matRiks

An R package for the automatic generation of Raven-like matrices

Ottavia M. Epifania, Andrea Brancaccio, Debora de Chiusole, Pasquale Anselmi, Luca Stefanutti

University of Padova

European Meeting of the Matemathical Psychology Group

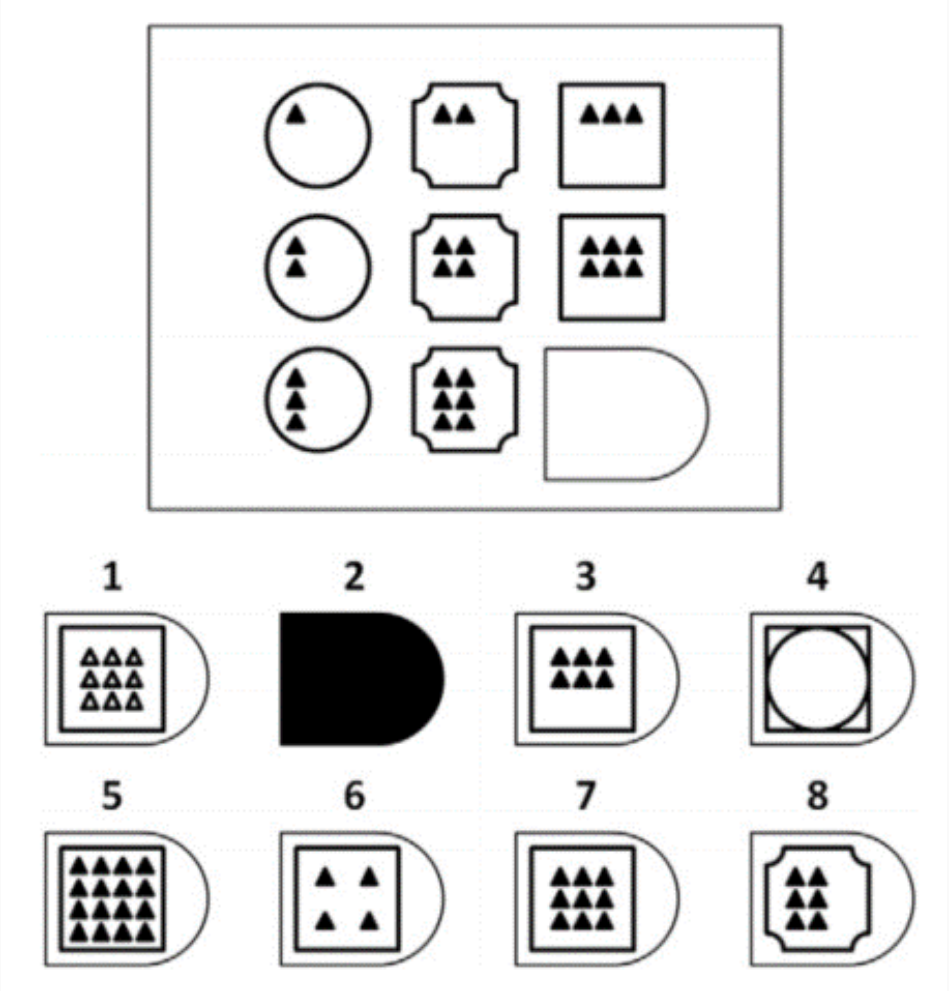
Raven and the generative rules

Measuring fluid intelligence without tapping into prior knowledge and by-passing what is learnt through acculturation... How?

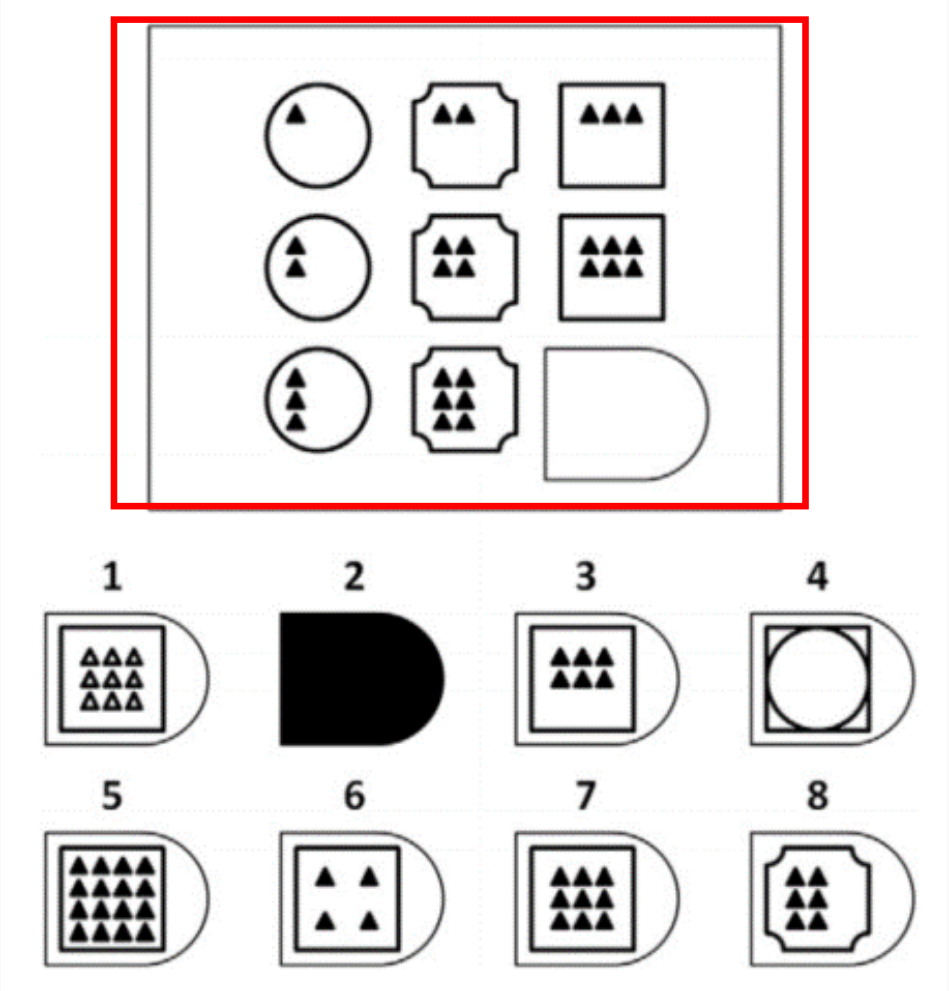
Visual analogies... But how?

Generative rules used for manipulating the visual and/or logical relationship between figures and objects

