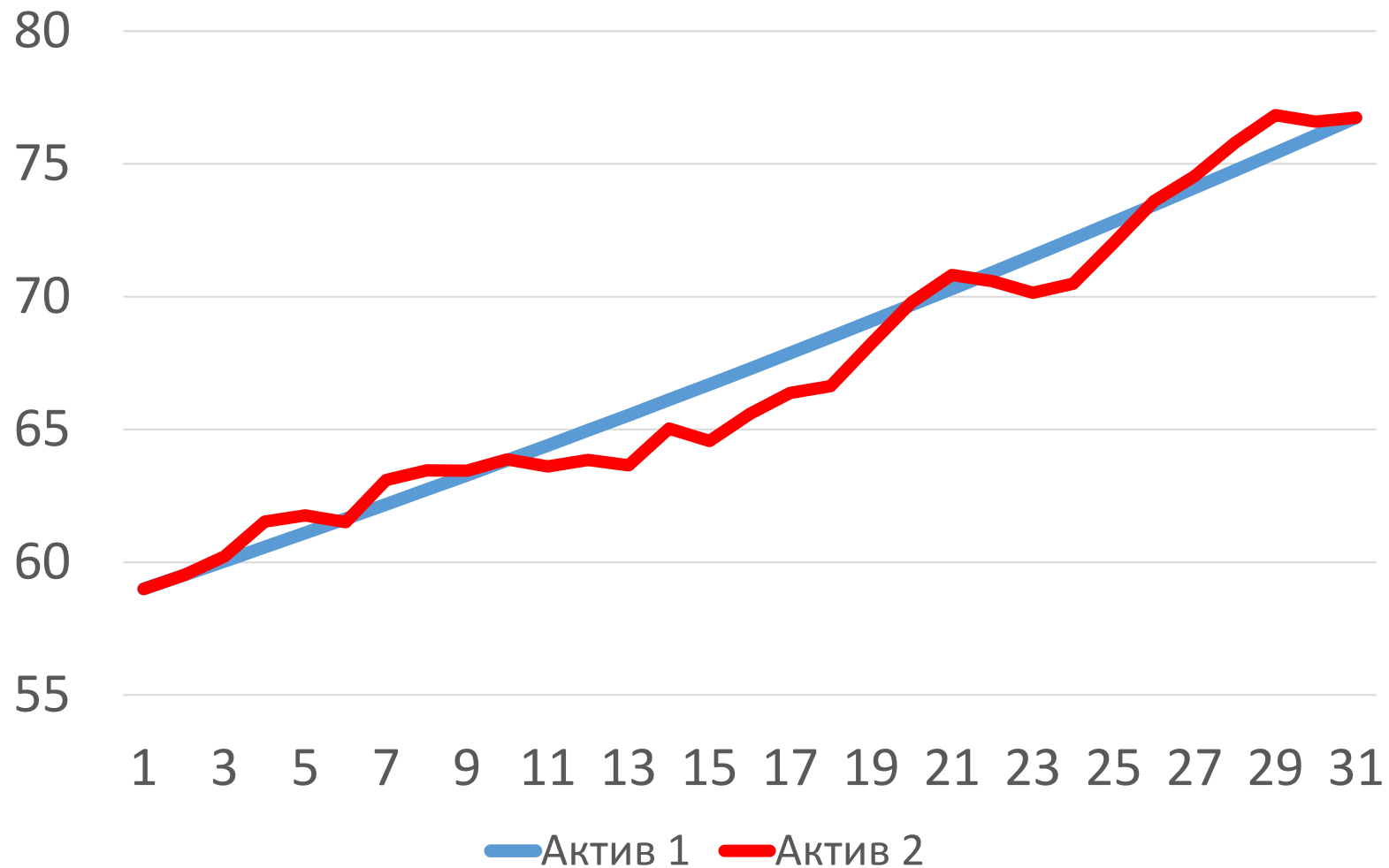


Задача об инвестиционном портфеле

Оценка актива (акции)



Прирост стоимости

Актив 1: ~30%

Актив 2: ~30%

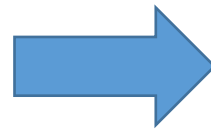
Что еще?

Оценка актива (акции). Риск

- Один из методов – дисперсия относительных суточных колебаний:

$$q_i = \frac{p_i}{p_{i-1}} - 1$$

День	Цена
1	59
2	59,5192
3	60,22153
4	61,53436



День	Изменение
1	
2	0,0088
3	0,0118
4	0,0218

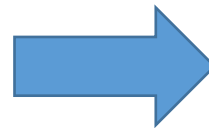
Оценка актива (акции). Риск

- Один из методов – дисперсия относительных суточных колебаний:

$$q_i = \frac{p_i}{p_{i-1}} - 1$$

Можно в
процентах

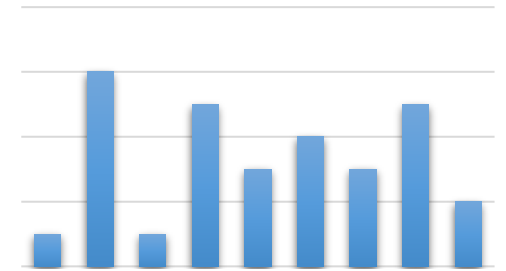
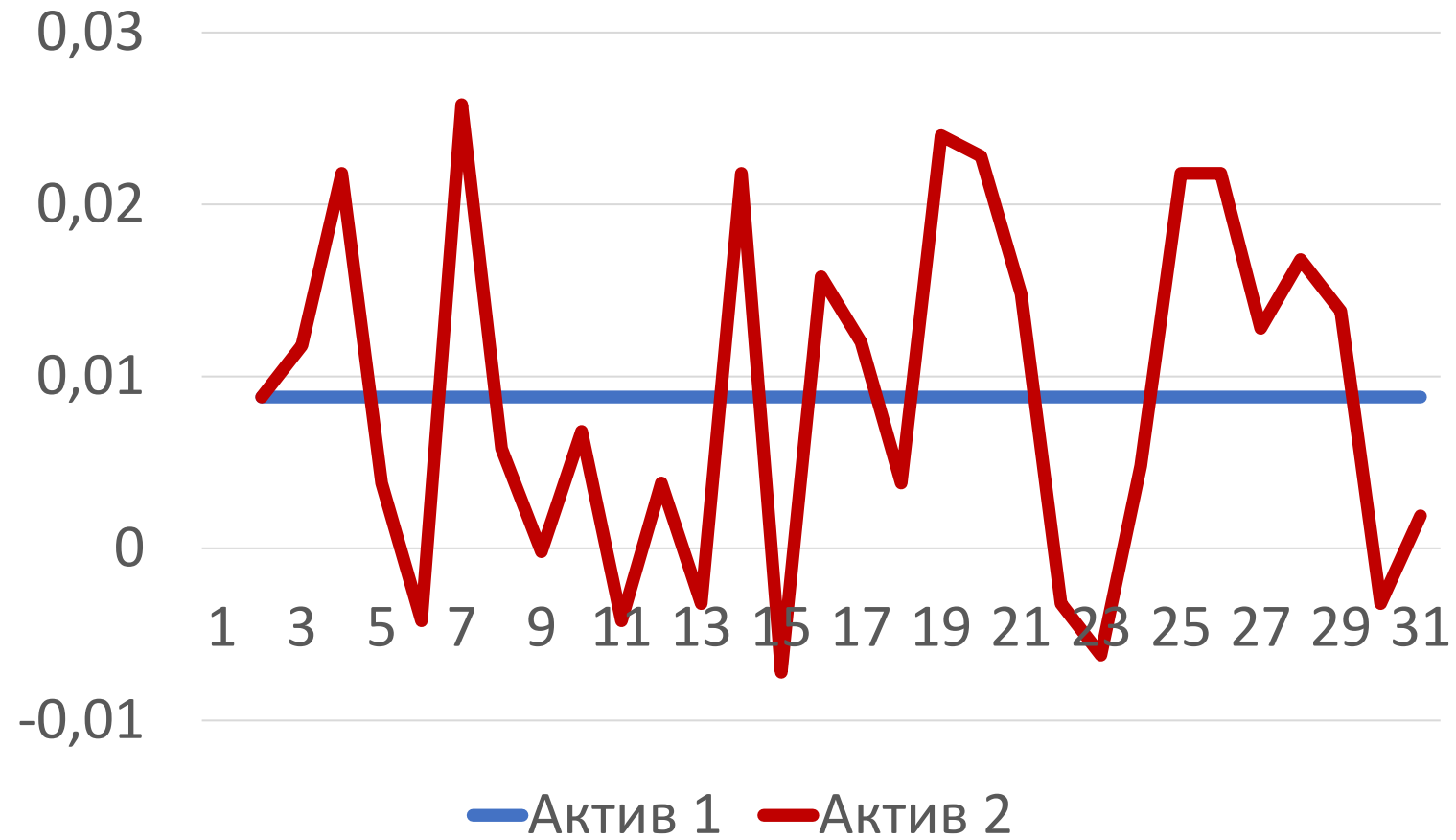
День	Цена
1	59
2	59,5192
3	60,22153
4	61,53436



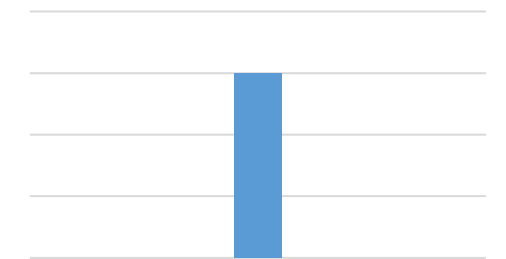
День	Изменение
1	
2	0,0088
3	0,0118
4	0,0218



