Principali tecniche di regressione con R

Versione 0.3 11 settembre 2006

Vito Ricci
vito_ricci@yahoo.com

Indice

1.0 Premessa

L'analisi della regressione, nelle sue varie e multiformi sfaccettature, è una delle tecniche statistiche maggiormente utilizzate. Il presente lavoro, senza avere alcuna pretesa di esaustività, vuole fornire una trattazione soprattutto pratica di questa metodologia, anche se alcuni riferimenti e accenni alla teoria non mancheranno, attraverso l'impiego del software statistico R[1].
Si cercheranno di affrontare le principali tipologie di regressioni (parecchia attenzione verrà data alla regressione lineare multipla), i metodi di stima (OLS, GLS, WLS, TSLS), la diagnostica, la verifica dei requisiti per l'applicazione del modello. Si affronterà la generalizzazione del modello lineare (GLM, *generalized linear model*) per la trattazione di variabili dicotomiche e di conteggio (regressione logistica e regressione di Poisson), così come la regressione non lineare, la regressione robusta (*resistant* e *robust regression*), *la ridge reggresion*, la regressione quantilica (*quantile regression*), i modelli lineari con effetti misti (*linear mixed effects model*), la regressione di Cox, la regressione Tobit. Verranno presentati degli esempi concreti con la trattazione dei comandi e dei packages di R utili a risolvere i problemi di calcolo relativi alle varie tecniche richiamate in precedenza.
Ai fini della comprensione del presente lavoro si richiede la conoscenza di tecniche statistiche abbastanza avanzate e una buona padronanza e conoscenza del software R.

2.0 Introduzione

L'analisi della regressione è usata per spiegare la relazione esistente tra una variabile Y (continua) detta variabile risposta, oppure output o variabile dipendente, e una o più variabili dette covariate, variabili esplicative, indipendenti, oppure repressori, predittori o variabili di input (X_1, X_2,… X_k). In termini di funzione abbiamo:

$$Y=f(X_1, X_2,\ldots X_k)+\varepsilon$$

che indica l'esistenza di un legame funzionale in media tra la variabile dipendente e i regressori, rappresentato dalla componente $f(X_1, X_2,\ldots X_k)$ e alla quale suole dare il nome di componente sistematica. A questa componente va ad aggiungersi un'altra denominata accidentale, casuale, erronea. Mentre la prima rappresenta la parte della variabile risposta spiegata dai predittori, la seconda componente rappresenta quella parte di variabilità della risposta che non può ricondursi a fattori sistematici oppure facilmente individuabili, ma dovuti al caso e, più in generale, a cause diverse non prese in considerazione nel modello regressivo. Il legame funzionale teoricamente può essere di qualsiasi tipo, tuttavia nella pratica si preferisce utilizzare una funzione di tipo lineare e pertanto si parla regressione lineare multipla o modello lineare che assume la seguente formulazione:

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_k X_k + \varepsilon$$

ove β_0 è detto termine noto, mentre $\beta_1,\ldots,\beta_k$ sono detti coefficienti di regressione e, insieme alla varianza dell'errore, sono i parametri del modello da stimare sulla base delle osservazioni campionarie. Diversi modelli, in apparenza non lineari, possono essere linearizzati tramite opportune trasformazioni di variabili. Ad esempio, il modello moltiplicativo:

$$Y = \beta_0 X_1^{\beta_1} \ldots X_k^{\beta_k} \varepsilon$$

può essere facilmente trasformato nel modello lineare prendendo i logaritmi di ambo i membri.
Si parla di regressione polinomiale quando i regressori nel modello figurano non solo con grado pari ad uno, ma anche con grado maggiore. Tuttavia il modello continua a rimanere lineare nei parametri. Quello che segue è un modello di regressione parabolica con due soli regressori:

$$Y = \beta_0 + \beta_1 X_1 + \beta_{12} X_1^2 + \beta_{13} X_1 X_2 + \beta_2 X_2 + \beta_{21} X_2^2$$

[1] R Development Core Team (2006). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

che figurano con il primo e il secondo grado; si è preso in considerazione anche il fattore di interazione tra le variabili esplicative (X_1X_2). Si parla di regressione non lineare quando i parametri risultano comparire in forma diversa da quella lineare.
Quando la variabile risposta non è di tipo continuo si ha una generalizzazione del modello lineare (GLM) del quale ci occuperemo di seguito che prende in esame il caso di risposte di tipo dicotomico (regressione logistica) o di conteggio (regressione di Poisson).

3.0 Il modello lineare

3.1 Richiami

Nel modello di regressione lineare multipla la variabile dipendente Y è spiegata da k regressori[2]. Per ciascuna di queste variabili sono disponibili n osservazioni:

$$y_1 = \beta_0 + \beta_1 x_{11} + ... + \beta_k x_{1k} + \varepsilon_1$$
…
$$y_i = \beta_0 + \beta_1 x_{i1} + ... + \beta_k x_{ik} + \varepsilon_i$$
…
$$y_n = \beta_0 + \beta_1 x_{n1} + ... + \beta_k x_{nk} + \varepsilon_n$$

Se utilizziamo la forma matriciale:

$$y = \begin{pmatrix} y_1 \\ ... \\ y_n \end{pmatrix} \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \\ ... \\ \beta_k \end{pmatrix} \varepsilon = \begin{pmatrix} \varepsilon_1 \\ ... \\ \varepsilon_n \end{pmatrix} X = \begin{pmatrix} 1 \; x_{11} ... x_{1k} \\ ... \\ 1 \; x_{n1} ... x_{nk} \end{pmatrix}$$

il modello lineare può esprimersi compattamente:

$$y = X\beta + \varepsilon$$

Di solito si fanno delle ipotesi di base relativamente agli errori (ε) che sintetizziamo di seguito:

$$\varepsilon \sim \mathrm{N_n}(0, \sigma^2)$$
$$E(\varepsilon) = 0$$
$$E(\varepsilon\varepsilon') = \sigma^2 I_n$$

ossia la distribuzione degli errori è di tipo normale multivariata, con media nulla e varianza costante (omoscedasticità); inoltre gli errori sono incorrelati a due a due. Queste ipotesi vanno opportunamente verificate tramite test statistici (test di specificazione del modello). Da queste ipotesi deriva che:

$$E(y) = X\beta$$
$$Cov(y) = \sigma^2 I_n$$

Per la stima dei parametri si sceglie il metodo dei minimi quadrati (OLS, Ordinary Least Squares) minimizzando la somma dei quadrati degli errori:

$$\varepsilon'\varepsilon = (y - X\beta)'(y - X\beta)$$

[2] Si veda F. DEL VECCHIO, *Analisi statistica di dati multidimensionali*, 1992, pag. 167 e segg. e A.POLLICE, *Dispense di statistica multivariata* , 2005, cap. 4, pag. 41 e segg.

da cui si ricava:

$$b = \hat{\beta} = (X'X)^{-1}X'y$$

che è uno stimatore BLUE (best, linear, umbiased, estimator) di β. Sinteticamente si riportano altri risultati utili ai fini della nostra trattazione:

$V(b) = \sigma^2(X'X)^{-1}$ (matrice delle varianze e covarianze degli stimatori)
$H = X(X'X)^{-1}X'$ (matrice di proiezione)
$e = (I_n - H)y$ (residui)
$RSS = e'e = \sum_{i=1}^{n} e_i^2 = y'(I_n - H)y$ (devianza residua)
$\hat{\sigma}^2 = RSS/n-k$ (stima della varianza dell'errore)
$Dev(Y) = (y-\bar{y})'(y-\bar{y}) = \sum_{i=1}^{n}(y_i - \bar{y})^2$ (devianza della variabile risposta)
$R^2 = 1 - RSS/Dev(Y)$ (indice di determinazione)
$R_{adj}^{\ 2} = 1 - \frac{RSS/(n-k)}{Dev(Y)/(n-1)}$ (indice di determinazione aggiustato)
$\hat{y} = Hy$ (valori stimato con il modello)

3.2 Stima dei parametri del modello

Fatta questa necessaria premessa di alcuni richiamati teorici, la stima dei parametri di un modello di regressione multipla con il software R avviene con il comando `lm()`[3]:

```
lm(formula, data, subset, weights, na.action,method = "qr", model = TRUE,
x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE, contrasts = NULL,
offset, ...)
```

nella quale i principali argomenti sono `formula` che rappresenta la descrizione simbolica del modello da stimare e `data` che indica il nome del dataframe opzionale nel quale sono presenti le variabili che figurano nel modello. Per spiegare l'argomento `formula` supponiamo che y è una variabile numerica risposta e x_1, x_2, . . .x_n sono i repressori. Le seguenti formule specificano alcuni modelli statistici mettendo in relazione la risposta (nella parte sinistra) con le variabili esplicative (nella parte destra):

`y~`x_1`+`x_2`+...+`x_n
`y~1+`x_1`+`x_2`+...+`x_n
entrambi indicano un modello di regressione lineare multipla di y su x_1, x_2, . . .x_n; il primo ha il termine noto (intercetta) implicito, nel secondo, invece, questo è esplicitato;

`y~0+`x_1`+`x_2`+...+`x_n
`y~-1+`x_1`+`x_2`+...+`x_n
`y~`x_1`+`x_2`+...+`x_n`-1`
modello di regressione lineare multipla di y su x_1, x_2, . . .x_n con termine noto (intercetta) uguale a zero;

`log(y)~`x_1`+`x_2`+...+`x_n
regressione lineare multipla della trasformata logaritmica di y su x_1, x_2, . . .x_n;

`y~poly(`x_1`,2)`

[3] Si può seguire anche la via del calcolo matriciale e per questa soluzione si rinvia a J. J. FARAWAY, *Practical Regression and Anova using R*, 2002, pag. 23 e segg.

`y~1 + x1 + I(x1^2)`
regressione polinomiale di secondo grado; con l'espressione `poly(x,n)` si possono stimare regressioni polinomiali di grado n;

`y~I(1/x1)`
modello di regressione di y sul reciproco di x_1; più in generale nell'operatore `I()` si può specificare una qualsiasi trasformata delle variabili dipendenti.

`y~x1+x2+...+xn+ x1:x2`
modello di regressione lineare multipla di y su x_1, x_2, . . .x_n che tiene conto anche del termine di interazione tra x_1 e x_2;

`y~x1*x2*...*xn`
modello di regressione lineare multipla "completo" di y su x_1, x_2, . . .x_n che tiene conto di tutte le possibili interazioni tra i regressori;

Oltre ai regressori di tipo quantitativo, si possono introdurre anche variabili esplicative di tipo qualitativo (fattori) e possono effettuarsi analisi della varianza (ANOVA) e della covarianza (ANCOVA).

Come applicazione utilizziamo il seguente esempio:

```
mortalita
```

```
                     mortal Calorie HS popphys popnurs
Afghanistan             206    2304  1   15770   24430
Algeria                 154    1683  5    8590   11770
...
Zambia                  121    2042  3   11380    5820
Zimbabwe                103    2044  5    8010     990
```

nel quale si vuole studiare il legame tra la variabile risposta `mortal` ed alcune variabili esplicative.

stimiamo il modello lineare con `lm()`

```
fm<-lm(mortal~ Calorie+ HS+ popphys+ popnurs, data=mortalita)
```

e visualizziamo il risultato della regressione multipla con `summary()`:

```
summary(fm)
```

```
Call:
lm(formula = mortal ~ Calorie + HS + popphys + popnurs, data = mortalita)

Residuals:
     Min       1Q   Median       3Q      Max
-59.4418 -18.3817   0.4307  15.4310  77.1677

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.891e+02  2.015e+01   9.384 3.75e-15 ***
Calorie     -2.957e-02  9.387e-03  -3.150  0.00219 **
HS          -1.231e+00  2.192e-01  -5.616 1.98e-07 ***
popphys      6.022e-04  1.950e-04   3.088  0.00265 **
popnurs      1.293e-03  5.956e-04   2.171  0.03248 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 27.81 on 94 degrees of freedom
Multiple R-Squared: 0.7832,     Adjusted R-squared: 0.774
F-statistic: 84.89 on 4 and 94 DF,  p-value: < 2.2e-16
```

per ottenere la lista degli attributi dell'oggetto:

```
attributes(fm)
$names
 [1] "coefficients"  "residuals"     "effects"       "rank"
 [5] "fitted.values" "assign"        "qr"            "df.residual"
 [9] "xlevels"       "call"          "terms"         "model"

$class
[1] "lm"
```

Dall'oggetto di classe `lm`, che abbiamo chiamato `fm` (che sta per fitted model), contente i principali risultati della regressione stimata (coefficienti e standard error, test t e p-value, R^2, test F e p-value) possiamo estrarre i singoli valori avvalendoci di alcune funzioni:

```
coef(fm) ## vettore dei coefficienti di regressione
fm$coef ## stesso risultato

  (Intercept)       Calorie            HS       popphys       popnurs
189.097165167  -0.029567944  -1.231002661   0.000602156   0.001292906

deviance(fm) ## devianza dei residui
[1] 72682.47

formula(fm) ## formula del modello
mortal ~ Calorie + HS + popphys + popnurs

plot(fm) ## traccia quattro grafici utili per la diagnostica

residuals(fm) ## vettore dei residui del modello
fm$residuals ##

fitted(fm)## valori della risposta stimati dal modello
fm$fitted ## stesso risultato

X<-model.matrix(fm) ## matrice X del modello di regressione
X
                     (Intercept) Calorie HS popphys popnurs
Afghanistan                    1    2304  1   15770   24430
Algeria                        1    1683  5    8590   11770
...
Yugoslavia                     1    3244 59    1200     850
Zambia                         1    2042  3   11380    5820
Zimbabwe                       1    2044  5    8010     990
attr(,"assign")
[1] 0 1 2 3 4

summary(fm, correlation=T)$correlation ## correlazioni tra i coefficienti
di regressione

            (Intercept)     Calorie         HS    popphys     popnurs
(Intercept)   1.0000000 -0.95279301  0.5128989 -0.2680213 -0.09570829
Calorie      -0.9527930  1.00000000 -0.7179157  0.1049791 -0.01240151
```

```
HS            0.5128989 -0.71791574  1.0000000  0.2339413  0.17654333
popphys      -0.2680213  0.10497913  0.2339413  1.0000000 -0.29154297
popnurs      -0.0957083 -0.01240151  0.1765433 -0.2915430  1.00000000

vcov(fm)## matrice delle varianze e covarianze dei coefficienti di
regressione

             (Intercept)       Calorie            HS       popphys
(Intercept) 406.033256016 -1.802139e-01  2.265392e+00 -1.052990e-03
Calorie      -0.180213863  8.810847e-05 -1.477111e-03  1.921260e-07
HS            2.265392038 -1.477111e-03  4.804648e-02  9.997976e-06
popphys      -0.001052990  1.921260e-07  9.997976e-06  3.801445e-08
popnurs      -0.001148713 -6.933692e-08  2.304959e-05 -3.385775e-08
                 popnurs
(Intercept) -1.148713e-03
Calorie     -6.933692e-08
HS           2.304959e-05
popphys     -3.385775e-08
popnurs      3.547826e-07
```

La maggior parte di questi comandi o metodi è applicabile, come vedremo di seguito, anche ad oggetti di altre classi ottenuti come stima di modelli di regressione diversi da quella lineare (GLM, regressione quantilica, PLS, etc.).

3.3 Test di specificazione

Dopo aver stimato il modello di regressione è necessario verificare che siano valide le ipotesi di base che abbiamo esposto in precedenza tramite opportuni test statistici.
In primo luogo verifichiamo che la media degli errori non sia significativamente diversa da zero attuando il test t di Student:

```
residui<-residuals(fm) ##vettore dei residui

t.test(residui)

 One Sample t-test

data: residui
t = 0, df = 98, p-value = 1
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 -5.431605 5.431605
sample estimates:
  mean of x
5.657652e-16
```

Successivamente verifichiamo la normalità della distribuzione degli errori con il test di Shapiro-Wilk[4]:

```
shapiro<-shapiro.test(residui)
shapiro

    Shapiro-Wilk normality test
```

[4] Per ulteriori test di normalità si veda il package `nortest`:
http://dssm.unipa.it/CRAN/src/contrib/Descriptions/nortest.html e V. RICCI, *Rappresentazione analitica delle distribuzioni statistiche*, 2005, pagg. 20-23

```
data: residui
W = 0.9839, p-value = 0.2724
```

graficamente si può usare il QQ-plot con il comando `qqnorm()` applicato ai residui standardizzati: (Fig. 1):

```
qqnorm(scale(residui))
abline(0,1)
```

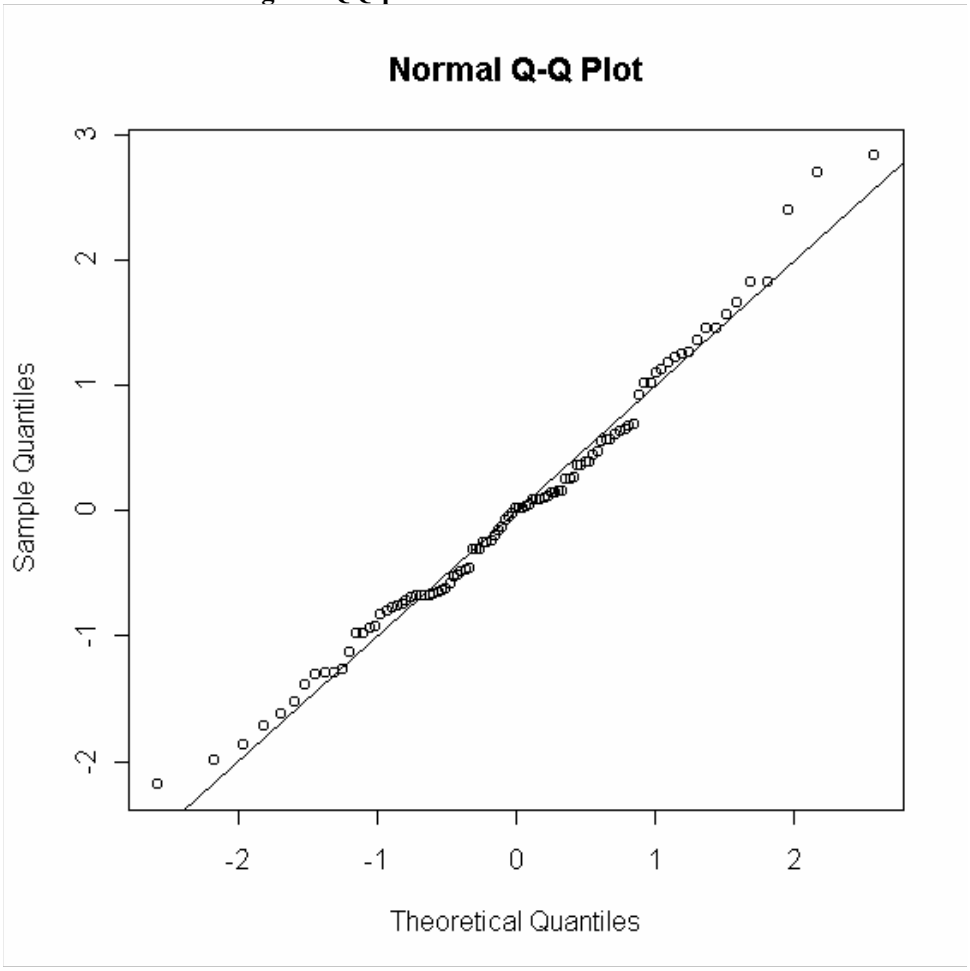
Proseguiamo con il verificare l'omoschedasticità dei residui utilizzando il test di Breusch-Pagan e l'assenza di correlazione seriale tramite il test di Durbin-Watson. Entrambi i test sono largamente impiegati nelle analisi econometriche. Occorre caricare il package `lmtest`[5] che contiene i comandi `bptest()`e `dwtest()`:

```
library(lmtest)
modello<-formula(fm)## memorizziamo la formula del modello in un oggetto
per facilità di manipolazione
testbp<-bptest(modello,data=mortalita) )## test di Breusch-Pagan
testbp

    studentized Breusch-Pagan test

data: modello
BP = 2.7408, df = 4, p-value = 0.6021


dw<-dwtest(modello,data=mortalita) ## test di Durbin-Watson
dw

    Durbin-Watson test

data: modello
DW = 2.2949, p-value = 0.9275
alternative hypothesis: true autocorrelation is greater than 0
```

come si può vedere per effettuare i due test è necessario specificare la formula del modello di regressione e il dataframe in cui sono contenti i dati.
Tutti i test di specificazione del modello hanno dato esito positivo, possiamo affermare che le ipotesi alla base del modello di regressione OLS sono valide. Se anche uno solo dei test dà esito negativo (non normalità dei residui, eteroschedasticità, correlazione seriale) il metodo di stima OLS non va più bene e bisogna optare per altre soluzioni che affronteremo nei prossimi paragrafi.

[5] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/lmtest.html

Fig. 1 – QQ-plot dei residui standardizzati



3.4 Intervalli di confidenza per i coefficienti di regressione

Per costruire degli intervalli di confidenza si utilizza il comando `confint()`: occorre specificare come argomenti l'oggetto di classe `lm` che contiene il modello di regressione stimato e il livello di confidenza (per default vengono forniti gli intervalli con confidenza pari al 95%).

```
confint(fm) ## intervalli di confidenza al 95%
                    2.5 %        97.5 %
(Intercept)  1.490883e+02  2.291060e+02
Calorie     -4.820528e-02 -1.093061e-02
HS          -1.666220e+00 -7.957857e-01
popphys      2.150326e-04  9.892793e-04
popnurs      1.102560e-04  2.475555e-03

confint(fm, level=0.99) ## intervalli di confidenza al 99%
                    0.5 %        99.5 %
(Intercept)  1.361191e+02 242.075191789
Calorie     -5.424673e-02  -0.004889157
HS          -1.807299e+00  -0.654706440
popphys      8.954334e-05   0.001114769
popnurs     -2.731100e-04   0.002858921
```

3.5 Verifica di ipotesi

La verifica dell'ipotesi che i coefficienti di regressione siano significativamente diversi da zero, ovvero che esista una relazione lineare tra la variabile risposta e il repressore dovuta a fattori sistematici e non casuali, è ottenuta nel momento che viene stimato il modello e si manda in stampa la sintesi. Infatti:

```
summary(fm)

Call:
lm(formula = mortal ~ Calorie + HS + popphys + popnurs, data = mortalita)

Residuals:
     Min       1Q   Median       3Q      Max
-59.4418 -18.3817   0.4307  15.4310  77.1677

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.891e+02  2.015e+01   9.384 3.75e-15 ***
Calorie     -2.957e-02  9.387e-03  -3.150  0.00219 **
HS          -1.231e+00  2.192e-01  -5.616 1.98e-07 ***
popphys      6.022e-04  1.950e-04   3.088  0.00265 **
popnurs      1.293e-03  5.956e-04   2.171  0.03248 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 27.81 on 94 degrees of freedom
Multiple R-Squared: 0.7832,     Adjusted R-squared: 0.774
F-statistic: 84.89 on 4 and 94 DF,  p-value: < 2.2e-16
```

Per ciascun coefficiente di regressione viene fornito il valore del test di Student $t = \frac{\hat{\beta}_i}{se(\hat{\beta}_i)}$ (sotto l'ipotesi $\beta=0$), il relativo p-value e il grado di significatività espresso dal numero degli asterischi. Analogo risultato si ottiene con il comando `coeftest()` del package `lmtest`:

```
library(lmtest)
coeftest(fm)

t test of coefficients of "lm" object 'fm':

             Estimate  Std. Error t value  Pr(>|t|)
(Intercept)  1.8910e+02  2.0150e+01  9.3844 3.745e-15 ***
Calorie     -2.9568e-02  9.3866e-03 -3.1500  0.002190 **
HS          -1.2310e+00  2.1920e-01 -5.6160 1.977e-07 ***
popphys      6.0216e-04  1.9497e-04  3.0884  0.002646 **
popnurs      1.2929e-03  5.9564e-04  2.1706  0.032478 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Della regressione stimata viene fornita la statistica $F = \frac{R^2/n}{(1-R^2)/(n-k-1)}$, i gradi di libertà e relativo p-value che esprimono una valutazione della significatività complessiva del modello di regressione.
Se vogliamo effettuare la verifica di ipotesi generica sui coefficienti di regressione abbiamo:

$H_0 : \beta_i = \beta_{0i}$

$H_1 : \beta_i \neq \beta_{0i}$

con il relativo test $t = \frac{\hat{\beta}_i - \beta_{0i}}{se(\hat{\beta}_i)}$

```
b<-coef(fm)[2:5]## vettore delle stime dei coeff. di regressione
b
 Calorie      HS   popphys   popnurs
-0.029567944 -1.231002661 0.000602156 0.001292906

b0<-c(0.03, -1.30, 0.01, 0.01) ## vettore beta ipotesi nulla
b0
[1] 0.03 -1.30 0.01 0.01

sdb<-summary(fm)$coefficients[,2] ## vettore standard error delle stime
dei coeff. di regressione
 sdb
(Intercept)   Calorie      HS   popphys   popnurs
2.015027e+01 9.386611e-03 2.191951e-01 1.949729e-04 5.956363e-04

tstat<-(b-b0)/sdb[2:5] ## vettore test t
tstat
        Calorie     HS   popphys   popnurs
 -6.3460542  0.3147760 -48.2007602 -14.6181401

pval<-2*pt(tstat, 94) ## vettore dei p-value
pval
   Calorie      HS   popphys   popnurs
7.715191e-09 1.246371e+00 4.395536e-68 6.124862e-26
```

Per verifiche di ipotesi su restrizioni lineari[6] di coefficienti di regressione si può usare il comando `linear.hypothesis()` contenuto nel package `car`[7]:

```
library(car)
a<-c(1,-1,0,0,1)
linear.hypothesis(fm, hypothesis.matrix=matrix(a,1,5),rsh=1)
Linear hypothesis test

Model 1: mortal ~ Calorie + HS + popphys + popnurs
Model 2: restricted model
 Res.Df  RSS Df Sum of Sq   F  Pr(>F)
1   94 72682
2   95 140739 -1  -68056 88.017 3.793e-15 ***
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.6 Intervalli di confidenza per valori stimati della variabile risposta e intervalli di previsione

Per ottenere degli intervalli di confidenza per i valori stimati della variabile dipendente oppure degli intervalli di previsione si può usare il comando `predict()` specificando l'oggetto di classe `lm` e come argomenti principali il tipo di intervallo (`interval = "confidence"` oppure `"prediction"`) e il livello di confidenza (`level = 0.95` per default). Se si vogliono ottenere degli intervalli di previsioni per nuove osservazioni delle variabili dipendenti (non compresi nei dati usati per la stima dei coefficienti di

[6] Si veda, L. SERLENGA, *Dispense di econometria*, a.a. 2003-04, pag. 3 e segg.
[7] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/car.html

regressione) occorre anche specificare il nuovo dataframe (argomento `newdata`) che contiene queste nuove osservazioni.
Nel caso di intervalli di confidenza abbiamo la seguente formula per l'intervallo:

$$\hat{y}_0 \pm t_{\alpha/2;n-k}\hat{\sigma}\sqrt{x_0'(X'X)^{-1}x_0}$$

```
conf<-predict(fm, level=0.99, interval="confidence")
conf
      fit        lwr      upr
1 160.823307 128.147181 193.49943
2 153.569323 134.186327 172.95232
3 142.536685 130.742511 154.33086
...
98 139.403662 128.485358 150.32197
99 128.608521 116.491767 140.72528
```

per ciascuna osservazione stimata della variabile dipendente abbiamo l'estremo inferiore (`lwr`) e quello superiore (`upr`) dell'intervallo di confidenza.
Il dataframe `newdat` contiene una nuova osservazione con tutte le variabili dipendenti

```
newdata
  Country Calorie HS popphys popnurs
1 Pakistan  2003 13  4580  6300
```

per ottenere l'intervallo di previsione ($\hat{y}_0 \pm t_{\alpha/2;n-k}\hat{\sigma}\sqrt{1+x_0'(X'X)^{-1}x_0}$) abbiamo:

```
pred<-predict(fm, newdata=newdata, interval="prediction")
pred
     fit   lwr   upr
[1,] 124.7727 68.73844 180.807
```

3.7 Selezione delle variabili e aggiornamento del modello di regressione

Uno dei problemi che spesso lo statistico deve affrontare quando effettua l'analisi della regressione è quello della scelta dei regressori[8] da inserire nel modello per descrivere il fenomeno oggetto di studio. Si tratta di un problema abbastanza delicato in quanto bisognerebbe includere nel modello solo quelle variabili esplicative la cui variazione apporta un contributo reale alla variazione della variabile risposta. In genere incrementando il numero dei regressori inseriti nel modello la devianza dei residui tende ad diminuire. Dobbiamo anche considerare che alcune variabili esplicative potrebbero risultare statisticamente significative, e quindi venire incluse nel modello, unicamente per fattori dovuti al caso. Viceversa variabili esplicative logicamente fondamentali potrebbero risultare statisticamente non significative ed essere così escluse dal modello. Di conseguenza, appare chiaro come sia difficile giungere ad un modello ottimo in generale. È più opportuno considerare un certo numero di modelli all'incirca ugualmente significativi dal punto di vista statistico: tra questi il ricercatore può scegliere quello che ritiene più idoneo, anche sulla base di considerazioni legate all'interpretazione del fenomeno oggetto di analisi.
La strategia complessiva della scelta di variabili si può articolare nelle seguenti fasi:
• decidere quali sono le variabili che costituiscono l'insieme più ampio dei k repressori;
• trovare uno o più sottoinsiemi di variabili (p) che spiegano bene la variabile di risposta;
• applicare una regola di arresto per decidere quante variabili esplicative utilizzare;
• stimare i coefficienti di regressione
• saggiare la bontà del modello ottenuto.

[8] Si veda F. DEL VECCHIO, *op. cit.*, pagg. 221-223 e A.POLLICE, *op. cit.,* cap. 4, pagg. 58-60

Il problema della scelta della regola di arresto viene risolto con l'indice C_p di Mallows -basato sull'indice di determinazione- oppure con l'Akaike Information Criterion (AIC) e il Bayes Information Criterion (BIC) - entrambi basati sui logaritmi delle verosimiglianza:[9]

$$C_p = \frac{RSS_p}{\hat{\sigma}^2} + 2p - n$$
$$AIC = n\log(RSS/n) + 2p$$
$$BIC = n\log(RSS/n) + p\log n$$

Tra gli algoritmi di scelta delle variabili abbiamo:

1) **Backward elimination**: si parte considerando il modello che include tutte le variabili a disposizione. Si fissa un livello di significatività. La variabile con il coefficiente di regressione meno significativo in base al test t viene eliminata, quindi si calcolano di nuovo le stime dei coefficienti delle variabili rimaste e si ripete il procedimento sino a quando non vi sono più covariate che risultano non significative al livello prefissato.
2) **Forward selection**: si parte con una sola covariata, quella con la maggiore correlazione significativa (test t) con la variabile risposta. Si fissa un livello di significatività. La seconda variabile da inserire è quella che presenta il coefficiente di correlazione parziale più elevato e significativo, si prosegue si inserendo una successiva variabile dipendente. Il procedimento ha fine quando il coefficiente di correlazione parziale dell'ultima variabile inserita non è più significativa rispetto al livello prefissato; il modello definitivo è quello ottenuto al penultimo passo.
3) **Stepwise regression**: è una combinazione dei due criteri precedenti. La selezione delle covariate da includere nel modello avviene come nella forward selection. Aggiungendo successivamente una nuova variabile, i coefficienti di regressione delle variabili già incluse potrebbero risultare singolarmente non significativi a causa della forte correlazione con la nuova variabile. Pertanto dopo l'inserimento di ciascuna variabile il modello viene riconsiderato per verificare se vi è qualche variabile da eliminare (come nella backward elimination).

Illustriamo ora come procedere nella scelta delle variabili nell'ambiente `R` introducendo un nuovo esempio nel quale si pone in relazione il livello di inquinamento (CO2) in 116 paesi con alcune variabili come l'utilizzo di energia, le esportazioni, il reddito pro capite, il tasso di crescita del PIL e quello della popolazione. Fissiamo il livello di significativa di arresto a 0.05. I dati sono contenuti nel dataframe `inquinamento`.

```
inquinamento
                      CO2   energy export GDPgrowth popgrowth   GNI
Albania              0.92   540.96  19.08      7.30      0.40  1160
Algeria              2.94   958.86  42.50      2.40      1.44  1580
Angola               0.52   651.95  92.37      3.04      2.77   470
...
Vietnam              0.73   476.89  55.03      6.79      1.29   380
Yemen, Rep.          0.48   216.65  42.58      6.55      2.84   420
Zambia               0.18   630.89  21.06      3.58      2.03   320
```

Nella scelta delle variabili usiamo prima la procedura della backward elimination, stimando da principio il modello con tutti i repressori:

```
fm<-lm(formula = CO2 ~ energy + export + GDPgrowth + popgrowth + GNI,
data = inquinamento)

summary(fm)
```

[9] Si veda J. J. FARAWAY, *op. cit*., par. 10.3

```
Call:
lm(formula = CO2 ~ energy + export + GDPgrowth + popgrowth +
    GNI, data = inquinamento)

Residuals:
     Min       1Q   Median       3Q      Max
-16.6036  -0.7481  -0.2488   0.6616   7.4539

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.953e-01  5.668e-01  -0.521   0.6034
energy       2.263e-03  1.512e-04  14.961   <2e-16 ***
export       1.881e-02  9.876e-03   1.904   0.0595 .
GDPgrowth    1.197e-02  8.226e-02   0.145   0.8846
popgrowth   -2.817e-02  1.893e-01  -0.149   0.8820
GNI         -7.443e-05  3.266e-05  -2.279   0.0246 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.428 on 110 degrees of freedom
Multiple R-Squared: 0.8053,     Adjusted R-squared: 0.7964
F-statistic: 90.97 on 5 and 110 DF,  p-value: < 2.2e-16
```

La variabile meno significativa risulta `GDPgrowth` e quindi la eliminiamo aggiornando il modello con il comando `update()`:

```
fm1<-update(fm, . ~ .-GDPgrowth)
summary(fm1)

Call:
lm(formula = CO2 ~ energy + export + popgrowth + GNI, data =
inquinamento)

Residuals:
     Min       1Q   Median       3Q      Max
-16.5948  -0.7511  -0.2545   0.6300   7.4224

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.668e-01  5.295e-01  -0.504   0.6153
energy       2.265e-03  1.494e-04  15.167   <2e-16 ***
export       1.936e-02  9.071e-03   2.134   0.0350 *
popgrowth   -2.858e-02  1.885e-01  -0.152   0.8797
GNI         -7.514e-05  3.215e-05  -2.337   0.0212 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.417 on 111 degrees of freedom
Multiple R-Squared: 0.8052,     Adjusted R-squared: 0.7982
F-statistic: 114.7 on 4 and 111 DF,  p-value: < 2.2e-16
```

la variabile meno significativa ora risulta `popgrowth` e quindi la eliminiamo aggiornando nuovamente il modello:

```
fm2<-update(fm1, .~.-popgrowth)
summary(fm2)
```

```
Call:
lm(formula = CO2 ~ energy + export + GNI, data = inquinamento)

Residuals:
     Min       1Q   Median       3Q      Max
-16.6158  -0.7596  -0.2651   0.6164   7.3845

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -3.113e-01  4.391e-01  -0.709   0.4798
energy       2.266e-03  1.486e-04  15.253   <2e-16 ***
export       1.945e-02  9.012e-03   2.158   0.0331 *
GNI         -7.461e-05  3.182e-05  -2.344   0.0208 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.406 on 112 degrees of freedom
Multiple R-Squared: 0.8052,     Adjusted R-squared:   0.8
F-statistic: 154.3 on 3 and 112 DF,  p-value: < 2.2e-16
```

A questo punto i regressori rimasti risultano tutti significativamente diversi da zero avendo un p-value inferiore al livello di significatività fissato. Il comando `update()` viene usato per aggiornare e stimare nuovamente il modello.
Volendo utilizzare criteri basati su C_p o su AIC si possono usare i comandi `drop1()` e `add1()` per eliminare o aggiungere variabili, il comando `step()` e il comando `leaps()` presente nell'omonimo package.

```
drop1(fm, test="F")
Single term deletions

Model:
CO2 ~ energy + export + GDPgrowth + popgrowth + GNI
           Df Sum of Sq     RSS     AIC  F value   Pr(F)
<none>                  648.29  211.61
energy      1   1319.21 1967.50  338.39 223.8413 < 2e-16 ***
export      1     21.37  669.66  213.37   3.6260 0.05949 .
GDPgrowth   1      0.12  648.41  209.63   0.0212 0.88462
popgrowth   1      0.13  648.42  209.63   0.0221 0.88199
GNI         1     30.61  678.90  214.96   5.1942 0.02459 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Il comando `step()` consente di effettuare una stepwise regression basata sul criterio AIC, nell'argomento `direction` si può indicare se usare una procedura backward (`direction="backward"`), forward (`direction="forward"`), oppure entrambe (`direction="both"`).

```
backsel<-step(fm, direction="backward") ## procedura backward
Start:  AIC= 211.61
 CO2 ~ energy + export + GDPgrowth + popgrowth + GNI

             Df Sum of Sq     RSS     AIC
- GDPgrowth   1      0.12  648.41  209.63
- popgrowth   1      0.13  648.42  209.63
<none>                    648.29  211.61
- export      1     21.37  669.66  213.37
```

```
- GNI        1     30.61  678.90  214.96
- energy     1   1319.21 1967.50  338.39

Step:  AIC= 209.63
 CO2 ~ energy + export + popgrowth + GNI

            Df Sum of Sq     RSS     AIC
- popgrowth  1      0.13  648.55  207.65
<none>                    648.41  209.63
- export     1     26.61  675.02  212.29
- GNI        1     31.90  680.31  213.20
- energy     1   1343.76 1992.17  337.83

Step:  AIC= 207.65
 CO2 ~ energy + export + GNI

         Df Sum of Sq     RSS     AIC
<none>                 648.55  207.65
- export  1     26.97  675.51  210.38
- GNI     1     31.83  680.37  211.21
- energy  1   1347.19 1995.73  336.04

step(fm3, scope=formula(fm), direction="forward") ## procedura forward
## nell'argomento scope si è specificato il modello con tutte le
variabili

Start:  AIC= 213.34
 CO2 ~ energy

            Df Sum of Sq    RSS    AIC
+ GNI        1     29.55 675.51 210.38
+ export     1     24.69 680.37 211.21
<none>                   705.06 213.34
+ GDPgrowth  1      8.90 696.16 213.87
+ popgrowth  1      0.01 705.05 215.34

Step:  AIC= 210.38
 CO2 ~ energy + GNI

            Df Sum of Sq    RSS    AIC
+ export     1     26.97 648.55 207.65
<none>                   675.51 210.38
+ GDPgrowth  1      5.48 670.03 211.43
+ popgrowth  1      0.49 675.02 212.29

Step:  AIC= 207.65
 CO2 ~ energy + GNI + export

            Df Sum of Sq    RSS    AIC
<none>                   648.55 207.65
+ popgrowth  1      0.13 648.41 209.63
+ GDPgrowth  1      0.13 648.42 209.63

Call:
lm(formula = CO2 ~ energy + GNI + export, data = inquinamento)

Coefficients:
```

```
(Intercept)       energy          GNI       export
 -3.113e-01    2.266e-03   -7.461e-05    1.945e-02

stepsel<-step(fm, direction="both") ## procedura stepwise

Start:  AIC= 211.61
 CO2 ~ energy + export + GDPgrowth + popgrowth + GNI

            Df Sum of Sq     RSS     AIC
- GDPgrowth  1      0.12  648.41  209.63
- popgrowth  1      0.13  648.42  209.63
<none>                    648.29  211.61
- export     1     21.37  669.66  213.37
- GNI        1     30.61  678.90  214.96
- energy     1   1319.21 1967.50  338.39

Step:  AIC= 209.63
 CO2 ~ energy + export + popgrowth + GNI

            Df Sum of Sq     RSS     AIC
- popgrowth  1      0.13  648.55  207.65
<none>                    648.41  209.63
+ GDPgrowth  1      0.12  648.29  211.61
- export     1     26.61  675.02  212.29
- GNI        1     31.90  680.31  213.20
- energy     1   1343.76 1992.17  337.83

Step:  AIC= 207.65
 CO2 ~ energy + export + GNI

            Df Sum of Sq     RSS     AIC
<none>                    648.55  207.65
+ popgrowth  1      0.13  648.41  209.63
+ GDPgrowth  1      0.13  648.42  209.63
- export     1     26.97  675.51  210.38
- GNI        1     31.83  680.37  211.21
- energy     1   1347.19 1995.73  336.04
```

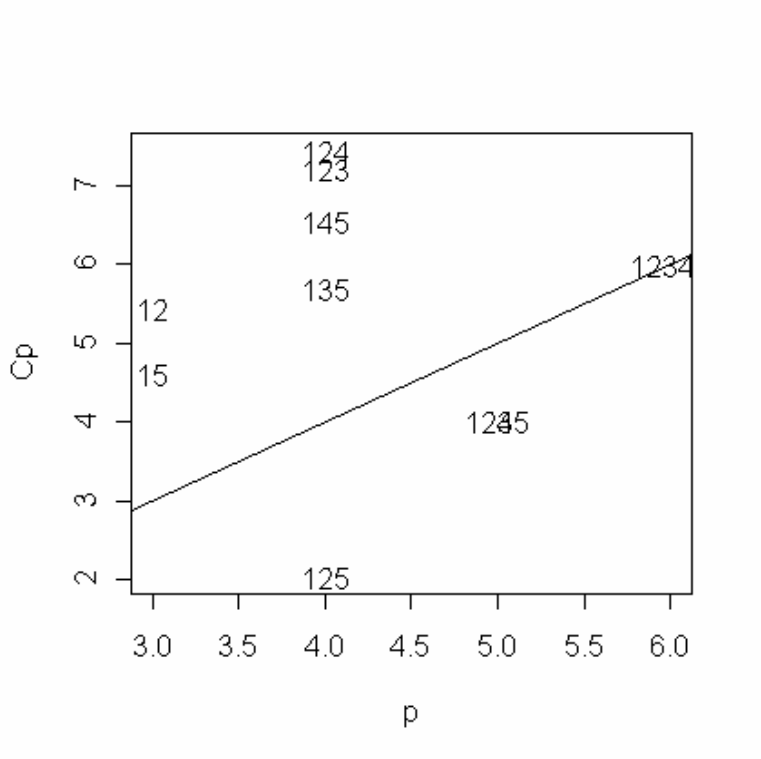
Il comando `leaps()` consente di ottenere il migliore subsest di regressori da usare nel modello prendendo in considerazione per la scelta delle variabili alcuni algoritmi basati su C_p, R^2 e adjusted R^2: occorre scegliere il tipo di indicatore nell'argomento `method=c("Cp", "adjr2", "r2")` e specificare la variabile dipendente (y) e la matrice dei predittori (x).

