Dati sul peso alla nascita

Autore: Vittorio Casula
Matricola: 7073230
Insegnamento: Foundation of Statistical Modelling
CdLM: Intelligenza Artificiale

Analisi del dataset “birthwt”

Descrizione Dataset

Si vuole analizzare i fattori di rischio del basso peso alla nascita.
Il dataset contiene dati raccolti su 189 bambini nati al Baystate Medical Center, Springfield, Mass nel 1986.

- Numerosità campionaria: 189
- Numero variabili: 10

Figure 1: Baystate Medical Center

Descrizione variabili

La variabile obiettivo è rappresentato dal peso del bambino espresso in grammi (btw) e dalla corrispettiva variabile dicotomizzata (low) rispetto a 2500 grammi.

- **low** → variabile btw(*) dicotomizzata
  - 1 se peso bambino < 2.5 Kg
  - 0 altrimenti
- **age** → età della madre (in anni)
- **lwt** → peso della madre all'ultimo periodo mestruale (in libbre, 1 Kg = 2,205 libbre)
- **race** → etnia della madre
  - 1 se bianca
  - 2 se nera
  - 3 altro
- **smoke** → madre fumatrice
  - 1 se la madre fuma
  - 0 se la madre non fuma
- **plt** → numero di parti prematuri (precedenti a quello del bimbo corrente)
- **ht** → storia familiare di ipertensione (pressione alta del sangue)
  - 1 se presente
  - 0 se assente
- **ui** → irritabilità uterina nella madre
  - 1 se presente
  - 0 se assente
- **ftw** → numero di visite dal ginecologo nel primo trimestre
- **bwt** (*) → peso del bambino alla nascita (in grammi)

Breve ricerca online

I neonati possono essere sottopeso perché i genitori sono di bassa statura, per un problema di salute della madre oppure del consumo da parte di quest'ultima di sostanze o alcolici durante la gravidanza.

Fra le patologie della madre che aumentano il rischio di avere un bambino sottopeso vi sono:

- Pressione arteriosa alta (ipertensione) associata alla gravidanza o cronica
- Anomalie dell'utero
- Diabete, insufficienza renale, cardiopatia, grave malnutrizione, ecc...

Oltre a questi fattori abbiamo anche:

- tagli cesarei non indicati
- età avanzata della madre
- nascite multiple (gemelli ad esempio)

Analisi esplorativa

Avendo a disposizione la variabile obiettivo sia nella forma continua (btw) sia discreta (low) è possibile fare un'analisi dei modelli sia per il modello di regressione lineare (multipla) e sia per modello di regressione logistica.

Inoltre dobbiamo considerare che per un numero così alto di varibili (10) e dati a disposizione sono pochi (189 unità).

Il dataset presenta dei duplicati: questo può essere dovuto alla presenza di bambini gemelli (oppure un parto trigimino cioè di 3 figli) o semplici coincidenze (it's quite difficult).

```
data("birthwt", package = "MASS")
nrow(birthwt)
```

```
## [1] 189
```

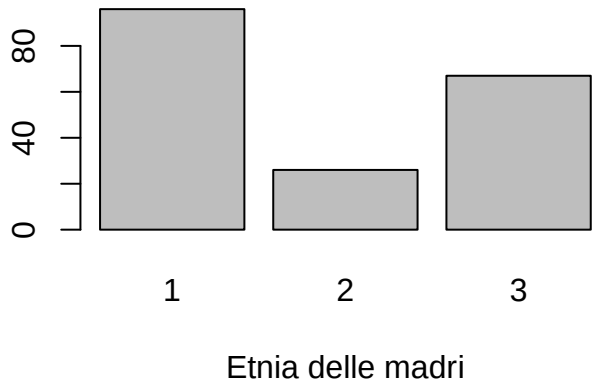
```
nrow(unique(birthwt))
```

```
## [1] 184
```

Andando a vedere il numero di righe uniche (distinte) notiamo che queste differiscono dal numero totale di righe nel dataframe. Ci sono 5 duplicati (189 totali meno 184 uniche).

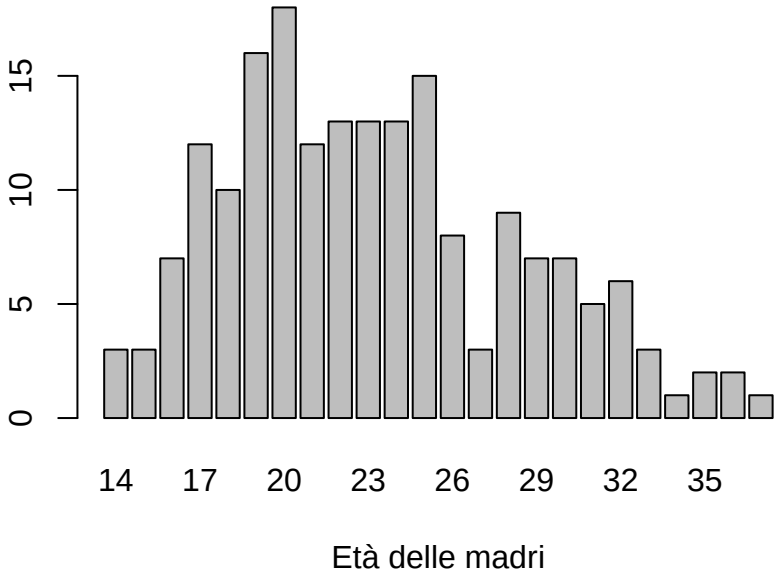
Andiamo a fare un'analisi esplorativa del dataset mediante plot di alcune variabili.

```
plot(factor(birthwt$race), xlab = "Etnia delle madri")
```

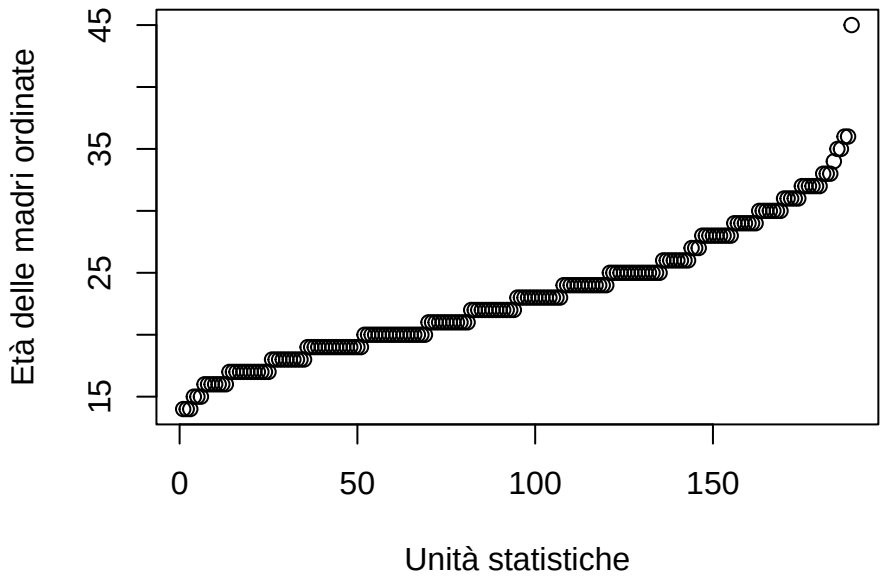


Notiamo che nel dataset abbiamo una maggioranza di madri di etnia bianca e che la somma tra le madri di etnia nera e altro equivalgono a quelle di etnia bianca.

```
plot(factor(birthwt$age), xlab = "Età delle madri")
```



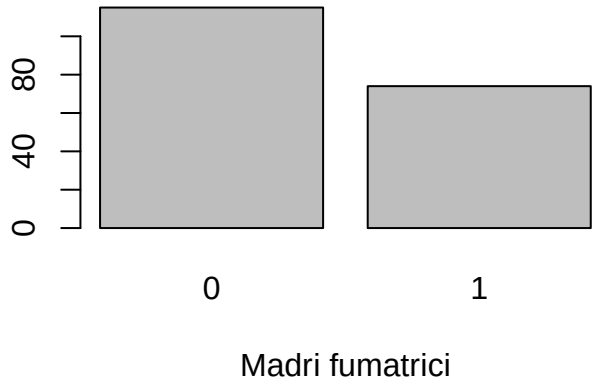

```
plot(sort(birthwt$age), ylab = "Età delle madri ordinate", xlab = "Unità statistiche")
```



Da un punto di vista dell'età le madri si concentrano nel range 18-25, quindi sono a tutti gli effetti madri

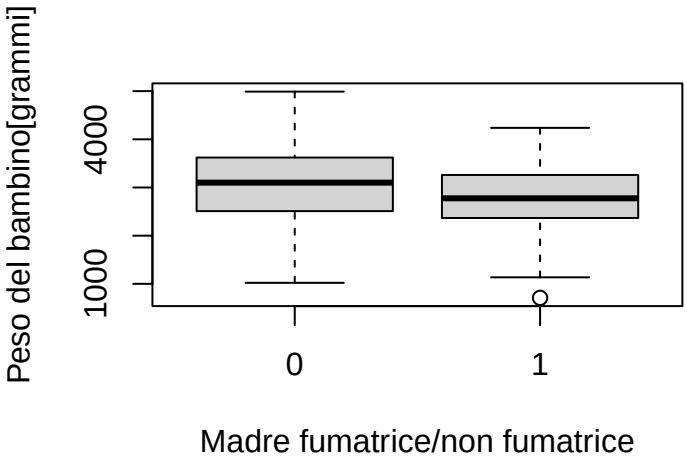
giovani. Però, come mostra il grafico delle età ordinate si possono raggiungere valori anche oltre i 35 anni e anche al di sotto dei 18.

```
plot(factor(birthwt$smoke), xlab = "Madri fumatrici")
```



Tra queste madri quelle non fumatrici superano in media quelle fumatrici, ciò nonostante dovremmo studiare qual'è l'impatto del fumo sul basso peso del bambino.

```
plot(factor(birthwt$smoke), birthwt$bwt, xlab = "Madre fumatrice/non fumatrice",
    ylab = "Peso del bambino[grammi]")
```



Il grafico tra madre fumatrici/non e il peso del bambino sembra non influire su quest'ultimo perchè mediamente siamo sempre al di sopra di 2500 grammi, soglia limite per dire se un bambino e sotto peso/sovrappeso.

Step-by-step

- Adattamento del modello di regressione lineare (multipla) completo e tramite procedure stepwise (backward, forward e both) con criteri di penalizzazione AIC, BIC e indice di Mallow, criterio dell'R^2 aggiustato.
- Adattamento del modello di regressione logistica completo e tramite procedure stepwise (backward, forward),
- Adattamento di modelli log lineari (Undirected Graph): sopratutto per capire la struttura di indipendenza
- Adattamento di Bayesian Networks (DAG): obbiettivo finale (modello più strutturato)

Regressione Lineare Multipla

```
library(Matrix)
```

Per adattare un modello di regressione lineare multipla considero come variabile obiettivo la variabile che indica il peso del bambino espresso in grammi (bwt) e converto tutte le variabili esplicative a tipo numeric

```
data("birthwt", package = "MASS")

dataset <- with(birthwt, {
    pesoBimboGrammi <- as.numeric(bwt)
    etaMadre = as.numeric(age)
    pesoMadre = as.numeric(lwt/2.205)
    etnia <- as.numeric(race) - 1  #così abbiamo 0 = bianco, 1 = nero, 2 = altro
    fumaMadre = as.numeric(smoke)
    nPartiPrematuri = as.numeric(ptl)
    ipertensioneStoria = as.numeric(ht)
    irritUterinaMadre = as.numeric(ui)
    visiteGine <- as.numeric(ftv)

    data.frame(pesoBimboGrammi, etaMadre, pesoMadre, etnia, fumaMadre,
        nPartiPrematuri, ipertensioneStoria, irritUterinaMadre,
        visiteGine)
})

attach(dataset)
```

Informazioni generali dataset modificato. Verifichiamo che il dataset non abbia ridondanze (matrice x rango massimo=10).

```
summary(dataset)
```

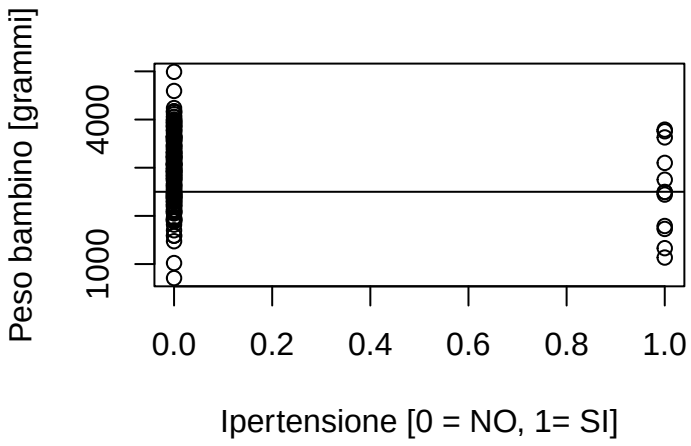
```
##  pesoBimboGrammi    etaMadre       pesoMadre          etnia
##  Min.   : 709    Min.   :14.00   Min.   : 36.28   Min.   :0.0000
##  1st Qu.:2414    1st Qu.:19.00   1st Qu.: 49.89   1st Qu.:0.0000
##  Median :2977    Median :23.00   Median : 54.88   Median :0.0000
##  Mean   :2945    Mean   :23.24   Mean   : 58.87   Mean   :0.8466
##  3rd Qu.:3487    3rd Qu.:26.00   3rd Qu.: 63.49   3rd Qu.:2.0000
##  Max.   :4990    Max.   :45.00   Max.   :113.38   Max.   :2.0000
##    fumaMadre      nPartiPrematuri  ipertensioneStoria irritUterinaMadre
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000    Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000    1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.00000    Median :0.0000
##  Mean   :0.3915   Mean   :0.1958   Mean   :0.06349    Mean   :0.1481
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0.00000    3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :3.0000   Max.   :1.00000    Max.   :1.0000
##    visiteGine
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.7937
##  3rd Qu.:1.0000
##  Max.   :6.0000
```

```
vect_one = rep(1, nrow(dataset))

mat_x = cbind(vect_one, pesoBimboGrammi, etaMadre, pesoMadre,
    etnia, fumaMadre, nPartiPrematuri, ipertensioneStoria, irritUterinaMadre,
    visiteGine)

rank = rankMatrix(mat_x)[1]  # rango massimo = 10-> no ridondanza
rank
```

