Multivariate Analysis Exam 2

107508006 歐西四 陳葳芃

Question 1

請以k-means替oliver資料做分群並將結果視覺化，並描述分析結果

Sol.:

首先載入資料並觀察資料型態，共計*572*筆樣本資料、*10*個變數且無缺失值的出現，而*Region*與*Area* 應為類別變數，再進行適當調整後，可以觀察出*Region* 分為*1*、*2*、*3*共計*3*個類別；*Area*分為*1*、*2*、*3*.....*9* 共計*9*個類別。

```
> str(olive)
'data.frame':   572 obs. of  10 variables:
 $ Region     : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Area       : int  1 1 1 1 1 1 1 1 1 1 ...
 $ Palmitic   : int  1075 1088 911 966 1051 911 922 1100 1082 1037 ...
 $ Palmitoleic: int  75 73 54 57 67 49 66 61 60 55 ...
 $ Stearic    : int  226 224 246 240 259 268 264 235 239 213 ...
 $ Oleic      : int  7823 7709 8113 7952 7771 7924 7990 7728 7745 7944 ...
 $ Linoleic   : int  672 781 549 619 672 678 618 734 709 633 ...
 $ Linolenic  : int  36 31 31 50 50 51 49 39 46 26 ...
 $ Arachidic  : int  60 61 63 78 80 70 56 64 83 52 ...
 $ Eicosenoic : int  29 29 29 35 46 44 29 35 33 30 ...
```

Figure.1 使用*R*觀察*olive*原始資料型態

欲進行*k-means*替資料進行分群分析，我預計採用*Elbow method*、 *Silhouette method*、*NbClust*套件（使用歐式距離）、*Gap statistic* 等*4*種方式，來挑選出適當的*k(*意即：分群個數*)*。在此之前，先將類別變數排除，並對所有數值行變數進行標準化使不同單位變數兼具可比性，另將其存成名為*df*的*dataframe*。

```
> head(df)
      Palmitic Palmitoleic    Stearic     Oleic   Linoleic  Linolenic  Arachidic Eicosenoic
[1,] -0.9297061  -0.9733313 -0.0779804 1.2598296 -1.2707124  0.3170625 0.08634028  0.9030934
[2,] -0.8525970  -1.0114306 -0.1324097 0.9789102 -0.8217818 -0.0684812 0.13173241  0.9030934
[3,] -1.9024672  -1.3733742  0.4663123 1.9744494 -1.7773038 -0.0684812 0.22251667  0.9030934
[4,] -1.5762364  -1.3162252  0.3030245 1.5777122 -1.4889997  1.3965850 0.90339865  1.3291301
[5,] -1.0720614  -1.1257286  0.8201026 1.1316909 -1.2707124  1.3965850 0.99418291  2.1101973
[6,] -1.9024672  -1.4686225  1.0650343 1.5087145 -1.2460006  1.4736938 0.54026160  1.9681851
```

Figure.2 數值變數進行標準化觀察

由下方*Figure.3*到*Figure.6*結果可以得知 *Elbow method*建議可分為*5*群、*Silhouette method* 建議可分為*5*群、*NbClust*套件（使用歐式距離）建議一樣可分為*5*群、而*Gap statistic*則給出分到*9*群的建議（與原本*Area* 變數的*level*相同），依據多數建議我想優先採用與原始類別變數不同的分群個數看看會有什麼發現，因此我決定將*k*訂為*5*。