```
library(leaps) ##caricamento package leaps
y<-inquinamento$CO2
x<-model.matrix(fm)[,-1]
leapcp<-leaps(x,y, method="Cp")
library(faraway)## caricamento package faraway
Cpplot(leapcp)
```

Osservando il Cpplot (Fig. 2) tracciato con il comando omonimo compreso nel package `faraway`[10] si vede che il valore minimo per l'indice di Mallow è in corrispondenza della combinazione di numeri 125, corrispondente al subset di variabili dipendenti (`energy, export, GNI`)

[10] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/faraway.html

Fig. 2



Cp plot

Analogamente si può procedere usando come indicatore R^2 adjusted pervenendo al medesimo risultato:

```
leapadjr<-leaps(x,y, method="adjr")
maxadjr(leapadjr,8)
  1,2,5  1,2,4,5  1,2,3,5   1,2,3,4,5    1,5    1,3,5    1,2    1,4,5
  0.800   0.798    0.798      0.796     0.793   0.793   0.792   0.792
```

3.8 Confronto tra modelli

Per confrontare due o più modelli di regressione che differiscono per il numero di variabili esplicative inserite si usa il comando `anova()` che mette in evidenza se le variabili in più o in meno di un modello rispetto all'altro apportano oppure no un contributo significativo nello spiegare la variabile risposta verificato tramite il test F:

```
anova(fm3,fm2)
Analysis of Variance Table

Model 1: CO2 ~ energy
Model 2: CO2 ~ energy + export + GNI
 Res.Df  RSS Df Sum of Sq  F  Pr(>F)
1  114 705.06
2  112 648.55  2   56.52 4.88 0.009289 **
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(fm,fm2)
Analysis of Variance Table
```

```
Model 1: CO2 ~ energy + export + GDPgrowth + popgrowth + GNI
Model 2: CO2 ~ energy + export + GNI
 Res.Df  RSS Df Sum of Sq   F Pr(>F)
1  110 648.29
2  112 648.55 -2   -0.26 0.022 0.9783
```

Come si evince facilmente nel primo caso l'aggiunta delle due variabili apporta un contributo significativo nello spiegare la variabile `CO2`, mentre non è così nel secondo esempio.
Un altro uso del comando `anova()` è quello di verificare dei modelli in sequenza, partendo da quello nullo (ossia senza repressori), e calcolando la devianza spiegata da ogni variabile aggiuntiva.

```
anova(fm)
Analysis of Variance Table

Response: CO2
           Df  Sum Sq Mean Sq  F value  Pr(>F)
energy      1 2623.88 2623.88 445.2143 < 2e-16 ***
export      1   24.69   24.69   4.1894 0.04306 *
GDPgrowth   1    1.41    1.41   0.2395 0.62552
popgrowth   1    0.06    0.06   0.0103 0.91930
GNI         1   30.61   30.61   5.1942 0.02459 *
Residuals 110  648.29    5.89
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Simile ad `anova()`è il comando `Anova()` del package `car` che consente di effettuare analisi della varianza di II e III tipo.

3.9 Diagnostica

Occorre specificare che la maggior parte dei comandi di `R` usati nella diagnostica possono essere applicati anche nel caso di modelli lineari generalizzati.
Riprendiamo l'esempio della regressione sui dati della mortalità gia esaminato in precedenza. Andremo ad effettuale la diagnostica della regressione su questo modello dopo aver fatto un'analisi grafica dei residui.

3.9.1 Richiami di teoria

Facciamo di seguito alcuni rapidi richiami di teoria utili nel prosieguo:

$e = (I - H)y$ è il vettore dei residui, la matrice H è la matrice di proiezione

$\text{var}(e) = (I - H)\sigma^2$ è la matrice delle varianze e covarianze dei residui, da cui: $\text{var}(e_i) = (1 - h_i)\sigma^2$ dove $h_i=H_{ii}$ sono i valori di leva (*leverage*) e sono gli elementi della diagonale principale di H; per i valori di leva si ha che $\sum_{i=1}^{n} h_i = p$ (numero di parametri da stimare nel modello di regressione), $h_i < \frac{1}{n} \ \forall i$, mentre sono da prendere in esame i valori di leverage $h_i \geq 2p/n$, poiché potenzialmente anomali;

residui standardizzati= $\frac{e_i}{\sqrt{(1-h_i)}}$

residui studentizzati= $\frac{e_i}{\hat{\sigma}\sqrt{(1-h_i)}}$;

$\hat{\sigma}^2_{(i)}$ è la stima della varianza dei residui ottenuta eliminando dal dataset l'i.esima osservazione, mentre $\hat{\beta}_{(i)}$ è la stima dei coefficienti di regressione ottenuta escludendo l'i.esima osservazione;

residui studentizzati jacknife= $\frac{e_i}{\hat{\sigma}_{(i)}\sqrt{1-h_i}}$;

una delle misure utilizzata nell'individuazione dei punti influenti è la distanza di Cook: $D_i = \frac{(\hat{\beta}-\hat{\beta}_{(i)})'(X'X)(\hat{\beta}-\hat{\beta}_{(i)})}{p\hat{\sigma}^2_{(i)}}$;

$DFFITS = \frac{\hat{y}_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}\sqrt{h_i}}$

$dfbeta = \hat{\beta}-\hat{\beta}_{(i)} = X_{(i)}(X'X)^{-1}(\frac{e_i}{1-h_i})$ dove $X_{(i)}$ è la matrice dei dati nei quali è stata soppressa l'i.esima osservazione.

3.9.2 Analisi grafica dei residui

Un'analisi preliminare che può essere effettuata ai fini della diagnostica è quella della verifica degli assunti di base del modello della regressione lineare[11] e cioè:

- linearità (la funzione che lega la variabile dipendente alle variabili indipendenti è lineare)
- normalità (la distribuzione dei residui è di tipo gaussiano)
- omoscedasticità (la varianza dei residui è costante)
- indipendenza (i residui sono tra loro indipendenti)

utilizzando il metodo grafico.
Per verificare la linearità occorre tracciate il grafico dei residui (ordinata) verso i valori previsti (ascissa) come in Fig. 3. I punti dovrebbero essere distribuiti in modo simmetrico intorno ad una linea orizzontale con intercetta uguale a zero. Andamenti di tipo diverso indicano la presenza di non linearità.

```
residui<-fm$res
yfit<-fitted(fm)
plot(yfit, residui, ylab="Residui", xlab="Fitted", main="Residui vs fitted")
abline(h=0)
```

Tracciando il grafico dei residui (ordinata) verso ciascun regressori (ascissa) si può verificare se è adatta la relazione lineare tra la variabile risposta e ciascuna variabile esplicativa (Fig. 4). La disposizione dei residui dovrebbe essere casuale.

```
par(mfrow=c(2,2), mar=c(4,4,1,1))
for (i in 2:5) plot(mortalita[,i],residui,xlab=names(mortalita)[i])
```

Come si può vedere dal Fig. 4, la linearità sembra non andare bene per la relazione con le variabili `popnur` e `popphys`. Tracciamo il plot tra questa variabili prese singolarmente e la variabile risposta (Fig. 5) per averne conferma:

```
par(mfcol=c(1,2))
for (i in 4:5) plot(mortalita[,i], mortalita$mortal, xlab=names(mortalita)[i], ylab="mortal")
```

Più in generale usiamo il comando `pairs()` per tracciare il plot tra tutte le variabili presenti nel dataframe (Fig. 5/bis):

```
pairs(mortalita)
```

[11] R. MICCIOLO, *Dispense di Econometria ed applicazioni ai servizi sanitari*, 2004

Per verificare la normalità si ricorre al QQ-plot dei residui standardizzati (Fig. 6), inoltre è opportuno verificare la normalità anche con un test statistico appropriato (si veda il paragrafo 3.3). Se gli errori seguono una distribuzione gaussiana, i punti del grafico dovrebbero concentrarsi intorno ad una retta a 45°:

```
qqnorm(scale(residui))
abline(0,1)
```

Fig. 3

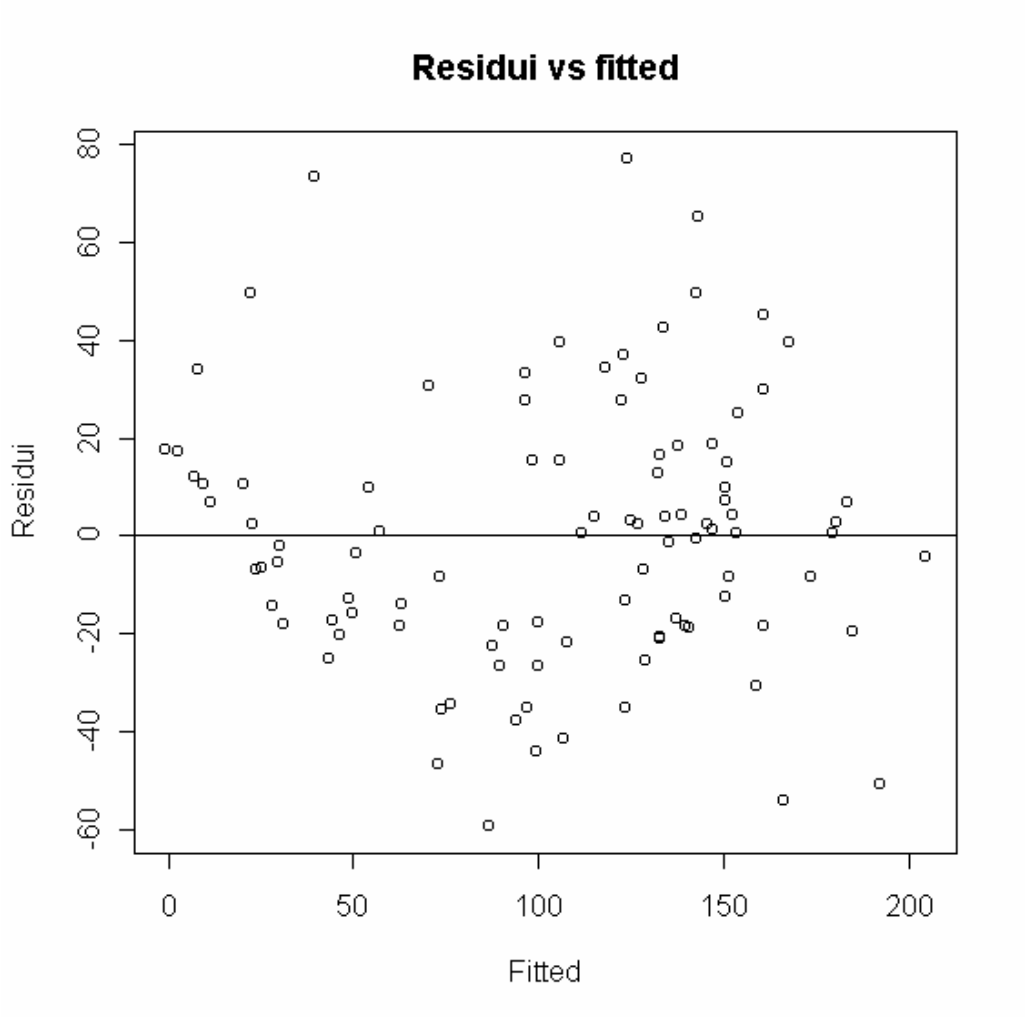


Fig. 4

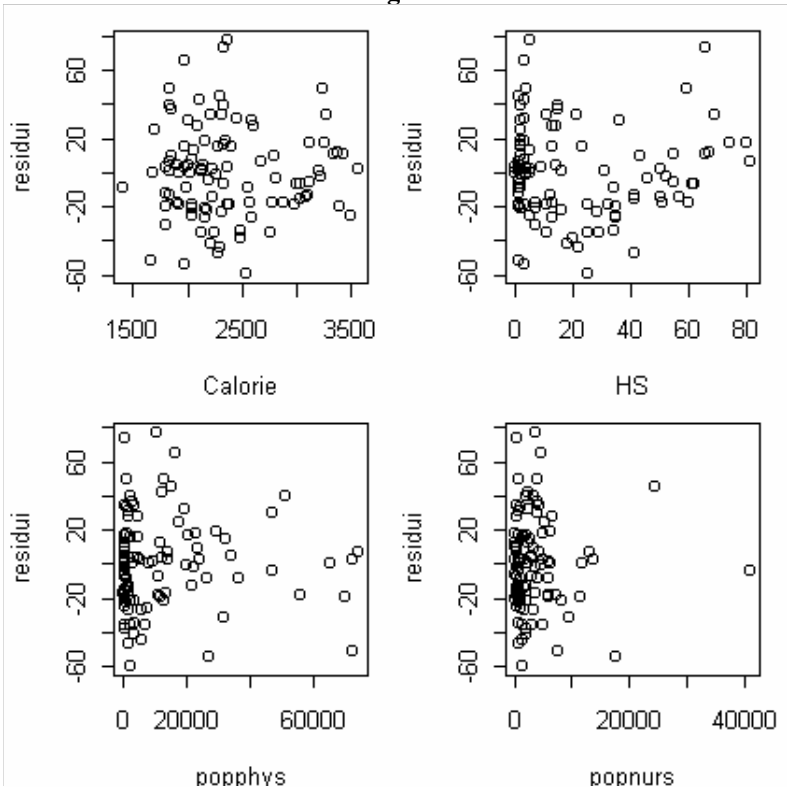

residui
Calorie
HS
popphys
popnurs

Fig. 5

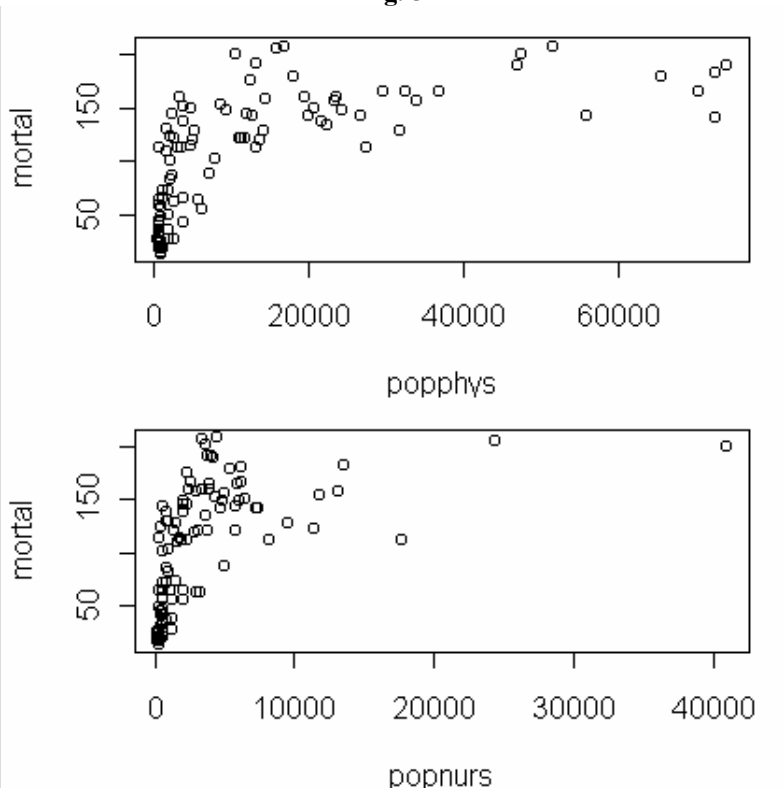
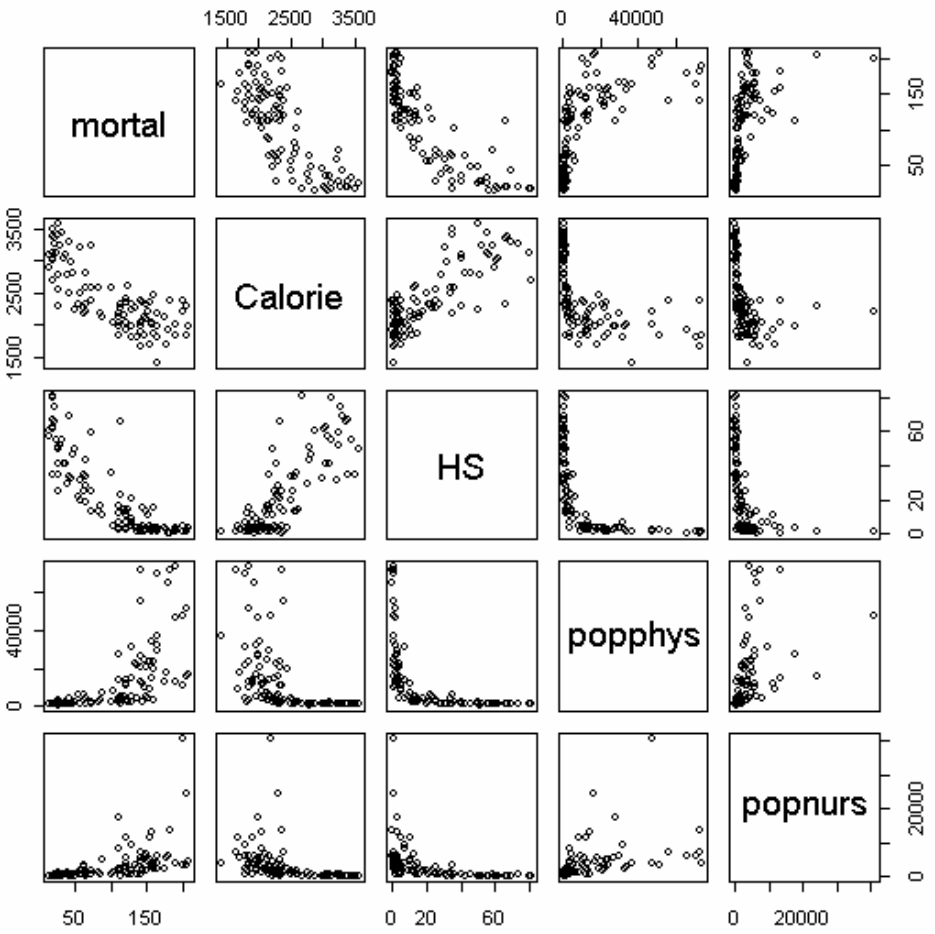

mortal
popphys
mortal
popnurs

Fig. 5/bis



Per verificare l'omoscedasticità occorre tracciare il grafico dei residui in valore assoluto (ordinata) verso i valori stimati con il modello: la dispersione verticale dovrebbe essere approssimativamente costante (Fig. 7). Anche in questo caso è opportuno ricorre ad un test statistico per verificare l'omoscedasticità (si veda il paragrafo 3.3).

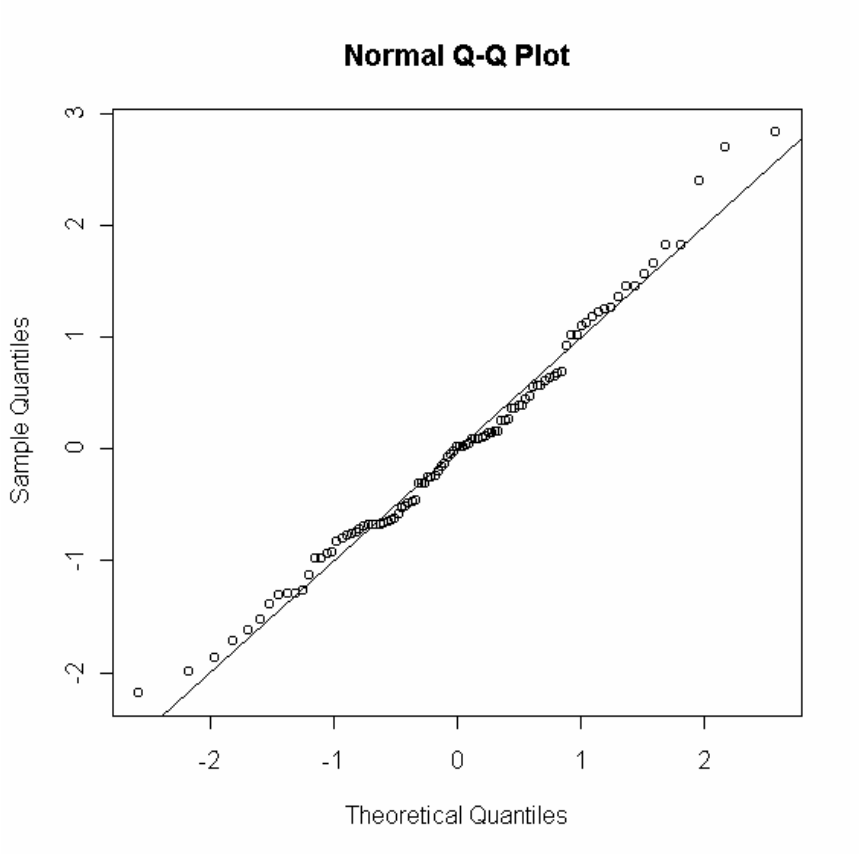
```
plot(yfit, abs(residui), ylab="Residui", xlab="Fitted", main="Residui in
valore assoluto vs fitted")
```

Un altro metodo per verificare l'omoscedasticità consiste nello stimare un modello regressivo con i valori assoluti dei residui come variabile dipendente e i valori previsti come variabile indipendente; se vi è omoscedasticità la pendenza della retta dovrebbe essere uguale a zero. Infatti abbiamo:

```
g<-lm(abs(residui)~ yfit)
summary(g)

Call:
lm(formula = abs(residui) ~ yfit)

Residuals:
   Min     1Q Median     3Q    Max
-21.296 -13.563  -3.072  10.367 55.875

Coefficients:
```

```
       Estimate Std. Error t value Pr(>|t|)
(Intercept) 20.229268  3.893252  5.196 1.13e-06 ***
yfit     0.008588  0.033441  0.257  0.798
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.13 on 97 degrees of freedom
Multiple R-Squared: 0.0006794, Adjusted R-squared: -0.009623
F-statistic: 0.06595 on 1 and 97 DF, p-value: 0.7979
```

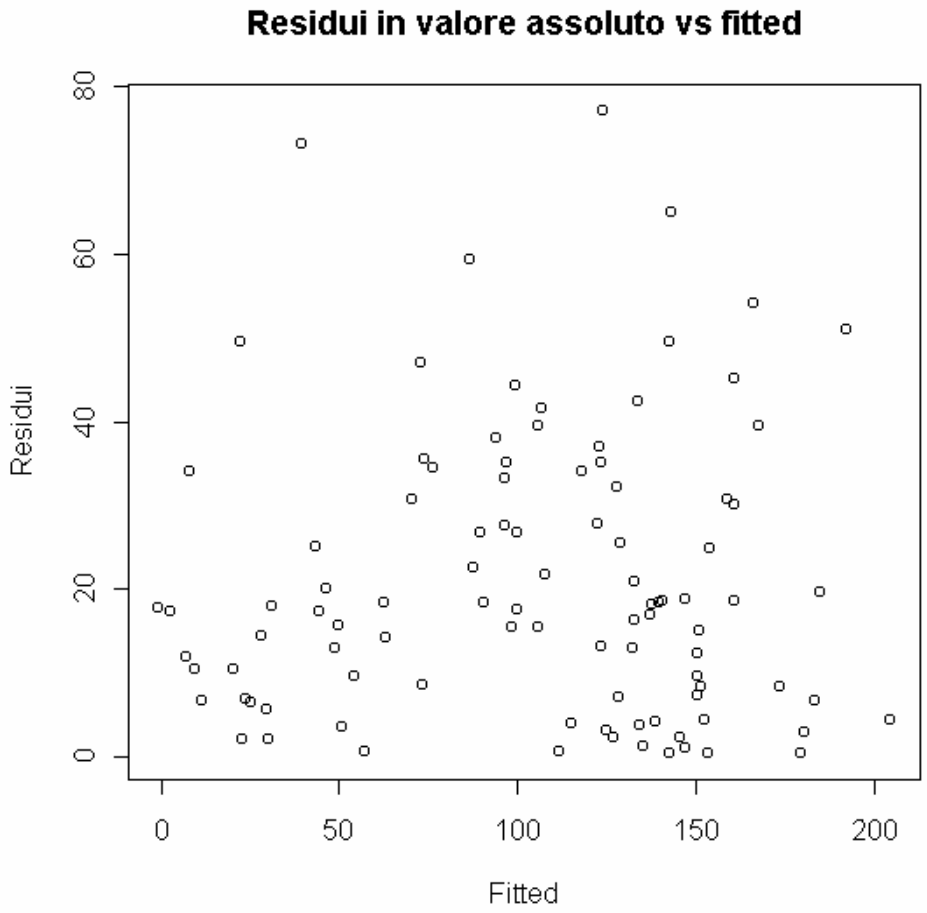
Fig. 6



Anche questa prova conferma la costanza della variabilità dei residui.
Per verificare l'assenza di correlazione seriale, oltre al test di Durbin-Watson (paragrafo 3.3) si può tracciare il grafico dei residui (ordinata) verso i residui precedenti"(ascissa) che non dovrebbe rivelare alcun pattern evidente (Fig. 8):

```
n<-length(residui)
plot(residui[-n], residui[-1])
```

Fig. 7



3.9.3 Outlier, leverage, influence

Un outlier è un punto che non è ben interpolato dal modello stimato. Se il residuo jacknife associato al punto i è " grande", allora il punto i è un outlier. L'i.esimo residuo jacknife dovrebbe seguire una distribuzione t di Student con (n - p - 1) gradi di libertà.
Calcoliamo i residui standardizzati, quelli studentizzati e quelli jacknife:

```
rstand<-rstandard(fm)

rstand<-rstandard(fm)## residui standardizzati
rstand
         Afghanistan              Algeria               Angola
          1.81616798           0.01606305           1.80243072
…
           Yugoslavia               Zambia             Zimbabwe
           1.82506296          -0.66934711          -0.93386015
```

tracciamo il grafico dei residui standardizzati (Fig. 9):

```
plot(rstand, main="Residui standardizzati")
abline(h=2)
abline(h=-2)
```

```
country<-names(rjack)
```

Fig. 8

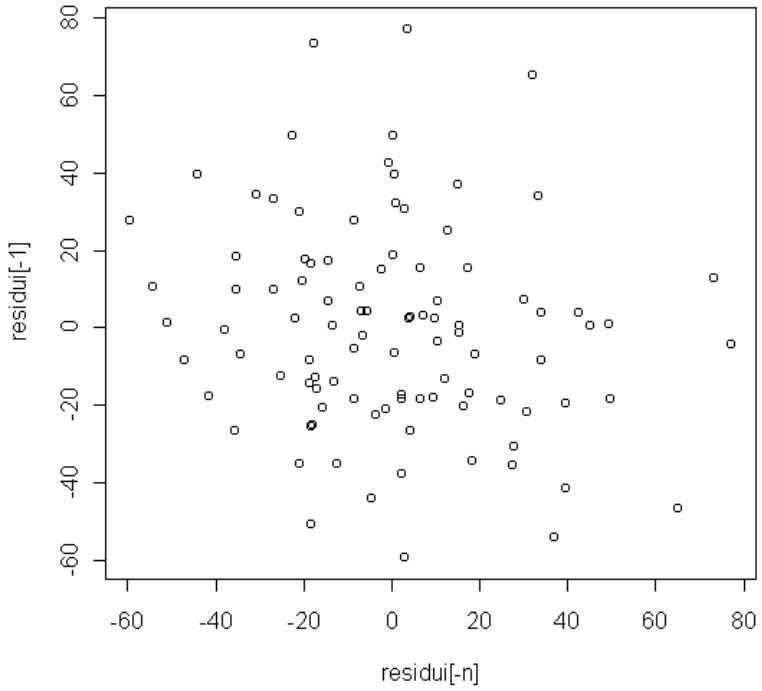
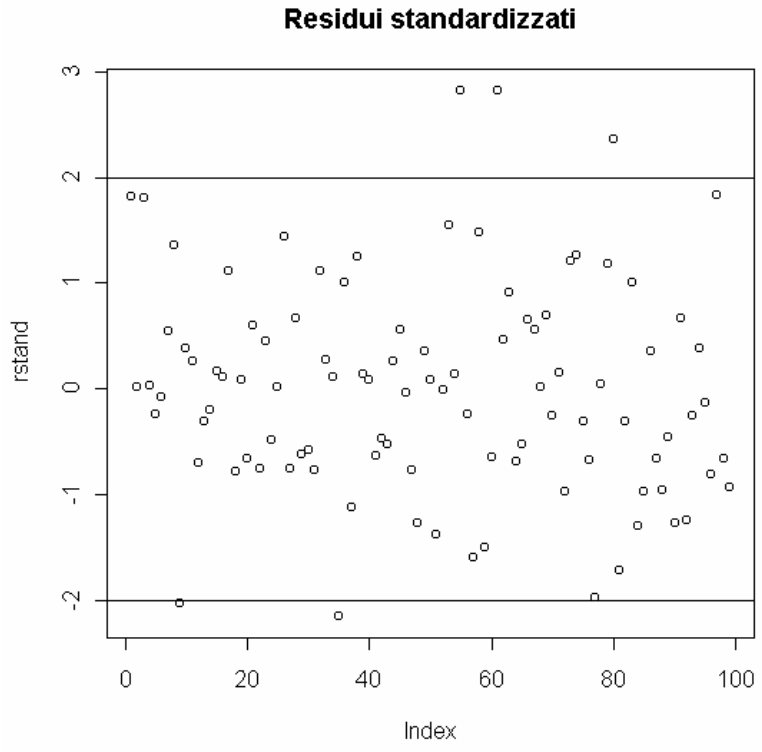


Fig. 9



mettiamo in evidenza i valori dei residui standardizzati esterni alla bandi confidenza della distribuzione normale standardizzati al 95% (-2, 2) che possono ritenersi anomali:

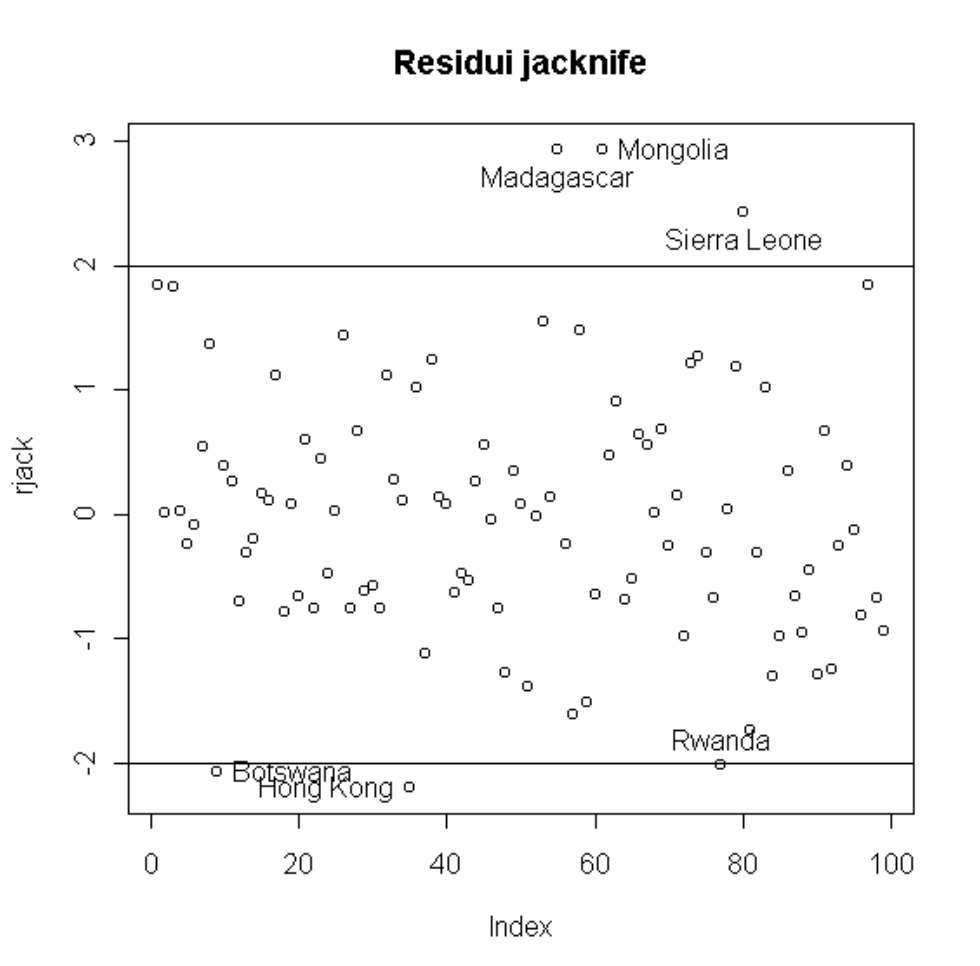
```
rstand[abs(rstand)>2]
    Botswana    Hong Kong   Madagascar     Mongolia Sierra Leone
   -2.034621    -2.154136     2.815123     2.820869     2.363657
```

Calcoliamo i residui studentizzati jacknife e il relativo grafico (Fig. 10):

```
rjack<-rstudent(fm)
rjack
         Afghanistan              Algeria               Angola
          1.83903576           0.01597740           1.82462560
…
    Yugoslavia               Zambia             Zimbabwe
          1.84837278           -0.66736955          -0.93321863

plot(rjack, main="Residui jacknife")
abline(h=-2)
abline(h=2)
country<-names(rjack)
identify(1:length(rjack),rjack, country)
```

Il commando `identify()` consente, cliccando con il mouse sul grafico, di aggiungere le etichette con il nome della nazione.

Fig. 10



Possiamo calcolare il p-value (con la correzione di Bonferroni) associato ad ogni residuo jacknife:

```
n<-length(rjack)
p<-fm$rank
pv<- 2*pt(abs(rjack),n-p-1,lower.tail= F)
```

Guardiamo i residui jacknife a cui è associato un p-value inferiore, per esempio, al 5%:

```
rjack[pv<0.05]
rjack[pv<0.05]
    Botswana    Hong Kong   Madagascar     Mongolia       Rwanda Sierra Leone
   -2.069861    -2.197575     2.926174     2.932699    -2.013996     2.424195
```

sono residui esterni alla banda di confidenza e vanno considerati come valori outliers.
Nel package `car` troviamo il comando `outlier.test()` che consente di effettuare il test per individuare i valori outlier fornendo il Bonferroni p-value per il valore outlier più estremo:

```
library(car)
outlier.test(fm)

max|rstudent| = 2.932699, degrees of freedom = 93,
unadjusted p = 0.0042304, Bonferroni p = 0.4188096

Observation: Mongolia
```

Talvolta, può essere utile ricorrere al *partial residual plot*[12]. Infatti, quando si stima un modello di regressione semplice, lo scatter plot tra la variabile risposta e la variabile esplicativa fornisce una buona indicazione circa la natura della relazione tra le due variabili. Quando, invece, i regressori sono più di uno, la relazione tra un dato regressore e la variabile risposta può essere influenzata dai restanti regressori. Il *partial residual plot* consente di mostrare la relazione tra una data variabile esplicativa e la variabile risposta al netto dell'influenza degli altri repressori del modello. In `R` si possono usare il comando `cr.plots()` del package `car` (Figg. 11 e 12) oppure il comando `prplot()` del package `faraway` (Fig. 13):

```
cr.plots(fm)

Selezione: 2
Selezione: 3
```

[12] http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/partresi.htm

Fig. 11

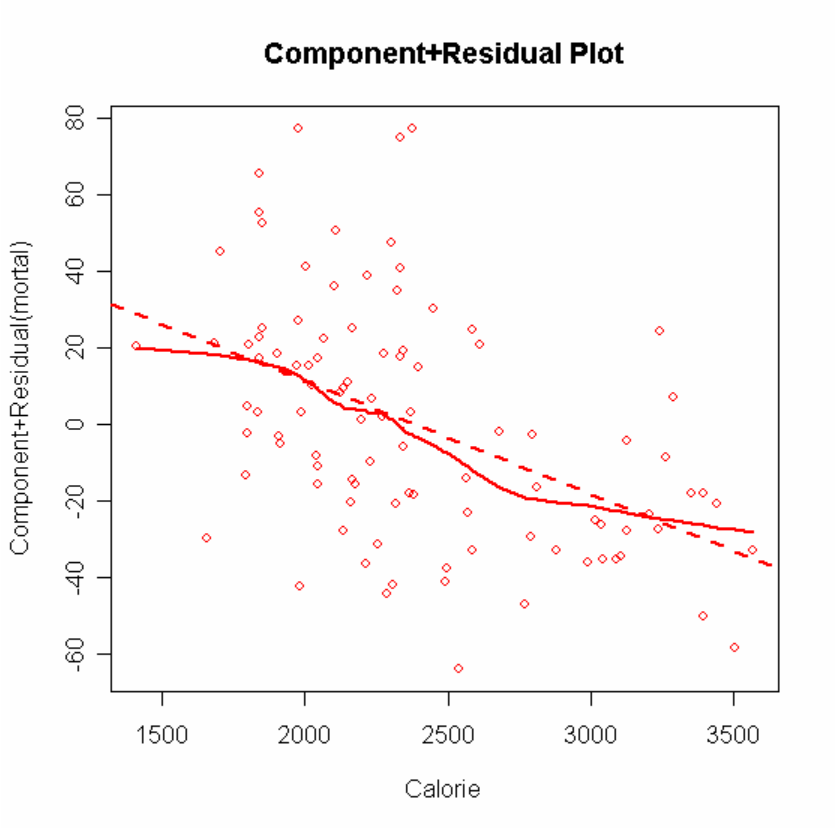
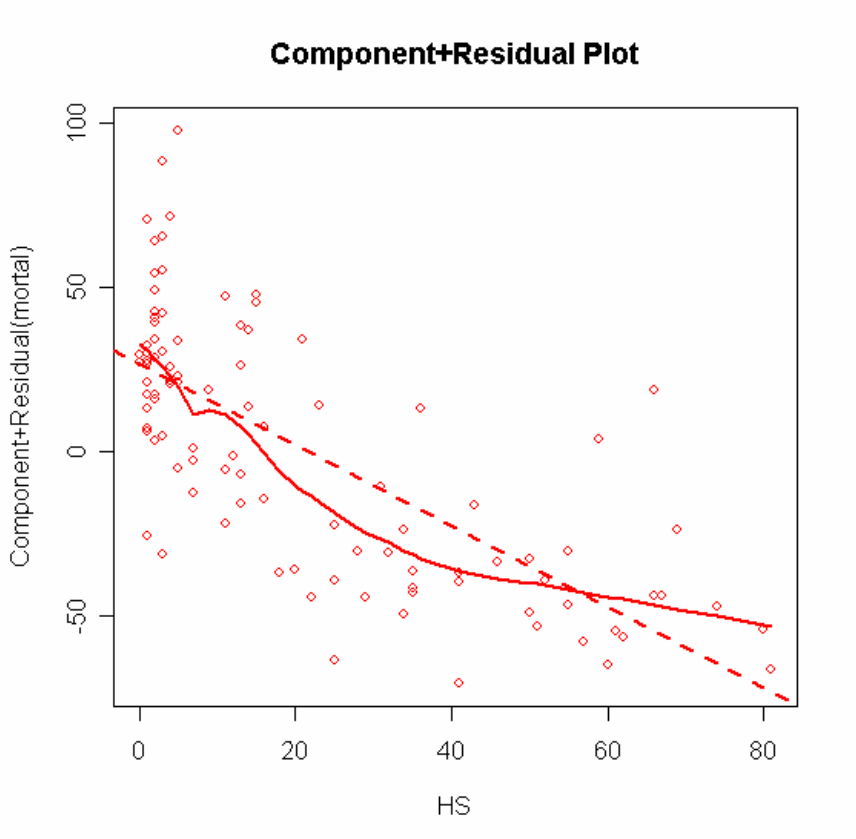
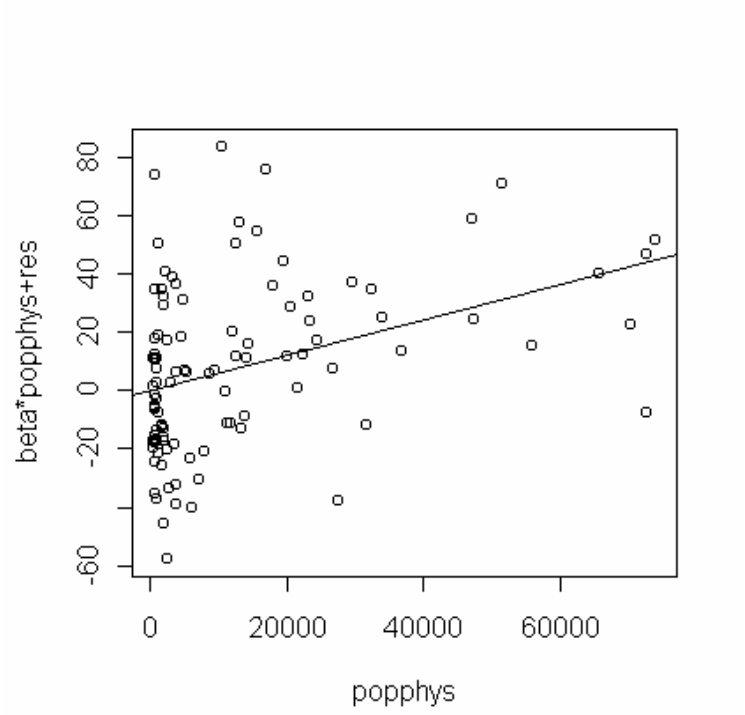

Component+Residual Plot
Component+Residual(mortal)
80
60
40
20
0
-20
-40
-60
1500
2000
2500
3000
3500
Calorie

Fig. 12


Component+Residual Plot
Component+Residual(mortal)
100
50
0
-50
0
20
40
60
80
HS

```
library(faraway)
prplot(fm, 3)
```

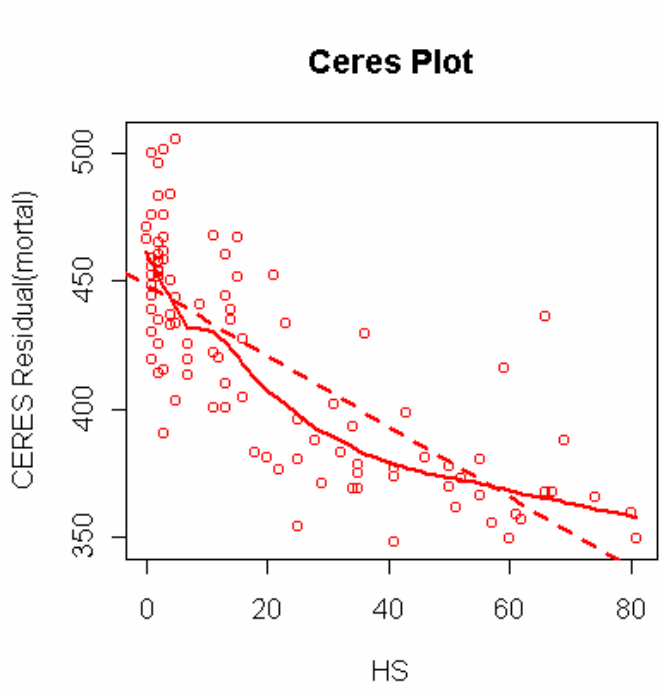
Fig. 13



Una generalizzazione del Component+Residual plot è il CERES plot che può essere ottenuto con il comando `ceres.plot()` disponibile nel package `car`, nell'argomento `variable` va specificato il repressore (Fig. 14):

```
library(car)
ceres.plots(fm, variable="HS")
```

