Advanced Statistical Modeling

Non-parametric models - Estimating conditional variance

Haoran Mo, Alexandra Yamaui

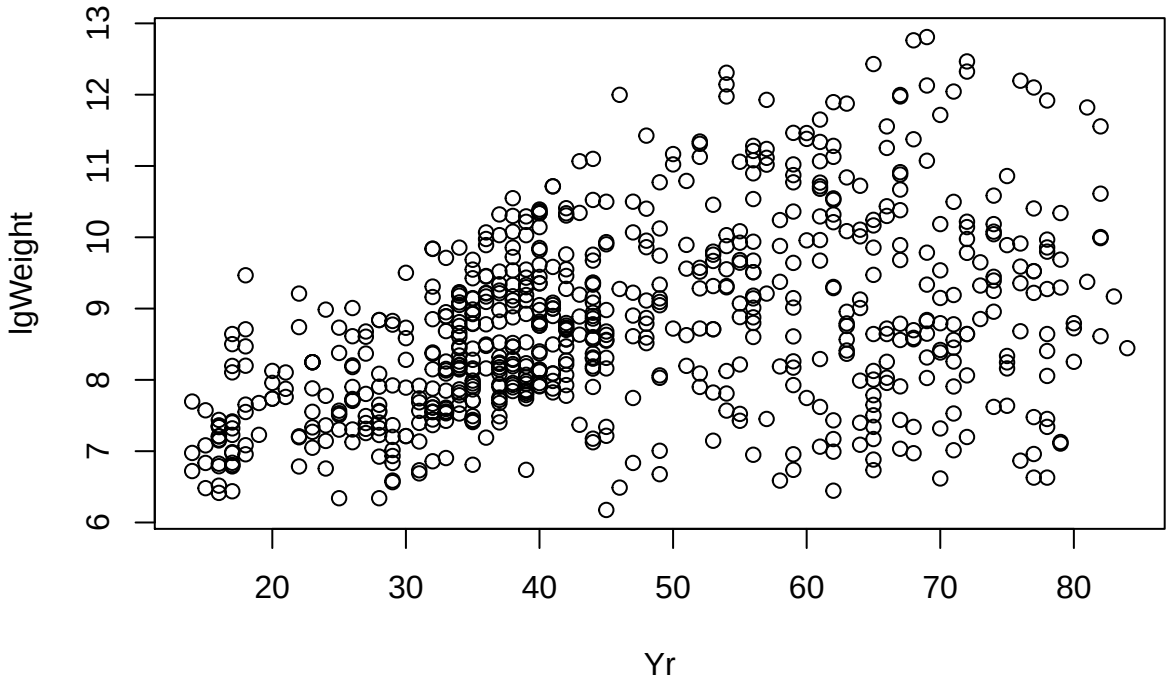
Having the heteroscedastic regression model

$$Y = m(x) + \sigma(x)\varepsilon = m(x) + \epsilon,$$

, where $E(\varepsilon) = 0, V(\varepsilon) = 1$

we want to estimate the function that represents the conditional variance σ^2 of the variable lgWeigth (log(Weight)) from the aircraft dataset, given that the explanatory variable (Yr) is equal to a value x. We will use nonparametric methods to do that. Below is the plot of the response variable vs the explanatory variable.

```
data(aircraft)
attach(aircraft)
lgWeight <-log(Weight)
plot(Yr,lgWeight)
```



