

Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Estatística

Manual do Usuário:
Programas para realização da
Análise Hierárquica

Autores: Giselle Silva de Carvalho (*)

Profa. Sueli Aparecida Mingoti (**)

(*) Bacharelada em Estatística pela UFMG.

(**) Profa. Adjunta do Depto. de Estatística da UFMG

Novembro de 2005

Sumário

<u>Parte 1: Descrição da técnica de Análise Hierárquica</u>	3
<u>Introdução</u>	3
<u>A Análise Hierárquica</u>	4
<u>Parte 2: Utilização dos Programas</u>	8
<u>Entrada de dados:</u>	9
<u>Execução dos programas e saídas</u>	11
<u>Referências bibliográficas</u>	15
<u>ANEXO</u>	16

PARTE 1: DESCRIÇÃO DA TÉCNICA DE ANÁLISE HIERÁRQUICA

Introdução

Atualmente tem-se presenciado um uso crescente da técnica de Estatística Multivariada chamada Método de Análise Hierárquica (Saaty, 1991), também conhecida como AHP - Analytical Hierarchy Process. Este método se trata de uma ferramenta de auxílio a tomadas de decisão, que foi proposto recentemente por Thomas L. Saaty e visa reproduzir o raciocínio humano no que diz respeito à comparação de elementos de um conjunto. Sendo assim, essa ferramenta auxilia na comparação de um conjunto de itens - que pode se tratar desde várias marcas de um produto como tênis, por exemplo, até um conjunto de diferentes tipos de investimentos - utilizando a opinião humana, ou seja, uma avaliação subjetiva. Para isto, ele usa uma escala de importância para confrontar os elementos dois a dois. A comparação se dá através de uma ordenação dos itens em questão de acordo com o nível de importância dos mesmos, e dos seus respectivos atributos (características).

Algumas das vantagens deste método é que através dele é possível retratar a opinião de uma determinada pessoa (jugador/decisor) em relação a aspectos não quantificáveis e ainda comparar elementos que inicialmente seriam incomparáveis entre si. Segundo Saaty, “os problemas de tomada de decisão, avaliam e consideram um grande número de elementos que aparentemente não são comparáveis entre si de forma direta. Entretanto se estabelecendo um procedimento que agregue tais elementos segundo propriedades comuns torna-se possível compará-los”. Em contrapartida, um ponto negativo é que os cálculos necessários para aplicação da técnica são longos caso se tenha uma amostra de julgadores razoavelmente grande (maior que 5 já faz com que exista um número inviável de contas para serem feitas manualmente). Além disso, o método AHP não está presente em nenhum software estatístico conhecido. Daí a inspiração para a realização do presente trabalho, que trata da implementação computacional desse método utilizando os softwares MINITAB e S-plus for Windows.

A Análise Hierárquica

Ao se deparar com um problema de comparação entre vários elementos de um conjunto a mente humana cria um processo de hierarquização. Baseado nisto a AHP é um método onde o problema analisado é estruturado hierarquicamente, sendo que no nível mais alto está o objetivo principal do estudo, nos níveis seguintes estão os critérios (propriedades através das quais as alternativas serão avaliadas) e no nível mais baixo estão as alternativas a serem decididas. A partir daí tem-se, de maneira simplificada, a estrutura hierárquica apresentada na Figura 1.

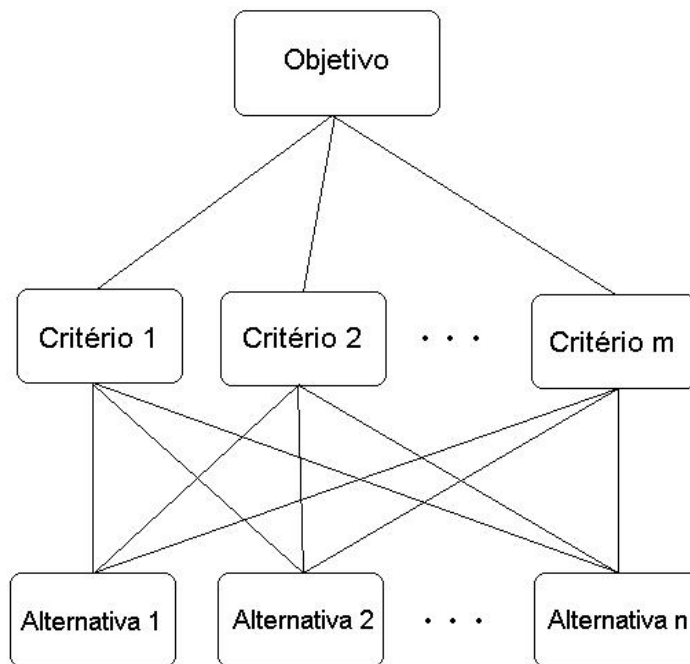


Figura 1: Árvore Hierárquica – modelo simples.

Os critérios citados na figura 1 podem ser entendidos como os atributos (ou características) relativos às alternativas a serem analisadas.

A etapa de estruturação do problema e construção da hierarquia é de extrema importância, pois é a partir daí que será possível realizar o processo de comparação dos critérios e, posteriormente das alternativas.

O método de análise hierárquica permite atribuir pesos onde valores numéricos não podem ser obtidos diretamente. Sendo assim trabalha-se com uma matriz de comparação onde é avaliada a importância de uma característica sobre a outra, utilizando-se para isto uma escala adequada (Saaty, 1991).

A matriz de comparação é construída de forma que o julgador analisa primeiramente os atributos dois a dois de acordo com o nível de importância de um para

com o outro. Depois é feita a mesma análise com os itens (alternativas) em relação a cada atributo separadamente, para no fim obter-se um ranqueamento por ordem de importância das alternativas. A escala de comparação é baseada na tabela de Saaty mostrada a seguir:

Intensidade de importância	Definição	Explicação
1	Igual importância	Duas atividades contribuem igualmente para o objetivo
3	Fraca importância de uma sobre a outra	Experiência e julgamento favorecem ligeiramente uma atividade e relação a outra
5	Essencial ou forte importância	Experiência e julgamento favorecem fortemente uma atividade em relação a outra
7	Importância muito grande ou demonstrada	Uma atividade é fortemente favorecida e sua dominância é demonstrada na prática
9	Absoluta importância	A evidência favorecendo uma atividade sobre a outra é a mais alta ordem de afirmação
2, 4, 6, 8	Valores intermediários entre dois julgamentos sucessivos	Quando se deseja um maior compromisso
Recíprocos dos valores acima	Se uma atividade i tem um dos valores não zero acima quando comparado com a atividade j , então j tem um valor recíproco quando comparado com i .	Uma designação razoável.
Racionais	Razões surgidas da escala	Se a consistência foi forçada para obtenção de n valores numéricos para cobrir a matriz

Figura 2: Tabela de Saaty

A partir dos dados fornecidos pelo julgador é construída a matriz de comparação abaixo:

$$A = \begin{bmatrix} 1 & a_{12} & \cdots & a_{1n} \\ 1/a_{12} & 1 & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 1/a_{1n} & 1/a_{2n} & \cdots & 1 \end{bmatrix}$$

Note que na matriz “A” mostrada acima os elementos obedecem à seguinte regra: $a_{ji} = 1/a_{ij}$, onde a_{ij} são valores reais positivos, i é o índice que representa a linha e j a coluna. Estas características fazem com que a matriz “A” seja denominada como recíproca positiva.

