Clustering Robusto con Tclust

Costanza Addari, Davide Pirro, Domenico Pierri, Gabriele D'Andrea, Romeo Silvestri

Caricamento delle librerie:

```
library(readxl)
library(SimDesign)
library(mclust)
library(tclust)
library(maptools)
library(ggplot2)
library(tibble)
library(viridis)
```

Confronto Mclust con Tclust

L'obbiettivo di questa prima analisi è quello di confrontare il pacchetto mclust con tclust tramite simulazione e porre in evidenza l'utilità del clustering robusto. Per fare questo, si generano i dati dalle seguenti distribuzioni: 600 osservazioni da una normale bivariata di medie μ=(0,3) e matrice di covarianze identità, 300 osservazioni da una normale bivariata con μ=(2,8) e matrice di covariazne non identità e 100 osservazioni da una uniforme bivariata le cui componenti hanno una relazione di dipendenza cosinusoidale. Di conseguenza, si vuole ricreare il caso in cui vi siano due popolazioni distinte e un insieme di dati che svolgono il ruolo di rumore pari al 10% del dataset

```
set.seed(123)
Y=runif(100,0,20)
X=15+2*cos(Y)
XY=cbind(X,Y)

sigma1=diag(2)
sigma2=matrix(c(3,-2,-2,3),2,2)

mixt=rbind(rmvnorm(600,mean=c(0,3),sigma=sigma1),rmvnorm(300,mean=c(2,8),sigma=sigma2),XY)
```

Si stimano quindi i modelli a mistura finita con componenti Gaussiane tramite mclust e si seleziona il modello migliore attraverso il Criterio di Informazione Bayesiana (BIC). Per simulare una situazione reale e, consci del vero numero di cluster, questo viene fissato a priori pari a k=2. La consistenza delle stime vengono inoltre assicurate da una inizializzazione dell'algoritmo tramite agglomerazioni casuali:

```
fit_mm=mclustBIC(mixt,G=2,verbose=FALSE,initialization=list(hcRandomPairs(mixt)))
summary(fit_mm,mixt)
```

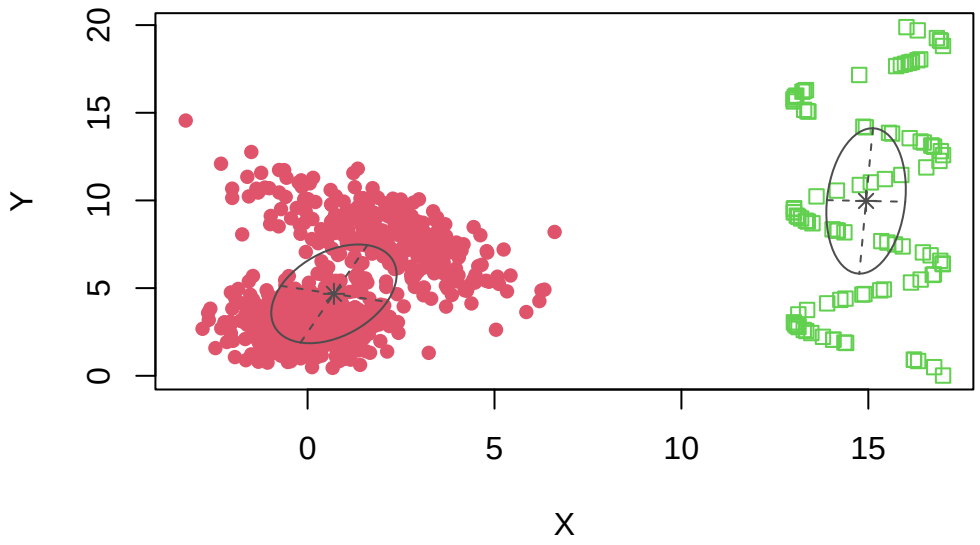
```
## Best BIC values:
##             EVV,2       VVE,2     VVV,2
```

```
## BIC      -9337.52 -9365.27963 -9370.878
## BIC diff     0.00   -27.75967   -33.358
##
## Classification table for model (EVV,2):
##
##   1   2
## 900 100
```

Il miglior modello risulta essere a distribuzione elissoidale, volume uguale e con forma e orientamento degli assi variabile (EVV). I 2 cluster così trovati sono composti rispettivamente da 900 osservazioni il primo e da 100 il secondo.

Si mostra di seguito il relativo grafico:

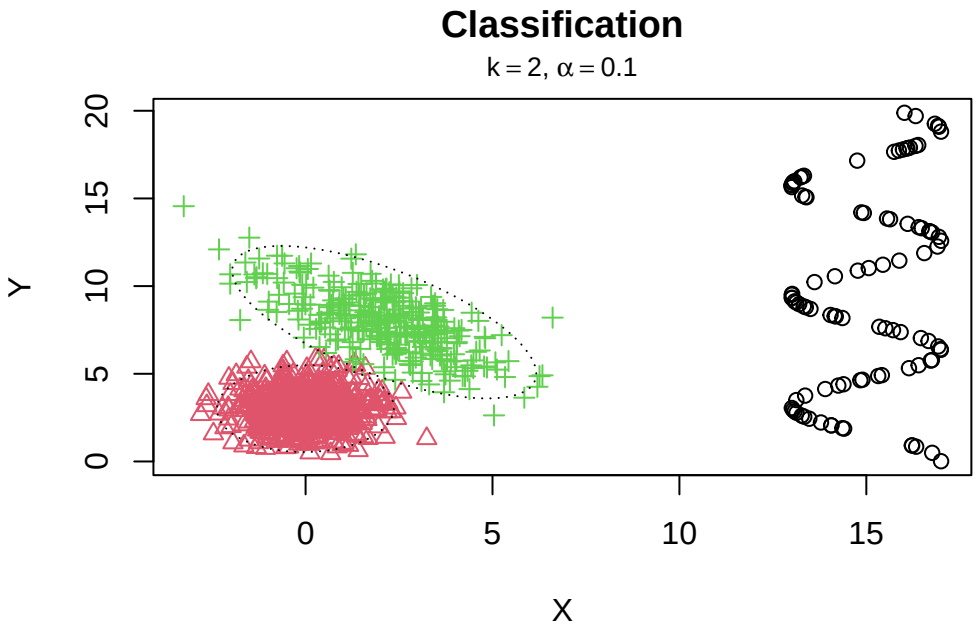
```
fit_mclust=Mclust(mixt,verbose=FALSE,modelNames="EVV",G=2)
plot(fit_mclust,what="classification",col=2:3)
```



In questo caso specifico si può osservare che i due veri cluster non vengono identificati distintamente l'uno con l'altro a causa della presenza di dati spuri. Paradossalmente, invece, l'insieme dei dati spuri costituisce un cluster a parte.

