

Санкт-Петербургский Национальный Исследовательский Университет  
Информационных Технологий, Механики и Оптики

Мегафакультет трансляционных и информационных технологий

**Лабораторная работа №3**

**Вариант №5**

Выполнил(и:)

Жаров Александр Павлович

Проверил

Мусаев А.А.

Санкт-Петербург,

2022

## Задание 1

### Текст задания

Анатолий, Борис и Евгений зимой 2016 изучали российские акции, выявляли наиболее и наименее зависящие друг от друга компании. В начале 2017 на основе полученных данных они решают вложить в эти компании 10.000.000 следующим образом: Анатолий – поровну в 3 пары компании с наименьшими (близкими к 0) коэффициентами парной корреляции и растущим трендом (стоимость в конце периода больше, чем стоимость в начале периода).

Борис – поровну в 3 пары компаний с наибольшими положительными (близкими к 1) коэффициентами парной корреляции и растущим трендом (стоимость в конце периода больше, чем стоимость в начале периода).

Евгений – во все компании пропорционально их капитализации. Каждые [чет. вариант – 6 месяцев, нечет. вариант – 3 месяца] они продают все акции и на полученную сумму закупают их заново по изначальным условиям, но на основе информации о котировках акций за прошедший период.

Требуется изучить, как менялись коэффициенты парных корреляций среди рассматриваемых компаний с начала инвестирования и до конца 2019 с шагом [чет. вариант – 6 месяцев, нечет. вариант – 3 месяца].

Сделайте выводы по всем трем типам вложения (оцените методы, насколько было важно производить разбалансировку, как менялись пары наиболее/наименее зависящих друг от друга компаний, какой метод оказался лучше и почему).

Перечень исследуемых компаний: Газпром, Татнефть, Сбербанк, ВТБ, Алроса, Аэрофлот, РусГидро, Московская Биржа, НЛМК, Северсталь, Детский Мир, Полиметалл, Яндекс, АФК Система, Группа ЛСР, Ленэнерго, Лукойл, МТС, Новатэк и ПИК. Котировки акций студент получает/вводит любым удобным способом. Комиссия не учитывается, разбалансировка происходит моментально. При невыполнении условия для Анатолия и Бориса, сокращается количество пар. Обратите внимание, что акции должны быть целыми числами (т.е. остаток просто «лежит»).

### Решение

Для получения котировок по данным акциям мы будем использовать API для тоех(Московской биржи). Сперва создаем папку с основными функциями (Рисунок 1), которые будут получать необходимые данные для работы. Первая функция возвращает словарь с количеством акций для каждой компании. Вторая получает на вход промежуток времени и список компаний и возвращает два словаря – первый со всеми котировками акций за данный период, второй – с ценой акции в начале периода и в конце.

Во втором файле находится основной код. Основная функция – Count(), в ней мы создаем словари содержащие информацию о 3 людях из условия – Анатолий, Борис и Евгений. Так же тут присутствуют еще 3 массива, которые понадобятся при построении диаграммы. Сперва мы получаем стартовую информацию – получаем ответы от функций с запросами для первого отрезка времени и записываем в переменные. Тут мы обращаемся к функции CorCoef(), где мы считаем все коэффициенты корреляции для каждой ценной бумаги со всеми другими, а так же сортируем их внутри словаря. Следующая функция – Buy(). При ее вызове мы передаем пользователя, информацию о котировках и начальную/конечную цену бумаги. Далее мы определяем пользователя, если это Анатолий или Борис, то мы ищем 3 самые большие/маленькие коэффициенты у компаний и записываем соответствующие компании в отдельный список. Затем, если это не Евгений, то проходимся по списку полученных компаний, вычитаем из общей суммы цену акции в

начале периода и добавляем цену акции в конце периода. Возвращаем прибыль и список купленных акций. В случае Евгения, нам понадобится еще одна функция `CorCount()`, которая считает долю капитализации компании среди других, суммируя произведение количества бумаг на их стоимость и деля стоимость одной компании на общую. Получая результат выполнения данной функции мы умножаем всю сумму Евгения на долю компании и закупаем все компании по такому произведению – доле капитализации компании. Сумму возвращаем пользователю.

Вернемся к основной функции `Count()`. Мы получили и записали стартовую информацию, а так же прошли первый цикл покупки/продажи акций, теперь нужно выполнить схожие действия для каждого промежутка времени. Для этого создаем цикл, который выполняется 8 раз (с 2017 по 2019). В ней выполняем те же действия, что описаны выше, и записываем результат.

