Laboratori di Statistica (cp)

1. Regressione binomiale

Leonardo Egidi

AA 2021/2022

1 Disastro del *Challenger*

La navetta spaziale *Challenger* esplode a 73 secondi dal suo decimo lancio, il 28 gennaio del 1986. La causa dell'esplosione è individuata nella rottura di una delle guarnizioni ad anello (*O-ring*) presenti nei razzi laterali[1].
Nell'indagine sul disastro si ipotizza che un guasto in un *O-ring* sia più probabile quando la temperatura è bassa. Si tratta di un punto fondamentale in quanto, se vero, renderebbe il disastro prevedibile essendosi verificata, quel giorno, una temperatura eccezionalmente bassa (intorno a 0^oC).
Per studiare il fenomeno, si dispone delle registrazioni dei guasti agli *O-ring* in 23 passate missioni.

```
chl=read.table("datichallenger.csv",sep=";")
head(chl)
```

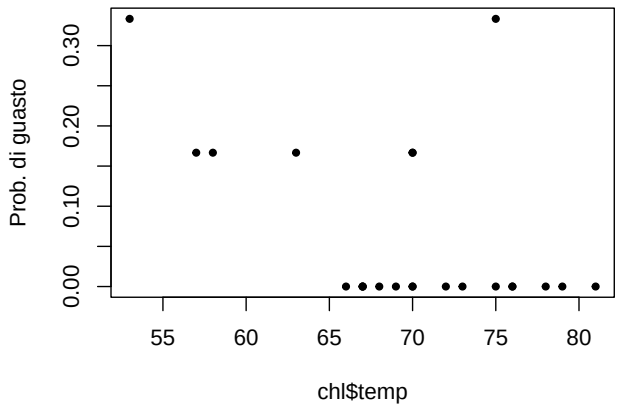
```
##   failures temp
## 1        0   66
## 2        1   70
## 3        0   69
## 4        0   68
## 5        0   67
## 6        0   72
```

Ciascuna riga si riferisce a una missione (lancio), le colonne sono rispettivamente il numero di guasti osservati tra le 6 guarnizioni e la temperatura in gradi Fahrenheit (la temperatura in gradi Celsius t_C si ottiene dalla temperatura in gradi Fahrenheit t_F come $t_C = (t_F - 32)/1 - 8)$).

[1]Ciascuno dei razzi è costituito da 4 segmenti, collegati da supporti chiusi da due guarnizioni ad anello (che diventano tre dopo il disastro del *Challenger*).

Si può stimare grossolanamente la probabilità di guasto in ciascun *O-ring* in un lancio come `chl$failures/6`.

```
plot(chl$failures/6 ~ chl$temp, pch=20, ylab="Prob. di guasto")
```



Appare chiaro che con temperature più alte si ha una minore probabilità di guasto, vogliamo quindi stimare un modello per tale relazione che permetta tra l'altro di stimare la probabilità di guasto con 31^oF che, si noti, richiede un'estrapolazione non indifferente.
In prima battuta stimiamo un modello lineare in cui la variabile risposta è la proporzione di guasti in un lancio e l'esplicativa è la temperatura durante il medesimo lancio.

```
modlin <- lm(failures/6 ~ temp,data=chl)
summary(modlin)

##
## Call:
## lm(formula = failures/6 ~ temp, data = chl)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09347 -0.06573 -0.01423  0.01760  0.31118
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.616402   0.203252   3.033  0.00633 **
## temp        -0.007923   0.002907  -2.725  0.01268 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09624 on 21 degrees of freedom
## Multiple R-squared:  0.2613,Adjusted R-squared:  0.2261
```

```
## F-statistic: 7.426 on 1 and 21 DF,  p-value: 0.01268
```

Comando 1: funzione `lm`

La funzione per stimare un modello lineare è `lm` e prende come argomenti

- la formula che specifica variabile risposta e esplicative, nella forma

  `risposta~esplicativa1+esplicativa2`

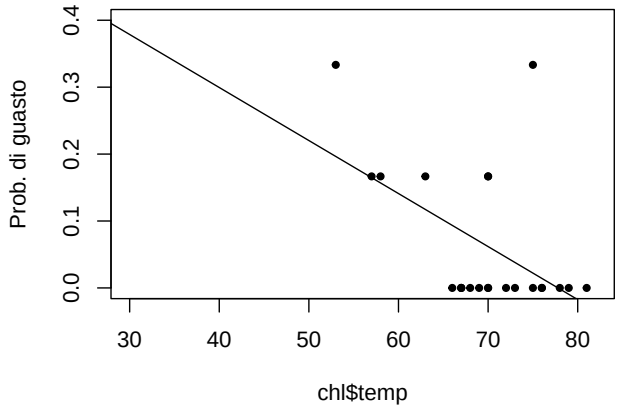
- il *data.frame* dove rintracciare le variabili nella formula.

