

In [116...

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sympy import *
from scipy.optimize import fsolve
```

Ozon — одна из крупнейших и наиболее известных российских компаний в сфере электронной коммерции. Основанная в 1998 году, компания стала пионером интернет-торговли в России и получила неофициальное название "русский Amazon". Сегодня Ozon предлагает разнообразный ассортимент товаров и услуг, включая электронику, одежду, продукты питания, товары для дома, книги и многое другое.

Ключевая информация о компании:

1. **Дата основания:** 1998 год.
 2. **Штаб-квартира:** Москва, Россия.
 3. **Основной вид деятельности:** Электронная коммерция (e-commerce).
 4. **Биржевая информация:**
 - С 2020 года акции компании торгуются на NASDAQ под тикером **OZON**.
 - Также представлены на Московской бирже (MOEX).
 5. **Платформа Ozon:**
 - Является маркетплейсом, где свои товары продают как компания, так и сторонние продавцы.
 - Включает в себя B2C, B2B и C2C направления.
-

Деятельность:

Ozon работает по модели маркетплейса, предоставляя сторонним продавцам доступ к широкой аудитории. Компания также развивает инфраструктуру логистики и финансовых технологий, включая:

- **Логистику:**
 - Собственная сеть складов, пунктов выдачи заказов и партнёрских точек.
 - Экспресс-доставка в крупные города России.
 - **Финансовые сервисы:**
 - Финтех-направление, включая Ozon Fintech, которое предлагает кредитные продукты для покупателей и продавцов.
 - **Технологии:**
 - Инвестиции в автоматизацию, аналитические инструменты и IT-инфраструктуру.
-

Основные показатели:

- **GMV (объем продаж):** Один из ключевых показателей, отражающий рост популярности платформы.
 - **Количество активных покупателей:** Сотни миллионов заказов в год от миллионов пользователей.
 - **Ассортимент:** Более 90 миллионов товаров в разных категориях.
-

Стратегия и инновации:

- **Ускоренный рост маркетплейса.**
 - **Развитие Ozon Express** (доставка продуктов питания и товаров за 2 часа).
 - **Поддержка малого и среднего бизнеса:** Программы для продавцов, включая льготные условия размещения.
 - **Экологические инициативы:** Развитие упаковки из перерабатываемых материалов и снижение углеродного следа.
-

Конкуренты:

Основные конкуренты Ozon в России:

- Wildberries
 - Яндекс Маркет
 - AliExpress Russia
-

Значение компании:

Ozon является лидером цифровой трансформации в ритейле России, предлагая клиентам удобство, широкий выбор и качественный сервис. Благодаря активному росту и внедрению новых технологий, компания удерживает позиции одного из крупнейших игроков на рынке e-commerce в России и СНГ.

In [117...

```
def checker(date):
    if date.month in [1, 12]:
        return True
    return False

def get_year(date):
    return date.year

def to_float(x):
    x = float(x.replace('.', '').replace(',', '.'))
    return x

def to_value(x):
    x = float(x[:-1].replace('.', '').replace(',', '.'))*1000
    return x

def to_perc(x):
    return float(x[:-1].replace('.', '').replace(',', '.'))/100
```

```
In [118... data = pd.read_csv('Прошлые данные - OZONDR.csv', parse_dates=['Дата'], dayfirst
data = data.sort_values(['Дата'])
data.head()
```

```
Out[118...
      Дата  Цена  Откр.  Макс.  Мин.  Объём  Изм. %
997 2020-11-25  2.954,00  3.102,00  3.282,00  2.888,00  1,29M  -2,60%
996 2020-11-26  2.944,00  3.002,00  3.002,00  2.900,50  194,32K  -0,34%
995 2020-11-27  3.025,00  2.967,50  3.082,00  2.913,00  336,31K  2,75%
994 2020-11-30  3.071,00  2.999,50  3.140,00  2.967,50  210,26K  1,52%
993 2020-12-01  3.059,50  3.048,00  3.097,00  3.000,50  150,54K  -0,37%
```

```
In [119... data['Цена'] = data['Цена'].apply(to_float)
data['Откр.'] = data['Откр.'].apply(to_float)
data['Макс.'] = data['Макс.'].apply(to_float)
data['Мин.'].apply(to_float)
# data['Объём'] = data['Объём'].apply(to_value)
data['Изм. %'] = data['Изм. %'].apply(to_perc)
data['Дата'] = data['Дата']
```

```
In [120... fig, ax = plt.subplots(figsize=(20, 8))

ax.plot(data['Дата'], data['Мин.'])
ax.plot(data['Дата'], data['Макс.'], alpha=1)

