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

Sven Panis[1] & Richard Ramsey[1]

[1] ETH Zürich

Author Note

Neural Control of Movement lab, Department of Health Sciences and Technology (D-HEST).

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

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

Abstract

Time-to-event data such as response times, saccade latencies, and fixation durations are ubiquitous in experimental psychology. To move beyond mean performance measures, various distributional analyses have been proposed. Here we focus on one particular distributional analysis known as discrete-time event history analysis, a.k.a. hazard analysis, duration analysis, failure-time analysis, survival analysis, and transition analysis. Across four tutorials that we make publicly available on Github and OSF, we illustrate how to calculate and interpret descriptive statistics, and how to implement Bayesian and frequentist regression models, using the R packages tidyverse, brms, and lme4. Along the way, we discuss how to manage inter-individual differences, implications for experimental design, and how to select among various options when analysing time-to-event data using discrete-time survival analysis.

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

Word count: X

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

Introduction

In experimental psychology, it is still standard practice to analyse response times (RTs), saccade latencies, and fixation durations using analysis-of-variance. However, differences in means conceal when an experimental effect starts, how long it lasts, and whether its onset is time-locked to other events. Such information is useful not only for interpretation, but also for cognitive psychophysiology and computational model selection (Panis, Schmidt, Wolkersdorfer, & Schmidt, 2020). In this tutorial we focus on a distributional method for analyzing time-to-event data that is known as discrete-time event history analysis (EHA), a.k.a. survival, hazard, duration, failure-time, and transition analysis (Allison, 1982, 2010; Singer & Willett, 2003). Across four tutorials that we make publicly available on Github and OSF, we illustrate how to calculate and interpret descriptive statistics, and how to implement Bayesian and frequentist regression models, using the R packages tidyverse, brms, and lme4.

To apply EHA, one must be able to define the event of interest (any qualitative change that can be situated in time, e.g., a button press, saccade onset, fixation offset), time point zero (e.g., target stimulus onset, fixation onset), and measure the passage of time between time point zero and event occurrence in discrete or continuous time units.

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

Similarly, after dividing time in discrete, contiguous time bins indexed by t, let RT be a discrete random variable denoting the rank of the time bin in which a particular person's response occurs in a particular experimental condition. Discrete-time EHA focuses on the discrete-time hazard function $\mathrm{h(t) = P(RT = t| \ RT \geq t)}$ and the discrete-time survivor function $\mathrm{S(t) = P(RT > t) = [1\text{-}h(t)].[1\text{-}h(t\text{-}1)].[1\text{-}h(t\text{-}2)]\ldots[1\text{-}h(1)]}$, and not on the probability mass function $\mathrm{p(t) = h(t).S(t\text{-}1)}$ and the cumulative distribution function $\mathrm{F(t) = 1\text{-}S(t)}$. The discrete-time hazard probability function gives you the probability that the event occurs (sometime) in bin t, given that the event has not occurred yet in previous bins. Unlike the hazard function, which assesses the unique risk associated with each time bin, the survivor function cumulates the bin-by-bin risks of event *non*occurrence. For two-choice RT data, the discrete-time hazard function can be extended with the conditional accuracy function $\mathrm{ca(t) = P(correct \mid RT = t)}$, which gives you the probability that a response is correct given that it has been emitted in time bin t (Allison, 2010; Kantowitz & Pachella, 2021; Wickelgren, 1977). This latter function is also known as the micro-level speed-accuracy tradeoff function.