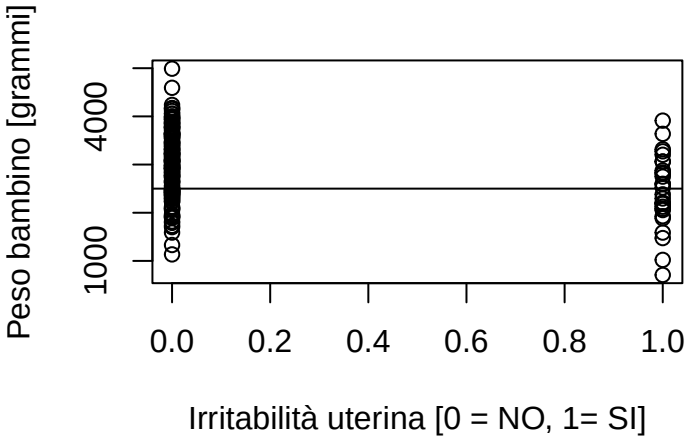
```
## [1] 10
```

Plot iniziali

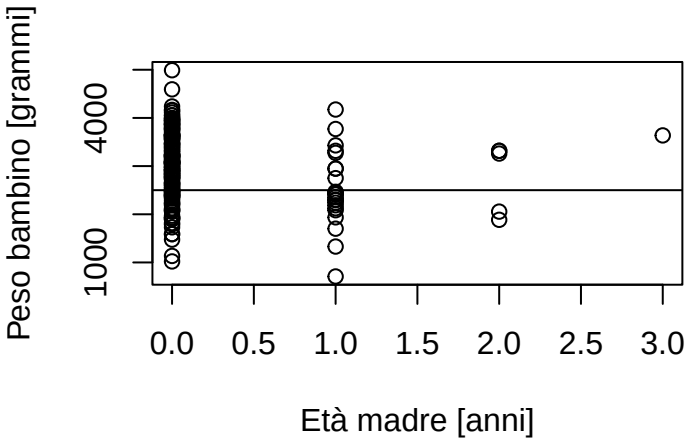
```
plot(ipertensioneStoria, pesoBimboGrammi, xlab = "Ipertensione [0 = NO, 1= SI]",
    ylab = "Peso bambino [grammi]")
abline(2500, 0)
```




```
plot(irritUterinaMadre, pesoBimboGrammi, xlab = "Irritabilità uterina [0 = NO, 1= SI]",
    ylab = "Peso bambino [grammi]")
abline(2500, 0)
```




```
plot(nPartiPrematuri, pesoBimboGrammi, xlab = "Età madre [anni]",
    ylab = "Peso bambino [grammi]")
abline(2500, 0)
```



Adatto un modello di regressione con tutte le variabili

```
full_model = lm(pesoBimboGrammi ~ etaMadre + pesoMadre + etnia +
    fumaMadre + nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
    visiteGine, data = dataset)
summary(full_model)
```

```
##
## Call:
## lm(formula = pesoBimboGrammi ~ etaMadre + pesoMadre + etnia +
```

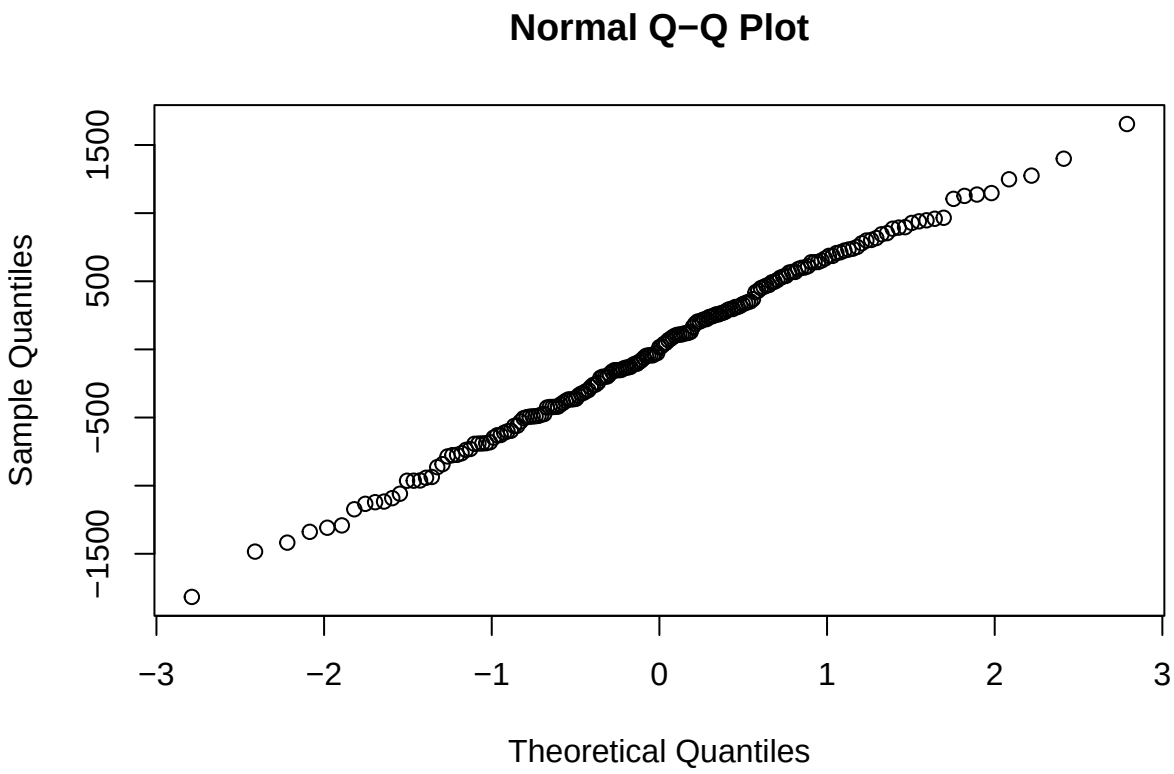
```
##     fumaMadre + nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
##     visiteGine, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1816.51  -426.79    16.29   492.06  1654.01
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2940.9699   316.0270   9.306  < 2e-16 ***
## etaMadre             -0.2658     9.5947  -0.028  0.97793
## pesoMadre             7.5745     3.7483   2.021  0.04478 *
## etnia              -188.4895    57.7339  -3.265  0.00131 **
## fumaMadre          -358.4552   107.5172  -3.334  0.00104 **
## nPartiPrematuri     -51.1526   103.0003  -0.497  0.62006
## ipertensioneStoria -600.6465   204.3454  -2.939  0.00372 **
## irritUterinaMadre  -511.2513   140.2792  -3.645  0.00035 ***
## visiteGine          -15.5358    46.9354  -0.331  0.74103
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 656.9 on 180 degrees of freedom
## Multiple R-squared:  0.223,  Adjusted R-squared:  0.1884
## F-statistic: 6.456 on 8 and 180 DF,  p-value: 2.232e-07
```

```
confint(full_model)
```

```
##                            2.5 %     97.5 %
## (Intercept)         2317.3756673 3564.56408
## etaMadre             -19.1984464   18.66683
## pesoMadre              0.1782487   14.97068
## etnia               -302.4118096  -74.56722
## fumaMadre           -570.6114961 -146.29888
## nPartiPrematuri     -254.3958763  152.09076
## ipertensioneStoria -1003.8672029 -197.42585
## irritUterinaMadre   -788.0544699 -234.44804
## visiteGine          -108.1501301   77.07853
```

Questo non è un modello che ci soddisfa da un punto di vista di interpretabilità. Infatti le misure di associazione (ovvero i valore delle stime puntuali dei parametri) sono molto elevati e nella stragrande maggioranza delle variabili coinvolte abbiamo un errore standard elevatissimo. Per questo modello le variabili altamente significative sono l'irritabilità uterina, la storia di ipertensione, l'etnia e se la madre fuma o meno. Molti di questi sono verosimili, per alcuni invece la dipendenza non era prevista con tanta significatività. Complessivamente da un punto di vista tecnico la statistica F è altamente significativa e quindi la devianza spiegata dal modello è molto più grande della devianza spiegata dall'errore, secondo la sua distribuzione e secondo i suoi gradi di libertà. L'indice di determinazione (sia quelo base sia quello aggiustato) non mostrano una grande bontà di adattamento del modello considerato. Questo livello di adattamento non soddisfa le aspettative ma non era così inaspettato: infatti le variabili coninvolte non saranno senz'altro legate da coefficienti lineari per la loro natura eterogenea (alcune variabili indicano caratteristiche di background della madre, altri possibili fattori di rischio, altri fatti precedenti alla gravidanza).

```
qqnorm(full_model$residuals)
```



Ha senso adesso considerare un modello di regressione lineare multipla con sole le variabili significative del modello completo.

```
model = lm(pesoBimboGrammi ~ fumaMadre + pesoMadre + etnia +
    ipertensioneStoria + irritUterinaMadre, data = dataset)
summary(model)
```

```
##
## Call:
## lm(formula = pesoBimboGrammi ~ fumaMadre + pesoMadre + etnia +
##     ipertensioneStoria + irritUterinaMadre, data = dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1818.56  -454.51    -2.53   475.70  1651.06
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2916.590    243.160  11.995  < 2e-16 ***
## fumaMadre          -366.135    104.342  -3.509 0.000566 ***
## pesoMadre             7.571      3.634   2.084 0.038581 *
## etnia              -187.849     56.349  -3.334 0.001037 **
## ipertensioneStoria -595.820    201.515  -2.957 0.003519 **
```

```
## irritUterinaMadre  -523.419    135.976  -3.849 0.000164 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 652.2 on 183 degrees of freedom
## Multiple R-squared:  0.2214, Adjusted R-squared:  0.2001
## F-statistic:  10.4 on 5 and 183 DF,  p-value: 8.451e-09
```

Quello che succede è che le varibili coinvolte sono significative, nessuna ha perso di significatività ma da un punto di vista dell'R^2 e della statistica F è cambiato poco o nulla.

A questo punto è bene farsi guidare da procedure più strutturate come quelle stepwise.Considero il modello nullo e quello completo.

Considero adesso tutte le possibili combinazioni.

```
forw_lik = step(empty_model, scope = formula(full_model), direction = "forward",
    k = 0)
forw_aic = step(empty_model, scope = formula(full_model), direction = "forward",
    k = 2)
forw_bic = step(empty_model, scope = formula(full_model), direction = "forward",
    k = log(length(pesoBimboGrammi)))

back_lik = step(full_model, scope = formula(empty_model), direction = "backward",
    k = 0)
back_aic = step(full_model, scope = formula(empty_model), direction = "backward",
    k = 2)
back_bic = step(full_model, scope = formula(empty_model), direction = "backward",
    k = log(length(pesoBimboGrammi)))

both_lik = step(full_model, scope = formula(full_model), direction = "both",
    k = 0)
both_aic = step(full_model, scope = formula(full_model), direction = "both",
    k = 2)
both_bic = step(full_model, scope = formula(full_model), direction = "both",
    k = log(length(pesoBimboGrammi)))
```

L'output dei summary dei vari modelli non viene mostrato ma questi sono molto simili tra di loro e non si discotano molto dal modello completo (statistica F altamente significativa, indice di determinazione attorno allo 0.20 e misure di associazione alte nel complesso). Possiamo però vedere quali variabili sono risultate facenti parte dei vari modelli a fine procedure. Di queste potremmo scegliere quelle che hanno occorrenze (totali) più alte.

Formule dei modelli risultati

```
formula(forw_lik)
```

```
## pesoBimboGrammi ~ irritUterinaMadre + etnia + fumaMadre + ipertensioneStoria +
##     pesoMadre + nPartiPrematuri + visiteGine + etaMadre
```

```
formula(forw_aic)
```

```
## pesoBimboGrammi ~ irritUterinaMadre + etnia + fumaMadre + ipertensioneStoria +
##     pesoMadre
```

```
formula(forw_bic)
```

```
## pesoBimboGrammi ~ irritUterinaMadre + etnia + fumaMadre + ipertensioneStoria
```

```
formula(back_lik)
```

```
## pesoBimboGrammi ~ etaMadre + pesoMadre + etnia + fumaMadre +
##     nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
##     visiteGine
```

```
formula(back_aic)
```

```
## pesoBimboGrammi ~ pesoMadre + etnia + fumaMadre + ipertensioneStoria +
##     irritUterinaMadre
```

```
formula(back_bic)
```

```
## pesoBimboGrammi ~ etnia + fumaMadre + ipertensioneStoria + irritUterinaMadre
```

```
formula(both_lik)
```

```
## pesoBimboGrammi ~ etaMadre + pesoMadre + etnia + fumaMadre +
##     nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
##     visiteGine
```

```
formula(both_aic)
```

```
## pesoBimboGrammi ~ pesoMadre + etnia + fumaMadre + ipertensioneStoria +
##     irritUterinaMadre
```

```
formula(both_bic)
```

```
## pesoBimboGrammi ~ etnia + fumaMadre + ipertensioneStoria + irritUterinaMadre
```

Occorrenze delle variabili nei modelli (modelli totali = 9):

- etaMadre: 3/9
- pesoMadre: 6/9
- etnia: 9/9
- fumaMadre: 8/9
- nPartiPrematuri: 3/9
- ipertensioneStoria: 9/9
- irritUterinaMadre: 9/9
- visiteGine: 3/9

E' interessante notare come le procedure stepwise con k=0 (ovvero con il solo confronto della funzione di verosimiglianza) selezionano in tutti e tre i casi il modello completo. Notiamo come la variabile che indica l'età della madre ha poche occorrenze nei modelli e lo stesso il numero di visite dal ginecologo (molto plausibile tale fatto). La cosa abbastanza strana è il numero di parti prematuri abbia un numero così piccolo di occorrenze.

