Progetto dataset oliveoil - Gruppo T

2024-06-24

Abbiamo analizzato il dataset oliveoil contenuto nel pacchetto pdfCluster. Contenete dati relativi alla composizione chimica di diversi oli d'oliva provenienti da diverse macro aree e regioni Italiane.

Librerie necessarie

```
library(cluster)
library(ggplot2)
library(ggcorrplot)
library(pdfCluster)

## pdfCluster 1.0-4

library(dbscan)

##
## Attaching package: 'dbscan'

## The following object is masked from 'package:stats':
##
##     as.dendrogram

library(moments)
library(compositions)

## Welcome to compositions, a package for compositional data analysis.
## Find an intro with "? compositions"

##
## Attaching package: 'compositions'

## The following objects are masked from 'package:stats':
##
##     anova, cor, cov, dist, var

## The following object is masked from 'package:graphics':
##
##     segments

## The following objects are masked from 'package:base':
##
##     %*%, norm, scale, scale.default

library(reshape2)
```

Funzioni usate nel progetto

```
# Funzione per le Variabili Quantitative
display_summary_and_var <- function(variabile){
  c(summary(variabile),
  var = var(variabile, na.rm = T),
  sd = sd(variabile, na.rm = T),
  sk = skewness(variabile, na.rm = T))
}

# Funzione per le Variabili Qualitative
display_table <- function(variabile, titolo){
  DistAs <- table(variabile)
  DistRe <- prop.table(table(variabile))
  barplot(prop.table(table(variabile)), main = titolo)
  print(rbind(DistAs, DistRe))
}
```

Import del dataset e pulizia dei dati

Si decide di normalizzare i dati nel seguente modo:

$$y_{ij} = \frac{x_{ij} + 1}{\sum_{j=3}^{10}(x_{ij} + 1)} \quad , \forall j \text{ colonna}, \forall i \text{ riga}$$

Questo perché i dati sono di natura compositiva, infatti ogni riga somma circa a 10000 e quindi possono essere visti come la percentuale di un particolare acido nell'olio Dalla formula si nota che ad ogni osservazione è stato sommato 1 perché nei dati originali ci sono degli zeri dovuti alle misurazioni al di sotto del livello di sensibilità degli strumenti con la quale è stata effettuata l'analisi

```
library(pdfCluster)
data("oliveoil")
str(oliveoil)
```

```
## 'data.frame':    572 obs. of  10 variables:
##  $ macro.area : Factor w/ 3 levels "South","Sardinia",..: 1 1 1 1 1 1 1 1
1 1 ...
##  $ region     : Factor w/ 9 levels "Apulia.north",..: 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 ...
```

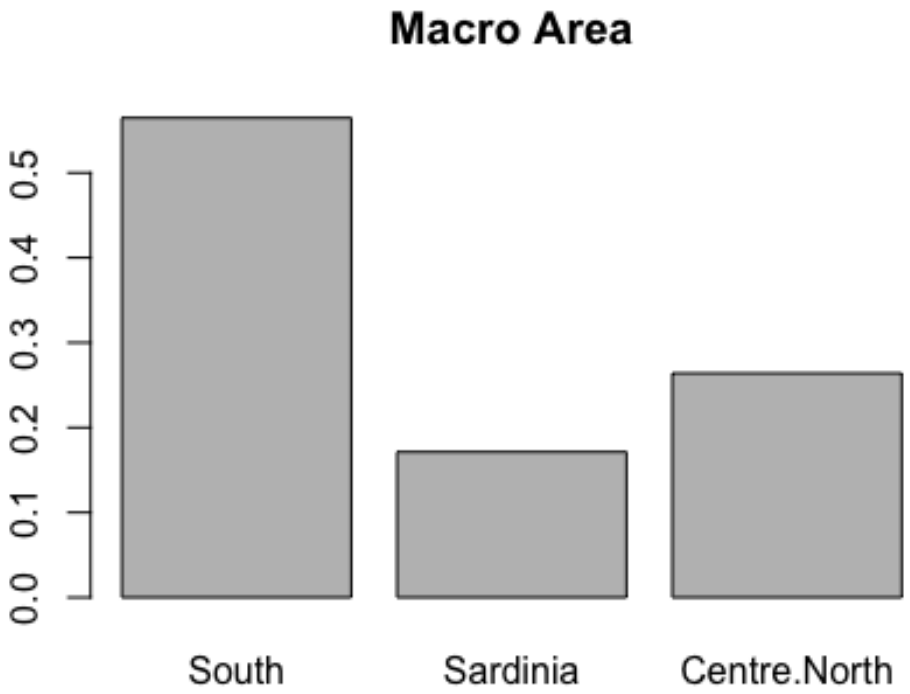
```
oliveoil[,3:10] <- oliveoil[,3:10]+1
for (i in 1:nrow(oliveoil)){
```

```
  oliveoil[i,3:10] <- oliveoil[i,3:10]/sum(oliveoil[i,3:10])
}
```

Anilisi univariata e bivariata del dataset

Variabile Macro Area

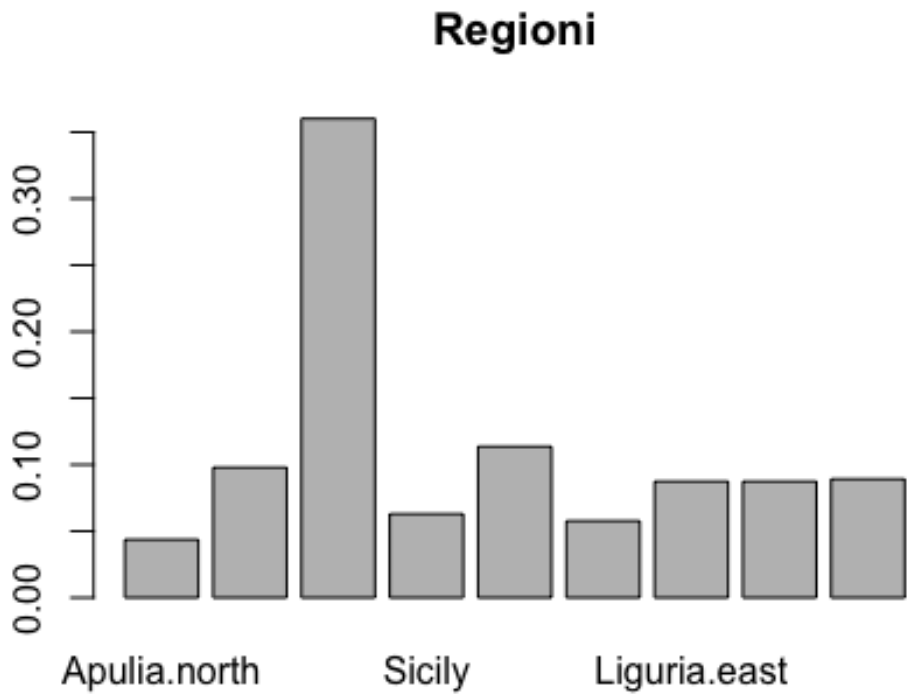
```
display_table(oliveoil$macro.area, "Macro Area")
```




```
##              South   Sardinia Centre.North
## DistAs 323.0000000 98.0000000   151.000000
## DistRe   0.5646853  0.1713287     0.263986
```

Variabile Regioni

```
display_table(oliveoil$region, "Regioni")
```




```
##         Apulia.north   Calabria Apulia.south      Sicily Sardinia.inland
## DistAs  25.00000000 56.0000000  206.0000000 36.00000000      65.0000000
## DistRe   0.04370629  0.0979021    0.3601399  0.06293706       0.1136364
##         Sardinia.coast Liguria.east Liguria.west      Umbria
## DistAs    33.00000000  50.00000000  50.00000000 51.00000000
## DistRe     0.05769231   0.08741259   0.08741259  0.08916084
```

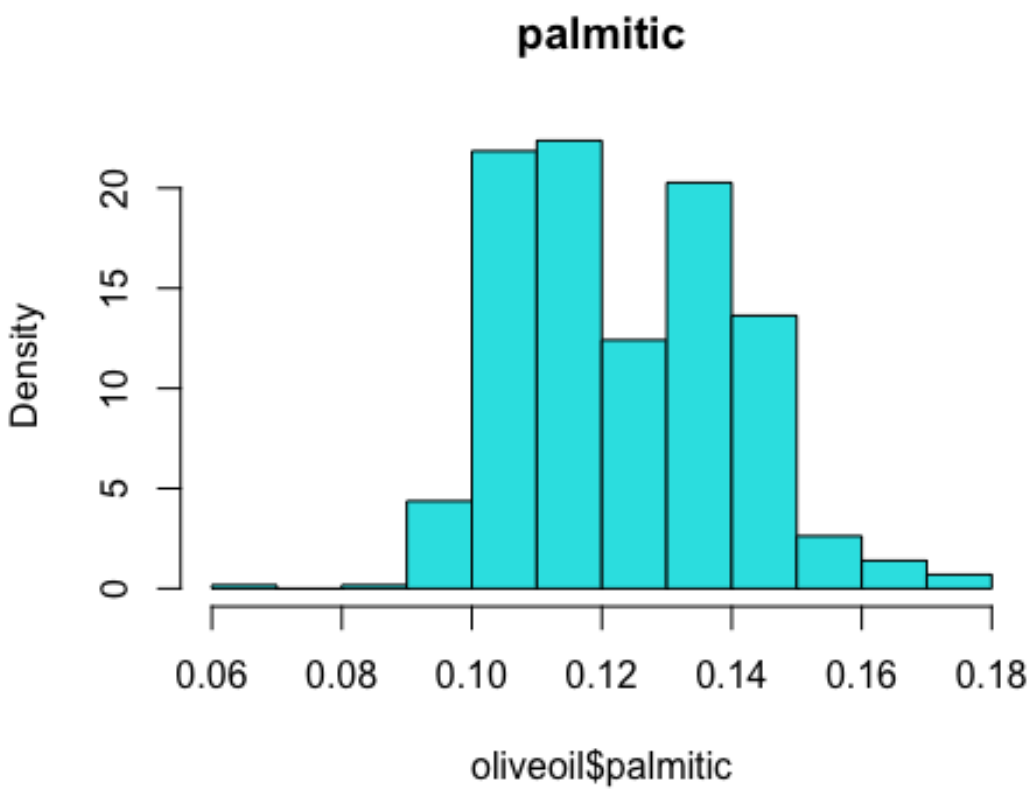
Variabile acido palmitic

È un acido grasso saturo con 16 atomi di carbonio. È uno degli acidi grassi più comuni presenti negli oli vegetali e negli animali. È solido a temperatura ambiente e contribuisce alla consistenza degli oli.

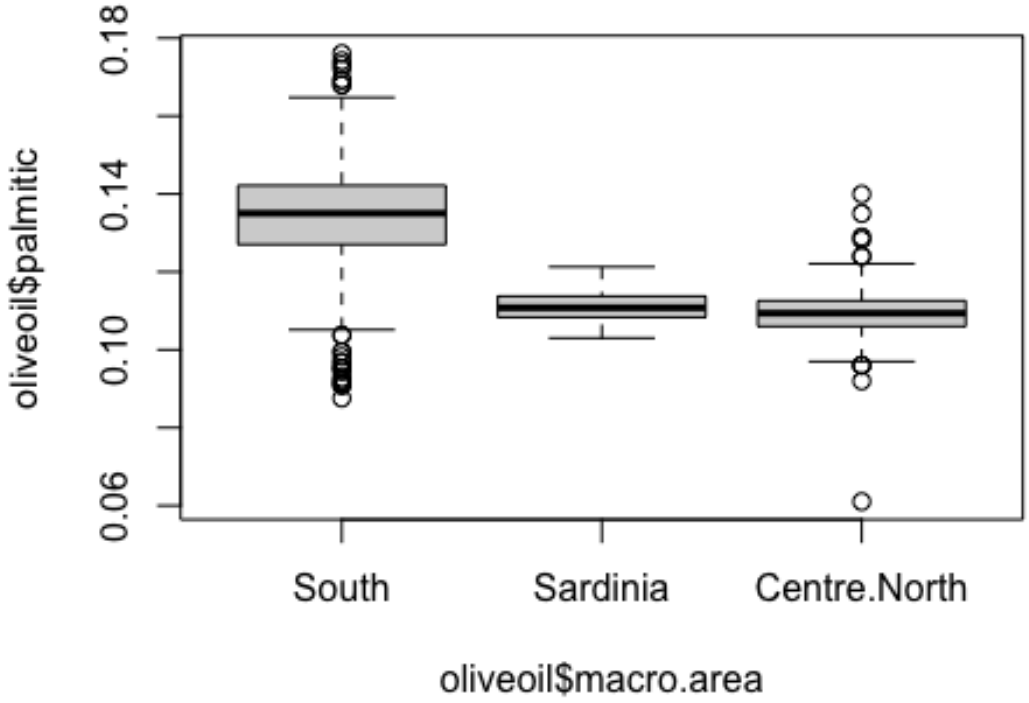
```
display_summary_and_var(oliveoil$palmitic)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.
Max.
## 0.0610328638 0.1095791166 0.1205078869 0.1233676198 0.1364502622
0.1760337214
##          var           sd           sk
## 0.0002868338 0.0169361680 0.3323142153
```

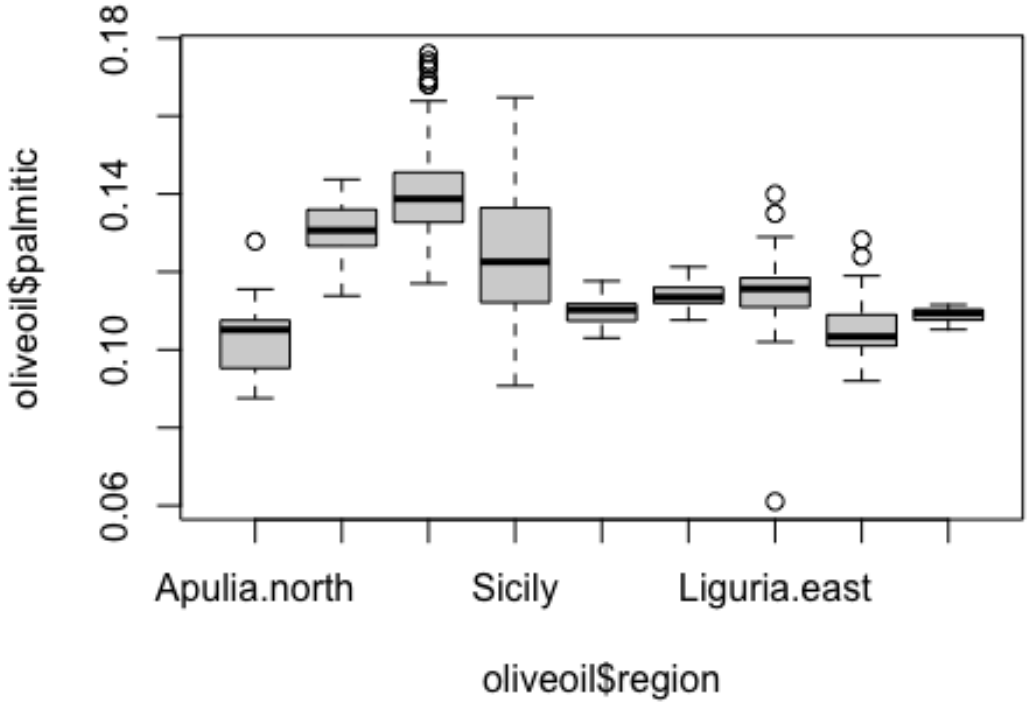
```
hist(oliveoil$palmitic, probability = T,col = 5, main = "palmitic")
```




```
boxplot(oliveoil$palmitic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$palmitic~oliveoil$region)
```



Istogramma: La distribuzione è leggermente asimmetrica a destra con la maggior parte dei valori compresi tra 0.10 e 0.16. Boxplot (Area Macro): La regione Sud ha un valore mediano più alto rispetto a Sardegna e Centro-Nord. Boxplot (Regione): I valori mediani più alti si trovano in Puglia Nord e Sicilia, con una significativa variabilità e outlier, indicando una gamma diversificata di composizioni.

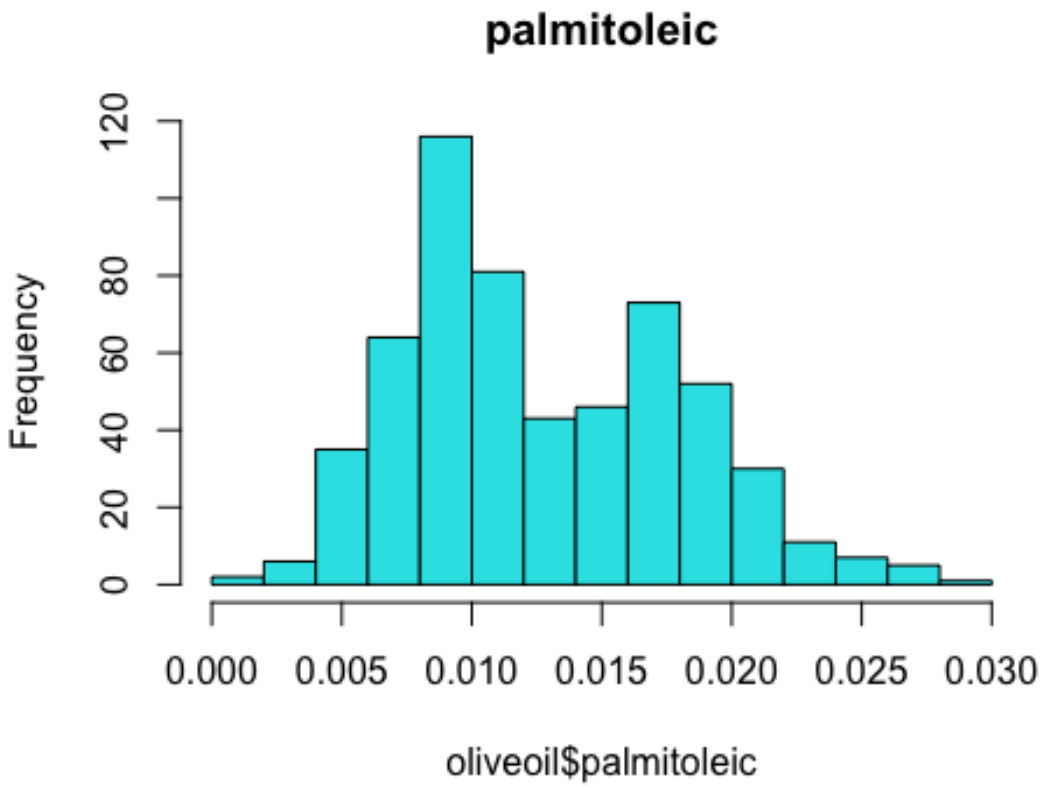
Variabile acido palmitoleic

È un acido grasso monoinsaturo con 16 atomi di carbonio e un doppio legame nella posizione 9. Si trova principalmente negli oli di pesce e in alcune piante. Ha proprietà emollienti e antiossidanti.

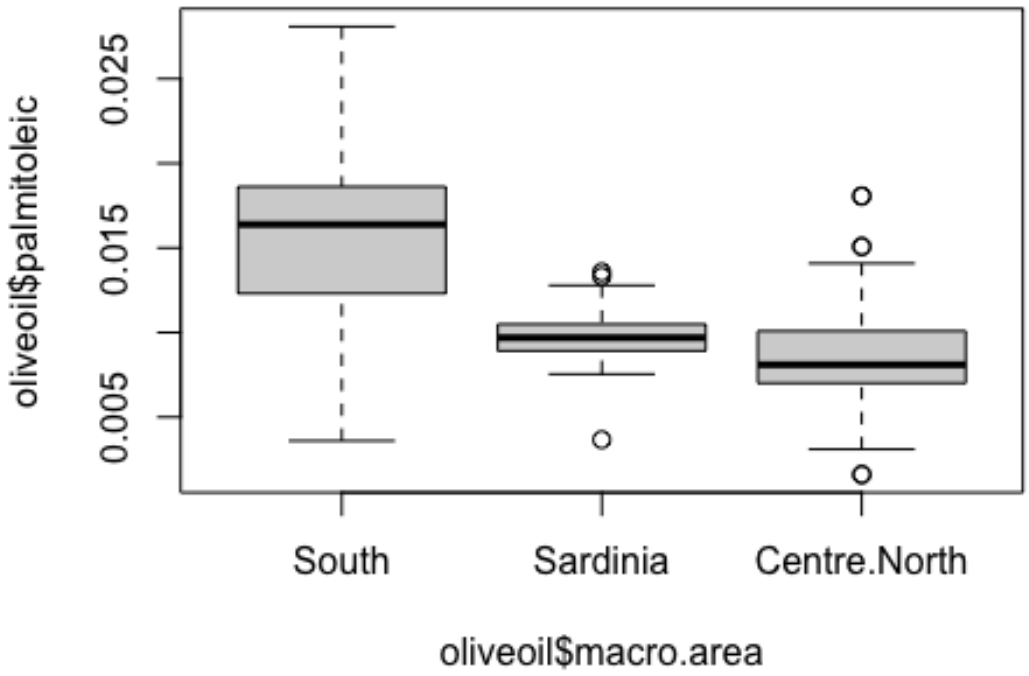
```
display_summary_and_var(oliveoil$palmitoleic)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.
Max.
## 1.603206e-03 8.868299e-03 1.110223e-02 1.271856e-02 1.705098e-02
2.808315e-02
##          var           sd           sk
## 2.760705e-05 5.254241e-03 4.540351e-01
```

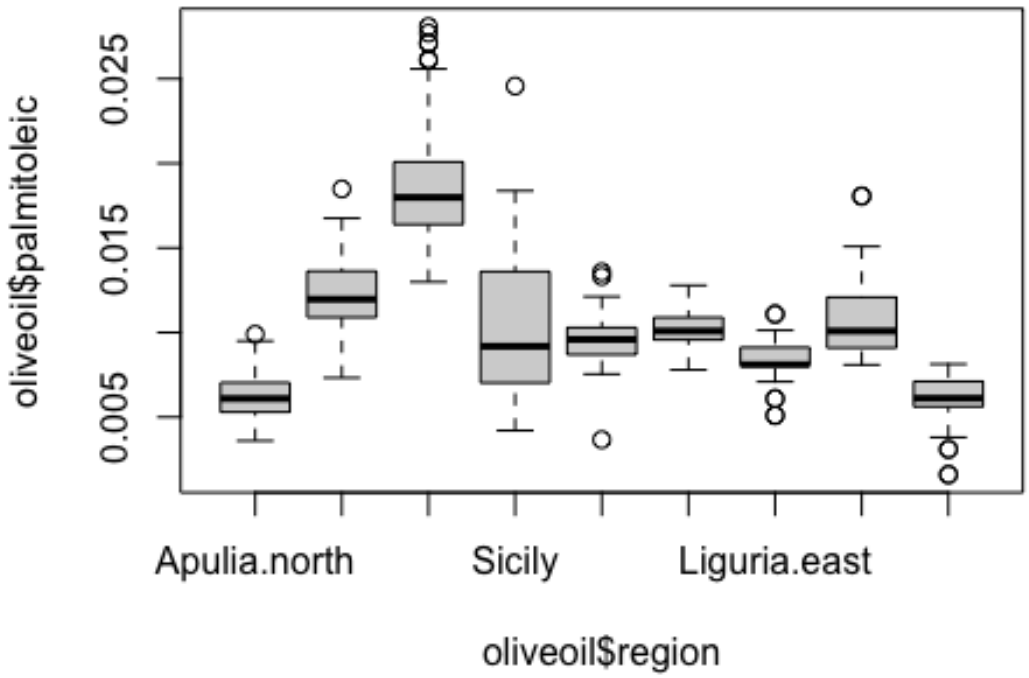
```
hist(oliveoil$palmitoleic,col = 5, main = "palmitoleic")
```




```r
boxplot(oliveoil$palmitoleic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$palmitoleic~oliveoil$region)
```



Istogramma: La distribuzione è leggermente asimmetrica a destra con un picco intorno a 0.015. Boxplot (Area Macro): La regione Sud ha i valori mediani più alti, mentre Sardegna e Centro-Nord hanno valori più bassi. Boxplot (Regione): Puglia Nord e Sicilia mostrano valori mediani più alti, con una significativa variabilità e outlier, indicando una gamma diversificata di composizioni.

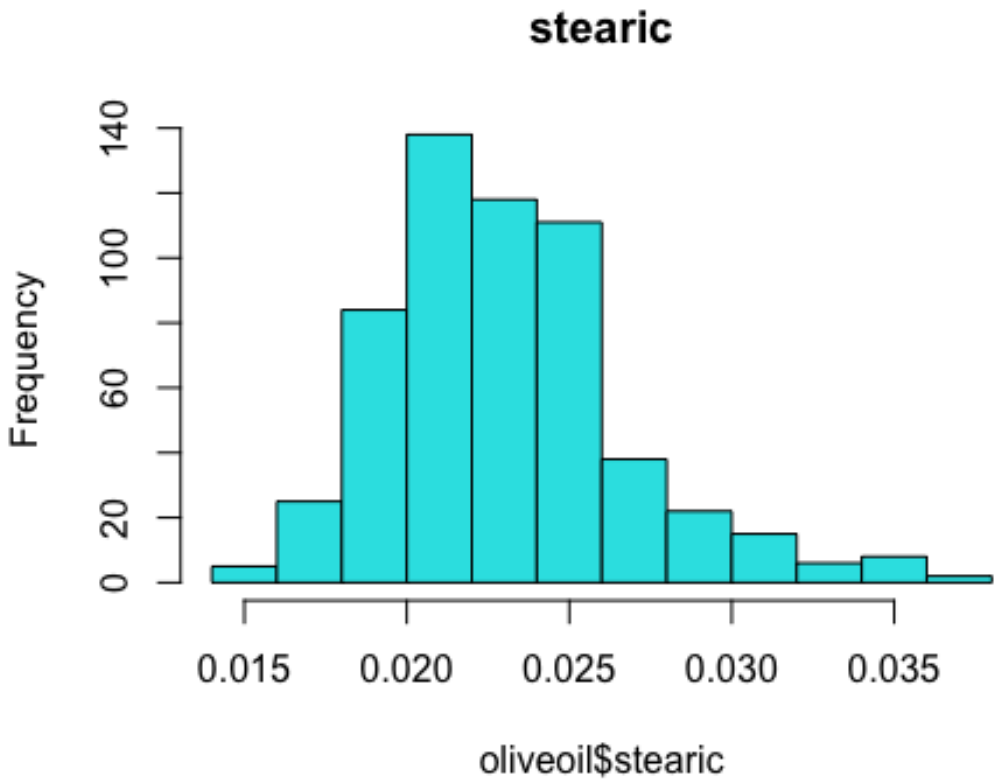
Variabile acido stearic

È un acido grasso saturo con 18 atomi di carbonio. Comunemente presente nel burro di cacao e nel sego, viene utilizzato in cosmetica e nella produzione di candele per la sua consistenza solida a temperatura ambiente.

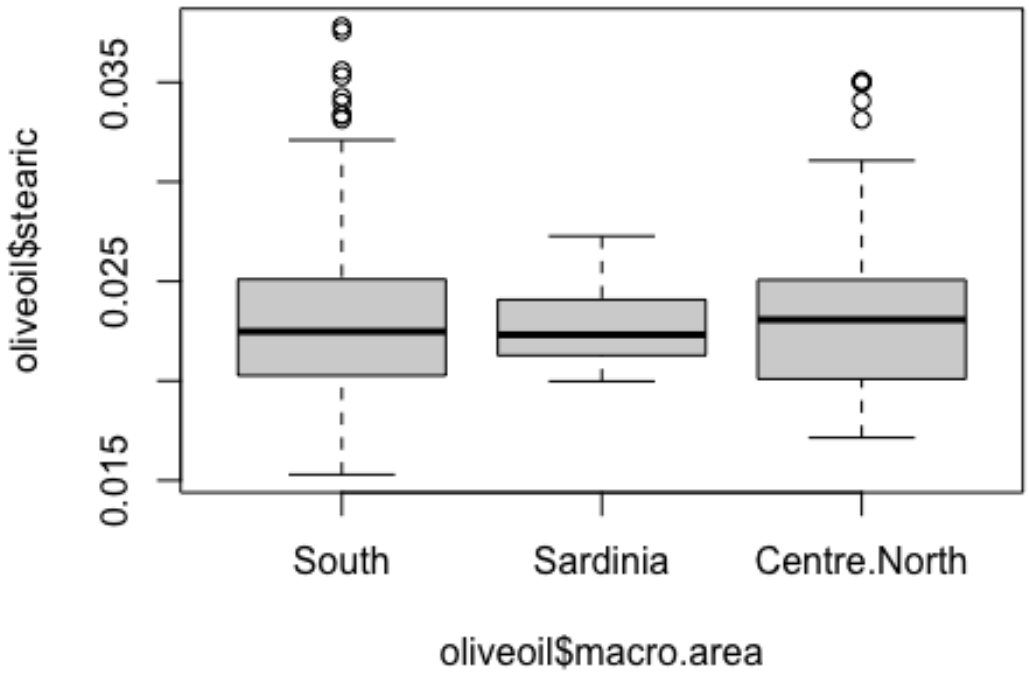
```
display_summary_and_var(oliveoil$stearic)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.
Max.
## 1.529235e-02 2.054622e-02 2.238321e-02 2.300381e-02 2.497253e-02
3.780034e-02
##          var           sd           sk
## 1.361223e-05 3.689476e-03 9.931820e-01
```

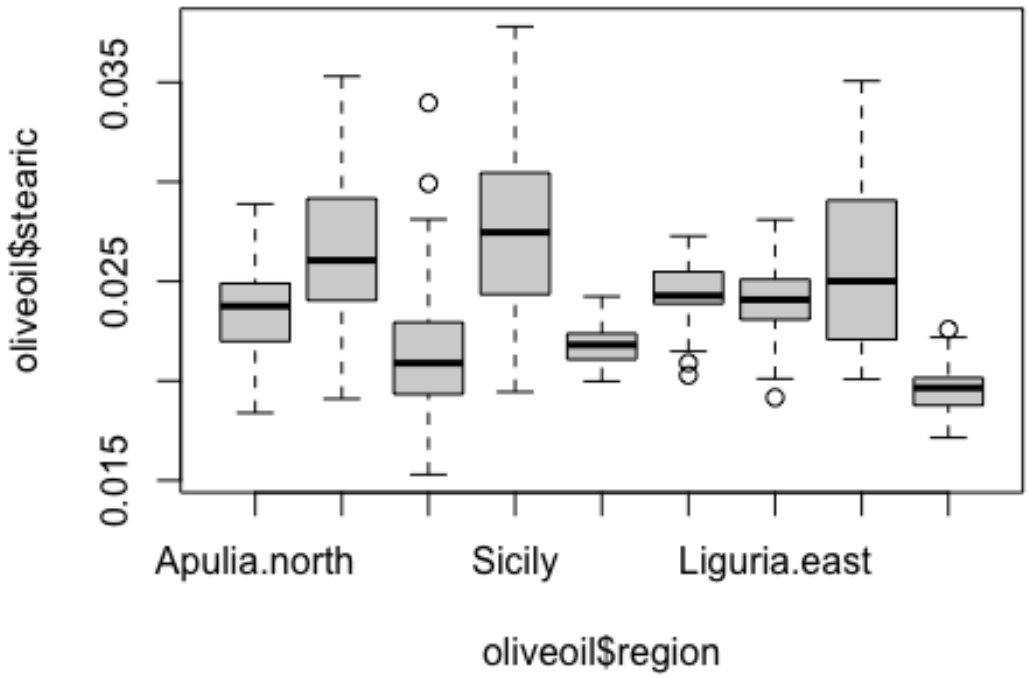
```
hist(oliveoil$stearic, col = 5, main = "stearic")
```




```r
boxplot(oliveoil$stearic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$stearic~oliveoil$region)
```



Istogramma: La distribuzione è leggermente asimmetrica a destra, con la maggior parte dei valori compresi tra 0.02 e 0.03. Boxplot (Area Macro): La regione Sud ha un valore mediano più alto, mentre Sardegna e Centro-Nord hanno valori più bassi. Boxplot (Regione): I valori mediani più alti si trovano in Puglia Nord e Sicilia. La variabilità è alta, con molti outlier in queste regioni.

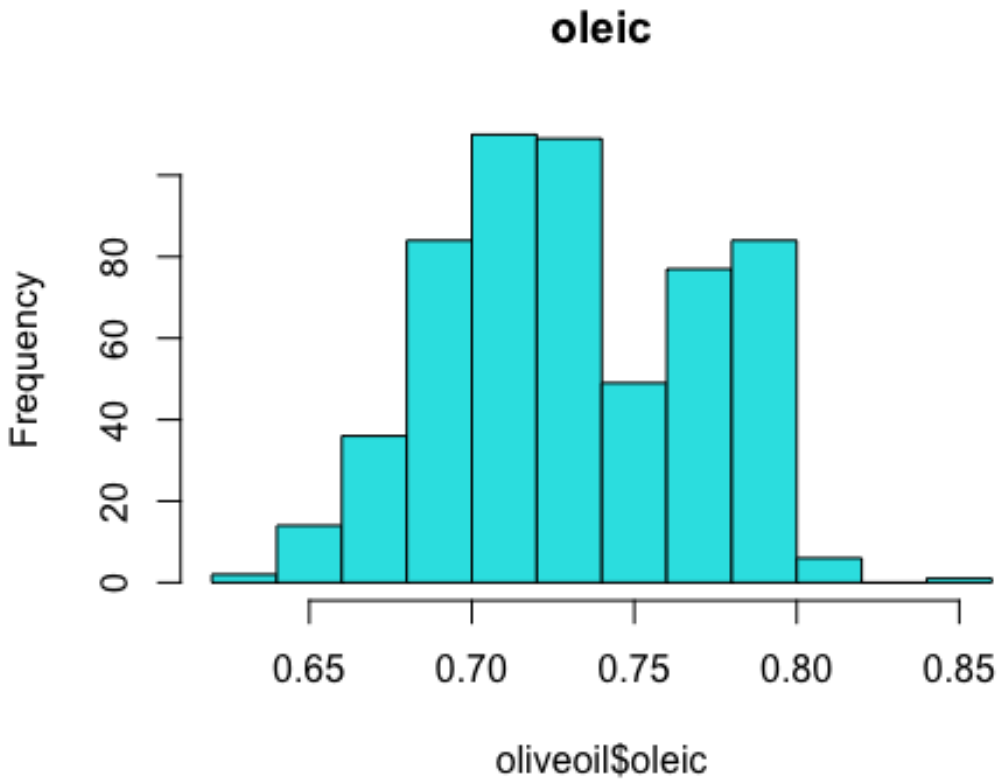
Variabile acido oleic

È un acido grasso monoinsaturo con 18 atomi di carbonio e un doppio legame nella posizione 9. È il principale componente dell'olio d'oliva e di molti altri oli vegetali, noto per le sue proprietà benefiche per la salute cardiovascolare.

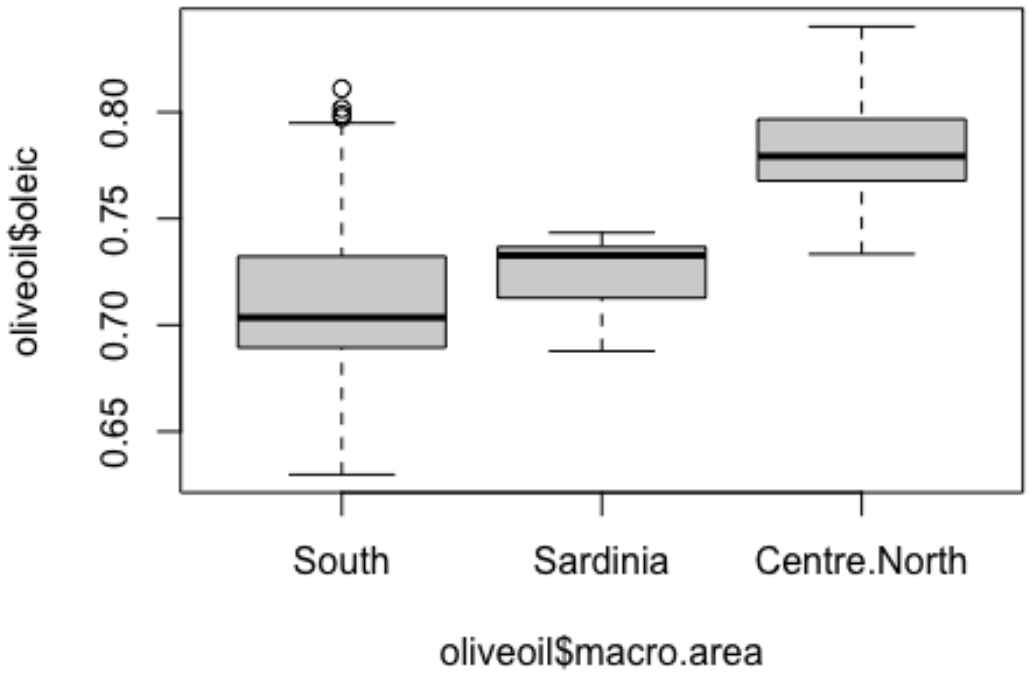
```
display_summary_and_var(oliveoil$oleic)
```

```
##        Min.     1st Qu.      Median        Mean     3rd Qu.        Max.
## 0.629785107 0.700292277 0.731948276 0.731767279 0.767939389 0.840175807
##         var          sd          sk
## 0.001642159 0.040523556 0.072586417
```

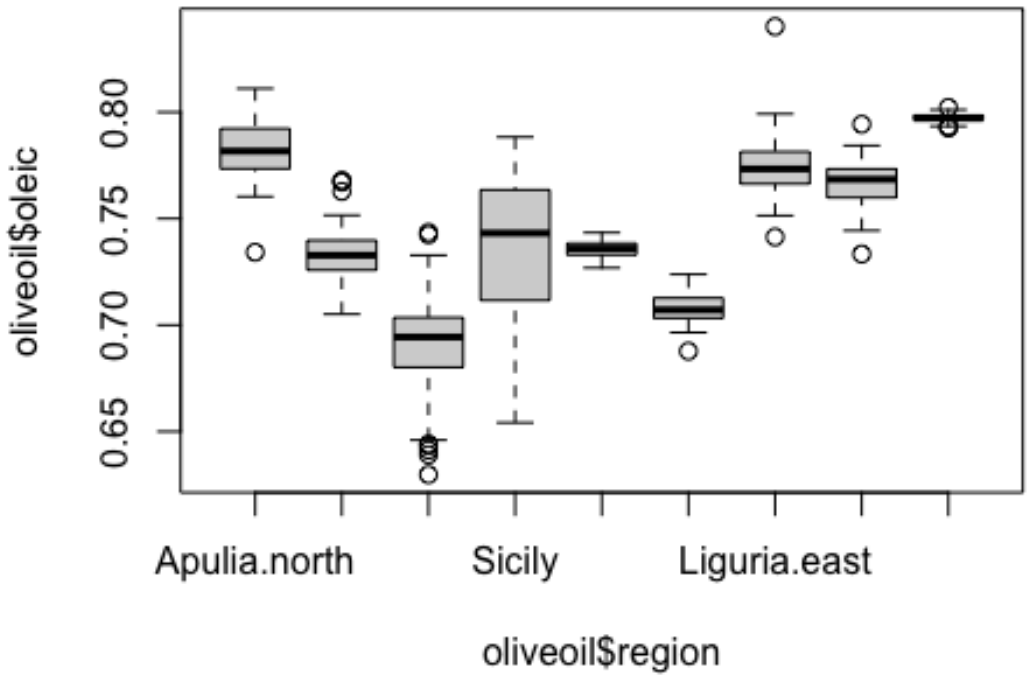
```
hist(oliveoil$oleic, col = 5, main = "oleic")
```




```r
boxplot(oliveoil$oleic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$oleic~oliveoil$region)
```



Istogramma: La distribuzione è approssimativamente normale, centrata intorno a 0.75. Boxplot (Area Macro): La Sardegna mostra un valore mediano più alto rispetto al Sud e al Centro-Nord. Boxplot (Regione): Puglia Nord, Sicilia e Liguria Est hanno valori mediani più alti, con la Sicilia che mostra la gamma più ampia e molti outlier, indicando una significativa variabilità nel contenuto di acido oleico in questa regione.

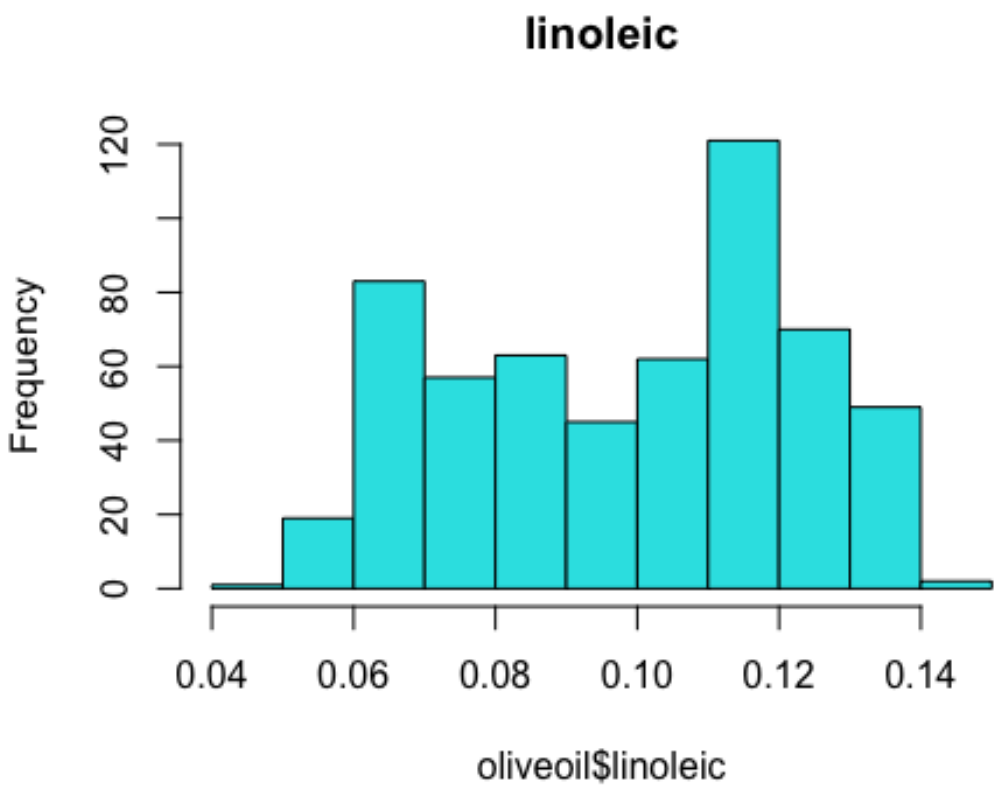
Variabile acido linoleic

È un acido grasso polinsaturo con 18 atomi di carbonio e due doppi legami nelle posizioni 9 e 12. È essenziale per il corpo umano, che non può sintetizzarlo, e si trova in oli come quello di girasole e di mais. È importante per la salute della pelle e la funzione cellulare.

