



# Laboratorio con R - 4

Metodi e Modelli per l'Inferenza Statistica - Ing. Matematica - a.a. 2018-19

05/06/2019

---

## 0. Librerie

---

```
library( rms )
## Loading required package: Hmisc
## Loading required package: lattice
## Loading required package: survival
## Loading required package: Formula
## Loading required package: ggplot2
## Registered S3 methods overwritten by 'ggplot2':
##   method      from
##   [.quosures   rlang
##   c.quosures   rlang
##   print.quosures rlang
##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##   format.pval, units
## Loading required package: SparseM
##
## Attaching package: 'SparseM'
## The following object is masked from 'package:base':
##
##   backsolve

library(ResourceSelection)
## ResourceSelection 0.3-4    2019-01-08
```

## Reference:

Agresti, A. (2003). Categorical data analysis (Vol. 482). John Wiley & Sons.

---

## 1. Regressione logistica semplice

---

Prendiamo in esame il dataset relativo ad uno studio clinico su pazienti affetti da disturbi coronarici. In particolare, l'obiettivo dello studio consiste nello spiegare la presenza o l'assenza di significativi disturbi

coronarici ( CHD ) in funzione dell'età ( variabile AGE ) dei pazienti. I dati si riferiscono a 100 pazienti. Le variabili del database sono descritte nel file *CHDAGE\_data\_description.txt*:

- **CHD** variabile dipendente binaria: 1 se il disturbo " presente, 0 se il disturbo " assente;
- **AGE** variabile indipendente ( continua ).

Sito da cui trarre dati e dataset <http://www.umass.edu/statdata/statdata/>

### Soluzione

Importiamo i dati.

```
chd = read.table( "CHDAGE_data.txt", head = TRUE )

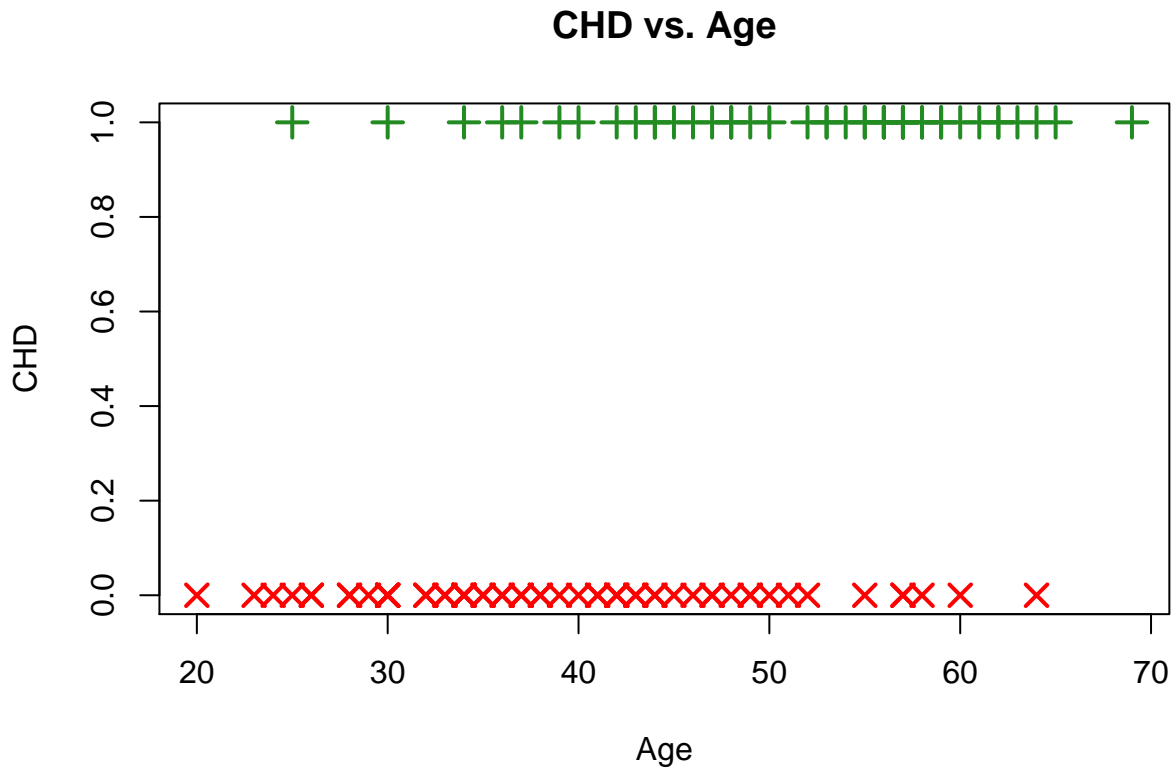
str( chd )
## 'data.frame':    100 obs. of  3 variables:
##  $ ID : int   1  2  3  4  5  6  7  8  9 10 ...
##  $ AGE: int   20 23 24 25 25 26 26 28 28 29 ...
##  $ CHD: int    0  0  0  0  1  0  0  0  0  0 ...

head( chd )
##    ID AGE CHD
## 1   1  20   0
## 2   2  23   0
## 3   3  24   0
## 4   4  25   0
## 5   5  25   1
## 6   6  26   0

attach( chd )
```

Visualizziamo i dati.

```
plot( AGE, CHD, pch = ifelse( CHD == 1, 3, 4 ),
      col = ifelse( CHD == 1, 'forestgreen', 'red' ),
      xlab = 'Age', ylab = 'CHD', main = 'CHD vs. Age', lwd = 2, cex = 1.5 )
```



Eseguiamo quindi un'analisi descrittiva del dataset.