Concludendo, potrebbe essere un'idea considerare il modello con le variabili che hanno un numero di occorrenze pari a 8/9 e 9/9, che si dimostra un modello parsimonioso perchè composto da solo 4 variabili (meno della metà della variabili totali). Un critero di questo tipo è gradito anche per il fatto che tecnicamente i modelli provati sono molto simili.

```
best_fit = lm(pesoBimboGrammi ~ etnia + fumaMadre + ipertensioneStoria +
    irritUterinaMadre, data = dataset)
summary(best_fit)
```

```
##
## Call:
## lm(formula = pesoBimboGrammi ~ etnia + fumaMadre + ipertensioneStoria +
##     irritUterinaMadre, data = dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1812.5  -443.0    26.6   470.6  1600.6
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         3389.40      88.18  38.436  < 2e-16 ***
## etnia               -210.77      55.76  -3.780 0.000212 ***
## fumaMadre           -389.38     104.68  -3.720 0.000265 ***
## ipertensioneStoria  -497.12     197.64  -2.515 0.012750 *
## irritUterinaMadre   -555.94     136.30  -4.079 6.73e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 658.1 on 184 degrees of freedom
## Multiple R-squared:  0.2029, Adjusted R-squared:  0.1856
## F-statistic: 11.71 on 4 and 184 DF,  p-value: 1.712e-08
```

Un'altro criterio che possiamo testare è quello dell'indice di Mallow che è specifico per il modello di regressione ed è molto simile al criterio di penalizzazione AIC. Il principio è quello di scegliere il modello per cui si ha il rischio di previsione minimo. Il rischio di previsione è definito come la somma tra la devianza spiegata dagli errori e la stima corretta della varianza del modello completo (moltiplicata per 2 volte il valore assoluto del numero di variabili del modello completo). L'indice di Mallow viene indicato spesso con "Cp". E' un algoritmo che si basa sul metodo Branch & Bound.

```
library(leaps)
```

```
## Warning: il pacchetto 'leaps' è stato creato con R versione 4.1.2
```

```
model_full = lm(pesoBimboGrammi ~ etaMadre + pesoMadre + etnia +
    fumaMadre + nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
    visiteGine)
y = pesoBimboGrammi
```

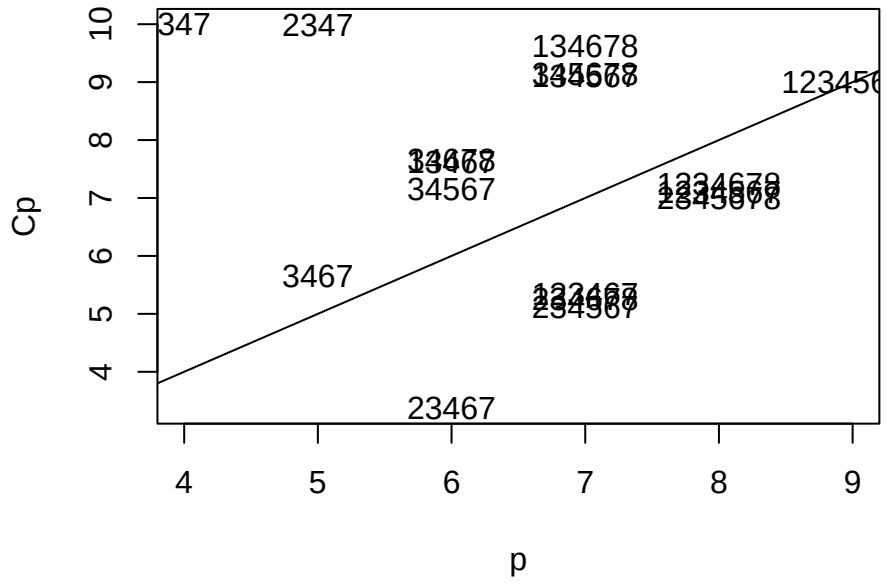
```
x = model.matrix(model_full)[, -1]  #rimuovo l'intercetta dalla matrice x
leapcp = leaps(x, y, method = "Cp")
# in quale riga si manifesta il minimo Cp, prendo l'intera
# riga della matrice per capire le variabili
leapcp$which[which.min(leapcp$Cp), ]
```

```
##     1     2     3     4     5     6     7     8
## FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
```

```
library(faraway)  #serve solo per usare la funzione Cpplot
```

```
## Warning: il pacchetto 'faraway' è stato creato con R versione 4.1.2
```

```
Cpplot(leapcp)  # mi conferma che il modello è quello selezionato dalla procedura leaps e corrisponde n
```




```
# 23467
```

Il criterio dell'indice di Mallow selezione il modello con le variabili: pesoMadre + etnia + fumaMadre + ipertensioneStoria + irritUterinaMadre. A livello grafico il modello è quello con Cp minimo dunque quello più in basso possibile. Come possiamo notare non si discosta molto dal modello "best_fit" e differiscono solo per la variabile che indica il peso della madre.

Per finire con il modello di regressione lineare multipla potremmo provare il criterio dell'R^2 aggiustato che tende a scegliere modelli complessi (perchè semplicemente se aumento le variabili aumento leggermente l'R^2 aggiustato).

```
leap_adjr = leaps(x, y, method = "adjr")
m = max(leap_adjr$adjr2)
pos = which(leap_adjr$adjr == m)
leap_adjr$which[pos, ]
```

```
##     1     2     3     4     5     6     7     8
## FALSE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE FALSE
```

Questa ultima prova selezione il modello selezionato con l'indice di Mallow.

```
detach()
```

Regressione Logistica

Prepariamoci ad adattare un modello di regressione logistica, idoneo al fatto che la variabile obiettivo è binaria e quindi la previsione coincide con la classificazione.

Prima di tutto è stata fatta una suddivisione in livelli di alcune variabili perchè ci sono casi in cui un livello presenta in percentuale sul totale molte più unità di altri livelli come ad esempio numero di visite dal ginecologo o il numero di parti prematuri.

E' bene considerare che per 10 variabili abbiamo troppi pochi dati. Infatti nel caso di modello di regressione logistica non abbiamo la distribuzione esatta in campioni finiti ma solo la distribuzione asintotica. La "non-significatività campionaria" dipende dalla poca numerosità campionaria (i p-value ne risentono).

```
data("birthwt", package = "MASS")
dataset <- with(birthwt, {
    pesoBimbo = factor(low, labels = c("sovrappeso", "sottopeso"))
    etaMadre = as.numeric(age)
    pesoMadre = as.numeric(lwt/2.205)
    etnia <- factor(race, labels = c("bianca", "nera", "altro"))
    fumaMadre = factor(smoke, labels = c("no", "si"))
    nPartiPrematuri = factor(ptl)
    levels(nPartiPrematuri)[-(1)] <- "1+"
    ipertensioneStoria = factor(ht, labels = c("no", "si"))
    irritUterinaMadre = factor(ui, labels = c("no", "si"))
    nVisiteGine <- factor(ftv)
    levels(nVisiteGine)[-(1:2)] <- "2+"

    data.frame(pesoBimbo, etaMadre, pesoMadre, etnia, fumaMadre,
        nPartiPrematuri, ipertensioneStoria, irritUterinaMadre,
        nVisiteGine)
})

str(dataset)
```

```
## 'data.frame':    189 obs. of  9 variables:
##  $ pesoBimbo         : Factor w/ 2 levels "sovrappeso","sottopeso": 1 1 1 1 1 1 1 1 1 ...
##  $ etaMadre          : num  19 33 20 21 18 21 22 17 29 26 ...
##  $ pesoMadre         : num  82.5 70.3 47.6 49 48.5 ...
##  $ etnia             : Factor w/ 3 levels "bianca","nera",..: 2 3 1 1 1 3 1 3 1 1 ...
##  $ fumaMadre         : Factor w/ 2 levels "no","si": 1 1 2 2 2 1 1 1 2 2 ...
##  $ nPartiPrematuri   : Factor w/ 2 levels "0","1+": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ipertensioneStoria: Factor w/ 2 levels "no","si": 1 1 1 1 1 1 1 1 1 1 ...
##  $ irritUterinaMadre : Factor w/ 2 levels "no","si": 2 1 1 2 2 1 1 1 1 1 ...
##  $ nVisiteGine       : Factor w/ 3 levels "0","1","2+": 1 3 2 3 1 1 2 2 2 1 ...
```

```
summary(dataset)
```

```
##       pesoBimbo      etaMadre       pesoMadre         etnia    fumaMadre
##  sovrappeso:130   Min.   :14.00   Min.   : 36.28   bianca:96   no:115
##  sottopeso : 59   1st Qu.:19.00   1st Qu.: 49.89   nera  :26   si: 74
##                   Median :23.00   Median : 54.88   altro :67
##                   Mean   :23.24   Mean   : 58.87
##                   3rd Qu.:26.00   3rd Qu.: 63.49
##                   Max.   :45.00   Max.   :113.38
```

```
##  nPartiPrematuri ipertensioneStoria irritUterinaMadre nVisiteGine
##  0 :159          no:177             no:161            0 :100
##  1+: 30          si: 12             si: 28            1 : 47
##                                                       2+: 42
##
##
##
```

```
head(dataset)
```

```
##    pesoBimbo etaMadre pesoMadre  etnia fumaMadre nPartiPrematuri
## 1 sovrappeso       19  82.53968   nera        no               0
## 2 sovrappeso       33  70.29478  altro        no               0
## 3 sovrappeso       20  47.61905 bianca        si               0
## 4 sovrappeso       21  48.97959 bianca        si               0
## 5 sovrappeso       18  48.52608 bianca        si               0
## 6 sovrappeso       21  56.23583  altro        no               0
##   ipertensioneStoria irritUterinaMadre nVisiteGine
## 1                 no                si           0
## 2                 no                no          2+
## 3                 no                no           1
## 4                 no                si          2+
## 5                 no                si           0
## 6                 no                no           0
```

```
attach(dataset)
```

```
## I seguenti oggetti sono mascherati da dataset (pos = 4):
##
##     etaMadre, etnia, fumaMadre, ipertensioneStoria, irritUterinaMadre,
##     nPartiPrematuri, pesoMadre
```

Adattiamo un medello di regressione logistica con tutte le variabili.

```
empty_model = glm(pesoBimbo ~ 1, family = binomial)
full_model = glm(pesoBimbo ~ etaMadre + pesoMadre + etnia + fumaMadre +
    nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
    nVisiteGine, family = "binomial")
summary(full_model)
```

```
##
## Call:
## glm(formula = pesoBimbo ~ etaMadre + pesoMadre + etnia + fumaMadre +
##     nPartiPrematuri + ipertensioneStoria + irritUterinaMadre +
##     nVisiteGine, family = "binomial")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7038  -0.8068  -0.5008   0.8835   2.2152
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)           0.82302    1.24471   0.661  0.50848
## etaMadre             -0.03723    0.03870  -0.962  0.33602
## pesoMadre            -0.03451    0.01561  -2.211  0.02705 *
## etnianera             1.19241    0.53596   2.225  0.02609 *
## etniaaltro            0.74068    0.46174   1.604  0.10869
## fumaMadresi           0.75553    0.42502   1.778  0.07546 .
## nPartiPrematuri1+     1.34376    0.48062   2.796  0.00518 **
## ipertensioneStoriasi  1.91317    0.72074   2.654  0.00794 **
## irritUterinaMadresi   0.68020    0.46434   1.465  0.14296
## nVisiteGine1         -0.43638    0.47939  -0.910  0.36268
## nVisiteGine2+         0.17901    0.45638   0.392  0.69488
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 195.48  on 178  degrees of freedom
## AIC: 217.48
##
## Number of Fisher Scoring iterations: 4
```

```
summary(full_model$fitted.values)  #probabilità
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01605 0.14098 0.26816 0.31217 0.44059 0.90432
```

```
glm.probs = predict(full_model, type = "response")  #equivalente a full_model$fitted
summary(glm.probs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.01605 0.14098 0.26816 0.31217 0.44059 0.90432
```

```
glm.pred = ifelse(glm.probs > 0.5, "sottopeso", "sovrappeso")
table(glm.pred, pesoBimbo)
```

```
##             pesoBimbo
## glm.pred     sovrappeso sottopeso
##   sottopeso          14        22
##   sovrappeso        116        37
```

```
mean(glm.pred == pesoBimbo)
```

```
## [1] 0.7301587
```

Il modello completo mostra che le variabili significative sono l'etnia (nei due livelli nera e altro), il fatto che la madre sia fumatrice, il numero di parti prematuri superiori o uguali a uno e infine la variabile che indica la presenza della storia dell'ipertensione. Si tratta di tutte variabili verosimili ma è stata esclusa l'irritabilità uterina della madre.

