



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

---

ФАКУЛЬТЕТ "Фундаментальные науки"

КАФЕДРА "Вычислительная математика и математическая физика" (ФН-11)

## ОТЧЕТ

к домашнему заданию №1 на тему

*Марковская цепь*

Дисциплина: *Теория случайных процессов*

Вариант: 7

Студент группы ФН11-61Б

\_\_\_\_\_  
(Подпись, дата)

А.Е. Каргополов  
(И.О. Фамилия)

Преподаватель, канд. физ.-мат.  
наук, доцент

\_\_\_\_\_  
(Подпись, дата)

Т.В. Облакова  
(И.О. Фамилия)

Оценка: \_\_\_\_\_

Москва, 2023 г.

## Задание

Дана однородная Марковская цепь с данным числом состояний  $\{S_1, S_2, \dots, S_m\}$ . Ненулевые переходные вероятности  $p_{ij}, i \neq j$ , заданы в таблице.

$$m = 6 \quad k = 25 \quad n = 130$$

$$p_{12} = 0.3 \quad p_{21} = 0.2 \quad p_{23} = 0.2 \quad p_{34} = 0.1$$

$$p_{45} = 0.3 \quad p_{56} = 0.2 \quad p_{61} = 0.4$$

1. Выпишите матрицу переходных вероятностей
2. Изобразите размеченный граф Марковской цепи
3. Докажите, что цепь эргодическая
4. Смоделируйте вектор начальных вероятностей  $p(0)$  согласно приложенному алгоритму
5. Вычислите безусловные вероятности состояний смоделированной цепи на  $k$  шаге по формуле из лекции 1 для полученного  $p(0)$
6. Смоделируйте  $n$  траекторий полученной цепи за  $k$  шагов (см алгоритм ниже). Несколько траекторий выведите на печать
7. По полученным реализациям траекторий найдите вектор эмпирических безусловных вероятностей состояний цепи на  $k$  шаге
8. Сравните найденные эмпирические вероятности с теоретическими для  $k$  шага
9. Вычислите финальные вероятности для рассматриваемой Марковской цепи (лекция 2) и сравните их с вероятностями состояний на  $k$  шаге
10. Сформулируйте выводы

## Решение

```
In [ ]: import numpy as np
import pandas as pd
import seaborn as sns
```

```
In [ ]: pd.set_option('display.max_columns', 100)
```

```
In [ ]: np.random.seed(42)
```

```
In [ ]: m = 6
k = 25
n = 130
```

