ASM Practice

Local Linear Regression

Maria Gkotsopoulou & Ricard Monge Calvo & Amalia Vradi

20/11/2019

The aim of this project is to compute the conditional variance $\sigma^2(x)$ of the variable $lgWeight = log(Weight)$ of the *aircraft* dataset (in *sm* package) given the year, *Yr*, variable.

```r
# Load data and pre-process
data("aircraft")
attach(aircraft)
lgPower <- log(Power)
lgSpan <- log(Span)
lgLength <- log(Length)
lgWeight <- log(Weight)
lgSpeed <- log(Speed)
lgRange <- log(Range)
```

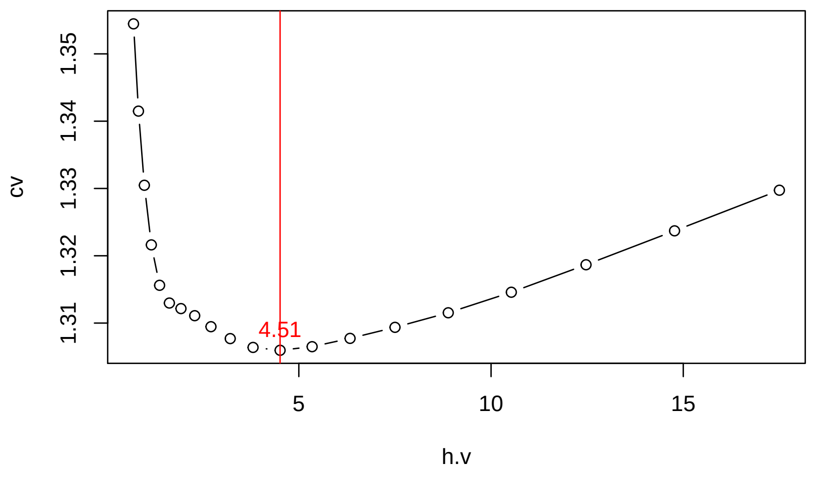
Estimation using *locpolreg* function

We load our local function *locpolreg* along with the *bandwith_selection* script which contains different functions for bandwidth selection. We will choose the bandwidth hyper-parameter by LOOCV. We use the appropriate function in the *bandiwth_selection* script to get the LOOCV and GCV estimates (which uses the *locpolreg* function). Regarding the Kernel choice, we decide to use the *normal* kernel.

```r
# Load local functions
source("locpolreg.R")
source("bandwith_selection.R")
```

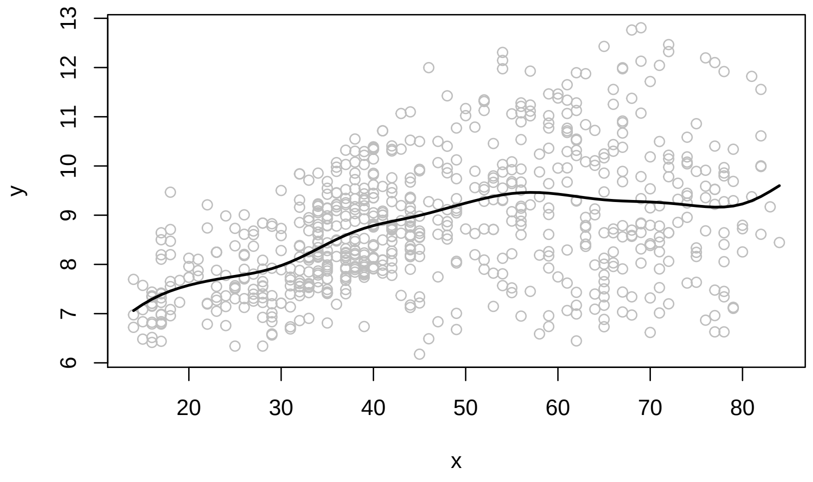
The first thing we need to do is to compute the local linear regression for the predicted variable *lgWeight* depending on *Yr*. We choose the bandwidth as the one that minimizes the LOOCV estimate:

```r
# Choice between "normal"  (Gaussian, default),
# "epan"    (Epanechnikov) or
# "rs.epan" (re-scaled Epanechnikov)
# "unif"    (Uniform Kernel in [-1,1])
kernel.type <- "normal"
# Define Bandwith candidates
h.v <- exp(seq(log(diff(range(Yr))/100), log(diff(range(Yr))/4),l=20))
# Get LOOCV and GCV estimates
h.result <- h.cv.gcv(x=Yr, y=lgWeight, h.v=h.v, q=1, type.kernel=kernel.type) %>%
  as.data.frame() %>% arrange(h.v)
plot(cv~h.v, h.result, type="b")
h.min <- h.result[which.min(h.result$cv),"h.v"]
abline(v=h.min, col=2)
text(x=h.min,y=h.result[which.min(h.result$cv),"cv"], labels
     =round(h.min,digits=2), pos = 3, col = 2)
```



