Metodi e Problematiche Avanzate nella Regressione Laboratorio di Data Science in Economia - CLEBA

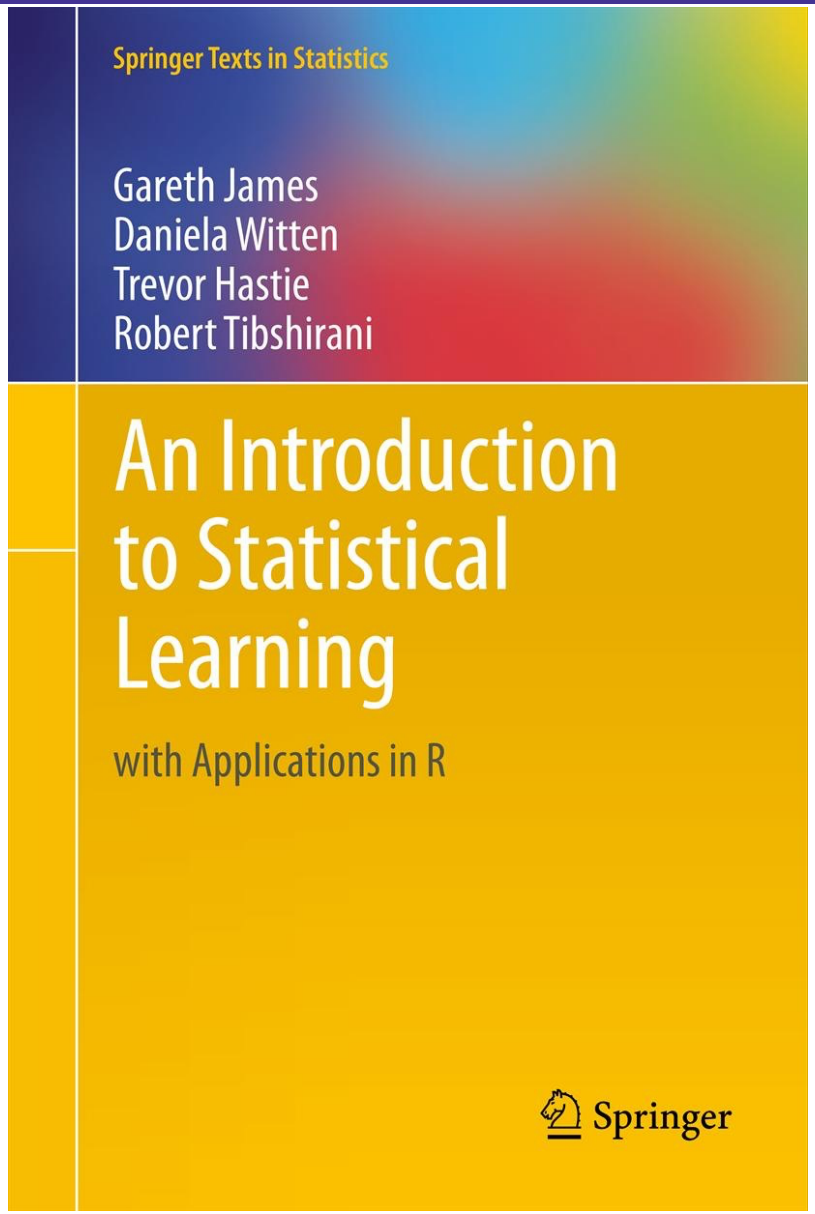
Roberto Benedetti

Dipartimento di Economia, email benedett@unich.it

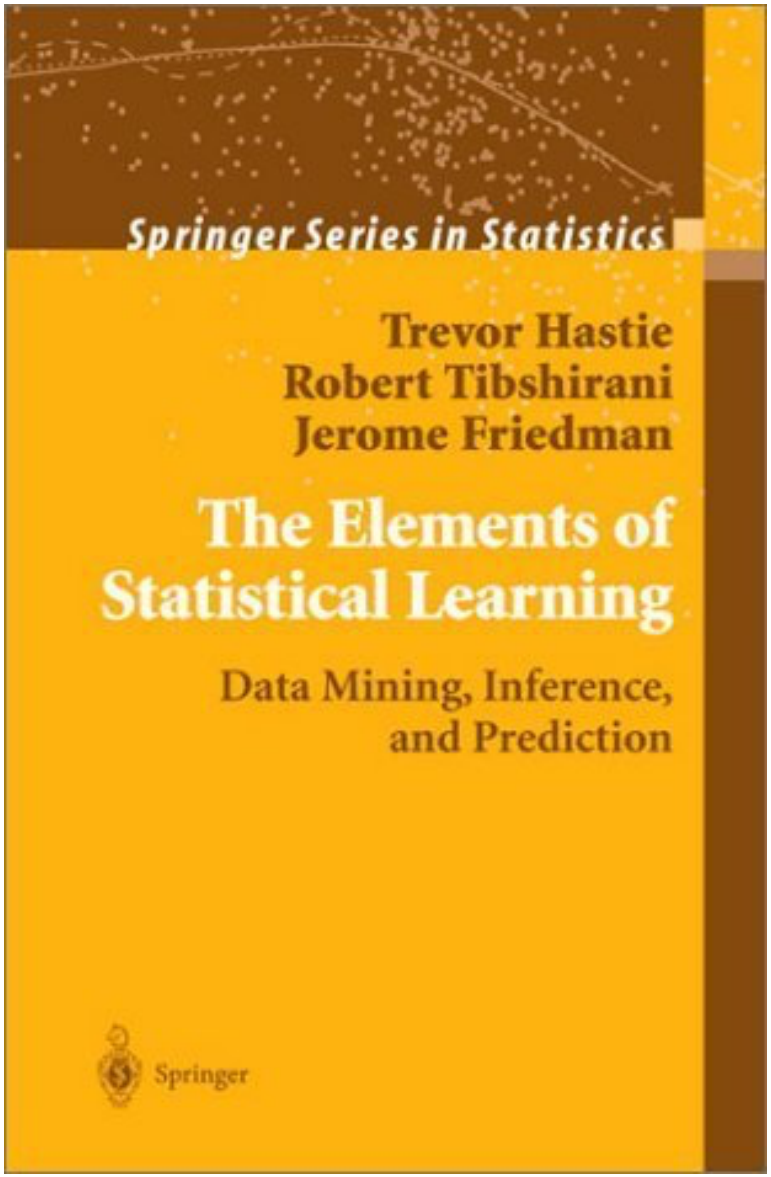
Argomanti trattati

- Dati anomali e regressione robusta
- Selezione delle covariate (Stepwise)
- Selezione di variabili basate sulle penalizzazione dei modelli di regressione (LASSO, Ridge)
- Approccio non parametrico alla regressione non lineare: regressione polinomiale locale
- Approccio non parametrico alla regressione non lineare: regressione spline
- Regressione robusta
- Regressione quantilica
- Misture finite di regressioni
- Trattamento dei dati mancanti (missing data)
- Applicazioni

Riferimento principale



Riferimento principale (versione avanzata)



Perché considerare alternative al classico metodo dei minimi quadrati?

- Accuratezza Predittiva: specialmente quando $p < n$, per avere un controllo migliore sulla varianza.
- Interpretabilità del modello: rimuovendo determinate caratteristiche è possibile ottenere un modello più facilmente interpretabile.

Per la selezione di caratteristiche da inserire nel modello, sono presenti tre approcci fondamentali:

Alcune alternative sulle X (var. esplicative, var. ausiliarie, covariate, predittori)

- **Subset Selection**: Si identifica un subset dei *p* predittori, che si presume relazionato alla variabile risposta. Successivamente si fitta il modello usando il metodo OLS sul set ridotto di variabili.
- **Shrinkage** (restringimento): Si fitta il modello costituito da tutti i *p* predittori, ma i coefficienti stimati vengono «contratti» verso lo zero relativamente alla stima OLS. Questo processo (chiamato *regolarizzazione*) ha l'effetto di ridurre la varianza e può anche effettuare una selezione di variabili.
- **Dimension Reduction**: Si proiettano i *p* predittori su un sottospazio M-dimensionale, dove M < p. Ciò viene ottenuto calcolando M differenti *combinazioni lineari, o proiezioni,* delle variabili. Le M proiezioni vengono poi utilizzate come predittori in un modello OLS di regressione lineare.

(Best) Subset Selection

1. Sia M_0 il *null model*, contenente zero predittori. Questo modello predice semplicemente la media campionaria per ogni osservazione.
2. Per *k = 1, 2, … p.*
   a) Si fittano tutti i $\binom{p}{k}$ modelli contenenti esattamente k predittori.
   b) Si prendono i migliori tra i $\binom{p}{k}$, chiamati M_k. Il modello è «migliore» se ha minore RSS, o equivalentemente, maggiore R^2.
3. Selezionare il singolo modello considerato migliore tra $M_0, \ldots, M_p$ usando misure di predizione dell'errore, C_p, AIC, BIC oppure l' *adjusted* R^2.

Stepwise Selection

Il metodo discusso in precedenza non può essere applicato per grandi valori di *p*. Perché?

Quando il numero di predittori *p* è grande (e di conseguenza lo spazio dimensionale è grande), c'è più alta probabilità di trovare modelli che appaiono buoni sui dati in possesso, ma potrebbero avere nessun potere predittivo sui dati futuri.

Per cui, con *p* elevato possono sorgere problemi di *overfitting* e di alta varianza dei coefficienti stimati.

Per queste ragioni sono proposti metodi «Stepwise» come alternativa al «best Subsect Selection», ed essi possono essere di due tipi: *Forward Stepwise e Backward Stepwise.*

Forward Stepwise Selection

Il metodo Forward stepwise si sostanzia nel partire col modello contenente zero predittori, e aggiungerne uno alla volta. In particolare, ad ogni step è aggiunta la variabile che fornisce il miglioramento addizionale maggiore al fit del modello. Quindi:

1. Sia M_0 il *null model*, contenente zero predittori.
2. Per k = 0, … , *p – 1:*
   - Considera tutti i *p-k* modelli che incrementano il numero di predittori in M_k con una variabile addizionale.
   - Scegli il migliore tra i *p-k* modelli, chiamato M_{k+1}. In questo caso «migliore» = minore RSS o più alto R^2.
3. Seleziona il singolo modello considerato migliore tra $M_0, \ldots, M_p$ usando misure di predizione dell'errore, C_p, AIC, BIC oppure l' *adjusted* R^2.

Backward Stepwise Selection

È anch'esso un metodo alternativo al «*best subsect selection*» ma, a differenza del Forward stepwise, esso inizia col modello completo contenente tutti i *p* predittori, e rimuove in maniera iterativa le variabili che meno contribuiscono al fit del modello, una per volta. Quindi:

1. Sia M_p il *modelo pieno*, contenente tutti i *p* predittori.
2. Per k = p, *p – 1, …, 1:*
   - Considera tutti i *k* modelli che contengono tutti i predittori meno uno in M_k, per un totale di k-1 predittori.
   - Scegli il migliore tra i *k* modelli, chiamato M_{k-1}. Anche in questo caso «migliore» = minore RSS o più alto R^2.
3. Seleziona il singolo modello considerato migliore tra $M_0, \ldots, M_p$ usando misure di predizione dell'errore, C_p, AIC, BIC oppure l'*adjusted* R^2.

Pro e contro di questi metodi

Il metodo «Forward» ha vantaggi computazionali rispetto al «best subset», ma non è garantita la possibilità di trovare il modello migliore tra tutti i 2^p modelli contenenti subset dei *p* predittori.

Il metodo «Backward» cerca tra tutti i $[1 + \frac{p(p+1)}{2}]$ modelli e può essere applicato quando *p* è troppo grande per applicare il *B.S.,* inoltre questo metodo richiede che la numerosità campionaria *n* sia maggiore del numero di variabili *p*, cosicchè possa essere creato il modello «pieno».

Contrariamente, il metodo forward può essere applicato anche quando $n < p$ per cui è l'unico metodo applicabile quando *p* è molto grande.

C_p, AIC, BIC, *adjusted* R^2

- *Mallow's* C_p*:*

$$C_p = \frac{1}{n}(RSS + 2d\hat{\sigma}^2)$$

Dove *d* è il numero totale dei parametri usati e $\hat{\sigma}^2$ è una stima della varianza dell'errore ε associato ad ogni misurazione.

- Il criterio *AIC* è definito con la massima verosimiglianza:

$$AIC = -2logL + 2d$$

Dove *L* è il valore massimizzato della funzione di verosimiglianza del modello stimato.

In caso di modelli lineari con errori normali, massima verosimiglianza e minimi quadrati sono equivalenti, per cui C_p e AIC sono equivalenti.

C_p, AIC, BIC, *adjusted* R^2

$$BIC = \frac{1}{n}(RSS + \log(n)d\hat{\sigma}^2)$$

Dove *n* è il numero delle osservazioni. Come C_p, il *BIC* tenderà ad un piccolo valore per modelli con errore contenuto, quindi si seleziona il modello col più piccolo valore.

- *Adjusted* R^2: per un modello OLS con *d* variabili, l' R^2 *«aggiustato» è:*

$$adjusted\ R^2 = 1 - \frac{RSS/(n-d-1)}{TSS/(n-1)}$$

Dove RSS è la devianza dei residui e TSS è la devianza totale.

Massimizzare l'adjusted R^2 vuol dire minimizzare $\frac{RSS}{n-d-1}$.

Mentre RSS decresce al crescere del numero di variabili nel modello, $\frac{RSS}{n-d-1}$ può crescere o decrescere a causa di *d* nel denominatore; l'adjusted R^2 quindi «paga il prezzo» dell'inclusione di variabili non necessarie nel modello.

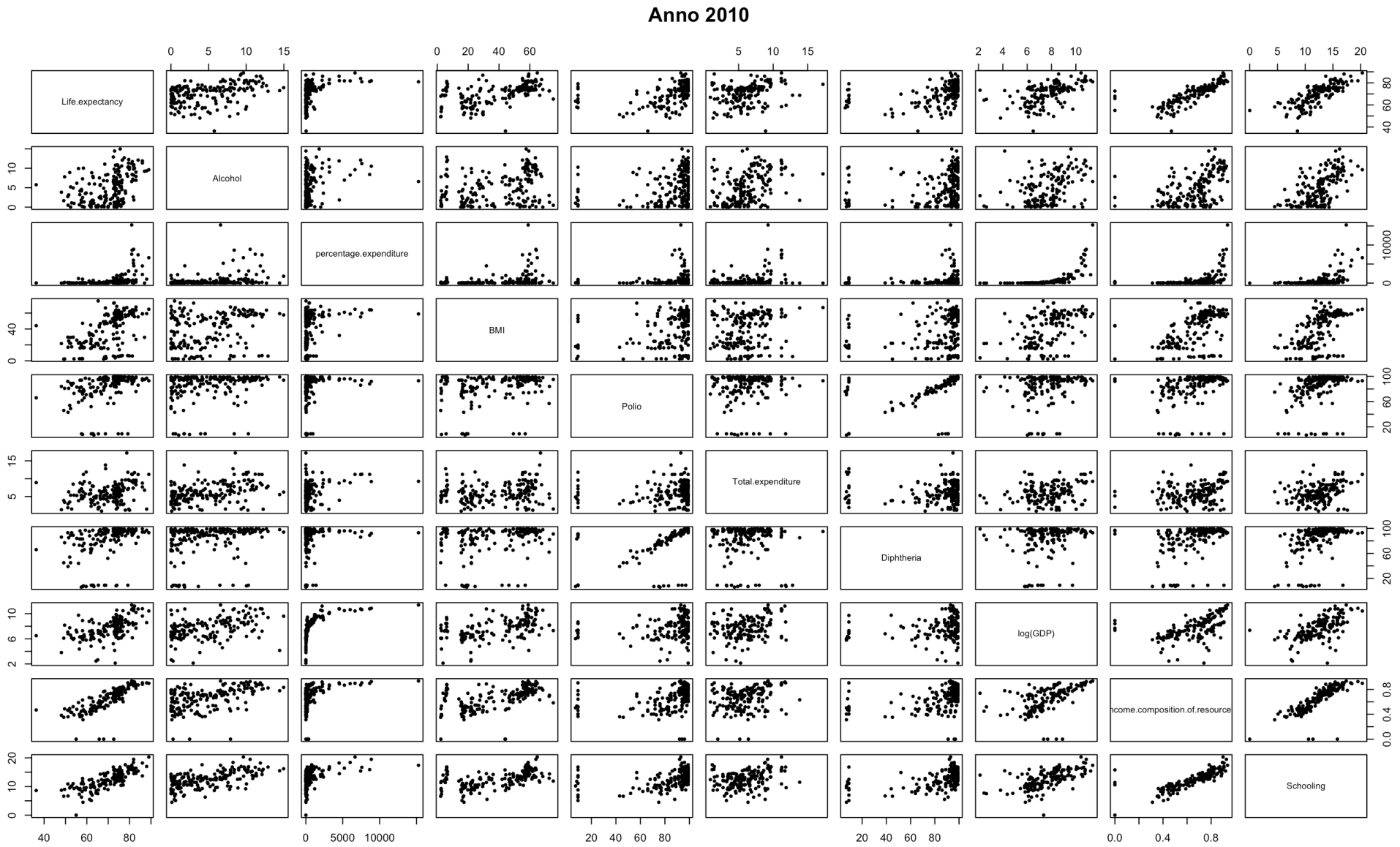
Esempio: aspettative di vita e reddito (curva di Preston)

La curva di Preston è una relazione empirica trasversale tra l'aspettativa di vita e il reddito pro capite reale. Prende il nome da Samuel H. Preston che lo descrisse per la prima volta nel 1975. Preston studiò la relazione per il 1900, gli anni '30 e gli anni '60 e ritenne che ben si adattava ai dati di ciascuno dei tre decenni.

