

# Bayesian Data Analysis - Changepoint Detection

*Anonymous*

*2019-12-08*

## Contents

<b>1</b>	<b>Loaded packages</b>	<b>1</b>
<b>2</b>	<b>Introduction</b>	<b>2</b>
<b>3</b>	<b>Description of the Problem</b>	<b>2</b>
<b>4</b>	<b>Description of the Data</b>	<b>2</b>
4.1	Coal Mining Dataset . . . . .	2
<b>5</b>	<b>Model Details and Stan Codes</b>	<b>4</b>
5.1	Gamma-Poisson Multiple Changepoint Model . . . . .	4
5.2	Hierarchical Multiple Changepoint Model . . . . .	6
5.3	Bayesian Online Changepoint Detection . . . . .	7
<b>6</b>	<b>Prior Choices and Fitting the Models</b>	<b>10</b>
6.1	Gamma-Poisson Multiple Changepoint Model . . . . .	10
6.2	Hierarchical Multiple Changepoint Model . . . . .	10
6.3	Bayesian Online Changepoint Detection . . . . .	11
<b>7</b>	<b>Histograms and Divergences of Parameters</b>	<b>11</b>
7.1	Gamma-Poisson Multiple Changepoint Model . . . . .	11
7.2	Hierarchical Multiple Changepoint Model . . . . .	17
7.3	Bayesian Online Changepoint Detection . . . . .	23
<b>8</b>	<b>Bayesian Online Changepoint Detection (Syntetic Data)</b>	<b>30</b>
<b>9</b>	<b>Posterior Prediction</b>	<b>31</b>
<b>10</b>	<b>Conclusion</b>	<b>31</b>

## 1 Loaded packages

```
library(rstan)
library(BaM)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

## 2 Introduction

In this project we have implemented three different changepoint detection algorithms. While two of them are detecting two changepoints in a given data, the third one is working online and can detect as many as changepoints which are possible.

In this report, we are comparing these models and sharing statistical workflow of these algorithms.

## 3 Description of the Problem

Given a sequential data we might interest in the changes in the data, especially these data change in some way, such as a machinery system could goes to failure or economic trends could be changed. If we assume that data is a result from such a generative process, then we can find the changes in this generative process.

Changepoint detection models aim to detect the changes in signals or time series data. The models of changepoint detection may detect only one changepoint, or multiple changepoints. In our project we implemented three different models.

- Gamma-Poisson Multiple Changepoint Detection for two changepoints
- Hierarchical Gamma-Poisson Multiple Changepoint Detection for two changepoints

The third model we implemented is an online changepoint model.

- Bayesian Online Changepoint Detection<sup>[\*]</sup>

The details of the models are given in the following sections.

[\*] Adams, Ryan Prescott, and David JC MacKay. “Bayesian online changepoint detection.” arXiv preprint arXiv:0710.3742 (2007).

## 4 Description of the Data

### 4.1 Coal Mining Dataset

We are using Coal Mining Disasters dataset from BaM library (BaM: Functions and Datasets for Books by Jeff Gill). The data is a vector of British Coal Mining Disasters in length 111.

The data is the number of deaths along a time period at the coal mine disasters.

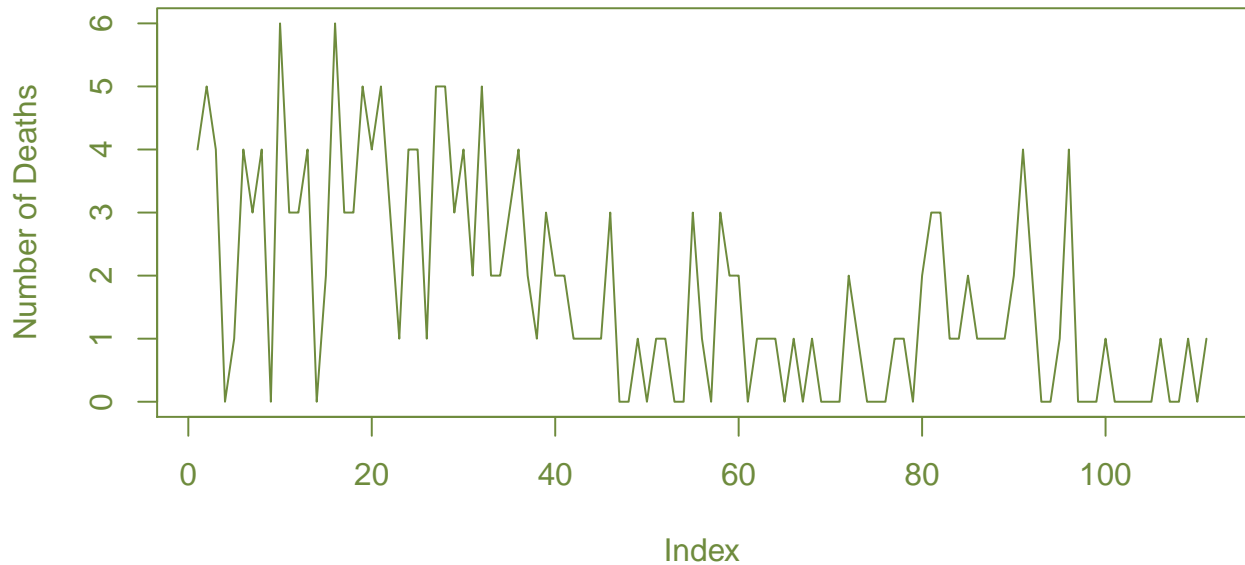
Source: Lynn, R. and Vanhanen, T. (2001). National IQ and Economic Development. Mankind Quarterly LXI, 415-437.

```
data <- list(T = length(coal.mining.disasters),
            D = coal.mining.disasters,
            H = 0.01,
            mean_D = mean(coal.mining.disasters))
```

```

color = 'darkolivegreen4'
par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
plot(coal.mining.disasters, type="l", ylab = 'Number of Deaths',)

```



## Synthetic Data

Additionally, in order to show the performance of the Bayesian Online Changepoint Model (BOCM) on multiple changepoints, we are using this synthetic data:

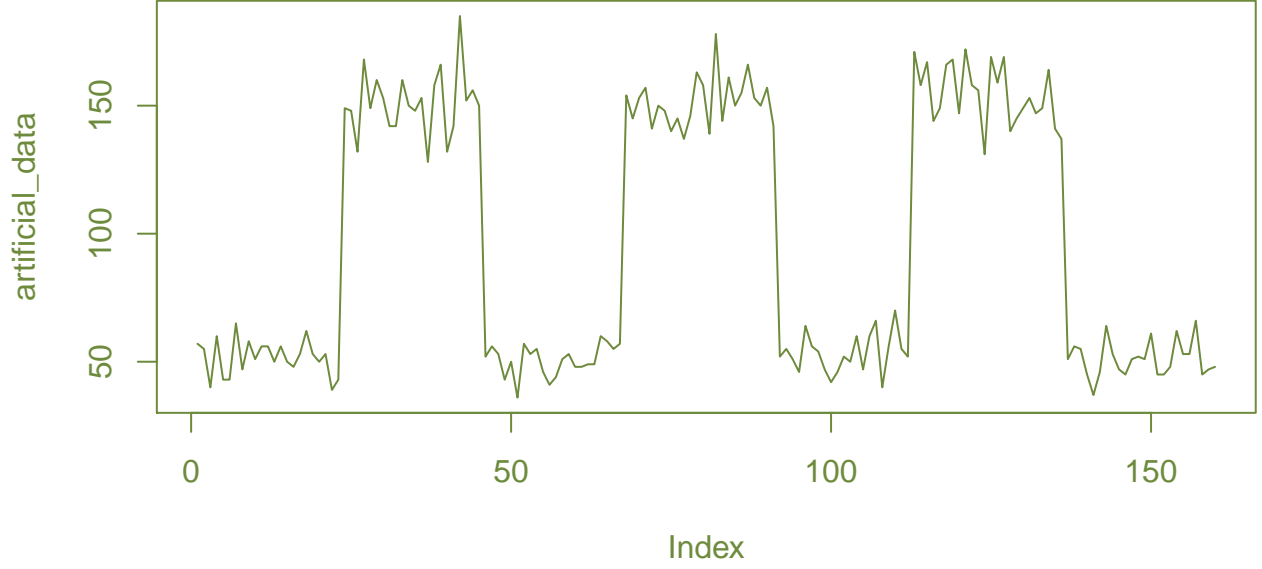
```

artificial_data <- c(read.csv("artificial_data.csv", header = FALSE))$V1

data_synthetic <- list(T = length(artificial_data),
  D = artificial_data,
  H = 0.01)

color = 'darkolivegreen4'
par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
plot(artificial_data, type="l")

```



## 5 Model Details and Stan Codes

### 5.1 Gamma-Poisson Multiple Changepoint Model

The assumptions of this model are: - There are two changepoints on the time series data, that is there are three intervals. - The data comes from a Poisson distribution in each interval. - The means of the Poisson distribution generated by underlying Gamma distributions.

The generative process of the model is:

$$\begin{aligned}
 e &\sim \text{Gamma}(r_e, 1) \\
 l &\sim \text{Gamma}(r_l, 1) \\
 m &\sim \text{Gamma}(r_m, 1) \\
 s_1 &\sim \text{Uniform}(1, T) \\
 s_2 &\sim \text{Uniform}(1, T)
 \end{aligned}$$

$$D_t \sim \text{Poisson}(\lambda) \quad \lambda = \begin{cases} e & \text{if } t < s_1 \\ l & \text{if } s_1 \leq t < s_2 \\ m & \text{else} \end{cases}$$

The likelihood of the model is:

$$\begin{aligned}
 p(D|e, l, m) &= \sum_{s_1=1}^T \sum_{s_2=s_1}^T p(s_1, s_2, D|e, l, m) \\
 &= \sum_{s_1=1}^T \sum_{s_2=s_1}^T p(s_1)p(s_2)p(D|s_1, s_2, e, l, m) \\
 &= \sum_{s_1=1}^T \sum_{s_2=s_1}^T \text{Uniform}(s_1|1, T) \text{Uniform}(s_2|1, T) \prod_{t=1}^T \text{Poisson}(D_t | t < s_1 ? e : (t < s_2) ? l : m)
 \end{aligned}$$

We modified the single Exponential-Poisson changepoint[\*,\*\*] model to handle two changepoints. And modified the linear single changepoint algorithm to handle two changepoints. The computation of our code is quadratic,  $O((n^2+3n)/2)$ , instead of  $O(n^3)$ .

[\*] Fonnesbeck, Chris, Anand Patil, David Huard, and John Salvatier. 2013. PyMC User's Guide. [\*\*] Stan's User Guide, Change Point Models; [https://mc-stan.org/docs/2\\_21/stan-users-guide/change-point-section.html](https://mc-stan.org/docs/2_21/stan-users-guide/change-point-section.html)

The Stan code of Gamma-Poisson Multiple Changepoint Model for two changepoints is as follows:

```
writeLines(readLines("./StanFiles/multiple_changepoint_gamma.stan"))

## data {
##   int<lower=1> T;
##   int<lower=0> D[T];
##   real<lower=0> mean_D;
## }
## transformed data {
##   real log_unif;
##   log_unif = -log(T);
## }
## parameters {
##   real<lower=0> e;
##   real<lower=0> l;
##   real<lower=0> m;
## }
## transformed parameters {
##   matrix[T, T] lp;
##
##   {
##     row_vector[T + 1] lp_e;
##     row_vector[T + 1] lp_l;
##     row_vector[T + 1] lp_m;
##     lp = rep_matrix(2*log_unif, T, T);
##     lp_e[1] = 0;
##     lp_l[1] = 0;
##     lp_m[1] = 0;
##
##     for (t in 1:T) {
##       lp_e[t + 1] = lp_e[t] + poisson_lpmf(D[t] | e);
##       lp_l[t + 1] = lp_l[t] + poisson_lpmf(D[t] | l);
##       lp_m[t + 1] = lp_m[t] + poisson_lpmf(D[t] | m);
##     }
##     for (s1 in 1:T){
##       for (s2 in s1:T){
##         lp[s1,s2] = lp_e[s1+1] - lp_l[s1+1] + lp_l[s2+1] - lp_m[s2+1] + lp_m[T+1];
##         lp[s2,s1] = -1e6;
##       }
##     }
##   }
## }
## model {
##   e ~ gamma(mean_D,1);
##   l ~ gamma(mean_D,1);
##   m ~ gamma(mean_D,1);
```

```

##   target += log_sum_exp(lp);
## }
## generated quantities {
##   int<lower=1,upper=T> s1s;
##   int<lower=1,upper=T> s2s;
##   int tmp;
##   tmp = categorical_logit_rng(to_vector(lp));
##   s1s = (tmp % T) == 0 ? T : (tmp % T);
##   s2s = (tmp / T) == T ? T : (tmp / T + 1);
##
##
## }

```

## 5.2 Hierarchical Multiple Changepoint Model

In the hierarchical multiple changepoint model, we assume that the shape parameters of the priors for the interval means have a common prior.

$$\begin{aligned}
r_{\{e,l,m\}} &\sim \text{Gamma}(\alpha, 1) \\
e &\sim \text{Gamma}(r_e, 1) \\
l &\sim \text{Gamma}(r_l, 1) \\
m &\sim \text{Gamma}(r_m, 1) \\
s_1 &\sim \text{Uniform}(1, T) \\
s_2 &\sim \text{Uniform}(1, T)
\end{aligned}$$

$$D_t \sim \text{Poisson}(\lambda) \quad \lambda = \begin{cases} e & \text{if } t < s_1 \\ l & \text{if } s_1 \leq t < s_2 \\ m & \text{else} \end{cases}$$

The Stan code of Hierarchical Gamma-Poisson Multiple Changepoint Model for two changepoints is as follows:

```

writeLines(readLines("./StanFiles/hierar_multiple_changepoint_gamma.stan"))

```

```

## data {
##   int<lower=1> T;
##   int<lower=0> D[T];
##   real<lower=0> mean_D;
## }
## transformed data {
##   real log_unif;
##   log_unif = -log(T);
## }
## parameters {
##   real<lower=0> e;
##   real<lower=0> l;
##   real<lower=0> m;
##   real<lower=0> r[3];
## }
## transformed parameters {
##   matrix[T, T] lp;

```

```

## {
##   row_vector[T + 1] lp_e;
##   row_vector[T + 1] lp_l;
##   row_vector[T + 1] lp_m;
##   lp = rep_matrix(2*log_unif, T, T);
##   lp_e[1] = 0;
##   lp_l[1] = 0;
##   lp_m[1] = 0;
##
##   for (t in 1:T) {
##     lp_e[t + 1] = lp_e[t] + poisson_lpmf(D[t] | e);
##     lp_l[t + 1] = lp_l[t] + poisson_lpmf(D[t] | l);
##     lp_m[t + 1] = lp_m[t] + poisson_lpmf(D[t] | m);
##   }
##   for (s1 in 1:T){
##     for (s2 in s1:T){
##       lp[s1,s2] = lp_e[s1+1] - lp_l[s1+1] + lp_l[s2+1] - lp_m[s2+1] + lp_m[T+1];
##       lp[s2,s1] = -1e6;
##     }
##   }
## }
## }
## model {
##   r ~ gamma(mean_D,1);
##   e ~ gamma(r[1],1);
##   l ~ gamma(r[2],1);
##   m ~ gamma(r[3],1);
##   target += log_sum_exp(lp);
## }
## generated quantities {
##   int<lower=1,upper=T> s1s;
##   int<lower=1,upper=T> s2s;
##   int tmp;
##   tmp = categorical_logit_rng(to_vector(lp));
##   s1s = (tmp % T) == 0 ? T : (tmp % T);
##   s2s = (tmp / T) == T ? T : (tmp / T + 1) ;
## }

```

### 5.3 Bayesian Online Changepoint Detection

In this section we will discuss about online change point algorithm and its Stan application with explaining Ryan P. Adams and David J.C. MacKay's technical report on Bayesian online changepoint detection.

In contrast to previous model we won't assume any change point in this model. Instead, the sequence of observation can be divided into non-overlapping subsequences. We will be interested in the transition of these non-overlapping subsequences.

Bayesian online changepoint detection focuses on run-length of the sub-sequences from last changepoint and run length is denoted as  $r_t$  and it is updated as follows:

$$P(r_t | r_{t-1}) = \begin{cases} H(r_{t-1} + 1) & \text{if } r_t = 0 \\ 1 - H(r_{t-1} + 1) & \text{if } r_t = r_{t-1} + 1 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In which  $H(\tau)$  is hazard function which is dependend on current run length. For the simplicity it could be treated as a constant also and we will show it as  $H(\tau) = 1/\lambda$ .

### 5.3.1 Recursive Run Length Estimation

If we assume that we can calculate the predictive distribution of most recent observation given the current run length then we can calculate the marginal predictive distribution by integrating over the posterior distribution.

$$P(x_{t+1} | \mathbf{x}_{1:t}) = \sum_{r_t} P(x_{t+1} | r_t, \mathbf{x}_t^{(r)}) P(r_t | \mathbf{x}_{1:t}) \quad (2)$$

where posterior distribution of run length at time is;

$$P(r_t | \mathbf{x}_{1:t}) = \frac{P(r_t, \mathbf{x}_{1:t})}{P(\mathbf{x}_{1:t})} \quad (3)$$

The joint distribution can be calculated by recursive algorithm which is similar to forward algorithm of the HMMs;

$$P(r_t, \mathbf{x}_{1:t}) = \sum_{r_{t-1}} P(r_t, r_{t-1}, \mathbf{x}_{1:t}) \quad (4)$$

$$= \sum_{r_{t-1}} P(r_t, x_t | r_{t-1}, \mathbf{x}_{1:t-1}) P(r_{t-1}, \mathbf{x}_{1:t-1}) \quad (5)$$

$$= \sum_{r_{t-1}} P(r_t | r_{t-1}) P(x_t | r_{t-1}, \mathbf{x}_t^{(r)}) P(r_{t-1}, \mathbf{x}_{1:t-1}) \quad (6)$$

Ultimately, we want to infer both the run-length posterior distribution  $p(r_t | \mathbf{x}_{1:t})$  and the posterior predictive distribution  $p(x_{t+1} | \mathbf{x}_{1:t})$  so that we can predict the next data point given all the data we have seen so far.

We implemented two versions of this model In the first one, we have implemeted the analytical solution of the BOCM. In the second one, we have implemented the Monte Carlo solution of this model. In both of them observations are coming from again Poisson distribution, while the mean of the Poisson distribution comes from the underlying Gamma distribution.

The Stan code of the analytical solution of BOCM is as follows:

```
writeLines(readLines("./StanFiles/online_analytic.stan"))
```

```
## data {
##   int<lower=1> T;
##   int<lower=0> D[T];
##   real<lower=0, upper=1> H;
## }
## transformed data {
##   real log_increase;
##   real log_decrease;
##   matrix[T+1, T+1] R;
##   matrix[T+1, T+1] alpha;
```



```

## matrix[T+1, T+1] beta;
##
## log_increase = log(1 - H);
## log_decrease = log(H);
## R = rep_matrix(-1e20, T+1, T+1);
## R[1,1] = 0;
## alpha = rep_matrix(1, T+1, T+1);
## beta = rep_matrix(1, T+1, T+1);
## for (t in 1:T){
##   for (j in 1:t){
##     R[j+1, t+1] = R[j, t] + neg_binomial_lpmf(D[t] | alpha[j,t], beta[j,t]) + log_increase;
##     R[1, t+1] = log_sum_exp(R[1, t+1], R[j, t] + neg_binomial_lpmf(D[t] | alpha[1,t], beta[1,t]) +
##     alpha[j+1, t+1] = alpha[j, t] + D[t];
##     beta[j+1, t+1] = beta[j, t] + 1;
##   }
## }
## }
## model {
##
## }
## generated quantities {
##   vector[T] meas;
##   for (t in 1:T)
##     meas[t] = categorical_logit_rng(col(R, t+1)[1:t+1]);
## }

```

The Stan code of the Monte Carlo solution of BOCM is as follows:

```

writeLines(readLines("./StanFiles/online_mcmc.stan"))

```

```

## data {
##   int<lower=1> T;
##   int<lower=0> D[T];
##   real<lower=0> mean_D;
##   real<lower=0, upper=1> H;
## }
## transformed data {
##   real log_increase;
##   real log_decrease;
##
##   log_increase = log(1 - H);
##   log_decrease = log(H);
## }
## parameters {
##   vector<lower=0>[T] alpha;
## }
## transformed parameters {
##   matrix[T+1, T+1] R;
##   R = rep_matrix(-1e20, T+1, T+1);
##   R[1,1] = 0;
##   for (t in 1:T){
##     for (j in 1:t){
##       R[j+1, t+1] = R[j, t] + poisson_lpmf(D[t] | alpha[t-j+1]) + log_increase;

```

```

##      R[1, t+1] = log_sum_exp(R[1, t+1], R[j, t] + poisson_lpmf(D[t] | alpha[t]) + log_decrease);
##    }
##  }
## }
## model {
##   alpha ~ gamma(mean_D,1);
##   target += log_sum_exp(R);
## }
## generated quantities {
##   vector[T] meas;
##   for (t in 1:T)
##     meas[t] = categorical_logit_rng(col(R, t+1)[1:t+1]);
## }

```

## 6 Prior Choices and Fitting the Models

### 6.1 Gamma-Poisson Multiple Changepoint Model

#### 6.1.1 Prior Choices

We have Gamma priors for interval means and Uniform prior on changepoints. We are using the mean of the data as the Gamma shape for the interval means priors.

$$\{r_e, r_l, r_m\} = \text{Gamma}(\text{mean}(D), 1)$$

Since we do not want to put any constraints on the changepoints, we are using uniform distribution over the data length.

#### 6.1.2 Fitting the Model

We fit the model for 5 chains and 5000 iterations with 2000 warm-up phase.

```

num_chains = 5
fit_mgamma <- stan('./StanFiles/multiple_changepoint_gamma.stan',
                  data=data,
                  iter=5000, warmup=2000, chains=num_chains, control = list(adapt_delta=0.9)
)
draws_separate_mgamma <- as.data.frame(fit_mgamma)

```

### 6.2 Hierarchical Multiple Changepoint Model

#### 6.2.1 Prior Choices

We are using the mean of the data as the shape of the Gamma distribution on interval mean prior Gamma distribution's shape. That is we set  $\alpha = \text{mean}(D)$  We are using 1 as scale on all Gamma distribution. So, using the mean of the data as Gamma shape of the priors seems plausible for both models.

Since we do not want to put any constraints on the changepoints, we are using uniform distribution over the data length.

### 6.2.2 Fitting the Model

We fit the model for 5 chains and 6000 iterations with 2000 warm-up phase.

```
num_chains = 5
fit_hmgamma <- stan('./StanFiles/hierar_multiple_changepoint_gamma.stan',
  data=data,
  iter=6000, warmup=2000, chains=num_chains, control = list(adapt_delta=0.99)
)
draws_separate_hmgamma <- as.data.frame(fit_hmgamma)
```

## 6.3 Bayesian Online Changepoint Detection

### 6.3.1 Prior Choices

We are using the mean of the data as the shape of the Gamma distribution on interval mean prior Gamma distribution's shape. Additionally we have a hazard prior, which corresponds to the rate of the changepoint. In our experiments we are using  $H = 0.01$ , which means a point is a changepoint with  $p = H$ .

### 6.3.2 Fitting the Model

In the first part we find the analytical solution and we use fixed parameters.

We fit the model for 5 chain and 2000 iterations with 200 warm-up phase.

```
fit_analytic <- stan("StanFiles/online_analytic.stan",
  data=data, algorithm = "Fixed_param",
  iter=2000, warmup=200, chains=5)
draws_analytic <- as.data.frame(fit_analytic)
t_draws_analytic <- t(draws_analytic)
```

In this part, we use MCMC algorithm.

We fit the model for 5 chain and 2000 iterations with 200 warm-up phase.

```
fit_mcmc <- stan("StanFiles/online_mcmc.stan",
  data=data,
  iter=2000, warmup=200, chains=5)

draws_mcmc <- extract(fit_mcmc)
t_draws_mcmc <- t(draws_mcmc$meas)
```

## 7 Histograms and Divergences of Parameters

### 7.1 Gamma-Poisson Multiple Changepoint Model

#### 7.1.1 Histogram of the Parameters

The histograms of the parameters are as follows:

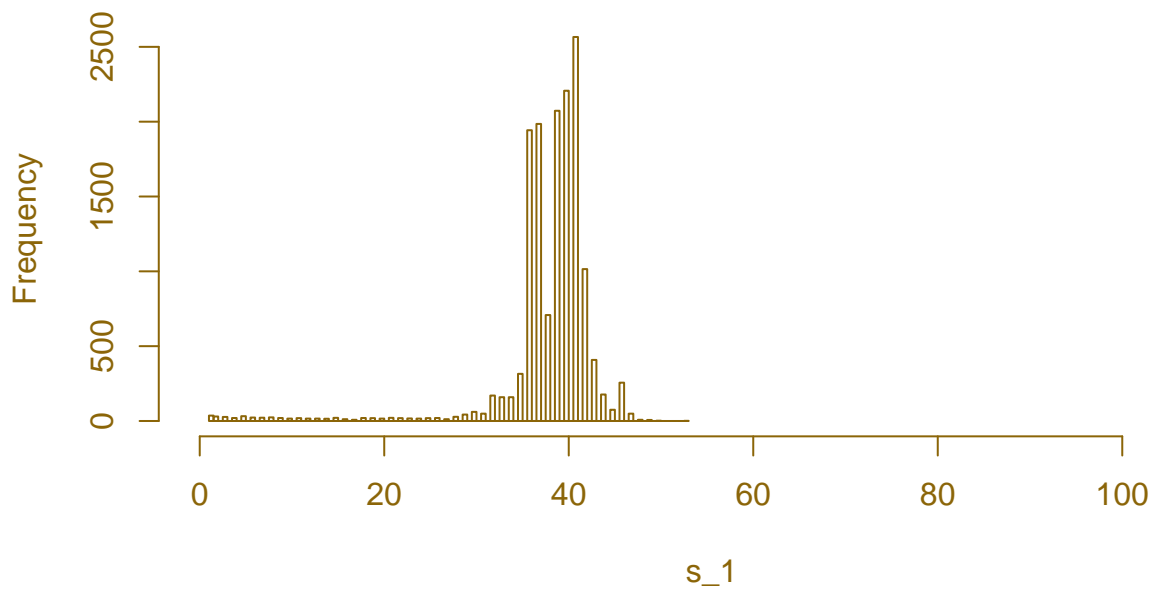
```