Vediamo adesso quanto questo modello riesce a classificare correttamente il peso del bambino. In particolare, trasformiamo le probabilità in classificazioni fissando una soglia a 0,5. Per fare ciò, utilizzo un comando

ifelse(test, yes, no). Dalla tabella, le istanze sulle diagonale secondaria indicano la classificazione corretta e fuori da questa la presenza di errori. La media fornisce una proporzione di 0,74, quindi tutto sommato non molti errori. Per valutare però la performance della predizione dovremmo provare con dataset diverso da quello di train (che purtroppo non abbiamo a disposizone).

```
pseudorsquared = 1 - (full_model$deviance/full_model$null.deviance)
pseudorsquared
```

```
## [1] 0.1670267
```

Questo modello è abbastanza soddisfacente da un punto di vista di variabili coinvolte ma se andiamo a calcolare l'indice di determinazione (pseudo) risulta nuovamente basso (come il modello di regressione lineare multipla).

Proviamo le procedure stepwise di tipo forward

```
forw_lik = step(empty_model, scope = formula(full_model), direction = "forward",
    k = 0)
forw_aic = step(empty_model, scope = formula(full_model), direction = "forward",
    k = 2)
forw_bic = step(empty_model, scope = formula(full_model), direction = "forward",
    k = log(length(pesoBimbo)))
```

```
formula(forw_lik)
```

```
## pesoBimbo ~ nPartiPrematuri + etnia + fumaMadre + pesoMadre +
##     ipertensioneStoria + irritUterinaMadre + nVisiteGine + etaMadre
```

```
formula(forw_aic)
```

```
## pesoBimbo ~ nPartiPrematuri + etaMadre + ipertensioneStoria +
##     pesoMadre + irritUterinaMadre
```

```
formula(forw_bic)
```

```
## pesoBimbo ~ nPartiPrematuri
```

Tra i tre modelli scelti il terzo è estremamente parsimonioso e non verosimile. Il criterio AIC coinvolge delle ottime variabili. Proviamo la procedura backward per il solo criterio AIC.

```
back_aic = step(full_model, scope = formula(empty_model), direction = "backward",
    k = 2)
formula(back_aic)
```

Differiscono solo per l'età della madre. Considero quindi il modello più parsimonioso (escludendo l'età della madre).

```
fit = glm(pesoBimbo ~ nPartiPrematuri + etaMadre + ipertensioneStoria +
    pesoMadre + irritUterinaMadre + etaMadre, family = binomial)
summary(fit)
```

```
##
## Call:
## glm(formula = pesoBimbo ~ nPartiPrematuri + etaMadre + ipertensioneStoria +
##     pesoMadre + irritUterinaMadre + etaMadre, family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6933  -0.7827  -0.6121   0.9314   2.2179
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.74560    1.09167   1.599  0.10982
## nPartiPrematuri1+      1.42123    0.44873   3.167  0.00154 **
## etaMadre              -0.05331    0.03555  -1.500  0.13372
## ipertensioneStoriasi   1.90580    0.71601   2.662  0.00778 **
## pesoMadre             -0.03168    0.01503  -2.108  0.03506 *
## irritUterinaMadresi    0.69193    0.45428   1.523  0.12773
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 205.15  on 183  degrees of freedom
## AIC: 217.15
##
## Number of Fisher Scoring iterations: 4
```

```
pseudorsquared = 1 - (fit$deviance/fit$null.deviance)
pseudorsquared
```

```
## [1] 0.1257862
```

```
glm.probs = predict(fit, type = "response")  #equivalente a fit$fitted.values
summary(glm.probs)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.03426 0.18712 0.26026 0.31217 0.38651 0.90303
```

```
glm.pred = ifelse(glm.probs > 0.5, "sottopeso", "sovrappeso")
table(glm.pred, pesoBimbo)
```

```
##             pesoBimbo
## glm.pred     sovrappeso sottopeso
##   sottopeso          11        21
##   sovrappeso        119        38
```

```
mean(glm.pred == pesoBimbo)
```

```
## [1] 0.7407407
```

Il modello fit ha le variabili nPartiPrematuri e ipertensioneStoria altamente significative e perdono di significatività le variabili etaMadre e irritUterinaMadre. Lo pseudo indice di determinazione è peggiorato a 0.12 rispetto al modello completo. La predizione in media mostra la stessa performance del modello completo. In base a queste due considerazioni è bene scegliere il modello completo che mostra un leggero miglioramento in termini di pseudo R^2.

```
detach()
```

Grafi non orientati

```
library(gRbase)
library(gRain)
library(gRim)
library(RBGL)  #is.triangulated
```

Il dataset presenta sia variabili fattore sia variabili continue. Al fine di ottenere un dataset con variabili discrete, dicotomizziamo rispetto alla mediana le variabili continue. La variabile del peso della madre e' stata convertita da libbre a chilogrammi per una maggiore interpretabilita'. Inoltre sono state fatte altre modifiche alle variabili fattore (soprattutto aggiunta di etichette).

```
data("birthwt", package = "MASS")

dataset <- with(birthwt, {
    pesoBimbo = factor(low, labels = c("sovraPeso", "sottoPeso"))
    etaMadre = factor(as.numeric(age > median(age)), labels = c("etaMinMedian",
        "etaMagMedian"))
    pesoMadre = factor(as.numeric(lwt/2.205 > median(lwt/2.205)),
        labels = c("pesoMinMedian", "pesoMagMedian"))
    etnia <- factor(race, labels = c("bianca", "nera", "altro"))
    fumaMadre = factor(smoke, labels = c("no", "si"))
    nPartiPrematuri = factor(ptl)
    # levels(nPartiPrematuri)[-(1)] <- '1+' # notare come
    # cambia la struttura del grafo aic backward
    ipertensioneStoria = factor(ht, labels = c("no", "si"))
    irritUterinaMadre = factor(ui, labels = c("no", "si"))
    nVisiteGine <- factor(ftv)
    levels(nVisiteGine)[-(1:2)] <- "2+"

    data.frame(pesoBimbo, etaMadre, pesoMadre, etnia, fumaMadre,
        nPartiPrematuri, ipertensioneStoria, irritUterinaMadre,
        nVisiteGine)
})

str(dataset)
```

```
## 'data.frame':    189 obs. of  9 variables:
##  $ pesoBimbo         : Factor w/ 2 levels "sovraPeso","sottoPeso": 1 1 1 1 1 1 1 1 1 1 ...
##  $ etaMadre          : Factor w/ 2 levels "etaMinMedian",..: 1 2 1 1 1 1 1 1 2 2 ...
##  $ pesoMadre         : Factor w/ 2 levels "pesoMinMedian",..: 2 2 1 1 1 2 1 1 2 1 ...
##  $ etnia             : Factor w/ 3 levels "bianca","nera",..: 2 3 1 1 1 3 1 3 1 1 ...
##  $ fumaMadre         : Factor w/ 2 levels "no","si": 1 1 2 2 2 1 1 1 2 2 ...
##  $ nPartiPrematuri   : Factor w/ 4 levels "0","1","2","3": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ipertensioneStoria: Factor w/ 2 levels "no","si": 1 1 1 1 1 1 1 1 1 1 ...
##  $ irritUterinaMadre : Factor w/ 2 levels "no","si": 2 1 1 2 2 1 1 1 1 1 ...
##  $ nVisiteGine       : Factor w/ 3 levels "0","1","2+": 1 3 2 3 1 1 2 2 2 1 ...
```

```
summary(dataset)
```

```
##      pesoBimbo           etaMadre           pesoMadre     etnia    fumaMadre
```

```
##  sovraPeso:130   etaMinMedian:107   pesoMinMedian:96   bianca:96   no:115
##  sottoPeso: 59   etaMagMedian: 82   pesoMagMedian:93   nera  :26   si: 74
##                                                       altro :67
##
##  nPartiPrematuri ipertensioneStoria irritUterinaMadre nVisiteGine
##  0:159           no:177             no:161            0 :100
##  1: 24           si: 12             si: 28            1 : 47
##  2:  5                                                2+: 42
##  3:  1
```

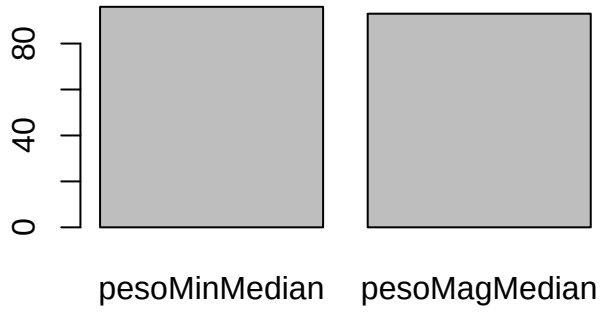
Ho notato subito un cambiamento dei modelli applicati anche per una piccola modifica al dataset come per esempio creare un livello specifico per una variabile. Con una piccola modifica si ottiene dei modelli non interpretabili con troppe dipendenze.

Viste le modifiche fatte andiamo a vedere alcuni plot delle variabili dicotomizzate.

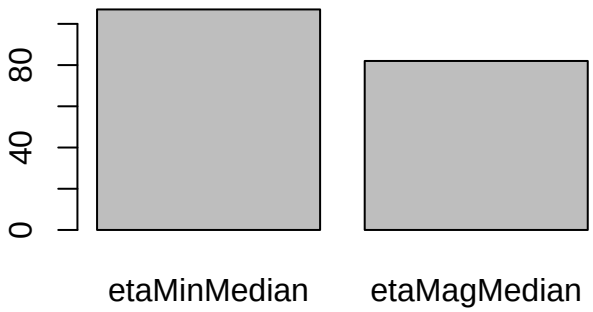
```
attach(dataset)
```

```
## I seguenti oggetti sono mascherati da dataset (pos = 10):
##
##     etaMadre, etnia, fumaMadre, ipertensioneStoria, irritUterinaMadre,
##     nPartiPrematuri, pesoMadre
```

```
plot(pesoMadre)
```

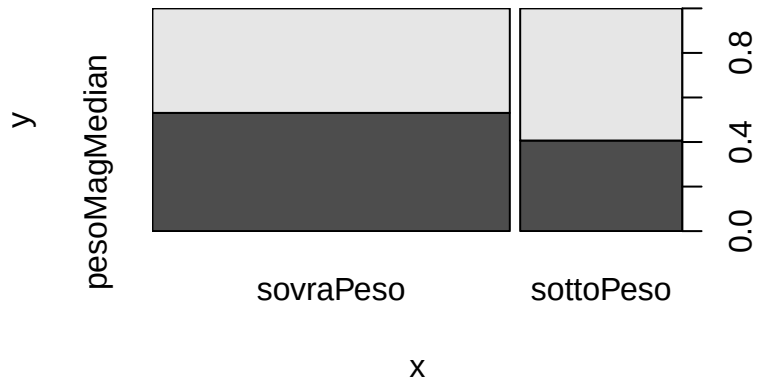



```
plot(etaMadre)
```

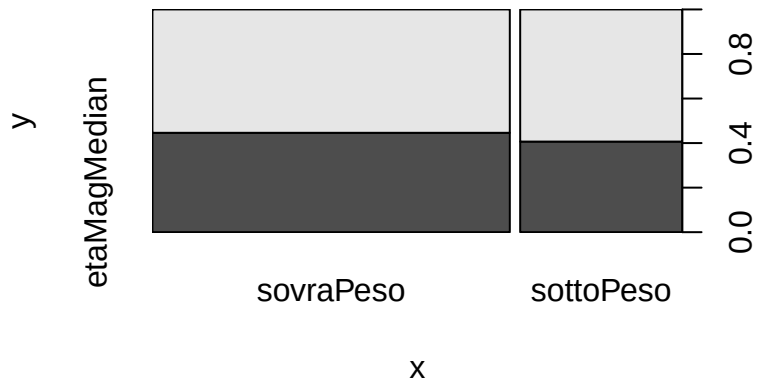


I grafici mostrano che le madri con peso al di sotto della mediana superano di poco quella al di sopra della mediana. Lo stesso vale in modo equivalente per l'età della madre.

```
plot(pesoBimbo, pesoMadre)
```




```
plot(pesoBimbo, etaMadre)  #--> mamme giovani
```




```
ftable(pesoBimbo ~ etaMadre + pesoMadre)  #(sub)tabella di contingenza per il modello log-lineare
```

```
##                            pesoBimbo sovraPeso sottoPeso
## etaMadre     pesoMadre
## etaMinMedian pesoMinMedian                37        22
##              pesoMagMedian                35        13
## etaMagMedian pesoMinMedian                24        13
##              pesoMagMedian                34        11
```

Inoltre notiamo che il fatto che i bambini siano sottopeso è preponderante per le madri con un peso al di sotto della mediana. A livello di età invece i bambini sottopeso superano di poco quelli sovrappeso quando l'età della madre è sotto la mediana. Questi fatti sono confermati dalla tabella delle frequenze (o tabella di contingenza parziale). Fare la tabella di contingenza per il modello log-lineare per un numero così alto di variabili e con così tanti livelli diventa non interpretabile (e inoltre il suo calcolo è oneroso in termini computazionali).

Prepariamoci ad una procedura stepwise. Creiamo il modello nullo composto da tutte indipendenze marginali (no archi tra i nodi) e il modello saturo (con nessuna indipendenza).

