Perché non generare test e questionari open... dal principio?

Risorse open per generare matrici di tipo Raven

Ottavia M. Epifania, Andrea Brancaccio, Debora de Chiusole

University of Padova

European Meeting of the Matemathical Psychology Group

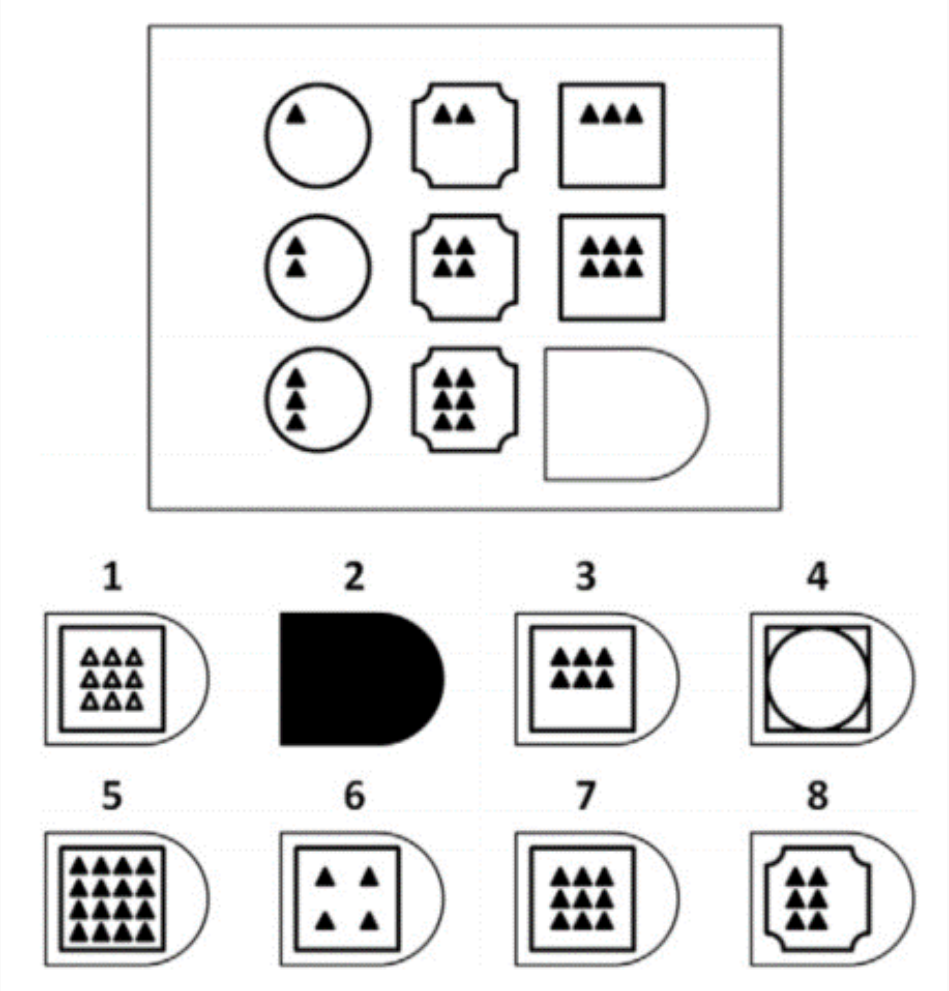
Raven e le regole generative

Misurare l'intelligenza fluida senza andare a toccare le conoscenze pregresse e bypassando tutto quello che si è appreso con il processo di acculturazione... ma come?

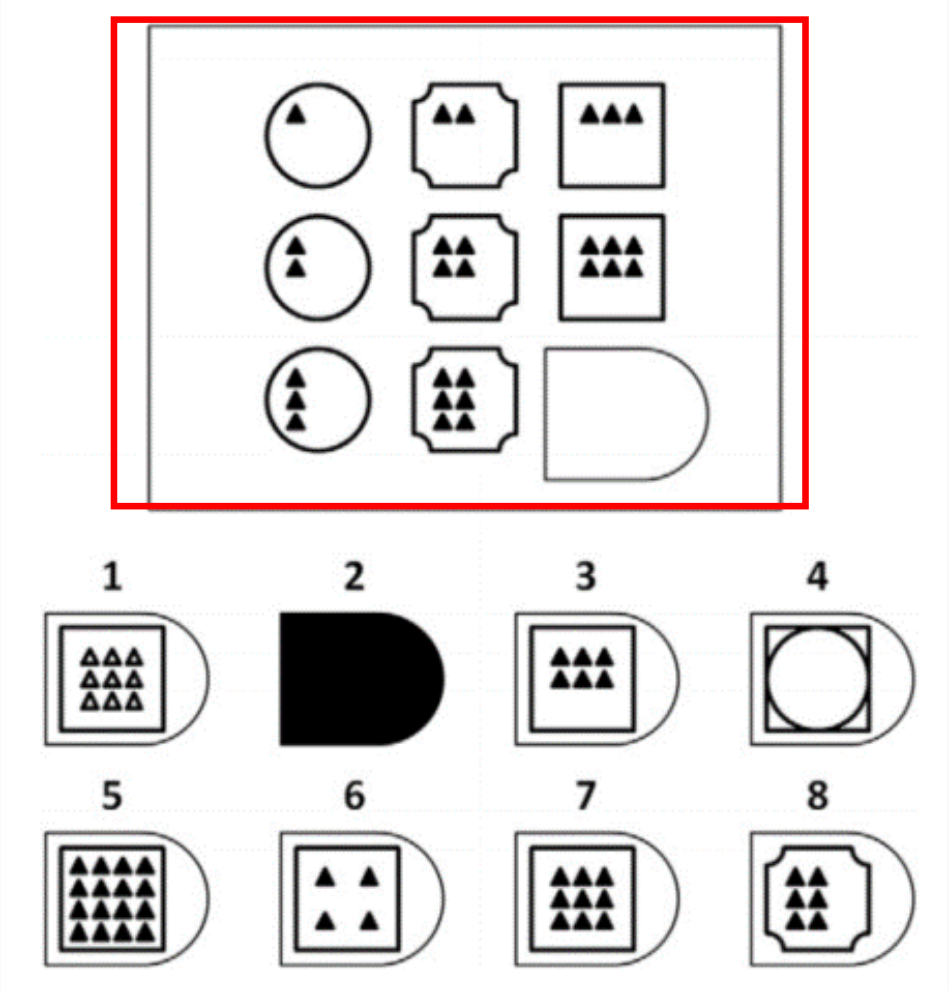
Analogie visive.. ma come?

