УНИВЕРСИТЕТ
ИННОПОЛИС

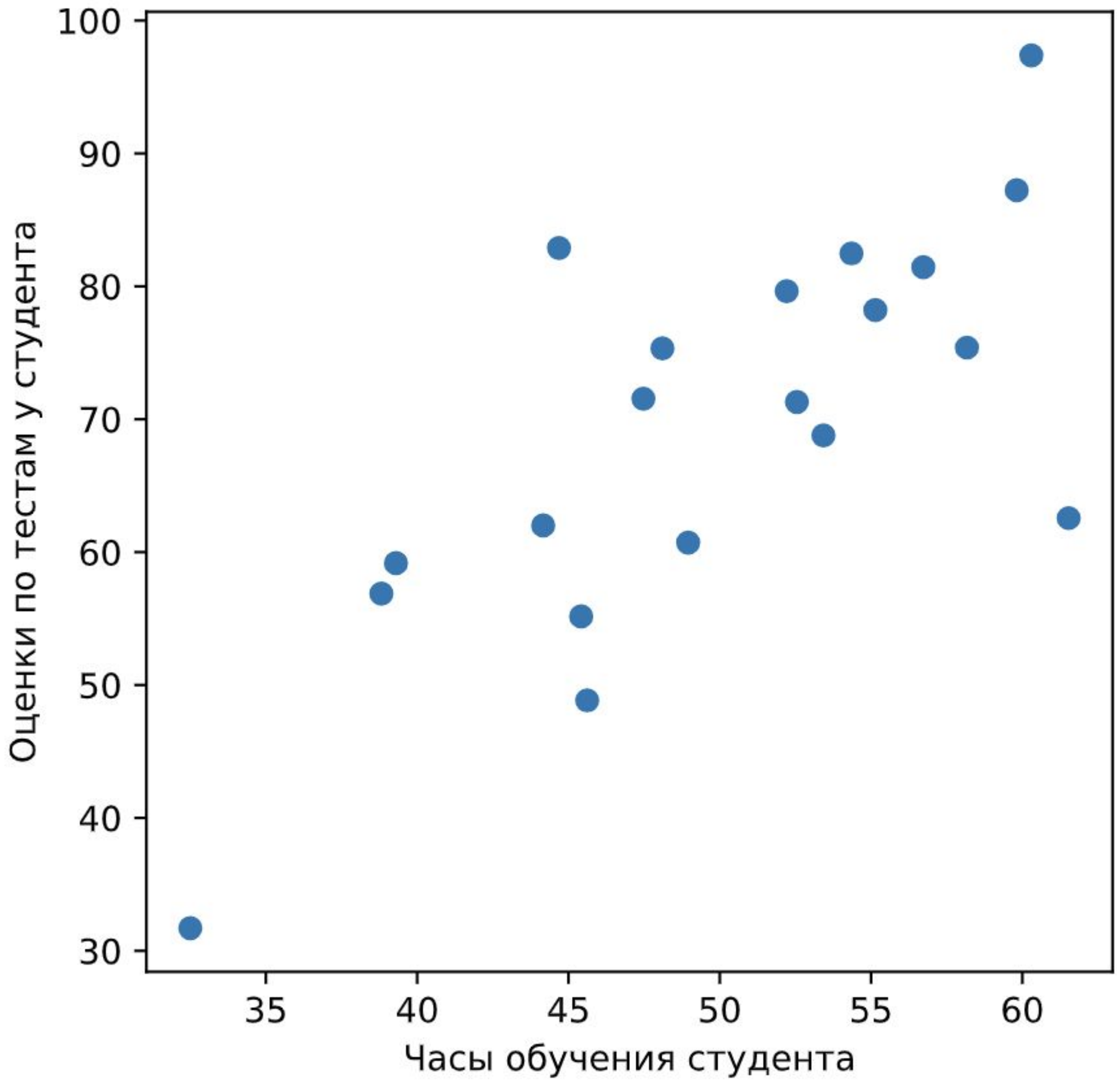
Линейная регрессия

Часть 2

Воробьёва Мария

- maria.vorobyova.ser@gmail.com
- @SparrowMaria

Парная линейная регрессия



У нас есть данные, мы построили график и увидели следующее…

Что делать?

Кажется, что в данных прослеживается закономерность…

И она похоже на прямую…

мы знаем, что уравнение прямой на плоскости
y = a + b * x

Парная линейная регрессия

Уравнение прямой - это и есть линейная регрессия, в данном случае парная линейная регрессия

$$Y = a + b \cdot x$$

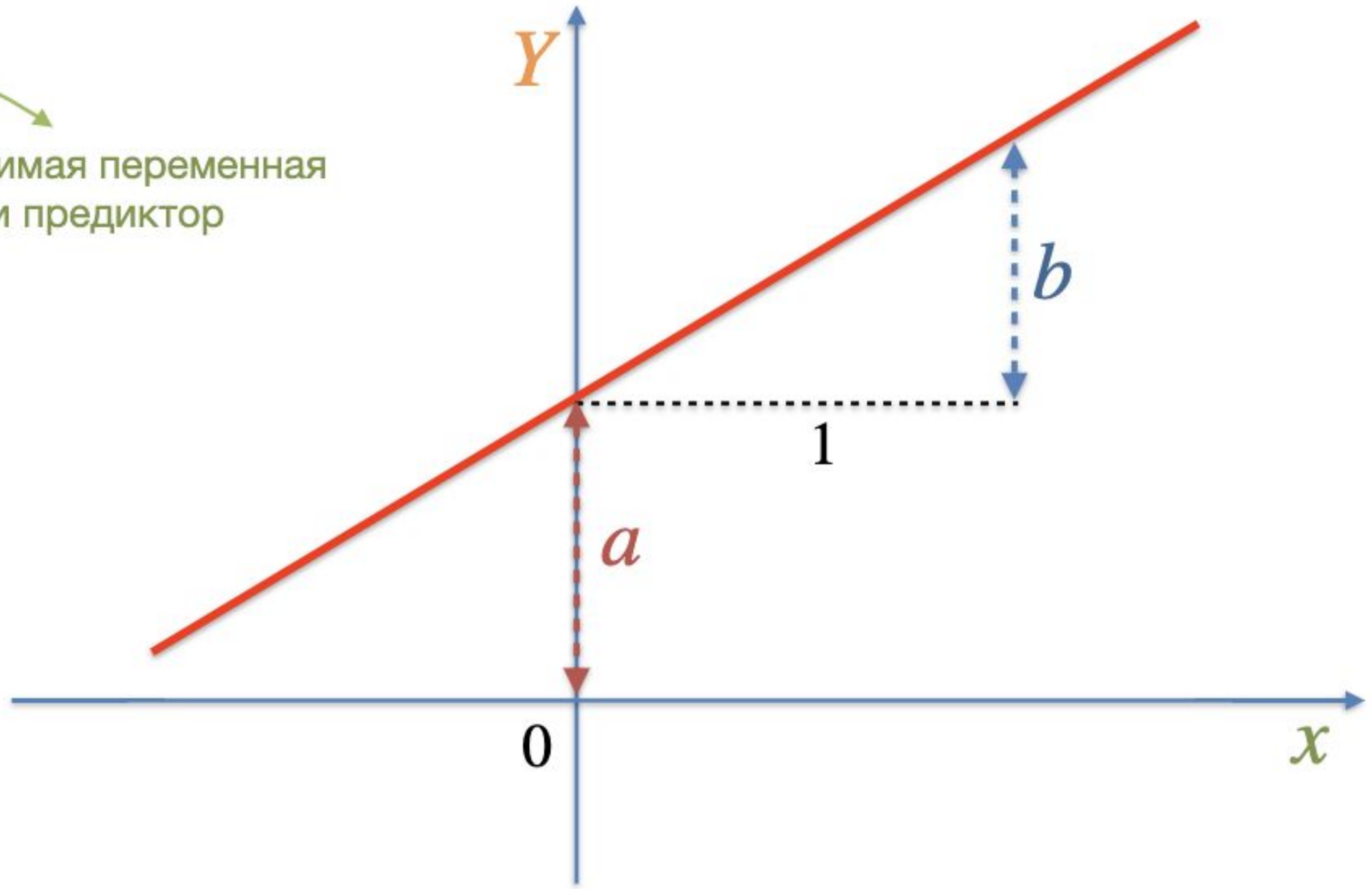
зависимая переменная или переменная отклика