### 1. Матрица переходных вероятностей

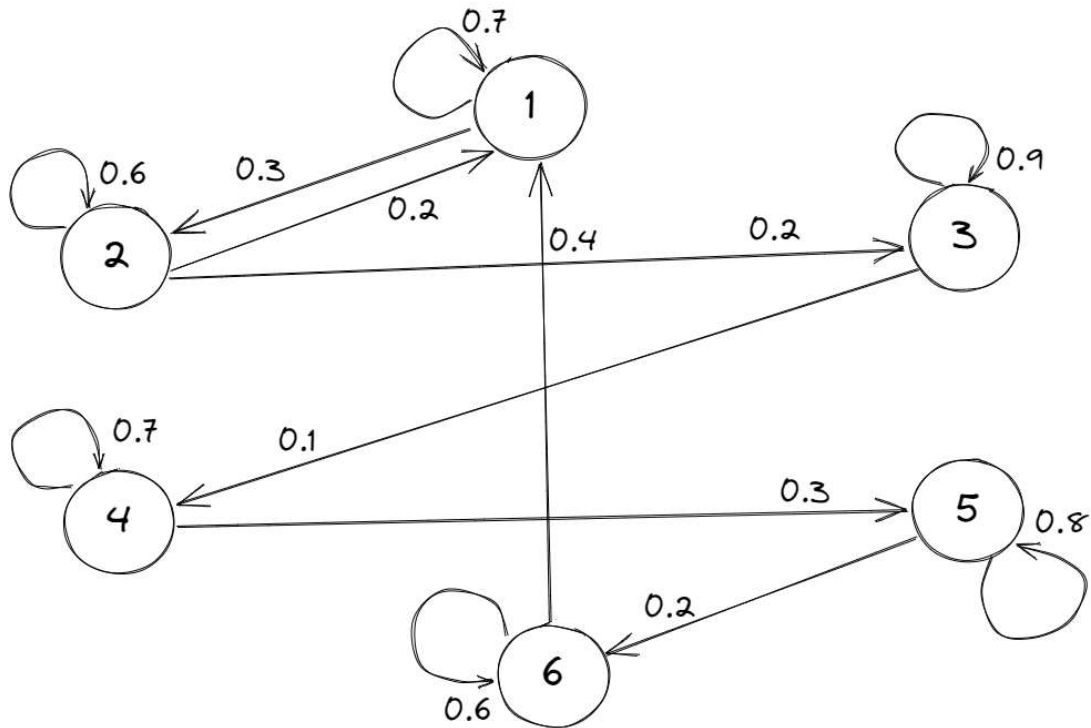
$$P = \begin{pmatrix} 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ 0.2 & 0.6 & 0.2 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 0 & 0.8 & 0.2 \\ 0.4 & 0 & 0 & 0 & 0 & 0.6 \end{pmatrix}$$

```
In [ ]: P = np.array([[0.7, 0.3, 0, 0, 0, 0],
                      [0.2, 0.6, 0.2, 0, 0, 0],
                      [0, 0, 0.9, 0.1, 0, 0],
                      [0, 0, 0, 0.7, 0.3, 0],
                      [0, 0, 0, 0, 0.8, 0.2],
                      [0.4, 0, 0, 0, 0, 0.6]])

print(P)
print(f'Размерность полученной матрицы: {P.shape}')
print(f'Проверим, является ли полученная матрица стохастической: {(P.sum(axis=1))

[[0.7 0.3 0.  0.  0.  0. ]
 [0.2 0.6 0.2 0.  0.  0. ]
 [0.  0.  0.9 0.1 0.  0. ]
 [0.  0.  0.  0.7 0.3 0. ]
 [0.  0.  0.  0.  0.8 0.2]
 [0.4 0.  0.  0.  0.  0.6]]
Размерность полученной матрицы: (6, 6)
Проверим, является ли полученная матрица стохастической: True
```