Regole generative che vengono utilizzate per manipolare i rapporti visuo-spaziali o logici tra figure e oggetti

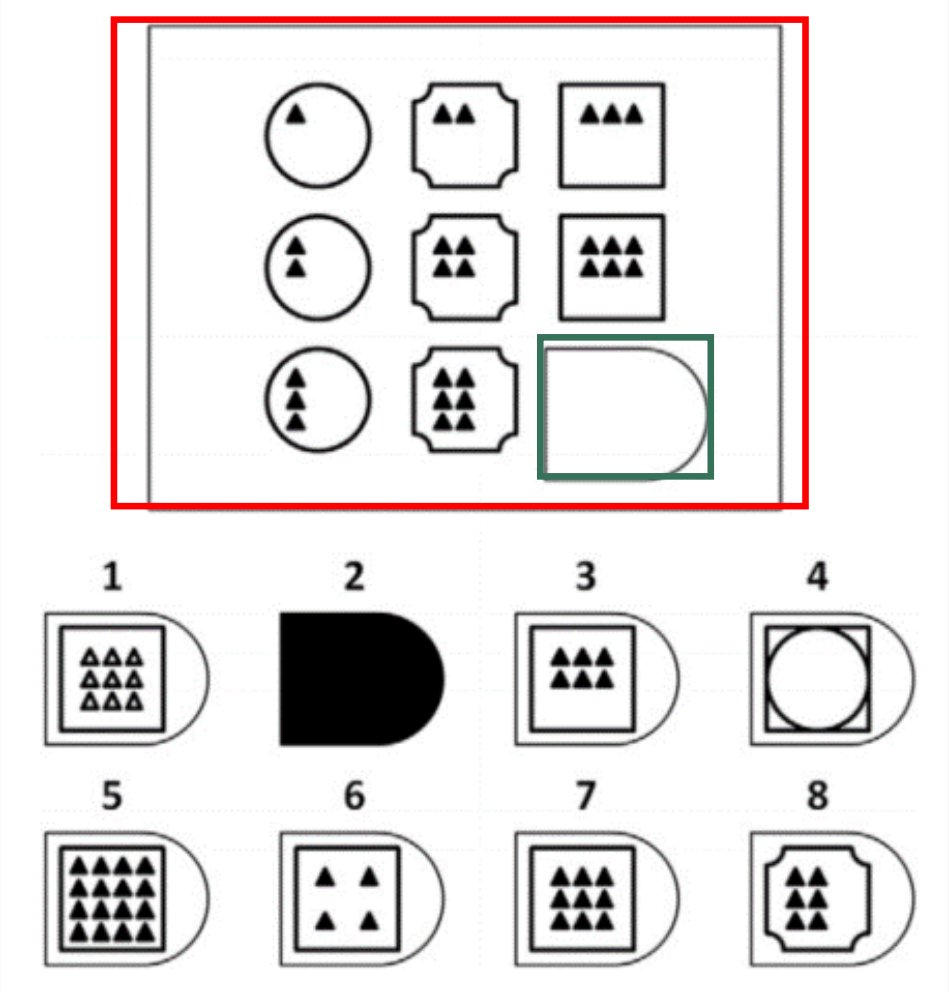
Un esempio



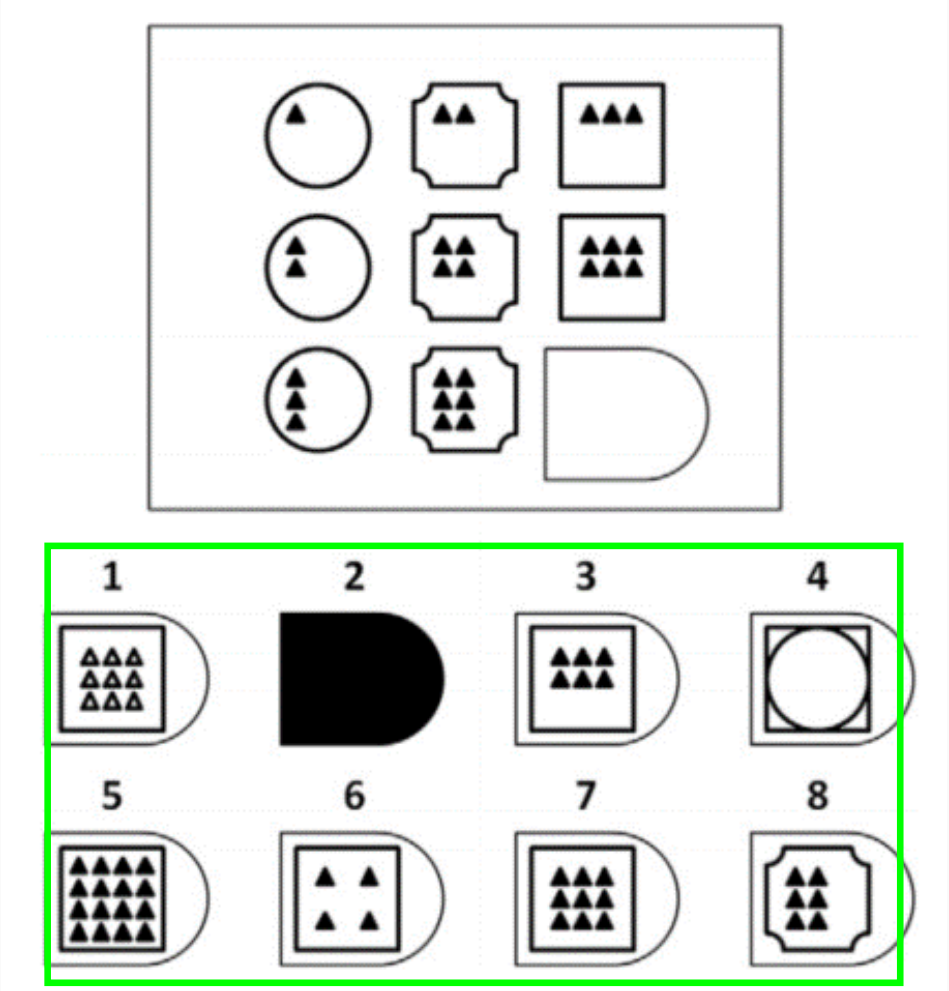
Un esempio



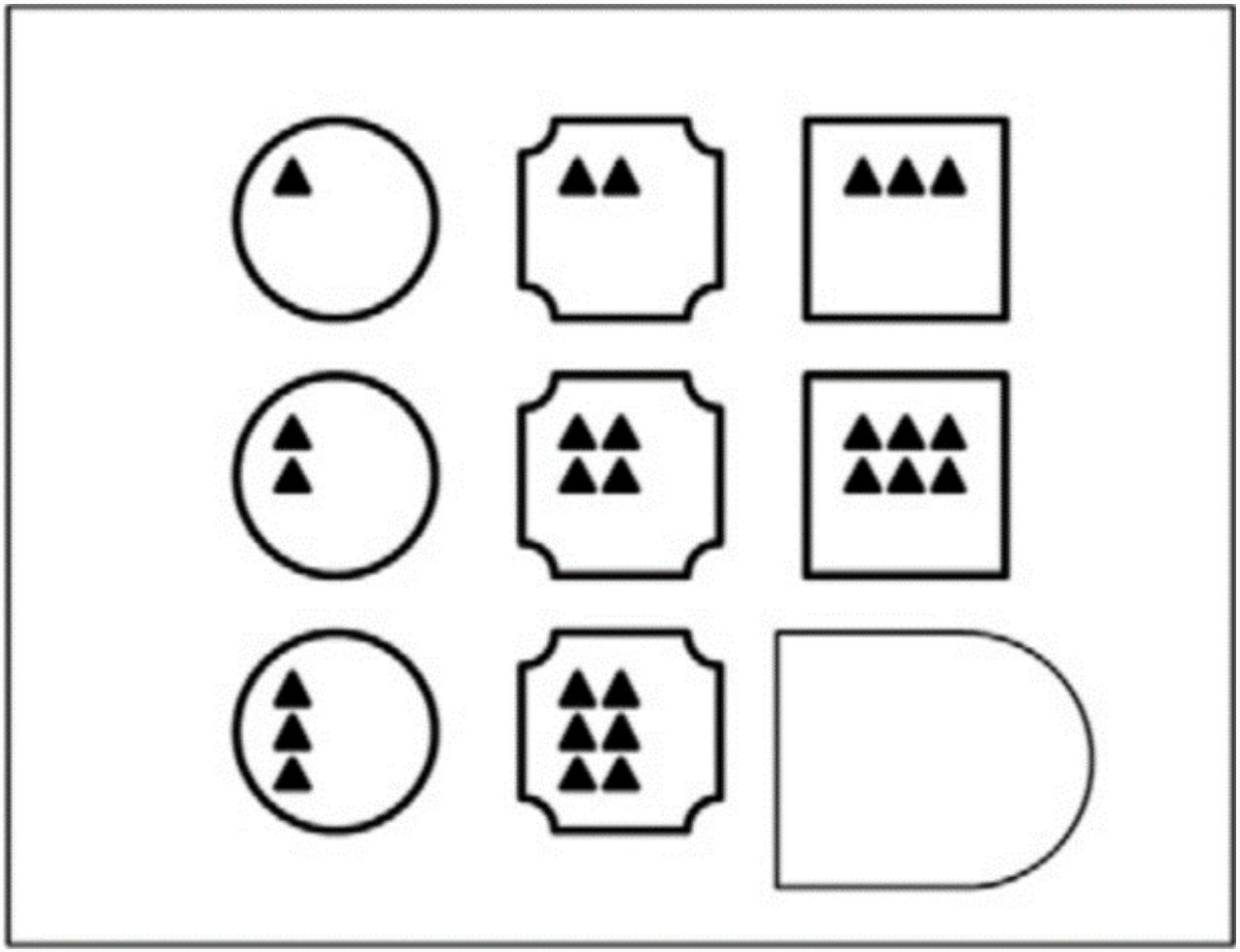
