Forecast Accuracy and Precision Assessment Using Synthetic Catalogues

Author: Zhile Xu UUN: s2500393

The code below assesses the forecast accuracy and precision of ETAS models using synthetic earthquake catalogues. The analysis includes setting up the environment, generating synthetic data, fitting the ETAS model, and evaluating its performance under different scenarios.

Set up the environment

```
# Remove all objects from the current R environment to start fresh
rm(list = ls())
```

```
# Load the required libraries
library(ETAS.inlabru)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
set.seed(123)
```

```
# Set the number of cores to use for parallel processing
num.cores <- 12
future::plan(future::multisession, workers = num.cores)
INLA::inla.setOption(num.threads = num.cores)
```

Fit the model to the synthetic catalogue

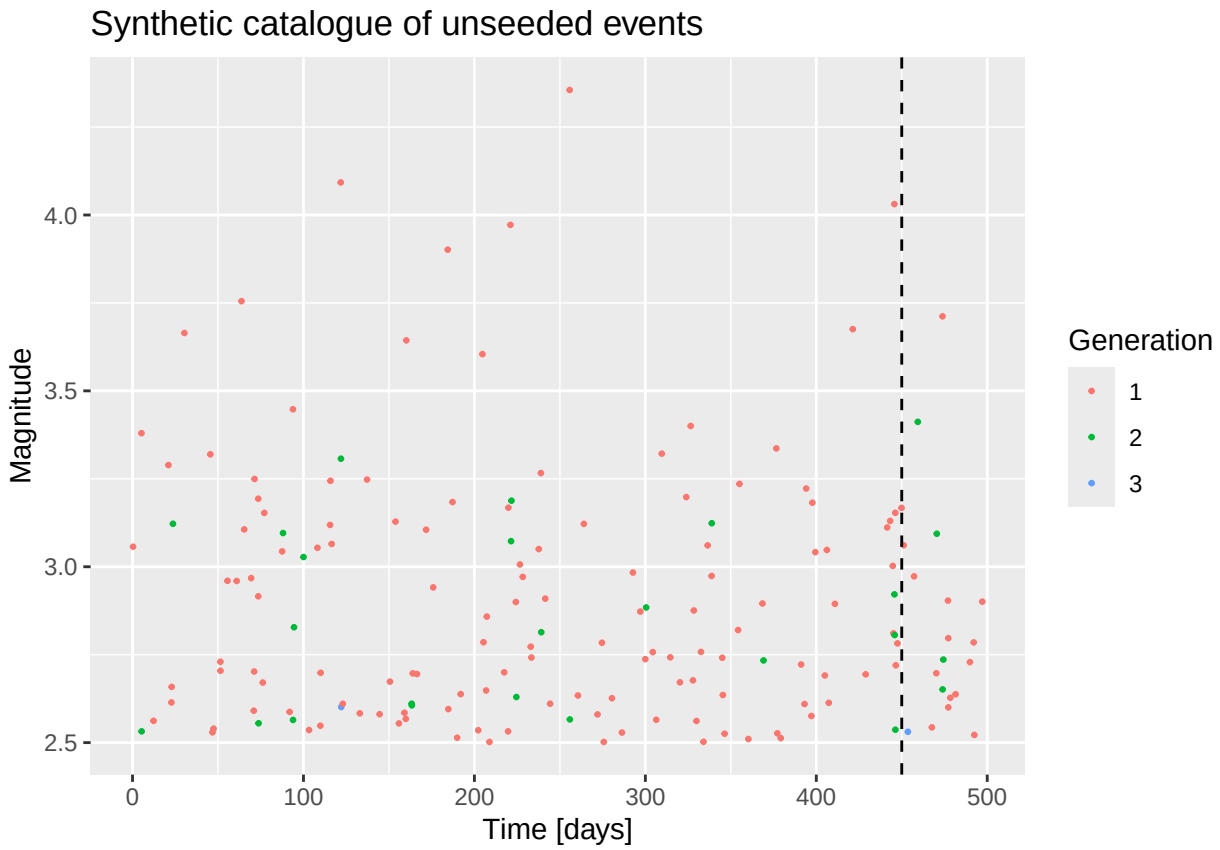
Generate the synthetic catalogue

```r
# set true ETAS parameters
true.param <- list(
  mu = 0.30106014, K = 0.13611399,
  alpha = 2.43945301, c = 0.07098607, p = 1.17838741
)
df.true.param <- data.frame(
  x = unlist(true.param),
  param = names(true.param)
)
# set magnitude distribution parameter
beta.p <- 2.353157
# set cutoff magnitude
M0 <- 2.5
# set starting time of the synthetic catalogue
T1 <- 0
# set end time of the synthetic catalogue
T2 <- 500
```

```r
synth.cat.list <- generate_temporal_ETAS_synthetic(
  theta = true.param,
  beta.p = beta.p,
  M0 = M0,
  T1 = T1,
  T2 = T2
)
# combine the synthetic catalogues
synth.cat.df <- do.call(rbind, synth.cat.list)
# order the synthetic catalogue by time
synth.cat.df <- synth.cat.df[order(synth.cat.df$ts), ]
# create a column with the index of the event
synth.cat.df$idx.p <- seq_len(nrow(synth.cat.df))
synth.unseeded <- synth.cat.df
```

