# PBSR Assignment 2

Shreyam Banerjee, Dipanjoy Saha, Richik Chakraborty

2022-11-17

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

## Question 2

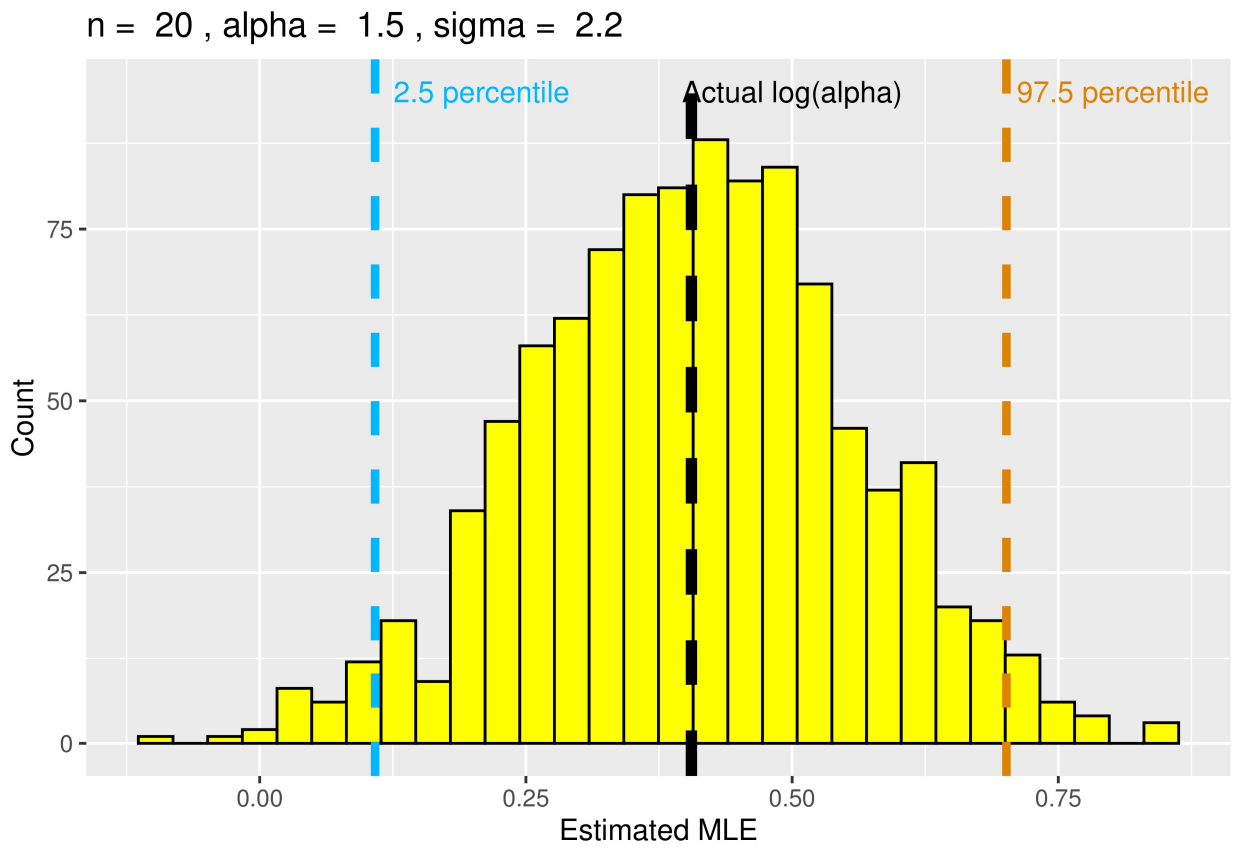
```
mle <- function(log_alpha, data, sigma) {
  l = sum(log(dgamma(data, shape = exp(log_alpha), scale = sigma)))
  return(-l)
}
MyMLE <- function(data, sigma) {
  log_alpha_initial <- log(mean(data)^2/var(data))
  estimator <- optim(log_alpha_initial,
                     mle,
                     data = data,
                     sigma = sigma)
  log_alpha_hat <- estimator$par
  return(log_alpha_hat)
}
```

```
get_estimates <- function(n, alpha, sigma) {
  estimates <- c()
  for (i in 1:1000) {
    samples <- rgamma(n, shape = alpha, scale = sigma)
    estimates <- append(estimates, MyMLE(data = samples, sigma = sigma))
  }
  return(estimates)
}
```

```
n = 20
alpha = 1.5
sigma = 2.2
```

```
estimated_mle <- tibble(get_estimates(n = n, alpha = alpha, sigma = sigma))
colnames(estimated_mle) <- c("estimate")
perc_2.5 <- quantile(estimated_mle$estimate, probs = 0.025, names = FALSE)
perc_97.5 <- quantile(estimated_mle$estimate, probs = 0.975, names = FALSE)
estimated_mle %>%
  ggplot(aes(estimate)) +
  geom_histogram(color = "black", fill = "yellow") +
  geom_vline(xintercept = log(alpha),
             size = 2,
             linetype = "dashed") +
  annotate("text", label = "Actual log(alpha)", x = 0.5, y = 95, color = "black") +
  geom_vline(xintercept = perc_2.5,
             color = "#00B9FF", size = 1.5, linetype = "dashed") +
  annotate("text", label = "2.5 percentile", x = perc_2.5 + 0.1, y = 95, color = "#00B9FF") +
  geom_vline(xintercept = perc_97.5,
             color = "#E08304", size = 1.5, linetype = "dashed") +
  annotate("text", label = "97.5 percentile", x = perc_97.5 + 0.1, y = 95, color = "#E08304") +
  labs(title = paste("n = ", n, ", alpha = ", alpha, ", sigma = ", sigma),
       x = "Estimated MLE",
       y = "Count")
```