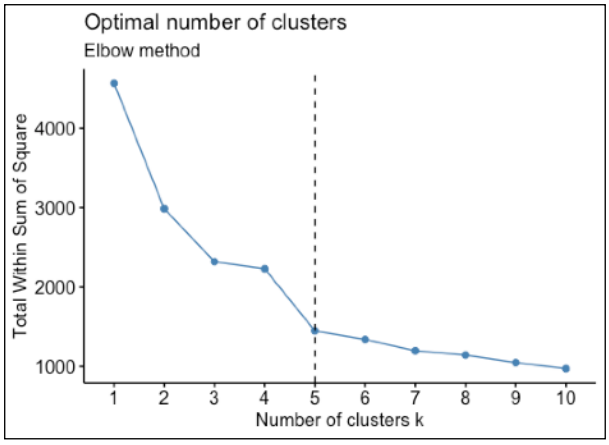


Figure.3 Elbow method

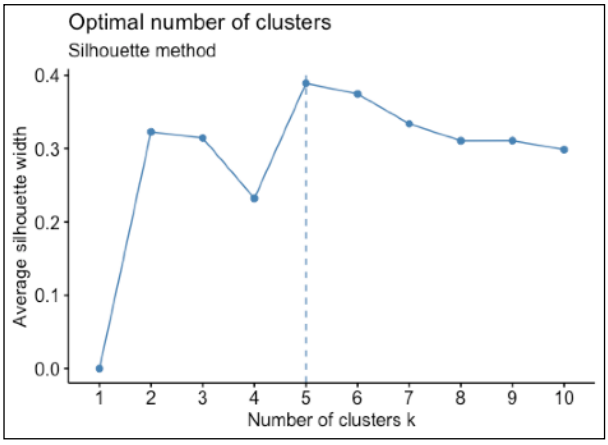


Figure.4 Silhouette method

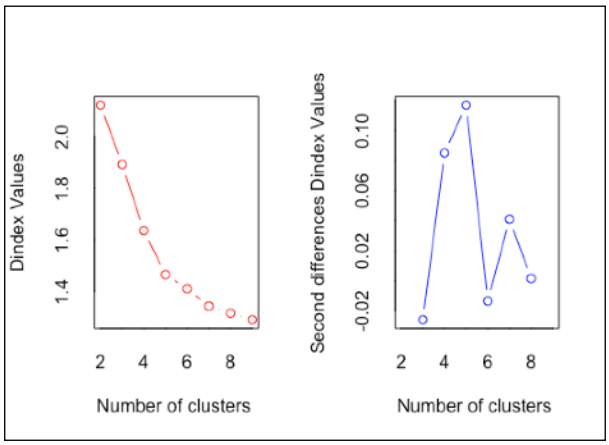


*Figure.5 NbClust*套件（使用歐式距離）

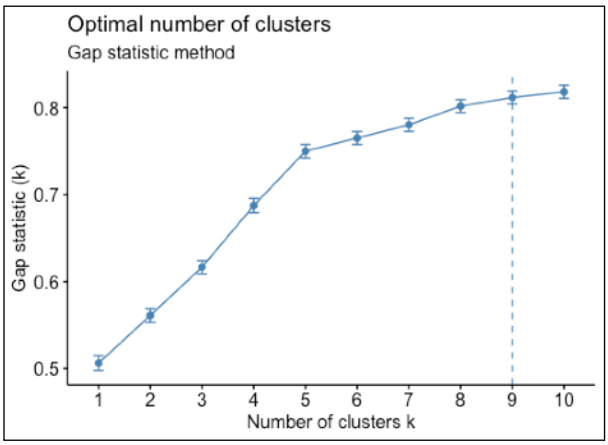


Figure.6 Gap statistic

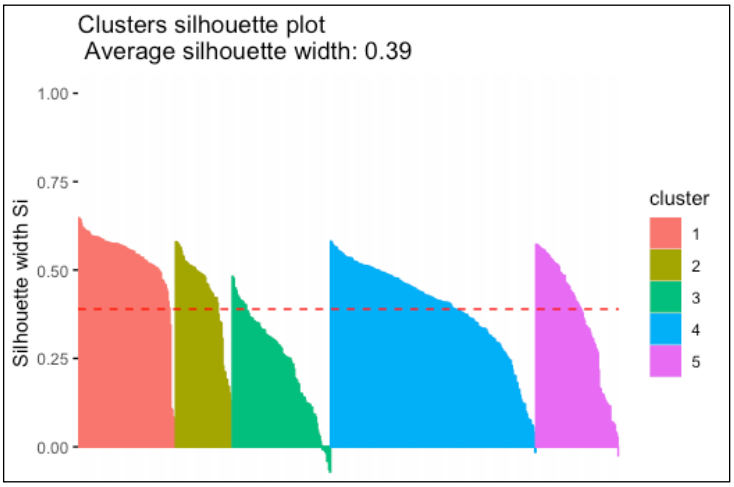


Figure.7 分群結果檢驗

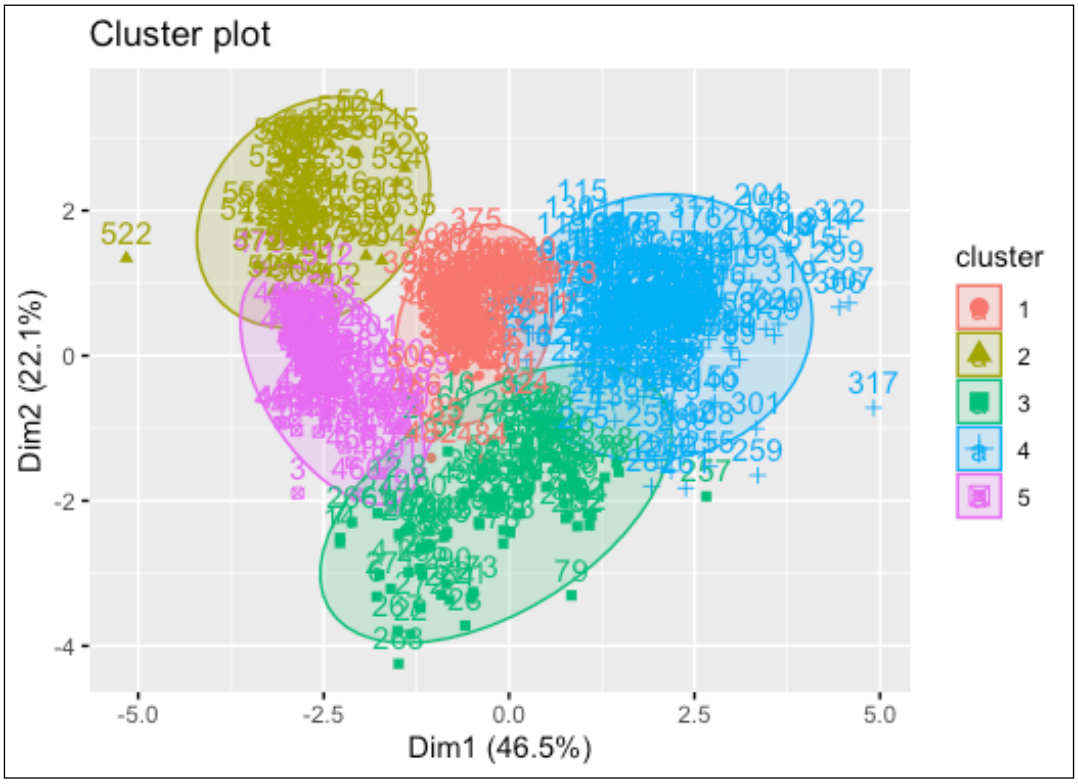


Figure.8 視覺化分群結果

可以由*Figure.7* 看出分*5*群情形大致算良好，只有在*Cluster3*時會有少數錯誤，而 *Figure.8*看出分群的情況界線算明顯，*xy*軸分別為*PCA*當中的*PC1*及*PC2*，詳細組成可以參酌*Figure.9*。

```
> pca_result <- prcomp(df, scale = TRUE)
> pca_result
Standard deviations (1, .., p=8):
[1] 1.92909565 1.32883314 1.00814455 0.89044867 0.57776956 0.49881727 0.34470293
[8] 0.04562633

Rotation (n x k) = (8 x 8):
                  PC1         PC2         PC3         PC4         PC5
Palmitic     0.46074351  0.04958406 -0.11445834 -0.28043124  0.53473943
Palmitoleic  0.45022576  0.24090732 -0.14260264 -0.21182252  0.13841908
Stearic     -0.09864471 -0.25837844 -0.80215910  0.47082168  0.21340068
Oleic       -0.49417494 -0.15866175  0.08011486 -0.20010742 -0.01552215
Linoleic     0.36569539  0.34339930  0.08747773  0.51249093 -0.40127538
Linolenic    0.21898707 -0.60483760  0.19103316 -0.09881321  0.12507081
Arachidic    0.22830362 -0.44719396  0.42664494  0.48165441  0.14659527
Eicosenoic   0.31186781 -0.40476916 -0.30085585 -0.33222211 -0.67153429
                  PC6         PC7         PC8
Palmitic    -0.07699892 -0.52540418 0.35438653
Palmitoleic -0.16728954  0.78680816 0.08856309
Stearic      0.03064009  0.07722664 0.07703841
Oleic       -0.11309403  0.18074878 0.79903372
Linoleic     0.30497855 -0.07768793 0.46687817
Linolenic    0.69784174  0.19096065 0.02943890
Arachidic   -0.55365142  0.06527504 0.03996552
Eicosenoic  -0.25657629 -0.13959613 0.04168750
```

Figure.9 主成分分析成分解釋表

*Cluster*結果分析：
*x*軸越靠近右邊的橄欖富含的*Palmitic*、*Palmitoleic 2*種酸類，其對於保濕、調節不同的代謝過程（例：增加肌肉對胰島素的敏感性、*β*細胞增殖、預防內質網應激以及抑制白色脂肪細胞的脂肪合成活性）具有一定成效；*x*軸越靠近左邊的橄欖富含*Oleic*（油酸），因屬單元不飽和脂肪酸建議重視心血管健康、抗發炎作用、促進皮膚健康、支援吸收營養素可參考此區的橄欖攝取，但仍不建議過量。

另一方面，分析*y*軸靠近上方所含*Linoleic(*亞麻酸較多*)*，對於發炎和免疫調節、心血管健康、營養素運輸 較有所助益；*y*軸靠近下方的橄欖所含的酸類*Linolenic(*亞麻油酸*)* 、*Arachidic(*阿拉酸*)*、*Eicosenoic(*二十碳一烯酸*)*成分較多，同樣對於保濕及心臟和腦部健康有益，而由於亞麻油酸人體不能自行合成，人體可從此群橄欖攝取。

對於選購不同*Cluster*的橄欖提出的相關建議及結論：

Cluster1 橄欖：所含酸類平均，適合對橄欖所含酸類效益沒有特別偏好的一般大眾選擇
Cluster2 橄欖：有心血管保健需求的則可優先考慮參考的橄欖群
Cluster3 橄欖：對亞麻油酸攝取有需求的民眾可考慮的群
Cluster4 橄欖：考慮代謝、抑制脂肪細胞生成
Cluster5 橄欖：若注重健康、身體調理的朋友可以優先參考的橄欖

Question 2

請以svm替oliver做分群並將結果視覺化，並描述分析結果

Sol.:

在SVM部分，我決定將Region列為我的目標變數因此預測變數剩下8個，我也將資料及切為80%的訓練集及20%的測試集，後續仍會將我們的數值型變數進行標準化來進行svm分析。

```
> head(traind)
  Region Palmitic Palmitoleic Stearic Oleic Linoleic Linolenic Arachidic Eicosenoic
1      1     1461         181     197  6783     1246        26        57         23
2      1      916          52     281  7870      694        42        64         58
3      1     1340         114     189  7337      820        48        72         21
4      1     1206         218     242  7193     1002        37        54         25
5      1     1419         159     215  6862     1193        35        60         31
6      1     1109          79     305  7576      763        45        64         36
```

Figure.10 訓練及前六筆資料觀察

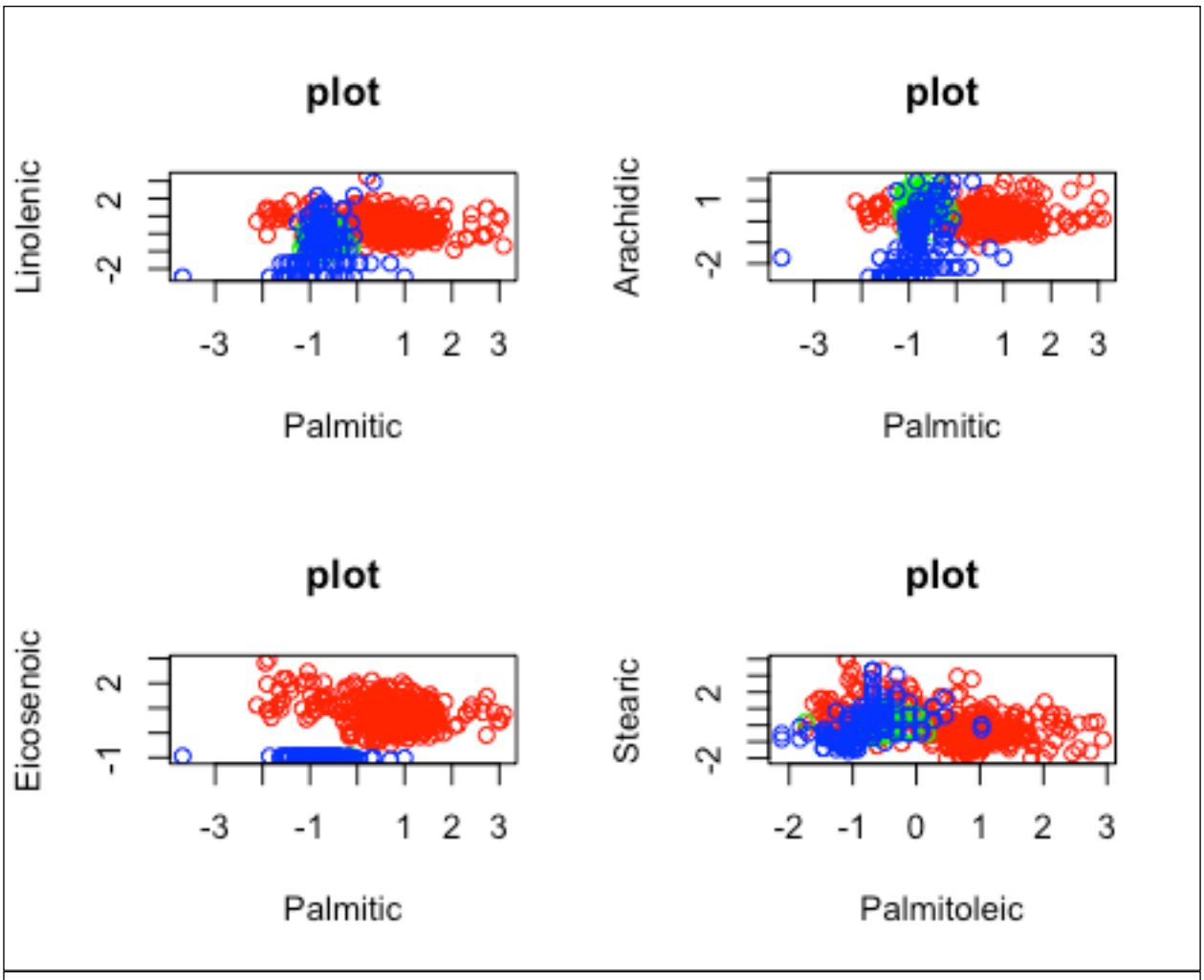
我使用所有變數下去進行svm建模，並使用測試集來確認預測結果，將預測結果呈現在Table.1 可以發現，使用SVM下去預測結果表現相當好，看起來都沒有分錯。