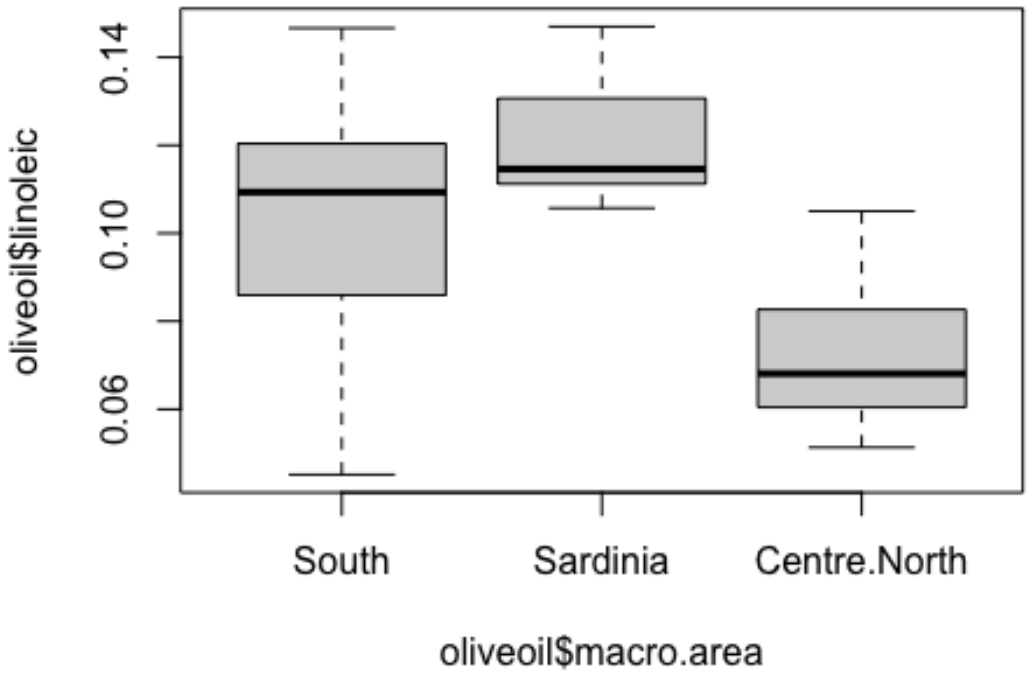
```
display_summary_and_var(oliveoil$linoleic)
```

```
##          Min.       1st Qu.        Median          Mean       3rd Qu.
##  0.0451392380  0.0774603568  0.1038897389  0.0982054895  0.1181070736
##          Max.           var            sd            sk
##  0.1469824141  0.0005874833  0.0242380548 -0.2130257102
```

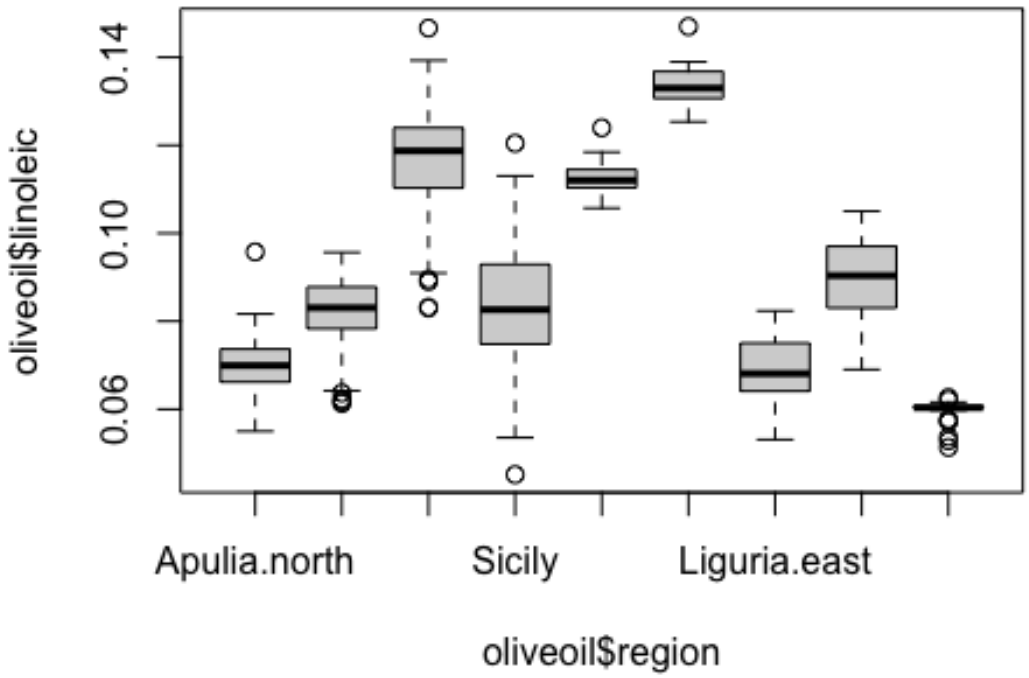
```
hist(oliveoil$linoleic, col = 5, main = "linoleic")
```




```r
boxplot(oliveoil$linoleic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$linoleic~oliveoil$region)
```



Istogramma: La distribuzione è leggermente asimmetrica a destra, con un intervallo di valori compreso tra 0.04 e 0.14. Boxplot (Area Macro): La regione Sud ha il valore mediano più alto, mentre Sardegna e Centro-Nord hanno valori mediani significativamente più bassi. Boxplot (Regione): Puglia Nord, Sicilia e Liguria Est mostrano valori mediani più alti. La variabilità all'interno di queste regioni è alta, indicando composizioni dell'olio molto diverse.

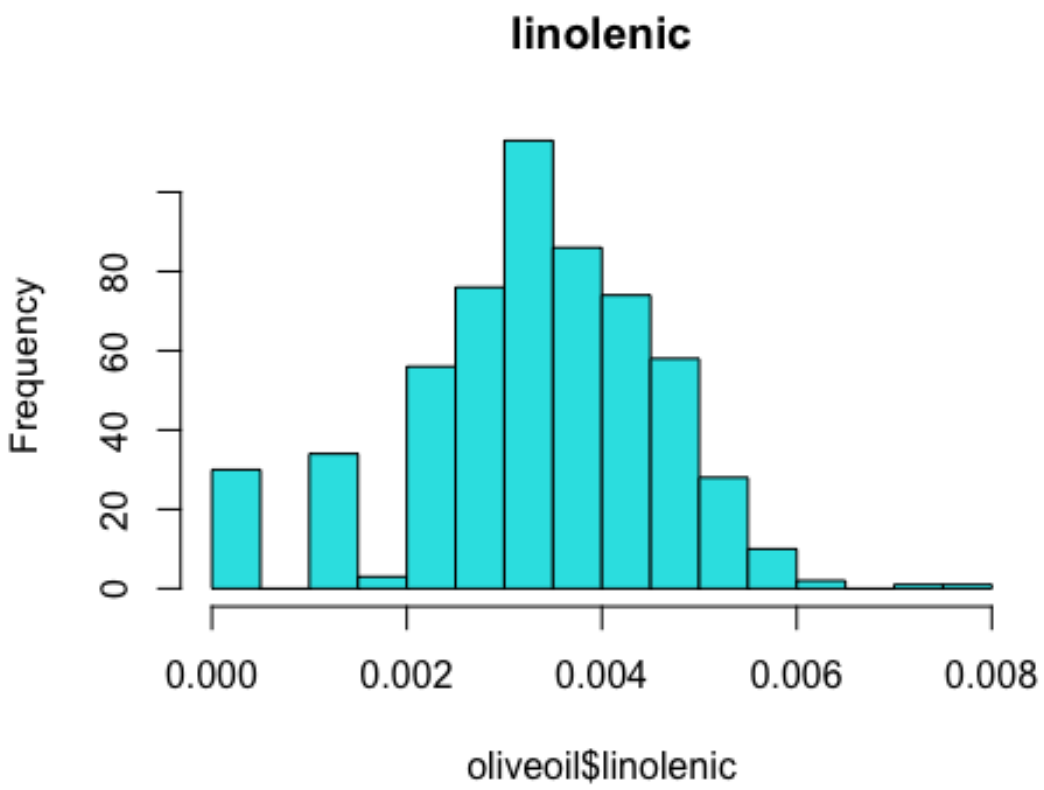
Variabile acido linolenic

È un acido grasso polinsaturo con 18 atomi di carbonio e tre doppi legami nelle posizioni 9, 12 e 15. È essenziale e si trova negli oli di semi di lino e di soia. Ha un ruolo cruciale nella funzione cerebrale e nella crescita normale.

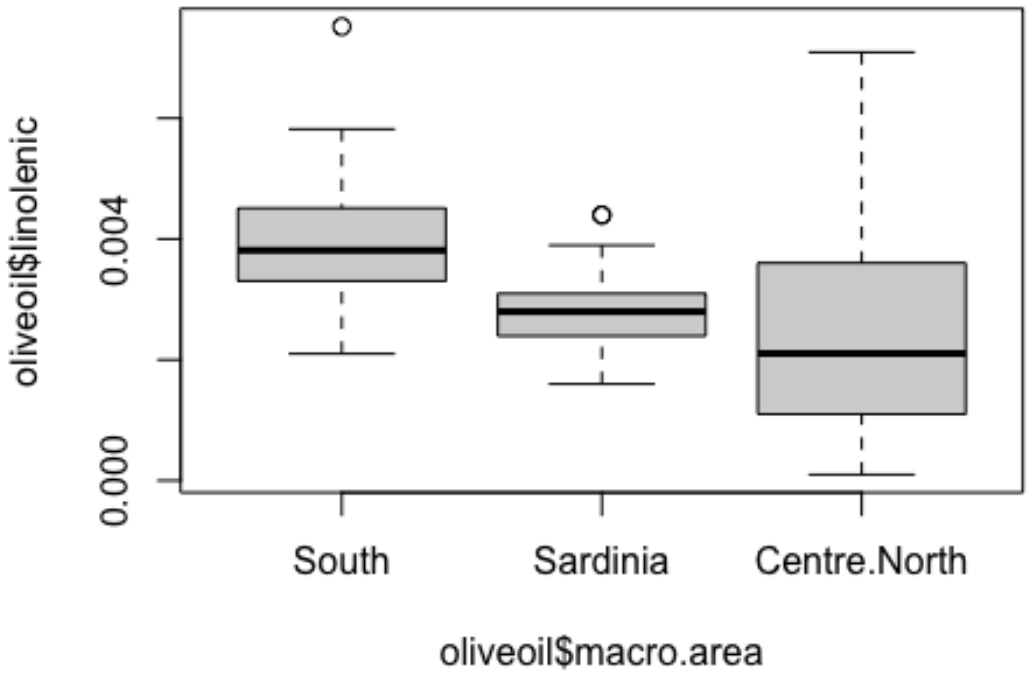
```
display_summary_and_var(oliveoil$linolenic)
```

```
##          Min.       1st Qu.        Median          Mean       3rd Qu.
##  9.891197e-05  2.697774e-03  3.372348e-03  3.292132e-03  4.148334e-03
##          Max.           var            sd            sk
##  7.517290e-03  1.690120e-06  1.300046e-03 -5.450932e-01
```

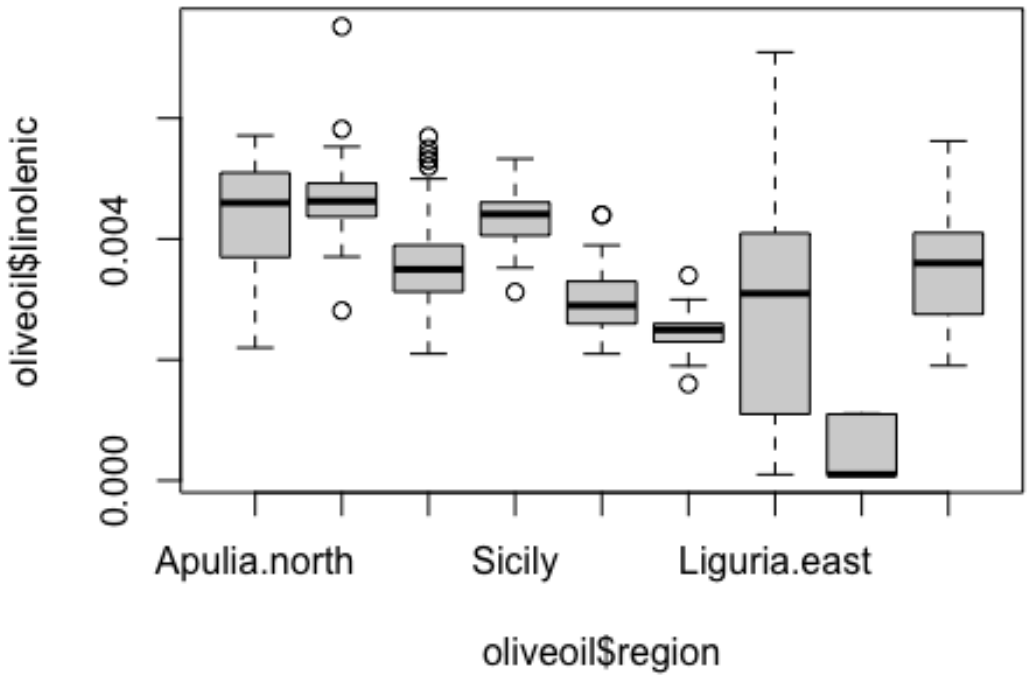
```
hist(oliveoil$linolenic, col = 5, main = "linolenic")
```




```r
boxplot(oliveoil$linolenic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$linolenic~oliveoil$region)
```



Istogramma: La distribuzione è approssimativamente normale ma leggermente asimmetrica a destra. La maggior parte dei valori rientra tra 0.002 e 0.006. Boxplot (Area Macro): La regione Sud mostra di nuovo valori mediani più alti. Centro-Nord e Sardegna mostrano valori più bassi, con la Sardegna che ha la mediana più bassa. Boxplot (Regione): I valori mediani più alti si trovano in regioni come la Puglia Nord e la Sicilia. La variabilità in queste regioni è alta, con diversi outlier nella Liguria Est.

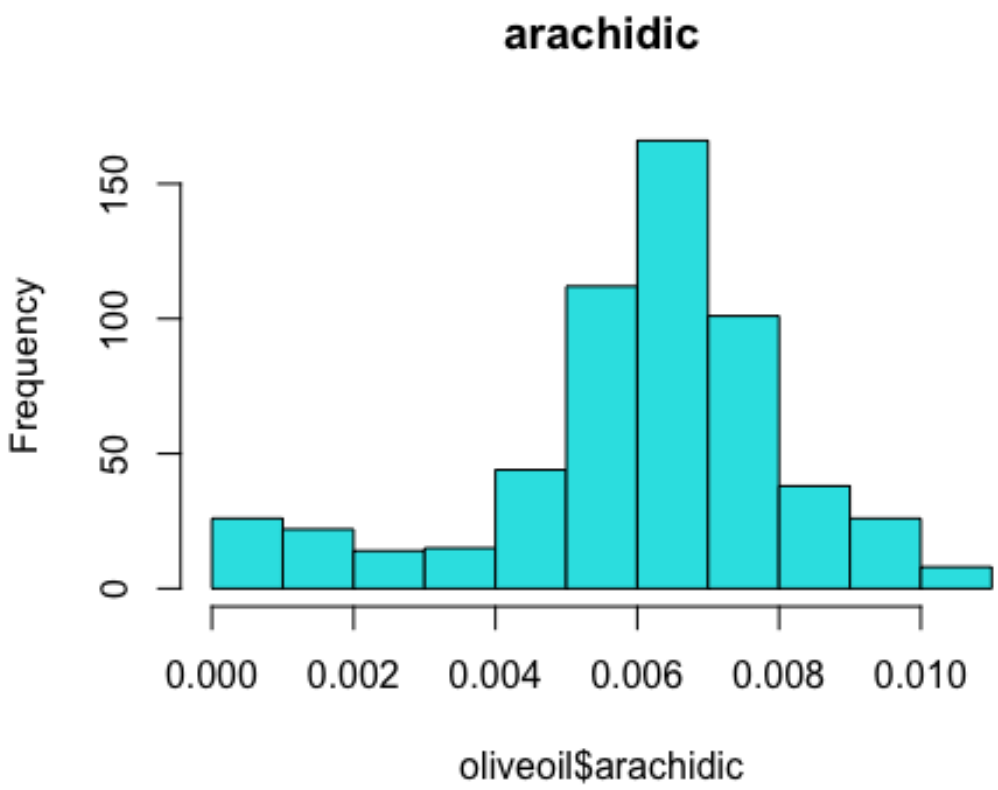
Variabile acido arachidic

È un acido grasso saturo con 20 atomi di carbonio. Si trova in piccole quantità nell'olio di arachidi e nel burro di cacao. È solido a temperatura ambiente e viene utilizzato in alcuni processi industriali.

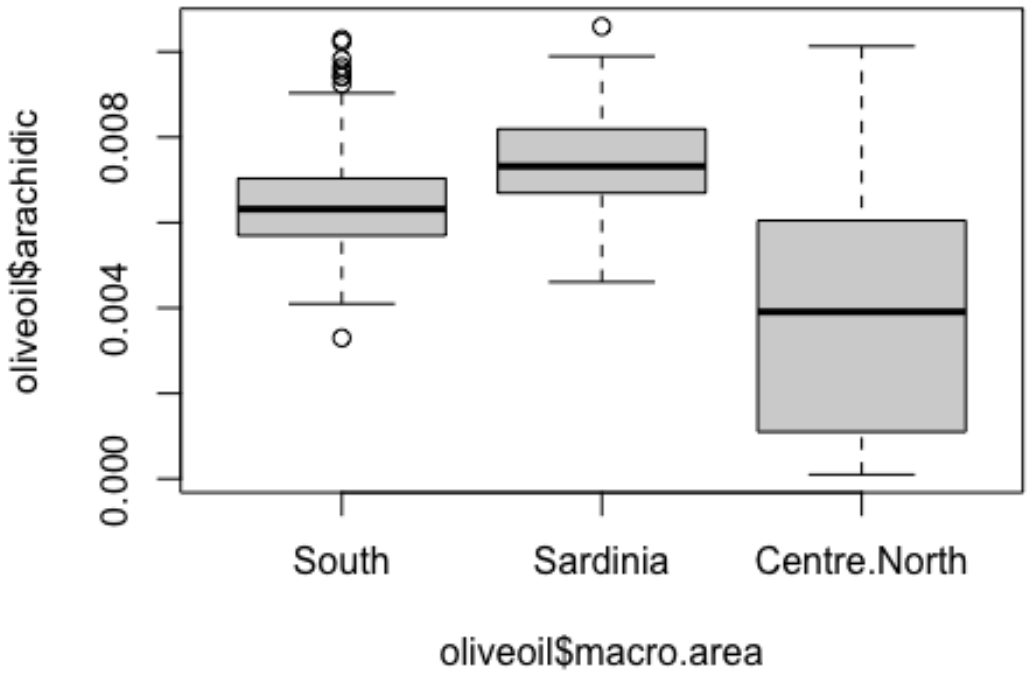
```
display_summary_and_var(oliveoil$arachidic)
```

```
##          Min.       1st Qu.        Median          Mean       3rd Qu.
##  9.891197e-05  5.121000e-03  6.227404e-03  5.914368e-03  7.116725e-03
##          Max.           var            sd            sk
##  1.059047e-02  4.857210e-06  2.203908e-03 -9.846961e-01
```

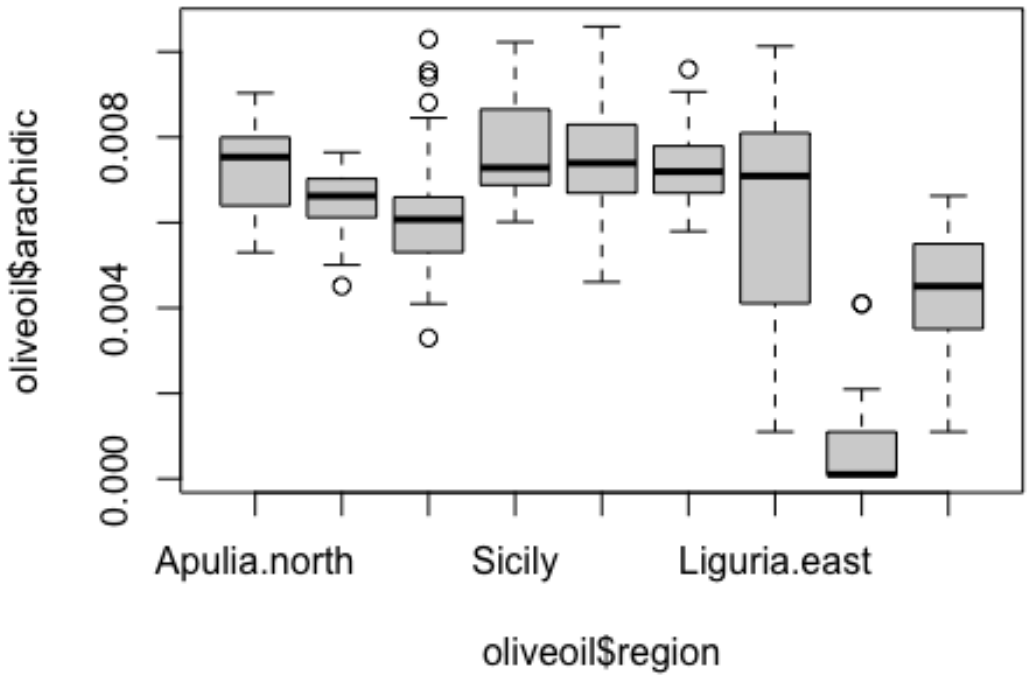
```
hist(oliveoil$arachidic, col = 5, main = "arachidic")
```




```
boxplot(oliveoil$arachidic~oliveoil$macro.area)
```




```r
boxplot(oliveoil$arachidic~oliveoil$region)
```



Istogramma: La distribuzione è leggermente asimmetrica a destra con un picco intorno a 0.006. Questo suggerisce che la maggior parte dei campioni di olio d'oliva ha livelli moderati di acido arachidico. Boxplot (Area Macro): La regione Sud mostra un valore mediano più alto, mentre Sardegna e Centro-Nord hanno valori mediani più bassi e simili. Boxplot (Regione): Regioni come la Puglia Nord e la Sicilia hanno valori mediani più alti rispetto ad altre come la Liguria Est. Ci sono outlier nel Sud e nella Puglia Nord.

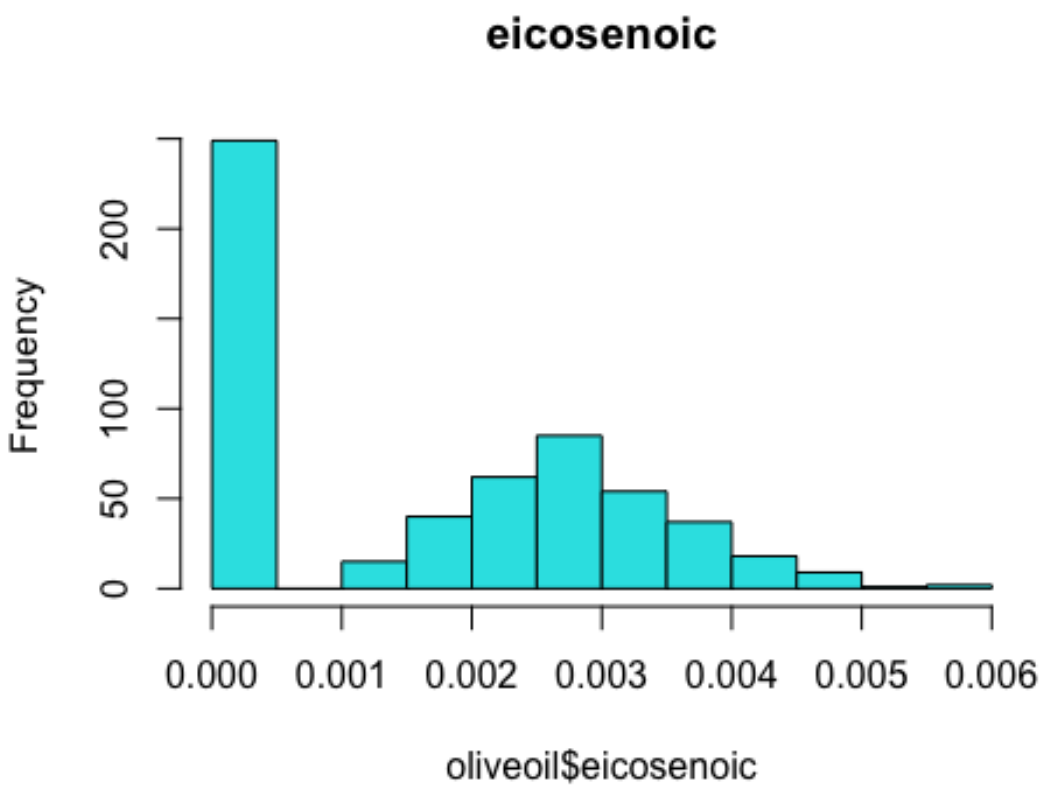
Variabile acido eicosenoic

È un acido grasso monoinsaturo con 20 atomi di carbonio e un doppio legame nella posizione 11. È presente in piccole quantità negli oli vegetali e ha proprietà simili ad altri acidi grassi monoinsaturi, contribuendo alla fluidità delle membrane cellulari.

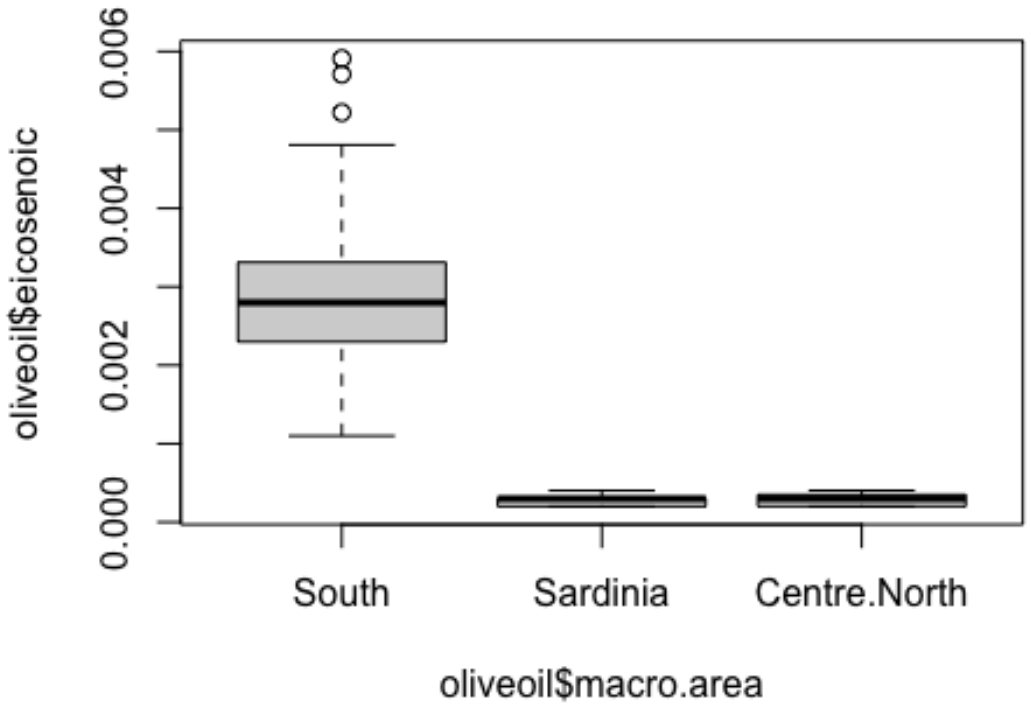
```
display_summary_and_var(oliveoil$eicosenoic)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.
Max.
## 1.982554e-04 2.998426e-04 1.798561e-03 1.730742e-03 2.898551e-03
5.908863e-03
##          var           sd           sk
## 1.992673e-06 1.411621e-03 3.434460e-01
```

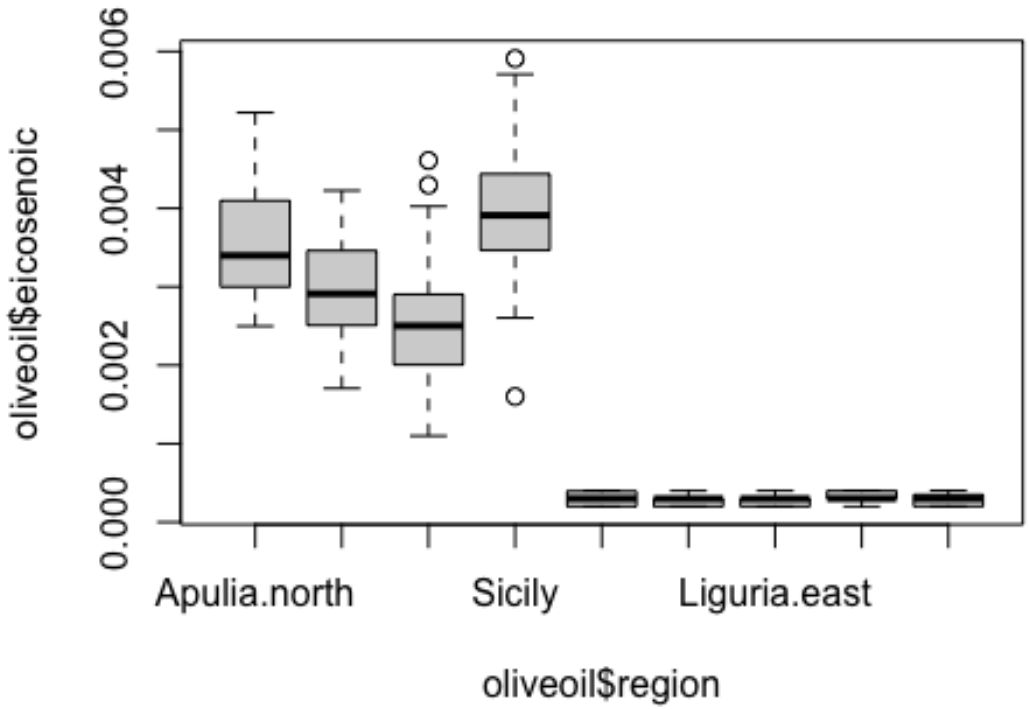
```
hist(oliveoil$eicosenoic, col = 5, main = "eicosenoic")
```




```r
boxplot(oliveoil$eicosenoic~oliveoil$macro.area)
```



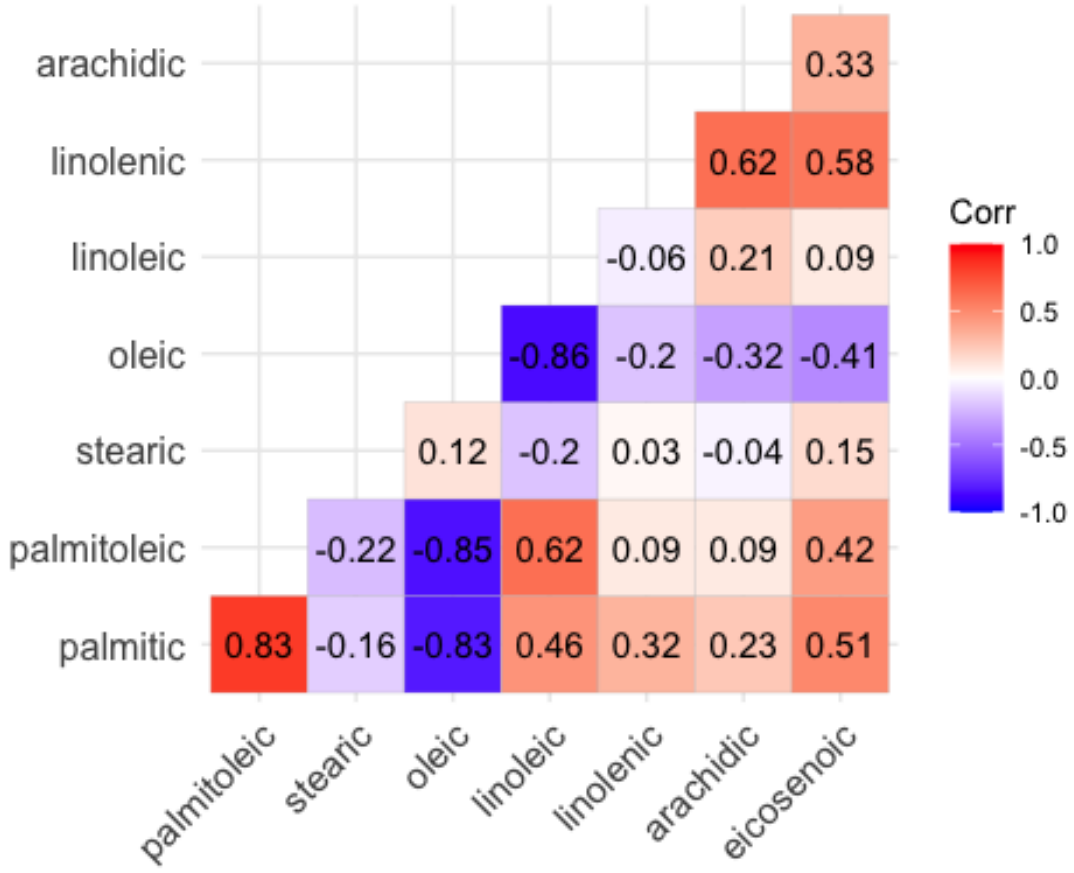

```r
boxplot(oliveoil$eicosenoic~oliveoil$region)
```



Istogramma: La distribuzione dell'acido eicosenoico è fortemente asimmetrica a destra, con la maggior parte dei campioni che presentano una concentrazione molto bassa (vicina a 0.000). Questo indica che alte concentrazioni di acido eicosenoico sono rare. Boxplot (Area Macro): Il boxplot mostra che la regione Sud ha una concentrazione mediana più alta rispetto alla Sardegna e al Centro-Nord, con queste ultime due che hanno valori molto bassi e simili. Boxplot (Regione): Le regioni come la Puglia Nord e la Sicilia mostrano valori mediani più alti, mentre regioni come la Liguria Est hanno concentrazioni costantemente basse. Ci sono diversi outlier nel Sud e in Sicilia.

Correlazione tra gli acidi

```
ggcorrplot(cor(oliveoil[,3:10]), type = "lower", lab = TRUE)
```



Notiamio che le variabili con correlazione maggiore sono - palmitic palmitoleic - oleic palmitic - oleic palmitoleic - linoleic palmitoleic - linoleic oleic - arachidic linolenic - eicosenoic palmitic - eicosenoic linolenic

analizzeremo in seguito queste coppie di variabili nel dettaglio.

Cluster Analysis

Abbiamo utilizzato tre diversi algoritmi di clusterizzazione: - k-means - PAM - DBSCAN

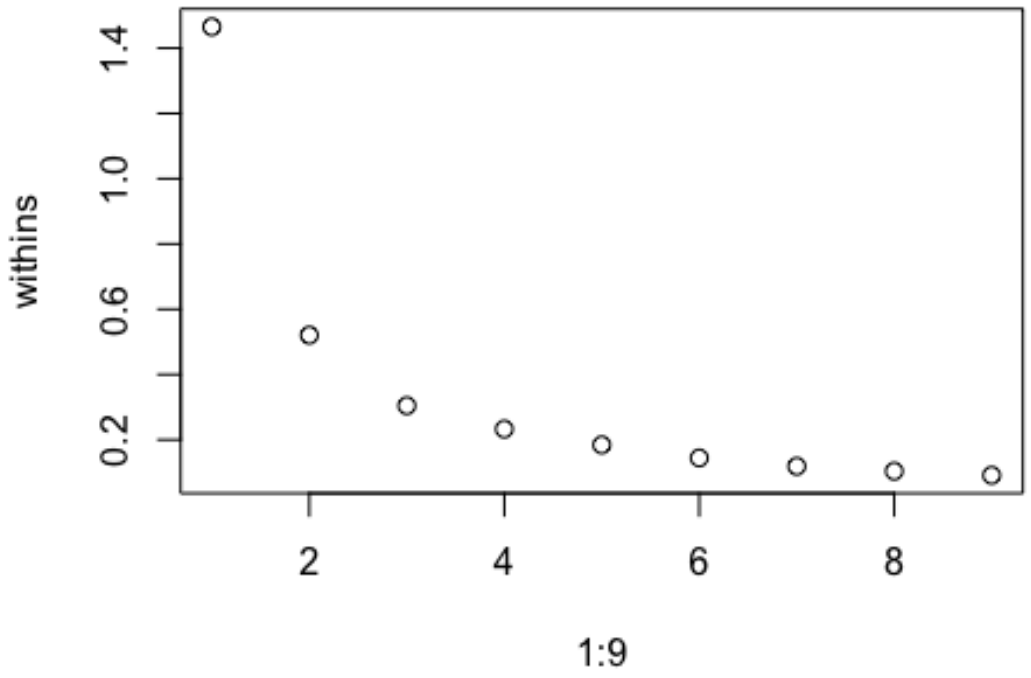
K-Means

K-Means è un metodo di raggruppamento in cluster che misura le distanze dei punti di un cluster dal suo centro e cerca di minimizzarla. L'algoritmo prende in input il dataset e il numero di cluster voluto e opera nel seguente modo: 1. si sceglie K punti casuali diversi dai punti del dataset che sono i centroidi dei cluster 2. si associa ogni dato al centroide più vicino 3. per ogni gruppo si trova il punto medio che diventerà il nuovo centroide di quel gruppo 4. itero dal punto 2 fino a quando nessun dato cambia gruppo tra un'iterazione e l'altra

L'algoritmo viene implementato in R attraverso la funzione kmeans()

Per la scelta del miglior K usiamo il metodo elbow: Si prova a implementare k-means con diversi valori di K, e per ognuno si calcola la withinss, ovvero la somma dei quadrati delle distanze tra i punti e il centro del cluster a cui appartengono. Quindi si fa il plot dei valori ottenuti e si sceglie il miglior compromesso tra una bassa withinss e un numero di cluster adeguato

```
withins <- c(1:9)
for (i in 1:9){
  km.out <- kmeans(oliveoil[, 3:10], centers = i, nstart = 15)
  withins[i] <- km.out$tot.withinss
}
plot(1:9, withins)
```

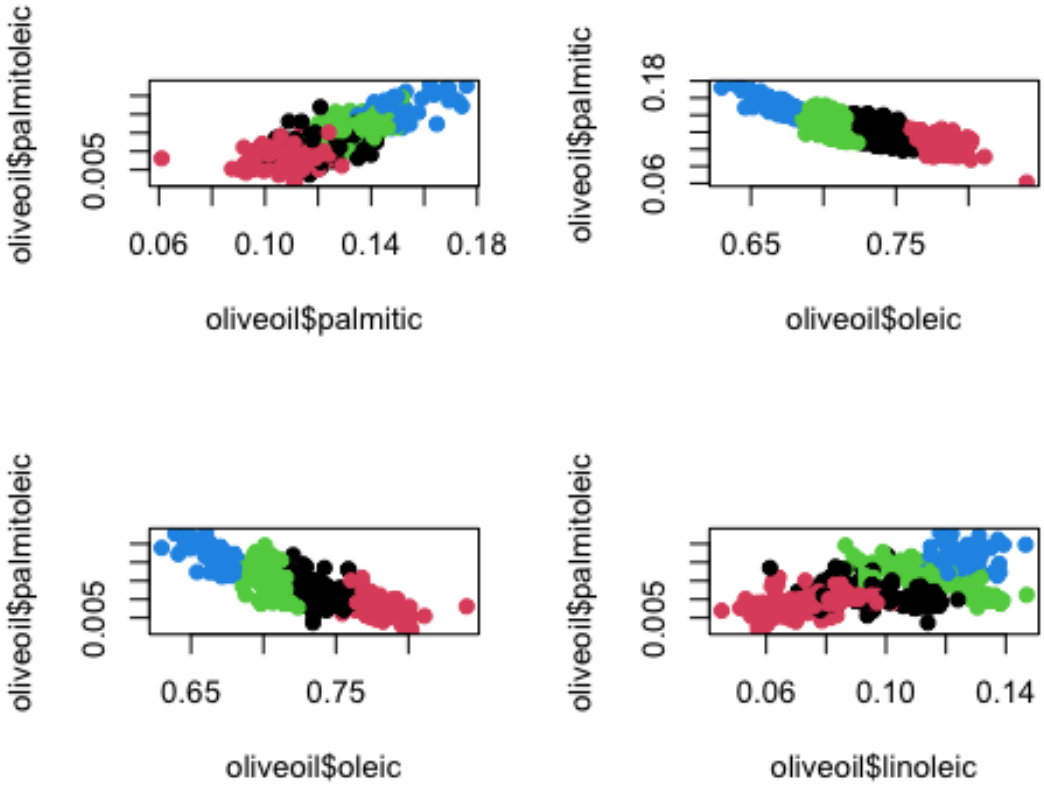



```
# dal grafico si nota che il numero migliore di cluster è 3 o 4

# si procede con l'implementazione di kmeans con K=4
set.seed(17)
km.out <- kmeans(oliveoil[, 3:10], centers=4, nstart = 15)
str(km.out)

## List of 9
##  $ cluster     : int [1:572] 2 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:4, 1:8] 0.119 0.108 0.132 0.151 0.011 ...
```

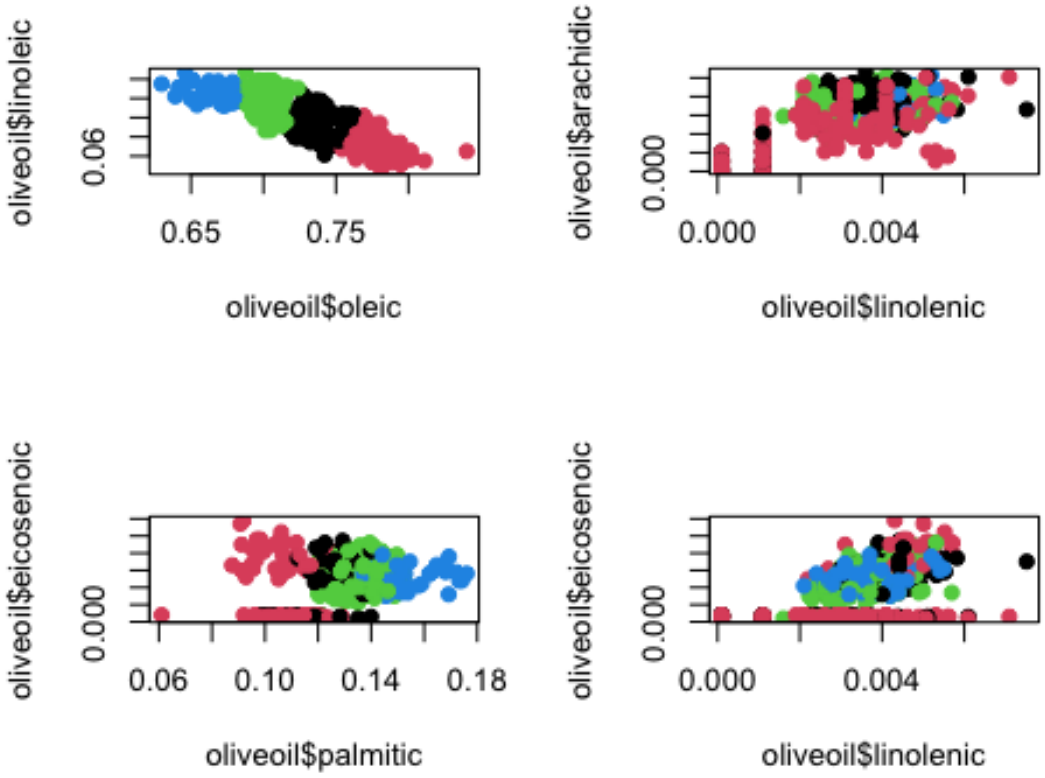
```
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:8] "palmitic" "palmitoleic" "stearic" "oleic" ...
##  $ totss       : num 1.47
##  $ withinss    : num [1:4] 0.079 0.0683 0.0637 0.0218
##  $ tot.withinss: num 0.233
##  $ betweenss   : num 1.23
##  $ size        : int [1:4] 164 167 177 64
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

```
par(mfrow=c(2,2))

# palmitic palmitoleic
plot(oliveoil$palmitic, oliveoil$palmitoleic, col = km.out$cluster, pch = 19)

# oleic palmitic
plot(oliveoil$oleic, oliveoil$palmitic, col = km.out$cluster, pch = 19)

# oleic palmitoleic
plot(oliveoil$oleic, oliveoil$palmitoleic, col = km.out$cluster, pch = 19)

# linoleic palmitoleic
plot(oliveoil$linoleic, oliveoil$palmitoleic, col = km.out$cluster, pch = 19)
```




```r
# linoleic oleic
plot(oliveoil$oleic, oliveoil$linoleic, col = km.out$cluster, pch = 19)

# arachidic linolenic
plot(oliveoil$linolenic, oliveoil$arachidic, col = km.out$cluster, pch = 19)

# eicosenoic palmitic
plot(oliveoil$palmitic, oliveoil$eicosenoic, col = km.out$cluster, pch = 19)

# eicosenoic linolenic
plot(oliveoil$linolenic, oliveoil$eicosenoic, col = km.out$cluster, pch = 19)
```



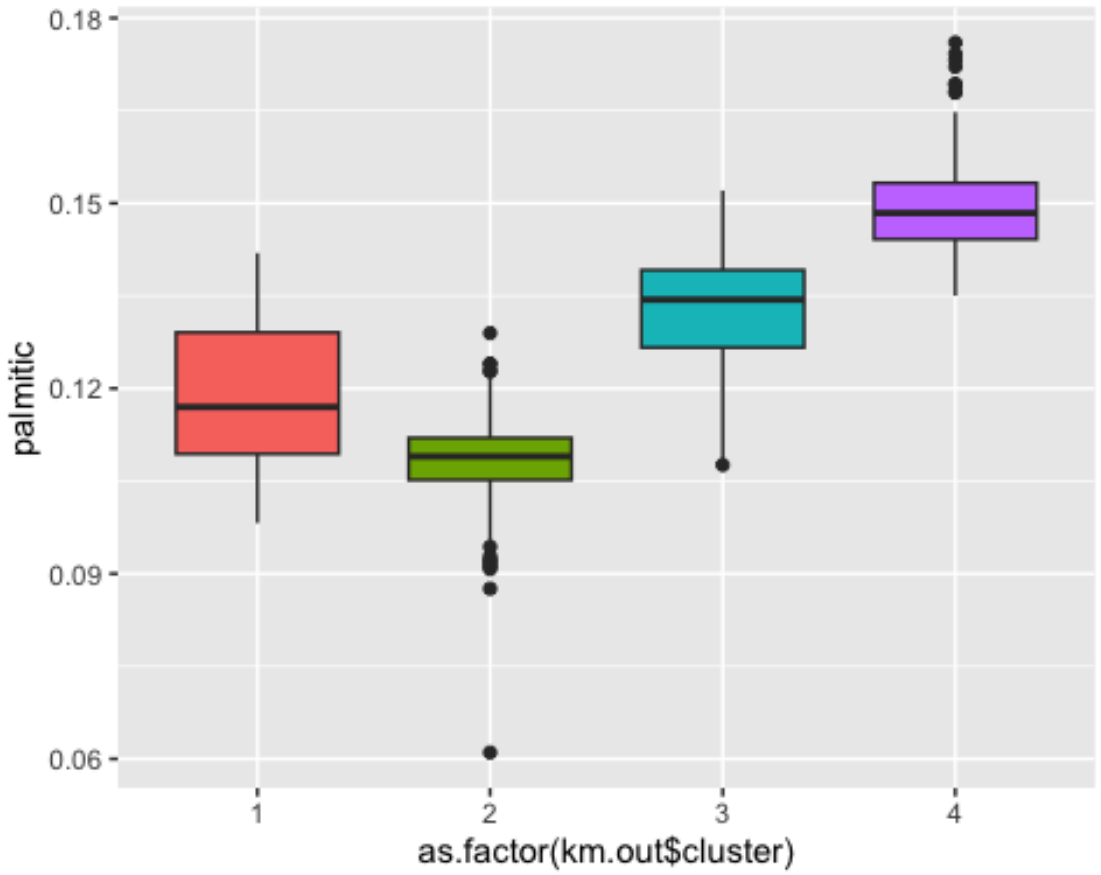

```
par(mfrow=c(1,1))
```

Notiamo che i plot con l'acido oleico sono quelli in cui la divisione tra cluster è meglio riuscita, questo perché esso è il componente principale degli oli analizzati

Variabile palmitic nei cluster

```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = palmitic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```

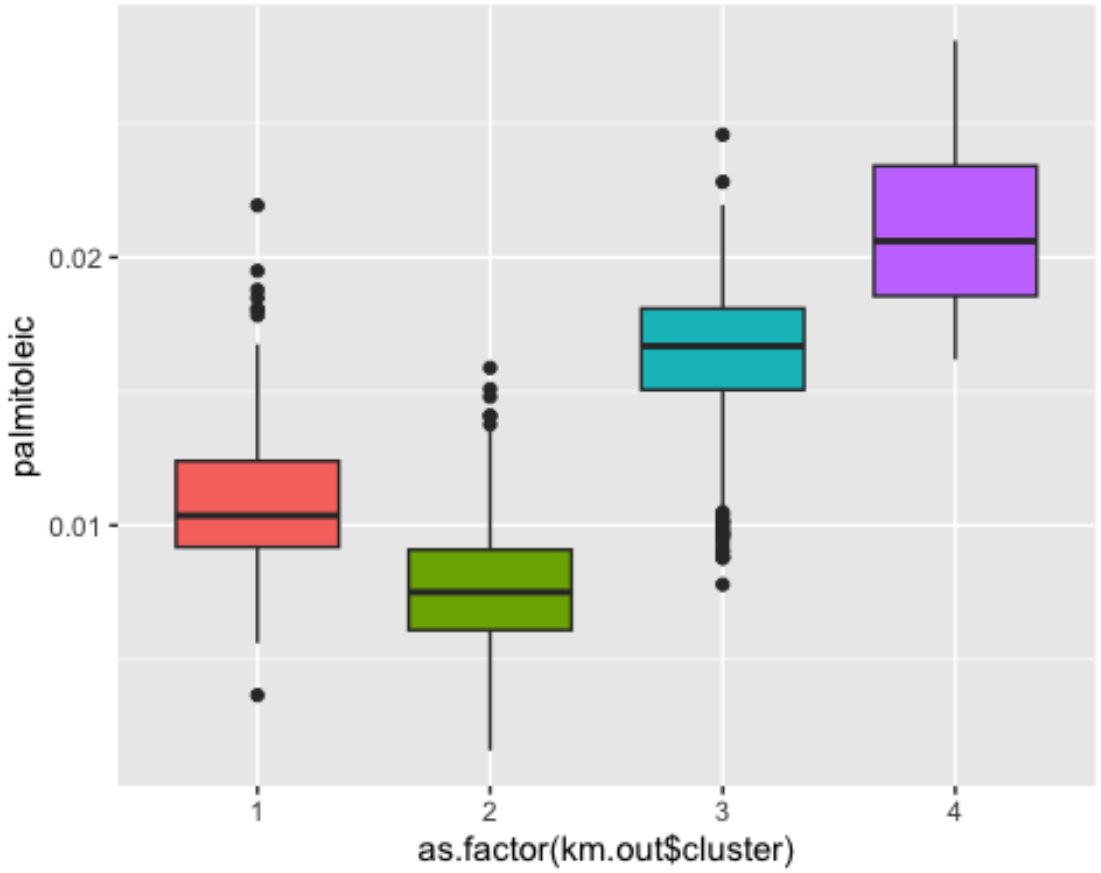
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use
"none" instead as
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Si nota che le differenze tra i cluster ripetto all'acido palmitico sono ben marcate, e presentano pochi valori outlier, la varianza è bassa. Il cluster 4 è quello che contiene oli con percentuali di acido palmitico più alte, mentre il secondo le percentuali più basse.