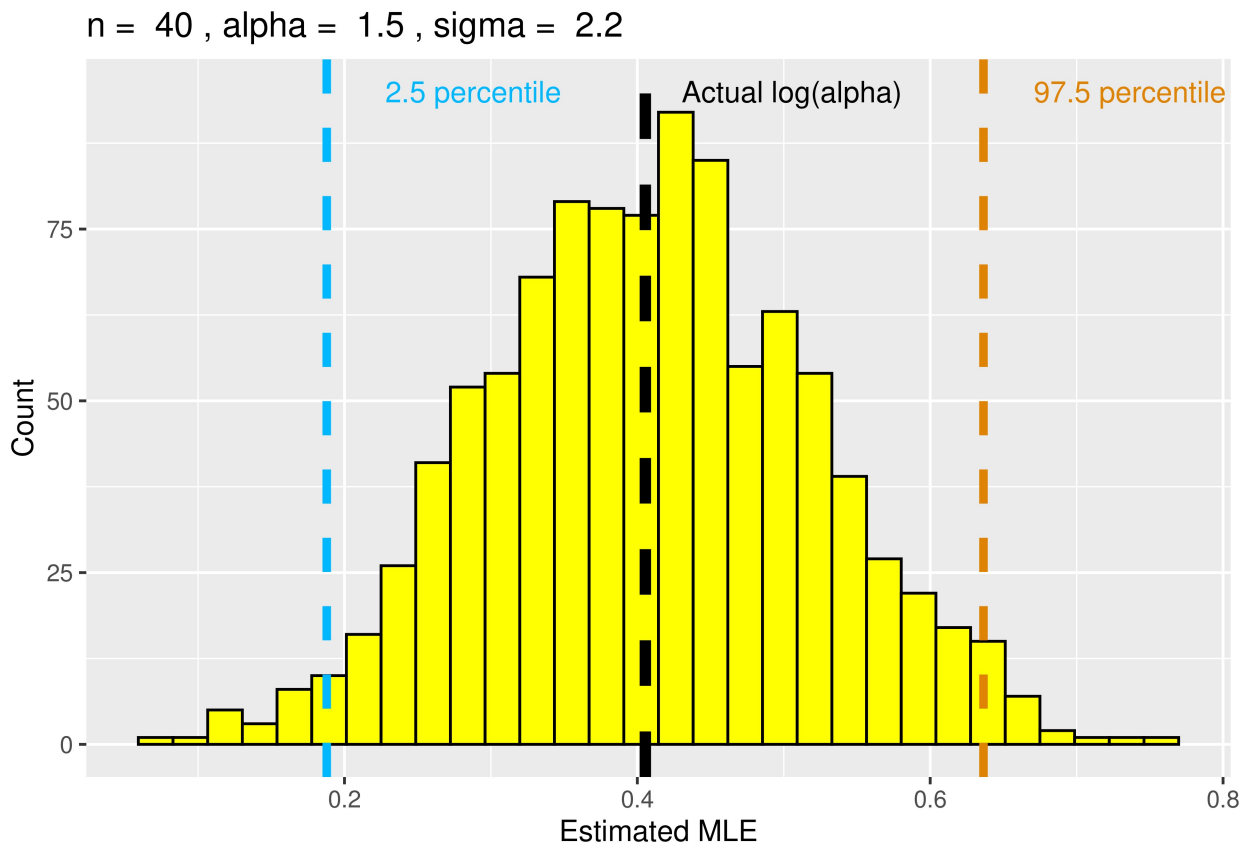
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
diff_20 <- perc_97.5 - perc_2.5
```

