

# Impact of Parameter Fixation on Posterior Distributions

Author: Zhile Xu

UUN: s2500393

The code below explores the impact of parameter fixation and mis-specification on the posterior distributions of ETAS model parameters using synthetic earthquake catalogues. The analysis includes setting up the environment, generating synthetic data, fitting the ETAS model, and evaluating its performance under various scenarios.

## Set up the environment

```
rm(list = ls())
```

```
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)
```

```
num.cores <- 12
future::plan(future::multisession, workers = num.cores)
INLA::inla.setOption(num.threads = num.cores)
```

## Generate a synthetic catalogue

```
# 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 <- 1000

# generate the synthetic catalogue
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$ts), ]
# create a column with the index of the event
synth.cat.df$id.p <- seq_len(nrow(synth.cat.df))
# create a column with the generation of the event
synth.cat.df$dgen <- case_when(
  synth.cat.df$dgen == 1 ~ "Background",
  synth.cat.df$dgen > 1 ~ "Triggered",
  .default = "Other"
)

```

Create a function to fix the parameters and return the posterior

```

get_fit_and_plot <- function(link.f, inv.link.f, th.init, title, fix = NULL) {
  # set up list of bru options
  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 <- Temporal.ETAS(
    total.data = synth.cat.df,
    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
  )

input.list <- list(model.fit = fit, link.functions = link.f)
post <- get_posterior_param(input.list = input.list)

df.true.param.filtered <- df.true.param
if (!is.null(fix)) {
  post$post.df <- post$post.df %>% filter(param != fix)
  df.true.param.filtered <- df.true.param %>% filter(param != fix)
}

ncol <- if (is.null(fix)) {
  3
} else {
  4
}
plot <- ggplot(post$post.df, aes(x = x, y = y)) +
  geom_line(size = 1) +
  facet_wrap(~param, scales = "free", ncol = ncol) +
  xlab("Parameter Value") +
  ylab("Density") +
  geom_vline(
    data = df.true.param.filtered,
    aes(xintercept = x), linetype = 2, color = "black", size = 1
  ) +
  theme_bw() +
  theme(
    text = element_text(size = 14),
    legend.position = "bottom",
    strip.background = element_rect(fill = "white", color = "black"),
  ) +
  ggtitle(title)

return(list(fit = fit, post = post, plot = plot))
}

```

**Without Fix the parameters and return the posterior**

```

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)
)

fit.without.fix <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Parameter Estimation Without Fixing Any Parameters"
)