Si applica ora un modello robusto per dati spuri con il pacchetto tclust. Si imposta anche in questo caso un numero di cluster k=2. Il livello di trimming α viene posto a 0.1 (si tagliano il 10% dei dati) e il vincolo è di tipo "eigen". Gli altri parametri vengono settati di default:

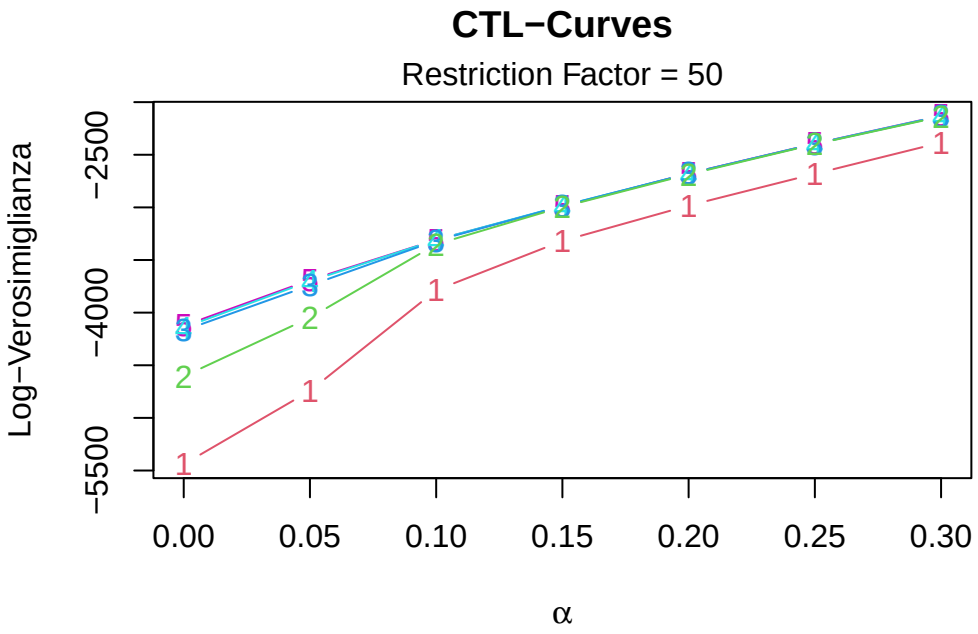
```
fit_tclust=tclust(mixt,k=2,alpha=0.1,restr="eigen")
plot(fit_tclust)
```



Graficamente si può vedere che questa volta i due diversi cluster sono stati identificati correttamente, mentre l'insieme dei dati spuri è stato escluso dai raggruppamenti prima dell'analisi.

Per verificare che, la scelta del numero di cluster k e il livello di trimming α, sia stata corretta, si mostrano le curve di verosimiglianza trimmate:

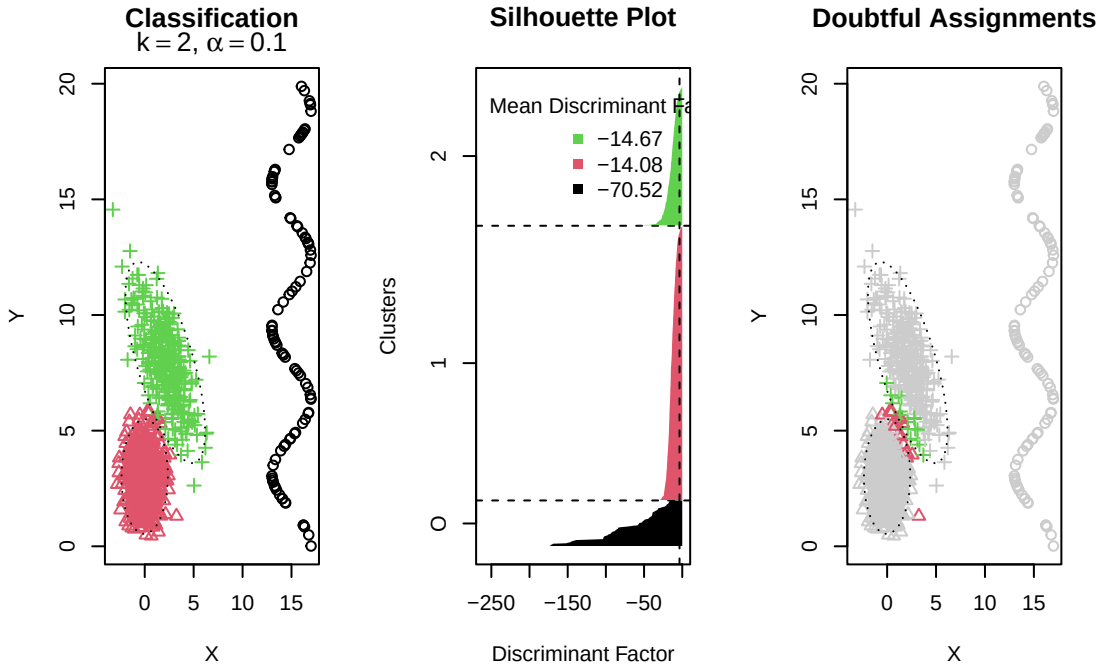
```
plot(ctlcurves(mixt,k=1:5,alpha=seq(0,0.3,by=0.05),trace=0),ylab="Log-Verosimiglianza")
```



Da questo grafico si può chiaramente vedere che la scelta di un numero di cluster pari a 2 con un livello di trimming di 0.1 risulti una adeguata per il problema. Per k=2, infatti, non si osserva alcun guadagno di verosimiglianza con l'aumentare di k e a parità di α, a partire da α=0.1.

Si esegue anche un'analisi grafica tramite Silhouette come diagnostica del modello:

```
discr_tclust=DiscrFact(fit_tclust,threshold=0.01)
plot(discr_tclust)
```



Fissata una soglia pari a -2, i cluster risultano tutti molto significativi. I valori delle medie dei fattori discriminanti nei cluster sono molto bassi, a simboleggiare la forte omogeneità delle unità appartenenti allo stesso gruppo. Si nota, inoltre, che l'insieme dei dati spuri sono ben identificati (fattori discriminati bassissimi). Nel grafico di destra sono mostrate anche le unità con assegnazioni dubbie, le quali si trovano sulla frontiera tra i due cluster.

Clustering delle Province Italiane al 2019

In questa applicazione con dati reali, si vogliono applicare le metodologie del clustering robusto. Il dataset selezionato si riferisce alle osservazioni sulle 107 province italiane al 2019 di 8 diversi indicatori demografici (variabili) rilevati dall'Istat.

Le variabili sono:

- tasso di natalità
- tasso di mortalità
- tasso di nuzialità
- saldo migratorio totale
- numero medio di figli per donna
- speranza di vita alla nascita
- indice di vecchiaia
- età media

La scelta di queste variabili è determinata dall'esigenza di esprimere i diversi aspetti che caratterizzano maggiormente l'andamento demografico di una popolazione. I 5 fenomeni che si vogliono studiare sono: natalità, mortalità, struttura per età, migratorietà e nuzialità. Fondamentale risulta il fatto che, i primi tre fenomeni elencati sono i più importanti nel riassumere il comportamento demografico di una popolazione stabile. Di conseguenza, per questi fenomeni si fa uso di due indicatori ciascuno. In contrapposizione, migratorietà e nuzialità sono rappresentati da un solo indicatore.

L'obbiettivo dell'analisi è quello di raggruppare le province per andamento demografico. In questo senso, province in uno stesso cluster avranno tra loro un andamento demografico simile e, distinto dalle province appartenenti ad altri cluster.

Analisi Esplorativa

Si carica il dataset degli indicatori dell'Istat e si mostrano le statistiche descrittive più rilevanti (quartili, mediana, media, valore massimo e valore minimo) per ciascuna variabile:

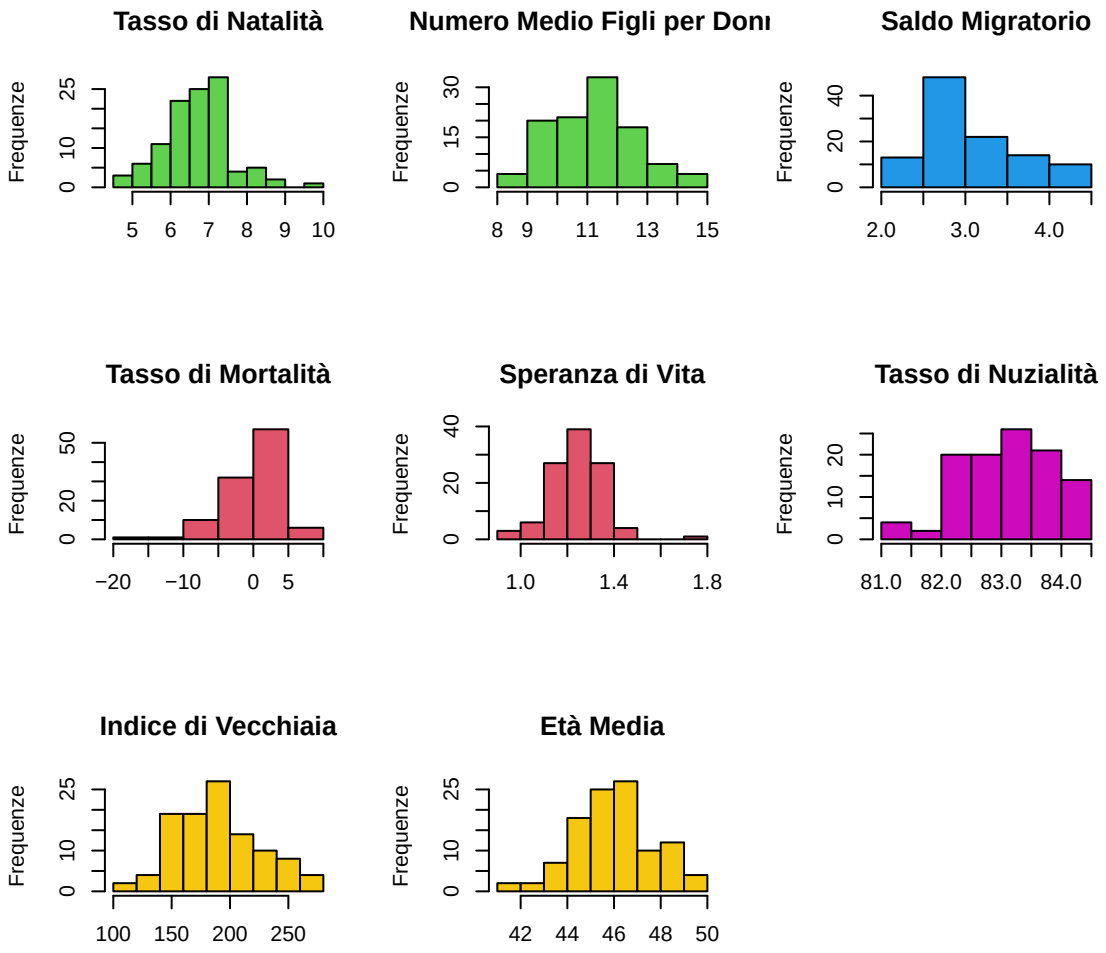
```
istat=as.data.frame(read_excel("istat.xlsx"))
dati=istat[,-1]

summary(dati)
```

```
##  tasso di natalità tasso di mortalità tasso di nuzialità
##  Min.   :4.900     Min.   : 8.40      Min.   :2.200
##  1st Qu.:6.200     1st Qu.:10.15      1st Qu.:2.700
##  Median :6.700     Median :11.30      Median :3.000
##  Mean   :6.737     Mean   :11.23      Mean   :3.121
##  3rd Qu.:7.300     3rd Qu.:12.10      3rd Qu.:3.500
##  Max.   :9.800     Max.   :14.70      Max.   :4.500
##  saldo migratorio totale numero medio di figli per donna
##  Min.   :-18.70000       Min.   :0.92
##  1st Qu.: -2.90000       1st Qu.:1.17
##  Median :  0.80000       Median :1.24
##  Mean   : -0.01495       Mean   :1.24
##  3rd Qu.:  2.75000       3rd Qu.:1.32
##  Max.   :  8.90000       Max.   :1.71
##  speranza di vita alla nascita indice di vecchiaia   età media
```

```
##  Min.   :81.10                Min.   :116.4       Min.   :41.80
##  1st Qu.:82.65                1st Qu.:164.7       1st Qu.:44.95
##  Median :83.20                Median :185.0       Median :46.00
##  Mean   :83.16                Mean   :190.5       Mean   :46.05
##  3rd Qu.:83.70                3rd Qu.:209.3       3rd Qu.:47.00
##  Max.   :84.50                Max.   :267.7       Max.   :49.40
```

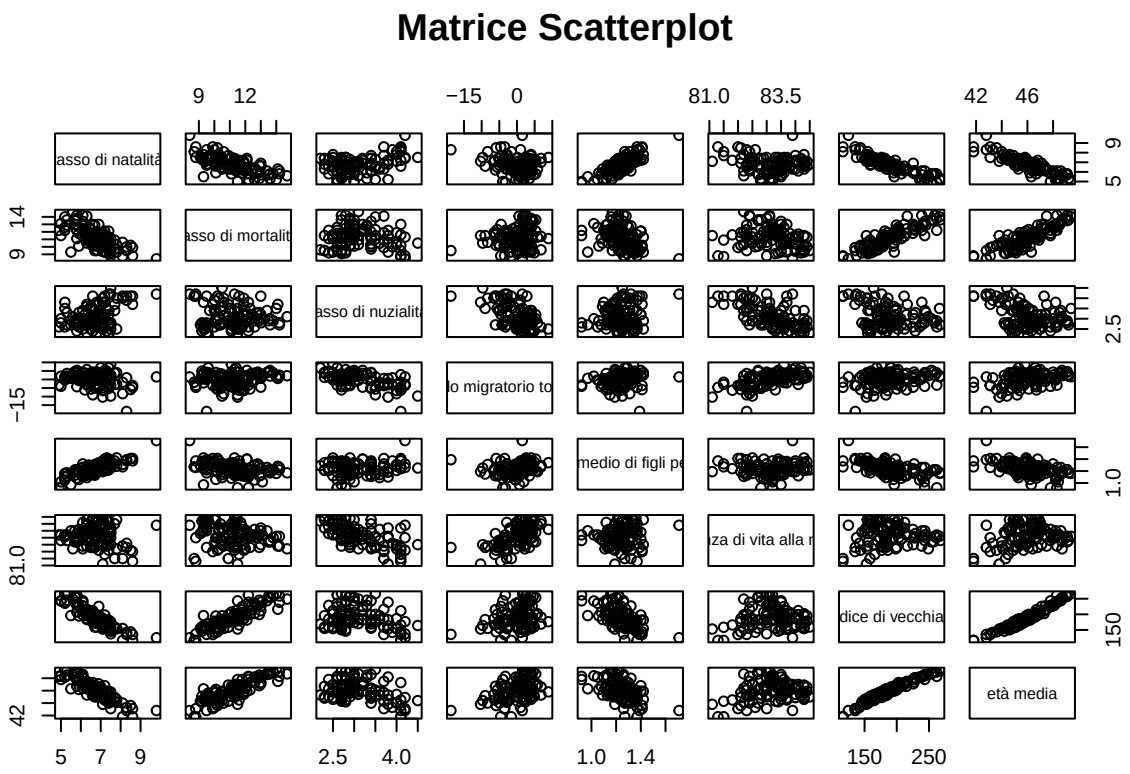
Per descrivere meglio la distribuzione delle variabili, vengono rappresentati anche i relativi istogrammi di frequenza:

```
par(mfrow=c(3,3))
hist(dati[,1],col=3,xlab="",ylab="Frequenze",main="Tasso di Natalità")
hist(dati[,2],col=3,xlab="",ylab="Frequenze",main="Numero Medio Figli per Donna")
hist(dati[,3],col=4,xlab="",ylab="Frequenze",main="Saldo Migratorio")
hist(dati[,4],col=2,xlab="",ylab="Frequenze",main="Tasso di Mortalità")
hist(dati[,5],col=2,xlab="",ylab="Frequenze",main="Speranza di Vita")
hist(dati[,6],col=6,xlab="",ylab="Frequenze",main="Tasso di Nuzialità")
hist(dati[,7],col=7,xlab="",ylab="Frequenze",main="Indice di Vecchiaia")
hist(dati[,8],col=7,xlab="",ylab="Frequenze",main="Età Media")
par(mfrow=c(1,1))
```



