

Familiar sequences are processed faster than unfamiliar sequences,
even when they do not match the count-list

Devlin et al. (2024)

Analysis

Data quality and accuracy

60 participants completed a comparative judgement task, an order verification task, and an arithmetic fluency task.

We first checked accuracy on the order verification task to establish whether participants understood the task instructions and completed the task correctly.

An example of the order verification task data is shown below:

Table 1: First five rows of order dataset

ppt	seq	key_resp	corr_resp	RT	block	order	consec	direction	dist_n	type
101	4 3 2	q	1	1.05	1	1	C	dsc	1	D1
101	4 2 6	p	1	0.91	1	0	NA	NA	2	N2
101	9 6 3	q	1	1.04	1	1	NC	dsc	3	D3
101	9 6 3	q	1	1.12	1	1	NC	dsc	3	D3
101	6 8 7	p	1	2.17	1	0	NA	NA	1	N1

We first checked participants overall accuracy on the order verification task:

```
percent_overall <- order_df %>%
  group_by(ppt) %>%
  count(corr_resp) %>%
  mutate(percent = n / sum(n)) %>%
  filter(corr_resp == 1) %>%
  select(ppt, percent_correct = percent)
```

```
macc <- mean(percent_overall$percent_correct)
sdacc <- sd(percent_overall$percent_correct)

lowerbound <- macc - (sdacc * 3)

below3 <- percent_overall %>%
  filter(percent_correct < lowerbound)
```

This revealed that two participants displayed unusually low accuracy (i.e., 3 SDs below mean):

Table 2: Accuracy 3 SDs below the mean

ppt	percent_correct
610	0.59
617	0.66

Furthermore, we also observed two other participants to display below chance accuracy on certain sequence types:

```
percent_type <- order_df %>%
  group_by(ppt, type) %>%
  count(corr_resp) %>%
  mutate(percent = n / sum(n)) %>%
  filter(corr_resp == 1) %>%
  select(ppt, percent_correct = percent)
```

```
## Adding missing grouping variables: 'type'
```

```
below_chance <- percent_type %>%
  filter(percent_correct < 0.5)
```

Table 3: Accuracy below chance

type	ppt	percent_correct
A3	215	0.40
D3	215	0.20
D3	220	0.40
D3	610	0.47
D2	617	0.34
D3	617	0.43

Therefore, data from these participants were removed from all tasks - resulting in 56 participants being included in the final analyses:

```
order_df <- order_df %>%
  filter(!ppt %in% c("220", "215", "610", "617"))

CJ_df <- CJ_df %>%
  filter(!judgeName %in% c("220", "215", "610", "617"))
```

Comparative judgement analysis

To produce a ranking and relative scoring of sequences, the raw judgement data (i.e., comparisons) from all participants were fitted to the Bradley-Terry (1952) model.

For ease of interpretation, we then re-scaled these scores between 0 and 100:

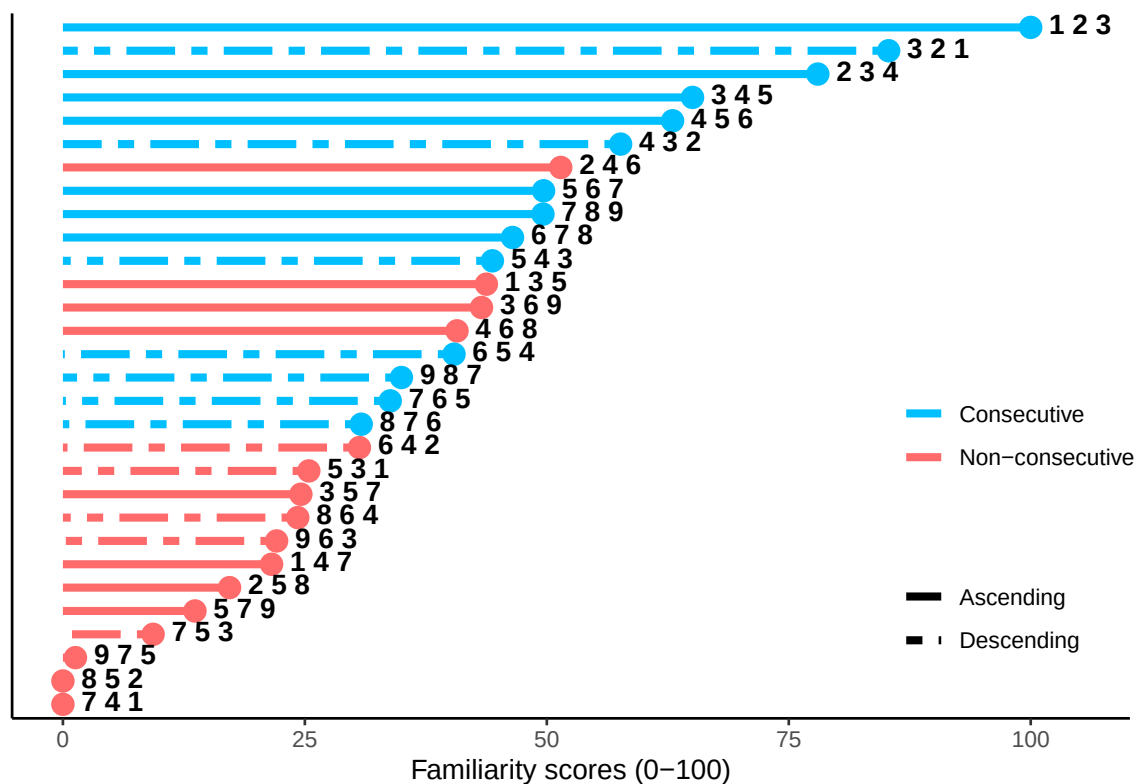
```
# Scale theta scores to between 0 and 100
scale <- function(x) (x-min(x))/(max(x) - min(x)) * 100
fam_scores$scaled_score <- scale(fam_scores$fam_score)
```

