Local Linear Regression

Azzarito Domenico, Daniel Reverter, Alexis Vendrix

11 November, 2025

Estimating the conditional variance by local linear regression

```r
# Libraries
library(sm)
library(KernSmooth)
source("locpolreg.R")
```

Aircraft Data and log transformation

```r
# Load data
data(aircraft)
attach(aircraft)

lgPower <- log(Power)
lgSpan <- log(Span)
lgLength <- log(Length)
lgWeight <- log(Weight)
lgSpeed <- log(Speed)
lgRange <- log(Range)
```

Estimating the conditional variance

Consider the heteroscedastic regression model

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

where $\mathrm{E}(\varepsilon) = 0$, $\mathrm{Var}(\varepsilon) = 1$ and $\sigma^2(x)$ is an unknown function that gives the conditional variance of Y given that the explanatory variable is equal to x. Let us define $Z = \log((Y - m(x))^2) = \log \epsilon^2$ and $\delta = \log \varepsilon^2$. Then

$$Z = \log \sigma^2(x) + \delta,$$

and $\delta = \log \varepsilon^2$ is a random variable with expected value close to 0 (observe that $\mathrm{E}(\log \varepsilon^2)$ is close to $\log \mathrm{E}(\varepsilon^2) = \log \mathrm{Var}(\varepsilon) = \log 1 = 0$, at least when $\mathrm{Var}(\varepsilon^2)$ is small) taking the role of *noise* in the regression of Z against x (that is, Z is the response variable and x is the predicting variable).

Given that the values of ε_i^2 are not observable, a way to estimate the function $\sigma^2(x)$ is as follows:

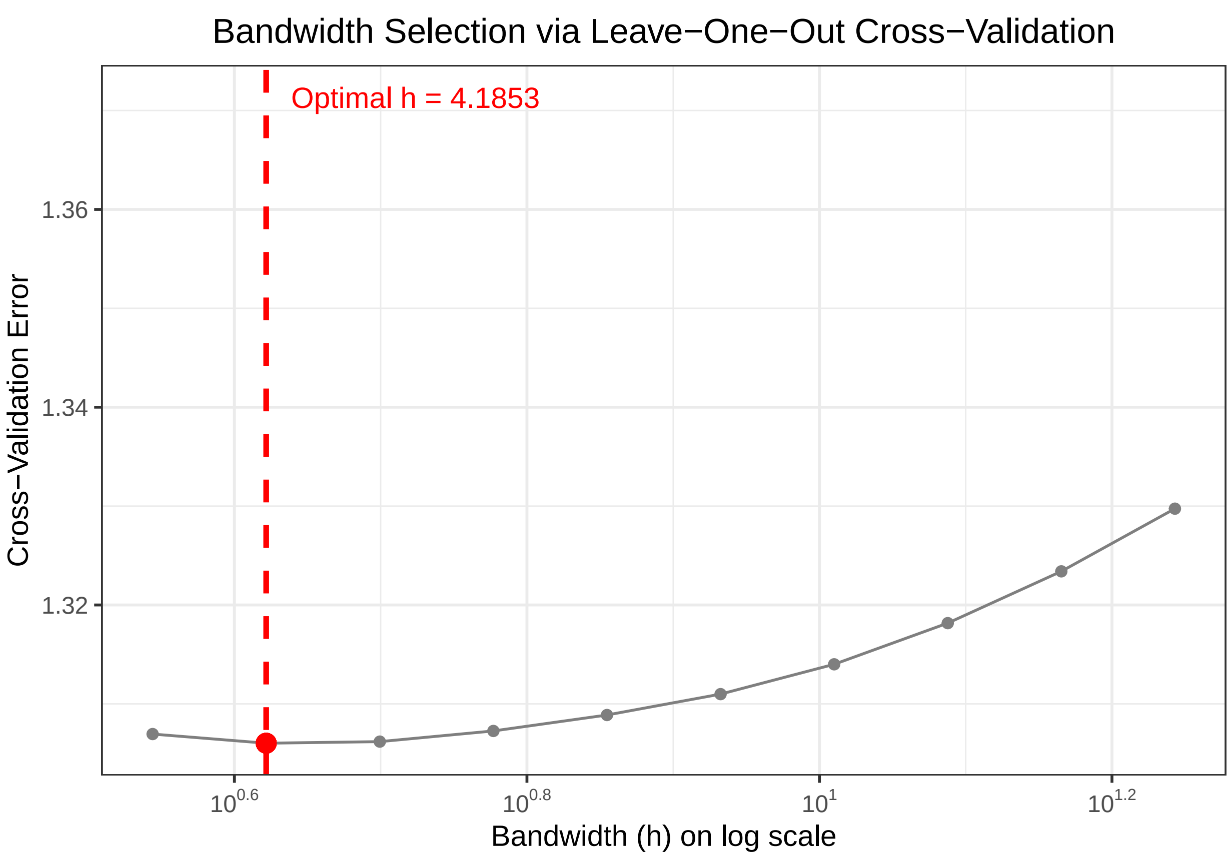
Part 1 - With function loc.pol.reg

1. Fit a nonparametric regression to data (x_i, y_i) and save the estimated values $\hat{m}(x_i)$.

```r
## Function in ATENEA to get loo-cv
h.cv.gcv <- function(x,y,h.v = exp(seq(log(diff(range(x))/20),
```

```
                                          log(diff(range(x))/4),l=10)),
                           p=1,type.kernel="normal"){
  n <- length(x)
  cv <- h.v*0
  gcv <- h.v*0
  for (i in (1:length(h.v))){
    h <- h.v[i]
    aux <- locpolreg(x=x,y=y,h=h,p=p,tg=x,
                     type.kernel=type.kernel, doing.plot=FALSE)
    S <- aux$S
    h.y <- aux$mtgr
    hii <- diag(S)
    av.hii <- mean(hii)
    cv[i] <- sum(((y-h.y)/(1-hii))^2)/n
    gcv[i] <- sum(((y-h.y)/(1-av.hii))^2)/n
  }
  return(list(h.v=h.v,cv=cv,gcv=gcv))
}

cv.res <- h.cv.gcv(Yr, lgWeight, type.kernel="normal")
h_cv <- cv.res$h.v[which.min(cv.res$cv)]
min_cv_error <- min(cv.res$cv)
```

