Earthquake Forecasting

Dissertation Project 2

Robin Lin s2435943

July 2023

```r
require(ETAS.inlabru)
require(ggplot2)
require(dplyr)
require(magrittr)
require(tidyquant)
require(rnaturalearth)
require(terra)
require(sf)
require(ggspatial)
require(rnaturalearthdata)
require(lubridate)

INLA::inla.setOption(num.threads = 2)
```

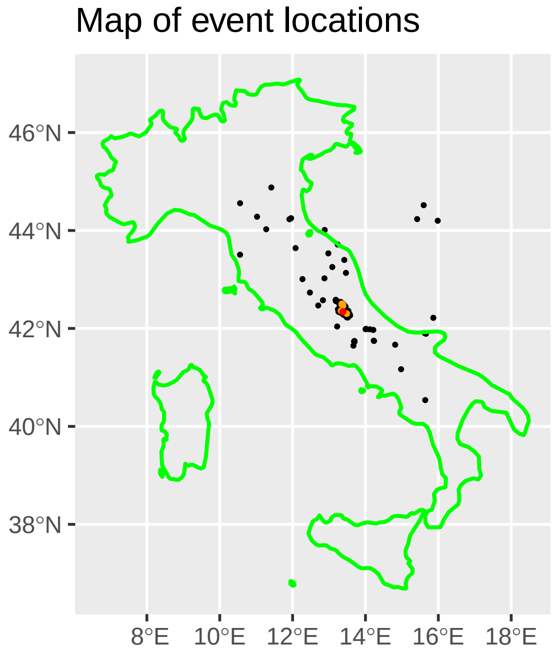
Copula transformation of the priors

```r
# set copula transformations list
link.f <- list(
  mu = \(x) gamma_t(x, 0.3, 0.6),
  K = \(x) unif_t(x, 0, 10),
  alpha = \(x) unif_t(x, 0, 10),
  c_ = \(x) unif_t(x, 0, 10),
  p = \(x) unif_t(x, 1, 10)
)

# set inverse copula transformations list
inv.link.f <- list(
  mu = \(x) inv_gamma_t(x, 0.3, 0.6),
  K = \(x) inv_unif_t(x, 0, 10),
  alpha = \(x) inv_unif_t(x, 0, 10),
  c_ = \(x) inv_unif_t(x, 0, 10),
  p = \(x) inv_unif_t(x, 1, 10)
)
```

Italy

```
# transform time string in Date object
horus$time_date <- as.POSIXct(
  horus$time_string,
  format = "%Y-%m-%dT%H:%M:%OS",
  tz = "UTC"
)
# There may be some incorrectly registered data-times in the original data set,
# that as.POSIXct() can't convert, depending on the system.
# These should ideally be corrected, but for now, we just remove the rows that
# couldn't be converted.
# horus <- na.omit(horus)

# set up parameters for selection
start.date <- as.POSIXct("2009-01-01T00:00:00",
                         format = "%Y-%m-%dT%H:%M:%OS")
end.date <- as.POSIXct("2010-01-01T00:00:00", format = "%Y-%m-%dT%H:%M:%OS")
min.longitude <- 10.5
max.longitude <- 16
min.latitude <- 40.5
max.latitude <- 45
M0 <- 2.5

# set up conditions for selection
aquila.sel <- (horus$time_date >= start.date) &
  (horus$time_date < end.date) &
  (horus$lon >= min.longitude) &
  (horus$lon <= max.longitude) &
  (horus$lat >= min.latitude) &
  (horus$lat <= max.latitude) &
  (horus$M >= M0)

# select
aquila <- horus[aquila.sel, ]
```

```r
italy.map <- ne_countries(country = 'Italy', returnclass = "sf",
                          scale = 'medium')

aquila.sf <- st_as_sf(aquila,
                     coords = c("lon", "lat"),
                     crs = st_crs('EPSG:4326'))
ggplot() +
  geom_sf(data = aquila.sf[aquila$M > 3,], size = 0.4) +
  geom_sf(data = italy.map, fill = alpha("lightgrey", 0), color = 'green',
          linewidth = 0.7) +
  geom_sf(data = aquila.sf[aquila$M > 5,], size = 0.5, color = 'orange') +
  geom_sf(data = aquila.sf[aquila$M > 6,], size = 0.6, color = 'red') +
  ggtitle("Map of event locations")
```



```r
ggplot(aquila, aes(time_date, M)) +
  geom_point() +
  theme_bw()
```

