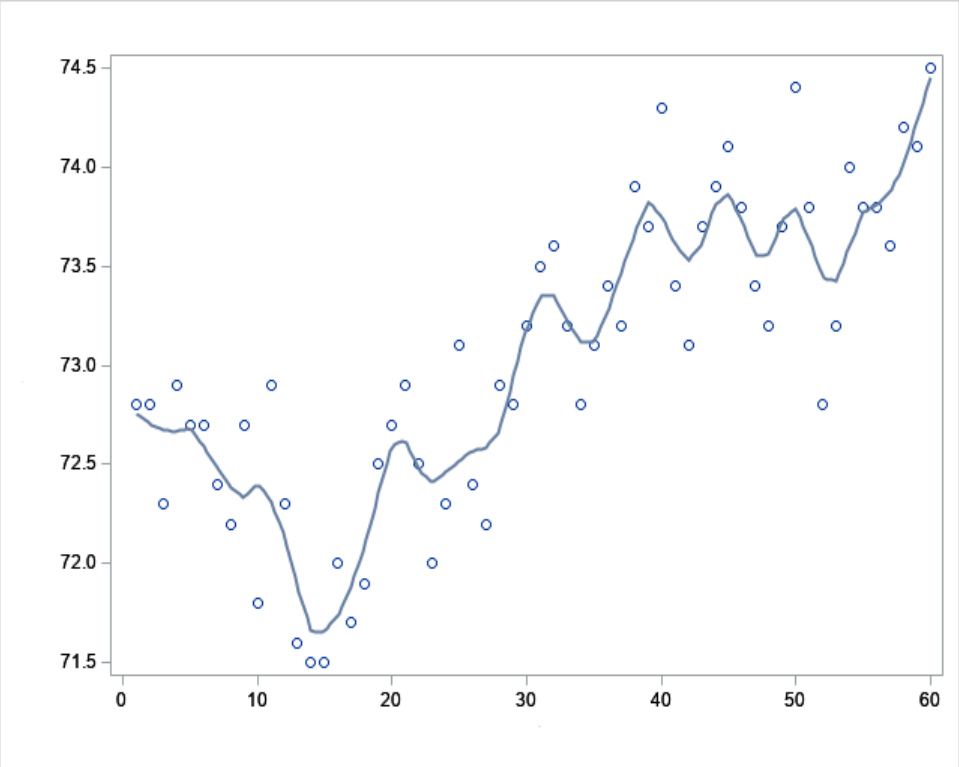
8.1 Nonparametric Regression for Continuous Response

Suppose the data contain a continuous response variable y and predictors $x_1, \ldots, x_k$, and we would like to fit a regression model. Sometimes the relationship is not linear, or polynomial, or exponential. If the data exhibit periodicity (seasonal variations, for example) then none of the traditional response functions from parametric world would have a satisfactory fit to these data.

A useful tool in this case is a nonparametric regression of the form $y = f(x_1, \ldots, x_k) + \varepsilon$ where f is a nonparametric response function with no explicit functional form, and ε's are independent and identically distributed random errors with a zero mean and a constant variance. No assumption is made on the form of the probability distribution of ε's.

As an illustration, consider the following scatterplot of a response y against a predictor x, and a fitted nonparametric curve. The pattern on the graph shows periodic spikes (sometimes termed *cycles*, *periodic variations* or *seasonal variations*). None of the traditional response functions from the parametric world would have a satisfactory fit to these data.



Definition. The **loess regression (locally estimated scatterplot smoothing) method** fits the response function $f(\cdot)$ and plots a curve (or surface in three or higher dimensions) on the scatterplot. In a loess regression, the estimation of the response function at the data points is done by fitting a polynomial regression function in the local neighborhood of each point. The radius of the neighborhoods is determined by a pre-specified fraction of data points, called **smoothing parameter**, that these neighborhoods encompass. A polynomial regression (linear or quadratic) is fit by the weighted least-squares method with less weight given to points more distant from the center.

Mathematically speaking, first the value of the smoothing parameter is specified as p/n, where n is the total number of data points and p is the number of points in the local neighborhood $\mathcal{N}_p(\mathbf{x}^0)$ of each point $\mathbf{x}^0 = (x_1^0, \ldots, x_k^0)$. Then a weighted linear regression is fitted through the p points in each local neighborhood $\mathcal{N}_p(\mathbf{x}^0)$. This linear function is defined as

$$l(\mathbf{x}) = l(x_1, \ldots, x_k) = \beta_0 + \beta_1(x_1 - x_1^0) + \cdots + \beta_k(x_k - x_k^0),$$

for every $\mathbf{x} = (x_1, \ldots, x_k)$ in $\mathcal{N}_p(\mathbf{x}^0)$.