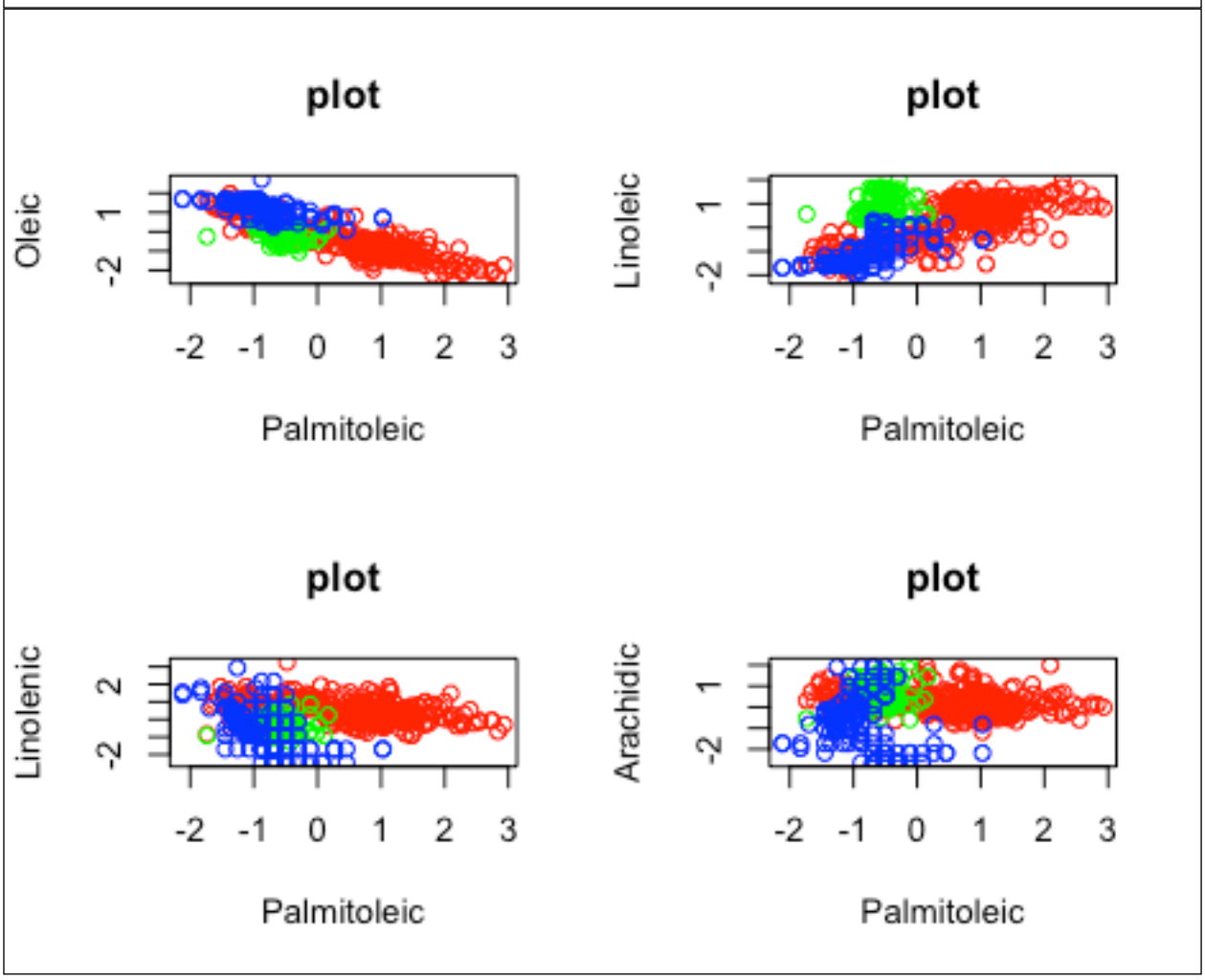
Table.1 SVM預測結果

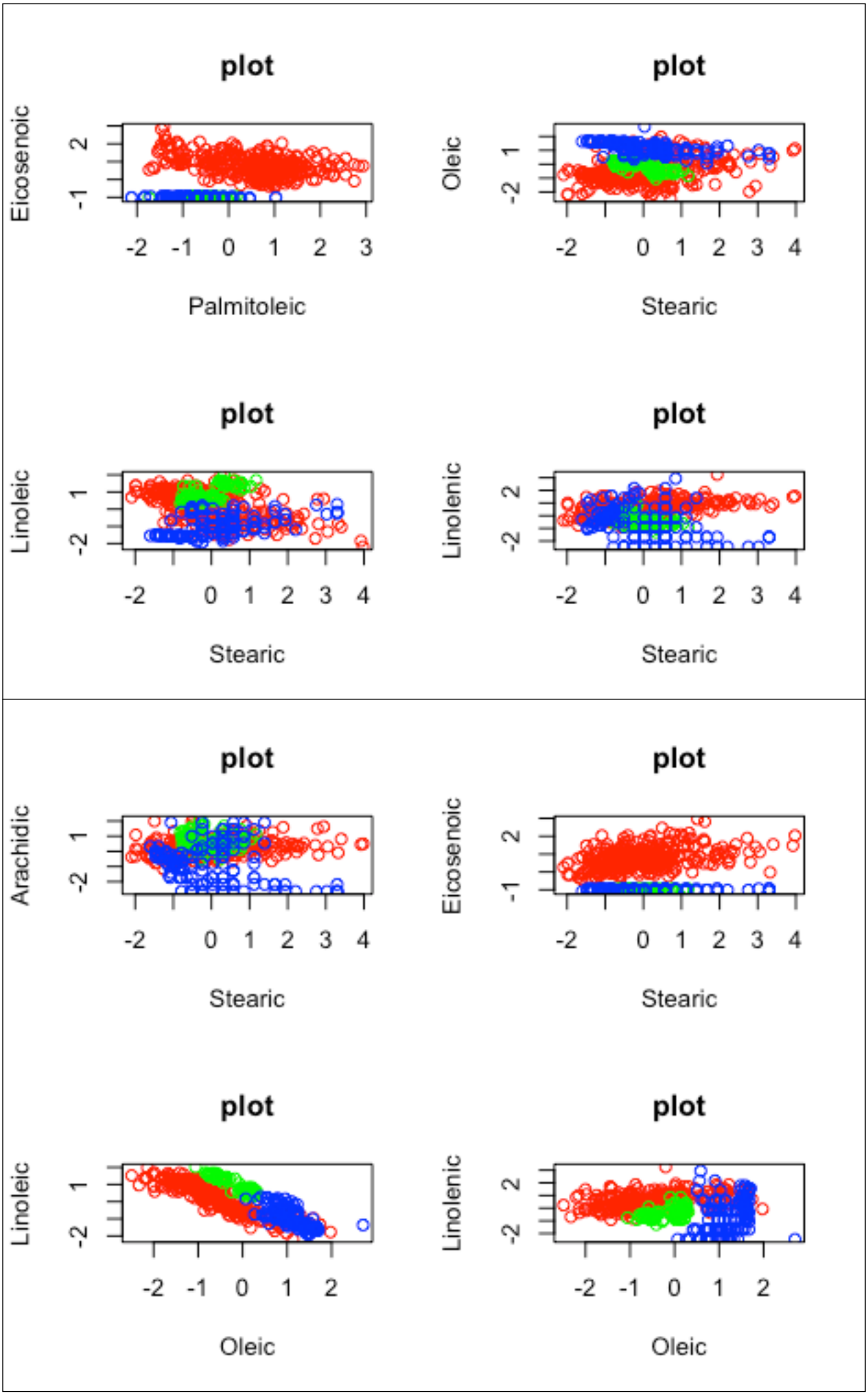
Real/Predict	1	2	3
1	65	0	0
2	0	20	0
3	0	0	30

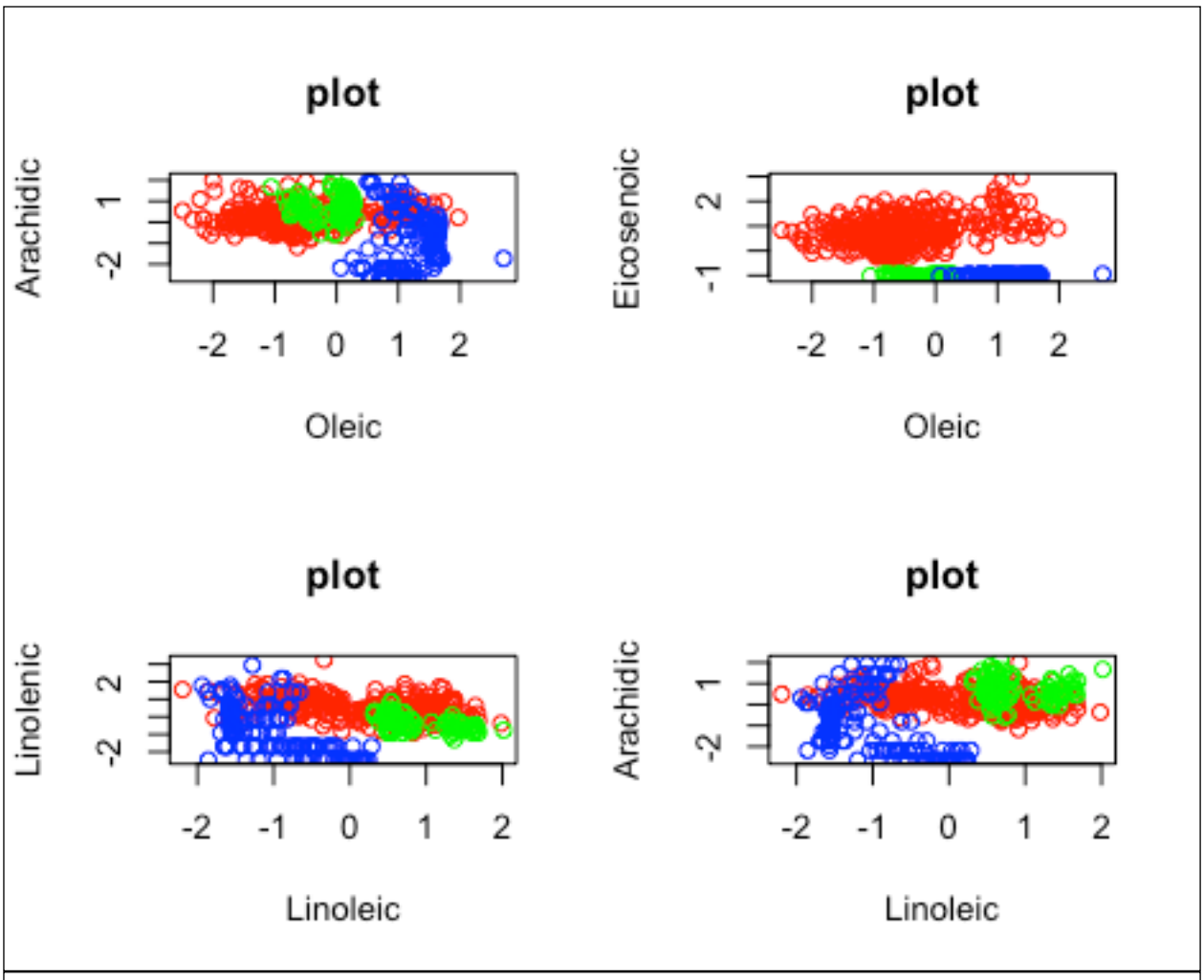
而在可視覺化的的部分，因只能呈現2個維度，故我選擇將原始資料集不同變數之間進行不重複選取的兩兩組合，將28個結果呈現在*Figure.11*，選出可辨識度較高的2個變數來進行最後的decision boundary 呈現。