Un esempio



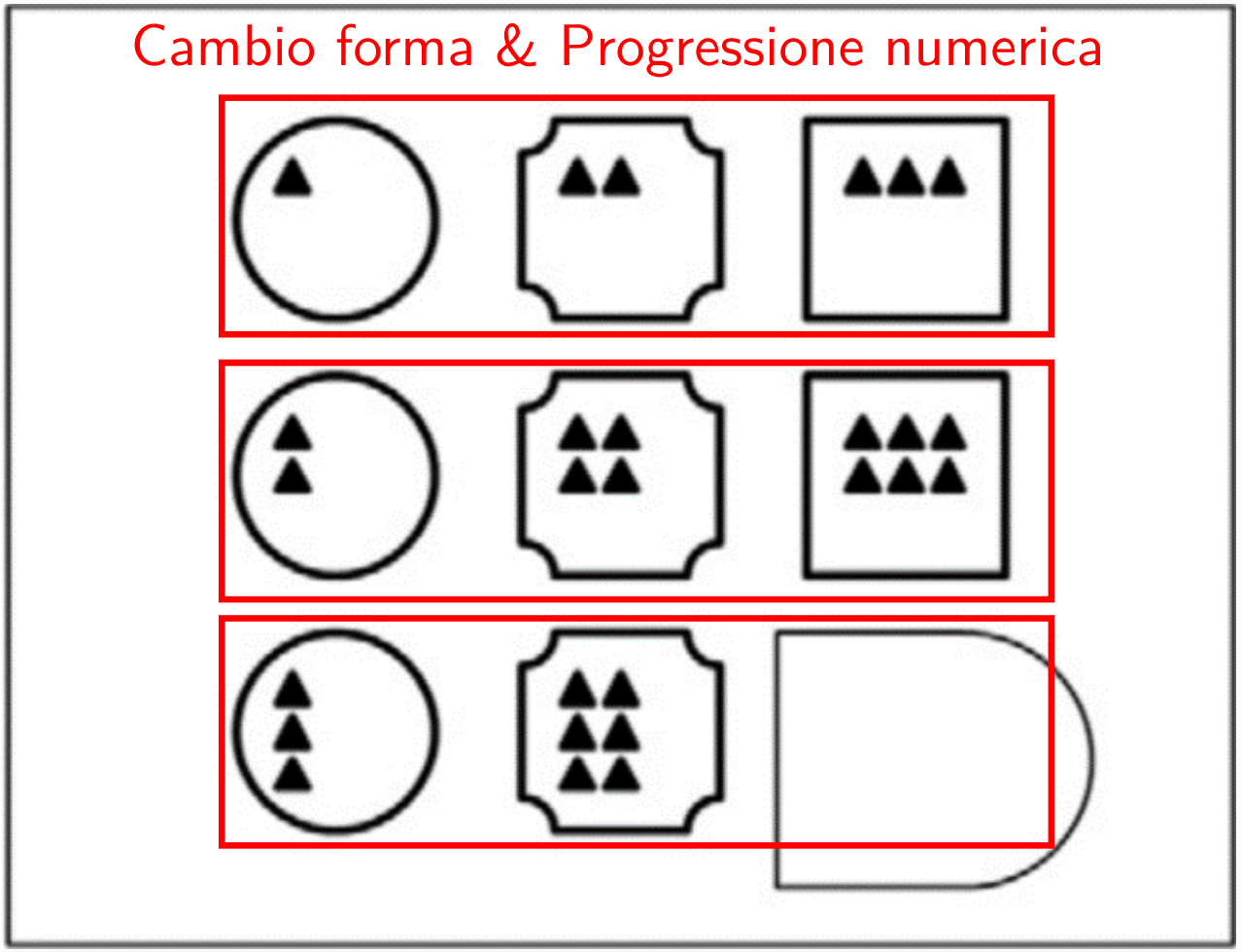
Un esempio



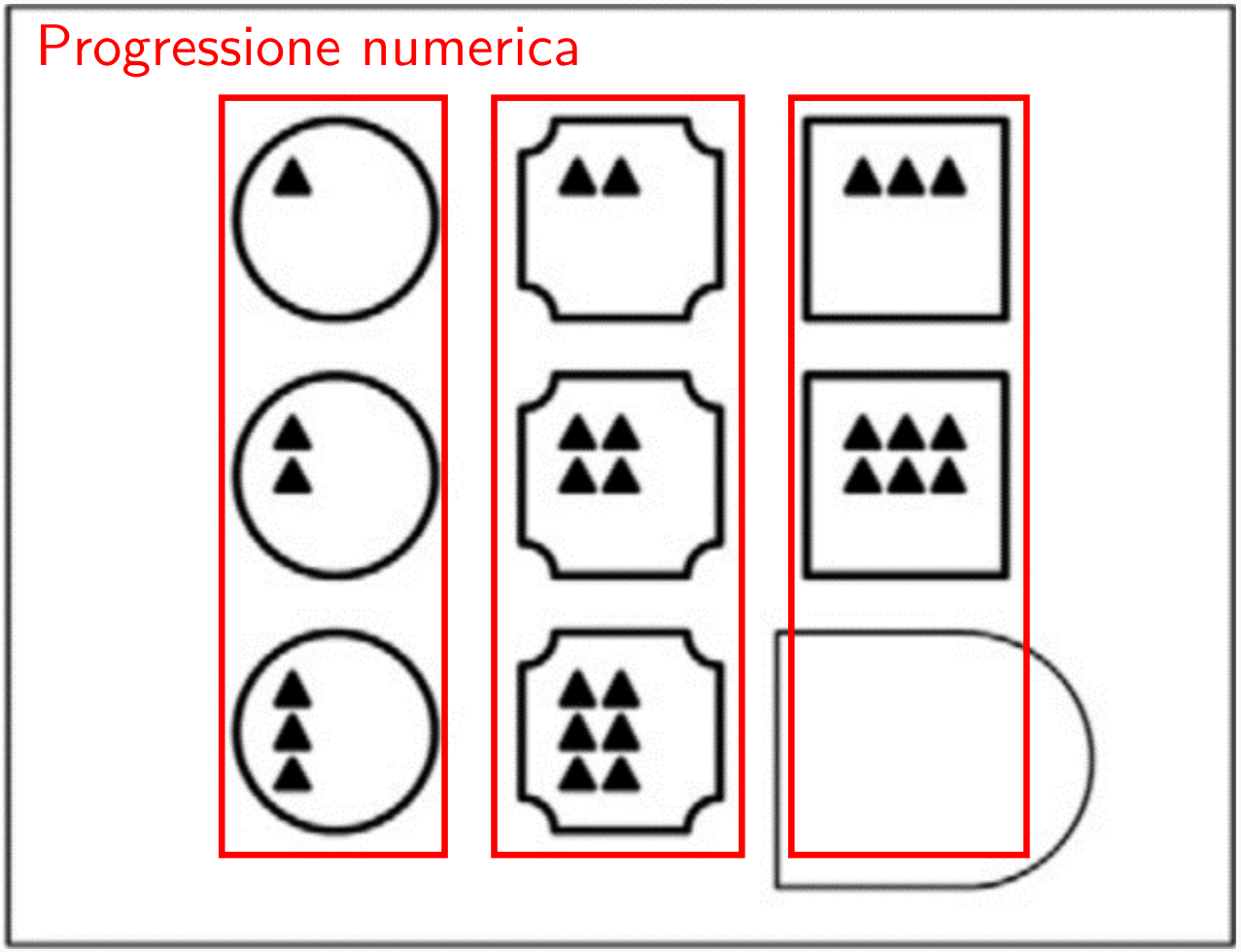
The matrix



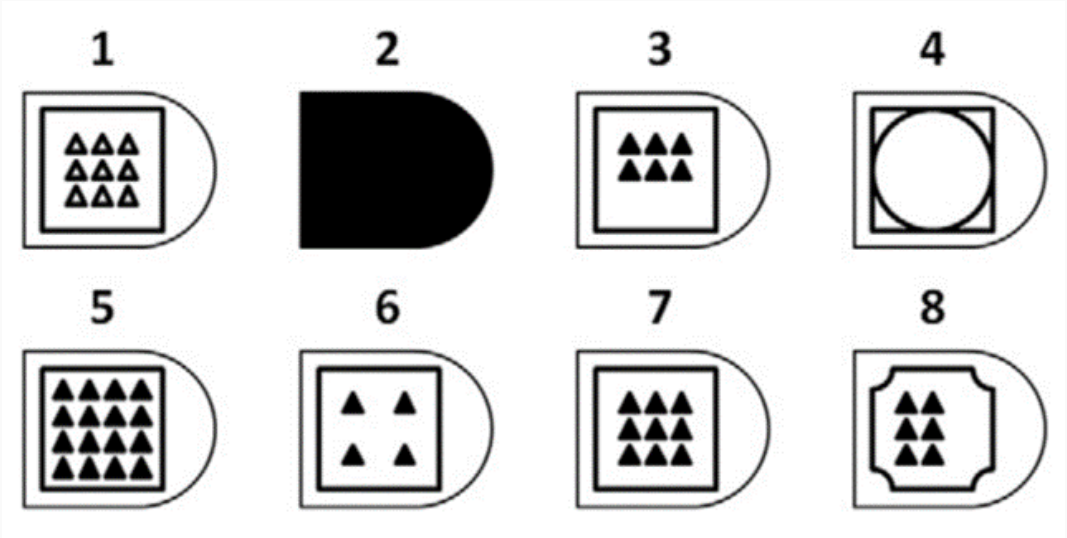
The matrix

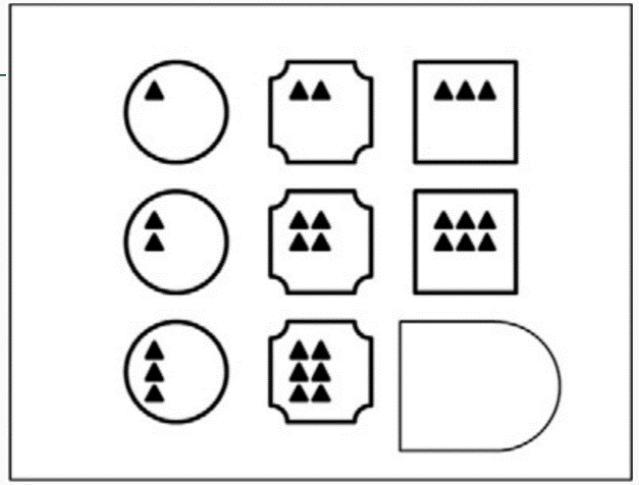


The matrix

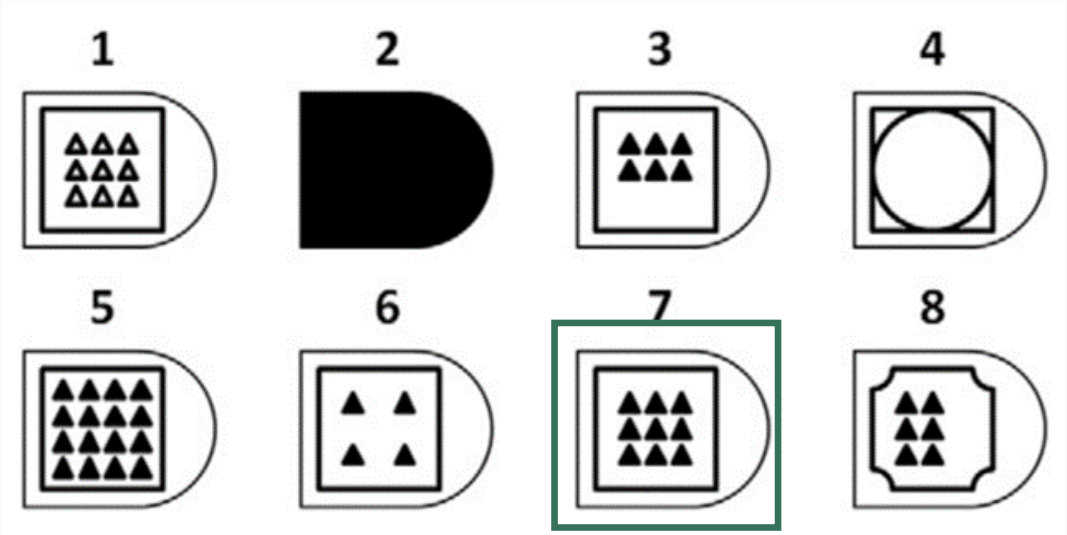


