Analisi e Modellazione statistica su Wine Dataset

Pennelli Lorenzo Maria
E-mail address
lorenzo.pennelli@edu.unifi.it
Github
https://github.com/Pennelli02/FSM_Exam

Abstract

In questo progetto analizziamo la composizione chimica di diversi campioni di vino con l'obiettivo di valutare in che modo alcune sostanze, tra cui flavonoidi e fenoli, influenzino l'intensità del colore. L'analisi prenderà in considerazione sia l'intera popolazione dei campioni sia specifiche sottopopolazioni, suddivise in base al tipo di coltivazione, al fine di verificare se esse seguano gli stessi trend generali o presentino pattern distinti.

Indice

1. Introduzione

Il dataset utilizzato è **Wine** [1], che raccoglie le analisi chimiche di vini prodotti nella stessa regione italiana, ma derivanti da tre differenti tipologie di coltivazione (**Cult**). Il dataset riporta tredici costituenti chimici comuni a tutte le tipologie considerate. La variabile target del nostro studio è **Clri**, una variabile continua che rappresenta l'intensità di colore del vino. Oltre a essa sono presenti numerose variabili chimiche (continue) che analizzeremo per valutarne l'influenza sulla variabile target.

- **Alch**: contenuto alcolico.
- **Mlca**: acido malico; acido organico che contribuisce all'acidità e al gusto del vino.
- **Ash**: residuo secco (cenere); indica la quantità di minerali presenti dopo combustione del campione.
- **Aloa**: alcalinità.
- **Mgns**: magnesio.
- **Ttlp**: fenoli totali; insieme dei composti fenolici che influenzano colore, sapore e proprietà antiossidanti.
- **Flvn**: flavonoidi; classe di polifenoli che contribuiscono al colore, all'amarezza e alle proprietà antiossidanti.
- **Nnfp**: fenoli non flavonoidi; si comportano come antiossidanti e supportano le difese immunitarie della pianta .
- **Prnt**: proantocianidine; polifenoli oligomerici responsabili dell'astringenza e della stabilità del colore.
- **Hue**: tonalità di colore; rapporto tra componenti di colore rosso/giallo.
- **Oodw**: assorbanza OD280/OD315 di vino diluito; misura la spettrofotometrica della concentrazione di composti fenolici.
- **Prln**: prolina; amminoacido presente nel vino, indica maturità delle uve.

Nel dataset è, inoltre, presente la variabile categoriale **Cult**, che verrà descritta nel dettaglio nella sezione 2. Pur non essendo considerata inizialmente come variabile esplicativa per l'intensità di colore, sarà comunque utilizzata per suddividere la popolazione dei campioni e per eventuali approfondimenti.

1.1. Software

L'analisi statistica è stata realizzata completamente in **R**, organizzando il dataset e la relativa struttura di lavoro all'interno dell'ambiente di sviluppo **RStudio**. Sono state usate diverse librerie e pacchetti per scaricare il dataset, per la visulizzazione grafica e per la costruzioni di modelli grafici indiretti e diretti. L'intero progetto è documentato in **latex** e saranno presenti immagini di grafici e script di codice su cui porre l'attenzione.

2. Osservazioni e Preprocessing dei dati

2.1. Visualizzazione Dataset

Un passaggio fondamentale, prima di avviare qualunque analisi statistica, consiste nell'esaminare il dataset oggetto di studio. Questo permette di comprenderne la struttura, il tipo di variabili presenti e il loro formato. Inoltre, consente di verificare l'eventuale presenza di dati mancanti e di valutare se tutte le informazioni disponibili siano effettivamente necessarie per l'analisi.

```r
#Carico dataset
require(gRbase)
data(wine)

#Visualizzazione struttura
str(wine)

#visualizzazioni statistiche del dataset
summary(wine)
```

```
> #Visualizzazione struttura
> str(wine)
'data.frame': 178 obs. of  14 variables:
 $ Cult: Factor w/ 3 levels "v1","v2","v3": 1 1 1 1 1 1 1 1 1 1 ...
 $ Alch: num  14.2 13.2 13.2 14.4 13.2 ...
 $ Mlca: num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
 $ Ash : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
 $ Aloa: num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
 $ Mgns: int  127 100 101 113 118 112 96 121 97 98 ...
 $ Ttlp: num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
 $ Flvn: num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
 $ Nnfp: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
 $ Prnt: num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
 $ Clri: num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
 $ Hue : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
 $ Oodw: num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
 $ Prln: int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
>
> #visualizzazioni statistiche del dataset
> summary(wine)
```

Cult	Alch	Mlca	Ash	Aloa	Mgns
v1:59	Min. :11.03	Min. :0.740	Min. :1.360	Min. :10.60	Min. : 70.00
v2:71	1st Qu.:12.36	1st Qu.:1.603	1st Qu.:2.210	1st Qu.:17.20	1st Qu.: 88.00
v3:48	Median :13.05	Median :1.865	Median :2.360	Median :19.50	Median : 98.00
	Mean :13.00	Mean :2.336	Mean :2.367	Mean :19.49	Mean : 99.74
	3rd Qu.:13.68	3rd Qu.:3.083	3rd Qu.:2.558	3rd Qu.:21.50	3rd Qu.:107.00
	Max. :14.83	Max. :5.800	Max. :3.230	Max. :30.00	Max. :162.00

Ttlp	Flvn	Nnfp	Prnt	Clri
Min. :0.980	Min. :0.340	Min. :0.1300	Min. :0.410	Min. : 1.280
1st Qu.:1.742	1st Qu.:1.205	1st Qu.:0.2700	1st Qu.:1.250	1st Qu.: 3.220
Median :2.355	Median :2.135	Median :0.3400	Median :1.555	Median : 4.690
Mean :2.295	Mean :2.029	Mean :0.3619	Mean :1.591	Mean : 5.058
3rd Qu.:2.800	3rd Qu.:2.875	3rd Qu.:0.4375	3rd Qu.:1.950	3rd Qu.: 6.200
Max. :3.880	Max. :5.080	Max. :0.6600	Max. :3.580	Max. :13.000

Hue	Oodw	Prln
Min. :0.4800	Min. :1.270	Min. : 278.0
1st Qu.:0.7825	1st Qu.:1.938	1st Qu.: 500.5

```
 Median :0.9650   Median :2.780   Median : 673.5
 Mean   :0.9574   Mean   :2.612   Mean   : 746.9
 3rd Qu.:1.1200   3rd Qu.:3.170   3rd Qu.: 985.0
 Max.   :1.7100   Max.   :4.000   Max.   :1680.0
```

L'ispezione della struttura del dataset mostra che esso contiene 178 istanze descritte tramite 14 variabili. La maggior parte delle variabili (13) è di tipo numerico (*num* e *int*), mentre *Cult* rappresenta una variabile categorica fattoriale. Essa identifica tre diverse tipologie di cultivar di vino, indicate mediante le etichette *v1*, *v2* e *v3*: si tratta quindi di una classificazione qualitativa priva di ordinamento naturale. Le osservazioni risultano distribuite in modo abbastanza uniforme tra i tre gruppi, anche se la categoria v2 presenta un numero leggermente maggiore di casi. Il dataset, infine, non contiene valori nulli.

2.2. Preparazione dei dati

La fase di preparazione dei dati ha lo scopo di rendere il dataset adeguato alle analisi che verranno effettuate successivamente. In questo caso, è stato necessario intervenire sulla struttura originale dei dati per isolare le variabili di interesse e organizzare le osservazioni in modo funzionale agli obiettivi dello studio. In particolare, si è scelto di rimuovere la variabile categorica Cult dal dataset principale e, parallelamente, di suddividere le osservazioni in tre insiemi distinti sulla base dei rispettivi cultivar, così da consentire analisi mirate sui singoli gruppi.

```r
#Togliamo dal dataset la variabile categorica Cult
wine_dataset<-subset(wine, select = -Cult)
str(wine_dataset)

# Creiamo tre dataset dei vini raggruppati per tipo di cultura (v1, v2, v3)
list_wine<-split(wine, wine$Cult)

# visualizziamo tutte e tre le liste di vini separati per Cult
str(list_wine)
```

Ottenendo così:

```
> str(wine_dataset)
'data.frame': 178 obs. of  13 variables:
 $ Alch: num  14.2 13.2 13.2 14.4 13.2 ...
 $ Mlca: num  1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
 $ Ash : num  2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
 $ Aloa: num  15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
 $ Mgns: int  127 100 101 113 118 112 96 121 97 98 ...
 $ Ttlp: num  2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
 $ Flvn: num  3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
 $ Nnfp: num  0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
 $ Prnt: num  2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
 $ Clri: num  5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
 $ Hue : num  1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
 $ Oodw: num  3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
 $ Prln: int  1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...

> # visualizziamo tutte e tre le liste di vini separati per Cult
```

```
> str(list_wine)
List of 3
 $ v1:'data.frame': 59 obs. of  14 variables:
  ..$ Cult: Factor w/ 3 levels "v1","v2","v3": 1 1 1 1 1 1 1 1 1 1 ...
  ..$ Alch: num [1:59] 14.2 13.2 13.2 14.4 13.2 ...
  ..$ Mlca: num [1:59] 1.71 1.78 2.36 1.95 2.59 1.76 1.87 2.15 1.64 1.35 ...
  ..$ Ash : num [1:59] 2.43 2.14 2.67 2.5 2.87 2.45 2.45 2.61 2.17 2.27 ...
  ..$ Aloa: num [1:59] 15.6 11.2 18.6 16.8 21 15.2 14.6 17.6 14 16 ...
  ..$ Mgns: int [1:59] 127 100 101 113 118 112 96 121 97 98 ...
  ..$ Ttlp: num [1:59] 2.8 2.65 2.8 3.85 2.8 3.27 2.5 2.6 2.8 2.98 ...
  ..$ Flvn: num [1:59] 3.06 2.76 3.24 3.49 2.69 3.39 2.52 2.51 2.98 3.15 ...
  ..$ Nnfp: num [1:59] 0.28 0.26 0.3 0.24 0.39 0.34 0.3 0.31 0.29 0.22 ...
  ..$ Prnt: num [1:59] 2.29 1.28 2.81 2.18 1.82 1.97 1.98 1.25 1.98 1.85 ...
  ..$ Clri: num [1:59] 5.64 4.38 5.68 7.8 4.32 6.75 5.25 5.05 5.2 7.22 ...
  ..$ Hue : num [1:59] 1.04 1.05 1.03 0.86 1.04 1.05 1.02 1.06 1.08 1.01 ...
  ..$ Oodw: num [1:59] 3.92 3.4 3.17 3.45 2.93 2.85 3.58 3.58 2.85 3.55 ...
  ..$ Prln: int [1:59] 1065 1050 1185 1480 735 1450 1290 1295 1045 1045 ...
 $ v2:'data.frame': 71 obs. of  14 variables:
  ..$ Cult: Factor w/ 3 levels "v1","v2","v3": 2 2 2 2 2 2 2 2 2 2 ...
  ..$ Alch: num [1:71] 12.4 12.3 12.6 13.7 12.4 ...
  ..$ Mlca: num [1:71] 0.94 1.1 1.36 1.25 1.13 1.45 1.21 1.01 1.17 0.94 ...
  ..$ Ash : num [1:71] 1.36 2.28 2.02 1.92 2.16 2.53 2.56 1.7 1.92 2.36 ...
  ..$ Aloa: num [1:71] 10.6 16 16.8 18 19 19 18.1 15 19.6 17 ...
  ..$ Mgns: int [1:71] 88 101 100 94 87 104 98 78 78 110 ...
  ..$ Ttlp: num [1:71] 1.98 2.05 2.02 2.1 3.5 1.89 2.42 2.98 2.11 2.53 ...
  ..$ Flvn: num [1:71] 0.57 1.09 1.41 1.79 3.1 1.75 2.65 3.18 2 1.3 ...
  ..$ Nnfp: num [1:71] 0.28 0.63 0.53 0.32 0.19 0.45 0.37 0.26 0.27 0.55 ...
  ..$ Prnt: num [1:71] 0.42 0.41 0.62 0.73 1.87 1.03 2.08 2.28 1.04 0.42 ...
  ..$ Clri: num [1:71] 1.95 3.27 5.75 3.8 4.45 2.95 4.6 5.3 4.68 3.17 ...
  ..$ Hue : num [1:71] 1.05 1.25 0.98 1.23 1.22 1.45 1.19 1.12 1.12 1.02 ...
  ..$ Oodw: num [1:71] 1.82 1.67 1.59 2.46 2.87 2.23 2.3 3.18 3.48 1.93 ...
  ..$ Prln: int [1:71] 520 680 450 630 420 355 678 502 510 750 ...
 $ v3:'data.frame': 48 obs. of  14 variables:
  ..$ Cult: Factor w/ 3 levels "v1","v2","v3": 3 3 3 3 3 3 3 3 3 3 ...
  ..$ Alch: num [1:48] 12.9 12.9 12.8 12.7 12.5 ...
  ..$ Mlca: num [1:48] 1.35 2.99 2.31 3.55 1.24 2.46 4.72 5.51 3.59 2.96 ...
  ..$ Ash : num [1:48] 2.32 2.4 2.4 2.36 2.25 2.2 2.54 2.64 2.19 2.61 ...
  ..$ Aloa: num [1:48] 18 20 24 21.5 17.5 18.5 21 25 19.5 24 ...
  ..$ Mgns: int [1:48] 122 104 98 106 85 94 89 96 88 101 ...
  ..$ Ttlp: num [1:48] 1.51 1.3 1.15 1.7 2 1.62 1.38 1.79 1.62 2.32 ...
  ..$ Flvn: num [1:48] 1.25 1.22 1.09 1.2 0.58 0.66 0.47 0.6 0.48 0.6 ...
  ..$ Nnfp: num [1:48] 0.21 0.24 0.27 0.17 0.6 0.63 0.53 0.63 0.58 0.53 ...
  ..$ Prnt: num [1:48] 0.94 0.83 0.83 0.84 1.25 0.94 0.8 1.1 0.88 0.81 ...
  ..$ Clri: num [1:48] 4.1 5.4 5.7 5 5.45 7.1 3.85 5 5.7 4.92 ...
  ..$ Hue : num [1:48] 0.76 0.74 0.66 0.78 0.75 0.73 0.75 0.82 0.81 0.89 ...
  ..$ Oodw: num [1:48] 1.29 1.42 1.36 1.29 1.51 1.58 1.27 1.69 1.82 2.15 ...
  ..$ Prln: int [1:48] 630 530 560 600 650 695 720 515 580 590 ...
```

3. Analisi Esplorativa

In questa sezione ci concentreremo sulla nostra variabile target **Clri**, che rappresenta l'intensità del colore del vino ed è di natura numerica. L'analisi riguarderà la sua distribuzione, la relazione con le altre variabili e la definizione di un suo modello esplicativo. Le valutazioni saranno svolte sia considerando l'intero dataset sia esaminando separatamente le diverse tipologie di coltivazione. A tal fine, sono state utilizzate diverse tipologie di visualizzazione, che vanno dalle statistiche descrittive fino alle matrici di correlazione rappresentate tramite heatmap. L'obiettivo principale di questa analisi esplorativa è comprendere la variabilità di **Clri**, identificare eventuali outlier e rilevare relazioni significative con le altre variabili, così da fornire informazioni fondamentali per le successive fasi di modellizzazione statistica.

3.1. Statistiche Descrittive e Density Plot

Per prima cosa osserviamo cosa ci dicono le *statistiche descrittive* sull'intera popolazione e sui singoli gruppi di cultivar. Le statistiche forniscono informazioni sulla tendenza centrale, la dispersione e l'eventuale presenza di valori estremi.

```
#mi fornisce la statistica colore dell'intera popolazione
summary(wine_dataset$Clri)

#mi fornisce le statistiche del colore per una singola lista di vini
#v1
summary(list_wine$v1$Clri)
#v2
summary(list_wine$v2$Clri)
#v3
summary(list_wine$v3$Clri)
```

```
> #mi fornisce la statistica colore dell'intera popolazione
> summary(wine_dataset$Clri)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.280   3.220   4.690   5.058   6.200  13.000
>
> #mi fornisce le statistiche del colore per una singola lista di vini
> #v1
> summary(list_wine$v1$Clri)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.520   4.550   5.400   5.528   6.225   8.900
> #v2
> summary(list_wine$v2$Clri)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.280   2.535   2.900   3.087   3.400   6.000
> #v3
> summary(list_wine$v3$Clri)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.850   5.438   7.550   7.396   9.225  13.000
```

Dall'analisi dei valori riassuntivi si osserva che la variabile **Clri** presenta una distribuzione con un'ampia variabilità, con valori compresi tra 1.28 e 13. In particolare, i valori medi e mediani differiscono notevolmente tra i

cultivar: il gruppo *v2* tende ad avere valori più bassi, mentre il gruppo *v3* mostra i valori più alti di intensità del colore. Il gruppo *v1* si colloca in una posizione intermedia. Questa differenziazione suggerisce una chiara variabilità tra i cultivar, confermata anche dai quartili.

Per visualizzare meglio la distribuzione, sono stati realizzati *density plot* sia sull'intero dataset sia per ciascun gruppo di cultivar:

```
# Calcolo densita' di tutti i gruppi
dens_pop <- density(wine_dataset$Clri)
dens_v1  <- density(list_wine$v1$Clri)
dens_v2  <- density(list_wine$v2$Clri)
dens_v3  <- density(list_wine$v3$Clri)

# Determino il massimo valore di y per includere tutte le curve
ymax <- max(dens_pop$y, dens_v1$y, dens_v2$y, dens_v3$y)

plot(dens_pop, main="Density plot di Clri - Popolazione e Cultivar", xlab="Clri",
lwd=4, col=rgb(0,0,0,0.7), ylim=c(0,ymax*1.1)) # aggiungo un 10% per margine

# Aggiunta delle altre densita'
lines(dens_v1, lwd=2, col=rgb(0,1,0,0.5))
lines(dens_v2, lwd=2, col=rgb(1,0,0,0.5))
lines(dens_v3, lwd=2, col=rgb(0,0,1,0.5))

# Legenda
legend("topright", legend=c("Popolazione","v1","v2","v3"),
       col=c("black","green","red","blue"), lwd=2)
```

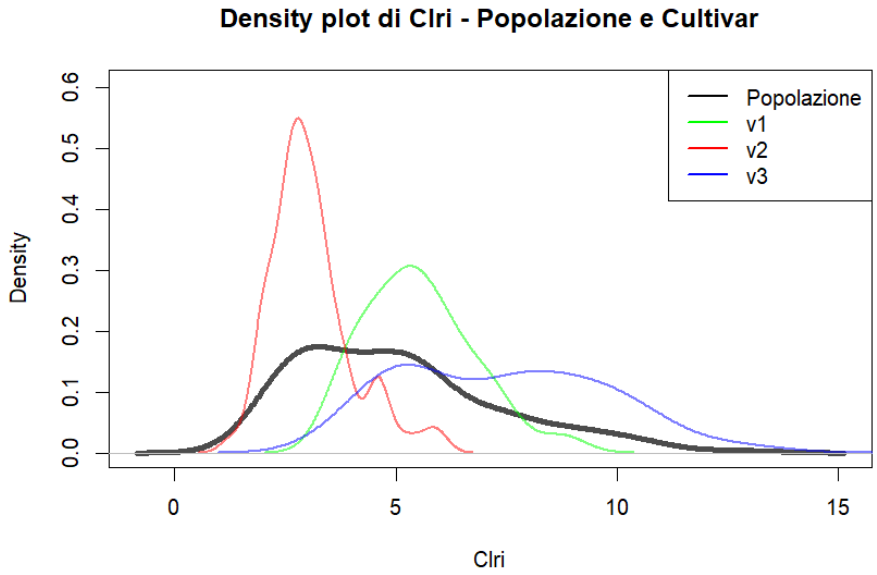


Figura 1: Density Plot del valore **Clri** per il confronto tra popolazione generale e cultivar

Il *density plot* conferma la netta differenziazione tra cultivar evidenziata dalle statistiche descrittive: *v2* presenta valori concentrati e bassi, *v3* valori alti e dispersi, mentre *v1* occupa una posizione intermedia. La distribuzione

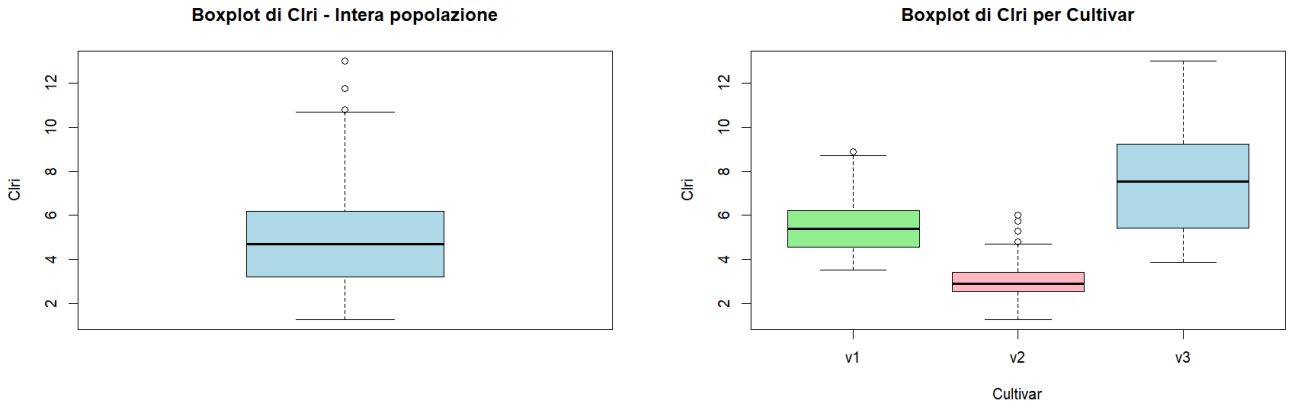
della popolazione totale riflette questa eterogeneità. Questa separazione suggerisce una possibile associazione tra tipo di coltivazione e intensità del colore.

3.2. BoxPlot

Per approfondire la distribuzione della variabile Clri, sono stati realizzati due tipi di *boxplot*: il primo sull'intera popolazione e il secondo suddiviso per cultivar. Il boxplot complessivo permette di osservare in maniera sintetica la variabilità generale, la posizione della mediana e l'eventuale presenza di outlier. La versione suddivisa per cultivar consente invece di confrontare direttamente i tre gruppi e di verificare se presentano differenze rilevanti in termini di posizione centrale e dispersione. Questa doppia visualizzazione supporta un'interpretazione più completa della variabile target sia a livello aggregato, sia rispetto alla variabile categoriale Cult.

```r
# Boxplot dell'intera popolazione
boxplot(wine_dataset$Clri,
        main = "Boxplot di Clri - Intera popolazione",
        ylab = "Clri",
        col = "lightblue")

# Boxplot suddiviso per cultivar
boxplot(Clri ~ Cult,
        data = wine,
        main = "Boxplot di Clri per Cultivar",
        xlab = "Cultivar",
        ylab = "Clri",
        col = c("lightgreen", "lightpink", "lightblue"))
```



(a) Intera popolazione

(b) divisione in base a Cult

Figura 2: Grafici Boxplot

Come si può notare dalla Figura 2a, nell'intera popolazione la variabile Clri presenta una variabilità moderata e alcuni valori anomali verso l'alto. Tuttavia, osservando la Figura 2b, emerge chiaramente che tale variabilità non è uniforme tra le tre cultivar:

- la v2 mostra valori nettamente più bassi e concentrati,
- la v1 presenta valori intermedi con una dispersione contenuta,

- la v3 evidenzia invece livelli più elevati e una maggiore variabilità.

Queste differenze suggeriscono che la tipologia di cultivar influenzi in modo significativo l'intensità del colore, confermando la rilevanza della variabile Cult nel processo produttivo e motivando ulteriori analisi sulle relazioni tra gruppo di appartenenza e variabile target.