To establish the reliability of these judgements, we considered two reliability measures common in the comparative judgement literature. First, we calculated scale separation reliability – a measure of internal consistency analogous to Cronbach’s alpha. This returned a score of .997, indicating very good internal consistency.

Second, we considered inter-rater reliability – a reliability measure calculated by repeatedly splitting the judges (i.e., participants) into two groups and then calculating the correlation between their produced scores.

Following this procedure with 1000 repetitions, we observed a median Pearson product-moment correlation coefficient of .99, suggesting very good inter-rater reliability. Overall, these measures suggest a high degree of agreement across participants in their judgements.

The familiarity scores generated by the comparative judgement task can be seen below:



```
fam_scores %>%
  group_by(distance) %>%
  summarise(mean = mean(scaled_score))
```

```
## # A tibble: 2 x 2
##   distance mean
##   <chr>     <dbl>
## 1 C         55.7
## 2 N         23.1
```

```
fam_scores %>%
  group_by(direction) %>%
  summarise(mean = mean(scaled_score))
```

```
## # A tibble: 2 x 2
##   direction mean
##   <chr>      <dbl>
## 1 A         47.2
## 2 D         29.4
```

Consistent with our expectations, consecutive sequences (mean familiarity score = 55.65) were judged as more familiar than non-consecutive sequences (mean = 23.08). Similarly, ascending sequences (mean = 47.20) were judged as more familiar than descending sequences (mean = 29.36).

Furthermore, we note that 1-2-3 was judged as the most familiar sequence (familiarity score = 100), which is consistent with suggestions this sequence is processed atypically fast due to its high familiarity (i.e., Sella et al., 2020).

We also observed 2-4-6 to be the most familiar non-consecutive sequence (familiarity score = 51.44). In fact, 2-4-6 was judged as being more familiar than over half (8 out of 14) of the consecutive sequences.

Similarly, 3-6-9 (familiarity score = 43.25) and 4-6-8 (familiarity score = 40.72) had relatively high familiarity scores despite being non-consecutive. This is consistent with suggestions that sequences that are a part of highly practiced multiplication tables are likely to be highly familiar (i.e., Vos et al., 2021).

Order verification analysis

Each participant completed 600 order verification trials, meaning 33,600 responses were recorded across 56 participants.

```
percent_overall <- order_df %>%
  group_by(ppt) %>%
  count(corr_resp) %>%
  mutate(percent = n / sum(n)) %>%
  filter(corr_resp == 1) %>%
  select(ppt, percent_correct = percent)

mean_accuracy <- round(mean(percent_overall$percent_correct * 100), 2)
```

Mean accuracy across all participants was 93.55%.