color = 'darkgoldenrod4'
samples = draws_separate_mgamma

par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
hist(samples$s1s, breaks = 100, main = 'Histogram of First Changepoint Samples ',
      xlab = 's_1', xlim = c(0,data$T))

```

**Histogram of First Changepoint Samples**

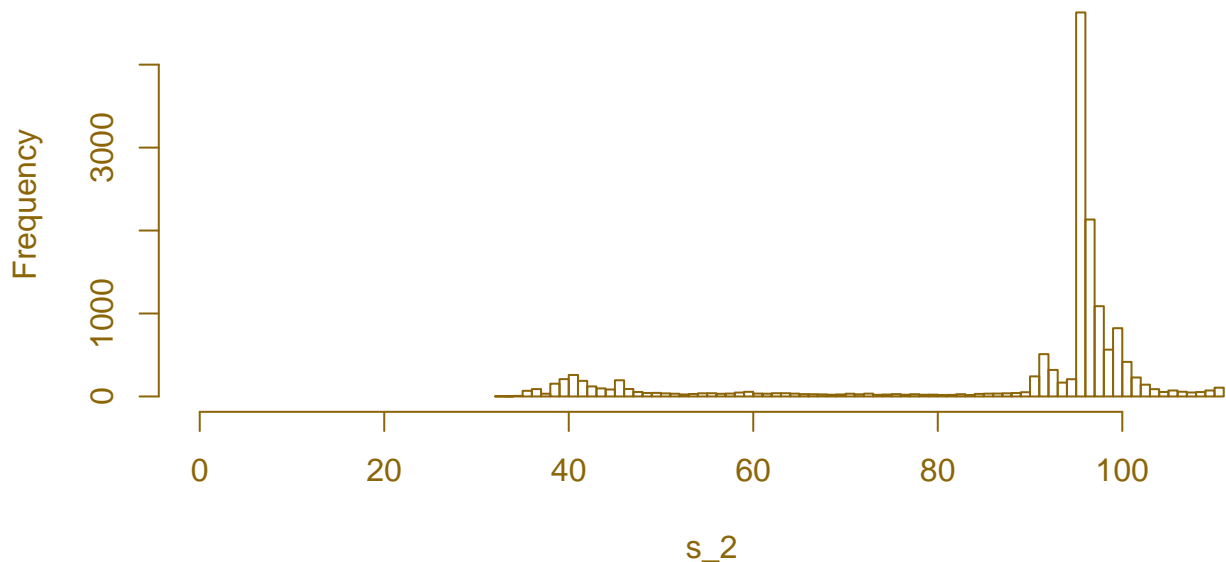


```

par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
hist(samples$s2s, breaks = 100, main = 'Histogram of Second Changepoint Samples ',
      xlab = 's_2', xlim = c(0,data$T))

```

**Histogram of Second Changepoint Samples**

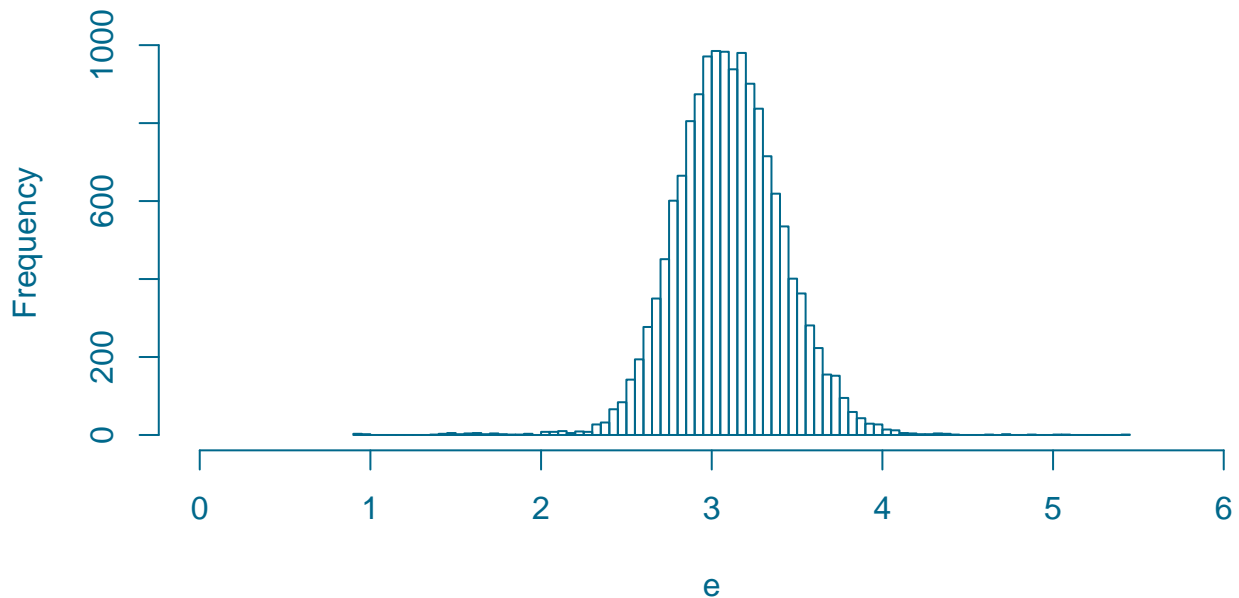


```

color = 'deepskyblue4'
par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
hist(samples$e, breaks = 100, main = 'Histogram of the Mean of the First Interval',
      xlab = 'e', xlim = c(0,6))

```

**Histogram of the Mean of the First Interval**

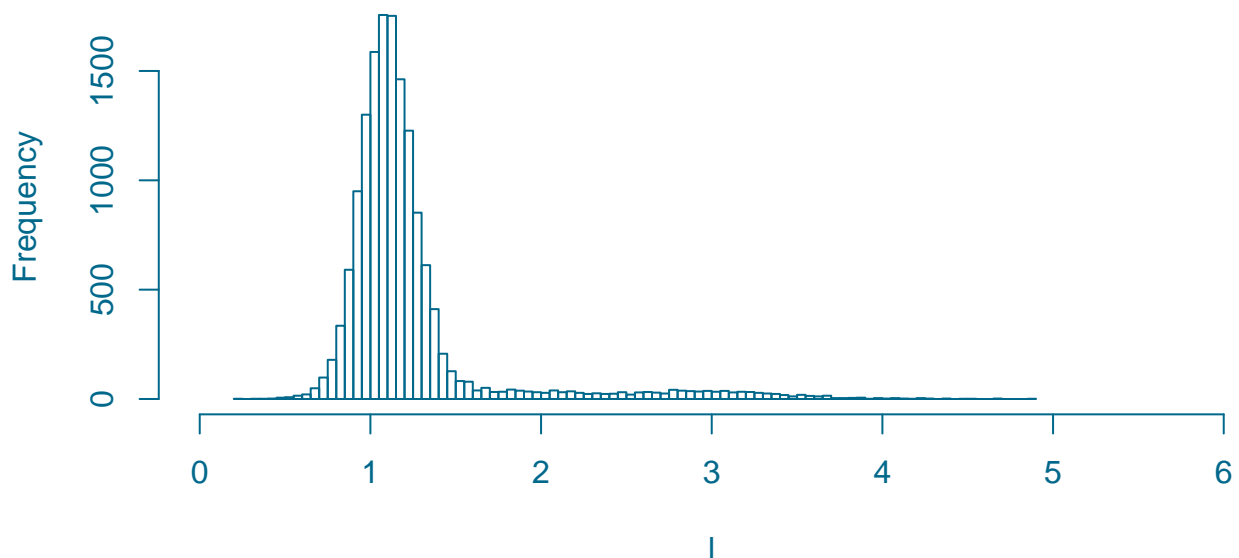


```

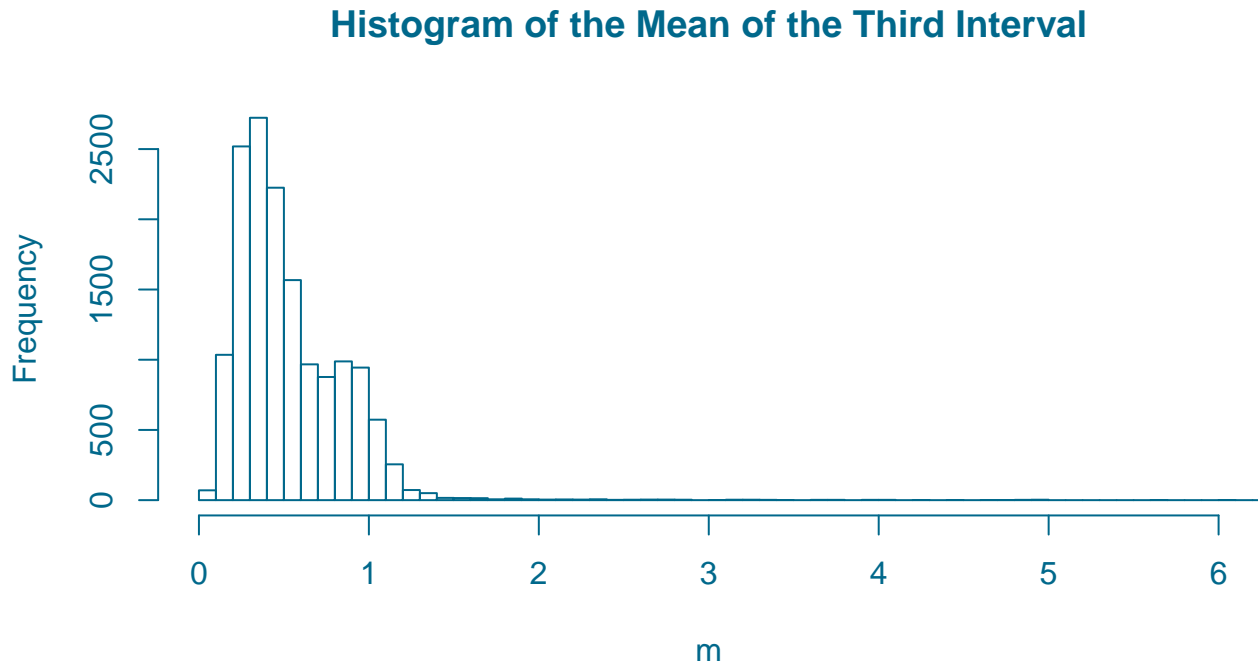
hist(samples$l, breaks = 100, main = 'Histogram of the Mean of the Second Interval',
      xlab = 'l', xlim = c(0,6))

```

**Histogram of the Mean of the Second Interval**



```
hist(samples$m, breaks = 100, main = 'Histogram of the Mean of the Third Interval',
      xlab = 'm', xlim = c(0,6))
```



#### 7.1.2 Convergence Diagnostics

The summary of model fit is as follows:

```
print(fit_mgamma, pars = c("e", "l", "m", "s1s", "s2s"),
      probs = c(0.025, 0.5, 0.975),
      digits_summary = 2, include = TRUE)
```

```
## Inference for Stan model: multiple_changepoint_gamma.
## 5 chains, each with iter=5000; warmup=2000; thin=1;
## post-warmup draws per chain=3000, total post-warmup draws=15000.
##
##      mean se_mean    sd  2.5%   50%  97.5% n_eff Rhat
## e      3.10    0.00  0.31  2.53   3.10   3.72  6014 1.00
## l      1.23    0.01  0.48  0.80   1.12   3.00  1120 1.01
## m      0.53    0.01  0.35  0.15   0.45   1.14  3041 1.00
## s1s 38.01    0.15  5.86 18.00  39.00  45.00  1544 1.00
## s2s 88.64    0.50 18.96 40.00  96.00 106.00  1424 1.00
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 23:36:49 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

As we can see that the Rhat values for all parameters and changepoint estimations are 1, that is our model fitted very well.

The effective sample sizes are looking also good. The ration  $n_{\text{eff}} / n_{\text{transitions}}$  is greater than 0.001 for all parameters.