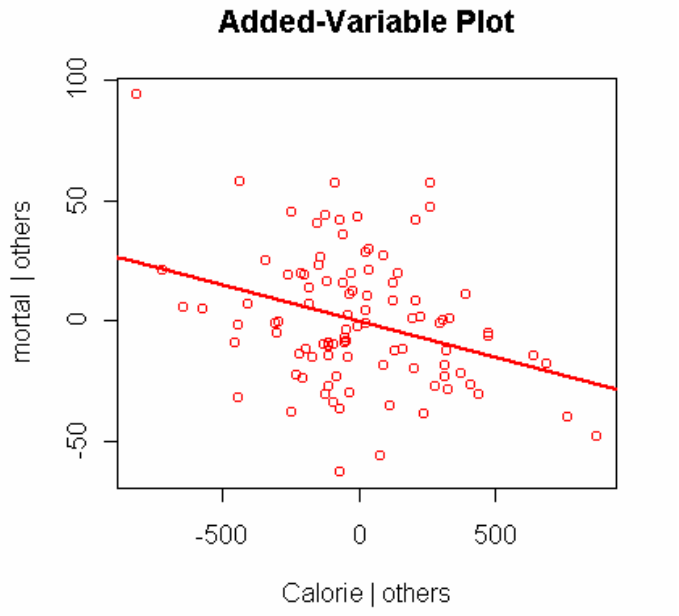
Fig. 14



Il package `car` mette a disposizione anche il comando `av.plot()` che permette di tracciare i *partial regression plots* (Fig. 15):

```
av.plots(fm, variable="Calorie")
```

Fig. 15



Si veda anche il comando `termplot()` nel package `stats`.
Gli elementi h_{ii} sulla diagonale principale della matrice H (hat) si chiamano *leverages* (punti di leva) [13]. Poiché $\text{var}(\hat{y}_i) = h_i\sigma^2$ h_i è la precisione con cui il valore è stimato relativamente a σ^2. Quindi, valori piccoli

[13] G. M. MARCHETTI, *Dispense di Statistica 3*, 2003, pagg. 39-41

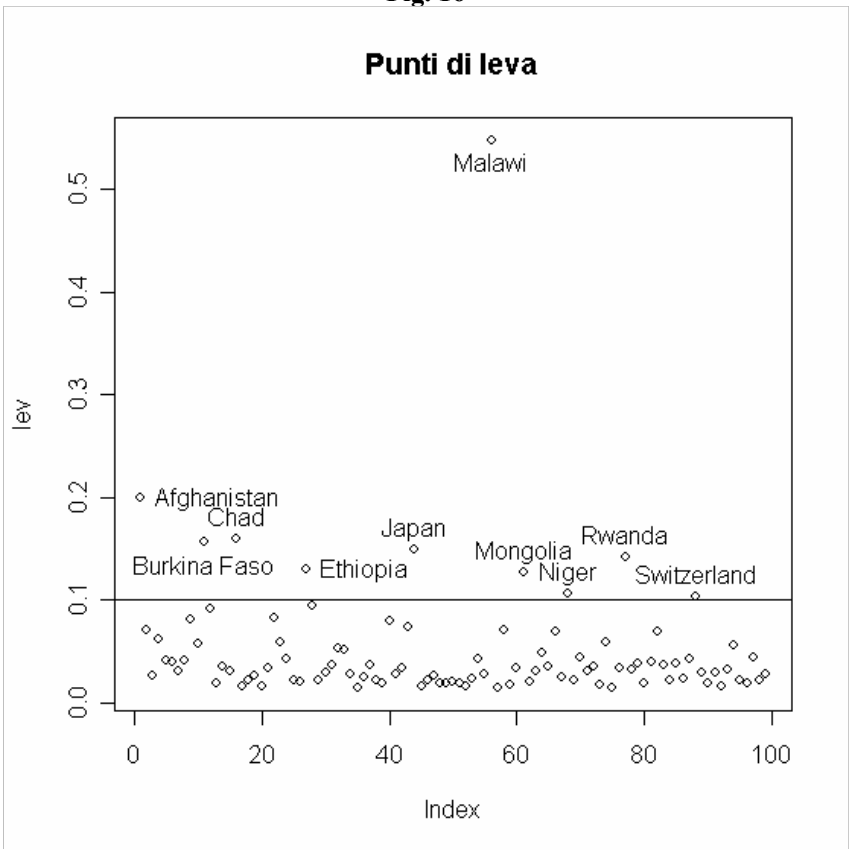
di h_i indicano che lo stimatore di y_i è basato sul contributo di molte osservazioni. Invece, valori grandi di h_i ossia molto vicini a 1, implicano che $\text{var}(y_i - \hat{y}_i) = (1-h_i)\sigma^2 \approx 0$ e che $\hat{y}_i$ tende a essere vicino a y_i e che $\hat{y}_i$ è determinato in modo predominante dalla singola osservazione y_i che quindi ha un effetto di leva importante. Un punto con " alto" leverage ha un residuo con varianza "piccola" (cioè la retta deve passare " vicino" a questo punto). Un punto con " alto" leverage è un punto " distante". Hoaglin e Wesh suggeriscono di segnalare come punti con un elevato effetto di leva quei punti per cui $h_i > 2p/n$.
Per calcolare i punti di leverage in `R` possiamo usare i comandi `hat()` e `hatvalues()`:

```
lev<-hat(model.matrix(fm))## oppure
lev<-hatvalues(fm)
```

Tracciamo il grafico degli hat values (Fig. 16):

```
n<-length(lev)
p<-sum(lev)
plot(lev, main="Punti di leva")
abline(h=2*p/n)
identify(1: length(lev), lev, country)
```

Fig. 16




```
hatvalues(fm)
         Afghanistan              Algeria               Angola
           0.19976958           0.07029267           0.02602576
           Argentina            Australia              Austria
           0.06261963           0.04181677           0.04004242
…
           Yugoslavia               Zambia             Zimbabwe
           0.04366330           0.02230379           0.02746887
```

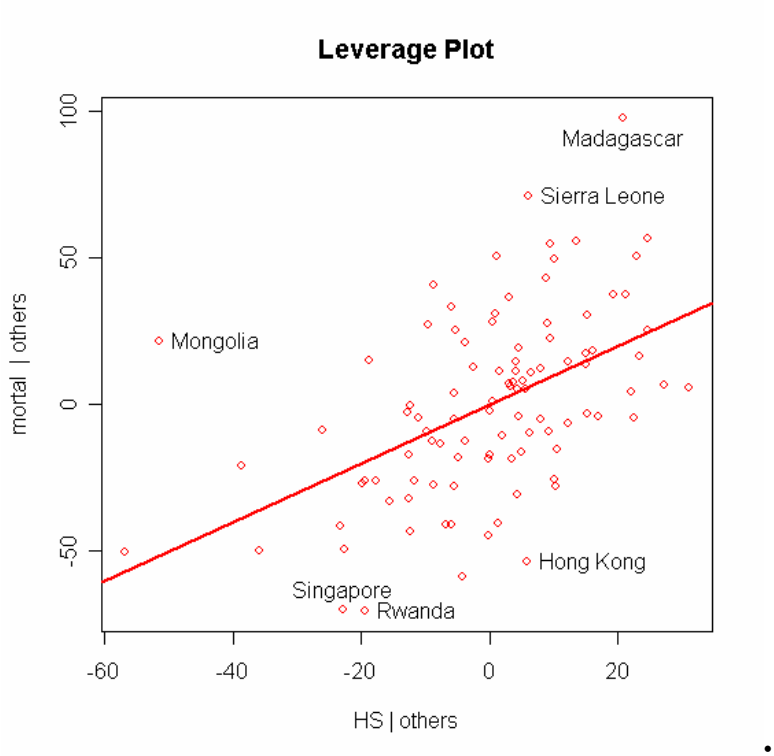
otteniamo i punti leverage superiori al valore soglia:

```
lev[lev>2*p/n]
 Afghanistan Burkina Faso        Chad     Ethiopia        Japan       Malawi
   0.1997696    0.1566912   0.1594579    0.1302486    0.1487301    0.5478660
    Mongolia        Niger       Rwanda  Switzerland
   0.1272775    0.1069104   0.1417091    0.1030682
```

Un altro strumento per la diagnostica è il *partial leverage plots*[14]. Quando le variabili esplicative sono più di una, la relazione tra i residui e una variabile esplicativa può essere influenzato per effetto degli altri regressori. Il *partial leverage plots* mette in evidenza queste relazioni. Sull'asse delle ascisse sono rappresentati i residui della regressione della i.esima variabile esplicativa sui rimanenti k-1 regressori; sull'asse delle ordinate sono rappresentati i residui della regressione della variabile risposta sui tutti i regressori escludento l'i.esimo.
Il *partial leverage plots* è usato per misurare il contributo della variabile indipendente al leverage di ciascuna osservazione, misura, cioè, come variano gli hat values quando si aggiunge un regressore al modello (Graff. 17, 18 e 19)

```
library(car)
leverage.plots(fm)

1: (Intercept)
2: Calorie
3: HS
4: popphys
5: popnurs

Selezione:
```

Fig. 17



[14] http://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/partleve.htm

Fig. 18

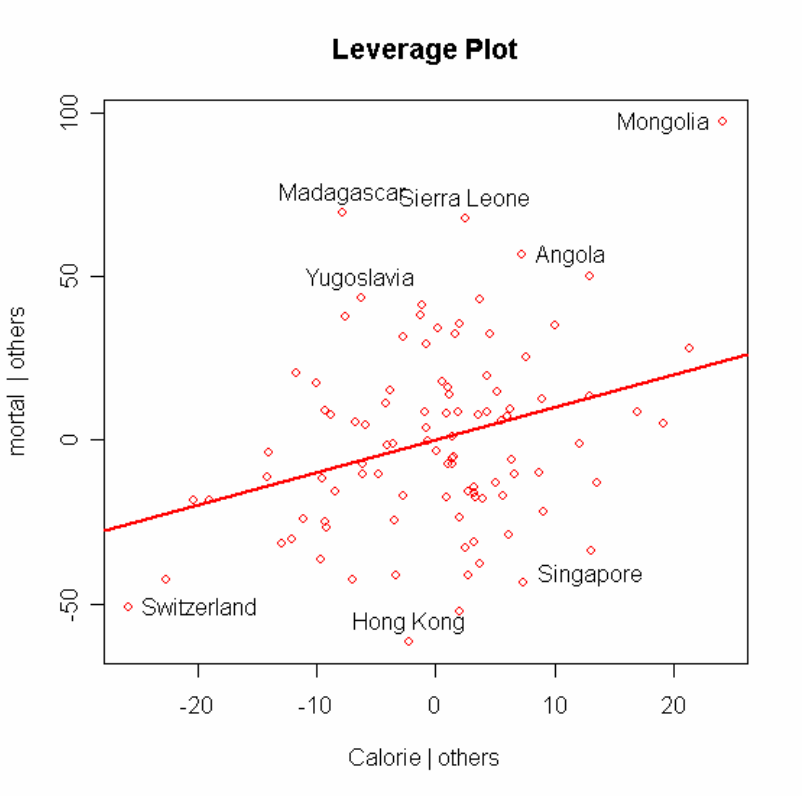
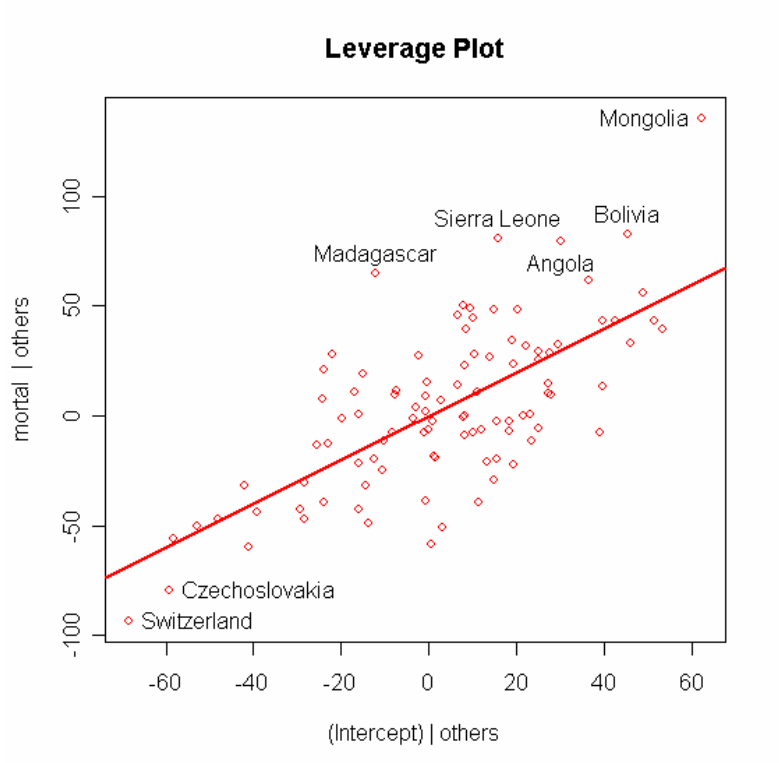

Leverage Plot
Mongolia
Madagascar
Sierra Leone
Angola
Yugoslavia
Singapore
Switzerland
Hong Kong
mortal | others
Calorie | others
100
50
0
-50
-20
-10
0
10
20

Fig.19


Leverage Plot
Mongolia
Sierra Leone
Bolivia
Madagascar
Angola
Czechoslovakia
Switzerland
mortal | others
(Intercept) | others
100
50
0
-50
-100
-60
-40
-20
0
20
40
60

Un punto influente (*influence*) è un punto che, se rimosso, produce un notevole cambiamento nella stima del modello. Un punto influente può o non può essere un outlier e può o non può avere un leverage elevato, ma, in generale ha almeno una di queste due caratteristiche.
Misure di influence sono date dai residui jacknife, dai cambiamenti nelle stime dei coefficienti di regressione (Dfbetas) e della varianza residua che si ottengono escludendo un punto dal stima e dalla distanza di Cook.

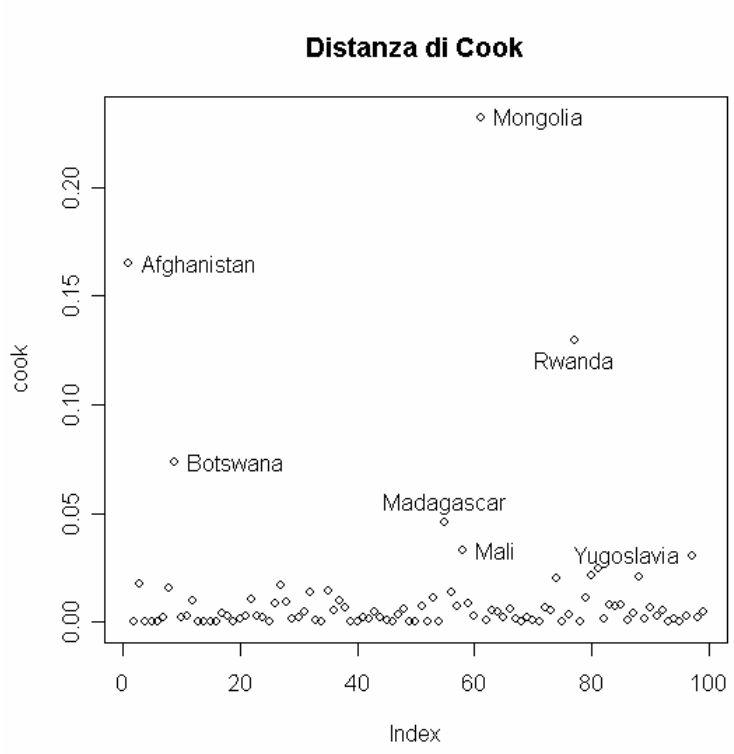
Per la distanza di Cook possiamo usare i comandi `cooks.distance()` nel package `stats` oppure `cookd()` del package `car`:

```
cook<-cooks.distance(fm)
library(car)
cookd(fm)

cook
        Afghanistan             Algeria              Angola
       1.646859e-01        3.901662e-06        1.736213e-02
          Argentina           Australia             Austria
       1.004911e-05        4.956263e-04        5.442490e-05
...
         Yugoslavia              Zambia            Zimbabwe
       3.041525e-02        2.044125e-03        4.926416e-03
```

Per ottenere il grafico delle distanze di Cook di tutte le osservazioni (Fig. 20):

```
plot(cook, main="Distanza di Cook")
identify(1:length(cook), cook, country)
```

Fig. 20



Il comando `influence.measures()` fornisce delle misure di influence: dfbeta, dffit, covratio, distanza di Cook e punti di leverage di tutte le osservazioni, mentre `summary(inf)` restituisce le informazioni sulle osservazioni potenzialmente punti di influenza:

```
inf<- influence.measures(fm)

summary(inf)
Potentially influential observations of
        lm(formula = mortal ~ Calorie + HS + popphys + popnurs, data =
mortalita) :

             dfb.1_ dfb.Calr dfb.HS  dfb.ppph dfb.ppnr dffit   cov.r   cook.d
Afghanistan  -0.19   0.18    -0.12   -0.31     0.85     0.92_*  1.10    0.16
Burkina Faso  0.00  -0.01     0.02    0.10    -0.04     0.11    1.25_*  0.00
Chad         -0.02   0.02     0.00    0.04     0.01     0.05    1.25_*  0.00
Ethiopia      0.00   0.03    -0.07   -0.26     0.07    -0.29    1.18_*  0.02
Hong Kong    -0.01  -0.06     0.08    0.11     0.05    -0.27    0.83_*  0.01
Japan         0.05  -0.07     0.10    0.02     0.01     0.11    1.23_*  0.00
Madagascar   -0.14   0.27    -0.40   -0.15    -0.05     0.50    0.70_*  0.05
Malawi        0.05  -0.02    -0.03    0.01    -0.25    -0.26    2.33_*  0.01
Mongolia      0.75  -0.87     1.03_*  0.04     0.06     1.12_*  0.78_*  0.23
Niger         0.00   0.00     0.00    0.01     0.00     0.01    1.18_*  0.00
Rwanda       -0.09   0.18    -0.27   -0.71     0.14    -0.82_*  0.99    0.13
Sierra Leone  0.15  -0.07    -0.10   -0.07    -0.03     0.34    0.79_*  0.02
             hat
Afghanistan   0.20_*
Burkina Faso  0.16_*
Chad          0.16_*
Ethiopia      0.13
Hong Kong     0.02
Japan         0.15
Madagascar    0.03
Malawi        0.55_*
Mongolia      0.13
Niger         0.11
Rwanda        0.14
Sierra Leone  0.02
```

L'asterisco indica valori superiori alle soglie a cui si fa di solito riferimento.

```
inf2<-lm.influence(fm)
attributes(inf2)
$names
[1] "hat" "coefficients"  "sigma" "wt.res"
```

fornisce alcuni indicatori utili per la diagnostica della regressione:
`hat`: vettore dei punti leverage (hat values)
`coefficients`: una matrice che nella riga i.esima contiene il cambiamento nella stima dei coefficienti che si ottiene quando la i.esima osservazione è esclusa dalla regressione
`sigma`: vettore nel quale l'i.esimo elemento contiene la stima della deviazione standard dei residui ottenuta quando l'i.esima osservazione è esclusa dalla regressione
`wt.res`: vettore di residui ponderati

```
inf2$coefficients
                      (Intercept)       Calorie            HS       popphys
Afghanistan          -3.7661930653  1.653506e-03 -2.694568e-02 -6.045375e-05
Algeria               0.0548150819 -2.305277e-05  2.257571e-04 -4.502968e-07
Angola                4.2511961677 -1.410599e-03 -2.969699e-03 -1.893932e-05
Argentina            -0.1049435324  5.769110e-05 -9.801877e-04 -1.270815e-07
                          popnurs
Afghanistan           4.999890e-04
Algeria               1.638432e-06
Angola               -1.801359e-05
```

```
Argentina            -4.690427e-07
...
Yugoslavia           -3.1653173414  1.250498e-03  2.281762e-02  1.187841e-05
Zambia               -0.6369505278  7.169046e-05  8.734842e-03  9.619019e-06
Zimbabwe             -1.3428310969  2.055354e-04  1.398590e-02  1.155545e-05

Yugoslavia            1.659224e-05
Zambia               -1.115905e-05
Zimbabwe              3.729537e-05
```

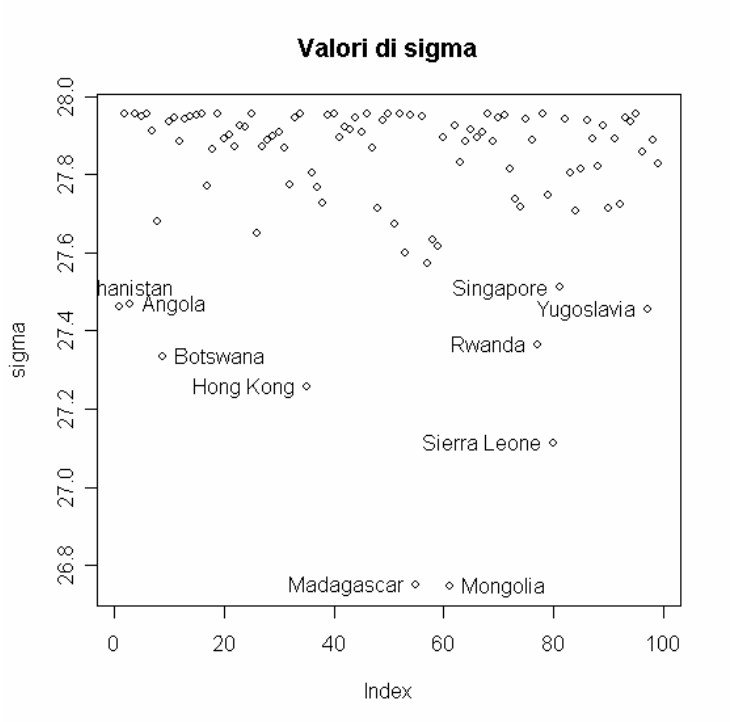
Analogo risultato si ottiene con il comando `dfbeta()`

```
inf2$sigma

         Afghanistan              Algeria               Angola
            27.46102             27.95585             27.46855
           Argentina            Australia              Austria
            27.95578             27.94745             27.95492
...
       Yugoslavia               Zambia             Zimbabwe
       27.45612             27.88919             27.82591
```

```
plot(inf2$sigma, main="Valori di sigma", ylab="sigma")
identify(1:length(inf2$sigma), inf2$sigma, country)
```

con questo grafico (Fig. 21) si mettono in evidenza le osservazioni che influiscono maggiormente nella stima di della deviazione standard dei residui.

Fig. 21



La statistica Covratio misura la variazione nel determinante della matrice delle covarianze delle stime quando si elimina la i.esima osservazione.

$$COVRATIO = [(det (\hat{\sigma}^2_{(i)} (X_{(i)}' X_{(i)})^{-1}))/(det (\hat{\sigma}^2 (X' X)^{-1}))]$$

Belsley, Kuh e Welsch[15] suggeriscono di tenere sotto controllo le osservazioni per le quali si verifica:

$$\left|\text{cov}\,ratio - 1\right| \geq \frac{3p}{n}$$

```
cvrat<-covratio(fm)
cvrat
         Afghanistan              Algeria               Angola
           1.1026644            1.1346772            0.9084493
...
          Yugoslavia               Zambia             Zimbabwe
           0.9210263            1.0535281            1.0353351

cvr<-abs(cvrat-1)
cvr[cvr>=3*p/n]

Burkina Faso         Chad     Ethiopia    Hong Kong        Japan   Madagascar
   0.2464143    0.2542238    0.1766181    0.1683165    0.2347396    0.3011276
      Malawi     Mongolia        Niger Sierra Leone
   1.3263259    0.2232538    0.1811959    0.2085120
```

Anche Dffits è una misura di influence: osservazioni con valori alti di dffits sono da considerarsi punti di influence; come valore soglia Belsley, Kuh e Welsch suggeriscono $2\sqrt{p/n}$

$$DFFITS = \frac{\hat{y}_i - \hat{y}_{(i)}}{\hat{\sigma}_{(i)}\sqrt{h_i}}$$

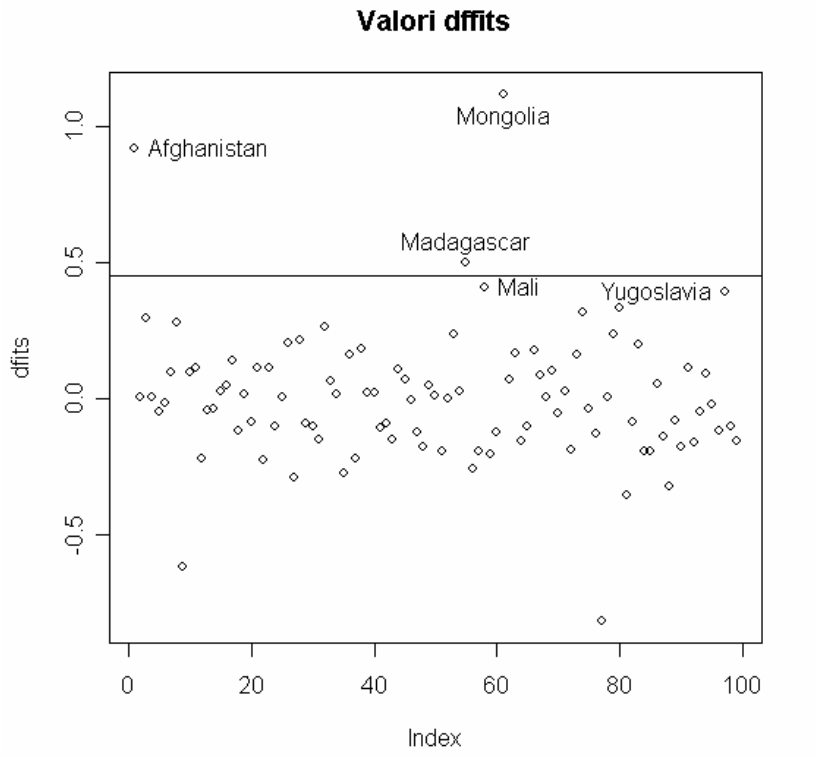
```
dfits<-dffits(fm)
dfits[dfits>2*sqrt(p/n)]

Afghanistan  Madagascar    Mongolia
  0.9188557   0.4985141   1.1199666
```

Nel Fig. 22 sono rappresentati graficamente i dffits di tutte le osservazioni con l'indicazione dei valori superiori alla soglia.

```
plot(dfits, main="Valori dffits")
abline(h=2*sqrt(p/n))
identify(1: length(dfits), dfits, country)
```

[15] D. A. BELSLEY, E. KUH, R. E. WELSCH, *Regression Diagnostics: Identifying Influential Data and Sources of Collinearity*, 2004

Fig. 22



Le statistiche Dfbetas sono una misura standardizzata della variazione nella stima di ciascun parametro della regressione calcolato eliminando la i.esima osservazione:

$$DFBETAS_j = \frac{\hat{\beta}_j - \hat{\beta}_{(i)j}}{\hat{\sigma}_{(i)}\sqrt{(X'X)_{jj}}}$$

dove $(\mathbf{X'X})_{jj}$ è l'elemento (j,j) della matrice $(\mathbf{X'X})^{-1}$ e $\hat{\beta}_{(i)j}$ è la stima del j.esimo coefficiente di regressione che si ottiene eliminando la i.esima osservazione.
In generale valori elevati di Dfbetas indicano osservazioni che influiscono molto nella stima dei parametri. Come valore soglia Belsley, Kuh e Welsch suggeriscono 2, oppure $2/\sqrt{n}$ che tiene conto del numero delle osservazioni.

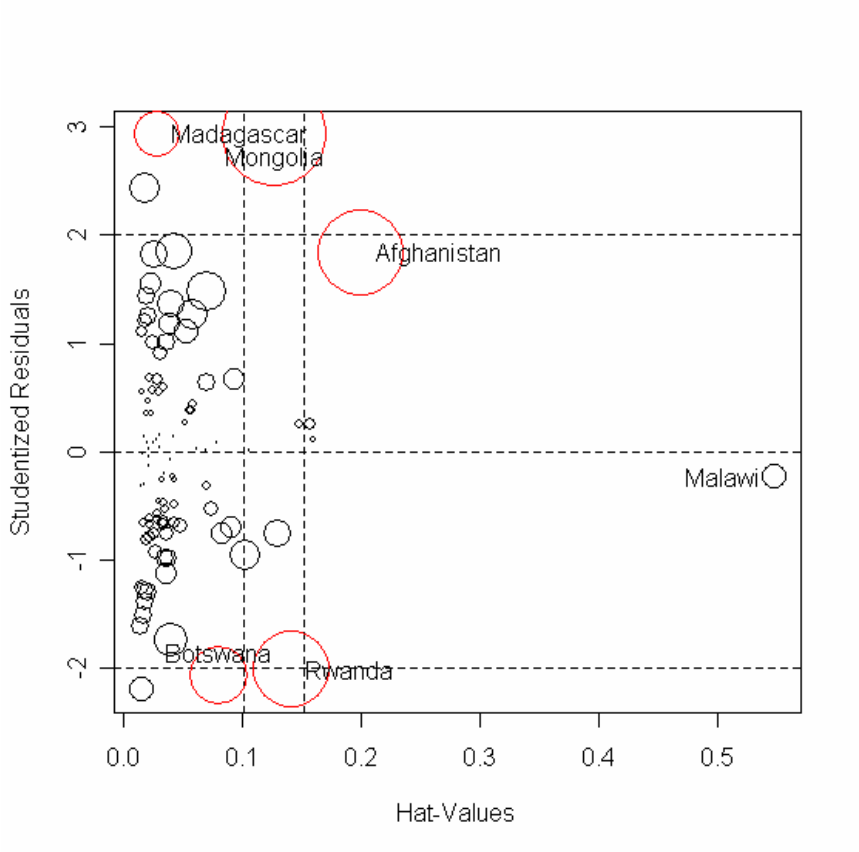
```
dfbet<-dfbetas(fm)

dfbet[,1][dfbet[,1]>2/sqrt(n)] ## intercetta
    Angola     Bolivia    Mongolia Switzerland
  0.2135726   0.2435633   0.7506161   0.2651442

dfbet[,2][dfbet[,2]>2/sqrt(n)] ## Calorie
Madagascar  Singapore
 0.2666322  0.2659709

dfbet[,3][dfbet[,3]>2/sqrt(n)] ## HS
  Mongolia Switzerland
  1.0326888   0.2012557

dfbet[,4][dfbet[,4]>2/sqrt(n)] ## popphys
```

```
   Guinea      Mali
0.2109819 0.3305104

dfbet[,5][dfbet[,5]>2/sqrt(n)] ## popnurs
Afghanistan
  0.8499893
```

la funzione `influence.plot()` genera un "bubble" plot dei residui studentizzati verso i punti leva (hat values), con le aree dei cerchi proporzionali alle distanze di Cook delle rispettive osservazioni. Nel Fig. 23 vengono tracciate due linee verticali in corrispondenza del doppio e del triplo del valore medio dei punti di leva, mentre in orizzontale (sulla scala dei residui studentizzati) sono tracciate tre linee in corrispondenza di -2, 0 e +2 che delimitano una banda di confidenza. In rosso sono evidenziate le osservazioni anomale, con maggiore influence:

```
library(car)
influence.plot(fm)
```

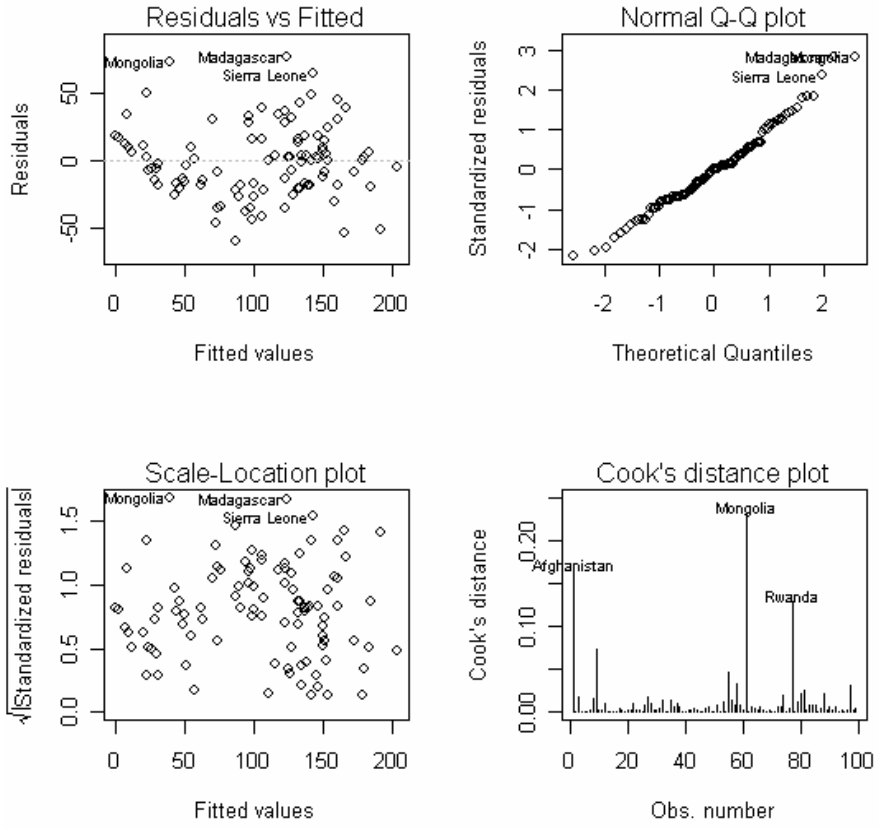
Fig. 23



Una sintesi grafica della diagnostica della regressione può essere ottenuta nel seguente modo (Fig. 24):

```
par(mfrow=c(2,2))
plot(fm)
```

Fig. 24



vengono stampati:
1) grafico dei residui contro i valori teorici: può rivelare la presenza di una residua dipendenza sistematica non individuata dal modello lineare stimato. In un buon modello questo grafico dovrebbe apparire come completamente casuale;
2) normal Q-Q plot dei residui standardizzati: verifica grafica dell'assunzione della normalità della componente erratica del modello lineare. Quanto più i punti che rappresentano i residui ordinati giacciono in prossimità della linea Q-Q, tanto più plausibile è detta assunzione;
3) grafico delle radici quadrate dei residui standardizzati contro i valori teorici: è utile nell'individuazione di valori outlier e per visualizzare strutture di dipendenza residue non individuate dal modello stimato;
4) grafico delle distanze di Cook: misure dell'influenza di ciascuna osservazione sulla stima dei parametri del modello.

Una sintesi dei principali indicatori utili ai fini della diagnostica della regressione si possono ottenere con il comando `ls.diag()`:

```
names(ls.diag(fm))
 [1] "std.dev"      "hat"          "std.res"      "stud.res"     "cooks"
 [6] "dfits"        "correlation"  "std.err"      "cov.scaled"   "cov.unscaled"
```

3.10 Trasformazioni di variabili

3.10.1 Trasformazioni della variabile risposta

Quando la variabile risposta non ha distribuzione normale, neppure approssimata, si pone il problema della trasformazione dei dati[16]. Box e Cox, nel 1964, hanno proposto un metodo iterativo per individuare quale trasformazione dei dati può meglio normalizzare la loro distribuzione. Il metodo ricorre a una famiglia di trasformazioni di potenze mediante la formula:

$$y_{tras} = \frac{y^{\lambda} - 1}{\lambda} \text{ per } \lambda \neq 0$$

$$y_{tras} = \log(y) \text{ per } \lambda = 0$$

con lambda che varia da -3 a +3.

Il valore di λ che meglio normalizza la distribuzione è quello che rende massima la funzione L (nota come log-likelihood function):

$$L = \frac{n-1}{2}\log(s^2_{TRAS}) + (\lambda - 1)\frac{n-1}{n}\sum_{i=1}^{n} y_i$$

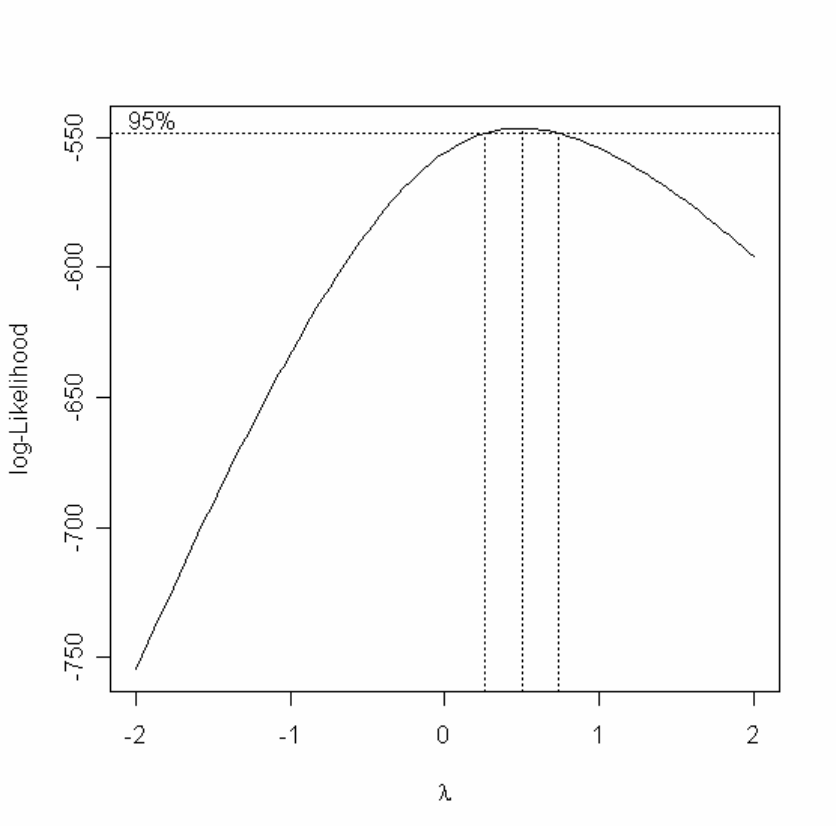
dove s^2_{TRAS} è la varianza dei dati trasformati;

Inoltre è possibile calcolare l'intervallo fiduciale di λ, entro il quale è conveniente scegliere la trasformazione più adeguata. Benché possa teoricamente assumere qualsiasi valore da –3 a +3 in una scala continua, in pratica λ ha significato pratico solo per alcuni valori.
Per stimare λ si usa il comando `boxcox()` del package `MASS`:
Si vedano anche i comandi `box.cox()`, `box.cox.powers()` e `box.cox.var()` nel package `car`.

```
library(MASS)
boxcox(fm, plotit=T) ## grafico
```

[16] Si veda: L. SOLIANI, *Statistica univariata e bivariata parametrica e non-parametrica per le discipline ambientali e biologiche*, 2005, cap. 13, pagg. 1-30

Fig. 25




```
boxcox(fm, plotit=F) ## valori di lamba e della log-verosimiglianza
$x
 [1] -2.0 -1.9 -1.8 -1.7 -1.6 -1.5 -1.4 -1.3 -1.2 -1.1 -1.0 -0.9 -0.8 -
0.7 -0.6
[16] -0.5 -0.4 -0.3 -0.2 -0.1  0.0  0.1  0.2  0.3  0.4  0.5  0.6  0.7
0.8  0.9
[31]  1.0  1.1  1.2  1.3  1.4  1.5  1.6  1.7  1.8  1.9  2.0

$y
 [1]  -754.7419  -741.5637  -728.5703  -715.7743  -703.1903  -690.8348  -
678.7273
 [8]  -666.8905  -655.3506  -644.1385  -633.2898  -622.8455  -612.8522  -
603.3620
[15]  -594.4315  -586.1205  -578.4897  -571.5974  -565.4956  -560.2261  -
555.8165
[22]  -552.2774  -549.6017  -547.7646  -546.7266  -546.4364  -546.8353  -
547.8614
[29]  -549.4525  -551.5490  -554.0958  -557.0431  -560.3465  -563.9673  -
567.8718
[36] -572.0311 -576.4202 -581.0177 -585.8050 -590.7661 -595.8870
```

come si vede dal grafico il valore ottimale per λ è 0.5. Stimiamo il modello di regressione con la variabile trasformata:

```
trasy<-((mortalal^0.5)-1)/0.5

fmtras<-lm(trasy~Calorie+HS+popphys+popnurs, data=mortalita)
summary(fmtras)
```

```
Call:
lm(formula = trasy ~ Calorie + HS + popphys + popnurs, data = mortalita)

Residuals:
      Min        1Q    Median        3Q       Max
-7.267991 -1.695006 -0.004735  1.489943  9.221714

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.842e+01  2.056e+00  13.821  < 2e-16 ***
Calorie     -3.636e-03  9.579e-04  -3.796  0.00026 ***
HS          -1.509e-01  2.237e-02  -6.748 1.22e-09 ***
popphys      4.206e-05  1.990e-05   2.114  0.03715 *
popnurs      1.072e-04  6.078e-05   1.763  0.08116 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.838 on 94 degrees of freedom
Multiple R-Squared: 0.8129,     Adjusted R-squared: 0.8049
F-statistic: 102.1 on 4 and 94 DF,  p-value: < 2.2e-16
```

3.10.2 Trasformazioni delle variabili esplicative

Talvolta può essere necessario ricorrere anche a delle trasformazioni nei regressori. Ad esempio quando tra due variabili esiste una relazione di tipo inverso secondo questo modello:

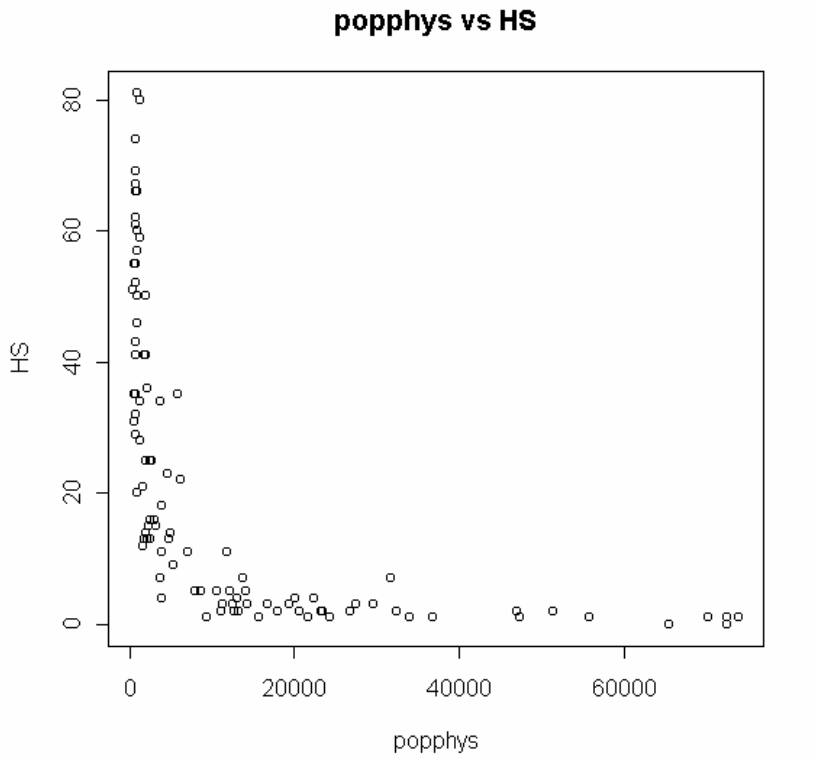
$$y = \beta_0 + \beta_1 \frac{1}{x} + \varepsilon$$

basta porre $z = \frac{1}{x}$ e far regredire y su z

Facendo riferimento ad dataframe `mortalità`, tracciamo il grafico tra la variabile `popphys` e la variabile `HS` (Fig. 25). Come si vede la relazione è di tipo inverso (iperbolica).

```
attach(mortalita)
plot(popphys, HS, main="popphys vs HS")
```

Fig. 25



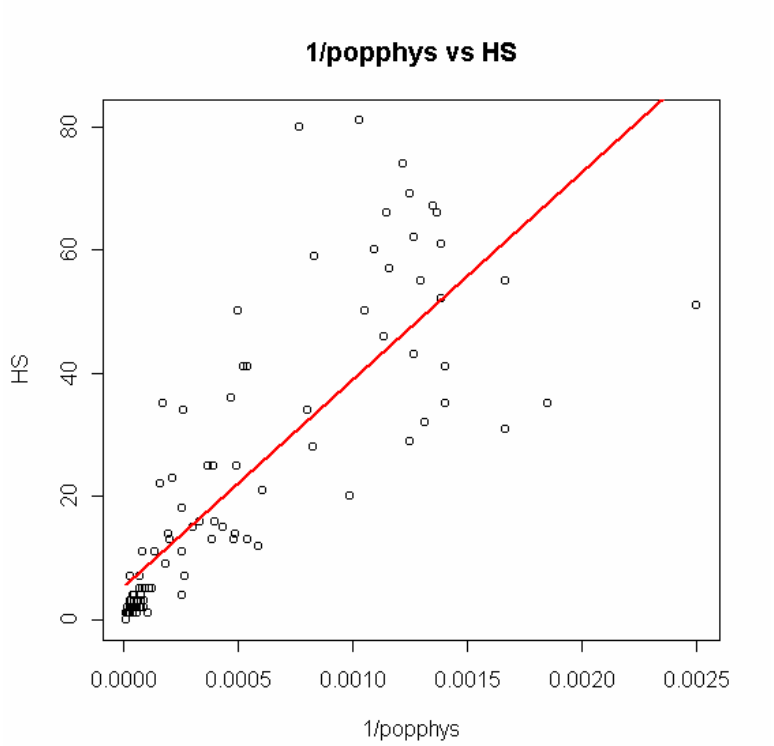
Possiamo continuare ad usare la funzione `lm()` avendo cura di inserire come regressore `I(1/popphys)`:

```
invfm<-lm(HS~I(1/popphys), data=mortalita)
summary(invfm)

Call:
lm(formula = HS ~ I(1/popphys), data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-38.422  -5.290  -3.933   3.619  48.874

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)      5.217      1.800   2.897  0.00465 **
I(1/popphys) 33681.928   2450.533  13.745  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 13.37 on 97 degrees of freedom
Multiple R-Squared: 0.6607,     Adjusted R-squared: 0.6572
F-statistic: 188.9 on 1 and 97 DF,  p-value: < 2.2e-16
```

La relazione è facilmente linearizzata (Fig. 26):

```
plot(1/popphys, HS, main="1/popphys vs HS")
library(car)
reg.line(invfm)## traccia la retta di regressione
```

Fig. 26



Si è usato il comando `reg.line()` del package `car` per tracciare la retta di regressione sullo scatterplot al posto di `abline()`.