Because we were only interested in correct responses for ordered sequences, we first removed all non-ordered trials (n = 16,800; 50%) and all incorrect responses (n = 1,027; 6.11%).

```
# remove non-ordered sequences
order_df <- order_df %>%
  filter(order == 1)

# remove incorrect responses
order_df <- order_df %>%
  filter(corr_resp == 1)
```

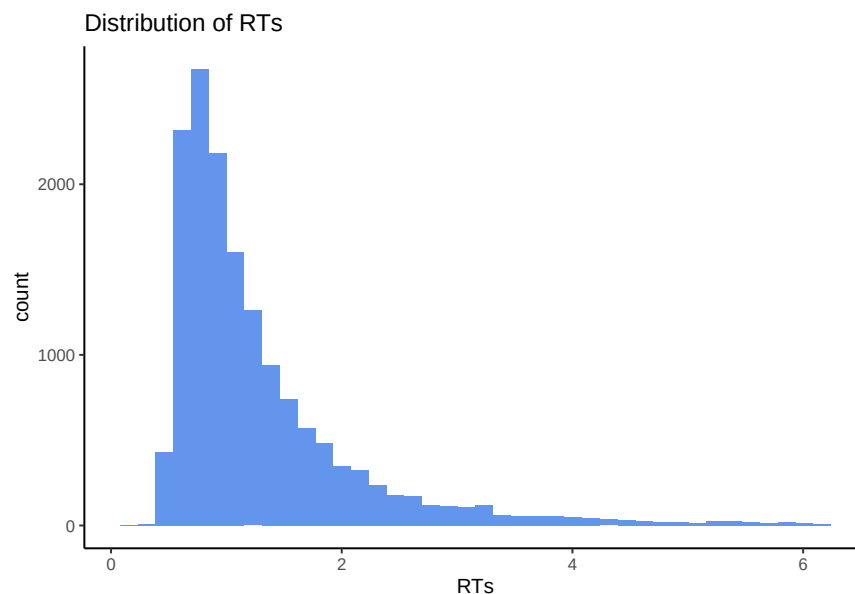
Consistent with previous studies (e.g., Dubinkina et al., 2021) we also removed responses that were shorter than 200 ms (n = 8; 0.05%) or longer than the overall mean response time plus three standard deviations (mean = 1432 ms, SD = 1598 ms) (n = 225; 1.43%) as these were considered unlikely to be genuine responses:

```
# remove RTs below 200 ms (consistent with Dubinkina21)
order_df <- order_df %>%
  filter(RT > 0.2)

# remove RTs greater than the mean + 3 SDs (consistent with Dubinkina21)
order_df <- order_df %>%
  filter(RT < mean(RT) + 3*(sd(RT)))
```

Following this trimming procedure, the distribution appeared somewhat positively skewed (skewness = 2.4) and leptokurtic (kurtosis = 13.17):

```
ggplot(order_df, aes(x=RT)) +
  geom_histogram(fill = '#6495ED', alpha = 1, bins = 40) +
  labs(title = "Distribution of RTs", x = "RTs") +
  theme_classic()
```



To account for this, unless otherwise specified, we considered the median response times for each participant. Means and standard deviations were then calculated for each sequence type:

```
# descriptives: means and sds for AC, DC, ANC, DNC
sum_df <- order_df %>%
  group_by(ppt, consec, direction) %>%
  summarise(median = median(RT) * 1000) %>%
  ungroup()

descriptives <- sum_df %>%
  group_by(consec, direction) %>%
  summarise(mean = round(mean(median),0), sd = round(sd(median),0)) %>%
  ungroup()
```

Table 4: Means and standard deviations for all direction/distance combinations (ms)

consec	direction	mean	sd
C	asc	972	293
C	dsc	1150	342
NC	asc	1071	343
NC	dsc	1254	476