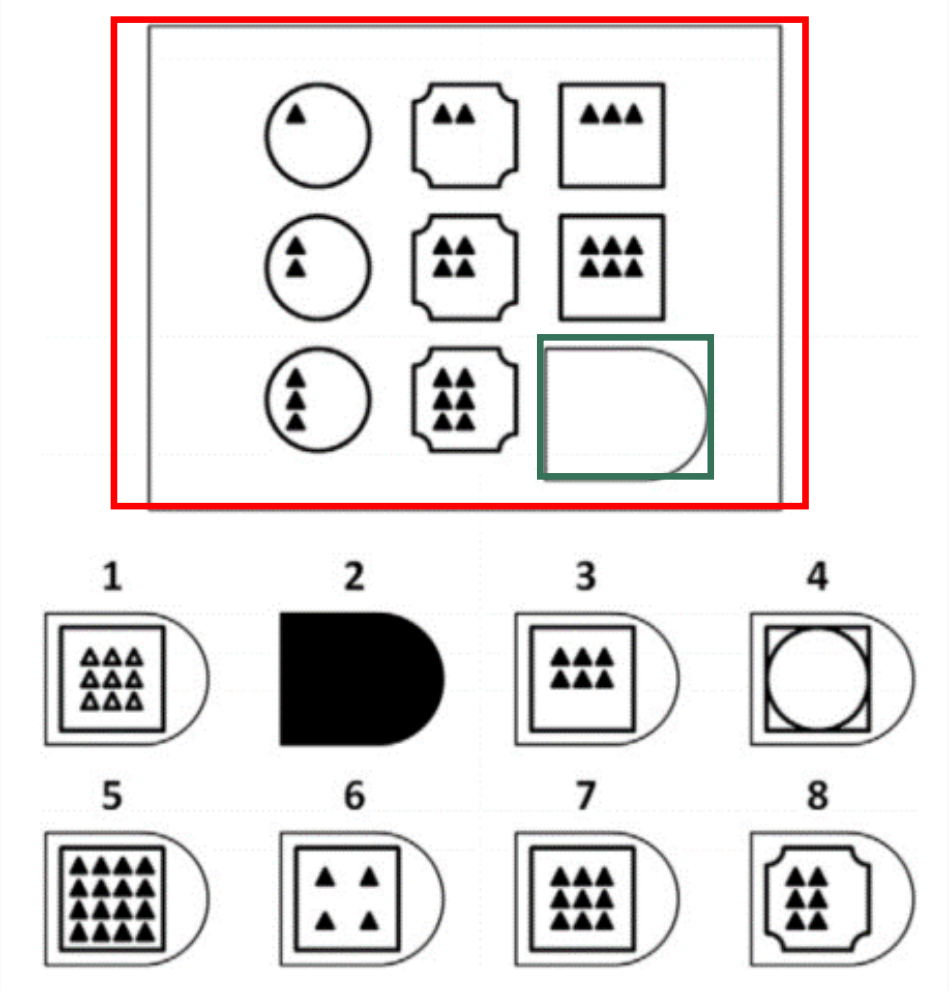
An example



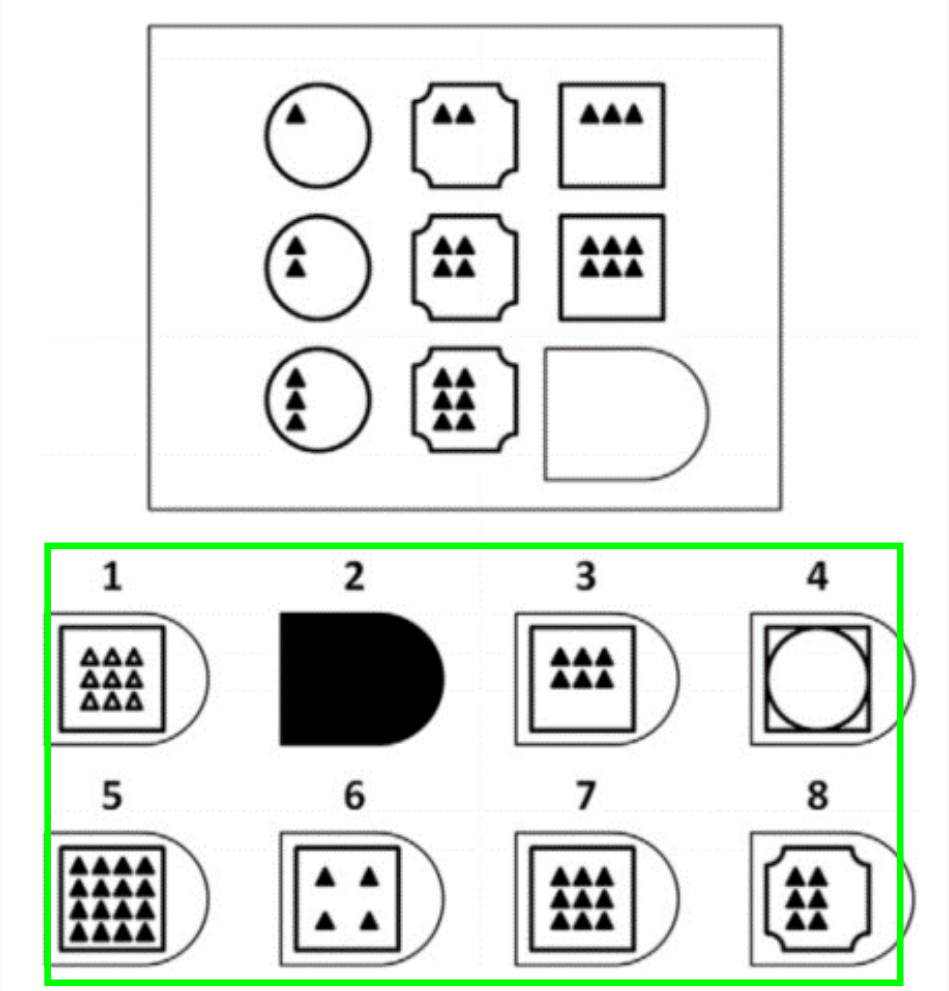
An example



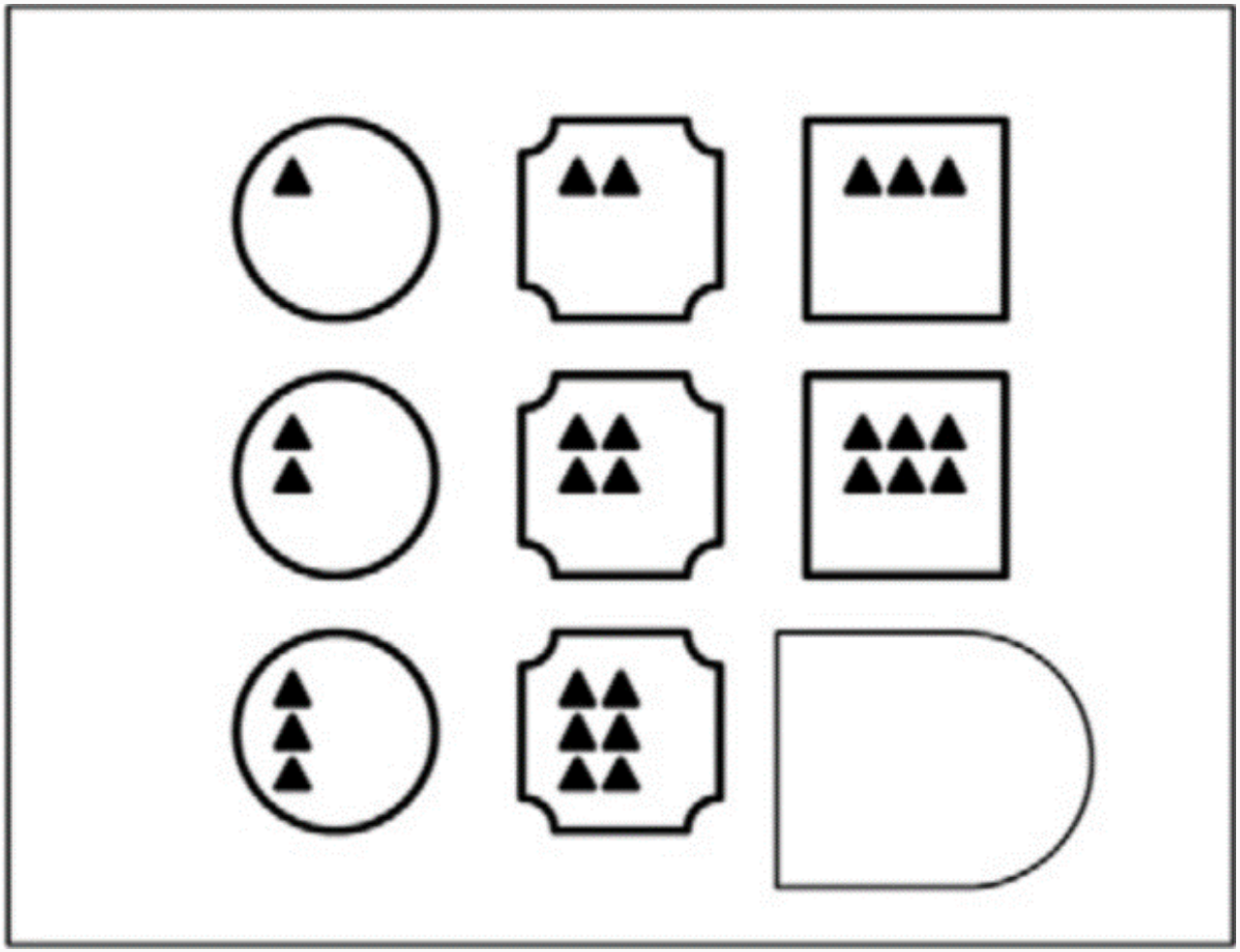
An example



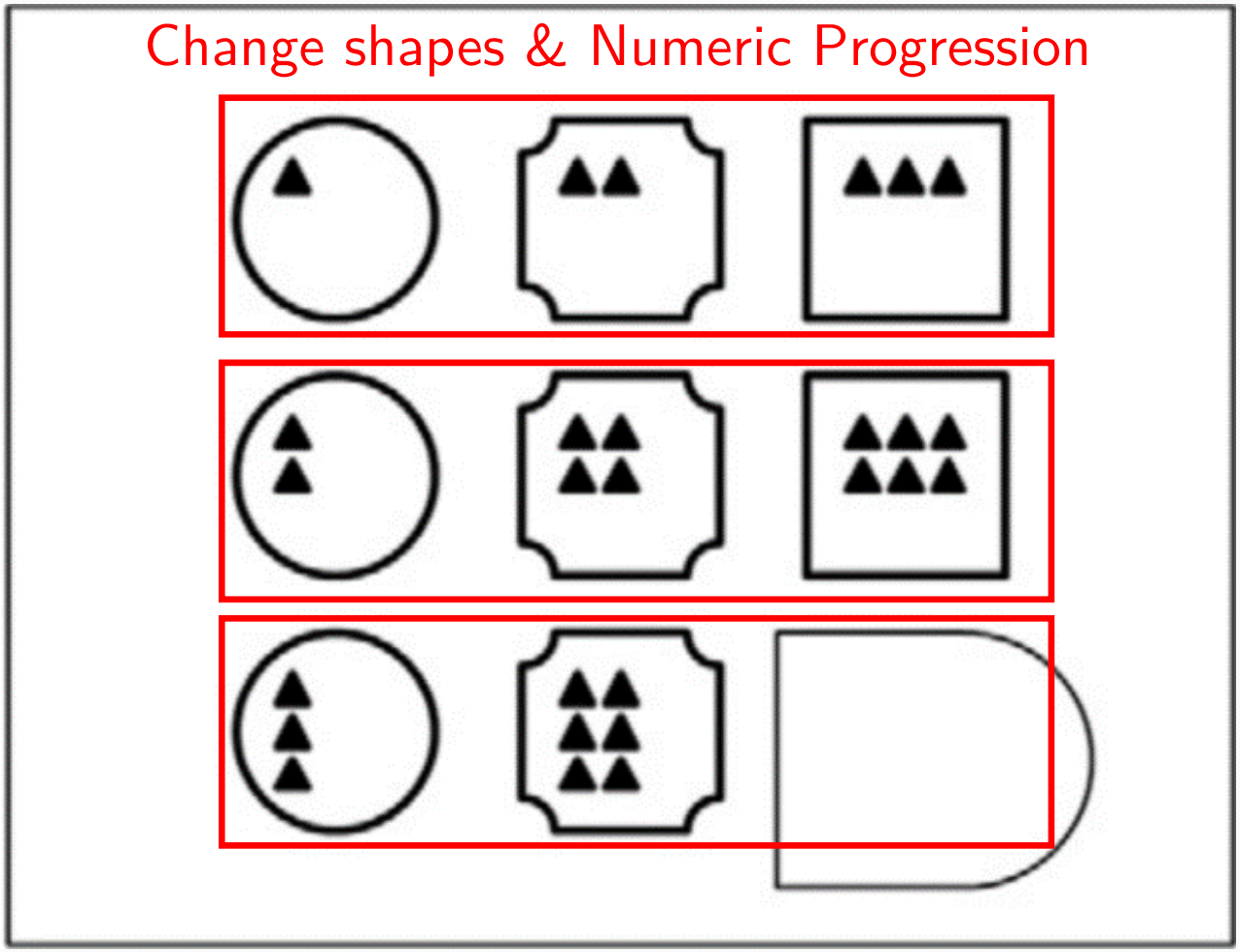
An example



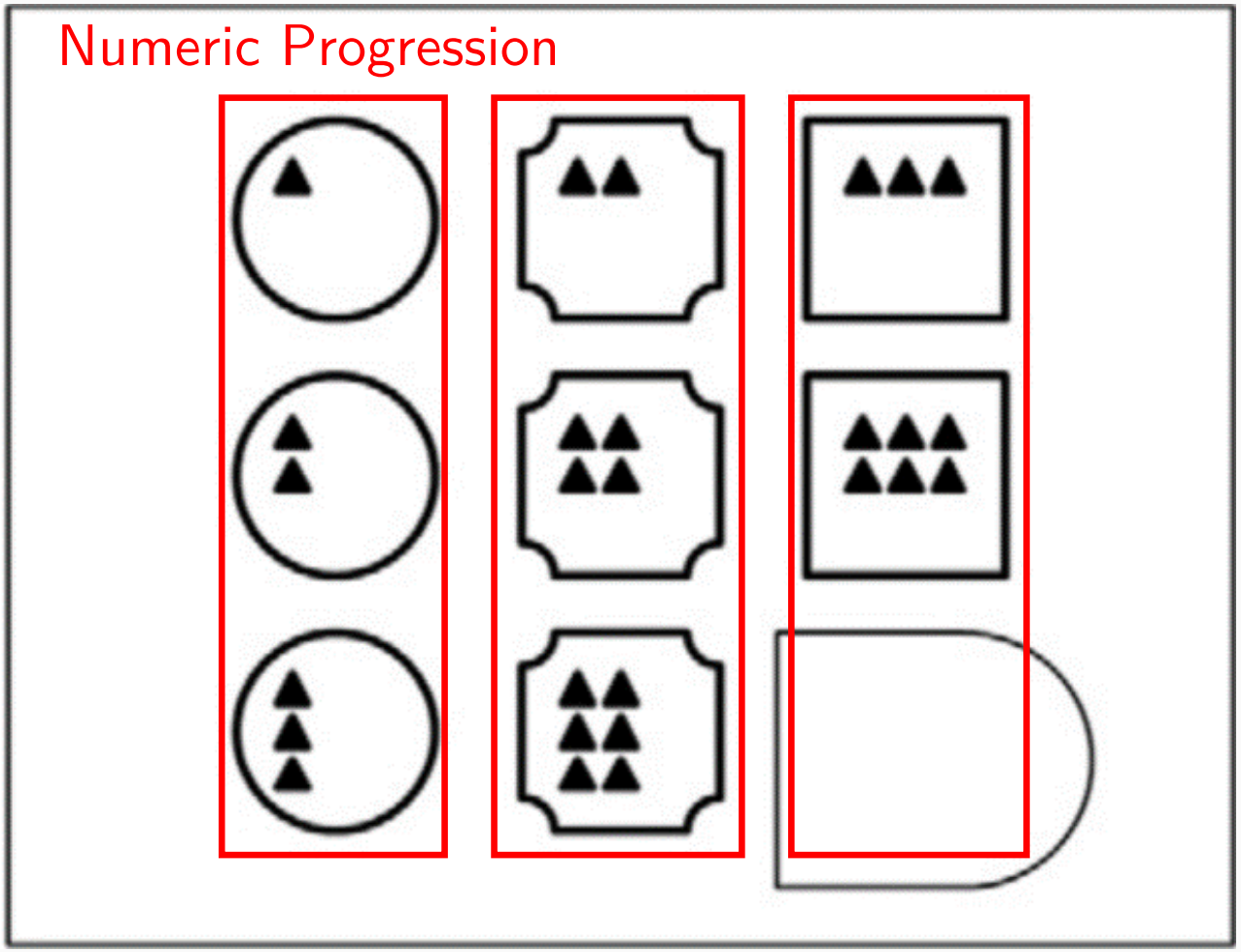
The matrix



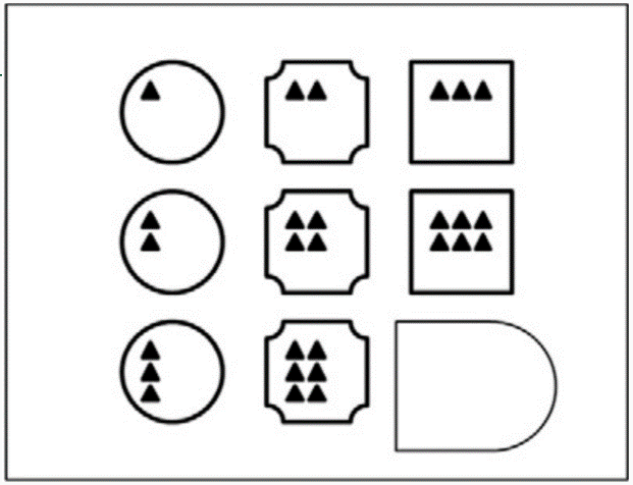
The matrix

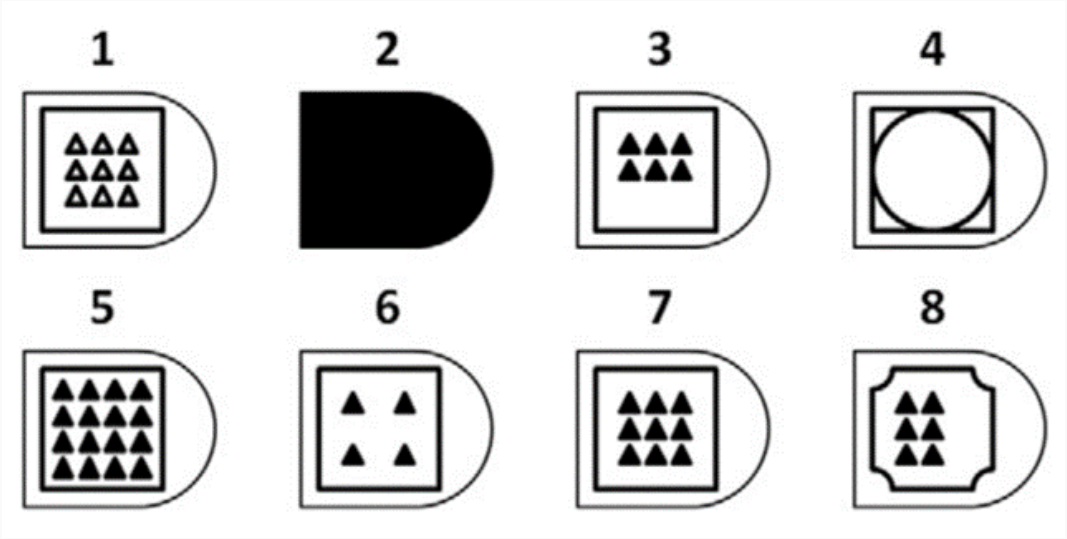


The matrix

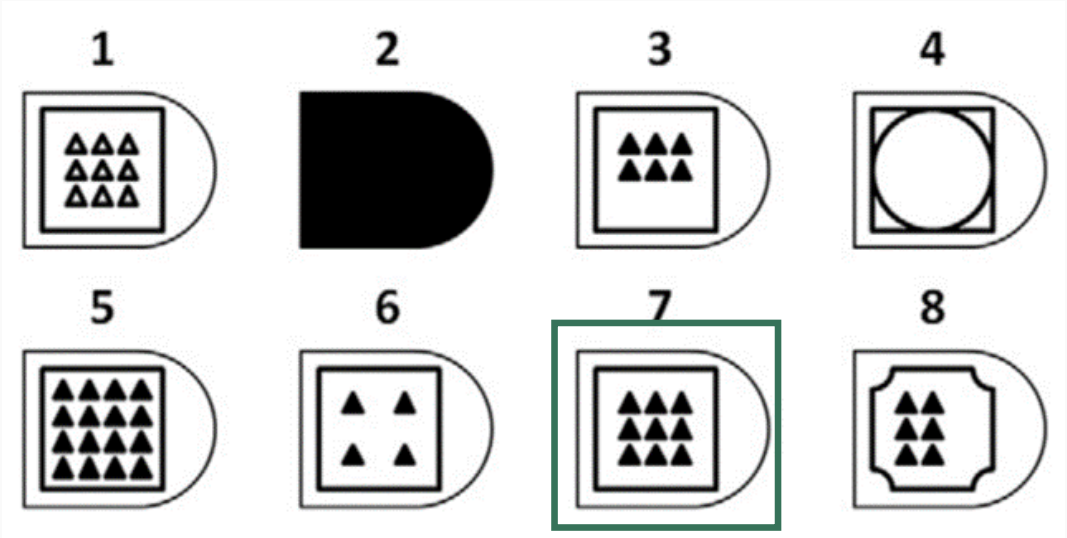


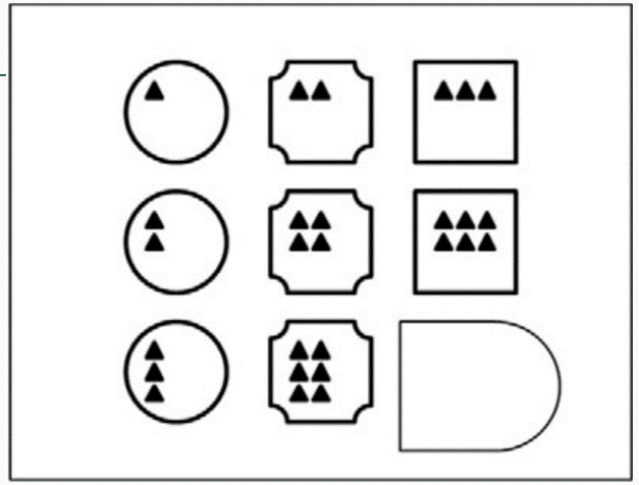
The response options



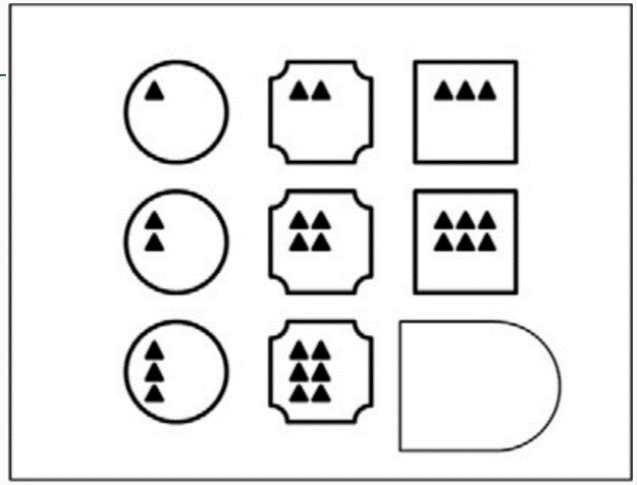


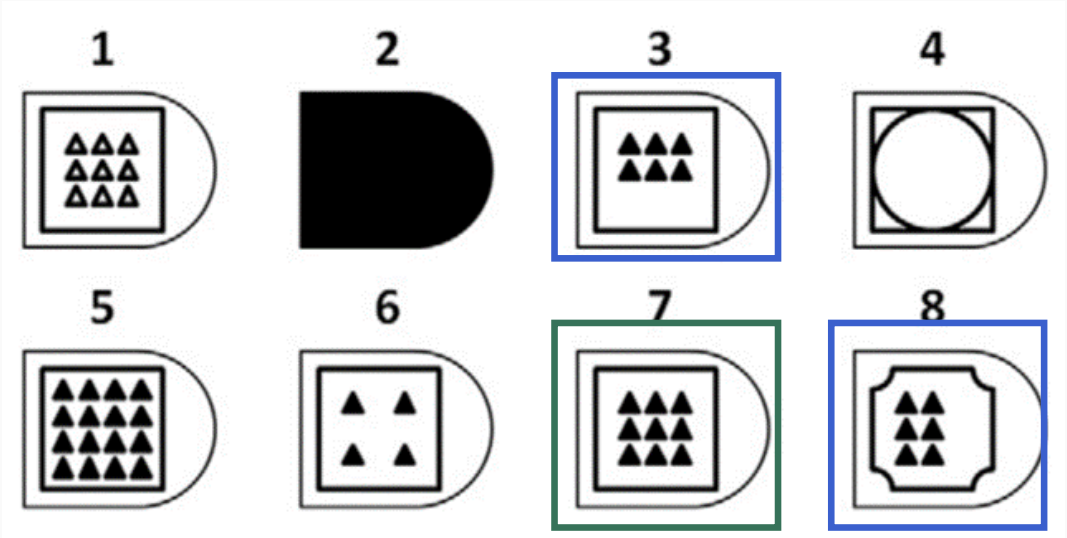
The response options





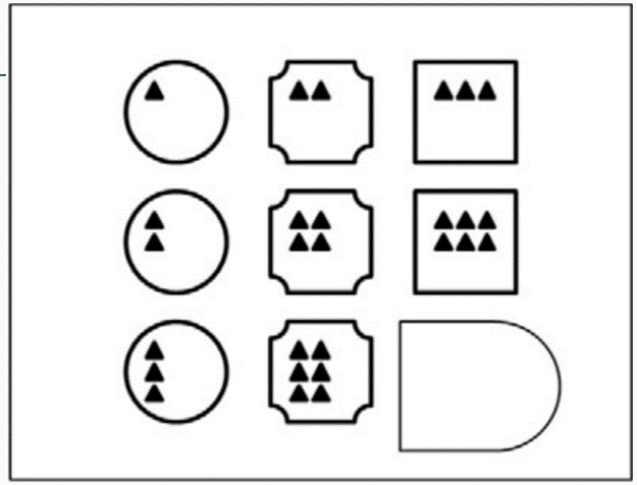
The response options

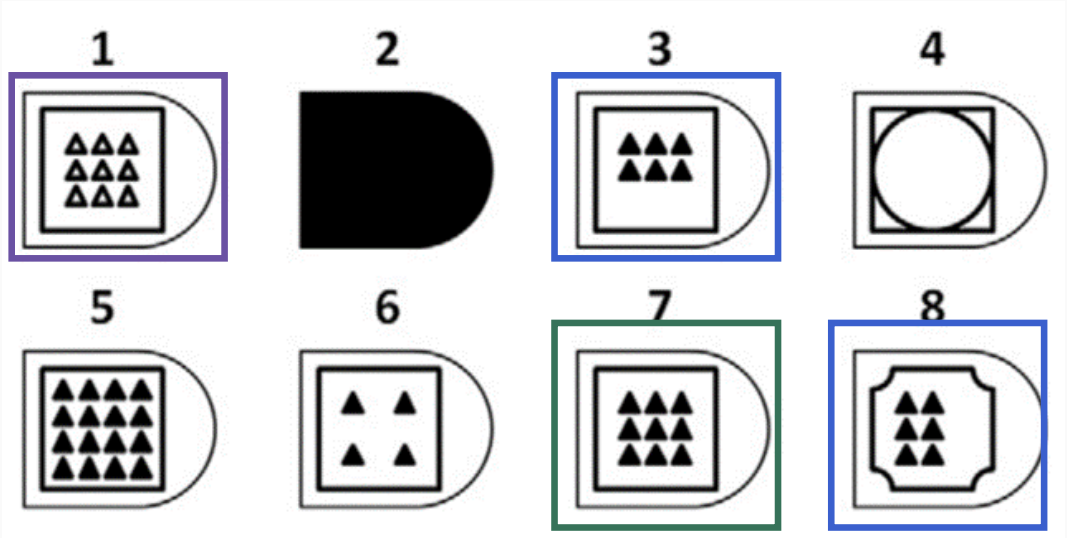




Repetition	Repetition of a cell adjacent to the empty cell
Incomplete Correlate	"Almost" the correct answer
Wrong Principle	An incorrect rule is used to solve the matrix
Difference	Pop-up effect

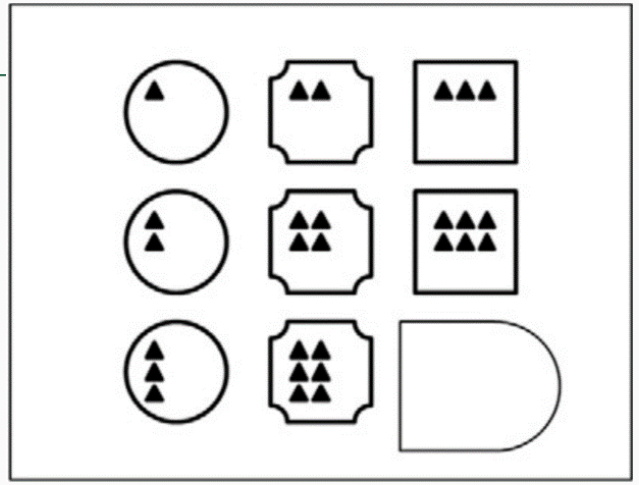
The response options

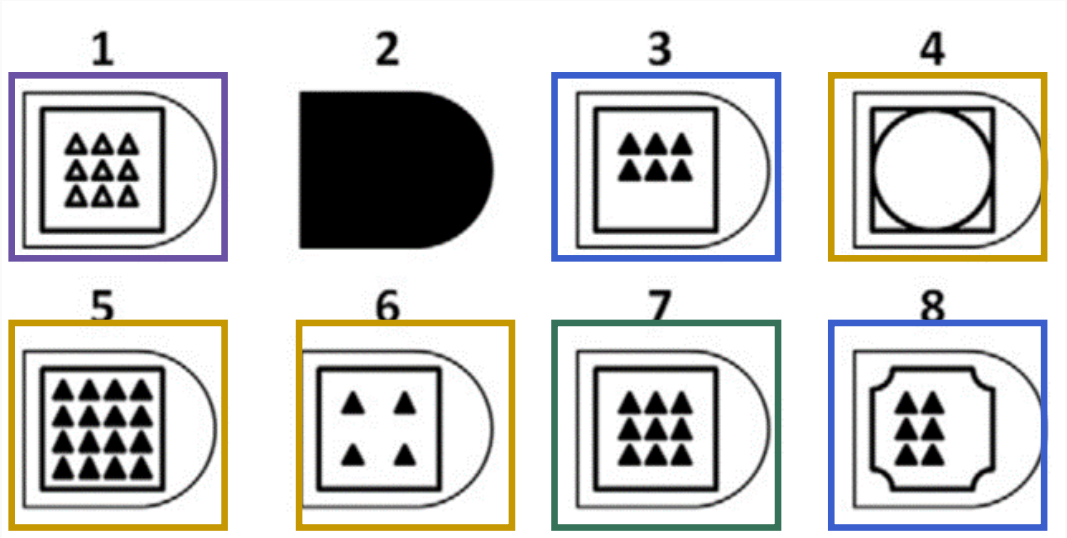




Repetition	Repetition of a cell adjacent to the empty cell
Incomplete Correlate	"Almost" the correct answer
Wrong Principle	An incorrect rule is used to solve the matrix
