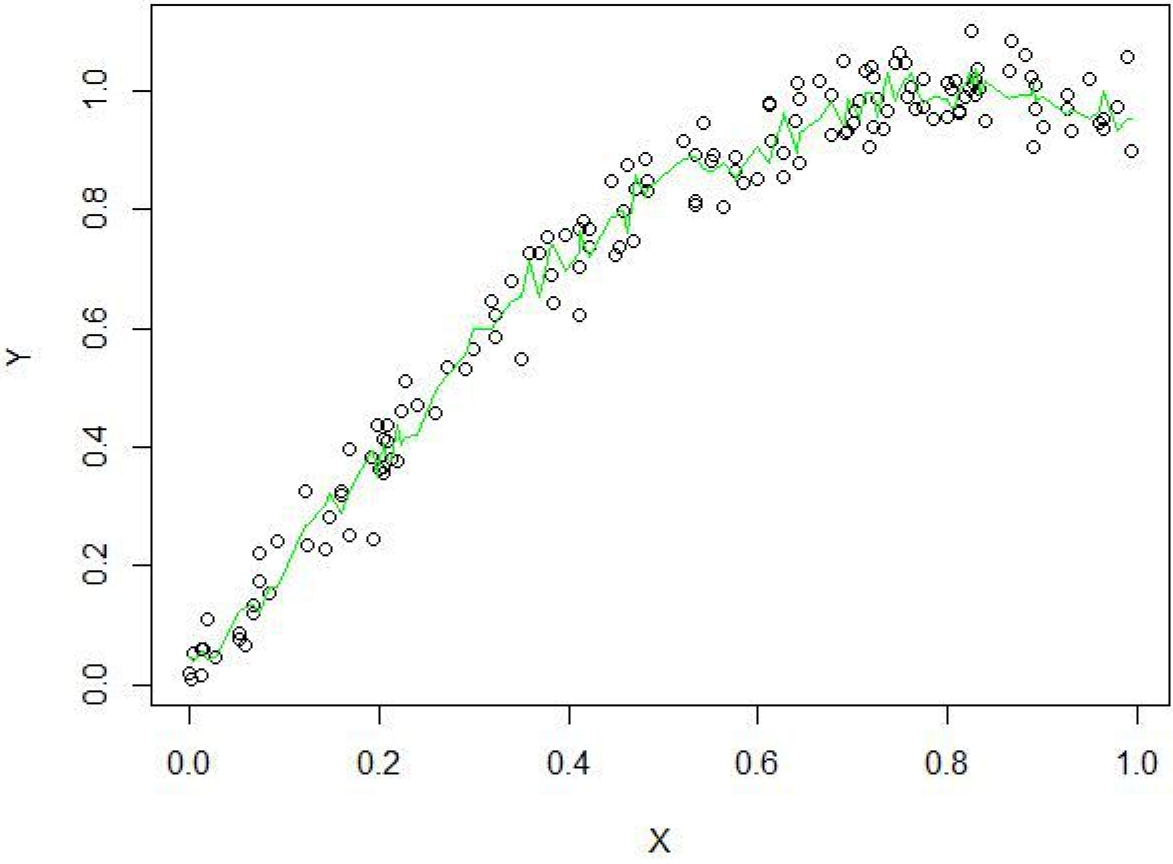
```r
N <- 150 #возьмем 150 наблюдений, а не 100
X <- runif(N, min=0, max=1) #генерируем равномерное
e <- rnorm(N, sd=0.05) #генерируем случайные нормальные ошибки
Y <- sin(2*X)+e #строим модель

#1,2
k <- 5 #будем брать пятерых соседей, как предложено
reord <- function(i){
  d <- abs(X-X[i])
  data <- cbind(X, e, d)
  data <- data[order(data[,3]),]
  return(data) #now ordered
}

#3
f5 <- rep(0, N) #сначала заполним нулями
for(i in (1:N)){
  f5[i] <- mean(sin(2*reord(i)[,1][2:k+1])+reord(i)[,2][2:k+1])
}
data <- cbind(X, f5)
data <- data[order(data[,1]),]
plot(X, Y)
lines(data, col = "green")