La curva di Preston indica che le persone nate nei paesi più ricchi, in media, possono aspettarsi di vivere più a lungo di quelle nate nei paesi poveri. Tuttavia, il legame tra reddito e aspettativa di vita si appiattisce, quindi ci sono rendimenti decrescenti del reddito in termini di aspettativa di vita.

Un'ulteriore conclusione importante dello studio di Preston è che la curva si è spostata verso l'alto nel corso del 20-esimo secolo. Ciò significa che l'aspettativa di vita è aumentata nella maggior parte dei paesi, indipendentemente dalle variazioni del reddito. Secondo Preston, gli aumenti indipendenti dell'aspettativa di vita sono stati maggiori nei paesi poveri. Diversi paesi poveri dell'Africa subsahariana hanno effettivamente assistito a una riduzione dell'aspettativa di vita negli anni '90 e 2000 a seguito dell'epidemia di HIV / AIDS, anche se i loro redditi pro capite sono aumentati durante questo periodo.

Esempio: aspettative di vita e reddito (curva di Preston)



Esempio: aspettative di vita e reddito (curva di Preston)

```
ledat2010 <- ledat[ledat$Year==2010,]
ledat2010 <- ledat2010[complete.cases(ledat2010),]
regfull <- lm(Life.expectancy ~
Alcohol+percentage.expenditure+BMI+Polio+Total.expenditure+Diphtheria+log(GDP)+Incom
e.composition.of.resources+Schooling, data = ledat2010)
regst   <- stepAIC(regfull, direction = "both",na.rm=T)
summary(regst)
```

Modello completo

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	33.6418803	3.2141021	10.467	< 2e-16	***
Alcohol	-0.5312124	0.1511370	-3.515	0.000625	***
percentage.expenditure	0.0001838	0.0003746	0.490	0.624703	
BMI	0.0312346	0.0265243	1.178	0.241332	
Polio	0.0320164	0.0265747	1.205	0.230703	
Total.expenditure	0.4003349	0.2016707	1.985	0.049454	*
Diphtheria	0.0097459	0.0247083	0.394	0.693969	
log(GDP)	0.4028168	0.3681084	1.094	0.276057	
Income.composition.of.resources	27.4085164	4.6976785	5.834	4.83e-08	***
Schooling	0.8670350	0.3330132	2.604	0.010409	*

Multiple R-squared: 0.7261, Adjusted R-squared: 0.7052

Esempio: aspettative di vita e reddito (curva di Preston)

```
Start:  AIC=410.58
Life.expectancy ~ Alcohol + percentage.expenditure + BMI + Polio +Total.expenditure
+ Diphtheria + log(GDP) + Income.composition.of.resources + Schooling
                                  Df Sum of Sq    RSS    AIC
- Diphtheria                       1      3.57 2710.1 408.75
- percentage.expenditure           1      5.52 2712.1 408.84
- log(GDP)                         1     27.47 2734.0 409.87
- BMI                              1     31.81 2738.3 410.07
- Polio                            1     33.29 2739.8 410.14
<none>                                       2706.5 410.58
- Total.expenditure                1     90.38 2796.9 412.78
- Schooling                        1    155.48 2862.0 415.73
- Alcohol                          1    283.35 2989.9 421.32
- Income.composition.of.resources  1    780.80 3487.3 441.02
Step:  AIC=408.75
Step:  AIC=407.01
Step:  AIC=406.47
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                      32.00293    2.64302  12.108  < 2e-16 ***
Alcohol                          -0.55179    0.14880  -3.708 0.000316 ***
Polio                             0.03870    0.02211   1.750 0.082641 .
Total.expenditure                 0.41746    0.19012   2.196 0.030021 *
log(GDP)                          0.56228    0.32333   1.739 0.084571 .
Income.composition.of.resources  28.69270    4.55343   6.301 4.97e-09 ***
Schooling                         0.96277    0.31921   3.016 0.003121 **
Multiple R-squared:  0.722,       Adjusted R-squared:  0.7082
```

Shrinkage

I metodi di subset selection utilizzano il metodo dei minimi quadrati per fittare un modello lineare contenente un subset di variabili esplicative (predittori).

Vi sono però, anche metodi alternativi che utilizzano tutti i *p* utilizzando tecniche che «vincolano» o «regolarizzano» i coefficienti delle stime, o meglio, li «contraggono» (dall'inglese *to shrink*) verso lo zero.

Due metodi fondamentali che utilizzano questo approccio sono:

- La regressione Ridge;
- Il Lasso.

Regressione Ridge

Mentre il metodo dei minimi quadrati, nella procedura di stima di $\beta_0, \beta_1, \ldots, \beta_p$ utilizza il valore che minimizza:

$$RSS = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2$$

I coefficienti stimati $\hat{\beta}^R$ col metodo della regressione Ridge, sono i valori che minimizzano:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}\beta_j^2 = RSS + \lambda \sum_{j=1}^{p}\beta_j^2$$

Dove $\lambda \geq 0$ è un parametro detto «*tuning parameter*», determinato separatamente.

Regressione Ridge

- Analogamente al metodo dei minimi quadrati, la regressione ridge ricerca coefficienti che meglio approssimano i dati, rendendo la devianza di regressione (RSS) il più piccola possibile.
- Tuttavia, il secondo termine $\lambda \sum_{j=1}^{p} \beta_j^2$, chiamato *«shrinkage penalty»,* è piccolo quando β_1,…, β_p sono prossime allo zero, per cui ha l'effetto di contrarre le stime di β_j verso lo zero.
- L'obiettivo primario di tale costrizione verso lo zero è quello di ridurre la varianza della stima dei coefficienti.
- Il parametro λ serve per controllare l'impatto relativo di questi due termini sui coefficienti di regressione stimati, per cui la sua scelta è fondamentale e vengono usati metodi di cross-validation per effettuarla.

Scaling dei predittori nella regressione Ridge

I coefficienti stimati col metodo dei minimi quadrati hanno la proprietà di «invarianza di scala».

Vale a dire che, moltiplicando le X_j per una costante *c* produce una variazione di scala dei coefficienti stimati, che verranno moltiplicati per un fattore *1/c*.

In altre parole, tenendo conto di come sia scalato il j-esimo predittore, $X_j\beta_j$ non avrà cambiamenti.

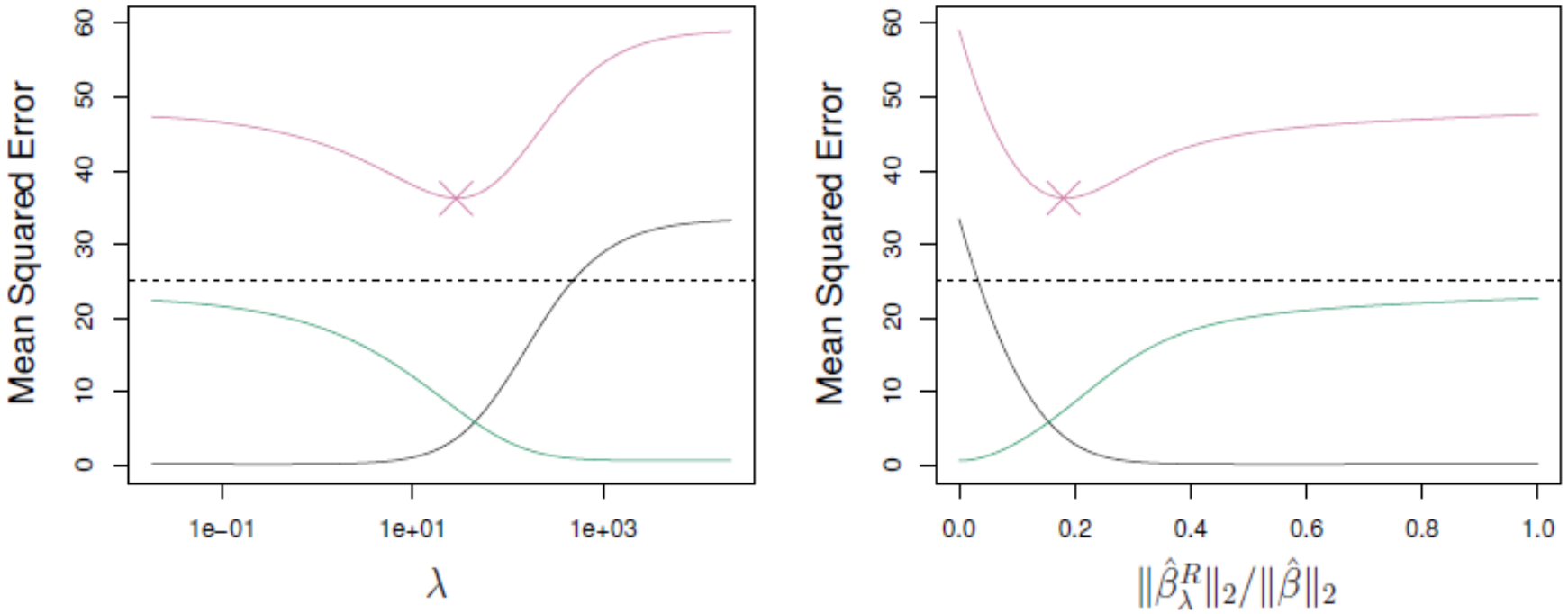
Scaling dei predittori nella regressione Ridge

Contrariamente, i coefficienti stimati con la regressione Ridge cambiano in maniera sostanziale se un dato predittore viene moltiplicato per una costante, a causa della somma di coefficienti al quadrato nella parte denominata «penalty».

Quindi, la cosa migliore da fare è applicare la regressione Ridge dopo aver standardizzato i predittori, usando la formula

$$x_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{ij} - x_j)^2}}$$

Scaling dei predittori nella regressione Ridge



I grafici riguardano una simulazione di dati con *n=50* osservazioni, *p=45* predittori, tutti con coefficient diversi da zero. Il bias (nero), la varianza (verde) e il mean square error (viola) per la regression ridge, come funzione di e del rapport tra il coefficient simato con la regression ridge, e quello stimato col metodo dei minimi quadrati. I punti in orizzontale indicano il MSE minimo possibile mentre le croci indicano il modello Ridge per il quale il MSE è minore.

Il Lasso

La regressione Ridge ha un solo svantaggio: a differenza del subset selection, che generalmente seleziona modelli contenenti solo un subset delle variabili, la regressione Ridge le include tutte nel modello.

Il metodo *Lasso* è relativamente recente e supera questo svantaggio. I coefficienti stimati con questo metodo, minimizzano:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j| =$$

$$RSS + \lambda \sum_{j=1}^{p} |\beta_j|.$$

Questo metodo usa un *penalty* denominato « l_1 », che sostituisce la forma al quadrato con i valori di β_j in valore assoluto.

Il Lasso

Come la regressione Ridge, il lasso contrae le stime dei coefficienti verso lo zero,

La differenza sostanziale è che il penalty l_1 ha l'effetto di «forzare» alcune stime ad essere esattamente uguali a zero, quando il parametro λ è sufficientemente grande.

Quindi, in maniera migliore del subset selection, il Lasso attua una selezione di variabili e fornisce un modello cosiddetto «sparso», il quale presenta solo una parte dei predittori.

Proprietà del metodo Lasso

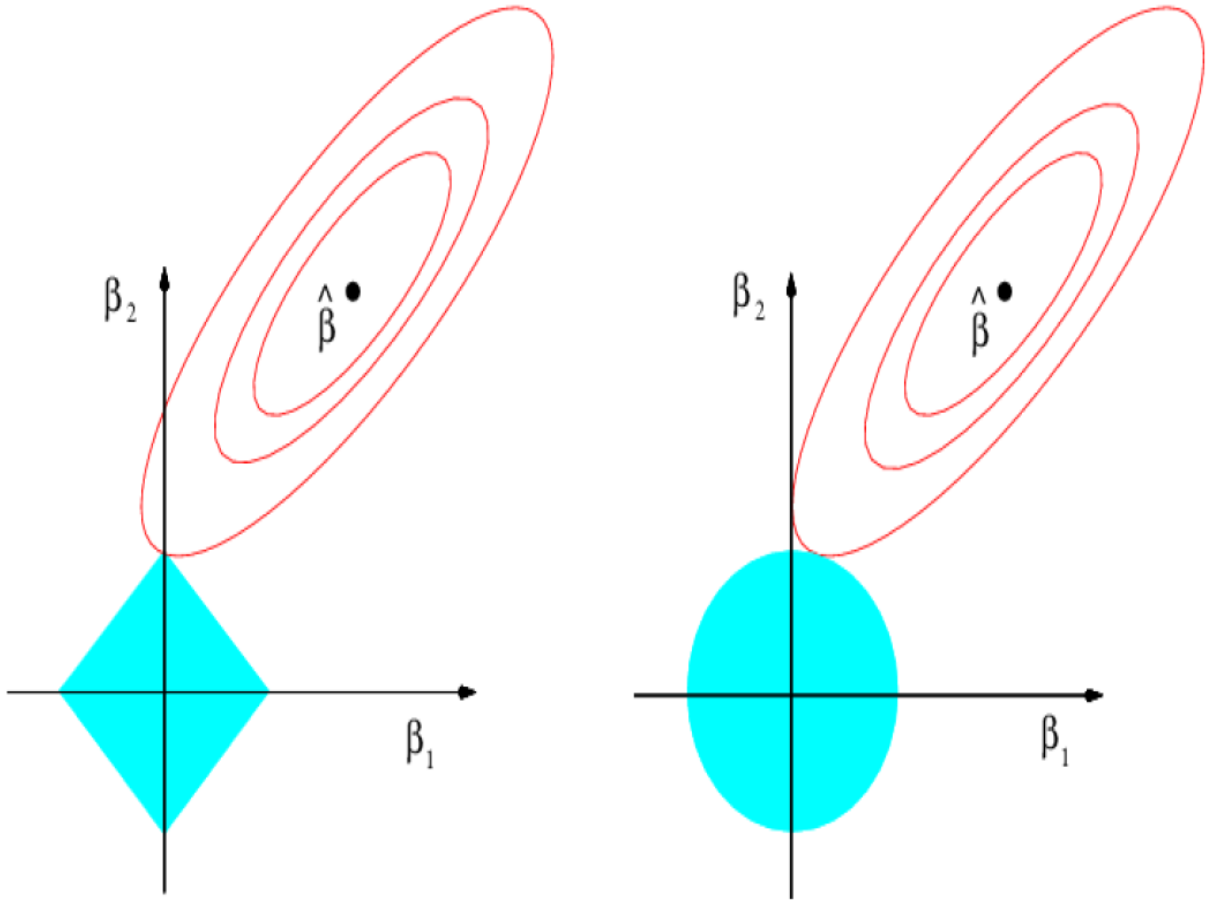
Perché il Lasso, a differenza della regressione Ridge, fornisce coefficienti esattamente uguali a zero? La differenza è nella minimizzazione di:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 \text{ soggetta al vincolo } \sum_{j=1}^{p}|\beta_j| \leq s$$