Statisticians and mathematical psychologists recommend focusing on the hazard function when analyzing time-to-event data for various reasons. First, as discussed by Holden, Van Orden, and Turvey (2009), "probability density functions can appear nearly identical, both statistically and to the naked eye, and yet are clearly different on the basis of their hazard functions (but not vice versa). Hazard functions are thus more diagnostic than density functions" (p. 331). Second, because RT distributions may differ from one another in multiple ways, Townsend (1990) developed a dominance hierarchy of statistical differences between two arbitrary distributions A and B. For example, if $\mathrm{F_A(t) > F_B(t)}$ for all t, then both cumulative distribution functions are said to show a complete ordering. Townsend (1990) showed that a complete ordering on the hazard functions —$\lambda_{\mathrm{A}}\mathrm{(t)} > \lambda_{\mathrm{B}}\mathrm{(t)}$ for all t— implies a complete ordering on both the cumulative distribution and survivor functions —$F_A(t) > F_B(t)$ and $S_A(t) < S_B(t)$— which in turn implies an ordering on the mean latencies —mean A < mean B. In contrast, an ordering on two means does not imply a complete ordering on the corresponding F(t) and S(t) functions, and a complete ordering on these latter functions does not imply a complete ordering on the corresponding hazard functions. This means that stronger conclusions can be drawn from data when comparing the hazard functions using EHA. For example, when mean A < mean B, the hazard functions might show a complete ordering (i.e., for all t), a partial ordering (e.g., only for t > 300 ms, or only for t < 500 ms), or they may cross each other one or more times. Third, EHA does not discard right-censored observations when estimating hazard functions, that is, trials for which we do not observe a response during the data collection period so that we only know that the RT must be larger than some value. This is important because although a few right-censored observations are inevitable in most RT tasks, a lot of right-censored observations are expected in experiments on masking, the attentional blink, and so forth. Fourth, hazard modeling allows incorporating time-varying explanatory covariates such as heart rate, electroencephalogram (EEG) signal amplitude, gaze location, etc. (Allison, 2010) which is useful for cognitive psychophysiology (Meyer, Osman, Irwin, & Yantis, 1988). Finally, as explained by Kelso, Dumas, and Tognoli (2013), it is crucial to first have a precise description of the macroscopic behavior of a system (here: h(t) and ca(t) functions) in order to know what to derive on the microscopic level. For example, fitting parametric functions or computational models to data without studying the shape of the h(t) and ca(t) functions can miss important features in the data (Panis, Moran, Wolkersdorfer, & Schmidt, 2020; Panis & Schmidt, 2016).

We focus on factorial within-subject designs in which a large number of observations are made on a relatively small number of participants (small-*N* designs). This approach emphasizes the precision and reproducibility of data patterns at the individual participant level to increase the inferential validity of the design (Baker et al., 2021; Smith & Little, 2018). In contrast to the large-*N* design that averages across many participants without being able to scrutinize individual data patterns, small-N designs retain crucial information about the data patterns of individual observers. This is of great advantage whenever participants differ systematically in their strategies or in the time-courses of their effects, so that blindly averaging them would lead to misleading data patterns. Indeed, Smith and Little (2018) argue that, "if psychology is to be a mature quantitative science, then its primary theoretical aim should be to investigate systematic functional relationships as they are manifested at the individual participant level" (p. 2083). Note that because statistical power derives both from the number of participants and from the number of repeated measures per participant and condition, small-N designs can have 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 this tutorial is mainly based on Allison (2010), Singer and Willett (2003), McElreath (2018), Kurz (2023a), and Kurz (2023b).

[1] We, furthermore, used the R-packages *bayesplot* (Version 1.11.1; Gabry, Simpson, Vehtari, Betancourt, & Gelman, 2019), *brms* (Version 2.21.0; Bürkner, 2017, 2018, 2021), *citr* (Version 0.3.2; Aust, 2019), *cmdstanr* (Version 0.8.1.9000; Gabry, Češnovar, Johnson, & Bronder, 2024), *dplyr* (Version 1.1.4; Wickham, François, Henry, Müller, & Vaughan, 2023), *forcats* (Version 1.0.0; Wickham, 2023a), *ggplot2* (Version 3.5.1; Wickham, 2016), *lubridate* (Version 1.9.3; Grolemund & Wickham, 2011), *papaja* (Version 0.1.2.9000; Aust & Barth, 2023), *patchwork* (Version 1.2.0; Pedersen, 2024), *purrr* (Version 1.0.2; Wickham & Henry, 2023), *RColorBrewer* (Version 1.1.3; Neuwirth, 2022), *Rcpp* (Eddelbuettel & Balamuta, 2018; Version 1.0.12; Eddelbuettel & François, 2011), *readr* (Version 2.1.5; Wickham, Hester, & Bryan, 2024), *RJ-2021-048* (Bengtsson, 2021, 2021, 2021, 2021, 2021, 2021), *standist* (Version 0.0.0.9000; Girard, 2024), *stringr* (Version 1.5.1; Wickham, 2023b), *tibble* (Version 3.2.1; Müller & Wickham, 2023), *tidybayes* (Version 3.0.6; Kay, 2023), *tidyr* (Version 1.3.1; Wickham, Vaughan, & Girlich, 2024), *tidyverse* (Version 2.0.0; Wickham et al., 2019), and *tinylabels* (Version 0.2.4; Barth, 2023).