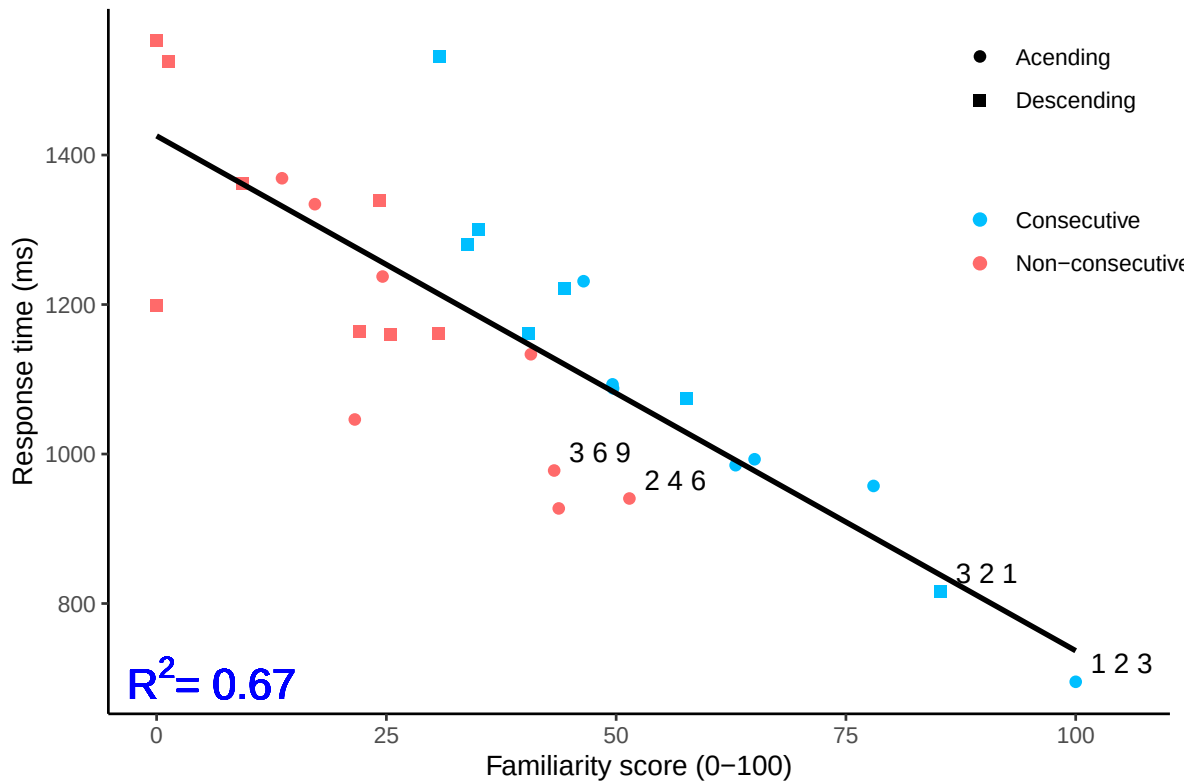
Relationship between response times and familiarity scores

Next, we considered the relationship between response times for each sequence on the order verification task and their associated familiarity scores generated by the comparative judgement task.

```
RTcorrFam <- cor.test(medRTs$meanRT, medRTs$scaled_score)
tidy(RTcorrFam)
```

```
## # A tibble: 1 x 8
##   estimate statistic      p.value parameter conf.low conf.high method alternative
##   <dbl>      <dbl>      <dbl>      <int>    <dbl>    <dbl> <chr>    <chr>
## 1   -0.819      -7.56    3.07e-8         28   -0.911   -0.651 Pears~ two.sided
```

We first conducted a Pearson's product-moment correlation which revealed a significant negative correlation, $r(28) = -0.82$, $p < .001$, suggesting higher familiarity scores were associated with faster response times:



However, although we observed a significant correlation between response times and familiarity scores, it could be argued this may simply be an artefact of ascending and consecutive sequences generally being more familiar than descending and non-consecutive sequences.

Consequently, rather than demonstrating a familiarity effect, this association may instead simply reflect the known findings that consecutive sequences are generally processed faster than non-consecutive sequences and ascending sequences are generally processed faster than descending sequences.

Therefore, we applied a linear mixed-effects modelling approach to establish whether familiarity scores explained any unique variance in response times beyond the effects of distance and direction alone.

Linear mixed-effects models have several advantages over more conventional approaches such as using ANOVAs or multiple regressions. For instance, not only are they compatible with both continuous and categorical explanatory variables, but they also allow taking into account variability across both participants and items simultaneously (Brown, 2021).

Therefore, using the lme4 package in R (Bates et al., 2015), we examined the fixed effects of distance (consecutive/non-consecutive), direction (ascending/descending), and familiarity (0–100) while accounting for the random effects of participants and sequences.

N.B., because mixed effects models use unaggregated data, median RTs are not used here.

```
# create lme4 models
mdl_null <- lmer(RT ~ 1 + (1 | ppt) + (1 | seq),
               data = df_unaggregated)

mdl_dist <- lmer(RT ~ distance + (1 | ppt) + (1 | seq),
               data = df_unaggregated)

mdl_dir <- lmer(RT ~ direction + (1 | ppt) + (1 | seq),
               data = df_unaggregated)

mdl_fam <- lmer(RT ~ scaled_score + (1 | ppt) + (1 | seq),
               data = df_unaggregated)

mdl_dist_dir <- lmer(RT ~ distance + direction + (1 | ppt) + (1 | seq),
                   data = df_unaggregated)

mdl_dist_dir_fam <- lmer(RT ~ distance + direction + scaled_score + (1 | ppt) + (1 | seq),
                       data = df_unaggregated)

# model comparisons
anova1 <- anova(mdl_null, mdl_dist_dir)
anova2 <- anova(mdl_dist_dir, mdl_dist_dir_fam)
anova3 <- anova(mdl_dist_dir_fam, mdl_fam)
```

We assessed model fit using likelihood-ratio tests. This revealed a model including distance and direction to better fit the data than a null model, $\chi^2(2) = 8.88$, $p = .012$:

Table 5: Model comparison

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mdl_null	4	249433.6	249464.3	-124712.8	249425.6	NA	NA	NA
mdl_dist_dir	6	249428.8	249474.7	-124708.4	249416.8	8.88	2	0.01

Furthermore, adding familiarity as a predictor further improved model fit, $\chi^2(1) = 40$, $p < .001$; thus, suggesting familiarity scores explained unique variance in response times beyond the effects of distance and direction alone:

Table 6: Model comparison

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mdl_dist_dir	6	249428.8	249474.7	-124708.4	249416.8	NA	NA	NA
mdl_dist_dir_fam	7	249390.8	249444.3	-124688.4	249376.8	40	1	0

Moreover, when comparing the model incorporating distance, direction, and familiarity with a model including only familiarity, we found that the former still explained more variance ($\chi^2(2) = 17.06$, $p < .001$).

However, the BIC for the model with only familiarity (BIC=249442) was smaller than the model with distance, direction, and familiarity (BIC=249444), suggesting that the model with only familiarity was the most parsimonious explanation of response times in the number order verification task:

Table 7: Model comparison

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mdl_fam	5	249403.8	249442.1	-124696.9	249393.8	NA	NA	NA
mdl_dist_dir_fam	7	249390.8	249444.3	-124688.4	249376.8	17.06	2	0

As an exploratory analysis, we also re-ran these models incorporating numerical size as a predictor. Here, we calculated the numerical size as the sum of the items within each sequence (e.g., 1-2-3 has a numerical size of 6, or an average numerical size of 2):

```
# create lme4 models including size as a predictor
mdl_size <- lmer(RT ~ size + (1 | ppt) + (1 | seq),
  data = df_unaggregated)

mdl_dist_dir_size <- lmer(RT ~ distance + direction + size + (1 | ppt) + (1 | seq),
  data = df_unaggregated)

mdl_dist_dir_size_fam <- lmer(RT ~ distance + direction + size + scaled_score + (1 | ppt) + (1 | seq),
  data = df_unaggregated)

# model comparisons
anova1 <- anova(mdl_null, mdl_dist_dir_size)
anova2 <- anova(mdl_null, mdl_dist_dir_size_fam)
anova3 <- anova(mdl_dist_dir_size_fam, mdl_fam)
```

We again assessed model fit using likelihood-ratio tests. This revealed a model including distance, direction, and numerical size to better fit the data than a null model, $\chi^2(3) = 34.47$, $p < .001$:

Table 8: Model comparison

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mdl_null	4	249433.6	249464.3	-124712.8	249425.6	NA	NA	NA
mdl_dist_dir_size	7	249405.2	249458.8	-124695.6	249391.2	34.47	3	0

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
--	------	-----	-----	--------	----------	-------	----	------------

Furthermore, adding familiarity as a predictor further improved model fit, $\chi^2(4) = 55.05$, $p < .001$; thus, suggesting familiarity scores explained unique variance in response times beyond the effects of distance, direction, and numerical size alone:

Table 9: Model comparison

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mdl_null	4	249433.6	249464.3	-124712.8	249425.6	NA	NA	NA
mdl_dist_dir_size_fam	8	249386.6	249447.8	-124685.3	249370.6	55.05	4	0

Moreover, when comparing the model incorporating distance, direction, numerical size, and familiarity with a model including only familiarity, we found that the former still explained more variance ($\chi^2(3) = 23.23$, $p < .001$).

However, as in the previous analysis, the BIC for the familiarity-only model (BIC=249442) was smaller than the model with distance, direction, numerical size, and familiarity (BIC=249448), suggesting that the model with only familiarity was the most parsimonious explanation of response times in the number order verification task:

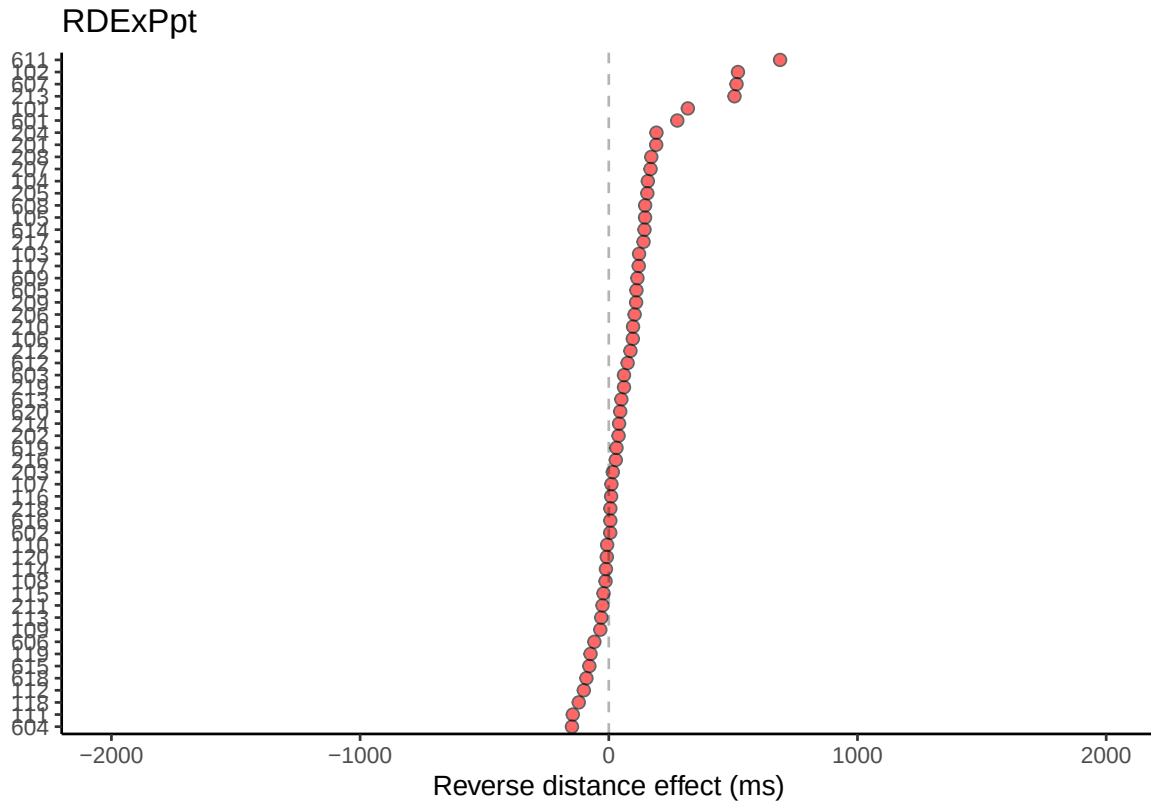
Table 10: Model comparison

	npar	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
mdl_fam	5	249403.8	249442.1	-124696.9	249393.8	NA	NA	NA
mdl_dist_dir_size_fam	8	249386.6	249447.8	-124685.3	249370.6	23.23	3	0

Reverse distance effect vs familiarity effect

Next, we considered the reverse distance effect and its stability. We first calculated skewness and kurtosis individually for consecutive, non-consecutive, high familiarity, and low familiarity sequences. Although skewness was within acceptable bounds (skewness = 0.64-0.79), all sequence types were leptokurtic (kurtosis = 5.38-5.66). Therefore, we also report non-parametric results for all analyses in this section.

We then checked whether the reverse distance effect across both ascending and descending sequences was present at the group level.



A paired-samples t-test revealed responses were indeed faster for consecutive sequences (mean = 1056 ms, SD = 312) than for non-consecutive sequences (mean = 1144 ms, SD = 373), $t(55) = 3.99$, $p < .001$ [V = 331, $p < .001$]:

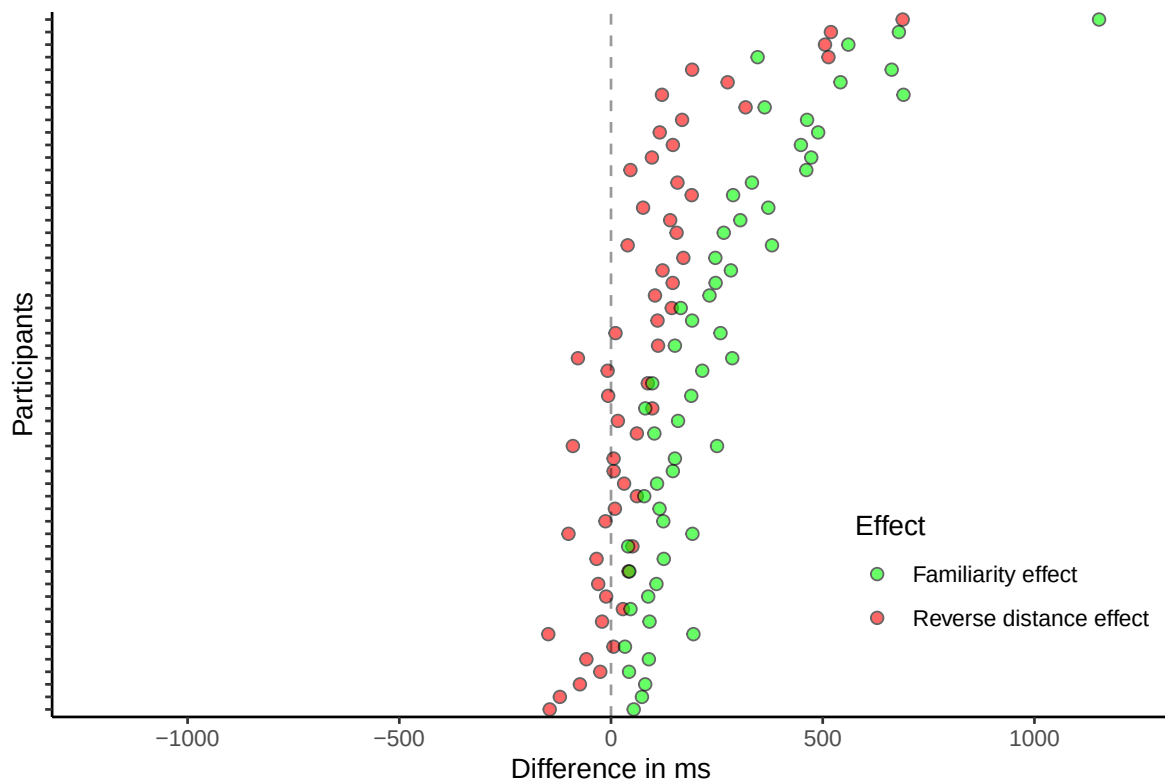
```
# t-test for RDE
RDE_ttest <- t.test(RDE_df$C, RDE_df$NC, paired = TRUE)
tidy(RDE_ttest)
```

```
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method   alternative
##   <dbl>      <dbl>   <dbl>      <dbl>   <dbl>      <dbl> <chr>      <chr>
## 1    -87.8     -3.99 0.000198         55    -132.     -43.7 Paired t~ two.sided
```

```
# Wilcoxon Signed rank test for RDE
RDE_WCtest <- wilcox.test(RDE_df$C, RDE_df$NC, paired = TRUE)
tidy(RDE_WCtest)
```

```
## # A tibble: 1 x 4
##   statistic p.value method                                alternative
##   <dbl>      <dbl> <chr>                                <chr>
## 1      331 0.000142 Wilcoxon signed rank test with continuity corr~ two.sided
```

Following this, we considered the stability of the reverse distance effect at the individual level. Therefore, we calculated the raw differences in response times between consecutive and non-consecutive sequences for each participant (mean = 88 ms, SD = 165):



We then considered the possibility of a familiarity effect. To calculate this, we used the familiarity scores from the comparative judgement task; the top half of sequences were coded as familiar, and the bottom half were coded as unfamiliar.

```
# t-test for familiarity effect
FE_ttest <- t.test(FE_df$low, FE_df$high, paired = TRUE)
tidy(FE_ttest)
```

```
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method alternative
##   <dbl>      <dbl>   <dbl>      <dbl>   <dbl>      <dbl> <chr>      <chr>
## 1    258.      9.07 1.64e-12         55    201.      315. Paired t~ two.sided
```

```
# Wilcoxon Signed rank test for FE
FE_WCtest <- wilcox.test(FE_df$low, FE_df$high, paired = TRUE)
tidy(FE_WCtest)
```

```
## # A tibble: 1 x 4
##   statistic p.value method alternative
##   <dbl>      <dbl> <chr>      <chr>
## 1    1596 7.76e-11 Wilcoxon signed rank test with continuity corr~ two.sided
```

A paired-samples t-test revealed response times were faster for familiar sequences (mean = 985 ms, SD = 285) than for unfamiliar sequences (mean = 1243 ms, SD = 428), $t(55) = 9.07$, $p < .001$ [$V = 1596$, $p < .001$]; thus, indicating a familiarity effect was present at the group level.

We then considered the stability of this familiarity effect at the individual level. In contrast to the reverse distance effect, we observed that every participant showed a familiarity effect larger than 0 ms (mean = 258 ms, SD = 213).

```
# difference between RDE and FE
RDE_FE1 <- RDE_df %>%
  left_join(FE_df, by = "ppt") %>%
  select(ppt, RDE, FE)

RDE_FE1 <- RDE_FE1 %>%
  mutate(difference = FE - RDE)

FEvsRDE <- t.test(RDE_FE1$FE, RDE_FE1$RDE, paired = TRUE)
tidy(FEvsRDE)

## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method alternative
##   <dbl>      <dbl>   <dbl>      <dbl>    <dbl>    <dbl> <chr>      <chr>
## 1      170.      8.83 3.99e-12         55      132.    209. Paired t~ two.sided

# Wilcoxon Signed rank test for RDE vs FE
FE_RDEtest <- wilcox.test(RDE_FE1$FE, RDE_FE1$RDE, paired = TRUE)
tidy(FE_RDEtest)

## # A tibble: 1 x 4
##   statistic p.value method alternative
##   <dbl>    <dbl> <chr>      <chr>
## 1      1555 6.79e-10 Wilcoxon signed rank test with continuity corr~ two.sided
```