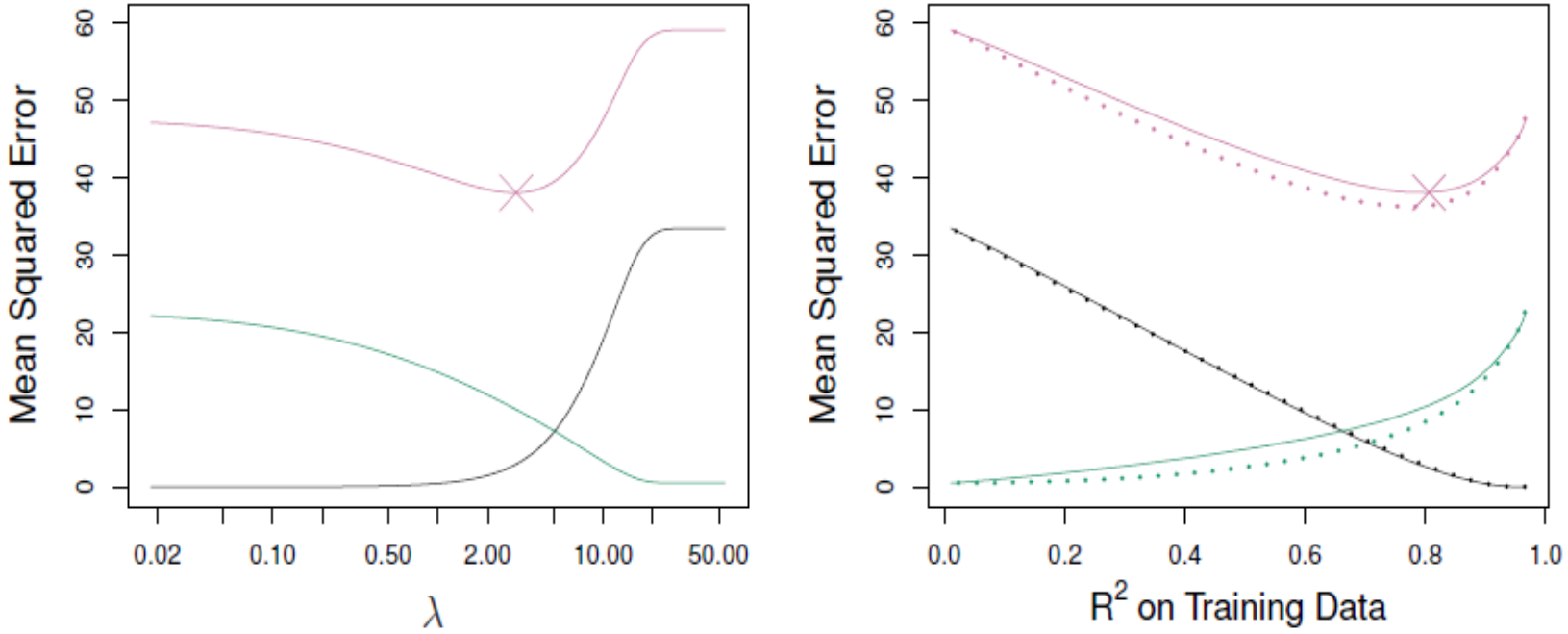
A differenza della minimizzazione (nella Ridge) di:

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 \text{ soggetta al vincolo } \sum_{j=1}^{p}\beta_j^2 \leq s$$

Differenze nella stima



Differenze nella stima



A sinistra sono plottati I valori di bias (nero), varianza (verde) e MSE test (viola) per il Lasso su valori del dataset di simulazione visto in precedenza.

A destra invece c'è la comparazione di bias, varianza e MSE tra Lasso (linea continua) e Ridge (tratteggiata). Entrambi sono plottati sull' del modello come forma di indicizzazione. Le croci sui grafici indicano il modello lasso per il cui il Mean Square Error è il più piccolo possibile.

Selezione del parametro λ per due tipi di regressione

- Come per il subset selection, per la Ridge e il Lasso c'è bisogno di un metodo per determinare quale modello considerato, è il migliore.
- C'è bisogno quindi di un metodo per selezionare il valore del parametro λ o equivalentemente, della restrizione *s.*
- La cross-validation fornisce un modo semplice per questo problema: si sceglie una griglia di valori di λ e si computano gli errori per ogni valore, si seleziona poi il parametro per il quale l'errore è minore.
- Infine, il modello è fittato di nuovo usando tutte le osservazioni disponibili ed il valore del parametro selezionato.

Alcune considerazioni sulle X

- Non c'è un metodo che prevale universalmente sull'altro;
- In generale, ci si può aspettare che il Lasso funzioni meglio quando la variabile risposta è funzione solo di un numero relativamente contenuto di predittori;
- Tuttavia, nei datasets reali il numero di predittori relazionato alla variabile risposta non è mai noto *a priori*;
- Una tecnica come la cross-validation può essere usata per determinare l'approccio che più si adatta ad un particolare dataset.

Esempio: relazioni extraconiugali

```
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  5.87201    1.13750   5.162 3.34e-07 ***
sexmale      0.05409    0.30049   0.180   0.8572
age         -0.05098    0.02262  -2.254   0.0246 *
childyes    -0.14262    0.35020  -0.407   0.6840
ym           0.16947    0.04122   4.111 4.50e-05 ***
religious   -0.47761    0.11173  -4.275 2.23e-05 ***
education   -0.01375    0.06414  -0.214   0.8303
occupation   0.10492    0.08888   1.180   0.2383
rate        -0.71188    0.12001  -5.932 5.09e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.095 on 592 degrees of freedom
Multiple R-squared:  0.1317,  Adjusted R-squared:   0.12
```

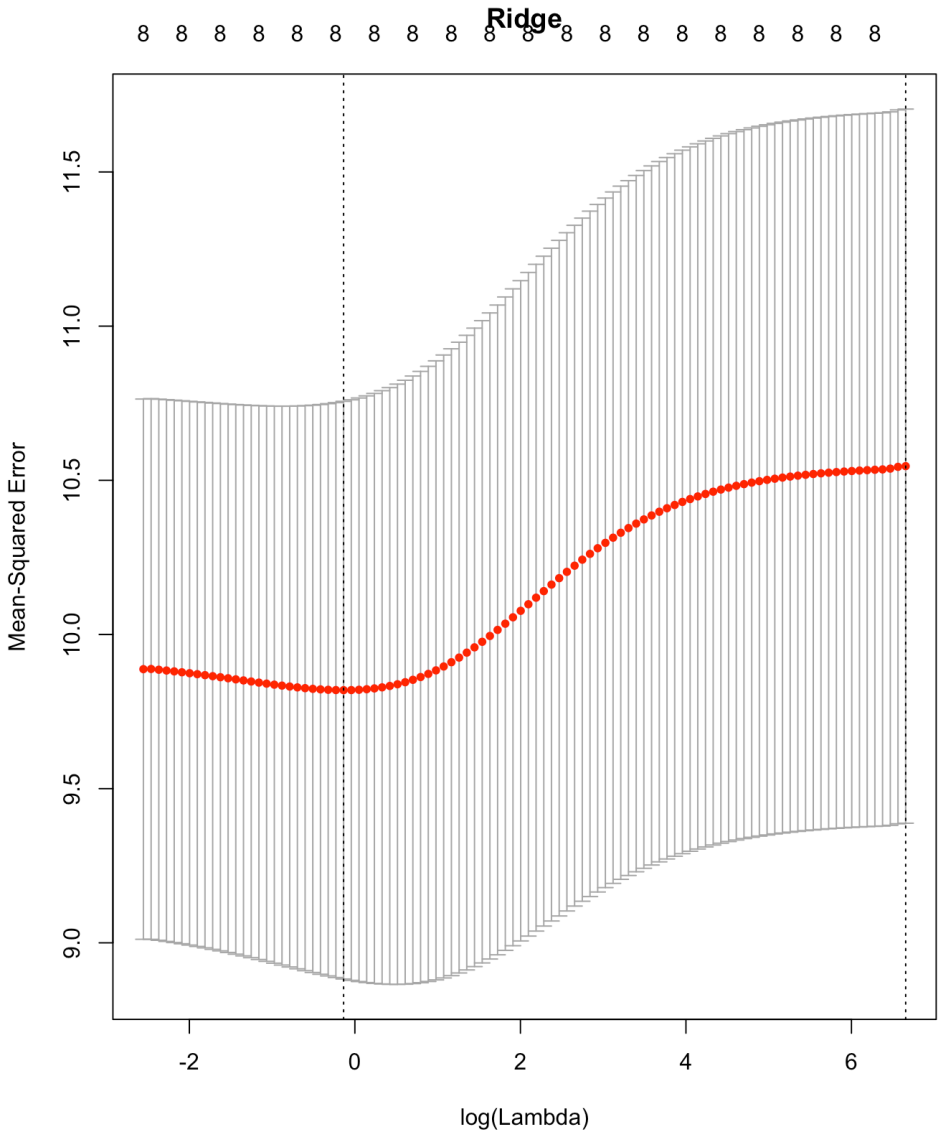
Esempio: relazioni extraconiugali

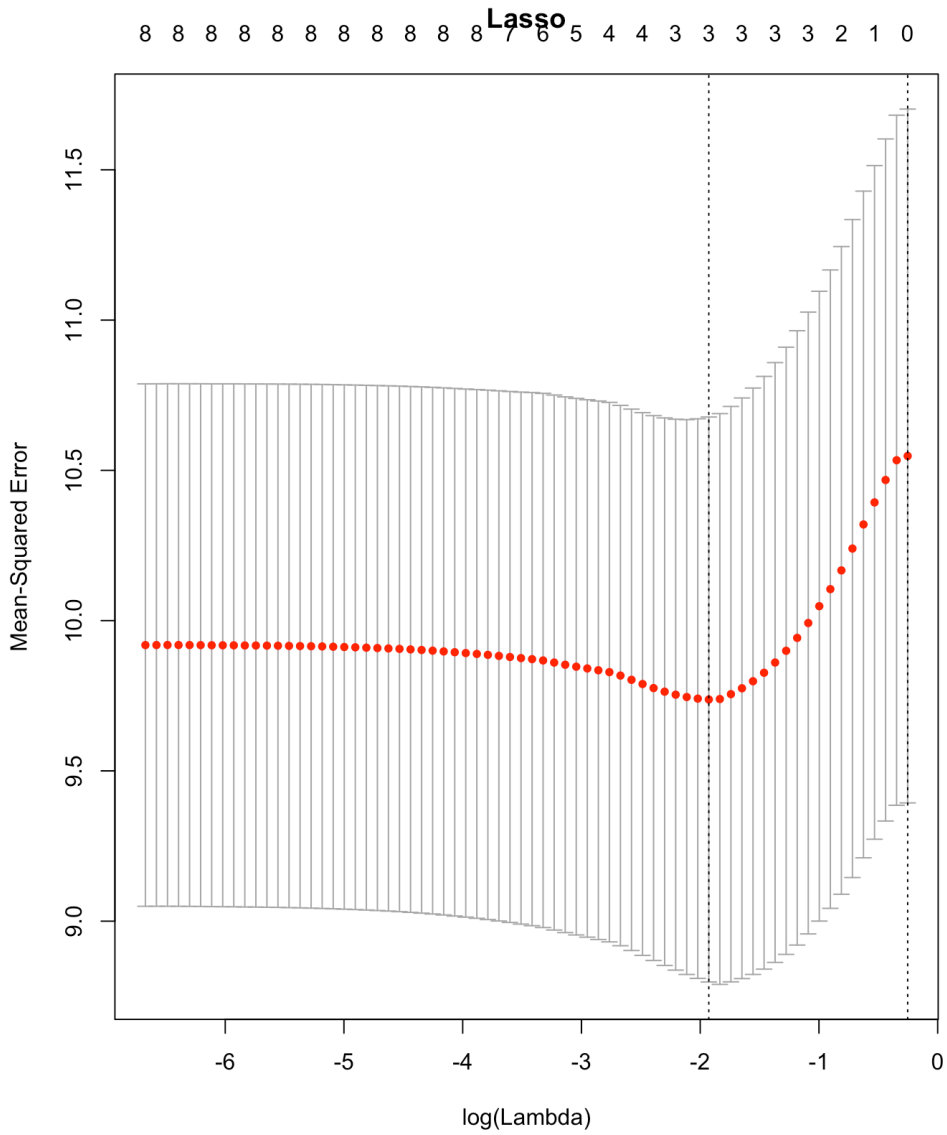
```
library(Ecdat)
reg = lm(nbaffairs~sex+age+child+ym+religious+education+occupation+rate,data=Fair)
x=model.matrix(nbaffairs~sex+age+child+ym+religious+education+occupation+rate,Fair)[,-1]
y=Fair$nbaffairs
library(glmnet)
grid=10^seq(10,-2,length=100)
ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))
ridge.mod$lambda[50]
coef(ridge.mod)[,50]
sqrt(sum(coef(ridge.mod)[-1,50]^2))
ridge.mod$lambda[60]
coef(ridge.mod)[,60]
sqrt(sum(coef(ridge.mod)[-1,60]^2))
predict(ridge.mod,s=50,type="coefficients")[1:9,]
set.seed(1)
train=sample(1:nrow(x), nrow(x)/2)
test=(-train)
y.test=y[test]
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=grid, thresh=1e-12)
ridge.pred=predict(ridge.mod,s=4,newx=x[test,])
mean((ridge.pred-y.test)^2)
mean((mean(y[train])-y.test)^2)
ridge.pred=predict(ridge.mod,s=1e10,newx=x[test,])
mean((ridge.pred-y.test)^2)
ridge.pred=predict(ridge.mod,s=0,newx=x[test,],exact=T,x=x[train,],y=y[train])
mean((ridge.pred-y.test)^2)
lm(y~x, subset=train)
predict(ridge.mod,s=0,exact=T,type="coefficients",x=x[train,],y=y[train])[1:9,]
```

Esempio: relazioni extraconiugali

```
set.seed(1)
cv.out=cv.glmnet(x[train,],y[train],alpha=0)
plot(cv.out)
bestlam=cv.out$lambda.min
bestlam
ridge.pred=predict(ridge.mod,s=bestlam,newx=x[test,])
mean((ridge.pred-y.test)^2)
out=glmnet(x,y,alpha=0)
predict(out,type="coefficients",s=bestlam)[1:9,]

lasso.mod=glmnet(x[train,],y[train],alpha=1,lambda=grid)
plot(lasso.mod)
set.seed(1)
cv.out=cv.glmnet(x[train,],y[train],alpha=1)
plot(cv.out)
bestlam=cv.out$lambda.min
lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])
mean((lasso.pred-y.test)^2)
out=glmnet(x,y,alpha=1,lambda=grid)
lasso.coef=predict(out,type="coefficients",s=bestlam)[1:9,]
lasso.coef
lasso.coef[lasso.coef!=0]
```

Esempio: relazioni extraconiugali





Oltre la linearità

Nell'analizzare ciò che accade nel mondo reale, è importante essere a conoscenza che *«la verità non è (quasi) mai lineare».*