Un altro modello che si incontra spesso è quello geometrico[17]:

$$y = \beta_0 x^{\beta_1} \varepsilon$$

che si trasforma in un modello lineare prendendo i logaritmi di entrambi i termini:

$$\log(y) = \log(\beta_0) + \beta_1 \log(x) + \log(\varepsilon)$$

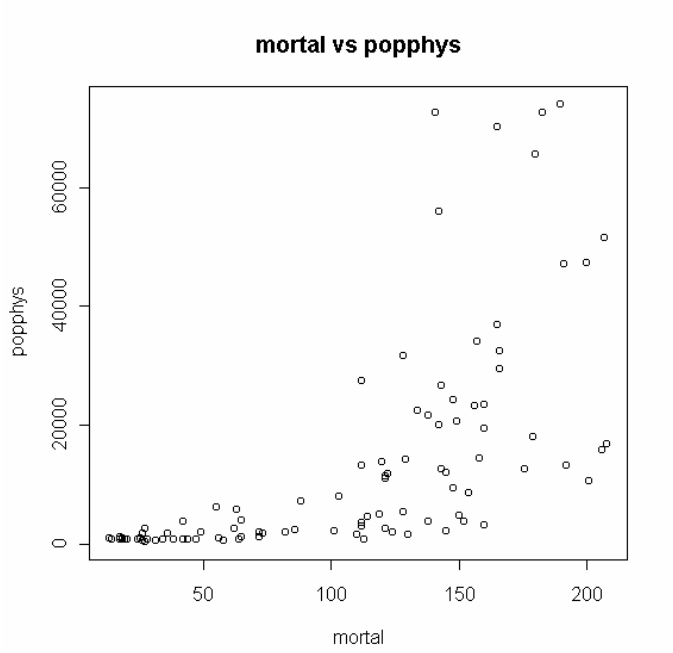
E' il caso della relazione tra la variabile `mortal` e la variabile `popphys` (Fig. 27):

```
attach(mortalita)
plot(mortal, popphys, main="mortal vs popphys")
logfm<-lm(log(popphys)~log(mortal), data=mortalita)
summary(logfm)

Call:
lm(formula = log(popphys) ~ log(mortal), data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-2.3866 -0.7458  0.1404  0.7042  1.8737

Coefficients:
```

[17] Occorre notare come in questo modello l'errore è di tipo moltiplicativo. Per una panoramica sulle funzioni di regressione linerarizzabili si veda: F. DEL VECCHIO, *Statistica per la ricerca sociale*, 1992, pagg. 357-362

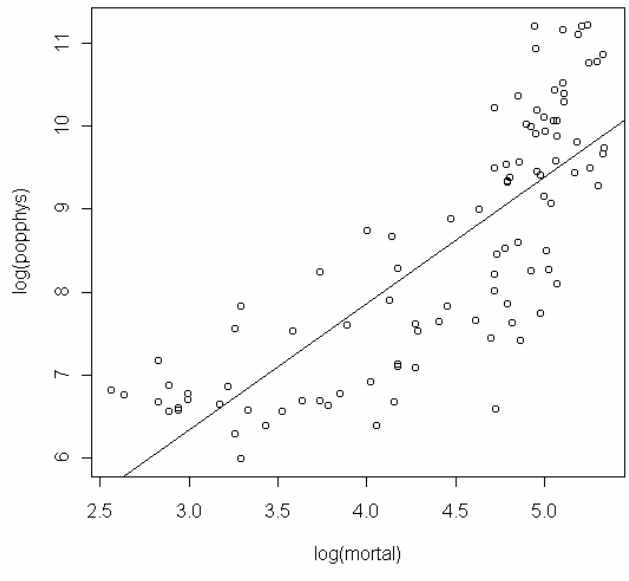
```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.7675     0.5223   3.384  0.00103 **
log(mortal)    1.5256     0.1166  13.081  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9061 on 97 degrees of freedom
Multiple R-Squared: 0.6382,     Adjusted R-squared: 0.6345
F-statistic: 171.1 on 1 and 97 DF,  p-value: < 2.2e-16
```

Fig. 27




```
plot(log(mortal), log(popphys)) ## grafico su scala doppiologaritmica
abline(logfm)
```

Fig. 28



Un altro tipo di trasformazione che si può appliccare è quella di Box-Tidwell:

```
library(car)
box.tidwell(mortal~ HS + popphys + popnurs, data=mortalita)
                      HS  popphys  popnurs
Initial Power    0.04074  0.80614  0.48230
Score Statistic  3.34723 -0.22191 -0.64226
p-value          0.00082  0.82438  0.52071
MLE of Power    14.18080 -0.13268 -0.29269

iterations =  16
```

3.11 Regressione polinomiale

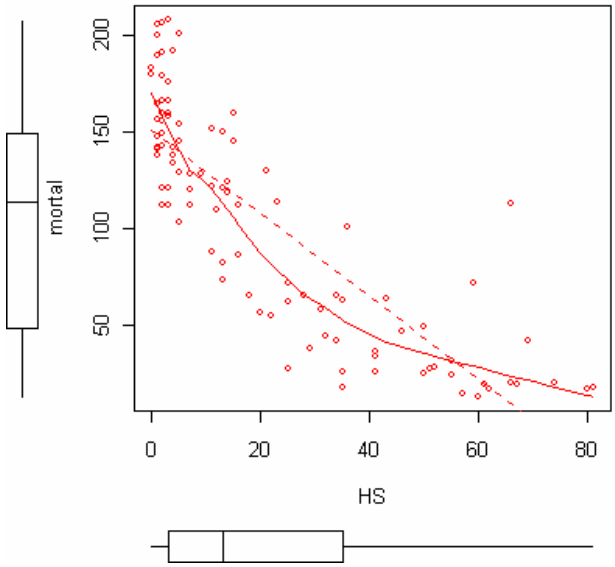
Nel modello lineare risulta rientrare anche la regressione polinomiale, ossia quella nella quale compaiono alcuni regressori con grado uguale o superiore a 2. Il modello in questione continua ad essere lineare nei parametri. Ad esempio, un modello di regressione parabolica di secondo grado si presenta in questo modo:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \varepsilon$$

e può essere facilmente stimato introducendo il termine di secondo grado nel modello di regressione. La relazione tra la variabile `mortal` e la variabile `HS` sembra essere di tipo parabolico (Graff. 29 e 30). Per tracciare il grafico usiamo il comando `scatterplot()` del package `car`: questa funzione ci consente di ottenere un grafico con il boxplot per entrambe le variabili in ascissa e ordinata (ai margini degli assi), la retta di regressione e la curva stimata con la tecnica della regressione locale:

```
library(car)
scatterplot(mortal ~ HS , data=mortalita)


fmpol<-lm(mortal~HS+I(HS^2), data=mortalita)
summary(fmpol)

Call:
lm(formula = mortal ~ HS + I(HS^2), data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-48.100 -15.560  -3.316  12.469  85.907

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 167.660917   4.468277  37.522  < 2e-16 ***
HS           -4.661347   0.400631 -11.635  < 2e-16 ***
I(HS^2)       0.038357   0.005863   6.543 2.94e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.54 on 96 degrees of freedom
Multiple R-Squared: 0.7982,     Adjusted R-squared: 0.794
F-statistic: 189.9 on 2 and 96 DF,  p-value: < 2.2e-16
```

Fig. 29



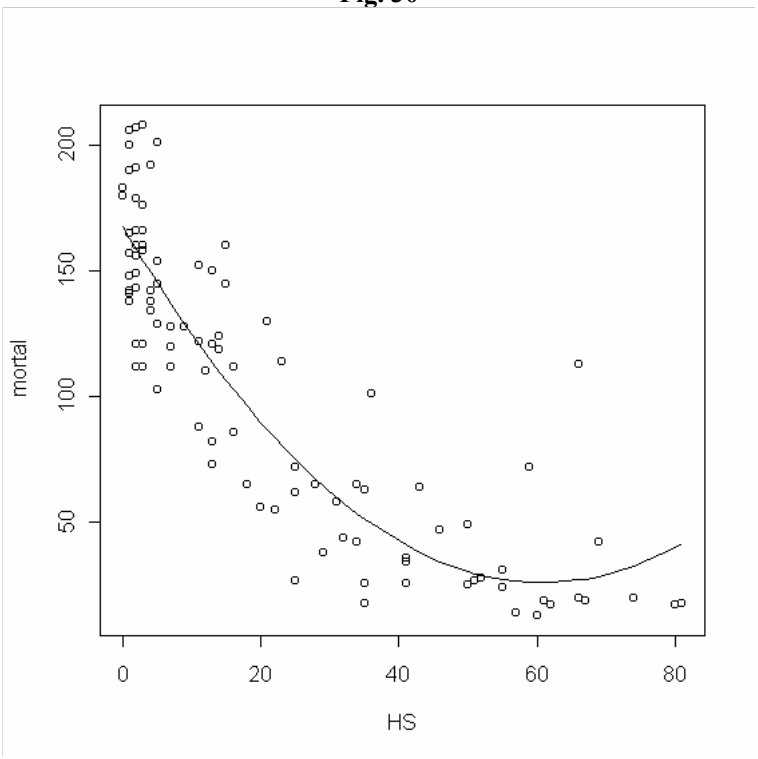
Come si può vedere, il termine quadratico risulta significativo. Una conferma viene dalla tabella ANOVA:

```
anova(fmpol, lm(mortal~HS, data=mortalita))

Analysis of Variance Table

Model 1: mortal ~ HS + I(HS^2)
Model 2: mortal ~ HS
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1     96  67641
2     97  97803 -1    -30161 42.807 2.944e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Tracciamo il grafico con la curva di regressione parabolica sovraimpressa:

```
HSfit<-sort(HS)
mortalfit<-fmpol$coef[1]+fmpol$coef[2]*HSfit+fmpol$coef[3]*HSfit^2
attach(mortalita)
plot(HS, mortal)
lines(HSfit, mortalfit)
```

Fig. 30



Per regressioni polinomiali di qualsiasi grado si può usare il comando `poly()`, nel quale va specificata la variabile da inserire come regressore polinomiale e il grado del polinomio:

```
fmpol3<-lm(mortal~ poly(HS,3), data=mortalita)
summary(fmpol3)

Call:
lm(formula = mortal ~ poly(HS, 3), data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-48.327 -17.327  -6.957  12.329  83.134

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   104.414      2.654  39.346  < 2e-16 ***
poly(HS, 3)1 -487.261     26.405 -18.454  < 2e-16 ***
poly(HS, 3)2  173.671     26.405   6.577 2.59e-09 ***
poly(HS, 3)3  -37.515     26.405  -1.421    0.159
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.4 on 95 degrees of freedom
Multiple R-Squared: 0.8024,     Adjusted R-squared: 0.7962
F-statistic: 128.6 on 3 and 95 DF,  p-value: < 2.2e-16
```

Se le variabili sono due o più, possiamo considerare un modello completo, il quale, oltre ai termini quadratici, prende in considerazione l'interazione tra variabili:

```
fmpolcomp<-lm(mortal~ HS + I(HS^2)+Calorie+I(Calorie^2)+HS:Calorie,
data=mortalita)
```

```
summary(fmpolcomp)

Call:
lm(formula = mortal ~ HS + I(HS^2) + Calorie + I(Calorie^2) +
    HS:Calorie, data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-48.130 -16.043  -4.128  13.599  64.277

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.186e+02  8.758e+01   2.496   0.0143 *
HS           -3.430e+00  1.416e+00  -2.422   0.0174 *
I(HS^2)       3.888e-02  9.112e-03   4.267 4.76e-05 ***
Calorie      -3.792e-02  8.089e-02  -0.469   0.6403
I(Calorie^2)  5.778e-06  1.897e-05   0.305   0.7613
HS:Calorie   -3.627e-04  6.948e-04  -0.522   0.6029
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.29 on 93 degrees of freedom
Multiple R-Squared: 0.8082,     Adjusted R-squared: 0.7979
F-statistic: 78.38 on 5 and 93 DF,  p-value: < 2.2e-16
```

Tuttavia i nuovi termini introdotti nella regressione sono risultano essere significativi, e, per tanto, appare non necessario prenderli in considerazione.

3.12 Segmented regression

Talvolta, si può avere ragione di ritenere che per un dato set di dati si debbano applicare due modelli di regressione diversi, a seconda, ad esempio, che i valori della variabile esplicativa siano inferiori o superiori ad un valore soglia. In questa circostanza si applica la *segmented regression* detta anche *broken stick regression*[18]. Esaminiamo la relazione esistente tra la variabile `mortal` e la variabile `HS` vista nel paragrafo precedente:

```
attach(mortalita)
plot(HS, mortal, main="Mortal vs HS")
```

Nel Fig. 31 si vede che la pendenza della retta di regressione è più elevata per valori di `HS` inferiori a 20, mentre per valori superiori la pendenza è minore. Per stimare la *segmented regression* introduciamo le seguenti funzioni:

$$B_l(x) = \begin{cases} c - x & se\ x < c \\ 0 & altrimenti \end{cases}$$

$$B_r(x) = \begin{cases} x - c & se\ x > c \\ 0 & altrimenti \end{cases}$$

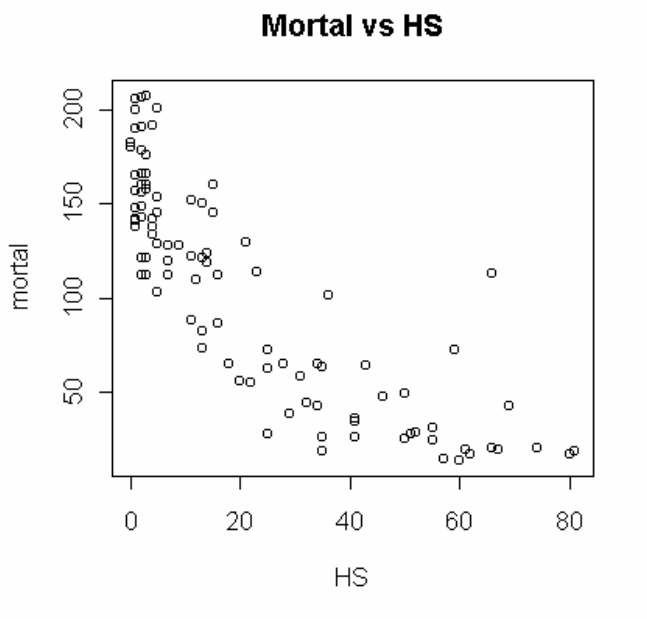
dove c è il valore soglia (*break point*). Per tanto il modello diventa:

$$y = \beta_0 + \beta_1 B_l(x) + \beta_2 B_r(x) + \varepsilon$$

[18] Si veda J. J. FARAWAY, *op. cit*., pagg. 98-100

che può essere stimato con la regressione ordinaria. Vediamo come operare in R:

Fig. 31



Innanzitutto introduciamo le due funzioni:

```
lhs <- function(x) ifelse(x < 20,20-x,0)
rhs <- function(x) ifelse(x < 20,0,20-x)
```

e stimiamo il modello:

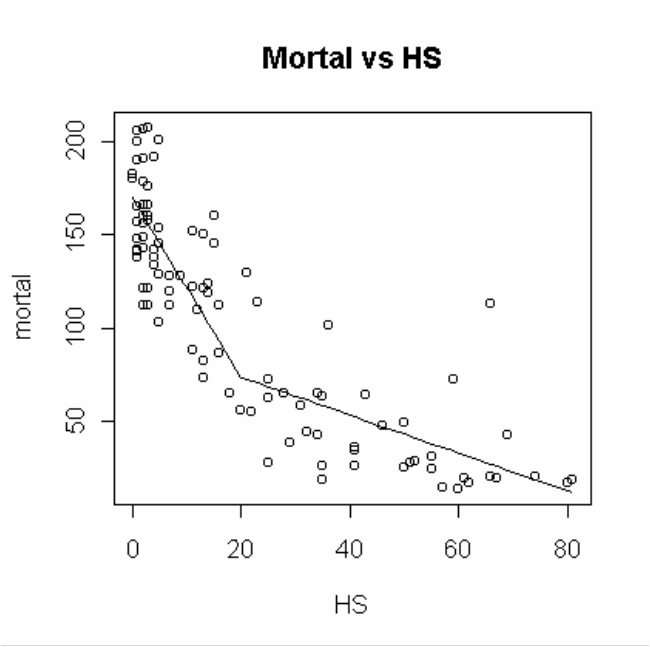
```
fmbr<-lm(mortal~lhs(HS)+rhs(HS), data=mortalita)
summary(fmbr)

Call:
lm(formula = mortal ~ lhs(HS) + rhs(HS), data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-48.377 -17.294  -4.377  13.371  86.478

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  73.3850     6.1438  11.945  < 2e-16 ***
lhs(HS)       4.8329     0.4423  10.928  < 2e-16 ***
rhs(HS)       1.0188     0.2164   4.709  8.4e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.87 on 96 degrees of freedom
Multiple R-Squared: 0.7933,     Adjusted R-squared: 0.789
F-statistic: 184.2 on 2 and 96 DF,  p-value: < 2.2e-16
```

Tracciamo il grafico (Fig. 32):

```
HSfit<-sort(HS)
```

```
pmortal<- fmbr$coef[1]+fmbr$coef[2]*lhs(HSfit)+fmbr$coef[3]*rhs(HSfit)
plot(HS, mortal, main="Mortal vs HS")
lines(HSfit,pmortal)
```

Fig. 32



Questo tipo di regressione trova applicazione, ad esempio, quando nella variabile esplicativa sono presenti due gruppi distinti di osservazioni. Nel nostro caso vi sono paesi sviluppati, con alti valori di `HS` (grado di istruzione) e paesi in via di sviluppo con valori bassi.
Un problema più complesso si pone quando è necessario stimare il valore soglia sulla base dei dati osservati[19]. In questa circostanza si può ricorrere al package `segmented`[20]:

```
library(segmented)
fmseg<-segmented(lm(mortal ~ HS, data=mortalita), Z=HS, psi=10)
summary(fmseg)

        ***Regression Model with Segmented Relationship(s)***

Call:
segmented.lm(obj = lm(mortal ~ HS, data = mortalita), Z = HS,
    psi = 10)

Estimated Break-Point(s):
   Est. St.Err
28.570  3.573

t value for the gap-variable(s) V:  -0.2236255

Meaningful coefficients of the linear terms:
             Estimate Std. Error    t value
(Intercept) 167.215008  4.5210205  36.986120
HS           -4.061351  0.3985538 -10.190222
U.HS          3.508408  0.5153775   6.807453
```

[19] V. A MUGGEO, *Estimating regression models with unknown break-points*, Stat Med. Oct 2003; 15; 22(19):3055-71
[20] http://cran.r-project.org/src/contrib/Descriptions/segmented.html

```
Residual standard error: 26.24 on 95 degrees of freedom
Multiple R-Squared: 0.8049,  Adjusted R-squared: 0.7988
Convergence attained in 5 iterations with relative change -4.108463e-05
```

Con il comando `segmented()`, specificando il modello di regressione stimato normalmente (oggetto di classe `lm`), la variabile esplicativa che ha la relazione di tipo segmentata e una stima iniziale del break point, abbiamo ottenuto il risultato di sopra con una stima del break point=`28.570`. La regressione segmentata può essere applicata anche nel caso di più variabili che andranno specificate, come matrice, nell'argomento `Z`. Il comando `segmented.slope()` ci permette di ottenere le stime dei coefficienti di regressione del modello segmentato con relativi intervalli di confidenza:

```
slope.segmented(fmseg)
$HS
             Est.   St.Err.    t value CI(95%).l   CI(95%).u
slope1 -4.0613514 0.3985538 -10.190222 -4.842502 -3.28020034
slope2 -0.5529434 0.3267550  -1.692226 -1.193371  0.08748472
```

3.13 Dummy variables

In un modello di regressione si possono introdurre anche delle variabili esplicative di tipo qualitativo dicotomico attraverso le *dummy variables* o variabili di comodo. Si tratta di variabili indicatrici che assumono valore 1 se la caratteristica qualitativa è posseduta e valore 0 se non è posseduta. Inoltre è possibile prendere in considerazione variabili qualitative politomiche attraverso l'introduzione di tante *dummy variables* quanti sono i livelli del carattere qualitativo. Occorre osservare che l'introduzione di variabili di comodo potrebbe causare multicollinearità, in quanto si potrebbe venire a determinare una dipendenza lineare tra le colonne della matrice dei regressori.
In una regressione semplice le *dummy variables* possono essere introdotte in due modi:

$$Y = \beta_0 + \beta_1 X + \beta_2 D + \varepsilon$$

quando D=0 l'intercetta è pari a β_0, mentre quando D=1 l'intercetta diventa $\beta_0 + \beta_2$. La variabile di comodo, quindi, introduce una variazione dell'intercetta. Nell'altro modello invece:

$$Y = \beta_0 + \beta_1 X + \beta_2 DX + \varepsilon$$

quando D=0 l'inclinazione è pari a β_1, mentre quando D=1 l'inclinazione diventa $\beta_1 + \beta_2$. La variabile di comodo, quindi, introduce una variazione della pendenza della retta di regressione.
Introduciamo nel dataframe `mortalità` la *dummy variable* `african` che assume valore 1 per i paesi africani e 0 per gli altri. Stimiamo i parametri del primo modello:

```
fmD<-lm(mortal~Calorie+HS+popphys+popnurs+african, data=mortalita)
summary(fmD)

Call:
lm(formula = mortal ~ Calorie + HS + popphys + popnurs + african,
    data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-64.715 -16.374  -2.735  13.686  69.541

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

```
(Intercept)  1.786e+02  1.958e+01   9.119 1.49e-14 ***
Calorie     -2.980e-02  8.984e-03  -3.317  0.00130 **
HS          -1.005e+00  2.221e-01  -4.523 1.80e-05 ***
popphys      2.962e-04  2.111e-04   1.403  0.16391
popnurs      1.532e-03  5.753e-04   2.664  0.00911 **
african      2.495e+01  8.044e+00   3.102  0.00255 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.61 on 93 degrees of freedom
Multiple R-Squared: 0.8035,     Adjusted R-squared: 0.7929
F-statistic: 76.06 on 5 and 93 DF,  p-value: < 2.2e-16
```

Mentre per il secondo modello si ottengono questi risultati:

```
fmD2<-lm(mortal~HS+Calorie+popphys+popnurs+african:HS, data=mortalita)
summary(fmD2)

Call:
lm(formula = mortal ~ HS + Calorie + popphys + popnurs + african:HS,
    data = mortalita)

Residuals:
    Min      1Q  Median      3Q     Max
-60.499 -16.011  -5.429  13.084  70.619

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.815e+02  1.955e+01   9.287 6.60e-15 ***
HS          -1.050e+00  2.196e-01  -4.781 6.53e-06 ***
Calorie     -3.035e-02  9.030e-03  -3.361 0.001128 **
popphys      6.624e-04  1.886e-04   3.512 0.000688 ***
popnurs      1.454e-03  5.753e-04   2.526 0.013214 *
HS:african   3.453e+00  1.173e+00   2.944 0.004090 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 26.74 on 93 degrees of freedom
Multiple R-Squared: 0.8017,     Adjusted R-squared: 0.791
F-statistic: 75.18 on 5 and 93 DF,  p-value: < 2.2e-16
```

La *dummy variable* `african` apporta un contributo significativo nella regressione in entrambe i modelli, tuttavia il secondo modello sembra essere più adatto poiché appare molto significativa l'interazione tra `HS` e `african`. Vogliamo fare riferimento in questo contesto all'argomento `subset` del comando `lm()` che permette di stimare i parametri della regressione su un sottoinsieme di dati presenti nel dataframe specificando un criterio di selezione che opera su una o più variabili. Stimiamo i coefficienti di regressione considerando i dati di tutti i paesi presenti, solo quelli dei paesi africani e solo quelli dei paesi non africani e confrontiamoli:

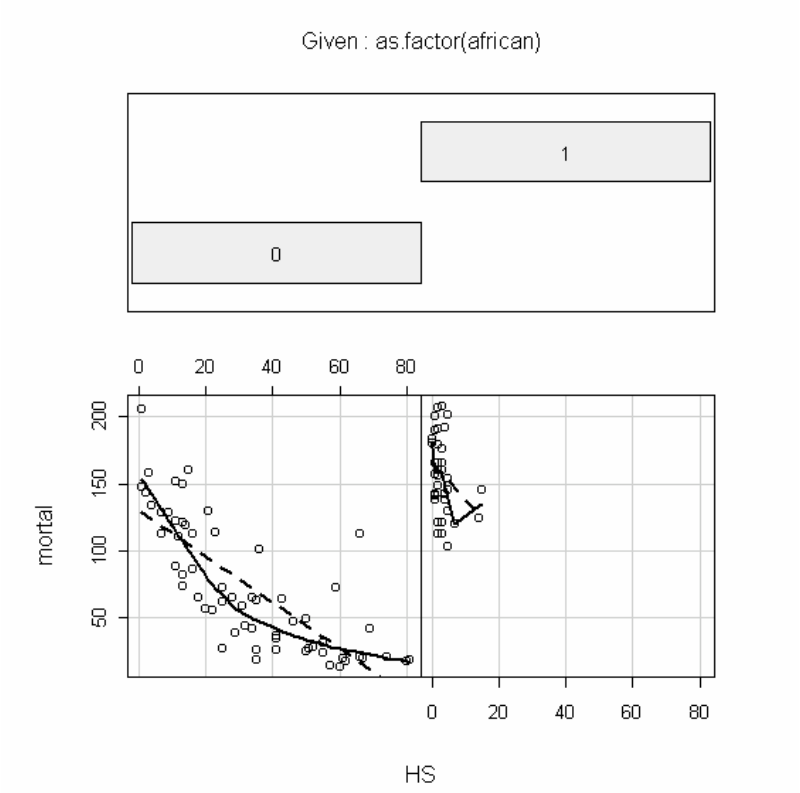
```
fmall<-lm(mortal ~ HS + Calorie + popphys + popnurs, data=mortalita)
fmafr<-lm(mortal ~ HS + Calorie + popphys + popnurs, data=mortalita,
subset=african==1)
fmoth<-lm(mortal ~ HS + Calorie + popphys + popnurs, data=mortalita,
subset=african==0)
confronto<-data.frame(coef(fmall), coef(fmafr), coef(fmoth))

confronto
```

```
              coef.fmall.   coef.fmafr.   coef.fmoth.
(Intercept) 189.097165167  1.554679e+02  1.751816e+02
HS           -1.231002661 -5.515283e-01 -7.095568e-01
Calorie      -0.029567944 -6.022272e-03 -3.441959e-02
popphys       0.000602156  4.284466e-04  4.328213e-04
popnurs       0.001292906  4.992023e-04  3.718801e-03
```

Tracciamo lo scatterplot tra `mortal` e `HS` per i paesi africani e quelli non africani (Fig. 33):

```
coplot(mortal ~ HS |as.factor(african), data=mortalita, panel=panel.car)
```

Fig. 33



3.14 Correlazione parziale

La correlazione parziale misura la correlazione lineare esistente tra due variabili al netto dell'influenza che su queste possono avere altre variabili. Ad esempio, abbiamo un gruppo di k variabili X_1, X_2, … X_k tra loro correlate e vogliamo verificare l'esistenza di correlazione tra X_1 e X_2 eliminando l'influenza che potrebbero avere le rimanenti k-2 variabili. Questo problema viene risolto calcolando il coefficiente di correlazione tra i residui della regressione di X_1 su X_3, … X_k e i residui della regressione di X_2 su X_3, … X_k.
Prendiamo in considerazione i dati sull'inquinamento già usati nei paragrafi precedenti e calcoliamo la matrice di correlazione tra le variabili :

```
cor(inquinamento)
                 CO2     energy     export   GDPgrowth   popgrowth         GNI
CO2        1.0000000  0.8878076  0.3585020  0.18459161 -0.17879523  0.59615769
energy     0.8878076  1.0000000  0.3116325  0.15032981 -0.19961188  0.74256850
export     0.3585020  0.3116325  1.0000000  0.40311260 -0.12600431  0.25716070
GDPgrowth  0.1845916  0.1503298  0.4031126  1.00000000 -0.05398184  0.03133970
popgrowth -0.1787952 -0.1996119 -0.1260043 -0.05398184  1.00000000 -0.22107226
GNI        0.5961577  0.7425685  0.2571607  0.03133970 -0.22107226  1.00000000
```

Vogliamo calcolare il coefficiente di correlazione parziale tra le variabili `CO2` e `energy`:

```
temp1<-lm(CO2 ~ export+GDPgrowth+popgrowth+GNI,data=inquinamento)
temp2<-lm(energy ~ export+GDPgrowth+popgrowth+GNI,data=inquinamento)
cor(temp1$res, temp2$res)
[1] 0.818842
```

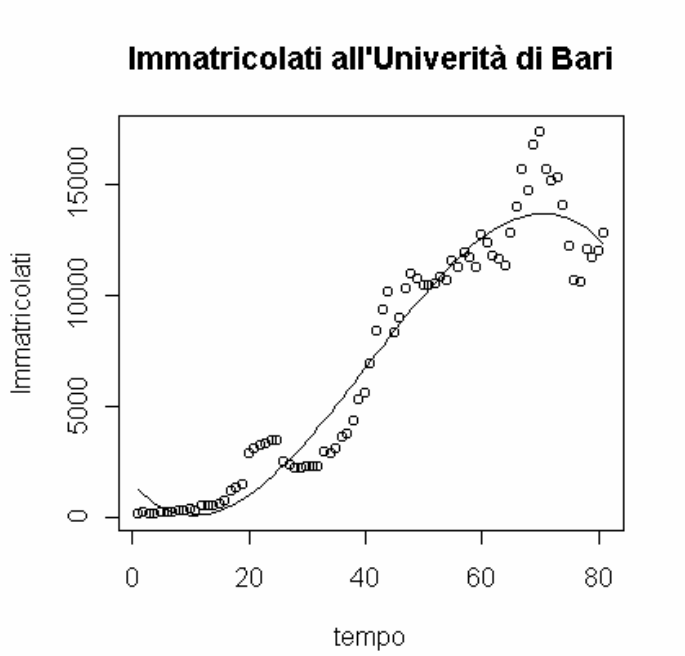
come si vede il coefficiente di correlazione parziale (=0.818842) è inferiore al coefficiente di correlazione (=0.8878076), infatti su entrambe le variabili c'è un'influenza abbastanza forte della variabile `GNI`.

3.15 Splines regression

Le splines sono delle funzioni interpolanti di grado m e nodi n. Una funzione interpolante è una funzione che permette di valutare un'approssimazione del valore di una data funzione di cui si conoscono i valori solo per un insieme discreto e finito di punti x_i con i = 1,2,…,n, in un punto x diverso da x_i. Le funzioni spline sono funzioni polinomiali a tratti che si ottengono suddividendo l'intervallo di definizione della funzione di partenza in sottointervalli e definendo in ognuno di essi un polinomio di grado opportuno, solitamente non troppo alto. Le splines solitamente più utilizzate sono quelle cubiche interpolanti nei nodi, ovvero spline di 3° grado i cui nodi corrispondono ai valori noti della funzione di partenza. Possiamo usare le splines nella regressione qualora l'andamento della funzione di regressione è piuttosto irregolare, come nell'esempio che segue. Si tratta della serie storica degli immatricolati all'Università degli studi di Bari dal 1925 al 2005. Tali dati sono contenuti nel dataframe `universita`. Usiamo dapprima la semplice regressione polinomiale (Fig. 34):

```
fmpol<-lm(Immatricolati~ poly(tempo, 5), data=universita)
summary(fmpol)

Call:
lm(formula = Immatricolati ~ poly(tempo, 5), data = universita)

Residuals:
    Min      1Q  Median      3Q     Max
-2547.9  -607.3  -145.7   582.5  3208.2

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)       6972.4      136.1  51.247  < 2e-16 ***
poly(tempo, 5)1  45271.2     1224.5  36.971  < 2e-16 ***
poly(tempo, 5)2   -901.2     1224.5  -0.736  0.46405
poly(tempo, 5)3 -11363.2     1224.5  -9.280 4.38e-14 ***
poly(tempo, 5)4  -4275.0     1224.5  -3.491  0.00081 ***
poly(tempo, 5)5    665.1     1224.5   0.543  0.58863
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1225 on 75 degrees of freedom
Multiple R-Squared: 0.9513,     Adjusted R-squared: 0.9481
F-statistic: 293.2 on 5 and 75 DF,  p-value: < 2.2e-16

attach(universita)
plot(tempo, Immatricolati,  main="Immatricolati all'Univerità di Bari")
lines(tempo, fitted(fmpol))
```

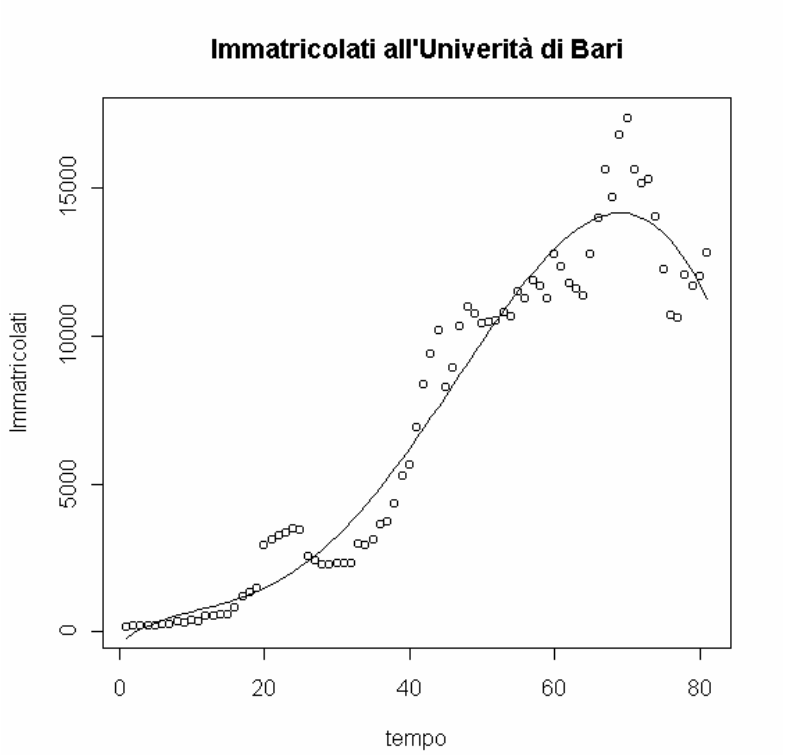
Fig. 34



tale tipo di regressione ha, però, l'inconveniente che ciascun punto influisce sulla stima complessiva del modello. Tale inconveniente può essere superato usando le B-splines, infatti il metodo della *broken stick regression* localizza l'influenza dei singoli valori dei dati ad un solo tratto. Le B-splines in `R` si trovano nel package `splines`:

```
library(splines)
fm<-lm(Immatricolati~ bs(tempo, df=3), data=universita)
summary(fm)

Call:
lm(formula = Immatricolati ~ bs(tempo, df = 3), data = universita)

Residuals:
     Min       1Q   Median       3Q      Max
-2626.88  -943.97   -28.69   608.63  3679.72

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)          1253.0      554.2   2.261   0.0266 *
bs(tempo, df = 3)1  -6632.9     1609.9  -4.120 9.46e-05 ***
bs(tempo, df = 3)2  18514.2     1041.7  17.772  < 2e-16 ***
bs(tempo, df = 3)3  11007.3      868.3  12.676  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1305 on 77 degrees of freedom
Multiple R-Squared: 0.9432,     Adjusted R-squared: 0.941
F-statistic: 426.4 on 3 and 77 DF,  p-value: < 2.2e-16

plot(tempo, Immatricolati,  main="Immatricolati all'Univerità di Bari")
lines(tempo, fitted(fm))
```

Fig. 35



Confrontiamo il modello di regressione polinomiale con quello della splines regression:

```
anova(fm, fmpol)
Analysis of Variance Table

Model 1: Immatricolati ~ bs(tempo, df = 3)
Model 2: Immatricolati ~ poly(tempo, 5)
  Res.Df       RSS Df Sum of Sq      F   Pr(>F)
1     77 131173407
2     75 112455358  2  18718050 6.2418 0.003108 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

3.16 Stima simultanea di più modelli di regressione

Con `R` è possibile stimare contemporaneamente due o più modelli lineari quando si vogliono far regredire due o più variabili risposte su uno stesso insieme di regressori. Nel dataframe `regioni` sono riportate una serie di variabili socio-economiche e demografiche delle 20 regioni italiane. Tra queste la spesa media mensile per l'istruzione (`istruzione`) e quella per il tempo libero (`tempolibero`)

```
names(regioni)
 [1] "Regione"     "sanita"      "istruzione"  "tempolibero" "spesatot"
 [6] "tattivita"   "tattivitaF"  "toccup"      "tdisocup"    "lavastov"
[11] "lavatrice"   "automobile"  "PC"          "TV"          "telefono"
[16] "tosped"      "mort_tum"    "mort_card"   "IVG"         "FLE"
[21] "IV"          "istr_univ"   "VA"
```

vogliamo stimare contemporaneamente due modelli di regressione che abbiamo queste due variabili come risposta:

```
fm<-
lm(cbind(regioni$istruzione,regioni$tempolibero)~toccup+istr_univ+FLE,
data=regioni)

summary(fm)
Response regioni$istruzione :

Call:
lm(formula = `regioni$istruzione` ~ toccup + istr_univ + FLE,
    data = regioni)

Residuals:
    Min      1Q  Median      3Q     Max
-7.1800 -3.5158  0.6328  2.9751  7.7414

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -478.1727   165.5210  -2.889  0.01068 *
toccup        -0.5655     0.2157  -2.622  0.01850 *
istr_univ     -2.1194     1.0779  -1.966  0.06687 .
FLE            6.5649     2.0901   3.141  0.00631 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.565 on 16 degrees of freedom
Multiple R-Squared: 0.4578,     Adjusted R-squared: 0.3561
F-statistic: 4.503 on 3 and 16 DF,  p-value: 0.01793


Response regioni$tempolibero :

Call:
lm(formula = `regioni$tempolibero` ~ toccup + istr_univ + FLE,
    data = regioni)

Residuals:
     Min       1Q   Median       3Q      Max
-20.4188  -5.1226   0.9789   5.9792  11.4769

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.9637   330.2714  -0.033   0.9739
toccup        2.3232     0.4304   5.398 5.92e-05 ***
istr_univ     5.1192     2.1508   2.380   0.0301 *
FLE          -0.2580     4.1704  -0.062   0.9514
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.108 on 16 degrees of freedom
Multiple R-Squared: 0.8134,     Adjusted R-squared: 0.7784
F-statistic: 23.24 on 3 and 16 DF,  p-value: 4.487e-06

coef(fm)
                    [,1]        [,2]
```

```
(Intercept) -478.1727089 -10.9637384
toccup        -0.5654997   2.3231613
istr_univ     -2.1193913   5.1191543
FLE            6.5648718  -0.2580278
```

4.0 Multicollinerità, principal component regression (PCR) e ridge regression

Una delle ipotesi alla base del modello di regressione classico (OLS) è che la matrice delle variabili esplicative **X** abbia rango k+1. Se il rango di questa matrice è inferiore a k+1 si ha che |**X'X**|=0 e non si può calcolare l'inversa di **X'X**. Ne consegue che la stima dei coefficienti di regressione non può essere determinata univocamente. In questa circostanza si è in presenza di multicollinearità[21]. Si dice invece che si è in presenza di quasi multicollinearità se |**X'X**| è assai prossimo a zero. In questo caso gli elementi della diagonale della matrice $(\mathbf{X'X})^{-1}$ assumono valori molto elevati e, di conseguenza, le stime delle varianze dei coefficienti di regressione risultano poco attendibili.
Vi sono diversi strumenti per diagnosticare la presenza di multicollinearità tra i regressori:

1) R_j : coefficiente di correlazione parziale tra la variabile j.esima e le restanti k-1 covariate;
2) Tolerances: $tol_j = 1 - R_j^2$;
3) Variance Inflation Factor (VIF): $VIF_j = \frac{1}{1 - R_j^2}$

Se la j-esima variabile non presenta alcuna relazione lineare con le altre k-1 si ha che $R_j = 0$ e $VIF_j = 1$

In presenza di quasi multicollinearità VIF_j misura l'entità dell'aumento della varianza di $\hat{\beta}_j$ dovuto a tale problema. Se la j.esima covariata dipende linearmente dalle altre k-1 si ha che $R_j^2 = 1$ e $VIF_j = \infty$;

4) Matrice di correlazione tra le covariate (R); in caso di multicollinearità si ha un'elevata correlazione tra i regressori. Inoltre si dimostra che i VIF sono dati dagli elementi della diagonale principale di R^{-1};
5) *Condition number*: per ciascuna variabile esplicativa è la radice quadrata del rapporto tra il più grande degli autovalori di **X'X** e il j.esimo autovettore: $\sqrt{\frac{\lambda_{\max}}{\lambda_j}}$. Si ritiene una covariata per la quale si condition numer è maggiore di 30;
6) Si ha un elevato valore dell'indice di determinazione (e quindi una significatività della regressione nel complesso) ma valori non significativi del test t per i coefficienti di regressione presi singolarmente. Inoltre le correlazioni parziali tra i regressori sono basse.

Come esemplificazione prendiamo un caso da manuale, quello di Longley, ma con i dati riferiti all'Italia nel periodo 1988-2003 contenuti nel dataframe `longley_ita`:

```
longley_ita
   employed Armed.force   pop15 unemployed       GDP deflatore anno
1     21374         556 48208.0       2868  880012.0     64.14 1988
2     21391         558 48090.0       2867  905146.0     68.29 1989
3     21764         549 47387.0       2751  922344.5     73.96 1990
4     21946         536 47642.0       2653  932505.0     79.79 1991
5     21813         543 47915.0       2535  941492.3     83.25 1992
6     21381         399 47528.8       2298  936272.8     86.23 1993
7     20372         400 47371.0       2507  958278.6     89.11 1994
8     20233         381 48083.3       2637  983386.9     93.86 1995
9     20319         368 48278.6       2653  994721.4     98.77 1996
```

[21] Si veda A. POLLICE, *op. cit*, cap. 4, pagg. 65-67 e J. J. FARAWAY, *op. cit*., pagg. 117-120

```
10    20413         385 48482.3       2668 1013645.4    101.25 1997
11    20618         375 48653.1       2744 1031661.5    104.01 1998
12    20863         371 48758.8       2669 1045985.2    105.93 1999
13    21225         351 48917.0       2495 1082137.5    107.80 2000
14    21634         334 49084.0       2267 1103885.5    110.39 2001
15    21922         309 49203.0       2163 1110510.2    113.52 2002
16    22134         312 49208.0       2096 1114221.5    116.76 2003
```

Si vuole far regredire la variabile `employed` sulle rimanenti.