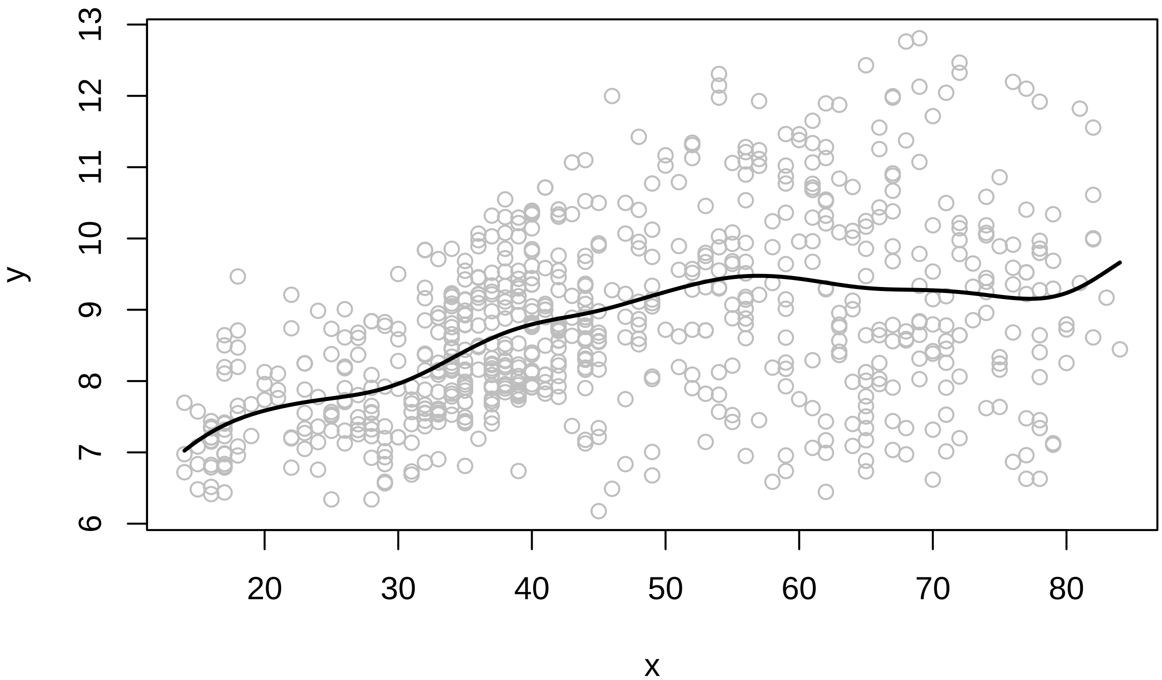
Bandwidth Selection via Leave-One-Out Cross-Validation



The optimal bandwidth selected by LOOCV is h = 4.1853.

$\hat{m}(x_i)$

```r
m_hat <- locpolreg(Yr, lgWeight, h=h_cv, q=1, tg=seq(min(Yr),max(Yr),length=200))
```

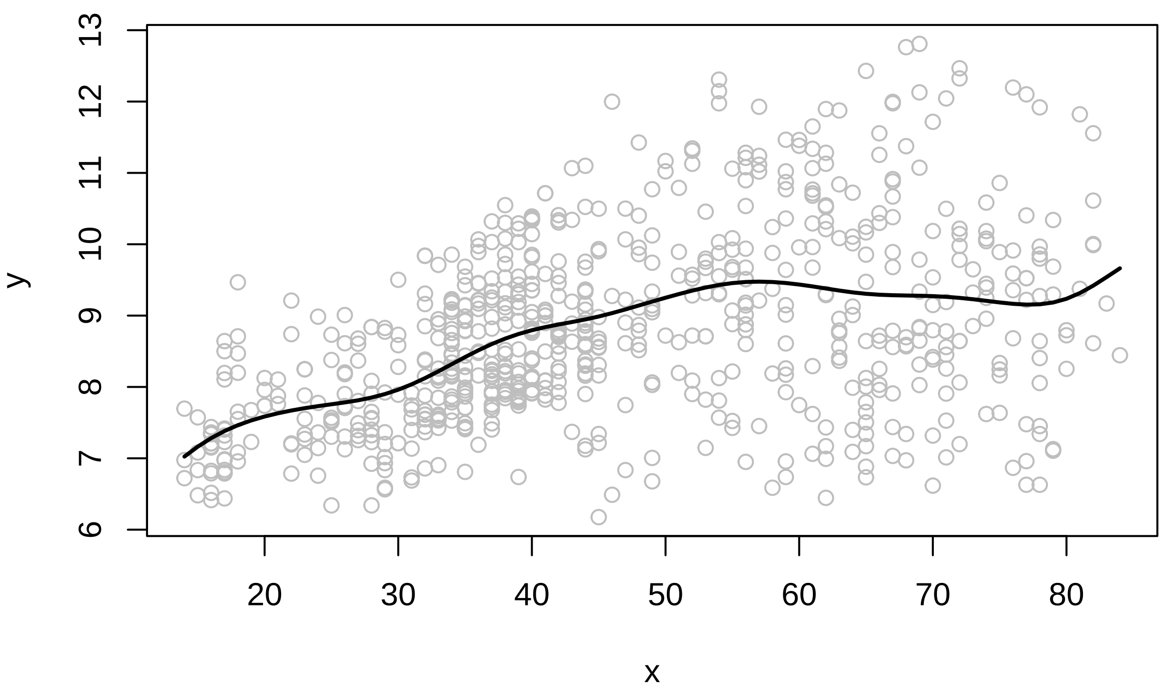


This plot displays the local linear regression fit (solid black line) over the raw data (grey circles). The smooth curve effectively captures the non-linear trend in the data.

