# Cluster Analysis

Alice Cappella

Per effettuare la *cluster analysis* sul dataset `cars` possiamo percorrere due strade differenti:

1. selezionare le variabili quantitative e qualitative maggiormente legate alla variabile di interesse (`prezzo_auto`) ed applicare metodologie che consentono la presenza di variabili di natura mista;
2. selezionare solamente le variabili quantitative maggiormente correlate con `prezzo_auto` in modo tale da poter sfruttare diversi metodi di *clustering*.

Percorriamo innanzitutto la prima strada e, dopo aver calcolato la correlazione tra le variabili quantitative e l'indice $\eta^2$ per quelle qualitative, selezioniamo solamente le variabili più legate con `prezzo_auto`.

```
library(dplyr)
library(reshape2)

cars = read.csv("cars-clean-v2-imputed.csv")
cars = cars %>%
  mutate_if(is.character,as.factor) %>%
  mutate(across(.cols = where( ~ n_distinct(.) < 6),as.factor))
cars_numeric = cars %>%
  select(where(is.numeric))
corr = cor(cars_numeric)
corr
```

```
##                 prezzo_auto   anno_prod chilometraggio         kW         cv
## prezzo_auto     1.000000000  0.63752274    -0.51652262 0.52384490 0.52440789
## anno_prod       0.637522737  1.00000000    -0.73344857 0.01276064 0.01329855
## chilometraggio -0.516522624 -0.73344857     1.00000000 0.14327581 0.14368187
## kW              0.523844905  0.01276064     0.14327581 1.00000000 0.99996172
## cv              0.524407894  0.01329855     0.14368187 0.99996172 1.00000000
## cilindrata      0.158240706 -0.29655606     0.42460803 0.58181038 0.58228208
## peso            0.485241771  0.01230163     0.19755467 0.76182209 0.76259711
## emissioni      -0.003456499 -0.23685878     0.22524855 0.35679034 0.35668196
## consumi        -0.006565334 -0.17318064     0.06380238 0.20449605 0.20407614
##                  cilindrata       peso    emissioni      consumi
## prezzo_auto     0.15824071 0.48524177 -0.003456499 -0.006565334
## anno_prod      -0.29655606 0.01230163 -0.236858781 -0.173180639
## chilometraggio  0.42460803 0.19755467  0.225248552  0.063802384
## kW              0.58181038 0.76182209  0.356790341  0.204496047
## cv              0.58228208 0.76259711  0.356681964  0.204076143
## cilindrata      1.00000000 0.66065242  0.287018630  0.057074325
## peso            0.66065242 1.00000000  0.304695758  0.098770857
## emissioni       0.28701863 0.30469576  1.000000000  0.730125752
## consumi         0.05707432 0.09877086  0.730125752  1.000000000
```

Prendendo come soglia 0.3 possiamo notare che le variabili più correlate con la variabile di interesse risultano: `anno_prod`, `chilometraggio` (negativamente), `kw` e `cv` (di cui ne verrà selezionata solo una dato che riportano informazioni simili) e `peso`.

Procediamo con il calcolo dell'indice $\eta^2$.

```
var_qualitative = names(select(cars,where(is.factor)))
eta2_results = sapply(var_qualitative,function(var) {
  eta2(cars$prezzo_auto,cars[[var]])})
eta2_df = data.frame(variabile = names(eta2_results),eta2 = eta2_results)
eta2_df[which(eta2_df$eta2 > 0.3),]
```

```
##         variabile      eta2
## marca       marca 0.3287687
## modello   modello 0.5750314
## marce       marce 0.3118093
```

Prendendo come soglia 0.3, le variabili con un valore maggiore dell'indice sono `marca`, `modello` e `marce`.

A questo punto possiamo selezionare le variabili individuate.

```
cars_cluster = cars[,c(rownames(eta2_df)[which(eta2_df$eta2 > 0.3)],
                       names(which(abs(corr)[,1] > 0.3)))]
```

Oltre alla variabile `kW`, rimuoviamo anche la variabile `modello`. Ne consegue che le variabili qualitative considerate per effettuare il *clustering* sono `marca` e `marce`.

```
cars_cluster$kW = NULL
cars_cluster$modello = NULL

head(cars_cluster)
```

```
##   marca      marce prezzo_auto anno_prod chilometraggio cv peso
## 1  Fiat          5        6900      2013          32958 69  940
## 2  Fiat    6 o piu       11000      2022          20132 69  980
## 3  Fiat Automatico        3800      2004         116000 60  935
## 4  Fiat    6 o piu       10890      2022          27685 69 1055
## 5  Fiat          5        5900      2011          98000 69  930
## 6  Fiat          5        4900      2009          99817 60  930
```

Passiamo alla standardizzazione del nuovo insieme di dati. Procediamo a passi, iniziamo con le variabili qualitative ed effettuiamo un *One-Hot Encoding*, ossia trasformiamo le variabili categoriali in variabili *dummy* una per ogni modalità che ciascuna variabile presenta.

```
cars_cluster = dummy.data.frame(cars_cluster,names = c("marca","marce"))
cars_cluster = cars_cluster %>%
  mutate_if(~ all(. %in% c(0,1)),as.factor)
head(cars_cluster)
```

```
##   marcaAlfa Romeo marcaAudi marcaBMW marcaCitroen marcaCUPRA marcaDacia
## 1               0         0        0            0          0          0
## 2               0         0        0            0          0          0
## 3               0         0        0            0          0          0
## 4               0         0        0            0          0          0
## 5               0         0        0            0          0          0
## 6               0         0        0            0          0          0
##   marcaFiat marcaFord marcaHyundai marcaJeep marcaKia marcaLancia
## 1         1         0            0         0        0           0
## 2         1         0            0         0        0           0
## 3         1         0            0         0        0           0
## 4         1         0            0         0        0           0
## 5         1         0            0         0        0           0
## 6         1         0            0         0        0           0
```

```
##   marcaMercedes-Benz marcaMG marcaMINI marcaNissan marcaOpel marcaPeugeot
## 1                  0       0         0           0         0            0
## 2                  0       0         0           0         0            0
## 3                  0       0         0           0         0            0
## 4                  0       0         0           0         0            0
## 5                  0       0         0           0         0            0
## 6                  0       0         0           0         0            0
##   marcaRenault marcaSuzuki marcaToyota marcaVolkswagen marce5 marce6 o piu
## 1            0           0           0               0      1            0
## 2            0           0           0               0      0            1
## 3            0           0           0               0      0            0
## 4            0           0           0               0      0            1
## 5            0           0           0               0      1            0
## 6            0           0           0               0      1            0
##   marceAutomatico prezzo_auto anno_prod chilometraggio cv peso
## 1               0        6900      2013          32958 69  940
## 2               0       11000      2022          20132 69  980
## 3               1        3800      2004         116000 60  935
## 4               0       10890      2022          27685 69 1055
## 5               0        5900      2011          98000 69  930
## 6               0        4900      2009          99817 60  930
```

Per le variabili quantitative effettuiamo la classica standardizzazione.

