Modelli non lineari

Strumenti quantitativi per la gestione

Emanuele Taufer

- Metodi per affrontare problemi non-lineari
- Regressione polinomiale
- Esempio: modellare i picchi di domanda di energia
- Fattori che determinano la domanda di energia
- I dati *Temp-2014-F.dat*
- Dataset *Temp*
- Plot Picco-Temperatura massima
- Regressione quadratica
- Regressione cubica
- Confronto grafico
- F-test parziale
- Regressione quartica
- Test-F parziali
- Struttura del fenomeno
- Regressione con funzioni costanti
-
- Basi di funzioni
- Splines
- Nodi e gradi di libertà
-
- La base di splines
- Esempio: spline lineare con un nodo
- Regressione spline cubica con K nodi
- Esempio: *Temp*
- Anova
- Plot splines lineare e cubica
- Spline naturali
- Confronto grafico
- Smoothing splines
- Aspetti formali
- Soluzione per g
- Dati *Temp*
- Smoothing splines
- Regressione locale
- Interpretazione grafica
-
- Dati *Temp*
- Confronto grafico

- Modelli additivi generalizzati (*GAM*)
-
- Pro e contro dei GAM
- Esempio: gam.m1 - spline lineare + costante
- Plot
- Esempio: gam.m2 - spline cubica naturale + costante
- Plot
- Esempio: gam.m3 - smoothing spline e costante
-
- Confronto gam.m1 gam.m3
- Esempio: gam.m4 - regressione locale (span=0.7) + costante
- gam.m5 - regressione locale (span=0.4) + costante
- gam.m6 - regressione locale (span=0.2) + costante
- Confronti
- Riferimenti bibliografici

Metodi per affrontare problemi non-lineari

- Regressione polinomiale
- Regressione con funzioni costanti (*step functions*)
- Regressione polinomiale locale (*regression splines*)
- *Smoothing splines*
- Regressione locale
- *Modelli additivi generalizzati*: approccio che permette di usare diversi tipi di tecniche simultaneamente

Regressione polinomiale

Il metodo più tradizionale per estendere il modello di regressione a contesti non-lineari è quello di usare una funzione polinomiale:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d + \varepsilon_i$$

- può essere stimato semplicemente con il metodo dei minimi quadrati: infatti è un modello di regressione standard con predittori $x_i, x_i^2, x_i^3, \ldots x_i^d$
- difficilmente si usa $d > 4$ poichè il polinomio tende a diventare troppo flessibile
- la stessa tecnica può essere estesa al caso della regressione logistica

Esempio: modellare i picchi di domanda di energia

Per operare efficacemente le aziende elettriche devono essere in grado di prevedere il picco giornaliero di richiesta di energia

La richiesta di energia è misurata in megawatt per ora

Il picco giornaliero è definito come il massimo di domanda nell'arco delle 24 ore

Le compagnie elettriche monitorano costantemente il fenomeno attraverso modelli di regressione

Fattori che determinano la domanda di energia

- Meteo-sensibili:

Il fattore principale in questo caso è la temperatura (collegata all'uso di impianti di riscaldamento e di condizionamento)

- Non meteo-sensibili

Il fattore principale in questo caso è il giorno della settimana: feriale, sabato o domenica (collegato all'uso di frigoriferi, luci, computer, impianti industriali)

I dati *Temp-2014-F.dat*

Osservazioni giornaliere di una compagnia elettrica nel periodo 1/1/14 - 31/12/14

- Y: Picco di domanda in megawatt (*Peak*)
- X_1: Temperatura in F° al momento del picco (Tmax)
- X_2, X_3: indicatori se sabato o domenica

Dataset *Temp*

```
Temp<-read.table("http://www.cs.unitn.it/~taufer/Data/Temp-2014-F.dat",header=T,sep="")
head(Temp)
```

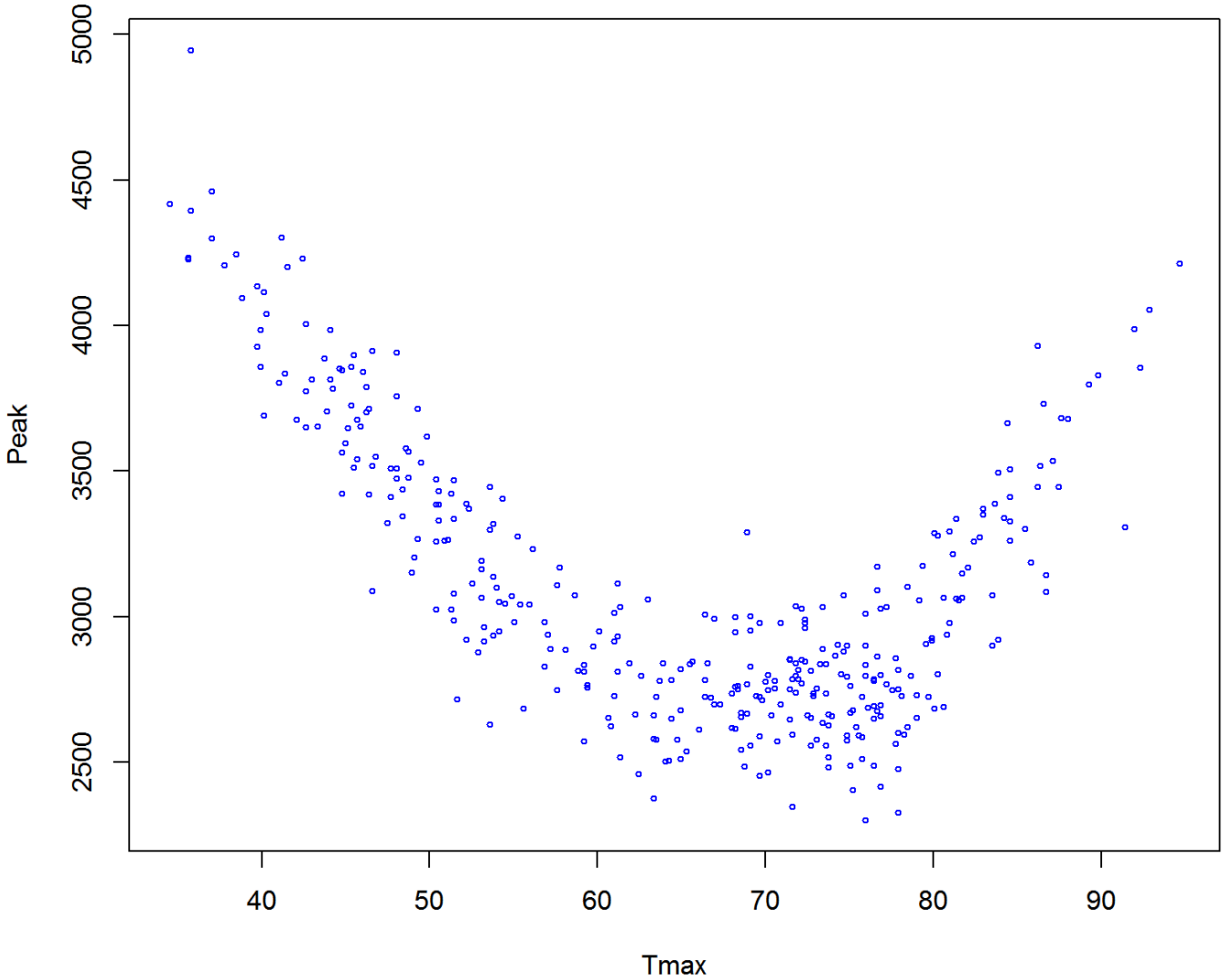
```
        Date  Tmax     Peak Sat Sun
1 2014-01-01 39.74 3928.294   0   0
2 2014-01-02 34.52 4419.036   0   0
3 2014-01-03 40.10 4114.751   0   0
4 2014-01-04 38.84 4096.177   1   0
5 2014-01-05 42.08 3676.929   0   1
6 2014-01-06 47.66 3410.767   0   0
```

Per comodità, costruiamo la variabile `factor` *Day*

```
attach(Temp)
Day<-Sat+2*Sun
Day<-factor(Day, c(0,1,2),c("Wday","Sat","Sun"))
str(Day)
```

```
 Factor w/ 3 levels "Wday","Sat","Sun": 1 1 1 2 3 1 1 1 1 1 ...
```

Plot Picco-Temperatura massima



Regressione quadratica

Concentriamoci per il momento sulla variabile `Tmax` e proviamo ad adattare delle regressioni polinomiali

```
fit2<-lm(Peak~poly(Tmax,2),data=Temp)
summary(fit2)
```