Para a solução do problema seria ideal ter-se as notas/pesos (W_i) atribuídas pelo julgador a cada atributo, contudo a matriz “A” fornece apenas a razão $a_{ij} = W_i/W_j$.

Na prática os valores a_{ij} são baseados em julgamentos subjetivos, assim poderá existir uma diferença entre estes valores e as razões teóricas W_i/W_j . Sendo assim, para resolver este problema e estimar os pesos W_i 's, utiliza-se a seguinte relação:

Existem λ_i , $i = 1, \dots, n$, tal que satisfaça a equação $Ax = \lambda x$, onde x é um vetor de pesos.

Sabe-se que a matriz "A" possui linhas linearmente dependentes, por isso ela fornecerá autovalores iguais a zero, com exceção de um que será o maior valor de λ_i .

Considerando os resultados acima para uma matriz "A" bem estimada, espera-se que pequenas variações nos a_{ij} mantenham o maior autovalor $\lambda_{m\acute{a}x}$ próximo de n . Portanto se faz necessário encontrar o autovetor w de pesos que satisfaça a equação: $Aw = \lambda_{m\acute{a}x} w$. O autovetor w fornecerá a ordenação por nível de importância dos atributos correspondentes à matriz "A", pois estará estimando os pesos reais dados aos atributos ou itens que estão sendo comparados.

Como já foi dito anteriormente as comparações são de caráter subjetivo, o que gera a necessidade de avaliar a proximidade entre $\lambda_{m\acute{a}x}$ e n . Para tal utiliza-se a razão de consistência (RC), que é calculada da seguinte forma:

$$RC = \frac{IC}{CR}$$

onde, $IC = \frac{(\lambda_{m\acute{a}x} - n)}{(n - 1)}$ e CR é o índice de consistência randômico.

O índice de consistência randômico é proveniente de uma amostra aleatória de 500 matrizes recíprocas positivas de dimensão 11 por 11. Note, na Figura 3, que existe uma limitação no número de itens ou atributos a serem comparados uma vez que a tabela fornece valores até n igual a 9.

n	1	2	3	4	5	6	7	8	9
CR Randômico	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45

Figura 3 - Tabela para se verificar a consistência

Considera-se uma matriz consistente quando o valor da razão de consistência é menor que 0,10. Caso contrário recomenda-se uma revisão da matriz de comparação.

Se a consistência da matriz de comparação dos atributos for confirmada, o próximo passo é verificar qual dos atributos é o mais importante na opinião do julgador. Para isto deve-se estimar o autovetor w , entretanto como é mais conveniente trabalhar com um autovetor normalizado calcula-se o autovetor T que é o autovetor w normalizado pela soma de suas entradas. Sendo assim calcula-se T através da seguinte fórmula:

$$T = \left[\frac{w_i}{\sum_{i=1}^n w_i} \right] \quad \text{onde} \quad w_i = \left(\prod_{j=1}^n a_{ij} \right)^{1/n}$$

Desta forma, o autovalor correspondente é dado por:

$$\lambda_{máx} = T.v$$

onde v é o somatório das colunas da matriz recíproca.

Após a análise dos atributos realiza-se uma comparação das alternativas duas-a-duas em relação a cada atributo. Então, o número de matrizes de comparação das alternativas será igual ao número de atributos, por exemplo: no caso de 4 atributos, ter-se-ia 4 matrizes de comparação. As comparações das alternativas são feitas com base na tabela de Saaty da Figura 2 e gera matrizes com as mesmas características da matriz "A" já citada. Assim o cálculo dos autovetores é realizado de maneira semelhante à mostrada anteriormente, porém agora para cada matriz separadamente.

Por fim para alcançar o objetivo de verificar qual das alternativas é a melhor, ou seja, a mais importante na opinião do julgador deve-se obter a ordenação final das mesmas. Para isso, após realizar a comparação das alternativas forma-se uma matriz com os seus respectivos autovetores em forma de coluna. Essa matriz será multiplicada pelo autovetor T . O resultado dessa multiplicação é um novo vetor onde se terá uma resposta de qual é o melhor item, considerando todos os atributos e o quanto cada um deles está presente nas alternativas que estão sendo comparadas. A melhor alternativa é aquela que possuir maior peso no vetor final.

PARTE 2: UTILIZAÇÃO DOS PROGRAMAS

Todo o processo para a realização da análise hierárquica descrito até agora é para apenas um julgador, ou seja, as matrizes citadas são feitas por uma só pessoa. Contudo, num caso real deseja-se saber a opinião de mais de uma pessoa, por isso é interessante realizar esta análise para vários julgadores. Sendo assim, foram desenvolvidos programas computacionais que realizassem a análise hierárquica para um ou mais decisores, tendo as opções de trabalhar com apenas as matrizes de comparação, ou com a matriz média das matrizes de comparação ou, ainda, com a média das ordenações finais das alternativas.

Para a implementação computacional da técnica em questão foram utilizados os softwares MINITAB e S-Plus for Windows. No MINITAB foi feita uma macro que realiza todo o processo da análise hierárquica para apenas um julgador. Já no S-plus for Windows foram desenvolvidas quatro funções:

1. A primeira realiza todos os cálculos da análise hierárquica apenas para um decisor;
2. A segunda faz os mesmos cálculos para um ou mais julgadores e apresenta os resultados da análise para cada um deles;
3. A terceira função também trabalha com o caso de um ou mais julgadores, porém no caso de vários decisores os cálculos são feitos com base na matriz das médias dos julgamentos. Isto é, tira-se a média das matrizes de comparação, tanto para as comparações dos atributos como das alternativas, gerando uma média das opiniões de todos os julgadores;
4. A última função é semelhante à anterior, entretanto no lugar de tirar a média das matrizes de comparações e realizar a análise com ela, agora se faz toda a análise hierárquica para cada julgador e no fim se calcula a média das ordenações finais das alternativas. Ou seja, tira-se a média dos vetores com os rankeamentos finais dos itens para se obter a ordenação final dos mesmos.

