Laboratoria 11 i 12

Paweł Matławski

9 06 2021

Regresogram i lokalne średnie

Niech $r(x) = 10\sin(2\pi x), n = 500, \sigma = 0.5$ oraz $x_i = i/n$ dla $i = 1, ..., n$

Wygenerujmy n obserwacji $Y_1, ..., Y_n$ postaci $Y_i = r(x_i) + \sigma\epsilon_i$, gdzie $\epsilon_1, ...\epsilon n$ i.i.d $N(0, 1)$.

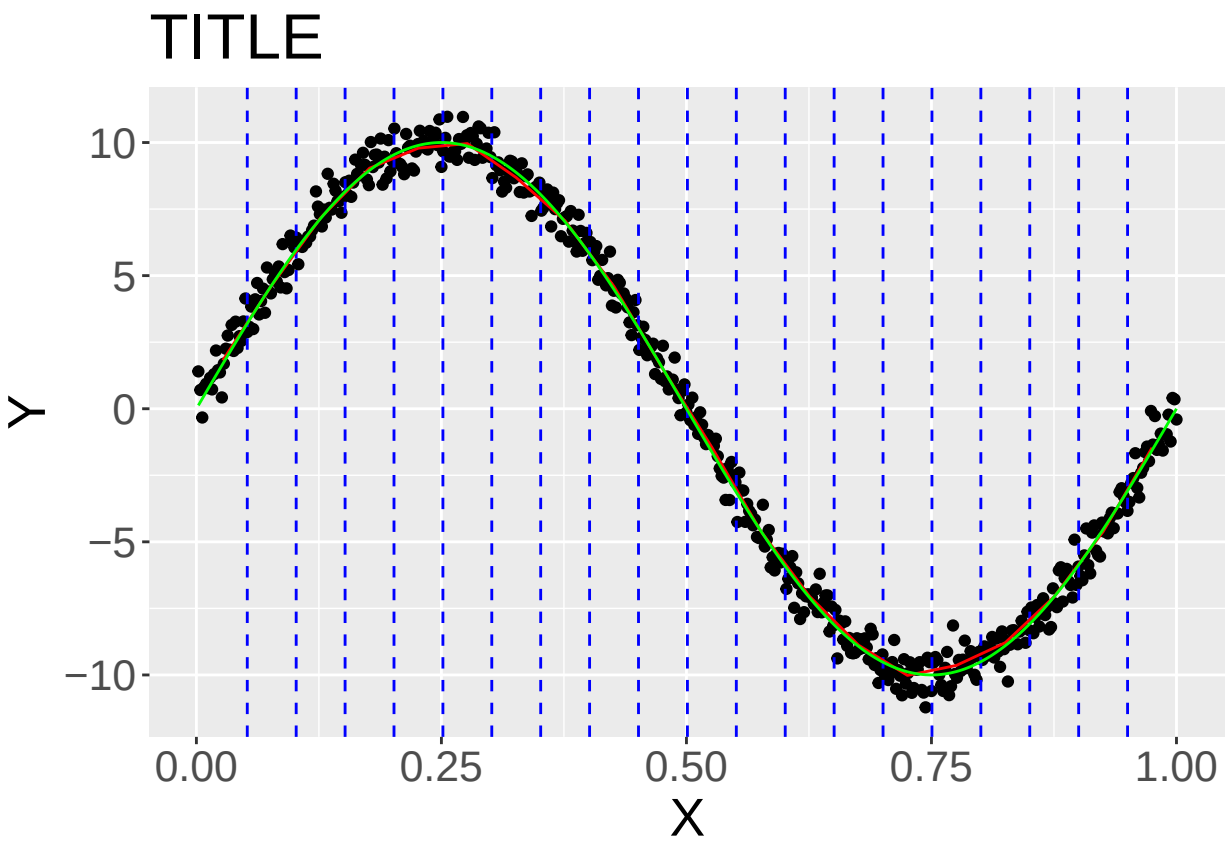
```
library(HoRM)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(ggplot2)
r <- function(x){
  10*sin(2*pi*x)
}

n <- 500
sigma <- 0.5
x_i <- ((1:n)/n)

Y <- r(x_i) + sigma*rnorm(n, mean = 0, sd = 1)
```

