Bayesian Data Analysis - Changepoint Detection

Anonymous

2020-01-31

Contents

1 Loaded packages

```
library(rstan)
library(BaM)
```

2 Introduction

The changepoint detection problem is one of the essential problems in time series data. During the time series analysis, there could be changes in the underlying dynamics, and it is valuable to detect them. Given sequential data, we might be interested in the changes in the data distribution, especially these data changes in some way. For example, with the help of effective and efficient changepoint algorithms, heart attacks, failures in the machinery system, or changes in the economic trends could be detected, even prevented. If we assume that data is a result of 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. 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 there may be. We are comparing these models and sharing statistical workflow of these algorithms.

- Poisson-Gamma Multiple Changepoint Detection for two changepoints
- Hierarchical Poisson-Gamma Multiple Changepoint Detection for two changepoints
- Bayesian Online Changepoint Detection[*]

In this report, we give a description of the datasets in Section 3. Section 4 gives the details of the models and the implemented STAN code. We report the prior choices and fit the models in Section 5. Section 6 gives the histograms of the parameters and posterior sampling histograms besides the divergences of parameters. We showed the advantage of the Bayesian Online Changepoint model on synthetic data in Section 7. Finally, our comments and conclusion of our report are given in Section 8.

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

3 Description of the Data

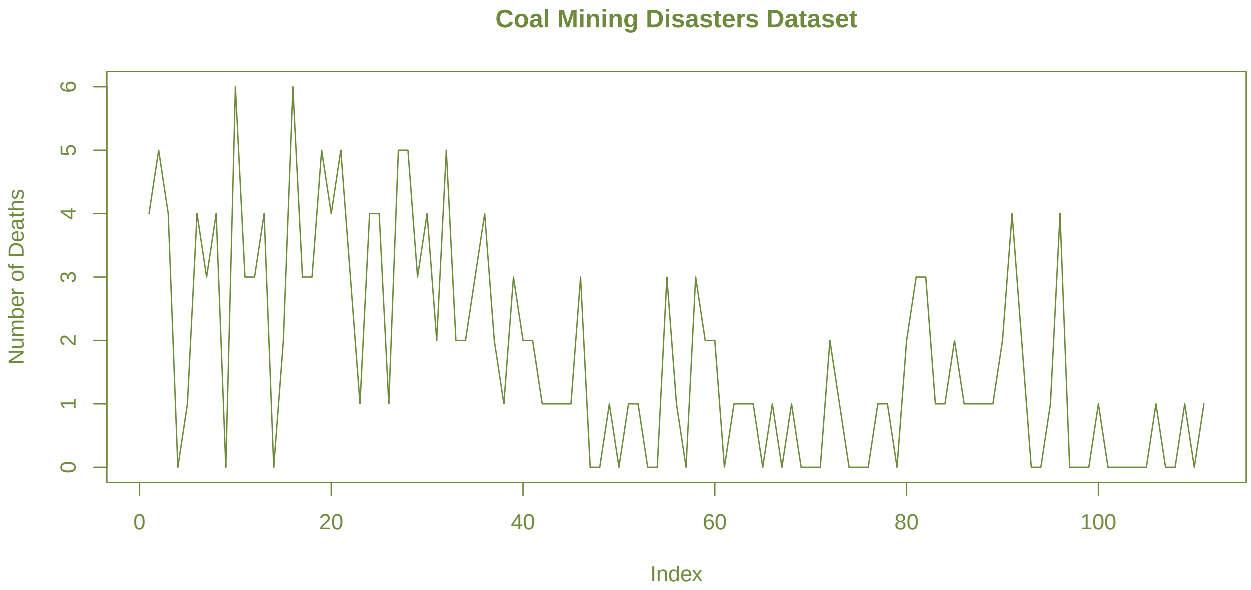
3.1 Coal Mining Disasters Dataset

We are using Coal Mining Disasters dataset from BaM library (BaM: Functions and Datasets for Books by Jeff Gill). The data consists of the number of deadly coal-mining disasters in England per year over for 112 years from 1851 to 1962. It is widely agreed in the statistical literature that a change in the intensity (the expected value of the number of disasters) occurs around the year 1890 after new health and safety regulations were introduced. Therefore it is common to use this dataset in the changepoint detection algorithms in the literature, as we do.

Source: Lynn et al. (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))
```

```
par(bg = NA, fg = color, col.lab = color, col.axis = color, col.main = color)
plot(coal.mining.disasters, type = "l", ylab = 'Number of Deaths',
     xlab = 'Index', main = 'Coal Mining Disasters Dataset')
```

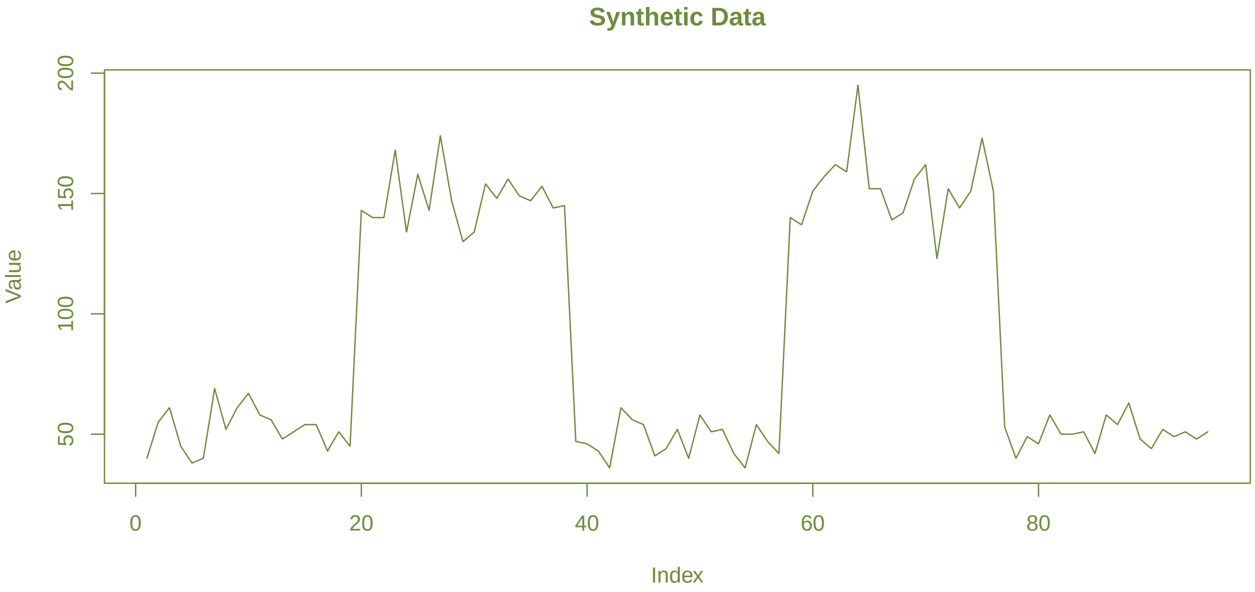


3.2 Synthetic Data

The coal-mining disasters dataset is the choice of the literature. However, it would be clearer to show the effectiveness of the developed models if the model is capable of capturing more than 2 changepoints. In this work, to show the performance and the efficiency of the Bayesian Online Changepoint Model (BOCM) on multiple changepoints, we have used the following synthetic data:

```
synthetic_data <- c(read.csv("synthetic_data.csv", header = FALSE))$V1
data_synthetic <- list(T = length(synthetic_data),
                       D = synthetic_data,
                       H = 0.01,
                       mean_D = mean(synthetic_data))
```

```
par(bg = NA, fg = color, col.lab = color, col.axis = color, col.main = color)
plot(synthetic_data, type = "l", ylab="Value",
     xlab = 'Index', main='Synthetic Data')
```



4 Model Details and Stan Codes

4.1 Poisson-Gamma Multiple Changepoint Model

The assumptions of this model are:

- There are two change points on the time series data; that is, there are three intervals.
- The data come 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:

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

$$D_t \sim \text{ Poisson } (\lambda) \quad \lambda = \begin{cases} e & \text{if } \ t < s_1 \\ l & \text{if } \ s_1 <= t < s_2 \\ m & \text{else} \end{cases} \tag{1}$$

The likelihood of the model is:

$$\begin{aligned} p(D|e,l,m) &= \sum_{s_1=1}^{T} \sum_{s_2=s_1}^{T} p(s_1, s_2, D \mid e, l, m) \qquad (2) \\ &= \sum_{s_1=1}^{T} \sum_{s_2=s_1}^{T} p(s_1) p(s_2 \mid s_1) p(D \mid s_1, s_2, e, l, m) \qquad (3) \\ &= \sum_{s_1=1}^{T} \sum_{s_2=s_1}^{T} U(s_1 \mid 1, T) U(s_2 \mid s_1, T) \prod_{t=1}^{T} \text{ Poisson } \ (D_t \mid t < s_1 ? \, e : (t < s_2) \ ? \ l : \ m) \qquad (4) \end{aligned}$$

We modified the single Exponential-Poisson changepoint[*,**] model to handle two changepoints, and 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

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

```
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) ;
}
```

4.2 Hierarchical Poisson-Gamma Multiple Changepoint Model

The assumptions of this model are:

- There are two change points on the time series data; that is, there are three intervals.
- The data come from a Poisson distribution in each interval.
- The means of the Poisson distribution generated by underlying Gamma distributions.
- the shape parameters of the priors for the interval means have a common Gamma prior.

The generative process of the model is:

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

$$D_t \sim \text{ Poisson } (\lambda) \quad \lambda = \begin{cases} e & \text{if } \ t < s_1 \\ l & \text{if } \ s_1 <= t < s_2 \\ m & \text{else} \end{cases} \tag{5}$$

The likelihood of the model is almost the same as previous one:

$$\begin{aligned}
p(D|e, l, m, \alpha) &= \sum_{s_1=1}^{T} \sum_{s_2=s_1}^{T} p(s_1, s_2, D \mid e, l, m, \alpha) & (6) \\
&= \sum_{s_1=1}^{T} \sum_{s_2=s_1}^{T} p(s_1) p(s_2 \mid s_1) p(D \mid s_1, s_2, e, l, m, \alpha) & (7) \\
&= \sum_{s_1=1}^{T} \sum_{s_2=s_1}^{T} U(s_1 \mid 1, T) U(s_2 \mid s_1, T) \prod_{t=1}^{T} \text{ Poisson } \ (D_t \mid t < s_1 ? \ e : (t < s_2) \ ? \ l : \ m) & (8)
\end{aligned}$$

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