3.3. Matrici Scatterplot

In questa sezione vengono presentate le *matrici scatterplot*, uno strumento utile per osservare visivamente le relazioni tra le variabili numeriche del dataset. Questi grafici permettono di individuare eventuali pattern, associazioni lineari o non lineari. Le matrici saranno analizzate sia per l'intera popolazione sia per ciascun sottogruppo definito dalla variabile Cult, così da valutare eventuali differenze nelle relazioni tra variabili all'interno delle diverse cultivar.

```
# Dividere i dataset rimuovendo la variabile categorica Cult per una
#questione estetica
v1 <- subset(list_wine$v1, select = -Cult)
v2 <- subset(list_wine$v2, select = -Cult)
v3 <- subset(list_wine$v3, select = -Cult)

#Matrici Scatter Plot
#Intera popolazione
plot(wine_dataset, main="Scatterplot Matrix dell' intera popolazione")

#Diviso per tipo di cultivar
plot(v1, main="Scatterplot Matrix di v1")
plot(v2, main="Scatterplot Matrix di v2")
plot(v3, main="Scatterplot Matrix di v3")
```

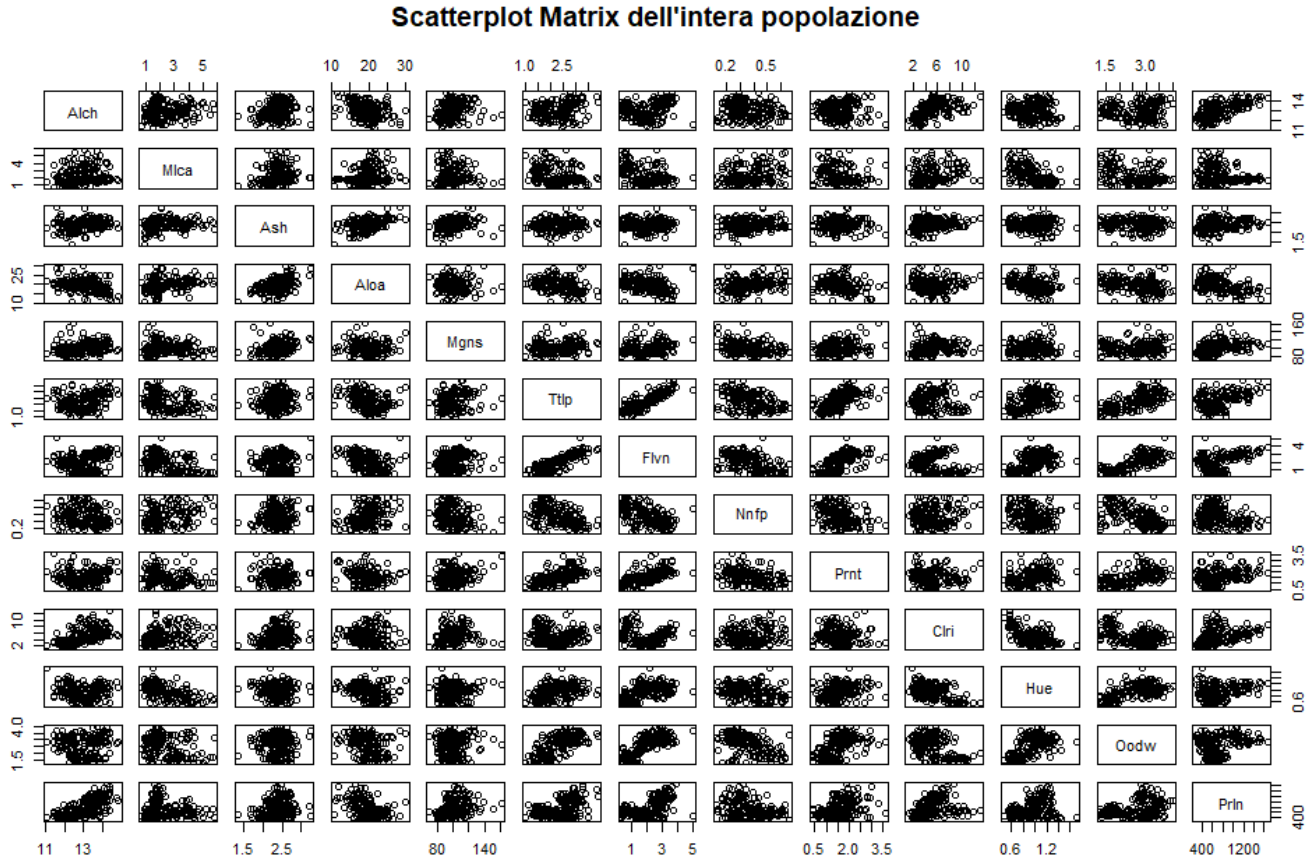


Figura 3: Scatter plot dell'intera popolazione

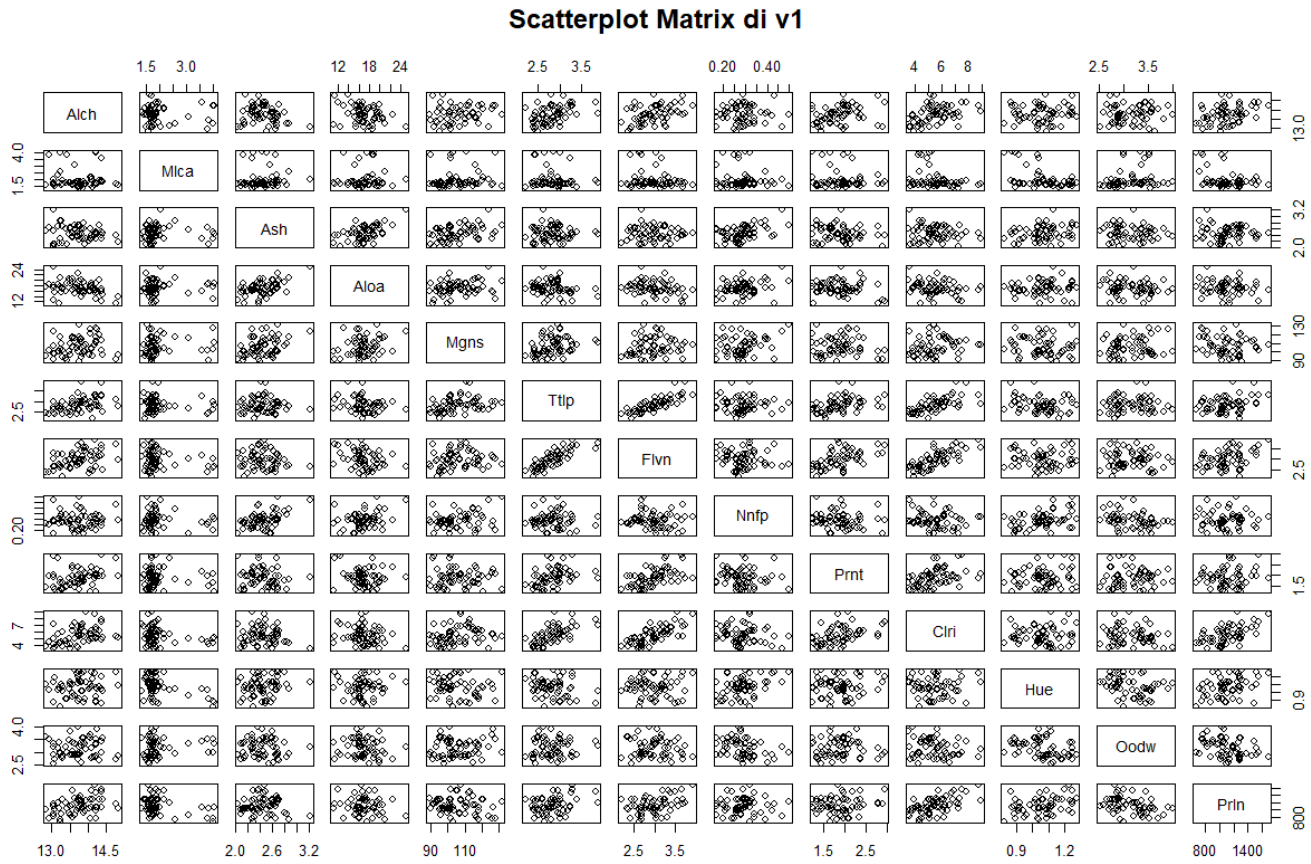


Figura 4: Scatter plot cultivar v1

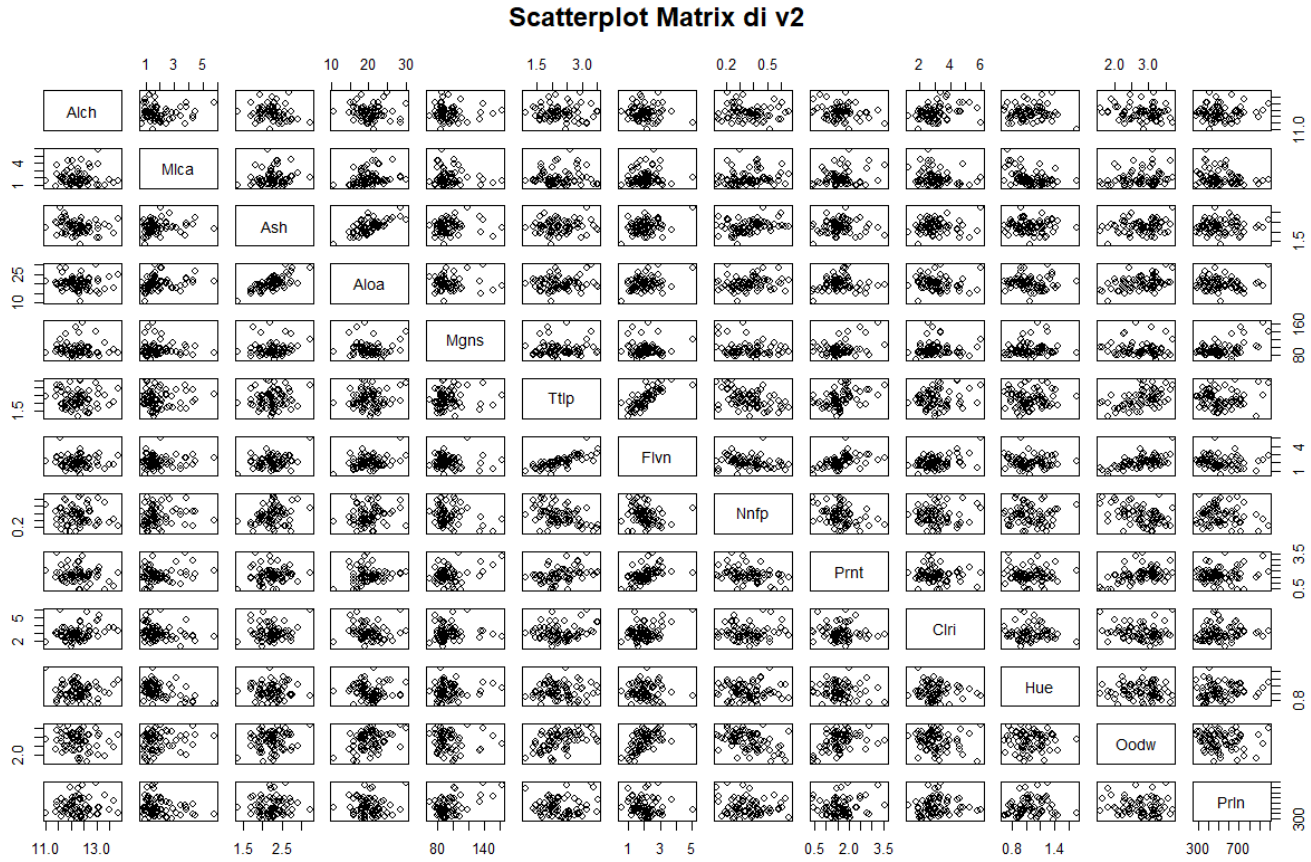


Figura 5: Scatter plot cultivar v2

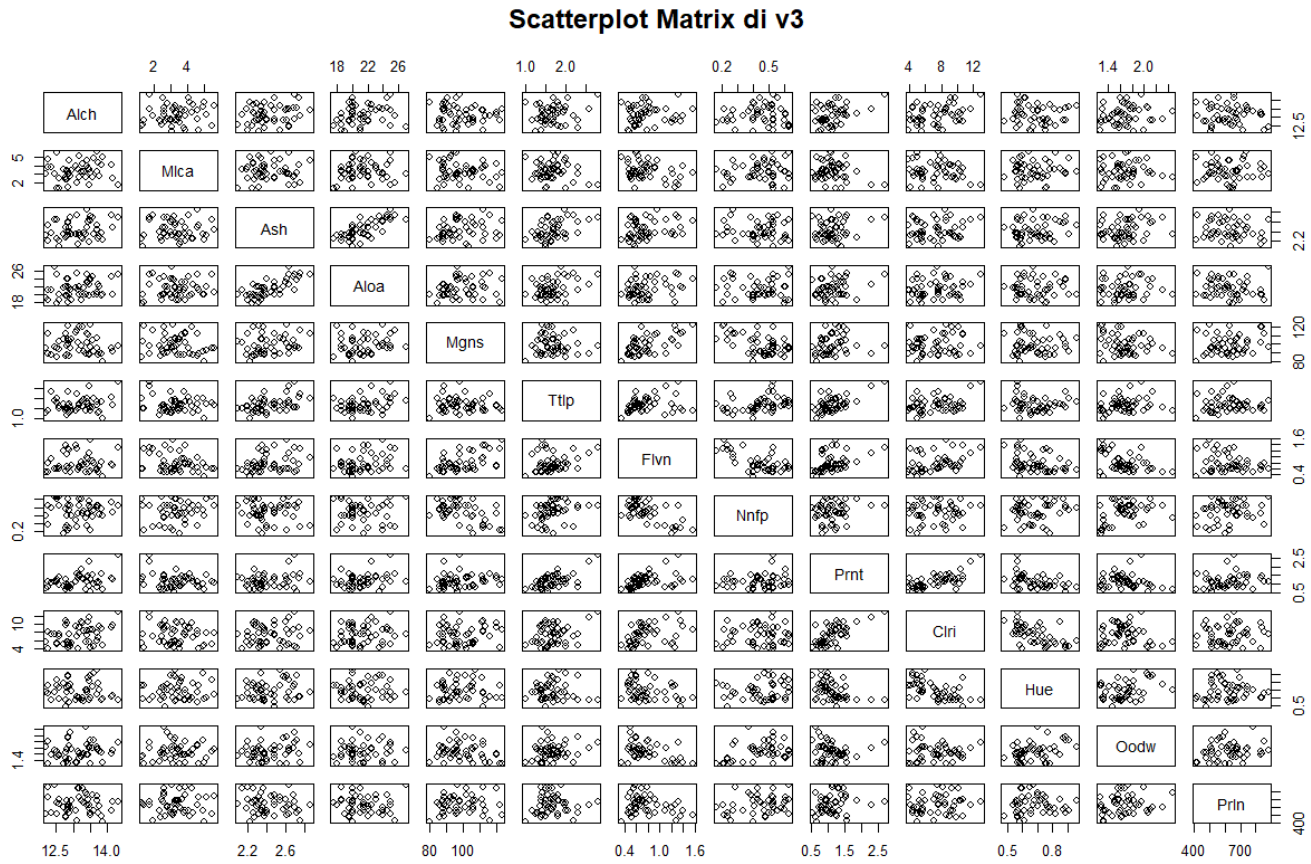


Figura 6: Scatter plot cultivar v3

3.4. Matrice di concentrazione

In questa sezione calcoleremo e analizzeremo le *matrici di concentrazione*, la cui importanza è introdotta nella sezione 4, che ci permettono di evidenziare le dipendenze condizionali tra le variabili attraverso informazioni numeriche e tabellari. Per facilitare il confronto tra le relazioni condizionali, verrà inoltre riportata la *matrice delle correlazioni parziali*, che fornisce una versione normalizzata e più interpretabile delle intensità di dipendenza tra le variabili.
Qui di seguito sarà presente il codice per le **matrici di concentrazione**

```
#Matrici di concentrazione
#intera popolazione
CovPop <- cov.wt(wine_dataset, method = "ML")$cov
concPop <-solve(CovPop)
round(100*concPop)

#diviso per cultivar
#V1
CovV1 <- cov.wt(v1, method = "ML")$cov
concV1 <- solve(CovV1)
round(100*concV1)
```

```r
#V2
CovV2 <- cov.wt(v2, method = "ML")$cov
concV2 <- solve(CovV2)
round(100*concV2)

#V3
CovV3 <- cov.wt(v3, method = "ML")$cov
concV3 <- solve(CovV3)
round(100*concV3)
```

```
> round(100*concPop)
      Alch Mlca   Ash Aloa Mgns Ttlp Flvn  Nnfp Prnt Clri   Hue Oodw Prln
Alch   375  -49   -52   14    0  -20   -3    78   57  -61   -81  -60    0
Mlca   -49  133   -68   -3    0    3   30   -97  -23   13   325   -2    0
Ash    -52  -68  2920 -143  -13   10 -349 -1645  298  -44    -4  -66   -1
Aloa    14   -3  -143   20    0    2   16    -1  -14   -1     8   -4    0
Mgns     0    0   -13    0    1   -1    3    26   -4    0    -3    3    0
Ttlp   -20    3    10    2   -1 1113 -500  -258  -77  -30    38 -156    0
Flvn    -3   30  -349   16    3 -500  709   709 -201    0  -311 -227    0
Nnfp    78  -97 -1645   -1   26 -258  709 11664 -144  -30 -1235  517    0
Prnt    57  -23   298  -14   -4  -77 -201  -144  606  -30     3  -65    0
Clri   -61   13   -44   -1    0  -30    0   -30  -30   57   239   86    0
Hue    -81  325    -4    8   -3   38 -311 -1235    3  239  4911  -53   -1
Oodw   -60   -2   -66   -4    3 -156 -227   517  -65   86   -53  755    0
Prln     0    0    -1    0    0    0    0     0    0    0    -1    0    0
```

```
> round(100*concV1)
      Alch Mlca   Ash Aloa Mgns  Ttlp  Flvn   Nnfp Prnt Clri    Hue Oodw Prln
Alch   802  -82   298   31   -8  -135   -47   -941 -150    7   -293 -198   -1
Mlca   -82  302  -137    0    0    68    22     36  -39   23    802   32    0
Ash    298 -137  3997 -139  -24  -351  -339  -2667  123  165  -1673 -187   -1
Aloa    31    0  -139   27   -1    29    37    -89  -20   -3     65   12    0
Mgns    -8    0   -24   -1    2   -16    11    -28    8   -7      1   -4    0
Ttlp  -135   68  -351   29  -16  4052 -2316  -3766  -23  -80   3995 -461   -1
Flvn   -47   22  -339   37   11 -2316  3202   1471 -495 -320  -1618  274    1
Nnfp  -941   36 -2667  -89  -28 -3766  1471  41705  794  535 -11031 2791    3
Prnt  -150  -39   123  -20    8   -23  -495    794 1014  -77   -968  -31    1
Clri     7   23   165   -3   -7   -80  -320    535  -77  252    146   88   -1
Hue   -293  802 -1673   65    1  3995 -1618 -11031 -968  146  17969 -105   -3
Oodw  -198   32  -187   12   -4  -461   274   2791  -31   88   -105 1231    1
Prln    -1    0    -1    0    0    -1     1      3    1   -1     -3    1    0
```

```
> round(100*concV2)
     Alch Mlca   Ash Aloa Mgns Ttlp Flvn  Nnfp Prnt Clri  Hue Oodw Prln
Alch  454  -34   336  -24    0   48  -56   334  103  -71  -54   89    0
Mlca  -34  148   -23   -4    0  -24   11  -238  -59   32  275  -15    0
Ash   336  -23  2758 -157   -9  186 -547 -1031  295   23    5  252    0
Aloa  -24   -4  -157   23    0    9   -7   -79    0   12    6  -52    0
Mgns    0    0    -9    0    1   -1    3    16   -6   -1   -5    3    0
Ttlp   48  -24   186    9   -1 1195 -965  1254  235  175  -84  162   -1
Flvn  -56   11  -547   -7    3 -965 1488  -748 -519 -390  -92 -528    1
Nnfp  334 -238 -1031  -79   16 1254 -748 13234  438   49 -484 1593    0
Prnt  103  -59   295    0   -6  235 -519   438  620  136   45   63    0
Clri  -71   32    23   12   -1  175 -390    49  136  258  133  163    0
Hue   -54  275     5    6   -5  -84  -92  -484   45  133 3095   21    0
Oodw   89  -15   252  -52    3  162 -528  1593   63  163   21 1060    0
Prln    0    0     0    0    0   -1    1     0    0    0    0    0    0

> round(100*concV3)
     Alch Mlca   Ash Aloa Mgns  Ttlp  Flvn  Nnfp  Prnt Clri   Hue  Oodw Prln
Alch  606  -59  -335    7    5   165   385  1056  -581  -41  -865  -144    1
Mlca  -59  109   -20  -10    0    13    66  -148   103    4   124    81    0
Ash  -335  -20 10630 -526  -35 -1638  -522  -478  1079  -67 -1383 -1211    2
Aloa    7  -10  -526   53    1    33   -14     4   -60    3    34    41    0
Mgns    5    0   -35    1    2    19   -42    31   -10    0   -40     4    0
Ttlp  165   13 -1638   33   19  2468 -1431 -2516 -1127   45  -595  -660    0
Flvn  385   66  -522  -14  -42 -1431  7211  9449 -1449 -134 -1054  1434    4
Nnfp 1056 -148  -478    4   31 -2516  9449 25938 -3352 -202 -5509  1293    3
Prnt -581  103  1079  -60  -10 -1127 -1449 -3352  2941  -97  1953   350   -2
Clri  -41    4   -67    3    0    45  -134  -202   -97   51   429   -54    0
Hue  -865  124 -1383   34  -40  -595 -1054 -5509  1953  429 17002 -1260   -1
Oodw -144   81 -1211   41    4  -660  1434  1293   350  -54 -1260  2567   -1
Prln    1    0     2    0    0     0     4     3    -2    0    -1    -1    0
>
```

Qui invece mostriamo il codice per le **matrici delle correlazioni parziali**

```
#Matrici delle correlazioni parziali

#intera popolazione
popCP <- cov2pcor(concPop)
round(100*popCP)

#diviso per cultivar

#V1
v1CP <- cov2pcor(concV1)
round(100*v1CP)

#V2
v2CP <- cov2pcor(concV2)
round(100*v2CP)
```

```
#V3
v3CP <- cov2pcor(concV3)
round(100*v3CP)
```

```
> round(100*popCP)
     Alch Mlca Ash Aloa Mgns Ttlp Flvn Nnfp Prnt Clri Hue Oodw Prln
Alch  100   -9 -21   31  -27  -29  -24   16  -14  -55   7   -7  -64
Mlca   -9  100 -16  -29    5   34   41  -29   22  -25  56   37   19
Ash   -21  -16 100  -44  -29  -13  -12  -19   -1  -26   7    0  -22
Aloa   31  -29 -44  100    8   32   35  -36   20   -2  27   28   44
Mgns  -27    5 -29    8  100  -21  -20   26  -24  -20  -6   -7  -39
Ttlp  -29   34 -13   32  -21  100  -86   45  -61    6 -43  -70  -50
Flvn  -24   41 -12   35  -20  -86  100   54  -65   17 -54  -79  -49
Nnfp   16  -29 -19  -36   26   45   54  100   37  -14  26   50   31
Prnt  -14   22  -1   20  -24  -61  -65   37  100    3 -30  -52  -33
Clri  -55  -25 -26   -2  -20    6   17  -14    3  100  52   43  -32
Hue     7   56   7   27   -6  -43  -54   26  -30   52 100  -57  -24
Oodw   -7   37   0   28   -7  -70  -79   50  -52   43 -57  100  -31
Prln  -64   19 -22   44  -39  -50  -49   31  -33  -32 -24  -31  100
```

```
> round(100*v1CP)
     Alch Mlca Ash Aloa Mgns Ttlp Flvn Nnfp Prnt Clri Hue Oodw Prln
Alch  100    4  15   32  -16  -42  -41   -2  -31  -41  -8   -7  -36
Mlca    4  100  -3   -6   -8    8   19    9    8   26  42  -17   37
Ash    15   -3 100  -55  -38    0    7  -47   15   12 -24    8    3
Aloa   32   -6 -55  100  -24   22   29  -30   17   21  -9   12   12
Mgns  -16   -8 -38  -24  100  -31  -12  -24    6  -18  11  -12   15
Ttlp  -42    8   0   22  -31  100  -80    2  -37  -65  22   -5  -29
Flvn  -41   19   7   29  -12  -80  100    9  -55  -74  -1    9  -38
Nnfp   -2    9 -47  -30  -24    2    9  100   14   15 -41   32    2
Prnt  -31    8  15   17    6  -37  -55   14  100  -42 -10    0  -14
Clri  -41   26  12   21  -18  -65  -74   15  -42  100  -3   19  -59
Hue    -8   42 -24   -9   11   22   -1  -41  -10   -3 100   31  -35
Oodw   -7  -17   8   12  -12   -5    9   32    0   19  31  100   35
Prln  -36   37   3   12   15  -29  -38    2  -14  -59 -35   35  100
```

```
> round(100*v2CP)
     Alch Mlca Ash Aloa Mgns Ttlp Flvn Nnfp Prnt Clri Hue Oodw Prln
Alch  100    2  21    6    3    5    4    7   19  -27   0   13   -4
Mlca    2  100 -15  -24    8   -4  -11  -13  -21   20  41  -16   22
Ash    21  -15 100  -70  -13  -11  -31  -30   -4   -6   3  -16   -4
Aloa    6  -24 -70  100    0  -13  -31  -18  -11    9   8  -38    1
Mgns    3    8 -13    0  100   -7    0   19  -30   -4 -12    8  -50
Ttlp    5   -4 -11  -13   -7  100  -77   42  -38  -17  -4  -48   -2
Flvn    4  -11 -31  -31    0  -77  100   24  -50  -38   3  -58   12
Nnfp    7  -13 -30  -18   19   42   24  100   32   -2   3   41   15
Prnt   19  -21  -4  -11  -30  -38  -50   32  100    7   5  -39  -12
Clri  -27   20  -6    9   -4  -17  -38   -2    7  100   3   12  -10
Hue     0   41   3    8  -12   -4    3    3    5    3 100    5  -11
Oodw   13  -16 -16  -38    8  -48  -58   41  -39   12   5  100   11
Prln   -4   22  -4    1  -50   -2   12   15  -12  -10 -11   11  100

> round(100*v3CP)
     Alch Mlca Ash Aloa Mgns Ttlp Flvn Nnfp Prnt Clri Hue Oodw Prln
Alch  100  -11 -25  -21    8  -21   -8   -4  -38  -35   3  -13    9
Mlca  -11  100  -2   -9   17   16   28  -14   22   16  -8   -1    0
Ash   -25   -2 100  -76  -21  -47  -28    2  -19  -13 -18  -23   15
Aloa  -21   -9 -76  100  -16  -36  -27    2  -26  -16  -3   -4   10
Mgns    8   17 -21  -16  100    4  -57   51  -15  -10   0   22  -19
Ttlp  -21   16 -47  -36    4  100  -24  -33  -62  -34   3  -20   -4
Flvn   -8   28 -28  -27  -57  -24  100   63  -41  -37  29   43   25
Nnfp   -4  -14   2    2   51  -33   63  100  -17   -3 -15  -31  -20
Prnt  -38   22 -19  -26  -15  -62  -41  -17  100  -68  42   13  -20
Clri  -35   16 -13  -16  -10  -34  -37   -3  -68  100  57   10  -12
Hue     3   -8 -18   -3    0    3   29  -15   42   57 100  -36    0
Oodw  -13   -1 -23   -4   22  -20   43  -31   13   10 -36  100  -20
Prln    9    0  15   10  -19   -4   25  -20  -20  -12   0  -20  100
>
```

Le tabelle presentate forniscono informazioni importanti sulle dipendenze condizionali tra le variabili. Si osserva come tali dipendenze differiscano se consideriamo l'intera popolazione o una singola cultivar. I valori più vicini a zero indicano una quasi-indipendenza condizionale, mentre valori più elevati segnalano una relazione più forte. Queste informazioni risultano fondamentali per la successiva costruzione dei grafi indiretti, come trattato nella sezione 4. Tenendo in considerazione la nostra variabile target **Clri** ci aspettiamo che nell'*intera popolazione* possa esserci una dipendenza con **Alch**, **Hue** e **Oodw** e che sia indipendente rispetto a **Prnt** e **Aloa**. Per le diverse cultivar invece ci aspettiamo:

- **v1**: Dipendenza: **Alch**, **Ttlp**, **Flvn**, **Prnt** e **Prln**. Indipendenza: **Hue** e **Ash**
- **v2**: Dipendenza: **Flvn** e **Alch**. Indipendenza: **Nnfp**, **Hue** e **Mgns**.
- **v3**: Dipendenza: **Prnt**, **Hue** e **Alch**. Indipendenza: **Nnfp**, **Mgns** e **Oodw**.

