

1 A tutorial on Bayesian and Frequentist Event History Analyses for psychological
2 time-to-event data

3 Sven Panis¹ & Richard Ramsey¹

4 ¹ ETH Zürich

5 Author Note

6 Neural Control of Movement lab, Department of Health Sciences and Technology

7 (D-HEST).

8 The authors made the following contributions. Sven Panis: Conceptualization,
9 Writing - Original Draft Preparation, Writing - Review & Editing; Richard Ramsey:
10 Conceptualization, Writing - Review & Editing, Supervision.

11 Correspondence concerning this article should be addressed to Sven Panis, ETH
12 GLC, room G16.2, Gloriastrasse 37/39, 8006 Zürich. E-mail: sven.panis@hest.ethz.ch

13

Abstract

14 Time-to-event data such as response times, saccade latencies, and fixation durations are
15 ubiquitous in experimental psychology. The orthodox method for analysing such data –
16 comparing means with analysis-of-variance – is actually hiding a lot of information about
17 psychological effects, such as their onset time and duration, and how they evolve with
18 increasing waiting time. Such information can change key conclusions about psychological
19 processes and can be revealed by using distributional measures that portray the detailed
20 shape of time-to-event distributions.

21 Here we provide a set of tutorials on how to implement one particular distributional
22 method known as discrete-time event history analysis, a.k.a. hazard analysis, duration
23 analysis, failure-time analysis, survival analysis, and transition analysis. We illustrate how
24 one can calculate the descriptive statistics, and how one can implement Bayesian and
25 frequentist regression models, using the R packages tidyverse, brms, and lme4. The R code
26 is publicly available on Github and OSF, and can easily be adapted for other data sets. We
27 discuss possible link functions and prior distributions, how to manage inter-individual
28 differences, implications for experimental design, the advantages of a hazard analysis over
29 other distributional methods available in the literature, limitations, and extensions. Our
30 ultimate goal is to convince readers to start using hazard analysis more often when dealing
31 with time-to-event data.

32 *Keywords:* response times, event history analysis, Bayesian regression models

33 Word count: X

34 A tutorial on Bayesian and Frequentist Event History Analyses for psychological
35 time-to-event data

36 **Introduction**

37 **Means versus distributional shapes**

38 In experimental psychology, it is still standard practice to analyse response times
39 (RTs), saccade latencies, and fixation durations by calculating average performance across
40 a series of trials. However, differences in mean RT conceal when an experimental effect
41 starts, how long it lasts, how it evolves with increasing waiting time, and whether its onset
42 is time-locked to other events. Such information is useful not only for interpretation of the
43 effects, but also for cognitive psychophysiology and computational model selection (Panis,
44 Schmidt, Wolkersdorfer, & Schmidt, 2020).

45 As a simple illustration, Figure 1 shows three examples of how an observed difference
46 in mean response times (RTs) between two experimental conditions conceals differences in
47 the shapes of the underlying RT distributions. In each example, the mean RT is lower in
48 condition 2 compared to condition 1. However, the distributions in the first example show
49 that the effect starts around 200 ms and is gone by 600 ms. In the second example, the
50 effect starts around 400 ms and is gone by 800 ms. And in the third example, the effect
51 reverses around 550 ms.

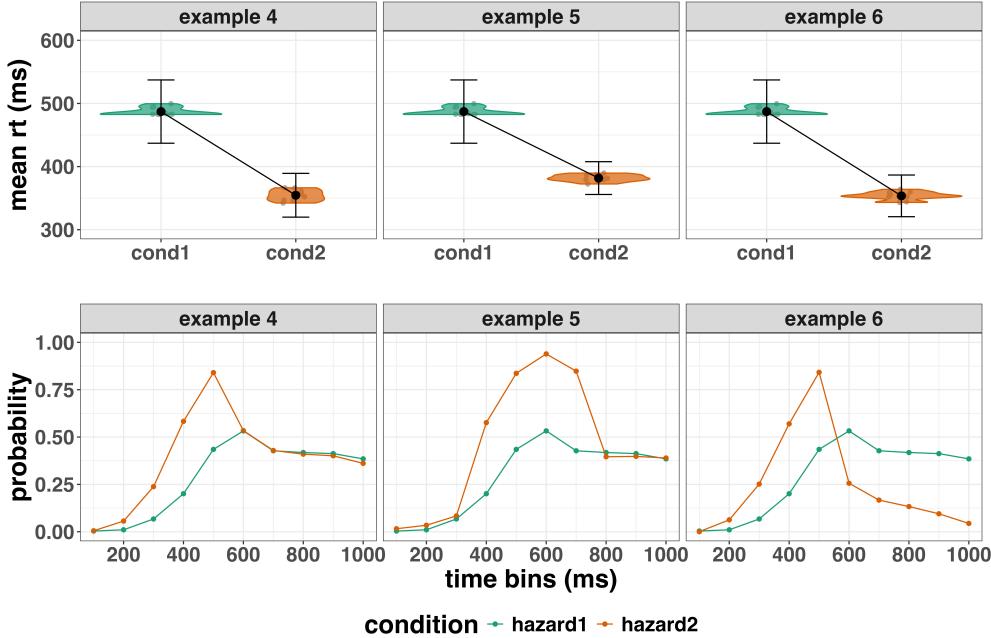


Figure 1. Means versus distributional shapes.

52 Outline of the paper

53 In this paper we focus on a distributional method known as discrete-time event
 54 history analysis, a.k.a. hazard analysis, duration analysis, failure-time analysis, survival
 55 analysis, and transition analysis. We first provide a brief overview of hazard analysis to
 56 orient the reader to the basic concepts that we will use throughout the paper. However,
 57 this will remain relatively short, and for detailed treatment, see Singer and Willett (2003),
 58 Allison (1982), and Allison (2010).

59 We then provide four different tutorials, each of which is written in R code and
 60 publicly available on Github and the Open Science Framework (OSF). The tutorials
 61 provide hands-on, concrete examples of key parts of the analytical process, so that others
 62 can apply the analyses to their own time-to-event data sets. In Tutorial 1 we illustrate how
 63 to calculate the descriptive statistics for a published RT data set when there is one
 64 independent variable. The descriptive statistics are plotted, and we comment on their

65 interpretation. In Tutorial 2 we illustrate how one can fit Bayesian hazard models to the
66 data. We discuss possible link functions, and plot the model-based effects of our predictors
67 of interest. In Tutorial 3 we illustrate how to fit hazard models in a frequentist framework.
68 Even though both frameworks generate similar parameter estimates, we note that model
69 convergence is often not obtained in the frequentist framework. In Tutorial 4 we illustrate
70 how to calculate the descriptive statistics when there are two independent variables.

71 **Overview of hazard analysis**

72 To apply event history analysis (EHA), one must be able to define the event of
73 interest (any qualitative change that can be situated in time, e.g., a button press, a saccade
74 onset, a fixation offset, etc.), time point zero (e.g., target stimulus onset, fixation onset),
75 and measure the passage of time between time point zero and event occurrence in discrete
76 or continuous time units. Both the definition of hazard and the type of models employed
77 depend on whether one is using continuous or discrete time units.

78 The shape of a distribution of waiting times can be described in multiple ways (Luce,
79 1991). Let RT be a continuous random variable denoting a particular person's response time
80 in a particular experimental condition. Because waiting times can only increase,
81 continuous-time EHA does not focus on the cumulative distribution function $F(t) = P(RT$
82 $\leq t)$ and its derivative, the probability density function $f(t) = F(t)'$, but on the survivor
83 function $S(t) = P(RT > t)$ and the hazard rate function $\lambda(t) = f(t)/S(t)$. The hazard rate
84 function gives you the instantaneous rate of event occurrence at time point t , given that
85 the event has not occurred yet.

86 Similarly, after dividing time in discrete, contiguous time bins indexed by t , let RT be
87 a discrete random variable denoting the rank of the time bin in which a particular person's
88 response occurs in a particular experimental condition. Discrete-time EHA focuses on the
89 discrete-time hazard function $h(t) = P(RT = t | RT \geq t)$ and the discrete-time survivor

90 function $S(t) = P(RT > t) = [1-h(t)].[1-h(t-1)].[1-h(t-2)] \dots [1-h(1)]$, and not on the
91 probability mass function $P(t) = P(RT = t) = h(t).S(t-1)$ nor the cumulative distribution
92 function $F(t) = 1-S(t)$. The discrete-time hazard probability gives you the probability that
93 the event occurs (sometime) in bin t , given that the event has not occurred yet in previous
94 bins. Unlike the discrete-time hazard function, which assesses the unique risk of event
95 occurrence associated with each time bin, the discrete-time survivor function cumulates
96 the bin-by-bin risks of event *non*occurrence to obtain the probability that the event occurs
97 after bin t .

98 For two-choice RT data, the discrete-time hazard function can be extended with the
99 discrete-time conditional accuracy function $ca(t) = P(\text{correct} \mid RT = t)$, which gives you
100 the probability that a response is correct given that it has been emitted in time bin t
101 (Allison, 2010; Kantowitz & Pachella, 2021; Wickelgren, 1977). This latter function is also
102 known as the micro-level speed-accuracy tradeoff function.

103 Here we focus on factorial within-subject designs in which a large number of
104 observations are made on a relatively small number of participants (small- N designs). This
105 approach emphasizes the precision and reproducibility of data patterns at the individual
106 participant level to increase the inferential validity of the design (Baker et al., 2021; Smith
107 & Little, 2018). In contrast to the large- N design that averages across many participants
108 without being able to scrutinize individual data patterns, small- N designs retain crucial
109 information about the data patterns of individual observers. This is of great advantage
110 whenever participants differ systematically in their strategies or in the time-courses of their
111 effects, so that blindly averaging them would lead to misleading data patterns. Indeed,
112 Smith and Little (2018) argue that, “if psychology is to be a mature quantitative science,
113 then its primary theoretical aim should be to investigate systematic functional
114 relationships as they are manifested at the individual participant level” (p. 2083). Note
115 that because statistical power derives both from the number of participants and from the
116 number of repeated measures per participant and condition, small- N designs can have

¹¹⁷ excellent power (Baker et al., 2021; Smith & Little, 2018).