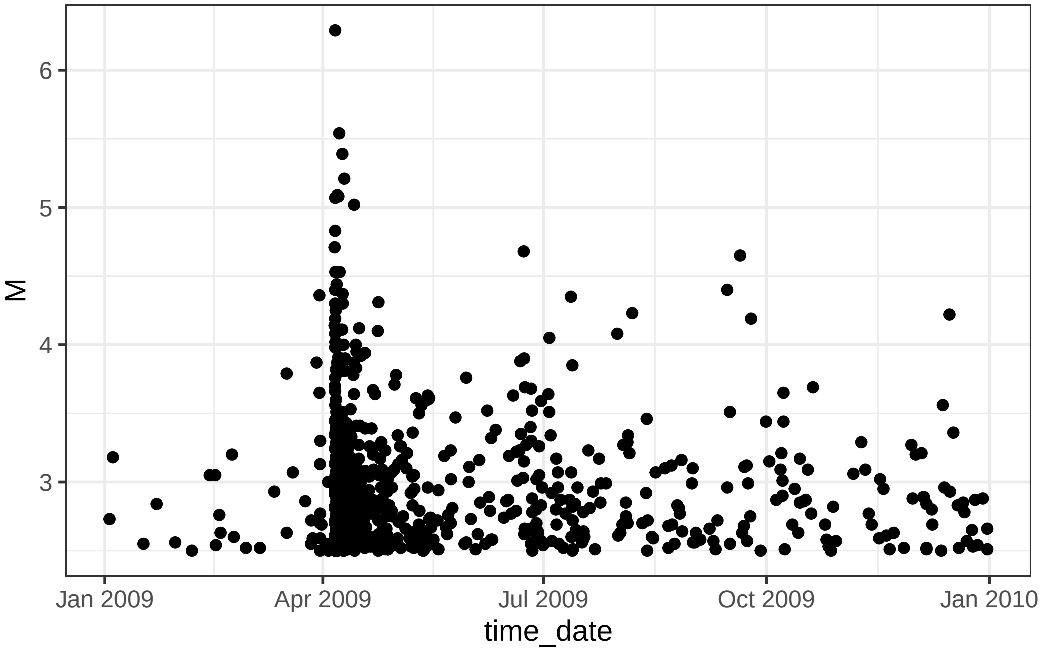


Figure 1: L'Aquila seismic sequence, times versus magnitudes

```r
# set up data.frame for model fitting
aquila.bru <- data.frame(
  ts = as.numeric(
    difftime(aquila$time_date, start.date, units = "days")
  ),
  magnitudes = aquila$M,
  idx.p = 1 : nrow(aquila)
)
```

```r
# set up list of initial values
th.init <- list(
  th.mu = inv.link.f$mu(0.5),
  th.K = inv.link.f$K(0.1),
  th.alpha = inv.link.f$alpha(1),
  th.c = inv.link.f$c_(0.1),
  th.p = inv.link.f$p(1.1)
)
```

