

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

Кафедра Систем Управления и Информатики

Лабораторная работа №3
Вариант №2

Выполнил
Тарабанов Алексей Вячеславович
Проверил
Мусаев А.А.

Санкт-Петербург,
2022

<https://github.com/AlekseyTarabanov/Programming>

Задание 1

Описание 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 покупают 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»).

Листинг (Решение):

Изначально программа перебирает данные с помощью `моих api`, это можно увидеть на рисунке 1. Так же эти данные фильтруются, например, не учитываются стоимость акции за период, если за этот период не хватает данных.

```
def getReq(name):
    m = 9
    y = 2016
    i = 1
    prices = {}
    while y < 2020:
        if (m + 1) // 10 >= 1:
            old_m = str(m+1)
        else:
            old_m = f"0{m+1}"
        new_y, m = formData(y,m,2)
        if new_y >= 2020:
            break
        if (m + 1) // 10 >= 1:
            new_m = str(m+1)
        else:
            new_m = f"0{m+1}"
        req = f"https://iss.moex.com/iss/history/engines/stock/markets/shares/boards/TQBR/-
securities/{name}.json?from={y}-{old_m}-01&till={new_y}-
{new_m}-01&history.columns=TRADEDATE,OPEN&iss.meta=off"
        bigdata = requests.get(req).json()["history"]["data"]
        if requests.get(req).json()["history"]["data"] == []:
            prices[str(i)] = []
        else:
            prices[str(i)] = []
            for j in range(len(bigdata)):
                if bigdata[j][1] is not None:
                    prices[str(i)].append(bigdata[j][1])
            y,m = formData(new_y,m,1)
            i += 1
    return prices

def getData():
    alldata = requests.get("https://iss.moex.com/iss/engines/stock/markets/shares/boards/TQBR/-
securities.json?-
iss.meta=off&marketdata.columns=SECID,ISSUECAPITALIZATION&securities.columns=SECID,ISSUESIZE").json()
    ["securities"]["data"]
    for i in range(len(alldata)):
        if ID.count(alldata[i][0]) > 0:
            data[alldata[i][0]] = alldata[i][1]

def formData(oldy,oldm,step):
    m = oldm + step
    y = oldy + (m // 12)
    m = m % 12
    return y,m

def enumeration():
    pairs = []
    for i in range(len(ID)):
        for j in range(i+1,len(ID)):
            pairs.append([ID[i],ID[j]])
    return pairs
```

Рисунок 1 - Функции формирования запросов на получение данных за период.

После этого за каждый период высчитываются коэффициенты корреляции для всех пар и сортируются, как изображено на рисунке 2.

```

def regcoef(array1, array2):
    middlex = 0
    middley = 0
    fluctsumxy = 0
    fluctsumx = 0
    fluctsumy = 0
    for i in array1:
        middlex += i
    for i in array2:
        middley += i
    middlex = middlex / len(array1)
    middley = middley / len(array2)
    for i in range(len(array1)):
        fluctsumxy += (array1[i] - middlex) * (array2[i] - middley)
        fluctsumx += (array1[i] - middlex) ** 2
        fluctsumy += (array2[i] - middley) ** 2
    rxy = fluctsumxy / ((fluctsumx * fluctsumy) ** (1 / 2))
    return rxy