2. Transform the estimated residuals $\hat{\epsilon} = y_i - \hat{m}(x_i)$:

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

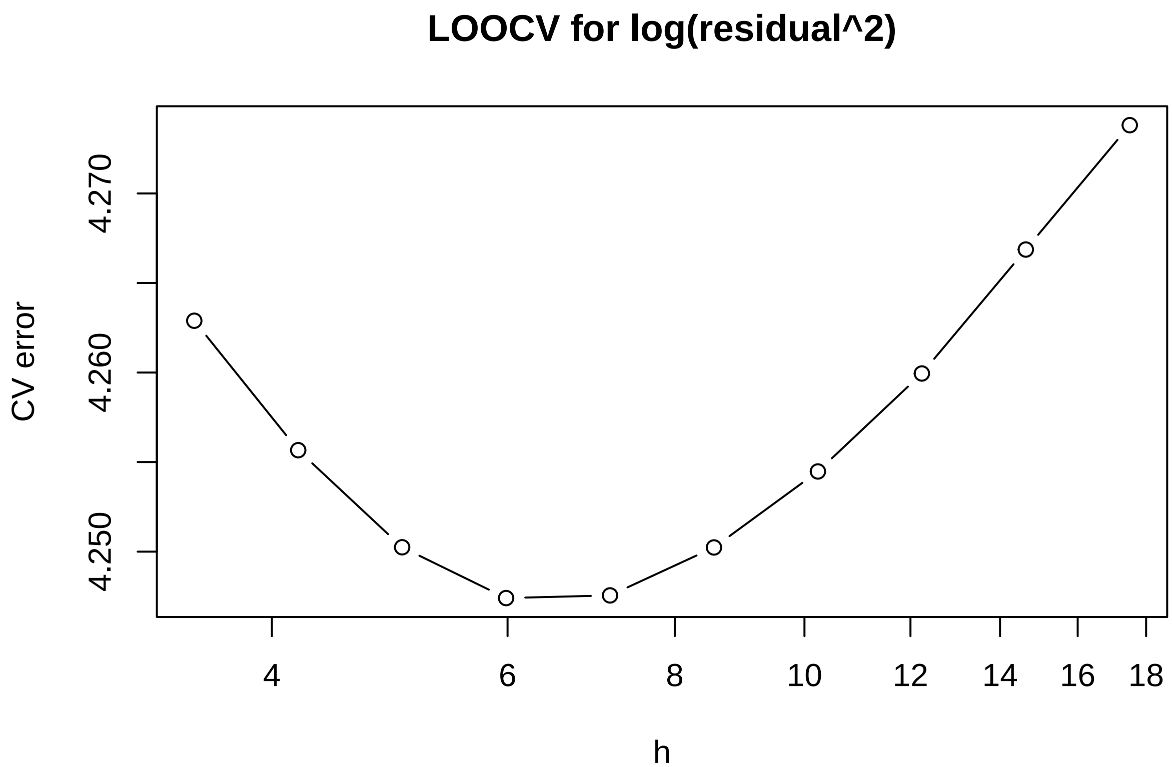
```r
eps_hat <- lgWeight - locpolreg(Yr, lgWeight, h=h_cv, q=1, r=0, tg=Yr)$mtgr
```



```r
z <- log(eps_hat^2)
```

3. Fit a nonparametric regression to data (x_i, z_i) and call the estimated function $\hat{q}(x)$. Observe that $\hat{q}(x)$ is an estimate of $log(\sigma^2(x))$.

```r
# Select bandwidth for z_i regression by LOOCV again
cv.res.var <- h.cv.gcv(Yr, z, type.kernel="normal")

# Plot CV curve for variance regression
plot(cv.res.var$h.v, cv.res.var$cv, type="b", log="x",
     main="LOOCV for log(residual^2)", xlab="h", ylab="CV error")
```