```

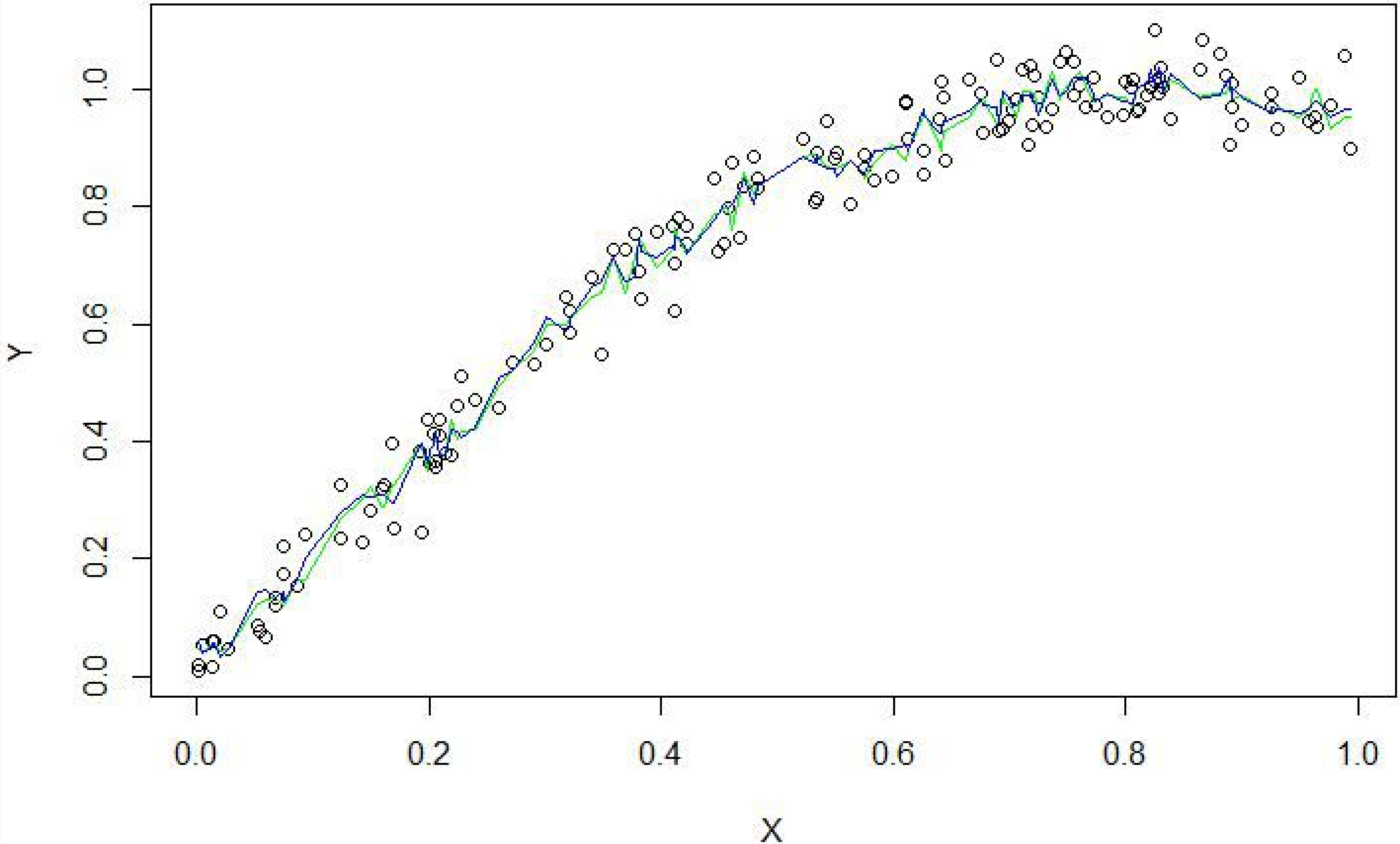


0.0
0.2
0.4
0.6
0.8
1.0
Y
0.0
0.2
0.4
0.6
0.8
1.0
X

```
#4
err <- f5 - Y #получаем вектор ошибок
delta <- rep(0, N) #сначала заполним нулями
for (i in (1:N)){
  if (abs(err[i])<1){ #не забываем про носитель
    delta[i] <- (1-abs(err[i]))**4
  }
}
print(err) # посмотрим что получилось
print(delta) # посмотрим что получилось
```

```
C:/Users/Лев/Downloads/
  [1]  7.900309e-04 -1.487967e-02 -3.172521e-02 -1.863363e-02  3.751306e-02  7.256384e-02 -5.946206e-02
  [8]  2.446431e-02 -4.844695e-02  1.042393e-01 -1.091088e-01 -4.457233e-05  4.155583e-02 -6.682585e-02
 [15]  2.432211e-02 -7.548872e-02 -8.050226e-02 -2.966342e-02 -3.172789e-02  6.506553e-02 -6.602891e-02
 [22] -4.220746e-02 -3.963526e-02  9.106048e-02 -1.252626e-02 -7.877198e-03 -4.168844e-02 -4.135941e-03
 [29] -1.012200e-01 -2.743963e-02  6.529661e-02  6.178483e-03 -7.517406e-02  1.365245e-02  1.029608e-02
 [36] -4.408882e-03  6.388950e-02  9.725067e-03 -3.619784e-02  4.290309e-02  7.691606e-02  5.184847e-02
 [43] -6.843349e-02  4.831420e-03  4.905925e-02  9.846390e-02  2.761101e-02  1.096763e-01  7.426325e-02
 [50] -6.353069e-02 -4.132072e-02  3.216290e-02  4.636697e-02 -8.351046e-03 -3.028348e-02  5.909644e-02
 [57] -1.989904e-02  2.559933e-02  5.113250e-02  3.308707e-02 -1.203673e-02 -2.823435e-02 -2.822353e-02
 [64]  2.905310e-02  2.963621e-02 -8.714717e-02  1.264950e-03 -1.942977e-02 -1.992255e-02  5.927877e-02
 [71]  3.639618e-02  3.611015e-02 -9.430967e-02  1.859774e-02 -3.598844e-02  1.018915e-01  5.357881e-02
