Give me Gestalt! - XPLab 2019

Final report

Group 04: Cora Burgwinkel, Lina Junctorius, Mirijam Schaaf, Annika Theisen

We aimed to replicate the “Give me Gestalt! Preference for Cubist artworks revealing high detectability of objects” experiment by Muth et al. (2012).

Background

Cubism is a highly influential visual art style from the 20th century. The three most famous and most influential painters for this particular style are Picasso, Braque and Gris.
In Cubist artworks, objects are analysed, fragmented and reassembled into an abstracted form. Therefore, Cubist paintings are very open to interpretation.
They are full of everyday objects, but because of the fragmentation, immediate object recognition is difficult.
With this study we aimed to show that the subjects’ enjoyability of the pictures is linked to the ability of detecting objects, also called Gestalt, in the paintings.


A leading theory behind rating enjoyability and detecting objects is to derive pleasure from searching for and finding recognisable everyday objects. For example, neurologists Ramachandran and Hirstein argue that perceptual grouping processes are in general linked with the neural structures known as the reward system.[1] Reber et al. assert that increased fluency in processing a complex topic enhances appreciation.[2]

[1] V.S. Ramachandran and W. Hirstein, “The science of art. A neurological theory of aesthetic experience,” *Journal of Consciousness Studies* 6, No.6-7 (1999) pp.15-51
[2] R. Reber, N. Schwarz, and P. Winkielman, “Processing fluency and aesthetic pleasure. Is beauty in the perceiver's processing experience?,” Personality and Social Psychology Review 8, No. 4 (2004) pp. 364-382.

Research question

The more easily individuals without expertise in cubism can detect objects in the presented paintings from Braque, Gris and Picasso, the higher the likeability is rated.

Design of the Experiment

General remarks about the Design

We conducted a within subject experiment. The independent variable is the detectability of objects in the painting while the dependent variable is the likeability of the paintings.
In contrast to the original experiment, our replication of the experiment was web-based. We used the architecture babe which is based on JavaScript and HTML while the statistical analysis was performed with R.

Sampling Plan

Our plan was to recruit at least 20 participants and we were able to recruit 70 participants in total. We had a time limit of one week. The participants were recruited among Cognitive Science students at the University Osnabrueck which are enrolled in the course “Experimental Psychology Lab” and among friends and family. The method we used is called convenience sampling as a type of non-probability sampling. That means we collected a sample from somewhere convenient to us and that the odds of any member being selected for a sample cannot be calculated. The major advantage of the non-probability sampling technique is that it is very time- and cost-effective. The main disadvantage of this technique is that it is impossible to know how well one is representing the original population, so there may be a shortcoming of representation.[3]
Participants were recruited via email with a link that provided access to the online experiment. The participants were no experts in cubist art and have corrected-to-normal vision as well as normal colour vision.

Procedure

In the original experiment 120 photographs of paintings by Pablo Picasso, George Braque and Juan Gris were used. The stimuli were limited to 30 images to shorten the duration of the experiment because participants did not get paid and thus, were likely not willing to invest a lot of time in the experiment. As we are in possession of the original stimuli, the stimuli set respectively contains 5 monochrome and 5 coloured paintings of each painter.
Before the experiment begins, we wanted to make sure that participants meet the required distance to their laptop. The distance should be 55cm at all times. The participants were easily able to measure this by putting their hand on their monitor and staying an arm length away. We introduced this method to the participants in order to assist them with maintaining the required distance as good as possible. We chose to do so because we are not able to correct their position as this experiment is not conducted in a laboratory setting.

---

[3] Non-Probability Sampling: Definition, Types Online URL: https://www.statisticshowto.datasciencecentral.com/non-probability-sampling/[last visited 27.06.2019]

The experiment consisted of two blocks and in both blocks 30 paintings were shown in a randomised order. Participants were shown written instructions about the task. In the first block the participants were asked to rate the paintings on how much they liked them. The participants rated on a 7 point Likert scale from 1 (“not at all”) to 7 (“very much”). During the second block, participants were asked to rate how well they were able to detect objects in the paintings. The participants rated on a 7 point Likert scale from 1 (“very hard”) to 7 (“very easy”) by clicking on one of the seven numbers on the scale. After the two blocks, the corrected-to-normal vision of the participants was tested by a self-constructed vision test and by a short version of the Ishihara colour vision test. There was no time limit because we wanted the subjects to have enough time to get involved with the paintings and the reaction times also did not matter for the statistical analysis. Finally, every participant was asked to indicate how much he is an expert in cubism by again choosing a number on a 7 point Likert scale from 1 (“not at all”) to 7 (“absolutely”).

Here is one example for each of the trials:

How much do you like the painting?

Not at all (1) (2) (3) (4) (5) (6) (7) Very much

How well can you detect objects in the painting?

Very hard (1) (2) (3) (4) (5) (6) (7) Very easy

Measured variables

- likeability, the dependent variable: measured by an ordinal 7-point Likert-scale from 1 (“not at all”) to 7 (“very much”)
- detectability, the independent variable: measured by an ordinal 7-point Likert-scale from 1 (“very hard”) to 7 (“very easy”)

Materials

We used the original photographs provided by Claudia Muth; the author of the study we are replicating. The pictures were part of the stimuli set used in Muth et al (2012). The photographs have been adapted to 450 pixels width and 600 pixels height. If the proportion was not 4.5:6, the pictures were cropped accordingly. A list of the stimuli can be found below.

- Muth, C., Pepperell, R., Carbon, C. (2012), “Give me Gestalt! Preference for Cubist artworks revealing high detectability of objects” Leonardo, 46

Besides the stimuli we used further materials which are referenced here:

The image of the hand is provided by Clipart.

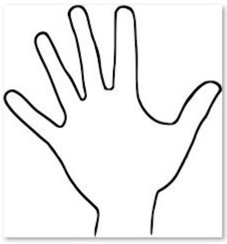
- *Hand Clipart Black and White 2* [picture]. Retrieved from https://www.clipart.email/download/177177.html [last visited 27.06.2019]

The image for the Ishihara Test is provided by specialist and eye surgeon Marek.

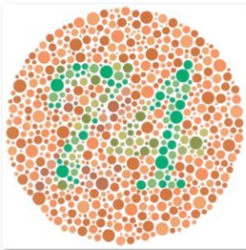
- Eyespecialist and eye surgeon Marek. *Ishihara Test* [image]. Retrieved from https://augenmedizin.at/wp-content/uploads/2014/11/Ischihara-5-300x300.jpg [last visited 27.06.2019]

We created the images for the vision test ourselves. Therefor we created two images with the same width and height as the used stimuli.

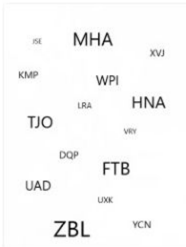
- Junctorius, L., Schaaf, M., Burgwinkel, C., Theisen, A. (2019). *Visiontest and Visiontest 2* [pictures].



Hand.png



ishihara.png



Visiontest.jpg

Visiontest_2.jpg

Stimuli

Total: 30 Stimuli

		Colour Scheme	
		Coloured	Monochrome
Painter	Pablo Picasso	5x	5x
	George Braque	5x	5x
	Juan Gris	5x	5x

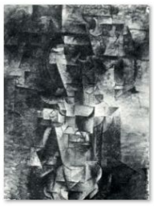
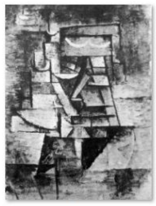
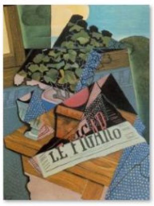
List of used Paintings

No	Painter	Painting	Year
1	Braque	The old castle at La Roche-Guyon	1909
2	Braque	Still life with mandola, metronome (vase and books)	1909
3	Braque	Woman holding a mandolin	1910
4	Braque	The Rio Tinto factories at L'Estaque	1910
5	Braque	Still life with a bottle	1910
6	Picasso	Still life with liqueur bottle	1909
7	Picasso	The rack (with glass, pipe and letter)	1912
8	Picasso	Bowl of fruit	1910
9	Picasso	Standing nude woman	1910
10	Picasso	Man rowing (with oars)	1910
11	Braque	Factory roofs at L'Estaque	1908
12	Braque	The chateau at La Roche-Guyon	1909
13	Braque	The mandolin (and bottle)	1910
14	Braque	Bottle and glass (on a table)	1911
15	Braque	Still life with harp and violin (glass and ink blotter)	1912
16	Picasso	Man with guitar	1913
17	Picasso	Glass with straws	1911
18	Picasso	Man with clarinet	1911
19	Picasso	Woman with triangular head	1910
20	Picasso	The female student	1911
21	Gris	The pot of geraniums	1915
22	Gris	Still life on a chair	1914
23	Gris	Still Life with Fruit Dish and Mandolin	1919
24	Gris	Water bottle, bottle and fruit dish	1915
25	Gris	The checked tablecloth	1915

26	Gris	Still life with playing cards	1916
27	Gris	The violin	1916
28	Gris	The checkerboard	1915
29	Gris	Still life with poem	1915
30	Gris	The lamp	1916


16.jpg

17.jpg

18.jpg

19.jpg

20.jpg


21.jpg

22.jpg

23.jpg

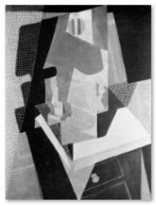
24.jpg

25.jpg

26.jpg

27.jpg

28.jpg

29.jpg

30.jpg

1.jpg

2.jpg

3.jpg

4.jpg

5.jpg

6.jpg

7.jpg

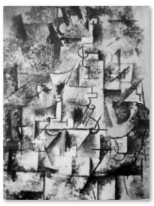
8.jpg

9.jpg

10.jpg

11.jpg

12.jpg

13.jpg

14.jpg

15.jpg

Analysis

Exclusion criteria

The first exclusion criterion was colour blindness because in order to recognise objects in the paintings, it is necessary to have unimpaired colour vision. To test the participants' colour vision, we made use of the Ishihara test. The Ishihara test is a colour perception test developed by Dr. Shinobu Ishihara at the University of Tokyo in 1917. It consists of multiple so called Ishihara plates. As our experiment needed to stay in a reasonable time frame for participants, we decided to only include plate 7. This plate mainly tests if subjects have red-green colour blindness which is one of the most common colour vision deficiencies. The number shown on plate 7 is 74. Viewers may see the numbers 71 or 21 instead. Viewers who see 21 are likely to have dichromacy (two types of cells functioning) or anomalous trichromacy (one of the cones is altered in its spectral sensitivity). Whilst viewers with monochromacy (one type of cone cells functioning) may see nothing. We have decided to exclude people whose answers differ from 74 or 71. We wanted to include participants that see 71 instead of 74 because given the image's composition and the colour contrast it is likely to see a 1 instead of a 4. If participants see a 71, their colour vision may be slightly impaired but we do not consider it as sufficiently severe as to affect the participants' responses.

