

Nonparametric curves are useful when the type of functional relationship between two variables is not clear (usually not linear). These curves can be the result of a smoothing algorithm or the use of many linear regression functions applied to overlapping intervals on the range of the predictor. To improve the fitting the regression functions may be polynomials and sometimes weighted least squares are also applied to observations with large residual values.

In this example use the `jaws.csv` to fit the following nonparametric curves.

- a) Fit a lowess smoothing curve to the data
- b) Fit a loess regression to the data and plot the smoothing curve
- c) Use library `mgcv` to fit a `gam` regression model to the data and plot the smoothing curve
- d) Fit a polynomial regression model to the data and plot the smoothing curve

```
# smooth.r

rm(list=ls())
setwd("C:/Users/USC Guest/Downloads2")
d0=read.table("jaws.txt",header=T)
head(d0)
#      age      bone
#1  0.000000  0.00000
#2  5.112000 20.22000
#3  1.320000 11.11130
#4 35.240000 140.65000
#5  1.632931 26.15218
#6  2.297635 10.00100

# lowess
par(mfrow=c(2,2))
plot(bone~age,d0,pch=16)
text(45,20,"lowess",pos=2)

d1=lowess(age,bone)      # d1, a list of (x,y) of smoothed lowess curve
d1=data.frame(d1)
lines(y~x,d1,col="red")

# loess
m2=loess(bone~age)        # m2 is a loess reg model
xv = 0:50
newval = data.frame(age=xv)
yv=predict(m2,newval)

plot(bone~age,d0,pch=16)
text(45,20,"loess",pos=2)
lines(xv,yv,col="red")

# gam
library(mgcv)
m3=gam(bone~s(age))       # m3 is a gam model
yv=predict(m3,newval)

plot(bone~age,d0,pch=16)
text(45,20,"gam",pos=2)
lines(xv,yv,col="red")

# polynomial
m4=lm(bone~age+I(age^2)+I(age^3))
yv=predict(m4,newval)

plot(bone~age,d0,pch=16)
text(45,20,"polynomial",pos=2)
lines(xv,yv,col="red")
```

