Fundamentos da Análise Combinatória: Permutação e Arranjo

Compreendendo os Princípios de Contagem para Cenários de Probabilidade

Márcio Nicolau

2025-08-13

Table of contents

List of Figures

Introdução à Análise Combinatória

A análise combinatória é um ramo da matemática que estuda os métodos de contagem. Sua importância na estatística e na ciência de dados reside na capacidade de quantificar o número de maneiras distintas de organizar ou selecionar elementos de um conjunto, o que é fundamental para o cálculo de probabilidades. Antes de determinar a probabilidade de um evento, muitas vezes precisamos saber quantos resultados possíveis existem e quantos deles correspondem ao evento de interesse.

Este material visa apresentar os conceitos fundamentais de permutação e arranjo, bem como os princípios básicos de contagem, que são a base para a compreensão de cenários probabilísticos mais complexos.

Objetivo de Aprendizagem

Ao final desta aula, você será capaz de:

- Compreender e aplicar os princípios fundamentais de contagem (aditivo e multiplicativo).
- Distinguir e aplicar os conceitos de permutação (simples e com repetição) em diferentes cenários.
- Distinguir e aplicar os conceitos de arranjo (simples) em diferentes cenários.
- Resolver problemas de contagem que envolvem permutação e arranjo.
- Conectar os conceitos de contagem ao cálculo de probabilidades.

Princípios Fundamentais de Contagem

Os princípios de contagem são as ferramentas básicas para resolver problemas combinatórios.

Princípio Aditivo

O Princípio Aditivo afirma que se uma tarefa pode ser realizada de m maneiras e uma segunda tarefa pode ser realizada de n maneiras, e se essas duas tarefas não podem ser realizadas simultaneamente (são mutuamente exclusivas), então há $m + n$ maneiras de realizar uma ou outra tarefa.

Exemplo

Um estudante pode escolher um projeto de pesquisa em três áreas distintas: biologia (5 opções de temas), química (4 opções de temas) ou física (3 opções de temas). Quantas opções de projetos o estudante tem ao todo?

1. Como o estudante escolherá *apenas uma* área, as escolhas são mutuamente exclusivas.
2. Número total de opções = Opções de Biologia + Opções de Química + Opções de Física
3. Número total de opções $= 5 + 4 + 3 = 12$ opções.

Princípio Multiplicativo (ou Princípio Fundamental da Contagem)

O Princípio Multiplicativo afirma que se uma decisão pode ser tomada de m maneiras e, para cada uma dessas maneiras, uma segunda decisão pode ser tomada de n maneiras, então o número total de maneiras de tomar as duas decisões é $m \times n$. Este princípio pode ser estendido para qualquer número de decisões sequenciais.

> Exemplo
>
> Uma lanchonete oferece 3 tipos de pão, 4 tipos de recheio e 2 tipos de molho. Quantos tipos diferentes de sanduíches podem ser montados escolhendo um de cada item?
> Para montar um sanduíche, o estudante toma 3 decisões sequenciais:
> 1. Escolher o pão: 3 maneiras.
> 2. Escolher o recheio: 4 maneiras.
> 3. Escolher o molho: 2 maneiras.
>
> Número total de sanduíches $= 3 \times 4 \times 2 = 24$ tipos diferentes de sanduíches.

Exemplo de Árvore de Decisão da Montagem de Sanduíches

Para ilustrar o Princípio Multiplicativo:

(Note: O diagrama completo para 24 combinações seria muito grande. Este é um exemplo simplificado que mostra a estrutura de como as escolhas se ramificam. Cada caminho do "Início" ao "Fim" representa uma combinação única.)

Permutação

Permutação refere-se ao número de maneiras de organizar um conjunto de itens em uma ordem específica. A ordem dos elementos é importante.

Permutação Simples

A permutação simples é usada quando queremos saber o número de maneiras de organizar n objetos distintos. A fórmula para a permutação de n objetos é $P_n = n!$ (n fatorial), onde $n! = n \times (n-1) \times \cdots \times 2 \times 1$. Por definição, $0! = 1$.