```
fm<-lm(employed~.+., data=longley_ita)
summary(fm)

Call:
lm(formula = employed ~ . + ., data = longley_ita)

Residuals:
    Min      1Q  Median      3Q     Max
-341.69 -155.15   26.76  137.76  278.21

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.705e+05  6.077e+05  -0.445  0.66679
Armed.force  1.077e+01  2.620e+00   4.110  0.00264 **
pop15        1.713e-01  2.925e-01   0.586  0.57236
unemployed  -2.458e+00  4.984e-01  -4.933  0.00081 ***
GDP          5.042e-03  9.265e-03   0.544  0.59951
deflatore   -3.897e+01  4.817e+01  -0.809  0.43933
anno         1.422e+02  3.131e+02   0.454  0.66041
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 249.7 on 9 degrees of freedom
Multiple R-Squared: 0.9123,     Adjusted R-squared: 0.8538
F-statistic:  15.6 on 6 and 9 DF,  p-value: 0.0002698
```

Il modello appare sospetto di multicollinearità, verifichiamolo con alcuni indicatori visti in precedenza:

```
library(car)
vif(fm)
Armed.force       pop15  unemployed         GDP   deflatore        anno
  14.266720    8.199176    3.349282  123.203277  149.948808  534.533470
```

Calcoliamo i *condition numbers*:

```
X<-as.matrix(longley_ita[,-1])
e<-eigen(t(X)%*%X) ## calcolo autovalori e autovettori
e$val ## autovalori
[1] 1.603937e+13 1.561108e+08 3.805496e+05 2.592723e+04 3.229973e+03
[6] 9.814119e+01
condnum<-sqrt(max(e$val)/e$val) ## condition number
condnum
[1] 1.0000    320.5363   6492.1425  24872.2787  70468.3741 404266.7512
```

Come si vede i *condition number* sono tutti superiori di molto a 30.
Calcoliamo la matrice di correlazione tra i regressori:

```
R<-cor(longley_ita[,-1])
round(R,3)
             Armed.force  pop15 unemployed    GDP deflatore   anno
Armed.force        1.000 -0.672      0.672 -0.879    -0.937 -0.927
pop15             -0.672  1.000     -0.447  0.863     0.776  0.828
unemployed         0.672 -0.447      1.000 -0.690    -0.676 -0.704
GDP               -0.879  0.863     -0.690  1.000     0.968  0.989
deflatore         -0.937  0.776     -0.676  0.968     1.000  0.991
anno              -0.927  0.828     -0.704  0.989     0.991  1.000
```

siamo in presenza di una forte correlazione tra le variabili `anno` e `GDP` e `deflatore`, sono queste le variabili che determinano la collinearità.

Per risolvere il problema della multicollinearità ci sono diverse strade che possono essere percorse:

1) l'aggiunta di nuove osservazioni che rendano la matrice **X** a rango pieno (anche se questo rimedio non è sempre applicabile);
2) l'esclusione dal modello delle variabili correlate ovvero di quelle per le quali la stima della varianza del coefficiente di regressione associato è elevata;
3) l'uso della *principal component regression* (PCR): si estraggono le componenti principali dai regressori originali (queste nuove variabili sono per definizione tra loro ortogonali) e si fa regredire la variabile risposta su queste;
4) l'uso della *ridge regression*.

Non disponendo di ulteriori osservazioni, applichiamo la seconda soluzione ed eliminiamo dal modello le variabili `GDP, delatore` e `pop15` che sono molto correlate con `anno`:

```
fm1<-lm(employed~ Armed.force+unemployed+anno, data=longley_ita)
summary(fm1)

Call:
lm(formula = employed ~ Armed.force + unemployed + anno, data =
longley_ita)

Residuals:
    Min      1Q  Median      3Q     Max
-554.99  -52.51   77.56  135.30  291.21

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.976e+05  8.226e+04  -3.618  0.00353 **
Armed.force  1.361e+01  1.999e+00   6.808 1.88e-05 ***
unemployed  -2.468e+00  4.143e-01  -5.957 6.64e-05 ***
anno         1.601e+02  4.070e+01   3.932  0.00199 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 269.1 on 12 degrees of freedom
Multiple R-Squared: 0.8642,     Adjusted R-squared: 0.8303
F-statistic: 25.46 on 3 and 12 DF,  p-value: 1.726e-05
```

Si evince facilmente che nel nuovo modello stimato non vi è più multicollinearità e tutti i coefficienti di regressione risultano significativi; inoltre i VIF delle variabili esplicative si sono ridimensionati di molto:

```
vif(fm1)
Armed.force  unemployed        anno
```

```
   7.152678    1.993773    7.781343
```

Volendo percorre la strada della PCR estraiamo le componenti principali (CP)[22] delle variabili esplicative. Si ricorda che CP sono delle combinazioni lineari dei predittori, tra loro sono ortogonali e sono correlate con le variabili originarie. Ciascuna CP spiega una quota della varianza delle variabili originarie. Per comodità calcoliamo le CP standardizzando i regressori e operiamo sulla matrice di correlazione:

```
cp<-princomp(longley_ita[,-1], cor=T)
summary(cp)
Importance of components:
                          Comp.1     Comp.2     Comp.3     Comp.4      Comp.5
Standard deviation     2.2477187 0.75933047 0.53190082 0.27301419 0.111174650
Proportion of Variance 0.8420399 0.09609713 0.04715308 0.01242279 0.002059967
Cumulative Proportion  0.8420399 0.93813701 0.98529009 0.99771288 0.999772844
                           Comp.6
Standard deviation     0.036917962
Proportion of Variance 0.000227156
Cumulative Proportion  1.000000000
```

La prima CP estratta spiega da sola più dell'84% della variabilità dei regressori, mentre le prime due quasi il 94%. Vediamo ora l'identificazione delle CP esaminando le correlazioni con le variabili originali:

```
cp$scores ## componenti principali
       Comp.1      Comp.2      Comp.3      Comp.4       Comp.5       Comp.6
1   3.3526574  0.80362788 -0.71318913 -0.55741707 -0.001670696  0.025861822
2   3.0751177  0.75492040 -0.54116452 -0.17098514 -0.088614339  0.008310234
3   2.9409083 -0.26513679  0.05737036  0.45365480 -0.187976148  0.010989386
4   2.2670978 -0.34033074 -0.18947722  0.39560268  0.039364031 -0.039485806
5   1.7172047 -0.45447569 -0.66492003  0.40355785  0.232827990 -0.013981832
6   0.7972521 -1.79099075  0.21133090 -0.42694498  0.075662100 -0.002352473
7   0.8985713 -1.15133181  0.84519967  0.02405744 -0.114776382  0.058664637
8   0.1925238  0.01446301  0.63937375 -0.21450305 -0.030219259 -0.040946632
9  -0.2603859  0.28193363  0.66932784 -0.19755442  0.094837191 -0.052185718
10 -0.5595025  0.59280389  0.44620659  0.02328945  0.113422667 -0.012866893
11 -0.8763394  1.05610198  0.55987852  0.09340203  0.066698667  0.012726199
12 -1.3011597  0.92861566  0.37129049  0.14246587  0.062364279  0.063658253
13 -2.1024357  0.52143818  0.00637250  0.11820524 -0.136761821 -0.013211209
14 -2.9093194 -0.08462291 -0.47168357  0.02066286 -0.144537598 -0.050174003
15 -3.4692982 -0.33348219 -0.57004895 -0.12387303 -0.057455875 -0.017892087
16 -3.7628923 -0.53353376 -0.65586721  0.01637947  0.076835192  0.062886122

cp$loadings ## cp loadings

Loadings:
            Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
Armed.force  0.414        -0.574  0.669  0.189
pop15       -0.373  0.572 -0.586 -0.398  0.171
unemployed   0.334  0.802  0.478  0.102
GDP         -0.438  0.120         0.429 -0.703 -0.332
deflatore   -0.436         0.290  0.370  0.659 -0.392
anno        -0.443         0.102  0.250         0.853

               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
SS loadings     1.000  1.000  1.000  1.000  1.000  1.000
Proportion Var  0.167  0.167  0.167  0.167  0.167  0.167
Cumulative Var  0.167  0.333  0.500  0.667  0.833  1.000
```

[22] Si veda F. DEL VECCHIO, *Analisi op. cit.*, pagg. 325-342 e A.POLLICE, *op. cit,* cap. 7, pagg. 103-112

```
round(cor(longley_ita[,-1], cp$scores),3)

            Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6
Armed.force  0.931  0.074 -0.305  0.183  0.021  0.003
pop15       -0.838  0.435 -0.312 -0.109  0.019 -0.002
unemployed   0.751  0.609  0.254  0.028 -0.008  0.001
GDP         -0.985  0.091 -0.038  0.117 -0.078 -0.012
deflatore   -0.980  0.029  0.154  0.101  0.073 -0.014
anno        -0.995  0.044  0.054  0.068  0.003  0.031
```

la prima CP è correlata in modo abbastanza forte con tutti i regressori, con alcuni positivamente (`Armed.force` e `unemployed`) e con la maggior parte negativamente; la seconda CP è correlata positivamente con `pop15` e `unemployed`.

Stimiamo la regressione multipla della variabile risposta sulle CP:

```
fmcp<-lm(longley_ita$employed~cp$scores)
summary(fmcp)

Call:
lm(formula = longley_ita$employed ~ cp$scores)

Residuals:
    Min      1Q  Median      3Q     Max
-341.69 -155.15   26.76  137.76  278.21

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     21212.62      62.42 339.821  < 2e-16 ***
cp$scoresComp.1   -11.49      27.77  -0.414   0.6887
cp$scoresComp.2  -237.19      82.21  -2.885   0.0180 *
cp$scoresComp.3 -1026.01     117.36  -8.743 1.08e-05 ***
cp$scoresComp.4   645.71     228.64   2.824   0.0199 *
cp$scoresComp.5  -413.19     561.49  -0.736   0.4805
cp$scoresComp.6   724.88    1690.86   0.429   0.6782
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 249.7 on 9 degrees of freedom
Multiple R-Squared: 0.9123,     Adjusted R-squared: 0.8538
F-statistic:  15.6 on 6 and 9 DF,  p-value: 0.0002698
```

Da cui emerge un legame statisticamente significativo tra la risposta `employed` e le CP 2, 3 e 4. Stimiamo il modello solo con questi regressori:

```
fmcp2<-lm(longley_ita$employed~cp$scores[,2]+cp$scores[,3]+cp$scores[,4])
summary(fmcp2)

Call:
lm(formula = longley_ita$employed ~ cp$scores[, 2] + cp$scores[,3] +
cp$scores[, 4])

Residuals:
    Min      1Q  Median      3Q     Max
-450.22 -117.41   26.91  162.19  254.42
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    21212.62      56.69 374.196  < 2e-16 ***
cp$scores[, 2]  -237.19      74.66  -3.177  0.00796 **
cp$scores[, 3] -1026.01     106.58  -9.627 5.39e-07 ***
cp$scores[, 4]   645.71     207.64   3.110  0.00902 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 226.8 on 12 degrees of freedom
Multiple R-Squared: 0.9036,     Adjusted R-squared: 0.8795
F-statistic: 37.48 on 3 and 12 DF,  p-value: 2.258e-06
```

Si può usare anche il comando `pcr()` del package `pls`[23]:

```
fm.pcr<-pcr(employed~Armed.force+pop15+unemployed+GDP+deflatore+anno,
ncomp=2,method = "svdpc", data=longley_ita, model=T)

fm.pcr
Principal components regression, fitted with the singular value
decomposition algorithm.
Call:
pcr(employed ~ Armed.force + pop15 + unemployed + GDP + deflatore +
anno, ncomp = 2, data = longley_ita, model = T, method = "svdpc")

summary(fm.pcr)
Data:   X dimension: 16 6
        Y dimension: 16 1
Fit method: svdpc
Number of components considered: 2
TRAINING: % variance explained
           1 comps  2 comps
X          99.9978  100.000
employed    0.7203    1.242

fm.pcr$coefficients
, , 1 comps

                 employed
Armed.force -7.593543e-07
pop15        5.058617e-06
unemployed  -1.516872e-06
GDP          7.176058e-04
deflatore    1.473901e-07
anno         4.375191e-08

, , 2 comps

                 employed
Armed.force  0.0070690553
pop15        0.1377309522
unemployed   0.0387421950
GDP         -0.0001637172
deflatore   -0.0007983749
anno        -0.0001060068
```

[23] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/pls.html

L'uso della *ridge regression*[24] consente di ottenere delle stime stabili dei coefficienti di regressione in presenza di multicollinearità con la matrice **X'X** assai prossima alla singolarità. Lo stimatore di tipo ridge è definito in questo modo:

$$\hat{\beta}_{(\lambda)} = ((X'X) + \lambda I_{k+1})^{-1} X'y$$

dove λ è una costante non negativa detta *shrinkage parameter* e di solito compresa tra 0 e 1 ($\lambda = 0$ corrisponde alle stime OLS). La scelta di questa costante viene effettuata in base all'intensità della multicollinearità esistente, cercando di garantire un opportuno bilanciamento tra la varianza e la distorsione dello stimatore. Un metodo esplorativo consiste nella costruzione di un grafico che rappresenti gli elementi del vettore $\hat{\beta}_{(\lambda)}$ (sull'asse delle ordinate) in funzione di λ. Si ritiene che le curve di tale grafico, detto traccia della *regressione ridge*, tendano a stabilizzarsi in corrispondenza di valori accettabili di λ. Per stimare i parametri della *ridge regression* applicata ai nostri dati con `R` usiamo il comando `lm.ridge()` presente nel package `MASS`:

```
library(MASS)
fmr<-lm.ridge(employed~ .+., data=longley_ita, lambda = seq(0,1,0.001))
matplot(fmr$lambda,t(fmr$coef),  type="l",  main="Ridge  trace  plot",
xlab=expression(lambda), ylab=expression(hat(beta))) ## ridge trace plot
abline(h=0,lwd=2)
```

Abbiamo tracciato il ridge trace plot (Fig. 35), vengono forniti diversi valori per il parametro λ:

```
select(fmr)
modified HKB estimator is 0.1122239
modified L-W estimator is 0.6835546
smallest value of GCV  at 0.234
```

prendiamo come stima di λ = 0.68, un valore vicino alla seconda stima che appare più convincente. Le stime dei coefficienti della *ridge regression* risultano quindi essere:

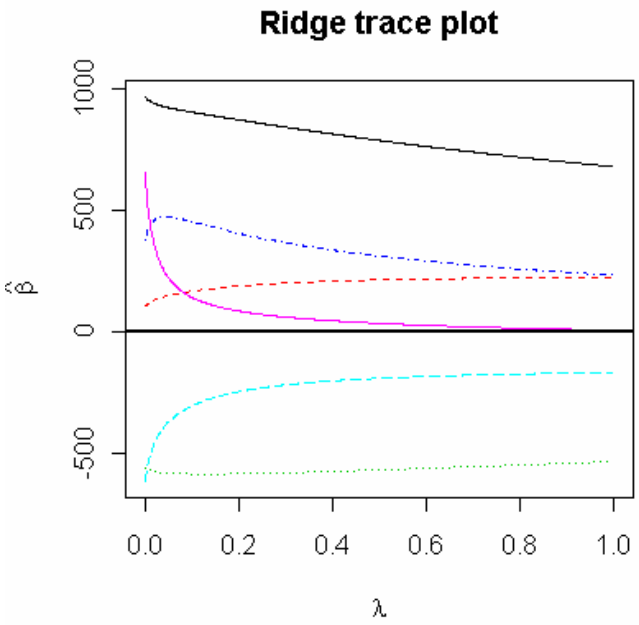
```
fmr$coef[,fmr$lam == 0.68]
Armed.force       pop15  unemployed         GDP   deflatore        anno
  744.85742   219.36956  -557.42308   276.58225  -179.74611    20.81831
```

che paragonate alle stime OLS:

```
fmr$coef[,fmr$lam == 0]
Armed.force       pop15  unemployed         GDP   deflatore        anno
   969.1546    104.7189   -563.4934    377.0803   -618.4613    655.5658
```

[24] Si veda A. POLLICE, *op. cit*, cap. 4, pagg. 67-69 e J. J. FARAWAY, *op. cit*., pagg. 120-123

Fig. 35



5.0 Autocorrelazione dei residui e stime GLS

Una delle ipotesi di base del modello lineare classico è che la matrice della varianze e covarianze degli errori sia scalare:

$$E(\varepsilon\varepsilon') = \sigma^2 I_n$$

In questo paragrafo esamineremo il caso in cui questa ipotesi venga meno e la matrice di varianze e covarianze assume la seguente espressione:

$$E(\varepsilon\varepsilon') = \sigma^2 \Omega$$

con Ω matrice quadrata simmetrica, definita positiva e con traccia pari a n e diversa da I_n e σ^2 un parametro incognito. In tale circostanza la stime OLS dei coefficienti di regressione, pur continuando ad essere corrette, non sono le più efficienti. Occorre usare i minimi quadrati generalizzati[25] (GLS, Generalized least squares).
Si consideri la decomposizione di Cholesky della matrice simmetrica, definita positiva Ω^{-1}:

$$\Omega^{-1} = A'A$$

premoltiplicando per A la forma standard del modello lineare generale si ottiene:

$$Ay = AX\beta + A\varepsilon$$

si dimostra che questo nuovo modello soddisfa le ipotesi di base del modello lineare classico in quanto:

$$E[(A\varepsilon)(A\varepsilon)'] = \sigma^2 I_n$$

applicando il metodo dei minimi quadrati ordinari al modello trasformato abbiamo stimatore GLS di β, noto anche come stimatore di Aitken:

$$\hat{\beta}_{GLS} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}y$$

[25] Si veda F. DEL VECCHIO, *Analisi op. cit.*, pag. 200 e segg. e A. POLLICE, *op. cit,* cap. 4, pag. 61 e segg.

la matrice di varianze e covarianze di questo stimatore è data da:

$$V(\hat{\beta}_{GLS}) = \sigma^2 (X'\Omega^{-1}X)^{-1}$$

mentre una stima di σ^2 si ottiene con:

$$\hat{\sigma}^2{}_{GLS} = \frac{(Ay - AX\hat{\beta}_{GLS})'(Ay - AX\hat{\beta}_{GLS})}{n-k-1}$$

Un caso in cui possono trovare applicazione i minimi quadrati generalizzati è quello della presenza di errori autocorrelati. Nel modello classico si suppone che $E(\varepsilon_i \varepsilon_j) = 0$ per ogni $i \neq j$. Quando ciò non avviene si dice che gli errori sono autocorrelati o che si è in presenza di correlazione seriale. Diverse possono essere le cause dell'autocorrelazione dei gli errori, tra tutte ricordiamo il caso di dati storici nel quale le variabili esplicative possono presentare una correlazione con il tempo.
Un caso assai frequente è il processo autoregressivo del 1° ordine:

$$\varepsilon_t = \rho \varepsilon_{t-1} + \eta_t$$

dove η_t è distribuita normalmente con media zero, varianza costante ed in correlata. Per verificare la presenza di tale processo si ricorre al test di Durbin-Watson:

$$d = \frac{\sum_{t=2}^{n} (e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2}$$

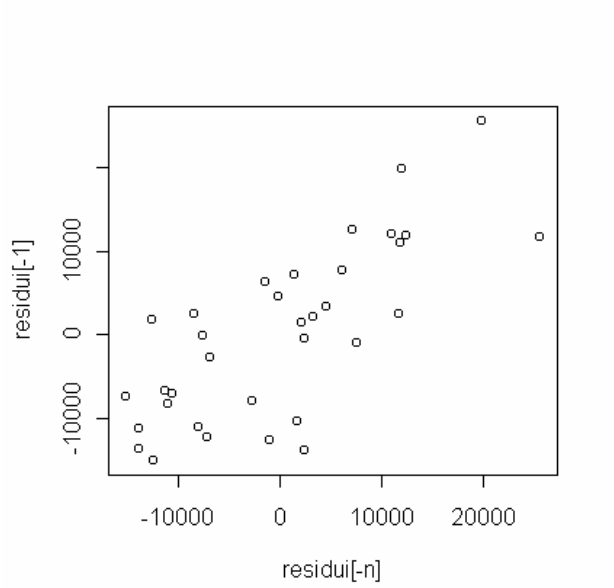
dove e_t sono i residui della regressione. Una volta accertata l'esistenza di un tale processo si può facilmente determinare la matrice Ω e procedere con la stima GLS.
Come applicazione prendiamo i dati del dataframe `contnaz`, che contiene i dati del PIL e dei consumi nazionali in Italia a prezzi costanti dal 1970 al 2004, e stimiamo la seguente regressione con il metodo OLS:

```
fm.ols<-lm( consumi ~ PIL, data = contnaz)
summary(fm.ols, cor=T)

Call:
lm(formula = consumi ~ PIL, data = contnaz)

Residuals:
     Min       1Q   Median       3Q      Max
-15166.1  -8139.4   -175.8   6699.4  25672.8

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4.032e+04  8.173e+03  -4.934 2.24e-05 ***
PIL          8.246e-01  1.014e-02  81.334  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10360 on 33 degrees of freedom
Multiple R-Squared: 0.995,      Adjusted R-squared: 0.9949
F-statistic:  6615 on 1 and 33 DF,  p-value: < 2.2e-16

Correlation of Coefficients:
    (Intercept)
PIL -0.98
```

La correlazione tra la stima del coefficiente di regressione e l'intercetta è decisamente elevata. Tracciamo il grafico dei residui vs residui ritardati (Fig. 36):

```
residui<-residuals(fm.ols)
n<-length(residui)
plot(residui[-n], residui[-1])
```

Fig. 36



Appare evidente una relazione di tipo lineare tra gli errori. Verifichiamo la presenza di autocorrelazione degli errori con il test di Durbin-Watson:

```
library(lmtest)
dwtest(formula(fm.ols), data=contnaz)

        Durbin-Watson test

data:  formula(fm.ols)
DW = 0.4876, p-value = 4.876e-09
      alternative hypothesis: true autocorrelation is greater than 0
```

il test è molto significativo, siamo in presenza di correlazione seriale. Per stimare i coefficienti di regressione occorre usare le stime GLS tramite il comando `gls()` presente nel package `nlme` oppure `lm.gls()` nel package `MASS`:

```
library(nlme)
fm.gls<-gls(consumi ~ PIL, data = contnaz, correlation=corAR1(form=
~anno))

fm.gls
Generalized least squares fit by REML
  Model: consumi ~ PIL
  Data: contnaz
  Log-restricted-likelihood: -350.527

Coefficients:
 (Intercept)          PIL
3.186552e+04 7.295035e-01
```

```
Correlation Structure: AR(1)
 Formula: ~anno
 Parameter estimate(s):
     Phi
0.999997
Degrees of freedom: 35 total; 33 residual
Residual standard error: 2854725

summary(fm.gls)
Generalized least squares fit by REML
  Model: consumi ~ PIL
  Data: contnaz
  Log-restricted-likelihood: -350.527

Coefficients:
 (Intercept)          PIL
3.186552e+04 7.295035e-01

Correlation Structure: AR(1)
 Formula: ~anno
 Parameter estimate(s):
     Phi
0.999997
Degrees of freedom: 35 total; 33 residual
Residual standard error: 2854725

summary(fm.gls)
Generalized least squares fit by REML
  Model: consumi ~ PIL
  Data: contnaz
      AIC    BIC   logLik
  709.054 715.04 -350.527

Correlation Structure: AR(1)
 Formula: ~anno
 Parameter estimate(s):
     Phi
0.999997

Coefficients:
               Value Std.Error   t-value p-value
(Intercept) 31865.52 2855020.9  0.011161  0.9912
PIL             0.73       0.1 12.227330  0.0000

 Correlation:
    (Intr)
PIL -0.016

Standardized residuals:
          Min            Q1           Med            Q3           Max
-0.0106110698 -0.0057983636  0.0009704235  0.0063673762  0.0131981522

Residual standard error: 2854725
Degrees of freedom: 35 total; 33 residual

intervals(fm.gls)
```

```
Approximate 95% confidence intervals

 Coefficients:
                     lower         est.        upper
(Intercept) -5.776718e+06 3.186552e+04 5.840449e+06
PIL          6.081208e-01 7.295035e-01 8.508862e-01
attr(,"label")
[1] "Coefficients:"

 Correlation structure:
    lower     est. upper
Phi    -1 0.999997     1
attr(,"label")
[1] "Correlation structure:"

 Residual standard error:
       lower         est.        upper
1.236228e-04 2.854725e+06 6.592191e+16
```

Con il comando `gls()`occorre specificare gli argomenti `correlation` per indicare il tipo di autocorrelazione (si veda ?`corClasses` per maggiori informazioni, qui basta ricordare le strutture `corAR1` per processi autoregressivi del primo ordine e `corARMA` per processi autoregressivi a media mobile (ARMA)), `weights` per indicare il sistema di ponderazione nel caso di errori eteroschedastici (si veda `?varClasses` per maggiori informazioni) e `method` per scegliere il metodo per la stima dei parametri che può essere: "`REML`" (massimizza la log-verosimiglianza ristretta) oppure "`ML`" (massimizza la verosimiglianza). Il metodo di default è "`REML`".
Il comando `lm.gls()`, invece, prevede che venga specificata la matrice dei pesi degli errori nell'argomento `W`; se viene posto `inverse=TRUE` la matrice `W` contiene le varianze e covarianze degli errori. Questo comando può essere usato in modo più generico di `gls()`per qualsiasi tipo di matrice di varianze e covarianze degli errori, purché sia nota. È possibile calcolare la funzione di autocorrelazione dei residui stimati con il metodo GLS usando il comando `ACF()`:

```
 ACF(fm.gls, form=~anno)
   lag          ACF
1    0  1.000000000
2    1  0.908873557
3    2  0.808208975
4    3  0.727748402
5    4  0.664204726
6    5  0.578488797
7    6  0.495674756
8    7  0.386701498
9    8  0.303154273
10   9  0.248020640
11  10  0.197988306
12  11  0.122564374
13  12  0.002822141
14  13 -0.171615405
15  14 -0.301998462
16  15 -0.384117356
```

verifichiamo la normalità mediante il QQ-plot (Fig. 37):

```
qqnorm(fm.gls)
```

e tracciare il grafico residui vs fitted values (Fig. 38):

```
plot(fm.gls, main="Residui GLS vs fitted values GLS")
```

Fig. 37
QQ-plot dei residui stimati con il metodo GLS

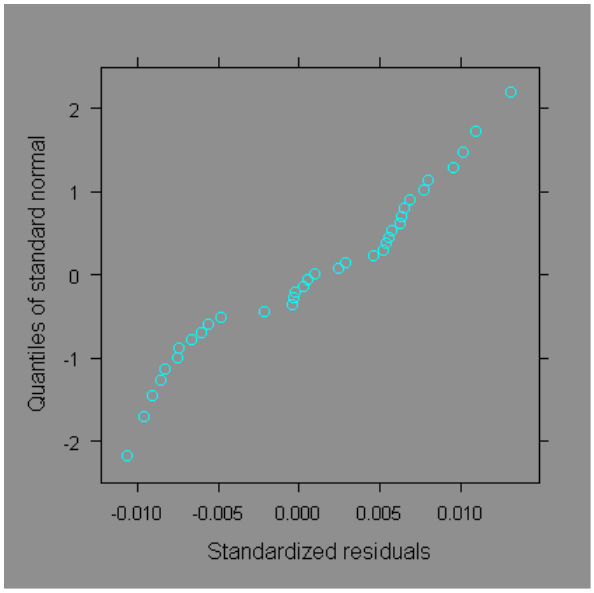
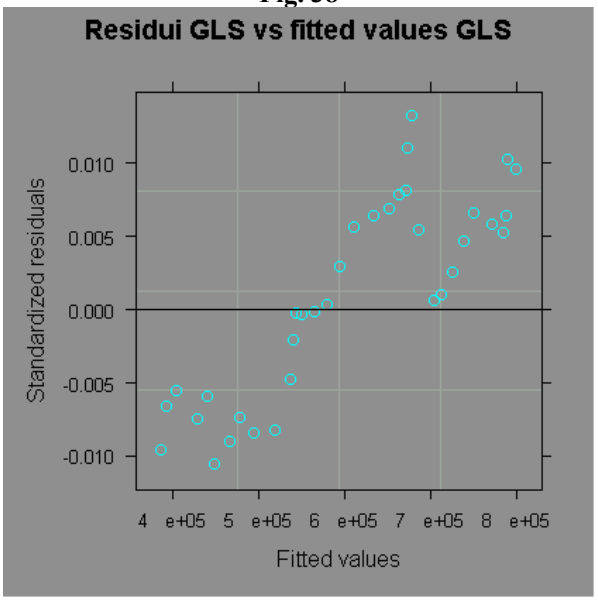


Fig. 38



Confrontiamo le stime dei coefficienti ottenute con il metodo OLS e il metodo GLS:

```
coef(fm.ols)## stime con OLS
  (Intercept)           PIL
-4.032337e+04  8.246065e-01

coef(fm.gls)## stime con GLS
  (Intercept)           PIL
3.186552e+04 7.295035e-01
```

Con il comando `methods()` possiamo vedere, specificando il tipo di classe, quali metodi si possono applicare agli oggetti di classe `gls`:

```
methods(class="gls")
 [1] ACF.gls*              anova.gls*            augPred.gls*
 [4] BIC.gls*              coef.gls*             comparePred.gls*
 [7] fitted.gls*           formula.gls*          getData.gls*
[10] getGroups.gls*        getGroupsFormula.gls* getResponse.gls*
[13] getVarCov.gls*        intervals.gls*        logLik.gls*
[16] plot.gls*             predict.gls*          print.gls*
[19] qqnorm.gls*           residuals.gls*        summary.gls*
[22] update.gls*           Variogram.gls*        vcov.gls*

Non-visible functions are asterisked
```

6.0 Eteroschedasticità e stime WLS

Un altro tipo di allontanamento dalle ipotesi di base del modello di regressione classico è la eteroschedasticità, ossia la non costanza delle vazianze dell'errore. Per tanto la matrice di varianze e covarianze, ipotizzando assenza di autocrrelazione, assume la seguente forma[26]:

$$\sigma^2\Omega = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \dots & & & \\ 0 & = & \dots & \sigma_n^2 \end{bmatrix}$$

in genere le varianze σ_i^2 sono incognite. Quando sono stimate la precedente relazione è valida a meno di una costante:

$$\hat{\sigma}^2\hat{\Omega} = k\begin{bmatrix} \hat{\sigma}_1^2 & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2^2 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & \hat{\sigma}_n^2 \end{bmatrix}$$

essendo $tr(\hat{\sigma}^2\hat{\Omega}) = n\hat{\sigma}^2 = k\sum_{i=1}^{n}\hat{\sigma}_i^2$ si ha $k / \hat{\sigma}^2 = n / \sum_{i=1}^{n}\hat{\sigma}_i^2$ da cui si ricava:

$$\hat{\Omega} = \frac{n}{\sum_{i=1}^{n}\hat{\sigma}_i^2}\begin{bmatrix} \hat{\sigma}_1^2 & 0 & \dots & 0 \\ 0 & \hat{\sigma}_2^2 & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & \hat{\sigma}_n^2 \end{bmatrix}$$

che può essere usata per le stime GLS. Tutto si riduce a stimare le varianze σ_i^2 o ipotizzando relazioni di tipo funzionale con i regressori oppure, quando possibile, tramite campioni.
Si può ipotizzare, nel caso di un solo, repressore che la varianza dell'errore sia funzione del regressore, ad esempio: $\sigma_i^2 = cx_i$ oppure $\sigma_i^2 = cx_i^2$ che sono delle relazioni abbastanza comuni. Basti pensare che analizzando il consumo delle famiglie in funzione del reddito è sensato assumere che la varianza del consumo dato il reddito sia anch'essa funzione del reddito. Un altro caso che può presentarsi è quello in cui le osservazioni sono medie basate su un numero diverso di repliche, in questa situazione i pesi sono noti e

[26] Si veda F. DEL VECCHIO, *Analisi op. cit*., pagg. 209-211

sono uguali al reciproco del numero delle repliche[27]. Il problema della eteroschedasticità viene risolto con l'impiego dei minimi quadrati ponderati (WLS, weighted least squares). Sia W = diag(w_1, . . . , w_n) la matrice diagonale dei pesi, lo stimatore dei minimi quadrati ponderati assume la seguente espressione:

$$\hat{\beta}_{WLS} = (X'WX)^{-1} X'Wy$$

In `R` è sufficiente specificare l'argomento `weights` nel comando `lm` per ottenere le stime WLS. Prendiamo questo esempio contenuto nel dataframe `education`: contiene i dati della spesa procapite per l'istruzione e il reddito procapite negli stati degli USA. Si usa un modello di regressione parabolica:

```
fm1<-lm(formula = Exp ~ Income + I(Income^2), data = education)
summary(fm1)

Call:
lm(formula = Exp ~ Income + I(Income^2), data = education)

Residuals:
     Min       1Q   Median       3Q      Max
-160.709  -36.896   -4.551   37.290  109.729

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.329e+02  3.273e+02   2.545  0.01428 *
Income      -1.834e-01  8.290e-02  -2.213  0.03182 *
I(Income^2)  1.587e-05  5.191e-06   3.057  0.00368 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 56.68 on 47 degrees of freedom
Multiple R-Squared: 0.6553,     Adjusted R-squared: 0.6407
F-statistic: 44.68 on 2 and 47 DF,  p-value: 1.345e-11
```

Tracciamo alcuni grafici relativi ai residui per verificare la presenza di eteroschedasticità negli errori (Fig. 39):

```
par(mfrow=c(2,2))
plot(fm1$res, main="Residui")
plot(fitted(fm1),fm1$res, main="Residui vs fitted values")
plot(fitted(fm1),abs(fm1$res), main="Residui in valore ass. vs fitted
values")
plot(fitted(fm1),(fm1$res)^2, main="Residui al quad. vs fitted values")
```

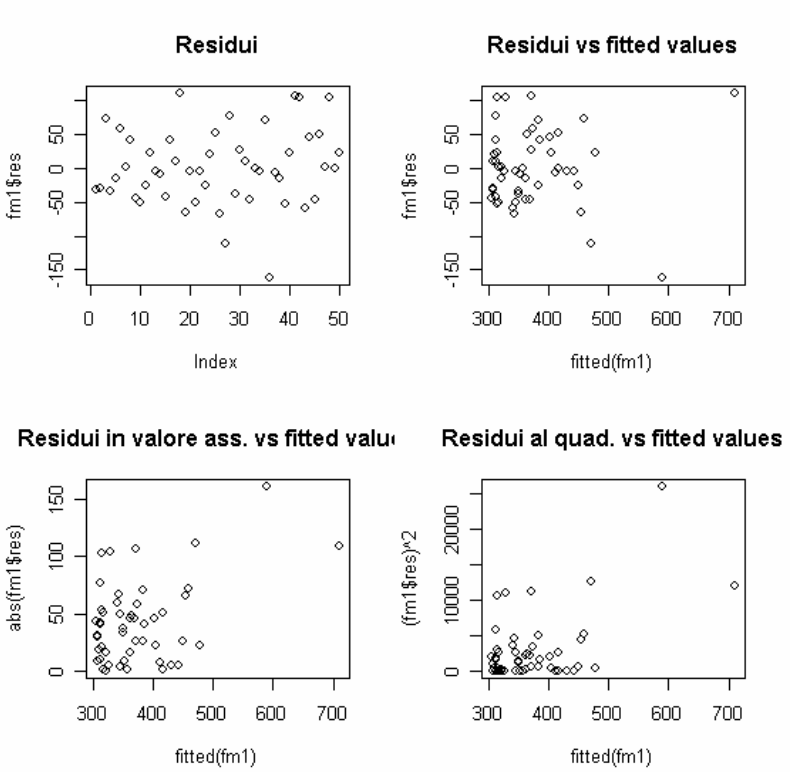
eseguiamo il testi di Breusch-Pagan ricorrendo ai comandi `bptest()` del package `lmtest` e `ncv.test()` del package `car`:

```
library(lmtest)
bptest(fm1)

        studentized Breusch-Pagan test

data:  fm1
BP = 15.8338, df = 2, p-value = 0.0003645

library(car)
```

[27] Cfr. G. M. MARCHETTI, *op. cit.*, pagg. 42-44

```
ncv.test(fm1)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 15.99945    Df = 1     p = 6.336086e-05
```

Fig. 39



Entrambi sono significativi, questo vuol dire che siamo in presenza di errori con varianza non costante. Stimiamo allora i coefficenti di regressione usando le stime WLS; valuteremo due versione usando come pesi prima il reciproco della variabile `Income` e poi il recipro del quadrato di questa:

```
fm2<-lm(formula = Exp ~ Income + I(Income^2), data = education, weights =
w)
summary(fm2)

Call:
lm(formula = Exp ~ Income + I(Income^2), data = education, weights = w)

Residuals:
     Min       1Q   Median       3Q      Max
-1.56222 -0.44232 -0.06254  0.40126  1.27254

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.464e+02  3.282e+02   2.274  0.02757 *
Income      -1.612e-01  8.448e-02  -1.908  0.06246 .
I(Income^2)  1.448e-05  5.377e-06   2.692  0.00981 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.6313 on 47 degrees of freedom
Multiple R-Squared: 0.6274,     Adjusted R-squared: 0.6115
```

```
F-statistic: 39.57 on 2 and 47 DF,  p-value: 8.412e-11

ncv.test(fm2)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 8.045736    Df = 1     p = 0.004561089
```

Come si può vedere il primo tipo di pesi non va ancora bene, la eteroschedasticità persiste. Vediamo cosa accade con il secondo tipo di pesi:

```
w<-1/(education$Income)^2
fm3<-lm(formula = Exp ~ Income + I(Income^2), data = education, weights =
w)
summary(fm3)

Call:
lm(formula = Exp ~ Income + I(Income^2), data = education, weights = w)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0151015 -0.0052971 -0.0007853  0.0044399  0.0157025

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.646e+02  3.336e+02   1.992   0.0522 .
Income      -1.399e-01  8.721e-02  -1.605   0.1153
I(Income^2)  1.311e-05  5.637e-06   2.326   0.0244 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.007121 on 47 degrees of freedom
Multiple R-Squared: 0.5983,     Adjusted R-squared: 0.5812
F-statistic:    35 on 2 and 47 DF,  p-value: 4.925e-10

ncv.test(fm3)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 3.022158    Df = 1     p = 0.08213411
```

L'etereschedasticità è stata finalmente eliminata, era da aspettarsi che la varianza dell'errore fosse proporzionale al quadrato di `Income`. Si veda anche il comando `hccm()`, sempre nel package `car`, che permette di ottenere la matrice di varianze e covarianze dei coefficienti corrette nel caso di un modello non pesato:

```
Var(fm1) ## varianze e covarianze del modello non pesato
              (Intercept)        Income   I(Income^2)
(Intercept)  1.071204e+05 -2.702115e+01  1.670894e-03
Income      -2.702115e+01  6.872169e-03 -4.284436e-07
I(Income^2)  1.670894e-03 -4.284436e-07  2.694407e-11

hccm(fm1) ## varianze e covarianze corrette
              (Intercept)        Income   I(Income^2)
(Intercept)  1.199026e+06 -3.256564e+02  2.180884e-02
Income      -3.256564e+02  8.853073e-02 -5.934046e-06
I(Income^2)  2.180884e-02 -5.934046e-06  3.980990e-10
```

Un ulteriore metodo per eliminare la eteroschedasticità, quando è possibile applicarlo, consiste nell'usare delle opportune trasformazioni della variabile risposta che stabilizzano la varianza[28].
Quando la relazione tra la varianza degli errori non è nota si può ricorrere a procedure di tipo iterativo, come gli Iteratively Reweighted Least Squares (IRLS). Ad esempio, se la relazione è del tipo:

$$\text{var}(\varepsilon_i) = \gamma_0 + \gamma_1 x_i$$

si può procedere in questo modo:

1) si inizia con pesi $w_i=1$
2) si stimano i coeffienti di regressione con il metodo OLS;
3) si usano i residui di questo regressione per stimare γ_0, γ_1 regredendo i quadrati dei residui su X
4) si ricalcono i nuovi pesi e si torna al passo 2

Si procede per iterazioni sino a convergenza del metodo.

Un approccio alternativo consiste nel modellare la relazione tra varianza e regressori e stimare congiuntamente i coeffienti di regressione e i pesi usando il metodo della massima verosimiglianza con il comando `gls()` del package `nlme`. Occorre specificando l'argomento `weights`. Si veda l'help: `?varClasses` per approfondimenti.