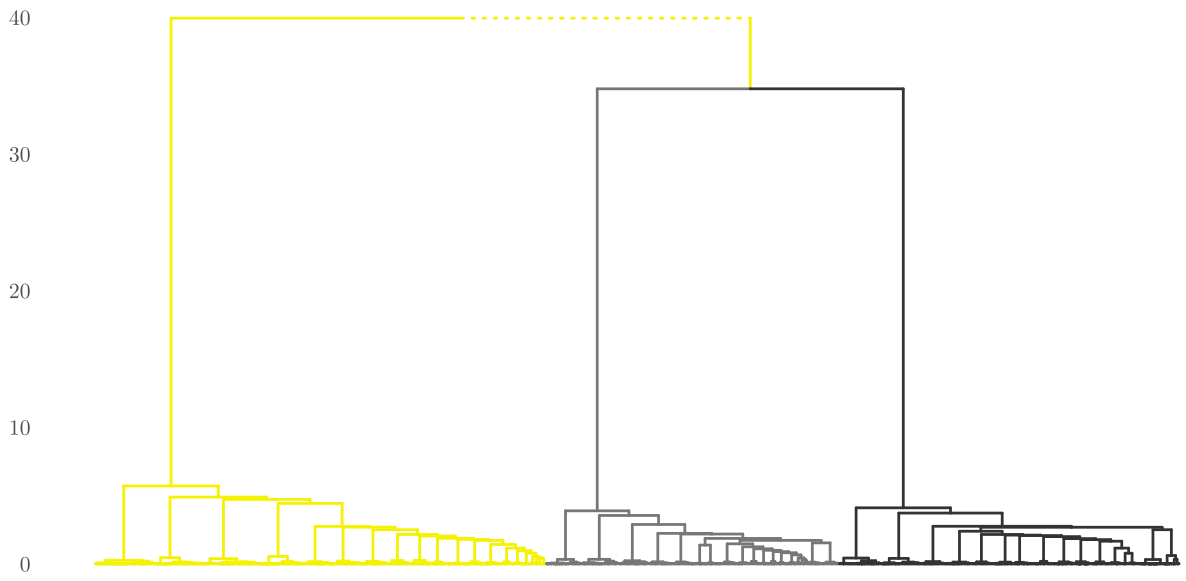
```
var_sel = c("prezzo_auto","anno_prod","chilometraggio","cv","peso")
cars_cluster[,var_sel] = scale(cars_cluster[,var_sel])
```

Possiamo ora calcolare una misura di dissimilarità per variabili di natura mista, il **coefficiente di *Gower***. Si specifica che considerando sia variabili quantitative che qualitative sarà possibile solamente applicare i metodi di tipo gerarchico. In particolare, come metodo di agglomerazione utilizziamo il legame di *Ward*.

```
library(cluster)
gower = daisy(cars_cluster,metric = "gower")
hc = hclust(gower,method = "ward.D")
```

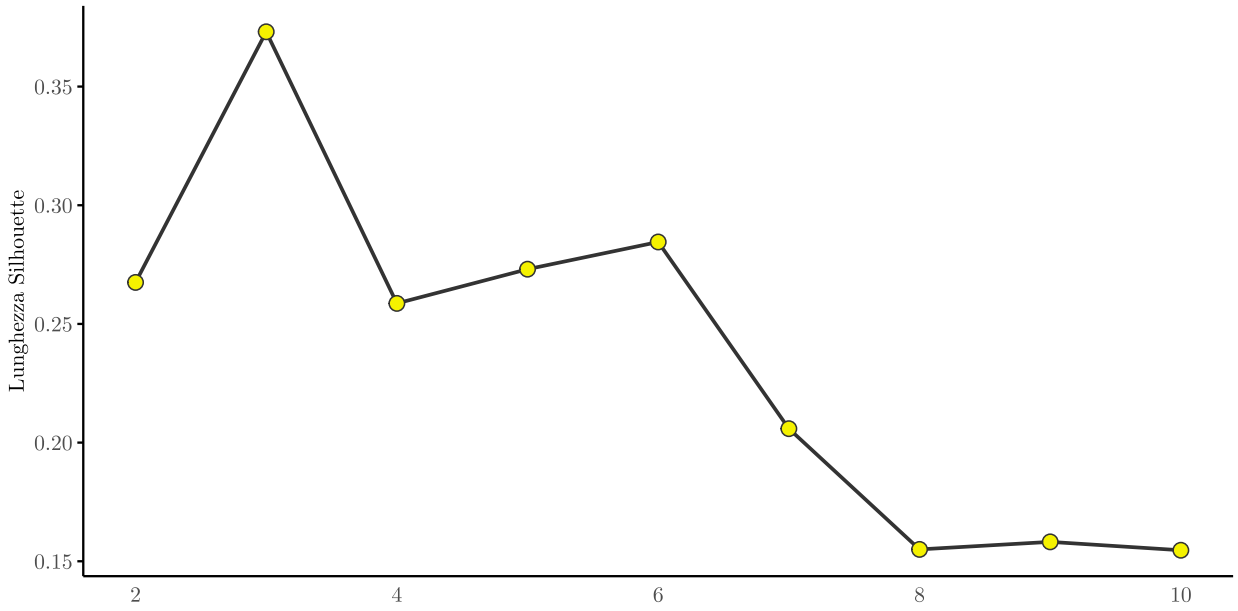
Rappresentiamo il dendrogramma da cui possiamo presumere la presenza di tre gruppi.

```
hc_data = dendro_data_k(hc,3)
plot_ggdendro(hc_data,show_labels = F,direction = "tb",expand.y = 0,
              scale.color = c("#F5F200","#333333","#F5F200","#777777"),
              branch.size = 0.6) +
  theme_minimal() +
  theme(text = element_text(family = "CMUSerif"),
        axis.title = element_blank(),
        axis.text = element_text(size = 10),
        panel.grid = element_blank())
```




Non avendo la possibilità di realizzare lo *screeplot* della varianza spiegata al crescere del numero di gruppi, sfuttiamo il valore di ***silhouette*** che fornisce un'indicazione della coesione interna e della separazione esterna dei *cluster*.

```
sil_width = c(NA)
for(i in 2:10){
  pam_fit = pam(gower,
                diss = TRUE,
                k = i)
  sil_width[i] = pam_fit$silinfo$avg.width
}
silhouette.df = data.frame(x = 2:10,
                      sil_width = sil_width[2:10])

silhouette.df %>%
  ggplot(aes(x,sil_width)) +
  geom_line(col = "#333333",size = 0.8) +
  geom_point(shape = 21,color="#333333",fill = "#F5F200",size = 3) +
  ylab("Lunghezza Silhouette") +
  theme_minimal()+
  theme(axis.text.x = element_text(size = 10),
        axis.ticks.x = element_blank(),
        axis.text.y = element_text(size = 10),
        axis.title.y = element_text(size = 10),
        axis.title.x = element_blank(),
        text = element_text(family = "CMUSerif"),
        panel.grid = element_blank(),
        axis.line  = element_line(colour = "black"),
        axis.ticks = element_line(colour = "black"))
```