Si esegue ora una breve analisi descrittiva bivariata mostrando la matrice scatterplot. Questo insieme di diagrammi risulta utile per individuare la distribuzione congiunta per coppie di variabili. Alla matrice si affianca, inoltre, la tabella riassuntiva che riporta i valori dei coefficienti di correlazione lineare

```
pairs(~`tasso di natalità`+`tasso di mortalità`+`tasso di nuzialità`+
        `saldo migratorio totale`+`numero medio di figli per donna`+
        `speranza di vita alla nascita`+`indice di vecchiaia`+`età media`,
      data=dati,main="Matrice Scatterplot")
```




```
round(cor(dati),2)
```

```
##                                 tasso di natalità tasso di mortalità
## tasso di natalità                            1.00               -0.68
## tasso di mortalità                          -0.68                1.00
## tasso di nuzialità                           0.41               -0.08
## saldo migratorio totale                     -0.22                0.17
## numero medio di figli per donna              0.81               -0.38
## speranza di vita alla nascita               -0.13               -0.22
## indice di vecchiaia                         -0.89                0.87
## età media                                   -0.88                0.85
##                                 tasso di nuzialità saldo migratorio totale
## tasso di natalità                             0.41                   -0.22
## tasso di mortalità                           -0.08                    0.17
## tasso di nuzialità                            1.00                   -0.56
## saldo migratorio totale                      -0.56                    1.00
```

```
## numero medio di figli per donna              0.18                    0.18
## speranza di vita alla nascita               -0.62                    0.58
## indice di vecchiaia                         -0.31                    0.27
## età media                                   -0.43                    0.43
##                                 numero medio di figli per donna
## tasso di natalità                                          0.81
## tasso di mortalità                                        -0.38
## tasso di nuzialità                                         0.18
## saldo migratorio totale                                    0.18
## numero medio di figli per donna                            1.00
## speranza di vita alla nascita                              0.08
## indice di vecchiaia                                       -0.59
## età media                                                 -0.51
##                                 speranza di vita alla nascita
## tasso di natalità                                       -0.13
## tasso di mortalità                                      -0.22
## tasso di nuzialità                                      -0.62
## saldo migratorio totale                                  0.58
## numero medio di figli per donna                          0.08
## speranza di vita alla nascita                            1.00
## indice di vecchiaia                                      0.06
## età media                                                0.21
##                                 indice di vecchiaia età media
## tasso di natalità                             -0.89     -0.88
## tasso di mortalità                             0.87      0.85
## tasso di nuzialità                            -0.31     -0.43
## saldo migratorio totale                        0.27      0.43
## numero medio di figli per donna               -0.59     -0.51
## speranza di vita alla nascita                  0.06      0.21
## indice di vecchiaia                            1.00      0.97
## età media                                      0.97      1.00
```

Analizzando i risultati trovati, si nota che molte variabili hanno una moderata o forte relazione lineare tra di loro. Questo avviene soprattutto se le variabili vanno a descrivere fenomeni affini. In particolare, si evidenzia che i tassi di natalità e di mortalità hanno una forte correlazione con l'indice di vecchiaia e con l'età media della popolazione anche se di segno opposto. Una strettissima correlazione lineare positiva (0.97) si ha tra l'indice di vecchiaia e l'età media. Si potrebbe pensare di eliminare una delle due variabili, ma ciò inficerebbe sulla logica con cui le variabili sono state selezionate: si vuole rappresentare il fenomeno della struttura per età con uguale intensità rispetto al fenomeno della natalità e della mortalità e, oltretutto, si vuole tenere conto che i due indici in questione definiscono aspetti diversi dello stesso fenomeno.

Per tenere conto della diversa variabilità e scala di misura delle variabili, queste si standardizzano in fase preliminare all'analisi. Questo implica che le variabili verranno messe tutte sullo stesso piano e avranno uguale rilevanza nell'applicazione del modello

```
dati_scale=scale(dati,center=TRUE,scale=TRUE)
```

Applicazione del Modello con Tclust

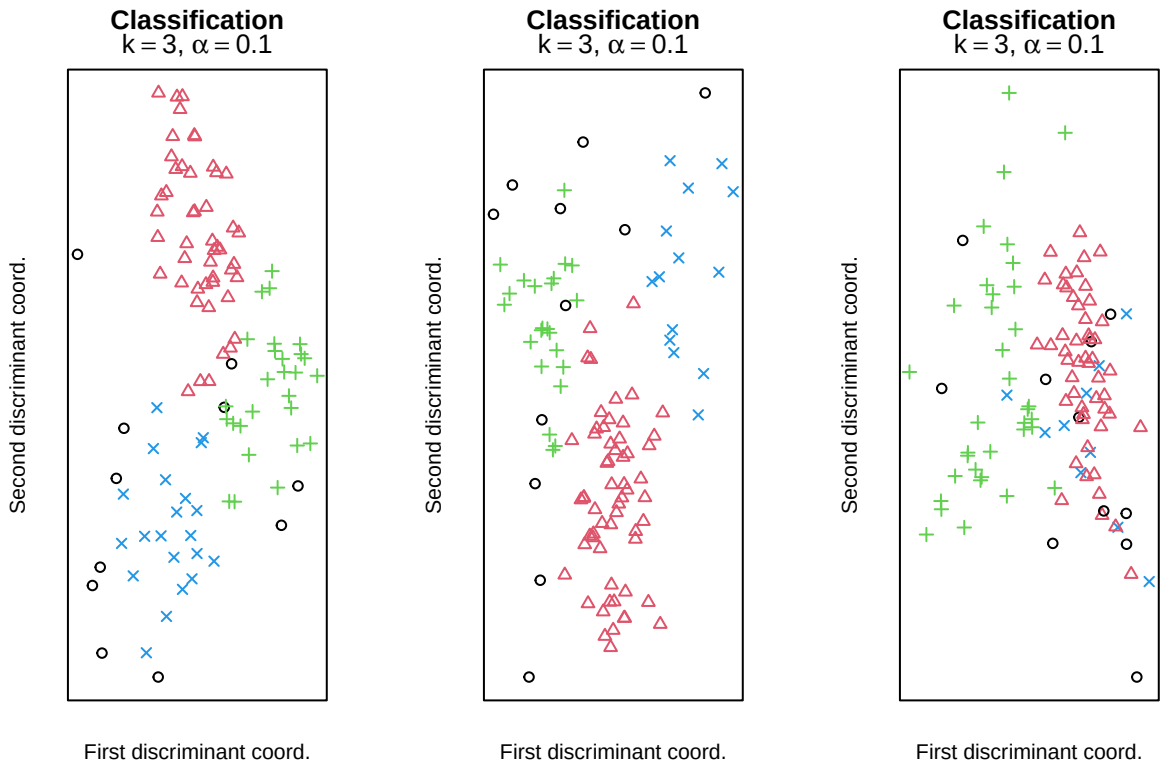
Si applica ora il modello di clustering per dati spuri attraverso il pacchetto tclust. Inizialmente il numero di cluster viene fissato a 3 e il livello di trimming a 0.1 per i tre diversi tipi di vincoli: "eigen", "sigma" e "deter". Questi parametri sono stati scelti per seguire la ripartizione geografica italiana: Nord, Centro e Sud; inoltre, un taglio del 10% sembra essere sufficientemente bilanciato. In ogni caso questi valori servono soltanto in

un primo momento per confrontare i vincoli e verranno verificati successivamente. Gli altri parametri sono invece quelli di default della funzione tclust:

```
set.seed(123)
fit_eigen=tclust(dati_scale,k=3,alpha=0.1,restr="eigen",warning=2)
fit_sigma=tclust(dati_scale,k=3,alpha=0.1,restr="sigma",warning=2)
fit_deter=tclust(dati_scale,k=3,alpha=0.1,restr="deter",warning=2)
```

Per confrontare i diversi vincoli, inizialmente si analizzano graficamente i 3 relativi diagrammi che mostrano i cluster così realizzati:

```
par(mfrow=c(1,3))
plot(fit_eigen)
plot(fit_sigma)
plot(fit_deter)
```




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

Dai grafici si può vedere che i primi due modelli formano dei cluster ben distinti tra loro. Per quanto riguarda i valori tagliati, sembra che il modello con restrizione "eigen" sia più preciso rispetto a quello con restrizione "sigma". Qualche valore tagliato nel secondo modello, infatti, si trova dentro ad uno specifico cluster senza ambiguità. Il terzo modello, per ultimo, sembra sbagliare completamente la classificazione, tanto che due dei cluster si sovrappongono completamente. Questo fatto può essere dovuto ad una errato numero di cluster e/o di livello di trimming associato al modello con restrizione "deter".