# ax.plot(data['Дата'], data['Изм. %'])

plt.show()
```



```
In [121... data1 = data[data['Дата'].apply(checker)]
data1.loc[:, 'Year'] = data1.loc[:, 'Дата'].apply(get_year)
data1
```

C:\Users\ivant\AppData\Local\Temp\ipykernel_6288\909759873.py:2: SettingWithCopyWarning:
 A value is trying to be set on a copy of a slice from a DataFrame.
 Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

```
data1.loc[:, 'Year'] = data1.loc[:, 'Дата'].apply(get_year)
```

Out[121...

	Дата	Цена	Откр.	Макс.	Мин.	Объём	Изм. %	Year
993	2020-12-01	3059.5	3048.0	3097.0	3000.5	150,54K	-0.0037	2020
992	2020-12-02	3184.0	3030.0	3230.0	3003.0	264,44K	0.0407	2020
991	2020-12-03	3614.5	3120.5	3649.0	3085.5	914,18K	0.1352	2020
990	2020-12-04	3556.0	3566.5	3984.0	3428.5	1,10M	-0.0162	2020
989	2020-12-07	3422.0	3570.0	3675.0	3351.5	404,94K	-0.0377	2020
...
211	2024-01-31	2909.5	2920.0	2932.5	2881.5	357,71K	-0.0007	2024
3	2024-12-02	2871.5	2935.0	2993.5	2843.5	1,03M	-0.0213	2024
2	2024-12-03	2730.5	2869.0	2874.5	2713.0	1,71M	-0.0491	2024
1	2024-12-04	2674.5	2730.0	2786.5	2626.0	1,59M	-0.0205	2024
0	2024-12-05	2690.5	2676.0	2748.5	2661.0	721,61K	0.0060	2024

172 rows × 8 columns

In [122...

```
mn = data1.loc[data1.groupby('Year')['Дата'].idxmin()][['Дата', 'Цена']].sort_values('Дата')
mx = data1.loc[data1.groupby('Year')['Дата'].idxmax()][['Дата', 'Цена']].sort_values('Дата')

mn.loc[:, 'Year'] = mn.loc[:, 'Дата'].apply(get_year)
mx.loc[:, 'Year'] = mx.loc[:, 'Дата'].apply(get_year)

mn = mn[['Цена', 'Year']]
mx = mx[['Цена', 'Year']]

m = pd.merge(mn, mx, on='Year')

m1 = m[['Цена_x', 'Цена_y']]
m1['Year'] = m['Year']
m1.columns = ['Цена акций начало года', 'Цена акций конец года', 'Год']
m1 = m1.set_index('Год')
m1 = m1.drop(2024, axis=0)
m1
```

Out[122...

Цена акций начало года Цена акций конец года

Год		
2023	1438.0	2804.5
2022	2238.5	1402.0
2021	3229.5	2316.5
2020	3059.5	3289.5

In [123...

```
databook = pd.read_excel('Ozon_Databook_Q3_24.xlsx', 'Public Databook', header =
databook = databook.drop([i for i in databook.columns if 'Q' in i or '2018' in i])
databook.columns = ['Год']+ [i[3:] for i in databook.columns[1:]]

databook = databook.drop(list(range(0,4))+list(range(9,12))+list(range(15,18)))+[
display(databook['Год'].unique())
```

```
array(['GMV incl. services ',
      'Share of Marketplace, as % of GMV incl. services ',
      'Number of orders, millions', 'Number of active buyers, millions',
      'Frequency, orders', 'Total revenue ',
      '(Loss)/profit for the period', 'Adjusted EBITDA',
      'Total non-current assets', 'Total current assets',
      'Cash and cash equivalents', 'Total assets ', 'Total equity ',
      'Total non-current liabilities ', 'Total current liabilities ',
      'Total liabilities ', 'Total equity and liabilities ',
      'Movements in working capital1',
      'Net cash (used in) / generated from operating activities1',
      'Capital expenditures',
      'Net cash (used in)/ generated from investing activities',
      'Net cash(used in)/ generated from financing activities2'],
      dtype=object)
```

In [124...

```
databook = databook.set_index('Год')
databook = databook.T.iloc[::-1]
for i in databook.columns:
    databook[i] *= 1000000 if i not in ['Frequency, orders', f'Share of Marketpl
databook['T'] = [0.2]*databook.shape[0]

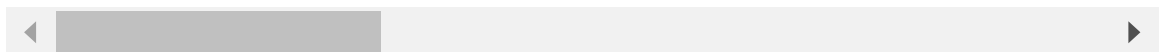
shares_q = [216414000, 203730000, 203730000, 203730000]
databook['Shares quantity'] = shares_q

databook
```

Out[124...

Год	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 24 columns



соединим таблички

In [125...

