Московский государственный технический университет
имени Н. Э. Баумана

Факультет: Информатика и системы управления

Кафедра: Программное обеспечение ЭВМ и информационные технологии

Дисциплина: Математическая статистика

# Лабораторная работа №3
# Метод наименьших квадратов

Выполнила: Щербатюк Дарья
Группа: ИУ7-64
Вариант: 12

Москва, 2018 г.

# 1 Постановка задачи

**Цель работы:** аппроксимация неизвестной зависимости параболой.

**Содержание работы:**

1. Для выборки $(y_i, t_i)$, $i = \overline{1; n}$, реализовать в виде программы на ЭВМ:
   (а) вычисление МНК-оценки $\vec{\theta} = (\theta_0, \theta_1, \theta_2)$ параметров модели $y = \theta_0 + \theta_1 t + \theta_2 t^2$;
   (б) вычисление среднеквадратичного отклонения $\Delta = \sqrt{\sum_{i=1}^{n} \left(y_i - y(t_i)\right)^2}$ полученной модели от результатов наблюдений;
   (в) построение на одном графике системы точек $(y_i, t_i)$, $i = \overline{1; n}$, и графика функции $y = y(t)$, $t \in \left[t_{(1)}; t_{(n)}\right]$ (для полученной оценки $\vec{\theta}$).
2. провести необходимые вычисления и построить соответствующие графики для выборки из индивидуального варианта.

**Содержание отчёта:**

1. постановка задачи аппроксимации неизвестной зависимости по результатам наблюдений;
2. понятие МНК-оценки параметров линейной модели;
3. формулы для вычисления МНК-оценки в рассматриваемом случае;
4. текст программы;
5. результаты расчетов и графики для выборки из индивидуального варианта.

# 2 Теоретическая часть

## 2.1 Аппроксимация неизвестной зависимости

Предположим, что в нашем распоряжении имеются результаты $n$ наблюдений:

$$\begin{cases} y_1 = \Phi(x_1) + \varepsilon_1 \\ \qquad \cdots \\ y_n = \Phi(x_n) + \varepsilon_n \end{cases}, \quad \text{где} \tag{1}$$

- $y_1, \ldots, y_n$ — $n$ реализаций $Y$;
- $\varepsilon_1, \ldots, \varepsilon_n$ — $n$ реализаций $\varepsilon$;
- $x_1, \ldots, x_n$ – известные значения.

Требуется на основе этих данных подобрать функцию $\widehat{\Phi}$ так, чтобы она наилучшим образом аппроксимировала неизвестную функцию $\Phi$.

## 2.2 Понятие МНК-оценки параметров линейной модели

Часто в качестве функции $\widehat{\Phi}(x)$ выбирают функцию следующего вида:

$$\widehat{\Phi}(x) = \theta_1\psi_1(x) + \cdots + \theta_p\psi_p(x), \quad \text{где} \tag{2}$$

- $\psi_1, \ldots, \psi_p$ — известные функции.

Параметры $\theta_1, \ldots, \theta_p$ подбирают так, чтобы $\widehat{\Phi}(x)$ наилучшим образом аппроксимировала $\Phi(x)$.

С учётом предположения о виде функции $\widehat{\Phi}$ результаты наблюдений можно записать в виде:

$$y_i = \theta_1\psi_1(x_i) + \cdots + \theta_p\psi_p(x_i) + \varepsilon_i, \quad i = \overline{1;n}\,. \tag{3}$$

В матричном виде:

$$\vec{y} = \Psi\vec{\theta} + \vec{\varepsilon}, \quad \text{где} \tag{4}$$

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad \Psi = \begin{pmatrix} \psi_1(x_1) & \psi_2(x_1) & \cdots & \psi_p(x_1) \\ \psi_1(x_2) & \psi_2(x_2) & \cdots & \psi_p(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_1(x_n) & \psi_2(x_n) & \cdots & \psi_p(x_n) \end{pmatrix}, \quad \vec{\theta} = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_p \end{pmatrix}, \quad \vec{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Задача заключается в подборе $\vec{\theta}$.

Будем предполагать, что:

1. $M\varepsilon = 0$, т. е. систематические ошибки отсутствуют;
2. $\varepsilon \sim N(0, \sigma^2)$.

**Определение 2.1.** Оценка $\hat{\vec{\theta}}$ вектора $\vec{\theta}$ называется *оценкой*, полученной по *методу наименьших квадратов (МНК-оценкой)*, если $\vec{\theta}$ доставляет минимальное значение функции $S(\vec{\theta}) = \|y - \Psi\vec{\theta}\|^2$.

