

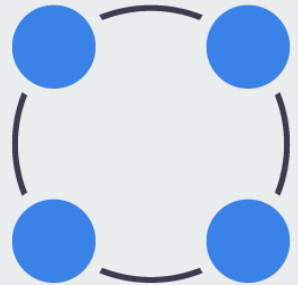


# Статистика и анализ данных в R

Лекция 7. Линейные модели. Часть 1

(29.10.2022)

Даниил Литвинов



---

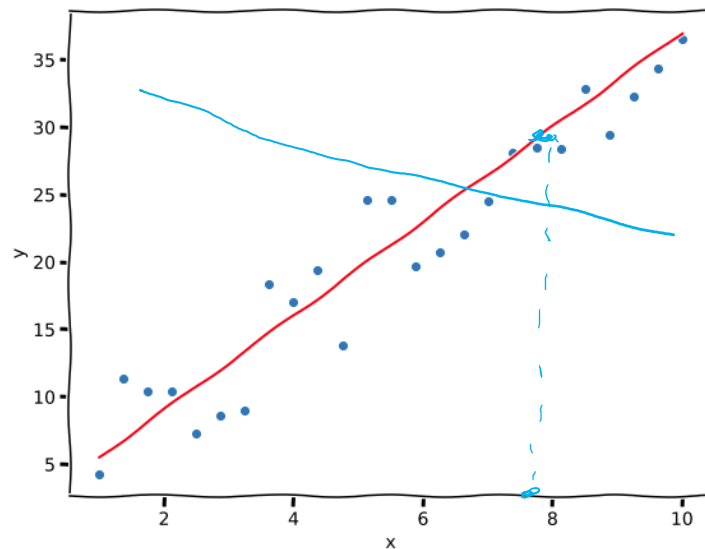
# Мягкое введение

# Что это такое?

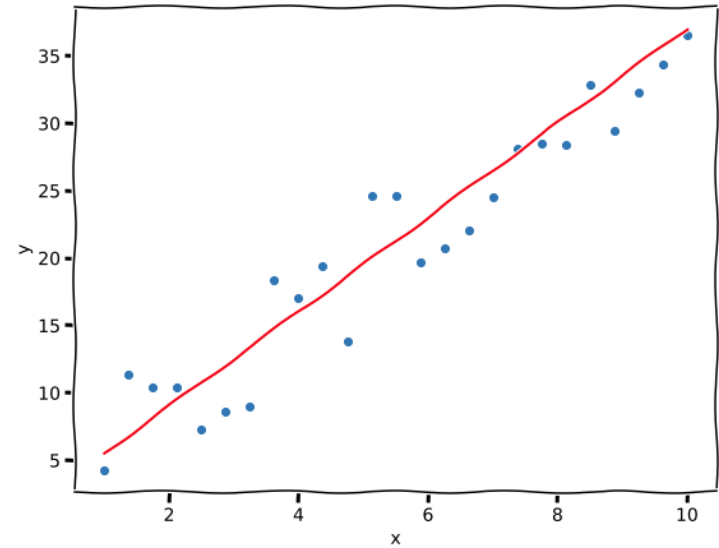
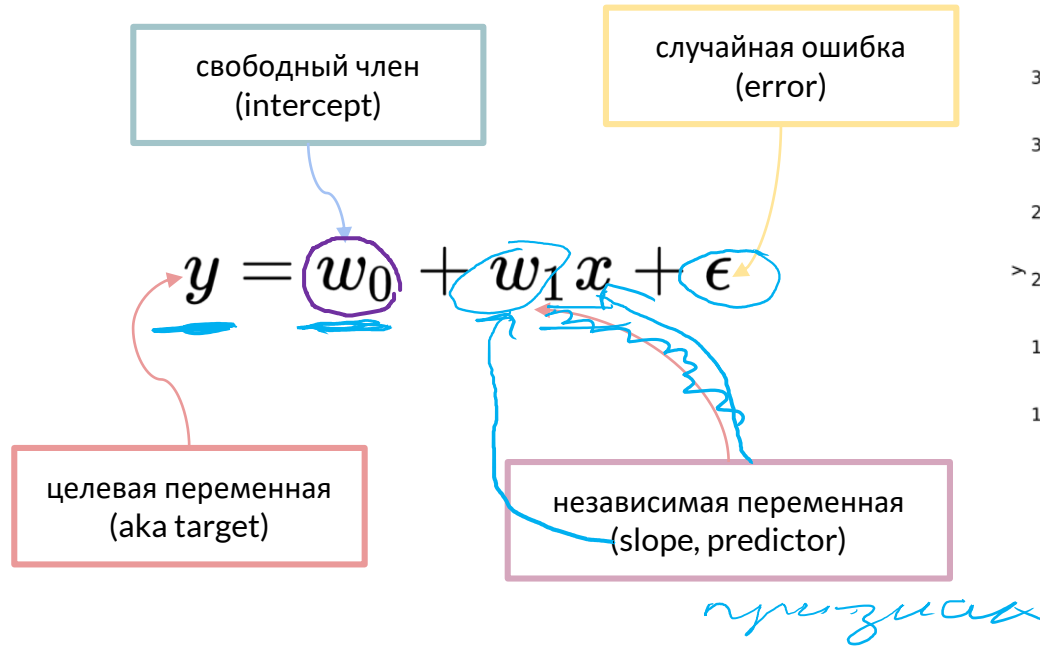
x — баллы за экзамен по английскому 1