```

databook.index=databook.index.astype('int64')

m1 = m1.reset_index()
databook = databook.reset_index()

databook = pd.merge(databook,m1,'inner',left_on='index', right_on='Год')

databook = databook.drop('index',axis=1).set_index('Год')

```

1. CCF (Cash Conversion Flow):

- **Описание:** Поток денежных средств, обычно рассчитываемый как чистый денежный поток от операционной деятельности за вычетом капитальных затрат (CapEx).
- **Формула:**

$$CCF = \text{Net Cash from Operating Activities} - \text{Capital Expenditures}$$

Или же

$$CCF = \text{Shares quantity} \cdot \frac{\text{Цена акций конец года} + \text{Цена акций начало года}}{2}$$

- **Данные:**
 - "Net cash (used in)/generated from operating activities"
 - "Capital expenditures"



In [126...

```

databook['CCF 2'] = databook['Net cash (used in) / generated from operating acti

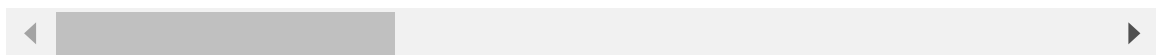
```

databook

Out[126...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 27 columns



2. μ (Рентабельность, доходность):

- **Описание:** μ может относиться к рентабельности продаж или маржинальной прибыли.
- **Формула для маржи EBITDA:**

$$\mu = \frac{\text{Adjusted EBITDA}}{\text{Total Revenue}}$$

- **Данные:**
 - "Adjusted EBITDA"
 - "Total revenue"

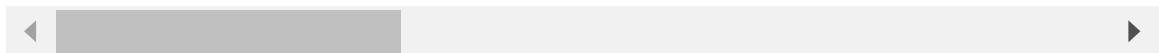
In [127...

```
databook['mu'] = databook['Adjusted EBITDA'] / databook['Total revenue ']  
databook
```

Out[127...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 28 columns



3. t (Ставка налогообложения):

- **Описание:** Средняя ставка налога, уплаченная компанией.
- **Формула:**

$$t = \frac{\text{Income Tax Expense}}{\text{Profit Before Tax}}$$

- **Данные:** Если эти строки отсутствуют, можно использовать стандартную ставку налога (например, для РФ — 20%).

4. k_d (Стоимость долга):

- **Описание:** Средневзвешенная стоимость долгового капитала.
- **Формула:**

$$k_d = \frac{\text{Interest Expense}}{\text{Total Debt}}$$

- **Данные:**
 - "Interest expense" можно найти в "PnL" или "Total liabilities".
 - "Total debt" = "Total non-current liabilities" + "Total current liabilities".

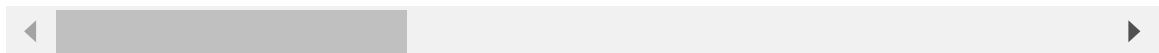
In [128...

```
databook['k_d'] = databook['Net cash(used in)/ generated from financing activiti'] / databook['Total Debt']
```


Out[128...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 29 columns



5. D (Долг):

- **Описание:** Общий долг компании.
- **Формула:**

$$D = \text{Total non-current liabilities} + \text{Total current liabilities}$$

- **Данные:**
 - "Total non-current liabilities"
 - "Total current liabilities"

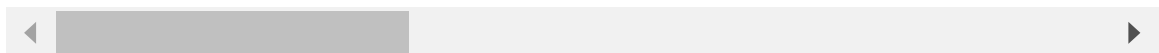
In [129...

```
databook['D'] = databook['Total liabilities ']  
databook
```

Out[129...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 30 columns



6. S (Акционерный капитал):

- **Описание:** Рыночная стоимость капитала (стоимость акций).
- **Формула:**

$$S = \text{Shares Outstanding} \times \text{Share Price}$$

- Если количество акций неизвестно, используйте данные "Total equity" для оценки.
- **Данные:**
 - "Цена акций начало года" и "Цена акций конец года"
 - "Total equity"

```
In [130... databook['S'] = databook['Shares quantity']*(databook['Цена акций начало года'] + databook
```

Out[130...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 31 columns

7. L :

- **Описание:** Текущая ливириджность компании.
- **Формула:**

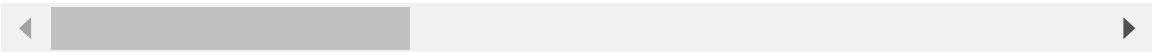
$$L = \frac{D}{S}$$

```
In [131... databook['L'] = databook['D']/databook['S'] databook
```

Out[131...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 32 columns



8. k_e :

- Формула:

$$k_e = \frac{\text{Цена в конце} - \text{Цена в начале}}{\text{Цена в начале}}$$

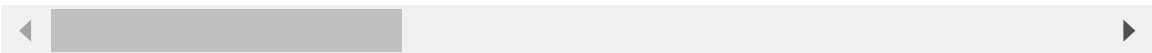
In [132...

