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(maps)

##
## Attaching package: 'maps'

## The following object is masked from 'package:cluster':
##
##     votes.repub

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

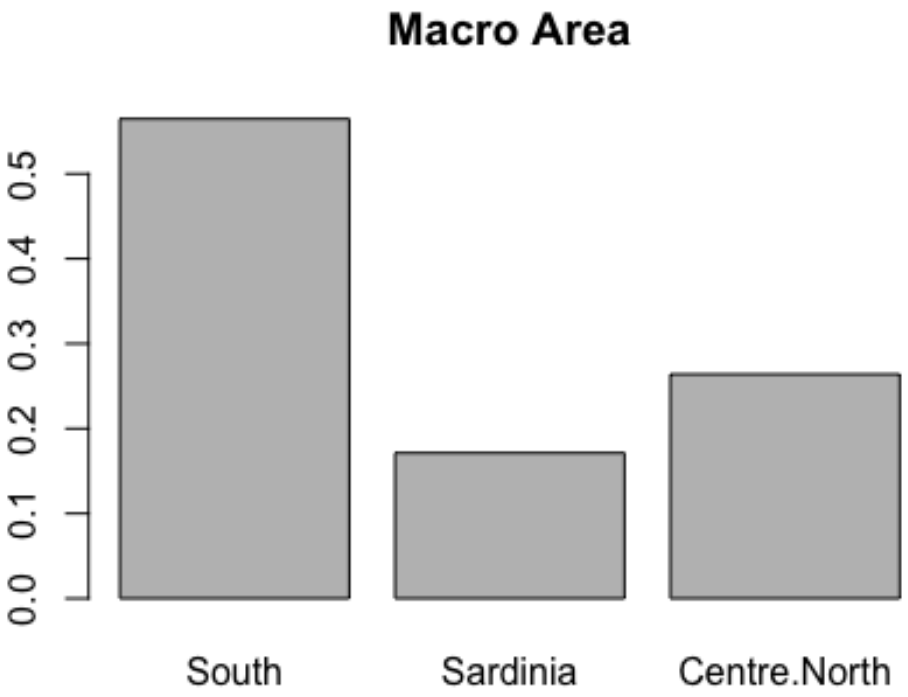
```
data("oliveoil")
str(oliveoil)

## '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

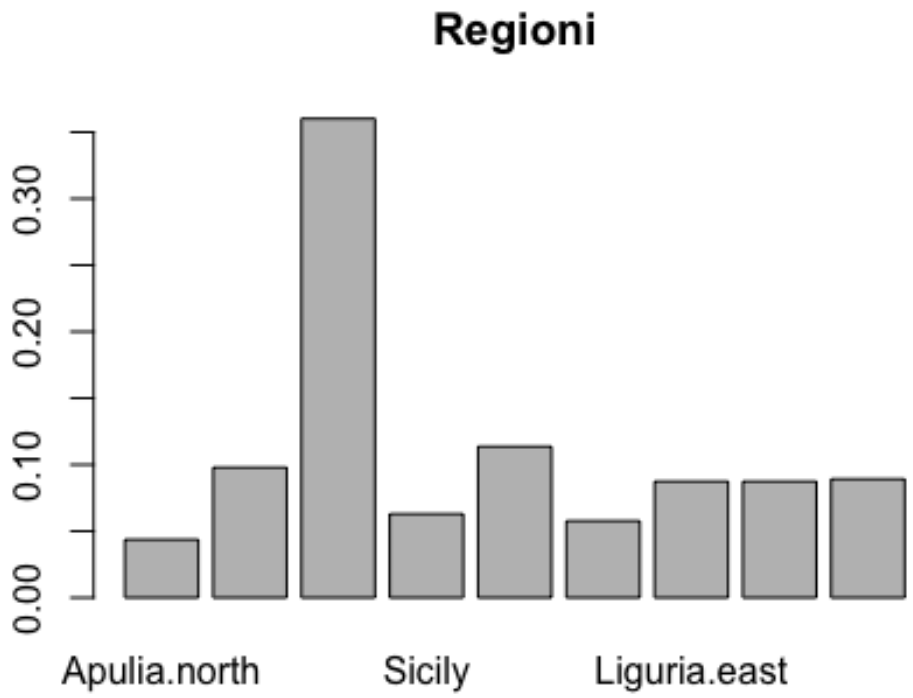
```
display_table(oliveoil$macro.area, "Macro Area")
```




```
##              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
```

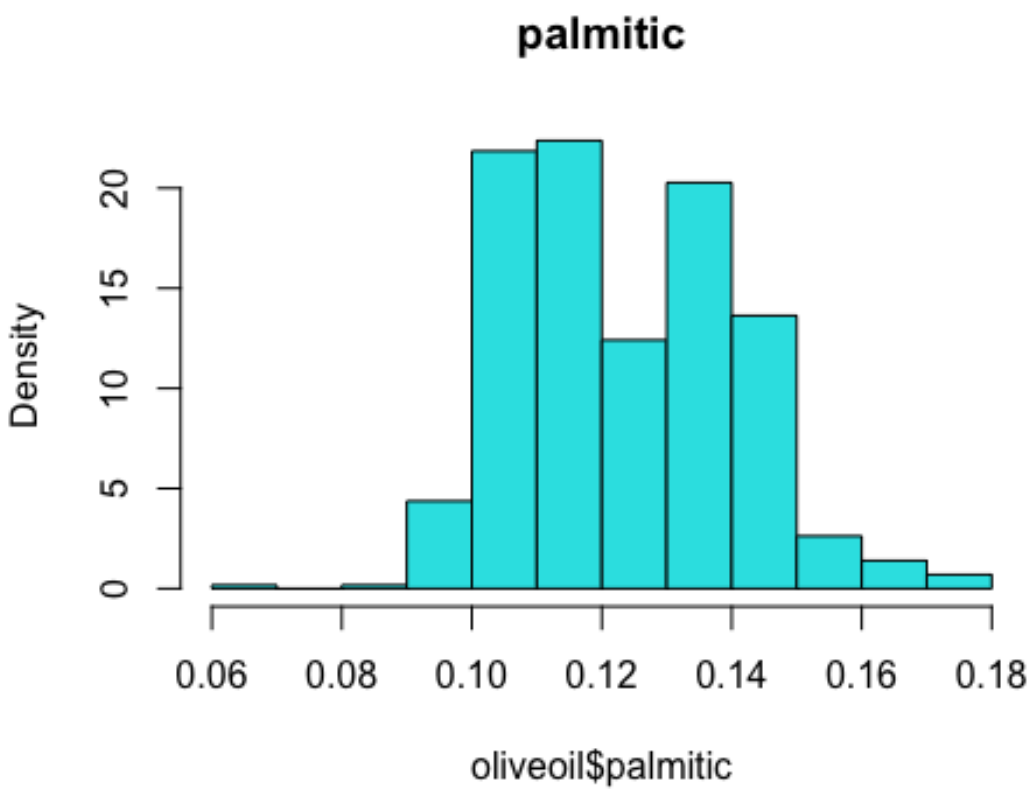
Variabile acido palmitic

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.

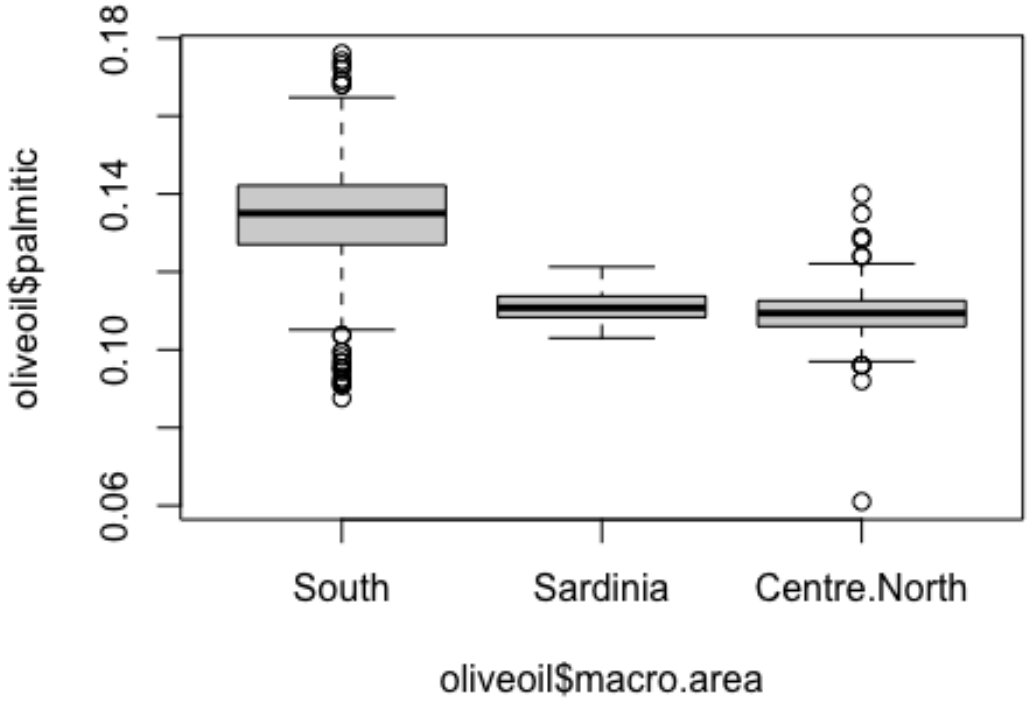
```
display_summary_and_var(oliveoil$palmitic)
```

```
##         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
```

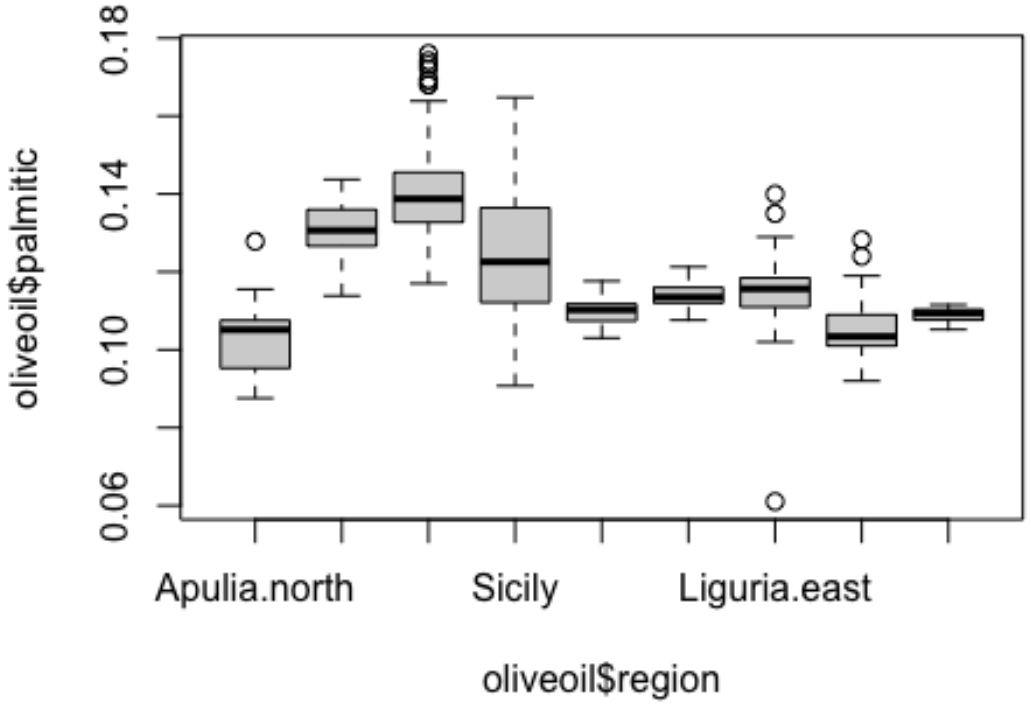
```
hist(oliveoil$palmitic, probability = T,col = 5, main = "palmitic")
```




```
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.

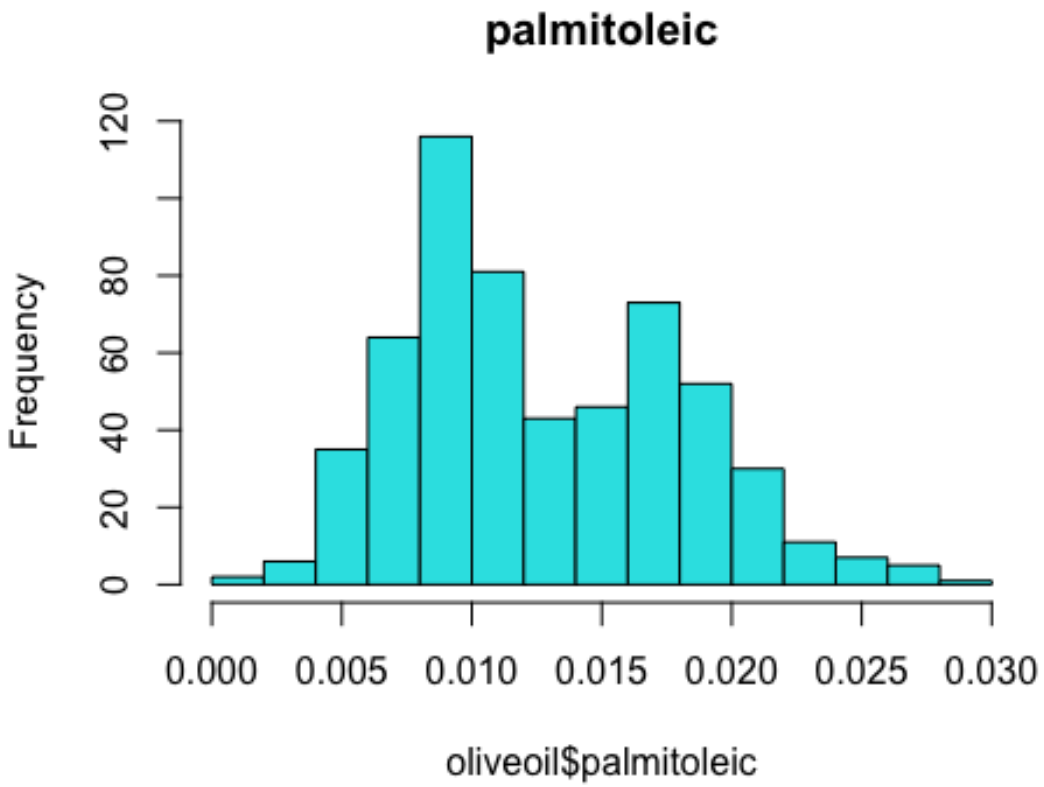
Variabile acido palmitoleic

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.

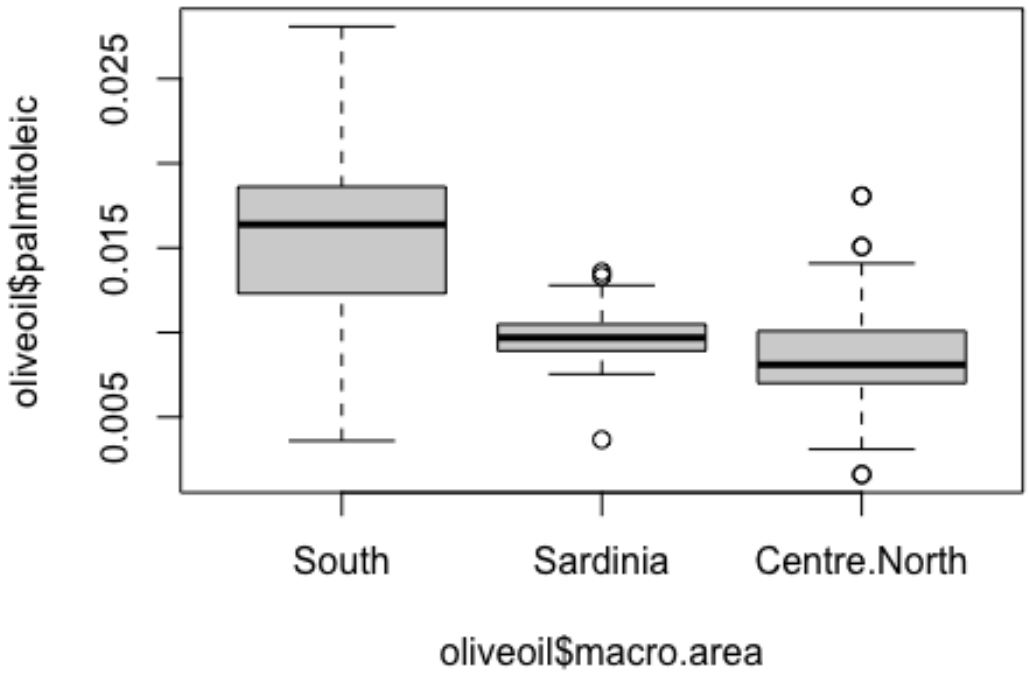
```
display_summary_and_var(oliveoil$palmitoleic)
```

```
##         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
```

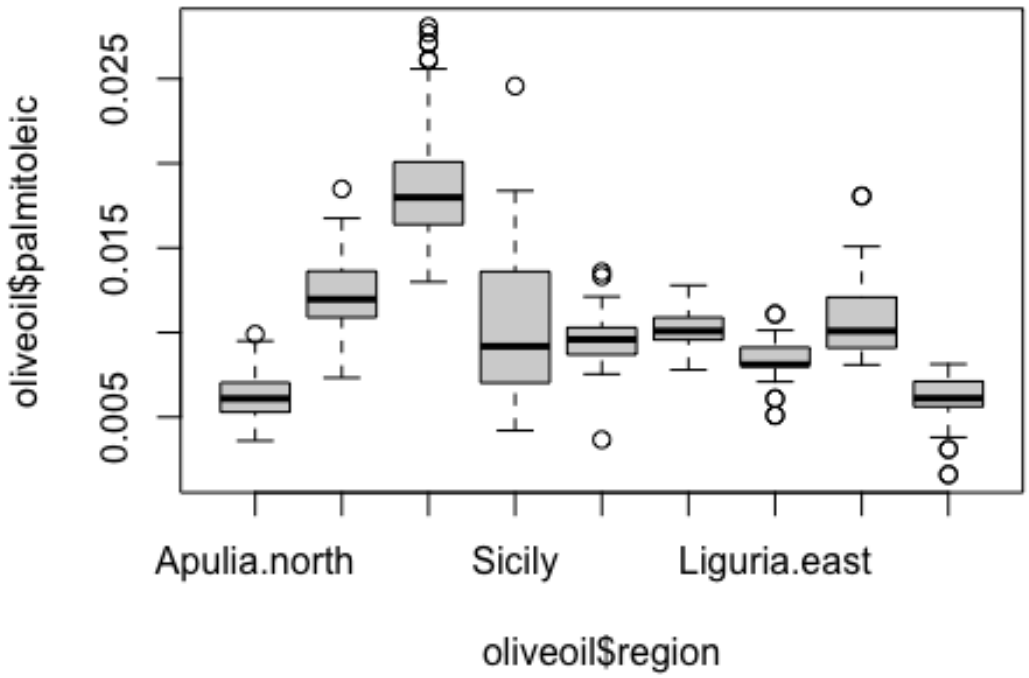
```
hist(oliveoil$palmitoleic,col = 5, main = "palmitoleic")
```




```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.

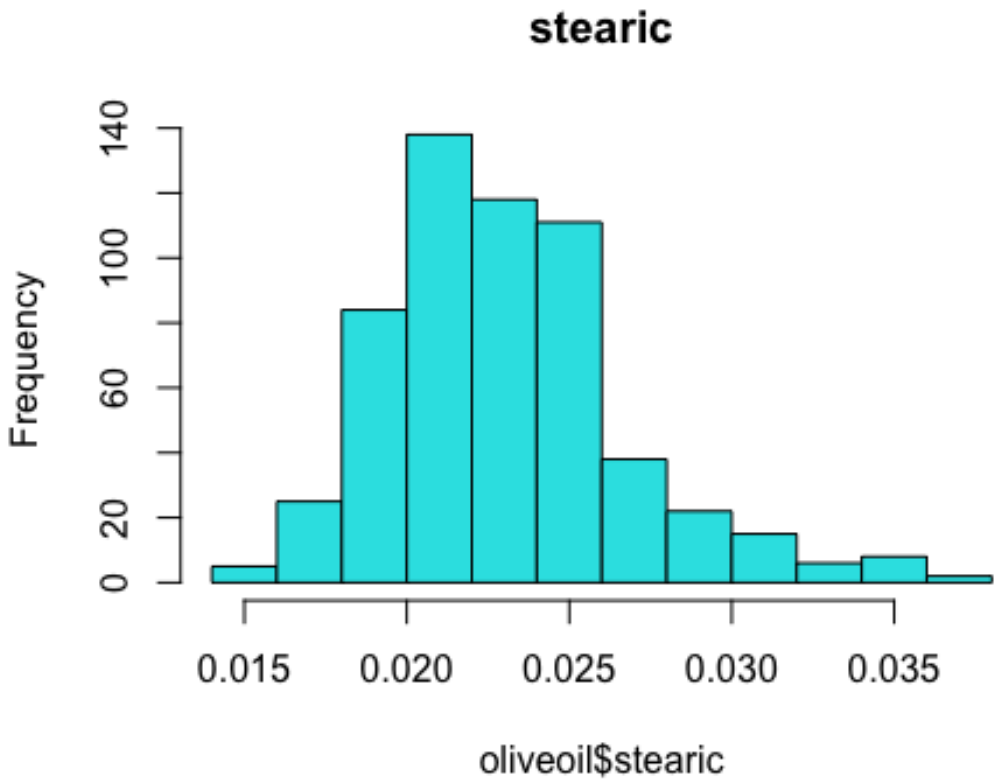
Variabile acido stearic

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.

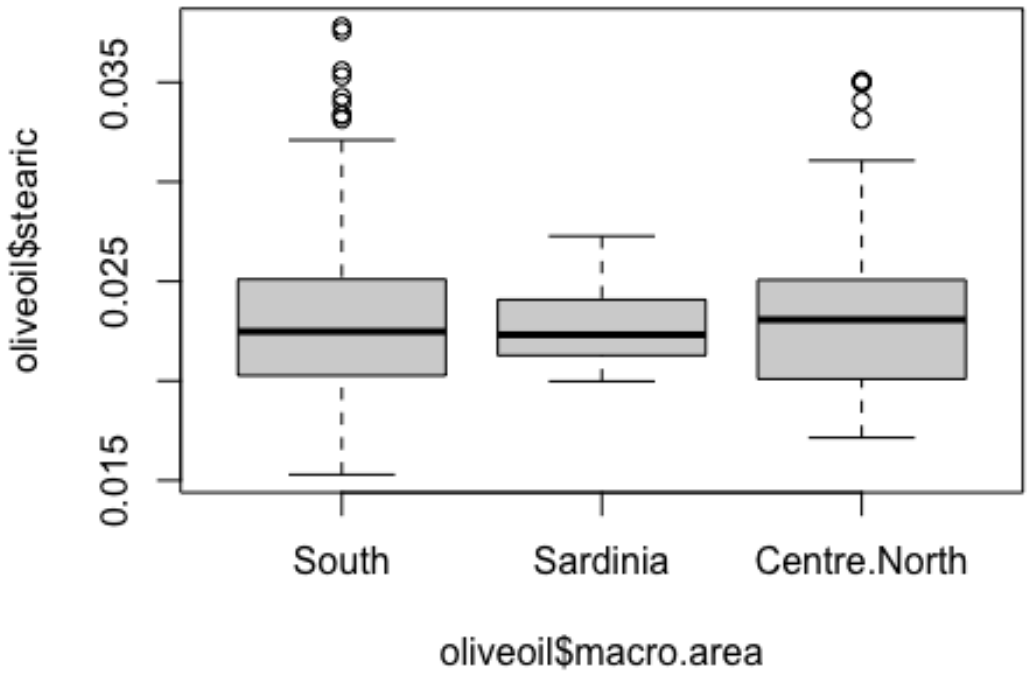
```
display_summary_and_var(oliveoil$stearic)
```

```
##         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
```

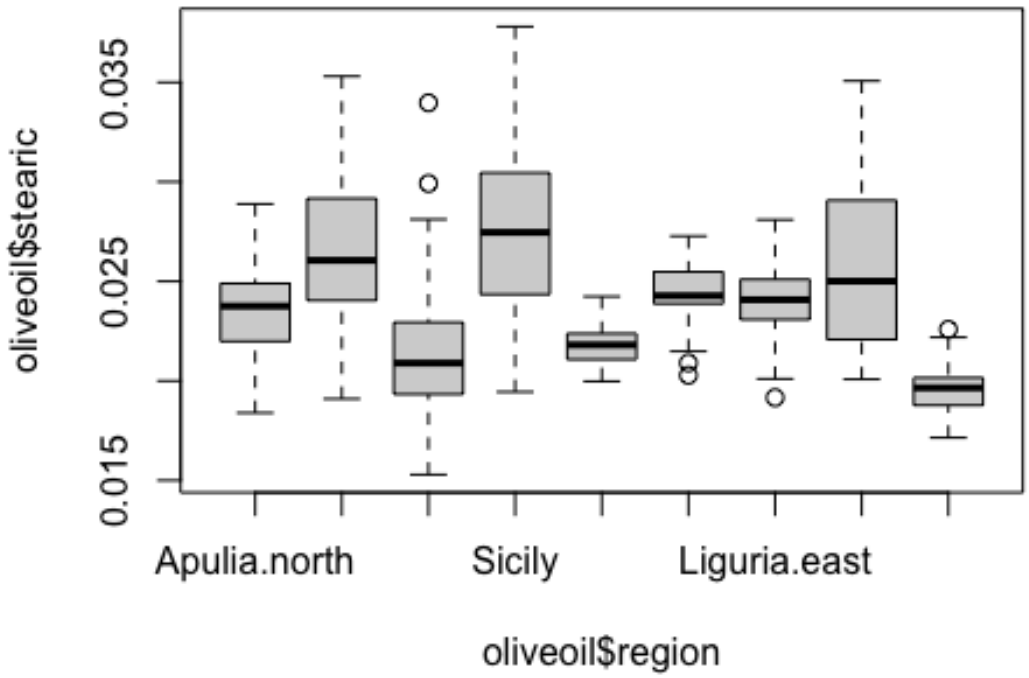
```
hist(oliveoil$stearic, col = 5, main = "stearic")
```




```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.

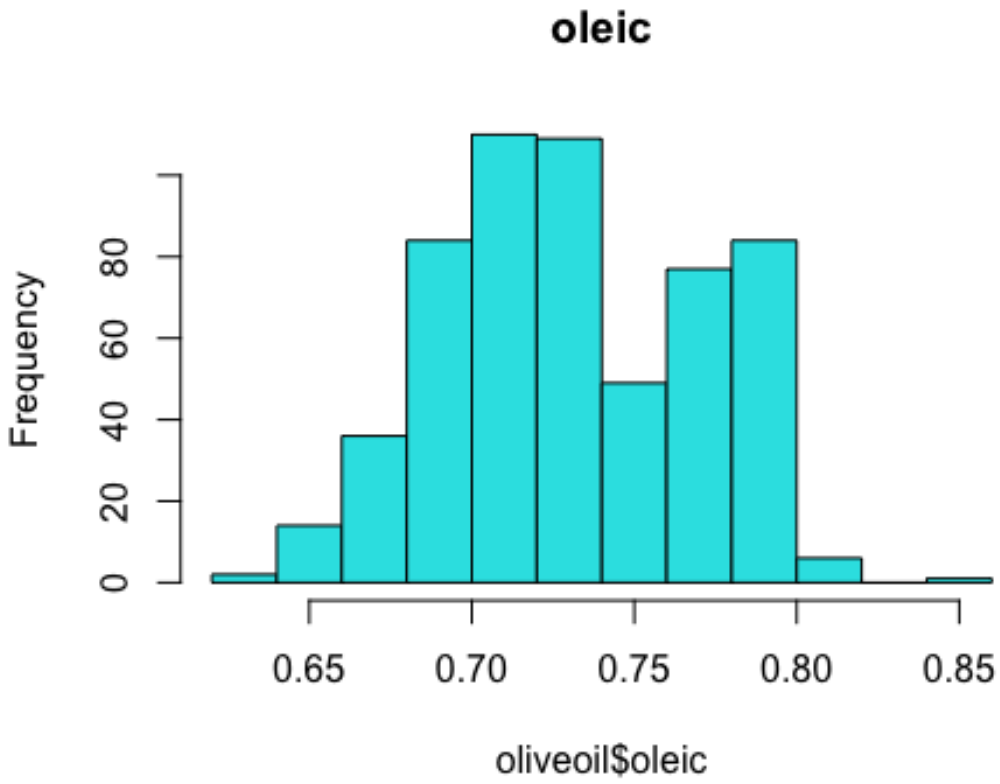
Variabile acido oleic

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.

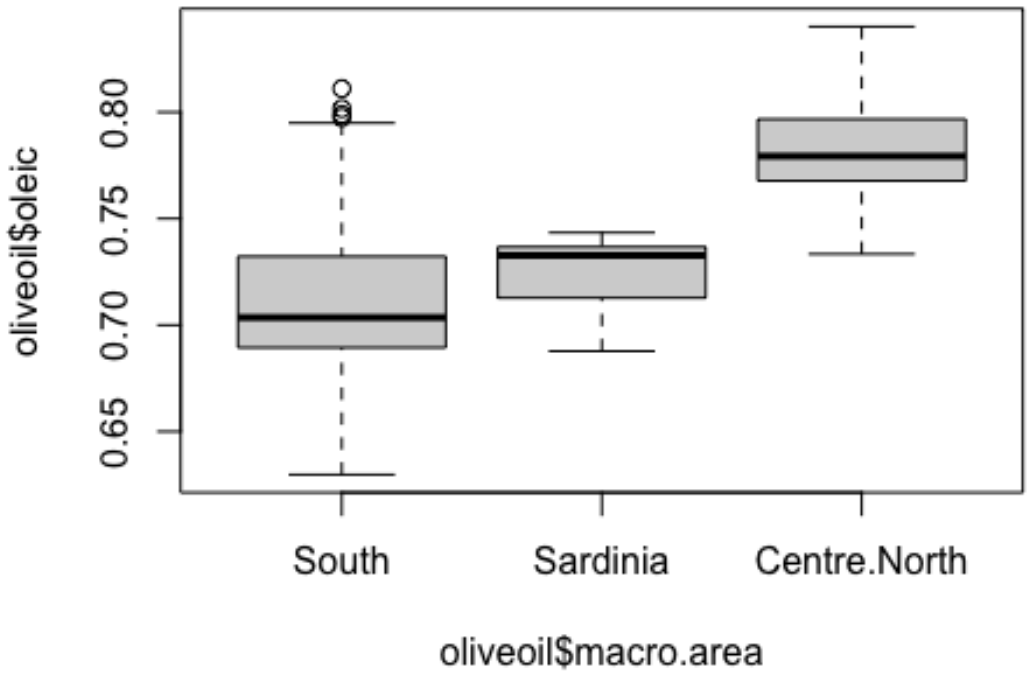
```
display_summary_and_var(oliveoil$oleic)
```

```
##        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
```

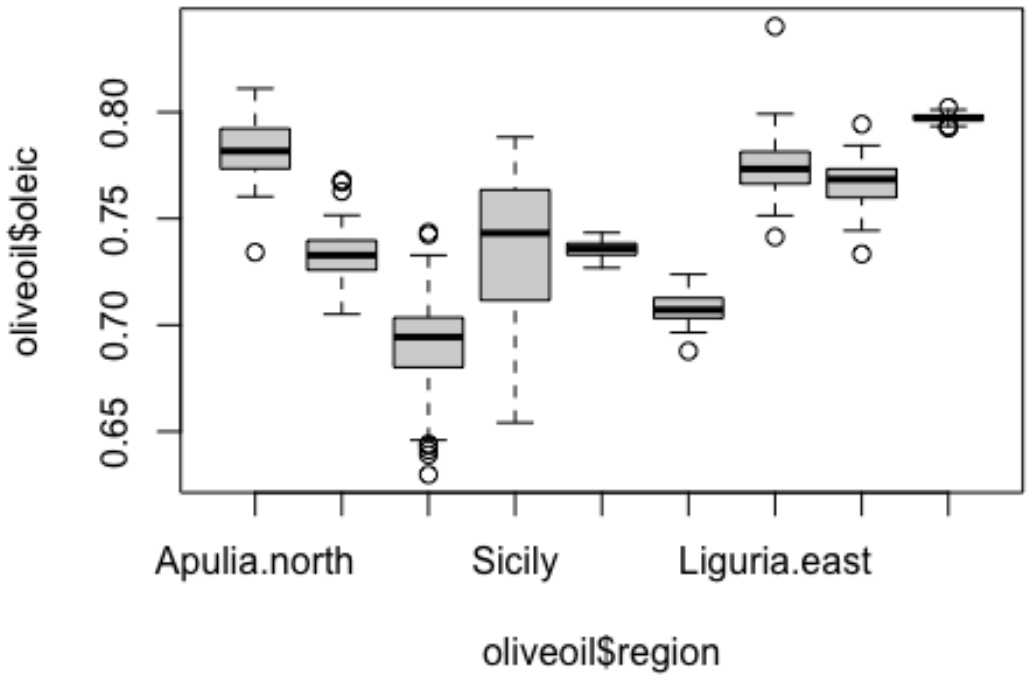
```
hist(oliveoil$oleic, col = 5, main = "oleic")
```




```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.

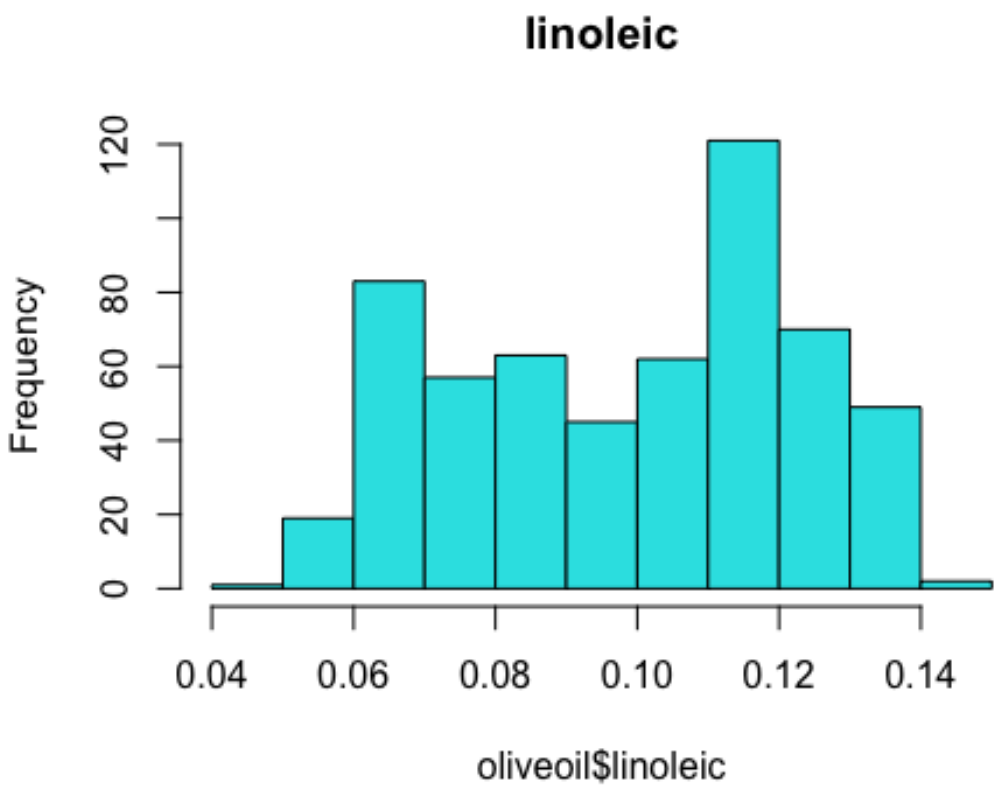
Variabile acido linoleic

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.

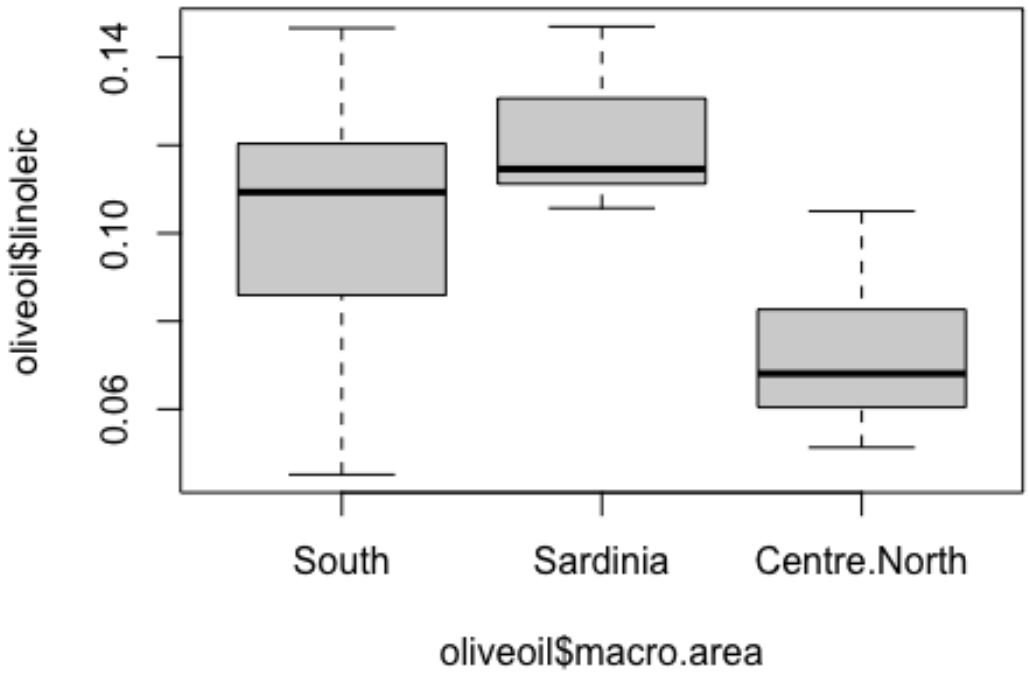
```
display_summary_and_var(oliveoil$linoleic)
```

```
##          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
```

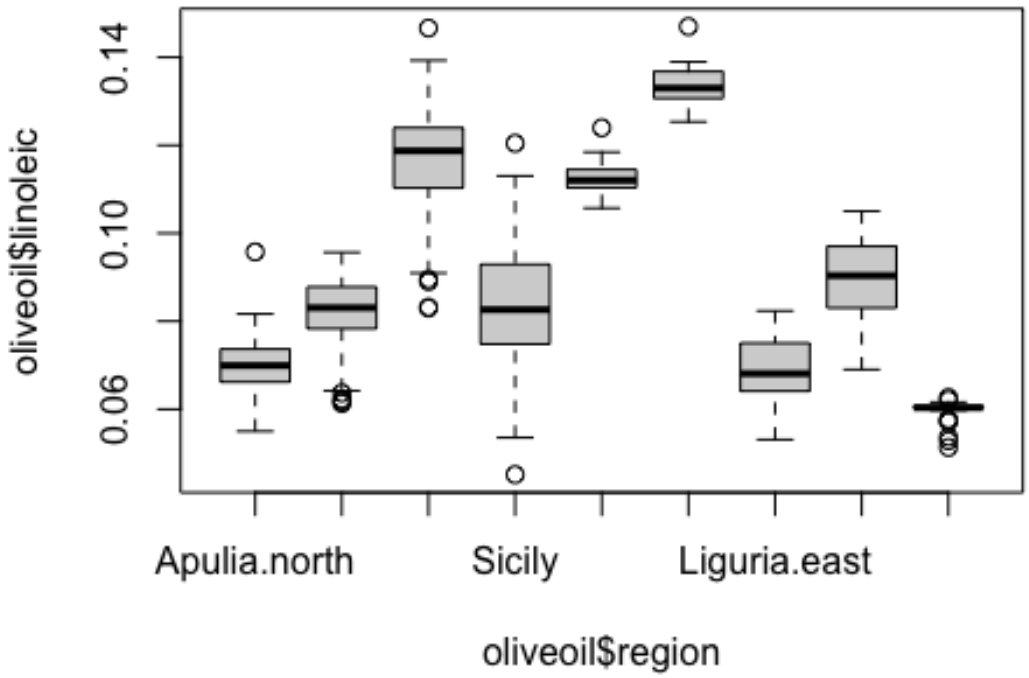
```
hist(oliveoil$linoleic, col = 5, main = "linoleic")
```




```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.

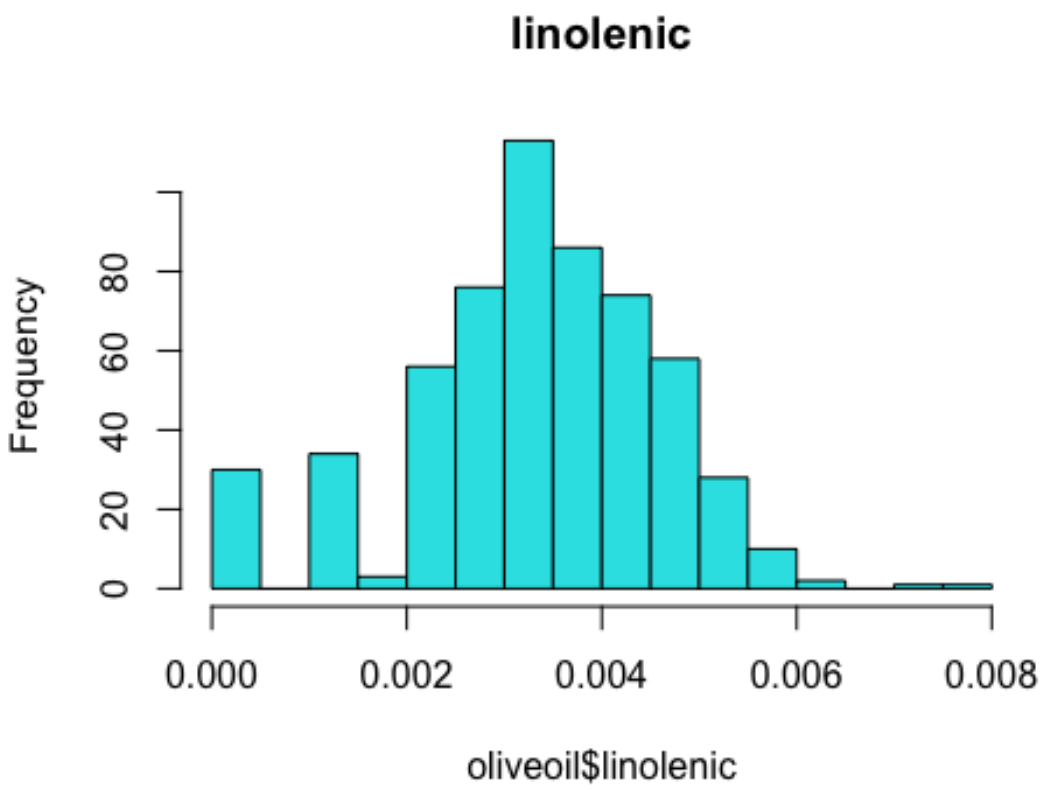
Variabile acido linolenic

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.

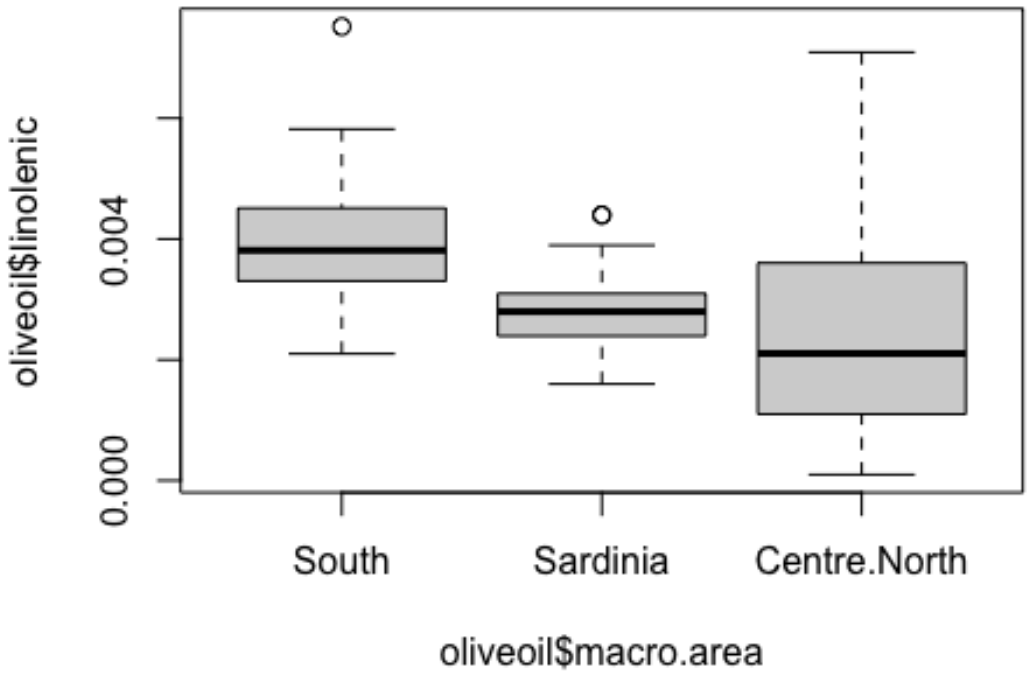
```
display_summary_and_var(oliveoil$linolenic)
```

```
##          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
```

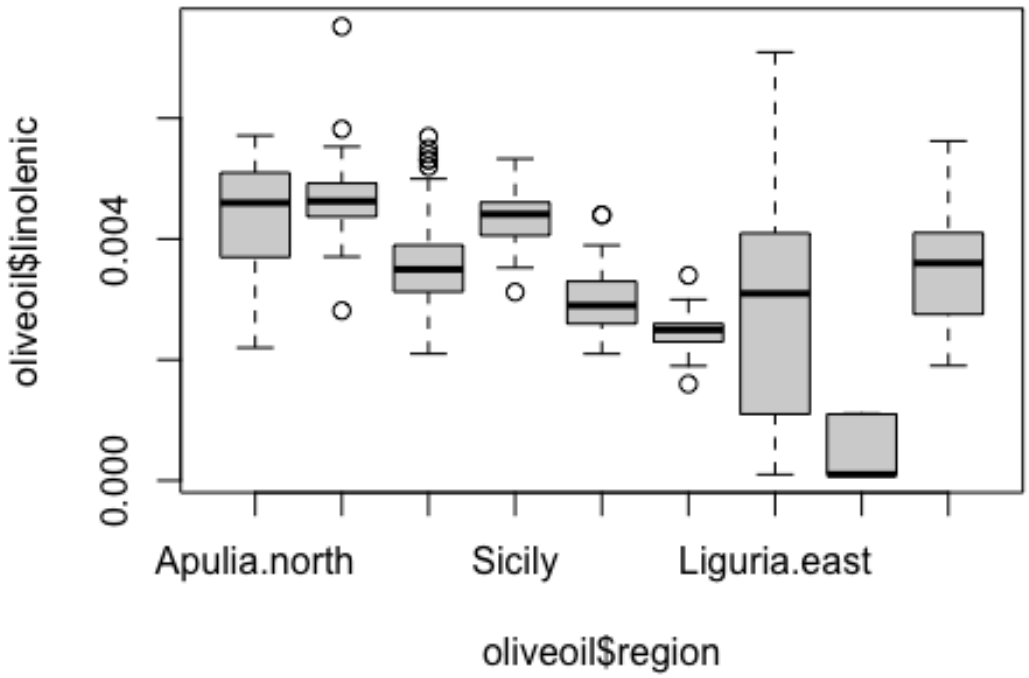
```
hist(oliveoil$linolenic, col = 5, main = "linolenic")
```




```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.

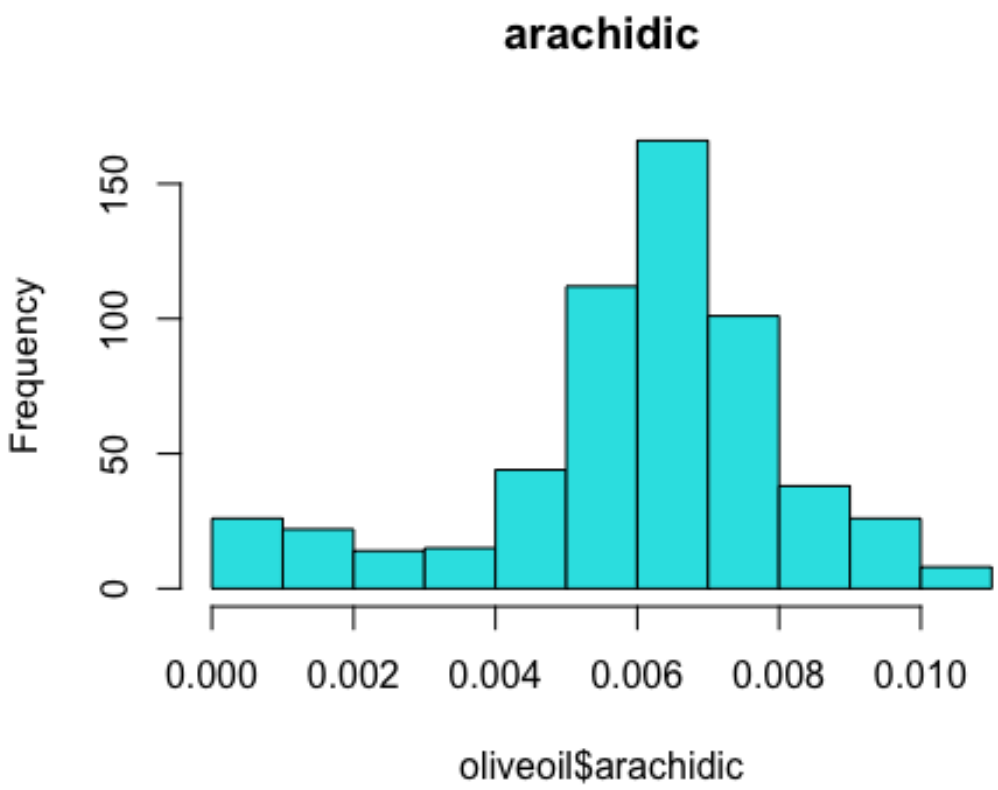
Variabile acido arachidic

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.

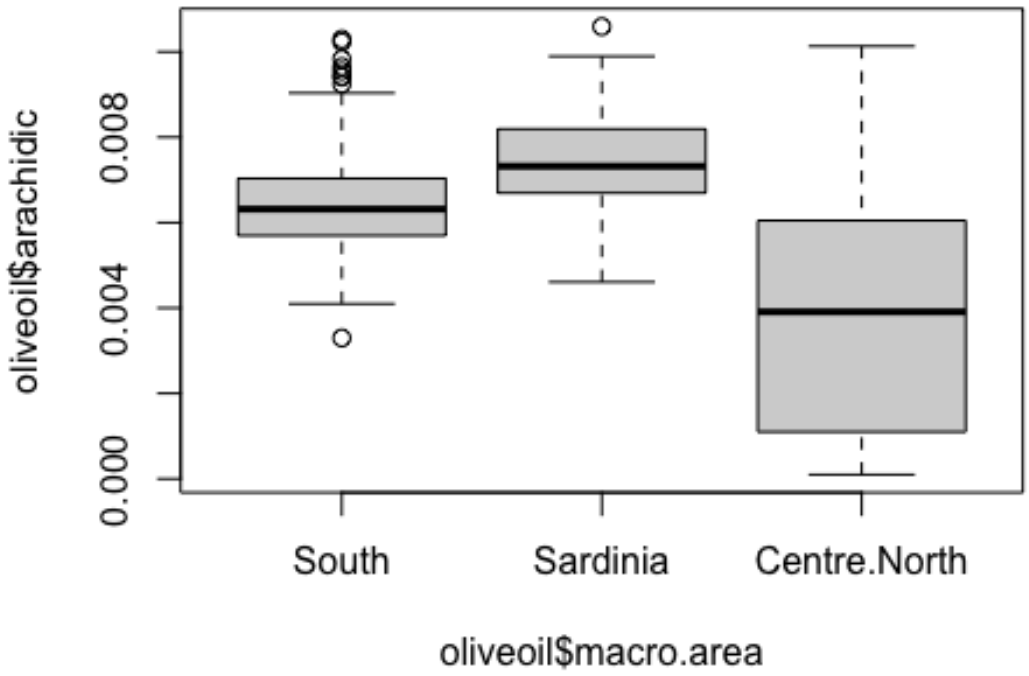
```
display_summary_and_var(oliveoil$arachidic)
```

```
##          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
```

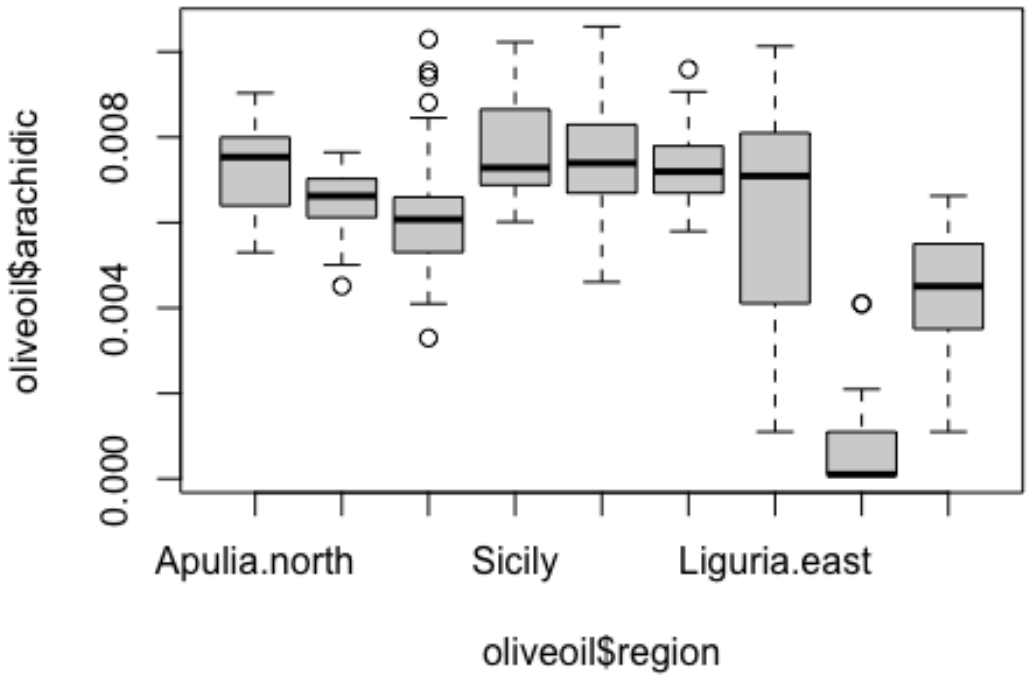
```
hist(oliveoil$arachidic, col = 5, main = "arachidic")
```




```
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.