```r
n = 40
alpha = 1.5
sigma = 2.2
estimated_mle <- tibble(get_estimates(n = n, alpha = alpha, sigma = sigma))
colnames(estimated_mle) <- c("estimate")
perc_2.5 <- quantile(estimated_mle$estimate, probs = 0.025, names = FALSE)
perc_97.5 <- quantile(estimated_mle$estimate, probs = 0.975, names = FALSE)
estimated_mle %>%
  ggplot(aes(estimate)) +
  geom_histogram(color = "black", fill = "yellow") +
  geom_vline(xintercept = log(alpha),
             size = 2,
             linetype = "dashed") +
  annotate("text", label = "Actual log(alpha)", x = log(alpha) + 0.1, y = 95, color = "black") +
  geom_vline(xintercept = perc_2.5,
             color = "#00B9FF", size = 1.5, linetype = "dashed") +
  annotate("text", label = "2.5 percentile", x = perc_2.5 + 0.1, y = 95, color = "#00B9FF") +
  geom_vline(xintercept = perc_97.5,
             color = "#E08304", size = 1.5, linetype = "dashed") +
  annotate("text", label = "97.5 percentile", x = perc_97.5 + 0.1, y = 95, color = "#E08304") +
  labs(title = paste("n = ", n, ", alpha = ", alpha, ", sigma = ", sigma),
       x = "Estimated MLE",
       y = "Count")
```

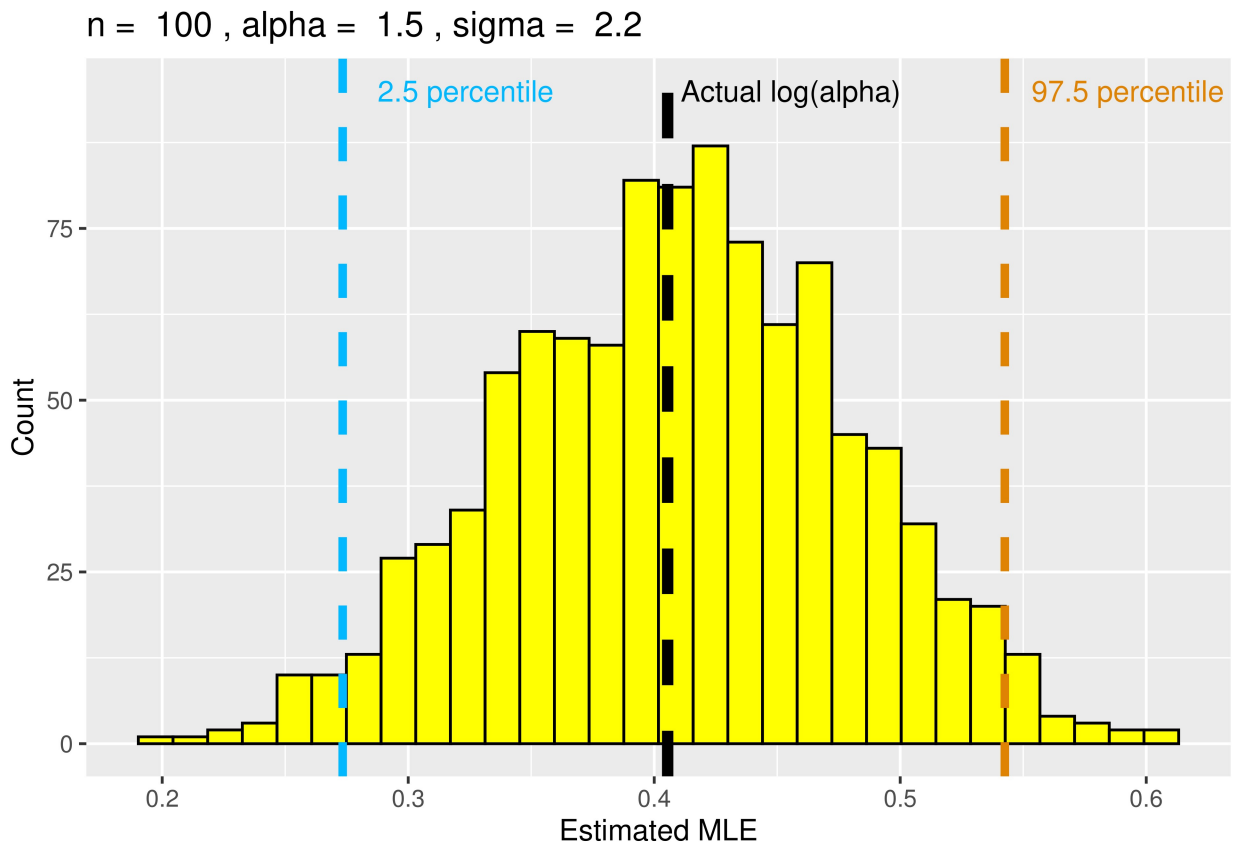
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
diff_40 <- perc_97.5 - perc_2.5
```

```
n = 100
alpha = 1.5
sigma = 2.2