L'assunzione della linearità però, è utilizzata spesso perché è un'approssimazione *abbastanza* buona della realtà, ma quando non siamo di fronte a questo caso si utilizzano altri modelli:

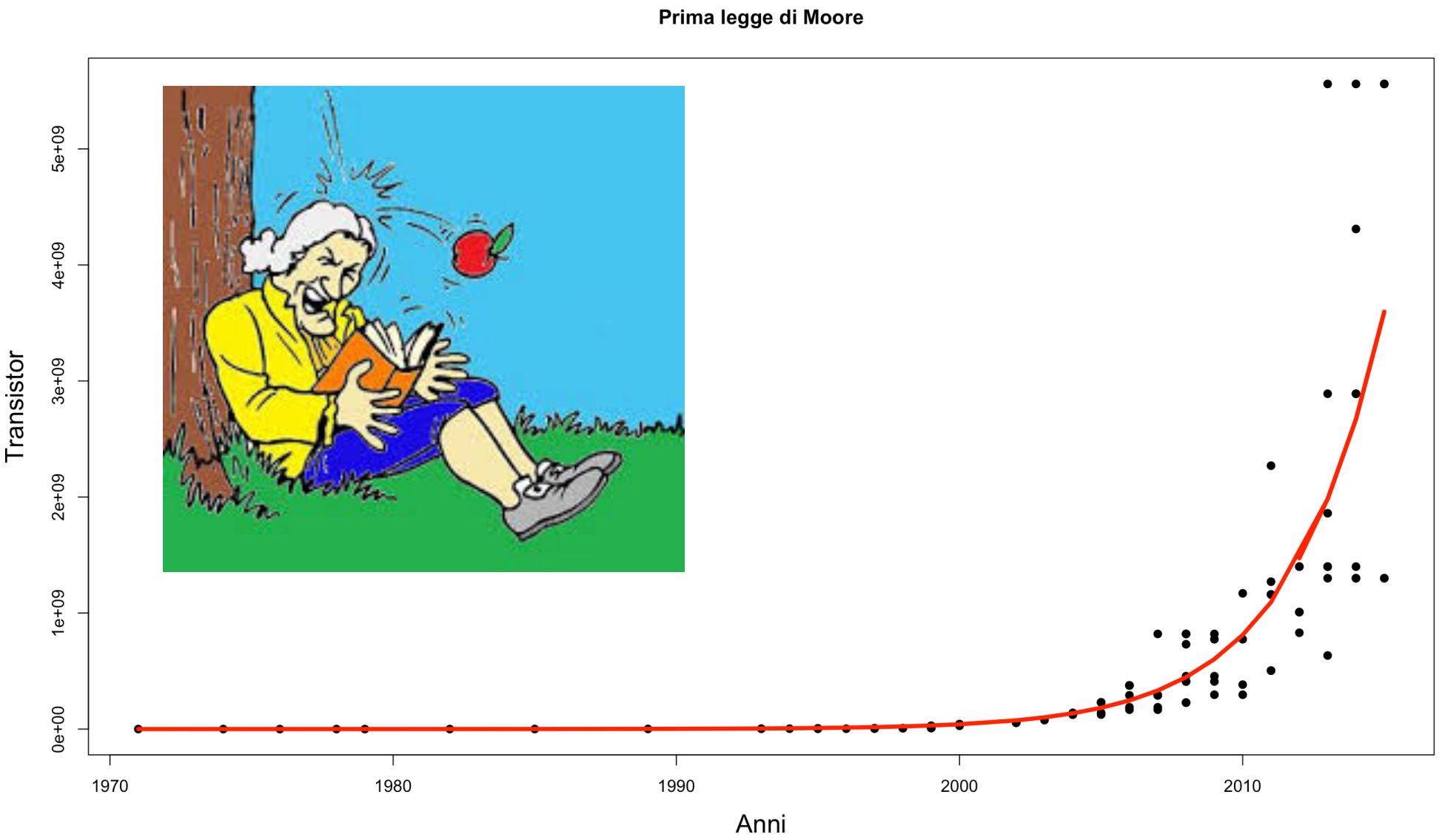
- Polinomiali,
- Step functions,
- Splines,
- Regressioni locali,
- Modelli additivi generalizzati.

Che offrono molta flessibilità, senza perdere la facilità di costruzione ed interpretazione dei modelli lineari.

Esempio: legge di Moore

Nel 1965 Moore, cofondatore di Intel con Robert Noyce, che all'epoca era a capo del settore R&D della Fairchild Semiconductor e tre anni dopo fondò la Intel, ipotizzò che il numero di transistor nei microprocessori sarebbe raddoppiato ogni 12 mesi circa. Nel 1975 questa previsione si rivelò corretta e prima della fine del decennio i tempi si allungarono a due anni, periodo che rimarrà valido per tutti gli anni ottanta. La legge, che verrà estesa per tutti gli anni novanta e resterà valida fino ai nostri giorni, viene riformulata alla fine degli anni ottanta ed elaborata nella sua forma definitiva, ovvero che il numero di transistori nei processori raddoppia ogni 18 mesi. Questa legge è diventata il metro e l'obiettivo di tutte le aziende che operano nel settore.

Esempio: legge di Moore



Minimi quadrati non lineari

In statistica la regressione nonlineare è un metodo di stima di una curva dalla forma generale:

$$y_i = f(x_i|\theta) + \varepsilon_i$$

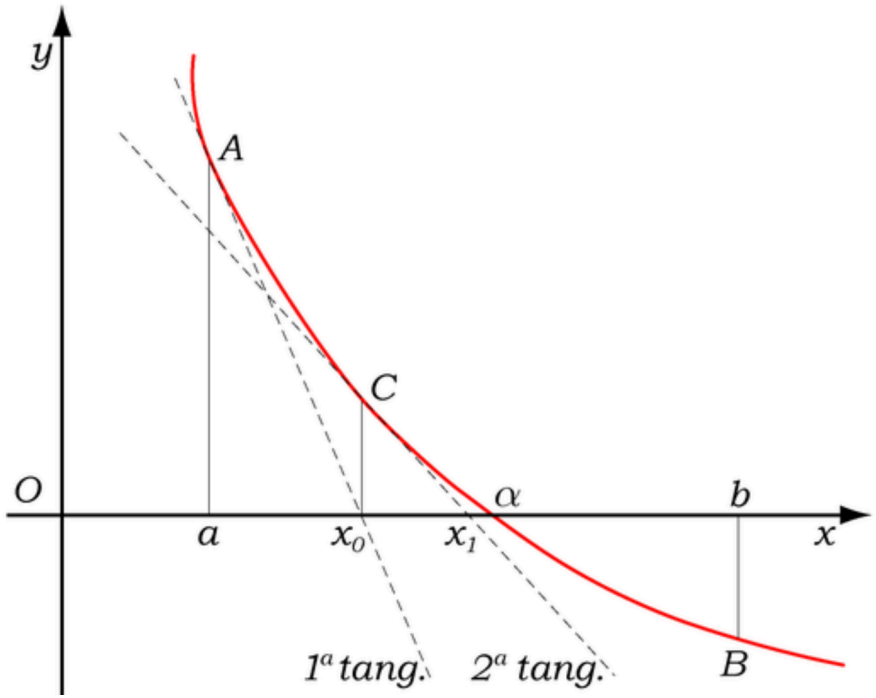
Spesso si ricorre ad una linearizzazione della *f*() usando ad esempio i logaritmi della curva, ma se questo non è possibile perché la non linearità è nei parametri, dobbiamo utilizzare i minimi quadrati non lineari:

$$\hat{\theta} = \min_{\theta} \sum_{i=1}^{n} \left(y_i - f(x_i|\theta)\right)^2$$

Diversamente da quanto accade nel caso della regressione lineare, non esiste un metodo generale per determinare i valori dei parametri che garantiscono di minimizzare i quadrati dei residui. A tal fine, si ricorre a classi di algoritmi numerici di ottimizzazione, che a partire da valori iniziali, scelti a caso o tramite un'analisi preliminare, giungono a punti ritenuti ottimali. Si potrebbero avere dei massimi locali della bontà del fitting, in contrasto ancora con il caso della regressione lineare, in cui il minimo è globale.

Algoritmo di Newton

Il metodo di Newton o metodo di Newton-Raphson, è uno dei metodi per il calcolo approssimato di una soluzione di un'equazione della forma f(x)=0. Il metodo consiste nel sostituire alla curva la tangente alla curva stessa, partendo da un qualsiasi punto; per semplicità si può iniziare da uno dei due punti che hanno come ascissa gli estremi dell'intervallo [a,b] e assumere, come valore approssimato della radice, l'ascissa x_t del punto in cui la tangente interseca l'asse delle x internamente all'intervallo.



Procedendo in modo iterativo si dimostra che dimostra che la seguente successione converge alla radice piuttosto rapidamente:

$$x_t = x_{t-1} - \frac{f(x_{t-1})}{f'(x_{t-1})}$$

Esempio:alg. di Newton per legge di Moore

```
mdati <- read.csv("moore.txt",header = T,sep=";")
plot(mdati$anno,mdati$transistors,cex=1,pch=19,xlab="Anni",ylab="Transistor",main="P
rima legge di Moore",cex.lab=1.5)
moore   <-   nls(transistors~exp(b0)*exp(b1*anno),data=mdati,start=list(b0=1,b1=0.2),
trace=TRUE,nls.control(maxiter=1000))
summary(moore)
summary(rr <- lm(log(transistors)~anno,data=mdati))
pred <- predict(moore,se.fit = T)
lines(mdati$anno,pred,lw=4,col="red")
1/(coef(moore)[2]/log(2)); 1/(coef(rr)[2]/log(2))

Parameters:
     Estimate Std. Error t value Pr(>|t|)
b0 -578.09133   72.72834  -7.949 2.27e-12 ***
b1    0.29781    0.03612   8.245 5.06e-13 ***

Number of iterations to convergence: 386
Achieved convergence tolerance: 1.432e-06

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.358e+02  1.028e+01  -61.86   <2e-16 ***
anno         3.264e-01  5.128e-03   63.65   <2e-16 ***
Multiple R-squared:  0.9747,      Adjusted R-squared:  0.9745

Tempo di raddoppio
nls: 2.327452  lm: 2.123412
```

Funzione logistica o sigmoidale

Una funzione logistica o curva logistica descrive una curva ad S di crescita di alcuni tipi di popolazioni P. All'inizio la crescita è quasi esponenziale, successivamente rallenta, diventando quasi lineare, per raggiungere una posizione asintotica dove non c'è più crescita.

L'equazione fu pubblicata per la prima volta da Pierre F. Verhulst nel 1838, che derivò la sua équation logistique (equazione logistica) per descrivere le auto-limitazioni di crescita di una popolazione biologica. L'equazione viene talvolta chiamata equazione di Verhulst-Pearl dopo che è stata riscoperta nel 1920.

La funzione logistica può essere utilizzata per illustrare il progresso della diffusione di un'innovazione tecnica, lungo il suo ciclo di vita. Storicamente quando vengono introdotti nuovi prodotti si investe molto in ricerca e sviluppo; ciò conduce a notevoli miglioramenti qualitativi e riduce i costi. Tutto questo comporta un periodo di crescita rapida dell'industria. I drastici aumenti di efficienza, nonché le associate opportunità di riduzione dei costi, si esauriscono; al contempo il prodotto o processo in questione si diffonde saturando il mercato, restando pochi potenziali nuovi acquirenti.

Funzione logistica o sigmoidale

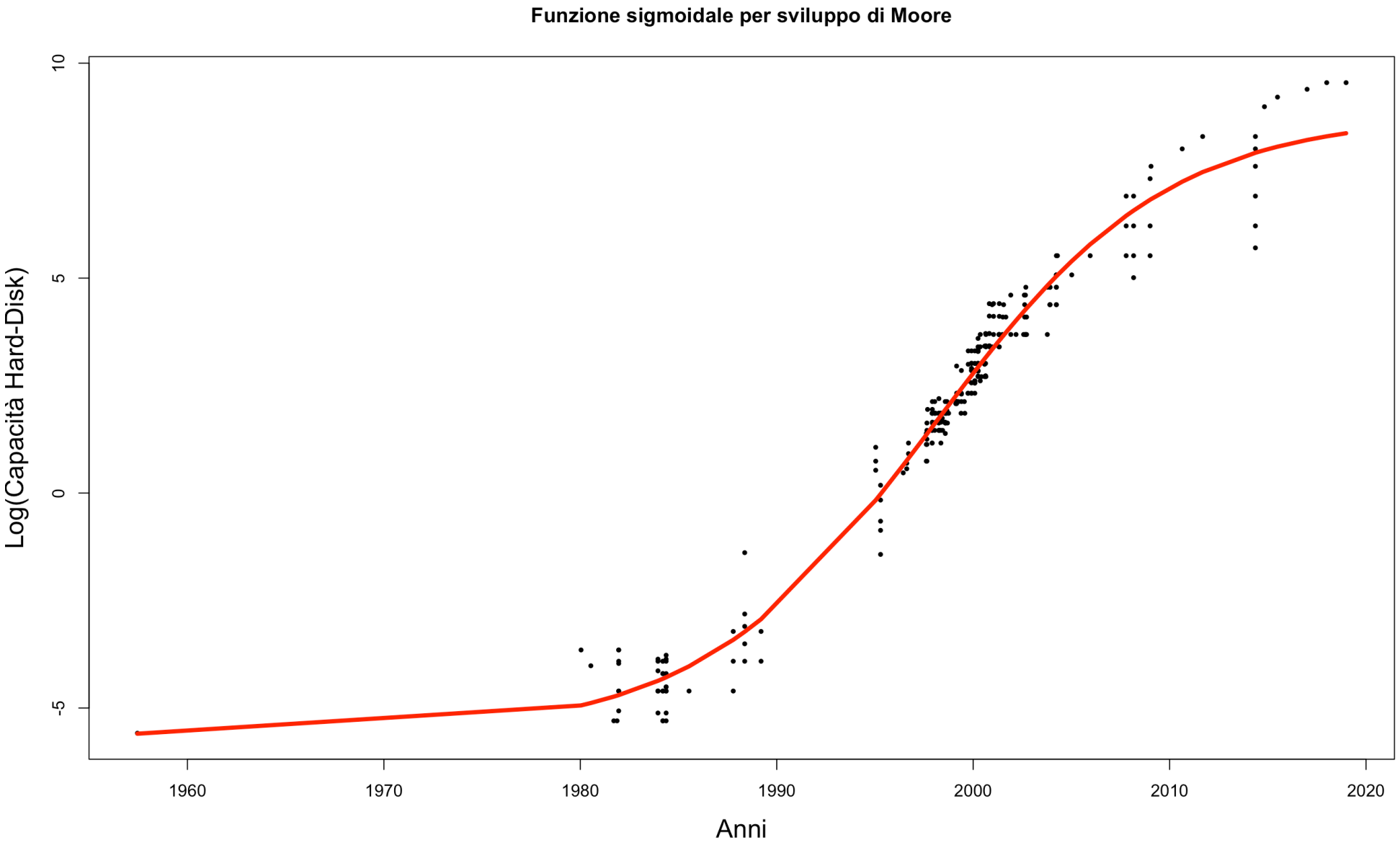
Data l'equazione logistica/sigmoidale in una forma generale:

$$y_i = \varphi_0 + \frac{\varphi_1}{1 + \varphi_2 e^{\varphi_3 x}} + \varepsilon_i$$

Non è possibile linearizzare nelle x per poter stimare i parametri φ, quindi dobbiamo ricorrere ai minimi quadrati non-lineari.

```
data <- as.Date(ISOdate(hdati$anno,hdati$mm,hdati$gg))
lncap <- log(hdati$cap)
plot(data,lncap,cex=0.5,pch=19,xlab="Anni",ylab="Log(Capacità     Hard-Disk)",
main="Funzione sigmoidale per sviluppo di Moore",cex.lab=1.5)
moore      <-      nls(lncap~phi0+phi1/(1+exp(-(phi2+phi3*as.numeric(data)))),
start=list(phi0=-5,phi1=15,phi2=-10,phi3=0.001),trace=TRUE)
pred <- predict(moore,se.fit = T)
lines(as.numeric(data),pred,lw=4,col="red")

1919.468 :  -5.000         15.000        -10.000          0.001
97.29888 :  -4.7309894254 12.2195140210 -5.5335658704  0.0005508794
80.44691 :  -5.5137827917 14.0862746921 -4.6686717098  0.0004568487
77.80665 :  -5.6130160969 14.3970119827 -4.6984456521  0.0004594828
77.80661 :  -5.6137306030 14.3968526416 -4.6974828387  0.0004594216
77.80661 :  -5.6136127219 14.3966472890 -4.6976302114  0.0004594352
```

Esempio: legge di Moore



Esempio funzione di produzione CES

Le funzioni di produzione CES (dall'inglese Constant Elasticity of Substitution) sono una particolare classe di funzioni di produzione, caratterizzate da elasticità di sostituzione costante tra due input.