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

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

- "pid", indicating unique participant IDs;
- "trial", indicating each unique trial per participant;
- "condition", a factor indicating the levels of the independent variable (1, 2, . . .) and the corresponding labels;
- "rt", indicating the response times in ms;
- "acc", inicating the accuracies (1/0).

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

```r
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")))
```

To set up the life tables and plots of the discrete-time functions h(t), S(t) and ca(t) using functional programming, one has to nest the data within participants using the group_nest() function, and supply a user-defined censoring time and bin width to our 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
```

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

- Removed 2 rows containing missing values or values outside the scale range (`geom_line()`).
- Removed 2 rows containing missing values or values outside the scale range (`geom_point()`).

- Removed 2 rows containing missing values or values outside the scale range (`geom_segment()`).

The warning messages are generated because some bins have no hazard and ca(t) estimates, and no error bars. They can thus safely be ignored. One can now inspect different aspects, including the life table for a particular condition of a particular subject, and a plot of the different functions for a particular participant.

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

Figure 1 displays the discrete-time hazard, survivor, and conditional accuracy functions for each prime condition for participant 6. By using discrete-time h(t) functions of event occurrence - in combination with ca(t) functions for two-choice tasks - one can provide an unbiased, time-varying, and probabilistic description of the latency and accuracy of responses based on all trials of any data set.

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

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

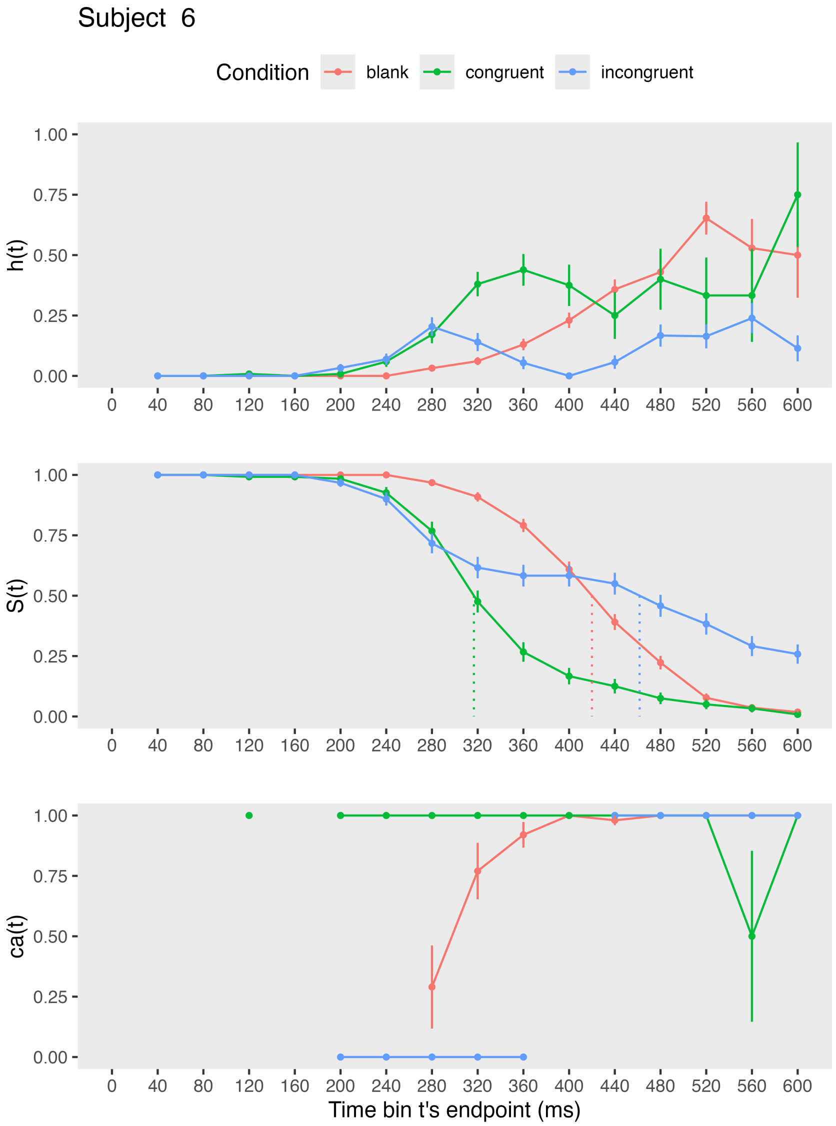

Figure 1. Estimated discrete-time hazard, survivor, and conditional accuracy functions for participant 6, as a function of the passage of discrete waiting time.

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

These results show that this participant is initially responding to the prime even though (s)he was instructed to only respond to the target, that response competition emerges in the incongruent prime condition around 300 ms, and that only later response are fully controlled by the target stimulus. Qualitatively similar results were obtained for the other five participants. Also, in their second Experiment, Panis and Schmidt (2016) showed that the negative compatibility effect in the mask-present conditions is time-locked to mask onset. This example shows that a simple difference between two means fails to reveal the dynamic behavior people display in many experimental paradigms (Panis, 2020; Panis, Moran, et al., 2020; Panis & Schmidt, 2022; Panis, Torfs, Gillebert, Wagemans, & Humphreys, 2017; Panis & Wagemans, 2009; Schmidt, Panis, Wolkersdorfer, & Vorberg, 2022). In other words, statistically controlling for the passage of time during data analysis is equally important as experimental control during the design of an experiment, to better understand human behavior in experimental paradigms. As we will show in Tutorials 2 and 3, statistical models for h(t) and ca(t) can each be implemented as generalized linear mixed regression models predicting event occurrence (1/0) and response accuracy (1/0) in each bin of a selected time range, respectively.

Tutorial 2: Fitting Bayesian hazard models