```
databook['k_e'] = (databook['Цена акций конец года'] - databook['Цена акций нача  
databook
```

Out[132...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 33 columns



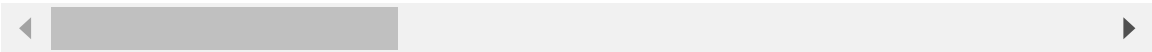
In [133...

```
databook['CCF 1'] = databook['Shares quantity']*(databook['Цена акций конец года  
databook
```

Out[133...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 34 columns



9. k_0 :

- Формула:

$$k_0 = \frac{k_e + L \cdot k_d \cdot (1 - T)}{1 + L \cdot (1 - T)}$$

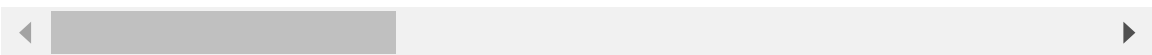
In [134...

```
databook['k_0'] = (databook['k_e'] + databook['L']*databook['k_d']*(1 - databook['T']))/databook['1 + L*(1 - T)']
```

Out[134...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 35 columns



10. w_d :

- Формула:

$$w_d = \frac{D}{D + S}$$

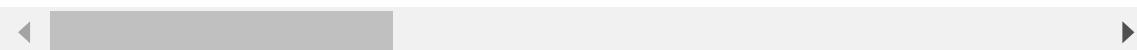
In [135...

```
databook['w_d'] = databook['D']/(databook['D'] + databook['S'])
databook
```

Out[135...

	GMV incl. services	Share of Marketplace, as % of GMV incl. services	Number of orders, millions	Number of active buyers, millions	Frequency, orders	Total revenue
Год						
2023	1.752277e+12	831300.797018	965667008.0	46089422.0	438.987612	4.242910e+11
2022	8.322403e+11	761522.301161	465387838.0	35169559.0	175.094141	2.771150e+11
2021	4.482600e+11	647764.690135	223300000.0	25600000.0	76.084732	1.782150e+11
2020	1.974139e+11	478128.156434	73872868.0	13760362.0	28.821077	1.043500e+11

4 rows × 36 columns



11. WACC:

- Формула:

$$\frac{1 - (1 + WACC)^{-n}}{WACC} = \frac{1 - (1 + k_0)^{-n}}{k_0 \cdot (1 - w_d[1 - (1 + k_d)^{-n}])}$$

In [136...

```
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator, MultipleLocator

def equation(wacc, n, f1):
    return (1 - (1 + wacc)**(-n)) / wacc - f1
time = 2019

W, k_0, k_d, w_d, n, T = symbols('W, k_0, k_d, w_d, n, T')

f = (1 - (1 + k_0)**(-n)) / (k_0 * (1 - w_d * T * (1 - (1 + k_d)**(-n))))

N_ = []
waccs0 = []
for i in databook.index[::-1]:
    n_ = i - time
    print(i, ': ', end='')
    if i:
        f1 = f.subs({n: n_, k_0: databook.loc[2023, 'k_0'], k_d: databook.loc[2023, 'k_d'], w_d: databook.loc[2023, 'w_d']})
        sols = [root.evalf() for root in solve((1 - (1 + W)**(-n_)) / W - f1) if root.is_real]
        waccs0.extend(sols)