Per verificare numericamente la nostra intuizione grafica, si utilizza il Criterio di Informazione Bayesiana (BIC). Secondo questo criterio, minore è il valore assunto dal BIC, migliore risulterà il modello di clustering applicato ai dati osservati. Non essendo però presente nel pacchetto tclust una specifica funzione per il calcolo del BIC, la si implementa manualmente. Si confrontano allora i valori del BIC di ognuno dei 3 modelli:

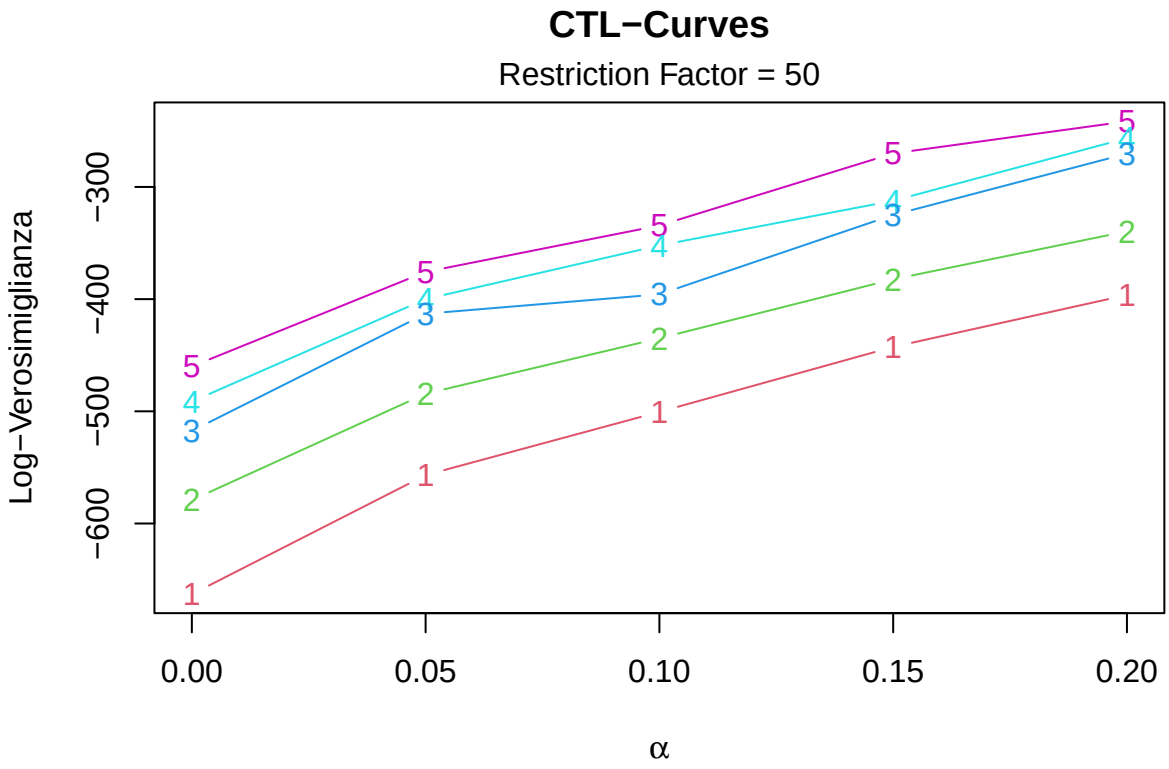
```
bic.tclust=function(x,data){    #funzione per il calcolo del BIC
  p=nrow(x$centers)  #numero di variabili
  n=length(x$cluster)-sum(x$cluster==0)  #numero di osservazioni (province)
  k=ncol(x$centers)  #numero di cluster
  twss=0  #somma delle varianze entro i gruppi
  dataset=cbind(as.data.frame(data),x$cluster)
  for(i in 1:k){
    clus=dataset[dataset$`x$cluster`==i,-c(length(dataset))]
    varianza=apply(clus,2,var)*(nrow(clus)-1)
    somma=sum(varianza)
    twss=twss+somma
  }
  bic=twss+log(n)*p*k
  return(bic)
}

bic_fit=cbind(bic.tclust(fit_eigen,dati_scale),bic.tclust(fit_sigma,dati_scale),
              bic.tclust(fit_deter,dati_scale))
colnames(bic_fit)=c("Eigen","Sigma","Deter")
bic_fit
```

```
##        Eigen    Sigma    Deter
## [1,] 382.062 500.8624 608.6205
```

Come ci si aspettava, il modello con restrizione "eigen" risulta il migliore, seguito da quello con restrizione "sigma" e, infine, da quello con "deter". Per un'analisi più accurata e per la ricerca del miglior modello possibile, si potrebbe mettere a confronto ciascun modello con diverse scelte di k e alpha. Ciò risulterebbe d'altra parte fuorviante per l'obbiettivo che si è posti all'inizio. Si decide quindi di proseguire con la valutazione di un modello con restrizione di tipo "eigen".

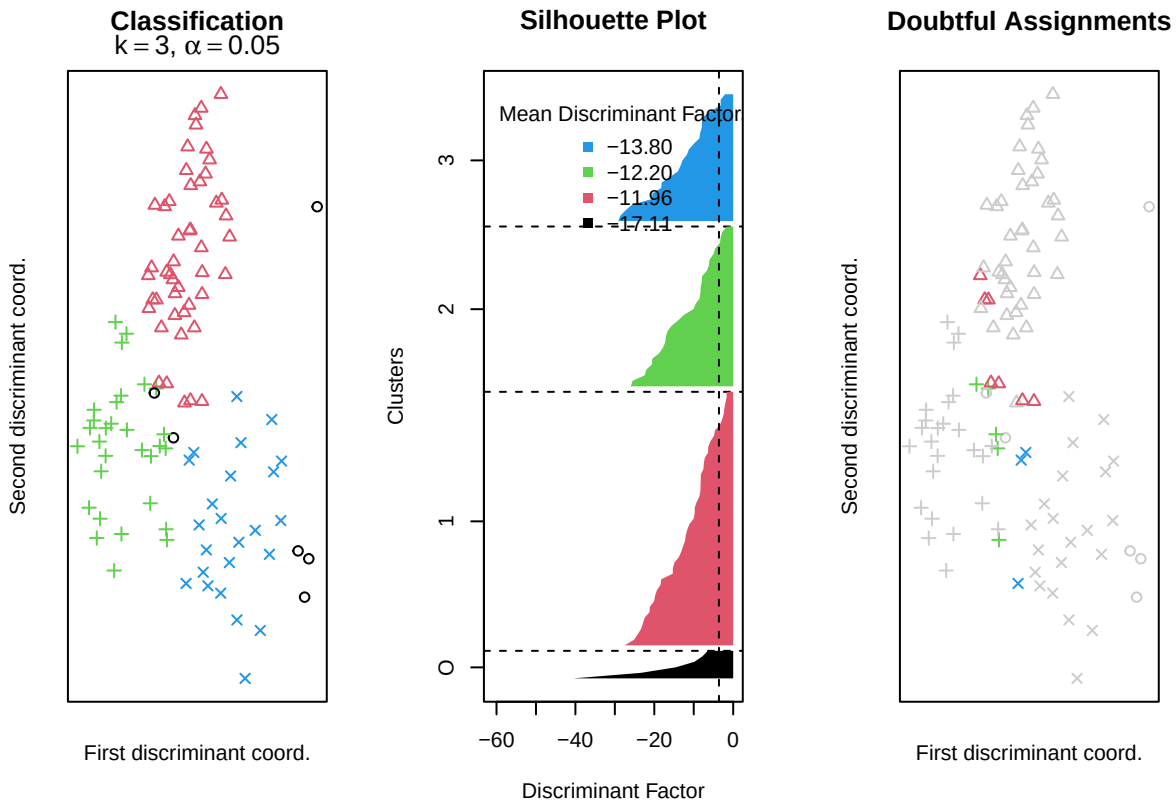
Per scegliere un adeguato numero di cluster k e un conseguente corretto livello di trimming α, si analizzano le seguenti curve di verosimiglianza trimmate:

```
plot(ctlcurves(dati_scale,k=1:5,alpha=seq(0,0.2,by=0.05),trace=0),ylab="Log-Verosimiglianza")
```