Variabile palmitoleic nei cluster

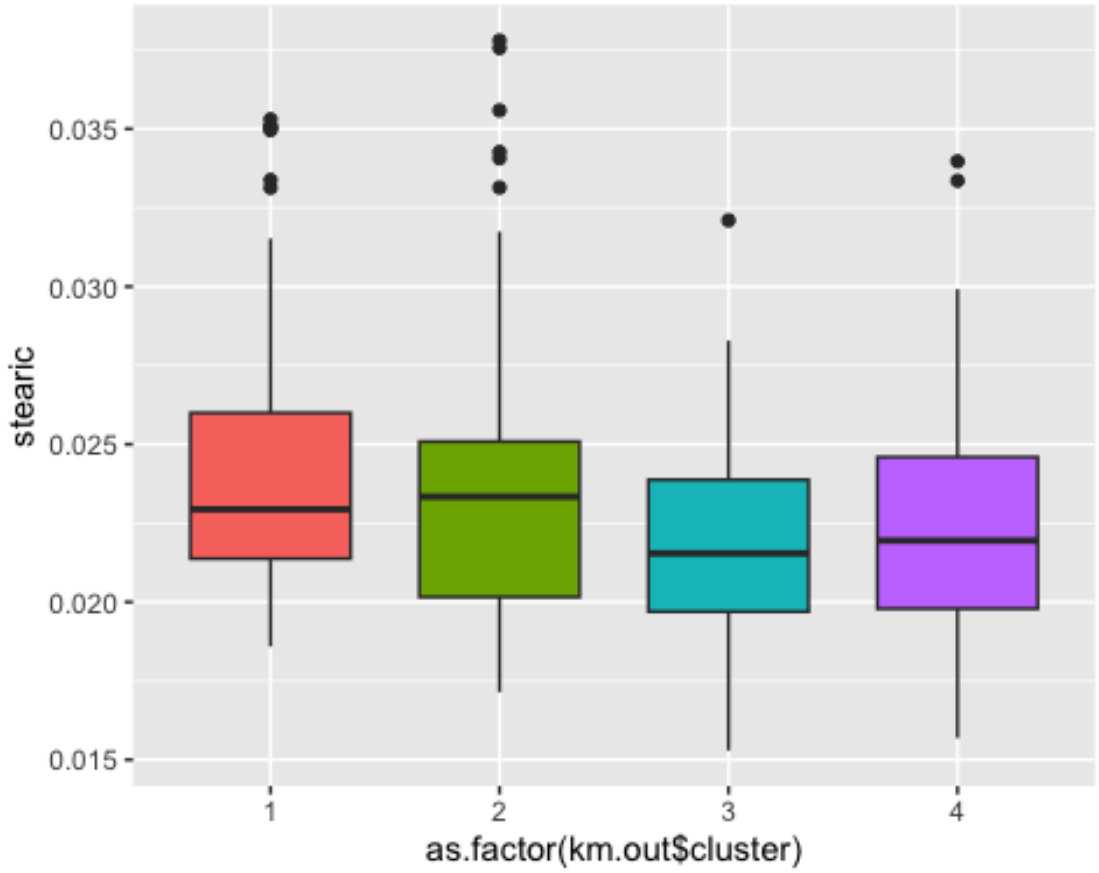
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = palmitoleic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Come nell'acido palmitico si nota che le differenze tra i cluster ripetto all'acido palmitoleico sono ben marcate, e presentano più valori outlier, ma la varianza è comunque bassa. Il cluster 4 è quello che contiene oli con percentuali di acido palmitoleico più alte, mentre il secondo le percentuali più basse.

Variabile `stearic` nei cluster

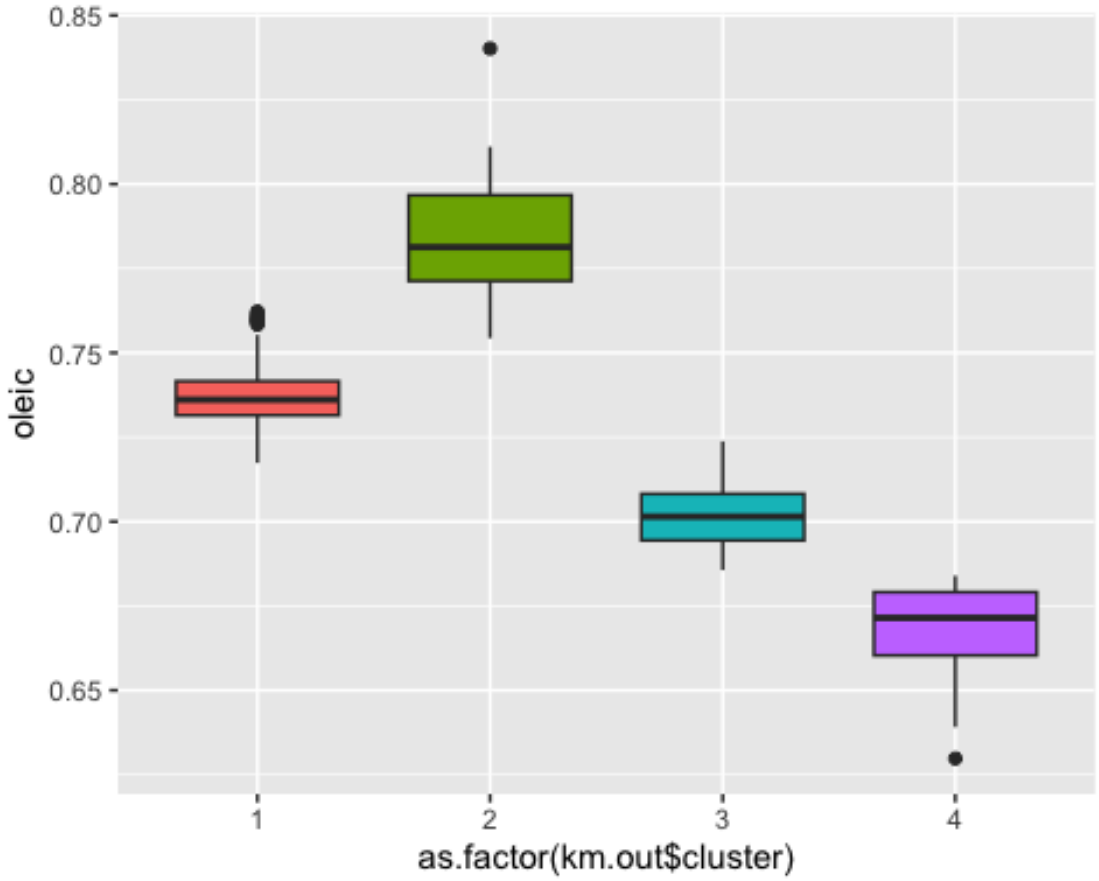
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = stearic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Si nota che le mediane sono molto vicine tra loro. L'acido stearico ha delle percentuali basse all'interno dell'olio, infatti la differenza tra i cluster è molto meno marcata

Variabile `oleic` nei cluster

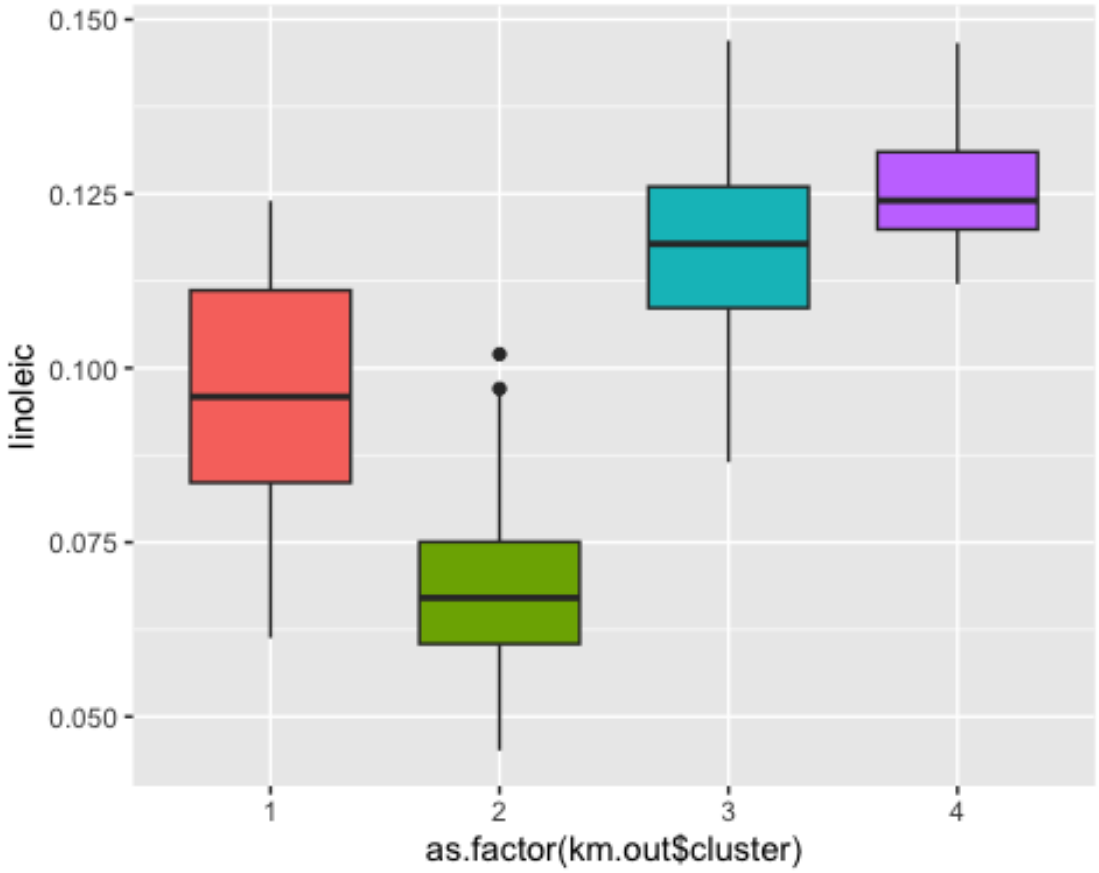
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = oleic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



L'acido oleico è in percentuale il più presente negli oli, le differenze tra i cluster sono le più marcate, come si vede anche dal boxplot Le varianze within sono basse e la varianza between è molto elevata Il cluster 2 contiene gli oli con la percentuale di acido oleico maggiore, infatti l'olio con meno acido oleico in quel cluster, è composto da almeno il 75%

Variabile `linoleic` nei cluster

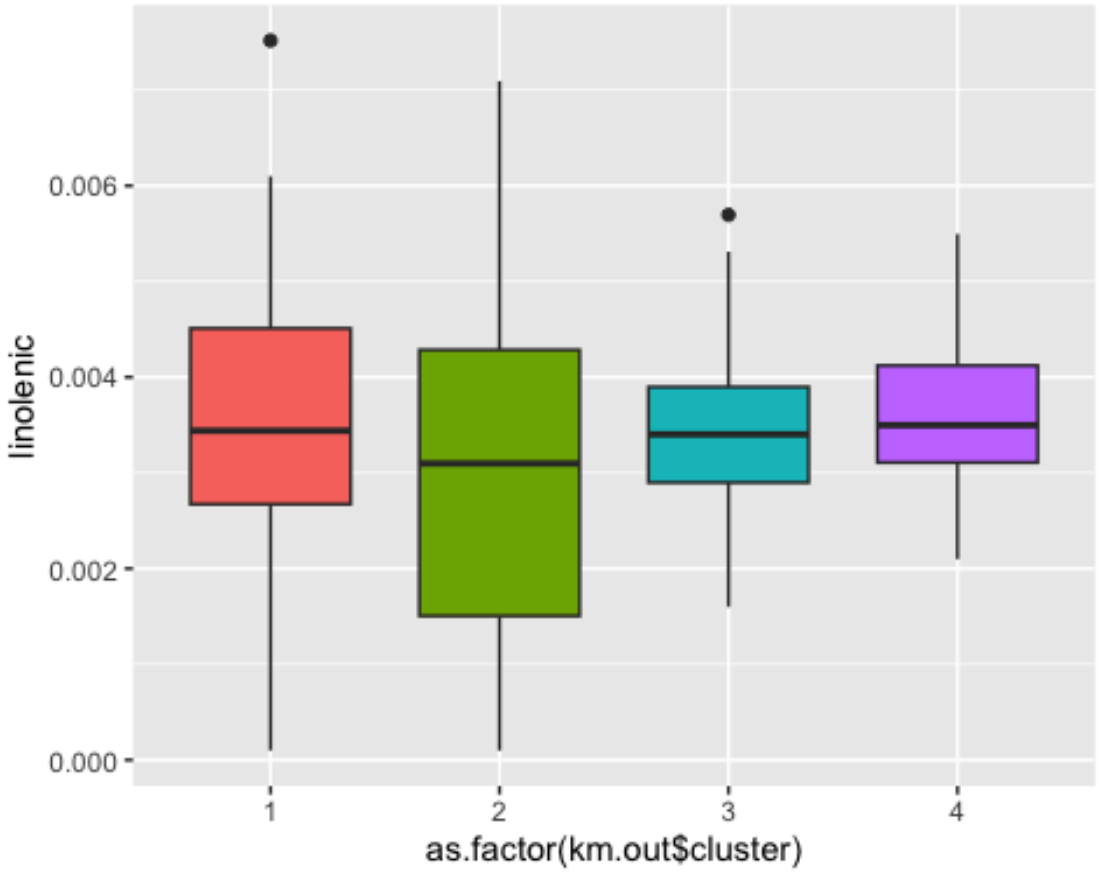
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = linoleic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



La distribuzione dei diversi cluster è molto simile all'acido palmitico e palmitoleico.

Variabile `linolenic` nei cluster

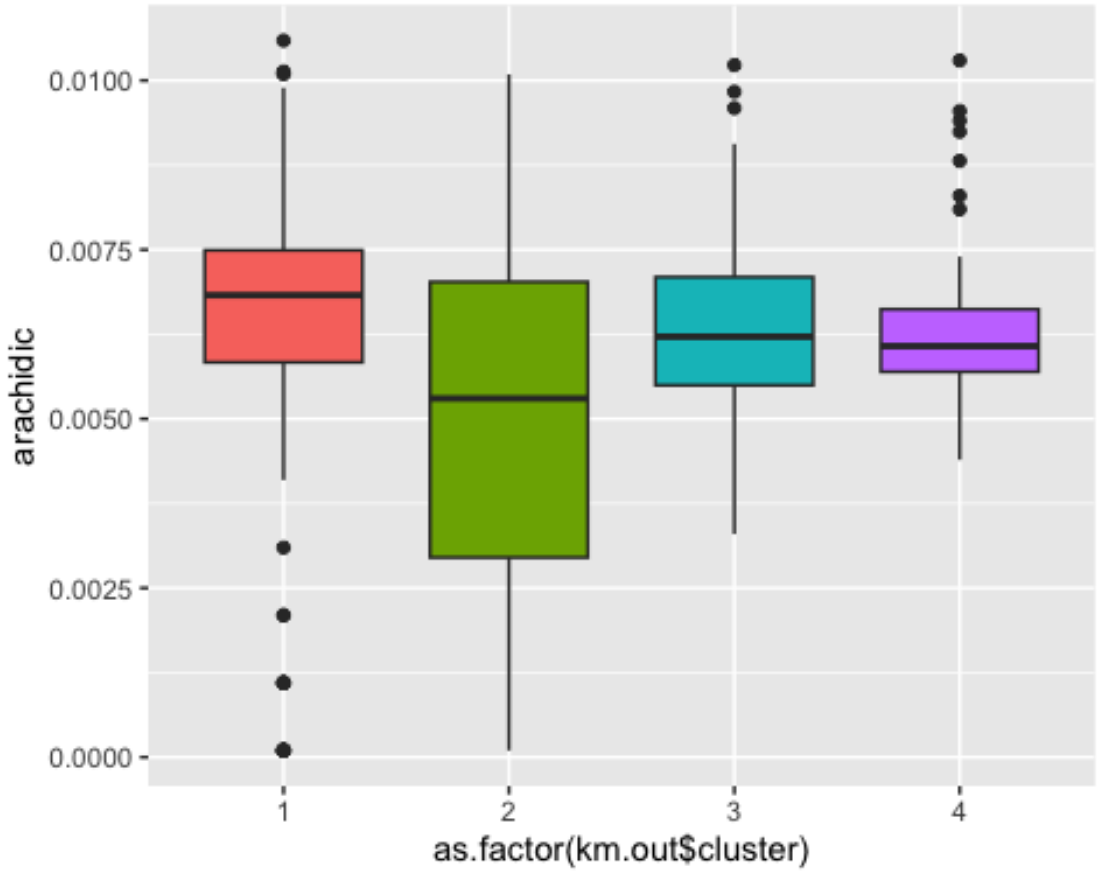
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = linolenic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Siccome è uno degli acidi con meno percentuali all'interno degli oli, l'algoritmo di k-menas con la distanza euclidea non separa in modo evidente i cluster rispetto all'acido linoleico. Infatti dal grafico si nota come le mediane sono all'incirca alla stessa altezza.

Variabile arachidic nei cluster

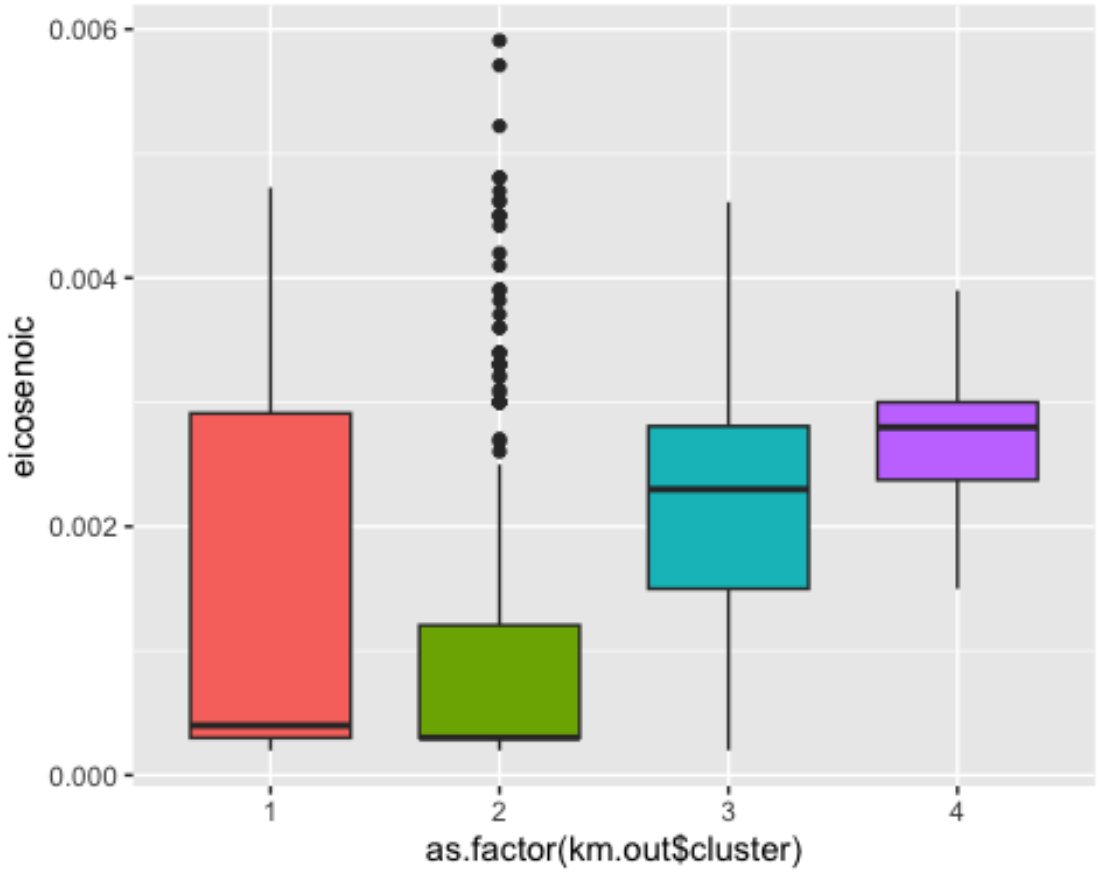
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = arachidic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Analogamente all'acido precedente non c'è differenza significativa tra cluster rispetto all'acido arachidico.

Variabile eicosenoic nei cluster

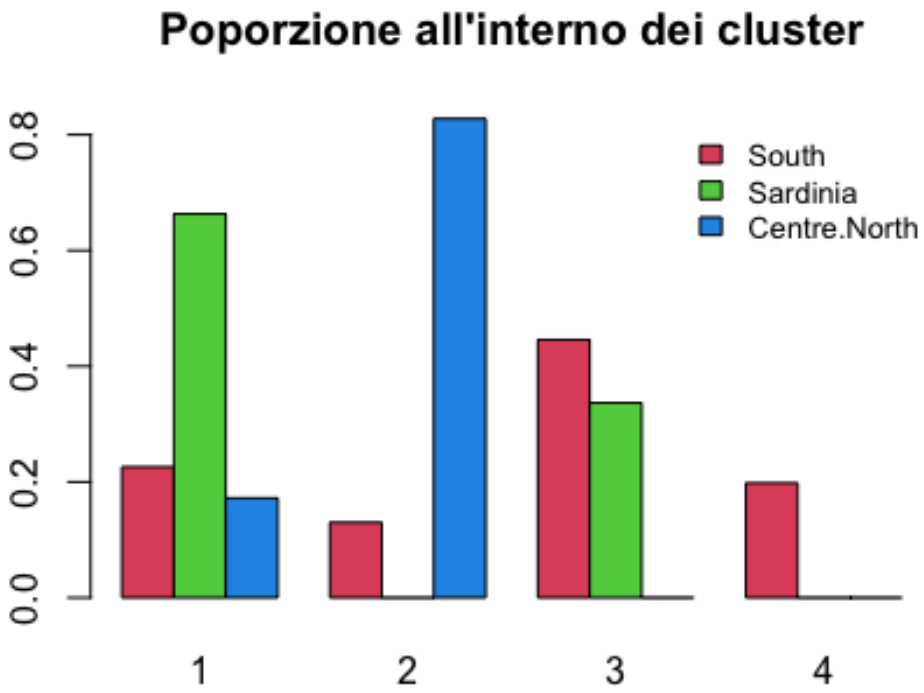
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = eicosenoic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



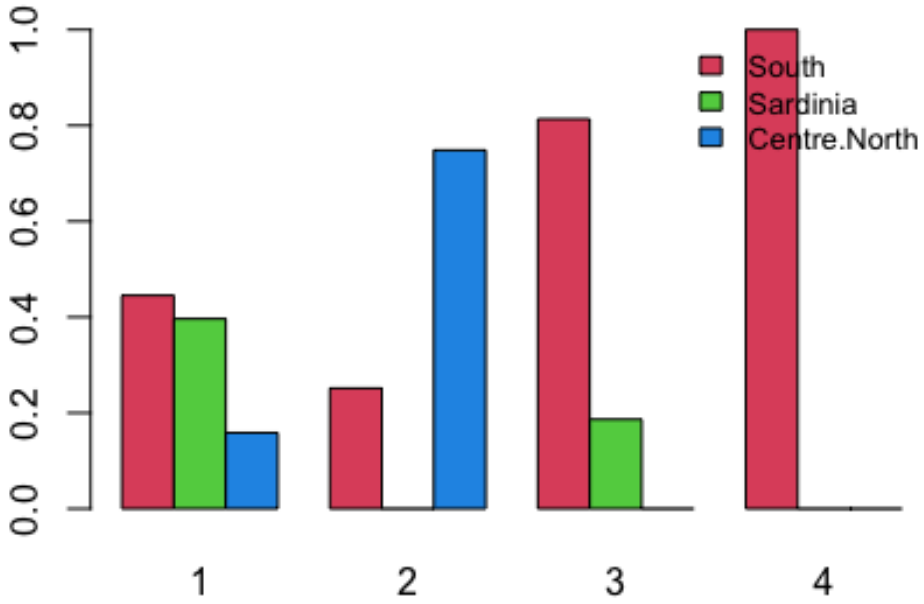
Si nota che i cluster 1 e 2 hanno ben individuato gli oli con percentuali di acido eicosenoico prossime allo zero. Oltre ai valori che tendono allo zero le distribuzioni in cluster sembrano quantomeno simili

Variabile macro.area nei cluster

```
prop.table(table(oliveoil$macro.area, km.out$cluster),1)

##
##                         1         2         3         4
##   South         0.2260062 0.1300310 0.4458204 0.1981424
##   Sardinia      0.6632653 0.0000000 0.3367347 0.0000000
##   Centre.North 0.1721854 0.8278146 0.0000000 0.0000000

barplot(prop.table(table(oliveoil$macro.area, km.out$cluster),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:4)
legend("topright", legend = rownames(prop.table(table(oliveoil$macro.area,
km.out$cluster),1)
), fill = 2:4, cex = 0.8, bty = "n")
```




```r
barplot(prop.table(table(oliveoil$macro.area, km.out$cluster),2), beside = T,
legend = F, main = "", col = 2:4)
legend("topright", legend = rownames(prop.table(table(oliveoil$macro.area,
km.out$cluster),1)
), fill = 2:4, cex = 0.8, bty = "n")
```



Il cluster 4 è composto solamente da aree del sud. Il cluster 2 comprende la maggior parte degli oli dal centro nord. Il cluster 3 comprende solamente oli dal sud e dalla sardegna. Anche se in proporzione i 3 oli sono vicini, si vede dal primo grafico che il cluster 1 comprende la gran pare deglio oli provenienti dalla sardegna.

Gli oli del sud sono per il 44% nel cluster 3, il 13% nel cluster 2, il 22% nel 1 e il 19% nel 4. Quelli della sardegna sono il 33% nel cluster 3 ,e il 66% nel 1. Mentre quelli del centro nord sono per l'83% nel 2 ,e il 17% nel 1.

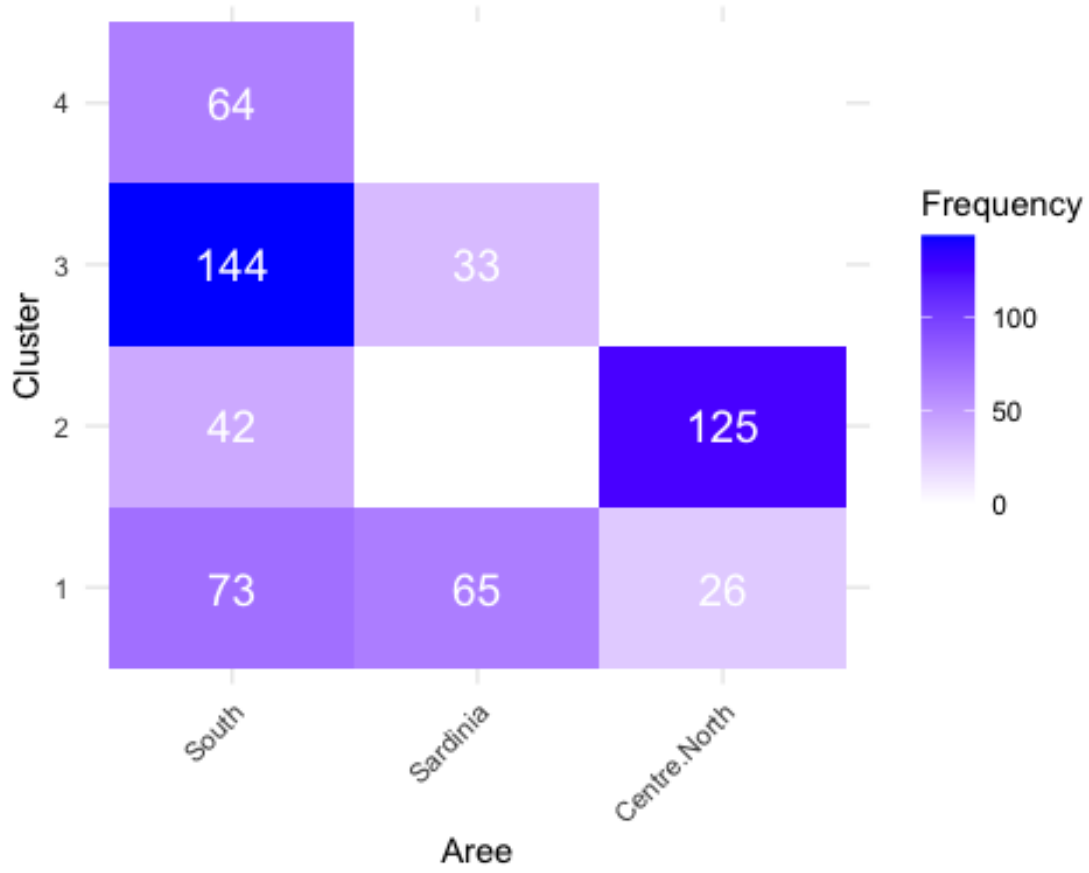
Solo gli oli del sud sono presenti in tutti e 4 i cluster, quindi hanno caratteristiche più diversivficate tra loro.

Questo si può vedere anche dalla Confuzion Matrix ovvero:

```
confusion_matrix <- table(Cluster = oliveoil$macro.area, Aree =
km.out$cluster)

table( Aree = km.out$cluster, Cluster = oliveoil$macro.area)

##     Cluster
## Aree South Sardinia Centre.North
##    1    73       65           26
##    2    42        0          125
```

```
##    3   144       33            0
##    4    64        0            0
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Aree, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Aree", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



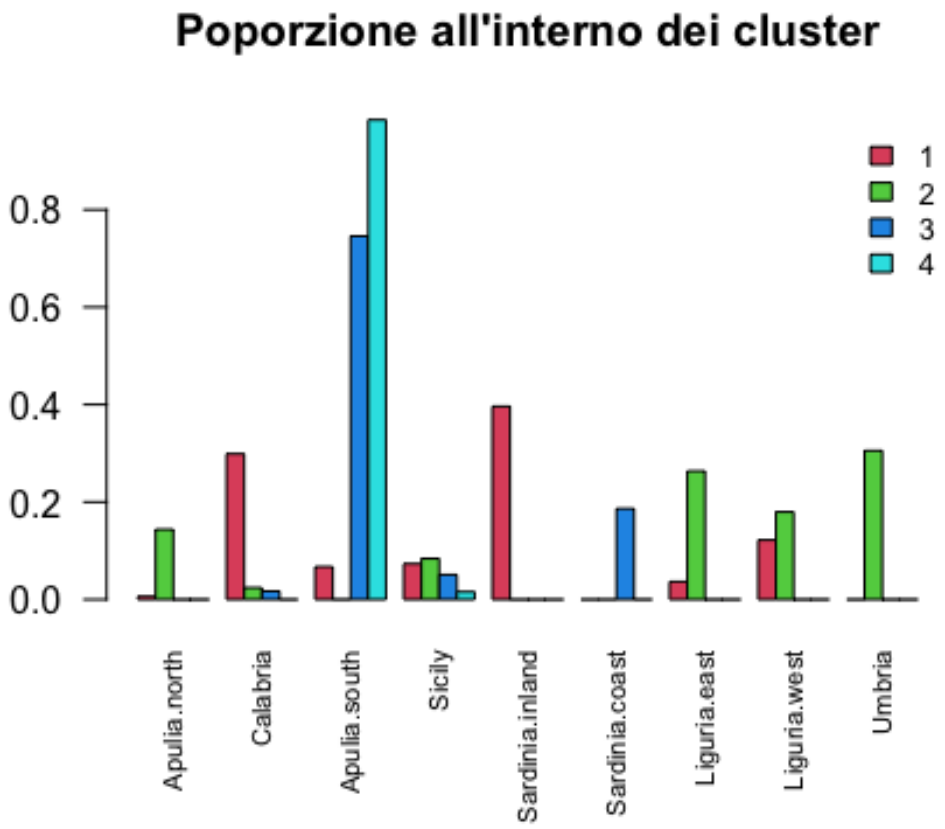
Variabile region nei cluster

```
prop.table(table(km.out$cluster, oliveoil$region),1)
```

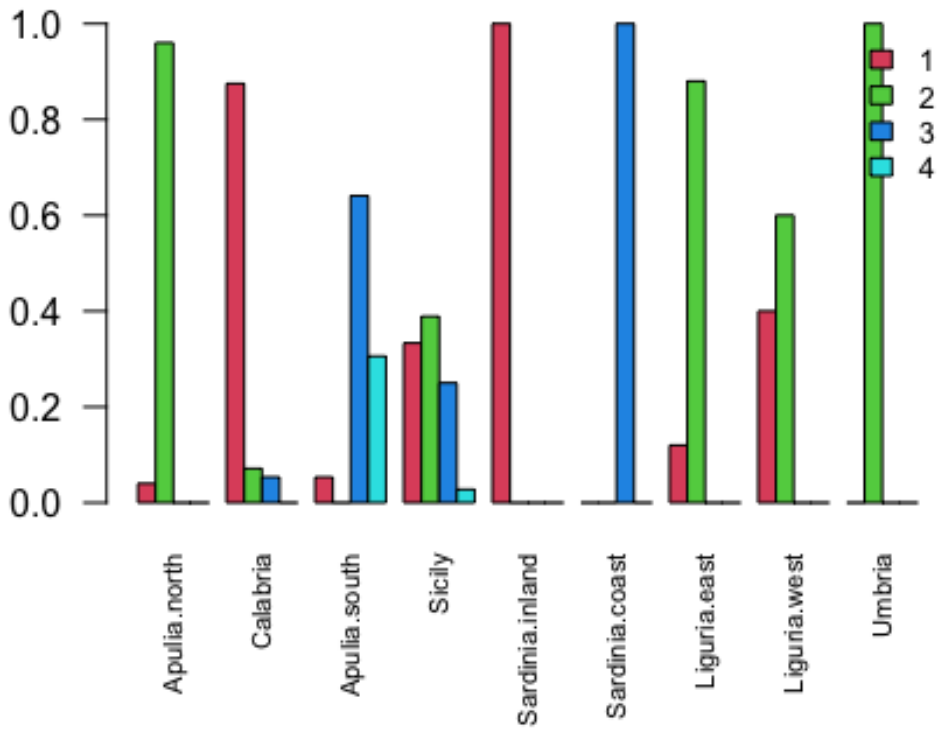
```
##
##      Apulia.north    Calabria Apulia.south      Sicily Sardinia.inland
##   1  0.006097561 0.298780488  0.067073171 0.073170732     0.396341463
##   2  0.143712575 0.023952096  0.000000000 0.083832335     0.000000000
##   3  0.000000000 0.016949153  0.745762712 0.050847458     0.000000000
##   4  0.000000000 0.000000000  0.984375000 0.015625000     0.000000000
##
##      Sardinia.coast Liguria.east Liguria.west      Umbria
```

```
##   1    0.000000000  0.036585366  0.121951220 0.000000000
##   2    0.000000000  0.263473054  0.179640719 0.305389222
##   3    0.186440678  0.000000000  0.000000000 0.000000000
##   4    0.000000000  0.000000000  0.000000000 0.000000000
```

```
barplot(prop.table(table(km.out$cluster, oliveoil$region),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:5, cex.names
= 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(km.out$cluster,
oliveoil$region),1)
), fill = 2:5, cex = 0.8, bty = "n")
```




```
barplot(prop.table(table(km.out$cluster, oliveoil$region),2), beside = T,
legend = F, main = "", col = 2:5, cex.names = 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(km.out$cluster,
oliveoil$region),1)
), fill = 2:5, cex = 0.8, bty = "n")
```



Dai grafici si nota che: I cluster 3 e 4 contengono prevelantemente oli provenienti da Puglia Nord, mentre i cluster 1 e 2 contengono oli provenienti da diverse regioni.

Gli oli provenienti da Calabria e Sardegna inland sono in gran parte nel cluster 1 Gli oli provenienti da Puglia Nord, Liguria e Umbria sono in gran parte nel cluster 2 Sardegna costa e Sardegna inland sono esclusivamente parte del cluster 1 e 3 rispettivamente. questo indica che gli oli della Sardegna hanno caratteristiche molto differenti tra quelli prodotti sulle coste e quelli prodotti inland

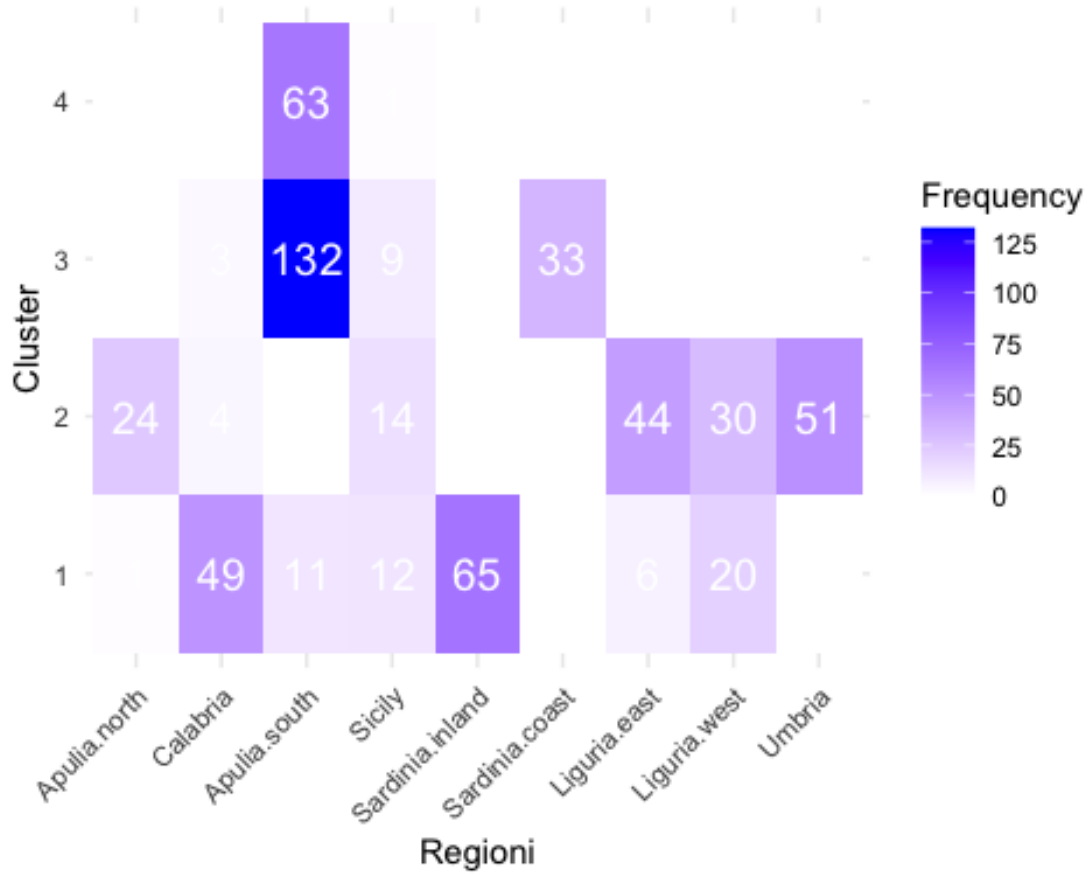
La Sicilia, al contrario, produce oli con caratteristiche molto diversificate, infatti gli oli della sicilia appartengono a tutti i cluster.

Di seguito la confusion matrix:

```
confusion_matrix <- table(Cluster = oliveoil$region, Regioni =
km.out$cluster)

table(Regioni = km.out$cluster, Cluster = oliveoil$region)

##         Cluster
## Regioni Apulia.north Calabria Apulia.south Sicily Sardinia.inland
##       1            1       49           11     12              65
##       2           24        4            0     14               0
```

```
##       3            0        3          132      9                0
##       4            0        0           63      1                0
##        Cluster
## Regioni Sardinia.coast Liguria.east Liguria.west Umbria
##       1              0           6          20      0
##       2              0          44          30     51
##       3             33           0           0      0
##       4              0           0           0      0
```

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Regioni, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Regioni", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



PAM

L'algoritmo PAM prende in input una matrice di dati numerica, un intero k che corrisponde al numero di cluster e una metrica. PAM opera nel seguente modo:

1. Si selezionano k punti (medoidi) tra i punti presenti nel dataset e si associa ogni punto al medoide più vicino secondo la metrica selezionata.
2. Si selezionano in modo casuale nuovi medoidi
3. Si calcola la somma di tutte le distanze tra ogni punto e il medoide al quale è associato.Si associa ogni punto al nuovo medoide più vicino e si calcola la somma di tutte le distanze tra ogni punto e il nuovo medoide al quale è associato.
4. Se la nuova distanza è minore della vecchia allora si scambiano i medoidi
5. Sitera dal punto 2 fino a quando non ci sono cambiamenti nell'insieme di medoidi.

Distanze

La divisione in cluster dell'algoritmo PAM dipende dalla funzione di distanza che si decide di utilizzare. Due distanze utilizzate dall'algoritmo sono:

La distanza euclidea:

$$d_E(p,q) = (\sum_{i=1}^{n}(p_i - q_i)^2)^{\frac{1}{2}}$$

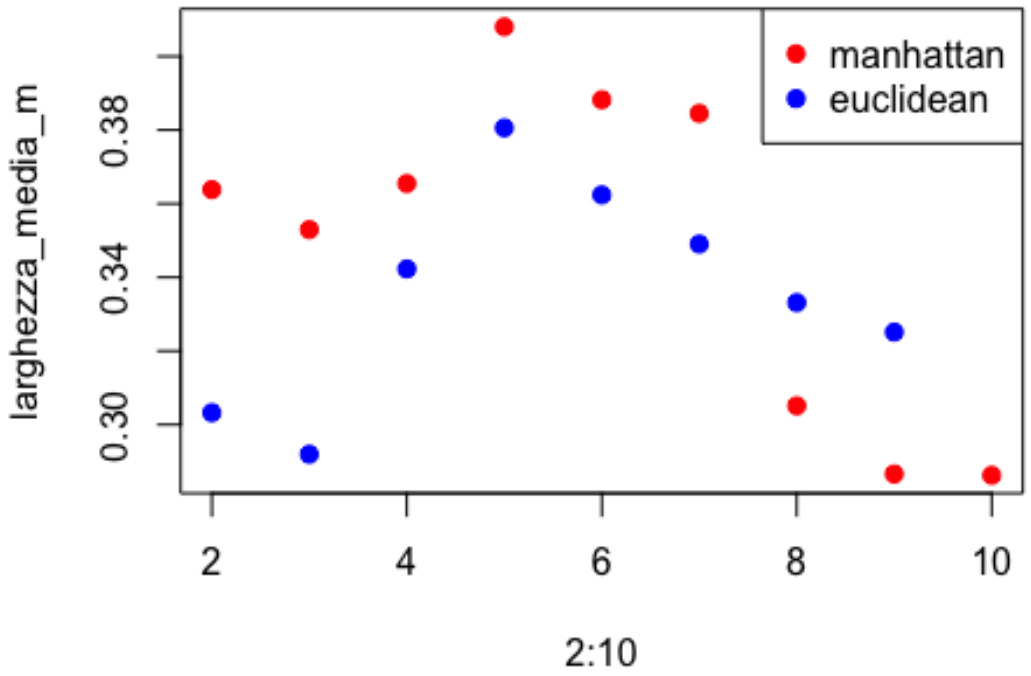
La distanza di Manatthan:

$$d_M(p,q) = \sum_{i=1}^{n}|p_i - q_i|$$

Si decide di testare l'algoritmo PAM con le due distanze e con diversi valori di K per scegliere i parametri che creano i cluster migliori.