Per meglio comprendere la natura della relazione  $\tilde{A}$  opportuno suddividere i pazienti in classi d'età e calcolare la media della variabile dipendente in ciascuna classe.

Inseriamo nel vettore  $x$  i limiti delle classi d'età che si vogliono creare ( questo passaggio  $\tilde{A}$  arbitrario, e va eseguito con buon senso ).

```
min( AGE )
## [1] 20
max( AGE )
## [1] 69

x = c( 20, 29, 34, 39, 44, 49, 54, 59, 70 )

# Calcoliamo i punti medi degli intervalli che abbiamo creato
mid = c( ( x [ 2:9 ] + x [ 1:8 ] )/2 )

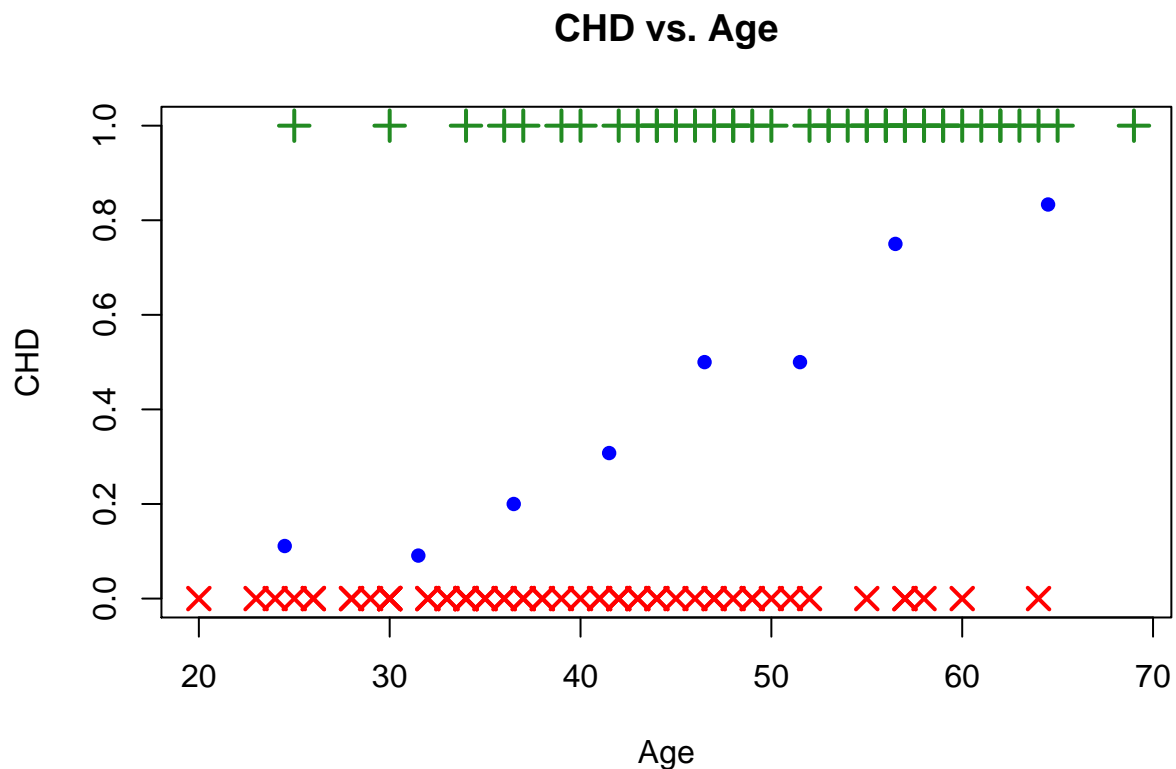
# Suddividiamo i dati nelle classi che abbiamo creato
GRAGE = cut( AGE, breaks = x, include.lowest = TRUE, right = FALSE )
GRAGE
## [1] [20,29) [20,29) [20,29) [20,29) [20,29) [20,29) [20,29) [20,29) [20,29)
## [9] [20,29) [29,34) [29,34) [29,34) [29,34) [29,34) [29,34) [29,34) [29,34)
## [17] [29,34) [29,34) [29,34) [29,34) [29,34) [34,39) [34,39) [34,39) [34,39)
## [25] [34,39) [34,39) [34,39) [34,39) [34,39) [34,39) [34,39) [34,39) [34,39)
## [33] [34,39) [34,39) [34,39) [34,39) [39,44) [39,44) [39,44) [39,44) [39,44)
## [41] [39,44) [39,44) [39,44) [39,44) [39,44) [39,44) [39,44) [39,44) [39,44)
## [49] [44,49) [44,49) [44,49) [44,49) [44,49) [44,49) [44,49) [44,49) [44,49)
```

```
## [57] [44,49) [44,49) [44,49) [44,49) [44,49) [44,49) [44,49) [49,54) [49,54)
## [65] [49,54) [49,54) [49,54) [49,54) [49,54) [49,54) [49,54) [49,54) [49,54)
## [73] [54,59) [54,59) [54,59) [54,59) [54,59) [54,59) [54,59) [54,59) [54,59)
## [81] [54,59) [54,59) [54,59) [54,59) [54,59) [54,59) [54,59) [54,59) [54,59)
## [89] [59,70] [59,70] [59,70] [59,70] [59,70] [59,70] [59,70] [59,70] [59,70]
## [97] [59,70] [59,70] [59,70] [59,70] [59,70]
## 8 Levels: [20,29) [29,34) [34,39) [39,44) [44,49) [49,54) ... [59,70]
```