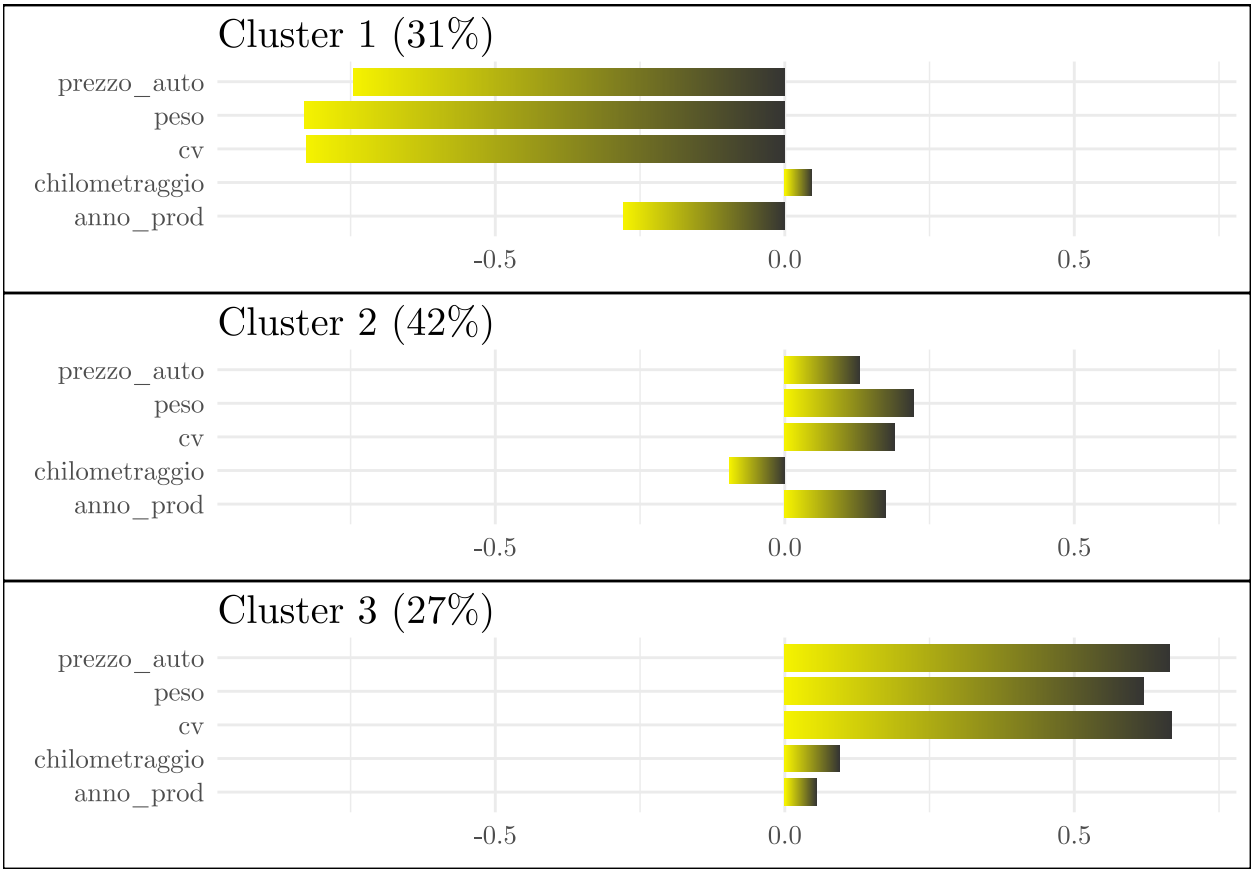



Anche la *silhouette* seleziona come numero ottimale di gruppi 3.

Cerchiamo di interpretare le caratteristiche dei *cluster* ottenuti. Valutiamo innanzitutto qual è la media delle variabili quantitative all'interno dei gruppi e confrontiamola con quella generale, che è zero considerato che queste variabili sono state standardizzate. Effettuiamo una rappresentazione grafica per un'interpretazione più diretta.

```
library(tidyr)
library(ggpattern)
library(ggpubr)
hc_gower3 = cutree(hc,k = 3)
hc_gower3.means = cars_cluster %>%
  select(where(is.numeric)) %>%
  aggregate(. ~ hc_gower3,mean)
hc_gower3.means = hc_gower3.means %>%
  gather(key = "variable",value = "value",2:ncol(hc_gower3.means))

p1 = hc_gower3.means %>%
  filter(hc_gower3 == 1) %>%
  ggplot(aes(x = variable,y = value)) +
  geom_col_pattern(pattern = "gradient",pattern_colour = "white",
                   pattern_fill = "#F5F200",pattern_fill2 = "#333333",
                   pattern_orientation = "horizontal",width = 0.8) +
  coord_flip() +
  ylim(-0.9,0.7) +
  labs(title = paste("Cluster 1 (",
                     round(length(which(hc_gower3 == 1))/length(hc_gower3),2)*100,
                     "%)",sep = "")) +
  theme_minimal() +
  theme(text = element_text(family = "CMUSerif"),
        axis.text = element_text(size = 10),
        axis.ticks.x = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(size = 15),
        plot.background = element_rect(color = "black"))
```

```
p2 = hc_gower3.means %>%
  filter(hc_gower3 == 2) %>%
  ggplot(aes(x = variable,y = value)) +
  geom_col_pattern(pattern = "gradient",pattern_colour = "white",
                   pattern_fill = "#F5F200",pattern_fill2 = "#333333",
                   pattern_orientation = "horizontal",
                   width = 0.8) +
  coord_flip() +
  ylim(-0.9,0.7) +
  labs(title = paste("Cluster 2 (",
                     round(length(which(hc_gower3 == 2))/length(hc_gower3),2)*100,
                     "%)",sep = "")) +
  theme_minimal() +
  theme(text = element_text(family = "CMUSerif"),
        axis.text = element_text(size = 10),
        axis.ticks.x = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(size = 15),
        plot.background = element_rect(color = "black"))
p3 = hc_gower3.means %>%
  filter(hc_gower3 == 3) %>%
  ggplot(aes(x = variable,y = value)) +
  geom_col_pattern(pattern = "gradient",pattern_colour = "white",
                   pattern_fill = "#F5F200",pattern_fill2 = "#333333",
                   pattern_orientation = "horizontal",
                   width = 0.8) +
  coord_flip() +
  ylim(-0.9,0.7) +
  labs(title = paste("Cluster 3 (",
                     round(length(which(hc_gower3 == 3))/length(hc_gower3),2)*100,
                     "%)",sep = "")) +
  theme_minimal() +
  theme(text = element_text(family = "CMUSerif"),
        axis.text = element_text(size = 10),
        axis.ticks.x = element_blank(),
        axis.title = element_blank(),
        plot.title = element_text(size = 15),
        plot.background = element_rect(color = "black"))

ggarrange(p1,p2,p3,ncol = 1)
```




Possiamo innanzitutto affermare che il *cluster* più numeroso sia il secondo, che contiene il 42% delle auto. Inoltre,

- il primo gruppo presenta solamente un valore di chilometraggio superiore alla media mentre un valore inferiore per le restanti variabili. Questo gruppo includerà dunque auto più vecchie, meno potenti e meno costose, con un elevato chilometraggio. Potrebbero essere ad esempio *city car* che, essendo sul mercato da più tempo, sono più accessibili economicamente;
- il secondo gruppo mostra caratteristiche opposte al primo *cluster*, vale a dire un valore di prezzo, peso, cavalli e anno di produzione superiore e un chilometraggio inferiore alla media generale. Questo gruppo rappresenta quindi auto più nuove, potenti e costose. Potrebbero essere ad esempio SUV moderni;
- il terzo gruppo evidenzia per tutte le variabili un valore superiore alla media, soprattutto per quanto riguarda prezzo, peso e cavalli. Il terzo gruppo include auto costose, molto pesanti e potenti. Potrebbero essere ad esempio SUV di alta gamma o auto sportive. Il fatto che tutte le variabili siano superiori alla media indica che si tratti di auto di qualità superiore e potrebbero essere meno comuni sul mercato.