    if sols == []:
        waccs0.append(0)
        print(0)
    else:
        print(*sols)
    N_.append(n_)
```

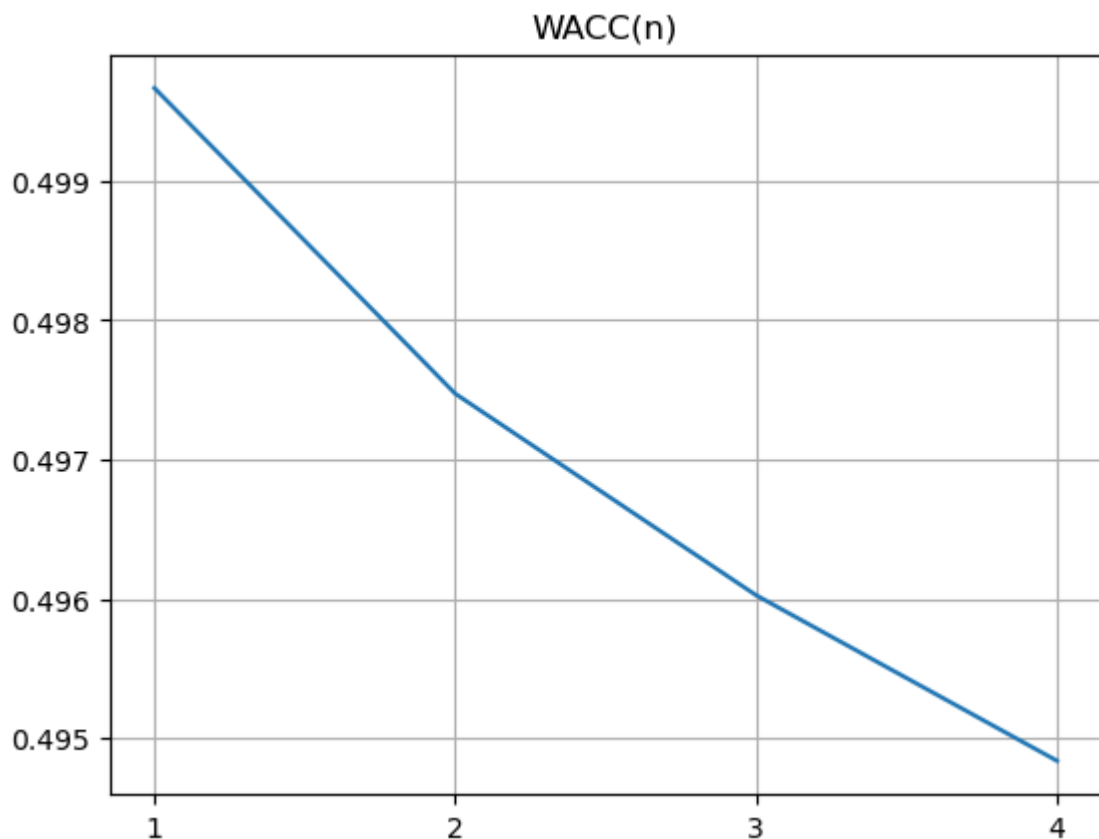
```
fig, ax = plt.subplots()
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.plot(N_,waccs0)
ax.grid()
plt.title('WACC(n)')
plt.show()
```

2020 :0.499665428458950

2021 :0.497472271130620

2022 :0.496025963667137

2023 :0.494838048106725



In [137... databook['WACC'] = waccs0

```
In [138... time = 2019

W, k_0, k_d, w_d, n, T = symbols('W, k_0, k_d, w_d, n, T')

f = (1-(1+k_0)**(-n))/(k_0*(1-w_d*T * (1-(1+k_d)**(-n))))

N_ = []
waccs5 = []
for i in list(databook.index[::-1]) + list(range(databook.index[::-1][-1]+1,data
n_=i - time
print(i,':',end='')
    '''if i in list(databook.index[::-1]):
        f1 = f.subs({n:n_, k_0:databook.loc[2023,'k_0'], k_d:databook.loc[:, 'k_d'
sols = [root.evalf() for root in solve((1-(1+W)**(-n_))/W-f1) if root.is
waccs.extend(sols)
    else:'''
    if i:
        f1 = f.subs({n:n_, k_0:databook.loc[2023,'k_0'], k_d:databook.loc[2023,'
```

```

sols = [root.evalf() for root in solve((1-(1+W)**(-n_))/W-f1) if root.is
waccs5.extend(sols)

if sols==[]:
    waccs5.append(0)
    print(0)
else:
    print(*sols)
N_.append(n_)