After choosing the bandwidth, we build the local linear regression model using the *locpolreg* function and compute the residual values.

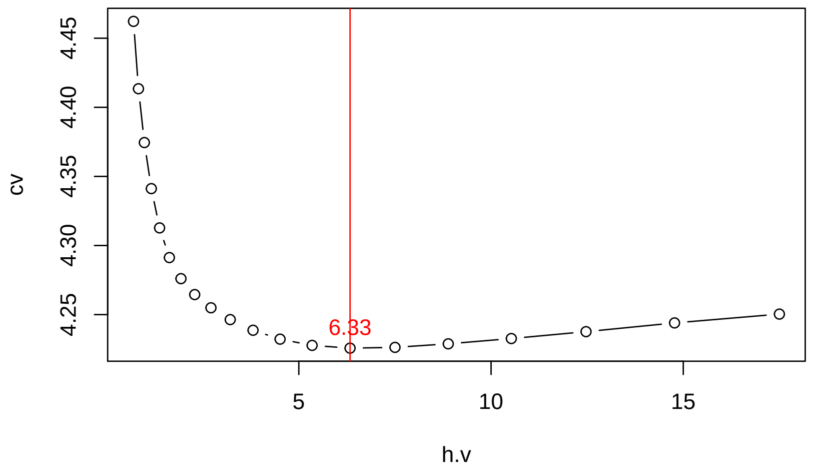
```
m.result <- locpolreg(x=Yr,y=lgWeight,h=h.min,q=1,type.kernel=kernel.type)
```



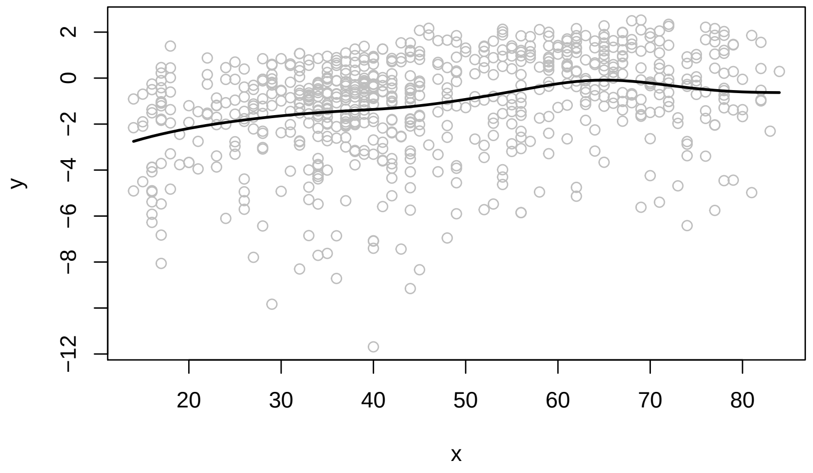
```
eps <- lgWeight - m.result$mtgr
eps2 <- eps*eps
Z <- log(eps2)
```