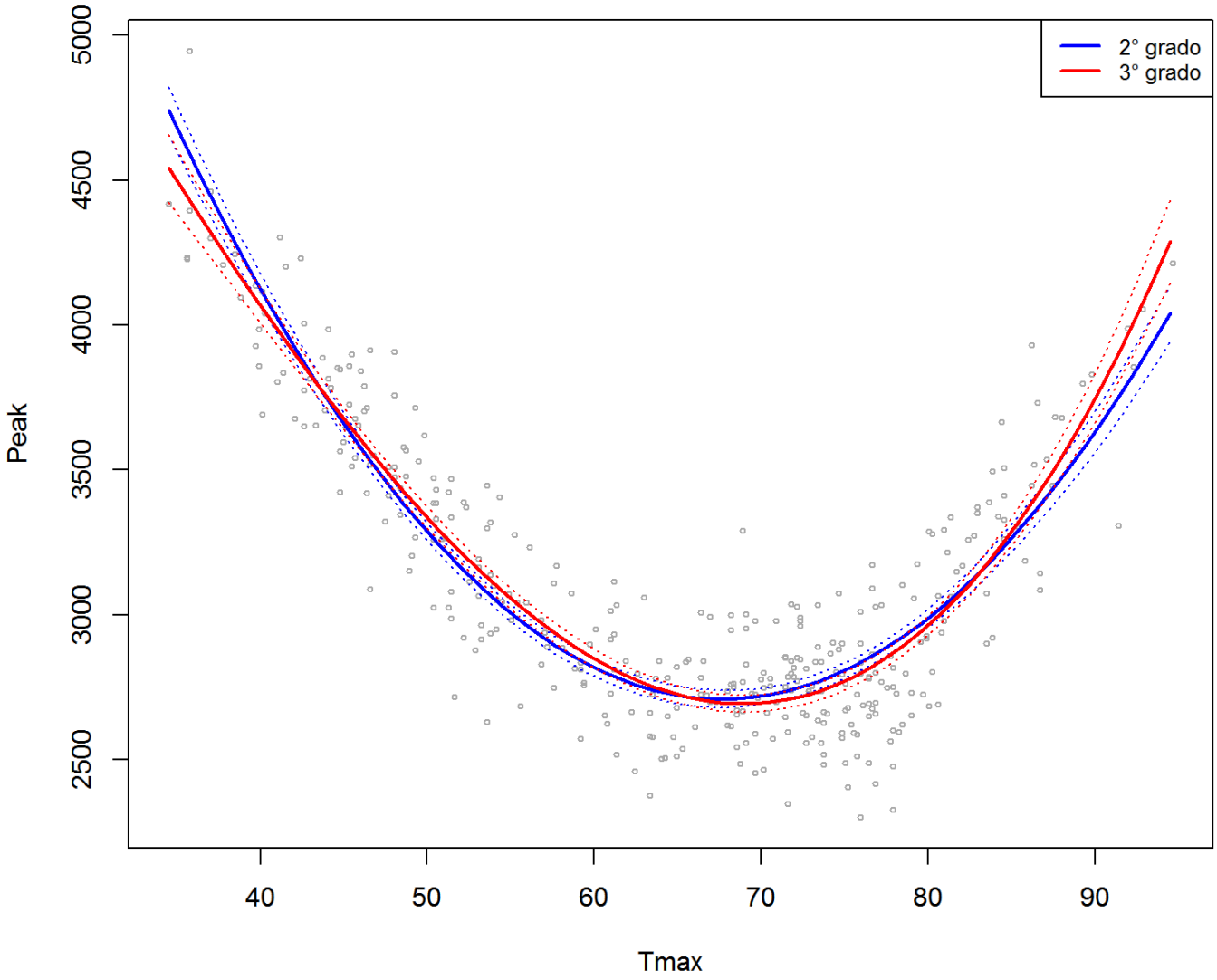
```

Call:
lm(formula = Peak ~ poly(Tmax, 2), data = Temp)

Residuals:
   Min     1Q Median     3Q    Max 
-575.0 -142.2    7.4  135.0  589.1 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     3094.48      10.47  295.57   <2e-16 ***
poly(Tmax, 2)1 -4611.35     200.02  -23.05   <2e-16 ***
poly(Tmax, 2)2  6868.01     200.02   34.34   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 200 on 362 degrees of freedom
Multiple R-squared:  0.8253,    Adjusted R-squared:  0.8244 
F-statistic: 855.3 on 2 and 362 DF,  p-value: < 2.2e-16
```

Regressione cubica

```
fit3<-lm(Peak~poly(Tmax,3),data=Temp)
summary(fit3)
```

```
Call:
lm(formula = Peak ~ poly(Tmax, 3), data = Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-593.65 -126.56    1.35  132.94  597.00

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     3094.48      10.19 303.686  < 2e-16 ***
poly(Tmax, 3)1 -4611.35     194.67 -23.688  < 2e-16 ***
poly(Tmax, 3)2  6868.01     194.67  35.279  < 2e-16 ***
poly(Tmax, 3)3   895.36     194.67   4.599 5.88e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 194.7 on 361 degrees of freedom
Multiple R-squared:  0.835, Adjusted R-squared:  0.8336
F-statistic:   609 on 3 and 361 DF,  p-value: < 2.2e-16
```

Confronto grafico



F-test parziale

```
anova(fit2,fit3)
```

```
Analysis of Variance Table

Model 1: Peak ~ poly(Tmax, 2)
Model 2: Peak ~ poly(Tmax, 3)
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1    362 14482887
2    361 13681218  1    801669 21.153 5.875e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Regressione quartica

```
fit4<-lm(Peak~poly(Tmax,4),data=Temp)
summary(fit4)
```

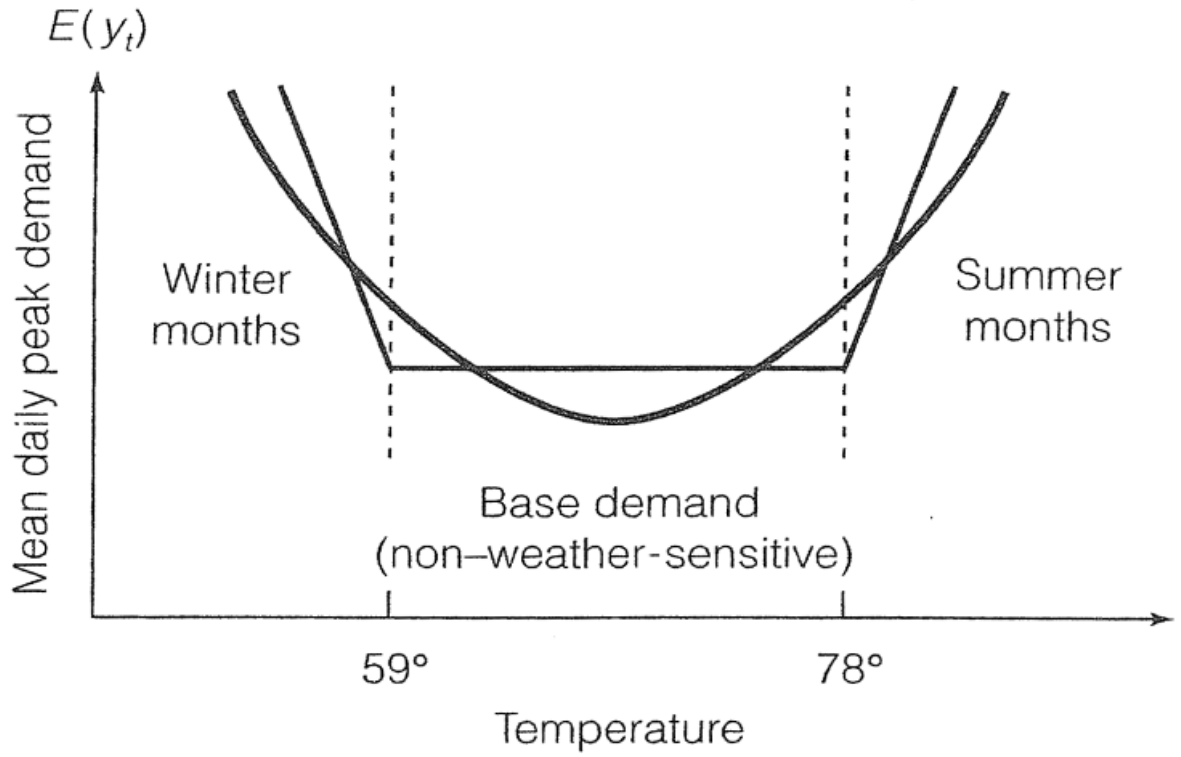
```

Call:
lm(formula = Peak ~ poly(Tmax, 4), data = Temp)

Residuals:
    Min      1Q  Median      3Q     Max 
-572.68 -126.12    3.08  132.23  603.29 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     3094.48      10.19 303.531  < 2e-16 ***
poly(Tmax, 4)1 -4611.35     194.77 -23.675  < 2e-16 ***
poly(Tmax, 4)2  6868.01     194.77  35.261  < 2e-16 ***
poly(Tmax, 4)3   895.36     194.77   4.597 5.94e-06 ***
poly(Tmax, 4)4  -154.74     194.77  -0.794    0.427    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 194.8 on 360 degrees of freedom
Multiple R-squared:  0.8353,    Adjusted R-squared:  0.8335 
F-statistic: 456.4 on 4 and 360 DF,  p-value: < 2.2e-16
```

Test-F parziali

```
anova(fit2,fit3,fit4)
```

```
Analysis of Variance Table

Model 1: Peak ~ poly(Tmax, 2)
Model 2: Peak ~ poly(Tmax, 3)
Model 3: Peak ~ poly(Tmax, 4)
  Res.Df      RSS Df Sum of Sq       F    Pr(>F)    
1    362 14482887                                   
2    361 13681218  1    801669 21.1317 5.943e-06 ***
3    360 13657274  1     23944  0.6312    0.4275    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Struttura del fenomeno



Regressione con funzioni costanti

La regressione polinomiale impone una struttura *globale* ai dati.

Se si vuole uscire da questa logica l'uso di funzioni costanti è una soluzione relativamente semplice

In questo caso il campo di variazione del predittore X è spezzato in diversi intervalli ed il valore di Y in ogni intervallo è stimato da una costante.

In pratica il predittore X è trasformato in variabile categorica ordinata ed il modello è stimato con gli OLS attraverso l'uso di variabili dummy (o *factor*)

Formalmente:

1. si creano gli intervalli definendo gli estremi $c_1, c_2, \ldots, c_K$ e si costruiscono le nuove $K+1$ variabili:

$$
\begin{aligned}
C_0(X) &= I(X < c_1) \\
C_1(X) &= I(c_1 \leq X < c_2) \\
&\ldots \\
C_{K-1}(X) &= I(c_{K-1} \leq X < c_K) \\
C_K(X) &= I(c_K \leq X)
\end{aligned}
$$

I è la funzione indicatore

2. Si stima con OLS il modello

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \cdots + C_K(x_i) + \varepsilon_i$$

Si noti che per evitare multicollinearità perfetta la variabile C_0 è omessa

C_0 diventa il livello di riferimento (o base)

I coefficienti delle dummy sono interpretabili come differenziali rispetto al livello base

Basi di funzioni

L'utilizzo di termini polinomiali o costanti sono dei casi particolari di un approccio generale basato sull'utilizzo di basi di funzioni

L'idea è quella di avere una famiglia di funzioni (o trasformazioni), *conosciute a priori*, da applicare al predittore X:

$$b_1(X), b_2(X), \ldots, b_K(X)$$

Per la regressione polinomiale $b_j(x) = x^j$, $j = 1, \ldots, K$

Per la regressione con funzioni costanti $b_j(x) = I(c_j < x \leq c_{j+1})$

Altre soluzioni sono possibili, tra le quali, l'uso delle *splines*

Splines

L'idea di base della tecnica delle *splines* è quella di utilizzare *più* polinomi di grado non troppo elevato (tipicamente non superiore a 3) su diverse regioni di X.

Ad esempio, supponamo che il campo di variazione di X sia spezzato in due regioni: $x < c$ e $x \geq c$.