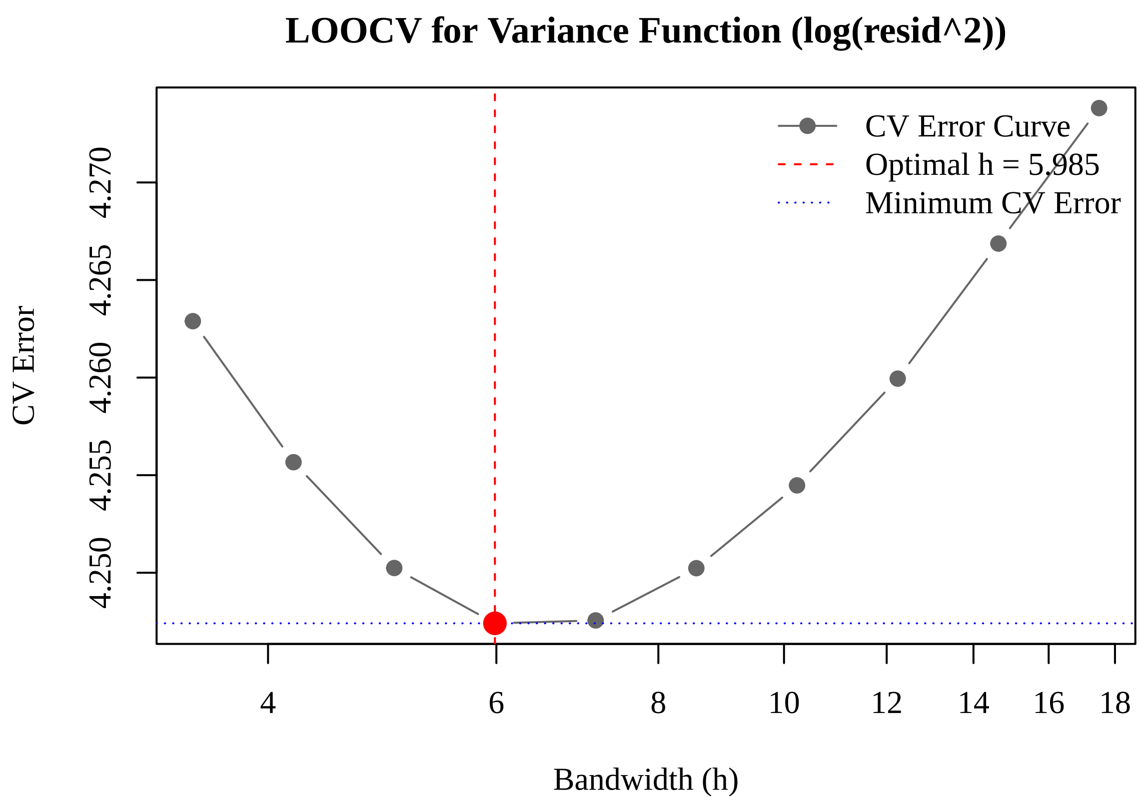



```r
# Optimal bandwidth for q-hat(x)
h_cv2 <- cv.res.var$h.v[which.min(cv.res.var$cv)]
h_cv2
```

```
## [1] 5.984916
```

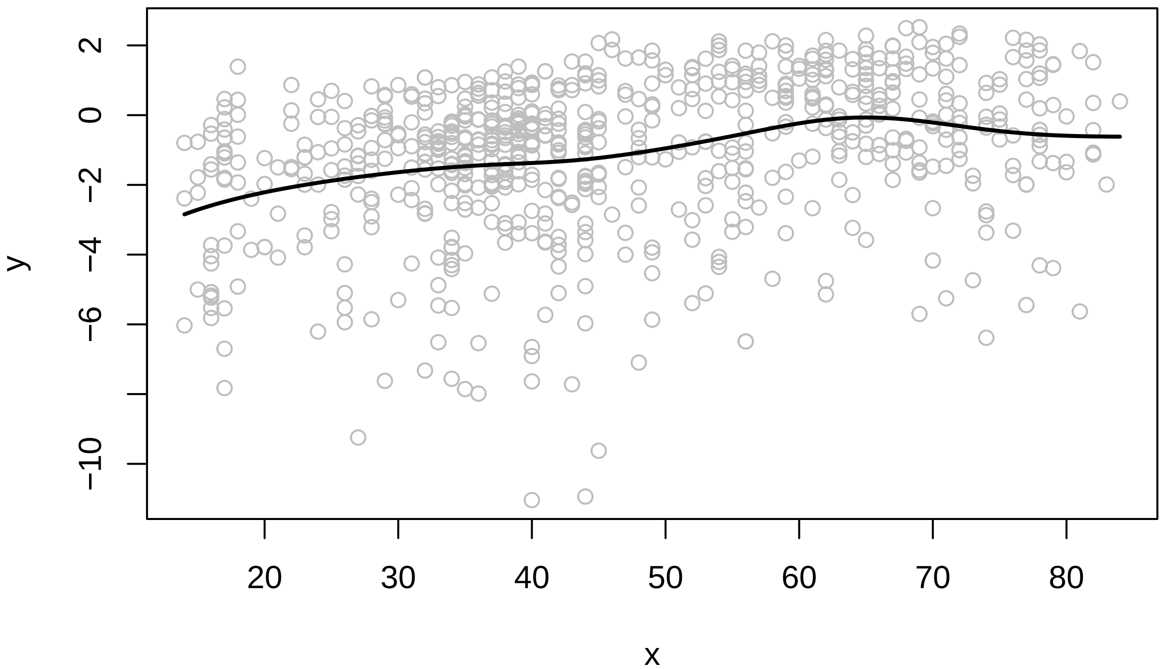
```r
cv.res.var <- h.cv.gcv(Yr, z, type.kernel="normal")
h_cv2 <- cv.res.var$h.v[which.min(cv.res.var$cv)]
min_cv_error_var <- min(cv.res.var$cv)

# --- Set up plot parameters (optional) ---
par(mar = c(5, 5, 3, 2), family = "serif")

# --- Create the main plot ---
plot(cv.res.var$h.v, cv.res.var$cv,
     type = "b",          # Both points and lines
     log = "x",           # Logarithmic x-axis
     main = "LOOCV for Variance Function (log(resid^2))",
     xlab = "Bandwidth (h)",
```

```r
     ylab = "CV Error",
     pch = 19,             # Use solid circles
     col = "gray40")

# --- Key Additions ---

# 1. Add a vertical line at the optimal h
abline(v = h_cv2,
       col = "red",
       lty = 2)           # Dashed line

# 2. Add a horizontal line at the minimum CV error
abline(h = min_cv_error_var,
       col = "blue",
       lty = 3)          # Dotted line

# 3. Add a more visible point at the minimum
points(h_cv2, min_cv_error_var,
       col = "red",
       pch = 19,
       cex = 1.5)         # Larger point

# 4. Add a legend
legend("topright",
       legend = c("CV Error Curve",
                  paste("Optimal h =", round(h_cv2, 3)),
                  "Minimum CV Error"),
       col = c("gray40", "red", "blue"),
       lty = c(1, 2, 3),
       pch = c(19, NA, NA),
       bty = "n")         # No box
```




```r
# Reset plot parameters (optional)
par(mar = c(5.1, 4.1, 4.1, 2.1), family = "")
```

```r
# Define evaluation grid for smooth plots
tg <- seq(min(Yr), max(Yr), length=200)

# Fit q-hat(x) = E[log(residual^2) | x]
q_hat_result <- locpolreg(Yr, z, h=h_cv2, q=1, r=0, tg=tg, type.kernel="normal")
```




```
q_hat <- q_hat_result$mtgr
```