```

```

## Start creating grid...
## Finished creating grid, time 0.496994

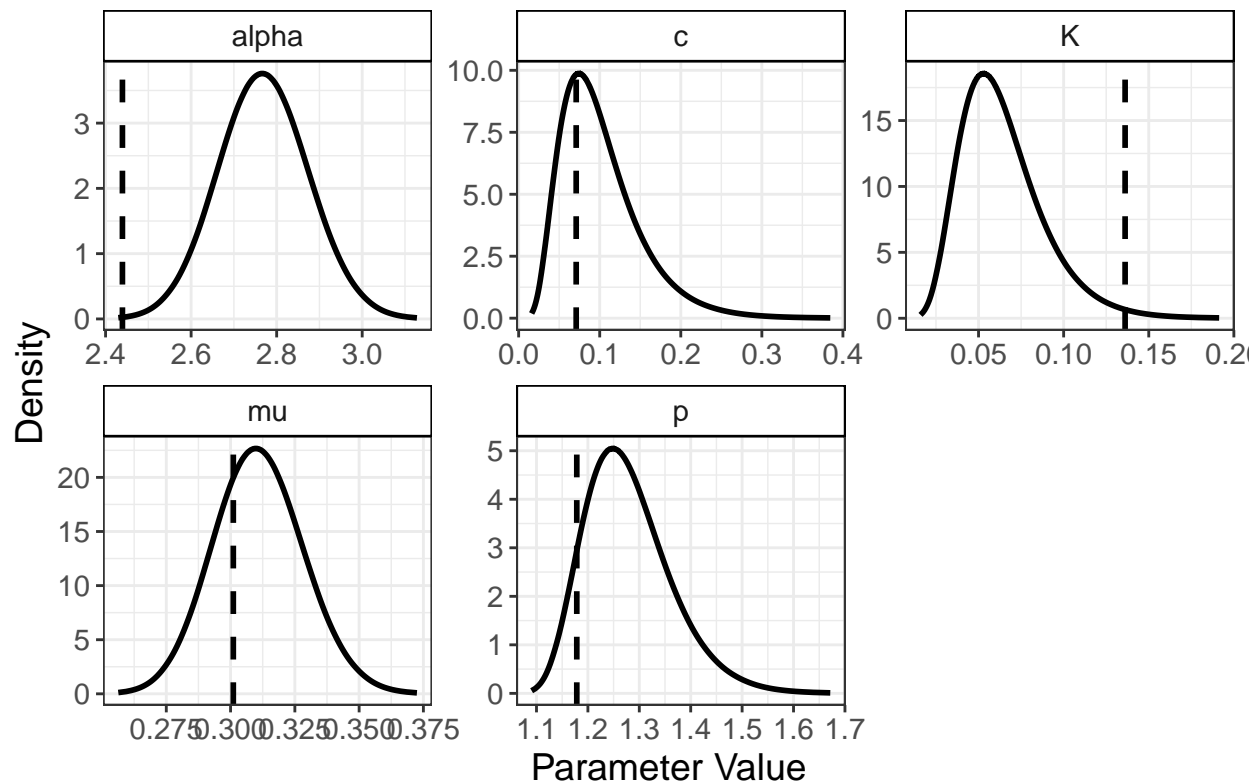
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.

```

Plot the posterior distribution without fixing any parameters

```
fit.without.fix$plot
```

## Parameter Estimation Without Fixing Any Parameters



```
ggsave("q3-without-fix.png", height = 8.27, width = 11.69, units = "in")
```

Fix  $\alpha$  to the true value and return the posterior

```
link.f <- list(
  mu = \(x) gamma_t(x, 0.3, 0.6),
  K = \(x) unif_t(x, 0, 10),
  alpha = \(x) unif_t(x, true.param$alpha, true.param$alpha),
  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, true.param$alpha - (1e-8), true.param$alpha + (1e-8)),
  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$alpha(true.param$alpha),
th.c = inv.link.f$c_(0.1),
th.p = inv.link.f$p(1.1)
)

fit.fix.alpha <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Parameter alpha Fixed at True Value",
  fix = "alpha"
)