Si vuole scegliere il più piccolo numero di cluster k, tale che il guadagno di verosimiglianza rispetto a k+1 cluster tende a 0 per un livello di trimming α sufficientemente elevato. In questo caso allora, una scelta di k=3 e α=0.15 sembrerebbe la migliore. Si hanno dubbi invece per una scelta di k=3 e α=0.05, in quanto il guadagno di verosimiglianza in quel caso è prossimo a 0, ma ciò non sembra valere per un α pari a 0.1. I valori delle curve di verosimiglianza sono in ogni caso soggetti a fattori aleatori e dipendono strettamente dal seme che si è fissato. Per questo motivo e, poichè un livello di trimming di 0.15 sembra eccessivo, si vuole verificare il clustering per k=3 e α=0.05. Come diagnostica si utilizza il grafico a Silhouette con i relativi valori medi dei fattori discriminanti associati ad ogni cluster individuato:

```
fit=tclust(dati_scale,k=3,alpha=0.05,warning=2)
discr_fit=DiscrFact(fit,threshold=0.01)
plot(discr_fit)
```

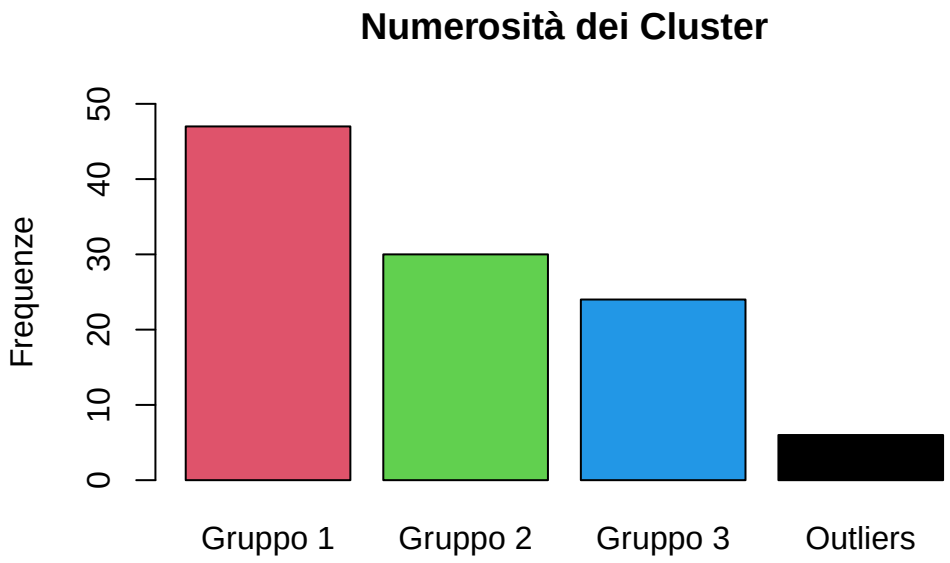


Fissando una soglia pari a -2, tutti i cluster risultano molto significativi. Particolare attenzione assume la media dei fattori discriminanti dei dati spuri: il valore che assume è molto basso (-17.11), testimoniando che il livello di trimming impostato è adeguato. Da notare anche che le assegnazioni dubbie appartengono in modo omogeneo a tutti i 3 cluster, mentre non ci sono dati spuri con queste caratteristiche. Infine si può vedere dal grafico che tra i 6 dati spuri ci siano 2 bridge point e 4 outliers. Il modello con 3 cluster e il 5% di dati tagliati si adatta quindi molto bene al problema e sarà quello di riferimento per il resto dell'analisi.

Caratterizzazione dei Cluster

Si vuole descrivere in questa sezione i cluster trovati col modello selezionato precedentemente. Si mostra inizialmente la numerosità di ciascun cluster:

```
dati_fin=cbind(istat,fit$cluster)
colnames(dati_fin)[10]="cluster"
clus0=dati_fin[dati_fin$cluster==0,-c(1,10)]
clus1=dati_fin[dati_fin$cluster==1,-c(1,10)]
clus2=dati_fin[dati_fin$cluster==2,-c(1,10)]
clus3=dati_fin[dati_fin$cluster==3,-c(1,10)]

num_cluster=data.frame(cbind(t(fit$size),dim(clus0)[1]))
barplot(as.numeric(num_cluster),names.arg=c("Gruppo 1","Gruppo 2","Gruppo 3","Outliers"),col=c(2,3,4,1)
```



Come si può osservare, i cluster sono in ordine descrescente di numerosità, composti rispettivamente da 47, 30 e 24 cluster. Si calcolano ora le medie delle variabili per ogni cluster:

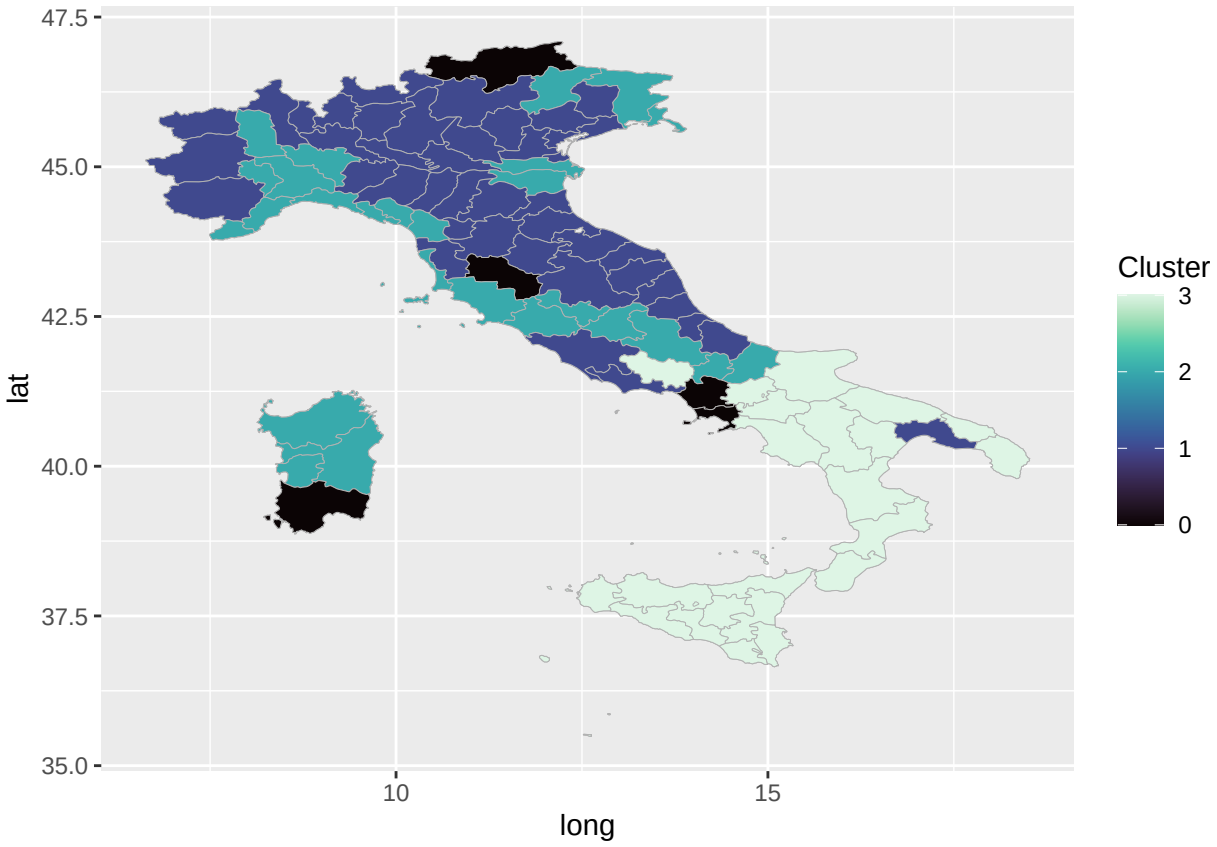
```
clus_mean=data.frame(round(cbind(apply(clus1,2,mean),apply(clus2,2,mean),apply(clus3,2,mean)),2))
colnames(clus_mean)=c("Gruppo 1","Gruppo 2","Gruppo 3")
clus_mean
```

```
##                                 Gruppo 1 Gruppo 2 Gruppo 3
## tasso di natalità                   6.91     5.80     7.30
## tasso di mortalità                 10.68    12.79    10.76
## tasso di nuzialità                  2.78     2.99     3.76
## saldo migratorio totale             2.24     0.80    -4.63
## numero medio di figli per donna     1.28     1.16     1.24
## speranza di vita alla nascita      83.73    82.95    82.43
## indice di vecchiaia               179.40   233.84   167.94
## età media                          45.86    47.94    44.64
```

Il primo cluster è quello più equilibrato tra i tre. Il suo tasso di natalità è nella media, così come gli indicatori relativi alla struttura per età. Il tasso di mortalità e di nuzialità sono bassi, mentre la speranza di vita alla nascita e il numero medio di figli per donna sono relativamente elevati rispetto alla media nazionale. Infine il saldo migratorio totale è molto positivo, segnalando che il fenomeno immigratorio è più intenso di quello emigratorio. Il secondo cluster è composto dalle province più longeve. La natalità è infatti un fenomeno molto limitato, la mortalità è diffusa e l'indice di vecchiaia elevato. Per questo cluster il saldo migratorio risulta leggermente positivo. Il terzo ed ultimo cluster è invece il più giovane e demograficamente attivo. Il tasso di natalità e nuzialità sono infatti elevati e gli indici di struttura per età molto bassi. Il saldo migratorio è invece estremamente negativo indicando delle fortissime emigrazioni per queste province.

Per analizzare in maniera chiara l'assegnazione di un'unità ad un cluster ci si avvale di una cartina geografica delle province italiane. Province appartenenti ad uno stesso cluster sono colorate con lo stesso colore. I colori assegnati ai rispettivi cluster sono presentati nella legenda

```
dati_map=dati_fin[,c(1,10)]
dati_map[c(9,26,41,46,91),1]=c("Aosta","Bolzano-Bozen","Reggio Emilia","Forli'","Reggio Calabria")
dati_map=dati_map[-c(7,8,23,24,25,47,57,64,86,92,93,107),]
dati_map=dati_map[order(dati_map[,1],decreasing=FALSE),]

italy_map=map_data("italy")
clus_prov=tibble(region=sort(unique(italy_map$region)),value=dati_map[,2])

map_istat=ggplot() +
  geom_map(data=italy_map,map=italy_map,aes(long,lat,map_id=region),color="#b2b2b2",size=0.1) +
  geom_map(data=clus_prov,map=italy_map,aes(fill=value,map_id=region),color="#b2b2b2",size=0.1) +
  scale_fill_viridis(option="G",name="Cluster")

map_istat
```



Ad appartenere al primo cluster sono province del Centro e Nord Italia: la quasi totalità delle province della pianura padana (provincie lombarde, venete ed emiliane), una parte del Piemonte, la Valle d'Aosta, le province umbre e delle zone limitrofe (province toscane, marchigiane e abruzzesi), la città di Roma e di Taranto. Al secondo cluster appartengono le province liguri-piemontesi, quasi tutte le province friulane e sarde e, per finire, la striscia di terra che va dalla Toscana al Molise. Il terzo cluster è invece particolarmente omogoneo per la sua collocazione geografica. Questo è infatti composto per la sua totalità da province del Sud Italia e della Sicilia, ad esclusione della città di Frosinone.

Si va ora a studiare più da vicino il gruppo dei dati spuri che sono stati esclusi dall'analisi. Per tali province si mostrano i valori assunti dai loro indicatori:

```
outliers=cbind(istat[dati_fin$cluster==0,1],clus0)
colnames(outliers)[1]=""
row.names(outliers)=c("Bolzano",outliers[-1,1])
as.data.frame(t(outliers)[-1,])
```

```
##                                 Bolzano Siena Caserta Napoli Crotone Cagliari
## tasso di natalità                   9.8   6.6     8.1    8.6     8.3      5.5
## tasso di mortalità                  8.4  13.0     8.7    8.8     9.5      9.3
## tasso di nuzialità                  4.2   4.1     4.1    4.1     4.1      2.7
## saldo migratorio totale             1.5   2.8    -1.6   -4.6   -18.7      0.7
## numero medio di figli per donna    1.71  1.25    1.27   1.37    1.39     0.97
## speranza di vita alla nascita      83.9  83.7    81.5   81.3    82.3     83.5
## indice di vecchiaia               125.0 211.4   116.4  116.5   135.7    196.1
## età media                          42.8  47.3    41.8   41.8    42.8     46.3
```

Le province di Siena e di Cagliari hanno il ruolo di bridge point (punti di frontiera), in quanto possiedono degli indicatori nell'insieme molto bilanciati. La loro attribuzione ad un cluster sarebbe difficile, perchè possiedono delle caratteristiche non identificabili con un solo cluster, ma con la combinazione di almeno due di questi. Le restanti province sono invece dei veri e propri outliers. I valori delle loro variabili sono infatti estremi. Le province di Napoli, Caserta e Crotone, nonostante siano generalmente più vicine al terzo cluster, possiedono dei valori tali da allontanarsi considerevolmente dalla media di anche questo cluster. Ne sono un esempio i tassi di natalità e di nuzialità altissimi e i tassi di mortalità ed età medie molto basse. Differente è invece la situazione della provincia di Bolzano. Se alcuni indicatori fanno pensare ad una forte vicinanza al terzo gruppo (come il tasso di natalità, nuzialità e l'età media), d'altra parte gli altri indici sono più vicini agli altri due gruppi (come il saldo migratorio positivo e la speranza di vita più alta).