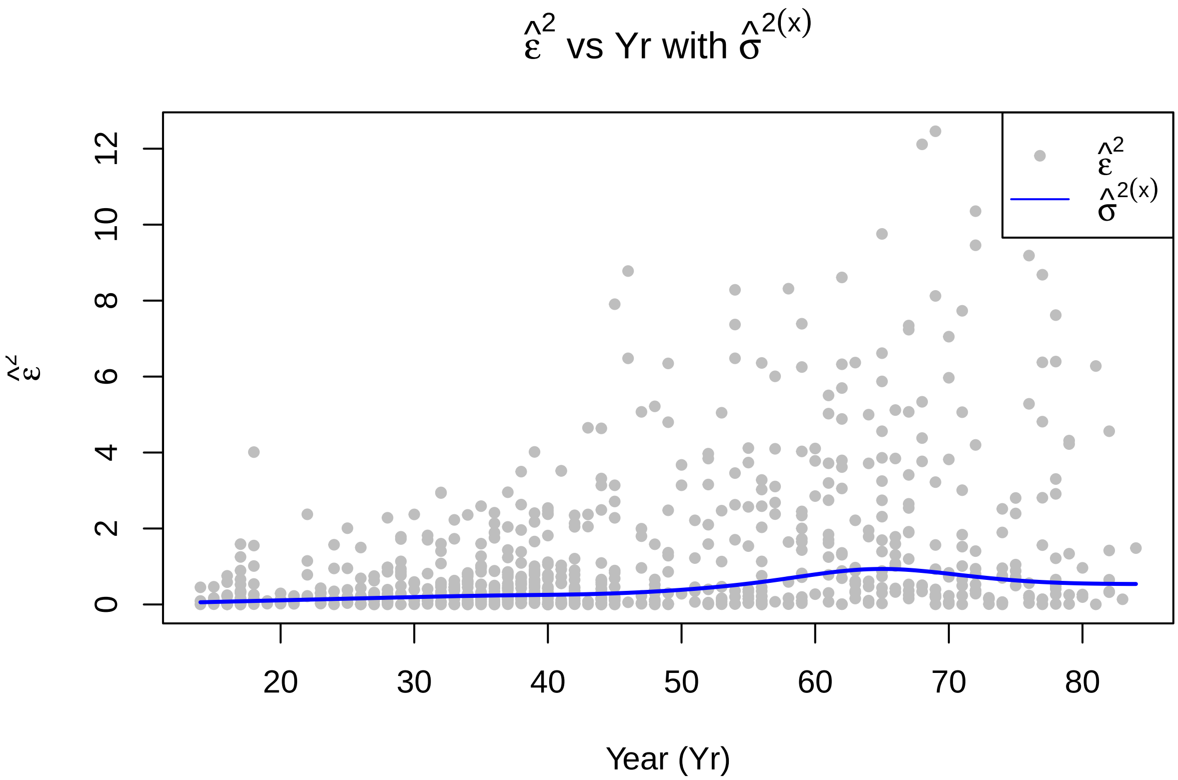
4. Estimate $\sigma^2(x)$ by

$$\hat{\sigma}^2(x) = e^{\hat{q}(x)}.$$

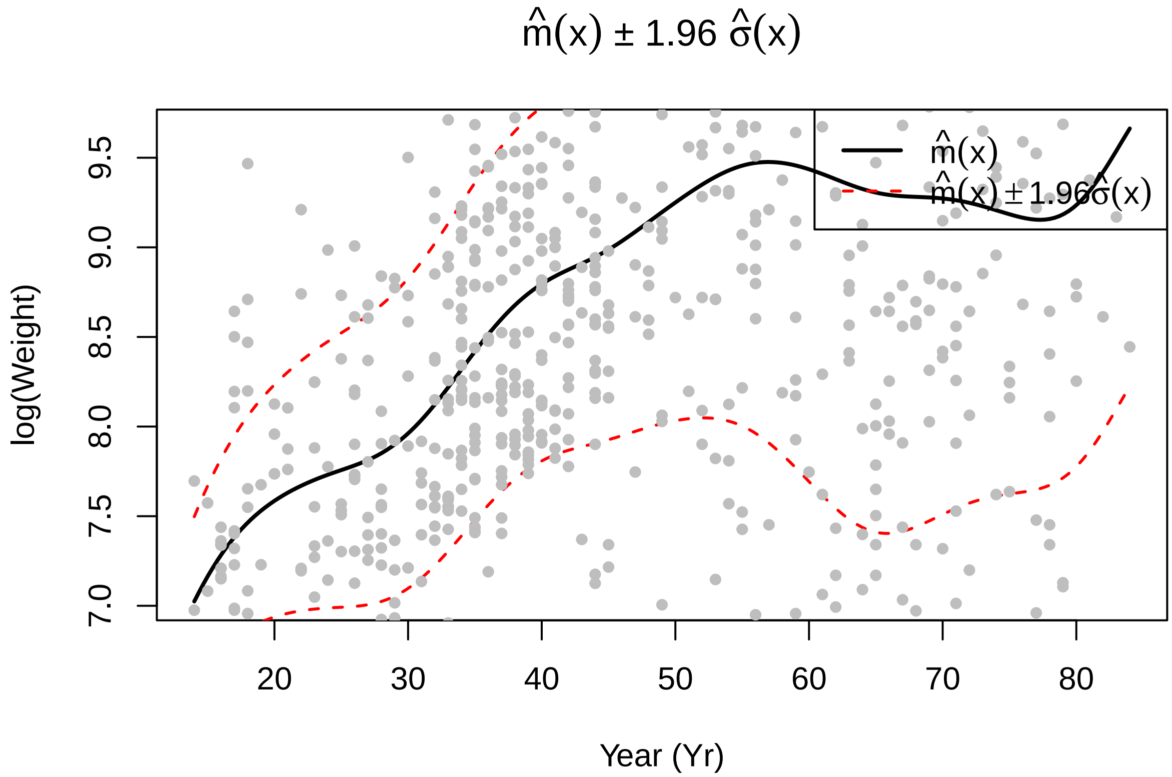
```
sigma2_hat <- exp(q_hat)
sigma_hat  <- sqrt(sigma2_hat)
```