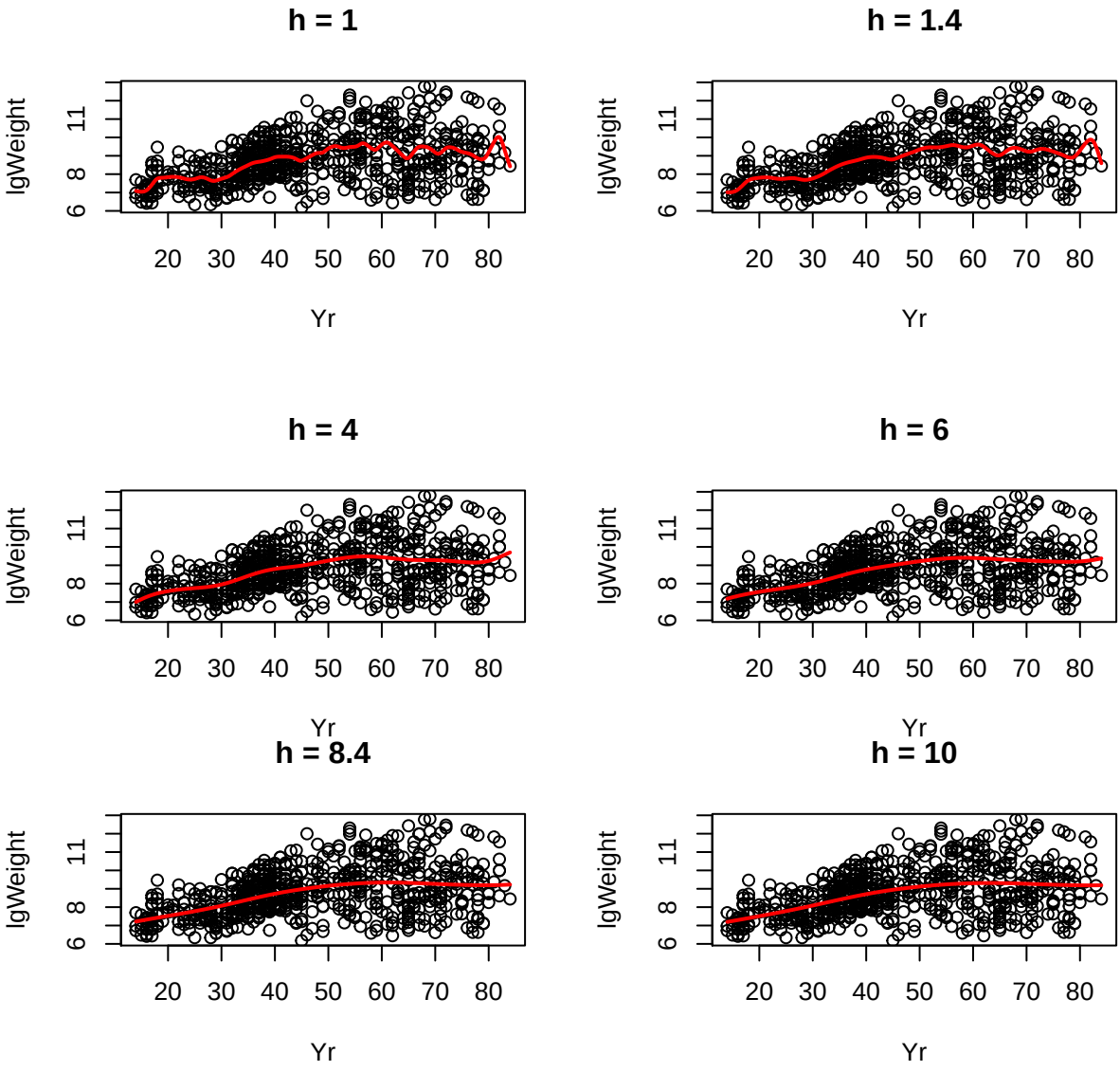
Initially, we are going to fit a local linear regression model to obtain an estimation $\hat{m}(x)$ of every point x. The general idea is to build a grid of intervals (t_i) centered around each point x and estimate a local linear regression in each interval. Doing this, we are going to try multiple values for the smoothing parameter h, which controls weight concentration around each point x.

To make the regression function smooth weights are assigned to each pair (t_i, y_i) using a kernel function, which, in this case, is the Normal density function centered at 0, with h as standard deviation.

The result is shown below, where we can see an smooth function (in red) of the estimates.

```
# step 1 Fit a nonparametric regression to data (xi,yi) and save
#   the estimated values m^ (xi).
par(mfrow=c(2,2))
```

```
hs <- c(1,1.4,4,6,8.4,10)
tg = seq(min(Yr), max(Yr), length=709)
for (h in hs) {
  llr <- loc.lin.reg(x=Yr, y=lgWeight, h=h, tg=tg)
  plot(Yr,lgWeight, main = paste("h = ", h, sep = ""))
  lines(tg, llr$mt, col=2, lwd=2)
}
```



Changing the smoothing parameter we can see that as h increases the function curve becomes smoother. We will choose the combination $h = 4$

```
h <- 4
llr <- loc.lin.reg(x=Yr, y=lgWeight, h=h, tg=tg)
```