Calcoliamo quindi la media della variabile AGE stratificata e sovrapponiamo i valori di y al grafico precedente.

```
y = tapply( CHD, GRAGE, mean )
y
## [20,29) [29,34) [34,39) [39,44) [44,49) [49,54)
## 0.11111111 0.09090909 0.20000000 0.30769231 0.50000000 0.50000000
## [54,59) [59,70]
## 0.75000000 0.83333333

plot( AGE, CHD, pch = ifelse( CHD == 1, 3, 4 ),
      col = ifelse( CHD == 1, 'forestgreen', 'red' ),
      xlab = 'Age', ylab = 'CHD', main = 'CHD vs. Age', lwd = 2, cex = 1.5 )
points( mid, y, col = "blue", pch = 16 )
```



Dal grafico si intuisce la natura della relazione fra AGE e CHD.

Identifichiamo un modello che descriva adeguatamente i nostri dati. Il modello più opportuno è un modello lineare generalizzato con link function di tipo `logit`.

```

help( glm )
## starting httpd help server ... done

mod = glm( CHD ~ AGE, family = binomial( link = logit ) )
summary( mod )
##
## Call:
## glm(formula = CHD ~ AGE, family = binomial(link = logit))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9718  -0.8456  -0.4576   0.8253   2.2859
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.30945     1.13365  -4.683 2.82e-06 ***
## AGE          0.11092     0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4

```

Il modello stimato  $\hat{A}$  quindi:

$$\text{logit}(\pi) = -5.30945 + 0.11092 \cdot \text{AGE}$$

in cui  $\pi$   $\hat{A}$  la probabilità  $\hat{A}$  che CHD sia pari ad 1.

Calcoliamo i valori stimati per il logit della probabilità  $\hat{A}$  di avere disturbi coronarici ( sono i logit di  $\pi_i$ , che giustamente hanno un range tipico di una variabile continua ).

```

mod$linear.predictors
##      1      2      3      4      5      6
## -3.09103053 -2.75826710 -2.64734596 -2.53642482 -2.53642482 -2.42550368
##      7      8      9     10     11     12
## -2.42550368 -2.20366139 -2.20366139 -2.09274025 -1.98181911 -1.98181911
##     13     14     15     16     17     18
## -1.98181911 -1.98181911 -1.98181911 -1.98181911 -1.75997682 -1.75997682
##     19     20     21     22     23     24
## -1.64905568 -1.64905568 -1.53813454 -1.53813454 -1.53813454 -1.53813454
##     25     26     27     28     29     30
## -1.53813454 -1.42721340 -1.42721340 -1.31629225 -1.31629225 -1.31629225
##     31     32     33     34     35     36
## -1.20537111 -1.20537111 -1.20537111 -1.09444997 -1.09444997 -0.98352883
##     37     38     39     40     41     42
## -0.98352883 -0.87260769 -0.87260769 -0.76168654 -0.76168654 -0.65076540
##     43     44     45     46     47     48
## -0.65076540 -0.65076540 -0.65076540 -0.53984426 -0.53984426 -0.53984426
##     49     50     51     52     53     54
## -0.42892312 -0.42892312 -0.42892312 -0.42892312 -0.31800197 -0.31800197

```

```
##          55          56          57          58          59          60
## -0.20708083 -0.20708083 -0.09615969 -0.09615969 -0.09615969 0.01476145
##          61          62          63          64          65          66
## 0.01476145 0.01476145 0.12568259 0.12568259 0.12568259 0.23660374
##          67          68          69          70          71          72
## 0.23660374 0.34752488 0.45844602 0.45844602 0.56936716 0.56936716
##          73          74          75          76          77          78
## 0.68028831 0.79120945 0.79120945 0.79120945 0.90213059 0.90213059
##          79          80          81          82          83          84
## 0.90213059 1.01305173 1.01305173 1.01305173 1.01305173 1.01305173
##          85          86          87          88          89          90
## 1.01305173 1.12397287 1.12397287 1.12397287 1.23489402 1.23489402
##          91          92          93          94          95          96
## 1.34581516 1.34581516 1.45673630 1.56765744 1.56765744 1.67857859
##          97          98          99         100
## 1.78949973 1.78949973 1.90042087 2.34410544
```