When you want to study how hazard depends on various predictors, you can fit regression models to the data (Singer & Willett, 2003). There are two analytic decisions one has to make. First, one has to select an analysis time range, i.e., a contiguous set of bins for which there is enough data for each participant. Second, one can choose the logit link function which transforms a (hazard) probability into the log of the odds ratio, or the complementary log-log (cloglog) link function, which yields the logarithm of the negated logarithm of the probability of event *non*occurrence. An important difference between these two link functions is that cloglog provides a discrete-time hazard model that has a built-in proportional hazards assumption, while logit provides a proportional odds assumption (see below). The cloglog link is preferred over the logit link when events can occur in principle at any time point within a bin, which is the case for RT data (Singer & Willett, 2003). Third, one has to choose a specification of the effect of discrete TIME (i.e., the time bin index t). One can choose a general specification (one intercept per bin) or a functional specification, such as a polynomial one.

An example discrete-time hazard model with three predictors (TIME, X1, X2) and the cloglog link function can be written as follows:

cloglog[h(t)] = ln(-ln[1-h(t)]) = $[\alpha_1\text{ONE} + \alpha_2(\text{TIME-1}) + \alpha_3(\text{TIME-1})^2] + [\beta_1\text{X}_1 + \beta_2\text{X}_2 + \beta_3\text{X}_2(\text{TIME-1})]$.

The main predictor variable TIME is the time bin index t that is centered on value 1 in this example. The first set of terms within brackets, the alpha parameters multiplied by their polynomial specifications of (centered) time, represents the shape of the baseline cloglog-hazard function (i.e., when all predictors X_i take on a value of zero). The second set of terms (the beta parameters) represents the vertical shift in the baseline cloglog-hazard for a 1 unit increase in the respective predictor. Predictors can be discrete, continuous, and time-varying or time-invariant. For example, the effect of a 1 unit increase in X_1 is to vertically shift the whole baseline cloglog-hazard function by β_1 cloglog-hazard units. However, if the predictor interacts linearly with time (see X_2 in the example), then the effect of a 1 unit increase in X_2 is to vertically shift the predicted cloglog-hazard in bin 1 by β_2 cloglog-hazard units (when TIME-1 = 0), in bin 2 by $\beta_2 + \beta_3$ cloglog-hazard units (when TIME-1 = 1), and so forth. To interpret the effects of the predictors, the parameter estimates are exponentiated, resulting in a hazard ratio (due to the use of the cloglog link).

In the case of a large-*N* design without repeated measurements, the parameters of a discrete-time hazard model can be estimated using standard logistic regression software (after expanding the typical person-trial-oriented data set into a person-trial-bin-oriented data set (see Tutorial 1); Allison (2010)). When there is clustering in the data, as in the case of a small-N design with repeated measurements, the parameters of a discrete-time hazard model can be estimated using population-averaged methods (e.g., Generalized Estimating Equations), and Bayesian or frequentist generalized linear mixed models (Allison, 2010).

In this second tutorial we illustrate how to fit a Bayesian hazard regression model for the masked response priming data set used in the first tutorial. In general, there are three assumptions one can make or relax when adding experimental predictor variables: The linearity assumption for continuous predictors (the effect of a 1 unit change is the same anywhere on the scale), the additivity assumption (predictors do not interact), and the proportionality assumption (predictors do not interact with TIME).

First, we select the analysis range (200,600] and the cloglog link, and use a polynomial to specify the effect of TIME in the “blank” prime condition. Second, based on previous work (Panis, 2020; Panis, Moran, et al., 2020; Panis & Schmidt, 2022; Panis et al., 2017; Panis & Wagemans, 2009) and because cognition is likely the behavior of a non-linear dynamical system [ref], we relax all three assumptions, as follows:

```
# load data
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 discrete TIME (period) on bin 9, and trial on trial 1000
ptb_data <- ptb_data %>% mutate(period_9 = period - 9,
                                trial_c = (trial - 1000)/1000)
# remove unnecessary columns before fitting a model
ptb_data <- ptb_data %>% select(-c(bl,tr,trial,period)) # 12840 obs. of 5 variables
```

```
priors <- c(
  set_prior("normal(0, 1)", class = "b"), # for beta parameters
  set_prior("student_t(7.61, 0, 1.57)", class = "b", coef = "Intercept"), # flat prior for intercept on hazard scale
  set_prior("normal(0, 1)", class = "sd"), # for standard deviation of RE
  set_prior("lkj(2)", class = "cor") # for correlations between RE
)
```

```
plan(multicore)
model_full_RE <-
   brm(data = ptb_data,
       family = binomial(link="cloglog"),
       event | trials(1) ~ 0 + Intercept +
                           condition*period_9*trial_c +
                           condition*I(period_9^2) +
                           condition*I(period_9^3) +
                           (1 + condition*period_9*trial_c +
                           condition*I(period_9^2) +
                           condition*I(period_9^3) | pid),
       prior = priors,
       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_full_RE")
```

To test whether (centered) trial number affects behavior, we fit a model without the variable trial_c.

Use WAIC to compare models.

Plot the effects of congruent and incongruent for each time bin for the selected model.

Plot the model-based hazard and survivor functions.

Tutorial 3: Fitting Frequentist hazard models

In this third tutorial we illustrate how to fit a frequentist hazard regression model for the data set used in the first tutorial.

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).

Discussion

Individual differences

- role of response deadlines, low-level vs. higher-level processes,
- clustering algorithms based on h(t) and ca(t) data
-

Cognitive psychophysiology and computational model selection

Power analysis

- example repo on github

Preregistration

- example preregistration for knot data

Conclusions

References

Allison, P. D. (1982). Discrete-Time Methods for the Analysis of Event Histories. *Sociological Methodology*, *13*, 61. https://doi.org/10.2307/270718

Allison, P. D. (2010). *Survival analysis using SAS: A practical guide* (2. ed). Cary, NC: SAS Press.

Aust, F. (2019). *Citr: 'RStudio' add-in to insert markdown citations*. Retrieved from https://github.com/crsh/citr

Aust, F., & Barth, M. (2023). *papaja: Prepare reproducible APA journal articles with R Markdown*. Retrieved from https://github.com/crsh/papaja

Baker, D. H., Vilidaite, G., Lygo, F. A., Smith, A. K., Flack, T. R., Gouws, A. D., & Andrews, T. J. (2021). Power contours: Optimising sample size and precision in experimental psychology and human neuroscience. *Psychological Methods*, *26*(3), 295–314. https://doi.org/10.1037/met0000337

Barth, M. (2023). *tinylabels: Lightweight variable labels*. Retrieved from https://cran.r-project.org/package=tinylabels

Bengtsson, H. (2021). A unifying framework for parallel and distributed processing in r using futures. *The R Journal*, *13*(2), 208–227. https://doi.org/10.32614/RJ-2021-048

Bürkner, P.-C. (2017). brms: An R package for Bayesian multilevel models using Stan. *Journal of Statistical Software*, *80*(1), 1–28. https://doi.org/10.18637/jss.v080.i01