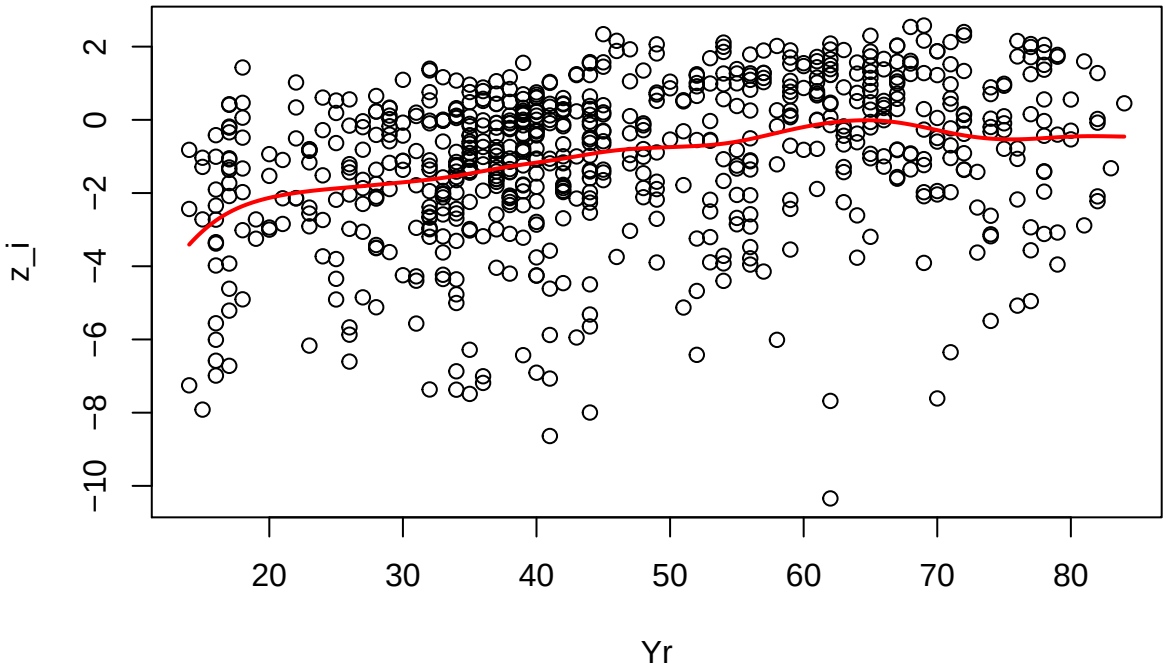
Now we are going to transform the estimated residuals applying logarithm to the square of it $\hat{\epsilon}_i = (y_i - \hat{m}(x_i))^2$, which represents the variance (the square deviation) of the model.

$$z_i = log\epsilon_i^2 = log((y_i - \hat{m}(x_i))^2)$$

```r
# step 2 Transform the estimated residuals z_i = log((y_i - llr$mt)^2)
z_i = log((lgWeight - llr$mt)^2)
```

Then we perform a nonparametric regression over (x_i, z_i) to obtain the estimation of the (logarithm of the) variance $log\sigma^2(x)$.

```r
# step 3 Fit a nonparametric regression to data (Yr,z_i) and
# call the estimated function q^(x).
llr2 <-loc.lin.reg(x=Yr, y=z_i, h=h, tg=tg)
plot(Yr,z_i)
lines(tg, llr2$mt, col=2, lwd=2)
```