We used R (Version 4.4.0; R Core Team, 2024)¹ for all reported analyses. Web links are printed in bold. The content of the tutorials is mainly based on Allison (2010), Singer and Willett (2003), McElreath (2018), Kurz (2023a), and Kurz (2023b).

121 Tutorial 1: Calculating descriptive statistics using a life table

To illustrate how to quickly set up life tables for calculating the descriptive statistics (functions of discrete time), we use a published data set on masked response priming from Panis and Schmidt (2016), available on **ResearchGate**. In their first experiment, Panis and Schmidt (2016) presented a double arrow for 94 ms that pointed left or right as the target stimulus with an onset at time point zero in each trial. Participants had to indicate

127 the direction in which the double arrow pointed using their corresponding index finger,
 128 within 800 ms after target onset. Response time and accuracy were recorded on each trial.
 129 Prime type (blank, congruent, incongruent) and mask type were manipulated. Here we
 130 focus on the subset of trials in which no mask was presented. The 13-ms prime stimulus
 131 was a double arrow with onset at -187 ms for the congruent (same direction as target) and
 132 incongruent (opposite direction as target) prime conditions.

133 After loading in the data file, one has to (a) supply required column names, and (b)
 134 specify the factor condition with the correct levels and labels. The required column names
 135 are as follows:

- 136 • “pid”, indicating unique participant IDs;
- 137 • “trial”, indicating each unique trial per participant;
- 138 • “condition”, a factor indicating the levels of the independent variable (1, 2, ...) and
 the corresponding labels;
- 140 • “rt”, indicating the response times in ms;
- 141 • “acc”, indicating the accuracies (1/0).

142 In the code of Tutorial 1, this is accomplished as follows.

```
data_wr <- read_csv("../Tutorial_1_descriptive_stats/data/DataExp1_6subjects_wrangled.csv")
colnames(data_wr) <- c("pid", "bl", "tr", "condition", "resp", "acc", "rt", "trial")
data_wr <- data_wr %>%
  mutate(condition = condition + 1, # original levels were 0, 1, 2.
        condition = factor(condition, levels=c(1,2,3), labels=c("blank", "congruent", "incongruent")))
```

143 To set up the life tables and plots of the discrete-time functions $h(t)$, $S(t)$ and $ca(t)$
 144 using functional programming, one has to nest the data within participants using the
 145 `group_nest()` function, and supply a user-defined censoring time and bin width to our
 146 function “`censor()`”, as follows.

```

data_nested <- data_wr %>% group_nest(pid)

data_final <- data_nested %>%
  mutate(censored = map(data, censor, 600, 40)) %>% # ! user input: censoring time, and bin width
  mutate(ptb_data = map(censored, ptb)) %>% # create person-trial-bin dataset
  mutate(lifetable = map(ptb_data, setup_lt)) %>% # create life tables without ca(t)
  mutate(condacc = map(censored, calc_ca)) %>% # calculate ca(t)
  mutate(lifetable_ca = map2(lifetable, condacc, join_lt_ca)) %>% # create life tables with ca(t)
  mutate(plot = map2(.x = lifetable_ca, .y = pid, plot_eha, 1)) # create plots

```

147 Note that the censoring time should be a multiple of the bin width (both in ms). The
 148 censoring time should be a time point after which no informative responses are expected
 149 anymore. In experiments that implement a response deadline in each trial the censoring
 150 time can equal that deadline time point. Trials with a RT larger than the censoring time,
 151 or trials in which no response is emitted during the data collection period, are treated as
 152 right-censored observations in EHA. In other words, these trials are not discarded, because
 153 they contain the information that the event did not occur before the censoring time.
 154 Removing such trials before calculating the mean event time can introduce a sampling bias.
 155 The person-trial-bin oriented dataset has one row for each time bin of each trial that is at
 156 risk for event occurrence. The variable “event” in the person-trial-bin oriented data set
 157 indicates whether a response occurs (1) or not (0) for each bin. When creating the plots
 158 using our function `plot_eha()`, some warning messages will likely be generated, like these:

- 159 • Removed 2 rows containing missing values or values outside the scale range
 160 `(geom_line())`.
- 161 • Removed 2 rows containing missing values or values outside the scale range
 162 `(geom_point())`.
- 163 • Removed 2 rows containing missing values or values outside the scale range
 164 `(geom_segment())`.

165 The warning messages are generated because some bins have no hazard and $ca(t)$

166 estimates, and no error bars. They can thus safely be ignored. One can now inspect

167 different aspects, including the life table for a particular condition of a particular subject,
168 and a plot of the different functions for a particular participant.

169 Table 1 shows the life table for condition “blank” (no prime stimulus presented) -
170 compare to Figure 1. A life table includes for each time bin, the risk set (number of trials
171 that are event-free at the start of the bin), the number of observed events, and the
172 estimates of $h(t)$, $S(t)$, $ca(t)$ and their estimated standard errors (se). At time point zero,
173 no events can occur and therefore $h(t)$ and $ca(t)$ are undefined.

174 Figure 2 displays the discrete-time hazard, survivor, and conditional accuracy
175 functions for each prime condition for participant 6. By using discrete-time $h(t)$ functions
176 of event occurrence - in combination with $ca(t)$ functions for two-choice tasks - one can
177 provide an unbiased, time-varying, and probabilistic description of the latency and
178 accuracy of responses based on all trials of any data set.

179 For example, for participant 6, the estimated hazard values in bin (240,280] are 0.03,
180 0.17, and 0.20 for the blank, congruent, and incongruent prime conditions, respectively. In
181 other words, when the waiting time has increased until 240 ms after target onset, then the
182 conditional probability of response occurrence in the next 40 ms is more than five times
183 larger for both prime-present conditions, compared to the blank prime condition.

184 Furthermore, the estimated conditional accuracy values in bin (240,280] are 0.29, 1,
185 and 0 for the blank, congruent, and incongruent prime conditions, respectively. In other
186 words, if a response is emitted in bin (240,280], then the probability that it is correct is
187 estimated to be 0.29, 1, and 0 for the blank, congruent, and incongruent prime conditions,
188 respectively.

Table 1

The life table for the blank prime condition of participant 6.

bin	risk_set	events	hazard	se_haz	survival	se_surv	ca	se_ca
0.00	220.00	NA	NA	NA	1.00	0.00	NA	NA
40.00	220.00	0.00	0.00	0.00	1.00	0.00	NA	NA
80.00	220.00	0.00	0.00	0.00	1.00	0.00	NA	NA
120.00	220.00	0.00	0.00	0.00	1.00	0.00	NA	NA
160.00	220.00	0.00	0.00	0.00	1.00	0.00	NA	NA
200.00	220.00	0.00	0.00	0.00	1.00	0.00	NA	NA
240.00	220.00	0.00	0.00	0.00	1.00	0.00	NA	NA
280.00	220.00	7.00	0.03	0.01	0.97	0.01	0.29	0.17
320.00	213.00	13.00	0.06	0.02	0.91	0.02	0.77	0.12
360.00	200.00	26.00	0.13	0.02	0.79	0.03	0.92	0.05
400.00	174.00	40.00	0.23	0.03	0.61	0.03	1.00	0.00
440.00	134.00	48.00	0.36	0.04	0.39	0.03	0.98	0.02
480.00	86.00	37.00	0.43	0.05	0.22	0.03	1.00	0.00
520.00	49.00	32.00	0.65	0.07	0.08	0.02	1.00	0.00
560.00	17.00	9.00	0.53	0.12	0.04	0.01	1.00	0.00
600.00	8.00	4.00	0.50	0.18	0.02	0.01	1.00	0.00

Note. The column named “bin” indicates the endpoint of each time bin (in ms), and includes time point zero. For example the first bin is (0,40] with the starting point excluded and the endpoint included. se = standard error. ca = conditional accuracy.