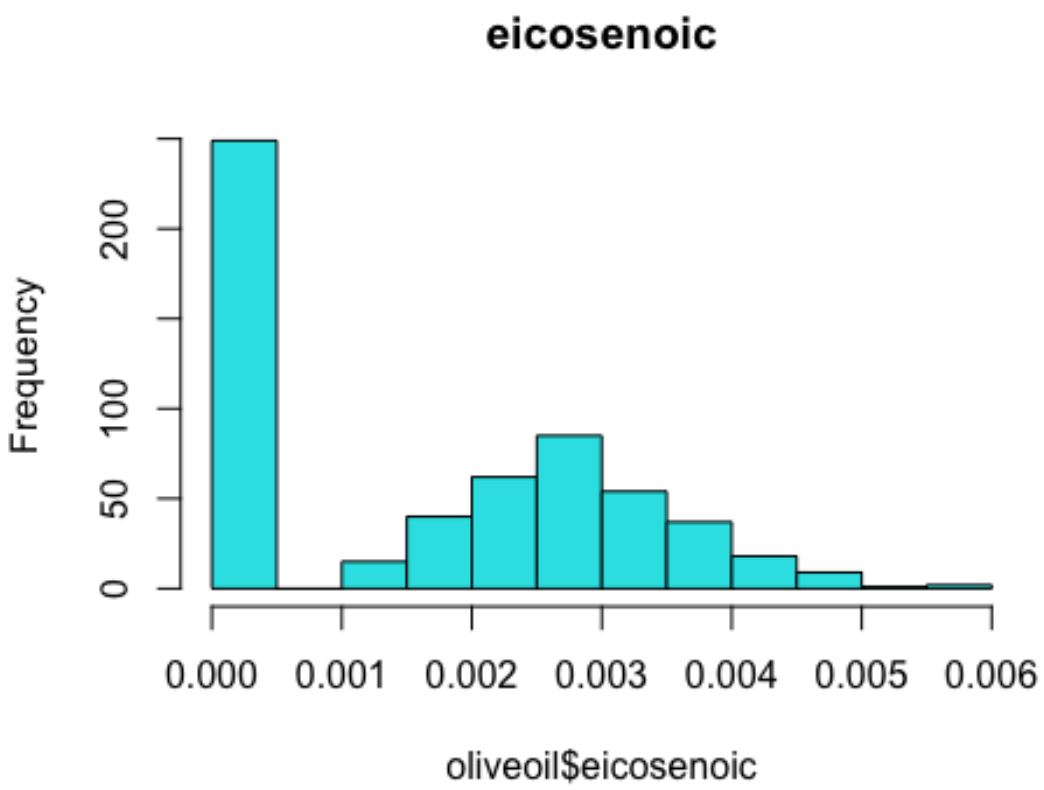
Variabile acido eicosenoic

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.

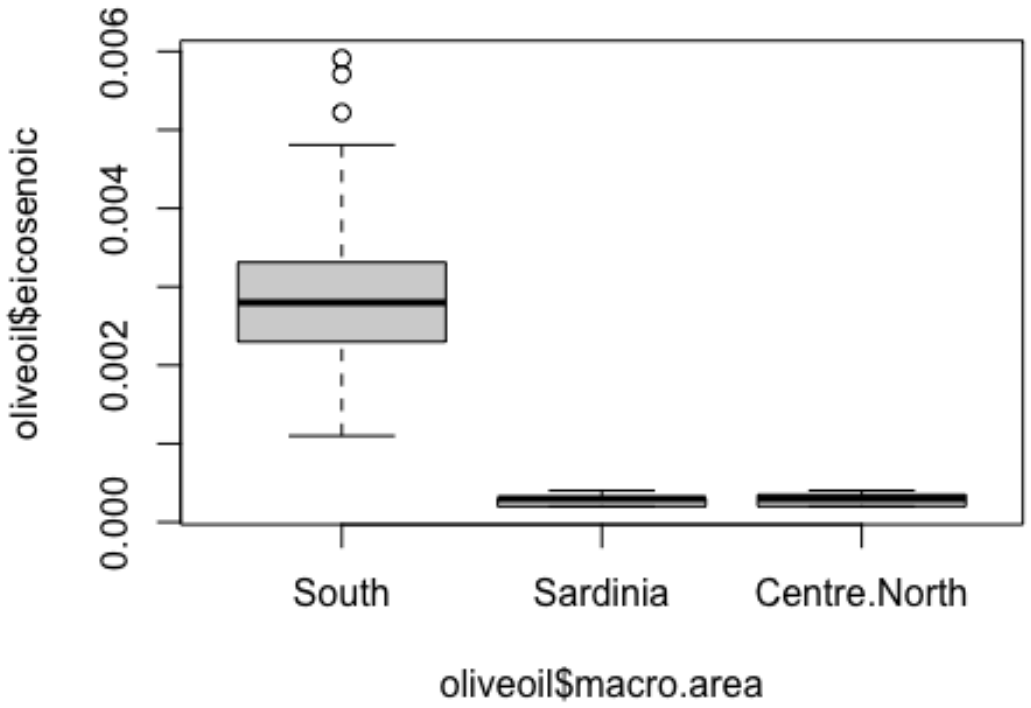
```
display_summary_and_var(oliveoil$eicosenoic)
```

```
##         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
```

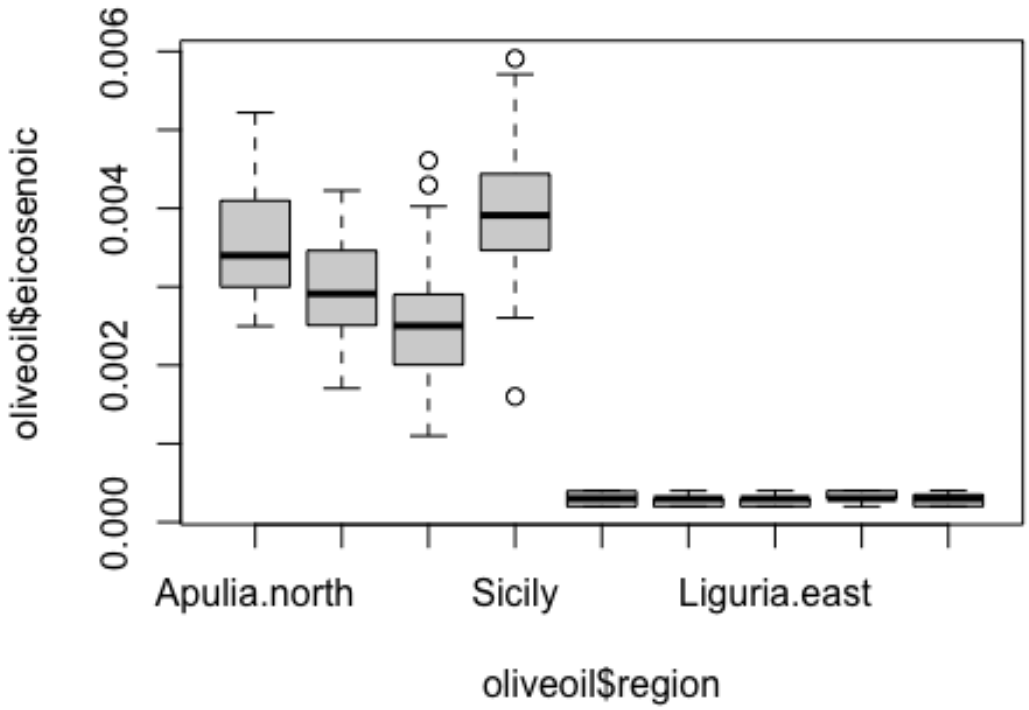
```
hist(oliveoil$eicosenoic, col = 5, main = "eicosenoic")
```




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



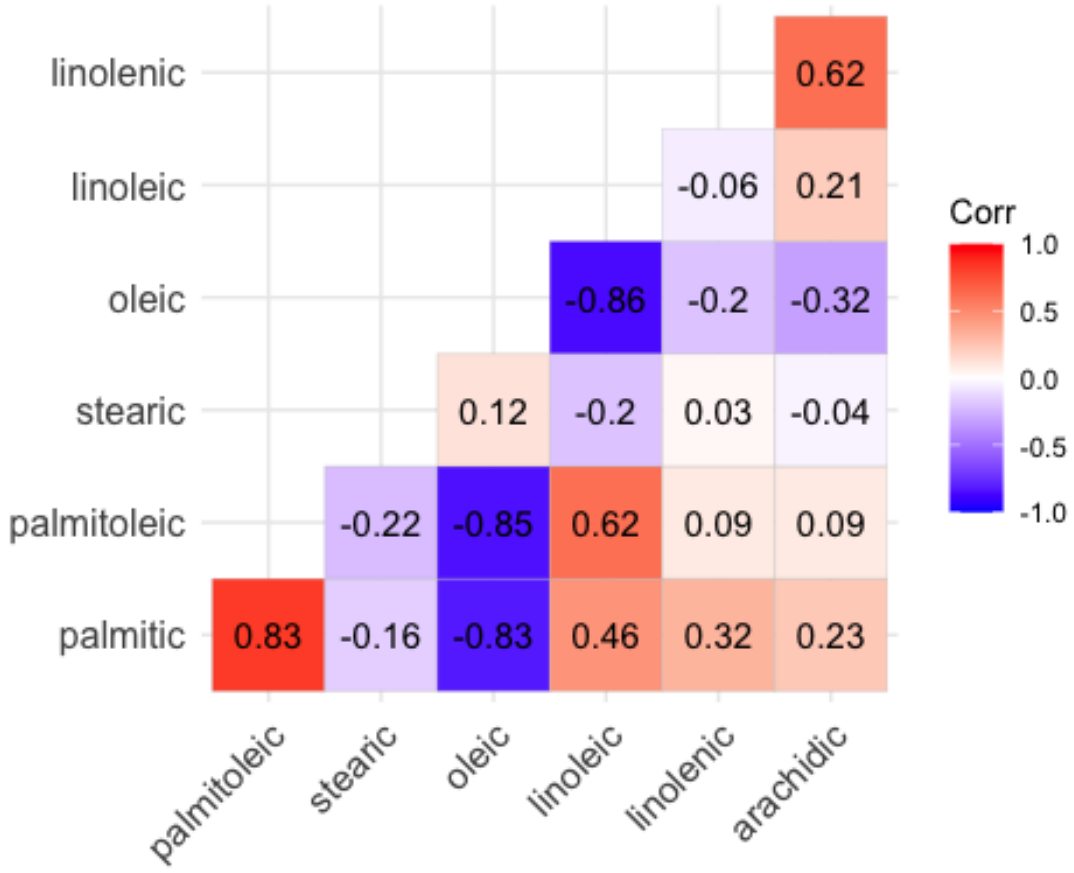

```r
boxplot(oliveoil$eicosenoic~oliveoil$region)
```



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[,3:9]), 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.

Trasformazione dei dati

Si sceglie di togliere dal dataset la colonna acido oleico, che non verrà considerata per la prima parte dell'analisi in cluster

```
oliveoil <- oliveoil[, -6]
```

Essa verrà reintrodotta in seguito per essere usata nella trasformazione ALR

Dataset con le coordinate

Creazione del dataset contenete le coordinate GPS degli oli usato in seguito per i plot delle mappe

Si è deciso di sommare un offset alle coordinate dei punti di ogni regione, in modo da evitare che ogni singola osservazione venisse sovrapposta e per meglio mostrare visivamente la divisione in cluster di ogni regione.

```r
oliveGPS <- oliveoil[,1:2]
oliveGPS$lat <- NA
oliveGPS$long <- NA

region_coords <- list(
  "Apulia.north" = c(lat = 41.4, long = 15.5),
  "Calabria" = c(lat = 39.0, long = 16.5),
  "Apulia.south" = c(lat = 40.0, long = 18.0),
  "Sicily" = c(lat = 37.6, long = 14.1),
  "Sardinia.inland" = c(lat = 40.1, long = 9.0),
  "Sardinia.coast" = c(lat = 39.1, long = 9.7),
  "Liguria.east" = c(lat = 44.3, long = 9.5),
  "Liguria.west" = c(lat = 44.0, long = 8.0),
  "Umbria" = c(lat = 42.9, long = 12.6)
)

for (i in 1:nrow(oliveGPS)) {
  region <- oliveGPS$region[i]
  if (region %in% names(region_coords)) {
    oliveGPS$lat[i] <- region_coords[[region]]["lat"]
    oliveGPS$long[i] <- region_coords[[region]]["long"]
  }
}

for (i in 1:nrow(oliveGPS)) {
  oliveGPS$lat[i] <- oliveGPS$lat[i] + runif(1, min = -0.4, max = 0.4)
  oliveGPS$long[i] <- oliveGPS$long[i] + runif(1, min = -0.4, max = 0.4)
}
```

Cluster Analysis

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

K-Means

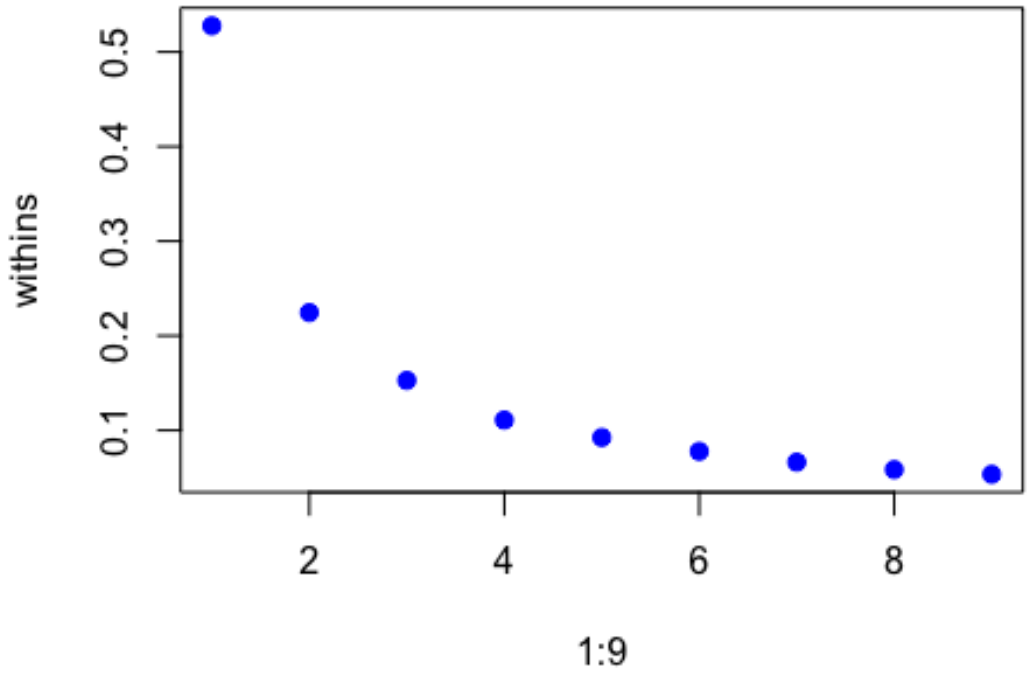
K-Means è un metodo di raggruppamento in cluster che misura le distanze dei punti di un cluster dal suo centro e cerca di minimizzarla. L'algoritmo prende in input il dataset e il numero di cluster voluto e opera nel seguente modo: 1. si scelgono K punti casuali diversi dai punti del dataset che sono 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[,3:9], centers = i, nstart = 15)
  withins[i] <- km.out$tot.withinss
}
plot(1:9, withins, pch=19, col = "blue")
```



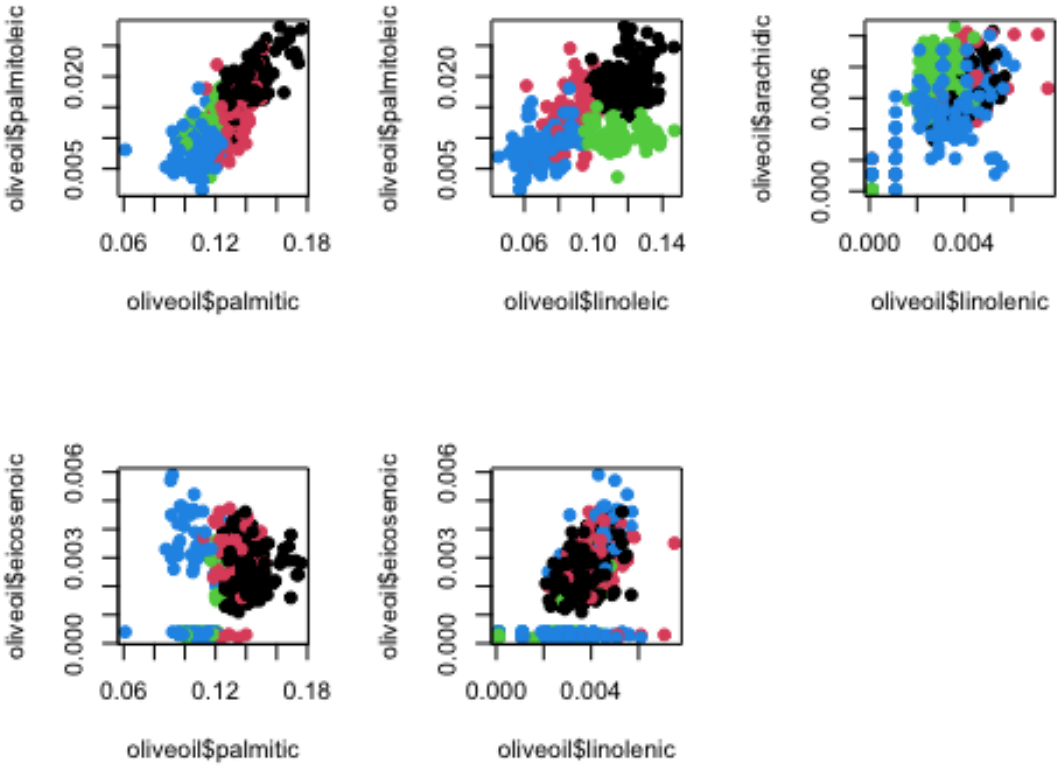

```
# dal grafico si nota che il numero migliore di cluster è 3 o 4

# si procede con l'implementazione di kmeans con K=4
set.seed(17)
km.out <- kmeans(oliveoil[,3:9], centers=4, nstart = 15)
str(km.out)
```

```
## List of 9
##  $ cluster     : int [1:572] 4 4 4 4 4 4 4 4 4 4 ...
##  $ centers     : num [1:4, 1:7] 0.1413 0.1311 0.1107 0.108 0.0186 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
```

```
##  $ totss       : num 0.528
##  $ withinss    : num [1:4] 0.0381 0.0155 0.0248 0.0325
##  $ tot.withinss: num 0.111
##  $ betweenss   : num 0.417
##  $ size        : int [1:4] 186 98 125 163
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

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

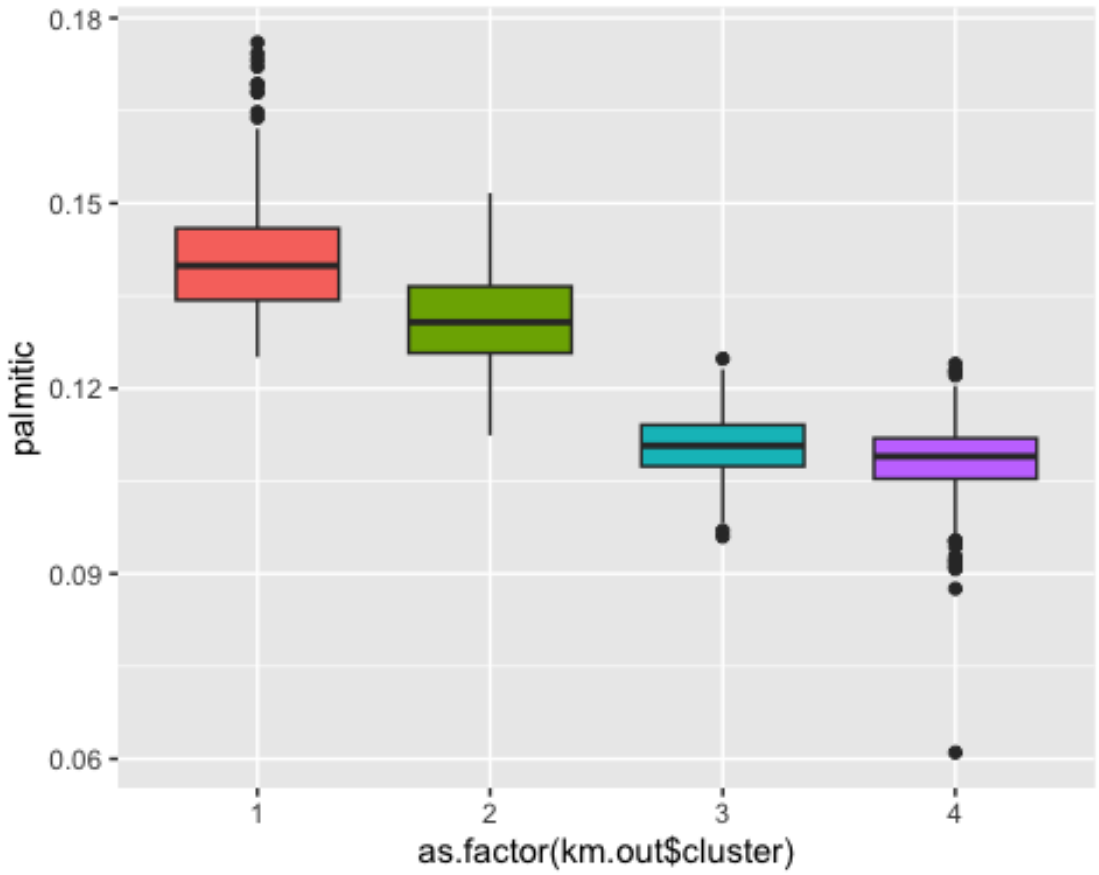
```
par(mfrow=c(2,3))

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

# linoleic palmitoleic
plot(oliveoil$linoleic, oliveoil$palmitoleic, 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))
```



Variabile palmitic nei cluster

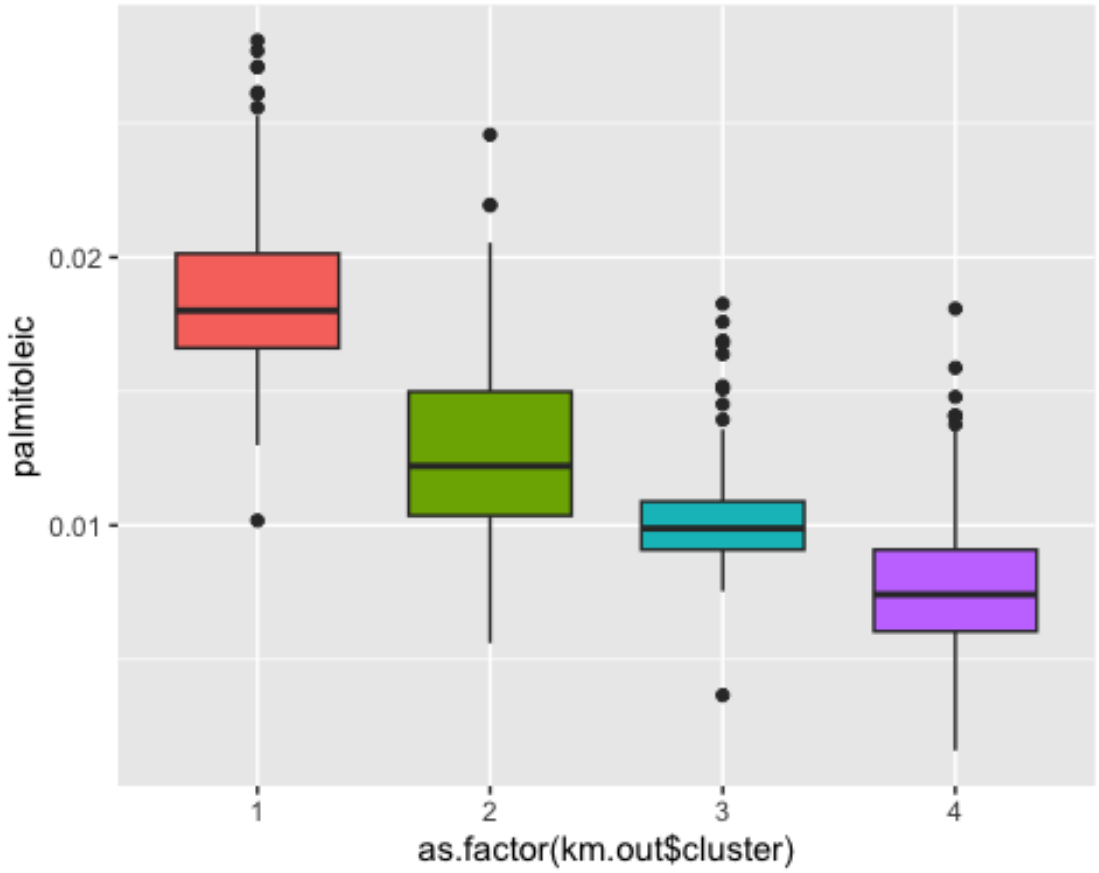
```
ggplot(oliveoil, aes(x = as.factor(km.out$cluster), y = palmitic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```

```
## 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.
```



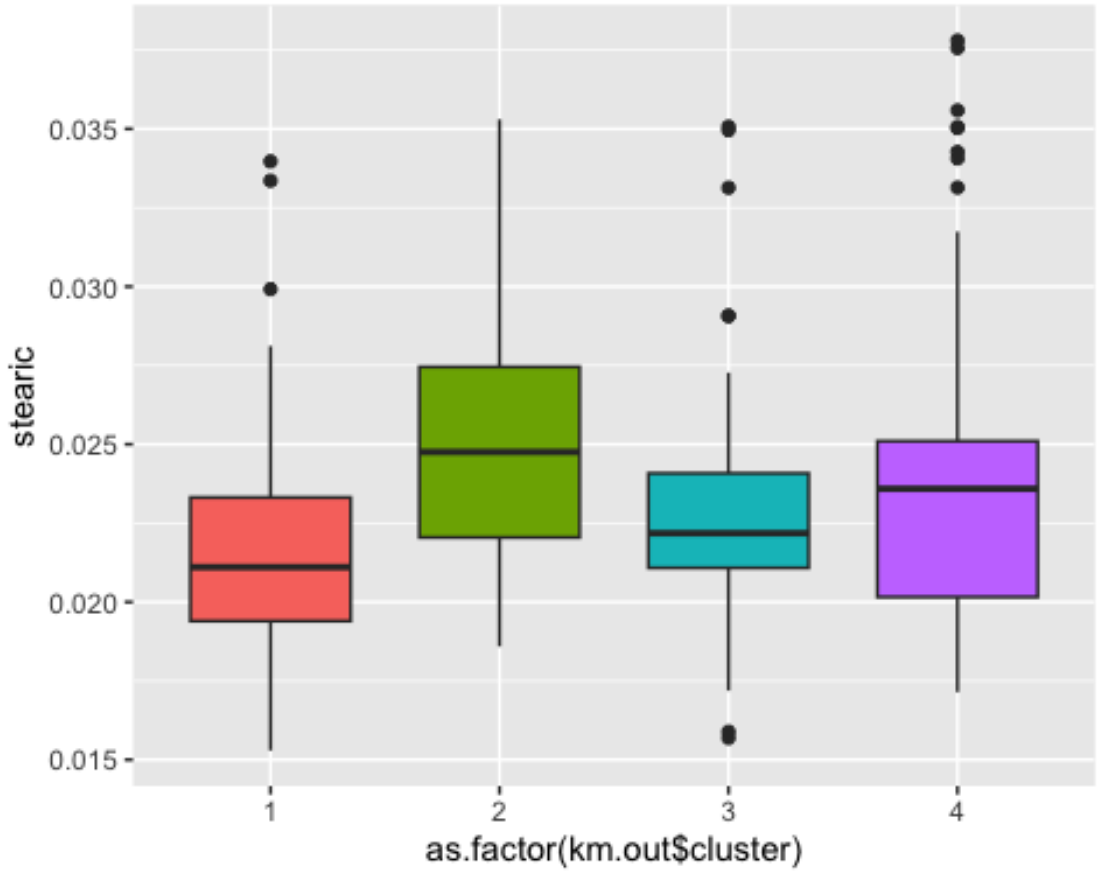
Variabile palmitoleic nei cluster

```
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)
```



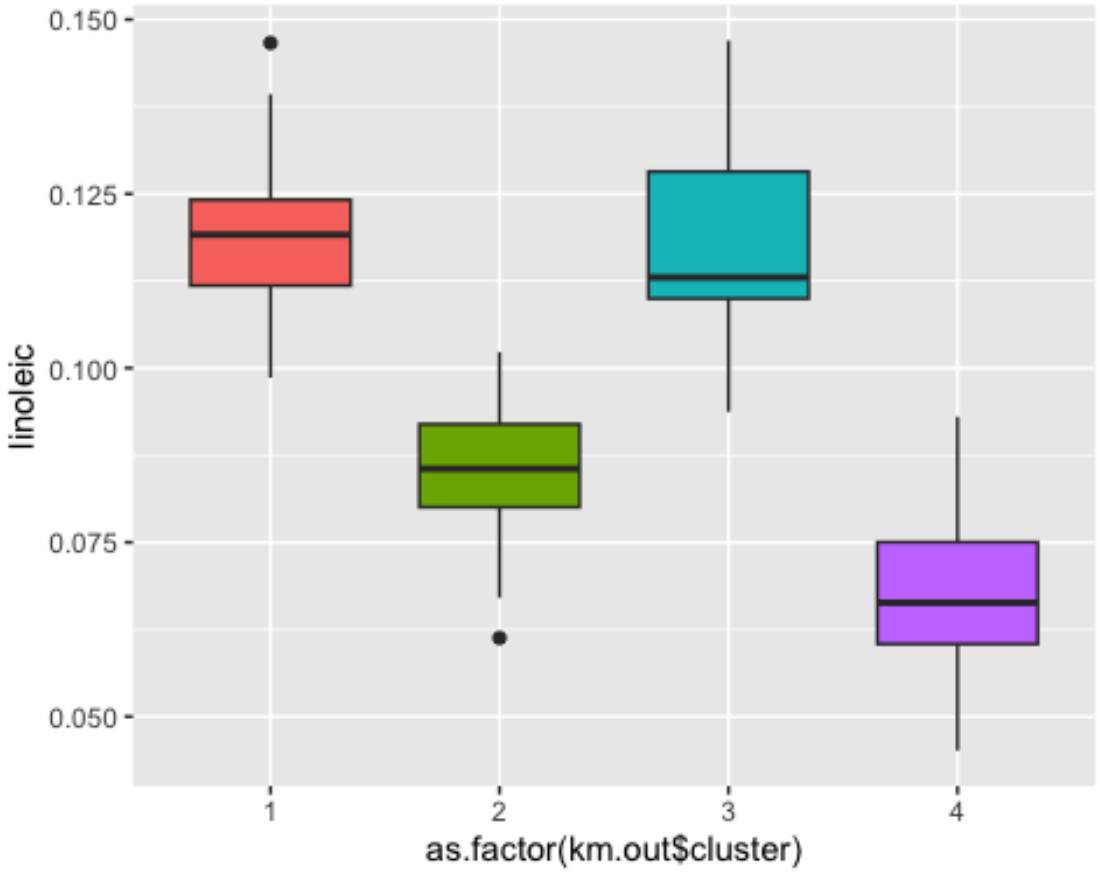
Variabile stearic nei cluster

```
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)
```



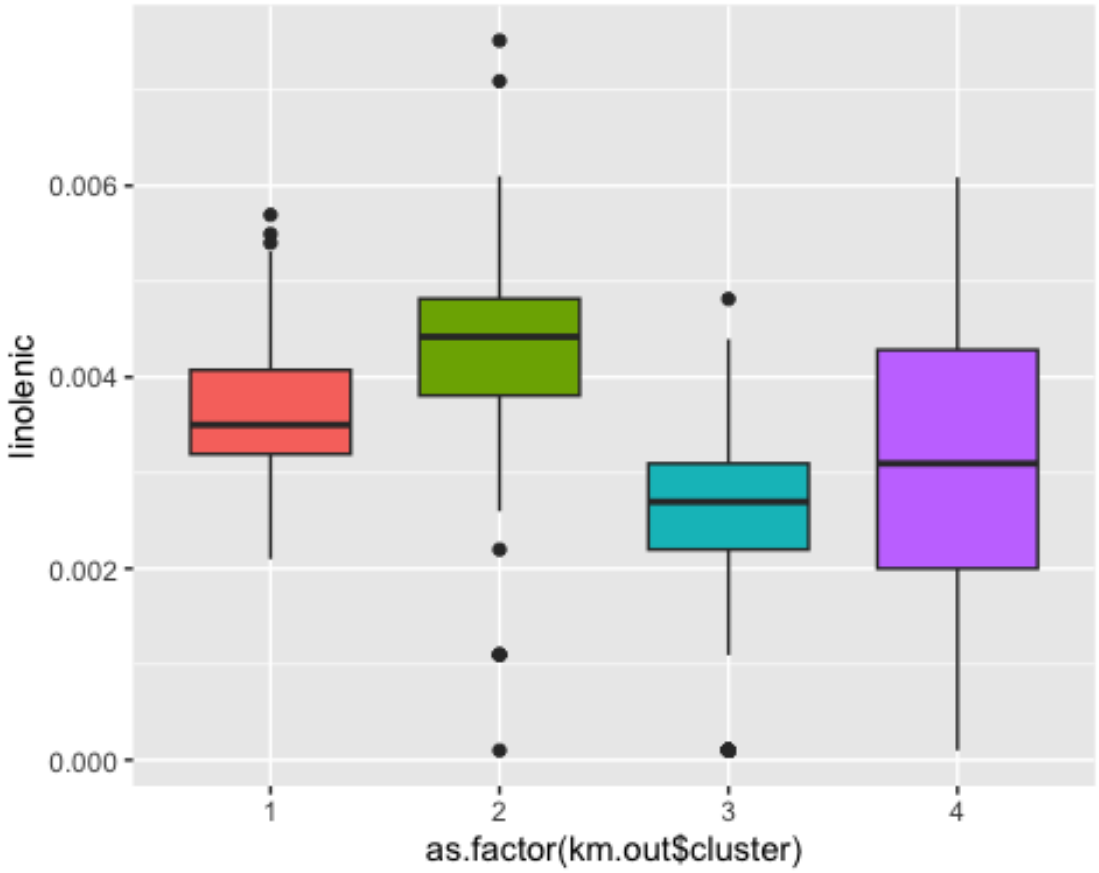
Variabile `linoleic` nei cluster

```
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)
```



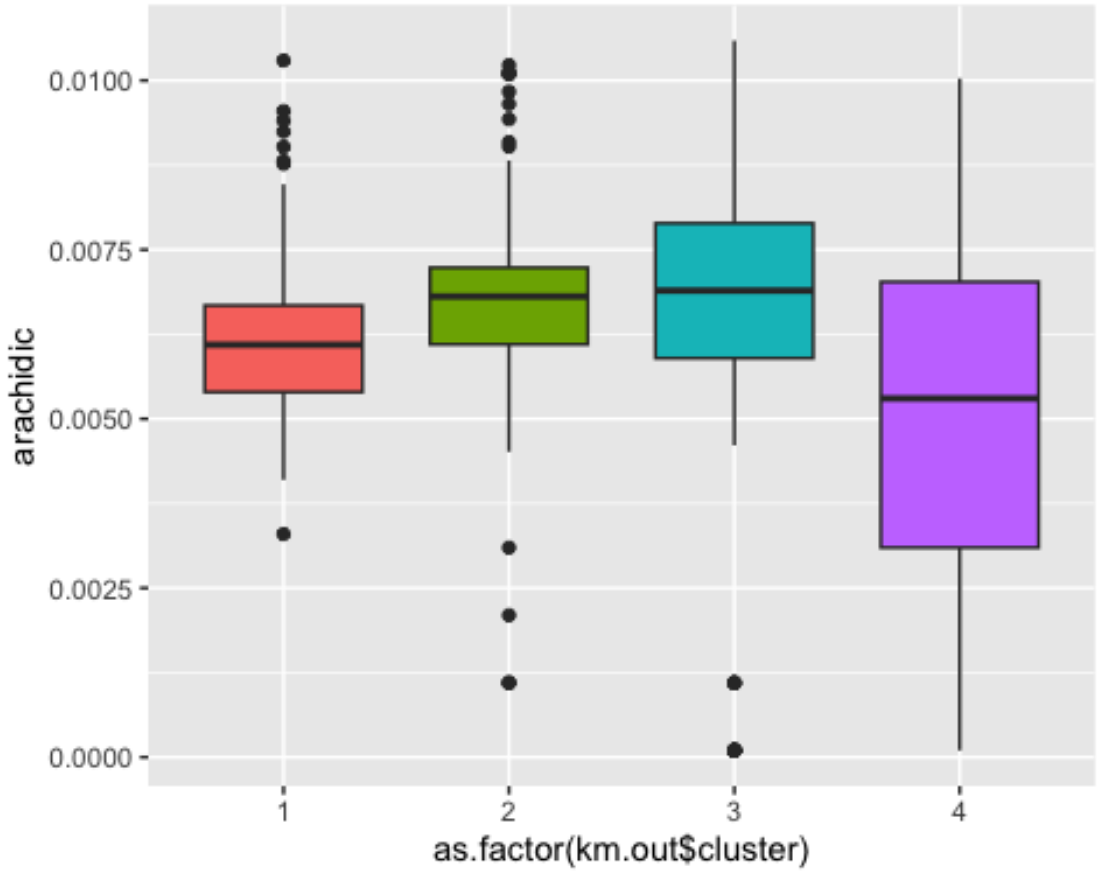
Variabile `linolenic` nei cluster

```
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)
```



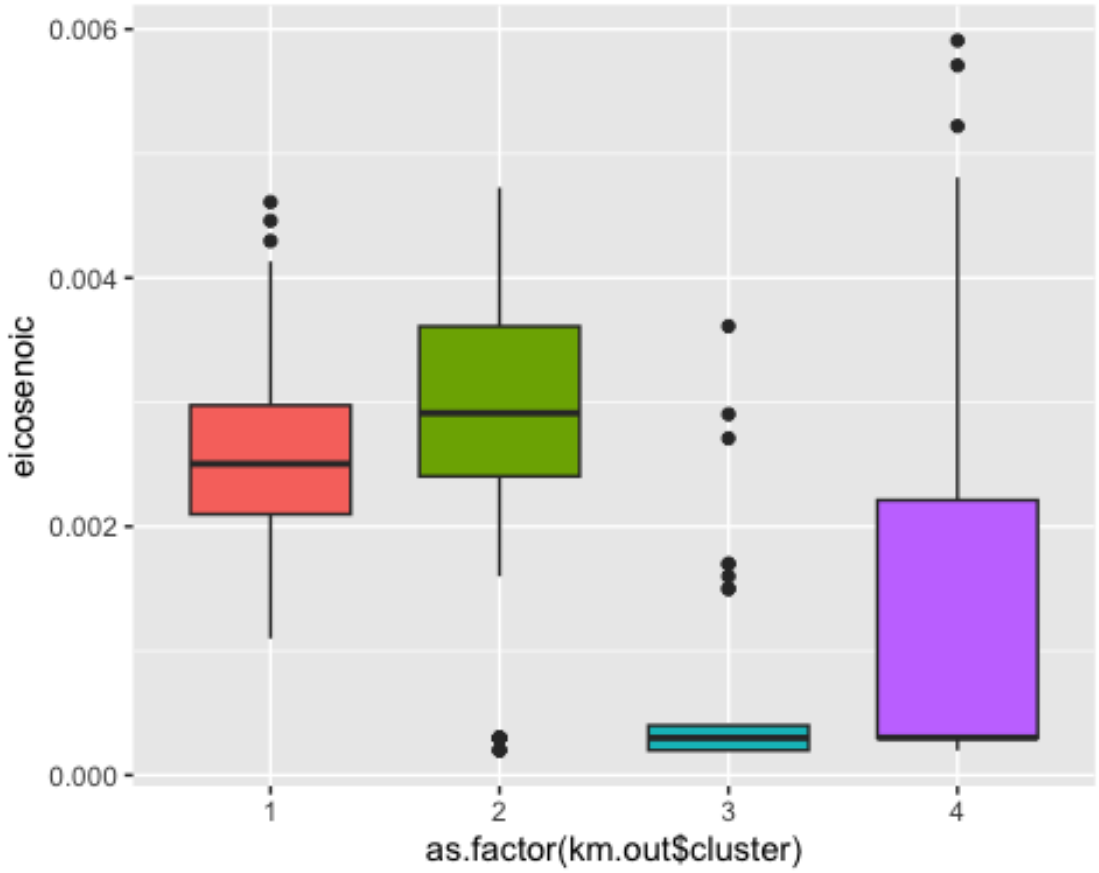
Variabile arachidic nei cluster

```
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)
```



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)
```

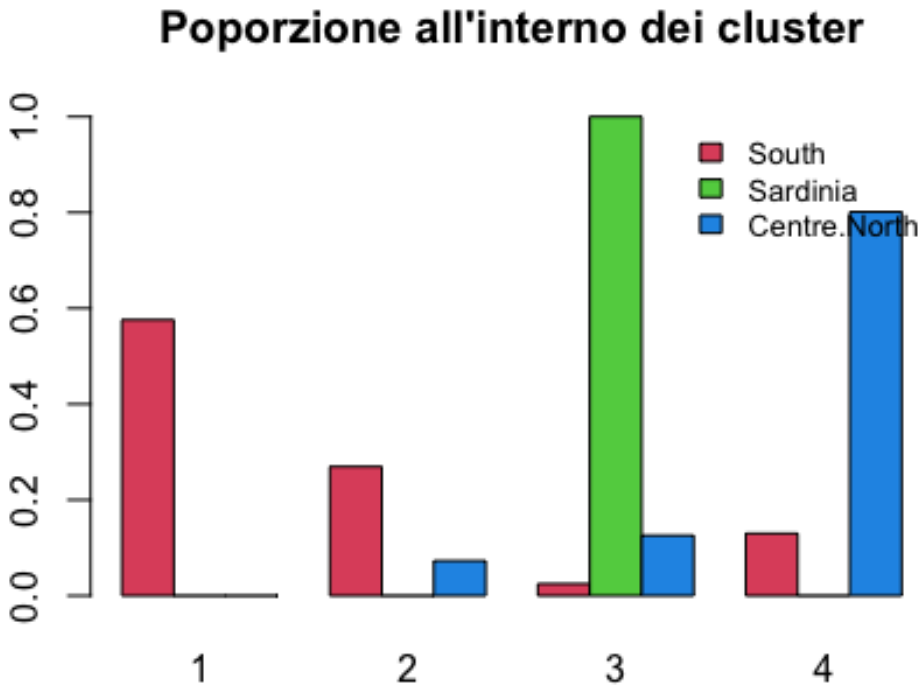


Variabile macro.area nei cluster

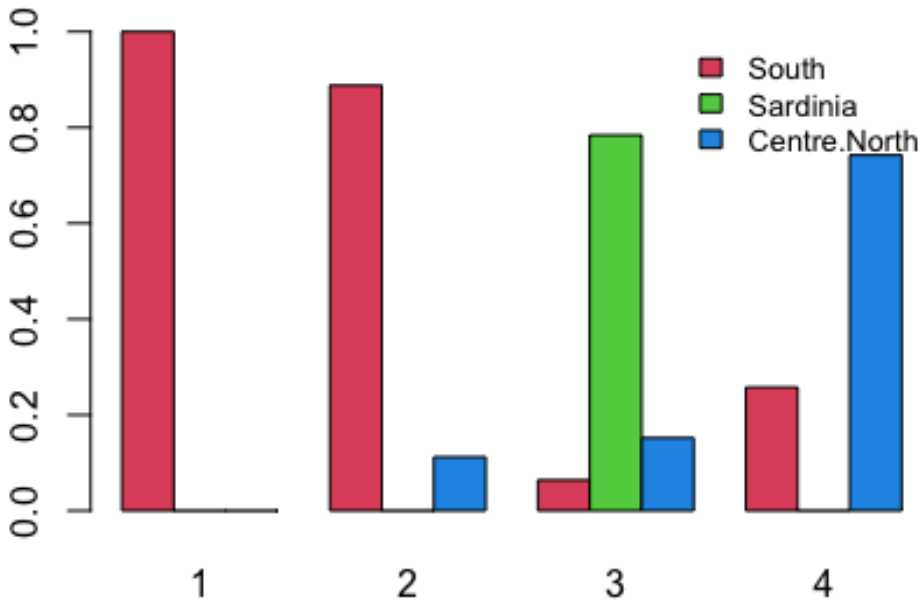
```
prop.table(table(oliveoil$macro.area, km.out$cluster),1)
```

```
##
##                          1          2          3          4
##   South         0.57585139 0.26934985 0.02476780 0.13003096
##   Sardinia      0.00000000 0.00000000 1.00000000 0.00000000
##   Centre.North  0.00000000 0.07284768 0.12582781 0.80132450
```

```
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")
```




```
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")
```

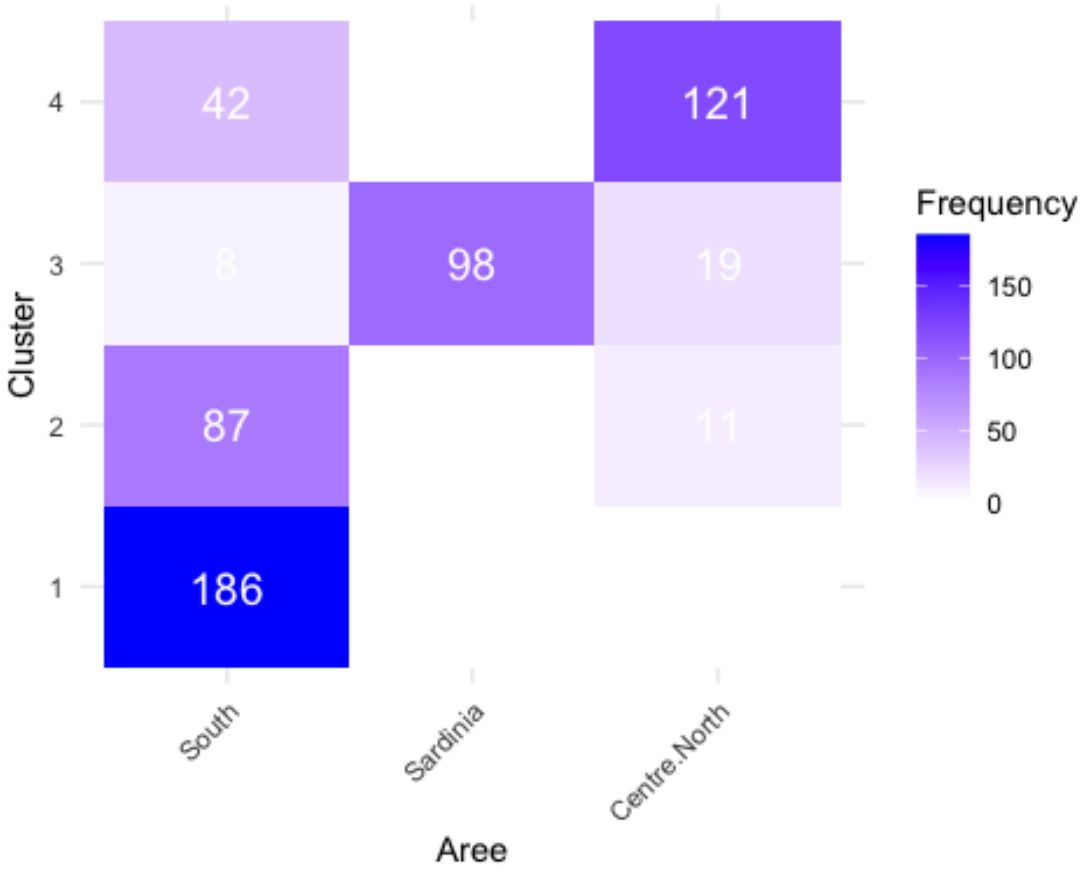


Si nota che nei cluster 1 e 2 sono concentrati gli oli provenienti dal Sud, nel cluster 3 gli oli della Sardegna e nel cluster 4 sono presenti principalmente oli del Centro Nord. La distribuzione in cluster ha quindi individuato le macro aree sufficientemente bene.