Plot the synthetic catalogue

```r
plot.synth.unseeded <- ggplot(
  synth.unseeded,
  aes(ts, magnitudes, color = as.factor(gen))
) +
  geom_point(size = 0.5) +
  labs(x = "Time [days]", y = "Magnitude") +
  scale_color_discrete(name = "Generation") +
  labs(title = "Synthetic catalogue of unseeded events") +
  # add a vertical line at the time of the last event
  geom_vline(xintercept = 450, color = "black", linetype = "dashed")
plot.synth.unseeded
```



Set up the model

```
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)
)

# 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)
)
```

```
bru.opt.list <- list(
  bru_verbose = 0, # 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
```

Fit the model

```
T1.train <- 0
T2.train <- 450

# create a training dataset
synth.unseeded.train <- synth.unseeded[synth.unseeded$ts <= T2.train, ]

synth.fit <- Temporal.ETAS(
  total.data = synth.unseeded.train,
  M0 = M0,
  T1 = T1.train,
  T2 = T2.train,
  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  0.8159029
```

```
input_list <- list(
  model.fit = synth.fit,
  link.functions = link.f
)
```

Posterior sampling

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

Forecasting

```r
T1.fore <- 450 + 1 / (24 * 60)
T2.fore <- 500
Ht.fore <- synth.unseeded[synth.unseeded$ts <= T2.train, ]

# maximum likelihood estimator for beta
beta.p <- 1 / (mean(synth.unseeded$magnitudes) - M0)

fore.unseeded <- Temporal.ETAS.forecast(
  post.samp = post.samp,
  n.cat = nrow(post.samp),
  beta.p = beta.p,
  M0 = M0,
  T1 = T1.fore,
  T2 = T2.fore,
  Ht = Ht.fore
)
```