The weighted least-squares predicted line $\hat{l}(\mathbf{x}) = \hat{\beta}_0 + \hat{\beta}_1(x_1 - x_1^0) + \cdots + \hat{\beta}_k(x_k - x_k^0)$ minimizes

$$\sum_{\mathbf{x}_i \in \mathcal{N}_p(\mathbf{x}^0)} \Big[y_i - l(\mathbf{x}_i)\Big]^2 w\Big(\frac{\|\mathbf{x}_i - \mathbf{x}^0\|}{r(\mathbf{x}^0)}\Big)$$

where $\mathbf{x}_i = (x_{1i}, \ldots, x_{ki}), i = 1, \ldots, p$. Here $\|\mathbf{x}_i - \mathbf{x}^0\|$ denotes the Euclidean distance, that is,

$$\|\mathbf{x}_i - \mathbf{x}^0\| = \sqrt{(x_{1i} - x_1^0)^2 + \cdots + (x_{ki} - x_k^0)^2},$$

$r(\mathbf{x}^0) = \max\limits_{\mathbf{x}_i \in \mathcal{N}_p(\mathbf{x}^0)} \|\mathbf{x}_i - \mathbf{x}^0\|$ is the radius of the neighborhood, and $w(\cdot)$ is the weight function.

Finally, the response function f is estimated at the point $\mathbf{x}^0$ by the corresponding value on the predicted line $\hat{l}$, that is, $\hat{f}(\mathbf{x}^0) = \hat{l}(\mathbf{x}^0) = \hat{\beta}_0$. Note that only the value of the fitted intercept comes into play.

Once the loess estimation of the response function at every data point is completed, the **loess curve** is drawn on a scatterplot by connecting the fitted points with straight lines.

If a considerable curvilinear relation is displayed on a scatterplot, a quadratic function

$$q(x_1,\ldots,x_k)=\beta_0+\beta_{1,1}(x_1-x_1^0)+\beta_{1,2}(x_1-x_1^0)^2+\cdots+\beta_{k,1}(x_k-x_k^0)+\beta_{k,2}(x_k-x_k^0)^2$$

is fitted instead of the linear function l .

As a weight function, SAS uses the **normalized tricube weight function** defined as

$$w(\mathbf{x})=\begin{cases}\frac{32}{5}\left(1-\|\mathbf{x}\|^3\right)^3, & \text{if } \|\mathbf{x}\|\leq 1,\\ 0, & \text{otherwise.}\end{cases}$$

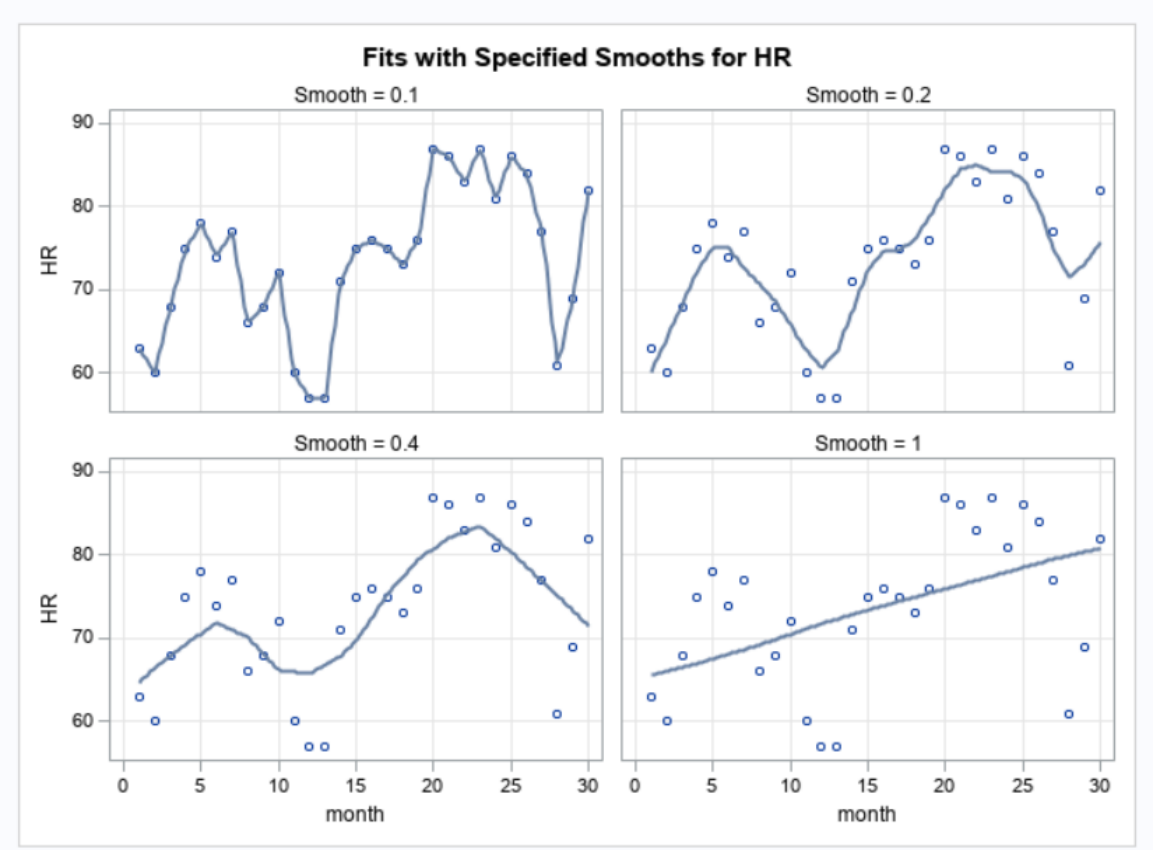
In R, the default weight function is the **tricube weight function**