Valutiamo anche come si suddividono le marche all'interno dei *cluster*.

```
cars_cluster2 = as.data.frame(cbind(cars$marca,cars$marce))
colnames(cars_cluster2) = c("marca","marce")
cars_cluster2$cluster = hc_gower3
```

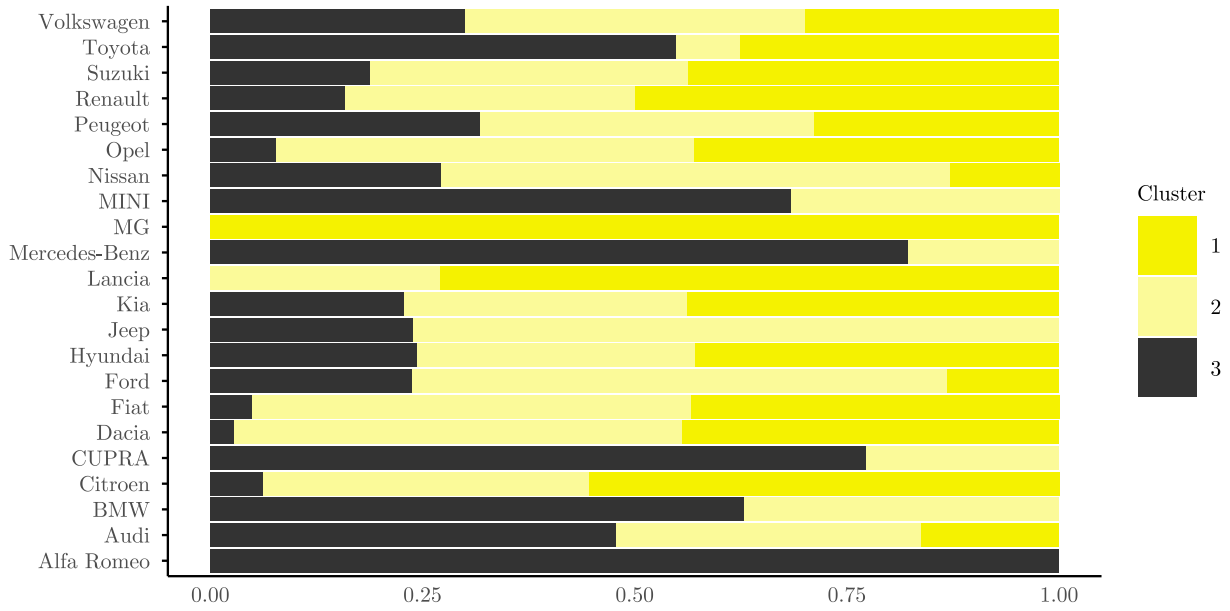
```r
table(cars_cluster2$marca,cars_cluster2$cluster)
```

```
##
##                   1  2  3
##   Alfa Romeo      0  0 21
##   Audi           15 33 44
##   BMW             0 13 22
##   Citroen        36 25  4
##   CUPRA           0  5 17
##   Dacia          32 38  2
##   Fiat           62 74  7
##   Ford           19 90 34
##   Hyundai        30 23 17
##   Jeep            0 70 22
##   Kia            25 19 13
##   Lancia         27 10  0
##   Mercedes-Benz   0  5 23
##   MG             24  0  0
##   MINI            0  6 13
##   Nissan          9 42 19
##   Opel           28 32  5
##   Peugeot        31 42 34
##   Renault        41 28 13
##   Suzuki         28 24 12
##   Toyota         35  7 51
##   Volkswagen     57 76 57
```

Già da questa tabella si può osservare che non ci sia una separazione netta delle marche all'interno dei *cluster*, eccetto per pochi casi come Alfa Romeo e MG.

Possiamo anche rappresentare graficamente questa informazione.

```r
cars_cluster2 %>%
  ggplot(aes(x = marca,fill = as.factor(cluster))) +
  geom_bar(position = "fill") +
  coord_flip() +
  labs(fill = "Cluster") +
  scale_fill_manual(values = palette_function(3)) +
  theme_minimal() +
  theme(text = element_text(family = "CMUSerif"),
        axis.text.x = element_text(size = 10),
        axis.ticks.x = element_blank(),
        axis.text.y = element_text(size = 10),
        legend.text = element_text(size = 10),
        legend.title = element_text(size = 10),
        legend.key.size = unit(1,"cm"),
        axis.title = element_blank(),
        panel.grid = element_blank(),
        axis.line  = element_line(colour = "black"),
        axis.ticks = element_line(colour = "black"))
```




Vediamo infine la suddivisione delle auto per numero di marce all'interno dei tre gruppi formati.

```
table(cars_cluster2$marce,cars_cluster2$cluster)
```

```
##
##                1   2   3
##   5          489   0   0
##   6 o piu      5 649   0
##   Automatico   4   0 414
```

La suddivisione è risulta più netta, rispetto alla marca dell'auto. Le auto con 5 marce sono contenute nel primo gruppo, quelle con 6 o più marce nel secondo e, infine, le auto con il cambio automatico si trovano nel terzo *cluster*.

Percorriamo ora la seconda strada e concentriamoci sulle variabili quantitative. Oltre a quelle già identificate in precedenza, creiamo un'altra variabile relativa alla somma del numero di *optional* che riporta ogni auto. Se questa nuova variabile presenza una correlazione con `prezzo_auto` superiore a 0.3 verrà inclusa nell'insieme di variabili da utilizzare per fare *clustering*.

```
cars = read.csv("cars-clean-v2-imputed.csv")
cars2 = cars[,names(which(abs(corr)[,1] > 0.3))]
cars2$kW = NULL
cars2$modello = NULL

cars2$optional = NA
for(i in 1:nrow(cars2)){
  cars2$optional[i] = sum(as.numeric(cars[i,32:ncol(cars)]))
}
cor(cars2$prezzo_auto,cars2$optional)
```

```
## [1] 0.3134447
```

Includiamo anche la variabile `optional` appena creata e standardizziamo `cars2`.

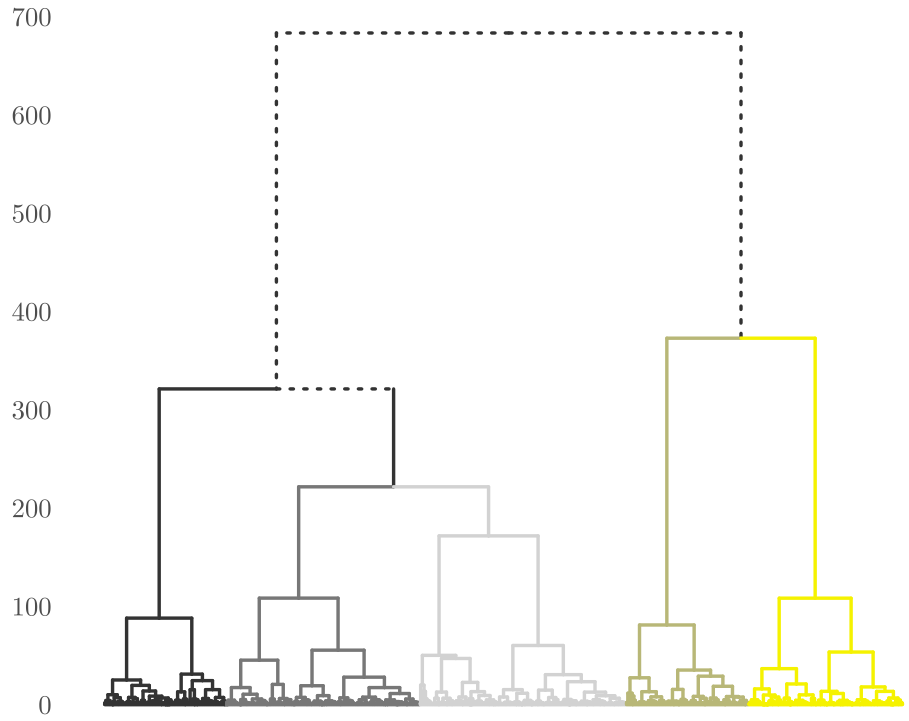
```
cars2 = scale(cars2) %>%
  as.data.frame()
```

