



TUTORIAL EM R PARA O MÉTODO SIMPLE AGREGATION OF PREFERENCES EXPRESSED BY ORDINAL VECTORS – MULTI DECISION MAKERS

Thiago marques Teixeira de Oliveira

t_hiagomarkes@hotmail.com

Marcos dos Santos

marcosdossantos_doutorado@yahoo.com.br

Paulo Afonso Lopes da Silva

pauloafonsolopes@ime.eb.br

O presente tutorial tem por objetivo apresentar uma solução computacional do método multicritério ordinal Simple Aggregation of Preferences Expressed by Ordinal Vectors - Multi Decision Makers (SAPEVO-M). Este método é uma nova versão do método tradicional SAPEVO (Gomes et al. 1997), que generaliza o método para múltiplos decisores. Para tal foi escolhido o R, que é um software livre e largamente utilizado no mundo todo, nas mais diversas áreas do conhecimento. O mesmo é de fácil operação e rápida resolução de problemas complexos. Desenvolveremos um tutorial, passo a passo, para a utilização dos pacotes desenvolvidos e que serão acessíveis abertamente por meio da plataforma Github, utilizando como exemplo um artigo desenvolvido por Santos et al. (2019) de ordenação de equipamentos de freezer em uma panificadora do RJ. Os pacotes terão como saída: os pesos dos critérios, a matriz de avaliação, a pontuação dos critérios e o ranking com as alternativas melhores classificadas pelo método para o apoio à decisão.

Palavras-chave: Sapevo-M, R, Pacote, Software Livre, Apoio a Decisão Multicritério; ordenação de Equipamentos para panificadora.

1. Introdução



Segundo Gomes e Gomes (2019), os métodos de apoio multicritério à decisão (AMD) são técnicas de assessoramento ao agente decisor (quer seja indivíduo, grupo de indivíduos, comitê de especialistas ou lideranças) para a tomada de decisões sobre problemas complexos, selecionado, avaliando ou ordenando alternativas mediante diferentes pontos de vista e dentro de critérios pré-estabelecidos.

Os problemas de AMD podem ser enquadrados em quatro tipos de problemática de referência, descritas por Gomes e Gomes (2019) como:

- a) Problemática $P.\alpha$ – visa esclarecer a decisão pela escolha de um subconjunto de alternativas o mais restrito possível. Portanto, o resultado pretendido é uma escolha;
- b) Problemática $P.\beta$ – visa esclarecer a decisão por uma triagem resultante da alocação de cada alternativa a uma classe (ou categoria). Portanto, o resultado pretendido é uma ação de classificação;
- c) Problemática $P.\gamma$ – visa esclarecer a decisão por uma organização obtida pelo reagrupamento de parte ou de todas as ações em classes de equivalência, que são ordenadas de modo parcial ou completo, conforme as preferências do(s) decisor(es). Portanto, o resultado pretendido é um procedimento de ordenação ou *ranking*;
- d) Problemática $P.\delta$ – visa esclarecer a decisão por uma descrição das ações e de suas consequências. Portanto, o resultado pretendido é um procedimento cognitivo ou uma descrição.

O método Sapevo-M se enquadra na problemática $P.\gamma$ e é uma evolução do método proposto por Gomes et al. (1997), que generaliza o método para múltiplos decisores, ao passo que a versão original possibilita apenas a utilização de apenas um único decisor. Trouxe também o incremento do processo de Normalização das matrizes de avaliação, trazendo uma maior consistência ao método, o que evitou um ordenamento prévio dos critérios e sendo a mesma resultado da vontade dos decisores.

Segundo TEIXEIRA et al. (2019), basicamente, o método consiste em dois processos. Preliminarmente, deve ser realizada a transformação da preferência ordinal entre critérios, a ser expressa por um vetor representando os pesos dos critérios. Em seguida, é feita a transformação ordinal da preferência entre alternativas dentro de um determinado conjunto de critérios, expressa por uma matriz. Uma série de comparações pareadas entre as opções quer seja critérios ou alternativas dentro de um determinado critério, denotam as informações de preferência individual de cada decisor.

O método é traduzido em uma sequência de passos, quais sejam:

- Transformação ordinal da preferência entre critérios, expressa por um vetor representando os pesos dos critérios;
- Transformação ordinal da preferência entre alternativas dentro de um determinado conjunto de critérios, expressa por uma matriz de avaliação.

1.1 Transformação ordinal da preferência entre critérios, expressa por um vetor representando os pesos dos critérios;

Sejam c_i e c_j dois critérios dentro de um conjunto de critérios $C = \{c_1, c_2, \dots, c_i, \dots, c_j, \dots\}$, o grau de preferência entre eles é dado por $\delta c_i c_j$, onde temos:

$$c_i c_j = 1 \leftrightarrow c_i \approx c_j$$



$$c_i c_j > 1 \leftrightarrow c_i > c_j$$

$$c_i c_j < 1 \leftrightarrow c_i < c_j$$

Sendo:

\approx = "Tão importante quanto"

$>$ " Mais importante"

$<$ "Menos importante"

Uma escala de sete pontos expressa a relação entre as alternativas, na qual são mensuradas, relativamente, a importância entre cada opção. A partir da avaliação entre alternativas, é obtido uma matriz com a representação numérica correspondente. A relação entre a escala de preferência e o valor numérico é expressa na Tabela 1.

Tabela 1 - A relação entre as opções de critérios e escala numérica de preferências

Escala 1 (Símbolo)	Escala Nominal	Escala Numérica
$<<< 1$	Absolutamente Pior	-3
$<< 1$	Muito Pior	-2
< 1	Pior	-1
1	Equivalente	0
> 1	Melhor	1
$>> 1$	Muito Melhor	2
$>>> 1$	Absolutamente Melhor	3

Seja D um conjunto de atores que decidem DM (Decision Makers), $D = \{DM1, DM2, \dots, DMk, \dots, DMn, \dots\}$ que expressam suas opiniões sobre os critérios em ordem de preferência. Para um agente decisor DMk, estas informações geram uma matriz de avaliação MDMk. A relação entre as duas escalas da tabela 1 permite a transformação da matriz MDMk = $[\delta c_i c_j]$ em um vetor coluna $[V_i]$ de tal forma que:

$V_i = \sum_{j=1}^m$, observando-se a escala 2 da tabela 1 para $i=\{1, \dots, m\}$ e $k=\{1, \dots, n\}$. O vetor V representa as preferências entre critérios de cada DM. Este método prevê, também, terminado o processo de integração da matriz, a normalização do vetor resultante. Esta normalização garante a geração de valores não negativos, por meio da equação $v = [(a_{ij} - \text{Mín } a_{ij}) \div (\text{Máx } a_{ij} - \text{Mín } a_{ij})]$. Durante a normalização dos critérios, os menores valores, dos pesos dos critérios, assumem valor zero, sendo substituídos por um valor arbitrado de 1% do valor imediatamente superior; este procedimento deve-se ao fato de não fazer sentido que um determinado critério possua peso nulo.



1.2 Transformação ordinal da preferência entre alternativas dentro de um determinado conjunto de critérios, expressa por uma matriz de avaliação

Desta forma, transforma-se a matriz $E = \{e_{i,k,l}\}$ em um vetor coluna A_i , onde $e_{i,k,l}$ é o valor da comparação pareada da alternativa a_k com a alternativa a_l dentro do critério I , para cada agente decisor DM . Logo, obtém-se o vetor coluna A , onde $A = \sum ADM$ da matriz M ($n \times m$). Alterando-se o critério I na qual as alternativas foram avaliadas, procede-se novamente a comparação pareada e a integração do vetor, de modo que a matriz M seja toda preenchida.