Modello nullo (tutte indipendenza marginali)

```
null_model = dmod(~.^1, data = dataset, fit = TRUE)
null_model
```

```
## Model: A dModel with 9 variables
##  -2logL    :        2213.79 mdim :   13 aic :      2239.79
##  ideviance :          -0.00 idf  :    0 bic :      2281.94
##  deviance  :         490.73 df   : 2290
```

```
formula(null_model)
```

```
## ~pesoBimbo + etaMadre + pesoMadre + etnia + fumaMadre + nPartiPrematuri +
##     ipertensioneStoria + irritUterinaMadre + nVisiteGine
```

```
plot(null_model)
```

Modello completo (con interazioni massimo fino al 2° ordine)

```
complete_model = dmod(~.^., data = dataset, interaction(2))
complete_model
```

```
## Model: A dModel with 9 variables
##  -2logL    :        1723.06 mdim : 2303 aic :      6329.06
##  ideviance :         490.73 idf  : 2290 bic :     13794.81
##  deviance  :           0.00 df   :    0
```

```
formula(complete_model)
```

```
## ~pesoBimbo * etaMadre * pesoMadre * etnia * fumaMadre * nPartiPrematuri *
##     ipertensioneStoria * irritUterinaMadre * nVisiteGine
```

```
is.triangulated(ug(formula(complete_model)))
```

```
## [1] TRUE
```

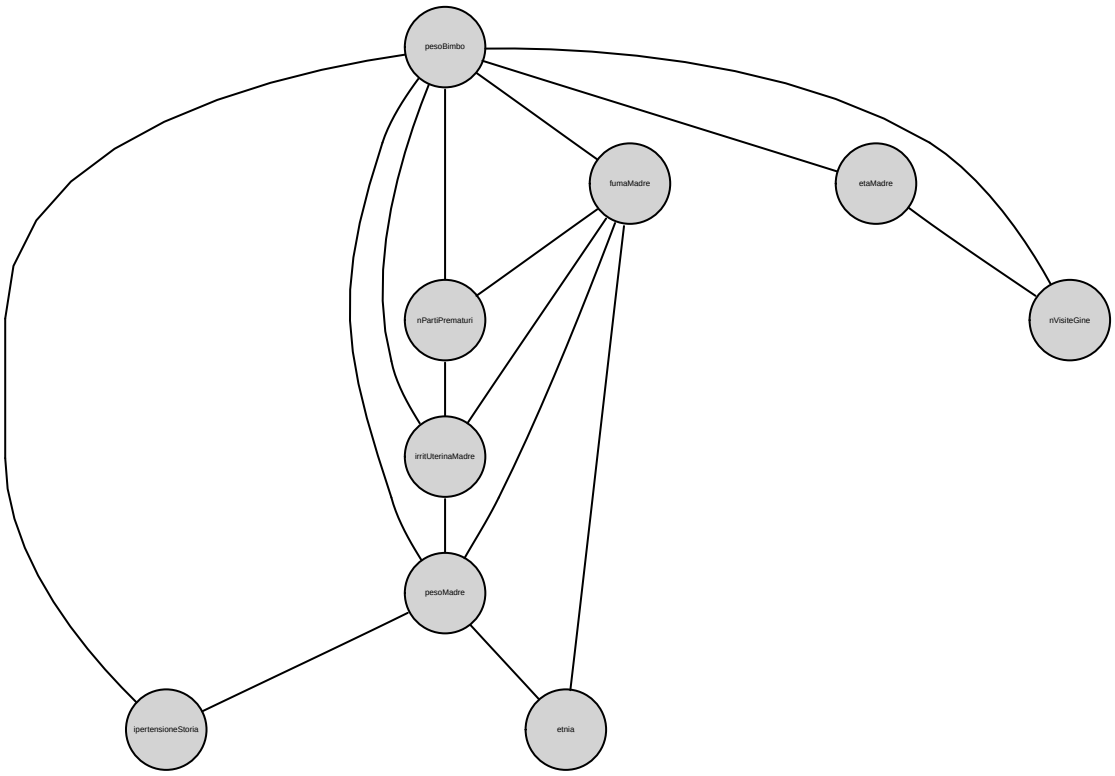
```
# plot(complete_model) #inutile vederlo
```

Il modello completo ha devianza 0 in quanto nel test del rapporto di verosimiglianza la devianza è data da $2(\hat{I}_s - \hat{I_m})$, dove $\hat{I}$ indica il valore della log-verosimiglianza nel punto di massimo, quindi il modello ridotto coincide con il modello saturo, ne consegue che tale differenza è nulla.

Indipendenza condizionele: X $\perp\!\!\!\perp$ Y | Z

X e Y sono condizionatamente indipendenti dato Z se e solo se, data la conoscenza che Z si verifichi, la conoscenza che X si verifichi non fornisce informazioni sulla probabilità che si verifichi Y (e viceversa tra Y e X)

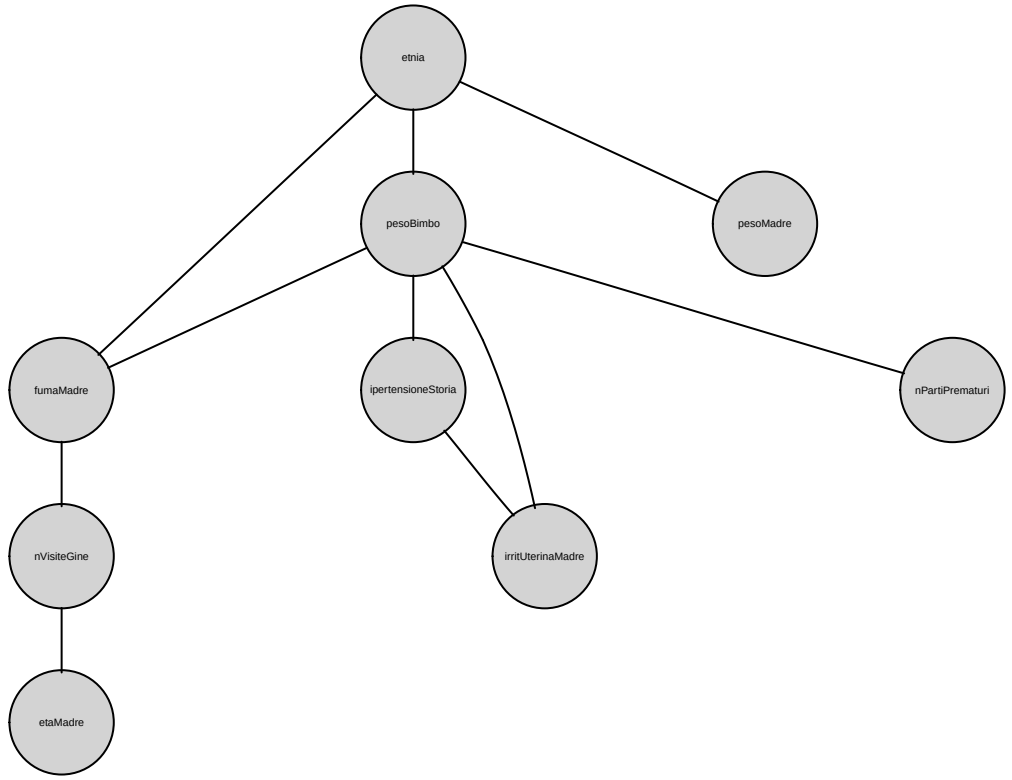
```
backward_aic = stepwise(complete_model)  #default: AIC
plot(backward_aic)
```



Questo modello ha troppe dipendenze. Il peso del bambino inoltre è influenzato dal numero di visite dal ginecologo, il che non è verosimile.

Un modello invece che si è dimostrato un ottimo compromesso tra parsimoniosità e numero di dipendenze è quello secondo il criterio AIC, direzione forward ma lo spazio di modelli ristretto solo ai grafi scomponibili.

```
forward_aic = stepwise(null_model, k = 2, type = "decomposable",
    direction = "forward")
plot(forward_aic)
```




```
formula(forward_aic)
```

```
## ~etnia * pesoBimbo * fumaMadre + ipertensioneStoria * irritUterinaMadre *
##     pesoBimbo + nPartiPrematuri * pesoBimbo + etnia * pesoMadre +
##     fumaMadre * nVisiteGine + etaMadre * nVisiteGine
```

Notiamo che il peso del bambino è dipendente dal numero di parti prematuri, dalla storia di ipertensione e dal fatto che la madre fuma e dall'irritabilità dell'utero.

Possiamo procedere a fare alcuni test di indipendenza condizionale (non tengono conto della scelta dei modelli ma solo del dataset).

Le variabili categoriali sono: pesoBimbo, etaMadre, pesoMadre, etnia, fumaMadre, ipertensioneStoria, irritUterinaMadre Le variabili ordinali sono: nVisiteGine, nPartiPrematuri

Facciamo uso della funzione ciTest(dataset,set = c("x","y","z")) che verifica che x cond. ind y | z .Se i p-value sono alti si rifiuta l'ipotesi di indipendenza condizionata altrimenti si accetta.

Per variabili ordinali usiamo il Jonckheere-Terpstra test Per variabili ordinali e categoriali il Kruskal test. Per variabili categoriali usiamo il test della devianza.

E' possibile usare anche la funzione ciTest() senza distinguere tra variabili categoriali e ordinali.

```
ciTest(dataset, set = c("pesoBimbo", "irritUterinaMadre", "pesoMadre"))
```

```
## Testing pesoBimbo _|_ irritUterinaMadre | pesoMadre
## Statistic (DEV):    7.785 df: 2 p-value: 0.0204 method: CHISQ
```

```
## Slice information:
##   statistic  p.value df     pesoMadre
## 1    0.3214 0.570760  1 pesoMinMedian
## 2    7.4636 0.006296  1 pesoMagMedian
```

Questo test sottolinea che il peso del bambino e irritabilità dell'utero sono indipendenti per le madri che hanno peso maggiore della mediana. Ciò non accade per le madri con peso al di sotto della mediana.

```
ciTest(dataset, set = c("pesoBimbo", "etnia", "etaMadre"))
```

```
## Testing pesoBimbo _|_ etnia | etaMadre
## Statistic (DEV):    5.572 df: 4 p-value: 0.2335 method: CHISQ
## Slice information:
##   statistic p.value df     etaMadre
## 1     1.292  0.5243  2 etaMinMedian
## 2     4.280  0.1176  2 etaMagMedian
```

In questo caso non c'è l'indipendenza condizionata tra il peso del bambino e l'etnia data l'età della madre.

```
ciTest_ordinal(dataset, set = c("nPartiPrematuri", "nVisiteGine",
    "etnia"), "jt", N = 1000)
```

```
## $JT
## [1] 901
##
## $EJT
## [1] 979.5
##
## $P
## [1] 0.4536544
##
## $montecarlo.P
## [1] 0.237
##
## $set
## [1] "nPartiPrematuri" "nVisiteGine"     "etnia"
```

Questo Jonckheere-Terpstra test ci informa che non c'è l'indipendenza condizionata tra le variabili ordinali.

```
ciTest_ordinal(dataset, set = c("etnia", "nVisiteGine", "etaMadre"),
    "kruskal", N = 1000)
```

```
## $KW
## [1] 11.15381
##
## $df
## [1] 4
##
## $P
## [1] 0.02488877
##
```

```
## $montecarlo.P
## [1] 0.021
##
## $set
## [1] "etnia"       "nVisiteGine" "etaMadre"
```

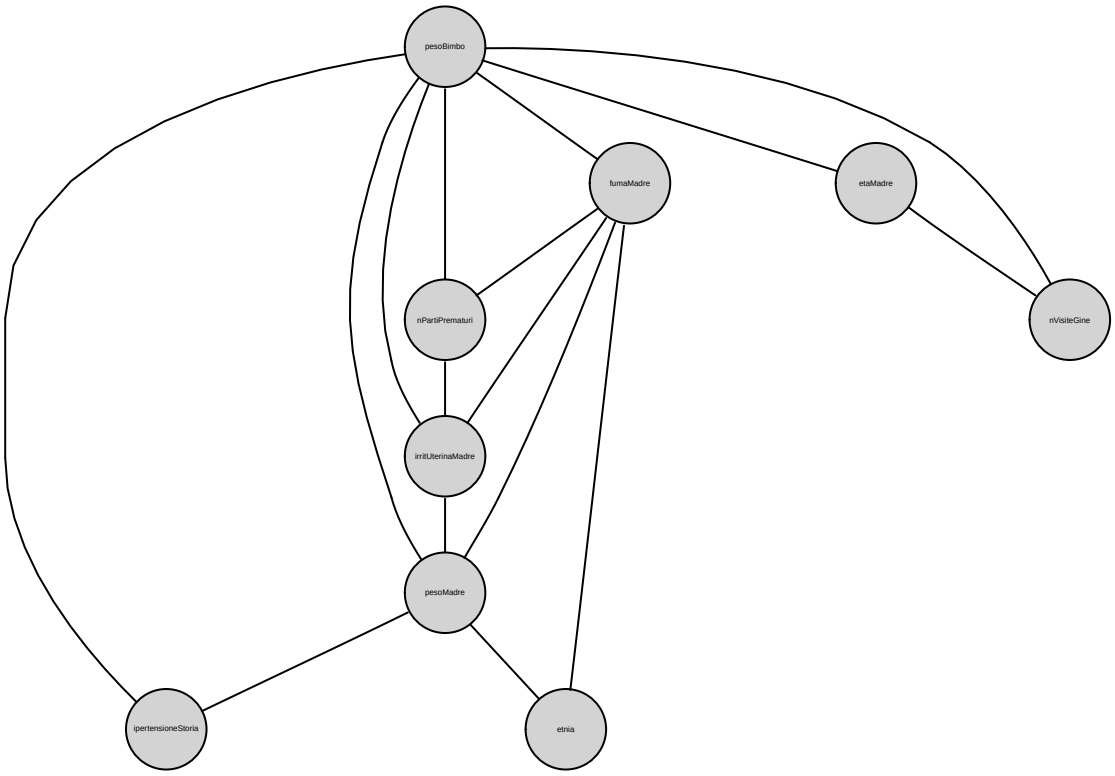
```
ciTest_ordinal(dataset, set = c("pesoBimbo", "etaMadre", "nVisiteGine"),
    "kruskal", N = 1000)
```

```
## $KW
## [1] 12.71942
##
## $df
## [1] 3
##
## $P
## [1] 0.005284383
##
## $montecarlo.P
## [1] 0.006
##
## $set
## [1] "pesoBimbo"   "etaMadre"    "nVisiteGine"
```

I test di Kruscal invece dimostrano l'accettazione dell'indipendenza condizionata.