 [78] -1.169931e-02 -5.951189e-02  3.065549e-02  5.166567e-03 -4.545260e-02  3.103510e-02 -3.893016e-02
 [85]  1.410134e-01  3.620349e-02  1.889926e-02  1.461127e-01 -1.058034e-01 -5.935796e-02 -6.696574e-02
 [92] -3.104367e-02  3.788496e-02 -1.139288e-01 -2.945600e-02 -7.311890e-03  6.025017e-02 -3.349139e-02
 [99] -3.176392e-02 -8.025448e-03 -5.213777e-02  2.285032e-02 -2.779513e-02  2.115671e-02  1.385904e-02
[106] -6.150527e-02  3.949681e-02 -2.459289e-02  7.427726e-02 -5.778548e-02  4.668113e-02 -4.237554e-03
[113] -8.674281e-02  2.279700e-02 -5.513053e-02 -9.592120e-02  6.168969e-02  8.076532e-02 -5.979913e-02
[120] -4.744986e-02 -5.807254e-02 -3.186585e-02  5.162515e-02  5.438966e-02 -1.909687e-03 -9.768556e-02
[127]  5.324474e-02  3.601772e-02  2.436433e-02 -1.324166e-02  4.072718e-02  5.035066e-02  5.475019e-02
[134] -4.919086e-02  6.274771e-02 -6.394659e-02  7.597646e-02 -4.056684e-02 -6.346491e-02 -1.293674e-02
[141]  1.414931e-02  3.624450e-02 -1.189149e-01  4.838815e-02  2.698896e-02 -9.950735e-02  6.653237e-02
[148] -3.601382e-02 -4.554689e-02 -6.259778e-04
> print(delta) # посмотрим что получилось
  [1] 0.9968436 0.9417966 0.8790114 0.9275230 0.8581820 0.7398371 0.7825377 0.9056756 0.8198455 0.6438254
 [11] 0.6299392 0.9998217 0.8438539 0.7583170 0.9062038 0.7305482 0.7148298 0.8865222 0.8790017 0.7640551
 [21] 0.7609108 0.8415614 0.8506381 0.6825586 0.9508286 0.9688616 0.8433870 0.9835586 0.6525498 0.8946770
 [31] 0.7633000 0.9755142 0.7315433 0.9464984 0.9594474 0.9824808 0.7679067 0.9616635 0.8628823 0.8391192