Finalmente, o resultado da preferência entre as alternativas é expresso pelo vetor resultante da multiplicação matricial entre o vetor peso dos critérios V e a matriz de avaliação das alternativas M . As alternativas são, então, ordenadas em ordem decrescente dos valores numéricos obtidos, gerando-se o ranking desejado.

2. A linguagem R e a IDE RStudio

R é uma linguagem de programação orientada a objetos, open Source e ambiente computacional utilizada no mundo todo e nas mais diversas áreas do conhecimento, é uma das linguagens mais utilizadas pelos cientistas de dados, Analistas de dados, Estatísticos, largamente utilizado na academia e em grandes empresas, como Google, Amazon entre outras... É utilizada para manipulação de dados, ferramenta estatística e ferramenta Gráfica de alto nível e é extremamente versátil, pois pode ser compilada e rodar em sistemas operacionais como Linux, Windows e Mac.

Foi inspirada na linguagem S, que foi desenvolvida na Bell Lab, criada por dois Estatísticos de renome internacional, quais sejam: Ross Lhaka e Robert Gentleman, no departamento de Estatística da universidade de Auckland, Nova Zelândia. Surgiu em 1995, como software livre, a versão 1.0.0 foi lançada no ano de 2000, hoje estamos com a versão 3.6.1 da linguagem.

O Rstudio é a IDE (Ambiente de desenvolvimento integrado) mais utilizada para a compilação da linguagem R, também compila outras linguagens como c++, Python e Java. Na prática, o objetivo de uma IDE é facilitar e potencializar a programação naquela linguagem, por meio de um console, gerenciando gráficos e facilitando o Debug dos erros.

Atualmente o R possui mais de 15.000 Pacotes só no CRAN (Comprehensive R Archive Network), fora os desenvolvidos e colocados no Github (Sistema de gerenciamento de projetos e versões de códigos assim como uma plataforma de rede social criado para desenvolvedores)

Os mesmos são funções que os usuários usaram reiteradas vezes e acharam melhor empacotar essa solução para que fosse usada frequentemente, o R tem um pacote chamado Base, que guarda todas as informações basilares da linguagem e quando o usuário precisa de utilizar outros pacotes, os mesmos precisam ser instalados e carregados, seja via CRAN ou Github.



3. Exemplo de aplicação: Ordenação de equipamentos de freezer em uma panificadora do RJ

Para exemplificar como utilizar os algoritmos desenvolvidos em R, os mesmos serão usados na ordenação de equipamentos de freezer em uma panificadora do RJ. Serão utilizados três modelos de freezer, propostos por *Santos et al. (2019)*, para seleção de um freezer para uma panificadora no RJ.

A panificadora e confeitaria Doce Alto localizada no Estado do Rio de Janeiro é um estabelecimento de pequeno porte atendendo aos consumidores locais e às pequenas empresas do entorno.

Segundo *Santos et al. (2019)*, existem diversas etapas para produzir os pães e uma delas é a passagem pelo freezer, vale ressaltar que é nessa fase que surge o processo crítico. Posto que, é através deste equipamento que ocorre o retardamento da fermentação fazendo com que sempre haja pães frescos disponíveis para o consumo dos clientes.

Desta forma *Santos et al. (2019)*, foram escolhidos três modelos, quais sejam:

Gelopar

O modelo GCCP2P da Gelopar é de inox e composto de duas portas. O freezer possui controlador eletrônico digital com indicador de temperatura e degelo automático natural. O equipamento é capaz de alcançar uma temperatura que vai de -20°C a -12°C , a refrigeração é de ar forçado com serpentina aletada, revestimento externo em aço inox 430 e interno aço galvanizado. Quanto às dimensões (CxAxL) 68x203x84,6 cm.



FIGURA 1 – Gelopar – *Santos et al. (2019)*,

Metal frio

O equipamento possui compressor de alta eficiência e condensador helicoidal, congela alimentos de forma rápida e eficiente, mesmo em condições climáticas extremas. Por meio de uma chave seletora, é possível escolher entre três funções: refrigerador de bebidas, conservador de gelos ou freezer para congelados. O modelo oferece uma faixa de temperatura específica para armazenar alimentos, seguindo os requisitos da Anvisa, dispensa



a limpeza frequente e proporciona maior vida útil ao motor e ao compressor. É o equipamento de mais baixo consumo de energia da categoria.



FIGURA 2 – Metalfrio – *Santos et al. (2019)*,

Frilux

O freezer vertical modelo RF01 é utilizado para armazenagem e conservação de gelo, produtos previamente congelados e refrigerados em geral. Possui dupla ação, grade de ventilação plástica, aquecimento no quadro das portas, refrigeração estática e prateleiras reguláveis e inclináveis. Quanta à temperatura, a mesma varia entre 5°C à -15°C, possui capacidade de 560 litros e dimensões de (CxAxL) 66x200x76 cm.



FIGURA 3 – Frilux – *Santos et al. (2019)*,

Segundo *Santos et al. (2019)* serão avaliados três critérios, quais sejam:



- 1) **Capacidade:** Neste ponto será avaliada a medida em litros, ou seja, a quantidade de itens que cada freezer é capaz de acomodar.
- 2) **Custo de Aquisição:** este critério é o valor em reais bruto para que seja possível efetuar a compra do equipamento.
- 3) **Custo de Manutenção:** É o custo de todo e qualquer reparo que engloba mão de obra, ferramentas, peças de reposição, depreciação do equipamento, etc.

A avaliação do projeto foi elaborada por dois gerentes dos diferentes turnos do estabelecimento, cujos nomes são: Renato (Decisor 1) e Branca (Decisor 2).

3.1. Instalação dos pacotes alocados no Github

Para utilização da função adequadamente, devemos primeiramente fazer o **download** dos **pacotes** desenvolvidos que estão **alocados no Github**, com a **sequência de passos** abaixo:

- 1) **Comando `install.packages("devtools")`** #Instala o pacote **devtools** que habilitará a função **install_github()**, a qual é responsável por efetuar o **download dos pacotes alocados no Github**;
- 2) **Comando `library(devtools)`** #Carrega o pacote **devtools** que habilitará a função **install_github()**, a qual é responsável por efetuar o download dos pacotes alocados no Github;
- 3) **Comando `install_github("ferpatugo/sapevo_input_pesos")`** #Instala a função que obterá os **inputs automáticos** da função **sapevo_pesos()**
- 4) **Comando `library(sapevoinputpesos)`** #Carrega o pacote com a função **sapevo_input_pesos()** que foi instalado;
- 5) **Comando `install_github("ferpatugo/sapevo_pesos")`** #Instala a função **sapevopesos()** que calculará os pesos dos critérios do método **Sapevo-M**, os mesmos são **fundamentais** para que a função **sapevoranking()** consiga efetuar o **cálculo da matriz avaliação** e as **ordenações das alternativas**;
- 6) **Comando `library(sapevopesos)`** #Carrega o pacote com a função **sapevo_pesos()** que foi instalado;
- 7) **Comando `install_github("ferpatugo/sapevo_input_ranking")`** #Instala a função que obterá os **inputs automáticos** da função **sapevo_ranking()**
- 8) **Comando `library(sapevoinputranking)`** #Carrega o pacote com a função **sapevo_input_ranking()** que foi instalado;
- 9) **Comando `install_github("ferpatugo/sapevo_ranking")`** #Instala a função que calculará tanto a **matriz avaliação do projeto**, o **peso das alternativas** e a **ordenação**



das **alternativas** inseridas;