Разброс приращений

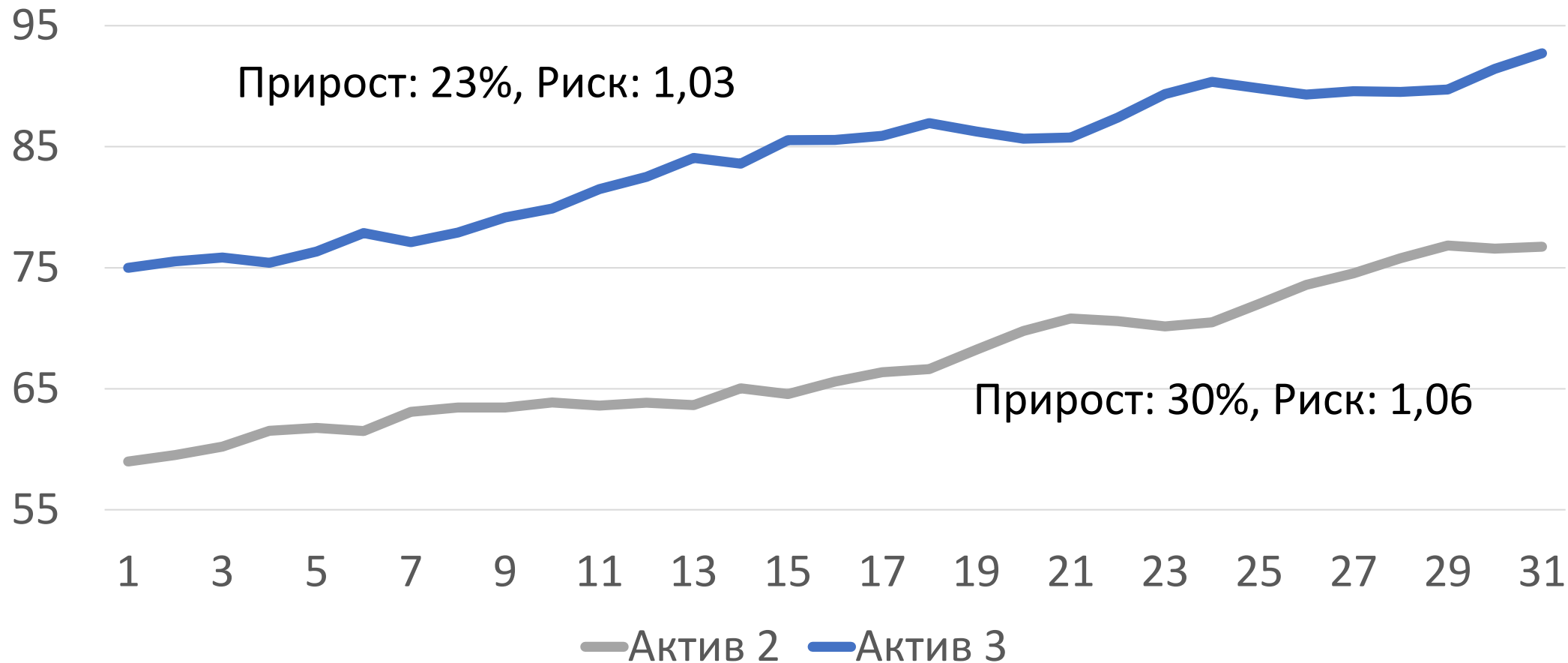


$D = 1,06$



$D = 0$

А что с портфелем?

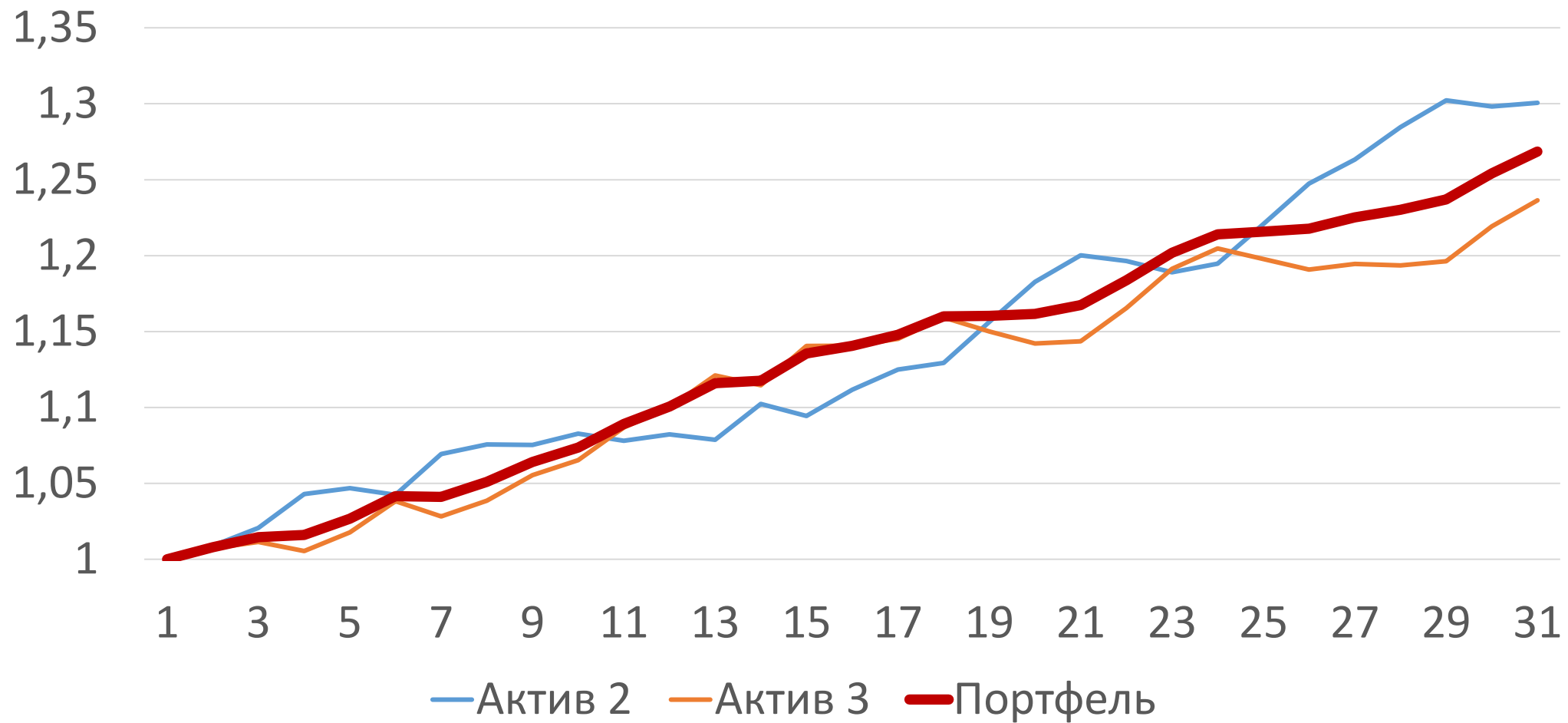


А что с портфелем?