Fórmula: $P_n = n!$

> Exemplo
>
> De quantas maneiras distintas 3 livros diferentes podem ser organizados em uma prateleira?
> 1. Temos 3 livros distintos. A ordem importa.
> 2. $P_3 = 3! = 3 \times 2 \times 1 = 6$ maneiras.
>
> As 6 maneiras seriam:
> - (Livro A, Livro B, Livro C)

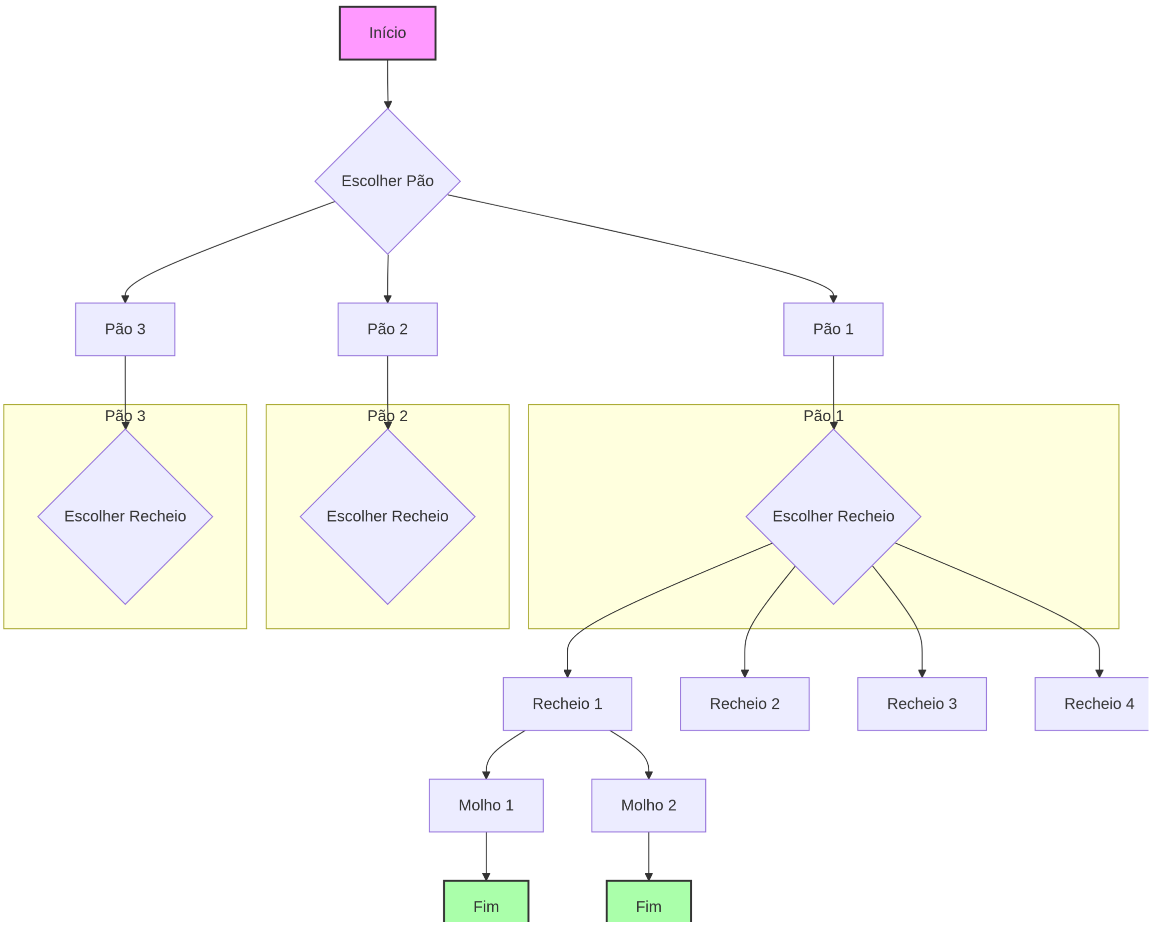

Figure 1: Árvore de Decisão da Montagem de Sanduíches

- (Livro A, Livro C, Livro B)
- (Livro B, Livro A, Livro C)
- (Livro B, Livro C, Livro A)
- (Livro C, Livro A, Livro B)
- (Livro C, Livro B, Livro A)