```

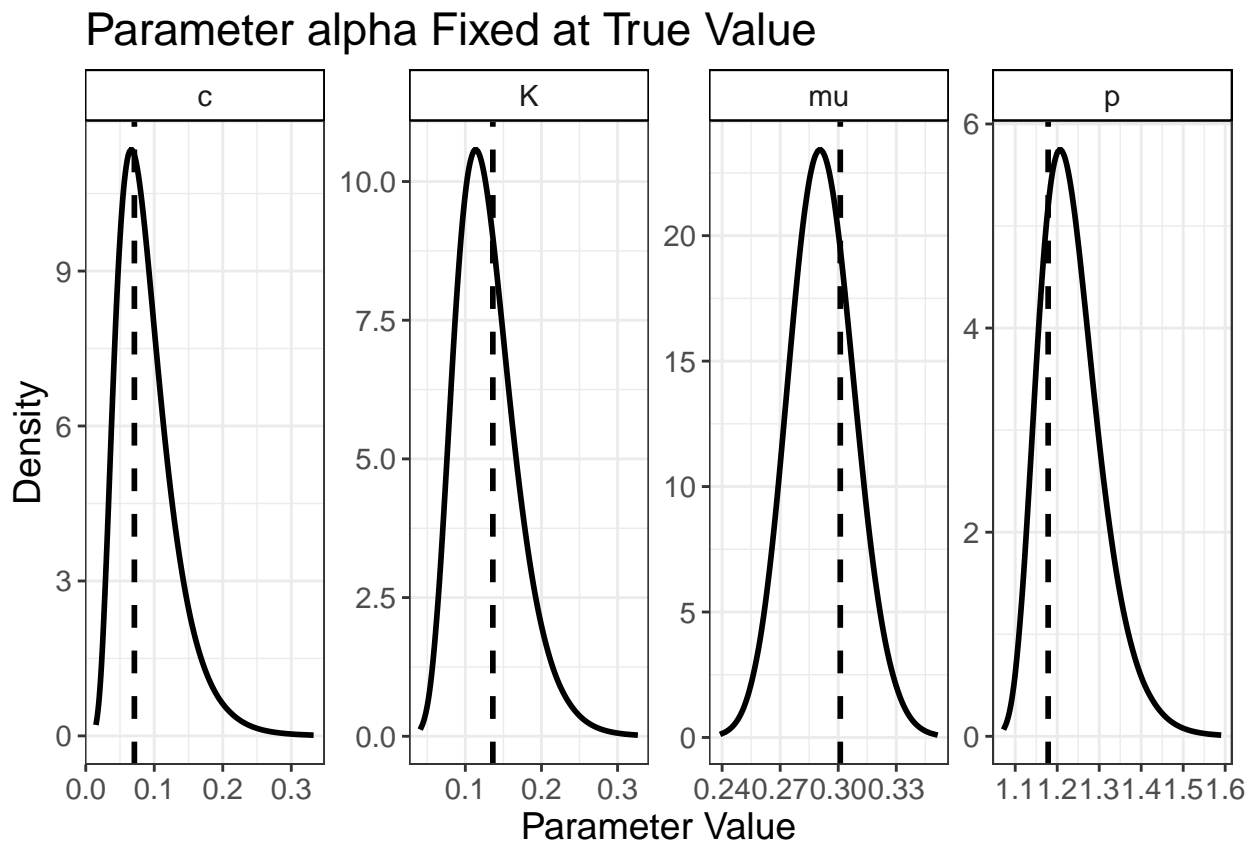
```

## Start creating grid...
## Finished creating grid, time 0.538126

```

Plot the posterior distribution with alpha fixed at the true value

```
fit.fix.alpha$plot
```



Fix  $c$  to the true value and return the posterior

```

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, true.param$c - (1e-8), true.param$c + (1e-8)),
  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, true.param$c - (1e-8), true.param$c + (1e-8)),
  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_(true.param$c),
  th.p = inv.link.f$p(1.1)
)

fit.fix.c <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Parameter c Fixed at True Value",
  fix = "c"
)