свободный член линии оценки

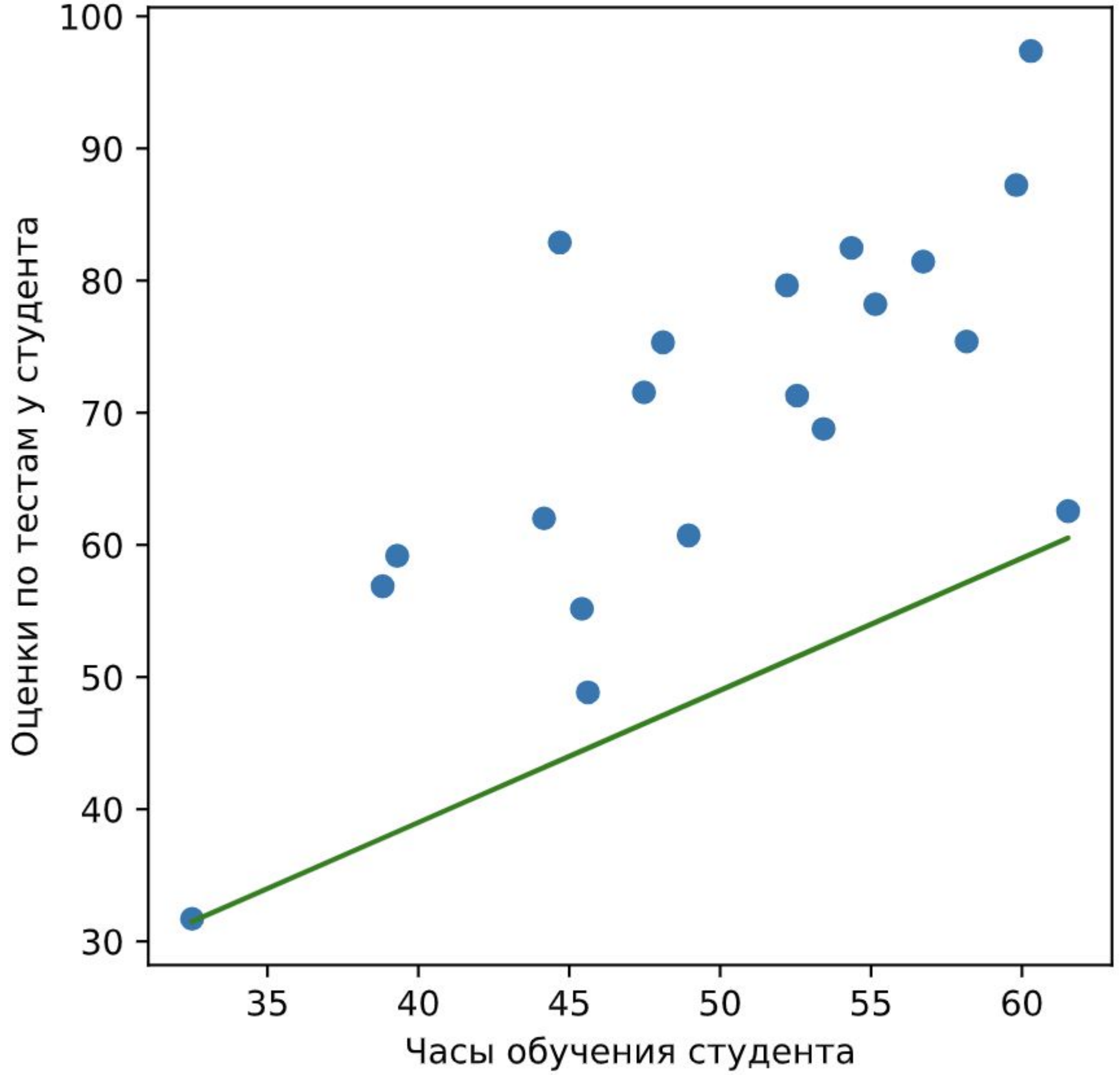
угловой коэффициент

независимая переменная или предиктор

коэффициенты регрессии оценённой линии



Парная линейная регрессия

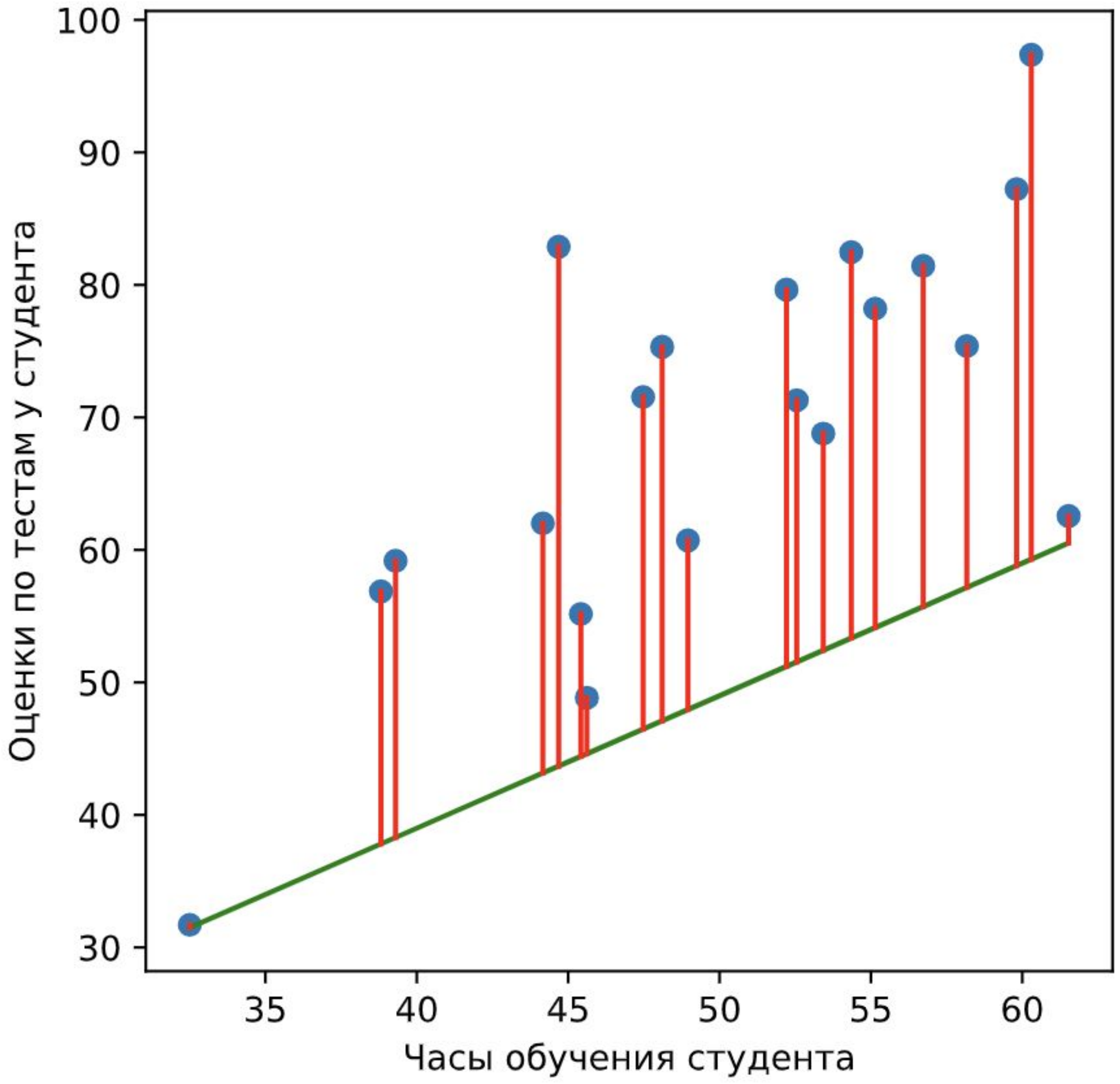


Отлично, построили прямую…

Но кажется, что-то не совсем то, что нам хотелось бы

А что нам не нравится?

Парная линейная регрессия



Попробуем оценить, на сколько наша прямая “не попадает” в наши точки

Посчитаем разности между фактическими данными и точками на прямой,

то есть посчитаем абсолютные суммы длин красных отрезков, получим 20.47

Кажется, можно лучше

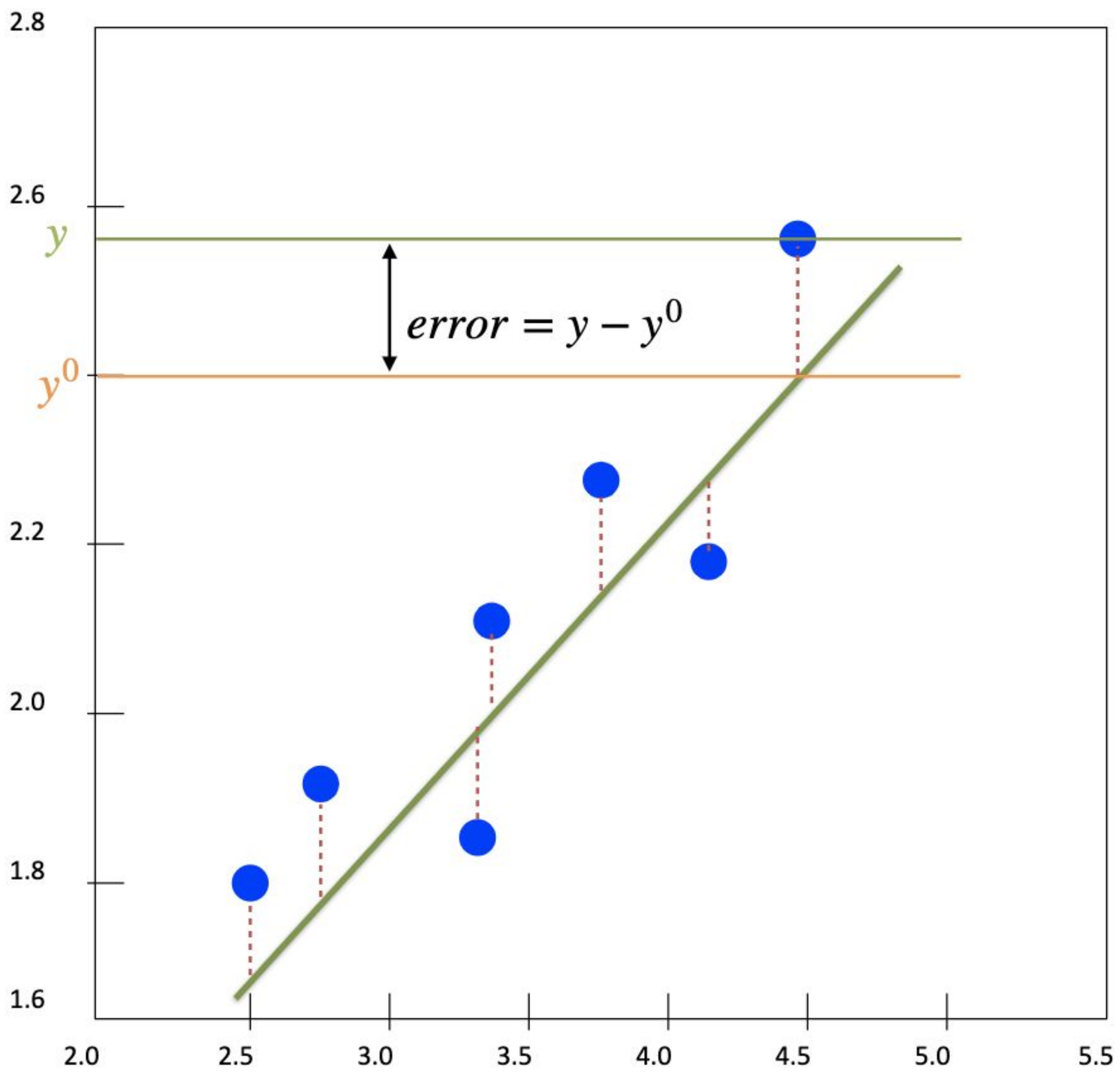
Парная линейная регрессия

Мы должны построить такую прямую, для которой отклонения от фактических точек будут минимальны

$$\sum_{i=1}^{N} (y_i - y_i^0)^2 =$$

$$\sum_{i=1}^{N} (y_i - f(x_i))^2 =$$

$$\sum_{i=1}^{N} (y_i - a - bx_i)^2$$



Парная линейная регрессия

Необходимо минимизировать сумму квадратов отклонений RSS (Residiual Sum of Squares)

$$\text{RSS} = \sum_{i=1}^{N} (y_i - a - bx_i)^2$$

Для того, чтобы определить прямую, необходимо найти a и b

нам поможет Метод Наименьших Квадратов МНК

Парная линейная регрессия.