y — баллы за экзамен по английскому 2

x	y
1	5
3	11
9	35
10	33



# Что это такое?

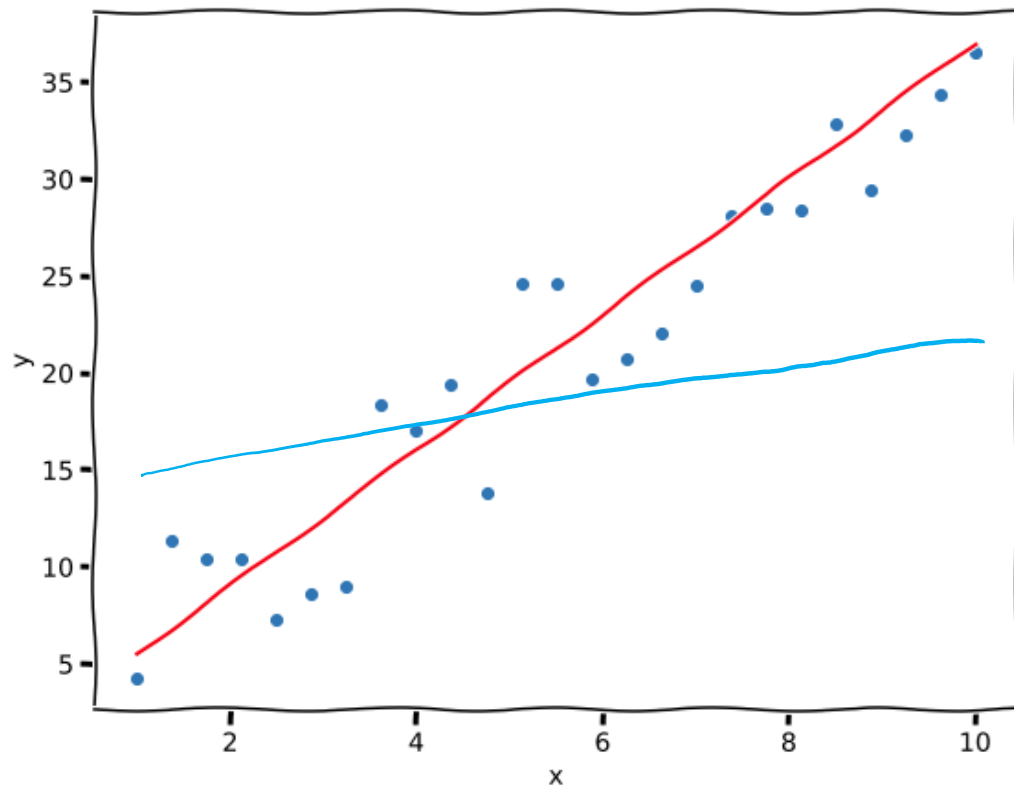


# А какая модель нам нужна?

$$y = w_0 + w_1 x + \epsilon$$

$$\hat{y} = w_0 + w_1 x$$

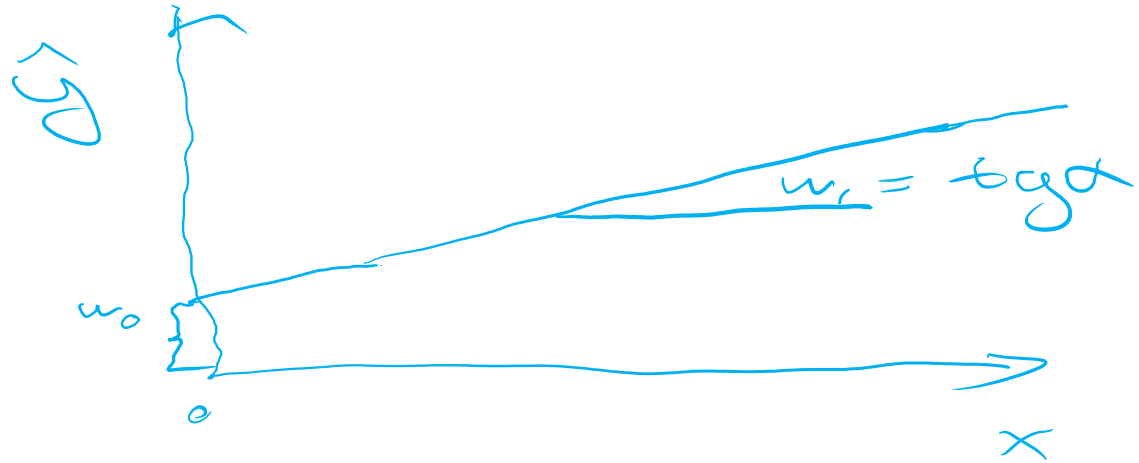
$$\sum_{i=1}^n (y_i - \hat{y}_i)^2$$



# Интерпретация коэффициентов



$$\hat{y} = w_0 + w_1 x$$



# Зачем нужны линейные модели?



$$\text{цена} = w_0 + w_1 \cdot S + w_2 \cdot \text{расст} \cdot \text{go} \cdot \text{миф}$$

1. Предсказание интересующей нас величины
2. Оценка влияния различных факторов на нашу целевую переменную
3. Линейные модели очень легко использовать и интерпретировать
4. Линейные модели могут восстанавливать даже **нелинейные зависимости**

t-test

lm

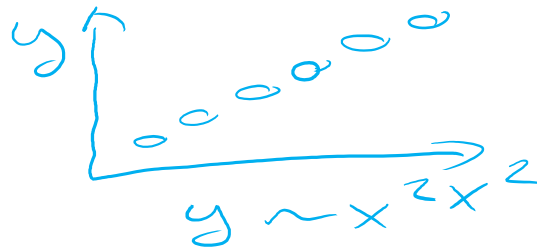
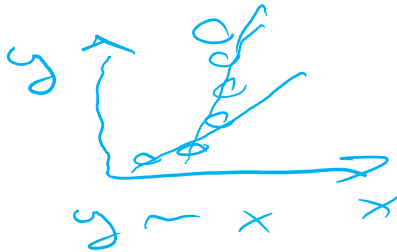
$$y = w_1 + 0 \cdot x$$

