Упражнение 1

Галиченко Евгений

24 02 2020

## Постновка задачи

**Задача 1.** На данных своего варианта повторить три графика из первой практики, выбрав число степеней свободы как компромисс между точностью (оценкой ошибки на тестовой выборке) и простотой модели (числом степеней свободы). Все рисунки сохранить в графические файлы в формате png.

**Задача 2.** Решить задачу 1, изменив характеристики данных (см. свой вариант). Почему при таком изменении данных MSE меняется именно так? Все рисунки сохранить в графические файлы в формате png.

Вариант 4. Функция для задачи 1:

Характеристики для задачи 2:

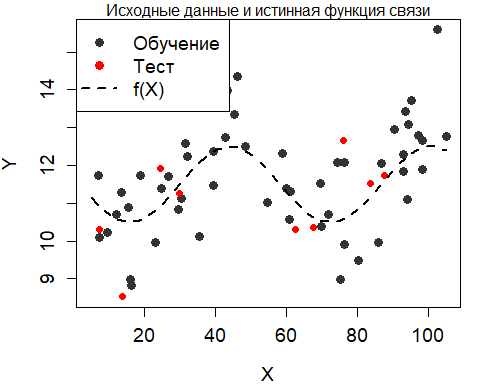
## Задача 1. Построение графиков

Сгенерируем и :  
\*   
\*

# Генерируем данные  
  
my.seed <- 1486372882 # ядро  
n.all <- 60 # наблюдений всего  
train.percent <- 0.85 # доля обучающей выборки  
res.sd <- 1 # стандартное отклонение случайного шума  
x.min <- 5 # границы изменения X: нижняя  
x.max <- 105 # и верхняя  
  
# Фактические значения X  
set.seed(my.seed)  
x <- runif(x.min, x.max, n = n.all)  
  
# Случайный шум  
set.seed(my.seed)  
res <- rnorm(mean = 0, sd = res.sd, n = n.all)  
  
# Отбираем наблюдения в обучающую выборку  
set.seed(my.seed)  
inTrain <- sample(seq\_along(x), size = train.percent\*n.all)  
  
# Истиная функция взаимосвязи  
y.func <- function(x) {8 + 3.5 + sin((x - 30) / 9)}  
  
# Для графика истинной взаимосвязи  
x.line <- seq(x.min, x.max, length = n.all)  
y.line <- y.func(x.line)  
  
# Фактические значения y (с шумом)  
y <- y.func(x) + res  
  
# Создаем векторы с данными для построения графиков  
  
# Наблюдения для обучающей выборки  
x.train <- x[inTrain]  
y.train <- y[inTrain]  
  
# Наблюдения для тестовой выборки  
x.test <- x[-inTrain]  
y.test <- y[-inTrain]

**Изобразим исходные данные на графике**

# График 1:Исходные данные на график  
  
# Убираем широкие поля рисунков  
par(mar = c(4, 4, 1, 1))  
  
# Наименьшие/наибольшие значения по осям  
x.lim <- c(x.min, x.max)  
y.lim <- c(min(y), max(y))  
  
# Наблюдения с шумом (обучающая выборка)  
plot(x.train, y.train,  
 col = grey(0.2), bg = grey(0.2), pch = 21,  
 xlab = 'X', ylab = 'Y',  
 xlim = x.lim, ylim = y.lim,  
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
# Заголовок  
mtext('Исходные данные и истинная функция связи', side = 3)  
# Наблюдения тестовой выборки  
points(x.test, y.test, col = 'red', bg = 'red', pch = 21)  
# Истинная функция  
lines(x.line, y.line, lwd = 2, lty = 2)  
# Легенда  
legend('topleft', legend = c('Обучение', 'Тест', 'f(X)'),  
 pch = c(16, 16, NA),  
 col = c(grey(0.2), 'red', 'black'),  
 lty = c(0, 0, 2), lwd = c(1, 1, 2), cex = 1.2)



В качестве модели используем сплайны со степенями свободы от 2 (прямая) до 40 (количество узлов равно 2/3 наблюдений). Строим модели с различным количеством степеней свободы и в каждом случае считаем среднеквадратическую ошибку модели на обучающей и тестовой выборках.