$$w(\mathbf{x})=\begin{cases}\left(1-\|\mathbf{x}\|^3\right)^3, & \text{if } \|\mathbf{x}\|\leq 1,\\ 0, & \text{otherwise.}\end{cases}$$

Example (One Predictor). The data below contain average monthly heart rate (HR) measurements for one individual. We regress HR on month, using linear and quadratic loess regressions.

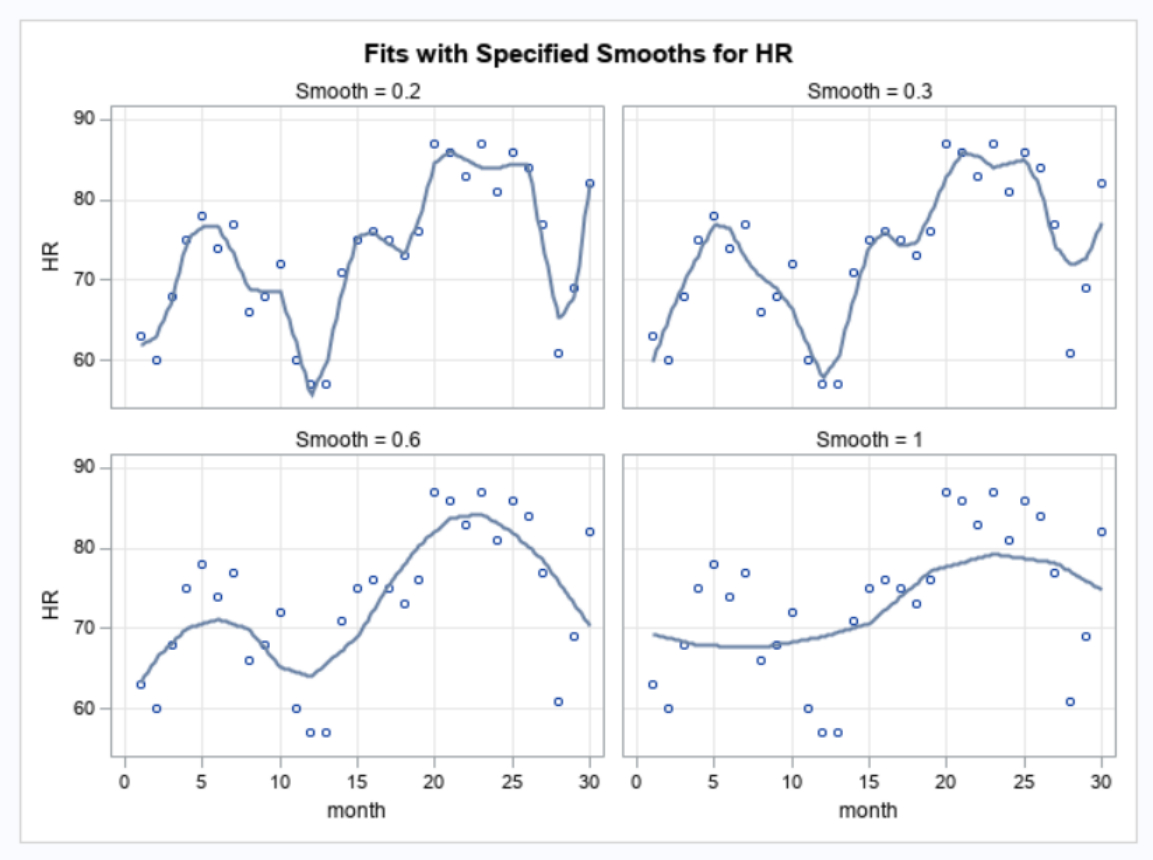
In SAS:

```
data heart_rate;
input HR @@;
 month=_N_;
 cards;
63 60 68 75 78 74 77 66 68 72
60 57 57 71 75 76 75 73 76 87
86 83 87 81 86 84 77 61 69 82
;

proc loess data=heart_rate;
 model HR = month / smooth= 0.1 0.2 0.4 1;
run;
```