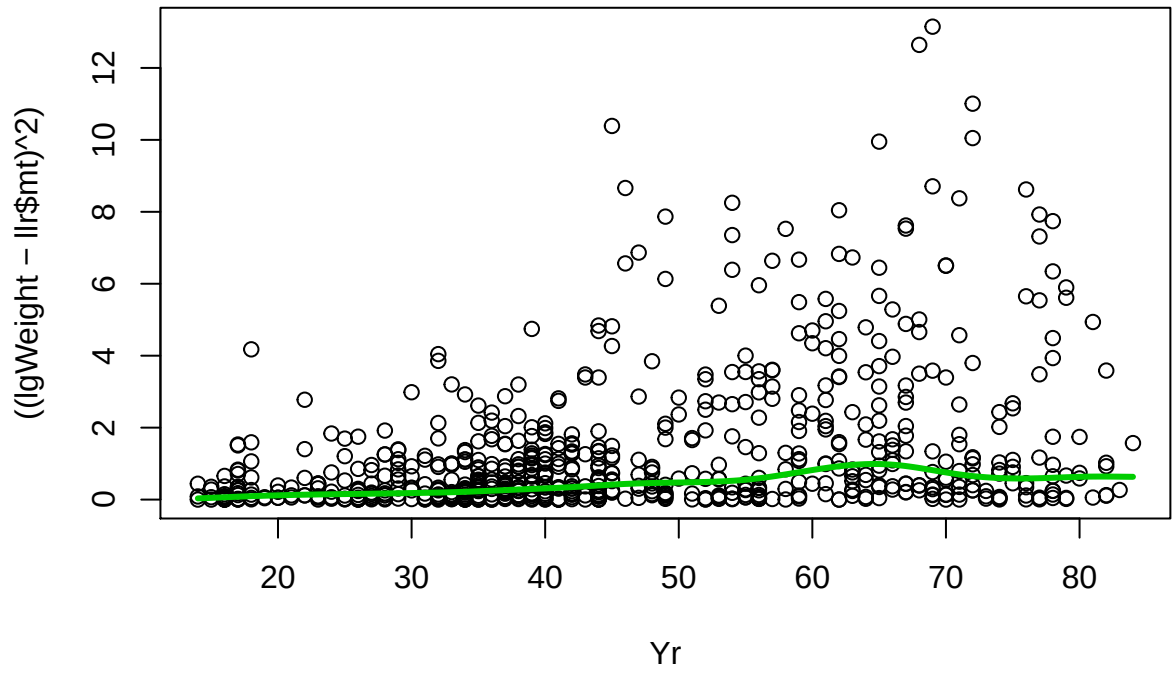


Finally, we can obtain the conditional variance applying exponential to the estimation obtained before

```r
# step 4 Estimate sigma_2(x)
sigma_2 =  exp(llr2$mt)
```

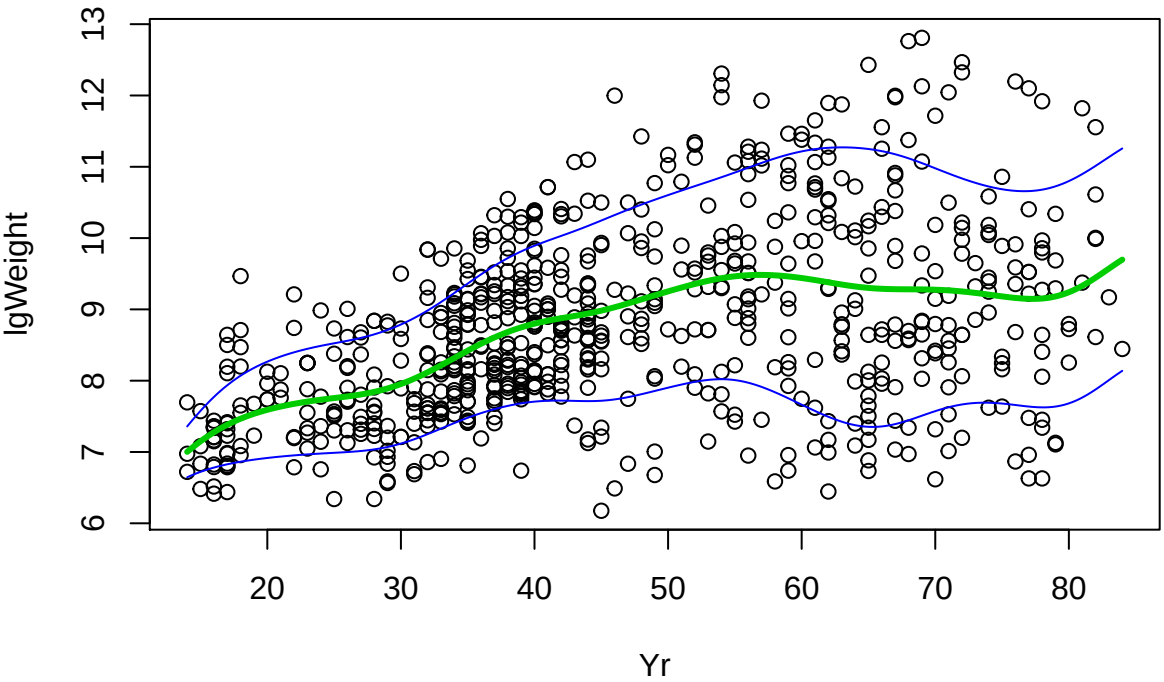
Plotting the residuals against x_i we superimpose the estimated function $\hat{\sigma}^2(x)$

```r
plot(Yr,((lgWeight - llr$mt)^2)) # residual square against x_i
lines(tg, sigma_2, col=3, lwd=3)
```