Difference	Pop-up effect

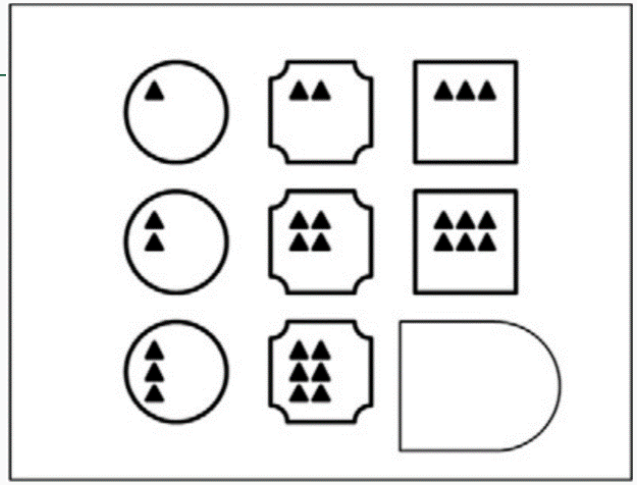
The response options

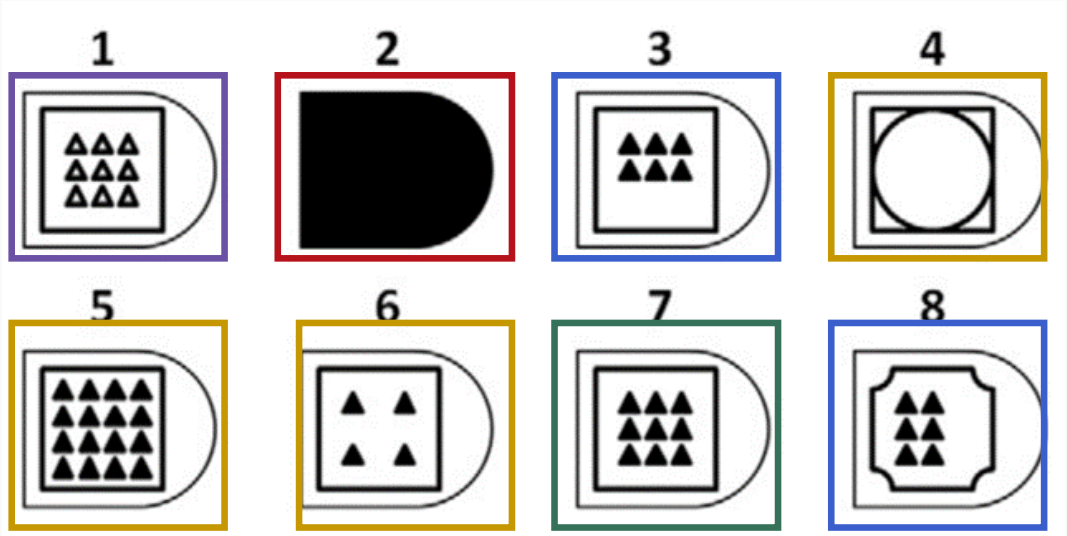




Repetition	Repetition of a cell adjacent to the empty cell
Incomplete Correlate	"Almost" the correct answer
Wrong Principle	An incorrect rule is used to solve the matrix
Difference	Pop-up effect

The response options





Repetition	Repetition of a cell adjacent to the empty cell
Incomplete Correlate	"Almost" the correct answer
Wrong Principle	An incorrect rule is used to solve the matrix
Difference	Pop-up effect

Generative rules

Category	Rule	Definition
Visuospatial	Completion	Identification of the missing portion of a figure
	Orientation	Manipulation of spatial orientation
	Shape	Manipulation of shape
	Filling	Manipulation of filling
	Size	Manipulation of size
Pre-inference	Object Addition	Overlapping objects present in different cells
	Object Subtraction	Deleting objects present in different cells
Logic	AND	The third cell is obtained from the intersection of the first two
	OR	The third cell is obtained from the union of the first two
	XOR	The third cell is obtained from the union of the first two but only for the elements that do not repeat
Directional Logic	Horizontal	The rules are applied horizontally
	Vertical	The rules are applied vertically
	Diagonal	The rules are applied simultaneously in vertical and horizontal

Distractors

Category	Definition	Specific	Definition
Repetition	Repetition of a cell adjacent to the empty cell	R-Left	Repetition of the cell to the left of the empty cell
		R-Top	Repetition of the cell above the empty cell
		R-Diag	Repetition of the top-left cell relative to the empty cell
Incomplete correlate	Correct answer with one element missing or modified	IC-Neg	Negative of the correct answer
		IC-Flip	Rotation of the correct answer (or one of its elements)
		IC-Size	Change in size of the correct answer (or one of its elements)
		IC-Inc	Correct answer with one element missing
Wrong Principle	Use of the wrong rule to solve the matrix	WP-Copy	Copy of a cell not adjacent to the empty cell
		WP-Matrix	Overlapping of two cells in the matrix
Difference	Pop-up effect		

Time has passed... but few open and easy-to-use resources are availaible for the generation of Raven's like matrices