$$y = w_1 + \underline{10}x$$



ко параметру 0  
 $x = \text{равное} - 1$

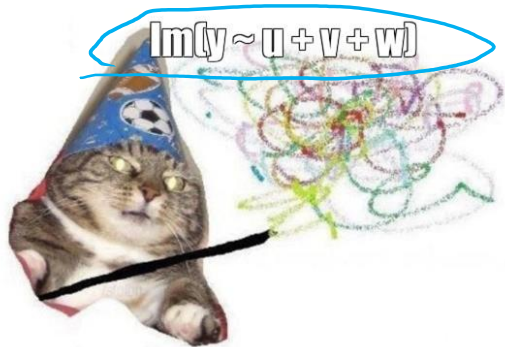
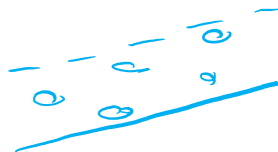
$$y \sim w_0 + \boxed{w_1}x$$



# А как искать эти ваши $w_0$ и $w_1$ ?

$$y - \hat{y} = \epsilon$$

↓  
min



$lm(y \sim u + v + w)$

regression

$$y = w_0 + w_1 u + w_2 v + w_3 w$$

> summary(m)

Call:

$lm(formula = y \sim u + v + w)$

Residuals:

Min	1Q	Median	3Q	Max
-3.3965	-0.9472	-0.4708	1.3730	3.1283

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.4222	1.4036	1.013	0.32029
u	1.0359	0.2811	3.685	0.00106 **
v	0.9217	0.3787	2.434	0.02211 *
w	0.7261	0.3652	1.988	0.05744 .

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.625 on 26 degrees of freedom

Multiple R-squared: 0.4981, Adjusted R-squared: 0.4402

F-statistic: 8.603 on 3 and 26 DF, p-value: 0.0003915



## Как оценивать коэффициенты модели?

$$y = w_0 + w_1 x + \varepsilon; \quad \hat{y} = w_0 + w_1 x; \quad y - \hat{y} = \varepsilon$$

$$\text{Loss} = \sum_{i=1}^n (y_i - w_0 - w_1 x_i)^2 \rightarrow \min$$

↑  
MLK  
OLS



# Таблица производных



$y = f(x)$	$\frac{dy}{dx} = f'(x)$
$k$ , any constant	0
$x$	1
$x^2$	$2x$
$x^3$	$3x^2$
$x^n$ , any constant $n$	$nx^{n-1}$
$e^x$	$e^x$
$e^{kx}$	$ke^{kx}$
$\ln x = \log_e x$	$\frac{1}{x}$
$\sin x$	$\cos x$
$\sin kx$	$k \cos kx$
$\cos x$	$-\sin x$
$\cos kx$	$-k \sin kx$

цели?  $w_0 = \text{med.} |y - \hat{y}|$

$$\frac{\partial \text{Loss}}{\partial w_0} = \sum_{i=1}^n -2 \cdot (y_i - w_0 - w_1 x_i)$$


$$\frac{\partial \text{Loss}}{\partial w_1} = \sum_{i=1}^n -2 \cdot x_i (y_i - w_0 - w_1 x_i)$$

$$\left. \begin{array}{l} \hat{y} = w_0 \\ w_0 = 15 \end{array} \right\} \text{merk} \quad \left\{ \begin{array}{l} \frac{\partial \text{Loss}}{\partial w_0} = 0 \\ \frac{\partial \text{Loss}}{\partial w_1} = 0 \end{array} \right.$$