```

```

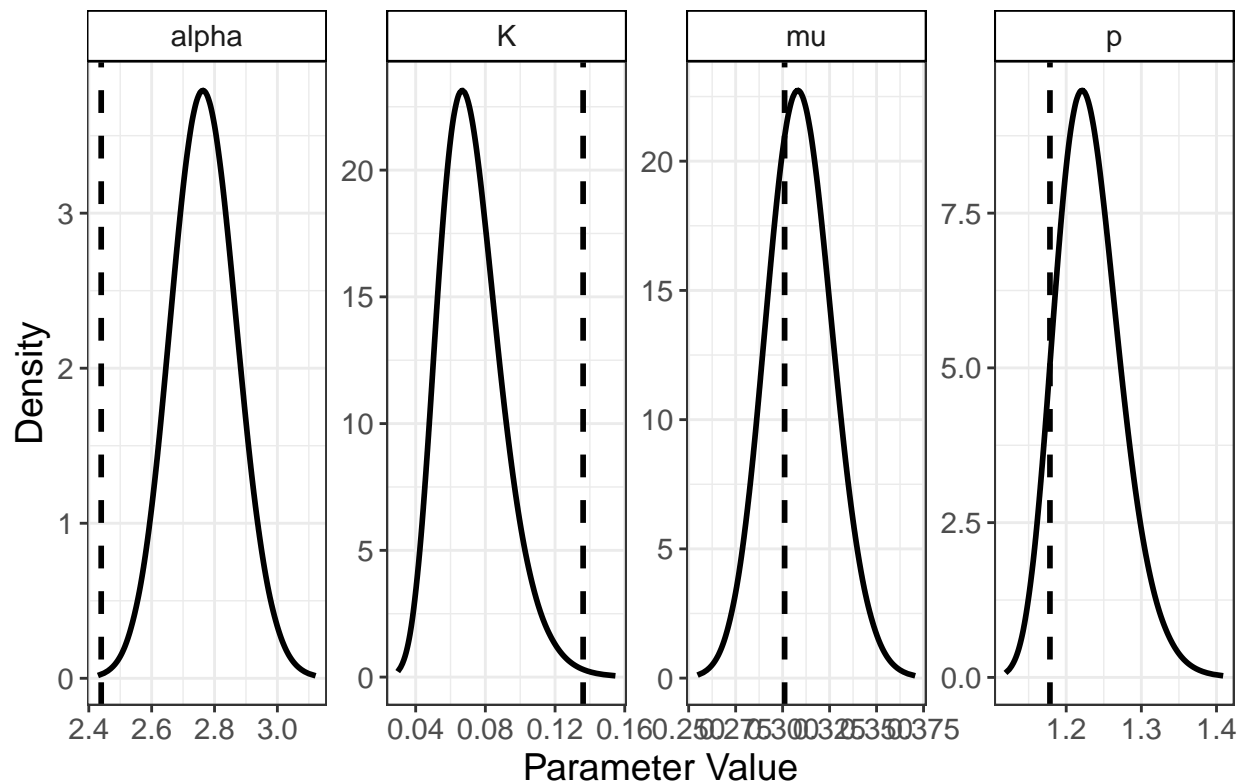
## Start creating grid...
## Finished creating grid, time 0.493686

```

Plot the posterior distribution with c fixed at the true value

```
fit.fix.c$plot
```

## Parameter c Fixed at True Value



Fix K to the true value and return the posterior

```
link.f <- list(
  mu = \(x) gamma_t(x, 0.3, 0.6),
  K = \(x) unif_t(x, true.param$K - (1e-8), true.param$K + (1e-8)),
  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, true.param$K - (1e-8), true.param$K + (1e-8)),
  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(true.param$K),
  th.alpha = inv.link.f$alpha(1),
  th.c = inv.link.f$c_(0.1),
```



```

    th.p = inv.link.f$p(1.1)
  )

fit.fix.K <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Parameter K Fixed at True Value",
  fix = "K"
)