estimated_mle <- tibble(get_estimates(n = n, alpha = alpha, sigma = sigma))
colnames(estimated_mle) <- c("estimate")
perc_2.5 <- quantile(estimated_mle$estimate, probs = 0.025, names = FALSE)
perc_97.5 <- quantile(estimated_mle$estimate, probs = 0.975, names = FALSE)
estimated_mle %>%
  ggplot(aes(estimate)) +
  geom_histogram(color = "black", fill = "yellow") +
  geom_vline(xintercept = log(alpha),
             size = 2,
             linetype = "dashed") +
  annotate("text", label = "Actual log(alpha)", x = log(alpha) + 0.05, y = 95, color = "black") +
  geom_vline(xintercept = perc_2.5,
             color = "#00B9FF", size = 1.5, linetype = "dashed") +
  annotate("text", label = "2.5 percentile", x = perc_2.5 + 0.05, y = 95, color = "#00B9FF") +
  geom_vline(xintercept = perc_97.5,
             color = "#E08304", size = 1.5, linetype = "dashed") +
  annotate("text", label = "97.5 percentile", x = perc_97.5 + 0.05, y = 95, color = "#E08304") +
  labs(title = paste("n = ", n, ", alpha = ", alpha, ", sigma = ", sigma),
       x = "Estimated MLE",
       y = "Count")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
diff_100 <- perc_97.5 - perc_2.5
```

```
diff_20
```

```
## [1] 0.5929186
```

```
diff_40
```

```
## [1] 0.4485025
```

```
diff_100
```

```
## [1] 0.2691561
```