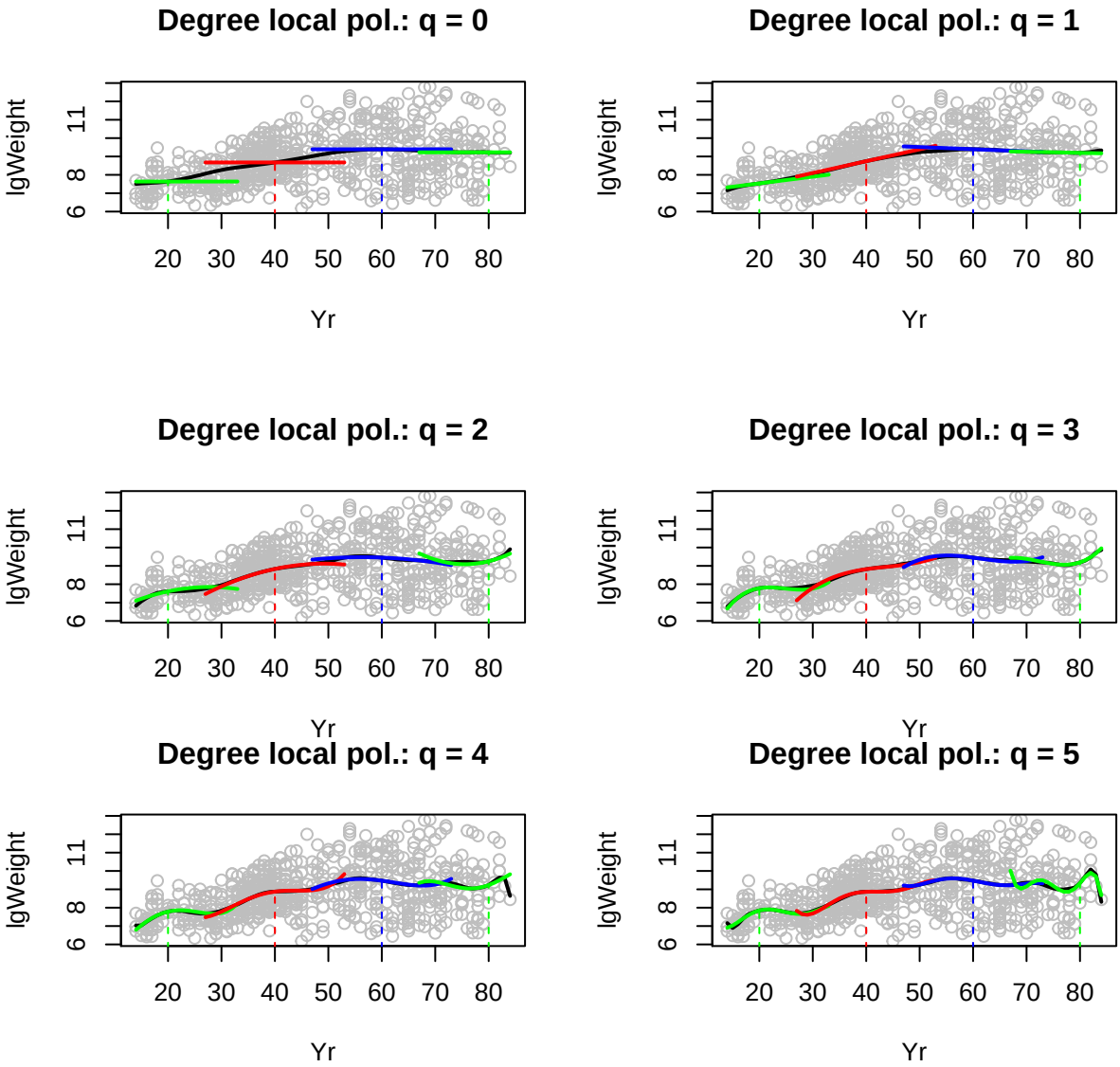
Here we present the draw of the function $\hat{m}(x)$ and the superimposed bands $\hat{m}(x) \pm 1.96\hat{\sigma}(x)$

```
sigma = sqrt(sigma_2)
plot(Yr,lgWeight)
lines(tg,llr$mt,col=3,lwd=3) # mt from step 1
lines(tg,llr$mt+1.96*sigma,col=4,lwd=1)
lines(tg,llr$mt-1.96*sigma,col=4,lwd=1)
```