Questa classe di funzioni venne originariamente proposta da Kenneth Arrow, Robert Solow e altri come generalizzazione delle proprietà delle funzioni di produzione Cobb-Douglas.

La forma originaria (esistono generalizzazioni a più di due input) della funzione CES è (con due fattori produttivi e rendimenti di scala costanti):

$$Y = b[\alpha K^{-\rho}(1-\alpha)L^{-\rho}]^{-\frac{1}{\rho}}$$

in cui:

b è la produttività totale dei fattori;

ρ è un parametro collegato all'elasticità di sostituzione (σ): ρ = (1-σ)/σ;

α determina la distribuzione del reddito (remunerazione) tra i fattori per un dato ρ

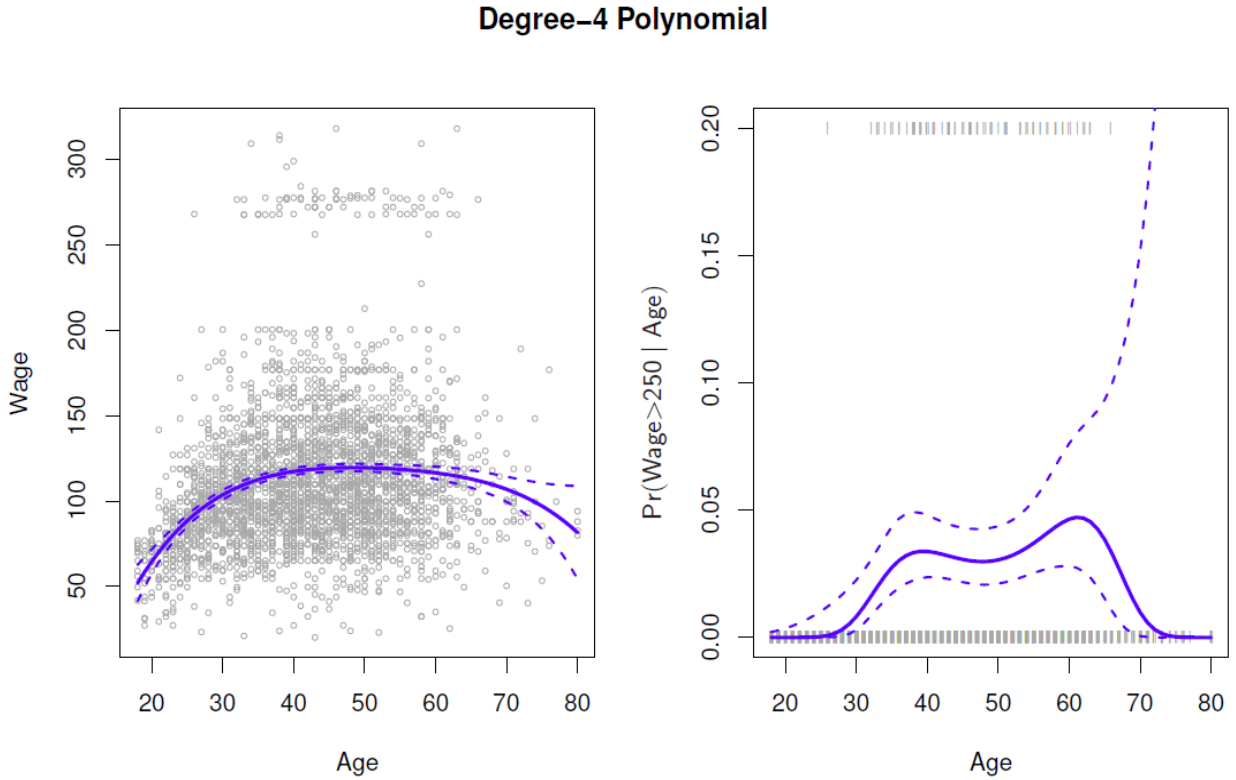
Esempio funzione di produzione CES

```
library(micEconCES)
data("GermanIndustry")
cesInd <- cesEst( "Y", c("K","A","E"), GermanIndustry,method = "NM")
summary(cesInd)

Coefficients:
        Estimate Std. Error t value Pr(>|t|)
gamma     3.7156     7.6430   0.486    0.627
delta_1   0.8198     0.2003   4.093 4.26e-05 ***
delta     0.1466     0.9174   0.160    0.873
rho_1     0.7252     1.0115   0.717    0.473
rho       0.9156     2.4723   0.370    0.711

Multiple R-squared: 0.9646831

Elasticities of Substitution:
                Estimate Std. Error t value Pr(>|t|)
E_1_2 (HM)        0.5797     0.3399   1.706   0.0881 .
E_(1,2)_3 (AU)    0.5220     0.6738   0.775   0.4385

HM = Hicks-McFadden (direct) elasticity of substitution
AU = Allen-Uzawa (partial) elasticity of substitution
```

Regressione polinomiale

Il modello specificato con una regressione polinomiale ha equazione:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \ \ldots + \beta_d x_i^d + \epsilon_i$$



Caratteristiche del modello

Nel modello di regressione polinomiale, vengono create nuove variabili $X_1 = X$, $X_2 = X^2$, ecc. e trattate come facenti parte di un modello di regressione lineare multipla.

Più che nel valore dei coefficienti, c'è interesse nei valori stimati per ogni x_0:

$$\hat{f}(x_0) = \widehat{\beta_0} b + \widehat{\beta_1} x_0 + \widehat{\beta_2} x_0^2 + \widehat{\beta_3} x_0^3 + \widehat{\beta_4} x_0^4$$

Dato che $\hat{f}(x_0)$ è una funzione lineare dei $\widehat{\beta_l}$, si può avere un'espressione della varianza puntuale $\mathrm{Var}[\hat{\mathrm{f}}(\mathrm{x}_0)]$ per ogni valore x_0. Nella figura sinistra precedente sono presenti fit e standard error puntuali per una griglia di valori di x_0.

In formule, $\hat{f}(x_0) \pm 2 \cdot se[\hat{f}(x_0)]$.

Caratteristiche del modello

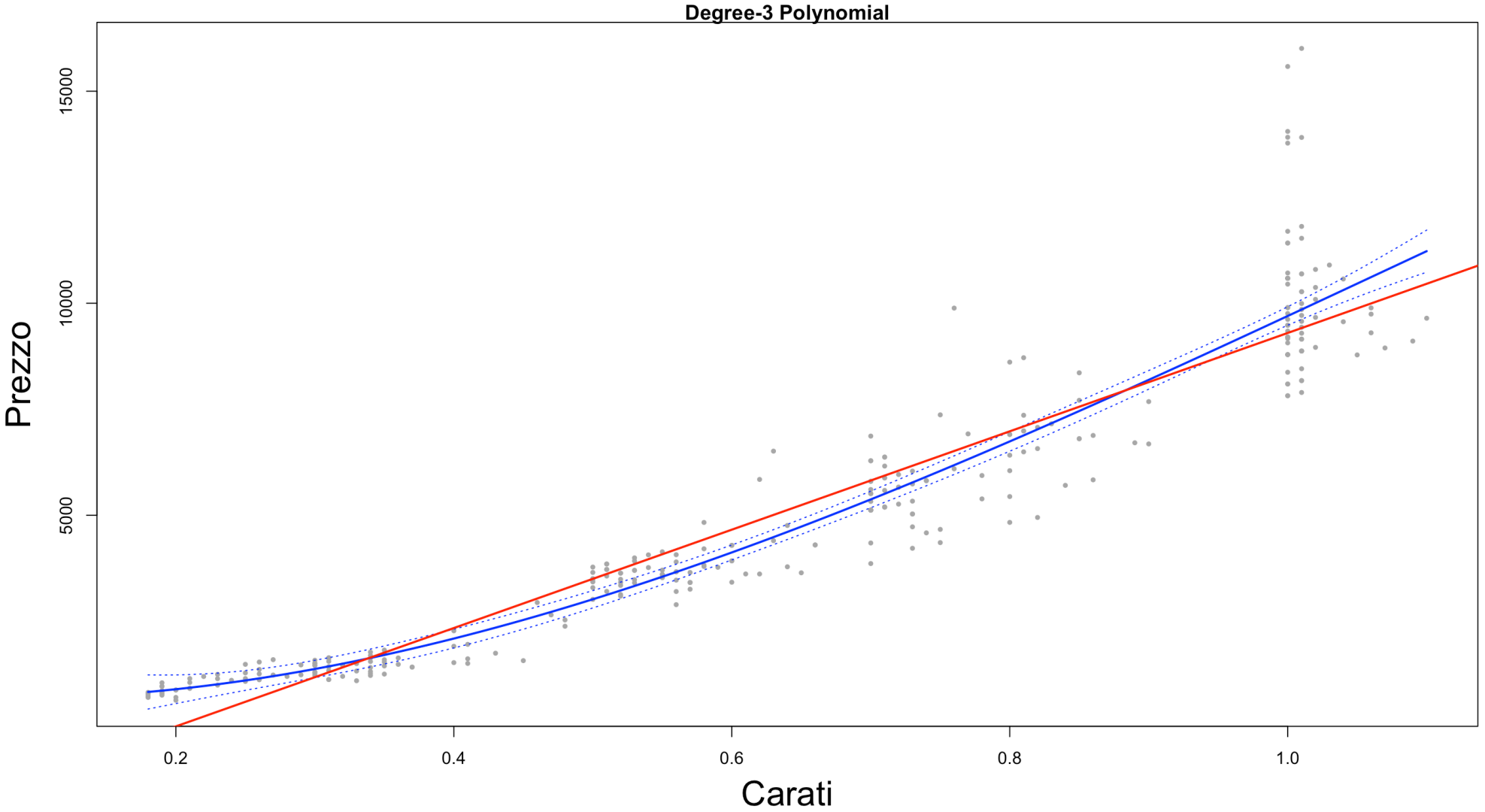
La regressione logistica polinomiale è un'estensione naturale. Ad esempio, nella figura destra precedente è presentato il modello:

$$\Pr(y_i > 250|x_i) = \frac{\exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)}{1 + \exp(\beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d)}$$

Mentre per gli intervalli di confidenza, si calcolano i limiti superiori ed inferiori in scala logit, invertendoli poi per ottenere i valori in probabilità.

Le regressioni polinomiali

Esempio: prezzo dei diamanti



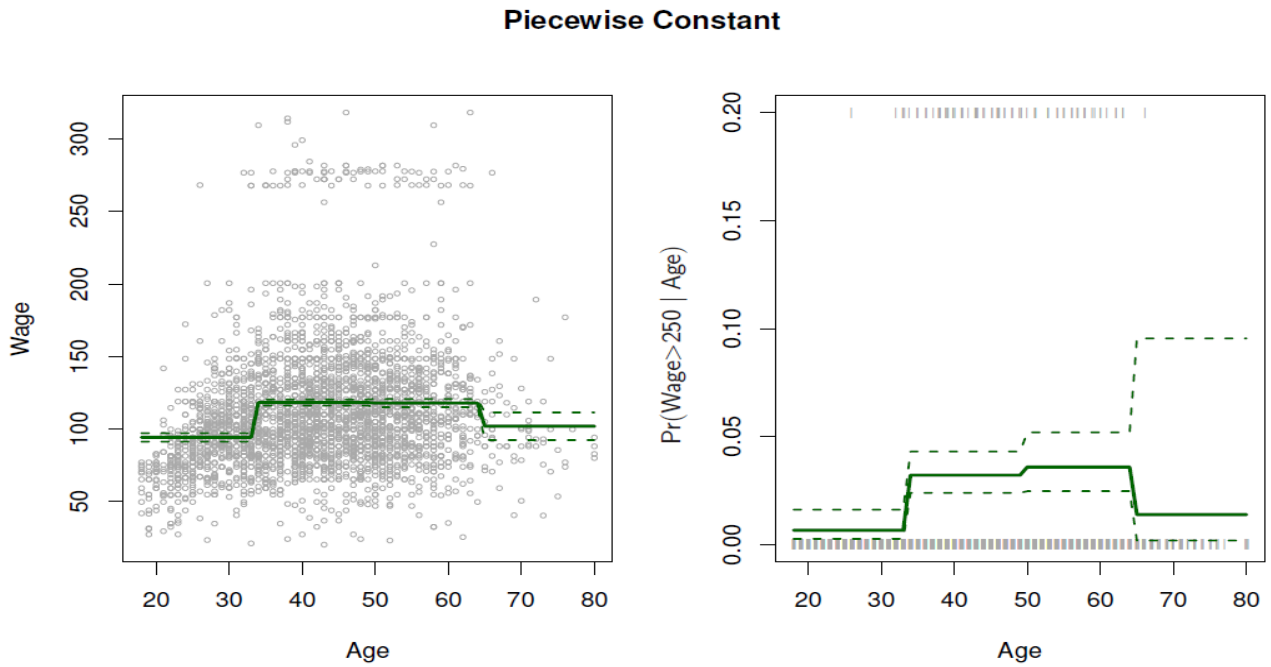
Esempio: prezzo dei diamanti

```
library(Ecdat)
lreg <- lm(price~carat,data=Diamond)
fit=lm(price~poly(carat,3),data=Diamond)
fit2=lm(price~poly(carat,3,raw=T),data=Diamond)
fit2a=lm(price~carat+I(carat^2)+I(carat^3),data=Diamond)
fit2b=lm(price~cbind(carat,carat^2,carat^3),data=Diamond)
caratlims=range(Diamond$carat)
carat.grid=seq(from=caratlims[1],to=caratlims[2],by=0.01)
preds=predict(fit,newdata=list(carat=carat.grid),se=TRUE)
se.bands=cbind(preds$fit+2*preds$se.fit,preds$fit-2*preds$se.fit)
par(mfrow=c(1,1),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
plot(Diamond$carat,Diamond$price,xlim=caratlims,cex=.5,pch=19,col="darkgrey",xlab=
"Carati",ylab="Prezzo",main="Degree-3 Polynomial",cex.lab=2)
lines(carat.grid,preds$fit,lwd=2,col="blue")
matlines(carat.grid,se.bands,lwd=1,col="blue",lty=3)
abline(lreg,lwd=2,col=2)
preds2=predict(fit2,newdata=list(carat=carat.grid),se=TRUE)
max(abs(preds$fit-preds2$fit))
fit.1=lm(price~carat,data=Diamond)
fit.2=lm(price~poly(carat,2),data=Diamond)
fit.3=lm(price~poly(carat,3),data=Diamond)
fit.4=lm(price~poly(carat,4),data=Diamond)
fit.5=lm(price~poly(carat,5),data=Diamond)
anova(fit.1,fit.2,fit.3,fit.4,fit.5)
```

Step function

Un altro metodo per creare trasformazioni di una variabile, consiste nel tagliarla in varie regioni distinte.

$$C_1(X) = I(X < 35),\ C_2(X) = I(35 \leq X < 50), \ldots,\ C_3(\mathrm{X}) = \mathrm{I}(\mathrm{X} \geq 65)$$



Step function

In sostanza, si creano una serie di variabili dummy rappresentanti ogni gruppo distinto.