By comparing these four graphs, we see that the one with the smoothing parameter of 0.2 has the best fit. It is neither too spiky nor too smooth.

```
proc loess data=heart_rate;
 model HR = month /degree=2 smooth=0.2 0.3 0.6 1;
run;
```



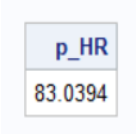
Of these four graphs, the one with the smoothing parameter 0.3 is the best.

```
/*prediction*/
data point4pred;
 input month;
cards;
20
;

proc loess data=heart_rate;
 model HR = month / smooth= 0.2;
  ods output ScoreResults=predicted;
    score data=point4pred;
run;

proc print data=predicted noobs;
var p_HR;
```

```
run;
```

p_HR
82.1224

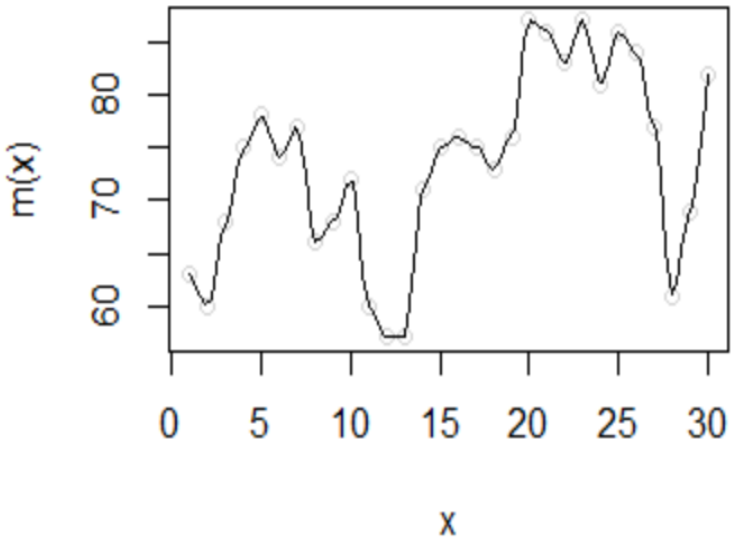
```
proc loess data=heart_rate;
 model HR = month /degree=2 smooth=0.3;
  ods output ScoreResults=predicted;
    score data=point4pred;
run;

proc print data=predicted noobs;
var p_HR;
run;
```



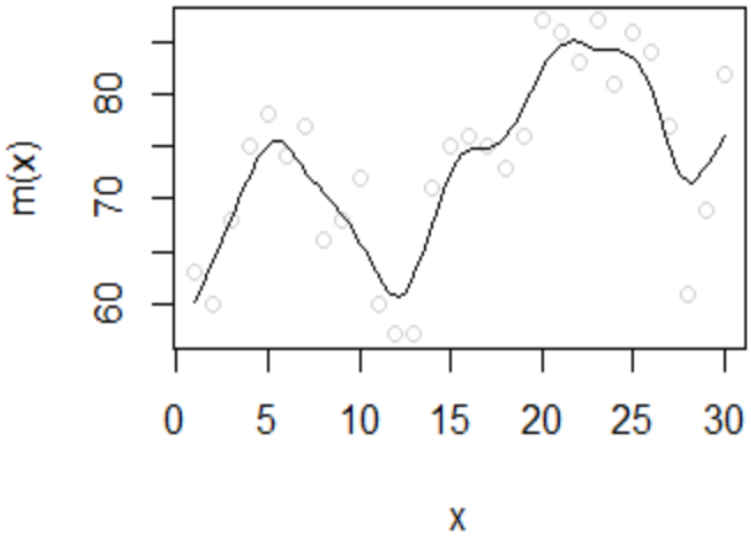
p_HR
83.0394

In R:
HR<- c(63, 60, 68, 75, 78, 74, 77, 66, 68, 72,
60, 57, 57, 71, 75, 76, 75, 73, 76, 87,
86, 83, 87, 81, 86, 84, 77, 61, 69, 82)
month <- 1:30