Exemplo de Código em Python (Permutação Simples)

```python
import math
from itertools import permutations
def permutacao_simples(n):
    """Calcula a permutação simples de n elementos."""
    return math.factorial(n)
# Exemplo: 3 livros na prateleira
n_livros = 3
resultado_permutacao = permutacao_simples(n_livros)
print(f"Número de maneiras de organizar {n_livros} livros: {resultado_permutacao}")
# Listando as permutações (para visualização)
itens = ["A", "B", "C"]
todas_permutacoes = list(permutations(itens))
print(f"Todas as permutações de {itens}: {todas_permutacoes}")
print(f"Total de permutações listadas: {len(todas_permutacoes)}")
```

Exemplo de Código em R (Permutação Simples)

```r
# Instalar e carregar o pacote 'gtools' se ainda não tiver
# install.packages("gtools")
library(gtools)
permutacao_simples <- function(n) {
  return(factorial(n))
}
# Exemplo: 3 livros na prateleira
n_livros <- 3
resultado_permutacao <- permutacao_simples(n_livros)
cat(sprintf(
  "Número de maneiras de organizar %d livros: %d\n",
  n_livros,
  resultado_permutacao
))
```

```
Número de maneiras de organizar 3 livros: 6
```

```r
# Listando as permutações (para visualização)
itens <- c('A', 'B', 'C')
todas_permutacoes <- permutations(
```

```
  n = length(itens),
  r = length(itens),
  v = itens
)
print(todas_permutacoes)
```

```
     [,1] [,2] [,3]
[1,] "A"  "B"  "C"
[2,] "A"  "C"  "B"
[3,] "B"  "A"  "C"
[4,] "B"  "C"  "A"
[5,] "C"  "A"  "B"
[6,] "C"  "B"  "A"
```

```
cat(sprintf("Total de permutações listadas: %d\n", nrow(todas_permutacoes)))
```

```
Total de permutações listadas: 6
```

Permutação com Repetição (ou Permutação de Itens Nem Todos Distintos)

Quando temos n objetos, onde n_1 são de um tipo, n_2 são de outro tipo, …, n_k são de um k-ésimo tipo, e $n_1 + n_2 + \cdots + n_k = n$, o número de permutações distintas é dado pela fórmula:

Fórmula: $P_n^{n_1,n_2,\ldots,n_k} = \frac{n!}{n_1!n_2!\ldots n_k!}$

Exemplo: Quantos anagramas (arranjos de letras) distintos podem ser formados com as letras da palavra "ARARA"?

A palavra "ARARA" tem $n = 5$ letras. Temos:

- 'A': 3 repetições ($n_1 = 3$)
- 'R': 2 repetições ($n_2 = 2$)

Número de anagramas $= \frac{5!}{3!2!} = \frac{5\times4\times3\times2\times1}{(3\times2\times1)(2\times1)} = \frac{120}{6\times2} = \frac{120}{12} = 10$ anagramas.

Exemplo de Código em Python (Permutação com Repetição)

```python
import math
from collections import Counter

def permutacao_com_repeticao(palavra):
    """Calcula o número de permutações distintas de uma string com repetições."""
    n = len(palavra)
    contagem_letras = Counter(palavra)
    denominador = 1
    for count in contagem_letras.values():
        denominador *= math.factorial(count)
```

```python
    return math.factorial(n) / denominador

# Exemplo: palavra "ARARA"
palavra_exemplo = "ARARA"
resultado_repeticao = permutacao_com_repeticao(palavra_exemplo)
print(
    f"Número de anagramas distintos da palavra '{palavra_exemplo}': {int(resultado_repeticao)}"
)
```