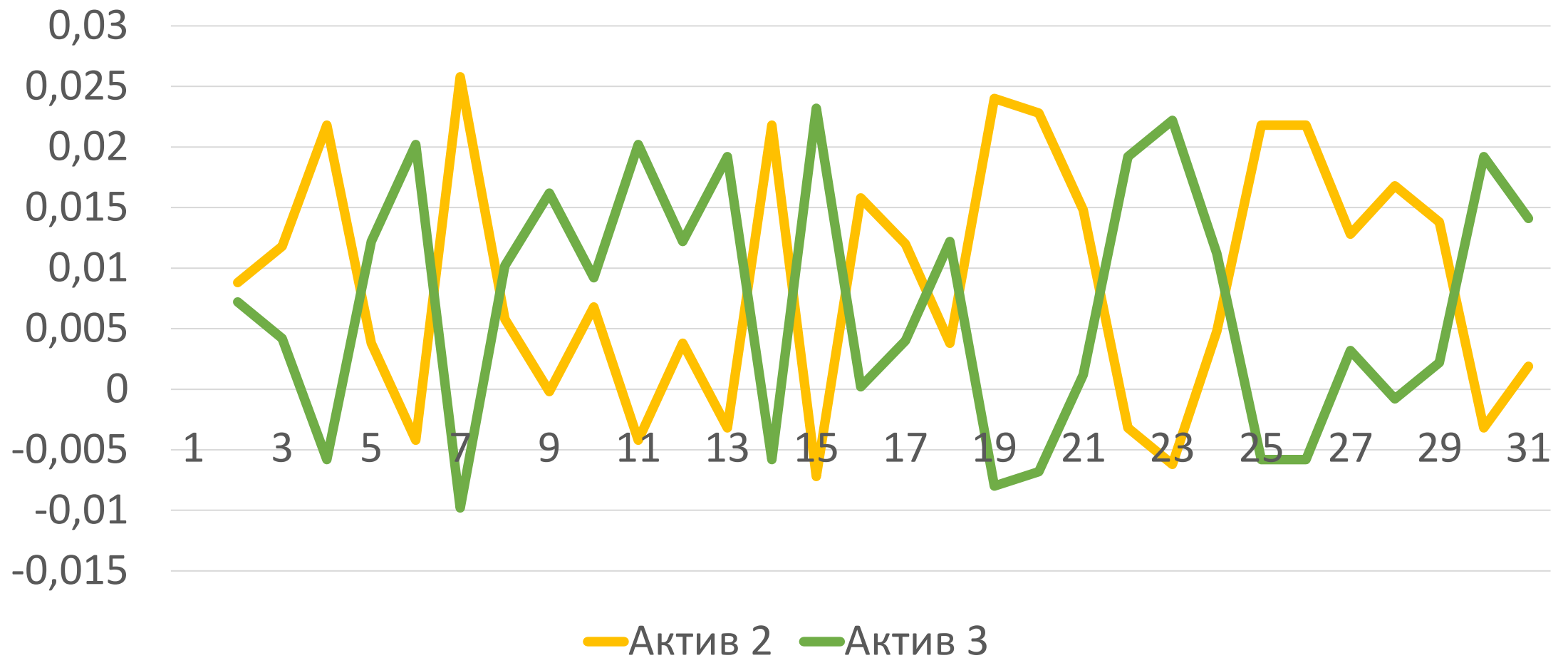
- Сформируем «портфель»:
 - 100 ед. актива 2 (на сумму 5900 ден. ед.)
 - 79 ед. актива 3 (на сумму 5925 ден. ед.)

День	Цена	Изменение
1	11825	
2	11919,58	0,007998
3	11997,02	0,006497
4	12015,1	0,001507

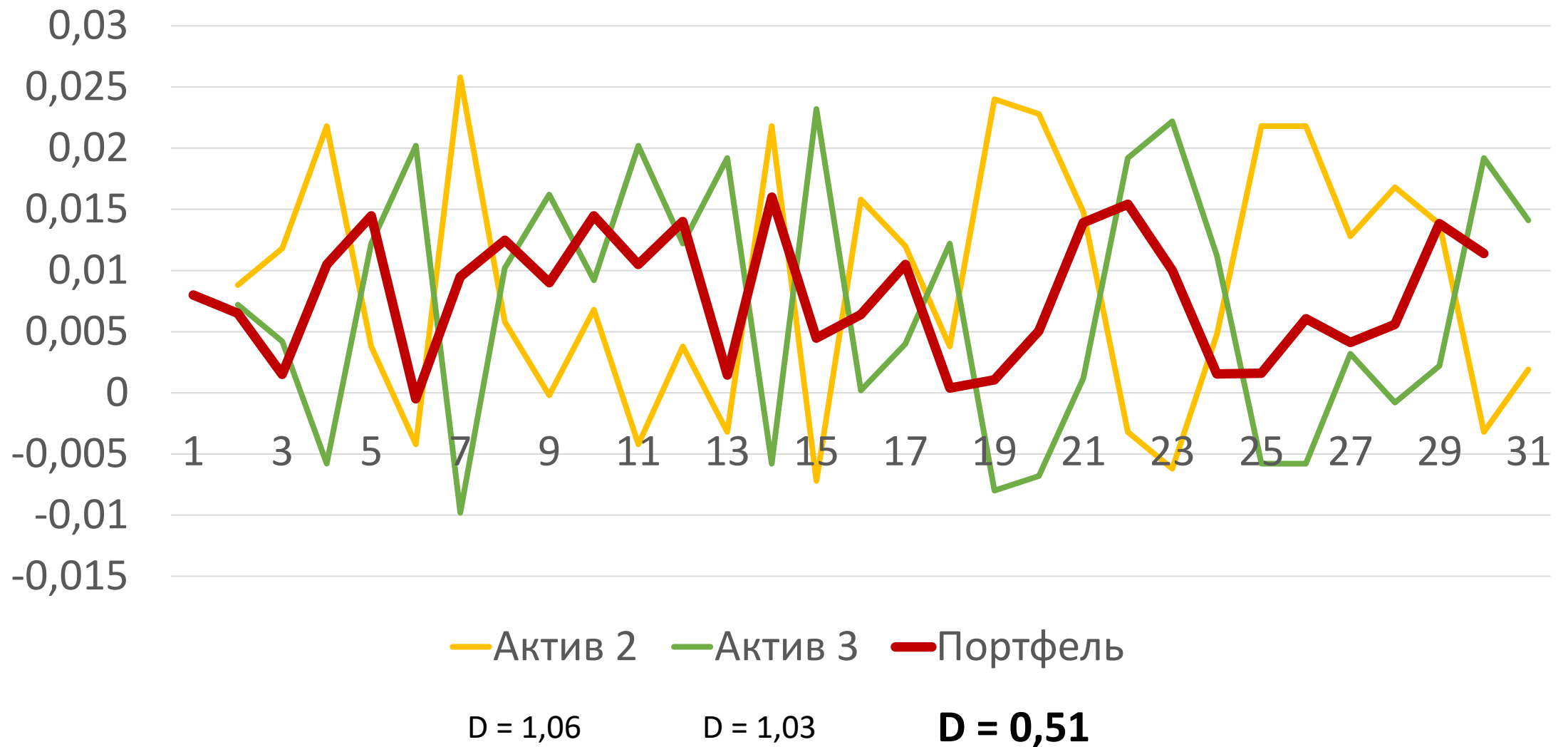
А что с портфелем?