Apply this procedure to estimate the conditional variance of lgWeigth (variable Y) given Yr (variable x). Draw a graphic of $\hat{\epsilon}_i^2$ against x_i and superimpose the estimated function $\hat{\sigma}^2(x)$. Lastly draw the function $\hat{m}(x)$ and superimpose the bands $\hat{m}(x) \pm 1,96\hat{\sigma}(x)$.

```
plot(Yr, eps_hat^2, col="grey", pch=20,
     xlab="Year (Yr)", ylab=expression(hat(epsilon)^2),
     main=expression(paste(hat(epsilon)^2," vs Yr with ",hat(sigma)^2(x))))
lines(tg, sigma2_hat, col="blue", lwd=2)
legend("topright", legend=c(expression(hat(epsilon)^2), expression(hat(sigma)^2(x))),
       col=c("grey","blue"), lty=c(NA,1), pch=c(20,NA))
```



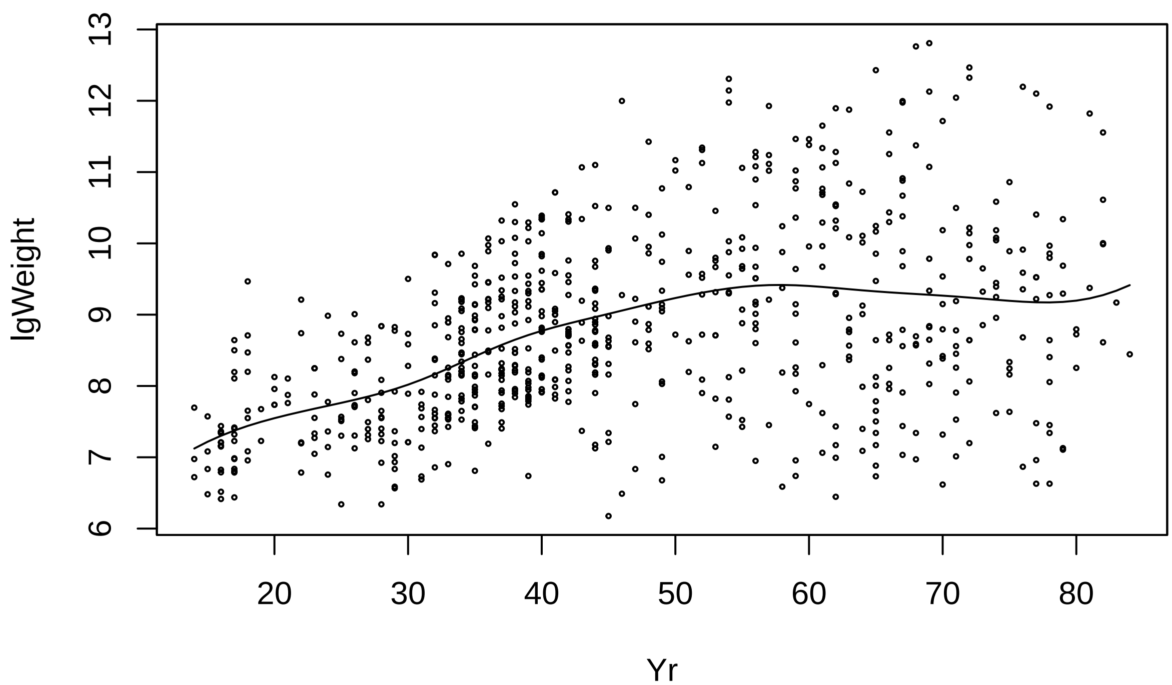

```r
plot(tg, m_hat$mtgr, type="l", lwd=2, col="black",
     main=expression(paste(hat(m)(x)," ± 1.96 ",hat(sigma)(x))),
     xlab="Year (Yr)", ylab="log(Weight)")
lines(tg, m_hat$mtgr + 1.96*sigma_hat, col="red", lty=2, lwd=1.5)
lines(tg, m_hat$mtgr - 1.96*sigma_hat, col="red", lty=2, lwd=1.5)
points(Yr, lgWeight, col="grey", pch=20)
legend("topright",
       legend=c(expression(hat(m)(x)), expression(hat(m)(x) %+-% 1.96*hat(sigma)(x))),
       lty=c(1,2), col=c("black","red"), lwd=c(2,1.5))
```



With sm.regression

1. Fit a nonparametric regression to data (x_i, y_i) and save the estimated values $\hat{m}(x_i)$.

```
h1 <- dpill(Yr, lgWeight)
sm.regression(Yr, lgWeight, h=h1, eval.points=Yr, model="none") -> sm_m
```



```
m_hat_sm <- sm_m$estimate
```