Entrada de dados:

No software MINITAB os dados devem estar nas colunas da planilha de forma que as matrizes de comparação fiquem lado a lado, como na figura abaixo:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
							a			b		
1	1,000	0,333	8,0	3,000	3,000	7	1	0,333	0,5	1	1	1
2	3,000	1,000	9,0	3,000	3,000	9	2	1,000	3,0	1	1	1
3	0,125	0,111	1,0	0,167	0,200	2	3	0,333	1,0	1	1	1
4	0,333	0,333	6,0	1,000	0,333	6						
5	0,333	0,333	5,0	3,000	1,000	6						
6	0,143	0,111	0,5	0,167	0,167	1						
7												

Figura 4 – Exemplo de entrada de tabela de dados no MINITAB

Na Figura 4 a matriz destacada em rosa é a matriz de comparação de atributos/ características e as matrizes marcadas de vermelho são as de comparação dos itens.

No software S-Plus os dados devem estar da mesma forma como No MINITAB (dentro de uma planilha de dados do próprio software). Caso o estudo tenha sido feito com mais de um julgador as matrizes devem estar como na Figura 5.

		1	2	3	4	5	6	7	8	9	10	11	12
		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12
Julgador 1	1	1.00	3.00	7.00	1.00	1.00	1.00	1.00	0.20	1.00	1.00	1.00	0.14
	2	0.33	1.00	3.00	1.00	1.00	1.00	5.00	1.00	5.00	1.00	1.00	5.00
	3	0.14	0.33	1.00	1.00	1.00	1.00	1.00	0.20	1.00	7.00	0.20	1.00
Julgador 2	4	1.00	3.00	5.00	1.00	1.00	9.00	1.00	0.20	0.33	1.00	7.00	5.00
	5	0.33	1.00	3.00	1.00	1.00	7.00	5.00	1.00	0.20	0.14	1.00	3.00
	6	0.20	0.33	1.00	0.11	0.14	1.00	3.00	5.00	1.00	0.20	0.33	1.00
Julgador 3	7	1.00	5.00	9.00	1.00	3.00	3.00	1.00	1.00	5.00	1.00	3.00	7.00
	8	0.20	1.00	1.00	0.33	1.00	0.33	1.00	1.00	5.00	0.33	1.00	5.00
	9	0.11	1.00	1.00	0.33	3.00	1.00	0.20	0.20	1.00	0.14	0.20	1.00
Julgador 4	10	1.00	3.00	7.00	1.00	3.00	5.00	1.00	3.00	5.00	1.00	3.00	5.00
	11	0.33	1.00	5.00	0.33	1.00	5.00	0.33	1.00	5.00	0.33	1.00	5.00
	12	0.14	0.20	1.00	0.20	0.20	1.00	0.20	0.20	1.00	0.20	0.20	1.00
Julgador 5	13	1.00	3.00	5.00	1.00	0.20	3.00	1.00	0.14	9.00	1.00	0.20	0.20
	14	0.33	1.00	5.00	5.00	1.00	5.00	7.00	1.00	9.00	5.00	1.00	1.00
	15	0.20	0.20	1.00	0.33	0.20	1.00	0.11	0.11	1.00	5.00	1.00	1.00
16													

Figura 5 – Exemplo de entrada de tabela de dados no S-Plus

Note que assim como no MINITAB a matriz de comparação dos atributos (retângulo rosa) é colocada primeiro na sequência as matrizes de comparação dos itens (retângulos vermelhos), sendo que as matrizes correspondentes a cada julgador é colocada uma abaixo da outra.

Execução dos programas e saídas

A seguir serão apresentados os comandos para a execução dos programas e os resultados gerados a partir dos mesmos. Os programas encontram-se no anexo.

Macro desenvolvida no software MINITAB:

Comando :

```
%arquivo.txt c1-cX;  
sub cY - cZ.
```

Onde *arquivo* é o nome que foi salvo o arquivo com a macro, *c1-cX* são as primeira e última colunas onde está a matriz de comparação dos atributos e *cY-cZ* são as primeira e última colunas onde estão as matrizes de comparação dos itens.

Dado estes comandos aparecerá a seguinte saída:

```
!!! A Matriz é consistente!!!!  
  
!!!!!!Comparações dos atributos (dois-a-dois)!!!!!!  
  
AUTOVALOR (lambda máx):  
Data Display  
  
lambda      6,48827  
  
AUTOVETOR NORMALIZADO (T):  
Data Display  
  
T  
0,260220    0,399180    0,034596    0,116181    0,162573    0,027250  
  
ÍNDICE DE CONSISTÊNCIA E RAZÃO DE CONSISTÊNCIA:  
Data Display  
  
IC      0,0976549  
RC      0,0787539  
  
!!!!!!Comparações dos itens, com relação à cada atributo!!!!!!  
  
AUTOVETORES NORMALIZADOS:  
Data Display  
  
Row      X.1      X.2      X.3      X.4      X.5      X.6  
1      0,163395    0,333333    0,454545    0,772005    0,25    0,690957  
2      0,539697    0,333333    0,090909    0,054545    0,50    0,091405  
3      0,296908    0,333333    0,454545    0,173450    0,25    0,217638  
  
ORDENAÇÃO FINAL DOS ITENS:  
Data Display  
  
ord  
0,340468    0,366759    0,292772
```

Figura 4: Exemplo de saída da macro do MINITAB.

☞ Funções desenvolvidas no software S-Plus for Windows:

Programa 1) Análise para apenas um julgador.

Comando:

```
AnaHier(n, m, dados)
```

Onde n é o número de alternativas, m é o número de atributos e $dados$ é o nome da planilha do S-Plus onde estão os dados.

Através do comando acima a seguinte saída pode ser obtida:

```
*****
A MATRIZ DE COMPARACAO DOS ATRIBUTOS E CONSISTENTE!!!

COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):

Autovetor (normalizado):
[1] 0.26021953 0.39918011 0.03459625 0.11618051 0.16257331 0.02725029

Autovalor Lambda: 6.48827435009174

Indice de consistencia: 0.0976548700183491

Razao de consistencia: 0.0787539274341525

COMPARACOES DOS ITENS, COM RELACAO A CADA ATRIBUTO:

Autovetores (normalizados):
      [,1]      [,2]      [,3]      [,4] [,5]      [,6]
[1,] 0.1633947 0.3333333 0.45454545 0.77200512 0.25 0.6909570
[2,] 0.5396974 0.3333333 0.09090909 0.05454497 0.50 0.0914052
[3,] 0.2969079 0.3333333 0.45454545 0.17344991 0.25 0.2176378

Vetor da ordenacao final dos itens: 0.340468154276007 0.366759490094986
0.292772355629008

*****
```

Figura 5: Exemplo de saída da função 1 do S-plus para uma matriz de comparação de atributos consistente.