Plot the forecast

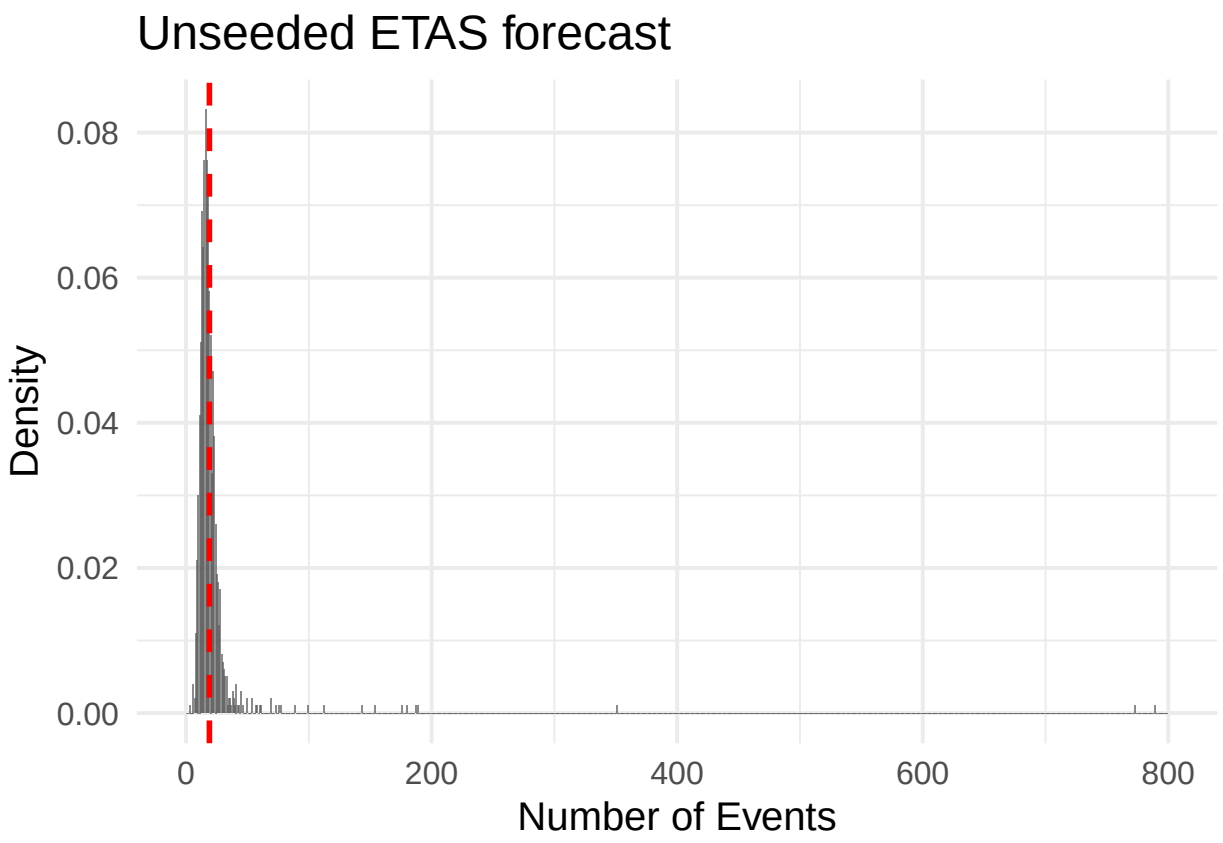
```r
# Find the number of events per catalogue
N.fore <- vapply(
  seq_len(fore.unseeded$n.cat),
  \(x) sum(fore.unseeded$fore.df$cat.idx == x), 0
)

# Find the number of observed events in the forecasting period
N.obs <- sum(synth.unseeded$ts >= T1.fore & synth.unseeded$ts <= T2.fore)

# Create the plot
plot.fore.unseeded <- ggplot() +
  geom_histogram(aes(x = N.fore, y = after_stat(density)),
    binwidth = 1, alpha = 0.7
  ) +
  geom_vline(
    xintercept = N.obs, color = "red",
    linetype = "dashed", size = 1
  ) +
  xlim(0, 800) +
  labs(
    title = "Unseeded ETAS forecast",
    x = "Number of Events",
    y = "Density"
  ) +
  theme_minimal(base_size = 15)
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Display the plot
plot.fore.unseeded
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



Fit the model to the seeded catalogue (seeded on day 400)

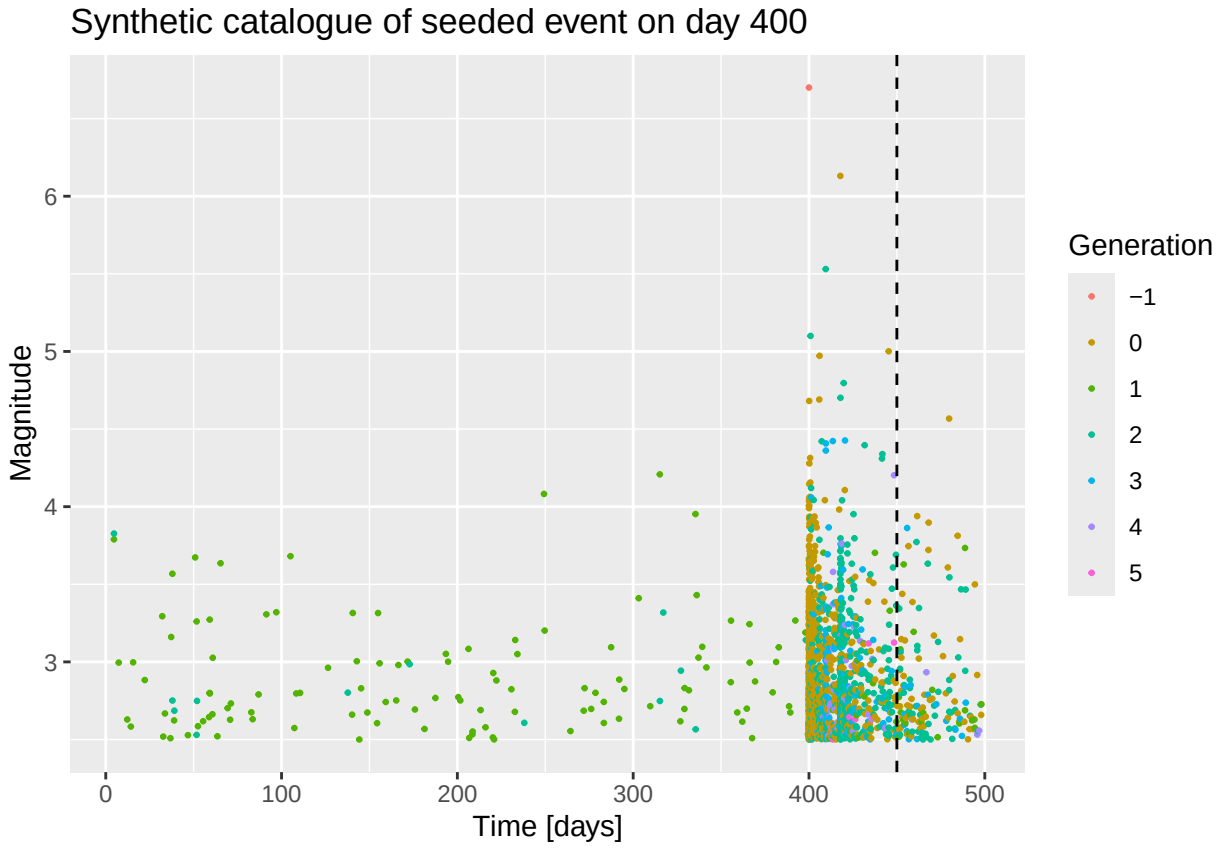
Generate the seeded catalogue