3.5. Heatmap delle correlazioni

In questa sezione analizziamo le correlazioni tra le variabili tramite heatmap, che permettono di visualizzare in modo immediato l'intensità e la direzione delle associazioni grazie a una scala cromatica. Valori elevati indicano una correlazione positiva, cioè che l'aumento di una variabile è associato all'aumento dell'altra, mentre valori bassi indicano una correlazione negativa, ossia che l'aumento di una variabile è associato alla diminuzione dell'altra. La correlazione misura la relazione lineare tra due variabili senza considerare l'influenza delle altre, ed è utile per un'esplorazione visiva preliminare. Anche in questo caso, l'analisi viene svolta sia sull'intera popolazione sia suddividendo i dati per cultivar, in modo da evidenziare eventuali differenze nella struttura delle relazioni tra i gruppi.

```
# Matrice di correlazione
#intera popolazione
corr <- cor(wine_dataset)

# Matrici di correlazione per ciascuna cultivar
corr1 <- cor(v1)
corr2 <- cor(v2)
corr3 <- cor(v3)

#heatmap
# install.packages("pheatmap")
library(pheatmap)
# Intera popolazione
pheatmap(corr,
         main = "Heatmap delle correlazioni - Popolazione",
         fontsize = 10)

# Per ciascuna cultivar
pheatmap(corr1, main = "Cultivar v1")
pheatmap(corr2, main = "Cultivar v2")
pheatmap(corr3, main = "Cultivar v3")
```

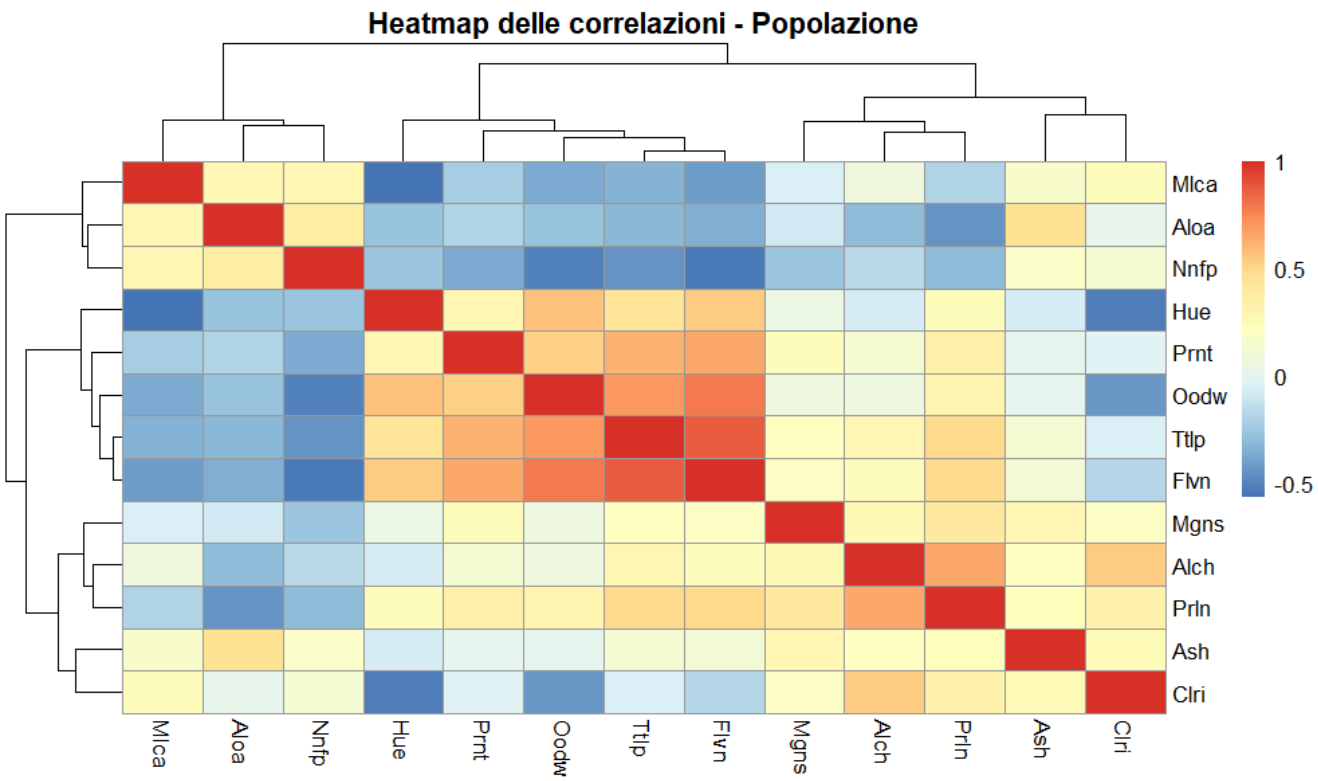


Figura 7: Heatmap correlazioni intera popolazione

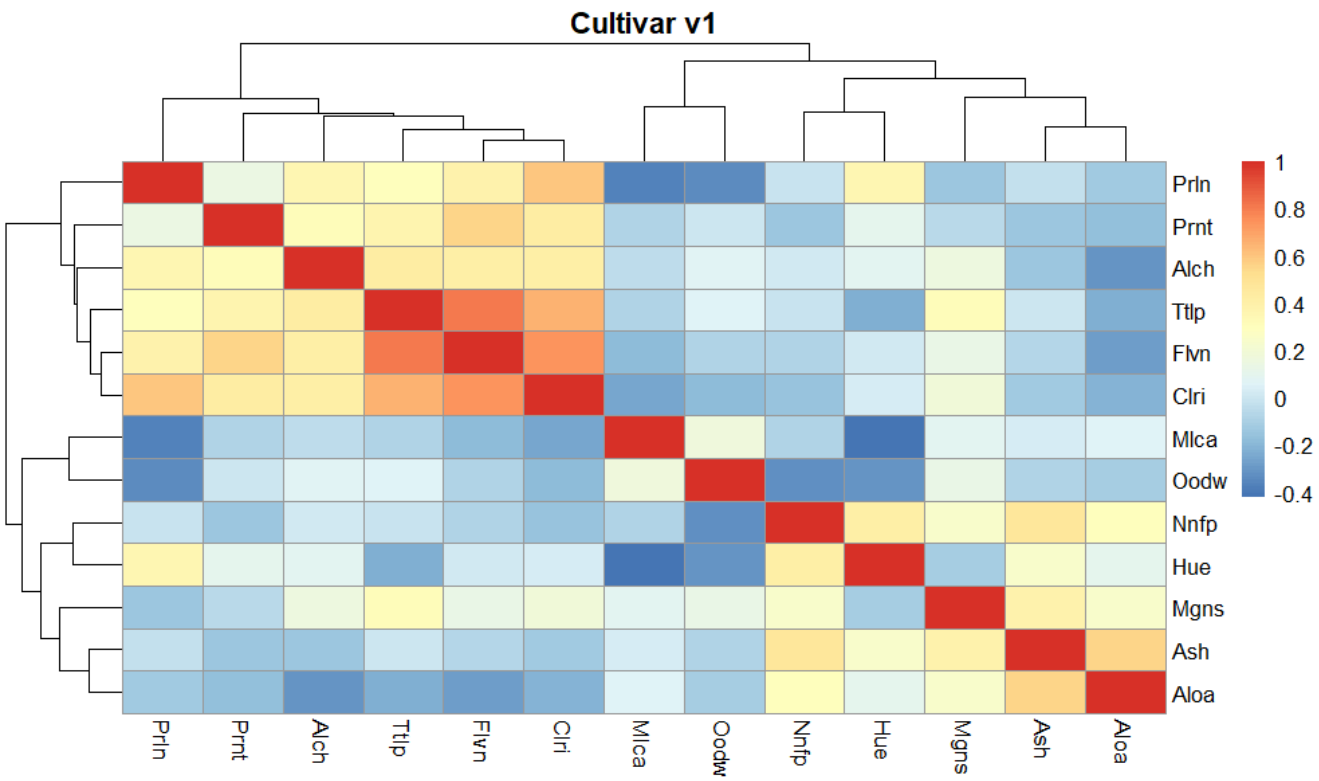


Figura 8: Heatmap correlazioni cultivar v1

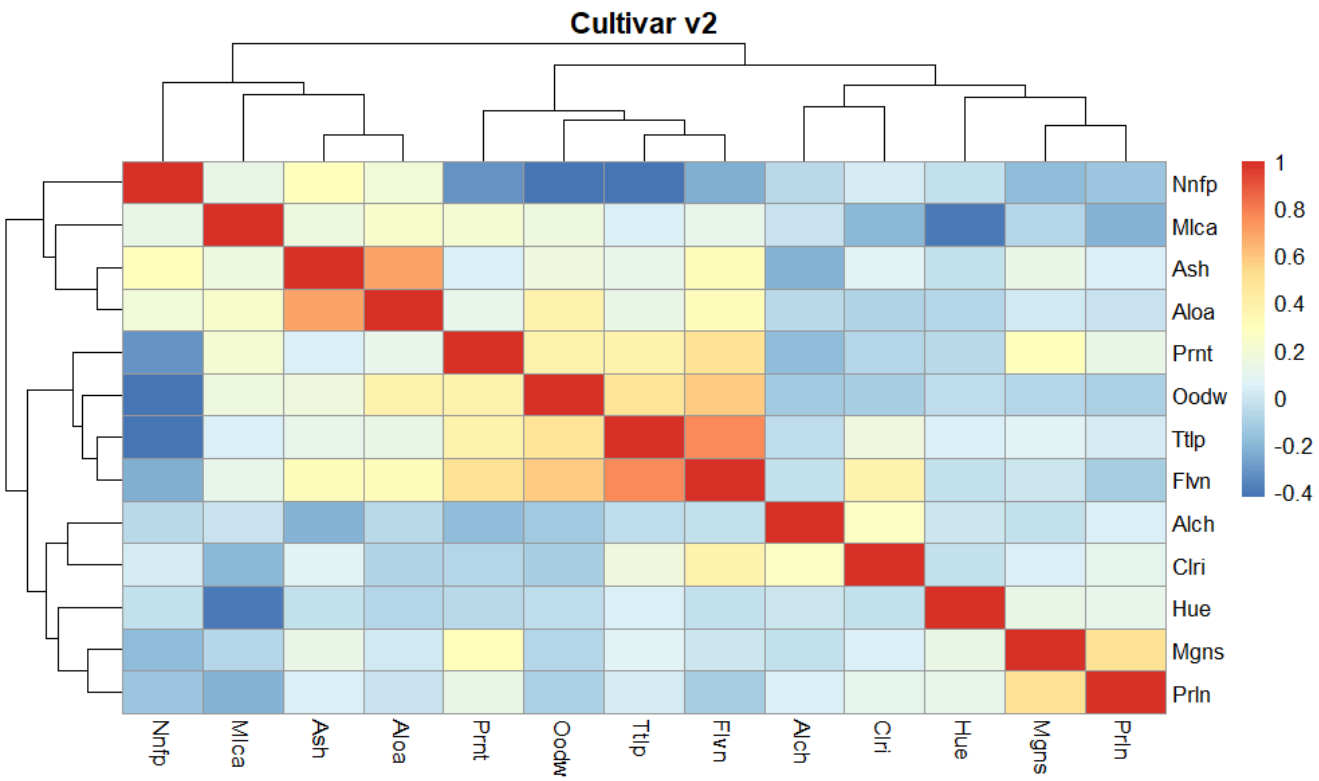


Figura 9: Heatmap correlazioni cultivar v2

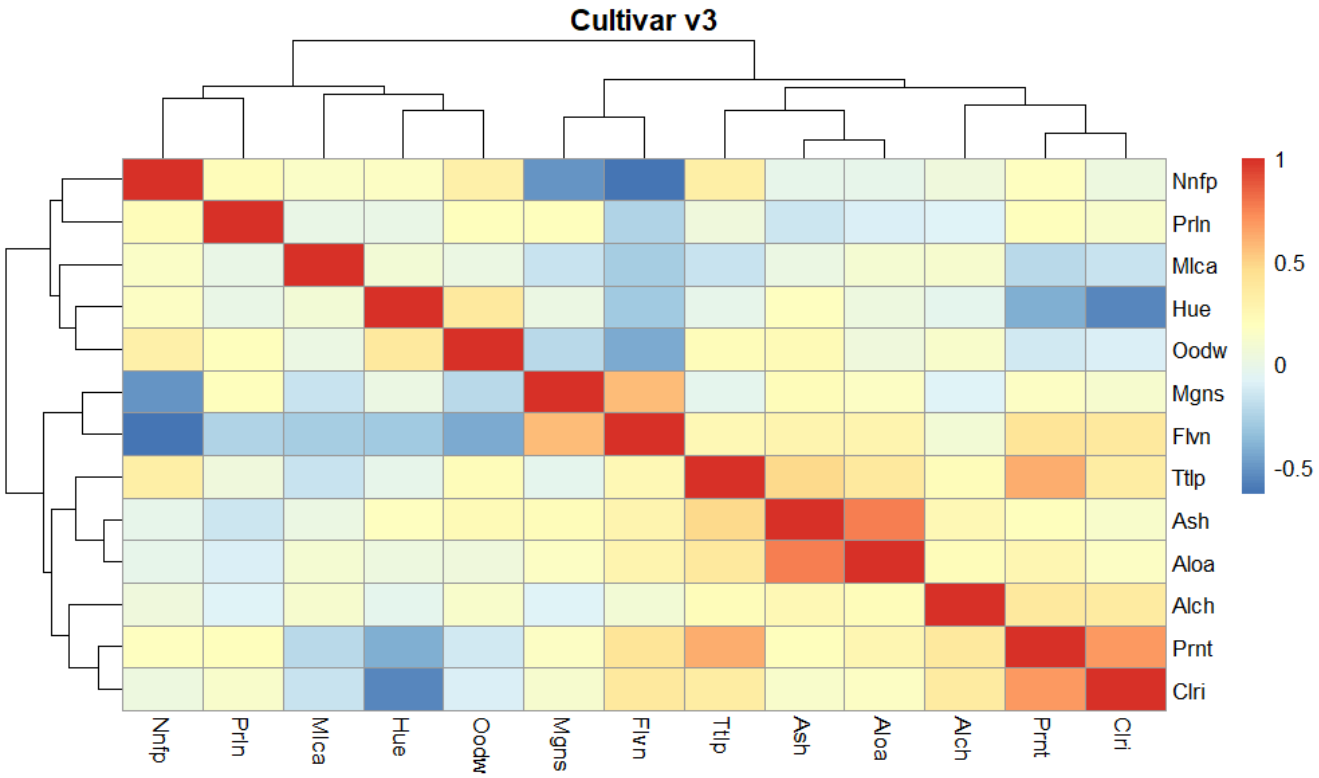


Figura 10: Heatmap correlazioni cultivar v3

Dalle heatmap e dalle sottosezione 3.1, sottosezione 3.2, sottosezione 3.3 e sottosezione 3.4, emergono differenze rilevanti nella struttura delle correlazioni. Nella popolazione complessiva si osservano relazioni ben definite tra alcune variabili chimico-fisiche, mentre l'analisi separata per cultivar mostra che tali associazioni non sono uniformi nei tre gruppi. In particolare, alcune correlazioni risultano più marcate o attenuate a seconda della cultivar, suggerendo che il processo produttivo e le caratteristiche del vitigno influenzano non solo i valori delle singole variabili, ma anche le relazioni tra esse. Queste evidenze confermano quindi l'importanza di considerare la variabile Cult nelle fasi successive dell'analisi.

4. Modelli Grafici Indiretti

I modelli grafici sono modelli probabilistici per rappresentare variabili casuali e le loro dipendenze tramite un **grafo non diretto**. Consentono un'analisi multivariata di sistemi complessi e riducono la dimensionalità del problema. I nodi rappresentano le variabili aleatorie, mentre gli archi indicano le dipendenze condizionali tra di esse. Un fatto molto utile è che l'assenza di un arco implica un'*indipendenza condizionale* tra le variabili corrispondenti. Nel nostro contesto dato che lavoriamo con variabili continue tale grafico può essere associato a una distribuzione *Gaussiana Multivariata* dove le dipendenze condizionate sono mostrate dall'inversa della matrice di covarianza e varianza ($\sum^{-1}$) e in particolare i valori zero nella matrice indicano un'indipendenza condizionale. L'obiettivo della nostra ricerca è trovare grafici parsimoniosi che però bilanciano l'adattamento ai dati.
Per usare su R tali grafici è necessario installare i dovuti pacchetti qui sotto mostrati.

```
#-------------------------------------
#installazioni pacchetti
install.packages("gRbase")
```

```
install.packages("gRain")
install.packages("gRim")
# pacchetti per i modelli grafi indiretti
library(gRbase)
library(gRain)
library(gRim)
```

Su R questi grafici vengono ottenuti grazie a due funzioni particolari

- **cmod()** serve a definire un modello grafico gaussiano, specificato tramite una formula che indica quali interazioni (condizionali) sono presenti, definendole la struttura, in particolare *il modello di indipendenza* (`~ .^1`) e *il modello saturo* (`~ .^`)
- **stepwise()** che ci permette un modo adeguato di identificare un modello grazie a un'esplorazione iterativa dello spazio dei modelli aggiungendo (*forward*) o rimuovendo (*backward*) interazioni.

4.1. Selezione Modello

La selezione del modello ottimale nei modelli grafici indiretti può essere effettuata tramite criteri di penalizzazione come l'**AIC** (Akaike Information Criterion) e il **BIC** (Bayesian Information Criterion). Questi criteri bilanciano la bontà di adattamento del modello ai dati con la complessità del modello stesso, penalizzando la presenza di un numero elevato di parametri e riducendo così il rischio di overfitting.
In particolare, dati un modello con log-verosimiglianza $\ell(\hat{\theta})$ e numero di parametri k, i criteri sono definiti come:

$$\text{AIC} = -2\ell(\hat{\theta}) + 2k \tag{1}$$

$$\text{BIC} = -2\ell(\hat{\theta}) + k\log(n) \tag{2}$$

dove n è il numero di osservazioni.

Nella funzione *stepwise()* è possibile specificare il tipo di penalità tramite il parametro `k`. Inoltre, grazie al parametro `direction`, è possibile scegliere la direzione dell'esplorazione iterativa, selezionando tra *forward*, *backward* o *both*. Quest'ultima opzione combina le due procedure, permettendo una ricerca più flessibile che sfrutta i vantaggi di entrambe.

4.1.1 Graphical Lasso

Dato che lavoriamo con un dataset di valori continui è importante menzionare il **Graphical Lasso** (glasso) [2]. Il Glasso è una procedura per la selezione del modello nei grafi gaussiani. Il metodo è rapido e si basa sulla massimizzazione della log-verosimiglianza penalizzata dalla norma L1 della matrice di concentrazione K.
La funzione è:

$$L_{\text{pen}}(K, \hat{\mu}) = \log\det(K) - \text{tr}(KS) - \rho\|K\|_1 \tag{3}$$

dove $\rho \geq 0$ è un parametro di penalizzazione che controlla la sparsità del grafo: valori più alti favoriscono grafi più sparsi e facilmente interpretabili. I valori di ρ sono stati scelti sperimentalmente per ottenere grafi chiari e interpretabili. Inoltre, questa procedura è stata applicata solo nei casi in cui **AIC** e **BIC** non erano sufficienti, come nel caso illustrato nella sottosezione 4.2.

4.2. Grafo indiretto sull'intera popolazione

In questa sezione vengono presentati il codice e i grafi indiretti ottenuti tramite le diverse procedure applicate al dataset relativo all'intera popolazione di vini.

```
#Intera popolazione
#creazione modello saturo
pop_mod_sat <- cmod(~ .^., data=wine_dataset)

#creazione modello di indipendenza
pop_mod_ind <- cmod(~.^1, data=wine_dataset)

#grafico usando penalizzazione AIC Forward
AIC_pop_F <- stepwise(pop_mod_ind, direction="forward")
plot(AIC_pop_F, "neato")
title(main="UG Forward AIC intera popolazione")

#grafico usando AIC Backward
AIC_pop_B <- stepwise(pop_mod_sat)
plot(AIC_pop_B, "neato")
title(main = "UG Backward AIC intera popolazione")

#grafico usando BIC forward
BIC_pop_F <- stepwise(pop_mod_ind, direction = "forward", k=log(nrow(wine_dataset)))
plot(BIC_pop_F, "neato")
title(main = "UG Forward BIC intera popolazione")

#grafico usando BIC backward
BIC_pop_B <- stepwise(pop_mod_sat, k=log(nrow(wine_dataset)))
plot(BIC_pop_B)
title(main="UG Backward BIC intera popolazione")

#Both directions
AIC_pop_FB <- stepwise(pop_mod_ind, direction="both")
BIC_pop_FB <- stepwise(pop_mod_ind, direction="both", k=log(nrow(wine_dataset)))

#grafico usando AIC Both
plot(AIC_pop_FB)
title(main = "UG Both AIC intera popolazione")

#grafico usando BIC Both
plot(BIC_pop_FB)
title(main="UG Both BIC intera popolazione")

#procedura glasso

#pacchetti da installare
#install.packages("glasso")
#install.packages("igraph")
library(glasso)
library(igraph)

# Matrice di correlazione
popCor <- cov2cor(CovPop)
```

```r
# Graphical Lasso
pop_lasso <- glasso(popCor, rho = 0.3)

# Matrice di adiacenza booleana
AM <- pop_lasso$wi != 0
diag(AM) <- FALSE

# Costruzione del grafo igraph
graf_lasso <- graph_from_adjacency_matrix(
  AM,
  mode = "undirected",
  diag = FALSE
)

# Etichette dei nodi
V(graf_lasso)$name <- colnames(wine_dataset)

plot(main="UG glasso intera popolazione",
  graf_lasso,
  layout = layout_with_kk,
)
```

