Estoques

Controle de Estoques Utilizando Cadeias de Markov

Salvador Netto (2022040141) Nicolas Monteiro (2022039950)
Gustavo Aledi (2022040290)

Índice

1 Introdução **2**

2 Aplicação **3**

Capítulo 1

Introdução

A gestão eficiente de estoque é um componente vital para o funcionamento bem-sucedido de qualquer negócio que lida com produtos tangíveis. A otimização dos níveis de estoque é crucial para garantir a disponibilidade de produtos para os clientes, ao mesmo tempo em que minimiza os custos de armazenamento e reposição.

Neste contexto, os modelos de Cadeias de Markov têm sido amplamente aplicados na previsão e gestão de estoques, oferecendo uma abordagem analítica poderosa para entender as mudanças no nível de estoque ao longo do tempo. Esses modelos, baseados na teoria das probabilidades e processos estocásticos, permitem prever o comportamento futuro do estoque, considerando as transições probabilísticas entre diferentes estados.

O **método s e S**, dentro do contexto das Cadeias de Markov, é uma estratégia popular para determinar os pontos de reabastecimento ideal. Ele consiste em estabelecer dois pontos de controle: o nível mínimo no estoque (s) e o nível máximo de estoque (S). Quando o estoque atinge o nível mínimo ou abaixo dele, uma nova encomenda é feita para reabastecer o estoque até o nível máximo estabelecido.

O principal objetivo deste trabalho é explorar e aplicar o modelo de Cadeias de Markov em conjunto com o método s e S para a gestão de estoque. Este estudo buscará analisar como esses modelos podem ser utilizados para otimizar os níveis de estoque, minimizando os custos associados à manutenção de inventário sem comprometer a disponibilidade dos produtos para os clientes.

Serão examinados os conceitos fundamentais das Cadeias de Markov, a aplicação do método s e S na gestão de estoque e como essas ferramentas analíticas podem ser empregadas na tomada de decisões estratégicas relacionadas à reposição de estoque. Além disso, serão discutidos casos práticos e exemplos para ilustrar a aplicação desses modelos em cenários do mundo real.

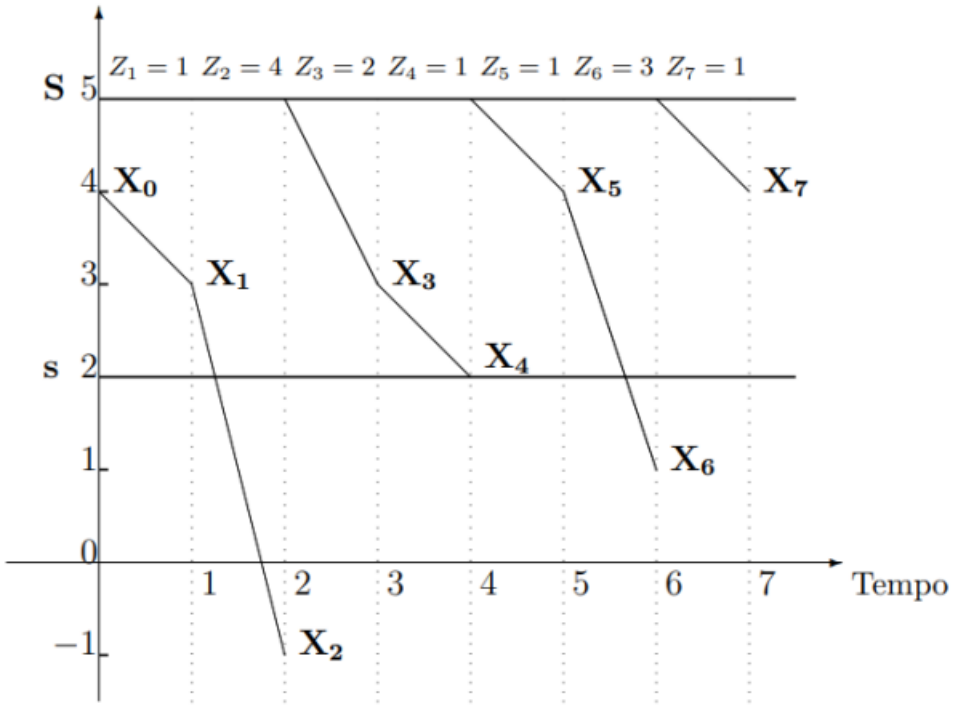
Por meio dessa pesquisa, busca-se fornecer insights valiosos sobre como os modelos de Cadeias de Markov e o método s e S podem ser implementados para melhorar a eficiência na gestão de estoques, contribuindo assim para o aprimoramento das práticas comerciais e a maximização dos resultados empresariais.