```
known.events.df <- data.frame(
  ts = c(400),
  magnitudes = c(6.7)
)

synth.seeded.400 <- generate_temporal_ETAS_synthetic(
  theta = true.param,
  beta.p = beta.p,
  M0 = M0,
  T1 = T1,
```

```
  T2 = T2,
  Ht = known.events.df
)

# combine the synthetic catalogues
synth.seeded.400.df <- do.call(rbind, synth.seeded.400)
# order the synthetic catalogue by time
synth.seeded.400.df <- synth.seeded.400.df[order(synth.seeded.400.df$ts), ]
# create a column with the index of the event
synth.seeded.400.df$idx.p <- seq_len(nrow(synth.seeded.400.df))
```

Plot the seeded catalogue

```
plot.synth.seeded.400 <- ggplot(
  synth.seeded.400.df,
  aes(ts, magnitudes, color = as.factor(gen))
) +
  geom_point(size = 0.5) +
  labs(x = "Time [days]", y = "Magnitude") +
  scale_color_discrete(name = "Generation") +
  labs(title = "Synthetic catalogue of seeded event on day 400") +
  # add a vertical line at the time of the last event
  geom_vline(xintercept = 450, color = "black", linetype = "dashed")
plot.synth.seeded.400
```



Fit the model

```
# create a training dataset
synth.seeded.400.df.train <-
  synth.seeded.400.df[synth.seeded.400.df$ts <= T2.train, ]

synth.fit.seeded.400 <- Temporal.ETAS(
  total.data = synth.seeded.400.df.train,
  M0 = M0,
  T1 = T1.train,
  T2 = T2.train,
  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  9.089671
```

```
input_list_seeded <- list(
  model.fit = synth.fit.seeded.400,
  link.functions = link.f
)
```

Posterior sampling

```
post.samp.seeded.400 <- post_sampling(
  input.list = input_list_seeded,
  n.samp = 1000,
  max.batch = 1000
)
```

Forecasting

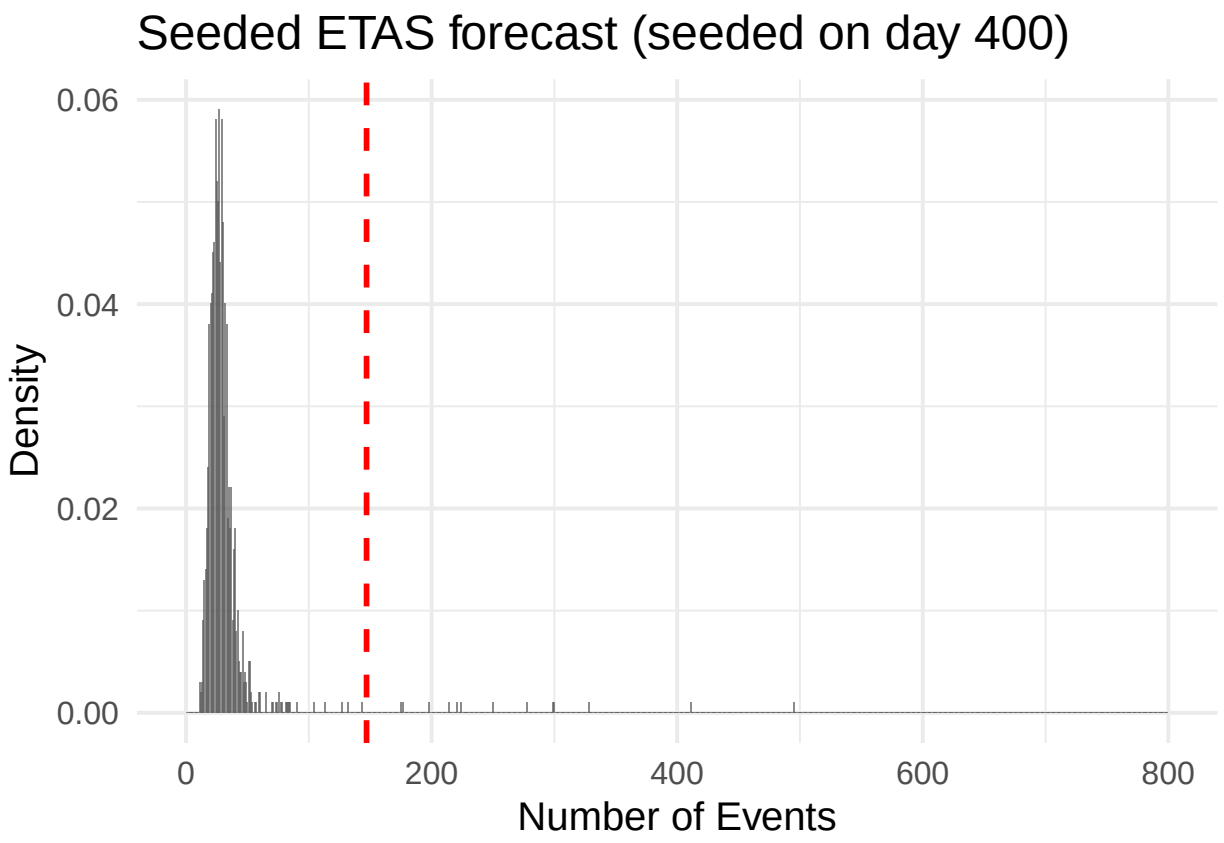
```
beta.p <- 1 / (mean(synth.seeded.400.df$magnitudes) - M0)