Criterio BIC (Procedura backward e forward) solo per grafi scomponibili (in modo da ottenere delle stime di massima verosimiglianza in forma chiusa e senza usare una procedura iterativa)

```
backward_bic = stepwise(complete_model, k = log(length(dataset)))
plot(backward_bic)
```




```
formula(backward_bic)
```

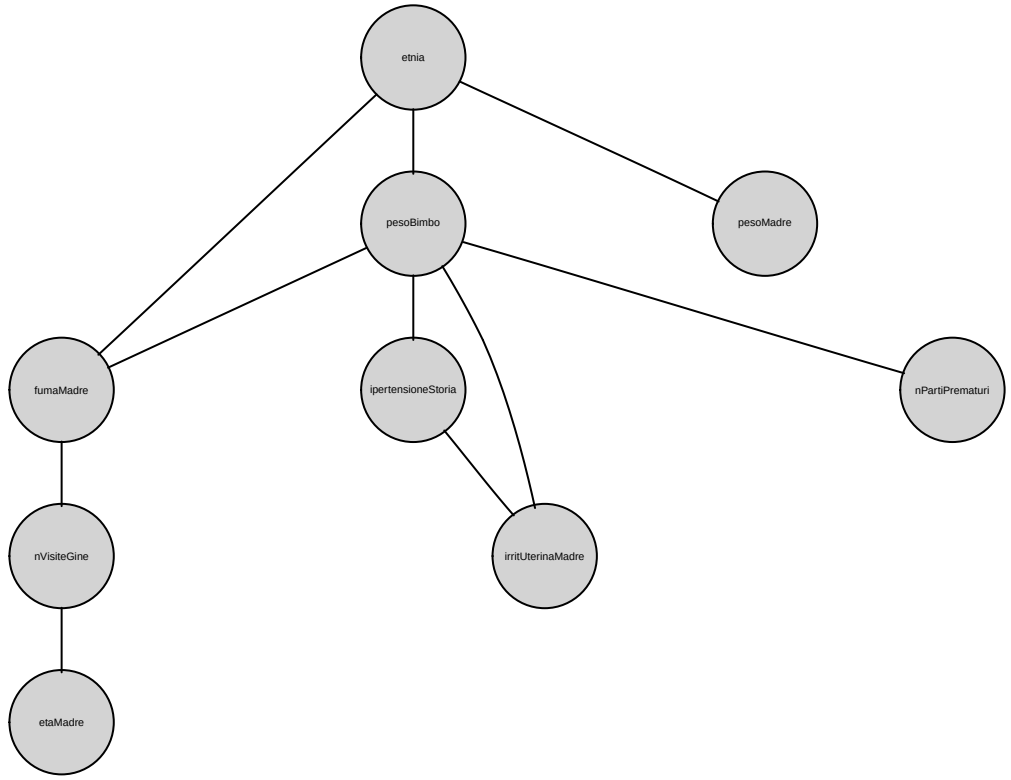
```
## ~pesoBimbo * fumaMadre * nPartiPrematuri * irritUterinaMadre +
##     pesoBimbo * pesoMadre * fumaMadre * irritUterinaMadre + pesoBimbo *
##     etaMadre * nVisiteGine + pesoBimbo * pesoMadre * ipertensioneStoria +
##     pesoMadre * etnia * fumaMadre
```

```
rip(ug(formula(backward_bic)))
```

```
## cliques
##   1 : pesoBimbo fumaMadre irritUterinaMadre nPartiPrematuri
##   2 : pesoBimbo fumaMadre irritUterinaMadre pesoMadre
##   3 : pesoBimbo ipertensioneStoria pesoMadre
##   4 : etnia pesoMadre fumaMadre
##   5 : pesoBimbo etaMadre nVisiteGine
## separators
##   1 :
##   2 : pesoBimbo fumaMadre irritUterinaMadre
##   3 : pesoBimbo pesoMadre
##   4 : pesoMadre fumaMadre
##   5 : pesoBimbo
## parents
##   1 : 0
##   2 : 1
##   3 : 2
```

```
##   4 : 2
##   5 : 3
```

```
forward_bic = stepwise(null_model, k = log(length(dataset)),
    type = "decomposable", direction = "forward")
plot(forward_bic)
```




```
formula(forward_bic)
```

```
## ~etnia * pesoBimbo * fumaMadre + ipertensioneStoria * irritUterinaMadre *
##     pesoBimbo + nPartiPrematuri * pesoBimbo + etnia * pesoMadre +
##     fumaMadre * nVisiteGine + etaMadre * nVisiteGine
```

Quello che accade è speculare al criterio AIC. Il modello completo usando il BIC preserva fin troppe dipendenze (sono proprio identici). Il modello con direzione forward e ristretto allo spazio dei modelli scomponibili è il medesimo trovato con criterio AIC. La scelta che spiega delle dipendenze logiche ricade sul modello più parsimonioso.

```
detach()
```

DAG

```
library(gRbase)
library(gRain)
library(gRim)
library(bnlearn)  #hc
library(igraph)
library(ggm)  #dSep
```

Al fine di dare un'ordinamento alle variabili facciamo uso di grafi orientati aciclici (DAG) per creare un modello di indipendenza (condizionata). Più nello specifico usiamo il modello grafico per variabili discrete basato sui DAG detto Bayesian Network. Prima di tutto rendiamo tutte le variabili del dataset delle variabili fattore. Le variabili continue sono state dicotomizzate rispetto alla mediana che rappresenta un indice robusto e che non risente dei valori estremi. Inoltre per alcune variabili sono stati creati dei livelli laddove i valori del dataset erano preponderanti e anche per una maggiore interpretabilità.

```
data("birthwt", package = "MASS")

dataset = with(birthwt, {
    pesoBimbo = factor(low, labels = c("sovraPeso", "sottoPeso"))
    etaMadre = factor(as.numeric(age > median(age)), labels = c("etaMinMedian",
        "etaMagMedian"))
    pesoMadre = factor(as.numeric(lwt/2.205 > median(lwt/2.205)),
        labels = c("pesoMinMedian", "pesoMagMedian"))
    etnia <- factor(race, labels = c("bianca", "nera", "altro"))
    fumaMadre = factor(smoke, labels = c("no", "si"))
    nPartiPrematuri = factor(ptl)
    levels(nPartiPrematuri)[-(1)] <- "1+"
    # levels(nPartiPrematuri)[-(1:2)] <- '2+'
    ipertensioneStoria = factor(ht, labels = c("no", "si"))
    irritUterinaMadre = factor(ui, labels = c("no", "si"))
    nVisiteGine <- factor(ftv)
    levels(nVisiteGine)[-(1:2)] <- "2+"

    data.frame(pesoBimbo, etaMadre, pesoMadre, etnia, fumaMadre,
        nPartiPrematuri, ipertensioneStoria, irritUterinaMadre,
        nVisiteGine)
})
str(dataset)
```

```
## 'data.frame':    189 obs. of  9 variables:
##  $ pesoBimbo         : Factor w/ 2 levels "sovraPeso","sottoPeso": 1 1 1 1 1 1 1 1 1 1 ...
##  $ etaMadre          : Factor w/ 2 levels "etaMinMedian",..: 1 2 1 1 1 1 1 1 2 2 ...
##  $ pesoMadre         : Factor w/ 2 levels "pesoMinMedian",..: 2 2 1 1 1 2 1 1 2 1 ...
##  $ etnia             : Factor w/ 3 levels "bianca","nera",..: 2 3 1 1 1 3 1 3 1 1 ...
##  $ fumaMadre         : Factor w/ 2 levels "no","si": 1 1 2 2 2 1 1 1 2 2 ...
##  $ nPartiPrematuri   : Factor w/ 2 levels "0","1+": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ipertensioneStoria: Factor w/ 2 levels "no","si": 1 1 1 1 1 1 1 1 1 1 ...
##  $ irritUterinaMadre : Factor w/ 2 levels "no","si": 2 1 1 2 2 1 1 1 1 1 ...
##  $ nVisiteGine       : Factor w/ 3 levels "0","1","2+": 1 3 2 3 1 1 2 2 2 1 ...
```

```
summary(dataset)
```

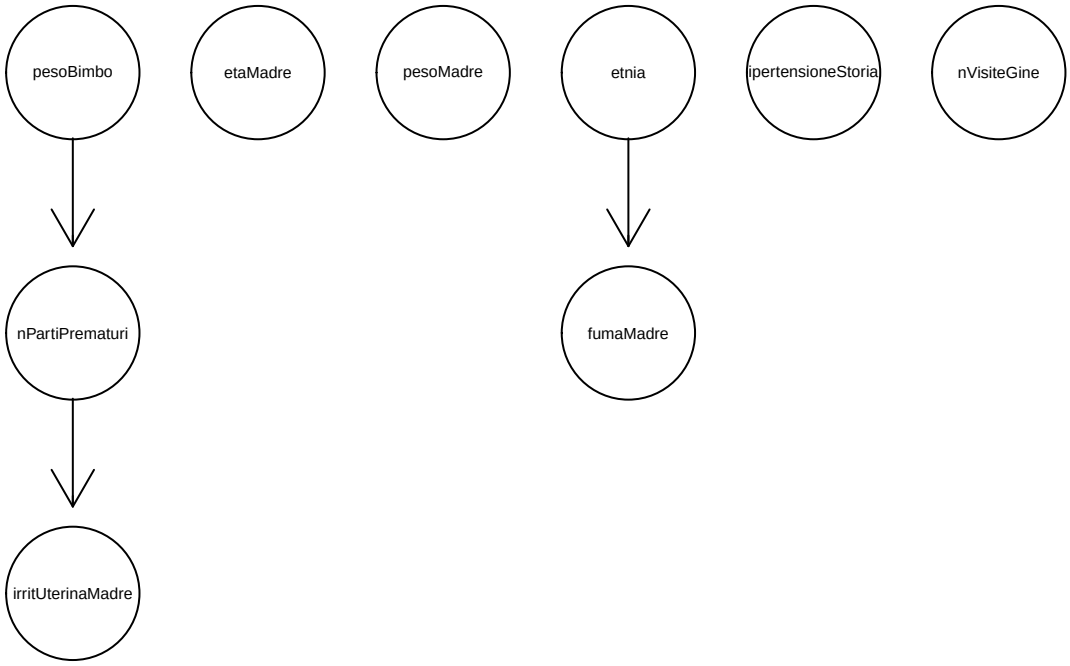
```
##      pesoBimbo           etaMadre          pesoMadre     etnia    fumaMadre
##  sovraPeso:130   etaMinMedian:107   pesoMinMedian:96   bianca:96   no:115
##  sottoPeso: 59   etaMagMedian: 82   pesoMagMedian:93   nera  :26   si: 74
##                                                        altro :67
##  nPartiPrematuri ipertensioneStoria irritUterinaMadre nVisiteGine
##  0 :159          no:177             no:161            0 :100
##  1+: 30          si: 12             si: 28            1 : 47
##                                                       2+: 42
```

```
attach(dataset)
```

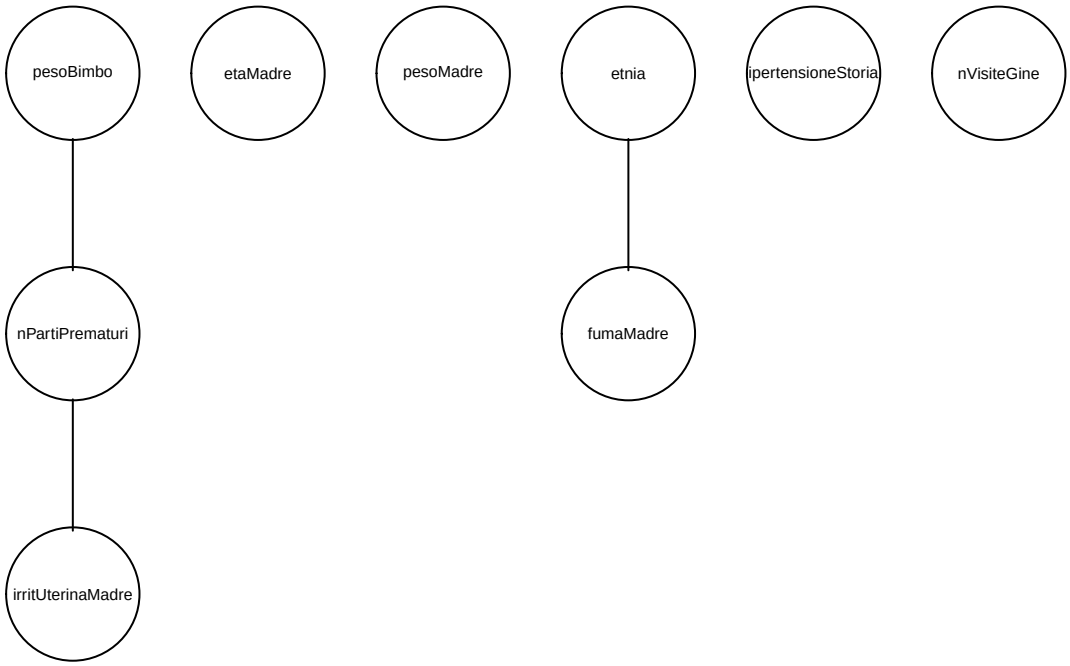
```
## I seguenti oggetti sono mascherati da dataset (pos = 13):
##
##     etaMadre, etnia, fumaMadre, ipertensioneStoria, irritUterinaMadre,
##     nPartiPrematuri, pesoMadre
```