È un modo molto utile per creare interazioni facili da interpretare. Ad esempio, l' effetto di interazione tra **Year** e **Age**:

$$I(\mathbf{Year} < 2005) \cdot \mathbf{Age}, \quad I(\mathbf{Year} \geq 2005) \cdot \mathbf{Age},$$

Avrà differenti funzioni lineari in ogni categoria. Il problema sorge nello scegliere i punti ti «taglio», o *knots.*

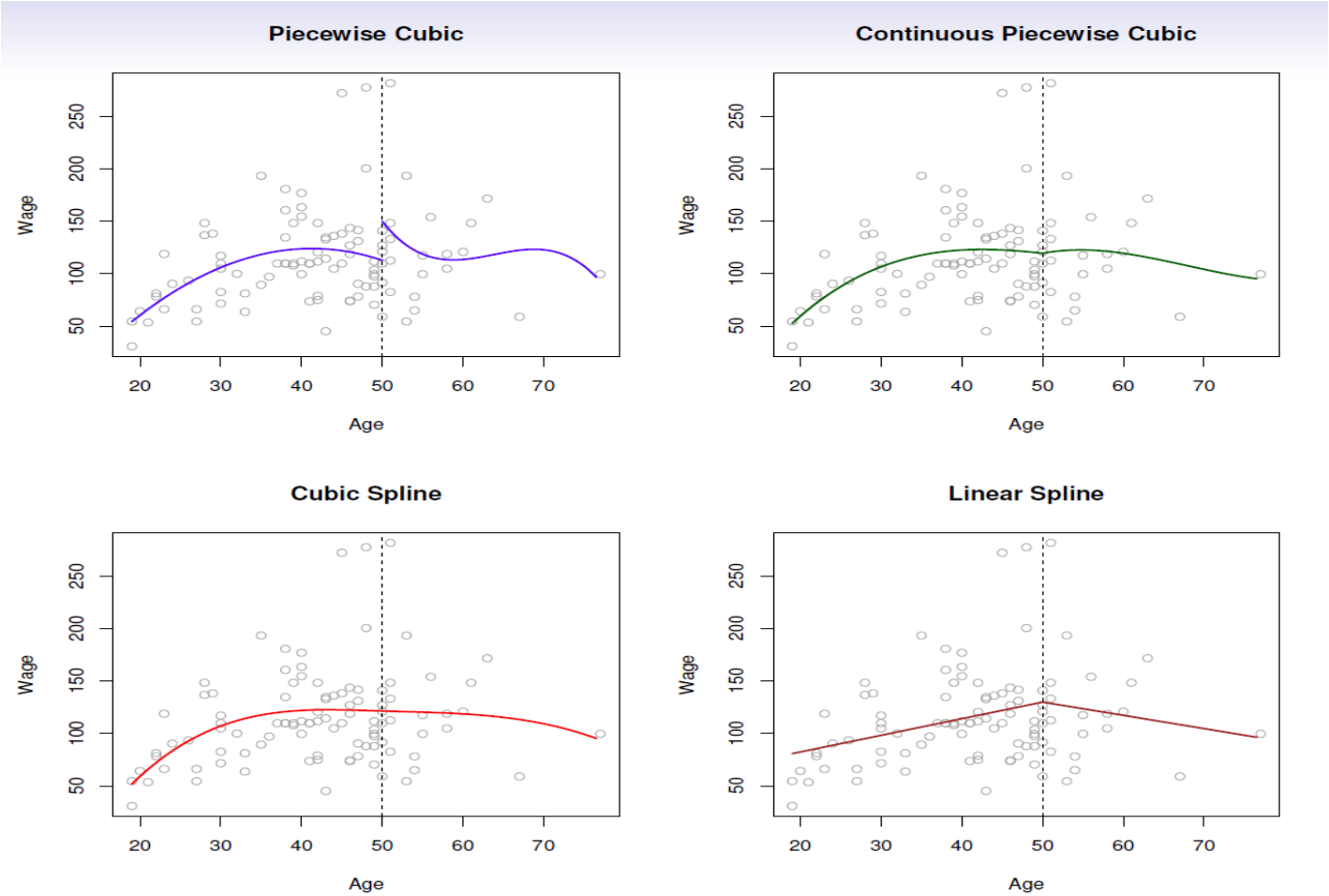
Regressioni Polinomiali locali

Invece di utilizzare una singola funzione polinomiale per rappresentare *X* sull'intero dominio, possiamo utilizzare differenti polinomi in differenti regioni, definite dai knots. Ad esempio

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \epsilon_i & se\ x_i < c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \epsilon_i & se\ x_i \geq c \end{cases}$$

Inoltre, è di fondamentale importanza aggiungere restrizioni ai polinomi, come ad esempio la continuità. (figura seguente). Un'alternativa cosiddetta *smooth* per gestire la non linearità, che incorpora il massimo grado di continuità nelle funzioni polinomiali, è la **Regressione Spline**.

Regressioni Polinomiali locali



Linear Splines

Una regressione spline lineare, con knots a ξ_k, k = 1,…,K è una regressione polinomiale a tratti, continua ad ogni knot.

Possiamo rappresentare tale modello con:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$$

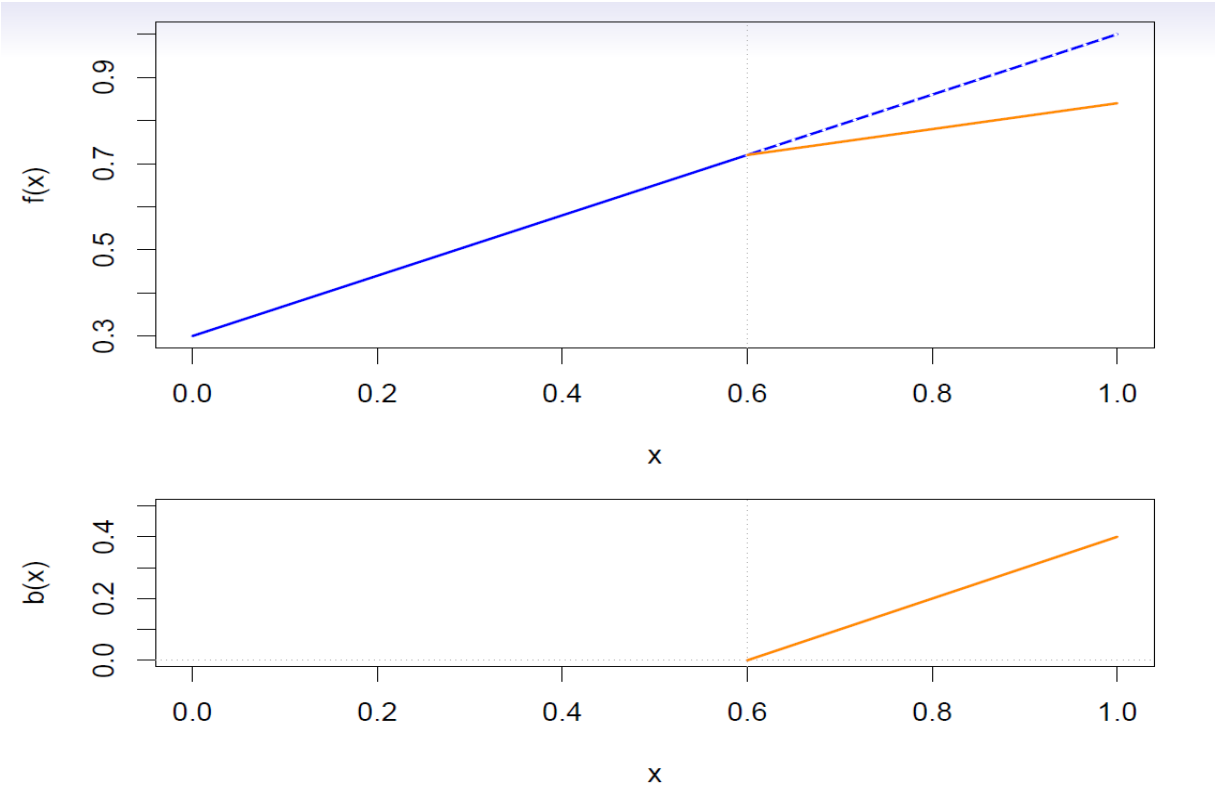
Dove i b_k sono denominate *funzioni base*:

- $b_1(x_i) = x_i$
- $b_{k+1}(x_i) = (x_i - \xi_k)_+ \qquad \mathrm{k} = 1, \ldots, \mathrm{K}$

Mentre $()_+$ significa *solo la parte positiva, cioè:*

$$(x_i - \xi_k)_+ = \begin{cases} (x_i - \xi_k) \; se \; x_i > \xi_k \\ 0 \quad altrimenti \end{cases}$$

Rappresentazione dello spline lineare



Cubic Splines

Una regressione spline cubica, con knots a ξ_k, k = 1,…,K è una regressione polinomiale a tratti, con derivate continue di ordine superiore a 2, ad ogni knot.

Possiamo rappresentare tale modello con *funzioni base*:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+3} b_{K+3}(x_i) + \epsilon_i$$

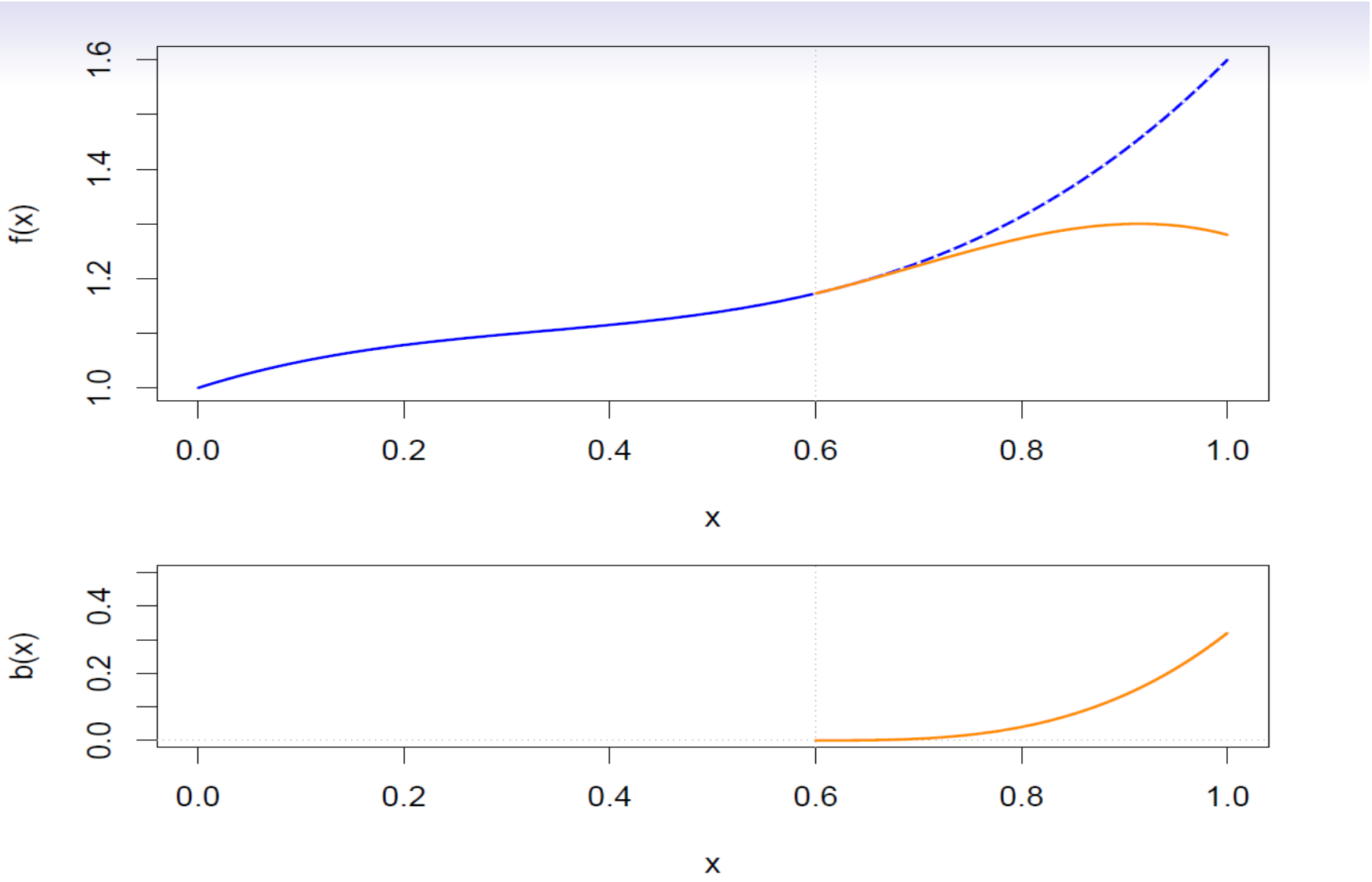
Dove:

- $b_1(x_i) = x_i$
- $b_2(x_i) = x_i^2$
- $b_3(x_i) = x_i^3$
- $b_{k+3}(x_i) = (x_i - \xi_k)_+ \qquad \mathrm{k} = 1, \ldots, \mathrm{K}$

Mentre $()_+$ significa *solo la parte positiva, cioè:*

$$(x_i - \xi_k)_+ = \begin{cases} (x_i - \xi_k) \; se \; x_i > \xi_k \\ 0 \quad altrimenti \end{cases}$$

Rappresentazione dello Spline cubico

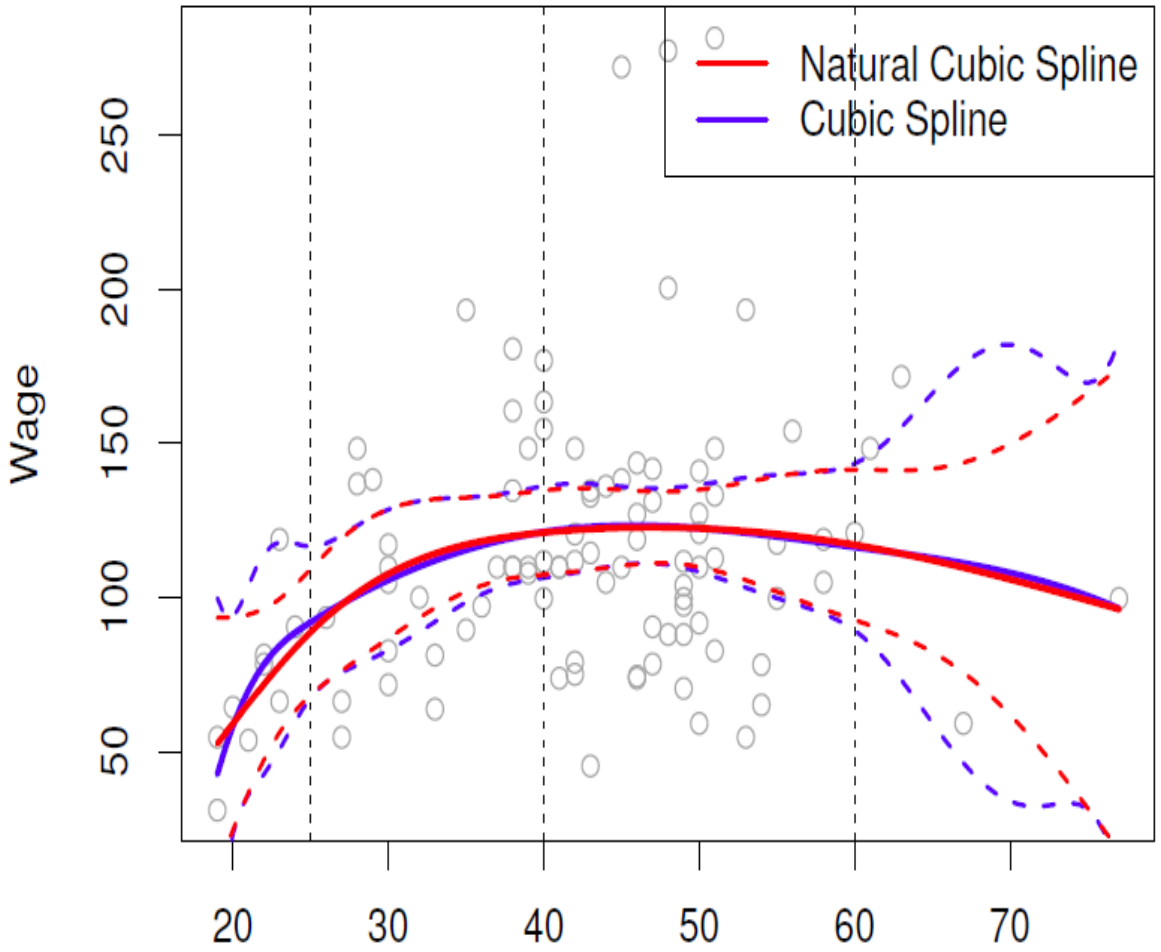


Natural Cubic Spline

Questo tipo di funzione «estrapola linearmente» oltre i knots di partenza.