Ht.fore <- synth.seeded.400.df[synth.seeded.400.df$ts <= T2.train, ]

fore.seeded.400 <- Temporal.ETAS.forecast(
  post.samp = post.samp.seeded.400,
  n.cat = nrow(post.samp.seeded.400),
  beta.p = beta.p,
  M0 = M0,
  T1 = T1.fore,
  T2 = T2.fore,
  Ht = Ht.fore
)
```

Plot the forecast

```
# find number of events per catalogue
N.fore.seeded.400 <- vapply(
  seq_len(fore.seeded.400$n.cat),
  \(x) sum(fore.seeded.400$fore.df$cat.idx == x), 0
)
# find number of observed events in the forecasting period
N.obs <- sum(synth.seeded.400.df$ts >= T1.fore & synth.seeded.400.df$ts <= T2.fore)
# plot the distribution
plot.fore.seeded.400 <- ggplot() +
  geom_histogram(aes(x = N.fore.seeded.400, y = after_stat(density)),
    binwidth = 1, alpha = 0.7
  ) +
  geom_vline(xintercept = N.obs, color = "red", linetype = "dashed", size = 1) +
  xlim(0, 800) +
  labs(
    title = "Seeded ETAS forecast (seeded on day 400)",
    x = "Number of Events",
    y = "Density"
  ) +
  theme_minimal(base_size = 15)
plot.fore.seeded.400
```

```
## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



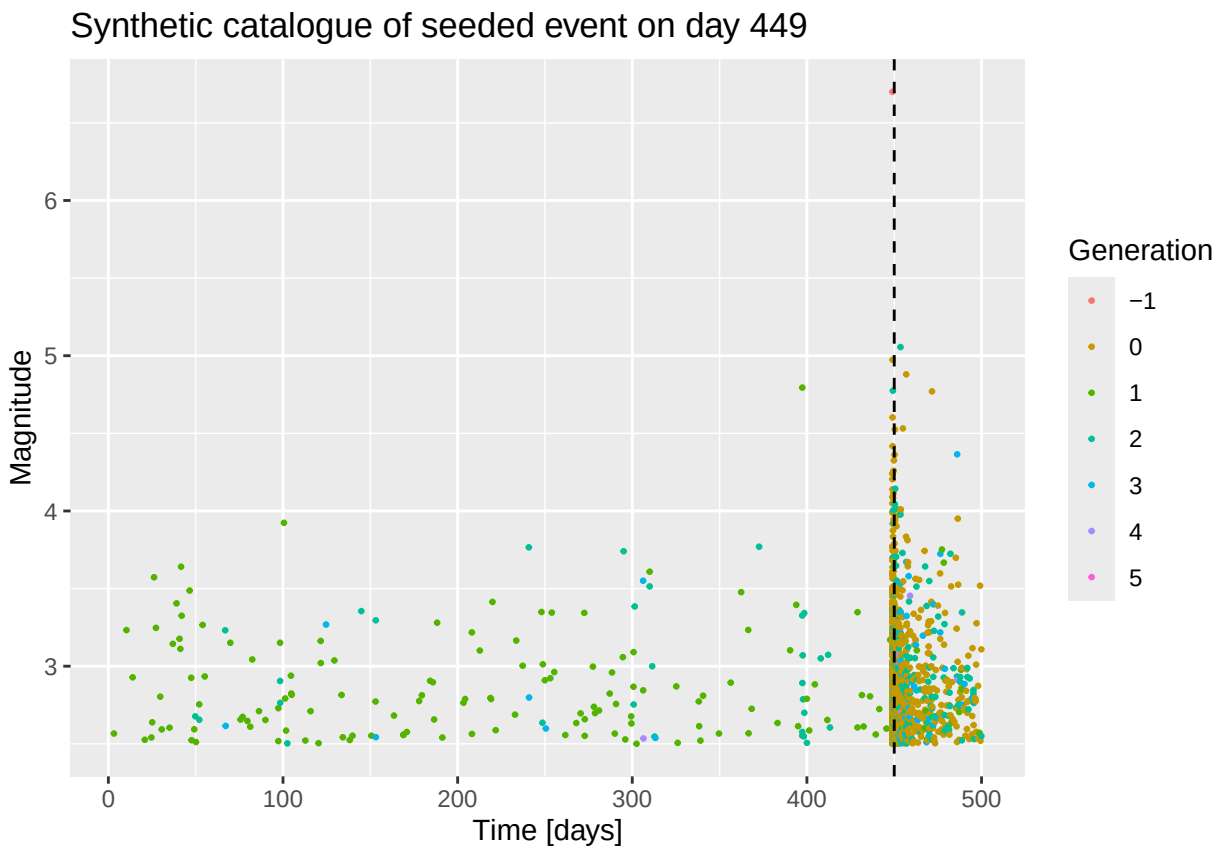
Fit the model to the seeded catalogue (seeded on day 449)