Exemplo de Código em R (Permutação com Repetição)

```r
permutacao_com_repeticao <- function(palavra) {
  n <- nchar(palavra)
  letras <- strsplit(palavra, "")[]

  # Contar as ocorrências de cada letra
  contagem_letras <- table(letras)

  denominador <- 1
  for (count in contagem_letras) {
    denominador <- denominador * factorial(count)
  }

  return(factorial(n) / denominador)
}

# Exemplo: palavra "ARARA"
palavra_exemplo <- "ARARA"
resultado_repeticao <- permutacao_com_repeticao(palavra_exemplo)
cat(sprintf(
  "Número de anagramas distintos da palavra '%s': %d\n",
  palavra_exemplo,
  as.integer(resultado_repeticao)
))
```

```
Número de anagramas distintos da palavra 'ARARA': 10
```

> **Note**
>
> **Extra:** atualizar os códigos acima para exibir os anagramas.

Arranjo

Arranjo refere-se ao número de maneiras de selecionar e organizar um subconjunto de itens de um conjunto maior, onde a ordem dos elementos selecionados é importante.

Arranjo Simples

O arranjo simples é usado quando selecionamos k objetos de um total de n objetos distintos, e a ordem de seleção é importante.

Fórmula: $A_n^k = \frac{n!}{(n-k)!}$

Exemplo: Em uma corrida com 8 atletas, de quantas maneiras diferentes os três primeiros lugares (ouro, prata, bronze) podem ser preenchidos?

Temos $n = 8$ atletas e queremos selecionar $k = 3$ para os três primeiros lugares, onde a ordem importa (ser ouro é diferente de ser prata). $A_8^3 = \frac{8!}{(8-3)!} = \frac{8!}{5!} = \frac{8 \times 7 \times 6 \times 5!}{5!} = 8 \times 7 \times 6 = 336$ maneiras.

Exemplo de Código em Python (Arranjo Simples)

```python
import math
from itertools import permutations  # Permutations com 'r' argumento para arranjos

def arranjo_simples(n, k):
    """Calcula o arranjo simples de n elementos tomados k a k."""
    if k > n:
        return 0  # Não é possível escolher mais itens do que o total disponível
    return math.factorial(n) // math.factorial(n - k)

# Exemplo: 8 atletas, 3 primeiros lugares
n_atletas = 8
k_lugares = 3
resultado_arranjo = arranjo_simples(n_atletas, k_lugares)
print(
    f"Número de maneiras de preencher os {k_lugares} primeiros lugares com {n_atletas} atletas: {resulta
)
# Listando os arranjos (para visualização, com itertools.permutations)
atletas = ["A1", "A2", "A3", "A4", "A5", "A6", "A7", "A8"]
primeiros_lugares = list(permutations(atletas, k_lugares))
# Note: A lista completa seria muito longa para imprimir, então mostramos um subconjunto.
print(
    f"Exemplo de 5 arranjos dos {k_lugares} primeiros lugares: {primeiros_lugares[:5]}..."
)
print(f"Total de arranjos listados (mesmo que o calculado): {len(primeiros_lugares)}")
```

Exemplo de Código em R (Arranjo Simples)

```
# install.packages("gtools")
library(gtools)
arranjo_simples <- function(n, k) {
  if (k > n) {
    return(0)
  }
  return(factorial(n) / factorial(n - k))
}
# Exemplo: 8 atletas, 3 primeiros lugares
n_atletas <- 8
k_lugares <- 3
resultado_arranjo <- arranjo_simples(n_atletas, k_lugares)
cat(sprintf(
  "Número de maneiras de preencher os %d primeiros lugares com %d atletas: %d\n",
  k_lugares,
  n_atletas,
  resultado_arranjo
))
```

```
Número de maneiras de preencher os 3 primeiros lugares com 8 atletas: 336
```

```
# Listando os arranjos (para visualização)
atletas <- c('A1', 'A2', 'A3', 'A4', 'A5', 'A6', 'A7', 'A8')
primeiros_lugares <- permutations(n = n_atletas, r = k_lugares, v = atletas)
# Note: A lista completa seria muito longa para imprimir, então mostramos um subconjunto.
print(head(primeiros_lugares, 5))
```

```
     [,1] [,2] [,3]
[1,] "A1" "A2" "A3"
[2,] "A1" "A2" "A4"
[3,] "A1" "A2" "A5"
[4,] "A1" "A2" "A6"
[5,] "A1" "A2" "A7"
```

```
cat(sprintf(
  "Total de arranjos listados (mesmo que o calculado): %d\n",
  nrow(primeiros_lugares)
))
```

```
Total de arranjos listados (mesmo que o calculado): 336
```