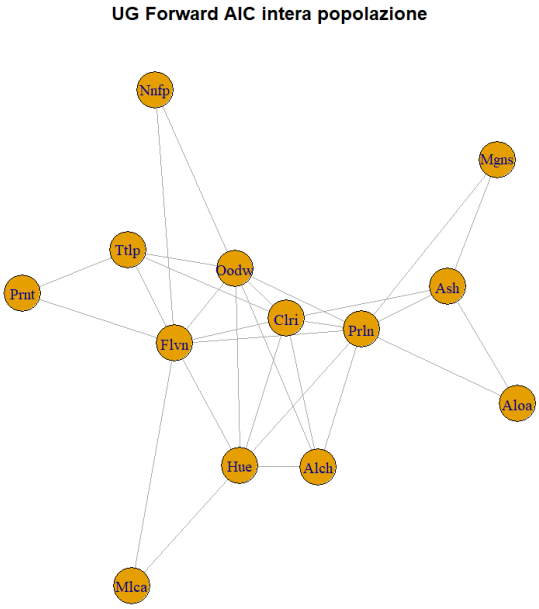


Figura 11: UG Forward AIC intera popolazione

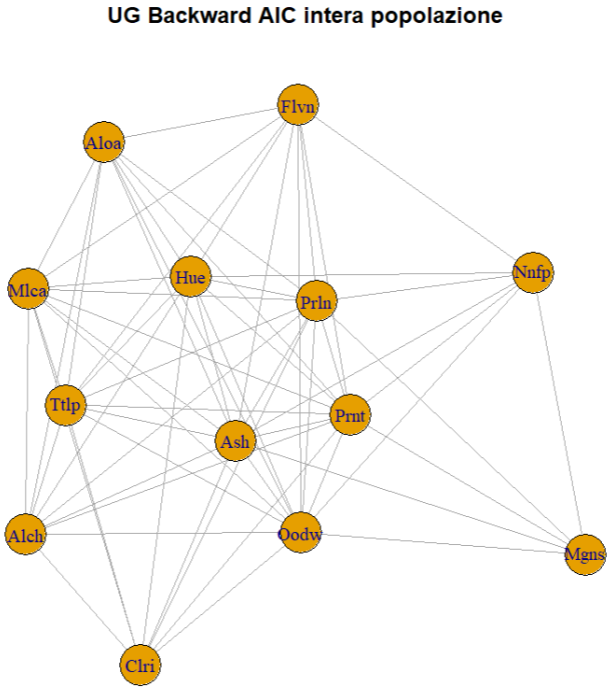


Figura 12: UG Backward AIC intera popolazione

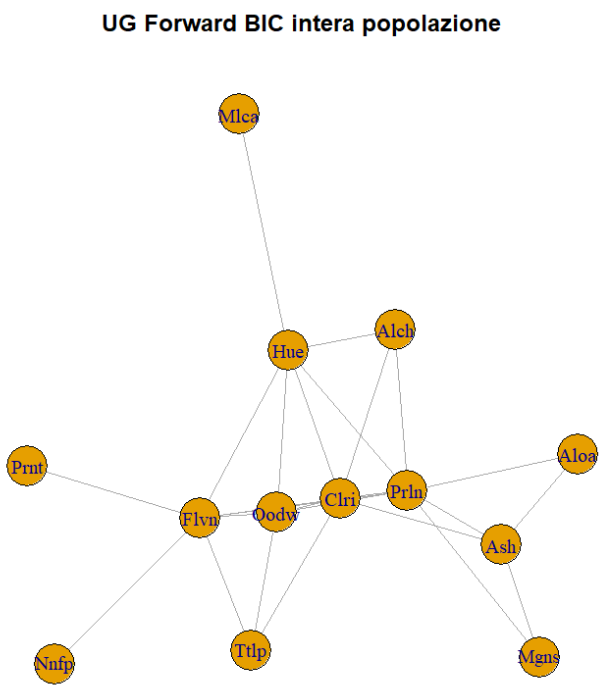


Figura 13: UG Forward BIC intera popolazione

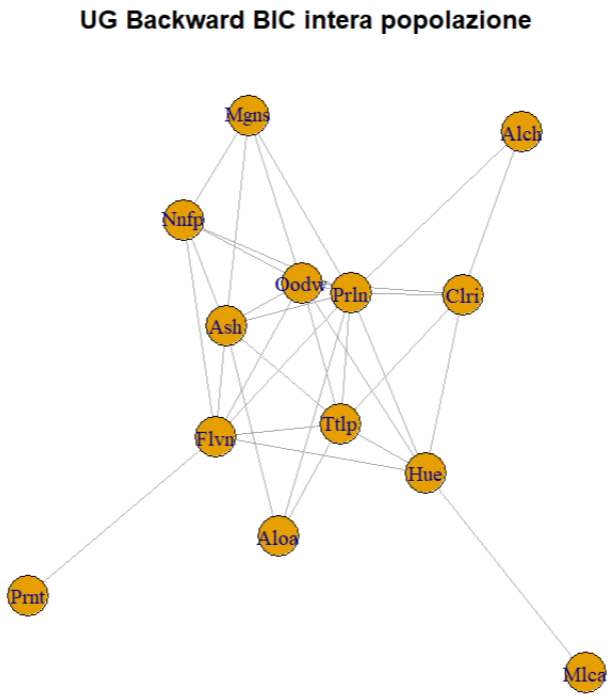


Figura 14: UG Backward BIC intera popolazione

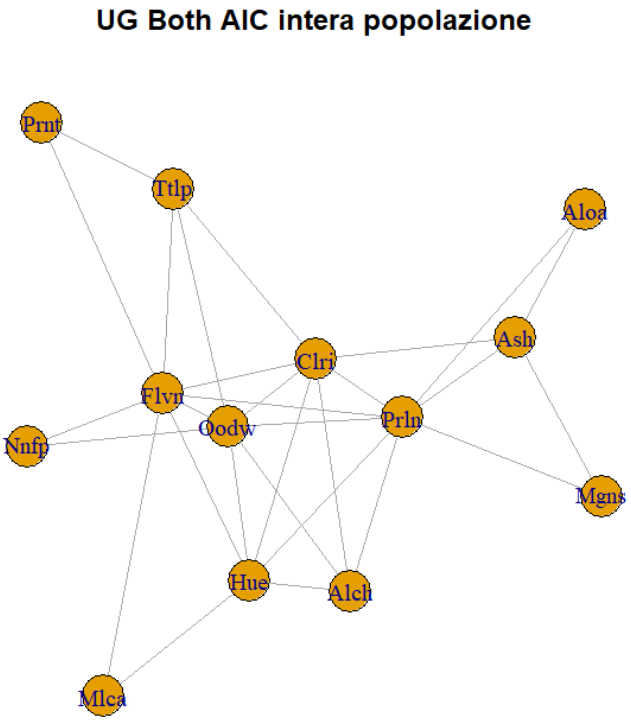


Figura 15: UG Both AIC intera popolazione

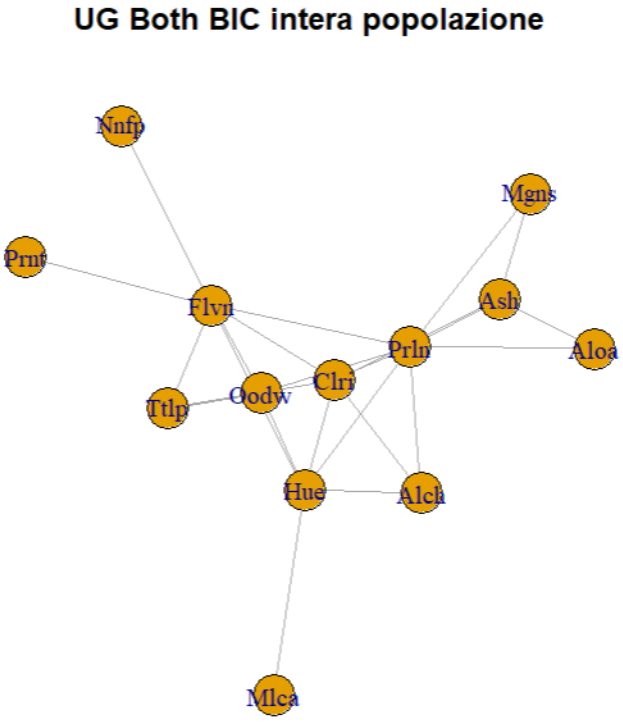


Figura 16: UG Both BIC intera popolazione

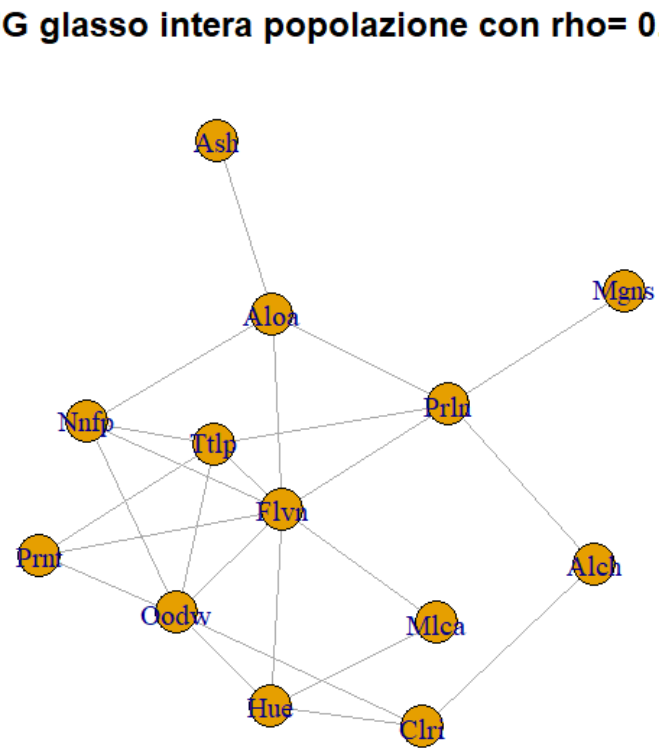


Figura 17: UG usando procedura Glasso

4.3. Grafi indiretti per il tipo di cultivar

In questa sezione tratteremo i grafi ottenuti per ogni singolo cultivar. Qui di seguito mostreremo il codice utilizzato.

```
#----------------------------------------
#V1
v1_mod_sat <- cmod(~ .^., data=v1)
v1_mod_ind <- cmod(~.^1, data=v1)

#grafico usando penalizzazione AIC Forward
AIC_v1_F <- stepwise(v1_mod_ind, direction="forward")
plot(AIC_v1_F, "neato")
title(main="UG Forward AIC cultivar v1")

#grafico usando AIC Backward
AIC_v1_B <- stepwise(v1_mod_sat)
plot(AIC_v1_B, "neato")
title(main = "UG Backward AIC cultivar v1")

#grafico usando BIC forward
BIC_v1_F <- stepwise(v1_mod_ind, direction = "forward", k=log(nrow(v1)))
plot(BIC_v1_F, "neato")
title(main = "UG Forward BIC cultivar v1")

#grafico usando BIC backward
```

```
BIC_v1_B <- stepwise(v1_mod_sat, k=log(nrow(v1)))
plot(BIC_v1_B)
title(main="UG Backward BIC cultivar v1")

#Both directions
AIC_v1_FB <- stepwise(v1_mod_ind, direction="both")
BIC_v1_FB <- stepwise(v1_mod_ind, k=log(nrow(v1)), direction="both")

#grafico AIC both
plot(AIC_v1_FB)
title(main = "UG Both AIC cultivar v1")

#grafico BIC both
plot(BIC_v1_FB)
title(main = "UG Both BIC cultivar v1")
#---------------------------------------
#V2
v2_mod_sat <- cmod(~ .^., data=v2)
v2_mod_ind <- cmod(~.^1, data=v2)

#garfico usando penalizzazione AIC Forward
AIC_v2_F <- stepwise(v2_mod_ind, direction="forward")
plot(AIC_v2_F, "neato")
title(main="UG Forward AIC cultivar v2")

#grafico usando AIC Backward
AIC_v2_B <- stepwise(v2_mod_sat)
plot(AIC_v2_B, "neato")
title(main = "UG Backward AIC cultivar v2")

#grafico usando BIC forward
BIC_v2_F <- stepwise(v2_mod_ind, direction = "forward", k=log(nrow(v2)))
plot(BIC_v2_F, "neato")
title(main = "UG Forward BIC cultivar v2")

#grafico usando BIC backward
BIC_v2_B <- stepwise(v2_mod_sat, k=log(nrow(v2)))
plot(BIC_v2_B)
title(main="UG Backward BIC cultivar v2")

#Both directions
AIC_v2_FB <- stepwise(v2_mod_ind, direction="both")
BIC_v2_FB <- stepwise(v2_mod_ind, k=log(nrow(v2)), direction="both")

#grafico AIC both
plot(AIC_v2_FB)
title(main = "UG Both AIC cultivar v2")

#grafico BIC both
plot(BIC_v2_FB)
title(main = "UG Both BIC cultivar v2")

#----------------------------------------
```

```r
#V3
v3_mod_sat <- cmod(~ .^., data=v3)
v3_mod_ind <- cmod(~.^1, data=v3)

#garfico usando penalizzazione AIC Forward
AIC_v3_F <- stepwise(v3_mod_ind, direction="forward")
plot(AIC_v3_F, "neato")
title(main="UG Forward AIC cultivar v3")

#grafico usando AIC Backward
AIC_v3_B <- stepwise(v3_mod_sat)
plot(AIC_v3_B, "neato")
title(main = "UG Backward AIC cultivar v3")

#grafico usando BIC forward
BIC_v3_F <- stepwise(v3_mod_ind, direction = "forward", k=log(nrow(v3)))
plot(BIC_v3_F, "neato")
title(main = "UG Forward BIC cultivar v3")

#grafico usando BIC backward
BIC_v3_B <- stepwise(v3_mod_sat, k=log(nrow(v3)))
plot(BIC_v3_B)
title(main="UG Backward BIC cultivar v3")

#Both directions
AIC_v3_FB <- stepwise(v3_mod_ind, direction="both")
BIC_v3_FB <- stepwise(v3_mod_ind, k=log(nrow(v3)), direction="both")

#grafico AIC both
plot(AIC_v3_FB)
title(main = "UG Both AIC cultivar v3")

#grafico BIC both
plot(BIC_v3_FB)
title(main = "UG Both BIC cultivar v3")
```