А что с портфелем?



А что с портфелем?



Дисперсия суммы СВ

$$D[X + Y] = D[X] + D[Y] + 2cov(X, Y)$$

$$cov(X, Y) = \frac{1}{n} \sum_{t=1}^n (X_t - \bar{X}) (Y_t - \bar{Y})$$

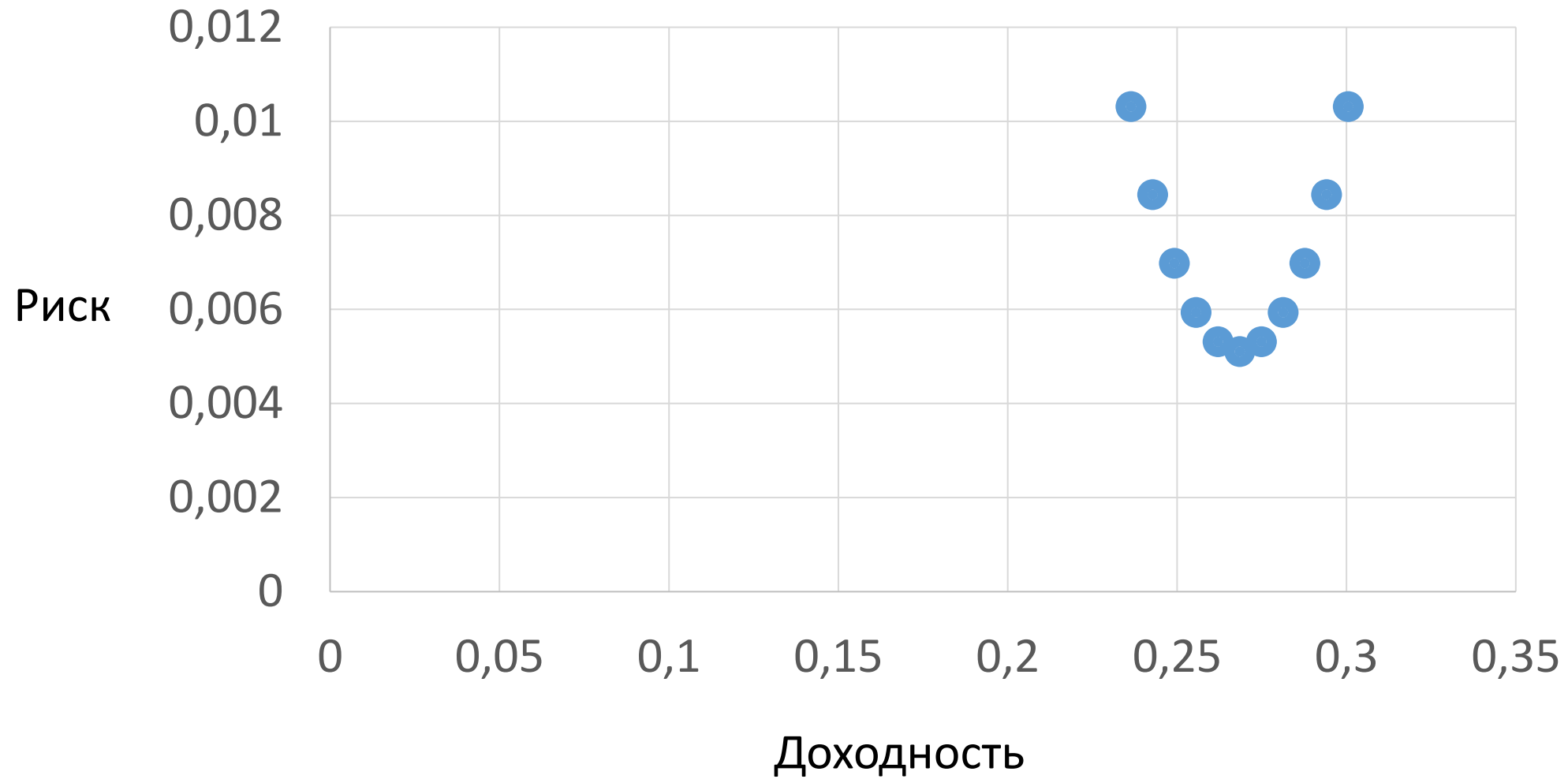
Формальная модель портфеля

- Пусть есть n различных активов, каждый из которых характеризуется приростом стоимости g_i и риском r_i (измеряемым дисперсией). Известна также попарная ковариация активов $cov_{i,j}$
- Тогда инвестиционный портфель описывается набором из n коэффициентов α_i , причем α_i показывает, какую долю в этом портфеле занимает соответствующий актив
 - Доходность портфеля: $\sum_n \alpha_i g_i$
 - Риск портфеля: $\sum_{1 \leq i \leq n} \alpha_i^2 r_i + 2 \sum_{1 \leq i < j \leq n} \alpha_i \alpha_j cov_{i,j}$