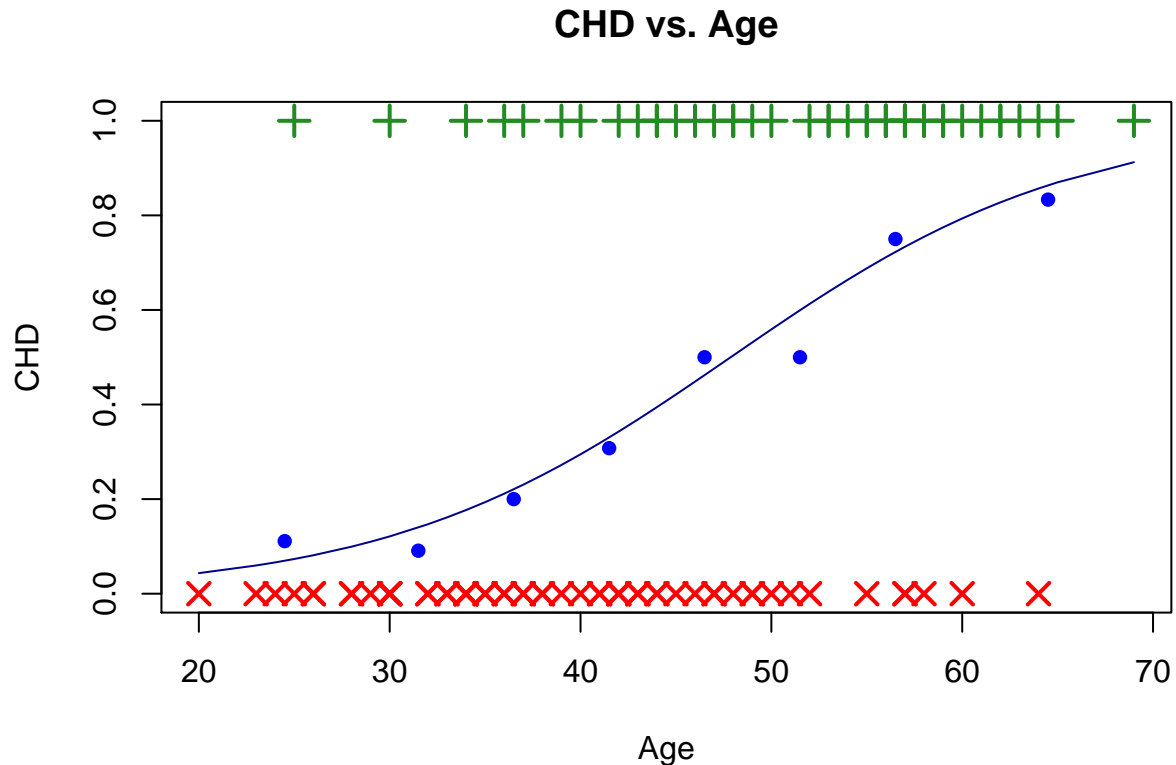
Caliamo i valori stimati per la probabilità  $\hat{A}$  di avere disturbi coronarici ( che coincidono con gli esponenziali dei valori ottenuti al punto prima ). Sono le  $\pi_i$  predette, pertanto comprese in  $[0, 1]$ .

```
mod$fitted.values
##          1          2          3          4          5          6
## 0.04347876 0.05962145 0.06615278 0.07334379 0.07334379 0.08124847
##          7          8          9         10         11         12
## 0.08124847 0.09942218 0.09942218 0.10980444 0.12112505 0.12112505
##          13         14         15         16         17         18
## 0.12112505 0.12112505 0.12112505 0.12112505 0.14679324 0.14679324
##          19         20         21         22         23         24
## 0.16123662 0.16123662 0.17680662 0.17680662 0.17680662 0.17680662
##          25         26         27         28         29         30
## 0.17680662 0.19353324 0.19353324 0.21143583 0.21143583 0.21143583
##          31         32         33         34         35         36
## 0.23052110 0.23052110 0.23052110 0.25078125 0.25078125 0.27219215
##          37         38         39         40         41         42
## 0.27219215 0.29471199 0.29471199 0.31828021 0.31828021 0.34281708
##          43         44         45         46         47         48
## 0.34281708 0.34281708 0.34281708 0.36822381 0.36822381 0.36822381
##          49         50         51         52         53         54
## 0.39438351 0.39438351 0.39438351 0.39438351 0.42116276 0.42116276
##          55         56         57         58         59         60
## 0.44841400 0.44841400 0.47597858 0.47597858 0.47597858 0.50369030
##          61         62         63         64         65         66
## 0.50369030 0.50369030 0.53137935 0.53137935 0.53137935 0.55887652
##          67         68         69         70         71         72
## 0.55887652 0.58601724 0.61264546 0.61264546 0.63861714 0.63861714
##          73         74         75         76         77         78
## 0.66380304 0.68809096 0.68809096 0.68809096 0.71138714 0.71138714
##          79         80         81         82         83         84
## 0.71138714 0.73361695 0.73361695 0.73361695 0.73361695 0.73361695
##          85         86         87         88         89         90
## 0.73361695 0.75472490 0.75472490 0.75472490 0.77467399 0.77467399
##          91         92         93         94         95         96
## 0.79344462 0.79344462 0.81103299 0.82744940 0.82744940 0.84271622
##          97         98         99         100
```

```
## 0.85686593 0.85686593 0.86993915 0.91246455
```

Facciamo un grafico della predizione del modello.

```
plot( AGE, CHD, pch = ifelse( CHD == 1, 3, 4 ),
      col = ifelse( CHD == 1, 'forestgreen', 'red' ),
      xlab = 'Age', ylab = 'CHD', main = 'CHD vs. Age', lwd = 2, cex = 1.5 )
points( mid, y, col = "blue", pch = 16 )
lines( AGE, mod$fitted, col = 'darkblue' )
```



### Interpretazione dei coefficienti

Uno dei motivi per cui la tecnica di regressione logistica  $\tilde{A}$  è largamente diffusa, specialmente in ambito clinico,  $\tilde{A}$  che i coefficienti del modello hanno una naturale interpretazione in termini di **odds ratio** ( nel seguito **OR** ).