No caso em que a matriz de comparação dos atributos não é consistente a saída do software é semelhante a da Figura 5, entretanto sem os resultados das comparações dos itens com relação a cada atributo.

Programa 2) Análise para um ou mais decisores e resultados para cada um deles.

Comando:

AHP(r, n, m, dados)

Onde *r* é o número de julgadores, *n* é o número de alternativas, *m* é o número de atributos e *dados* é o nome da planilha do S-Plus onde estão os dados.

A saída desta função é semelhante à mostrada na Figura 5, porém no caso de mais de um julgador para cada um deles haverá uma saída como a do programa 1. Quando a matriz de comparação dos atributos não é consistente a saída do software é semelhante a da Figura 5, entretanto sem os resultados das comparações dos itens com relação a cada atributo.

Programa 3) Análise para um ou mais decisores através da matriz da média das matrizes de comparações.

Comando:

AHPmediaA(r, n, m, dados)

Onde *r* é o número de julgadores, *n* é o número de alternativas, *m* é o número de atributos e *dados* é o nome da planilha do S-Plus onde estão os dados.

```
*****
A MATRIZ DE COMPARACAO DOS ATRIBUTOS E CONSISTENTE!!!

COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):

Autovetor (normalizado):
[1] 0.67557070 0.23951342 0.08491588

Autovalor Lambda: 3.03500599897076

Indice de consistencia: 0.0175029994853817

Razao de consistencia: 0.0301775853196236

COMPARACOES DOS ITENS, COM RELACAO A CADA ATRIBUTO:

Autovetores (normalizados):
      [,1]      [,2]      [,3]
[1,] 0.5267333 0.4220716 0.5842741
[2,] 0.3619945 0.4768113 0.3003413
[3,] 0.1112721 0.1011171 0.1153847

Vetor da ordenacao final dos itens: 0.506551559324792 0.384259343474904
0.109189097200304

*****
```

Figura 6: Exemplo de saída da função 3 do S-plus.

Programa 4) Análise para um ou mais decisores e resultado final através da média dos vetores com as ordenações finais das alternativas.

Comando:

AHPmediaD(r, n, m, dados)

Onde *r* é o número de julgadores, *n* é o número de alternativas, *m* é o número de atributos e *dados* é o nome da planilha do S-Plus onde estão os dados.

Através dos comandos citados poderá se obter saídas como as das figuras 7 e 8.

```
A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR: 5
NAO E CONSISTENTE, POIS A RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!
Razao de consistencia: 0.116905900567279

*****
Vetor da media das ordenacoes finais dos itens: 0.470022681757993 0.325053982249725
0.204923335992282
*****
```

Figura 7: Exemplo de saída da função 4 do S-plus no caso em que uma matriz de comparação dos atributos não é consistente.

```
A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR: 1
NAO E CONSISTENTE, POIS A RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!
Razao de consistencia: 0.833627854806995

A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR: 2
NAO E CONSISTENTE, POIS A RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!
Razao de consistencia: 0.607597770942171

A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR: 3
NAO E CONSISTENTE, POIS A RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!
Razao de consistencia: 0.988360080750919

A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR: 4
NAO E CONSISTENTE, POIS A RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!
Razao de consistencia: 0.231268344736933

*****
Vetor da media das ordenacoes finais dos itens: NaN NaN NaN
Nenhuma das matrizes de comparacao dos atributos e consistente!
*****
```

Figura 8: Exemplo de saída da função 4 do S-plus no caso em que nenhuma matriz de comparação dos atributos é consistente.

Para as funções 2, 3 e 4 do S-Plus pode-se ter ainda saídas para os casos em que o número de julgadores é inválido (igual a 0, ou -1, por exemplo), ou o número de atributos é maior que 9.

REFERÊNCIAS BIBLIOGRÁFICAS

CARVALHO, Giselle S.; MINGOTI, Sueli A. (Orientadora); *Implementação Computacional da Técnica de Estatística Multivariada Análise Hierárquica*. UFMG, Instituto de Ciências Exatas, Departamento de Estatística, Março, 2005.

MONTEVEQUI, José Arnaldo B., PAMPLONA, Edson de Oliveira; *Análise hierárquica em análise de investimentos*. 16o ENEGEP. Piracicaba, SP; outubro de 1996.

MURAKAMI, Milton; ALMEIDA, Martinho Isnard Ribeiro de (Orientador); *Decisão estratégica em TI: estudo de caso*. Faculdade de Economia, Administração e Contabilidade; 2003.

COSTA, Helder Gomes, *Introdução ao método de análise hierárquica: análise multicritério no auxílio à decisão*. UFF, Niterói, RJ; 2002.

LISBOA, Marcus Vinícius; WAISMAN, Jaime (Orientador); *Contribuição para a tomada de decisão para a classificação e seleção e alternativas de traçado para rodovias em trechos urbanizados*. USP, Escola Politécnica; São Paulo; 2002.

Reis, Edna Afonso; *Noções Básicas de S-Plus for Windows*. UFMG, Instituto de Ciências Exatas, Departamento de Estatística, Setembro - 1997.

VENABLES, W. N; RIPLEY, Brian D.. *Modern applied statistics with S-PLUS*. New York: Springer, c1994.