```

Рисунок 2 -Функция подсчёта коэффициента корреляции

После этого программа выбирает нужные пары, исходя из значений их коэффициентов. Далее моделируются три способа закупки в начале месяца и продажа в конце месяца, это можно увидеть на рисунке 3.

```

def borisTolya(sum,pair,step):
    comp1,comp2 = pair.split(":")
    total1 = sum // (prices1[comp1][str(step)][0] * 6)
    total2 = sum // (prices1[comp2][str(step)][0] * 6)
    return total1 * prices1[comp1][str(step)][-1] + total2 * prices1[comp2][str(step)][-1]
ID = ["GAZP", "TATN", "SBER", "VTBR", "ALRS",
      "AFLT", "HYDR", "MOEX", "NLMK", "CHMF",
      "DSKY", "RUSP", "YNDX", "AFKS", "LSRG",
      "LSNG", "LKOH", "MTSS", "NVTK", "PIKK"]
pairs = enumeration()
data = {}
prices1 = {}
borya = 10000000
tolya = 10000000
andrey = 10000000
getData()
list3 = []
list4 = []
list5 = []
kl = []
for i in ID:
    prices1[i] = getReq(i)

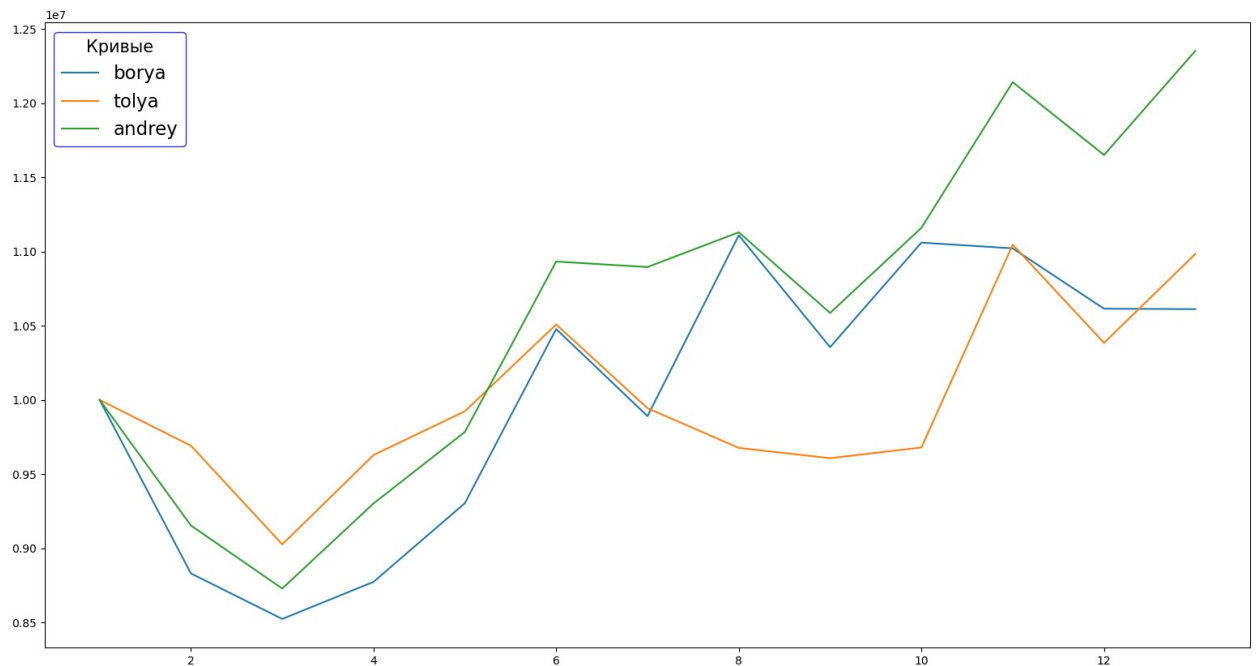
coeffs = {}
for i in range(1,14):
    borya_new = 0
    tolya_new = 0
    new_andrey = 0
    capitalization = 0
    z = 0
    q = 0
    if i != 1:
        while q < 3 and z <= len(f1):
            if prices1[f1[z].split(":")[0]][str(i)] != [] and prices1[f1[z].split(":")[1]][str(i)] !=
[]:
                borya_new += borisTolya(borya,f1[z],i)
                q += 1
                print(f"Боря {f1[z]}, {coeffs[f1[z]]}")
                z+=1
            borya = borya_new
            z = 0
            q = 0
            while q < 3 and z <= len(f2):
                if prices1[f2[z].split(":")[0]][str(i)] != [] and prices1[f2[z].split(":")[1]][str(i)] !
= []:
                    tolya_new += borisTolya(tolya, f2[z], i)
                    q += 1
                    print(f"Толя {f2[z]}, {coeffs[f2[z]]}")
                    z += 1
                tolya = tolya_new
            for c in ID:
                if prices1[c][str(i)] != []:
                    capitalization += data[c] * prices1[c][str(i)][0]
            for c in ID:
                if prices1[c][str(i)] != []:
                    percent = (data[c] * prices1[c][str(i)][0]) / capitalization
                    summ = andrey * percent
                    count = summ // prices1[c][str(i)][0]
                    new_andrey += count * prices1[c][str(i)][-1]
                andrey = new_andrey
            for j in pairs:
                list1 = prices1[j[0]][str(i)]
                list2 = prices1[j[1]][str(i)]
                if abs(len(list1) - len(list2)) > 10:
                    continue
                else:
                    list1 = list1[:min(len(list1),len(list2))]
                    list2 = list2[:len(list1)]
                    if len(list1) > 0 and len(list2) > 0:
                        coeffs[f"{j[0]}:{j[1]}"] = regcoef(list1,list2)
            f1 = sorted(coeffs, key=coeffs.get, reverse=True)
            f2 = sorted(coeffs,key=lambda dict_key: abs(coeffs[dict_key]))
            for v in f1:
                if prices1[v.split(":")[0]][str(i)][0] > prices1[v.split(":")[0]][str(i)][-1]:
                    f1.pop(f1.index(v))
            for v in f2:
                if prices1[v.split(":")[0]][str(i)][0] > prices1[v.split(":")[0]][str(i)][-1]:
                    f2.pop(f2.index(v))

            list3.append(int(borya))
            list4.append(int(tolya))
            list5.append(int(andrey))
            kl.append(i)

fig = plt.plot(kl,list3,label="borya")
plt.plot(kl,list4,label="tolya")
plt.plot(kl,list5,label="andrey")
plt.legend(fontsize=16.5,
           ncol=1,
           edgecolor='b',
           title='Кривые',
           title_fontsize='15')
plt.show()
print(list3,list4,list5,kl)

```

Результат выполнения программы:



```
Боря NLMK:CHMF, 0.9682209351317289
Боря TATN:NLMK, 0.9516456528790873
Боря GAZP:TATN, 0.936018342640538
Толя RUSP:MTSS, 0.008322332331055818
Толя MOEX:NVTK, -0.01883231938661006
Толя ALRS:AFLT, 0.032530850060718236
Боря GAZP:RUSP, 0.9156518913428604
Боря TATN:CHMF, 0.8629600473659687
Боря VTBR:LKOH, 0.8281073845100192
Толя LSNG:NVTK, -0.0030833775194064193
Толя AFLT:NVTK, -0.0040459792380336205
Толя LSRG:LSNG, -0.008623505402868423
```

Вывод:

Если посмотреть на графики то видно ,что наиболее эффективным и стабильным методом оказался способ Андрея, потому что по факту его портфель рос пропорционально российскому рынку. Поэтому можно сказать, что балансировка по капитализации в данном случае оказалась лучшей среди предложенных. Пары наиболее зависящих и независящих друг от друга компаний менялись часто, что видно на приложенном рисунке 9. По ходу выполнения лабораторной работы мы начали понимать, каким образом можно анализировать рынок биржи и поняли, что он находится в состоянии неконтролируемого хаоса (так как цена акции зависит от множества факторов), поэтому риски вложений в рынок высоки.