Si consideri un predittore  $x$  dicotomico a livelli 0 e 1. Si definisce odds che  $y = 1$  fra gli individui con  $x = 0$  la quantità  $\tilde{A}$  :

$$\frac{\mathbb{P}(y = 1|x = 0)}{1 - \mathbb{P}(y = 1|x = 0)}.$$

Analogamente per i soggetti con  $x = 1$ , l'odds che  $y = 1$   $\tilde{A}$  :

$$\frac{\mathbb{P}(y = 1|x = 1)}{1 - \mathbb{P}(y = 1|x = 1)}.$$

L'OR  $\tilde{A}$  definito come il rapporto degli odds per  $x = 1$  e  $x = 0$ .

Dato che:

$$\mathbb{P}(y = 1|x = 1) = \frac{\exp(\beta_0 + \beta_1 \cdot x)}{1 + \exp(\beta_0 + \beta_1 \cdot x)}$$
$$\mathbb{P}(y = 1|x = 0) = \frac{\exp(\beta_0)}{1 + \exp(\beta_0)}$$

Il che implica:

$$\text{OR} = \exp(\beta_1)$$

Si possono costruire intervalli di confidenza e generalizzazioni al caso di variabile  $x$  con  $\pi^{\tilde{A}}$  categorie in modo immediato.

Calcoliamo quindi l'OR relativo a AGE.

```
summary( mod )
##
## Call:
## glm(formula = CHD ~ AGE, family = binomial(link = logit))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9718  -0.8456  -0.4576   0.8253   2.2859
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.30945     1.13365  -4.683 2.82e-06 ***
## AGE          0.11092     0.02406   4.610 4.02e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 136.66  on 99  degrees of freedom
## Residual deviance: 107.35  on 98  degrees of freedom
## AIC: 111.35
##
## Number of Fisher Scoring iterations: 4
```

Il coefficiente della variabile AGE vale 0.111. Quindi l'OR per un incremento di 10 anni d'età  $\tilde{A}$ :

```
exp( 10 * coef( mod ) [ 2 ] )
##      AGE
## 3.031967
```

per ogni incremento di 10 anni d'età  $\tilde{A}$ , il rischio di disturbo coronarico aumenta di 3 volte circa.

**N.B.:** il modello sottointende che il logit sia lineare nella variabile età  $\tilde{A}$ , ossia che l'OR fra persone di 20 contro 30 anni sia lo stesso che fra individui di 40 contro 50 anni.

### IC per la regressione logistica

Calcoliamo un intervallo di confidenza al 95% per l'OR per un incremento di 10 anni d'età  $\tilde{A}$ .



```

alpha = 0.05
qalpha = qnorm( 1 - alpha/2 )
qalpha
## [1] 1.959964

IC.sup = exp( 10 * coef( mod ) [ 2 ] + qalpha * 10 * summary( mod )$coefficients[ 2, 2 ] )
IC.inf = exp( 10 * coef( mod ) [ 2 ] - qalpha * 10 * summary( mod )$coefficients[ 2, 2 ] )
c( IC.inf, IC.sup )
##      AGE      AGE
## 1.892025 4.858721

```

Per costruire in R l'intervallo di confidenza del logit si può partire dal calcolo della matrice di covarianza dei parametri  $\beta$  stimati:

```

V = vcov( mod )
V
##      (Intercept)      AGE
## (Intercept)  1.28517059 -0.0266769747
## AGE         -0.02667697  0.0005788748

