FIGUMEM – USER'S MANUAL

Correspondence: David Jendryczko (david.jendryczko@uni-konstanz.de)

Contents

Preface

This manual describes how to use the item generator *figumem* for the creation of figural memory items for cognitive ability assessments. It is strongly recommended to read the article *Introducing a computerized figural memory test based on automatic item generation: An analysis with the Rasch Poisson counts model* (Jendryczko, Berkemeyer & Holling, 2020; available under: https://www.frontiersin.org/articles/10.3389/fpsyg.2020.00945/full). first, as it explains figumem and delivers the theoretical backgrounds on automatic item generation and the construct of figural memory[1].

The use of figumem requires knowledge about the very basics of the functionality of R. Literature on this is given, for example, by Teetor (2011). We also recommend the study of the magick software package (Ooms, 2018) for further insight into the possibilities of image-manipulation in R. For that purpose visit https://cran.r-project.org/web/packages/magick/vignettes/intro.html.

Explanation of the figumem_item-function

figumem_item creates figumem items. The function contains the following arguments:

```
figumem_item(objn = 20, mode = "classic", challenge = 1, seed = sample(1:999999999,1), ntrys = 100)
```

objn determines how many emblem-frame-associations shall be used. The default is 20 but it can be any natural number between one and twenty.

mode determines the item format. Currently, four formats are available. The default is

[1] Please note that figumem was provided as an R-script in the supplementary material of the article by Jendryczko et al. (2020). The version of the manual you are reading refers to the R-package version of figumem. If you have the R-package version you do not need the original version of figumem from the article supplementary material.

"classic".

"classic" creates figumem item material for an easy digital implementation of a figural memory item into common online survey tools. One image is produced that contains the memorizing display, i.e. the emblems in their assigned frames arranged in a matrix. One image per emblem is produced that contains a large representation of the emblem. One image per emblem is produced that contains the target frame of the emblem that surrounds it in the memorizing display. Three images per emblem are produced that contain distractor frames for the emblem.

"classic_paper" reduces the amount of produced images. One image is produced that contains the memorizing display. One image is produced that contains a matrix in which the emblems are presented in a different order with four frames (one target, three distractors) arranged underneath each emblem.

"draw" creates figumem item material for an open response format that requires testees to draw the frames. One image is produced that contains the memorizing display. One image is produced that contains the emblems in the same order as before but without the frames.

"draw_random" is the same as "draw" but the order of the emblems for the second image (without the frames) is randomized.

challenge can be seen as the item generator radical and determines the item difficulty, which in turn is given by magnitude of visual load. 1 produces items of the easiest difficulty level (low visual load), 2 produces items of a medium difficulty level (medium visual load) and 3 produces items of the hardest difficulty level (high visual load). The default is 1.

seed sets the seed for the item generator. The assignment of frames to emblems, the order of the figural associations and the selection of distractors are determined randomly (these properties are incidentals). By specifying a seed number an item can be exactly reproduced in spite of that. The default is to use a random seed number.

`ntrys` determines how many tries the item generator shall take before a figumem item is successfully produced or before the generation process is cancelled. This is only relevant for the modes `"classic"` and `"classic_paper"` because the algorithm that randomly assigns distractors to emblems while simultaneously ensuring that every frame is used equally often does not always converge. The default is 100. If convergence does not occur within the specified number of tries, a warning message will be printed in the console that explains that a certain distractor could not be decided for and that a new attempt is required. It is recommended to always produce items using a fixed seed number. If the algorithm does not converge, the seed number can be altered and documented and a new attempt can be made.

Example items

To create an example figumem item in `"classic"`-format use the following code:

```
eg1 <- figumem_item(objn = 20, mode = "classic", challenge = 1, seed = 1, ntrys = 100)
```

Since in this case all function arguments except for `seed` are set to their defaults, the code can be written in an abbreviated form and will produce the same item:

```
eg1 <- figumem_item(seed = 1)
```

It creates the object `eg1` that contains all the produced images. The images can be assessed. In order to examine the memorizing display use:

```
eg1[[1]]
```

In order to examine the first emblem use:

```
eg1[[2]][[1]][1]
```

In order to examine the second emblem use:

```
eg1[[2]][[2]][1]
```

In order to examine any of the third to the twentieth emblem replace the 2 in the middle set of brackets of the last code-line by the respective number. In order to examine the target frame, the first distractor, the second distractor and the third distractor of the first emblem in that order run the following lines successively:

```
eg1[[2]][[1]][2]
eg1[[2]][[1]][3]
eg1[[2]][[1]][4]
eg1[[2]][[1]][5]
```

The target frame and the distractors for the second to the twentieth emblem can be assessed by replacing the 1 in the middle sets of brackets of the last four code-lines with the respective number.