INNOPOLIS UNIVERSITY

Метод Наименьших Квадратов (МНК). Аналитическое решение

$$\sum_{i=1}^{N}(y_i - a - bx_i)^2$$ **—> MIN**

$$b = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sum_{i=1}^{n}(x_i-\bar{x})^2}$$

$$a = \bar{y} - b\bar{x}$$

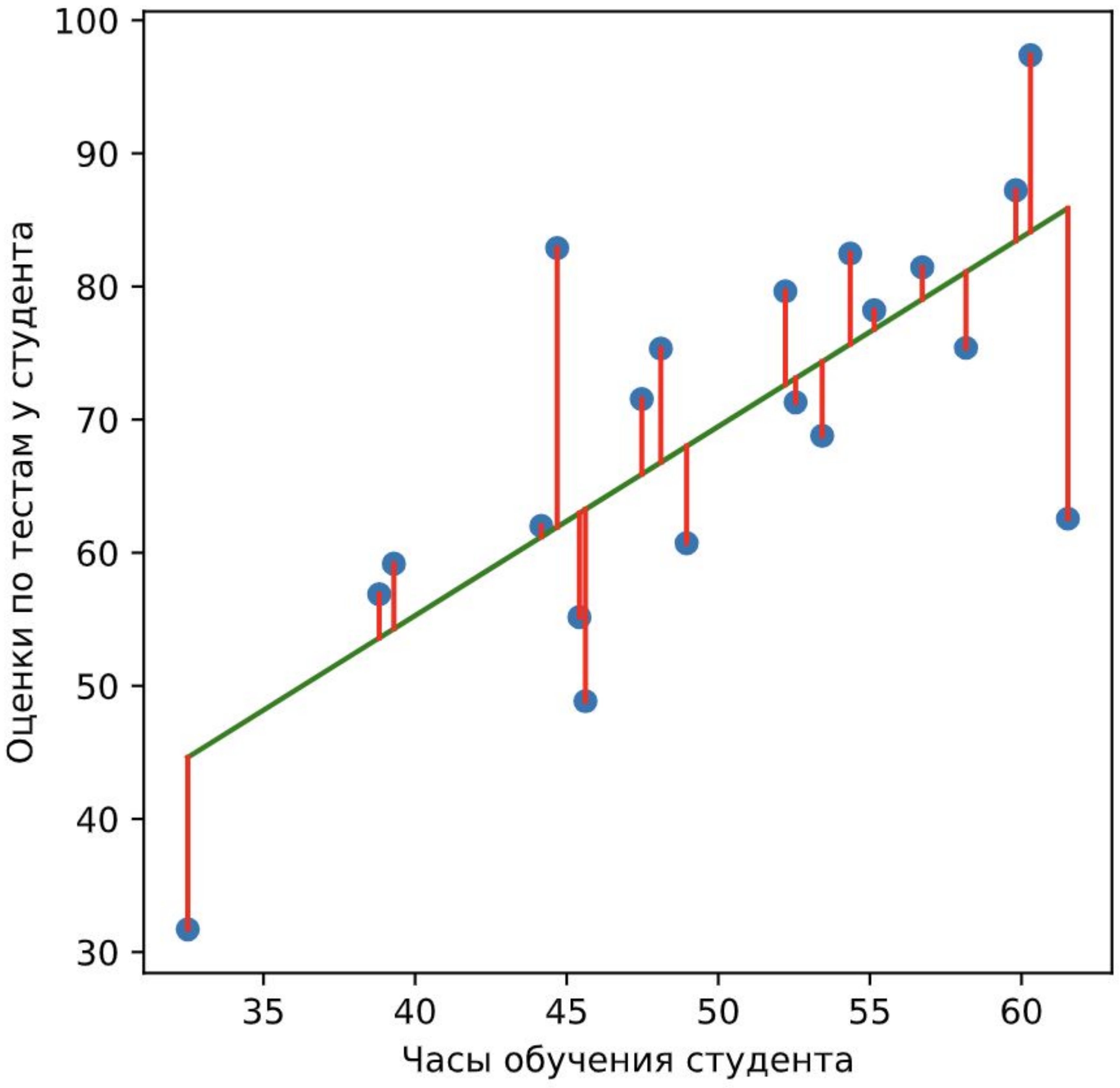
где: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Для минимизации суммы квадратов ошибок S мы берем *частные производные* по a и b и приравниваем их к нулю:

$$\frac{\partial S}{\partial b} = -2\sum_{i=1}^{n}(y_i - a - b\cdot x_i) = 0$$

$$\frac{\partial S}{\partial a} = -2\sum_{i=1}^{n} x_i\cdot(y_i - a - b\cdot x_i) = 0$$

Парная линейная регрессия



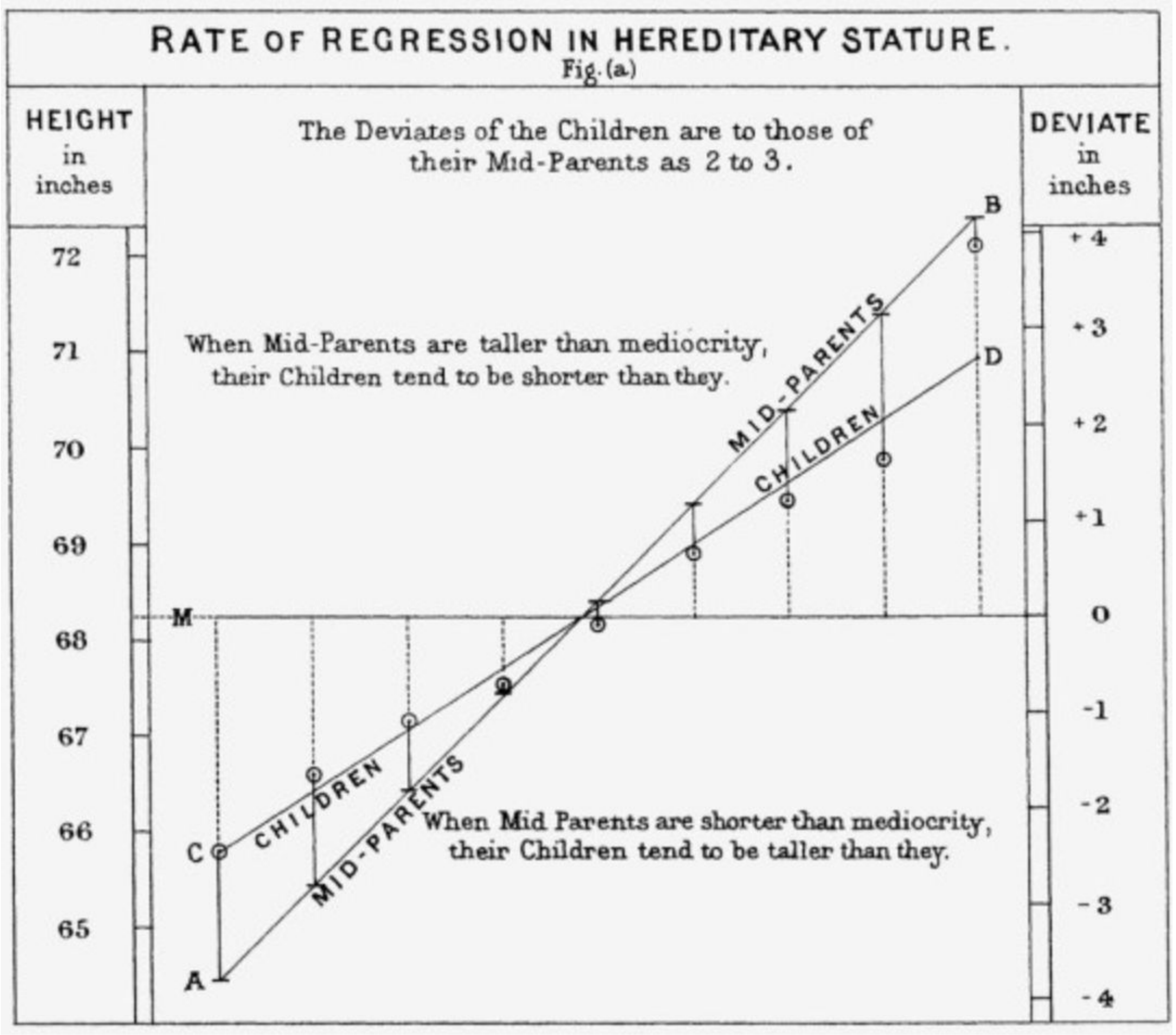
Построим линейную регрессию

сумма всех красных отрезков по модулю равна 7.88,

то есть

- MAE = 7.88,
- MSE = 98.58,
- RMSE = 9.93

Линейная регрессия. История



Почему же регрессия?

В 1886 году Понятие регрессии ввел сэр Френсис Гальтон, английский исследователь широкого профиля.

http://log-lessons.ru/chto-takoe-regressiya-k-srednemu/

Линейная регрессия. Общий случай

пусть дано d переменных(столбцов) x_i, составим линейную комбинацию:

$$w_1*x_1 + w_2*x_2 + \dots w_n*x_n$$

Линейная регрессия: $a(x) = w_0 + w_1*x_1 + w_2*x_2 + \dots w_n*x_n$

В компактном виде $$a(x) = w_0 + \sum_{j=1}^{d} w_j x_j.$$

w_i - веса/коэффициенты
w_0 - свободным коэффициентом/сдвиг

можно еще в более компактном виде $$a(x) = w_0 + \langle w, x \rangle,$$

а если предположить, что существует еще один столбец со всеми 1, то можно записать еще компактнее $$a(x) = \langle w, x \rangle.$$

http://wiki.cs.hse.ru/Машинное_обучение_1

Обучение линейной регрессии

Минимизируем среднеквадратическую ошибку $\frac{1}{\ell}\sum_{i=1}^{\ell}\left(\langle w, x_i\rangle - y_i\right)^2 \to \min_w$

Если продифференцировать функционал и приравнять к 0, то получим решение

$$w = (X^T X)^{-1} X^T y.$$

Но на практике так никто не делает, на практике используют численные методы, в частности используют метод градиентного спуска (и его модификации)

Линейная регрессия. Теорема Гаусса-Маркова

Если данные обладают следующими свойствами:

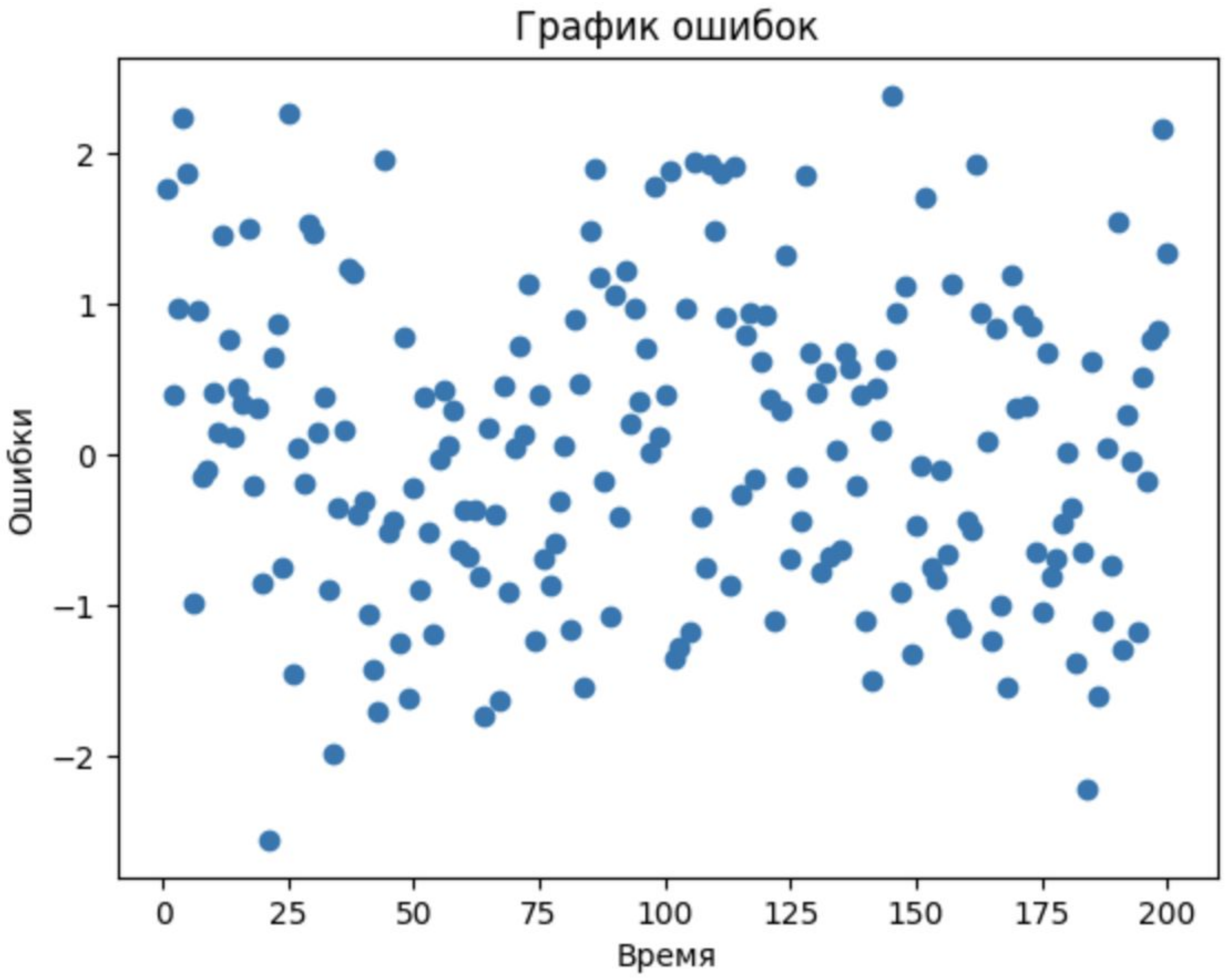
1. Модель данных правильно специфицирована;
2. Все X_i детерминированы и не все равны между собой;
3. Ошибки не носят систематического характера, то есть $\mathbb{E}(\varepsilon_i \mid X_i) = 0 \; \forall i$;
4. Дисперсия ошибок одинакова и равна некоторой σ^2;
5. Ошибки некоррелированы, то есть $\mathrm{Cov}(\varepsilon_i, \varepsilon_j) = 0 \; \forall i, j$;

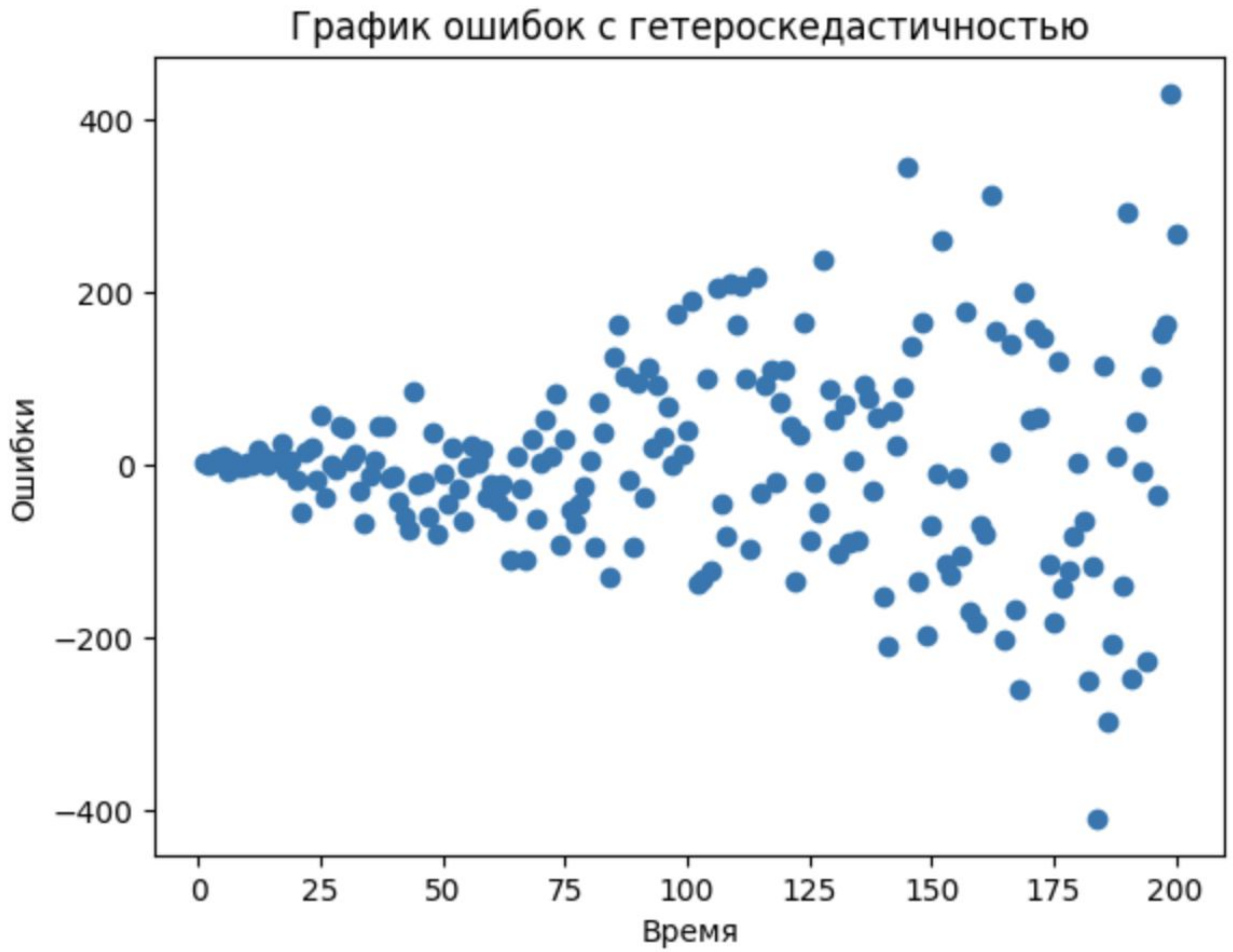
в этих условиях оценки метода наименьших квадратов оптимальны в классе линейных несмещённых оценок,

Проще говоря: метод наименьших квадратов даёт самые точные и справедливые оценки

Линейная регрессия. Теорема Гаусса-Маркова

Независимость ошибок: Ошибки должны быть независимыми и одинаково распределенными и иметь постоянную дисперсию (гомоскедастичность).