Corvus

Sandia (prendi lo screenshot di Dedra)

A parte che sono brutte, i generatori o non funzionano o sono difficili da usare ma hanno tutti una roba in comune:

NON permettono le riproducibilità degli stimoli in modo semplice

matRiks

```
install.packages("matRiks")
library(matRiks)
# how to generate an RMarkdown file with your matrices!
vignette("generate_matriks")
```

Generates 2×2 and 3×3 Raven-like matrices and the related set of distractors

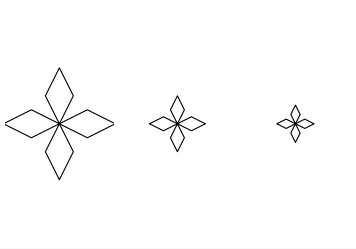
Allows for concatenating figures together

Allows for creating multi-layer matrices by combining concatenating single-layer matrices together

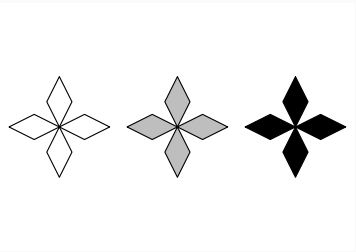
Allows for creating new figures from scratch

Visuo-spatial rules

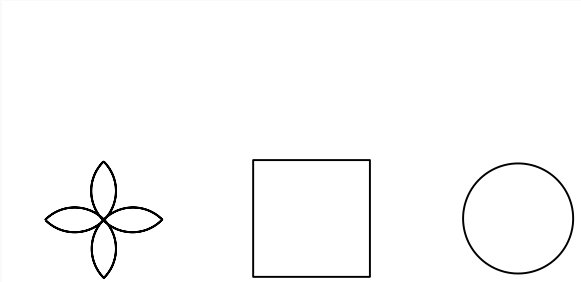
Size



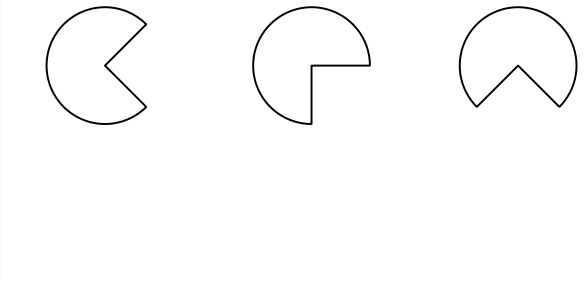
Shade



Shape



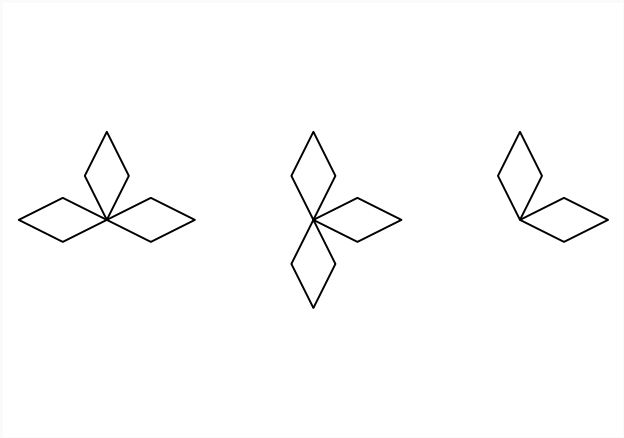
Rotate



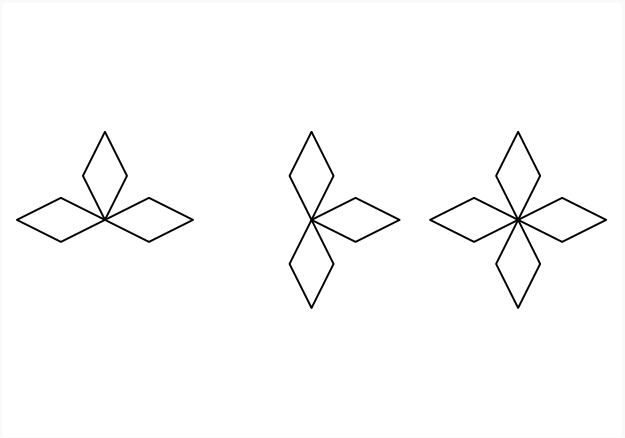
. . .

Logical rules

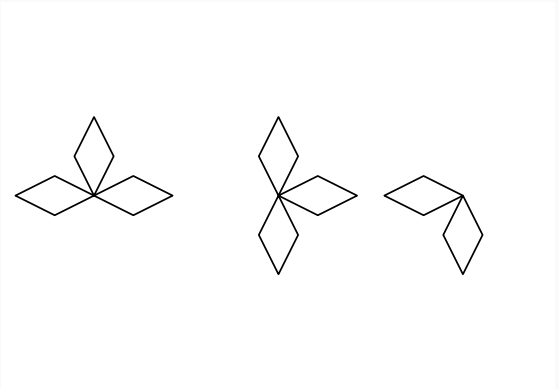
AND ($\cap$)



OR ($\cup$)



XOR (Δ)



The “workflow”

Qui farò un grafico con le frecce direzionate, ora non ho idea di come farlo

Choose a figure or a concatenation of figures

Choose the rule or a combination of rules to be applied vertically, horizontally, or diagonally.

Generate and draw the matrix

Generate the set of distractors

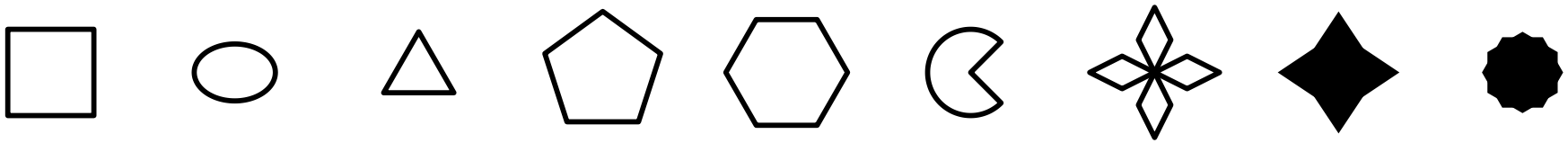
figure

```
$shape
[1] "square"

$size.x
$size.x[[1]]
[1] 15


$size.y
$size.y[[1]]
[1] 15


$theta.1
$theta.1[[1]]
[1] 0


$theta.2
$theta.2[[1]]
[1] 0
```

```
$rotation
$rotation[[1]]
[1] 0.7853982



$pos.x
$pos.x[[1]]
[1] 0


$pos.y
$pos.y[[1]]
[1] 0


$lty
$lty[[1]]
[1] 1


$lwd
$lwd[[1]]
[1] 3
```

```
$num
$num[[1]]
[1] 1


$nv
$nv[[1]]
[1] 4


$shade
$shade[[1]]
[1] NA


$visible
[1] 1

$tag
$tag[[1]]
[1] "simple" "fill"   "d.ex
```

Built-in figures



cof()

concatenation of figures

`single = FALSE`

```
draw(cof(square(),
    size(ninja())))
```

`single = TRUE`

```
draw(cof(square(),
    size(ninja())),
    single = TRUE,
    name = "my_figure")
```

cof()

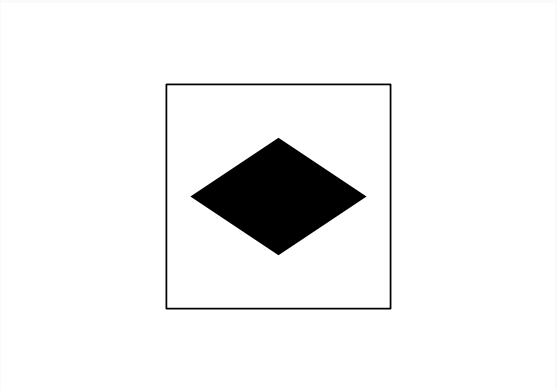
concatenation of figures

`single = FALSE`

```
draw(cof(square(),
     size(ninja())))
```

`single = TRUE`

```
draw(cof(square(),
     size(ninja())),
     single = TRUE,
     name = "my_figure")
```



```
List of 15
 $ shape   : chr [1:3] "square" "luck" "luck"
 $ size.x  :List of 3
  ..$ : num 15
  ..$ : num 5.56
  ..$ : num 5.56
 $ size.y  :List of 3
  ..$ : num 15
  ..$ : num 8.33
  ..$ : num 8.33
 $ theta.1 :List of 3
  ..$ : num 0
  ..$ : num 0
  ..$ : num 1.57
 $ theta.2 :List of 3
  ..$ : num 0
  ..$ : num 0
  ..$ : num 1.57
 $ rotation:List of 3
  ..$ : num 0.785
....
```

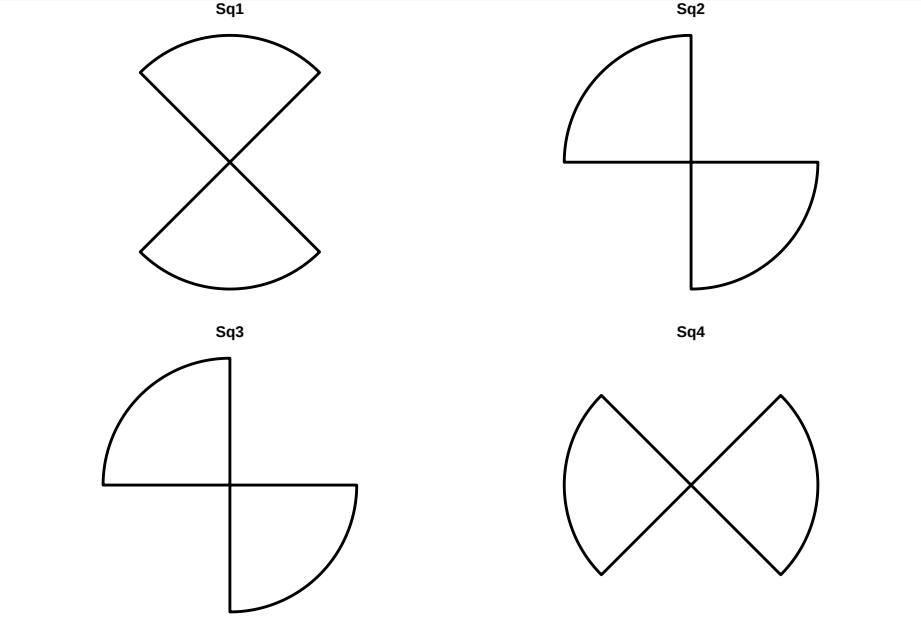
```
List of 15
 $ shape   : chr "my_figure"
 $ size.x  :List of 2
  ..$ : num [1:2] 15 5.56
  ..$ : num [1:2] 15 5.56
 $ size.y  :List of 2
  ..$ : num [1:2] 15 8.33
  ..$ : num [1:2] 15 8.33
 $ theta.1 :List of 2
  ..$ : num [1:2] 0 0
  ..$ : num [1:2] 0 1.57
 $ theta.2 :List of 2
  ..$ : num [1:2] 0 0
  ..$ : num [1:2] 0 1.57
 $ rotation:List of 2
  ..$ : num [1:2] 0.785 1.571
  ..$ : num [1:2] 0.785 3.142
 $ pos.x   :List of 2
  ..$ : num [1:2] 0 0
  ..$ : num [1:2] 0 0
....
```

mat_apply(): 2 × 2

```
mat_apply(Sq1, hrules, vrules, mat.type)
```

```
mat_apply(axe(),
          vrules = "rotate",
          hrules = "rotate",
          mat.type = 4)
```