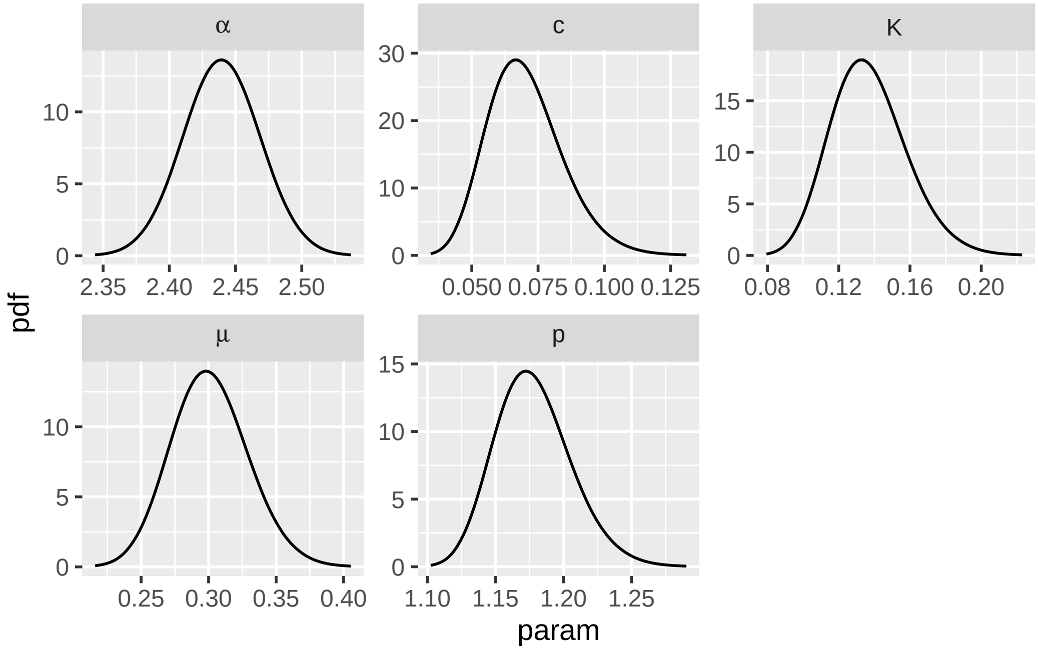
```r
# set up list of bru options
bru.opt.list <- list(
  bru_verbose = 3, # type of visual output
  bru_max_iter = 70, # maximum number of iterations
  # bru_method = list(max_step = 0.5),
  bru_initial = th.init # parameters' initial values
)
```

```r
# set starting and time of the time interval used for model fitting. In this case,
# we use the interval covered by the data.
T1 <- 0
T2 <- max(aquila.bru$ts) + 0.2 # Use max(..., na.rm = TRUE) if there may still be
# NAs here
# fit the model
aquila.fit <- Temporal.ETAS(
  total.data = aquila.bru,
  M0 = M0,
  T1 = T1,
  T2 = T2,
  link.functions = link.f,
  coef.t. = 1,
  delta.t. = 0.1,
  N.max. = 5,
  bru.opt = bru.opt.list
)
```

```
Start creating grid...
Finished creating grid, time  2.6984
```

```r
# create input list to explore model output
input_list <- list(
  model.fit = aquila.fit,
  link.functions = link.f
)
```

```r
# get marginal posterior information
post.list <- get_posterior_param(input.list = input_list)

# plot marginal posteriors
post.list$post.plot
```