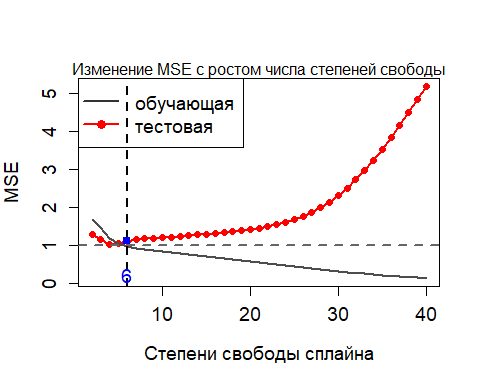
**Модель с df от 2 до 40**

# Теперь строим модели с df от 2 до 40   
  
# Максимальное число степеней свободы для модели сплайна  
max.df <- 40  
  
tbl <- data.frame(df = 2:max.df) # таблица для записи ошибок  
tbl$MSE.train <- 0 # столбец: ошибки на обучающей выборке  
tbl$MSE.test <- 0 # столбец: ошибки на тестовой выборке  
  
# цикл по степеням свободы  
for (i in 2:max.df) {  
 # строим модель  
 mod <- smooth.spline(x = x.train, y = y.train, df = i)  
   
 # модельные значения для расчёта ошибок  
 y.model.train <- predict(mod, data.frame(x = x.train))$y[, 1]  
 y.model.test <- predict(mod, data.frame(x = x.test))$y[, 1]  
   
 # считаем средний квадрат ошибки на обучающей и тестовой выборке  
 MSE <- c(sum((y.train - y.model.train)^2) / length(x.train),  
 sum((y.test - y.model.test)^2) / length(x.test))  
   
 # записываем ошибки в таблицу  
 tbl[tbl$df == i, c('MSE.train', 'MSE.test')] <- MSE  
}  
  
# первые строки таблицы  
head(tbl, 10)

## df MSE.train MSE.test  
## 1 2 1.6983810 1.306921  
## 2 3 1.4883145 1.156282  
## 3 4 1.2012109 1.028525  
## 4 5 1.0488299 1.057342  
## 5 6 0.9770188 1.115524  
## 6 7 0.9374750 1.156874  
## 7 8 0.9081777 1.181937  
## 8 9 0.8809298 1.198601  
## 9 10 0.8539576 1.213256  
## 10 11 0.8274070 1.229086

**Изобразим на графике поведение ошибок при различном количестве степеней свободы**

# График 2: Зависимость MSE от гибкости модели  
  
plot(x = tbl$df, y = tbl$MSE.test,   
 type = 'l', col = 'red', lwd = 2,  
 xlab = 'Степени свободы сплайна', ylab = 'MSE',  
 ylim = c(min(tbl$MSE.train, tbl$MSE.test),   
 max(tbl$MSE.train, tbl$MSE.test)),  
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
  
# заголовок  
mtext('Изменение MSE с ростом числа степеней свободы', side = 3)  
  
points(x = tbl$df, y = tbl$MSE.test,  
 pch = 21, col = 'red', bg = 'red')  
lines(x = tbl$df, y = tbl$MSE.train, col = grey(0.3), lwd = 2)  
# неустранимая ошибка  
abline(h = res.sd, lty = 2, col = grey(0.4), lwd = 2)  
  
# легенда  
legend('topleft', legend = c('обучающая', 'тестовая'),  
 pch = c(NA, 16),   
 col = c(grey(0.2), 'red'),   
 lty = c(1, 1), lwd = c(2, 2), cex = 1.2)  
  
# степени свободы у наименьшей ошибки на тестовой выборке  
min.MSE.test <- min(tbl$MSE.test)  
df.min.MSE.test <- tbl[tbl$MSE.test == min.MSE.test, 'df']  
  
# компромисс между точностью и простотой модели по графику  
df.my.MSE.test <- 6  
my.MSE.test <- tbl[tbl$df == df.my.MSE.test, 'MSE.test']  
  
