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

Кафедра Систем Управления и Информатики

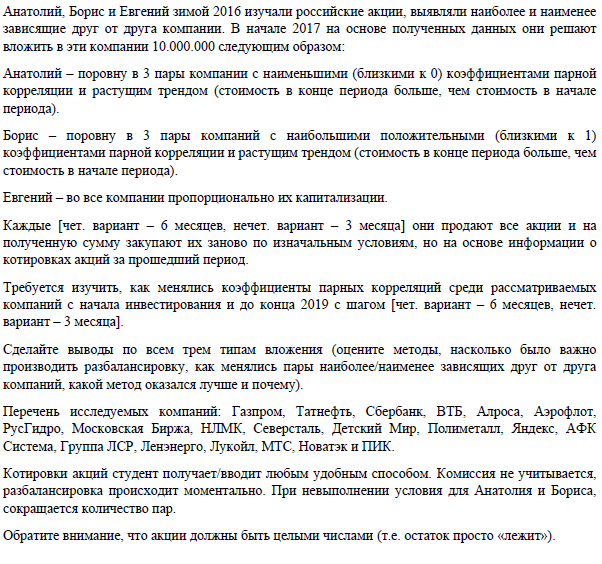
# Лабораторная работа №3

Выполнил(и:) Байков Иван

Проверил Мусаев А.А.

Санкт-Петербург, 2022

**Задание 1**



import yfinance as yf  
import matplotlib.pyplot as plt  
import numpy as np  
  
  
def is\_digit(string): # Проверка на число  
 if string.isdigit():  
 return True  
 else:  
 try:  
 float(string)  
 return True  
 except ValueError:  
 return False  
  
  
def deleteNan(array): # Удаляем nan в листах  
 fix\_array = []  
 for i in range(len(array)):  
 if array[i] > 0:  
 fix\_array.append(array[i])  
  
 return fix\_array  
  
  
def correlation(array, array2): # находим кэф коррелляции  
 list\_y = []  
 list\_x = []  
 sum\_x = 0  
 sum\_y = 0  
 ln1 = len(array)  
 ln2 = len(array2)  
 ln = min(ln1, ln2)  
 for i in range(ln):  
 if array[i] > 0:  
 if array2[i] > 0:  
 list\_y.append(array[i])  
 list\_x.append(array2[i])  
 else:  
 array2[i] = array2[i - 1]  
 else:  
 array[i] = array[i - 1]  
 if array2[i] > 0:  
 list\_y.append(array[i])  
 list\_x.append(array2[i])  
 else:  
 array2[i] = array2[i - 1]  
 list\_y.append(array[i])  
 list\_x.append(array2[i])  
 sum\_x += sum(list\_x)  
 sum\_y += sum(list\_y)  
 average\_x = sum\_x / (len(list\_x))  
 average\_y = sum\_y / (len(list\_y))  
 disp\_sum\_up = 0  
 disp\_sum\_down\_x = 0  
 disp\_sum\_down\_y = 0  
  
 for i in range(len(list\_x)):  
 if list\_y[i] != 0:  
 disp\_sum\_up += (list\_x[i] - average\_x) \* (list\_y[i] - average\_y)  
 disp\_sum\_down\_x += (list\_x[i] - average\_x) \*\* 2  
 disp\_sum\_down\_y += (list\_y[i] - average\_y) \*\* 2  
 r = disp\_sum\_up / ((disp\_sum\_down\_x \* disp\_sum\_down\_y) \*\* 0.5)  
  
 return r  
  
  
# ticker\_list = ['OGZPY', 'TATN.ME', 'SBER.ME'] # , 'VTBR.ME']  
ticker\_list = ['OGZPY', 'TATN.ME', 'SBER.ME', 'VTBR.ME', 'ALRS.ME', 'AFLT.ME', 'HYDR.ME',  
 'MOEX.ME', 'NLMK.ME', 'CHMF.ME', 'DSKY.ME', 'POLY.ME', 'YNDX', 'AFKS.ME',  
 'LSRG.ME', 'LSNG.ME', 'LKOH.ME', 'MTSS.ME', 'NVTK.ME', 'PIKK.ME']  
st = 60  
array = {}  
volume = {}  
anatoliy\_sum = 10000000  
boris\_sum = 10000000  
evgeniy\_sum = 10000000  
anatoliy = {}  
boris = {}  
evgeniy = {}  
  
for ticker in ticker\_list:  
 anatoliy[ticker] = 0  
 boris[ticker] = 0  
 evgeniy[ticker] = 0  
 data = yf.download(ticker, start="2017-01-01", end="2019-12-31", interval="1d")  
 volume[ticker] = data['Volume']  
 array[ticker] = data['Adj Close']  
 array[ticker] = deleteNan(array[ticker])  
  