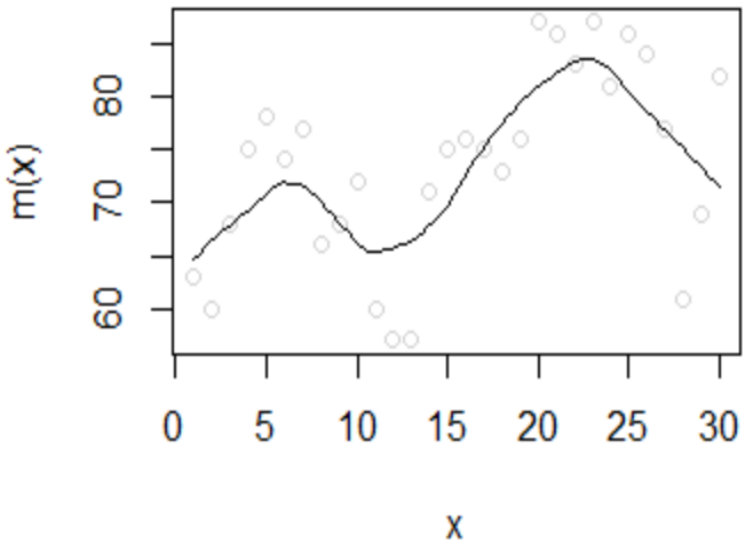
library("fANCOVA")
loess.as(month, HR, degree = 1, user.span = 0.1, plot=TRUE)



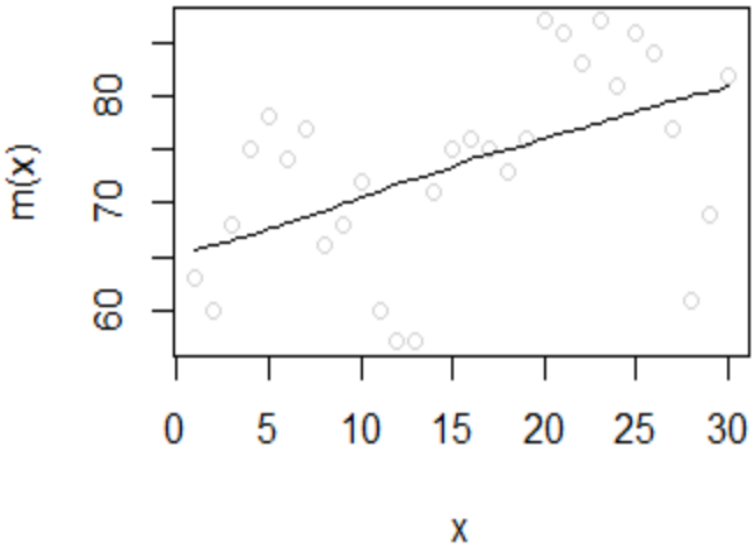
loess.as(month, HR, degree = 1, user.span = 0.2, plot=TRUE)



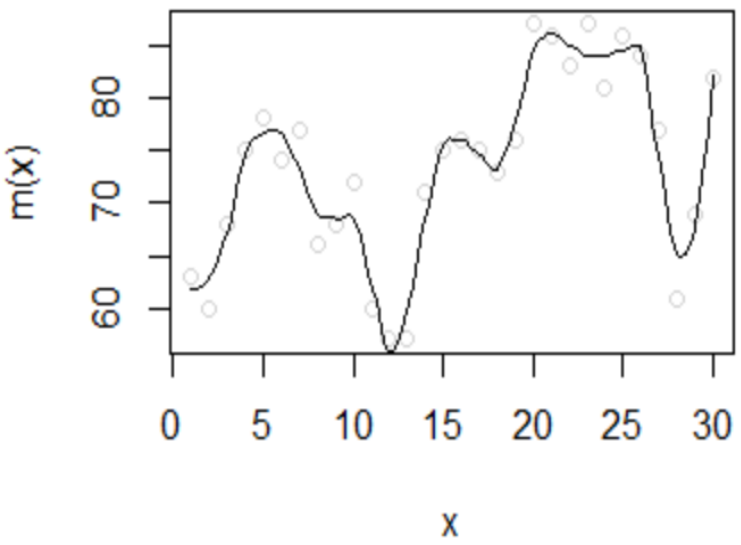
loess.as(month, HR, degree = 1, user.span = 0.4, plot=TRUE)