4.3.1 Grafici V1

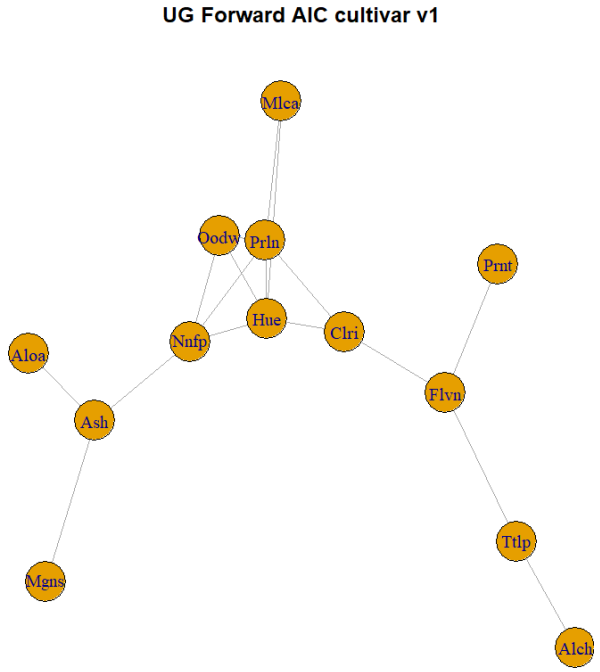


Figura 18: UG Forward AIC cultivar v1

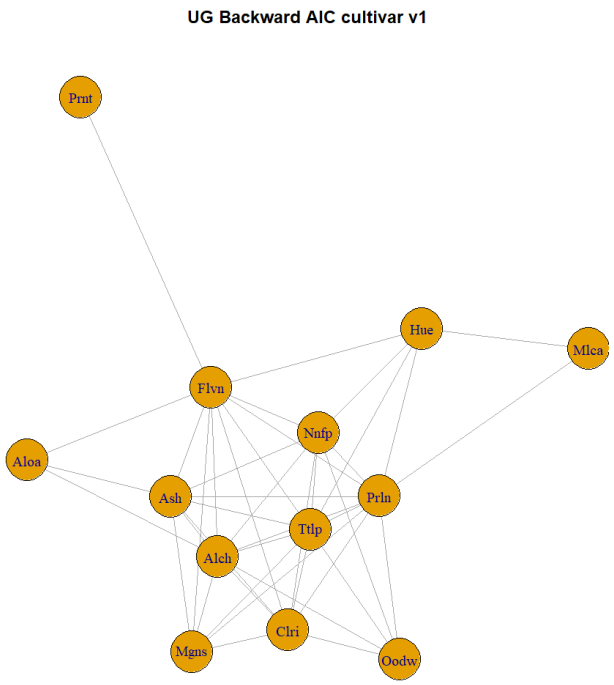


Figura 19: UG Backward AIC cultivar v1

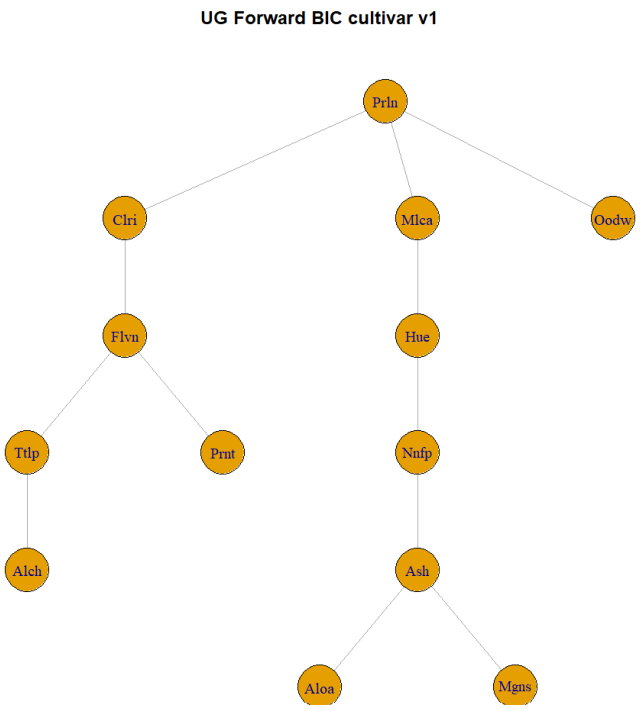


Figura 20: UG Forward BIC cultivar v1

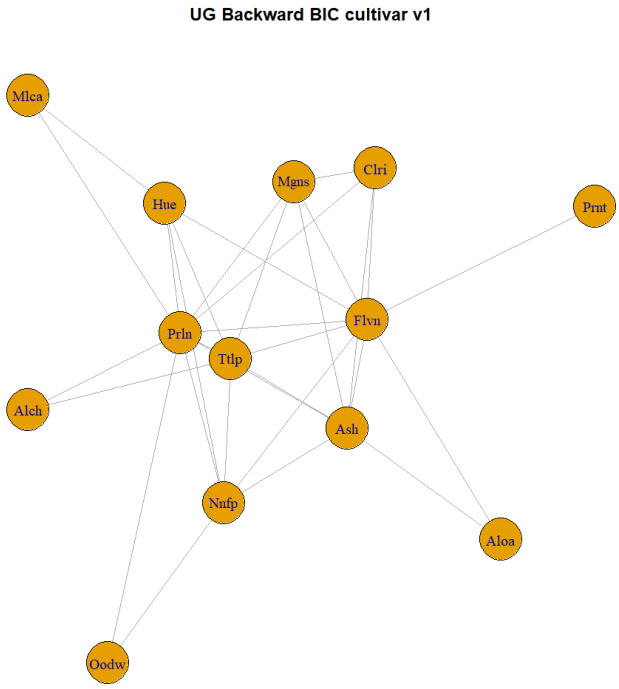


Figura 21: UG Backward BIC cultivar v1

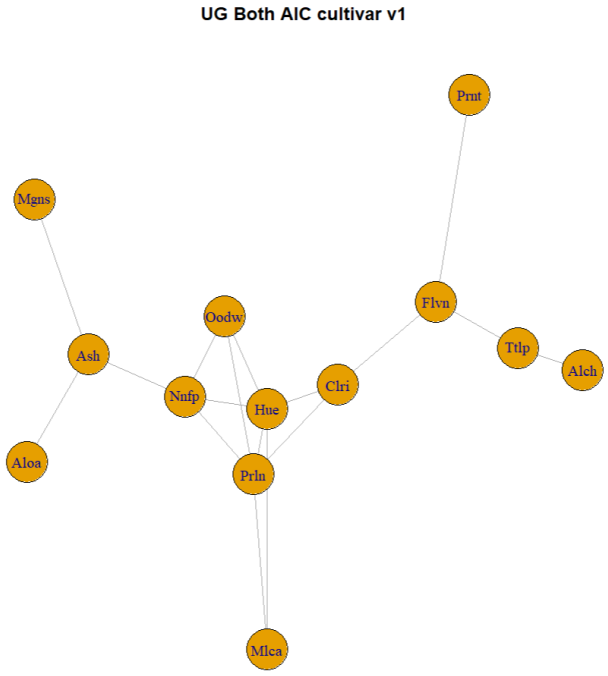


Figura 22: UG Both AIC cultivar v1

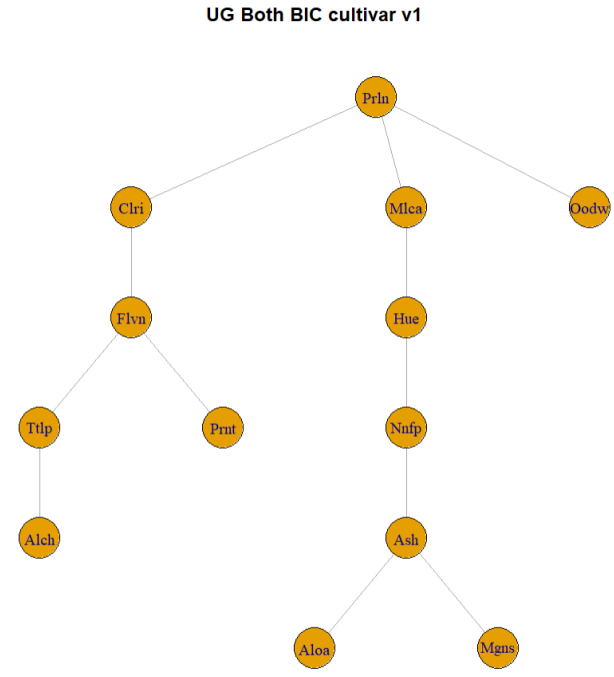


Figura 23: UG Both BIC cultivar v1

4.3.2 Grafici V2

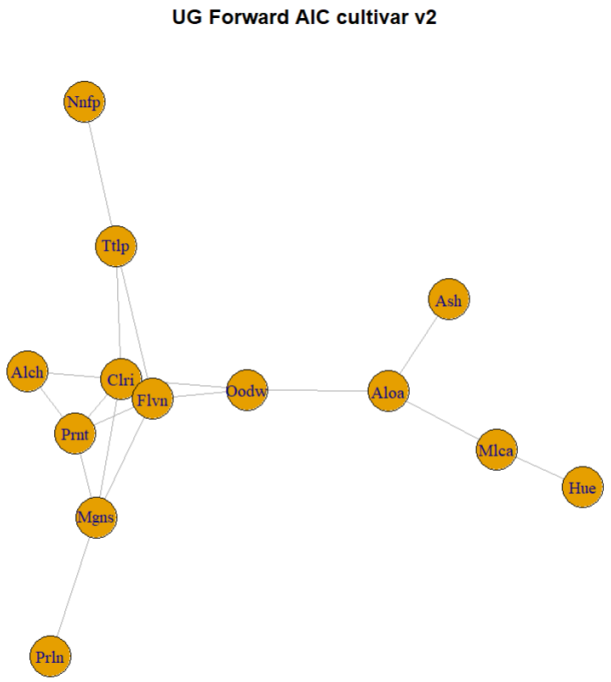


Figura 24: UG Forward AIC cultivar v2

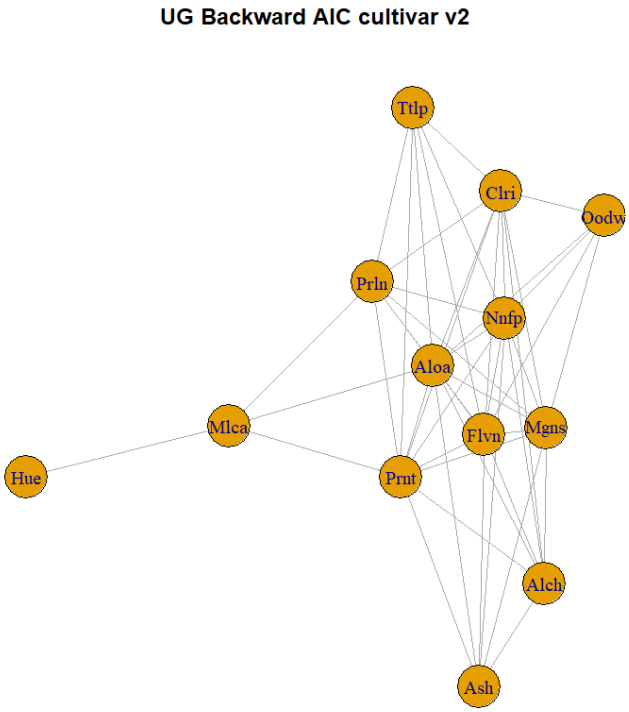


Figura 25: UG Backward AIC cultivar v2

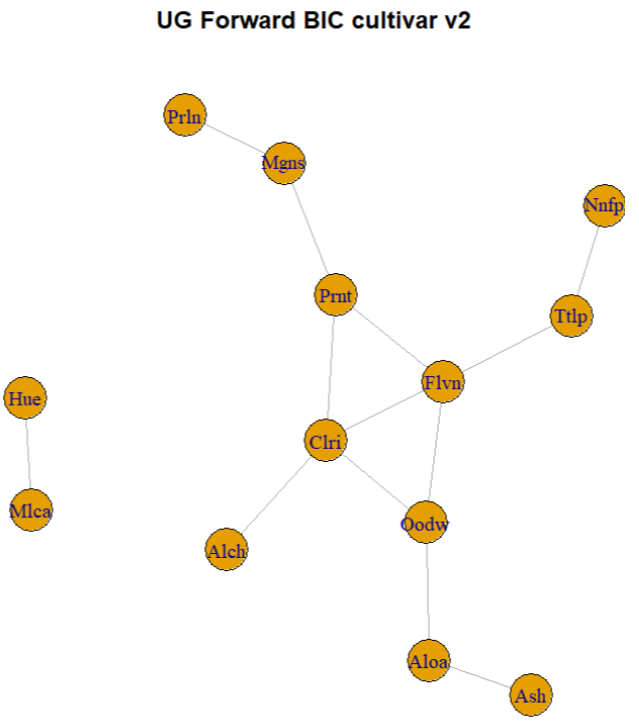


Figura 26: UG Forward BIC cultivar v2

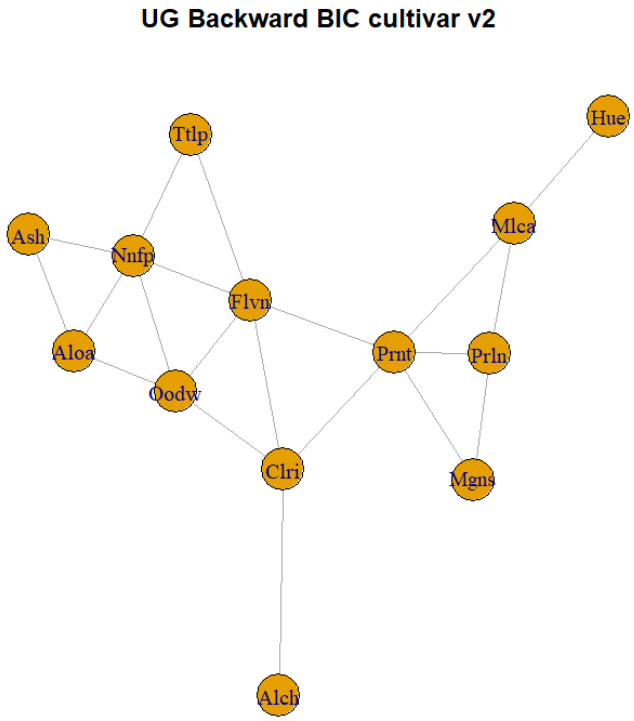


Figura 27: UG Backward BIC cultivar v2

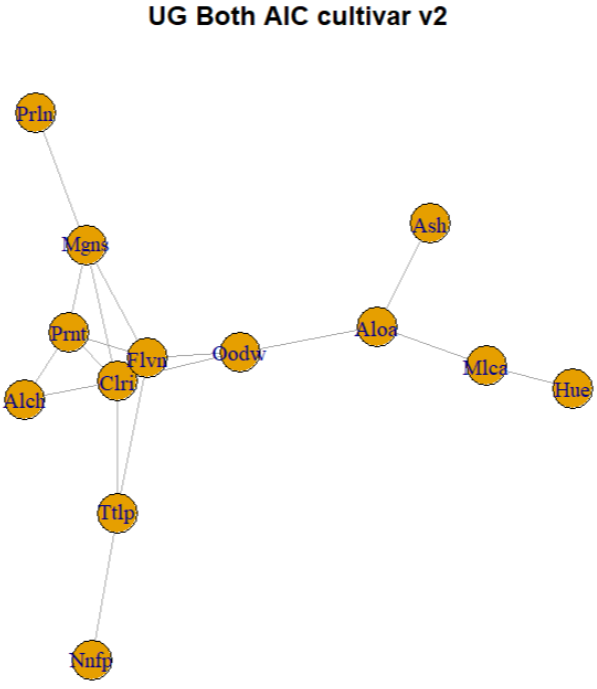


Figura 28: UG Both AIC cultivar v2

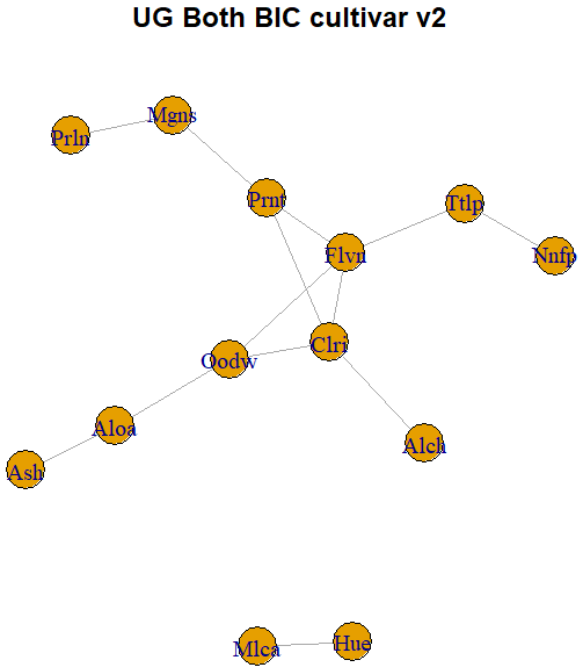


Figura 29: UG Both BIC cultivar v2

4.3.3 Grafici V3

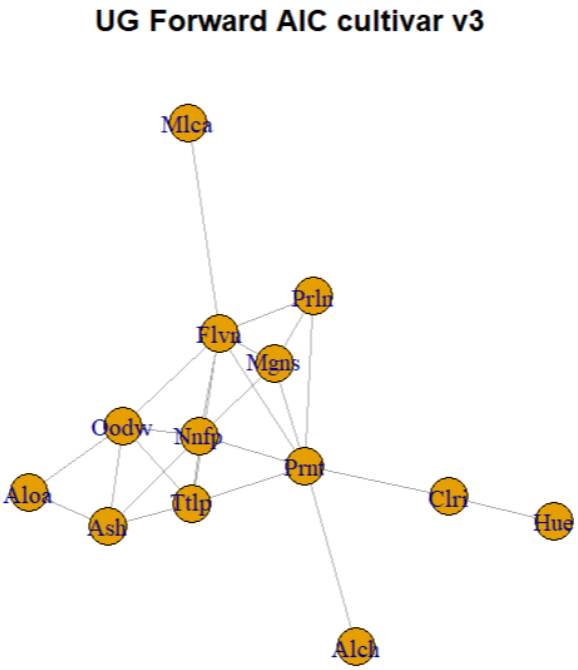


Figura 30: UG Forward AIC cultivar v3

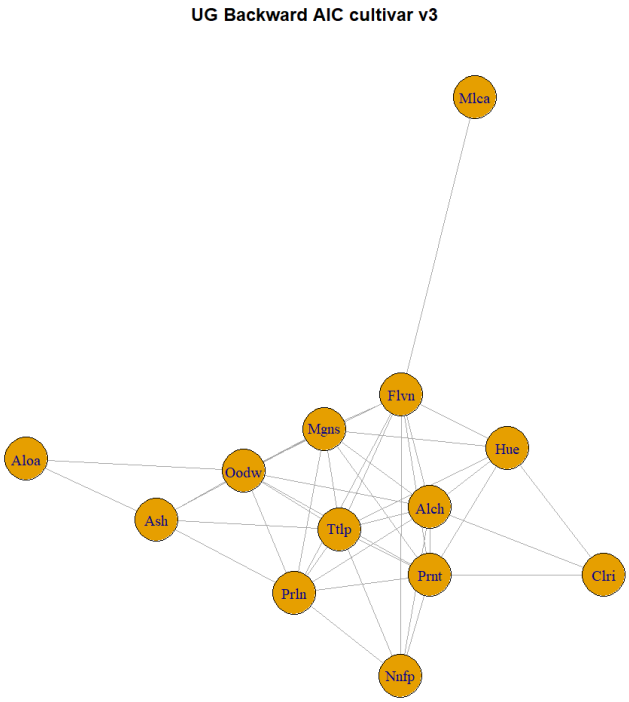


Figura 31: UG Backward AIC cultivar v2

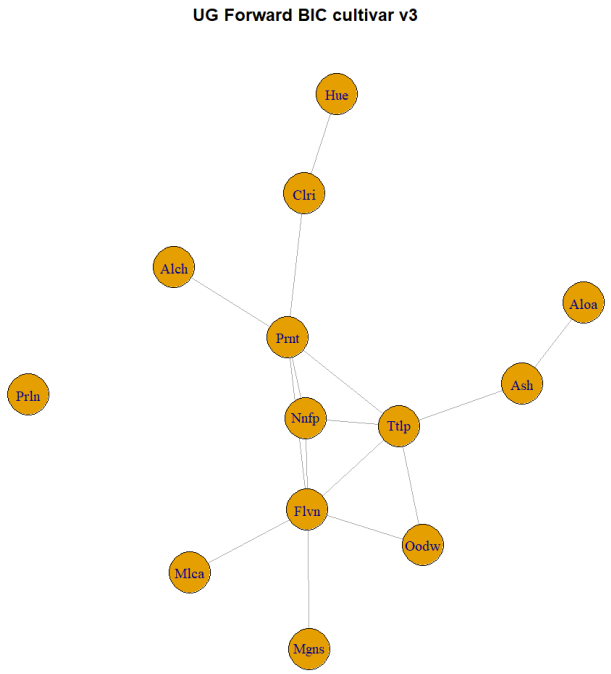


Figura 32: UG Forward BIC cultivar v3

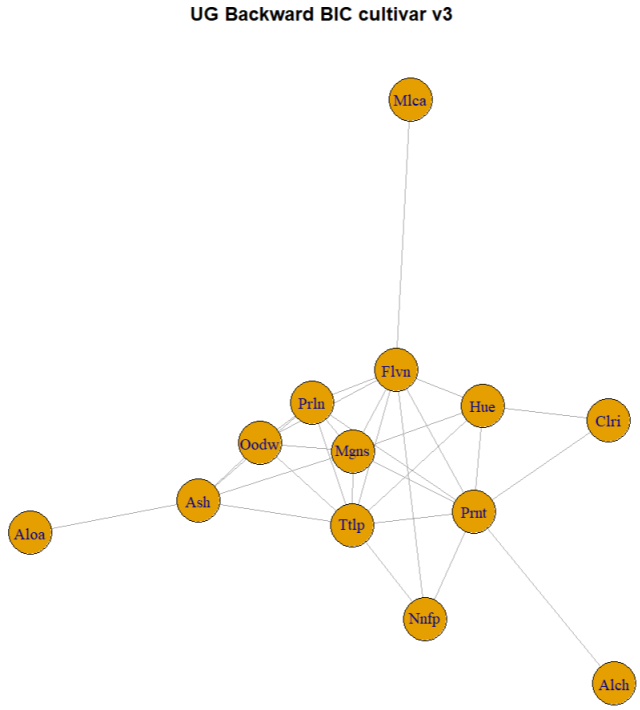


Figura 33: UG Backward BIC cultivar v3

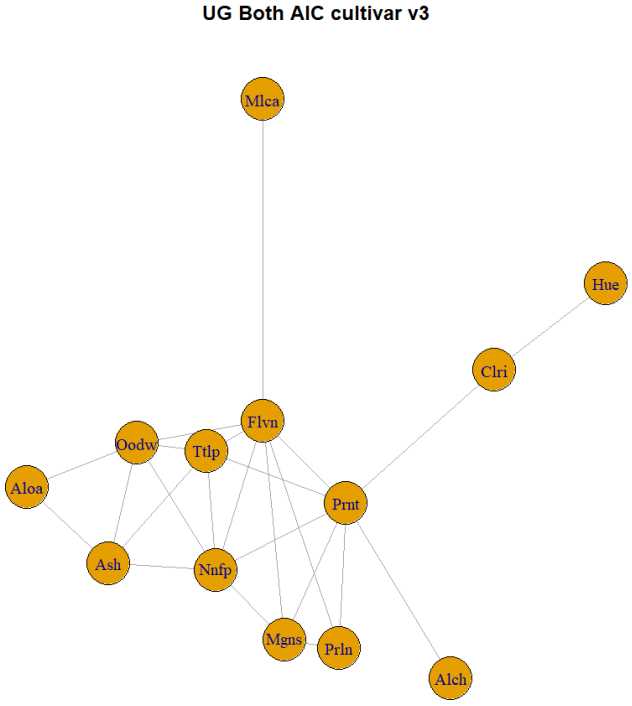


Figura 34: UG Both AIC cultivar v3

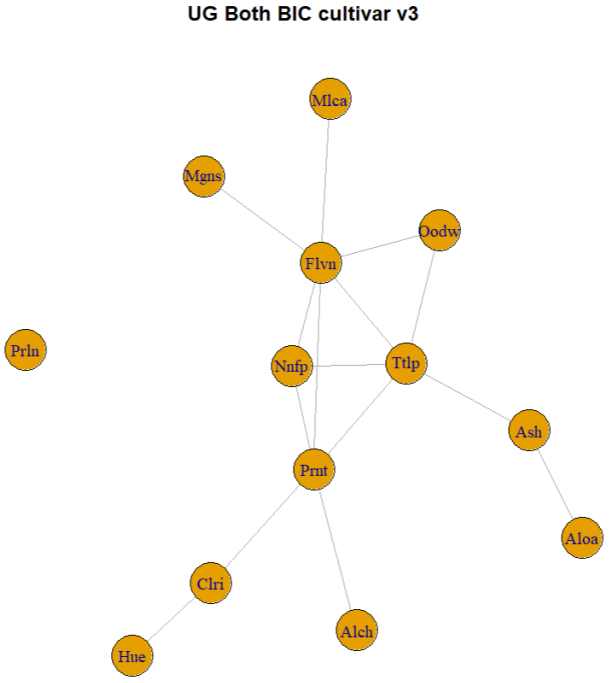


Figura 35: UG Both BIC cultivar v3

4.4. Risultati ottenuti

Dai grafi ottenuti si osserva che i modelli selezionati tramite **BIC** risultano più parsimoniosi rispetto a quelli ottenuti con penalizzazione **AIC**. Ciò avviene perché l'AIC tende a privilegiare modelli più complessi, migliorando

l'adattamento ai dati. Poiché l'obiettivo è individuare un modello che descriva al meglio il processo generativo dei dati, si preferisce adottare la penalizzazione BIC. Per quanto riguarda la struttura del grafo, si cerca di individuare un orientamento che contenga archi significativi, ossia presenti anche in altri grafi, mantenendo comunque la parsimonia del modello. L'unica eccezione è stata nel caso dell'intera popolazione dove per avere una maggiore chiarezza è stato necessario usare anche il metodo **Glasso**.
Sulla base di queste considerazioni e dell'analisi esplorativa precedente, sono stati selezionati i seguenti modelli:

- **Intera popolazione**: **Glasso** (Figura 17)
- **Cultivar v1**: **Both BIC** (Figura 23)
- **Cultivar v2**: **Forward BIC** (Figura 26)
- **Cultivar v3**: **Both BIC** (Figura 35)

5. Modelli Grafici Diretti

I modelli grafici diretti, noti anche come **DAG** (Directed Acyclic Graph), sono una classe di modelli probabilistici in cui le relazioni tra variabili sono rappresentate mediante archi orientati. A differenza dei grafi indiretti analizzati nella sezione precedente, i DAG permettono di modellare relazioni di tipo causale o condizionale diretto tra le variabili, fornendo informazioni sulla direzione delle dipendenze.

Un **DAG** è costituito da:

- **Nodi**: rappresentano le variabili aleatorie del sistema
- **Archi orientati**: indicano dipendenze dirette tra variabili, con una direzione che va dalla variabile "genitore" alla variabile "figlia"
- **Aciclicità**: non esistono percorsi che, seguendo la direzione degli archi, permettono di ritornare al nodo di partenza

Nel contesto della nostra analisi, i **DAG** ci permettono di:

- Identificare possibili relazioni causali tra le variabili chimiche del vino
- Comprendere come diverse sostanze influenzino l'intensità del colore (**Clri**)
- Confrontare le strutture di dipendenza tra le diverse cultivar

Dato che la struttura della rete è parzialmente sconosciuta utilizzeremo la funzione *hc()* che implementa un algoritmo per l'apprendimento della struttura che parte dal dataset e fornisce una struttura che massimizza un criterio di bonta' che può essere AIC o BIC. Ad ogni passo hill climb valuta piccole modifiche alla struttura, accettando quella che migliora maggiormente lo score, fino a raggiungere un ottimo locale.
Come metodo di valutazione dei DAG generati è stata usata la funzione *fitDag()* che prende in ingresso un grafo DAG e permette di calcolare la devianza e i gradi di libertà del modello, fornendo una misura statistica della bontà di adattamento. Su R è stato necessario installare alcuni pacchetti qui di seguito mostrati.

```r
#Grafi diretti (DAG)
install.packages("bnlearn")
install.packages("ggm")
install.packages("graph")

if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install("graph")

library(bnlearn)
library(graph)
library(ggm)
```

Nelle prossime sezioni presenteremo i grafici ottenuti con relativi valori effettuati sull'intera popolazione e per ogni tipo di cultivar usando esclusivamente il *BIC Score* continuo.

5.1. DAG Intera popolazione

```r
# Convertire tutte le colonne integer in numeric
wine_dataset <- data.frame(lapply(wine_dataset, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG intera popolazione BIC
DAG_pop <- hc(wine_dataset, score="bic-g")
plot(DAG_pop, main="DAG intera popolazione BIC Score")


DAG_pop <- amat(DAG_pop)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_pop) <- colnames(wine_dataset)
colnames(DAG_pop) <- colnames(wine_dataset)
fdag <- fitDag(DAG_pop, CovPop, nrow(wine_dataset))

fdag$dev
fdag$df
```

Da questo codice otteniamo come risultato

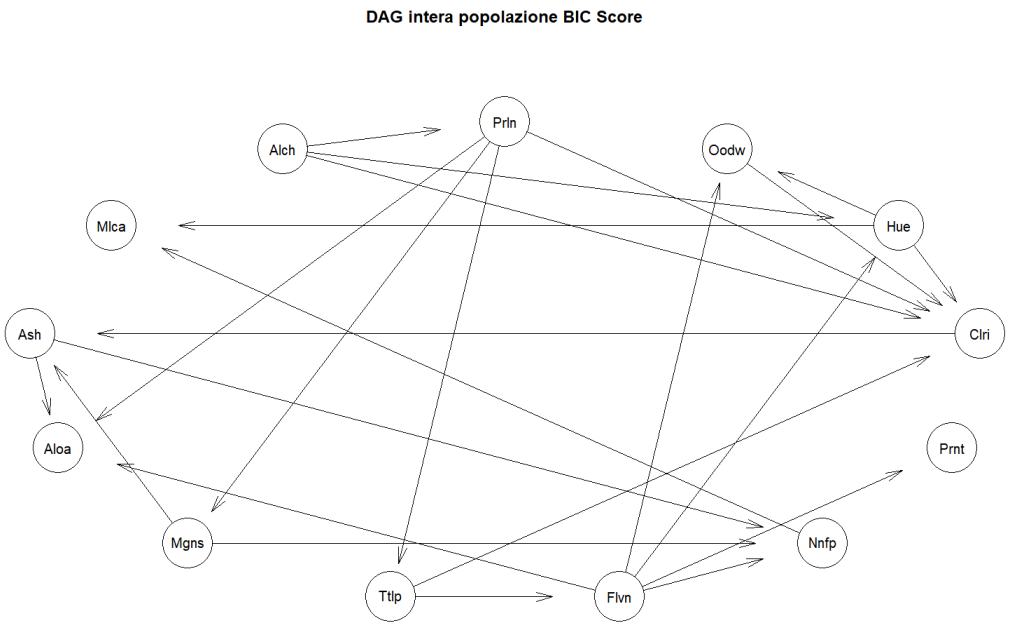


Figura 36: Grafico DAG sull'intera popolazione

```
> fdag$dev
[1] 79.04673
> fdag$df
[1] 54
```

5.2. DAG Cultivar

```
v1 <-data.frame(lapply(v1, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG cultivar v1 BIC
DAG_v1 <- hc(v1)
plot(DAG_v1, main="DAG cultivar v1")

DAG_v1 <- amat(DAG_v1)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v1) <- colnames(v1)
colnames(DAG_v1) <- colnames(v1)
fdag <- fitDag(DAG_v1, CovV1, nrow(v1))

fdag$dev
fdag$df

v2 <-data.frame(lapply(v2, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG cultivar v2 BIC
```

```
DAG_v2 <- hc(v2)
plot(DAG_v2, main="DAG cultivar v2")

DAG_v2 <- amat(DAG_v2)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v2) <- colnames(v2)
colnames(DAG_v2) <- colnames(v2)
fdag <- fitDag(DAG_v2, CovV2, nrow(v2))

fdag$dev
fdag$df

v3 <-data.frame(lapply(v3, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG cultivar v3 BIC
DAG_v3 <- hc(v3)
plot(DAG_v3, main="DAG cultivar v3")

DAG_v3 <- amat(DAG_v3)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v3) <- colnames(v3)
colnames(DAG_v3) <- colnames(v3)
fdag <- fitDag(DAG_v3, CovV3, nrow(v3))

fdag$dev
fdag$df
```

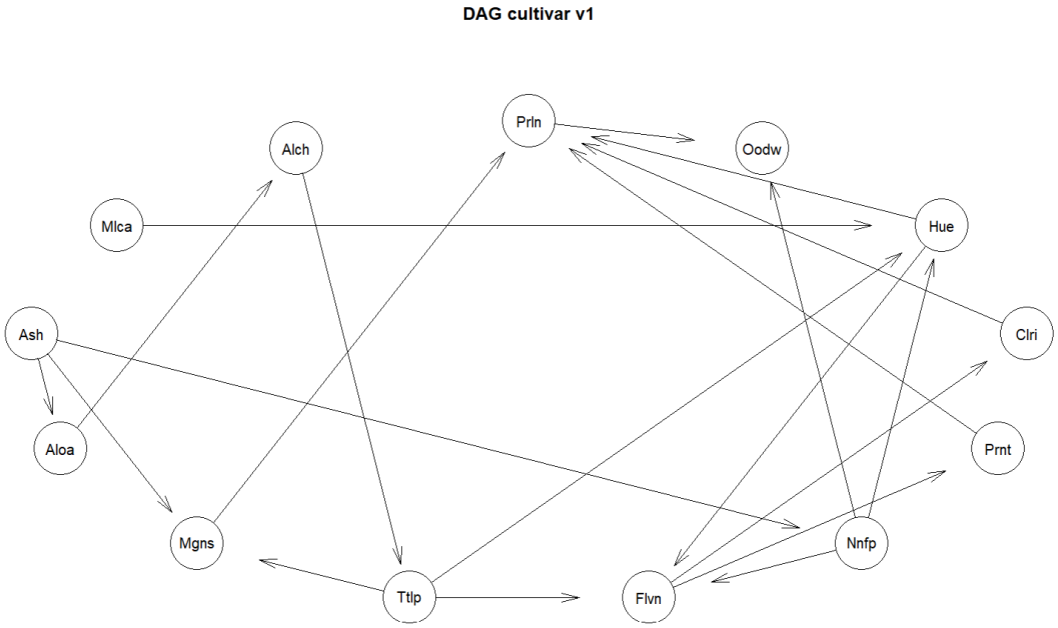


Figura 37: Grafico DAG sulla cultivar v1

```
> fdag$dev
[1] 51.79765
> fdag$df
[1] 58
```

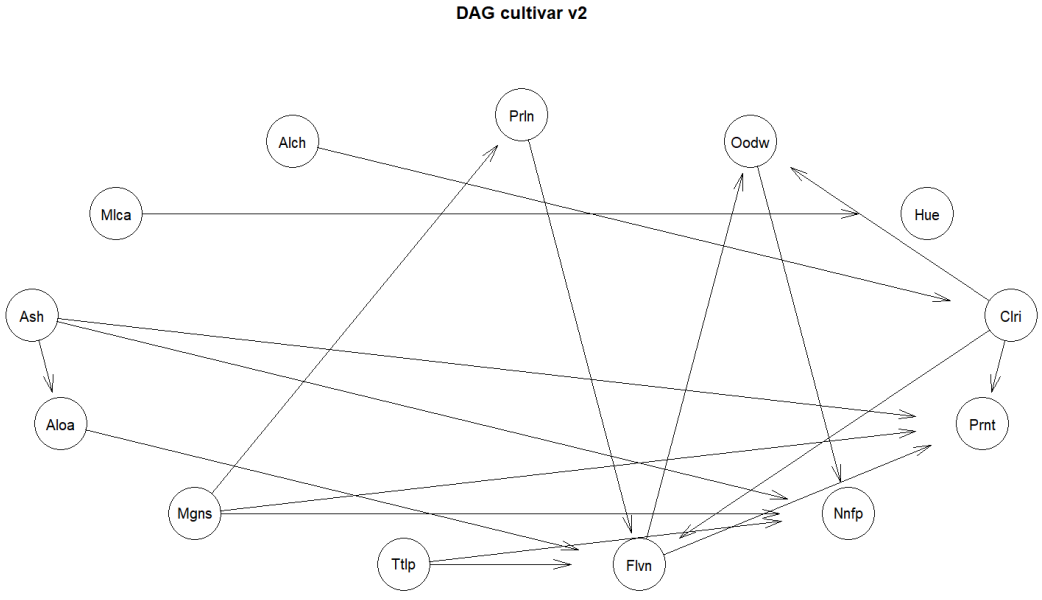


Figura 38: Grafico DAG sulla cultivar v2

```
> fdag$dev
[1] 60.26931
```

```
> fdag$df
[1] 60
```

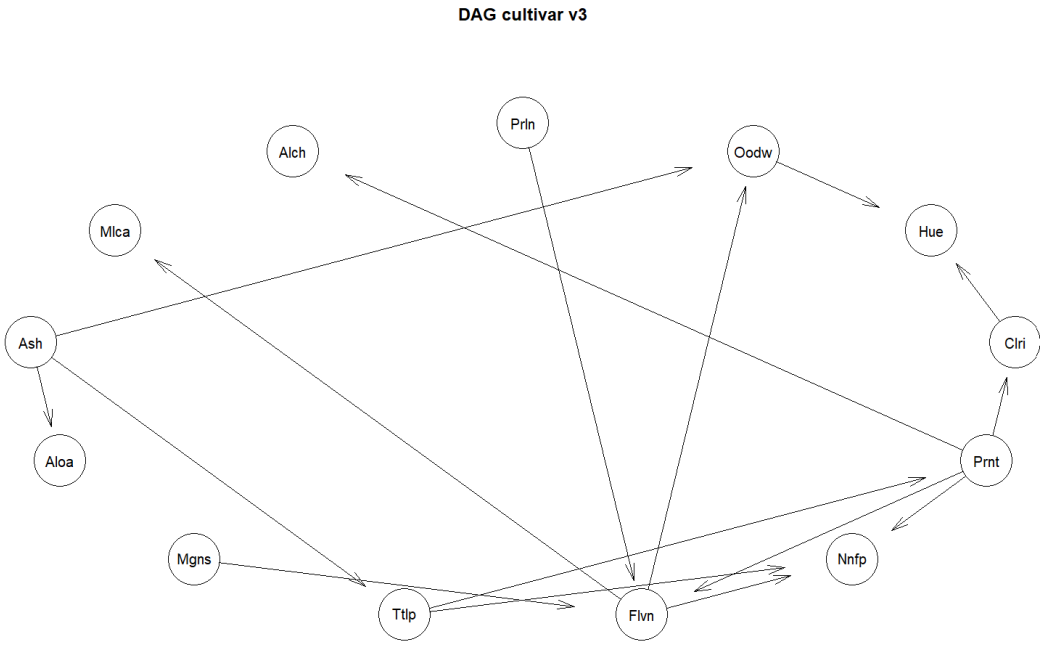


Figura 39: Grafico DAG sulla cultivar v3

```
> fdag$dev
[1] 59.28951
> fdag$df
[1] 62
```