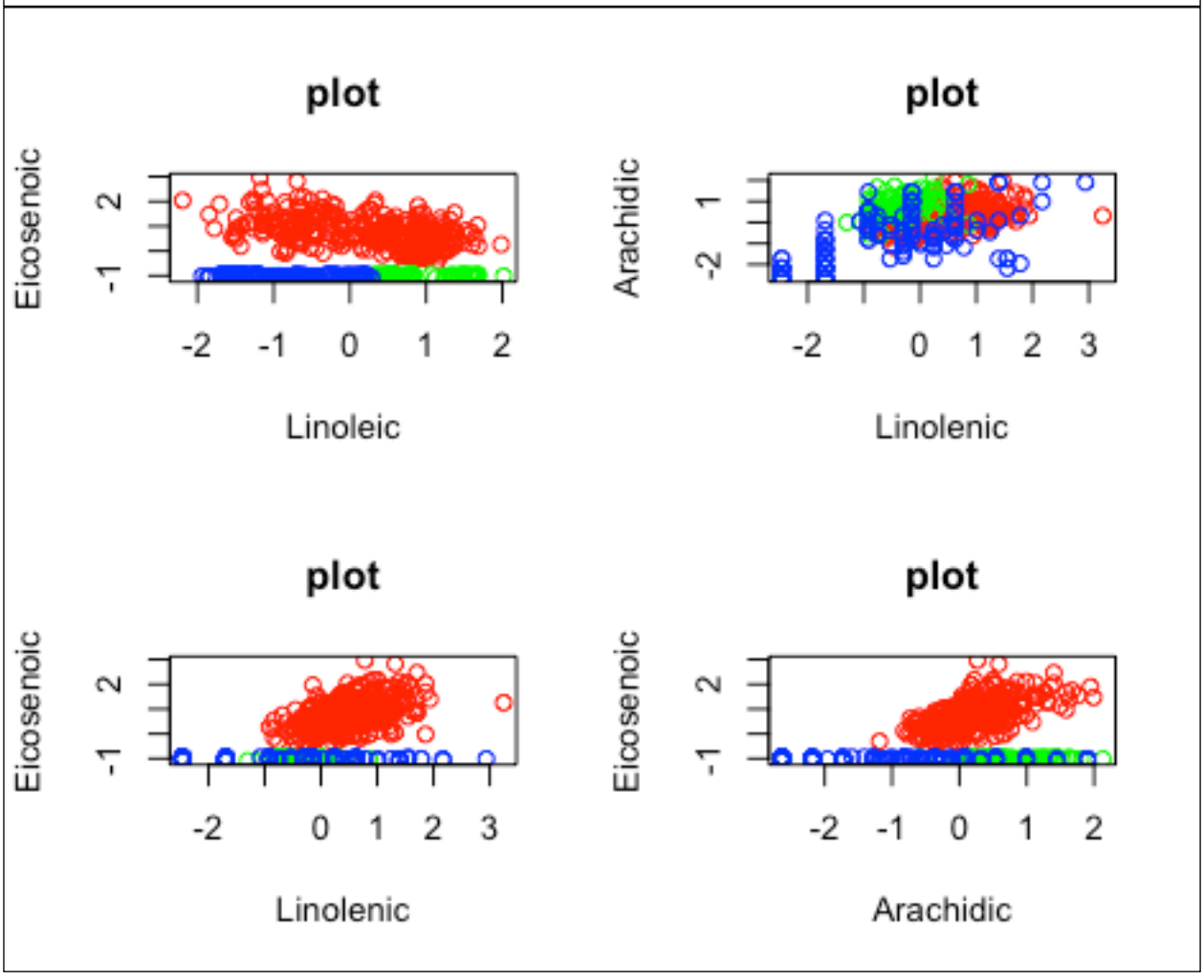
其中，紅色代表Region中的cluster1，綠色代表Region中的cluster2，藍色代表Region中的cluster3。


plot
Linolenic
Palmitic
plot
Arachidic
Palmitic
plot
Eicosenoic
Palmitic
plot
Stearic
Palmitoleic


plot
Oleic
Palmitoleic
plot
Linoleic
Palmitoleic
plot
Linolenic
Palmitoleic
plot
Arachidic
Palmitoleic


plot
Eicosenoic
Palmitoleic
plot
Oleic
Stearic
plot
Linoleic
Stearic
plot
Linolenic
Stearic
plot
Arachidic
Stearic
plot
Eicosenoic
Stearic
plot
Linoleic
Oleic
plot
Linolenic
Oleic


plot
Arachidic
Oleic
plot
Eicosenoic
Oleic
plot
Linolenic
Linoleic
plot
Arachidic
Linoleic


plot
Eicosenoic
Linoleic
plot
Arachidic
Linolenic
plot
Eicosenoic
Linolenic
plot
Eicosenoic
Arachidic