Subject 6

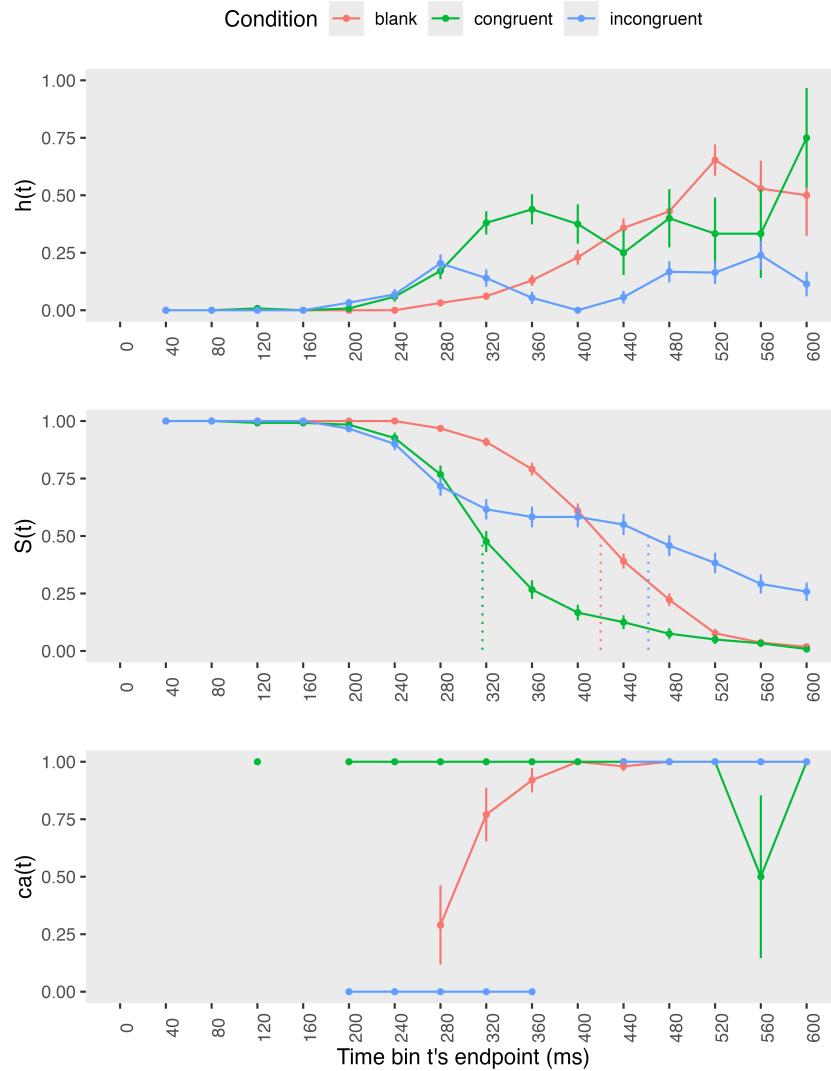


Figure 2. Estimated discrete-time hazard, survivor, and conditional accuracy functions for participant 6, as a function of the passage of discrete waiting time.

189 However, when the waiting time has increased until 400 ms after target onset, then
 190 the conditional probability of response occurrence in the next 40 ms is estimated to be
 191 0.36, 0.25, and 0.06 for the blank, congruent, and incongruent prime conditions,
 192 respectively. And when a response does occur in bin (400,440], then the probability that it
 193 is correct is estimated to be 0.98, 1, and 1 for the blank, congruent, and incongruent prime

194 conditions, respectively.

195 These results show that this participant is initially responding to the prime even
196 though (s)he was instructed to only respond to the target, that response competition
197 emerges in the incongruent prime condition around 300 ms, and that only later response
198 are fully controlled by the target stimulus. Qualitatively similar results were obtained for
199 the other five participants. These results go against the (often implicit) assumption that all
200 observed responses are primed responses to the target stimulus.

201 Also, in their second Experiment, Panis and Schmidt (2016) showed that the negative
202 compatibility effect in the mask-present conditions (see Tutorial 4) is time-locked to mask
203 onset. This example shows that a simple difference between two means fails to reveal the
204 dynamic behavior people display in many experimental paradigms (Panis, 2020; Panis,
205 Moran, Wolkersdorfer, & Schmidt, 2020; Panis & Schmidt, 2022; Panis, Torfs, Gillebert,
206 Wagemans, & Humphreys, 2017; Panis & Wagemans, 2009; Schmidt, Panis, Wolkersdorfer,
207 & Vorberg, 2022). In other words, statistically controlling for the passage of time during
208 data analysis is equally important as experimental control during the design of an
209 experiment, to better understand human behavior in experimental paradigms. As we will
210 show in Tutorials 2 and 3, statistical models for $h(t)$ can be implemented as generalized
211 linear mixed regression models predicting event occurrence (1/0) in each bin of a selected
212 time range.

213 **Tutorial 2: Fitting Bayesian hazard models**

214 In this second tutorial we illustrate how to fit Bayesian hazard regression models to
215 the masked response priming data set used in the first tutorial. Fitting (Bayesian or
216 non-Bayesian) regression models to the data is important when you want to study how the
217 shape of the hazard function depends on various predictors (Singer & Willett, 2003).

218 There are several analytic decisions one has to make when fitting a hazard model.

219 First, one has to select an analysis time window, i.e., a contiguous set of bins for which
 220 there is enough data for each participant. Second, given that the dependent variable is
 221 binary, one has to select a link function (see Figure 3).

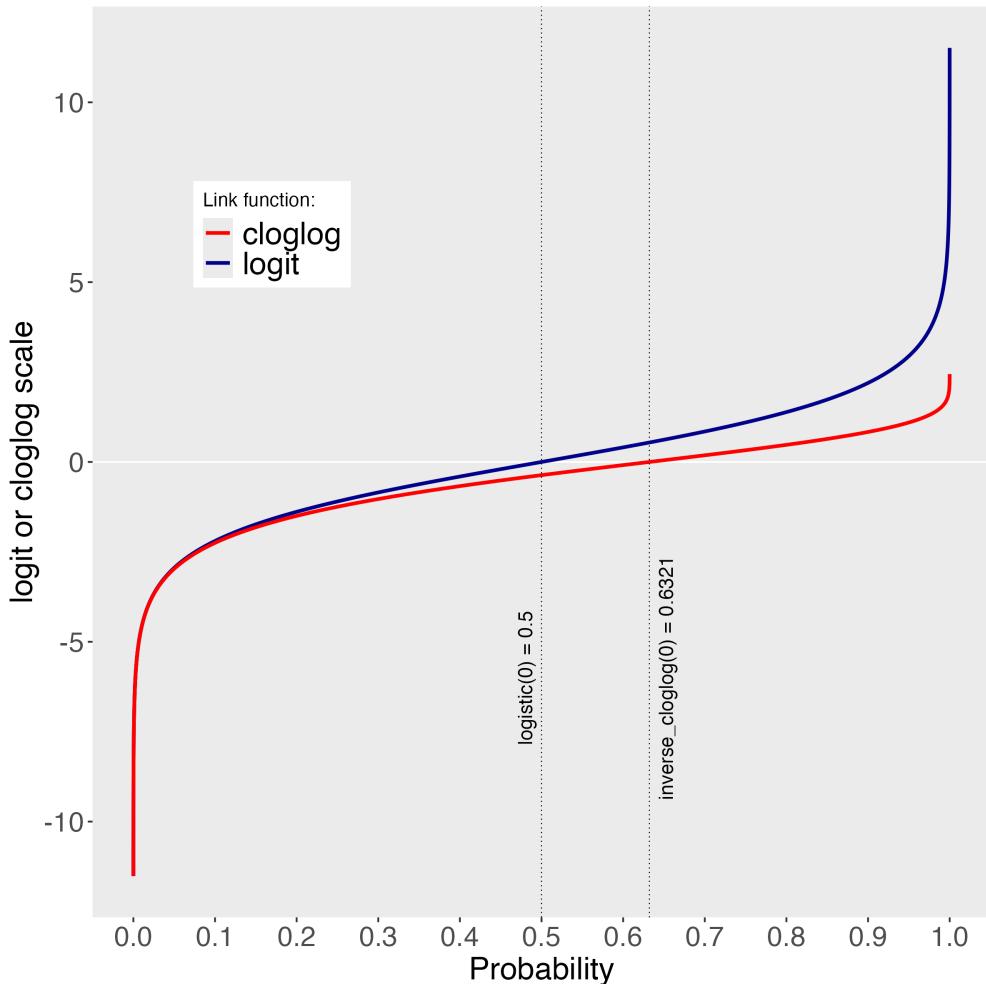


Figure 3. The logit and cloglog link functions.

222 The symmetric logit link function transforms a (hazard) probability into the log of
 223 the odds ratio. The asymmetric complementary log-log (cloglog) link function transforms
 224 hazard into the logarithm of the negated logarithm of the probability of event
 225 nonoccurrence. An important difference between these two link functions is that cloglog
 226 provides a discrete-time hazard model that has a built-in proportional hazards assumption,
 227 while logit provides a proportional odds assumption. The cloglog link is preferred over the

228 logit link when events can occur in principle at any time point within a bin, which is the
 229 case for RT data (Singer & Willett, 2003). Third, one has to choose a specification of the
 230 effect of discrete TIME (i.e., the time bin index t). One can choose a general specification
 231 (one intercept per bin) or a functional specification, such as a polynomial one (compare
 232 model 1 with models 2, 3, and 4 below).

233 An example (single-level) discrete-time hazard model with three predictors (TIME,
 234 X₁, X₂), the cloglog link function, and a third-order polynomial specification for TIME can
 235 be written as follows:

$$236 \quad \text{cloglog}[h(t)] = \ln(-\ln[1-h(t)]) = [\alpha_1 \text{ONE} + \alpha_2(\text{TIME}-9) + \alpha_3(\text{TIME}-9)^2] + [\beta_1 X_1 + \\ 237 \quad \beta_2 X_2 + \beta_3 X_2(\text{TIME}-9)].$$

238 The main predictor variable TIME is the time bin index t that is centered on value 9
 239 in this example. The first set of terms within brackets, the alpha parameters multiplied by
 240 their polynomial specifications of (centered) time, represents the shape of the baseline
 241 cloglog-hazard function (i.e., when all predictors X_i take on a value of zero). The second set
 242 of terms (the beta parameters) represents the vertical shift in the baseline cloglog-hazard
 243 for a 1 unit increase in the respective predictor variable. Predictors can be discrete,
 244 continuous, and time-varying or time-invariant. For example, the effect of a 1 unit increase
 245 in X₁ is to vertically shift the whole baseline cloglog-hazard function by β_1 cloglog-hazard
 246 units. However, if the predictor interacts linearly with TIME (see X₂ in the example), then
 247 the effect of a 1 unit increase in X₂ is to vertically shift the predicted cloglog-hazard in bin
 248 9 by β_2 cloglog-hazard units (when TIME-9 = 0), in bin 10 by $\beta_2 + \beta_3$ cloglog-hazard
 249 units (when TIME-9 = 1), and so forth. To interpret the effects of a predictor, its β
 250 parameter is exponentiated, resulting in a hazard ratio (due to the use of the cloglog link).
 251 When using the logit link, exponentiating a β parameter results in an odds ratio.