Bürkner, P.-C. (2018). Advanced Bayesian multilevel modeling with the R package brms. *The R Journal*, *10*(1), 395–411. https://doi.org/10.32614/RJ-2018-017

Bürkner, P.-C. (2021). Bayesian item response modeling in R with brms and Stan. *Journal of Statistical Software*, *100*(5), 1–54. https://doi.org/10.18637/jss.v100.i05

Eddelbuettel, D., & Balamuta, J. J. (2018). Extending R with C++: A Brief Introduction to Rcpp. *The American Statistician*, *72*(1), 28–36. https://doi.org/10.1080/00031305.2017.1375990

Eddelbuettel, D., & François, R. (2011). Rcpp: Seamless R and C++ integration. *Journal* *of Statistical Software*, *40*(8), 1–18. https://doi.org/10.18637/jss.v040.i08

Gabry, J., Češnovar, R., Johnson, A., & Bronder, S. (2024). *Cmdstanr: R interface to 'CmdStan'*. Retrieved from https://github.com/stan-dev/cmdstanr

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in bayesian workflow. *J. R. Stat. Soc. A*, *182*, 389–402. https://doi.org/10.1111/rssa.12378

Girard, J. (2024). *Standist: What the package does (one line, title case)*. Retrieved from https://github.com/jmgirard/standist

Grolemund, G., & Wickham, H. (2011). Dates and times made easy with lubridate. *Journal of Statistical Software*, *40*(3), 1–25. Retrieved from https://www.jstatsoft.org/v40/i03/

Holden, J. G., Van Orden, G. C., & Turvey, M. T. (2009). Dispersion of response times reveals cognitive dynamics. *Psychological Review*, *116*(2), 318–342. https://doi.org/10.1037/a0014849

Kantowitz, B. H., & Pachella, R. G. (2021). The Interpretation of Reaction Time in Information-Processing Research 1. *Human Information Processing*, 41–82. https://doi.org/10.4324/9781003176688-2

Kay, M. (2023). *tidybayes: Tidy data and geoms for Bayesian models.* https://doi.org/10.5281/zenodo.1308151

Kelso, J. A. S., Dumas, G., & Tognoli, E. (2013). Outline of a general theory of behavior and brain coordination. *Neural Networks: The Official Journal of the International Neural Network Society*, *37*, 120–131. https://doi.org/10.1016/j.neunet.2012.09.003

Kurz, A. S. (2023a). *Applied longitudinal data analysis in brms and the tidyverse* (version 0.0.3). Retrieved from https://bookdown.org/content/4253/

Kurz, A. S. (2023b). *Statistical rethinking with brms, ggplot2, and the tidyverse: Second edition* (version 0.4.0). Retrieved from https://bookdown.org/content/4857/

Luce, R. D. (1991). *Response times: Their role in inferring elementary mental organization*

(1. issued as paperback). Oxford: Univ. Press.

McElreath, R. (2018). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan* (1st ed.). Chapman and Hall/CRC. https://doi.org/10.1201/9781315372495

Meyer, D. E., Osman, A. M., Irwin, D. E., & Yantis, S. (1988). Modern mental chronometry. *Biological Psychology*, *26*(1-3), 3–67. https://doi.org/10.1016/0301-0511(88)90013-0

Müller, K., & Wickham, H. (2023). *Tibble: Simple data frames.* Retrieved from https://CRAN.R-project.org/package=tibble

Neuwirth, E. (2022). *RColorBrewer: ColorBrewer palettes.* Retrieved from https://CRAN.R-project.org/package=RColorBrewer

Panis, S. (2020). How can we learn what attention is? Response gating via multiple direct routes kept in check by inhibitory control processes. *Open Psychology*, *2*(1), 238–279. https://doi.org/10.1515/psych-2020-0107

Panis, S., Moran, R., Wolkersdorfer, M. P., & Schmidt, T. (2020). Studying the dynamics of visual search behavior using RT hazard and micro-level speed–accuracy tradeoff functions: A role for recurrent object recognition and cognitive control processes. *Attention, Perception, & Psychophysics*, *82*(2), 689–714. https://doi.org/10.3758/s13414-019-01897-z