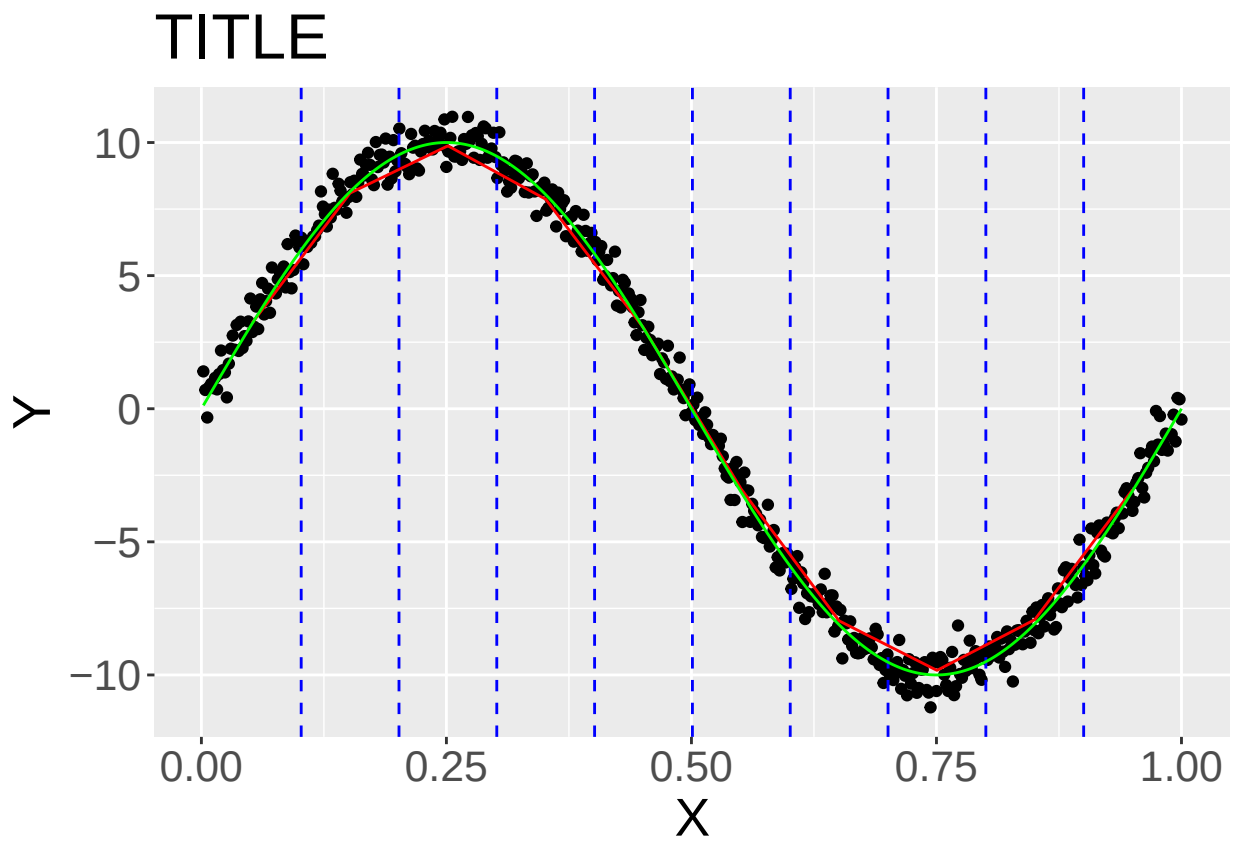
Podzielmy teraz odcinek na 20 przedziałów i skonstruujmy regressogram. Umieścimy na jednym rysunku trzy wykresy: $(x_i, Y_i) : i = 1, ..., n$,$(x_i, r(x_i)) : i = 1, ..., n$ i $(x_i, \hat{r_n}(x_i)) : i = 1, ..., n$.

```
regressogram(x = x_i, y = Y, nbins = 20, show.means = FALSE)+
  geom_line(mapping = aes(x_i, r(x_i)), color = 'green')
```