```

Intervallo di confidenza in corrispondenza di un valore di  $x$  ( ad esempio  $x = 50$  anni ).

```

x = 50

# errore standard
predict( mod, data.frame( AGE = 50 ), se = TRUE )
## $fit
##      1
## 0.2366037
##
## $se.fit
## [1] 0.2542835
##
## $residual.scale
## [1] 1

# oppure
sqrt( V [ 1, 1 ] + x^2 * V [ 2, 2 ] + 2 * x * V [ 1, 2 ] )
## [1] 0.2542835

# Rappresentazione grafica dell'intervallo di confidenza ( al 95% ) della regressione

# griglia di valori di x in cui valutare la regressione
grid = ( 20:69 )

se = predict( mod, data.frame( AGE = grid ), se = TRUE )
# errori standard corrispondenti ai valori della griglia

help( binomial )
gl = binomial( link = logit ) # funzione di link utilizzata
# Family objects provide a convenient way to specify the details of the models
# used by functions such as glm.

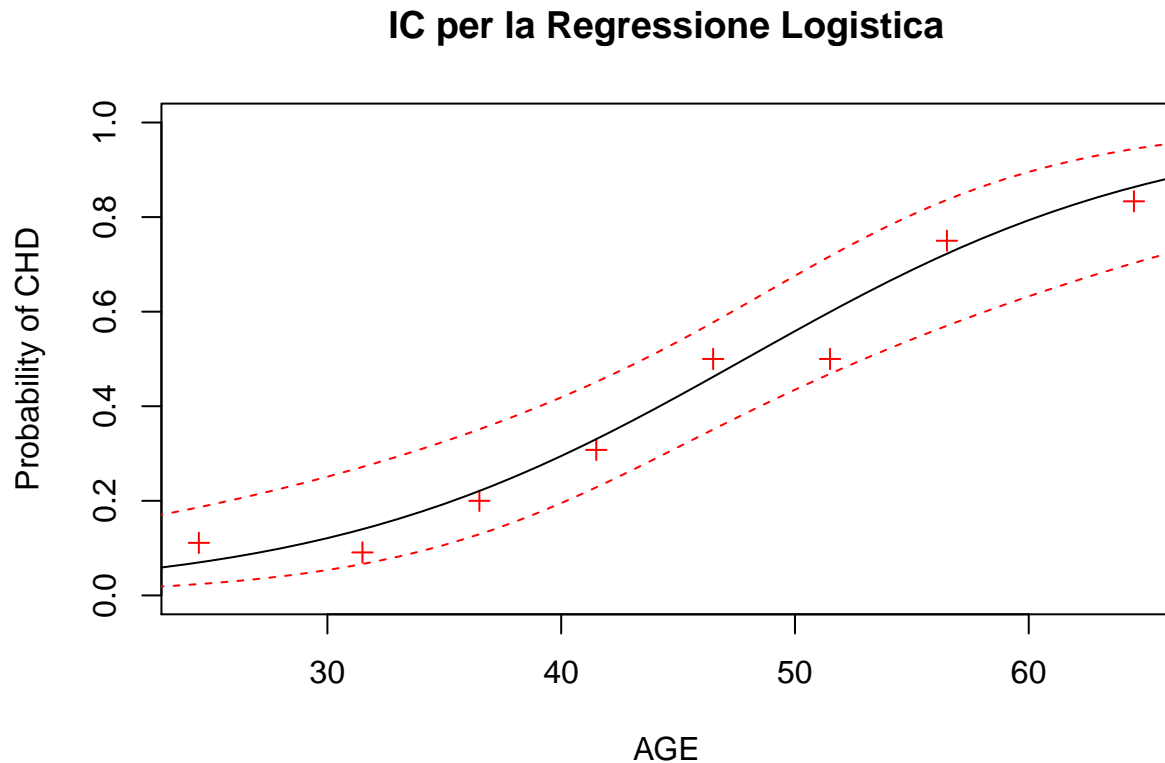
plot( mid, y, col = "red", pch = 3, ylim = c( 0, 1 ), ylab = "Probability of CHD",

```

```

xlab = "AGE", main = "IC per la Regressione Logistica" )
lines( grid, gl$linkinv( se$fit ) )
lines( grid, gl$linkinv( se$fit - qnorm( 1-0.025 ) * se$se ), col = "red", lty = 2 )
lines( grid, gl$linkinv( se$fit + qnorm( 1-0.025 ) * se$se ), col = "red", lty = 2 )

```



**N.B.** la funzione `gl$linkinv` permette di ottenere il valore delle probabilità a partire dalla link function (logit).

#### Goodness of fit

Varie tecniche sono state sviluppate e confrontate per stabilire la bontà del fit di una regressione logistica. Problema: tali tecniche soffrono di una limitata potenza (tipicamente non superiore al 50%) per campioni di dimensione  $n < 400$ .

Se la variabile indipendente è categorica si possono paragonare i valore di Devianza del modello fittato con il valore critico di una distribuzione  $\chi^2(n - p)$ , dove  $p$  è il numero di parametri del modello. Se  $D$  è maggiore del valore critico si rifiuta l'ipotesi nulla che il modello sia un buon fit.

Se la variabile indipendente è continua (es in questione), la procedura precedente perde di validità e i valori  $P$  che si ottengono non sono corretti. L'alternativa che R fornisce richiede l'installazione di due librerie supplementari (`Design` e `Hmisc`), che contengono le funzioni `lrm` e `residuals` per calcolare tale statistica.

```

#library( rms )
#help( lrm )

mod2 = lrm( CHD ~ AGE, x = TRUE, y = TRUE )
mod2
## Logistic Regression Model

```

```
##
## lrm(formula = CHD ~ AGE, x = TRUE, y = TRUE)
##
##               Model Likelihood      Discrimination      Rank Discrim.
##               Ratio Test           Indexes           Indexes
## Obs           100   LR chi2      29.31   R2           0.341   C           0.800
## 0              57   d.f.           1     g           1.504   Dxy          0.600
## 1              43   Pr(> chi2) <0.0001  gr          4.497   gamma         0.612
## max |deriv| 7e-06                                gp          0.297   tau-a         0.297
##                                           Brier          0.178
##
##           Coef      S.E.   Wald Z Pr(>|Z|)
## Intercept -5.3095 1.1337 -4.68 <0.0001
## AGE        0.1109 0.0241  4.61 <0.0001
##
anova( mod2 )
##           Wald Statistics           Response: CHD
##
## Factor      Chi-Square d.f. P
## AGE         21.25      1   <.0001
## TOTAL       21.25      1   <.0001
```

La funzione `lrm` Ã una procedura alternativa per fittare una regressione logistica. I risultati coincidono con quelli ottenuti in precedenza. Il test di goodness-of-fit si esegue con la chiamata:

```
#help(residuals.lrm)

residuals( mod2, "gof" )
## Sum of squared errors      Expected value|H0      SD
##      17.8301075           17.9263123           0.2204658
##              Z              P
##      -0.4363705           0.6625679
```

dal valore di Z ( e del valore P associato ) si conclude che l'ipotesi  $H_0$  che il modello sia un buon fit non puÃ essere rifiutata.