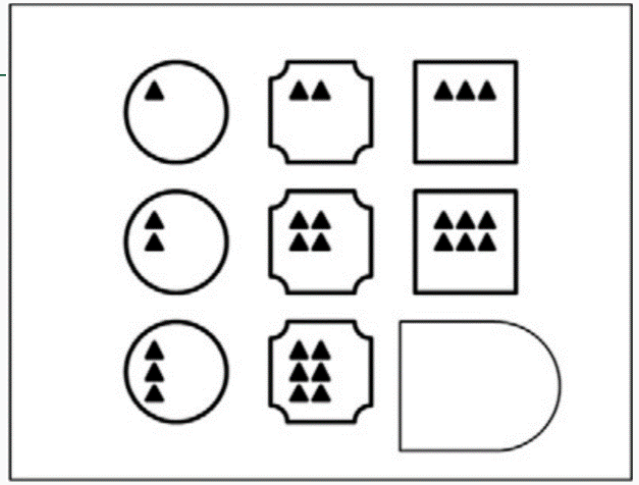
The response options



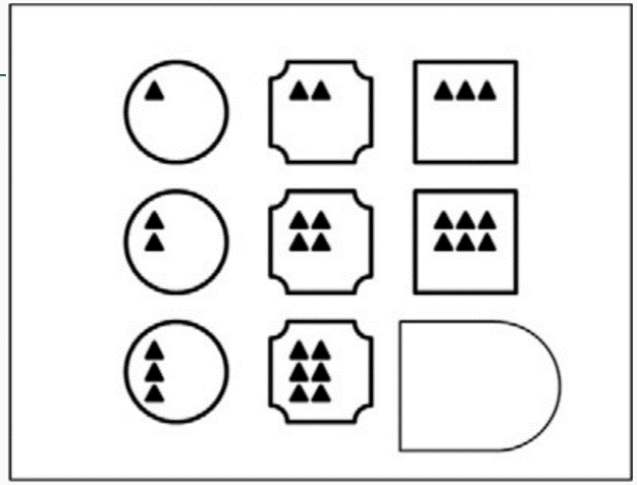


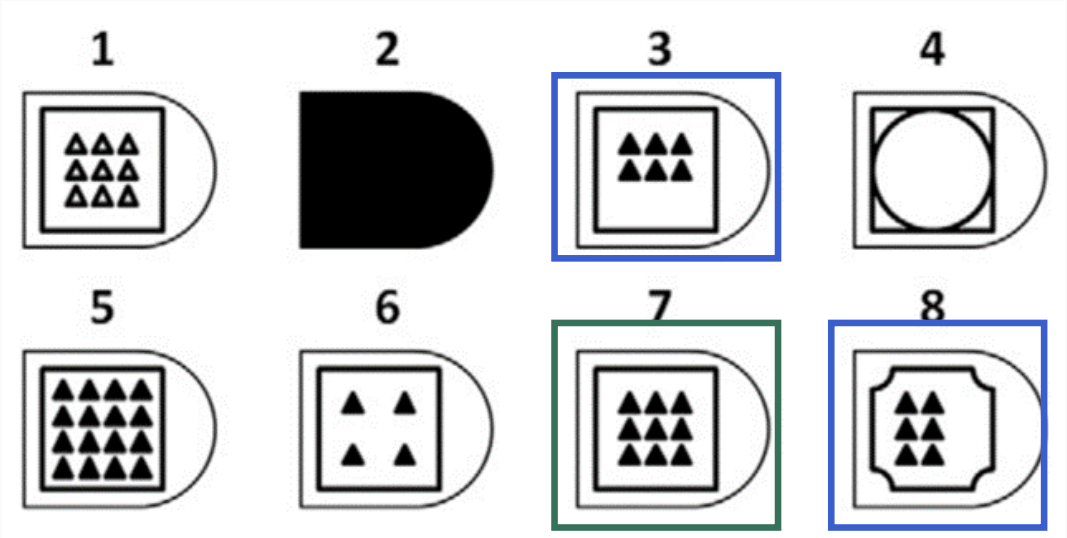
The response options





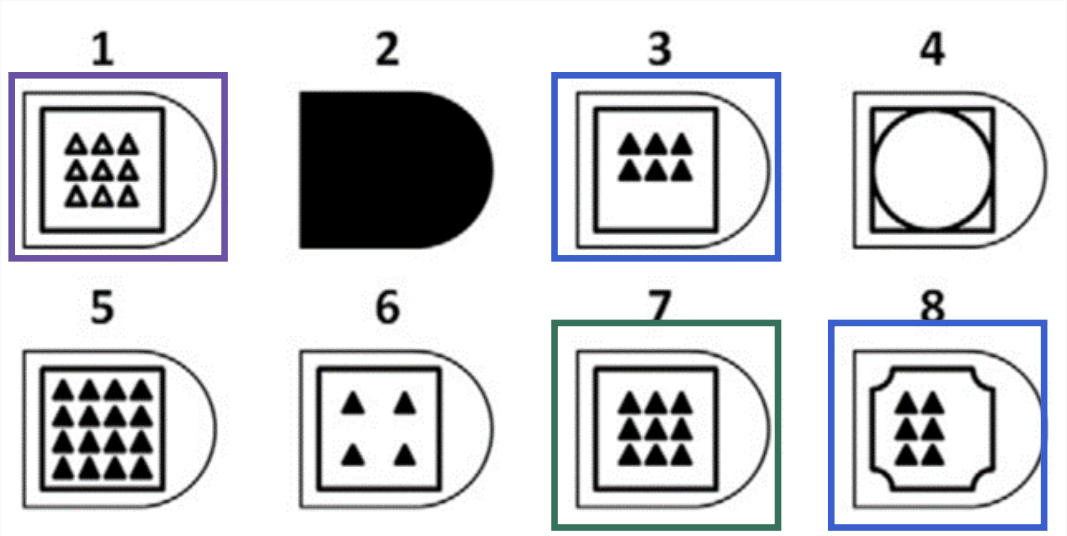
The response options

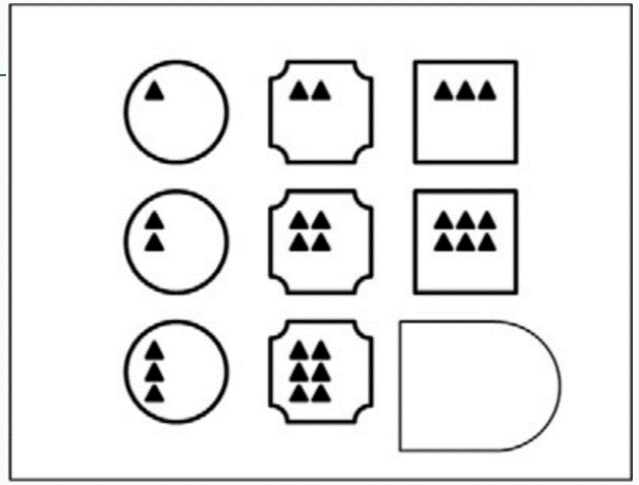




Repetition	Ripetizione di una cella adiacente alla cella vuota
Incomplete Correlate	'Quasi" la risposta corretta
Wrong Principle	Viene usata una regola non corretta per risolvere la matrice
Difference	Effetto pop-up

