Кобак Ф.А.
Домашнее задание по дисциплине
«Многомерный статистический анализ»

Задание состояло в автоматизации вычислительной процедуры рассмотренной на предыдущей лекции. Реализация была проведена с использование языка программирования python3 и представляет собой функцию:

```
def fin_num_of_steps(P, R, n, full_info = False):
# функция реализующую вычислительную процедуру из пункта 13.2
# вернет лист из табличек соотвесвующих шагам
# формирует аналогичиный решению на листике "вывод" при установлении
агрумента full_info в состояние True
# входные данные:
# P - массив матриц переходных вероятностей (list содержащий ряд np.matrix)
# R - массив матриц выйгрышей (list содержащий ряд np.matrix)
# n - число шагов для задачи

result = []

v = {};

strat_count = P[0].shape[0]

# на первом шаге значения функций как 0
f = np.zeros(strat_count)

for i,p in enumerate(P):
v["v_" +str(i+1)] = np.diag(np.dot(P[i], np.transpose(R[i])))


if full_info:
v_df = pd.DataFrame(v)
v_df.index = pd.RangeIndex(1,strat_count+1)
print()
print()
print('средние выйгрыши для i го состояния (по строкам)')
print('и для k-й стратерии (по сторокам)')
print(v_df)


# цикл переибрает периоды
for i in range(n,0,-1):
f_comp = {}

# цикл перебирает статегии
for j,p in enumerate(P):
f_comp['k='+str(j+1)] = np.array(np.dot(p,f) + v['v_' + str(j+1)])[0,:]

new_f = pd.DataFrame(f_comp).apply(max, axis = 1)
f_index = pd.DataFrame(f_comp).apply(lambda x: list(x).index(max(x)) + 1, axis = 1)
```

```
f_comp['f'] = new_f
f_comp['статерия'] = f_index
f_comp = pd.DataFrame(f_comp)
result.append(f_comp)
f = np.array(new_f)
if full_info:
print()
print()
print('шаг номер ' + str(i))
print(f_comp)
return result;
```

Можно использовать для любого числа матриц любой размерности и для любого числа шагов. В результате можно получить в Excel или просто в консоль таблицы подобные тем, что были приведены на занятии.
Пример решения задачи разобранной на занятии

```
# Example1
P1 = np.matrix([[0.2, 0.5, 0.3], [0, 0.5, 0.5],[0, 0, -1]])
P2 = np.matrix([[0.3, 0.6, 0.1], [0.1, 0.6, 0.3],[0.05, 0.4, 0.55]])

R1 = np.matrix([[7, 6, 3], [0, 5, 1], [0, 0, 1]])
R2 = np.matrix([[6, 5, -1], [7, 4, 0], [6, 3, -2]])

fin_num_of_steps([P1, P2], [R1, R2], 3, True)
```

В результате в консоли будет получено

```
средние выйгрыши для i го состояния (по строкам)
и для k-й стратерии (по сторокам)
   v_1  v_2
1  5.3  4.7
2  3.0  3.1
3 -1.0  0.4


шаг номер 3
   k=1  k=2    f  статерия
0  5.3  4.7  5.3         1
1  3.0  3.1  3.1         2
2 -1.0  0.4  0.4         2


шаг номер 2
    k=1    k=2      f  статерия
0  8.03  8.190  8.190         2
1  4.75  5.610  5.610         2
2 -1.40  2.125  2.125         2


шаг номер 1
       k=1       k=2         f  статерия
0  10.3805  10.73550  10.73550         2
1   6.8675   7.92250   7.92250         2
2  -3.1250   4.22225   4.22225         2
[dranik@fedora multivariate_statistical_analysis]$
```