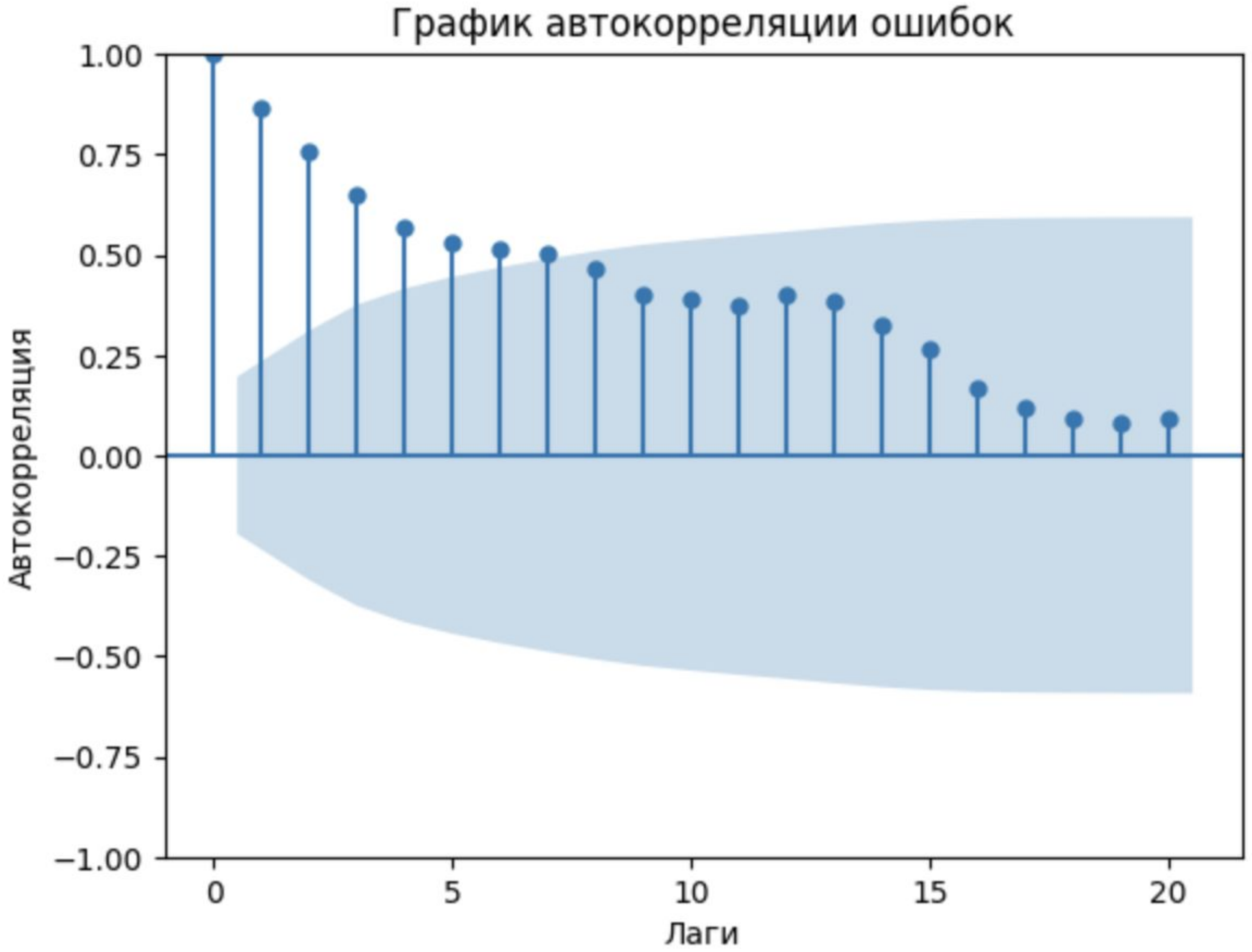


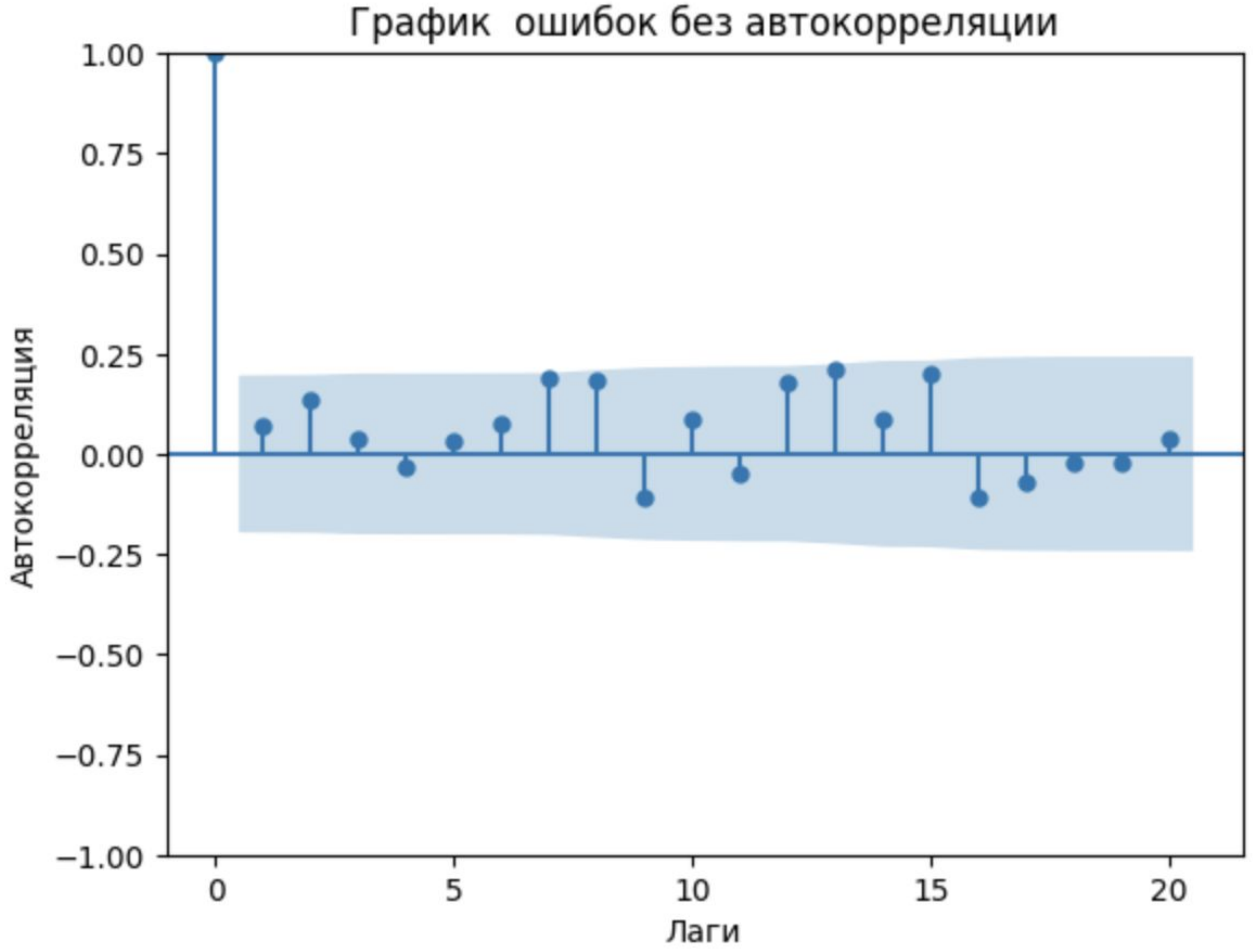


Линейная регрессия. Теорема Гаусса-Маркова

Отсутствие автокорреляции ошибок: Ошибки должны быть некоррелированными между собой.

Автокорреляция считается так: $\rho(\tau) = \mathrm{corr}(x(t), x(t+\tau)) = \frac{\mathrm{cov}(x(t), x(t+\tau))}{\sqrt{D(x(t))}\sqrt{D(x(t+\tau))}}$





Линейная регрессия. Сложности аналитического решения.

Обратная матрица $(X^T X)^{-1}$ существует не всегда. Если матрица $(X^T X)$ является вырожденной или почти вырожденной (плохо обусловленной), **обратная матрица не может быть вычислена**, что делает **аналитическое решение невозможным**.

<u>Причины вырожденности матрицы:</u>

1) Линейная зависимость признаков (мультиколлинеарность)

Если один или несколько столбцов матрицы *X* могут быть выражены как линейные комбинации других столбцов, то матрица $(X^T X)$ будет вырожденной

2) Недостаток наблюдений

   Если число наблюдений (строк в матрице X) меньше числа признаков (столбцов в матрице X), то матрица $(X^T X)$ будет вырожденной

Линейная регрессия. Сложности аналитического решения.

Проверка:

1) Определитель матрицы $(X^T X)$ равен нулю
2) Ранг матрицы $(X^T X)$ меньше числа признаков

*Ранг матрицы — это максимальное число линейно независимых строк (или столбцов) матрицы.

Способы решения проблемы вырожденности:

1) Удаление коррелированных признаков
2) Регуляризация
3) Увеличение количества наблюдений

Линейная регрессия. Сложности аналитического решения.

Сложности аналитического решения линейной регрессии:

1) Вырожденность матрицы $(X^T X)$
2) С увеличением числа признаков (независимых переменных) размер матрицы X и сложность вычислений возрастают экспоненциально. Это приводит к увеличению вычислительной нагрузки и потребности в памяти
3) Когда независимые переменные коррелированы друг с другом, матрица $(X^T X)$ становится плохо обусловленной. Это приводит к нестабильности коэффициентов и высоким стандартным ошибкам.

*Плохо обусловленная матрица в численном анализе — это матрица, для которой небольшие изменения входных данных (например, элементов матрицы или правой части системы уравнений) могут приводить к большим изменениям в результатах решения системы уравнений. К ним относят почти вырожденные матрицы и матрицы с элементами, сильно различающимися по масштабу

Альтернатива аналитическому решению. Численный метод - Градиентный спуск

1) У нас есть данные x_i, y_i
2) Мы сделали сильное предложение, что зависимость y от [$x_1, x_2, \dots x_n$] - линейная
3) Далее будем смотреть на сколько хорошо мы приближаем гиперплоскостью данные. Нам нужна функция, которая будет показывать на сколько гиперплоскость "ошибается", то есть нам нужна функция потерь (loss function)
4) В итоге перед нами стоит проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} L(w, x_i, y_i) \to \min_{w}$$

Градиентный спуск

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} L(w, x_i, y_i) \to \min_{w}$$

Градиент указывает направление максимального роста

$$\nabla L(w) = \left(\frac{\partial L(w)}{\partial w_0}, \frac{\partial L(w)}{\partial w_2}, \dots, \frac{\partial L(w)}{\partial w_k} \right)$$

Градиентный спуск

!Напоминание

Производная функции показывает скорость изменения функции

$$f'(x) = \lim_{\Delta x \to 0} \frac{\Delta f(x)}{\Delta x}$$

где $\Delta f(x)$ — изменение функции $f(x)$ при изменении аргумента на Δx.

Градиентный спуск

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} L(w, x_i, y_i) \to \min_{w}$$

Градиент указывает направление максимального роста

$$\nabla L(w) = \left(\frac{\partial L(w)}{\partial w_0}, \frac{\partial L(w)}{\partial w_2}, \dots, \frac{\partial L(w)}{\partial w_k} \right)$$

Идём в противоположную сторону:

$$w_t = w_{t-1} - \underbrace{\eta}_{\text{скорость обучения}} \cdot \nabla L(w_{t-1})$$

Градиентный спуск

Проблема оптимизации:

$$L(w) = \sum_{i=1}^{n} L(w, x_i, y_i) \to \min_{w}$$

Инициализация w_0

$g_t = \frac{1}{n} \sum_{i=1}^{n} \nabla L(w, x_i, y_i)$

$w_t = w_{t-1} - \eta_t \cdot g_t$

if $||w_t - w_{t-1}|| < \varepsilon$:

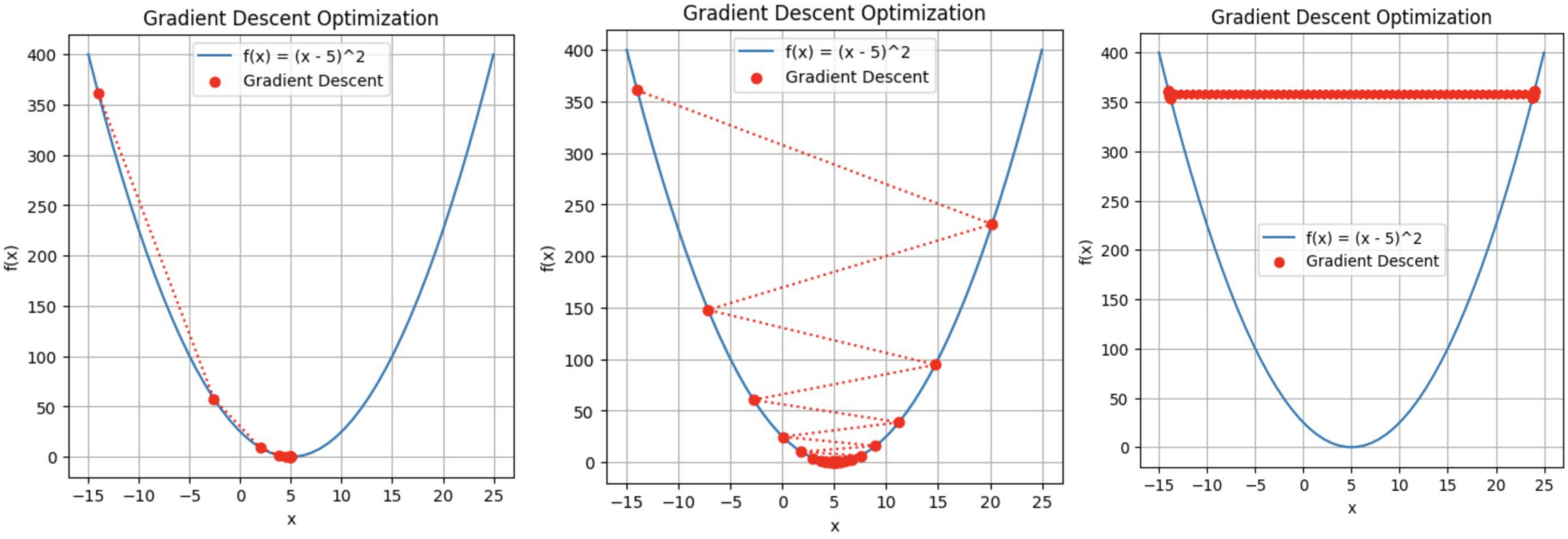
break

Градиентный спуск

INNOPOLIS UNIVERSITY

$$w_t = w_{t-1} - \eta \cdot \nabla L(w_{t-1})$$

скорость обучения



https://colab.research.google.com/drive/1omByd-RQ5tOJrKNV5JJr0t74AhUg5Zof?usp=sharing