Aggiunge 4 = 2 x 2 restrizioni in più, permettendo di posizionare più knots interni di quanti ne avremmo posizionati con un regular cubic spline (agli stessi gradi di libertà).

In **R**, si utilizzano per gli splines bs(x,…) e ns(x,…) contenute nel pacchetto *splines*



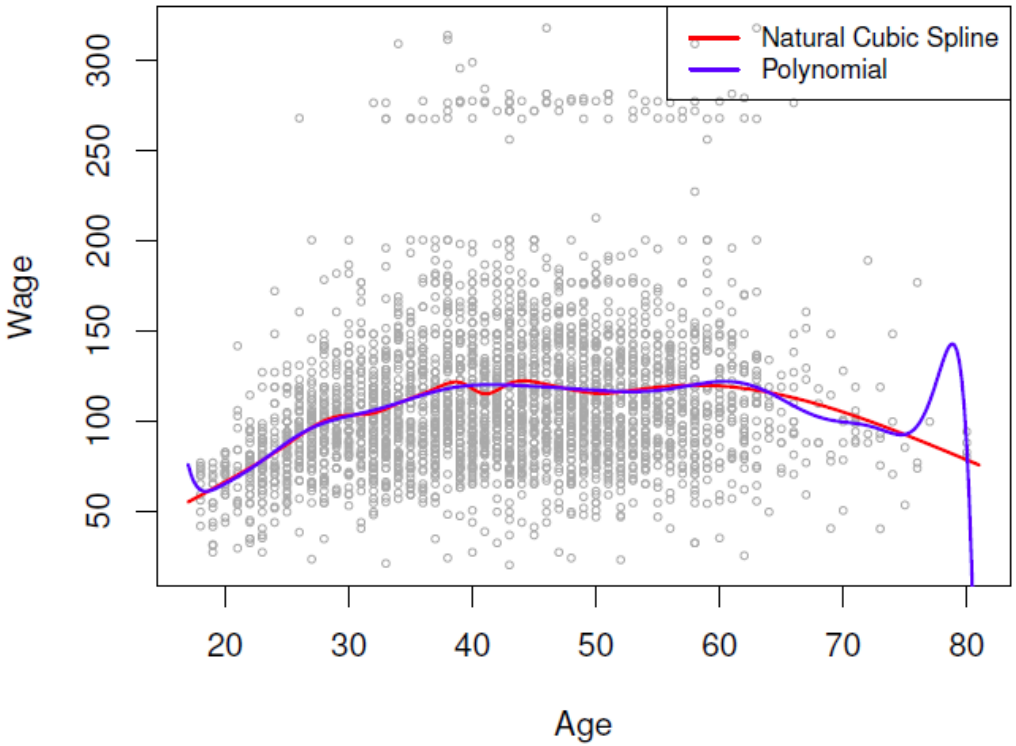
Selezionare i Knots

Il metodo più comune è quello di decidere il numero di knots, e piazzarli nei quantili appropriati della variabile X osservata.

- Un cubic spline con K knots ha $K + 4$ parametri o gradi di libertà;
- Un natural spline con K knots ha invece K gradi di libertà.

Di seguito una comparazione grafica tra i due tipi di funzione, entrambe con 15 gradi di libertà.

Selezionare i Knots



Comparison of a degree-14 polynomial and a natural cubic spline, each with 15df.

```
ns(age, df=14)
poly(age, deg=14)
```

Smoothing Splines

Consideriamo il criterio per interpolare una funzione «smooth» *g(x)* per alcuni dati:

$$\min_{g \in S} \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

- Dove il primo termine è l'RSS, che cerca di rendere *g(x)* adatta per ogni x_i;
- Il secondo termine è chiamato «*roughness penalty*», e controlla la stabilità di *g(x).* È modulato grazie al parametro $\lambda \geq 0$.
  - Minore è λ, più instabile sarà la funzione;
  - Se $\lambda \rightarrow \infty$, la funzione *g(x)* diviene lineare

Soluzione del problema

La soluzione è uno spline cubico, con un knot corrispondente ad ogni valore di x_i. Il termine di penalizzazione controlla la «ruvidità» *(roughness)* grazie al parametro λ.

Dettagli:

- Il metodo Smoothing Splines elimina il problema della selezione dei knots, utilizzando un solo singolo parametro λ.
- La funzione per questo tipo di fit in **R** è *smooth.spline()*
- Il vettore degli n valori interpolati può essere scritto come $\widehat{\boldsymbol{g}_\lambda} = \boldsymbol{S}_\lambda \mathbf{y}$, dove $\boldsymbol{S}_\lambda$ è una matrice n x n determinata dagli x_i e da λ
- L'effettivo numero dei gradi di libertà è dato da:

$$df_\lambda = \sum_{i=1}^{n} \{S_\lambda\}_{ii}$$

Scelta del parametro

Invece di scegliere λ, si può specificare *df.*

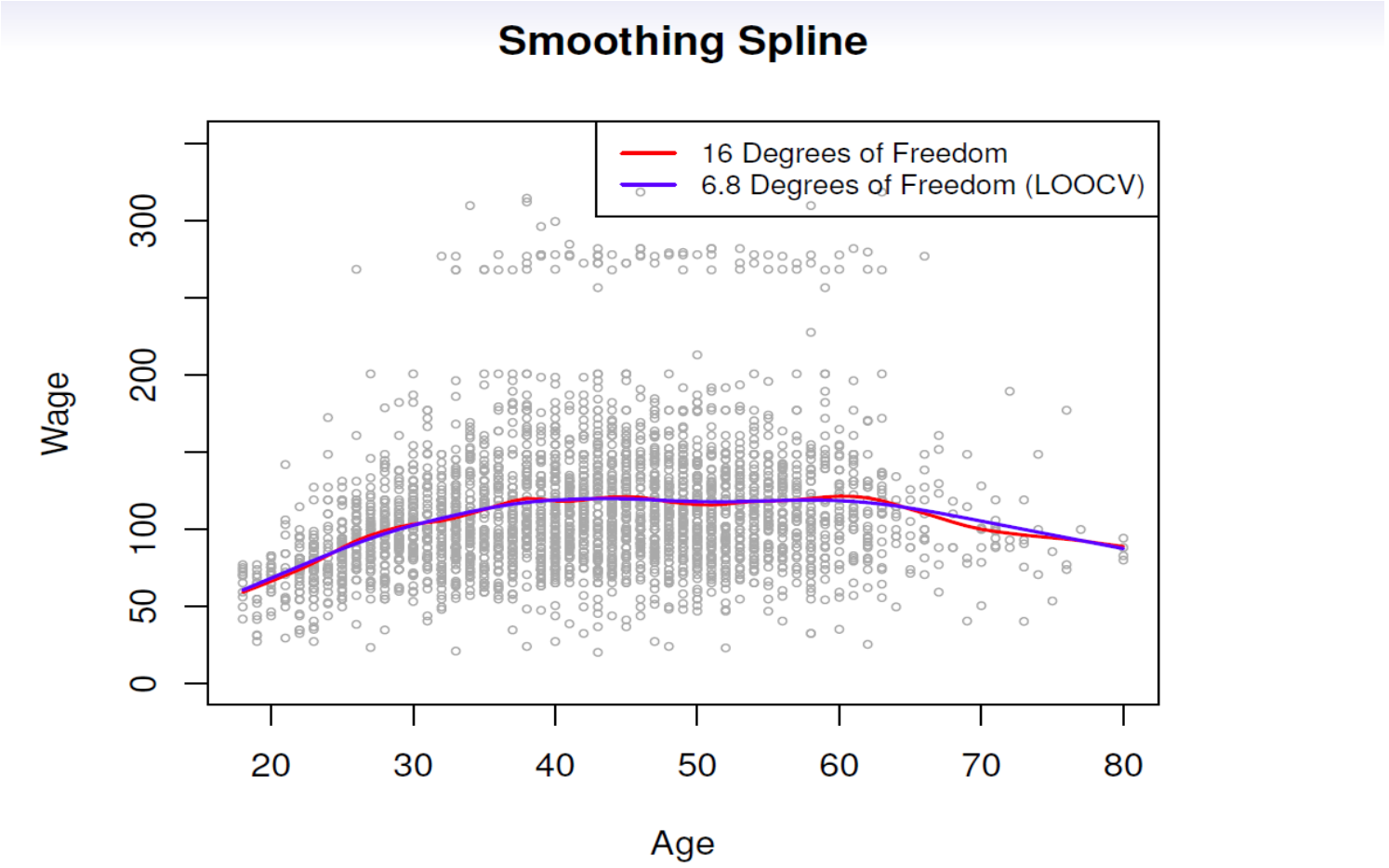
In **R:** smooth.spline (age,wage,df = 10)

L'errore di cross validazione chiamato *«leave-one-out» (LOO)* è dato da:

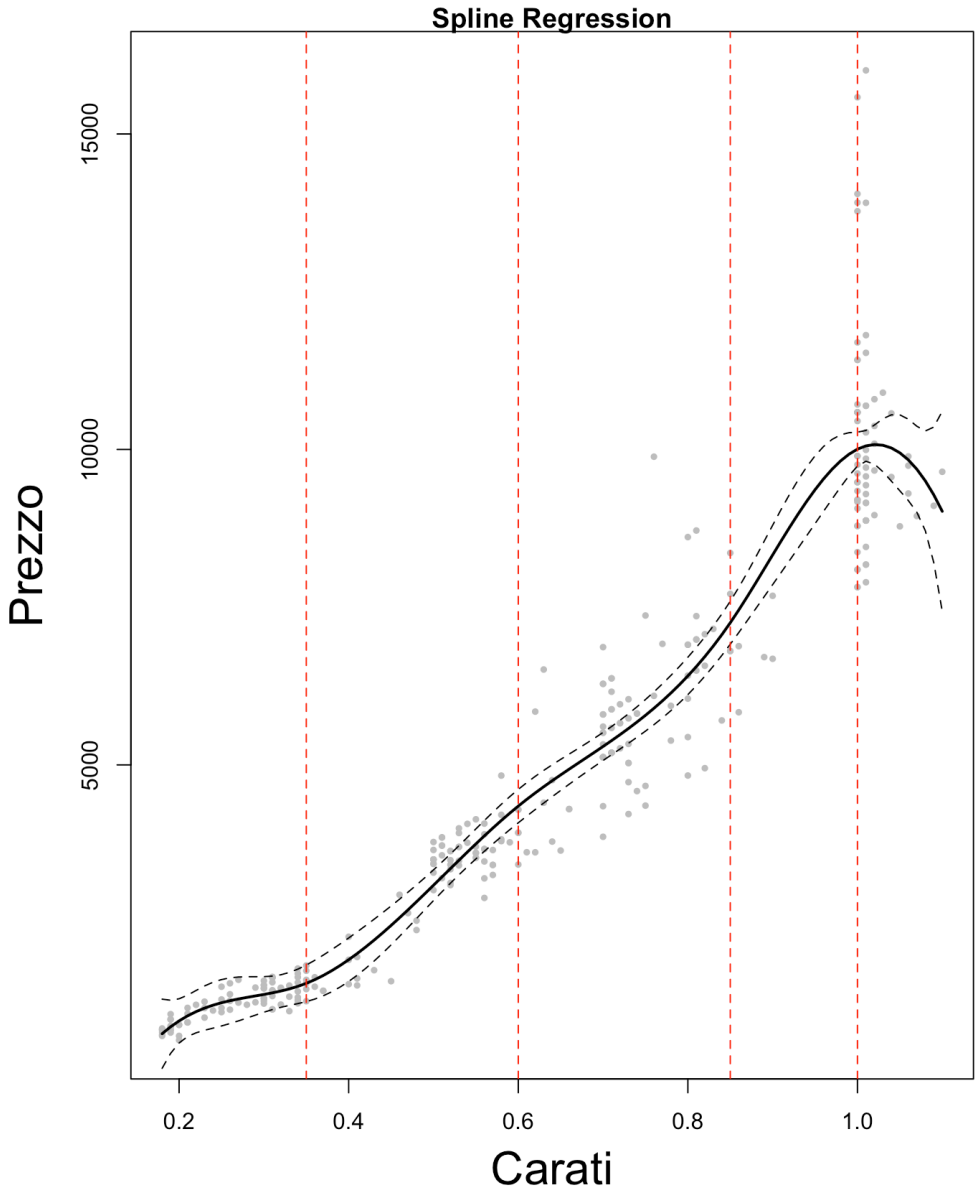
$$RSS_{cv}(\lambda) = \sum_{i=1}^{n} (y_i - \widehat{g_\lambda^{(-i)}}(x_i))^2 = \sum_{i=1}^{n} \left[\frac{y_i - \widehat{g_\lambda}(x_i))}{1 - \{S_\lambda\}_{ii}}\right]^2$$

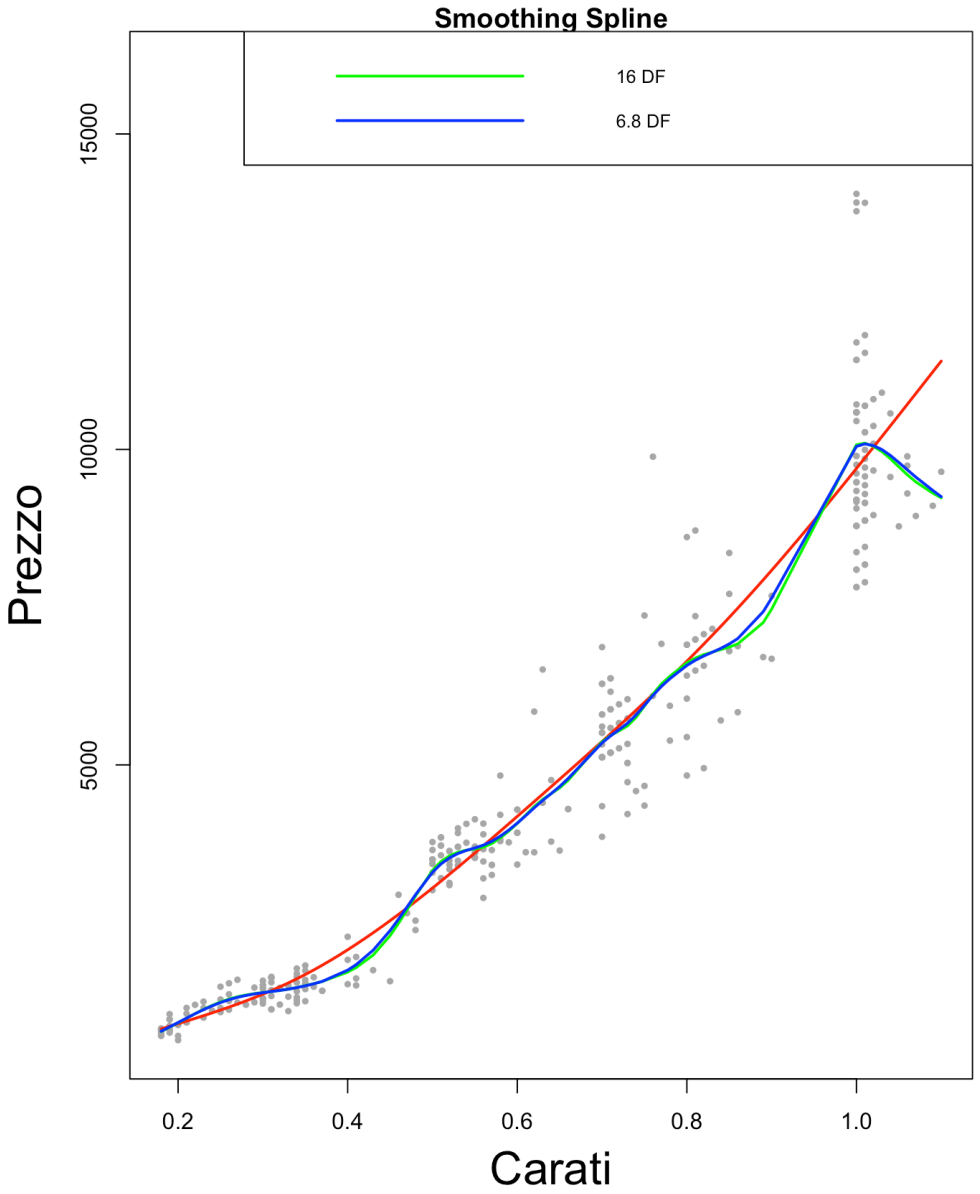
In **R:** smooth.spline (age,wage)