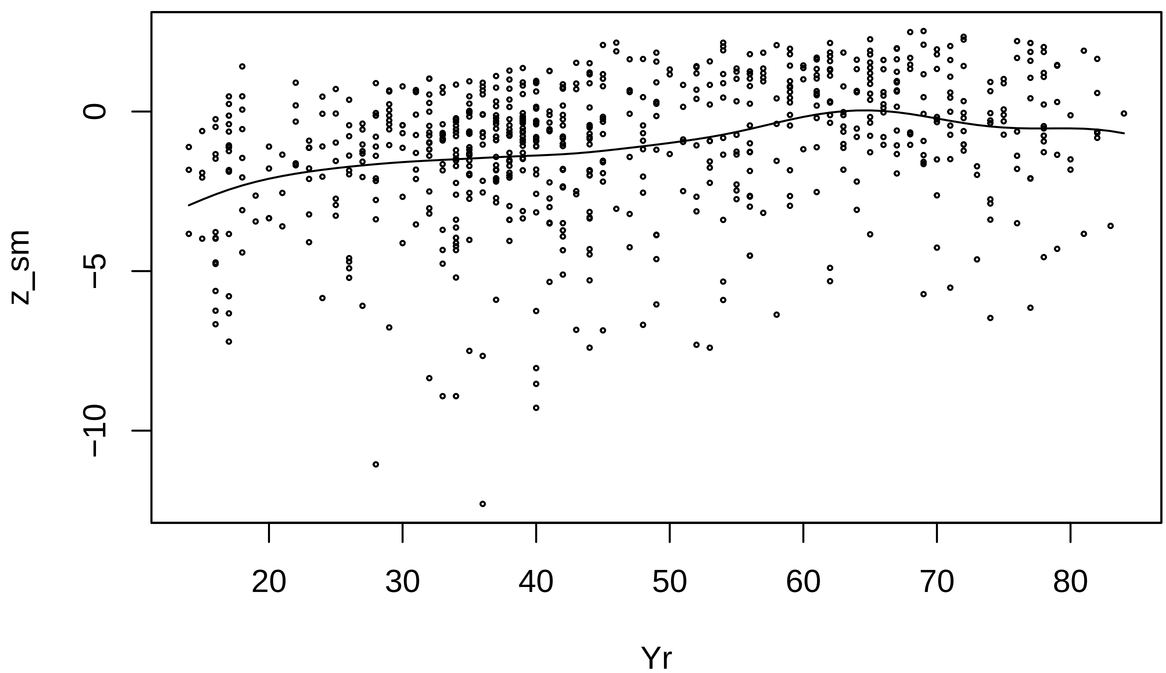
2. Transform the estimated residuals $\hat{\epsilon} = y_i - \hat{m}(x_i)$:

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

```
eps_hat_sm <- lgWeight - sm_m$estimate
z_sm <- log(eps_hat_sm^2)
```

3. Fit a nonparametric regression to data (x_i, z_i) and call the estimated function $\hat{q}(x)$. Observe that $\hat{q}(x)$ is an estimate of $log(\sigma^2(x))$.

```
h2 <- dpill(Yr, z_sm)
sm.regression(Yr, z_sm, h=h2, model="none", eval.points=Yr) -> sm_q
```



