



UNIFACS

RELATÓRIO TÉCNICO A3

ESTRUTURAS

MATEMÁTICAS

ALUNOS:

Alisson Rayan | **RA:** 1272314418

Júlio César Souza | **RA:** 12723120855

Wesley Dantas | **RA:** 1272311443

DESCRIÇÃO

A **cadeia de Markov** é um modelo probabilístico utilizado para descrever sistemas que evoluem de um estado para outro ao longo do tempo, com a característica de que a probabilidade de transição depende apenas do estado atual. Neste projeto, implementamos uma simulação interativa que permite ao usuário definir:

- A quantidade de estados;
- A matriz de transição probabilística;
- O vetor de estado inicial;
- A quantidade de passos da simulação.

A aplicação foi projetada para rodar no terminal, preferencialmente no **cmd**, de forma clara e didática, permitindo análises visuais por meio de **gráficos** gerados ao final da simulação.

→ CÓDIGO FONTE

O código fonte do projeto, em Julia, pode ser encontrado pelo GitHub em: https://github.com/zhucssz/julia_base-a3-em, pois ocuparia muito espaço nesta documentação.

INSTALAÇÃO

Primeiramente, é necessário instalar a linguagem Julia, podendo ser encontrada no site oficial:

- <https://julialang.org/downloads/>

Posteriormente, é necessário instalar o pacote **Plots**, que foi usado para a criação dos gráficos, para demonstrar melhor o funcionamento do código. Para instalar o pacote **Plots**, basta abrir o REPL do Julia (aplicativo que aparece ao instalar a linguagem) e digitar as linhas:

```
using Pkg  
  
Pkg.add("Plots")
```

→ EXECUÇÃO

Após a instalação, para executar o código basta seguir os passos:

- Salve o código em um arquivo chamado `cadeia_markov.jl`;
- No cmd, navegue até o diretório onde está o arquivo;
- Execute digitando “`julia cadeia_markov.jl`”, sem aspas.

Ao executar o programa, ele irá perguntar o número dos estados, o vetor inicial, a matriz de transição e o número de passos, que devem ser digitados no terminal como requisitado.

Após concluído, o programa irá gerar os resultados, assim como um gráfico, em png, na mesma pasta em que o código está sendo executado.

CASO ESTUDADO:

No artigo "**Modelos Markovianos para Planes CSP de muestreo por aceptación**" (Revista Espacios, vol. 38, n.º 22, 2017), os autores aplicam a teoria de cadeias de Markov para modelar e avaliar planos de amostragem por aceitação (CSP) em processos industriais contínuos. Eles constroem matrizes de transição entre estados como "inspeção 100%" e "inspeção por amostragem", e utilizam a distribuição de probabilidades resultante para calcular métricas fundamentais, como o número esperado de unidades inspecionadas e a proporção de itens não conformes detectados. Esse modelo probabilístico, que se baseia em cadeias de estados com somatório de 1 por linha, é compatível com o simulador implementado no código em Julia.

Link: <https://www.revistaespacios.com/a17v38n02/a17v38n02p03.pdf>

APLICAÇÃO MANUAL DO CASO

→ **Cálculo manual:**

$$\begin{bmatrix} 0,997 & 0,136 & 0,072 \\ 0,002 & 0,856 & 0,006 \\ 0,003 & 0,008 & 0,922 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,997 \\ 0,002 \\ 0,003 \end{bmatrix} \quad \text{1º PASSO}$$

$$\begin{bmatrix} 0,997 & 0,136 & 0,072 \\ 0,002 & 0,856 & 0,006 \\ 0,003 & 0,008 & 0,922 \end{bmatrix} \cdot \begin{bmatrix} 0,997 \\ 0,002 \\ 0,003 \end{bmatrix} = \begin{bmatrix} 0,994009 + 0,000272 + 0,000072 \\ 0,003994 + 0,003732 + 0,000006 \\ 0,000997 + 0,000036 + 0,000922 \end{bmatrix}$$

$$2^\circ \text{ PASSO} = \begin{bmatrix} 0,997353 \\ 0,003732 \\ 0,003935 \end{bmatrix} \dots$$