10) Comando `library(sapevoranking)` #Carrega o pacote com a função `sapevo_ranking()` que foi instalado;

Conforme a figura 4:

```
#tutorial
install.packages("devtools") #Instala o pacote devtools que habilitará a função intall_github() , a qual é responsável por efetuar o dow
library(devtools) #Carrega o pacote devtools que habilitará a função intall_github() , a qual é responsável por efetuar o dowload dos pa

install_github("ferpatugo/sapevo_input_pesos",force = T) #Instala a função que obterá os inputs automáticos da função sapevo_pesos()
library(sapevoinputpesos) #Carrega o pacote sapevo_input_pesos que foi instalado;

install_github("ferpatugo/sapevo_pesos") #Instala a função que calculará os pesos dos critérios do método Sapevo-M, os mesmos são funda
library(sapevopesos) #Carrega o pacote sapevo_pesos que foi instalado;

install_github("ferpatugo/sapevo_input_ranking") #Instala a função que obterá os inputs automáticos da função sapevo_ranking()
library(sapevoinputranking) #Carrega o pacote sapevo_input_pesos que foi instalado;

install_github("ferpatugo/sapevo_ranking") #Instala a função que calculará tanto a matriz avaliação do projeto, o peso das alternativas
library(sapevoranking) #Carrega o pacote sapevo_pesos que foi instalado;
```

FIGURA 4 – Carregando e Instalando os Pacotes via Github

3.2. Utilizando a função `sapevo_pesos()` com inputs manuais

Uma vez instalados e devidamente carregados, começaremos inserindo o nome do projeto, nome dos decisores, alternativas e critérios de avaliação, como ilustra a figura 5:

```
sapevo_pesos("Panificadora",
             c("Renato","Branca"),
             c("Gelopar","MetaIfrio","Frilux"),
             c("Capacidade","Custo Aquisição","Custo Manutenção"),
```

FIGURA 5 – Inserindo o nome do projeto, decisores, alternativas e critérios

O próximo passo é colocar os vetores resultantes da avaliação dos critérios pelos decisores:

A avaliação é feita através de uma comparação pareada dos critérios, por cada decisor:

Decisor 1: Renato

Tabela 2 – Comparação pareada dos critérios

vs	Capacidade	Custo de aquisição	Custo Manutenção
Capacidade	Equivalente (0)	Melhor (1)	Muito Melhor (2)
Custo de aquisição	Pior (-1)	Equivalente (0)	Absolutamente Melhor (3)
Custo Manutenção	Muito pior (-2)	Absolutamente Melhor(-3)	Equivalente (0)



Inserimos no algoritmo conforme a figura 6 ilustra:

```
sapevo_pesos("Panificadora",
             c("Renato", "Branca"),
             c("Gelopar", "Metalfrio", "Frilux"),
             c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
             #vetor_notas_decisores(critérios)
             c(0, 1, 2, -1, 0, 3, -2, -3, 0),
```

FIGURA 6 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 1

Primeira linha da Tabela 2	Segunda linha da Tabela 2	Terceira linha da Tabela 2
----------------------------------	---------------------------------	----------------------------------

Decisor 2: Branca

Tabela 3 – Comparação pareada dos critérios

vs	Capacidade	Custo de aquisição	Custo Manutenção
Capacidade	Equivalente (0)	Muito Melhor (2)	Melhor (1)
Custo de aquisição	Muito pior (-2)	Equivalente (0)	Melhor (1)
Custo Manutenção	Pior (-1)	Pior (-1)	Equivalente (0)

Inserimos no algoritmo conforme a figura 7 ilustra:

```
sapevo_pesos("Panificadora",
             c("Renato", "Branca"),
             c("Gelopar", "Metalfrio", "Frilux"),
             c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
             #vetor_notas_decisores(critérios)
             c(0, 1, 2, -1, 0, 3, -2, -3, 0),
             c(0, 2, 1, -2, 0, 1, -1, -1, 0)
)
```

FIGURA 7 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 2

Primeira linha da Tabela 3	Segunda linha da Tabela 3	Terceira linha da Tabela 3
----------------------------------	---------------------------------	----------------------------------



Selecionar todas as linhas e pressionar **ctrl+enter**, ou o **botão Run**, mostrados nas figuras 8 e 9 abaixo:



FIGURA 8 – Run

```
sapevo_pesos("Panificadora",
             c("Renato", "Branca"),
             c("Gelopar", "Metalfrio", "Frilux"),
             c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
             #vetor_notas_decisores(critérios)
             c(0, 1, 2, -1, 0, 3, -2, -3, 0),
             c(0, 2, 1, -2, 0, 1, -1, -1, 0))
```

FIGURA 9 – Selecionando as linhas e rodando

Vamos ver o Output da função sapevo_pesos() na figura 10:

```
-----O nome do seu projeto: Panificadora -----
-----As alternativas do seu projeto: Gelopar Metalfrio Frilux -----
-----Os Criterios do seu projeto: Capacidade Custo Aquisição Custo Manutenção -----
-----Os pesos do metodo sapevo-M: 2 1.075 0.01075 -----
```

FIGURA 10 – Output sapevo_pesos()

A função cria também o vetor_peso internamente que será usada na sapevo_ranking().

3.3. Utilizando a função sapevo_ranking() com inputs manuais

O próximo passo é colocar os vetores resultantes da avaliação das alternativas segundo os critérios pelos decisores:

A avaliação é feita através de uma comparação pareada dos critérios, por cada decisor para cada critério:

Decisor 1: Renato

Critério 1: Capacidade

Tabela 4 – Comparação pareada dos critérios

vs	Gelopar	Metalfrio	Frilux
Gelopar	Equivalente (0)	Equivalente (0)	Muito pior (-2)
Metalfrio	Equivalente (0)	Equivalente (0)	pior (-1)



Frilux	Muito melhor (2)	Melhor(1)	Equivalente (0)
--------	------------------	-----------	-----------------

Obs1:

Incluir o **mesmo input utilizado no sapevo_pesos()**, **nome** do projeto, **decisores**, **alternativas**, **critérios** e inserir o **vetor_peso**, que foi **calculado automaticamente** pela função **sapevo_pesos()**.

```
sapevo_ranking("Panificadora",
  c("Renato", "Branca"),
  c("Gelopar", "Metalfrio", "Frilux"),
  c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
```

FIGURA 11 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 1

Decisor 2: Branca

Critério 1: Capacidade

Primeira
linha da
Tabela 4

Segunda
linha da
Tabela 4

Terceira
linha da
Tabela 4

Tabela 5 – Comparação pareada dos critérios

vs	Gelopar	Metalfrio	Frilux
Gelopar	Equivalente (0)	pior (-1)	Muito pior (-2)
Metalfrio	Melhor(1)	Equivalente (0)	Muito pior (-2)
Frilux	Muito melhor (2)	Muito melhor (2)	Equivalente (0)

```
sapevo_ranking("Panificadora",
  c("Renato", "Branca"),
  c("Gelopar", "Metalfrio", "Frilux"),
  c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2criterio1=c(0,-1,-2, 1,0,-2, 2,2,0),
```