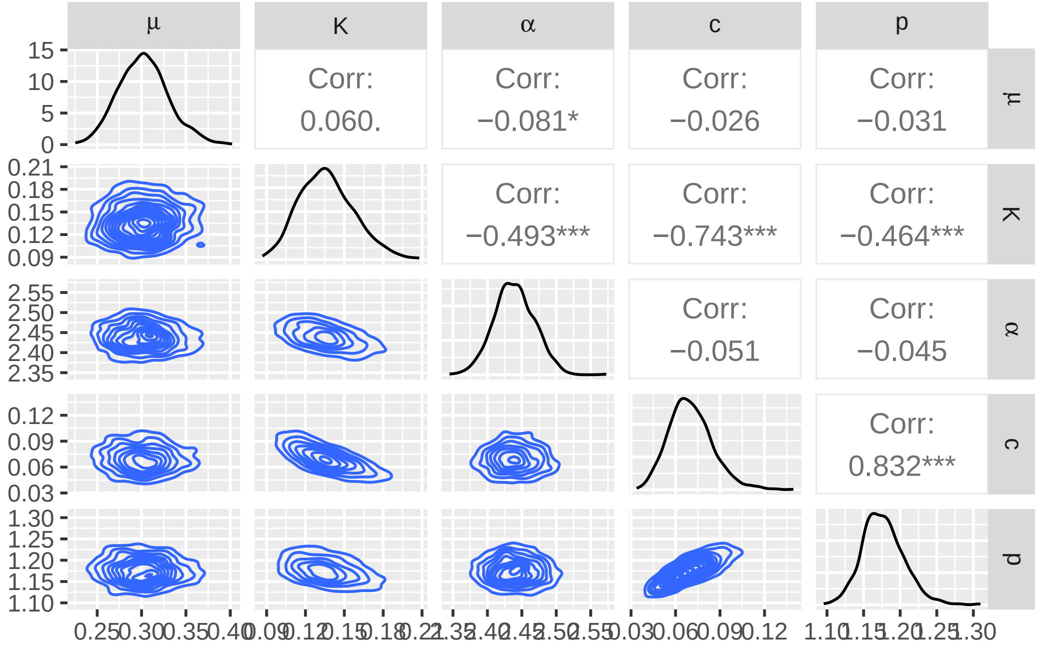
```r
post.samp <- post_sampling(
  input.list = input_list,
  n.samp = 1000,
  max.batch = 1000,
  ncore = num.cores
)

head(post.samp)
```

```
         mu         K    alpha          c        p
1 0.2928805 0.1268222 2.423836 0.08286909 1.204344
2 0.3603557 0.1214513 2.448041 0.08049121 1.179511
3 0.3072864 0.1322905 2.412915 0.07267388 1.169592
4 0.2833596 0.1660960 2.407091 0.05631260 1.152117
5 0.2539323 0.1206450 2.508145 0.06516970 1.174202
6 0.3142158 0.1313280 2.443396 0.06644567 1.174966
```

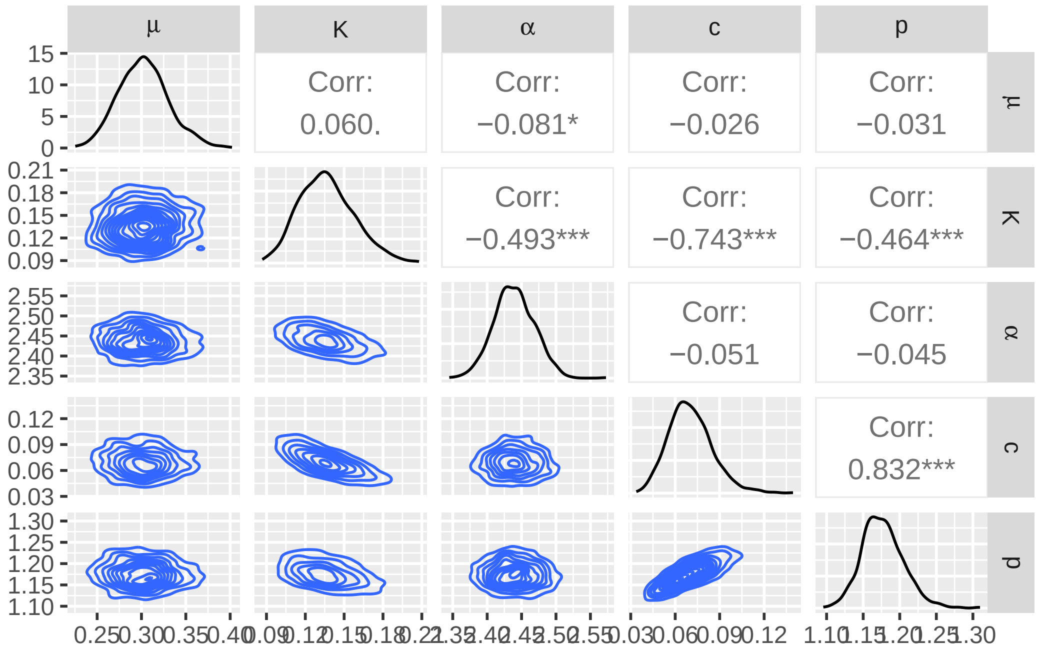
```r
pair.plot <- post_pairs_plot(
  post.samp = post.samp,
  input.list = NULL,
  n.samp = NULL,
  max.batch = 1000
)
```

```
Registered S3 method overwritten by 'GGally':
  method from
  +.gg   ggplot2
```