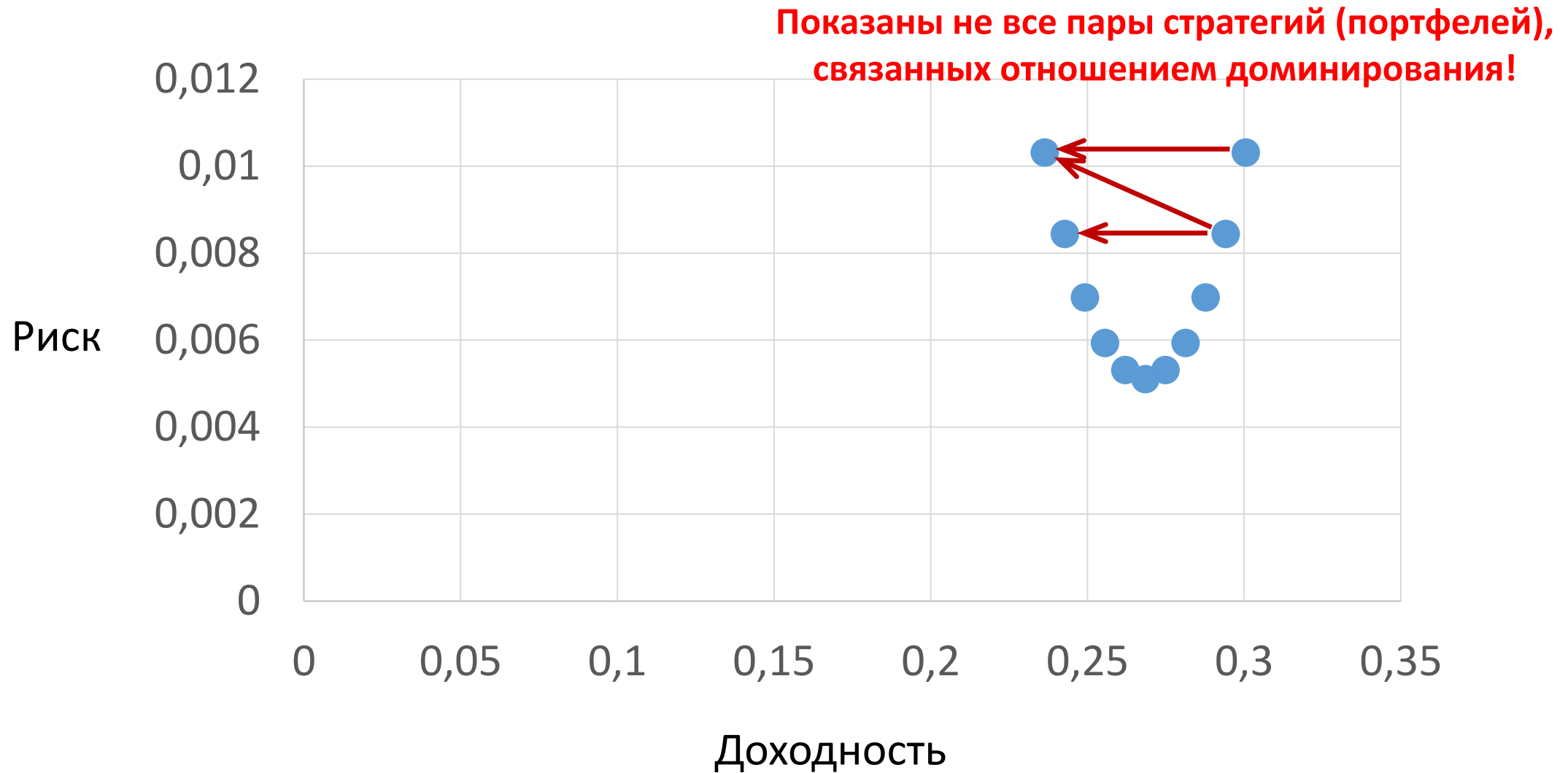

$$\alpha^T C \alpha$$

См. также портфельная теория Марковица (предложена в 50-х, удостоена Нобелевской премии в 1990).

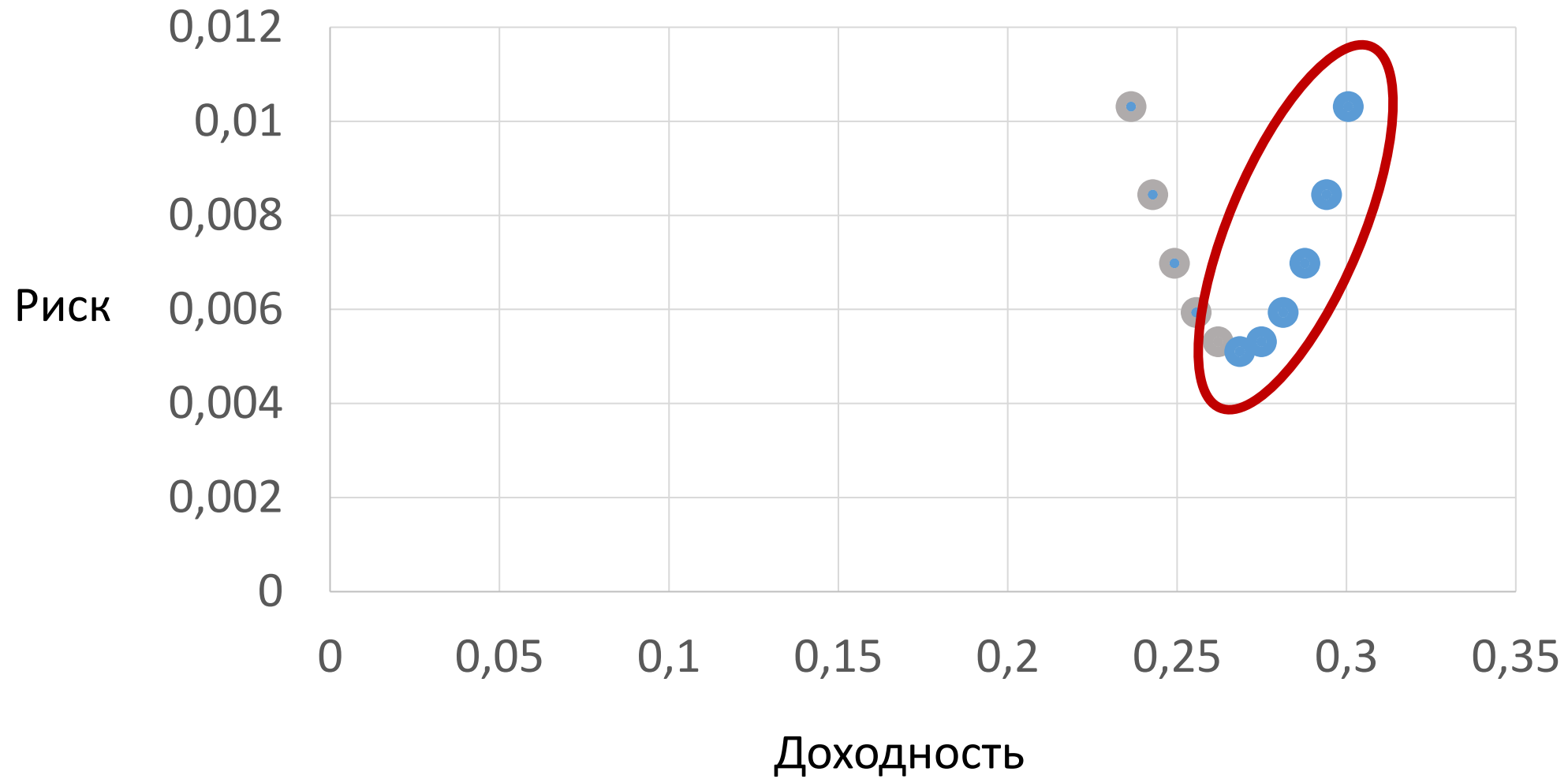
Пространство портфелей и множество Парето