---

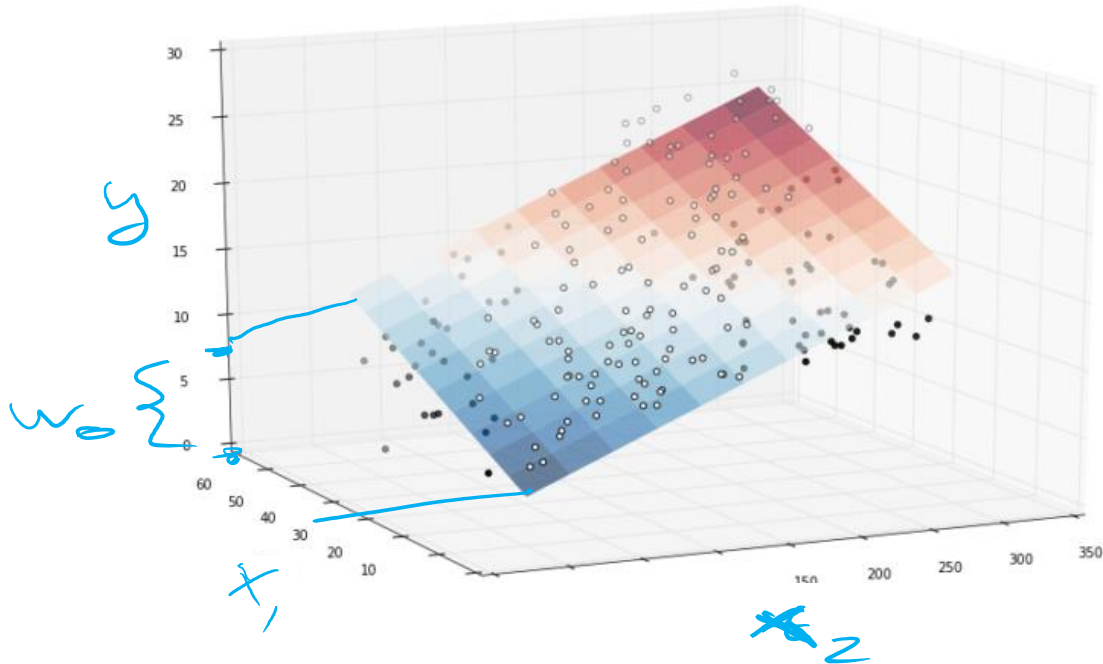
# Отдых

## А если у нас много независимых переменных?


$$\underline{y} = \underline{w_0} + \underline{w_1 x} + \underline{w_2 z} + \underline{\dots} + \underline{w_n t} + \epsilon$$

площадь	число комнат	школа близко	цена квартиры
50	2	нет	5000
1000	7	да	11000
30	1	нет	3500
100	4	нет	33333

# Множественная линейная регрессия дает нам плоскость



## Как оценивать коэффициенты модели теперь?

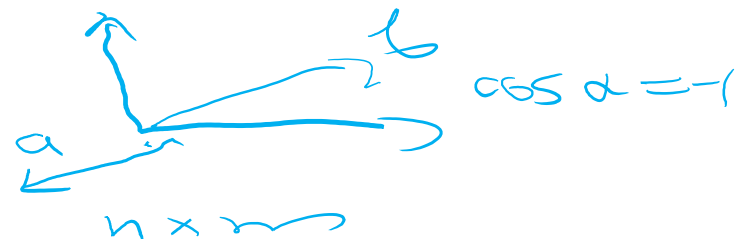

$$\hat{y} = w_0 + w_1 x_1 + w_2 x_2$$

$$\text{Loss} = \sum (y - w_0 - w_1 x_1 - w_2 x_2)^2$$

# Капелька линейной алгебры



$$M = \left( \begin{array}{c} \\ \end{array} \right) \quad \begin{array}{l} n - \text{строк} \\ m - \text{столбцов} \end{array}$$