After completing the Ishihara test, our participants needed to perform a vision test. Therefore, we created an image which has the same width and height as the stimuli and shows different triples of uppercase letters in different sizes distributed on a white background. The different triples contained no similar letter combinations so that they cannot be confounded. The smallest triple had approximately the same size as the smallest object contained in the paintings. The participants were first asked to write down the smallest triple they were able to read in a text field. With this test we wanted to examine the participants' vision under conditions similar to the trials to make sure that they were able to see all objects appearing in the stimuli. Next, the participants saw a similar image in which all triples except the smallest one are faded out so that the smallest triple stands out. The participants were asked to write this triple down in another text field again. With this we wanted to make sure that they did not miss the smallest triple in the first part of the vision test and again test if they were able to read the smallest triple.

If a participant entered a wrong letter combination in the first and second part of the vision test, his or her data was excluded from the analysis. If a participant entered a wrong letter combination in the first part but the correct combination in the second part, his or her data was not excluded. If a participant entered the correct letter combination in the first part but a wrong combination in the second part, his or her data was excluded depending on the deviation from the correct triple. In this case, if the deviation involved more than one of the three letters, the data was excluded. By doing so, we avoided excluding participants whose vision was good enough for the experiment but that made a typographical error in the second part of the vision test.

Another exclusion criterion was whether the participants were already familiar with or even experts regarding the cubist art style. We asked them to indicate on a Likert rating scale from 1 ("not at all") to 7 ("absolutely"), if they were an expert in Cubism. When having answered with a 6 or 7, the participant was excluded. The method of self-reporting used here was appropriate because there is no reason for the participants to not answer this question honestly. The participant knows best if he or she has knowledge about the topic.

Finally, we excluded participants who needed less than 5 minutes to complete the experiment because we think that you cannot work through it seriously in such a short time frame. Additionally we excluded participants who rated 50 trials and more with the same number because this leads us to believe that the participant just clicked through the experiment without engaging with the paintings.

However, we did not exclude participants who did not use the most extreme values (1 and 7) because often one avoids the extremes and might not feel that strongly about the paintings.

Participant exclusion:

70 subjects participated in our experiment. According to our specified exclusion criteria, the following participants had to be excluded: 387, 374, 338 and 330 because of rating themselves as an expert in cubism (rated as 6); 376, 368, 362 and 346 because they failed the Ishihara colour vision test (seeing an 11, 21 or 24); 333 because of failing the general vision test and 380, 364, 357, 350, 345, 343, 342, 340, 354 and 355 because they completed the experiment in under 5 minutes of time. 1 data block (326) was from our own test run. The analysis was therefore executed with 50 subjects. 35 of these are female and 15 are male. The average age for the female participants is 24.49 and the average age for the male participants is 25.26. The overall average age is 24.7.

Confirmatory hypothesis testing

For our main analysis, we tested the hypothesis "The more easily individuals without expertise in cubism can detect objects in the presented paintings from Braque, Gris and Picasso, the higher the likeability is rated." by using Bayesian regression models. We analyzed three models treating the object-ratings as ordinal data and three models treating the object-ratings as interval/ metric data. The basic goal of the models was that we wanted to regress the dependent variable likeability against the independent variable detectability of objects (formula = likeability ~ objects). To model this, we used the brms package in R.[4] The models that treated the data as ordinal were modeled with the new brms monotonic models.[5]

The models of the section 1 (ordinal) and section 2 (interval) are built in the same way: the first model in each section is a model with only fixed effects, the second model is a hierarchical model with by-item (pictures) and by-subject random intercepts and the third model is a hierarchical model with by-subject random intercepts and fixed effect of artist. There are two important things to be considered regarding the choice of our models. The models treating the data as interval is not totally appropriate.

[4] P. Bürkner, "brms: An R package for Bayesian General Linear Mixed Models using stan" (2016)
[5] P. Bürkner, "Estimating Monotonic Effects with brms", in review (2019)

According to Bürkner (2019), the often used Likert-scales in Psychology research should be treated as ordinal instead of only treating them as continuous or as unordered categorical data out of convenience.[6] Therefore, we use the newly included brms monotonic models for the ordinal data. The second point is that the first and the fourth model (the two with only fixed effects) do not treat subject as a random effect. This means that these models do not include repeated measurements and therefore assume that we had 1500 participants (50 participants multiplied with 30 pictures) that were presented with one picture only.

Afterwards we compared the models in order to know which model we should choose and base our inference on. One formal approach to model comparison is to investigate the relative fit to the data of each of these models. One method to do so is called approximate leave-one-out cross-validation (LOO). The LOO function can be interpreted in the following way: the model with a lower LOO value fits the data better if the number of observations is large enough.[7] It is possible to even use the loo function when comparing models with interval data and our monotonic models.[8]

Besides the Bayesian analysis, we also used the frequentist approach from the original study for our analysis. This is only done to compare it to our main analysis and in order to replicate the analysis used in the original study. The authors tested the hypothesis by using a Pearson correlation coefficient.
As done by Muth et al (2012), we report the value of r and the p-value indicating whether the correlation is significant or not. The results are also depicted in a regression graph: on the x-axis, the detectability of objects is shown and on the y-axis, it shows how much subjects liked the painting. One point in the graph shows the values for a single painting. We further report the amount of explained variance r^2.

We did not choose this approach as our main analysis because it has a lot of limitations and is not totally appropriate for the experiment. The Pearson correlation presupposes interval data for both variables. When working with a Likert scale, the data is actually only ordinal. Furthermore, the authors did not test the assumption of a linear relationship as well as a bivariate normal distribution. Especially the last assumption requires to also execute a power analysis in order to determine the sample size needed to find a certain effect. All of this was not done in the original study. We stuck to the original analysis just to compare our results, but rely on the Bayesian approach for our main analysis.

[6] P. Bürkner, “Estimating Monotonic Effects with brms”, in review (2019)
[7] P. Bürkner, M. Vuorre, “Ordinal Regression Models in Psychology: A Tutorial”,
Advances in Methods and Practices in Psychological Science, Vol. 2, No. 1 (2019), pp. 18-19
[8] P. Bürkner, “Estimating Monotonic Effects with brms”, in review (2019)

Give Me Gestalt! Final Paper

Bayesian Analysis

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.1.1       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(brms)
```

```
## Loading required package: Rcpp
```

```
## Loading 'brms' package (version 2.8.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
```