Il modello conferma che la probabilità diminuisce all'aumentare della temperatura, la probabilità associata a una temperatura di 31^oF è

```
predict(modlin,newdata=data.frame(temp=31))

##         1
## 0.3707804

coef(modlin)[1]+coef(modlin)[2]*31

## (Intercept)
##   0.3707804
```

L'uso di questo modello è però quantomai dubbio, le probabilità stimate possono essere al di fuori dell'intervallo tra 0 e 1, come in effetti avviene per temperature superiori a 78.

```
plot(chl$failures/6 ~ chl$temp,pch=20, ylab="Prob. di guasto",
     xlim=c(30,82), ylim=c(0,0.4))
abline(modlin)
```



Più ragionevolmente, adottiamo un modello di regressione logistica, con la temperatura come esplicativa, cioè assumiamo che

$$\pi_i = P(Y_i = 1)$$

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_1 + \beta_2 x_i$$

questo si adatta in R con la funzione `glm`, in particolare la sintassi è

```
modlogit <- glm(cbind(failures,6-failures) ~ temp,
                data=chl,
                family=binomial(link=logit))
```

Notiamo che:

- in luogo della variabile risposta si ha una matrice di dati $n \times 2$ in cui la prima colonna rappresenta il numero di successi della binomiale – qui i guasti – e la seconda colonna il numero di insuccessi – qui gli *O-ring* intatti.
- Le variabili esplicative sono specificate come in `lm`.
- L'argomento `family` specifica la distribuzione della variabile risposta, in aggiunta si specifica la funzione legame (si noti che se la funzione legame desiderata è la logistica non occorrerebbe specificare nulla in quanto è l'opzione predefinita).

Esaminiamo poi il risultato della stima.

```
summary(modlogit)
```

```
##
## Call:
## glm(formula = cbind(failures, 6 - failures) ~ temp, family = binomial(link = logit),
##     data = chl)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.95227  -0.78299  -0.54117  -0.04379   2.65152
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  5.08498    3.05247   1.666   0.0957 .
## temp        -0.11560    0.04702  -2.458   0.0140 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 24.230  on 22  degrees of freedom
## Residual deviance: 18.086  on 21  degrees of freedom
## AIC: 35.647
##
## Number of Fisher Scoring iterations: 5
```

Il risultato di `summary` contiene

- il comando di cui `fit` è il risultato (quindi il modello);
- alcune statistiche – quantili in particolare – sui residui di devianza;
- la tabella dei coefficienti, formata da due righe, una per coefficiente, che contengono:
  - la stima $\hat{\beta}_j$,
  - la deviazione standard stimata dello stimatore $\sqrt{\hat{V}(\hat{\beta}_j)}$,
  - la statistica z per il test di nullità del coefficiente: $z_j = \hat{\beta}_j / \sqrt{\hat{V}(\hat{\beta}_j)}$,
  - il valore p per il test di nullità del coefficiente contro l'alternativa bilaterale, $2(1-\Phi(|z_j|))$,
  - a seguire sono riportati simboli che sintetizzano il livello di significatività osservato, interpretabili secondo lo schema spiegato sotto la tabella (*** se il valore p è compreso tra 0 e 0.001, ** se compreso tra 0.001 e 0.01, * tra 0.01 e 0.05, . tra 0.05 e 0.1, nulla altrimenti);
- viene ricordato che il parametro di dispersione è strutturalmente pari a 1 data l'ipotesi distributiva;
- sono riportate la devianza nulla e quella residua con i relativi gradi di libertà;
- è riportato l'AIC del modello;
- infine, è riportato il numero di iterazioni dell'algoritmo di ottimizzazione utilizzato.

La stima mostra che la probabilità diminuisce al crescere della temperatura, la previsione quando la temperatura è 31 è calcolata in tre modi diversi,

```
predict(modlogit,newdata=data.frame(temp=31),type="response")

##         1
## 0.8177744

exp(predict(modlogit,newdata=data.frame(temp=31)))/
  (1+exp(predict(modlogit,newdata=data.frame(temp=31))))

##         1
## 0.8177744
```

```
exp(coef(modlogit)[1]+coef(modlogit)[2]*31)/
  (1+exp(coef(modlogit)[1]+coef(modlogit)[2]*31))

## (Intercept)
##   0.8177744
```

dove notiamo che `predict` fornisce, se l'argomento `type` è `"response"`, la probabilità stimata:

$$\hat{\pi}_0 = \frac{e^{\hat{\beta}_1+\hat{\beta}_2 x_0}}{1+e^{\hat{\beta}_1+\hat{\beta}_2 x_0}};$$

se invece `type` è `"link"` (valore predefinito),

$$\hat{\beta}_1 + \hat{\beta}_2 x_0.$$

Stimiamo anche dei modelli con funzioni legame alternative

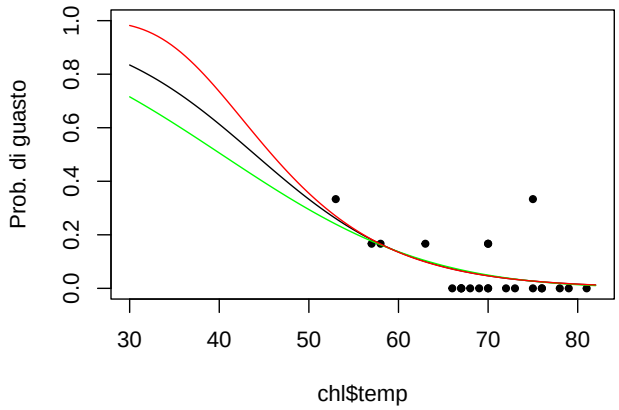
```
modprobit <- glm(cbind(failures,6-failures) ~ temp,
                 data=chl,
                 family=binomial(link=probit))
modcloglog <- glm(cbind(failures,6-failures) ~ temp,
                  data=chl,
                  family=binomial(link=cloglog))
```

e confrontiamo le previsioni

```
predict(modlogit,newdata=data.frame(temp=31),type="response")

##         1
## 0.8177744

predict(modprobit,newdata=data.frame(temp=31),type="response")

##         1
## 0.6963991

predict(modcloglog,newdata=data.frame(temp=31),type="response")

##         1
## 0.9726399
```