# ставим точку на графике  
abline(v = df.my.MSE.test,   
 lty = 2, lwd = 2)  
points(x = df.my.MSE.test, y = my.MSE.test,   
 pch = 15, col = 'blue')  
mtext(df.my.MSE.test,   
 side = 1, line = -1, at = df.my.MSE.test, col = 'blue', cex = 1.2)



На этом графике:

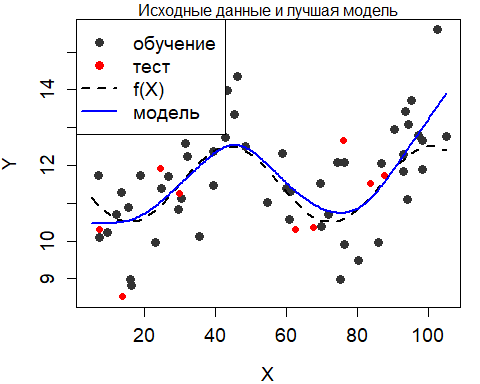
При движении слева направо MSE на обучающей выборке (серая кривая) сокращается, потому что с ростом числа степеней свободы растет число узлов, по которым строится сплайн. При этом модельная кривая подгоняется по всему возрастающему кол-ву точек и становится всё более гибкой. В результате индивидуальные расстояния от фактических наблюдений за Y до их иодельных оценок сокращается, что приводит к сокращению MSE.

При движении слева на право MSE на тестовой выборке (красная кривая) сначала резко сокращается, затем растет. Нам известна истиная форма связи Y с X, она описывается тригонометрической функцией. Резкое падение ошибки на тестовой выборке при небольшом числе степеней свободы связано с тем, что модель приближается по гибкости к истинной функции связи. Затем MSE на тестовой выбоке довольно долго остается стабильной, а затем начинает расти. Этот рост объясняется эффектом переобучения модели: она все лучше описывает обучающую выборку, и при этом постепенно становится неприменимой ни к одному другому набору наблюдений.

Наименьшее значение MSE на тестовой выборке соответствует числу степеней свободы 4 и равно 1.028525. Визуально по графику мы можем установить, что первое значение , близкое к стабильно низким, соответствует df = 3. Ошибка здесь равна 1.156282, что ненамного отличается от минимума. Именно df = 3 было выбрано в качестве компромисса между точностью (минимальной MSE на тестовой выборке) и простотой модели (чем меньше степеней свободы, тем модель проще).

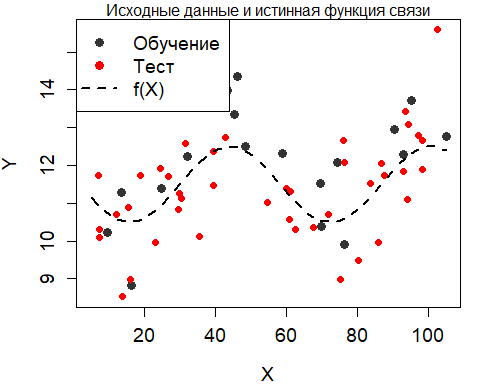
**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**

# График 3: Лучшая модель (компромисс между гибкостью и точностью)  
  
mod.MSE.test <- smooth.spline(x = x.train, y = y.train, df = df.my.MSE.test)  
  
# для гладких графиков модели  
x.model.plot <- seq(x.min, x.max, length = 250)  
y.model.plot <- predict(mod.MSE.test, data.frame(x = x.model.plot))$y[, 1]  
  
# убираем широкие поля рисунка  
par(mar = c(4, 4, 1, 1))  
  
# наименьшие/наибольшие значения по осям  
x.lim <- c(x.min, x.max)  
y.lim <- c(min(y), max(y))  
  
# наблюдения с шумом (обучающая выборка)  
plot(x.train, y.train,   
 col = grey(0.2), bg = grey(0.2), pch = 21,  
 xlab = 'X', ylab = 'Y',   
 xlim = x.lim, ylim = y.lim,   
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
  