```
library(ggplot2)
library(ggpubr)
```

```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##     set_names
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
# read in the filtered csv results and select the columns needed for analysis
d = read_csv2("C:/Users/annik/Desktop/uni/4 Semester/Psycho Lab/HW/3/rotation-task-with-_babe-master/GiveMeGestalt_filtered_results.csv") %>%
  filter(trial_name %in% c("rating_scale_object", "rating_scale_like")) %>%
  select(submission_id, trial_name, response, picture_nr, artist)
```

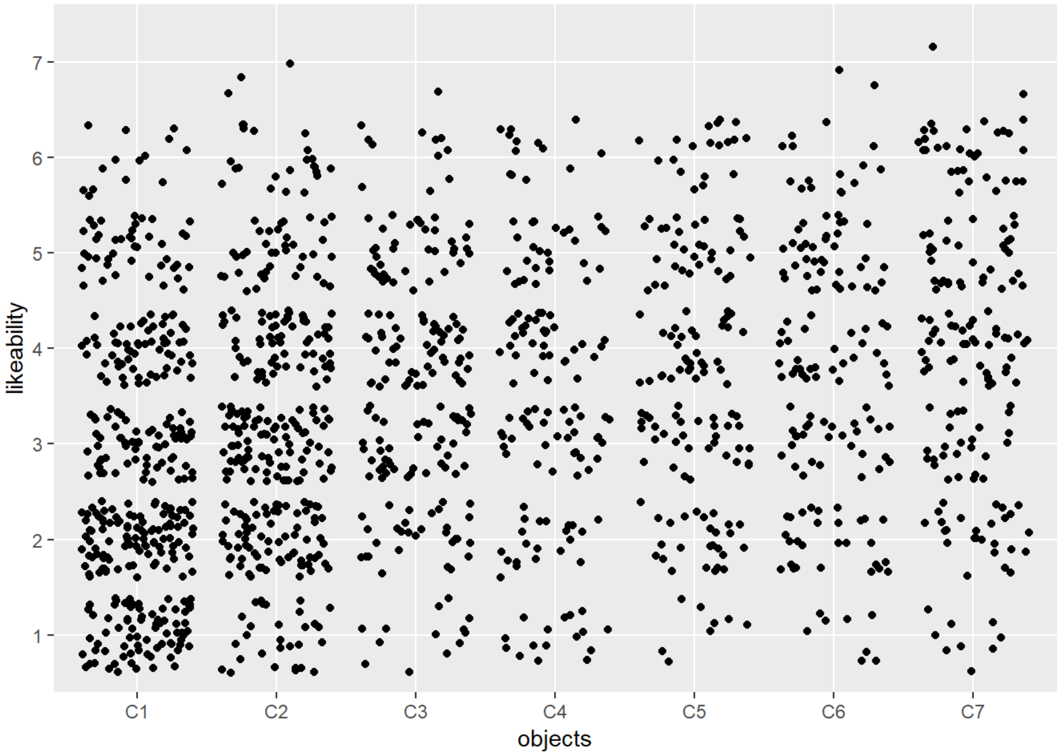
```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Warning: Missing column names filled in: 'X27' [27], 'X28' [28]
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   submission_id = col_double(),
##   QUD = col_logical(),
##   RT = col_double(),
##   age = col_double(),
##   endTime = col_double(),
##   experiment_id = col_double(),
##   min_chars = col_double(),
##   picture_nr = col_double(),
##   startTime = col_double(),
##   timeSpent = col_number(),
##   trial_number = col_double(),
##   X27 = col_logical(),
##   X28 = col_double()
## )
```

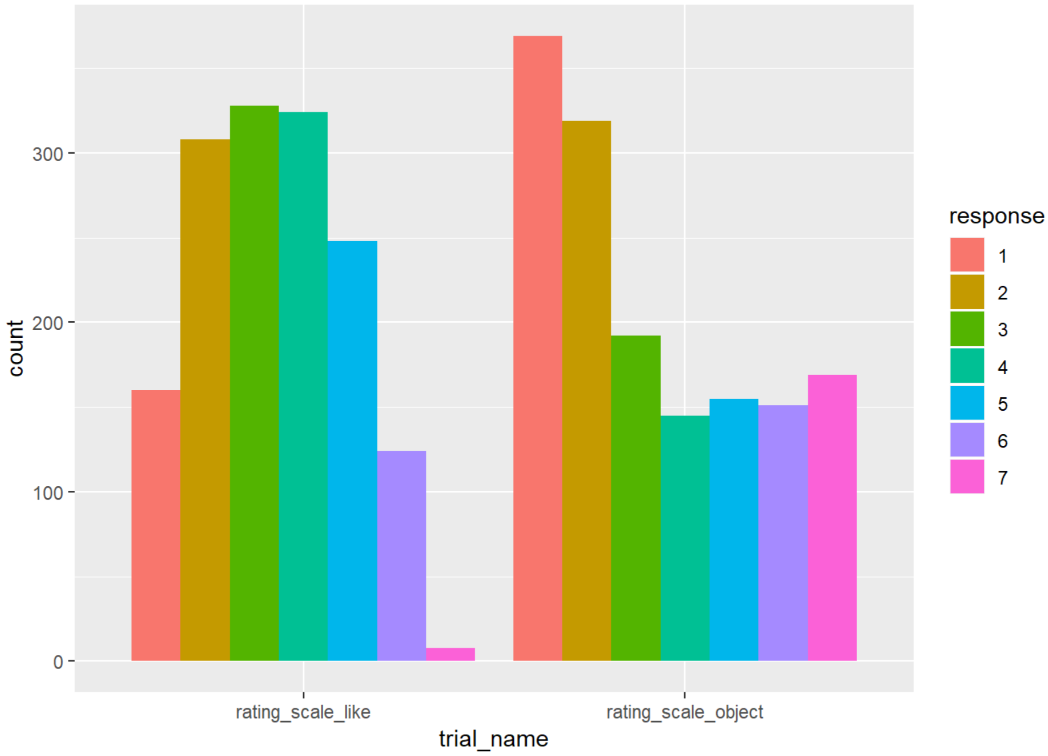
```
## See spec(...) for full column specifications.
```

```
# cast data into appropriate type
d_wide = spread(d, key = trial_name, value = response) %>%
  mutate(likeability = factor(rating_scale_like, ordered = T),
         objects = factor(paste0("C", rating_scale_object), ordered = T),
         artist = factor(artist),
         picture_nr = factor(picture_nr),
         submission_id = factor(submission_id),
         objects_forward = objects)
# inspect data
ggplot(d_wide, aes(x = objects, y= likeability)) + geom_jitter() + geom_smooth(method = "lm")
```

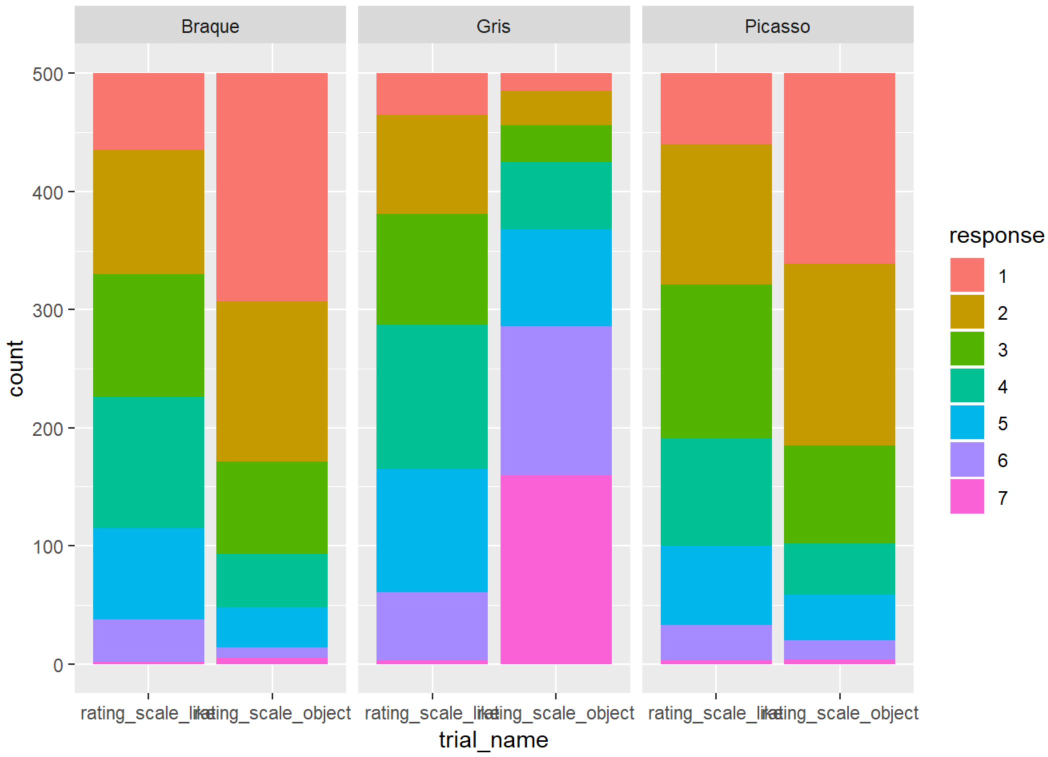


Visual display of the data

```
# absolute frequency of likeability and detecting objects
a <- ggplot(data=d)+
  geom_bar(mapping= aes(x = trial_name, fill = response), position = "dodge")
a
```




```r
# absolute frequency of likeability and detecting objects regarding the artists
b <- ggplot(data=d)+
  geom_bar(mapping= aes(x = trial_name, fill = response))+
  facet_wrap(~artist)
b
```



Section 1:

the first three models treat the object-ratings as ordinal using the new brms monotonic models

Hierarchical model with only fixed effects

```r
model_1 = brm(data = d_wide,
                formula = likeability ~ mo(objects),
                family=cumulative("logit")
)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL '43fb6a6fcb32083957aa1b13fbc3d9fd' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 29.348 seconds (Warm-up)
## Chain 1:                17.187 seconds (Sampling)
## Chain 1:                46.535 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '43fb6a6fcb32083957aa1b13fbc3d9fd' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 31.959 seconds (Warm-up)
## Chain 2:                16.563 seconds (Sampling)
## Chain 2:                48.522 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '43fb6a6fcb32083957aa1b13fbc3d9fd' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
```

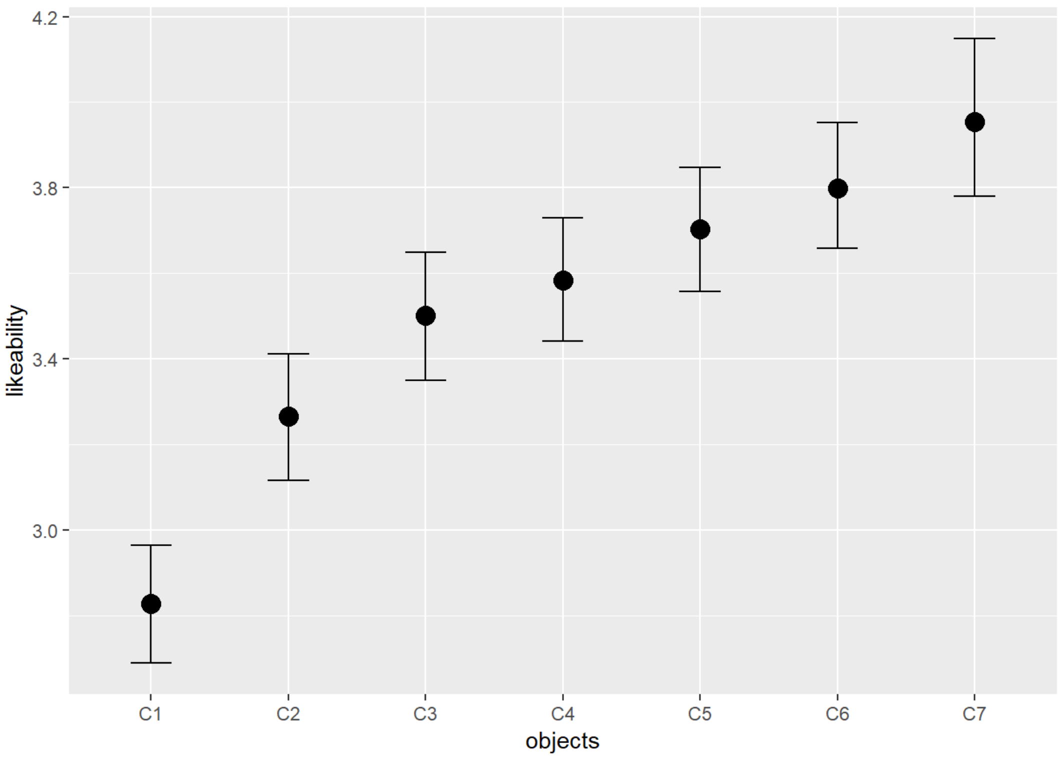
```
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 28.646 seconds (Warm-up)
## Chain 3:                20.776 seconds (Sampling)
## Chain 3:                49.422 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL '43fb6a6fcb32083957aa1b13fbc3d9fd' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 35.343 seconds (Warm-up)
## Chain 4:                20.65 seconds (Sampling)
## Chain 4:                55.993 seconds (Total)
## Chain 4: 
```