È interessante confrontare le previsioni al variare della temperatura, disegniamo quindi tre curve con i seguenti comandi:

```
plot(chl$failures/6 ~ chl$temp,pch=20,ylab="Prob. di guasto",
     xlim=c(30,82),ylim=c(0,1))
curve(predict(modlogit,newdata=data.frame(temp=x),type="response"),add=TRUE)
curve(predict(modprobit,newdata=data.frame(temp=x),type="response"),
      add=TRUE,col="green")
curve(predict(modcloglog,newdata=data.frame(temp=x),type="response"),
      add=TRUE,col="red")
```



Si noti che la differenza è contenuta, ma non se si guarda all'estrapolazione.

```
modlogit0 <- glm(cbind(failures,6-failures) ~ 1,
                 data=chl,
                 family=binomial(link=logit))
anova(modlogit0,modlogit)

## Analysis of Deviance Table
##
## Model 1: cbind(failures, 6 - failures) ~ 1
## Model 2: cbind(failures, 6 - failures) ~ temp
##   Resid. Df Resid. Dev Df Deviance
## 1        22     24.230
## 2        21     18.086  1    6.144
```

2 Dati sulla sopravvivenza di pazienti in un reparto di cura intensiva

Per 200 pazienti ammessi in un reparto di cura intensiva sono state registrate alcune caratteristiche rilevanti per il loro stato di salute, oltre all'esito finale, cioè se siano sopravvissuti o meno.
In particolare si sono osservate le variabili

- `stato`: stato del paziente (0 = vivo, 1 = deceduto)
- `eta`: età in anni del paziente
- `causa`: causa dell'ammissione (1= programmata, 2 = emergenza)
- `cosciente`: livello di coscienza all'ammissione (1 = no coma o stupor, 2 = coma o stupor)

Gli obiettivi che ci si pone sono da una parte stabilire quali delle caratteristiche rilevate abbiano un'influenza sulla probabilità di sopravvivenza, dall'altra predire la probabilità di sopravvivenza date le anzidette caratteristiche.
Si carica il campione con il comando:

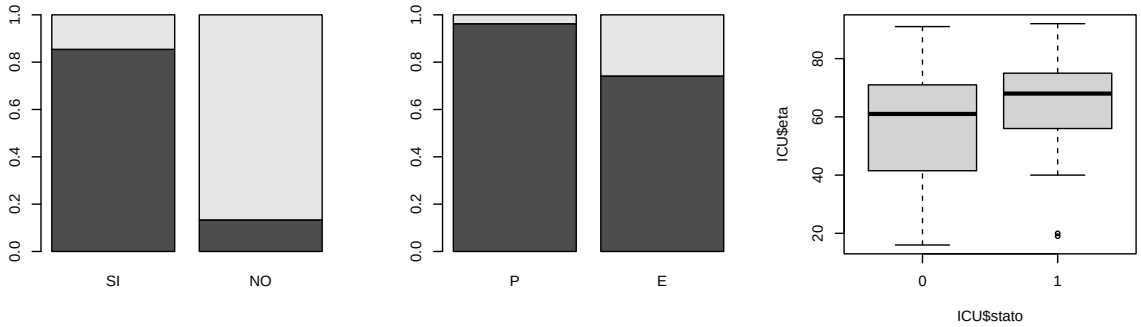
```
ICU <- read.table("ICUdata.dat",header=TRUE)
head(ICU)

##   stato eta causa cosciente
## 1     0  27     2         1
## 2     0  59     2         1
## 3     0  77     1         1
## 4     0  54     2         1
## 5     0  87     2         1
## 6     0  69     2         1
```

Ricodifichiamo i fattori in maniera opportuna, assegnando nomi interpretabili per i livelli.

```
ICU$causa <- factor(ICU$causa)
levels(ICU$causa) <- c("P","E")
ICU$cosciente <- factor(ICU$cosciente)
levels(ICU$cosciente) <- c("SI","NO")
```

Analizziamo la relazione tra variabile risposta e variabili esplicative usando, as esempio, diagrammi a barre e diagrammi a scatola.

```
barplot(prop.table(table(ICU$stato,ICU$cosciente),2))
barplot(prop.table(table(ICU$stato,ICU$causa),2))
boxplot(ICU$eta~ICU$stato)
```



Per stimare l'effetto delle variabili esplicative è naturale specificare un modello di regressione logistica in cui, con Y_i la variabile che vale 1 se il paziente decede, x_{i2} l'età, x_{i3} la causa di ammissione al reparto (1 se l'ammissione è non programmata (emergenza), 0 altrimenti) e x_{i4} è la variabile che vale 1 se il paziente entra incosciente e 0 altrimenti. Si assume allora

$$\pi_i = P(Y_i = 1)$$

$$\log\left(\frac{\pi_i}{1-\pi_i}\right) = \beta_1 + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4}$$

La stima di massima verosimiglianza del modello si ottiene con

```
ICU.fit <- glm(stato~.,family=binomial,data=ICU)
summary(ICU.fit)

##
## Call:
## glm(formula = stato ~ ., family = binomial, data = ICU)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4804  -0.6551  -0.3487  -0.1976   2.5621
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.77153    1.14618  -5.035 4.77e-07 ***
## eta          0.03370    0.01208   2.789  0.00529 **
## causaE       2.34470    0.80684   2.906  0.00366 **
## coscienteNO  3.45622    0.82935   4.167 3.08e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 200.16  on 199  degrees of freedom
```

```
## Residual deviance: 145.31  on 196  degrees of freedom
## AIC: 153.31
##
## Number of Fisher Scoring iterations: 6
```