```
# DISTANZA MANHATTAN
larghezza_media_m <- c(1:9)
for (i in 2:10){
  pam.out<-pam(oliveoil[,3:10], i, metric="manhattan", stand=TRUE, nstart =
10)
  larghezza_media_m[i-1] <- pam.out$silinfo$avg.width
}

# DISTANZA EUCLIDEA
larghezza_media_e <- c(1:9)
for (i in 2:10){
  pam.out<-pam(oliveoil[,3:10], i, metric="euclidean", stand=TRUE, nstart =
10)
  larghezza_media_e[i-1] <- pam.out$silinfo$avg.width
}


plot(2:10, larghezza_media_m, col = "red", pch = 19)
```

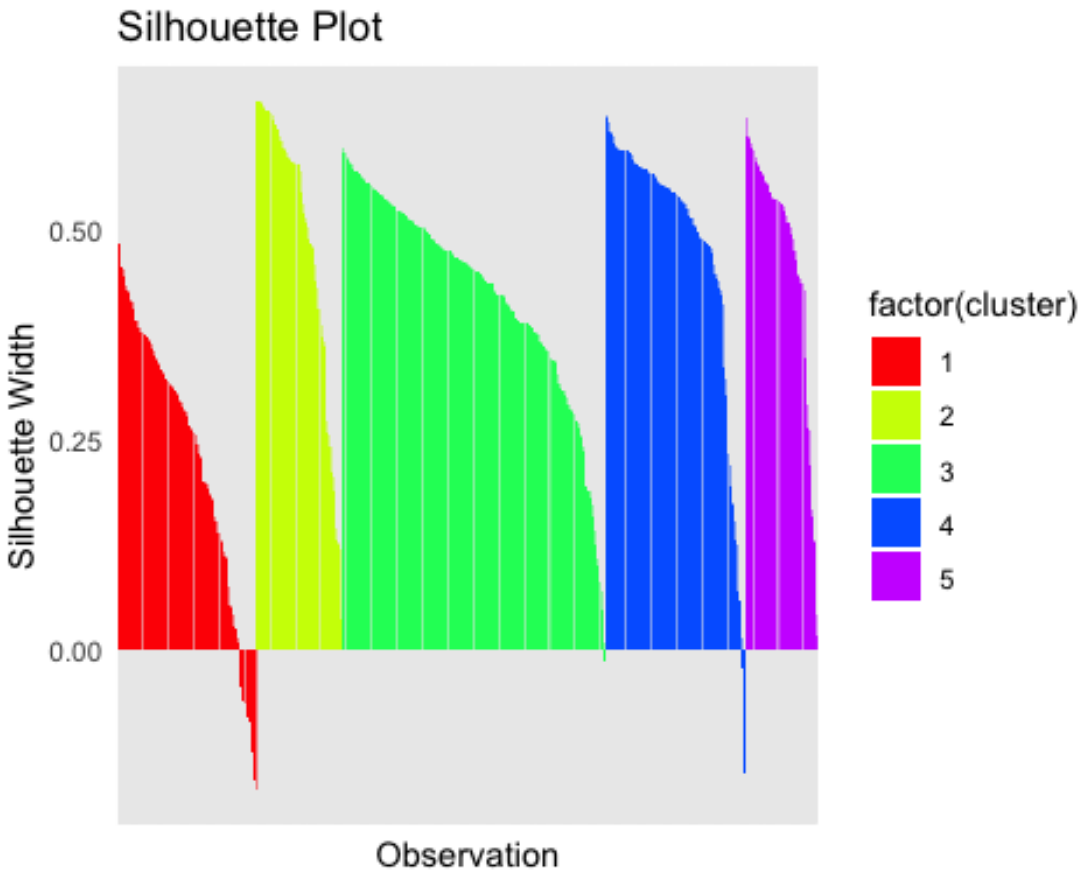
```
points(2:10, larghezza_media_e, col = "blue", pch = 19)
legend("topright", legend = c("manhattan", "euclidean"), col = c("red",
"blue"), pch =19)
```



Il numero di cluster migliore sembra essere 5. La distanza manhattan è migliore della distanza euclidea a parità di numero di cluster, come si vede dal grafico.

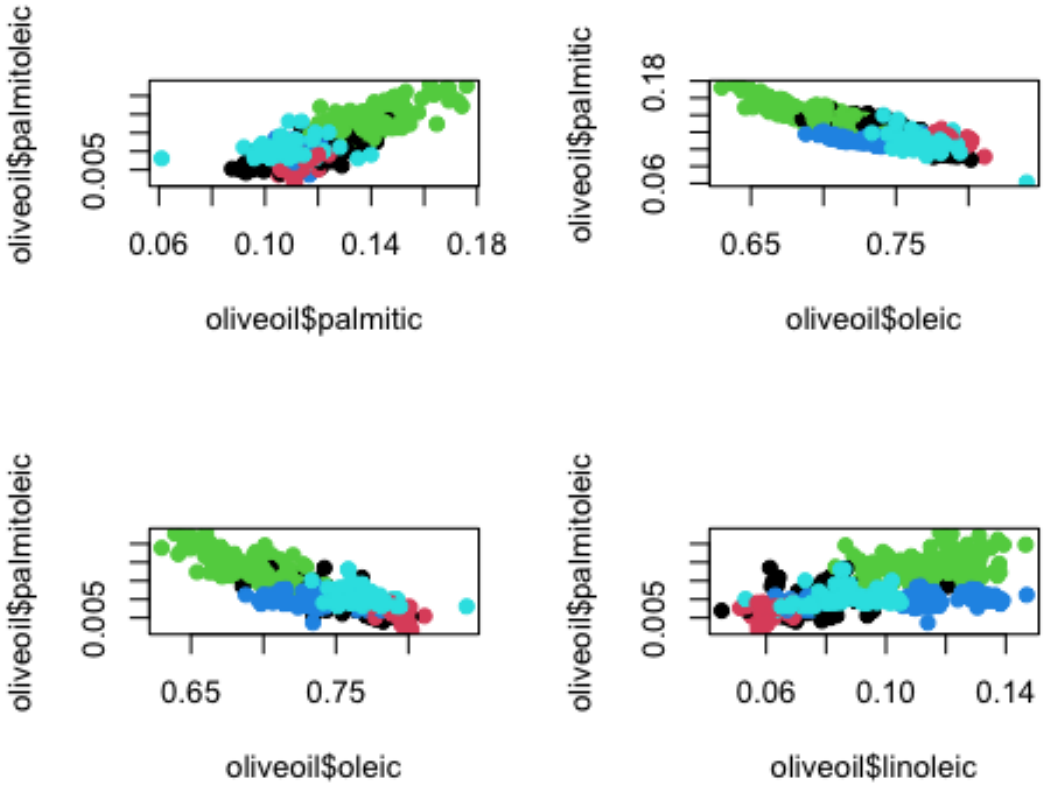
```
set.seed(17)
pam.out<-pam(oliveoil[,3:10], 5, metric="manhattan", stand=TRUE, nstart = 10)
str(pam.out)

## List of 10
##  $ medoids   : num [1:5, 1:8] 0.127 0.109 0.142 0.106 0.103 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:8] "palmitic" "palmitoleic" "stearic" "oleic" ...
##  $ id.med    : int [1:5] 51 438 239 343 555
##  $ clustering: int [1:572] 1 1 2 1 1 1 1 1 1 2 ...
##  $ objective : Named num [1:2] 5.5 4.08
##   ..- attr(*, "names")= chr [1:2] "build" "swap"
##  $ isolation : Factor w/ 3 levels "no","L","L*": 1 1 1 1 1
##   ..- attr(*, "names")= chr [1:5] "1" "2" "3" "4" ...
##  $ clusinfo  : num [1:5, 1:5] 113 70 215 115 59 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   .. ..$ : NULL
##   .. ..$ : chr [1:5] "size" "max_diss" "av_diss" "diameter" ...
##  $ silinfo   :List of 3
##   ..$ widths         : num [1:572, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:572] "78" "51" "293" "74" ...
##   .. .. ..$ : chr [1:3] "cluster" "neighbor" "sil_width"
##   ..$ clus.avg.widths: num [1:5] 0.228 0.483 0.424 0.478 0.468
##   ..$ avg.width      : num 0.408
##  $ diss      : NULL
##  $ call      : language pam(x = oliveoil[, 3:10], k = 5, metric =
"manhattan", nstart = 10, stand = TRUE)
##  $ data      : num [1:572, 1:8] -1.089 -0.999 -2.217 -1.84 -1.254 ...
##   ..- attr(*, "scaled:center")= num [1:8] 0.1234 0.0127 0.023 0.7318
0.0982 ...
##   ..- attr(*, "scaled:scale")= num [1:8] 0.01452 0.00452 0.00281 0.03393
0.02141 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:8] "palmitic" "palmitoleic" "stearic" "oleic" ...
##  - attr(*, "class")= chr [1:2] "pam" "partition"

# GRAFICO
sil_df <- as.data.frame(silhouette(pam.out)[, 1:3])
colnames(sil_df) <- c("cluster", "neighbor", "sil_width")
sil_df$obs <- 1:nrow(sil_df)
sil_df <- sil_df[order(sil_df$cluster, -sil_df$sil_width),]
sil_df$obs_ordered <- factor(sil_df$obs, levels = sil_df$obs)

ggplot(sil_df, aes(x = obs_ordered, y = sil_width, fill = factor(cluster))) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = rainbow(5)) +
  labs(title = "Silhouette Plot", x = "Observation", y = "Silhouette Width")
+
  theme_minimal() +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```

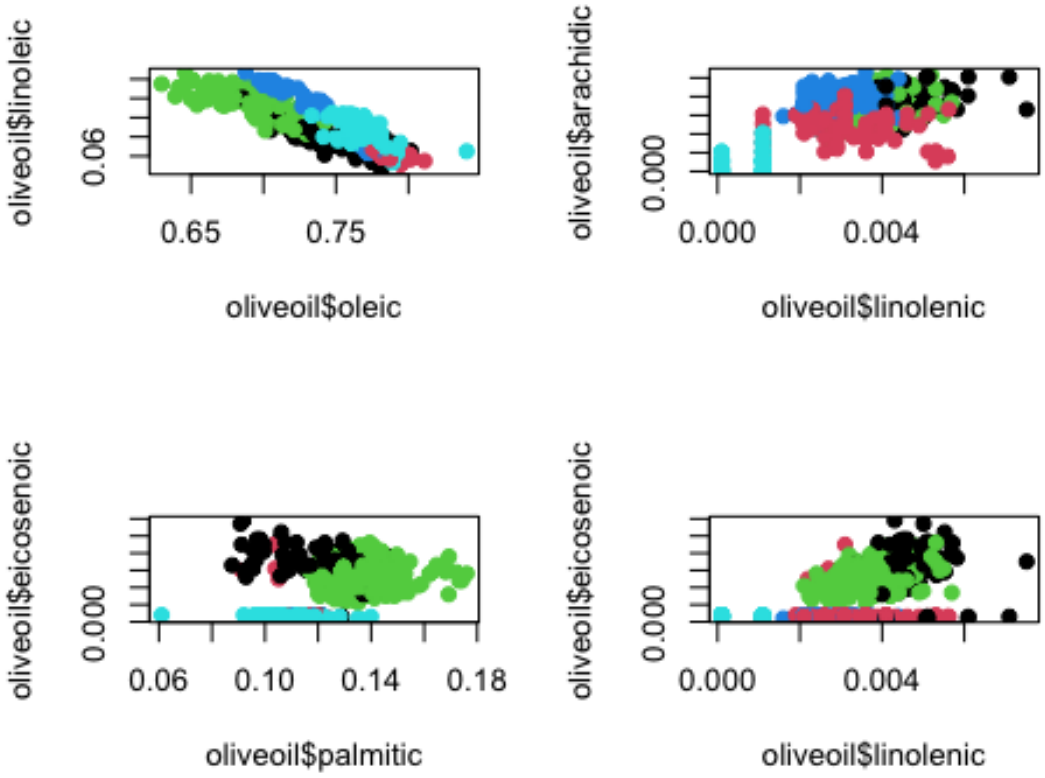


Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

```
par(mfrow=c(2,2))

# palmitic palmitoleic
plot(oliveoil$palmitic, oliveoil$palmitoleic, col = pam.out$cluster, pch =
19)

# oleic palmitic
plot(oliveoil$oleic, oliveoil$palmitic, col = pam.out$cluster, pch = 19)

# oleic palmitoleic
plot(oliveoil$oleic, oliveoil$palmitoleic, col = pam.out$cluster, pch = 19)

# linoleic palmitoleic
plot(oliveoil$linoleic, oliveoil$palmitoleic, col = pam.out$cluster, pch =
19)
```




```r
# linoleic oleic
plot(oliveoil$oleic, oliveoil$linoleic, col = pam.out$cluster, pch = 19)

# arachidic linolenic
plot(oliveoil$linolenic, oliveoil$arachidic, col = pam.out$cluster, pch = 19)

# eicosenoic palmitic
plot(oliveoil$palmitic, oliveoil$eicosenoic, col = pam.out$cluster, pch = 19)

# eicosenoic linolenic
plot(oliveoil$linolenic, oliveoil$eicosenoic, col = pam.out$cluster, pch =
19)
```

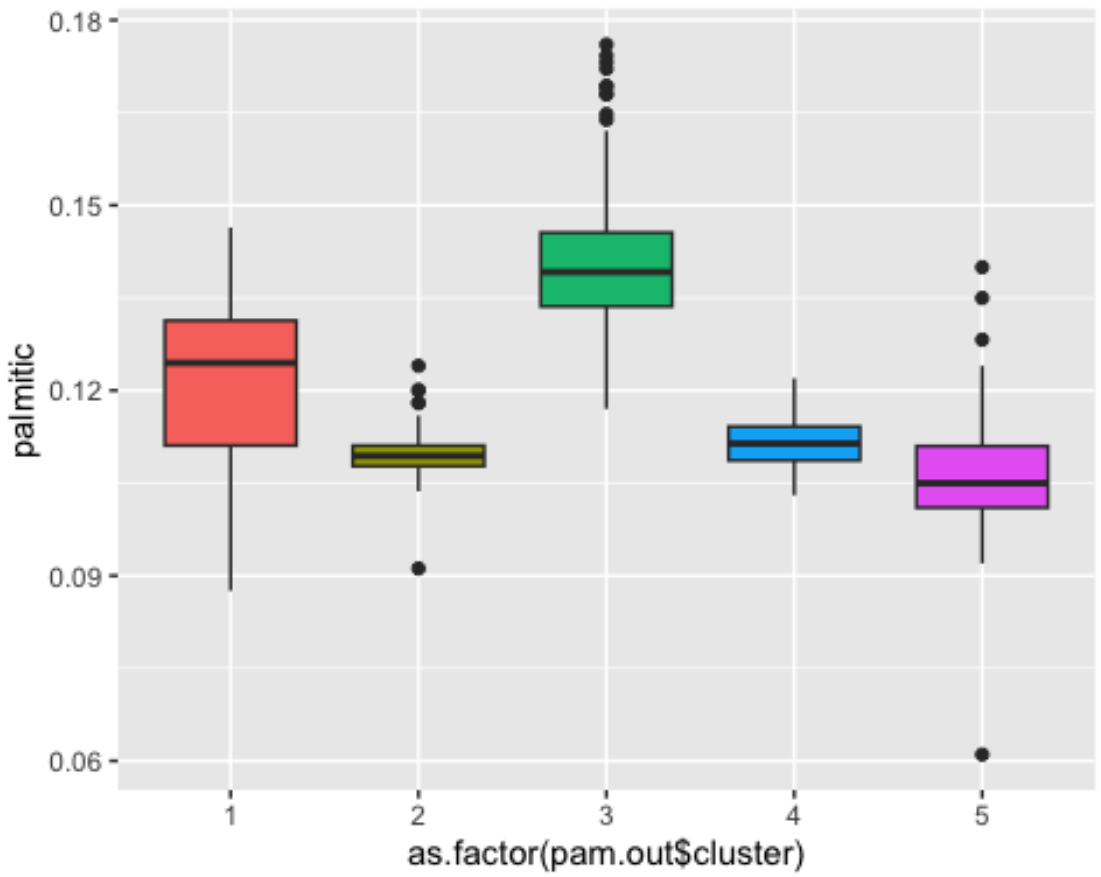



```
par(mfrow=c(1,1))
```

Anche in questo caso i grafici in cui si nota meglio la divisione in cluster sono quelli con l'acido oleico, in quanto questo è altamente correlato con l'acido palmitico e palmitoleico. E ha i valori più alti nel dataset.

Variabile palmitic

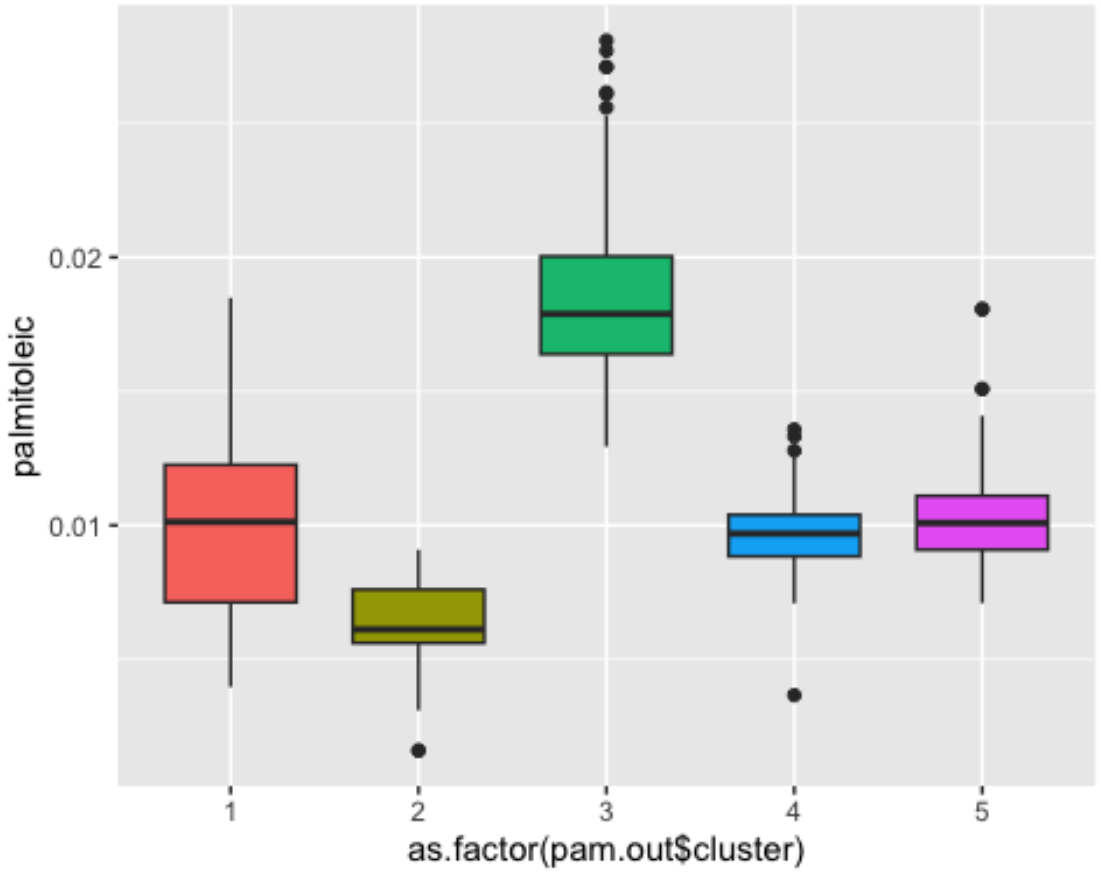
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = palmitic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Si nota una buona differenza tra le mediane dei diversi cluster e una bassa varianza all'interno degli stessi. Nel cluster 3 sono raggrupati gli oli con le percentuali maggiori di acido palmitico.

Variabile palmitoleic

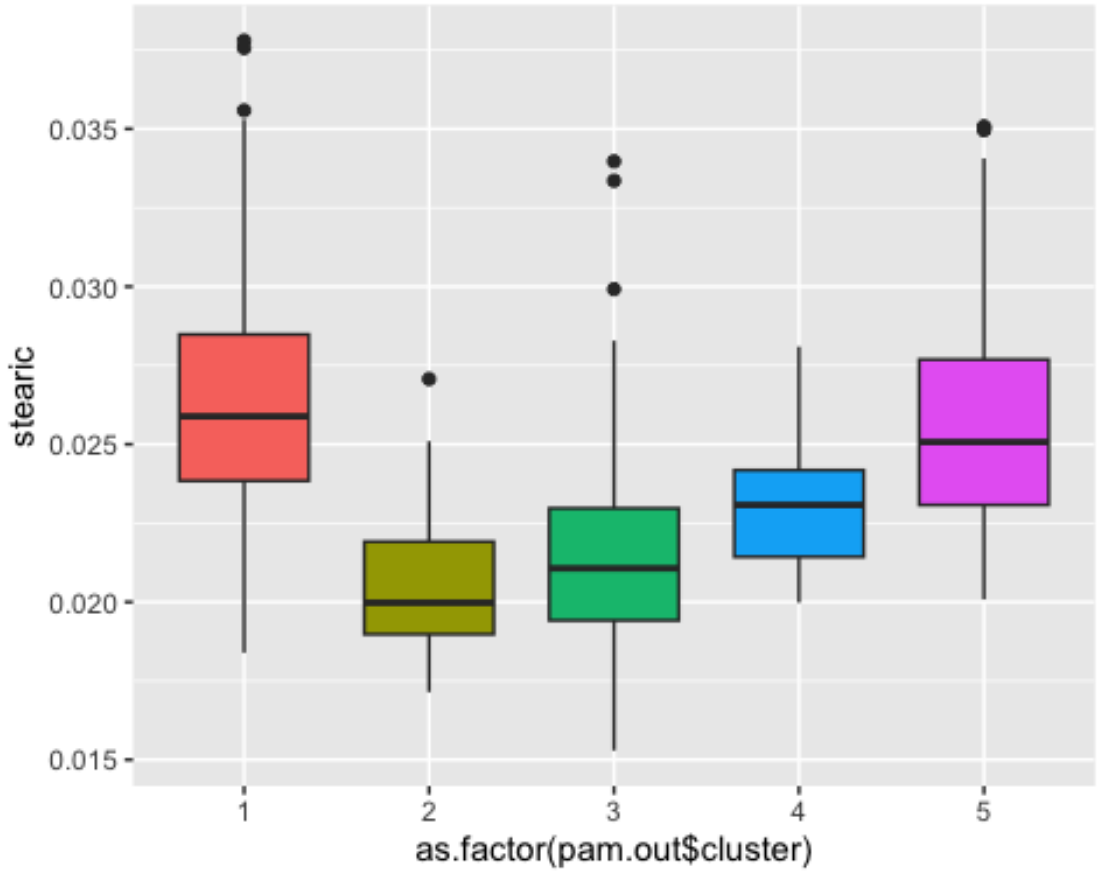
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = palmitoleic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Le osservazioni sono analoghe a quelle relattive all'acido palmitico, il cluster che contiene gli oli con maggior acido palmitoleico è lo stesso che contiene gli oli con il maggior acido palmitico.

Variabile stearic

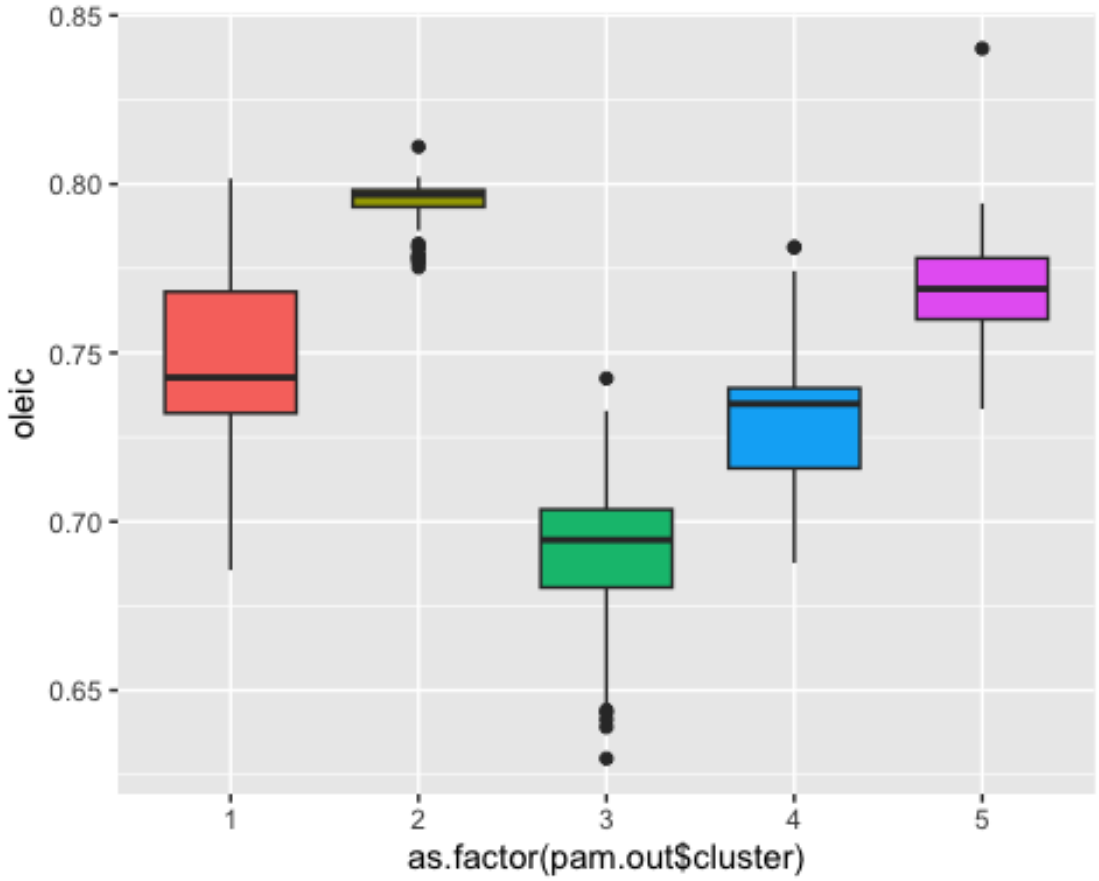
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = stearic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Rispetto ai boxplot ottenuti col metodo k-means osserviamo dai grafici una devianza tra i gruppi maggiore

Variabile oleic

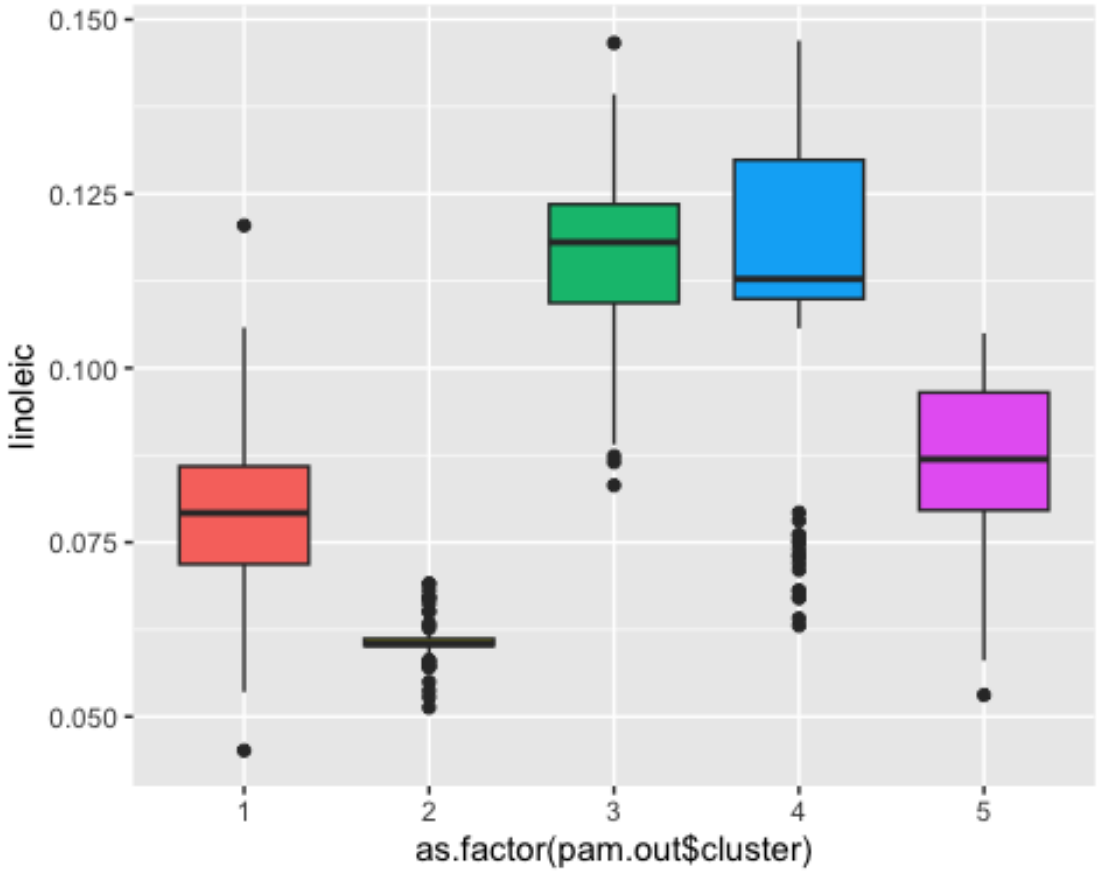
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = oleic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Osserviamo una grande variabilità tra i cluster, come ottenuto in k-means.

Variabile linoleic

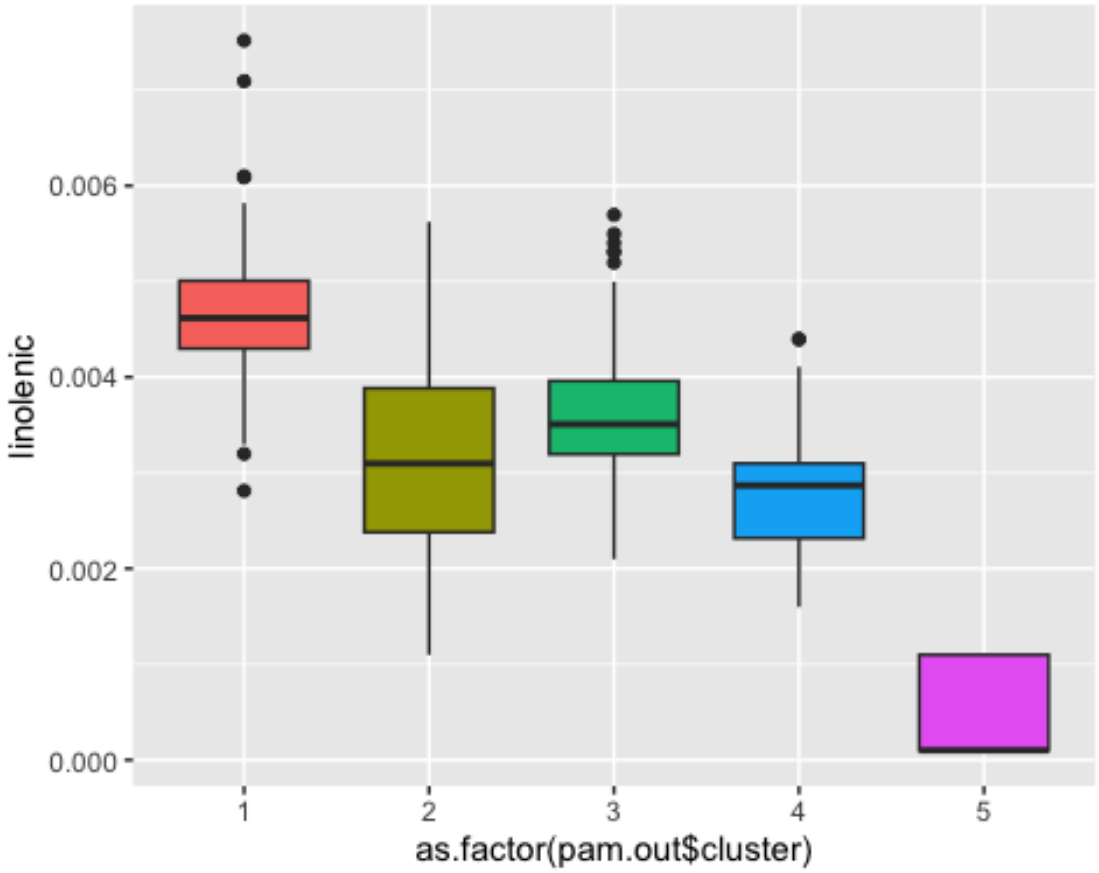
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = linoleic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Grande variabilità tra i cluster. I cluster 3 e 4 hanno una mediana vicina, si osservano tuttavia diversi valori outliers nel cluster 4.

Variabile `linolenic`

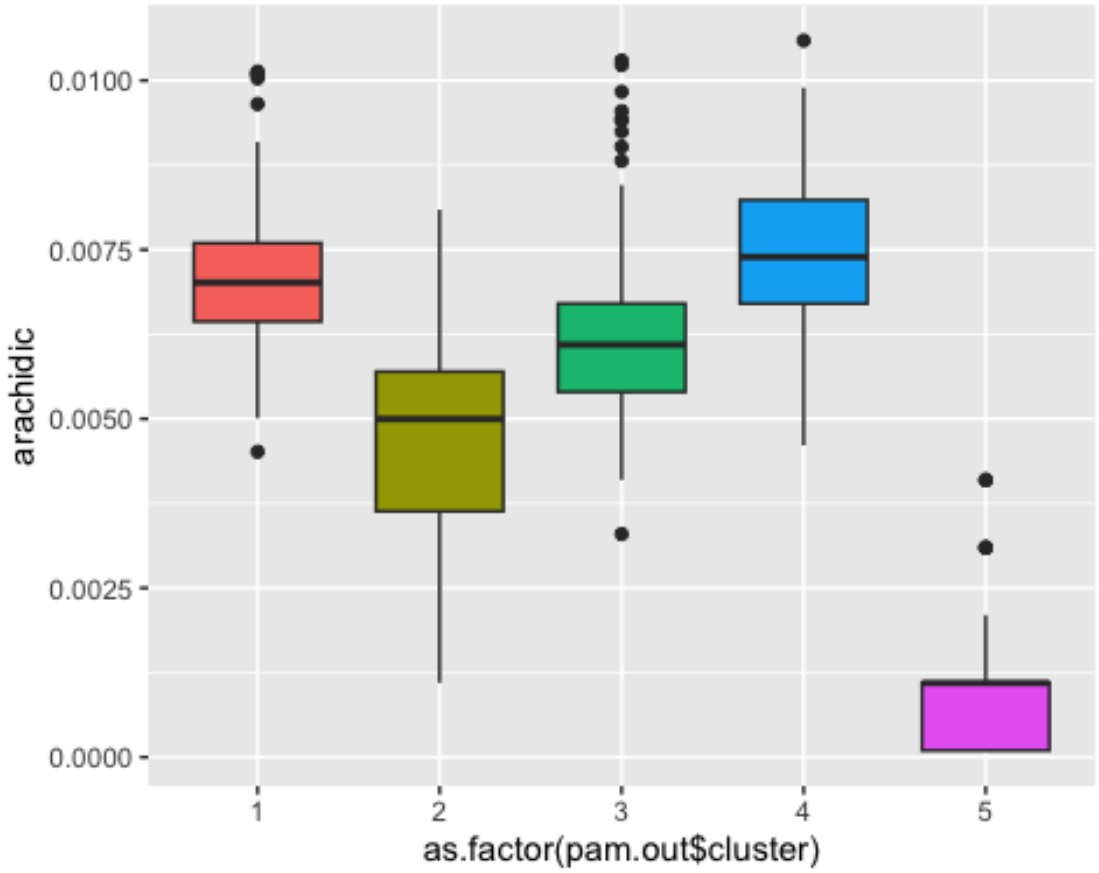
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = linolenic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Anche qui come nel caso dell'acido stearico osserviamo che l'algoritmo PAM rispwtto a k-means individua dei cluster con una variabilità maggiore. In k-means le mediane sono vicine tra di loro in tutti i cluster, con PAM si osserva per esempi come nel cluster 5 vi siano oli con percentuali molto prossime allo zero di acido linolenico e nel cluster 1 oli con percentuali intorno allo 0.005. La presenza di acido linolenico è molto bassa il massimo è minore di 0.008.

Variabile arachidic

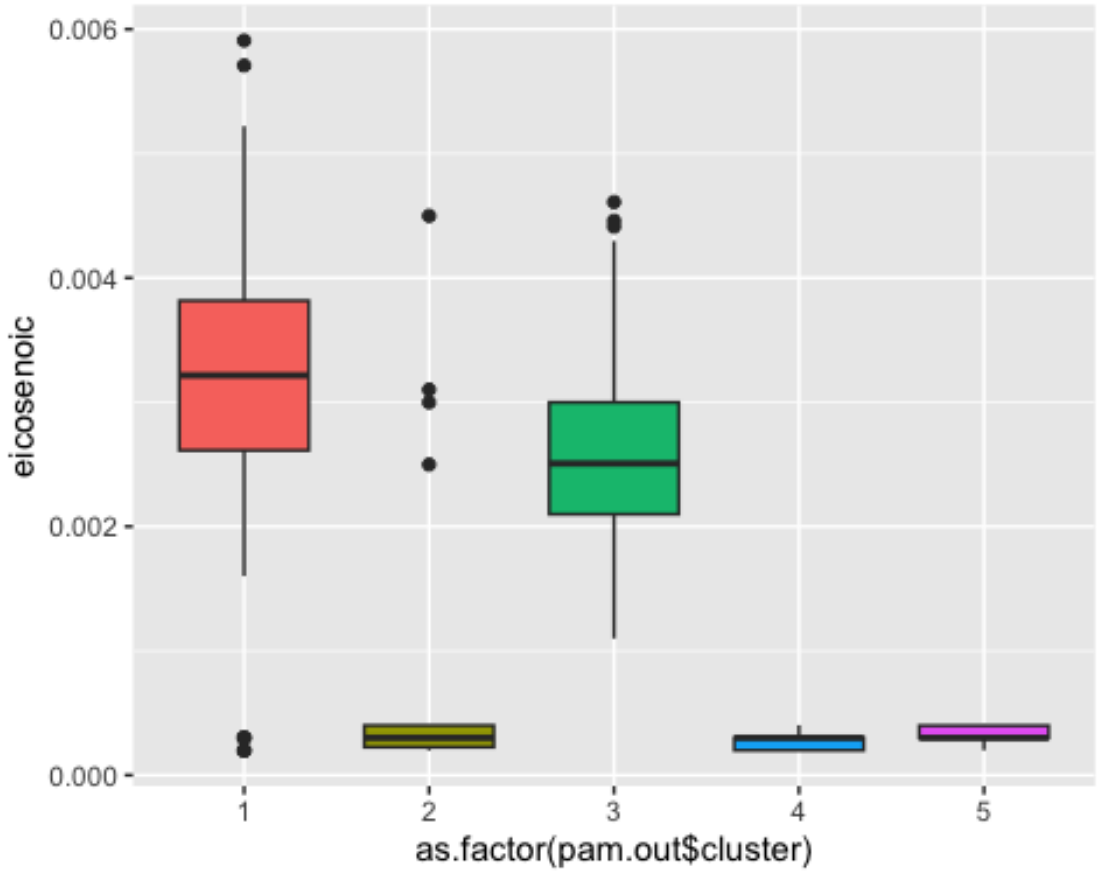
```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = arachidic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Anche per l'acido arachidico si nota una grande variabilità tra i cluster, cluster 5 contiene oli con percentali prossime allo zero.

Variabile eicosenoic

```
ggplot(oliveoil, aes(x = as.factor(pam.out$cluster), y = eicosenoic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```

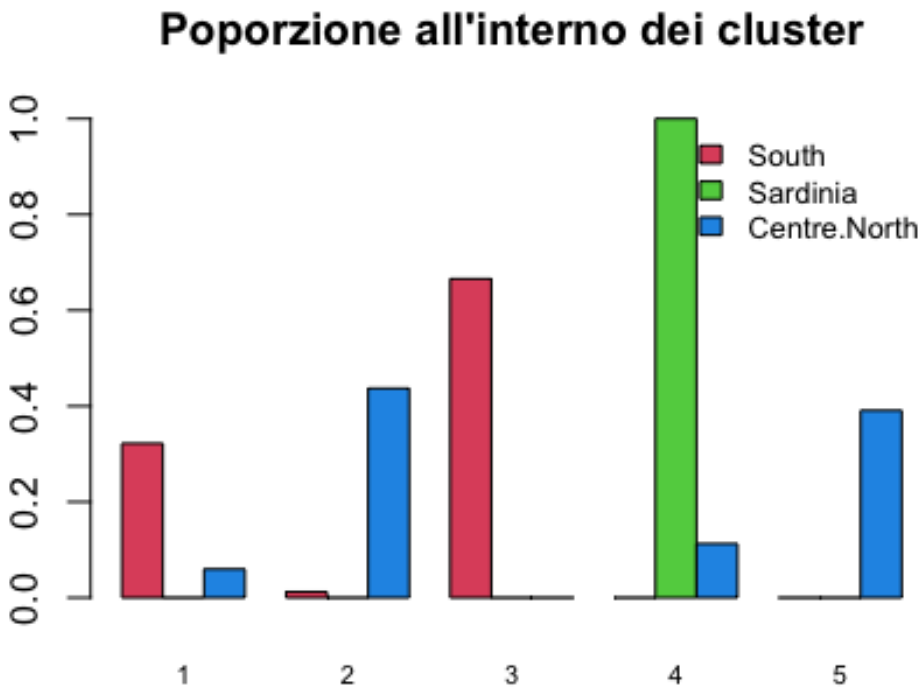


Cluster 1 e 3 sono gli unici nei quali la mediana si discosta significativamente dallo zero.