After fitting a linear regression we are going to try with local polynomial regression of different degress, using Epanechnikov as the kernel function.

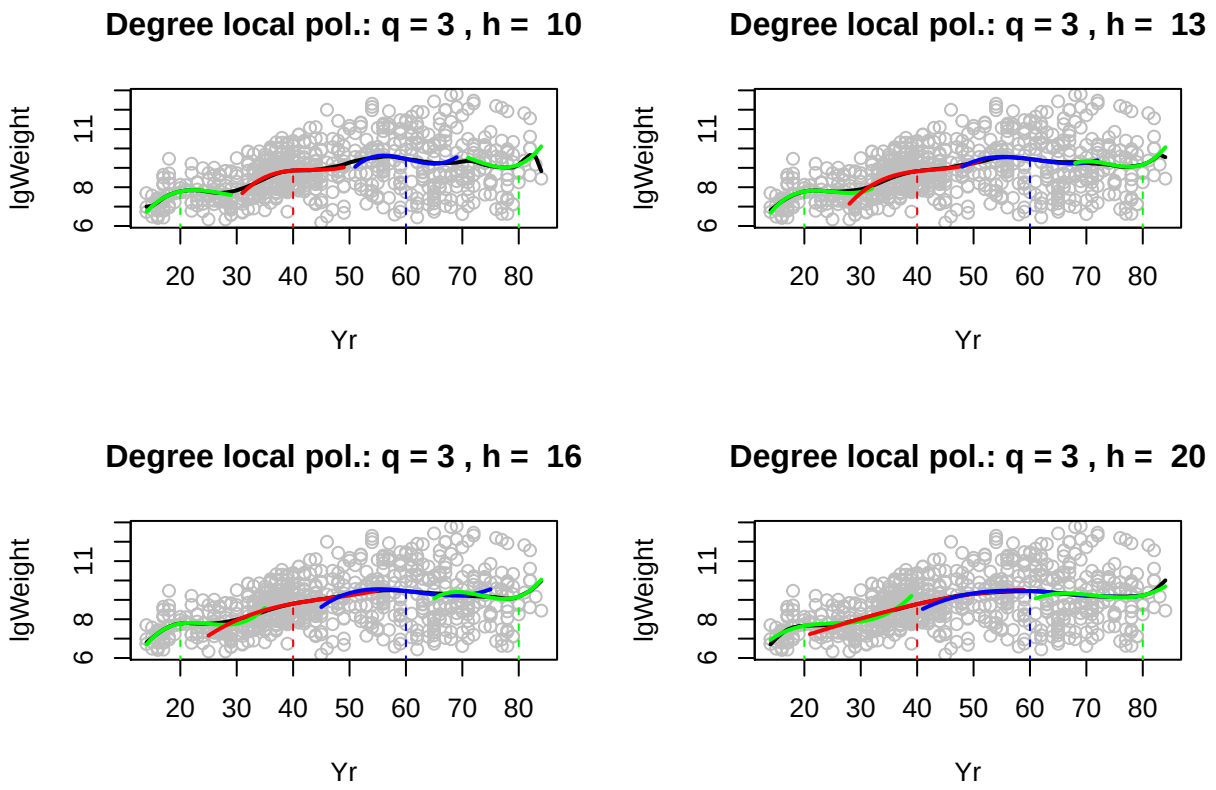
```
par(mfrow=c(2,2))

degrees <- c(0,1,2,3,4,5)

for (d in degrees) {
  lpr_visual(x=Yr, y=lgWeight, q=d, tg=c(20,40,60,80), # h=10
             xlab="Yr", ylab="lgWeight",
             main=paste('Degree local pol.: q = ', d, sep = ""), type.kernel="epan")
}
```



We can see that the degree that allows to approximate better the curve is 3, therefore, we are going to use this degree and variate the smoothing parameter to obtain the best fit.