 [41] 0.7260471 0.8081854 0.7531049 0.9808139 0.8177374 0.6605908 0.8940465 0.6283357 0.7344293 0.7690847
 [51] 0.8446822 0.8774231 0.8270374 0.9670119 0.8842584 0.7837552 0.9227483 0.9014679 0.8106293 0.8740766
 [61] 0.9527154 0.8917563 0.8917960 0.8887547 0.8866217 0.6943894 0.9949498 0.9245168 0.9226598 0.7831479
 [71] 0.8621723 0.8631964 0.6728510 0.9276587 0.8636325 0.6506018 0.8023019 0.9540176 0.7823719 0.8829022
 [81] 0.9794933 0.8302139 0.8815200 0.8531390 0.5444343 0.8628621 0.9265192 0.5316212 0.6393401 0.7828842
 [91] 0.7578624 0.8814888 0.8568563 0.6164167 0.8872804 0.9710717 0.7799181 0.8726155 0.8788708 0.9682826
[101] 0.8071995 0.9116841 0.8933696 0.9180211 0.9457057 0.7757600 0.8511287 0.9051982 0.7343849 0.7881324
[111] 0.8259481 0.9831572 0.6956205 0.9118831 0.7970531 0.6680749 0.7751504 0.7140121 0.7814165 0.8232872
[121] 0.7871724 0.8785008 0.8089471 0.7995559 0.9923831 0.6628750 0.8034353 0.8635276 0.9060469 0.9480761
[131] 0.8467760 0.8133043 0.7983372 0.8172848 0.7716601 0.7677194 0.7290077 0.8473423 0.7693009 0.9492486
[141] 0.9445927 0.8627152 0.6026586 0.8200482 0.8963365 0.6575377 0.7592714 0.8635416 0.8298859 0.9974984
> #Как мы видим, полученные значения весьма близки к 1..
> 
```

```
#5
kvar <- round(k/delta)
fk <- rep(0, N) #сначала заполним нулями
for(i in (1:N)){
  fk[i] <- mean(sin(2*reord(i)[,1][2:kvar[i]+1])+reord(i)[,2][2:kvar[i]+1])
}
data <- cbind(X, fk)
data <- data[order(data[,1]),]
lines(data, col = "blue")
```