Thus we can see from the the plot that as the sample size increases, the gap between the percentile points keep on decreasing.

## Problem 3

```r
library(tidyverse)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
data_q3 <- faithful %>%
  as_tibble()
x <- sort(data_q3$waiting)
# hist(x, xlab = 'waiting', probability = T, col='pink', main='')
```

comparing 3 models

```r
# model 1
p <- length(x[x<65])/length(x)
as <- mean(x[x<65])
ass <- var(x[x<65])
s <- ass/as
a <- as/s
mu <- mean(x[x>=65])
sigma <- sd(x[x>=65])
theta_inital <- c(p, a, s, mu, sigma)
neg_log_likelihood <- function(theta, data){
  n = length(data)

  p = theta[1]
  a = theta[2]
  s = theta[3]
  mu = theta[4]
  sigma = theta[5]

  l = 0
  for (i in 1:n) {
    l = l + log(p*dgamma(data[i], shape = a, scale = s) + (1-p)*dnorm(data[i], mean = mu, sd = sigma))
  }
  return(-l)
}
fit = optim(theta_inital,
            neg_log_likelihood,
            data = x,
            control = list(maxit = 1500),
            lower = c(0, 0, 0, -Inf, 0),
            upper = c(1, Inf, Inf, Inf, Inf),
```

```
            method="L-BFGS-B")
theta_1 = fit$par
theta_1
```

```
## [1]   0.3574036 101.4861312   0.5371140  80.0176038   5.9487453
```

```
p = theta_1[1]
a = theta_1[2]
s = theta_1[3]
mu = theta_1[4]
sigma = theta_1[5]
model_1 = p*dgamma(x, shape = a, scale = s) + (1-p)*dnorm(x, mean = mu, sd = sigma)
aic_1 <- 2*5 + 2*neg_log_likelihood(theta_1, x)
# hist(x, xlab = 'waiting', probability = T, col='pink', main='')
# lines(x, model_1)
```

```
# model 2
p <- length(x[x<65])/length(x)
as_1 <- mean(x[x<65])
ass_1 <- var(x[x<65])
s_1 <- ass_1/as_1
a_1 <- as_1/s_1
as_2 <- mean(x[x>=65])
ass_2 <- var(x[x>=65])
s_2 <- ass_2/as_2
a_2 <- as_2/s_2
theta_inital <- c(p, a_1, s_1, a_2, s_2)
neg_log_likelihood <- function(theta, data){
  n <- length(data)

  p <- theta[1]
  a_1 <- theta[2]
  s_1 <- theta[3]
  a_2 <- theta[4]
  s_2 <- theta[5]

  l <- 0
  for (i in 1:n) {
    l = l + log(p*dgamma(data[i], shape = a_1, scale = s_1) + (1-p)*dgamma(data[i], shape = a_2, scale =
  }
  return(-l)
}
fit = optim(theta_inital,
            neg_log_likelihood,
            data = x,
            control = list(maxit = 1500),
            lower = c(0, 0, 0, 0, 0),
            upper = c(1, Inf, Inf, Inf, Inf),
            method="L-BFGS-B")
theta_2 <- fit$par
theta_2
```

```
## [1]   0.3582592 101.5126436   0.5371146 169.2757878   0.4728250
```

```r
p <- theta_2[1]
a_1 <- theta_2[2]
s_1 <- theta_2[3]
a_2 <- theta_2[4]
s_2 <- theta_2[5]
model_2 <- p*dgamma(x, shape = a_1, scale = s_1) + (1-p)*dgamma(x, shape = a_2, scale = s_2)
aic_2 <- 2*5 + 2*neg_log_likelihood(theta_2, x)
# hist(x, xlab = 'waiting', probability = T, col='pink', main='')
# lines(x, model_2)
```