• скалярное произв:  
 $a, b$  - векторы



$$|a| \cdot |b| \cdot \cos \alpha$$

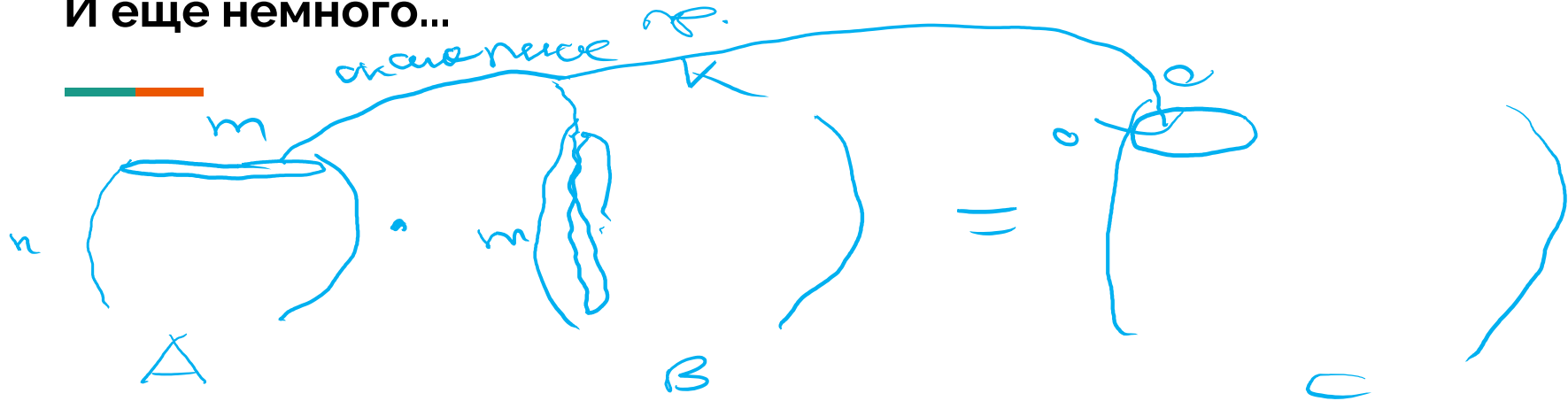
$$|a| = \sqrt{a_0^2 + a_1^2 + \dots}$$

$$\sum_{i=1}^n a_i \cdot b_i = (a, b)$$

$$\cos \alpha = \frac{(a, b)}{|a| \cdot |b|}$$



И еще немного...



$$2 \cdot \frac{1}{2} = 1$$

$$n \underset{m}{A} \cdot n \underset{n}{A}^{-1} = E \quad n \begin{pmatrix} 1 & & \\ 0 & 1 & \\ & & \ddots \\ & & & 1 \end{pmatrix}$$

$$A \cdot E = D$$

# МНК в матричном виде

$$y = w_0 + w_1 x + w_2 z + \dots + w_n t + \epsilon$$

$y$  цена за кв.  $n \times 1$   
 $\begin{bmatrix} 100 \\ \vdots \\ 500 \end{bmatrix}$   
 $X$  ~~матрица~~  $S, \text{кв.ст.}, \dots$   $n \times 4$   
 $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 15 \\ 20 \\ \vdots \end{bmatrix}$   
 $n$   $\nearrow$   $\text{ценные метры}$   
 $w$   $4 \times 1$ :  $\begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{pmatrix}$   
 $x_1 \cdot w_1 + x_2 \cdot w_2 + \dots$   
 $\hat{y} = X \cdot W$   
 $n \times 3 \quad 3 \times 1 \quad n \times 1$   
 $\begin{matrix} 3 \\ X \end{matrix} \cdot \begin{pmatrix} 1 \\ 15 \\ 20 \end{pmatrix} \rightarrow y_1$

## МНК в матричном виде. Функция потерь

$$\text{Loss} = \sum_i (y_i - \hat{y}_i)^2 = (y - Xw) \cdot (y - Xw)$$

Dimensions:  $1 \times n$  (for  $y$ ),  $n \times 4$  (for  $X$ ),  $4 \times 1$  (for  $w$ )

Matrix representation of residuals:

$$\begin{bmatrix} y_1 - \hat{y}_1 \\ y_2 - \hat{y}_2 \\ \vdots \end{bmatrix}$$

Handwritten note:  $\hat{y} = Xw$

$$\text{Loss} = (y - Xw)^T \cdot (y - Xw)$$

# Подбор коэффициентов в R и Python



## Продолжаем сами искать коэффициенты

$$\text{Loss} = (y - Xw)^T \cdot (y - Xw)$$

$w$   
 $4 \times 1$   
 $\uparrow$   
 $w_0, w_1, w_2, w_3$

$$\begin{bmatrix} \frac{\partial \text{Loss}}{\partial w_0} \\ \frac{\partial \text{Loss}}{\partial w_1} \\ \vdots \\ \frac{\partial \text{Loss}}{\partial w_3} \end{bmatrix} = 0 = \frac{\partial \text{Loss}}{\partial w} - \text{градиент}$$