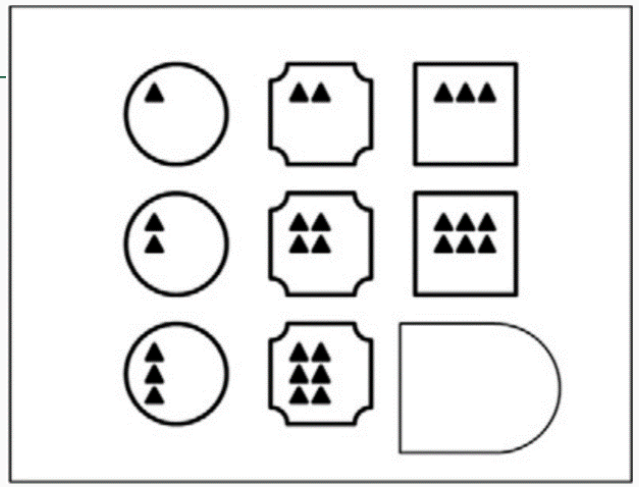
The response options

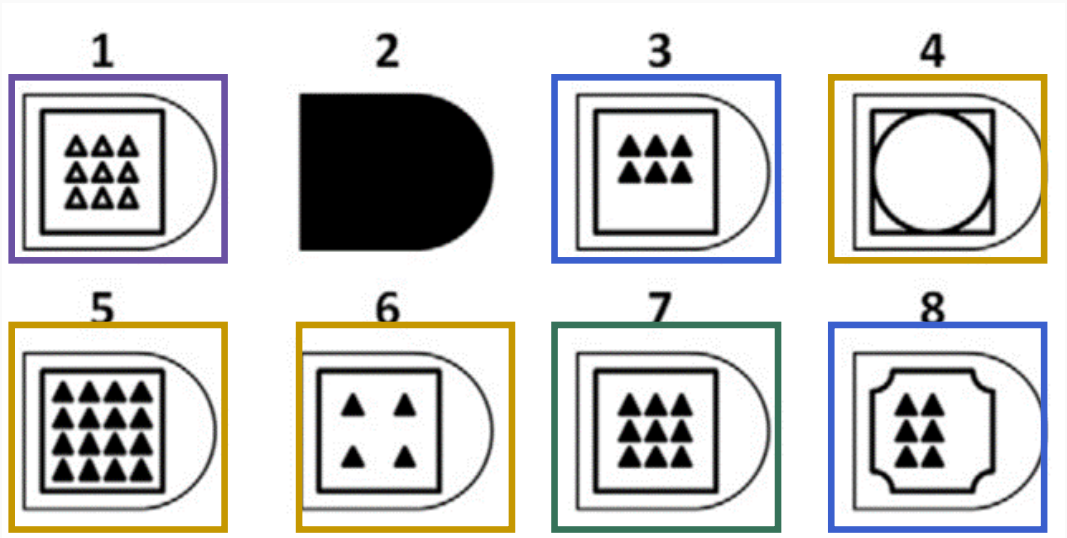




Repetition	Ripetizione di una cella adiacente alla cella vuota
Incomplete Correlate	'Quasi" la risposta corretta
Wrong Principle	Viene usata una regola non corretta per risolvere la matrice
Difference	Effetto pop-up

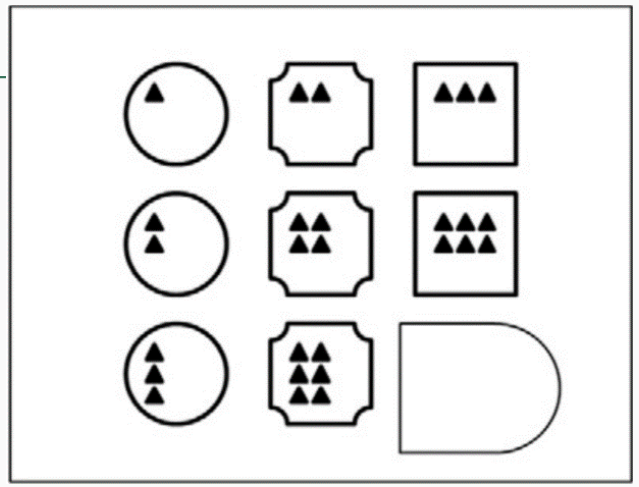
The response options

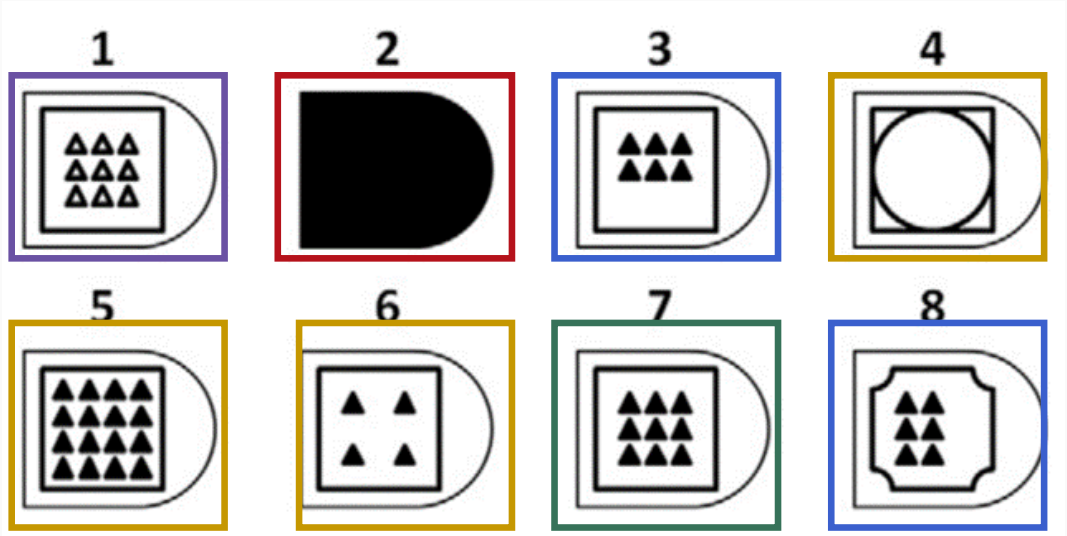




Repetition	Ripetizione di una cella adiacente alla cella vuota
Incomplete Correlate	'Quasi" la risposta corretta
Wrong Principle	Viene usata una regola non corretta per risolvere la matrice
Difference	Effetto pop-up

The response options





Repetition	Ripetizione di una cella adiacente alla cella vuota
Incomplete Correlate	'Quasi" la risposta corretta
Wrong Principle	Viene usata una regola non corretta per risolvere la matrice
Difference	Effetto pop-up

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

Corvus

It's on https://github.com/Thimbleby/Corvus (and the maintainer is super nice :)!
Based in Javascript but provided with with an intuitive UI

Sandia

No longer maintained
"Rudimentary stimuli"

They don't allow for reproducing the stimuli generation process

matRiks

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

matRiks: Generates Raven-Like Matrices According to Rules

Generates Raven like matrices according to different rules and the response list associated to the matrix. The package can generate matrices composed of 4 or 9 cells, along with a response list of 11 elements (the correct response + 10 incorrect responses). The matrices can be generated according to both logical rules (i.e., the relationships between the elements in the matrix are manipulated to create the matrix) and visual-spatial rules (i.e., the visual or spatial characteristics of the elements are manipulated to generate the matrix). The graphical elements of this package are based on the 'DescTools' package. This package has been developed within the PRIN2020 Project (Prot. 2020SWKCLL) titled "Computerized, Adaptive and Personalized Assessment of Executive Functions and Fluid Intelligence" and founded by the Italian Ministry of Education and Research.