```
model_1
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: likeability ~ mo(objects)
##    Data: d_wide (Number of observations: 1500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]    -1.51      0.11    -1.72    -1.30       3155 1.00
## Intercept[2]    -0.12      0.10    -0.31     0.07       3922 1.00
## Intercept[3]     0.85      0.10     0.65     1.04       3925 1.00
## Intercept[4]     1.85      0.11     1.64     2.06       4137 1.00
## Intercept[5]     3.15      0.13     2.90     3.40       4578 1.00
## Intercept[6]     6.10      0.38     5.40     6.90       6009 1.00
## moobjects        1.41      0.15     1.11     1.71       4438 1.00
##
## Simplex Parameters:
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## moobjects1[1]     0.39      0.08     0.23     0.56       3226 1.00
## moobjects1[2]     0.21      0.09     0.03     0.41       2941 1.00
## moobjects1[3]     0.07      0.06     0.00     0.23       5419 1.00
## moobjects1[4]     0.10      0.07     0.01     0.27       6046 1.00
## moobjects1[5]     0.09      0.06     0.00     0.24       6448 1.00
## moobjects1[6]     0.13      0.08     0.01     0.30       5607 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(marginal_effects(model_1), categorical = T)
```

```
## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.
```



Hierarchical model with by-item (pictures) and by-subject random intercepts

```
model_2 = brm(data = d_wide,
              formula = likeability ~ mo(objects) + (1 | picture_nr) + (1 | submission_id),
              family=cumulative("logit")
)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL '73aac43371536bf55caf7bcba11bb5fb' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.002 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 52.765 seconds (Warm-up)
## Chain 1:                39.514 seconds (Sampling)
## Chain 1:                92.279 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '73aac43371536bf55caf7bcba11bb5fb' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 47.385 seconds (Warm-up)
## Chain 2:                40.453 seconds (Sampling)
## Chain 2:                87.838 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '73aac43371536bf55caf7bcba11bb5fb' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.002 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

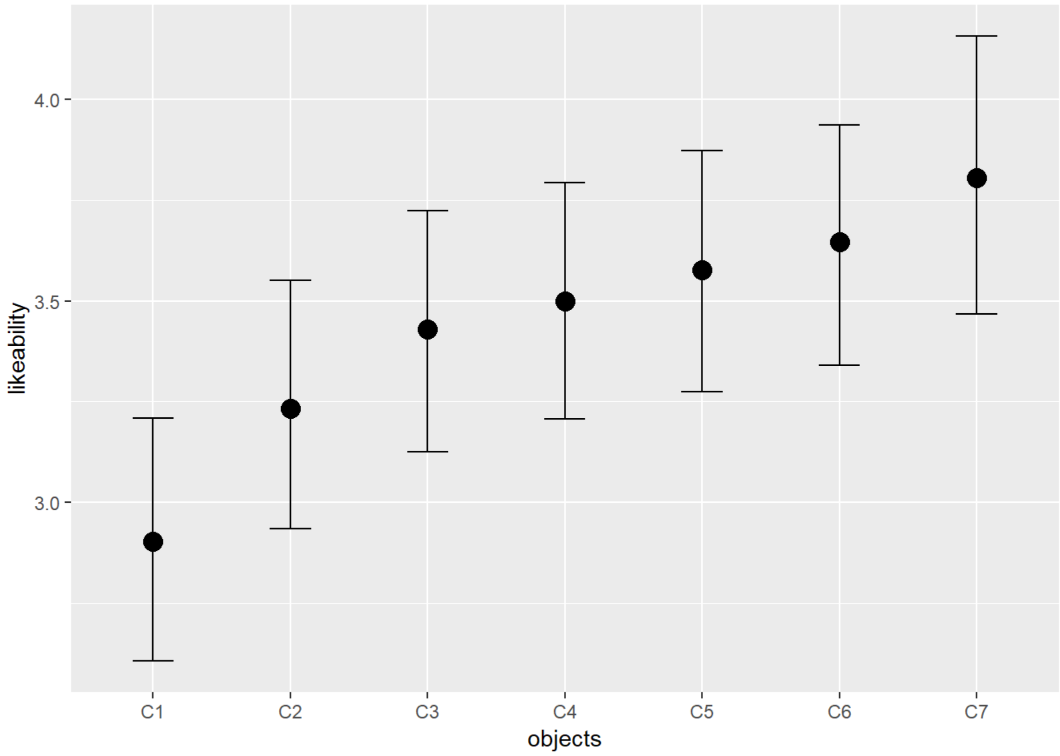
```
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3: 
## Chain 3:  Elapsed Time: 52.663 seconds (Warm-up)
## Chain 3:                48.386 seconds (Sampling)
## Chain 3:                101.049 seconds (Total)
## Chain 3: 
## 
## SAMPLING FOR MODEL '73aac43371536bf55caf7bcba11bb5fb' NOW (CHAIN 4).
## Chain 4: 
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4: 
## Chain 4: 
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4: 
## Chain 4:  Elapsed Time: 48.839 seconds (Warm-up)
## Chain 4:                43.504 seconds (Sampling)
## Chain 4:                92.343 seconds (Total)
## Chain 4: 
```

```
model_2
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: likeability ~ mo(objects) + (1 | picture_nr) + (1 | submission_id)
##    Data: d_wide (Number of observations: 1500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~picture_nr (Number of levels: 30)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.46      0.09     0.31     0.66       1441 1.00
##
## ~submission_id (Number of levels: 50)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.21      0.14     0.97     1.51       1002 1.00
##
## Population-Level Effects:
##              Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]    -2.14      0.25    -2.63    -1.65        943 1.00
## Intercept[2]    -0.34      0.24    -0.80     0.13        893 1.00
## Intercept[3]     0.90      0.24     0.44     1.36        905 1.01
## Intercept[4]     2.13      0.24     1.66     2.59        945 1.01
## Intercept[5]     3.59      0.25     3.10     4.07       1031 1.01
## Intercept[6]     6.66      0.44     5.82     7.53       2611 1.00
## moobjects        1.30      0.23     0.86     1.77       2693 1.00
##
## Simplex Parameters:
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## moobjects1[1]     0.38      0.11     0.18     0.59       4772 1.00
## moobjects1[2]     0.21      0.10     0.03     0.43       4717 1.00
## moobjects1[3]     0.08      0.06     0.00     0.23       5938 1.00
## moobjects1[4]     0.09      0.07     0.00     0.25       6427 1.00
## moobjects1[5]     0.07      0.06     0.00     0.22       6582 1.00
## moobjects1[6]     0.17      0.10     0.01     0.38       4977 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(marginal_effects(model_2), categorical = T)
```

```
## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.
```



Hierarchical model with by-subject random intercepts and fixed effect of artist

```
model_3 = brm(data = d_wide,
                formula = likeability ~ mo(objects) + artist + (1 | submission_id),
                family=cumulative("logit")
)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL 'a84b2e54d20b4a8ad3940a14d75398f4' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.003 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 30 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 38.794 seconds (Warm-up)
## Chain 1:                49.487 seconds (Sampling)
## Chain 1:                88.281 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'a84b2e54d20b4a8ad3940a14d75398f4' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.004 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 40 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
```

```
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 37.601 seconds (Warm-up)
## Chain 2:                38.668 seconds (Sampling)
## Chain 2:                76.269 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'a84b2e54d20b4a8ad3940a14d75398f4' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 51.188 seconds (Warm-up)
## Chain 3:                56.319 seconds (Sampling)
## Chain 3:                107.507 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'a84b2e54d20b4a8ad3940a14d75398f4' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.003 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 30 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 63.762 seconds (Warm-up)
## Chain 4:                34.323 seconds (Sampling)
## Chain 4:                98.085 seconds (Total)
## Chain 4:
```