Iniziamo con il metodo gerarchico dell'analisi dei gruppi. Calcoliamo le distanze tra le unità utilizzando la distanza euclidea mentre come metodo di agglomerazione il legame di *Ward*.

```
dist_df = dist(cars2)
hc2 = hclust(dist_df,method = "ward.D")
```

Rappresentiamo il dendogramma da cui, anche in questo caso, si può notare la presenza di cinque gruppi.

```
hc2_data = dendro_data_k(hc2,5)
plot_ggdendro(hc2_data,show_labels = F,direction = "tb",expand.y = 0,
              scale.color = c("#333333","#F5F200","#B8B777","#777777","#D2D2D2","#333333"),
              branch.size = 0.6) +
  theme_minimal() +
  theme(text = element_text(family = "CMUSerif"),
        axis.title = element_blank(),
        axis.text = element_text(size = 10),
        panel.grid = element_blank())
```




```
hc5 = cutree(hc2,k = 5)
hc5.means = cars2[,-7] %>%
  aggregate(. ~ hc5,mean)
hc5.means
```

```
##   hc5 prezzo_auto  anno_prod chilometraggio          cv        peso   optional
## 1   1  -0.5380501  0.4435971     -0.5946337 -1.03384333 -1.15227255 -0.0509841
## 2   2  -1.3058503 -1.5315563      1.0725367 -0.67526946 -0.63140621 -0.8529704
## 3   3   0.2403136  0.4026939     -0.3276308 -0.03783562 -0.04636715 -0.4072825
## 4   4   1.0201034  0.5822158     -0.5123905  0.67049132  0.56526402  0.7281840
## 5   5  -0.1161098 -0.6469757      1.0613645  0.91023979  1.20734566  0.3362317
```

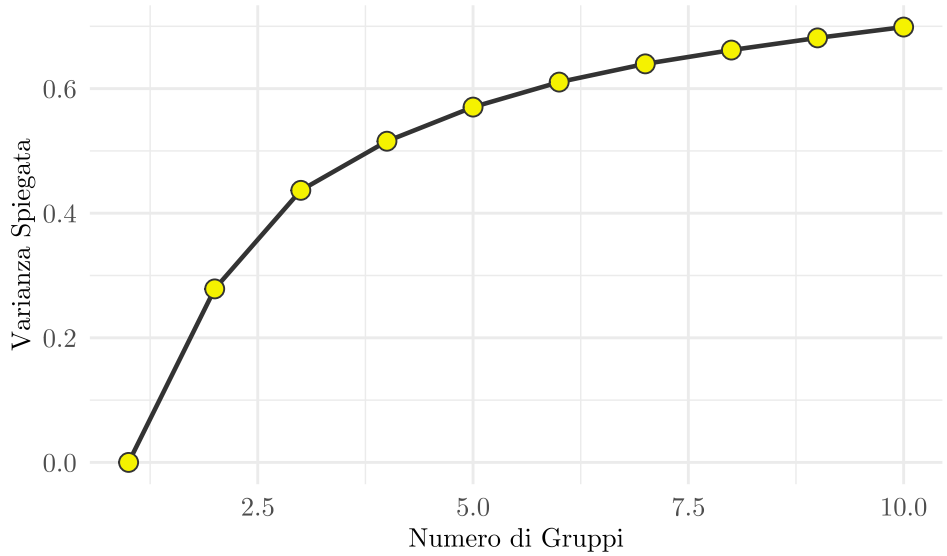
Analizziamo più nel dettaglio le caratteristiche dei gruppi individuati utilizzando il metodo gerarchico:

- il primo gruppo presenta solamente per l'anno di produzione un valore superiore alla media mentre per le altre variabili un valore inferiore. Si può notare, però, come la media del numero di *optional* per questo *cluster* sia molto vicina a zero, vale a dire la media generale. Possiamo affermare che le auto appartenenti a questo gruppo siano auto utilitarie di recente produzione, non molto potenti e con dotazioni essenziali;
- il secondo gruppo mostra un valore superiore alla media solamente per quanto riguarda il chilometraggio. Le auto di questo *cluster* saranno probabilmente auto usate, con un alto chilomeraggio, non molto potenti e pesanti e con pochi *optional*;
- il terzo gruppo riporta un prezzo e un anno di produzione superiore alla media, un valore di chilometraggio e di numero di *optional* inferiore a quest'ultima, mentre le variabili cavalli e peso presentano un valore molto vicino allo zero, ossia alla media generale. In questo gruppo potrebbero far parte auto di fascia media relativamente nuove, avendo un basso chilometraggio, e un prezzo più elevato rispetto alla media;
- il quarto gruppo riporta solamente nel caso del chilometraggio un valore inferiore alla media, l'opposto di quello che avviene per il secondo gruppo. Le auto incluse in questo gruppo saranno dunque auto nuove, più costose, con un basso chilometraggio e molti *optional*. Potrebbero dunque essere auto di fascia più alta o auto sportive;
- il quinto gruppo presenta un valore inferiore alla media solamente per quanto riguarda il prezzo e l'anno di produzione. Essendo auto pesanti e potenti ma relativamente economiche potremmo pensare che le auto raggruppate in questo *cluster* siano usate ma che offrono comunque buone prestazioni e un numero adeguato di dotazioni.

Procediamo ora ai metodi non gerarchici e utilizziamo l'algoritmo *k-means*. Applichiamo questo algoritmo per un numero di gruppi che va da 1 a 10. In questo caso sarà possibile rappresentare lo *screeplot* della varianza spiegata all'aumentare del numero di *cluster* e sfruttarlo per capire qual è il numero più adeguato di gruppi.