Devianza e adeguatezza del modello
I test sulla nullità dei singoli coefficienti rifiutano tutti l'ipotesi nulla, consideriamo comunque anche il test per la significatività del modello nel suo complesso, cioè verifichiamo l'ipotesi

$$H_0 : \beta_2 = \beta_3 = \beta_4 = 0$$

confrontando, mediante il test del rapporto di verosimiglianza, il modello con la sola intercetta e il modello completo, appena stimato. Si considera dunque la statistica:

$$W = 2(\ell(\hat{\boldsymbol{\beta}}) - \ell(\hat{\boldsymbol{\beta}}_{MR})) = \frac{D_{MR} - D}{\phi},$$

e la si confronta con un χ^2_3, dove
- $D_{MR} = 2\phi(\ell(\tilde{\boldsymbol{\beta}}) - \ell(\hat{\boldsymbol{\beta}}_{MR}))$ è la devianza del modello ridotto;
- $D = 2\phi(\ell(\tilde{\boldsymbol{\beta}}) - \ell(\hat{\boldsymbol{\beta}}))$ è la devianza del modello corrente,

e $\ell(\tilde{\boldsymbol{\beta}})$ è la verosimiglianza del modello saturo (tanti parametri quante osservazioni statistiche).
La stima del modello con la sola intercetta si ottiene come

```
ICU.fit0 <- glm(stato~1,family=binomial,data=ICU)
summary(ICU.fit0)
```

```
##
## Call:
## glm(formula = stato ~ 1, family = binomial, data = ICU)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6681  -0.6681  -0.6681  -0.6681   1.7941
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3863     0.1768  -7.842 4.43e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 200.16  on 199  degrees of freedom
## Residual deviance: 200.16  on 199  degrees of freedom
## AIC: 202.16
##
## Number of Fisher Scoring iterations: 4
```

Vale la pena notare che la probabilità stimata π_0, la stessa per tutti, è

```
p0 <- exp(coef(ICU.fit0))/(1+exp(coef(ICU.fit0)))
p0

## (Intercept)
##         0.2
```

che è pari alla frequenza osservata di decessi nel campione, infatti

```
table(ICU$stato)

##
##   0   1
## 160  40
```

La log verosimiglianza nel modello ridotto è pertanto

$$\ell(\boldsymbol{\beta}_{MR}) = \left(\sum_{i=1}^{n} y_i\right) \log(\pi_0) + \left(n - \sum_{i=1}^{n} y_i\right) \log(1 - \pi_0),$$

si ha infatti

```
n <- nrow(ICU)
s <- sum(ICU$stato)
s*log(p0)+(n-s)*log(1-p0)

## (Intercept)
##   -100.0805
```

Questo stesso risultato si può ottenere con

```
logLik(ICU.fit0)

## 'log Lik.' -100.0805 (df=1)
```

Il test del rapporto di verosimiglianza è dunque

```
W <- 2*(logLik(ICU.fit)-logLik(ICU.fit0))
W

## 'log Lik.' 54.85033 (df=4)
```

e quindi il valore p è dato da

```
1-pchisq(W,3)

## 'log Lik.' 7.389978e-12 (df=4)
```

Questo stesso risultato si può ottenere con

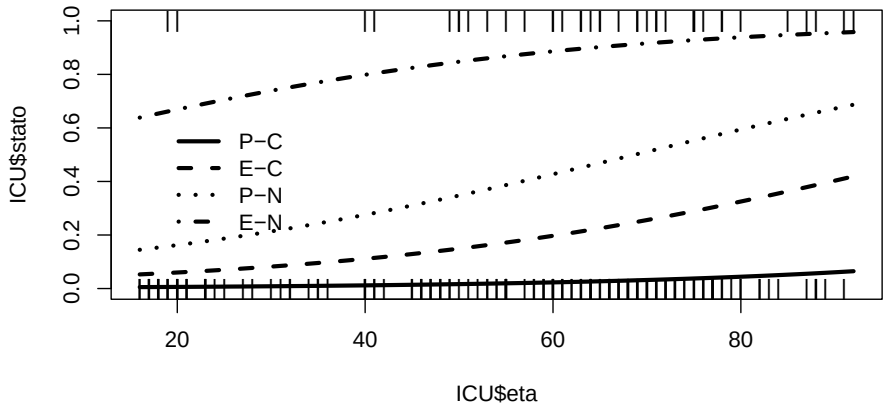
```
anova(ICU.fit0,ICU.fit,test="Chisq")

## Analysis of Deviance Table
##
## Model 1: stato ~ 1
## Model 2: stato ~ eta + causa + cosciente
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       199     200.16
## 2       196     145.31  3    54.85 7.39e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Il valore 54.83 è un valore estremamente grande secondo la distribuzione χ^2_3 (si trova nella coda destra della distribuzione): si rifiuta quindi l'ipotesi nulla di nullità di tutti i coefficienti $\beta_2, \beta_3, \beta_4$. La capacità del modello di prevedere il risultato finale del ricovero si può verificare con

```
table((predict(ICU.fit,type="response")>0.5),ICU$stato)