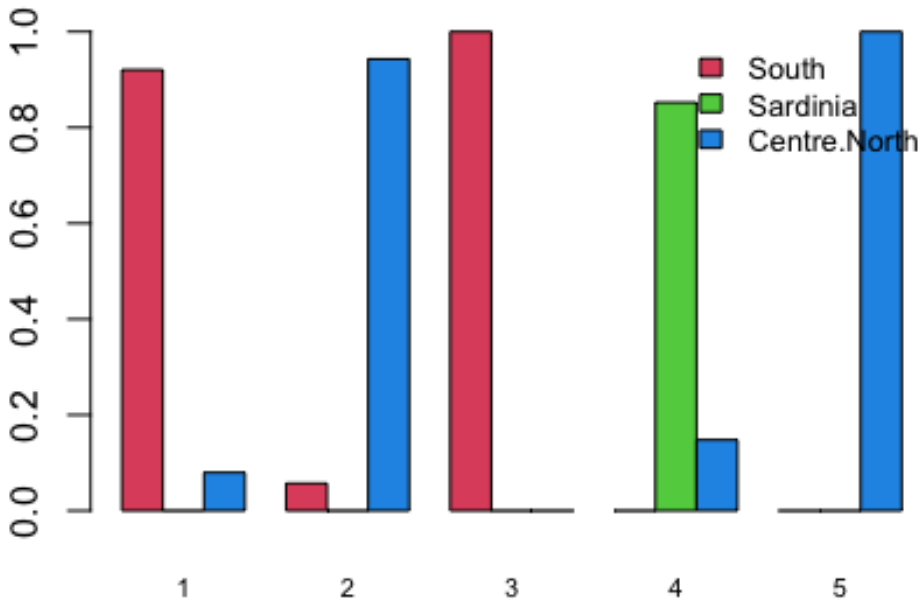
Considerazioni generali sui boxplot: si nota che la divisione in cluster dell'algoritmo PAM fornisce più informazioni dei cluster prodotti da k-means per le variabili stearico, arachidico e linolenico. In k-means le le mediane dei cluster di queste variabili sono tutte vicine Mentre i cluster di Pam hanno una devianza tra i gruppi significativa.

Variabile macro.area

```
prop.table(table(oliveoil$macro.area, pam.out$cluster),1)

##
##                          1          2          3          4          5
##   South         0.32198142 0.01238390 0.66563467 0.00000000 0.00000000
##   Sardinia      0.00000000 0.00000000 0.00000000 1.00000000 0.00000000
##   Centre.North 0.05960265 0.43708609 0.00000000 0.11258278 0.39072848

barplot(prop.table(table(oliveoil$macro.area, pam.out$cluster),1), beside =
T, legend = F, main = "Poporzione all'interno dei cluster", col = 2:4,
cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveoil$macro.area,
pam.out$cluster),1)
), fill = 2:4, cex = 0.8, bty = "n")
```




```
barplot(prop.table(table(oliveoil$macro.area, pam.out$cluster),2), beside =
T, legend = F, main = "", col = 2:4, cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveoil$macro.area,
pam.out$cluster),1)
), fill = 2:4, cex = 0.8, bty = "n")
```



Il cluster 5 è composto esclusivamente da oli del centro nord il cluster 4 è composto all'80% circa da oli della Sardegna e per il resto da oli del centro-nord. il cluster 3 è composto esclusivamente da oli del sud. Il cluster 2 è composto per oltre il 90% da oli del centro nord e i rimanenti sono oli del sud. Il cluster 1 è composto al 92% da oli del sud e il resto da oli del centro nord.

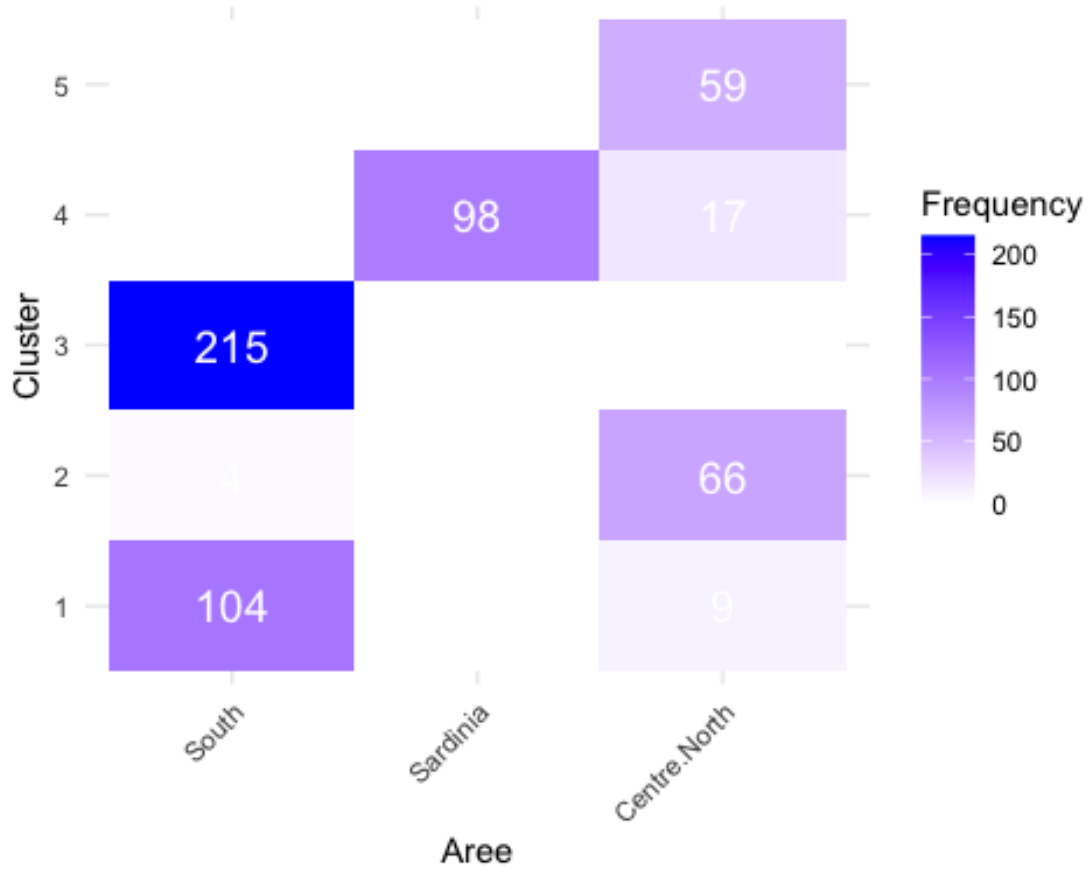
Gli oli del sud si distribuiscono soprattutto nel cluster 3 (66%). I rimanenti si dividono tra cluster 1 (32%) e 2 (12%). Gli oli della Sardegna si trovano esclusivamente nel cluster 4. Gli oli del centro nord sono presenti in tutti i cluster tranne il 3. In particolare al 5% nel cluster 1, al 44% nel cluster 2, 11% nel cluster 4 e 39% nel cluster 5.

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveoil$macro.area, Aree =
pam.out$cluster)

table( Aree = pam.out$cluster, Cluster = oliveoil$macro.area)

##     Cluster
## Aree South Sardinia Centre.North
##    1   104        0            9
##    2     4        0           66
##    3   215        0            0
```

```
##    4     0       98           17
##    5     0        0           59
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Aree, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Aree", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



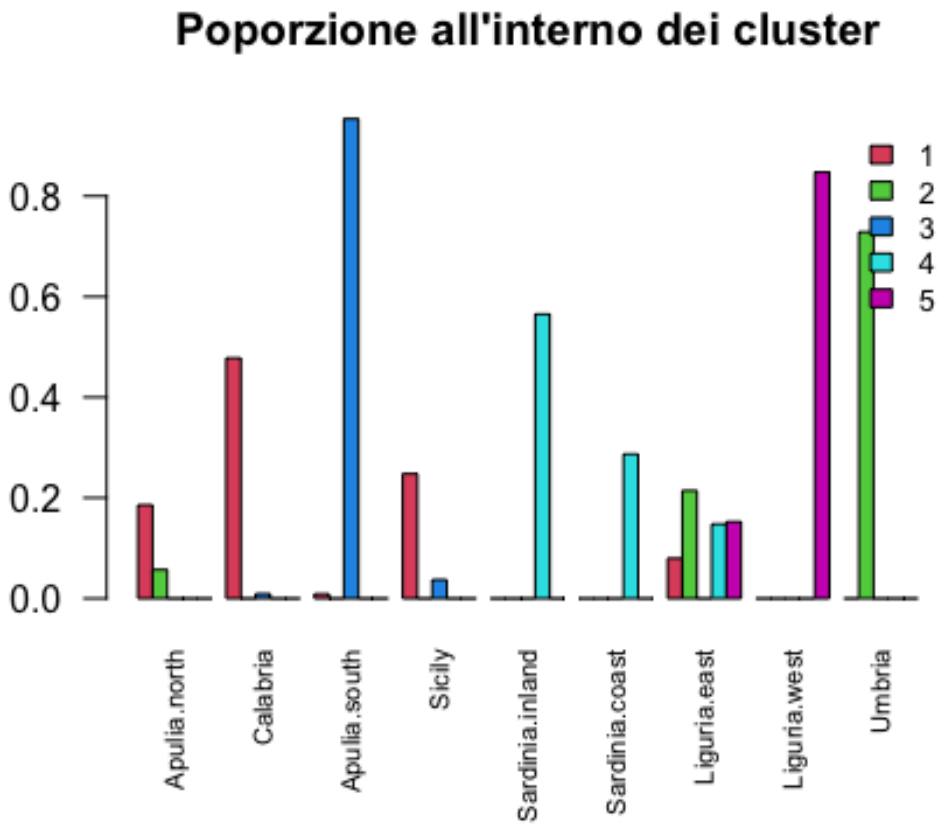
Variabile region

```
prop.table(table(pam.out$cluster, oliveoil$region),1)
```

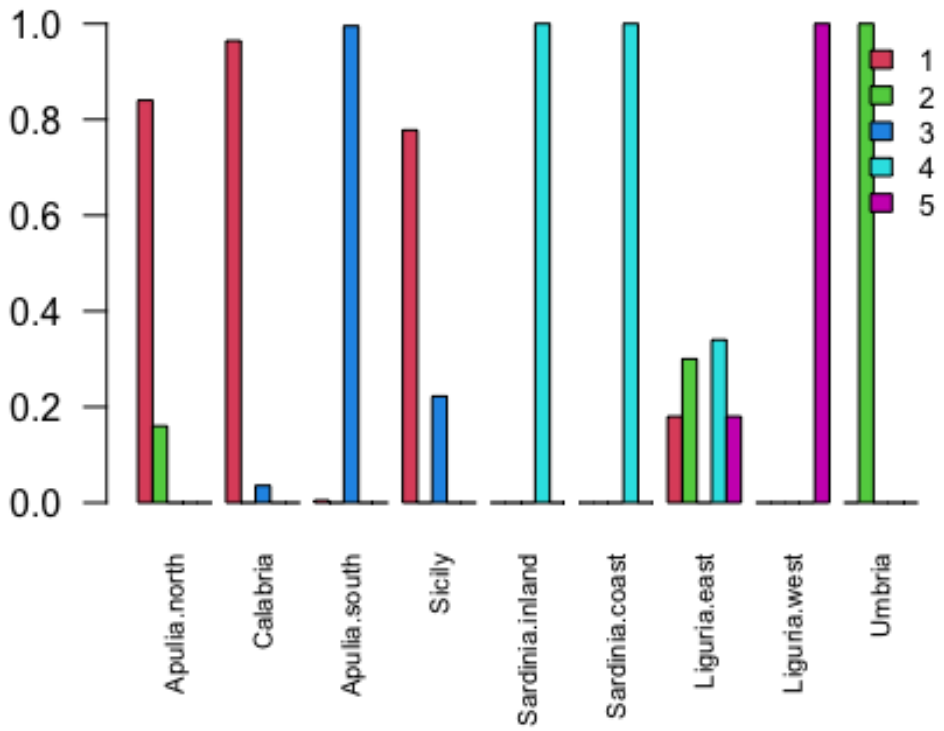
```
##
##      Apulia.north    Calabria Apulia.south      Sicily Sardinia.inland
##   1  0.185840708 0.477876106  0.008849558 0.247787611     0.000000000
##   2  0.057142857 0.000000000  0.000000000 0.000000000     0.000000000
##   3  0.000000000 0.009302326  0.953488372 0.037209302     0.000000000
##   4  0.000000000 0.000000000  0.000000000 0.000000000     0.565217391
##   5  0.000000000 0.000000000  0.000000000 0.000000000     0.000000000
##
```

```
##     Sardinia.coast Liguria.east Liguria.west      Umbria
##   1    0.000000000  0.079646018  0.000000000 0.000000000
##   2    0.000000000  0.214285714  0.000000000 0.728571429
##   3    0.000000000  0.000000000  0.000000000 0.000000000
##   4    0.286956522  0.147826087  0.000000000 0.000000000
##   5    0.000000000  0.152542373  0.847457627 0.000000000
```

```
barplot(prop.table(table(pam.out$cluster, oliveoil$region),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:6, cex.names
= 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(pam.out$cluster,
oliveoil$region),1)), fill = 2:6, cex = 0.8, bty = "n")
```




```
barplot(prop.table(table(pam.out$cluster, oliveoil$region),2), beside = T,
legend = F, main = "", col = 2:6, cex.names = 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(pam.out$cluster,
oliveoil$region),1)), fill = 2:6, cex = 0.8, bty = "n")
```



Gli oli della puglia del nord si trovano per l'84% nel cluster 1 e per il 16% Gli oli della calabria si trovano al 96 % nel cluster 1 e i rimanenti nel cluster 3. Gli oli della puglia del sud al 99% nel cluster 3. Gli oli della Sardegna Inland e Coast finiscono entrambi nel cluster 4 al 100% Gli oli della liguria dell est sono divisi tra cluster 1 (18%), cluster 2 (30%), cluster 4 (34%) e 5 (18%) solo il cluster 3 non presenta oli della liguria est. Gli oli della liguria ovest si trovano tutti nel cluster 5. Gli oli dell'umbria si trovano tutti nel cluster 2.

Il cluster 1 è composto al 19% da oli della puglia del nord al 47% da oli della calabria e al 24% da oli della sicilia, vi è circa un 1% di oli della liguria dell'est. IL cluster 2 è composto per il 5% da oli della puglia del nord al 21% da oli della liguria dell'est e al 73% da oli dell'Umbria. Il cluster 3 é composto al 95% da oli della puglia del sud, 3% Sicilia e 2% Calabria. Il cluster 4 è composto al 57% da Oli della Sardegna inland, al 29% da oli della Sardegna coast e per il resto da oli della liguria dell'est. Il cluster 5 è composto al 15% da oli della liguria dell'est e all'85% da oli della liguria dell'ovest.

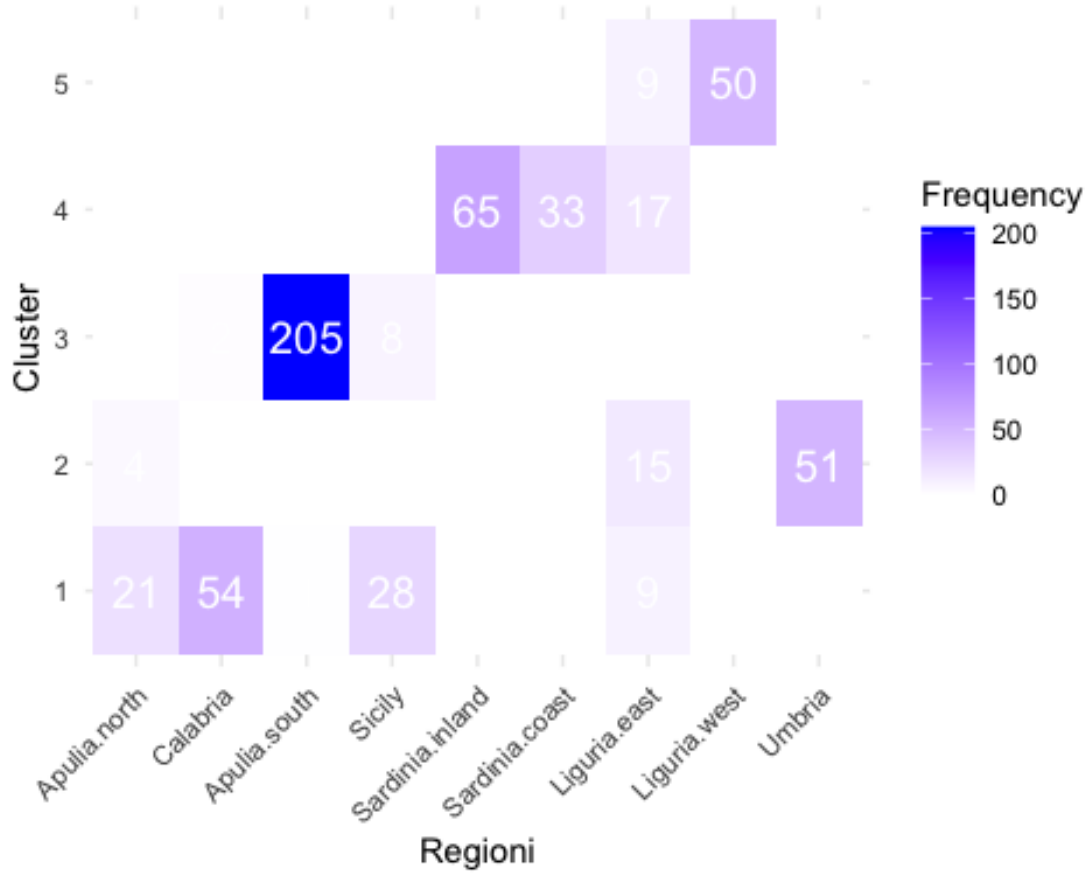
Si puòaffermare che gli oli della liguria dell'est si trovano in 4 cluster su 5.

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveoil$region, Regioni =
pam.out$cluster)

table(Regioni = pam.out$cluster, Cluster = oliveoil$region)
```

```
##         Cluster
## Regioni Apulia.north Calabria Apulia.south Sicily Sardinia.inland
##       1           21       54            1     28               0
##       2            4        0            0      0               0
##       3            0        2          205      8               0
##       4            0        0            0      0              65
##       5            0        0            0      0               0
##         Cluster
## Regioni Sardinia.coast Liguria.east Liguria.west Umbria
##       1              0            9            0      0
##       2              0           15            0     51
##       3              0            0            0      0
##       4             33           17            0      0
##       5              0            9           50      0
```

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Regioni, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Regioni", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

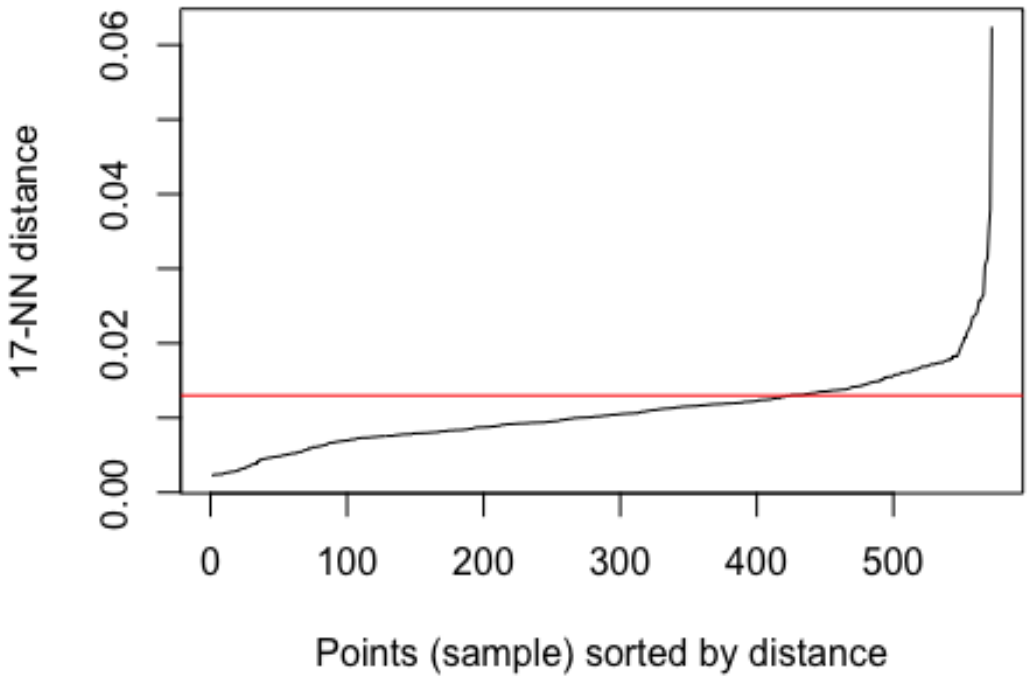


DB SCAN

DBSCAN è un algoritmo di clustering basato sulla densità. L'algoritmo prende in input una matrice numerica di dati, un numero ε e un intero Q. Dove Q rappresenta il minimo numeo di punti neccessari a formare un cluster e ε rappresenta il raggio di un intorno. L'algoritmo opera nel seguente modo: 1. Per ogni punto calcola gli intorni di raggio ε e identifica i centri che hanno un numero > Q di punti nel proprio interno. 2. Trova le componenti connesse dei centri dell'intorno. 3. Associa ogni punto a un cluster se dista < ε dal centro, i punti esclusi sono noise.

Con kNNdistplot() otteniamo il grafico delle deistanze tra un ponto e il suo 17esimo punto più vicino. Sceglaimo poi una distanza (raggio) con cui andremo poi a raggruppare i punti in cluster.

```
kNNdistplot(oliveoil[,3:10], k = 17)
abline(h=0.013, col = "red")
```



Il parametri miglior sono 0.013 come eps, e 18 come minPts

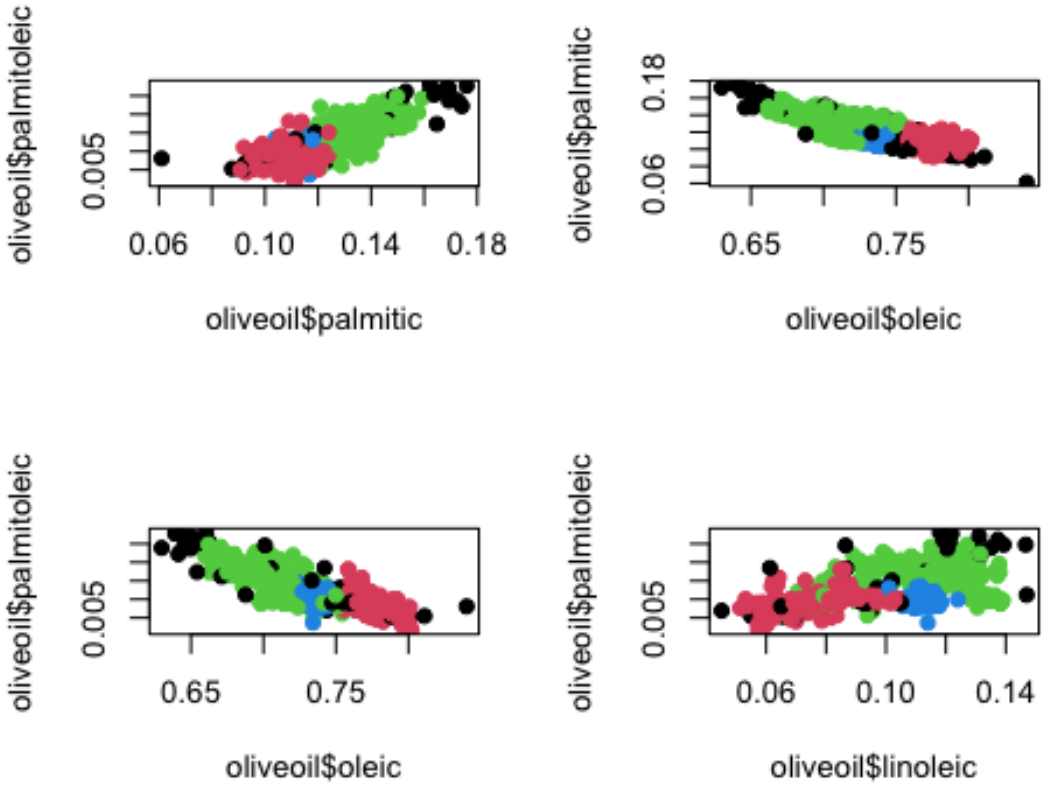
```
set.seed(17)

db.out <- dbscan(oliveoil[,3:10], eps = 0.013, minPts = 18)
str(db.out)
```

```
## List of 5
##  $ cluster     : int [1:572] 1 1 0 1 1 0 0 1 1 1 ...
##  $ eps         : num 0.013
##  $ minPts      : num 18
##  $ dist        : chr "euclidean"
##  $ borderPoints: logi TRUE
##  - attr(*, "class")= chr [1:2] "dbscan_fast" "dbscan"

db.out

## DBSCAN clustering for 572 objects.
## Parameters: eps = 0.013, minPts = 18
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 3 cluster(s) and 40 noise points.
##
##   0   1   2   3
##  40 169 295  68
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

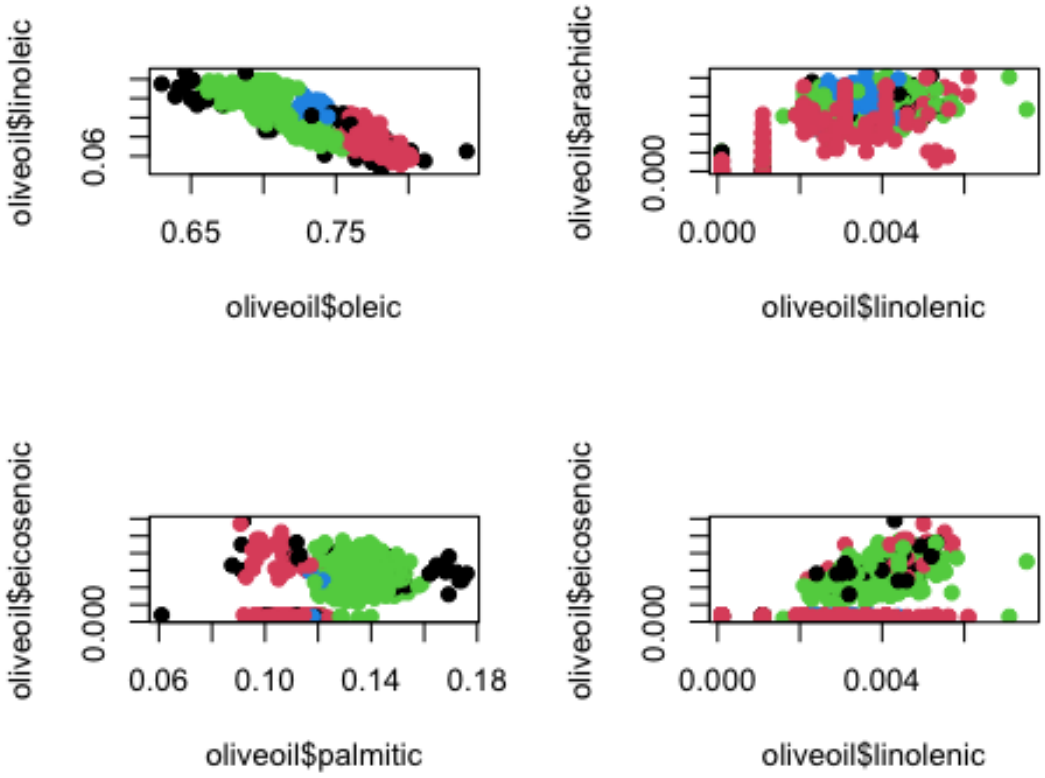
dbscan() ci diviede i dati in cluster, in questo caso 3 con 40 punti di "noise".

Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

```
par(mfrow=c(2,2))

# palmitic palmitoleic
plot(oliveoil$palmitic, oliveoil$palmitoleic, col = db.out$cluster+1, pch =
19)

# oleic palmitic
plot(oliveoil$oleic, oliveoil$palmitic, col = db.out$cluster+1, pch = 19)

# oleic palmitoleic
plot(oliveoil$oleic, oliveoil$palmitoleic, col = db.out$cluster+1, pch = 19)

# linoleic palmitoleic
plot(oliveoil$linoleic, oliveoil$palmitoleic, col = db.out$cluster+1, pch =
19)
```




```r
# linoleic oleic
plot(oliveoil$oleic, oliveoil$linoleic, col = db.out$cluster+1, pch = 19)

# arachidic linolenic
plot(oliveoil$linolenic, oliveoil$arachidic, col = db.out$cluster+1, pch =
19)

# eicosenoic palmitic
plot(oliveoil$palmitic, oliveoil$eicosenoic, col = db.out$cluster+1, pch =
19)

# eicosenoic linolenic
plot(oliveoil$linolenic, oliveoil$eicosenoic, col = db.out$cluster+1, pch =
19)
```

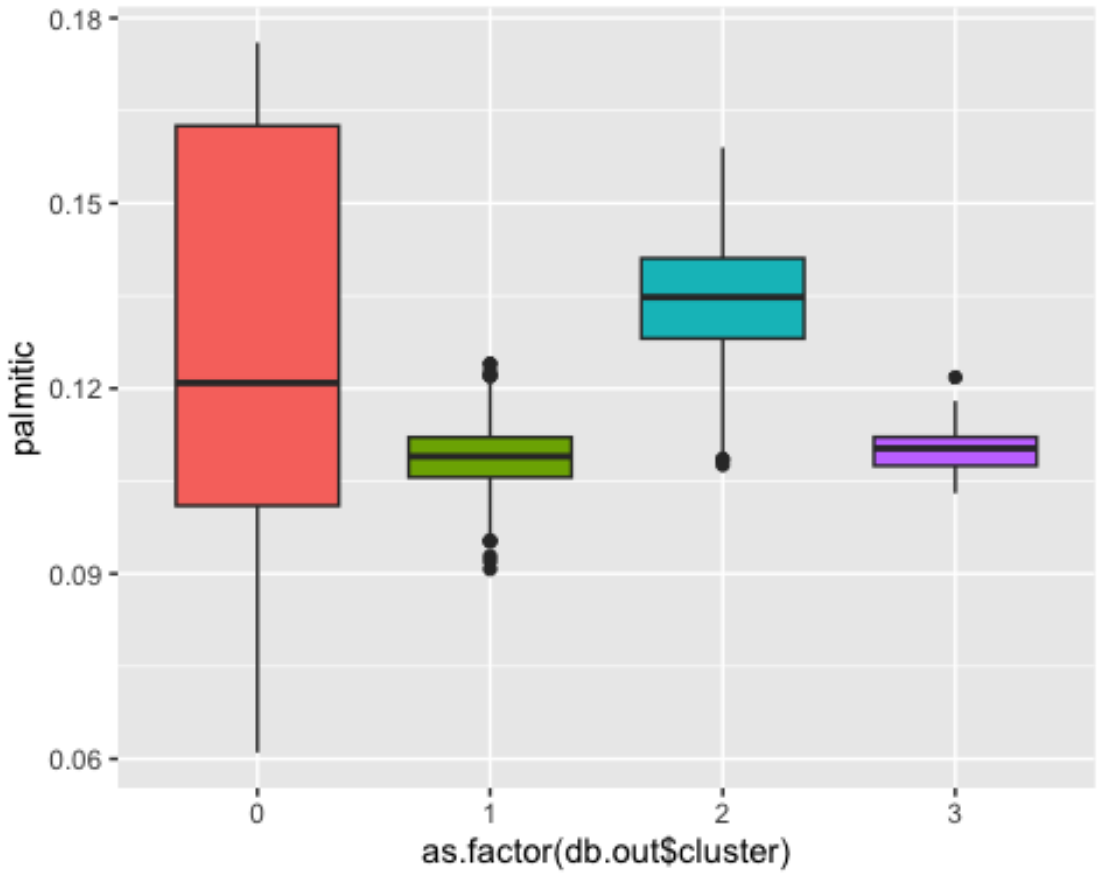



```
par(mfrow=c(1,1))
```

Come nei casi precedenti, gli scatterplot creati con l'acido oleico sono quelli che meglio visualizzano la distribuzione dei cluster.

Variabile palmitic

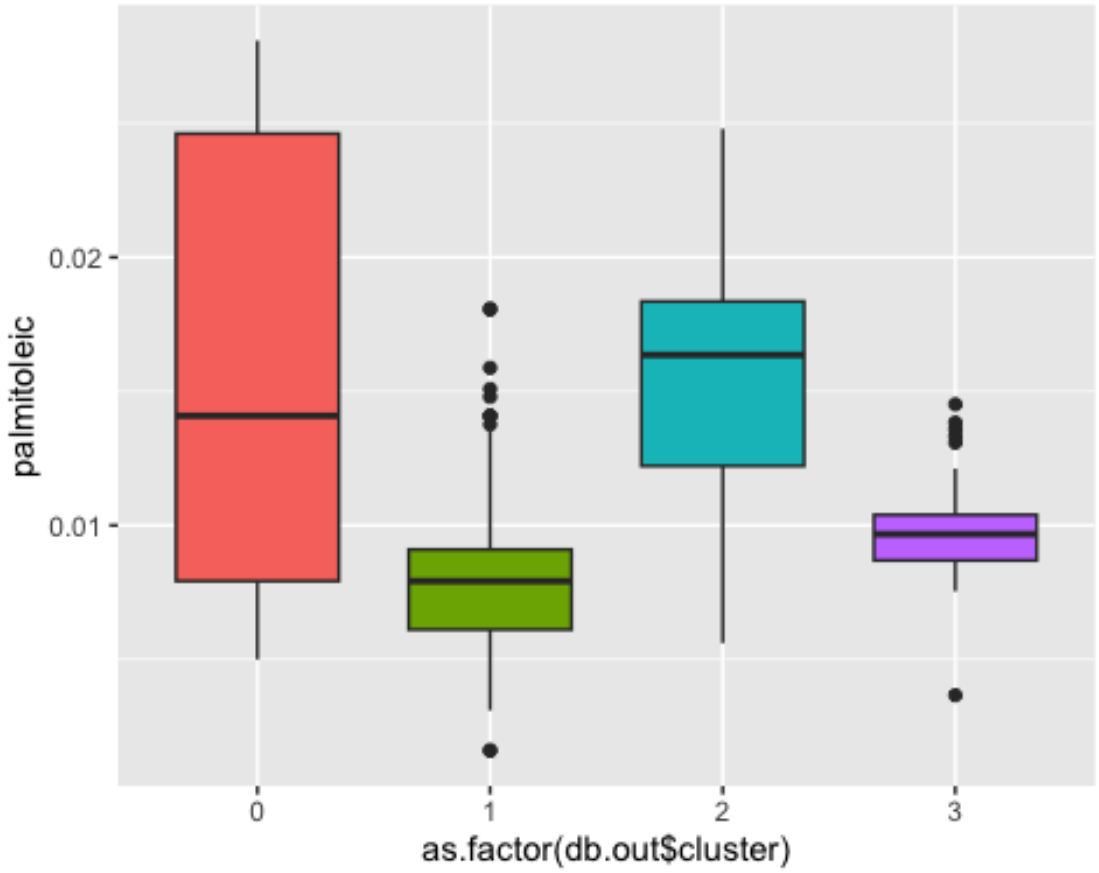
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = palmitic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Si vede suzito che i punti di "noise" sono quelli con il range di valori più ampio, caratteristica che terranno per gran parte dei boxplot. I tre cluster invece hanno in confronto poca varianza tra i valori interni, con il cluster 3 che ha il range più piccolo di valori. Questo è anche aiutato dal fatto che il cluster 3 è quello meno numeroso. Il cluster 2 è in contrasto quello a cui appartengono più tipi di oli ed ha infatti un range di valori più ampio rispetto agli altri due. Tra i gruppi, il cluster 2 è quello con percentuali più alte di acido palmitico.

Variabile palmitoleic

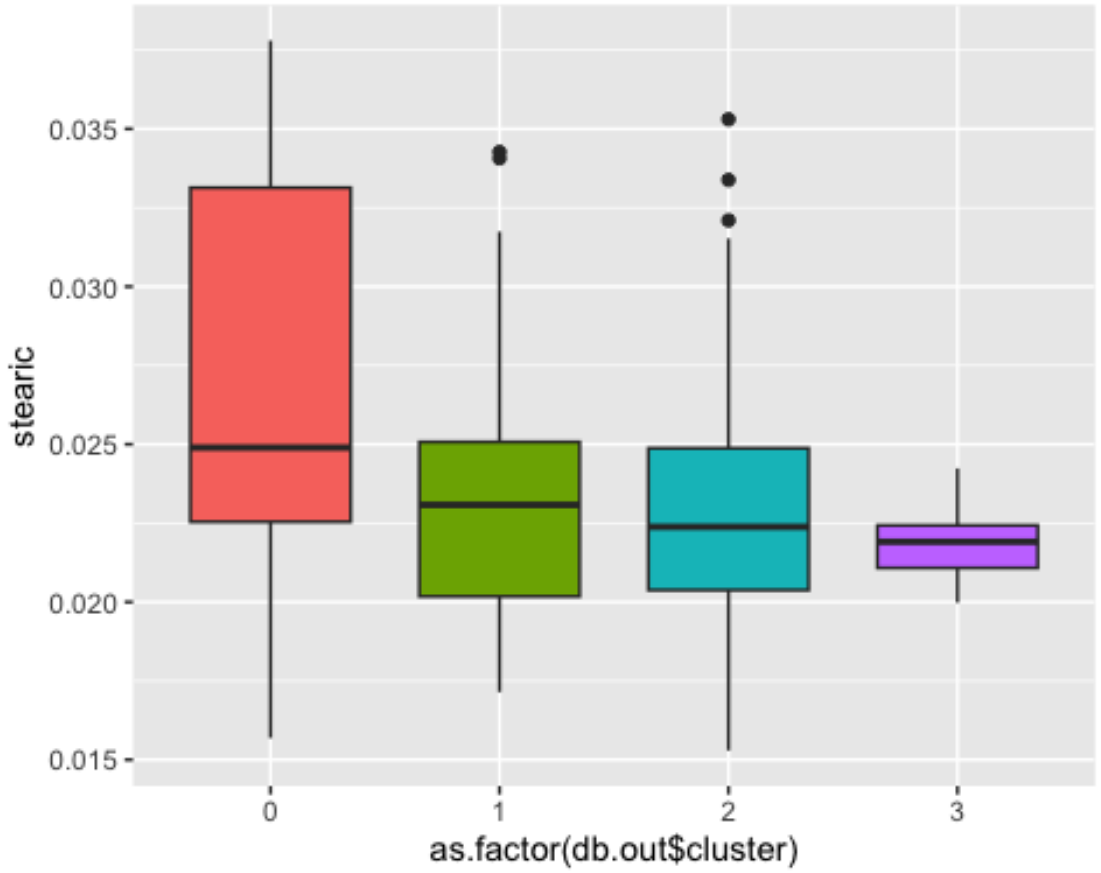
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = palmitoleic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



La distribuzione in cluster di questo acido è sempre vicina alla precedente, essendo i due fortemente correlati. Si vede infatti che il cluster 2 è sempre quello con gli oli aventi percentuali di acido maggiore.

Variabile stearic

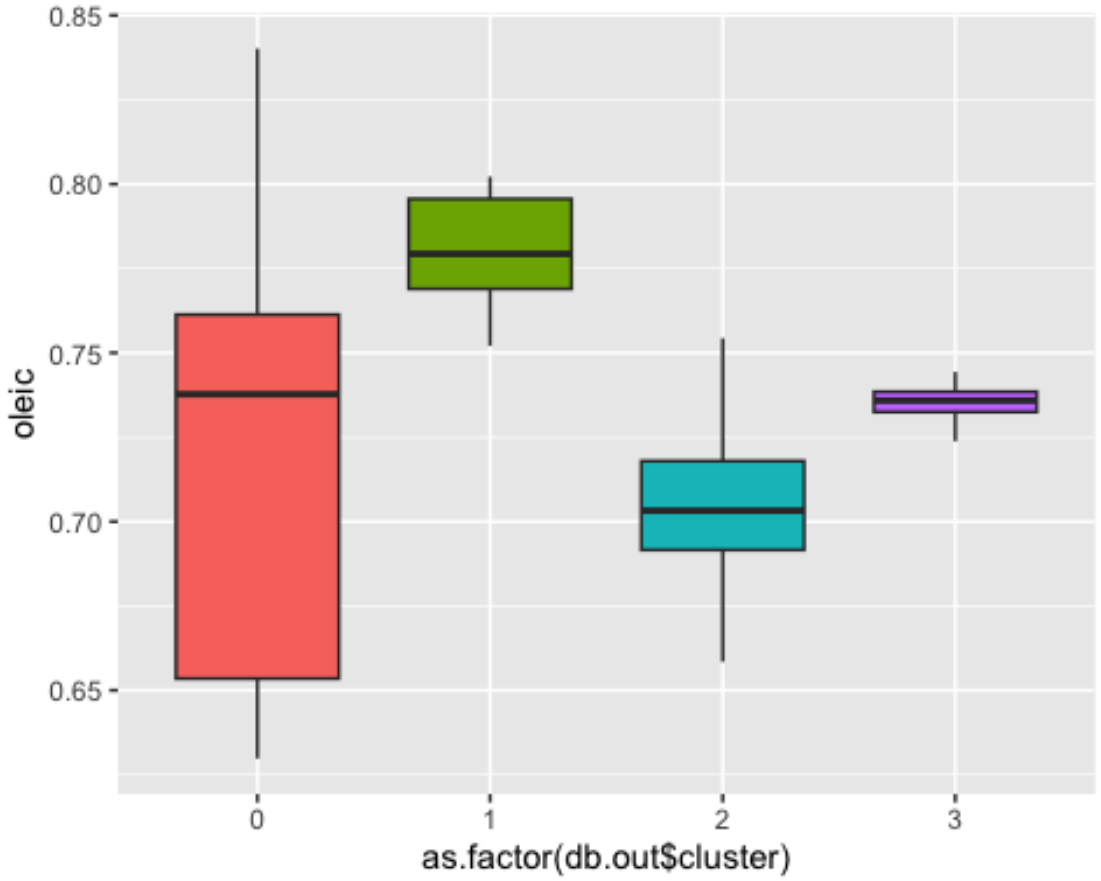
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = stearic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Anche per la distribuzione in cluster con dbscan l'acido stearico, non essendo presente in grandi quantità e non essendo fortemente correlato a nessun acido tra i più presenti, ha distribuzioni simili in tutti i cluster. Si nota che il cluster 3 è quello con la varianza interna minore, sempre probabilmente poichè esse è il meno numeroso.