Possiamo costruire un modello con due rette (polinomi di primo grado) come

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \varepsilon_i & \text{se } x_i < c \\ \beta_{02} + \beta_{12} x_i + \varepsilon_i & \text{se } x_i \geq c \end{cases}$$

Allo stesso modo possiamo definire un modello con due polinomi di 3° grado come

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

Nodi e gradi di libertà

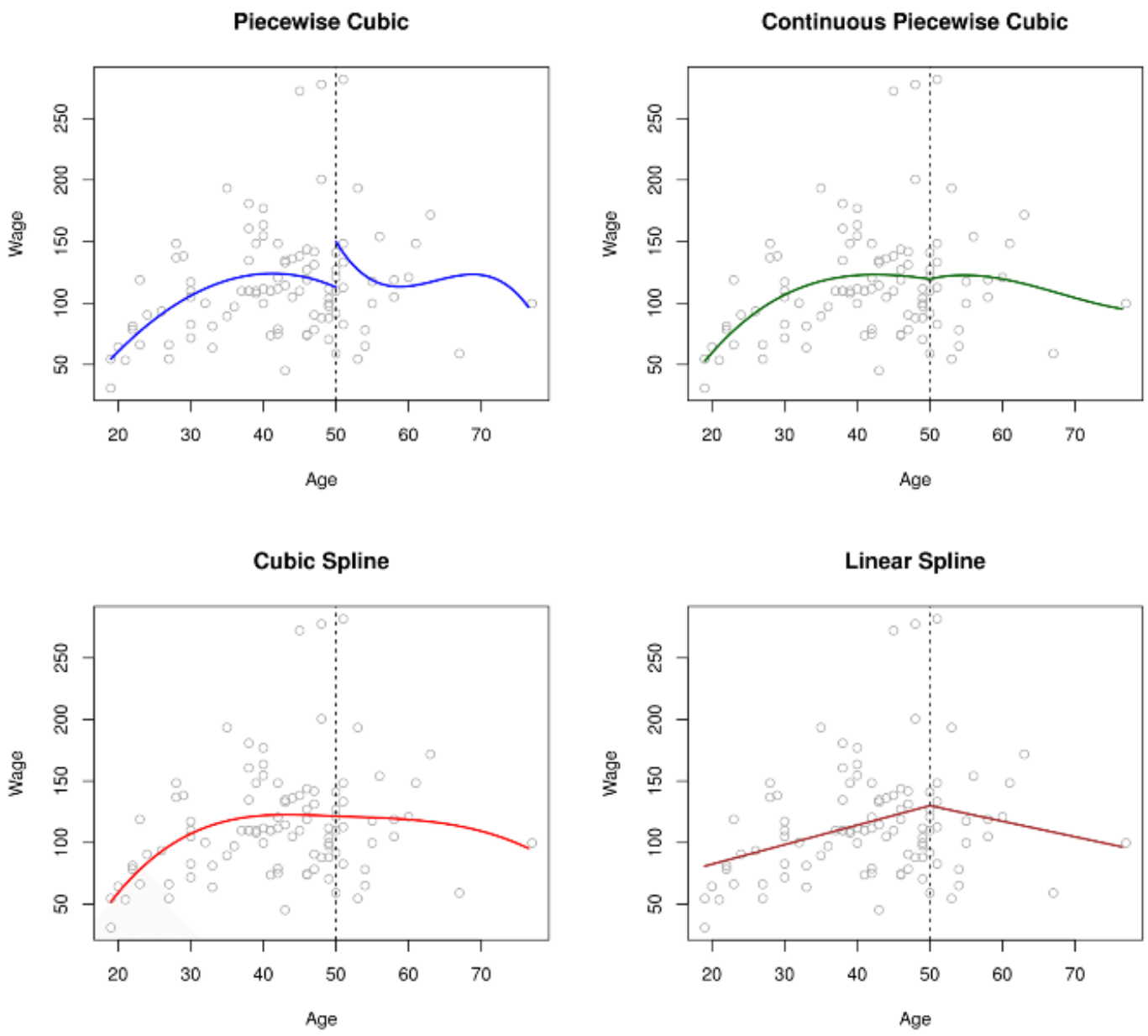
Il punto c viene definito *nodo* (o *knot*)

L'utilizzo di più *nodi* aumenta la flessibilità del modello. Se vi sono K nodi si stimano $K + 1$ polinomi.

Ad esempio, se $K = 1$ e si utilizzano polinomi cubici, si stimano in pratica $4 + 4$ parametri. In altro modo possiamo dire che si utilizzano 8 *gradi di libertà* per stimare i parametri.

In totale i gradi di libertà disponibili sono n, il numero di osservazioni.

Si noti che per quanto fatto finora, i polinomi stimati non sono in alcun modo legati tra loro (si veda il grafico seguente).



E' necessario l'utilizzo di vincoli per ottenere funzioni continue. L'introduzione dei vincoli ha inoltre il vantaggio di ridurre il numero di gradi di libertà utilizzati.

Più vincoli (sulla funzione e sulle sue derivate) possono essere introdotti per ottenere diversi gradi di continuità dei polinomi

La base di splines

Dal punto di vista pratico, una regressione spline può essere molto semplicemente rappresentata attraverso basi di funzioni.

Nel caso di una regressione spline con polinomi di primo grado e K nodi si ha

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots + \beta_{K+1} b_{K+1}(x_i) + \varepsilon_i$$

In questo caso $b_1(x) = x$ e le successive funzioni sono utilizzate per indicare i nodi

Le funzioni da usare per indicare i nodi sono funzioni troncate, ossia, nel caso lineare

$$h(x, c) = (x - c)_+ = \begin{cases} (x - c) & \text{se } x > c \\ 0 & \text{se } x \leq c \end{cases}$$

Se vi sono K nodi, $b_2(x_i) = h(x_i, c_1), \ldots b_{K+1} = h(x_i, c_K)$

La base di funzioni per una spline lineare con K nodi è dunque formata da

$$x, h(x, c_1), \ldots h(x, c_K)$$

In totale vi sono $K + 2$ parametri da stimare. Ossia il modello utilizza $K + 2$ gradi di libertà.

Esempio: spline lineare con un nodo

- In questo caso la tecnica spline richiede di stimae con OLS la funzione

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 (x_i - c)_+ + \varepsilon$$

- Se $x_i \leq c$ allora $(x_i - c)_+ = 0$ e

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon$$

- Se $x > c$ allora $(x_i - c)_+ = (x_i - c)$ e

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_i + \beta_2 (x_i - c) + \varepsilon \\ &= \beta_0 - \beta_2 c + (\beta_1 + \beta_2) x_i + \varepsilon \end{aligned}$$

- Si noti che le due rette hanno lo stesso valore $\beta_0 + \beta_1 c$ se $x = c$.

Regressione spline cubica con K nodi

In questo caso la regressione diventa

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

le funzioni di base sono

$$x, x^2, x^3, h(x, c_1), \ldots, h(x, c_K)$$

dove, in questo caso

$$h(x, c) = (x - c)_+^3 = \begin{cases} (x - c)^3 & \text{se } x > c \\ 0 & \text{se } x \leq c \end{cases}$$

In totale il modello utilizza $K + 4$ gradi di libertà

Esempio: *Temp*

Adattiamo una spline lineare ed una cubica con i nodi $c_1 = 59$ e $c_2 = 78$ ai dati Temp.

Confrontiamo i due modelli con un test F parziale. Si noti che si utilizza la funzione `lm()` utilizzando le spline `bs()`

```
library(splines)
fit=lm(Peak~bs(Tmax,knots=c(59,78),degree=1),data=Temp)
pred=predict(fit,newdata=list(Tmax=T.grid),se=T)
summary(fit)
```

```

Call:
lm(formula = Peak ~ bs(Tmax, knots = c(59, 78), degree = 1),
    data = Temp)

Residuals:
    Min      1Q  Median      3Q     Max
-592.79 -121.82    5.54  105.20  609.88

Coefficients:
                                       Estimate Std. Error t value
(Intercept)                             4423.13      37.56 117.760
bs(Tmax, knots = c(59, 78), degree = 1)1 -1668.40      53.19 -31.364
bs(Tmax, knots = c(59, 78), degree = 1)2 -1671.61      39.77 -42.034
bs(Tmax, knots = c(59, 78), degree = 1)3  -245.64      73.95  -3.322
                                       Pr(>|t|)
(Intercept)                             < 2e-16 ***
bs(Tmax, knots = c(59, 78), degree = 1)1  < 2e-16 ***
bs(Tmax, knots = c(59, 78), degree = 1)2  < 2e-16 ***
bs(Tmax, knots = c(59, 78), degree = 1)3 0.000986 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 189.3 on 361 degrees of freedom
Multiple R-squared:  0.844, Adjusted R-squared:  0.8427
F-statistic: 651.1 on 3 and 361 DF,  p-value: < 2.2e-16
```

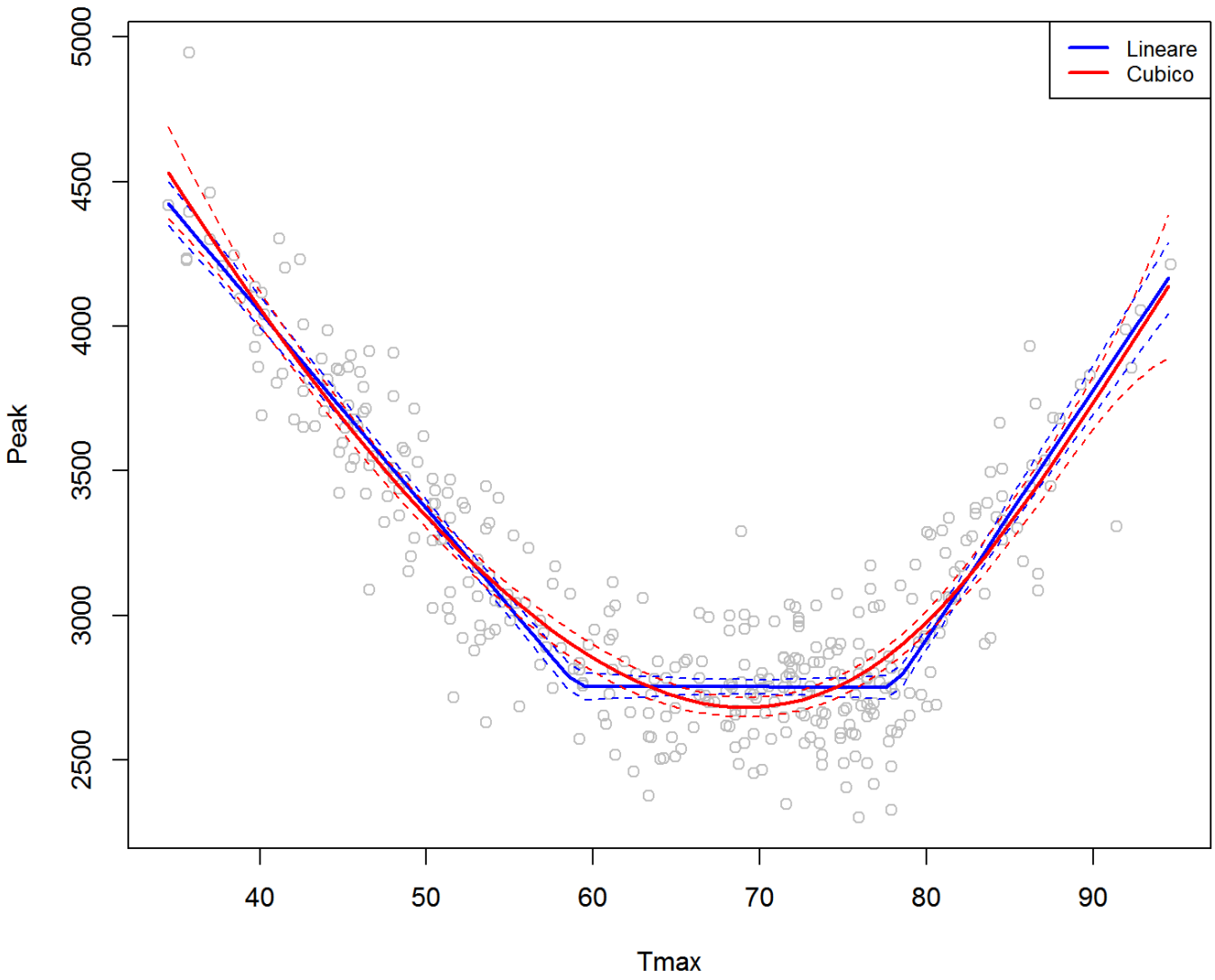
Anova

```
fit2=lm(Peak~bs(Tmax,knots=c(59,78),degree=3),data=Temp)
pred2=predict(fit2,newdata=list(Tmax=T.grid),se=T)
```

```
anova(fit,fit2)
```

```
Analysis of Variance Table

Model 1: Peak ~ bs(Tmax, knots = c(59, 78), degree = 1)
Model 2: Peak ~ bs(Tmax, knots = c(59, 78), degree = 3)
  Res.Df      RSS Df Sum of Sq F Pr(>F)
1    361 12934642
2    359 13592816  2   -658174
```