$$\frac{d \text{Loss}}{d w} = 2 \cdot (y - Xw) \cdot \frac{\partial (y - Xw)}{\partial w} \quad \left\{ \begin{array}{l} \text{Loss} = (y - Xw)^T (y - Xw) \\ \text{SS} \end{array} \right.$$

$$= 2 \cdot \underbrace{X^T}_{4 \times 4} \cdot \underbrace{(y - Xw)}_{n \times 1}$$

↓  
4 × 1

X  
n × 4

= 0

$$2 \cdot X^T \cdot (Xw - y)$$

~~~~~

$$(y - \hat{y})^2$$

Приравняем градиент к нулю...

$$2. X^T \cdot (Xw - y) = 0$$

$$X^T \cdot Xw - X^T y = 0$$

$$X^T X w = X^T y$$

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

$\Rightarrow$

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

# Проблемы с обратной матрицей





# p-value для коэффициентов

```
> summary(m)
```

Call:

```
lm(formula = y ~ u + v + w)
```

Residuals:

| Min     | 1Q      | Median  | 3Q     | Max    |
|---------|---------|---------|--------|--------|
| -3.3965 | -0.9472 | -0.4708 | 1.3730 | 3.1283 |

Coefficients:

|             | Estimate | Std. Error | t value | Pr(> t ) |    |
|-------------|----------|------------|---------|----------|----|
| (Intercept) | 1.4222   | 1.4036     | 1.013   | 0.32029  |    |
| u           | 1.0359   | 0.2811     | 3.685   | 0.00106  | ** |
| v           | 0.9217   | 0.3787     | 2.434   | 0.02211  | *  |
| w           | 0.7261   | 0.3652     | 1.988   | 0.05744  | .  |

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.625 on 26 degrees of freedom

Multiple R-squared: 0.4981, Adjusted R-squared: 0.4402

F-statistic: 8.603 on 3 and 26 DF, p-value: 0.0003915



# А как нам посчитать дисперсию коэффициентов?



```
X = np.array([
    [1, 2, 3],
    [2, 3, 4],
    [10, 11, 9],
    [2, 7, 6]
])

x_t_x_inv = np.linalg.inv(X.T @ X)
x_t_x_inv.round(7) == x_t_x_inv.T.round(7)