Distinção entre Permutação e Arranjo

A principal diferença entre permutação e arranjo reside na natureza da seleção:

- **Permutação:** Lida com a organização de *todos* os elementos de um conjunto (ou subconjunto quando a permutação é de n objetos distintos tomados k a k, que é equivalente ao arranjo), onde a ordem é crucial. Se você está reorganizando um conjunto completo, é uma permutação. Se você está escolhendo um subconjunto e a ordem importa, é um arranjo.
- **Arranjo:** Lida com a seleção e organização de um *subconjunto* de elementos de um conjunto maior, onde a ordem também é crucial.

Um bom "macete" para diferenciar é:

- **A ordem importa?**
  - **SIM:** É um Arranjo ou Permutação.
    - **Você usa *todos* os itens disponíveis?**
      - **SIM:** Permutação (simples ou com repetição).
      - **NÃO (escolhe um subconjunto):** Arranjo (simples).
  - **NÃO:** É uma Combinação (tópico de aula futura).

Diagrama de Fluxo de Decisão

Aplicações em Probabilidade

Os princípios de contagem são a espinha dorsal para o cálculo de probabilidades, especialmente em espaços amostrais finitos e equiprováveis (onde cada resultado tem a mesma chance de ocorrer). A probabilidade de um evento E é frequentemente calculada como:

$P(E) = \frac{\text{\# resultados favoráveis ao evento E}}{\text{\# resultados possíveis no espaço amostral}}$

Onde o "# resultados favoráveis" e o "# resultados possíveis" são determinados usando os princípios de contagem (permutação, arranjo, e futuramente, combinação).

Exemplo

Três amigos (A, B, C) sentam-se aleatoriamente em 3 cadeiras. Qual a probabilidade de A sentar-se na primeira cadeira, B na segunda e C na terceira?

1. **Número total de resultados possíveis (Espaço Amostral):** São 3 amigos em 3 cadeiras, e a ordem importa. Isso é uma permutação simples de 3 elementos: $P_3 = 3! = 6$ maneiras.
2. **Número de resultados favoráveis ao evento (Evento E):** O evento é " A na 1^a, B na 2^a, C na 3^a ". Existe apenas 1 maneira de isso acontecer: (A, B, C).
3. **Probabilidade:** $P(\text{A na } 1^a,\ \text{B na } 2^a,\ \text{C na } 3^a) = \frac{1}{6}$

Este exemplo simples demonstra como os conceitos de contagem se aplicam diretamente ao cálculo de probabilidades.