Here we check the tree depth, E-BFMI and divergences. All are looking good.

```
check_treedepth(fit_mgamma)
```

```
## 0 of 15000 iterations saturated the maximum tree depth of 10.
```

```
check_energy(fit_mgamma)
```

```
## E-BFMI indicated no pathological behavior.
```

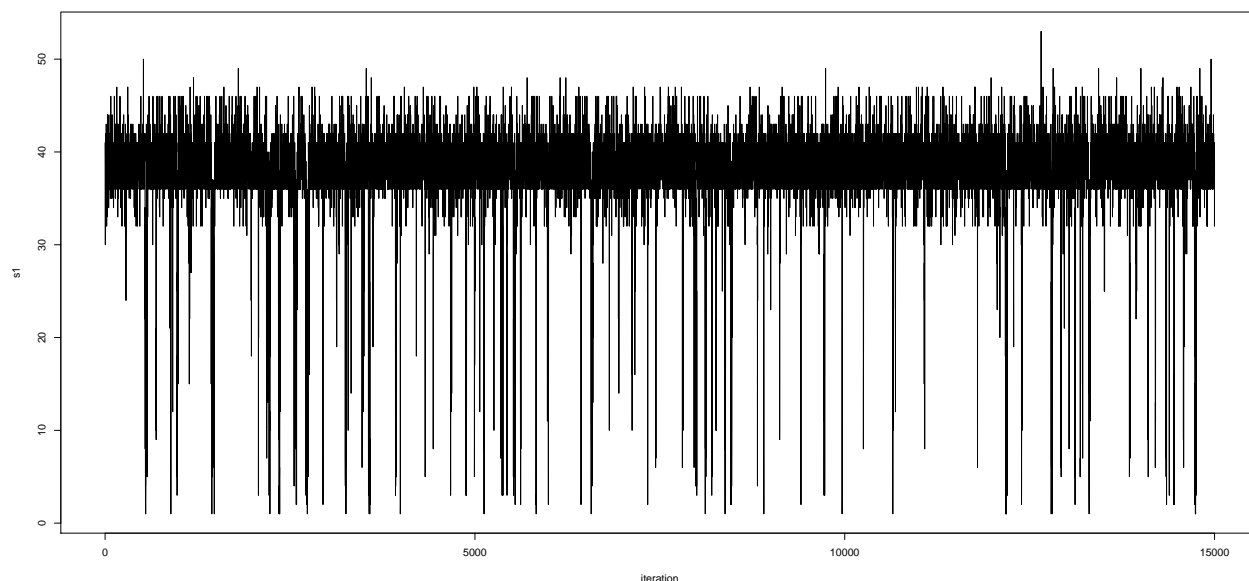
```
check_divergences(fit_mgamma)
```

```
## 0 of 15000 iterations ended with a divergence.
```

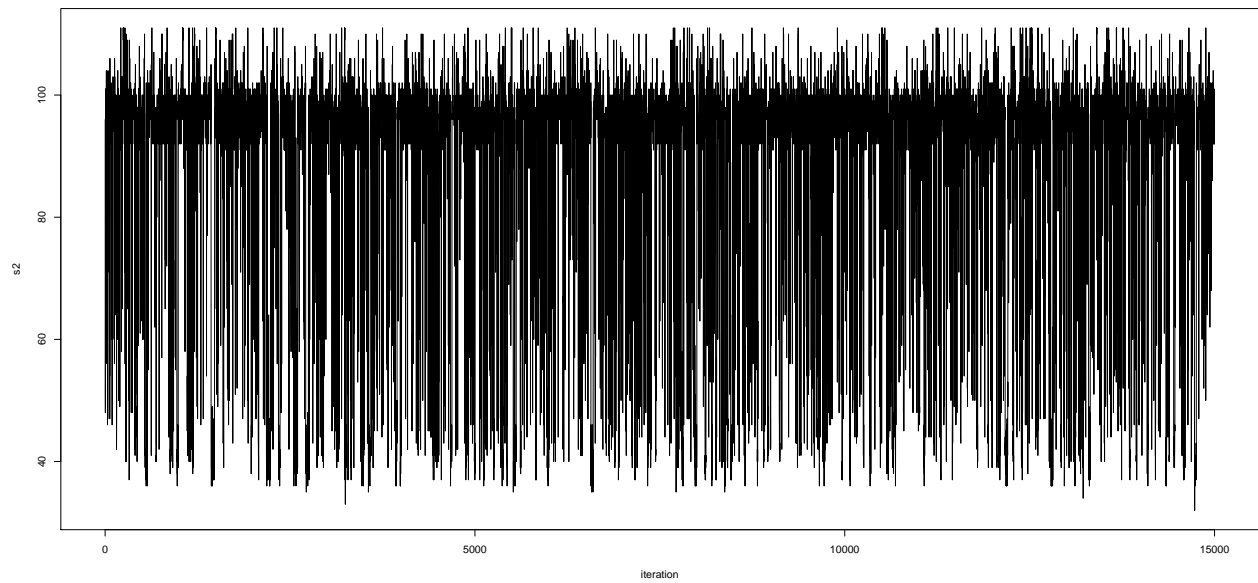
As we can see from the following plots of the samples, the plot of the chains of the mean of first interval indicates that it is converged. The other means of the second and third intervals are also converged but there are some divergences.

When we investigate the plots of the chains for first and second changepoint there are some divergences. However we can also see that the divergence is not random. The divergence on the first changepoint is less than the divergence of the second changepoint. As the histograms of the changepoints also shows that there is a slight probability that the first changepoint is closer to the beginning of the data. And also there is some probability that the second changepoint is around 40. This posterior distribution of the changepoints causes the divergence of the chains.

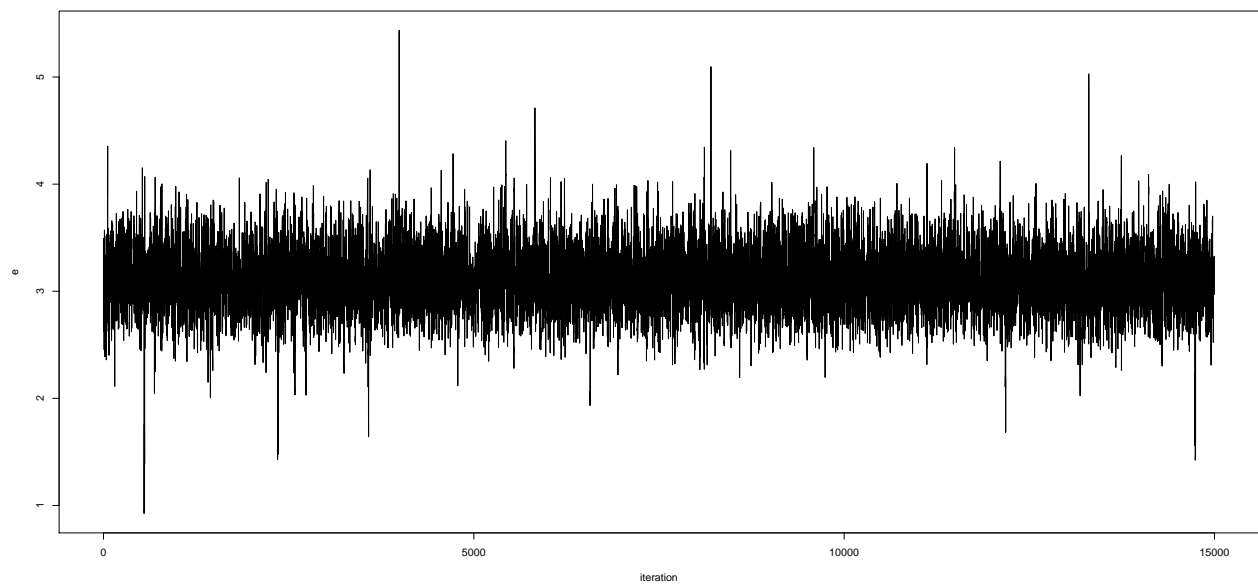
```
matplot(draws_separate_mgamma$s1s, type = c("l"), col = 1:num_chains,  
        xlab = 'iteration', ylab = 's1')
```



```
matplot(draws_separate_mgamma$s2s, type = c("l"), col = 1:num_chains,
        xlab = 'iteration', ylab = 's2')
```

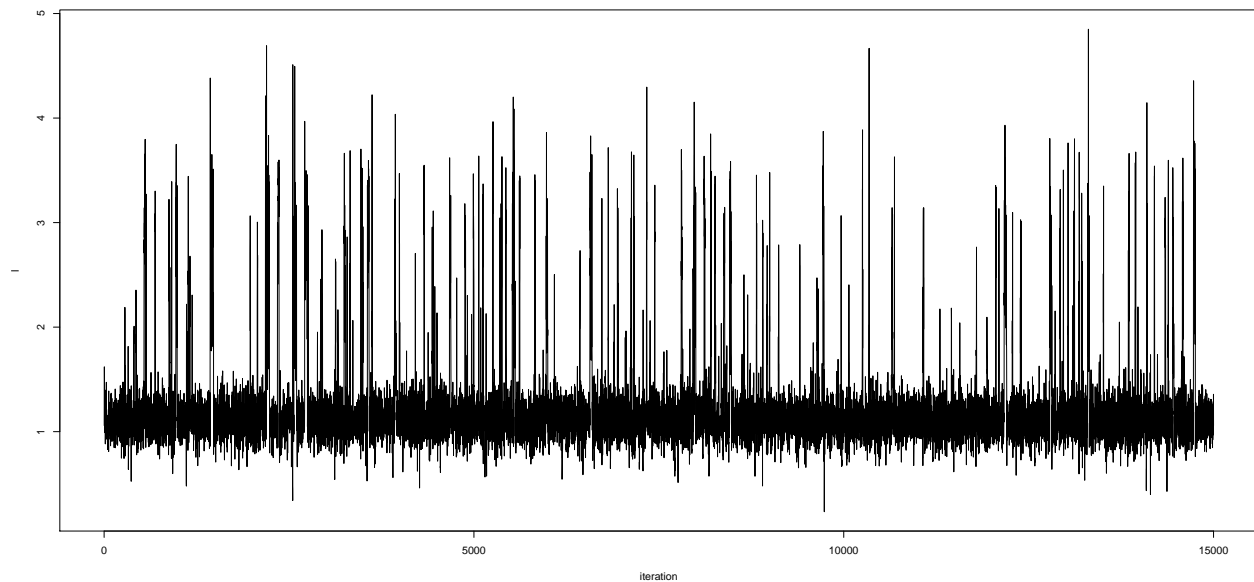


```
matplot(draws_separate_mgamma$e, type = c("l"), col = 1:num_chains,
        xlab = 'iteration', ylab = 'e')
```

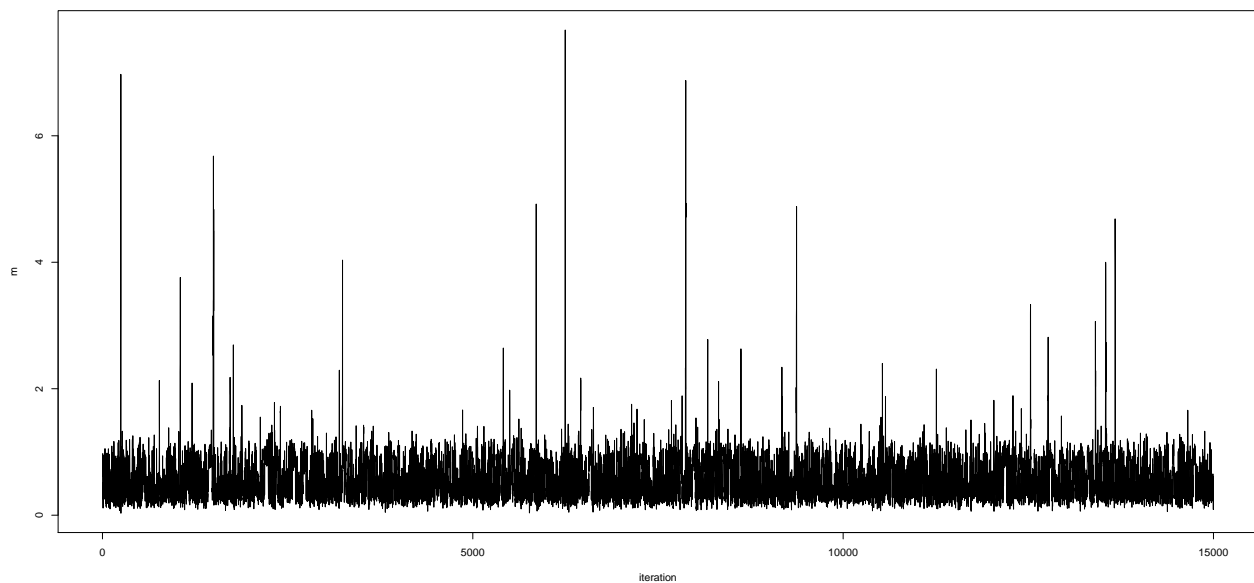


```
matplot(draws_separate_mgamma$l, type = c("l"), col = 1:num_chains,
        xlab = 'iteration', ylab = 'l')
```





```
matplot(draws_separate_mgamma$m, type = c("l"), col = 1:num_chains,
        xlab = 'iteration', ylab = 'm')
```