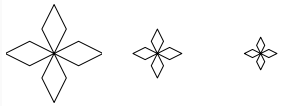
Version:	0.1.3
Imports:	DescTools
Suggests:	devtools, knitr, rmarkdown, testthat (≥ 3.0.0), VR
Published:	2024-02-16
DOI:	10.32614/CRAN.package.matRiks
Author:	Andrea Brancaccio [aut, cre, cph, cre], Ottavia M. Epifania [aut, ctb, cosu], Debora de Chiusole [ctb]
Maintainer:	Andrea Brancaccio <andrea.brancaccio at unipd.it>
License:	MIT + file LICENSE
NeedsCompilation:	no
Materials:	README NEWS
CRAN checks:	matRiks results

Documentation:

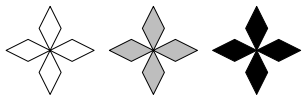
Reference manual:	matRiks.pdf
Vignettes:	black-figures circle-sections closed-figures eight-shapes-figures flowers-figures generate_matriks lines other-figures

Regole disponibili

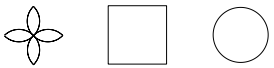
Cambi di dimensione, `size`



Cambi di riempimento, `shade`



Cambi di forma, `shape`



Cambi di orientamento, `rotate`

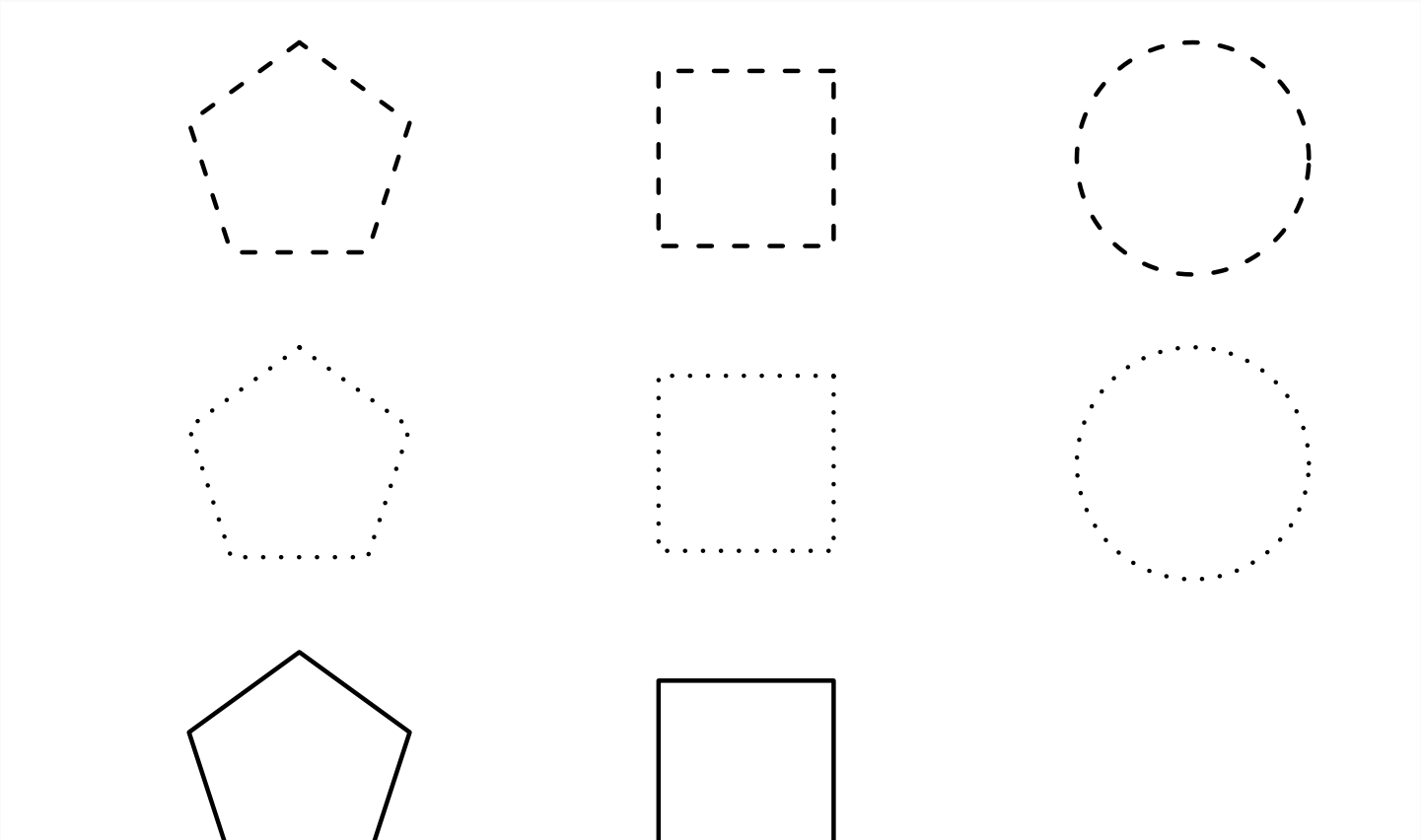
AND ($\cap$), `AND`

OR ($\cup$), `OR`

XOR (Δ), `XOR`

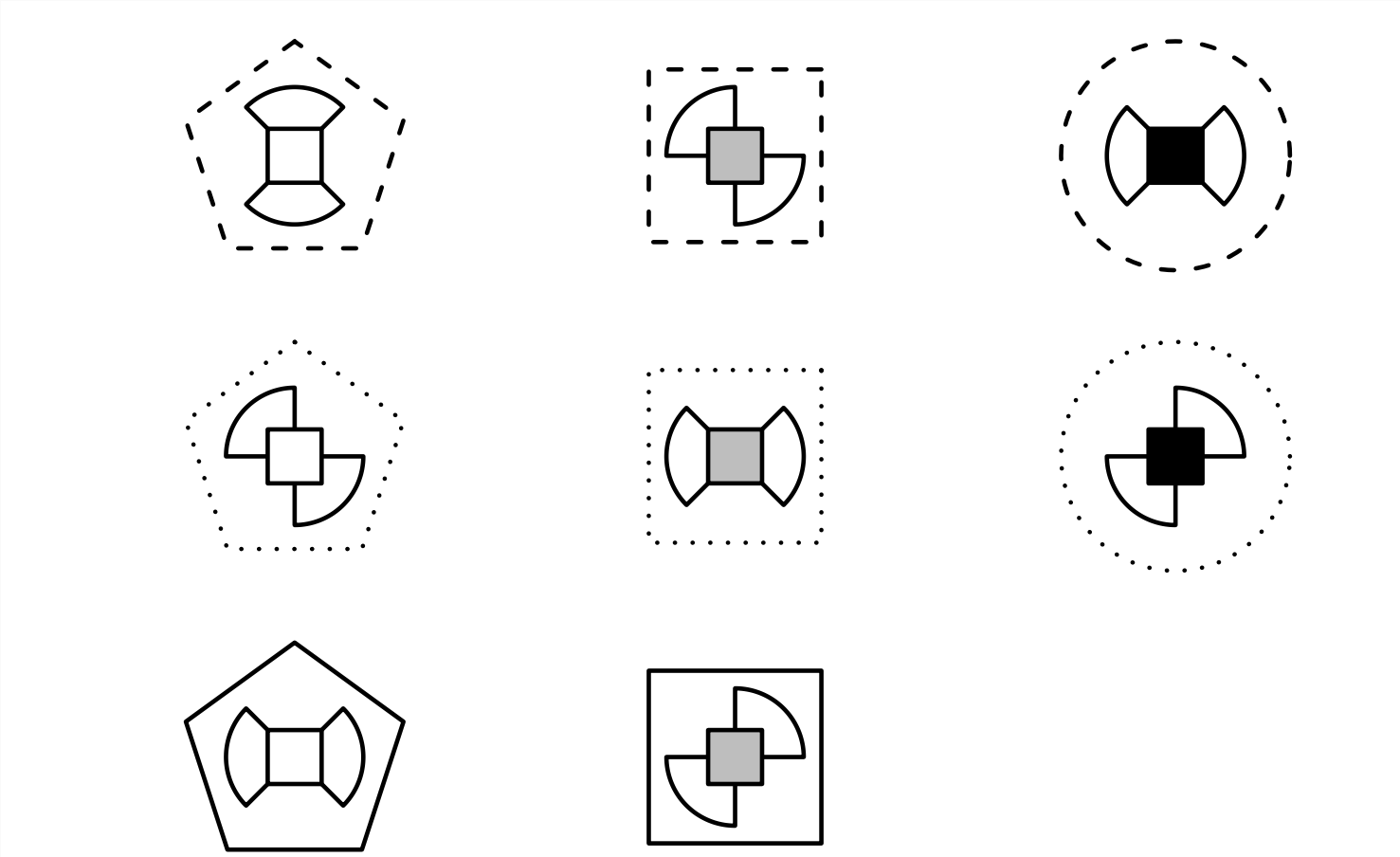
Generare gli stimoli

```