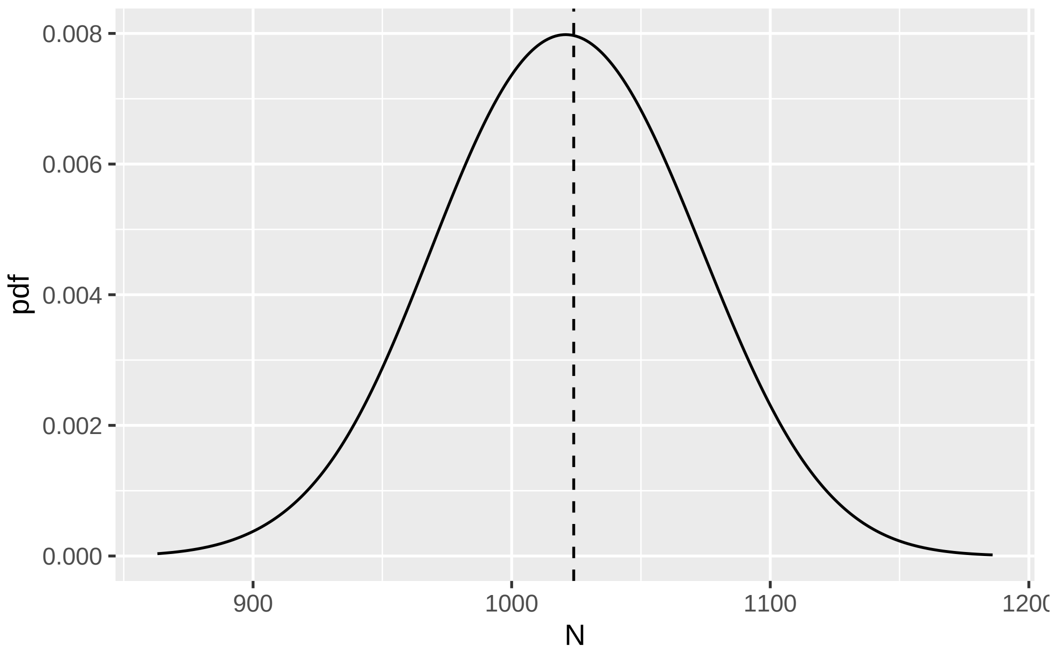
```
pair.plot$pair.plot
```



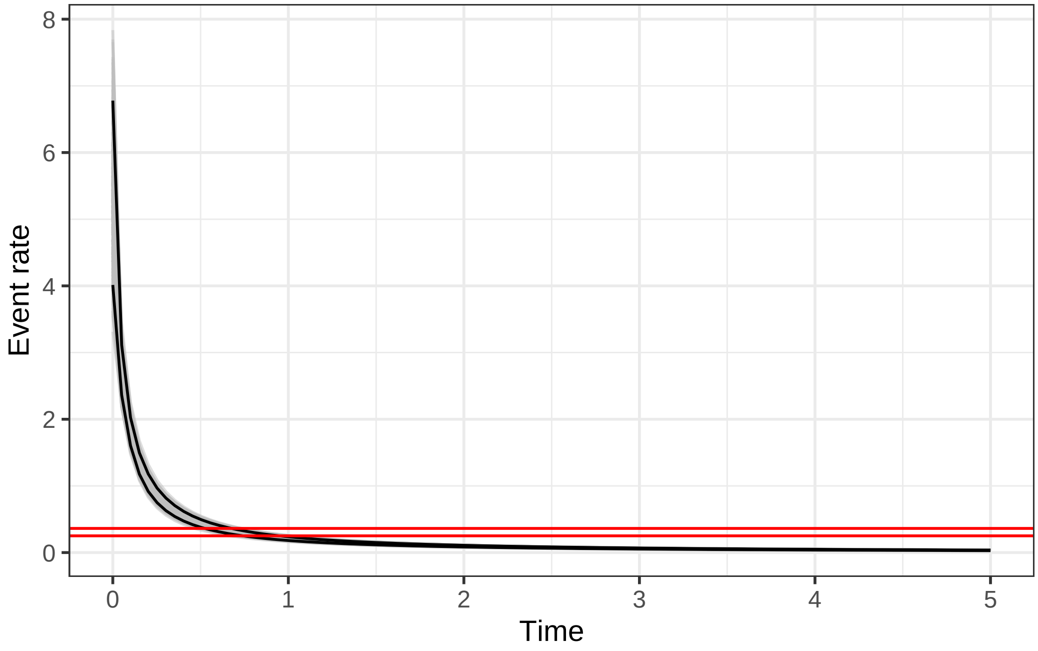
```
pair.plot <- post_pairs_plot(
  post.samp = post.samp,
  input.list = NULL,
  n.samp = NULL,
  max.batch = 1000
)
pair.plot$pair.plot
```