Questo si può vedere anche dalla Confusion Matrix ovvero:

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

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

##               Cluster
## Aree             1   2   3   4
##   South        186  87   8  42
##   Sardinia       0   0  98   0
##   Centre.North   0  11  19 121

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



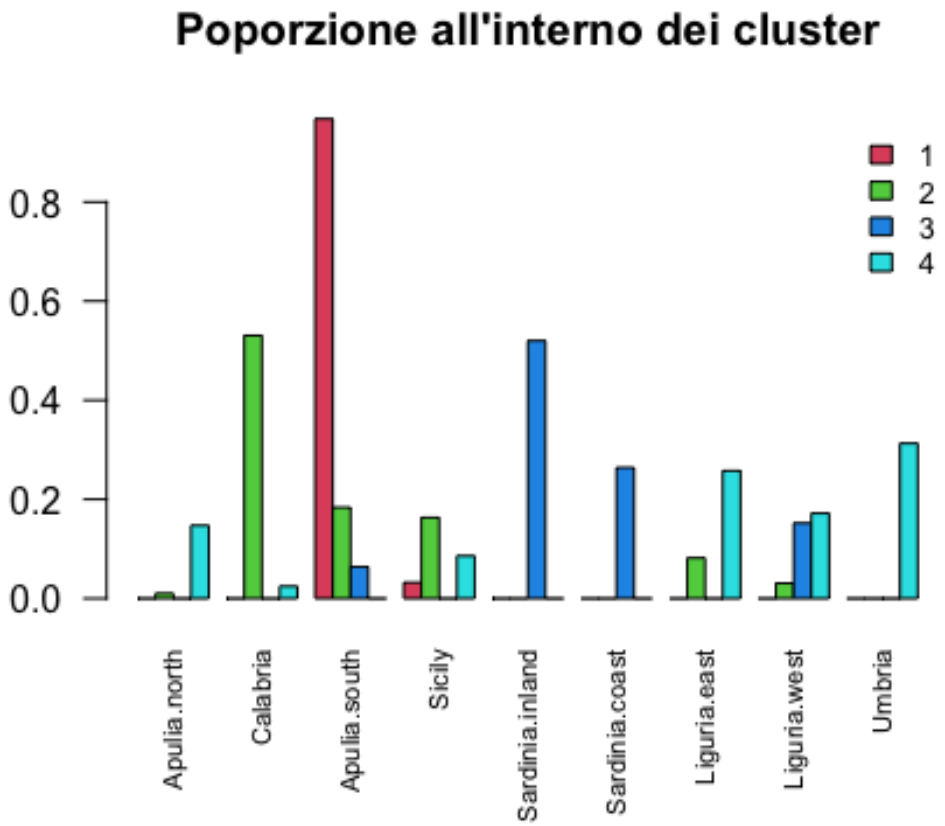
Variabile region nei cluster

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

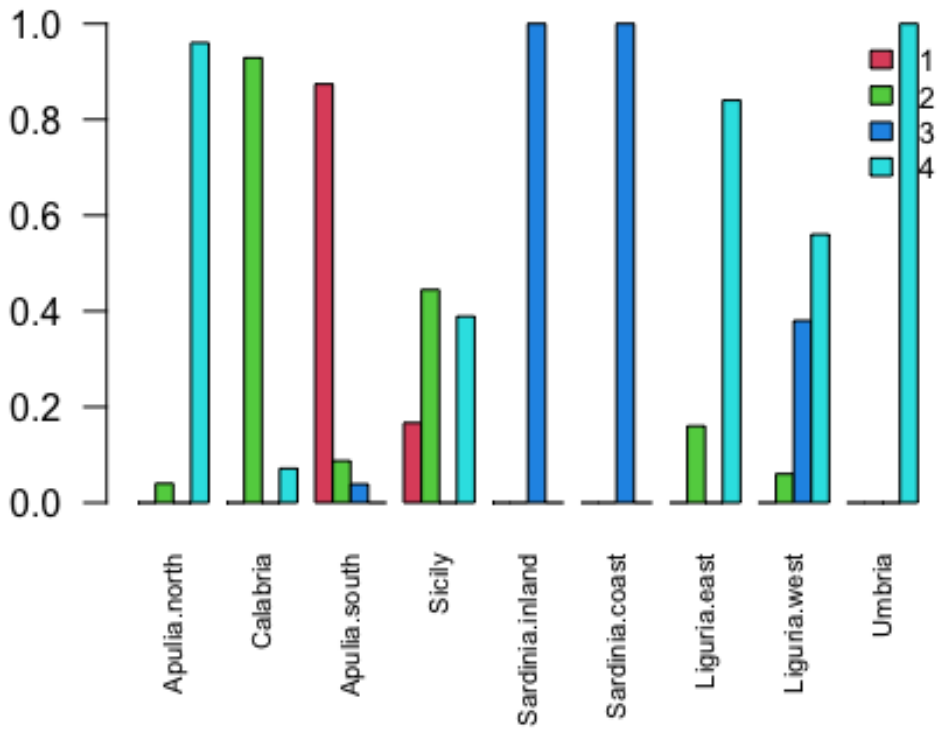
```
##
##     Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   1   0.00000000 0.00000000   0.96774194 0.03225806      0.00000000
##   2   0.01020408 0.53061224   0.18367347 0.16326531      0.00000000
##   3   0.00000000 0.00000000   0.06400000 0.00000000      0.52000000
##   4   0.14723926 0.02453988   0.00000000 0.08588957      0.00000000
##
##     Sardinia.coast Liguria.east Liguria.west     Umbria
##   1     0.00000000   0.00000000   0.00000000 0.00000000
##   2     0.00000000   0.08163265   0.03061224 0.00000000
##   3     0.26400000   0.00000000   0.15200000 0.00000000
##   4     0.00000000   0.25766871   0.17177914 0.31288344
```

```
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")
```



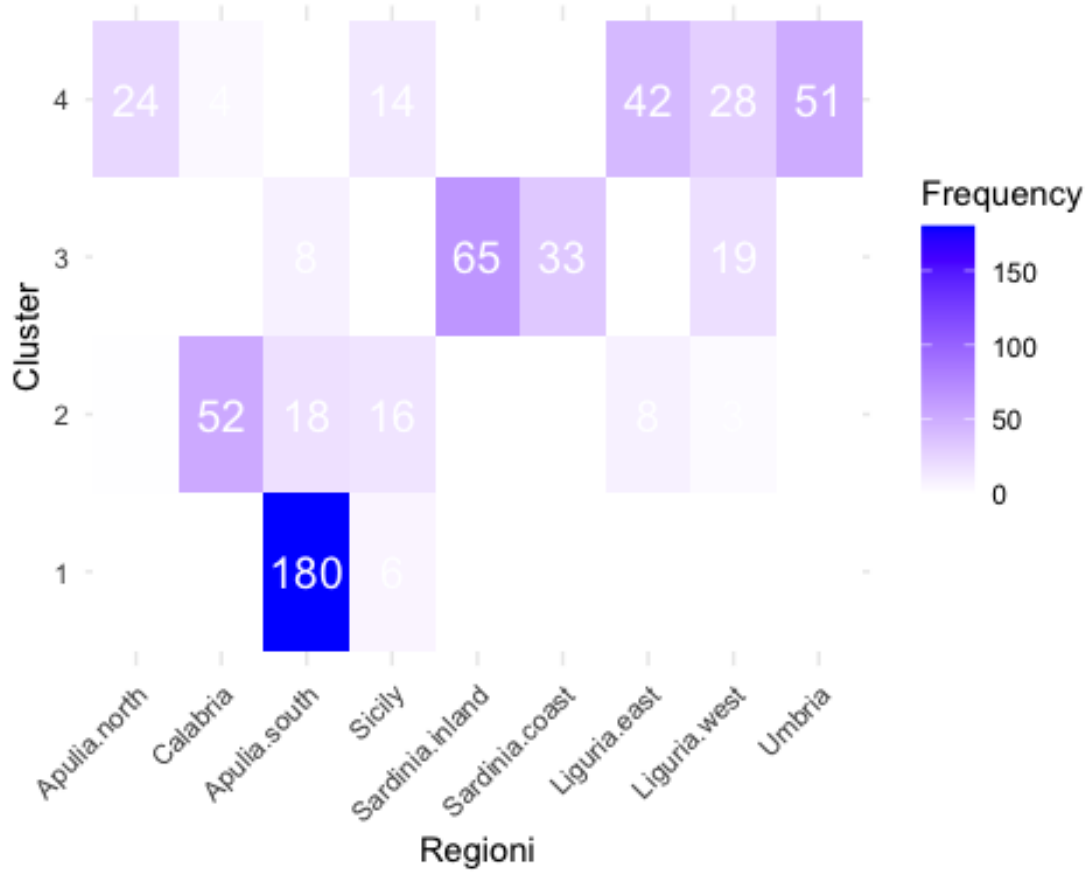
Vediamo subito che, come indicava anche il grafico precedente, la Sardegna corrisponde quasi perfettamente al cluster 3. Con la divisione in regioni notiamo che il sud, il quale era diviso tra cluster 1 e 2 principalmente, era tale perchè il cluster 1 è quasi totalmente composto da oli provenienti dalla Puglia del Sud. Le restanti regioni del sud sono invece divise principalmente nel cluster 2 e anche nel cluster 4. Anche guardando le regioni la divisione in cluster è riuscita bene, con le uniche 2 regioni che producono oli che tendono a finire in cluster diversi essere la Sicilia e la zona della Liguria dell'est.

Di seguito la confusion matrix:

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

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

##                 Cluster
## Regioni            1   2   3   4
##   Apulia.north     0   1   0  24
##   Calabria         0  52   0   4
##   Apulia.south   180  18   8   0
##   Sicily           6  16   0  14
##   Sardinia.inland  0   0  65   0
##   Sardinia.coast   0   0  33   0
```

```
##   Liguria.east      0   8   0  42
##   Liguria.west      0   3  19  28
##   Umbria            0   0   0  51
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Regioni, y =
Cluster, 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))
```



Adjusted Rand Index - K Means

```
ari_km <- adj.rand.index(oliveoil$macro.area, km.out$cluster)
ari_km
```

```
## [1] 0.430368
```

Mappa dei cluster sulla cartina Italiana

```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=km.out$cluster+1, pch=19, cex=0.3)
```

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:

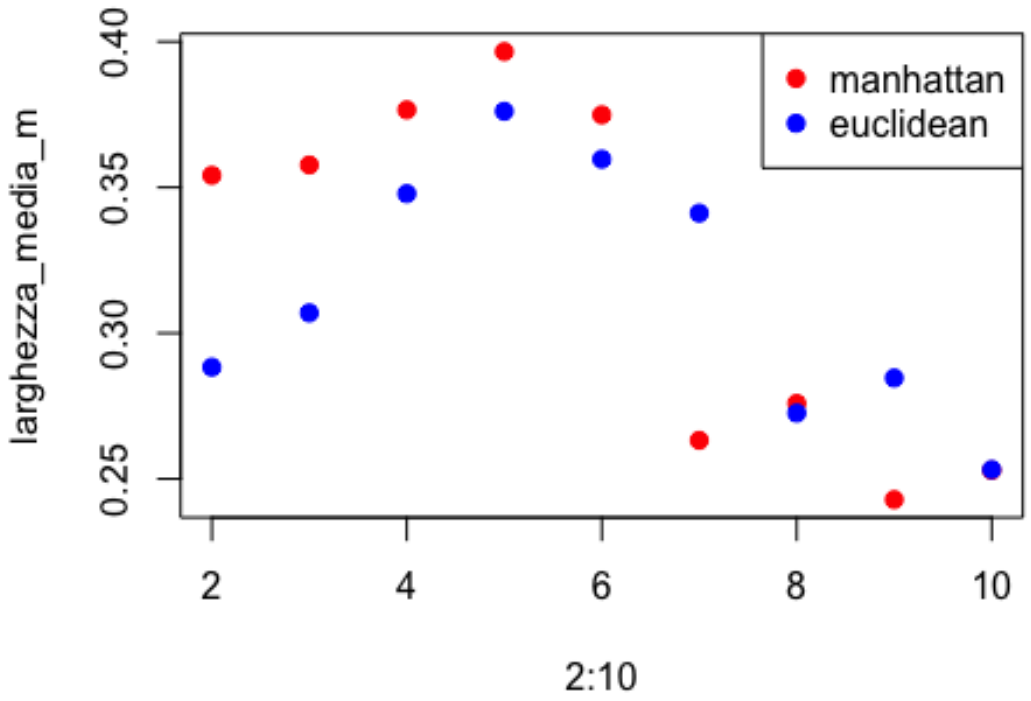
$$d_E(p,q) = (\sum_{i=1}^{n}(p_i - q_i)^2)^{\frac{1}{2}}$$

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[,3:9], 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(oliveoil[,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)
```



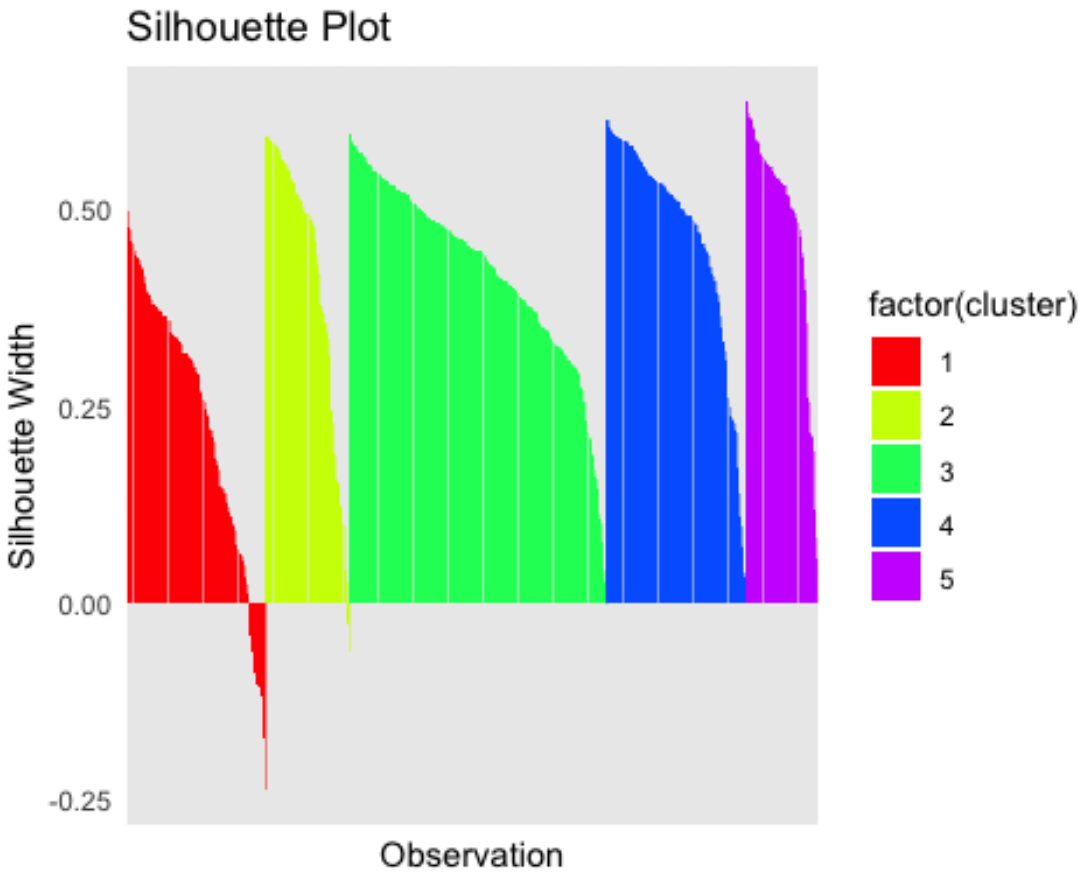
Il numero di cluster migliore sembra essere 5. La distanza manhattan è migliore della distanza euclidea a parità di numero di cluster, come si vede dal grafico.

```
set.seed(17)
pam.out<-pam(oliveoil[,3:9], 5, metric="manhattan", stand=TRUE, nstart = 10)
str(pam.out)
```

```
## List of 10
##  $ medoids   : num [1:5, 1:7] 0.127 0.109 0.142 0.106 0.103 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##  $ id.med    : int [1:5] 51 438 239 343 555
##  $ clustering: int [1:572] 1 1 1 1 1 1 1 1 1 2 ...
##  $ objective : Named num [1:2] 5.31 3.65
##   ..- 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] 115 69 213 116 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" "39" ...
##   .. .. ..$ : chr [1:3] "cluster" "neighbor" "sil_width"
##   ..$ clus.avg.widths: num [1:5] 0.23 0.414 0.421 0.464 0.481
##   ..$ avg.width      : num 0.397
##  $ diss      : NULL
##  $ call      : language pam(x = oliveoil[, 3:9], k = 5, metric =
"manhattan", nstart = 10, stand = TRUE)
##  $ data      : num [1:572, 1:7] -1.089 -0.999 -2.217 -1.84 -1.254 ...
##   ..- attr(*, "scaled:center")= num [1:7] 0.12337 0.01272 0.023 0.09821
0.00329 ...
##   ..- attr(*, "scaled:scale")= num [1:7] 0.014523 0.004525 0.002813
0.021405 0.000984 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##  - attr(*, "class")= chr [1:2] "pam" "partition"
```

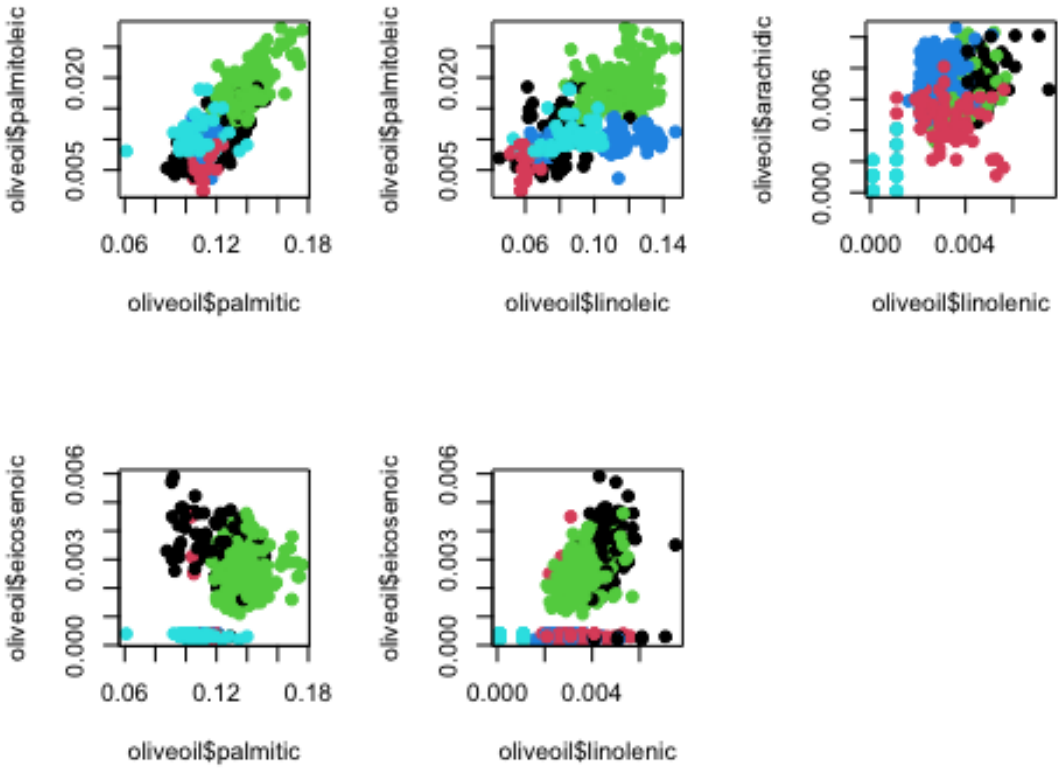
```r
# GRAFICO
sil_df <- as.data.frame(silhouette(pam.out)[, 1:3])
colnames(sil_df) <- c("cluster", "neighbor", "sil_width")
sil_df$obs <- 1:nrow(sil_df)
sil_df <- sil_df[order(sil_df$cluster, -sil_df$sil_width),]
sil_df$obs_ordered <- factor(sil_df$obs, levels = sil_df$obs)

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.

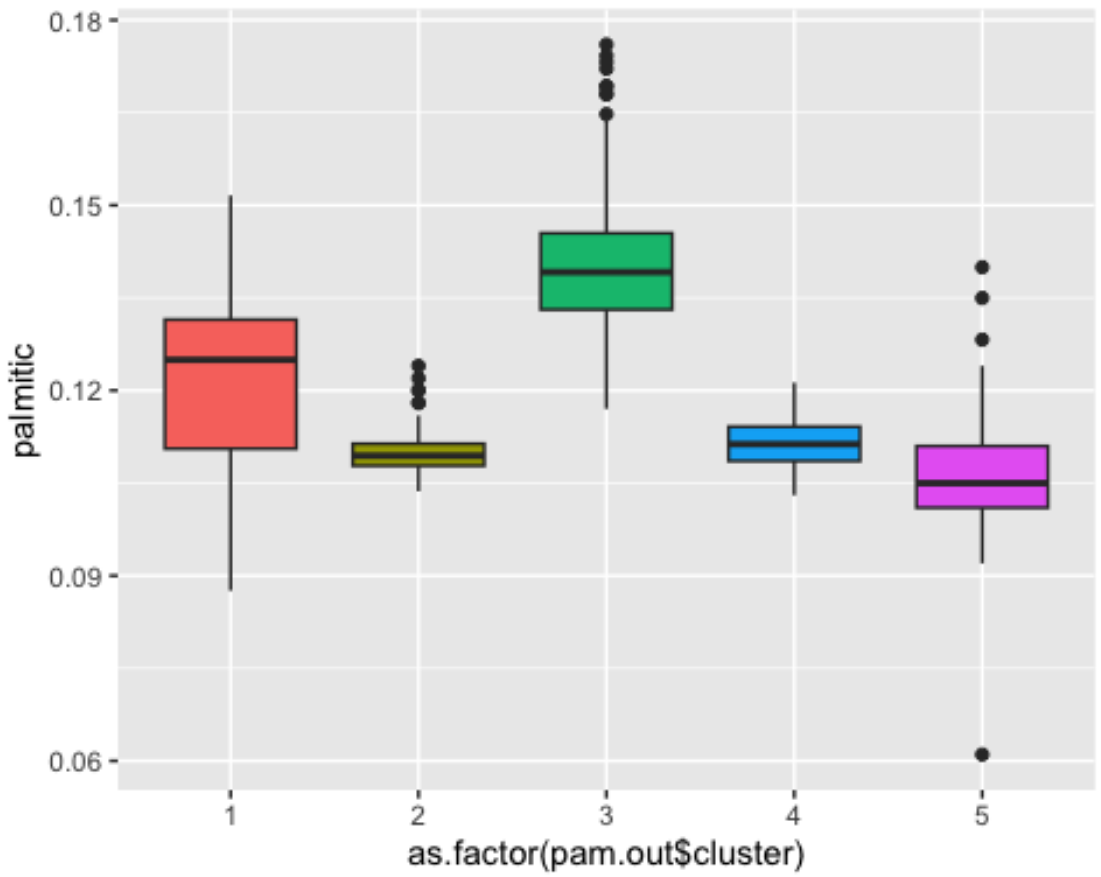
```
par(mfrow=c(2,3))

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

# linoleic palmitoleic
plot(oliveoil$linoleic, oliveoil$palmitoleic, 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))
```



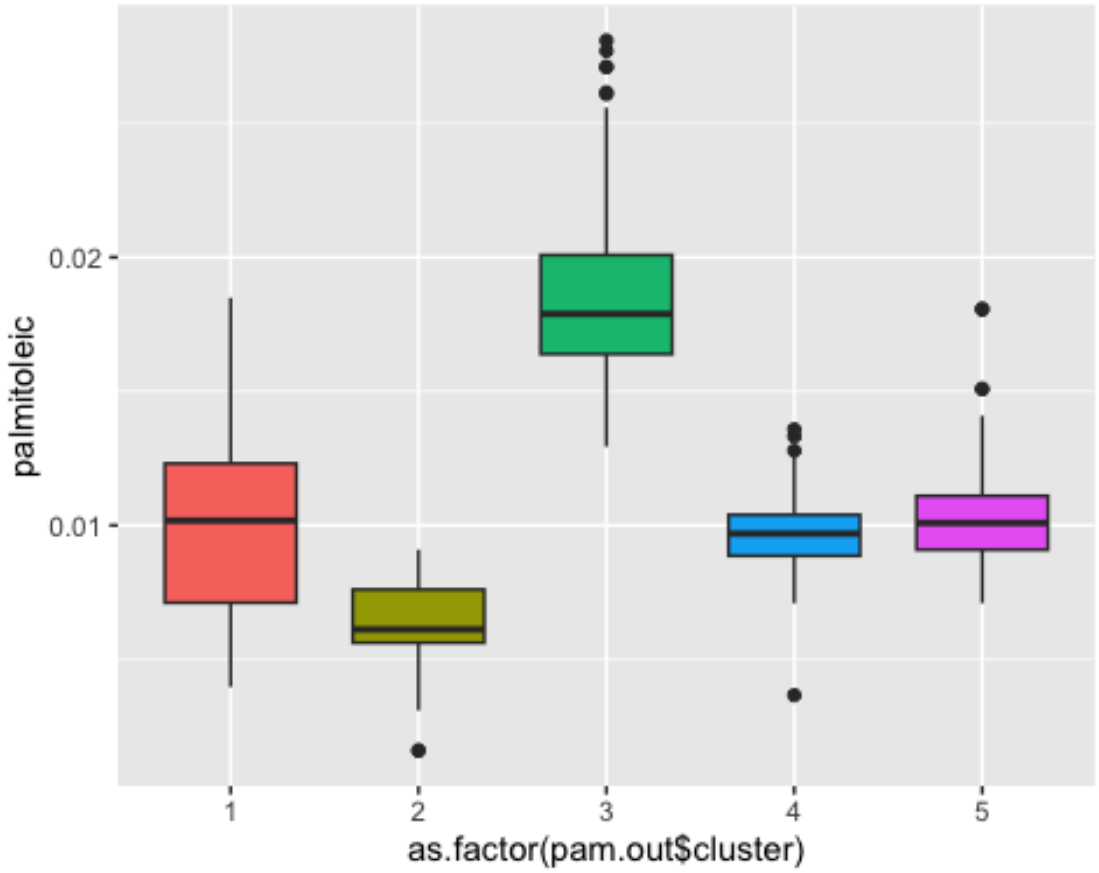
Variabile palmitic

```
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)
```



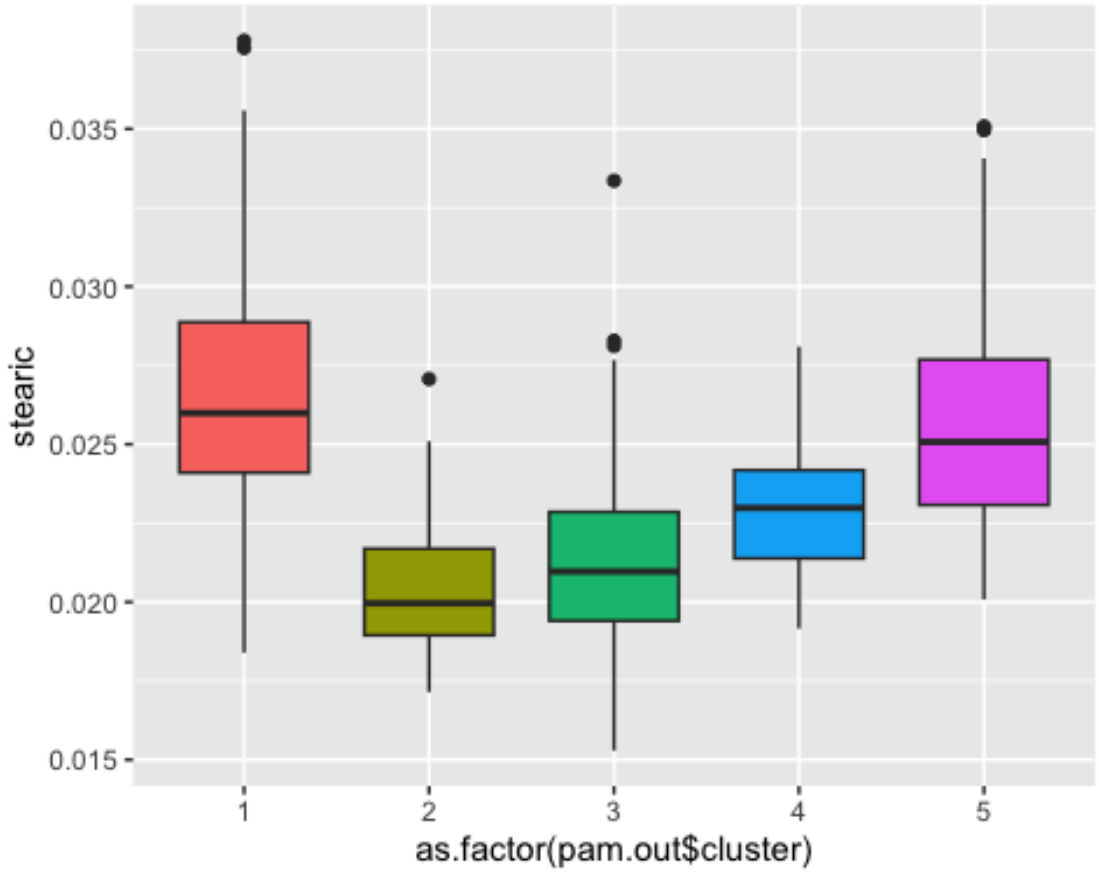
Variabile palmitoleic

```
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)
```



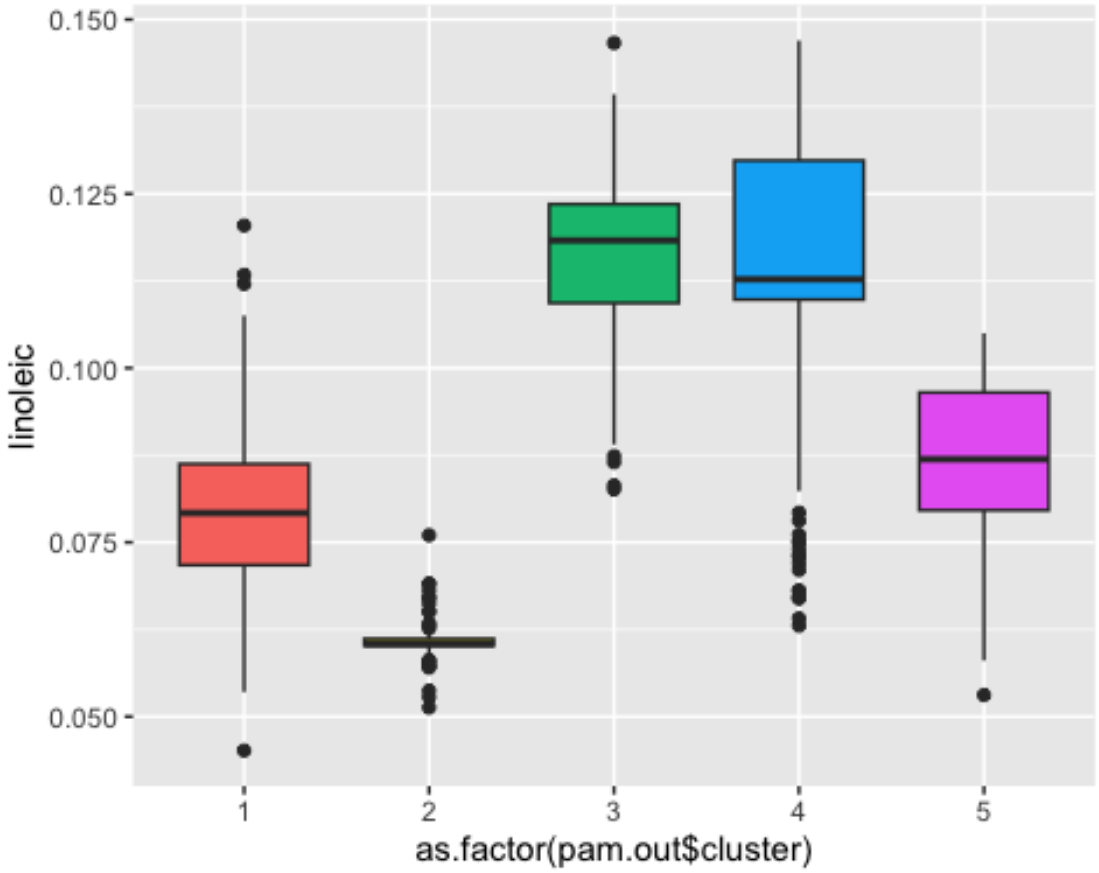
Variabile stearic

```
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)
```



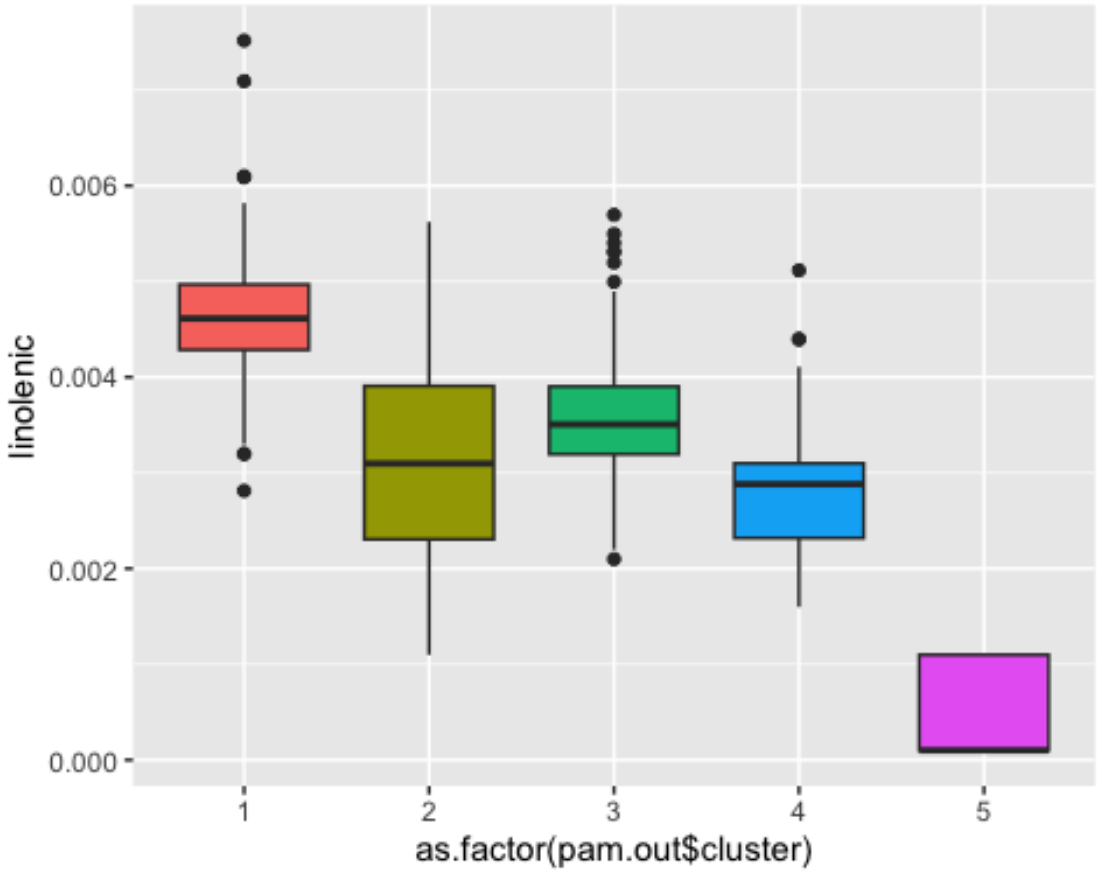
Variabile linoleic

```
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)
```



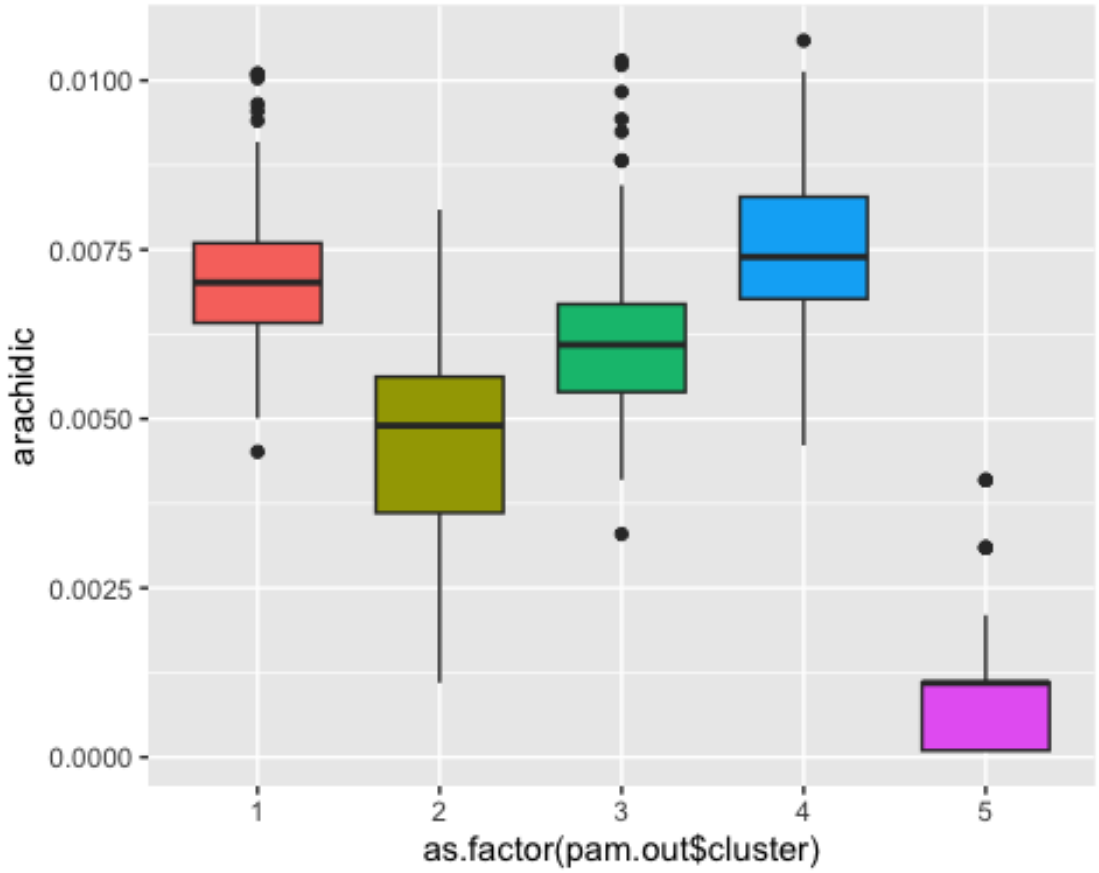
Variabile Linolenic

```
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)
```



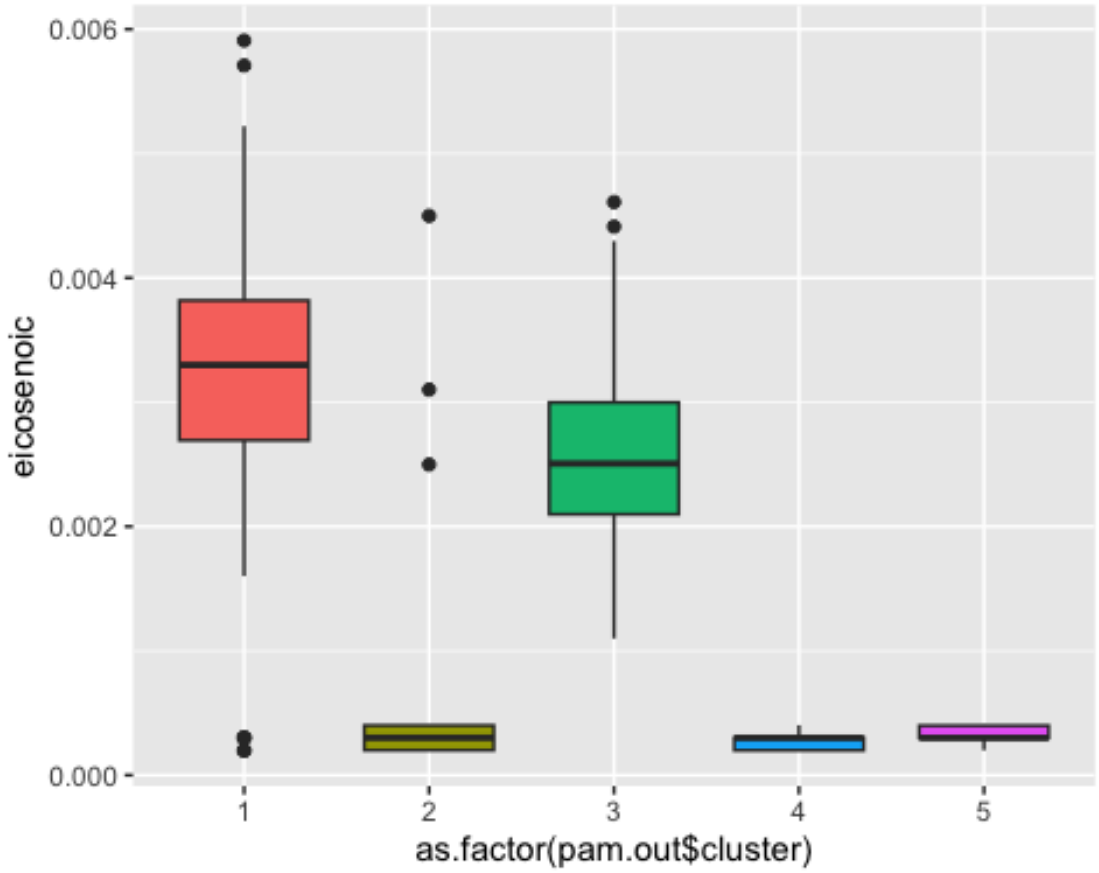
Variabile arachidic

```
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)
```



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)
```

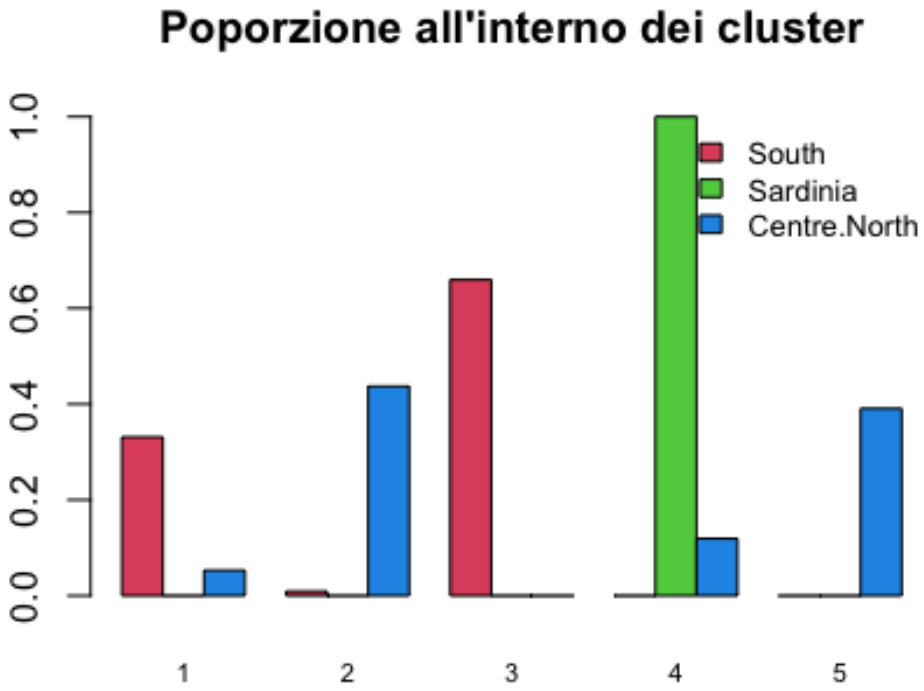


Variabile macro.area

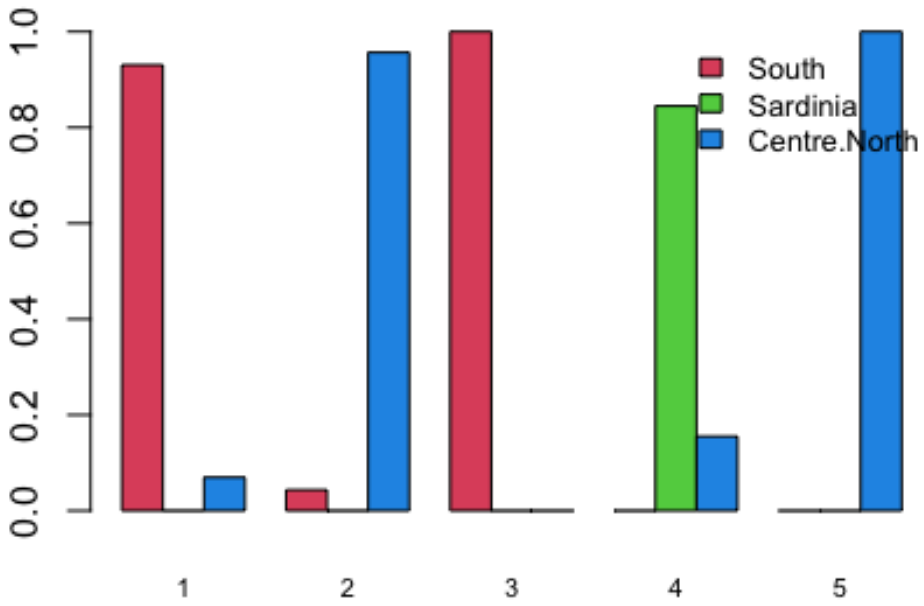
```
prop.table(table(oliveoil$macro.area, pam.out$cluster),1)
```

```
##
##                          1           2           3           4           5
##   South        0.331269350 0.009287926 0.659442724 0.000000000 0.000000000
##   Sardinia     0.000000000 0.000000000 0.000000000 1.000000000 0.000000000
##   Centre.North 0.052980132 0.437086093 0.000000000 0.119205298 0.390728477
```

```
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")
```

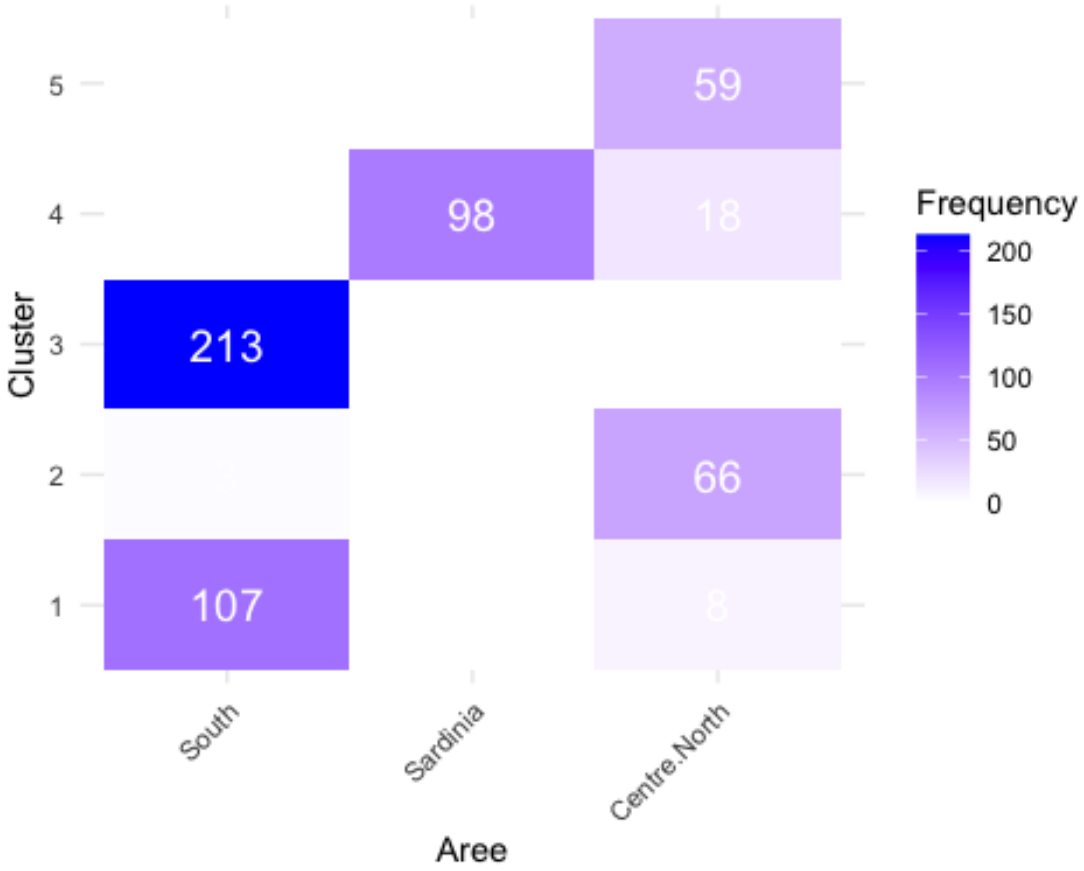


Anche in questo caso la distribuzione in cluster mi va a individuare bene quelle che sono le macro aree, con il Sud che viene diviso tra cluster 1 e 3; il Centro Nord tra il 2 e 5 principalmente; e infine la Sardegna nel 4.