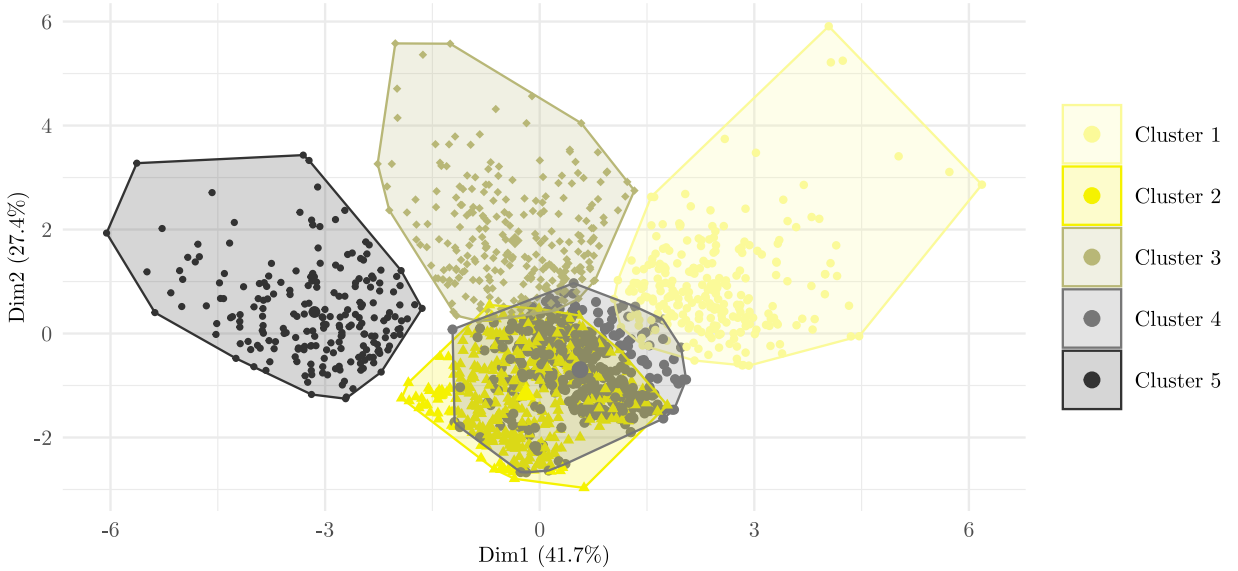
```
set.seed(1234)
list_of_output = vector("list")
list_of_df = vector("list")
var_expl = vector()
tot_within = vector()
i = 1
n_max = 10
for(n_groups in 1:n_max) {
  list_of_output[[i]] = kmeans(cars2,centers = n_groups,nstart = 1000,iter.max = 100)
  temp = cars2
  temp$ID_Group = list_of_output[[i]]$cluster
  list_of_df[[i]] = temp
  var_expl[i] = list_of_output[[i]]$betweenss/list_of_output[[i]]$totss
  tot_within[i] = list_of_output[[i]]$tot.withinss
  i = i + 1
}
names(list_of_df) = 1:n_max
names(list_of_output) = 1:n_max
names(var_expl) = 1:n_max

var_expl_df = data.frame(
  n_groups = names(var_expl),
  var_expl = unlist(var_expl))
var_expl_df$n_groups = as.numeric(as.character(var_expl_df$n_groups))
```

```r
ggplot(var_expl_df, aes(x = n_groups, y = var_expl)) +
  geom_line(col = "#333333",size = 0.8) +
  geom_point(shape = 21,color="#333333",fill = "#F5F200",size = 3) +
  labs(x = "Numero di Gruppi", y = "Varianza Spiegata")+
  theme_minimal()+
  theme(axis.text = element_text(size = 10),
        axis.title = element_text(size = 10),
        text = element_text(family = "CMUSerif"))
```




Lo *screeplot* in questa situazione non è molto d'aiuto, infatti non è evidente la presenza di un gomito. Potremmo quindi sfruttuare i risultati ottenuti tramite i metodi gerarchici e selezionare un numero di gruppi pari a 5. Rappresentiamo graficamente come le unità sono suddivise in un piano a due dimensioni, ottenuto utilizzando l'analisi delle componenti principlali.

```r
cars2$cluster = list_of_output[[5]]$cluster
fviz_cluster(list_of_output[[5]],cars2,geom = "point",main = "")+
  scale_fill_manual(values = c("#FBFA99","#F5F200","#B8B777","#777777","#333333"),
                    labels = paste("Cluster",1:5,sep = " "))+
  scale_colour_manual(values = c("#FBFA99","#F5F200","#B8B777","#777777","#333333"),
                      labels = paste("Cluster",1:5,sep = " "))+
  scale_shape_manual(labels = rep(" ",5),values = c(16:(16 + 5)),guide = "none")+
  labs(paste("Cluster",1:5,sep = " "))+
  theme_minimal()+
  theme(axis.text.x = element_text(size = 10),
        axis.ticks.x = element_blank(),
        axis.text.y = element_text(size = 10),
        axis.title = element_text(size = 10),
        text = element_text(family = "CMUSerif"),
        legend.title = element_blank(),
        legend.text = element_text(size = 10),
        legend.key.size = unit(1,'cm'))
```




Dal grafico si nota una grande sovrapposizione tra i *cluster* 2 e 4.

Per cercare di interpretare le due dimensioni effettuiamo l'analisi delle componenti principali e vediamo quali variabili sono rappresentate dalle prime due componenti.

```
prcomp(cars2[,-7])
```

```
## Standard deviations (1, .., p=6):
## [1] 1.6185877 1.3854426 0.9022415 0.5152510 0.4606638 0.4110816
##
## Rotation (n x k) = (6 x 6):
##                        PC1         PC2         PC3         PC4         PC5
## prezzo_auto      0.5695319 -0.02665187  0.22959085  0.08367238 -0.05294089
## anno_prod        0.4390948 -0.42593625  0.05533174 -0.55641314 -0.45779303
## chilometraggio -0.3341466  0.54045914 -0.15142279 -0.45239242 -0.51634393
## cv               0.3730302  0.50313474  0.14956559  0.50164308 -0.43400094
## peso             0.3555952  0.52127415  0.12370245 -0.47003544  0.57460325
## optional         0.3249241  0.03001009 -0.94001352  0.07851977  0.04986017
##                         PC6
## prezzo_auto      0.78256461
## anno_prod       -0.32178008
## chilometraggio   0.31945442
## cv              -0.38122358
## peso            -0.18820409
## optional         0.03531131
```

La prima componente principale rappresenta maggiormente il prezzo dell'auto, l'anno di produzione e il numero di optional mentre la seconda include il chilometraggio, i cavalli, il peso e l'anno di produzione (con segno negativo). Resta comunque complesso fornire un'interpretazione del grafico appena realizzato ma ci aiuta a visualizzare come le unità sono state suddivise.

Valutiamo anche in questo caso le caratteristiche dei 5 *cluster* ottenuti.

```
hc5_k = list_of_output[["5"]][["cluster"]]
hc5_k.means = cars2[,-7] %>%
  aggregate(. ~ hc5_k,mean)
hc5_k.means
```

```
##   hc5_k prezzo_auto  anno_prod chilometraggio          cv       peso
## 1     1   1.3272522  0.3734250     -0.2894323  1.19740440  1.2389196
## 2     2  -0.2893352  0.4380180     -0.5038684 -0.66866580 -0.7362442
## 3     3  -0.3909576 -0.7403855      1.1938671  0.68024054  0.9012156
## 4     4   0.4604201  0.6218264     -0.6206307 -0.02193167 -0.1266537
## 5     5  -1.3882530 -1.6190122      1.0725108 -0.87424353 -0.8596332
##        optional
## 1  0.2852806785
## 2 -0.6092604862
## 3 -0.0006519656
## 4  0.8803922389
## 5 -0.8508385645
```

- Il primo gruppo riporta solamente per chilometraggio un valore inferiore alla media. Analogamente al quarto *cluster* ottenuto tramite i metodi gerarchici, possiamo ritenere che le auto appartenenti a questo gruppo siano di alta gamma, come nuovi SUV o auto sportive;
- Il secondo gruppo presenta, così come il primo *cluster* nei metodi gerarchici, solamente per l'anno di produzione un valore superiore alla media. Le auto appartenenti a questo gruppo saranno quindi economiche, non molto potenti, con dotazione basica ma di recente produzione;
- Il terzo gruppo comprende auto con un prezzo e un anno di produzione inferiori alla media mentre per le restanti, eccetto *optional* che ha una media prossima allo zero, un valore superiore. Possiamo allora pensare che il terzo *cluster* raggruppi, ad esempio, SUV o auto di fascia media usate, considerato che il prezzo e l'anno di produzione sono più ridotti e le prestazioni, invece, sono migliori della media.
- Il quarto gruppo presenta caratteristiche opposte al terzo. Avendo un prezzo e un anno di produzione inferiori alla media mentre un valore di chilometraggio inferiore si può ritenere che in questo *cluster* siano presenti auto nuove caratterizzate, inoltre, da molti *optional*. È però da sottolineare che non si tratta di auto molto potenti e pesati. Potrebbero rientrare in questa categoria auto berline con un *focus* maggiore sui *comfort* piuttosto che nelle prestazioni;
- Il quinto gruppo presenta solamente per l'anno di produzione un valore superiore alla media. Potrebbero essere auto di fascia media relativamente nuove, con un chilometraggio moderato e un prezzo leggermente inferiore alla media.