FIGURA 11 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 2

Primeira
linha da
Tabela 5

Segunda
linha da
Tabela 5

Terceira
linha da
Tabela 5



Decisor 1: Renato

Critério 2: Custo Aquisição

Tabela 6 – Comparação pareada dos critérios

vs	Gelopar	Metalfrio	Frilux
Gelopar	Equivalente (0)	Absolutamente pior (-3)	Muito pior (-2)
Metalfrio	Absolutamente Melhor (3)	Equivalente (0)	pior (-1)
Frilux	Muito melhor (2)	Melhor(1)	Equivalente (0)

```
sapevo_ranking("Panificadora",
  c("Renato", "Branca"),
  c("Gelopar", "Metalfrio", "Frilux"),
  c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2criterio1=c(0,-1,-2, 1,0,-2, 2,2,0),
  #criterio2
  vetor_notas_decisor1criterio2=c(0,-3,-2, 3,0,-1, 2,1,0),
```

FIGURA 12 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 1

Primeira linha da Tabela 6	Segunda linha da Tabela 6	Terceira linha da Tabela 6
----------------------------------	---------------------------------	----------------------------------

Decisor 2: Branca

Critério 2: Custo Aquisição

Tabela 7 – Comparação pareada dos critérios

vs	Gelopar	Metalfrio	Frilux
Gelopar	Equivalente (0)	Absolutamente pior (-3)	Absolutamente pior (-3)
Metalfrio	Absolutamente Melhor (3)	Equivalente (0)	Equivalente (0)



Frilux	Absolutamente Melhor (3)	Equivalente (0)	Equivalente (0)
---------------	--------------------------	-----------------	-----------------

```
sapevo_ranking("Panificadora",
  c("Renato","Branca"),
  c("Gelopar","Metalfrio","Frilux"),
  c("Capacidade","Custo Aquisição","Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2criterio1=c(0,-1,-2, 1,0,-2, 2,2,0),
  #criterio2
  vetor_notas_decisor1criterio2=c(0,-3,-2, 3,0,-1, 2,1,0),
  vetor_notas_decisor2criterio2=c(0,-3,-3, 3,0,0, 3,0,0),
  #criterio3
  vetor_notas_decisor1criterio3=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2criterio3=c(0,-2,-1, 2,0,0, 1,0,0))
```

FIGURA 13 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 2

Decisor 1: Renato

Critério 3: Custo Manutenção

Primeira
linha da
Tabela 7

Segunda
linha da
Tabela 7

Terceira
linha da
Tabela 7

Tabela 8 – Comparação pareada dos critérios

vs	Gelopar	Metalfrio	Frilux
Gelopar	Equivalente (0)	Muito pior (-2)	pior (-1)
Metalfrio	Muito melhor (2)	Equivalente (0)	Equivalente (0)
Frilux	Melhor (1)	Equivalente (0)	Equivalente (0)

```
sapevo_ranking("Panificadora",
  c("Renato","Branca"),
  c("Gelopar","Metalfrio","Frilux"),
  c("Capacidade","Custo Aquisição","Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2criterio1=c(0,-1,-2, 1,0,-2, 2,2,0),
  #criterio2
  vetor_notas_decisor1criterio2=c(0,-3,-2, 3,0,-1, 2,1,0),
  vetor_notas_decisor2criterio2=c(0,-3,-3, 3,0,0, 3,0,0),
  #criterio3
  vetor_notas_decisor1criterio3=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2criterio3=c(0,-2,-1, 2,0,0, 1,0,0))
```

FIGURA 14 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 1



Primeira
linha da
Tabela 8

Segunda
linha da
Tabela 8

Terceira
linha da
Tabela 8

Decisor 2: Branca

Critério 3: Custo Manutenção

Tabela 9 – Comparação pareada dos critérios

vs	Gelopar	Metalfrio	Frilux
Gelopar	Equivalente (0)	pior (-1)	Muito pior (-2)
Metalfrio	Melhor (1)	Equivalente (0)	Melhor (1)
Frilux	Muito melhor (2)	pior (-1)	Equivalente (0)

```
sapevo_ranking("Panificadora",
  c("Renato", "Branca"),
  c("Gelopar", "Metalfrio", "Frilux"),
  c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2criterio1=c(0,-1,-2, 1,0,-2, 2,2,0),
  #criterio2
  vetor_notas_decisor1criterio2=c(0,-3,-2, 3,0,-1, 2,1,0),
  vetor_notas_decisor2criterio2=c(0,-3,-3, 3,0,0, 3,0,0),
  #criterio3
  vetor_notas_decisor1criterio3=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2criterio3=c(0,-1,-2, 1,0,1, 2,-1,0)
)
```

FIGURA 15 – Inserindo o Vetor associado as respostas pareadas em relação a importância dos critérios pelo Decisor 2

Primeira
linha da
Tabela 9

Segunda
linha da
Tabela 9

Terceira
linha da
Tabela 9

Selecionar todas as linhas e pressionar **ctrl+enter**, ou o **botão Run**, mostrados nas



figuras 16 e 17 abaixo:



FIGURA 16 – Run

```
sapevo_ranking("Panificadora",
  c("Renato", "Branca"),
  c("Gelopar", "Metalfrio", "Frilux"),
  c("Capacidade", "Custo Aquisição", "Custo Manutenção"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #criterio1
  vetor_notas_decisor1criterio1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2criterio1=c(0,-1,-2, 1,0,-2, 2,2,0),
  #criterio2
  vetor_notas_decisor1criterio2=c(0,-3,-2, 3,0,-1, 2,1,0),
  vetor_notas_decisor2criterio2=c(0,-3,-3, 3,0,0, 3,0,0),
  #criterio3
  vetor_notas_decisor1criterio3=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2criterio3=c(0,-1,-2, 1,0,1, 2,-1,0)
```

FIGURA 17 – Selecionando as linhas e rodando

Vamos ver o Output da função sapevo_ranking() na figura 18:

```
-----O nome do seu projeto ?: Panificadora -----
----- Alternativas do seu projeto: Gelopar Metalfrio Frilux -----
-----Crit?rios do seu projeto: Capacidade Custo Aquisição Custo Manutenção -----
-----Os pesos do m?todo sapevo-m s?o: 2 1.075 0.01075 -----

-----A matriz avalia??o do seu projeto ?: -----
          Capacidade  Custo Aquisição  Custo Manutenção
Gelopar      0.0000000          0.000          0.0
Metalfrio    0.4857143          1.875          2.0
Frilux       2.0000000          2.000          1.6

-----A pontua??o dos crit?rios foi: -----
          Capacidade  Custo Aquisição  Custo Manutenção
          2.485714          3.875000          3.600000

-----O ranking do seu projeto ?: -----
          grau obtido ranking
Frilux          6.167200          1
Metalfrio        3.008554          2
Gelopar          0.000000          3
```

FIGURA 18 –Output sapevo_ranking()

Ratificando o que havia sido proposto por Santos et al. (2019), os algoritmos sapevo_pesos() e sapevo_ranking() representam **fielmente** o método ordinal SAPEVO-M e **concluiu** que a opção do **Equipamento freezer Frilux** seria a **melhor ordenada em relação** às demais, para ser **objeto de aquisição** pela **panificadora**, segundo seus próprios gerentes.