252 An example (single-level) discrete-time hazard model with a general specification for
 253 TIME (separate intercepts for each of six bins, where D1 to D6 are binary variables

254 identifying each bin) and a single predictor (X_1) can be written as follows:

$$255 \quad \text{cloglog}[h(t)] = \ln(-\ln[1-h(t)]) = [\alpha_1 D1 + \alpha_2 D2 + \alpha_3 D3 + \alpha_4 D4 + \alpha_5 D5 + \alpha_6 D6] + \\ 256 \quad [\beta_1 X_1].$$

257 In the case of a large- N design without repeated measurements, the parameters of a
 258 discrete-time hazard model can be estimated using standard logistic regression software
 259 after expanding the typical person-trial-oriented data set into a person-trial-bin-oriented
 260 data set (Allison, 2010). When there is clustering in the data, as in the case of a small- N
 261 design with repeated measurements, the parameters of a discrete-time hazard model can be
 262 estimated using population-averaged methods (e.g., Generalized Estimating Equations),
 263 and Bayesian or frequentist generalized linear mixed models (Allison, 2010).

264 In general, there are three assumptions one can make or relax when adding
 265 experimental predictor variables: The linearity assumption for continuous predictors (the
 266 effect of a 1 unit change is the same anywhere on the scale), the additivity assumption
 267 (predictors do not interact), and the proportionality assumption (predictors do not interact
 268 with TIME).

269 In this tutorial we will fit four Bayesian multilevel models (i.e., generalized linear
 270 mixed models) to the person-trial-bin oriented data set that we created in Tutorial 1. We
 271 select the analysis range (200,600] and the cloglog link. The data is prepared as follows.

```
# load person-trial-bin oriented data set
ptb_data <- read_csv("../Tutorial_1_descriptive_stats/data/inputfile_hazard_modeling.csv")

# select analysis time range: (200,600] with 10 bins (time bin ranks 6 to 15)
ptb_data <- ptb_data %>% filter(period > 5)

# create factor condition, with "blank" as the reference level
ptb_data <- ptb_data %>% mutate(condition = factor(condition, labels = c("blank", "congruent", "incongruent")))

# center TIME (period) on bin 9, and trial on trial 1000 and rescale; Add dummy variables for each bin.
ptb_data <- ptb_data %>%
  mutate(period_9 = period - 9,
```

```

trial_c = (trial - 1000)/1000,
d6  = if_else(period == 6, 1, 0),
d7  = if_else(period == 7, 1, 0),
d8  = if_else(period == 8, 1, 0),
d9  = if_else(period == 9, 1, 0),
d10 = if_else(period == 10, 1, 0),
d11 = if_else(period == 11, 1, 0),
d12 = if_else(period == 12, 1, 0),
d13 = if_else(period == 13, 1, 0),
d14 = if_else(period == 14, 1, 0),
d15 = if_else(period == 15, 1, 0))

```

272 Prior distributions

273 To get the posterior distribution of each parameters given the data, we need to
274 specify a prior distribution for each parameter. The middle column of Figure 4 shows seven
275 examples of prior distributions on the logit and/or cloglog scales.

276 While a normal distribution with relatively large variance is often used as a weakly
277 informative prior for continuous dependent variables, rows A and B in Figure 3 show that
278 specifying such distributions on the logit and cloglog scales leads to rather informative
279 distributions on the original probability (i.e., discrete-time hazard) scale, as most mass is
280 pushed to probabilities of 0 and 1.

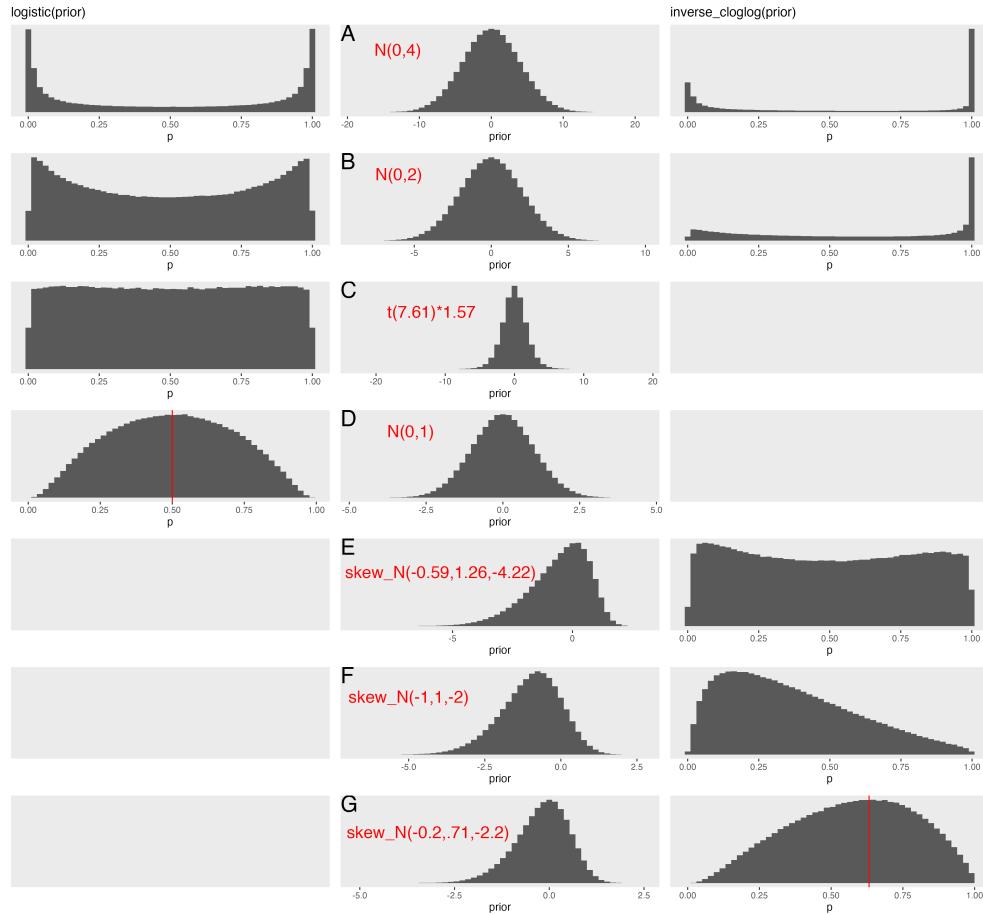


Figure 4. Prior distributions on the logit and/or cloglog scales (middle column), and their implications on the probability scale after applying the inverse-logit (or logistic) transformation (left column), and the inverse-cloglog transformation (right column).

281 To gain a sense of what prior *logit* values would approximate a uniform distribution

282 on the probability scale, Kurz (2023a) simulated a large number of draws from the

283 Uniform(0,1) distribution, converted those draws to the log-odds metric, and fitted a

284 Student's t distribution. Row C in Figure 4 shows that using a t-distribution with 7.61

285 degrees of freedom and a scale parameter of 1.57 as a prior on the logit scale, approximates

286 a uniform distribution on the probability scale. According to Kurz (2023a), such a prior

287 might be a good prior for the intercept(s) in a logit-hazard model, while the N(0,1) prior in

288 row D might be a good prior for the non-intercept parameters in a logit-hazard model, as it

289 gently regularizes p towards .5 (i.e., a zero effect on the logit scale).

290 To gain a sense of what prior *cloglog* values would approximate a uniform distribution
 291 on the hazard probability scale, we followed Kurz's approach and simulated a large number
 292 of draws from the Uniform(0,1) distribution, converted them to the cloglog metric, and
 293 fitted a skew-normal model (due to the asymmetry of the cloglog link function). Row E
 294 shows that using a skew-normal distribution with a mean of -0.59, a standard deviation of
 295 1.26, and a skewness of -4.22 as a prior on the cloglog scale, approximates a uniform
 296 distribution on the probability scale. However, because hazard values below .5 are more
 297 likely in RT studies, using a skew-normal distribution with a mean of -1, a standard
 298 deviation of 1, and a skewness of -2 as a prior on the cloglog scale (row F), might be a good
 299 weakly informative prior for the intercept(s) in a cloglog-hazard model. A skew-normal
 300 distribution with a mean of -0.2, a standard deviation of 0.71, and a skewness of -2.2 might
 301 be a good weakly informative prior for the non-intercept parameters in a cloglog-hazard
 302 model as it gently regularizes p towards .6321 (i.e., a zero effect on the cloglog scale).