Come emerso dal grafico in cui le unità venivano suddivise nei cinque gruppi formati nel piano delle prime due componenti principali, possiamo vedere che i *cluster* 3, 4 e 5 abbiamo molte caratteristiche in comune. Si potrebbe dunque pensare di aggregare questi tre gruppi in modo da raggruppare auto di fascia medio-alta.

Passiamo alle procedure di *clustering* basate sul modello. Stimiamo diverse misture gaussiane con un numero di componenti che va da 1 a 5. Non consideriamo un numero superiore di componenti per parsimonia.

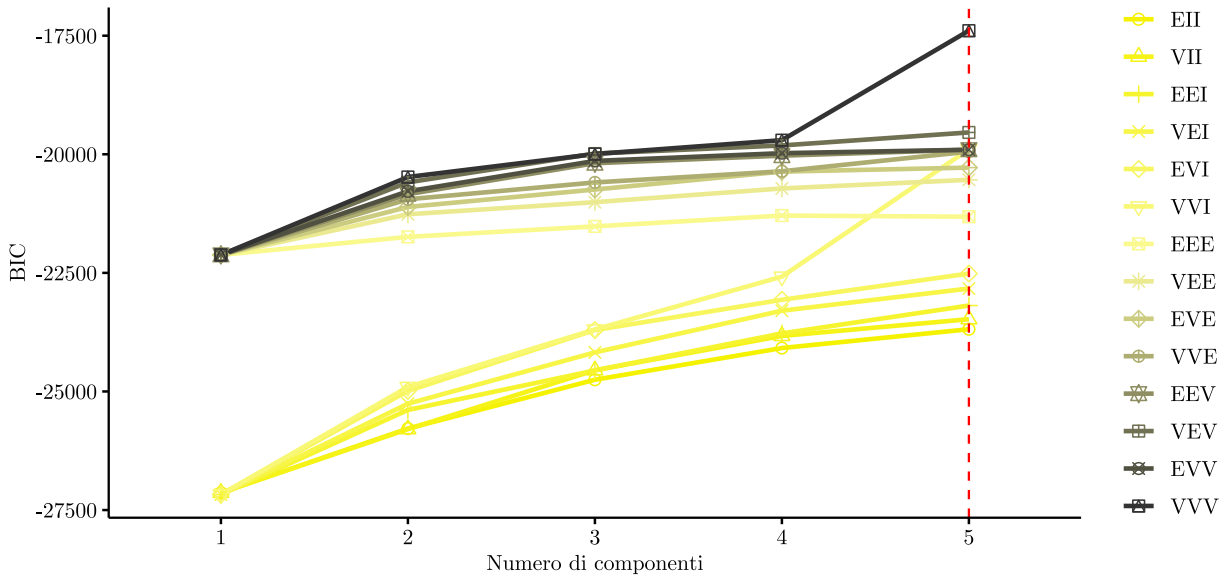
```
library(mclust)
mbc = Mclust(cars2[,-7],G = 1:5,verbose = F)
summary(mbc)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 5
## components:
##
##  log-likelihood    n  df       BIC      ICL
##       -8184.647 1591 139 -17394.02 -17681.1
##
## Clustering table:
##   1   2   3   4   5
## 408 139 757 115 172
```

Vengono selezionate 5 componenti, ossia vengono individuati 5 gruppi. Dal *summary* possiamo poi notare che è stato selezionato il modello più flessibile (VVV), che prevede una forma ellissoidale delle componenti, con volume, forma e orientamento differente per ognuna di esse.

La struttura di matrice di varianze e covarianze delle componenti viene selezionata mediante l'indicatore BIC. Possiamo visualizzare l'andamento dell'indicatore di informazione automatica per le 14 possibili strutture di matrice di varianze e covarianze all'aumentare del nuemro di componenti.

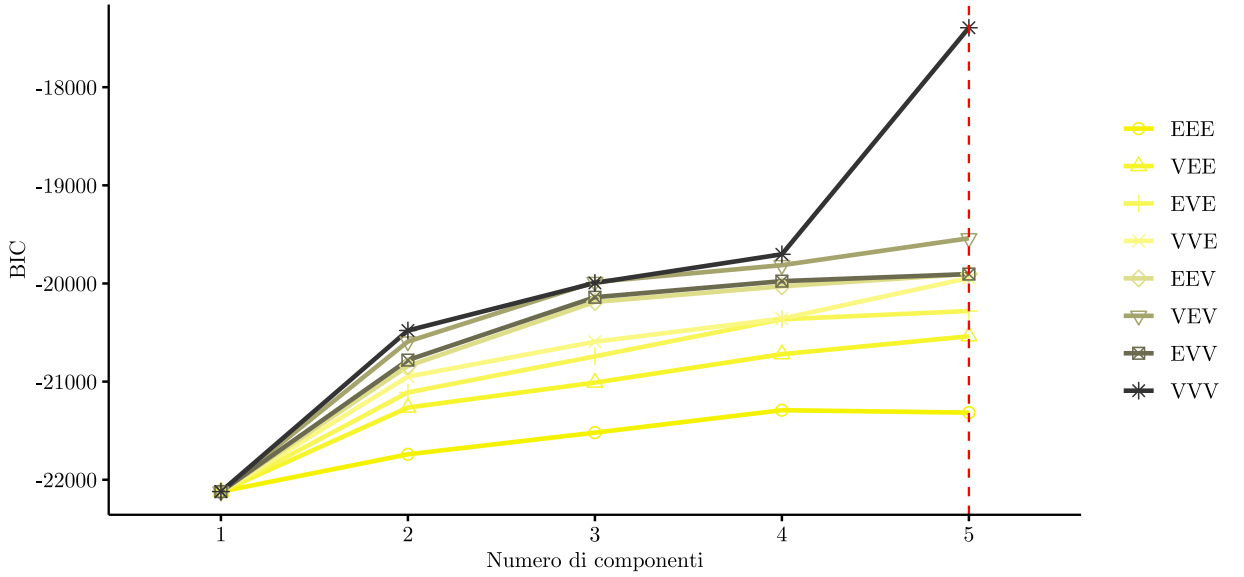
```
fviz_mclust_bic(mbc,legend = "right",shape = "model",size = 1,
                palette = palette_function(14))+
  labs(x = "Numero di componenti")+
  theme(legend.text = element_text(size = 10),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 10),
        title = element_blank(),
        text = element_text(family = "CMUSerif"))
```



Il BIC è riportato con segno negativo, cerchiamo quindi la struttura che massimizza l'indicatore.