# заголовок  
mtext('Исходные данные и лучшая модель', side = 3)  
  
# наблюдения тестовой выборки  
points(x.test, y.test,   
 col = 'red', bg = 'red', pch = 21)  
  
# истинная функция  
lines(x.line, y.line,   
 lwd = 2, lty = 2)  
  
# модель  
lines(x.model.plot, y.model.plot,   
 lwd = 2, col = 'blue')  
  
# легенда  
legend('topleft', legend = c('обучение', 'тест', 'f(X)', 'модель'),  
 pch = c(16, 16, NA, NA),   
 col = c(grey(0.2), 'red', 'black', 'blue'),   
 lty = c(0, 0, 2, 1), lwd = c(1, 1, 2, 2), cex = 1.2)



## Задача 2

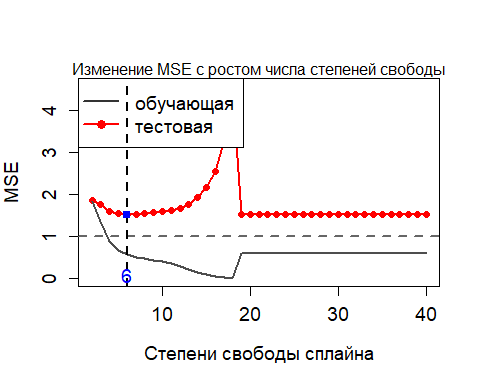
**Изобразим исходные данные на графике**



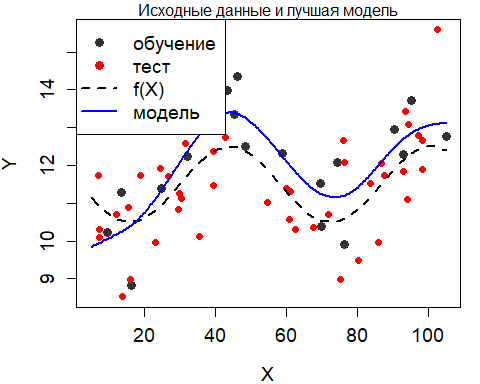
**Модель с df от 2 до 40**

## df MSE.train MSE.test  
## 1 2 1.8352906 1.848769  
## 2 3 1.3527642 1.750708  
## 3 4 0.8866378 1.602521  
## 4 5 0.6715537 1.543047  
## 5 6 0.5656323 1.513967  
## 6 7 0.5079905 1.514711  
## 7 8 0.4695698 1.534719  
## 8 9 0.4346019 1.561972  
## 9 10 0.3948413 1.590979  
## 10 11 0.3447321 1.623891

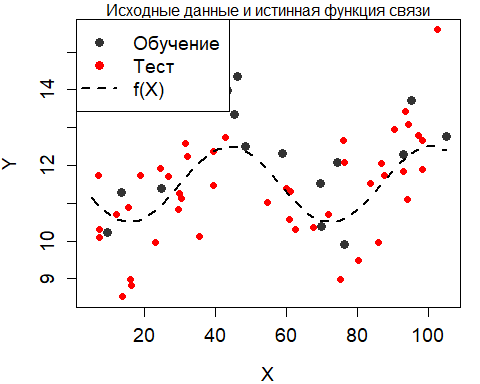
**Изобразим на графике поведение ошибок при различном количестве степеней свободы**



**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**



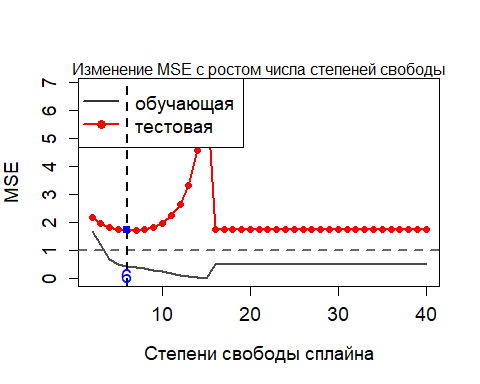
**Изобразим исходные данные на графике**