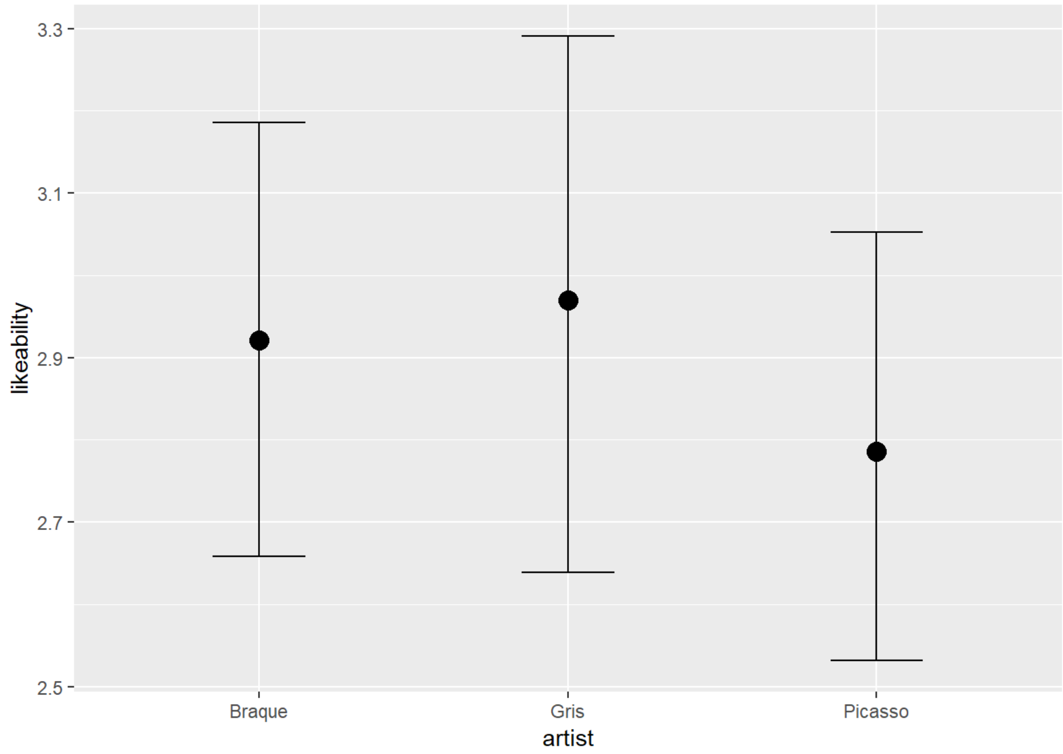
```
model_3
```

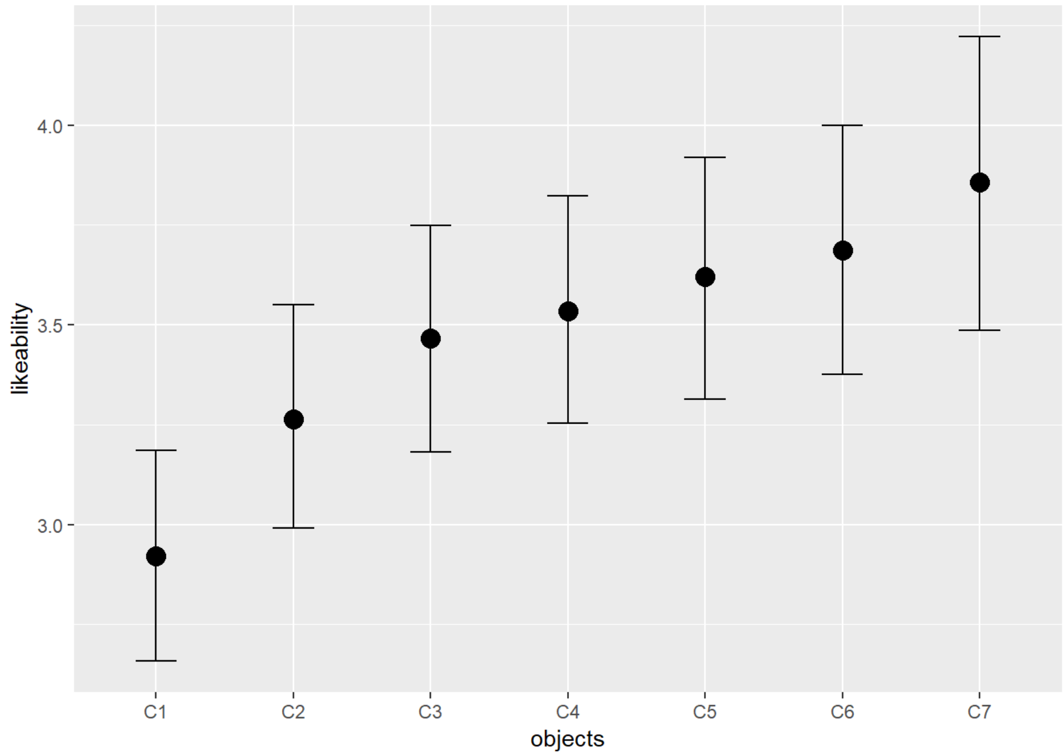
```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: likeability ~ mo(objects) + artist + (1 | submission_id)
##    Data: d_wide (Number of observations: 1500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~submission_id (Number of levels: 50)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.16      0.14     0.93     1.45        850 1.00
##
## Population-Level Effects:
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]     -2.09      0.21    -2.50    -1.67        839 1.00
## Intercept[2]     -0.34      0.20    -0.73     0.06        848 1.00
## Intercept[3]      0.85      0.20     0.46     1.26        922 1.00
## Intercept[4]      2.04      0.21     1.64     2.45        889 1.00
## Intercept[5]      3.47      0.22     3.04     3.91       1079 1.00
## Intercept[6]      6.50      0.41     5.72     7.36       2502 1.00
## artistGris        0.07      0.15    -0.23     0.36       4024 1.00
## artistPicasso    -0.20      0.12    -0.43     0.03       4585 1.00
## moobjects         1.32      0.23     0.88     1.79       3374 1.00
##
## Simplex Parameters:
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## moobjects1[1]     0.38      0.11     0.17     0.59       4253 1.00
## moobjects1[2]     0.22      0.11     0.03     0.44       4209 1.00
## moobjects1[3]     0.07      0.06     0.00     0.22       4955 1.00
## moobjects1[4]     0.09      0.07     0.00     0.25       5191 1.00
## moobjects1[5]     0.07      0.06     0.00     0.22       5522 1.00
## moobjects1[6]     0.17      0.10     0.01     0.36       3874 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(marginal_effects(model_3), categorical = T)
```

```
## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.

## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.
```





Section 2:

The following three models treat the object-ratings as interval-scale/ metric

Hierarchical model with only fixed effects

```
model_4 = brm(data = d_wide,
        formula = likeability ~ as.double(objects),
        family=cumulative("logit")
)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL 'd9fe23fafc57c2777615d13508974caf' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 10.718 seconds (Warm-up)
## Chain 1:                11.135 seconds (Sampling)
## Chain 1:                21.853 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'd9fe23fafc57c2777615d13508974caf' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 10.971 seconds (Warm-up)
## Chain 2:                13.713 seconds (Sampling)
## Chain 2:                24.684 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'd9fe23fafc57c2777615d13508974caf' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 14.134 seconds (Warm-up)
## Chain 3:                10.511 seconds (Sampling)
```

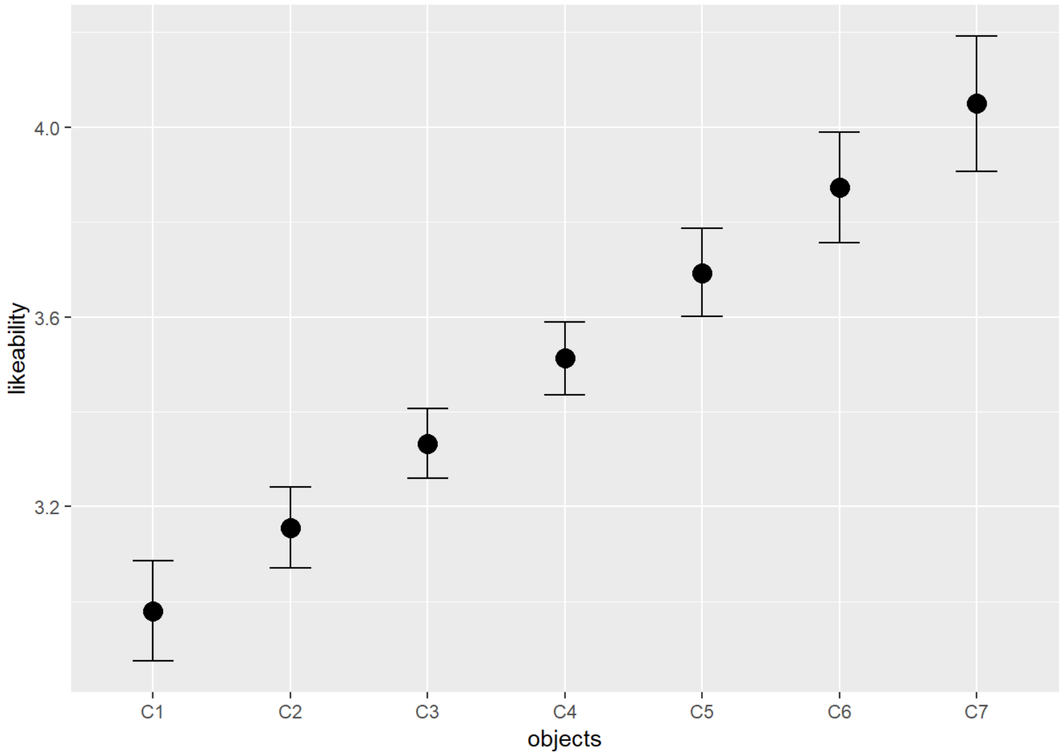
```
## Chain 3:                24.645 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'd9fe23fafc57c2777615d13508974caf' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 16.005 seconds (Warm-up)
## Chain 4:                14.275 seconds (Sampling)
## Chain 4:                30.28 seconds (Total)
## Chain 4:
```

```
model_4
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: likeability ~ as.double(objects)
##    Data: d_wide (Number of observations: 1500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]       -1.47      0.11    -1.67    -1.26       3828 1.00
## Intercept[2]       -0.09      0.09    -0.26     0.09       4973 1.00
## Intercept[3]        0.87      0.09     0.69     1.04       4591 1.00
## Intercept[4]        1.87      0.10     1.67     2.06       4419 1.00
## Intercept[5]        3.17      0.13     2.92     3.42       4477 1.00
## Intercept[6]        6.13      0.38     5.46     6.93       4114 1.00
## as.doubleobjects    0.22      0.02     0.18     0.27       4342 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(marginal_effects(model_4))
```

```
## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.
```