ANEXO

I) Macro no MINITAB

```
MACRO
#template
AnalHier A.1-A.N;
  sub B.1-B.L.
##A.1-A.N é a matriz de comparação dos atributos dois-a-dois.
##B.1-B.M é a matriz de comparação dos itens, com relação à cada atributo.
#variáveis:
MCOLUMN A.1-A.N W1 Z.1-Z.N v T B.1-B.L X.1-X.N ord
MCONSTANT i j k m n c p a1 a2 b1 somaW1 somaZk somaZ1 lambda IC CR RC
#
let m=count(A.1)
let n=count(B.1)
let a2=0
let lambda=0
let RC=0
#
##Análise dos atributos:
#
##Calculando o autovetor T (normalizado):
do i=1:m
  let a1=1
  do j=1:m
    let a1=a1*A.j(i)
  enddo
  let W1(i)=(a1**(1/m))
enddo
let somaW1=sum(W1)
let T=W1*(1/somaW1)
#
##Calculando o autovalor Lambda:
do i=1:m
  let a2=sum(A.i)
  let v(i)=a2
enddo
do i=1:m
  let lambda=lambda+(T(i)*v(i))
enddo
#
##Calculando o Índice de consistência
let IC=(lambda-m)/(m-1)
#
##calculando a razão de consistência
if m=3
  let CR=0.58
  let RC=IC/CR
elseif m=4
  let CR=0.9
  let RC=IC/CR
elseif m=5
  let CR=1.12
  let RC=IC/CR
elseif m=6
  let CR=1.24
  let RC=IC/CR
elseif m=7
  let CR=1.32
  let RC=IC/CR
elseif m=8
```



```

let CR=1.41
let RC=IC/CR
elseif m=9
let CR=1.45
let RC=IC/CR
else
NOTE !!!!NÃO É POSSIVEL CALCULAR A RAZAO DE CONSISTÊNCIA!!!!
endif
#
##Testando a consistência da matriz:
if RC=0
if IC>0.10
NOTE !!! A matriz analisada não é consistente!!!!
#resultados:
NOTE #
NOTE !!!!!Comparações dos atributos (dois-a-dois)!!!!!!
Note #
NOTE AUTOVALOR (lambda máx):
print lambda
Note #
NOTE AUTOVETOR NORMALIZADO (T):
print T
Note #
NOTE ÍNDICE DE CONSISTÊNCIA:
print IC
EXIT
else
NOTE !!! A Matriz é consistente!!!!
#
##Análise dos itens em relação aos atributos:
#
##Calculando os autovetores (normalizados) V's:
#
do i=1:n
let b1=1
do j=1:n
let b1=b1*B.j(i)
enddo
let Z.1(i)=(b1**(1/n))
enddo
let somaZ1=sum(Z.1)
let X.1=Z.1*(1/somaZ1)
#
let c=n+1
let p=c+(n-1)
do k =2:m
do i=1:n
let b1=1
do j=c:p
let b1=b1*B.j(i)
enddo
let Z.k(i)=(b1**(1/n))
enddo
let c=c+n
let p=c+(n-1)
let somaZk=sum(Z.k)
let X.k=Z.k*(1/somaZk)
enddo
#
#ordenação final dos itens em estudo:
do i=1:n
let ord(i)=0
do j=1:m
let ord(i)=ord(i)+T(j)*X.j(i)

```

```

        enddo
    enddo
#
#resultados
NOTE #
NOTE !!!!!Comparações dos atributos (dois-a-dois)!!!!!!
Note #
NOTE AUTOVALOR (lambda máx):
print lambda
Note #
NOTE AUTOVETOR NORMALIZADO (T):
print T
Note #
NOTE ÍNDICE DE CONSISTÊNCIA E RAZÃO DE CONSISTÊNCIA::
print IC RC
NOTE #
NOTE !!!!!Comparações dos itens, com relação à cada atributo!!!!!!
NOTE AUTOVETORES NORMALIZADOS:
print X.1-X.n
NOTE #
NOTE ORDENAÇÃO FINAL DOS ITENS:
print ord
endif
elseif RC>0.10
    NOTE !!! A matriz analisada NÃO é consistente, pois a razão de consistencia é maior que 0,10 !!!
#resultados
    NOTE #
    NOTE !!!!!Comparações dos atributos (dois-a-dois)!!!!!!
    Note #
    NOTE AUTOVALOR (lambda máx):
    print lambda
    Note #
    NOTE AUTOVETOR NORMALIZADO (T):
    print T
    Note #
    NOTE ÍNDICE DE CONSISTÊNCIA E RAZÃO DE CONSISTÊNCIA:
    print IC RC
    EXIT
else
NOTE !!! A Matriz é consistente!!!!
#
##Análise dos itens em relação aos atributos:
#
##Calculando os autovetores (normalizados) V's:
#
do i=1:n
    let b1=1
    do j=1:n
        let b1=b1*B.j(i)
    enddo
    let Z.1(i)=(b1**(1/n))
enddo
let somaZ1=sum(Z.1)
let X.1=Z.1*(1/somaZ1)
#
let c=n+1
let p=c+(n-1)
#
do k =2:m
    do i=1:n
        let b1=1
        do j=c:p
            let b1=b1*B.j(i)

```

```

        enddo
        let Z.k(i)=(b1**(1/n))
    enddo
let c=c+n
let p=c+(n-1)
let somaZk=sum(Z.k)
let X.k=Z.k*(1/somaZk)
enddo
#
#ordenação final dos itens em estudo:
do i=1:n
    let ord(i)=0
    do j=1:m
        let ord(i)=ord(i)+T(j)*X.j(i)
    enddo
enddo
#
#resultados
NOTE #
NOTE !!!!!Comparações dos atributos (dois-a-dois)!!!!!!
Note #
NOTE AUTOVALOR (lambda máx):
print lambda
Note #
NOTE AUTOVETOR NORMALIZADO (T):
print T
Note #
NOTE ÍNDICE DE CONSISTÊNCIA E RAZÃO DE CONSISTÊNCIA:
print IC RC
NOTE #
NOTE !!!!!Comparações dos itens, com relação à cada atributo!!!!!!
Note #
NOTE AUTOVETORES NORMALIZADOS:
print X.1-X.M
NOTE #
NOTE ORDENAÇÃO FINAL DOS ITENS:
print ord
#
endif
ENDMACRO

```

II) Programa 1 do S-Plus

```

AnaHier_function(n, m, dados){
    #n é o número de itens e m é o números de atributos.
    cat( , fill=T)
    cat("*****",
*****",
fill=T)
    W_c()
    TN_c()
    v_c()
    X_c()
    XN_array(dim=c(n, m))
    RC_0
    R_c()
    S_0
    nm_m+(n*m)
    dados1_dados[1:m,1:m]
    dados2_dados[1:n,(m+1):nm]
    #índice de consistencia randomico:
    CR_c(0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45)

```

```

#Cálculo do autovetor TN (normalizado):
for (i in 1:m){
    W[i]_(prod(dados1[i,]))^(1/m)
}
TN_W/(sum(W))
#cálculo do autovalor Lâmbda:
for (i in 1:m){
    v[i]_sum(dados1[,i])
}
lambda_TN%%v
#Cálculo do Índice de consistência:
IC_(lambda-m)/(m-1)
#cáculo da razão de consistência:
if (m==1||m==2) { RC_IC/CR[1] }
    else if (m==3) { RC_IC/CR[3] }
        else if (m==4) { RC_IC/CR[4] }
            else if (m==5) { RC_IC/CR[5] }
                else if (m==6) { RC_IC/CR[6] }
                    else if (m==7) { RC_IC/CR[7] }
                        else if (m==8) { RC_IC/CR[8] }
                            else if (m==9) { RC_IC/CR[9] }
                                else { cat("NÃO É POSSIVEL
CALCULAR A RAZAO DE CONSISTENCIA!!!", fill=T)}
#testando a consistencia da matriz:
if (RC==0) {
    if (IC>=0.10){ cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS NAO E
CONSISTENTE, POIS O INDICE DE CONSISTENCIA É MAIOR QUE 0.10!!!", fill=T)
cat( , fill=T)
cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
cat( , fill=T)
cat("Autovetor (normalizado):", fill=T)
print(TN)
cat( , fill=T)
cat("Autovalor Lambda:", lambda, fill=T)
cat( , fill=T)
cat("Indice de consistencia:", IC, fill=T)
cat( , fill=T) }
    else {
cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS E
CONSISTENTE!!!", fill=T)
#Análise dos itens em relação a cada atributo:
#calculo dos autovetores (normalizados):
for (j in 1:m){
    for (i in seq(1, (n*m), by=n)){
        for (k in i:(i+(n-1))) {
            X[k]_(prod(dados2[k,]))^(1/n)
        }
        S_S+1
        XN[,S]_X[i:(i+(n-1))]/(sum(X[i:(i+(n-1))]))
    }
}
R_TN%%XN
cat( , fill=T)
cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
cat( , fill=T)
cat("Autovetor (normalizado):", fill=T)
print(TN)
cat( , fill=T)
cat("Autovalor Lambda:", lambda, fill=T)
cat( , fill=T)
cat("Indice de consistencia:", IC, fill=T)
cat( , fill=T)
cat("Razao de consistencia:", RC, fill=T)
cat( , fill=T)