Confusion Matrix:

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

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

##               Cluster
## Aree             1   2   3   4   5
##   South        107   3 213   0   0
##   Sardinia       0   0   0  98   0
##   Centre.North   8  66   0  18  59

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



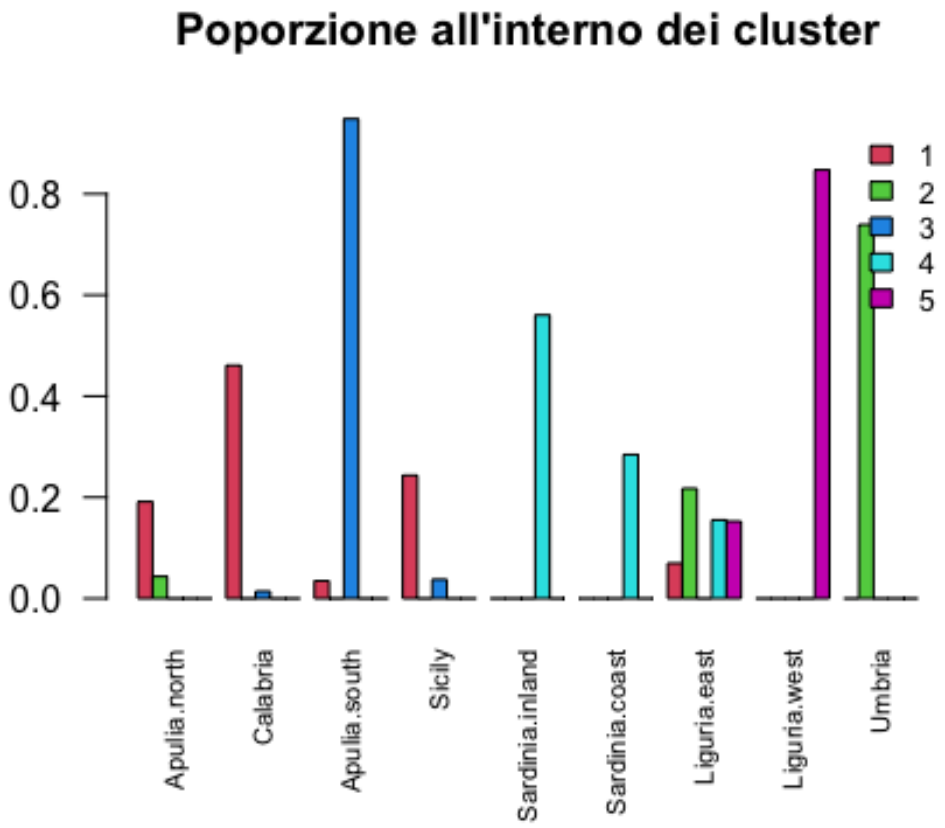
Variabile region

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

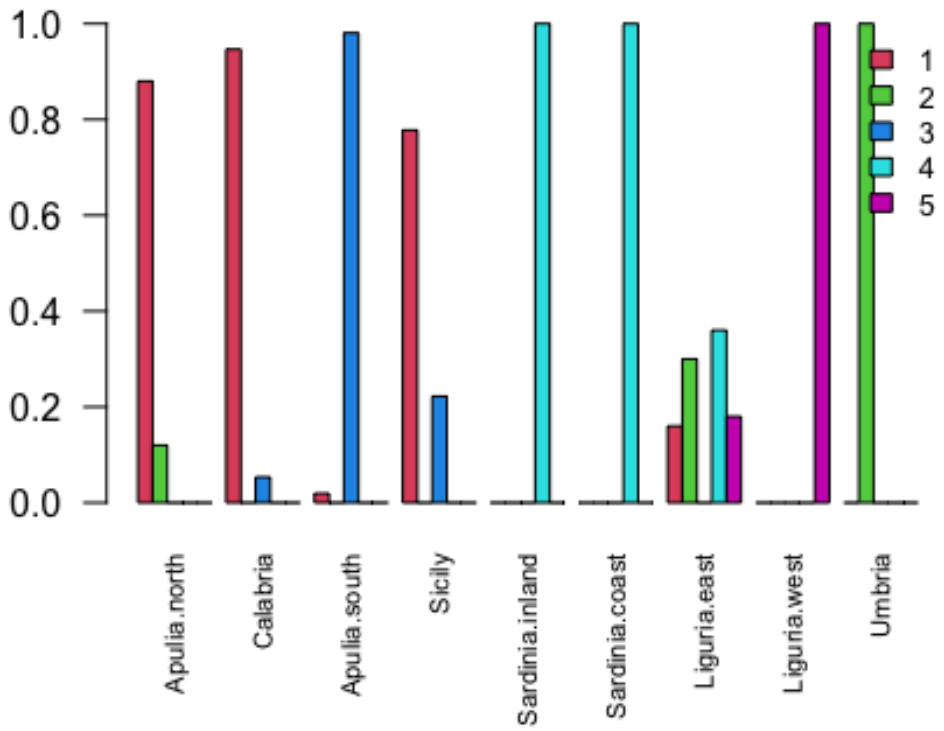
```
##
##     Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   1   0.19130435 0.46086957   0.03478261 0.24347826      0.00000000
##   2   0.04347826 0.00000000   0.00000000 0.00000000      0.00000000
##   3   0.00000000 0.01408451   0.94835681 0.03755869      0.00000000
##   4   0.00000000 0.00000000   0.00000000 0.00000000      0.56034483
##   5   0.00000000 0.00000000   0.00000000 0.00000000      0.00000000
##
##     Sardinia.coast Liguria.east Liguria.west     Umbria
##   1     0.00000000   0.06956522   0.00000000 0.00000000
##   2     0.00000000   0.21739130   0.00000000 0.73913043
##   3     0.00000000   0.00000000   0.00000000 0.00000000
##   4     0.28448276   0.15517241   0.00000000 0.00000000
##   5     0.00000000   0.15254237   0.84745763 0.00000000
```

```
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")
```



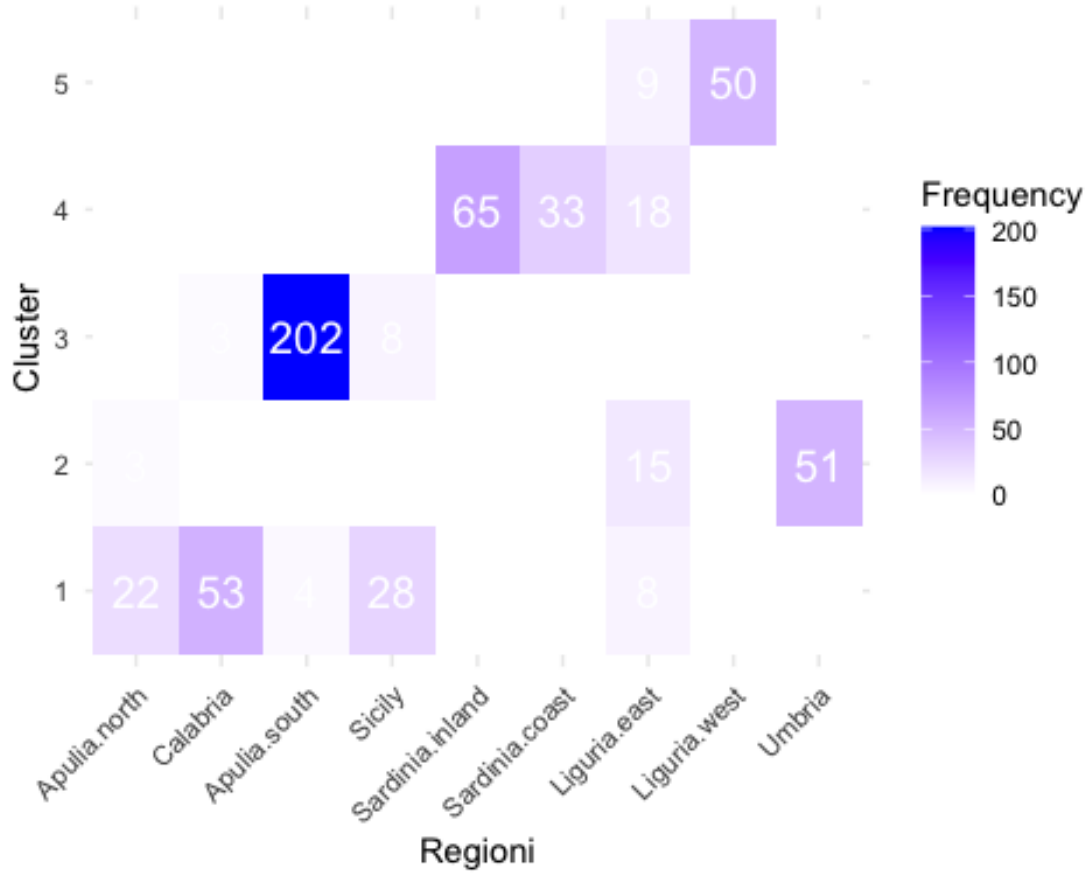
Nella divisione in regioni notiamo ancora che la Puglia del Sud rimane isolata rispetto alle altre regioni del sud Italia, il che indica una qualità di oli prodotti che, anche se comunque simili, ha differenze rispetto alle altre del Sud Italia. La Liguria dell'Est rimane la regione con i più oli che appartengono a cluster diversi. Gli oli prodotti in questa zona sembrano quindi essere molto diversi tra loro anche se provenienti dalla stessa regione.

Confusion Matrix:

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

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

##                  Cluster
## Regioni             1   2   3   4   5
##   Apulia.north     22   3   0   0   0
##   Calabria         53   0   3   0   0
##   Apulia.south      4   0 202   0   0
##   Sicily           28   0   8   0   0
##   Sardinia.inland   0   0   0  65   0
##   Sardinia.coast    0   0   0  33   0
##   Liguria.east      8  15   0  18   9
```

```
##   Liguria.west      0   0   0   0  50
##   Umbria            0  51   0   0   0
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Regioni, y =
Cluster, 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))
```



Adjusted Rand Index - Pam

```
ari_pam <- adj.rand.index(oliveoil$macro.area, pam.out$cluster)
ari_pam
```

```
## [1] 0.5467989
```

Mappa dei cluster sulla cartina Italiana

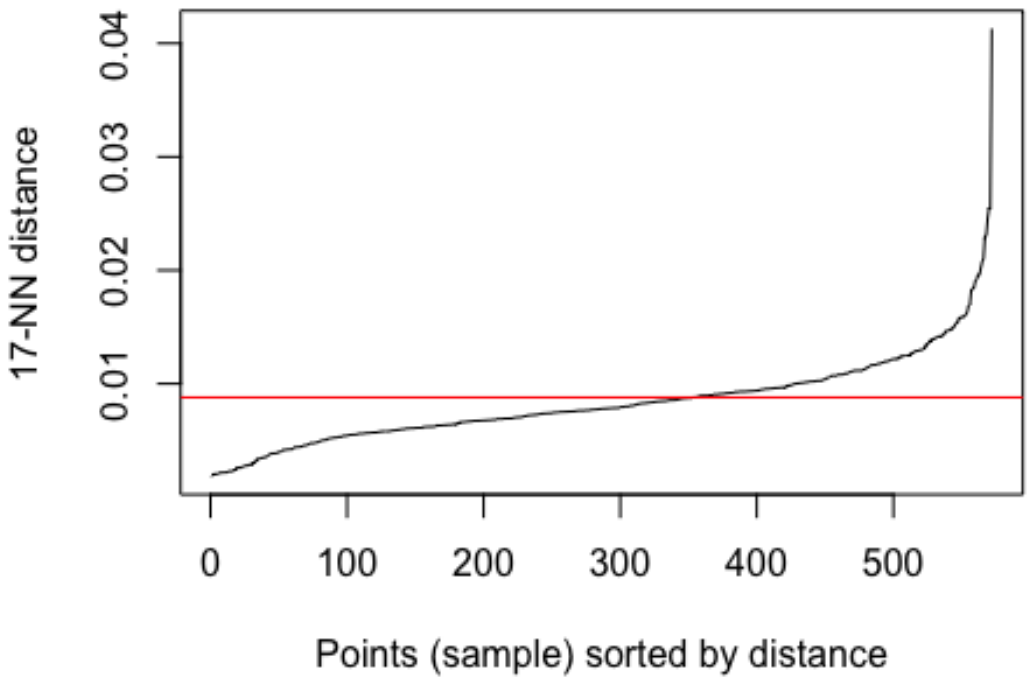
```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=pam.out$cluster+1, pch=19, cex=0.3)
```

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[,3:9], k = 17)
abline(h=0.0088, col = "red")
```



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

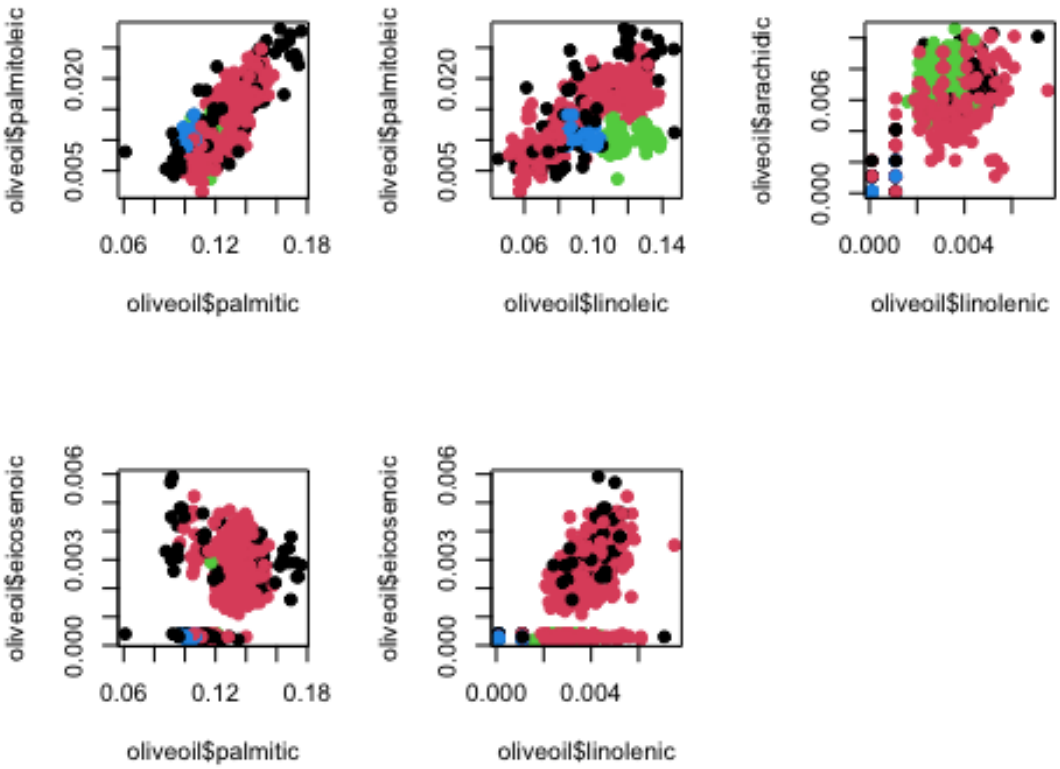
```
set.seed(17)

db.out <- dbscan(oliveoil[,3:9], eps = 0.0088, minPts = 16)
str(db.out)

## List of 5
##  $ cluster     : int [1:572] 1 1 0 1 1 0 0 1 1 1 ...
##  $ eps         : num 0.0088
##  $ minPts      : num 16
##  $ dist        : chr "euclidean"
##  $ borderPoints: logi TRUE
##  - attr(*, "class")= chr [1:2] "dbscan_fast" "dbscan"

db.out

## DBSCAN clustering for 572 objects.
## Parameters: eps = 0.0088, minPts = 16
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 3 cluster(s) and 76 noise points.
##
##   0   1   2   3
##  76 374  98  24
```

```
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

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

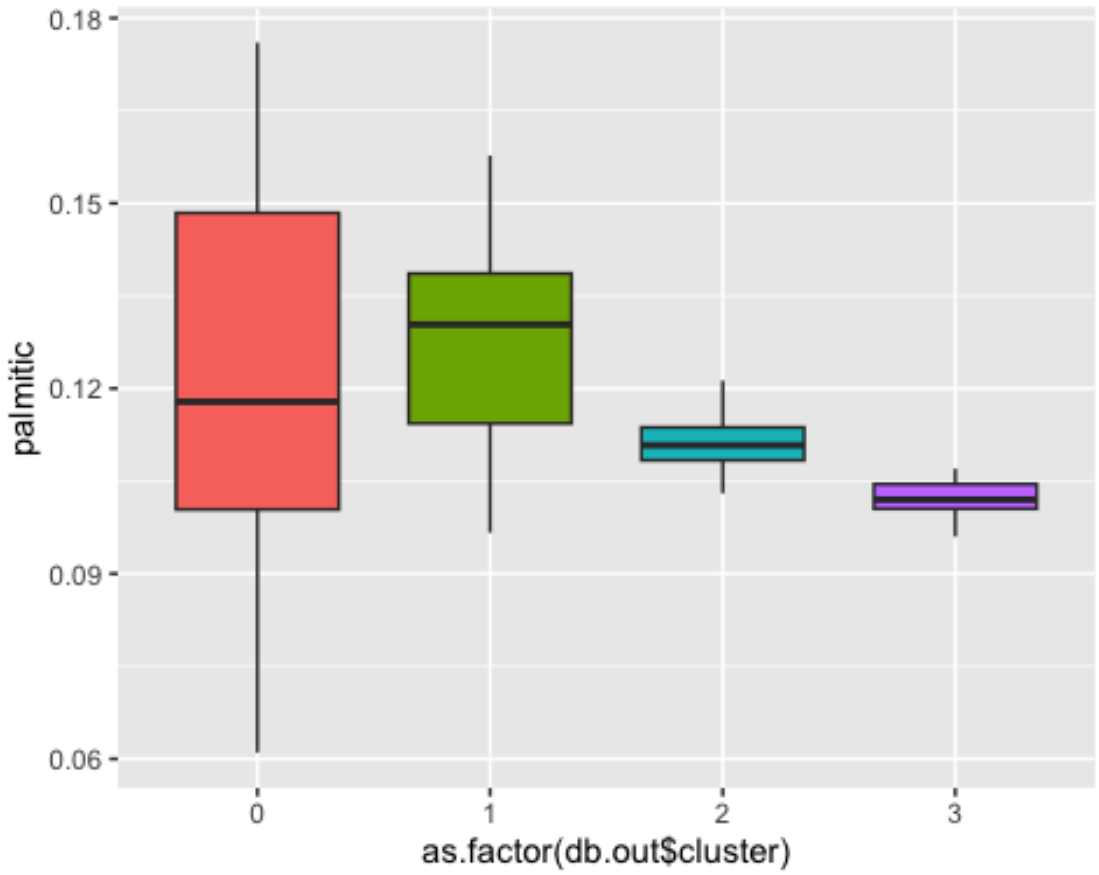
Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

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

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

# linoleic palmitoleic
plot(oliveoil$linoleic, oliveoil$palmitoleic, 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))
```



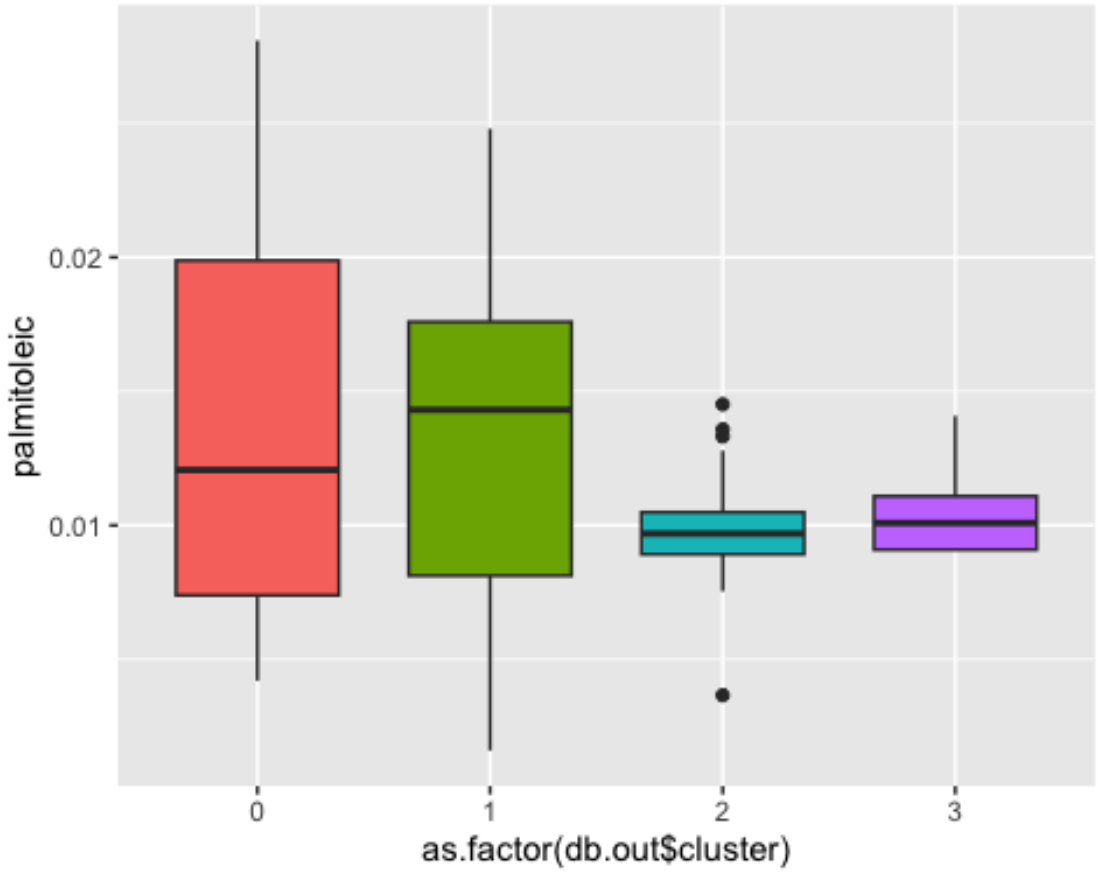
Variabile palmitic

```
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)
```



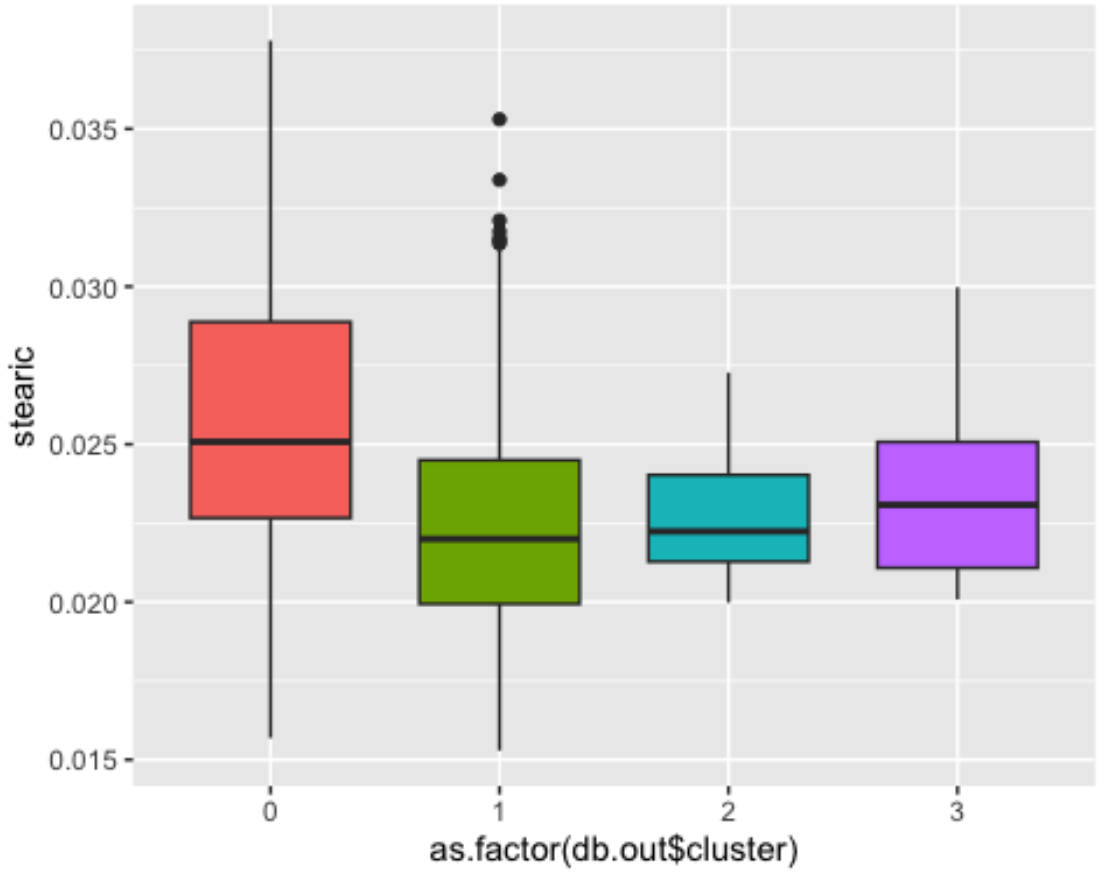
Variabile palmitoleic

```
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)
```



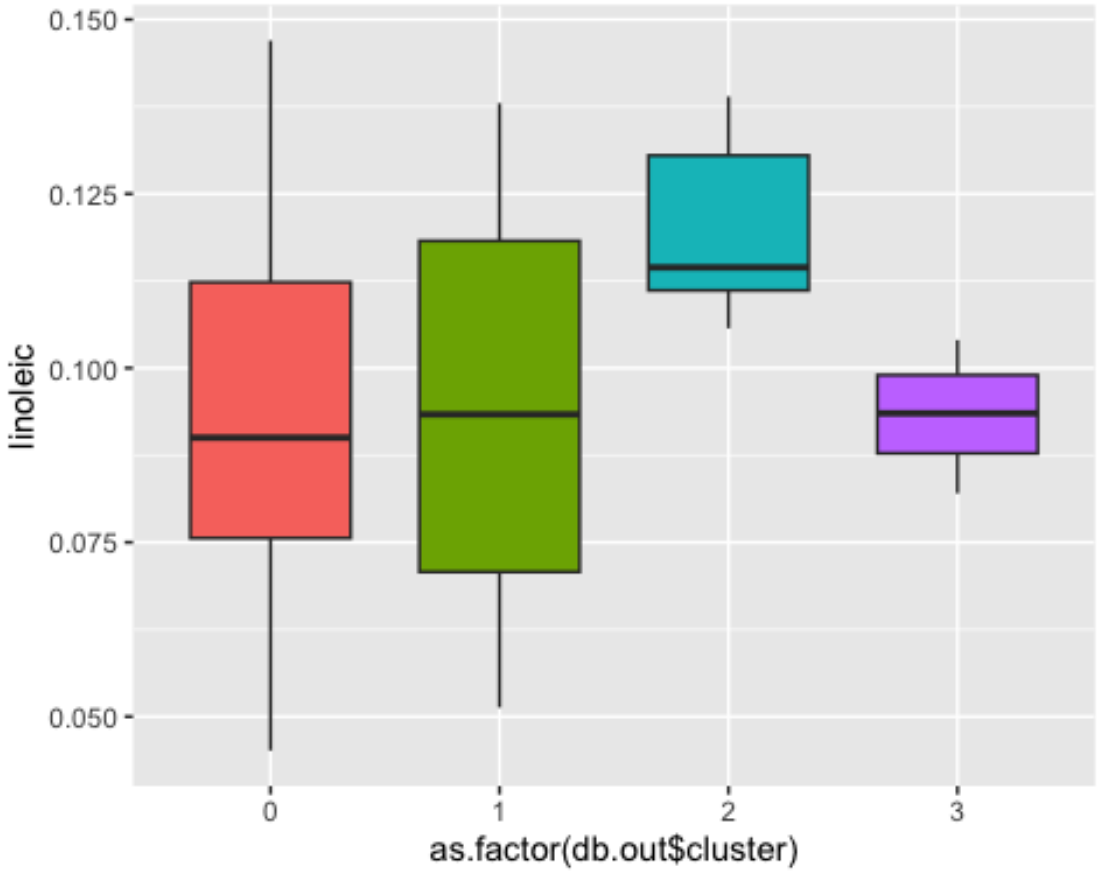
Variabile stearic

```
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)
```



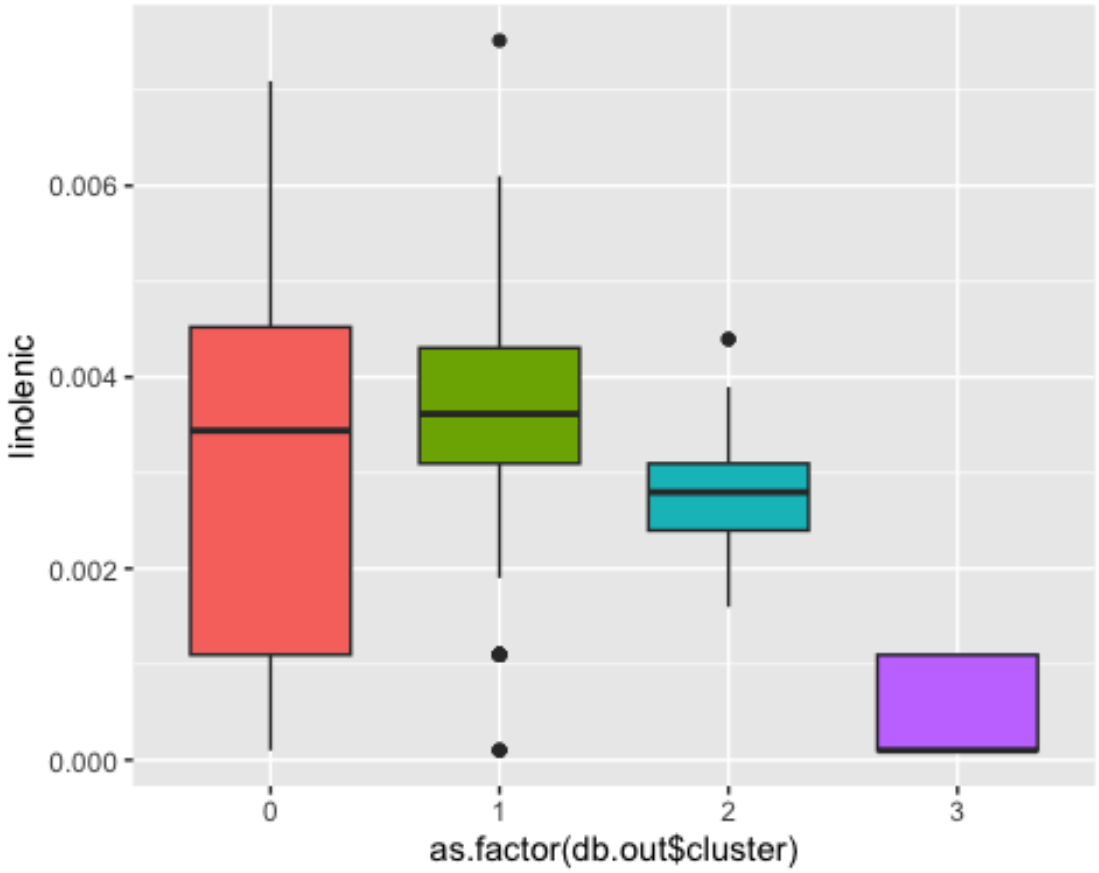
Variabile Linoleic

```
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)
```



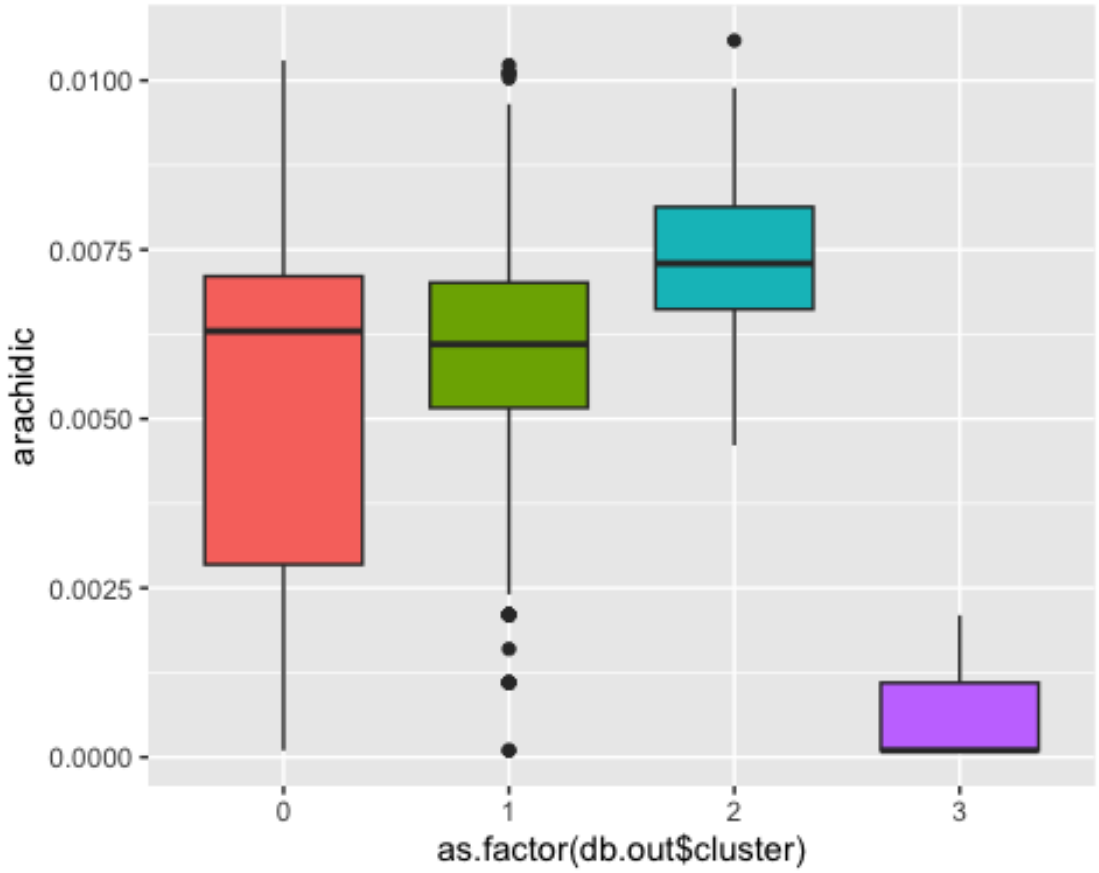
Variabile Linolenic

```
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)
```



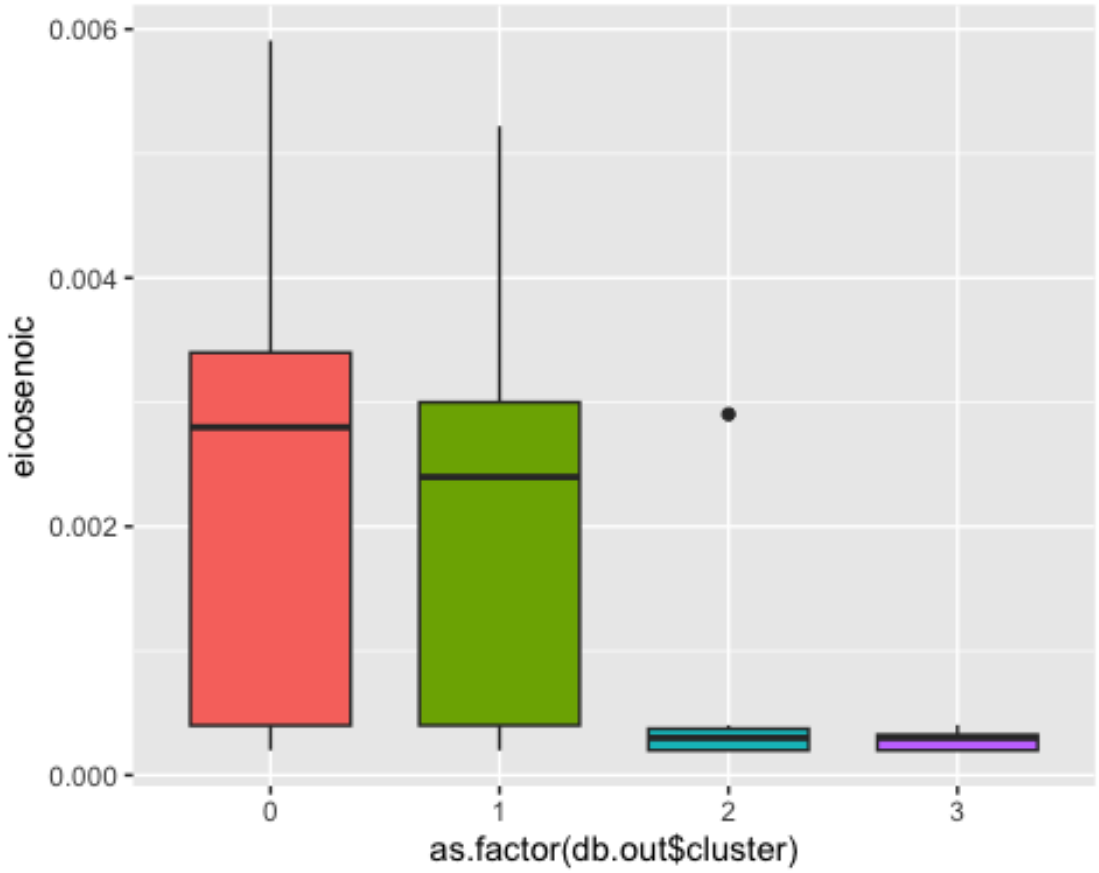
Variabile arachidic

```
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)
```



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)
```

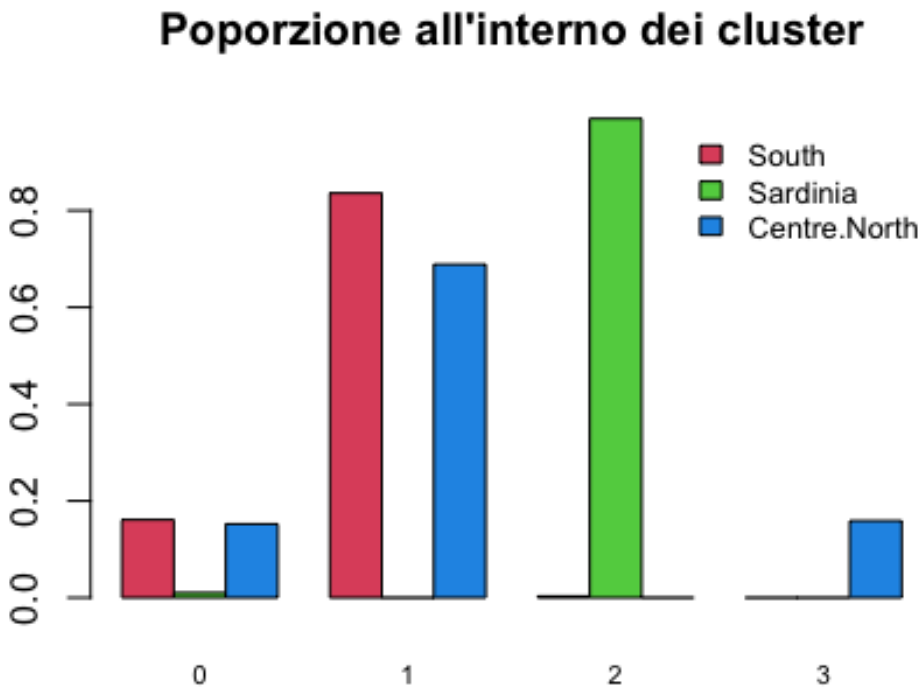


Variabile macro.area

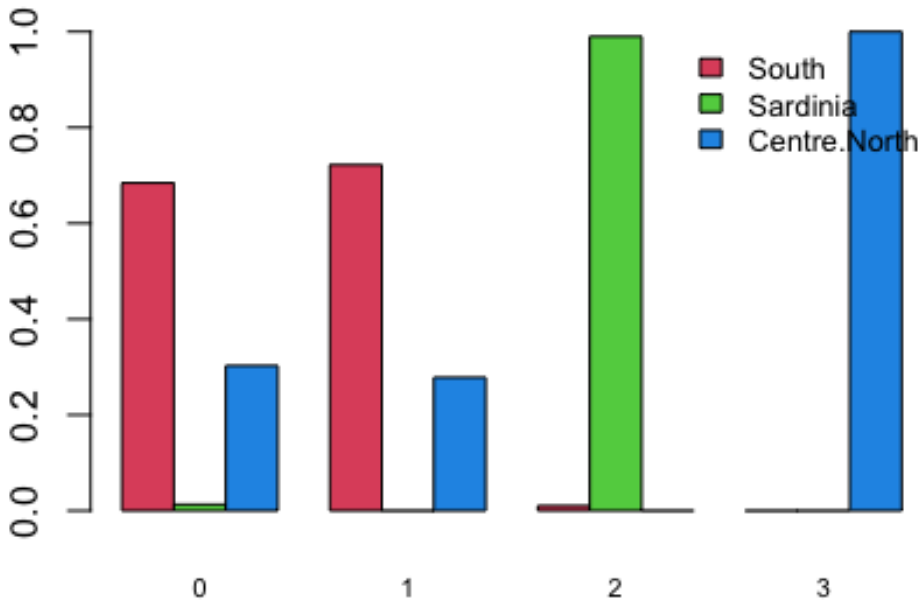
```
prop.table(table(db.out$cluster, oliveoil$macro.area),1)
```

```
##
##          South   Sardinia Centre.North
##   0 0.68421053 0.01315789   0.30263158
##   1 0.72192513 0.00000000   0.27807487
##   2 0.01020408 0.98979592   0.00000000
##   3 0.00000000 0.00000000   1.00000000
```

```
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")
```

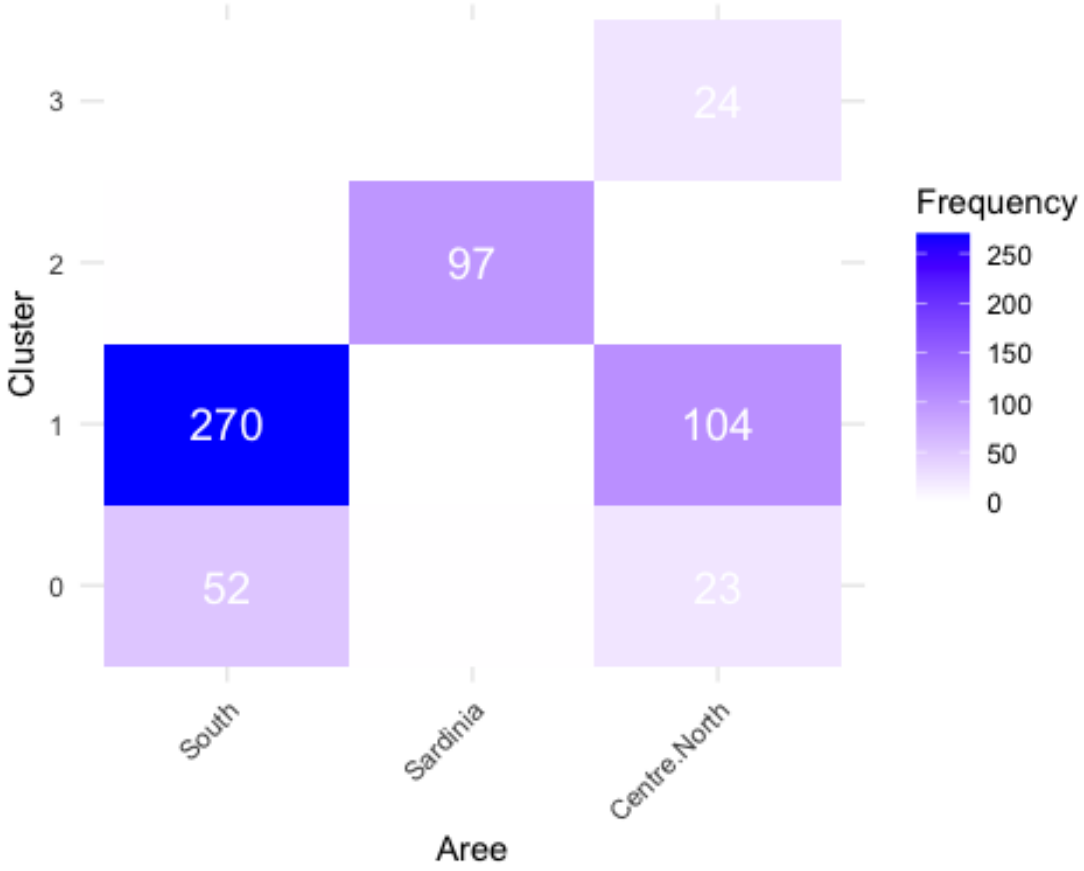


Delle tre distribuzioni in cluster il metodo dbscan senza trasformazione è quello che peggio riesce a dividere le varie macro aree in diversi cluster. Vediamo infatti che il primo cluster contiene una gran parte degli oli provenienti sia dal Sud che dal Centro Nord. La Sardegna rimane ancora la macro area meglio divisa in cluster essendo quasi perfettamente indicata con il cluster 2. Possiamo quindi dire che gli oli prodotti in questa Area hanno caratteristiche estremamente simili tra loro che oli provenienti da altre regioni non hanno.

Confusion Matrix:

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

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

##               Cluster
## Aree             0   1   2   3
##   South         52 270   1   0
##   Sardinia       1   0  97   0
##   Centre.North  23 104   0  24

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

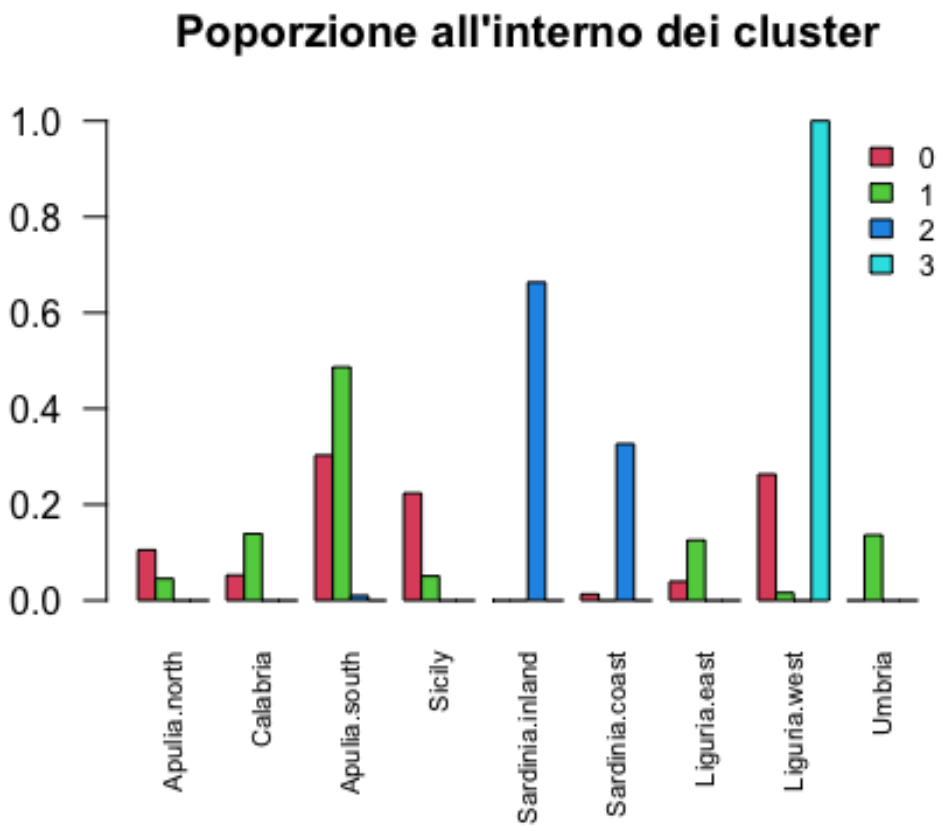


Variabile region

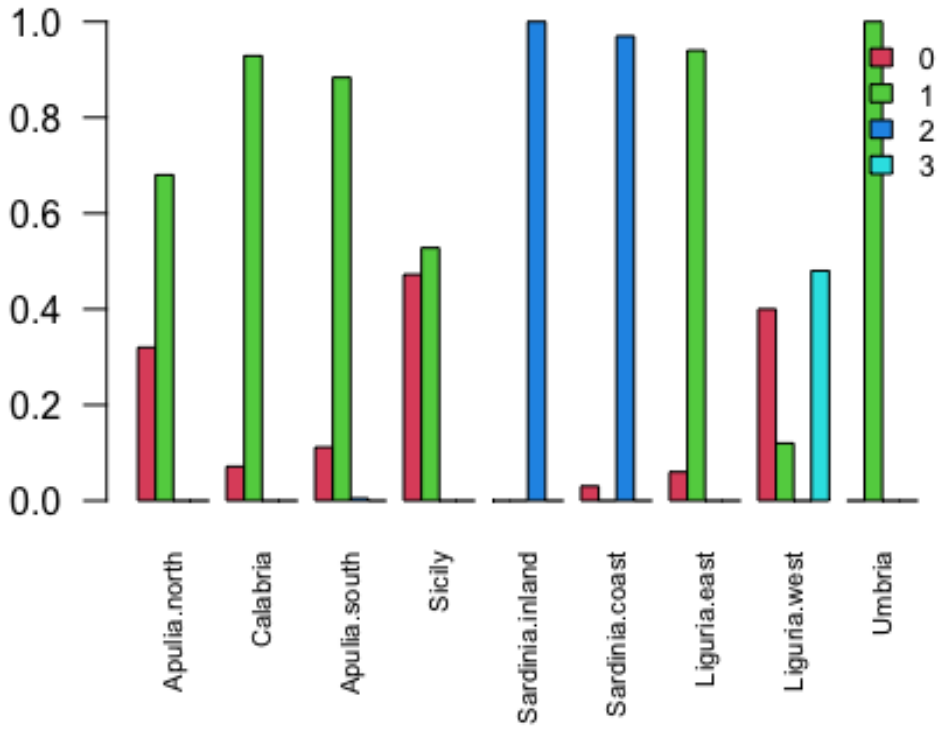
```
prop.table(table(db.out$cluster, oliveoil$region),1)
```