for month in range(12): # Для всех лет  
  
 results1 = {}  
 results0 = {}  
 start = st \* month  
 end = start + 60  
  
 for ticker in ticker\_list:  
 anatoliy\_sum += anatoliy[ticker] \* array[ticker][end]  
 anatoliy[ticker] = 0  
 boris\_sum += boris[ticker] \* array[ticker][end]  
 boris[ticker] = 0  
 # Евгеха  
 evgeniy\_sum += evgeniy[ticker] \* array[ticker][end]  
 evgeniy[ticker] = 0  
 #print(evgeniy\_sum)

for i in range(0, len(ticker\_list) - 1): # Берем первый элемент списка тикеров  
 temp\_arr1 = []  
 for h in range(start, end):  
 temp\_arr1.append(array[ticker\_list[i]][h])  
 for j in range(i + 1, len(ticker\_list)): # Находим корр первого с каждым следующим  
 temp\_arr2 = []  
 for h in range(start, end):  
 temp\_arr2.append(array[ticker\_list[j]][h])  
 r = correlation(temp\_arr1, temp\_arr2)  
 results1[r] = (ticker\_list[i], ticker\_list[j])  
 results0[abs(r)] = (ticker\_list[i], ticker\_list[j])  
 #print(results1) # Посмотретьь котировки ( их 4560 )  
 # Сортировка r  
 sorted\_results1 = list(results1)  
 sorted\_results1 = sorted(sorted\_results1)  
 sorted\_results1.reverse()  
 sorted\_results0 = list(results0)  
 sorted\_results0 = sorted(sorted\_results0)  
  
 # Этап вложения  
 for i in range(3):  
 anatoliy[results1[sorted\_results1[i]][0]] += (anatoliy\_sum / 6) / array[results1[sorted\_results1[i]][0]][end]  
 anatoliy[results1[sorted\_results1[i]][1]] += (anatoliy\_sum / 6) / array[results1[sorted\_results1[i]][1]][end]  
 boris[results0[sorted\_results0[i]][0]] += (boris\_sum / 6) / array[results0[sorted\_results0[i]][0]][end]  
 boris[results0[sorted\_results0[i]][1]] += (boris\_sum / 6) / array[results0[sorted\_results0[i]][1]][end]  
 anatoliy\_sum = 0  
 boris\_sum = 0  
 # Евгений---------------------------------------------------  
 total\_value\_sum = 0  
 for ticker in ticker\_list:  
 total\_value\_sum += volume[ticker][end]  
 for ticker in ticker\_list:  
 evgeniy[ticker] = (evgeniy\_sum \* (volume[ticker][end] / total\_value\_sum)) \* 5.4 / (  
 array[ticker][end] \* array[results1[sorted\_results1[0]][0]][end])  
 evgeniy\_sum = 0  
  
# Убираем оставшиеся акции  
for ticker in ticker\_list:  
 anatoliy\_sum += anatoliy[ticker] \* array[ticker][-1]  
 anatoliy[ticker] = 0  
 boris\_sum += boris[ticker] \* array[ticker][-1]  
 boris[ticker] = 0  
 evgeniy\_sum += evgeniy[ticker] \* volume[ticker][-1]  
 evgeniy[ticker] = 0  
print(anatoliy\_sum, boris\_sum, evgeniy\_sum)

Сначала были получены котировки акций 20 крупных компаний РФ за 3 года. Для этого использовался модуль yfinance, который упрощает доступ к получению информации с Yahoo!Finance. Для восстановления некоторых пропавших значений используется функция deleteNan, для корреляции акций двух компаний – correlation. Для начала получаем скорректированные акции за каждый день торгов. Сразу же фильтруем данные. Далее идет основной цикл, в котором просчитываются коэффициенты корреляции всех 20 компаний попарно. Полученные коэффициенты заносятся в массивы и сортируются. Далее по установленным правилам участники вкладывают деньги. В начале следующего цикла они их выведут и вложат заново. По итогу получится баланс каждого участника.

**Результат выполнения**



**Вывод**

Самым выгодным оказался метод Анатолия, так как со временем цена акций стабильно растет, он вкладывал в компании, которые наиболее связаны друг с другом, поэтому рост одной из них обуславливает рост других. У Бориса получается диверсифицированный портфель, где компании практически не связаны с собой, поэтому если одна из них теряет в стоимости акций, то это не отражается на других. Метод Евгения оказался хуже, так как он продает акции растущих компаний и может вложить их в менее прибыльные**.**