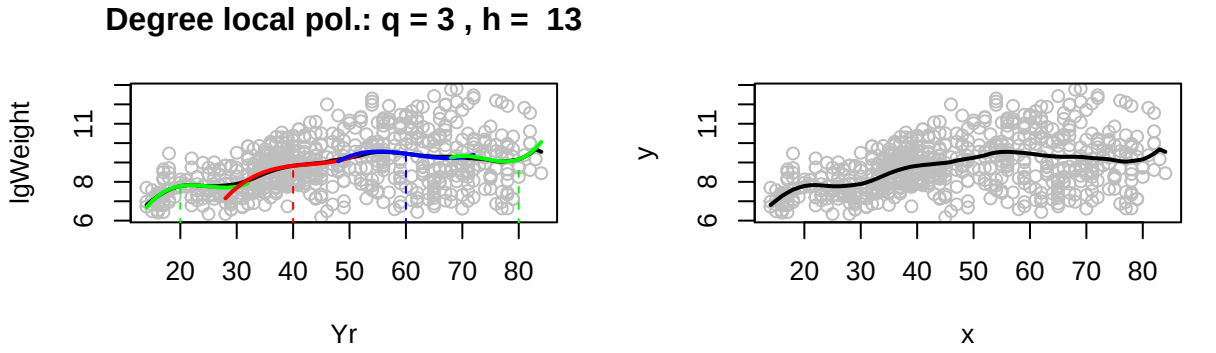
```
par(mfrow=c(2,2))
degree <- 3
hs <- c(10,13,16,20)
for (h in hs) {
  lpr_visual(x=Yr, y=lgWeight, h=h, q=degree, tg=c(20,40,60,80),
             xlab="Yr", ylab="lgWeight",
```

```
            main=paste('Degree local pol.: q =', paste(degree, paste(", h = ", h))),
            type.kernel="epan")
}
```



After tunning the smoothing parameter h we see that the best fit was obtainend using $h = 13$, and whith this we will build the polynomial regression

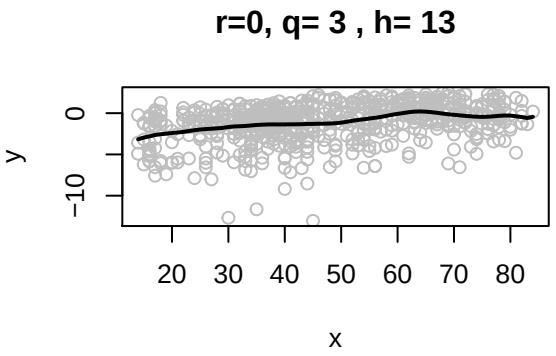
```
par(mfrow=c(2,2))
h <- 13
lpr_visual(x=Yr, y=lgWeight, h=h, q=degree, tg=c(20,40,60,80),
           xlab="Yr", ylab="lgWeight",
           main=paste('Degree local pol.: q =', paste(degree, paste(", h = ", h))), type.kernel="epan")
lpr <- locpolreg(x=Yr, y=lgWeight, h=h, q=degree, r=0, type.kernel="epan")
```



Repeating the same proceure we used before for the local linear regression, we will transform the estimated residuals and perform the polynomial regression over z_i