OBS2:

Vamos supor que temos **3 decisores**, **3 alternativas** e **3 critérios**, a inclusão de um **Decisor Adicional**, acarretará a **adição de um vetor amais de input** no sapevo_pesos() e



acarretará um vetor amais pra cada critério que tivermos no `sapevo_ranking()`, conforme ilustrado a seguir nas figuras 19 e 20, respectivamente:

```
sapevo_pesos("Projeto",
  c("Decisor1","Decisor2","Decisor3"),
  c("Alternativa1","Alternativa2","Alternativa3"),
  c("Critério1","Critério2","Critério3"),
  #vetor_notas_decisores(critérios)
  c(0,1,2, -1,0,3, -2,-3,0),
  c(0,2,1, -2,0,1, -1,-1,0)
  #vetor adicional pela inclusão do Decisor 3
  c(0,1,1, -1,0,1, -1,-1,0)
)
```

FIGURA 19 – Inclusão de um decisor adicional (`sapevo_pesos()`)

```
sapevo_ranking("Projeto",
  c("Decisor1","Decisor2","Decisor3"),
  c("Alternativa1","Alternativa2","Alternativa3"),
  c("Critério1","Critério2","Critério3"),
  vetor_peso,
  #vetor_notas_decisores_criterios(alternativas)
  #critério1
  vetor_notas_decisor1critério1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2critério1=c(0,-1,-2, 1,0,-2, 2,2,0),
  #vetor adicional pela inclusão do Decisor 3 critério 1
  vetor_notas_decisor3critério1c(0,1,1, -1,0,1, -1,-1,0)

  #critério2
  vetor_notas_decisor1critério2=c(0,-3,-2, 3,0,-1, 2,1,0),
  vetor_notas_decisor2critério2=c(0,-3,-3, 3,0,0, 3,0,0),
  #vetor adicional pela inclusão do Decisor 3 critério 2
  vetor_notas_decisor3critério2c(0,1,1, -1,0,1, -1,-1,0)

  #critério3
  vetor_notas_decisor1critério3=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2critério3=c(0,-1,-2, 1,0,1, 2,-1,0)
  #vetor adicional pela inclusão do Decisor 3 critério3
  vetor_notas_decisor3critério3c(0,1,1, -1,0,1, -1,-1,0)
)
```

FIGURA 20 – Inclusão de um decisor adicional (`sapevo_ranking()`)

OBS3:



Vamos supor que temos **2 decisores, 4 alternativas e 3 critérios**, a inclusão de uma **Alternativa Adicional**, não modificará o input no `sapevo_pesos()` e acarretará um **vetor amais pra cada alternativa** que tivermos no `sapevo_ranking()`, conforme ilustrado a seguir nas figuras 19 e 20, respectivamente:

Tabela 10 – Comparação pareada dos critérios

vs	Alternativa1	Alternativa2	Alternativa3	Alternativa4
Alternativa1	Equivalente (0)	pior (-1)	Muito pior (-2)	Comparação _{nova1}
Alternativa2	Melhor (1)	Equivalente (0)	Melhor (1)	Comparação _{nova2}
Alternativa3	Muito melhor (2)	pior (-1)	Equivalente (0)	Comparação _{nova3}
Alternativa4	Comparação _{nova_1}	Comparação _{nova_2}	Comparação _{nova_3}	Equivalente (0)

Ou seja: A **cada alternativa nova**, geraremos um **novo vetor**, representado na **tabela 10** pela **quarta linha** e também uma **nova entrada**, para **cada vetor já existente**, representado pela **coluna que tem como rótulo a alternativa 4**.

A **figura 21** exemplifica a **inclusão de uma alternativa**:

```
sapevo_ranking("Projeto",
c("Decisor1","Decisor2","Decisor3"),
c("Alternativa1","Alternativa2","Alternativa3","Alternativa4"),
c("Critério1","Critério2","Critério3"),
vetor_peso,
#vetor_notas_decisores_criterios(alternativas)
#critério1
vetor_notas_decisor1critério1=c(0,0,-2, Comparaçãonova1 , 0,0,-1, Comparaçãonova2 , 2,1,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0),
vetor_notas_decisor2critério1=c(0,-1,-2, Comparaçãonova1 , 1,0,-2, Comparaçãonova2 , 2,2,0, Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0),
#critério2
vetor_notas_decisor1critério2=c(0,-3,-2, Comparaçãonova1 , 3,0,-1, Comparaçãonova2 , 2,1,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0),
vetor_notas_decisor2critério2=c(0,-3,-3, Comparaçãonova1 , 3,0,0, Comparaçãonova2 , 3,0,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0),
#critério3
vetor_notas_decisor1critério3=c(0,-2,-1, Comparaçãonova1 , 2,0,0, Comparaçãonova2 , 1,0,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0),
vetor_notas_decisor2critério3=c(0,1,1, Comparaçãonova1 , -1,0,1, Comparaçãonova2 , -1,-1,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0)
)
```

FIGURA 21 – Inclusão de uma alternativa adicional (`sapevo_ranking()`)

```
sapevo_ranking("Projeto",
c("Decisor1","Decisor2","Decisor3"),
c("Alternativa1","Alternativa2","Alternativa3","Alternativa4"),
c("Critério1","Critério2","Critério3"),
vetor_peso,
#vetor_notas_decisores_criterios(alternativas)
#critério1
vetor_notas_decisor1critério1=c(0,0,-2, 1 , 0,0,-1, 2 , 2,1,0 , 3 , -1,-2,-3,0),
vetor_notas_decisor2critério1=c(0,-1,-2, 1 , 1,0,-2, 2 , 2,2,0, 3 , -1,-2,-3,0),
#critério2
vetor_notas_decisor1critério2=c(0,-3,-2, 1 , 3,0,-1, 2 , 2,1,0 , 3 , -1,-2,-3,0),
vetor_notas_decisor2critério2=c(0,-3,-3, 1 , 3,0,0, 2 , 3,0,0 , 3 , -1,-2,-3,0),
#critério3
vetor_notas_decisor1critério3=c(0,-2,-1, 1 , 2,0,0, 2 , 1,0,0 , 3 , -1,-2,-3,0),
vetor_notas_decisor2critério3=c(0,1,1, 1 , -1,0,1, 2 , -1,-1,0 , 3 , -1,-2,-3,0)
)
```



FIGURA 22 – Inclusão de uma alternativa adicional Exemplo (sapevo_ranking())

```

-----O nome do seu projeto?: Projeto -----
----- Alternativas do seu projeto: Alternativa1 Alternativa2 Alternativa3 Alternativa4 -----
-----Crit?rios do seu projeto: Crit?rio1 Crit?rio2 Crit?rio3 -----
-----Os pesos do m?todo sapevo-m s?o: 2 1.075 0.01075 -----

-----A matriz avalia??o do seu projeto?: -----
          Crit?rio1 Crit?rio2 Crit?rio3
Alternativa1 0.724359      0.25 1.400000
Alternativa2 1.121795      1.75 1.888889
Alternativa3 2.000000      2.00 1.777778
Alternativa4 0.000000      0.00 0.000000

-----A pontua??o dos crit?rios foi: -----
Crit?rio1 Crit?rio2 Crit?rio3
3.846154 4.000000 5.066667

-----O ranking do seu projeto?: -----
          grau obtido ranking
Alternativa3 6.169111      1
Alternativa2 4.145145      2
Alternativa1 1.732518      3
Alternativa4 0.000000      4