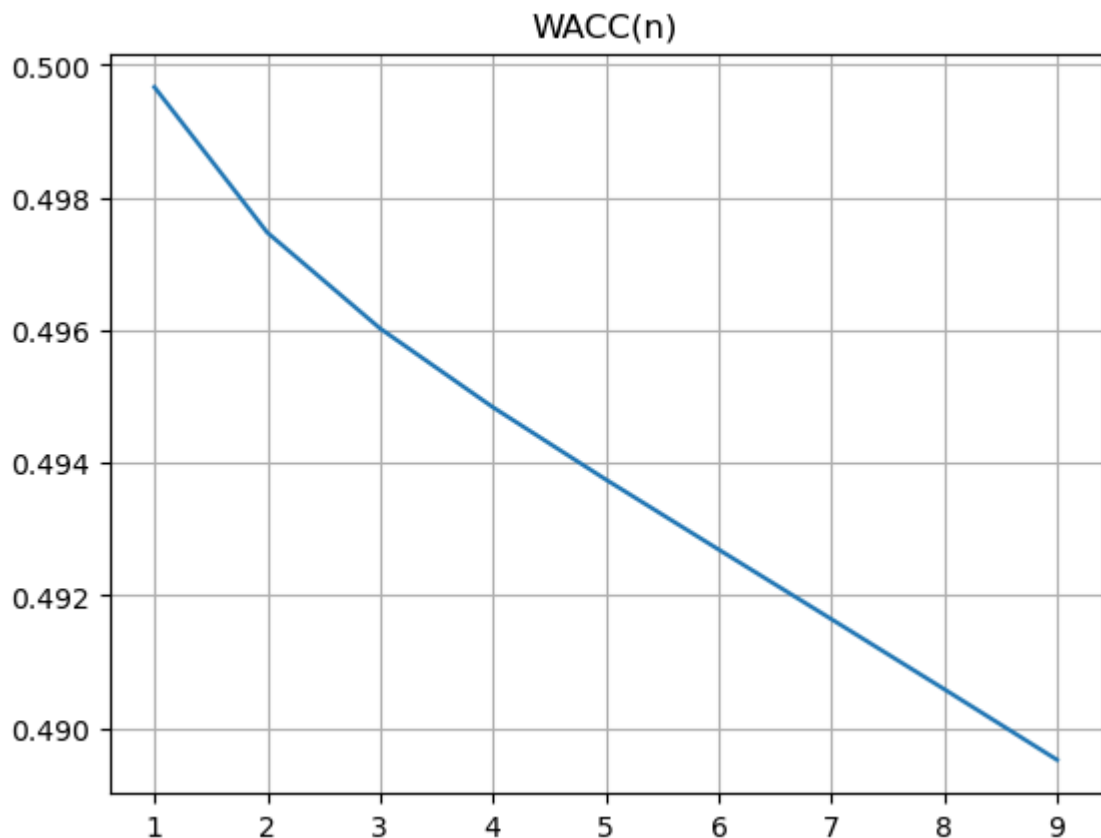
fig, ax = plt.subplots()
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.plot(N_,waccs5)
ax.grid()
plt.title('WACC(n)')
plt.show()

```

```

2020 :0.499665428458950
2021 :0.497472271130620
2022 :0.496025963667137
2023 :0.494838048106725
2024 :0.493748682330397
2025 :0.492693089615003
2026 :0.491642992828668
2027 :0.490586813589494
2028 :0.489521460327363

```



In [139...

```

time = 2019

W, k_0, k_d, w_d, n, T = symbols('W, k_0, k_d, w_d, n, T')

f = (1-(1+k_0)**(-n))/(k_0*(1-w_d*T * (1-(1+k_d)**(-n))))

N_ = []
waccs10 = []

```

```

for i in list(databook.index[::-1]) + list(range(databook.index[::-1][-1]+1,data
n_=i - time
print(i,':',end='')
'''if i in list(databook.index[::-1]):
    f1 = f.subs({n:n_, k_0:databook.loc[2023,'k_0'], k_d:databook.loc[:, 'k_d
sols = [root.evalf() for root in solve((1-(1+W)**(-n_))/W-f1) if root.is
waccs.extend(sols)
else:'''
if i:
    f1 = f.subs({n:n_, k_0:databook.loc[2023,'k_0'], k_d:databook.loc[2023,'
sols = [root.evalf() for root in solve((1-(1+W)**(-n_))/W-f1) if root.is
waccs10.extend(sols)

if sols==[]:
    waccs10.append(0)
    print(0)
else:
    print(*sols)
N_.append(n_)