```
fmgls<-gls(Exp~Income+I(Income^2),weights=varPower(0.5,~Income),
data=education,na.action=na.omit)

summary(fmgls)
Generalized least squares fit by REML
  Model: Exp ~ Income + I(Income^2)
  Data: education
       AIC      BIC    logLik
  570.5405 579.7912 -280.2703

Variance function:
 Structure: Power of variance covariate
 Formula: ~Income
 Parameter estimates:
   power
1.694810

Coefficients:
               Value Std.Error   t-value p-value
(Intercept) 553.3851  347.1736  1.593972  0.1176
Income       -0.1105    0.0926 -1.193326  0.2387
I(Income^2)   0.0000    0.0000  1.834102  0.0730

 Correlation:
            (Intr) Income
Income      -0.997
I(Income^2)  0.986 -0.996

Standardized residuals:
       Min         Q1        Med         Q3        Max
-1.6580458 -0.7914569 -0.1138791  0.6104651  2.4395231
```

[28] Si veda F. DEL VECCHIO, *Statistica op. cit.*, pagg. 366-369

```
Residual standard error: 1.427063e-05
Degrees of freedom: 50 total; 47 residual
```

7.0 Structural Equation Models (SEM)

In questo paragrafo esamineremo alcuni modelli di regressione che trovano particolare applicazione in campo econometrico.
I modelli SEM[29], modelli di equazioni strutturali o ad equazioni simultanee, sono dei modelli di regressione multi-equazione nei quali le variabili risposta di un'equazione del SEM possono comparire come regressori in un'altra equazione, ovvero le variabili di un SEM si influenzano a vicenda tra loro. Le equazioni strutturali rappresentano delle relazioni tra le variabili di di un modello economico. Un tipo esempio di SEM è il modello macro-econonomico di Klein:

$$C_t = \gamma_{10} + \gamma_{11} P_t + \gamma_{12} P_{t-1} + \beta_{11}(W_t^p + W_t^g) + \zeta_{1t}$$
$$I_t = \gamma_{20} + \gamma_{21} P_t + \gamma_{22} P_{t-1} + \beta_{21} K_{t-1} + \zeta_{2t}$$
$$W_t^p = \gamma_{30} + \gamma_{31} A_t + \beta_{31} X_t + \beta_{32} X_{t-1} + \zeta_{3t}$$
$$X_t = C_t + I_t + G_t$$
$$P_t = X_t - T_t - W_t^p$$
$$K_t = K_{t-1} + I_t$$

Le variabili sul lato sinistro delle equazioni strutturali sono le variabili endogene, i cui valori sono determinati dal modello. In genere ad ogni variabile endogena corrisponde un'equazione. Le variabili ζ sono gli errori ζ che di solito non sono indipendenti tra loro. Le variabili nel lato destro sono le variabili esogene che sono fisse e indipendenti dagli errori. I parametri strutturali γ sono coefficienti di regressione che legano le variabili endogene a quelle esogene, mentre i parametri β legano tra loro le variabili endogene. Le ultime tre equazioni sono senza errori né parametri strutturali: si tratta di identità. Occorre stimare i coefficienti di regressione ignoti delle prime tre equazioni. Un metodo di stima è quello dei minimi quadrati a due stadi (2SLS, Two Stage Least Squares) basato sulle variabili strumentali (instrumental variables, IV)[30].
Le variabili strumentali sono delle variabili incorrelate con gli errori delle equazioni strutturali: nel contesto in cui stiamo operando, ad esempio, le variabili esogene possono essere usate come IV. Scriviamo un'equazione strutturale in forma matriciale:

$$y = X\delta + \zeta$$

dove **y** è il vettore (n x 1) della variabile risposta, **X** è la matrice (n x p) che comprende le variabili esogene e quelle endogene più una colonna di 1 per l'intercetta, mentre $\boldsymbol{\delta}$ è il vettore (p x 1) dei parametri strutturali e $\boldsymbol{\zeta}$ è il vettore (n x 1) degli errori. Sia **Z** la matrice (n x p) delle variabili strumentali. Moltiplicando ambo i membri dell'equazione strutturale per **Z** si ha:

$$Z'y = Z'X\delta + Z'\zeta$$

poiché $p\lim \frac{1}{n} Z'\zeta = 0$ (per definizione le IV sono al limite incorrelate con gli errori) la stima di $\hat{\delta}$ con le IV è una stima consistente di $\boldsymbol{\delta}$:

$$\hat{\delta} = (Z'X)^{-1} Z'y$$

[29] Si veda: J. FOX, *An R and S-PLUS Companion to Applied Regression*, 2002
[30] Cfr. F. DEL VECCHIO, *Analisi op. cit*., pagg. 229-231

I minimi quadrati a 2 stadi (Two-Stage Least Squares, 2SLS) si possono pensare come una concatenazione di 2 regressioni ordinarie (OLS). Nel primo stadio, i predittori **X** sono fatti regredire sulla variabili strumentali **Z** e si ottengono i valori stimati:

$$\hat{X} = X(Z'Z)^{-1}Z'X$$

nel secondo stadio, la variabile risposta **y** viene fatta regredire sui valori stimati al primo stadio $\hat{X}$ e si ottiene la stima 2SLS di δ:

$$\hat{\delta} = (\hat{X}'\hat{X})^{-1}\hat{X}'y$$

I due stadi dei 2SLS possono essere combinati algebricamente e si ottiene la seguente espressione della stima di δ:

$$\hat{\delta} = [X'Z(Z'Z)^{-1}Z'X]^{-1}X'Z(Z'Z)^{-1}Z'y$$

Vogliamo usare il metodo 2SLS per stimare le equazioni strutturali del modello di Klein. Richiamiamo i dati presenti nel package `sem`[31]:

```
library(sem)
data(Klein)
Klein
   year    c    p   wp    i k.lag    x  wg    g    t
1  1920 39.8 12.7 28.8  2.7 180.1 44.9 2.2  2.4  3.4
2  1921 41.9 12.4 25.5 -0.2 182.8 45.6 2.7  3.9  7.7
3  1922 45.0 16.9 29.3  1.9 182.6 50.1 2.9  3.2  3.9
4  1923 49.2 18.4 34.1  5.2 184.5 57.2 2.9  2.8  4.7
5  1924 50.6 19.4 33.9  3.0 189.7 57.1 3.1  3.5  3.8
6  1925 52.6 20.1 35.4  5.1 192.7 61.0 3.2  3.3  5.5
7  1926 55.1 19.6 37.4  5.6 197.8 64.0 3.3  3.3  7.0
8  1927 56.2 19.8 37.9  4.2 203.4 64.4 3.6  4.0  6.7
9  1928 57.3 21.1 39.2  3.0 207.6 64.5 3.7  4.2  4.2
10 1929 57.8 21.7 41.3  5.1 210.6 67.0 4.0  4.1  4.0
11 1930 55.0 15.6 37.9  1.0 215.7 61.2 4.2  5.2  7.7
12 1931 50.9 11.4 34.5 -3.4 216.7 53.4 4.8  5.9  7.5
13 1932 45.6  7.0 29.0 -6.2 213.3 44.3 5.3  4.9  8.3
14 1933 46.5 11.2 28.5 -5.1 207.1 45.1 5.6  3.7  5.4
15 1934 48.7 12.3 30.6 -3.0 202.0 49.7 6.0  4.0  6.8
16 1935 51.3 14.0 33.2 -1.3 199.0 54.4 6.1  4.4  7.2
17 1936 57.7 17.6 36.8  2.1 197.7 62.7 7.4  2.9  8.3
18 1937 58.7 17.3 41.0  2.0 199.8 65.0 6.7  4.3  6.7
19 1938 57.5 15.3 38.2 -1.9 201.8 60.9 7.7  5.3  7.4
20 1939 61.6 19.0 41.6  1.3 199.9 69.5 7.8  6.6  8.9
21 1940 65.0 21.1 45.0  3.3 201.2 75.7 8.0  7.4  9.6
22 1941 69.7 23.5 53.3  4.9 204.5 88.4 8.5 13.8 11.6
```

apportiamo alcune modifiche e costruiamo le variabili mancanti:

```
attach(Klein)
p.lag<-c(NA,p[-length(p)])
x.lag<-c(NA,x[-length(x)])
a<-year-1931
```

[31] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/sem.html

```
cbind(year,a,p,p.lag,x,x.lag)
      year   a    p p.lag    x x.lag
 [1,] 1920 -11 12.7    NA 44.9    NA
 [2,] 1921 -10 12.4  12.7 45.6  44.9
 [3,] 1922  -9 16.9  12.4 50.1  45.6
 [4,] 1923  -8 18.4  16.9 57.2  50.1
 [5,] 1924  -7 19.4  18.4 57.1  57.2
 [6,] 1925  -6 20.1  19.4 61.0  57.1
 [7,] 1926  -5 19.6  20.1 64.0  61.0
 [8,] 1927  -4 19.8  19.6 64.4  64.0
 [9,] 1928  -3 21.1  19.8 64.5  64.4
[10,] 1929  -2 21.7  21.1 67.0  64.5
[11,] 1930  -1 15.6  21.7 61.2  67.0
[12,] 1931   0 11.4  15.6 53.4  61.2
[13,] 1932   1  7.0  11.4 44.3  53.4
[14,] 1933   2 11.2   7.0 45.1  44.3
[15,] 1934   3 12.3  11.2 49.7  45.1
[16,] 1935   4 14.0  12.3 54.4  49.7
[17,] 1936   5 17.6  14.0 62.7  54.4
[18,] 1937   6 17.3  17.6 65.0  62.7
[19,] 1938   7 15.3  17.3 60.9  65.0
[20,] 1939   8 19.0  15.3 69.5  60.9
[21,] 1940   9 21.1  19.0 75.7  69.5
[22,] 1941  10 23.5  21.1 88.4  75.7
```

Usiamo il comando `tsls` presente nel package `sem` per la stima 2SLS delle equazioni strutturali. Si tenga presente che:

1) le equazioni strutturali da stimare vengono specificate tramite una formula, come nel comando `lm`;
2) le variabili strumentali vengono indicate come formula nell'argomento `instruments`;

```
eq1<-tsls(c~p+p.lag+I(wp+wg), instruments=~g+t+wg+a+p.lag+k.lag+x.lag)

summary(eq1)

 2SLS Estimates

Model Formula: c ~ p + p.lag + I(wp + wg)

Instruments: ~g + t + wg + a + p.lag + k.lag + x.lag

Residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-1.89e+00 -6.16e-01 -2.46e-01  4.34e-11  8.85e-01  2.00e+00

            Estimate Std. Error t value  Pr(>|t|)
(Intercept) 16.55476    1.46798 11.2772 2.587e-09
p            0.01730    0.13120  0.1319 8.966e-01
p.lag        0.21623    0.11922  1.8137 8.741e-02
I(wp + wg)   0.81018    0.04474 18.1107 1.505e-12

Residual standard error: 1.1357 on 17 degrees of freedom
```

In modo del tutto analogo possono ottenersi le stime delle altre 2 equazioni strutturali del modello di Klein:

```
eq2<-tsls(i~p+p.lag+k.lag, instruments=~g+t+wg+a+p.lag+k.lag+x.lag)
eq3<-tsls(wp~x+x.lag+a, instruments=~g+t+wg+a+p.lag+k.lag+x.lag)
```

Come variabili strumentali abbiamo usato le 4 variabili esogene e 3 variabili endogene predeterminate. Agli oggetti di classe `tsls` possono applicarsi i seguenti metodi:

```
methods(class="tsls")
[1] anova.tsls        coefficients.tsls fitted.tsls       print.tsls
[5] residuals.tsls    summary.tsls
```

Il comando `systemfit()` dell'omonimo package[32] consente di stimare i parametri un sistema di equazioni strutturali lineari usando diversi metodi: Ordinary Least Squares (OLS), Weighted Least Squares (WLS), Seemingly Unrelated Regression (SUR), Two-Stage Least Squares (2SLS), Weighted Two-Stage Least Squares (W2SLS) or Three-Stage Least Squares (3SLS). Vediamo qualche esemplificazione pratica.
Occorre specificare il metodo di stima nell'argomento `method` ("OLS", "WLS", "SUR", "2SLS", "W2SLS" o "3SLS"), la lista delle equazioni strutturali da stimare in `eqns`, una lista opzionale di etichette (label) per i nomi delle equazioni strutturali in `eqnlabels`, le variabili strumentali (necessarie solo per le stime 2SLS, W2SLS e 3SLS) in `inst`, la matrice di eventuali restrizioni lineari in `R.restr` e il dataframe contenente i dati in `data`.

```
data(kmenta)
demand <- q ~ p + d
supply <- q ~ p + f + a
labels <- list( "demand", "supply" )
system <- list( demand, supply )


## stima OLS
fitols <- systemfit("OLS", system, labels, data=kmenta )
print(fitols)

systemfit results
method: OLS

        N DF     SSR     MSE    RMSE       R2   Adj R2
demand 20 17 63.3317 3.72539 1.93013 0.763789 0.735999
supply 20 16 92.5511 5.78444 2.40509 0.654807 0.590084

The covariance matrix of the residuals
        demand  supply
demand 3.72539 4.13696
supply 4.13696 5.78444

The correlations of the residuals
         demand   supply
demand 1.000000 0.891179
supply 0.891179 1.000000

The determinant of the residual covariance matrix: 4.43485
OLS R-squared value of the system: 0.709298

OLS estimates for demand  (equation 1 )
Model Formula: q ~ p + d
```

[32] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/systemfit.html

```
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 99.895423   7.519362 13.285093        0 ***
p           -0.316299   0.090677 -3.488177 0.002815  **
d            0.334636   0.045422  7.367285    1e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.930127 on 17 degrees of freedom
Number of observations: 20 Degrees of Freedom: 17
SSR: 63.33165 MSE: 3.725391 Root MSE: 1.930127
Multiple R-Squared: 0.763789 Adjusted R-Squared: 0.735999


## stima SUR iterativa
fitsur <- systemfit("SUR", system, labels, data=kmenta, maxit=100 )
print( fitsur )

systemfit results
method: iterated SUR

convergence achieved after 35 iterations

        N DF     SSR     MSE    RMSE       R2   Adj R2
demand 20 17 105.389 6.19935 2.48985 0.606925 0.560681
supply 20 16 146.061 9.12884 3.02140 0.455227 0.353082

The covariance matrix of the residuals used for estimation
        demand  supply
demand 6.19907 7.49338
supply 7.49338 9.12855

The covariance matrix of the residuals
        demand  supply
demand 6.19935 7.49367
supply 7.49367 9.12884

The correlations of the residuals
         demand   supply
demand 1.000000 0.996125
supply 0.996125 1.000000

The determinant of the residual covariance matrix: 0.437694
OLS R-squared value of the system: 0.531076
McElroy's R-squared value for the system: 0.832629

SUR estimates for demand  (equation 1 )
Model Formula: q ~ p + d

             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 97.516307   9.663008 10.091713        0 ***
p           -0.143687    0.09971  -1.44104  0.16774
d             0.18202   0.022572  8.064129        0 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.489849 on 17 degrees of freedom
Number of observations: 20 Degrees of Freedom: 17
```

```
SSR: 105.388914 MSE: 6.199348 Root MSE: 2.489849
Multiple R-Squared: 0.606925 Adjusted R-Squared: 0.560681


SUR estimates for supply  (equation 2 )
Model Formula: q ~ p + f + a

             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 77.900537   12.12427 6.425173    8e-06 ***
p            0.105094   0.117246 0.896349 0.383355
f             0.10841   0.020513 5.284965  7.4e-05 ***
a            0.191543   0.032047 5.976971  1.9e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.021397 on 16 degrees of freedom
Number of observations: 20 Degrees of Freedom: 16
SSR: 146.06139 MSE: 9.128837 Root MSE: 3.021397
Multiple R-Squared: 0.455227 Adjusted R-Squared: 0.353082

## stima 2SLS
inst <- ~ d + f + a
fit2sls <- systemfit( "2SLS", system, labels, inst, kmenta )
print( fit2sls )

systemfit results
method: 2SLS

        N DF     SSR     MSE    RMSE       R2   Adj R2
demand 20 17 65.7291 3.86642 1.96632 0.754847 0.726005
supply 20 16 96.6332 6.03958 2.45756 0.639582 0.572004

The covariance matrix of the residuals
        demand  supply
demand 3.86642 4.35744
supply 4.35744 6.03958

The correlations of the residuals
         demand   supply
demand 1.000000 0.901724
supply 0.901724 1.000000

The determinant of the residual covariance matrix: 4.36424
OLS R-squared value of the system: 0.697214

2SLS estimates for demand  (equation 1 )
Model Formula: q ~ p + d
Instruments: ~d + f + a

             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 94.633304   7.920838 11.947385        0 ***
p           -0.243557   0.096484 -2.524313 0.021832   *
d            0.313992   0.046944  6.688695    4e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.966321 on 17 degrees of freedom
```

```
Number of observations: 20 Degrees of Freedom: 17
SSR: 65.729088 MSE: 3.866417 Root MSE: 1.966321
Multiple R-Squared: 0.754847 Adjusted R-Squared: 0.726005


2SLS estimates for supply  (equation 2 )
Model Formula: q ~ p + f + a
Instruments: ~d + f + a

             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 49.532442  12.010526 4.124086 0.000795 ***
p            0.240076   0.099934 2.402347 0.028785   *
f            0.255606    0.04725 5.409637  5.8e-05 ***
a            0.252924   0.099655 2.537996 0.021929   *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.457555 on 16 degrees of freedom
Number of observations: 20 Degrees of Freedom: 16
SSR: 96.633244 MSE: 6.039578 Root MSE: 2.457555
Multiple R-Squared: 0.639582 Adjusted R-Squared: 0.572004
```

8.0 Regressione non lineare e non linear least squares (NLS)

Nel caso in cui i parametri della funzione di regressione da stimare figurino in questa in forma non lineare, ossia compaiono con grado diverso dal primo, i minimi quadrati ordinari (OLS) non possono più essere applicati e occorre percorre altre strade[33]. Sia $f(\mathbf{x},\boldsymbol{\beta})$ la funzione di regressione, anche in questo caso occorre rendere minima la somma dei quadrati degli scarti per stimare il vettore dei parametri $\boldsymbol{\beta}$:

$$\sum_{i=1}^{n}[y_i - f(x_i,\boldsymbol{\beta})]^2 = \min.$$

A causa della non linearità dei parametri la soluzione del problema avviene mediante metodi iterativi di calcolo numerico. In ambiente `R` possiamo optare tra diverse alternative. Ad esempio nel caso della regressione logistica o di quella di Poisson si ricorre al comando `glm()`(si veda il relativo paragrafo). A parte questo caso, vi sono due tipi di soluzione che possono essere adoperate. Si tratta dei comandi `nlm()`(non linear minimizazion) e `nls()`(non linear least squares)[34].
Il comando `nlm()` permette di minimizzare una funzione specificata, indicando i valori iniziali dei parametri da stimare, ricorrendo ad algoritmo di tipo Newton. Prendiamo una semplice funzione non lineare nei due suoi parametri:

$$y = \frac{\alpha x}{x + \beta} + \varepsilon$$

si tratta del modello di Michaelis-Menten usato nella cinetica enzimatica. Proveremo ad usare questo modello per descrivere la relazione tra la variabile `mortal` e la variabile `popphys` nel dataframe `mortalità` ed useremo il comando `nlm()`. Scriviamo prima la funzione da minimizzare, ossia la sommatoria degli scarti al quadrato, il vettore `p` contiene i parametri da stimare:

```
fn <- function(p) sum((y - (p[1] * x)/(p[2] + x))^2)## funzione da minimizzare
```

[33] Per approfondimenti si vedano: J. FOX, *An R... op. cit.*, cap. 23 e J. FOX, *Statistical Applications in Social Research: Lecture Notes and R Scripts,* 2004
[34] Si veda: R DEVELOPMENT CORE TEAM, An *introduction to R R. 2.3.1*., 1 giugno 2006, pag. 58

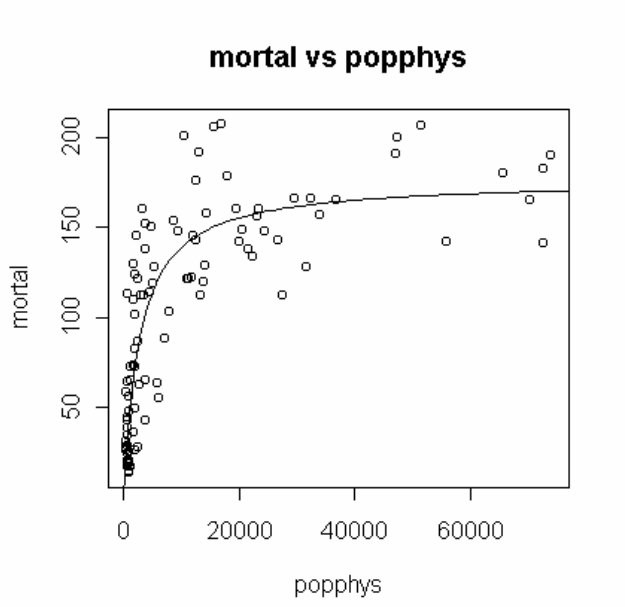
```
y<-mortalita$mortal
x<-mortalita$popphys
out <- nlm(fn, p = c(200, 0.1), hessian = TRUE)

out ## risultato della minimizzazione

$minimum
[1] 90809.47

$estimate
[1]  176.6069 2712.1727

$gradient
[1] -3.913868e-05  7.511572e-08

$hessian
          [,1]        [,2]
[1,] 84.882644 -1.10543426
[2,] -1.105434  0.02615106

$code
[1] 1

$iterations
[1] 22

p<-out$estimate ## stime dei parametri
```

le stime dei parametri del modello di Michaelis-Menten sono pari a 176,6069 e 2.712,1727. Tracciamo ora il grafico dei valori osservati con sovraimpressa la curva dei valori stimati:

```
xfit<-seq(0,80000, 1000)
yfit <-p[1] * xfit/(p[2] + xfit)
plot(x,y, xlab="popphys", ylab="mortal", main="mortal vs popphys")
lines(xfit, yfit)
```

come può vedersi dal grafico (Fig. 40) l'adattamento risulta essere molto buono.
Il comando `nlm()` può essere anche applicato con il metodo di stima della massima verosimiglianza: in questo caso la funzione da minimizzare sarà il negativo della log-verosimiglianza.
La stima dei minimi quadrati in caso di non linearità nei parametri può essere anche risolto ricorrendo al comando `nls()`. Riprendiamo l'esempio precedente:

Fig. 40




```
df <- data.frame(x=x, y=y) ## creaiamo un dataframe con le due variabili
fit <- nls(y ~ SSmicmen(x, Vm, K), data=df)
summary(fit)

Formula: y ~ SSmicmen(x, Vm, K)

Parameters:
   Estimate Std. Error t value Pr(>|t|)
Vm  176.607      7.013  25.182  < 2e-16 ***
K  2712.185    399.937   6.782 9.36e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 30.6 on 97 degrees of freedom

Correlation of Parameter Estimates:
      Vm
K 0.7426
```

Nella formula del modello in questo caso si è fatto riferimento alla funzione di Michaelis-Menten già prevista dall'ambiente `R`, tuttavia si possono inserire anche espressioni definite dall'utente. Si consideri il seguente esempio di funzione di regressione non lineare nei parametri (α, β, γ)

$$y = \alpha - \beta e^{-\gamma x} + \varepsilon$$

la stima dei parametri con il comando `nls()` sarà effettuata in questo modo, specificando le stime iniziali degli stessi:

```
fit2<-nls(y~a-b*exp(-c*x),start=list(a=50, b=100, c=0.5))
```

Il comando `gnls()` presesente nel package `nlme` consente di effettuare delle stime con il metodo GLS per funzioni di regressione non lineari.

9.0 Regressione ortogonale

La regressione ortogonale, detta anche regressione di Deming, viene usata quando si trattano variabili affette da errori come nel caso delle analisi cliniche quando si studia il rapporto di concentrazione di due sostanze nel sangue[35]. Siano Y ed X due variabili misurabili affette da errore:

$X_i = x_i + \varepsilon_i$ e $Y_i = y_i + \eta_i$ con i=1…n

dove ε_i e η_i sono gli errori con distribuzione:

$\varepsilon_i \sim \mathrm{N}(0, \sigma^2_{\varepsilon_i})$ e $\eta_i = N(0, \sigma^2_{\eta_i})$.

Inoltre abbiamo che: $\mathrm{cov}(\varepsilon_i, \varepsilon_j) = \mathrm{cov}(\eta_i, \eta_j) = 0 \quad \forall i \neq j$ e $\mathrm{cov}(\varepsilon_i, \eta_i) = 0$, ossia gli errori sono in correlati tra loro; le varianze non sono costanti, ma è costante il loro rapporto: $\lambda = \frac{\sigma^2_{\varepsilon_i}}{\sigma^2_{\eta_i}}$. Se tra le variabili Y e X esiste una relazione lineare:

$Y_i = \alpha + \beta X_i$

la regressione di Deming stima i parametri della retta minimizzando la somma dei quadrati delle distanze perpendicolari tra i punti (Y_i, X_i) e la retta stessa. Gli scarti ortogonali sono ottenuti applicando la formula della distanza euclidea:

$$\frac{| Y_i - \alpha - \beta X_i |}{\sqrt{1+\beta^2}}$$

per cui i parametri della retta vanno stimati minimizzando la quantità:

$$S = \frac{\sum_{i=1}^{n} (Y_i - \alpha - \beta X_i)^2}{1+\beta^2}$$

Se poniamo $k = \frac{Dev(X)}{Dev(Y)}$ la stima di β è la seguente:

$$\hat{\beta} = k[Dev(Y) - Dev(X)] + \frac{\sqrt{[Dev(X) - kDev(Y)]^2 + 4kCod(X,Y)^2}}{2kCod(X,Y)}$$

mentre l'intercetta può stimarsi facilmente:

$\hat{\alpha} = \overline{Y} - \hat{\beta}\overline{X}$

La stima dei parametri della retta di Deming in R può ottenersi agevolmente ricorrendo al comando `optim()` che minimizza una funzione con procedure di calcolo numerico. La funzione da minimizzare è S. Procediamo con dei dati simulati:

```
set.seed(13)
x<-rnorm(50,5,1)
y<-100+7*x+rnorm(50)

f<-function(p)(sum(y-p[1]-p[2]*x)^2)/(1+p[2]^2)## funzione da minimizzare

fit<-optim(c(100,15),f, method=c("Nelder-Mead"))
```

[35] Cfr. L. SOLIANI, *Statistica univariata e bivariata parametrica e non-parametrica per le discipline ambientali e biologiche*, 2005, cap. 24, pagg. 25 e segg.

```
fit
$par
[1] 108.93887   5.18306

$value
[1] 3.230574e-05

$counts
function gradient
      45       NA

$convergence
[1] 0

$message
NULL
```

Le stime dei parametri della retta di Deming sono le seguenti:

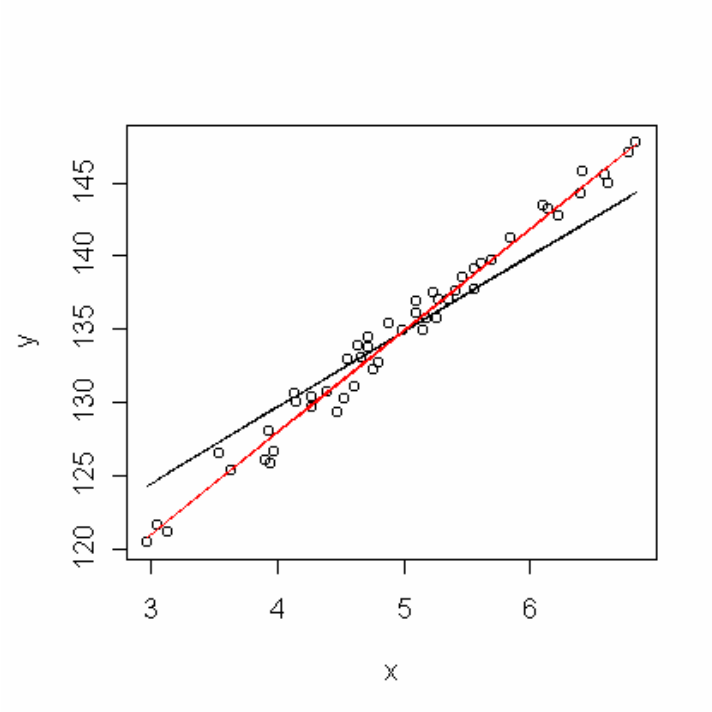
```
fit$par
[1] 108.93887   5.18306

yfitdeming<-fit$par[1]+fit$par[2]*x ##sono i valori della y stimati con
la retta di Deming
```

Per un confronto le stime OLS sono:

```
fm<-lm(y~x)
coef(fm)
(Intercept)           x
 100.115778    6.957172

yfit<-coef(fm)[1]+coef(fm)[2]*x ##sono i valori della y stimati con la
retta OLS

plot(x,y)
lines(x,yfitdeming)
lines(x,yfit, col="red")
```

Nella Fig 41 abbiamo riportato sia la retta di regressione di Deming (linea nera) che la retta di regressione OLS.

Fig. 41



10.0 Regressione robusta

Quando nella regressione gli errori non sono distribuiti normalmente oppure si hanno molti valori outliers le stime OLS non sono buone e si deve ricorrere alla regressione robusta[36]. Il metodo più usato per questo tipo di regressione è quello della *M-estimation* introdotto da Huber. Tali stimatori possono essere considerati una generalizzazione delle stime di massima verosimiglianza. Abbiamo il modello lineare generale (con k regressori) riferito all'i.esima di n osservazioni ed espresso in notazione vettoriale:

$$y_i = \mathbf{x_i}'\boldsymbol{\beta} + \varepsilon_i$$

sia il modello stimato:

$$y_i = \mathbf{x_i}'\mathbf{b} + e_i$$

Il generico *M-estimator* è ottenuto minimizzando la seguente *funzione obiettivo*:

$$\sum_{i=1}^{n} \rho(e_i) = \sum_{i=1}^{n} \rho(y_i - \mathbf{x_i}'\mathbf{b})$$

dove la funzione ρ fornisce il contributo di ogni singolo residuo alla funzione obiettivo. La funzione ρ deve godere delle seguenti proprietà:

1) $\rho(e) \geq 0$
2) $\rho(0) = 0$
3) $\rho(e) = \rho(-e)$

[36] Cfr. J.FOX, *An R.... op cit.*

4) $\rho(e_i) \geq \rho(e_j) \quad se \quad |e_i| > |e_j|$

Ci sono diverse funzioni obiettivo che possono essere utilizzate e che soddisfano questi requisiti: se $\rho(e_i) = e_i^2$ abbiamo le stime dei minimi quadrati ordinari (OLS), se $\rho(e_i) = |e_i|$ abbiamo le stime LAD (Least absolute deviation regression), al funzione sono quella di Huber e la biquadratica (si veda la Tabella 1).

Tabella 1

Method	*Objective Function*	*Weight Function*
Least-Squares	$\rho_{LS}(e) = e^2$	$w_{LS}(e) = 1$
Huber	$\rho_H(e) = \begin{cases} \frac{1}{2}e^2 & \text{for } \lvert e\rvert \leq k \\ k\lvert e\rvert - \frac{1}{2}k^2 & \text{for } \lvert e\rvert > k \end{cases}$	$w_H(e) = \begin{cases} 1 & \text{for } \lvert e\rvert \leq k \\ k/\lvert e\rvert & \text{for } \lvert e\rvert > k \end{cases}$
Bisquare	$\rho_B(e) = \begin{cases} \frac{k^2}{6}\left\{1 - \left[1 - \left(\frac{e}{k}\right)^2\right]^3\right\} & \text{for } \lvert e\rvert \leq k \\ k^2/6 & \text{for } \lvert e\rvert > k \end{cases}$	$w_B(e) = \begin{cases} \left[1 - \left(\frac{e}{k}\right)^2\right]^2 & \text{for } \lvert e\rvert \leq k \\ 0 & \text{for } \lvert e\rvert > k \end{cases}$

Fonte: J. FOX, *An R and S-PLUS Companion to Applied Regression*, 2002

Se poniamo $\psi = \rho'$ (la derivata prima) e differenziamo la funzione obiettivo rispetto ai coefficienti **b**, poniamo uguali a zero le derivate parziali, otteniamo un sistema di k+1 equazioni di stima per i coefficienti:

$$\sum_{i=1}^{n} \psi(y_i - \mathbf{x_i}'\mathbf{b})\mathbf{x_i}' = \mathbf{0}$$

Se definiamo la *funzione peso* $w(e) = \psi(e)/e$ e poniamo $w_i = w(e_i)$ le equazioni possono essere riscritte in questi termini:

$$\sum_{i=1}^{n} w_i(y_i - \mathbf{x_i}'\mathbf{b})\mathbf{x_i}' = \mathbf{0}$$

La soluzione di tale sistema è un problema di minimi quadrati ponderati, minimizzando $\sum w_i^2 e_i^2$, e si ottiene attraverso una procedura iterativa (IRLS=*iteratively reweigheted least squares*).
Un'altra tecnica di regressione robusta è quella della regressione LTS (*least trimmed squares*) . In questo caso i quadrati dei residui vengono ordinati in ordine crescente:

$(e^2)_{(1)}, (e^2)_{(2)}, \ldots (e^2)_{(n)}$

le stime LTS dei coefficienti di regressione **b** sono ottenute minimizzando la somma dei piccoli m valori dei quadrati dei residui:

$$LTS(\mathbf{b}) = \sum_{i=1}^{m} (e^2)_{(i)}$$

con $m = \lfloor n/2 \rfloor + \lfloor (k+2)/2 \rfloor$ dove $\lfloor \ \rfloor$ indica l'approssimazione all'intero più piccolo.

Applicheremo i comandi di R per la regressione robusta ai dati contenuti nel dataframe mortalità in precedenza già utilizzato.
Il primo comando per la regressione robusta che usa la funzione di Huber o la biquadratica è `rlm()`(robust linear model) contenuta nel package `MASS`, sempre nello stesso package è compreso il comando `lqs()` per la regressione LTS.

```
library(MASS)
fm.rlm1<-rlm(mortal~.+., data=mortalita, psi=psi.huber) ## con funzione
di Huber
summary(fm.rlm1)
```

```
Call: rlm(formula = mortal ~ . + ., data = mortalita, psi = psi.huber)
Residuals:
     Min       1Q   Median       3Q      Max
-57.9787 -16.8596   0.1901  16.7643  78.0593

Coefficients:
            Value    Std. Error t value
(Intercept) 183.6433  20.6034     8.9133
Calorie      -0.0278   0.0096    -2.8995
HS           -1.2526   0.2241    -5.5888
popphys       0.0007   0.0002     3.3229
popnurs       0.0013   0.0006     2.0771

Residual standard error: 25.08 on 94 degrees of freedom

Correlation of Coefficients:
        (Intercept) Calorie HS      popphys
Calorie -0.9528
HS       0.5129     -0.7179
popphys -0.2680      0.1050  0.2339
popnurs -0.0957     -0.0124  0.1765 -0.2915


fm.rlm2<-rlm(mortal~.+.,   data=mortalita,   psi=   psi.bisquare)##   con
funzione biquadratica
summary(fm.rlm2)

Call: rlm(formula = mortal ~ . + ., data = mortalita, psi = psi.bisquare)
Residuals:
     Min       1Q   Median       3Q      Max
-57.8461 -16.7200   0.4317  16.7660  77.9710

Coefficients:
            Value    Std. Error t value
(Intercept) 182.0746  20.4456     8.9053
Calorie      -0.0271   0.0095    -2.8423
HS           -1.2718   0.2224    -5.7184
popphys       0.0007   0.0002     3.3028
popnurs       0.0013   0.0006     2.1154

Residual standard error: 25 on 94 degrees of freedom

Correlation of Coefficients:
        (Intercept) Calorie HS      popphys
Calorie -0.9528
HS       0.5129     -0.7179
popphys -0.2680      0.1050  0.2339
popnurs -0.0957     -0.0124  0.1765 -0.2915
```

Come si può vedere la funzione `rlm()` è molto simile nella sintassi e nell'output al comando `lm()`, in più occorre specificare il tipo della funzione dei pesi nell'argomento `psi` (che può assumere i valori "`psi.huber`", "`psi.bisquare`", "`psi.hampel`").

Il comando `lqs()` consente di calcolare tra l'altro la regressione LTS, specificando l'argomento `method=lts`; è possibile usare altri metodi come lqs e lms (least quantile squared residuals) che differiscono leggermente tra loro per il calcolo del valore m (si veda l'help del comando).

```
fm.lqs<-lqs(mortal~.+., data=mortalita, method="lqs") ## stima lqs
fm.lqs
Call:
lqs.formula(formula = mortal ~ . + ., data = mortalita, method = "lqs")

Coefficients:
(Intercept)      Calorie           HS      popphys      popnurs
  76.488801     0.001574    -0.985841     0.002043     0.003309

Scale estimates 19.32 20.65

fm.lts<-lqs(mortal~.+., data=mortalita, method="lts") ## stima lts
fm.lts
Call:
lqs.formula(formula = mortal ~ . + ., data = mortalita, method = "lts")

Coefficients:
(Intercept)      Calorie           HS      popphys      popnurs
 138.301782    -0.024175    -0.692202     0.001967     0.003466

Scale estimates 20.92 19.42
```

Confrontiamo le stime dei coefficienti di regressione calcolati con i vari metodo di regressione robusta e quelli calcolati con il metodo OLS:

```
confronto<-data.frame(coef(fm.rlm1),coef(fm.rlm2),coef(fm.lqs),coef(fm.lts),
coef(fm.ols))

options(digits=3)
confronto
           coef.fm.rlm1. coef.fm.rlm2. coef.fm.lqs. coef.fm.lts. coef.fm.ols.
(Intercept)      1.84e+02      1.82e+02     76.48880    138.30178     1.89e+02
Calorie         -2.78e-02     -2.71e-02      0.00157     -0.02418    -2.96e-02
HS              -1.25e+00     -1.27e+00     -0.98584     -0.69220    -1.23e+00
popphys          6.62e-04      6.53e-04      0.00204      0.00197     6.02e-04
popnurs          1.27e-03      1.28e-03      0.00331      0.00347     1.29e-03
```

11.0 Regressione Quantilica

La regressione quantilica[37] può essere in alcune circostanze una valida alternativa alla regressione ordinaria quando non sono verificati tutti i requisiti di base per applicare i minimi quadrati ordinari (OLS), in particolare quando si hanno valori outlier (la regressione quantilica è una tecnica robusta) e le stime dei coefficienti di regressione risultano più efficienti quando gli errori non sono distribuiti normalmente.
Si abbia una variabile risposta Y e un insieme di regressori (covariate) **X**; sia $F_{Y|X}(y \mid x)$ la funzione di ripartizione di Y condizionata a **X**, sia $0 < \tau < 1$ definiamo quantile condizionato la seguente funzione: $Q_{Y|X}(\tau \mid x) = \inf\{y : F_{Y|X}(y \mid x) \geq \tau\}$; con alcune dimostrazioni è facile vedere che $Q_{Y|X}(\tau \mid x) = \arg\min_a E[\rho_\tau(y-a) \mid x]$ dove $\rho_\tau(u) = u(\tau - I(u<0))$ con I la funzione indicatrice.

[37] Cfr R. KOENKER, K. HALLOCK, *Quantile Regression*, Journal of Economic Perspectives, 15, 2001, 143-156

Se assumiamo che $Q_{Y|X}(\tau \mid x) = x'\beta(\tau)$ si dimostra che a livello di popolazione si ha che $\beta(\tau) = \underset{b}{\operatorname{argmin}}\, E[\rho_\tau(Y - X'b)]$. Una stima di questo parametro con i dati campionari è data da:

$\hat{\beta}(\tau) = \underset{b}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} \rho_\tau(y_i - x_i'b)$. Si possono avere diverse regressioni quantiliche a seconda del valore di τ e quindi del quantile da stimare.
Nell'ambiente R per effettuare una regressione quantilica si ricorre al comando `rq()` contenuto nel package `quantreg`[38] (questo package va scaricato ed installato). Il comando `rq()` funziona in maniera simile a `lm()` specificando la formula del modello stimare e fissare l'argomento tau (un numero compreso tra 0 e 1 per specificare il quantile da stimare, per default è 0.5, cioè la mediana)

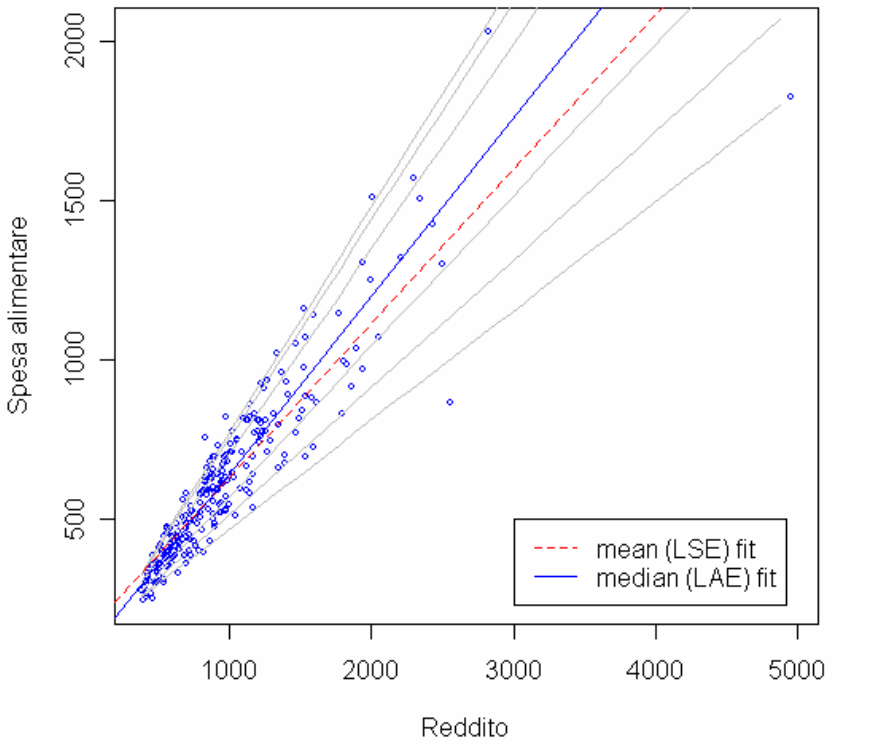
```
library(quantreg) ## per caricare il package dopo averlo scaricato

data(engel)
attach(engel)
plot(x,y,xlab="Reddito",ylab="Spesa alimentare",type = "n", cex=.5)
points(x,y,cex=.5,col="blue")
taus <- c(.05,.1,.25,.75,.9,.95)
xx <- seq(min(x),max(x),100)
f <- coef(rq((y)~(x),tau=taus))
yy <- cbind(1,xx)%*%f
for(i in 1:length(taus)){
    lines(xx,yy[,i],col = "gray")
    }
abline(lm(y~x),col="red",lty = 2)
abline(rq(y~x),col="blue")
legend(3000,500,c("mean   (LSE)   fit",   "median   (LAE)   fit"),col   =
c("red","blue"),lty = c(2,1))
```

In questo esempio si sono presi di dati di Engel usati per lo studio della spesa alimentare delle famiglie in relazione al reddito delle stesse; in particolare (Fig. 42) si sono stimate sei rette di regressione quantilica (riportate in colore grigio sulla Fig. 42) per sei valori di τ (0.05;0.1;0.25;0.75;0.9;0.95); inoltre sempre nel medesimo grafico sono riportale in colore blu la retta di regressione quantilica corrispondente a $\tau = 0.5$, ossia la mediana, ed in colore rosso la retta di regressione OLS.
Possiamo visualizzare i valori dei parametri delle rette di regressione:

```
f
              tau= 0.05   tau= 0.10  tau= 0.25  tau= 0.75  tau= 0.90  tau= 0.95
(Intercept) 124.8800408 110.1415742 95.4835396 62.3965855 67.3508721 64.1039632
x             0.3433611   0.4017658  0.4741032  0.6440141  0.6862995  0.7090685

coef(rq(y~x,tau=0.5)) ##  regr. quantilica in corrispondenza della mediana
(Intercept)           x
 81.4822474   0.5601806

coef(lm(y~x)) ## regr. OLS
(Intercept)           x
147.4753885   0.4851784
```

[38] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/quantreg.html, http://www.maths.lth.se/help/R/.R/library/quantreg/doc/rq.pdf e R. KOENKER, *Quantile Regression in R: a vignette* per la documentazione

Fig.42




```
summary(rq(y~x))

Call: rq(formula = y ~ x)

tau: [1] 0.5

Coefficients:
            coefficients lower bd  upper bd
(Intercept)  81.48225     53.25915 114.01156
x             0.56018      0.48702   0.60199
```

Nel caso di modello non lineare nei parametri si può usare il comando `nlrq()` contenuto sempre nel package `quantreg`.

12.0 Regressione non parametrica

La regressione non parametrica[39] studia la dipendenza tra una variabile risposta (y) e uno o più regressori $(x_1 ... x_p)$ senza che sia specificata la funzione che lega la risposta ai regressori:

$$y_i = f(x_{1i}, ... x_{pi}) + \varepsilon_i$$

con gli errori ε_i che si distribuiscono normalmente con media 0 e varianza costante.

[39] Cfr. J.FOX, *An R..op. cit.* e J. FOX , *Nonparametric Regression*, Febbraio 2004

Prendiamo in considerazione due tipologie comuni di regressione non polinomiale nel caso di un solo regressore: la *kernel estimation* e la *local polinomial regression* (regressione locale polinomiale) che è una generalizzazione della precedente. Con la *kernel estimation* si procede in questo modo:

1) abbiamo n osservazioni; sia x_0 il valore focale del regressore x in corrispondenza del quale $f(x)$ deve essere stimata. Occorre trovare gli m valori più vicini (prossimi) a x_0 partendo dal parametro di span del kernel smoother $s = m/n$, quindi $m = s \cdot n$; sia h la semi-ampiezza dell'intervallo che comprende gli m valori prossimi a x_0;

2) si definisce una funzione peso simmetrica, unimodale e centrata su x_0, ad esempio la tricubica:

$$W_T(x) = \begin{cases} \left[1 - \left(\frac{|x - x_0|}{h}\right)^3\right]^3 \; se \cdots \left(\frac{|x - x_0|}{h}\right) < 1 \\ 0 \ldots se \ldots \left(\frac{|x - x_0|}{h}\right) \geq 1 \end{cases}$$

3) usando i pesi ottenuti con tale funzione $W_T(x)$ si calcola la media ponderata dei valori y_i per ottenere il valore stimato $\hat{y}_0 = \hat{f}(x_0) = \frac{\sum W_T(x_i) y_i}{\sum W_T(x_i)}$; un peso maggiore è attribuito ai valori prossimi a x_0

4) Si ripete la procedura per tutti i valori ordinati delle osservazioni $x_{(1)} \ldots . x_{(n)}$, si otterrà una serie di valori stimati $\hat{y}_1 \ldots \hat{y}_n$; unendo tali valori stimati si ottiene la stima della funzione di regressione non parametrica;

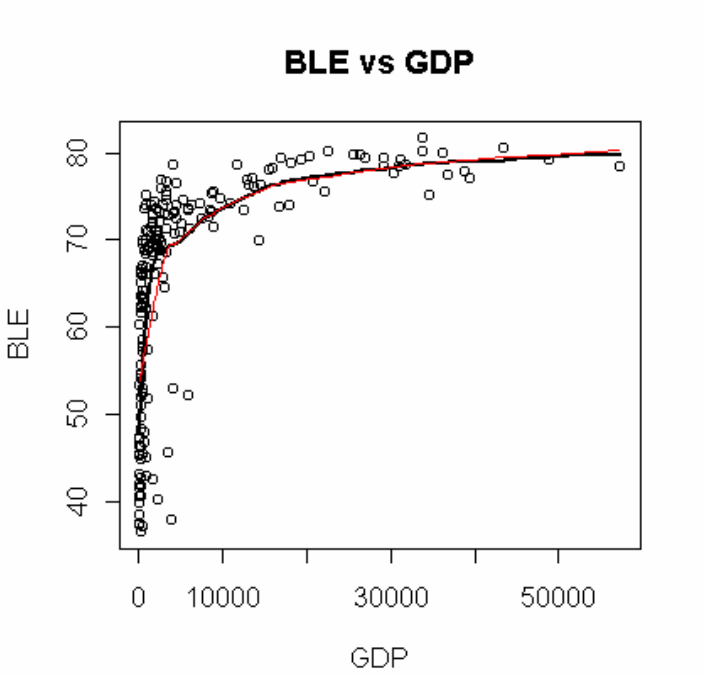
La *local polynomial regression* è simile alla precedente, ma i valori stimati sono ottenuti con una regressione locale ponderata, anziché con una media locale ponderata; $\hat{y}_0$ nel passo 3 è ottenuto con una regressione polinomiale di y su x stimata minimizzando la somma dei quadrati dei residui ponderati: $\sum W_T(x_i)(y_i - a - b_1 x_i - b_2 x_i^2 - \ldots b_k x_i^k)^2$ in genere si usa k=1 e si ha una regressione di tipo lineare.

Nell'ambiente `R` si possono usare le funzioni `lowess()` e `loess()`. Nella prima occorre specificare il vettore della variabile risposta (y) e quello del regressore (x) e il parametro di lisciamento (smoother span) f, questo parametro fornisce la proporzione dei punti che nel grafico influenzano il lisciamento (smooth) di ogni singolo valore. Per default `f=2/3`. Questa funzione usa la regressione locale lineare.
Come applicazione usiamo i dati relativi al reddito procapite (GDP) e alla speranza di vita alla nascita (BLE) di un insieme di paesi dell'ONU.