Plot splines lineare e cubica



Spline naturali

Le spline, se di grado superiore al secondo, hanno elevata variabilità nelle regioni estreme.

Una *spline naturale* è una spline con vincoli aggiuntivi di linearità nelle regioni estreme.

Per i dati *Temp* proviamo a stimare una spline cubica ed una spline naturale cubica

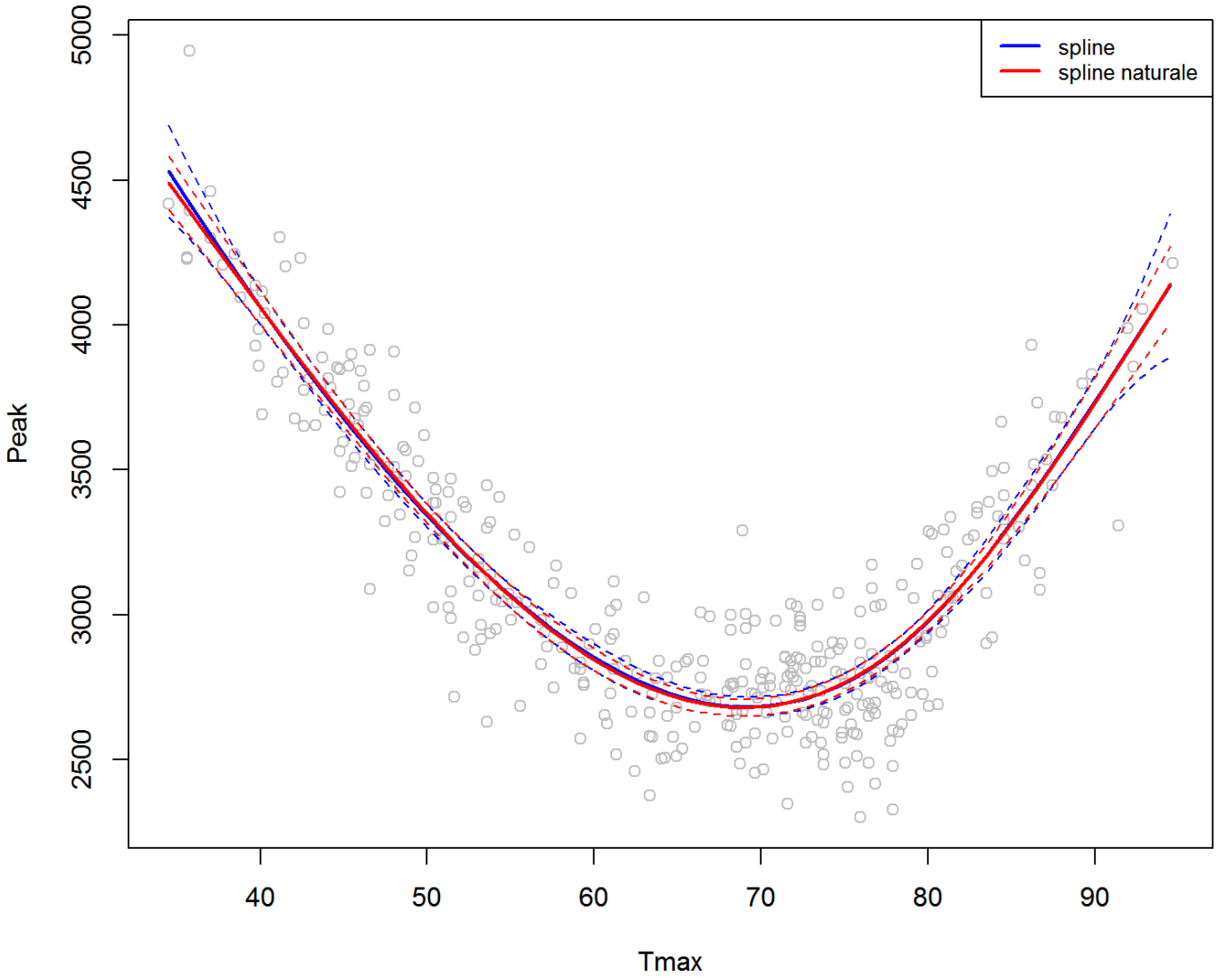
```
fit=lm(Peak~bs(Tmax,knots=c(59,78)),data=Temp)
pred=predict(fit,newdata=list(Tmax=T.grid),se=T)
```

```
fit2=lm(Peak~ns(Tmax,knots=c(59,78)),data=Temp)
pred2=predict(fit2,newdata=list(Tmax=T.grid),se=T)
```

```
anova(fit2,fit)
```

```
Analysis of Variance Table

Model 1: Peak ~ ns(Tmax, knots = c(59, 78))
Model 2: Peak ~ bs(Tmax, knots = c(59, 78))
  Res.Df      RSS Df Sum of Sq      F Pr(>F)
1    361 13607213
2    359 13592816  2     14397 0.1901 0.8269
```

Confronto grafico



Si noti che la spline naturale ha intervalli di confidenza meno ampi agli estremi

Smoothing splines

Le spline richiedono di specificare una sequenza di nodi: spesso questi sono scelti dal software in modo uniforme sul campo di variazione di X sulla base dei gradi di libertà stabiliti dall'utente.

Un modo alternativo di procedere è quello di imporre un vincolo di variazione più o meno lenta (smoothness) ad una generica funzione interpolante

Il metodo prende il nome di *smoothing splines*

Aspetti formali

Il metodo delle smoothing splines determina un funzione interpolante g che minimizza

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

Si noti che

- $\sum_{i=1}^{n}(y_i - g(x_i))^2 = RSS$
- $g''(t)$ è la derivata seconda
- $\int g''(t)^2 dt$ è una misura del cambiamento totale di g, determina una penalità sulla variabilità di g
- λ un parametro di *shrinkage* che calibra la penalità; da determinare con cross-validazione

Soluzione per g

- La soluzione generale per g è una spline cubica naturale con nodi nei punti $x_1, \ldots x_n$. λ decide l'effettivo numero di nodi da usare.
- Si dimostra che, per λ che cresce da 0 a ∞ gli effettivi gradi di libertà della spline decrescono da n a 2 (la retta, la funzione con minor variabilità possibile)

Tipicamente λ è scelto con LOOCV che è possibile calcoare in modo molto efficiente per le *smoothing splines*