Having obtained the residual values $\hat{\epsilon}_i = y_i - \hat{m}(x_i)$ we compute their logarithm $z_i = \log \hat{\epsilon}_i^2$. We need to build a new model for z_i against x_i. We choose the new bandwidth and build the model:

```
h.v <- exp(seq(log(diff(range(Yr))/100), log(diff(range(Yr))/4),l=20))
# Get LOOCV and GCV estimates
h.result <- h.cv.gcv(x=Yr, y=Z, h.v=h.v, q=1, type.kernel=kernel.type) %>%
  as.data.frame() %>% arrange(h.v)
plot(cv~h.v, h.result, type="b")
h.min <- h.result[which.min(h.result$cv),"h.v"]
abline(v=h.min, col=2)
text(x=h.min,y=h.result[which.min(h.result$cv),"cv"], labels
     =round(h.min,digits=2), pos = 3, col = 2)
```



```
q.result <- locpolreg(x=Yr,y=Z,h=h.min,q=1,type.kernel=kernel.type)
```

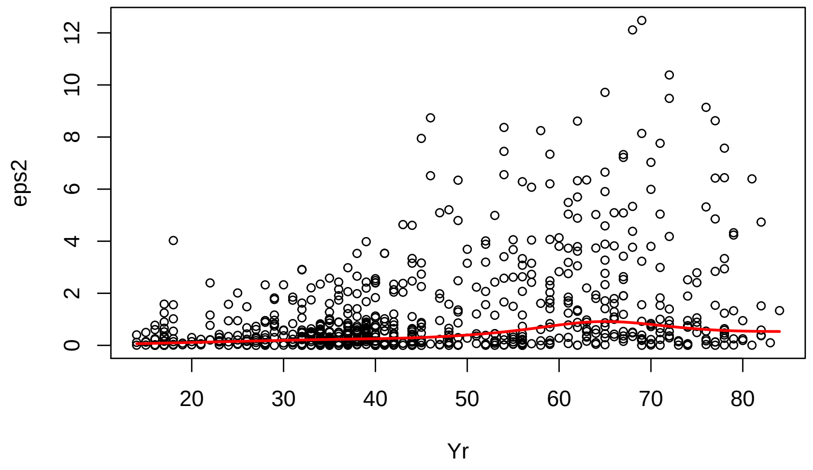


Finally, the conditional variance $\hat{\sigma}^2(x) = \exp \hat{q}(x)$ where $\hat{q}(x)$ is the estimate of the previous model.

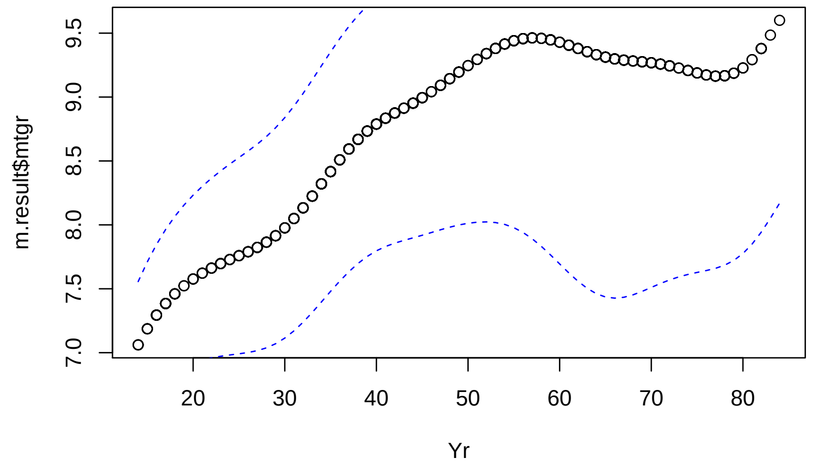
```
sigma2 <- exp(q.result$mtgr)
sigma <- sqrt(sigma2)
```