Generate the seeded catalogue

```
known.events.df <- data.frame(
  ts = c(449),
  magnitudes = c(6.7)
)

synth.seeded.449 <- generate_temporal_ETAS_synthetic(
  theta = true.param,
  beta.p = beta.p,
  M0 = M0,
  T1 = T1,
  T2 = T2,
  Ht = known.events.df
)

# combine the synthetic catalogues
synth.seeded.449.df <- do.call(rbind, synth.seeded.449)
# order the synthetic catalogue by time
synth.seeded.449.df <- synth.seeded.449.df[order(synth.seeded.449.df$ts), ]
# create a column with the index of the event
synth.seeded.449.df$idx.p <- seq_len(nrow(synth.seeded.449.df))
```

Plot the seeded catalogue

```
plot.synth.seeded.449 <- ggplot(
  synth.seeded.449.df,
  aes(ts, magnitudes, color = as.factor(gen))
) +
  geom_point(size = 0.5) +
  labs(x = "Time [days]", y = "Magnitude") +
  scale_color_discrete(name = "Generation") +
  labs(title = "Synthetic catalogue of seeded event on day 449") +
  # add a vertical line at the time of the last event
  geom_vline(xintercept = 450, color = "black", linetype = "dashed")
plot.synth.seeded.449
```



Fit the model

```
# create a training dataset
synth.seeded.449.df.train <- synth.seeded.449.df[synth.seeded.449.df$ts <= T2.train, ]

synth.fit.seeded.449 <- Temporal.ETAS(
  total.data = synth.seeded.449.df.train,
  M0 = M0,
  T1 = T1.train,
```

```
  T2 = T2.train,
  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  1.21876
```

```
input_list_seeded <- list(
  model.fit = synth.fit.seeded.449,
  link.functions = link.f
)
```

Posterior sampling

```
post.samp.seeded.449 <- post_sampling(
  input.list = input_list_seeded,
  n.samp = 1000,
  max.batch = 1000
)
```

Forecasting

```
beta.p <- 1 / (mean(synth.seeded.449.df$magnitudes) - M0)

Ht.fore <- synth.seeded.449.df[synth.seeded.449.df$ts <= T2.train, ]

fore.seeded.449 <- Temporal.ETAS.forecast(
  post.samp = post.samp.seeded.449,
  n.cat = nrow(post.samp.seeded.449),
  beta.p = beta.p,
  M0 = M0,
  T1 = T1.fore,
  T2 = T2.fore,
  Ht = Ht.fore
)
```