303 **Model 1: A general specification of TIME, and main effects of congruency and
 304 trial number**

305 For the first model, we use a general specification of TIME (i.e., one intercept per
 306 time bin) for the baseline condition (blank prime), and assume that the effects of
 307 prime-target congruency and trial number are proportional and additive, and that the
 308 effect of trial number is linear. Before we fit model 1, we remove unnecessary columns from
 309 the data, and specify our priors. In the code of Tutorial 2, this is accomplished as follows.

```
# remove unnecessary columns before fitting a model
M1_data <- ptb_data %>% select(-c(bl,tr,trial,period, period_9,d9))

# Specify priors
priors_M1 <- c(
  set_prior("skew_normal(-0.2,0.71,-2.2)", class = "b"),
```

```

set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d6"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d7"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d8"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d10"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d11"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d12"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d13"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d14"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "d15"),
set_prior("skew_normal(-1,1,-2)", class = "b", coef = "Intercept"),
set_prior("normal(0, 1)", class = "sd"),
set_prior("lkj(2)", class = "cor")

)

```

310 We can now estimate our first Bayesian regression model, as follows.

```

plan(multicore)

model_M1 <-
  brm(data = M1_data,
       family = binomial(link="cloglog"),
       event | trials(1) ~ 0 + d6 + d7 + d8 + Intercept + d10 + d11 + d12 + d13 + d14 + d15 +
               condition + trial_c +
               (d6 + d7 + d8 + 1 + d10 + d11 + d12 + d13 + d14 + d15 + condition + trial_c | pid),
       prior = priors_M1,
       chains = 4, cores = 4, iter = 3000, warmup = 1000,
       control = list(adapt_delta = 0.999, step_size = 0.04, max_treedepth = 12),
       seed = 12, init = "O",
       file = "Tutorial_2_Bayesian/models/model_M1")

```

311 Estimating model M1 took about 70 minutes on a MacBook Pro (Sonoma 14.6.1 OS,
 312 18GB Memory, M3 Pro Chip).

313 Model 2: A polynomial specification of TIME, and main effects of congruency
 314 and trial number

315 For the second model, we use a third-order polynomial specification of TIME for the
 316 baseline condition (blank prime), and again assume that the effects of prime-target
 317 congruency and trial number are proportional and additive, and that the effect of trial
 318 number is linear. We first remove unnecessary columns and specify our priors.

```
# remove unnecessary columns
M2_data <- ptb_data %>% select(-c(bl,tr,trial,period, d6, d7, d8, d9, d10, d11, d12, d13, d14, d15))

# Specify priors
priors_M2 <- c(
  set_prior("skew_normal(-0.2,0.71,-2.2)", class = "b"),
  set_prior("skew_normal(-1,1,-2)", class = "b", coef = "Intercept"),
  set_prior("normal(0, 1)", class = "sd"),
  set_prior("lkj(2)", class = "cor")
)
```

319 Now we can fit model 2.

```
plan(multicore)

model_M2 <-
  brm(data = M2_data,
    family = binomial(link="cloglog"),
    event | trials(1) ~ 0 + Intercept + period_9 + I(period_9^2) + I(period_9^3) +
      condition + trial_c +
      (1 + period_9 + I(period_9^2) + I(period_9^3) +
      condition + trial_c | pid),
    prior = priors_M2,
    chains = 4, cores = 4, iter = 3000, warmup = 1000,
    control = list(adapt_delta = 0.999, step_size = 0.04, max_treedepth = 12),
    seed = 12, init = "0",
    file = "Tutorial_2_Bayesian/models/model_M2")
```

320 Estimating model M2 took about 144 minutes.

321 Model 3: A polynomial specification of TIME, and relaxing the proportionality
 322 assumption

323 For the third model, we use a third-order polynomial specification of TIME for the
 324 baseline condition (blank prime), and relax the proportionality assumption for the
 325 predictor variables congruency (variable “condition”) and trial number (variable “trial_c”).
 326 We use the same data set and priors as for model 2.

```
M3_data <- M2_data
priors_M3 <- priors_M2
plan(multicore)

model_M3 <-
  brm(data = M3_data,
       family = binomial(link="cloglog"),
       event | trials(1) ~ 0 + Intercept + # Note that duplicate terms in the model formula are ignored
              condition*period_9 +
              condition*I(period_9^2) +
              condition*I(period_9^3) +
              trial_c*period_9 +
              trial_c*I(period_9^2) +
              trial_c*I(period_9^3) +
              (1 + condition*period_9 +
              condition*I(period_9^2) +
              condition*I(period_9^3) +
              trial_c*period_9 +
              trial_c*I(period_9^2) +
              trial_c*I(period_9^3) | pid),
       prior = priors_M3,
       chains = 4, cores = 4, iter = 3000, warmup = 1000,
       control = list(adapt_delta = 0.999, step_size = 0.04, max_treedepth = 12),
       seed = 12, init = "0",
       file = "Tutorial_2_Bayesian/models/model_M3")
```

327 Estimating model M3 took about 268 minutes.

328 Model 4: A polynomial specification of TIME, and relaxing all three
 329 assumptions

330 Based on previous work (Panis, 2020; Panis, Moran, et al., 2020; Panis & Schmidt,
 331 2022; Panis et al., 2017; Panis & Wagemans, 2009), we relax all three assumptions in
 332 model 4. We use the same data set and priors as for model 2.

```
M4_data <- M2_data
priors_M4 <- priors_M2
plan(multicore)

model_M4 <-
  brm(data = M4_data,
    family = binomial(link="cloglog"),
    event | trials(1) ~ 0 + Intercept + # Note that duplicate terms in the model formula are ignored
          condition*period_9*trial_c +
          condition*period_9*I(trial_c^2) +
          condition*I(period_9^2)*trial_c +
          condition*I(period_9^2)*I(trial_c^2) +
          condition*I(period_9^3) +
          trial_c*I(period_9^3) +
          (1 + condition*period_9*trial_c +
          condition*period_9*I(trial_c^2) +
          condition*I(period_9^2)*trial_c +
          condition*I(period_9^2)*I(trial_c^2) +
          condition*I(period_9^3) +
          trial_c*I(period_9^3) | pid),
    prior = priors_M4,
    chains = 4, cores = 4, iter = 3000, warmup = 1000,
    control = list(adapt_delta = 0.999, step_size = 0.04, max_treedepth = 12),
    seed = 12, init = "0",
    file = "Tutorial_2_Bayesian/models/model_M4")
```

333 Estimating model M4 took about 8 hours.

334 **Compare the models.**

335 We can compare the four models using the Widely Applicable Information Criterion
 336 (WAIC) and Leave-One-Out (LOO) cross-validation, and look at model weights (Kurz,
 337 2023a; McElreath, 2018).

```
model_M1 <- readRDS("../Tutorial_2_Bayesian/models/model_M1.rds")
model_M2 <- readRDS("../Tutorial_2_Bayesian/models/model_M2.rds")
model_M3 <- readRDS("../Tutorial_2_Bayesian/models/model_M3.rds")
model_M4 <- readRDS("../Tutorial_2_Bayesian/models/model_M4.rds")

#model_M1 <- add_criterion(model_M1, c("loo", "waic"))
#model_M2 <- add_criterion(model_M2, c("loo", "waic"))
#model_M3 <- add_criterion(model_M3, c("loo", "waic"))
#model_M4 <- add_criterion(model_M4, c("loo", "waic"))

loo_compare(model_M1, model_M2, model_M3, model_M4, criterion = "loo") %>% print(simplify = F)
```

```
338 ##          elpd_diff se_diff elpd_loo se_elpd_loo p_loo    se_p_loo looic
339 ## model_M4      0.0     0.0 -5094.4    62.9     127.6     4.2 10188.9
340 ## model_M3   -21.0    10.1 -5115.4    62.5     77.3     2.5 10230.8
341 ## model_M1  -254.9    24.4 -5349.3    64.8     72.7     1.7 10698.6
342 ## model_M2  -256.8    23.6 -5351.2    64.8     39.7     1.0 10702.5
343 ##          se_looic
344 ## model_M4    125.8
345 ## model_M3    125.1
346 ## model_M1    129.6
347 ## model_M2    129.7
```

```
loo_compare(model_M1, model_M2, model_M3, model_M4, criterion = "waic") %>% print(simplify = F)
```

```
348 ##          elpd_diff se_diff elpd_waic se_elpd_waic p_waic    se_p_waic waic
349 ## model_M4      0.0     0.0 -5092.8    62.9     125.9     4.0 10185.5
350 ## model_M3   -22.1    10.0 -5114.9    62.5     76.8     2.4 10229.7
351 ## model_M1  -256.3    24.3 -5349.1    64.8     72.5     1.6 10698.2
352 ## model_M2  -258.4    23.6 -5351.2    64.8     39.6     1.0 10702.4
353 ##          se_waic
354 ## model_M4    125.8
355 ## model_M3    125.1
356 ## model_M1    129.6
357 ## model_M2    129.7
```

```
# model weights  
model_weights(model_M1, model_M2, model_M3, model_M4, weights = "loo") %>% round(digits = 3)
```

```
358 ## model_M1 model_M2 model_M3 model_M4  
359 ##      0      0      0      1
```

```
model_weights(model_M1, model_M2, model_M3, model_M4, weights = "waic") %>% round(digits = 3)
```

```
360 ## model_M1 model_M2 model_M3 model_M4  
361 ##      0      0      0      1
```

362 Clearly, both weighting schemes prefer model M4.

363 **Plot congruency effects and subject-specific fits for model M4.**

364 Figure 5 shows the effects of congruent and incongruent primes relative to neutral
365 primes, for each time bin in trial number 1000 for the selected model.