To sum up the results, we plot the value of ϵ_i^2 against x_i superimposing the values of $\hat{\sigma}^2(x)$, and also the values of $\hat{m}(x)$ with the bands $\hat{m}(x) \pm 1.96\hat{\sigma}(x)$.

```
plot(Yr, eps2, cex=0.8)
points(Yr, sigma2, type="l", col="red", lwd=2)
```



```r
plot(Yr,m.result$mtgr)
points(Yr,m.result$mtgr+1.96*sigma, type="l", col="blue", lty=2)
points(Yr,m.result$mtgr-1.96*sigma, type="l", col="blue", lty=2)
```



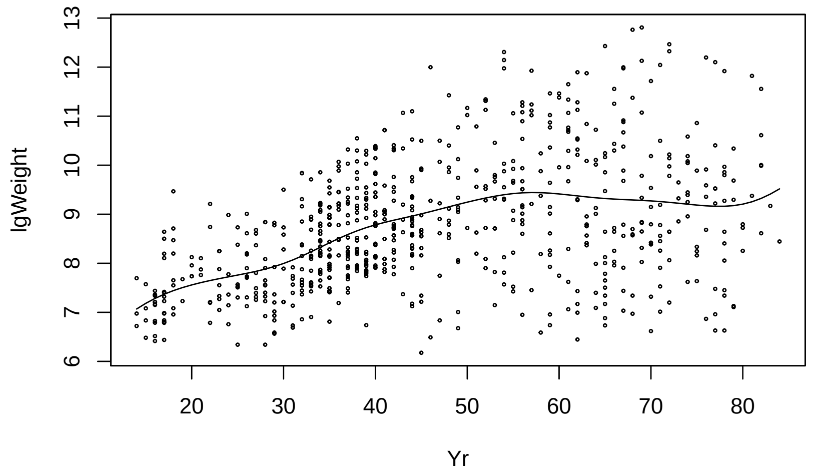
Estimation using *sm.regression* function

We use the function *dpill* in the *KernSmooth* package in order to compute the *Plug-in* bandwidth parameter. Afterwards, we use this bandwidth with the *sm.regression* function in the *sm* package to compute the local linear regression models, both $\hat{m}(x)$ and $\hat{q}(x)$. Regarding the Kernel choice, we decide to use the *normal* kernel.

```r
h.min <- dpill(x=Yr, y=lgWeight)
```

After choosing the bandwidth, we build the local linear regression model using the *sm.regression* function and compute the residual values.

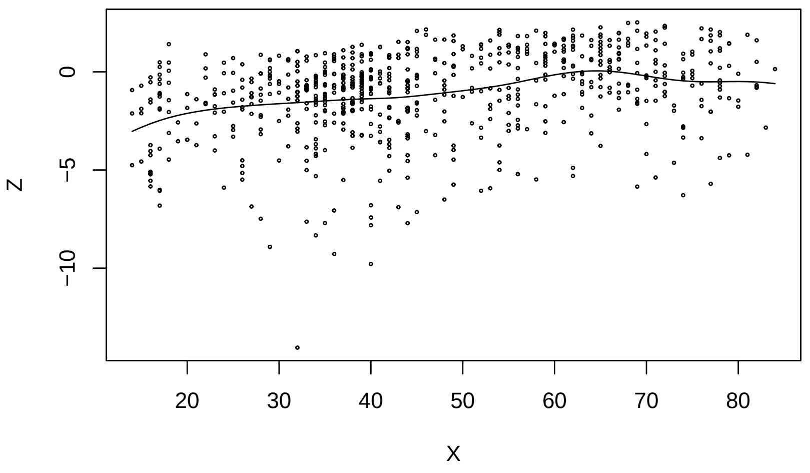
```r
m.result <- sm.regression(x=Yr,y=lgWeight,h=h.min,
                          eval.grid=FALSE, eval.points=Yr)
```



```r
X <- m.result$eval.points # == Yr
eps <- lgWeight - m.result$estimate
eps2 <- eps*eps
Z <- log(eps2)
```