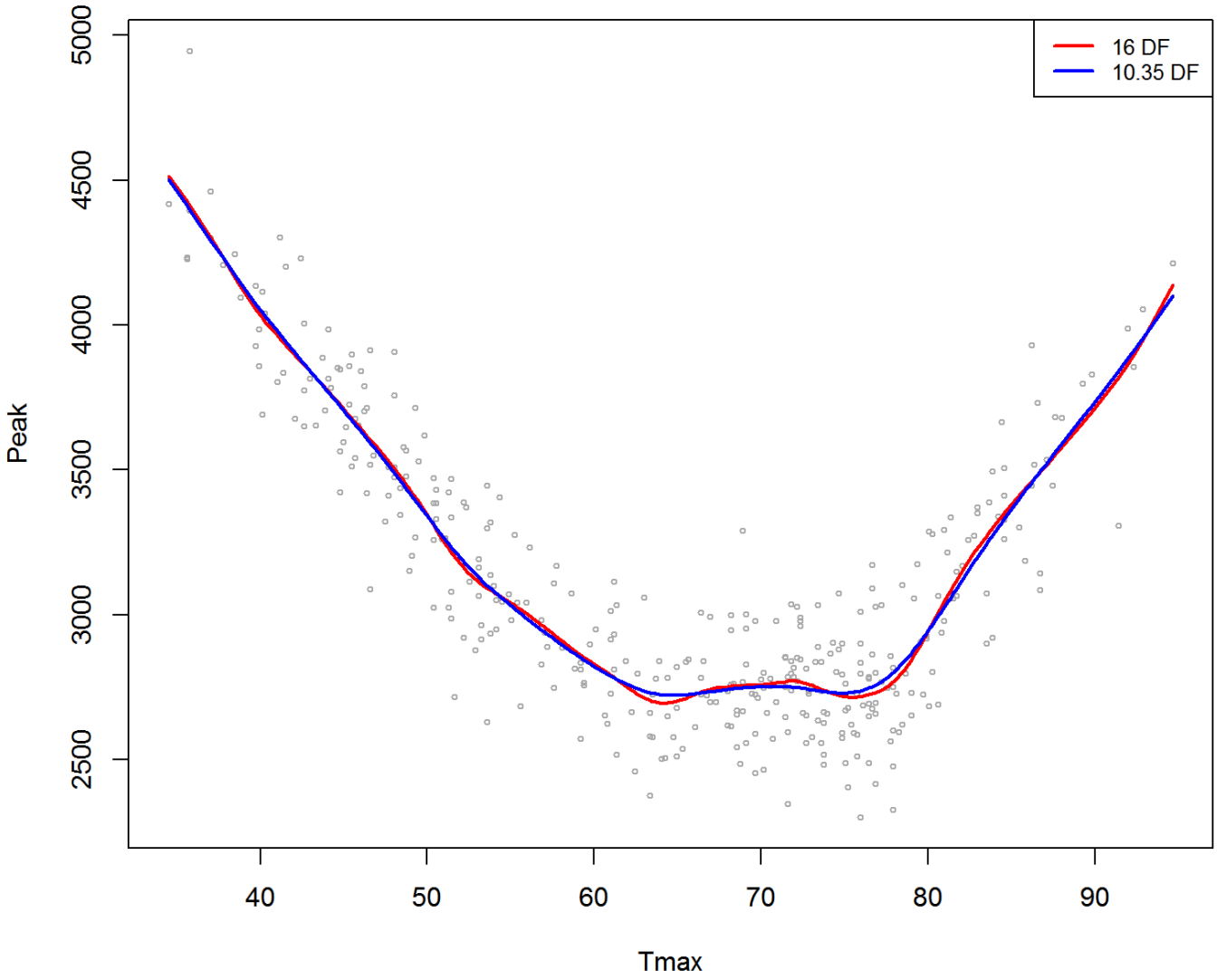
Dati *Temp*

Utilizziamo la funzione smooth.spline() con i dati *Temp* con 16 gradi di libertà e con gradi di libertà effettivi scelti attraverso LOOCV su λ

```
fit=smooth.spline(Tmax,Peak,df=16)
fit2=smooth.spline(Tmax,Peak,cv=TRUE)
fit2$df
```

```
[1] 10.34988
```

Smoothing splines



Regressione locale

Regressione locale a $X = x_0$

1. considera una frazione $s = k/n$ di punti x_i più vicini a x_0
2. assegna peso $K_{i0} = K(x_i, x_0)$ a ogni punto del vicinato di x_0. I pesi decrescono all'aumentare della distanza tra x_i e x_0. Tutti i punti al difuori del vicinato scelto hanno peso 0
3. determina $\hat{\beta}_0$, $\hat{\beta}_1$ minimizzando

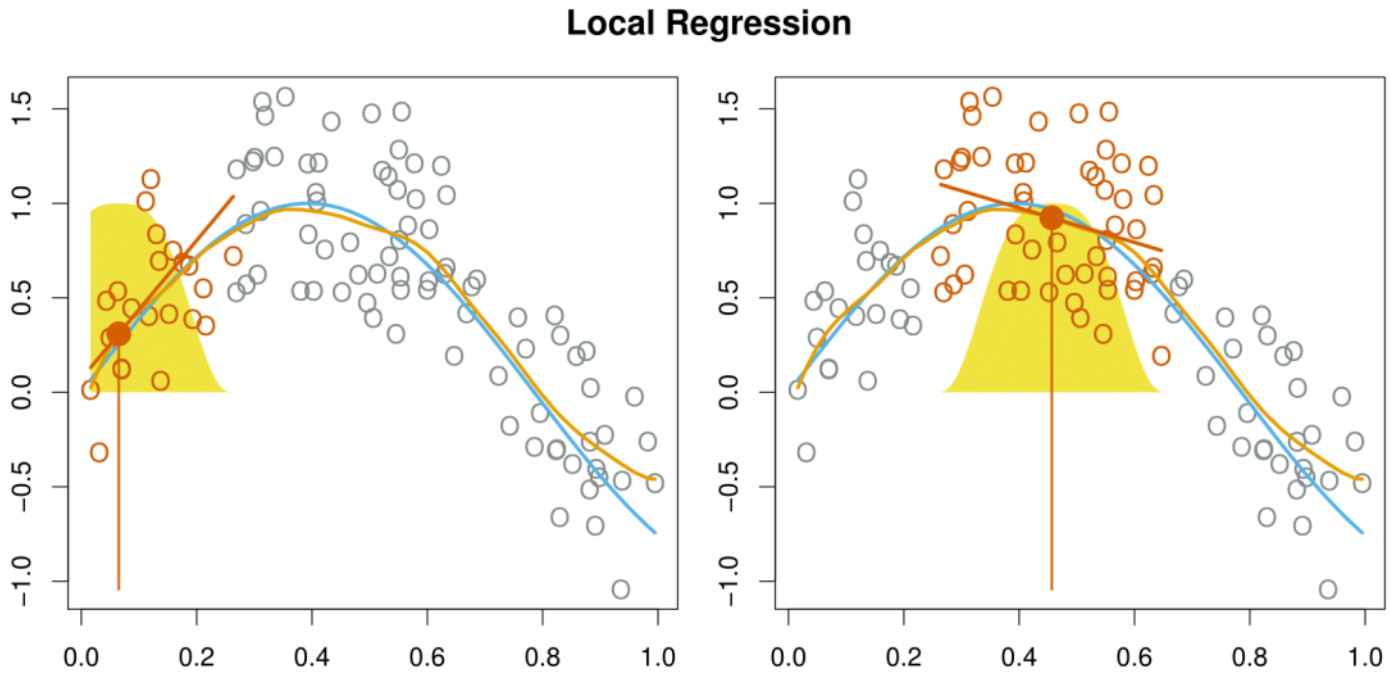
$$\sum_{i=1}^{n} K_{i0}(y_i - \beta_0 - \beta_1 x_i)^2$$

   (minimi quadrati pesati)

4. il valore $\hat{y}_0$ a x_0 è dato da

$$\hat{f}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$$

Interpretazione grafica



La regressione locale interpola localmente i dati stimando delle regressioni lineari ad ogni punto x_0 dei dati training basandosi su un certo numero di "vicini"

Richiede la scelta di due parametri: il peso K e la "finestra" o "vicinato" s.

La scelta della finestra s è particolarmente importante poiché controlla la flessibilità del modello: tanto più piccolo s tanto maggiore la flessibilità del modello. E' possibile specificare direttamente s o sceglierlo con cross-validazione.

La regressione locale può essere fatta con costanti, regressioni lineari o quadratiche

Si possono adattare regressioni locali su più predittori contemporaneamente

La regressione locale soffre del problema della dimensionalità (*curse of dimensionality*): se p è elevato ci possono essere in pratica pochi vicini ad un dato punto x_0

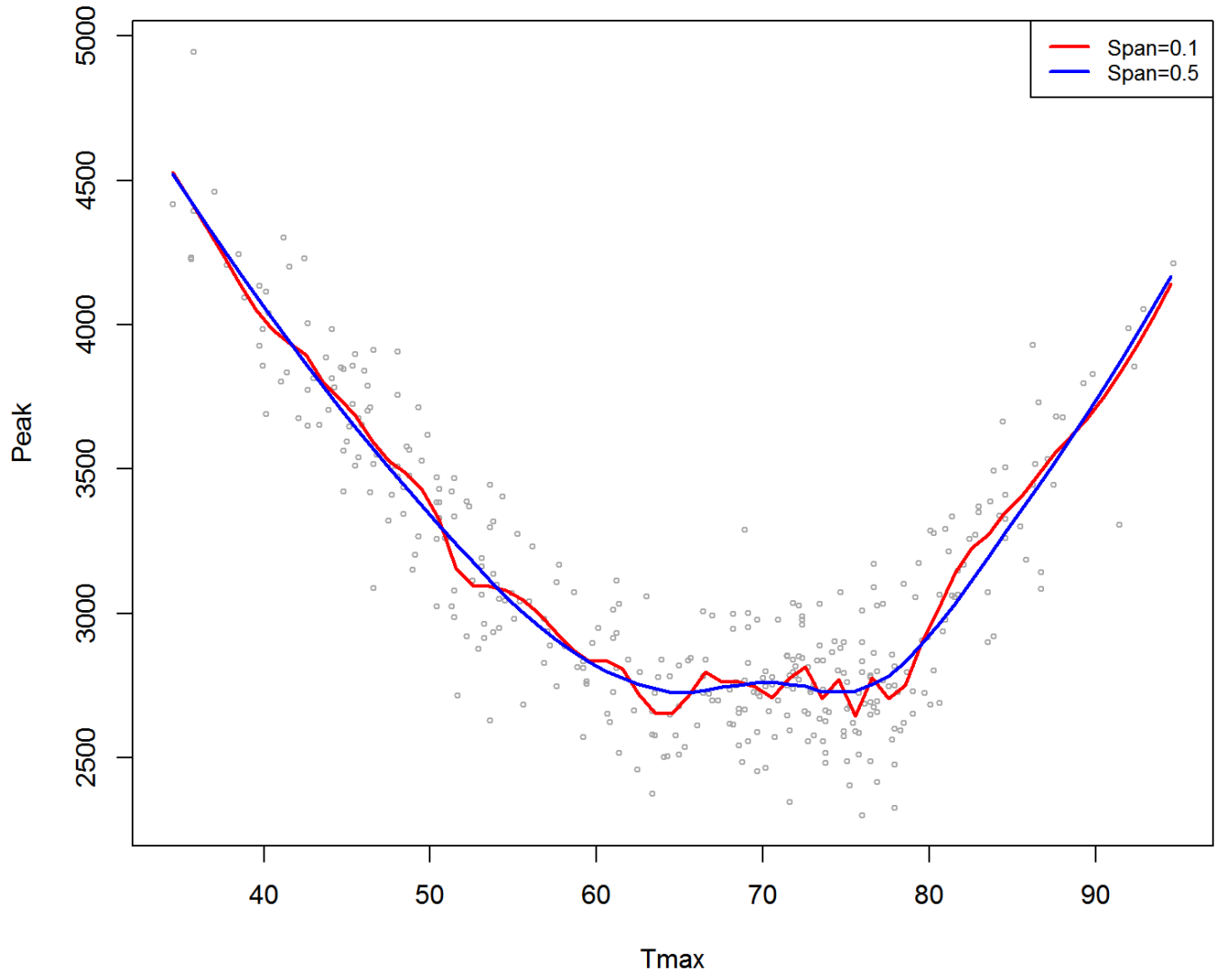
Dati *Temp*

La regressione locale si può effettuare attraverso la funzione `loess()`. La scelta di s è effettuata tramite l'opzione `span`

```
fit=loess(Peak~Tmax,span=.1,data=Temp)
fit2=loess(Peak~Tmax,span=.5,data=Temp)
summary(fit)
```

```
Call:
loess(formula = Peak ~ Tmax, data = Temp, span = 0.1)

Number of Observations: 365
Equivalent Number of Parameters: 30.37
Residual Standard Error: 188.3
Trace of smoother matrix: 33.59

Control settings:
  normalize:  TRUE
  span      :  0.1
  degree    :  2
  family    :  gaussian
  surface   :  interpolate     cell = 0.2
```

Confronto grafico



Modelli additivi generalizzati (*GAM*)

Finora abbiamo visto una serie di tecniche applicate a regressioni semplici. Proviamo a metterle assieme e consideriamo il caso generale di p predittori

I *GAM* forniscono un framework generale per l'estensione di modello lineare standard attraverso l'uso di funzioni non lineari di ciascuna delle variabili, mantenendo l'additività.