```

FIGURA 23 –Output Exemplo (sapevo_ranking())

OBS4:

Vamos supor que temos **2 decisores, 3 alternativas e 4 critérios**, a inclusão de um **Critério Adicional**, geraremos uma **nova entrada**, para **cada vetor já existente** no sapevo_pesos() e **acarretará um vetor a mais para cada decisor** que tivermos no sapevo_ranking(), conforme ilustrado a seguir nas figuras 24 e 25, respectivamente:

Tabela 11 – Comparação pareada dos critérios

vs	Critério1	Critério2	Critério3	Critério4
Critério1	Equivalente (0)	pior (-1)	Muito pior (-2)	Comparaçãonova1
Critério2	Melhor (1)	Equivalente (0)	Melhor (1)	Comparaçãonova2
Critério3	Muito melhor (2)	pior (-1)	Equivalente (0)	Comparaçãonova3
Critério4	Comparaçãonova_-1	Comparaçãonova_-2	Comparaçãonova_-3	Equivalente (0)

Ou seja: A **cada critério novo**, geraremos um **novo vetor**, representado na **tabela 11** pela **quarta linha** e também uma **nova entrada**, para **cada vetor já existente**, representado pela **coluna que tem como rótulo o critério 4**.



```
sapevo_pesos("Projeto",
  c("Decisor1","Decisor2"),
  c("Alternativa1","Alternativa2","Alternativa3"),
  c("Critério1","Critério2","Critério3","Critério4"),
  #vetor_notas_decisores(critérios)
  c(0,1,2, Comparaçãonova1 , -1,0,3, Comparaçãonova2 , -2,-3,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0 ),
  c(0,2,1, Comparaçãonova1 , -2,0,1, Comparaçãonova2 , -1,-1,0 , Comparaçãonova3 , Comparaçãonova_1,Comparaçãonova_2,Comparaçãonova_3,0 )
)
```

FIGURA 24 – Inclusão de um critério adicional (sapevo_pesos())

```
sapevo_pesos("Projeto",
  c("Decisor1","Decisor2"),
  c("Alternativa1","Alternativa2","Alternativa3"),
  c("Critério1","Critério2","Critério3","Critério4"),
  #vetor_notas_decisores(critérios)
  c(0,1,2, 1 , -1,0,3, 2 , -2,-3,0 , 3 , -1,-2,-3,0 ),
  c(0,2,1, 1 , -2,0,1, 2 , -1,-1,0 , 3 , -1,-2,-3,0 )
)
```

FIGURA 25 – Inclusão de um critério adicional Exemplo (sapevo_pesos())

```
-----O nome do seu projeto: Projeto -----
-----As alternativas do seu projeto: Alternativa1 Alternativa2 Alternativa3 -----
-----Os Critérios do seu projeto: Critério1 Critério2 Critério3 Critério4 -----
-----Os pesos do metodo sapevo-M: 2 1.7 1.1 0.017 -----
```

FIGURA 26 –Output Exemplo sapevo_pesos()



```
sapevo_ranking("Projeto",
  c("Decisor1","Decisor2"),
  c("Alternativa1","Alternativa2","Alternativa3"),
  c("Critério1","Critério2","Critério3","Critério4"),
  vetor_peso,
  #vetor_notas_decisores_critérios(alternativas)
  #critério1
  vetor_notas_decisor1critério1=c(0,0,-2, 0,0,-1, 2,1,0),
  vetor_notas_decisor2critério1=c(0,-1,-2, 1,0,-2, 2,2,0),

  #critério2
  vetor_notas_decisor1critério2=c(0,-3,-2, 3,0,-1, 2,1,0),
  vetor_notas_decisor2critério2=c(0,-3,-3, 3,0,0, 3,0,0),

  #critério3
  vetor_notas_decisor1critério3=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2critério3=c(0,-1,-2, 1,0,1, 2,-1,0),

  #vetor adicional pela inclusão do Critério4
  vetor_notas_decisor1critério4=c(0,-2,-1, 2,0,0, 1,0,0),
  vetor_notas_decisor2critério4=c(0,-1,-2, 1,0,1, 2,-1,0)
)
```

FIGURA 27 – Inclusão de um critério adicional (sapevo_ranking())

```
-----0 nome do seu projeto ?: Projeto -----
----- Alternativas do seu projeto: Alternativa1 Alternativa2 Alternativa3 -----
-----Crit?rios do seu projeto: Critério1 Critério2 Critério3 Critério4 -----
-----Os pesos do m?todo sapevo-m s?o: 2 1.7 1.1 0.017 -----

-----A matriz avalia??o do seu projeto ?: -----
          Critério1 Critério2 Critério3 Critério4
Alternativa1 0.0000000 0.000 0.0 0.0
Alternativa2 0.4857143 1.875 2.0 2.0
Alternativa3 2.0000000 2.000 1.6 1.6

-----A pontua??o dos crit?rios foi: -----
Critério1 Critério2 Critério3 Critério4
2.485714 3.875000 3.600000 3.600000

-----0 ranking do seu projeto ?: -----
          grau obtido ranking
Alternativa3 9.187200 1
Alternativa2 6.392929 2
Alternativa1 0.000000 3
```

FIGURA 23 –Output Exemplo (sapevo_ranking())

Tabela 12 – Tabela resumo da inclusão de **Alternativas**, **Critérios** e **Decisores**

Algoritmos	+k alternativas	+k Critérios	+k Decisores
sapevo_pesos()	Não modificará o input na função	k novas entradas, para cada linha já	+K vetores a mais de input na função



		existente e mais k entradas para cada linha adicional na Tabela 11	
sapevo_ranking()	k novas entradas, para cada linha já existente e mais k entradas para cada linha adicional na Tabela 10	+K vetores a mais de input na função para cada Decisor	+K vetores a mais de input na função para cada Critério

3.4. Utilizando a função sapevo_pesos() com inputs automáticos da função sapevo_input_pesos

Como as bibliotecas já estão carregadas como foi ensinado no módulo 3.1, agora iremos utilizar a função sapevo_input_pesos() para gerar o input automaticamente da função sapevo_pesos, para isso teremos que digitar na sequência a seguir:

- 1) sapevo_input_pesos(), selecionar e apertar Run ou ctrl+Enter

```
sapevo_input_pesos()
```

- 2) Nome do projeto:

```
> sapevo_input_pesos()
Qual o nome do projeto? Panificadora
```

- 3) Decisores do projeto:

```
> sapevo_input_pesos()
Qual o nome do projeto? Panificadora
Quem sao os decisores? Renato,Branca
```

- 4) Alternativas do projeto:

```
> sapevo_input_pesos()
Qual o nome do projeto? Panificadora
Quem sao os decisores? Renato,Branca
Quais sao as alternativas? Gelopar,Metalfrio,Frilux
```

- 5) Critérios do projeto:

```
> sapevo_input_pesos()
Qual o nome do projeto? Panificadora
Quem sao os decisores? Renato,Branca
Quais sao as alternativas? Gelopar,Metalfrio,Frilux
Quais são os critérios do projeto? Capacidade,Custo Aquisição,Custo Manutenção
```