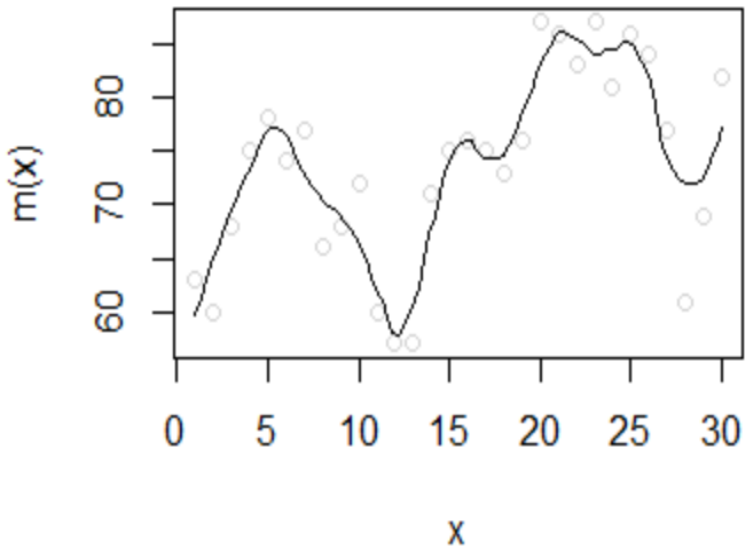
loess.as(month, HR, degree = 1, user.span = 1, plot=TRUE)



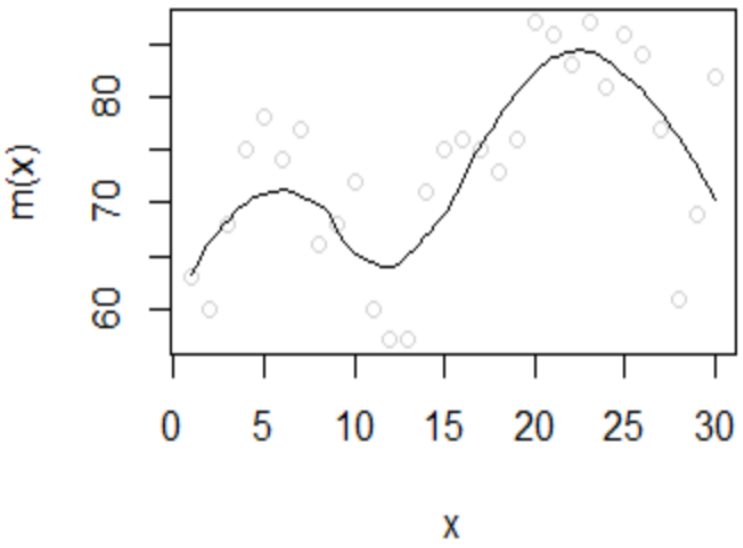
loess.as(month, HR, degree = 2, user.span = 0.2, plot=TRUE)



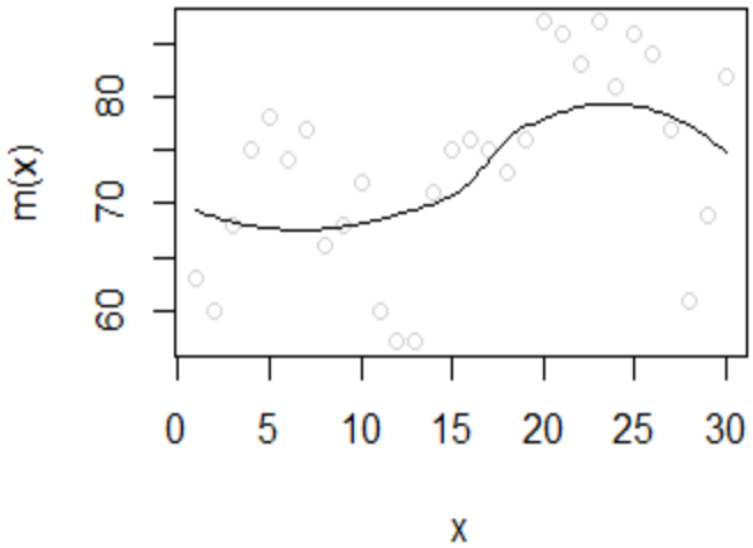
loess.as(month, HR, degree = 2, user.span = 0.3, plot=TRUE)



loess.as(month, HR, degree = 2, user.span = 0.6, plot=TRUE)



loess.as(month, HR, degree = 2, user.span = 1, plot=TRUE)



#prediction
loess.fit1<- loess.as(month, HR, degree = 1, user.span = 0.2)
predict(loess.fit1, 20)

```
82.1224
```

loess.fit2<- loess.as(month, HR, degree=2, user.span = 0.3)
predict(loess.fit2, 20)

```
83.03941
```

Example (Two Predictors). In this example, we regress LDL measurements on stress level and month, using the loess method. The data are from the previous example on random slope and intercept model with normally distributed response.