Градиентный спуск

- Останавливаем процесс, если

$$||w_t - w_{t-1}|| < \varepsilon$$

- Другой вариант:

$$||\nabla L(w_t)|| < \varepsilon$$

- Обычно в глубоком обучении: останавливаемся, когда ошибка на валидационной выборке перестаёт убывать

Градиентный спуск

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w)^2 \to \min_{w}$$

Градиент:

$$\nabla L(w) = -2 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w) \cdot x_i$$

Идём в противоположную сторону:

$$w_t = w_{t-1} + 0.001 \cdot 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w_{t-1}) \cdot x_i$$

Градиентный спуск

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w)^2 \to \min_w$$

Градиент:

$$\nabla L(w) = -2 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w) \cdot x_i$$

Идём в противоположную сторону:

$$w_t = w_{t-1} + 0.001 \cdot 2 \cdot \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w_{t-1}) \cdot x_i$$

Дорого постоянно считать такие суммы по всей выборке

Стохастический градиентный спуск (SGD)

Проблема оптимизации:

$$L(w) = \sum_{i=1}^{n} L(w, x_i, y_i) \to \min_w$$

Инициализация w_0

рандомно выбрали i

$g_t = \nabla L(w_{t-1}, x_i, y_i)$

$w_t = w_{t-1} - \eta_t \cdot g_t$

if $||w_t - w_{t-1}|| < \varepsilon$:

break

Обратите внимание! Знака суммы нет!

так было в обычном градиентном спуске

Проблема оптимизации:

$$L(w) = \sum_{i=1}^{n} L(w, x_i, y_i) \to \min_w$$

Инициализация w_0

$g_t = \frac{1}{n} \sum_{i=1}^{n} \nabla L(w, x_i, y_i)$

$w_t = w_{t-1} - \eta_t \cdot g_t$

if $||w_t - w_{t-1}|| < \varepsilon$:

break

Градиентный спуск

Проблема оптимизации:

$$L(w) = \frac{1}{n} \cdot \sum_{i=1}^{n} (y_i - x_i^T w)^2 \rightarrow \min_w$$

Градиент:

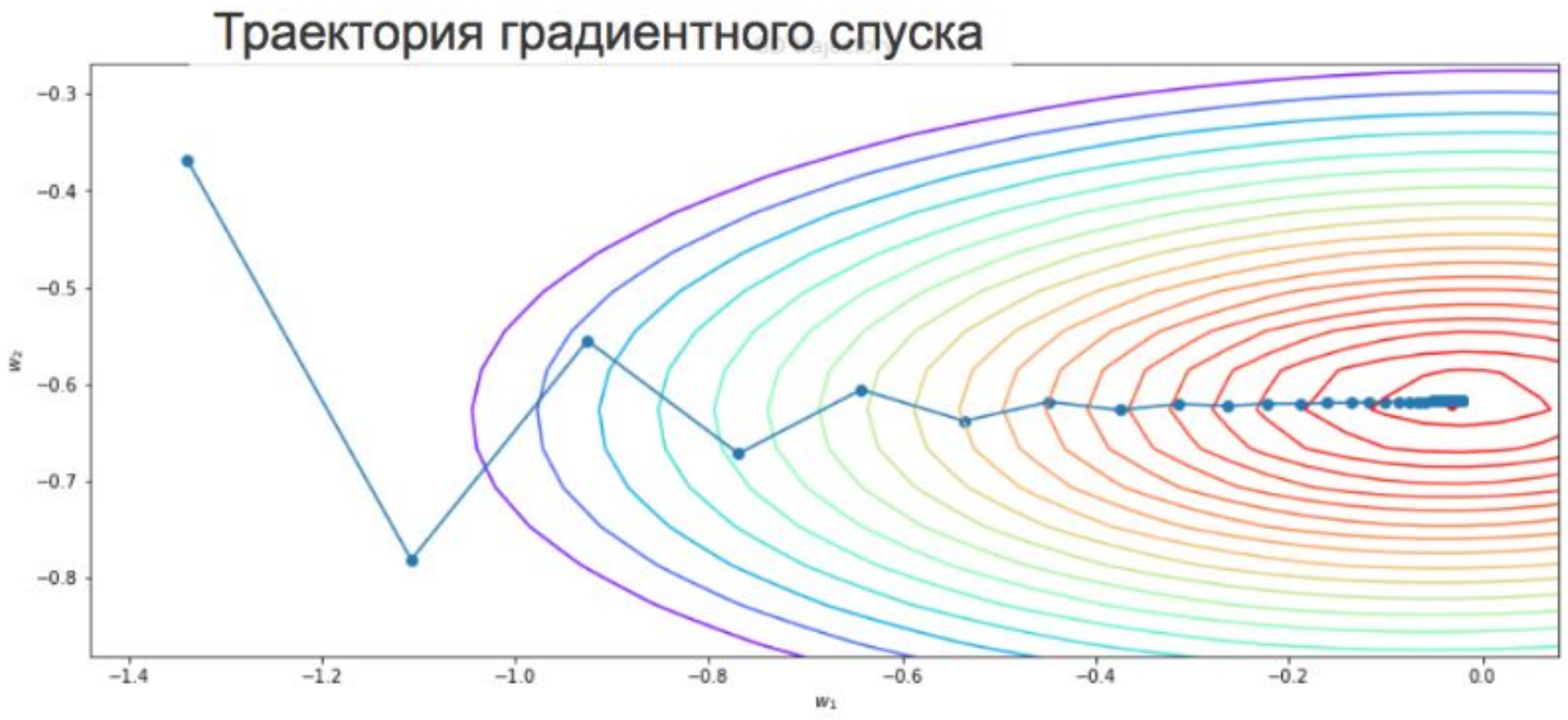
$$\nabla L(w) = -2 \cdot (y_i - x_i^T w) \cdot x_i$$

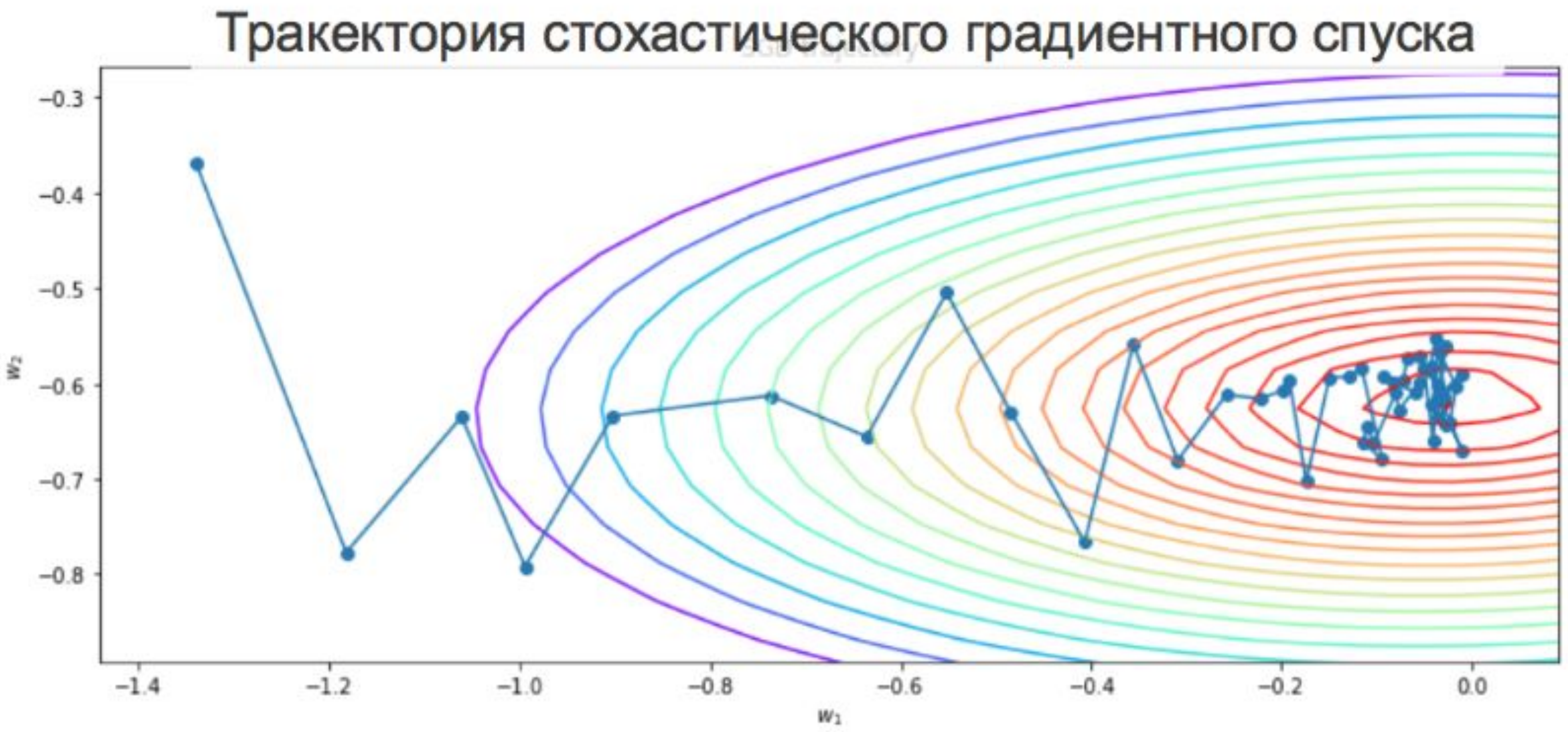
Идём в противоположную сторону:

$$w_t = w_{t-1} + 0.001 \cdot 2 \cdot (y_i - x_i^T w_{t-1}) \cdot x_i$$

Обратите внимание! Знака суммы нет!