```

```

        cat("COMPARACOES DOS ITENS, COM RELACAO A CADA
ATRIBUTO:", fill=T)
        cat( , fill=T)
        cat("Autovetores (normalizados):", fill=T)
        print(XN)
        cat( , fill=T)
        cat("Vetor da ordenacao final dos itens:", R, fill=T)
        cat( , fill=T)
    }
}
else if (RC>0.10) {
    cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS NAO E CONSISTENTE, POIS A
RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!", fill=T)
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
    cat( , fill=T)
    cat("Autovalor Lambda:", lambda, fill=T)
    cat( , fill=T)
    cat("Indice de consistencia:", IC, fill=T)
    cat( , fill=T)
    cat("Razao de consistencia:", RC, fill=T)
    cat( , fill=T)}
else {
    cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS E CONSISTENTE!!!", fill=T)
    #Análise dos itens em relação aos atributos:
    #calculo dos autovetores (normalizados):
    for (j in 1:m){
        S_0
        for (i in seq(1, (n*m), by=n)){
            c_i
            p_i+(n-1)
            for (k in 1:3) {
                X[k]_(prod(dados2[k,c:p]))^(1/n)
            }
            S_S+1
            XN[,S]_X/(sum(X))
        }
    }
    R_XN%*%TN
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
    cat( , fill=T)
    cat("Autovalor Lambda:", lambda, fill=T)
    cat( , fill=T)
    cat("Indice de consistencia:", IC, fill=T)
    cat( , fill=T)
    cat("Razao de consistencia:", RC, fill=T)
    cat( , fill=T)
    cat("COMPARACOES DOS ITENS, COM RELACAO A CADA ATRIBUTO:", fill=T)
    cat( , fill=T)
    cat("Autovetores (normalizados):", fill=T)
    print(XN)
    cat( , fill=T)
    cat("Vetor da ordenacao final dos itens:", R, fill=T)
    cat( , fill=T)
}

```

```

cat("*****
*****",
fill=T)
}

```

III) Programa 2 do S-Plus

```

AHP_function(r, n, m, dados){
  #r e o numero de repeticoes, ou seja, o tamanho da amostra de julgadores.
  cat( , fill=T)
  if (r==1){

    cat("*****
*****",
fill=T)
      AnaHier(n, m, dados)

    cat("*****
*****",
fill=T)
  }
  else if (r<1){
    cat("!!!!!! NUMERO DE JULGADORES INVALIDO!!!!!!", fill=T) }
  else {
    for (k in seq(1, r*m, by=m)){
      dadosN_dados[k:(k+(m-1)), 1:(m+(n*m))]
      AnaHier(n, m, dadosN)

    cat("*****
*****",
fill=T)
      }
  }
}

```

```

AnaHier_function(n, m, dados){
  #n é o número de itens e m é o números de atributos.
  W_c()
  TN_c()
  v_c()
  X_c()
  XN_array(dim=c(n, m))
  RC_0
  R_c()
  S_0
  nm_m+(n*m)
  dados1_dados[1:m,1:m]
  dados2_dados[1:n,(m+1):nm]
  #índice de consistencia randomico:
  CR_c(0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45)
  #Cálculo do autovetor TN (normalizado):
  for (i in 1:m){
    W[i]_(prod(dados1[i,]))^(1/m)
  }
  TN_W/(sum(W))
  #cálculo do autovalor Lâmbda:
  for (i in 1:m){
    v[i]_sum(dados1[,i])
  }
  lambda_TN%%v
}

```

```

#Cálculo do Índice de consistência:
IC_(lambda-m)/(m-1)
#cáculo da razão de consistência:
if (m==1||m==2) { RC_IC/CR[1] }
  else if (m==3) { RC_IC/CR[3] }
    else if (m==4) { RC_IC/CR[4] }
      else if (m==5) { RC_IC/CR[5] }
        else if (m==6) { RC_IC/CR[6] }
          else if (m==7) { RC_IC/CR[7] }
            else if (m==8) { RC_IC/CR[8] }
              else if (m==9) { RC_IC/CR[9] }
                else { cat("NAO E POSSIVEL
CALCULAR A RAZAO DE CONSISTENCIA!!!", fill=T)}
  #testando a consistencia da matriz:
  if (RC==0) {
    if (IC>=0.10){ cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS NAO E
CONSISTENTE, POIS O INDICE DE CONSISTENCIA É MAIOR QUE 0.10!!!", fill=T)
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
    cat( , fill=T)
    cat("Autovalor Lambda:", lambda, fill=T)
    cat( , fill=T)
    cat("Indice de consistencia:", IC, fill=T)
    cat( , fill=T) }
    else {
CONSISTENTE!!!", fill=T)
    #Análise dos itens em relação a cada atributo:
    #calculo dos autovetores (normalizados):
    for (j in 1:m){
      for (i in seq(1, (n*m), by=n)){
        for (k in i:(i+(n-1))) {
          X[k]_(prod(dados2[k,]))^(1/n)
        }
        S_S+1
        XN[,S]_X[i:(i+(n-1))]/(sum(X[i:(i+(n-1))]))
      }
    }
    R_TN%*%XN
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
    cat( , fill=T)
    cat("Autovalor Lambda:", lambda, fill=T)
    cat( , fill=T)
    cat("Indice de consistencia:", IC, fill=T)
    cat( , fill=T)
    cat("Razao de consistencia:", RC, fill=T)
    cat( , fill=T)
    cat("COMPARACOES DOS ITENS, COM RELACAO A CADA
ATRIBUTO:", fill=T)
    cat( , fill=T)
    cat("Autovetores (normalizados):", fill=T)
    print(XN)
    cat( , fill=T)
    cat("Vetor da ordenacao final dos itens:", R, fill=T)
    cat( , fill=T)
  }
}