Evidenziando due fascie di strutture separate, si può pensare di visualizzare solamente quelle con valore del BIC maggiore. Ristimiamo allora la mistura considerando solamente queste strutture per la matrice di varianze e covarianze delle componenti.

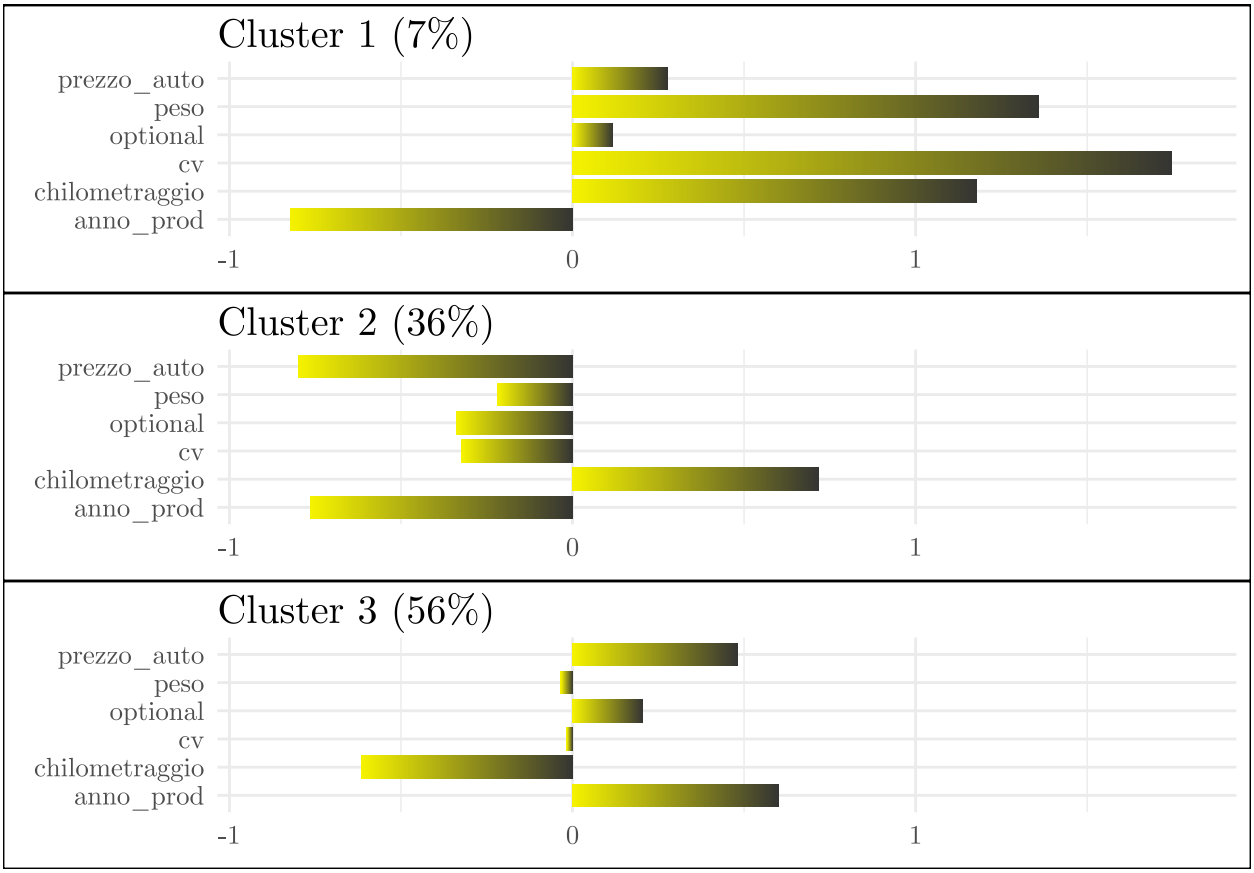
```
var_type = colnames(as.matrix(mbc[["BIC"]] > -2500))[7:14]
mbc2 = Mclust(cars2[,-7],G = 1:5,modelNames = var_type,verbose = F)
fviz_mclust_bic(mbc2,legend = "right",shape = "model",size = 1,
                palette = palette_function(8))+
  labs(x = "Numero di componenti")+
  theme(legend.text = element_text(size = 10),
        axis.title = element_text(size = 10),
        axis.text = element_text(size = 10),
        title = element_blank(),
        text = element_text(family = "CMUSerif"))
```




Il modello selezionato è, come già specificato, un modello VVV con 5 componenti. Nonostante il grande guadagno che si ottiene passando da 4 a 5 gruppi, il modello selezionato, essendo molto flessibile nelle sue caratteristiche, riporta un grande numero di parametri. Potremmo provare a ridurre il numero di componenti a 3, dato che l'aumento del BIC passando da 3 a 4 gruppi non è elevato. Inoltre, in corrispondenza di tre componenti si nota che il modello VEV che, a differenza di VVV, prevede un uguale forma per le componenti, riporta un valore prossimo a quello ottenuto quest'ultimo.

Proviamo dunque a stimare una mistura gaussiana con $G = 3$ componenti e considerando come struttura per la matrice di varianze e covarianze VEV.

```
mbc = Mclust(cars2[,-7],G = 3,modelNames = "VEV",verbose = F)
```




I *cluster* ottenuti mediante il *model-based clustering* presentano le seguenti caratteristiche:

- il primo gruppo, che risulta il meno numeroso, riporta per tutte le variabili, eccetto l'anno di produzione, un valore molto superiore alla media generale. Queste caratteristiche, già riscontrate nei gruppi ottenuti con gli altri metodi considerati, sono tipiche di auto di alta gamma;
- il secondo gruppo presenta solamente per chilometraggio un valore superiore alla media. Possiamo affermare in questo *cluster* siano state raggruppate le auto usate, più vecchie e meno potenti;
- il terzo gruppo ha caratteristiche più diversificate rispetto agli altri due. Il prezzo dell'auto, il numero di *optional* e l'anno di produzione sono superiori alla media mentre il chilometraggio, il peso e i cavalli inferiori, anche se gli ultimi due presentano un valore prossimo allo zero (media generale). Questo *cluster* conterrà quindi auto nuove di fascia media non molto potenti.

In conclusione, sono stati utilizzati diversi metodi per cercare di individuare gruppi all'interno del nostro insieme di dati sulle automobili presenti in *AutoScout24*. Inizialmente, sono state considerate sia variabili quantitative che qualitative legate alla variabile `prezzo_auto` con le quali, sfruttando i metodi gerarchici di *clustering* e il coefficiente di *Gower*, sono stati individuati 3 gruppi. Successivamente si è passati alla considerazione delle sole variabili quantitative legate con il prezzo dell'auto e sono stati utilizzati metodi gerarchici, non gerarchici (*k-means*) e metodi basati sul modello (*model-based clustering*). Partendo dai metodi gerarchici sono stati individuati 5 gruppi e questo risultato è stato sfruttato anche per la scelta del numero di *cluster* nell'applicazione dell'algoritmo *k-means*. Anche con i metodi basati sul modello sono stati identificati 5 gruppi ed è stato selezionato il modello più flessibile tra quelli disponibili. Per parsimonia, si è però deciso di ridurre il numero di componenti e di prediligere un modello con un numero inferiore di parametri.

Risulta chiaro di come le tecniche di *clustering* abbiano comunque una natura esplorativa e ci consentano di sfruttare metodi differenti per cercare di individuare e analizzare gruppi di unità all'interno di un insieme di dati.