Градиентный спуск





Градиентный спуск

Функция потерь MSE:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - (ax_i + b))^2$$

Градиенты функции потерь MSE по параметрам a и b определяются как:

$$\frac{\partial \text{MSE}}{\partial a} = -\frac{2}{n} \sum_{i=1}^{n} x_i (y_i - (ax_i + b))$$

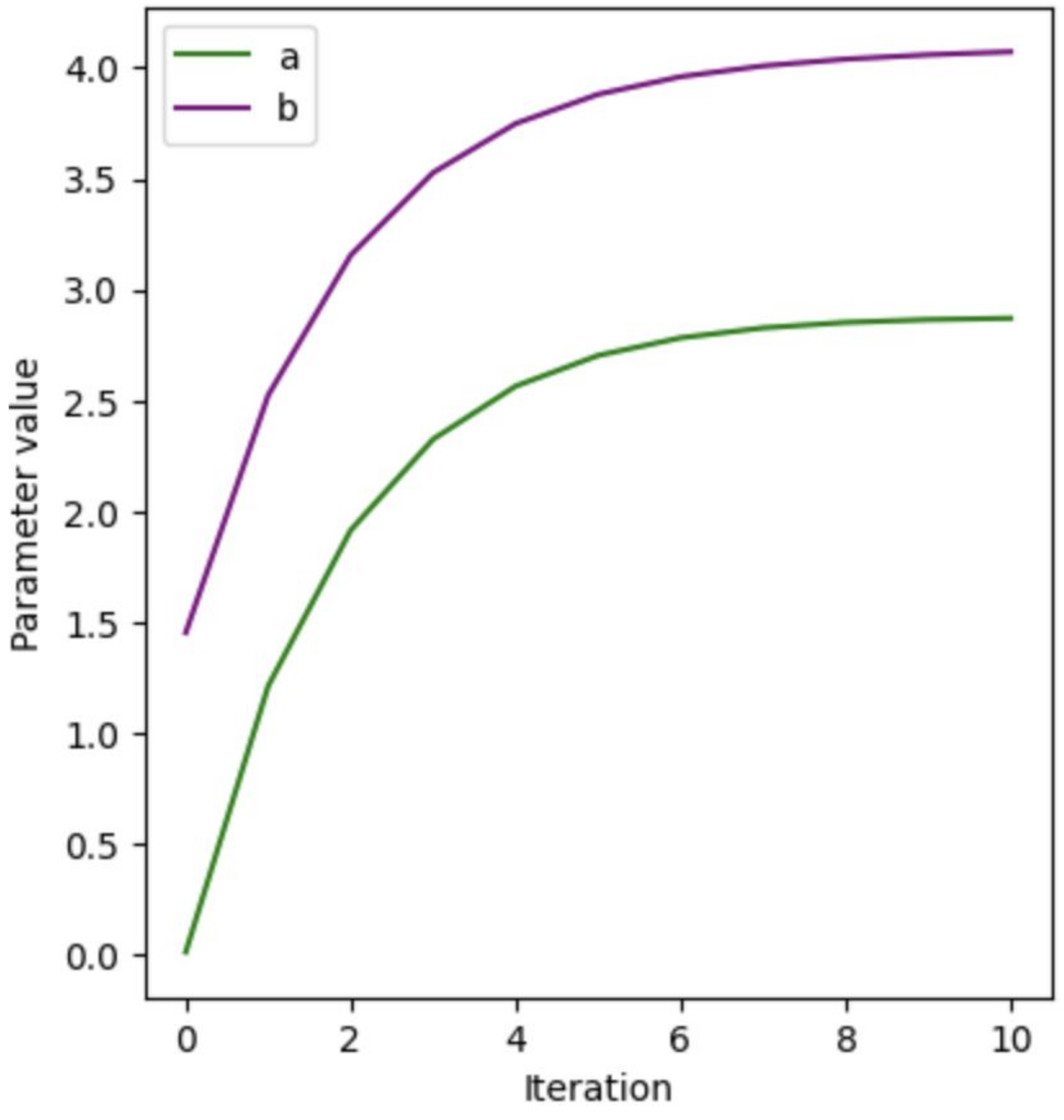
$$\frac{\partial \text{MSE}}{\partial b} = -\frac{2}{n} \sum_{i=1}^{n} (y_i - (ax_i + b))$$

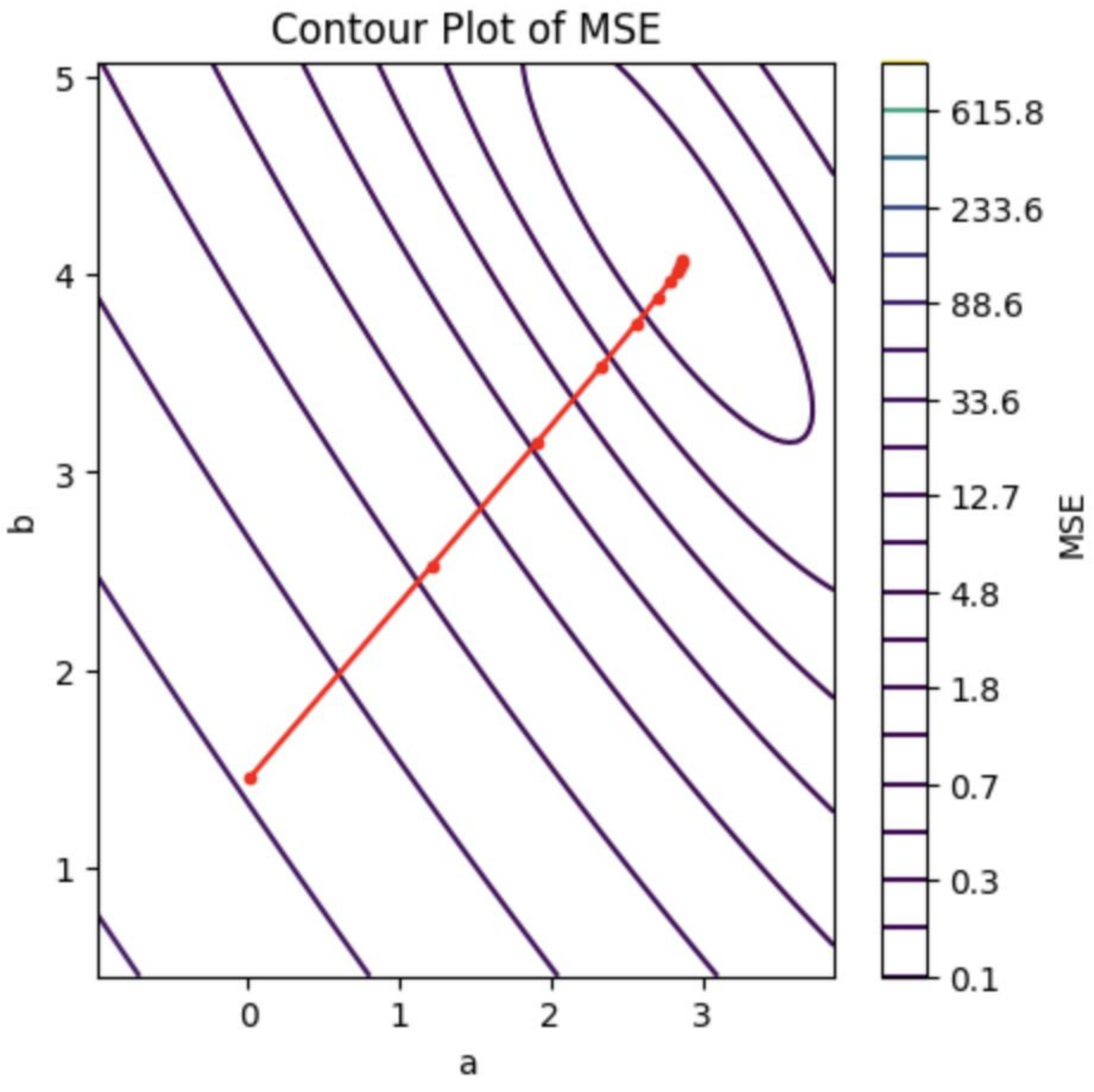
Параметры линейной регрессии будем обновлять следующим образом:

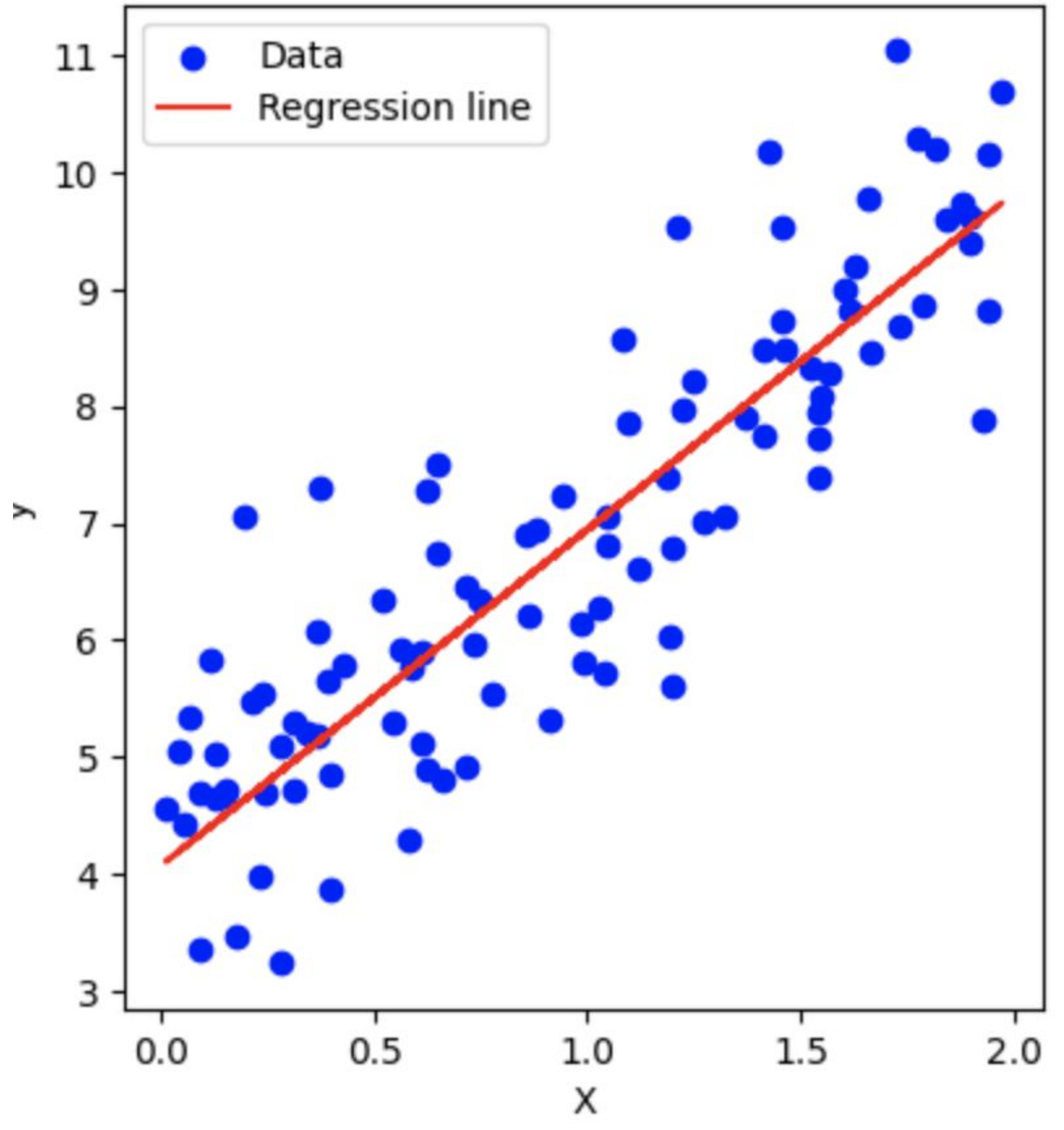
$$a_{n+1} = a_n - \text{learning rate} \cdot \frac{\partial \text{MSE}}{\partial a}$$