Secondo le modifiche introdotte, cerchiamo di apprendere la struttura della Bayesian Network dal dataset. A tal proposito usiamo la funziona **hc()** dalla libreria bnlearn. Tale funzione è basata sull'algoritmo hill-climbing. È un algoritmo iterativo che inizia con una soluzione arbitraria a un problema e tenta di trovare una soluzione migliore apportando una modifica incrementale alla soluzione. Se la modifica produce una soluzione migliore, viene apportata un'altra modifica incrementale alla nuova soluzione e così via fino a quando non è possibile trovare ulteriori miglioramenti. Nella sua versione di default, viene minimizzato il BIC.

```
model = hc(dataset)
dag = as(amat(model), "graphNEL")
plot(dag)
```



```
dag_moralise = moralize(dag)
plot(dag_moralise)
```



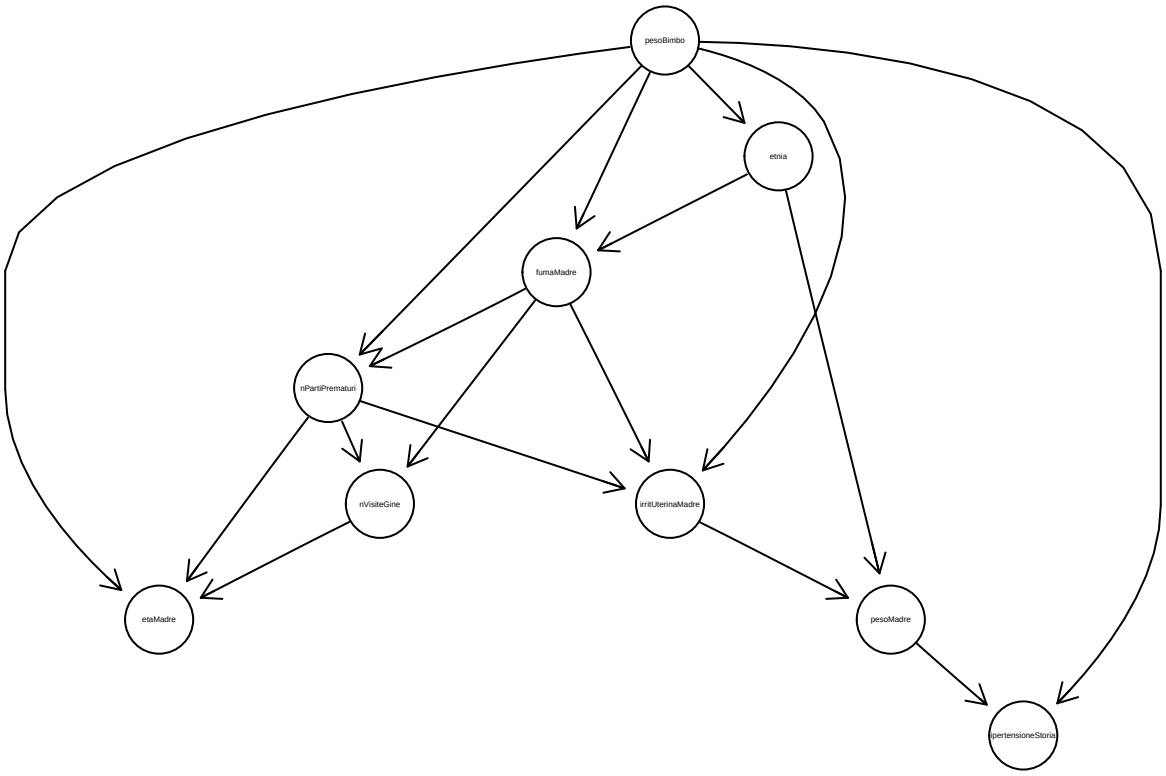
Il modello risultante non è informativo, è estremamente parsimoniovo e presenta diversi problemi. Prima di tutto sembra che il peso del bambino influenza il numero di parti prematuri della madre ma ciò non è possibile essendo quest'ultimo un antecedente logico. Inoltre molte delle variabili che sono candidate ad essere fattori di rischio per il basso peso del bambino non sono dipendenti con la variabile obiettivo. Una delle poche, se non l'unica, indipendenza marginale che ha un senso è quella legata al numero di visite dal ginecologo.

A questo punto abbiamo due strade:

- creare una struttura del DAG secondo le informazioni logiche a nostra conoscenza
- cambiare criterio: provare l'approccio AIC.

La strada scelta è ibrida: proviamo il criterio AIC, suddividiamo le variabili in categorie e introduciamo l'ordinamento tra le variabili. Adatto il modello con strategia AIC

```
model = hc(dataset, score = "aic")
dag = as(amat(model), "graphNEL")
plot(dag)
```



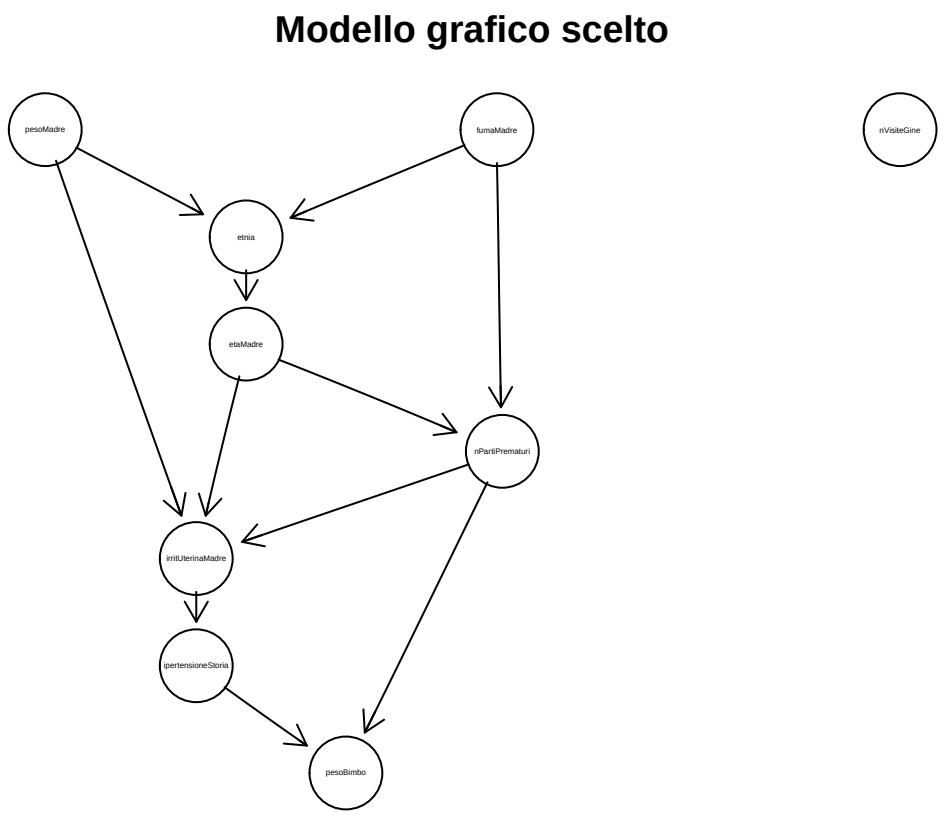
Questo DAG ci fornisce evidenza che il peso del bambino influenza le altri variabili esplicative (etnia, madre fumatrice, il numero di parti prematuri, ipertensione, età della madre). E' il momento giusto di suddividere le variabili nelle seguenti categorie:

1. pesoBimbo –> objective
2. etaMadre –> background
3. pesoMadre –> background
4. etnia –> background
5. fumaMadre –> background
6. nPartiPrematuri –> previous fact
7. ipertensioneStoria –> previous fact
8. irritUterinaMadre –> previous fact
9. nVsiteGine –> background

Infine escludiamo dal grafico delle dipendenze non logiche facendo affidamento sull'ordinamento delle variabili.

```
block <- c(2, 1, 1, 1, 1, 3, 3, 3, 1)  # 1=background 2=objective 3=previous fact
blM <- matrix(0, nrow = 9, ncol = 9)
rownames(blM) = names(dataset)
colnames(blM) = names(dataset)

blM[block == 2, block == 1] = 1  #objective non influenza background
blM[block == 3, block == 1] = 1  #previous fact non influenza background
blM[block == 2, block == 3] = 1  #objective non influenza previous fact
for (i in 1:9) {
```

```
    # nVisiteGine non influenza e non viene influenzato da
    # tutto il resto
    blM[9, i] = 1
    blM[i, 9] = 1
}

blackL <- data.frame(get.edgelist(as(blM, "igraph")))
names(blackL) <- c("from", "to")
fit <- hc(dataset, blacklist = blackL, score = "aic")
dag = as(amat(fit), "graphNEL")
plot(dag, main = "Modello grafico scelto")
```

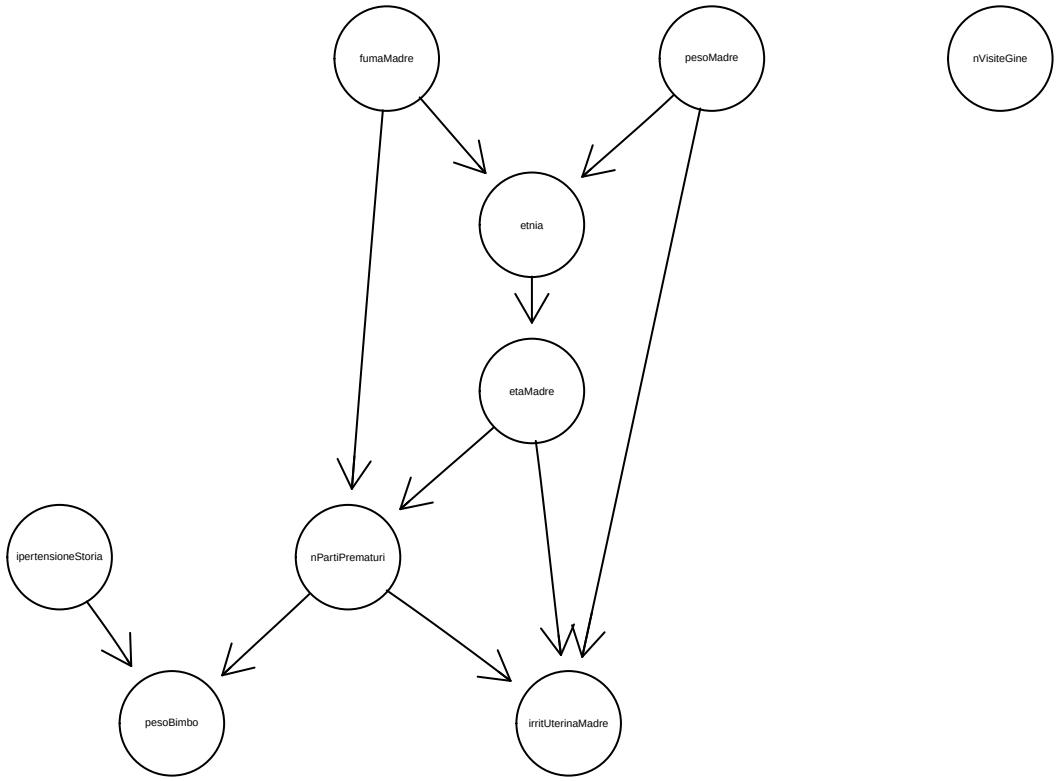


Il modello è piuttosto chiaro: in alto possiamo vedere le variabili di background e le loro dipendenza, in basso la variabile obiettivo e al centro abbiamo le varie dipendenze tra le variabile che verosimilmente sono i fattori di rischio per la nascita di un bambino sottopeso. Notiamo che il numero di parti prematuri della madre (precedenti a parto corrente) influenza direttamente la variabile obiettivo mentre la irritabilità influenza la storia di ipertensione che a sua volta influenza il peso del bambino.