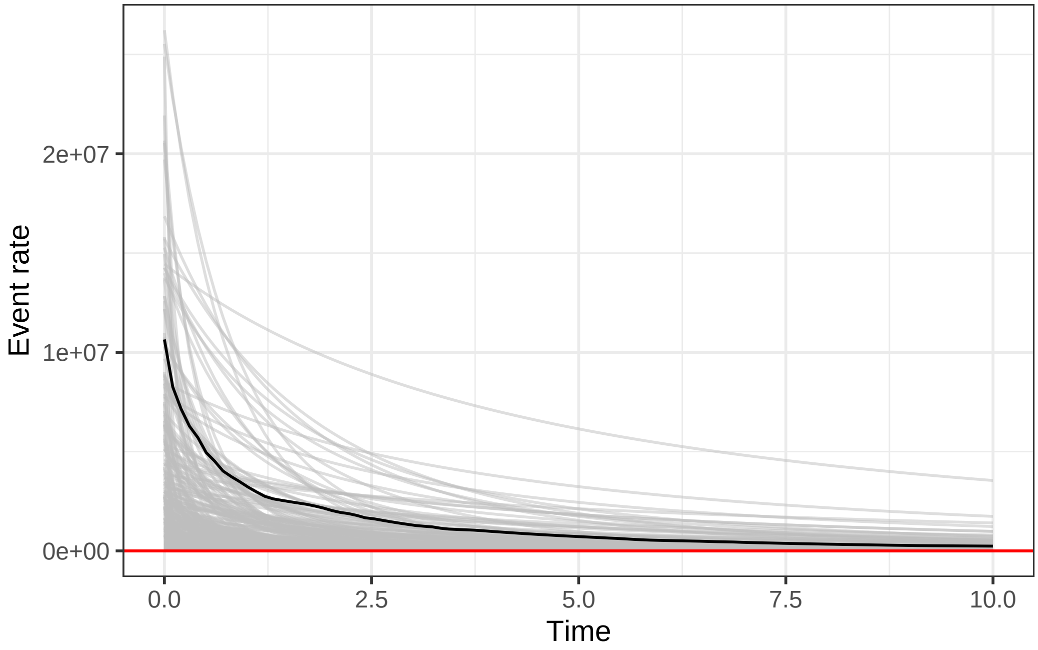
```r
# set additional elements of the list
input_list$T12 <- c(T1, T2)
input_list$M0 <- M0
input_list$catalog.bru <- aquila.bru
N.post <- get_posterior_N(input.list = input_list)
N.post$post.plot
```