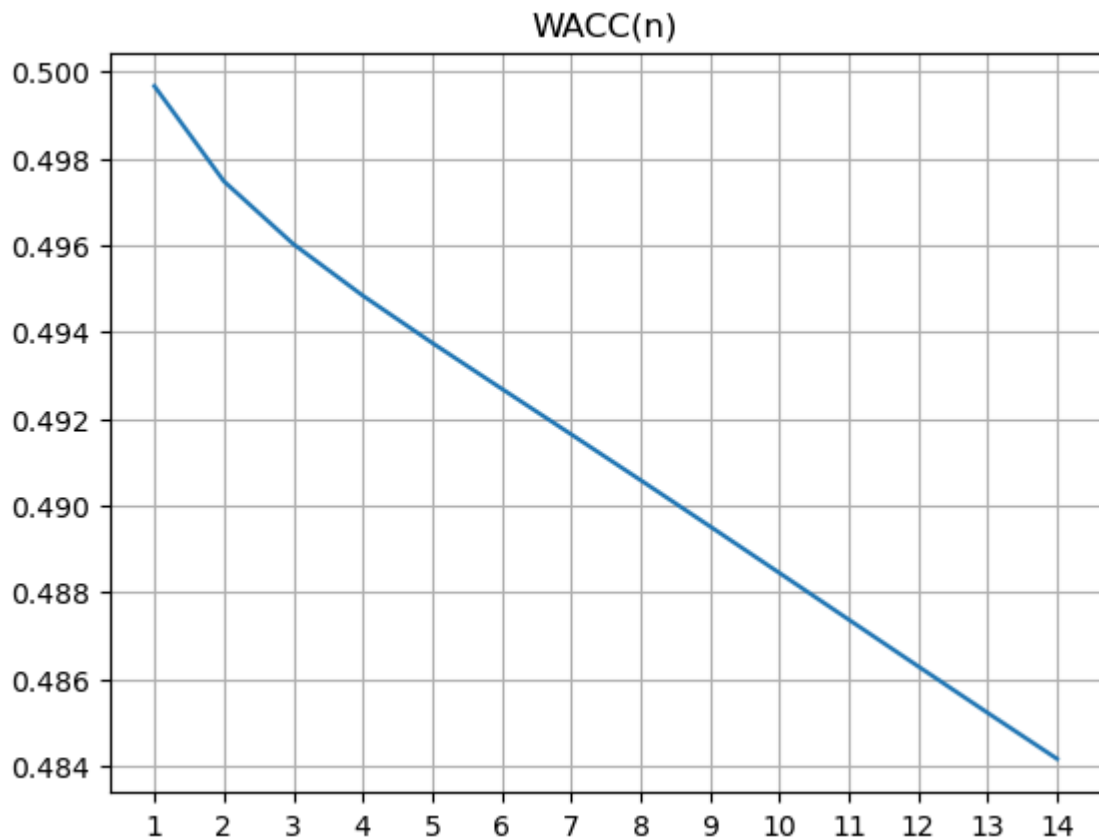
fig, ax = plt.subplots()
ax.xaxis.set_major_locator(MultipleLocator(1))
ax.plot(N_,waccs10)
ax.grid()
plt.title('WACC(n)')
plt.show()

```

```

2020 :0.499665428458950
2021 :0.497472271130620
2022 :0.496025963667137
2023 :0.494838048106725
2024 :0.493748682330397
2025 :0.492693089615003
2026 :0.491642992828668
2027 :0.490586813589494
2028 :0.489521460327363
2029 :0.488448339973515
2030 :0.487371171475608
2031 :0.486294688000237
2032 :0.485223840849869
2033 :0.484163318773233

```

```
In [140... c = dict()
for i in databook.index:
    c[i] = databook['CCF 1'][i]
    if i!=databook.index[0]:

        for n in range(databook.index[0]-i):
            c[i] *= (1+databook['WACC'][i+n+1])

print(sum(c.values()))
```

5561027715438.70

```
In [141... c = dict()
for i in databook.index:
    c[i] = databook['CCF 2'][i]
    if i!=databook.index[0]:

        for n in range(databook.index[0]-i):
            c[i] *= (1+databook['WACC'][i+n+1])

print(sum(c.values()))
```

196741236374.203

```
In [142... def rrstr(x,n): # округление до n знаков после запятой
    fmt = '{:.'+str(n)+'f}'
    return fmt.format(x).replace('.',',')

print(*np.vectorize(rrstr)(waccs0,5),sep='\n')
```

0,49967
0,49747
0,49603
0,49484

In [143...