## 7.2 Hierarchical Multiple Changepoint Model

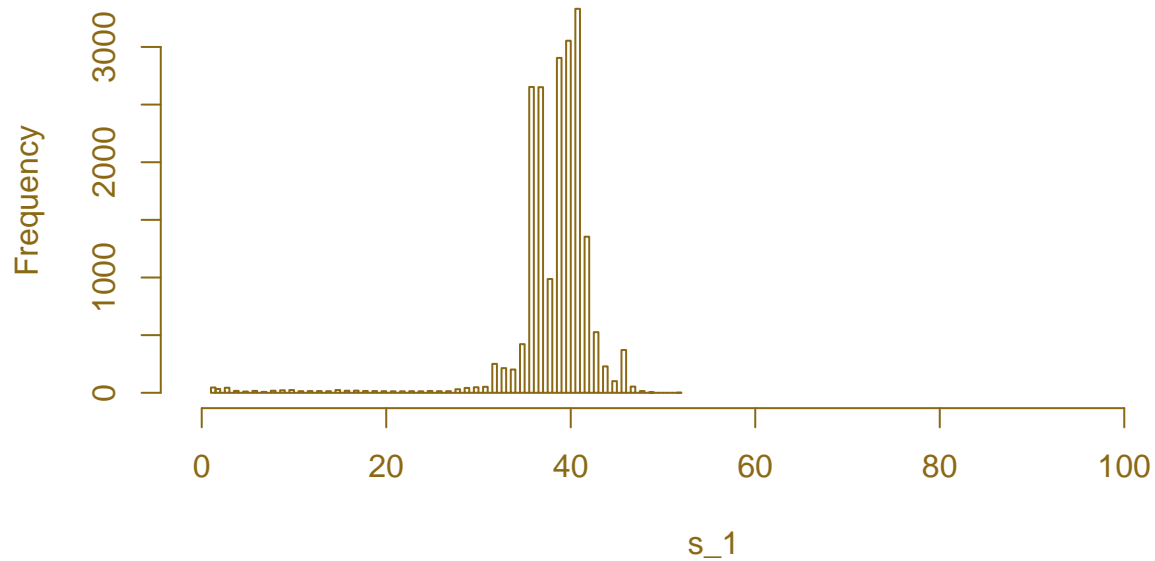
### 7.2.1 Histogram of Parameters

The histograms of the parameters are as follows:

```
color = 'goldenrod4'
samples = draws_separate_hmgamma

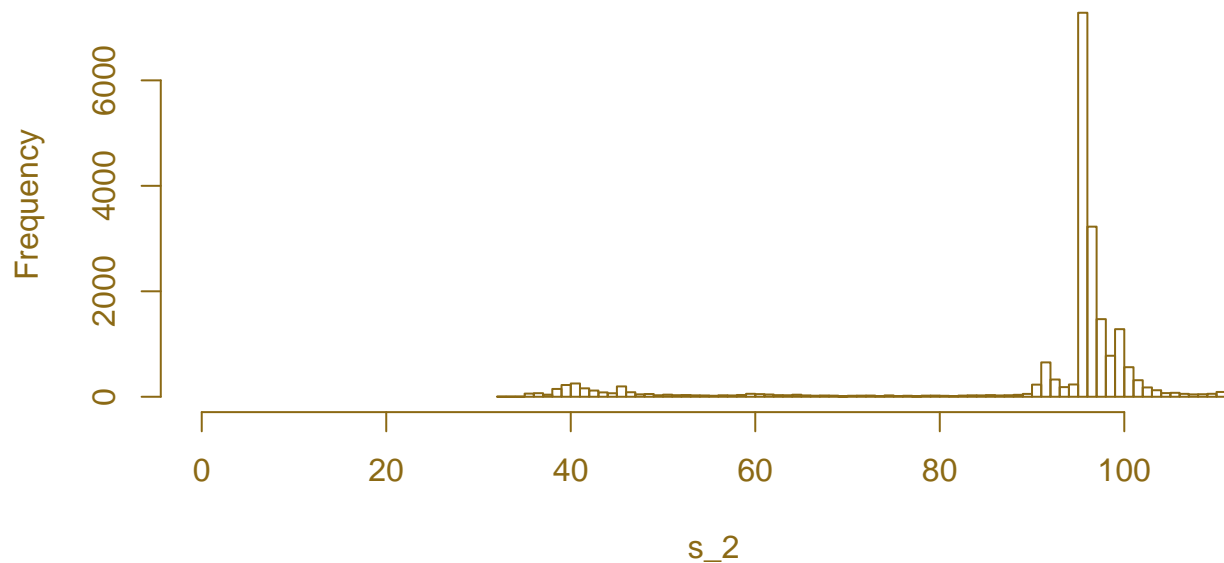
par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
hist(samples$s1s, breaks = 100, main = 'Histogram of First Changepoint Samples ',
     xlab = 's_1', xlim = c(0,data$T))
```

## Histogram of First Changepoint Samples



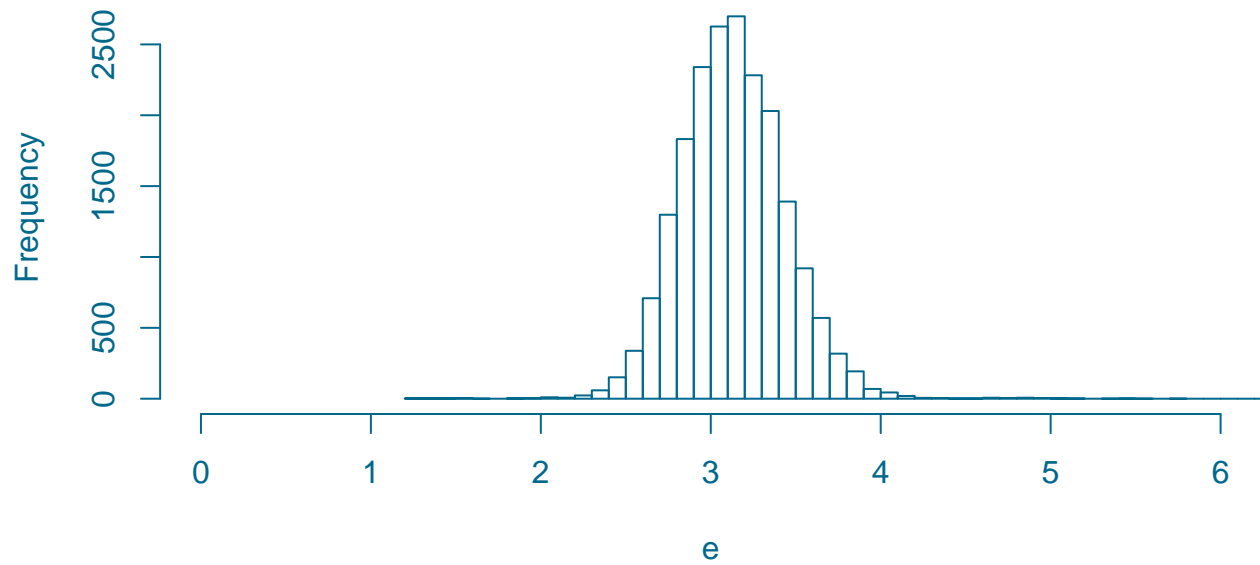
```
par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
hist(samples$s2s, breaks = 100, main = 'Histogram of Second Changepoint Samples ',
      xlab = 's_2', xlim = c(0,data$T))
```

## Histogram of Second Changepoint Samples



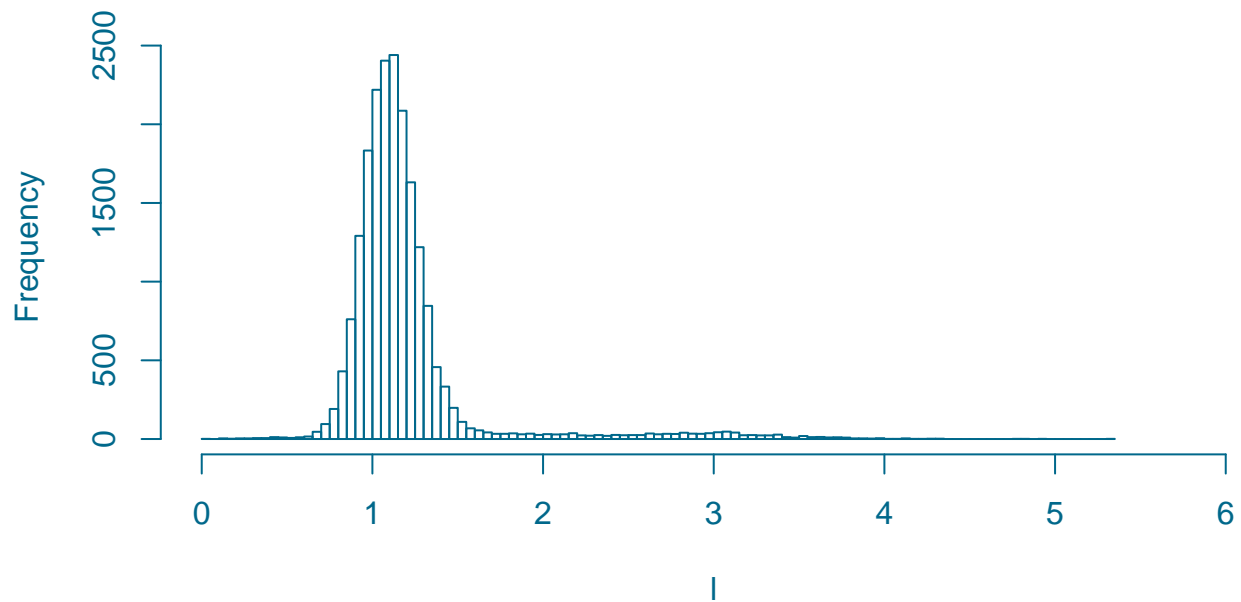
```
color = 'deepskyblue4'
par(bg=NA, fg = color, col.lab=color, col.axis=color, col.main=color)
hist(samples$e, breaks = 100, main = 'Histogram of the Mean of the First Interval',
      xlab = 'e', xlim = c(0,6))
```

### Histogram of the Mean of the First Interval



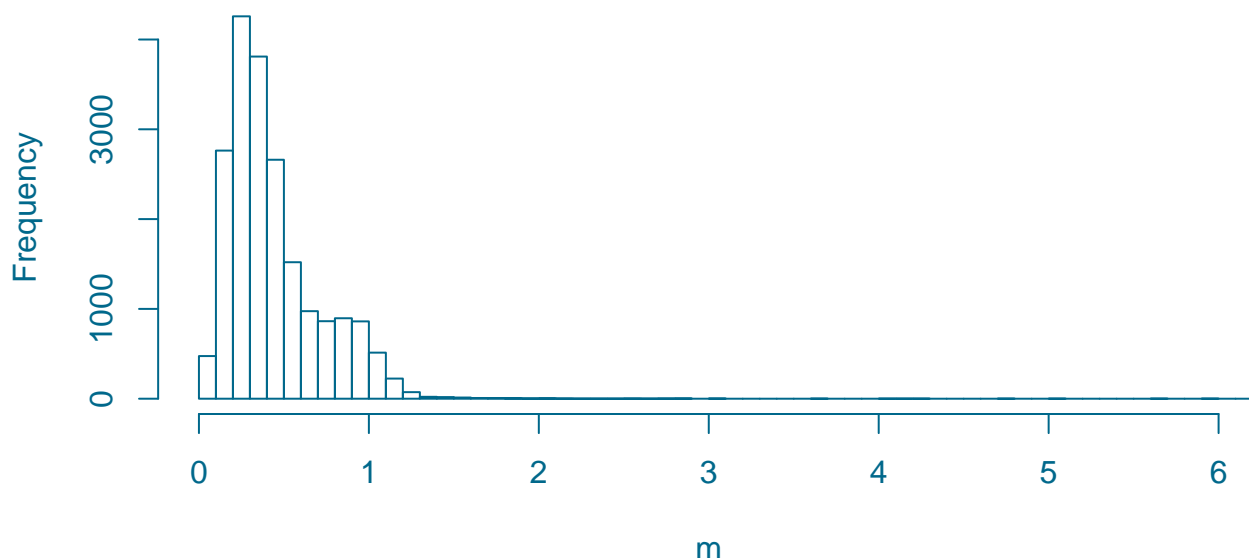
```
hist(samples$l, breaks = 100, main = 'Histogram of the Mean of the Second Interval',  
      xlab = 'l',xlim = c(0,6))
```

### Histogram of the Mean of the Second Interval



```
hist(samples$m, breaks = 100, main = 'Histogram of the Mean of the Third Interval',  
      xlab = 'm',xlim = c(0,6))
```

## Histogram of the Mean of the Third Interval



### 7.2.2 Convergence Diagnostics

The summary of the model fit is as follows:

```
print(fit_hmgamma, pars = c("e", "l", "m", "s1s", "s2s"),
      probs = c(0.025, 0.5, 0.975),
      digits_summary = 2, include = TRUE)
```

```
## Inference for Stan model: hierar_multiple_changepoint_gamma.
## 5 chains, each with iter=6000; warmup=2000; thin=1;
## post-warmup draws per chain=4000, total post-warmup draws=20000.
##
##      mean se_mean    sd  2.5%   50%  97.5% n_eff Rhat
## e      3.13    0.00  0.32   2.58   3.12   3.75 11444    1
## l      1.20    0.01  0.43   0.81   1.11   2.83  1431    1
## m      0.44    0.00  0.34   0.10   0.36   1.08  4529    1
## s1s 38.29    0.12  5.20  27.00  39.00  45.00  1979    1
## s2s 91.20    0.38 16.38  40.00  96.00 104.00  1838    1
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 23:41:21 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

As we can see that the Rhat values for all parameters and changepoint estimations are 1, that is our model fitted very well.

The effective sample sizes are looking also good. The ration  $n_{\text{eff}} / n_{\text{transitions}}$  is greater than 0.001 for all parameters.

Here we check the tree depth, E-BFMI and divergences. All are looking good.