##
##           0   1
##   FALSE 158  27
##   TRUE    2  13
```

Infine, aggiungiamo al grafico le curve stimate

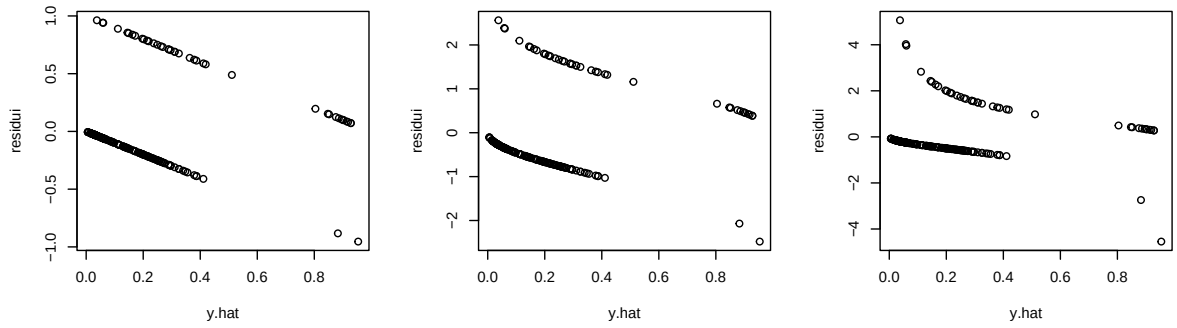
```
plot(ICU$eta,ICU$stato,pch="|",ylim=c(0,1))
curve(predict(ICU.fit,newdata=data.frame(eta=x,causa="P",cosciente="SI"),
              type="response"),add=TRUE,lty=1,lwd=3)
curve(predict(ICU.fit,newdata=data.frame(eta=x,causa="E",cosciente="SI"),
              type="response"),add=TRUE,lty=2,lwd=3)
curve(predict(ICU.fit,newdata=data.frame(eta=x,causa="P",cosciente="NO"),
              type="response"),add=TRUE,lty=3,lwd=3)
curve(predict(ICU.fit,newdata=data.frame(eta=x,causa="E",cosciente="NO"),
              type="response"),add=TRUE,lty=4,lwd=3)
legend(18,0.65,lty=c(1,2,3,4),lwd=3,legend=c("P-C","E-C","P-N","E-N"),bty="n")
```



2.1 Analisi dei residui

L'analisi dei residui con dati bernoulliani non offre indicazioni chiare.

```
y.hat <- predict(ICU.fit,type="response")
residui <- residuals(ICU.fit,type="response")
plot(y.hat,residui)
residui <- residuals(ICU.fit,type="deviance")
plot(y.hat,residui)
residui <- residuals(ICU.fit,type="pearson")
plot(y.hat,residui)
```



Possiamo però impiegare i *binned residuals* (proposti in Gelman e Hill, 2007, da cui è tratto il codice seguente), cioè suddividere le osservazioni in intervalli, rispetto a $\hat{y}$. Per ciascun intervallo si calcolano la media dei residui e la media di $\hat{y}$ e si rappresenta il punto sul grafico. Come riferimento, si può anche calcolare la deviazione standard della media dei residui e quindi rappresentare un intervallo di confidenza per ciascun *binned residual*. La funzione sottostante calcola le quantità rilevanti:

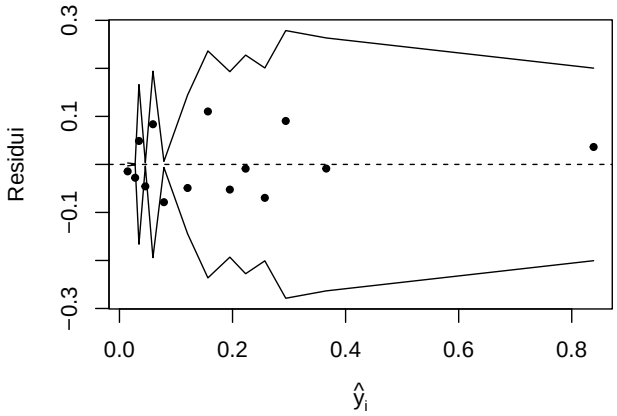
```
binned.resids <- function (x, y, nclass=sqrt(length(x))){
  ## calcolo gli indici delle osservazioni che delimiteranno
  ##   gli intervalli
  breaks.index <- floor(length(x)*(1:(nclass-1))/nclass)
  ## ottengo i limiti degli intervalli
  breaks <- c (-Inf, sort(x)[breaks.index], Inf)
  ## inizializzo le varibaili che conterranno i risultati
  output <- NULL
  xbreaks <- NULL
  ## per ciascun valore di x ottengo l'indice dell'intervallo
  ##  cui appartiene
  x.binned <- as.numeric (cut (x, breaks))
  ## il ciclo calcola le quantitÃ  rilevanti per ciascun intervallo
  for (i in 1:nclass){
    ## indici delle osservazioni nell'intervallo
    items <- (1:length(x))[x.binned==i]
    ## range di x nell'intervallo
    x.range <- range(x[items])
    ## media di x nell'intervallo
    xbar <- mean(x[items])
    ## media di y nell'intervallo
    ##  questa Ã¨ la media dei residui
    ybar <- mean(y[items])
    ## num osservazioni nell'intervallo
    n <- length(items)
    ## deviazione standard dei residui nell'intervallo
    sdev <- sd(y[items])
    ## output sarÃ  una matrice in cui l'i-ma riga Ã¨ riferita
    ##  all' i-mo intervallo
    output <- rbind (output, c(xbar, ybar, n, x.range, 2*sdev/sqrt(n)))
  }
  colnames (output) <- c ("xbar", "ybar", "n", "x.lo", "x.hi", "2se")
  return (list (binned=output, xbreaks=xbreaks))
}
```

Calcolati poi valori teorici e residui eseguiamo la funzione

```
y.hat <- predict(ICU.fit,type="response")
residui <- residuals(ICU.fit,type="response")
br <- binned.resids(y.hat,residui)
```

E infine disegnamo il grafico, la cui interpretazione è analoga a quella di un usuale grafico dei residui per il modello lineare. Va tenuto presente che il grafico dipende dalla scelta, arbitraria, del numero di intervalli, può essere opportuno rifarlo con scelte diverse.