I GAM possono essere applicati sia a probelmi di regressione che di classificazione

I modelli GAM estendono il modello di regressione lineare

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i$$

sostituendo il temine $\beta_j x_{ij}$ con una generica funzione nonlineare $f_j(x_{ij})$:

$$y_i = \beta_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_p(x_{ip}) + \varepsilon_i$$

I GAM sono definiti additivi poiché calcolano funzioni separate per ciascun predittore e poi ne effettuano la somma

I diversi metodi discussi finora possono essere combinati in un GAM con diverse scelte per le singole f_j. Ad esempio possiamo combinare spline naturali con predittori lineari e costanti.

Pro e contro dei GAM

- I GAM consentono di adattare una f_j non lineare per ogni X_j: possiamo modellare automaticamente relazioni non lineari. Dal punto di vista pratico evita di provare singole trasformazioni su ogni variabile come tradizionalmente fatto nei modelli di regressione lineare
- L'adattamento non lineare è potenzialmente più accurato
- I GAM permettono di analizzare agevolmente la struttura dei fenomeni: poiché il modello è additivo, possiamo ancora esaminare l'effetto di ogni X_j su Y mantenendo le altre variabili fisse.
- La flessibilità della funzione f_j è riassunta nei gradi di libertà: la funzione è tanto meno variabile tanti meno i gradi di libertà (2 gl per la retta, il caso più rigido)
- La principale limitazione dei GAM è data dal fatto che sono limitati a modelli additivi. Possiamo però aggiungere nuove variabili ottenute da interazioni tra i predittori disponibili o aggiungere funzioni a più variabili (es regressione locale o spline bidimensionali)

Esempio: gam.m1 - spline lineare + costante

Per i dati *Temp*, costruiamo un modello che utilizza una spline lineare con nodi 59 e 78 per *Tmax* e analizza l'effetto della variabile *Day*

Poiché questo,con una scelta appropriata di funzioni di base, è un modello di regressione lineare, lo possiamo stimareutilizzando la funzione `lm()` .

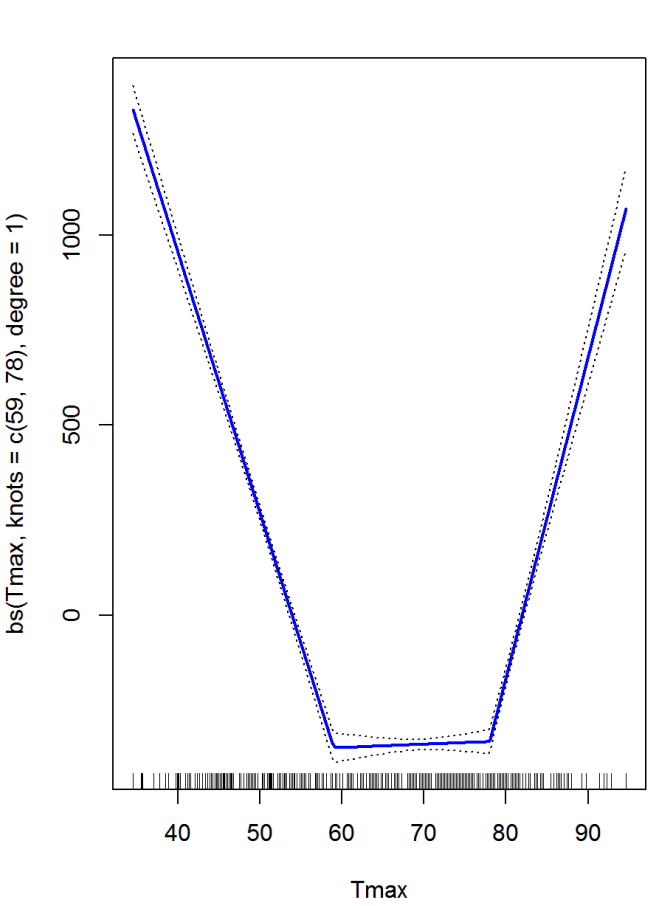
```
gam.m1=lm(Peak~bs(Tmax,knots=c(59,78),degree=1)+Day,data=Temp)
summary(gam.m1)
```

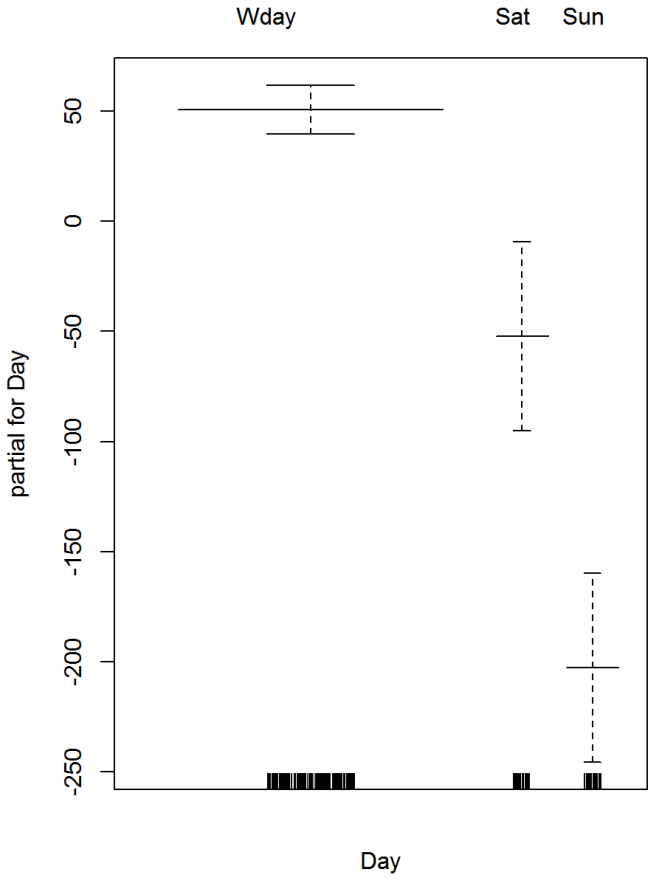
```
Call:
lm(formula = Peak ~ bs(Tmax, knots = c(59, 78), degree = 1) + 
    Day, data = Temp)

Residuals:
    Min      1Q  Median      3Q     Max 
-406.30 -102.10   -8.99  108.17  610.14 

Coefficients:
                                       Estimate Std. Error t value
(Intercept)                             4476.53      33.70 132.821
bs(Tmax, knots = c(59, 78), degree = 1)1 -1680.45      46.95 -35.794
bs(Tmax, knots = c(59, 78), degree = 1)2 -1664.16      35.07 -47.450
bs(Tmax, knots = c(59, 78), degree = 1)3  -261.88      65.24  -4.014
DaySat                                    -102.88      25.39  -4.051
DaySun                                    -253.35      25.38  -9.983
                                       Pr(>|t|)
(Intercept)                             < 2e-16 ***
bs(Tmax, knots = c(59, 78), degree = 1)1  < 2e-16 ***
bs(Tmax, knots = c(59, 78), degree = 1)2  < 2e-16 ***
bs(Tmax, knots = c(59, 78), degree = 1)3 7.27e-05 ***
DaySat                                   6.24e-05 ***
DaySun                                    < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 166.9 on 359 degrees of freedom
Multiple R-squared:  0.8794,    Adjusted R-squared:  0.8777 
F-statistic: 523.7 on 5 and 359 DF,  p-value: < 2.2e-16
```

Plot

Per il summary grafico dei risultati utilizziamo la funzione `plot.gam()` dalla libreria `gam`

```
library(gam)
par(mfrow=c(1,2))
plot.gam(gam.m1, se=TRUE, col="blue",lwd=2,cex=.8)
```





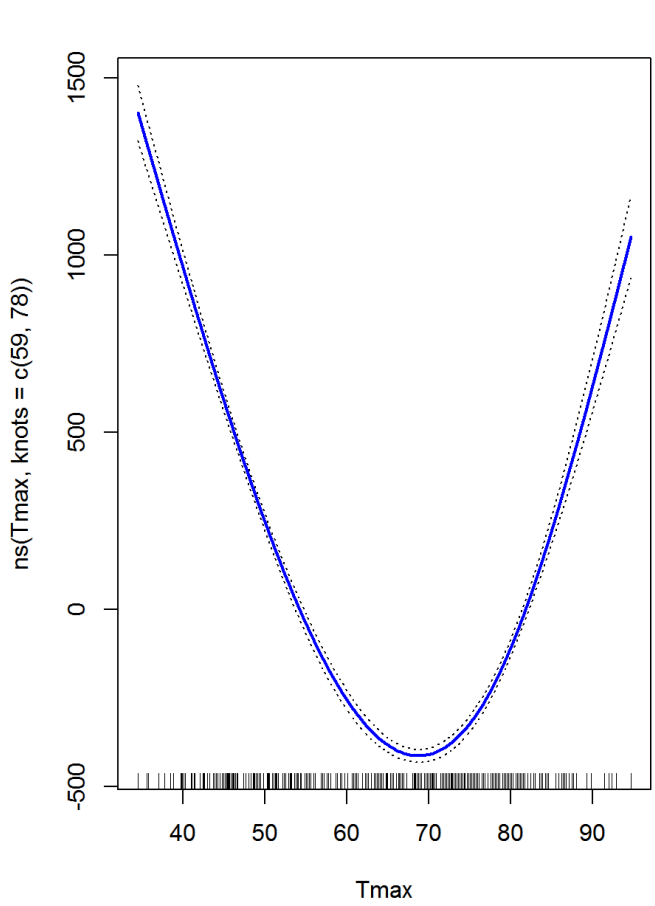
Esempio: gam.m2 - spline cubica naturale + costante