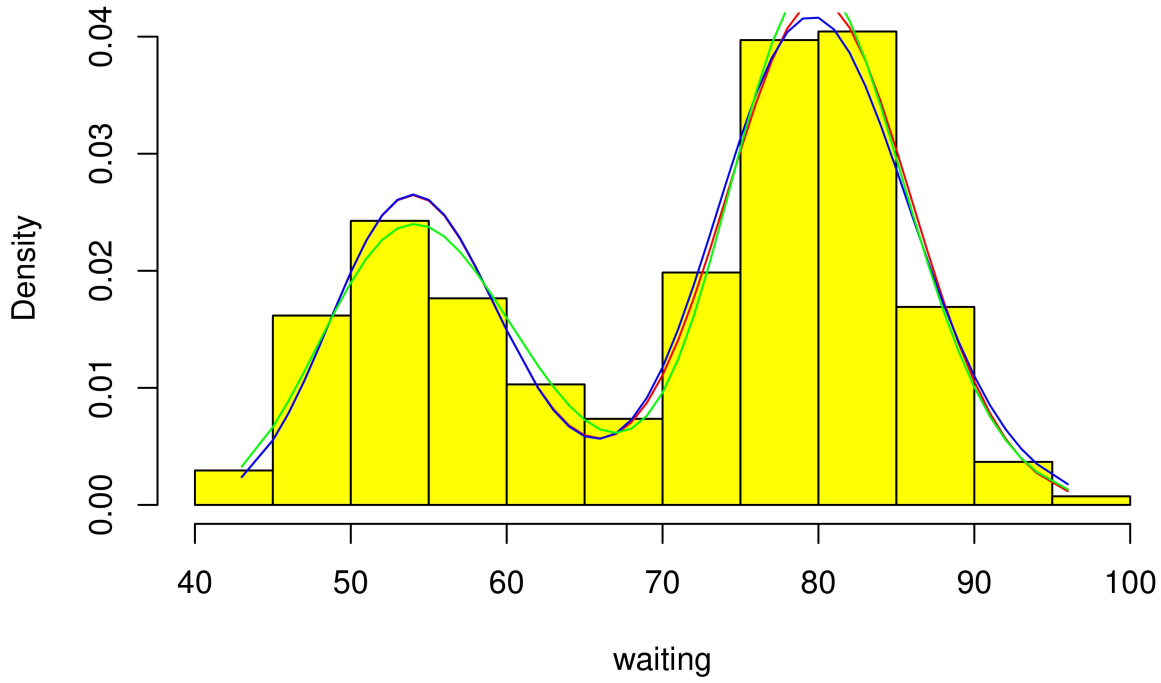
```r
# model 3
p <- length(x[x<65])/length(x)
m_1 <- mean(x[x<65])
v_1 <- var(x[x<65])
sigma2_1 <- log((v_1/m_1^2) + 1)
mu_1 <- log(m_1) - sigma2_1/2
m_2 <- mean(x[x>=65])
v_2 <- var(x[x>=65])
sigma2_2 <- log((v_2/m_2^2) + 1)
mu_2 <- log(m_2) - sigma2_2/2
theta_inital <- c(p, mu_1, sqrt(sigma2_1), mu_2, sqrt(sigma2_2))
neg_log_likelihood <- function(theta, data) {
  n <- length(data)

  p <- theta[1]
  mu_1 <- theta[2]
  sigma_1 <- theta[3]
  mu_2 <- theta[4]
  sigma_2 <- theta[5]

  l <- 0
  for (i in 1:n) {
    l = l + log(p*dlnorm(data[i], meanlog = mu_1, sdlog = sigma_1) + (1-p)*dlnorm(data[i], meanlog = mu
  }

  return(-l)
}
fit = optim(theta_inital,
            neg_log_likelihood,
            data = x,
            control = list(maxit = 1500),
            lower = c(0, -Inf, 0, -Inf, 0),
            upper = c(1, Inf, Inf, Inf, Inf),
            method="L-BFGS-B")
theta_3 <- fit$par
theta_3
```

```
## [1] 0.37613816 4.00383608 0.11485512 4.38430182 0.06973823
```

```r
p <- theta_3[1]
mu_1 <- theta_3[2]
sigma_1 <- theta_3[3]
mu_2 <- theta_3[4]
```

```r
sigma_2 <- theta_3[5]
model_3 <- p*dlnorm(x, meanlog = mu_1, sdlog = sigma_1) + (1-p)*dlnorm(x, meanlog = mu_2, sdlog = sigma
aic_3 <- 2*5 + 2*neg_log_likelihood(theta_3, x)
# hist(x, xlab = 'waiting', probability = T, col='pink', main='')
# lines(x, model_3)
```

```r
hist(x, xlab = 'waiting', probability = T, col='yellow', main='')
lines(x, model_1, col = "red")
lines(x, model_2, col = "blue")
lines(x, model_3, col = "green")
```




```r
results <- data.frame(
  models = c("Gamma + Normal", "Gamma + Gamma", "Lognormal + Lognormal"),
  AIC = c(aic_1, aic_2, aic_3)
)
results
```

```
##                  models      AIC
## 1        Gamma + Normal 2077.495
## 2         Gamma + Gamma 2078.725
## 3 Lognormal + Lognormal 2075.420
```