Variabile oleic

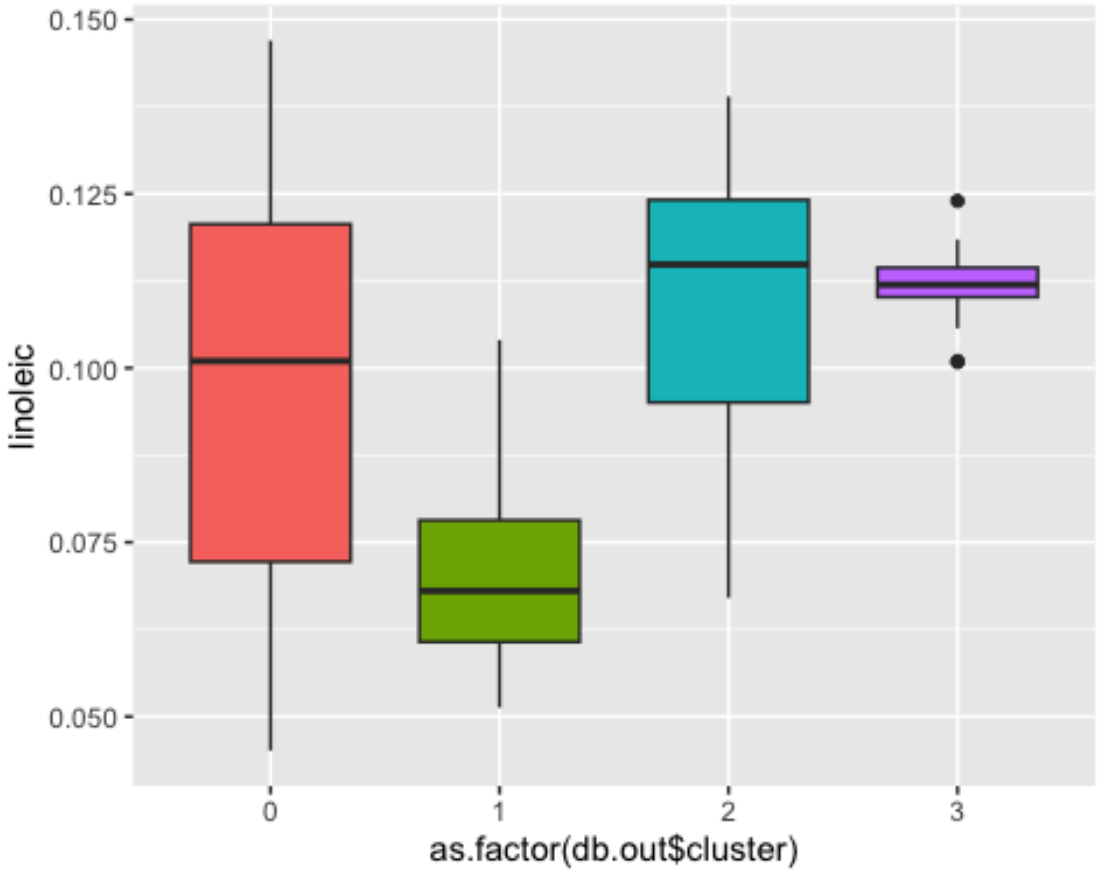
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = oleic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Come per i metodi K-means e PAM, l'acido oleico è quello che presenta differenze più marcate tra i cluster. In questo caso è il primo che contiene oli con percentuali di acido maggiori e il secondo che contiene percentuali minori. Il terzo gruppo rimane essere quello con la varianza interna minore.

Variabile linoleic

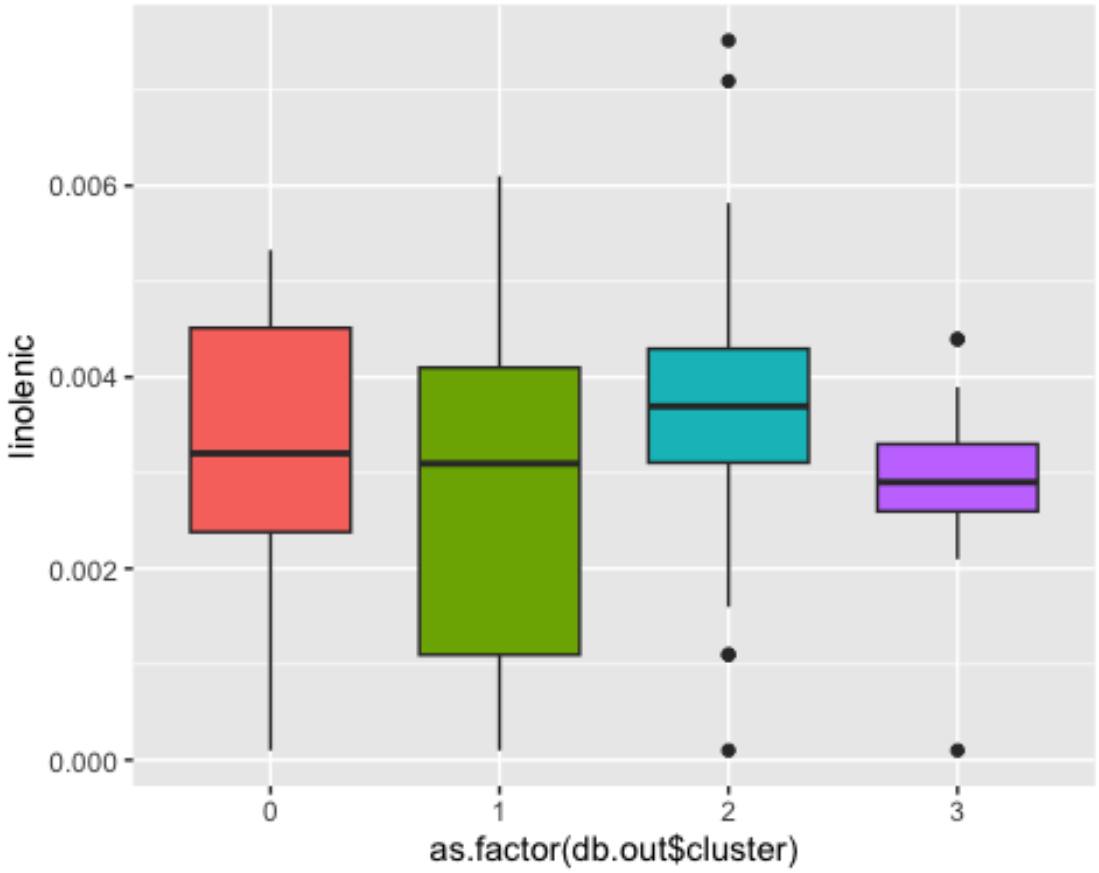
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = linoleic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Si vede che nel caso dell'acido linoleico il primo cluster contiene oli con la percentuale di acido minore mentre i cluster 2 e 3 hanno percentuali maggiori. Questi ultimi hanno inoltre mediane molto vicine anche se il 3 rimane quello con varianza minore.

Variabile `linolenic`

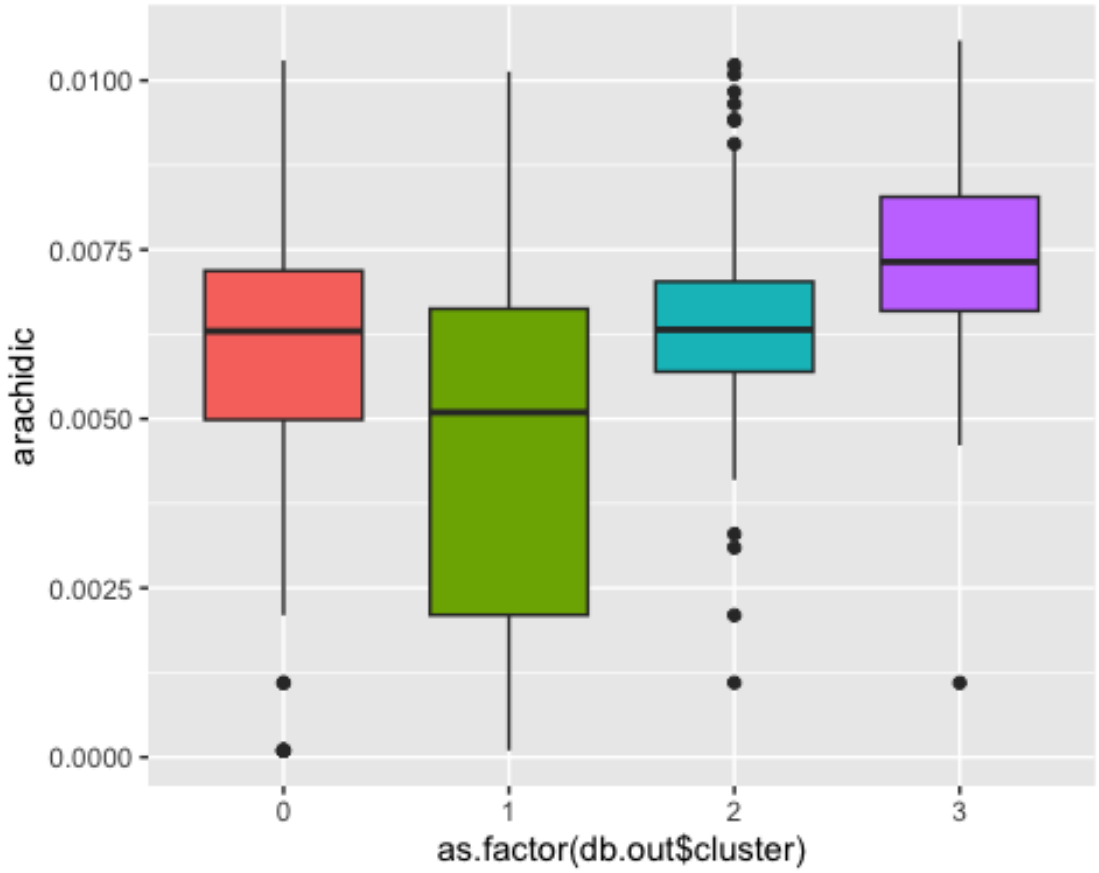
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = linolenic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



In questo caso vediamo che il range dei punti di "noise" è minore del range di valori del cluster 1 e 2. L'acido linolenico è infatti uno degli acidi meno presenti e quindi non andrà ad influire fortemente sulla scelta dei cluster. I cluster sono dunque molto simili per quanto riguarda le loro distribuzioni in riferimento a questo acido.

Variabile arachidic

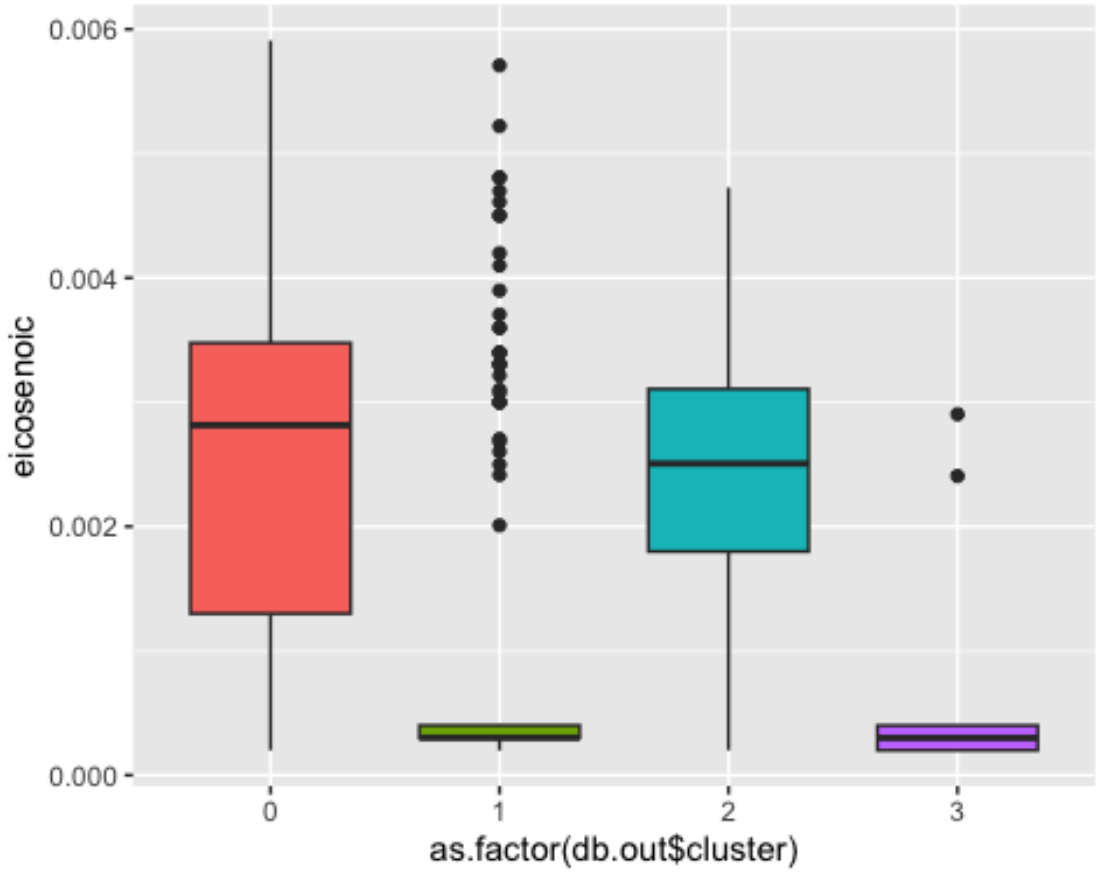
```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = arachidic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Un altro acido presente in percentuali basse all'interno degli oli. Si vede che i cluster hanno varianze interne alte, soprattutto il numero 1, e che il cluster3 è quello che tende ad avere al suo interno oli con percentuale di acido arachidico maggiore.

Variabile eicosenoic

```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = eicosenoic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



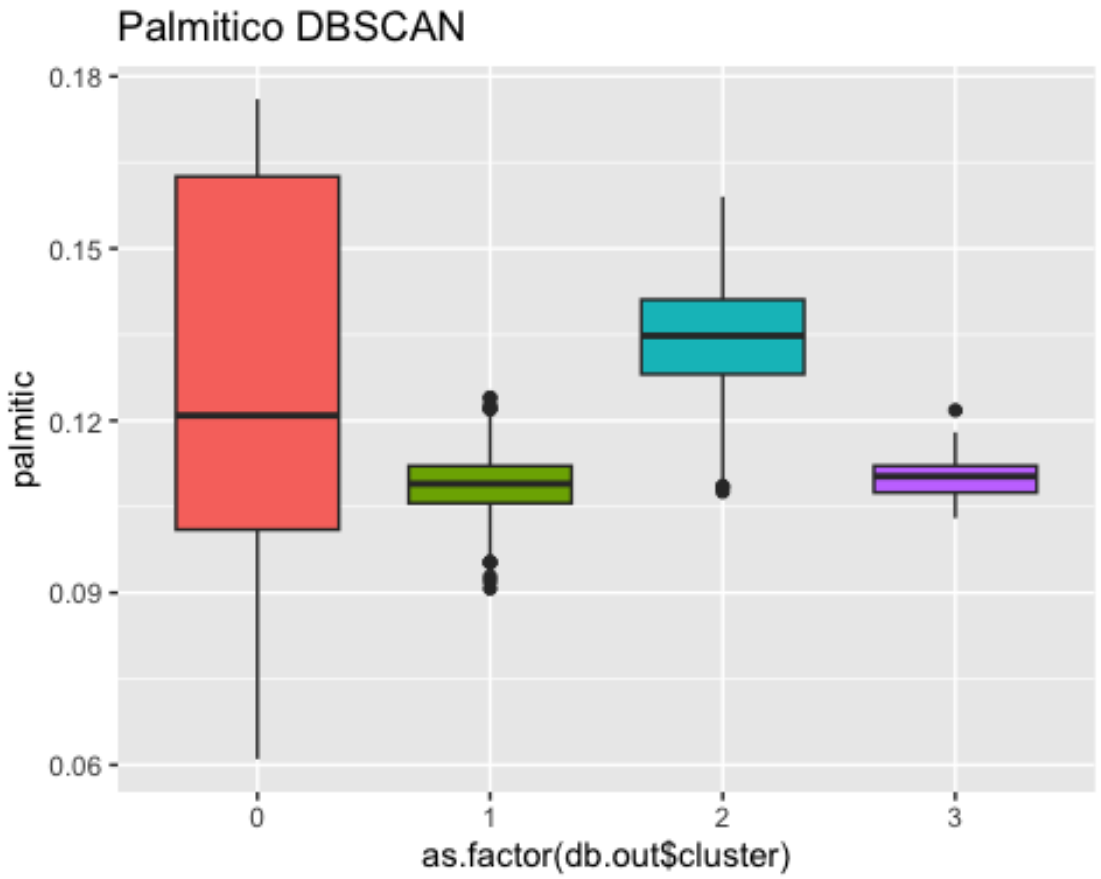
In questo caso vediamo che i cluster che contengono i valori prossimi allo zero sono l'1 ed il 3. I punti di "noise" rimangono quelli con li range maggiore anche se il cluster 1 contiene anche esso valori per l'acido eicosenoico molto diversi.

Comparazione con k-means

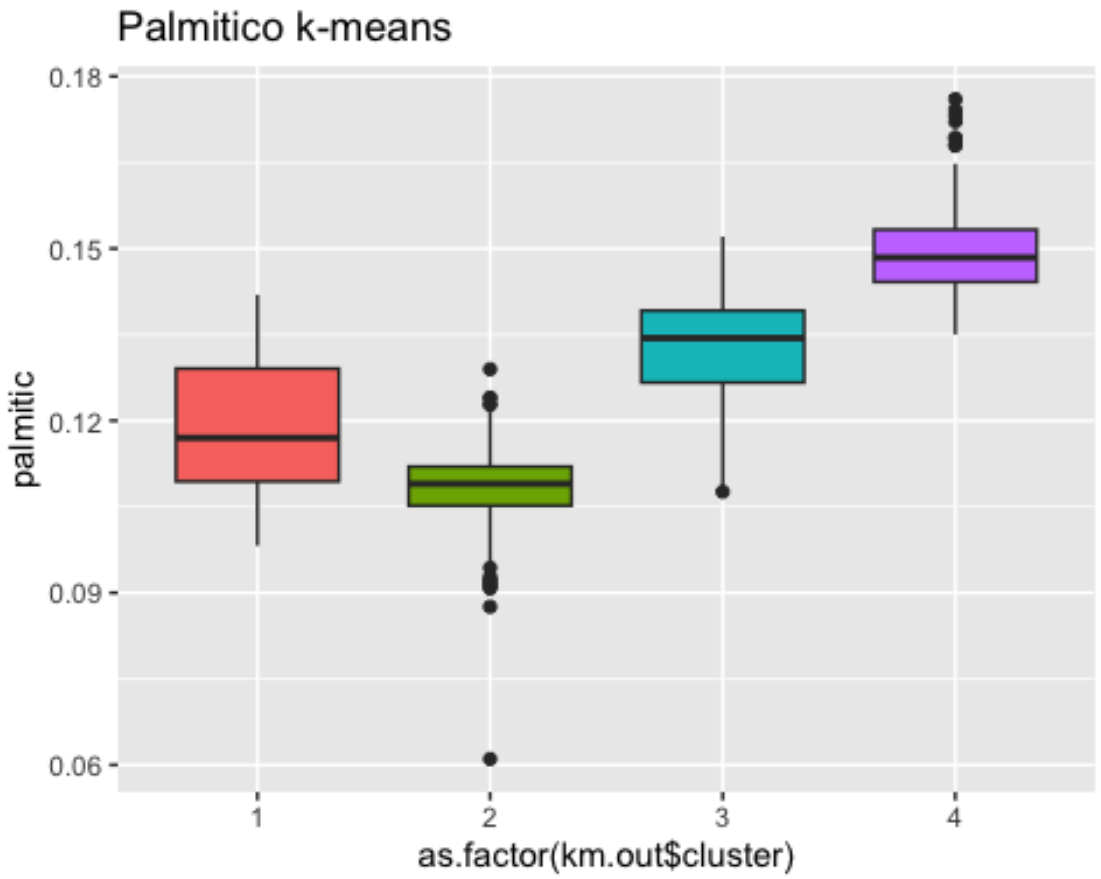
Paragonando le distribuzioni in boxplot di kmeans e dbscan si notano delle corrispondenze tra le due divisioni in cluster: - Notiamo che molti dei vaori estremi sei vari cluster di kmenas vengono assegnati all' insieme dei punti di "noise". - Soprattutto il cluster 4 di kmeans sembra contenere valori che dbscan conta come punti di "noise".

Come si vede nel caso dell'acido palmitico

```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = palmitic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE) + labs( title ="Palmitico DBSCAN")
```

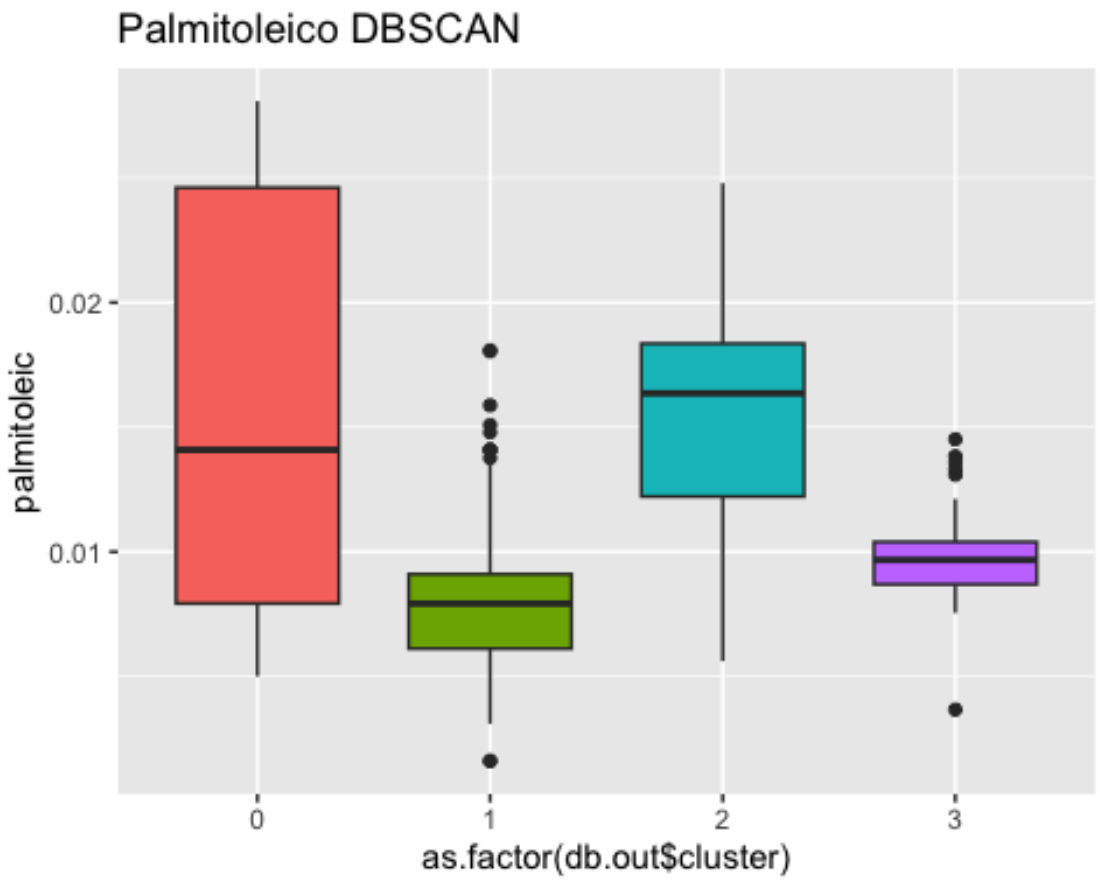



```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = palmitic, fill =
as.factor(km.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE) + labs( title ="Palmitico k-means")
```

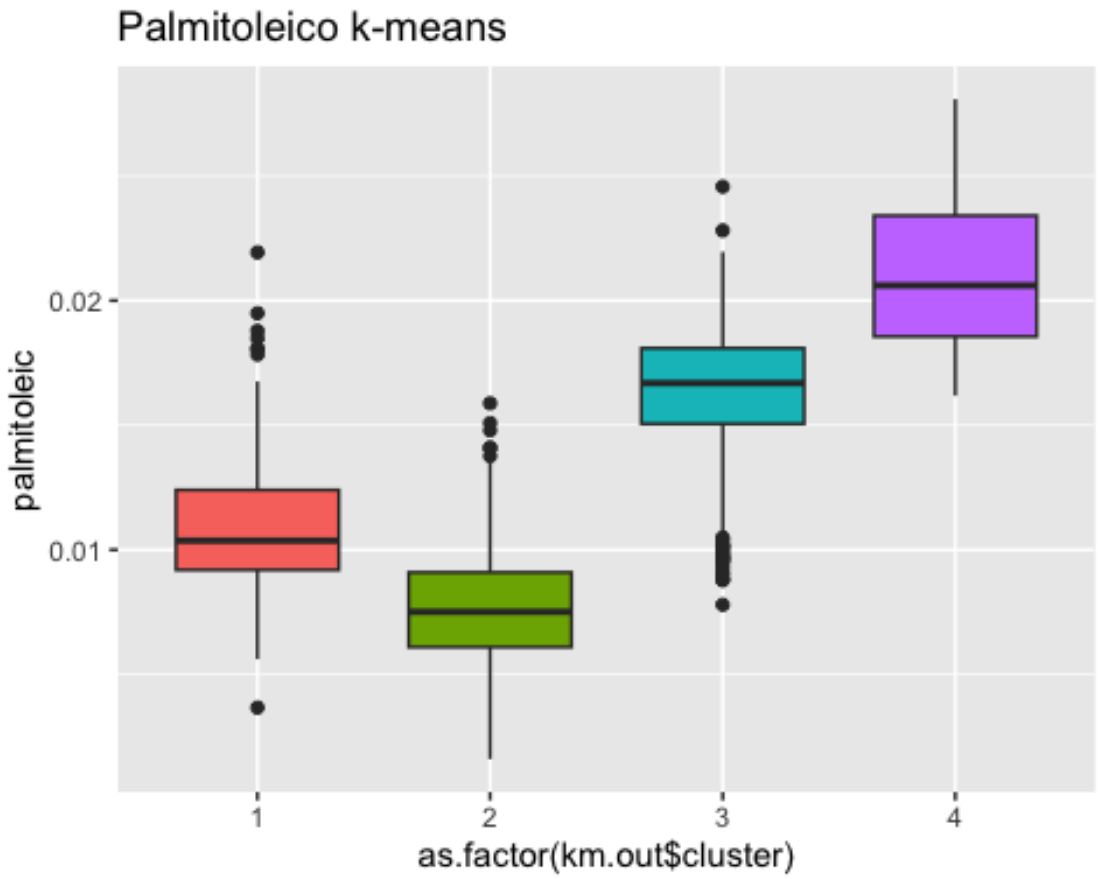


e palmitoleico

```
ggplot(oliveoil, aes(x = as.factor(db.out$cluster), y = palmitoleic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE) + labs( title ="Palmitoleico DBSCAN")
```



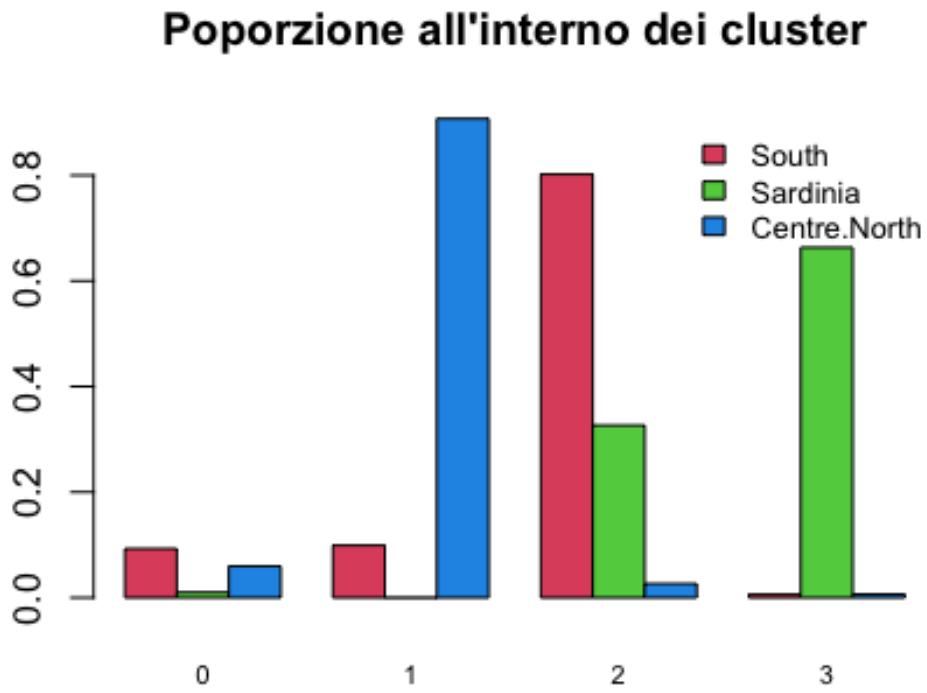

```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = palmitoleic, fill =
as.factor(km.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE) + labs( title ="Palmitoleico k-means")
```



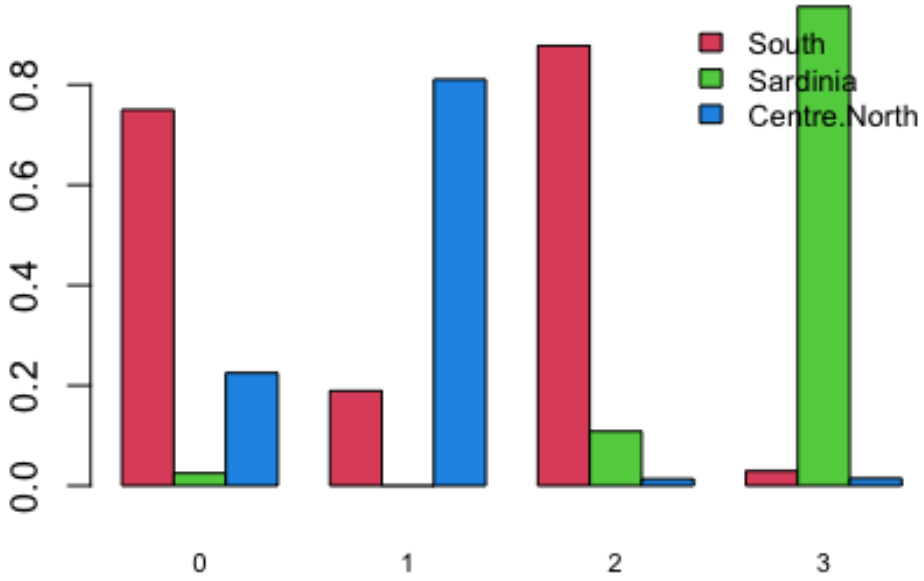
- È presente una grande somiglianza inoltre tra i cluster 2 e 3 di kmeans e i cluster 1 e 2 di dbscan, come si vede anche nei grafici sopra.

Variabile macro.area

```
prop.table(table(db.out$cluster, oliveoil$macro.area),1)

##
##          South   Sardinia Centre.North
##   0 0.75000000 0.02500000   0.22500000
##   1 0.18934911 0.00000000   0.81065089
##   2 0.87796610 0.10847458   0.01355932
##   3 0.02941176 0.95588235   0.01470588

barplot(prop.table(table(oliveoil$macro.area, db.out$cluster),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:4, cex.names
= 0.70)
legend("topright", legend = rownames(prop.table(table(oliveoil$macro.area,
db.out$cluster),1)), fill = 2:4, cex = 0.8, bty = "n")
```




```r
barplot(prop.table(table(oliveoil$macro.area, db.out$cluster),2), beside = T,
legend = F, main = "", col = 2:4, cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveoil$macro.area,
db.out$cluster),1)), fill = 2:4, cex = 0.8, bty = "n")
```

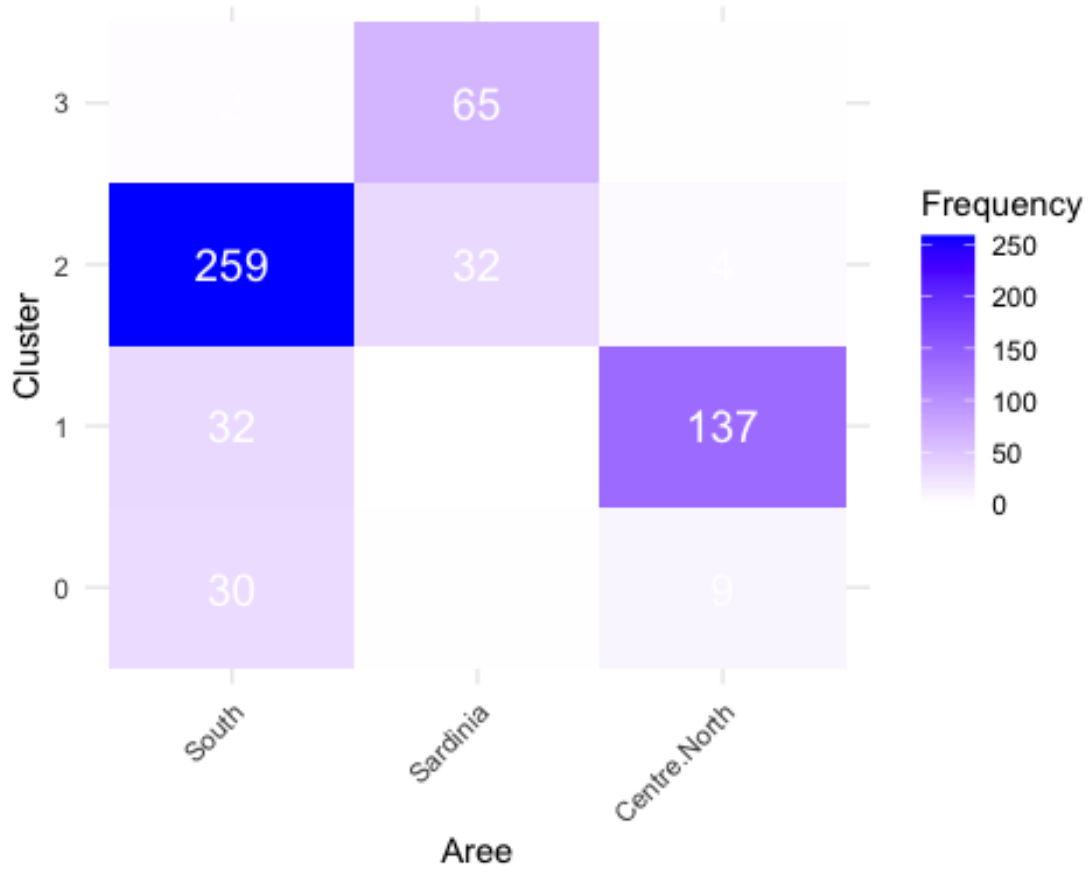


Gli oli del sud sono principalmente stati raggruppati nel cluster 3, con più di 75% delle osservazioni totali di oli del sud. Gli oli della sardegna sono invece presenti per la maggiorparte nel cluster 3 e nel cluster 2 in quantità minore. Gli oli del Centro nord sono infine quasi completamente nel cluster 1. Infine si nota che l'insieme dei punti di "noise" è composto principalmente da oli del sud. CONFRONTO CON KMENAS Si nota che: - gli oli del centro nord in entrambi i casi sono quasi esclusivamente in un singolo cluster - Gli oli della Sardegna hanno anche distribuzioni simili, sono finfatti divisi principalmente tra due cluser per entrambi i metodi - invece gli oli del sud sono distribuiti più uniformemente con kmeans che con dbscan.

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveoil$macro.area, Aree =
db.out$cluster)

table( Aree = db.out$cluster, Cluster = oliveoil$macro.area)

##     Cluster
## Aree South Sardinia Centre.North
##    0    30        1            9
##    1    32        0          137
##    2   259       32            4
##    3     2       65            1
```

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Aree, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Aree", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



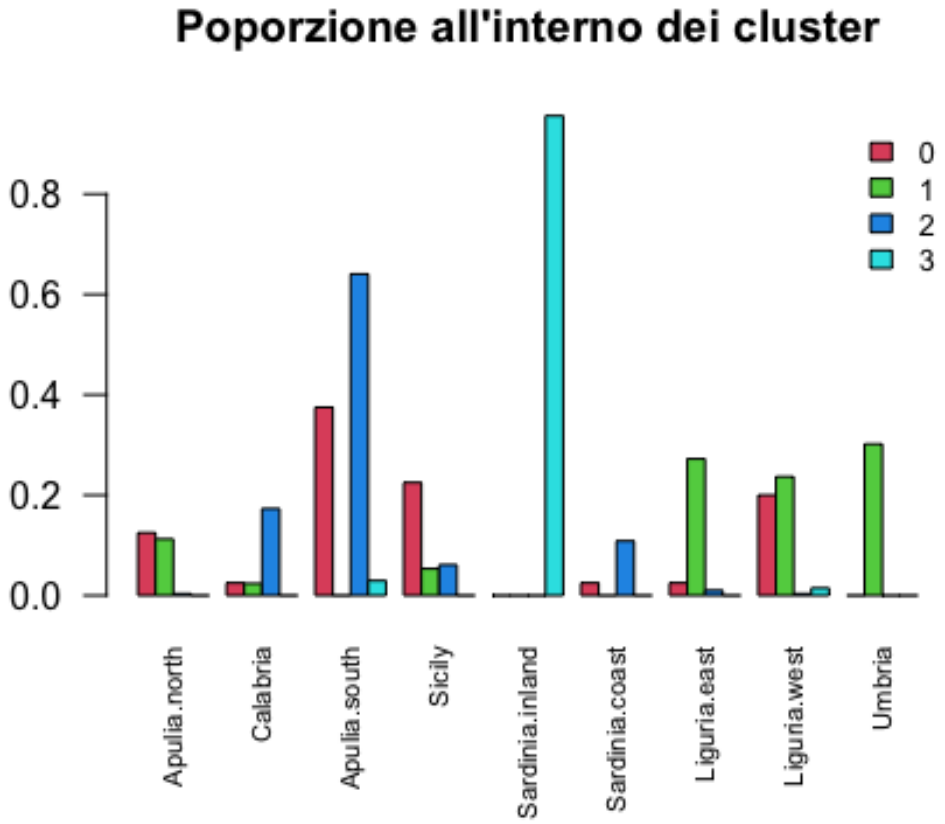
Variabile region

```r
prop.table(table(db.out$cluster, oliveoil$region),1)
```

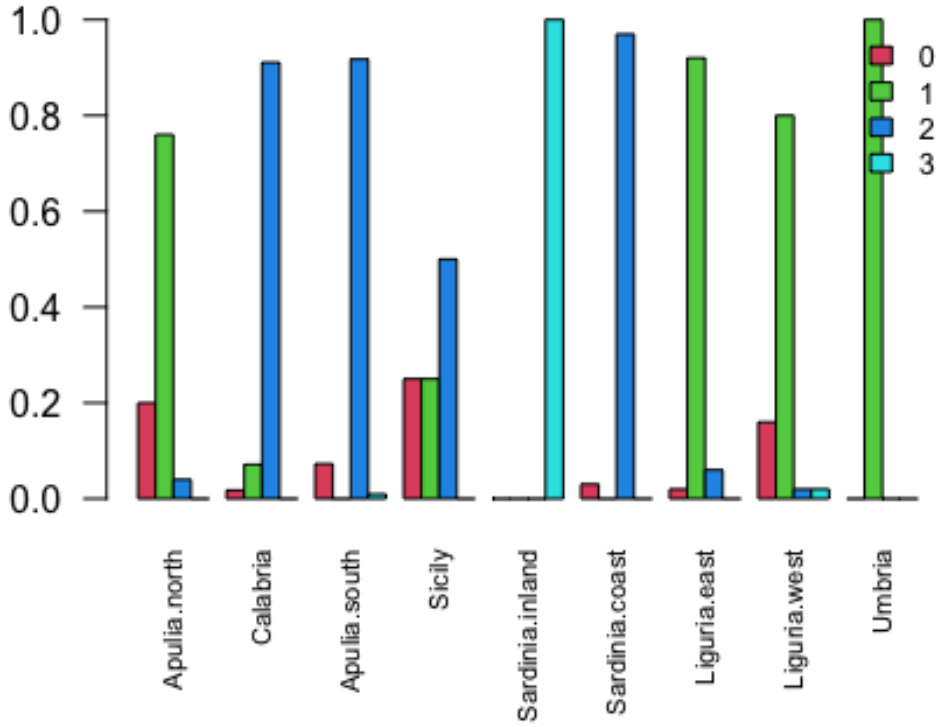
```
##
##     Apulia.north    Calabria Apulia.south      Sicily Sardinia.inland
##   0  0.125000000 0.025000000  0.375000000 0.225000000     0.000000000
##   1  0.112426036 0.023668639  0.000000000 0.053254438     0.000000000
##   2  0.003389831 0.172881356  0.640677966 0.061016949     0.000000000
##   3  0.000000000 0.000000000  0.029411765 0.000000000     0.955882353
##
##     Sardinia.coast Liguria.east Liguria.west      Umbria
##   0    0.025000000  0.025000000  0.200000000 0.000000000
##   1    0.000000000  0.272189349  0.236686391 0.301775148
```

```
##   2    0.108474576  0.010169492  0.003389831 0.000000000
##   3    0.000000000  0.000000000  0.014705882 0.000000000
```

```r
counts <- prop.table(table(db.out$cluster, oliveoil$region), 1)

barplot(prop.table(table(db.out$cluster, oliveoil$region),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:5, cex.names
= 0.70, las=2)
legend("topright", legend = rownames(counts), fill = 2:5, cex = 0.8, bty =
"n")
```




```r
barplot(prop.table(table(db.out$cluster, oliveoil$region),2), beside = T,
legend = F, main = "", col = 2:5, cex.names = 0.70, las=2)
legend("topright", legend = rownames(counts), fill = 2:5, cex = 0.8, bty =
"n")
```



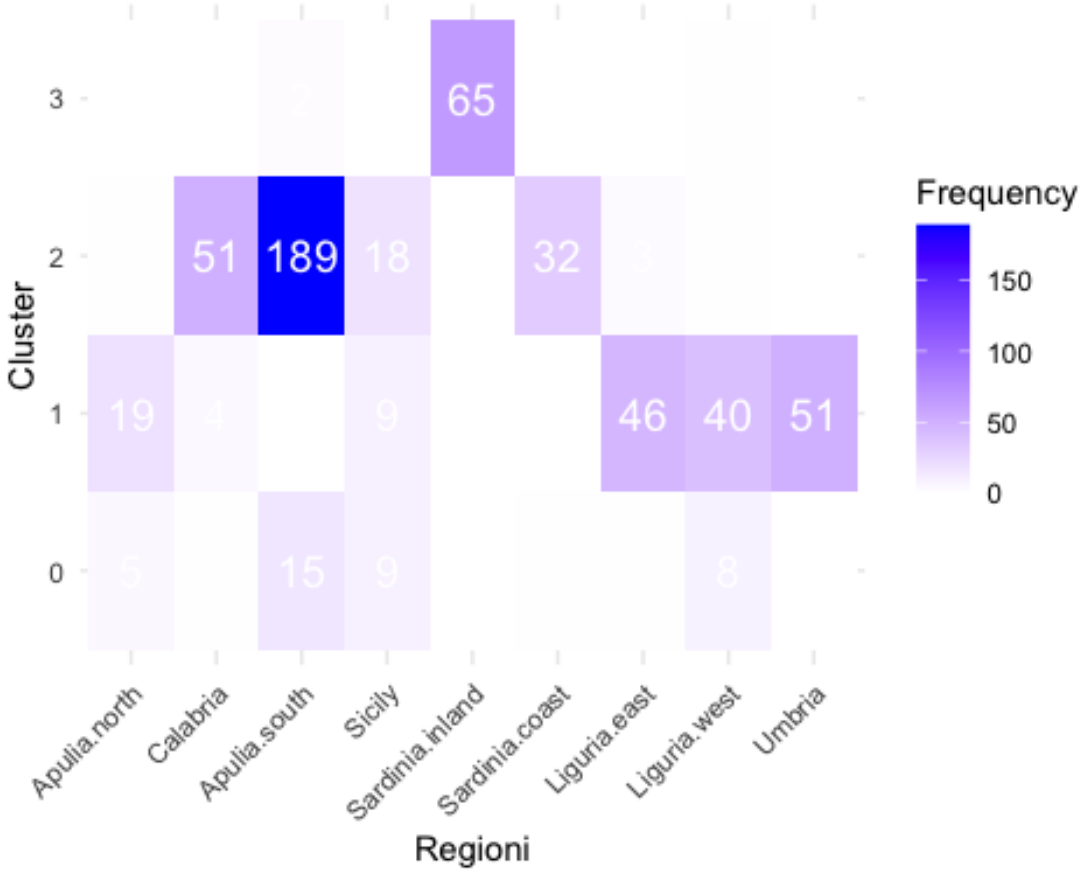
Fatta eccezione del cluster 3 che è quasi esclusivamente composto da oli prodotti nella Sardegna inland, nei cluster sono presenti oli da più regioni. In particolare: - oli provenienti dalla Puglia nord, Liguria del est, Liguria dell'Ovest e Umbria sono principalmente nel cluster 1. - oli provenienti dalla Calabria, Puglia sud e Sardegna costa sono in granparte nel cluster 2. - gli oli che vengono considerati come punti di "noise" invece sono presenti nella maggiorparte delle regioni.

Inoltre confrontrando le regioni si evince che: - Oli della puglia nord, liguria est, liguria ovest e umbria sono in entrambi i casi principalemente presenti nello stesso cluster. - La sardegna inland non condivide più il cluster principalmente con Calbria, Sardegna e Liguria dell'ovest ma adesso ha un cluster composto quasi esclusivamente dai propri oli.