```

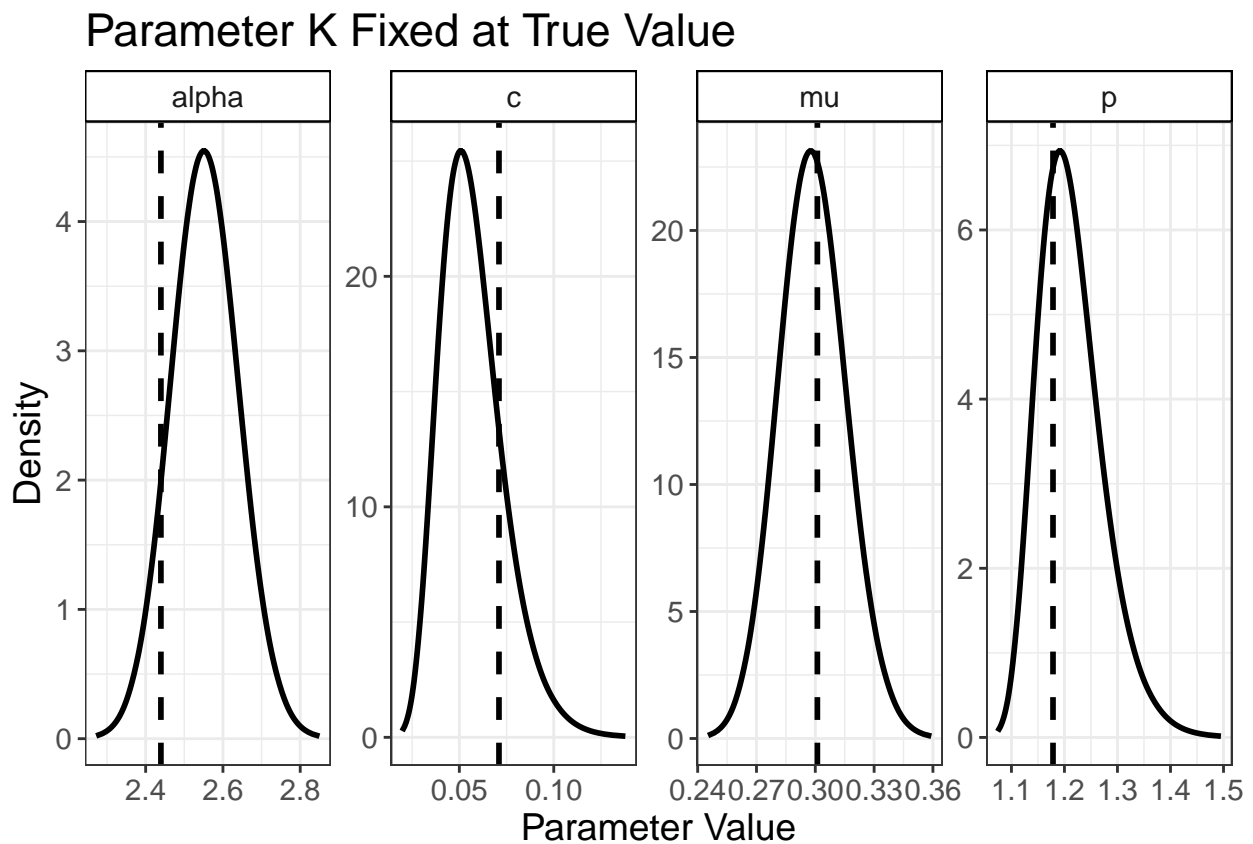
```

## Start creating grid...
## Finished creating grid, time  0.4798069

```

Plot the posterior distribution with K fixed at the true value

```
fit.fix.K$plot
```



Fix mu to the true value and return the posterior

```

link.f <- list(
  mu = \(x) gamma_t(x, true.param$mu - (1e-8), true.param$mu + (1e-8)),

```

```

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, true.param$mu - (1e-8), true.param$mu + (1e-8)),
  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$mu(true.param$mu),
  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)
)

fit.fix.mu <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Parameter mu Fixed at True Value",
  fix = "mu"
)