$$b_{n+1} = b_n - \text{learning rate} \cdot \frac{\partial \text{MSE}}{\partial b}$$

Градиентный спуск







Градиентный спуск

Обычный градиентый спуск

$$\frac{\partial \mathrm{MSE}}{\partial a} = -\frac{2}{n}\sum_{i=1}^{n} x_i(y_i - (ax_i + b))$$
$$\frac{\partial \mathrm{MSE}}{\partial b} = -\frac{2}{n}\sum_{i=1}^{n}(y_i - (ax_i + b))$$

$$a_{n+1} = a_n - \text{learning rate} \cdot \frac{\partial \mathrm{MSE}}{\partial a}$$

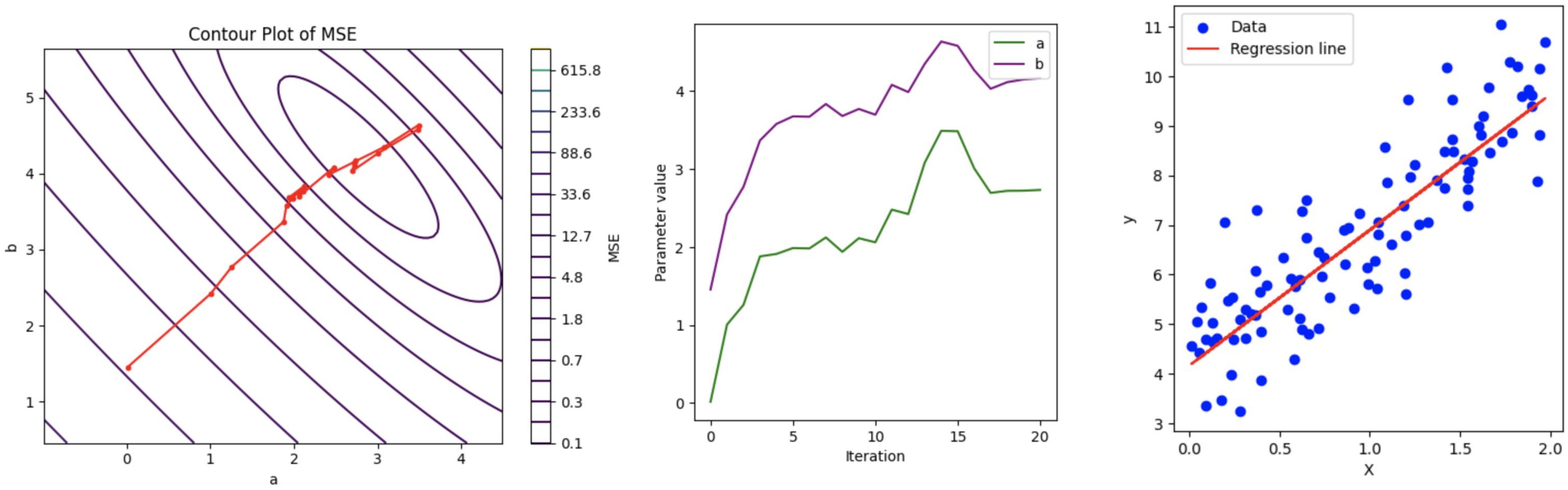
$$b_{n+1} = b_n - \text{learning rate} \cdot \frac{\partial \mathrm{MSE}}{\partial b}$$

В **стохастическом** градиентном спуске обновление параметров происходит на основе одного случайно выбранного примера данных за раз.

$$a_{n+1} = a_n - \text{learning rate} \cdot (-2x_i(y_i - (a_n x_i + b_n)))$$

$$b_{n+1} = b_n - \text{learning rate} \cdot (-2(y_i - (a_n x_i + b_n)))$$

Стохастический градиентный спуск



Разные функции потерь

$$\mathrm{MSE}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2$$

$$\mathrm{MAE}(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} |a(x_i) - y_i|$$

Какую функцию потерь (Loss function) выбрать?

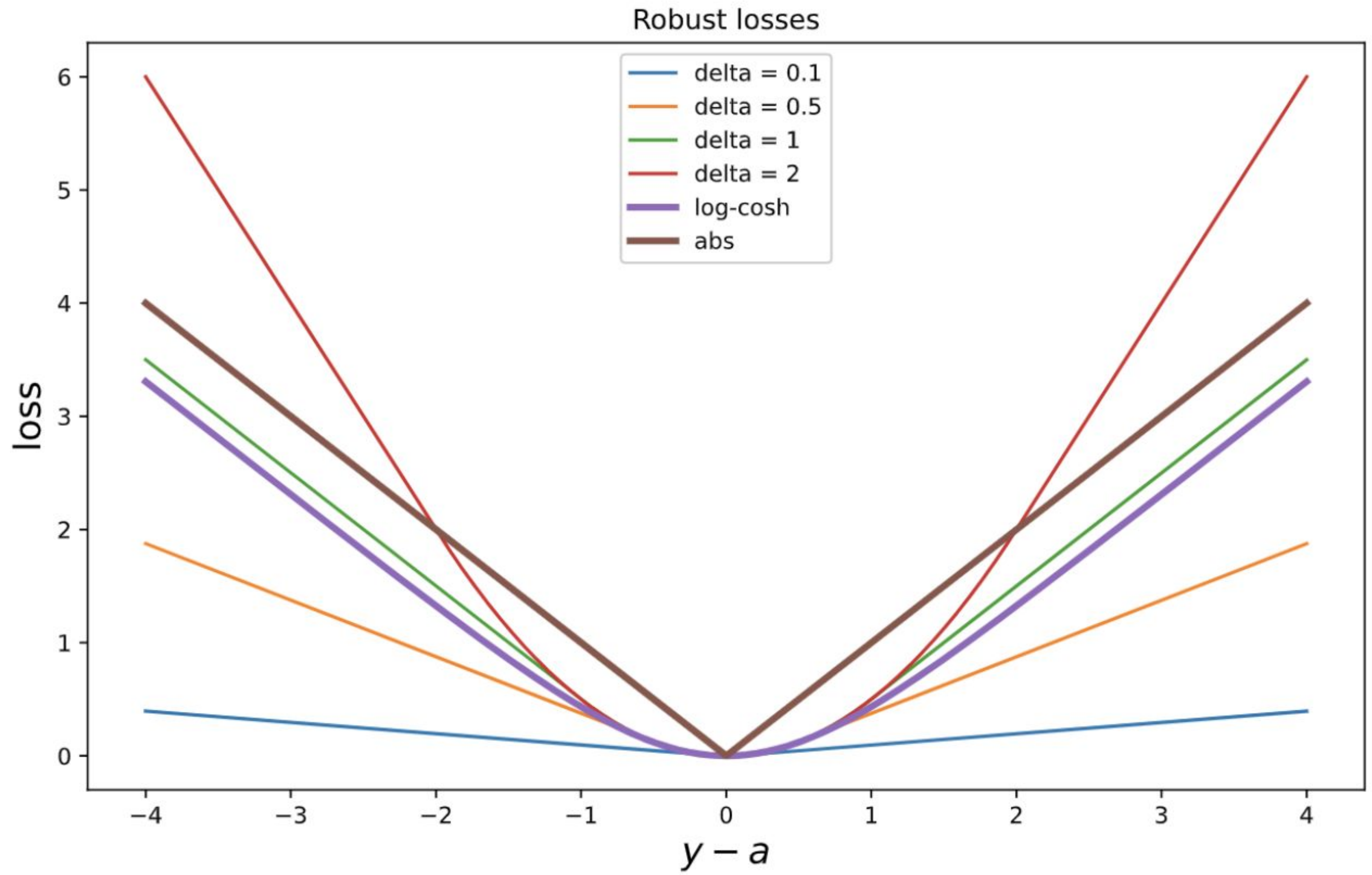
y	$a_1(x)$	$(a_1(x) - y)^2$	$\|a_1(x) - y\|$	$a_2(x)$	$(a_2(x) - y)^2$	$\|a_2(x) - y\|$
1	2	1	1	4	9	3
2	1	1	1	5	9	3
3	2	1	1	6	9	3
4	5	1	1	7	9	3
5	6	1	1	8	9	3
100	7	8649	93	10	8100	90
7	6	1	1	10	9	3
		MSE = 1236	MAE = 14.14		MSE = 1164	MAE = 15.43

Линейная регрессия. Общий случай
Обучение

Но у модуля проблемы с производной в 0, поэтому используют немного другие функции: Huber loss и Log-Cosh

$$L_\delta(y,a) = \begin{cases} \frac{1}{2}(y-a)^2, & |y-a| < \delta \\ \delta\left(|y-a| - \frac{1}{2}\delta\right), & |y-a| \geqslant \delta \end{cases}$$

$$L(y,a) = \log\cosh(a-y)$$



УНИВЕРСИТЕТ
ИННОПОЛИС

ВОПРОСЫ И ОТВЕТЫ