5.3. Variabili Background

Fino ad ora abbiamo generato i DAG utilizzando un approccio standard, costruendo i grafi direttamente a partire dal dataset senza introdurre conoscenze a *priori*. In alcuni casi, i grafi ottenuti mostrano che la variabile target *Clri* influenza altre variabili; tuttavia, nel nostro contesto di studio, questa informazione non è rilevante. Per questo motivo, imponiamo un ordine alle variabili, specificando che *Clri* deve essere considerata esclusivamente come variabile target.

```r
# Selezione delle variabili di background e target
backgnd_vars <- setdiff(names(wine_dataset), "Clri")
target_vars  <- c("Clri")

# Creazione della blacklist
blacklist <- expand.grid(
  from = target_vars,
  to   = backgnd_vars
)
```

I risultati ottenuti sono qui mostrati

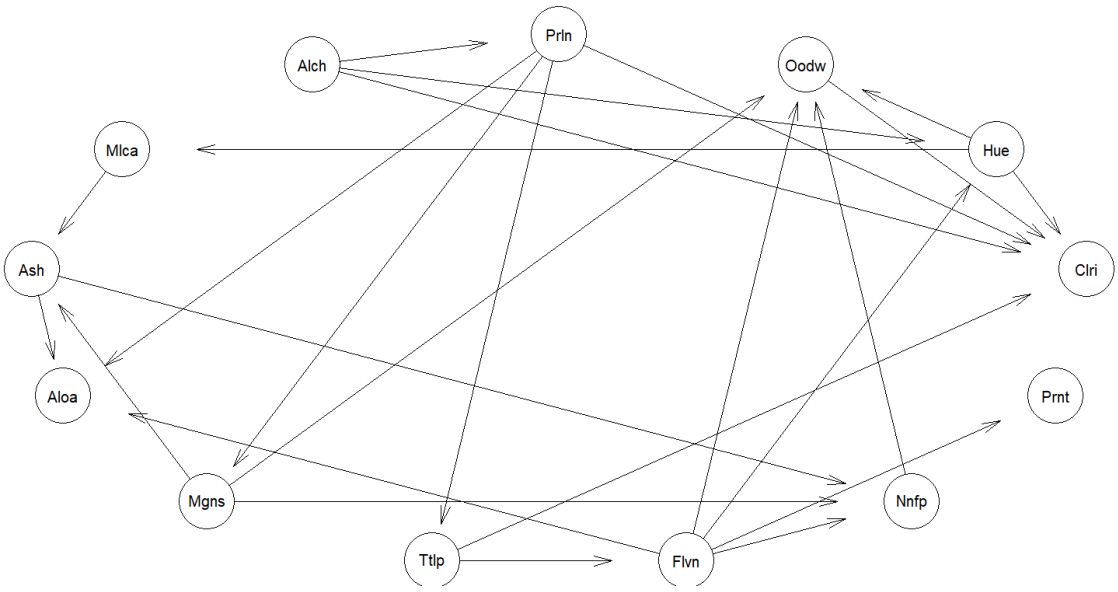


Figura 40: Grafico DAG sull'intera popolazione con target **Clri**

```
> fdag$dev
[1] 76.53876
> fdag$df
[1] 53
```

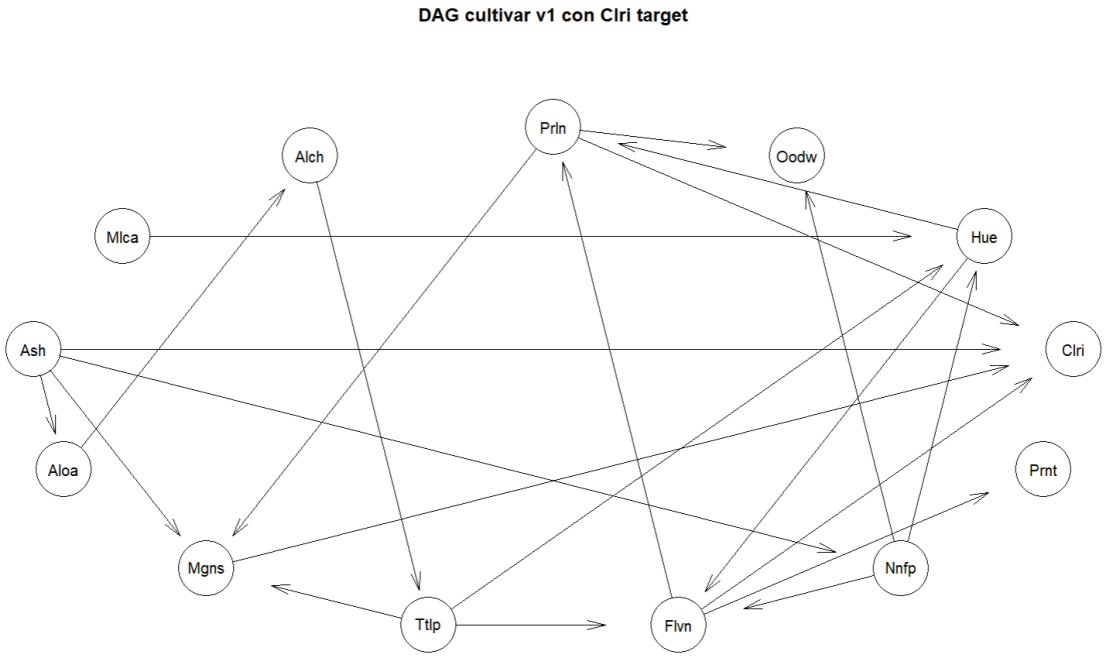


Figura 41: Grafico DAG sulla cultivar v1 con target **Clri**

```
> fdag$dev
[1] 49.18287
> fdag$df
[1] 56
```

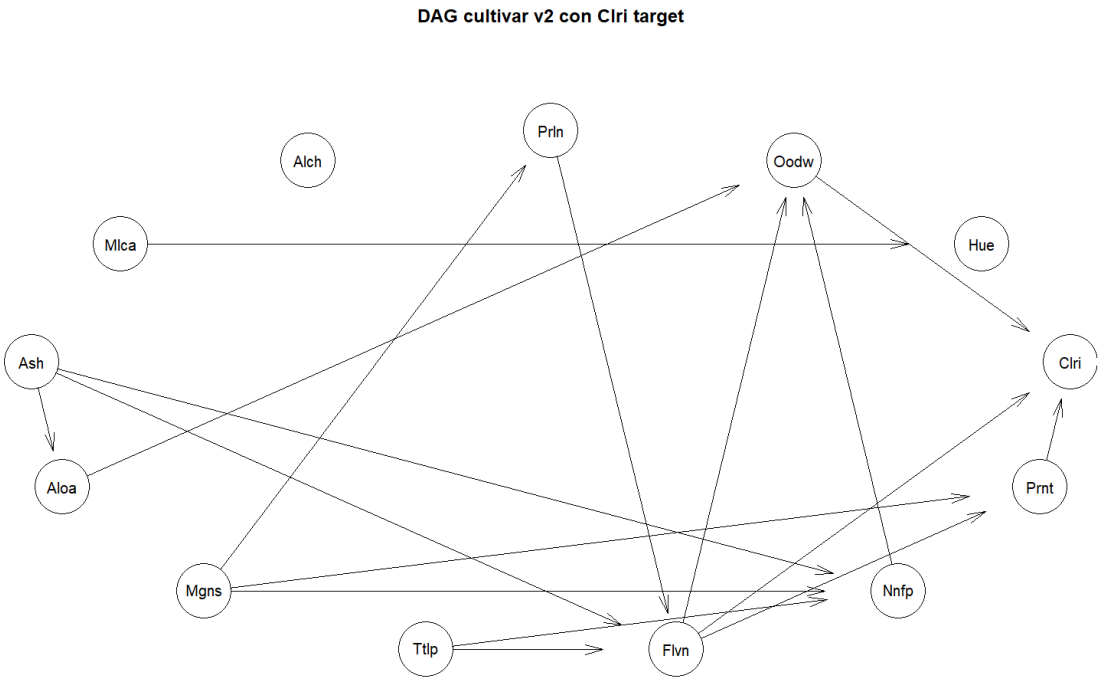


Figura 42: Grafico DAG sulla cultivar v2 con target **Clri**

```
> fdag$dev
[1] 70.88987
> fdag$df
[1] 61
```

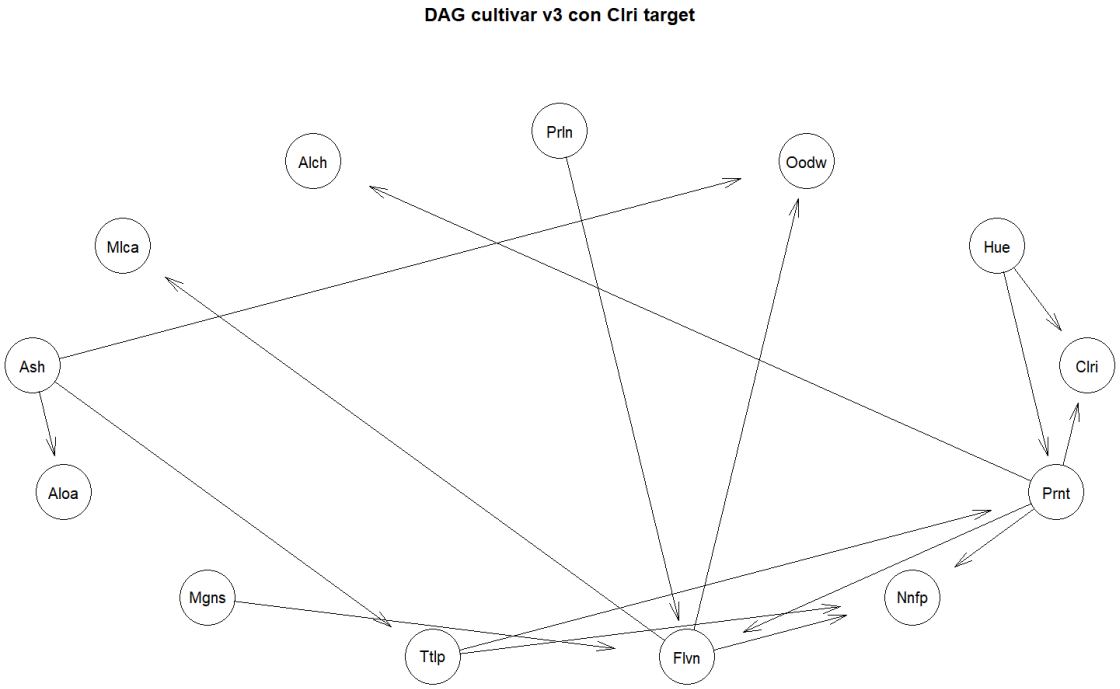


Figura 43: Grafico DAG sulla cultivar v3 con target **Clri**

```
> fdag$dev
[1] 60.83253
> fdag$df
[1] 62
```

5.4. Considerazioni

L'analisi dei DAG ottenuti evidenzia come la struttura delle dipendenze tra le variabili chimiche del vino vari sensibilmente a seconda che si consideri l'intera popolazione o le singole cultivar. Inoltre, l'introduzione di un vincolo strutturale che impone *Clri* come variabile che non può influenzare le altre ha permesso di ottenere DAG più coerenti e leggibili nel contesto dell'analisi. Tale scelta ha inoltre portato a un miglioramento, seppur moderato, dei valori di devianza e dei gradi di libertà, evidenziando l'importanza dell'integrazione di conoscenze *a priori* nel processo di apprendimento della struttura.

Analizzando nello specifico i diversi casi di studio, si osserva che il DAG relativo all'intera popolazione presenta una struttura più densa, con un numero maggiore di interazioni. In questo contesto, la variabile **Clri** risulta influenzata da diverse variabili, in particolare **Oodw**, **Hue**, **Ttlp**, **Prln** e **Alch**.

Nel caso della cultivar *v1*, si osserva un insieme differente di variabili che influenzano la variabile target, ovvero **Prln**, **Flvn**, **Mgns** e **Ash**. Tali differenze rispetto alla struttura osservata sull'intera popolazione suggeriscono una specificità delle relazioni all'interno della cultivar. Un andamento analogo si riscontra anche per le cultivar *v2* e *v3*: nel primo caso, **Clri** risulta influenzata da **Oodw**, **Flvn** e **Prnt**, mentre nel secondo caso le variabili rilevanti sono **Hue** e **Prnt**.

Un aspetto rilevante è che, per ciascuna cultivar, emerge almeno una variabile che non risulta influente nelle altre, ma risulta influente nel contesto della popolazione generale, indicando come il tipo di cultivar rappresenti un fattore discriminante nella determinazione dell'intensità del colore (**Clri**).

5.4.1 Caso di studio con il dataset Wine

I risultati discussi nelle sezioni precedenti suggeriscono che il tipo di cultivar rappresenti un elemento rilevante nella determinazione dell'intensità del colore (**Clri**). Per approfondire questo aspetto, viene ora considerato il dataset **Wine** nella sua configurazione originale, senza le trasformazioni descritte nella sezione 2, includendo esplicitamente la variabile categorica **Cult**, che identifica la cultivar di appartenenza.

L'obiettivo è valutare come la presenza della variabile **Cult** influenzi la struttura del DAG e le relazioni tra le variabili chimiche. A tal fine, vengono integrate conoscenze *a priori*, imponendo **Clri** come variabile target e **Cult** come variabile esclusivamente influenzante, ma non influenzabile.

```
# Con conoscenze a priori
# Conversione di eventuali colonne integer in numeric
wine <- data.frame(lapply(wine, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

# Definizione delle variabili target e esogene
target_vars <- c("Clri")   # variabile target
exogenous_vars <- c("Cult") # variabile esogena

# Tutte le altre variabili
other_vars <- setdiff(names(wine), c(target_vars, exogenous_vars))

# Creazione della blacklist
# Target non puo' influenzare altre variabili
# Variabile esogena non puo' essere influenzata da altre
blacklist <- rbind(
  expand.grid(from = target_vars, to = other_vars),
  expand.grid(from = other_vars, to = exogenous_vars),
  expand.grid(from = target_vars, to = exogenous_vars)
)

# Apprendimento del DAG con Hill-Climbing usando BIC-CG
DAG_wine <- hc(wine, score = "bic-cg", blacklist = blacklist)

# Visualizzazione del DAG
plot(DAG_wine, main = "DAG Wine dataset con target Clri e Cult esogena")
```

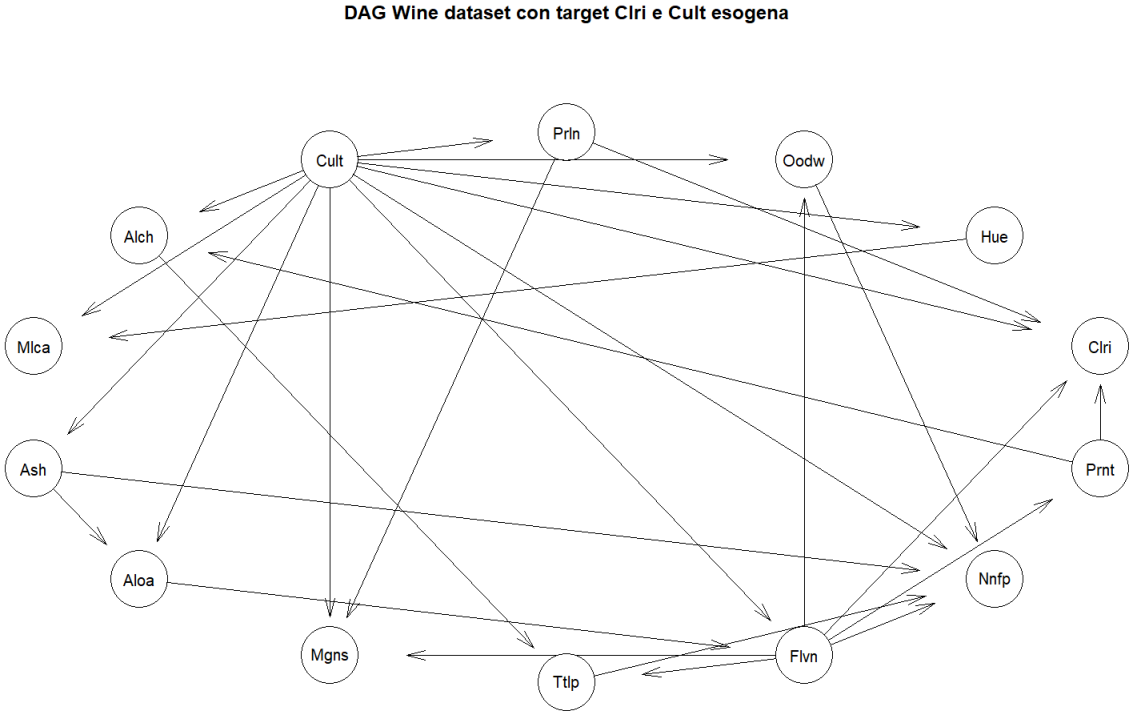


Figura 44: Grafico DAG sul Wine Dataset

Dal grafico (Figura 44) emerge che la variabile **Cult** ha un ruolo predominante nelle relazioni tra le variabili: essa risulta padre di molte variabili ed è un antenato di tutte le variabili del dataset. Osservando nello specifico la nostra variabile target **Clri**, essa risulta influenzata da **Prln**, **Prnt**, **Flvn** e **Cult**. Queste variabili, ad eccezione di **Prln**, non erano emerse nel DAG calcolato sull'intera popolazione (Figura 40). Tuttavia, osservando tutti i grafici, si nota che esse risultano comunque padri o antenati di **Clri**. In altre parole, alcune variabili mostrano la loro influenza solo in analisi più ristrette, mentre nel contesto generale il loro effetto viene mascherato dall'influenza dominante della variabile **Cult**.

6. Modello Predittivo

In questa sezione andremo a stimare un modello di multipla regressione lineare, abbandonando la logica puramente grafica vista fino ad adesso, con lo scopo di analizzare i fattori che influenzano la nostra variabile target **Clri**. La regressione lineare è una tecnica appropriata per modellare una variabile dipendente continua. Per la stima del modello utilizzeremo la funzione *lm()* (Linear Model) che stima i parametri nella forma:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon$$

dove:

- y è la variabile dipendente (target)
- $x_1, x_2, \ldots, x_n$ sono le variabili indipendenti (predittori)
- $\beta_0, \beta_1, \ldots, \beta_n$ sono i coefficienti da stimare
- ε è l'errore $\varepsilon \sim N(0, \sigma^2)$

Successivamente applicheremo la funzione *step()* per effettuare una selezione automatica delle variabili da includere nel modello, utilizzando il criterio **AIC** o **BIC** come metrica di decisione e useremo come procedimento di selezione sia *Forward*, *Backward* e *Both*. Questa scelta è dovuta al fatto che noi ricerchiamo un modello parsimonioso senza ricadere nei casi di *underfitting*, quando un modello è troppo semplice per catturare i pattern presenti nei dati, e *overfitting*, quando un modello è troppo complesso quindi performa molto bene sui dati di training, ma male su dati nuovi. Nel nostro caso per avere una panoramica completa faremo una predizione e mostreremo i risultati per 5 casi di studio:

- **Intera Popolazione**: intera popolazione senza la variabile *Cult*.
- **Cultivar v1**
- **Cultivar v2**
- **Cultivar v3**
- **Wine dataset**: intera popolazione con la variabile *Cult*.

6.1. Intera popolazione

```
#-----------------------------------------
#Intera Popolazione

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = wine_dataset)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = wine_dataset)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope = scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope=scope, direction = "backward",
k=log(nrow(wine_dataset)))
summary(back_model_BIC)

#forward
```

```
forw_model_BIC <- step(null_model, scope=scope, direction = "forward",
k=log(nrow(wine_dataset)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope=scope, direction = "both",
k=log(nrow(wine_dataset)))
summary(both_model_BIC)
```

```
> summary(back_model_AIC)

Call:
lm(formula = Clri ~ Alch + Mlca + Ash + Ttlp + Prnt + Hue + Oodw +
    Prln, data = wine_dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2440 -0.9240 -0.1228  0.6684  4.1653

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.3545348  2.3812970  -2.249   0.0258 *
Alch         1.0651938  0.1773421   6.006 1.13e-08 ***
Mlca        -0.2199606  0.1153942  -1.906   0.0583 .
Ash          0.9475868  0.3972895   2.385   0.0182 *
Ttlp         0.5144795  0.2765596   1.860   0.0646 .
Prnt         0.5255205  0.2316991   2.268   0.0246 *
Hue         -4.1962215  0.6220795  -6.745 2.32e-10 ***
Oodw        -1.5599254  0.2258880  -6.906 9.67e-11 ***
Prln         0.0012194  0.0004926   2.475   0.0143 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.366 on 169 degrees of freedom
Multiple R-squared:  0.6686, Adjusted R-squared:  0.6529
F-statistic: 42.62 on 8 and 169 DF,  p-value: < 2.2e-16

> summary(forw_model_AIC)

Call:
lm(formula = Clri ~ Alch + Hue + Oodw + Ttlp + Prln + Prnt +
    Ash + Mlca, data = wine_dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2440 -0.9240 -0.1228  0.6684  4.1653

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.3545348  2.3812970  -2.249   0.0258 *
Alch         1.0651938  0.1773421   6.006 1.13e-08 ***
Hue         -4.1962215  0.6220795  -6.745 2.32e-10 ***
Oodw        -1.5599254  0.2258880  -6.906 9.67e-11 ***
Ttlp         0.5144795  0.2765596   1.860   0.0646 .
Prln         0.0012194  0.0004926   2.475   0.0143 *
Prnt         0.5255205  0.2316991   2.268   0.0246 *
Ash          0.9475868  0.3972895   2.385   0.0182 *
Mlca        -0.2199606  0.1153942  -1.906   0.0583 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.366 on 169 degrees of freedom
Multiple R-squared:  0.6686, Adjusted R-squared:  0.6529
F-statistic: 42.62 on 8 and 169 DF,  p-value: < 2.2e-16

> summary(both_model_AIC)

Call:
lm(formula = Clri ~ Alch + Hue + Oodw + Ttlp + Prln + Prnt +
    Ash + Mlca, data = wine_dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.2440 -0.9240 -0.1228  0.6684  4.1653

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.3545348  2.3812970  -2.249   0.0258 *
Alch         1.0651938  0.1773421   6.006 1.13e-08 ***
Hue         -4.1962215  0.6220795  -6.745 2.32e-10 ***
Oodw        -1.5599254  0.2258880  -6.906 9.67e-11 ***
Ttlp         0.5144795  0.2765596   1.860   0.0646 .
Prln         0.0012194  0.0004926   2.475   0.0143 *
Prnt         0.5255205  0.2316991   2.268   0.0246 *
Ash          0.9475868  0.3972895   2.385   0.0182 *
Mlca        -0.2199606  0.1153942  -1.906   0.0583 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.366 on 169 degrees of freedom
Multiple R-squared:  0.6686, Adjusted R-squared:  0.6529
F-statistic: 42.62 on 8 and 169 DF,  p-value: < 2.2e-16


> summary(back_model_BIC)
```

```
Call:
lm(formula = Clri ~ Alch + Ash + Prnt + Hue + Oodw + Prln, data = wine_dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4926 -0.8851 -0.1688  0.6490  3.9704

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.1010875  2.4038744  -2.538  0.01204 *
Alch         1.0541395  0.1766718   5.967 1.36e-08 ***
Ash          0.9058577  0.3951388   2.293  0.02309 *
Prnt         0.6949958  0.2185621   3.180  0.00175 **
Hue         -3.6128176  0.5765402  -6.266 2.90e-09 ***
Oodw        -1.3362637  0.2007359  -6.657 3.65e-10 ***
Prln         0.0015455  0.0004874   3.171  0.00180 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.39 on 171 degrees of freedom
Multiple R-squared:  0.6528, Adjusted R-squared:  0.6406
F-statistic: 53.59 on 6 and 171 DF,  p-value: < 2.2e-16


>summary(forw_model_BIC)

Call:
lm(formula = Clri ~ Alch + Hue + Oodw + Ttlp + Prln, data = wine_dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4766 -0.7853 -0.1291  0.7147  3.9403

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.3269993  2.2778573  -1.461  0.14595
Alch         0.9962287  0.1788931   5.569 9.71e-08 ***
Hue         -3.8968758  0.5796411  -6.723 2.52e-10 ***
Oodw        -1.5144813  0.2288457  -6.618 4.45e-10 ***
Ttlp         0.8485098  0.2602911   3.260  0.00134 **
Prln         0.0015698  0.0004943   3.176  0.00177 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.402 on 172 degrees of freedom
Multiple R-squared:  0.6448, Adjusted R-squared:  0.6344
```

```
F-statistic: 62.44 on 5 and 172 DF,  p-value: < 2.2e-16


>summary(both_model_BIC)

Call:
lm(formula = Clri ~ Alch + Hue + Oodw + Ttlp + Prln, data = wine_dataset)

Residuals:
    Min      1Q  Median      3Q     Max
-3.4766 -0.7853 -0.1291  0.7147  3.9403

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.3269993  2.2778573  -1.461  0.14595
Alch         0.9962287  0.1788931   5.569 9.71e-08 ***
Hue         -3.8968758  0.5796411  -6.723 2.52e-10 ***
Oodw        -1.5144813  0.2288457  -6.618 4.45e-10 ***
Ttlp         0.8485098  0.2602911   3.260  0.00134 **
Prln         0.0015698  0.0004943   3.176  0.00177 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.402 on 172 degrees of freedom
Multiple R-squared:  0.6448, Adjusted R-squared:  0.6344
F-statistic: 62.44 on 5 and 172 DF,  p-value: < 2.2e-16
```

L’analisi dell’intera popolazione rivela un trade-off tra complessità e parsimonia: i modelli AIC privilegiano l’accuratezza includendo 8 predittori, mentre BIC preferisce modelli più semplici con 5-6 variabili . L’intensità del colore (Clri) risulta fortemente influenzata da caratteristiche chimiche specifiche: il contenuto alcolico (Alch) mostra un effetto positivo, mentre la tonalità (Hue) e la densità ottica diluita (Oodw) esercitano un’influenza negativa marcata. Questa coerenza tra tutti i modelli suggerisce che queste tre variabili rappresentano i determinanti fondamentali del colore del vino, indipendentemente dalla cultivar.