```
br <- br$binned
plot(br[,1], br[,2],
     ylim=range(br[,2],br[,6],-br[,6]),
     xlab=expression(hat(y)[i]), ylab="Residui", pch=20)
abline (0,0, lty=2)
lines (br[,1], br[,6])
lines (br[,1], -br[,6])
```



3 Un esperimento clinico

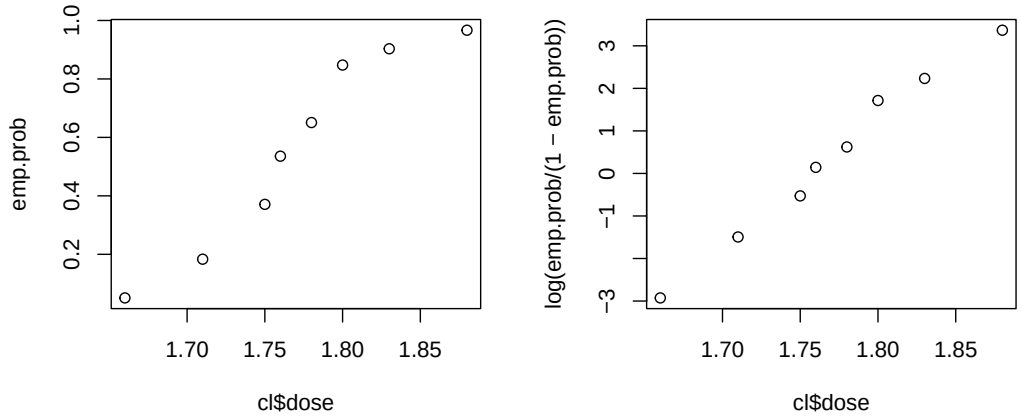
Consideriamo il seguente campione proveniente da un esperimento clinico.

```
cl <- read.table("clintrial.txt")
cl
```

```
##   number num.positive dose
## 1     59            3 1.66
## 2     60           11 1.71
## 3     62           23 1.75
## 4     56           30 1.76
## 5     63           41 1.78
## 6     59           50 1.80
## 7     62           56 1.83
## 8     60           58 1.88
```

A 481 pazienti affetti da una malattia è stato somministrato, in dosi diverse, un farmaco. Si è poi registrato il numero di pazienti che hanno risposto positivamente al farmaco (cioè sono guariti). Possiamo calcolare la probabilità di successo in corrispondenza ai diversi dosaggi, è piuttosto evidente che la probabilità cresce al crescere del dosaggio, e che la crescita è non lineare. Trasformando i risultati secondo la logistica, osserviamo che la relazione è sostanzialmente lineare.

```
emp.prob <- cl$num.positive/cl$number
plot(emp.prob~cl$dose)
plot(log(emp.prob/(1-emp.prob))~cl$dose)
```



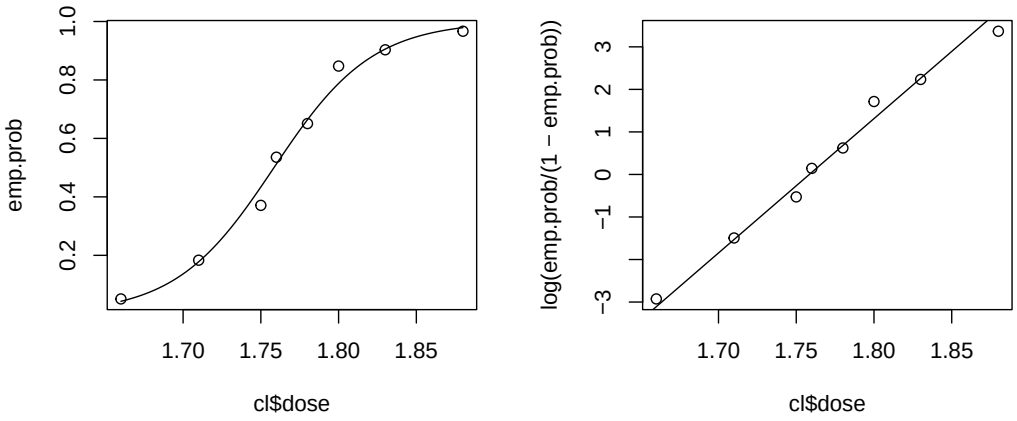
Stimiamo dunque il modello logistico, che conferma l'effetto positivo della dose.

```
fit.logit <- glm(cbind(cl$num.positive,cl$number-cl$num.positive)~dose,
              data=cl,family=binomial)
fitsum.logit <- summary(fit.logit)
fitsum.logit

##
## Call:
## glm(formula = cbind(cl$num.positive, cl$number - cl$num.positive) ~
##     dose, family = binomial, data = cl)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.00540  -0.31925   0.02226   0.31223   1.16137
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -55.545      5.301  -10.48   <2e-16 ***
## dose          31.588      3.003   10.52   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 229.6492  on 7  degrees of freedom
```

```
## Residual deviance:   3.0109  on 6  degrees of freedom
## AIC: 37.532
##
## Number of Fisher Scoring iterations: 4
```

Possiamo visualizzare l'adattamento del modello ai dati aggiungendo ai grafici precedenti, nelle due scale, il modello stimato.