### 2. Размеченный граф Марковской цепи



### 3. Доказательство эргодичности

```
In [ ]: def count_zeros(pow : int) -> np.int32:
        return (np.linalg.matrix_power(P, pow).ravel() == 0).sum()

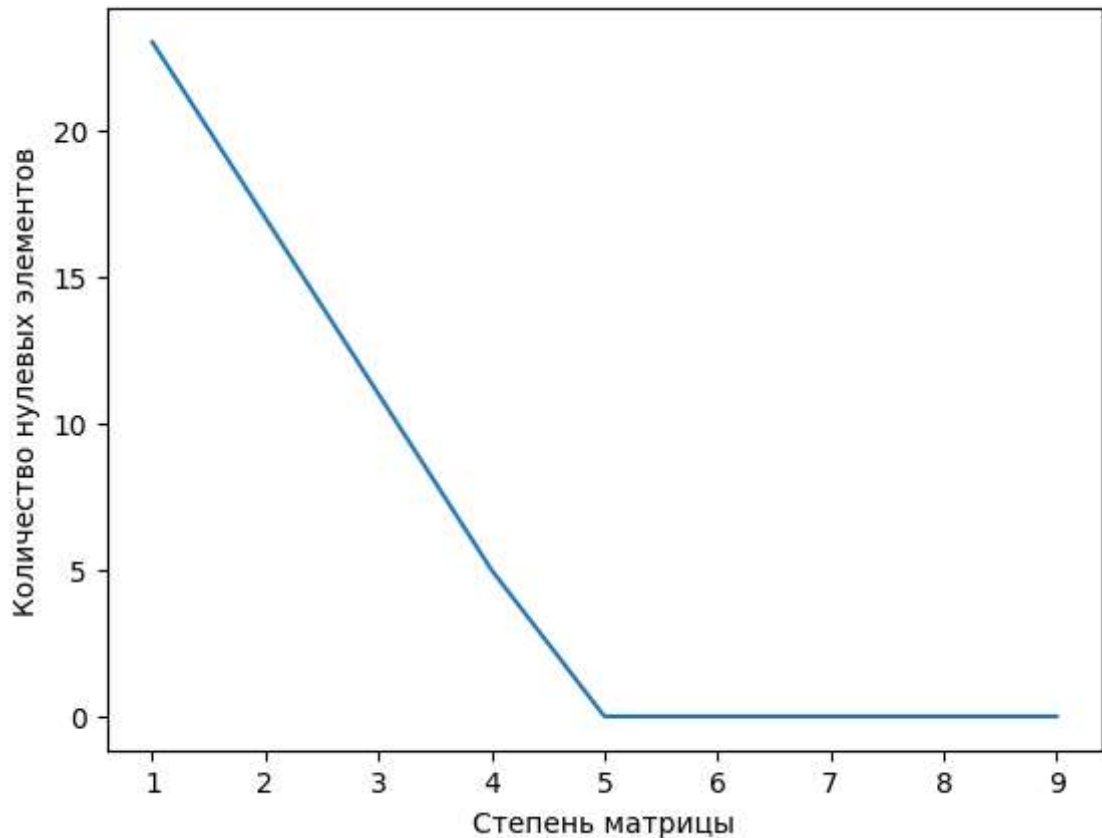
count_zeros = np.vectorize(count_zeros)
```

```
In [ ]: df = pd.DataFrame()
df['Степень матрицы'] = np.arange(1, 10)
df['Количество нулевых элементов'] = count_zeros(df['Степень матрицы'])
df
```

```
Out[ ]:   Степень матрицы  Количество нулевых элементов
0           1           23
1           2           17
2           3           11
3           4            5
4           5            0
5           6            0
6           7            0
7           8            0
8           9            0
```

```
In [ ]: sns.lineplot(df, x='Степень матрицы', y='Количество нулевых элементов')
```

```
Out[ ]: <AxesSubplot: xlabel='Степень матрицы', ylabel='Количество нулевых элементов'>
```



Так как существует  $n_0 = 5$  такое, что  $\min_{i,j} p_{ij}^{(n_0)} > 0$ , цепь является эргодической.

Выведем  $P^5$ .

```
In [ ]: np.round(np.linalg.matrix_power(P, 5), 4).tolist()
```

```
Out[ ]: [[0.3584, 0.3481, 0.2544, 0.032, 0.0067, 0.0004],
         [0.2326, 0.2423, 0.4017, 0.0848, 0.0343, 0.0043],
         [0.0089, 0.0007, 0.5905, 0.2112, 0.1548, 0.0339],
         [0.1193, 0.0245, 0.0014, 0.1681, 0.4788, 0.2079],
         [0.2935, 0.1111, 0.0173, 0.0005, 0.3277, 0.2499],
         [0.4642, 0.3292, 0.1198, 0.0084, 0.0007, 0.0778]]
```

#### 4. Моделирование вектора начальных вероятностей $p(0)$ согласно приложенному алгоритму

- Генерируем вектор  $\vec{r}$  из независимых и равномерно распределенных на отрезке  $[0, 1]$  случайных величин;
- Строим вариационный ряд, сортируя сгенерированный вектор;
- Находим длины отрезков, на которые вектор  $\vec{r}$  разбивает отрезок  $[0, 1]$  - получаем вектор начальных вероятностей.

```
In [ ]: r = np.random.random(m-1)
         r_series = np.sort(r)
         p_0 = np.diff(np.insert(r_series, [0, len(r_series)], [0, 1]))
         print(f'Вектор из независимых и равномерно распределенных на отрезке [0,1] случа')
         print(f'Вариационный ряд: {np.round(r_series, 4).tolist()}')
         print(f'Вектор начальных вероятностей: {np.round(p_0, 4).tolist()}')
```