```
##
##     Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   0   0.10526316 0.05263158   0.30263158 0.22368421      0.00000000
##   1   0.04545455 0.13903743   0.48663102 0.05080214      0.00000000
##   2   0.00000000 0.00000000   0.01020408 0.00000000      0.66326531
##   3   0.00000000 0.00000000   0.00000000 0.00000000      0.00000000
##
##     Sardinia.coast Liguria.east Liguria.west     Umbria
##   0     0.01315789   0.03947368   0.26315789 0.00000000
##   1     0.00000000   0.12566845   0.01604278 0.13636364
##   2     0.32653061   0.00000000   0.00000000 0.00000000
##   3     0.00000000   0.00000000   1.00000000 0.00000000
```

```
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")
```




```
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")
```



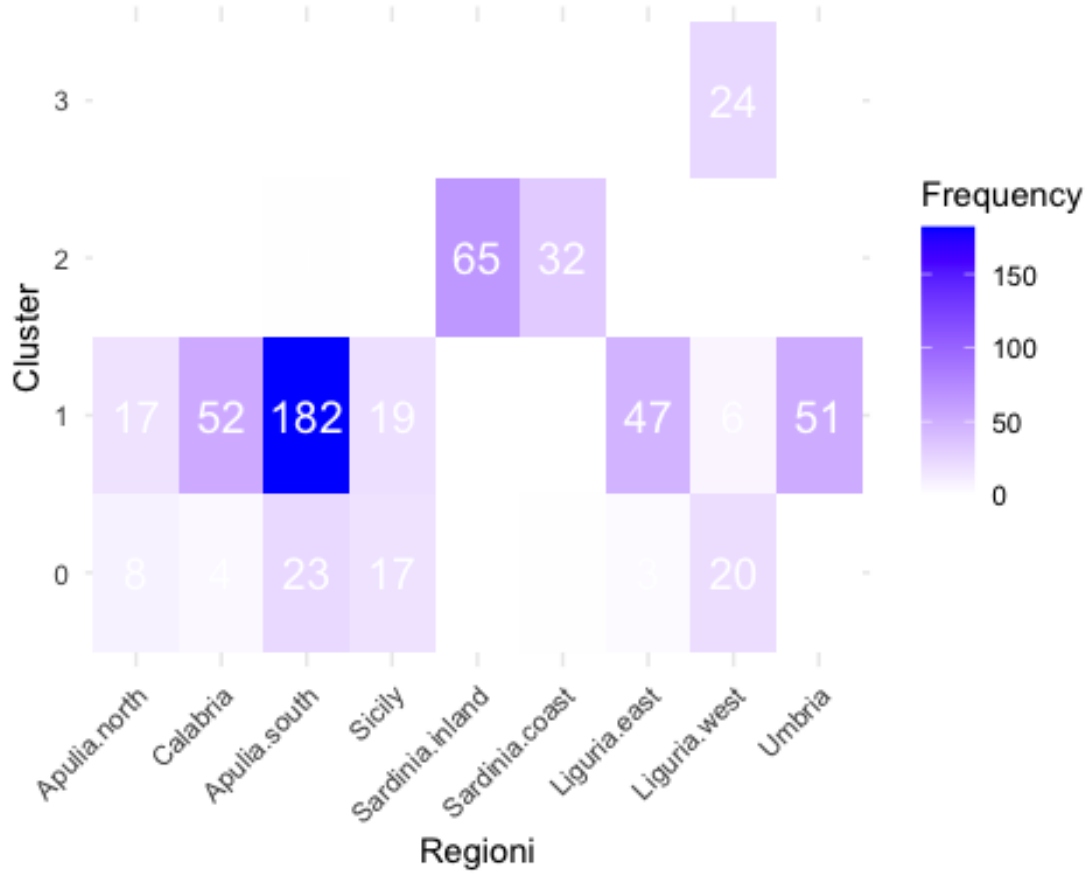
Vediamo subito che quasi tutta la penisola italiana e la Sicilia fanno parte del cluster 1 Delle osservazioni che si possono fare guardando i grafici delle regioni sono che il cluster 3 va in questo caso a prendere esclusivamente oli prodotti in Liguria dell'Ovest; e che i punti di "noise" sono presenti in tutte le regioni fatta eccezione dell'Umbria e dell'entroterra Sardo.

Confusion Matrix:

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

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

```
##                  Cluster
## Regioni            0   1   2   3
##   Apulia.north     8  17   0   0
##   Calabria         4  52   0   0
##   Apulia.south    23 182   1   0
##   Sicily          17  19   0   0
##   Sardinia.inland  0   0  65   0
##   Sardinia.coast   1   0  32   0
##   Liguria.east     3  47   0   0
##   Liguria.west    20   6   0  24
##   Umbria           0  51   0   0
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Regioni, y =
Cluster, 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))
```



Adjusted Rand Index - DBSCAN

```
ari_db <- adj.rand.index(oliveoil$macro.area, db.out$cluster)
ari_db
```

```
## [1] 0.3906598
```

Mappa dei cluster sulla cartina Italiana

```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=db.out$cluster+1, pch=19, cex=0.3)
```

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

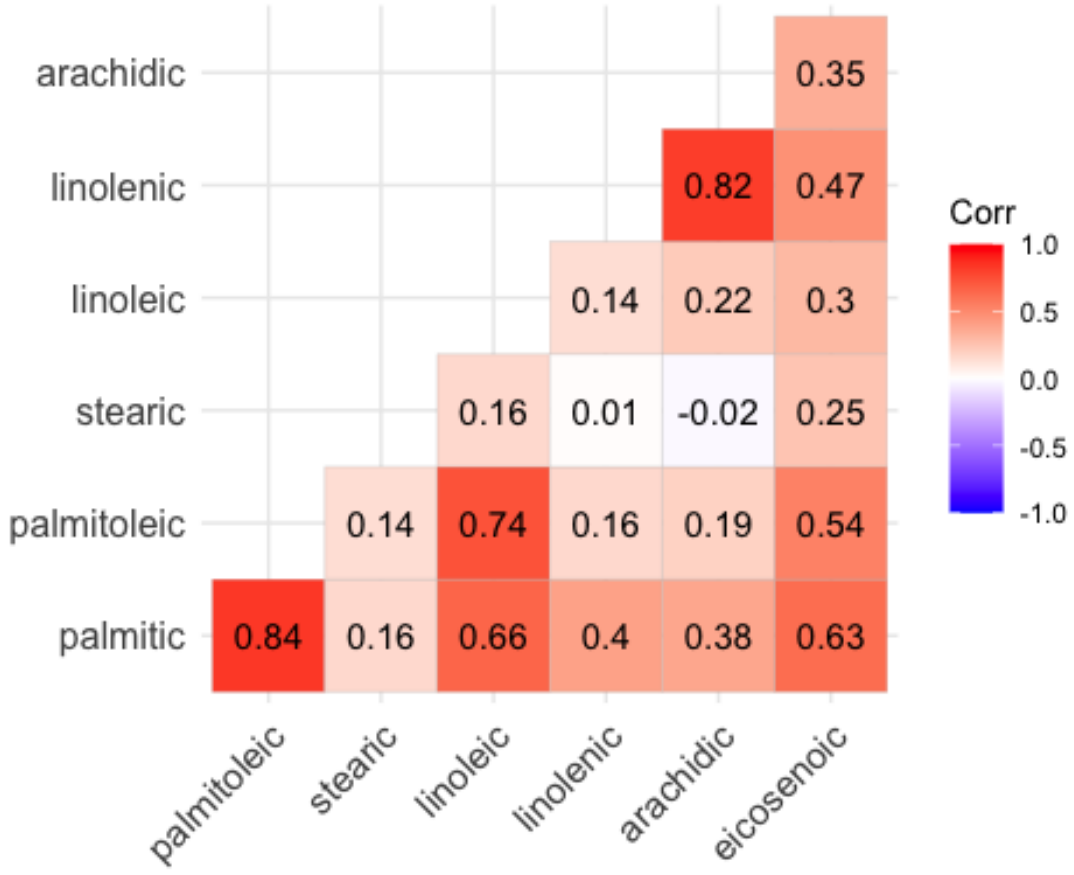
cluster nei dati meno correlati. In quanto l'acido oleico è il più presente ed il più correlato con le altre variabili

re-import del dataset

```
data("oliveoil")

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])
}

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

Si vuole analizzare ora il dataset trasformato 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.

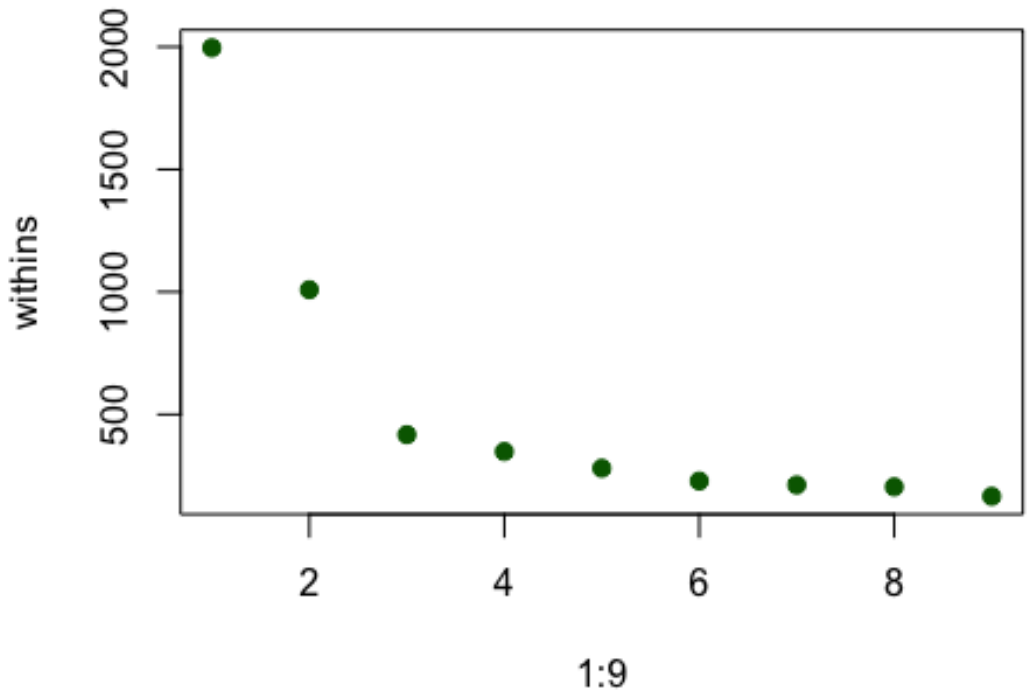
Altre informazioni degne di nota riguardano:

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, pch = 19, col ="darkgreen")
```



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)
```

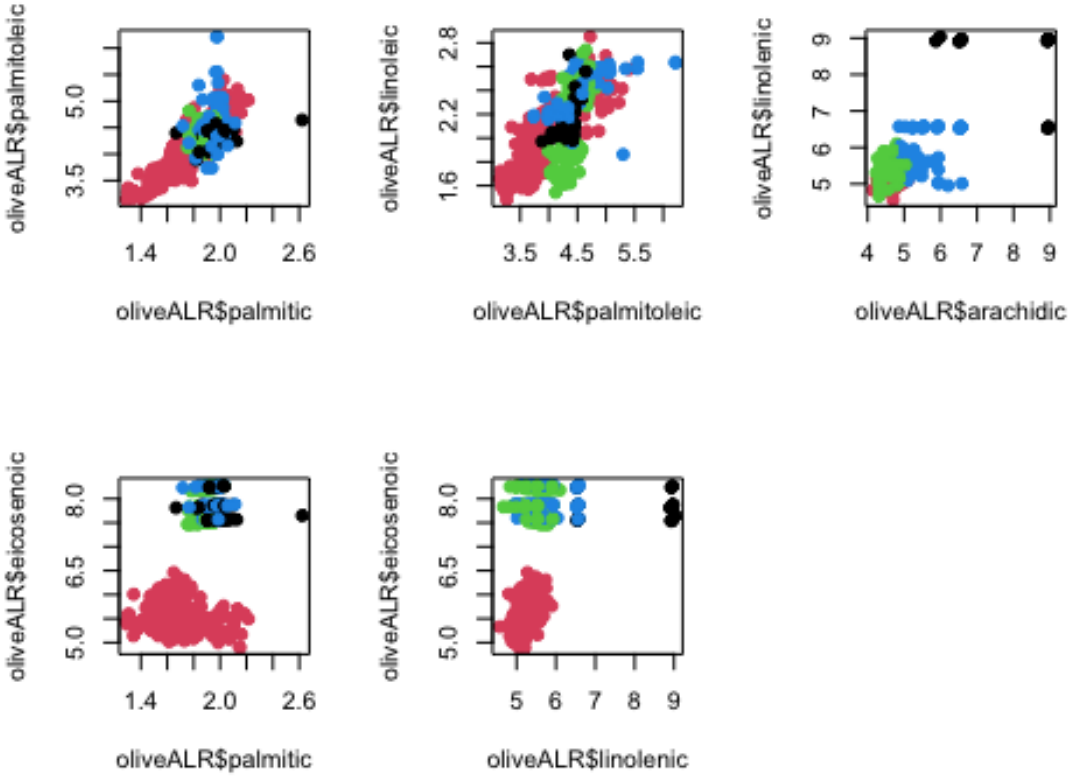
```
## List of 9
##  $ cluster     : int [1:572] 2 2 2 2 2 2 2 2 2 ...
##  $ centers     : num [1:4, 1:7] 2 1.68 1.89 1.97 4.34 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##  $ totss       : 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"
```

```
km.out$size
```

```
## [1]  37 323 132  80
```

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.

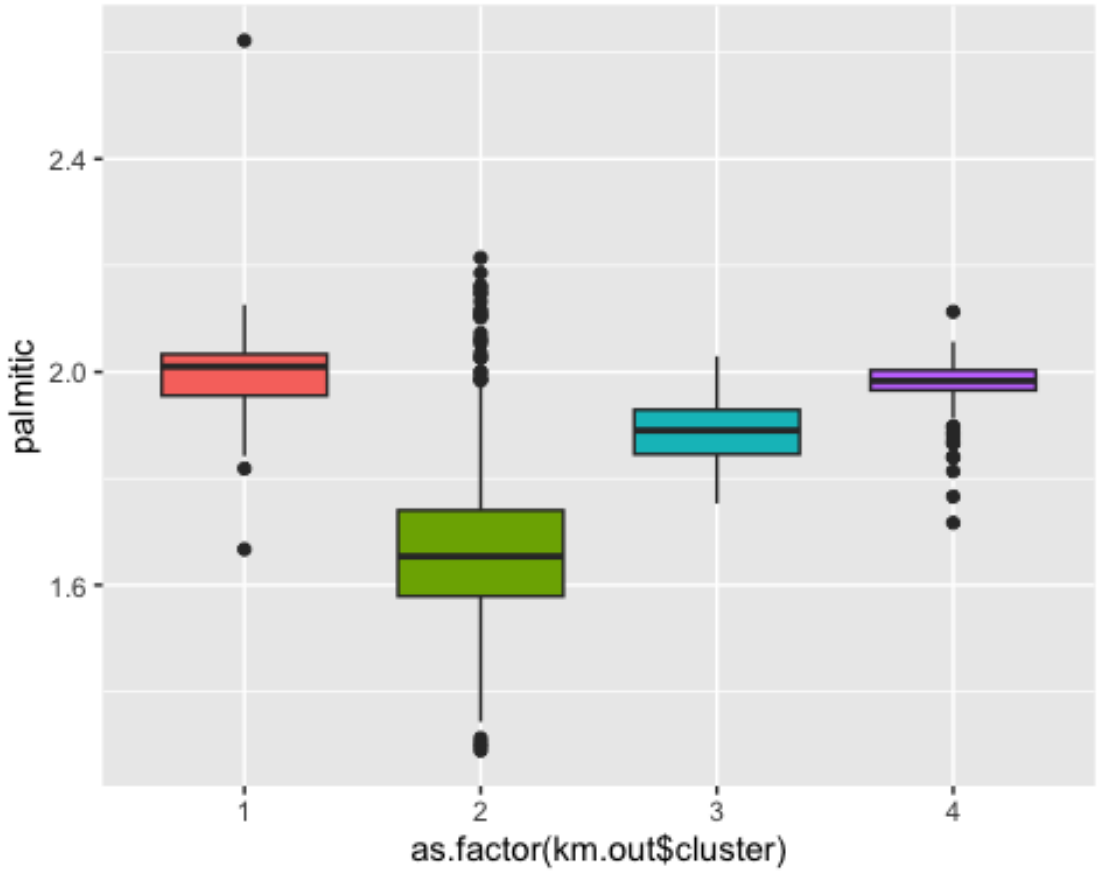
```
par(mfrow=c(2,3))

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

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

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

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

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

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



Si nota una buona divisione in cluster rispetto alle variabili palmitic e palmitoleic con un cluster identificato dal colore rosso per valori di concentrazione più bassi

Variabile palmitic nei cluster

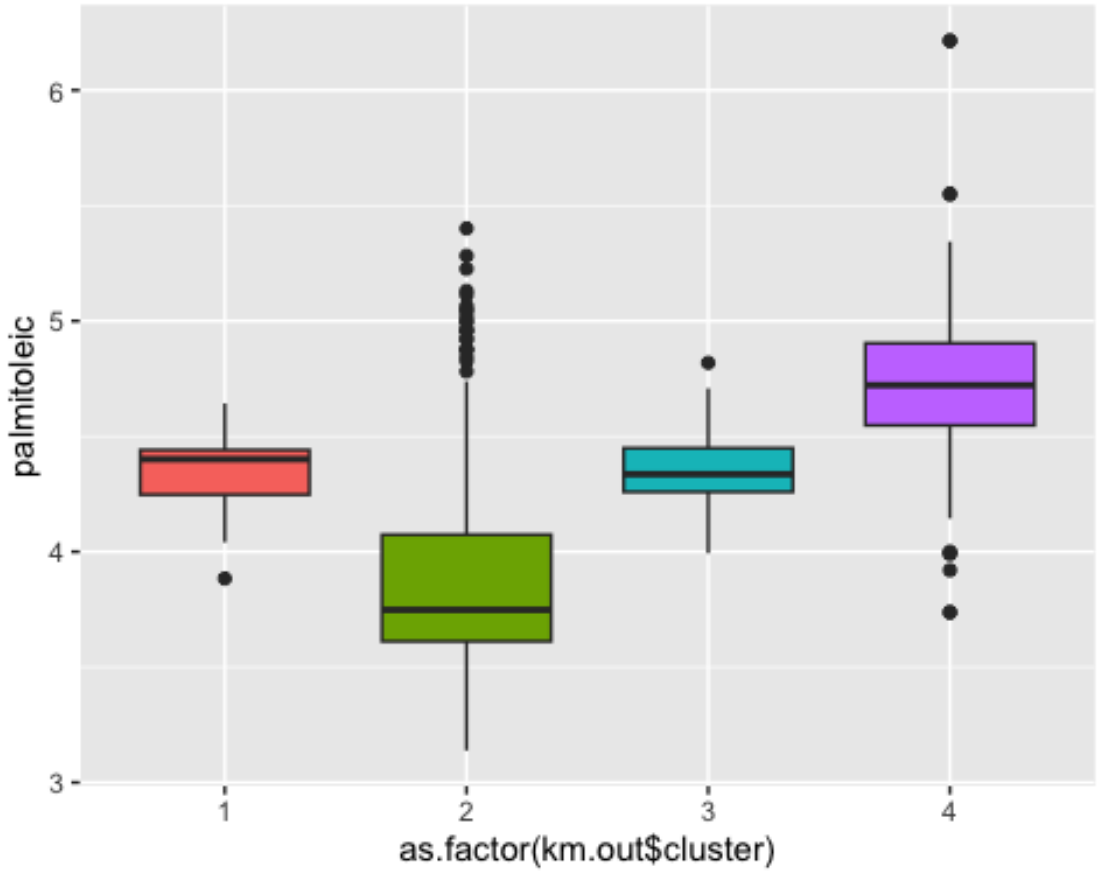
```
ggplot(oliveALR, aes(x = as.factor(km.out$cluster), y = palmitic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Nei gruppi 1, 3 e 4 la mediana dei valori è circa allineata Mentre il gruppo 2 presenta numerosi valori outlier oltre che ad avere una varianza più elevata.

Variabile palmitoleic nei cluster

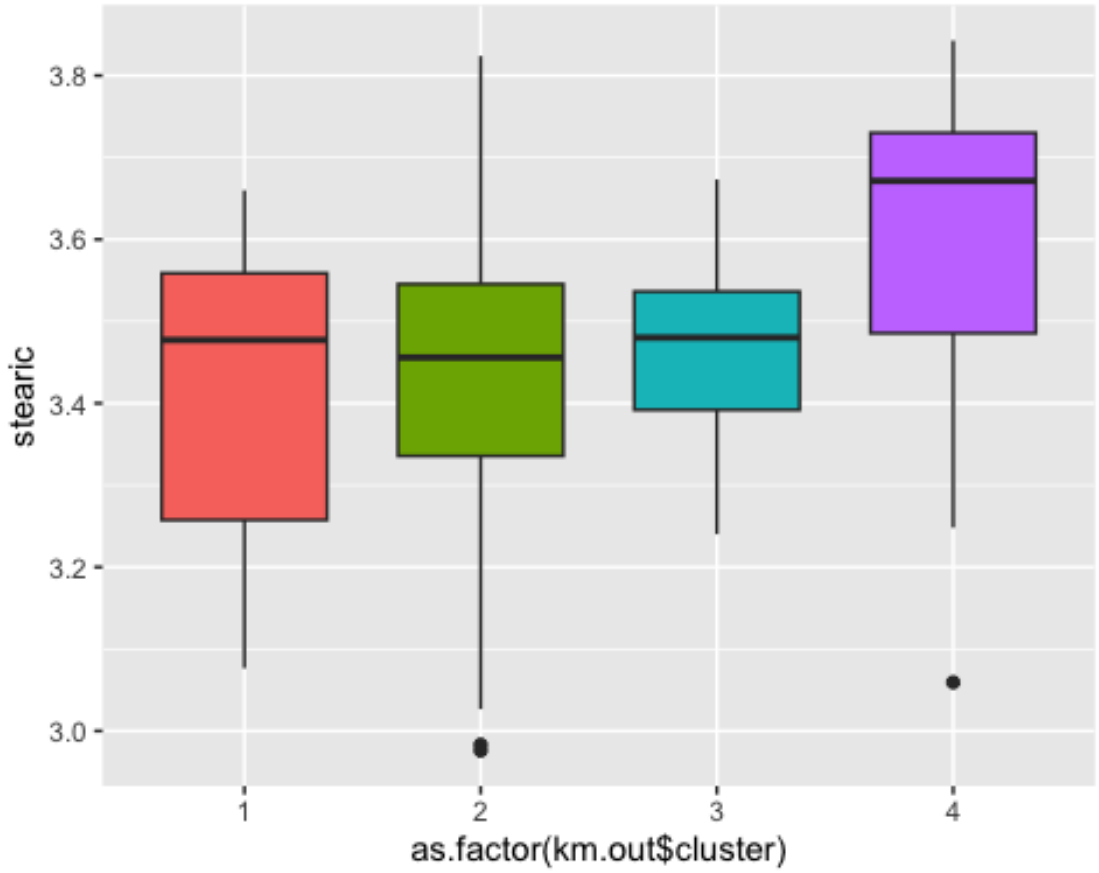
```
ggplot(oliveALR, aes(x = as.factor(km.out$cluster), y = palmitoleic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Analogamente alla variabile palmitic, si nota un cluster 2 con una devianza within più elevata rispetto agli altri gruppi, che hanno anch'essi una mediana più vicina: Il cluster 1 e 3 sono abbastanza simili tra loro, Il cluster 2 preneta numerosi valori outlier. Il cluster 4 ha dei valori outlier più distanti dalle code della distribuzione.

Variabile `stearic` nei cluster

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

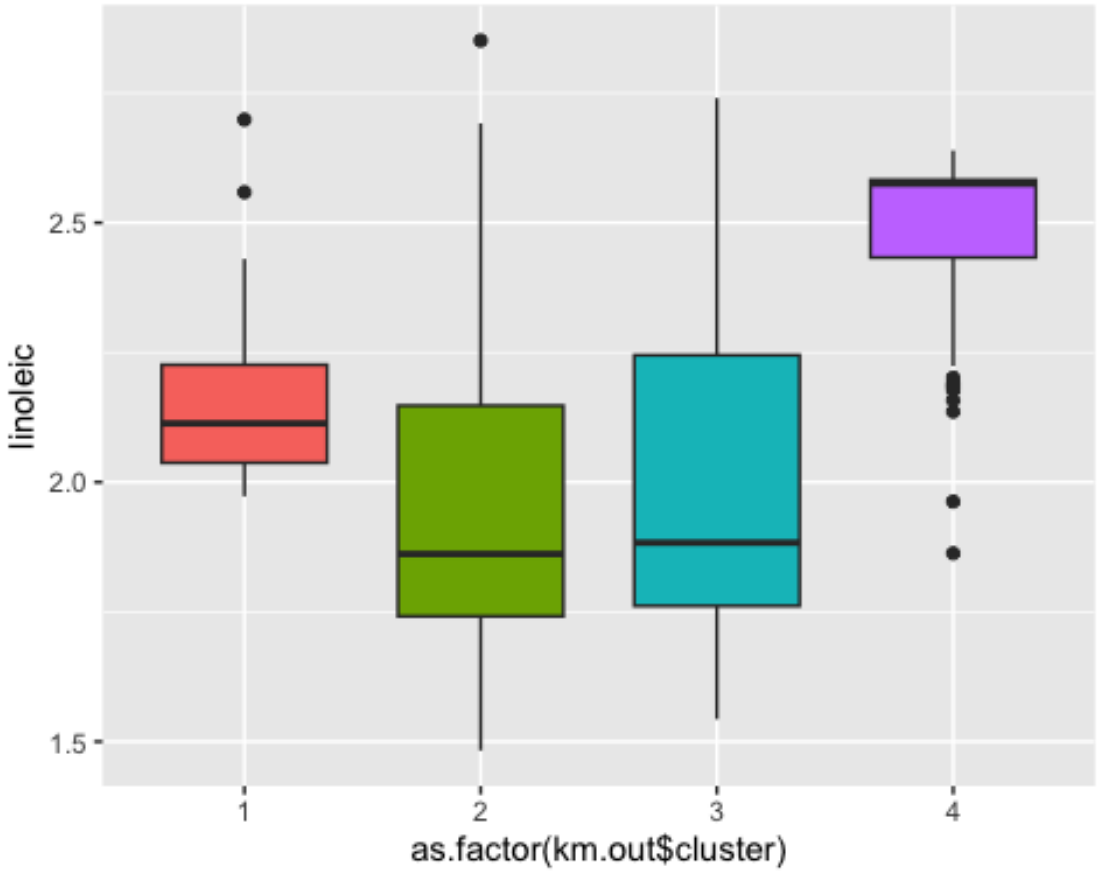



```
median(oliveALR$stearic[km.out$cluster == 4])

## [1] 3.67144
```

I cluster 1, 2 e 3 hanno delle mediane abbastanza simili tra loro, il che indica una bassa devianza between. Il cluster 1 ha una devianza within più elevata ripetto agli altri. Il cluster 4 ha una mediana superiore a 3.65

Variabile `linoleic` nei cluster

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

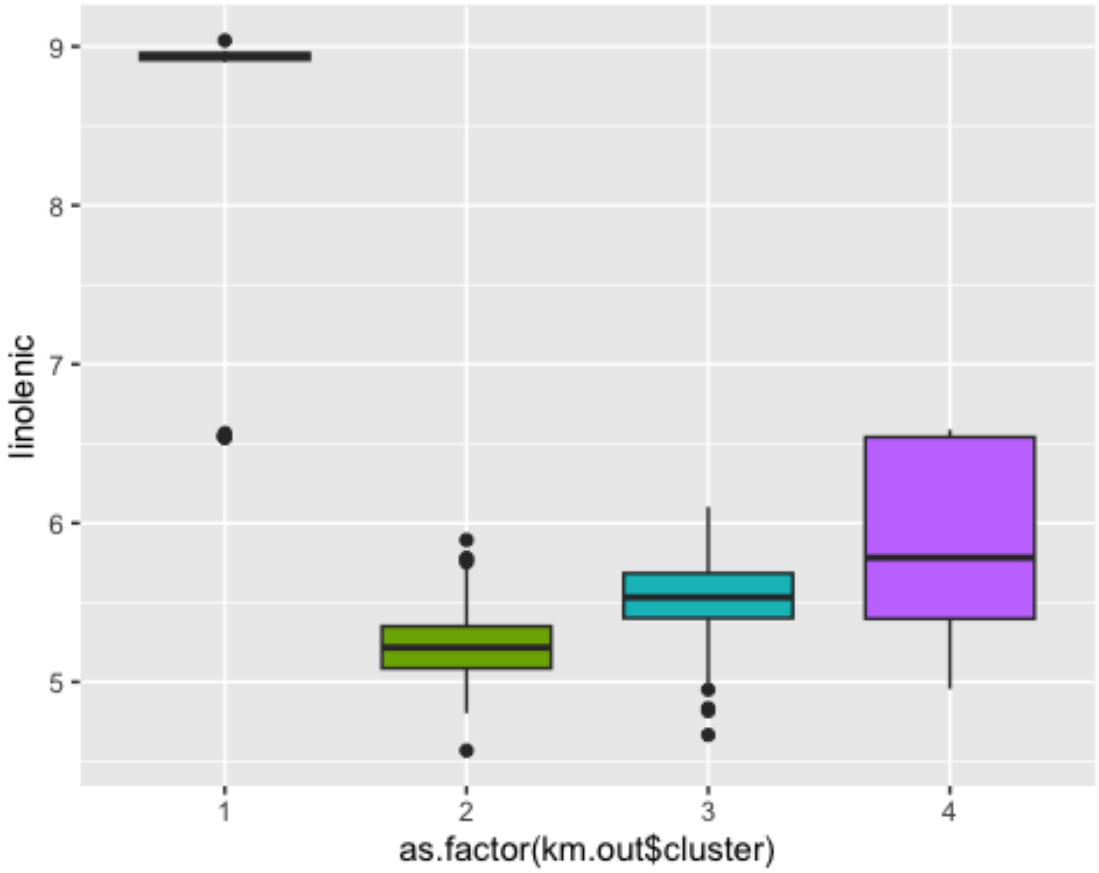



```
median(oliveALR$linoleic[km.out$cluster == 4])

## [1] 2.575116
```

Si nota che i cluster 2 e 3 hanno delle distribuzioni di acido linoleico simili, con una mediana vicina a 1.8. Inoltre questi cluster hanno una devianza within più alta. Il quarto cluster presenta una mediana superiore a 2.57 e numerosi valori outlier.

Variabile `linolenic` nei cluster

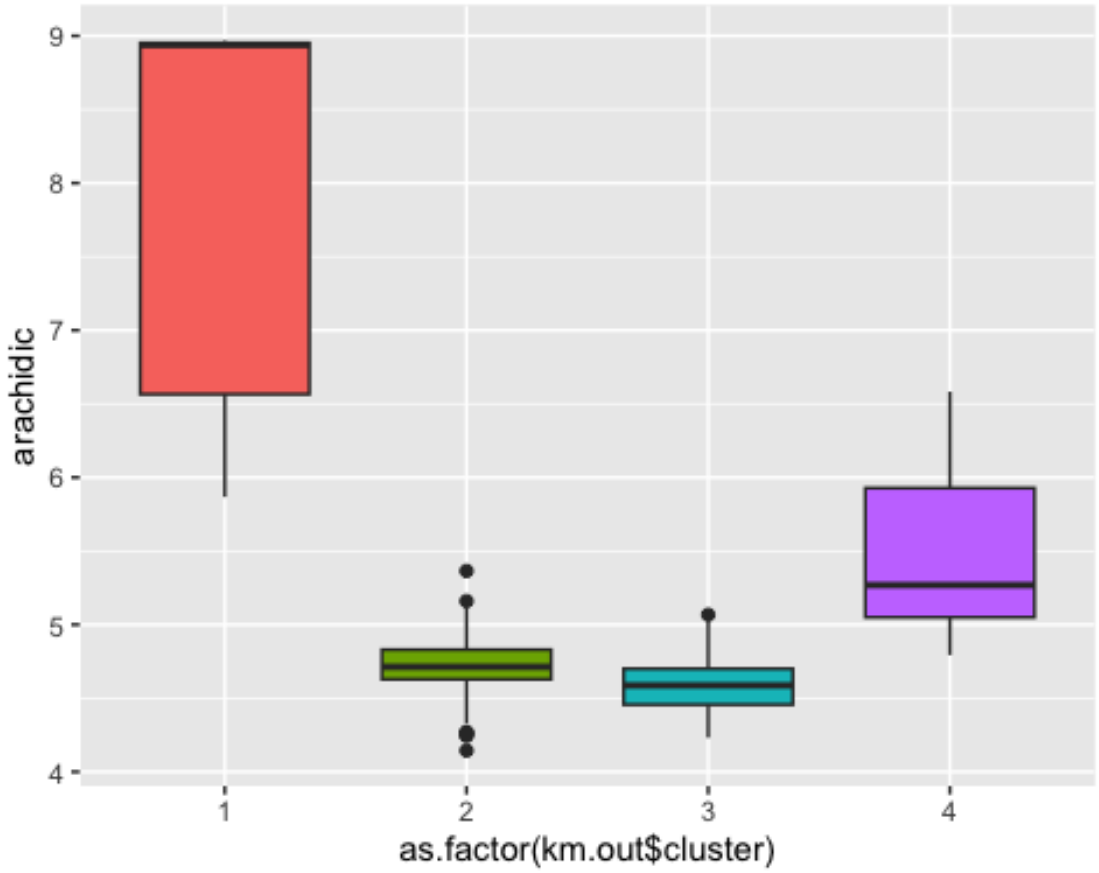
```
ggplot(oliveALR, aes(x = as.factor(km.out$cluster), y = linolenic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Si nota una maggiore disparità tra i cluster rispetto agli altri acidi, Il cluster 1 contiene tutti gli oli che hanno una quantità di acido linoleico circa uguale a 9, Il cluster 3 presenta alcuni valori outlier mentre il cluster 4 sembra avere una devianza within maggiore degli altri cluster.

Variabile arachidic nei cluster

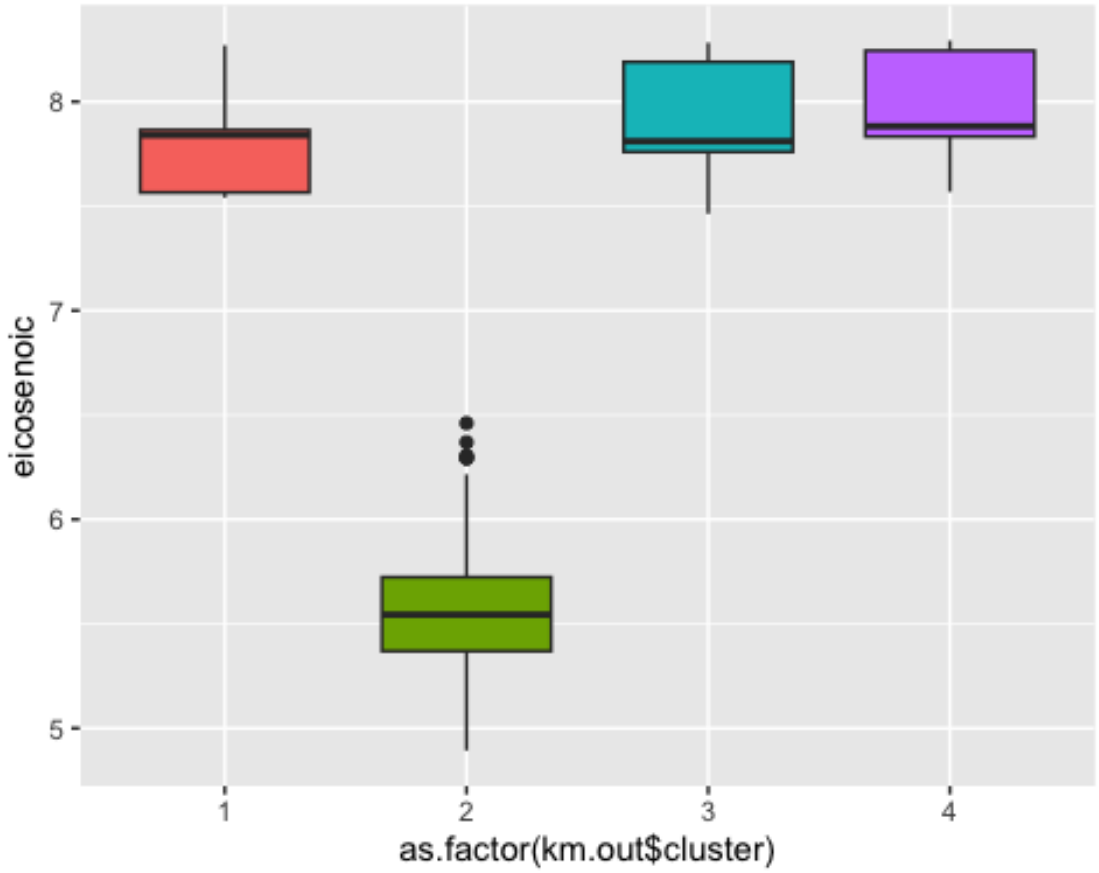
```
ggplot(oliveALR, aes(x = as.factor(km.out$cluster), y = arachidic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



La concentrazione di acido arachidico è distribuito tra i cluster in modo simile all'acido linolenico, si nota in questo caso che gli oli contenuti nel cluster 1 hanno una devianza within molto estera, e che i cluster 2 e 3 hanno una distribuzione abbastanza simile

Variabile eicosenoic nei cluster

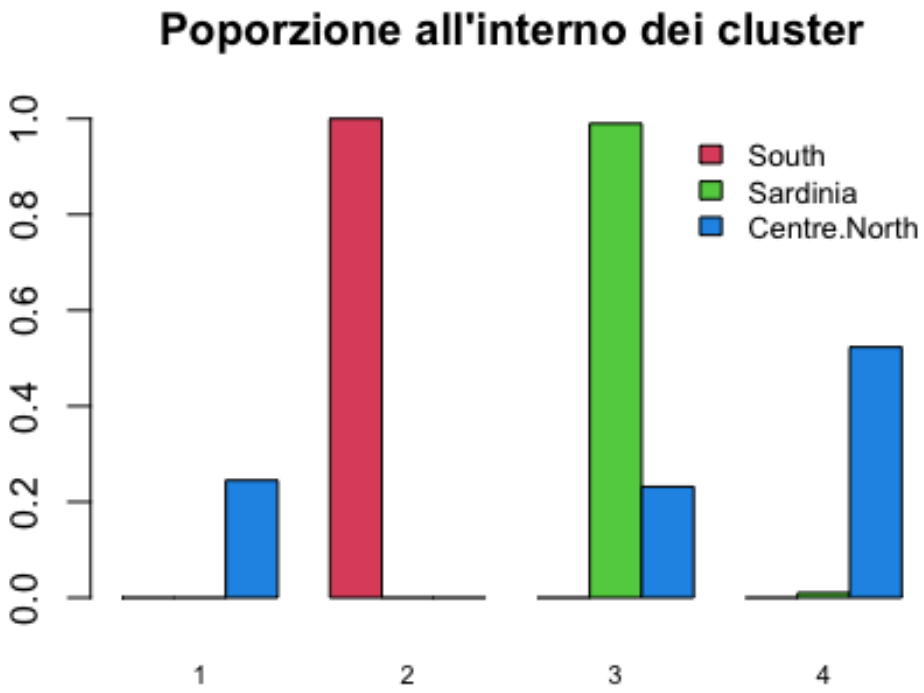
```
ggplot(oliveALR, aes(x = as.factor(km.out$cluster), y = eicosenoic, fill =
as.factor(km.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



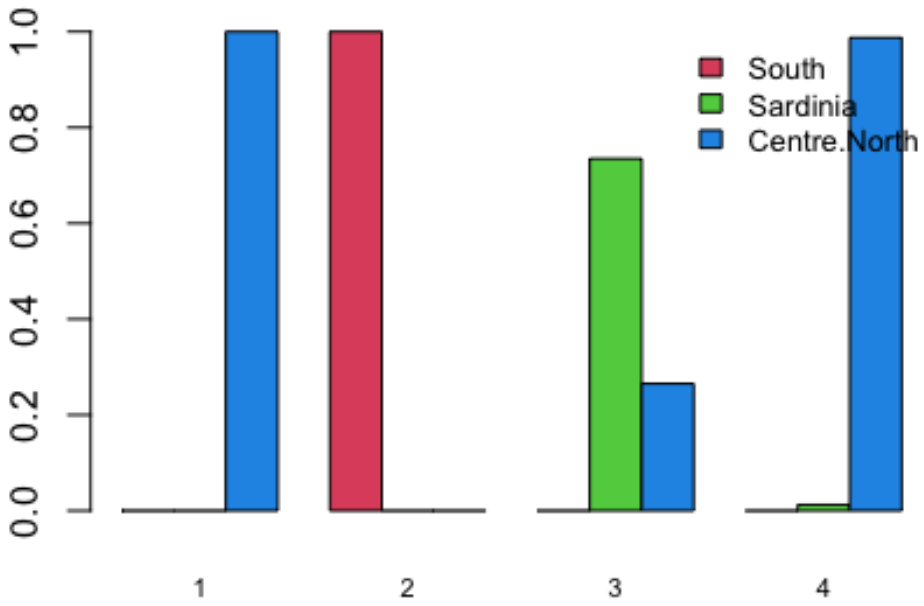
Il cluster 2 contiene alcuni valori outlier, mentre gli altri cluster hanno tutti una mediana vicina tra loro, intorno al 7.9 I cluster 3 e 4 presentano una distribuzione simile.

Variabile macro.area

```
round(prop.table(table(oliveALR$macro.area, km.out$cluster),1), 3)

##
##                    1     2     3     4
##   South        0.000 1.000 0.000 0.000
##   Sardinia     0.000 0.000 0.990 0.010
##   Centre.North 0.245 0.000 0.232 0.523

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")
```



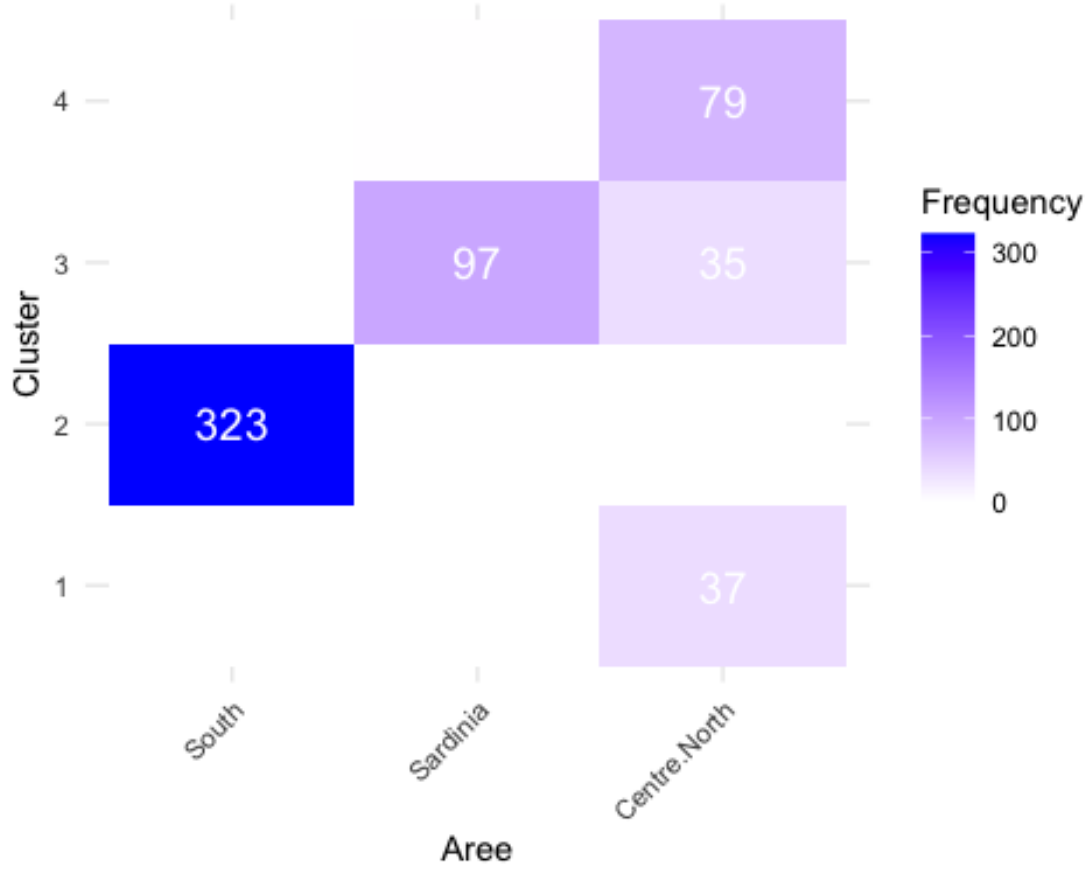
Gli oli del Sud si trovano al 100% nel cluster 2 Gli oli della Sardegna al 99% nel cluster 3 e 1% nel cluster 4. Gli oli del centro nord al 25% nel cluster 1, 23% nel cluster 3 e 52% nel cluster 4.