Confronto Tclust - Mclust

In quest'ultima parte dell'analisi si vuole confrontare per questo specifico problema il clustering fatto dal pacchetto tclust con mclust. Si controlla inanzitutto quale sia il miglior modello a mistura finita con componenti Gaussiane:

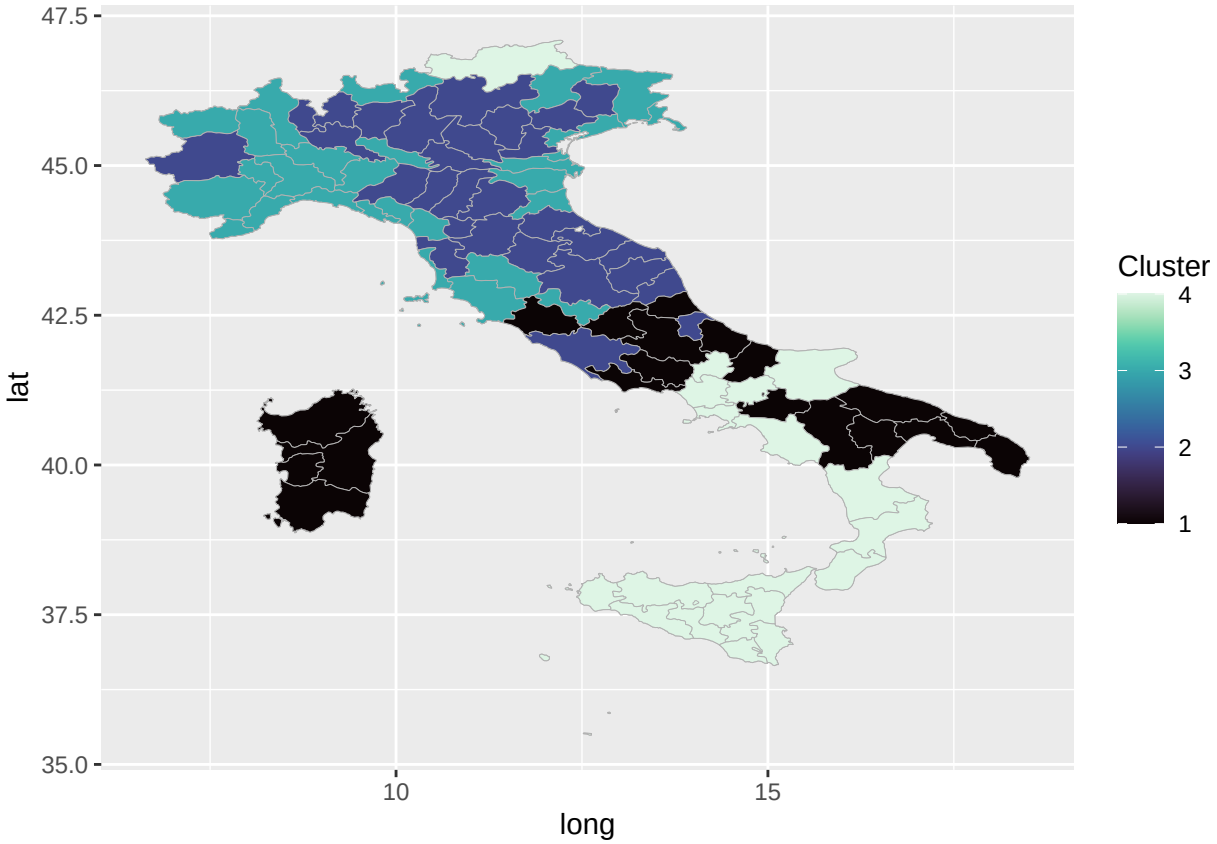
```
fit_mm_istat=mclustBIC(dati_scale,verbose=FALSE,initialization=list(hcRandomPairs(dati_scale)))
summary(fit_mm_istat,dati_scale)
```

```
## Best BIC values:
##              VEE,4        EVE,3        VEE,3
## BIC      -1331.116 -1335.733950 -1339.806612
## BIC diff     0.000    -4.618209    -8.690872
##
## Classification table for model (VEE,4):
##
##  1  2  3  4
## 20 31 34 22
```

Il miglior modello è di tipo VEE con 4 cluster. Si stima allora questo modello e si mette a grafico la mappa geografica di classificazione:

```
fit2=Mclust(dati_scale,verbose=FALSE,modelNames="VEE",G=4)

dati_fin_2=cbind(istat,fit2$classification)
```

```
dati_map_2=dati_fin_2[,c(1,10)]
dati_map_2[c(9,26,41,46,91),1]=c("Aosta","Bolzano-Bozen","Reggio Emilia","Forli'","Reggio Calabria")
dati_map_2=dati_map_2[-c(7,8,23,24,25,47,57,64,86,92,93,107),]
dati_map_2=dati_map_2[order(dati_map_2[,1],decreasing=FALSE),]
dati_map_2[dati_map_2[,2]==2,2]=5
dati_map_2[dati_map_2[,2]==3,2]=2
dati_map_2[dati_map_2[,2]==5,2]=3

clus_prov_2=tibble(region=sort(unique(italy_map$region)),value=dati_map_2[,2])

map_istat_2=ggplot() +
  geom_map(data=italy_map,map=italy_map,aes(long,lat,map_id=region),color="#b2b2b2",size=0.1) +
  geom_map(data=clus_prov_2,map=italy_map,aes(fill=value,map_id=region),color="#b2b2b2",size=0.1) +
  scale_fill_viridis(option="G",name="Cluster")

map_istat_2
```



La classificazione realizzata con mclust appare molto simile a quella fatta con tclust. Zone che erano state associate ad un certo cluster con tclust appaiono molto spesso associate ad un cluster analogo in mclust. Di particolare interesse risulta essere la collocazione delle province che erano state tagliate con tclust. Queste province sono state in genere associate al cluster geograficamente più vicino, ad esclusione della provincia di Bolzano che, sorprendentemente, è stata collocata nel gruppo delle province più a sud.

Per avere un confronto numerico più facile tra i due pacchetti, si sceglie un confronto tra modelli aventi un ugual numero di cluster. Si sceglie allora per il pacchetto mclust il secondo modello migliore secondo il criterio BIC: questo è un modello di tipo EVE a 3 cluster

```
fit3=Mclust(dati_scale,verbose=FALSE,modelNames="EVE",G=3)
dati_fin_3=cbind(istat,fit3$classification)
colnames(dati_fin_3)[10]="cluster"
clus1_3=dati_fin_2[dati_fin_3$cluster==2,-c(1,10)]
clus2_3=dati_fin_2[dati_fin_3$cluster==1,-c(1,10)]
clus3_3=dati_fin_2[dati_fin_3$cluster==3,-c(1,10)]

clus_mean_3=data.frame(round(cbind(apply(clus1_3,2,mean),apply(clus2_3,2,mean),apply(clus3_3,2,mean)),2
colnames(clus_mean_3)=c("Gruppo 1","Gruppo 2","Gruppo 3")
clus_mean_3
```

```
##                                 Gruppo 1 Gruppo 2 Gruppo 3
## tasso di natalità                   6.81     5.59     7.53
## tasso di mortalità                 10.96    12.90    10.46
## tasso di nuzialità                  2.85     2.95     3.85
## saldo migratorio totale             1.62     1.64    -4.87
## numero medio di figli per donna     1.26     1.15     1.27
## speranza di vita alla nascita      83.56    83.05    82.40
## indice di vecchiaia               184.52   242.58   160.73
## età media                          46.03    48.32    44.26
```

I cluster trovati possiedono valori più estremi per le variabili d'interesse rispetto a quelli con tclust. Questo fenomeno è da ricondursi alla metodologia di clustering robusto di tclust: non prendere in considerazione le unità con valori più estremi porta a delle medie di cluster meno variabili. Le stime di tclust, eliminando i dati spuri, risultano così più consistenti. In questo problema, inoltre, le province eliminate sono poche e facilemnte caratterizzabili. Per questi motivi, il pacchetto tclust risulta essere una valida alternativa ai metodi di clustering tradizionali