```

```

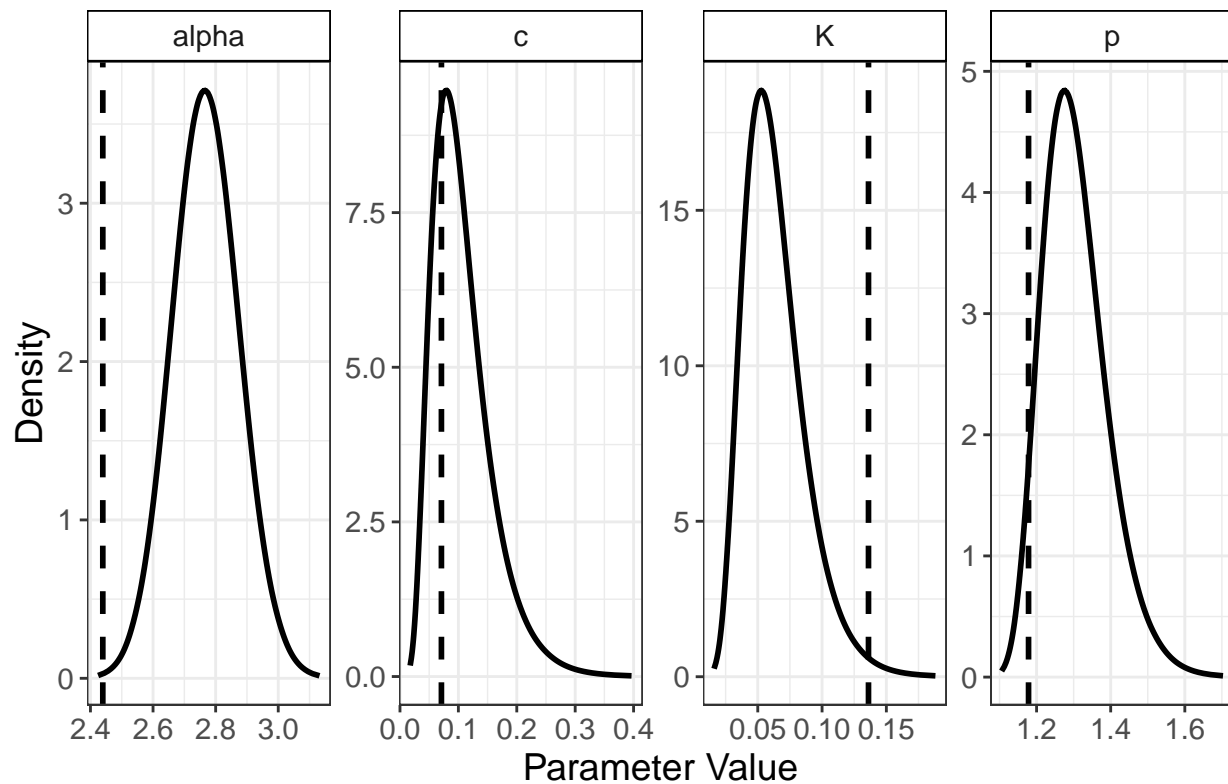
## Start creating grid...
## Finished creating grid, time 0.537791

```

Plot the posterior distribution with mu fixed at the true value

```
fit.fix.mu$plot
```

## Parameter mu Fixed at True Value



Fix p to the true value and return the posterior

```
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, true.param$p - (1e-8), true.param$p + (1e-8))
)

# 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, true.param$p - (1e-8), true.param$p + (1e-8))
)

# 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$p(true.param$p)
)

fit.fix.p <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Parameter p Fixed at True Value",
  fix = "p"
)