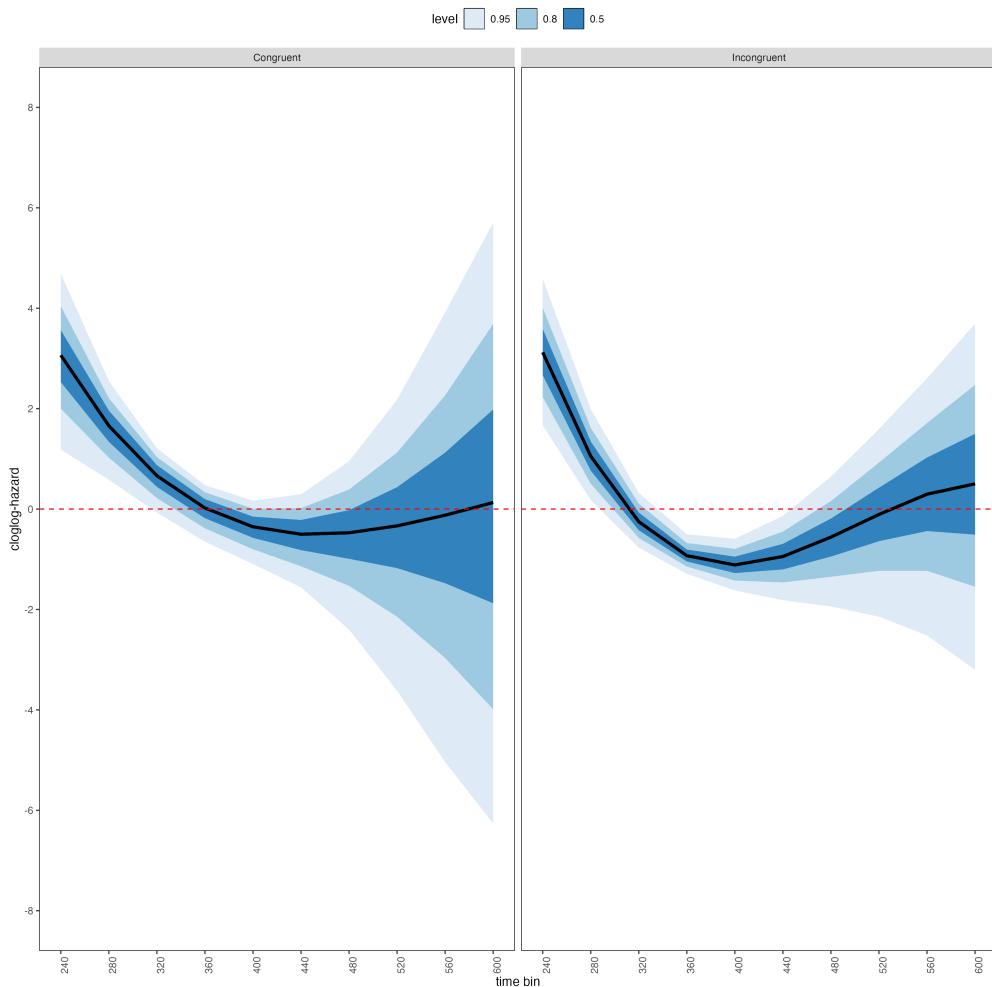


Figure 5. 50/80/95 percentile intervals of the draws from the posterior distributions representing the effect of congruent and incongruent primes relative to neutral primes in trial number 1000.

366 Figure 6 shows the model-based hazard functions for each prime type for participant
 367 6, in trial 500, 1000, and 1500.

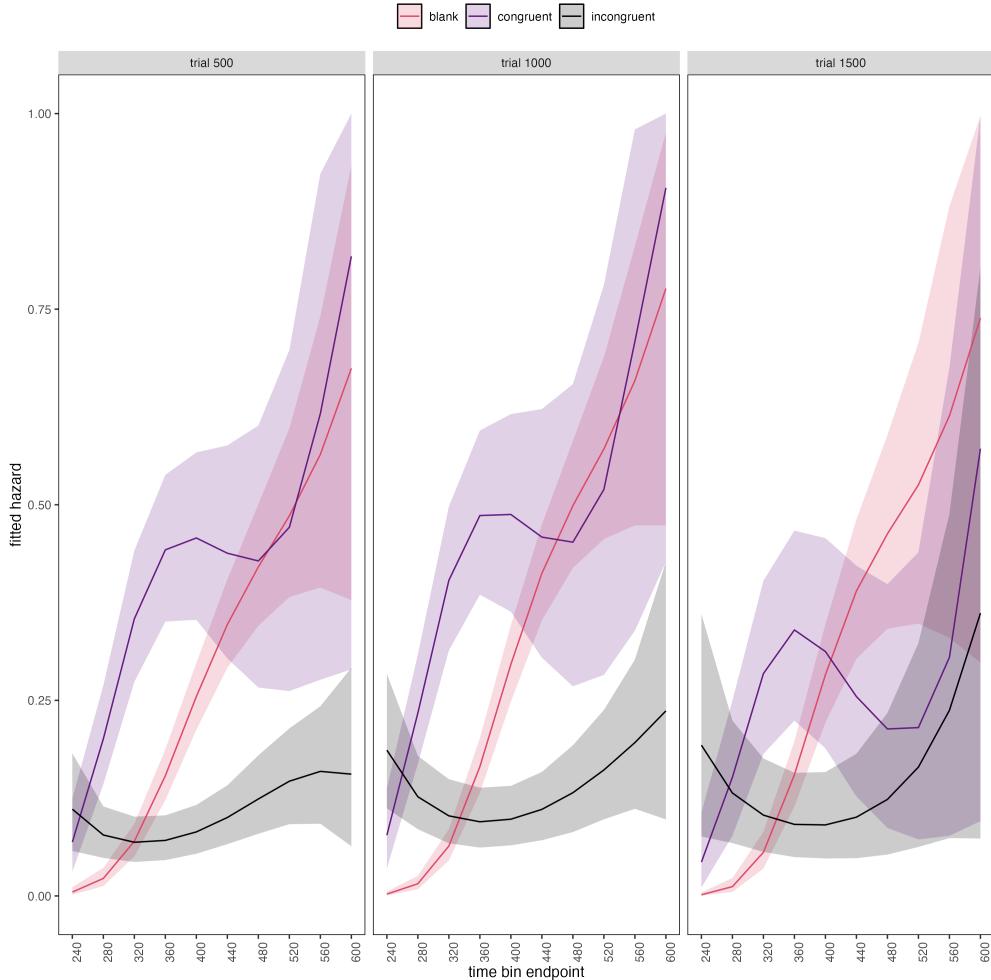


Figure 6. Model-based hazard functions for participant 6 in trial 500, 1000, and 1500.

368

Tutorial 3: Fitting Frequentist hazard models

369

In this third tutorial we illustrate how to fit a multilevel hazard regression model in

370

the frequentist framework, for the data set used in the first tutorial. For illustration

371

purposes, we only fitted model M3 using the function `glmer()` from the package `lme4`.

```
model_M3_f <- glmer(event ~ 1 + condition*period_9 +
  condition*I(period_9^2) +
  condition*I(period_9^3) +
  trial_c*period_9 +
  trial_c*I(period_9^2) +
  trial_c*I(period_9^3) +
```

```
(1 + condition*period_9 +
  condition*I(period_9^2) +
  condition*I(period_9^3) +
  trial_c*period_9 +
  trial_c*I(period_9^2) +
  trial_c*I(period_9^3) | pid),  
  
# control parameters, data set, and complementary log-log link function
control = glmerControl(optimizer = c("nlminbwrap"),
                        optCtrl = list(maxfun=10000000)),
data=M3_data,
family=binomial(link="cloglog"))
```

372 In Figure 7 we compare the parameter estimates of model M3 from brm() with those
373 of glmer().

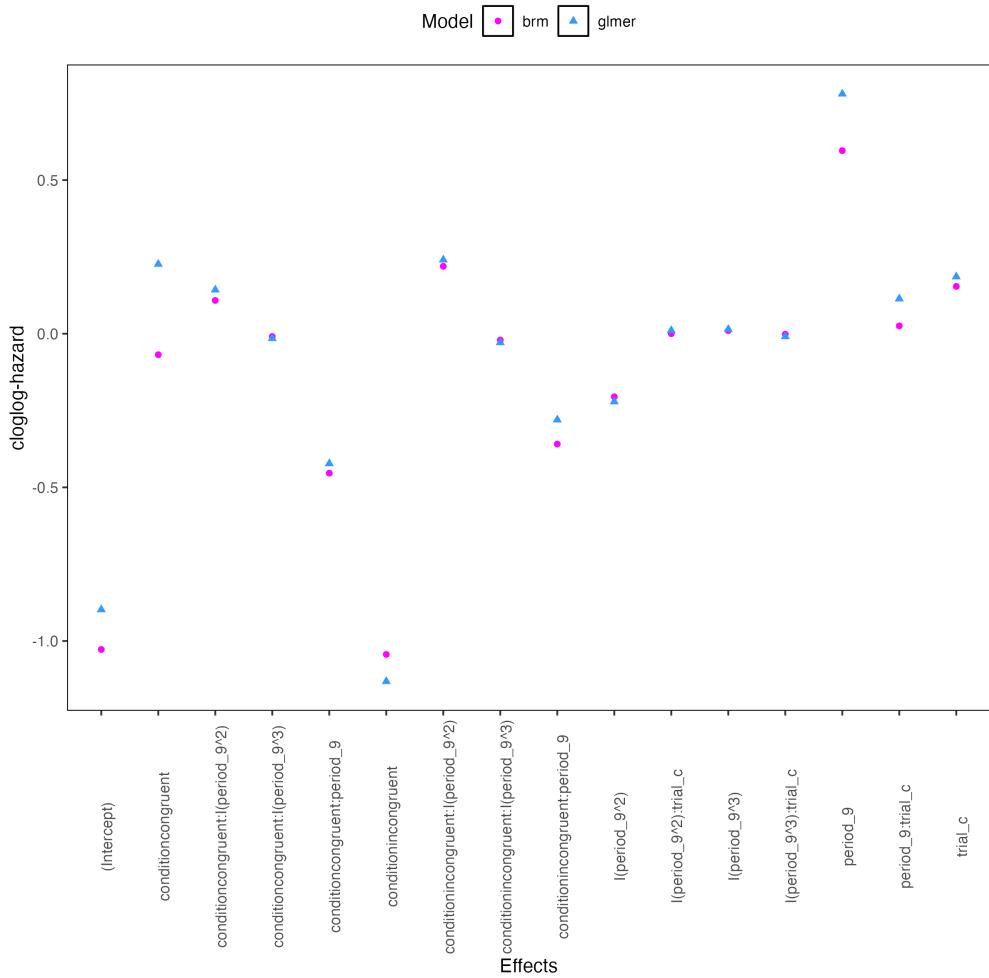


Figure 7. Parameter estimates for model M3 from brm() and glmer().

374 Figure 7 confirms that the parameter estimates from both Bayesian and frequentist
 375 models are pretty similar. However, the random effects structure of model M3 was already
 376 too complex for the frequentist model as it did not converge and resulted in a singular fit.
 377 This is of course one of the reasons why Bayesian modeling has become so popular in the
 378 last years.