```
databook.loc[list(databook.index[::-1])].T
```

Out[143...

Год	2020	2021	2022	
GMV incl. services	197413948486.131012	448260000000.0	832240336815.215698	17522760
Share of Marketplace, as % of GMV incl. services	478128.156434	647764.690135	761522.301161	8
Number of orders, millions	73872868.0	223300000.0	465387838.0	
Number of active buyers, millions	13760362.0	25600000.0	35169559.0	
Frequency, orders	28.821077	76.084732	175.094141	
Total revenue	104350000000.0	178215000000.0	277115000000.0	4
(Loss)/profit for the period	-22264000000.0	-56779000000.0	-58187000000.0	-4
Adjusted EBITDA	-11716000000.0	-41156000000.0	-3215000000.0	
Total non-current assets	29800000000.0	74752000000.0	129918000000.0	19
Total current assets	124808000000.0	166586000000.0	154903000000.0	27
Cash and cash equivalents	103702000000.0	108037000000.0	90469000000.0	16
Total assets	154608000000.0	241338000000.0	284821000000.0	47
Total equity	79257000000.0	30608000000.0	-16647000000.0	-6
Total non-current liabilities	15140000000.0	86794000000.0	106594000000.0	20
Total current liabilities	60211000000.0	123936000000.0	194874000000.0	34
Total liabilities	75351000000.0	210730000000.0	301468000000.0	54
Total equity and liabilities	154608000000.0	241338000000.0	284821000000.0	47

Год	2020	2021	2022	
Movements in working capital1	20119000000.0	32183000000.0	1247000000.0	10
Net cash (used in) / generated from operating activities1	6570000000.0	-13626000000.0	-18753000000.0	8
Capital expenditures	-6840000000.0	-19341000000.0	-35813000000.0	-5
Net cash (used in)/ generated from investing activities	-6580000000.0	-35403000000.0	-16040000000.0	-5
Net cash(used in)/ generated from financing activities2	102567000000.0	53892000000.0	18394000000.0	5
T	0.2	0.2	0.2	
Shares quantity	203730000	203730000	203730000	
Цена акций начало года	3059.5	3229.5	2238.5	
Цена акций конец года	3289.5	2316.5	1402.0	
CCF 2	13410000000.0	5715000000.0	17060000000.0	1
mu	-0.112276	-0.230935	-0.011602	
k_d	1.36119	0.25574	0.061015	
D	75351000000.0	210730000000.0	301468000000.0	54
S	646740885000.0	564943290000.0	370839532500.0	41
L	0.116509	0.373011	0.812934	
k_e	0.075176	-0.282706	-0.373688	
CCF 1	520345756286.195618	1098021448090.88501	592100066691.333862	41
k_0	0.184821	-0.158957	-0.202386	
w_d	0.104351	0.271674	0.448408	
WACC	0.494838048106725	0.496025963667137	0.497472271130620	0.499

In [144... databook.loc[list(databook.index[::-1])).T.to_excel('Ozon_T.xlsx')

via CCF 1

```
In [145... from functools import reduce
c = dict()
w = set()
for i in range(len(waccs5[-5:]))[::-1]:
    c[i] = databook['CCF 1'][2023]
    for n in range(len(waccs5[-5:])-i):
        c[i] /= (1+waccs5[-5:][n])

    w.add((1+waccs5[-5:][i]))

v1 = sum(c.values())
w=list(w)[::-1]
v2 = databook['CCF 1'][2023] / reduce(lambda x,y : x*y,w ) / waccs5[-1]
v = v1+v2
display(v1,v2,v)
```

806006624350.137

126995543879.29

933002168229.427

```
In [146... c = dict()
w = set()
for i in range(len(waccs10[-10:]))[::-1]:
    c[i] = databook['CCF 1'][2023]
    for n in range(len(waccs10[-10:])-i):
        c[i] /= (1+waccs10[-10:][n])

    w.add((1+waccs10[-10:][i]))

v1 = sum(c.values())
w=list(w)[::-1]
v2 = databook['CCF 1'][2023] / reduce(lambda x,y : x*y,w ) / waccs10[-1]
v = v1+v2
display(v1,v2,v)
```

916038287339.876

17702584570.3661

933740871910.242

via CCF 2

```
In [147... from functools import reduce
c = dict()
w = set()
for i in range(len(waccs5[-5:]))[::-1]:
```

```

c[i] = databook['CCF 2'][2023]
for n in range(len(waccs5[-5:])-i):
    c[i] /= (1+waccs5[-5:][n])

w.add((1+waccs5[-5:][i]))

v1 = sum(c.values())
w=list(w)[::-1]
v2 = databook['CCF 2'][2023] / reduce(lambda x,y : x*y,w ) / waccs5[-1]
v = v1+v2
display(v1,v2,v)

```

198873219354.604

31334745760.094

230207965114.698

In [148...

```

c = dict()
w = set()
for i in range(len(waccs10[-10:]))[::-1]:
    c[i] = databook['CCF 2'][2023]
    for n in range(len(waccs10[-10:])-i):
        c[i] /= (1+waccs10[-10:][n])

    w.add((1+waccs10[-10:][i]))

v1 = sum(c.values())
w=list(w)[::-1]
v2 = databook['CCF 2'][2023] / reduce(lambda x,y : x*y,w ) / waccs10[-1]
v = v1+v2
display(v1,v2,v)

```

226022315141.941

4367916935.22914

230390232077.17