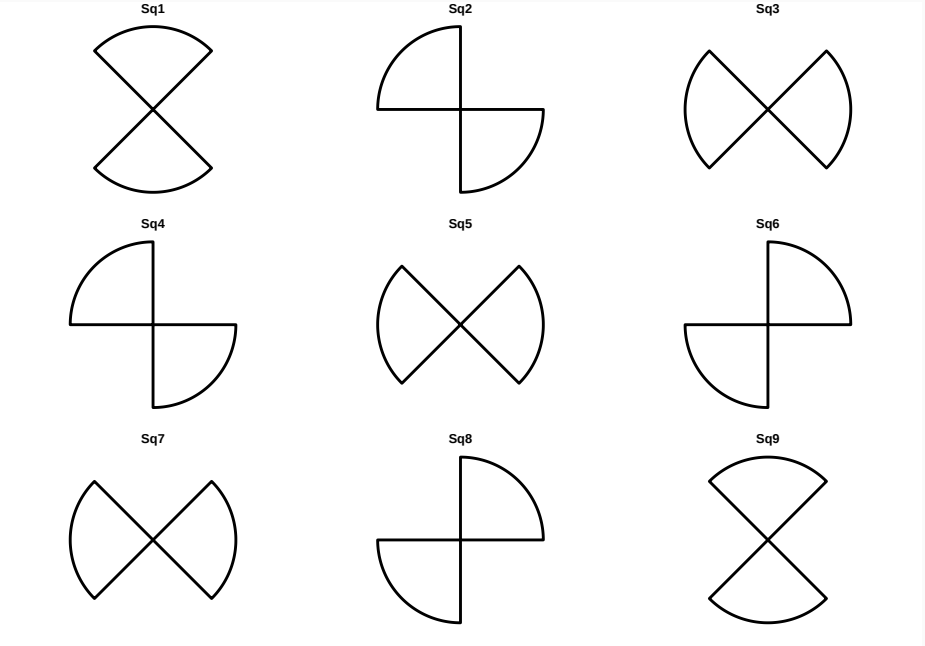
```
     [,1]  [,2]
[1,] "Sq1" "Sq2"
[2,] "Sq3" "Sq4"
```



mat_apply(): 3 × 3

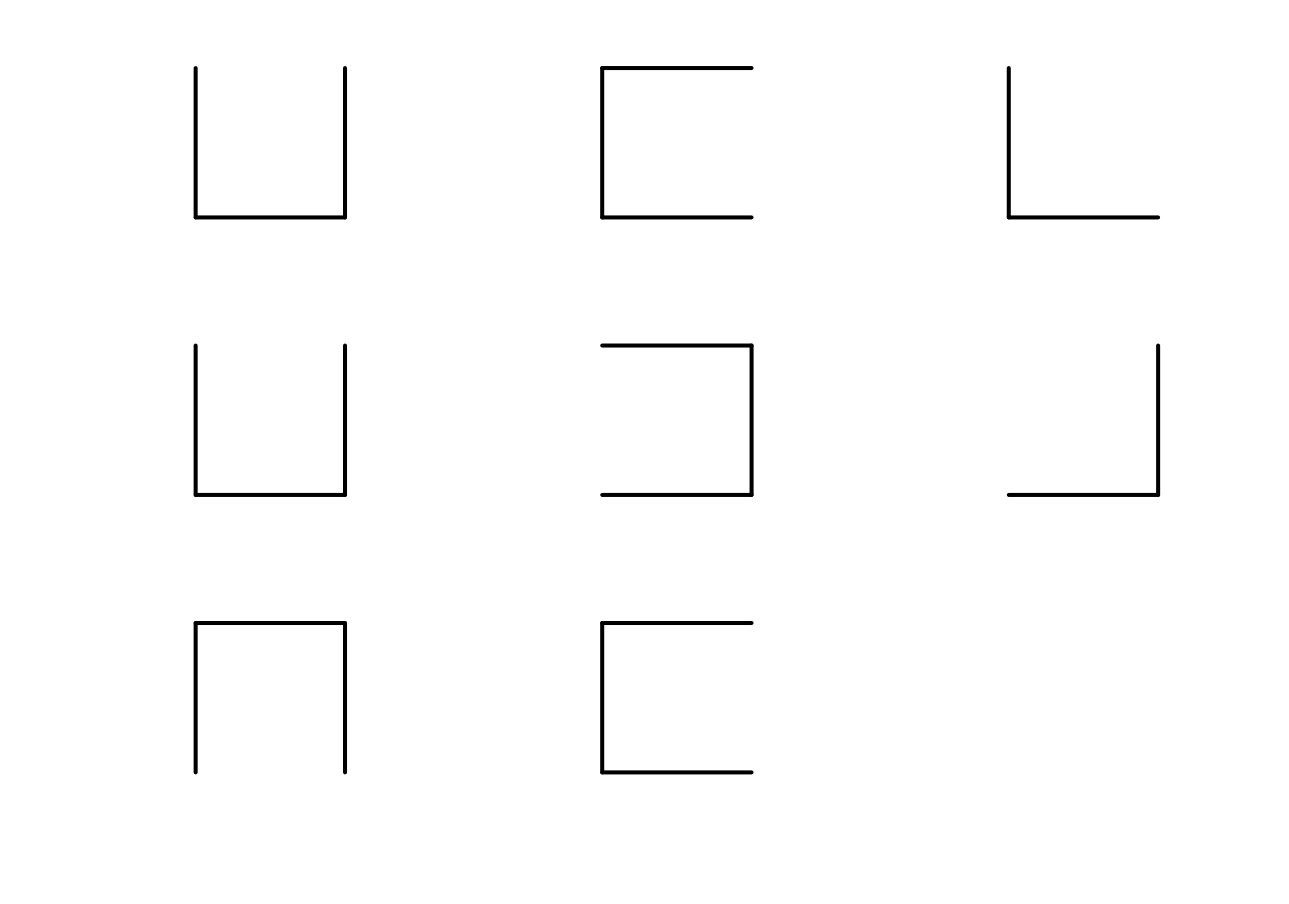
```
mat_apply(axe(),
          vrules = "rotate",
          hrules = "rotate",
          mat.type = 4)
```

```
     [,1]  [,2]  [,3]
[1,] "Sq1" "Sq2" "Sq3"
[2,] "Sq4" "Sq5" "Sq6"
[3,] "Sq7" "Sq8" "Sq9"
```



Single-layer vs. multi-layer matrices

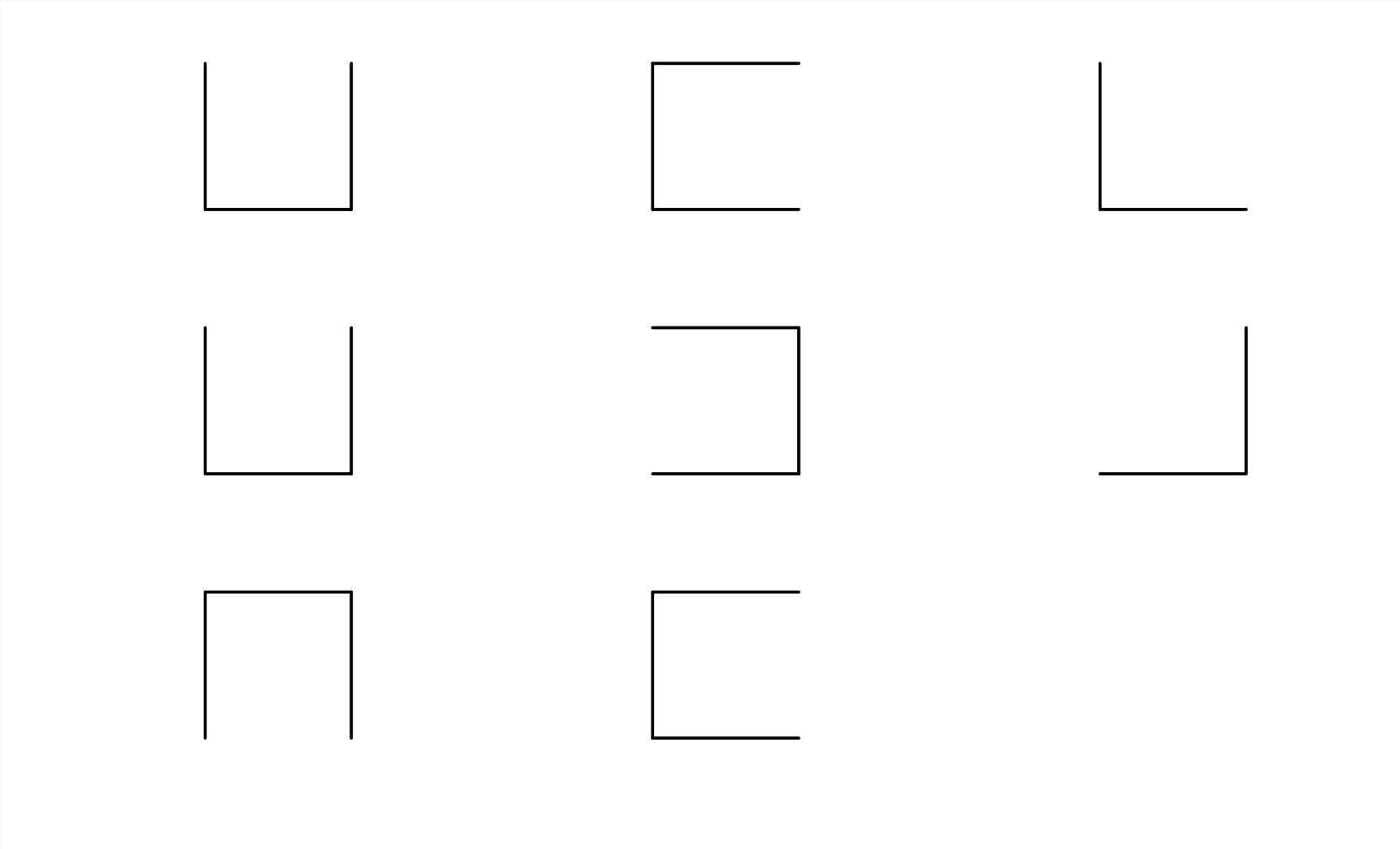
```
draw(mat_apply(cof(square4()), hrules = "AND"), hide = TRUE)
```



Single-layer vs. multi-layer matrices

concatenation of matrices: com()