### 2.3 Формулы для вычисления МНК-оценки в рассматриваемом случае

В рассматриваемом случае МНК-оценка вектора $\vec{\theta}$ имеет вид:

$$\hat{\vec{\theta}} = (\Psi^T \Psi)^{-1} \cdot \Psi^T \vec{y}, \quad \text{где} \tag{5}$$

- $\mathsf{rg}(\Psi) = p$ — числу столбцов.

причём так как $y = \theta_0 + \theta_1 t + \theta_2 t^2$, то

$$\Psi = \begin{pmatrix} 1 & t_1 & t_1^2 \\ 1 & t_2 & t_2^2 \\ \vdots & \vdots & \vdots \\ 1 & t_n & t_n^2 \end{pmatrix}. \tag{6}$$

### 2.4 Среднеквадратичное отклонение

Среднеквадратичное отклонение полученной модели от результатов наблюдений будем вычислять как

$$\Delta = \sqrt{\sum_{i=1}^{n} (y_i - y(t_i))^2}, \quad \text{где} \tag{7}$$

## 3 Текст программы

```
function lw3()
    t0Array = csvread('t.csv');
    y0Array = csvread('y.csv');

    y1Array = OLS(y0Array, t0Array);

    delta = calculateDelta(y0Array, y1Array);
    fprintf('Delta = %3.2f', delta);

    plot(t0Array, y0Array, '.b', t0Array, y1Array, 'r');
    legend({
        'Original sample';
        'Output model';
    });
end

function Y = OLS(yArray, tArray)
    psiMatrix = makePsiMatrix(tArray);

    thetaArray = (psiMatrix' * psiMatrix) \ (psiMatrix' * yArray');
    disp(thetaArray);

    n = length(yArray);
```

```
    Y = zeros(1, n);
    for i = 1:n
        Y(i) = thetaArray(1) + ...
               thetaArray(2)*tArray(i) + ...
               thetaArray(3)*tArray(i)*tArray(i);
    end
end

function Psi = makePsiMatrix(tArray)
    n   = length(tArray);
    p   = 3;
    Psi = zeros(n, p);
    for i = 1:n
        Psi(i, 1) = 1;
        Psi(i, 2) = tArray(i);
        Psi(i, 3) = tArray(i)*tArray(i);
    end
end

function X = calculateDelta(yArray, yNewArray)
    n   = length(yArray);
    sum = 0;
    for i = 1:n
        sum = sum + power((yArray(i) - yNewArray(i)), 2);
    end
    X = sqrt(sum);
end
```

# 4 Результаты расчётов

*МНК-оценка* в данном варианте имеет вид

$$\hat{\vec{\theta}} = \begin{pmatrix} 4.8027 \\ 2.3381 \\ 1.8644 \end{pmatrix}. \tag{8}$$

В свою очередь *среднеквадратичного отклонение* полученной модели от результатов наблюдений равна

$$\Delta = 125.44\,. \tag{9}$$

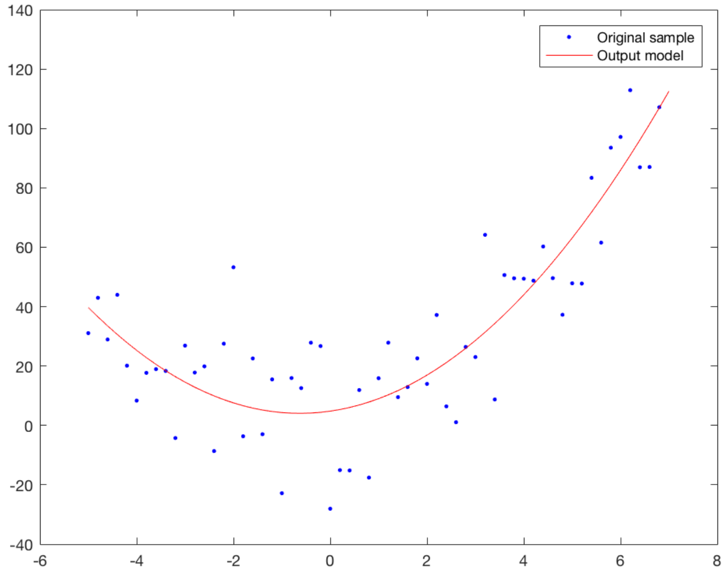


Рис. 1: График исходной выборки и полученной модели