Tutorial 4: Calculating descriptive statistics when there are two independent variables

In this final tutorial we illustrate how to calculate and plot the descriptive statistics

for the full data set of Experiment 1 of Panis and Schmidt (2016), which includes two independent variables: mask type and prime type. As we use the same functional programming approach as in tutorial 1, we simply present the sample-based functions for participant 6 in Figure 8. Note the negative compatibility effect in the hazard and conditional accuracy functions when a (relevant, irrelevant, or lines) mask is present.



Figure 8. Sample-based discrete-time hazard, survivor, and conditional accuracy functions for participant 6, as a function of prime type (blank, congruent, incongruent) and mask type (no mask, relevant, irrelevant, lines).

387

Discussion

388 We noticed that many researchers are still reluctant to abandon analysis-of-variance
389 and switch to event history analysis when analyzing time-to-event data. By providing this
390 tutorial, we hope that researchers will start using descriptive and inferential hazard
391 analysis more often, due to the many advantages described below. While we focused here
392 on within-subject, factorial, small- N designs, it is important to realize that event history
393 analysis can be applied to other designs as well (large- N designs with only one
394 measurement per subject, between-subject designs, etc.). We also discuss the issue of
395 individual differences, limitations, and extensions.

396 **Advantages of hazard analysis**

397 Statisticians and mathematical psychologists recommend focusing on the hazard
398 function when analyzing time-to-event data for various reasons. First, as discussed by
399 Holden, Van Orden, and Turvey (2009), “probability density functions can appear nearly
400 identical, both statistically and to the naked eye, and yet are clearly different on the basis
401 of their hazard functions (but not vice versa). Hazard functions are thus more diagnostic
402 than density functions” (p. 331) when one is interested in studying the detailed shape of a
403 RT distribution (see also Figure 1 in Panis, Schmidt, et al., 2020).

404 Second, because RT distributions may differ from one another in multiple ways,
405 Townsend (1990) developed a dominance hierarchy of statistical differences between two
406 arbitrary distributions A and B. For example, if $F_A(t) > F_B(t)$ for all t , then both
407 cumulative distribution functions are said to show a complete ordering. Townsend (1990)
408 showed that a complete ordering on the hazard functions $-\lambda_A(t) > \lambda_B(t)$ for all t —
409 implies a complete ordering on both the cumulative distribution and survivor functions
410 $-F_A(t) > F_B(t)$ and $S_A(t) < S_B(t)$ —which in turn implies an ordering on the mean
411 latencies—mean A < mean B. In contrast, an ordering on two means does *not* imply a

412 complete ordering on the corresponding $F(t)$ and $S(t)$ functions, and a complete ordering
413 on these latter functions does *not* imply a complete ordering on the corresponding hazard
414 functions. This means that stronger conclusions can be drawn from data when comparing
415 the hazard functions using EHA. For example, when mean A < mean B, the hazard
416 functions might show a complete ordering (i.e., for all t), a partial ordering (e.g., only for t
417 > 300 ms, or only for t < 500 ms), or they may cross each other one or more times. As a
418 result, instead of using delta-plots for RT – differences in quantiles from $F(t)^{-1}$ – one can
419 simply plot delta-h(t) functions (see Panis, 2020).

420 Third, EHA does not discard right-censored observations when estimating hazard
421 functions, that is, trials for which we do not observe a response during the data collection
422 period in a trial so that we only know that the RT must be larger than some value (i.e., the
423 response deadline). This is important because although a few right-censored observations
424 are inevitable in most RT tasks, a lot of right-censored observations are expected in
425 experiments on masking, the attentional blink, and so forth. In other words, by using EHA
426 you can analyze RT data from experiments that typically do not measure response times.
427 As a result, EHA can also deal with long RTs in experiments without a response deadline,
428 which are typically treated as outliers and are discarded before calculating a mean. This
429 orthodox procedure can lead to a sampling bias, however, which results in underestimation
430 of the mean. By introducing a fixed censoring time for all trials at the end of the analysis
431 time window, trials with long RTs are not discarded but contribute to the risk set of each
432 bin.

433 Fourth, hazard modeling allows incorporating time-varying explanatory covariates
434 such as heart rate, electroencephalogram (EEG) signal amplitude, gaze location, etc.
435 (Allison, 2010). This is useful for linking physiological effects with behavioral effects when
436 performing cognitive psychophysiology (Meyer, Osman, Irwin, & Yantis, 1988).

437 Finally, as explained by Kelso, Dumas, and Tognoli (2013), it is crucial to first have a

438 precise description of the macroscopic behavior of a system (here: $h(t)$ and $ca(t)$ functions)
439 in order to know what to derive on the microscopic level. EHA can thus solve the problem
440 of model mimicry, i.e., the fact that different computational models can often predict the
441 same mean RTs as observed in the empirical data, but not necessarily the detailed shapes
442 of the empirical RT hazard distributions. Also, fitting parametric functions or
443 computational models to data without studying the shape of the empirical discrete-time
444 $h(t)$ and $ca(t)$ functions can miss important features in the data (Panis, Moran, et al.,
445 2020; Panis & Schmidt, 2016).

446 Individual differences

447 One important issue is that of possible individual differences in the overall location of
448 the distribution, and the time course of psychological effects. For example, when you wait
449 for a response of the participant on each trial, you allow the participant to have control
450 over the trial duration, and some participants might respond only when they are confident
451 that their emitted response will be correct. These issues can be avoided by introducing a
452 (relatively short) response deadline in each trial, e.g., 600 ms for simple detection tasks,
453 1000 ms for more difficult discrimination tasks, or 2 s for tasks requiring extended
454 high-level processing. Because EHA can deal in a straightforward fashion with
455 right-censored observations (i.e., trials without an observed response), introducing a
456 response deadline is recommended when designing RT experiments. Furthermore,
457 introducing a response deadline and asking participants to respond before the deadline as
458 much as possible, will also lead to individual distributions that overlap in time, which is
459 important when selecting a common analysis time window when fitting hazard models.
460 But even when using a response deadline, participants can differ qualitatively in the effects
461 they display (see Panis, 2020). One way to deal with this is to describe and interpret the
462 different patterns. Another way is to run a clustering algorithm on the individual hazard
463 estimates across all conditions. The obtained dendrogram can then be used to identify a

464 (hopefully big) cluster of participants that behave similarly, and to identify a (hopefully
465 small) cluster of participants with outlying behavioral patterns. One might then exclude
466 the outlying participants before fitting a hazard model.

467 **Limitation(s)**

468 Compared to the orthodox method – comparing means with ANOVA –, the most
469 important limitation of multilevel hazard modeling is that it might take a long time to
470 estimate the parameters. Another issue that might initially look as a limitation is that you
471 need a relatively large number of trials per condition to estimate the hazard function with
472 high temporal resolution. However, as nature does not reveal itself easily, obtaining insight
473 into behavioral dynamics simply requires more data per condition than is usually collected
474 under the orthodox method. In general, there is a tradeoff between the number of trials per
475 condition and the temporal resolution (i.e., bin width) of the hazard function; We therefore
476 recommend to design as many trials as possible per experimental condition given the
477 available resources.

478 **Extensions**

479 The hazard models in this tutorial assume that there is one event of interest. For RT
480 data, this event constitutes a single transition between an “idle” state and a “responded”
481 state. However, in certain situations, more than one event of interest might exist. For
482 example, an individual might transition back and forth between a “healthy” state and a
483 “depression” state, before being absorbed in a final “death” state. When you have data on
484 the timing of these transitions, one can apply multi-state models which generalize survival
485 analysis to transitions between three or more states (Steele, Goldstein, & Browne, 2004).
486 Also, the predictor variables in this tutorial are time-invariant, i.e., their value did not
487 change over the course of a trial. Thus, another extension is to include time-varying
488 predictors, i.e., predictors whose value can change across the time bins within a trial.

489

Conclusions

490 RT and accuracy distributions are a rich source of information on the time course of
491 cognitive processing. The changing effects of our experimental manipulations with
492 increases in waiting time become strikingly clear when looking at response hazards and
493 microlevel speed-accuracy trade-off functions. We hope that experimental psychologists
494 and cognitive neuroscientists are less reluctant to embrace event history analysis after
495 reading this tutorial.