Neste trabalho estamos simulando um caso bem simples de estoque de lojas aplicado. Com isso em mente, vamos supor que estamos pegando dados de uma loja que vende celulares na gutierez, nosso produto em questão é o novo iphone, iphone 13.

Capítulo 2

Aplicação

Figura 2.1: Exemplo



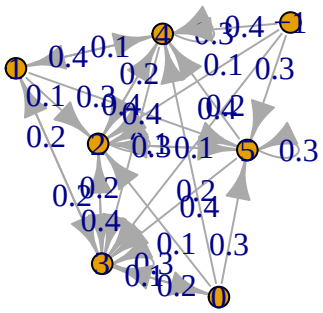
- Z_n é a demanda no n-ésimo dia
- s é o nível mínimo do estoque
- S o nível máximo do estoque
- X_n é a quantidade no estoque depois de satisfazer a demanda e antes de utilizar a estratégia

Se, após satisfazer a demanda Z_n, o estoque atinge o nível (inferior) s então é feita a reposição do estoque até o nível (superior) S. Se o estoque não atinge o nível inferior s então não é feita a reposição.

$$X_{n+1} = \begin{cases} X_n - Z_{n+1}, & s < X_n < S. \\ S - Z_{n+1}, & X_n \leq s. \end{cases}$$

```
   -1   0   1   2   3   4   5
-1  0 0.0 0.0 0.1 0.2 0.4 0.3
0   0 0.0 0.0 0.1 0.2 0.4 0.3
1   0 0.0 0.0 0.1 0.2 0.4 0.3
2   0 0.0 0.0 0.1 0.2 0.4 0.3
3   0 0.1 0.2 0.4 0.3 0.0 0.0
4   0 0.0 0.1 0.2 0.4 0.3 0.0
5   0 0.0 0.0 0.1 0.2 0.4 0.3
```

Figura 2.2: Topologia da Cadeia



Podemos observar através da topologia da cadeia em que não temos estados absorventes, ou seja, não há nenhum estado em que após chegar nele nunca mais sai, e isso faz sentido pois no nosso problema estamos lidando com estoques de loja, em nenhum momento o estoque de determinado produto vai ficar naquela mesma quantidade para sempre, pois temos as demandas, etc. Conseguimos ver também que temos uma classe não essencial que é referente ao estado -1 da cadeia, isso se deve ao fato de que ao atingirmos o limite inferior do estoque(definido como $s = 2$), nós repomos esse estoque até seu valor máximo (definido como $S = 5$), então a partir do momento em que chegamos numa quantidade de estoque inferior a "s" repomos ele até "S". Os outros estados que estão abaixo do limite inferior ou o próprio não são considerados não essenciais pois, a nossa variável Z_n(a demanda do n-ésimo dia) foi definida como tendo as seguintes probabilidades:

$P(Z_1 = 0) = 0.3$

$P(Z_1 = 1) = 0.4$

$P(Z_1 = 2) = 0.2$

$P(Z_1 = 3) = 0.1$

Visto isso, conseguimos observar que podemos ter $3, 2, 1$ vendas no dia e através da nossa matriz de transição os outros estados abaixo ou igual ao limite inferior são alcançados.

```
is.irreducible(cadeia)
```

```
[1] FALSE
```

Através do comando **is.irreducible()** podemos observar que a cadeia não é irredutível com a saída sendo "FALSE", isso se deve ao fato de como vimos anteriormente na topologia da cadeia, em que nem todos os estados se comunicam ente si, portanto a cadeia não é irredutível, ela será se pegarmos uma classe apenas das essenciais. A vantagem de acharmos uma classe irredutível é por que com ela conseguimos achar a distribuição invariante, e conseguimos fazer previsões de estoque e demandas para n dias, conforme os dias vão passando a probabilidade da cadeia vai se tornando muito parecida com o próximo dia e se for aumentando os dias com $n \to \infty$ a diferença das probabilidades de transição estará na décima, vigésima... casa decimal.

```
communicatingClasses(cadeia)
```

```
[[1]]
[1] "-1"

[[2]]
[1] "0" "1" "2" "3" "4" "5"
```