Строим график с помощью `matplotlib` и оцениваем результат (Рисунок 7).

```

1 import requests
2
3 #Стартовые данные
4 def Kapit(comp):
5     comp_kap = {}
6     for i in comp:
7         kap = requests.get('https://iss.moex.com/iss/engines/stock/markets/shares/boards/TOBR/securities/'+i+'_'+i+'.json?iss.meta=off&marketdata.columns=SECID,ISSUECAPITALIZATION&securities.columns=ISSUESIZE')
8         comp_kap.update({i: kap.json()["securities"]["data"]})
9     return comp_kap
10
11 def newDate(comp_date_s, date_f):
12     comp_date = {}
13     comp_info = {}
14     for i in comp:
15         kat = requests.get('https://iss.moex.com/iss/history/engines/stock/markets/shares/boards/TOBR/securities/'+i+'_'+i+'.json?from='+date_s+'&to='+date_f+'&history.columns=CLOSE&iss.meta=off')
16         comp_date.update({i: (kat.json()["history"]["data"])}))
17         for i in comp_date:
18             comp_info.update({i: (comp_date[i][0], comp_date[i][-1])})
19     return comp_date, comp_info
20
21 #Доп функции
22 def Date(year=2016, month=1):
23     data = str(year) + '-' + str(month)
24     return data

```

Рисунок 1 – Задание 1

```

1 import numpy as np
2 import module as m
3 import matplotlib.pyplot as plt
4
5 comp = ["GAZP", "TATN", "SBER", "VTBR", "ALRS", "AFLT", "HYDR", "MOEX",
6         "CHMF", "DSKY", "POLY", "YNDX", "AFKS", "LSRG", "LSNG", "LKON", "MTSS", "NVTK", "PIKK"]
7
8 def CopCount(date, prise):
9     cop = {}
10    for i in date:
11        k = date[i][0][0] * prise[i][0][0]
12        cop.update({i: k})
13    prop = {}
14    sum = 0
15    for i in cop:
16        sum += cop[i]
17    for i in cop:
18        prop.update({i: (cop[i]/sum)})
19    return prop
20
21 def CorCoef(date):
22     coef = {}
23     prise = {}
24    for i in date:
25        comp_prise = []
26        for j in range(len(date[i])):
27            comp_prise.append(date[i][j][0])
28        prise.update({i: comp_prise})
29    for i in prise:
30        cor = {}
31        comp_1 = np.array(prise[i])
32        for j in prise:
33            if i != j:
34                comp_2 = np.array(prise[j])
35                k = np.corrcoef(comp_1, comp_2)
36                cor.update({j: abs(k[0][1])})
37    cor_sorted = sorted(cor.items(), key=lambda x: x[1])

```

Рисунок 2 – Задание 1.

```

37 cor_sorted = sorted(cor.items(), key=lambda x: x[1])
38 cor = dict(cor_sorted)
39 coef.update({i: cor})
40 #print(coef)
41 return coef
42
43 def Buy(user, data, prise):
44     money = user["cash"]
45     person = user["name"]
46     company = []
47     profit = 0
48     purch = {}
49     #print(person)
50
51     if person == 'anatoly':
52         count = 0
53         for i in data:
54             count_2 = 0
55             if count != 3:
56                 for j in data[i]:
57                     if count_2 != 1:
58                         company.append(i)
59                         if prise[j][0][0] - prise[j][1][0] > 0:
60                             #print(data[i][j], i, j, prise[j][0][0] - prise[j][1][0])
61                             company.append(j)
62                         count_2 += 1
63                         #print(count_2)
64                     else:
65                         break
66                 count += 1
67             else:
68                 break
69         #print(company)
70     if person == 'boris':
71         count = 0
72         #print(data)
73         for i in data:

```

Рисунок 3 – Задание 1

```

73     for i in data:
74         count_2 = 0
75         if count != 3:
76             for j in data[i]:
77                 if count_2 != 1:
78                     company.append(i)
79                     if prise[j][0][0] - prise[j][1][0] > 0:
80                         company.append(j)
81                         count_2 += 1
82                 else:
83                     break
84             count += 1
85         else:
86             break
87     if person == 'evgeny':
88         n = CopCount(m.Kapit(comp), prise)
89
90     for i in data:
91         spend = money*n[i]
92         count = 0
93         while spend >= prise[i][0][0]:
94             count += 1
95             profit += prise[i][1][0]
96             spend -= prise[i][0][0]
97         purch.update({i: count})
98     print(profit, purch)
99     return profit, purch
100
101     if person != "evgeny":
102         for i in company:
103             spend = money / len(company)
104             count = 0
105             #print(i, prise[i][1][0], prise[i][0][0])
106             while spend >= prise[i][0][0]:
107                 count += 1
108                 profit += prise[i][1][0]
109                 spend -= prise[i][0][0]

```

Рисунок 4 – Задание 1

```

109         spend -= prise[i][0][0]
110         purch.update({i: count})
111     print(profit, purch)
112     return profit, purch
113
114
115 def Count():
116     anatoly = {'name': 'anatoly', 'cash': 10000000, 'info': {}}
117     boris = {'name': 'boris', 'cash': 10000000, 'info': {}}
118     evgeny = {'name': 'evgeny', 'cash': 10000000, 'info': {}}
119     anatoly_profit = []
120     boris_profit = []
121     evgeny_profit = []
122
123
124     #Получение стартовой информации
125     test_2 = m.newDate(comp, m.Data(2017, 2), m.Data(2017, 3))
126     start = CorCoef(test_2[0])
127     prise = test_2[1]
128
129     answ1 = Buy(anatoly, start, prise)
130     answ2 = Buy(boris, start, prise)
131     answ3 = Buy(evgeny, start, prise)
132     anatoly["cash"] = answ1[0]
133     anatoly_profit.append(answ1[0])
134     anatoly["info"].update(answ1[1])
135     boris["cash"] = answ2[0]
136     boris_profit.append(answ2[0])
137     boris.update(answ2[1])
138     evgeny["cash"] = answ3[0]
139     evgeny_profit.append(answ3[0])
140     evgeny.update(answ3[1])
141     print(anatoly_profit)
142
143     date = [1, 0]