In fact, not only were familiarity effects more pronounced than reverse distance effects at the group level ($t(55) = 8.83$, $p < .001$) [$V = 1555$, $p < .001$], but also at the individual level, 94.64% of participants (53 out of 56) displayed a familiarity effect larger than their reverse distance effect.

This effect remained even when only considering those who displayed a RDE:

```
RDEgreaterthan0only <- RDE_FE1 %>%
  filter(RDE > 0)

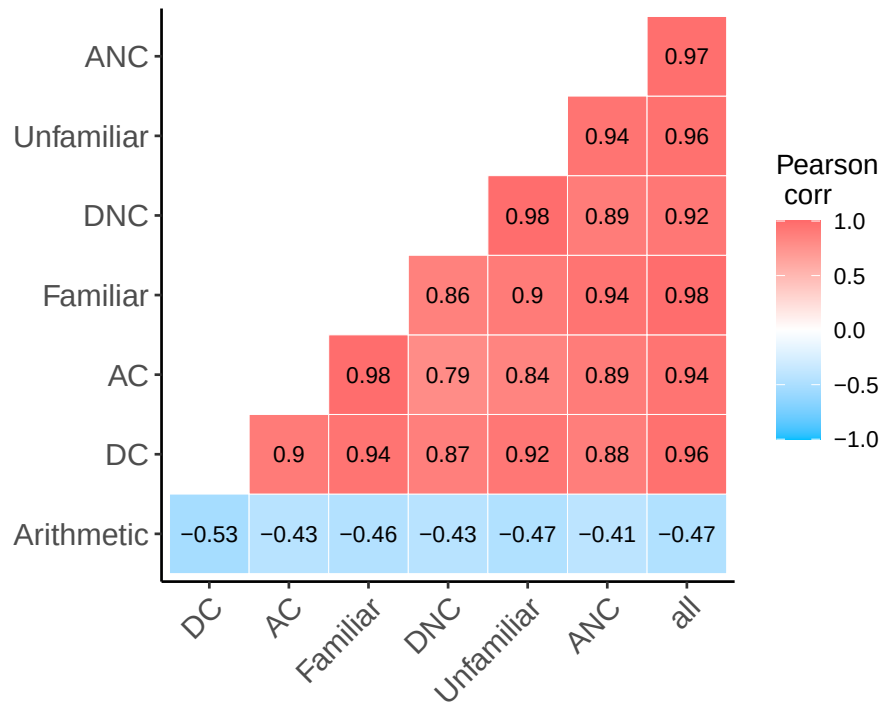
## # A tibble: 1 x 8
##   estimate statistic p.value parameter conf.low conf.high method alternative
##   <dbl>      <dbl>   <dbl>      <dbl>    <dbl>    <dbl> <chr>      <chr>
## 1      159.      6.28 0.000000211         39      108.    211. Paire~ two.sided

##
## Wilcoxon signed rank exact test
##
## data: RDEgreaterthan0only$FE and RDEgreaterthan0only$RDE
## V = 787, p-value = 5.844e-09
## alternative hypothesis: true location shift is not equal to 0
```

Notably, 28.57% of participants (16 out of 56) had a difference smaller than 0 ms, indicating they did not display a reverse distance effect but rather processed non-consecutive sequences faster than consecutive sequences (i.e., a standard distance effect).

Arithmetic fluency analysis

Finally, we explored the association between order verification and arithmetic fluency. Correlations were calculated between overall arithmetic scores and response times for six sequence types: ascending consecutive, ascending non-consecutive, descending consecutive, descending non-consecutive, familiar, and unfamiliar:



*All displayed correlations are statistically significant ($p < .05$)

Overall, there was a moderate negative correlation between response times and arithmetic scores, suggesting faster order verification is associated with higher arithmetic fluency.

This association was strongest for descending consecutive sequences and weakest for ascending non-consecutive sequences. However, there appeared no obvious pattern of associations between specific sequence types and arithmetic scores, with correlations ranging between -0.41 and -0.53.

Notably, response times for familiar (-0.46) and unfamiliar (-0.47) sequences showed similar associations with arithmetic fluency.