496

References

- 497 Allison, P. D. (1982). Discrete-Time Methods for the Analysis of Event Histories.
498 *Sociological Methodology*, 13, 61. <https://doi.org/10.2307/270718>
- 499 Allison, P. D. (2010). *Survival analysis using SAS: A practical guide* (2. ed). Cary, NC:
500 SAS Press.
- 501 Aust, F. (2019). *Citr: 'RStudio' add-in to insert markdown citations*. Retrieved from
502 <https://github.com/crsh/citr>
- 503 Aust, F., & Barth, M. (2023). *papaja: Prepare reproducible APA journal articles with R*
504 *Markdown*. Retrieved from <https://github.com/crsh/papaja>
- 505 Baker, D. H., Vilidaite, G., Lygo, F. A., Smith, A. K., Flack, T. R., Gouws, A. D., &
506 Andrews, T. J. (2021). Power contours: Optimising sample size and precision in
507 experimental psychology and human neuroscience. *Psychological Methods*, 26(3),
508 295–314. <https://doi.org/10.1037/met0000337>
- 509 Barth, M. (2023). *tinylabes: Lightweight variable labels*. Retrieved from
510 <https://cran.r-project.org/package=tinylabes>
- 511 Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects
512 models using lme4. *Journal of Statistical Software*, 67(1), 1–48.
513 <https://doi.org/10.18637/jss.v067.i01>
- 514 Bates, D., Maechler, M., & Jagan, M. (2024). *Matrix: Sparse and dense matrix classes and*
515 *methods*. Retrieved from <https://CRAN.R-project.org/package=Matrix>
- 516 Bengtsson, H. (2021). A unifying framework for parallel and distributed processing in r
517 using futures. *The R Journal*, 13(2), 208–227. <https://doi.org/10.32614/RJ-2021-048>
- 518 Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan.
519 *Journal of Statistical Software*, 80(1), 1–28. <https://doi.org/10.18637/jss.v080.i01>
- 520 Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms.
521 *The R Journal*, 10(1), 395–411. <https://doi.org/10.32614/RJ-2018-017>
- 522 Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal*

- 523 *of Statistical Software*, 100(5), 1–54. <https://doi.org/10.18637/jss.v100.i05>
- 524 Eddelbuettel, D., & Balamuta, J. J. (2018). Extending R with C++: A Brief Introduction
525 to Rcpp. *The American Statistician*, 72(1), 28–36.
526 <https://doi.org/10.1080/00031305.2017.1375990>
- 527 Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal
528 of Statistical Software*, 40(8), 1–18. <https://doi.org/10.18637/jss.v040.i08>
- 529 Gabry, J., Češnovar, R., Johnson, A., & Brodner, S. (2024). *Cmdstanr: R interface to
530 'CmdStan'*. Retrieved from <https://github.com/stan-dev/cmdstanr>
- 531 Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization
532 in bayesian workflow. *J. R. Stat. Soc. A*, 182, 389–402.
533 <https://doi.org/10.1111/rssa.12378>
- 534 Girard, J. (2024). *Standist: What the package does (one line, title case)*. Retrieved from
535 <https://github.com/jmgirard/standist>
- 536 Grolemund, G., & Wickham, H. (2011). Dates and times made easy with lubridate.
537 *Journal of Statistical Software*, 40(3), 1–25. Retrieved from
538 <https://www.jstatsoft.org/v40/i03/>
- 539 Holden, J. G., Van Orden, G. C., & Turvey, M. T. (2009). Dispersion of response times
540 reveals cognitive dynamics. *Psychological Review*, 116(2), 318–342.
541 <https://doi.org/10.1037/a0014849>
- 542 Kantowitz, B. H., & Pachella, R. G. (2021). The Interpretation of Reaction Time in
543 Information-Processing Research 1. *Human Information Processing*, 41–82.
544 <https://doi.org/10.4324/9781003176688-2>
- 545 Kay, M. (2023). *tidybayes: Tidy data and geoms for Bayesian models*.
546 <https://doi.org/10.5281/zenodo.1308151>
- 547 Kelso, J. A. S., Dumas, G., & Tognoli, E. (2013). Outline of a general theory of behavior
548 and brain coordination. *Neural Networks: The Official Journal of the International
549 Neural Network Society*, 37, 120–131. <https://doi.org/10.1016/j.neunet.2012.09.003>

- 550 Kurz, A. S. (2023a). *Applied longitudinal data analysis in brms and the tidyverse* (version
551 0.0.3). Retrieved from <https://bookdown.org/content/4253/>
- 552 Kurz, A. S. (2023b). *Statistical rethinking with brms, ggplot2, and the tidyverse: Second*
553 *edition* (version 0.4.0). Retrieved from <https://bookdown.org/content/4857/>
- 554 Luce, R. D. (1991). *Response times: Their role in inferring elementary mental organization*
555 (1. issued as paperback). Oxford: Univ. Press.
- 556 McElreath, R. (2018). *Statistical Rethinking: A Bayesian Course with Examples in R and*
557 *Stan* (1st ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/9781315372495>
- 558 Meyer, D. E., Osman, A. M., Irwin, D. E., & Yantis, S. (1988). Modern mental
559 chronometry. *Biological Psychology*, 26(1-3), 3–67.
560 [https://doi.org/10.1016/0301-0511\(88\)90013-0](https://doi.org/10.1016/0301-0511(88)90013-0)
- 561 Müller, K., & Wickham, H. (2023). *Tibble: Simple data frames*. Retrieved from
562 <https://CRAN.R-project.org/package=tibble>
- 563 Neuwirth, E. (2022). *RColorBrewer: ColorBrewer palettes*. Retrieved from
564 <https://CRAN.R-project.org/package=RColorBrewer>
- 565 Panis, S. (2020). How can we learn what attention is? Response gating via multiple direct
566 routes kept in check by inhibitory control processes. *Open Psychology*, 2(1), 238–279.
567 <https://doi.org/10.1515/psych-2020-0107>
- 568 Panis, S., Moran, R., Wolkersdorfer, M. P., & Schmidt, T. (2020). Studying the dynamics
569 of visual search behavior using RT hazard and micro-level speed–accuracy tradeoff
570 functions: A role for recurrent object recognition and cognitive control processes.
571 *Attention, Perception, & Psychophysics*, 82(2), 689–714.
572 <https://doi.org/10.3758/s13414-019-01897-z>
- 573 Panis, S., Schmidt, F., Wolkersdorfer, M. P., & Schmidt, T. (2020). Analyzing Response
574 Times and Other Types of Time-to-Event Data Using Event History Analysis: A Tool
575 for Mental Chronometry and Cognitive Psychophysiology. *I-Perception*, 11(6),
576 2041669520978673. <https://doi.org/10.1177/2041669520978673>

- 577 Panis, S., & Schmidt, T. (2016). What Is Shaping RT and Accuracy Distributions? Active
578 and Selective Response Inhibition Causes the Negative Compatibility Effect. *Journal of*
579 *Cognitive Neuroscience*, 28(11), 1651–1671. https://doi.org/10.1162/jocn_a_00998
- 580 Panis, S., & Schmidt, T. (2022). When does “inhibition of return” occur in spatial cueing
581 tasks? Temporally disentangling multiple cue-triggered effects using response history
582 and conditional accuracy analyses. *Open Psychology*, 4(1), 84–114.
583 <https://doi.org/10.1515/psych-2022-0005>
- 584 Panis, S., Torfs, K., Gillebert, C. R., Wagemans, J., & Humphreys, G. W. (2017).
585 Neuropsychological evidence for the temporal dynamics of category-specific naming.
586 *Visual Cognition*, 25(1-3), 79–99. <https://doi.org/10.1080/13506285.2017.1330790>
- 587 Panis, S., & Wagemans, J. (2009). Time-course contingencies in perceptual organization
588 and identification of fragmented object outlines. *Journal of Experimental Psychology:*
589 *Human Perception and Performance*, 35(3), 661–687.
590 <https://doi.org/10.1037/a0013547>
- 591 Pedersen, T. L. (2024). *Patchwork: The composer of plots*. Retrieved from
592 <https://CRAN.R-project.org/package=patchwork>
- 593 Pinheiro, J. C., & Bates, D. M. (2000). *Mixed-effects models in s and s-PLUS*. New York:
594 Springer. <https://doi.org/10.1007/b98882>
- 595 R Core Team. (2024). *R: A language and environment for statistical computing*. Vienna,
596 Austria: R Foundation for Statistical Computing. Retrieved from
597 <https://www.R-project.org/>
- 598 Schmidt, T., Panis, S., Wolkersdorfer, M. P., & Vorberg, D. (2022). Response inhibition in
599 the Negative Compatibility Effect in the absence of inhibitory stimulus features. *Open*
600 *Psychology*, 4(1), 219–230. <https://doi.org/10.1515/psych-2022-0012>
- 601 Singer, J. D., & Willett, J. B. (2003). *Applied Longitudinal Data Analysis: Modeling*
602 *Change and Event Occurrence*. Oxford, New York: Oxford University Press.
- 603 Smith, P. L., & Little, D. R. (2018). Small is beautiful: In defense of the small-N design.

- 604 *Psychonomic Bulletin & Review*, 25(6), 2083–2101.
605 <https://doi.org/10.3758/s13423-018-1451-8>
- 606 Steele, F., Goldstein, H., & Browne, W. (2004). A general multilevel multistate competing
607 risks model for event history data, with an application to a study of contraceptive use
608 dynamics. *Statistical Modelling*, 4(2), 145–159.
609 <https://doi.org/10.1191/1471082X04st069oa>
- 610 Townsend, J. T. (1990). Truth and consequences of ordinal differences in statistical
611 distributions: Toward a theory of hierarchical inference. *Psychological Bulletin*, 108(3),
612 551–567. <https://doi.org/10.1037/0033-2909.108.3.551>
- 613 Wickelgren, W. A. (1977). Speed-accuracy tradeoff and information processing dynamics.
614 *Acta Psychologica*, 41(1), 67–85. [https://doi.org/10.1016/0001-6918\(77\)90012-9](https://doi.org/10.1016/0001-6918(77)90012-9)
- 615 Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New
616 York. Retrieved from <https://ggplot2.tidyverse.org>
- 617 Wickham, H. (2023a). *Forcats: Tools for working with categorical variables (factors)*.
618 Retrieved from <https://forcats.tidyverse.org/>
- 619 Wickham, H. (2023b). *Stringr: Simple, consistent wrappers for common string operations*.
620 Retrieved from <https://stringr.tidyverse.org>
- 621 Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., ...
622 Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43),
623 1686. <https://doi.org/10.21105/joss.01686>
- 624 Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A
625 grammar of data manipulation*. Retrieved from <https://dplyr.tidyverse.org>
- 626 Wickham, H., & Henry, L. (2023). *Purrr: Functional programming tools*. Retrieved from
627 <https://purrr.tidyverse.org/>
- 628 Wickham, H., Hester, J., & Bryan, J. (2024). *Readr: Read rectangular text data*. Retrieved
629 from <https://readr.tidyverse.org>
- 630 Wickham, H., Vaughan, D., & Girlich, M. (2024). *Tidyr: Tidy messy data*. Retrieved from

631 https://tidyverse.org