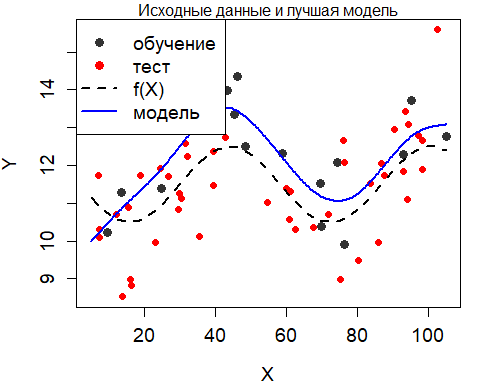
**Модель с df от 2 до 40**

## df MSE.train MSE.test  
## 1 2 1.6867503 2.167319  
## 2 3 1.2060548 1.962062  
## 3 4 0.6821843 1.800713  
## 4 5 0.5000714 1.760574  
## 5 6 0.4271848 1.722995  
## 6 7 0.3815988 1.714722  
## 7 8 0.3379422 1.748579  
## 8 9 0.2879253 1.828043  
## 9 10 0.2304329 1.970402  
## 10 11 0.1702142 2.224993

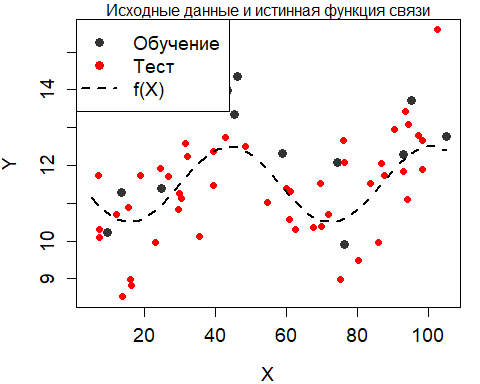
**Изобразим на графике поведение ошибок при различном количестве степеней свободы**



**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**



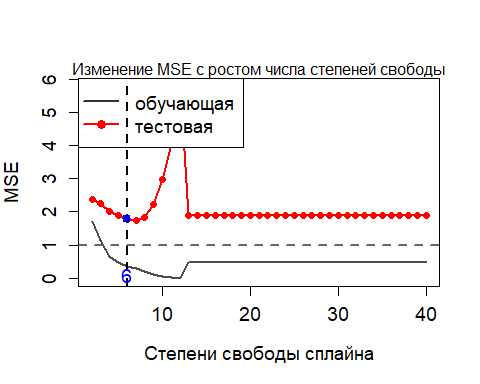
**Изобразим исходные данные на графике**



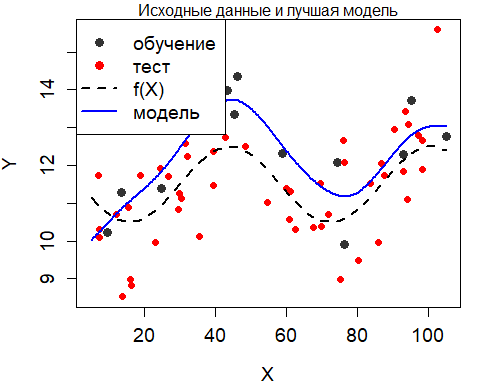
**Модель с df от 2 до 40**

## df MSE.train MSE.test  
## 1 2 1.71660937 2.363464  
## 2 3 1.15181008 2.261464  
## 3 4 0.67150796 2.022497  
## 4 5 0.47028411 1.880270  
## 5 6 0.37207199 1.782779  
## 6 7 0.30388330 1.748696  
## 7 8 0.22166551 1.843403  
## 8 9 0.13239548 2.210211  
## 9 10 0.06544407 2.968534  
## 10 11 0.02753881 4.035293

**Изобразим на графике поведение ошибок при различном количестве степеней свободы**



**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**



Оптимальной моделью по-прежнему можно считать модель с 3 степенями свободы.

|  |  |  |
| --- | --- | --- |
| train.percent |  |  |
| 0.3 | 1.3527642 | 1.750708 |
| 0.25 | 1.2060548 | 1.962062 |
| 0.2 | 1.15181008 | 2.261464 |

При уменьшении объема обучаемого множества среднеквадратическая ошибка на обучаемом множестве уменьшается, а на тестовом увеличивается. Это может говорить, что при увеличении объема выборки существует эффект переобучения модели.