In SAS:

```
proc import out=ldldata datafile="./LDLdata.csv" dbms=csv replace;

/*creating long-form data*/
data longform;
 set ldldata;
   array m[4] (0 6 9 24);
    array s[4] stress0 stress6 stress9 stress24;
  array c[4] LDL0 LDL6 LDL9 LDL24;
       do i=1 to 4;
  month=m[i];
   stress=s[i];
        LDL=c[i];
     output;
    end;
keep id month stress LDL;
run;

/*fitting loess regression and plotting fitted loess surface*/

proc means data=longform min max;
var stress month;
run;
```

The MEANS Procedure

Variable	Minimum	Maximum
stress	9.0000000	36.0000000
month	0	24.0000000

```
data grid_points;
    do stress=9 to 36;
       do month=0 to 24;
           output;
       end;
    end;
run;

proc loess data=longform;
 model LDL=stress month;
  ods output scoreresults=score_results;
    score data=grid_points;
run;

proc print data=score_results;
run;
```

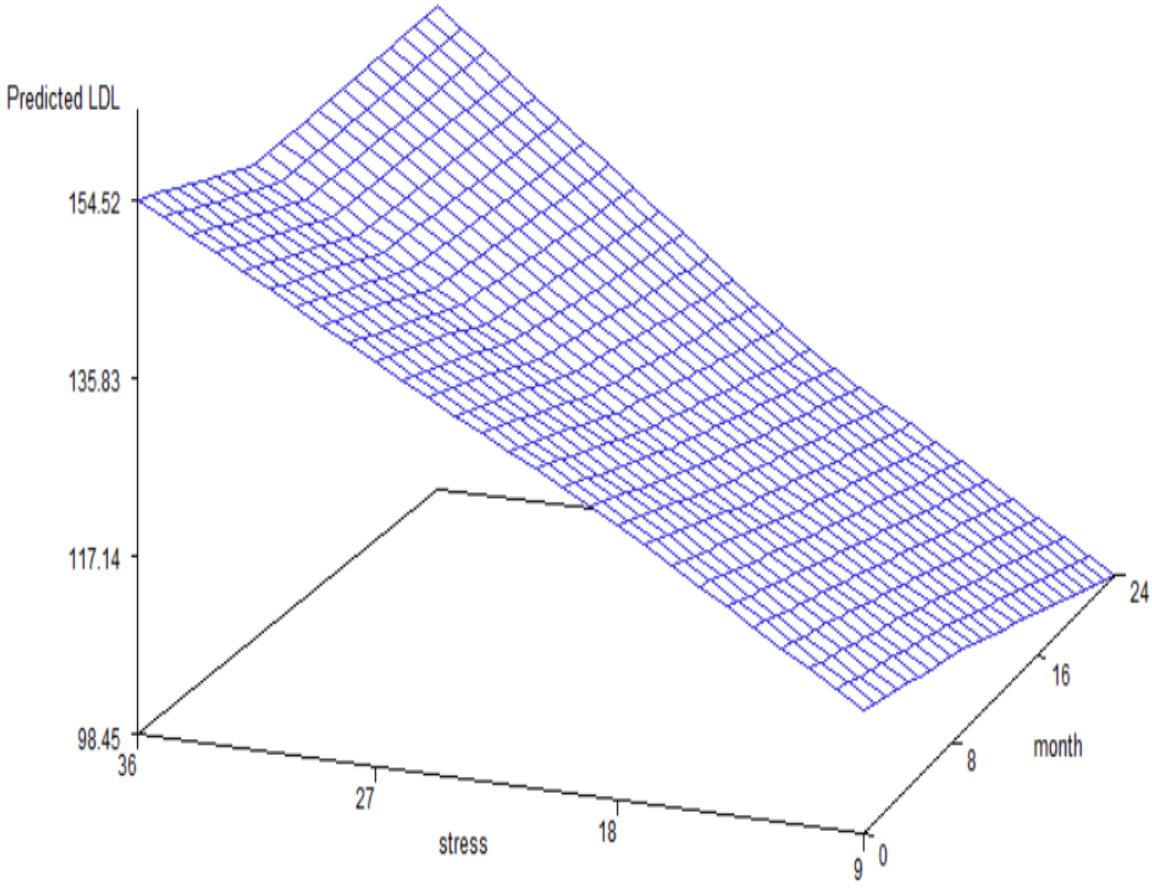
The SAS System

Obs	SmoothingParameter	ScoreData	Obs	stress	month	p_LDL
1	1	GRID_POINTS	1	9	0	110.991
2	1	GRID_POINTS	2	9	1	110.554
3	1	GRID_POINTS	3	9	2	110.118
4	1	GRID_POINTS	4	9	3	109.681
5	1	GRID_POINTS	5	9	4	109.245
6	1	GRID_POINTS	6	9	5	108.808
7	1	GRID_POINTS	7	9	6	108.372
8	1	GRID_POINTS	8	9	7	107.935
9	1	GRID_POINTS	9	9	8	107.499
10	1	GRID_POINTS	10	9	9	107.063

Quite a few rows are omitted.

691	1	GRID_POINTS	691	36	15	151.039
692	1	GRID_POINTS	692	36	16	151.285
693	1	GRID_POINTS	693	36	17	151.531
694	1	GRID_POINTS	694	36	18	151.777
695	1	GRID_POINTS	695	36	19	152.023
696	1	GRID_POINTS	696	36	20	152.269
697	1	GRID_POINTS	697	36	21	152.515
698	1	GRID_POINTS	698	36	22	152.761
699	1	GRID_POINTS	699	36	23	153.007
700	1	GRID_POINTS	700	36	24	153.253