- 6) Avaliação de Critérios: Decisor 1:Renato

```
Avaliar critérios: Renato
Capacidade vs Custo Aquisição

Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
1
```

```
Avaliar critérios: Renato
Capacidade vs Custo Manutenção

Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
2
```

- 7)



8) Avaliar critérios: Renato
Custo Aquisição vs Custo Manutenção
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
3
O nome do seu projeto?: Panificadora
Alternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[1] 0 1 2 -1 0 3 -2 -3 0

9) Avaliação de Critérios: Decisor 2:Branca

Avaliar critérios: Branca
Capacidade vs Custo Aquisição
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
2

10) Avaliar critérios: Branca
Capacidade vs Custo Manutenção
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
1

11) Avaliar critérios: Branca
Custo Aquisição vs Custo Manutenção
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
1
O nome do seu projeto?: Panificadora
Alternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[1] 0 1 2 -1 0 3 -2 -3 0
[[2]]
[1] 0 2 1 -2 0 1 -1 -1 0

No retângulo laranja se encontram os inputs para o sapevo_pesos() gerado automaticamente, foi gerado também um vetor internamente (**Figura 24**) os objetos: vetor_final, alternativas, critérios e projeto, que vão ser usados de input no sapevo_pesos(), conforme a **figura 25**

Global Environment	
Data	
vetor_final	List of 2
: num [1:9]	0 1 2 -1 0 3 -2 -3 0
: num [1:9]	0 2 1 -2 0 1 -1 -1 0
Values	
alternativas	chr [1:3] "Gelopar" "Metalfrio" "Frilux"
critérios	chr [1:3] "Capacidade" "Custo Aquisição" ...
decisores	chr [1:2] "Renato" "Branca"
projeto	"Panificadora"

FIGURA 24 –Objetos internos criados para input no (sapevo_pesos())



```
sapevo_pesos(projeto,
              decisores,
              alternativas,
              criterios,
              #vetor_notas_decisores(critérios)
              vetor_final[[1]],
              vetor_final[[2]]
            )
```

FIGURA 25 –input automatico no (sapevo_pesos())

```
-----O nome do seu projeto: Panificadora -----
-----As alternativas do seu projeto: Gelopar Metalfrio Frilux -----
-----Os Criterios do seu projeto: Capacidade Custo Aquisição Custo Manutenção -----
-----Os pesos do metodo sapevo-M: 2 1.075 0.01075 -----
```

FIGURA 26 –Output (sapevo_pesos())

3.4. Utilizando a função sapevo_ranking() com inputs automáticos da função sapevo_input_ranking()

Como as bibliotecas já estão carregadas como foi ensinado no módulo 3.1, agora iremos utilizar a função sapevo_input_ranking() para gerar o input automaticamente da função sapevo_ranking, para isso teremos que digitar na sequência a seguir:

1) sapevo_input_ranking(), selecionar e apertar Run ou ctrl+Enter `sapevo_input_ranking()`

2) Nome projeto, decisores, alternativas e critérios, conforme o 1) a 5) sapevo_input_pesos()

3) Avaliação de Alternativas: Decisor 1:Renato e Critério: Capacidade

```
Avaliar alternativas: Renato Capacidade
Gelopar vs Metalfrio
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente
elhor (3)
0
```

4) Avaliar alternativas: Renato Capacidade

```
Gelopar vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente m
elhor (3)
-2
```

5) Avaliar alternativas: Renato Capacidade

```
Metalfrio vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente
elhor (3)
-1
O nome do seu projeto?: Panificadora
Alternativas do seu projeto: Gelopar Metalfrio Frilux
Criterios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
list()
[[2]]
[1] 0 0 -2 0 0 -1 2 1 0
```



6) Avaliação de Alternativas: Decisor 1: Renato e Critério: Custo Aquisição

7) Avaliar alternativas: Renato Custo Aquisição
Gelopar vs Metalfrio
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente elhor (3)
-1

8) Avaliar alternativas: Renato Custo Aquisição
Gelopar vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente elhor (3)
-2

9) Avaliar alternativas: Renato Custo Aquisição
Metalfrio vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente elhor (3)
-2
O nome do seu projeto?: Panificadora
Aternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[[1]][[1]]
list()
[[1]][[2]]
[1] 0 0 -2 0 0 -1 2 1 0
[[2]]
[1] 0 -1 -2 1 0 -2 2 2 0

10) Avaliação de Alternativas: Decisor 1: Renato e Critério: Custo Manutenção

11) Avaliar alternativas: Renato Custo Manutenção
Gelopar vs Metalfrio
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente elhor (3)
-3

12) Avaliar alternativas: Renato Custo Manutenção
Gelopar vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente m elhor (3)
-2

13) Avaliar alternativas: Renato Custo Manutenção
Metalfrio vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente elhor (3)
-1
O nome do seu projeto?: Panificadora
Aternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[[1]][[1]]
[[1]][[1]][[1]]
list()
[[1]][[1]][[2]]
[1] 0 0 -2 0 0 -1 2 1 0
[[1]][[2]]
[1] 0 -1 -2 1 0 -2 2 2 0
[[2]]
[1] 0 -3 -2 3 0 -1 2 1 0

14) Avaliação de Alternativas: Decisor 2: Branca e Critério: Capacidade



15) Avaliar alternativas: Branca Capacidade
Gelopar vs Metalfrio
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
-3

16) Avaliar alternativas: Branca Capacidade
Gelopar vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
-3

17) Avaliar alternativas: Branca Capacidade
Metalfrio vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
0
O nome do seu projeto?: Panificadora
Alternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[[1]][[1]]
[[1]][[1]][[1]]
[[1]][[1]][[1]][[1]]
list()
[[1]][[1]][[1]][[2]]
[1] 0 0 -2 0 0 -1 2 1 0
[[1]][[1]][[2]]
[1] 0 -1 -2 1 0 -2 2 2 0
[[1]][[2]]
[1] 0 -3 -2 3 0 -1 2 1 0
[[2]]
[1] 0 -3 -3 3 0 0 3 0 0

18) Avaliação de Alternativas: Decisor 2: Branca e Critério: Custo de Aquisição

Avaliar alternativas: Branca Custo Aquisição
Gelopar vs Metalfrio
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
-2

19) Avaliar alternativas: Branca Custo Aquisição
Gelopar vs Frilux
Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente melhor (3)
-1



```

Avaliar alternativas: Branca Custo Aquisição

Metalfrio vs Frilux

Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente
melhor (3)
0
O nome do seu projeto?: Panificadora
Alternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[[1]][[1]]
[[1]][[1]][[1]]
[[1]][[1]][[1]][[1]]
[[1]][[1]][[1]][[1]][[1]]
[[1]][[1]][[1]][[1]][[1]]
list()

[[1]][[1]][[1]][[1]][[2]]
[1] 0 0 -2 0 0 -1 2 1 0

[[1]][[1]][[1]][[2]]
[1] 0 -1 -2 1 0 -2 2 2 0

[[1]][[1]][[2]]
[1] 0 -3 -2 3 0 -1 2 1 0

[[1]][[2]]
[1] 0 -3 -3 3 0 0 3 0 0

[[2]]
[1] 0 -2 -1 2 0 0 1 0 0
20)

```

21) Avaliação de Alternativas: Decisor 2: Branca e Critério: Custo de Manutenção

```

Avaliar alternativas: Branca Custo Manutenção

Gelopar vs Metalfrio

Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente m
elhor (3)
-1
22)