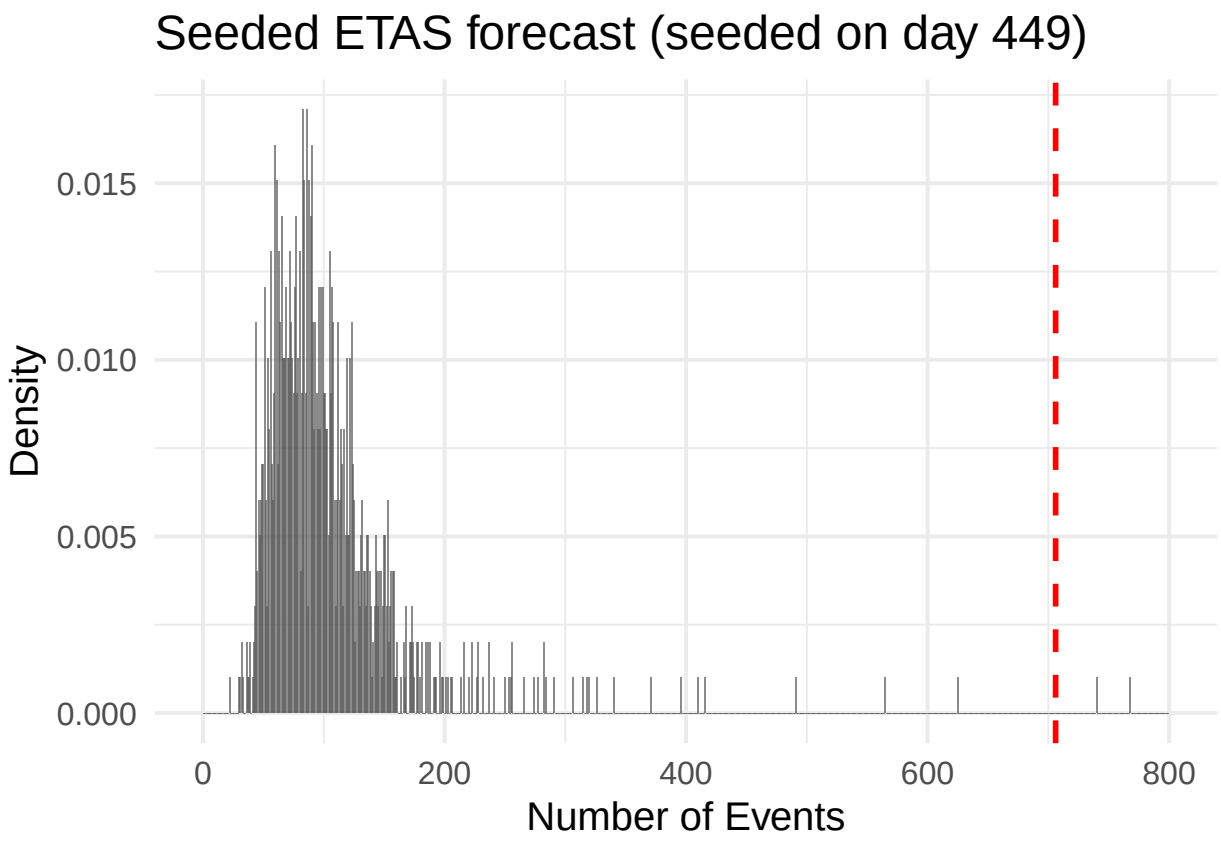
Plot the forecast

```
# find number of events per catalogue
N.fore.seeded.449 <- vapply(
  seq_len(fore.seeded.449$n.cat),
  \(x) sum(fore.seeded.449$fore.df$cat.idx == x), 0
)
```

```r
# find number of observed events in the forecasting period
N.obs <- sum(synth.seeded.449.df$ts >= T1.fore & synth.seeded.449.df$ts <= T2.fore)
# plot the distribution
plot.fore.seeded.449 <- ggplot() +
  geom_histogram(aes(x = N.fore.seeded.449, y = after_stat(density)),
    binwidth = 1, alpha = 0.7
  ) +
  geom_vline(xintercept = N.obs, color = "red", linetype = "dashed", size = 1) +
  xlim(0, 800) +
  labs(
    title = "Seeded ETAS forecast (seeded on day 449)",
    x = "Number of Events",
    y = "Density"
  ) +
  theme_minimal(base_size = 15)
plot.fore.seeded.449
```

```
## Warning: Removed 5 rows containing non-finite outside the scale range
## (‘stat_bin()‘).
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



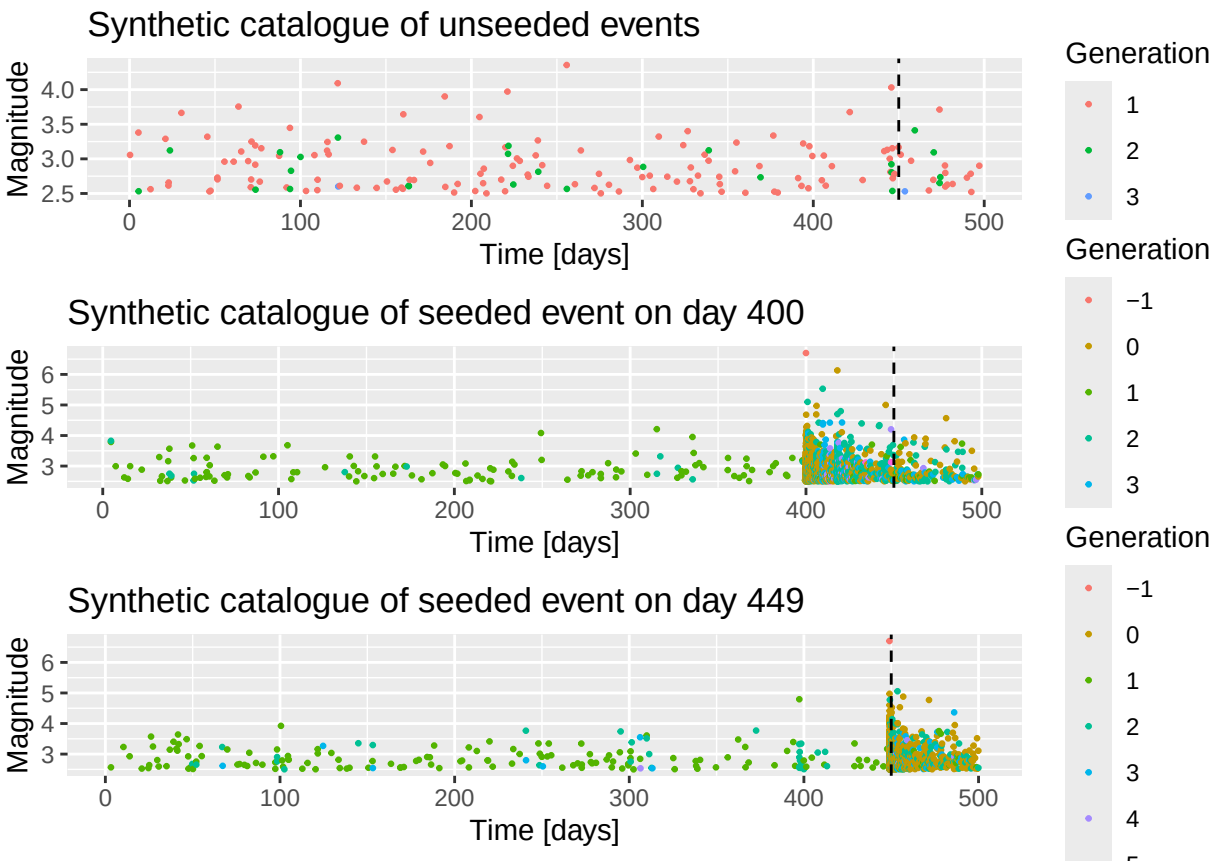
stack vertically the plots of the three synthetic catalogues

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
q4.synth.plot <- grid.arrange(plot.synth.unseeded,
  plot.synth.seeded.400,
  plot.synth.seeded.449,
  ncol = 1
)
```