Como falado anteriormente, nem todos os estados se comunicam entre si e através do comando **communicationClasses()**, podemos notar que o estado -1 é o único que não se comunica com os demais. A comunicação entre os estados não precisa ser direta para haver comunicação, existe uma propriedade que diz que um estado i que se comunica com um estado j, e esse estado j se comunica com um estado k, o estado i se comunica com k, se $i \to j, j \to k$ então $i \to k$.

```
steadyStates(cadeia)
```

```
     -1          0          1         2         3         4         5
[1,]  0 0.02803738 0.08224299 0.2102804 0.2803738 0.2616822 0.1373832
```

Com o comando **steadyStates()** conseguimos obter a distribuição invariante, nessa função já é pegada apenas as classes essenciais para realizar esse cálculo. A seguir, mostraremos que ao decorrer do dias a matriz de transição vai se tornando cada vez mais próxima da distribuição invariante. Realizaremos para n (número de dias) igual a $2, 4, 6, 8 e 10$.

```
n2 = trans_matrix %*% trans_matrix
n4 = trans_matrix %*% trans_matrix %*% trans_matrix %*% trans_matrix
n6 = trans_matrix %*% trans_matrix %*% trans_matrix %*% trans_matrix  %*% trans_matrix %*% trans_m
n10 = trans_matrix %*% trans_matrix %*% trans_matrix %*% trans_matrix  %*% trans_matrix %*% trans_

n2
```

```
   -1    0    1    2    3    4    5
-1  0 0.02 0.08 0.20 0.30 0.28 0.12
0   0 0.02 0.08 0.20 0.30 0.28 0.12
1   0 0.02 0.08 0.20 0.30 0.28 0.12
2   0 0.02 0.08 0.20 0.30 0.28 0.12
3   0 0.03 0.06 0.19 0.23 0.28 0.21
4   0 0.04 0.11 0.25 0.30 0.21 0.09
5   0 0.02 0.08 0.20 0.30 0.28 0.12
```

```
n4
```

```
   -1      0      1      2      3      4      5
-1  0 0.0286 0.0824 0.2110 0.2790 0.2604 0.1386
0   0 0.0286 0.0824 0.2110 0.2790 0.2604 0.1386
1   0 0.0286 0.0824 0.2110 0.2790 0.2604 0.1386
2   0 0.0286 0.0824 0.2110 0.2790 0.2604 0.1386
3   0 0.0279 0.0838 0.2117 0.2839 0.2604 0.1323
4   0 0.0272 0.0803 0.2075 0.2790 0.2653 0.1407
5   0 0.0286 0.0824 0.2110 0.2790 0.2604 0.1386
```

```
n6
```

```
   -1        0        1        2        3        4        5
-1  0 0.027998 0.082232 0.210230 0.280470 0.261772 0.137298
0   0 0.027998 0.082232 0.210230 0.280470 0.261772 0.137298
1   0 0.027998 0.082232 0.210230 0.280470 0.261772 0.137298
2   0 0.027998 0.082232 0.210230 0.280470 0.261772 0.137298
3   0 0.028047 0.082134 0.210181 0.280127 0.261772 0.137739
4   0 0.028096 0.082379 0.210475 0.280470 0.261429 0.137151
5   0 0.027998 0.082232 0.210230 0.280470 0.261772 0.137298
```

```
n10
```

```
   -1          0          1         2         3         4         5
-1  0 0.02803719 0.08224294 0.2102801 0.2803743 0.2616827 0.1373828
0   0 0.02803719 0.08224294 0.2102801 0.2803743 0.2616827 0.1373828
1   0 0.02803719 0.08224294 0.2102801 0.2803743 0.2616827 0.1373828
2   0 0.02803719 0.08224294 0.2102801 0.2803743 0.2616827 0.1373828
3   0 0.02803743 0.08224246 0.2102799 0.2803726 0.2616827 0.1373849
4   0 0.02803767 0.08224366 0.2102813 0.2803743 0.2616810 0.1373820
5   0 0.02803719 0.08224294 0.2102801 0.2803743 0.2616827 0.1373828
```

Observando as saídas, podemos notar que a partir do sexto dia a matriz de transição já se torna extremamente semelhante aos valores da distribuição invariante.

```
meanRecurrenceTime(cadeia)
```

```
       0        1        2        3        4        5
35.666667 12.159091  4.755556  3.566667  3.821429  7.278912
```