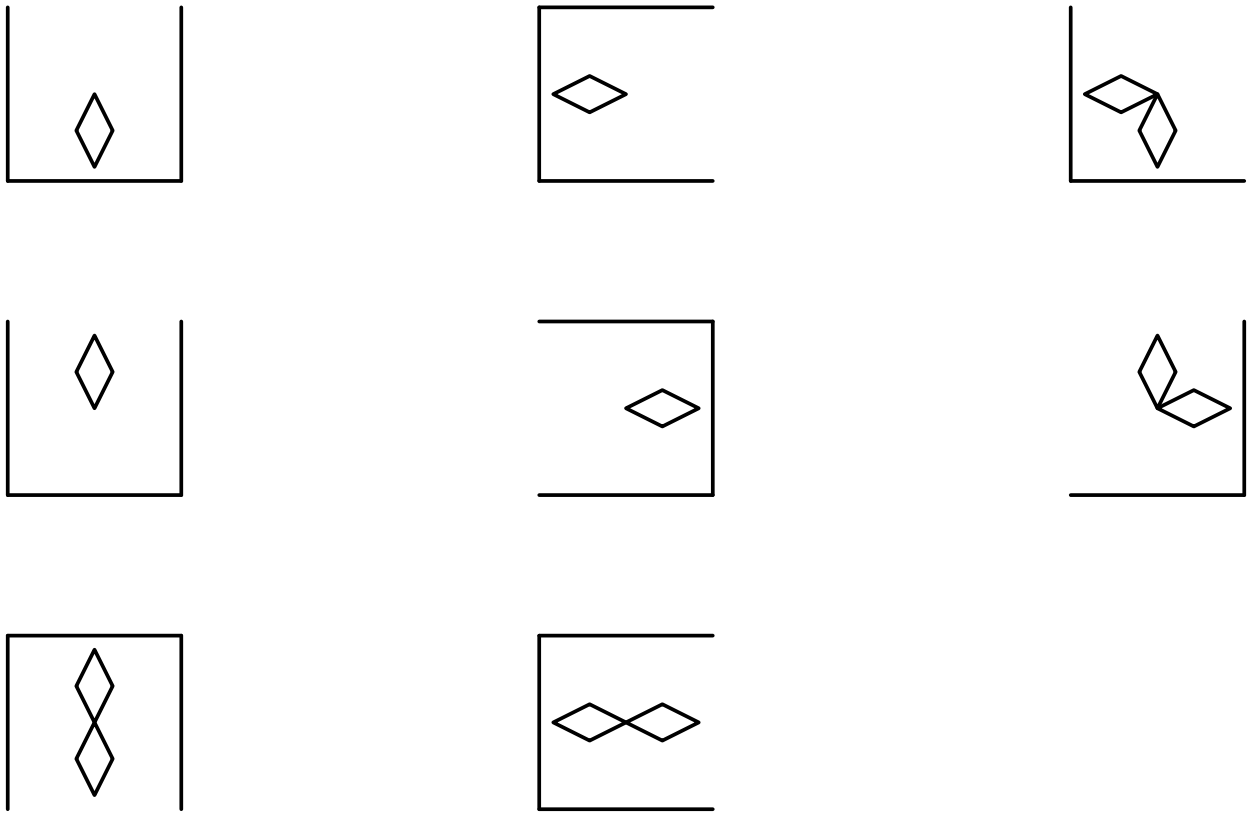
```
draw(com(mat_apply(square4(), hrules = "AND"),
         mat_apply(size(maxi(), 2), vrules = "OR", hrules = "OR")), hide = TRUE)
```



Single-layer vs. multi-layer matrices

concatenation of matrices: com()

```
draw(com(mat_apply(square4(), hrules = "AND"),
         mat_apply(size(maxi(), 2), vrules = "OR", hrules = "OR")), hide = TRUE)
```



The distractors

Distractors	Definition
R-Left	Sq8
R-Top	Sq6
R-diag	Sq5
Wp-Copy	One within SQ1, SQ2, SQ3, SQ4, SQ7
WP-Matrix	One within SQ1, SQ2, SQ3, SQ4, SQ7 with the superimposition of another cell.
Difference	One within SQ1, SQ2, SQ3, SQ4, SQ7 with the superimposition of a figure which is not manipulated in the matrix.

The distractors

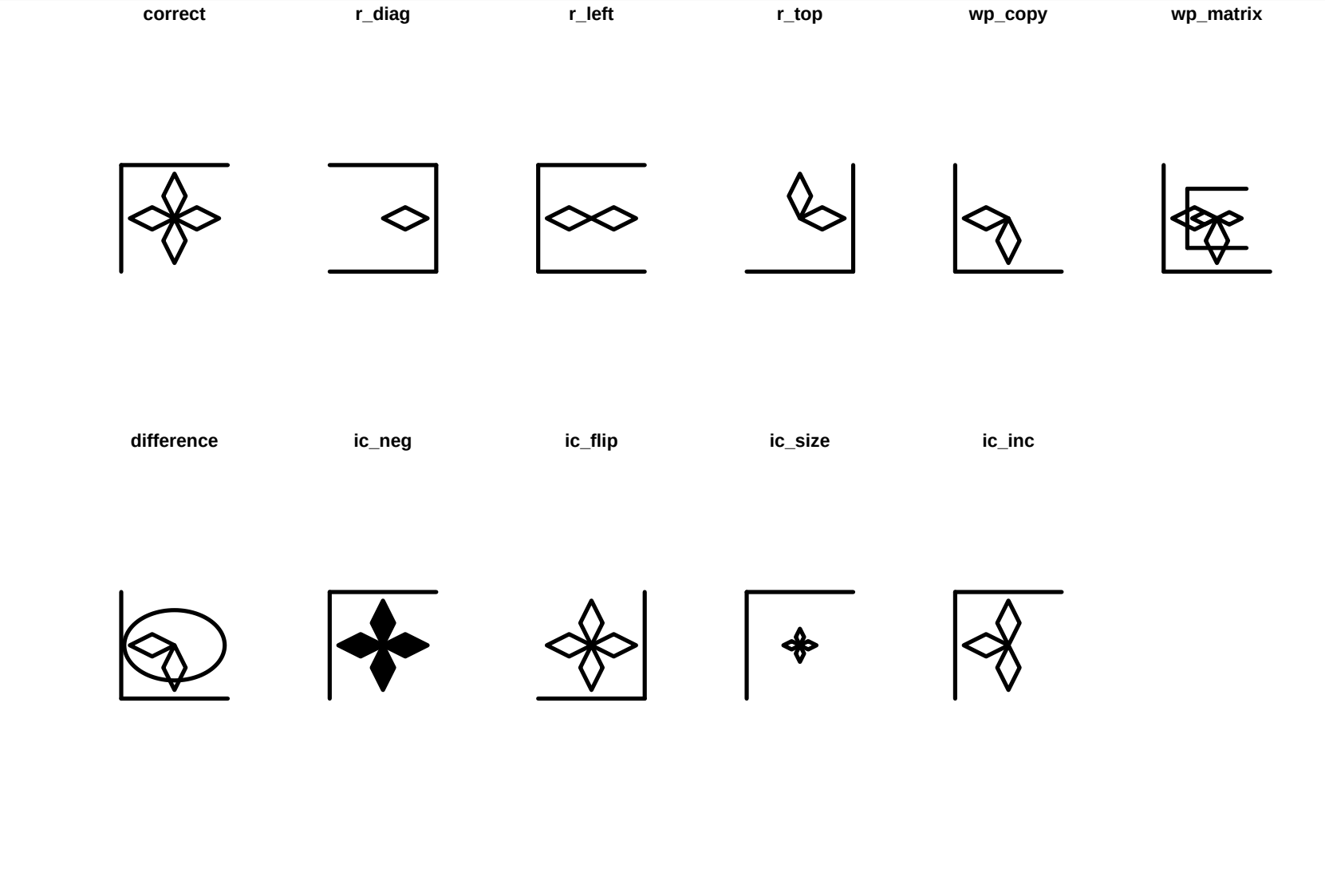
Distractors	Definition
R-Left	Sq8
R-Top	Sq6
R-diag	Sq5
Wp-Copy	One within SQ1, SQ2, SQ3, SQ4, SQ7
WP-Matrix	One within SQ1, SQ2, SQ3, SQ4, SQ7 with the superimposition of another cell.
Difference	One within SQ1, SQ2, SQ3, SQ4, SQ7 with the superimposition of a figure which is not manipulated in the matrix.
IC-Inc	Correct response with a missing element
	Single-Layer: Not possible
IC-Neg	Multi-layer: Color inversion of the correct response
	Single-layer matrix: Color inversion of the figure in the correct response
IC-Flip	Multi-layer matrix: Rotation of the correct response
	Single-layer matrix: Rotation of the figure in the correct response
IC-Scale	Multi-layer matrix: Resize of the correct response
	Single-layer: Resize of the figure in the correct response
	Multi-layer matrix:

The distractors

Distractors	Definition
R-Left	Sq8
R-Top	Sq6
R-diag	Sq5
Wp-Copy	One within SQ1, SQ2, SQ3, SQ4, SQ7
WP-Matrix	One within SQ1, SQ2, SQ3, SQ4, SQ7 with the superimposition of another cell.
Difference	One within SQ1, SQ2, SQ3, SQ4, SQ7 with the superimposition of a figure which is not manipulated in the matrix.
IC-Inc	Correct response with a missing element
	Single-Layer: Not possible
	Multi-layer: The most internal figure is removed from the correct response.
IC-Neg	Color inversion of the correct response
	Single-layer matrix: Color inversion of the figure in the correct response
	Multi-layer matrix: Color inversion of the foreground figure of the correct response
IC-Flip	Rotation of the correct response
	Single-layer matrix: Rotation of the figure in the correct response
	Multi-layer matrix: Rotation of the foreground figure of the correct response
IC-Scale	Resize of the correct response
	Single-layer: Resize of the figure in the correct response
	Multi-layer matrix: Resize of the foreground figure of the correct response