```
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) ;
}
```

4.3 Bayesian Online Changepoint Detection

In this section, we discuss the online changepoint detection algorithm and its Stan application, according to Ryan P. Adams and David J.C. MacKay's technical report on Bayesian online changepoint detection.

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

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

$$P\left(r_t \mid r_{t-1}\right)= \begin{cases}H\left(r_{t-1}+1\right) & \text{if } r_t=0 \\ 1-H\left(r_{t-1}+1\right) & \text{if } r_t=r_{t-1}+1 \\ 0 & \text{otherwise}\end{cases} \tag{9}$$

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 $H(\tau)=1/\lambda$.

4.3.1 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.

$$P\left(x_{t+1} \mid \boldsymbol{x}_{1:t}\right)=\sum_{r_t} P\left(x_{t+1} \mid r_t, \boldsymbol{x}_t^{(r)}\right) P\left(r_t \mid \boldsymbol{x}_{1:t}\right) \tag{10}$$

where posterior distribution of run length at time is;

$$P\left(r_t \mid \boldsymbol{x}_{1:t}\right)=\frac{P\left(r_t, \boldsymbol{x}_{1:t}\right)}{P\left(\boldsymbol{x}_{1:t}\right)} \tag{11}$$

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

$$\begin{aligned} P\left(r_t, \boldsymbol{x}_{1:t}\right) &= \sum_{r_{t-1}} P\left(r_t, r_{t-1}, \boldsymbol{x}_{1:t}\right) && (12) \\ &= \sum_{r_{t-1}} P\left(r_t, x_t \mid r_{t-1}, \boldsymbol{x}_{1:t-1}\right) P\left(r_{t-1}, \boldsymbol{x}_{1:t-1}\right) && (13) \\ &= \sum_{r_{t-1}} P\left(r_t \mid r_{t-1}\right) P\left(x_t \mid r_{t-1}, \boldsymbol{x}_t^{(r)}\right) P\left(r_{t-1}, \boldsymbol{x}_{1:t-1}\right) && (14) \end{aligned}$$

Ultimately, we want to infer both the run-length posterior distribution $p(r_t \mid \boldsymbol{x}_{1:t})$ and the posterior predictive distribution $p(x_{t+1} \mid \boldsymbol{x}_{1:t})$ 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 implemented 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:

```
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]) + l
      alpha[j+1, t+1] = alpha[j, t] + D[t];
      beta[j+1, t+1] = beta[j, t] + 1;
    }
  }
}

model {

}

generated quantities {
  vector[T] run_length;
  for (t in 1:T)
    run_length[t] = categorical_logit_rng(col(R, t+1)[1:t+1]);
}
```

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

```
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] run_length;
  for (t in 1:T)
    run_length[t] = categorical_logit_rng(col(R, t+1)[1:t+1]);
}
```

5 Prior Choices and Fitting the Models

5.1 Poisson-Gamma Multiple Changepoint Model

5.1.1 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.

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

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

5.1.2 Fitting the Model

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

```
num_chains <- 5
fit_mgamma <- stan('./StanFiles/multiple_changepoint_gamma.stan',
                   data = data,
                   iter = 6000,
                   warmup = 2000,
                   chains = num_chains,
                   control = list(adapt_delta=0.9)
                   )

draws_mgamma <- as.data.frame(fit_mgamma)
```

5.2 Hierarchical Poisson-Gamma Multiple Changepoint Model

5.2.1 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 $\alpha = \bar{D}$ We are using 1 as a scale on all Gamma distribution. So, using the mean of the data as the 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 a uniform distribution over the data length.

5.2.2 Fitting the Model

We fit the model for 5 chains and 6000 iterations with a 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_hmgamma <- as.data.frame(fit_hmgamma)
```

5.3 Bayesian Online Changepoint Detection

5.3.1 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 at $p = H$.

5.3.2 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 2500 iterations with 500 warm-up phase.

```
num_chains <- 5
fit_analytic <- stan("StanFiles/online_analytic.stan",
                     data = data,
                     algorithm = "Fixed_param",
                     iter = 2500,
                     warmup = 500,
                     chains=num_chains
                     )

draws_analytic <- extract(fit_analytic, pars = c("run_length"))$run_length
```

In this part, we use MCMC sampling and find the posterior of the samples with these samples.

We fit the model for 5 chain and 2500 iterations with 500 warm-up phase.

```
num_chains <- 5
fit_mcmc <- stan("StanFiles/online_mcmc.stan",
                 data = data,
                 iter = 2500,
                 warmup = 500,
                 chains = num_chains
                 )

draws_mcmc <- extract(fit_mcmc, pars = c("run_length"))$run_length
```

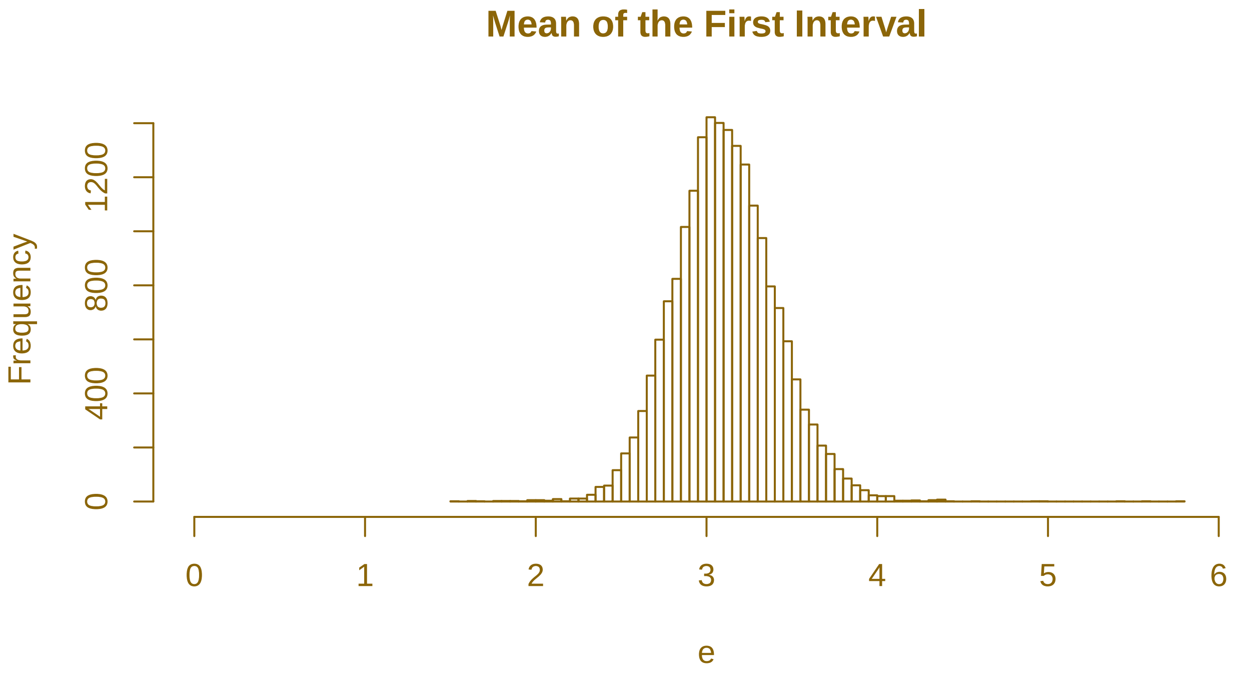
6 Histograms and Divergences of Parameters

6.1 Poisson-Gamma Multiple Changepoint Model

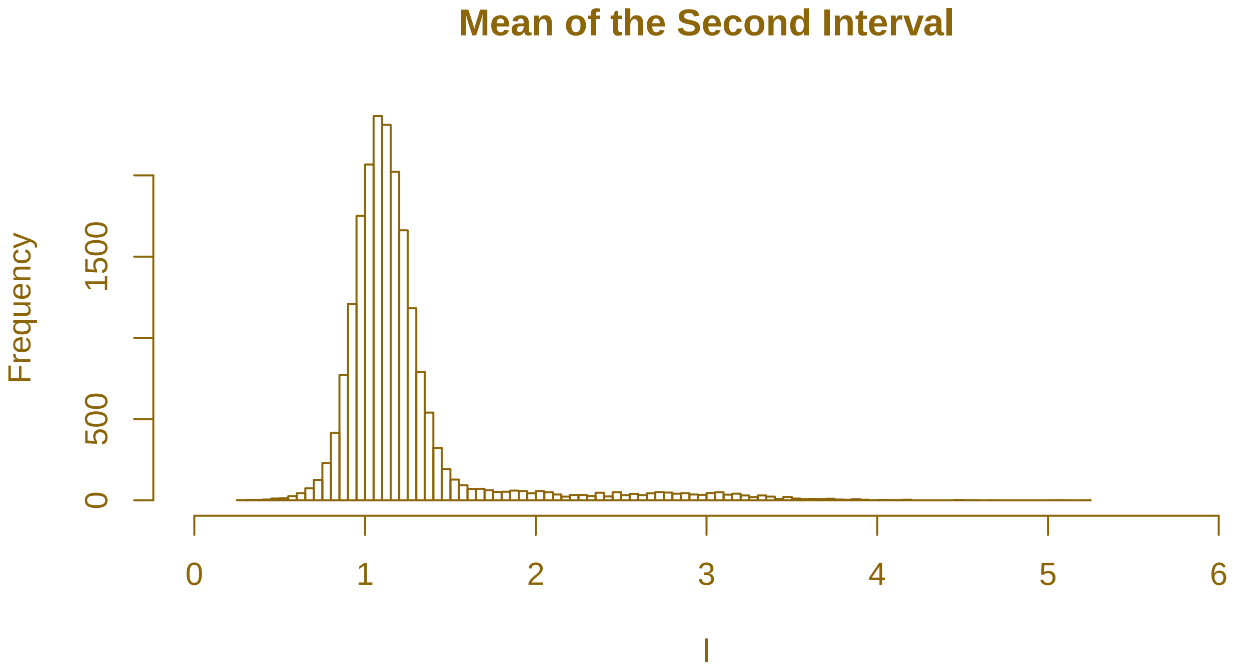
6.1.1 Histogram of the Parameters

The histograms of the parameters are as follows:

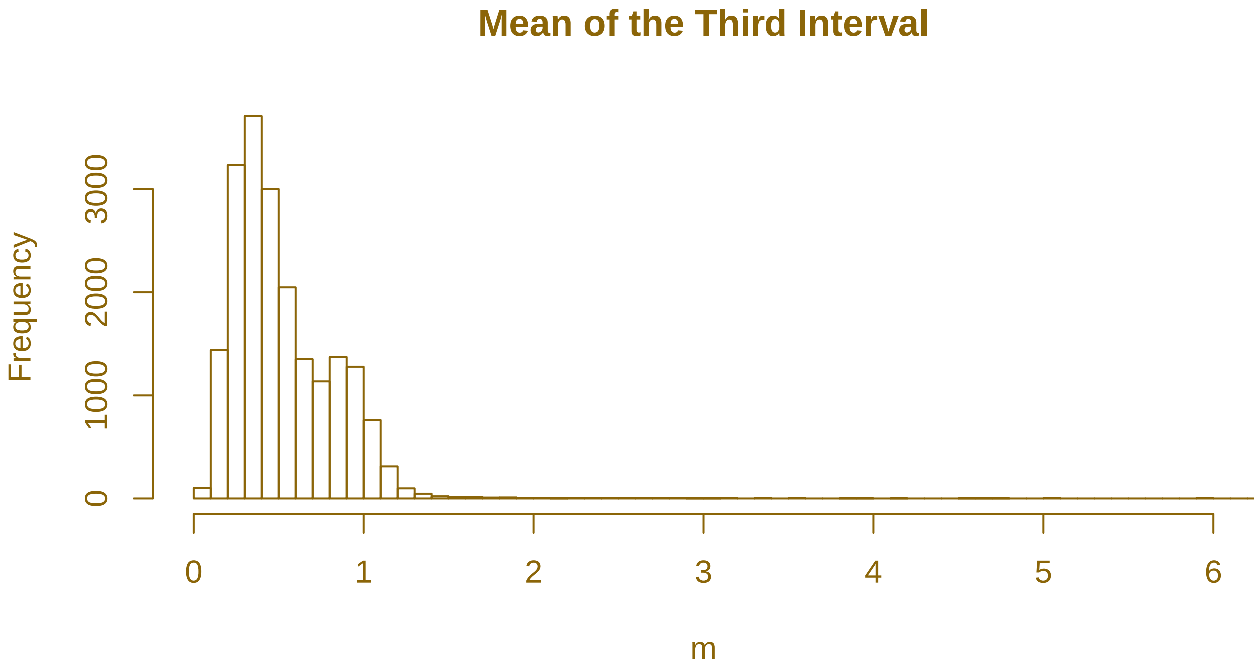
```
par(bg = NA, fg = color, col.lab = color, col.axis = color, col.main = color)
hist(draws_mgamma$e, breaks=100, main='Mean of the First Interval', xlab='e', xlim=c(0,6))
```



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

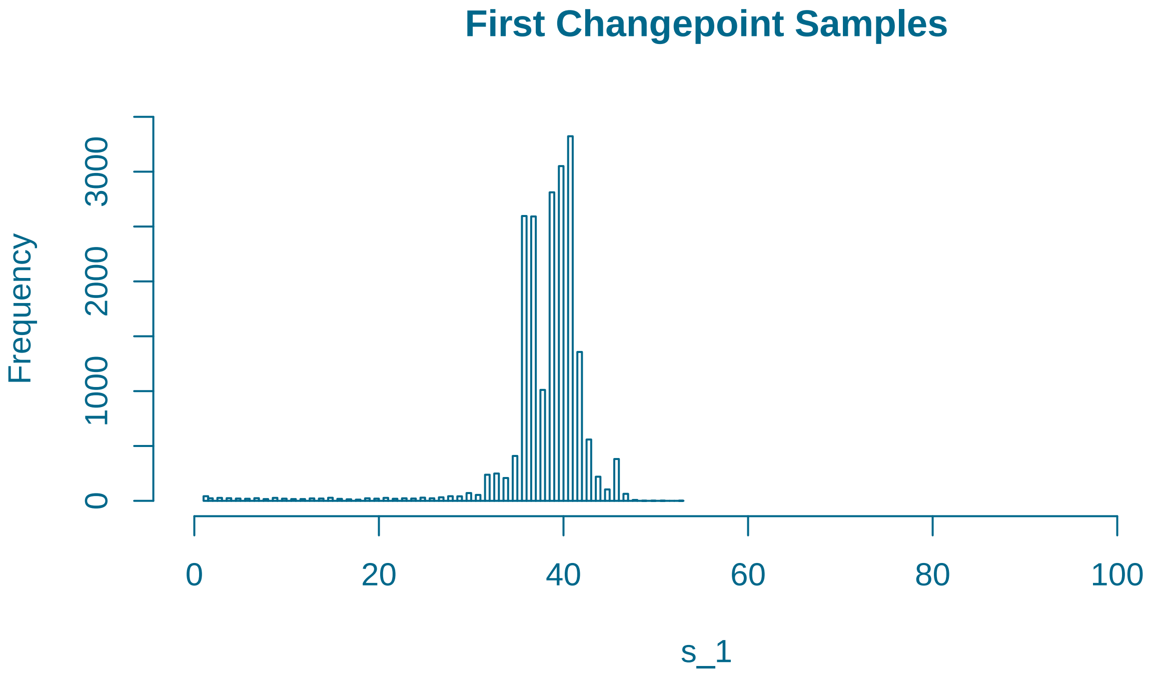


```r
hist(draws_mgamma$m, breaks=100, main='Mean of the Third Interval', xlab='m', xlim=c(0,6))
```

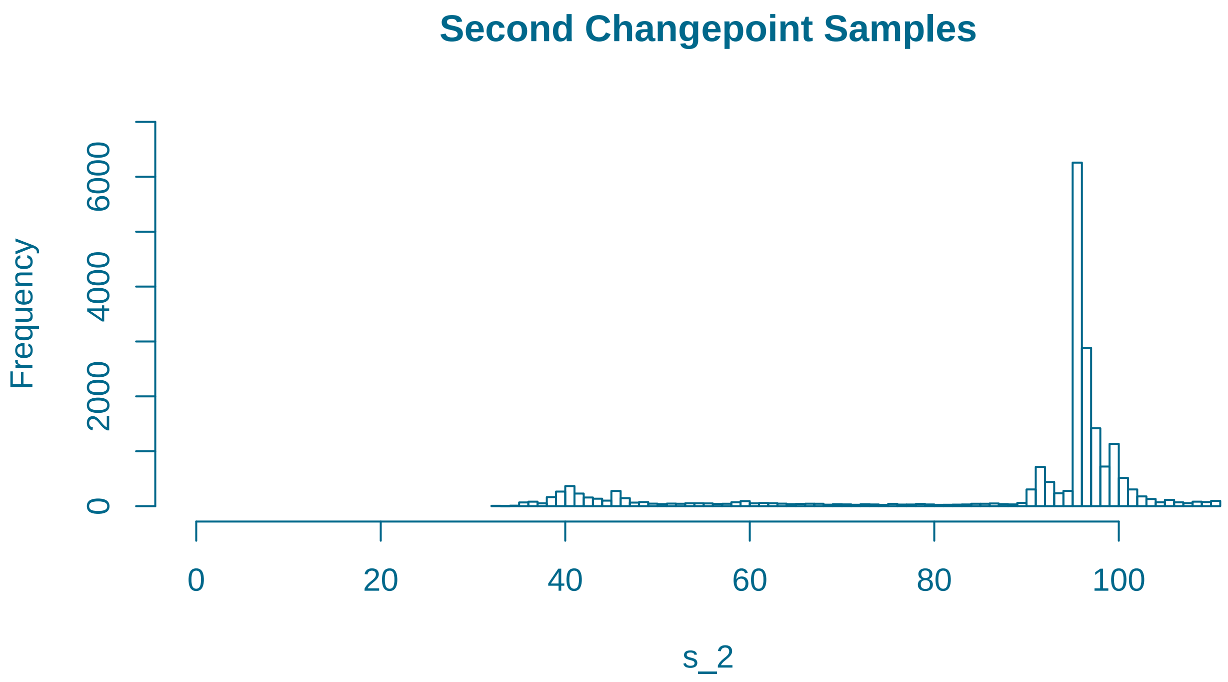


6.1.2 Posterior Sampling

```r
par(bg = NA, fg = color, col.lab = color, col.axis = color, col.main = color)
hist(draws_mgamma$s1s, breaks=100, main='First Changepoint Samples',
     xlab = 's_1', xlim = c(0,data$T), ylim = c(0,3500))
```



```r
hist(draws_mgamma$s2s, breaks=100, main='Second Changepoint Samples',
     xlab = 's_2', xlim = c(0,data$T), ylim = c(0,7000))
```



6.1.3 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=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    3.11    0.00  0.30  2.55  3.10   3.72 11362    1
l    1.22    0.01  0.45  0.79  1.12   2.86  1191    1
m    0.53    0.01  0.31  0.15  0.45   1.11  3459    1
s1s 38.20    0.12  5.34 23.00 39.00  45.00  1866    1
s2s 88.71    0.46 18.73 40.00 96.00 105.00  1627    1

Samples were drawn using NUTS(diag_e) at Fri Jan 31 03:57:31 2020.
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 also looking good. The ration n_eff / n_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 20000 iterations saturated the maximum tree depth of 10.
```