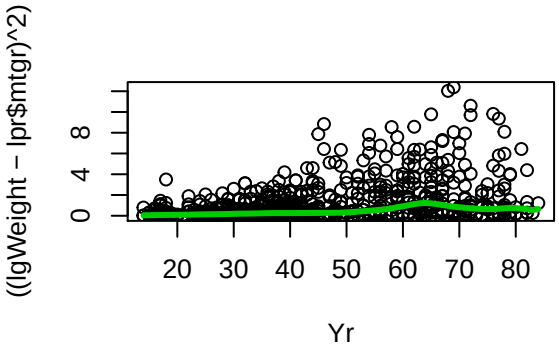
```
par(mfrow=c(2,2))
# step 2 Transform the estimated residuals z_i = log((y_i - lpr$mtgr)^2)
```

```
z_i = log((lgWeight - lpr$mtgr)^2)

# step 3 Fit a nonparametric regression to data (Yr,z_i) and
# call the estimated function q^(x).
lpr2 <-locpolreg(x=Yr, y=z_i, h=h, q=degree, r=0, type.kernel="epan",
                 main=paste("r=0, q=",paste(degree, paste(", h=",h))))
```



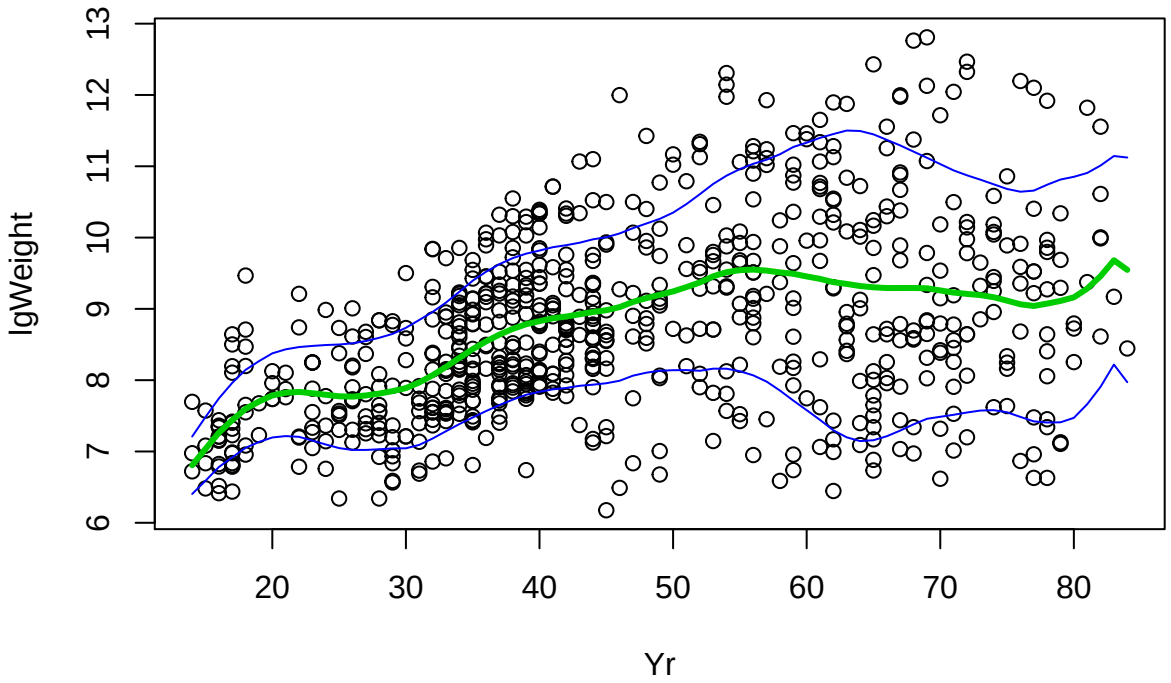
Finally, we estimate the variance σ^2

```
par(mfrow=c(2,2))
# step 4 Estimate sigma_2(x)
sigma_2 =  exp(lpr2$mtgr)
plot(Yr,((lgWeight - lpr$mtgr)^2)) # residual square against x_i
lines(Yr, sigma_2, col=3, lwd=3)
```



Function $\hat{m}(x)$ and the superimposed bands $\hat{m}(x) \pm 1.96\hat{\sigma}(x)$

```
sigma = sqrt(sigma_2)
plot(Yr,lgWeight)
lines(Yr,lpr$mtgr,col=3,lwd=3)
lines(Yr,lpr$mtgr+1.96*sigma,col=4,lwd=1)
lines(Yr,lpr$mtgr-1.96*sigma,col=4,lwd=1)
```



After performing performing a local linear and polynomial regressions we can see that the results are comparable, the polynomial showing a smoother curve. Hence, we can say that for this escenario of the aircraft data both approaches are satifactory.