Hierarchical model with by-item (pictures) and by-subject random intercepts

```
model_5 = brm(data = d_wide,
        formula = likeability ~ as.double(objects) + (1 | picture_nr) + (1 | submission_id),
        family=cumulative("logit")
)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL 'ef9da17aa5dc87bf80d86915ebda8075' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0.001 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 34.442 seconds (Warm-up)
## Chain 1:                23.605 seconds (Sampling)
## Chain 1:                58.047 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL 'ef9da17aa5dc87bf80d86915ebda8075' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.001 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
```

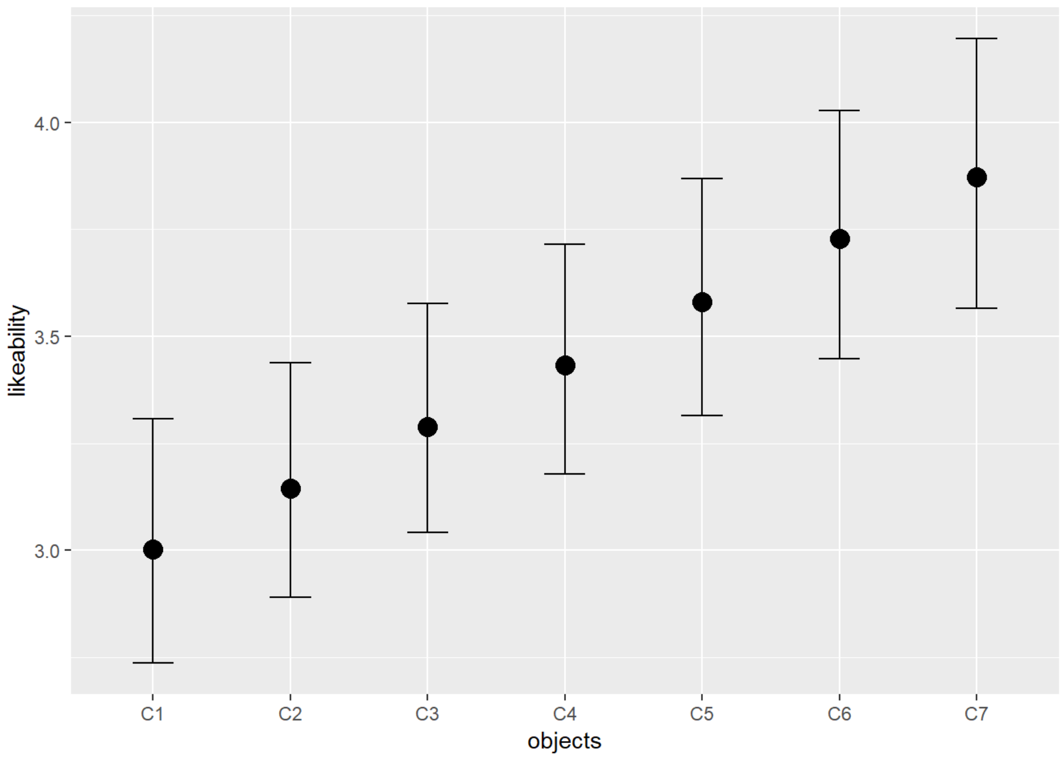
```
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 40.449 seconds (Warm-up)
## Chain 2:                48.695 seconds (Sampling)
## Chain 2:                89.144 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL 'ef9da17aa5dc87bf80d86915ebda8075' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 33.399 seconds (Warm-up)
## Chain 3:                23.757 seconds (Sampling)
## Chain 3:                57.156 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL 'ef9da17aa5dc87bf80d86915ebda8075' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 31.373 seconds (Warm-up)
## Chain 4:                29.579 seconds (Sampling)
## Chain 4:                60.952 seconds (Total)
## Chain 4:
```

```
model_5
```

```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: likeability ~ as.double(objects) + (1 | picture_nr) + (1 | submission_id)
##    Data: d_wide (Number of observations: 1500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~picture_nr (Number of levels: 30)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     0.45      0.08     0.31     0.63       1399 1.00
##
## ~submission_id (Number of levels: 50)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.21      0.14     0.97     1.51        643 1.00
##
## Population-Level Effects:
##                  Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]        -2.07      0.25    -2.57    -1.60        698 1.00
## Intercept[2]        -0.29      0.23    -0.76     0.15        644 1.00
## Intercept[3]         0.94      0.24     0.47     1.38        630 1.00
## Intercept[4]         2.17      0.24     1.68     2.64        663 1.00
## Intercept[5]         3.64      0.26     3.12     4.12        745 1.00
## Intercept[6]         6.71      0.44     5.90     7.58       1499 1.00
## as.doubleobjects     0.21      0.04     0.14     0.28       2207 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(marginal_effects(model_5))
```

```
## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.
```



Hierarchical model with by-subject random intercepts and fixed effect of artist

```
model_6 = brm(data = d_wide,
        formula = likeability ~ as.double(objects) + artist + (1 | submission_id),
        family=cumulative("logit")
)
```

```
## Compiling the C++ model
```

```
## Start sampling
```

```
##
## SAMPLING FOR MODEL '1b726ff50c235e73d05586071a94906d' NOW (CHAIN 1).
## Chain 1:
## Chain 1: Gradient evaluation took 0 seconds
## Chain 1: 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Chain 1: Adjust your expectations accordingly!
## Chain 1:
## Chain 1:
## Chain 1: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 1: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 1: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 1: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 1: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 1: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 1: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 1: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 1: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 1: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 1: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 1: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 1:
## Chain 1:  Elapsed Time: 25.727 seconds (Warm-up)
## Chain 1:                23.468 seconds (Sampling)
## Chain 1:                49.195 seconds (Total)
## Chain 1:
##
## SAMPLING FOR MODEL '1b726ff50c235e73d05586071a94906d' NOW (CHAIN 2).
## Chain 2:
## Chain 2: Gradient evaluation took 0.002 seconds
## Chain 2: 1000 transitions using 10 leapfrog steps per transition would take 20 seconds.
## Chain 2: Adjust your expectations accordingly!
## Chain 2:
## Chain 2:
## Chain 2: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 2: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 2: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 2: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 2: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 2: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 2: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 2: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 2: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 2: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 2: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 2: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 2:
## Chain 2:  Elapsed Time: 38.263 seconds (Warm-up)
## Chain 2:                17.78 seconds (Sampling)
## Chain 2:                56.043 seconds (Total)
## Chain 2:
##
## SAMPLING FOR MODEL '1b726ff50c235e73d05586071a94906d' NOW (CHAIN 3).
## Chain 3:
## Chain 3: Gradient evaluation took 0.001 seconds
## Chain 3: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 3: Adjust your expectations accordingly!
## Chain 3:
## Chain 3:
## Chain 3: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 3: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 3: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 3: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 3: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 3: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 3: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 3: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 3: Iteration: 1400 / 2000 [ 70%]  (Sampling)
```

```
## Chain 3: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 3: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 3: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 3:
## Chain 3:  Elapsed Time: 26.018 seconds (Warm-up)
## Chain 3:                18.125 seconds (Sampling)
## Chain 3:                44.143 seconds (Total)
## Chain 3:
##
## SAMPLING FOR MODEL '1b726ff50c235e73d05586071a94906d' NOW (CHAIN 4).
## Chain 4:
## Chain 4: Gradient evaluation took 0.001 seconds
## Chain 4: 1000 transitions using 10 leapfrog steps per transition would take 10 seconds.
## Chain 4: Adjust your expectations accordingly!
## Chain 4:
## Chain 4:
## Chain 4: Iteration:    1 / 2000 [  0%]  (Warmup)
## Chain 4: Iteration:  200 / 2000 [ 10%]  (Warmup)
## Chain 4: Iteration:  400 / 2000 [ 20%]  (Warmup)
## Chain 4: Iteration:  600 / 2000 [ 30%]  (Warmup)
## Chain 4: Iteration:  800 / 2000 [ 40%]  (Warmup)
## Chain 4: Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Chain 4: Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Chain 4: Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Chain 4: Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Chain 4: Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Chain 4: Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Chain 4: Iteration: 2000 / 2000 [100%]  (Sampling)
## Chain 4:
## Chain 4:  Elapsed Time: 23.593 seconds (Warm-up)
## Chain 4:                16.334 seconds (Sampling)
## Chain 4:                39.927 seconds (Total)
## Chain 4:
```