```
 dati
                            COUNTRY     GDP  BLE
1                           Burundi    86.4 41.6
2                          Ethiopia    91.1 42.0
3                Congo, Dem. Rep.   107.4 45.2
...
182                         Ireland 38864.2 77.7
183                         Denmark 39497.0 77.1
184                     Switzerland 43486.1 80.5
185                          Norway 48880.5 79.1
186                      Luxembourg 57379.3 78.3
```

```
attach(dati)
plot(BLE~GDP, main="BLE vs GDP")
lines(lowess(GDP, BLE, f=0.5, iter=0), lwd=2)
lines(lowess(GDP, BLE, f=0.75, iter=0), col="red")
```

Fig. 43



Abbiamo usato come parametro di span due valori 0,5 e 0,75; i risultati si vedono nella Fig. 43.
La funzione `loess()` è una evoluzione di `lowess()` e rispetto a questa è molto più raffinata.

Tra gli argomenti della funzione `loess()` da specificare abbiamo:
`formula`: il modello che mette in relazione la risposta e i regressori (da 1 a 4)
`data`: il data frame in cui sono contenute le variabili
`span`: il parametro di span che controllo il grado di lisciamento (smoothing)
`degree`: il grado del polinomio da usare nella regressione locale ponderata, fino a secondo grado
`family`: se è posto uguale a `"gaussian"` la stima è fatta con i minimi quadrati, se è posto uguale a `"symmetric"` è usato un M-estimator basato sulla Tukey's biweight function.

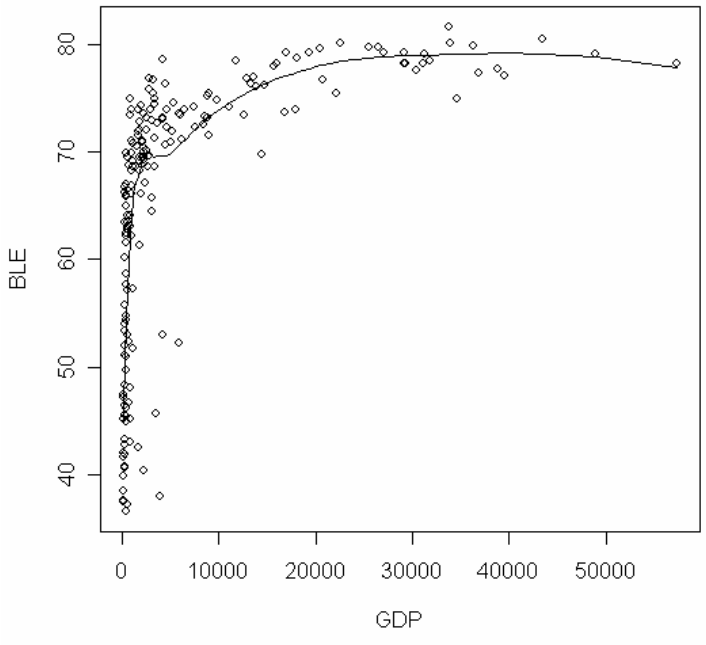
Procediamo con una esemplificazione pratica usando gli stessi dati impiegati precedentemente.

```
dati.lo<-loess(BLE~GDP, span=0.6, family="gaussian")

dati.lo
Call:
loess(formula = BLE ~ GDP, span = 0.6, family = "gaussian")

Number of Observations: 186
Equivalent Number of Parameters: 7.27
Residual Standard Error: 7.763

summary(dati.lo)
Call:
loess(formula = BLE ~ GDP, span = 0.6, family = "gaussian")
```

```
Number of Observations: 186
Equivalent Number of Parameters: 7.27
Residual Standard Error: 7.763
Trace of smoother matrix: 8.02

Control settings:
  normalize:  TRUE
  span      :  0.6
  degree    :  2
  family    :  gaussian
  surface   :  interpolate         cell = 0.2

 attributes(dati.lo)
$names
 [1]  "n"                "fitted"       "residuals"  "enp"             "s"
"one.delta"
 [7] "two.delta" "trace.hat" "divisor"   "pars"      "kd"        "call"
[13] "terms"     "xnames"    "x"         "y"         "weights"

$class
[1] "loess"

plot(GDP, BLE)
lines(dati.lo$x, dati.lo$fitted)
```

In questo modo abbiamo una visualizzazione grafica dei punti dello scatterplot e la funzione di regressione non parametrica stimata con il comando `loess()` (Fig. 44); i valori stimati della variabile dipendente (y) sono contenuti in `dati.lo$fitted`

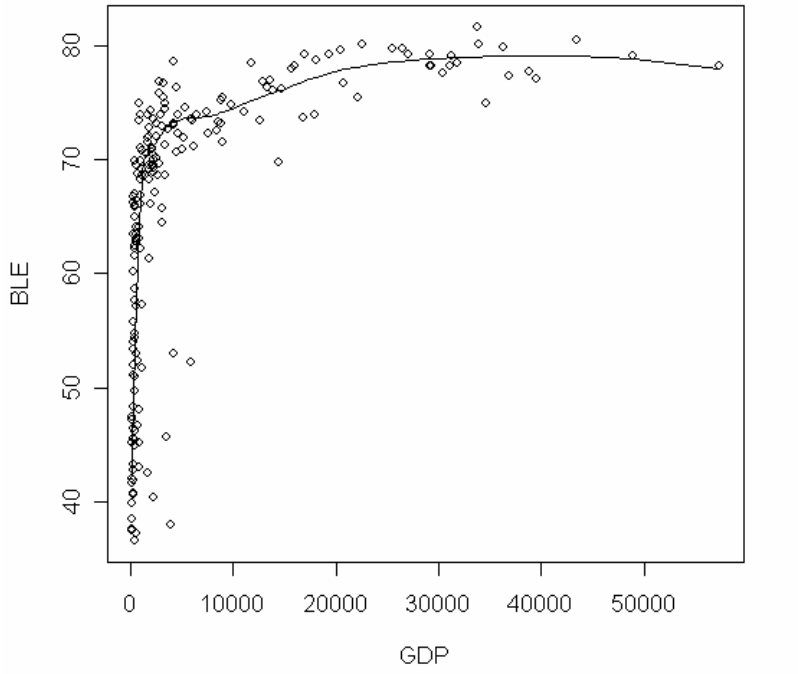
Fig.44




```
dati.lo2<-loess(BLE~GDP, span=0.6, family="symmetric")
Messaggio di avviso:
k-d tree limited by memory. ncmax= 200
summary(dati.lo2)
```

```
Call:
loess(formula = BLE ~ GDP, span = 0.6, family = "symmetric")

Number of Observations: 186
Equivalent Number of Parameters: 7.27
Residual Scale Estimate: 3.706
Trace of smoother matrix: 8.02

Control settings:
  normalize:  TRUE
  span      :  0.6
  degree    :  2
  family    :  symmetric     iterations = 4
  surface   :  interpolate         cell = 0.2

plot(GDP, BLE)
lines(dati.lo$x, dati.lo2$fitted)
```

Fig.45



13.0 Analisi della sopravvivenza e regressione di Cox

L'analisi della sopravvivenza[40] studia il tempo in cui un evento avviene, di solito l'evento esaminato è la morte, ma potrebbe essere la verifica di un guasto. Essa si occupa di esaminare i tempi di sopravvivenza osservati in dato ambito. Prima di trattare della regressione di Cox, introduciamo alcuni concetti basilari dell'analisi della sopravvivenza. Sia T il tempo di sopravvivenza, lo consideriamo come una variabile casuale con funzione cumulativa di probabilità $P(t) = \Pr(T < t)$ e con funzione di densità della probabilità

[40] Cfr. J. FOX, *Statistical...op. cit.*

$p(t) = \frac{dP(t)}{dt}$. Chiamiamo funzione di sopravvivenza $S(t) = 1 - P(t)$, mentre la funzione di rischio è data da $h(t) = \frac{p(t)}{S(t)}$, a seconda del tipo di funzione di rischio possiamo avere il modello esponenziale ($h(t) = \nu$), il modello di Gompertz ($\log h(t) = \nu + \rho t$) e quello di Weibull ($\log h(t) = \nu + \rho \log(t)$) nella distribuzione dei tempi di sopravvivenza.
Una caratteristica dei dati dell'analisi della sopravvivenza è che essi sono censurati (*censored*) a destra (*right censoring*) quando il periodo di osservazione degli individui termina o si perde un individuo nel corso dello studio, a sinistra (*left-censoring*), quando non è noto il tempo di inizio, o entrambe (*interval censored*). L'analisi della sopravvivenza studia la relazione tra i tempi di sopravvivenza e delle covariate. Se si considera un modello esponenziale abbiamo per la i.esima unità:

$$\log h_i(t) = \alpha + \beta_1 x_{i1} + ... + \beta_k x_{ik} \text{ ovvero } h_i(t) = \exp(\alpha + \beta_1 x_{i1} + ... + \beta_k x_{ik})$$

cioè il logaritmo del rischio è funzione lineare di k covariate, la costante α rappresenta una log-baseline del rischio, ossia il rischio che si ha quando tutte le covariate assumono valori uguali a zero. Nel modello di Cox si ha che α è funzione del tempo:

$$\log h_i(t) = \alpha(t) + \beta_1 x_{i1} + ... + \beta_k x_{ik} \text{ ovvero } h_i(t) = h_0(t)\exp(\beta_1 x_{i1} + ... + \beta_k x_{ik})$$

la baseline di rischio $h_0(t)$ può essere una qualsiasi funzione del tempo, mentre le covariate sono in relazione lineare. Il modello per tale motivo si dice semi-parametrico. Si può agevolmente dimostrare che il rapporto tra i rischi di due osservazioni non dipende da t: per questo il modello di Cox è detto a rischi proporzionali e i suoi parametri possono essere stimati con il metodo della massima verosimiglianza indipendentemente dalla forma della funzione di baseline di rischio.
Nell'ambiente `R` il package `survival` è dedicato all'analisi della sopravvivenza. Esamineremo alcuni comandi di questo package utili nel contesto della regressione. In particolare il comando è usato per creare oggetti di classe survival usati come variabili risposta nei modelli di regressione che esamineremo di seguito. Il comando può avere una duplice sintassi:

1) `Surv(time, event Surv()): time` (per dati censurati a destra) indica il tempo di follow up, mentre `event` è un indicatore di stato (0=vivo 1=morto);

```
library(survival)
with(aml, Surv(time, status))
 [1]   9   13   13+  18   23   28+  31   34   45+  48  161+   5    5    8
8
[16]  12   16+  23   27   30   33   43   45
```

2) `Surv(time, time2, event, type)`: (per dati censurati riferiti ad un intervallo) `time` è il tempo iniziale dell'intervallo, `time2` è il tempo finale dell'intervallo, `event` è un indicatore di stato (`0=right censored, 1= event at 'time', 2=left censored, 3=interval censored)`, `type` è una stringa che specifica il tipo di censura dei dati; i possibili valori sono `"right", "left", "counting", "interval", "interval2"`

```
     with(heart, Surv(start,stop,event))
  [1] (  0.0,  50.0 ] (  0.0,   6.0 ] (  0.0,   1.0+] (  1.0,  16.0 ]
  [5] (  0.0,  36.0+] ( 36.0,  39.0 ] (  0.0,  18.0 ] (  0.0,   3.0 ]
  [9] (  0.0,  51.0+] ( 51.0, 675.0 ] (  0.0,  40.0 ] (  0.0,  85.0 ]
 [13] (  0.0,  12.0+] ( 12.0,  58.0 ] (  0.0,  26.0+] ( 26.0, 153.0 ]
 ....
[165] (  0.0,  96.0+] ( 96.0, 109.0+] (  0.0,  21.0 ] (  0.0,  38.0+]
[169] ( 38.0,  39.0+] (  0.0,  31.0+] (  0.0,  11.0+] (  0.0,   6.0 ]
```

```
library(survival)
data(cancer)
```

La stima dei parametri di un modello di regressione di Cox avviene tramite il comando `coxph()` dalla sintassi praticamente uguale a quella del comando `lm()`:

```
coxfit<-coxph(Surv(time,status)~age+ sex+ ph.ecog+ ph.karno+ pat.karno+
meal.cal+ wt.loss, data=cancer)
summary(coxfit)
Call:
coxph(formula = Surv(time, status) ~ age + sex + ph.ecog + ph.karno +
    pat.karno + meal.cal + wt.loss, data = cancer)

  n=168 (60 observations deleted due to missing)
              coef exp(coef) se(coef)      z      p
age        1.06e-02     1.011 0.011611  0.917 0.3600
sex       -5.51e-01     0.576 0.200833 -2.743 0.0061
ph.ecog    7.34e-01     2.084 0.223271  3.288 0.0010
ph.karno   2.25e-02     1.023 0.011240  1.998 0.0460
pat.karno -1.24e-02     0.988 0.008054 -1.542 0.1200
meal.cal   3.33e-05     1.000 0.000259  0.128 0.9000
wt.loss   -1.43e-02     0.986 0.007771 -1.844 0.0650

          exp(coef) exp(-coef) lower .95 upper .95
age           1.011      0.989     0.988     1.034
sex           0.576      1.735     0.389     0.855
ph.ecog       2.084      0.480     1.345     3.228
ph.karno      1.023      0.978     1.000     1.045
pat.karno     0.988      1.012     0.972     1.003
meal.cal      1.000      1.000     1.000     1.001
wt.loss       0.986      1.014     0.971     1.001

Rsquare= 0.155   (max possible= 0.998 )
Likelihood ratio test= 28.3  on 7 df,   p=0.000192
Wald test            = 27.6  on 7 df,   p=0.000262
Score (logrank) test = 28.4  on 7 df,   p=0.000185
```

Il comando `survreg()`è utlizzato per stimare la regressione per un modello sopravvivenza di tipo parametrico; occorre specificare la formula -in modo simile al comando `lm()`-, si può specificare la distribuzione della variabile dipendente nel parametro dist ed eventualmente i valori dei parametri della distribuzione, mentre l'argomento scale (che per default è uguale a zero) serve per definire un valore fisso per il parametro di scala oppure se bisogna stimarlo:

```
fitsurv<-survreg(Surv(futime,   fustat)   ~   ecog.ps   +   rx,   ovarian,
dist='weibull',scale=1)
summary(fitsurv)

Call:
survreg(formula = Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian,
    dist = "weibull", scale = 1)
             Value Std. Error      z        p
(Intercept)  6.962      1.322  5.267 1.39e-07
ecog.ps     -0.433      0.587 -0.738 4.61e-01
rx           0.582      0.587  0.991 3.22e-01
```

```
Scale fixed at 1

Weibull distribution
Loglik(model)= -97.2   Loglik(intercept only)= -98
        Chisq= 1.67 on 2 degrees of freedom, p= 0.43
Number of Newton-Raphson Iterations: 4
n= 26

fitsurv2<-survreg(Surv(futime,   fustat)   ~   ecog.ps   +   rx,   ovarian,
dist='weibull')

summary(fitsurv2)

Call:
survreg(formula = Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian,
    dist = "weibull")
             Value Std. Error      z        p
(Intercept)  6.897      1.178  5.857 4.72e-09
ecog.ps     -0.385      0.527 -0.731 4.65e-01
rx           0.529      0.529  0.999 3.18e-01
Log(scale)  -0.123      0.252 -0.489 6.25e-01

Scale= 0.884

Weibull distribution
Loglik(model)= -97.1   Loglik(intercept only)= -98
        Chisq= 1.74 on 2 degrees of freedom, p= 0.42
Number of Newton-Raphson Iterations: 5
n= 26

survfit3<-survreg(Surv(futime,   fustat)   ~   ecog.ps   +   rx,   ovarian,
dist="exponential")
summary(survfit3)

Call:
survreg(formula = Surv(futime, fustat) ~ ecog.ps + rx, data = ovarian,
    dist = "exponential")
             Value Std. Error      z        p
(Intercept)  6.962      1.322  5.267 1.39e-07
ecog.ps     -0.433      0.587 -0.738 4.61e-01
rx           0.582      0.587  0.991 3.22e-01

Scale fixed at 1

Exponential distribution
Loglik(model)= -97.2   Loglik(intercept only)= -98
        Chisq= 1.67 on 2 degrees of freedom, p= 0.43
Number of Newton-Raphson Iterations: 4
n= 26
```

Il comando `survreg()` può essere utilizzato per stimare regressioni con dati censurati o troncati.

14.0 Regressione Tobit

Il modello Tobit è un modello econometrico proposto da James Tobin per descrivere la relazione esistente tra una variabile dipendente osservabile y_i - che non può assumere valori inferiori a zero - e una o più variabili

dipendenti x_i. Il modello suppone l'esistenza di una variabile latente non osservabile y_i^* che dipende linearmente dalle variabili x_i. Si introducono degli errori ε_i che comprendono gli effetti accidentali aventi distribuzione di tipo normale con media nulla e varianza costante. La variabile osservabile y_i è posta uguale alla variabile latente, se la variabile latente è maggiore di zero, ed è uguale a zero altrimenti:

$$y_i = \begin{cases} y_i^* & se\ y_i^* > 0 \\ 0 & se\ y_i^* \le 0 \end{cases}$$

con $y_i^* = \beta' x_i + \varepsilon_i \qquad i = 1...n$

Il vettore dei coefficienti di regressione β se viene stimato con i minimi quadrati ordinari (OLS) regredendo y_i su x_i fornisce delle stime distorte e non consistenti. Le stime corrette del modello di Tobin sono ottenute con il metodo della massima verosimiglianza. La regressione Tobit è uno caso particolare della regressione censurata, poiché la variabile latente non può essere sempre osservata.

In `R` la regressione Tobit può essere stimata usando il comando `survreg()` che si trova nel package `survival`. In questo comando occorre specificare il modello di regressione seguendo lo schema analogo a quello usato per l'analisi della sopravvivenza `(Surv(time, event, type))`, il dataframe che contiene i dati (argomento `data`), il tipo di distribuzione della variabile dipendente (argomento `dist`, si può scegliere tra `"weibull","exponential", "gaussian", "logistic", "lognormal"` e `"loglogistic"`). Procediamo con un esempio usando dei dati contenuti del dataframe tobin presente nel package `survival`:

```
library(survival)## viene richiamato il package

data(tobin)
tobin
   durable  age quant
1      0.0 57.7   236
2      0.7 50.9   283
3      0.0 48.5   207
4      0.0 41.7   220
5      0.0 47.7   238
6      0.0 59.8   216
7      0.0 44.3   284
8      3.7 45.1   221
9      0.0 51.7   275
10     3.0 50.0   269
11    10.4 46.8   207
12     0.0 58.0   249
13     0.0 58.9   246
14     0.0 40.0   277
15     1.5 34.1   231
16     0.0 39.9   219
17     0.0 33.4   240
18     3.5 48.1   266
19     6.1 46.6   214
20     0.0 53.1   251
```

In questo modello si pone in relazione l'acquisto di beni durevoli (*durable*) con l'età (*age*) e le disponibilità di denaro liquido (*quant*=liquidity ratio*1000); stimiamo i parametri del modello:

```
tobinfit<-survreg(Surv(durable,   durable>0,   type="left")   ~age+quant,
data=tobin, dist="gaussian")

tobinfit
Call:
survreg(formula = Surv(durable, durable > 0, type = "left") ~
    age + quant, data = tobin, dist = "gaussian")

Coefficients:
(Intercept)         age       quant
15.14486636 -0.12905928 -0.04554166

Scale= 5.57254

Loglik(model)= -28.9   Loglik(intercept only)= -29.5
        Chisq= 1.1 on 2 degrees of freedom, p= 0.58
n= 20

summary(tobinfit)

Call:
survreg(formula = Surv(durable, durable > 0, type = "left") ~
    age + quant, data = tobin, dist = "gaussian")
             Value Std. Error      z        p
(Intercept) 15.1449    16.0795  0.942 3.46e-01
age         -0.1291     0.2186 -0.590 5.55e-01
quant       -0.0455     0.0583 -0.782 4.34e-01
Log(scale)   1.7179     0.3103  5.536 3.10e-08

Scale= 5.57

Gaussian distribution
Loglik(model)= -28.9   Loglik(intercept only)= -29.5
        Chisq= 1.1 on 2 degrees of freedom, p= 0.58
Number of Newton-Raphson Iterations: 3
n= 20

attributes(tobinfit)
$names
 [1] "coefficients"      "icoef"             "var"
 [4] "loglik"            "iter"              "linear.predictors"
 [7] "df"                "scale"             "idf"
[10] "df.residual"       "terms"             "means"
[13] "call"              "dist"              "y"

$class
[1] "survreg"
```

Per curiosità calcoliamo le stime che si otterrebbero applicando il metodo OLS :

```
fit<-lm(durable~  age + quant, data = tobin)
fit

Call:
lm(formula = durable ~ age + quant, data = tobin)

Coefficients:
```

```
(Intercept)          age        quant
   11.07428     -0.02607     -0.03457

summary(fit)

Call:
lm(formula = durable ~ age + quant, data = tobin)

Residuals:
    Min      1Q  Median      3Q     Max
-2.6528 -1.6778 -0.8257  0.9137  7.7029

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 11.07428    6.79405   1.630    0.121
age         -0.02607    0.08174  -0.319    0.754
quant       -0.03457    0.02407  -1.437    0.169

Residual standard error: 2.709 on 17 degrees of freedom
Multiple R-Squared: 0.1169,     Adjusted R-squared: 0.01298
F-statistic: 1.125 on 2 and 17 DF,  p-value: 0.3477
```

15.0 Modelli lineari generalizzati (Generalized Linear Models GLM)

I modelli lineari generalizzati (GLM)[41] sono un'estensione dei modelli lineari, potendosi applicare nel caso di distribuzione della variabile risposta sia diversa da quella normale, nel caso di legame più complesso di quello lineare con le variabili dipendenti e nel caso di varianza dell'errore non costante.
La variabile risposta Y_i deve una distribuzione appartenente alla famiglia esponenziale:

$$f_Y(y;\theta,\phi) = \exp\{(y\theta - b(\theta))/a(\phi) + c(y,\phi)\}$$

dove le funzioni a(), b() e c() definiscono la distribuzione. Il parametro ϕ è detto *parametro di dispersione o di scala* e deve essere necessariamente noto in una distribuzione affinché questa appartenga alla famiglia esponenziale, mentre il parametro θ è detto *parametro canonico o naturale* della distribuzione. Alla famiglia esponenziale appartengono le principali distribuzioni: normale, gamma, poisson, binomiale, gaussiana inversa. Abbiamo che:

$$E(Y) = \mu = b'(\theta) \text{ e } V(Y) = b''(\theta)a(\phi)$$

Le caratteristiche delle principali distribuzioni della famiglia esponenziale sono riportate in Tabella 2:

[41] C. TRIVISANO, *Introduzione ai modelli lineari generalizzati*, novembre 2004

Tabella 2

	Normale	Poisson	Binomiale	Gamma	Gaussiana Inversa
Notazione	$N(\mu,\sigma^2)$	$P(\mu)$	$B(m,\pi)/m$	$G(\mu,\nu)$	$IG(\mu,\sigma^2)$
Campo di variazione di y	$(-\infty,+\infty)$	$(0(1)\infty))$	$\frac{0(1)m}{m}$	$(0,\infty)$	$(0,\infty)$
Parametro di dispersione ϕ	σ^2	1	$1/m$	ν^{-1}	σ^2
Parametro canonico	μ	$\log(\mu)$	$\pi/(1-\pi)$	$1/\mu$	$-\frac{1}{2\mu^2}$
$b(\theta)$	$\theta^2/2$	$\exp(\theta)$	$\log(1+e^\theta)$	$-\log(-\theta)$	$-(-2\theta)^{1/2}$
$c(y;\phi)$	$-\frac{1}{2}\left(\frac{y^2}{\phi}+\log(2\pi\phi)\right)$	$-\log(y!)$	$\log\binom{m}{my}$	$\nu\log(\nu y)-\log(y)-\log\Gamma(\nu)$	$-\frac{1}{2}\left(\log(2\pi\phi y^3)+\frac{1}{\phi y}\right.$
$\mu(\theta)$	θ	$\exp(\theta)$	$e^\theta/(1+e^\theta)$	$-1/\theta$	$(-2\theta)^{-1/2}$
Link canonico	identità	log	logit	reciproco	$1/\mu^2$
Funzione di Varianza	1	μ	$\mu/(1-\mu)$	μ^2	μ^3

Fonte: C. TRIVISANO, *Introduzione ai modelli lineari generalizzati*, novembre 2004

Un modello lineare generalizzato è individuato da queste caratteristiche:

1) la variabile risposta Y_i deve una distribuzione appartenente alla *famiglia esponenziale*;
2) le variabili indipendenti influiscono sulla risposta in modo lineare:

$$\eta = \sum_{j=1}^{p} \beta_j X_j$$

tale funzione prende il nome di *predittore lineare*

3) la media è funzione del predittore lineare:

$\mu = m(\eta)$ e $\eta = m^{-1}(\mu) = \ell(\mu)$ con m() funzione monotona e differenziabile; la funzione $\ell()$ è detta *funzione link* ed esplicita la relazione tra il predittore lineare e il valore atteso della distribuzione.

Per ciascuna distribuzione della famiglia esponenziale esiste una particolare funzione link per la quale si ha: $\theta = \mu$, tale funzione è detta *link canonico*.
I parametri dei GLM vengono stimati con il metodo della massima verosimiglianza (ML) che è quello usato da R. Con i GLM si possono trattare, tra gli altri, la regressione logistica (quando la variabile risposta è di tipo dicotomico e rinomale) e la regressione di Poisson (quando la risposta è una variabile di conteggio con distribuzione poissoniana).
La stima dei parametri di un GLM con R[42] viene effettuata con il comando glm()che funzione in maniera pressoché analoga al comando lm() usato per la stima nel caso di modelli lineari. Nel caso di glm()occorre specificare in più la famiglia della distribuzione e il tipo di link (per default è impostato il link canonico) tramite l'attributo family . Nella Tabella 3 per ciascuna famiglia sono indicate le funzioni link corrispondenti.

Tabella 3

Nome della famiglia	***Funzione link***
`binomial`	`logit,probit, log, cloglog`
`gaussian`	`identity, log, inverse`
`Gamma`	`identity,inverse, log`

[42] Si vedano: R DEVELOPMENT CORE TEAM, *An introduction op. cit*., pagg. 55 e segg.; P.M.E. ALTHAM, *Introduction to Generalized Linear Modelling in R, Statistical laboratory*, giugno 2005; V. M. R. MUGGEO, G. FERRARA, *Il linguaggio R: concetti introduttivi ed esempi*, settembre 2005, pagg. 53 e segg.; C. J. GEYER, *Generalized linaer models in R*, dicembre 2003

inverse.gaussian	1/mu^2, identity,inverse, log
poisson	identity,log, sqrt
Quasi	log,1/mu^2, sqrt

15.1 Regressione logistica e probit

Nel caso in cui la variabile risposta è di tipo dicotomico o binomiale si parla di regressione logistica, riconducibile ai GLM con distribuzione di $Y_i \sim bin(n,\pi_i)$ e funzione link di tipo logistico $\log\left(\frac{\pi_i}{1-\pi_i}\right)$. In altri termini si ha il seguente modello di regressione:

$$\log\left(\frac{\pi_i}{1-\pi_i}\right)=\beta_0+\sum_{j=1}^{p}\beta_j X_j$$

Procediamo con un'esemplificazione con R considerando il dataframe heart:

```
heart[1:5,]## sono visualizzate le prime 5 righe del dataframe
  row.names sbp tobacco  ldl adiposity famhist typea obesity alcohol age chd
1         1 160   12.00 5.73     23.11 Present    49   25.30   97.20  52   1
2         2 144    0.01 4.41     28.61  Absent    55   28.87    2.06  63   1
3         3 118    0.08 3.48     32.28 Present    52   29.14    3.81  46   0
4         4 170    7.50 6.41     38.03 Present    51   31.99   24.26  58   1
5         5 134   13.60 3.50     27.78 Present    60   25.99   57.34  49   1
```

la variabile risposta (dicotomica) è chd; stimiamo il GLM:

```
fml<-
glm(chd~sbp+tobacco+ldl+adiposity+famhist+typea+obesity+alcohol+age,data=
heart, family=binomial(link=logit))## link logit

fml

Call:  glm(formula = chd ~ sbp + tobacco + ldl + adiposity + famhist +
typea + obesity + alcohol + age, family = binomial(link = logit),
data = heart)

Coefficients:
   (Intercept)                 sbp             tobacco                 ldl
adiposity
    -6.1507209           0.0065040           0.0793764           0.1739239
0.0185866
famhistPresent               typea             obesity             alcohol
age
     0.9253704           0.0395950          -0.0629099           0.0001217
0.0452253

Degrees of Freedom: 461 Total (i.e. Null);  452 Residual
Null Deviance:      596.1
Residual Deviance: 472.1        AIC: 492.1

summary(fml)

Call:
glm(formula = chd ~ sbp + tobacco + ldl + adiposity + famhist +
    typea + obesity + alcohol + age, family = binomial(link = logit),
    data = heart)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7781  -0.8213  -0.4387   0.8889   2.5435

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -6.1507209  1.3082600  -4.701 2.58e-06 ***
sbp             0.0065040  0.0057304   1.135 0.256374
tobacco         0.0793764  0.0266028   2.984 0.002847 **
ldl             0.1739239  0.0596617   2.915 0.003555 **
adiposity       0.0185866  0.0292894   0.635 0.525700
famhistPresent  0.9253704  0.2278940   4.061 4.90e-05 ***
typea           0.0395950  0.0123202   3.214 0.001310 **
obesity        -0.0629099  0.0442477  -1.422 0.155095
alcohol         0.0001217  0.0044832   0.027 0.978350
age             0.0452253  0.0121298   3.728 0.000193 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 472.14  on 452  degrees of freedom
AIC: 492.14

Number of Fisher Scoring iterations: 5

coef(fml)
   (Intercept)            sbp        tobacco            ldl      adiposity
 -6.1507208650   0.0065040171   0.0793764457   0.1739238981   0.0185865682
famhistPresent          typea        obesity        alcohol            age
  0.9253704194   0.0395950250  -0.0629098693   0.0001216624   0.0452253496

anova(fml)
Analysis of Deviance Table

Model: binomial, link: logit

Response: chd

Terms added sequentially (first to last)


           Df Deviance Resid. Df Resid. Dev
NULL                         461     596.11
sbp         1    16.79       460     579.32
tobacco     1    33.46       459     545.86
ldl         1    20.12       458     525.74
adiposity   1     2.62       457     523.13
famhist     1    22.56       456     500.57
typea       1     6.24       455     494.33
obesity     1     7.68       454     486.64
alcohol     1     0.11       453     486.53
age         1    14.39       452     472.14
```

possiamo calcolare lo pseudo R^2 per verificare il grado di adattamento del modello stimato:

```
pseudoR2<-function(mod) {1-(deviance(mod)/mod$null.deviance)}
pseudoR2(fml)
[1] 0.2079628
```

l'indice è ottenuto calcolando il complemento ad 1 del rapporto tra la devianza del modello stimato e la devianza del modello con la sola intercetta.
Se con funzione link si utilizza la funzione di ripartizione della distribuzione normale, al posto della funzione logistica, si parla di regressione probit.

```
fmp<-glm(chd~sbp+tobacco+ldl+adiposity+famhist+typea+obesity+alcohol+age,
data=heart, family=binomial(link=probit))##link probit

summary(fmp)

Call:
glm(formula = chd ~ sbp + tobacco + ldl + adiposity + famhist +
    typea + obesity + alcohol + age, family = binomial(link = probit),
    data = heart)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7608  -0.8351  -0.4303   0.8881   2.6368

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)    -3.570e+00  7.518e-01  -4.749 2.04e-06 ***
sbp             3.789e-03  3.428e-03   1.105 0.268965
tobacco         4.822e-02  1.584e-02   3.044 0.002331 **
ldl             1.028e-01  3.529e-02   2.914 0.003569 **
adiposity       1.240e-02  1.738e-02   0.713 0.475756
famhistPresent  5.390e-01  1.348e-01   3.998 6.39e-05 ***
typea           2.356e-02  7.188e-03   3.277 0.001049 **
obesity        -4.016e-02  2.628e-02  -1.528 0.126518
alcohol         1.955e-05  2.686e-03   0.007 0.994193
age             2.627e-02  7.038e-03   3.733 0.000189 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 596.11  on 461  degrees of freedom
Residual deviance: 471.92  on 452  degrees of freedom
AIC: 491.92

Number of Fisher Scoring iterations: 5


anova(fmp)
Analysis of Deviance Table

Model: binomial, link: probit

Response: chd

Terms added sequentially (first to last)
```

```
          Df Deviance Resid. Df Resid. Dev
NULL                        461     596.11
sbp        1    16.69       460     579.42
tobacco    1    33.83       459     545.59
ldl        1    20.20       458     525.39
adiposity  1     2.73       457     522.66
famhist    1    22.24       456     500.42
typea      1     6.42       455     494.00
obesity    1     7.77       454     486.24
alcohol    1     0.09       453     486.15
age        1    14.22       452     471.92

pseudoR2(fm2)
[1] 0.2083251
```

come si può vedere i risultati della regressione logistica e di quella probit sono differenti.
Se si vuole applicare la procedura stepwise regression, procede in modo analogo ai modelli lineari:

```
step(fmp)
Start:  AIC= 491.92
 chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
    alcohol + age

           Df Deviance    AIC
- alcohol    1   471.92 489.92
- adiposity  1   472.44 490.44
- sbp        1   473.15 491.15
<none>           471.92 491.92
- obesity    1   474.38 492.38
- ldl        1   480.58 498.58
- tobacco    1   481.49 499.49
- typea      1   482.98 500.98
- age        1   486.15 504.15
- famhist    1   487.88 505.88

Step:  AIC= 489.92
 chd ~ sbp + tobacco + ldl + adiposity + famhist + typea + obesity +
    age

           Df Deviance    AIC
- adiposity  1   472.44 488.44
- sbp        1   473.17 489.17
<none>           471.92 489.92
- obesity    1   474.38 490.38
- ldl        1   480.67 496.67
- tobacco    1   481.84 497.84
- typea      1   482.99 498.99
- age        1   486.24 502.24
- famhist    1   487.97 503.97

Step:  AIC= 488.44
 chd ~ sbp + tobacco + ldl + famhist + typea + obesity + age

          Df Deviance    AIC
- sbp      1   473.81 487.81
<none>         472.44 488.44
```

```
- obesity  1   474.82 488.82
- tobacco  1   482.41 496.41
- ldl      1   482.57 496.57
- typea    1   483.17 497.17
- famhist  1   488.41 502.41
- age      1   495.26 509.26

Step:  AIC= 487.81
 chd ~ tobacco + ldl + famhist + typea + obesity + age

          Df Deviance    AIC
- obesity  1   475.76 487.76
<none>         473.81 487.81
- tobacco  1   484.03 496.03
- ldl      1   484.17 496.17
- typea    1   484.21 496.21
- famhist  1   489.60 501.60
- age      1   501.66 513.66

Step:  AIC= 487.76
 chd ~ tobacco + ldl + famhist + typea + age

          Df Deviance    AIC
<none>         475.76 487.76
- ldl      1   484.45 494.45
- typea    1   485.55 495.55
- tobacco  1   486.11 496.11
- famhist  1   491.29 501.29
- age      1   501.91 511.91

Call:  glm(formula = chd ~ tobacco + ldl + famhist + typea + age, family
= binomial(link = probit),      data = heart)

Coefficients:
   (Intercept)         tobacco             ldl  famhistPresent
typea
      -3.78143         0.04892         0.09533         0.52820
0.02198
           age
       0.02924

Degrees of Freedom: 461 Total (i.e. Null);  456 Residual
Null Deviance:      596.1
Residual Deviance: 475.8        AIC: 487.8
```

Per ottenere le stime della probabilità di successo $prob(Y_i = 1 \mid \mathbf{X_i})$ ottenute con i modelli:

```
fitted(fml)[1:15]## prime 15 osservazioni a titolo di esempio
         1          2          3          4          5          6          7
0.71218288 0.33101091 0.28095703 0.71712327 0.69306085 0.61766214 0.21910724
         8          9         10         11         12         13         14
0.63054843 0.14565061 0.61115330 0.65988886 0.70762412 0.04222893 0.02888889
        15
0.51048478

fitted(fmp)[1:15] ## prime 15 osservazioni a titolo di esempio
         1          2          3          4          5          6          7
```

```
0.70382840 0.33548377 0.28801821 0.70740934 0.68951107 0.61377492 0.23082604
         8          9         10         11         12         13         14
0.62040597 0.14749376 0.60160816 0.65528279 0.70716614 0.03324755 0.01945644
        15
0.51056925
```

Nei casi sopra esaminati la variabile risposta chd era di tipo dicotomico potendo assumere i valori 1 o 0. Il software R consente di esprimere la variabile risposta anche in un'altra forma, ossia come una matrice di due colonne con nella prima il numero dei successi e nella seconda il numero di insuccessi. Si veda il seguente esempio:

```
bliss
  disolphure total death
1       1.69    59     6
2       1.72    60    13
3       1.76    62    18
4       1.78    56    28
5       1.81    63    52
6       1.84    59    53
7       1.86    62    61
8       1.88    60    60
```

nel dataframe bliss si hanno in corrispondenza di diverse quantità di un insetticida (disolphure) il numero totale di insetti a cui si è somministrato il prodotto (total) e quelli che sono morti (death). Si vuole studiare la frazione di insetti morti al variare della quantità dell'insetticida.