```
plot(emp.prob~cl$dose)
curve(predict(fit.logit,newdata=data.frame(dose=x),type="response"),add=TRUE)
plot(log(emp.prob/(1-emp.prob))~cl$dose)
abline(coef(fit.logit))
```



Può essere di interesse, nel contesto di un esperimento clinico, calcolare la *dose efficace*, dove con ciò si intende la dose che implica una probabilità di guarigione p: EDp. Per ottenere EDp dobbiamo risolvere, in x, l'equazione

$$\log\left(\frac{p}{1-p}\right) = \beta_1 + \beta_2 x$$

da cui

$$x = \frac{1}{\beta_2}\left(\log\left(\frac{p}{1-p}\right) - \beta_1\right)$$

Si ottiene una stima sostituendo $\hat{\beta}_j$ a β_j.
Consideriamo il caso particolare $p = 0.5$, la $ED_{0.5}$ è

$$ED_{0.5} = -\frac{\beta_1}{\beta_2},$$

e di conseguenza la stima è

$$\widehat{ED_{0.5}} = -\frac{\hat{\beta}_1}{\hat{\beta}_2} = -\frac{-55.545}{31.588} = 1.7584$$

La varianza dello stimatore $\widehat{ED_{0.5}}$ può essere ottenuta con il metodo delta, per una generica funzione $g(\hat{\boldsymbol{\beta}})$ la varianza è data approssimativamente da

$$\text{var}(g(\hat{\boldsymbol{\beta}})) \approx g'(\hat{\boldsymbol{\beta}})^T \text{var}(\hat{\boldsymbol{\beta}}) g'(\hat{\boldsymbol{\beta}})$$

nel caso specifico il gradiente della funzione g, g' è dato da

$$g'(\hat{\boldsymbol{\beta}}) = \begin{bmatrix} -1/\hat{\beta}_2 \\ \hat{\beta}_1/\hat{\beta}_2^2 \end{bmatrix}$$

e possiamo quindi calcolare la varianza di $\widehat{ED_{0.5}}$ con

```
grad.g <- c(-1/coef(fit.logit)[2],coef(fit.logit)[1]/coef(fit.logit)[2]^2)
var.ED50 <- grad.g %*% fitsum.logit$cov.unscaled %*% grad.g
var.ED50

##               [,1]
## [1,] 1.454641e-05

sd.ED50 <- sqrt(var.ED50)
sd.ED50

##              [,1]
## [1,] 0.003813975
```

Dove si noti che la matrice di varianza dello stimatore $\hat{\boldsymbol{\beta}}$ si ottiene da `summary`

```
fitsum.logit$cov.unscaled

##             (Intercept)       dose
## (Intercept)    28.10351 -15.913672
## dose          -15.91367   9.015676
```

Possiamo dunque calcolare un intervallo di confidenza al 95% per $ED_{0.5}$ come

```
inf<- -coef(fit.logit)[1]/coef(fit.logit)[2] - qnorm(0.975)*sd.ED50
sup<- -coef(fit.logit)[1]/coef(fit.logit)[2] + qnorm(0.975)*sd.ED50
c(inf,sup)

## [1] 1.750957 1.765908
```

Notiamo che nel pacchetto `MASS` c'è una funzione per il calcolo della dose

```
library(MASS)
dose.p(fit.logit,p=c(0.5,0.95))

##                Dose          SE
## p = 0.50: 1.758432 0.003813975
## p = 0.95: 1.851646 0.009044579
```

La devianza residua del modello è

```
fitsum.logit$deviance

## [1] 3.010939
```

Data la natura del campione, in particolare il fatto che le dosi siano fissate dallo sperimentatore, e quindi sia ragionevole assumere che le numerosità di ciascuna classe tendano a infinito, è rilevante confrontare la devianza residua con la distribuzione χ^2, in particolare χ^2_6: il valore qui ottenuto è coerente con le assunzioni.

4 Credit scoring

4.1 Illustrazione dei dati e obiettivo dell'analisi

Quando le banche concedono un credito controllano la "solvibilità" o l'"affidabilità" del cliente, cioè la sua capacità di rifondere il credito. La "solvibilità" potrebbe essere valutata tramite un modello statistico che prova appunto a valutare la probabilità che un credito venga ripagato utilizzando come predittori alcune caratteristiche personali o economiche relative a chi chiede il prestito. Modelli di regressione per variabili dipendenti binarie sono adatti allo scopo.
Considereremo un dataset relativi a $n = 1000$ crediti concessi da una banca tedesca. A ogni cliente è associata la seguente variabile binaria che sarà nel caso in questione la variabile risposta:

$$\begin{array}{ll} y = 0 & \text{il cliente ha ripagato il prestito} \\ y = 1 & \text{il cliente non ha ripagato il prestito} \end{array}$$

Per ogni cliente (credito concesso) sono state osservate inoltre le seguenti altre caratteristiche:

Variable	description
acc	non ha conto corrente; ha conto corrente in rosso; ho conto corrente in attivo
duration	tempo per la restituzione del prestito
amount	ammontare in K-euro
moral	comportamento in precedenti prestiti 1 = good
intuse	motivo della richiesta del credito 1 = privato 0 = affari

Le informazioni sono in un file di testo `credit1.txt`, e possono esser lette col comando

```
Credit <- read.table("credit1.txt", header=TRUE)
```

I dati sono ora organizzati in un data frame con 1000 righe e 7 colonne. Diamo un'occhiata ai primi 5 records:

```
Credit[1:5,]

##   y  acc duration    amount moral intuse
## 1 0   no       24 1.5118900     1      1
## 2 0  bad       12 0.7280797     1      1
## 3 0 good       18 1.0486600     1      1
## 4 0 good       12 2.3902900     1      1
## 5 0 good       24 1.5226270     1      0
```

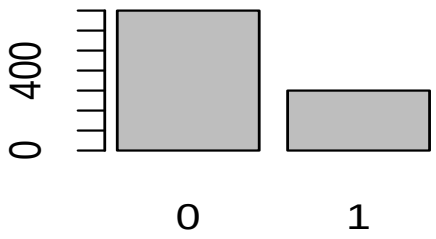
e rendiamo le variabili direttamente disponibili per il seguito

```
attach(Credit)
```

4.2 Un modello di regressione logistica

I dati sono ora disponibili nel workspace di R.
Possiamo in primo luogo avwere un'idea delle caratteristiche delle principali variabili. Nel campione quanti non hanno ripagato il credito?