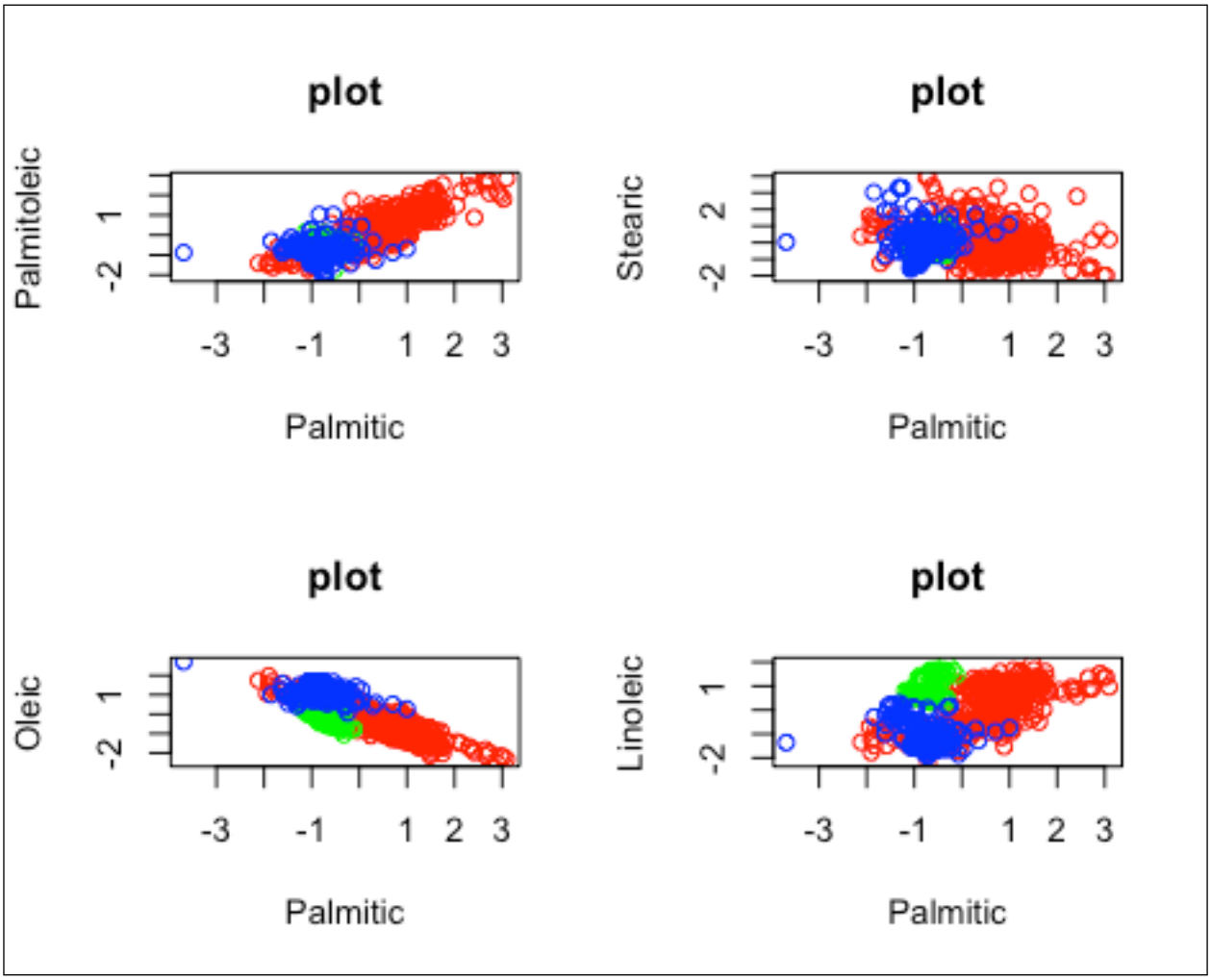


*Figure.11 8*個變數中取*2*個不重複的組合針對*Region*不同群的*scatter plot*

可以從Figure.11 看出Linoleic、Eicosenoic 此兩個變數下，變數的cluster相對清晰，因此決定將配飾視覺化結果的decision buoundary xy軸以此2變數呈現。

*SVM*結果分析及結論：

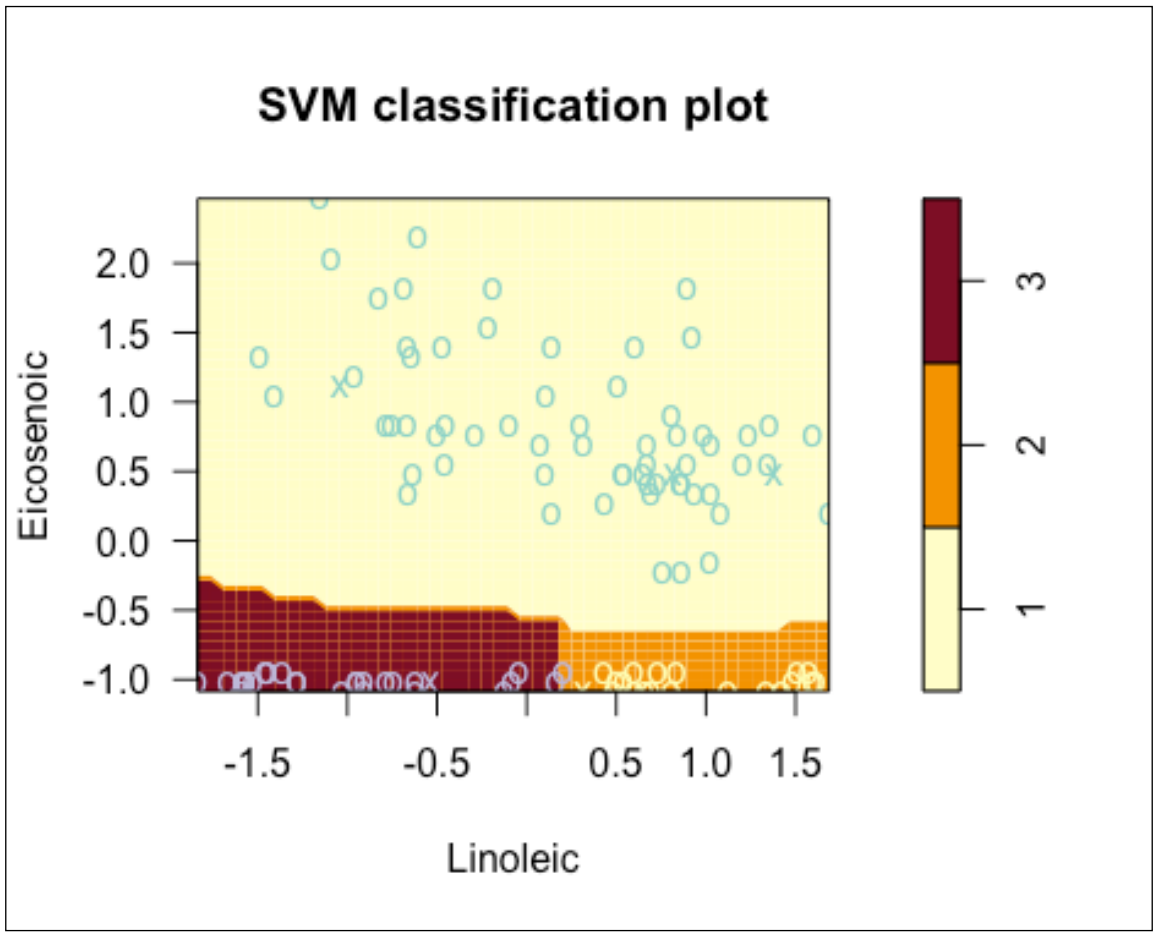


*Figure.12 SVM*支持向量機*decision boundary*

可以由*Figure.12*看到支持向量機的分類結果表現算良好，至少在2個維度以下每個Cluster少有分錯（O、X圖示表示在該Cluster分類正確與否），decision boundary 切在Linoleic 變數標準畫值0.2左右，對於Eicosenoic變數則是切在-0.25~-0.5左右

根據分類結果與可視覺化圖形，我們可以宣稱使用*SVM*支持向量機進行分析，透過橄欖的酸類含量，可以精準分類出橄欖來自不同的區域。

Question 3

Fit the following 3 regression models by using R and evaluate their performance in terms of the accuracy of predicting the response y = "Academic_Reputation" (using a 10-fold Cross Validation):

Model 1: The Least Squares (LS) regression model without the intercept term.
Model 2: The Principal Component Regression (PCR) without the intercept term. For this method, please choose the best number of components based on the model predictability.
Model 3: The Partial Least Squares (PLS) regression without the intercept term. Analogously, please choose the best number of components based on the model predictability.