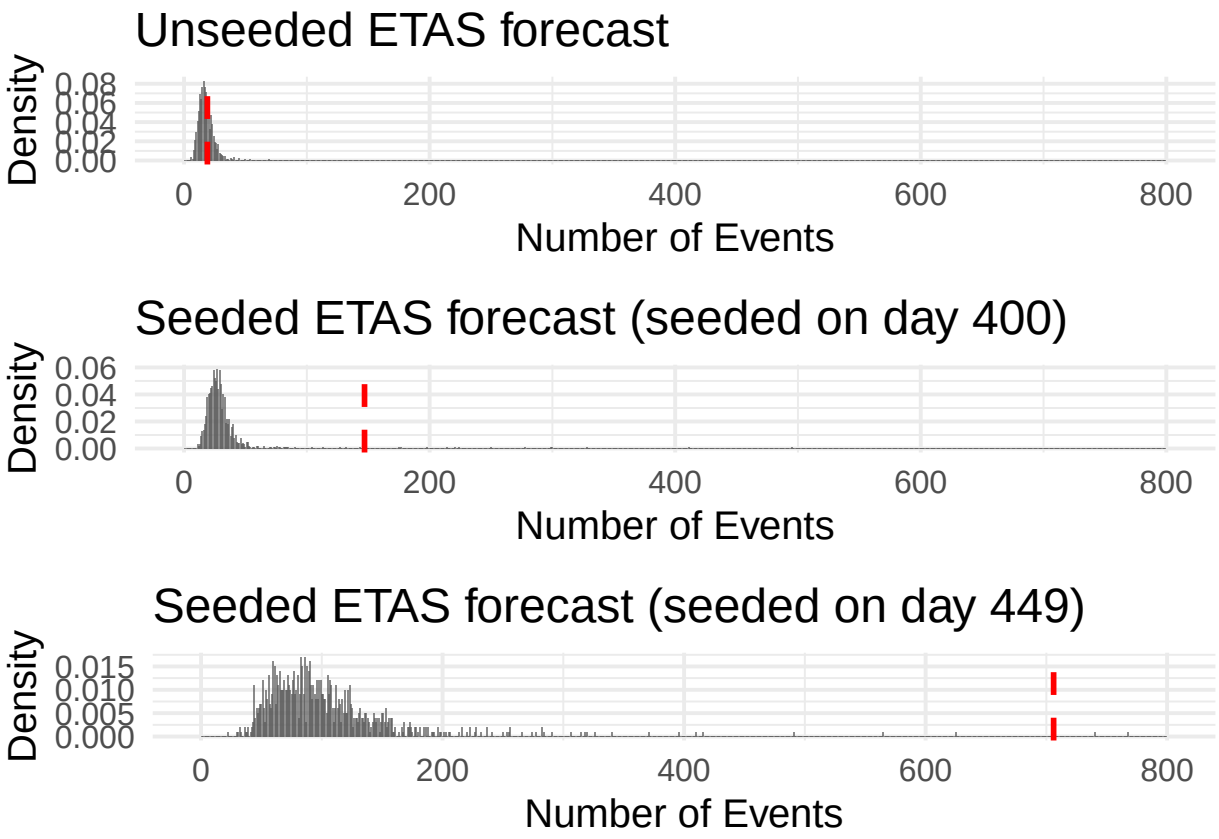



```
ggsave("q4-synthetic-catalogues.png", q4.synth.plot,
  width = 11.69, height = 8.27, units = "in"
)
```

stack vertically the plots of the three forecasts

```
q4.fore.plot <- grid.arrange(plot.fore.unseeded,
  plot.fore.seeded.400, plot.fore.seeded.449,
  ncol = 1
)
```

```
## Warning: Removed 2 rows containing non-finite outside the scale range
## (‘stat_bin()‘).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).

## Warning: Removed 1 row containing non-finite outside the scale range
## (‘stat_bin()‘).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).

## Warning: Removed 5 rows containing non-finite outside the scale range
## (‘stat_bin()‘).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (‘geom_bar()‘).
```



```r
ggsave("q4-forecasts.png", q4.fore.plot,
  width = 11.69, height = 8.27, units = "in"
)
```