Vorrei fare notare però che la variabile ipertensioneStoria non ci dice se la madre ha o meno l'ipertensione ma solo se è predisposta ad averla. Dunque anche se c'è la presenza della storia di ipertensione non è detto che la madre soffra di ipertensione (pressione alta del sangue). Andiamo ad indagare meglio su tale fatto aggiungendo l'informazioni che la storia di ipertensione non può essere influenza da tutto il resto delle variabili.

```
block <- c(2, 1, 1, 1, 1, 3, 3, 3, 1)
blM <- matrix(0, nrow = 9, ncol = 9)
```

```
rownames(blM) = names(dataset)
colnames(blM) = names(dataset)

blM[block == 2, block == 1] = 1  #objective non influenza background
blM[block == 3, block == 1] = 1  #previous fact non influenza background
blM[block == 2, block == 3] = 1  #objective non influenza i previous fact
for (i in 1:9) {
    blM[9, i] = 1  #numero di visite dal ginecologo non influenza tutto il resto
    blM[i, 9] = 1  #numero di visite dal ginacologo non viene influenzato da tutto il resto
    blM[i, 7] = 1  ### nessuna variabile possa influenzare la storia di ipertensione
}

blackL <- data.frame(get.edgelist(as(blM, "igraph")))
names(blackL) <- c("from", "to")
model <- hc(dataset, blacklist = blackL, score = "aic")
plot(as(amat(model), "graphNEL"))
```



Notiamo adesso che l'ipertensione influenza direttamente il peso del bambino ma l'irritabilità dell'utero non più. Dunque è bene considerare il modello precedente per il motivo che qual'ora la storia dell'ipertensione sia presente si tenga conto che anche la madre con molta probabilità soffrirà di tale patologia. Dunque andiamo avanti con il modello denominato **fit** e creiamo il network per la propagazione usando le funzioni **grain()** e propaghiamo la probabilità condizionata derivante dai dati, secondo la struttura della rete e secondo le assunzioni di indipendenza condizionata mediante la funzione **compile()**. Per finire, al fine di ottenere delle probabilità positive settiamo la variabile smooth per aggiungere una quantità infinitesimale che consente di evitare conteggi pari a zero e svolgere computazionalmente il calcolo delle probabilità.

```
fit_propagate = grain(dag, data = dataset)
fit_propagate = compile(fit_propagate, propagate = TRUE, smooth = 0.1)
summary(fit_propagate)
```

```
## Independence network: Compiled: TRUE Propagated: TRUE
##  Nodes : Named chr [1:9] "pesoBimbo" "etaMadre" "pesoMadre" "etnia" "fumaMadre" ...
##  - attr(*, "names")= chr [1:9] "pesoBimbo" "etaMadre" "pesoMadre" "etnia" ...
##  Number of cliques:                 6
##  Maximal clique size:               4
##  Maximal state space in cliques:   24
```

Cerchiamo adesso di ottenere le probabilità condizionata,marginali,congiunte facendo uso della funzione **querygrain()**. Studiamo prima di tutto la parte che riguarda le variabili obiettivo e i fattori di rischio più plausibili. I risultati delle query non vengono da un'inferenza ma semplicemente prima stimo il modello e poi chiedo le probabilità di interesse (sulla base della propagazione del network).

```
querygrain(fit_propagate, nodes = c("pesoBimbo"), type = "marginal")
```

```
## $pesoBimbo
## pesoBimbo
## sovraPeso sottoPeso
## 0.6875156 0.3124844
```

Questa è la probabilità della variabile pesoBimbo di tipo marginale. Ci evidenzia che il 31% dei bambini è sotto peso e il 69 % sovrappeso. Questo è un dato che si poteva benissimo notare applicando la legge principe della probabilità (casi favorevoli su casi possibili).

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "ipertensioneStoria"),
    type = "joint")
```

```
##            ipertensioneStoria
## pesoBimbo          no         si
##   sovraPeso 0.6600232 0.02749238
##   sottoPeso 0.2764615 0.03602292
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "nPartiPrematuri"),
    type = "joint")
```

```
##            nPartiPrematuri
## pesoBimbo           0         1+
##   sovraPeso 0.6222890 0.06522660
##   sottoPeso 0.2169851 0.09549933
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "irritUterinaMadre"),
    type = "joint")
```

```
##            irritUterinaMadre
## pesoBimbo          no         si
##   sovraPeso 0.5906602 0.09685534
##   sottoPeso 0.2615034 0.05098104
```

Per quanto riguarda le probabilità congiunte queste ci dicono che la probabilità che il bambino sia sottopeso per madri che hanno una storia di ipertensione è del 3.6 %, e di 27% se non hanno la storia di ipertensione. Inoltre per madri che hanno avuto parti prematuri la probabilità di avere il bambino sottopeso è del 21%, e per un numero di parti prematuri superiore a 1 è del 9.5%. Per finire la probabilità di avere un bambino sottopeso e contemporaneamente la madre soffre di irritabilità uterina è del 5.1%.

Andiamo ad indagare meglio su un insieme di evidenze calcolando probabilità condizionate.

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "ipertensioneStoria"),
    type = "conditional")
```

```
##            ipertensioneStoria
## pesoBimbo         no        si
##   sovraPeso 0.704788 0.4328466
##   sottoPeso 0.295212 0.5671534
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "irritUterinaMadre"),
    type = "conditional")
```

```
##            irritUterinaMadre
## pesoBimbo          no        si
##   sovraPeso 0.6931301 0.6551523
##   sottoPeso 0.3068699 0.3448477
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "nPartiPrematuri"),
    type = "conditional")
```

```
##            nPartiPrematuri
## pesoBimbo           0       1+
##   sovraPeso 0.741461 0.405825
##   sottoPeso 0.258539 0.594175
```

E' interessante notare come la probabilità che il bimbo sia sottopeso sapendo che la madre ha una storia l'ipertensione è del 56% e del 29% se non ha l'ipertensione. Per quanto riguarda l'irritabilità dell'utero vi è una differenza del 4% tra il caso in cui la madre ne soffra o meno. Infine se la madre ha avuto più di un parto prematuro la probabilità che il bambino sia sottopeso è del 59% e se non ha avuto parti prematuri del 26%.

```
querygrain(fit_propagate, nodes = c("pesoBimbo", c("nPartiPrematuri",
    "fumaMadre")), type = "conditional")
```

```
## , , fumaMadre = no
##
##            nPartiPrematuri
## pesoBimbo           0        1+
##   sovraPeso 0.7414627 0.4056914
##   sottoPeso 0.2585373 0.5943086
##
## , , fumaMadre = si
##
##            nPartiPrematuri
## pesoBimbo           0        1+
##   sovraPeso 0.7414577 0.4059073
##   sottoPeso 0.2585423 0.5940927
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", c("irritUterinaMadre",
    "fumaMadre")), type = "conditional")
```

```
## , , fumaMadre = no
##
##            irritUterinaMadre
## pesoBimbo          no        si
##   sovraPeso 0.7104773 0.6898218
##   sottoPeso 0.2895227 0.3101782
##
## , , fumaMadre = si
##
##            irritUterinaMadre
## pesoBimbo         no        si
##   sovraPeso 0.665319 0.6099633
##   sottoPeso 0.334681 0.3900367
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "ipertensioneStoria",
    "fumaMadre"), type = "conditional")
```

```
## , , fumaMadre = no
##
##            ipertensioneStoria
## pesoBimbo          no        si
##   sovraPeso 0.7247767 0.4581709
##   sottoPeso 0.2752233 0.5418291
##
## , , fumaMadre = si
##
##            ipertensioneStoria
## pesoBimbo          no        si
##   sovraPeso 0.6737897 0.3922468
##   sottoPeso 0.3262103 0.6077532
```

La prima query sottolinea come il fatto che la madre fumi/non fumi non influisce sul peso del bambino per ogni numero di parti prematuri (anche zero). La seconda invece ci dice che l'effetto di essere fumatrice dato che hai irritabilità uterina aumenta la probabilità che il bambino sia sottopeso dal 31% (non fumatrice) al 39% (fumatrice). Infine per quanto riguarda l'ipertensione abbiamo un aumento del 6% nel caso in cui la madre sia fumatrice. Studiamo adesso le probabilità condizionandoci all'età della madre (ricordando che è stata dicotomizzata rispetto alla mediana)

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "ipertensioneStoria",
    "etaMadre"), type = "conditional")
```

```
## , , etaMadre = etaMinMedian
##
##            ipertensioneStoria
## pesoBimbo         no        si
##   sovraPeso 0.722052 0.4698905
##   sottoPeso 0.277948 0.5301095
##
```

```
## , , etaMadre = etaMagMedian
##
##            ipertensioneStoria
## pesoBimbo          no       si
##   sovraPeso 0.6821146 0.387986
##   sottoPeso 0.3178854 0.612014
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "irritUterinaMadre",
    "etaMadre"), type = "conditional")
```

```
## , , etaMadre = etaMinMedian
##
##            irritUterinaMadre
## pesoBimbo          no        si
##   sovraPeso 0.7185053 0.6506736
##   sottoPeso 0.2814947 0.3493264
##
## , , etaMadre = etaMagMedian
##
##            irritUterinaMadre
## pesoBimbo          no        si
##   sovraPeso 0.6624003 0.6644306
##   sottoPeso 0.3375997 0.3355694
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "nPartiPrematuri",
    "etaMadre"), type = "conditional")
```

```
## , , etaMadre = etaMinMedian
##
##            nPartiPrematuri
## pesoBimbo           0       1+
##   sovraPeso 0.7416871 0.413773
##   sottoPeso 0.2583129 0.586227
##
## , , etaMadre = etaMagMedian
##
##            nPartiPrematuri
## pesoBimbo           0        1+
##   sovraPeso 0.7411181 0.4010056
##   sottoPeso 0.2588819 0.5989944
```

Con la presenza dell'ipertensione, abbiamo che la probabilità che il bambino sia sottopeso è del 61% per età della madre al di sopra della mediana e del 53% al di sotto. In assenza di ipertensione e madri giovani siamo sul 28% di probabilità e per madri più anziane al 32%. Per quanto riguarda l'irritabilità uterina appare più grave essere madri giovani per avere un bimbo sottopeso (ma di molto poco). Qualora la madre non soffra questa patologià abbiamo un 33% di probabilità che il bimbo sia sottopeso per madri non giovani. Il numero di parti prematuri non sembra variare in funzione dell'età della madre.

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "ipertensioneStoria",
    "pesoMadre"), type = "conditional")
```

```
## , , pesoMadre = pesoMinMedian
```

```
##
##            ipertensioneStoria
## pesoBimbo          no        si
##   sovraPeso 0.7052244 0.4283077
##   sottoPeso 0.2947756 0.5716923
##
## , , pesoMadre = pesoMagMedian
##
##            ipertensioneStoria
## pesoBimbo          no        si
##   sovraPeso 0.7043342 0.4370498
##   sottoPeso 0.2956658 0.5629502
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "irritUterinaMadre",
    "pesoMadre"), type = "conditional")
```

```
## , , pesoMadre = pesoMinMedian
##
##            irritUterinaMadre
## pesoBimbo          no        si
##   sovraPeso 0.6900209 0.6825461
##   sottoPeso 0.3099791 0.3174539
##
## , , pesoMadre = pesoMagMedian
##
##            irritUterinaMadre
## pesoBimbo          no       si
##   sovraPeso 0.6960093 0.600932
##   sottoPeso 0.3039907 0.399068
```

```
querygrain(fit_propagate, nodes = c("pesoBimbo", "nPartiPrematuri",
    "pesoMadre"), type = "conditional")
```

```
## , , pesoMadre = pesoMinMedian
##
##            nPartiPrematuri
## pesoBimbo           0        1+
##   sovraPeso 0.7425447 0.4054984
##   sottoPeso 0.2574553 0.5945016
##
## , , pesoMadre = pesoMagMedian
##
##            nPartiPrematuri
## pesoBimbo           0        1+
##   sovraPeso 0.7403406 0.4061595
##   sottoPeso 0.2596594 0.5938405
```

Per quanto riguarda il peso della madre minore della mediana, la probabilità di avere un bimbo sottopeso con la presenza di una storia di ipertensione è del 57% e del 56% per i pesi maggiori della mediana. Non sembra influire molto. Notiamo invece un aumento del 9 % tra mamme con peso sotto la mediana e mamme con peso sopra nel caso questa soffra di irritabilità dell'utero.Mentre il numero di parti prematuri non ha alcuna influenza.

Concludendo i 3 candidati:

- storia dell'ipertensione
- irritabilità uterina
- numero di parti prematuri

si sono dimostrati essere a tutti gli effetti dei fattori di rischio importanti per determinare su un bambino nasce sottopeso.

Poichè i DAG sono sequenza di regressioni logistiche possiamo procedere a stimare i parametri della regressione.

```
out.pesoBimbo = glm(pesoBimbo ~ ipertensioneStoria + nPartiPrematuri,
    family = binomial)
summary(out.pesoBimbo)
```

```
##
## Call:
## glm(formula = pesoBimbo ~ ipertensioneStoria + nPartiPrematuri,
##     family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3230  -0.7398  -0.7398   1.0385   1.6909
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -1.1560     0.1911  -6.048 1.47e-09 ***
## ipertensioneStoriasi   1.2879     0.6269   2.054 0.039940 *
## nPartiPrematuri1+      1.4919     0.4193   3.558 0.000373 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 217.66  on 186  degrees of freedom
## AIC: 223.66
##
## Number of Fisher Scoring iterations: 4
```

```
out.pesoBimbo = glm(pesoBimbo ~ irritUterinaMadre + nPartiPrematuri,
    family = binomial)
summary(out.pesoBimbo)
```

```
##
## Call:
## glm(formula = pesoBimbo ~ irritUterinaMadre + nPartiPrematuri,
##     family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5918  -0.7391  -0.7391   1.0946   1.6918
##
## Coefficients:
```

```
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.1580     0.1942  -5.962  2.5e-09 ***
## irritUterinaMadresi   0.7383     0.4382   1.685  0.09206 .
## nPartiPrematuri1+     1.3558     0.4219   3.214  0.00131 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 234.67  on 188  degrees of freedom
## Residual deviance: 219.12  on 186  degrees of freedom
## AIC: 225.12
##
## Number of Fisher Scoring iterations: 4
```

Queste stime ci evidenziano che il numero di parti prematuri è forse una delle cause più importanti della nascita di un bimbo sottopeso e questo viene sottolineato sia dalla dipendenza diretta nel modello usato e sia dal fatto che le stime prevedono che la variabile nPartiPrematuri sia altamente significativa nel caso specifico di uno o più parti prematuri.

```
detach()
```