```
check_energy(fit_mgamma)
```

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

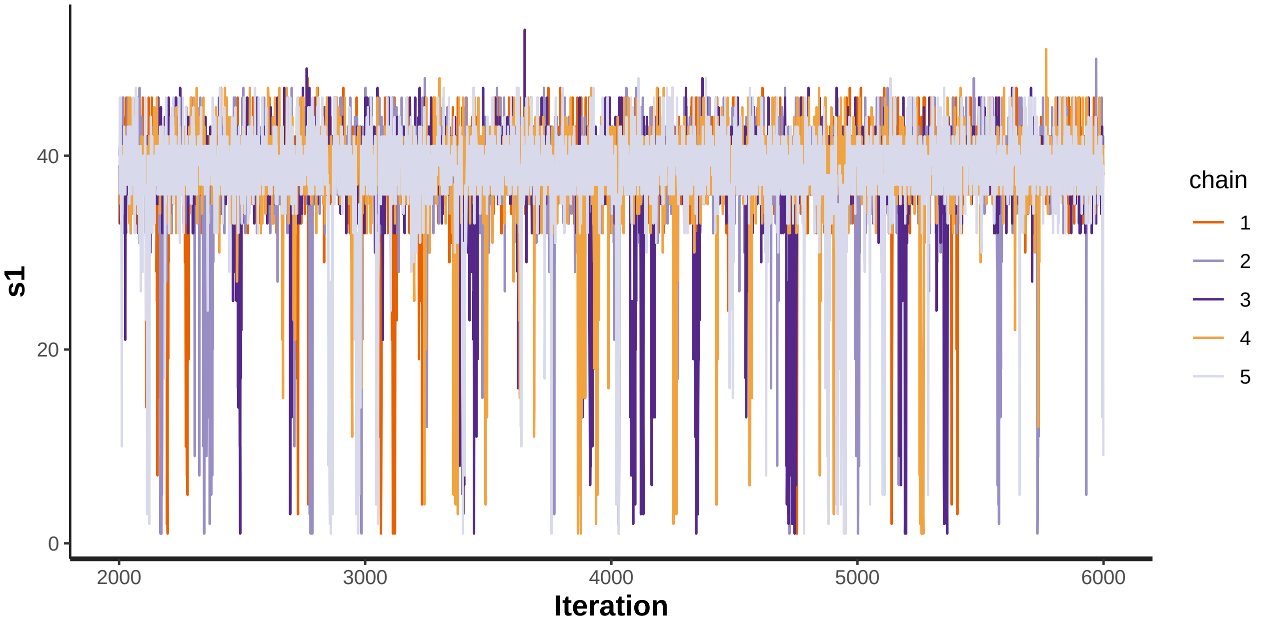
```
check_divergences(fit_mgamma)
```

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

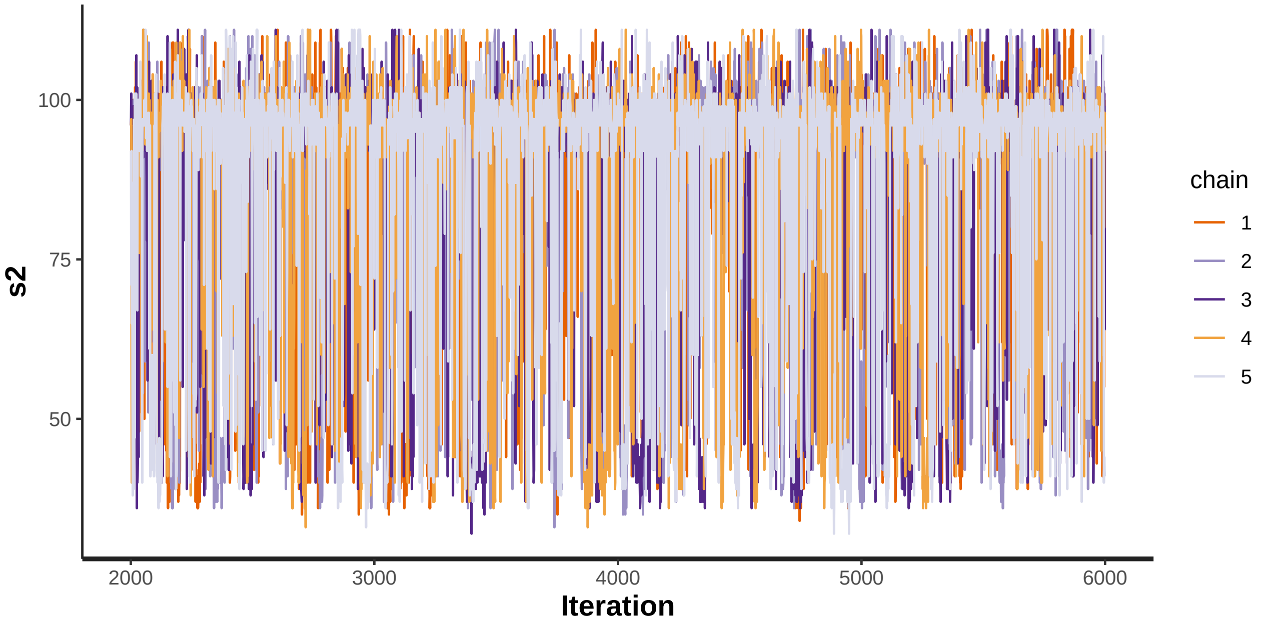
As we can see from the following plots of the samples, the plot of the chains of the mean of the first interval indicates that it is converged. The other means of the second and third intervals are also converged, but there are some divergences.

When we investigate the plots of the chains for the 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 that the divergence of the second changepoint. As the histograms of the changepoints also show, there is a slight probability that the first changepoint is closer to the beginning of 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.

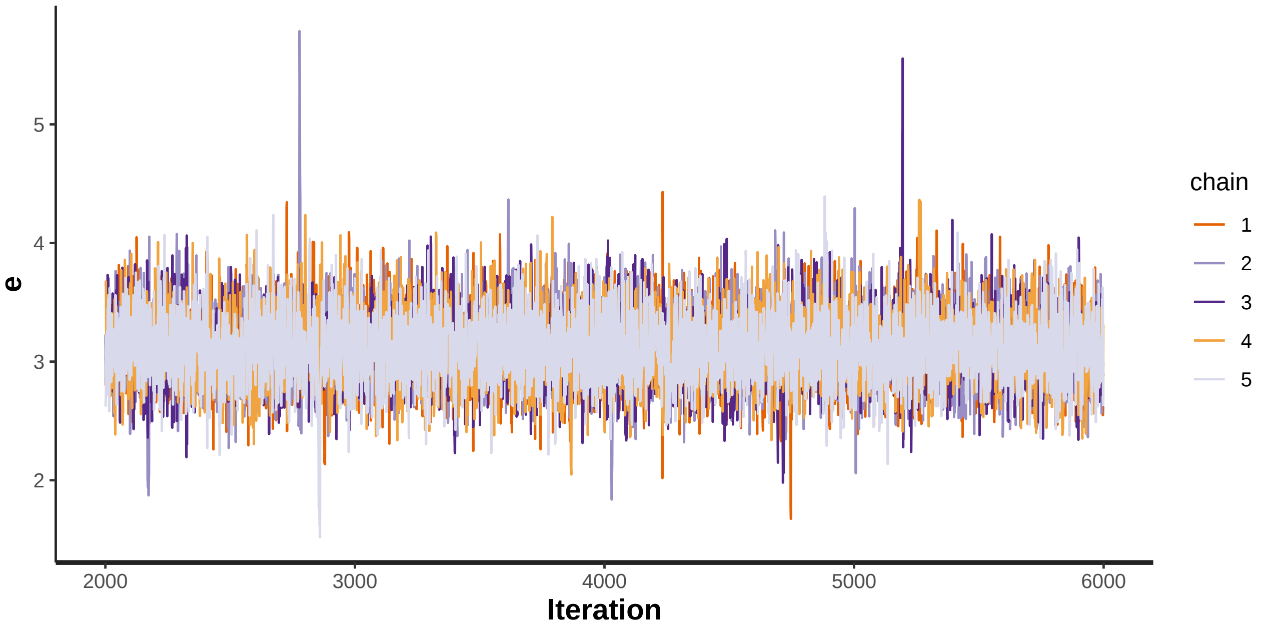
```
traceplot(fit_mgamma, pars = c("s1s")) + labs(x = "Iteration", y = "s1")
```



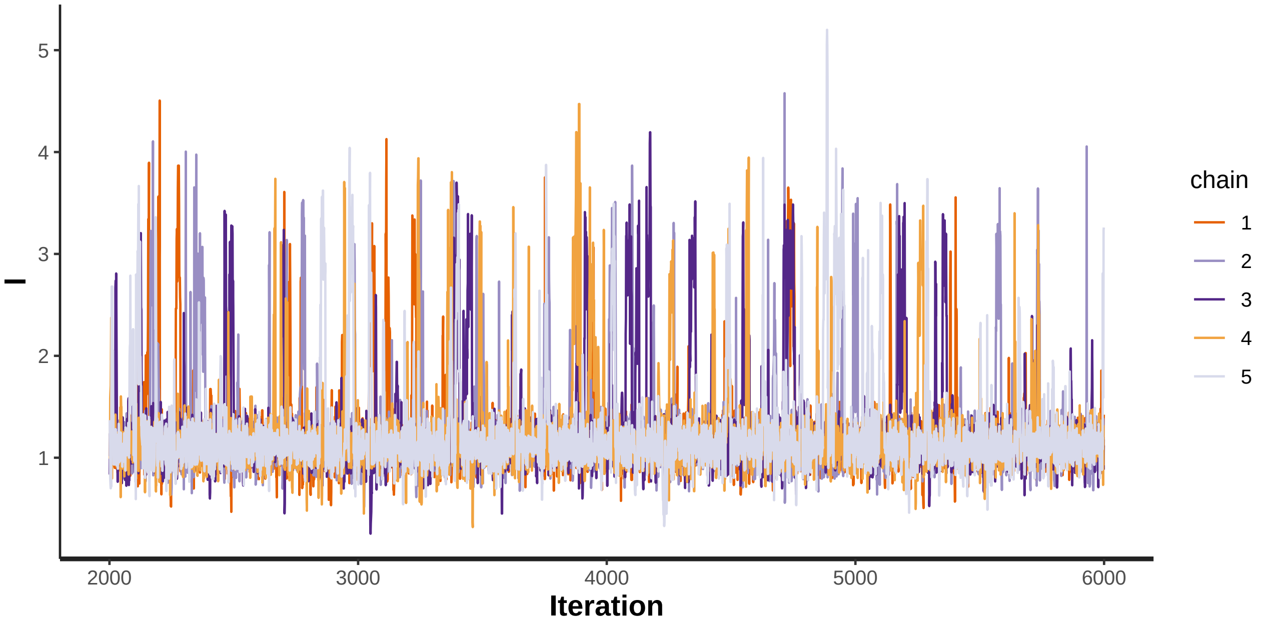
```
traceplot(fit_mgamma, pars = c("s2s")) + labs(x = "Iteration", y = "s2")
```



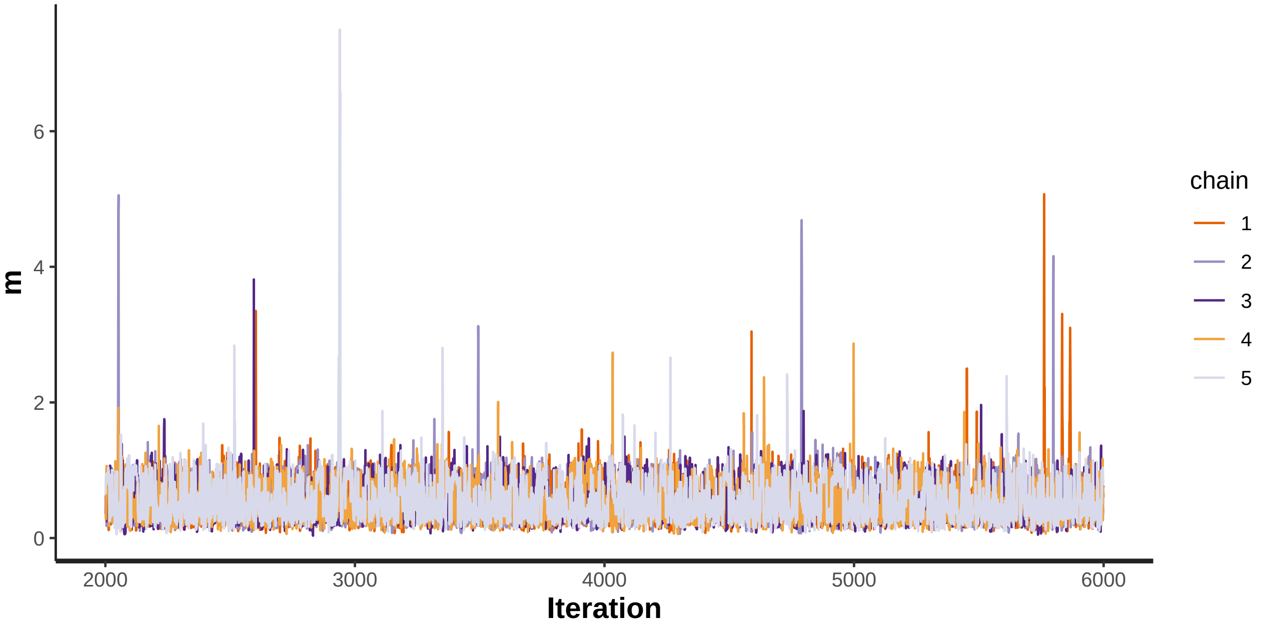
```r
traceplot(fit_mgamma, pars = c("e")) + labs(x = "Iteration", y = "e")
```