(1) Are the above 3 prediction models similar, or different?
(2) Which model is best for predicting the college's "Academic Reputation"? Explain why.

[Hint]:
For Q1, running the regression model by using function lm(); then the CV error (prediction error) can be computed by using function cv.lm(), which requires installation of package "lmvar".
For Q2, you need to install package "pls".

Sol.:

(1)
>>model.1 Least Squares (LS) regression model 配飾：

首先，同樣先觀察資料，在針對模型配飾的步驟，移除College_Name進行，並將資料標準化。接著，使用stepwise 方式挑選適合的變數來選出適合變數進行分析。

```
> str(data)
spc_tbl_ [100 × 9] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Rank                  : num [1:100] 1 2 3 4 5 6 7 8 9 10 ...
 $ College_Name          : chr [1:100] "Massachusetts Institute of Technology (MIT)" "Stanford
University" "Harvard University" "University of Oxford" ...
 $ Academic_Reputation   : num [1:100] 100 100 100 100 97.8 98.4 100 99.3 98.6 99.5 ...
 $ Employer_Reputation   : num [1:100] 100 100 100 100 81.2 96.7 100 98.7 99.9 93.7 ...
 $ Faculty_Student       : num [1:100] 100 100 98.7 100 100 85 100 98.1 99.8 96.5 ...
 $ Faculty_Citation      : num [1:100] 99.8 98.6 99.6 84.7 100 98.4 74.2 76.7 72.1 78.5 ...
 $ International_Faculty : num [1:100] 100 99.8 86.3 99.7 99.4 100 100 99.1 100 70.2 ...
 $ International_Students: num [1:100] 94.1 67.7 62.2 98.5 87.3 98 97.6 100 100 81 ...
 $ Overall_Score         : num [1:100] 100 98.4 97.4 97.2 96.9 95.9 95 94.8 94.1 92 ...
```

Figure.13 使用R觀察學校排名原始資料型態

可從*Figure.14*看出*Rank* 對於，預測學術名聲的顯著性較不佳，故後續分析也會將此變數做移除。而重新一次檢驗後，亦確認所有變數均對目標變數：學術名聲在alpha=0.05的情形下是顯著的。

```
> summary(forward.lm)

Call:
lm(formula = Academic_Reputation ~ Rank + Faculty_Student + Faculty_Citation +
    International_Faculty + Overall_Score + Employer_Reputation +
    International_Students, data = train)

Residuals:
      Min        1Q    Median        3Q       Max
-0.007178 -0.002685 -0.000359  0.003014  0.007745

Coefficients:
                         Estimate Std. Error   t value Pr(>|t|)
(Intercept)           -0.0001014  0.0004639    -0.219   0.8275
Rank                   0.0055715  0.0031892     1.747   0.0849 .
Faculty_Student       -0.8415510  0.0008027 -1048.457   <2e-16 ***
Faculty_Citation      -0.6981305  0.0007733  -902.738   <2e-16 ***
International_Faculty -0.2318330  0.0006560  -353.413   <2e-16 ***
Overall_Score          1.7152188  0.0035736   479.975   <2e-16 ***
Employer_Reputation   -0.3003340  0.0007838  -383.156   <2e-16 ***
International_Students -0.2227101  0.0006720  -331.413   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.004073 on 72 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 5.234e+05 on 7 and 72 DF,  p-value: < 2.2e-16
```

Figure.14 stepwise 配飾linear model出對於每個變數的顯著情形

挑選完合適的變數之後，我們一樣將資料切成80%的訓練集及20%的測試集，並以測試集配飾一個無截距項的*lm*模型。 可獲得以下結果

Call:
lm(formula = Academic_Reputation ~ 0 + ., data = train)

Residuals:
Min 1Q Median 3Q Max
-0.007452 -0.002917 -0.000730 0.003251 0.007202

Coefficients:
Estimate Std. Error t value Pr(>|t|)
*Employer_Reputation -0.3002251 0.0007871 -381.4 <2e-16 ****
*Faculty_Student -0.8410595 0.0007540 -1115.4 <2e-16 ****
*Faculty_Citation -0.6977420 0.0007452 -936.3 <2e-16 ****
*International_Faculty -0.2318093 0.0006606 -350.9 <2e-16 ****
*International_Students -0.2225058 0.0006669 -333.7 <2e-16 ****
*Overall_Score 1.7092830 0.0011701 1460.8 <2e-16 ****
---
*Signif. codes: 0 ‘***’0.001 ‘**’0.01 ‘*’0.05 ‘.’0.1 ‘ ’1*

Residual standard error: 0.004104 on 74 degrees of freedom
Multiple R-squared: 1, Adjusted R-squared: 1
F-statistic: 6.152e+05 on 6 and 74 DF, p-value: < 2.2e-16

另再將測試及放入進行預測，經由計算可得其衡量預測效果的指標*RMSE(*衡量實際觀測值和預測值之間的平均差異*)*為*0.004498708*。 CV error 為*0.00004549712*。

而根據上述模型結果，可得知 *adj R square* 值（模型的解釋能力）為*1*。

>>model.2 Principal Component Regression (PCR) 配飾：

而根據PCR的結果，我們應該選擇CV error 值最小的，亦即當Component為6時。

summary(model2)
Data: X dimension: 80 6
Y dimension: 80 1
Fit method: svdpc
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 10 random segments.
(Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps
CV 0.8828 0.7542 0.6810 0.6631 0.6146 0.6301 0.004310
adjCV 0.8828 0.7527 0.6793 0.6607 0.6126 0.6277 0.004291

TRAINING: % variance explained
1 comps 2 comps 3 comps 4 comps 5 comps 6 comps
X 38.71 63.64 81.47 93.71 98.62 100
Academic_Reputation 27.97 41.86 48.10 56.68 56.77 100

Number of Components: 1 RMSE: 1.238701
Number of Components: 2 RMSE: 1.165388
Number of Components: 3 RMSE: 1.129138
Number of Components: 4 RMSE: 1.081523 *R square* 值（模型的解釋能力）為*0.5668*
Number of Components: 5 RMSE: 1.051318 *R square* 值（模型的解釋能力）為*0.5677*
Number of Components: 6 RMSE: 0.9597193 , *R square* 值（模型的解釋能力）為*1*

但透過*R-square*判斷的話，component是3時就能解釋80%的變異，故3或4可能也是另一個可以考慮的好的選擇。

>>model.3 Partial Least Squares (PLS) 配飾：

而根據PCR的結果，我們應該選擇CV error 值最小的，亦即當Component為6時。

summary(model3)
Data: X dimension: 80 6
Y dimension: 80 1
Fit method: kernelpls
Number of components considered: 6

VALIDATION: RMSEP
Cross-validated using 10 random segments.