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

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

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

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



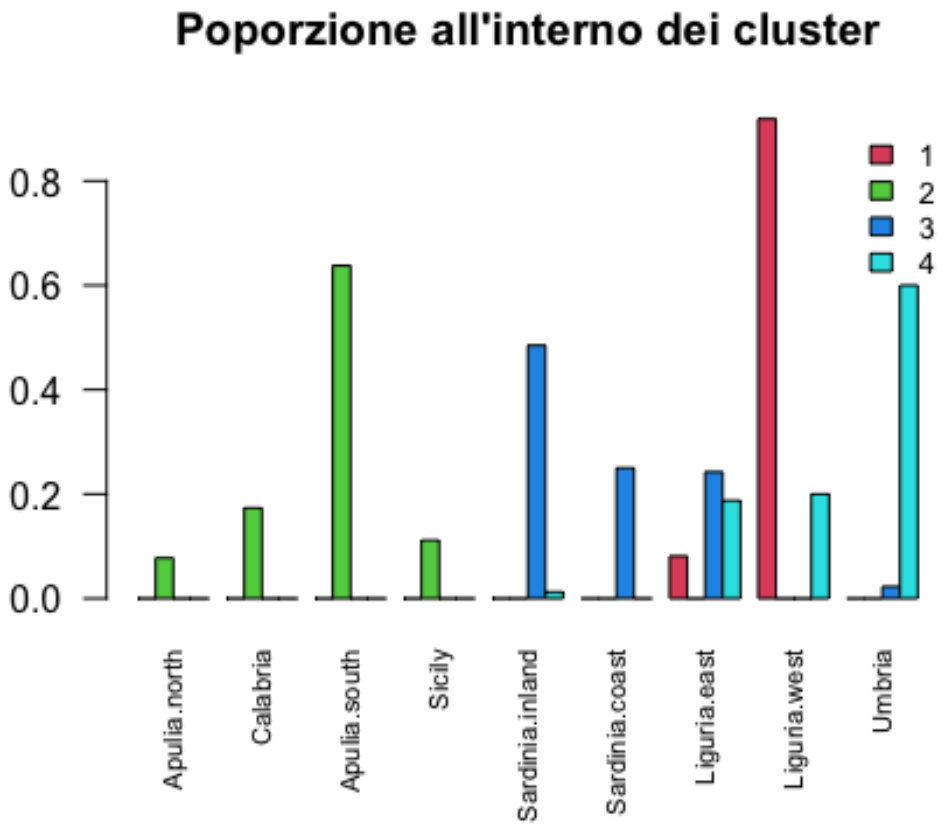
Variabile region

```
round(prop.table(table(km.out$cluster, oliveALR$region),1), 3)
```

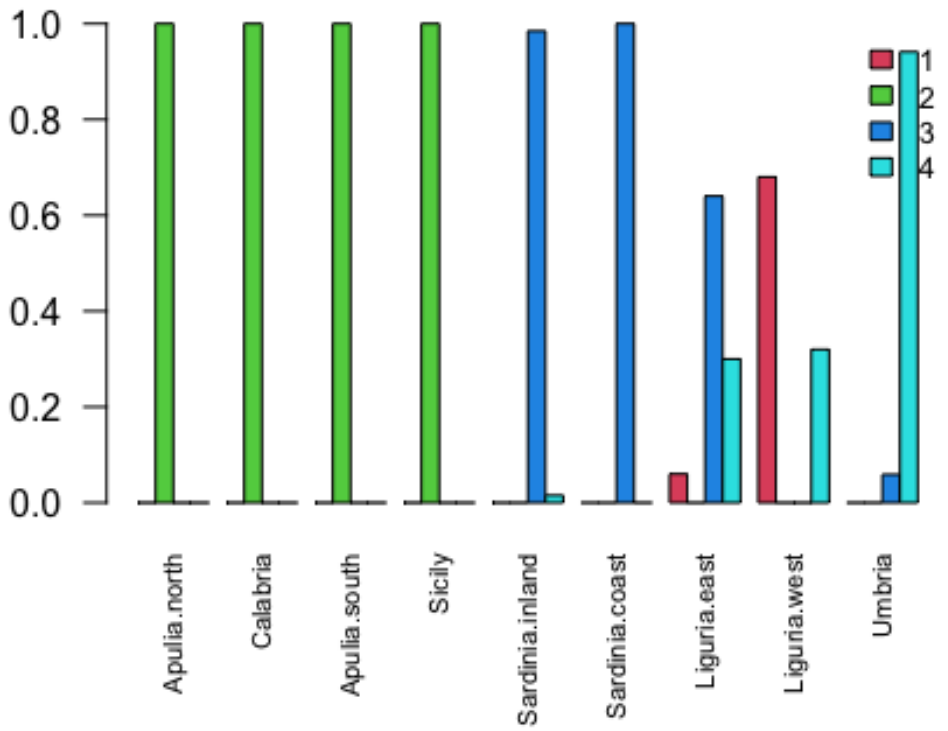
```
##
##     Apulia.north Calabria Apulia.south Sicily Sardinia.inland
Sardinia.coast
##   1        0.000    0.000        0.000  0.000           0.000
0.000
##   2        0.077    0.173        0.638  0.111           0.000
0.000
##   3        0.000    0.000        0.000  0.000           0.485
0.250
##   4        0.000    0.000        0.000  0.000           0.013
0.000
##
##     Liguria.east Liguria.west Umbria
##   1        0.081        0.919  0.000
```

```
##    2        0.000         0.000  0.000
##    3        0.242         0.000  0.023
##    4        0.188         0.200  0.600
```

```
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")
```




```
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.

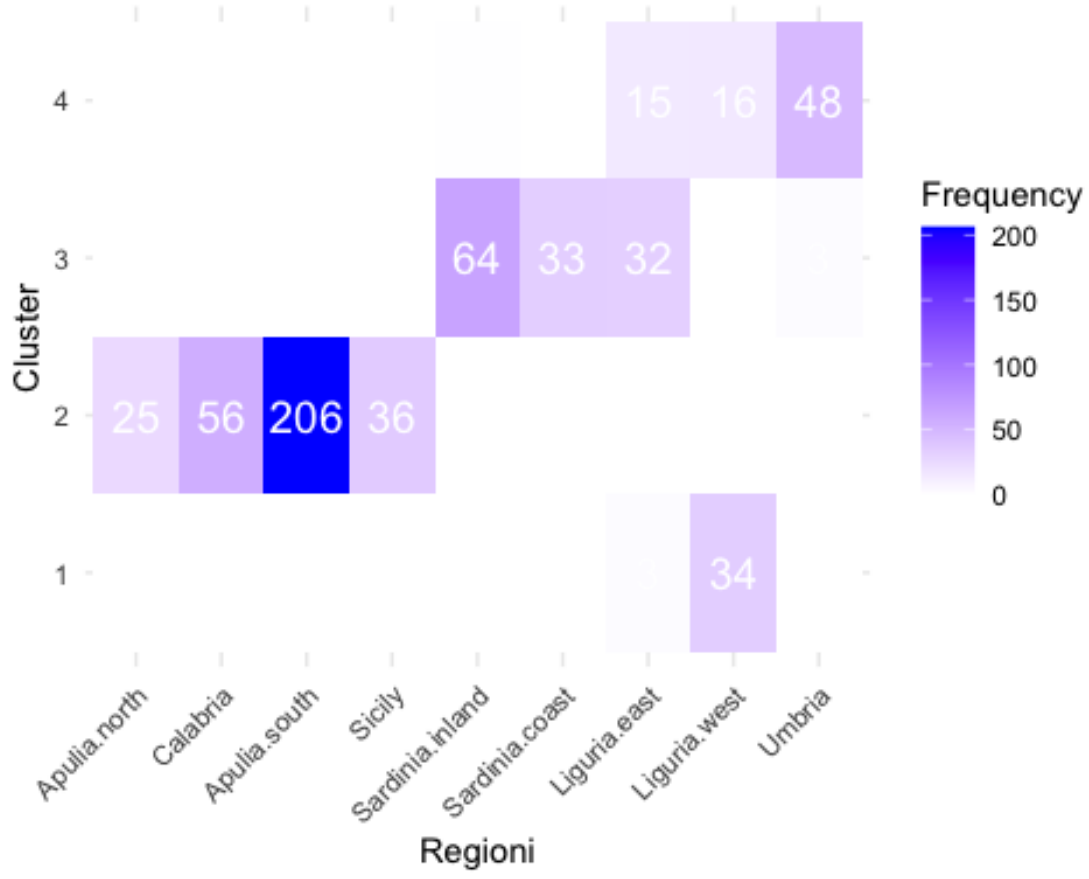
Il cluster 1 è formato al 91% da oli della liguria ovest e al 9% da oli della liguria est. Il cluster 2 è formato al 17% da oli della calabria, al 64% da oli della puglia sud, 8% da oli della puglia del nord e 11% da oli della Sicilia. Il cluster 3 è formato al 48% da oli della Sardegna inland, al 25% 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(Regioni = oliveALR$region, Cluster =
km.out$cluster)

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

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

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Regioni, y =
Cluster, 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))
```



Adjusted Rand Index - K Means

```
ari_km_ln <- adj.rand.index(oliveALR$macro.area, km.out$cluster)
ari_km_ln

## [1] 0.8660622
```

Mappa dei cluster sulla cartina Italiana

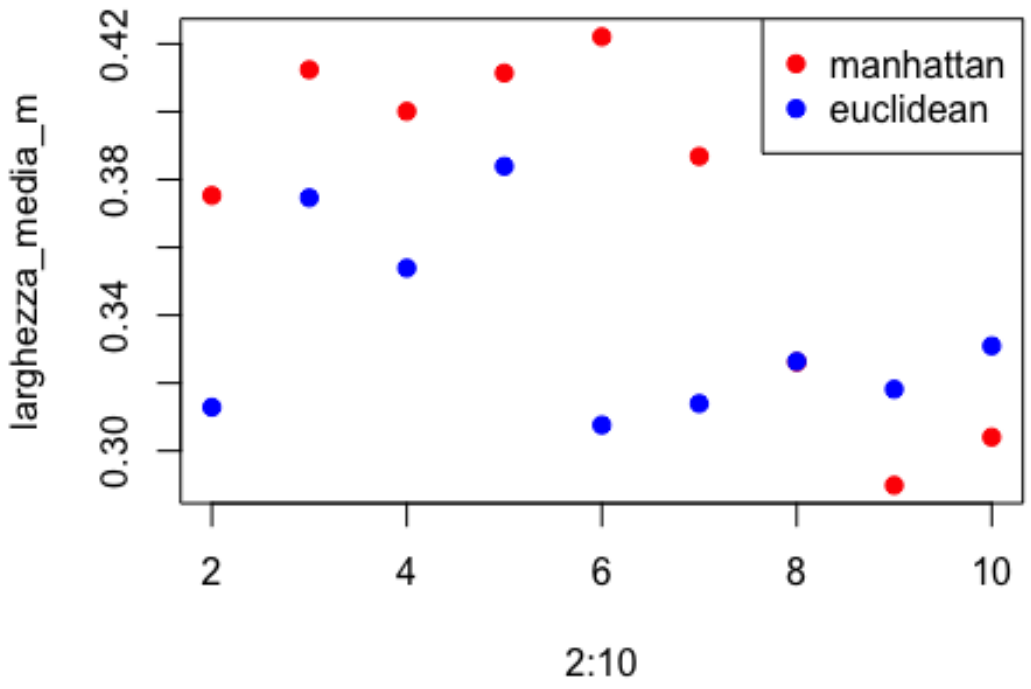
```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=km.out$cluster+1, pch=19, cex=0.3)
```

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
```

```r
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

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

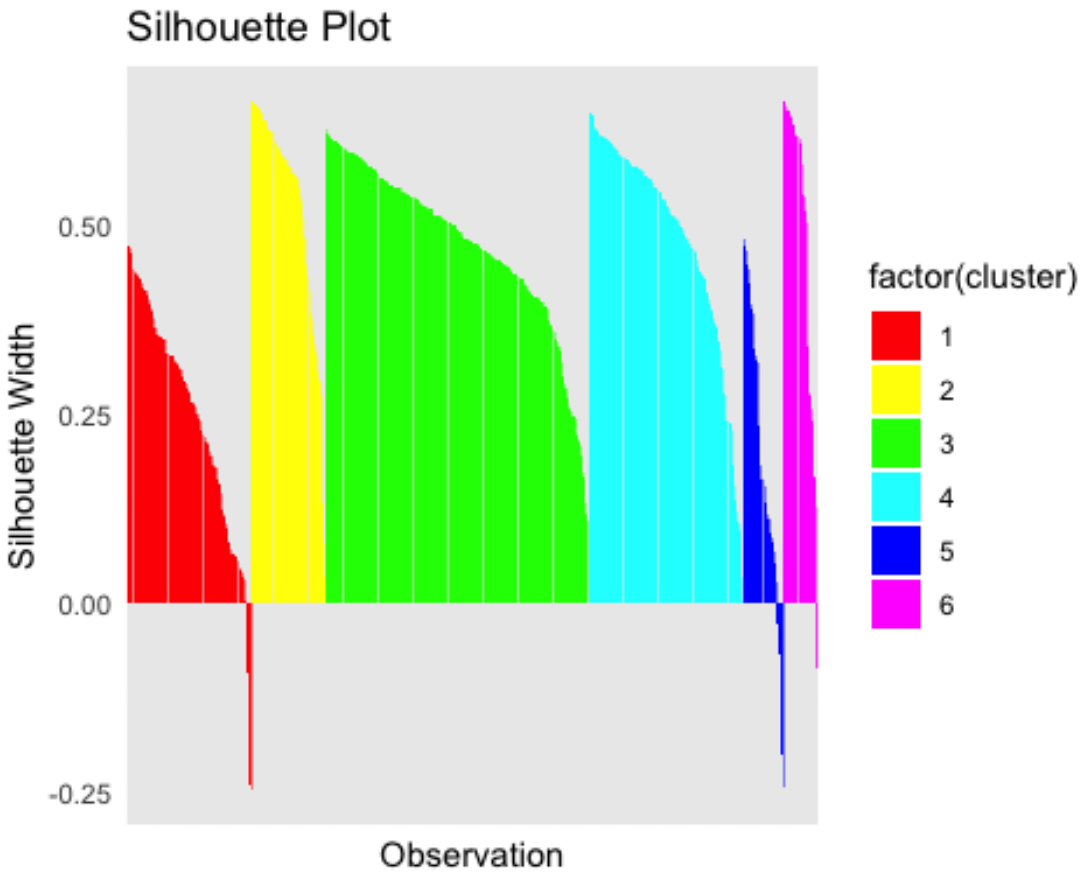
```
## List of 10
##  $ medoids    : num [1:6, 1:7] 1.77 1.99 1.58 1.94 2.02 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
```

```
##   .. ..$ : 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"
```

```
pam.out$clusinfo[, "size"]
```

```
## [1] 103  61 219 127  34  28
```

```
# 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())
```

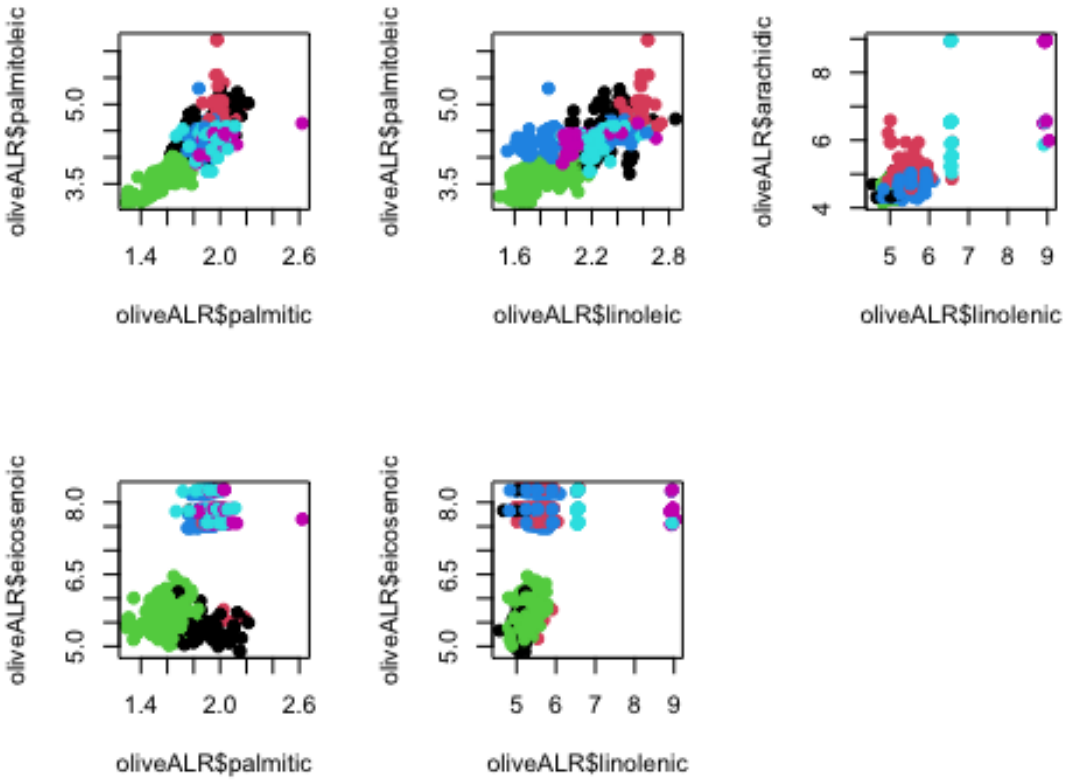


Si nota che - Cluster 3 ha valori tutti molto distanti dagli altri gruppi, in quanto l'indice cala drasticamente - Cluster 5 e 6 contengono meno elementi rispetto agli altri - Cluster 1 e 5 contengono valori inseriti nel cluster incorretto

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)
```

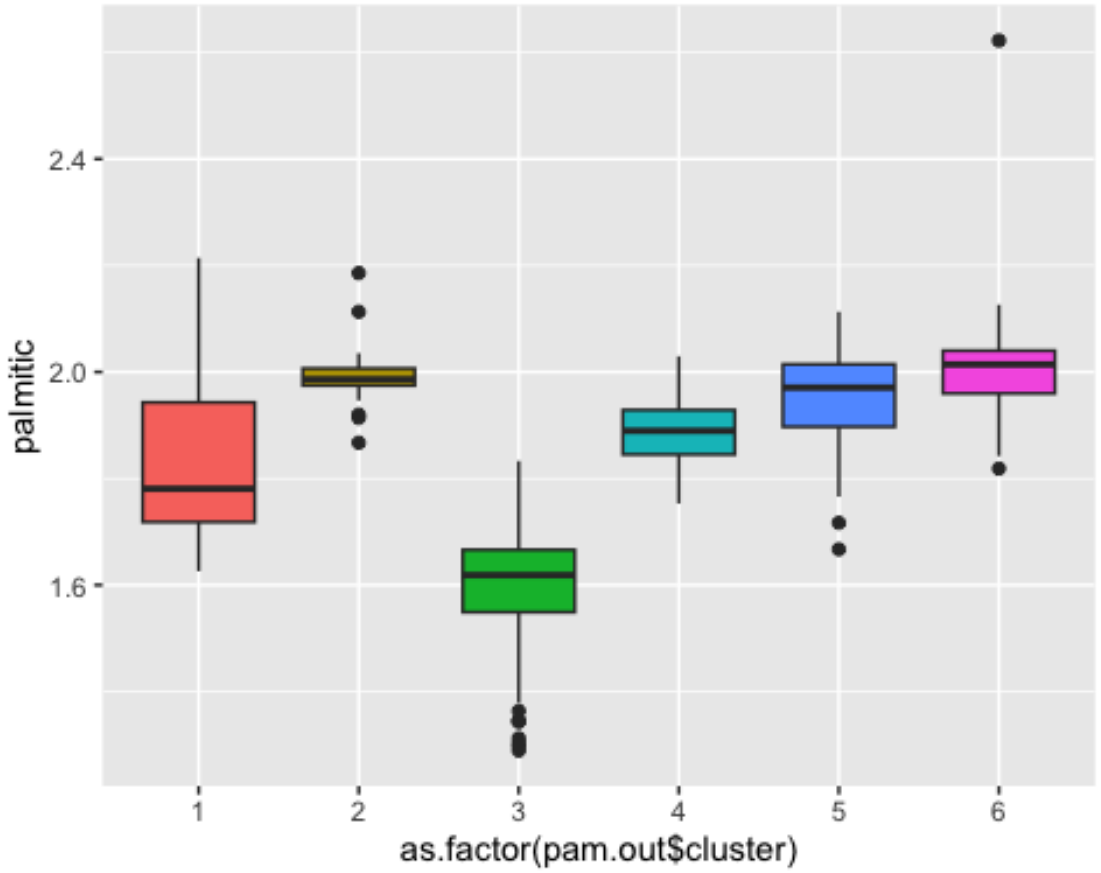
```
# linolenic eicosenoic
plot(oliveALR$linolenic, oliveALR$eicosenoic, col = pam.out$cluster, pch =
19)

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 palmitic

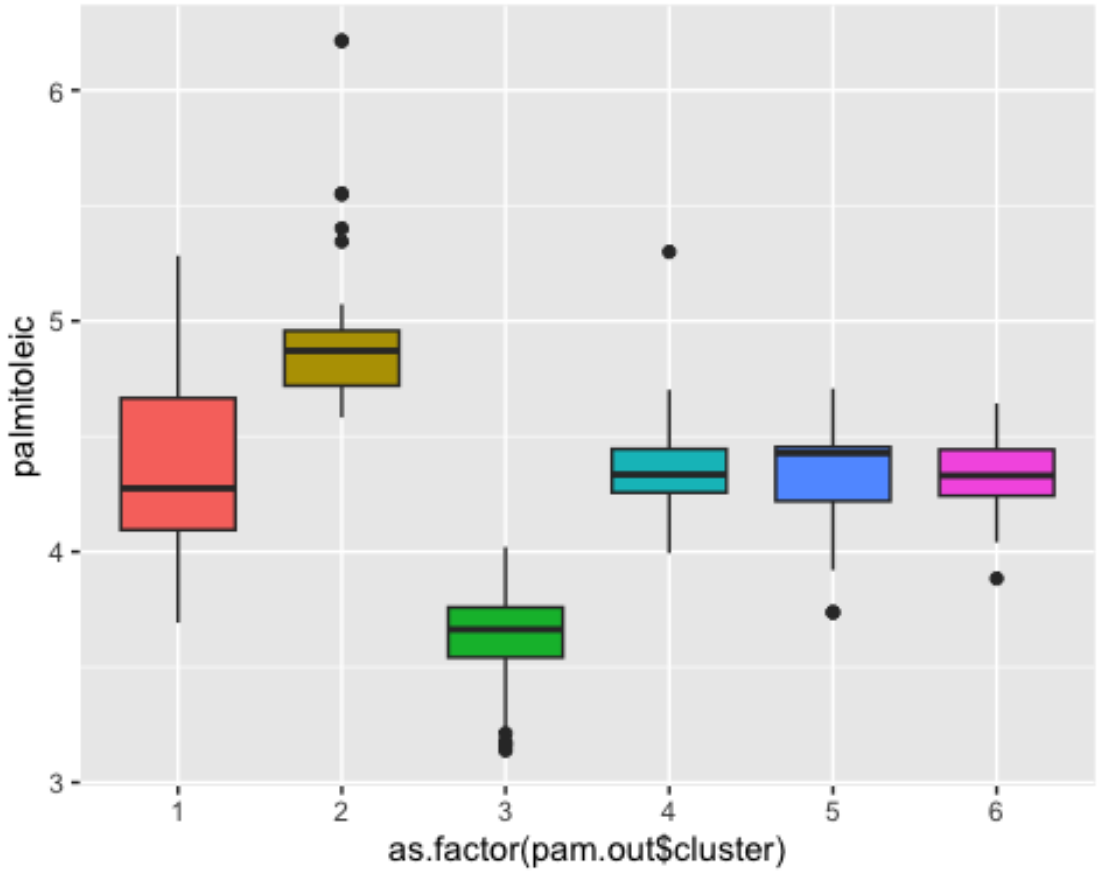
```
ggplot(oliveALR, aes(x = as.factor(pam.out$cluster), y = palmitic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Quasi tutti i valori sono compresi tra 1.4 e 2.4, vi sono alcuni valori inferiori a 1.4 e si trovano tutti nel cluster 3, la maggior parte di questi valori sono outlier, Il cluster 2 ha la devianza interna più bassa.

Variabile palmitoleic

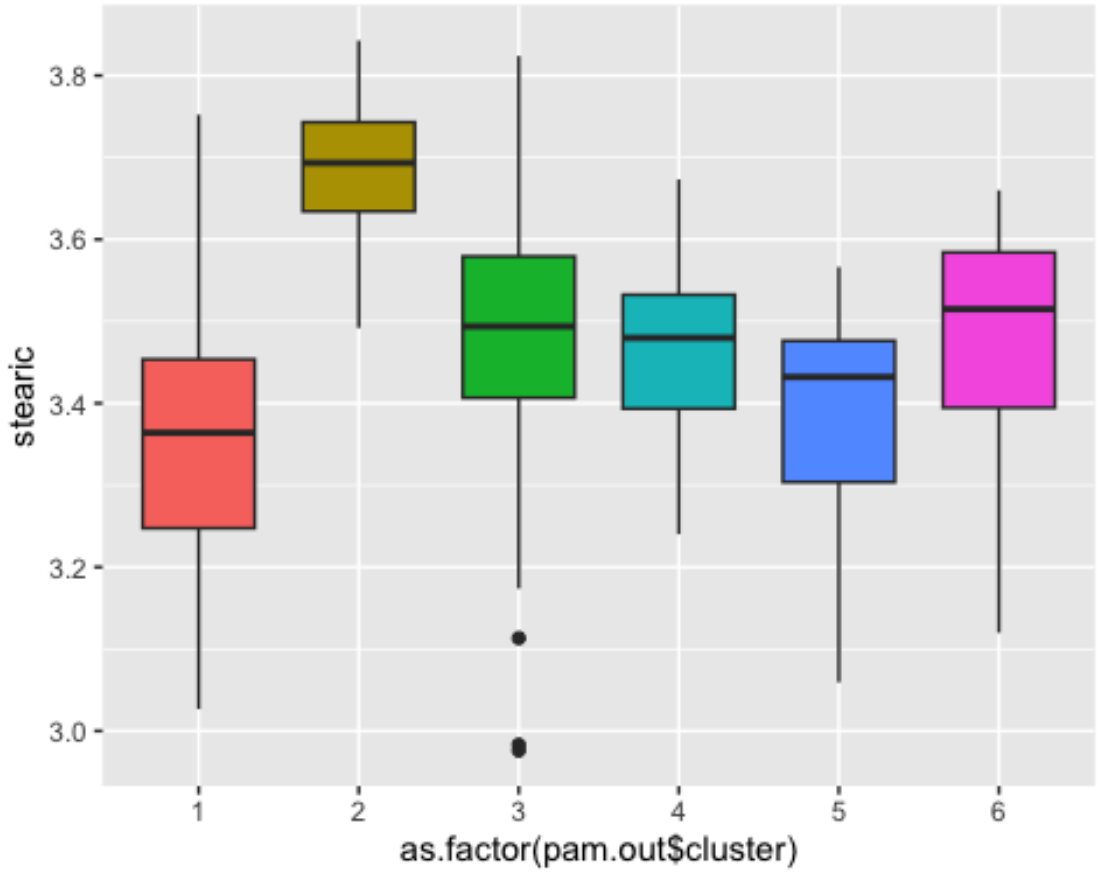
```
ggplot(oliveALR, aes(x = as.factor(pam.out$cluster), y = palmitoleic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



La mediana dei cluster 1,4,5 e 6 è compresa tra 4 e 4.5, cluster 2 e 3 si scostano da questo range.

Variabile stearic

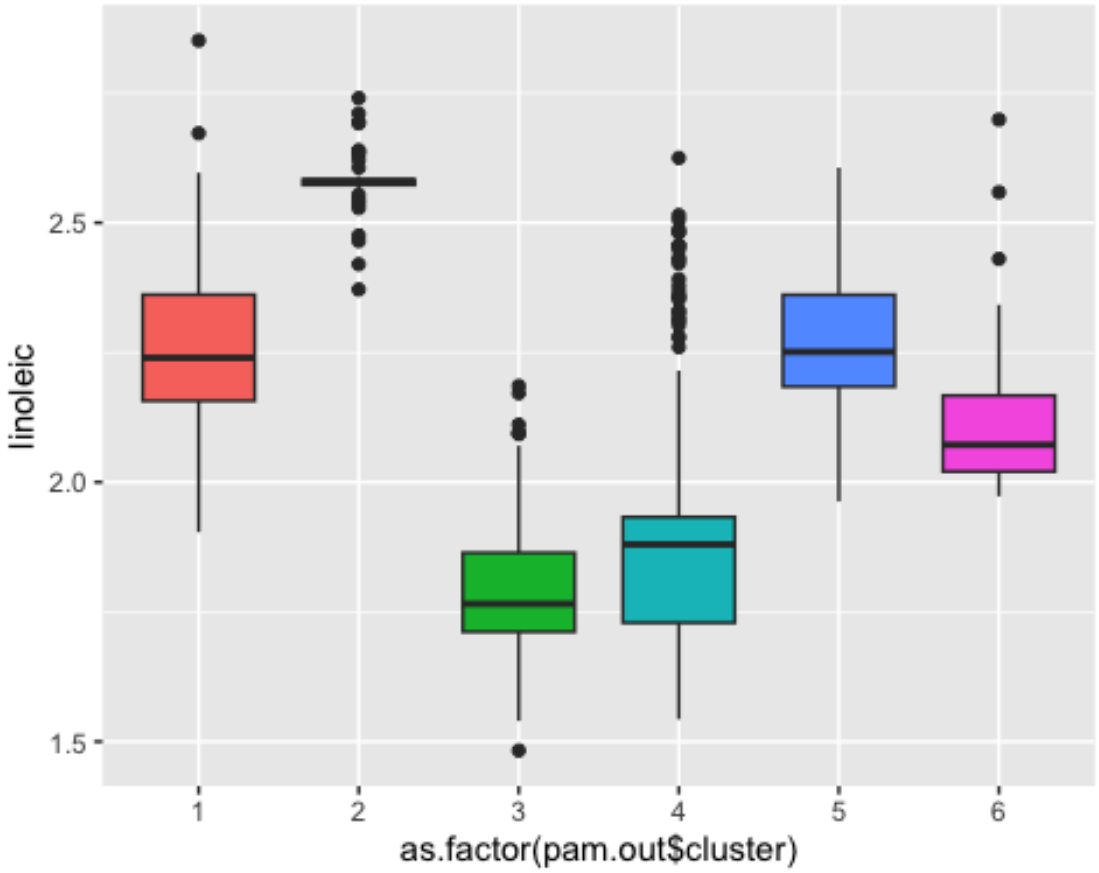
```
ggplot(oliveALR, aes(x = as.factor(pam.out$cluster), y = stearic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Cluster 3,4,5,6 hanno una mediana vicina. I cluster 1 e 2 si discostano significativamente, il primo verso il basso, l'altro verso l'alto.

Variabile `linoleic`

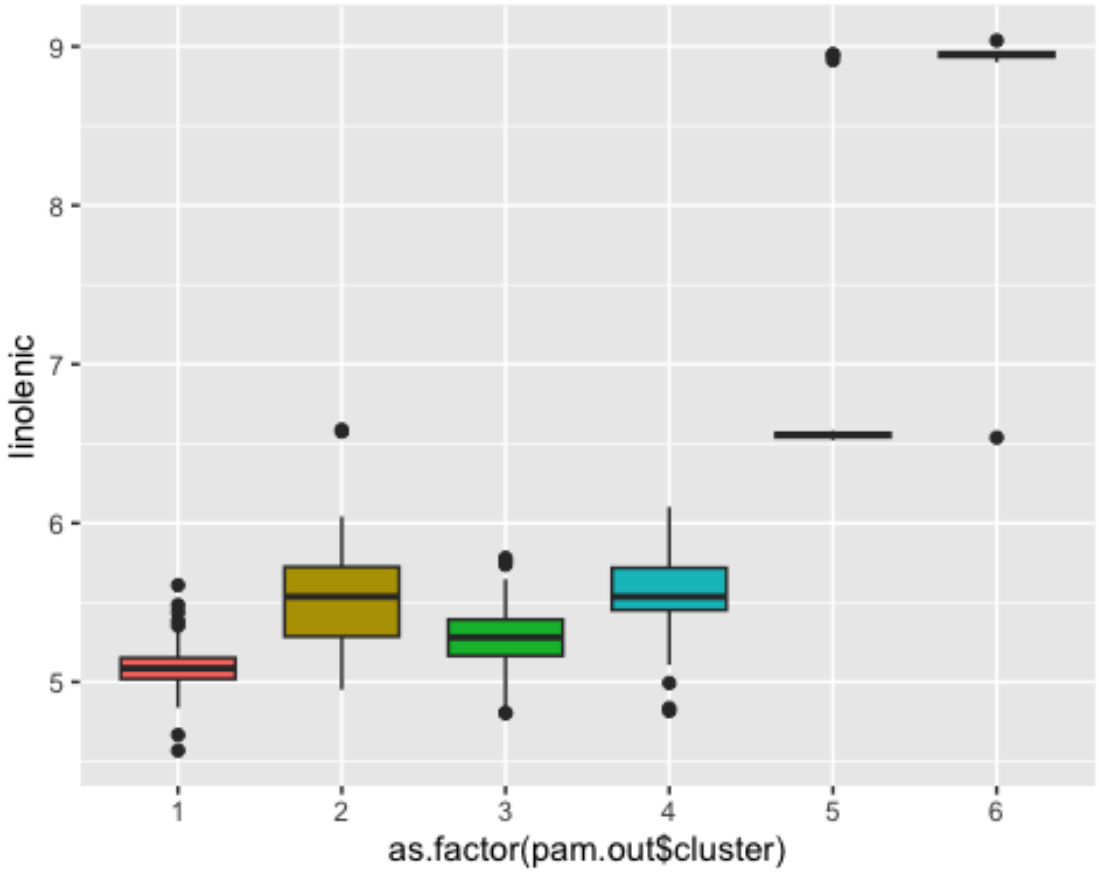
```
ggplot(oliveALR, aes(x = as.factor(pam.out$cluster), y = linoleic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



Devianza fra i gruppi alta, cluster 3 e 4 hanno mediane vicine, il cluster 4 presenta un discreto numero di valori outlier nel range 2.25-2.5. I cluster 1, 5 e 6 hanno la mediana nel range 2-2.25.

Variabile Linolenic

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

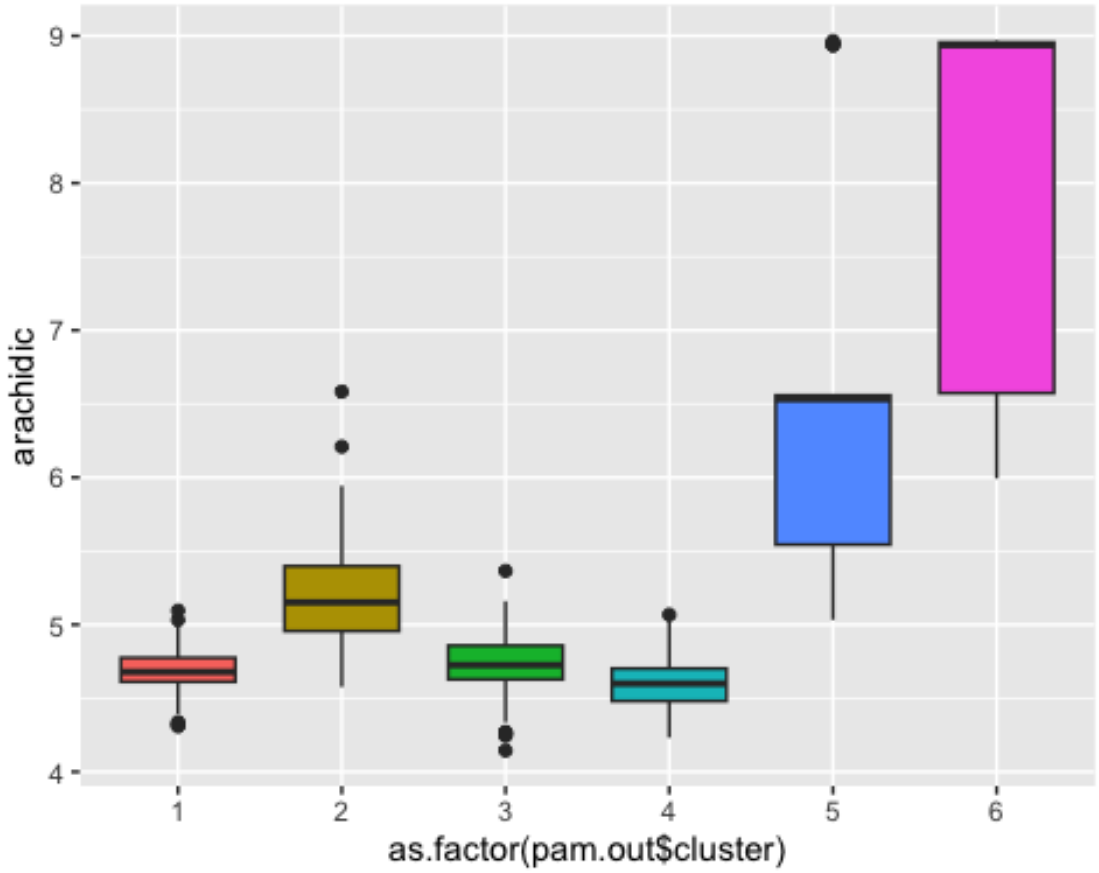



```
median(oliveALR$linolenic[pam.out$cluster == 5])

## [1] 6.554451
```

Primi 4 cluster hanno mediana compresa tra 5 e 6. I cluster 5 e 6 si discostano significativamente, nel cluster 5 si osservano i valori concentrati intorno a 6.5 e nel cluster 6 attorno a 9. Vi sono valori outlier in tutti i cluster.

Variabile arachidic

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

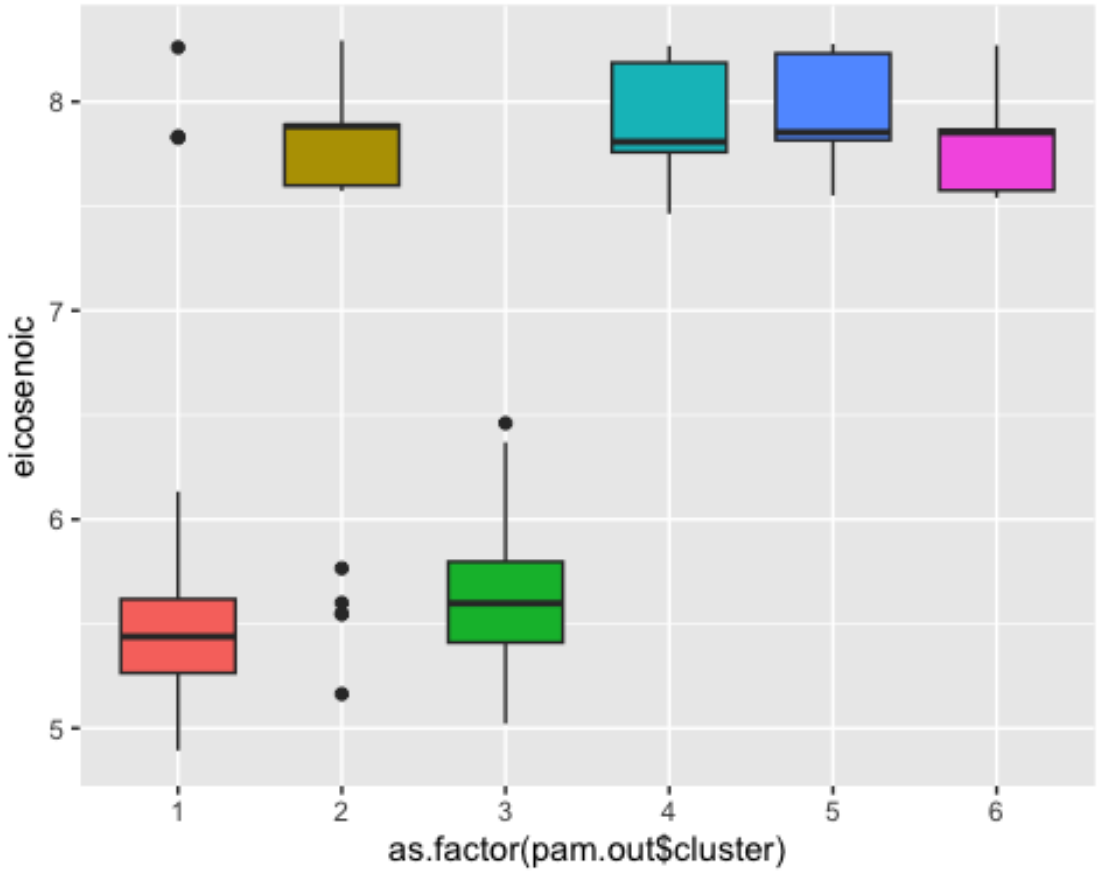



```
median(oliveALR$arachidic[pam.out$cluster == 5])

## [1] 6.531508
```

Primi 4 cluster hanno mediana compresa tra 5 e 6. I cluster 5 e 6 si discostano significativamente, nel cluster 5 si osservano i valori concentrati intorno a 6.5 e nel cluster 6 attorno a 9. Vi sono valori outlier in tutti i cluster. Gli oli presenti in questo cluster hanno percentuali di acido arachidico prossime allo zero.

Variabile eicosenoic

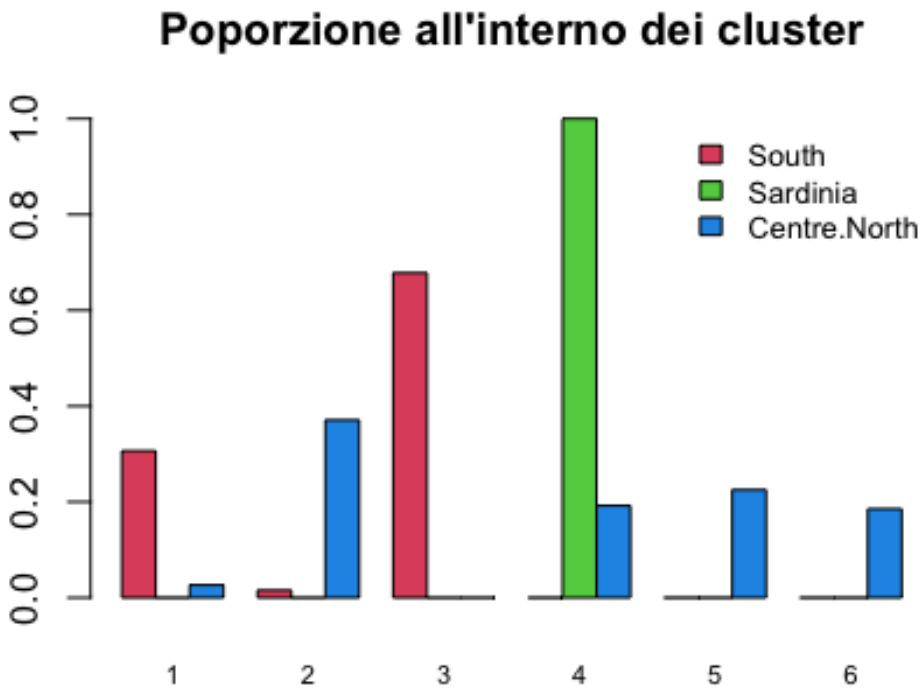
```
ggplot(oliveALR, aes(x = as.factor(pam.out$cluster), y = eicosenoic, fill =
as.factor(pam.out$cluster))) + geom_boxplot(width=0.7) + guides(fill = FALSE)
```



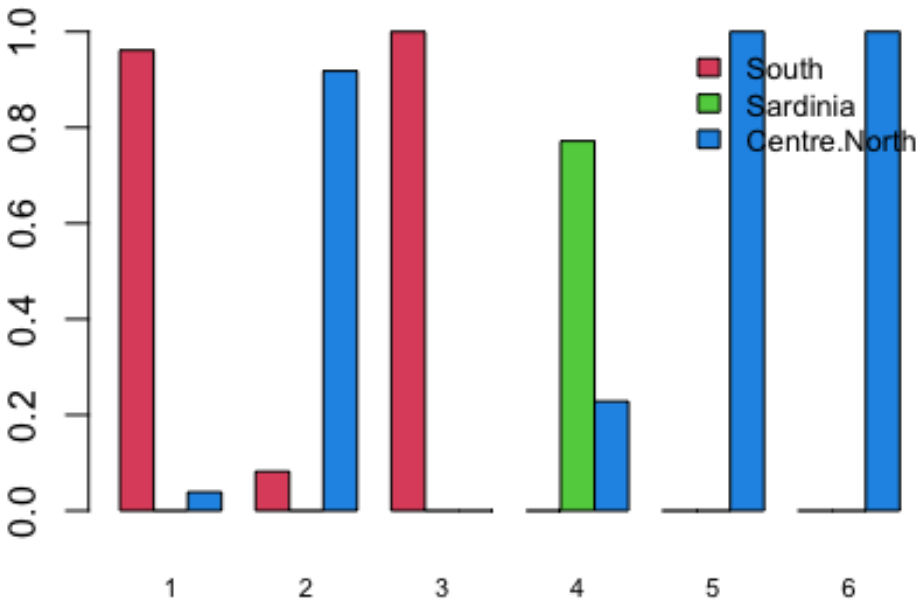
Si osservano chiaramente due mode, il cluster 1 e il cluster 3 si trovano nel range 5-6 mentre gli altri cluster si trovano nel range 7.5 - 8.5. Nel range 6.5-7.5 non vi è alcun valore. Gli oli nei cluster più alti hanno percentuali di acido ecosenoico prossime allo zero.

Variabile macro.area

```
round(prop.table(table(oliveALR$macro.area, pam.out$cluster),1), 3)

##
##                     1     2     3     4     5     6
##   South         0.307 0.015 0.678 0.000 0.000 0.000
##   Sardinia      0.000 0.000 0.000 1.000 0.000 0.000
##   Centre.North 0.026 0.371 0.000 0.192 0.225 0.185

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")
```



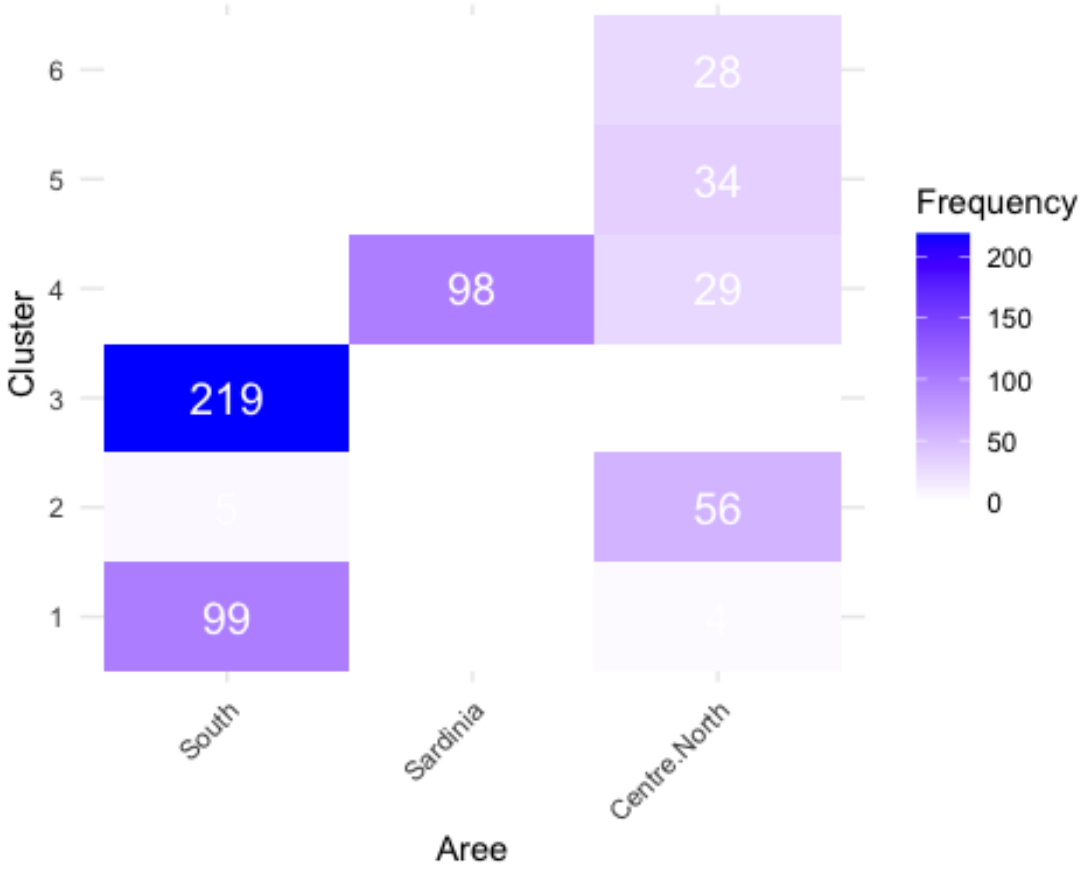
Dai grafici si nota un netto miglioramento nel raggruppamento dei cluster di pam ALR ripetto agli altri algoritmi, se confrontati con i gruppi creati dalle macro areee. Gli oli della sardegna vengono inseriti interamente nel cluster 4. Gli oli del sud vengono inseriti per il 30% nel cluster 1 e 67% nel 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(Aree = oliveALR$macro.area, Cluster =
pam.out$cluster)

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

##               Cluster
## Aree             1   2   3   4   5   6
##   South         99   5 219   0   0   0
##   Sardinia       0   0   0  98   0   0
##   Centre.North   4  56   0  29  34  28

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



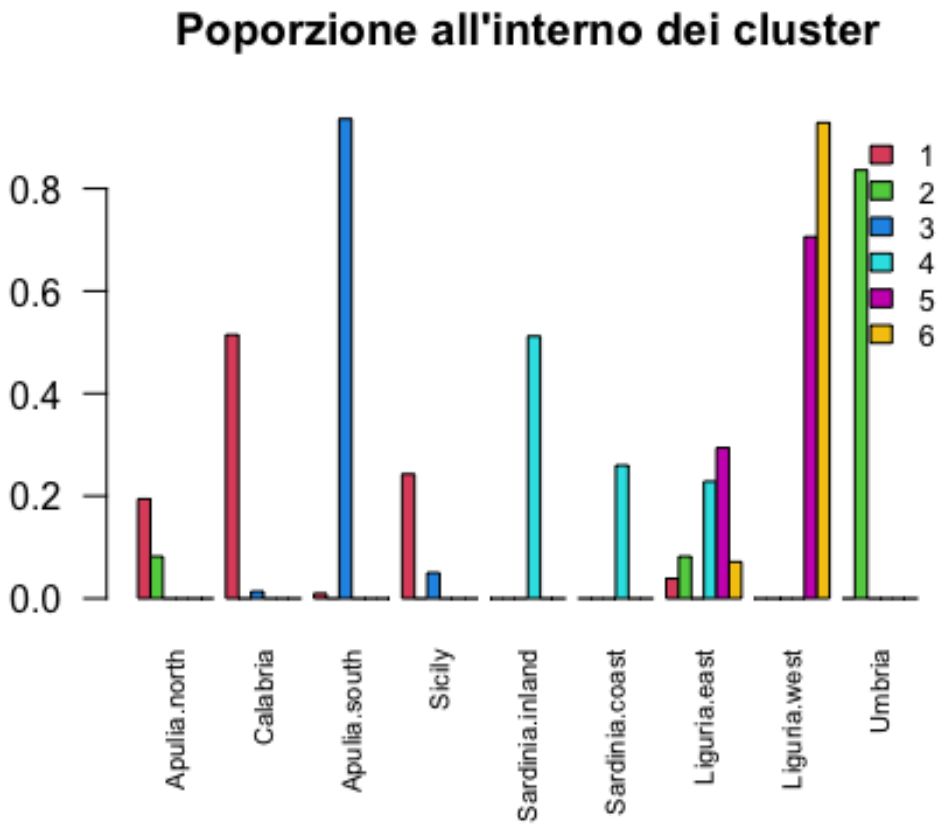
Variabile region

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

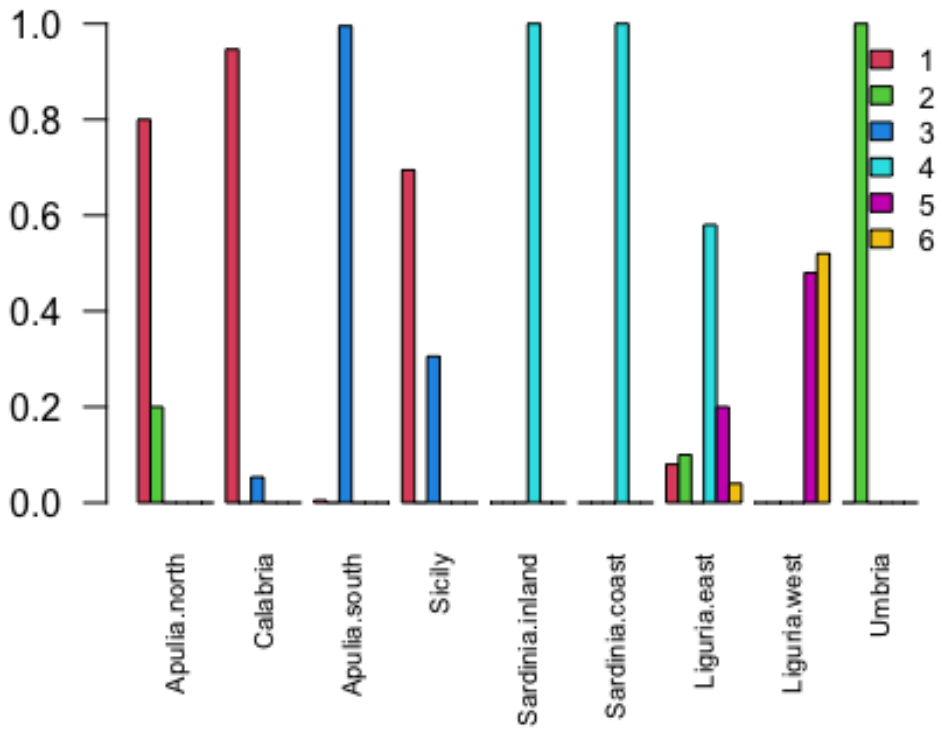
```
##
##     Apulia.north    Calabria Apulia.south      Sicily Sardinia.inland
##   1  0.194174757 0.514563107  0.009708738 0.242718447     0.000000000
##   2  0.081967213 0.000000000  0.000000000 0.000000000     0.000000000
##   3  0.000000000 0.013698630  0.936073059 0.050228311     0.000000000
##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

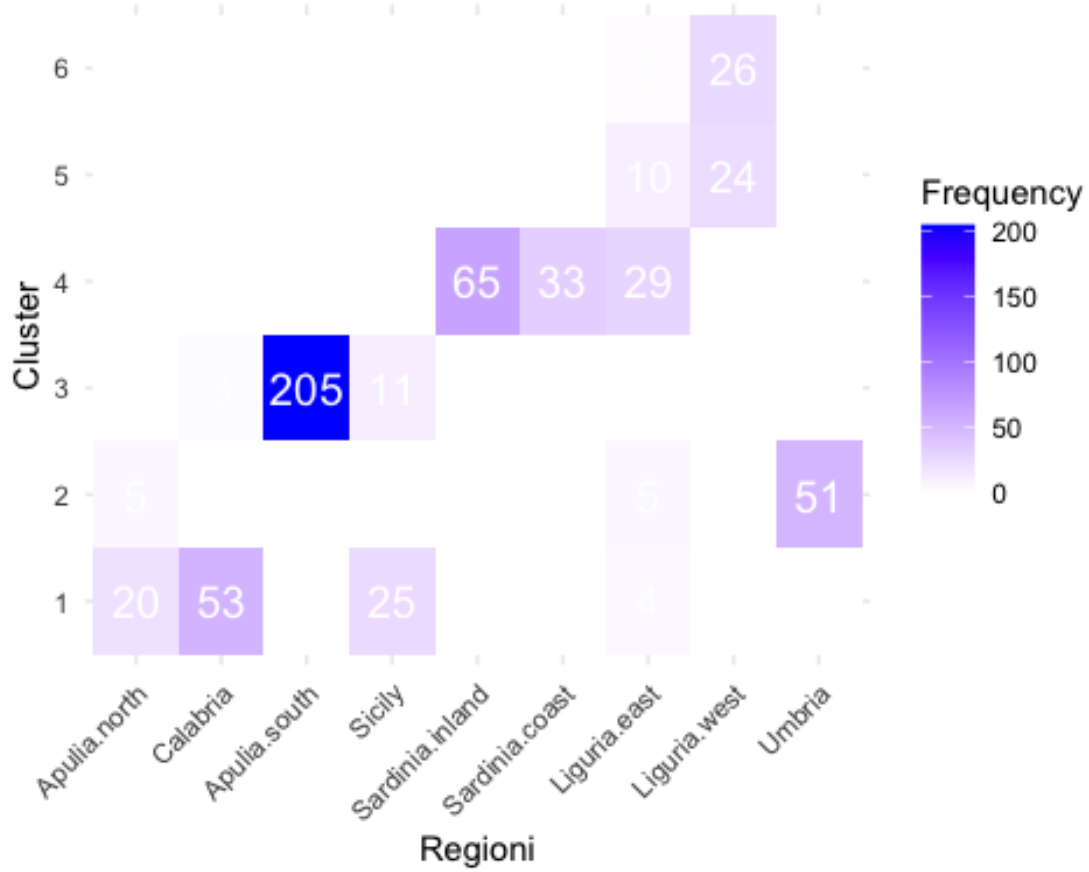
Guardando i cluster invece possiamo affermare che il cluster 1 contiene oli della puglia nord, calabria e sicilia il cluster 2 contiene oli della dell'umbria e in parte della puglia nord il cluster 3 contiene oli della puglia sud, e in parte della sicilia il cluster 4 contiene oli della sardegna e liguria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

Confusion Matrix:

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

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

##                  Cluster
## Regioni             1   2   3   4   5   6
##   Apulia.north     20   5   0   0   0   0
##   Calabria         53   0   3   0   0   0
##   Apulia.south      1   0 205   0   0   0
```

```
##   Sicily           25   0  11   0   0   0
##   Sardinia.inland   0   0   0  65   0   0
##   Sardinia.coast    0   0   0  33   0   0
##   Liguria.east      4   5   0  29  10   2
##   Liguria.west      0   0   0   0  24  26
##   Umbria            0  51   0   0   0   0
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Regioni, y =
Cluster, 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))
```