```r
traceplot(fit_mgamma, pars = c("l")) + labs(x = "Iteration", y = "l")
```



```r
traceplot(fit_mgamma, pars = c("m")) + labs(x = "Iteration", y = "m")
```

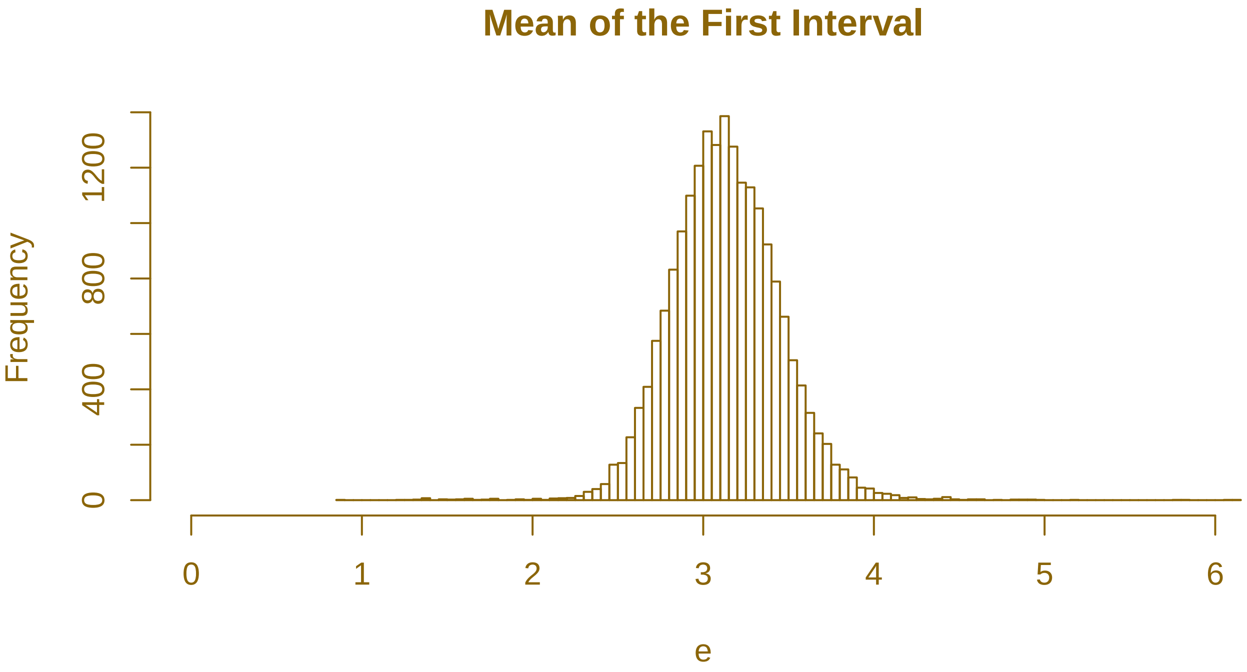


6.2 Hierarchical Poisson-Gamma Multiple Changepoint Model

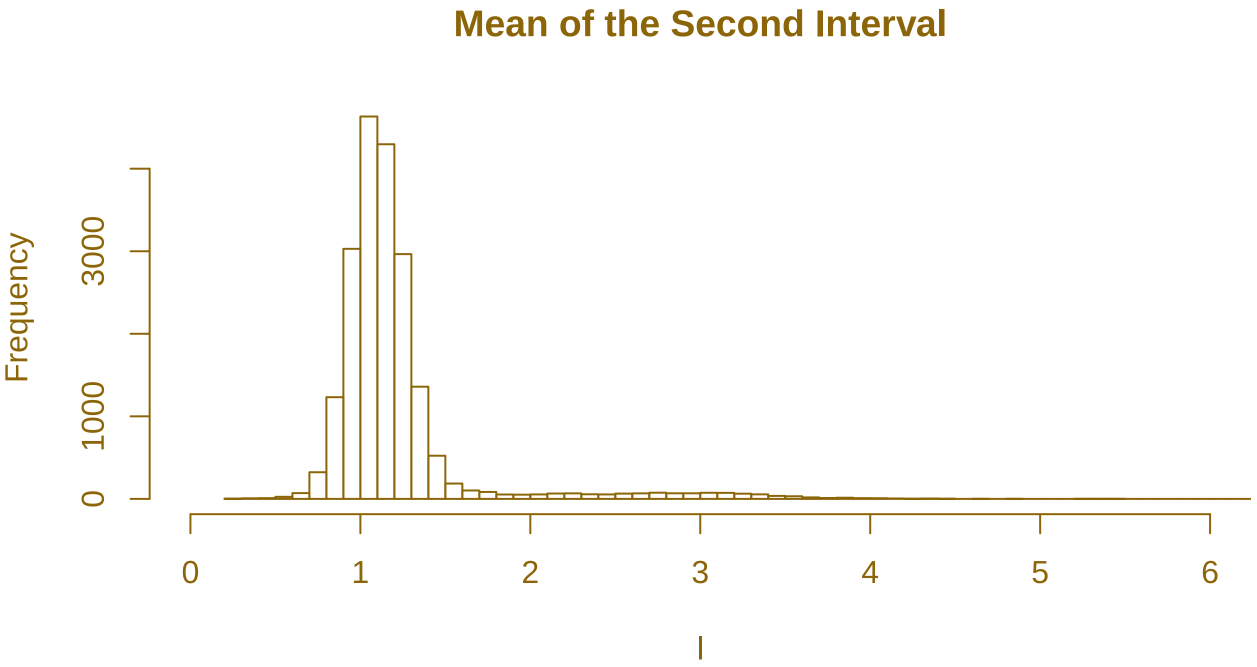
6.2.1 Histogram of Parameters

The histograms of the parameters are as follows:

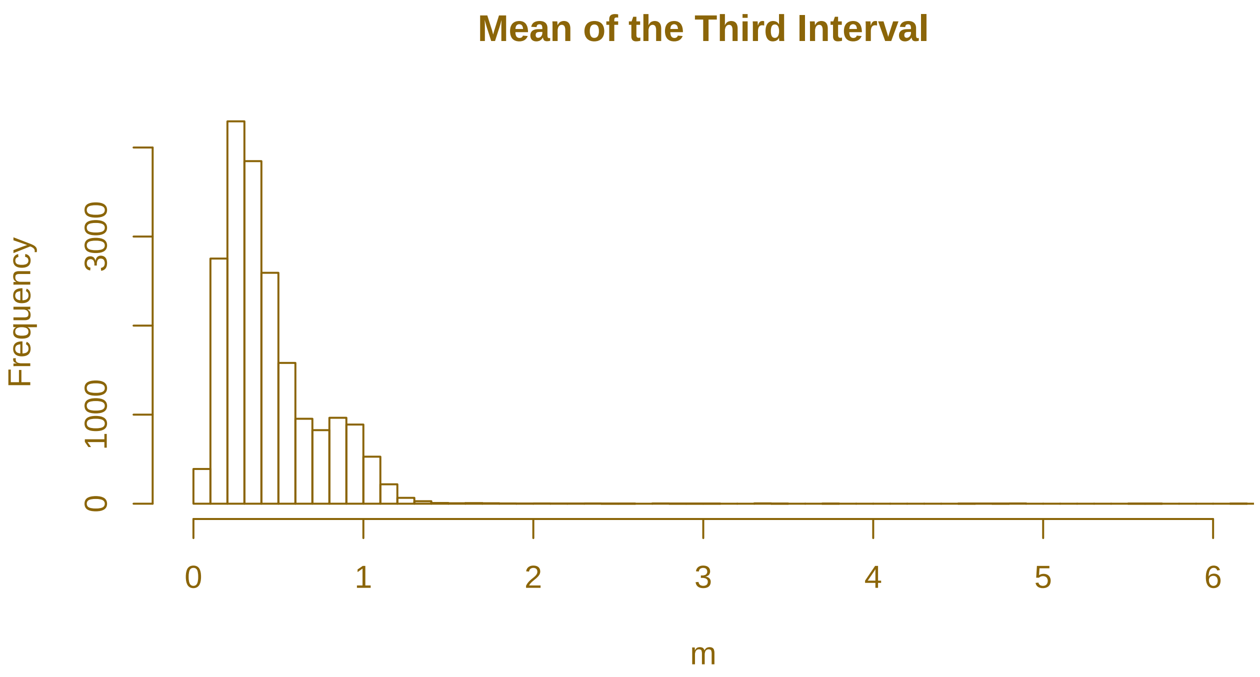
```
par(bg = NA, fg = color, col.lab = color, col.axis = color, col.main = color)
hist(draws_hmgamma$e, breaks=100, main='Mean of the First Interval', xlab='e', xlim=c(0,6))
```



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

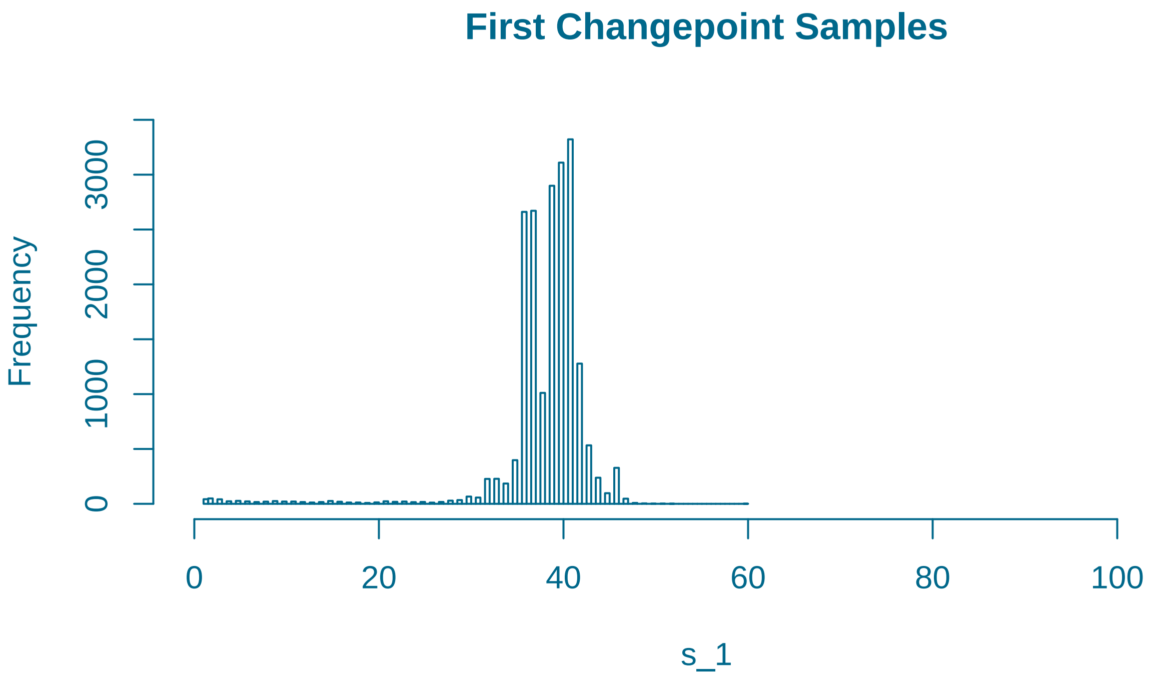