```
check_treepdepth(fit_hmgamma)
```

```
## 0 of 20000 iterations saturated the maximum tree depth of 10.
```

```
check_energy(fit_hmgamma)
```

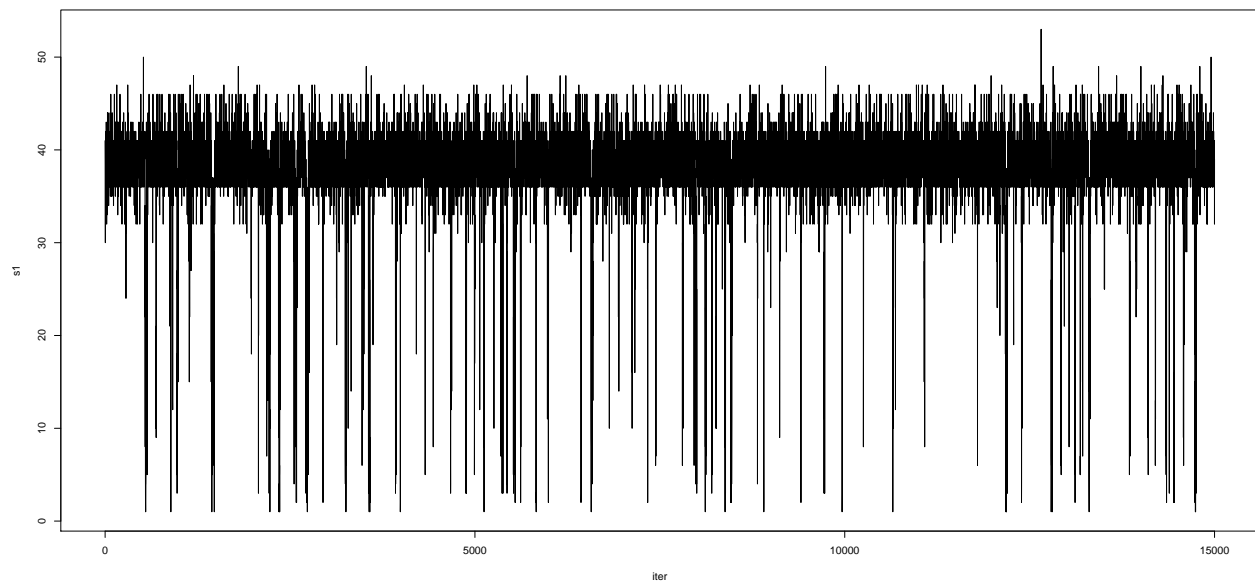
```
## E-BFMI indicated no pathological behavior.
```

```
check_divergences(fit_hmgamma)
```

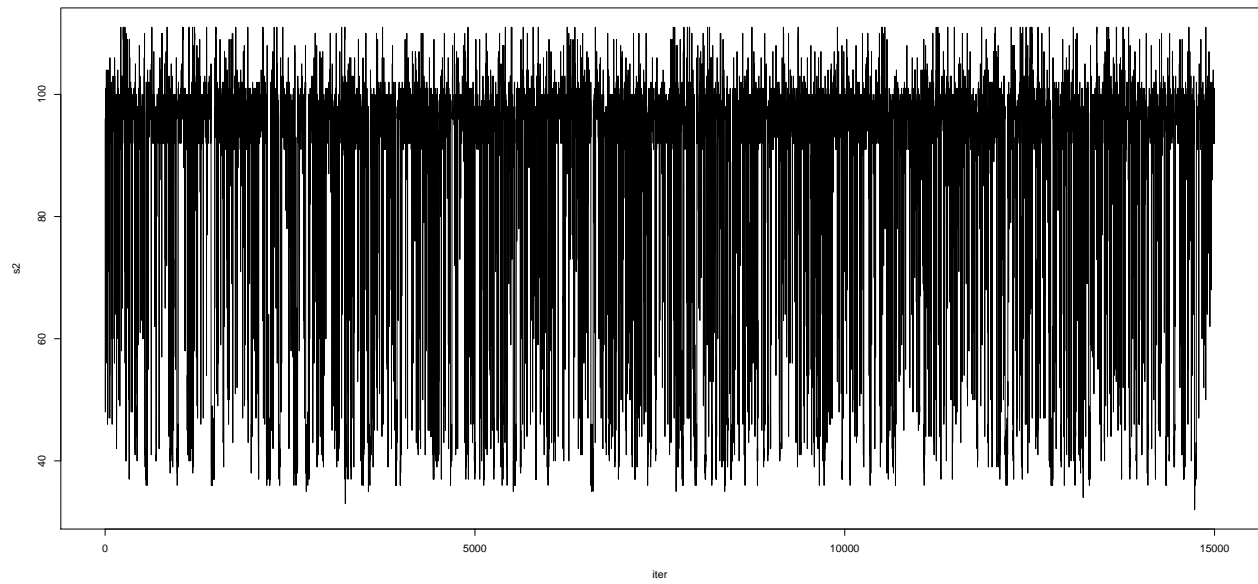
```
## 0 of 20000 iterations ended with a divergence.
```

The same case as in the Non-hierarchical model happens here. The best converged parameter is the mean of the first interval. The divergence on the second changepoint is the highest because of the two possible changepoint locations.

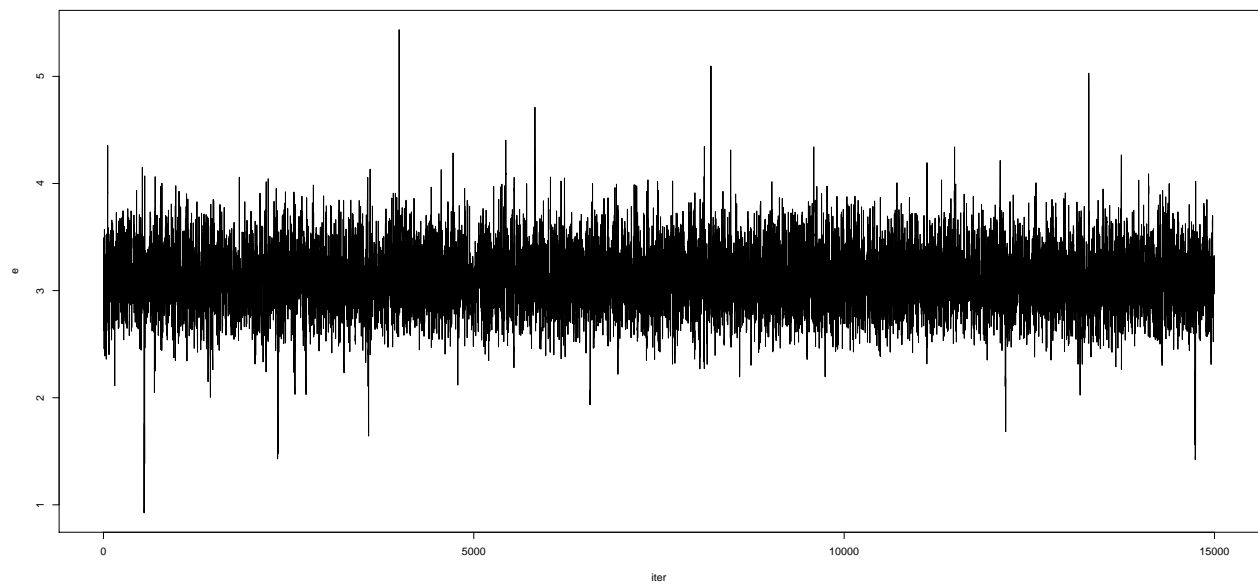
```
matplot(draws_separate_mgamma$s1s, type = c("l"), col = 1:num_chains,  
        xlab = 'iter', ylab = 's1')
```



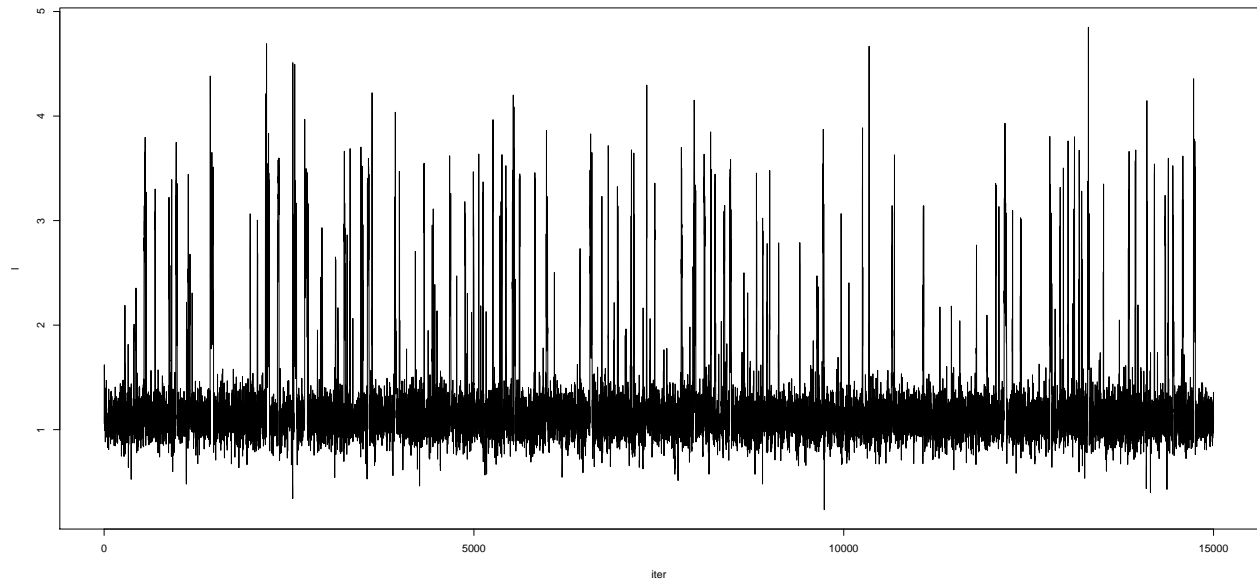
```
matplot(draws_separate_mgamma$s2s, type = c("l"), col = 1:num_chains,  
        xlab = 'iter', ylab = 's2')
```



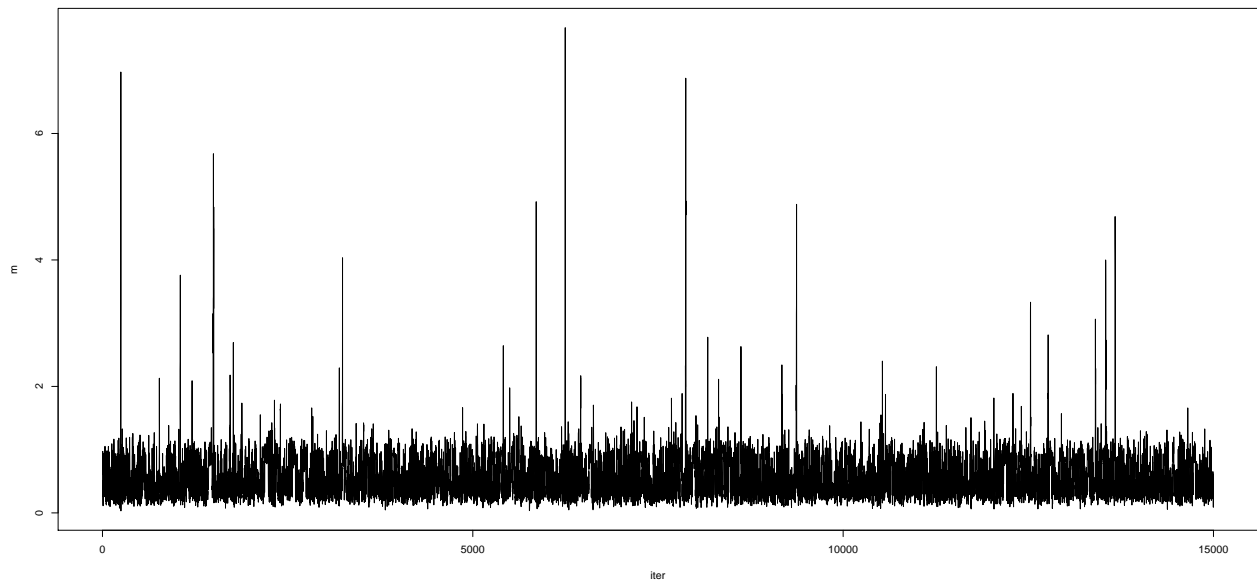
```
matplot(draws_separate_mgamma$e, type = c("l"), col = 1:num_chains,
        xlab = 'iter', ylab = 'e')
```



```
matplot(draws_separate_mgamma$l, type = c("l"), col = 1:num_chains,
        xlab = 'iter', ylab = 'l')
```



```
matplot(draws_separate_mgamma$m, type = c("l"), col = 1:num_chains,
        xlab = 'iter', ylab = 'm')
```



## 7.3 Bayesian Online Changepoint Detection

### 7.3.1 Posterior Heatmaps

#### 7.3.1.1 Analytical Solution

```
t_draws <- t_draws_analytic
r = matrix(0, nrow = max(t_draws), ncol=dim(t_draws)[1])
for (t in 1:dim(t_draws)[1]) {
  for (i in 1:dim(t_draws)[2]) {
    r[t_draws[t,i],t] <- r[t_draws[t,i],t] + 1
```

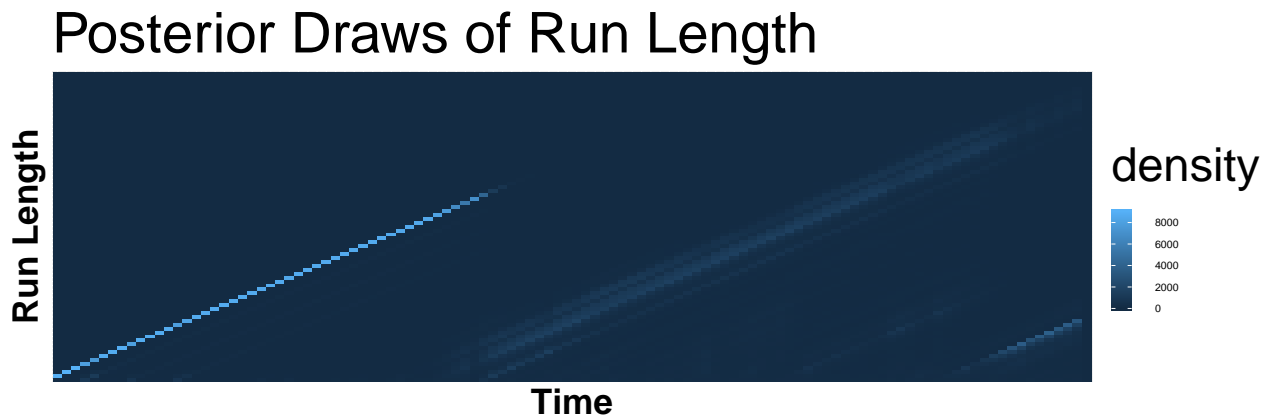
```