6.2. Cultivar v1

```
#Cultivar v1

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = v1)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = v1)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC
```

```r
#backward
back_model_AIC <- step(full_model, scope=scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope=scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope=scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
k=log(nrow(v1)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
k=log(nrow(v1)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
k=log(nrow(v1)))
summary(both_model_BIC)
```

```
> summary(back_model_AIC)

Call:
lm(formula = Clri ~ Ash + Mgns + Flvn + Prln, data = v1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36330 -0.38044 -0.06496  0.37576  1.87926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.9057279  1.3651278  -2.129  0.03787 *
Ash         -0.9082703  0.4369037  -2.079  0.04239 *
Mgns         0.0286182  0.0097240   2.943  0.00478 **
Flvn         1.6909562  0.2542961   6.650 1.53e-08 ***
Prln         0.0023107  0.0004543   5.087 4.71e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6898 on 54 degrees of freedom
Multiple R-squared:  0.7112, Adjusted R-squared:  0.6899
F-statistic: 33.25 on 4 and 54 DF,  p-value: 5.494e-14
```

```
> summary(forw_model_AIC)

Call:
lm(formula = Clri ~ Flvn + Prln + Mgns + Ash, data = v1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36330 -0.38044 -0.06496  0.37576  1.87926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.9057279  1.3651278  -2.129  0.03787 *
Flvn         1.6909562  0.2542961   6.650 1.53e-08 ***
Prln         0.0023107  0.0004543   5.087 4.71e-06 ***
Mgns         0.0286182  0.0097240   2.943  0.00478 **
Ash         -0.9082703  0.4369037  -2.079  0.04239 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6898 on 54 degrees of freedom
Multiple R-squared:  0.7112, Adjusted R-squared:  0.6899
F-statistic: 33.25 on 4 and 54 DF,  p-value: 5.494e-14

> summary(both_model_AIC)

Call:
lm(formula = Clri ~ Flvn + Prln + Mgns + Ash, data = v1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36330 -0.38044 -0.06496  0.37576  1.87926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.9057279  1.3651278  -2.129  0.03787 *
Flvn         1.6909562  0.2542961   6.650 1.53e-08 ***
Prln         0.0023107  0.0004543   5.087 4.71e-06 ***
Mgns         0.0286182  0.0097240   2.943  0.00478 **
Ash         -0.9082703  0.4369037  -2.079  0.04239 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6898 on 54 degrees of freedom
Multiple R-squared:  0.7112, Adjusted R-squared:  0.6899
F-statistic: 33.25 on 4 and 54 DF,  p-value: 5.494e-14
```

```
> summary(back_model_BIC)

Call:
lm(formula = Clri ~ Ash + Mgns + Flvn + Prln, data = v1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36330 -0.38044 -0.06496  0.37576  1.87926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.9057279  1.3651278  -2.129  0.03787 *
Ash         -0.9082703  0.4369037  -2.079  0.04239 *
Mgns         0.0286182  0.0097240   2.943  0.00478 **
Flvn         1.6909562  0.2542961   6.650 1.53e-08 ***
Prln         0.0023107  0.0004543   5.087 4.71e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6898 on 54 degrees of freedom
Multiple R-squared:  0.7112, Adjusted R-squared:  0.6899
F-statistic: 33.25 on 4 and 54 DF,  p-value: 5.494e-14


> summary(forw_model_BIC)

Call:
lm(formula = Clri ~ Flvn + Prln + Mgns + Ash, data = v1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36330 -0.38044 -0.06496  0.37576  1.87926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.9057279  1.3651278  -2.129  0.03787 *
Flvn         1.6909562  0.2542961   6.650 1.53e-08 ***
Prln         0.0023107  0.0004543   5.087 4.71e-06 ***
Mgns         0.0286182  0.0097240   2.943  0.00478 **
Ash         -0.9082703  0.4369037  -2.079  0.04239 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6898 on 54 degrees of freedom
Multiple R-squared:  0.7112, Adjusted R-squared:  0.6899
F-statistic: 33.25 on 4 and 54 DF,  p-value: 5.494e-14
```

```
> summary(both_model_BIC)

Call:
lm(formula = Clri ~ Flvn + Prln + Mgns + Ash, data = v1)

Residuals:
     Min       1Q   Median       3Q      Max
-1.36330 -0.38044 -0.06496  0.37576  1.87926

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.9057279  1.3651278  -2.129  0.03787 *
Flvn         1.6909562  0.2542961   6.650 1.53e-08 ***
Prln         0.0023107  0.0004543   5.087 4.71e-06 ***
Mgns         0.0286182  0.0097240   2.943  0.00478 **
Ash         -0.9082703  0.4369037  -2.079  0.04239 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6898 on 54 degrees of freedom
Multiple R-squared:  0.7112, Adjusted R-squared:  0.6899
F-statistic: 33.25 on 4 and 54 DF,  p-value: 5.494e-14
```

La cultivar v1 mostra un comportamento predittivo notevolmente stabile: tutti i metodi di selezione convergono sullo stesso modello a 4 variabili, suggerendo una struttura chiara e ben definita. I flavonoidi (Flvn) emergono come il fattore dominante per il colore in questa cultivar, seguiti dalle proline (Prln). Il magnesio (Mgns) e le ceneri (Ash) giocano ruoli opposti: il primo aumenta l'intensità del colore mentre il secondo la diminuisce. L'elevata capacità predittiva (69%) e la consistenza tra AIC e BIC indicano che il colore della v1 è governato da un insieme relativamente ristretto di componenti chimici ben identificabili.

6.3. Cultivar v2

```r
#-----------------------------------------
#Cultivar v2

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = v2)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = v2)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
```

```r
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope = scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
k=log(nrow(v2)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
k=log(nrow(v2)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
k=log(nrow(v2)))
summary(both_model_BIC)
```

```
> summary(back_model_AIC)

Call:
lm(formula = Clri ~ Alch + Aloa + Ttlp + Flvn + Prnt + Oodw +
    Prln, data = v2)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1379 -0.3792 -0.0628  0.2971  2.0420

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.0229321  2.0908935   0.489  0.62638
Alch         0.2824332  0.1547092   1.826  0.07265 .
Aloa        -0.0602569  0.0271873  -2.216  0.03029 *
Ttlp        -0.6712585  0.2440423  -2.751  0.00776 **
Flvn         1.5077122  0.2169506   6.950 2.45e-09 ***
Prnt        -0.5326711  0.1650955  -3.226  0.00199 **
Oodw        -0.6074163  0.2132698  -2.848  0.00593 **
Prln         0.0014523  0.0005463   2.659  0.00994 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.675 on 63 degrees of freedom
```

```
Multiple R-squared:  0.5207, Adjusted R-squared:  0.4674
F-statistic: 9.776 on 7 and 63 DF,  p-value: 3.511e-08

> summary(forw_model_AIC)

Call:
lm(formula = Clri ~ Flvn + Oodw + Prnt + Prln + Ttlp + Aloa +
    Alch, data = v2)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1379 -0.3792 -0.0628  0.2971  2.0420

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.0229321  2.0908935   0.489  0.62638
Flvn         1.5077122  0.2169506   6.950 2.45e-09 ***
Oodw        -0.6074163  0.2132698  -2.848  0.00593 **
Prnt        -0.5326711  0.1650955  -3.226  0.00199 **
Prln         0.0014523  0.0005463   2.659  0.00994 **
Ttlp        -0.6712585  0.2440423  -2.751  0.00776 **
Aloa        -0.0602569  0.0271873  -2.216  0.03029 *
Alch         0.2824332  0.1547092   1.826  0.07265 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.675 on 63 degrees of freedom
Multiple R-squared:  0.5207, Adjusted R-squared:  0.4674
F-statistic: 9.776 on 7 and 63 DF,  p-value: 3.511e-08

> summary(both_model_AIC)

Call:
lm(formula = Clri ~ Flvn + Oodw + Prnt + Prln + Ttlp + Aloa +
    Alch, data = v2)

Residuals:
    Min      1Q  Median      3Q     Max
-1.1379 -0.3792 -0.0628  0.2971  2.0420

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.0229321  2.0908935   0.489  0.62638
Flvn         1.5077122  0.2169506   6.950 2.45e-09 ***
Oodw        -0.6074163  0.2132698  -2.848  0.00593 **
Prnt        -0.5326711  0.1650955  -3.226  0.00199 **
Prln         0.0014523  0.0005463   2.659  0.00994 **
```

```
Ttlp        -0.6712585  0.2440423  -2.751  0.00776 **
Aloa        -0.0602569  0.0271873  -2.216  0.03029 *
Alch         0.2824332  0.1547092   1.826  0.07265 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.675 on 63 degrees of freedom
Multiple R-squared:  0.5207, Adjusted R-squared:  0.4674
F-statistic: 9.776 on 7 and 63 DF,  p-value: 3.511e-08

> summary(back_model_BIC)

Call:
lm(formula = Clri ~ Aloa + Ttlp + Flvn + Prnt + Oodw + Prln,
    data = v2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.13426 -0.41822 -0.08136  0.24817  2.07081

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.6234883  0.7067420   6.542 1.18e-08 ***
Aloa        -0.0625045  0.0276500  -2.261 0.027195 *
Ttlp        -0.6911723  0.2482017  -2.785 0.007038 **
Flvn         1.5556748  0.2192435   7.096 1.26e-09 ***
Prnt        -0.5930403  0.1646712  -3.601 0.000619 ***
Oodw        -0.6391397  0.2163998  -2.954 0.004390 **
Prln         0.0015386  0.0005541   2.777 0.007190 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6872 on 64 degrees of freedom
Multiple R-squared:  0.4953, Adjusted R-squared:  0.448
F-statistic: 10.47 on 6 and 64 DF,  p-value: 4.599e-08

> summary(forw_model_BIC)

Call:
lm(formula = Clri ~ Flvn + Oodw + Prnt + Prln + Ttlp + Aloa,
    data = v2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.13426 -0.41822 -0.08136  0.24817  2.07081

Coefficients:
```

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.6234883  0.7067420   6.542 1.18e-08 ***
Flvn         1.5556748  0.2192435   7.096 1.26e-09 ***
Oodw        -0.6391397  0.2163998  -2.954 0.004390 **
Prnt        -0.5930403  0.1646712  -3.601 0.000619 ***
Prln         0.0015386  0.0005541   2.777 0.007190 **
Ttlp        -0.6911723  0.2482017  -2.785 0.007038 **
Aloa        -0.0625045  0.0276500  -2.261 0.027195 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6872 on 64 degrees of freedom
Multiple R-squared:  0.4953, Adjusted R-squared:  0.448
F-statistic: 10.47 on 6 and 64 DF,  p-value: 4.599e-08

> summary(both_model_BIC)

Call:
lm(formula = Clri ~ Flvn + Oodw + Prnt + Prln + Ttlp + Aloa,
    data = v2)

Residuals:
     Min       1Q   Median       3Q      Max
-1.13426 -0.41822 -0.08136  0.24817  2.07081

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.6234883  0.7067420   6.542 1.18e-08 ***
Flvn         1.5556748  0.2192435   7.096 1.26e-09 ***
Oodw        -0.6391397  0.2163998  -2.954 0.004390 **
Prnt        -0.5930403  0.1646712  -3.601 0.000619 ***
Prln         0.0015386  0.0005541   2.777 0.007190 **
Ttlp        -0.6911723  0.2482017  -2.785 0.007038 **
Aloa        -0.0625045  0.0276500  -2.261 0.027195 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6872 on 64 degrees of freedom
Multiple R-squared:  0.4953, Adjusted R-squared:  0.448
F-statistic: 10.47 on 6 and 64 DF,  p-value: 4.599e-08
```

La cultivar v2 presenta la dinamica più complessa tra le tre varietà, richiedendo 6-7 predittori, ma raggiungendo comunque una capacità esplicativa moderata (47%). I flavonoidi (Flvn) rimangono il predittore dominante, ma emergono interessanti effetti antagonisti: mentre alcaloidi (Aloa) e polifenoli totali (Ttlp) riducono l'intensità del colore, le proline (Prln) la aumentano. La densità ottica (Oodw) e i proantocianidini (Prnt) mostrano effetti negativi significativi.

6.4. Cultivar v3

```
#-----------------------------------------
#Cultivar v3

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = v3)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = v3)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope=scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
k=log(nrow(v3)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
k=log(nrow(v3)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
k=log(nrow(v3)))
summary(both_model_BIC)
```

```
> summary(back_model_AIC)

Call:
lm(formula = Clri ~ Alch + Prnt + Hue, data = v3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3992 -0.8060 -0.1611  0.6312  3.8603
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1408     5.8586   0.024 0.980939
Alch          0.7005     0.4615   1.518 0.136188
Prnt          2.6599     0.6597   4.032 0.000217 ***
Hue          -7.3633     2.1847  -3.370 0.001571 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.537 on 44 degrees of freedom
Multiple R-squared:  0.586, Adjusted R-squared:  0.5577
F-statistic: 20.76 on 3 and 44 DF,  p-value: 1.572e-08

> summary(forw_model_AIC)

Call:
lm(formula = Clri ~ Prnt + Hue + Alch, data = v3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3992 -0.8060 -0.1611  0.6312  3.8603

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.1408     5.8586   0.024 0.980939
Prnt          2.6599     0.6597   4.032 0.000217 ***
Hue          -7.3633     2.1847  -3.370 0.001571 **
Alch          0.7005     0.4615   1.518 0.136188
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.537 on 44 degrees of freedom
Multiple R-squared:  0.586, Adjusted R-squared:  0.5577
F-statistic: 20.76 on 3 and 44 DF,  p-value: 1.572e-08

> summary(both_model_AIC)

Call:
lm(formula = Clri ~ Prnt + Hue + Alch, data = v3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.3992 -0.8060 -0.1611  0.6312  3.8603

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
```

```
(Intercept)   0.1408     5.8586   0.024 0.980939
Prnt          2.6599     0.6597   4.032 0.000217 ***
Hue          -7.3633     2.1847  -3.370 0.001571 **
Alch          0.7005     0.4615   1.518 0.136188
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.537 on 44 degrees of freedom
Multiple R-squared:  0.586, Adjusted R-squared:  0.5577
F-statistic: 20.76 on 3 and 44 DF,  p-value: 1.572e-08

> summary(back_model_BIC)

Call:
lm(formula = Clri ~ Prnt + Hue, data = v3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4127 -0.8870  0.0367  0.6434  3.6816

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.5571     1.9192   4.459 5.44e-05 ***
Prnt          3.0602     0.6135   4.988 9.56e-06 ***
Hue          -6.8710     2.1915  -3.135  0.00302 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.559 on 45 degrees of freedom
Multiple R-squared:  0.5643, Adjusted R-squared:  0.5449
F-statistic: 29.14 on 2 and 45 DF,  p-value: 7.626e-09

> summary(forw_model_BIC)

Call:
lm(formula = Clri ~ Prnt + Hue, data = v3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4127 -0.8870  0.0367  0.6434  3.6816

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.5571     1.9192   4.459 5.44e-05 ***
Prnt          3.0602     0.6135   4.988 9.56e-06 ***
Hue          -6.8710     2.1915  -3.135  0.00302 **
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.559 on 45 degrees of freedom
Multiple R-squared:  0.5643, Adjusted R-squared:  0.5449
F-statistic: 29.14 on 2 and 45 DF,  p-value: 7.626e-09

> summary(both_model_BIC)

Call:
lm(formula = Clri ~ Prnt + Hue, data = v3)

Residuals:
    Min      1Q  Median      3Q     Max
-4.4127 -0.8870  0.0367  0.6434  3.6816

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   8.5571     1.9192   4.459 5.44e-05 ***
Prnt          3.0602     0.6135   4.988 9.56e-06 ***
Hue          -6.8710     2.1915  -3.135  0.00302 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.559 on 45 degrees of freedom
Multiple R-squared:  0.5643, Adjusted R-squared:  0.5449
F-statistic: 29.14 on 2 and 45 DF,  p-value: 7.626e-09
```

La cultivar v3 si distingue per una struttura predittiva parsimoniosa: solo 2-3 variabili sono sufficienti. I proantocianidini (Prnt) emergono come il fattore chiave con un effetto positivo molto marcato, mentre la tonalità (Hue) esercita un'influenza negativa forte. Il ruolo marginale dell'alcol (Alch) nei modelli AIC e la sua esclusione nei modelli BIC suggeriscono che, a differenza delle altre cultivar, il colore della v3 dipende principalmente da componenti fenolici piuttosto che dalla composizione alcolica. Questa semplicità strutturale potrebbe riflettere caratteristiche genetiche distintive di questa varietà che rendono il suo profilo cromatico più uniforme e prevedibile.

6.5. Wine dataset

```r
#-----------------------------------------
#Wine dataset

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = wine)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = wine)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))
```

```r
#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope = scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
k=log(nrow(wine)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
k=log(nrow(wine)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
k=log(nrow(wine)))
summary(both_model_BIC)
```

```
>summary(back_model_AIC)

Call:
lm(formula = Clri ~ Cult + Alch + Mlca + Flvn + Nnfp + Prnt +
    Hue + Oodw + Prln, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8039 -0.6942 -0.0731  0.5471  3.4604

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.4737659  2.7193881  -2.013 0.045735 *
Cultv2       0.1420256  0.4807952   0.295 0.768057
Cultv3       3.7687292  0.6256005   6.024 1.05e-08 ***
Alch         0.7722731  0.1807530   4.273 3.24e-05 ***
Mlca        -0.3420391  0.1043041  -3.279 0.001267 **
Flvn         0.9564979  0.2108019   4.537 1.08e-05 ***
Nnfp         1.5552341  0.8898717   1.748 0.082353 .
Prnt         0.4438324  0.2149799   2.065 0.040514 *
```

```
Hue         -2.4542630  0.6184535  -3.968 0.000107 ***
Oodw        -0.6644000  0.2506342  -2.651 0.008801 **
Prln         0.0014637  0.0005643   2.594 0.010336 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.206 on 167 degrees of freedom
Multiple R-squared:  0.7446, Adjusted R-squared:  0.7293
F-statistic: 48.69 on 10 and 167 DF,  p-value: < 2.2e-16

> summary(forw_model_AIC)

Call:
lm(formula = Clri ~ Cult + Flvn + Alch + Oodw + Prln + Hue +
    Mlca + Prnt + Nnfp, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8039 -0.6942 -0.0731  0.5471  3.4604

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.4737659  2.7193881  -2.013 0.045735 *
Cultv2       0.1420256  0.4807952   0.295 0.768057
Cultv3       3.7687292  0.6256005   6.024 1.05e-08 ***
Flvn         0.9564979  0.2108019   4.537 1.08e-05 ***
Alch         0.7722731  0.1807530   4.273 3.24e-05 ***
Oodw        -0.6644000  0.2506342  -2.651 0.008801 **
Prln         0.0014637  0.0005643   2.594 0.010336 *
Hue         -2.4542630  0.6184535  -3.968 0.000107 ***
Mlca        -0.3420391  0.1043041  -3.279 0.001267 **
Prnt         0.4438324  0.2149799   2.065 0.040514 *
Nnfp         1.5552341  0.8898717   1.748 0.082353 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.206 on 167 degrees of freedom
Multiple R-squared:  0.7446, Adjusted R-squared:  0.7293
F-statistic: 48.69 on 10 and 167 DF,  p-value: < 2.2e-16

> summary(both_model_AIC)

Call:
lm(formula = Clri ~ Cult + Flvn + Alch + Oodw + Prln + Hue +
    Mlca + Prnt + Nnfp, data = wine)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-3.8039 -0.6942 -0.0731  0.5471  3.4604

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.4737659  2.7193881  -2.013 0.045735 *
Cultv2       0.1420256  0.4807952   0.295 0.768057
Cultv3       3.7687292  0.6256005   6.024 1.05e-08 ***
Flvn         0.9564979  0.2108019   4.537 1.08e-05 ***
Alch         0.7722731  0.1807530   4.273 3.24e-05 ***
Oodw        -0.6644000  0.2506342  -2.651 0.008801 **
Prln         0.0014637  0.0005643   2.594 0.010336 *
Hue         -2.4542630  0.6184535  -3.968 0.000107 ***
Mlca        -0.3420391  0.1043041  -3.279 0.001267 **
Prnt         0.4438324  0.2149799   2.065 0.040514 *
Nnfp         1.5552341  0.8898717   1.748 0.082353 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.206 on 167 degrees of freedom
Multiple R-squared:  0.7446, Adjusted R-squared:  0.7293
F-statistic: 48.69 on 10 and 167 DF,  p-value: < 2.2e-16

> summary(back_model_BIC)

Call:
lm(formula = Clri ~ Cult + Alch + Mlca + Flvn + Hue + Oodw +
    Prln, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2700 -0.5994 -0.1066  0.5543  3.4764

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.7598393  2.7241632  -1.747 0.082408 .
Cultv2       0.3386517  0.4791335   0.707 0.480664
Cultv3       3.9516956  0.6272708   6.300 2.49e-09 ***
Alch         0.7747883  0.1834958   4.222 3.94e-05 ***
Mlca        -0.3056363  0.1047083  -2.919 0.003991 **
Flvn         1.0843258  0.1887959   5.743 4.22e-08 ***
Hue         -2.4306963  0.6227878  -3.903 0.000137 ***
Oodw        -0.6980701  0.2495336  -2.798 0.005748 **
Prln         0.0016182  0.0005649   2.865 0.004705 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

```
Residual standard error: 1.224 on 169 degrees of freedom
Multiple R-squared:  0.7336, Adjusted R-squared:  0.721
F-statistic: 58.18 on 8 and 169 DF,  p-value: < 2.2e-16

> summary(forw_model_BIC)

Call:
lm(formula = Clri ~ Cult + Flvn + Alch + Oodw + Prln + Hue +
    Mlca, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2700 -0.5994 -0.1066  0.5543  3.4764

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.7598393  2.7241632  -1.747 0.082408 .
Cultv2       0.3386517  0.4791335   0.707 0.480664
Cultv3       3.9516956  0.6272708   6.300 2.49e-09 ***
Flvn         1.0843258  0.1887959   5.743 4.22e-08 ***
Alch         0.7747883  0.1834958   4.222 3.94e-05 ***
Oodw        -0.6980701  0.2495336  -2.798 0.005748 **
Prln         0.0016182  0.0005649   2.865 0.004705 **
Hue         -2.4306963  0.6227878  -3.903 0.000137 ***
Mlca        -0.3056363  0.1047083  -2.919 0.003991 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.224 on 169 degrees of freedom
Multiple R-squared:  0.7336, Adjusted R-squared:  0.721
F-statistic: 58.18 on 8 and 169 DF,  p-value: < 2.2e-16

> summary(both_model_BIC)

Call:
lm(formula = Clri ~ Cult + Flvn + Alch + Oodw + Prln + Hue +
    Mlca, data = wine)

Residuals:
    Min      1Q  Median      3Q     Max
-4.2700 -0.5994 -0.1066  0.5543  3.4764

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.7598393  2.7241632  -1.747 0.082408 .
Cultv2       0.3386517  0.4791335   0.707 0.480664
Cultv3       3.9516956  0.6272708   6.300 2.49e-09 ***
```

```
Flvn         1.0843258  0.1887959   5.743 4.22e-08 ***
Alch         0.7747883  0.1834958   4.222 3.94e-05 ***
Oodw        -0.6980701  0.2495336  -2.798 0.005748 **
Prln         0.0016182  0.0005649   2.865 0.004705 **
Hue         -2.4306963  0.6227878  -3.903 0.000137 ***
Mlca        -0.3056363  0.1047083  -2.919 0.003991 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.224 on 169 degrees of freedom
Multiple R-squared:  0.7336, Adjusted R-squared:  0.721
F-statistic: 58.18 on 8 and 169 DF,  p-value: < 2.2e-16
```

L'inclusione della variabile cultivar (Cult) rivela differenze fondamentali tra le varietà: la v3 mostra un'intensità di colore significativamente superiore rispetto alle altre (coefficiente +3.77 per AIC), mentre v1 e v2 risultano più simili tra loro. Il modello completo raggiunge la migliore performance predittiva (73%), confermando che l'appartenenza varietale è un determinante cruciale del colore. Tuttavia, l'analisi congiunta evidenzia anche fattori chimici trasversali: i flavonoidi (Flvn) e l'alcol (Alch) incrementano universalmente l'intensità cromatica, mentre la tonalità (Hue), l'acido malico (Mlca) e la densità ottica (Oodw) la riducono. Interessante notare che alcune variabili significative nelle analisi separate (come Mgns per v1) perdono rilevanza nel modello globale, suggerendo che i loro effetti sono specifici di cultivar e mediati da interazioni più complesse. Questo modello integrato offre la visione più completa, bilanciando le specificità varietali con i pattern chimici comuni.

7. Conclusioni

In questo progetto abbiamo analizzato come interagiscono tra loro le variabili chimiche in particolare quali influenzano l'intensità del colore del vino **Clri**. L'analisi è partita dal dataset *WINE* il quale è stato modificato e diviso in cinque casi di studio per avere una panoramica sia generale che specifica del comportamento della nostra variabile target.
Come prima fase, abbiamo costruito dei grafi indiretti che si basano sulle dipendenze condizionali tra le variabili, che ci hanno permesso di identificare in modo visivo e rapido le relazioni più forti. Questi grafi, seppur non orientati né causali, sono utili per una prima esplorazione, fornendo uno spunto per le successive analisi.
Successivamente, abbiamo stimato dei modelli grafici Gaussiani orientati ottenendo una rappresentazione grafica direzionata delle dipendenze probabilistiche tra le variabili. Attraverso l'osservazione dei diversi casi di studio, si evidenzia come le interazioni tra le sostanze chimiche varino in funzione del contesto analizzato.