```
hist(draws_hmgamma$m, breaks=100, main='Mean of the Third Interval', xlab='m', xlim=c(0,6))
```

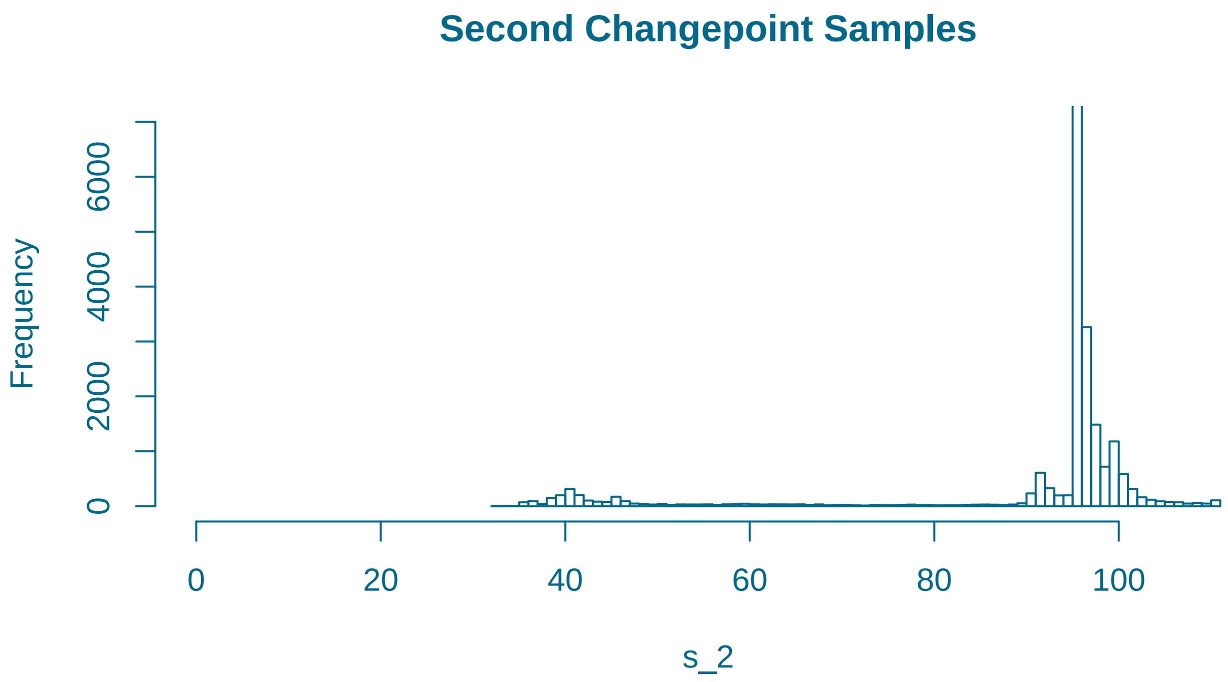


6.2.2 Posterior Sampling

```
par(bg = NA, fg = color, col.lab = color, col.axis = color, col.main = color)
hist(draws_hmgamma$s1s, breaks=100, main='First Changepoint Samples',
     xlab = 's_1', xlim = c(0,data$T), ylim = c(0,3500))
```



```
hist(draws_hmgamma$s2s, breaks=100, main='Second Changepoint Samples',
     xlab = 's_2',xlim = c(0,data$T), ylim = c(0,7000))
```



6.2.3 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  2.56  3.12   3.76  8139    1
l    1.21    0.01  0.45  0.81  1.11   2.86  1788    1
m    0.45    0.00  0.31  0.11  0.36   1.07  4379    1
s1s 38.17    0.12  5.45 23.00 39.00  44.00  2136    1
s2s 90.95    0.36 16.76 40.00 96.00 104.00  2170    1

Samples were drawn using NUTS(diag_e) at Fri Jan 31 04:02:36 2020.
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 also looking good. The ration n_eff / n_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_hmgamma)
```

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

```
check_energy(fit_hmgamma)
```

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

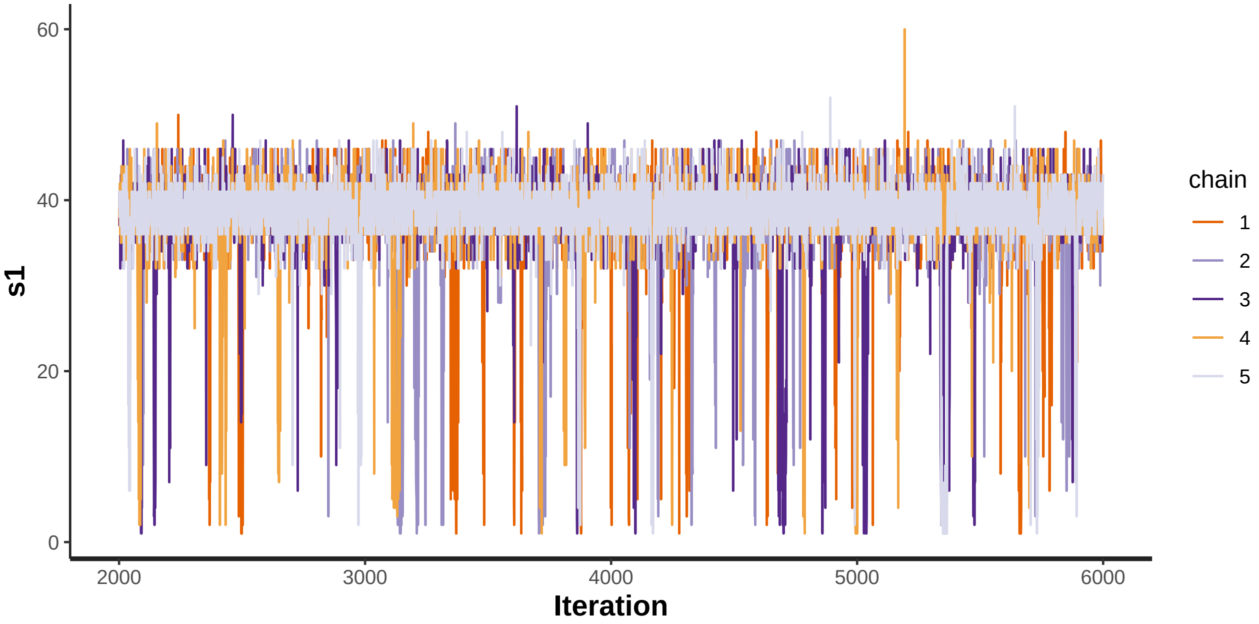
```
check_divergences(fit_hmgamma)
```

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

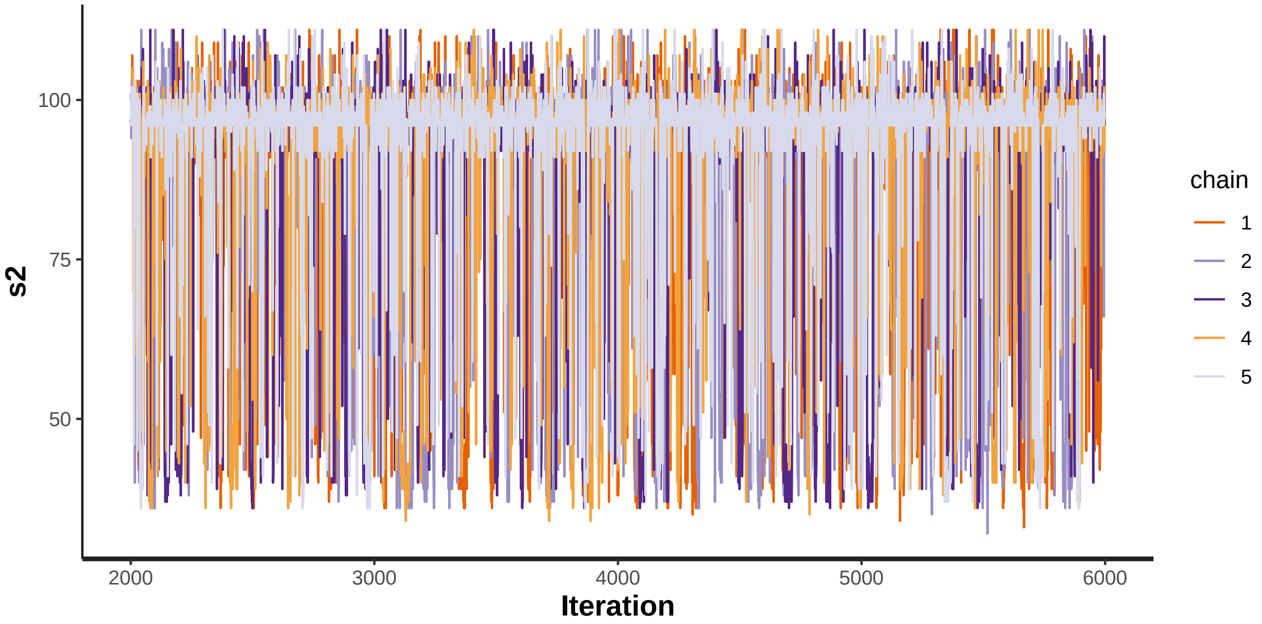
As we can see from the following plots of the samples, the plot of the chains of the mean of the first interval indicates that it is converged. The other means of the second and third intervals are also converged, but there are some divergences.