Now we have the residual values $\hat{\epsilon}_i = y_i - \hat{m}(x_i)$ and their logarithm $z_i = \log \hat{\epsilon}_i^2$. We need to build a new model for z_i against x_i. We choose the new bandwidth and build the model:

```r
h.min <- dpill(x=X, y=Z)
q.result <- sm.regression(x=X,y=Z,h=h.min,
                          eval.grid=FALSE, eval.points=X)
```

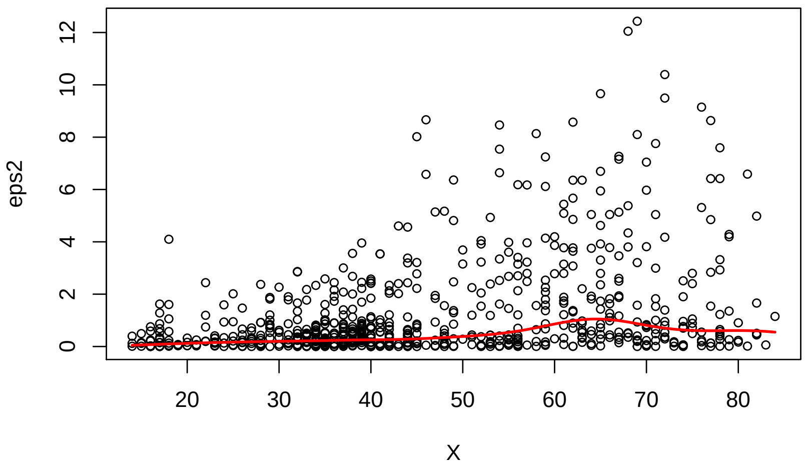


Finally, the conditional variance $\hat{\sigma}^2(x) = \exp \hat{q}(x)$ where $\hat{q}(x)$ is the estimate of the previous model.

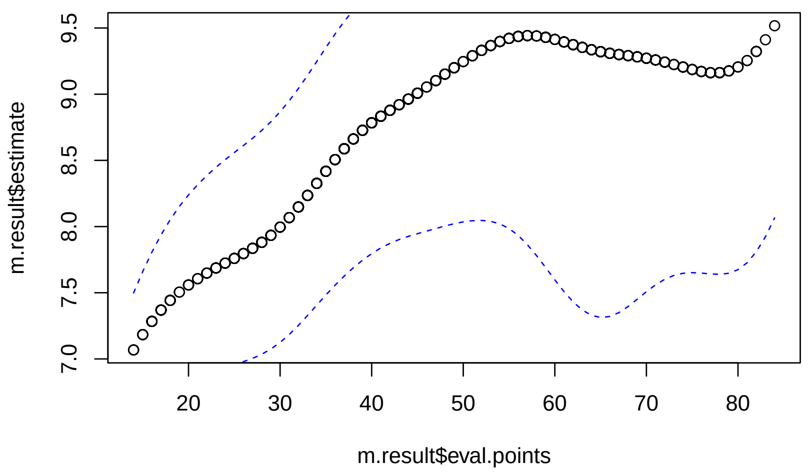
```r
sigma2 <- exp(q.result$estimate)
sigma <- sqrt(sigma2)
```

To sum up the results, we plot the value of ϵ_i^2 against x_i superimposing the values of $\hat{\sigma}^2(x)$, and also the values of $\hat{m}(x)$ with the bands $\hat{m}(x) \pm 1.96\hat{\sigma}(x)$.

```r
plot(X, eps2, cex=0.8)
points(X, sigma2, type="l", col="red", lwd=2)
```



```r
plot(m.result$eval.points, m.result$estimate)
points(m.result$eval.points, m.result$estimate+1.96*sigma,
       type="l", col="blue", lty=2)
points(m.result$eval.points, m.result$estimate-1.96*sigma,
       type="l", col="blue", lty=2)
```


m.result$estimate
7.0
7.5
8.0
8.5
9.0
9.5
20
30
40
50
60
70
80
m.result$eval.points