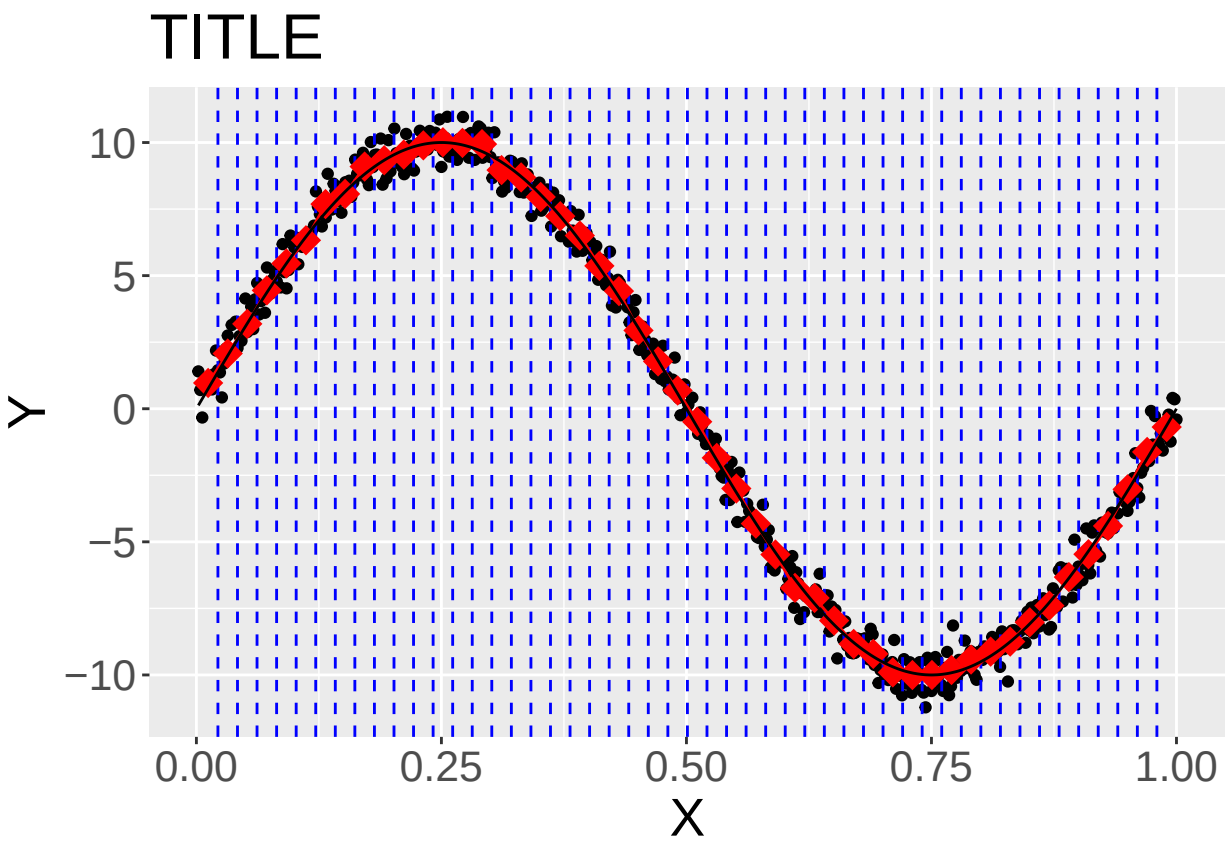
Powtórzmy teraz powyższe czynności dzieląc odcinek na 10 przedziałów.

```
regressogram(x = x_i, y = Y, nbins = 10, show.means = FALSE)+
  geom_line(mapping = aes(x_i, r(x_i)), color = 'green')
```



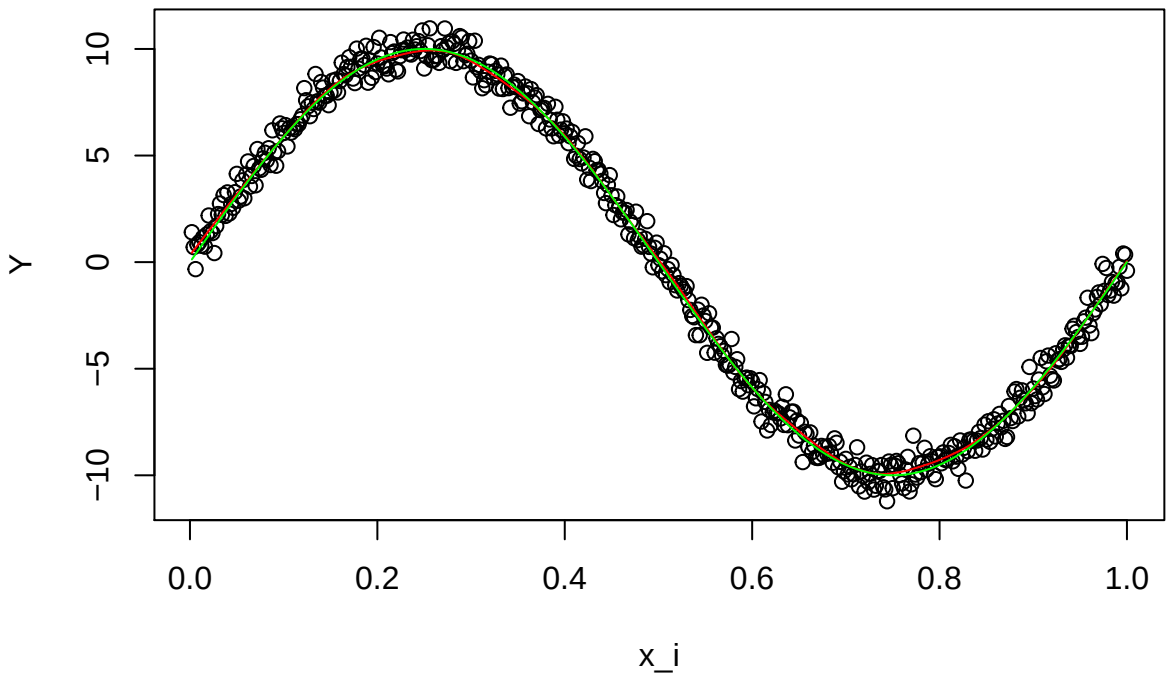
Powtórzmy teraz powyższe czynności dzieląc odcinek na 50 przedziałów.

```
regressogram(x = x_i, y = Y, nbins = 50, show.bins = TRUE)+
  geom_line(mapping = aes(x_i, r(x_i)))
```