```
table(y)

## y
##   0   1
## 700 300

barplot(table(y))
```



Il 30% non ha restituito il prestito.
Possiamo ora provare a stimare un primo modello di regressione logistica

```r
mod1 <- glm(y~ acc+duration+amount+moral+intuse,family=binomial(link=logit))
summary(mod1)
```

```
##
## Call:
## glm(formula = y ~ acc + duration + amount + moral + intuse, family = binomial(link = logit
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8876  -0.8440  -0.4628   0.9629   2.3620
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.284402   0.302579  -0.940 0.347255
## accgood     -1.337748   0.201127  -6.651 2.91e-11 ***
## accno        0.617659   0.174728   3.535 0.000408 ***
## duration     0.033233   0.007746   4.290 1.78e-05 ***
## amount       0.045875   0.064092   0.716 0.474134
## moral       -0.986066   0.250891  -3.930 8.49e-05 ***
## intuse      -0.425536   0.158272  -2.689 0.007174 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1221.7  on 999  degrees of freedom
## Residual deviance: 1029.8  on 993  degrees of freedom
## AIC: 1043.8
##
## Number of Fisher Scoring iterations: 4
```

I p-values associati a ciascun coefficiente rivelano che per tutte le variabili non si esclude un effetto significativo fatta eccezione per *amount*.
I segni dei coefficienti stimati sono tutti con il segno che ci si attendeva. Ai fini dell'interpretazione si ricordi che si sta valutanmdo l'effetto delle covariate sulla probabilità di non essere un cliente affidabile (di essere un cattivo pagatore quindi).
Non avere un conto corrente aumenta la probabilità di essere insolvente rispetto a coloro che hanno il conto corrente in rosso. Mentre avere un conto corrente in attivo decresce in modo significativo la probabilità di essere insolvente. L'effetto sugli odds può essere valutato calcolando

```r
exp(mod1$coefficients[3])
```

```
##    accno
## 1.854582
```

L'odds di essere insolvente, a parità delle altre condizioni, per coloro che hanno un conto corrente in rosso è 2.5 volte superiore rispetto allo stesso odds per coloro con un conto corrente in attivo. Il coefficiente associato con *duration* è significativamente diverso da 0 e ha segno positivo. Questo significa che più lungo è il periodo previsto per rendere il prestito maggiore è la probabilità di essere insolvente.

4.3 Un modello più complesso

Il modello può essere complicato valutando se c'è un effetto non lineare delle due variabili quantitative presenti fra le covariate.
Calcoliamo ad esempio il quadrato delle due variabili:

```
amountsq <- amount^2
durationsq <- duration^2
```

e ora proviamo il nuovo modello

```
mod2 <- glm(y~acc+duration+durationsq+amount+amountsq+moral+intuse,family=binomial(link=logit
summary(mod2)
```

```
##
## Call:
## glm(formula = y ~ acc + duration + durationsq + amount + amountsq +
##     moral + intuse, family = binomial(link = logit))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4157  -0.8124  -0.4719   0.9585   2.6326
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4877826  0.3896495  -1.252 0.210625
## accgood     -1.3374022  0.2024364  -6.607 3.93e-11 ***
## accno        0.6178347  0.1761733   3.507 0.000453 ***
## duration     0.0921909  0.0252941   3.645 0.000268 ***
## durationsq  -0.0009094  0.0004133  -2.200 0.027781 *
## amount      -0.5165866  0.1926907  -2.681 0.007342 **
## amountsq     0.0878646  0.0285982   3.072 0.002124 **
## moral       -0.9953315  0.2551618  -3.901 9.59e-05 ***
## intuse      -0.4039789  0.1601534  -2.522 0.011654 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 1221.7  on 999  degrees of freedom
## Residual deviance: 1017.4  on 991  degrees of freedom
## AIC: 1035.4
##
## Number of Fisher Scoring iterations: 4
```

Il nuovo modello sembra migliorare quello precedente (ad esempio si può fare il confronto fra le devianze dei due modelli).

4.4 Prevedere l'insolvenza

Un possibile uso del modello è di prevedere l'insolvenza per un nuovo cliente che vuole chiedere un prestito per un dato ammontare dalla banca. Il modello stimato può fornire una guida per decidere se concedere o meno il prestito se la probabilità di insolvenza è giudicata troppo alta.
In questo caso si assume che nel mommento in cui la banca deve decidere dispone dei dati sulle covariate per il cliente in questione.
Si consideri ad esempio un cliente con le seguenti caratteristiche: non ha conto corrente, il termine per ripagare il debito è 36 mesi, l'ammontare richiesto è 10000 euro, il comporatemento nei precedenti pagamenti non era buono, il motivo della richiesta è affari.
Quanto rischioso è concedere il prestito?
La probabilità di insolvenza prevista dal modello è:

```
mio=data.frame(acc="no",duration=36,durationsq=36^2,amount=10,amountsq=100,moral=0,intuse=0)
predict(mod2,newdata=mio, type="response")
```

```
##         1
## 0.9972431
```

Dare il prestito è estremamente rischioso! Si potrebbe facilmente provare a stimare un modello probit equivalente. Tuttavia ci si aspetta che i risultati saranno molto simili a quelli già ottenuti.