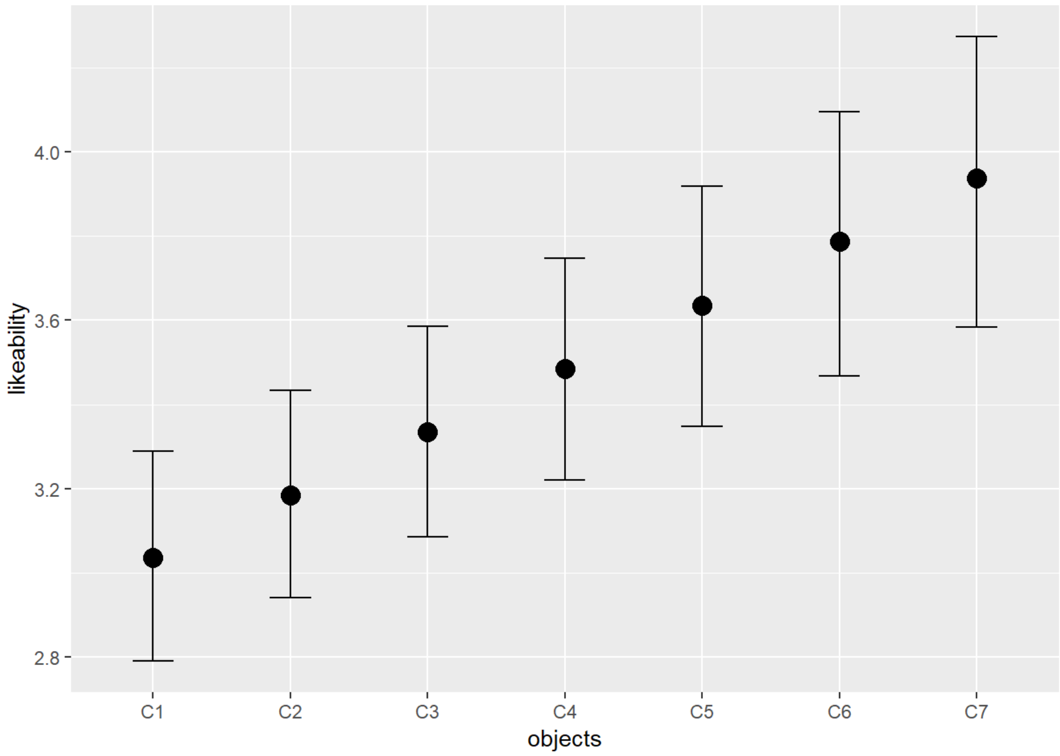
```
model_6
```

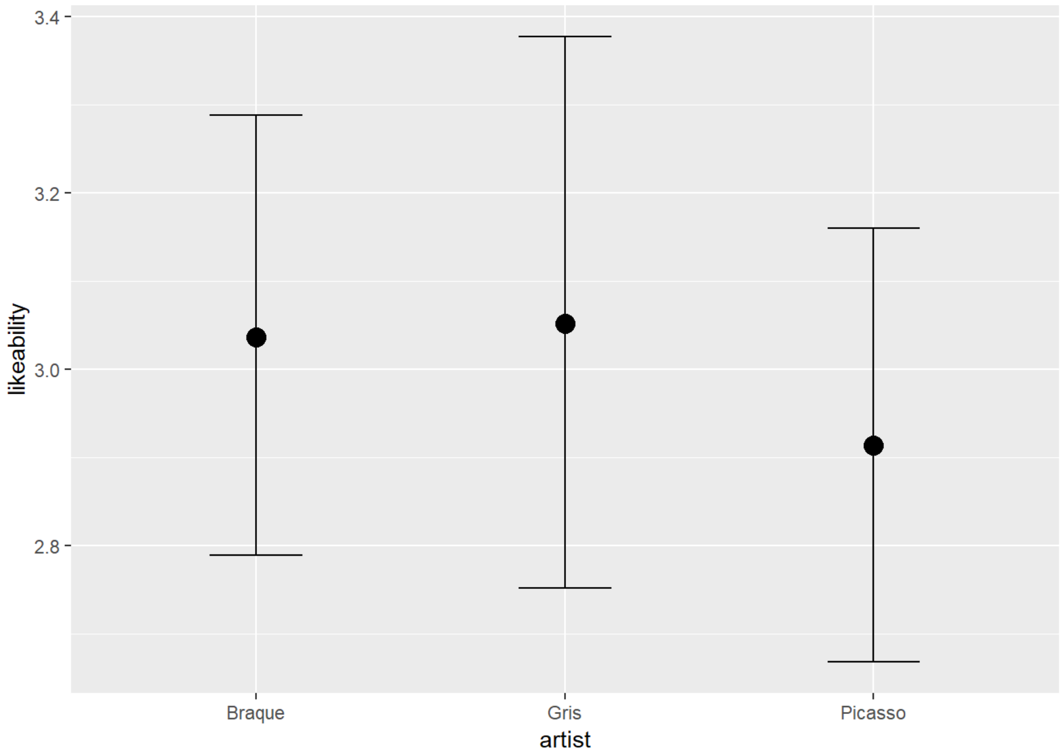
```
##  Family: cumulative
##   Links: mu = logit; disc = identity
## Formula: likeability ~ as.double(objects) + artist + (1 | submission_id)
##    Data: d_wide (Number of observations: 1500)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~submission_id (Number of levels: 50)
##               Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## sd(Intercept)     1.16      0.13     0.93     1.45        742 1.00
##
## Population-Level Effects:
##                  Estimate Est.Error l-95% CI u-95% CI Eff.Sample Rhat
## Intercept[1]        -2.03      0.21    -2.45    -1.62        547 1.00
## Intercept[2]        -0.30      0.20    -0.69     0.09        523 1.00
## Intercept[3]         0.89      0.20     0.50     1.29        542 1.00
## Intercept[4]         2.08      0.20     1.68     2.49        570 1.00
## Intercept[5]         3.50      0.22     3.08     3.94        662 1.00
## Intercept[6]         6.53      0.41     5.77     7.38       1573 1.00
## as.doubleobjects     0.21      0.04     0.14     0.28       2564 1.00
## artistGris           0.02      0.16    -0.27     0.34       2369 1.00
## artistPicasso       -0.18      0.12    -0.41     0.04       3902 1.00
##
## Samples were drawn using sampling(NUTS). For each parameter, Eff.Sample
## is a crude measure of effective sample size, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
plot(marginal_effects(model_6))
```

```
## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.

## Warning: Predictions are treated as continuous variables in
## 'marginal_effects' by default, which is likely invalid for ordinal
## families. Please set 'categorical' to TRUE.
```





Model comparison

```
loo <- loo(model_1, model_2, model_3, model_4, model_5, model_6)
```

```
## Warning: Passing multiple brmsfit objects to 'loo' and related methods is
## deprecated. Please see ?loo.brmsfit for the recommended workflow.
```

```r
loo
```

```
## Output of model 'model_1':
##
## Computed from 4000 by 1500 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2592.9 17.9
## p_loo         9.6  0.4
## looic      5185.8 35.8
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Output of model 'model_2':
##
## Computed from 4000 by 1500 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2356.2 24.8
## p_loo        80.2  2.0
## looic      4712.5 49.6
## ------
## Monte Carlo SE of elpd_loo is 0.2.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Output of model 'model_3':
##
## Computed from 4000 by 1500 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2386.4 24.1
## p_loo        60.2  1.5
## looic      4772.8 48.2
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Output of model 'model_4':
##
## Computed from 4000 by 1500 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2597.6 17.7
## p_loo         7.1  0.4
## looic      5195.2 35.3
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Output of model 'model_5':
##
## Computed from 4000 by 1500 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2360.2 24.6
## p_loo        78.3  1.9
## looic      4720.3 49.2
## ------
## Monte Carlo SE of elpd_loo is 0.2.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
```

```
##
## Output of model 'model_6':
##
## Computed from 4000 by 1500 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2390.1 24.0
## p_loo        57.9  1.4
## looic      4780.1 47.9
## ------
## Monte Carlo SE of elpd_loo is 0.1.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
##
## Model comparisons:
##         elpd_diff se_diff
## model_2    0.0       0.0
## model_5   -3.9       2.5
## model_3  -30.1       7.7
## model_6  -33.8       8.1
## model_1 -236.7      20.7
## model_4 -241.4      20.9
```

Frequentist Analysis: Pearson correlation

```
data <- read_csv2("GiveMeGestalt_filtered_results.csv")
```

```
## Using ',' as decimal and '.' as grouping mark. Use read_delim() for more control.
```

```
## Warning: Missing column names filled in: 'X27' [27], 'X28' [28]
```

```
## Parsed with column specification:
## cols(
##   .default = col_character(),
##   submission_id = col_double(),
##   QUD = col_logical(),
##   RT = col_double(),
##   age = col_double(),
##   endTime = col_double(),
##   experiment_id = col_double(),
##   min_chars = col_double(),
##   picture_nr = col_double(),
##   startTime = col_double(),
##   timeSpent = col_number(),
##   trial_number = col_double(),
##   X27 = col_logical(),
##   X28 = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

Data formatting

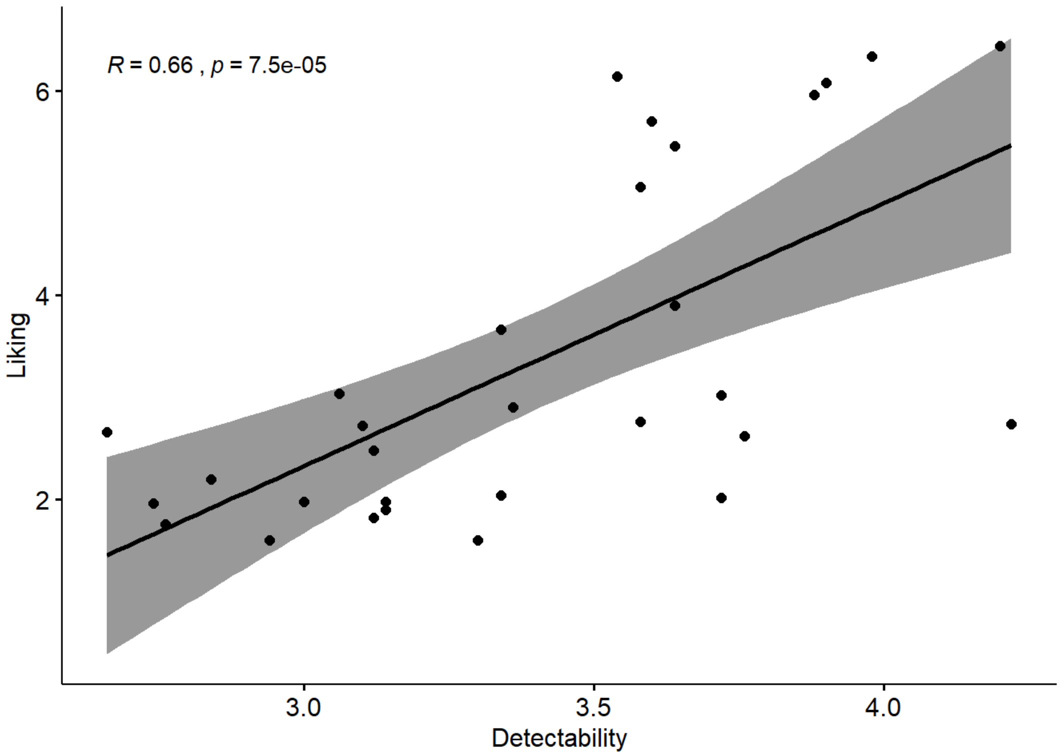
```
data_temp <- as_tibble(data) %>% mutate(response = as.integer(response))
```