Dokonamy teraz takiej samej analizy jak powyżej zmieniając estymator regresji liniowej z *regressogramu* na *local averages*.

```
plot(x_i, Y)
lines(supsmu(x_i, Y), col='red', type = 'l')
lines(x_i, r(x_i), col = 'green')
```



Estymator Nadaraya-Watsona i wielomiany lokalne

Niech r będzie funkcją Dopplera: $r(x) = \sqrt{x(1-x)}\sin(\frac{2.1\pi}{x+0.05}), 0 \leq x \leq 1$ i niech $n = 1000, \sigma = 1$ oraz $x_i = i/n, i = 1, ..., n$. Wygenerujemy n obserwacji $Y_1, ..., Y_n$ postaci $Y_i = r(x_i) + \sigma\epsilon_i$, gdzie $\epsilon_1, ..., \epsilon_n$ i.i.d $N(0, 1)$.

```
library(KernSmooth)
```

```
## KernSmooth 2.23 loaded
## Copyright M. P. Wand 1997-2009
```

```
library(caret)
```

```
## Loading required package: lattice
```

```
r_2 <- function(x){
  sqrt(x*(1-x))*sin((2.1*pi)/(x+0.05))
}

n_2 <- 1000
xi_2 <- seq(from = 1/1000, by = 1/1000, lenght.out = n_2)
```

```
## Warning: In seq.default(from = 1/1000, by = 1/1000, lenght.out = n_2) :
##  extra argument 'lenght.out' will be disregarded
```

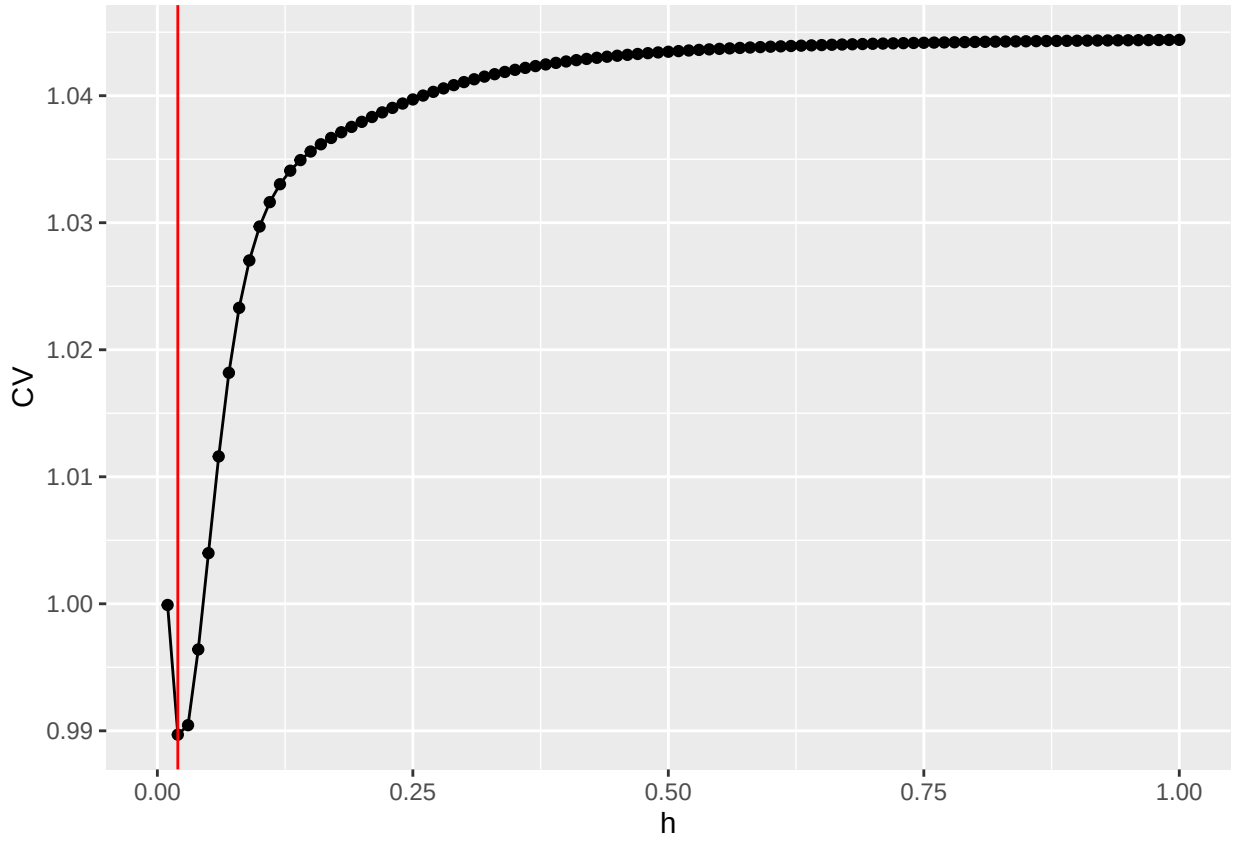
```
sigma_2 <- 1

Y_2 <- r_2(xi_2) + sigma_2*rnorm(n_2, mean = 0, sd = 1)
```