```

```

Avaliar alternativas: Branca Custo Manutenção

Gelopar vs Frilux

Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente m
elhor (3)
-2
23)

```



```

Avaliar alternativas: Branca Custo Manutenção

Metalfrio vs Frilux

Diga uma Nota comparativa : Abolutamente pior (-3), Muito pior (-2), Pior (-1), Equivalente (0), Melhor (1), Muito melhor (2), Absolutamente m
elhor (3)
1
O nome do seu projeto?: Panificadora
Alternativas do seu projeto: Gelopar Metalfrio Frilux
Critérios do seu projeto: Capacidade Custo Aquisição Custo Manutenção
O input para a funcao sapevo_pesos é o vetor:
[[1]]
[[1]][[1]]
[[1]][[1]][[1]]
[[1]][[1]][[1]][[1]]
[[1]][[1]][[1]][[1]][[1]]
[[1]][[1]][[1]][[1]][[1]][[1]]
[[1]][[1]][[1]][[1]][[1]][[1]][[1]]
list()

[[1]][[1]][[1]][[1]][[1]][[1]][[2]]
[1] 0 0 -2 0 0 -1 2 1 0

[[1]][[1]][[1]][[1]][[2]]
[1] 0 -1 -2 1 0 -2 2 2 0

[[1]][[1]][[1]][[2]]
[1] 0 -3 -2 3 0 -1 2 1 0

[[1]][[1]][[2]]
[1] 0 -3 -3 3 0 0 3 0 0

[[1]][[2]]
[1] 0 -2 -1 2 0 0 1 0 0

[[2]]
[1] 0 -1 -2 1 0 1 2 -1 0

```

No retângulo **laranja** se encontram os inputs para o sapevo_ranking() gerado automaticamente, foi gerado também um vetor internamente (**Figura 27**) os objetos: vetor_final, alternativas, critérios e projeto, que vão ser usados de input no sapevo_pesos(), conforme a **figura 28**

```

vetor_final      List of 2
:List of 2
..$ :List of 2
.. ..$ :List of 2
.. .. ..$ :List of 2
.. .. .. ..$ :List of 2
.. .. .. .. ..$ : list()
.. .. .. .. ..$ : num [1:9] 0 0 -2 0 0 -1 2 1 0
.. .. .. ..$ : num [1:9] 0 -1 -2 1 0 -2 2 2 0
.. .. ..$ : num [1:9] 0 -3 -2 3 0 -1 2 1 0
.. ..$ : num [1:9] 0 -3 -3 3 0 0 3 0 0
..$ : num [1:9] 0 -2 -1 2 0 0 1 0 0
: num [1:9] 0 -1 -2 1 0 1 2 -1 0

Values
alternativas    chr [1:3] "Gelopar" "Metalfrio" "Frilux"
critérios       chr [1:3] "Capacidade" "Custo Aquisição" ...
decisores       chr [1:2] "Renato" "Branca"
projeto         "Panificadora"
vetor_peso      Named num [1:3] 2 1.075 0.0108

```

FIGURA 27 –Objetos internos criados para input no (sapevo_ranking())



```
#gerado automaticamente
sapevo_ranking(projeto,
               decisores,
               alternativas,
               criterios,
               vetor_peso,
               #vetor_notas_decisores_criterios(alternativas)
               #criterio1
               vetor_notas_decisor1criterio1=vetor_final[[1]][[1]][[1]][[1]][[1]][[2]],
               vetor_notas_decisor2criterio1=vetor_final[[1]][[1]][[1]][[1]][[2]],
               #criterio2
               vetor_notas_decisor1criterio2=vetor_final[[1]][[1]][[1]][[2]],
               vetor_notas_decisor2criterio2=vetor_final[[1]][[1]][[2]],
               #criterio3
               vetor_notas_decisor1criterio3=vetor_final[[1]][[2]],
               vetor_notas_decisor2criterio3=vetor_final[[2]]
)
```

FIGURA 28 –input automático no (sapevo_pesos())

```
-----O nome do seu projeto ?: Panificadora -----
----- Alternativas do seu projeto: Gelopar Metalfrio Frilux -----
-----Crit?rios do seu projeto: Capacidade Custo Aquisição Custo Manutenção -----
-----Os pesos do m?todo sapevo-m s?o: 2 1.075 0.01075 -----

-----A matriz avalia??o do seu projeto ?: -----
          Capacidade Custo Aquisição Custo Manutenção
Gelopar    0.0000000          0.000          0.0
Metalfrio  0.4857143          1.875          2.0
Frilux     2.0000000          2.000          1.6

-----A pontua??o dos crit?rios foi: -----
          Capacidade  Custo Aquisição  Custo Manutenção
          2.485714          3.875000          3.600000

-----O ranking do seu projeto ?: -----
          grau obtido ranking
Frilux      6.167200          1
Metalfrio   3.008554          2
Gelopar     0.000000          3
```

FIGURA 29 – Output automático no (sapevo_ranking())

4.0 Conclusão

Foram desenvolvidos quatro algoritmos na linguagem R, com um input manual ou automático, para facilitar o entendimento e para que problemas complexos de tomada de decisões, envolvendo multicritérios e multidecisores, fossem solucionados, rapidamente, abarcando tanto as análises qualitativas, quanto quantitativas, de forma a dirimir conflitos e melhorar as tomadas de decisões, sejam elas advindas de qualquer esfera pública ou privada, disseminando assim também a aplicação dos métodos da Pesquisa Operacional, tão úteis e que ainda não muito utilizados em larga escala no Brasil.

Os mesmos foram alocados no Github, para serem utilizados publicamente sob a licença MIT, cabe ressaltar que os algoritmos desenvolvidos, tem limitações do número de Decisores e Critérios, a serem inclusos, respectivamente (Até 10 Decisores, Até 10 Critérios), já as alternativas, quantas forem desejáveis e para sejam gerados os inputs automaticamente, as funções tem limitações de até (5 alternativas e 5 critérios)



Esse tutorial teve como objetivo ensinar, passo a passo, como utilizar os algoritmos desenvolvidos para operacionalizar o Sapevo-M, método ordinal multicritério para múltiplos decisores que não só cria um ranking das alternativas desejadas, como também uma importância relativa de cada uma delas e de seus critérios, para que os decisores se utilizem e tomem como subsídio para tomada de decisões.



REFERÊNCIAS

- [1] GOMES, L. F. A. M., MURY, A. R., GOMES, C. F. S. (1997). **Multicriteria Ranking with Ordinal Data. Systems Analysis Modelling Simulation.** (1997).
- [2] Gomes, L. F. A. M., e Gomes, C. F. S. **Princípios e métodos para a tomada de decisão: Enfoque multicritério** (6a ed.). São Paulo: Atlas. (2019).
- [3] Teixeira, L. F. H. S. B., Santos, M. e Gomes, C. F. S. SapevoWeb Software (v.1). (2018). Disponível em: <<http://www.sapevoweb.com>> Acesso em: 30/04/ 2019.
- [4] SANTOS, M.; OLIVEIRA, N. C.; OLIVEIRA, P. F. C.; GOMES, C. F. S. Aplicação do Método multicritério SAPEVO-M na seleção de equipamentos: estudo de caso em uma panificadora do RJ. In. **Anais do XIX Simpósio de Pesquisa Operacional e Logística da Marinha – SPOLM 2019.** Rio de Janeiro / RJ.