→ Programa no terminal(cmd):

```
[i] Quantidade de estados da cadeia: 3

[i] Digite a matriz de transição (linha por linha, separados por espaço):
Linha 1: 0.997 0.002 0.001
Linha 2: 0.136 0.856 0.008
Linha 3: 0.072 0.006 0.922

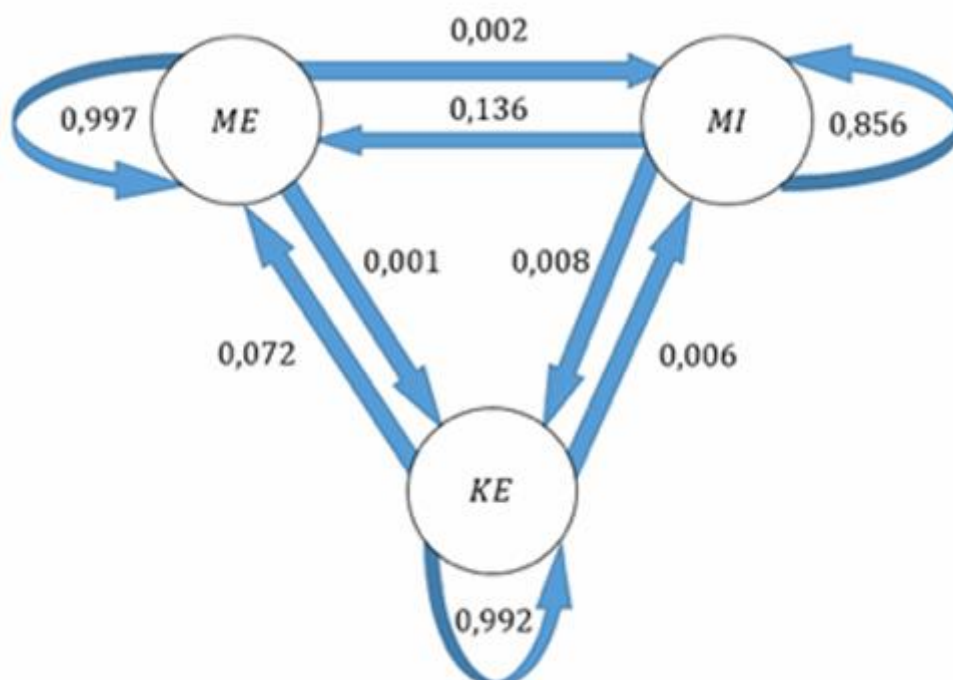
[i] Digite o vetor de estado inicial (separado por espaço): 1 0 0

[i] Quantos passos deseja simular? 10

[#] Estado inicial: [1.0, 0.0, 0.0]
[-] Após passo (1): [0.997, 0.002, 0.001]
[-] Após passo (2): [0.9944, 0.0037, 0.0019]
[-] Após passo (3): [0.992, 0.0052, 0.0028]
[-] Após passo (4): [0.9899, 0.0064, 0.0036]
[-] Após passo (5): [0.9881, 0.0075, 0.0044]
[-] Após passo (6): [0.9865, 0.0084, 0.0051]
[-] Após passo (7): [0.985, 0.0092, 0.0057]
[-] Após passo (8): [0.9837, 0.0099, 0.0064]
[-] Após passo (9): [0.9826, 0.0105, 0.0069]
[-] Após passo (10): [0.9816, 0.011, 0.0074]
```

Como é possível observar, os resultados apresentados batem com os passos 1 e 2 do programa, comprovando que o cálculo foi realizado corretamente.

→ Diagrama de transição:



SOLUÇÃO APLICADA

A solução foi construída utilizando conceitos fundamentais de cadeias de Markov com foco didático e interativo. O sistema simula a evolução dos estados ao longo do tempo multiplicando iterativamente o vetor de estados pelo transposto da matriz de transição.

Foi implementado também um controle de entrada de dados robusto, garantindo que a matriz de transição e o vetor inicial sejam válidos (com soma igual a 1 e consistência numérica).

A visualização dos resultados é feita com o pacote **Plots.jl**, gerando um gráfico com a evolução da distribuição de probabilidade de cada estado ao longo dos passos simulados.