```

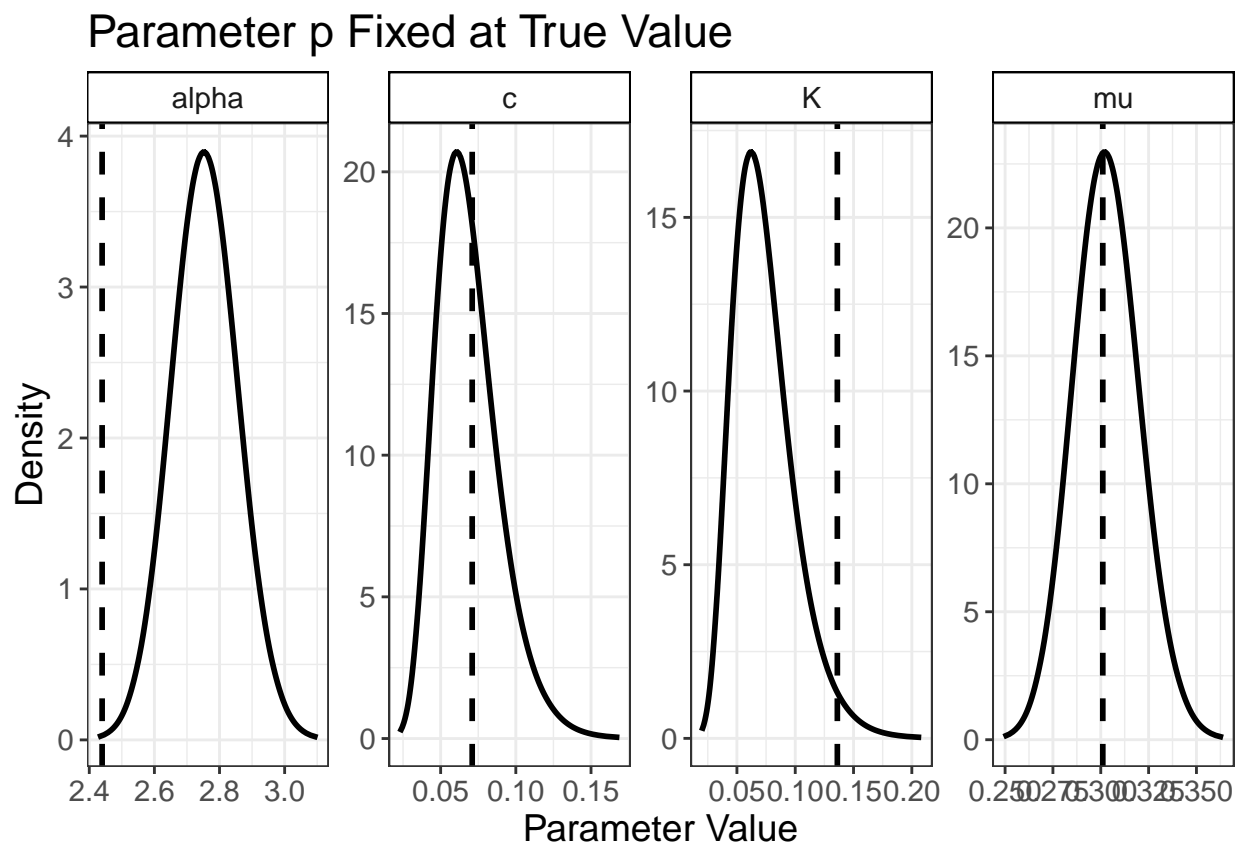
```

## Start creating grid...
## Finished creating grid, time 1.476572

```

Plot the posterior distribution with p fixed at the true value

```
fit.fix.p$plot
```



Stack the plots of fixing the parameters

```
library(gridExtra)
```

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

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

```
plot.true <- grid.arrange(
  fit.fix.alpha$plot,
  fit.fix.c$plot,
  fit.fix.K$plot,
  fit.fix.mu$plot,
  fit.fix.p$plot,
  ncol = 1
);
```

```
# plot.true
# save the plot
ggsave("q3-fix-the-parameter.png", plot.true,
  width = 8.27,
  height = 11.69, units = "in"
)
```

Mis-specify alpha to 3.0 and return the posterior

```
link.f <- list(
  mu = \ (x) gamma_t(x, 0.3, 0.6),
  K = \ (x) unif_t(x, 0, 10),
  alpha = \ (x) unif_t(x, 3 - 1e-8, 3 + 1e-8),
  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, 3 - 1e-8, 3 + 1e-8),
  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$alpha(3),
  th.c = inv.link.f$c_(0.1),
  th.p = inv.link.f$p(1.1)
)
```

```
fit.mis.alpha <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Posterior Distribution with Mis-specified alpha",
  fix = "alpha"
)
```

```
## Start creating grid...
## Finished creating grid, time 1.847243
```

Plot the posterior distribution with alpha mis-specified

```
fit.mis.alpha$plot <- fit.mis.alpha$plot +
  labs(subtitle = "True alpha = 2.439, Mis-specified alpha = 3.0")
```