The same case, as in the Non-hierarchical model, happens here. For the 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 that the divergence of the second changepoint. As the histograms of the changepoints also show, there is a slight probability that the first changepoint is closer to the beginning of data. Based on that, there is some probability that the second changepoint is around 40. So, as in the previous model, this posterior distribution of the changepoints causes the divergence of the chains.

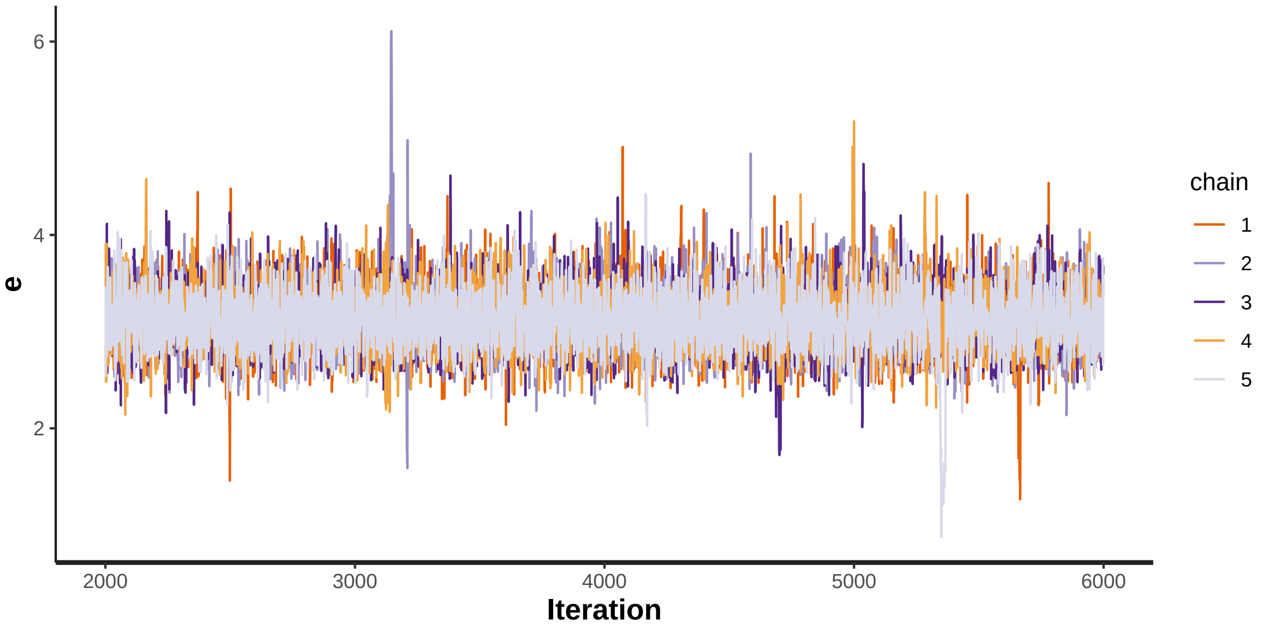
```
traceplot(fit_hmgamma, pars = c("s1s")) + labs(x = "Iteration", y = "s1")
```



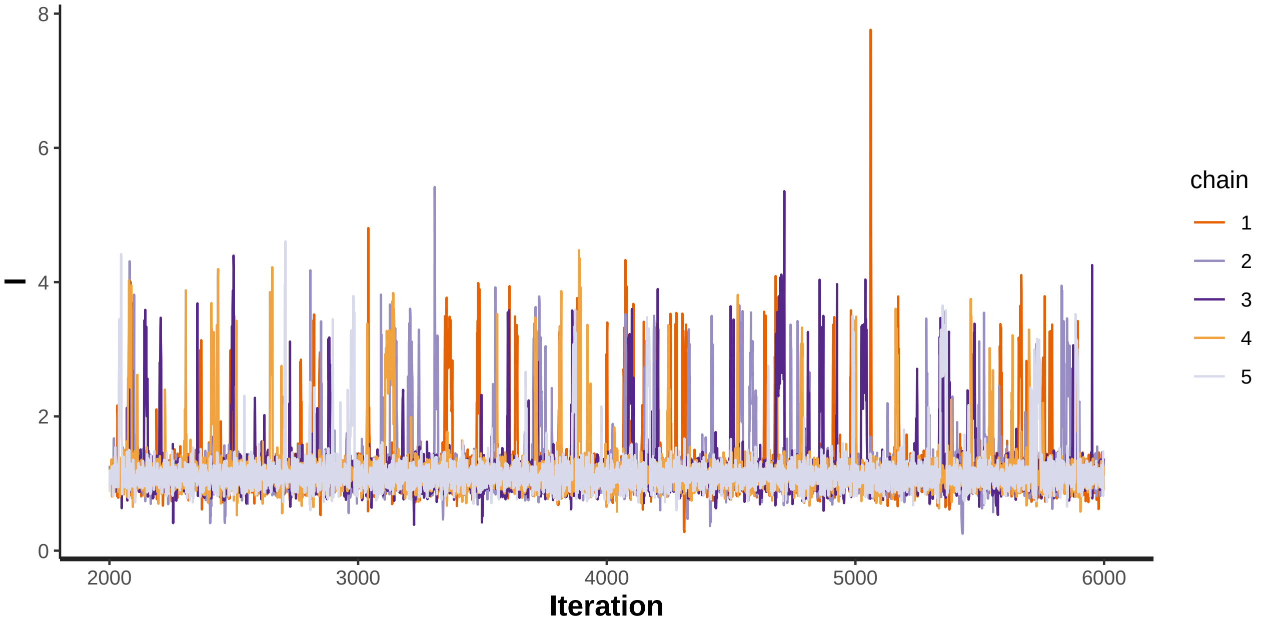
```
traceplot(fit_hmgamma, pars = c("s2s")) + labs(x = "Iteration", y = "s2")
```



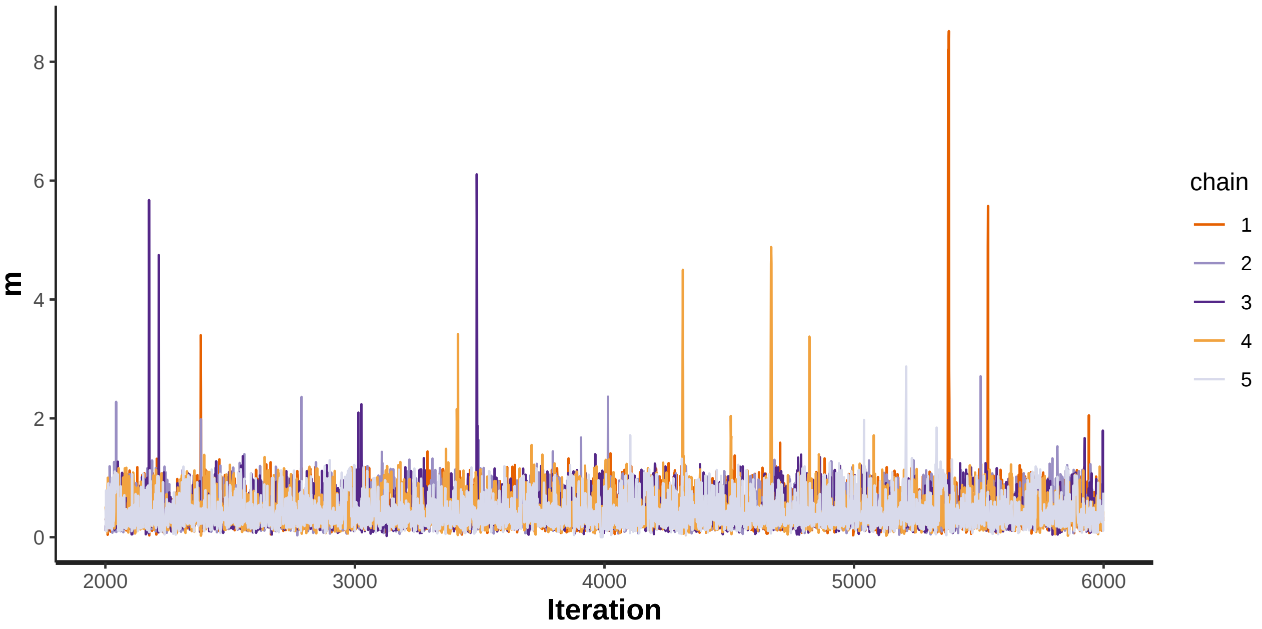
```r
traceplot(fit_hmgamma, pars = c("e")) + labs(x = "Iteration", y = "e")
```



```r
traceplot(fit_hmgamma, pars = c("l")) + labs(x = "Iteration", y = "l")
```



```r
traceplot(fit_hmgamma, pars = c("m")) + labs(x = "Iteration", y = "m")
```