Adjusted Rand Index - PAM

```
ari_pam_ln <- adj.rand.index(oliveALR$macro.area, pam.out$cluster)
ari_pam_ln
```

```
## [1] 0.5285497
```

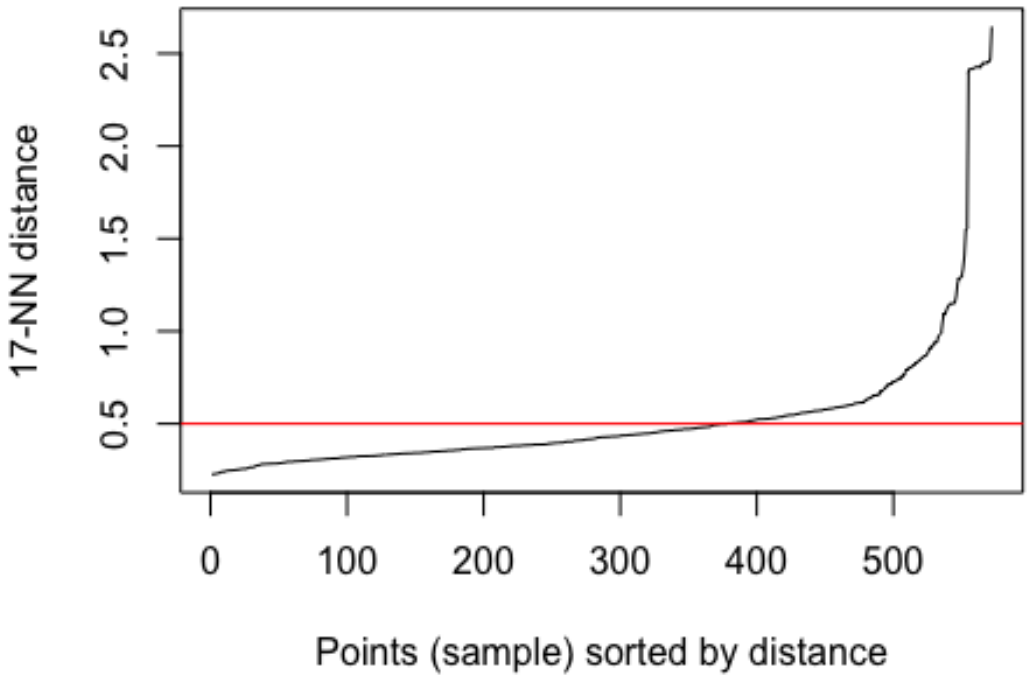
Mappa dei cluster sulla cartina Italiana

```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=pam.out$cluster+1, pch=19, cex=0.3)
```

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

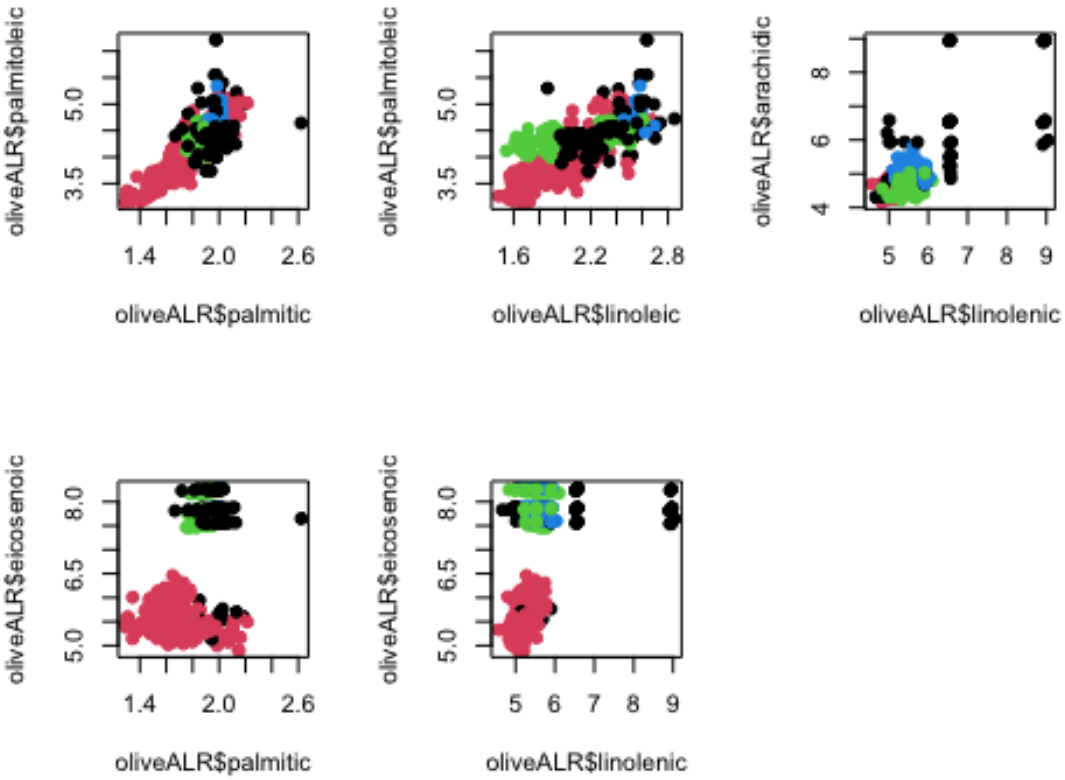
```
set.seed(17)

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".

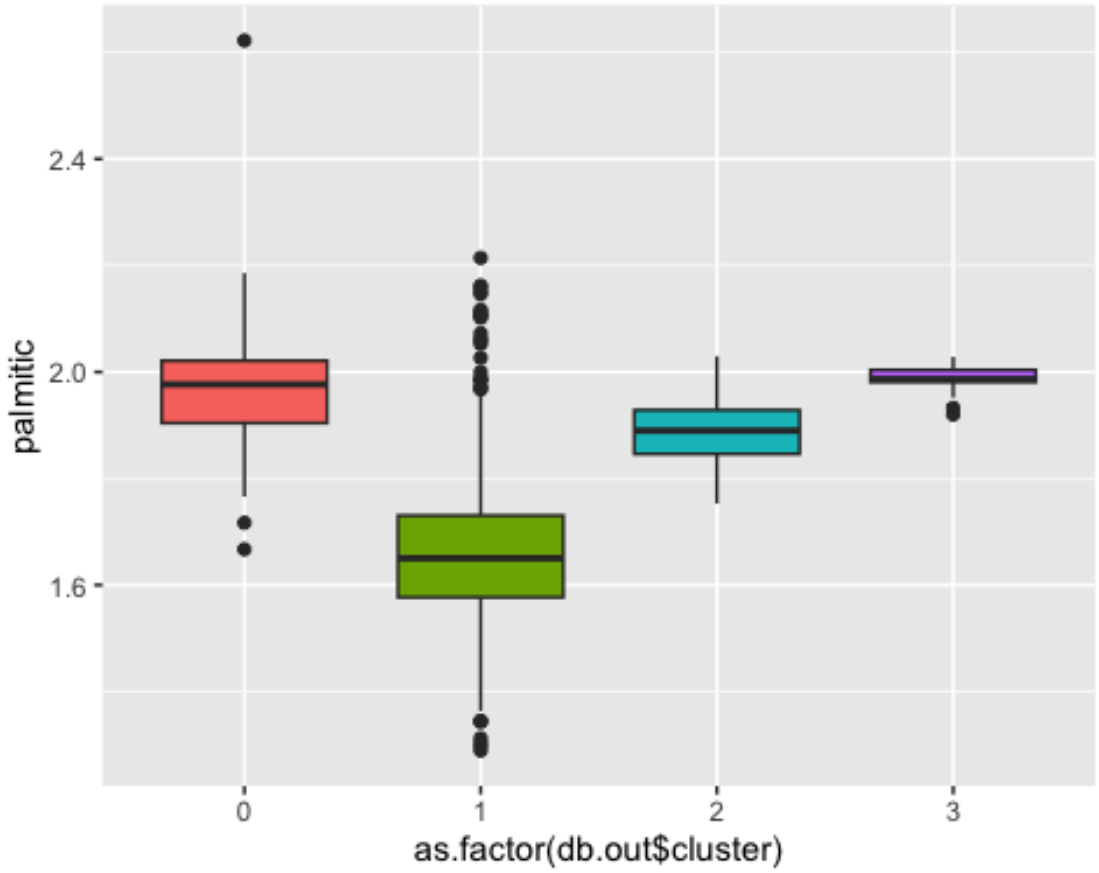
Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

```
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 palmitic

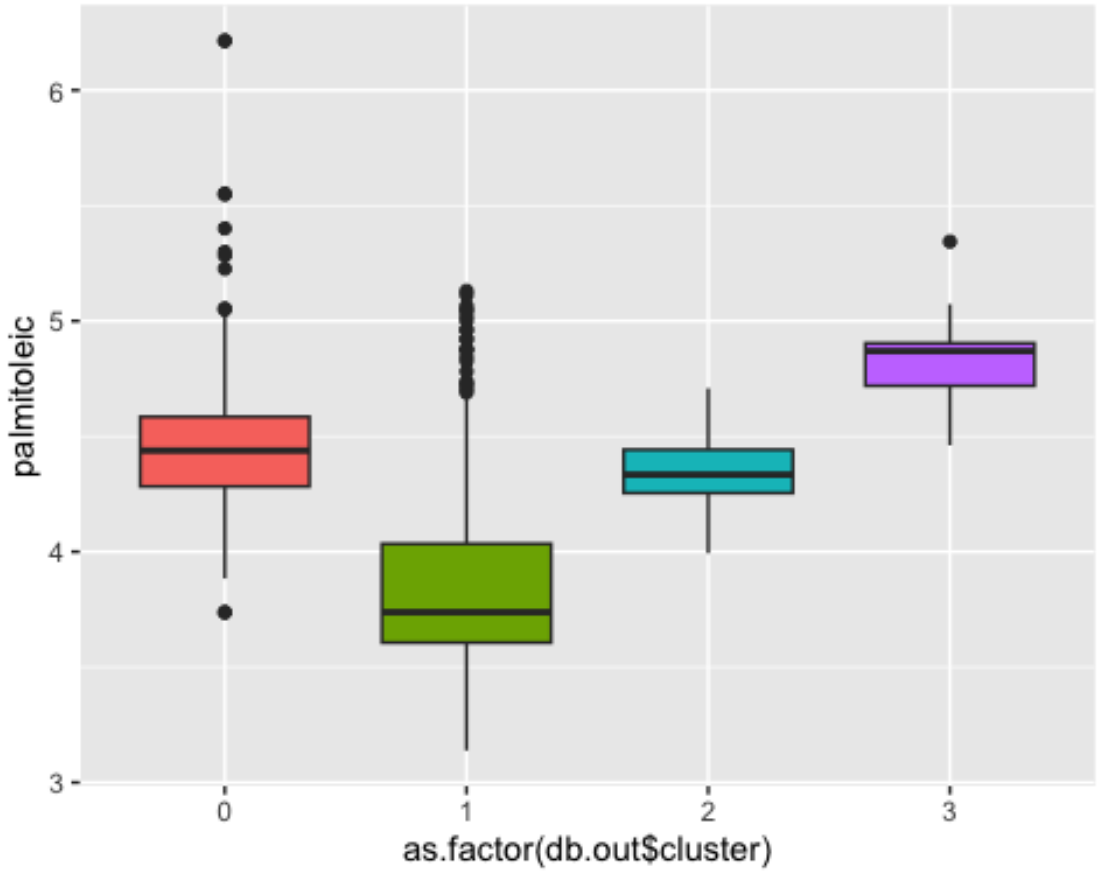
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = palmitic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



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 1 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 3 è quello con in media percentuali più basse di acido palmitico, mentre il cluster 1 tende ad avere valori maggiori per questo acido. Ricordiamo che i valori più vicini allo 0 dopo la trasformazione saranno quelli con percentuali di acido maggiori.

Variabile palmitoleic

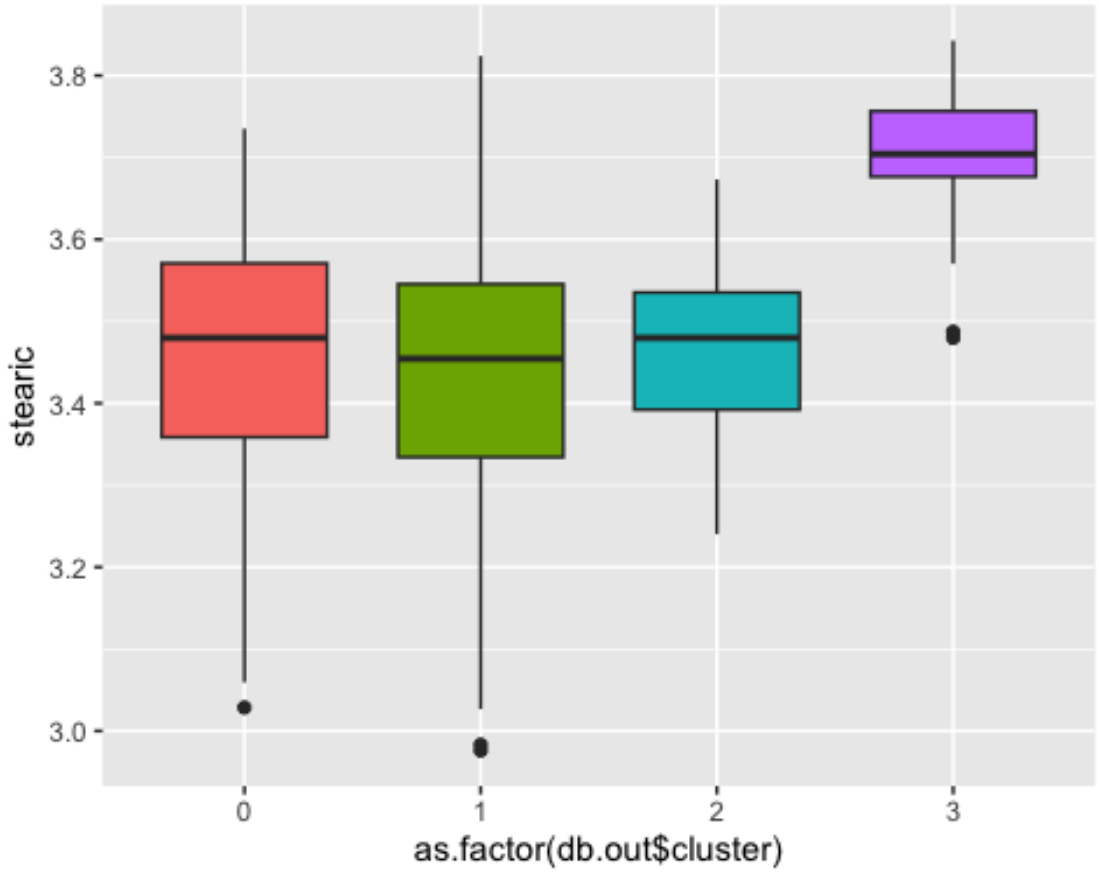
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = palmitoleic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



L'acido palmitileico rimane molto simile a quello palmitico sempre per via dell'alta correlazione tra le due variabili. Come nel caso del palmitico, il cluster 3 ha le percentuali di acido minori e il cluster 1 quelle maggiori.

Variabile stearic

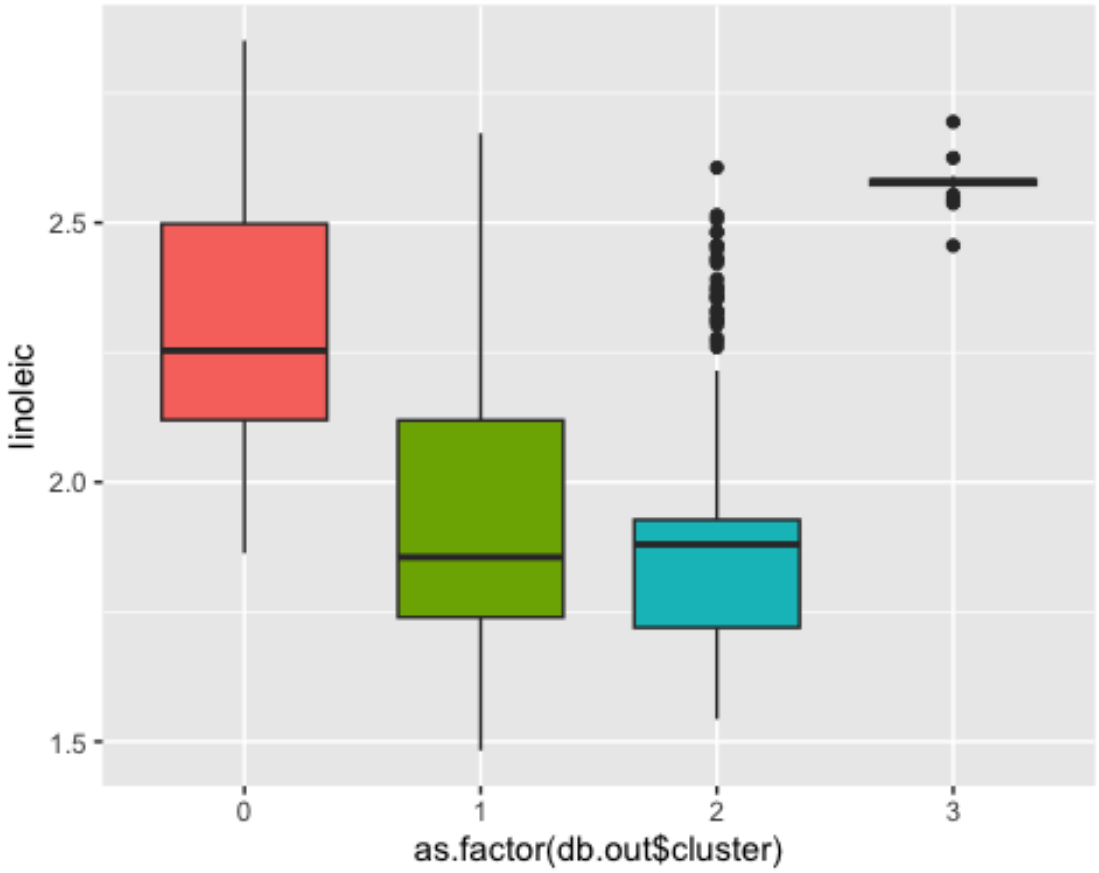
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = stearic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Notiamo che l'acido Stearico ha mediane molto simili per i vari cluster e punti di "noise". L'unico cluster che si discosta dagli altri è il 3 che comprende oli con percentuali di acido stearico minori.

Variabile Linoleic

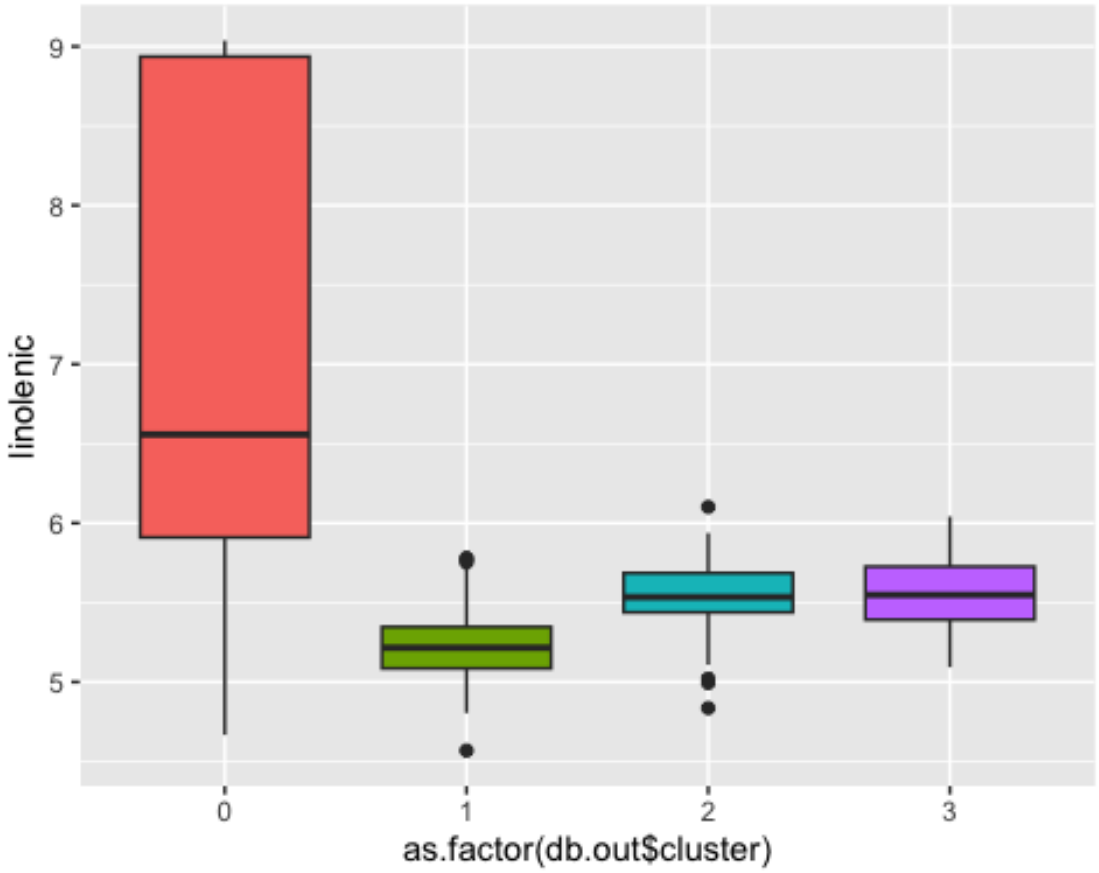
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = linoleic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



In questo caso vediamo che sia il cluster 1 che il cluster 2 contengono oli con un range di percentuali di acido Linoleico decisamente maggiore rispetto al cluster 3. Quest'ultimo è infatti molto concentrato tra 2.5 e 3.0, il che è parzialmente dovuto anche dal fatto che è il cluster meno numeroso. Come nei precedenti 3 esempi, il cluster 3 è quello con percentuali di acido minori.

Variabile linolenic

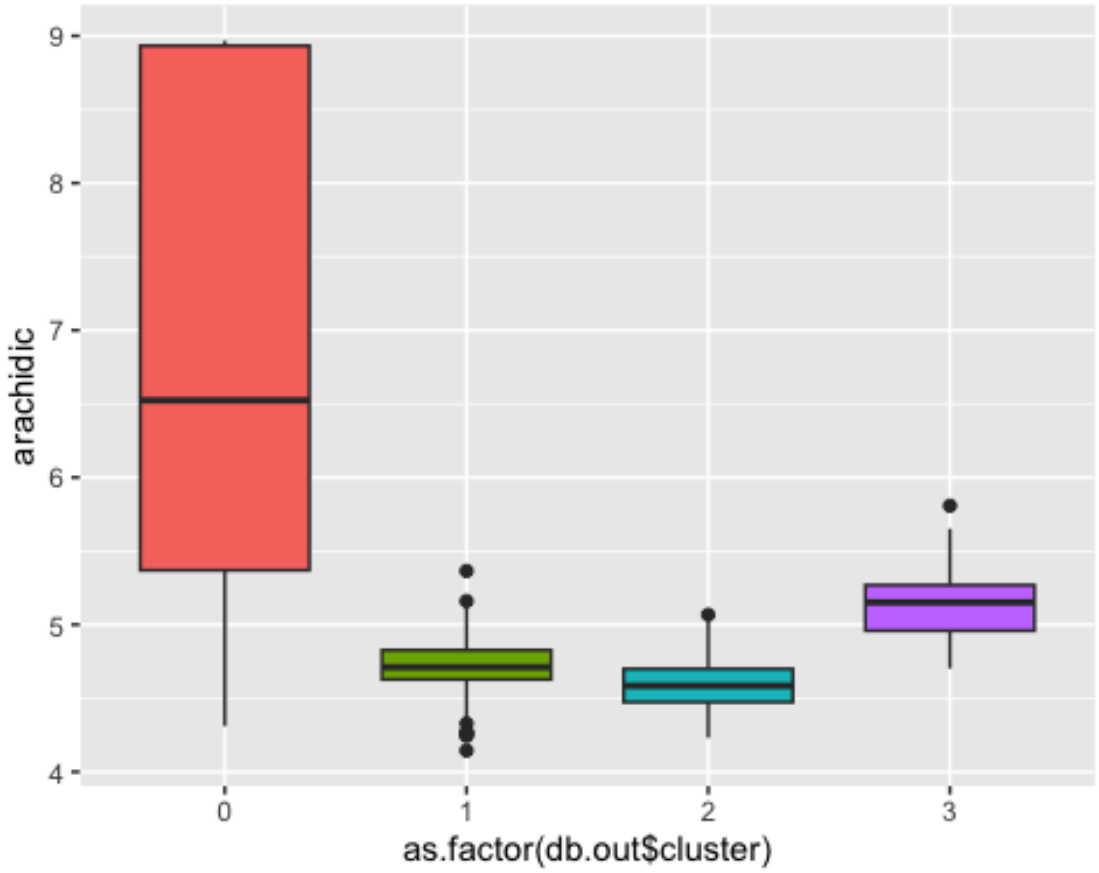
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = linolenic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Guardando l'olio linolenico negli oli dei diversi cluster notiamo che i tre cluster non differiscono molto. Invece, l'insieme dei punti di "noise" ha un range di valori decisamente maggiore. Questo arriva fino a valori vicino al "9", il che indica percentuali estremamente basse di acido linolenico all'interno degli oli considerati come "noise".

Variabile arachidic

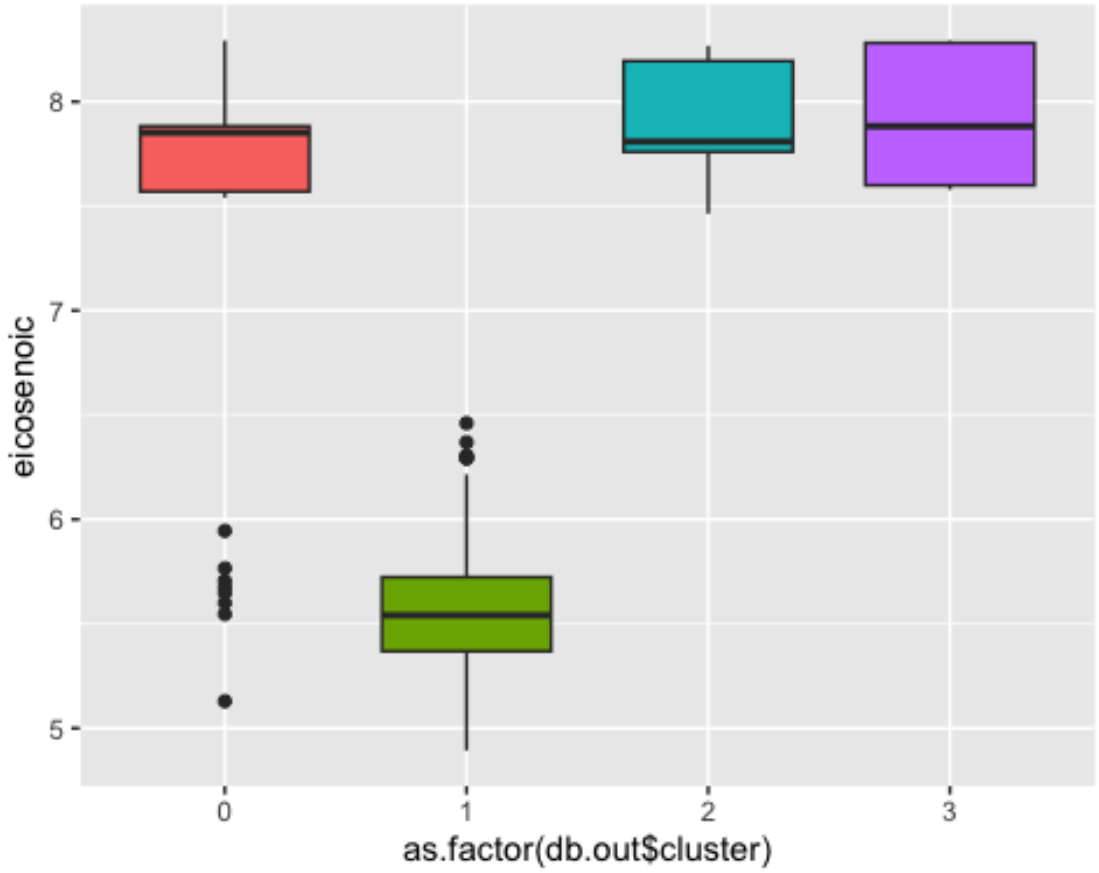
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = arachidic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Per l'acido arachidico notiamo una distribuzione molto simile a quella dell'acido linolenico. In questo caso però vediamo che il cluster 3 si scosta maggiormente dagli altri due.

Variabile eicosenoic

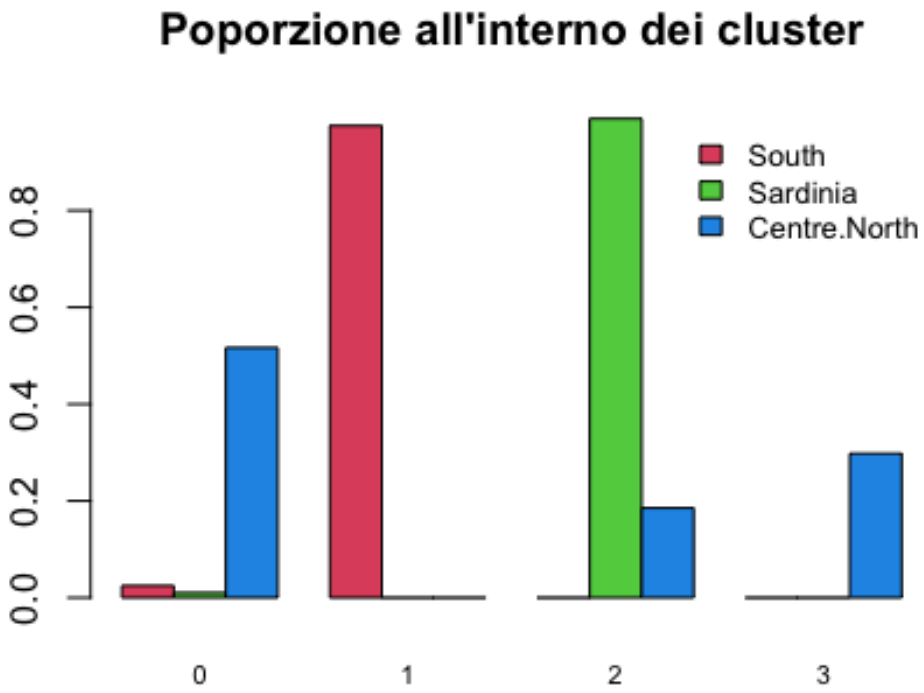
```
ggplot(oliveALR, aes(x = as.factor(db.out$cluster), y = eicosenoic, fill =
as.factor(db.out$cluster+1))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



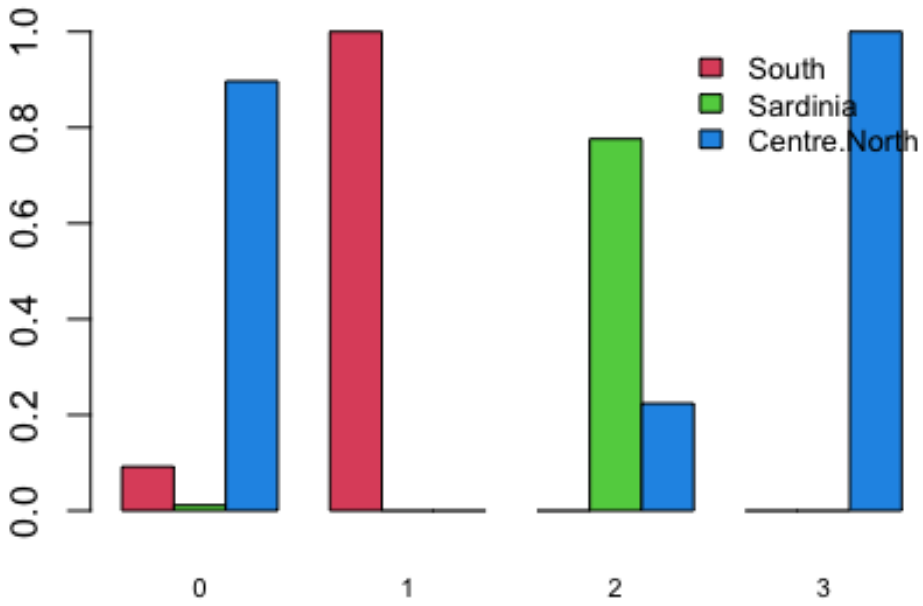
La distribuzione in cluster della quantità di acido eicosenoico negli oli è molto particolare. Sia i cluster 2 e 3 che l'insieme di punti di "noise" sembrano contenere valori estremamente bassi di acido eicosenoico, con valori che trasformati si aggirano intorno all'8. Ricordiamo che sono presenti diversi oli con percentuali di questo acido anche inferiori alla sensibilità delle macchine usate per calcolarla. L'unico cluster che non contiene oli con percentuali così basse è l'1. Questo infatti raccoglie la gran parte dei valori lontani dallo 0.

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:

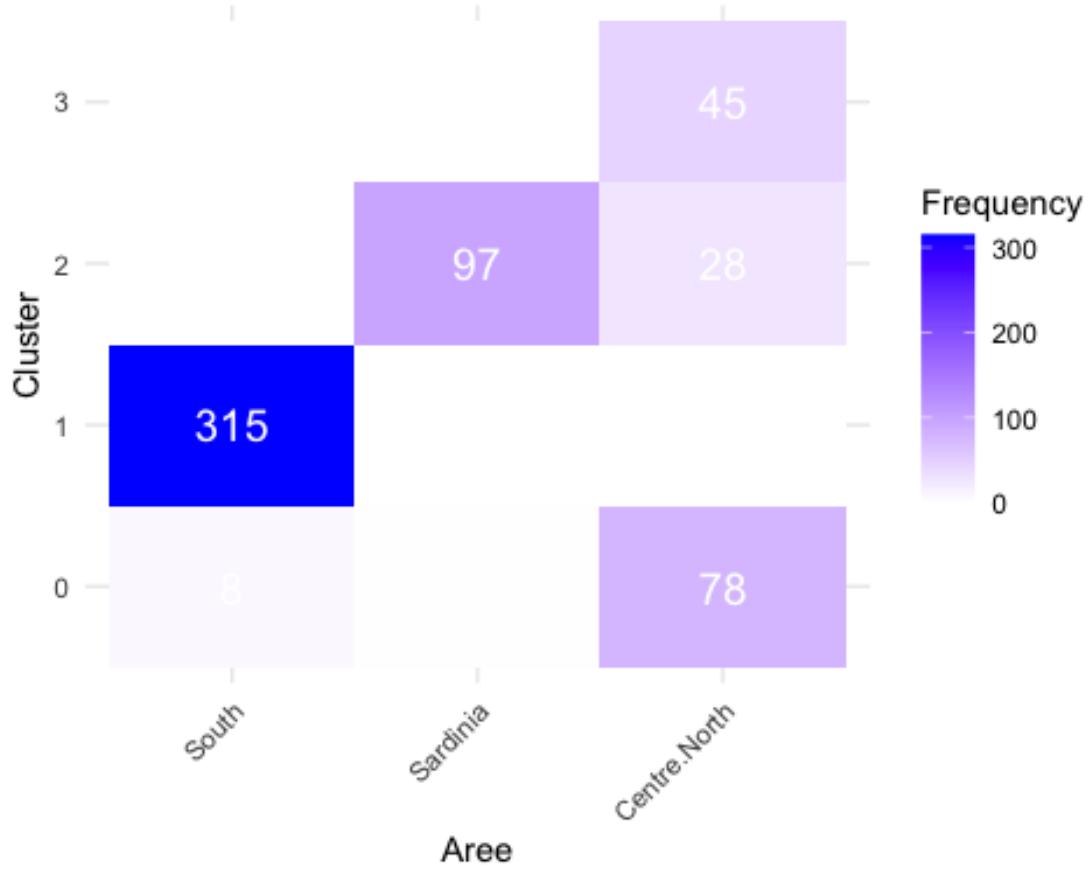
Con la trasformazione logaritmica notiamo che gli oli del Sud sono meglio divisi dagli altri. Infatti questi solo quasi esclusivamente appartenenti al cluster 1. Mentre in precedenza gli oli che venivano considerati come punti di "noise" erano principalmente provenienti dal Sud, adesso sono principalmente del Centro Nord (anche se questo è dovuto in gran parte alla Liguria).

Confusion Matrix:

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

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

##               Cluster
## Aree             0   1   2   3
```

```
##   South          8 315   0   0
##   Sardinia       1   0  97   0
##   Centre.North  78   0  28  45
```

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



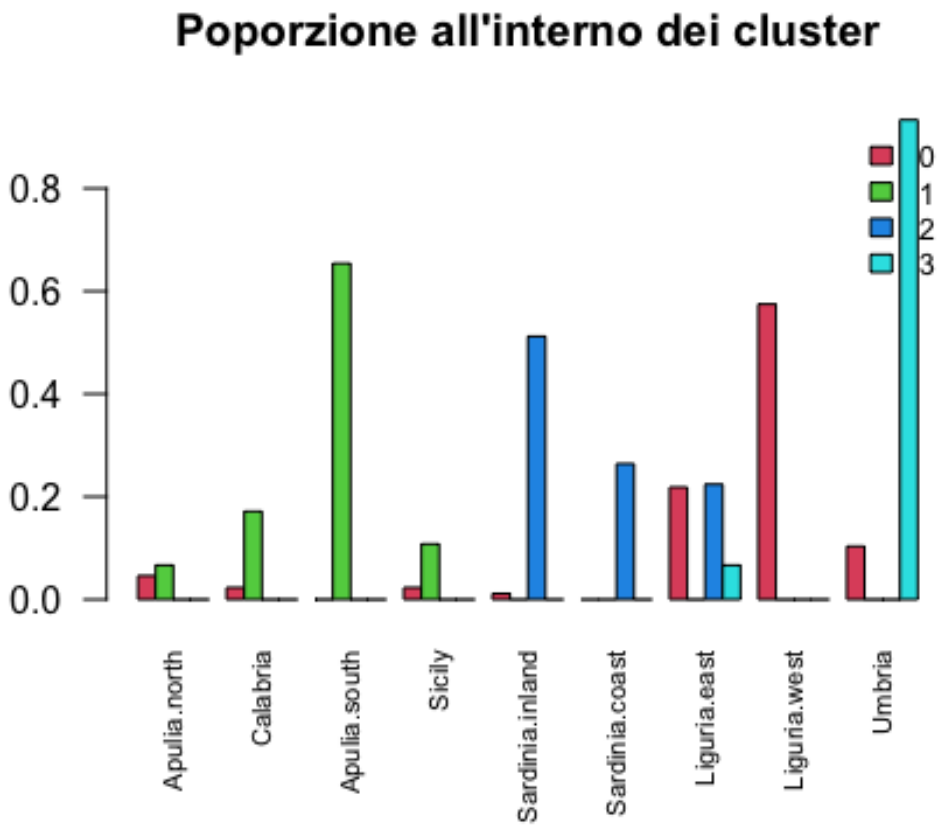
Variabile region

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

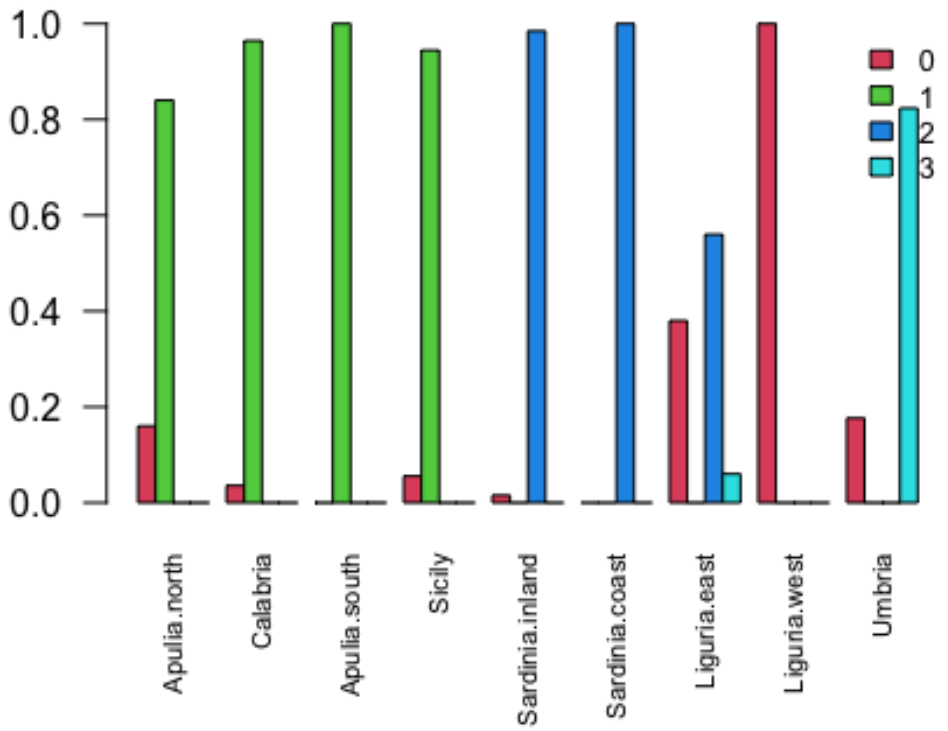
```
##
##     Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   0   0.04597701 0.02298851   0.00000000 0.02298851      0.01149425
##   1   0.06666667 0.17142857   0.65396825 0.10793651      0.00000000
##   2   0.00000000 0.00000000   0.00000000 0.00000000      0.51200000
##   3   0.00000000 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:

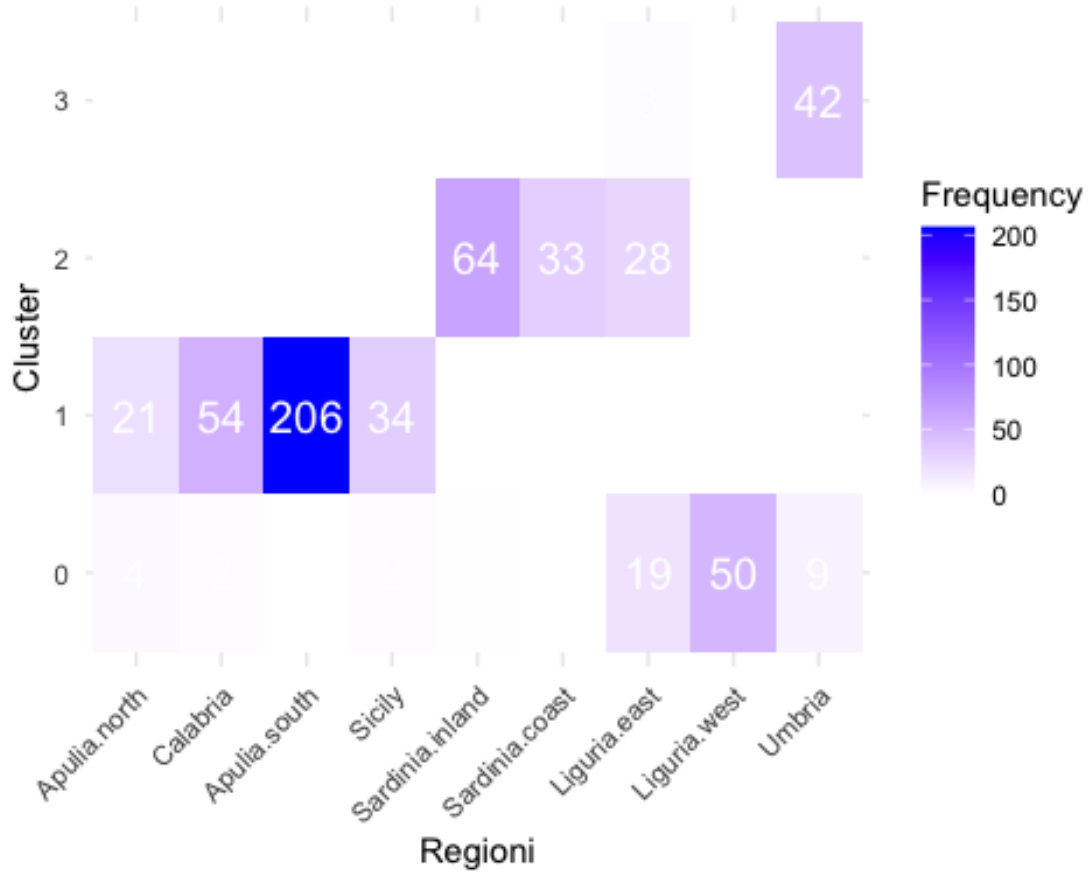
La prima cosa che salta all’occhio è la differenza nella distribuzione dei punti di “noise”: mentre nel caso senza trasformazione ALR questi erano presenti in più regioni in percentuali più basse, adesso questo tipo di punti sono quasi esclusivamente presenti in Liguria. Notiamo anche che Liguria dell’Est e Umbria non sono più parte dello stesso cluster delle regioni del sud. E infatti il cluster 1 rappresenta in modo ottimo le regioni del Sud.

Confusion Matrix:

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

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

##                  Cluster
## Regioni             0   1   2   3
```

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

```r
ggplot(data = as.data.frame(as.table(confusion_matrix)), aes(x = Regioni, y =
Cluster, 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))
```



Adjusted Rand Index - DB SCAN

```r
ari_db_ln <- adj.rand.index(oliveALR$macro.area, db.out$cluster)
ari_db_ln
```

```
## [1] 0.8341706
```

Mappa dei cluster sulla cartina Italiana

```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=db.out$cluster+1, pch=19, cex=0.3)
```