```

```

else if (RC>0.10) {
  cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS NAO E CONSISTENTE, POIS A
RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!", fill=T)
  cat( , fill=T)
  cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
  cat( , fill=T)
  cat("Autovetor (normalizado):", fill=T)
  print(TN)
  cat( , fill=T)
  cat("Autovalor Lambda:", lambda, fill=T)
  cat( , fill=T)
  cat("Indice de consistencia:", IC, fill=T)
  cat( , fill=T)
  cat("Razao de consistencia:", RC, fill=T)
  cat( , fill=T) }
else {
  cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS E CONSISTENTE!!!", fill=T)
  #Análise dos itens em relação aos atributos:
  #calculo dos autovetores (normalizados):
  for (j in 1:m){
    S_0
    for (i in seq(1, (n*m), by=n)){
      c_i
      p_i+(n-1)
      for (k in 1:3) {
        X[k]_(prod(dados2[k,c:p]))^(1/n)
      }
      S_S+1
      XN[,S]_X/(sum(X))
    }
  }
  R_XN%*%TN
  cat( , fill=T)
  cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
  cat( , fill=T)
  cat("Autovetor (normalizado):", fill=T)
  print(TN)
  cat( , fill=T)
  cat("Autovalor Lambda:", lambda, fill=T)
  cat( , fill=T)
  cat("Indice de consistencia:", IC, fill=T)
  cat( , fill=T)
  cat("Razao de consistencia:", RC, fill=T)
  cat( , fill=T)
  cat("COMPARACOES DOS ITENS, COM RELACAO A CADA ATRIBUTO:", fill=T)
  cat( , fill=T)
  cat("Autovetores (normalizados):", fill=T)
  print(XN)
  cat( , fill=T)
  cat("Vetor da ordenacao final dos itens:", R, fill=T)
  cat( , fill=T)
}
}

```


IV) Programa 3 do S-Plus

```
AHPmediaA_function(r, n, m, dados){
  #r é o numero de repeticoes, ou seja, o tamanho da amostra de julgadores.
  #
  #Esta funcao trabalha com as medias das matrizes de comparacao!!
  #
  cat( , fill=T)
  dadosSoma_array(data=0, dim=c(m, (n*m+m)))
  #caso se tenha apenas um julgador:
  if (r==1){

    cat("*****
*****",
fill=T)
    AnaHier(n, m, dados)

    cat("*****
*****",
fill=T)
  }
  #caso o numero de julgadores fornecido pelo usuario nao seja válido:
  else if (r<1){
    cat("!!!!!! NUMERO DE JULGADORES INVALIDO!!!!!!", fill=T)
  }
  #
  #calculando as matrizes de médias dos julgamentos:
  #matriz das medias das comparações dos atributos:
  else {
    for (k in seq(1, r*m, by=m)){
      dadosN_dados[k:(k+(m-1)), 1:(m+(n*m))]
      dadosSoma_dadosSoma+dadosN
    }
    media_dadosSoma*(1/r)
    a_0
    for (i in 1:m){
      a_a+1
      for (j in a:m){
        media[j, i]_1/(media[i, j])
      }
    }
    #matriz reciproca das medias das comparações dos itens:
    for (l in 1:m){
      for (i in seq((m+1), ((n*m)+m), by=n)){
        a_0
        for (k in i:(i+(n-1))) {
          a_a+1
          for (j in a:n){
            media[j, k]_1/(media[k-(i-1), j+(i-1)])
          }
        }
      }
    }

    cat("*****
*****",
fill=T)
    AnaHier(n, m, media)

    cat("*****
*****",
fill=T)
  }
}
```

```

}

AnaHier_function(n, m, dados){
  #n é o número de itens e m é o números de atributos.
  W_c()
  TN_c()
  v_c()
  X_c()
  XN_array(dim=c(n, m))
  RC_0
  R_c()
  S_0
  nm_m+(n*m)
  dados1_dados[1:m,1:m]
  dados2_dados[1:n,(m+1):nm]
  #índice de consistencia randomico:
  CR_c(0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45)
  #Cálculo do autovetor TN (normalizado):
  for (i in 1:m){
    W[i]_(prod(dados1[i,]))^(1/m)
  }
  TN_W/(sum(W))
  #cálculo do autovalor Lâmbda:
  for (i in 1:m){
    v[i]_sum(dados1[,i])
  }
  lambda_TN%*%v
  #Cálculo do Índice de consistência:
  IC_(lambda-m)/(m-1)
  #cáculo da razão de consistência:
  if (m==1||m==2) { RC_IC/CR[1] }
    else if (m==3) { RC_IC/CR[3] }
      else if (m==4) { RC_IC/CR[4] }
        else if (m==5) { RC_IC/CR[5] }
          else if (m==6) { RC_IC/CR[6] }
            else if (m==7) { RC_IC/CR[7] }
              else if (m==8) { RC_IC/CR[8] }
                else if (m==9) { RC_IC/CR[9] }
                  else { cat("NÃO E POSSIVEL
CALCULAR A RAZAO DE CONSISTENCIA!!!", fill=T)}
  #testando a consistencia da matriz:
  if (RC==0) {
    if (IC>=0.10){ cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS NAO E
CONSISTENTE, POIS O INDICE DE CONSISTENCIA É MAIOR QUE 0.10!!!", fill=T)
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
    cat( , fill=T)
    cat("Autovalor Lambda:", lambda, fill=T)
    cat( , fill=T)
    cat("Índice de consistencia:", IC, fill=T)
    cat( , fill=T) }
    else {
      cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS E
CONSISTENTE!!!", fill=T)
    #Análise dos itens em relação a cada atributo:
    #calculo dos autovetores (normalizados):
    for (j in 1:m){
      for (i in seq(1, (n*m), by=n)){
        for (k in i:(i+(n-1))) {
          X[k]_(prod(dados2[k,]))^(1/n)
        }
      }
    }
  }
}