To create the same item in `"classic_paper"`-format use the following code:

```
eg2 <- figumem_item(mode = "classic_paper", seed = 1)
```

It creates the object `eg2` that contains all the produced images. In order to examine the memorizing display use:

```
eg2[[1]]
```

In order to examine the answering display, in which the testee has to mark the correct options, use:

```
eg2[[2]]
```

To create the same example figumem item in `"draw"`-format use the following code:

```
eg3 <- figumem_item(mode = "draw", seed = 1)
```

It creates the object `eg3` that contains all the produced images. In order to examine the memorizing display use:

```
eg3[[1]]
```

In order to examine the answering display, in which the testee has to draw the frames around the emblems, use:

```
eg3[[2]]
```

To create the same item in `"draw_random"`-format use the following code:

```
eg4 <- figumem_item(mode = "draw_random", seed = 1)
```

It creates the object `eg4` that contains all the produced images. In order to examine the memorizing display use:

```
eg4[[1]]
```

In order to examine the answering display, in which the testee has to draw the frames around the emblems (that are now presented in a different order), use:

```
eg4[[2]]
```

Explanation of the `figumem_write`-function

`figumem_write` saves figumem items as PNG, JPG, or PDF files. The function contains the following arguments:

```
figumem_write(item, name = "figural_memory_item", format = "png")
```

`item` specifies which item shall be saved. This argument has no default.

`name` specifies the general name for the PNG data files. The default is "figural_memory_item". The files will be saved in the current working directory. The working directory can be changed before the function is used (see "Preparations"). Alternatively, you can specify a complete path in order to save the files somewhere different than your current working directory.

`format` specifies which format shall be used. figumem items can be saved in PNG-, JPG-, and PDF-format. Use `"png"`, `"jpg"`, and `"pdf"`, respectively. The default is `"png"`. `"pdf"` is only recommended for items of modes `"classic"`, `"classic_paper"`, `"draw"`, and `"draw_random"` as items of these modes are usually used for paper-pencil versions of the test and PDF allows efficient printing.

Examples

In order to save the figumem item `eg1` in PNG-format use:

```
figumem_write(item = eg1, name = "figural_memory_item", format = "png")
```

Since in this case all function arguments are set to their defaults (except `item` which has no default), the code can be written in an abbreviated form and will work the same way:

```
figumem_write(item = eg1)
```

The function recognizes that eg1 is a figumem item in "classic"-mode. It will create several PNG data files in the current working directory. All of the file names will start with "figural_memory_item". One file will be named "figural_memory_item_display.png". It will contain the memorizing display, i.e. the image retrieved from eg1[[1]].

Twenty files will be named "figural_memory_item_stimulus_x_emblem.png", whereas "x" is to be replaced with a number from 1 to 20, respectively. They will contain the emblems. Twenty files will be named "figural_memory_item_stimulus_x_target.png", whereas "x" is to be replaced with a number from 1 to 20, respectively. They will contain the target frames for the respective emblems.

Sixty files will be named "figural_memory_item_stimulus_x_distractor_w.png", whereas "x" is to be replaced with a number from 1 to 20, respectively and "w" is to be replaced with a number from 1 to 3 for every given "x", respectively. They will contain the distractors for the respective emblem given by "x".

In order to save the figumem item eg2 in PDF-format use:

```
figumem_write(item = eg2, name = "figumem_paper_item", format = "pdf")
```

By altering the expression for the name argument of the function one can assure not to overwrite previously saved items. The function recognizes that eg2 is a figumem item in "classic_paper"-mode. It will create two PDF data files in the current working directory. Both file names will start with "figumem_paper_item". One file will be named "figumem_ paper_item_display.pdf". It will contain the memorizing display, i.e. the image retrieved from eg2[[1]]. One file will be named "figumem_ paper_item_choices.pdf". It will contain the display in which the testee has to mark the correct options, i.e. the image retrieved from eg2[[2]].

The function will also recognize items of the remaining modes (`"draw"` and `"draw_random"`). In these cases, two files – one file with the postfix "_display" and one file with the postfix "_choices" – will be produced, as well. The file with the postfix "_choices.png" will contain the image in which only the emblems and not the frames are presented.

References

Jendryczko, D., Berkemeyer, L., & Holling, H. (2020). Introducing a Computerized Figural Memory Test Based on Automatic Item Generation: An Analysis With the Rasch Poisson Counts Model. *Frontiers in Psychology, 11*, 945. doi:10.3389/fpsyg.2020.00945

Ooms, J. (2018). *magick: Advanced Graphics and Image-Processing in R.* R package version 1.9. https://CRAN.R-project.org/package=magick

Teetor, P. (2011). *R Cookbook*. Sebastopol, CA: O’Reilly Media, Inc.