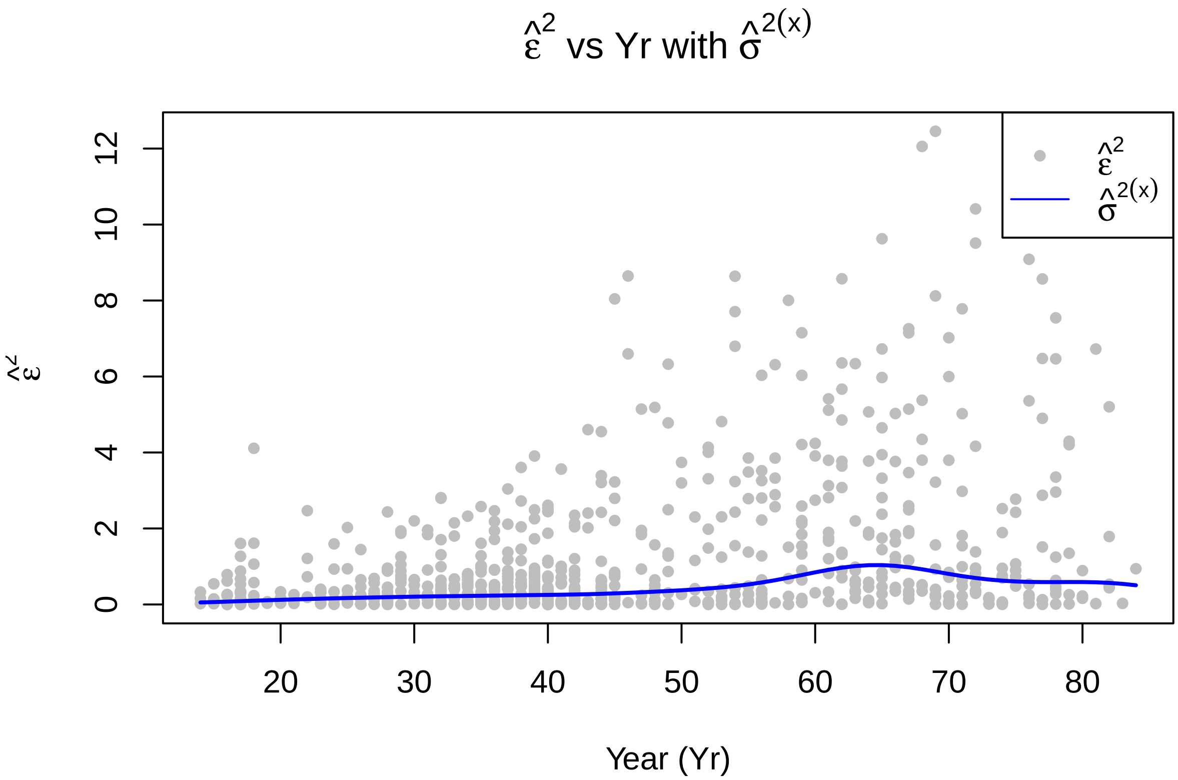
4. Estimate $\sigma^2(x)$ by

$$\hat{\sigma}^2(x) = e^{\hat{q}(x)}.$$

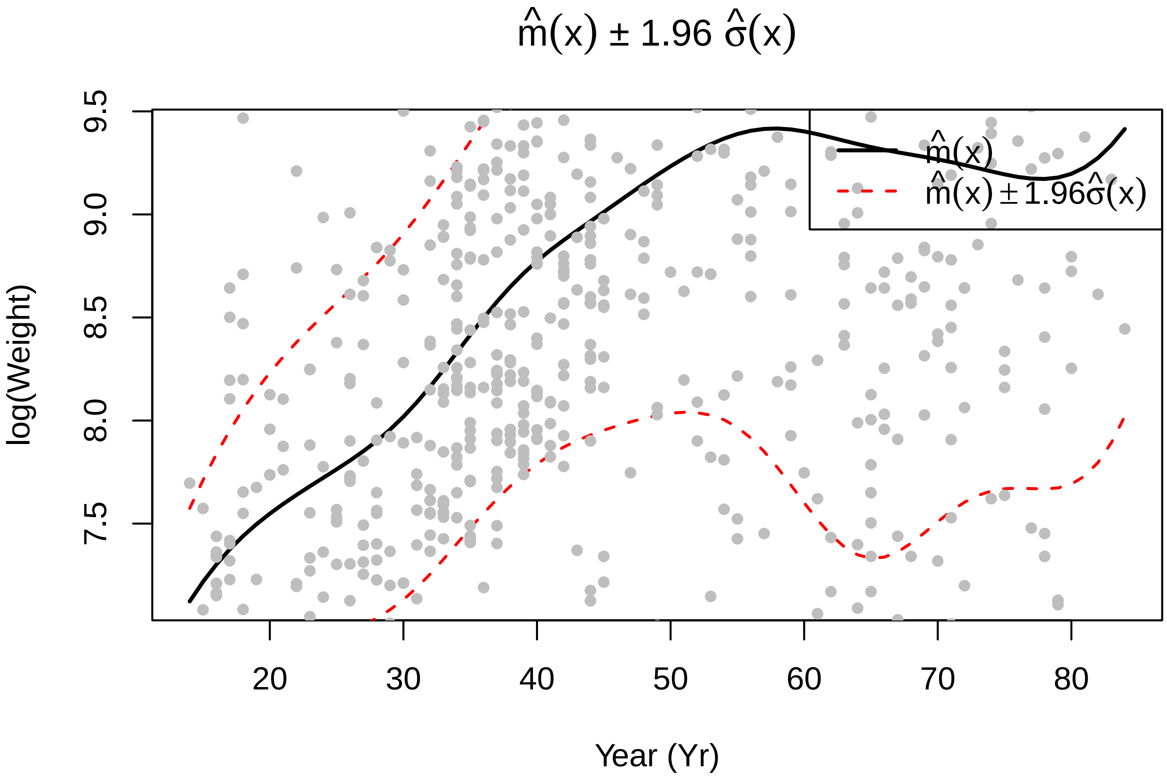
```
sigma2_hat_sm <- exp(sm_q$estimate)
sigma_hat_sm <- sqrt(sigma2_hat_sm)
```

Apply this procedure to estimate the conditional variance of lgWeigth (variable Y) given Yr (variable x). Draw a graphic of $\hat{\epsilon}_i^2$ against x_i and superimpose the estimated function $\hat{\sigma}^2(x)$. Lastly draw the function $\hat{m}(x)$ and superimpose the bands $\hat{m}(x) \pm 1,96\hat{\sigma}(x)$.

```
plot(Yr, eps_hat_sm^2, col="grey", pch=20,
     xlab="Year (Yr)", ylab=expression(hat(epsilon)^2),
     main=expression(paste(hat(epsilon)^2," vs Yr with ",hat(sigma)^2(x))))
lines(Yr, sigma2_hat_sm, col="blue", lwd=2)
legend("topright", legend=c(expression(hat(epsilon)^2), expression(hat(sigma)^2(x))),
       col=c("grey","blue"), lty=c(NA,1), pch=c(20,NA))
```




```r
plot(Yr, m_hat_sm, type="l", lwd=2, col="black",
     main=expression(paste(hat(m)(x)," ± 1.96 ",hat(sigma)(x))),
     xlab="Year (Yr)", ylab="log(Weight)")
lines(Yr, m_hat_sm + 1.96*sigma_hat_sm, col="red", lty=2, lwd=1.5)
lines(Yr, m_hat_sm - 1.96*sigma_hat_sm, col="red", lty=2, lwd=1.5)
points(Yr, lgWeight, col="grey", pch=20)
legend("topright",
       legend=c(expression(hat(m)(x)), expression(hat(m)(x) %+-% 1.96*hat(sigma)(x))),
       lty=c(1,2), col=c("black","red"), lwd=c(2,1.5))
```


$\hat{m}(x) \pm 1.96\, \hat{\sigma}(x)$
log(Weight)
9.5
9.0
8.5
8.0
7.5
20
30
40
50
60
70
80
Year (Yr)
$\hat{m}(x)$
$\hat{m}(x) \pm 1.96\hat{\sigma}(x)$