6.3 Bayesian Online Changepoint Detection

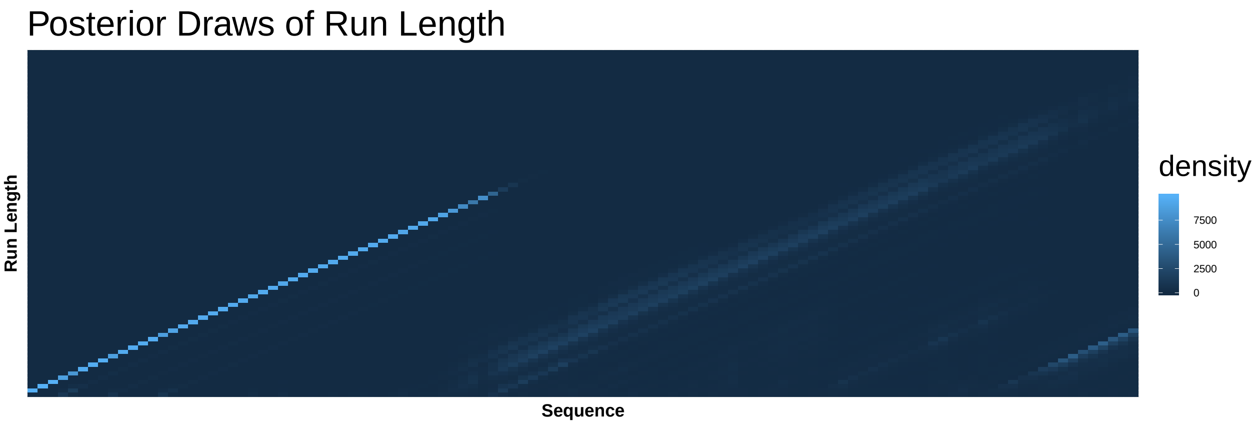
6.3.1 Posterior Heatmaps

The following cell of code plots the posterior heatmaps of the Bayesian online changepoint detection algorithm.

```r
plot_online <- function(draws){

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

  x <- paste0("t", seq(1,dim(draws)[2]))
  y <- paste0("r", seq(1,max(draws)))
  posterior_data <- expand.grid(X = x, Y = y)
  posterior_data$density <- c(r)

  # Heatmap
  ggplot(posterior_data, aes(X, Y, fill = density)) +
    geom_tile() + theme(
      title = element_text(size=25),
      axis.text.x = element_blank(),
      axis.text.y = element_blank(),
      axis.title = element_text(size = 15, face = "bold"),
      axis.ticks = element_blank()
    ) + labs(title = "Posterior Draws of Run Length",
             x = "Sequence", y = "Run Length")
}
```

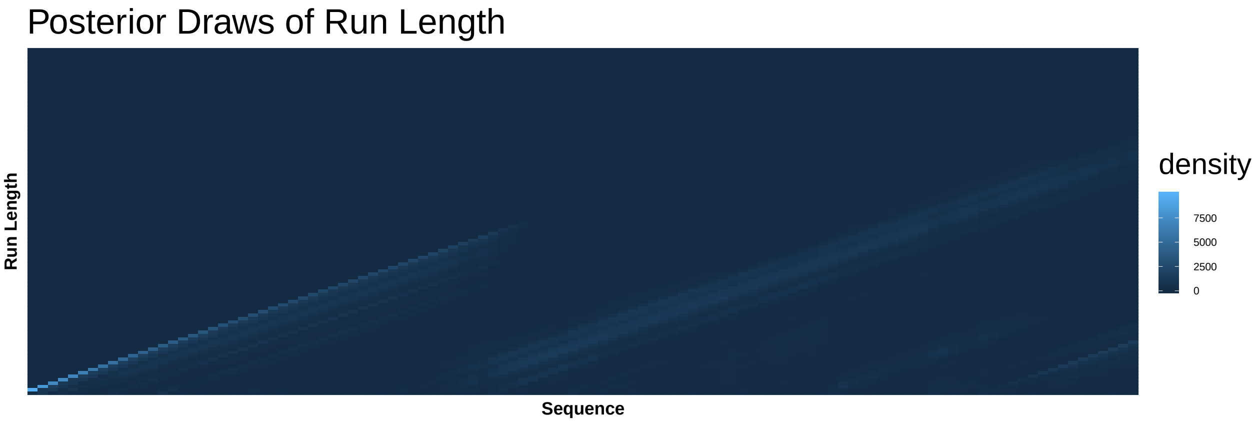
6.3.1.1 Analytical Solution

```r
plot_online(draws_analytic)
```



6.3.1.2 MCMC Solution

```
plot_online(draws_mcmc)
```



6.3.2 Convergence Diagnostics

```
fit_analytic_summary <- summary(fit_analytic, pars = c('run_length'),
                                digits_summary = 2, include = TRUE)
length(fit_analytic_summary$summary[fit_analytic_summary$summary[,"Rhat"] > 1.05, "Rhat"])
```

```
[1] 0
```

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

```
fit_mcmc_summary <- summary(fit_mcmc, pars = c('run_length'),
                            digits_summary = 2, include = TRUE)
min(fit_mcmc_summary$summary[,"n_eff"],na.rm = TRUE)
```

```
[1] 7067.467
```

```
length(fit_mcmc_summary$summary[fit_mcmc_summary$summary[,"Rhat"] > 1.05, "Rhat"])
```

```
[1] 0
```

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

The effective sample sizes are also looking good. The ration n_eff / n_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_mcmc)
```

```
0 of 10000 iterations saturated the maximum tree depth of 10.
```

```
check_energy(fit_mcmc)
```

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

```
check_divergences(fit_mcmc)
```

```
0 of 10000 iterations ended with a divergence.
```

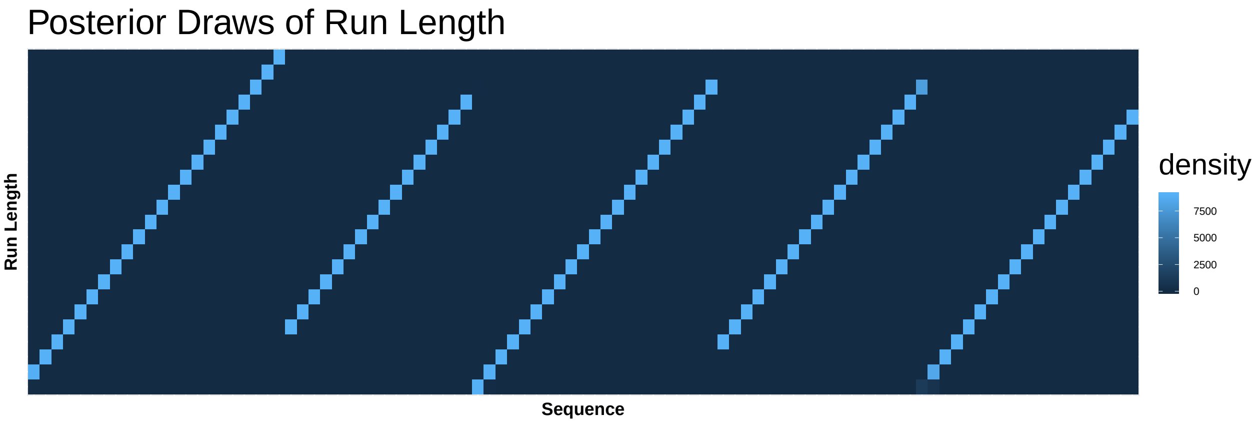
7 Bayesian Online Changepoint Detection (Syntetic Data)

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

7.1 Analytical Solution

```
num_chains <- 5
fit_syntetic_analytic <- stan("StanFiles/online_analytic.stan",
                              data = data_synthetic, algorithm = "Fixed_param",
                              iter = 2000, warmup = 200,
                              chains = num_chains)
draws_syntetic_analytic <- extract(fit_syntetic_analytic, pars=c("run_length"))$run_length
```

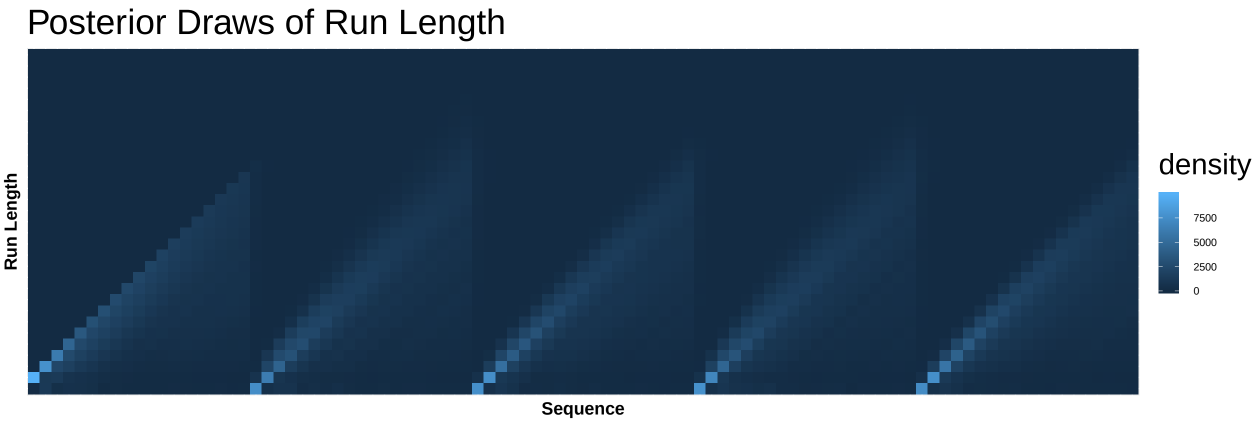
```
plot_online(draws_syntetic_analytic)
```



7.2 MCMC Solution

```
num_chains <- 5
fit_syntetic_mcmc <- stan("StanFiles/online_mcmc.stan",
                          data = data_synthetic,
                          iter = 2500, warmup = 500,
                          chains = num_chains)
draws_syntetic_mcmc <- extract(fit_syntetic_mcmc, pars=c("run_length"))$run_length
```

```
plot_online(draws_syntetic_mcmc)
```



8 Conclusion

In our project, we showed that the changepoints on the coal mining disasters dataset. We use it because it is widely agreed in the statistical literature, and it has changepoint in the year of the 1890 (40th index). Both of our models solve this changepoint almost precisely. On the other hand, all of our models find a second, slightly weak changepoint. It could be expected because our first two models are biassed on finding two changepoints. What we did not expect is that the bayesian online changepoint detection algorithm also finds the exact same changepoint. Typically, this dataset is used for the one changepoint detection; however, we showed that this data actually has two changepoints.

As a starting point, we compared the multiple changepoint model (MCM) and a hierarchical version of multiple changepoint model(HMCM). MCM and HMCM are performing similarly. However, when we compare the standard deviations of the parameters for MCM and HMCM, we see that HMCM has a slightly lower standard deviation then MCM.

In the Bayesian online changepoint model (BOCM), we have compared our MCMC draws results with the analytical solution of the problem because it is not possible to directly compare BOCM with other models since they do not share the same likelihood. BOCM approaches the changepoint problem with a different idea and calculates the posterior distribution of the run-lengths of the subsequences, instead of solving the changepoint between subsequences.

In this project, we implemented Poisson observation and Gamma underlying process version of the algorithm. We specifically used a conjugate prior model to compare with the analytical solution. As future work, more generic versions of the MCMC solution of the model can be implemented because it can be used with models that do not have conjugate priors. This allows for more complex models.

Overall, Bayesian Online Changepoint Model is superior to other models. It can find any number of changepoints in linear time, and it converges much faster. One drawback of the BOCM is memory usage because it keeps the log of entire possible run-lengths. The implementation of the Sequential Importance Sampling could solve this problem.

In our work, we made several contributions. We implemented a multiple changepoint model for two changepoints efficiently. More importantly, we wrote the STAN code of the online Bayesian changepoint model, both analytical and MCMC versions.