Mis-specify K to 0.3 and return the posterior

```
link.f <- list(
  mu = \ (x) gamma_t(x, 0.3, 0.6),
  K = \ (x) unif_t(x, 0.3 - 1e-8, 0.3 + 1e-8),
  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, 0.3 - 1e-8, 0.3 + 1e-8),
  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(0.3),
  th.alpha = inv.link.f$alpha(1),
  th.c = inv.link.f$c_(0.1),
  th.p = inv.link.f$p(1.1)
)

fit.mis.K <- get_fit_and_plot(
  link.f = link.f,
  inv.link.f = inv.link.f,
  th.init = th.init,
  title = "Posterior Distribution with Mis-specified K",
```

```

fix = "K"
)

## Start creating grid...
## Finished creating grid, time 1.985029

# add subtitle to tell the true value of K and the mis-specified value
fit.mis.K$plot <- fit.mis.K$plot +
  labs(subtitle = "True K = 0.136, Mis-specified K = 0.3")

```

Stack the plots of mis-specifying the parameters

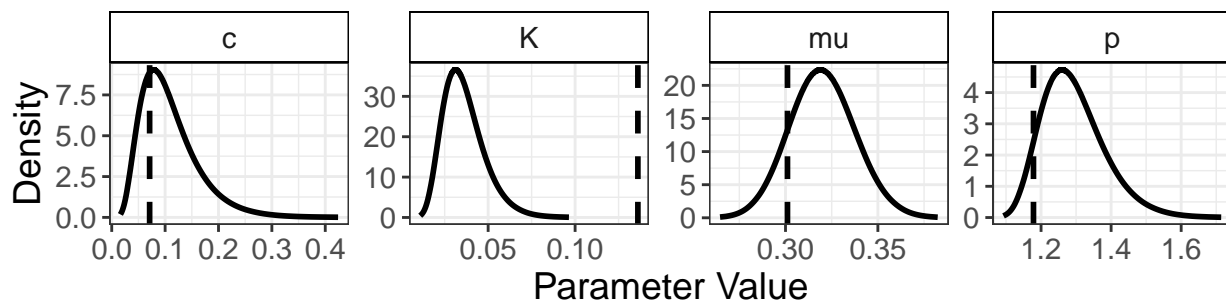
```

plot.mis <- grid.arrange(
  fit.mis.alpha$plot,
  fit.mis.K$plot,
  ncol = 1
)

```

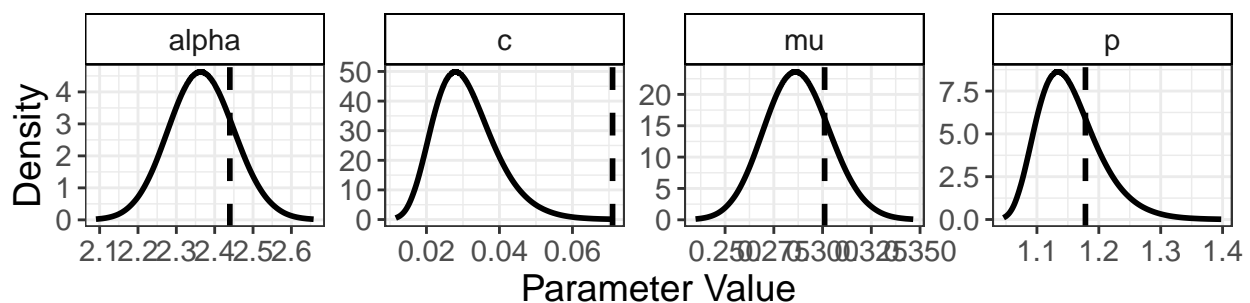
## Posterior Distribution with Mis-specified alpha

True alpha = 2.439, Mis-specified alpha = 3.0



## Posterior Distribution with Mis-specified K

True K = 0.136, Mis-specified K = 0.3



```
plot.mis
```

```

## TableGrob (2 x 1) "arrange": 2 grobs
##   z      cells  name      grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (2-2,1-1) arrange gtable[layout]

```

```
# save the plot
ggsave("q3-mis-specify-the-parameter.png",
  plot.mis,
  width = 11.69, height = 8.27, units = "in"
)
```