

Métodos Multivariados de Análise de Dados*

3ª Atividade

Alberson da Silva Miranda

30 de setembro de 2024

*Código disponível em https://github.com/albersonmiranda/analise_multivariada.

Índice

1	INTRODUÇÃO	3
2	ANÁLISE	4
2.1	Modelando a DEA	5

1 INTRODUÇÃO

A atividade consiste na análise envoltória de dados considerando as 8 DMUs (*Decision Making Units*) da DEA. Cada DMU tem os seguintes insumos:

- MP: Matéria prima;
- MOD: Mão de obra direta;

E os seguintes produtos:

- Produtos: Quantidades produzidas no mês;
- Perfeição: Taxa de produtos sem defeito em %.

Uma DMU é classificada como completamente eficiente se, e somente se, a performance das outras DMUs não mostrarem que algum de seus insumos ou produtos pode ser melhorado sem piorar outro de seus insumos ou produtos, ou seja, se a DMU for Pareto-eficiente.

2 ANÁLISE

```
1 # importando dados
2 dados <- readxl::read_excel(
3   "data-raw/dea/EXERCICIO.xlsx",
4   sheet = "Planilha1"
5 )
6 print(dados)
```

```
# A tibble: 8 x 5
  DMU      MP  MOD Produtos Perfeição
<chr> <dbl> <dbl>    <dbl>    <dbl>
1 A      200  400    1000      90
2 B      300  500    1200      92
3 C      400  580    1300      95
4 D      350  500    1280      94
5 E      280  420    1100      89
6 F      300  200    1280      90
7 G      420  300    1800      85
8 H      192  300     980      97
```

```
1 # passando dados para o {deaR}
2 dados_dea <- deaR::make_deadata(
3   dados,
4   ni = 2,
5   no = 2,
6   dmus = 1,
7   inputs = 2:3,
8   outputs = 4:5
9 )
```

2.1 Modelando a DEA

No modelo orientado a insumo, os insumos dos DMUs eficientes irão permanecer os mesmos, uma vez que já são eficientes. Para os demais, os valores dos insumos são alterados para que eles se tornem eficientes.

Os DMUs eficientes são aqueles que possuem eficiência igual a 1 na saída da função `efficiencies()`. O valores ótimos dos insumos e produtos são obtidos com a função `targets()`.

Considerando retornos constantes de escala, F e H são os DMUs completamente eficientes.

```
1 # modelo orientado a insumo, crs
2 crs_insumo <- deaR::model_basic(
3   dados_dea,
4   # "io" significa "input oriented"
5   orientation = "io",
6   # rts significa "return to scale". "crs" significa "constant return to scale"
7   rts = "crs"
8 )
9
10 deaR::efficiencies(crs_insumo)
```

	A	B	C	D	E	F	G	H
	0.9795918	0.7836735	0.6501310	0.7345196	0.7785923	1.0000000	0.9940920	1.0000000

```
1 deaR::targets(crs_insumo)
```

`$target_input`

	MP	MOD
A	195.9184	306.1224
B	235.1020	367.3469
C	260.0524	377.0760
D	257.0819	367.2598
E	218.0058	327.0088
F	300.0000	200.0000
G	417.5186	298.2276
H	192.0000	300.0000

`$target_output`

	Produtos	Perfeição
A	1000	98.97959
B	1200	118.77551

C	1300	124.67901
D	1280	121.99288
E	1100	107.01721
F	1280	90.00000
G	1800	129.80992
H	980	97.00000

Já com retornos variáveis, temos C, F, G e H como DMUs completamente eficientes.

```

1 # modelo orientado a insumo, crs
2 vrs_insumo <- deaR::model_basic(
3   dados_dea,
4   # "io" significa "input oriented"
5   orientation = "io",
6   # rts significa "return to scale". "vrs" significa "variable return to scale"
7   rts = "vrs"
8 )
9
10 deaR::efficiencies(vrs_insumo)

```

	A	B	C	D	E	F	G	H
	0.9878049	0.8439024	1.0000000	0.9631169	0.8048780	1.0000000	1.0000000	1.0000000

```

1 deaR::targets(vrs_insumo)

```

\$target_input

	MP	MOD
A	197.5610	300.0000
B	253.1707	300.0000
C	400.0000	580.0000
D	337.0909	445.0909
E	225.3659	300.0000
F	300.0000	200.0000
G	420.0000	300.0000
H	192.0000	300.0000

\$target_output

	Produtos	Perfeição
A	1000	96.70732
B	1200	93.78049
C	1300	95.00000

D	1280	94.000000
E	1100	95.24390
F	1280	90.000000
G	1800	85.000000
H	980	97.000000

No modelo orientado a produto, são os valores dos produtos dos DMUs ineficientes que são alterados para que eles se tornem eficientes.

No modelo orientado a produto, considerando retornos constantes de escala, temos F e H como DMUs completamente eficientes.

```

1 # modelo orientado a produto, crs
2 crs_produto <- deaR::model_basic(
3   dados_dea,
4   # "io" significa "input oriented"
5   orientation = "oo",
6   # rts significa "return to scale". "crs" significa "constant return to scale"
7   rts = "crs"
8 )
9
10 deaR::efficiencias(crs_produto)

```

	A	B	C	D	E	F	G	H
	1.020833	1.276042	1.538151	1.361434	1.284369	1.000000	1.005943	1.000000

```

1 deaR::targets(crs_produto)

```

\$target_input

	MP	MOD
--	----	-----

A	200	312.50
---	-----	--------

B	300	468.75
---	-----	--------

C	400	580.00
---	-----	--------

D	350	500.00
---	-----	--------

E	280	420.00
---	-----	--------

F	300	200.00
---	-----	--------

G	420	300.00
---	-----	--------

H	192	300.00
---	-----	--------

\$target_output

	Produtos	Perfeição
--	----------	-----------

A	1020.833	101.0417
B	1531.250	151.5625
C	1999.597	191.7752
D	1742.636	166.0853
E	1412.806	137.4496
F	1280.000	90.0000
G	1810.698	130.5814
H	980.000	97.0000

Já com retornos variáveis, temos C, F, G e H como DMUs completamente eficientes.

```

1 # modelo orientado a produto, crs
2 vrs_produto <- deaR::model_basic(
3   dados_dea,
4   # "io" significa "input oriented"
5   orientation = "io",
6   # rts significa "return to scale". "vrs" significa "variable return to scale"
7   rts = "vrs"
8 )
9
10 deaR::efficiencias(vrs_produto)

```

	A	B	C	D	E	F	G	H
	0.9878049	0.8439024	1.0000000	0.9631169	0.8048780	1.0000000	1.0000000	1.0000000

```

1 deaR::targets(vrs_produto)

```

```

$target_input
      MP      MOD
A 197.5610 300.0000
B 253.1707 300.0000
C 400.0000 580.0000
D 337.0909 445.0909
E 225.3659 300.0000
F 300.0000 200.0000
G 420.0000 300.0000
H 192.0000 300.0000

```

```

$target_output
      Produtos Perfeição
A      1000  96.70732

```


B	1200	93.78049
C	1300	95.00000
D	1280	94.00000
E	1100	95.24390
F	1280	90.00000
G	1800	85.00000
H	980	97.00000