array([[ True,  True,  True],
       [ True,  True,  True],
       [ True,  True,  True]])
```



---

**Отдых -> самое интересное**

# А если обратную матрицу совсем не хочется считать?

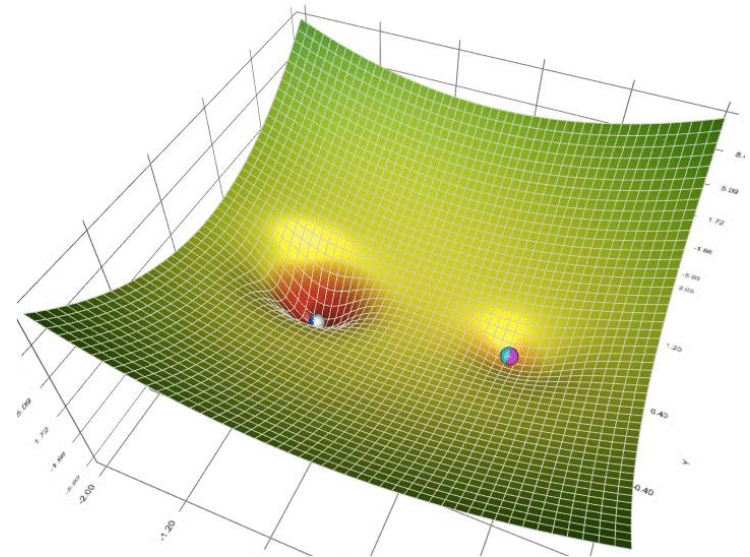
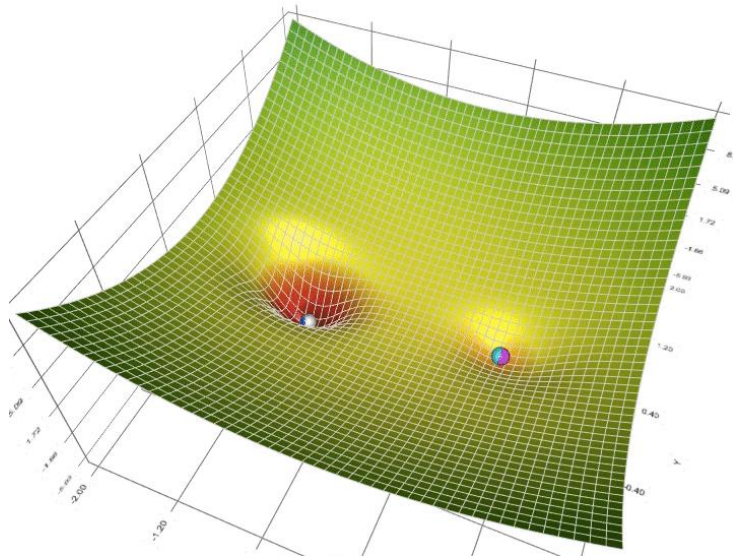


```
X = np.array([
    [1, 2, 3],
    [1, 2, 3.0001],
    [10, 11, 9]
])

x_inv = np.linalg.inv(X)
x_inv

array([[ -1.66678889e+04,  1.66666667e+04,  2.22222222e-01],
       [ 2.33344444e+04, -2.33333333e+04, -1.11111111e-01],
       [-1.00000000e+04,  1.00000000e+04, -0.00000000e+00]])
```

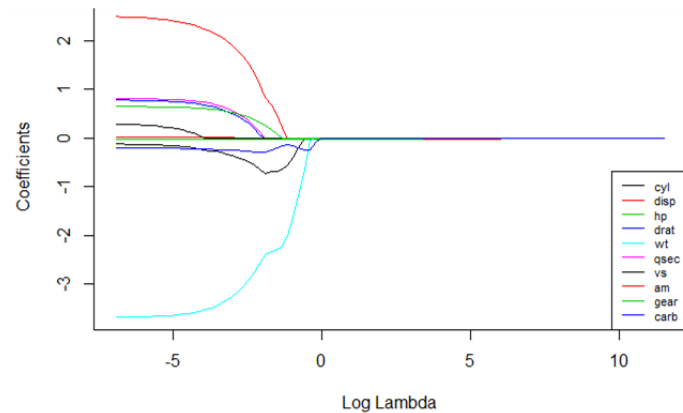
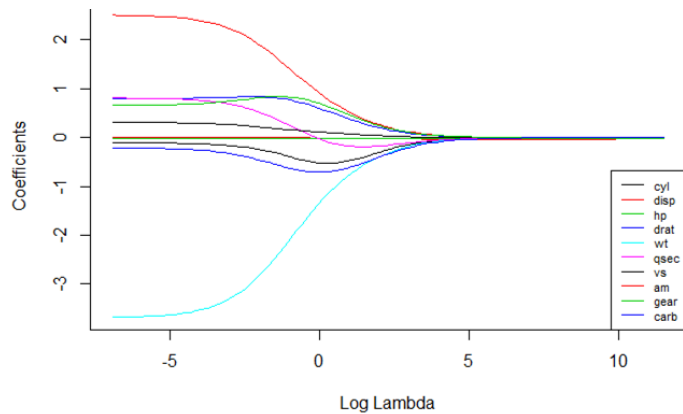
# Градиентный спуск



# Регуляризация



# Регуляризация





# Регуляризация