- Nel caso dell'**intera popolazione**, senza considerare l'influenza del tipo di cultivar, si osserva che la variabile *Clri* risulta influenzata dalla prolina (*Prln*), dalla concentrazione di composti fenolici (*Oodw*), dal contenuto alcolico (*Alch*), dalla tonalità del colore (*Hue*) e dai fenoli totali (*Ttlp*). È interessante notare come i flavonoidi (*Flvn*), che dal punto di vista chimico sono noti per il loro ruolo nella determinazione del colore, non emergano come influenze dirette; tuttavia, un'analisi più approfondita mostra come il loro effetto sia mediato da altre variabili, in particolare *Oodw* e *Hue*.

- Nel contesto delle **singole cultivar** si osserva come le variabili che influenzano l'intensità del colore del vino cambino in funzione del tipo di cultivar e spesso non coincidano con il trend osservato sull'intera popolazione. Un aspetto particolarmente rilevante è che ciascuna cultivar presenta almeno una variabile influenzante che non emerge nelle altre, ma che risulta significativa nel modello generale (*v1* con *Prln*, *v2* con *Oodw* e *v3* con *Hue*). Ciò evidenzia come il tipo di cultivar rappresenti un fattore discriminante nella determinazione della variabile *Clri*.

- Nello studio dell'**intero dataset**, che include anche la variabile fattoriale *Cult*, si osserva come il tipo di cultivar svolga un ruolo centrale nelle interazioni tra le variabili chimiche. In particolare, *Cult* risulta essere un nodo antenato dell'intera rete e influenza direttamente, come atteso, la variabile target *Clri*. Inoltre, il colore del vino risulta influenzato anche da variabili quali i flavonoidi (*Flvn*), le proantocianidine (*Prnt*) e la prolina (*Prln*), tutte sostanze chimicamente rilevanti nella determinazione del colore.

Come ultimo passo, è stata effettuata un'analisi di regressione lineare multipla con *Clri* come variabile risposta, per individuare i principali predittori. Abbiamo applicato la selezione automatica delle variabili tramite i criteri AIC e BIC, con approcci forward, backward e both. Dai risultati possiamo trarre:

- In alcuni contesti i modelli predetti rispettano i grafici DAG ottenuti (*Intera popolazione*, *v1*, *v3*), mentre in altri si riscontra la necessità di usare variabili diverse o più predittori.
- La predizione della cultivar *v2* risulta differente da ciò che è stato determinato dal DAG. Infatti, anche utilizzando il criterio parsimonioso BIC, il modello richiede 6 variabili predittive rispetto alle sole 3 identificate dal DAG, suggerendo una maggiore complessità nelle relazioni lineari rispetto alle dipendenze condizionali.
- Nel contesto dell'intero dataset senza modifiche (*WINE*), notiamo che la variabile Cult sia importante per la predizione e che rispetto agli altri modelli aggiunge più espressività. Tuttavia, notiamo che rispetto a ciò che abbiamo visto dal DAG il modello predittivo necessita di più variabili.

Nel complesso, il progetto mostra come l'integrazione di modelli grafici e regressione lineare rappresenti un approccio efficace per studiare sistemi chimici complessi, consentendo di cogliere sia le relazioni globali sia le specificità locali. I risultati ottenuti sottolineano il ruolo centrale della cultivar nella determinazione del colore del vino.

Riferimenti bibliografici

[1] S. Aeberhard and M. Forina. Wine. UCI Machine Learning Repository, 1992. DOI: https://doi.org/10.24432/C5PC7J.

[2] J. H. Friedman and B. E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3), Sept. 2008.

8. Appendice

Di seguito viene riportato il codice completo in R, utilizzato per l'analisi in questione. Il tutto può essere anche visualizzato all'interno della repository su Github.

```
# Inizializzazione
rm(list=ls(all=TRUE))

#Carico dataset
require(gRbase)
data(wine)

#Visualizzazione struttura
str(wine)

#visualizzazioni statistiche del dataset
summary(wine)

#Togliamo dal dataset la variabile categorica Cult
wine_dataset<-subset(wine, select = -Cult)
str(wine_dataset)

# Creiamo tre dataset dei vini raggruppati per tipo di cultura (v1, v2, v3)
list_wine<-split(wine, wine$Cult)

# visualizziamo tutte e tre le liste di vini separati per Cult
str(list_wine)

# cos  ne visulizziamo solo una
str(list_wine$v1)
str(list_wine$v2)
str(list_wine$v3)
#summary(list_wine)

summary(list_wine$v1)
summary(list_wine$v2)
summary(list_wine$v3)

#mi fornisce la statistica colore dell'intera popolazione
summary(wine_dataset$Clri)

#mi fornisce le statistiche del colore per una singola lista di vini
#v1
summary(list_wine$v1$Clri)
#v2
summary(list_wine$v2$Clri)
#v3
summary(list_wine$v3$Clri)

#Usando R base per la realizzazione grafico

# Calcolo densit  di tutti i gruppi
dens_pop <- density(wine_dataset$Clri)
dens_v1  <- density(list_wine$v1$Clri)
```

```r
dens_v2  <- density(list_wine$v2$Clri)
dens_v3  <- density(list_wine$v3$Clri)

# Determino il massimo valore di y per includere tutte le curve
ymax <- max(dens_pop$y, dens_v1$y, dens_v2$y, dens_v3$y)

plot(dens_pop, main="Density plot di Clri - Popolazione e Cultivar",
xlab="Clri", lwd=4, col=rgb(0,0,0,0.7), ylim=c(0,ymax*1.1)) # aggiungo un 10%
                    #per margine

# Aggiunta delle altre densit
lines(dens_v1, lwd=2, col=rgb(0,1,0,0.5))
lines(dens_v2, lwd=2, col=rgb(1,0,0,0.5))
lines(dens_v3, lwd=2, col=rgb(0,0,1,0.5))

# Legenda
legend("topright", legend=c("Popolazione","v1","v2","v3"),
       col=c("black","green","red","blue"), lwd=2)

# install.packages("ggplot2")
#library(ggplot2)

# Boxplot dell'intera popolazione
boxplot(wine_dataset$Clri,
        main = "Boxplot di Clri - Intera popolazione",
        ylab = "Clri",
        col = "lightblue")

# Boxplot suddiviso per cultivar
boxplot(Clri ~ Cult,
        data = wine,
        main = "Boxplot di Clri per Cultivar",
        xlab = "Cultivar",
        ylab = "Clri",
        col = c("lightgreen", "lightpink", "lightblue"))

# Dividere i dataset rimuovendo la variabile categorica Cult
v1 <- subset(list_wine$v1, select = -Cult)
v2 <- subset(list_wine$v2, select = -Cult)
v3 <- subset(list_wine$v3, select = -Cult)

#Matrici Scatter Plot
#Intera popolazione
plot(wine_dataset, main="Scatterplot Matrix dell'intera popolazione")

#Diviso per tipo di cultivar
plot(v1, main="Scatterplot Matrix di v1")
plot(v2, main="Scatterplot Matrix di v2")
plot(v3, main="Scatterplot Matrix di v3")



# Matrice di correlazione
```

```r
#intera popolazione
corr <- cor(wine_dataset)

# Matrici di correlazione per ciascuna cultivar
corr1 <- cor(v1)
corr2 <- cor(v2)
corr3 <- cor(v3)

#heatmap
# install.packages("pheatmap")
library(pheatmap)
# Intera popolazione
pheatmap(corr,
         main = "Heatmap delle correlazioni - Popolazione",
         fontsize = 10)

# Per ciascuna cultivar
pheatmap(corr1, main = "Cultivar v1")
pheatmap(corr2, main = "Cultivar v2")
pheatmap(corr3, main = "Cultivar v3")

#-----------------------------------
#install.packages("gRbase")
#install.packages("gRain")
#install.packages("gRim")
# pacchetti per i modelli grafi indiretti
library(gRbase)
library(gRain)
library(gRim)

#Matrici di concentrazione
#intera popolazione
CovPop <- cov.wt(wine_dataset, method = "ML")$cov
concPop <-solve(CovPop)
round(100*concPop)

#diviso per cultivar
#V1
CovV1 <- cov.wt(v1, method = "ML")$cov
concV1 <- solve(CovV1)
round(100*concV1)

#V2
CovV2 <- cov.wt(v2, method = "ML")$cov
concV2 <- solve(CovV2)
round(100*concV2)

#V3
CovV3 <- cov.wt(v3, method = "ML")$cov
concV3 <- solve(CovV3)
round(100*concV3)
```

```r
#Matrici delle correlazioni parziali

#intera popolazione
popCP <- cov2pcor(concPop)
round(100*popCP)

#diviso per cultivar

#V1
v1CP <- cov2pcor(concV1)
round(100*v1CP)

#V2
v2CP <- cov2pcor(concV2)
round(100*v2CP)

#V3
v3CP <- cov2pcor(concV3)
round(100*v3CP)

#-----------------------------------------------------------------
#Grafi indiretti

library(gRbase)
library(gRain)
library(gRim)

#Intera popolazione
#creazione modello saturo
pop_mod_sat <- cmod(~ .^., data=wine_dataset)

#creazione modello di indipendenza
pop_mod_ind <- cmod(~.^1, data=wine_dataset)

#grafico usando penalizzazione AIC Forward
AIC_pop_F <- stepwise(pop_mod_ind, direction="forward")
plot(AIC_pop_F, "neato")
title(main="UG Forward AIC intera popolazione")

#grafico usando AIC Backward
AIC_pop_B <- stepwise(pop_mod_sat)
plot(AIC_pop_B, "neato")
title(main = "UG Backward AIC intera popolazione")

#grafico usando BIC forward
BIC_pop_F <- stepwise(pop_mod_ind, direction = "forward",
                      k=log(nrow(wine_dataset)))
plot(BIC_pop_F, "neato")
title(main = "UG Forward BIC intera popolazione")

#grafico usando BIC backward
BIC_pop_B <- stepwise(pop_mod_sat, k=log(nrow(wine_dataset)))
plot(BIC_pop_B)
```

```r
title(main="UG Backward BIC intera popolazione")

#Both directions
AIC_pop_FB <- stepwise(pop_mod_ind, direction="both")
BIC_pop_FB <- stepwise(pop_mod_ind, direction="both",
                       k=log(nrow(wine_dataset)))

#grafico usando AIC Both
plot(AIC_pop_FB)
title(main = "UG Both AIC intera popolazione")

#grafico usando BIC Both
plot(BIC_pop_FB)
title(main="UG Both BIC intera popolazione")

#procedura glasso

#pacchetti da installare
#install.packages("glasso")
#install.packages("igraph")
library(glasso)
library(igraph)

# 1. Matrice di correlazione
popCor <- cov2cor(CovPop)

# 2. Graphical Lasso
pop_lasso <- glasso(popCor, rho = 0.3)

# 3. Matrice di adiacenza booleana
AM <- pop_lasso$wi != 0
diag(AM) <- FALSE

# 4. Costruzione del grafo igraph
graf_lasso <- graph_from_adjacency_matrix(
  AM,
  mode = "undirected",
  diag = FALSE
)

# 5. Etichette dei nodi
V(graf_lasso)$name <- colnames(wine_dataset)

plot(main="UG glasso intera popolazione con rho= 0.3",
  graf_lasso,
  layout = layout_with_kk,
)


#----------------------------------------
#V1
v1_mod_sat <- cmod(~ .^., data=v1)
v1_mod_ind <- cmod(~.^1, data=v1)
```

```r
#garfico usando penalizzazione AIC Forward
AIC_v1_F <- stepwise(v1_mod_ind, direction="forward")
plot(AIC_v1_F, "neato")
title(main="UG Forward AIC cultivar v1")

#grafico usando AIC Backward
AIC_v1_B <- stepwise(v1_mod_sat)
plot(AIC_v1_B, "neato")
title(main = "UG Backward AIC cultivar v1")

#grafico usando BIC forward
BIC_v1_F <- stepwise(v1_mod_ind, direction = "forward", k=log(nrow(v1)))
plot(BIC_v1_F, "neato")
title(main = "UG Forward BIC cultivar v1")

#grafico usando BIC backward
BIC_v1_B <- stepwise(v1_mod_sat, k=log(nrow(v1)))
plot(BIC_v1_B)
title(main="UG Backward BIC cultivar v1")

#Both directions
AIC_v1_FB <- stepwise(v1_mod_ind, direction="both")
BIC_v1_FB <- stepwise(v1_mod_ind, k=log(nrow(v1)), direction="both")

#grafico AIC both
plot(AIC_v1_FB)
title(main = "UG Both AIC cultivar v1")

#grafico BIC both
plot(BIC_v1_FB)
title(main = "UG Both BIC cultivar v1")

# Matrice di correlazione
varCor <- cov2cor(CovV1)

# Graphical Lasso
var_lasso <- glasso(varCor, rho = 0.3)

# Matrice di adiacenza booleana
AM <- var_lasso$wi != 0
diag(AM) <- FALSE

# Costruzione del grafo igraph
graf_lasso <- graph_from_adjacency_matrix(
  AM,
  mode = "undirected",
  diag = FALSE
)

# Etichette dei nodi
V(graf_lasso)$name <- colnames(v1)
```

```
plot(main="UG glasso cultivar v1 con rho= 0.3",
     graf_lasso,
     layout = layout_with_kk,
)

#----------------------------------------
#V2
v2_mod_sat <- cmod(~ .^., data=v2)
v2_mod_ind <- cmod(~.^1, data=v2)

#garfico usando penalizzazione AIC Forward
AIC_v2_F <- stepwise(v2_mod_ind, direction="forward")
plot(AIC_v2_F, "neato")
title(main="UG Forward AIC cultivar v2")

#grafico usando AIC Backward
AIC_v2_B <- stepwise(v2_mod_sat)
plot(AIC_v2_B, "neato")
title(main = "UG Backward AIC cultivar v2")

#grafico usando BIC forward
BIC_v2_F <- stepwise(v2_mod_ind, direction = "forward", k=log(nrow(v2)))
plot(BIC_v2_F, "neato")
title(main = "UG Forward BIC cultivar v2")

#grafico usando BIC backward
BIC_v2_B <- stepwise(v2_mod_sat, k=log(nrow(v2)))
plot(BIC_v2_B)
title(main="UG Backward BIC cultivar v2")

#Both directions
AIC_v2_FB <- stepwise(v2_mod_ind, direction="both")
BIC_v2_FB <- stepwise(v2_mod_ind, k=log(nrow(v2)), direction="both")

#grafico AIC both
plot(AIC_v2_FB)
title(main = "UG Both AIC cultivar v2")

#grafico BIC both
plot(BIC_v2_FB)
title(main = "UG Both BIC cultivar v2")

# Matrice di correlazione
varCor <- cov2cor(CovV2)

# Graphical Lasso
var_lasso <- glasso(varCor, rho = 0.2)

# Matrice di adiacenza booleana
AM <- var_lasso$wi != 0
diag(AM) <- FALSE

# Costruzione del grafo igraph
```

```
graf_lasso <- graph_from_adjacency_matrix(
  AM,
  mode = "undirected",
  diag = FALSE
)

# Etichette dei nodi
V(graf_lasso)$name <- colnames(v2)

plot(main="UG glasso cultivar v2 con rho= 0.3",
     graf_lasso,
     layout = layout_with_kk,
)

#---------------------------------------
#V3
v3_mod_sat <- cmod(~ .^., data=v3)
v3_mod_ind <- cmod(~.^1, data=v3)

#garfico usando penalizzazione AIC Forward
AIC_v3_F <- stepwise(v3_mod_ind, direction="forward")
plot(AIC_v3_F, "neato")
title(main="UG Forward AIC cultivar v3")

#grafico usando AIC Backward
AIC_v3_B <- stepwise(v3_mod_sat)
plot(AIC_v3_B, "neato")
title(main = "UG Backward AIC cultivar v3")

#grafico usando BIC forward
BIC_v3_F <- stepwise(v3_mod_ind, direction = "forward", k=log(nrow(v3)))
plot(BIC_v3_F, "neato")
title(main = "UG Forward BIC cultivar v3")

#grafico usando BIC backward
BIC_v3_B <- stepwise(v3_mod_sat, k=log(nrow(v3)))
plot(BIC_v3_B)
title(main="UG Backward BIC cultivar v3")

#Both directions
AIC_v3_FB <- stepwise(v3_mod_ind, direction="both")
BIC_v3_FB <- stepwise(v3_mod_ind, k=log(nrow(v3)), direction="both")

#grafico AIC both
plot(AIC_v3_FB)
title(main = "UG Both AIC cultivar v3")

#grafico BIC both
plot(BIC_v3_FB)
title(main = "UG Both BIC cultivar v3")

# Matrice di correlazione
varCor <- cov2cor(CovV3)
```

```r
# Graphical Lasso
var_lasso <- glasso(varCor, rho = 0.3)

# Matrice di adiacenza booleana
AM <- var_lasso$wi != 0
diag(AM) <- FALSE

# Costruzione del grafo igraph
graf_lasso <- graph_from_adjacency_matrix(
  AM,
  mode = "undirected",
  diag = FALSE
)

# Etichette dei nodi
V(graf_lasso)$name <- colnames(v3)

plot(main="UG glasso cultivar v3 con rho= 0.3",
     graf_lasso,
     layout = layout_with_kk,
)

#-------------------------------------------------------------
#Grafi diretti (DAG)
# install.packages("bnlearn")
# install.packages("ggm")
# install.packages("graph")

#if (!requireNamespace("BiocManager", quietly = TRUE))
#  install.packages("BiocManager")
#BiocManager::install("graph")

library(bnlearn)
library(graph)
library(ggm)


# Convertire tutte le colonne integer in numeric
wine_dataset <- data.frame(lapply(wine_dataset, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG intera popolazione BIC
DAG_pop <- hc(wine_dataset, score="bic-g")
plot(DAG_pop, main="DAG intera popolazione BIC Score")


DAG_pop <- amat(DAG_pop)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_pop) <- colnames(wine_dataset)
colnames(DAG_pop) <- colnames(wine_dataset)
```

```r
fdag <- fitDag(DAG_pop, CovPop, nrow(wine_dataset))

fdag$dev
fdag$df
#fdag$Shat
#fdag$Ahat
#fdag$Dhat

v1 <-data.frame(lapply(v1, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG cultivar v1 BIC
DAG_v1 <- hc(v1)
plot(DAG_v1, main="DAG cultivar v1")

DAG_v1 <- amat(DAG_v1)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v1) <- colnames(v1)
colnames(DAG_v1) <- colnames(v1)
fdag <- fitDag(DAG_v1, CovV1, nrow(v1))

fdag$dev
fdag$df

v2 <-data.frame(lapply(v2, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG cultivar v2 BIC
DAG_v2 <- hc(v2)
plot(DAG_v2, main="DAG cultivar v2")

DAG_v2 <- amat(DAG_v2)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v2) <- colnames(v2)
colnames(DAG_v2) <- colnames(v2)
fdag <- fitDag(DAG_v2, CovV2, nrow(v2))

fdag$dev
fdag$df

v3 <-data.frame(lapply(v3, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG cultivar v3 BIC
DAG_v3 <- hc(v3)
plot(DAG_v3, main="DAG cultivar v3")

DAG_v3 <- amat(DAG_v3)   # amat() restituisce la matrice binaria del DAG
```

```r
# Assicurati che la matrice abbia nomi
rownames(DAG_v3) <- colnames(v3)
colnames(DAG_v3) <- colnames(v3)
fdag <- fitDag(DAG_v3, CovV3, nrow(v3))

fdag$dev
fdag$df

# Selezione delle variabili di background e target
backgnd_vars <- setdiff(names(wine_dataset), "Clri")
target_vars  <- c("Clri")

# Creazione della blacklist
blacklist <- expand.grid(
  from = target_vars,
  to   = backgnd_vars
)

# Creazione del modello bayesiano con variabile target specificata
#Intera popolazione
target_DAG_pop <- hc(
  wine_dataset,
  blacklist = blacklist
)
plot(target_DAG_pop, main="DAG intera popolazione con Clri target")

DAG_pop <- amat(target_DAG_pop)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_pop) <- colnames(wine_dataset)
colnames(DAG_pop) <- colnames(wine_dataset)
fdag <- fitDag(DAG_pop, CovPop, nrow(wine_dataset))

fdag$dev
fdag$df

#Cultivar v1 con target
target_DAG_v1 <- hc(
  v1,
  blacklist = blacklist
)

plot(target_DAG_v1, main="DAG cultivar v1 con Clri target")

DAG_v1 <- amat(target_DAG_v1)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v1) <- colnames(wine_dataset)
colnames(DAG_v1) <- colnames(wine_dataset)
fdag <- fitDag(DAG_v1, CovV1, nrow(v1))

fdag$dev
```

```
fdag$df

#Cultivar v2 con target
target_DAG_v2 <- hc(
  v2,
  blacklist = blacklist
)

plot(target_DAG_v2, main="DAG cultivar v2 con Clri target")

DAG_v2 <- amat(target_DAG_v2)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v2) <- colnames(wine_dataset)
colnames(DAG_v2) <- colnames(wine_dataset)
fdag <- fitDag(DAG_v2, CovV2, nrow(v2))

fdag$dev
fdag$df

#Cultivar v3 con target
target_DAG_v3 <- hc(
  v3,
  blacklist = blacklist
)

plot(target_DAG_v3, main="DAG cultivar v3 con Clri target")

DAG_v3 <- amat(target_DAG_v3)   # amat() restituisce la matrice binaria del DAG

# Assicurati che la matrice abbia nomi
rownames(DAG_v3) <- colnames(wine_dataset)
colnames(DAG_v3) <- colnames(wine_dataset)
fdag <- fitDag(DAG_v3, CovV3, nrow(v3))

fdag$dev
fdag$df

#-------------------------------------------
#Caso con Wine dataset

wine <-data.frame(lapply(wine, function(x) {
  if (is.integer(x)) as.numeric(x) else x
}))

#Creazione grafico DAG intera popolazione con la variabile discreta cultivar BIC
DAG_wine <-hc(wine)
plot(DAG_wine, main="DAG wine dataset")


# Con conoscenze a priori
# Conversione di eventuali colonne integer in numeric
wine <- data.frame(lapply(wine, function(x) {
```

```
  if (is.integer(x)) as.numeric(x) else x
}))

# Definizione delle variabili target e esogene
target_vars <- c("Clri")    # variabile target
exogenous_vars <- c("Cult") # variabile esogena

# Tutte le altre variabili
other_vars <- setdiff(names(wine), c(target_vars, exogenous_vars))

# Creazione della blacklist
# 1) Target non può influenzare nessun'altra variabile
# 2) Variabile esogena non può essere influenzata da nessun'altra
blacklist <- rbind(
  expand.grid(from = target_vars, to = other_vars),
  expand.grid(from = other_vars, to = exogenous_vars),
  expand.grid(from = target_vars, to = exogenous_vars)
)

# Apprendimento del DAG con Hill-Climbing usando BIC-CG
DAG_wine <- hc(wine, score = "bic-cg", blacklist = blacklist)

# Visualizzazione del DAG
plot(DAG_wine, main = "DAG Wine dataset con target Clri e Cult esogena")

#-------------------------------------------------------------
#Stimare un modello predittivo

#-----------------------------------------
#Intera Popolazione

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = wine_dataset)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = wine_dataset)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope = scope, direction = "both")
summary(both_model_AIC)
```

```r
#BIC
#backward
back_model_BIC <-step(full_model, scope=scope, direction = "backward",
                      k=log(nrow(wine_dataset)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope=scope, direction = "forward",
                       k=log(nrow(wine_dataset)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope=scope, direction = "both",
                       k=log(nrow(wine_dataset)))
summary(both_model_BIC)
#-----------------------------------------
#Cultivar v1

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = v1)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = v1)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope=scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope=scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope=scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
                      k=log(nrow(v1)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
                       k=log(nrow(v1)))
summary(forw_model_BIC)
```

```
#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
                       k=log(nrow(v1)))
summary(both_model_BIC)

#-----------------------------------------
#Cultivar v2

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = v2)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = v2)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope = scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
                      k=log(nrow(v2)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
                       k=log(nrow(v2)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
                       k=log(nrow(v2)))
summary(both_model_BIC)

#-----------------------------------------
#Cultivar v3

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = v3)
```

```r
#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = v3)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)

#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope=scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
                      k=log(nrow(v3)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
                       k=log(nrow(v3)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
                       k=log(nrow(v3)))
summary(both_model_BIC)

#-----------------------------------------
#Wine dataset

#definiamo un modello nullo
null_model <- lm ( Clri ~ 1, data = wine)

#definiamo un modello saturo
full_model <-lm(Clri ~ . , data = wine)

# Definizioni dello scope per definire gli intervalli di lavoro
scope <- list ( lower = formula ( null_model ), upper = formula ( full_model ))

#AIC

#backward
back_model_AIC <- step(full_model, scope = scope, direction = "backward")
summary(back_model_AIC)
```

```
#forward
forw_model_AIC <- step(null_model, scope = scope, direction = "forward")
summary(forw_model_AIC)

#both
both_model_AIC <- step(null_model, scope = scope, direction = "both")
summary(both_model_AIC)

#BIC
#backward
back_model_BIC <-step(full_model, scope = scope, direction = "backward",
                      k=log(nrow(wine)))
summary(back_model_BIC)

#forward
forw_model_BIC <- step(null_model, scope = scope, direction = "forward",
                       k=log(nrow(wine)))
summary(forw_model_BIC)

#both
both_model_BIC <- step(null_model, scope = scope, direction = "both",
                       k=log(nrow(wine)))
summary(both_model_BIC)
```