```
proc g3d data=score_results;
plot stress*month=p_LDL;
run;
```




```
/*prediction*/
data point4pred;
   input stress month;
cards;
15 3
;

proc loess data=longform;
 model LDL=stress month;
  ods output ScoreResults=predicted;
     score data=point4pred;
```

```
run;

proc print data=predicted;
run;
```

p_LDL
119.840

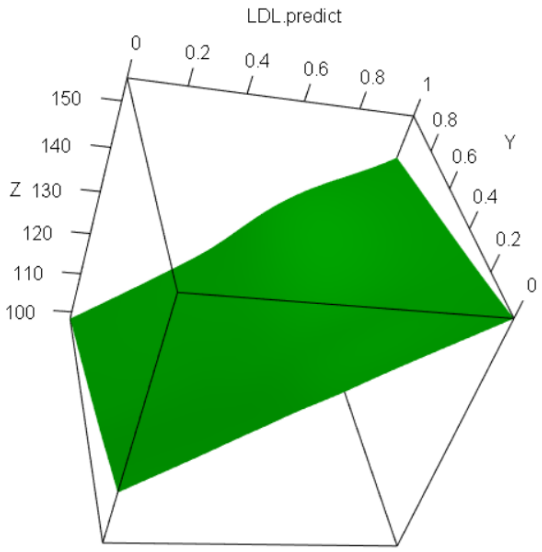
In R:

```
LDL.data<- read.csv(file="./LDLdata.csv", header=TRUE, sep=",")

#creating long-form data set
library(reshape2)
data1<- melt(LDL.data[,c("id","Stress0","Stress6",
"Stress9","Stress24")], id.vars=c("id"),
variable.name = "stressmonth", value.name="stress")

data2<- melt(LDL.data[,c("id","LDL0","LDL6","LDL9","LDL24")],id.vars="id",
variable.name="LDLmonth", value.name="LDL")
data2<- data2[c("LDL")]

longform.data<- cbind(data1,data2)

#creating numeric variable for time
longform.data$month<- ifelse(longform.data$stressmonth=="Stress0", 0,
ifelse(longform.data$stressmonth=="Stress6", 6,
ifelse(longform.data$stressmonth=="Stress9", 9, 24)))

longform.data$stress<- as.numeric(longform.data$stress)
longform.data$LDL<- as.numeric(longform.data$LDL)

longform.data<- na.omit(longform.data)

#fitting the loess model
LDL.loess<- loess(LDL ~ stress + month, data=longform.data,
degree=1, span=1)
```

#plotting fitted surface
grid<- expand.grid(list(stress=seq(min(longform.data$stress),
max(longform.data$stress), 1), month=seq(0, 24, 1)))

LDL.predict<- predict(LDL.loess, newdata=grid)

library(rgl) #R Graphics Library
persp3d(LDL.predict, col="green")



#using the fitted model for prediction
predict(LDL.loess, data.frame(stress=15, month=3))

```
119.1828
```

□