}
}

x <- paste0("t", seq(1,dim(t_draws)[1]))
y <- paste0("r", seq(1,max(t_draws)))
data <- expand.grid(X=x, Y=y)
data$density <- c(t(r))

# Heatmap
ggplot(data, aes(X, Y, fill= density)) +
  geom_tile() + theme(
    title = element_text(size=40),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.title=element_text(size=30,face="bold"),
    axis.ticks = element_blank()
  ) + labs(title = "Posterior Draws of Run Length",
           x = "Time", y = "Run Length")

```



### 7.3.1.2 MCMC Solution

```

t_draws <- t_draws_mcmc
r = matrix(0, nrow = max(t_draws), ncol=dim(t_draws)[1])
for (t in 1:dim(t_draws)[1]) {
  for (i in 1:dim(t_draws)[2]) {
    r[t_draws[t,i],t] <- r[t_draws[t,i],t] + 1
  }
}

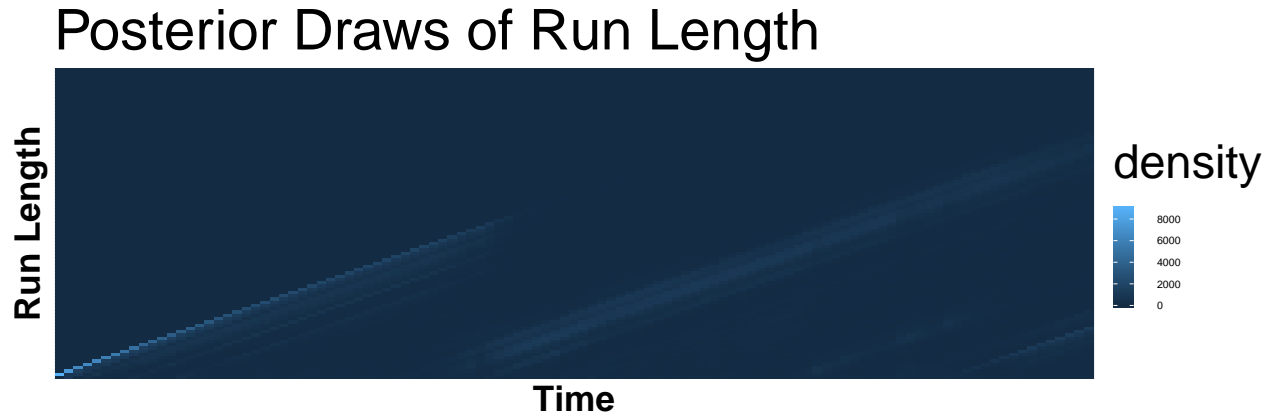
x <- paste0("t", seq(1,dim(t_draws)[1]))
y <- paste0("r", seq(1,max(t_draws)))
data <- expand.grid(X=x, Y=y)
data$density <- c(t(r))

# Heatmap
ggplot(data, aes(X, Y, fill= density)) +
  geom_tile() + theme(
    title = element_text(size=40),
    axis.text.x = element_blank(),

```



```
axis.text.y = element_blank(),
axis.title=element_text(size=30,face="bold"),
axis.ticks = element_blank()
) + labs(title = "Posterior Draws of Run Length",
        x = "Time", y = "Run Length")
```



### 7.3.2 Convergence Diagnostics

```
print(fit_analytic,
      probs = c(0.025, 0.5, 0.975),
      digits_summary = 2, include = TRUE)
```

```
## Inference for Stan model: online_analytic.
## 5 chains, each with iter=2000; warmup=200; thin=1;
## post-warmup draws per chain=1800, total post-warmup draws=9000.
##
##          mean se_mean      sd  2.5% 50% 97.5% n_eff Rhat
## meas[1]   1.99   0.00   0.10   2.00  2   2.00  8876   1
## meas[2]   2.99   0.00   0.11   3.00  3   3.00  9088   1
## meas[3]   3.99   0.00   0.14   4.00  4   4.00  9067   1
## meas[4]   4.53   0.01   1.25   1.00  5   5.00  9015   1
## meas[5]   5.34   0.02   1.48   2.00  6   6.00  9165   1
## meas[6]   6.82   0.01   0.83   3.00  7   7.00  9337   1
## meas[7]   7.84   0.01   0.85   4.00  8   8.00  9052   1
## meas[8]   8.87   0.01   0.78   5.00  9   9.00  8219   1
## meas[9]   9.10   0.03   2.56   1.00 10  10.00  9081   1
## meas[10] 10.74   0.01   1.39   7.00 11  11.00  9300   1
## meas[11] 11.74   0.02   1.41   8.00 12  12.00  8559   1
## meas[12] 12.74   0.02   1.45   9.00 13  13.00  9088   1
## meas[13] 13.75   0.01   1.39  10.00 14  14.00  9393   1
## meas[14] 13.47   0.04   4.19   1.00 15  15.00  9010   1
## meas[15] 15.01   0.04   3.41   2.00 16  16.00  9178   1
## meas[16] 16.56   0.02   2.19   8.00 17  17.00  9028   1
## meas[17] 17.55   0.02   2.25   9.00 18  18.00  8594   1
## meas[18] 18.59   0.02   2.15  10.00 19  19.00  9172   1
## meas[19] 19.60   0.02   2.10  11.00 20  20.00  9080   1
## meas[20] 20.57   0.02   2.23  12.00 21  21.00  8203   1
```

## meas[21]	21.56	0.02	2.26	13.00	22	22.00	8697	1
## meas[22]	22.51	0.03	2.54	14.00	23	23.00	8655	1
## meas[23]	23.12	0.04	3.93	6.00	24	24.00	8137	1
## meas[24]	24.54	0.03	2.52	16.00	25	25.00	9021	1
## meas[25]	25.53	0.03	2.53	17.00	26	26.00	9160	1
## meas[26]	25.98	0.05	4.50	4.00	27	27.00	8928	1
## meas[27]	27.52	0.03	2.73	19.00	28	28.00	8703	1
## meas[28]	28.44	0.03	2.95	20.00	29	29.00	8717	1
## meas[29]	29.41	0.03	3.10	21.00	30	30.00	9091	1
## meas[30]	30.40	0.03	3.10	22.00	31	31.00	9036	1
## meas[31]	31.39	0.04	3.44	23.00	32	32.00	9081	1
## meas[32]	32.47	0.03	2.90	24.00	33	33.00	8684	1
## meas[33]	33.22	0.04	3.98	20.00	34	34.00	8913	1
## meas[34]	34.19	0.04	4.27	21.00	35	35.00	9099	1
## meas[35]	35.40	0.04	3.51	27.00	36	36.00	8585	1
## meas[36]	36.42	0.03	3.26	28.00	37	37.00	9064	1
## meas[37]	37.20	0.05	4.28	24.00	38	38.00	8046	1
## meas[38]	37.47	0.07	6.80	2.98	39	39.00	9111	1
## meas[39]	39.01	0.05	5.15	26.00	40	40.00	8954	1
## meas[40]	39.82	0.06	5.86	18.00	41	41.00	8951	1
## meas[41]	40.75	0.07	6.03	19.00	42	42.00	8467	1
## meas[42]	40.08	0.10	9.81	2.00	43	43.00	8918	1
## meas[43]	38.77	0.14	13.04	2.00	44	44.00	8385	1
## meas[44]	35.33	0.18	16.82	2.00	45	45.00	8821	1
## meas[45]	30.78	0.21	19.27	2.00	46	46.00	8791	1
## meas[46]	38.84	0.17	15.64	4.00	47	47.00	8719	1
## meas[47]	26.82	0.21	20.16	1.00	15	48.00	8940	1
## meas[48]	13.19	0.16	14.74	2.00	8	49.00	8838	1
## meas[49]	12.10	0.12	11.70	2.00	9	50.00	9060	1
## meas[50]	9.54	0.07	6.66	2.00	9	18.00	9092	1
## meas[51]	10.57	0.06	5.35	3.00	10	18.00	8742	1
## meas[52]	11.51	0.05	4.84	4.00	11	18.00	9012	1
## meas[53]	11.53	0.04	3.81	4.00	12	18.00	8978	1
## meas[54]	11.82	0.04	3.89	3.00	13	18.00	8982	1
## meas[55]	14.03	0.05	4.46	2.00	14	20.00	9231	1
## meas[56]	15.10	0.05	4.47	3.00	16	21.00	9028	1
## meas[57]	15.66	0.05	4.40	3.00	16	22.00	8967	1
## meas[58]	17.06	0.05	5.01	2.00	18	24.00	9273	1
## meas[59]	18.21	0.05	5.18	3.00	19	25.00	9085	1
## meas[60]	19.14	0.06	5.61	4.00	20	27.00	8417	1
## meas[61]	19.96	0.06	5.66	2.00	21	27.00	9093	1
## meas[62]	21.08	0.06	5.52	4.00	22	28.00	9171	1
## meas[63]	22.11	0.06	5.47	4.00	23	29.00	9131	1
## meas[64]	23.11	0.06	5.49	5.00	24	30.00	8973	1
## meas[65]	23.40	0.07	6.28	3.00	25	30.00	9299	1
## meas[66]	24.39	0.07	6.30	3.00	26	31.00	9134	1
## meas[67]	24.64	0.08	7.07	3.00	26	32.00	8696	1
## meas[68]	25.84	0.07	6.94	4.00	28	33.00	8630	1
## meas[69]	25.80	0.08	8.00	3.00	28	34.00	8927	1
## meas[70]	25.59	0.10	9.10	2.00	29	34.00	9089	1
## meas[71]	24.50	0.11	10.54	2.98	30	35.00	8740	1
## meas[72]	28.34	0.09	8.53	3.00	31	36.00	8680	1
## meas[73]	29.64	0.09	8.25	4.00	32	38.00	8967	1
## meas[74]	29.72	0.10	9.15	4.00	33	38.00	9053	1

```

## meas[75] 29.41 0.11 10.30 3.00 34 39.00 8154 1
## meas[76] 28.34 0.12 11.75 3.00 35 40.00 9114 1
## meas[77] 30.48 0.12 10.91 4.00 36 41.00 8949 1
## meas[78] 32.42 0.11 10.26 5.00 37 42.00 9022 1
## meas[79] 31.65 0.12 11.58 4.00 38 43.00 8902 1
## meas[80] 34.54 0.11 10.64 3.00 39 44.00 9165 1
## meas[81] 35.62 0.13 11.79 3.00 40 45.00 8269 1
## meas[82] 33.68 0.16 15.05 2.00 41 47.00 8918 1
## meas[83] 35.99 0.15 13.89 4.00 42 48.00 9130 1
## meas[84] 38.02 0.14 13.13 4.00 43 49.00 8881 1
## meas[85] 37.61 0.15 14.41 5.00 44 49.02 8672 1
## meas[86] 39.27 0.15 14.02 5.00 45 51.00 8647 1
## meas[87] 40.70 0.14 13.69 6.00 46 52.00 9158 1
## meas[88] 41.72 0.15 13.69 6.00 47 53.00 8559 1
## meas[89] 42.90 0.14 13.62 6.00 48 54.00 8928 1
## meas[90] 43.07 0.16 14.56 6.00 49 55.00 8516 1
## meas[91] 37.74 0.19 18.60 2.00 49 56.00 9126 1
## meas[92] 37.18 0.20 19.15 3.98 50 57.00 9315 1
## meas[93] 41.27 0.19 18.38 2.00 52 58.00 9305 1
## meas[94] 43.23 0.20 18.50 2.00 53 59.00 8650 1
## meas[95] 45.59 0.18 17.58 3.00 54 60.00 9121 1
## meas[96] 41.40 0.21 19.83 3.00 54 61.00 9028 1
## meas[97] 44.86 0.21 19.35 2.00 56 62.00 8860 1
## meas[98] 45.33 0.21 20.57 2.00 57 63.00 9190 1
## meas[99] 43.40 0.25 23.23 2.00 58 64.00 8939 1
## meas[100] 47.98 0.23 20.94 3.00 59 65.00 8338 1
## meas[101] 45.57 0.25 23.46 3.00 60 65.00 8705 1
## meas[102] 39.93 0.28 26.36 2.00 58 66.00 8908 1
## meas[103] 32.59 0.29 27.21 3.00 12 67.00 8573 1
## meas[104] 24.65 0.27 25.17 3.00 9 68.00 8779 1
## meas[105] 18.49 0.23 20.97 4.00 10 68.00 8600 1
## meas[106] 26.90 0.27 25.45 3.00 11 70.00 8643 1
## meas[107] 21.44 0.23 21.81 4.00 12 70.00 9356 1
## meas[108] 17.91 0.19 17.53 4.00 13 70.00 8323 1
## meas[109] 23.19 0.24 21.87 3.98 14 72.00 8337 1
## meas[110] 19.78 0.19 17.67 4.00 15 71.00 9028 1
## meas[111] 24.20 0.23 21.40 3.00 16 74.00 8813 1
## lp__ 0.00 NaN 0.00 0.00 0 0.00 NaN NaN
##
## Samples were drawn using (diag_e) at Sun Dec 8 23:44:04 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

```

For analytical solution there is no need for convergence diagnostics because we do not have any random parameters.