DB SCAN con ALR

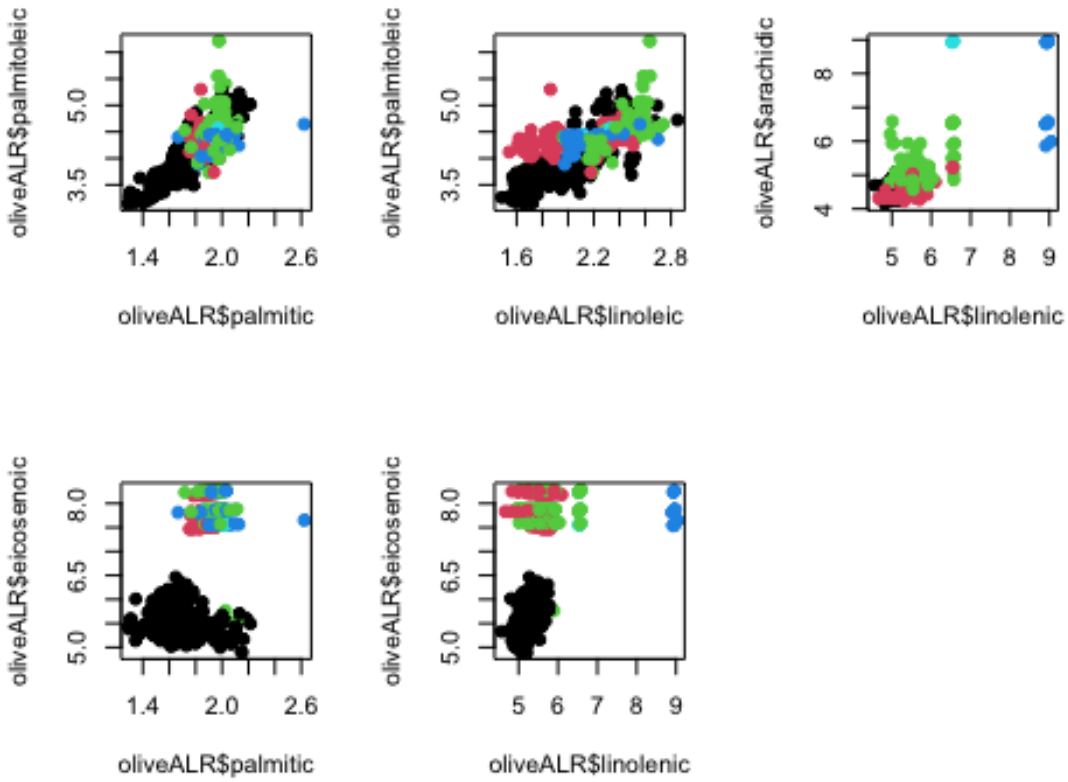
```
set.seed(17)
pdf.out <- pdfCluster(oliveALR[, 3:9], graphtype = 'pairs', lambda = 0.10)
str(pdf.out)
```

```
## Formal class 'pdfCluster' [package "pdfCluster"] with 10 slots
##   ..@ call         : language pdfCluster(x = oliveALR[, 3:9], graphtype =
"pairs", lambda = 0.1)
##   ..@ x            : num [1:572, 1:7] 1.98 1.96 2.19 2.11 2 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : NULL
##   .. .. ..$ : chr [1:7] "palmitic" "palmitoleic" "stearic" "linoleic" ...
##   ..@ pdf          :List of 4
##   .. ..$ kernel  : chr "gaussian"
##   .. ..$ bwtype  : chr "fixed"
##   .. ..$ par     :List of 2
##   .. .. ..$ h : Named num [1:7] 0.0968 0.2572 0.0817 0.1641 0.4574 ...
```

```
##   .. .. .. ..- attr(*, "names")= chr [1:7] "palmitic" "palmitoleic"
"stearic" "linoleic" ...
##   .. .. ..$ hx: NULL
##   .. ..$ estimate: num [1:572] 0.257 0.235 0.123 0.26 0.324 ...
##   ..@ nc           :List of 4
##   .. ..$ nc: Named num [1:116] 0 1 1 1 1 1 1 1 1 1 ...
##   .. .. ..- attr(*, "names")= chr [1:116] "0" "0.0087" "0.0174" "0.0261"
...
##   .. ..$ p : num [1:116] 0 0.0087 0.0174 0.0261 0.0348 ...
##   .. ..$ id: num [1:572, 1:116] -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr [1:116] "" "0.0087" "0.0174" "0.0261" ...
##   .. ..$ pq: num [1:116, 1:2] 0 0.0087 0.0174 0.0261 0.0348 ...
##   .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. ..$ : NULL
##   .. .. .. ..$ : chr [1:2] "p" "q"
##   ..@ graph        :List of 3
##   .. ..$ type     : chr "pairs"
##   .. ..$ lambda   : num 0.1
##   .. ..$ comp.area:List of 2
##   .. .. ..$ area     : num [1:163306] 0 0 0 0 0 0 0 0 0 0 ...
##   .. .. ..$ pairs.ord: int [1:2, 1:163306] 1 2 1 3 1 4 1 5 1 6 ...
##   ..@ cluster.cores: num [1:572] NA NA NA NA NA NA NA NA NA NA ...
##   ..@ tree         :  .. ..--[dendrogram w/ 1 branches and 5 members at h
= 1]
##   .. ..  `--[dendrogram w/ 3 branches and 5 members at h = 0.948, label =
{1,2,3,4,5}]
##   .. ..     |--[dendrogram w/ 2 branches and 3 members at h = 0.574, label
= {1,2,3}, leaf = FALSE]
##   .. ..     |  |--[dendrogram w/ 2 branches and 2 members at h = 0.504,
label = {1,2}, leaf = FALSE]
##   .. ..     |  |  |--leaf "1 "
##   .. ..     |  |  `--leaf "2 " (h= 0.0957  )
##   .. ..     |  `--leaf "3 " (h= 0.27  )
##   .. ..     |--leaf "4 " (h= 0.713  )
##   .. ..     `--leaf "5 " (h= 0.861  )
##   ..@ noc          : num 5
##   ..@ stages       :List of 5
##   .. ..$ 1: num [1:572] NA NA NA NA NA NA NA NA NA NA ...
##   .. ..$ 2: num [1:572] 1 1 NA NA 1 NA NA 1 1 NA ...
##   .. ..$ 3: num [1:572] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..$ 4: num [1:572] 1 1 1 1 1 1 1 1 1 1 ...
##   .. ..$ 5: num [1:572] 1 1 1 1 1 1 1 1 1 1 ...
##   ..@ clusters     : num [1:572] 1 1 1 1 1 1 1 1 1 1 ...
```

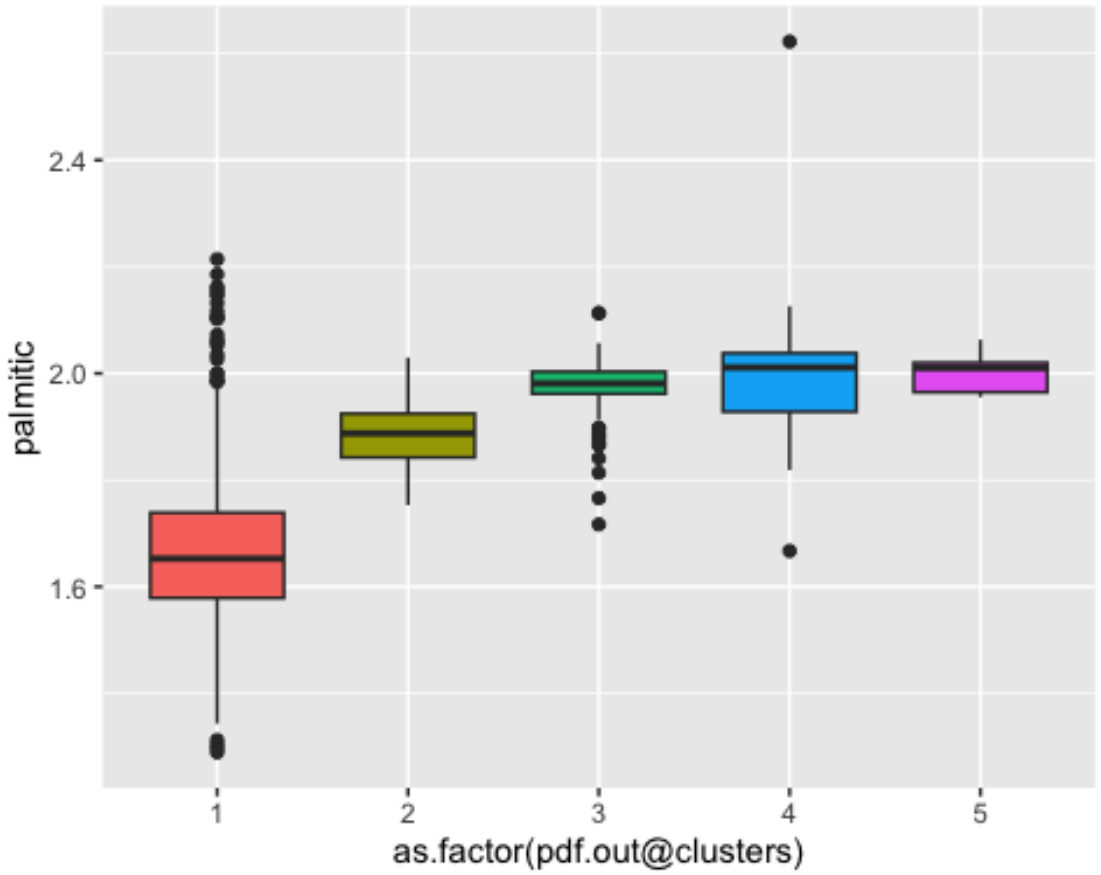
Per meglio visualizzare i cluster abbiamo selezionato le coppie di acidi con correlazione maggiore.

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

# palmitic palmitoleic
plot(oliveALR$palmitic, oliveALR$palmitoleic, col = pdf.out@clusters, pch =
19)
# linoleic palmitoleic
plot(oliveALR$linoleic, oliveALR$palmitoleic, col = pdf.out@clusters, pch =
19)

# arachidic linolenic
plot(oliveALR$linolenic, oliveALR$arachidic, col = pdf.out@clusters, pch =
19)

# eicosenoic palmitic
plot(oliveALR$palmitic, oliveALR$eicosenoic, col = pdf.out@clusters, pch =
19)

# eicosenoic linolenic
plot(oliveALR$linolenic, oliveALR$eicosenoic, col = pdf.out@clusters, pch =
19)

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



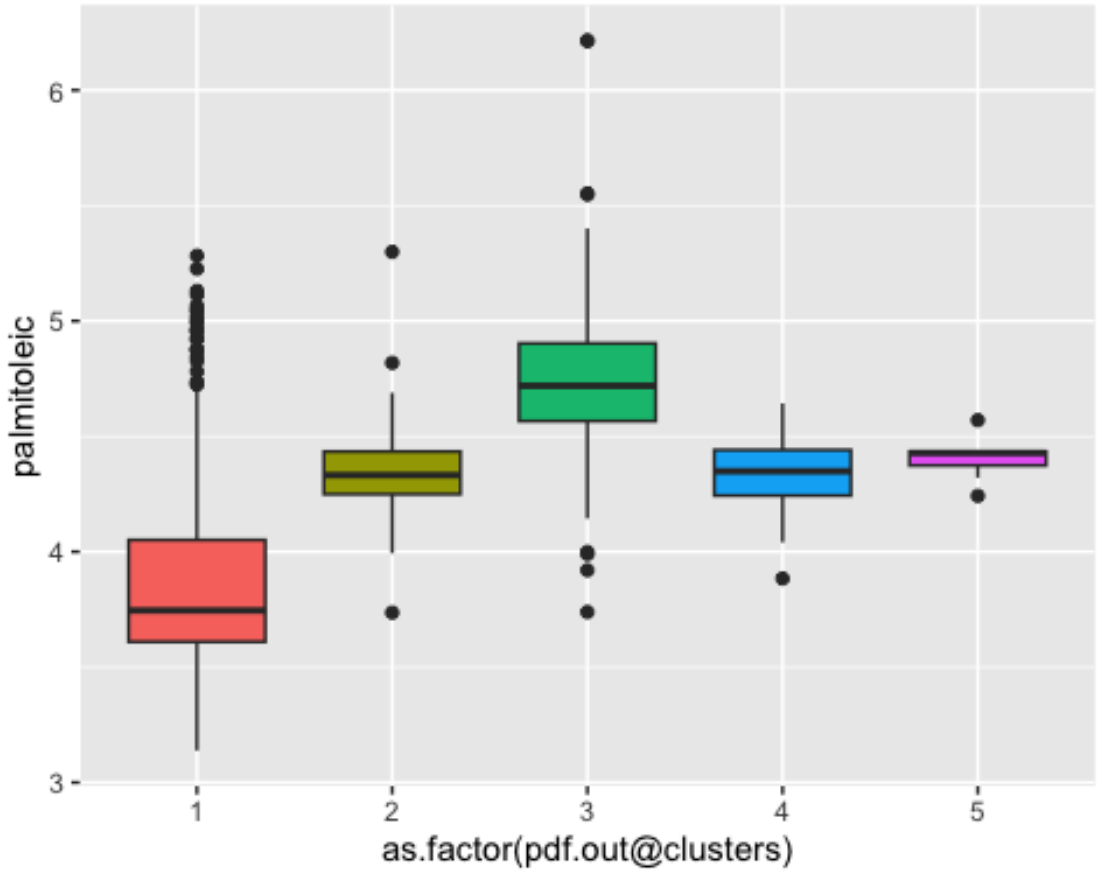
Variabile palmitic

```
ggplot(oliveALR, aes(x = as.factor(pdf.out@clusters), y = palmitic, fill =
as.factor(pdf.out@clusters))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



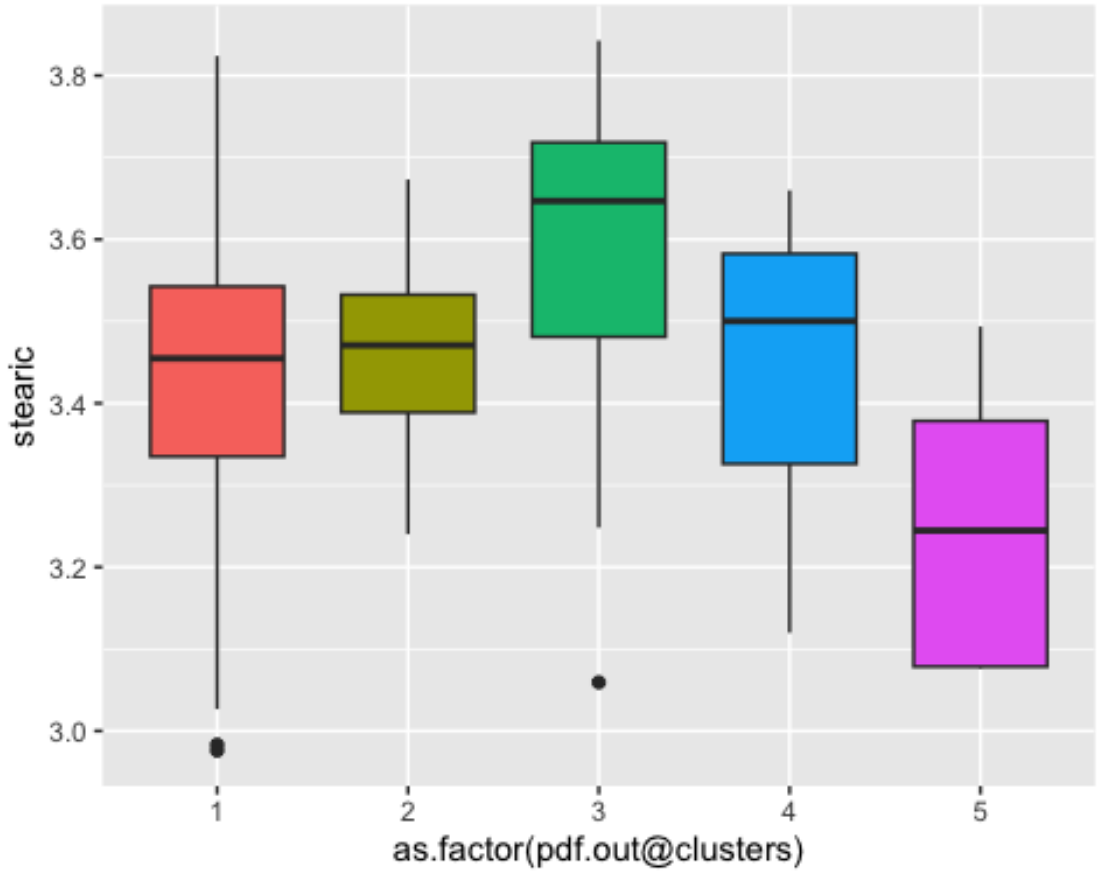
Variabile palmitoleic

```
ggplot(oliveALR, aes(x = as.factor(pdf.out@clusters), y = palmitoleic, fill =
as.factor(pdf.out@clusters))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



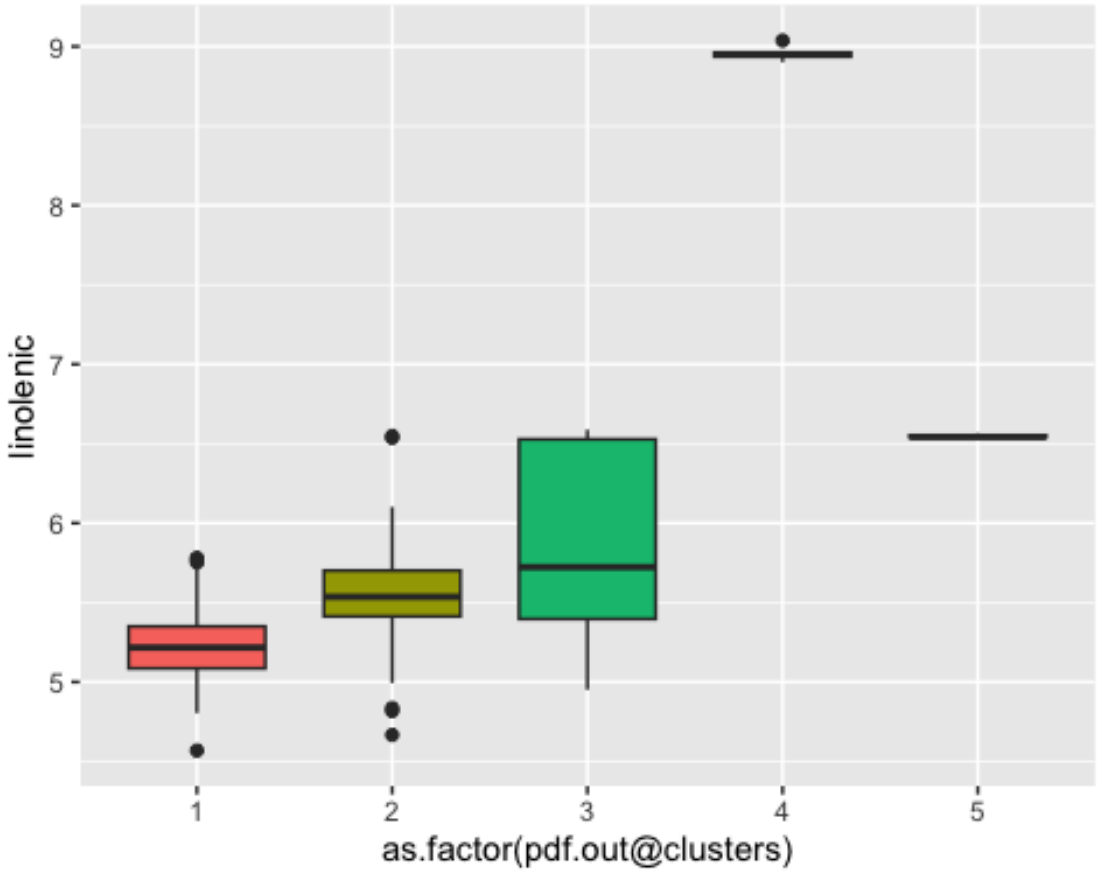
Variabile stearic

```
ggplot(oliveALR, aes(x = as.factor(pdf.out@clusters), y = stearic, fill =
as.factor(pdf.out@clusters))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



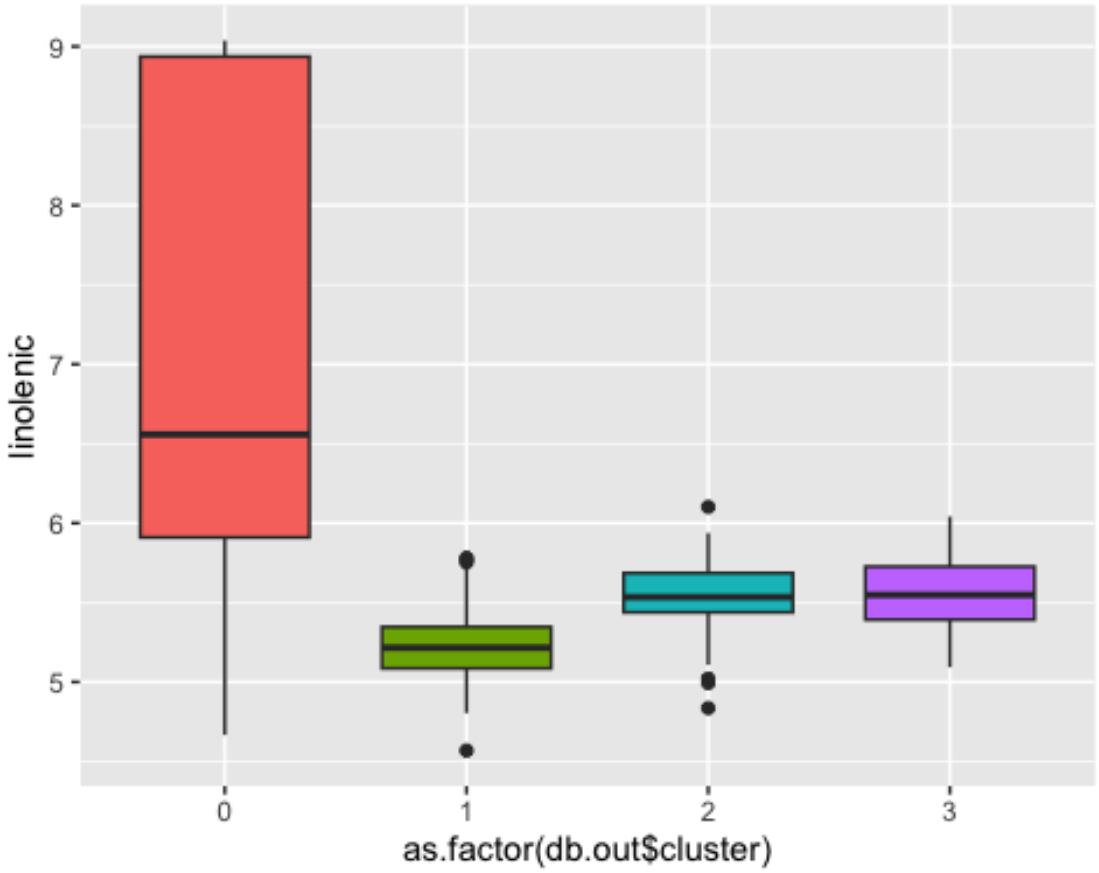
Variabile Linoleic

```
ggplot(oliveALR, aes(x = as.factor(pdf.out@clusters), y = linolenic, fill =
as.factor(pdf.out@clusters))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



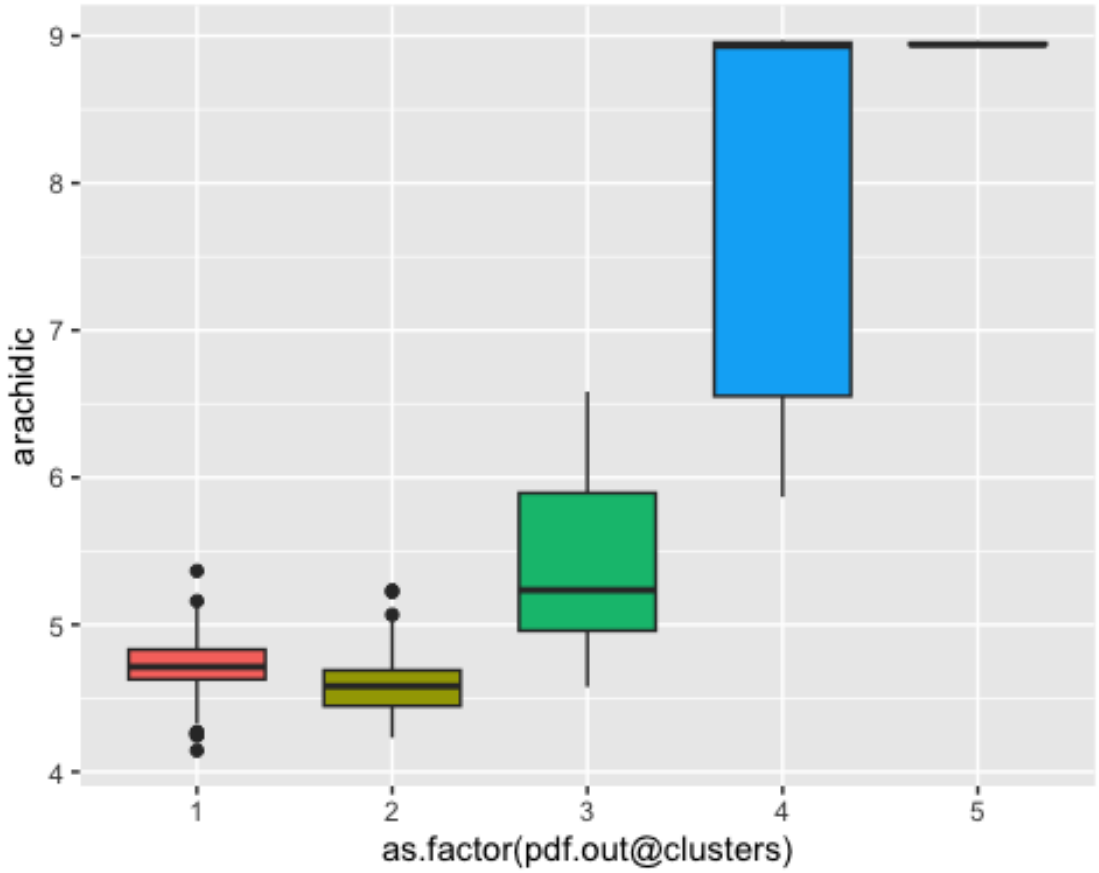
Variabile Linolenic

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



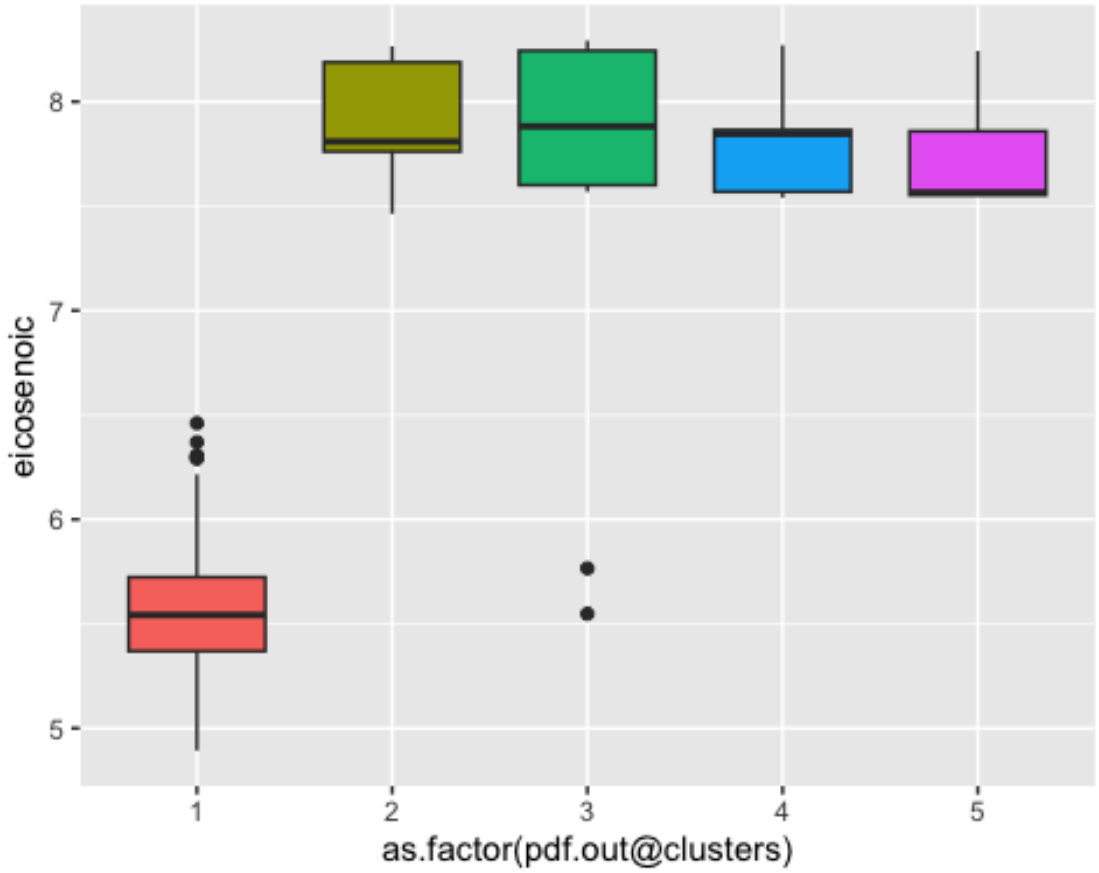
Variabile arachidic

```
ggplot(oliveALR, aes(x = as.factor(pdf.out@clusters), y = arachidic, fill =
as.factor(pdf.out@clusters))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```



Variabile eicosenoic

```
ggplot(oliveALR, aes(x = as.factor(pdf.out@clusters), y = eicosenoic, fill =
as.factor(pdf.out@clusters))) + geom_boxplot(width=0.7) + guides(fill =
FALSE)
```

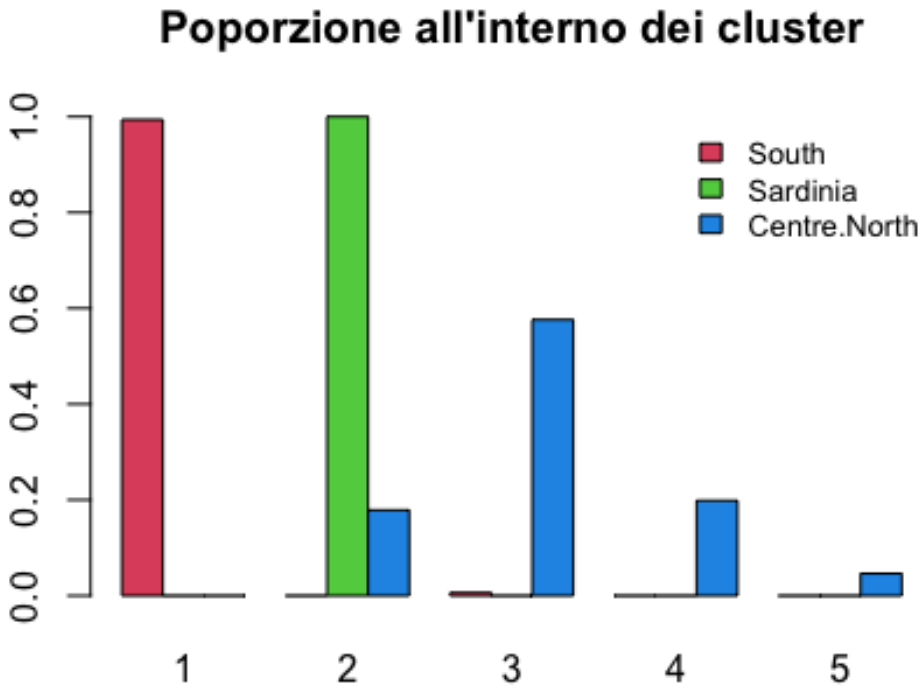


Variabile macro.area

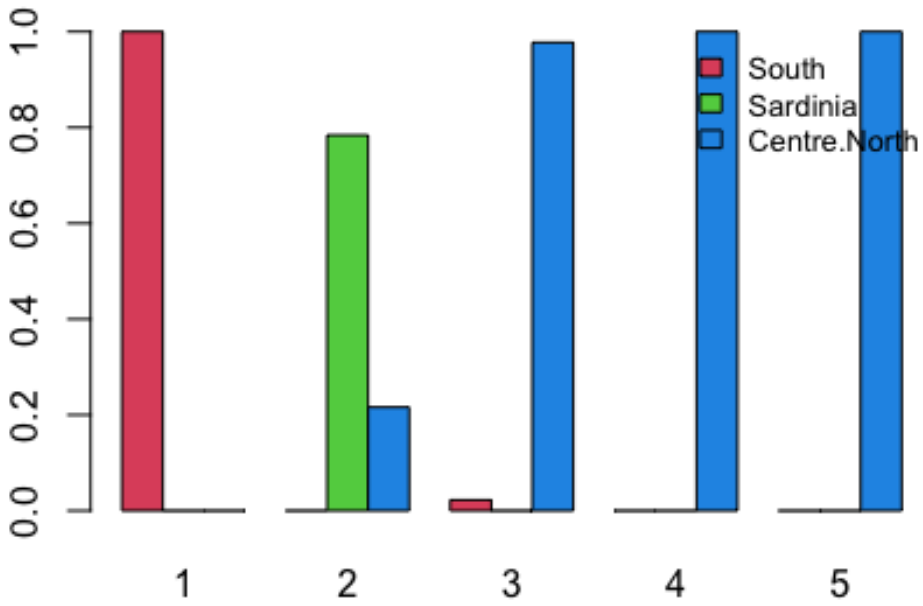
```
prop.table(table(oliveALR$macro.area, pdf.out@clusters),1)
```

```
##
##                          1          2          3          4          5
##   South        0.99380805 0.00000000 0.00619195 0.00000000 0.00000000
##   Sardinia     0.00000000 1.00000000 0.00000000 0.00000000 0.00000000
##   Centre.North 0.00000000 0.17880795 0.57615894 0.19867550 0.04635762
```

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




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

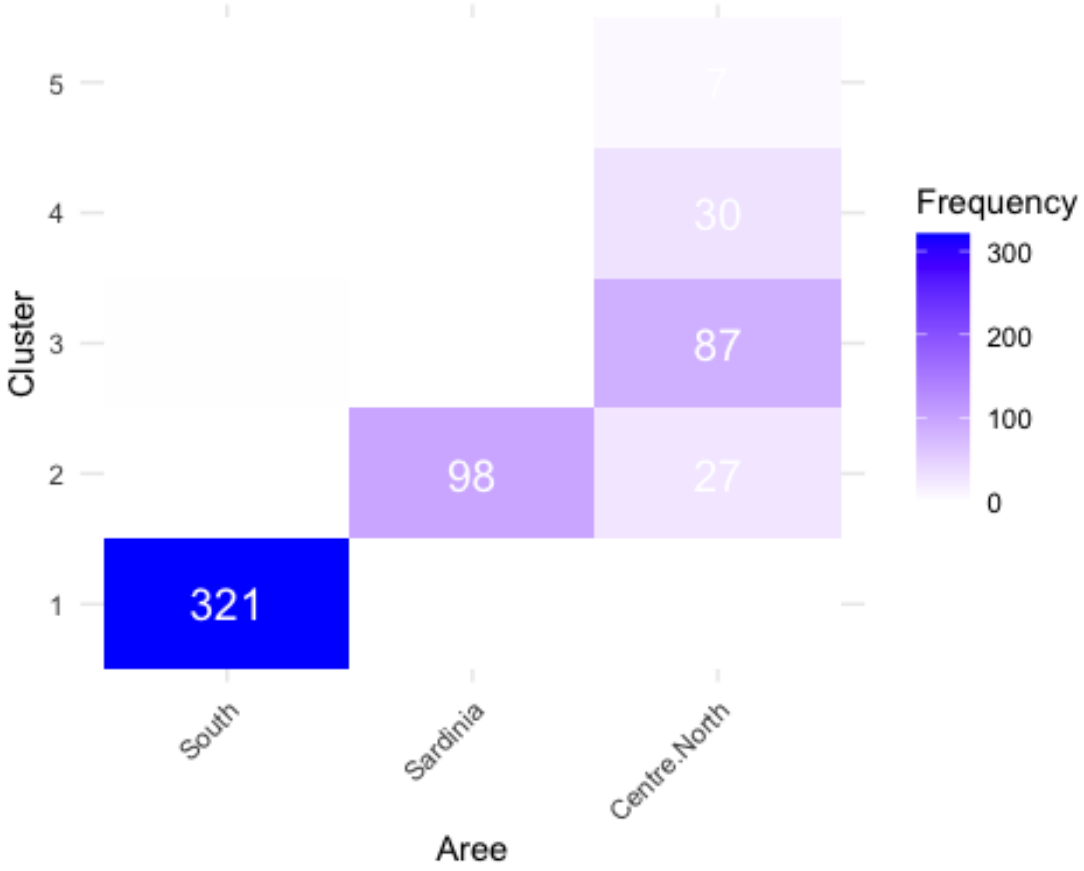


Gli oli del sud si concentrano al 99% nel cluster 1, 1% nel cluster 3. Gli oli della Sardegna si trovano al 100% nel cluster 2. Gli oli del centro nord si dividono al 18% nel cluster 2, al 58% nel cluster 3, al 20% nel cluster 4, al 4% nel cluster 5. IL cluster 1 è formato al 100% da oli del sud. IL cluster 2 è composto al 78% da oli della sardegna e al 22% da oli del centro nord. Il cluster 4 è composto al 100% da oli del centro nord. Il cluster 5 al 100% da oli del centro nord.

Confusion Matrix:

```
confusion_matrix <- table(Aree = oliveALR$macro.area, Cluster =
pdf.out@clusters)

table(Aree = oliveALR$macro.area, Cluster = pdf.out@clusters)

##               Cluster
## Aree             1   2   3   4   5
##   South        321   0   2   0   0
##   Sardinia       0  98   0   0   0
##   Centre.North   0  27  87  30   7

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

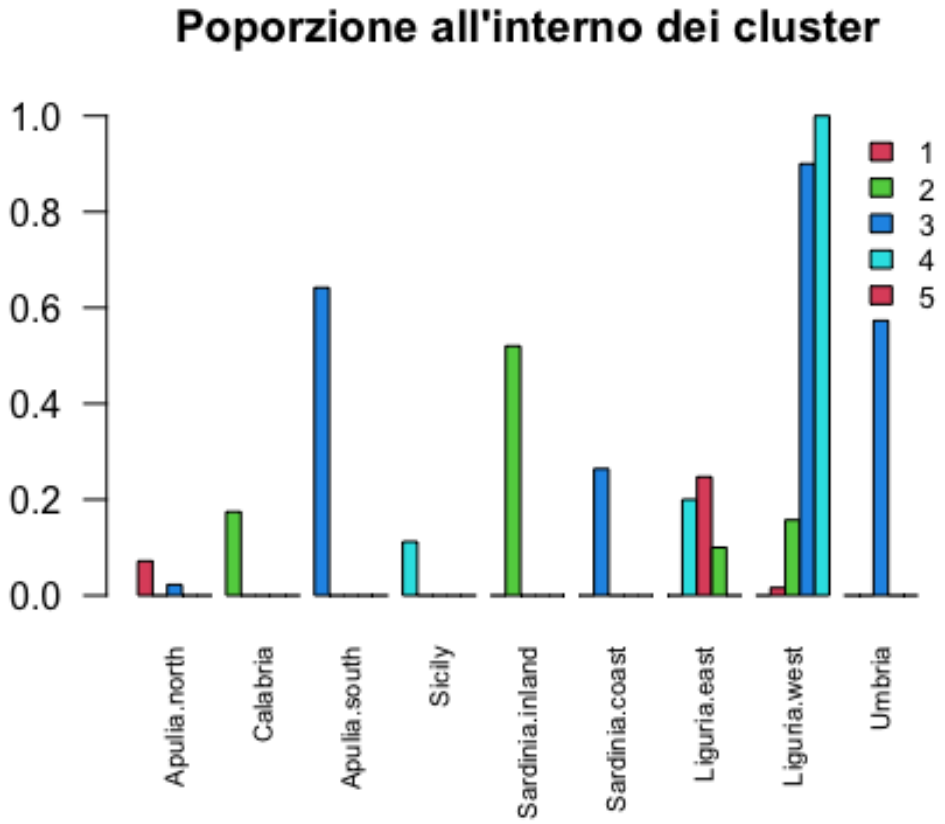


Variabile region

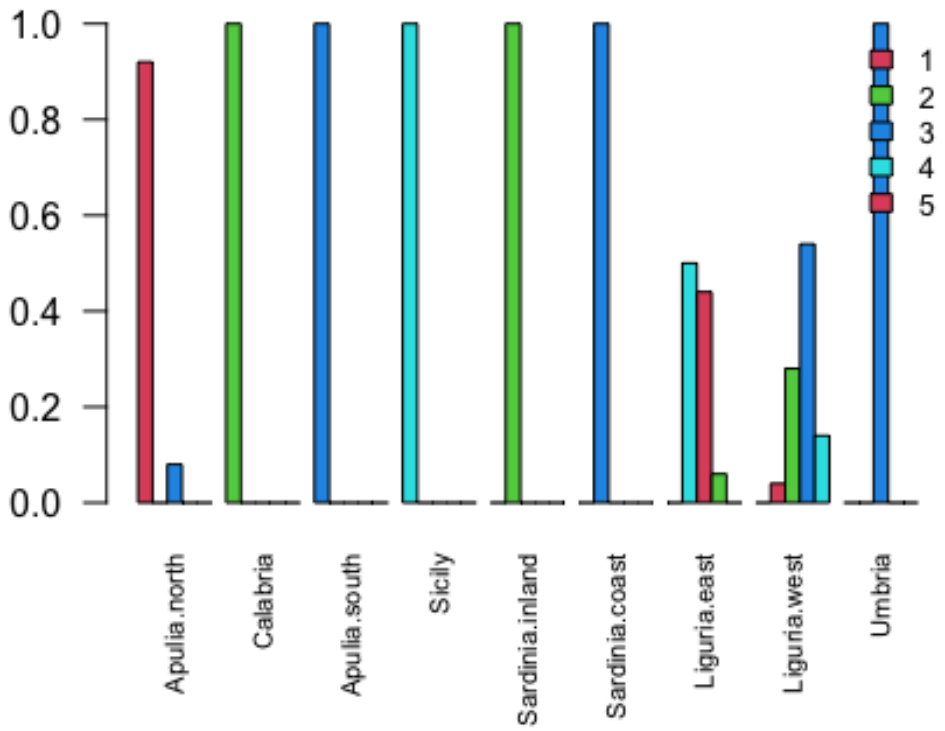
```
prop.table(table(pdf.out@clusters, oliveALR$region),1)
```

```
##
##      Apulia.north   Calabria Apulia.south     Sicily Sardinia.inland
##   1   0.07165109 0.17445483   0.64174455 0.11214953      0.00000000
##   2   0.00000000 0.00000000   0.00000000 0.00000000      0.52000000
##   3   0.02247191 0.00000000   0.00000000 0.00000000      0.00000000
##   4   0.00000000 0.00000000   0.00000000 0.00000000      0.00000000
##   5   0.00000000 0.00000000   0.00000000 0.00000000      0.00000000
##
##      Sardinia.coast Liguria.east Liguria.west     Umbria
##   1      0.00000000   0.00000000   0.00000000 0.00000000
##   2      0.26400000   0.20000000   0.01600000 0.00000000
##   3      0.00000000   0.24719101   0.15730337 0.57303371
##   4      0.00000000   0.10000000   0.90000000 0.00000000
##   5      0.00000000   0.00000000   1.00000000 0.00000000
```

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




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



Il cluster 1 è composto al 64% da oli della puglia del sud, al 11% da oli della sicilia, al 17% da oli della calabria, e al 7% da oli della puglia del nord. Il cluster 2 è composto al 52% da oli della sardegna Inland, al 26% da oli della sardegna coast, al 20% da oli della liguria est. Il cluster 3 è composto al 2% da oli della puglia del nord,al 25% da oli della liguria est, al 16% da oli della liguria ovest, al 57% da oli dell'umbria. Il cluster 4 è composto al 100% da oli della liguria est. Il cluster 5 è composto al 100% da oli della liguria oovest.

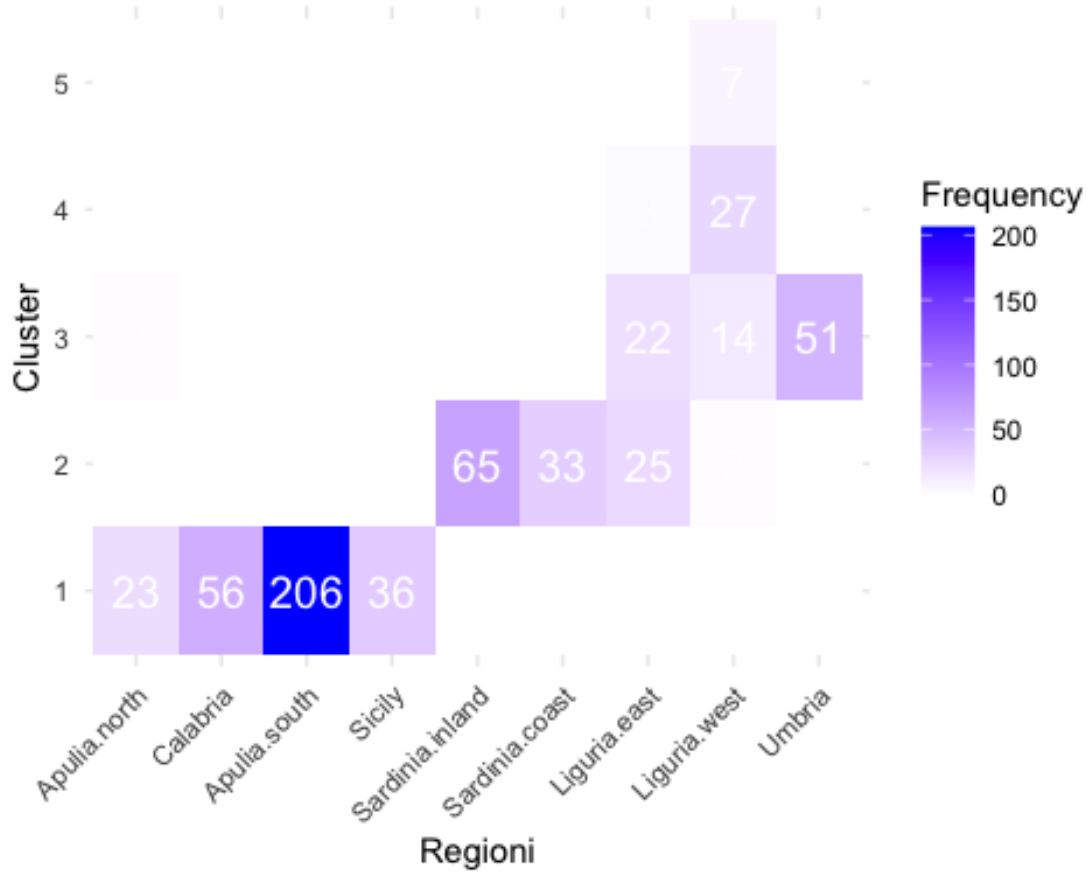
Gli oli della puglia nord si dividono tra cluster 1 (92%) e 3 (8%). Gli oli della Calabria si trovano al 100% nel cluster 1. Gli oli della puglia del sud si trovano al 100% nel cluster 1. Gli oli della Sicilia si trovano al 100% nel cluster 1. Gli oli della Sardegna inland si trovano al 100% nel cluster 2. Gli oli della liguria est si trovano nel cluster 2 (50%), nel cluster 3 (44%), nel cluster 4 (6%). Gli oli della liguria ovest si trovano nel cluster 2 (4%), cluster 3 (28%), cluster 4 (54%) e cluster 5 (14%). Gli oli dell'Umbria si trovano al 100% nel cluster 3.

Confusion Matrix:

```
confusion_matrix_regioni <- table(Regioni = oliveALR$region, Cluster =
pdf.out@clusters)

table(Regioni = oliveALR$region, Cluster = pdf.out@clusters)
```

```
##                   Cluster
## Regioni             1   2   3   4   5
##   Apulia.north     23   0   2   0   0
##   Calabria         56   0   0   0   0
##   Apulia.south    206   0   0   0   0
##   Sicily           36   0   0   0   0
##   Sardinia.inland   0  65   0   0   0
##   Sardinia.coast    0  33   0   0   0
##   Liguria.east      0  25  22   3   0
##   Liguria.west      0   2  14  27   7
##   Umbria            0   0  51   0   0
```

```
ggplot(data = as.data.frame(as.table(confusion_matrix_regioni)), aes(x =
Regioni, y = Cluster, 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))
```



Adjusted Rand Index - pdfCluster

```
ari_pdf_ln <- adj.rand.index(oliveALR$macro.area, pdf.out@clusters)
ari_pdf_ln
```

```
## [1] 0.869787
```

Mappa dei cluster sulla cartina Italiana

```
map("italy", col="white", fill=TRUE, lty=1, lwd=1, border="black")
points(oliveGPS$long, oliveGPS$lat, col=pdf.out@clusters+1, pch=19, cex=0.3)
```

Confronto dei diversi algoritmi

Per confrontare l'accuratezza degli algoritmi di clustering visti si sceglie di usare l'indice ARI (adjusted rand index) e comparare gli i gruppi formati con le macro aree fornite dal dataset

Valori prossimi a 1 indicano che i cluster combaciano perfettamente con le macro aree, mentre indici prossimi allo 0 indicano una cattiva suddivisione in cluster

Si ottengono infatti i seguenti indici:

```
print(c("k-means: ", ari_km_ln), quote = FALSE)
```

```
## [1] k-means:          0.866062163653925
```

```r
print(c("PAM: ", ari_pam_ln), quote = FALSE)
```

```
## [1] PAM:              0.528549655697799
```

```r
print(c("DBSCAN:", ari_db_ln), quote = FALSE)
```

```
## [1] DBSCAN:           0.834170598481717
```

```r
print(c("pdfCluster: ", ari_pdf_ln), quote = FALSE)
```

```
## [1] pdfCluster:       0.869786991818853
```

Guardando l'indice si evince che l'algorimo che suddividono in cluster con più accuratezza è pdfCluster, mentre PAM è l'algoritmo con ARI peggiore.

Sorprendente è l'accuratezza di k-means, che si avvicina molto a quella di pdfCluster pur utilizzando un'algorimo meno complesso.

Considerazioni finali

- La Sardegna risulta essere sempre isolata in un cluster
- La Liguria è consistentemente la regione che presenta più varianza al suo interno: tutti gli algoritmi hanno diviso gli oli provenienti da essa in diversi cluster
- L'Umbria è nella gran parte dei casi individuata perfettamente
- gli oli provenienti dalla macre area Sud presentano caratteristiche molto simili tra loro
- La Puglia del Sud viene considerata cluster separato in alcuni casi come usando l'algoritmo PAM