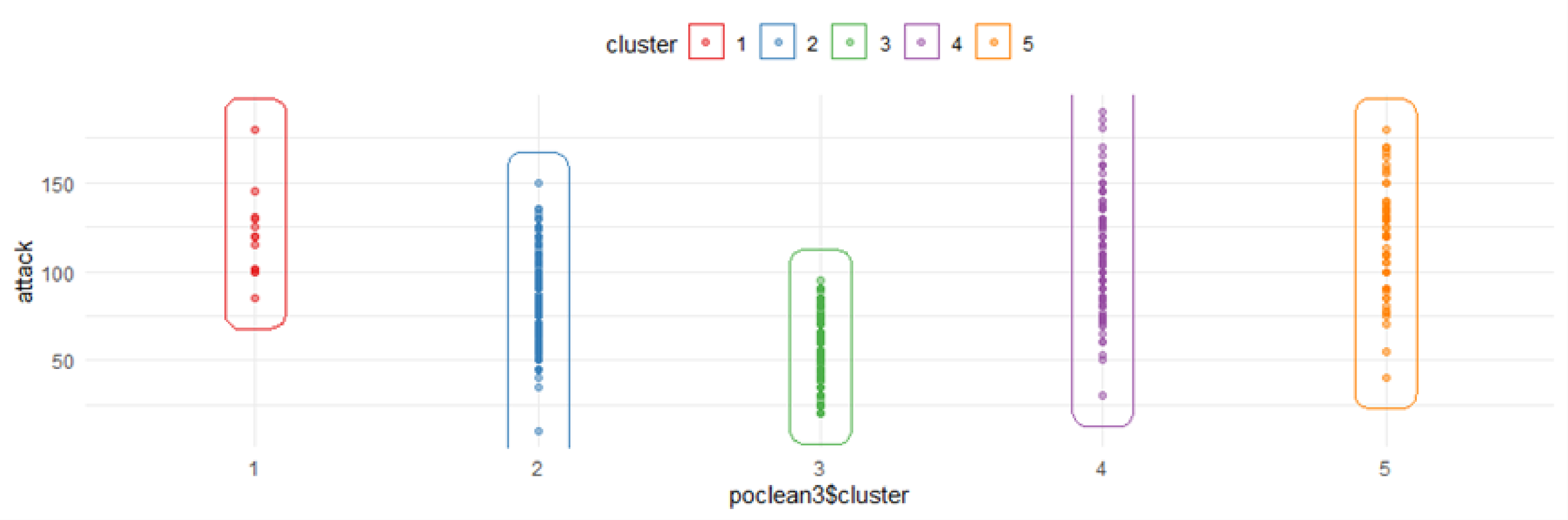
5群和總能力值的關係


cluster 1 2 3 4 5
total_points
900
600
300
1
2
3
4
5
poclean3$cluster

5群和身高的關係

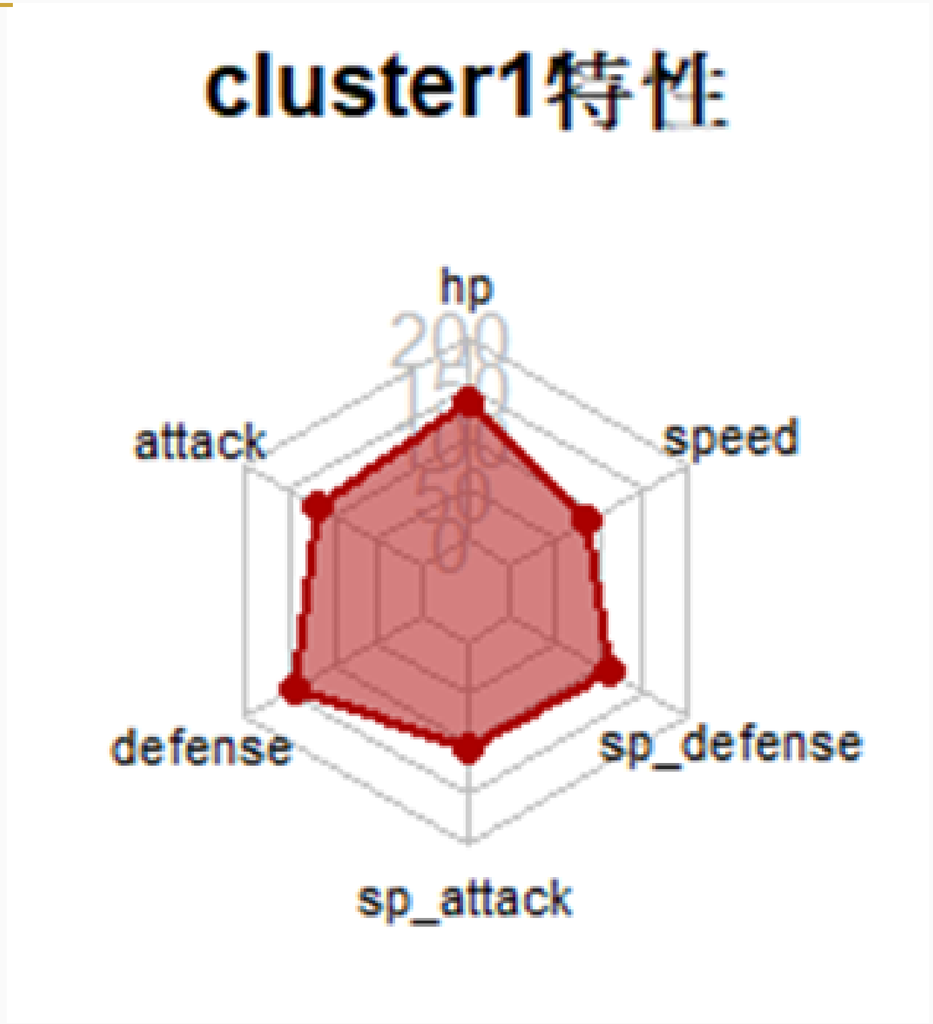


5群和攻擊力的關係



第1群雷達圖

血量	133.76923
攻擊	119.46154
防禦	142.92308
特攻	106
特防	111.92308
速度	83.84615



第２群雷達圖

血量	73
攻擊	83.19298
防禦	82.48246
特攻	76.09649
特防	79.44737
速度	73.32456

第３群雷達圖

血量	49.47651
攻擊	54.24161
防禦	54.04027
特攻	47.43624
特防	50.32886
速度	50.12081