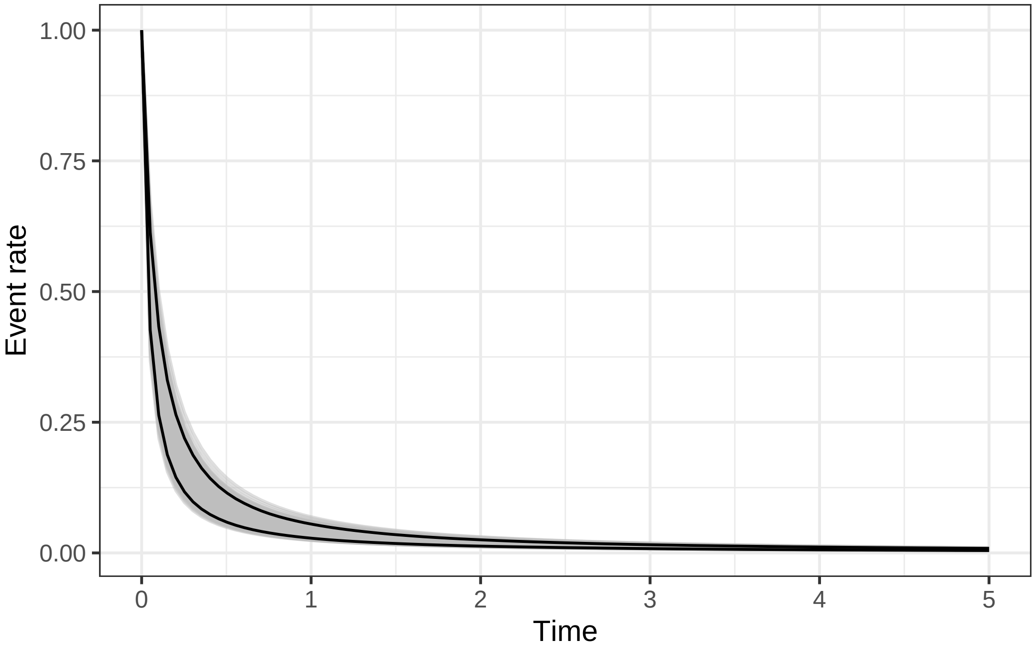
```r
triggering_fun_plot(
  input.list = input_list,
  post.samp = post.samp,
  n.samp = NULL, magnitude = 4,
  t.end = 5, n.breaks = 100
)
```



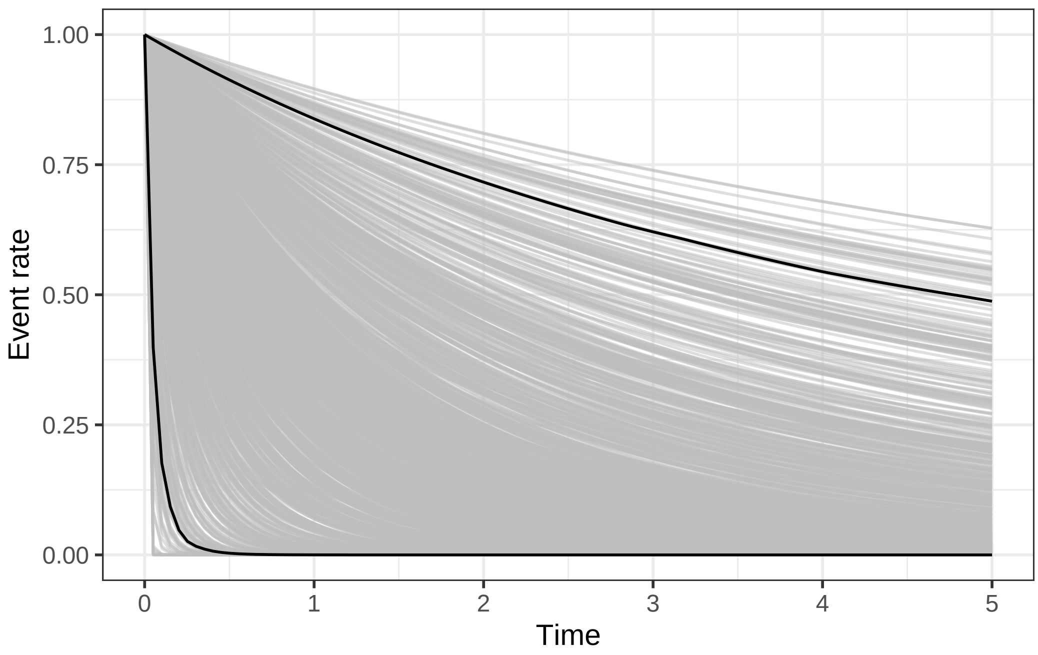
```
triggering_fun_plot_prior(input.list = input_list, magnitude = 4, n.samp = 1000, t.end = 1
```



```
omori_plot_posterior(input.list = input_list, post.samp = post.samp, n.samp = NULL, t.end
```