Através do comando **meanRecurrenceTime()**, ela nos retorna o número esperado de etapas necessárias para que o processo retorne a um determinado estado, começando desse estado. Isso é conhecido como "tempo de recorrência". Formalmente, a fórmula para o Tempo de Recorrência Médio é dado por: $M_i = \frac{1}{\pi_i}$, onde π_i é a distribuição invariante. Pela saída dessa função, podemos observar novamente em que o estado -1 não tem um tempo médio de retorno para ele pois como já foi dito antes é uma classe não essencial, sendo mais uma forma de validar tudo que já foi dito anteriormente. O estado 3 retorna para ele em média 3,57 passos e é o estado que retorna mais rápido de nossa cadeia, isso faz sentido pois conforme vimos nos resultados da distribuição invariante $P_{ii} = 0.2803738$ com $i = 3$, ou seja, e o estado com a probabilidade de retorno mais alta da cadeia, e consequentemente em média retorna mais rápido também.

```
period(cadeia)
```

```
[1] 0
```

```
period(cadeia2)
```

```
[1] 1
```

Observando o período da cadeia com a classe não essencial através do comando **period()** ele nós retorna o valor 0, isso se deve ao fato de que como existe uma classe não essencial e por isso a cadeia é não irredutível não existe probabilidade de retorno do estado -1 pra ele mesmo,por isso o período é 0. Agora quando nós retiramos a classe não essencial da cadeia e ficamos apenas com as essenciais temos que o período da cadeia é 1, isso se deve ao fato de que $d_i = mdc(n \geq p_{ii} > 0)$, o mínimo divisor comum de $1, 2, 3...$ é 1, logo, o período da cadeia é 1 podemos ver isso para as matrizes de ordem superior 2 e 3 abaixo:

```
n2 = trans_matrix2 %*% trans_matrix2
n3 = trans_matrix2 %*% trans_matrix2 %*% trans_matrix2

n2
```

```
     0    1    2    3    4    5
0 0.02 0.08 0.20 0.30 0.28 0.12
1 0.02 0.08 0.20 0.30 0.28 0.12
2 0.02 0.08 0.20 0.30 0.28 0.12
3 0.03 0.06 0.19 0.23 0.28 0.21
4 0.04 0.11 0.25 0.30 0.21 0.09
5 0.02 0.08 0.20 0.30 0.28 0.12
```

```
n3
```

```
      0     1     2     3     4     5
0 0.030 0.088 0.218 0.286 0.252 0.126
1 0.030 0.088 0.218 0.286 0.252 0.126
2 0.030 0.088 0.218 0.286 0.252 0.126
3 0.023 0.074 0.197 0.279 0.280 0.147
4 0.030 0.081 0.211 0.272 0.259 0.147
5 0.030 0.088 0.218 0.286 0.252 0.126
```

```
previsao = markovchainSequence(n = 100, markovchain = cadeia, t0 = "-1")
previsao
```

```
  [1] "5" "3" "2" "5" "5" "4" "3" "2" "5" "5" "5" "3" "2" "3" "2" "4" "3" "3"
 [19] "3" "2" "5" "3" "2" "3" "3" "3" "3" "1" "2" "3" "2" "5" "4" "3" "0" "4"
 [37] "3" "3" "0" "5" "4" "3" "1" "5" "5" "4" "3" "1" "4" "2" "5" "5" "3" "1"
 [55] "3" "2" "3" "2" "4" "1" "4" "4" "3" "1" "5" "4" "3" "2" "4" "3" "2" "4"
 [73] "3" "2" "2" "4" "2" "2" "2" "4" "3" "1" "3" "2" "4" "4" "4" "3" "2" "4"
 [91] "3" "2" "4" "1" "5" "4" "3" "2" "3" "1"
```

Por meio do comando **markovchainSequence()**, nós conseguimos fazer previsões de qual a quantidade de estoques que nos temos em n passos, no nosso caso quisemos ver qual a quantidade de produtos que teremos em 100 dias começando com o estoque negativo, ao final de 100 dias nos estamos com 2 produtos no estoque.

Tabela 2.1: Frequência

previsao	Freq
0	**2**
1	9
2	**22**
3	31
4	**21**
5	15

Essa tabela mostra a frequência com que cada quantidade de produtos no estoque nos tivemos ao decorrer dos 100 dias, e podemos ver que as quantidades mais frequentes são $2, 3 e 4$.