```

```

        S_S+1
        XN[,S]_X[i:(i+(n-1))]/(sum(X[i:(i+(n-1))]))
    }
}
R_TN%%XN
cat( , fill=T)
cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
cat( , fill=T)
cat("Autovetor (normalizado):", fill=T)
print(TN)
cat( , fill=T)
cat("Autovalor Lambda:", lambda, fill=T)
cat( , fill=T)
cat("Indice de consistencia:", IC, fill=T)
cat( , fill=T)
cat("Razao de consistencia:", RC, fill=T)
cat( , fill=T)
cat("COMPARACOES DOS ITENS, COM RELACAO A CADA
ATRIBUTO:", fill=T)
cat( , fill=T)
cat("Autovetores (normalizados):", fill=T)
print(XN)
cat( , fill=T)
cat("Vetor da ordenacao final dos itens:", R, fill=T)
cat( , fill=T)
}
else if (RC>0.10) {
    cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS NAO E CONSISTENTE, POIS A
RAZAO DE CONSISTENCIA E MAIOR QUE 0.10!", fill=T)
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
    cat( , fill=T)
    cat("Autovalor Lambda:", lambda, fill=T)
    cat( , fill=T)
    cat("Indice de consistencia:", IC, fill=T)
    cat( , fill=T)
    cat("Razao de consistencia:", RC, fill=T)
    cat( , fill=T) }
else {
    cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS E CONSISTENTE!!!", fill=T)
    #Análise dos itens em relação aos atributos:
    #calculo dos autovetores (normalizados):
    for (j in 1:m){
        S_0
        for (i in seq(1, (n*m), by=n)){
            c_i
            p_i+(n-1)
            for (k in 1:3) {
                X[k]_(prod(dados2[k,c:p]))^(1/n)
            }
            S_S+1
            XN[,S]_X/(sum(X))
        }
    }
    R_XN%%TN
    cat( , fill=T)
    cat("COMPARACOES DOS ATRIBUTOS (DOIS-A-DOIS):", fill=T)
    cat( , fill=T)
    cat("Autovetor (normalizado):", fill=T)
    print(TN)
}

```

```

cat( , fill=T)
cat("Autovalor Lambda:", lambda, fill=T)
cat( , fill=T)
cat("Indice de consistencia:", IC, fill=T)
cat( , fill=T)
cat("Razao de consistencia:", RC, fill=T)
cat( , fill=T)
cat("COMPARACOES DOS ITENS, COM RELACAO A CADA ATRIBUTO:", fill=T)
cat( , fill=T)
cat("Autovetores (normalizados):", fill=T)
print(XN)
cat( , fill=T)
cat("Vetor da ordenacao final dos itens:", R, fill=T)
cat( , fill=T)
}
}

```

V) Programa 4 do S-Plus

```

AHPmediaD_function(r, n, m, dados){
  #r é o numero de repeticoes, ou seja, o tamanho da amostra de julgadores.
  #
  #Esta funcao devolve a media da ordenacao final dos itens que estao sendo
comparados!!
  #
  cat( , fill=T)
  cont_0
  soma_array(data=0, dim=c(n, 1))
  if (r==1){

    cat("*****
*****",
fill=T)
    AnaHier(n, m, dados)

    cat("*****
*****",
fill=T)
  }
  else if (r<1){
    cat("!!!!!! NUMERO DE JULGADORES INVALIDO!!!!!!", fill=T) }
  else {
    numMatriz_0
    for (k in seq(1, r*m, by=m)){
      dadosN_dados[k:(k+(m-1)), 1:(m+(n*m))]
      numMatriz_numMatriz+1
      #
      #A Analise Hierarquica:
      #
      #n é o número de itens e m é o números de atributos.
      W_c()
      TN_c()
      v_c()
      X_c()
      XN_array(dim=c(n, m))
      RC_0
      R_c()
      S_0
      nm_m+(n*m)
      dados1_dadosN[1:m,1:m]
      dados2_dadosN[1:n,(m+1):nm]
      #indice de consistencia randomico:
      CR_c(0, 0, 0.58, 0.9, 1.12, 1.24, 1.32, 1.41, 1.45)

```

```

#Cálculo do autovetor TN (normalizado):
for (i in 1:m){
  W[i]_(prod(dados1[i,]))^(1/m)
}
TN_W/(sum(W))
#cálculo do autovalor Lâmbda:
for (i in 1:m){
  v[i]_sum(dados1[,i])
}
lambda_TN%%v
#Cálculo do Índice de consistência:
IC_(lambda-m)/(m-1)
#cáculo da razão de consistência:
if (m==1||m==2) { RC_IC/CR[1] }
  else if (m==3) { RC_IC/CR[3] }
    else if (m==4) { RC_IC/CR[4] }
      else if (m==5) { RC_IC/CR[5] }
        else if (m==6) { RC_IC/CR[6] }
          else if (m==7) { RC_IC/CR[7] }
            else if (m==8) { RC_IC/CR[8] }
              else if (m==9) { RC_IC/CR[9] }
                else {
JULGADOR: ", numMatriz, fill=T)
                cat("NAO E POSSIVEL
CALCULAR A RAZAO DE CONSISTENCIA, POIS m > 9!!!", fill=T)
                cat( , fill=T)
                R_NA
                }

#testando a consistencia da matriz:
if (RC==0) {
  if (IC>=0.10){
    cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR:
", numMatriz, fill=T)
    cat("NAO E CONSISTENTE, POIS O INDICE DE CONSISTENCIA É
MAIOR QUE 0.10!!!", fill=T)
    cat( , fill=T)
    R_NA
  }
  else {
    #A matriz e consistente!
    #Análise dos itens em relação a cada atributo:
    #calculo dos autovetores (normalizados):
    for (j in 1:m){
      for (i in seq(1, (n*m), by=n)){
        for (k in i:(i+(n-1))) {
          X[k]_(prod(dados2[k,]))^(1/n)
        }
        S_S+1
        XN[,S]_X[i:(i+(n-1))]/(sum(X[i:(i+(n-1))]))
      }
    }
    R_TN%%XN
  }
}
else if (RC>0.10) {
  cat("A MATRIZ DE COMPARACAO DOS ATRIBUTOS DO JULGADOR: ", numMatriz,
fill=T)
  cat("NAO E CONSISTENTE, POIS A RAZAO DE CONSISTENCIA E MAIOR QUE
0.10!", fill=T)
  cat( , fill=T)
  R_NA
}
else {

```

```

#A matriz e consistente!
#Análise dos itens em relação aos atributos:
#calculo dos autovetores (normalizados):
for (j in 1:m){
  S_0
  for (i in seq(1, (n*m), by=n)){
    c_i
    p_i+(n-1)
    for (k in 1:3) {
      X[k]_(prod(dados2[k,c:p]))^(1/n)
    }
    S_S+1
    XN[,S]_X/(sum(X))
  }
}
R_XN%%TN
}
testeR_is.na(R)
  #Calculo da media apenas para as matrizes consistentes!!
  if (testeR == F){
    #Calculo da media das ordenacoes dos itens:
    #
    soma_soma+R
    cont_cont+1
  }
}
mediafinal_soma/cont

cat("*****
*****",
fill=T)
  cat("Vetor da media das ordenacoes finais dos itens: ", mediafinal,
fill=T)
  if (cont == 0){
    cat("Nenhuma das matrizes de comparacao dos atributos e
consistente!", fill=T)
  }

  cat("*****
*****",
fill=T)
}
}

```