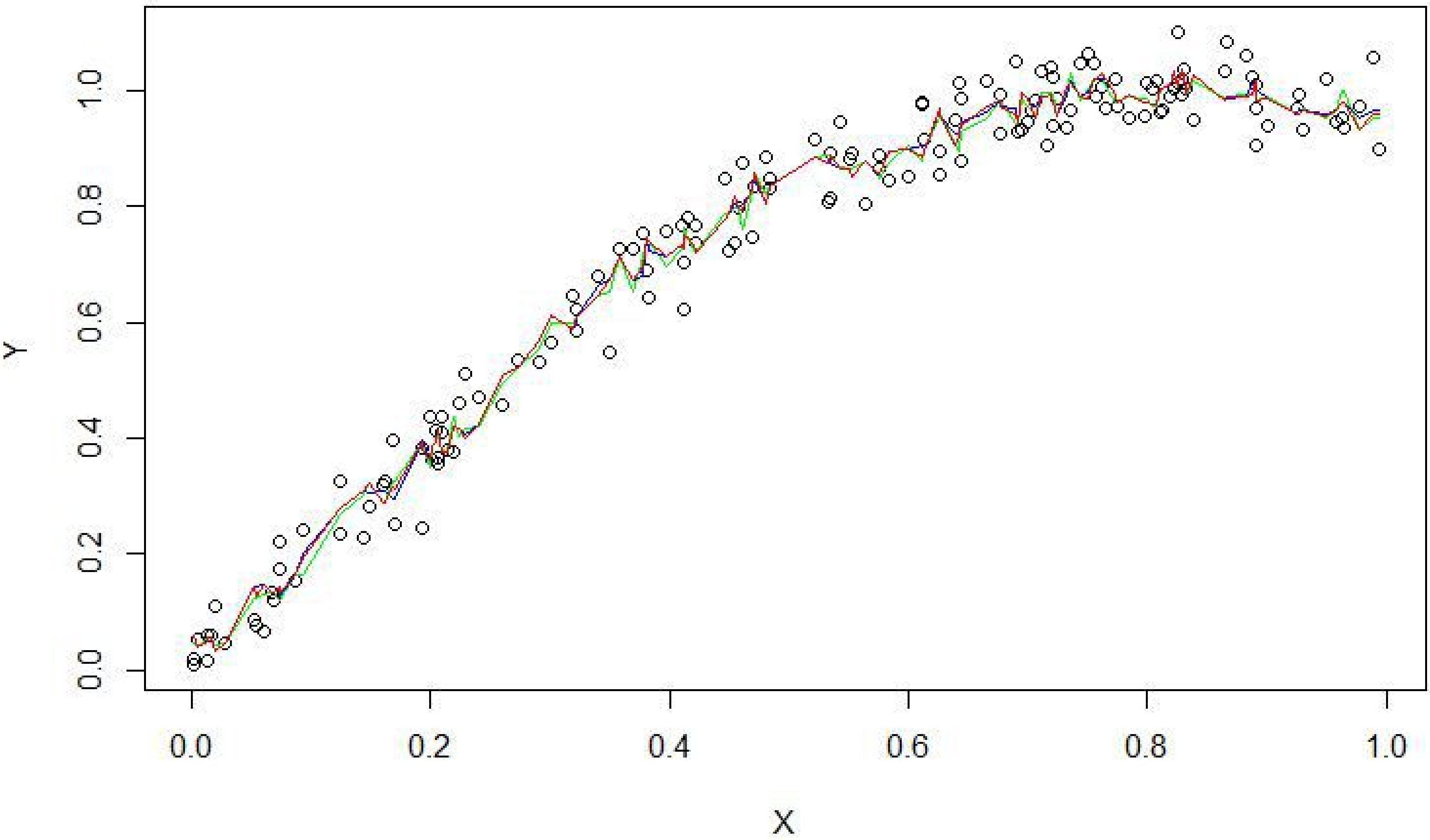
Y
0.0
0.2
0.4
0.6
0.8
1.0
X
0.0
0.2
0.4
0.6
0.8
1.0

```r
#6
for (step in (1:15)){ #сделаем на всякий случай 15 раз, а не 10
  err <- fk - Y #получаем вектор ошибок
  delta <- rep(0, N) #сначала заполним нулями
  for (i in (1:N)){
    if (abs(err[i])<1){ #не забываем про носитель
      delta[i] <- (1-abs(err[i]))**4
    }
  }
  kvar <- round(k/delta)
  fk <- rep(0, N) #сначала заполним нулями
  for(i in (1:N)){
    fk[i] <- mean(sin(2*reord(i)[,1][2:kvar[i]+1])+reord(i)[,2][2:kvar[i]+1])
  }
}
data <- cbind(X, fk)
data <- data[order(data[,1]),]
lines(data, col = "red")

#Ответ: Если сравнить по MSE, то
#MSE of estimator with k = 5:
mean((f5 - Y)**2)
#MSE of estimator with k = k(i):
mean((fk - Y)**2)

#Как мы видим, MSE уменьшилось, и наша оценка улучшилась
```