```
confusion_matrix <- table(Cluster = oliveoil$region, Regioni =
db.out$cluster)

table(Regioni = db.out$cluster, Cluster = oliveoil$region)

##         Cluster
## Regioni Apulia.north Calabria Apulia.south Sicily Sardinia.inland
##       0            5        1           15      9               0
##       1           19        4            0      9               0
##       2            1       51          189     18               0
##       3            0        0            2      0              65
```

```
##         Cluster
## Regioni Sardinia.coast Liguria.east Liguria.west Umbria
##       0              1            1            8      0
##       1              0           46           40     51
##       2             32            3            1      0
##       3              0            0            1      0
```

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Regioni, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Regioni", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Trasformazione ALR - Additive Log-Ratio Transofmation

A causa della natura compositiva dei dati, ovvero per il fatto che essi sommano a 10000 sulle righe, si decide di applicare una ulteriore trasformazione di tipo Additive Log-Ratio, in modo da rapportare tutte le colonne con una colonna fissata

Questa trasformazione consiste nel dividere ogni colonna del dataset per una colonna scelta arbitrariamente, e di applicare al risultato l'opposto del logatirmo: come nella seguente formula:

$$y_{ij} = -log\left(\frac{x_{ij}}{x_{ik}}\right) \quad , \ \forall j \neq k \ \text{ colonna}, \forall i \ \text{ riga, con k fissata}$$

La colonna k viene rimossa in quanto il rapporto

$$\frac{x_{ij}}{x_{ik}}$$

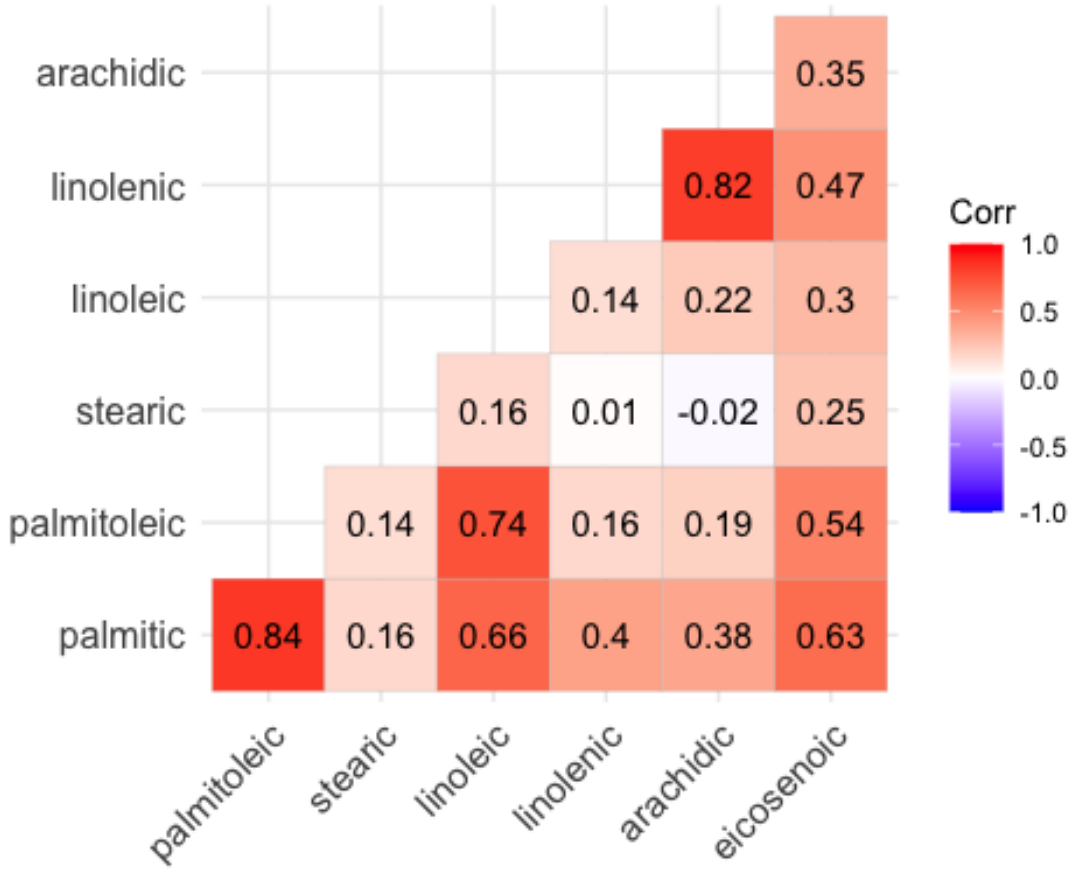
è sempre 1 e di conseguenza si ottengono valori nulli una volta applicato il logaritmo

Quindi si è deciso di applicare la trasformazione alr al dataset oliveoil in modo da evidenziare eventuali differenze. Si sceglie la colonna 6 in modo da cercare di evidenziare cluster nei dati meno correlati. In quanto l'acido oleico è il più presente ed il più correlato con le altre variabili

```
oliveALR <- -log(oliveoil[,-c(1,2,6)]/oliveoil[,6])
oliveALR <- cbind(oliveoil[,1:2], oliveALR)
```

Si vuole analizzare ora il nuvo dataset oliveALR

```
ggcorrplot(cor(oliveALR[,3:9]), type = "lower", lab = TRUE)
```



Dal grafico si vede che le correlazioni ora sono sempre positive, inoltre le variabili che erano fortemente correlate prima della trasformazione, rimangono fortemente correlate anche dopo la trasformazione logaritmica.

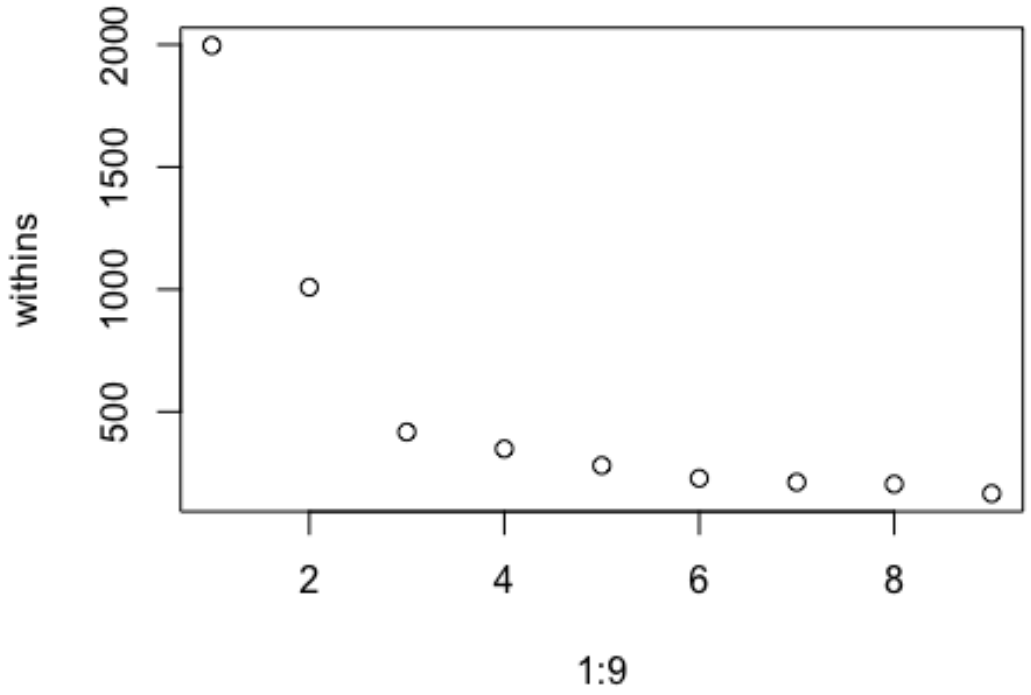
Altre informazioni degne di nota riguardano:

K-means con ALR

Testiamo l'algoritmo con dati trasformati tramite ALR.

Cerchiamo un buon numero di cluster con i dati trasformati usando il metodo elbow. In questo caso osserviamo che un buon numero per k è 4 oppure 5. Prima abbiamo trovato 3 o 4.

```
# METODO DEL ELBOW
withins <- c(1:9)
for (i in 1:9){
  km.out <- kmeans(oliveALR[,3:9], centers = i, nstart = 15)
  withins[i] <- km.out$tot.withinss
}
par(mfrow=c(1,1))
plot(1:9, withins)
```



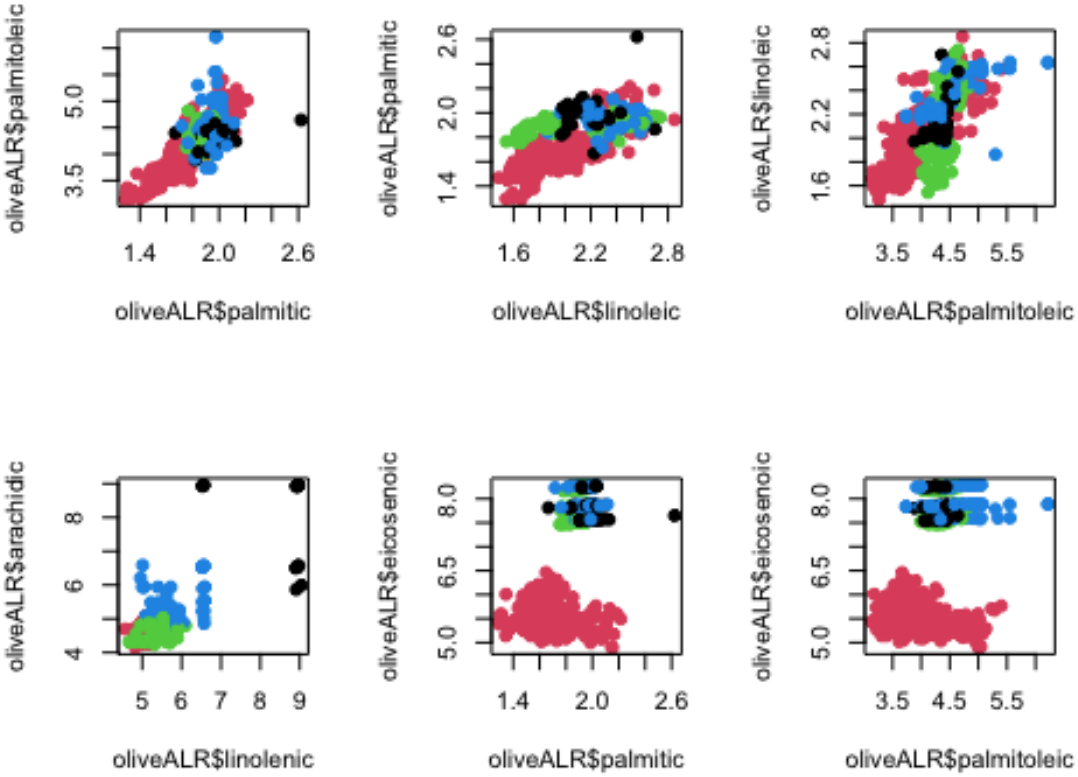
Si scieglie k = 4 e si usa la funzione kmeans()

```
set.seed(17)
km.out <- kmeans(oliveALR[,3:9], centers=4, nstart = 15)
str(km.out)
```

```
## List of 9
##  $ cluster     : int [1:572] 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:4, 1:7] 2 1.68 1.89 1.97 4.34 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##  $ totss       : num 1996
##  $ withinss    : num [1:4] 88 152.9 36.6 71.2
##  $ tot.withinss: num 349
##  $ betweenss   : num 1648
##  $ size        : int [1:4] 37 323 132 80
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

Facciamo lo scatterplot di alcune variabili per visualizzare il cluster in particolare scegliamo degli acidi con una buona correlazione sulla base di quanto ottenuto dalla matrice di correlazione.

```
par(mfrow=c(2,3))

# palmitic palmitoleic
plot(oliveALR$palmitic, oliveALR$palmitoleic, col = km.out$cluster, pch = 19)

# linoleic palmitoleic
plot(oliveALR$linoleic, oliveALR$palmitic, col = km.out$cluster, pch = 19)

# linoleic palmitoleic
plot(oliveALR$palmitoleic, oliveALR$linoleic, col = km.out$cluster, pch = 19)

# arachidic linolenic
plot(oliveALR$linolenic, oliveALR$arachidic, col = km.out$cluster, pch = 19)

# eicosenoic palmitic
plot(oliveALR$palmitic, oliveALR$eicosenoic, col = km.out$cluster, pch = 19)

# eicosenoic palmitoleic
plot(oliveALR$palmitoleic, oliveALR$eicosenoic, col = km.out$cluster, pch =
19)
```

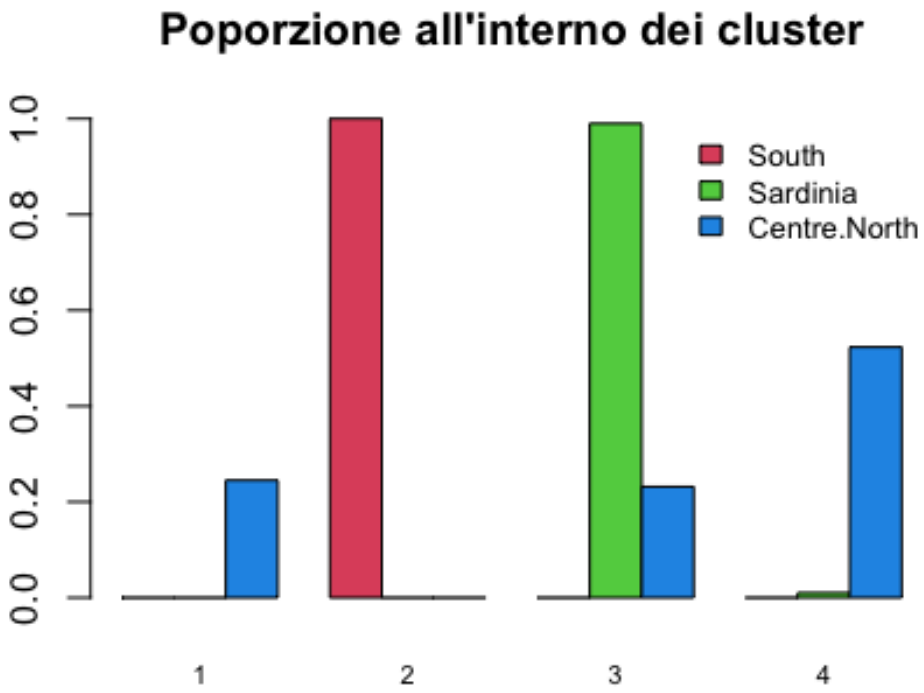


Variabile macro.area

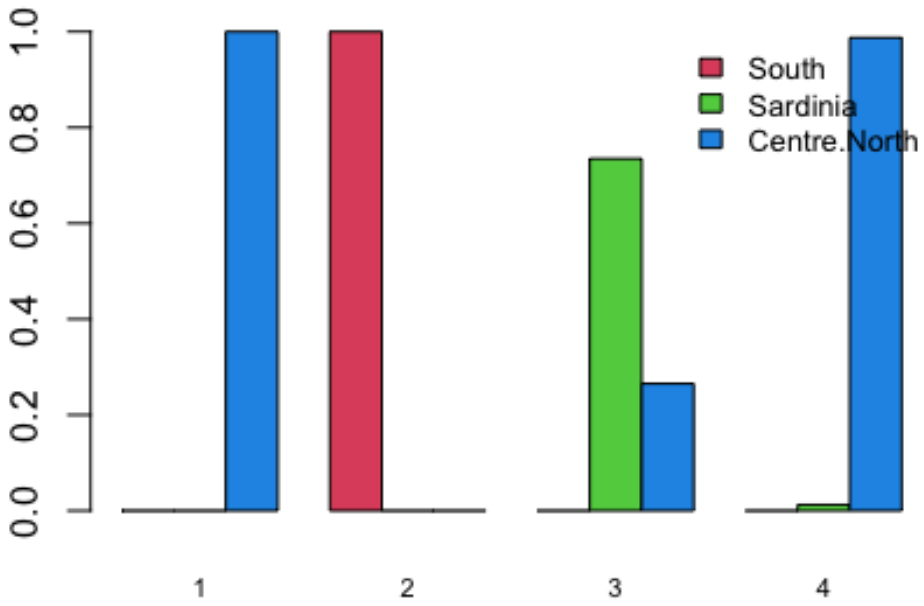
```
prop.table(table(oliveALR$macro.area, km.out$cluster),1)
```

```
##
##                          1          2          3          4
##   South         0.00000000 1.00000000 0.00000000 0.00000000
##   Sardinia      0.00000000 0.00000000 0.98979592 0.01020408
##   Centre.North 0.24503311 0.00000000 0.23178808 0.52317881
```

```
barplot(prop.table(table(oliveALR$macro.area, km.out$cluster),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:4, cex.names
= 0.70)
legend("topright", legend = rownames(prop.table(table(oliveALR$macro.area,
km.out$cluster),1)
), fill = 2:4, cex = 0.8, bty = "n")
```




```r
barplot(prop.table(table(oliveALR$macro.area, km.out$cluster),2), beside = T,
legend = F, main = "", col = 2:4, cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveALR$macro.area,
km.out$cluster),1)
), fill = 2:4, cex = 0.8, bty = "n")
```



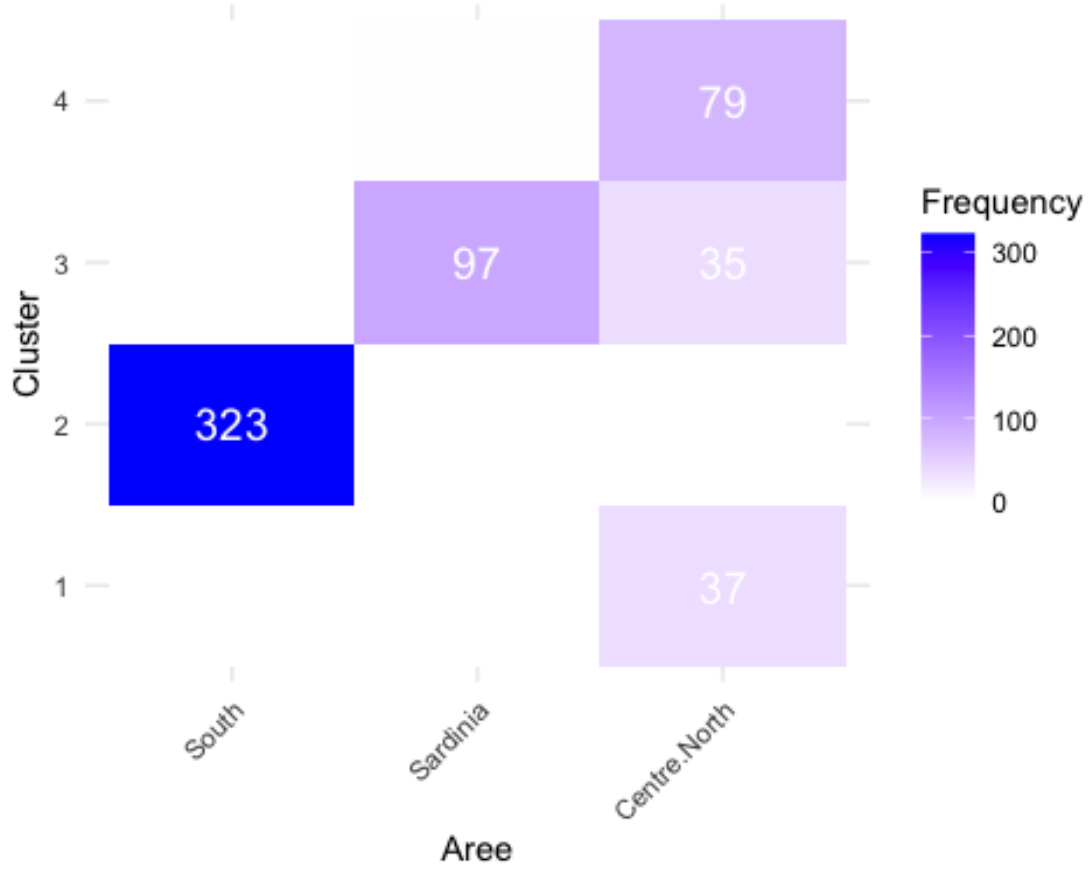
Gli oli del Sud si trovano al 100% nel cluster 2 Gli oli della Sardegna al 99% nel cluster 3 e 1% nel cluster 4. Gli oli del centro nord al 25% nel cluster 1, 23% nel cluster 3 e 52% nel cluster 4.

Il cluster 1 è composto al 100% da oli del centro nord. Il cluster 2 al 100% da oli del sud. Il cluster 3 al 73% da oli della Sardegna e al 27% da oli del centro nord. Il cluster 4 al 99% da oli del centro nord e all'1% da oli della sardegna.

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveALR$macro.area, Aree =
km.out$cluster)

table( Aree = km.out$cluster, Cluster = oliveALR$macro.area)

##     Cluster
## Aree South Sardinia Centre.North
##    1     0        0           37
##    2   323        0            0
##    3     0       97           35
##    4     0        1           79

ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Aree, fill = Freq)) +
```

```
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Aree", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

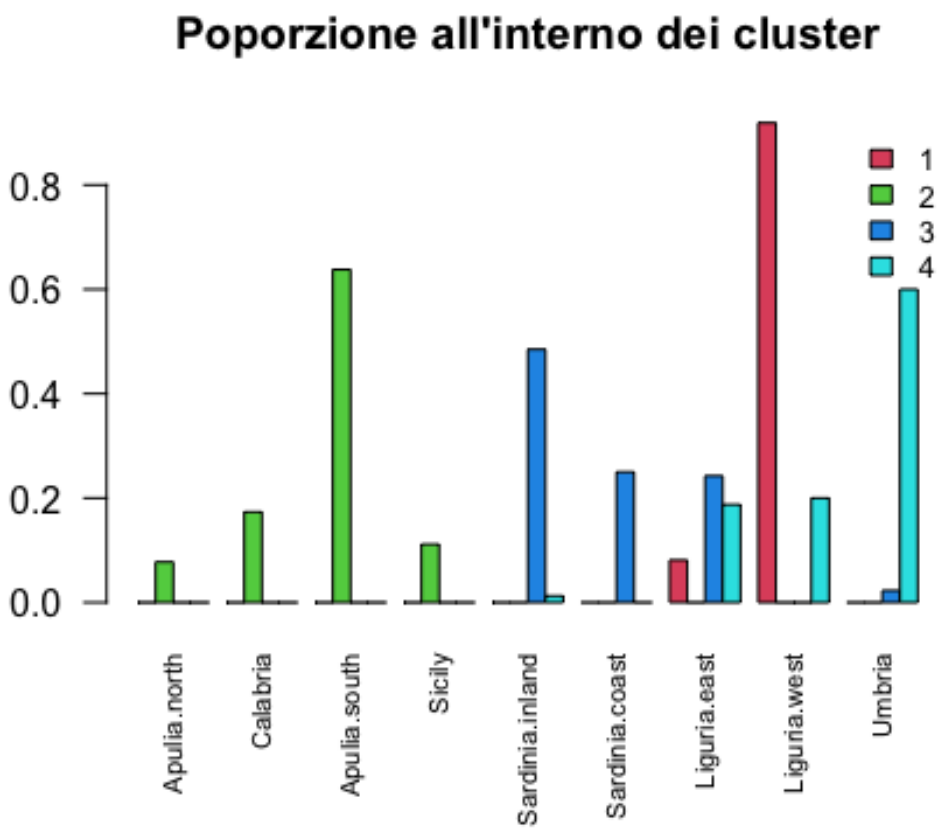


Variabile region

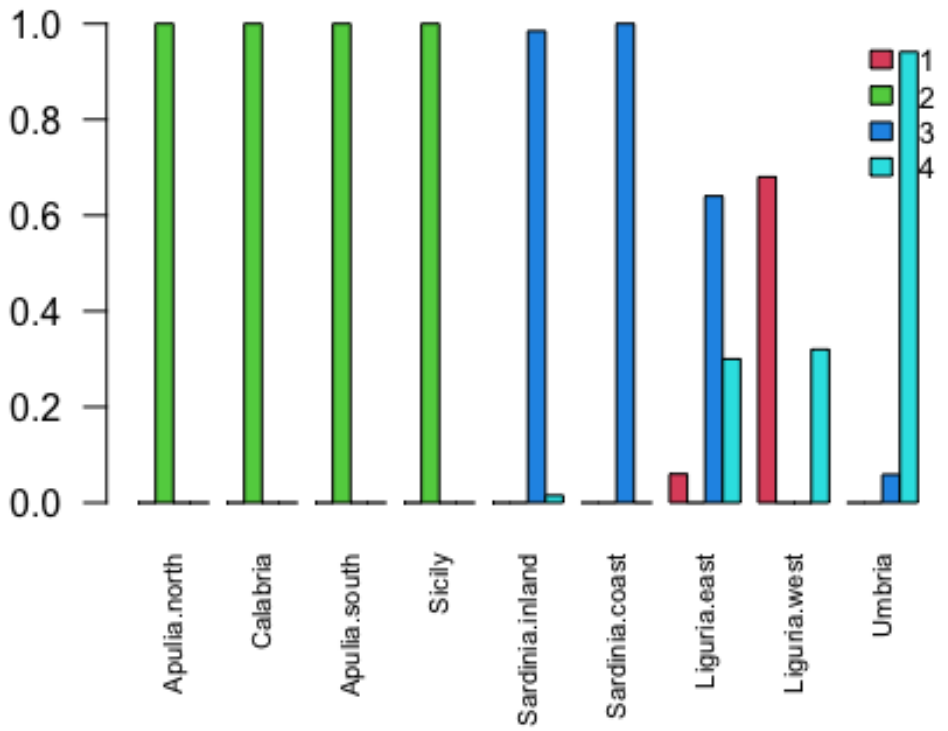
```
prop.table(table(km.out$cluster, oliveALR$region),1)
```

```
##
##     Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   1   0.00000000 0.00000000   0.00000000 0.00000000      0.00000000
##   2   0.07739938 0.17337461   0.63777090 0.11145511      0.00000000
##   3   0.00000000 0.00000000   0.00000000 0.00000000      0.48484848
##   4   0.00000000 0.00000000   0.00000000 0.00000000      0.01250000
##
##     Sardinia.coast Liguria.east Liguria.west     Umbria
##   1     0.00000000   0.08108108   0.91891892 0.00000000
##   2     0.00000000   0.00000000   0.00000000 0.00000000
##   3     0.25000000   0.24242424   0.00000000 0.02272727
##   4     0.00000000   0.18750000   0.20000000 0.60000000
```

```r
barplot(prop.table(table(km.out$cluster, oliveALR$region),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:5, cex.names
= 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(km.out$cluster,
oliveALR$region),1)), fill = 2:5, cex = 0.8, bty = "n")
```




```r
barplot(prop.table(table(km.out$cluster, oliveALR$region),2), beside = T,
legend = F, main = "", col = 2:5, cex.names = 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(km.out$cluster,
oliveALR$region),1)), fill = 2:5, cex = 0.8, bty = "n")
```



Gli oli della puglia del nord si trovano al 100% nel cluster 2. Gli oli della calabria si trovano al 100% nel cluster 2. Gli oli della puglia del sud si trovano al 100% nel cluster 2. Gli oli della Sicilia si trovano al 100% nel cluster 2. Gli oli della Sardegna inland so trovano al 99% nel cluster 3 e 1% nel cluster 4. Gli oli della Sardegna coast si trovano al 100% nel cluster 3. Gli oli della liguria est si trovano al 64% nel cluster 3, al 30% nel cluster 4 e al 6% nel cluster 1. Gli oli della liguria ovest si trovano al 68% nel cluster 1 e al 32% nel 4. Gli oli dell'umbria si trovano al 94% nel cluster 4 e al 6% nel cluster 3.

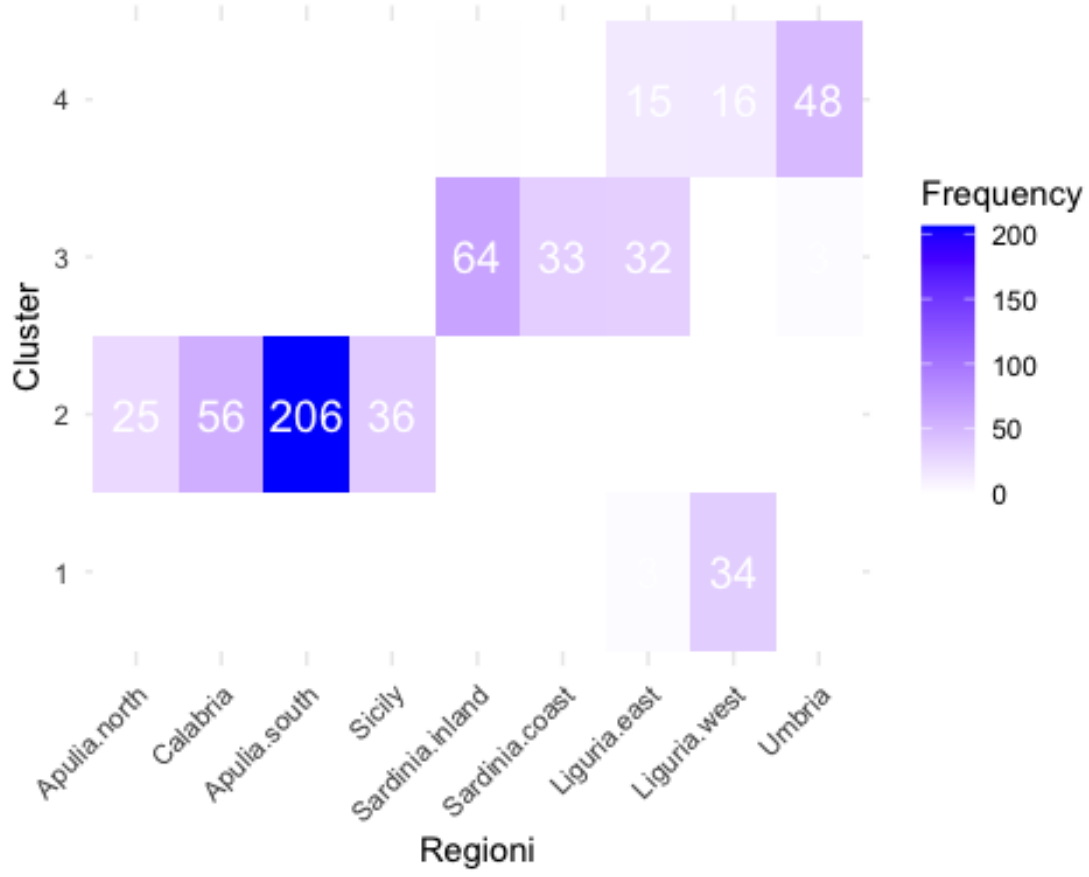
Il cluster 1 è formato al 91% da oli della liguria ovest e al 9% da oli della liguria est. Il cluster 2 è formato al 17% da oli della calabria, al 64% da oli della puglia sud, 8% da oli della puglia del nord e 11% da oli della Sicilia. Il cluster 3 è formato al 48% da oli della Sardegna inland, al 25% da oli della Sardegna coast e al 24% da oli della liguria est, al 1% da oli dell'umbria. Il cluster 4 contiene un 1% di oli della Sardegna inland, 19% di oli della liguria est, 20% liguria ovest e 60 % Umbria.

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveALR$region, Regioni =
km.out$cluster)

table(Regioni = km.out$cluster, Cluster = oliveALR$region)
```

```
##         Cluster
## Regioni Apulia.north Calabria Apulia.south Sicily Sardinia.inland
##       1            0        0            0      0               0
##       2           25       56          206     36               0
##       3            0        0            0      0              64
##       4            0        0            0      0               1
##         Cluster
## Regioni Sardinia.coast Liguria.east Liguria.west Umbria
##       1              0            3           34      0
##       2              0            0            0      0
##       3             33           32            0      3
##       4              0           15           16     48
```

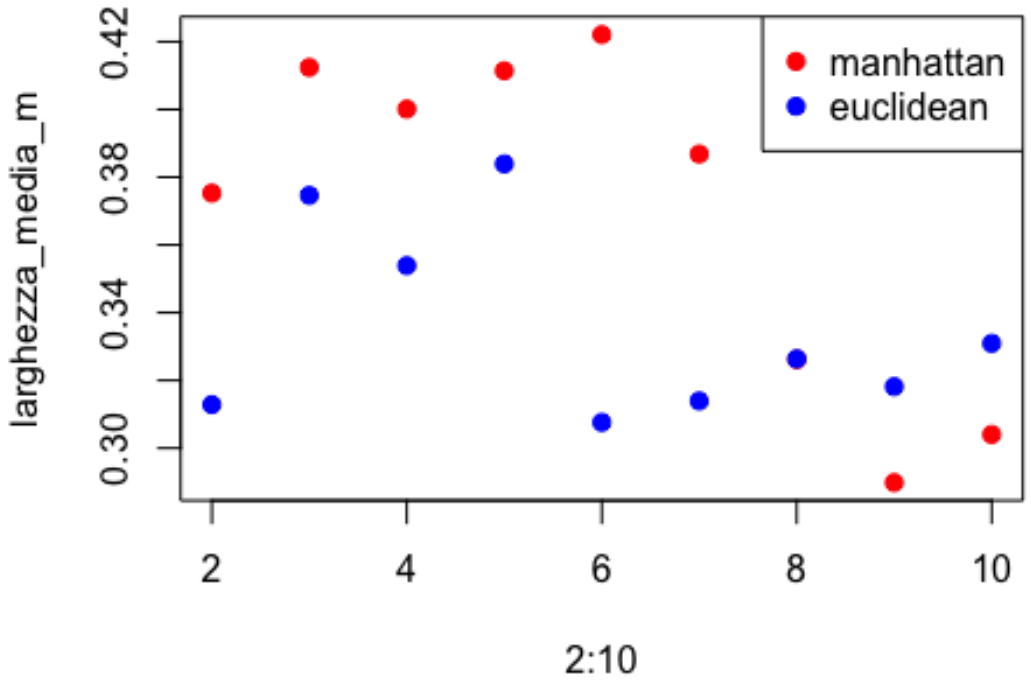
```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Regioni, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Regioni", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



PAM con ALR

Come visto prima, si testa l'algoritmo PAM con le due distanze e con diversi valori di K per scegliere i parametri che creano i cluster migliori.

```
# DISTANZA MANHATTAN
larghezza_media_m <- c(1:9)
for (i in 2:10){
  pam.out<-pam(oliveALR[,3:9], i, metric="manhattan", stand=TRUE, nstart =
10)
  larghezza_media_m[i-1] <- pam.out$silinfo$avg.width
}

# DISTANZA EUCLIDEA
larghezza_media_e <- c(1:9)
for (i in 2:10){
  pam.out<-pam(oliveALR[,3:9], i, metric="euclidean", stand=TRUE, nstart =
10)
  larghezza_media_e[i-1] <- pam.out$silinfo$avg.width
}

plot(2:10, larghezza_media_m, col = "red", pch =19)
points(2:10, larghezza_media_e, col = "blue", pch =19)
legend("topright", legend = c("manhattan", "euclidean"), col = c("red",
"blue"), pch =19)
```

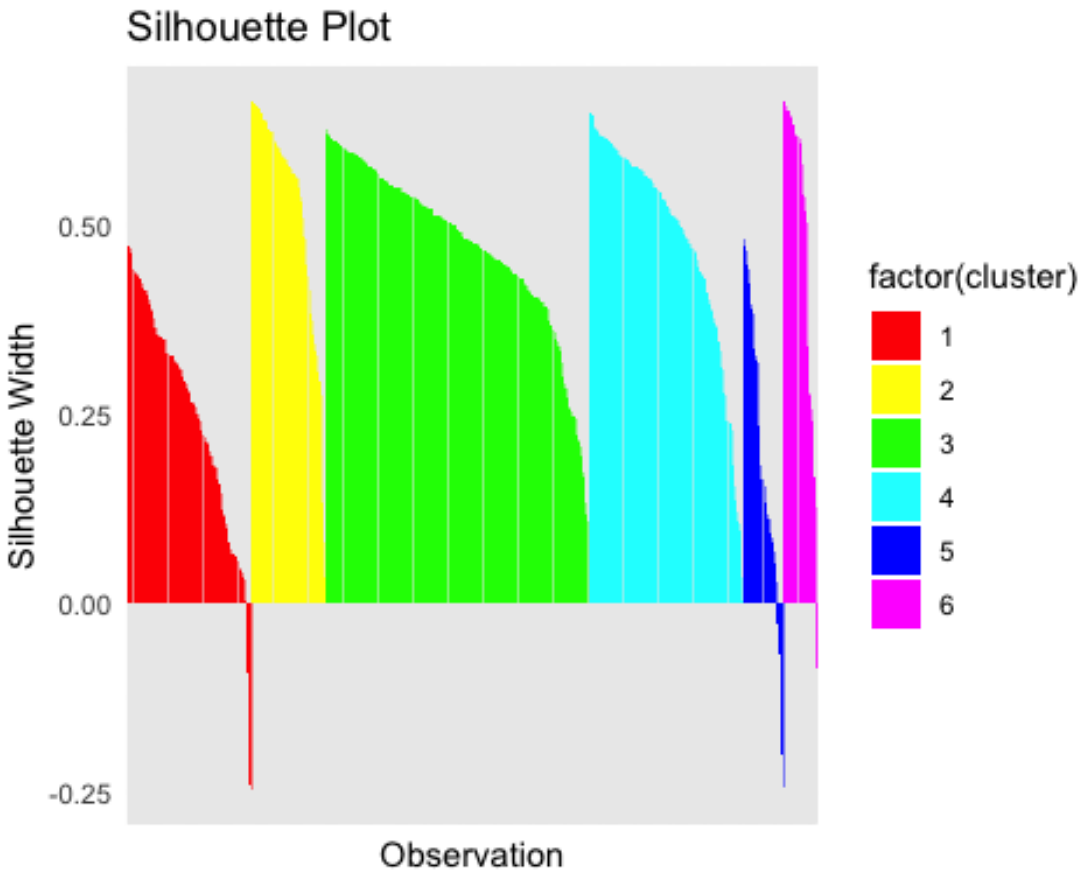


Il numero di cluster migliore sembra essere 6 La distanza manhattan è nettamente migliore della distanza euclidea a parità di numero di cluster, come si vede dal grafico

```
set.seed(17)
pam.out<-pam(oliveALR[,3:9], 6, metric="manhattan", stand=TRUE, nstart = 10)
str(pam.out)
```

```
## List of 10
##  $ medoids   : num [1:6, 1:7] 1.77 1.99 1.58 1.94 2.02 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##  $ id.med    : int [1:6] 51 438 239 343 550 530
##  $ clustering: int [1:572] 1 1 2 1 1 1 1 1 1 2 ...
##  $ objective : Named num [1:2] 4.73 3.08
##   ..- attr(*, "names")= chr [1:2] "build" "swap"
##  $ isolation : Factor w/ 3 levels "no","L","L*": 1 1 1 1 1 1
##   ..- attr(*, "names")= chr [1:6] "1" "2" "3" "4" ...
##  $ clusinfo  : num [1:6, 1:5] 103 61 219 127 34 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:5] "size" "max_diss" "av_diss" "diameter" ...
##  $ silinfo   :List of 3
```

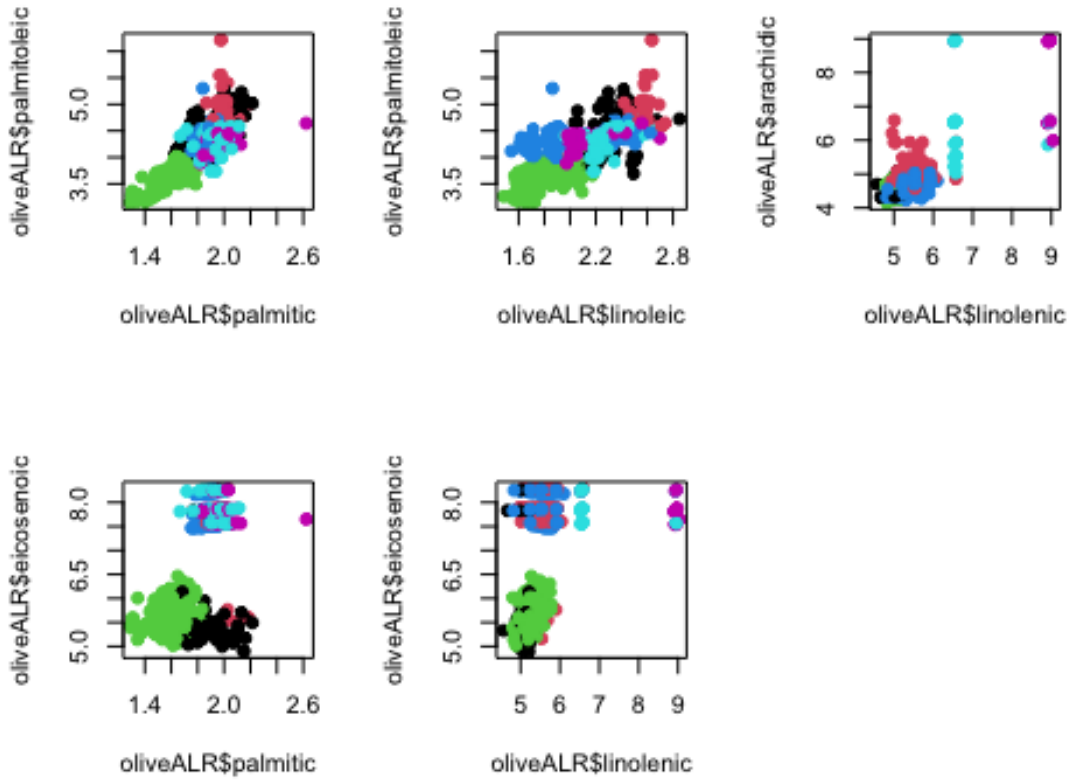
```
##   ..$ widths         : num [1:572, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:572] "78" "293" "51" "37" ...
##   .. .. ..$ : chr [1:3] "cluster" "neighbor" "sil_width"
##   ..$ clus.avg.widths: num [1:6] 0.246 0.521 0.473 0.481 0.187 ...
##   ..$ avg.width      : num 0.422
##  $ diss      : NULL
##  $ call      : language pam(x = oliveALR[, 3:9], k = 6, metric =
"manhattan", nstart = 10, stand = TRUE)
##  $ data      : num [1:572, 1:7] 1.24 1.07 2.51 2.01 1.34 ...
##   ..- attr(*, "scaled:center")= num [1:7] 1.79 4.14 3.47 2.04 5.59 ...
##   ..- attr(*, "scaled:scale")= num [1:7] 0.159 0.411 0.122 0.273 0.514 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##  - attr(*, "class")= chr [1:2] "pam" "partition"

# GRAFICO
sil_df <- as.data.frame(silhouette(pam.out)[, 1:3])
colnames(sil_df) <- c("cluster", "neighbor", "sil_width")
sil_df$obs <- 1:nrow(sil_df)
sil_df <- sil_df[order(sil_df$cluster, -sil_df$sil_width),]
sil_df$obs_ordered <- factor(sil_df$obs, levels = sil_df$obs)

ggplot(sil_df, aes(x = obs_ordered, y = sil_width, fill = factor(cluster))) +
  geom_bar(stat = "identity") +
  scale_fill_manual(values = rainbow(6)) +
  labs(title = "Silhouette Plot", x = "Observation", y = "Silhouette Width")
+
  theme_minimal() +
  theme(axis.text.x = element_blank(), axis.ticks.x = element_blank())
```



Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

```
par(mfrow=c(2,3))

# palmitic palmitoleic
plot(oliveALR$palmitic, oliveALR$palmitoleic, col = pam.out$cluster, pch =
19)

# linoleic palmitoleic
plot(oliveALR$linoleic, oliveALR$palmitoleic, col = pam.out$cluster, pch =
19)

# linolenic arachidic
plot(oliveALR$linolenic, oliveALR$arachidic, col = pam.out$cluster, pch = 19)

# palmitic eicosenoic
plot(oliveALR$palmitic, oliveALR$eicosenoic, col = pam.out$cluster, pch = 19)

# linolenic eicosenoic
plot(oliveALR$linolenic, oliveALR$eicosenoic, col = pam.out$cluster, pch =
19)
```

```
par(mfrow=c(1,1))
```



In questo caso i grafici in cui si nota meglio la divisione in cluster sono linoleic-palmitoleic e palmitic-palmitoleic

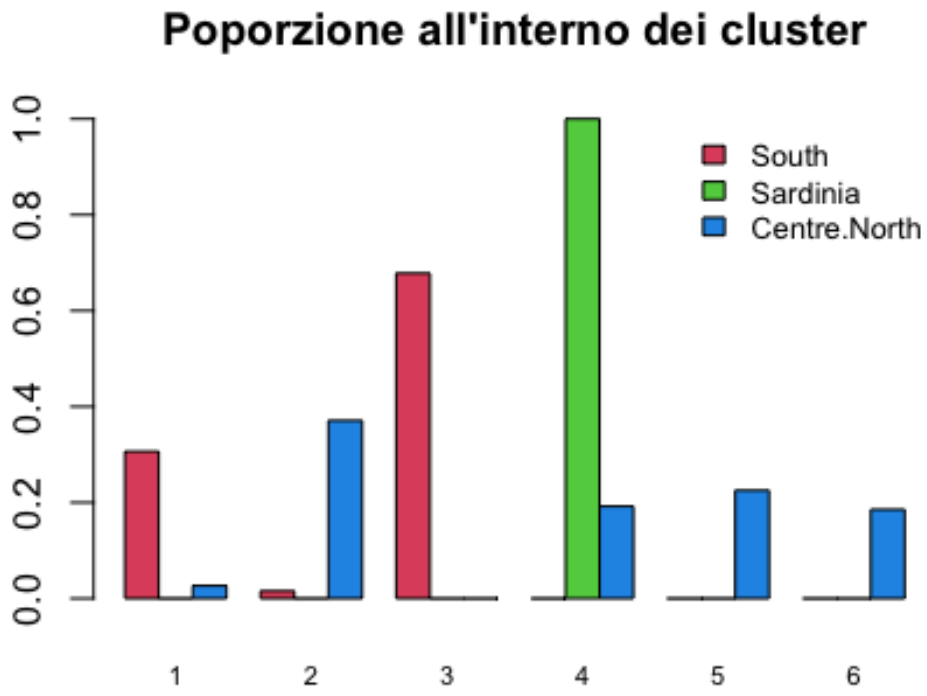
Variabile macro.area

```
prop.table(table(oliveALR$macro.area, pam.out$cluster),1)
```

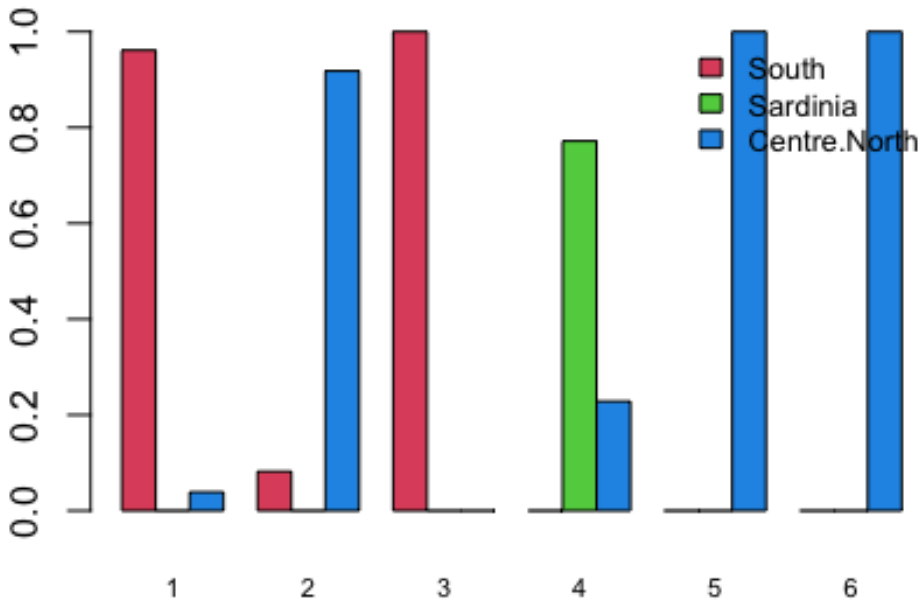
```
##
##                          1          2          3          4          5
##   South         0.30650155 0.01547988 0.67801858 0.00000000 0.00000000
##   Sardinia      0.00000000 0.00000000 0.00000000 1.00000000 0.00000000
##   Centre.North  0.02649007 0.37086093 0.00000000 0.19205298 0.22516556
##
##                          6
##   South         0.00000000
##   Sardinia      0.00000000
##   Centre.North  0.18543046
```

```
barplot(prop.table(table(oliveALR$macro.area, pam.out$cluster),1), beside =
T, legend = F, main = "Poporzione all'interno dei cluster", col = 2:4,
cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveALR$macro.area,
```

```
pam.out$cluster),1)
), fill = 2:5, cex = 0.8, bty = "n")
```




```
barplot(prop.table(table(oliveALR$macro.area, pam.out$cluster),2), beside =
T, legend = F, main = "", col = 2:4, cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveALR$macro.area,
pam.out$cluster),1)
), fill = 2:5, cex = 0.8, bty = "n")
```



Dai grafici si nota un netto miglioramento nel raggruppamento dei cluster di pam ALR ripetto agli altri algoritmi, se confrontati con i gruppi creati dalle macro areee. Gli oli della sardegna vengono inseriti interamente nel cluster 4. Gli oli del sud vengono inseriti per il 30% nel cluster 1 e 67% nel 3 Gli oli del centro nord invece non sono ben identificati e vengono smistati nei cluster 2, 4, 5 e 6

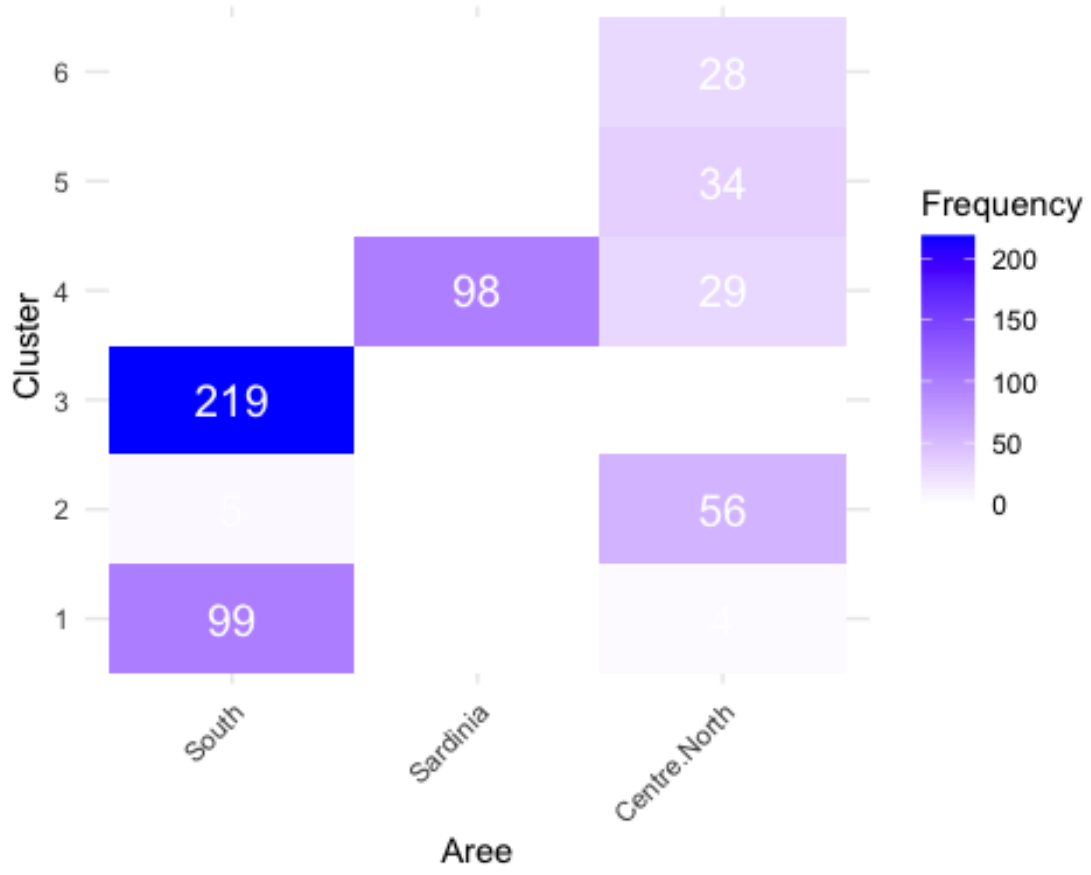
Per quanto riguarda i cluster invece, il primo e il terzo contegono oli provenienti quasi escusivamente dal sud Il quinto e il sesto contengono escusivamente oli del centro nord

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveALR$macro.area, Aree =
pam.out$cluster)

table( Aree = pam.out$cluster, Cluster = oliveALR$macro.area)

##     Cluster
## Aree South Sardinia Centre.North
##    1    99        0            4
##    2     5        0           56
##    3   219        0            0
##    4     0       98           29
```

```
##    5     0        0           34
##    6     0        0           28
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Aree, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Aree", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



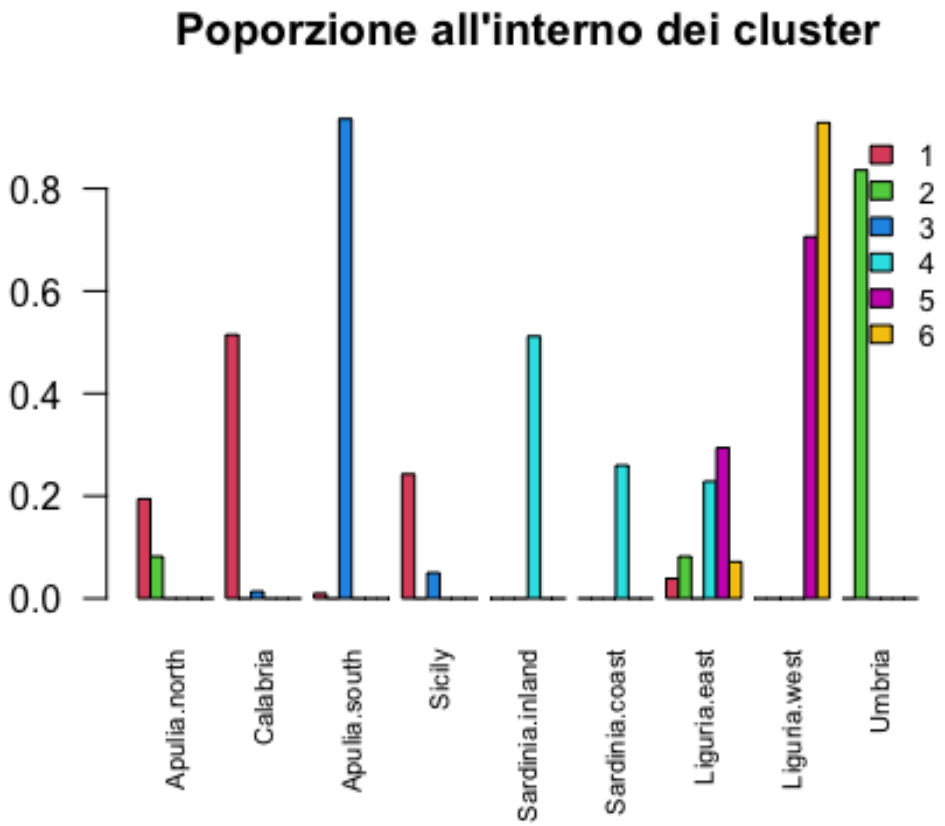
Variabile region

```
prop.table(table(pam.out$cluster, oliveALR$region),1)
```

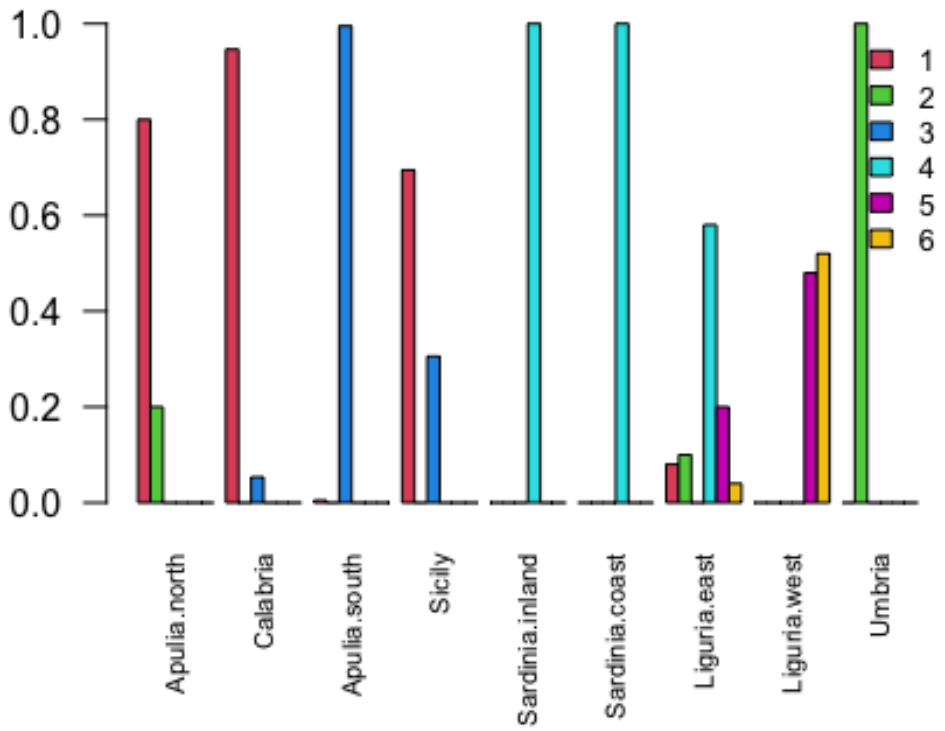
```
##
##      Apulia.north    Calabria Apulia.south      Sicily Sardinia.inland
##   1  0.194174757 0.514563107  0.009708738 0.242718447     0.000000000
##   2  0.081967213 0.000000000  0.000000000 0.000000000     0.000000000
##   3  0.000000000 0.013698630  0.936073059 0.050228311     0.000000000
##   4  0.000000000 0.000000000  0.000000000 0.000000000     0.511811024
##   5  0.000000000 0.000000000  0.000000000 0.000000000     0.000000000
##   6  0.000000000 0.000000000  0.000000000 0.000000000     0.000000000
```

```
##
##     Sardinia.coast Liguria.east Liguria.west      Umbria
##   1    0.000000000  0.038834951  0.000000000 0.000000000
##   2    0.000000000  0.081967213  0.000000000 0.836065574
##   3    0.000000000  0.000000000  0.000000000 0.000000000
##   4    0.259842520  0.228346457  0.000000000 0.000000000
##   5    0.000000000  0.294117647  0.705882353 0.000000000
##   6    0.000000000  0.071428571  0.928571429 0.000000000
```

```
barplot(prop.table(table(pam.out$cluster, oliveALR$region),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:7, cex.names
= 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(pam.out$cluster,
oliveALR$region),1)), fill = 2:7, cex = 0.8, bty = "n")
```




```
barplot(prop.table(table(pam.out$cluster, oliveALR$region),2), beside = T,
legend = F, main = "", col = 2:7, cex.names = 0.70, las=2)
legend("topright", legend = rownames(prop.table(table(pam.out$cluster,
oliveALR$region),1)), fill = 2:7, cex = 0.8, bty = "n")
```



Oltre alle precedenti osservazioni sulla macro area Sardegna (che sono ocnfermate dall'analisi riapetto alle regioni) si nota che: Gli oli della puglia sud vengono inseriti interamente nel cluster 3 Gli oli dell'umbria vengono inseriti interamente nel custer 2, che non contiene oli provenienti da altre regioni Gli oli della liguria est sono smistati in diversi cluster

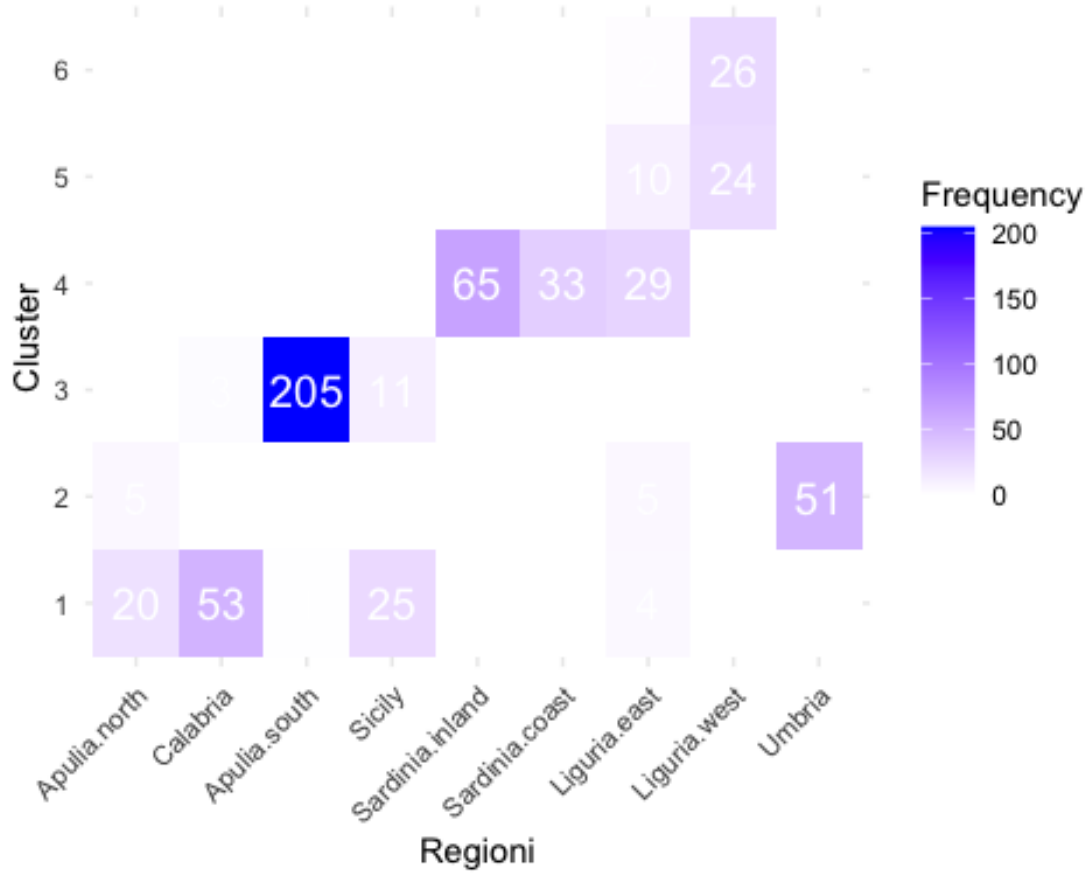
Guardando i cluster invece possiamo affermare che il cluster 1 contiene oli della puglia nord, calabria e sicilia il cluster 2 contiene oli della dell'umbria e in parte della puglia nord il cluster 3 contiene oli della puglia sud, e in parte della sicilia il cluster 4 contiene oli della sardegna e liguria est il cluster 5 e 6 contiene esclusivamente oli della liguria

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveALR$region, Regioni =
pam.out$cluster)

table(Regioni = pam.out$cluster, Cluster = oliveALR$region)

##         Cluster
## Regioni Apulia.north Calabria Apulia.south Sicily Sardinia.inland
##       1           20       53            1     25               0
##       2            5        0            0      0               0
##       3            0        3          205     11               0
```

```
##       4            0        0           0      0              65
##       5            0        0           0      0               0
##       6            0        0           0      0               0
##        Cluster
## Regioni Sardinia.coast Liguria.east Liguria.west Umbria
##       1              0            4            0      0
##       2              0            5            0     51
##       3              0            0            0      0
##       4             33           29            0      0
##       5              0           10           24      0
##       6              0            2           26      0
```

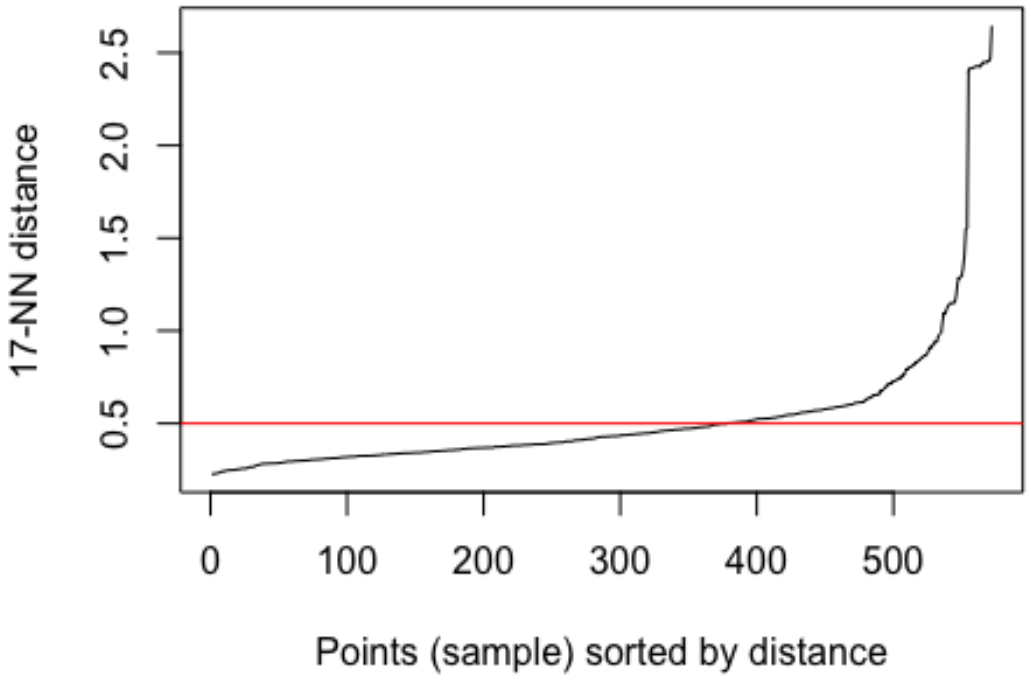
```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Regioni, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Regioni", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



DB SCAN con ALR

Come visto prima, si testa l'algoritmo DBSCAN attraverso la funzione kNNdistplot() e si analizza il grafico ottenuto.

```
kNNdistplot(oliveALR[,3:9], k = 17)
abline(h=0.5, col = "red")
```



I parametri migliori trovati sono raggio 0.5 e punti minimi 18

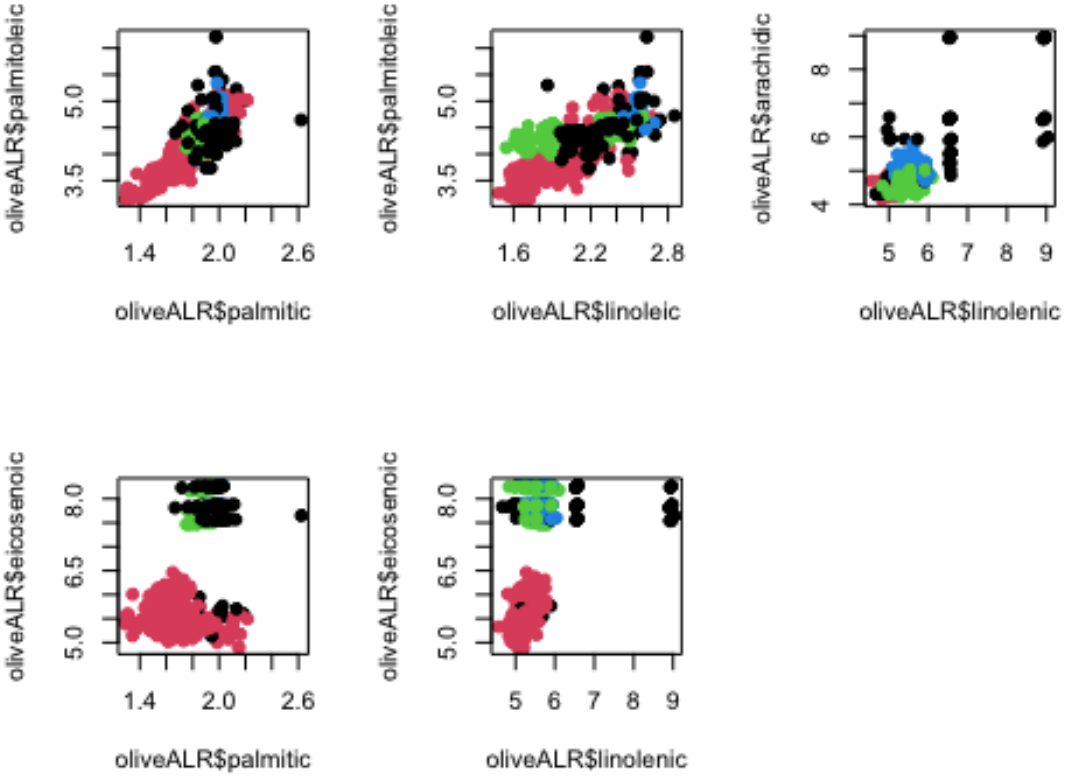
```
set.seed(17)

db.out <- dbscan(oliveALR[,3:9], eps = 0.5, minPts = 18)
str(db.out)

## List of 5
##  $ cluster     : int [1:572] 1 1 0 1 1 1 1 1 1 0 ...
##  $ eps         : num 0.5
##  $ minPts      : num 18
##  $ dist        : chr "euclidean"
##  $ borderPoints: logi TRUE
##  - attr(*, "class")= chr [1:2] "dbscan_fast" "dbscan"

db.out
```

```
## DBSCAN clustering for 572 objects.
## Parameters: eps = 0.5, minPts = 18
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 3 cluster(s) and 87 noise points.
##
##   0   1   2   3
##  87 315 125  45
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

dbscan() ci diviede i dati in cluster, in questo caso 3 con 87 punti di "noise".

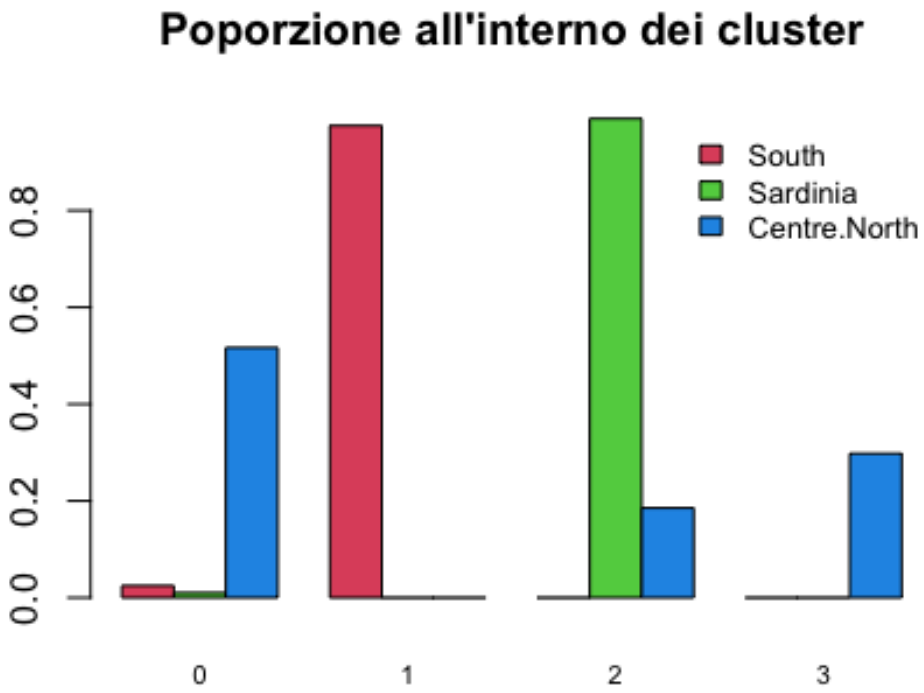
Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

```
par(mfrow=c(2,3))

# palmitic palmitoleic
plot(oliveALR$palmitic, oliveALR$palmitoleic, col = db.out$cluster+1, pch =
19)

# linoleic palmitoleic
plot(oliveALR$linoleic, oliveALR$palmitoleic, col = db.out$cluster+1, pch =
19)

# arachidic linolenic
plot(oliveALR$linolenic, oliveALR$arachidic, col = db.out$cluster+1, pch =
19)

# eicosenoic palmitic
plot(oliveALR$palmitic, oliveALR$eicosenoic, col = db.out$cluster+1, pch =
19)

# eicosenoic linolenic
plot(oliveALR$linolenic, oliveALR$eicosenoic, col = db.out$cluster+1, pch =
19)

par(mfrow=c(1,1))
```

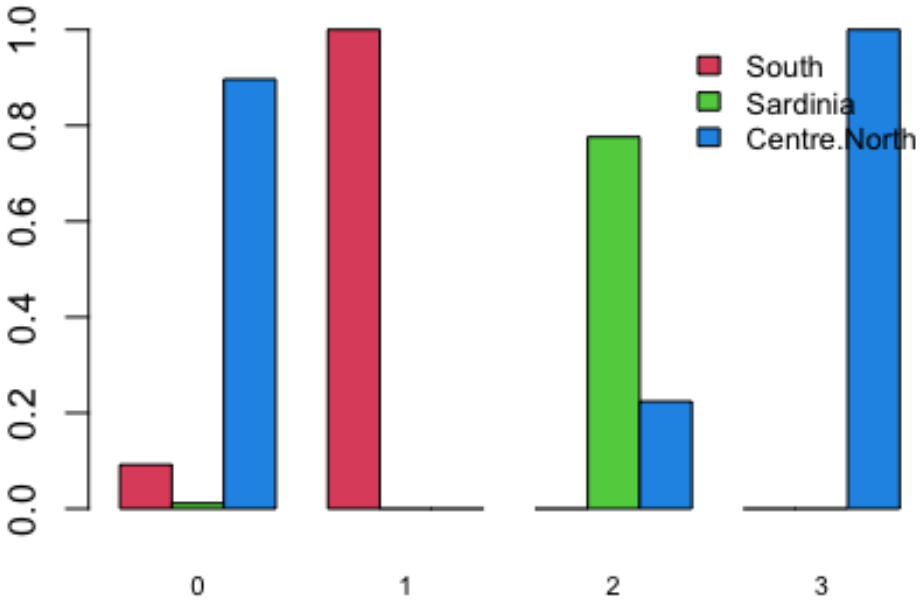


Variabile macro.area

```
prop.table(table(db.out$cluster, oliveALR$macro.area),1)

##
##          South   Sardinia Centre.North
##   0 0.09195402 0.01149425   0.89655172
##   1 1.00000000 0.00000000   0.00000000
##   2 0.00000000 0.77600000   0.22400000
##   3 0.00000000 0.00000000   1.00000000

barplot(prop.table(table(oliveALR$macro.area, db.out$cluster),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:4, cex.names
= 0.70)
legend("topright", legend = rownames(prop.table(table(oliveALR$macro.area,
db.out$cluster),1)), fill = 2:4, cex = 0.8, bty = "n")
```




```r
barplot(prop.table(table(oliveALR$macro.area, db.out$cluster),2), beside = T,
legend = F, main = "", col = 2:4, cex.names = 0.70)
legend("topright", legend = rownames(prop.table(table(oliveALR$macro.area,
db.out$cluster),1)), fill = 2:4, cex = 0.8, bty = "n")
```



Si nota che i tre cluster contengono tutti prevalentemente un area geografica diversa: - il cluster 1 contiene solo oli del Sud - il cluster 2 è principalmente composto da oli della Sardegna - il cluster 3 è formato da oli provenienti solamente dal Centro Nord Gli oli del centro nord sono comunque distribuiti tra il cluster 2, 3 e sono anche presenti in gran parte nell'insieme sei punti di "noise"

Paragone con dati non trasformati:

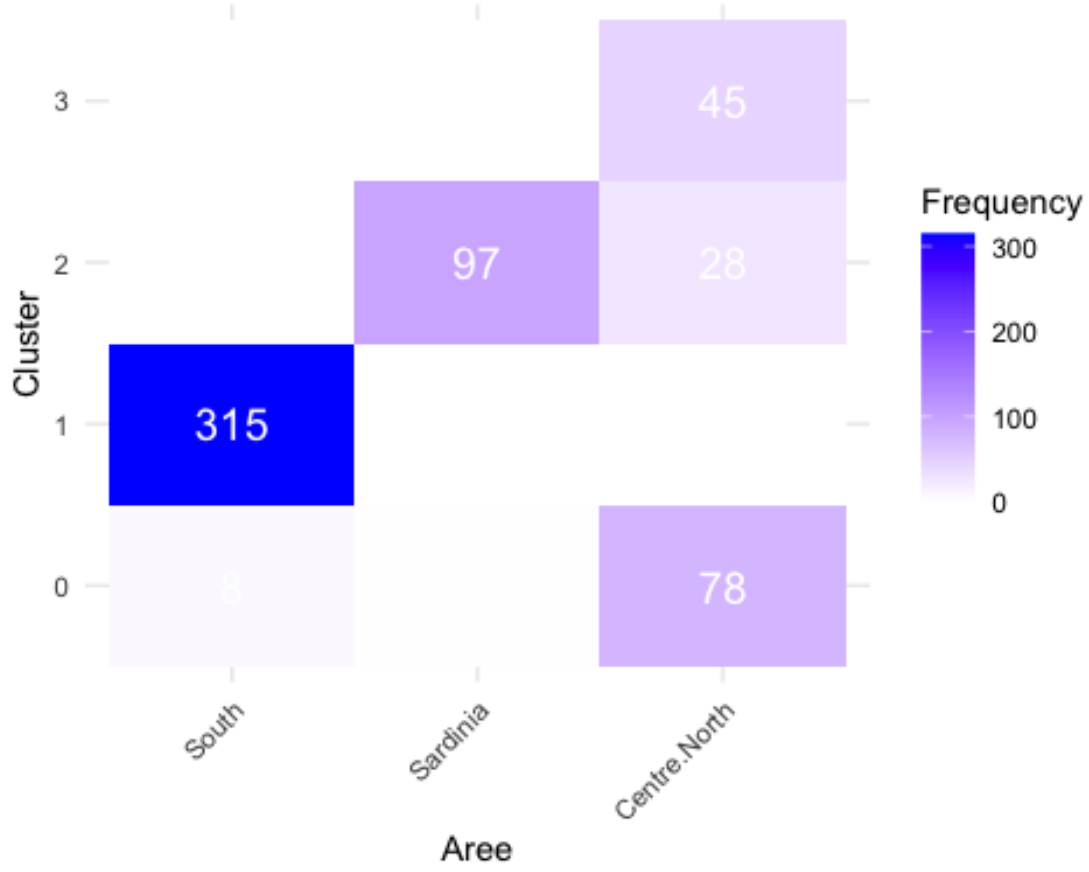
Con la trasformazione logaritmica notiamo che gli oli del Sud e della Sardegna hanno una distribuzione migliore tra i cluster, infatti questi solo quasi esclusivamente appartenenti ad un singolo cluster ciascuno. Per gli oli del Nord/Centro è invece accaduto il contrario, nella distribuzione in cluster senza trasformazione questi sono concentrati in un singolo cluster mentre con la trasformazione si dividono in 2 cluster con una gran parte di essi che sono considerati punti di "noise"

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveALR$macro.area, Aree =
db.out$cluster)

table( Aree = db.out$cluster, Cluster = oliveALR$macro.area)
```

```
##     Cluster
## Aree South Sardinia Centre.North
##    0     8        1           78
##    1   315        0            0
##    2     0       97           28
##    3     0        0           45
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Aree, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Aree", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



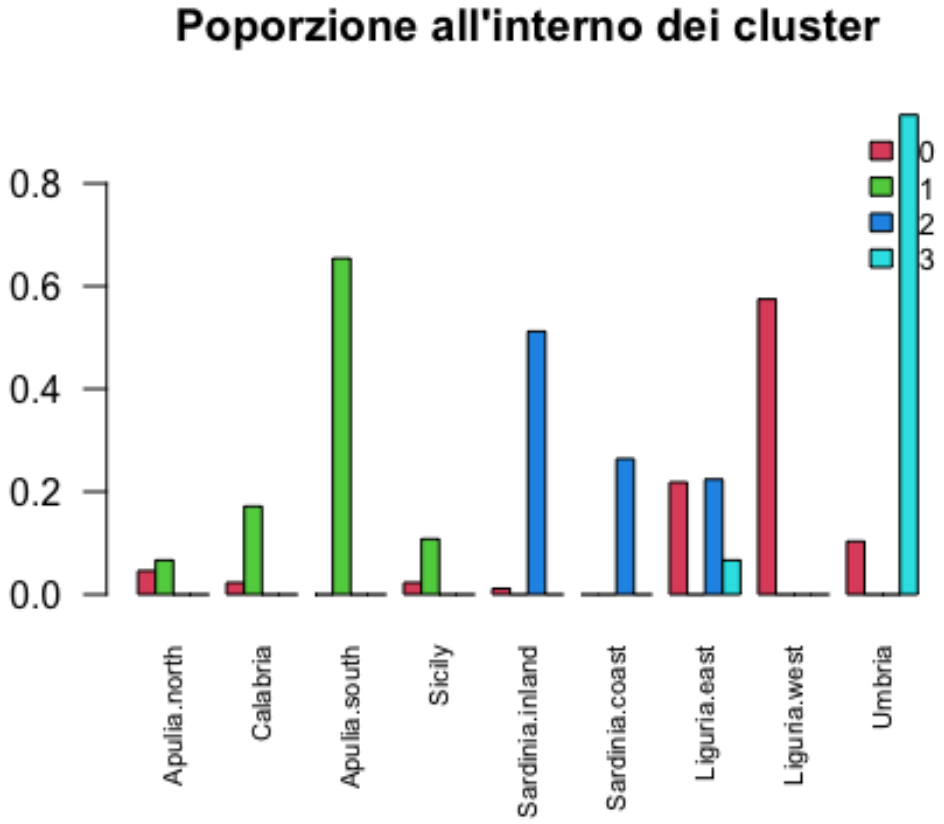
Variabile region

```
prop.table(table(db.out$cluster, oliveALR$region),1)
```

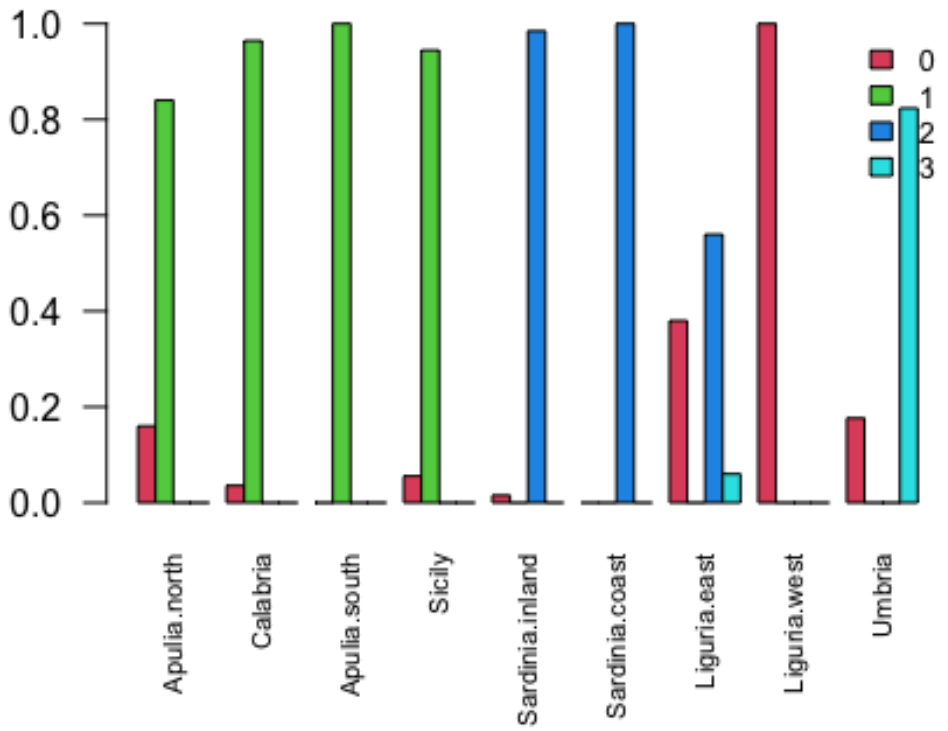
```
##
##     Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   0   0.04597701 0.02298851   0.00000000 0.02298851      0.01149425
##   1   0.06666667 0.17142857   0.65396825 0.10793651      0.00000000
```

```
##   2   0.00000000 0.00000000   0.00000000 0.00000000      0.51200000
##   3   0.00000000 0.00000000   0.00000000 0.00000000      0.00000000
##
##     Sardinia.coast Liguria.east Liguria.west     Umbria
##   0     0.00000000   0.21839080   0.57471264 0.10344828
##   1     0.00000000   0.00000000   0.00000000 0.00000000
##   2     0.26400000   0.22400000   0.00000000 0.00000000
##   3     0.00000000   0.06666667   0.00000000 0.93333333
```

```r
counts <- prop.table(table(db.out$cluster, oliveALR$region), 1)

barplot(prop.table(table(db.out$cluster, oliveALR$region),1), beside = T,
legend = F, main = "Poporzione all'interno dei cluster", col = 2:5, cex.names
= 0.70, las=2)
legend("topright", legend = rownames(counts), fill = 2:5, cex = 0.8, bty =
"n")
```




```r
barplot(prop.table(table(db.out$cluster, oliveALR$region),2), beside = T,
legend = F, main = "", col = 2:5, cex.names = 0.70, las=2)
legend("topright", legend = rownames(counts), fill = 2:5, cex = 0.8, bty =
"n")
```



Notiamo che in questo caso, rimossi i valori di “noise”, la distribuzione dei cluster tra le regioni è incredibilmente precisa: * Oli dalla Puglia, Calabria e Sicilia sono esclusivamente presenti nel cluster 1 * Oli da Sardegna e Liguria dell’est sono quasi totalmente nel cluster 2 * Infine, gli oli umbri sono tutti nel cluster 3 Notiamo però che gli oli provenienti dalla Liguria dell’ovest sono tutti considerati punti di “noise”

Paragone con dati non trasformati:

Si vede che il metodo con trasformazione logaritmica riesce a raggruppare con più precisone oli provenienti dalla stessa regione all’interno di un singolo cluster. I gruppi di regioni che andavano a raggrupparsi in un singolo cluster sono cambiati. Per esempio la Puglia del nord non è più nello stesso cluster con Liguria e Puglia ma adesso è situato le cluster con Calabria, Puglia del sud e Sicilia. Infine, la distribuzione tra cluster degli oli provenienti dalla Liguria dell’ovest peggiora con la trasformazione. Tutti gli oli provenienti da quella regione sono infatti considerati punti di “noise” nel secondo caso.

Confusion Matrix:

```
confusion_matrix <- table(Cluster = oliveALR$region, Regioni =
db.out$cluster)

table(Regioni = db.out$cluster, Cluster = oliveALR$region)
```

```
##         Cluster
## Regioni Apulia.north Calabria Apulia.south Sicily Sardinia.inland
##       0            4        2            0      2                1
##       1           21       54          206     34                0
##       2            0        0            0      0               64
##       3            0        0            0      0                0
##         Cluster
## Regioni Sardinia.coast Liguria.east Liguria.west Umbria
##       0              0           19           50      9
##       1              0            0            0      0
##       2             33           28            0      0
##       3              0            3            0     42
```

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Cluster, y =
Regioni, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), color = "white", size = 5) +
  scale_fill_gradient(low = "white", high = "blue") +
  labs(x = "Regioni", y = "Cluster", fill = "Frequency") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```