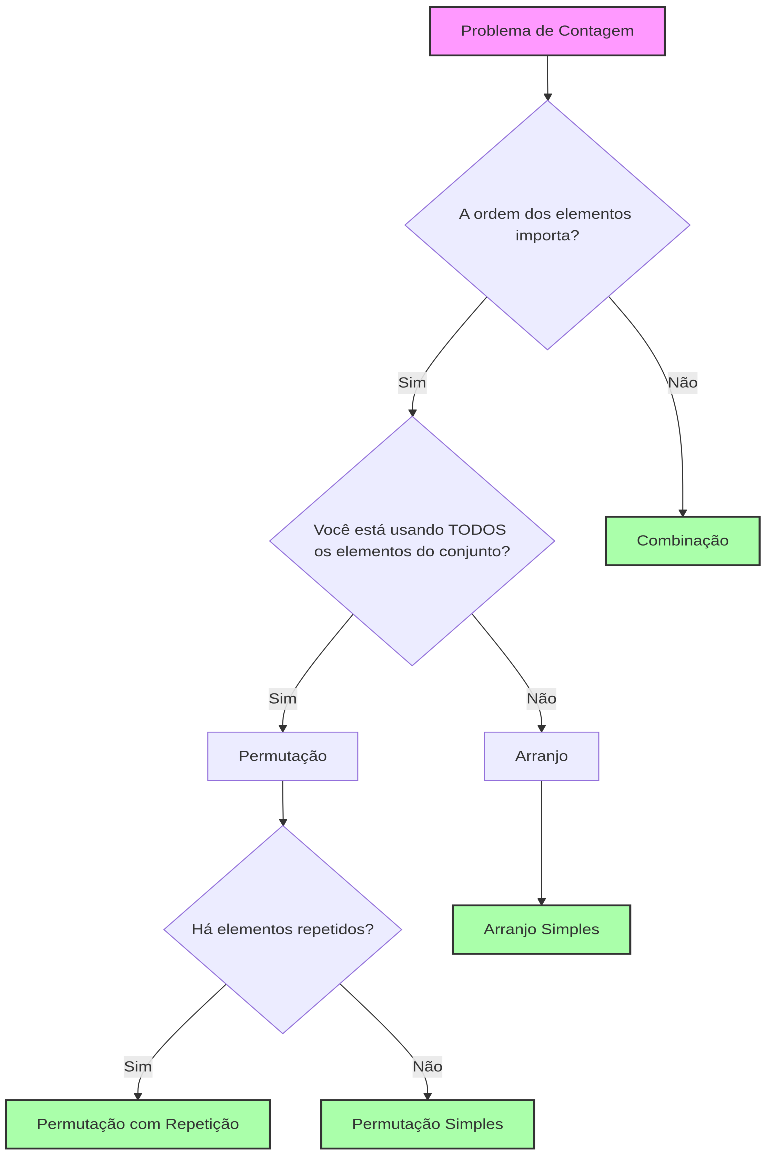

Figure 2: Diagrama de Fluxo de Decisão

Verificação de Aprendizagem

Resolva os problemas abaixo, identificando se é um caso de Permutação (simples ou com repetição) ou Arranjo (simples) e aplicando a fórmula correta.

1. **Senha de Banco:** Um banco exige que as senhas dos caixas eletrônicos sejam formadas por 4 dígitos distintos. Quantas senhas diferentes podem ser criadas? (Dígitos de 0 a 9).
2. **Pódio Olímpico:** Em uma final olímpica de natação com 10 nadadores, de quantas maneiras diferentes os atletas podem ganhar as medalhas de ouro, prata e bronze?
3. **Formação de Palavras:** Quantos anagramas distintos podem ser formados com as letras da palavra "MATEMATICA"?
4. **Ordem de Atendimento:** Uma fila de espera em um consultório médico tem 5 pacientes. De quantas maneiras diferentes esses pacientes podem ser atendidos?
5. **Placas de Carro - Simplificado:** Considerando que as placas de carro antigas no Brasil eram formadas por 2 letras seguidas por 4 dígitos (ex: AB-1234), quantas placas distintas poderiam ser formadas se:
   a) As letras e os dígitos podem se repetir.
   b) As letras devem ser distintas, mas os dígitos podem se repetir. (Considere 26 letras no alfabeto e 10 dígitos).

Referências Bibliográficas

BUSSAB, Luiz O. de M.; MORETTIN, Pedro A. **Estatística Básica**. 9. ed. São Paulo: Saraiva, 2017.

MAGALHÃES, Marcos N.; LIMA, Antonio C. P. de. **Noções de Probabilidade e Estatística**. 7. ed. São Paulo: Edusp, 2013.

TRIOLA, Mario F. **Introdução à Estatística**. 12. ed. Rio de Janeiro: LTC, 2017.