Panis, S., Schmidt, F., Wolkersdorfer, M. P., & Schmidt, T. (2020). Analyzing Response Times and Other Types of Time-to-Event Data Using Event History Analysis: A Tool for Mental Chronometry and Cognitive Psychophysiology. *I-Perception*, *11*(6), 2041669520978673. https://doi.org/10.1177/2041669520978673

Panis, S., & Schmidt, T. (2016). What Is Shaping RT and Accuracy Distributions? Active and Selective Response Inhibition Causes the Negative Compatibility Effect. *Journal of Cognitive Neuroscience*, *28*(11), 1651–1671. https://doi.org/10.1162/jocn_a_00998

Panis, S., & Schmidt, T. (2022). When does "inhibition of return" occur in spatial cueing tasks? Temporally disentangling multiple cue-triggered effects using response history and conditional accuracy analyses. *Open Psychology*, *4*(1), 84–114. https://doi.org/10.1515/psych-2022-0005

Panis, S., Torfs, K., Gillebert, C. R., Wagemans, J., & Humphreys, G. W. (2017). Neuropsychological evidence for the temporal dynamics of category-specific naming. *Visual Cognition*, *25*(1-3), 79–99. https://doi.org/10.1080/13506285.2017.1330790

Panis, S., & Wagemans, J. (2009). Time-course contingencies in perceptual organization and identification of fragmented object outlines. *Journal of Experimental Psychology: Human Perception and Performance*, *35*(3), 661–687. https://doi.org/10.1037/a0013547

Pedersen, T. L. (2024). *Patchwork: The composer of plots.* Retrieved from https://CRAN.R-project.org/package=patchwork

R Core Team. (2024). *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Schmidt, T., Panis, S., Wolkersdorfer, M. P., & Vorberg, D. (2022). Response inhibition in the Negative Compatibility Effect in the absence of inhibitory stimulus features. *Open Psychology*, *4*(1), 219–230. https://doi.org/10.1515/psych-2022-0012

Singer, J. D., & Willett, J. B. (2003). *Applied Longitudinal Data Analysis: Modeling Change and Event Occurrence*. Oxford, New York: Oxford University Press.

Smith, P. L., & Little, D. R. (2018). Small is beautiful: In defense of the small-N design. *Psychonomic Bulletin & Review*, *25*(6), 2083–2101. https://doi.org/10.3758/s13423-018-1451-8

Townsend, J. T. (1990). Truth and consequences of ordinal differences in statistical distributions: Toward a theory of hierarchical inference. *Psychological Bulletin*, *108*(3), 551–567. https://doi.org/10.1037/0033-2909.108.3.551

Wickelgren, W. A. (1977). Speed-accuracy tradeoff and information processing dynamics. *Acta Psychologica*, *41*(1), 67–85. https://doi.org/10.1016/0001-6918(77)90012-9

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis.* Springer-Verlag New York. Retrieved from https://ggplot2.tidyverse.org

Wickham, H. (2023a). *Forcats: Tools for working with categorical variables (factors).* Retrieved from https://forcats.tidyverse.org/

Wickham, H. (2023b). *Stringr: Simple, consistent wrappers for common string operations.* Retrieved from https://stringr.tidyverse.org

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation.* Retrieved from https://dplyr.tidyverse.org

Wickham, H., & Henry, L. (2023). *Purrr: Functional programming tools.* Retrieved from https://purrr.tidyverse.org/

Wickham, H., Hester, J., & Bryan, J. (2024). *Readr: Read rectangular text data.* Retrieved from https://readr.tidyverse.org

Wickham, H., Vaughan, D., & Girlich, M. (2024). *Tidyr: Tidy messy data.* Retrieved from https://tidyr.tidyverse.org