```

Рисунок 5 – Задание 1

```

143     date = [1, 0]
144     for i in range(8):
145         info = m.newDate(comp, m.Data(2017 + date[0], date[1]), m.Data(2017 + date[0], date[1] + 3))
146         if date[1] >= 9:
147             date[1] += -12
148             date[0] += 1
149             date[1] += 3
150         info_coef = CorCoef(info[0])
151         answ_1 = Buy(anatoly, info_coef, prise)
152         answ_2 = Buy(boris, info_coef, prise)
153         answ_3 = Buy(evgeny, info_coef, prise)
154         anatoly["cash"] = answ_1[0]
155         anatoly_profit.append(answ_1[0])
156         anatoly["info"].update(answ_1[1])
157         boris["cash"] = answ_2[0]
158         boris_profit.append(answ_2[0])
159         boris.update(answ_2[1])
160         evgeny["cash"] = answ_3[0]
161         evgeny_profit.append(answ_3[0])
162         evgeny.update(answ_3[1])
163         print(anatoly)
164     return anatoly_profit, boris_profit, evgeny_profit
165     stat = Count()
166
167     graph1 = plt.plot(stat[0])
168     graph2 = plt.plot(stat[1])
169     graph3 = plt.plot(stat[2])
170     plt.show()

```

Рисунок 6 – Задание 1

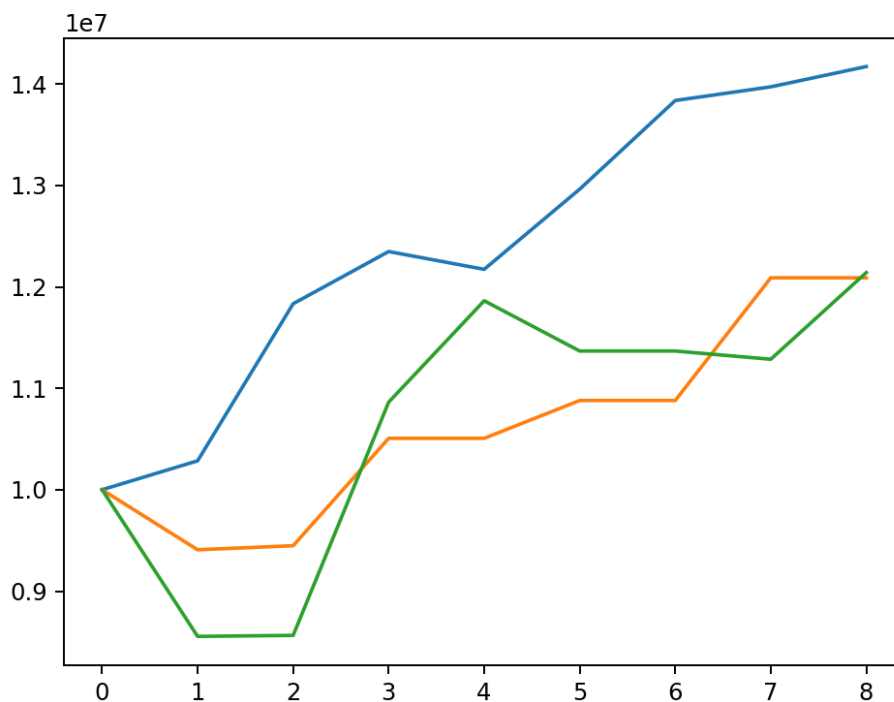


Рисунок 7 - Результат



### Вывод

Мы видим, что каждый из инвесторов остался в плюсе, но их успех объясним (Рисунок 7). Анатолий – синий график – покупал акции, которые в наименьшей степени зависели друг от друга, т.е их коэффициенты корреляции стремились к 0. Таким образом он уменьшил свой риск на потерю денег, ведь когда одна ценная бумага падала в цене, другая поднималась. Борис – оранжевый график – рисковал намного больше Анатолия, ведь он выбирал акции наиболее зависящие друг от друга, т.е если бы одна акция упала в цене, то наиболее вероятно, что и остальные последовали бы за ней. Однако Борису повезло и он не потерял свои деньги. Евгений же – зеленый график – распределил свои инвестиции наиболее равномерно, купив при этом все акции, которые были предоставлены. Таким образом он распределил риски и так же как и остальные не потерял свои деньги.

### Ссылка на гит

1. <https://github.com/SashaZharov/programming-lab3>