(Intercept) 1 comps 2 comps 3 comps 4 comps 5 comps 6 comps
CV 0.8828 0.6325 0.5532 0.4305 0.1575 0.03878 0.004408
adjCV 0.8828 0.6305 0.5505 0.4304 0.1566 0.02609 0.004384

TRAINING: % variance explained

	1 comps	*2 comps*	*3 comps*	*4 comps*	*5 comps*	*6 comps*
X	*35.05*	*57.43*	*72.32*	*80.65*	*95.05*	*100*
Academic_Reputation	*53.95*	*66.22*	*79.91*	*97.62*	*99.94*	*100*

但透過*R-square*判斷的話，component是4時就能解釋80%的變異，故4可能也是另一個可以考慮的好的選擇。

結論，我認爲3個模型在Cross validation時，線性模型表現得最佳，而與其不一樣的則是PCR 與 PLS模型，兩者較為相似都到component為6時才有比較小的CV error。另一方面，我認為PLS應更劣於PCR 原因在於PCR 可以透過3個Component即有解釋超過80%變異的好表現。

(2)根據CV-error判斷我認為 model.1 Least Squares (LS) regression model具有較佳的預測效果，雖然除了rank的變數全放了，但在CV error 與 R -square的表現上均優於其他兩者。

Appendix: R code

```
##### Multivariate Analysis Exam 2 #####
### Student ID: 107508006
### Department: 歐西四
### Name      : 陳葳芃

# Question 1 k means ####

olive <- read.table("~/Desktop/1112-NCCU/多變量分析/Code and TA/小考/olive.txt",
header=TRUE, quote="\"")
which(is.na(olive))
str(olive)
summary(olive)

olive$Region<-as.factor(olive$Region)
olive$Area<-as.factor(olive$Area)

library(factoextra)
library(NbClust)
# Elbow method 6 根據每個資料點的分散以及聚合來衡量分群的結果
df<-olive[,-c(1,2)]
df<-scale(df)

#3
fviz_nbclust(df, kmeans, method = "wss") +
  geom_vline(xintercept = 5, linetype = 2) + # add line for better visualisation
  labs(subtitle = "Elbow method") # add subtitle


#5
fviz_nbclust(df, kmeans, method = "silhouette") +
  labs(subtitle = "Silhouette method")

#3
library(NbClust)
NbClust(data = df, distance = "euclidean", min.nc = 2, max.nc = 9, method =
"kmeans", index = "all", alphaBeale = 0.1)

# Gap statistic 9
set.seed(42)
fviz_nbclust(df, kmeans,
             nstart = 25,
             method = "gap_stat",
             nboot = 500
) + # reduce it for lower computation time (but less precise results)
  labs(subtitle = "Gap statistic method")


library(cluster)
set.seed(42)
km_res <- kmeans(df, centers = 5, nstart = 20)
sil <- silhouette(km_res$cluster, dist(df))
fviz_silhouette(sil)
library(factoextra)
fviz_cluster(km_res, df, ellipse.type = "norm")

pca_result <- prcomp(df, scale = TRUE)
```

```
summary(pca_result)
# Question 2 svm ####
library(tidyverse)
olive$index =c(1:nrow(olive))

olive[,c(3:10)]<-scale(olive[,c(3:10)])

train_df <- olive %>% group_by(Region) %>% sample_frac(0.8)
test_df  <- anti_join(olive, train_df, by = 'index')
traind <-as.data.frame(train_df[,-c(2,11)])
testd <-as.data.frame(test_df[,-c(2,11)])
head(traind)

library(e1071)
s <- svm(Region ~ Linoleic+Eicosenoic, data = traind, probability = TRUE)
results <- predict(s, testd, probability = TRUE)
table(Real = testd$Region, Predict = results)

str(olive)
length(which(olive$Region=="1"))
length(which(olive$Region=="2"))
length(which(olive$Region=="3"))

col<-colnames(olive)
col<-col[-c(1,2,11)]
num_combinations <- choose(length(col), 2)
combinations <- combn(col, 2)

olive_color <- c(rep("red", 323), rep("green", 98), rep("blue", 151))
par(mfrow=c(2,2))
for (i in 1:ncol(combinations)) {
  element1 <- combinations[1, i]
  element2 <- combinations[2, i]

  plot(
    data = olive,
    x = olive[[element1]],
    y = olive[[element2]],
    main = "plot",
    xlab = element1,
    ylab = element2,
    col = olive_color
  )
}

# Draw Data and Decision Boundary
#install.packages("RColorBrewer")
library(RColorBrewer)
display.brewer.all()

rcols <- palette(brewer.pal(n = 3, name = "Set3"))
plot(s, traind, Eicosenoic~Linoleic,
     slice = list(Eicosenoic = 100, Linoleic = 4),col = rcols)

plot(s, Eicosenoic~Linoleic, data=testd)

# Question 3 model comparison####

##model1
library(readr)
```

```
data <- read_csv("~/Desktop/1112-NCCU/多變量分析/Code and TA/2020-QS-World-
University-Rankings-100_(1).csv")

str(data)
lm_data<-data[,-c(1,2)]
lm_data<-scale(lm_data)
lm_data<-as.data.frame(lm_data)

# 先把資料區分成 train=0.8, test=0.2
set.seed(22)
train.index <- sample(x=1:nrow(lm_data), size=ceiling(0.8*nrow(lm_data) ))

train = lm_data[train.index, ]
test = lm_data[-train.index, ]

# 1.建立空的線性迴歸(只有截距項)
null = lm(Academic_Reputation ~ 1, data = train)
full = lm(Academic_Reputation~ ., data = train) # 建立上界，也就是完整的線性迴歸

# 2.使用step()，一個一個把變數丟進去
forward.lm = step(null,
                  scope=list(lower=null, upper=full),
                  direction="forward")
summary(forward.lm)

model1 <- lm(Academic_Reputation ~ 0 + ., data = train)

library(DAAG)
cv_error <- cv.lm(data=test, model, m = 10)
# View the CV error
print(cv_error)

predictions <- predict(model1, test)
errors <- predictions - test$Academic_Reputation
rmse <- sqrt(mean(errors^2))
rmse

summary(model1)

##model2
# Load the necessary library
library(pls)

# Set the maximum number of components to consider
max_components <- 6

# Create an empty vector to store the RMSE values for each number of components
rmse_values <- numeric(max_components)

# Perform PCR with different numbers of components
for (i in 1:max_components) {
  model2 <- pcr(Academic_Reputation ~ . - 1, data = train, scale = TRUE, ncomp =
i, validation = "CV")
  predicted <- predict(model2, newdata = test)
  rmse_values[i] <- sqrt(mean((predicted - test$Academic_Reputation)^2))
}


# View the RMSE values for different numbers of components
```

```
for (i in 1:max_components) {
  cat("Number of Components:", i, "RMSE:", rmse_values[i], "\n")
}


summary(model2)
validationplot(model2, val.type="RMSE")
validationplot(model2, val.type="R2")


##model3

library(pls)

# 建立PLS模型
model <- plsr(Academic_Reputation ~ . - 1, data = train, scale = TRUE,
validation = "CV")

for (i in 1:max_components) {
  model3 <- plsr(Academic_Reputation ~ . , data = train, scale = TRUE, ncomp =
i, validation = "CV", intercept = FALSE)
  predicted <- predict(model3, newdata = test)
  rmse_values[i] <- sqrt(mean((predicted - test$Academic_Reputation)^2))
}

for (i in 1:max_components) {
  cat("Number of Components:", i, "RMSE:", rmse_values[i], "\n")
}

validationplot(model3, val.type="RMSE")
validationplot(model3, val.type="R2")

summary(model3)
```