# genera una matrice
my_mat1 = mat_apply(cof(pentagon(),square(size.x = 16, size.y = 16),
                        circle(size.x = 15)),
          hrules = "shape", # regola applicata attraverso le colonne
          vrules = "lty") # regola applicata attraverso le righe
# disegna la matrice
draw(my_mat1, hide = TRUE)
```



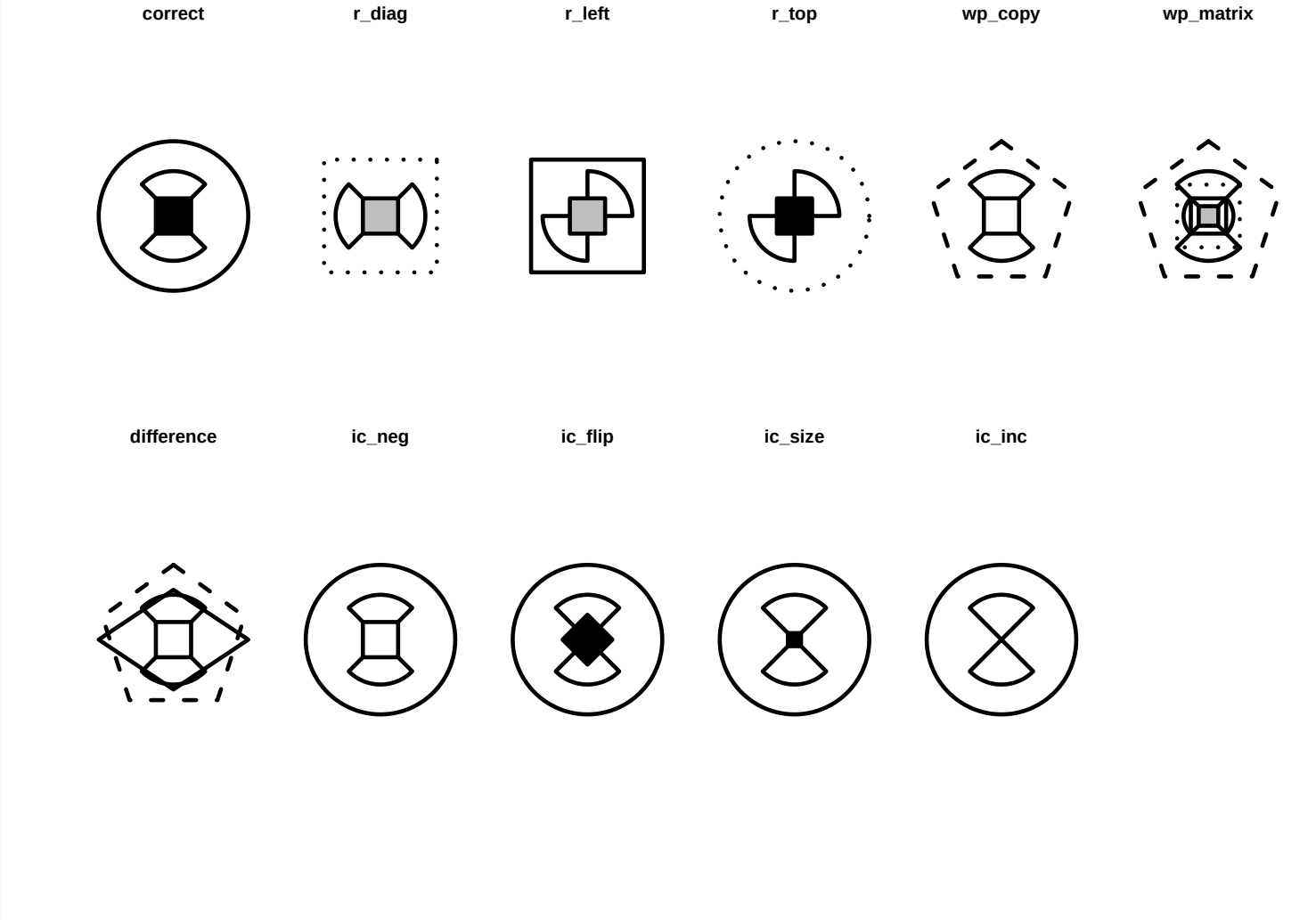
Generare stimoli più complessi

```
my_mat2 = mat_apply(axe(size.x=9), hrules ="rotate", vrules = "rotate")
my_mat3 = mat_apply(square(size.x = 5),  hrules = "shade")
the_mat = com(my_mat1, my_mat2, my_mat3) # combina le matrici
draw(the_mat, hide = TRUE)
```



A ogni stimolo i suoi distrattori

```
my_responses = response_list(the_mat)
draw(my_responses, main = T)
```



Estremante facile da usare, rende la generazione degli stimoli accessibile a tutt*

Costringe a pensare a livello teorico agli stimoli che si vogliono generare, alla loro complessità, alle loro caratteristiche

Il codice rimane e si può inserire all'interno di un RMarkdown per avere stimoli e distrattori sempre insieme