Scelta del parametro

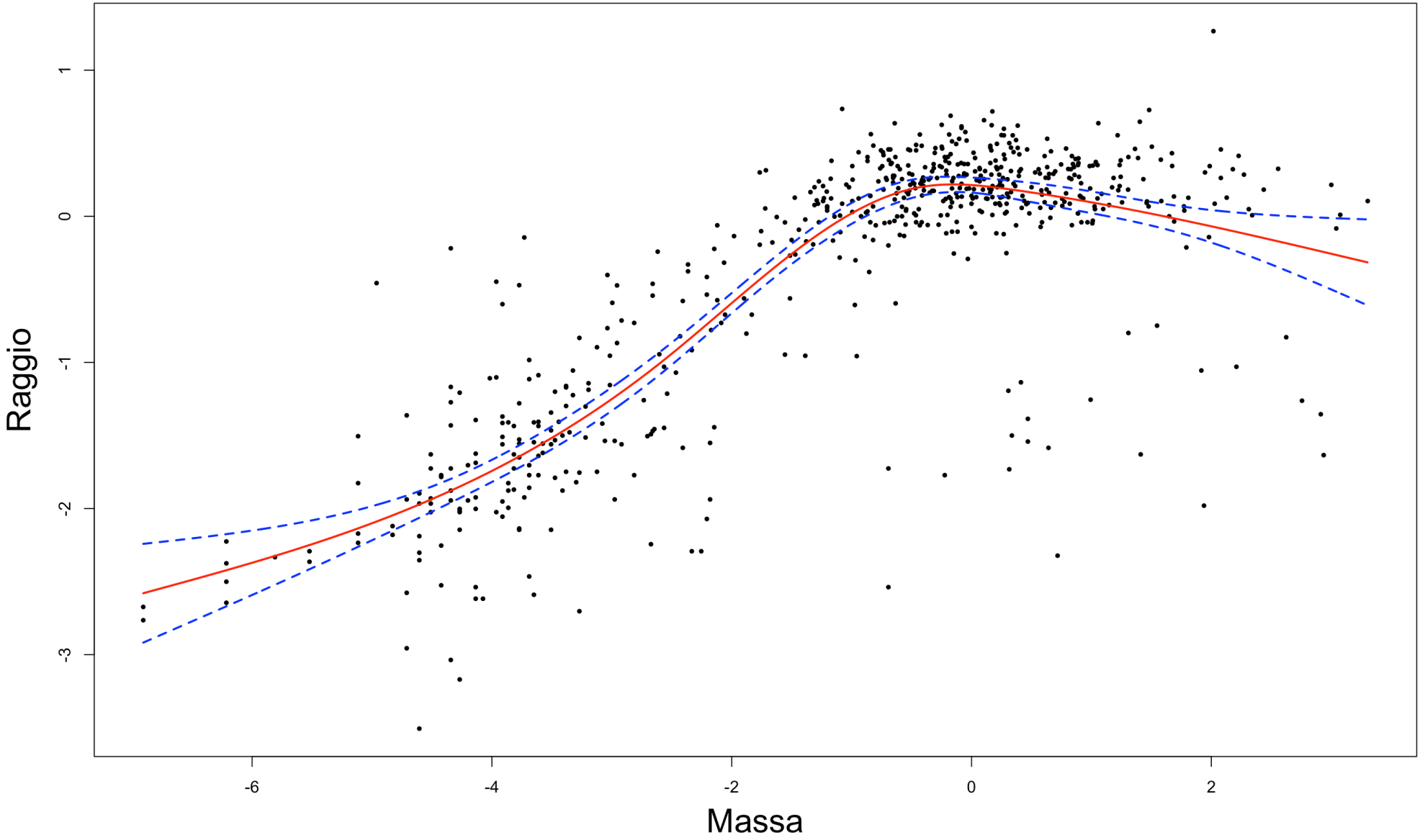


Esempio: prezzo dei diamanti

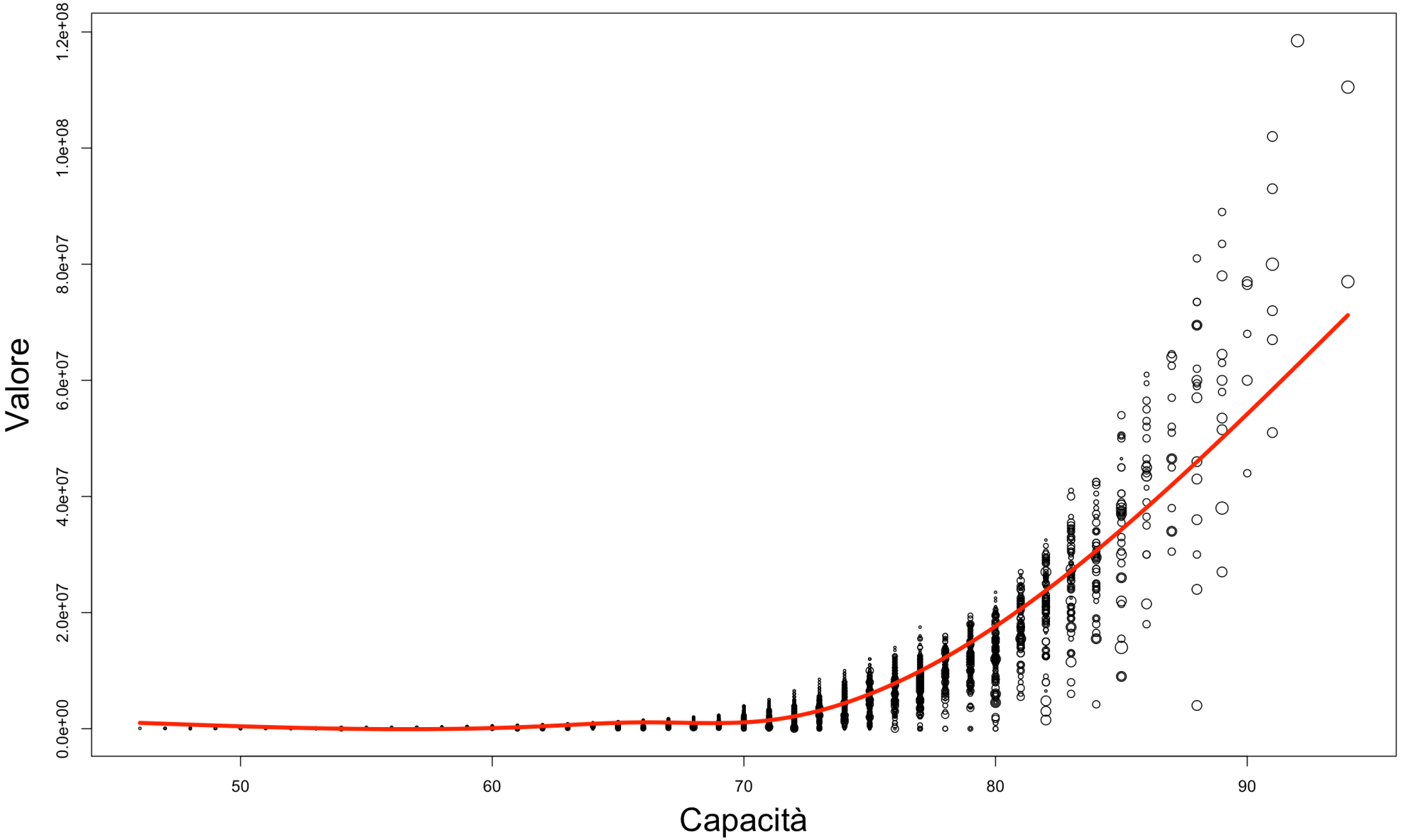




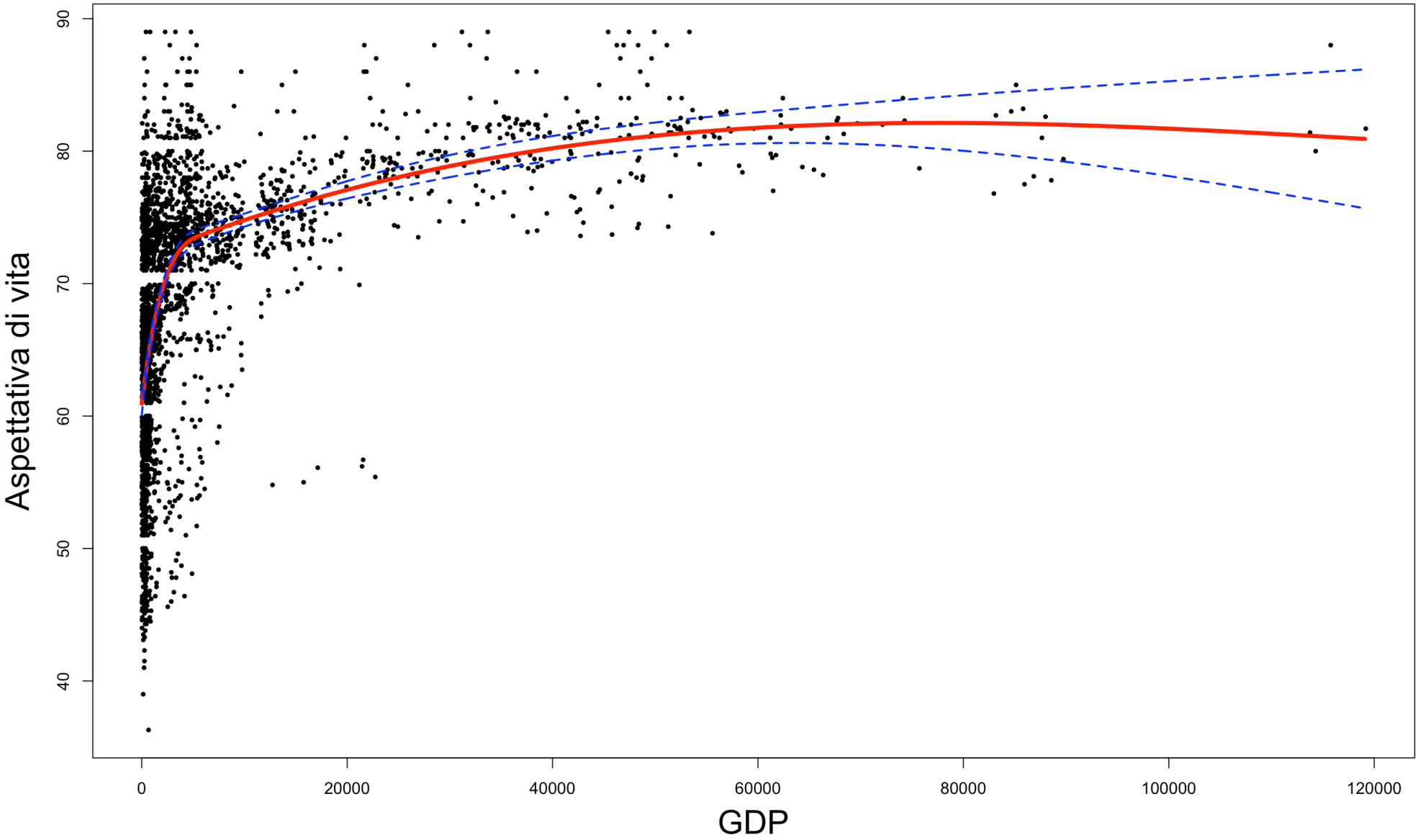
Esempio: Massa e Raggio degli esopianeti



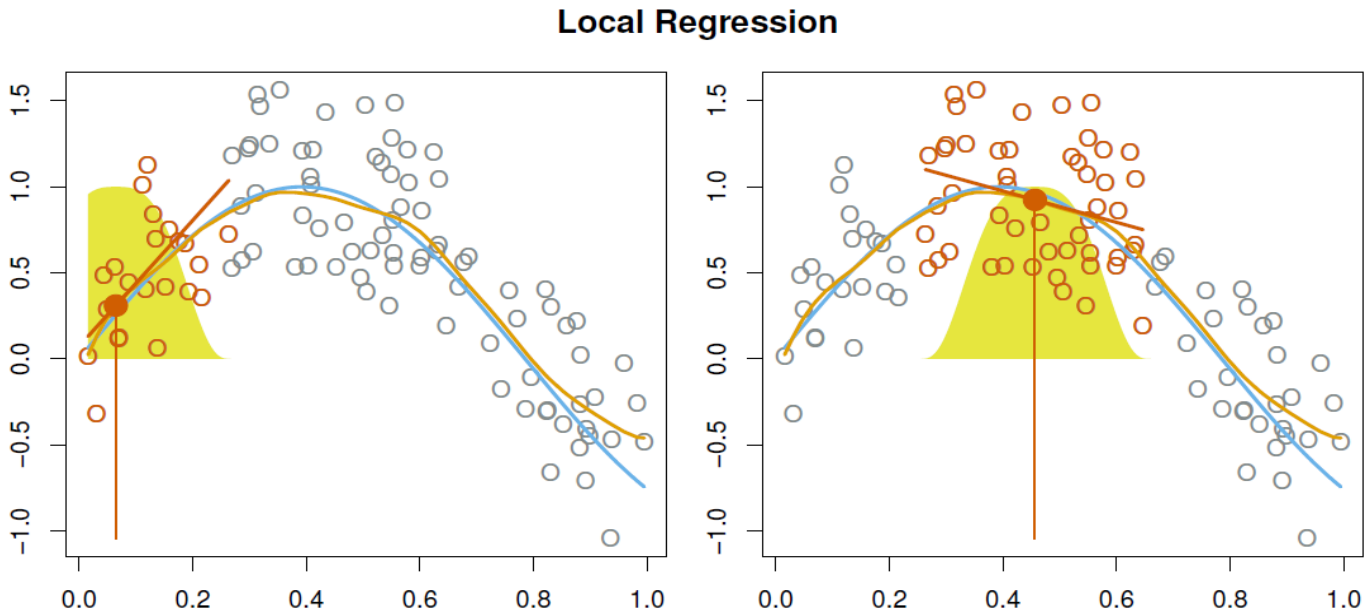
Esempio: valore dei calciatori



Esempio: curva di Preston



Regressione Locale



Nella regression locale, si utilizza una funzione di peso scorrevole, per interpolare in maniera lineare ed iterativa, i punti su tutto il range della variabile X, attraverso minimi quadrati pesati.

In **R**, la funzione per attuare questo tipo di modello è loess()