Alternativamente possiamo usare come GOF test, il test di Hosmer-Lemeshow.

```
hoslem.test( mod$y, fitted( mod ), g = 10 )
##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod$y, fitted(mod)
## X-squared = 2.2243, df = 8, p-value = 0.9734
```

In questo test dobbiamo scegliere  $g$ , numero di gruppi. Nel paper originale Ã suggerito di scegliere  $g > p$ , in questo caso quindi  $g > 2$ . Si vede che, anche cambiando  $g$ , giungiamo alla stessa conclusione, ovvero il modello fitta bene i dati. In generale la scelta del numero di gruppi a priori Ã un limite di questo test.

---

## 2. Regressione logistica multipla

---

In questo esercizio analizzeremo un dataset clinico inerente al peso di neonati. Lo scopo dello studio consiste nell'identificare i fattori di rischio associati con il partorire bambini di peso inferiore ai 2500 grammi ( low birth weight ). I dati si riferiscono a  $n = 189$  donne.

Le variabili del database sono descritte nel file "LOWBWT\_data\_description.txt":

- **LOW**: variabile dipendente binaria ( 1 se il neonato pesa meno di 2500 grammi, 0 viceversa );
- **AGE**, **LWT**, **FTV** variabili indipendenti continue;
- **RACE** variabile indipendente discreta a 3 livelli.

### Soluzione

Importiamo i dati.

```
lw = read.table( "LOWBWTdata.txt", head = TRUE )
attach( lw )
## The following objects are masked from chd:
##
##      AGE, ID

RACE = factor( RACE ) # tratto la variabile RACE come categorica

mod.low = glm( LOW ~ LWT + RACE + AGE + FTV, family = binomial( link = logit ) )
summary( mod.low )
##
## Call:
## glm(formula = LOW ~ LWT + RACE + AGE + FTV, family = binomial(link = logit))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4163  -0.8931  -0.7113   1.2454   2.0755
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.295366   1.071443   1.209   0.2267
## LWT         -0.014245   0.006541  -2.178   0.0294 *
## RACE2        1.003898   0.497859   2.016   0.0438 *
## RACE3        0.433108   0.362240   1.196   0.2318
## AGE         -0.023823   0.033730  -0.706   0.4800
## FTV         -0.049308   0.167239  -0.295   0.7681
## ---
## 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: 222.57  on 183  degrees of freedom
## AIC: 234.57
```



```
##
## Number of Fisher Scoring iterations: 4
```

Se ci si attiene alla sola significatività statistica si conclude che è possibile fittare un modello ‘parsimonioso’, contenente la sola variabile indipendente LWT. Tuttavia, come nel caso di regressione lineare multipla, l’inclusione di una variabile nel modello può avvenire per motivi differenti. Ad esempio, in questo caso, la variabile RACE è considerata in letteratura come importante nel predire l’effetto in questione, quindi la si include nel modello ristretto.

```
mod.low2 = glm( LOW ~ LWT + RACE, family = binomial( link = logit ) )

summary( mod.low2 )
##
## Call:
## glm(formula = LOW ~ LWT + RACE, family = binomial(link = logit))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3491  -0.8919  -0.7196   1.2526   2.0993
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.805753   0.845167   0.953   0.3404
## LWT         -0.015223   0.006439  -2.364   0.0181 *
## RACE2        1.081066   0.488052   2.215   0.0268 *
## RACE3        0.480603   0.356674   1.347   0.1778
## ---
## 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: 223.26  on 185  degrees of freedom
## AIC: 231.26
##
## Number of Fisher Scoring iterations: 4
```

Notiamo che AIC diminuisce e anche RACE acquista significatività.

```
anova( mod.low2, mod.low, test = "Chisq" )
## Analysis of Deviance Table
##
## Model 1: LOW ~ LWT + RACE
## Model 2: LOW ~ LWT + RACE + AGE + FTV
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      185      223.26
## 2      183      222.57  2   0.68618   0.7096
```

Non c’è motivo di ritenere che il modello contenente solamente LWT e RACE sia meno informativo del modello completo.

## Odds ratio

Il predittore RACE è discreto a 3 livelli. In questo caso il livello 1 ( RACE = White ) viene assunto come categoria di riferimento.

```
model.matrix( mod.low2 ) [ 1:15, ]
##      (Intercept) LWT RACE2 RACE3
## 1             1 182      1      0
## 2             1 155      0      1
## 3             1 105      0      0
## 4             1 108      0      0
## 5             1 107      0      0
## 6             1 124      0      1
## 7             1 118      0      0
## 8             1 103      0      1
## 9             1 123      0      0
## 10            1 113      0      0
## 11            1  95      0      1
## 12            1 150      0      1
## 13            1  95      0      1
## 14            1 107      0      1
## 15            1 100      0      0
```

*# OR 2 vs 1 ( Black vs White )*

```
exp( coef( mod.low2 ) [ 3 ] )
##      RACE2
## 2.947821
```

Le donne nere sono una categoria con rischio di parto prematuro quasi 3 volte superiore alle donne bianche.