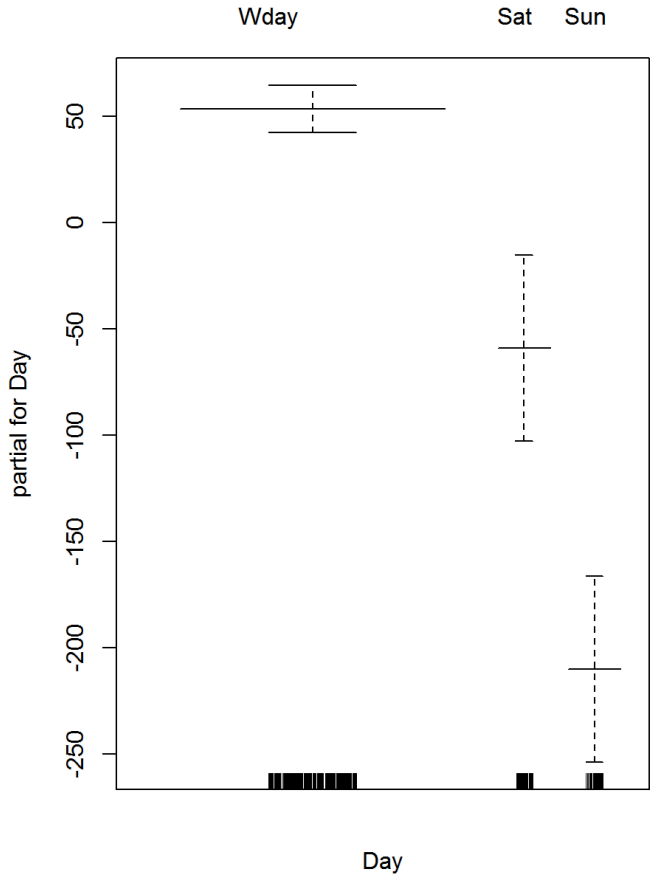
```
gam.m2=lm(Peak~ns(Tmax,knots=c(59,78))+Day,data=Temp)
summary(gam.m2)
```

```
Call:
lm(formula = Peak ~ ns(Tmax, knots = c(59, 78)) + Day, data = Temp)

Residuals:
    Min      1Q  Median      3Q     Max 
-472.61 -119.38    5.58  102.32  645.50 

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)    
(Intercept)                   4550.26      40.89 111.292  < 2e-16 ***
ns(Tmax, knots = c(59, 78))1 -1441.57      35.95 -40.095  < 2e-16 ***
ns(Tmax, knots = c(59, 78))2 -2397.38     108.60 -22.076  < 2e-16 ***
ns(Tmax, knots = c(59, 78))3   701.41      56.17  12.487  < 2e-16 ***
DaySat                        -112.51      25.90  -4.344 1.82e-05 ***
DaySun                        -263.74      25.85 -10.202  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 170.1 on 359 degrees of freedom
Multiple R-squared:  0.8747,    Adjusted R-squared:  0.873 
F-statistic: 501.3 on 5 and 359 DF,  p-value: < 2.2e-16
```

Plot

```
par(mfrow=c(1,2))
plot.gam(gam.m2, se=TRUE, col="blue",lwd=2,cex=.8)
```





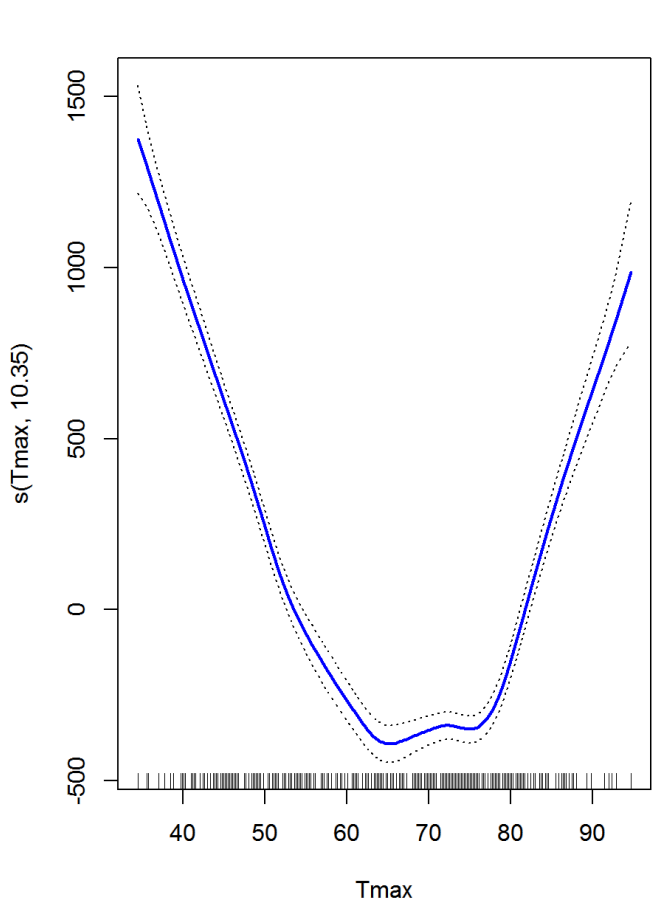
Esempio: gam.m3 - smoothing spline e costante

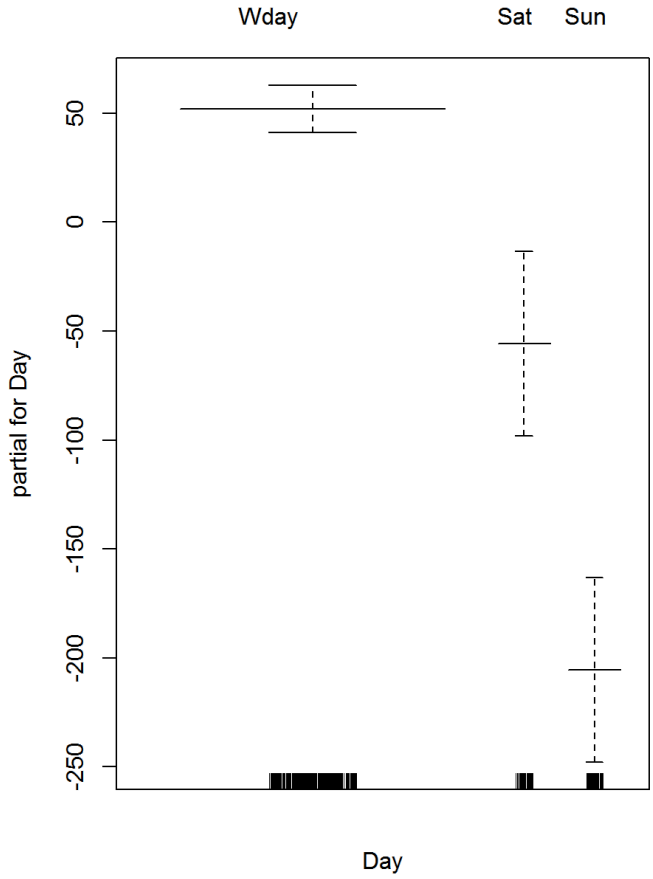
Per stimare GAM di tipo più generale, con spline smoothing o altri componenti che non possono essere espressi in termini di funzioni di base, abbiamo bisogno di usare la funzione `gam()` dall'omonima libreria

La funzione `s()`, che è parte della libreria `gam`, viene usata per indicare una smoothing spline. Scegliamo 10.35 gradi di libertà come ottenuto precedentemente da cross validazione

```
gam.m3=gam(Peak~s(Tmax,10.35)+Day,data=Temp)
```

```
par(mfrow=c(1,2))
plot(gam.m3, se=TRUE,col="blue",lwd=2,cex=.8)
```