Вектор из независимых и равномерно распределенных на отрезке  $[0,1]$  случайных величин:  $[0.3745, 0.9507, 0.732, 0.5987, 0.156]$

Вариационный ряд:  $[0.156, 0.3745, 0.5987, 0.732, 0.9507]$

Вектор начальных вероятностей:  $[0.156, 0.2185, 0.2241, 0.1333, 0.2187, 0.0493]$

## 5. Вычисление безусловных вероятностей состояний цепи через $k$ шагов

$$p(k)^T = p(0)^T \cdot P^k, k = 10$$

```
In [ ]: p_k = p_0 @ np.linalg.matrix_power(P, k)
p_k
```

```
Out[ ]: array([0.20349169, 0.15347911, 0.31096943, 0.10356577, 0.15288001,
0.07561398])
```

## 6. Моделирование $n$ траекторий полученной цепи за $k$ шагов

```
In [ ]: num_states = np.zeros(m)
trajectories = []
for i in range(n):
    state = np.argmax(p_0.cumsum() > np.random.random(1))
    trajectory = [state]
    for j in range(k):
        state = np.argmax(P[state].cumsum() > np.random.random(1))
        trajectory.append(state)
    num_states[state] += 1
    trajectories.append(trajectory)
```

Выведем 3 случайных траектории.

```
In [ ]: for i in range(3):
    ind = int(np.random.random(1) * len(trajectories))
    print(f'Номер в списке: {ind}\n{trajectories[ind]}')
```

Номер в списке: 31

[4, 4, 5, 5, 5, 5, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1]

Номер в списке: 72

[2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]

Номер в списке: 5

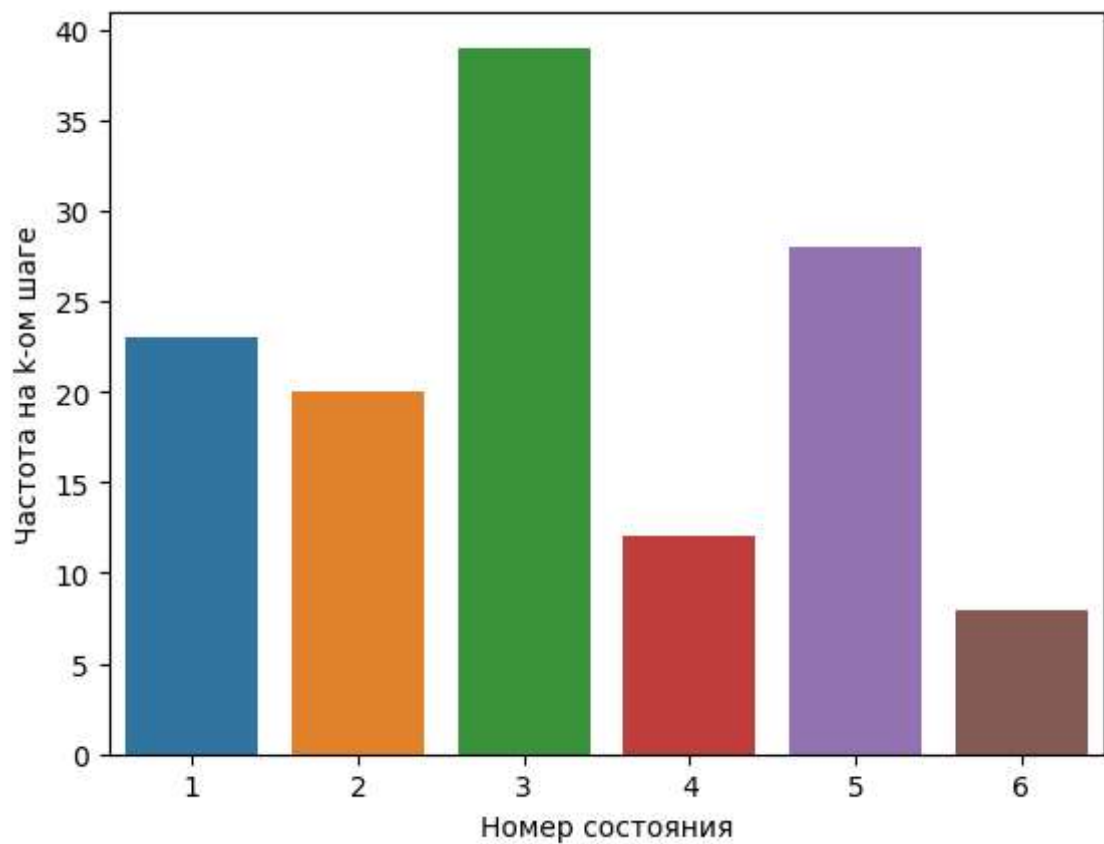
[1, 1, 1, 1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 0, 0, 1, 1, 1, 1]

```
In [ ]: state_df = pd.DataFrame({'Номер состояния': np.arange(1,7), 'Частота на k-ом шаге':
state_df
```

```
Out[ ]:   Номер состояния  Частота на k-ом шаге
0          1          23.0
1          2          20.0
2          3          39.0
3          4          12.0
4          5          28.0
5          6           8.0
```

```
In [ ]: sns.barplot(state_df, x='Номер состояния', y='Частота на k-ом шаге')
```

```
Out[ ]: <AxesSubplot: xlabel='Номер состояния', ylabel='Частота на k-ом шаге'>
```