Skonstruujmy teraz estymator jądrowu Nadaraya-Watsona funkcji regresji $r(x)$ oparty na jądrze gaussowskim, dobierając h za pomocą metody *leave-one-out cross-validation*.

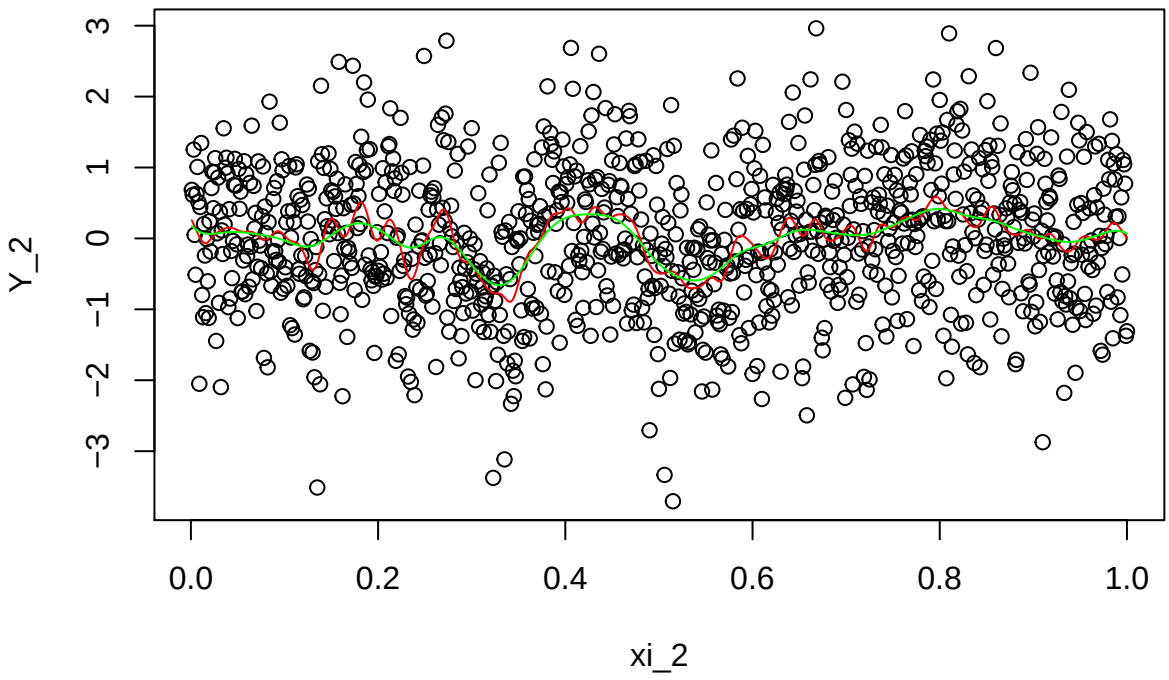
```
kern <- function(x) exp(- x^2 / 2)

loocv <- function(h) {
  Kij <- outer(xi_2, xi_2, function(x, y) kern((x - y) / h))
  S <- Kij / rowSums(Kij)
  mean(((Y_2 - S %*% Y_2) / (1 - diag(S)))^2)
}

h <- seq(0, 1, 0.01)
CV <- sapply(h, loocv)
h_opt <- h[which.min(CV)]
qplot(h, CV) + geom_line() + geom_vline(xintercept = h_opt, color = "red")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
NWK <- ksmooth(xi_2, Y_2, kernel = "normal", bandwidth = h_opt)
LPK <- locpoly(xi_2, Y_2, kernel = 'normal', bandwidth = h_opt)

plot(xi_2, Y_2)
lines(NWK, col = 'red')
lines(LPK, col = 'green')
```