第4群雷達圖

血量	82.93684
攻擊	109.75789
防禦	91.72632
特攻	112.13684
特防	92.52632
速度	95.65263

第５群雷達圖

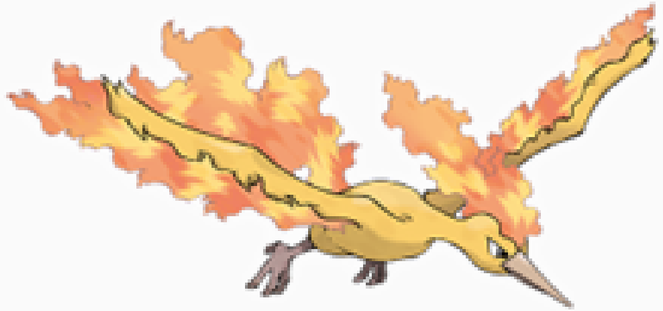
血量	93.40351
攻擊	117.21053
防禦	107.91228
特攻	99.87719
特防	95.24561
速度	72.75439

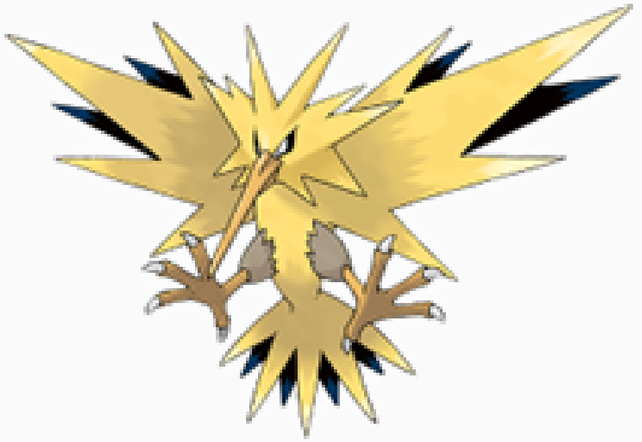
第1群代表

第２群代表

第３群代表

第４群代表





第５群代表

VI

資料來源

- Complete Pokemon Dataset

  https://www.kaggle.com/datasets/mariotormo/complete-pokemon-dataset-updated-090420

Thank you for listening.