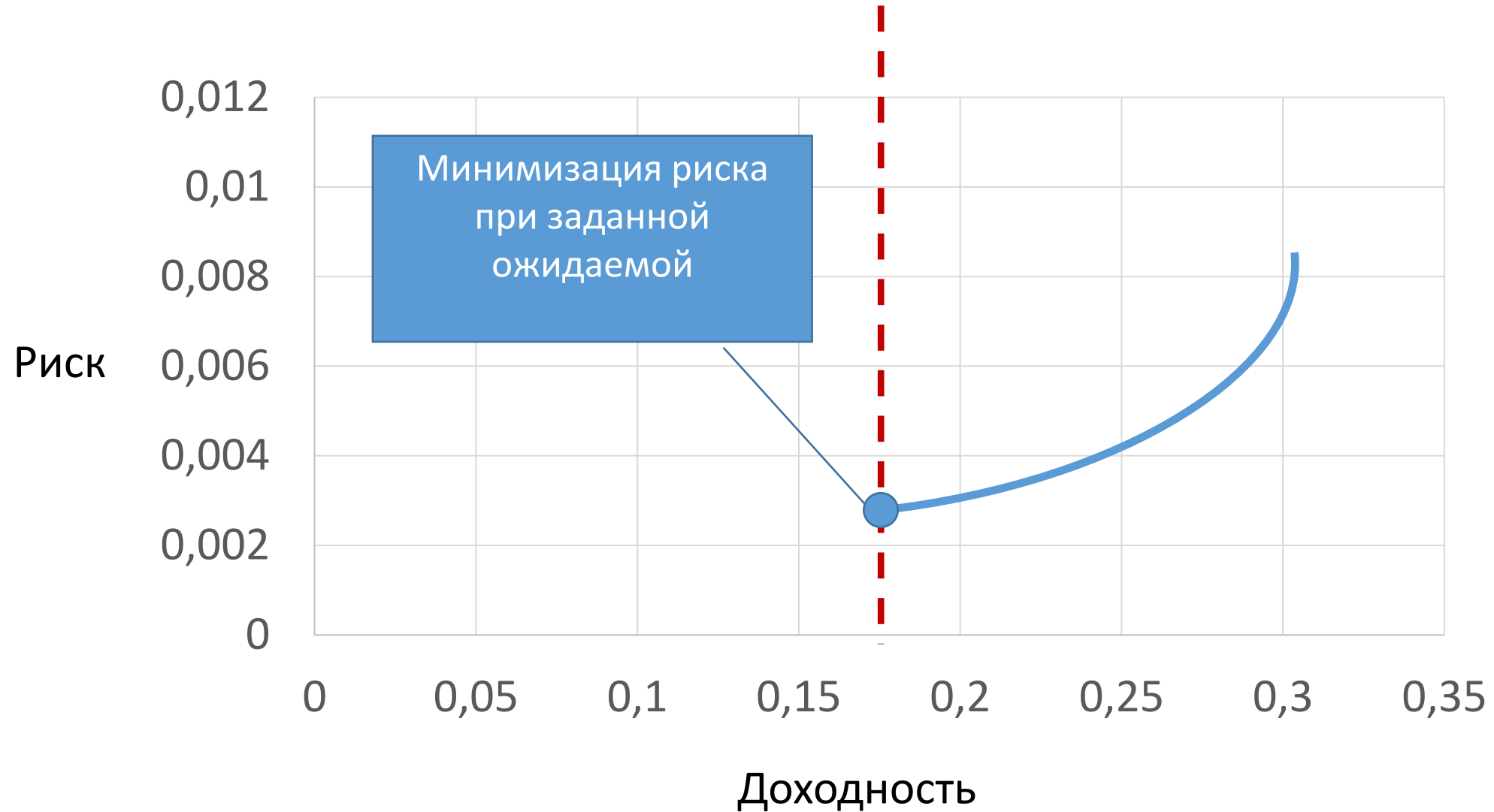
Пространство портфелей и множество Парето



Пространство портфелей и множество Парето



Пространство портфелей и множество Парето



Как это сделать в Python

Получение данных с сайта Финам

<https://www.finam.ru/profile/moex-akcii/pllc-yandex-n-v/export/>

МосБиржа акции

Yandex cIA

☆

Интервал и периодичность

01.01.2017

—

31.12.2017

1 день

Имя выходного файла

YNDX_170101_171231

.csv

Имя контракта

YNDX

Формат

даты

ггггммдд

времени

ччммсс

Выдавать время

☐ начала свечи

☒ окончания свечи

☒ московское

Разделитель

полей

запятая (,)

разрядов

нет

Формат записи в файл

TICKER, PER, DATE, TIME, CLOSE

Добавить заголовок файла

☒

Заполнять периоды без сделок

☐

Получить файл

Получение данных с сайта Финам.