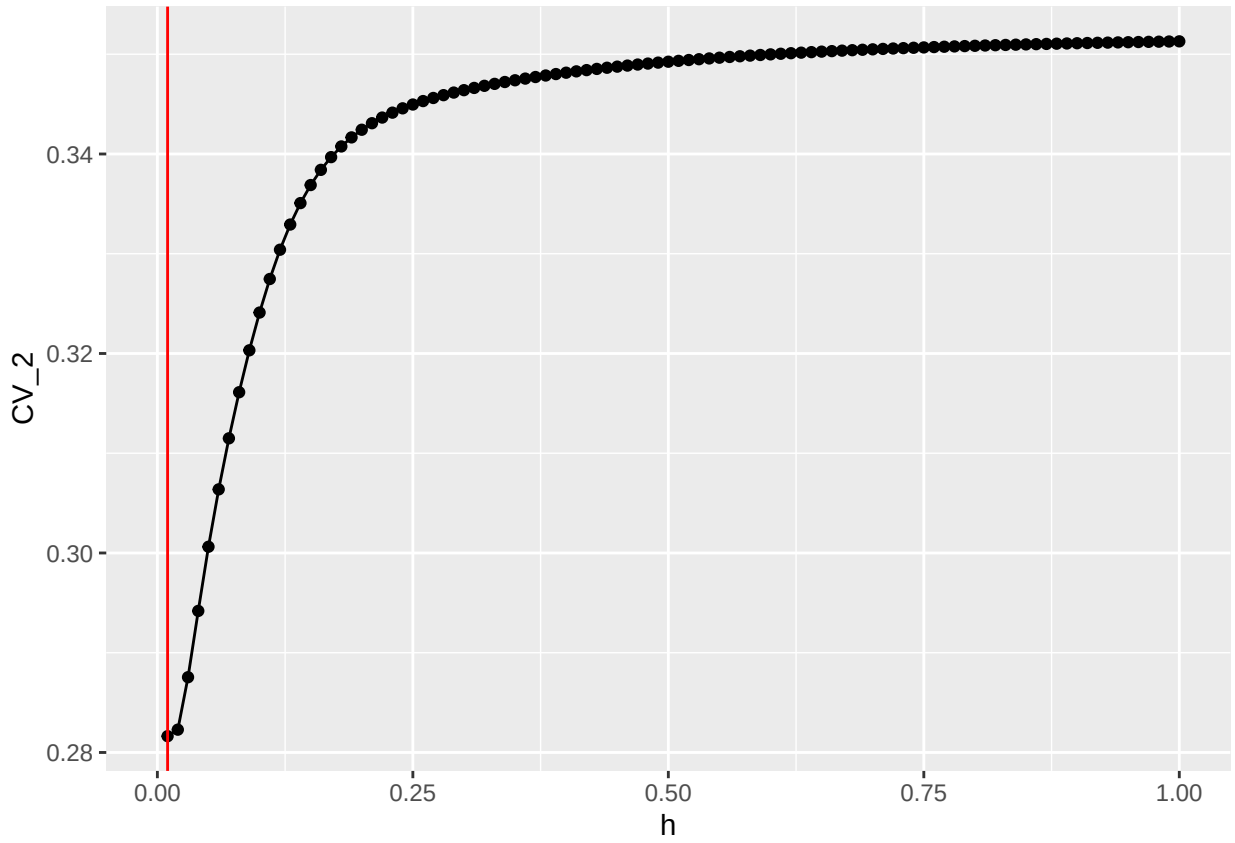
Powtórzmy analizę przy $\sigma = 0.5$

```
sigma_3 <- 0.5

Y_3 <- r_2(xi_2) + sigma_3*rnorm(n_2, mean = 0, sd = 1)
```