## 7. Вычисление эмпирических вероятностей состояний цепи на $k$ -ом шаге и сравним их с теоретическими

```
In [ ]: state_df['Эмпирическая вероятность'] = state_df['Частота на k-ом шаге']/n
state_df['Теоретическая вероятность'] = p_k
state_df
```

Out[ ]:

	Номер состояния	Частота на k-ом шаге	Эмпирическая вероятность	Теоретическая вероятность
0	1	23.0	0.176923	0.203492
1	2	20.0	0.153846	0.153479
2	3	39.0	0.300000	0.310969
3	4	12.0	0.092308	0.103566
4	5	28.0	0.215385	0.152880
5	6	8.0	0.061538	0.075614

## 8. Вычисление финальных вероятностей состояний цепи

Для того, чтобы найти финальные вероятности состояний цепи, необходимо решить следующую систему:

$$\begin{cases} p_1 = 0.7p_1 + 0.2p_2 + 0.4p_6 \\ p_2 = 0.3p_1 + 0.6p_2 \\ p_3 = 0.2p_2 + 0.9p_3 \\ p_4 = 0.1p_3 + 0.7p_4 \\ p_5 = 0.3p_4 + 0.8p_5 \\ p_6 = 0.2p_5 + 0.6p_6 \\ p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 1 \end{cases}$$

```
In [ ]: p_final = np.array([0.205128, 0.15384, 0.307692, 0.102564, 0.153846, 0.0769231])
state_df['Финальная вероятность'] = p_final
state_df[['Эмпирическая вероятность', 'Теоретическая вероятность', 'Финальная ве
```

Out[ ]:

	Эмпирическая вероятность	Теоретическая вероятность	Финальная вероятность
0	0.176923	0.203492	0.205128
1	0.153846	0.153479	0.153840
2	0.300000	0.310969	0.307692
3	0.092308	0.103566	0.102564
4	0.215385	0.152880	0.153846
5	0.061538	0.075614	0.076923

## Выводы

При относительно большом количестве  $n = 130$  смоделированных траекторий, эмпирические вероятности состояний цепи близки к теоретическим. Кроме того, теоретические вероятности слабо отличаются от финальных. Таким образом, мы проверили эргодичность системы.