Формат файла

```
<TICKER>,<PER>,<DATE>,<TIME>,<CLOSE>  
YNDX,D,20170103,000000,1241.0000000  
YNDX,D,20170104,000000,1239.0000000  
YNDX,D,20170105,000000,1252.0000000  
YNDX,D,20170106,000000,1291.0000000  
YNDX,D,20170109,000000,1317.0000000  
YNDX,D,20170110,000000,1319.5000000  
YNDX,D,20170111,000000,1299.0000000
```

Загрузка в Python

```
prices_df = pd.read_csv('YNDX_170101_171231.csv')
```

	<TICKER>	<PER>	<DATE>	<TIME>	<CLOSE>
0	YNDX	D	20170103	0	1241.0
1	YNDX	D	20170104	0	1239.0
2	YNDX	D	20170105	0	1252.0
3	YNDX	D	20170106	0	1291.0
4	YNDX	D	20170109	0	1317.0

Загрузка в Python

```
prices_df = pd.read_csv('YNDX_170101_171231.csv')
```

	<TICKER>	<PER>	<DATE>	<TIME>	<CLOSE>
0	YNDX	D	20170103	0	1241.0
1	YNDX	D	20170104	0	1239.0
2	YNDX	D	20170105	0	1252.0
3	YNDX	D	20170106	0	1291.0
4	YNDX	D	20170109	0	1317.0

Загрузка в Python

```
prices_df = pd.read_csv('YNDX_170101_171231.csv')  
prices = prices_df['<CLOSE>']
```

```
0      1241.0  
1      1239.0  
2      1252.0  
3      1291.0  
4      1317.0
```

Переход к относительным приращениям

```
prices_df = pd.read_csv('YNDX_170101_171231.csv')
prices = prices_df['<CLOSE>']
Dm = (df['<CLOSE>'].shift(-1)/df['<CLOSE>'] - 1).dropna()
```

D =

0	1241.0
1	1239.0
2	1252.0
3	1291.0
4	1317.0
...	

Dm =

0	-0.001612
1	0.010492
2	0.031150
3	0.020139
4	0.001898
...	

$$q_i = \frac{p_i}{p_{i-1}} - 1$$

Вычисление матрицы ковариаций

```
stocks = [  
    'portfolio_data\\AFLT_160501_170510.txt',  
    'portfolio_data\\TRNFP_160501_170510.txt',  
    'portfolio_data\\ROSN_160501_170510.txt'  
]  
  
variations = []  
for filename in stocks:  
    df = pd.read_csv(filename)  
    variations.append((df['<CLOSE>'].shift(-1)/df['<CLOSE>'] - 1)  
                      .dropna()  
                      .values  
                      .reshape(1, -1))  
variations = np.concatenate(variations, axis=0)  
C = np.cov(variations)
```


Решение задачи квадратичного программирования

Quadratic Programming

The function `qp` is an interface to `coneqp` for quadratic programs. It also provides the option of using the quadratic programming solver from MOSEK.

```
cvxopt.solvers.qp(P, q [ , G, h [ , A, b [ , solver [ , initvals ] ] ] )
```

Solves the pair of primal and dual convex quadratic programs

$$\begin{array}{ll}\text{minimize} & (1/2)x^T P x + q^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b\end{array}$$

Как это сделать в Octave

Загрузка в Octave

```
D = csvread("YNDX_170101_171231.csv");
```

```
D =  
0.0000e+000    0.0000e+000    0.0000e+000    0.0000e+000    0.0000e+000  
0.0000e+000    0.0000e+000    2.0170e+007    0.0000e+000    1.2410e+003  
0.0000e+000    0.0000e+000    2.0170e+007    0.0000e+000    1.2390e+003  
0.0000e+000    0.0000e+000    2.0170e+007    0.0000e+000    1.2520e+003  
...
```

Загрузка в Octave

```
D = csvread("YNDX_170101_171231.csv");  
D = D(2:end, 5);
```

```
D =  
1.2410e+003  
1.2390e+003  
1.2520e+003  
...
```

Переход к относительным приращениям

```
D = csvread("YNDX_170101_171231.csv");  
D = D(2:end, 5);  
Dm = diff(D) ./ D(1:end-1);
```

```
D =  
1.2410e+003  
1.2390e+003  
1.2520e+003  
...
```

```
Dm =  
-0.00161  
0.01049  
0.03115  
...
```

Вычисление ковариации

Dm1 = ...

Dm2 = ...

c = cov(Dm1, Dm2);

Вычисление ковариации

```
data = {"AFLT_160501_170510.txt",  
        "ALRS_160501_170510.txt",  
        ...};  
  
Dm = zeros(size(D,1)-2, size(data, 1));  
for i = 1:size(Dm,2)  
    D = csvread(data{i});  
    D = D(2:end, 5);  
    Dm(:, i) = diff(D) ./ D(1:end-1));  
endfor  
  
H = cov(Dm);
```

Решение задачи квадратичного программирования

```
-- [X, OBJ, INFO, LAMBDA] = qp (X0, H, Q, A, B)  
-- ...
```

Solve the quadratic program

$$\begin{aligned} & \min_x 0.5 x' H x + x' q \\ & \text{subject to} \\ & A x = b \\ & lb \leq x \leq ub \\ & A_{lb} \leq A_{in} x \leq A_{ub} \end{aligned}$$