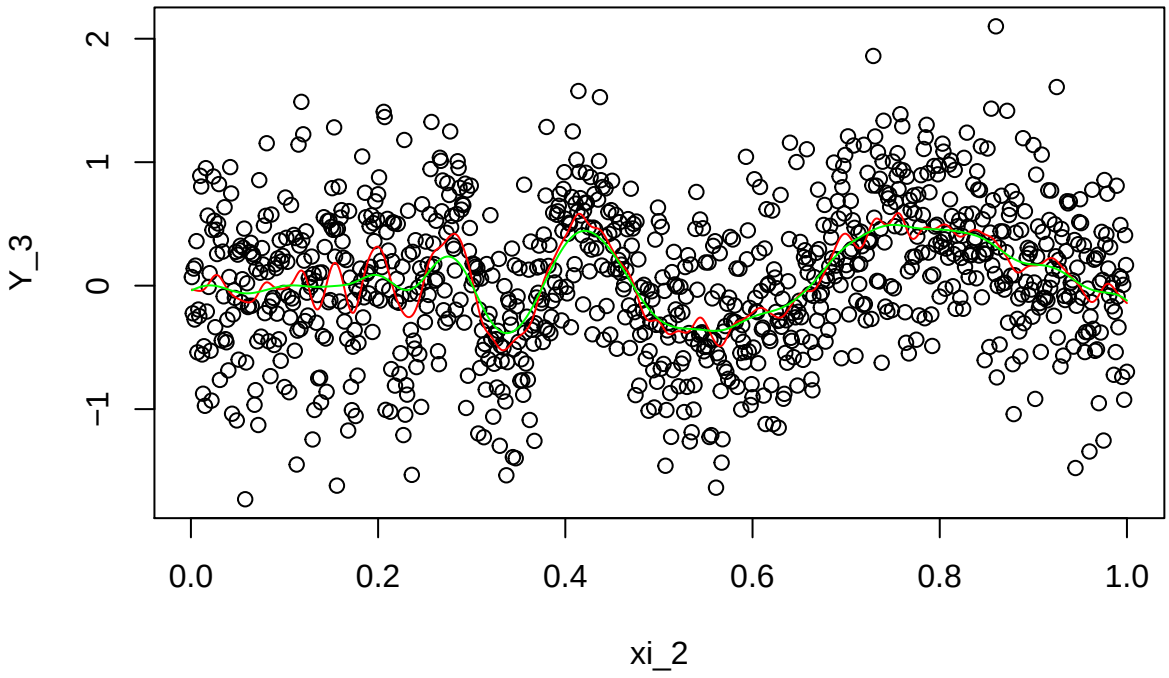
```
loocv_2 <- function(h) {
  Kij <- outer(xi_2, xi_2, function(x, y) kern((x - y) / h))
  S <- Kij / rowSums(Kij)
  mean(((Y_3 - S %*% Y_3) / (1 - diag(S)))^2)
}

CV_2 <- sapply(h, loocv_2)
h_opt_2 <- h[which.min(CV_2)]
qplot(h, CV_2) + geom_line() + geom_vline(xintercept = h_opt_2, color = "red")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
NWK_2 <- ksmooth(xi_2, Y_3, kernel = "normal", bandwidth = h_opt)
LPK_2 <- locpoly(xi_2, Y_3, kernel = 'normal', bandwidth = h_opt)

plot(xi_2, Y_3)
lines(NWK_2, col = 'red')
lines(LPK_2, col = 'green')
```



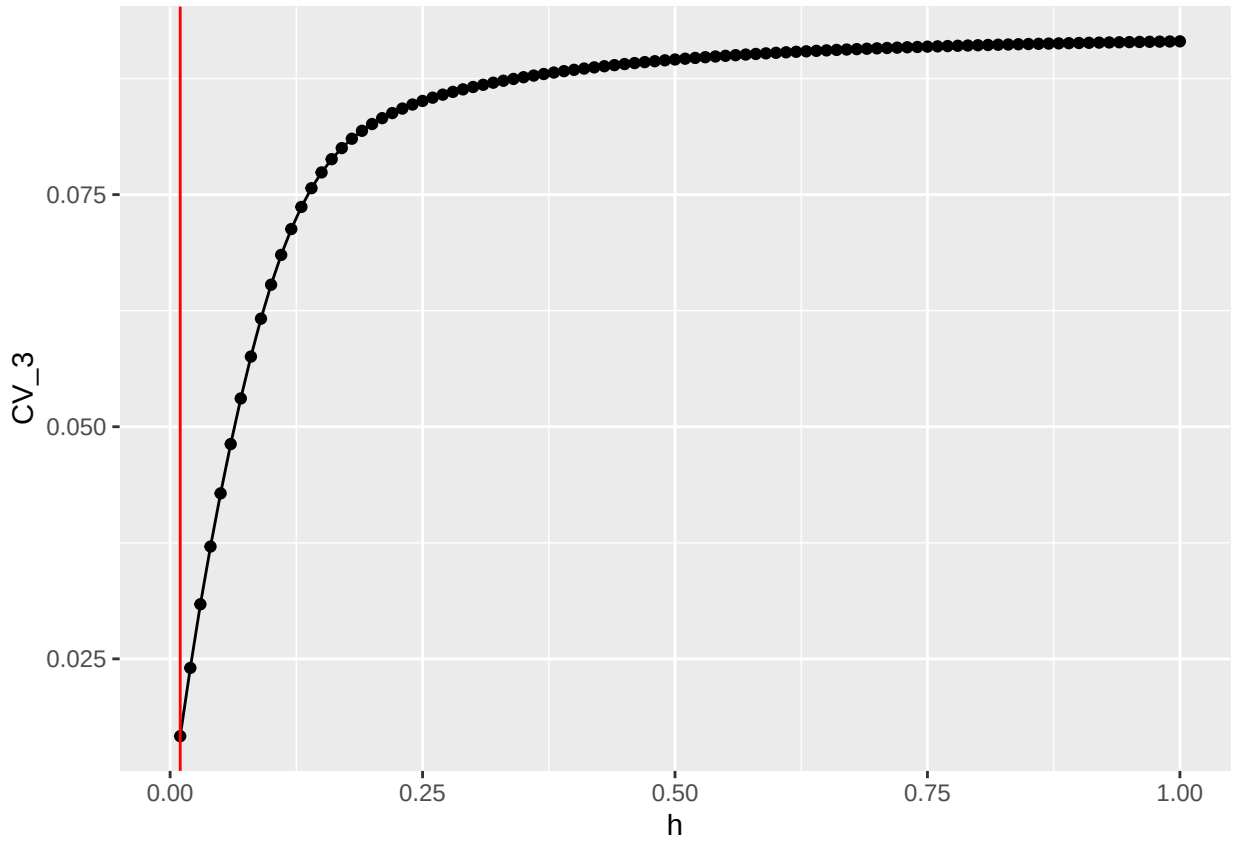
Wykonajmy jeszcze tą samą analizę dla $\sigma = 0.1$.

```
sigma_4 <- 0.1

Y_4 <- r_2(xi_2) + sigma_4*rnorm(n_2, mean = 0, sd = 1)
```