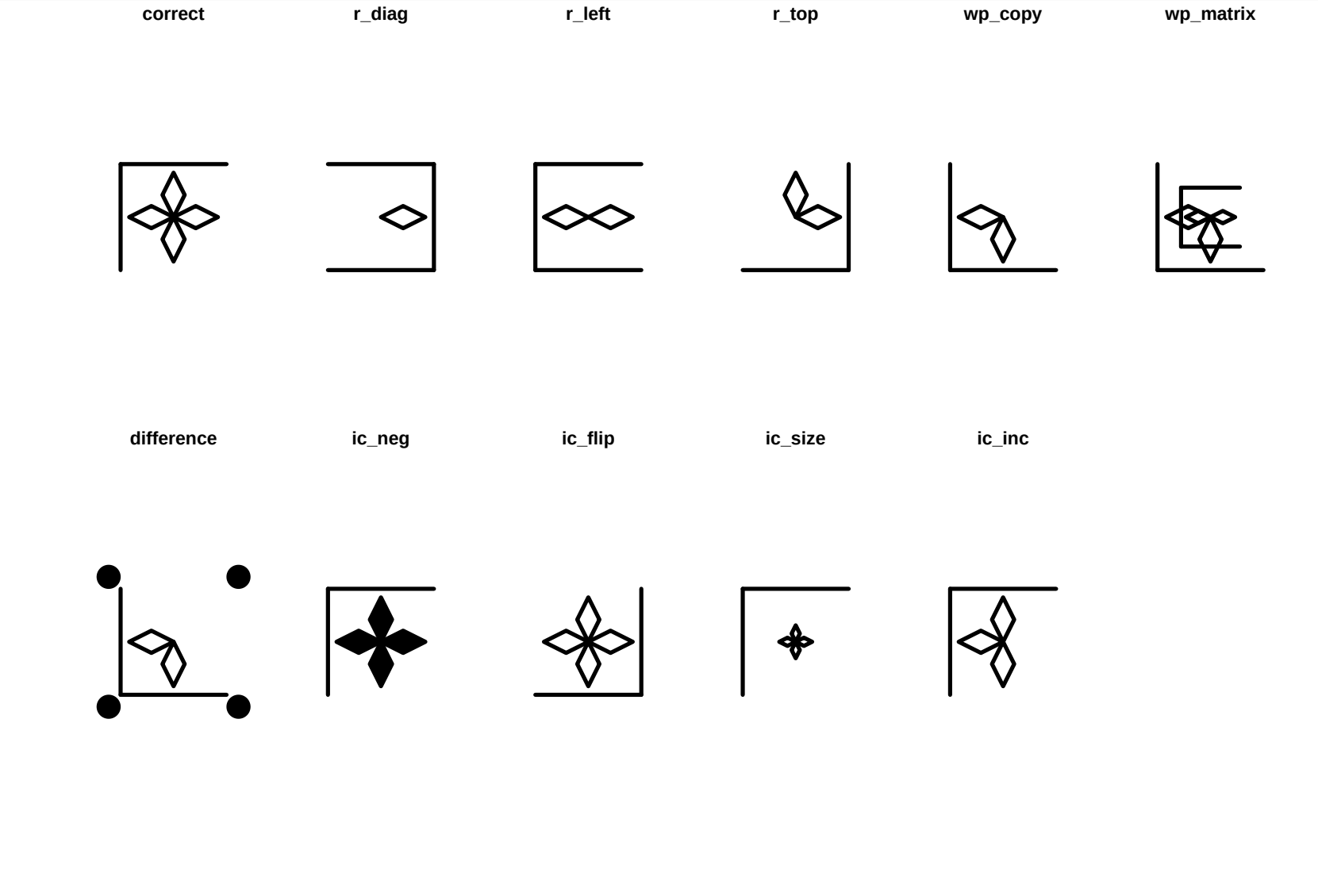
response_list()

```
draw(response_list(my_mat), main = TRUE)
```



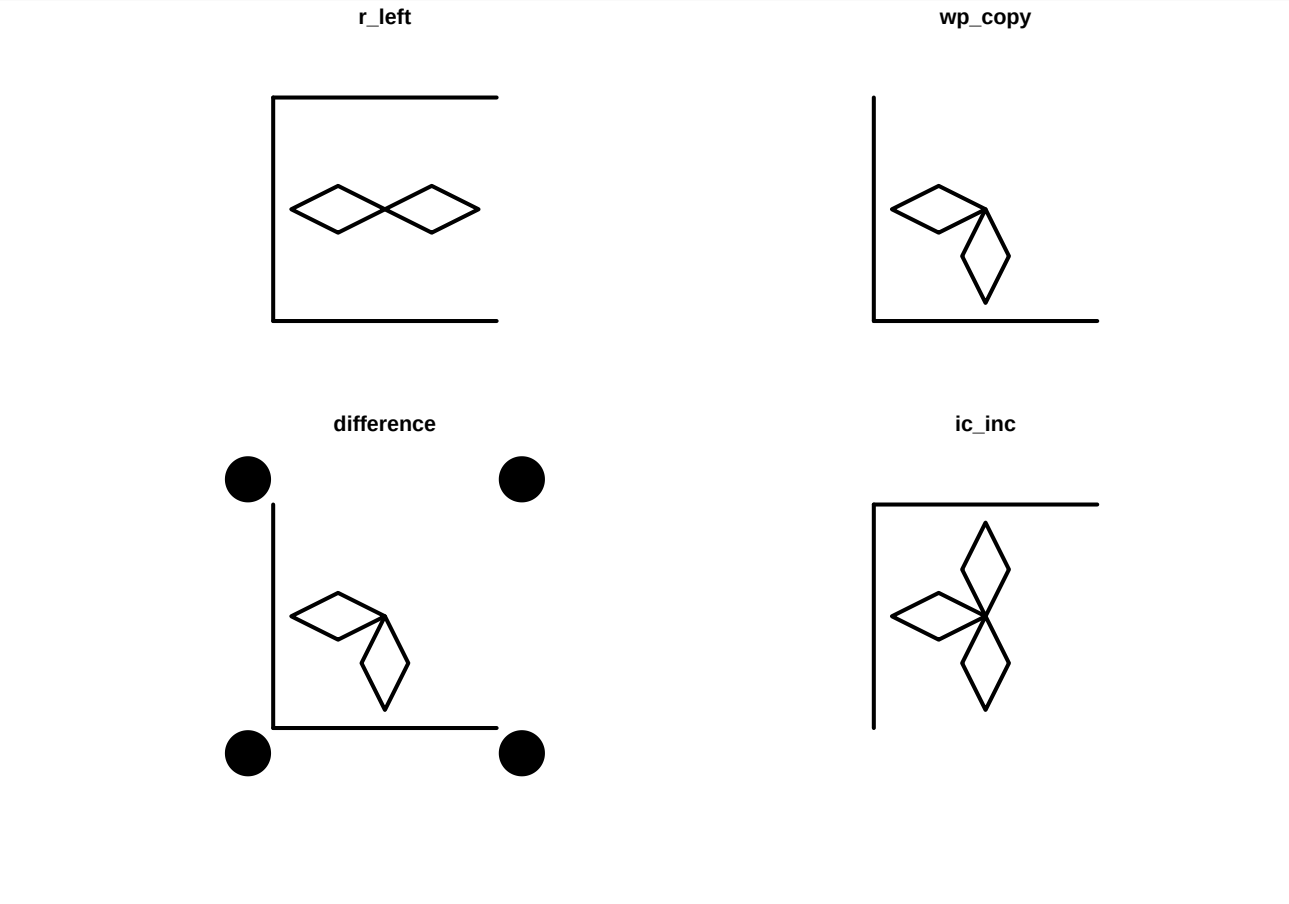
Don't like the difference distractor?

Change the random seed

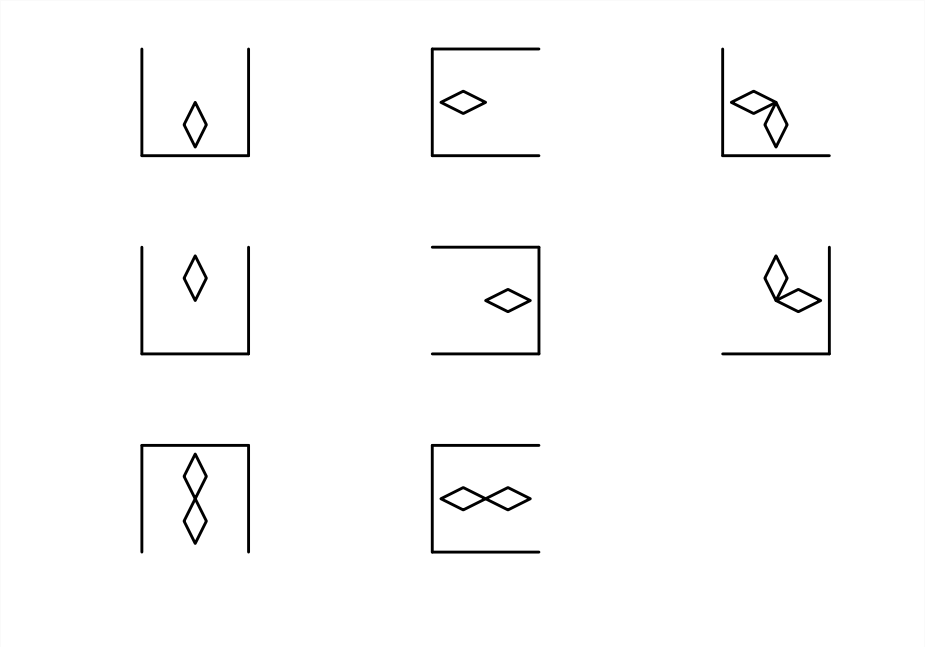


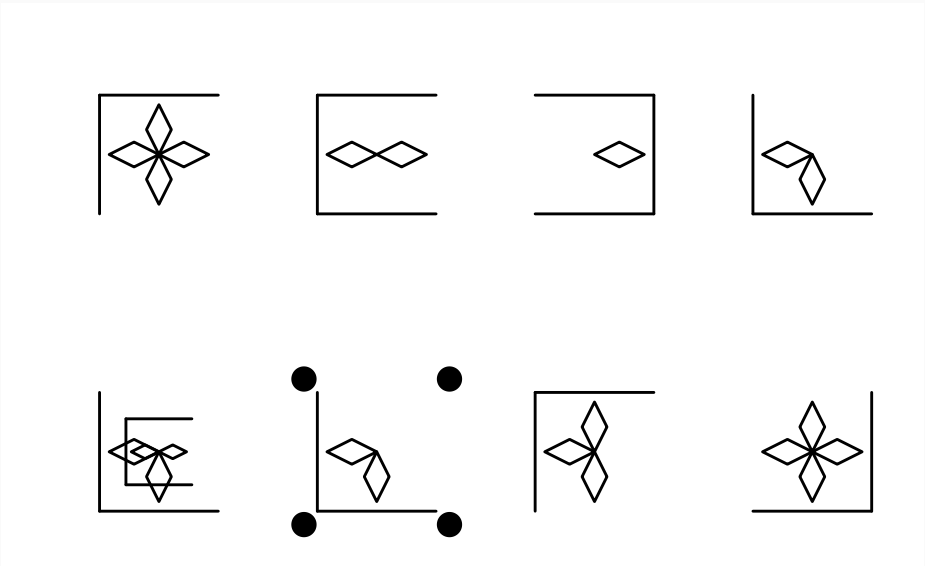
A handful of distractors

```
draw(response_list(my_mat, seed = 17),
     main = TRUE, distractors = c("r_left", "wp_copy",
                                  "difference", "ic_inc"))
```



The final result





PsycAssist


Psyc Assist
Login
Home
Il progetto di ricerca
Il team di ricerca
Collaborazioni
PsycAssist
A Psycological Assistent for accurate and adaptive neuropsychological assessments
Missione del progetto
Sviluppare un sistema intelligente di web-app per la valutazione neuropsicologica che somministra test, raccoglie e analizza dati, fornisce report personalizzati comprensivi di suggerimenti per la riabilitazione.

PsycAssist

Campione

le scuole

Come è andataa con rasch