43:21 (Top Level) R Script



Y
X
0.0
0.2
0.4
0.6
0.8
1.0

```
> #Ответ: Если сравнить по MSE, то
> #MSE of estimator with k = 5:
> mean((f5 - Y)**2)
[1] 0.003029655
> #MSE of estimator with k = k(i):
> mean((fk - Y)**2)
[1] 0.002896992
>
> #Как мы видим, MSE уменьшилось, и наша оценка улучшилась
>
```

Новиков Лев. ДЗ-02

Теоретическая часть.

1. Треугольное ядро, $K(x) = (1-|x|)\mathbb{1}_{\{|x|\leq 1\}}$

Даны $\{(X_i, Y_i)\}_{i=1}^{6}$, $X_i = i \ \forall i$.

Оценка Надарая-Ватсона имеет вид

$$\widehat{m}_h(x) = \frac{\sum_{i=1}^{n} K_h(x-x_i)\cdot y_i}{\sum_{i=1}^{n} K_h(x-x_i)}.$$

Найти сглаживающую матрицу H и эффективные степени свободы.

i) $h = 1/2$, т.е. $K_h(x-x_i) = K\left(\frac{x-x_i}{h}\right) =$
$= K(2(x-x_i))$. Поскольку треугольное ядро имеет носитель $[-1;1]$, для $|x-x_i| \geq \frac{1}{2}$ получим $K(2(x-x_i)) = 0$.

А это значит, что $K(2\cdot(1-1)) = K(2\cdot(2-2)) =$
$= K(2\cdot(3-3)) = K(2\cdot(4-4)) = K(2\cdot(5-5)) = K(2\cdot(6-6)) = 1$

$\forall x \neq x_i \quad K(2(x-x_i)) = 0$ (т.к. $\mathbb{1}_{\{2|x-x_i|\geq 2\}} = 1$)

Таким образом, $\sum_{i=1}^{6} K_h(x-x_i) = 1$ независимо от x, откуда оценка Н-В $\widehat{m}_h(x) = \sum_{i=1}^{6} y_i$

$$\hat{m}_h(x) = H \cdot y \Rightarrow \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_6 \end{pmatrix} = H \cdot \begin{pmatrix} y_1 \\ \vdots \\ y_6 \end{pmatrix}$$

$$H = \begin{pmatrix} 1&0&0&0&0&0 \\ 0&1&0&0&0&0 \\ 0&0&1&0&0&0 \\ 0&0&0&1&0&0 \\ 0&0&0&0&1&0 \\ 0&0&0&0&0&1 \end{pmatrix} = I_6 \quad tr\, H = tr\, I_6 = 6.$$

Ответ: $H = I_6$, число степеней свободы — 6.

ii) $h = \frac{3}{2} \Rightarrow K_h(x - x_i) = K\left(\frac{x - x_i}{3/2}\right) = K\left(\frac{2}{3}\cdot(x - x_i)\right)$

$\forall x : |x - x_i| = 0,\ K_h(x - x_i) = K_h(0) = 1$

$\forall x : |x - x_i| = 1,\ K_h(x - x_i) = K\left(\pm\frac{2}{3}\right) = 1 - \frac{2}{3} = \frac{1}{3}$

$\forall x : |x - x_i| \geq 2,\ K_h(x - x_i) = 0 \quad \left(K\left(\frac{4+2n}{3}\right) = 0\right)$

Получается, что $\sum_{i=1}^{6} K\left(\frac{2}{3}(1 - x_i)\right) = 1 + \frac{1}{3} = \frac{4}{3} = \sum_{i=1}^{6} K\left(\frac{2}{3}(6 - x_i)\right)$, и $\sum_{i=1}^{6} K\left(\frac{2}{3}(2 - x_i)\right) = \sum_{i=1}^{6} K\left(\frac{2}{3}(3 - x_i)\right) = \sum_{i=1}^{6} K\left(\frac{2}{3}(4 - x_i)\right) = \sum_{i=1}^{6} K\left(\frac{2}{3}\cdot(5 - x_i)\right) = \frac{1}{3} + 1 + \frac{1}{3} = \frac{5}{3}$.