```
attach(bliss)
Y<-cbind(death,total-death)
fml<-glm(Y~disolphure,  family=binomial(link=logit),data=bliss)  ##  link
logit
summary(fml)

Call:
glm(formula = Y ~ disolphure, family = binomial(link = logit),
    data = bliss)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8986  -0.5475   0.9842   1.3315   1.7179

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -60.103      5.164  -11.64   <2e-16 ***
disolphure    33.934      2.903   11.69   <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: 284.202  on 7  degrees of freedom
Residual deviance:  13.633  on 6  degrees of freedom
AIC: 43.831

Number of Fisher Scoring iterations: 4
```

```
fmp<-glm(Y~disolphure,  family=binomial(link=probit),data=bliss)  ##  link
probit
summary(fmp)

Call:
glm(formula = Y ~ disolphure, family = binomial(link = probit),
    data = bliss)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1365  -0.5795   0.9642   1.1333   1.4017

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -34.466      2.608  -13.21   <2e-16 ***
disolphure    19.471      1.465   13.29   <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: 284.202  on 7  degrees of freedom
Residual deviance:  12.641  on 6  degrees of freedom
AIC: 42.840

Number of Fisher Scoring iterations: 4
```

Se si vuole verificare il grado di adattamento dei modelli:

```
pseudoR2(fml)
[1] 0.9520293

pseudoR2(fmp)
[1] 0.9555195
```

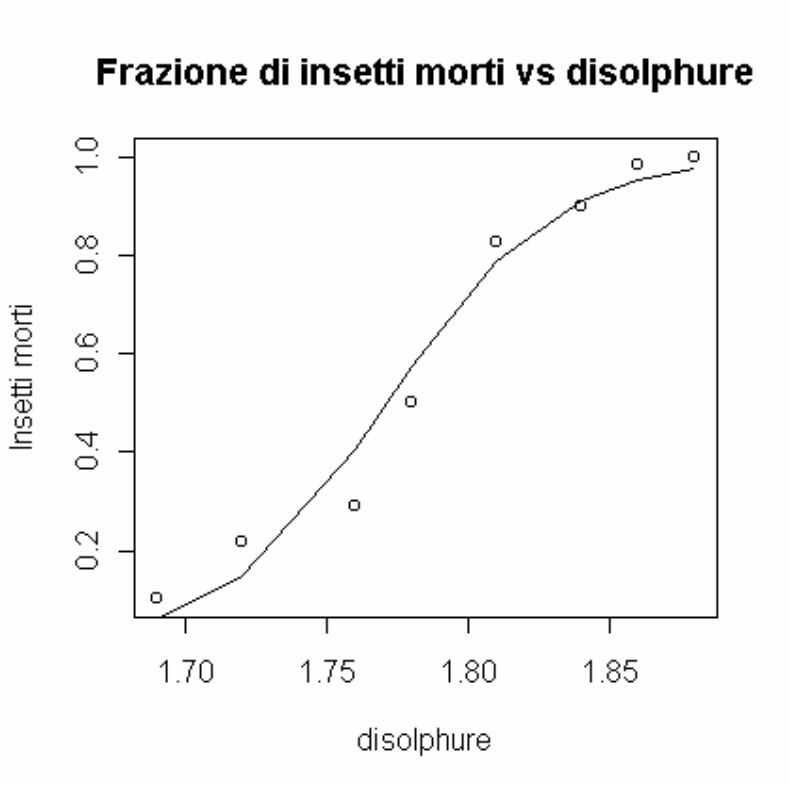
mentre per le probabilità stimate con i modelli:

```
fitted(fml)
         1          2          3          4          5          6          7
0.05983071 0.14975601 0.40633208 0.57432858 0.78877528 0.91178151 0.95321452
         8
0.97570644

fitted(fmp)
      1          2          3          4          5          6          7
0.05927022 0.16433122 0.42152721 0.57590319 0.78099010 0.91303108 0.95986061
         8
0.98376120
```

Volendo realizzare il grafico (Fig. 46) con la frazione di insetti morti in corrispondenza del livello di insetticida e la relativa stima della probabilità con il modello logit (linea continua):

```
plot(disolphure,death/total,   main="Frazione   di   insetti   morti   vs
disolphure",ylab=”Insetti morti”)
lines(disolphure,fitted(fml))
```

Fig. 46



è interessante calcolare il valore LD50 per i due modelli, ossia la quantità di insetticida in corrispondenza della quale muore il 50% degli insetti:

```
ld50<-function(mod) as.vector(-coef(mod)[1]/coef(mod)[2])

ld50(fml)
   1.771173

ld50(fmp)
   1.770169
```

15.2 Regressione di Poisson

Se la variabile risposta è una variabile di conteggio (e può assumere solo valori interi) e segue la distribuzione di Poisson $Y_i \sim Poiss(\lambda_i)$ con $P(Y_i = k) = \frac{\exp(-\lambda_i)\lambda_i^k}{k!}$ e si vuole studiare la seguente relazione:

$$E(Y_i \mid \mathbf{X}_i) = \exp\left(\beta_0 + \sum_{j=1}^{p} X_j \right)$$

ossia la dipendenza di Y_i da un insieme di variabili $\mathbf{X}_i$ (regressione di Poisson) si possono utilizzare i GLM. Vediamo un esempio in R. Si vuole verificare se la mortalità degli anziani in estate è influenzata dalla temperatura. Si prendono in considerazione i decessi di anziani in alcune città italiane nei mesi estivi e le relative temperature nel dataframe `decessi`:

```
decessi[1:7,]## alcune righe iniziali
```

```
    citta Tappmax d_over75 d_65_74 d_tot   mese
1  torino    23.9      384     104   574 giugno
2  torino    28.6      343     114   529 luglio
3  torino    28.4      339     110   511 agosto
4 brescia    26.2       75      28   112 giugno
5 brescia    29.6       72      25   109 luglio
6 brescia    29.4       77      23   120 agosto
7  milano    27.8      529     134   762 giugno
```

abbiamo i decessi di persone anziane distinti in due variabili (`d_over75` e `d_65_74`) in base all'età. Prenderemo in considerazione solo i decessi degli over 75 e utilizziamo `family=poisson` con sia `link=log` e che `link=sqrt`:

```
fmlog<- glm(formula = d_over75 ~ Tappmax, family = poisson(link =
log),data=decessi)

summary(fmlog)

Call:
glm(formula = d_over75 ~ Tappmax, family = poisson(link = log),
    data = decessi)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-19.031   -9.294   -4.556    1.818   31.850

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  5.51940    0.15956  34.593   <2e-16 ***
Tappmax      0.01342    0.00540   2.485   0.0129 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4417.0  on 23  degrees of freedom
Residual deviance: 4410.8  on 22  degrees of freedom
AIC: 4594.7

Number of Fisher Scoring iterations: 5

fmsqrt<- glm(formula = d_over75 ~ Tappmax, family = poisson(link =
sqrt),data=decessi)

summary(fmsqrt)

Call:
glm(formula = d_over75 ~ Tappmax, family = poisson(link = sqrt),
    data = decessi)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-19.034   -9.292   -4.556    1.821   31.844

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 15.47910    1.51512  10.216   <2e-16 ***
```

```
Tappmax      0.12799    0.05137   2.492   0.0127 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 4417.0  on 23  degrees of freedom
Residual deviance: 4410.8  on 22  degrees of freedom
AIC: 4594.8

Number of Fisher Scoring iterations: 4
```

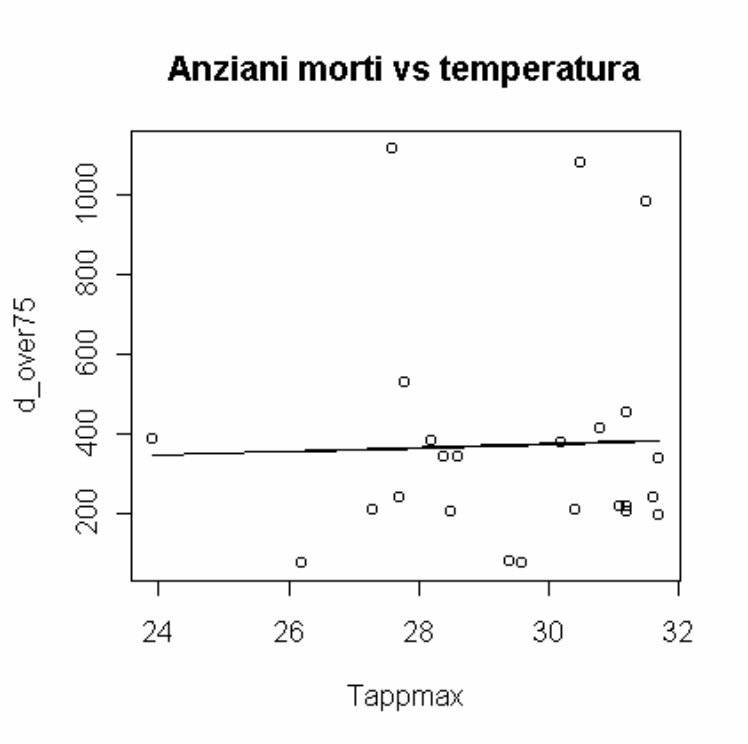
Come si può vedere anche se i valori numerici ottenuti con i due modelli sono diversi la conclusione a cui possiamo pervenire è che esiste un legame statisticamente significativo tra decessi di anziani over 75 e temperatura dei mesi estivi, anche se questo legame non è molto forte, come mostrato nel grafico in Fig. 47 (la linea continua è la relazione stimata con un modello).

```
attach(decessi)
plot(d_over75,Tappmax, main="Decessi anziani vs temperatura")
lines(Tappmax,fitted(fmlog))
```

In `R` il comando `glm()` può essere impiegato anche quando la distribuzione della variabile risposta è di tipo gaussiano (con `link=identity` si tratta di un semplice modello lineare), gaussiano inverso, Gamma. Quando si specifica family=quasi si tratta di *quasi-likelihood models*. Si riporta un esempio con distribuzione Gamma:

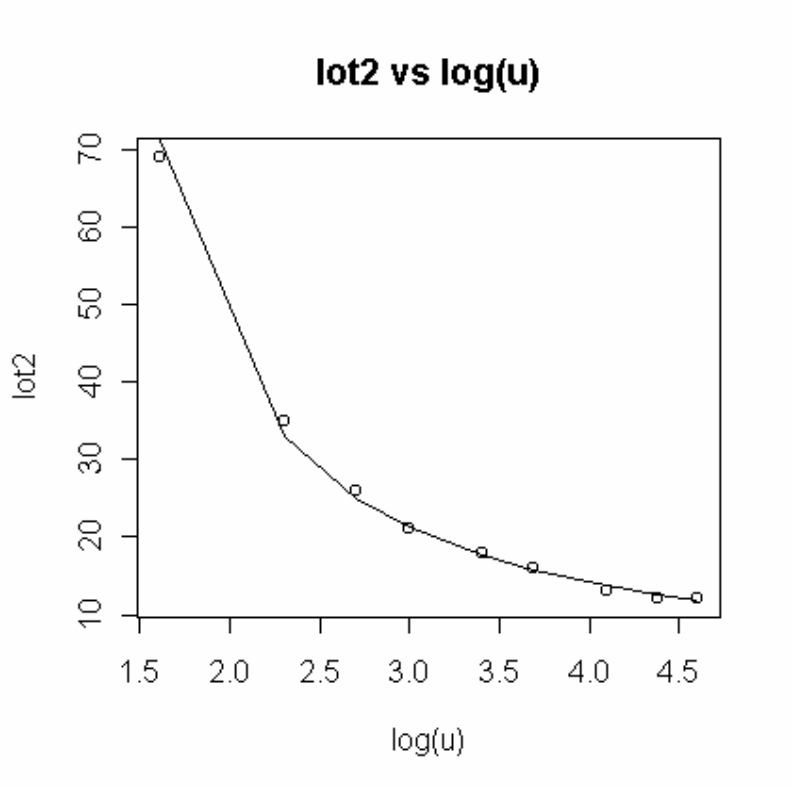
```
clotting <- data.frame(u = c(5,10,15,20,30,40,60,80,100),
         lot1 = c(118,58,42,35,27,25,21,19,18),
         lot2 = c(69,35,26,21,18,16,13,12,12))
fm1<-glm(lot1 ~ log(u), data=clotting, family=Gamma)
fm2<-glm(lot2 ~ log(u), data=clotting, family=Gamma)

summary(fm1)
Call:
glm(formula = lot1 ~ log(u), family = Gamma, data = clotting)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.04008  -0.03756  -0.02637   0.02905   0.08641

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0165544  0.0009275  -17.85 4.28e-07 ***
log(u)       0.0153431  0.0004150   36.98 2.75e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.002446013)

    Null deviance: 3.512826  on 8  degrees of freedom
Residual deviance: 0.016730  on 7  degrees of freedom
AIC: 37.99

Number of Fisher Scoring iterations: 3
```

Fig. 47




```
summary(fm2)
Call:
glm(formula = lot2 ~ log(u), family = Gamma, data = clotting)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-0.05574  -0.02925   0.01030   0.01714   0.06372

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.0239085  0.0013265  -18.02 4.00e-07 ***
log(u)       0.0235992  0.0005768   40.91 1.36e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Gamma family taken to be 0.001813340)

    Null deviance: 3.118557  on 8  degrees of freedom
Residual deviance: 0.012672  on 7  degrees of freedom
AIC: 27.032

Number of Fisher Scoring iterations: 3

attach(clotting)
plot(lot2 ~ log(u))
lines(log(u),fitted(fm2))
title("lot2 vs log(u)")
```

Fig. 48



16.0 Modelli multivel (mixed effect models)

I modelli di regressione multilevel[43], conosciuti anche come modelli gerarchici o anche *linear mixed effects models*, sono dei modelli di regressione che rispetto ai semplici modelli lineari prevedono dei termini *random effects* aggiuntivi e dovrebbero essere utilizzati, ad esempio, quando i dati presentano una struttura gerarchica o sono raggruppati in cluster e viene utilizzato un campionamento a due stadi che implica la dipendenza tra le osservazioni appartenenti allo stesso gruppo. Un tipico esempio è quello di voler esaminare delle caratteristiche degli studenti appartenenti a diverse classi di una stessa scuola o a diverse scuole. Nei modelli multilevel è possibile considerare variabili relative sia alle unità di base (studenti) sia ai gruppi (scuole o classi). In tali circostanze la dipendenza esistente tra le unità di primo livello (micro) appartenenti alla stessa unità di 2 livello (macro) è cruciale per l'analisi.
Diamo una breve trattazione teorica dei modelli multilevel adottando la forma compatta matriciale[44]:

$$\mathbf{y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\mathbf{b}_i + \boldsymbol{\varepsilon}_i$$
$$\mathbf{b}_i \sim \mathbf{N}_{\mathbf{q}}(\mathbf{0}, \boldsymbol{\Psi})$$
$$\boldsymbol{\varepsilon}_i \sim \mathbf{N}_{\mathbf{n_m}}(\mathbf{0}, \boldsymbol{\sigma}^m \boldsymbol{\Lambda}_i)$$

dove:

$\mathbf{y}_i$ è il vettore $n_i \times 1$ della variabile risposta delle osservazioni nel gruppo i.esimo;

$\mathbf{X}_i$ è la matrice $n_i \times p$ dei regressori degli effetti fissi per le osservazioni del gruppo i.esimo;

$\boldsymbol{\beta}$ è il vettore $p \times 1$ dei coefficienti degli effetti fissi, essi risultano identici per tutti i gruppi;

$\mathbf{Z}_i$ è la matrice $n_i \times q$ dei regressori degli effetti casuali per le osservazioni del gruppo i.esimo;

[43] Per una trattazione approfondita di tali modelli si rinvia a: L. RIZZI, *"I modelli multilevel" Aspetti teorici*, settembre 2002
[44] Si è fatti riferimento a: J. FOX, *An R op.cit.*

$\mathbf{b}_i$ è il vettore $q \times 1$ per i coefficienti degli effetti casuali per il gruppo i.esimo, tali coefficienti variano a seconda del gruppo;
$\boldsymbol{\varepsilon}_i$ è il vettore $n_i \times 1$ dei termini di errore per le osservazioni del gruppo i.esimo;
$\boldsymbol{\Psi}$ è la matrice $q \times q$ delle varianze e covarianze degli effetti casuali
$\boldsymbol{\sigma}^m \boldsymbol{\Lambda}_i$ è la matrice $n_i \times n_i$ delle varianze e covarianze degli errori nel gruppo i.esimo

la stima dei parametri del modello avviene con il metodo della massima verosimiglianza (ML) o della *Restricted Maximum Likelihood* (REML).
Nell'ambiente R il comando per la stima dei linear mixed effects models è lme()presente all'interno del package nlme. In questo comando occorre specificare la parte degli effetti fissi nell'argomento fixed, mentre quella degli e effetti casuali nell'argomento random, la specificazione avviene attraverso delle formule (si veda l'esempio). Procediamo con un esemplificazione pratica:

```
library(nlme)## per richiamare il package

data(MathAchieve)## per caricare il primo dataframe

MathAchieve[1:7,]## visualizza le prime 7 righe del dataframe
Grouped Data: MathAch ~ SES | School
  School Minority    Sex    SES MathAch MEANSES
1   1224       No Female -1.528   5.876  -0.428
2   1224       No Female -0.588  19.708  -0.428
3   1224       No   Male -0.528  20.349  -0.428
4   1224       No   Male -0.668   8.781  -0.428
5   1224       No   Male -0.158  17.898  -0.428
6   1224       No   Male  0.022   4.583  -0.428
7   1224       No Female -0.618  -2.832  -0.428

data(MathAchSchool)# per caricare il secondo dataframe

MathAchSchool[1:7,] # visualizza le prime 7 righe del dataframe
School Size   Sector PRACAD DISCLIM HIMINTY MEANSES
1224   1224  842   Public   0.35   1.597       0  -0.428
1288   1288 1855   Public   0.27   0.174       0   0.128
1296   1296 1719   Public   0.32  -0.137       1  -0.420
1308   1308  716 Catholic   0.96  -0.622       0   0.534
1317   1317  455 Catholic   0.95  -1.694       1   0.351
1358   1358 1430   Public   0.25   1.535       0  -0.014
1374   1374 2400   Public   0.50   2.016       0  -0.007
```

I dati si riferiscono ad un'indagine condotta su 7185 studenti di scuola superiore provenienti da 160 scuole. Il dataframe `MathAchieve` contiene i dati relativi degli studenti, mentre il dataframe `MathAchSchool` contiene i dati delle scuole. Esaminiamo le variabili di interesse:

`School`: identificativo della scuola;
`SES`: indicatore della condizione socioeconomica della famiglia dello studente (scarti rispetto alla media generale)
`MathAch`: punteggio dello studente nell'apprendimento della matematica;
`Sector`: variabile per distinguere le scuole pubbliche e quelle cattoliche.

Calcoliamo la media dei `SES` per scuola:

```
attach(MathAchieve)
mses <- tapply(SES, School, mean) # medie SES delle scuole
```

```
detach(MathAchieve)
```

si crea un nuovo dataframe con le variabili di interesse:

```
Bryk <- as.data.frame(MathAchieve[, c("School", "SES", "MathAch")])
names(Bryk) <- c("school", "ses", "mathach")# si assegnano nome con
lettere tutte minuscole alle variabili
```

si aggiungono due nuove variabili esterne: l'indicatore `cses` centrato sulla media della scuola che abbiamo calcolato e il settore (`sector`):

```
Bryk$meanses <- mses[as.character(Bryk$school)]
Bryk$cses <- Bryk$ses - Bryk$meanses
sector <- MathAchSchool$Sector
names(sector) <- row.names(MathAchSchool)
Bryk$sector <- sector[as.character(Bryk$school)]

sample10 <- sort(sample(7185, 10)) # campione casuale di 10 studenti

Bryk[sample10,]
     school    ses mathach     meanses        cses   sector
524    1499 -1.118   4.267 -0.46592453 -0.65207547   Public
1591   2818 -0.548   2.991  0.10866667 -0.65666667   Public
1678   2990  0.482  13.859  0.65470833 -0.17270833 Catholic
2945   4253 -0.578   7.424 -0.37368966 -0.20431034 Catholic
3693   5619  1.122  23.746  0.42033333  0.70166667 Catholic
5150   7232 -0.018  -0.955 -0.09011538  0.07211538   Public
5490   7364 -0.778  11.335 -0.08936364 -0.68863636 Catholic
6724   9104 -0.448  10.807  0.74345455 -1.19145455 Catholic
6746   9158  0.862   3.295 -0.40215094  1.26415094   Public
7008   9359  0.132  23.281  0.35407547 -0.22207547 Catholic
```

prima di stimare il modello mixed effects procediamo alla ricodifica della variabile sector con i valori 0 e 1:

```
Bryk$sector <- factor(Bryk$sector, levels=c('Public', 'Catholic'))
contrasts(Bryk$sector)
     Catholic
Public          0
Catholic        1
```

Stimiamo il modello con il comando `lme()`:

```
fm1<-lme(mathach ~ meanses*cses + sector*cses, random = ~ cses | school,
data=Bryk)
summary(fm1)
Linear mixed-effects model fit by REML
 Data: Bryk
       AIC      BIC    logLik
  46524.78 46593.57 -23252.39

Random effects:
 Formula: ~cses | school
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 1.54117685 (Intr)
cses        0.01817364 0.006
```

```
Residual    6.06349216

Fixed effects: mathach ~ meanses * cses + sector * cses
                        Value Std.Error   DF  t-value p-value
 Intercept)         12.128207 0.1991964 7022 60.88567   0e+00
meanses              5.336665 0.3689784  157 14.46335   0e+00
cses                 2.942145 0.1512240 7022 19.45554   0e+00
sectorCatholic       1.224531 0.3061139  157  4.00025   1e-04
meanses:cses         1.044406 0.2910747 7022  3.58810   3e-04
cses:sectorCatholic -1.642148 0.2331162 7022 -7.04433   0e+00
 Correlation:
                    (Intr) meanss cses   sctrCt mnss:c
meanses              0.256
cses                 0.000  0.000
sectorCatholic      -0.699 -0.356  0.000
meanses:cses         0.000  0.000  0.295  0.000
cses:sectorCatholic  0.000  0.000 -0.696  0.000 -0.351

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-3.17010624 -0.72487654  0.01489162  0.75426269  2.96549829

Number of Observations: 7185
Number of Groups: 160
```

Volendo stimare un altro modello, aggiornando il primo modello stimato, omettendo il random effect della variabile `cses`:

```
fm2 <- update(fm1, random =~1 |school)
anova(fm1, fm2)
Model df      AIC      BIC    logLik   Test     L.Ratio p-value
fm1     1 10 46524.78 46593.57 -23252.39
fm2     2  8 46520.79 46575.82 -23252.39 1 vs 2 0.003206865  0.9984
```

l'omissione della variabile `cses` nel random effect non produce un risultato significativo; proviamo ad eliminare l'intercetta casuale:

```
fm3<- update(fm1, random =~cses -1 |school)
anova(fm1, fm3)

Model df      AIC      BIC    logLik   Test  L.Ratio p-value
fm1     1 10 46524.78 46593.57 -23252.39
fm3     2  8 46740.23 46795.26 -23362.11 1 vs 2 219.4425  <.0001
```

il risultato in questa circostanza è significativo, eliminando l'intercetta casuale si perde informazione nel modello.

17.0 Generalized Additive Models (GAM)

I modelli GAM assumono che la media della variabile risposta è data dalla somma di termini ciascuno dipendente da un singolo predittore:

$$Y = \alpha + \sum_{i=1}^{p} f_i(x_i) + \varepsilon$$

dove le funzioni f_i non sono note a priori ma vanno stimate con tecniche di smoothing.

In R ci sono due comandi per stimare i modelli GAM uno nel package `mgcv` e l'altro nel package `gam`[45], in ambo i casi il comando si chiama `gam()`. Vediamo degli esempi pratici[46]. In primo luogo il package `gam` va scaricato dal CRAN e installato.

```
Iowa <- read.csv("Iowa.csv") ##carica il file Iowa.csv dalla directory corrente
Iowa[1:5,] # mostra prime 5 righe a titolo di esempio
  Year Rain0 Temp1 Rain1 Temp2 Rain2 Temp3 Rain3 Temp4 Yield
1 1930 17.75  60.2  5.83  69.0  1.49  77.9  2.42  74.4  34.0
2 1931 14.76  57.5  3.83  75.0  2.72  77.2  3.30  72.6  32.9
3 1932 27.99  62.3  5.17  72.0  3.12  75.8  7.10  72.2  43.0
4 1933 16.76  60.5  1.64  77.8  3.45  76.4  3.01  70.5  40.0
5 1934 11.36  69.5  3.49  77.2  3.85  79.7  2.84  73.4  23.0
```

Iniziamo con il stimare un semplice modello lineare considerando la variabile come risposta `Yield` e gli altri come regressori (si utilizza la stepwise regression per la selezione delle variabili significative):

```
require(MASS) # carica package MASS
iowa.lm1 <- lm(Yield ~ ., Iowa)
iowa.step <- stepAIC(iowa.lm1, scope = list(lower =  ~ Year, upper =  ~ .),
          k = log(nrow(Iowa)), trace = TRUE)
dropterm(iowa.step, test = "F", k = log(nrow(Iowa)),sorted = T)
Single term deletions

Model:
Yield ~ Year + Rain0 + Rain2 + Temp4
       Df Sum of Sq    RSS    AIC F Value     Pr(F)
<none>              1554.6  144.6
Temp4   1     188.0 1742.6  144.9     3.4   0.07641 .
Rain0   1     196.0 1750.6  145.0     3.5   0.07070 .
Rain2   1     240.2 1794.8  145.9     4.3   0.04680 *
Year    1    1796.2 3350.8  166.5    32.4 4.253e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

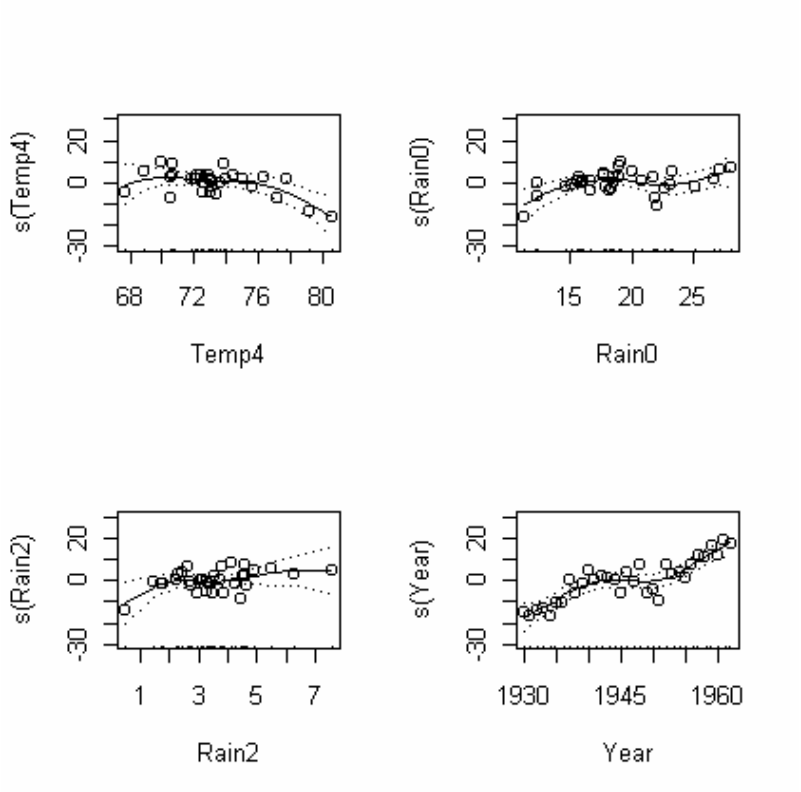
Procediamo con lo stimare il modello GAM usando il comando `gam()` nell' omonimo package:

```
require(gam)
iowa.gam <- gam(Yield ~ s(Temp4) + s(Rain0) + s(Rain2) + s(Year),data = Iowa)
par(mfrow = c(2,2))
plot(iowa.gam, se = T, ylim = c(-30, 30), resid = T)
summary(iowa.gam)
```

[45] http://dssm.unipa.it/CRAN/src/contrib/Descriptions/gam.html

[46] P. KUHNERT, B. VENABLES, *An Introduction to R: Software for Statistical Modelling & Computing*, 2005, pagg. 169 e segg.

Fig. 49




```
Call: gam(formula = Yield ~ s(Temp4) + s(Rain0) + s(Rain2) + s(Year),
    data = Iowa)
Deviance Residuals:
   Min     1Q Median     3Q    Max
-9.862 -2.174  0.314  2.438  7.775

(Dispersion Parameter for gaussian family taken to be 31.1181)

    Null Deviance: 5565.06 on 32 degrees of freedom
Residual Deviance: 497.8949 on 16.0002 degrees of freedom
AIC: 219.2077

Number of Local Scoring Iterations: 2

DF for Terms and F-values for Nonparametric Effects

            Df Npar Df Npar F   Pr(F)
(Intercept)  1
s(Temp4)     1       3 2.4707 0.09919 .
s(Rain0)     1       3 3.0300 0.05994 .
s(Rain2)     1       3 1.3742 0.28649
s(Year)      1       3 3.6838 0.03437 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nella Fig. 49 è riportato il grafico del modello GAM stimato. I predittori che influenzano maggiormente la variabile risposta risultano essere `Year` e, in misura in minore, `Temp4` e `Rain0`

In un altro esempio faccio riferimento al dataframe rock; la variabile risposta è il logaritmo della permeabilità della roccia (`perm`) per ricerche petrolifere e i predittori sono l'area, perimetro e forma (`area`, `peri`, `shape`), stimiamo in prima battuta il modello lineare classico:

```
data(rock) # carica il dataframe
rock.lm <- lm(log(perm) ~ area + peri + shape, data = rock)
summary(rock.lm)
Call:
lm(formula = log(perm) ~ area + peri + shape, data = rock)

Residuals:
    Min      1Q  Median      3Q     Max
-1.8092 -0.5413  0.1735  0.6493  1.4788

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.333e+00  5.487e-01   9.720 1.59e-12 ***
area         4.850e-04  8.657e-05   5.602 1.29e-06 ***
peri        -1.527e-03  1.770e-04  -8.623 5.24e-11 ***
shape        1.757e+00  1.756e+00   1.000    0.323
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8521 on 44 degrees of freedom
Multiple R-Squared: 0.7483,     Adjusted R-squared: 0.7311
F-statistic:  43.6 on 3 and 44 DF,  p-value: 3.094e-13
```

Stimiamo il modello GAM:

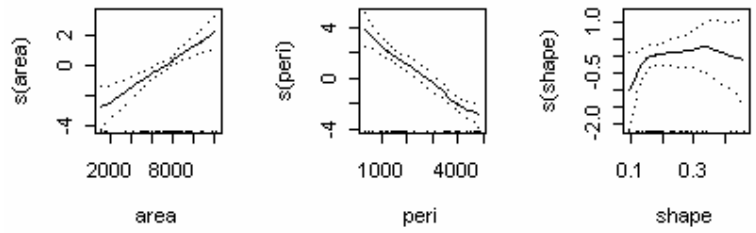
```
require(gam) # carica il package
rock.gam <- gam(log(perm) ~ s(area) + s(peri) + s(shape), control =
gam.control(maxit = 50, bf.maxit = 50), data = rock)

summary(rock.gam)
Call: gam(formula = log(perm) ~ s(area) + s(peri) + s(shape), data =
rock,
    control = gam.control(maxit = 50, bf.maxit = 50))
Deviance Residuals:
    Min      1Q  Median      3Q     Max
-1.6848 -0.4702  0.1235  0.5429  1.2955

(Dispersion Parameter for gaussian family taken to be 0.7445)

    Null Deviance: 126.9322 on 47 degrees of freedom
Residual Deviance: 26.0589 on 34.9997 degrees of freedom
AIC: 134.8984

Number of Local Scoring Iterations: 2

DF for Terms and F-values for Nonparametric Effects

            Df Npar Df  Npar F  Pr(F)
(Intercept)  1
s(area)      1       3 0.34169 0.7953
s(peri)      1       3 0.94069 0.4314
s(shape)     1       3 1.43206 0.2499
anova(rock.lm, rock.gam) # confronta il modello lineare e quello GAM
```

```
Analysis of Variance Table

Model 1: log(perm) ~ area + peri + shape
Model 2: log(perm) ~ s(area) + s(peri) + s(shape)
   Res.Df    RSS      Df Sum of Sq      F Pr(>F)
1 44.0000 31.949
2 34.9997 26.059  9.0003     5.890 0.8789 0.5528
```

l'aggiunta dei termini splines comporta un lieve miglioramento nella stima del modello, anche se non è statisticamente significativo;

```
par(mfrow = c(1, 3), pty = "s")
plot(rock.gam, se = TRUE,rug=TRUE)
```

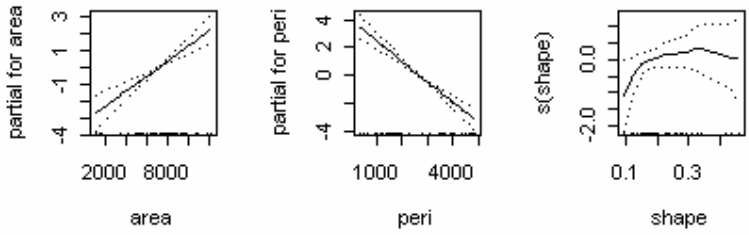
Fig. 50



la Fig. 50 ci suggerisce di inserire le variabili `area` e `peri` in forma lineare, mentre la variabile `shape` come termine splines, stimiamo questo modello:

```
rock.gam1 <- gam(log(perm) ~ area + peri + s(shape), data = rock)
plot(rock.gam1, se = TRUE,rug=TRUE)
par(mfrow=c(1,1))
```

Fig. 51



riconfrontando i modelli :

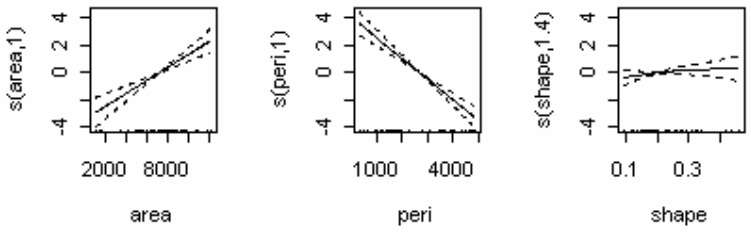
```
anova(rock.lm, rock.gam1, rock.gam)
Analysis of Variance Table

Model 1: log(perm) ~ area + peri + shape
Model 2: log(perm) ~ area + peri + s(shape)
Model 3: log(perm) ~ s(area) + s(peri) + s(shape)
   Res.Df    RSS      Df Sum of Sq      F Pr(>F)
1 44.0000 31.949
2 41.0000 28.999  3.0000     2.950 1.3205 0.2833
3 34.9997 26.059  6.0003     2.940 0.6582 0.6835
```

proviamo a stimare un ulteriore modello usando la funzione `gam()` del package `mgcv` (implementazione di Simon Wood):

```
require(mgcv)
rock.gamSW <- gam(log(perm) ~ s(area) + s(peri) + s(shape), data = rock)
summary(rock.gamSW)

Family: gaussian
Link function: identity

Formula:
log(perm) ~ s(area) + s(peri) + s(shape)

Parametric coefficients:
             Estimate  std. err.    t ratio    Pr(>|t|)
```

```
(Intercept)     5.1075     0.1222      41.81    < 2.22e-16

Approximate significance of smooth terms:
               edf       chi.sq     p-value
 s(area)         1       29.878     2.0928e-06
 s(peri)         1       72.664     7.7719e-11
s(shape)     1.402       9.3958     0.42197

R-sq.(adj) =  0.735   Deviance explained = 75.4%
GCV score = 0.78865   Scale est. = 0.71631   n = 48

par(mfrow = c(1, 3), pty = "s")
plot(rock.gamSW)
```

Fig. 52

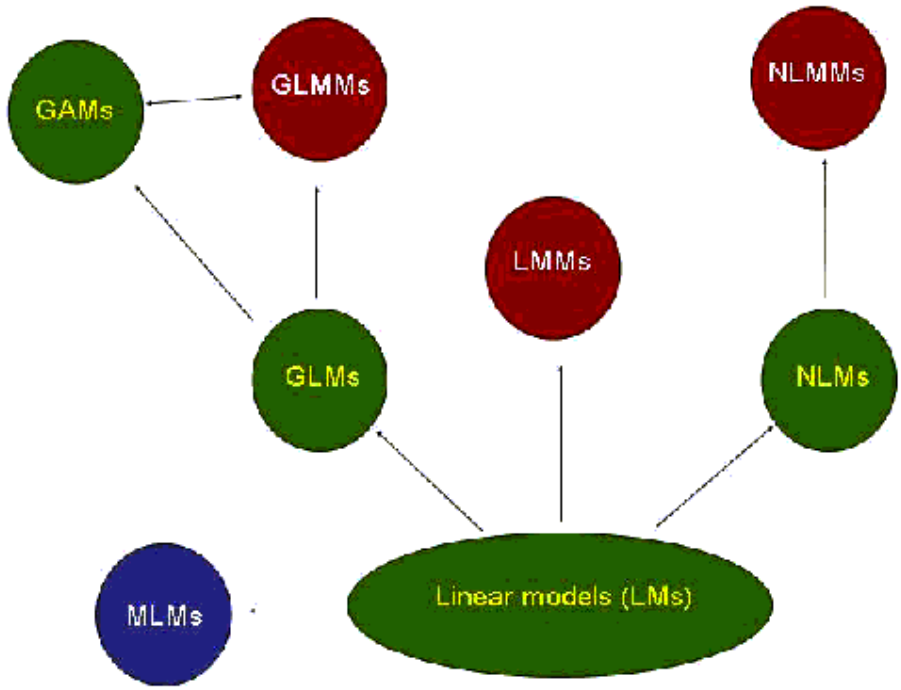


18.0 Conclusioni

Abbiamo esposto nei paragrafi precedenti l'utilizzo di R per la stima dei principali modelli di regressioni senza pretese di esaustività. Ci si è soffermati principalmente sul modello lineare classico, trattandone i principali aspetti, problematiche e tecniche.
Nella Fig. 53 è riportata una sorta di mappa dei modelli di regressione che comprende qualche modello che non è rientrato nella presente trattazione.

LM	Modelli lineari
MLM	Modelli lineari multivariati (MANOVA)
GLM	Modelli lineari generalizzati
LMM	Modelli lineari con effetti misti, modelli multilevel
NLM	Modelli non lineari
NLMM	Modelli non lineari con effetti misti, modelli multilevel
GLMM	Modelli lineari con effetti misti generalizzati
GAM	Modelli GAM (Generalized Additive Model)

Fig. 53



Fonte: P. KUHNERT, B. VENABLES, *An Introduction to R: Software for Statistical Modelling & Computing*, 2005

Riferimenti

Articoli e dispense

C. AGOSTINELLI, *Introduzione a R*, ottobre 2002
http://www.dst.unive.it/~claudio/R/manuale.0.3.pdf.zip

P.M.E. ALTHAM, *Introduction to Generalized Linear Modelling in R, Statistical laboratory*, giugno 2005
http://www.statslab.cam.ac.uk/~pat/redwsheets.pdf

B. ANDERSEN, *Generalized linear models*, marzo 2006
http://socserv.mcmaster.ca/andersen/soc740/10.GLM740.pdf

J. FARAWAY, *Practical Regression and Anova using R* , luglio 2002
http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf

J. FOX, *An R and S-PLUS Companion to Applied Regression*, 2002
http://cran.r-project.org/doc/contrib/Fox-Companion/appendix.html

J. FOX , *Nonparametric Regression*, Febbraio 2004
http://socserv.mcmaster.ca/jfox/Nonparametric-regression.pdf

J. FOX, *Statistical Applications in Social Research: Lecture Notes and R Scripts,* 2004,
http://socserv.socsci.mcmaster.ca/jfox/Courses/soc761/#lecture-notes

F. FRASCATI, *Formulario di Statistica con R*, novembre 2005
http://cran.r-project.org/doc/contrib/Frascati-FormularioR.pdf

C. J. GEYER, *Generalized linaer models in R*, dicembre 2003
http://www.stat.umn.edu/geyer/5931/mle/glm.pdf

R. KOENKER, K. HALLOCK, *Quantile Regression*, Journal of Economic Perspectives, 15, 2001, 143-156 http://www.econ.uiuc.edu/~roger/research/rq/QRJEP.pdf

R. KOENKER, *Quantile Regression in R: a vignette*
http://www.econ.uiuc.edu/~roger/research/rq/vig.pdf

P. KUHNERT, B. VENABLES, *An Introduction to R: Software for Statistical Modelling & Computing*, 2005 http://cran.r-project.org/doc/contrib/Kuhnert+Venables-R_Course_Notes.zip

J.MAINDONALD, *Using R for Data Analysis and Graphics - Introduction, Examples and Commentary*, Novembre 2004
http://cran.r-project.org/doc/contrib/usingR-2.pdf

G.M. MARCHETTI, *Dispense di Statistica 3*, 2003
http://www.ds.unifi.it/gmm/papers/master-stat3.pdf

R. MICCIOLO, *Dispense di Econometria ed applicazioni ai servizi sanitari*, 2004
http://www.economia.unitn.it/micciolo/eass/check.pdf

A. M. MINEO, *Una guida all'utilizzo dell'ambiente statistico R*, 2003
http://cran.r-project.org/doc/contrib/Mineo-dispensaR.pdf

V. M. R. MUGGEO, G. FERRARA, *Il linguaggio R: concetti introduttivi ed esempi*, settembre 2005
http://cran.r-project.org/doc/contrib/nozioniR.pdf

V. A. MUGGEO, *"Estimating regression models with unknown break-points"*, Stat Med. Oct 2003; 15; 22(19):3055-71).

A. POLLICE, *Dispense di statistica multivariata e Dispense su R,* gennaio 2005
http://www.dip-statistica.uniba.it/html/docenti/pollice/materiale.htm

R DEVELOPMENT CORE TEAM, An *introduction to R R. 2.3.1*., 1 giugno 2006
http://cran.r-project.org/doc/manuals/R-intro.pdf

V. RICCI, *Rappresentazione analitica delle distribuzioni statistiche con R*, febbraio 2005
http://cran.r-project.org/doc/contrib/Ricci-distributions-it.pdf

V. RICCI, *Regression reference card*, ottobre 2005
http://cran.r-project.org/doc/contrib/Ricci-refcard-regression.pdf

L. RIZZI, *"I modelli multilevel" Aspetti teorici*, settembre 2002
http://www.statistica.unimib.it/utenti/lovaglio/lucidi_rizzi.doc

L. SERLENGA, *Dispense di econometria*, a.a. 2003-04
http://www.dse.uniba.it/Corsi/docenti/Serlenga/lec2_04.pdf,

L. SOLIANI, *Statistica univariata e bivariata parametrica e non-parametrica per le discipline ambientali e biologiche*, 2005 http://www.dsa.unipr.it/soliani/soliani.html

C. TRIVISANO, *Introduzione ai modelli lineari generalizzati*, novembre 2004
http://www2.stat.unibo.it/cocchi/GLM.pdf

Libri

F. DEL VECCHIO, *Analisi statistica di dati multidimensionali*, 1992

F. DEL VECCHIO, *Statistica per la ricerca sociale*, 1992

J. J. FARAWAY, *Linear Models with R*, 2004
http://www.stat.lsa.umich.edu/~faraway/LMR/

J. FOX , *Applied Regression Analysis, Linear Models, and Related Methods*, 1997