```
## Warning: NAs durch Umwandlung erzeugt
```

```
x <- filter(data_temp, trial_name == 'rating_scale_like') %>%
  select(c('response', 'picture_nr'))%>%
  group_by(picture_nr) %>%
  summarise(response = mean(response))
y <- filter(data_temp, trial_name == 'rating_scale_object') %>%
  select(c('response', 'picture_nr')) %>%
  group_by(picture_nr) %>%
  summarise(response = mean(response))
data_formatted <- merge(x,y, by = 'picture_nr')
```

Correlation test and regression graph

```
ggscatter(data_formatted, x = "response.x", y = "response.y",
          add = "reg.line", conf.int = TRUE,
          cor.coef = TRUE, cor.method = "pearson",
          xlab = "Detectability", ylab = "Liking")
```




```
res <- cor.test(data_formatted$response.x, data_formatted$response.y,
                    method = "pearson")
res
```

```
##
##  Pearson's product-moment correlation
##
## data:  data_formatted$response.x and data_formatted$response.y
## t = 4.6356, df = 28, p-value = 7.508e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3916580 0.8236774
## sample estimates:
##       cor
## 0.6589481
```

```
# extract the p.value
res$p.value
```

```
## [1] 7.508036e-05
```

```
# extract the correlation coefficient
res$estimate
```

```
##       cor
## 0.6589481
```

```
# the amount of variance explained
res$estimate^2
```

```
##       cor
## 0.4342126
```

Visual display of the data

Before we attend to our models and further visual displays, we first take a look at our initial scatterplot (ggplot). The scatterplot shows the correlation between likeability and object detectability ratings for each picture. The data points in our plot are very scattered and one cannot see a strong correlation or development in the data. There is however, a noticeable cluster of data points in the low rated section ("1" on the Likert scale).

The first visual display of the data, indicated in a bar chart, shows the absolute frequency of how often each response of the Likert-scale was chosen. Regarding the likeability rating, the responses 2,3 and 4 were chosen most often. This demonstrates that the participants tended to choose the central values above-average. Whilst as for the ratings of object detectability, the responses 1 and 2 were chosen most frequently. We conclude from this that the detectability of objects was generally low for our stimuli. Particularly striking is also that the other responses were chosen approximately equally often.

The second chart presents the different responses for the three artists individually. One can see that for Gris the least chosen option was "1", for both likeability and detecting objects. Overall Gris' paintings were liked the most. Furthermore, the response "7" was rated most often for how well one can detect objects. Surprisingly the charts for Braque and Picasso are quite similar. A small difference between the two is that for Braque the response "1" was selected most frequently.

Model analysis

For our main Bayesian analysis, only certain aspects of the calculated models are relevant. The intercepts are usually not that relevant for the modeling process, the focus lies on the regression coefficients.[1] In essence this means that we look at the slope of the regression. This value can be found in the last row of the respective models in the column "estimate". The credible interval with its boundaries shown in the second and third column indicates whether the result is significant or not. It shows that with 95% probability, the slope is different from 0 and that its value lies in the calculated interval.

All of our models are indicating that the slope is significantly different from 0, implicating that there is a positive relation between the detectability of objects and likeability. This signals that the models all seem to support the hypothesis, even though they fit and describe the data differently well. In the next step, we want to identify which model fits the data best.

When using the LOO function to compare the models, the output shows the looic value for each model. The best model is pointed out by having the smallest looic value. Under "Model comparisons", the column "elpd_diff" shows the value differences between the models. They are ordered from the best fitting one to the worst one.

We can see that model 2 is the best one, since it has the lowest value. Model 2 is the hierarchical model with by-item (pictures) and by-subject random intercepts. The estimate of model 2 is 1.30. The limits of the credible intervals are 0.86 and 1.77. The looic value is 4.712,5 and the elpd_diff is 0.0 (since it is the best model). The graph of model 2 shows the positive relation between the detectability of the objects and the likeability, i.e. the pictures are liked more if objects could be recognised more easily.

[1]P. Bürkner, M. Vuorre, "Ordinal Regression Models in Psychology: A Tutorial",
Advances in Methods and Practices in Psychological Science, Vol. 2, No. 1 (2019), p.14

The largest increase in likeability ratings occurs between the two lowest ratings for object detectability, namely options “1” and “2” on the Likert scale. The second largest increase in likeability ratings occurs between “2” and “3”.

Model 2 is followed by model 5, 3, 6, 1 and then model 4. These results fit our expectations in the sense that the most complex model treating the data as ordinal is the one that actually fits the data best. Since we stated above that it is controversial to use the interval models for analysis, we also look at the model comparisons of only the ordinal models, i.e. excluding model 4, 5 and 6 and only including model 1, 2 and 3. When doing so, model 2, 3 and then 1 are the best models in descending order. Since we further stated that the models with fixed effects actually do not treat subject as a random effect, we also look at which model is the best when excluding the interval models and the models with only fixed effects. Then only the models 2 and 3 are good models.

Looking at the monotonic model 3 reveals how much the participants’ ratings for likeability and object detectability diverged for the different painters. For Picasso the estimate is -0.20 and one can infer that the participants tended to slightly dislike the paintings of Picasso in comparison to Braque and Gris. However, the limits of the credible interval are 0.12 and -0.43, so we are not able to state this with 95% probability. For Gris the estimate is 0.07 and the limits of the credible interval are -0.23 and 0.36. Thus one can interpret this as Gris being slightly more liked by participants. As 0 is again included in the credible interval, we cannot claim this with a 95% probability. Overall we can conclude that the depicted population is in accordance with the visual display of the data.

For the sake of comparison to the original paper, we consider the frequentist approach. The Pearson correlation coefficient is $r=0.66$ with $p < 0.0001$. The amount of variance explained is $R^2 = 0.43$. Relying on the common frequentist interpretation of the p-value and the fact that 0 is not included in the confidence interval, it can be said that the alternative hypothesis is supported: the true correlation is significantly different from 0. This suggests that participants liked a painting more, the easier they could detect objects in it. Our result of the frequentist analysis section corresponds to the result of the original paper. The correlation coefficient is also in the same range ($r = 0.781$ in the original paper). Both correlations would be labeled as a large effect by the conventionally used classification of Cohen (large starting at $r=0.50$).[2]

When comparing the Bayesian and frequentist approach, we can see that both indicate that there is a positive relation between the detectability of objects in a painting and the likeability of the painting. This confirms the results of the original paper. The Bayesian results show a weaker relation than is indicated by the Pearson correlation coefficient - as far as one can compare the two approaches.

Discussion and Limitations

One critical point to be discussed is the definition of an object. During the experiment, participants were asked how easily they were able to detect objects in the paintings. This raises the question of what is meant by an object. For some people, this might mean that they recognize some geometrical forms whereas others interpret this as seeing specific objects from daily life. It is also possible that people understood the question as how many objects they were able to see in the paintings and indicated this number in the Likert-scale.

[2] J. Cohen, “Statistical Power Analysis for the Behavioral Sciences”, Second Edition, LEA (1988), p.83

Another important aspect is the definition of being an art expert. The participants themselves had to indicate how much they are an expert on cubism. Some people rather assess themselves as being experts in something with having only little knowledge while others already know a lot but do not consider themselves an expert in the field. Therefore, this criterion is a little vague and dependent on interpretation.

As we did an online replication of an experiment performed in a laboratory setting, we cannot guarantee that all pre-set requirements were complied with. This especially includes the required distance of 55cm between the desktop and the participant. Also we cannot be sure if someone participated on their smartphone. Additionally, the eyesight tests we utilised are not equivalent to a professional vision test performed by an optician as they were cut very short and are adjusted to our online setting.

Conclusion

In conclusion our results support our hypothesis “The more easily individuals without expertise in cubism can detect objects in the presented paintings from Braque, Gris and Picasso, the higher the likeability is rated”. Both of our analytic approaches - Bayesian and frequentist - indicate a positive relation between detectability and liking. Therefore, we can conclude that the data confirms the results from the original study. As mentioned earlier this replication study does not guarantee the compliance with all preset criteria and thus is not completely reliable.