*# OR 3 vs 1 ( Other vs White )*

```
exp( coef( mod.low2 ) [ 4 ] )
##      RACE3
## 1.61705
```

Le donne di altre etnie sono una categoria con rischio di parto prematuro circa 1.5 volte superiore alle donne bianche.

Facciamo un check sul GOF del modello.

```
mod.low2lrn = lrm( LOW ~ LWT + RACE, x = TRUE, y = TRUE )
residuals( mod.low2lrn, "gof" )
## Sum of squared errors      Expected value|H0      SD
##           38.2268160           38.2138614      0.1733477
##           Z           P
##           0.0747321           0.9404279
```

```
hoslem.test( mod.low2$y, fitted( mod.low2 ), g = 6 )
##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  mod.low2$y, fitted(mod.low2)
## X-squared = 3.1072, df = 4, p-value = 0.5401
#g > 3
```

Anche in questo caso, possiamo concludere che il modello dà un buon fit dei dati.

### Tabelle di classificazione

Un modo spesso utilizzato per presentare i risultati di un fit tramite regressione logistica sono le tabelle di classificazione. In queste tabelle i dati vengono classificati secondo due chiavi:

1. il valore della variabile dipendente dicotoma  $y$ ;
2. il valore di una variabile dicotoma  $y_{mod}$ , che si deriva dalla stima della probabilità  $\hat{A}$  ottenuta dal modello. I valori di questa variabile si ottengono confrontando il valore della probabilità  $\hat{A}$  con un cut-off ( valore usuale 0.5 )

```
soglia = 0.5

valori.reali = lw$LOW
valori.predetti = as.numeric( mod.low2$fitted.values > soglia )
# 1 se > soglia, 0 se <= soglia
valori.predetti
## [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 0 0
## [36] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [71] 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [106] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [141] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0
## [176] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0

tab = table( valori.reali, valori.predetti )

tab
##          valori.predetti
## valori.reali  0    1
##           0 124    6
##           1  53    6

# % di casi classificati correttamente:
round( sum( diag( tab ) ) / sum( tab ), 2 )
## [1] 0.69

# % di casi misclassificati:
round( ( tab [ 1, 2 ] + tab [ 2, 1 ] ) / sum( tab ), 2 )
## [1] 0.31
```

**SENSITIVITA':**

$$\mathbb{P}(\text{predetto} = 1 | \text{reale} = 1)$$

```
sensitivita = tab [ 2, 2 ] / ( tab [ 2, 1 ] + tab [ 2, 2 ] )
sensitivita
## [1] 0.1016949
```

**SPECIFICITA':**

$$\mathbb{P}(\text{predetto} = 0 | \text{reale} = 0)$$

```
specificita = tab [ 1, 1 ] / ( tab [ 1, 2 ] + tab [ 1, 1 ] )
specificita
## [1] 0.9538462
```

---

### 3. Curva ROC

---

Costruire la Curva ROC a partire dai valori predetti per la risposta dal modello `mod.low2` dell'analisi della variabile LOWBT.

#### Soluzione

```
fit2 = mod.low2$fitted

#media campionaria della prob di sopravvivenza nel campione

soglia_roc = seq( 0, 1, length.out = 2e2 )
lens = length( soglia_roc )-1
ascissa_roc = rep( NA, lens )
ordinata_roc = rep( NA, lens )

for ( k in 1 : lens )
{
  soglia = soglia_roc [ k ]

  classification = as.numeric( sapply( fit2, function( x ) ifelse( x < soglia, 0, 1 ) ) )

  # ATTENZIONE, voglio sulle righe il vero e sulle colonne il predetto
  # t.misc = table( lw$LOW, classification )

  ordinata_roc[ k ] = sum( classification[ which( lw$LOW == 1 ) ] == 1 ) /
    length( which( lw$LOW == 1 ) )

  ascissa_roc[ k ] = sum( classification[ which( lw$LOW == 0 ) ] == 1 ) /
    length( which( lw$LOW == 0 ) )

  # ordinata_roc [ k ] = t.misc [ 1, 1 ] / ( t.misc [ 1, 1 ] + t.misc [ 1, 2 ] )
  #
  # ascissa_roc [ k ] = t.misc [ 2, 1 ] / ( t.misc [ 2, 1 ] + t.misc [ 2, 2 ] )
}
```

Visualizziamo la curva ROC.

```
plot( ascissa_roc, ordinata_roc, type = "l", xlab = "1 - Specificity", ylab = "Sensitivity",
      main = "Curva ROC", lwd = 2, col = 'darkblue', ylim = c( 0, 1 ), xlim = c( 0, 1 ) )
abline( h = c( 0, 1 ), v = c( 0, 1 ), lwd = 1, lty = 2, col = 'red' )
abline( a = 0, b = 1, lty = 2, col = 'black' )

# qual era il nostro punto?
abline( v = 1 - specificita, h = sensitivita, lty = 3, col = 'blue' )
points( 1 - specificita, sensitivita, pch = 4, lwd = 3, cex = 1.5, col = 'blue' )
```



## Curva ROC