```

print(fit_mcmc, pars = c('meas'),
      probs = c(0.025, 0.5, 0.975),
      digits_summary = 2, include = TRUE)

```

```

## Inference for Stan model: online_mcmc.
## 5 chains, each with iter=2000; warmup=200; thin=1;

```

## post-warmup draws per chain=1800, total post-warmup draws=9000.

##

##		mean	se_mean	sd	2.5%	50%	97.5%	n_eff	Rhat
##	meas[1]	1.99	0.00	0.10	2	2	2	9190	1
##	meas[2]	2.72	0.01	0.63	1	3	3	7152	1
##	meas[3]	3.57	0.01	0.76	2	4	4	6992	1
##	meas[4]	4.42	0.01	1.08	1	5	5	6360	1
##	meas[5]	5.28	0.02	1.27	2	6	6	6524	1
##	meas[6]	6.19	0.02	1.41	2	7	7	6695	1
##	meas[7]	7.04	0.02	1.57	2	8	8	7142	1
##	meas[8]	7.75	0.02	1.86	3	9	9	7733	1
##	meas[9]	8.50	0.03	2.36	1	10	10	7563	1
##	meas[10]	9.15	0.03	2.53	2	10	11	8595	1
##	meas[11]	10.02	0.03	2.64	3	11	12	8601	1
##	meas[12]	10.87	0.03	2.77	4	12	13	8171	1
##	meas[13]	11.58	0.03	3.00	4	13	14	8292	1
##	meas[14]	12.14	0.04	3.86	1	14	15	8781	1
##	meas[15]	13.25	0.04	3.75	2	15	16	8588	1
##	meas[16]	14.08	0.04	3.65	3	16	17	8545	1
##	meas[17]	14.98	0.04	3.72	4	16	18	9213	1
##	meas[18]	15.77	0.04	3.93	5	17	19	9340	1
##	meas[19]	16.29	0.04	4.29	5	18	20	9305	1
##	meas[20]	17.05	0.05	4.44	6	19	21	9540	1
##	meas[21]	17.53	0.05	4.80	6	19	22	9631	1
##	meas[22]	18.50	0.05	4.86	7	20	23	9554	1
##	meas[23]	19.35	0.06	5.46	2	21	24	8802	1
##	meas[24]	20.39	0.05	5.12	7	22	25	8753	1
##	meas[25]	21.26	0.06	5.19	8	23	26	8506	1
##	meas[26]	21.95	0.06	5.90	3	24	27	9032	1
##	meas[27]	22.96	0.06	5.64	7	24	28	8556	1
##	meas[28]	23.54	0.06	5.88	6	25	29	8436	1
##	meas[29]	24.56	0.06	5.89	8	26	30	8895	1
##	meas[30]	25.31	0.06	6.12	8	27	31	8897	1
##	meas[31]	26.33	0.07	6.28	8	28	32	8689	1
##	meas[32]	27.11	0.07	6.17	10	29	33	8781	1
##	meas[33]	28.00	0.07	6.57	8	30	34	8015	1
##	meas[34]	28.77	0.07	7.16	5	31	35	9256	1
##	meas[35]	29.83	0.08	7.03	8	32	36	8600	1
##	meas[36]	30.81	0.07	6.87	11	33	37	9138	1
##	meas[37]	31.57	0.08	7.41	7	34	38	9188	1
##	meas[38]	31.60	0.10	9.19	2	35	39	8884	1
##	meas[39]	32.92	0.09	8.61	4	36	40	8557	1
##	meas[40]	33.70	0.10	9.16	4	37	41	8552	1
##	meas[41]	34.16	0.10	9.89	4	38	42	8956	1
##	meas[42]	33.32	0.13	12.16	2	38	43	9257	1
##	meas[43]	31.29	0.16	14.60	2	38	44	8850	1
##	meas[44]	28.56	0.18	16.40	2	37	45	8370	1
##	meas[45]	25.78	0.19	17.37	2	32	46	8002	1
##	meas[46]	30.77	0.17	16.34	4	39	47	8753	1
##	meas[47]	24.14	0.19	17.57	1	15	48	8372	1
##	meas[48]	17.05	0.16	15.55	2	11	49	9062	1
##	meas[49]	16.10	0.15	14.25	3	11	50	8537	1
##	meas[50]	12.74	0.11	10.40	3	10	50	8750	1
##	meas[51]	12.87	0.09	9.02	3	11	50	9091	1

## meas[52]	13.42	0.09	8.22	4	12	49	8514	1
## meas[53]	12.90	0.06	5.88	4	13	21	9155	1
## meas[54]	13.01	0.05	4.78	4	13	20	9023	1
## meas[55]	14.96	0.07	6.48	2	15	24	7842	1
## meas[56]	15.91	0.06	6.05	2	16	24	9002	1
## meas[57]	16.34	0.05	5.22	3	17	24	9346	1
## meas[58]	17.54	0.07	6.72	2	18	27	8963	1
## meas[59]	18.70	0.07	7.08	2	19	28	9194	1
## meas[60]	19.73	0.09	8.07	3	20	30	8946	1
## meas[61]	20.56	0.07	6.97	4	21	29	8996	1
## meas[62]	21.44	0.07	6.67	4	22	30	8939	1
## meas[63]	22.33	0.07	6.43	5	23	31	8570	1
## meas[64]	23.33	0.07	6.35	5	24	32	8795	1
## meas[65]	23.73	0.07	6.59	4	25	32	8401	1
## meas[66]	24.80	0.07	6.41	5	26	33	8432	1
## meas[67]	25.08	0.08	7.04	4	27	34	8729	1
## meas[68]	26.20	0.08	7.02	5	28	35	8446	1
## meas[69]	26.52	0.08	7.58	4	29	35	8972	1
## meas[70]	26.59	0.09	8.27	4	29	36	8517	1
## meas[71]	26.49	0.10	9.14	4	30	36	9009	1
## meas[72]	28.74	0.09	8.51	3	31	38	9147	1
## meas[73]	29.92	0.09	8.34	4	32	39	9036	1
## meas[74]	30.08	0.09	8.89	5	33	39	9202	1
## meas[75]	30.08	0.10	9.62	4	34	40	8490	1
## meas[76]	30.16	0.11	10.11	4	34	41	8336	1
## meas[77]	31.59	0.11	9.94	5	35	42	8876	1
## meas[78]	32.76	0.11	9.93	6	37	43	7877	1
## meas[79]	32.85	0.11	10.53	6	37	44	8785	1
## meas[80]	34.66	0.11	10.58	3	39	45	8983	1
## meas[81]	34.61	0.14	12.87	2	40	46	8770	1
## meas[82]	31.48	0.18	16.51	2	40	48	8285	1
## meas[83]	34.51	0.16	15.28	3	42	49	8691	1
## meas[84]	36.27	0.16	14.76	3	43	50	8590	1
## meas[85]	35.84	0.17	16.01	4	44	51	8504	1
## meas[86]	37.80	0.17	15.42	4	45	52	8118	1
## meas[87]	39.47	0.16	14.77	5	46	53	8354	1
## meas[88]	40.75	0.15	14.68	6	47	54	9021	1
## meas[89]	41.89	0.16	14.69	5	48	55	8840	1
## meas[90]	41.82	0.17	15.84	4	49	56	8679	1
## meas[91]	35.31	0.21	20.06	1	47	58	8949	1
## meas[92]	34.86	0.22	20.45	2	47	58	8673	1
## meas[93]	39.75	0.20	19.23	3	51	60	8946	1
## meas[94]	42.53	0.21	18.74	3	52	61	8349	1
## meas[95]	43.87	0.20	18.59	3	53	61	8862	1
## meas[96]	39.23	0.23	21.09	2	52	63	8197	1
## meas[97]	43.20	0.22	20.25	2	55	64	8445	1
## meas[98]	45.32	0.22	20.14	3	56	65	8486	1
## meas[99]	46.24	0.22	20.70	3	57	66	8896	1
## meas[100]	47.95	0.21	20.21	4	58	67	8963	1
## meas[101]	48.31	0.23	21.24	4	59	68	8680	1
## meas[102]	47.18	0.26	22.86	4	60	68	7976	1
## meas[103]	44.33	0.27	24.71	4	59	69	8325	1
## meas[104]	42.18	0.28	25.86	4	58	69	8543	1
## meas[105]	39.06	0.30	26.85	4	51	70	8128	1

```
## meas[106] 42.68    0.28 26.37    5 60    71 9022    1
## meas[107] 39.69    0.29 26.95    5 47    72 8584    1
## meas[108] 36.36    0.30 26.92    5 18    73 8072    1
## meas[109] 40.21    0.29 27.12    5 37    74 8902    1
## meas[110] 36.75    0.29 26.83    6 19    75 8390    1
## meas[111] 40.67    0.29 27.16    5 25    76 8484    1
##
## Samples were drawn using NUTS(diag_e) at Sun Dec  8 23:44:54 2019.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

As we can see that the Rhat values for all parameters and changepoint estimations are 1, that is our model fitted very well.

The effective sample sizes are looking also good. The ration  $n_{\text{eff}} / n_{\text{transitions}}$  is greater than 0.001 for all parameters.

Here we check the tree depth, E-BFMI and divergences. All are looking good.

```
check_tredepth(fit_mcmc)
```

```
## 0 of 9000 iterations saturated the maximum tree depth of 10.
```

```
check_energy(fit_mcmc)
```

```
## E-BFMI indicated no pathological behavior.
```

```
check_divergences(fit_mcmc)
```

```
## 0 of 9000 iterations ended with a divergence.
```

## 8 Bayesian Online Changepoint Detection (Syntetic Data)

In this section we have expanded our work to show how effectively Bayesian online changepoint detection algorithm is find more than one changepoints.

```
fit_syntetic <- stan("StanFiles/online_analytic.stan",
                    data=data_synthetic, algorithm = "Fixed_param",
                    iter=2000, warmup=200, chains=1)
```

```
draws_syntetic <- as.data.frame(fit_syntetic)
t_draws_syntetic <- t(draws_syntetic)
```

```
t_draws <- t_draws_syntetic
r = matrix(0, nrow = max(t_draws), ncol=dim(t_draws)[1])
for (t in 1:dim(t_draws)[1]) {
  for (i in 1:dim(t_draws)[2]) {
    r[t_draws[t,i],t] <- r[t_draws[t,i],t] + 1
  }
}
```

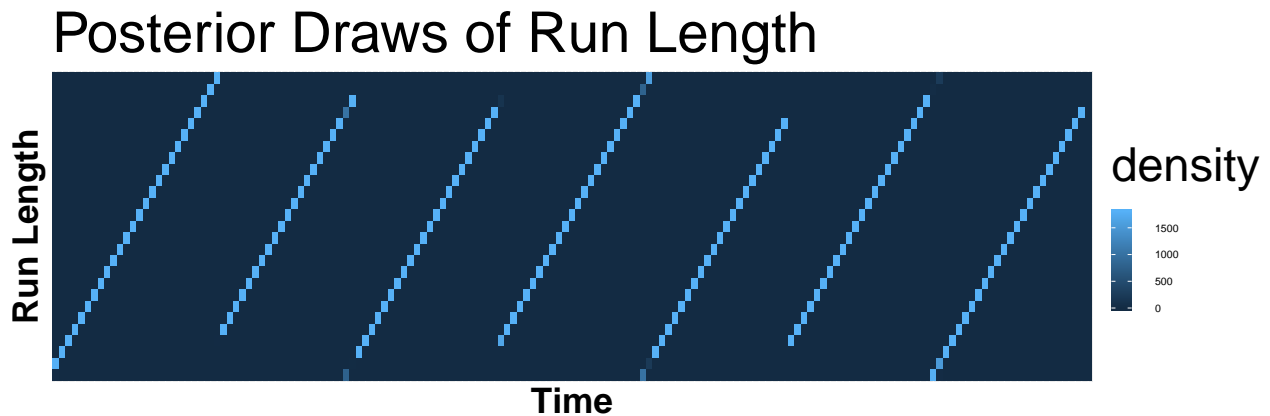
```

}

x <- paste0("t", seq(1,dim(t_draws)[1]))
y <- paste0("r", seq(1,max(t_draws)))
data <- expand.grid(X=x, Y=y)
data$density <- c(t(r))

# Heatmap
ggplot(data, aes(X, Y, fill= density)) +
  geom_tile() + theme(
    title = element_text(size=40),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.title=element_text(size=30,face="bold"),
    axis.ticks = element_blank()
  ) + labs(title = "Posterior Draws of Run Length",
           x = "Time", y = "Run Length")

```



## 9 Posterior Prediction

Because of the nature of our problem, posterior prediction is not possible. The problem is a time series problem but we are not trying to predict the future sequences.

## 10 Conclusion

The multiple changepoint model (MCM) and hierarchical multiple changepoint model (HMCM) are performing in a similar fashion. However, when we compare the standard deviations of the parameters for MCM and HMCM, we see that HMCM has slightly lower standard deviation than MCM. It can also be seen from the histograms of the estimate for second changepoint,  $s_2$ .

In the online changepoint model we have compared our MCMC draws results with analytical solution of the problem.

Overall, Bayesian Online Changepoint Model is superior to other models. It can find any number of change-points in linear time and it converges much faster. In this project we implemented Poisson observation and Gamma underlying process version of the algorithm.

We specifically used conjugate prior model in order to compare with the analytical solution. As a future work, more generic versions of the MCMC solution of the model can be implemented. Because it can be used with models which do not have conjugate priors. This allows more complex models.