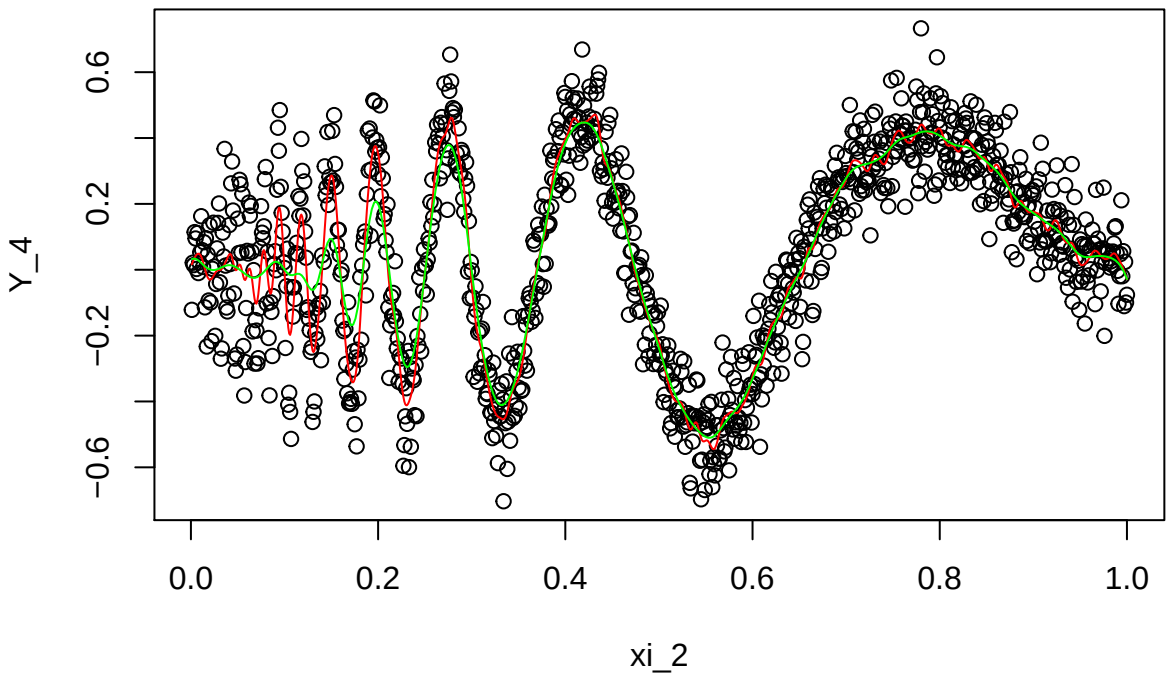
```
loocv_3 <- function(h) {
  Kij <- outer(xi_2, xi_2, function(x, y) kern((x - y) / h))
  S <- Kij / rowSums(Kij)
  mean(((Y_4 - S %*% Y_4) / (1 - diag(S)))^2)
}

CV_3 <- sapply(h, loocv_3)
h_opt_3 <- h[which.min(CV_3)]
qplot(h, CV_3) + geom_line() + geom_vline(xintercept = h_opt_3, color = "red")
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```




```
NWK_3 <- ksmooth(xi_2, Y_4, kernel = "normal", bandwidth = h_opt_3)
LPK_3 <- locpoly(xi_2, Y_4, kernel = 'normal', bandwidth = h_opt_3)

plot(xi_2, Y_4)
lines(NWK_3, col = 'red')
lines(LPK_3, col = 'green')
```


Y_4
0.6
0.2
−0.2
−0.6
0.0
0.2
0.4
0.6
0.8
1.0
xi_2