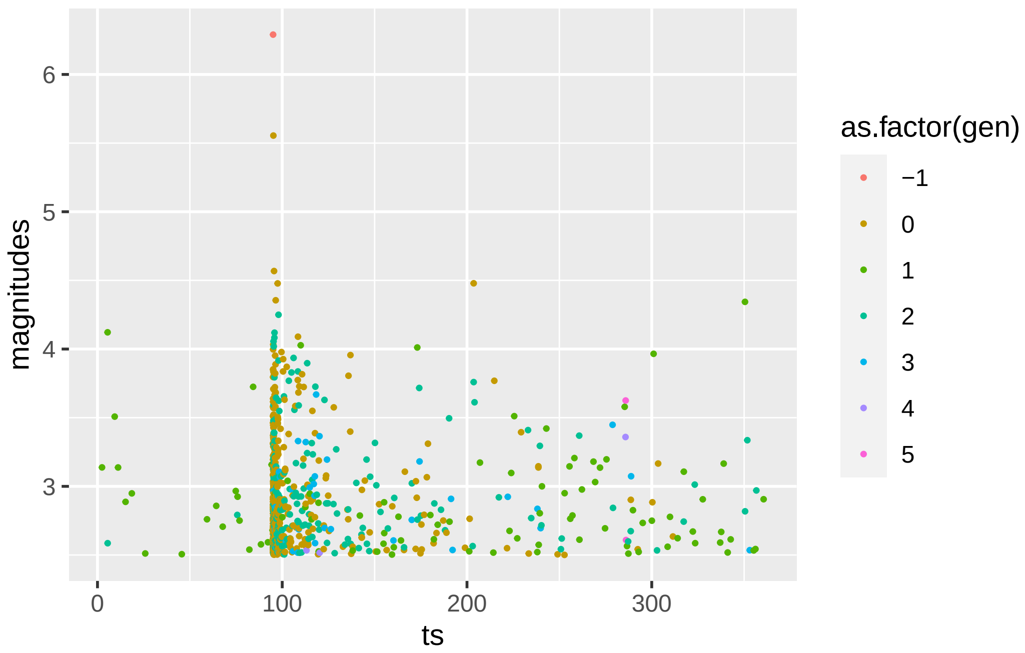
```
omori_plot_prior(input.list = input_list, n.samp = 1000, t.end = 5)
```



Synthetic catalogues generation

```r
# maximum likelihood estimator for beta
beta.p <- 1 / (mean(aquila.bru$magnitudes) - M0)
```

```r
synth.cat.list <- generate_temporal_ETAS_synthetic(
  theta = post.samp[1, ], # ETAS parameters
  beta.p = beta.p, # magnitude distribution parameter
  M0 = M0, # cutoff magnitude
  T1 = T1, # starting time
  T2 = T2, # end time
  Ht = aquila.bru[which.max(aquila.bru$magnitudes), ] # known events
)
# merge into unique data.frame
synth.cat.df <- do.call(rbind, synth.cat.list)
# order events by time
synth.cat.df <- synth.cat.df[order(synth.cat.df$ts), ]

ggplot(synth.cat.df, aes(ts, magnitudes, color = as.factor(gen))) +
  geom_point(size = 0.5)
```


magnitudes
6
5
4
3
0
100
200
300
ts
as.factor(gen)
−1
0
1
2
3
4
5