Поэтому в этом случае из $\hat{m}_h(x) = H \cdot y$ получаем

$$\underbrace{\begin{pmatrix} 3/4 & 1/4 & 0 & 0 & 0 & 0 \\ 1/5 & 3/5 & 1/5 & 0 & 0 & 0 \\ 0 & 1/5 & 3/5 & 1/5 & 0 & 0 \\ 0 & 0 & 1/5 & 3/5 & 1/5 & 0 \\ 0 & 0 & 0 & 1/5 & 3/5 & 1/5 \\ 0 & 0 & 0 & 0 & 1/4 & 3/4 \end{pmatrix}} \cdot \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix}$$

Матрица $H \rightarrow$

Число степеней свободы $f = tr\, H = 2 \cdot \frac{3}{4} + 4 \cdot \frac{3}{5} = 3(0.5 + 0.8) = 3.9$

2. Квадратичная модель $Y_i = (1+X_i)^2 + \varepsilon_i$ (т.е. $\varphi(x) = (1+x)^2$). $\mathbb{E}\varepsilon_i = 0$, $\mathbb{E}\varepsilon_i^2 = V(\varepsilon_i) = \sigma^2$. $X \sim U_{[0,1]}$. $K(x)$ – гауссова ядро, т.е. $K(x) = \frac{1}{\sqrt{2\pi}} \cdot e^{-x^2/2}$. Найти h_{opt}.

Поскольку $f(x)$ имеет компактный $\operatorname{supp} f$ ($X \sim U_{[0,1]}$), все $\{X_i\}_{i=1}^n$ – независимы и распределены как $\sim f(x)$, можем воспользоваться теоремой о том, что в этом случае $MISE(h) = AMISE(h) + o(h^4 + \frac{1}{nh})$

$$AMISE \simeq \mathbb{E}\left(\int(\hat{g}_n(x) - g(x))^2 dx\right) =$$

$$g(x) = Y = (1+x)^2.$$

$$= \frac{\sigma^2}{nh} \cdot \int_{\mathbb{R}} (K(x))^2 dx \cdot \int_0^1 \frac{1}{f(x)} dx + \frac{h^4}{4} \cdot \left(\int_{\mathbb{R}} x^2 K(x) dx\right)^2 \cdot$$

$$\cdot \int_0^1 \left(g''(x) + 2g'(x) \cdot \frac{f'(x)}{f(x)}\right)^2 dx =$$

$g'(x) = 2(x+1)$; $g''(x) = 2$.

Т.к. $f(x) = \mathbb{1}_{\{x \in [0,1]\}}$, $\frac{1}{f(x)} = \mathbb{1}_{\{x \in [0,1]\}}$;

$f'(x) = \frac{\partial}{\partial x}(\mathbb{1}) = 0$

Получаем:

$$AMISE(h) \approx \frac{\sigma^2}{nh} \cdot \frac{1}{2\sqrt{\pi}} \cdot 1 + \frac{h^4}{4} \cdot 1 \cdot \int_0^1 g''(x)^2 dx =$$

$$= \frac{\sigma^2}{2n\sqrt{\pi} \cdot h} + \frac{h^4}{4} \cdot 4 = h^4 + \frac{\sigma^2}{2n\sqrt{\pi}} \cdot h^{-1}.$$

Поскольку $AMISE(h) \to \min\limits_h$,

$$\frac{\partial}{\partial h}(AMISE(h)) = 4h^3 - \frac{\sigma^2}{2n\sqrt{\pi}} \cdot h^{-2} = 0$$

Отсюда $4h_{opt}^3 = \frac{\sigma^2}{2n\sqrt{\pi}} \cdot h_{opt}^{-2} \Rightarrow$

$$\Rightarrow h_{opt}^5 = \frac{\sigma^2}{8n\sqrt{\pi}} \Rightarrow h_{opt} = \left(\frac{\sigma^2}{8n\sqrt{\pi}}\right)^{1/5}.$$

Итого $h_{opt} = \left(\frac{\sigma^2}{8n\sqrt{\pi}}\right)^{1/5}$

Ответ: $h_{opt} = \left(\frac{\sigma^2}{8n\sqrt{\pi}}\right)^{1/5}$