Confronto gam.m1 gam.m3

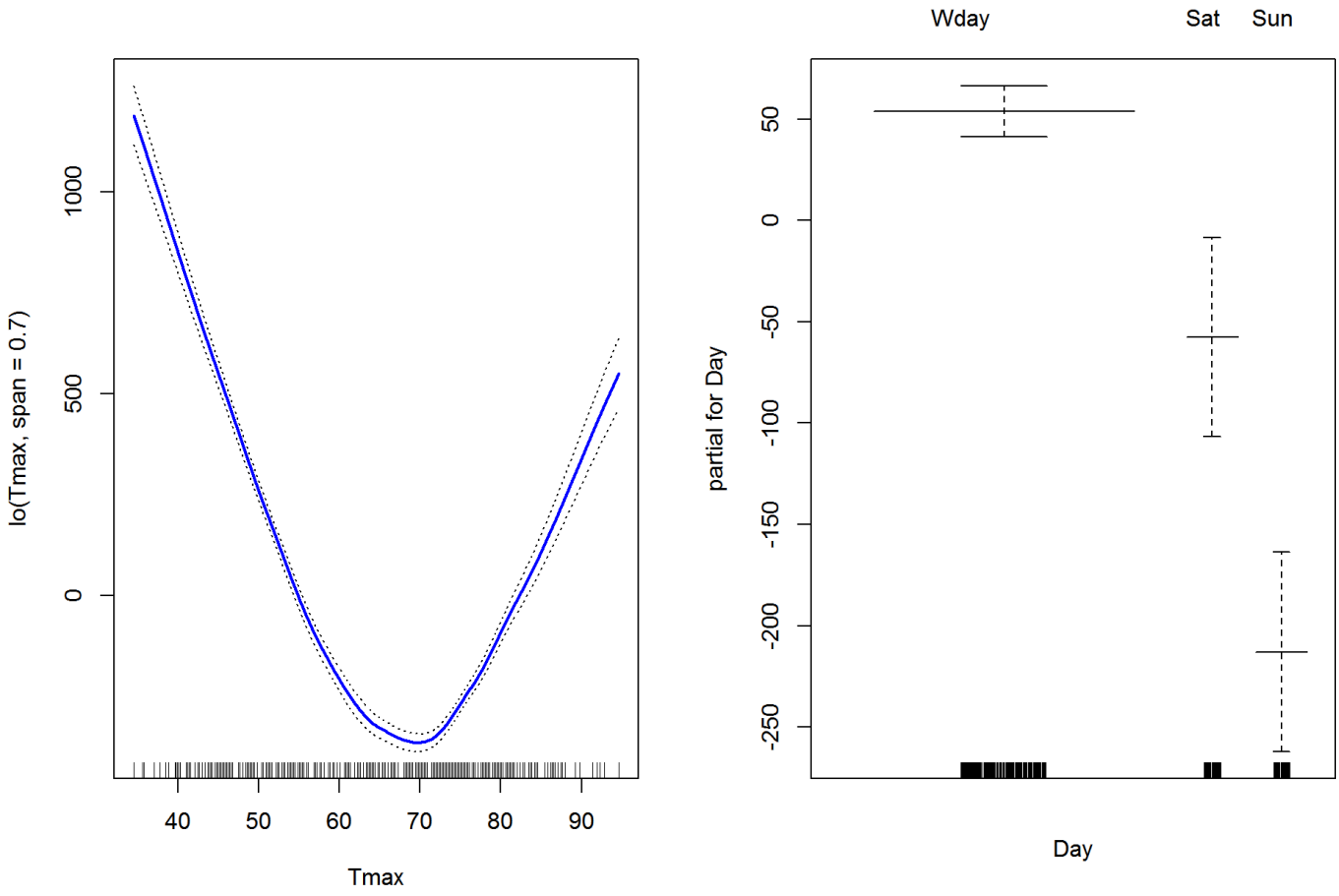
```
anova(gam.m1,gam.m3, test="F")
```

```
Analysis of Variance Table

Model 1: Peak ~ bs(Tmax, knots = c(59, 78), degree = 1) + Day
Model 2: Peak ~ s(Tmax, 10.35) + Day
  Res.Df     RSS   Df Sum of Sq      F  Pr(>F)
1 359.00 9998054
2 351.65 9574340 7.35    423715 2.1173 0.03833 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Esempio: gam.m4 - regressione locale (span=0.7) + costante

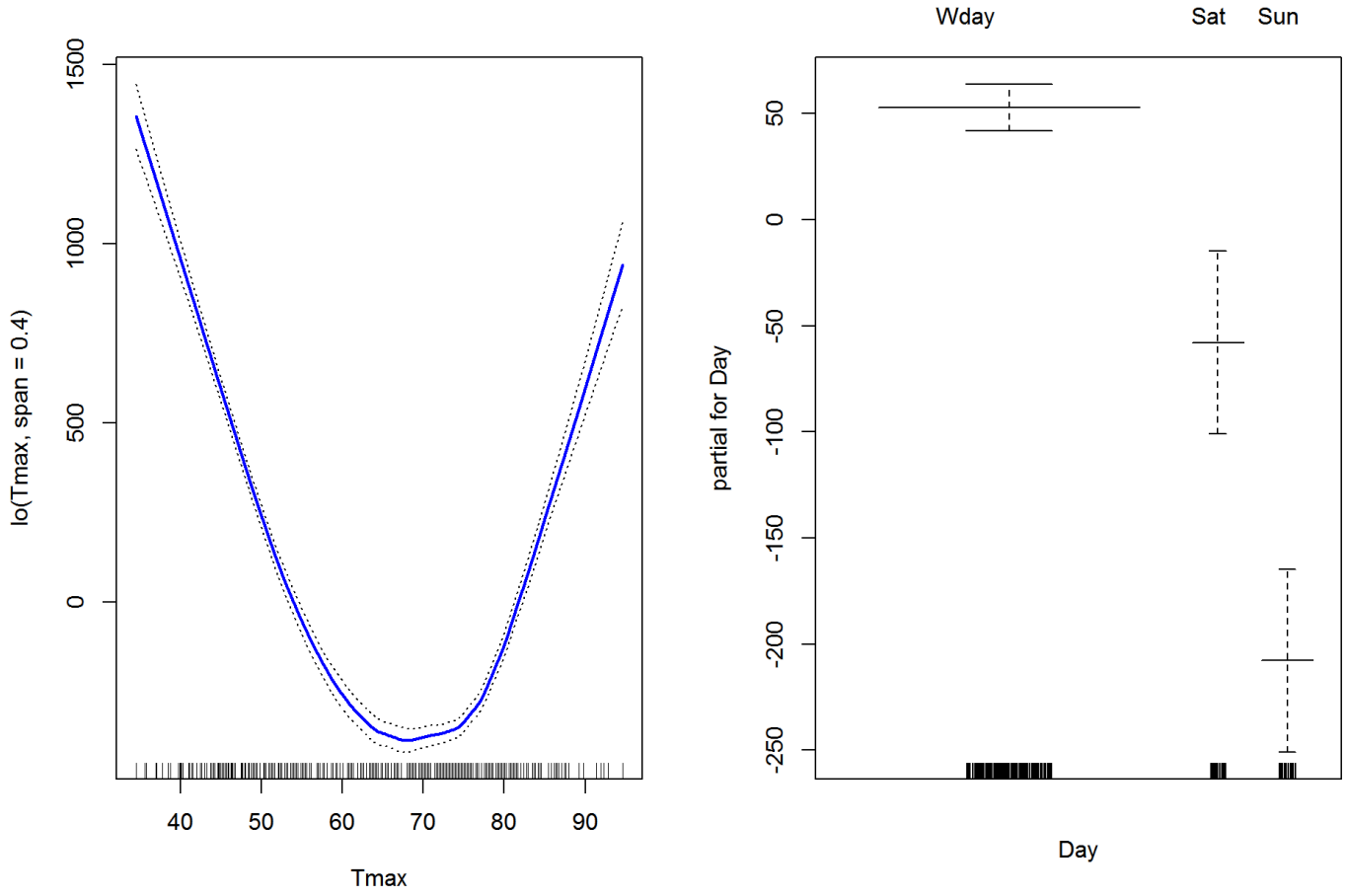
Possiamo anche usare la regressione locale utilizzando la funzione `lo()` .

```
gam.m4=gam(Peak~lo(Tmax,span=0.7)+Day,data=Temp)
par(mfrow=c(1,2))
plot.gam(gam.m4, se=TRUE, col="blue",lwd=2,cex=.8)
```



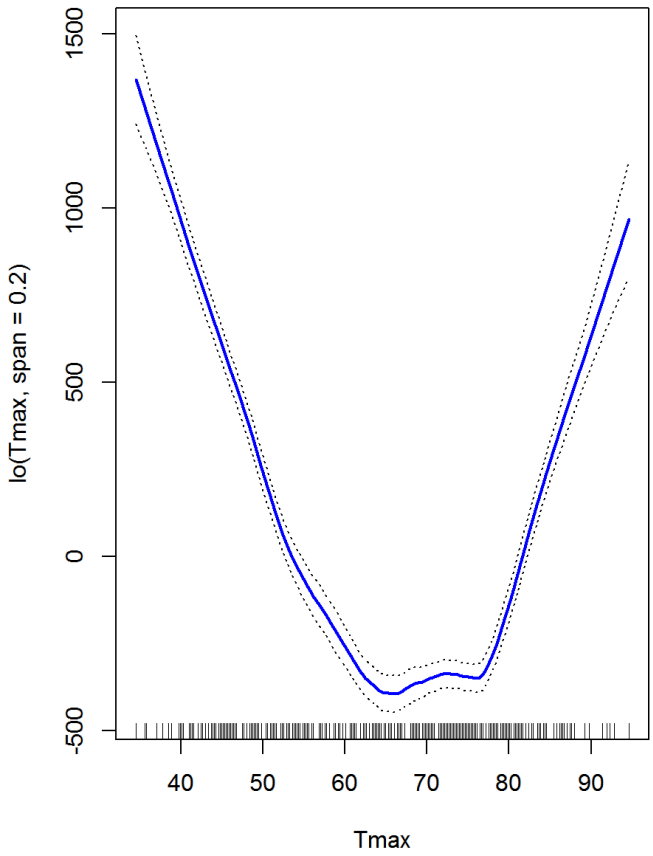
gam.m5 - regressione locale (span=0.4) + costante

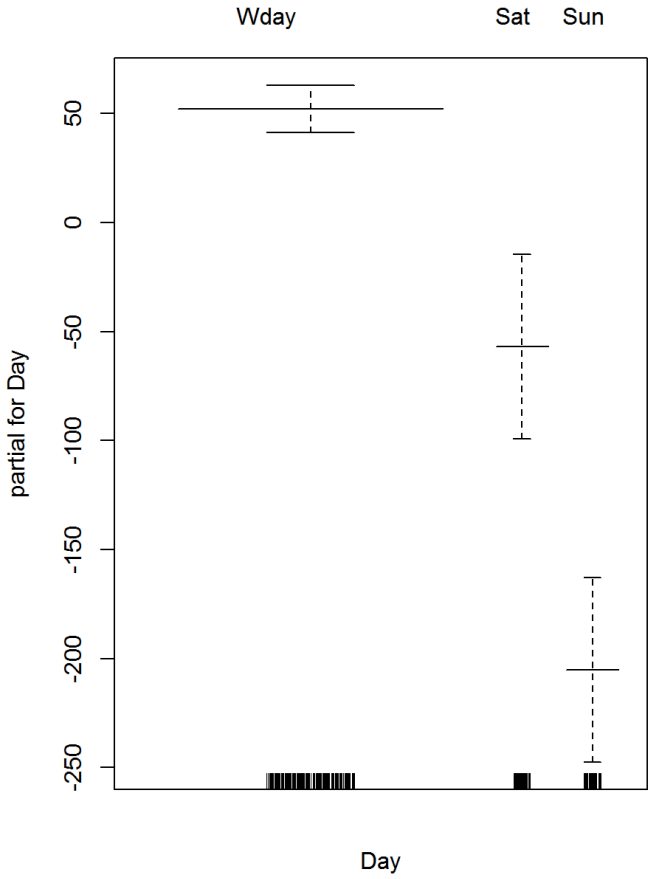
```
gam.m5=gam(Peak~lo(Tmax,span=0.4)+Day,data=Temp)
par(mfrow=c(1,2))
plot.gam(gam.m5, se=TRUE, col="blue",lwd=2,cex=.8)
```



gam.m6 - regressione locale (span=0.2) + costante

```
gam.m6=gam(Peak~lo(Tmax,span=0.2)+Day,data=Temp)
par(mfrow=c(1,2))
plot.gam(gam.m6, se=TRUE, col="blue",lwd=2,cex=.8)
```





Confronti

```
anova(gam.m4,gam.m5,gam.m6)
```

```
Analysis of Deviance Table

Model 1: Peak ~ lo(Tmax, span = 0.7) + Day
Model 2: Peak ~ lo(Tmax, span = 0.4) + Day
Model 3: Peak ~ lo(Tmax, span = 0.2) + Day
  Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
1    359.58   13172023
2    357.44   10067134 2.1410  3104890 < 2.2e-16 ***
3    352.50    9594384 4.9413   472750  0.003676 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Riferimenti bibliografici

An Introduction to Statistical Learning, with applications in R. (Springer, 2013)

Alcune delle figure in questa presentazione sono tratte dal testo con il permesso degli autori: G. James, D. Witten, T. Hastie e R. Tibshirani